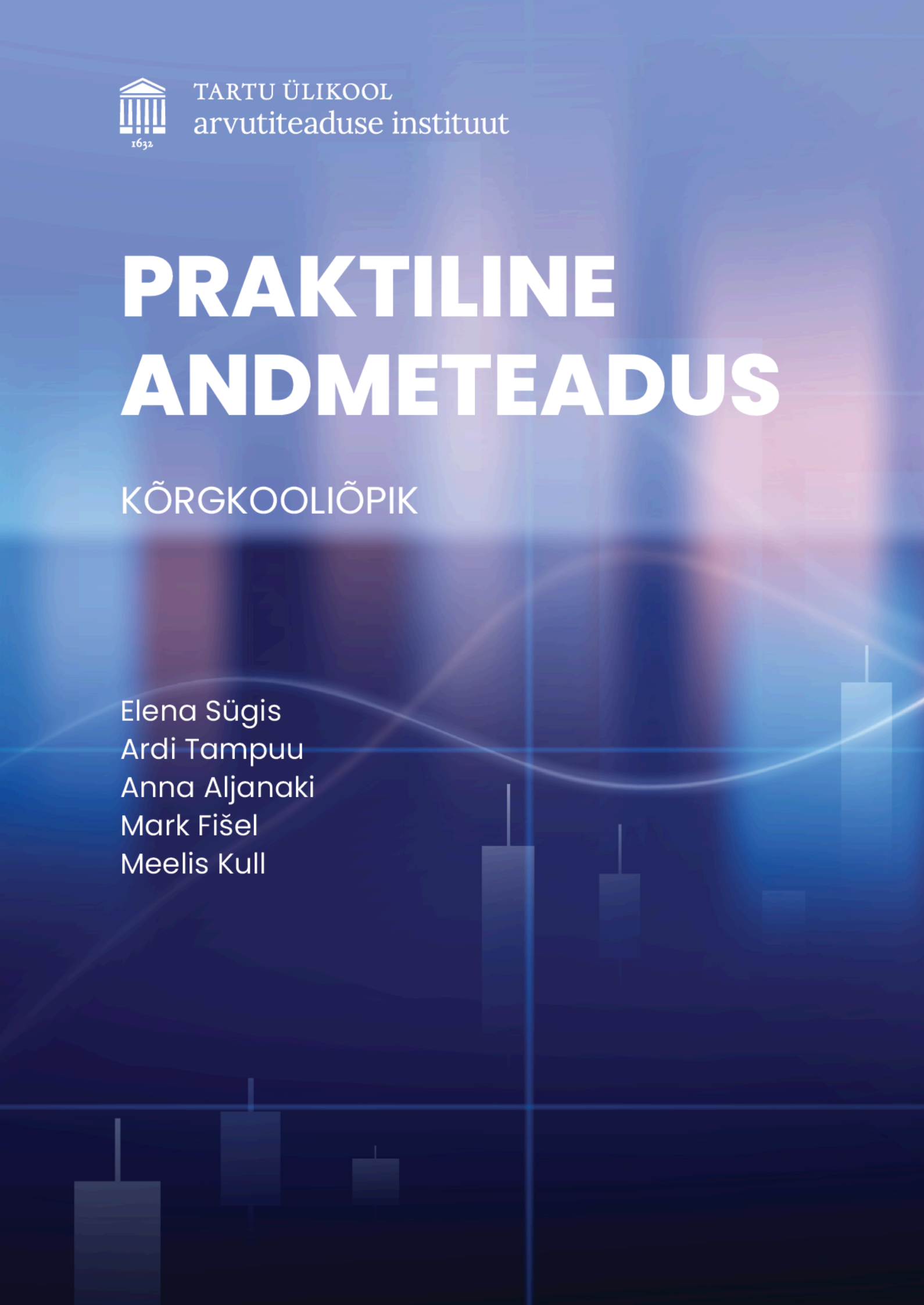


TARTU ÜLIKOOL
arvutiteaduse instituut

PRAKTILINE ANDMETEADUS

KÕRGKOOLIÕPIK

Elena Sügis
Ardi Tampuu
Anna Aljanaki
Mark Fišel
Meelis Kull

Praktiline andmeteadus

Kõrgkooliõpik

Elena Sügis
Ardi Tampuu
Anna Aljanaki
Mark Fišel
Meelis Kull

Õpiku väljaandmist on toetatud Haridus- ja Teadusministeeriumi programmist „Eestikeelsete kõrgkooliõpikute loomise toetamise põhimõtted 2018–2027“.

Retsensendid: Innar Liiv, Janika Aan, Sven Laur, Jaak Vilo, Anastassia Kolde
Autorid: Elena Sügis, Ardi Tampuu, Anna Aljanaki, Mark Fišel, Meelis Kull

Keeletoimetaja: Eva Saul

Väljaandja: Tartu Ülikooli arvutiteaduse instituut
ISBN 978-9985-4-1453-8 (pdf)
Välja antud 2024. aastal

Eessõna

Meil on hea meel, et teie kätte on jõudnud õpik „Praktiline andmeteadus“!

See raamat on loodud nii üliõpilastele kui ka andmeteaduse entusiastidele ja valdkonna ekspertidele, kes soovivad oma teadmisi selles valdkonnas süvendada. Meie eesmärk on pakkuda tasakaalustatud ülevaadet andmeteadusest, keskendudes nii teoreetilistele aspektidele kui ka praktilistele rakendustele.

Andmeteadus on dünaamiline ja kiiresti arenev valdkond, mis ühendab statistika, masinõppe, arvutiteaduse ja eri valdkondade teadmised. Selles õpikus me ei käsitle ainult teoreetilisi käsitusviise, vaid pakume ka praktilisi koodinäiteid ja juhtumiuuringuid. Need näited illustreerivad, kuidas andmeid uurida, reaalsetel andmetel statistilisi teste ja masinõppe mudeleid rakendada ning saadud tulemusi tõlgendada, et teha teadlikke ja põhjendatud otsuseid.

Iga peatükk sisaldab praktilisi harjutusi ja näpunäiteid, mis aitavad õpitud raamistikke ja meetodeid rakendada. Kasutame populaarseid andmeteaduse tööriistu ja tarkvara nagu Python ja Google Colaboratory, et anda teile praktilisi oskusi, mida saate kasutada igapäevases töös. Näitame, kuidas kasutada neid tööriistu andmete puhastamiseks, analüüsimiseks ja visualiseerimiseks, samuti masinõppe mudelite loomiseks ja hindamiseks.

Erilist tähelepanu pöörame CRISP-DM (*cross-industry standard process for data mining*) raamistikule, mis on üks populaarsemaid lähenemisi andmeteadusprojektide läbiviimiseks. CRISP-DM raamistik koosneb kuuest etapist: äriliste eesmärkide seadmine, andmete mõistmine, andmete ettevalmistamine, mudeldamine, tulemuste hindamine ja juurutamine. Meie õpikus uurime iga etappi põhjalikult, pakkudes praktilisi juhiseid ja näiteid, kuidas neid samme oma projektides tõhusalt rakendada. Näitame, kuidas CRISP-DM aitab struktureerida ja suunata teie andmeteadusprojekte, tagades, et kõik olulised aspektid on arvesse võetud ja projektid on edukad.

Meie eesmärk on muuta keerulised statistilised ja andmeteaduse käsitlusviisid arusaadavaks igale lugejale. Olenemata sellest, kas olete algaja või kogenud andmeteadlane, leiate siit väärtuslikke infoallikaid ja teadmisi, mis aitavad teil oma karjääri edendada ja andmeteaduse projektides edukalt hakkama saada.

Täname teid, et olete valinud selle õpiku oma andmeteadusõpingute kaaslaseks. Loodame, et see raamat inspireerib teid ning annab teile vajalikud teadmised, oskused ja vahendid, et saada andmetest tõelist väärtust.

Parimate soovidega
õpiku autorid

Sisukord

1. Sissejuhatus andmeteadusesse

1.1 Mis on andmed?

Enne kui räägime andmeteadusest, peaksime mõtlema, mis on **andmed**. Eesti keele seletav sõnaraamat defineerib need järgmiselt: „Andmed on informatsioon kellegi või millegi kohta, faktid, mida kellegi või millegi kohta teada saadakse või teatakse.“

Tänapäeval mõeldakse andmete all tihtilugu **digiteeritud andmeid**. Kui kellelgi on kodus vihik, kus on kirjas viimase 50 aasta ilmaolude kirjeldus, siis see tõesti sisaldab andmeid – arve temperatuuri, sademete jne kohta. Aga nende andmetega ei saa suurt midagi teha enne, kui need on digiteeritud ehk arvutisse sisestatud. Tõepoolest, võiks ju ka kalkulaatorit kasutades arvutada kuude keskmisi ja joonistada vihikulehele graafikuid ning ka see oleks andmete kirjeldamine ja uurimine (ehk andmeteaduse osa). Aga teha seda käsitsi sadade või tuhandete arvudega muutub kiiresti keeruliseks, tüütuks ja veaaltiks, samal ajal kui ka kõige lihtsamate arvutiprogrammidega saab neid toiminguid edukalt automatiseerida.

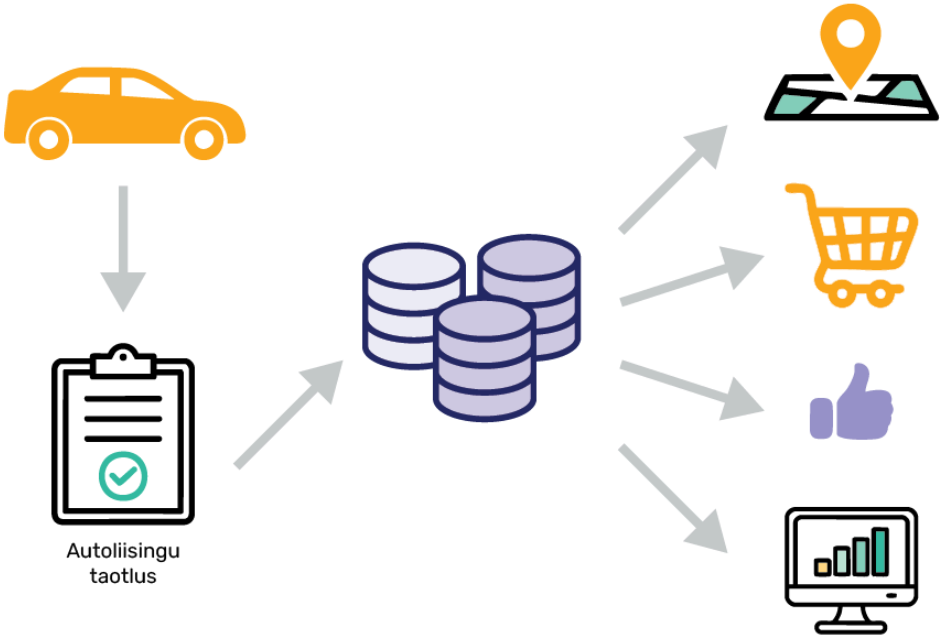
Ei ole üllatav, et **andmete maht kasvab** ajas, sest varem tekkinud andmeid talletatakse ja uusi andmeid salvestatakse pidevalt juurde. Lihtsaimaks näiteks on toidupoed: iga kaup, mis kassas läbi lüüakse, salvestatakse andmebaasi, ja nii iga kassa ning kaupluseketi poe kohta. Mida rohkem inimesed tarbivad, seda suuremaks see andmebaas kasvab. Selle toidupoe juhul on tegemist juba digiteeritud andmetega, mis on esitatavad tabelina, kus read vastavad mingitele vaatluskordadele/objektidele/isikutele ja veerud mingitele vaatlustulemustele/mõõtudele. Antud juhul võiks ridades olla ostetud tooted ja veergudes erinevad mõõtmised/tunnused/muutujad – kliendi ID, ostuaeg, toote hind, toote kaal jne (tabel 1.1).

Unikaalne ID	Klient ID	Kuupäev	Toode	Ühik	Ühiku hind	Kogus
1	Klient 1	01/06/2020	banaan	kg	1.09	0.8
2	Klient 1	01/06/2020	leib	tk	0.77	1.0
3	Klient 2	23/09/2020	šampoon	tk	3.45	1.0
4	Klient 3	24/09/2020	grill-liha	kg	8.18	1.5

Tabel 1.1 Ostuandmete näidistabel. *Üldjuhul vastavad read vaatluskordadele ja veerud vaatlustulemustele.*

Pange tähele, et eelnevas seadsime andmetabelitele olulise tingimuse: need peavad olema **struktureeritud**. Kokkuvõttes peame andmeteks ikkagi mingi loogika järgi kogutud ja digitaalselt talletatud infot. Digiteeritud andmed on näiteks ka pildid, tekstid ja helifailid. Ka neid andmeid on võimalik struktureeritud tabelite kujul esitada.

Nagu öeldud, praegusel ajal me kogume rohkem andmeid kui kunagi varem. Andmete suurem kättesaadavus on ka üks põhjus, miks andmeteaduse valdkond nüüd nii populaarne on. Näiteks, oletame, et te soovite osta autot ja täidate selleks kõik vajalikud dokumendid (joonis 1.1). Kõik need andmed sisestatakse töövoo käigus arvutisse ja talletatakse andmebaasi, kuhu on koondatud ka kõikide teiste autoostjate andmed. Kui need andmed on olemas, on edasi juba lihtne kasutada teie sisestatud e-postiaadressi, et siduda autoostu andmed teie sotsiaalmeedia või veebikasutuse ajaloo andmetega. Selline andmekogu võimaldab saada põhjaliku ülevaate kõikidest inimestest, kes on näiteks viimase aasta jooksul auto ostnud: nende vanus, meeldimised sotsiaalmeedias, kes on nende perekond ja sõbrad, jne. Kogu see lisainfo võimaldab ennustada, kui palju te saaksite uue auto eest maksta, mida võiksite veel juurde osta või kuidas teile kõige paremini autokindlustust müüa. Sarnaseid andmestikke on tänapäeval paljudel suurtel ettevõtetel ning pidevalt lisandub uusi teenusepakkujaid, veebiplatvorme ja seadmeid, mis inimestele paremate teenuste pakkumiseks nende kohta infot salvestavad.



Joonis 1.1. Andmed on igal pool meie ümber.

Ülesanne
Mõelge, milliseid andmeid teie ise kogute. Millised neist on digiteeritud? Millised neist on rohkem struktureeritud ehk ühtlasel ja arvuti jaoks üheselt mõistetaval kujul?
Ühe võimaliku vastuse leiate siit.

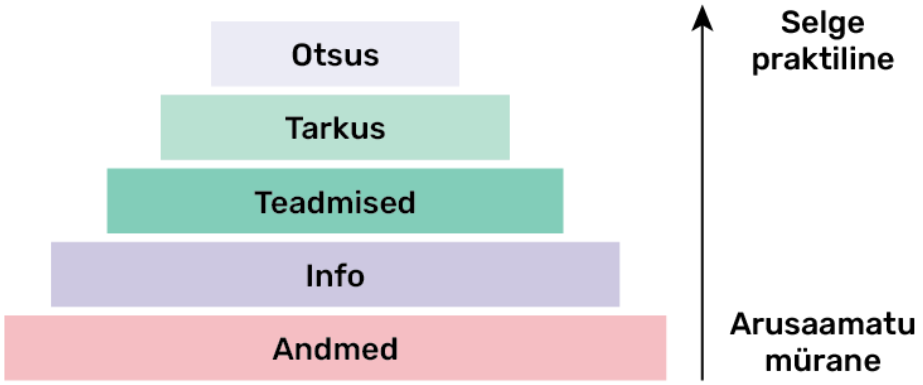
1.2 Mis on andmeteadus?

„Andmeteaduse eesmärk on andmete põhjal paremate otsuste tegemine."

Hästi kogutud andmed võimaldavad

- kirjeldada mingi protsessi hetkeseisu;
- tuvastada anomaaliaid;
- tuvastada mingite sündmuste või käitumismustrite põhjuseid;
- prognoosida tulevikusündmusi.

Andmeteadus on interdistsiplinaarne valdkond, mis rakendab teaduslikke meetodeid, algoritme ja mõtteviisi, et saada andmetest kätte **informatsiooni,** mille abil teha tähelepanekuid tegelike sündmuste ja seaduspärasuste kohta. Saadud **teadmised** võimaldavad omakorda teha **paremaid otsuseid** (joonis 1.2).



Joonis 1.2. Andmeteaduse eesmärk. *Andmeteadus võtab andmed ja rakendab neile algoritme, mille tulemusel saab teha targemaid otsuseid. Andmeteaduse protsess algandmetest otsustusetapini:* ***andmed*** *– kättesaadavad andmed, mille alusel otsust teha, näiteks viimase kuue kuu ostude ajalugu;* ***info*** *– andmete analüüsimise tulemused, näiteks „viimase kuu aja jooksul oli naissoost klientide keskmine ostusumma 10% suurem kui meestel";* ***teadmised*** *– paljude infokildude põhjal loodud üldistavad järeldused, näiteks „vahetult enne pühi on naissoost klientide ostusummad üldjuhul suuremad". See pole vaatlustulemus teatud andmestikul, vaid üldistatud järeldus. Samuti on näiteks viimase kolme kuu käibe suurus andmetest eraldatud „info", aga „teadmine" on sellest järeldatud fakt „poe käive langeb" või „Tartu poe käive kasvab kiiremini kui Tallinna poe oma". Märksõnadeks on trendid ja seaduspärad;* ***tarkus*** *– teadmiste põhjal tekkinud taju sellest, mida ja miks edasi teha. Kasutades eelmist näidet: Kas vallandada Tallinna poest üks müüja? Võib-olla on käibelangus ajutine? Võib-olla halvendab teenindajate vähesus müüki veelgi? Selliste otsuste tegemisel ei piisa kindlasti mõnest arvutatud keskmisest. Tarkus on tihti intuitiivne ja kogemusepõhine, aga see saab kõige paremini tekkida siis, kui otsuseid ei tehta pimesi, ilma infot teadmata, vaid teadlikult;* ***otsus*** *– lõplik omandatud tarkuse ja info põhjal tehtud otsus konkreetses olukorras.*

Saadud teadmised ja tarkus võivad olla formaliseeritud ka mingi mudeli kujul, mis suudab uue info põhjal ennustusi ning vajaduse korral ka otsuseid teha.

Andmeteaduse mõiste on veidi umbmäärane, selle alla kuulub kogu tegevus, mis aitab andmete põhjal kasulikke otsuseid teha – kirjeldav statistika, andmete ettevalmistamine ja analüüs, visualiseerimine (graafikute ja jooniste tegemine), masinõpe ja veel palju

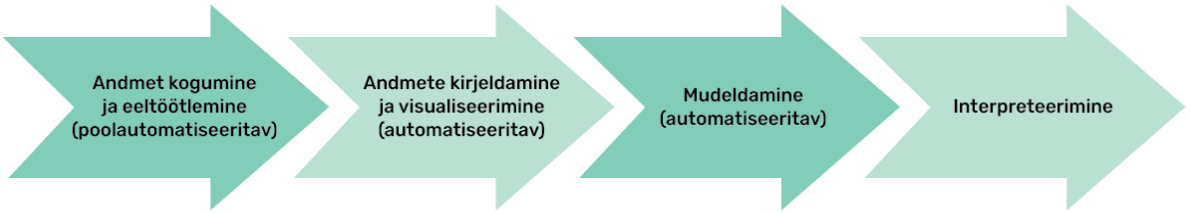
muud. Ka mõisted andmekaeve, suurandmed, andmebaaside haldamine, tehisintellekt, protsesside automatiseerimine jne seostuvad andmeteadusega. Andmeteaduse meetodid on lähedalt seotud ka valdkondadega nagu tarkvaraarendus (programmeerimine, paralleelarvutus) ja matemaatika (eriti optimeerimine, informatsiooniteooria ja kõrgedimensionaalsed ruumid). Vahel öeldakse ka, et andmeteadus on lihtsalt statistika ja IT kokkusulamisel tekkinud valdkond. Siiski on tänapäevase rakendusliku andmeteaduse ja statistika vahel selged erinevused, seega pole andmeteadus lihtsalt digirevolutsiooni läbi teinud statistika.

NB!
Andmeteadus on interdistsiplinaarne valdkond, mis rakendab teaduslikke meetodeid, algoritme ja mõtteviisi, et saada andmetest kätte informatsiooni, mille abil teha tähelepanekuid tegelike sündmuste ja seaduspärasuste kohta.

Kokkuvõttes on andmeteadus lai mõiste, mis on kasutusele võetud seni eraldiseisvatena vaadeldud valdkondade ühiseks nimetamiseks, sest need valdkonnad on tihti omavahel väga läbi põimunud ja saanud lahutamatuteks. Näiteks on masinõppe tavapraktika osa ka andmete esmane analüüs, visualiseerimine ja lõplike tulemuste statistiline analüüs – nii ongi lihtsam öelda selliste protsessidega tegeleva inimese kohta **andmeteadlane**, mitte statistik, andmeanalüütik või masinõppe spetsialist.

Kõige üldisemalt vaadates koosneb iga andmeteaduse projekt neljast suuremast osast (joonis 1.3):

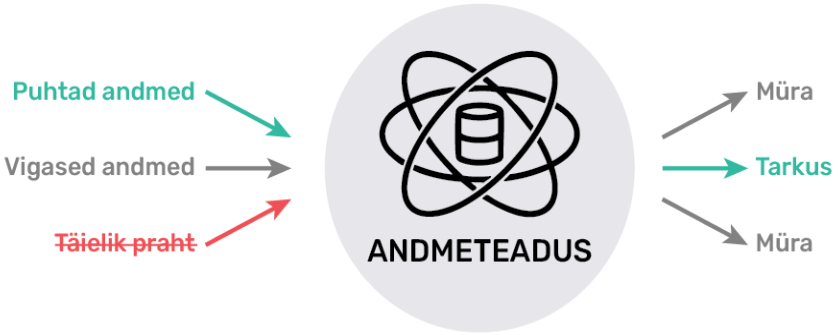
1) andmete kogumine;
2) kogutud andmete kirjeldamine ja visualiseerimine;
3) andmete mudeldamine, et näiteks tuleviku näitajaid prognoosida;
4) tulemuste interpreteerimine ehk tõlgendamine.



Joonis 1.3. Andmeteaduse tööprotsess.

Joonisel 1.3 on kujutatud andmeteaduse rakendamise üldine vooskeem. Andmeteaduse rakendamine algab sobilike andmete kogumisest vajalikul kujul. Andmeteadlaste seas on tuntud väljend „prügi sisse, prügi välja“ (ingl k *garbage in garbage out*) – kui andmed, mida analüüsitakse, on puudulikud, ei saa ka analüüsist oodata head tulemust (joonis 1.4). Seega on andmete kogumise planeerimine („andmete disainimine“, ingl k *data design*) väga oluline etapp andmeteaduse rakendamises. Ärivaldkonnas kasutatakse suurte andmestike ja andmevoogude kirjeldamiseks nelja V mudelit (ingl k *volume, variety, velocity, veracity*) ehk eesti keeles maht, mitmekesisus, kiirus ja tõesus,

millest viimane viitab andmete kvaliteedi olulisusele, sarnaselt „prügi sisse, prügi välja“ motoga.



Joonis 1.4. Prügi sisse, prügi välja. *Mittekvaliteetsete andmete alusel andmeteaduse tegemine ei lõpe tarkuse saavutamisega, vaid mingite juhuslike tulemustega, mis tegelikku olukorda ei peegelda.*

Pärast andmete kogumist on tihti vaja neid eeltöödelda. Väga oluliseks ja palju aega võtvaks sammuks võib osutuda andmete puhastamine – mingid väärtused võivad olla valesti sisestatud või puudu. Enne kui andmetele saab mudeleid rakendada, tuleb vigased andmed tuvastada ja eemaldada või mingil muul viisil korda teha. Samuti pole andmed tihti esitatud struktureeritud kujul, mida arvutiprogrammid lugeda suudaksid.

Järgmine väga tähtis samm andmeteaduslikus protsessis on andmete esmane uurimine ja visualiseerimine. On oluline teada, kuidas andmed on jaotunud, mis on erinevate näitajate keskmised ja kui suur on väärtuste varieeruvus. Selle kõige paremaks tajumiseks kasutatakse jooniseid ja graafikuid. Visualiseerimisest räägime täpsemalt kolmandas peatükis.

Alles pärast andmete kvaliteedi ja puhtuse kindlaks tegemist ning andmetega tutvumist visualiseerimise abil rakendatakse neile sobivaid analüüsimeetodeid. Seda tehes lähtutakse konkreetsest küsimusest ja probleemist. Tehtud analüüsi tulemused on mingid arvud või graafikud, mitte lõplikud otsused. Tulemusi on vaja tõlgendada, et need aitaksid näiteks ettevõtte juhtidel teha paremaid otsuseid või teadlasel oma eksperimendi tulemusi mõista.

Kõiki neid vajalikke samme aitab loogilisse järjekorda panna ja töid planeerida CRISP-DM raamistik (ingl k *cross-industriy standard process for data mining* ehk valdkondadeülene andmekaeve standardprotsess), mida selle raamatu järgmises peatükis pikemalt tutvustame.

Soovitame kuulata ka head arutelu teemal „Mis on andmeteadus?“ (1,5 h), mis peeti 2019. aasta arvamusfestivalil. Selle arutelu osalisteks on inimesed, kes tegelevad andmeteaduse eri tahkudega – statistikud ja masinõppe eksperdid (Mart Mägi, Ene-Margit Tiit, Taivo Pungas, Meelis Kull, Krista Fischer).

Enesekontrolli küsimused

1) Milline järgmistest väidetest on tõene andmete kohta?
    a) Andmed on alati struktureeritud tabelite kujul.
    b) Digiteeritud andmed võivad sisaldada pilte, tekste ja helifaile.
    c) Andmeid saab kasutada ainult arvutisüsteemide abil.
    d) Kõik digiteeritud andmed on automaatselt kvaliteetsed.

2) Andmeteaduse peamine eesmärk on andmete põhjal _____.
    e) andmebaase luua
    f) tarkvaraarendust lihtsustada
    g) paremate otsuste tegemine
    h) andmete kogumine

3) Miks on andmete kvaliteet andmeteaduse protsessis oluline ja kuidas "prügi sisse, prügi välja" (garbage in, garbage out) põhimõte seda selgitab? Milliseid samme saab astuda, et vältida andmete kvaliteediprobleeme?

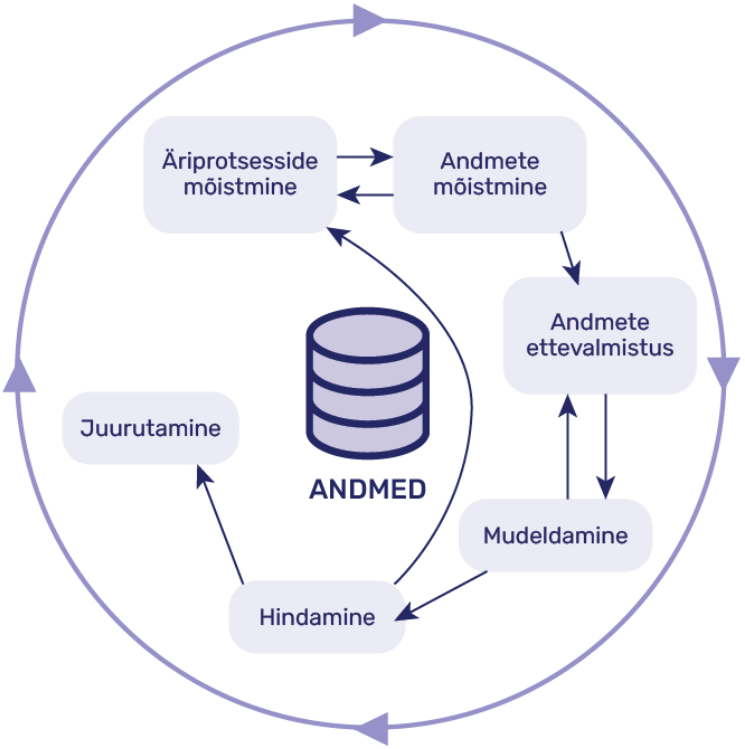
2. CRISP-DM metoodika

CRISP-DM (ingl k *cross-industry standard process for data mining*) ehk valdkondadeülene andmekaeve standardprotsess on 1999. aastal kolme suure ettevõtte (NCR, SPSS ja DaimlerChrysler) initsiatiivil välja töötatud ühtne protsessimudel andmekaeve rakendamiseks ükskõik millises valdkonnas. Mudeli originaalkirjeldus on saadaval siin.

Praktiseerivad andmeteadlased avastasid, et eri valdkondade andmetega seotud projektid on protseduurilt paljuski sarnased, ja soovisid defineerida kindla protsessi, mis standardiseeriks lähenemist, minimeeriks vigu ja arusaamatusi ning viiks paremate tulemusteni. Sooviti, et andmeteadlane ei peaks olema ekspert valdkonnas, kust andmed pärinevad, see defineeritud protsess peaks lubama tal analüüsida ükskõik millise valdkonna andmeid. CRISP-DM on nüüdseks kasutusel olnud juba kaks aastakümmet ja on ennast tõestanud kui väga efektiivne lähenemine, mida rakendatakse nii äri- kui ka teadusvaldkondades, kus andmete analüüs on olulisel kohal. Kuigi nimetuse järgi arendati CRISP-DM raamistik algselt andmekaeve projektide jaoks, on see üldistatav ka teistsugustele andmeteaduse projektidele.

CRISP-DM koosneb reast järjestikustest sammudest, mis annavad andmeteaduse rakendamisele formaalse raamistiku ning aitavad kogu seda protsessi tugevamalt siduda konkreetsete eesmärkidega ja loodud teadmiste või mudelite hilisema rakendamisega. Selline piiritlemine on vajalik, sest andmeid on võimalik analüüsida lõputul hulgal viisidel ja seeläbi genereerida väga palju tulemusi, millest pole otsuste tegemisel lõpuks mingit kasu. CRISP-DM mudeli järgi algab projekti planeerimine ärilise, teadusliku või muud tüüpi projekti vajaduste selge väljaselgitamisega ning lõpeb tulemuste rakendamise läbi mõtlemisega, et püstitatud eesmärke ka reaalselt täita. On oluline, et kui analüüsi tehakse näiteks ettevõtte jaoks, siis firma ärilisi küsimusi ja eripära tundvad inimesed ning analüütikud teeksid koostööd, et saavutada ühine arusaam, mida ja kuidas on vaja analüüsida.

CRISP-DM metoodika võib jagada kuueks suuremaks etapiks (joonis 2.1), mida enamasti hakatakse lahendama antud järjekorras, aga tihtilugu pöördutakse nendes sammudes ka tagasi, sest alati on võimalik avastada lisanüansse, millega peaks analüüsis arvestama. Joonisel 2.1 on nooltega märgitud põhilised etapid, mille vahel on sageli tarvis edasi-tagasi liikuda. Väline ring sümboliseerib seda, et andmeteaduse projekt üldjuhul ei lõpe siis, kui lahendus on kasutusele võetud, vaid analüüsi käigus saadud õppetunnid ja kogemused võivad tekitada uusi, sageli konkreetsemaid küsimusi.



Joonis 2.1. CRISP-DM protsessimudel.

Lühidalt saab etappe kirjeldada järgmiselt:

1) **eesmärkide seadmine** – projekti eesmärkide püstitamine nii valdkonnaspetsiifiliste terminite kaudu (näiteks käive, kliendi poolt tagasi saadetud toodete osakaal, ajasääst lahendust kasutades jne) kui ka andmeteaduslike mõistete abil (näiteks ennustustäpsus). Esialgse plaani kindlaks määramine;
2) **andmete mõistmine** – see etapp algab andmete kogumisega ning sisaldab tegevust juba olemas olevate andmete hulga, hoiustamisviisi ja struktuuri tuvastamiseks. Samuti tuvastatakse võimalikud andmekvaliteedi probleemid;
3) **andmete ettevalmistamine** – kasulike andmete välja valimine, andmete puhastamine, andmetabelite loomine ja ühendamine. Selle etapi lõpuks tuleks andmed viia sellisele kujule, mida suudavad kasutada andmeanalüüsi tarkvara ja algoritmid;
4) **mudeldamine** – sobivate meetodite ja tehnikate valimine ning nende rakendamine eelmises sammus ette valmistatud andmetele. Lisaks hinnatakse selles etapis loodud mudelite sobivust matemaatiliselt, vastavalt andmeteaduslikele mõõdikutele nagu täpsus, keskmine ruutviga jne;
5) **tulemuste hindamine** – selleks etapiks on andmeanalüütiliste kriteeriumite järgi heaks kiidetud tulemid juba välja valitud ja nüüd hinnatakse neid rakendusvaldkonna vaatest. Olenevalt eesmärgist võib projekti lõpptulemus olla väga erinev, alates lihtsalt analüüsi lõppraportist kuni keerukate masinõppe rakenduste püsti panekuni. Kontrollitakse üle kõik varem tehtud sammud ja eeldused ning hinnatakse, kas kõik on rakendusvaldkonna terminites püstitatud eesmärkidega vastavuses. Selle etapi lõpuks valitakse toimiv lahendus, mida saab eesmärgipäraselt kasutada;

6) **kasutusele võtmine ehk juurutamine** – saadud teadmiste või mudelite rakendamine otsuste langetamisel igapäevatöös, plaani loomine tulemuste paikapidavuse jälgimiseks aja jooksul ja lahenduse hooldamiseks tulevikus. Analüüside ja mudelite sisendiks olevad andmejaotused ja neis peituvad seaduspärad võivad ajas muutuda ning ükski mudel pole igavene. On vaja kokku leppida, kes ja kuidas lahendusi uuendab.

CRISP-DM-i etapid on intuitiivsed ja kasulikud eelkõige andmeteaduse projekti planeerimiseks. Toodud etapikirjelduste järgi on võimalik koostada küllaltki detailne projektiplaan ja see ongi CRISP-DM protsessimudeli põhiline rakendus. Vaatame järgnevas neid etappe eraldi veidi detailsemalt.

2.1 Esimene etapp – kliendi eesmärkide seadmine

Selles etapis määratakse kindlaks, milliseid ärilisi või muu valdkonna spetsiifilisi eesmärke soovitakse andmete analüüsi abil saavutada. Ilma seda etappi korralikult läbi mõtlemata võib juhtuda, et kulutatakse palju ressursse, et saada täpsed vastused valedele küsimustele. Nende eesmärkide püstitamiseks on antud ka mõned pidepunktid, millest järgnevas räägimegi.

2.1.1 Mis on kogu projekti soovitud väljundid?

Eesmärkide püstitamine seisneb projekti peamiste sihtide kirjeldamises oma tegevusvaldkonna vaatenurgast. Sealjuures võib tekkida ka kõrvaleesmärke. Näiteks võib peamiseks eesmärgiks olla olemasolevate klientide hoidmine, mida loodetakse saavutada, tuvastades ennetavalt kõige tõenäolisemalt konkurendi juurde lahkuda võivad kliendid, et neile soodustusi pakkudes neid lahkumast hoida. Analüüsi kõrvaltulemustena võib lisaks tekkida hulk kasulikke andmekirjeldusi: milline on keskmine tulu, mida iga klient aastas toodab, mis on uute klientide omandamise hind (reklaamikulu kliendi kohta) jne. Kaasatud peaks saama kõik osapooled (juhid, osakonnajuhid, analüütikud jne), et tagada projekti eesmärkide vastavus ettevõtte strateegilistele vajadustele ja kõigi osapoolte hilisem koostöö ühise eesmärgi nimel.

Edasi on vaja **defineerida projekti edukuse kriteeriumid**. See tähendab, et juba enne andmeanalüüsiga alustamist otsustatakse, mis on vähim äriline tulemus, mille puhul projekt õnnestunuks loetakse. Need tulemused peaksid olema selgelt piiritletud ja mõõdetavad, näiteks „vähendada aasta jooksul lahkunud klientide arvu 5% võrra võrreldes eelmise aastaga“. Andmeid kirjeldavates projektides on vaja neid eesmärke vahel ka kvalitatiivsemal kujul kirja panna, näiteks „projekt loetakse edukalt lõppenuks, kui on loodud kasulikud arusaamad firma klienditüüpide ja nende vajaduste kohta“. Sellisel juhul peab juba projekti alguses selge olema, kes langetab subjektiivse otsuse, et tulemused olid kasulikud. Kui tegu on pikema projektiga, võivad ettevõtte eesmärgid ja turuolukord ajas muutuda ning projekti võib sisse planeerida eesmärkide regulaarse uuendamise.

2.1.2 Hetkeolukorra hindamine

Selles alametapis selgitatakse välja kõik **andmeanalüüsi mõjutada võivad tegurid**. Näiteks olemasolevad ressursid (andmed, arvutusvõimsus, inimressurss), võimalikud piirangud (privaatsed andmed) ja eeldused (eeldame, et teades potentsiaalseid lahkujaid, suudame neist 50% veenda mitte lahkuma). Nendest asjaoludest sõltub projekti plaan ja kliendi eesmärkide „tõlkimine“ andmeteaduslikeks eesmärkideks.

Koostatakse projektile pühendatud:

- **ressursside loetelu:** kaasatud töötajad (valdkonna eksperdid, andmeteadlased, tehniline tugi jne), andmed (mis mahus, mis formaadis), saadaval olevad arvutid/serverid, tarkvara;
- **nõuded, eeldused, piirangud:** nimekiri nõuetest projektile – valmimistähtajad, tulemused ja nende mõistetavus, andmete turvaline hoiustamine. Nimekiri piirangutest, näiteks „kas andmeid tohib analüüsi teostava firmaga jagada?“. Nimekiri eeldustest, millest osa analüüsi käigus ka kontrollitakse (nt eeldus, et aastaaeg ei mõjuta läbimüüki), aga osa võivad ka olla mittetõestatavad eeldused nagu see, mitu protsenti klientidest on soodustuse tõttu nõus teenusepakkujat mitte vahetama. Eeldus on ka see, et olemasoleva info põhjal on üldse võimalik vajaliku täpsusega mudel luua;
- **ohud ja varuplaanid:** riskianalüüs võimalike viivituste või läbikukkumiste põhjuste kohta projekti teostamisel ja plaan, mida ette võtta, kui need juhtuvad;
- **terminid:** on vaja luua sõnastik olulistest terminitest, mida kliendi tegevusvaldkonna ja andmeteaduse poolelt projekti raames kasutatakse. Vastasel juhul on valdkonna ekspertide ja andmeteadlaste suhtlus raskendatud;
- **kulu-tulu analüüs:** mõista projektile kuluvate ressursside kulu ja võimaliku teenitud tulu vahekorda. See analüüs võiks olla võimalikult täpne, näiteks arvestades loodetud tulemused ümber rahalisse vääringusse.

2.1.3 Andmekaeve/andmeanalüüsi eesmärkide seadmine

Punktis 1.1 defineeritud eesmärgid on sõnastatud valdkonnaspetsiifilistes terminites, kasutades rakendusvaldkonnas tavapäraseid mõõdikuid. **Andmeteaduslikud eesmärgid** defineerivad oodatud tulemused tehniliste terminite kaudu. See on punkt, kus andmeteadlased ja valdkonna eksperdid (nt ettevõtjad, teadlased, arstid) peavad tegema koostööd ja suutma üksteist mõista.

Näiteks, kui äriline eesmärk oli „olemasolevate klientide hoidmine“ või „lahkuvate klientide hulga vähendamine 10% võrra aastal 2025 võrreldes aastaga 2024“, siis andmeteaduslik eesmärk on „kasutada viimase kolme aasta ostuajalugu, demograafilist infot (sugu, vanus, elukoht) ja muid andmeid, et tuvastada varem lahkunud klientide profiilid, et sarnaseid kliente tulevikus ära tunda ja neile soodustusi pakkuda, lootes neid seeläbi siiski hoida. Lahkujate tuvastamise täpsust mõõdetakse viimase kuue kuu andmetel ning eesmärgiks on saavutada 80% ennustustäpsus ja 80% saagis.“

Andmeteaduslike eesmärkide seadmine sõltub ärilistest eeldustest ja faktidest. Kui otsustame lahkumisohus klientidele soodustusi pakkuda, siis 80-protsendilise

ennustustäpsuse puhul pakume me viiendikul juhtudest soodustust kliendile, kes tegelikult pole lahkumisohus, ja seeläbi kaotame raha. Kui soodustuste efektiivsus klientide hoidmisel on madal (mis on meie sellekohane eeldus?) ja ennustustäpsus väike (palju asjatuid soodustusi), võib firma hoopis raha kaotada. Madal saagis tähendaks aga, et meil jääb palju lahkujaid tuvastamata. Kui täpne peab olema mudel, et see tegelikult äriliselt kasulik oleks, sõltub tehtud eeldustest (nt eeldatav soodustuste efektiivsus, kasumimarginaal). Andmeteadusliku eesmärgi püstitamisel tuleb arvesse võtta tegelikku kasulikkust rakendusvaldkonnas.

2.1.4 Projektiplaani kirjalik koostamine

Kirjeldatakse **projekti esialgne plaan**, et andmeteaduslikud ja seeläbi ka valdkonnaspetsiifilised (nt ärilised) eesmärgid saaksid täidetud. See plaan peab sisaldama kõiki samme kogu ülejäänud projektis, sealhulgas esialgset töövahendite ja meetodite valikut. Projektiplaanis sõnastatakse kõik sammud, mis projekti jooksul tuleb teha, koos nende kestuse, vajalike ressursside, sisendite ja väljundite ning riskihinnangutega. Põhimõtteliselt on see alametapp vajalik selle jaoks, et pärast eesmärkide seadmist saaksid need ka formaalselt kirja pandud koos edasiste CRISP-DM-i etappide plaaniga.

2.2 Teine etapp – andmete mõistmine

CRISP-DM-i teises etapis hakkavad andmeteadlased lõpuks andmetega tõsisemat tööd tegema. Selle ja järgnevate etappide tegevus peaks olema mõistlikkuse piires detailsel tasemel ette planeeritud ning kokku lepitud juba punktis 1.4 koostatud projektiplaanis.

2.2.1 Andmete kättesaamine või kogumine

Kõigepealt tuleb andmeteadlastel saada **ligipääs** projekti ressursside all loetletud **andmetele** ning teha kindlaks, kas kõik failid avanevad ja on kasutatavad nii, nagu ette nähtud. Vajaduse korral tuleb mitu andmeallikat ühendada (nt tabel ostude ajalooga ja tabel klientide demograafilise infoga või tabelid andmetega sama firma kahest kauplusest).

Luuakse esmane **andmete kogumise raport** (ingl k *data collection report*[1]) – nimekiri andmeallikatest, millele on saadud ligipääs. Kirja pannakse nende asukohad, nende kätte saamise viisid (nt kas on salasõnad, kas on vaja teisest osakonnast eraldi luba küsida, kas saab ligi ainult peakontori sisevõrgus viibides) ja tuvastatud probleemid koos leitud lahendustega. Näiteks: „Failidel puudus failiformaati näitav laiend (.pdf, .xml vms), tuvastasime, et kõik failid on tegelikult .csv tüüpi, ja täiendasime failinimesid“. See raport aitab tulevikus projekti täpselt korrata või sarnast projekti läbi viia.

2.2.2 Andmete kirjeldamine

Lisaks tehakse **andmete** pinnapealne kirjeldus, kirjeldades nende **mahtu** (mitu gigabaiti? mitme kliendi kohta andmeid on? kui palju on infot iga kliendi kohta?),

[1] https://www.ibm.com/docs/en/spss-modeler/saas?topic=understanding-describing-data.

formaati (Exceli tabel, andmebaas, tekstifail), **kvaliteeti** (kui palju on puuduvaid väärtusi?), mis **mõõdetud tunnused** on olemas (koostatakse täielik nimekiri) jne. Koostatakse **andmete kirjelduse raport** (ingl k *data description report*[2]), kus hinnatakse ka andmete piisavust eesmärkide täitmiseks.

2.2.3 Andmete uurimine, visualiseerimine ja kirjeldamine

Selles etapis vastatakse andmeteaduslikele küsimustele lihtsate meetodite abil nagu **päringute esitamine**, **andmete visualiseerimine** ja **raporteerimine**.

See hõlmab näiteks järgmist:

- oluliste tunnuste jaotuste uurimine (nt klientide vanuste jaotus);
- olulisemate tunnusepaaride omavaheliste seoste uurimine (kas keskmine ostusumma korreleerub kliendi vanusega?);
- olulisemate alamgruppide omaduste kirjeldamine (kas meeste ostusummad on suuremad? kas ühe poe ostusummad erinevad teise poe summadest?);
- lihtsam statistiline analüüs.

Need **kirjeldavad analüüsid** võivad juba otseselt vastata mõnele 1. punktis seatud eesmärkidest. Tihtilugu aga aitavad need pigem andmeid ja nende kvaliteeti täpsemalt kirjeldada (andmete kirjelduse raportis) ning annavad aimu, kuidas oleks andmeid vaja töödelda enne põhjalikumat analüüsi. Näiteks võib vanuse jaotust visualiseerides märgata kliente, kelle vanus on 0, mis võib tähendada, et andmeid sisestades jättis inimene vanuse märkimata. Sellised puuduvad andmed võivad muutuda probleemiks hilisemas andmeanalüüsis.

Andmete uurimise raportis (ingl k *data exploration report*[3]) kirjeldatakse avastatud trende, seoseid ja puudusi ning hinnatakse **nende mõju** ülejäänud projektile. Näiteks terviseandmetes, kui 50% patsientide kohta puudub info vanuse kohta, tuleb patsientide haiguskulgu ennustav mudel luua kas ilma vanust arvesse võtmata või eemaldada need 50% patsientidest edasisest analüüsist. Kui selgub, et mingil osal andmetest leiti midagi huvitavat (nt, et kindel patsiendigrupp on suure haigusriskiga), tuleks seda kirjeldada konkreetsete jooniste ja arvudega.

2.2.4 Andmete kvaliteedis veendumine

Andmete kvaliteedi kontrollimiseks tuleb kindlaks teha, kas

- **andmed on täielikud** – need katavad kõiki projektis ette nähtud juhud. Näiteks, tuleks kontrollida, et andmetest ei puuduks talvised andmed või andmed mingite toodete kohta, mille müüki analüüs peaks ennustama;
- **andmed on õiged** – neis pole vigu (sisestamisel tehtud vigu). Kui leiduvad vead, siis kas need on tuvastatavad ja kui sagedased nad on;
- **andmetes esineb puuduvaid väärtusi** – kuidas need tähistatud on (kas tühja väljaga, nulliga, sõnaga „puudu“ või muul viisil). Selgitatakse välja, kui paljud read ja veerud sisaldavad puuduvaid väärtusi.

[2] https://www.ibm.com/docs/en/spss-modeler/saas?topic=data-writing-description-report.
[3] https://www.ibm.com/docs/en/spss-modeler/saas?topic=data-writing-exploration-report.

Andmekvaliteedi raport (ingl k *data quality report*[4]) koondab kõik need olulised tähelepanekud ja pakub välja võimalikud lahendused. Lahenduste valik sõltub valdkonna ja andmeteaduslikest teadmistest, näiteks võib meditsiiniekspert aimata, et ilma vanust arvestamata pole võimalik saada head tulemust. Samuti võib andmeteadlane aimata, et eemaldades analüüsist pooled patsiendid, pole võimalik saada üldistatavaid tulemusi.

2.3 Kolmas etapp – andmete ettevalmistamine

Kui esialgne tutvus andmetega on tehtud ja vajaduse korral seatud eesmärgid tuvastatud oludega kohandatud, siis **viiakse andmed sobivale kujule**, et neile projektiplaanis (ptk 1.4) kokku lepitud analüüsimeetodeid rakendada.

2.3.1 Andmete valimine

Andmete valimise etapis tuleb otsustada, millist **alamhulka** saadaval olevatest andmetest kasutada. Selle jaoks tuleb mõelda, kas olemasolevates andmetes sisalduv info on püstitatud eesmärkide saavutamiseks relevantne ja piisavalt kvaliteetne. See tähendab, et valitakse nii tunnuseid (vanus, ostusumma jne) kui ka näiteid (kliendid), mida edasises analüüsis kasutada. Näiteks võime otsustada analüüsist välja jätta mingist konkreetsest allikast tulevad andmed, sest neis on liiga palju valesti sisestatud väärtusi. Samuti võib olla mingite andmete kasutamine tehniliselt liiga keeruline, näiteks võivad andmed olla liiga mahukad. Põhjendused mingite andmete kaasamiseks või välja jätmiseks peavad raportis kirjas olema, et hiljem tehtut korrates või analüüsides ei tekiks korduvalt samu küsimusi.

Näide 1: ebakvaliteetsed andmed
Firma on kogunud andmeid klientide kohta. Kliendikaarti luues täidab klient andmevormi ja klienditeenindaja sisestab andmed hiljem arvutisse. Andmetest selgub, et enne 2018. aastat ei sisaldanud see vorm küsimust selle kohta, kas klient soovib reklaame. Samuti paistab, et üks andmeid sisestanud töötaja on teinud väga palju vigu. Andmete valimise etapis võib otsustada, et vastavalt seatud eesmärkidele on optimaalne jätta kasutamata andmed, mis pärinevad enne aastat 2018 ja mille on sisestanud see tähelepanematu töötaja (isegi kui mõned tema sisestatud ridadest on õiged, on vigu raske tuvastada).

Näide 2: liiga mahukad andmed
Soovime võrrelda 100 ettevõtte aktsiahinna muutust viimase aasta jooksul. Meil on olemas andmed iga aktsia hinnast sekundi täpsusega, mis võtab kõvakettal kokku ruumi üle 100 GB. Sellise andmemahuga töötamiseks oleks vaja arvutusklastrit või serverit, mille kasutamine oleks projekti eelarvet arvestades väga suur kulu. Siin peaks mõtlema, kas analüüsiks on vaja kasutada kõiki neid andmeid või piisab, kui jälgime hindu tunni täpsusega, ehk kasutame 3600 korda vähem andmeid.

[4] https://www.ibm.com/docs/en/spss-modeler/saas?topic=quality-writing-data-report.

2.3.2 Andmete puhastamine

Selle alametapi eesmärk on olemasolevate **andmete kvaliteedi tõstmine** tasemeni, mis on vajalik edasiseks analüüsiks. Nagu ennist mainitud, on üks variant selle saavutamiseks kasutada ainult mingit alamhulka andmetest. Aga on olemas ka meetodeid, mis võimaldavad puuduvad väärtused asendada (ehk imputeerida) mingi vaikimisi väärtusega, mis ei mõjuta edasise analüüsi tulemusi, näiteks tunnuse keskmise väärtusega. Kategooriliste väärtuste puhul võime lisada võimalike väärtuste hulka variandi „teadmata". Milline lahendus, kas eemaldamine või asendamine, on parem, sõltub andmestikust ja järgmises sammus kasutatavatest mudelitest, ning selle valiku tegemine on andmeteadlase töö osa. On ka keerukamaid meetodeid puuduvate väärtuste ennustamiseks olemasolevate väärtuste põhjal. Tavaliselt on aga probleemsete andmete eemaldamine kõige kindlam viis andmeid puhastada.

Andmete puhastamise raport (ingl k *data cleaning report*[5]) võtab kokku kõik andmete puhastamisega seotud otsused ja tegevuse, et neid saaks hiljem vajadusel korrata. Lisaks on oluline mõelda, mis võib olla rakendatud meetmete mõju lõpptulemuste usaldusväärsusele. Näiteks, kui asendame sisendandmetes kõik puuduvad „vanuse" väärtused kõigi klientide keskmise vanusega, siis kas on endiselt mõttekas võrrelda eri vanusegruppide keskmisi ostusummasid?

2.3.3 Uute andmete loomine

Vahel on vajalik mitme olemasoleva tunnuse põhjal arvutada uusi, **tuletatud tunnuseid** või olemasolevaid tunnuseid kuidagi sobivamaks muuta. Näiteks võib ostude arvu ja suuruse põhjal arvutada uue tunnuse „keskmine ostusumma" või ostude aega teades „keskmine ostusumma kuus". Samuti võib esineda juhte, kus on kasulik valitud tunnuste ühikuid muuta, näiteks ruutmeetrid arvutada hektaritesse. Mõne analüüsi eelduseks on aga, et erinevad tunnused omavad väärtusi samas suurusjärgus, mistõttu on vaja tunnuste väärtused enne analüüsi standardida.

Samuti võib olla vajalik ise luua täiesti uusi tabeliridu. Kui meil on klientide andmebaasis isikuid, kes pole sooritanud ühtegi ostu, võivad nende andmed meie olemasolevatest andmetabelitest puududa. Olenevalt analüüsi eesmärgist võib osutuda vajalikuks ka need nullostu sooritanud inimesed andmetesse kaasata.

2.3.4 Andmetabelite ühendamine

Sageli on andmed talletatud mitmesse erinevasse andmetabelisse. **Tabelite ühendamine** tähendab mitme samade objektide (nt klientide) kohta infot sisaldava tabeli üheks suureks tabeliks kokku panemist. Siinkohal eristame kahte tabelite ühendamise viisi.

Kahe sarnaseid andmeid (samu tunnuseid) sisaldava **tabeli liitmine** rida rea haaval on tehniliselt väga lihtne. Näiteks kahe poe ostude nimekirju saab lihtsalt liita, kopeerides

[5] https://www.ibm.com/docs/en/spss-modeler/saas?topic=data-writing-cleaning-report.

ühe tabeli read teise tabeli lõppu (eeldusel, et tabelid sisaldavad sama tähendusega veerge).

Samade objektide kohta erinevat infot sisaldavate tabelite ühendamine on keerulisem. Näiteks võib firmal olla tabel iga oma poe üldtunnuste kohta (pindala, asukoht, töötajate arv jne), teine tabel iga poe müügitulemuste kohta (läbimüük, läbimüük nädalavahetustel, aastane kasv jne) ning kolmas tabel täpsema infoga iga poe asukoha kohta (kesklinnas või äärelinnas, elanike vanus ja arv piirkonnas, elanikkonna jõukus jne). Tavajuhul on kõiki neid tabeleid võimalik kokku ühendada üheks „laiaks" tabeliks, kus iga rida vastab ühele poele ja veerud sisaldavad kogu infot, mis pärineb kõigist eelmainitud tabelitest. Nii võib aga tekkida liiga lai tabel, liiga paljude veergudega, millest kõik pole järgneva analüüsi jaoks vajalikud. Seega valitakse tabelite ühendamisel enamasti, millised veerud alles jätta.

2.4 Neljas etapp – mudeldamine

Kui andmed on valitud, puhastatud ja õigele kujule viidud, siis alustab andmeteadlane analüüsi ehk mudeldamisega.

> **NB!**
> Siin alapeatükis võite kohata termineid, mille tähendust te veel ei tea. Ärge selle pärast muretsege, peatükis 5 „Mudeldamine" keskendume juba detailsemalt sellele, mis on mudelid ja milliseid erinevaid mudeli tüüpe on olemas, ning räägime ka teistest mudeldamisega seotud mõistetest.

2.4.1 Mudeli tüübi valik

Kuigi juba esimeses etapis projektiplaani kirjutades otsustati üldsõnaliselt, millised on rakendatavad andmeteaduslikud tehnikad, oleme me eelnevate sammudega andmetest paremini aru saanud ja saame teha täpsemaid valikuid. Andmete olemusest ja hulgast võib sõltuda, millist **programmeerimiskeelt** või **tööriista** hakatakse kasutama ja millist **mudelitüüpi** hakatakse rakendama. Näiteks, kas visualiseerimine tehakse Excelis, R-is või Pythoni programmeerimiskeeles ja kas kasutatakse lineaarset mudelit, otsustusmetsi või tehisnärvivõrke. Lisaks on vaja selgelt kirja panna **mudeldamise** käigus tehtavad **eeldused** – paljud (statistilised) meetodid töötavad ainult juhul, kui mingid eeldused on täidetud. Näiteks, sõltuvalt meetodist, peavad mingid mõõtmised olema normaaljaotusega või andmetes ei tohi olla puuduvaid väärtusi. Mudeli valik määrab ka, kas tunnused peavad olema arvulised või kategoorilised. Kõik eeldused tuleb kirja panna, et vajaduse korral hiljem kontrollida, kas need ka paika pidasid.

2.4.2 Mudeli hindamismeetodi valik

Enne mudelite loomist tuleb veel planeerida, milline on protseduur **mudeli headuse hindamiseks**. Näiteks, kui luuakse masinõppemudel, mis peaks ennustama, kes klientidest võivad lähitulevikus lahkuda, siis võime jätta mudeli loomisest välja viimase

kolme kuu andmed. Selles ajavahemikus teame, kes päriselt lahkusid. Kasutame loodud mudelit, et nende kolme kuu andmetel ennustada, kes lahkub, ja saame ennustusi võrrelda tegelike andmetega. Nii on võimalik hinnata loodud mudeli täpsust „tuleviku andmetel".

Tüüpiliselt jagataksegi ettevalmistatud andmestik nn **treeninghulgaks** ja **testhulgaks** ehk andmeteks, millel mudeleid luuakse, ja andmeteks, millel nende edukust testitakse. Testimiseks mõeldud andmeid mudelite loomise käigus kasutada ei tohi, sest mõned mudelid võivad treenimisel kasutatud andmed teatud viisil „meelde jätta", samal ajal uutele andmetele üldistumata. Seda nimetatakse masinõppe keeles ülesobitamiseks ehk ületreenimiseks. Mudeli jaoks seni peidetuna hoitud andmete kasutamine testimisel aitab seda probleemi tuvastada ja üldistusvõimele õiglase hinnangu anda. See on oluline, sest mudel peabki lõpuks töötama uutel andmetel tulevikus, mitte ainult kirjeldama mineviku infot.

2.4.3 Mudeli loomine

Selles etapis **luuakse mudel** vastavalt kõigile eelnevates punktides defineeritud tingimustele.

Loodud mudeli kohta talletatakse järgmine info:

- **mudel** ise. Mudelit saab talletada eraldi failina, mida saab sama tarkvara kasutades uuesti sisse lugeda ja kasutada;
- mudeli loomiseks kasutatud **eeldefineeritud parameetrid** ehk hüperparameetrid. Mudelit luues fikseeritakse teatud parameetrite väärtused, mis mõjutavad mudeli suurust, kuju, õppimise kulgu ja seega ka lõpptulemust. See tähendab, et muutes mõnd sellise parameetri väärtust, muutuvad ka loodud mudel ja selle tulemused. Näiteks, on algoritme, millele tuleb ette öelda, mitu kliendigruppi nad otsida võiksid. Muutes seda hüperparameetrit, muutuvad ka tulemuseks saadud grupid. Et mudelit oleks võimalik taasluua, on oluline need otsused kirja panna ja neid ka põhjendada;
- **mudeli selgitused**. Siin pannakse kirja, kuidas tuleks loodud mudelit ja selle tulemusi tõlgendada ning mis on võimalikud (tihti inimeste igapäevaloogika alusel loomulikud) valetõlgendused või valearusaamad.

2.4.4 Mudeli hindamine

Selles etapis kasutatakse varem valitud kriteeriumeid, et loodud **mudeli võimekust/kvaliteeti hinnata**, ja saadakse tulemuseks konkreetsed arvulised hinnangud. Kasutades valdkonna teadmisi, on võimalik hinnata, kas mudel on piisav kliendi eesmärkide saavutamiseks, aga selles etapis hinnatakse mudeli töötamist ainult andmeteaduslikus võtmes.

Tavaliselt proovitakse mudeldamise jooksul mitut eri tüüpi mudelit mitme erineva hüperparameetriga. Loodud mudelid järjestatakse hindamiskriteeriumite alusel ja valitakse neist parim. Kui võimalik, võetakse arvesse ka rakendusvaldkonna teadmisi ja fakte. Näiteks, mudeli A üldine lahkujate ennustamise täpsus (andmeteaduslik

kriteerium) oli parem kui mudelil B, aga mudel B ennustas täpsemini väga kasumlike klientide lahkumist. Seega ärilisi teadmisi arvesse võttes on kasulikum kasutada mudeli A asemel mudelit B.

2.5 Viies etapp – projekti hindamine

2.5.1 Tulemuste hindamine

Eelmises etapis hinnati loodud lahendusi mõõdikutega nagu mudeli täpsus ja üldistusvõime. CRISP-DM-i viiendas etapis hinnatakse põhjalikumalt mudeli võimet saavutada püstitatud valdkonnaspetsiifilisi eesmärke ja otsitakse, kas mudel võib mingil põhjusel olla nende eesmärkide saavutamiseks puudulik. Puuduste teoreetilise otsimise asemel võib mudeli headust ka mõõta, proovides seda päriselus rakendada, kui see ei too kaasa liigseid riske ega võta liialt aega. Projekti hindamise etapis vaadatakse üle ka kõik muud avastused ja tulemused, mis projekti käigus, tihti planeerimatult, saadi. Seega andmeteaduse projekti tulemused pole ainult see info, mis on seotud esialgu projekti kirjutatud eesmärkidega, vaid kõik leiud, mis aitavad esile tuua mingeid uusi probleeme või võimalusi.

Projekti tulemuste hindamise lõppraport sisaldab hinnangut andmeteaduslikus mõttes parimate mudelite kasulikkusele rakendusvaldkonna, kasutusjuhu vaatenurgast. Raporti lõppsõna ütleb, kas tulemused saavutavad kliendi püstitatud eesmärgid või on vaja mudeleid muuta või parandada.

2.5.2 Kvaliteedikontroll

Selleks etapiks on läbi viidud analüüs, mis on loodetavasti piisavalt hea, et täita nii andmeteaduslikud kui ka valdkonnaspetsiifilised eesmärgid. Siinkohal on oluline veel kord läbi vaadata kogu andmete töötlemise ja ettevalmistamise etapp. Võib juhtuda, et andmetest unustati eemaldada mõni tunnus, mida ei tohiks analüüsis arvesse võtta. Üle tuleks vaadata ka kõigi eelduste paikapidavus. Selle etapi tulemusena peaksid olema selged kõik tehtud vead ja CRISP-DM-i sammud, mis tuleks uuesti ja paremini läbi teha.

2.5.3 Järgmiste sammude kindlaks määramine

Teades tulemusi ja võimalikke puudujääke kvaliteedis, tuleb otsustada, kas lõpetada projekt ja liikuda edasi tulemuste rakendamisele päris elus, või liikuda tagasi ning korrata mingeid etappe. Näiteks juhul, kui tulemused pole piisavalt head, võib muuta mudelitüüpi või mudelite parameetreid ja uuesti proovida. Samuti võib otsuseks olla, et on vaja koguda rohkem andmeid ja siis uuesti proovida. Kui tulemused tunduvad head, kuid meil on kahtlusi läbi viidud hindamiste piisavuses (kahju halva mudeli juurutamisel on suur, peame olema kindlad) võib koguda lisa-andmeid ainult lahenduse testimise eesmärgil. Muidugi tuleb hinnata, kas jätkamiseks on alles jäänud piisavalt ressursse (raha, tööaega).

2.6 Kuues etapp – juurutamine ehk kasutusele võtmine

Eduka projekti viimases etapis planeeritakse, kuidas saadud tulemused praktikas kasutusele võtta.

2.6.1 Juurutamise planeerimine

Juurutamine on äärmiselt tähtis samm, et projekt ettevõttele kasulik oleks. Juba projektiplaanis peab juurutamine läbi mõeldud olema, et kallihinnaline lahendus riiulile seisma ei jääks. Seega mõeldakse sellele juba üldisi eesmärke valides. Näiteks, sooviti potentsiaalseid lahkujaid ennustavat mudelit eesmärgiga neile 10%-list allahindlust pakkuda. Selleks on juba ette nähtud ressursid (nt inimesed) ja teavituskanalid. Analüüsi tulemusena tuvastatakse need kliendid, kellele ettevõte peaks allahindluspakkumise saatma.

2.6.2 Jälgimise ja haldamise planeerimine

Kui andmeteadust rakendavast lahendusest saab ettevõtte igapäevategevuse osa, siis on oluline, et see lahendus oleks alati töökorras ja et tulemused oleksid ka aja möödudes endiselt täpsed. Seega on kasutamise jooksul oluline pidevalt jälgida mudelite täpsust ja võimet saavutada ärilisi eesmärke. Kõige selle jaoks on vaja koostada koos andmeteadlastega plaan, kuidas lahenduse efektiivsust ja õigesti rakendamist jälgitakse ning kuidas vajaduse korral sekkuma peaks.

2.6.3 Lõppraporti koostamine

Projekti lõpus koostatakse lõppraport. Olenevalt juurutamisplaanist võib see raport või ettekanne lihtsalt veelkord kokku võtta kogu projekti ja selle käigus õpitu. Kui aga mudel on juba täielikult juurutatud ja tulemused teada, võib see olla ka viimane ja lõplik hinnang projekti edukusele.

2.6.4 Tagasivaade

Selle etapi tulem on kogetu ja õpitu kokkuvõte. Kirja tuleb panna kogemused, mis omandati. Näiteks võidi avastada, et teatud lähenemine tundus esialgu paljulubav, aga osutus petlikuks ja seda ei peaks tulevikus kasutama. Igasugused vihjed, milliseid lähenemisi järgmisel korral sarnaste andmetega kasutada, võivad olla selle kokkuvõtte osad.

2.7 CRISP-DM-i lõppsõna

CRISP-DM on projekti planeerimise ning läbi viimise raamistik, mis põhineb intuitsioonil ja loogikal. See on kasulik mõtteviis, et kaardistada ühe andmeteaduse projekti pidepunkte ja piiritleda selle projekti ulatus ehk skoop. Isegi kui praktiseeriv andmeteadlane ei koosta oma planeeringuid sihilikult CRISP-DM-i etappe järgides, esindavad need etapid kogenud andmeteadlase alateadlikku mõttekäiku kasumliku projekti läbiviimisel. Selle raamistiku kasutamine ei välista tööülesannete hilisemat jupitamist ja ülesannete haldamise tarkvara kasutamist, vaid pakub lisandväärtust

selgelt struktureeritud lähenemise, selguse ja parema tulemuslikkuse mõõtmise kaudu. Standardse lähenemise kasutamine võimaldab töötajaid koolitada.

Toome siinkohal veel kokkuvõtva tabeli CRISP-DM-i etappidest (tabel 2.1) koos nendele kuluva hinnangulise töömahuga ja osapooltega, kes vastavas etapis kõige rohkem panustavad. Siiski on kogu projekti eduka läbiviimise eelduseks osapoolte pidev koostöö.

Etapp	Ajakulu (%)	Alamprotsessid	Äripool	Andmeteadlane	IT
Äriliste eesmärkide seadmine	5–10	Eesmärkide püstiatmine, edukuse mõõdikute määratlemine	✓		
Andmete mõistmine	10–15	Andmete kogumine, uurimine, kvaliteedi hindamine	✓	✓	
Andmete ettevalmistamine	30–60	Andmete valimine, puhastamine, loomine, tabelite ühendamine		✓	✓
Mudeldamine	20–30	Tehnikate valimine, mudeli loomine, andmeteaduslik hinnang		✓	
Tulemuste hindamine	20–30	Äriliselt kasulikuima mudeli valimine, tulemuste interpreteerimine	✓	✓	
Juurutamine	5–10	Teadmiste rakendamine, lahenduse monitoorimine ja haldamine	✓	✓	✓

Tabel 2.1. CRISP-DM-i hinnaguline töömaht etappide kaupa.[6]

Tabel 2.1 esindab üsna hästi andmeteaduse projekti töömahu jaotust läbi etappide. Nagu näha, kulub kõige rohkem aega tavaliselt andmete ettevalmistamisele. Mõnes uuringus on leitud, et andmete kogumisele ja ettevalmistamisele kulub lausa 80% kogu projektile ette nähtud ajast. Selle ajakulu vähendamiseks oleks hea, kui andmeteadust tulevikus rakendada lootvad osapooled (nt riik, omavalitsused, ettevõtted) oleksid teadlikud andmete kogumise ja korraldamise headest tavadest, mida tutvustame peatükis 3. Täpsustuseks: kui ettevõtte IT-osakonnas või muudes osakondades on analüütikuid, peaks neid kindlasti kaasama ka eesmärkide seadmisel ja andmete mõistmisel.

CRISP-DM raamistiku rakendamist ettevõtetes takistavad andmepõhiste otsuste tegemise tava ja oskuste (õigete küsimuste esitamine, tulemuste tõlgendamine) puudumine, puudulik andmehaldus ning piiratud ressursid (andmed, aeg). Üldiselt peab ettevõte strateegilisel tasandil otsustama, et andmepõhiste otsuste tegemine on oluline, vastasel juhul ei leita ka vajalikke ressursse, ei võeta kasutusele hea analüüsi eelduseks olevaid andmehaldustavasid jne.

[6] Allikas: Wikipedia.org, link litsentsile.

Enesekontrolli küsimused

1) Järjesta CRISP-DM-i sammud
    a) Andmete ettevalmistamine
    b) Andmete visualiseerimine
    c) Mudeli loomine
    d) Probleemi püstitamine

2) Vii kokku tegevus ja CRISP-DM-i samm
    a) Mudeli loomine
    b) Andmete ettevalmistamine
    c) Andmete visualiseerimine
    d) Probleemi püstitamine

    e) Tulpdiagrammide koostamine
    f) Puuduvate väärtuste andmetest eemaldamine
    g) Suurendada käivet, kasutades kliendipõhiste ostusoovituste pakkumist
    h) R-i programmi loomine, mis jaotab kliendid erinevatesse gruppidesse

3) Lünktekst: _____ on valdkondadeülene andmekaeve standardprotsess.

4) Üks õige vastus. Andmete tulemuste hindamise etapis peavad osalema
    a) ainult andmeteadlane
    b) ainult tegevjuht või müügijuht
    c) nii andmeteadlane kui ka äripool

5) Vali kõik õiged vastused. Andmete kvaliteedi kontrollimiseks tuleb kindlaks teha,
    a) kas andmed on täielikud
    b) kas andmed on õiged
    c) kuidas tähistatakse andmetes esinevaid puuduvaid väärtusi

6) Mida järgnevatest tehakse CRISP-DM-i andmete ettevalmistamise etapil?
    a) andmete mudeldamine
    b) projekti tulemuste hindamine
    c) andmete puhastamine
    d) mudeli sisendiks sobiva tabeli koostamine

3. Andmete mõistmine

Andmete mõistmine hõlmab mitut olulist tegevust, sealhulgas andmete ülevaatamist, andmetes esinevate probleemide või ebakõlade tuvastamist ning sobivate andmete puhastamist ja eeltöötluse meetodite määramist. Selle CRISP-DM-i sammu jooksul peab andmeteadlane tuvastama ka puuduvad ja anomaalsed väärtused ning otsustama, kuidas nendega kõige paremini toime tulla. See on oluline samm tagamaks, et andmed sobivad analüüsiks ning mudeldamise tulemused on täpsed ja usaldusväärsed.

Selles peatükis keskendume meetoditele, mis võimaldavad olemasolevaid andmeid kirjeldada ja mõista. Milliseid meetodeid ja kus rakendada, sõltub aga andmetüübist. Seega alustame kõigepealt erinevate tunnuste tüüpide tutvustamisest.

3.1 Tunnused ja andmestikud

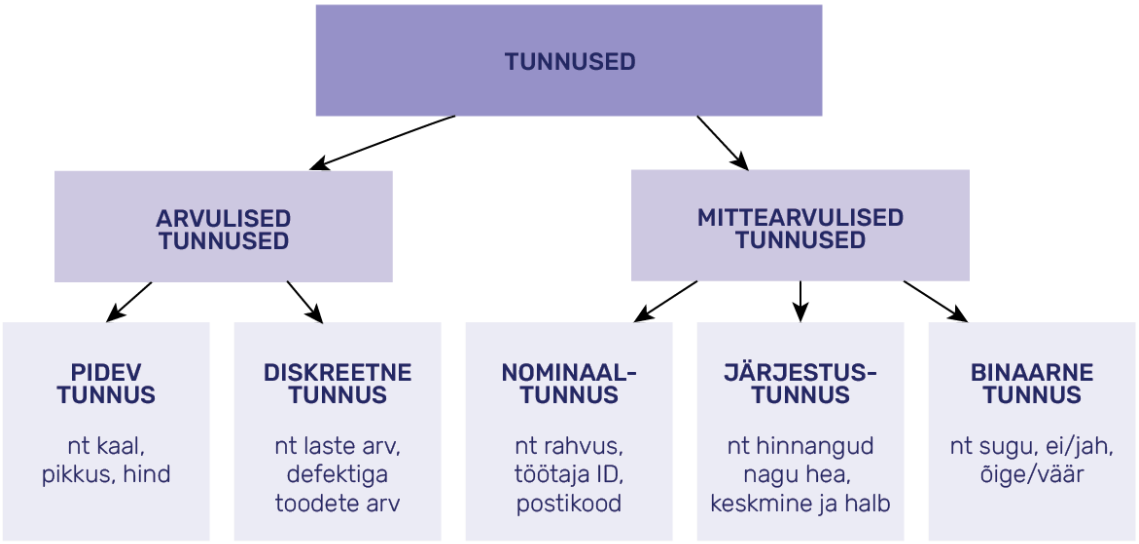
3.1.1 Tunnused ja tunnuste tüübid

Uurimisaluste objektide (ehk näidete, vaatluste; tavaliselt andmetabeli ridade) kohta mõõdetud omadusi nimetatakse **muutujateks** ehk **tunnusteks** (ingl k *attributes, features, variables*) ja üldjuhul vastab iga tunnus ühele veerule struktureeritud andmetabelis. Näiteks, kui vaatlusalusteks on kaupluse kliendid, võivad tunnusteks olla kliendi ID, nimi, aadress, e-postiaadress, sugu jne. Igaüks neist tunnustest erineb teistest selle poolest, milliseid väärtusi ta omandada võib. Vastavalt võimalike väärtuste hulgale võib tunnused jagada kaheks tüübiks: **arvulised tunnused** (nimetatakse ka kvantitatiivseteks tunnusteks) ja **mittearvulised tunnused** (nimetatakse ka kvalitatiivseteks ehk **kategoorilisteks tunnusteks**). Tunnuste jagunemine on kokkuvõtvalt illustreeritud joonisel 3.1.

Nagu nimigi ütleb, siis arvuliste tunnuste puhul on väärtuseks arv. Arvulised tunnused saab veel omakorda jaotada **pidevateks** ja **diskreetseteks tunnusteks**. Pideva arvtunnuse puhul tulevad väärtustena kõne alla kõik punktid tunnuse väärtuste vahemikus, enamasti saame me pidevaid arvulisi väärtusi midagi mõõtes. Tüüpilisteks näideteks on pikkus ja kaal, aga ka sissetulek ja toote hind. Diskreetsete arvtunnuste puhul on tunnuse võimalike väärtuste hulk piiratud – lubatud väärtuste vahel on mingi vahemaa. Selliseid tunnuseid saame enamasti midagi loendades. Näiteks laste arv peres ja defektiga toodete arv laos (mõlemad on positiivsed täisarvud, minimaalne vahemaa variantide vahel on 1), aga ka näiteks jalanõu suurus (mingi ettevõtte jaoks näiteks väärtused vahemikus 20–50, ⅓ suuruse sammuga).

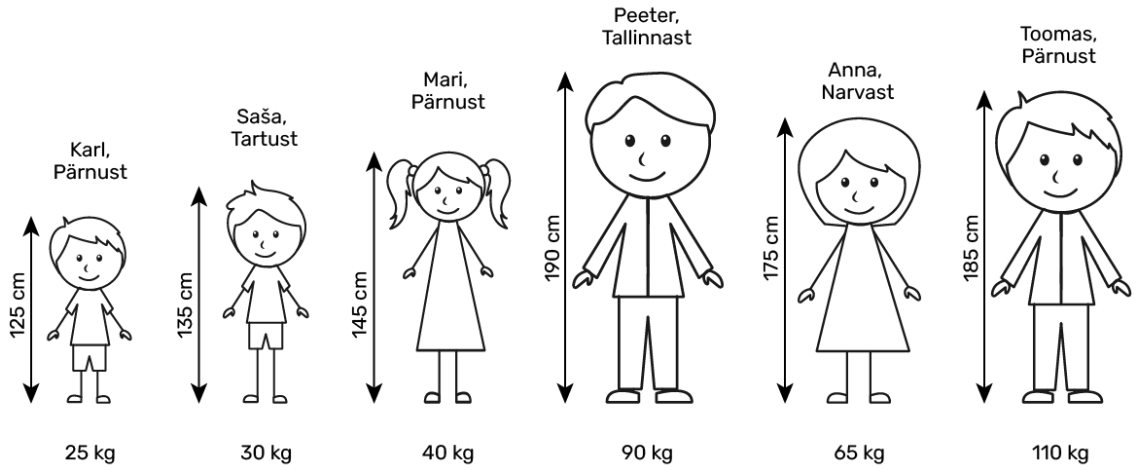
Kategoorilised tunnused saavad väärtusi mingist piiratud võimalike kategooriate hulgast. Need tunnused võib omakorda jaotada kaheks: **nominaaltunnused** ja **järjestustunnused**. Järjestustunnuste puhul on tunnuse väärtused mingi loogika põhjal järjestatavad. Tüüpilisteks järjestustunnuse näideteks on haridustase, igasugused meeldivuse ja rahulolu hinnangud. Nominaaltunnuste puhul ei ole lubatud vastusevariantide jaoks loomulikku (sisulist) järjestust. Nominaaltunnuse tüüpilisteks näideteks on inimese kodulinn, rahvus või perekonnaseis. Eraldi tüübina võib välja tuua

ka **binaarse** ehk kaheväärtuselise tunnuse. Tegemist on nominaaltunnusega, millel on ainult kaks võimalikku väärtust (nt kas uuritav isik on täiskasvanu või mitte).



Joonis 3.1. Tunnused ja nende tüübid.

Joonisel 3.2 kujutame mõningaid inimesi ja mõnda tunnust, mida me nende kohta teame. Mõned neist tunnustest on arvulised, mõned kategoorilised.



Joonis 3.2. Näide tunnuste tüüpidest. *Inimesi on võimalik kirjeldada mitme tunnuse abil. Joonisel olevate isikute kohta teame nende nime, elukohta, sugu, pikkust ja kaalu. On veel väga palju tunnuseid, mida võiks nende kohta teada, aga hetkel piirdume nende tunnustega.*

Ülesanne
Mis tüüpi on joonisel 3.2 näidatud tunnused? (Vastuseid saab kontrollida siit.)

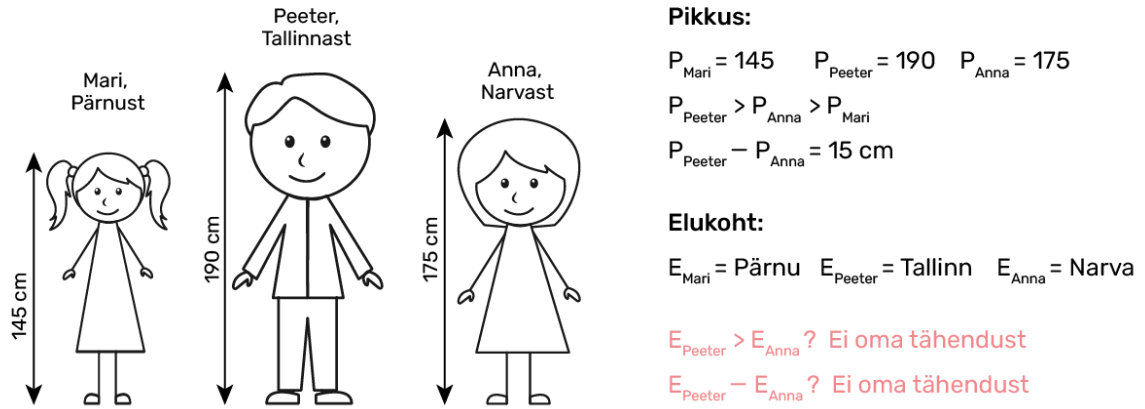
Vaatlusaluste objektide tunnuste väärtustest moodustubki **andmestik**, mida saab kujutada struktureeritud tabelina, kus iga rida vastab ühele objektile, meie näite puhul siis ühele inimesele. Iga veerg vastab mõõdetud tunnusele (tabel 3.1).

Nimi	Sugu	Elukoht	Pikkus	Kaal
Karl	M	Pärnu	125	25
Saša	M	Tartu	135	30
Mari	N	Pärnu	145	40
Peeter	M	Tallinn	190	90
Anna	N	Narva	175	65
Toomas	M	Pärnu	185	110

Tabel 3.1. Andmetabel joonisel 3.2 kujutatud isikute tunnustega. *Esimene rida tabelis pole andmestiku osa, vaid tunnuste nimetuste loend. Seega koosneb see andmestik kuuest vaatlusalusest ja viiest tunnusest (nimi, sugu, elukoht, pikkus ja kaal). Esimese vaatluse puhul on tunnuse „nimi" väärtuseks „Karl", tunnuse „sugu" väärtuseks „M" jne.*

Masinõppe- ja andmekaevealgoritmidel on vaja erinevate vaatluste tunnuste väärtusi võrrelda, et leida seoseid. Arvuliste tunnuste puhul on selge, kui sarnased on kaks väärtust või milline väärtus on suurem. Näiteks, on selge, et Toomase ja Peetri pikkus on sarnane, aga Karli pikkus on mõlemast märkimisväärselt väiksem (tabel 3.1). Samasuguseid sarnasuse hinnanguid saab teha ka kaalu kohta.

Seevastu kategooriliste tunnuste puhul on võimalik ainult öelda, kas nad on täpselt samad või mitte. Igasugused muud võrdlused pole võimalikud. Tehted „Tartu miinus Tallinn" või „Tartu miinus Pärnu" ei oma tähendust ja seega me ei saa öelda, millised kaks tunnuse „elukoht" väärtust on omavahel sarnasemad. Samuti pole võimalik neid väärtusi kuidagi suuruse järgi ritta seada, sest küsimus „Kas elukoht Tartu > Tallinn?" ei oma tähendust (joonis 3.3). Sama kehtib ka nimede puhul, pole ju võimalik öelda, kas Peeter on nimena sarnasem Toomase või Karliga või kas ta on neist suurem.[7]



Joonis 3.3. Tunnuste võrdlemine. *Pikkuste võrdlemine ja nendevaheliste erinevuste arvutamine omab tähendust, sest arve saab nii võrrelda. Sõnalisi ja muid kategoorilisi tunnuseid seevastu ei ole võimalik mingil viisil reastada ega nende erinevust mõõta – saame ainult konstateerida, kas väärtused on samad või mitte.*

[7] Linnana on Tallinn muidugi suurem, kuid tunnuse „elukoht" (mitte „kodulinna suurus") vaatenurgast on mõlema puhul tegu lihtsalt ühe kategooriaga mitmest. Samuti, sõnadena on Peeter ja Meeter omavahel sarnasemad kui Peeter ja Toomas, aga nimedena pole meil alust arvata, et sama kõlaga nimedega inimesed alati sarnasemalt käituksid või et muul moel oleks loogiline neid järjestada.

Kuna masinõppe ja andmekaeve puhul on tegu algoritmidega ehk kindlas järjekorras matemaatilisi operatsioone tegevate protsessidega, räägivad need algoritmid arvude keelt. Kui me tahame, et algoritm suudaks mõista mingite objektide tunnuseid, on vaja need arvudena kirja panna. Arvuliste väärtuste puhul on see triviaalne – panemegi kirja vastava arvu. Kategooriliste muutujate (nagu nimi või kodulinn) puhul aga on vaja meeles pidada, et neid ei saa võrrelda ega öelda, et mingid väärtusevariandid on omavahel sarnasemad. Samas saavad need olla identsed ehk täpselt samad. Selle jaoks on välja mõeldud vastavate andmete erilised esitusviisid, näiteks üks-mitmest-esitusviis (ingl k *one-hot representation*). Sageli viiakse andmed sellele kujule automaatselt masinõppe algoritmi sees. Andmete ettevalmistamisest mudelite kasutamise jaoks räägib pikemalt peatükk 4.

3.1.2 Andmestikud ja nende kvaliteet

Andmestik moodustub, kui paljude vaatlusaluste objektide (antud juhul inimeste) tunnused struktureeritud tabelisse kokku koguda. Igal real on ühe objekti (inimese) tunnused. Andmestiku üheks oluliseks omaduseks ongi ridade arv – mitme asja kohta me infot kogunud oleme. Teine oluline omadus on kogutud info hulk – mitut tunnust me iga asja kohta teame. Ehk siis tabeli veergude arv. Olulised on muidugi ka muud aspektid – näiteks kui täpselt me mingeid väärtusi mõõtnud oleme – aga need pole tabelile peale vaadates kohe ilmsed.

Mida suurem on andmestik näidete (ridade) arvult, seda parem – meil on rohkem infot, mille põhjal andmeteadusega midagi huvitavat avastada. Seevastu suurem veergude arv ei pruugi alati tähendada, et andmestik on parem. Näiteks kui lisame oma andmestikku palju tunnuseid, mis dubleerivad üksteist või on lihtsalt väheinformatiivsed, ei anna see meile midagi kasulikku juurde, vaid pigem koormab infoga. Seega peab andmeteadlane andmete ettevalmistamise faasis (vt ptk 4) ise otsustama, millised tunnused (veerud) tabelist välja visata, et ei peaks aega raiskama kasututest tunnustest info otsimisele. Ka inimesel on ju lihtsam õppida, kui talle räägitakse asjast ega koormata teda üle tähtsusetute detailidega.

Kui ennist mainisime, et rohkem ridu on alati hea, siis lihtsalt suurest hulgast näidetest reaalelulisteks rakendusteks alati ei piisa. Need näited peavad ka olema mitmekesised. Ideaalis võiks andmestik olla **esinduslik** (moodustada esindusliku valimi). Esinduslik andmestik sisaldab kõiki näiteid, mis päriselus ette tulla võivad, ja sisaldab neid samas proportsioonis, kui neid päriselus kohata võiks. Kui me võtame andmestikust juhusliku rea, on see võrdväärne juhusliku uue näite võtmisega reaalelust. Kui teeme järeldusi kallutatud andmetel, ei peegelda need tulemused lahenduse võimekust pärismaailmas rakendamisel. Näiteks, kui hindame ravimi efektiivsust ainult noorte katseisikute hulgas, ei peegelda see efektiivsust kogu rahvastiku hulgas. Samuti ei võimalda Tallinna poodide ostuandmed ennustada ostukäitumist Muhumaal.

Päris inimeste poolt reaalsest elust kogutud andmestike puhul on veel üks oluline kvaliteedinäitaja – puuduvate või vigaste väärtuste arv. Andmeid tabelisse sisestades võib juhtuda, et mingit tunnuse väärtust lihtsalt ei teata (nt kehakaalu unustati küsida

või inimene keeldus seda avaldamast) või on kogemata sisestatud vale arv (nt koma pandud valesse kohta, tabel 3.2).

Nimi	Sugu	Elukoht	Pikkus	Kaal
Karl	M	Pärnu	125	25
Saša	M	Tartu	135	
Mari	N	Pärnu	145	40
Peeter	M	Tallinn	190	90
Anna	N	Narva	17	65
Toomas	M	Pärnu	185	110

Tabel 3.2. Puuduvad ja vigased väärtused. *Meie algne andmestik, aga oleme mõnda väärtust muutnud.*

Ülesanne
Leia tabelis 3.2 toodud andmestikust puuduv tunnuse väärtus. Leia tõenäoliselt valesti märgitud tunnuse väärtus. (Vastuseid saad kontrollida siit.)

Kuna andmed võivad olla esitatud erieval kujul ja eri formaadis, räägitakse andmeteaduses **toorandmetest** ja **andmete puhastamisest** (vt ka ptk 4). Toorandmed (ingl k *raw data*) on need andmed, mida ei ole veel kuidagi töödeldud. Need on algallikast kogutud/saadud andmed täpselt sellisel kujul, nagu nad on. Näiteks, väljavõte andmebaasist, kogutud pildid ja videod, tekstid, veebisaidilt kraabitud andmed, makseterminali salvestatud andmed.

Andmete puhastamine (ingl k *data cleaning*) on protsess, mille käigus korrastatakse toorandmed nii, et neid saaks andmeteaduse meetodeid kasutades analüüsida. Korrastamine tähendab, et andmed viiakse struktureeritud kujule, valitakse vastavalt püstitatud eesmärgile sobivad tunnused, vajaduse korral moodustatakse uued tunnused, eemaldatakse või asendatakse puuduvad väärtused, tuvastatakse võimalikud valesti sisestatud väärtused jne.

Siinjuures on toorandmete mõiste suhteline. See tähendab, et ühe projekti raames puhastatud andmed võivad olla sisendiks mõnele teisele analüüsile, kus neid andmeid koheldakse uuesti kui toorandmeid, sest need ei ole veel planeeritud analüüsi jaoks sobivad.

3.2 Andmete kogumise ja korraldamise hea tava

Andmeanalüüsi protsess algab andmete kogumise ja mõistmisega. Andmed võivad pärineda erinevatest allikatest – kliendiküsimustikest, veebilehitseja ajaloost, sooritatud finantstehingutest jne. Peamine nõue andmetele, et neid saaks lihtsasti analüüsida, on see, et **andmed peavad olema arvutile loetavad ehk digiteeritud**.

Andmeteadus algab **andmete kogumisest**. Kui andmete kogumine ei ole hästi planeeritud, võib see rikkuda eduka analüüsi läbiviimist ja see omakorda takistab kasulike otsuste tegemist. Näiteks, kui olete loomas tootmisettevõtte hooldusplaani ja soovite selle tarbeks kasutada andmeid, et ennustada, kas mõni masin läheb katki või mitte, siis peate veenduma, et kogute iga masina kohta märgendatud andmeid. See tähendab, et iga masina kohta on andmetes kirjas, kas see on „katki" või „töökorras". Kui te ei kogu seda informatsiooni, või olete kogunud seda ainult vahetevahel ja osaliselt, siis ei saa luua mudelit, mis ennustab masinate katki minemist.

Andmete kogumisel on oluline teada, et **andmed peavad olema mitmekesised**, ideaaljuhul moodustama esindusliku valimi. Esinduslik andmestik sisaldab kõiki näiteid, mis päriselus ette võivad tulla, ja sisaldab neid samas proportsioonis, kui neid päriselus kohata võiks. Ehk siis, kui me võtame juhusliku rea andmestikust, siis on see võrdväärne juhusliku uue vaatluse võtmisega reaalelust. Kui hindame suutlikust kallutatud andmetel, ei peegelda tulemus seda, kui usaldusväärsed meie analüüsi tulemused tegelikult pärismaailmas rakendades oleksid. Näiteks, kui kogume andmeid ainult tudengite hulgas, siis nendel andmetel saadud tulemusi ei saa hiljem üldistada kogu elanikkonnale.

NB!
Andmete kogumise ja korraldamise hea tava mõistmine ja rakendamine on eduka andmeteadusprojekti alus.

Korrastatud ja struktureeritud andmeteta on andmeteaduslikke meetodeid keeruline rakendada. On näidatud, et **andmeteadlased kulutavad** kuni **80%** oma **tööajast andmete omandamisele (kätte saamisele teiste osapoolte käest, kogumisele), puhastamisele ja analüüsiks ettevalmistamisele** ning ainult 20% ajast tegelikule modelleerimisele. Seda ajakulu on võimalik vähendada, kui kasutada häid andmete kogumise strateegiaid.

Kuna paljud inimesed on oma elu jooksul sisestanud andmeid Exceli tabelisse, toome siinkohal mõned praktilised näited **andmetabelite korraldamise ja kasutamise heast tavast**, mis õpetab andmeid struktureerima nii, et hilisemas programmaatilises analüüsis oleks vähem probleeme.

Selles alapeatükis õpime:
- kuidas organiseerida andmetabeleid, et neid oleks pärast lihtne kasutada;
- kuidas vältida andmete sisestamise levinumaid vigu;
- põhilisi andmekvaliteedi kontrollimise nippe.

Selles õpikus me ei õpeta andmete analüüsi Excelis. Üldiselt kehtib teadmine, et ei ole hea mõte mahukamat andmeanalüüsi läbi viia Excelis või muus sarnases tabelarvutusprogrammis. Selleks on kolm põhilist põhjust:

- andmeanalüüs nõuab tabelarvutuse programmides tavaliselt palju käsitööd – kui soovite mõnda parameetrit muuta või teha sama analüüsi uutel andmetel, tuleb kogu protsess uuesti käsitsi läbi viia. On küll olemas makrod (toimingukogumid, mis kirjeldavad sisendandmete töötlemiseks tehtud samme), mis aitavad seda osa natukene leevendada kuid need ei ole piisav lahendus;
- Excelis tehtud analüüsi ja tehtud teisenduste samme on raske või isegi võimatu jälgida ning seetõttu on ka raske tehtud statistilisi analüüse ja graafikuid hiljem reprodutseerida;
- kui andmemaht läheb väga suureks, muutub failide avamine ja andmete analüüs väga aeglaseks või lausa võimatuks.

3.2.1 Kuidas organiseerida andmetabeleid?

Levinuim viga andmetabelite loomisel Excelis on nende kasutamine märkmikuna. See tähendab, et info edastamisel tuginetakse kontekstile, ääremärkustele ning andmete ja väljade ruumilisele paigutusele. Inimesed suudavad küllaltki hästi sellisel kujul andmetest aru saada, aga arvutid ei näe infot samamoodi nagu inimesed. Seetõttu tuleb andmeid hoiustada sellisel kujul, millest arvutid aru saavad, et saaksime kasutada kõiki suurepäraseid andmeanalüüsi võimalusi, mida arvutid suudavad pakkuda.

Tehtud muudatuste jälgimine

Töödeldes andmeid arvutustabelites, on väga lihtne jõuda tabelini, mis näeb välja väga erinev sellest tabelist, millega tööd alustasite. Selleks, et tehtud samme oleks võimalik hiljem korrata, on hea mõte **hoida algandmete fail muutumatuna** ning hoopis luua sellest failist koopia, milles andmeid puhastada ja töödelda.

Pange kirja kõik tehtud sammud, mis viisid algandmetest korrastatud andmeteni. Soovitame kõik see kirja panna **lihtsasse tekstifaili** (nt Wordi dokumenti), mitte andmefaili endasse, ja hoiustada seda koos andmefailidega. Siis on hiljem lihtsam analüüsi alguspunkti tagasi minna ja kõiki tehtud samme korrata.

Andmete struktureerimine

Andmetabelite struktureerimise põhireegel on, et andmed tuleb hoida puhtad ja korras.

Soovitused andmete halduseks:

- hoidke kõik mõõdetud muutujad ehk tunnused (kaal, temperatuur jne) veergudes;
- vaatlusi hoidke ridades (iga rida ühe inimese kohta);
- ärge kombineerige mitut infokildu ühte lahtrisse. Selle eristamiseks on kasulik mõelda, kas sellisel kujul sisestatud lahtriväärtused on ainus viis, kuidas te soovite neid väärtusi kasutada või sortida;
- ärge muutke andmete algfaili, töötage koopiatega;
- enne andmete jagamist salvestage need tekstiformaati, näiteks CSV-faili (komaga eraldatud fail), sest seda on programmaatiliselt lihtsam töödelda.

NB!
Rusikareegel andmete struktureerimiseks:

- veerud = muutujad;
- read = vaatlused;
- lahtrid = väärtused.

Näiteks, kui me hoiaksime klientide ostuandmeid sellisel kujul nagu on näidatud tabelis 3.3 (toote kogus ja ühik ühes veerus), siis me ei saaks ühiku hinna abil arvutada toote eest makstud kogusummat, sest veerus „Kogus-Ühik“ pole arvulised väärtused.

Klient ID	Kuupäev	Toode	Kogus-ühik	Ühiku hind
Klient 1	01/06/2020	banaan	0.8 kg	1.09
Klient 1	01/06/2020	leib	1 tk	0.77
Klient 2	23/09/2020	šampoon	1 tk	3.45
Klient 3	24/09/2020	grill-liha	1.5 kg	8.18

Tabel 3.3. Näide halvasti struktureeritud andmetabelist. *Veerg „Kogus-Ühik“ sisaldab korraga infot kahe asja kohta: kui suures koguses toodet osteti ja mis on selle toote mõõtühik. See takistab nende mõõtmiste kasutamist edasises analüüsis. Näiteks ei saa selle tulba abil arvutada toote kogumaksumust ega teha kokkuvõtvat ülevaadet kasutatavatest ühikutest.*

3.2.2 Levinud vead andmetabelite korraldamisel

Siin alapeatükis toome näiteid **levinud probleemidest**, mis tekivad andmetabelite haldamisel, ning räägime sellest, mis on nende mõju ja kuidas neid vältida. Need on põhilised vead, millele peaksite tähelepanu pöörama, kui panete kokku oma andmestikku või kasutate kellegi teise koostatud andmestikke (nt internetist leitud või koostööpartnerite andmeid). Eesmärgiks on, et kui te olete teadlikud vigadest ja nende võimalikust **negatiivsest mõjust** andmeanalüüsile ja tulemuste tõlgendamisele, siis see võiks motiveerida teid ja teie kolleege neid vältima. Andmete kogumise faasis tehtud väikestel muudatustel on suur mõju andmete puhastamise ja analüüsi tõhususele ning usaldusväärsusele.

Mitu tabelit ühel lehel

Üks levinud strateegia on mitme andmetabeli paigutamine ühele lehele. Kui inimene saab peale vaadates aru, mida see leht sisaldab, siis arvuti ajab see vaatepilt segadusse, seega ärge seda tehke! Kui paigutate ühele lehele mitu erinevat tabelit, siis tekitate sellega arvuti jaoks valesid seoseid, sest arvuti näeb andmeid nii, nagu kirjeldavad struktureerimise reeglid – iga rida on üks vaatlus. Võimalik ka, et nendes tabelites on sama veeru nimi kasutusel mitmes kohas, mis raskendab andmete puhastamist (milline sama nimega veerg on õige? Valida saame ainult ühe). Joonisel 3.4 toodud kuvatõmmis demonstreerib seda probleemi hästi.

29. mai				Kuupäev: 29. ma		12. juuni				Kuupäev: 12. juuni		19. juuni				Kuupäev: 19. juuni	
Kauplus	**Toote ID**	**Laos**	**Kaal**	**keskmine**	**sd**	**Kauplus**	**Toote ID**	**Laos**	**Kaal**	**keskmine**	**sd**	**Kauplus**	**Toote ID**	**Laos**	**Kaal**	**keskmine**	**sd**
Tartu	2	JAH	37	37,18	7,49	Tartu	7	EI	43	33,14	13,55	Tartu	2	JAH	10	33,86	16,19
Tartu	7	EI	33			Tartu	3	EI	20			Tartu	7	EI	65		
Tartu	3	EI				Tartu	1	EI	11			Tartu	3	EI	20		
Tartu	1	EI				Tartu	3	EI	50			Tartu	1	EI	50		
Tartu	3	EI	40			Tartu	7	EI	48			Tartu	3	EI	50		
Tartu	7	EI	48			Tartu	4	JAH	29			Tartu	7	EI	48		
Tartu	4	JAH	46			Tartu	4	JAH	46			Tartu	4	JAH	49		
Tartu	7	EI	36			Tartu	7	EI	36			Tartu	7	EI	36		
Tartu	7	JAH	29			Tartu	7	JAH	29			Tartu	7	JAH	29		
Tartu	8	JAH	22			Tartu	8	JAH	22			Tartu	8	JAH	33		
Tartu	7	JAH	42			Tartu	7	JAH	32			Tartu	7	JAH	32		
Tartu	4	JAH	41			Tartu	4	JAH	41			Tartu	4	JAH	41		
Tartu	6	JAH	37			Tartu	6	JAH	47			Tartu	6	JAH	27		

***Joonis 3.4. Näide halvasti struktureeritud andmefailist**, kus mitu tabelit on ühel lehel.*

Antud juhul eeldab arvuti kuuendat rida (joonisel 3.4 märgitud punasega) vaadates, et kõik veerud A–X viitavad ühele ja samale vaatlusele. Tegelikult aga katab see rida kolme eraldi mõõtmist sama vaatluse kohta (st ühte objekti on mõõdetud kolmel kuupäeval – 29. mail, 12. juunil ja 19. juunil). Lisaks sisaldab see rida arvutatud väärtusi (keskmine ja standardhälve (sd)), mis omakorda rikub tabeli loogikat. Ka teised read selles tabelis ei vasta standarditele ja tekitavad sarnaseid probleeme.

Mitme vahelehe kasutamine

Aga kui panna need andmed eraldi vahelehtedele, kas siis on need tabelid hästi vormistatud? Vastus on JAH ja EI. Näiteks, kui tekitate lihtsalt eraldi vahelehed iga mõõtmise päeva jaoks, siis arvuti ei näe nende tabelite vahelisi seoseid. See tähendab, et andmete ühenduse tagamiseks tuleb hiljem kasutada spetsiifilisi funktsioone või skripte.

See ei ole hea tava kahel põhjusel:

1) on suurem tõenäosus, et andmetabelite vahel tekib ebakõlasid, kui te iga kord, kui te midagi mõõdate, tekitate uuele vahelehele uue tabeli (nt veergude nimed või järjekord võivad erinevatel vahelehtedel tahtmatult muutuda);
2) isegi kui õnnestub kõiki ebakõlasid vältida, tekitate sellega ühe lisasammu, mida tuleb kindlasti enne andmete analüüsi teha – nende tabelite ühendamise ühtseks andmetabeliks. Arvutile tuleb detailselt ette öelda, kuidas neid vahelehti kombineerida, ja kui nendes olevad tabelid ei ole järjekindlad, tuleb seda võib-olla isegi käsitsi teha.

Järgmine kord, kui sisestate andmeid ja soovite luua uue vahelehe või tabeli, siis mõelge, kas saaksite äkki seda vältida, lisades näiteks esialgsele tabelile hoopis uusi veerge või ridu. Nii tehes võib juhtuda, et andmetabel venib väga laiaks või lisatud ridade pärast pikaks, mis raskendab uute andmete sisestamist, sest te ei näe enam esimestes veergudes olevaid olulisi indentifikaatoreid ehk veeru nimesid. Kindlasti ei tohiks siis vahepeale uuesti sama sisuga (nt kliendi nimega) veerge või veeru pealkirjadega rida lisada. Selle asemel saab Excelis esimesi ridu või veerge lukustada nii, et need on nähtavad ka siis, kui olete tabeli alumises otsas või paremas servas.

Nullide sisestamata jätmine

Võib juhtuda, et millegi mõõtmisel on selle väärtus tavaliselt null. Näiteks mõõtes, kas klient on oma maksevõime kaotanud (enamasti on vastus ei ehk 0). Miks on vaja sisestada sellesse veergu väärtusi „0“, kui selle veeru väärtused on enamasti kõik nullid? Kas ei võiks lahtrit lihtsalt tühjaks jätta ja täita ainult neil juhtudel, kui vastus on „jah“?

Seda ei tohi teha. Andmete kogumise seisukohast on väärtuse „0“ ja tühja lahtri vahel erinevus. Arvuti jaoks on „0“ osa informatsioonist, tühi lahter aga tähendab, et seda väärtust ei mõõdetud, ja arvuti tõlgendab seda kui puuduvat väärtust. See võib põhjustada probleeme edasistes arvutustes ja analüüsides. Nii võib näiteks juhtuda, et meie „on maksevõimetu“ veeru keskmine on 1, sest kõik 0-väärtused olid sisestamata.

Puuduvate väärtuste tähistamine

Puuduvate väärtuste tähistamiseks kasutatakse vahel „–999“ või muid arvulisi väärtusi (sealhulgas väärtust „0“). On erinevaid põhjuseid, miks puuduvaid väärtusi tähistatakse eri andmekogudes erinevalt. Mõnikord on põhjus juba mõõtmisseadmetes, mis tähistavad puuduvaid väärtusi vahel kummaliselt. Sellisel juhul ei saa te palju ära teha, kuid andmete puhastamisel enne analüüsi tuleb neid kindlasti arvesse võtta ja vastavalt käsitleda. Mõnikord kasutatakse erinevaid tähiseid, et edastada erinevaid põhjuseid, miks andmed puuduvad. See on küll oluline lisateave, kuid tegelikult kasutatakse seda tehes mitut infokildu ühes lahtris (mäletate, see on hea tava vastane). Parem oleks kasutada ühtset tähist kõigi puuduvate väärtuste jaoks, aga luua uus veerg nimega „miks_puudub“, kuhu saab vastava põhjuse kirja panna.

Ükskõik, mis on selle põhjuseks, aga kui tundmatud või puuduvad väärtused sisestatakse kui 999, –999 või 0, põhjustab see tihti edasises analüüsis probleeme. Paljud statistikaprogrammid ei mõista, et need väärtused on mõeldud puuduvate väärtuste esitamiseks. Nende väärtuste tõlgendamine sõltub kasutatavast tarkvarast, aga pigem vaadeldakse neid kui reaalselt mõõdetud arvulisi väärtusi. Oluline on kasutada selgelt määratletud ja järjepidevat indikaatorit. Tühjaks jäetud lahtrid (enamiku rakenduste jaoks) ja „NA“ (R-i jaoks) on hea valik. Igal juhul on oluline teada ja üles märkida, kuidas puuduvad väärtused on andmetes tähistatud, et sellega teaks arvestada ka analüüsi teostav inimene.

Stiilivormingute kasutamine info edastamiseks

Üks põhilisi asju, mida tuleb meeles pidada Excelis andmete töötlemisel, on see, et lahtrite, ridade või veergude esiletõstmine, kasutades selleks värve, kirja suurust või muid vorminguid, ei ole hiljem arvuti jaoks loetav informatsioon. Hea viis, kuidas kontrollida, millised näevad andmed välja erinevatele andmeteaduses kasutatavatele tööriistadele ja programmeerimiskeeltele, on salvestada loodud Exceli fail tekstifailina (nt *Save as CSV*) ja vaadata seda mõnes tekstitöötlusprogrammis nagu NotePad++. Siis on selgelt näha, et info värvitud lahtrite kohta ei ole sinna faili mitte kuidagi salvestatud.

Selle asemel, et ära värvida mingi eriomadusega lahtrid, on parem lahendus tekitada uus veerg, mille jah/ei-väärtused tähistavad, kas sellel vaatlusel esineb see eriomadus või mitte (joonis 3.5).

Ära nii tee

Kuupäev	Kauplus	Laos	Kaal
09.01.2019	Tartu	EI	40
09.01.2019	Tartu	JAH	36
09.01.2019	Tallinn	JAH	135
20.01.2019	Tartu	JAH	39
20.01.2019	Tartu	EI	43
20.01.2019	Tallinn	JAH	144
13.03.2019	Tartu	JAH	51
13.03.2019	Tartu	JAH	44
13.03.2019	Tallinn	JAH	146

kaal poolnud kalibreeritud

Tee hopis nii

Kuupäev	Kauplus	Laos	Kaal	Kalibreeritud
09.01.2019	Tartu	EI	40	JAH
09.01.2019	Tartu	JAH	36	JAH
09.01.2019	Tallinn	JAH	135	JAH
20.01.2019	Tartu	JAH	39	JAH
20.01.2019	Tartu	EI	43	EI
20.01.2019	Tallinn	JAH	144	JAH
13.03.2019	Tartu	JAH	51	JAH
13.03.2019	Tartu	JAH	44	JAH
13.03.2019	Tallinn	JAH	146	EI

Joonis 3.5. Näide, kuidas vormindada andmetabelis eriomadused.

Kommentaaride lisamine

Sarnaselt stiilivormingutega ei kajastu standardses andmeformaadis ka lahtritele lisatud kommentaarid. Saate seda samamoodi kontrollida, nagu kirjeldasime eelmises punktis. Selle asemel on hea luua uus lahter nimega „kommentaarid“ ja lisada kõik märkused sinna (sarnaselt joonisel 3.5 näidatud lahendusega).

Mitme info lisamine lahtrisse

Nagu juba korduvalt mainitud, ei tohiks üks veerg sisaldada rohkem kui ühte infokildu, vaid looge iga vajaliku teabe jaoks eraldi tulp. Näiteks nagu tabelis 1 on eraldi toodud koguse ja ühiku veerg.

Probleemsete väljanimede kasutamine

Tavaliselt pannakse tabeli formaadis igale veerule pealkiri ehk nimi, mis peaks kirjeldama seal sisalduvat infot või mõõtmist. Selleks kasutage hästi kirjeldavaid nimesid, kuid ärge kasutage tühikuid ega mistahes erimärke. Tühikud veerunimedes võivad põhjustada palju segadust, sest on programme, mis tõlgendavad tühikuid veergude eraldajatena, ja seega võib teie ühest veerust kogemata saada hoopis kaks eraldi veergu. Mõnele programmile ei meeldi ka numbriga algavad väljanimed. Seega on targem selliseid nimesid vältida. Tühikule on hea alternatiiv näiteks alakriips (_) või kõigi sõnade suure algustähega kirjutamine, näiteks „MaxTemp“.

Lisaks pidage meeles, et täna kirjutatud mõistlikud lühendid ei pruugi kuue kuu pärast enam nii ilmsed olla, kuid ärge pingutage üle ka liiga pikkade nimedega. Hea oleks eraldi dokumenti üles kirjutada veergude pikemad kirjeldused. Ühikute lisamine veerunimesse võib aidata segadust vältida, kuid soovitame seda teha kujul „pikkus_cm“.

Erisümbolite kasutamine

Kui andmetabelis on lahtrid, mis sisaldavad pikemaid tekste koos reavahetuste, sidekriipsude ja muude erisümbolitega (hüüumärk, küsimärk, protsentmärk jne), võib

nende andmete programmeerimiskeskkonda või andmebaasi eksportimisel ilmneda ootamatuid asju, näiteks on ühtäkki andmeread pooleks lõigatud või kuvatakse veateateid. Parim tava on vältida selliste märkide lisamist ja käsitleda tekstilahtrit nii, nagu see oleks lihtne veebivorm, mis võib sisaldada ainult teksti ja tühikuid.

Metaandmete sisestamine andmetabelisse

Võite juba aimata, et veergude tähendusi, ühikuid, lühendeid, erandeid jms selgitava legendi lisamine andmetabeli üla- või alaossa ei ole hea mõte. Põhjuseks see, et legend on justkui eraldiseisev tabel ja punktis „Mitu tabelit ühel lehel“ näitasime, et sellisel kujul andmeid on hiljem väga raske andmeanalüüsiks töödelda. Ometigi on taolised metaandmed väga vajalikud, sest sisaldavad andmetest arusaamise andmeid. Praegu võite te detailselt oma andmetest aru saada ja teate, mida iga veeru nime lühend tähendab, aga on vähe tõenäoline, et te poole aasta pärast kõike seda detailselt mäletate. Vahel peab keegi teine kasutama teie andmeid ja selleks peaks ta samamoodi mõistma, mida nendes kajastatud on.

Parim lähenemine on salvestada kõik andmete kirjeldused ja tehtud sammud eraldi tekstifaili, kindlasti mitte andmefaili endasse. Kuna metaandmete failid on vaba teksti vormis, võimaldavad need dokumenteerida kommentaare ja ühikuid, edastada teavet puuduvate väärtuste kodeerimise kohta jne. Metaandmed kirjeldavad ka seda, kuidas andmestikus olevad erinevad failid üksteisega seotud on ja mis vormingus nad on. Tüüpiliselt on selle faili nimeks README.txt.

Kokkuvõte:

- Vältige mitme tabeli kasutamist ühel vahelehel.
- Võimaluse korral vältige mitme vahelehe kasutamist.
- Sisestage ka nullväärtused.
- Tähistage puuduvad väärtused sobivalt.
- Ärge kasutage teabe edastamiseks ega tabeli ilusaks muutmiseks stiilivorminguid.
- Kommentaarid lisage eraldi veergu.
- Lisage igasse lahtrisse ainult üks väärtus.
- Vältige veerunimedes tühikuid ja erimärke.
- Vältige andmetes erimärke.
- Metaandmed salvestage eraldi tekstifaili.

3.2.3 Andmekvaliteedi kontrollimise nippe Excelis

Kui andmed on hästi struktureeritud ning loodud eelmises peatükis toodud reeglite järgi, siis on Excelis võimalik kasutada mitmesuguseid nippe, millega kontrollida andmekvaliteeti ja teha kindlaks, et andmetes ei esine vigu. Kvaliteedikontrolliks kasutage alati algandmetest tehtud koopia faili.

Sortimine

Kahtlased väärtused kipuvad sortimisel paiknema veeru algusesse või lõppu. Näiteks, kui veerg peaks sisaldama arvulisi väärtusi, siis sortides on tekstilised ja puuduvad väärtused reastatud veeru lõppu. Niimoodi igat veergu ükshaaval sortides ning siis vastava veeru algust ja lõppu kontrollides on juba võimalik tuvastada, ega andmete sisestamisel ole vigu tehtud.

Tingimuslik vorming

Tingimuslik vorming võimaldab tulbas olevaid väärtusi värvida mingi kriteeriumi alusel. Värvitud tulbast on lihtsam leida näiteks esilekerkivaid arvulisi väärtusi ehk erindeid (ehk anomaalseid ehk võõrväärtusi), mis võivad olla vigaselt sisestatud. Näiteks võib veergu üldreeglina olla sisestatud pikkus meetrites, aga ühe inimese puhul on seda tehtud sentimeetrites. Siis paistab see suur arv teiste seast erindina välja. See viga tuleb välja ka pikkuse veergu sortides.

Üldiselt on andmete puhastamine ja kontrollimine andmeteadlaste töö ja nemad teevad seda, kasutades mõnda programmeerimiskeelt või muud tarkvara. Eeltoodud kirjeldus võimaldab teil ette kujutada, mida andmete ettevalmistamine endast kujutab ja millist loogikat selles järgitakse.

Ülesanne
Järgides kirjeldatud andmete hoiustamise parimaid tavasid, tuvastage ja parandage ühes Exceli andmefailis tehtud vead.

1. **Laadige alla see andmefail** (kasutage selleks valikut: Laadi alla → Microsoft Excel (.xlsx)) või tehke sellest oma Google Drive'i koopia: algandmete link
2. Avage see fail Excelis (või kui Excelit ei ole, siis nt LibreOffice Calc programmis või Google Drive arvutustabelis).
3. Need fiktiivsed andmed kajastavad ühe kaubandusettevõtte eri linnades asuvate kaupluste laoseisu, kus hoiustatakse sama toodet erinevates kogustes valmis kaalutuna. Selles ettevõttes töötas kaks inimest, kes neid andmeid kogusid ja sisestasid. Üks neist tegi seda 2018. ja teine 2019. aastal. Vastavalt sellele on näha, et failis on kaks vahelehte: 2018 ja 2019. Nüüd olete teie need tabelid saanud ja tahate neid aastaid koos analüüsida, et näiteks leida iga toote kogukaalu kõigi kaupluste ladudes kokku.
4. **Tuvastage, mis võib olla selle andmefaili koostamisel valesti tehtud.** Lisaks mõelge, mis samme tuleks teha, et saaksite need kaks andmetabelit puhastada (korda teha) ja lõpuks üheks andmetabeliks kokku panna. Võite proovida neid samme ka ise läbi teha. Link ühe võimaliku lahenduse juurde on toodud allpool.

Et näha, kuidas neid andmeid näevad arvutiprogrammid, võib kasutada „Save As"-funktsiooni ja salvestada selle faili CSV-failina. Märkate, et tekstifailiks saab

salvestada ainult ühe vahelehe kaupa, aga eraldi failidena ei ole mõistlik neid analüüsima hakata, sest nad sisaldavad sama infot, mis tuleks koondada. Seega on parem need tabelid kõik üheks andmetabeliks kokku panna. Aga kui salvestate ainult ühe esialgse vahelehe CSV-failiks ning avate selle tekstitöötlusprogrammis nagu NotePad++ või Word, siis täpselt sellisena andmetöötluse tööriistad ja programmeerimiskeeled neid andmeid näevadki. Need loetakse rida rea haaval sisse, eeldades struktureeritud tabeli formaati, kus esimene rida sisaldab veergude pealkirju. Märkate ka, et lisatud kommentaare ega värvivorminguid CSV-failis näha ei ole.

Üks kokku pandud tabeli variant on saadaval siin: lahenduse link.
Teie lahendus võib sellest natukene erineda, aga üldine struktuur peaks olema sarnane.

3.2.4 Andmete hoiustamine

Heade andmekogumistavade kõrval tuleb tänapäeval järjest enam tähelepanu pöörata ka andmete haldamise ja säilitamise reeglitele. Andmehalduse reeglid (tagavarakoopiate loomine, versioneerimine, formaadid ja standardid) on hea kindlaks määrata kohe projekti alguses, luues vastavasisulise dokumentatsiooni. Tundlike andmete puhul kehtib minimaalsuse printsiip: tohib koguda ainult nii palju andmeid, kui on konkreetse analüüsi või projekti jaoks vajalik. Siit järeldub ka minimaalsuse printsiip andmete talletamisel: iga andmestik peab olema seotud kindlate säilitustähtaegadega, et vältida andmete ülemäärast kogumist ja pikaajalist kasutamist väljaspool algset eesmärki. Kuigi näiteks sensorite mõõtmistulemused ei ole tundlikud andmed ja arvatavasti ei reeda ka ärisaladusi, ei ole ka neid mõtet igavesti säilitada, sest suurte andmekoguste talletamine pole ju tasuta. Andmed aeguvad ja see teeb liiga vanade andmete talletamise mõttetuks.

Andmete anonüümimine on hoiustamisel üha olulisem praktika, eriti tundlike isikuandmete puhul. Anonüümimine tähendab teabest kõikide jälgede kaotamist, mis võiksid viia tuvastatavate isikuteni. Kui pseudonüümimine ja krüptimine on tagasipööratav umbisikustamine, siis anonüümimine tähendab tagasipööramatut ehk lõplikku umbisikustamist. Anonüümimise kohta võite pikemalt lugeda lisamaterjalidest.

3.3 Andmete kirjeldamine ja visualiseerimine

Pärast andmete kogumist erinevatest andmeallikatest ning struktureeritud kujule viimist on andmeanalüüsi protsessi järgmine oluline samm andmete esmane uurimine ja visualiseerimine. Seda nimetatakse **kirjeldavaks analüüsiks**. Kirjeldav analüüs võimaldab andmeteadlastel ja analüütikutel saada kiire ja põhjaliku ülevaate andmete struktuurist ja omadustest. On oluline teada, kuidas tunnused jaotunud on, mis on erinevate väärtuste keskmised ja kui varieeruvad need väärtused on, milliseid mustreid või trende andmetes esineb – kõike seda käsitleb **kirjeldav statistika**. Selle kõige

paremaks tajumiseks kasutatakse tihti ka abistavaid jooniseid ja graafikuid. Kirjeldav statistika ja tunnuste **visualiseerimine** ongi kirjeldava analüüsi põhilised meetodid. Peale kirjeldamise aitavad need meetodid andmetest tuvastada võimalikke vigaseid väärtusi. Kirjeldav statistika aitab esile tuua olulised arvulised kokkuvõtted ning visualiseerimine muudab need mustrid ja trendid hõlpsamini mõistetavaks ja tõlgendatavaks.

Lisaks Excelile, mis on lihtsate ja väiksemamahuliste andmete visualiseerimisel populaarne valik, loetleme siin veel mõned vahendid, mis sobivad keerulisemate andmekogumite visualiseerimiseks ja analüüsiks.

- **Tableau** on kasutajasõbralik tööriist, mis lubab visualiseerida keerukaid andmeid interaktiivsete graafikutega. Tableau sobib hästi äriliste andmete analüüsiks.
- **Power BI** on Microsofti loodud andmete visualiseerimise platvorm, mis on samuti mõeldud interaktiivsete aruannete loomiseks ning sobib äri- ja andmeanalüütikutele.
- **Matplotlib** ja **Seaborn** on **Python**i visualiseerimisteegid. Neid tööriistu kasutatakse sageli akadeemilistes ja tehnilistes projektides, kus on vaja suuremat kohandatavust.
- **ggplot2** on **R** programmeerimiskeele teek, mis pakub keerukate visualiseeringute loomiseks kergesti kohandatavaid vahendeid. Seda kasutatakse laialdaselt teaduslikes ja statistilistes uuringutes.

Selles alapeatükis

- tutvustame kirjeldavat statistikat ja erinevaid graafikutüüpe, mis aitavad andmete mõistmise etapis teha informeeritud otsuseid ja avastada olulisi seoseid;
- õpime, kuidas andmeid kirjeldada ja milliseid statistilisi vahendeid selleks kasutada;
- õpime, kuidas valida visualiseerimismeetodeid ja tõlgendada graafikuid.

3.3.1 Kirjeldav statistika

Kirjeldav statistika on statistika haru, mis tegeleb andmete kokkuvõtmise ja esitamisega. Arvtunnuste korral on võimalik leida mitmesuguseid kokkuvõtvaid statistikuid, mis kirjeldavad tunnuste väärtuste omadusi. Tõenäoliselt kõige sagedamini kasutatav kokkuvõttev näitaja arvtunnuste korral on **aritmeetiline keskmine** ehk **keskväärtus**. Selle leidmiseks liidetakse kokku kõik ühe tunnuse väärtused ja jagatakse saadud summa vaatlusaluste objektide arvuga. Tulemuseks on näitaja, mida võib käsitleda kui vaatlusaluse tunnuse tüüpilist väärtust. Näiteks, kui ühe ettevõtte viie kuu käibed on vastavalt 550 eurot, 600 eurot, 700 eurot, 850 eurot ja 900 eurot, siis on keskmine käive (550 + 600 + 700 + 850 + 900) / 5 = 720 eurot. Pange tähele, et kuna keskväärtus on leitud kõigi kuude peale kokku, võib selle väärtus olla ka selline, mida tegelikult üheski kuus täpselt olnud pole.

Tähelepanelik tuleb keskväärtuse tõlgendamisel olla juhul, kui tunnuse väärtuste hulgas esineb **erindeid** ehk tavapärasest väga palju erinevaid väärtusi (vt alapeatükk 3.2). Sellisel juhul on keskväärtus kallutatud erindi suunas. Kui näiteks eelmises näites oleks

viimase kuu käive olnud 900 euro asemel 9000 eurot, oleks keskmine käive olnud 2340 eurot. See summa ei iseloomustaks ettevõtte igakuist käivet kuigi hästi. Samuti tuleb tähelepanelik olla, kui väärtuste jaotus ei ole sümmeetriline (vt joonis 3.7). Sellisel juhul võiks kaaluda keskmise asemel **mediaani** kasutamist. Mediaan on kasvavalt järjestatud väärtuste keskmise liikme väärtus ehk mediaanist suuremaid ja väiksemaid väärtusi on tunnuse väärtuste reas ühepalju. Toodud näite mediaankäive on 700 eurot, mis kirjeldab tavapärast igakuist käivet paremini kui 2340 eurot. Mediaani saab leida nii arvuliste kui ka järjestustunnuste puhul.

Statistik, mis sobib nii arvuliste, järjestus- kui nominaaltunnuste puhul, on **mood**. Moodi kasutatakse siis, kui soovitakse tunnust iseloomustada tema kõige sagedamini esineva väärtuse alusel. Eriti kasulik on mood kategooriliste tunnuste korral, sest nende puhul ei saa keskmist ega mediaani väärtust leida. Näiteks, oletame, et meil on andmed, mis peegeldavad ettevõtte klientide ostukogemuse hinnanguid. Kliendid andsid tagasisidet, kas ostukogemus oli väga hea, rahuldav või halb. Soovime arvutada iga kategooria moodi, et mõista, millist hinnangut anti kõige sagedamini. Oletame, et saime järgmised küsitluse tulemused: väga hea, rahuldav, väga hea, halb, rahuldav, väga hea, rahuldav, halb, väga hea, rahuldav, väga hea, väga hea, väga hea. „Väga hea“ esineb 7, „rahuldav“ 4 ja „halb“ 2 korda. Tulemus „väga hea“ esineb kõige sagedamini, see tähendab, et see on mood. Moodi leidmine aitab ettevõttel mõista, millist ostukogemuse hinnangut on kliendid andnud kõige sagedamini.

Hajuvuse karakteristikud iseloomustavad tunnuse väärtuste hajuvust (st kas väärtused erinevad üksteisest vähe või palju). Hajuvust keskväärtuse ümber kirjeldab **dispersioon**. Dispersiooni saab arvutada, kui leida kõigi tunnuse väärtuste erinevus aritmeetilisest keskmisest ja arvutada nende erinevuste ruutude keskmine. Näiteks, kui kuu käibed on 550 eurot, 600 eurot, 700 eurot, 850 eurot ja 900 eurot, siis dispersioon on

$$[(600 - 720)^2 + (600 - 720)^2 + (700 - 720)^2 + (850 - 720)^2 + (900 - 720)^2] / 5 = 15\,700.$$

Standardhälve on ruutjuur dispersioonist ja tunnuse standardhälbe ühikuks on sellesama tunnuse ühik (meie näites euro). **Seega näitab standardhälve tüüpilist erinevust tunnuse keskväärtusest**. Antud näite puhul on standardhälve $\sqrt{15\,700} = 125,20$ eurot. Kui standardhälve on suurem, siis võib arvata, et väärtused on enamasti aritmeetilisest keskmisest kaugel. Kui standardhälve on väiksem, on väärtused aritmeetilise keskmise lähedal. Hajuvust kirjeldavad veel ka arvtunnuse **minimaalne** (ehk vähim) ja **maksimaalne** (ehk suurim) **väärtus** ning nende väärtuste vahe, samuti suhtelised mõõdikud nagu **protsentiilid** (või ka kvartiilid), mis jaotavad tunnuse väärtuste jada teatud arvuks võrdseteks osadeks. Mediaan on 50. protsentiil ehk 2. kvartiil ja jagab väärtused kaheks võrdseks osaks. Sageli kasutatakse väärtuste jaotuse kirjeldamiseks ka **25. protsentiili** (ehk 1. kvartiili) ja **75. protsentiili** (ehk 3. kvartiili). 25. protsentiil tähistab tunnuse väärtust, millest väiksemaid või millega võrdseid väärtusi on 25%, ja 75. protsentiil tähistab väärtust, millest suuremaid või millega võrdseid tunnuseid on 25%. Analoogselt saab leida ka teisi protsentiile.

Veel üks andmete kirjeldamise võimalus on mitmesuguste kokkuvõtvate **risttabelite loomine**, kus on kokku loetud erinevate väärtuste esinemissagedused või kogusummad. Kui andmetabelid võivad sisaldada tuhandeid ridu, mille järgi on raske midagi kokkuvõtvat öelda, siis risttabelid võimaldavad kuvada koondandmeid mingite konkreetsete tunnuste kaupa. Näiteks on risttabel see, kui kuu ostusündmuste tabelis (tuhanded ja tuhanded andmeread) koondada kõik andmeread toodete kaupa (näiteks jahu, riis, suhkur) ja leida iga toote kohta summaarne müügitulu ja kogus üle kõikide ostude (tabel 3.4).

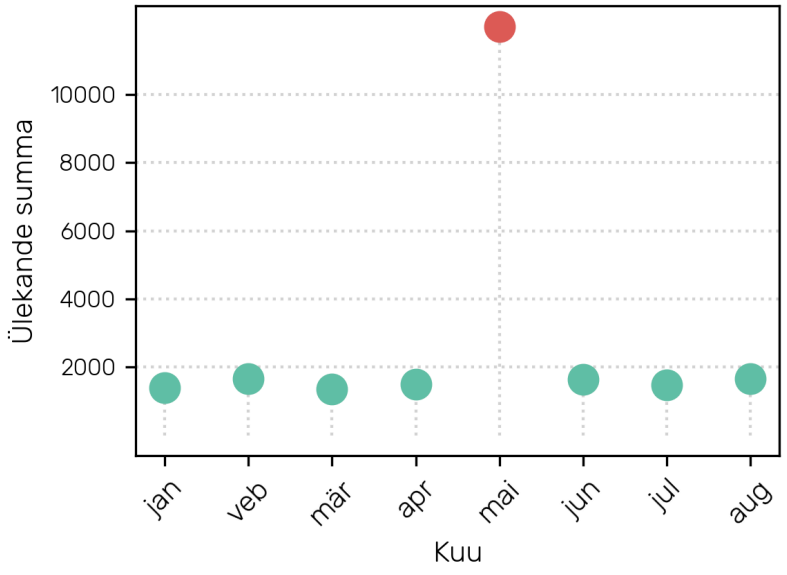
Toode	Kogus	Summa
Jahu	3219	17 334,40
Riis	1625	11 835,00
Suhkur	1307	8615,0

Tabel 3.4. Toodete risttabel.

3.3.2. Anomaaliate tuvastamine

Anomaalia on seaduspärasusest või normist kõrvalekalle ja anomaaliate tuvastamine on ootamatute fenomenide, sündmuste või vaatluste tuvastamine, mis äratavad kahtlust, sest need erinevad oluliselt ülejäänud andmetest (joonis 3.6). Anomaalseid andmepunkte ehk **erindeid** võib seostada mingisuguste probleemide või haruldaste sündmustega, näiteks pangapettustega, meditsiiniliste probleemidega, struktuursete defektidega, seadmete talitlushäiretega jne. Või hoopiski veaga andmete sisestamisel.

Andmete mõistmise etapis võib kasutada erindite tuvastamist näiteks andmete puhastamise eesmärgil. Selle analüüsi jooksul avastatakse sündmusi, mis vajavad lisauurimist, sest need tekitavad küsimusi, millele ei ole võimalik lihtsalt andmetele peale vaadates vastata.



***Joonis 3.6. Anomaalia tuvastamine**. Ühe inimese tehtud ülekannete ajalugu jaanuarist augustini. Rohelise värviga on esitatud tavapärased tehingumahud ja punase värviga on märgitud anomaalselt suur tehing. Lähtekood.*

Kirjeldav analüüs püüab andmeid kokku võtta, kasutades kirjeldavaid statistikuid nagu keskmine, mediaan, mood ja standardhälve. Erindid on andmepunktid, mis võivad kirjeldavate statistikute arvutamist oluliselt mõjutada. Need võivad viidata ka vigadele või kõrvalekalletele. Kirjeldava analüüsi jooksul peaks uurima ja otsustama, kas järgmisel andmete ettevalmistamise etapil filtreerida anomaalseid väärtusi sisaldavad näited välja või jätta need sisse. **Anomaaliate tuvastamine ja käsitlemine tagab, et andmete kokkuvõte ei ole moonutatud ebatavaliste vaatluste tõttu.**

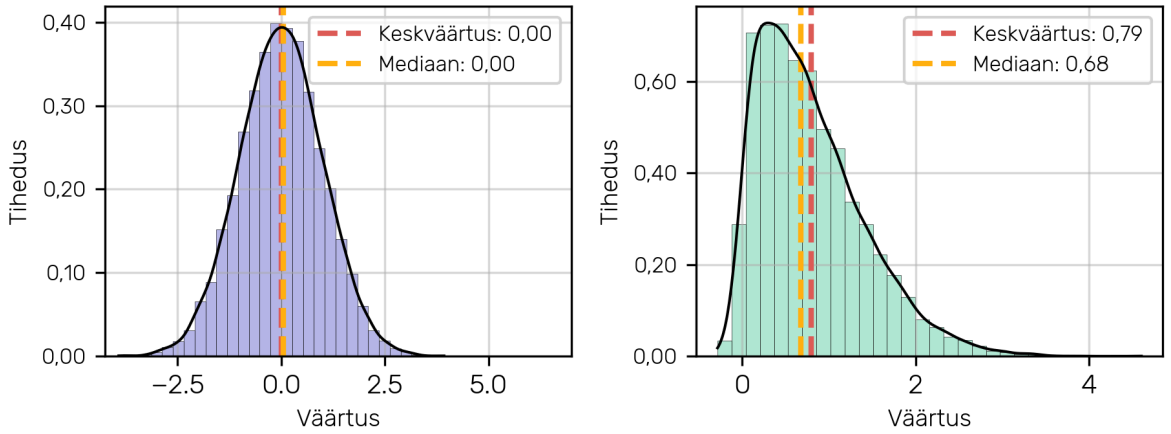
Anomaaliate tuvastamiseks võib kasutada erinevaid meetodeid, mis hõlmavad statistilisi meetodeid, nagu kvartiilhaare (ingl k *interquartile range,* mis on kolmanda ja esimese kvartiili vahe; vt alapeatükk 3.3.1), visualiseerimismeetodeid nagu karpdiagramm (joonis 3.17) või juhendamata masinõpet (alapeatükk 5.3.3.).

3.3.3 Andmete jaotuse uurimine

Andmete jaotuse uurimine on oluline andmete mõistmise komponent. Tunnuste jaotuse mõistmine aitab kindlaks teha andmete kuju ja võib olla kasulik erindite ehk anomaaliate tuvastamisel. Erinevad andmestikud järgivad erinevaid tõenäosusjaotusi (ehk teoreetiliselt kirjeldatud andmete jaotuse viis, mis võimaldab andmete kohta järeldusi teha) ning jaotuse mõistmine aitab valida sobivaid statistilisi ja masinõppe meetodeid mudeldamiseks. Andmete jaotuse visualiseerimiseks kasutatakse tihedusfunktsioonide graafikuid, sest nendest graafikutest nähtub, kui sageli paiknevad uuritava tunnuse väärtused erinevates vahemikes. See võimaldab visuaalselt mõista, kui keskväärtuse lähedal ja kui sümmeetriliselt keskväärtuse ümber tunnuse väärtused paiknevad.

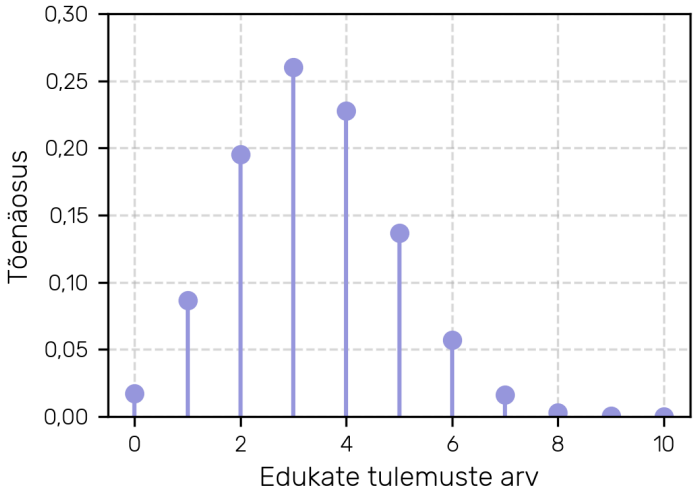
On olemas palju erinevaid tõenäosusjaotusi, millest igaühel on oma omadused ja rakendused. Nad võivad olla diskreetsed või pidevad olenevalt tunnuse tüübist. Allpool tutvustame mõningaid kõige levinumaid andmete jaotusi.

- **Normaaljaotus** (ingl k *normal distribution*) ehk Gaussi jaotus on statistikas üks olulisemaid tõenäosusjaotusi. Seda iseloomustab kellukesekujuline kõver, mis on sümmeetriline ümber keskväärtuse (joonis 3.7). Normaaljaotus on kirjeldatud kahe parameetriga: keskväärtus (μ) ja standardhälve (σ). Normaaljaotuse peamine omadus on sümmeetrilisus, mis tähendab, et jaotuse vasak pool on parema poole peegelpilt ning keskväärtus, mediaan ja mood on võrdsed. Ebasümmeetriliste jaotuste korral (joonis 3.7) on need kolm väärtust aga erinevad. Praktikas võivad andmed järgida normaaljaotust, kuid jaotuse graafik ei pruugi olla täiesti sümmeetriline kellukesekujuline kõver. See võib juhtuda erindite tõttu. Kui jaotuse kuju tekitab kahtlust, kas see on normaaljaotus või mitte, on oluline andmeid normaalsuse suhtes testida. Seda saab teha Shapiro-Wilki testiga (loe testi kohta rohkem raamatust „Statistilise andmetöötluse algõpetus“, mille autorid on Anne-Mai Parring jt, ja selle rakendamise kohta Pythoni keeles siit).



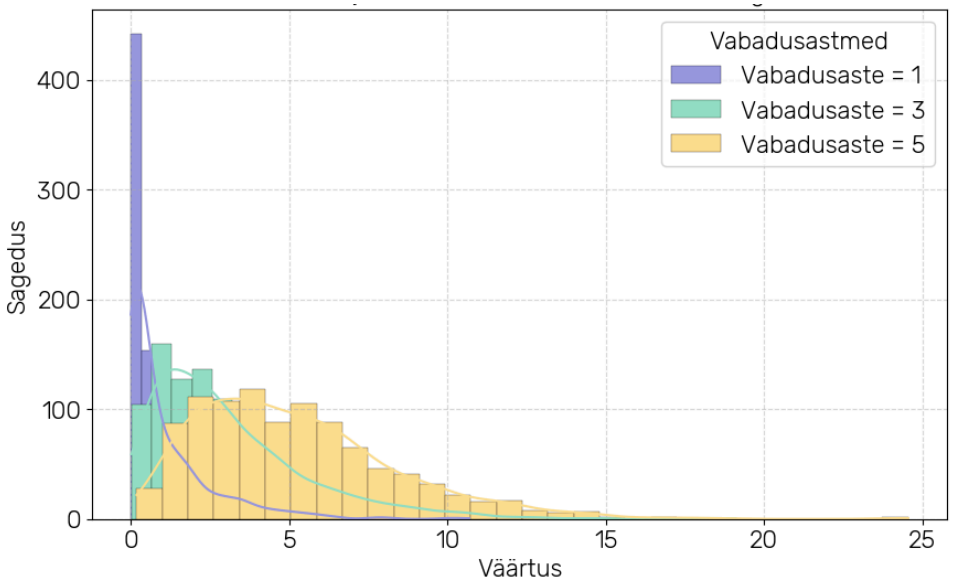
Joonis 3.7. Normaaljaotus. *Vasakpoolne graafik näitab normaaljaotust, mis on sümmeertiline ja moodustab kellukesekujulise kõvera. Keskväärtus ja mediaan on võrdsed. Parempoolne graafik illustreerib jaotust, mis on välja venitatud paremale. Ebasümmeetrilise jaotuse võivad põhjustada erindid. Graafikud sisaldavad histogrammi ja sujuvat joont selle ümber. Histogramm näitab andmepunktide sagedust kindlates vahemikes või kastides. Sujuv joon tähistab tihedusfunktsiooni kõverat, mis aitab paremini kuvada tihedusfunktsiooni kuju. Lähtekood.*

- **Binoomjaotus** (ingl k *binomial distribution*) on diskreetne jaotus (joonis 3.8). Binoomjaotus käsitleb ainult kahte olekut, mida tavaliselt esitatakse kui 1 (edu) või 0 (ebaedu). Kahe võimaliku olekuga protsessi heaks näiteks on mündivisked, aga ka täpselt number 6 saamine täringul on binaarne sündmus (juhtub või ei juhtu). Binoomjaotus annab valemi tõenäosusele saada *x* edukat katset *n* katsest, kus iga katse edutõenäosus on *p* (tõenäosus saada kull on 1/2, tõenäousus saada täringul 6 on 1/6). Binoomjaotus on aluseks igale binaarse klassifikatsiooni mudelile. Binoomjaotust kasutatakse erinevates valdkondades nagu kliinilised uuringud, rahandus, tootmine jpm. Näiteks tootmises, kui igal tootel on teatud tõenäosus olla defektne, saab binoomjaotus aidata kindlaks määrata tõenäosust leida kindel arv defektseid tooteid sõltuvalt antud partii suurusest.



Joonis 3.8 Binoomjaotus. *Graafik näitab binoomjaotust 10 katse (n = 10) ja iga katse edu tõenäosuse 0,33 (p = 0,33) puhul. X-telg esindab edukate tulemuste arvu (0 kuni 10) kümnest katsest. Y-telg esindab tõenäosust saavutada vastav arv edukaid tulemusi. Vertikaalste joonte pikkused vastavad tõenäosusele saada täpselt nii palju edukaid tulemusi. Näiteks, x = 5 asuva joone kõrgus esindab tõenäosust saada täpselt 5 edukat tulemust 10 katsest. Kuna edu tõenäosus on 0,33, on kõige tõenäolisema tulemused kolm ja neli. Lähtekood.*

- **Hii-ruutjaotus** (ingl k *Chi-squared distribution*) on pidev tõenäosusjaotus, mis on määratletud ainult mittenegatiivsete väärtuste jaoks ja mida kasutatakse paljudes hüpoteeside testides. Hii-ruutjaotuse kuju määrab parameeter *k* ehk vabadusastmete arv. Vabadusastmete arvu suurenemisega muutub jaotus laiemaks ja läheneb normaaljaotusele. Joonis 3.9 näitab hii-ruutjaotuse näiteid erinevate *k* väärtustega. Hii-ruutjaotus tekib sageli vaatluste võrdlemisel oodatavate väärtustega, mistõttu on see paljude statistiliste testide oluline alus. Lähemalt teeme juttu hii-ruuttestist peatükis 5.1.4.



Joonis 3.9. Hii-ruut jaotus. *Graafik sisaldab kolme ülekattuvat histogrammi, millest igaüks esindab hii-ruutjaotust erineva vabadusastmega (df = 1, df = 3, df = 5). X-telg esindab hii-ruutjaotuste väärtusi ning Y-telg nende väärtuste esinemissagedust. Vabadusastmete arvu suurenedes muutub jaotus laiemaks ja vähem venitatuks.*

Praktilised näited

Hea lugeja, et suurendada arusaamist õpikus käsitletud mõistetest ja saada andmete uurimise ja visualiseerimise praktilist kogemust, oleme koostanud praktilised näited koos Pythoni koodiga. Selle tarbeks kasutame Google Colabi keskkonda, mis on veebipõhine ega vaja lisatarkvara installeerimist. Selle keskkonna kasutamisega tutvumiseks saate juhiseid lugeda siit. Lisaks saate vaadata ka videoõpetust.

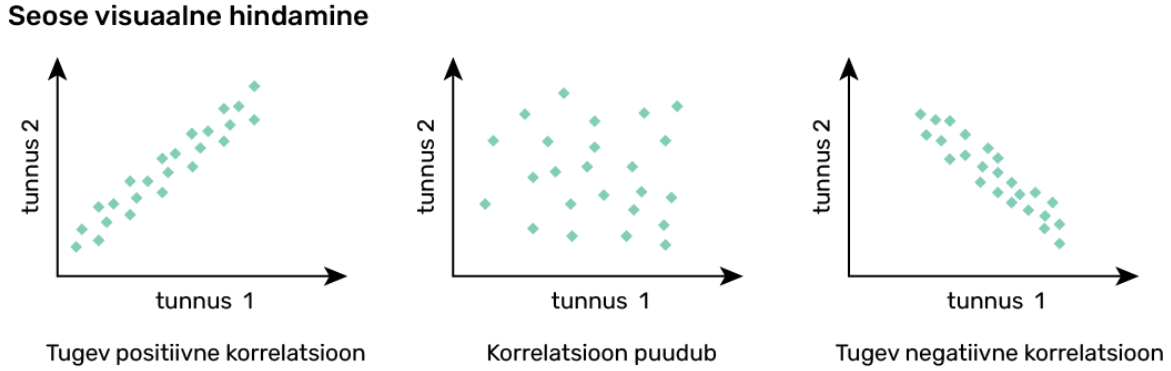
Järgige neid samme, et sellest praktikast maksimumi võtta.

1. Avage Google Colabi vihik.
2. Järgige vihikus toodud juhiseid. Iga samm on mõeldud teid juhendama andmete uurimise ja visualiseerimise protsessis, aidates teil rakendada vastavas peatükis õpitut.
3. Katsetage ja uurige. Võite vabalt muuta koodi ning uurida erinevaid andmestikke. Praktilised harjutused on olulised arusaamise kinnistamiseks ja oskuste arendamiseks.

Praktiline näide visualiseerimise kohta
Käesolevas alapeatükis kirjeldatud andmejaotuste visualiseerimise näiteid leiate sellest Google Colab vihikust.

3.3.4 Korrelatsioonianalüüs

Võimalike seoste otsimiseks ja selgitamiseks on korrelatsioonianalüüs vast kõige intuitiivsem. **Korrelatsioonianalüüs** on statistiline meetod, mis võimaldab tuvastada kahe tunnuse vahelist lineaarset seost ja selle tugevust. Seda saab kasutada siis, kui mõlemad tunnused on arvulised ja moodustaksid mingi loomuliku järjestuse. Korrelatsiooni saab hinnata nii visuaalselt kui ka arvutuslikult, koostades **hajuvusdiagrammi** (nimetatakse ka punktdiagrammiks), kus *x*-teljel on ühe tunnuse ja *y*-teljel teise tunnuse väärtused ning iga vaatluse jaoks on koordinaatteljestiku vastaval positsioonil punkt. Näiteks saame kuvada *x*-teljel toodete hinnad ja *y*-teljel selle toote ostude arvu. Joonisel 3.10 on toodud kolm võimalikku seost, mida võib hajuvusdiagrammilt olla näha: tugev positiivne korrelatsioon kahe tunnuse vahel, korrelatsiooni puudumine ja tugev negatiivne korrelatsioon. Näiteks, toote hinna ja nõudluse vahel on tavaliselt negatiivne korrelatsioon – kui hind tõuseb, siis üldiselt läheb nõudlus alla.

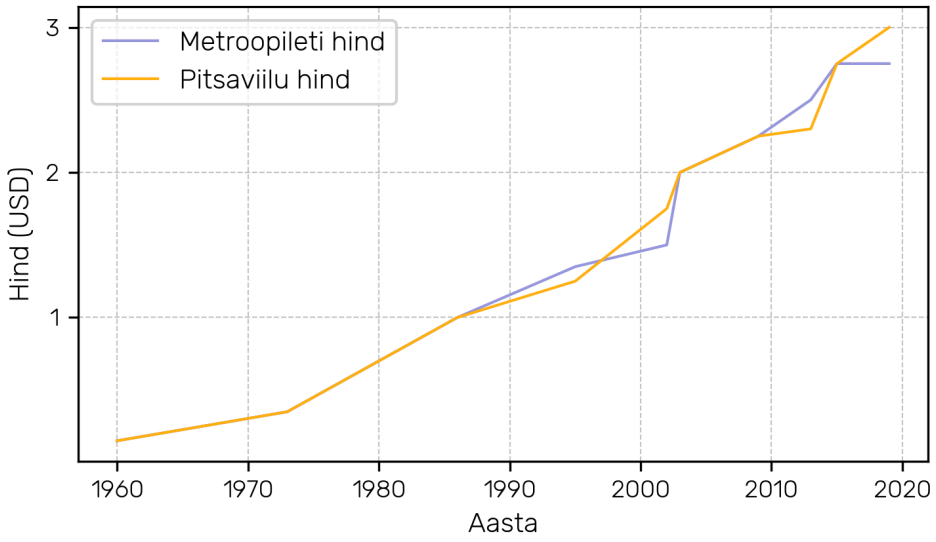


Joonis 3.10. Korrelatsiooni visuaalne hindamine.

Arvutuslikult on võimalik leida ka korrelatsiooni väärtus ja vastavalt sellele hinnata seose tugevust. Selle jaoks on erinevaid mõõte, populaarsemad on Pearsoni kordaja ja Spearmani kordaja (järjestustunnuste korral). Korrelatsioonikordaja väärtused on lõigust [–1; 1], mida suurem on kordaja absoluutväärtus, seda tugevam on seos. Kordaja märk näitab seose suunda: kui see on positiivne, siis ühe tunnuse suuruste kasvamisel suurenevad ka teise tunnuse väärtused, ja vastupidi, kui on negatiivne korrelatsioon. Korrelatsiooni võib ka tõlgendada nii, et kaks (või mitu tunnust) järgivad mingis ajaraamis sama trendi (joonised 3.10 ja 3.11).

Oluline on meeles pidada, et **korrelatsioon ei ütle midagi seose põhjuslikkuse kohta** (kas sündmuse B toimumine põhjustas sündmuse A toimumise). Avastatud seosed võivad olla ka näilised – tegelikult põhjustas mõlema näitaja muutumise mingi muu muutuja, mille kohta meil infot pole. Seega tulemuste raporteerimisel saame rääkida

muutujatevahelisest seosest, mitte ühe muutuja mõjust teisele. Klassikaline näide on tähelepanek New Yorgis, et metroosõidu hind kipub langema ja tõusma koos pitsaviilu hinnaga (joonis 3.11). See on korrelatsioon, kuid ei ole siiski põhjuslik seos: hind, mille pitsamüüjad kehtestavad oma pitsale, ei mõjuta tegelikult metroopileti hinda, ega vastupidi.

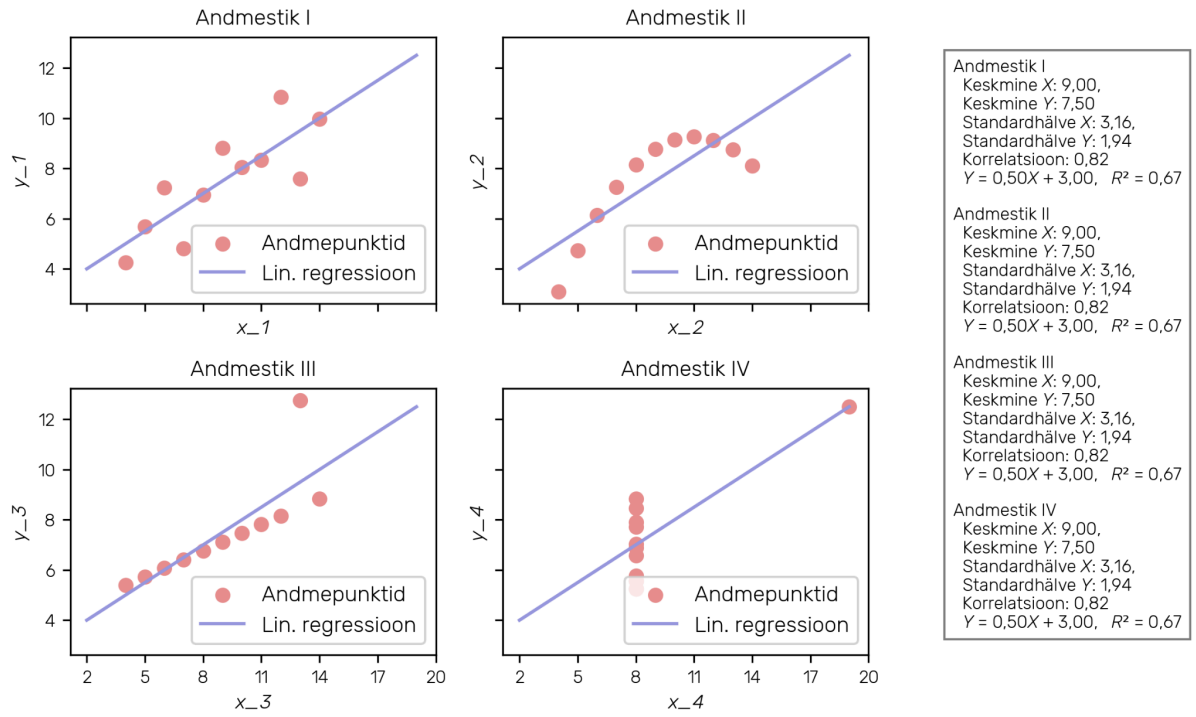


Joonis 3.11. Pitsaviilu hinna ja metroopileti hinna korrelatsioon New Yorgis. *Lähtekood.*

Paljudel juhtudel on mingite tunnuste korrelatsioon intuitiivne ja valdkonna eksperdid suudavad pikemalt mõtlemata neid seoseid selgitada. Korrelatsioonianalüüs võimaldab neid loomulikke seoseid ka andmetele tuginedes kinnitada või leida hoopis uusi, ootamatuid seoseid. Diagnostilises analüütikas näeb korrelatsiooni kasutamine välja umbes selline, et andmete kirjeldusest leitakse mingi huvitav fenomen. Siis mõeldakse andmeid vaadates ja ekspertidega vesteldes, mis võiksid olla tegurid, mis seda fenomeni põhjustavad. Siis otsitakse nende tegurite kohta andmeid kas olemasolevast andmestikust või kuskilt lisaandmeid hankides. Lõpuks leitakse hulga huvipakkuvate tegurite ja uuritava tunnuse vahelised korrelatsioonid ja hinnatakse, kas leitud tugevad seosed võiksid olla olulised ja kasulikud. Näiteks turunduses on tavaks jälgida ettevõtte võtmenäitajate (ingl k *key performance indicators*, KPI) korrelatsiooni võimalike turundusnäitajatega, et hinnata turundustegevuse mõju ettevõtte kasumlikkusele.

3.3.5 Visualiseerimine

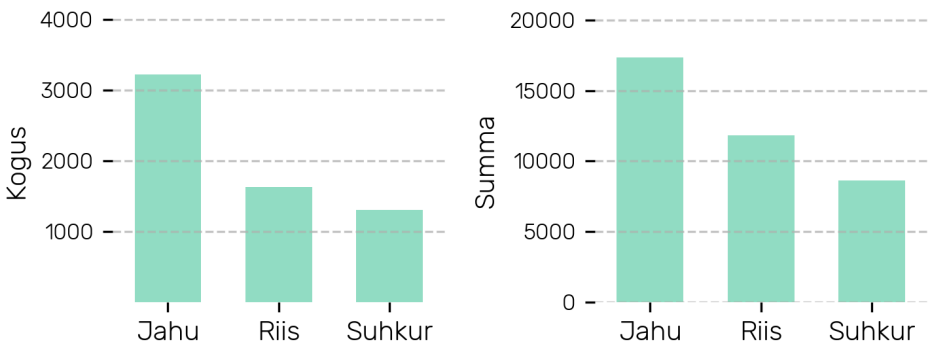
Kuigi statistikute arvulised väärtused kirjeldavad andmeid juba päris hästi, annab visuaalne ülevaade palju rohkem aimu sellest, mis andmetes tegelikult toimub. Ka kirjeldavate statistikute visualiseerimine annab palju juurde nende tõlgendamisele. Joonisel 3.12 on toodud näide, kus nelja andmestiku peamised kirjeldavad statistikud on identsed, aga andmete jaotus on tegelikult erinev ja seda on lihtsam märgata just visualiseerides.



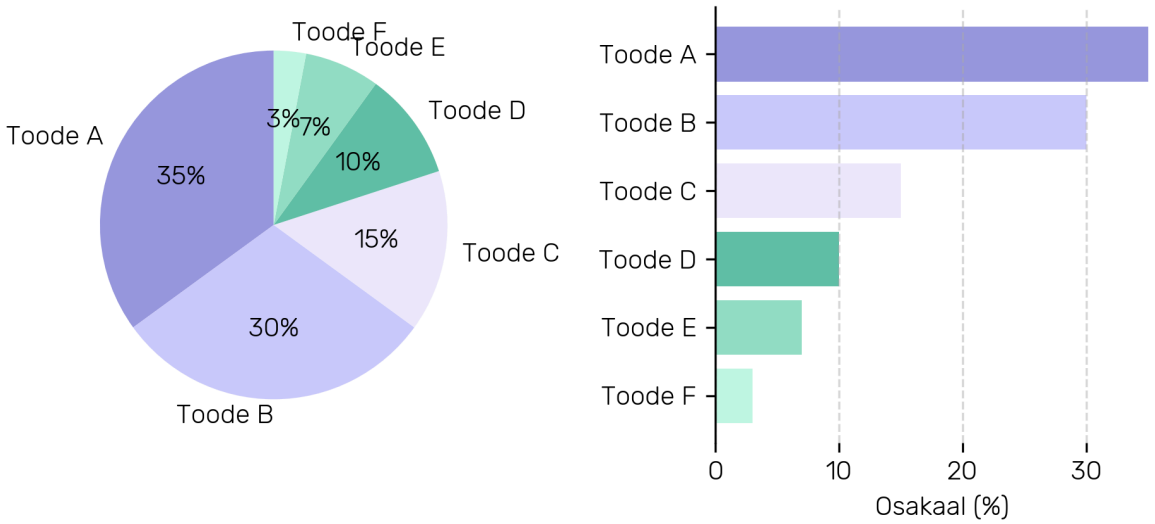
Joonis 3.12. Andmete visualiseerimise olulisus. *Anscombe'i kvartett koosneb neljast kahe tunnusega (x, y) andmestikust, mille tavaliselt andmete kirjeldamiseks arvutatavad mõõdikud on identsed, kuid mis tegelikult ei ole sarnased. See kvartett demonstreerib elavalt, et oma silm on kuningas ja kokkuvõtvatest arvudest ei pruugi piisata, et andmete olemust mõista. Lähtekood.*

Andmete visualiseerimise eesmärke on kaks: informatsiooni kuvamine ja andmete analüüs. Viimane kuulub pigem diagnostilise analüütika juurde, sest me saame graafikuid vaadates analüüsida, miks me mingeid seoseid näeme. Andmeid saab visualiseerida mitmel viisil ehk kasutades eri tüüpi graafikuid. Olenevalt tunnuste tüübist ja visualisatsiooni eesmärgist on võimalik valida ka sobiv graafikutüüp. Põhilised graafikud, mida kirjeldavas analüüsis kasutatakse, on tulpdiagramm, histogramm, karpdiagramm ja joondiagramm. Neid graafikuid saab kasutada nii üksikute tunnuste väärtuste visualiseerimiseks kui ka väärtuste võrdlemiseks erinevate gruppide vahel.

Tulpdiagramm on joonis, kus tulba kõrgus ehk pikkus näitab hulka. Sisuliselt on tulpdiagrammi joonistamise alusandmeteks väärtused mingist risttabelist (joonis 3.13). Erinevalt risttabelist on jooniselt palju lihtsam võrrelda erinevaid arvulisi koguseid. Klassikaline tulpdiagramm võimaldab tulpade abil hõlpsasti võrrelda eri kategooriate arvulisi väärtusi. Joonise ühel teljel (tavaliselt *x*-teljel) näidatakse tunnuse kategooriaid, teisel teljel (tavaliselt *y*-teljel) väärtuste skaalat. Väärtusi saab esitada nii absoluutarvudes kui ka protsentidena. Tulbad järjestatakse ajalise või suuruse kasvamise/kahanemise järjekorras. Kui tunnusel on väga palju kategooriaid, siis on visuaalselt kasulik tulpdiagramm ümber pöörata nii, et kategooriad on *y*-teljel ja vastavad arvulised väärtused *x*-teljel. Osakaalude (protsentide) näitamiseks on populaarne kasutada ka **sektordiagrammi**. Kuigi sektordiagrammid on näiliselt ilusad, võivad need olla visuaalselt petlikud: sektori tegelik osakaal tervikust ei ole hästi arusaadav, kui on palju eri kategooriaid (joonis 3.14). Seega on osakaalude näitamiseks soovitatav kasutada ikkagi tulpdiagramme.

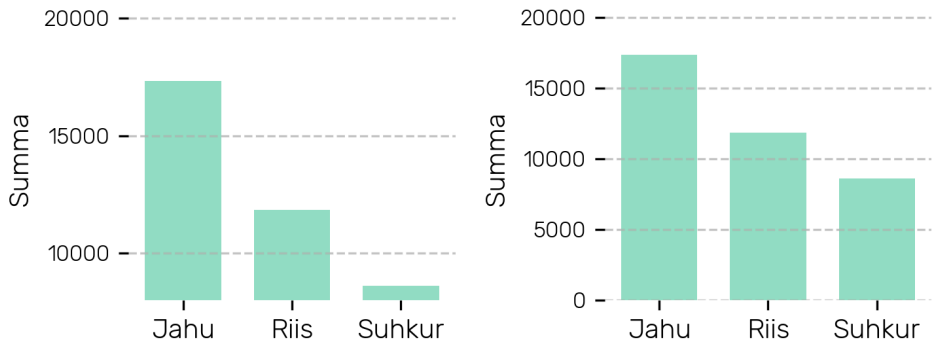


Joonis 3.13. Toodete koguse risttabel tulpdiagrammidena. *Algandmeid on tabelis 3.4.* *Lähtekood.*



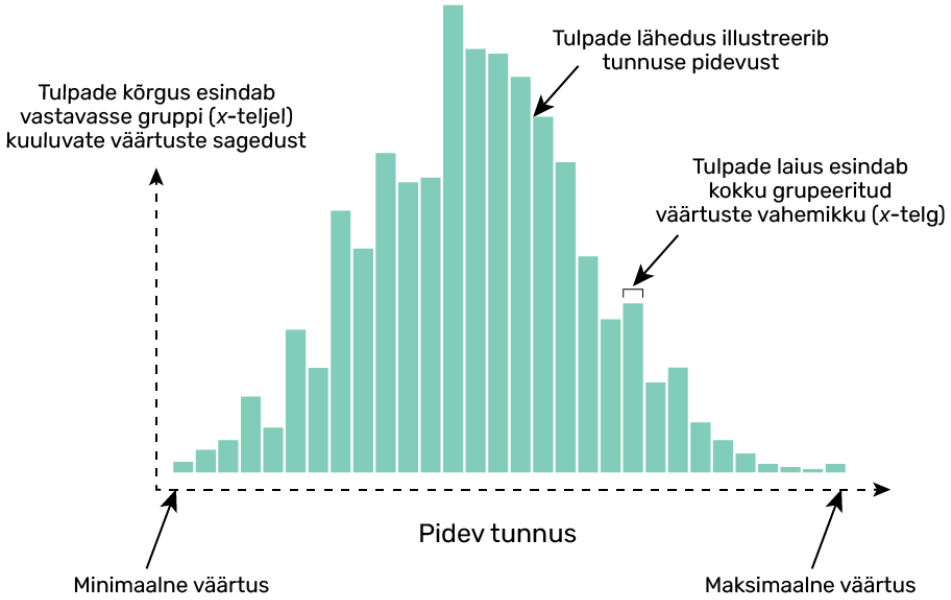
Joonis 3.14. Sektordiagramm ja tulpdiagramm. *Sektordiagramm võib olla visuaalselt petlik ja parem on kasutada tulpdiagrammi. Kas te suudate toodud sektordiagrammil olevad tooted järjestada vastavalt nende osakaalule müügist? Aga paremal asuvalt tulpdiagrammilt?* *Lähtekood.*

Üks olulisi asju, mida nii tulpdiagrammide kui ka muude diagrammide koostamisel tuleb jälgida, on telgedel kuvatavate väärtuste vahemikud. Oluline on, et kuvatud oleks kogu võimalik vahemik. Vastasel korral on tegelikud andmed visuaalselt moonutatud ja selle tagajärjeks võivad olla väga valed otsused. Näiteks, joonisel 3.15 on kõrvuti toodud kaks tulpdiagrammi, mis tegelikult näitavad täpselt samu andmeid, ainuke erinevus on *y*-telje väärtuste vahemikus.



Joonis 3.15. Andmete visuaalne moonutamine. *Visuaalne andmete esitamine on võimas vahend teabe edastamiseks. Kuid andmete esitamise viis võib oluliselt muuta vaataja tõlgendust. Vasakul graafikul on y-telje minimaalseks väärtuseks valitud 8000, mis loob mulje, et jahu müügisumma erineb drastiliselt teiste toodete omast. Hulgimüügi ettevõtte juhid võivad järeldada, et jahu müük on nii palju suurem, et ettevõte peaks oma strateegias keskenduma ainult jahu müügile. Muutes y-telje väärtuste ulatuse 0-st 20 000-ni, on pilt palju usaldusväärsem. Müügi erinevused on palju väiksemad, pakkudes täpsemat tuge otsuste tegemiseks.* [*Lähtekood.*]

Arvtunnuste (eriti pidevate tunnuste) korral, millel on palju erinevaid võimalikke väärtusi, kasutatakse **histogrammi** (joonis 3.16). Näiteks on histogramm sobilik inimeste pikkuste jaotuse või ettevõtte päevaste külastuste ajalise jagunemise näitamiseks. Histogrammi koostamiseks ei sobi näiteks tunnused nagu sugu ja tootegrupp, sest need ei ole arvulised tunnused. Ehituselt on tegemist tulpdiagrammiga. Pideva arvulise tunnuse väärtuste ulatus (suurima ja vähima väärtuse vahe) jagatakse ette määratud (tavaliselt võrdseteks) osadeks ning iga osa jaoks leitakse andmetest neid väärtusi esindav kogus (sagedus) ja kuvatakse see *y*-teljel. Visuaalselt tähendab see ka seda, et histogrammil on tulbad üksteisele väga lähedal.

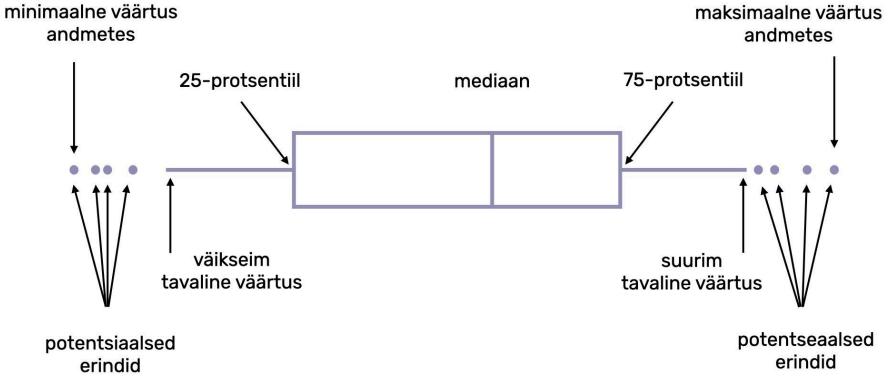


Joonis 3.16. Histogrammi koostamine.

Kui pideva tunnuse kohta teha tulpdiagramm, siis see tähendaks, et iga võimalik arvuline väärtus, mis andmetes esineb, on eraldi kuvatud *x*-teljel, mis teeb pildi kasutuks ja

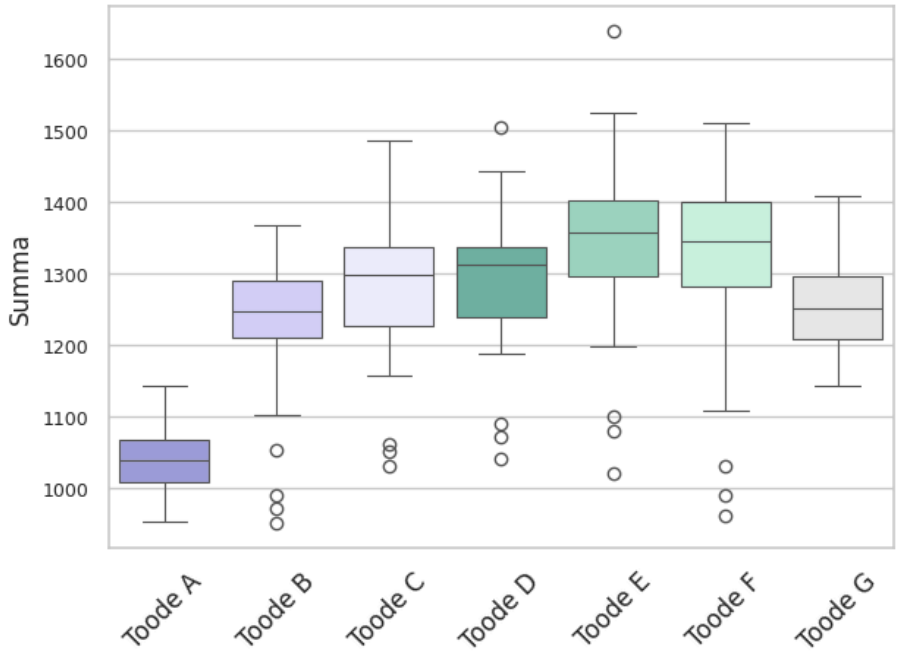
koledaks. Histogrammilt on võimalik näha, kas mingi tunnuse väärtused meie andmestikus jagunevad ühtlaselt – kõik tulbad on sama kõrged – või leidub mingeid trende – näiteks on kaks väärtuste vahemikku, kus sagedus on suurem ja tulbad kõrgemad kui teistes vahemikes.

Arvuliste tunnuste väärtuste jaotuse uurimiseks on hea kasutada ka **karpdiagrammi** (joonis 3.17), mis visualiseerib kirjeldava statistika näitajaid nagu mediaan ja kvartiilid ning võimaldab näha ka võimalikke erindeid.



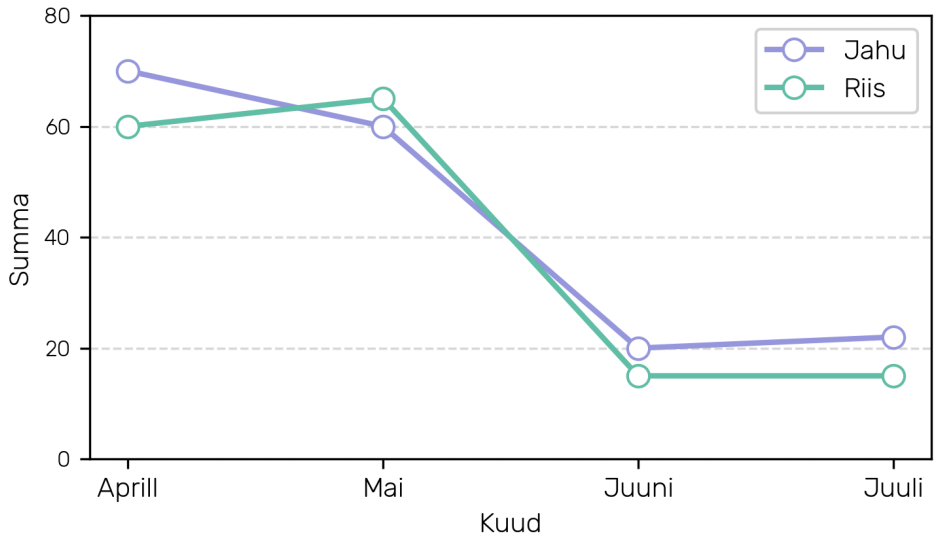
***Joonis 3.17. Karpdiagrammi osad**.*

See joonis võimaldab ka võrrelda tunnuse jaotusi eri kategooriates. Näiteks saab võrrelda erinevate tootegruppide müüki (joonis 3.18). Karpdiagrammi abil on võimalik tuvastada ka võimalikke valesti sisestatud väärtusi ja muid erindeid, mille peaks võib-olla analüüsist eemaldama. Näiteks kuu käive 9000 eurot võib olla valesti sisestatud ja tuleks karpdiagrammi jooniselt erindina kohe välja



***Joonis 3.18. Karpdiagramm erinevate toodete müügisummade kohta.** Jooniselt on selgelt näha, millised tooted erinevad üksteisest mediaani poolest (joon karbi keskel), kui suur on andmete varieeruvus (kastide servad ja joone pikkus) ja kas leidub erindeid (vahemikust välja jäävad punktid). Veelgi informatiivsema alternetiivina võib kasutada viiuldiagrammi. Lähtekood.*

Dünaamika ja tendentside visualiseerimiseks on väga sobilik **joondiagramm**. Joondiagramm saadakse andmepunktide koordinaatteljestikku kandmisel ja nende ühendamisel joonega. Enamasti kujutab *x*-telg ajaskaalat või muud pidevat mõõtmist ja *y*-telg huvipakkuva tunnuse mõõdetud või arvutatud väärtust. Joondiagramm sobibki väga hästi just ajaliste trendide visualiseerimiseks pidevate andmete korral, näiteks müügitulu muutused läbi kuude (joonis 3.19). Joonisel 3.19 toodud joondiagrammil on võrdluseks toodud kahe toote müük läbi nelja kuu. Kohe on näha, et nende toodete müügimaht on sarnane (jooned on üksteisele lähedal) ja ka müügitrendid on samasugused (jooned on paralleelsed). Lisaks on märgata, et mõlema toote müük vähenes suve saabudes (juunis).



Joonis 3.19. Joondiagramm kuu müügisummade võrdluseks. Lähtekood.

Siin alapeatükis kirjeldasime lühidalt põhiliste graafikutüüpide omadusi ja nende kasutamise eesmärke. Vastavalt huvipakkuvale küsimusele saab kõiki neid graafikuid koostada näiteks erinevate kategooriliste tunnuste kaupa grupeerides (näidates gruppe eri värviga). Üldiselt võiks kirjeldavaks analüüsiks kasutatud graafikute koostamisel jälgida järgmisi reegleid:

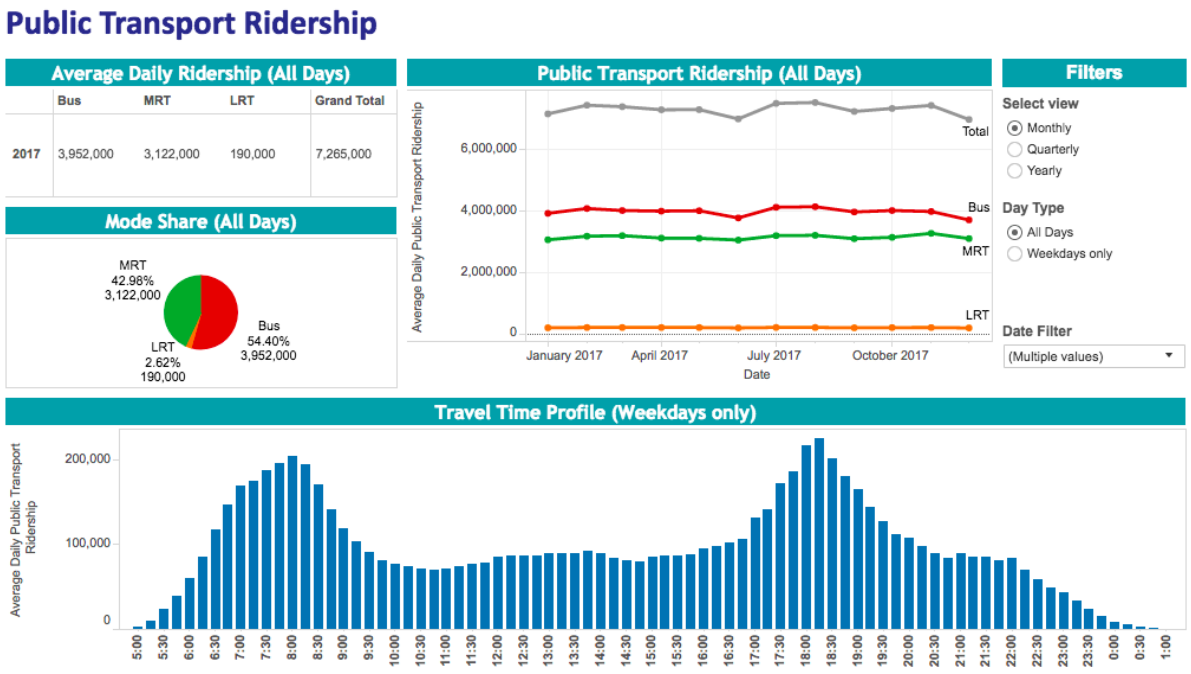
- kasutada tuleb tunnusetüübile sobivat graafikut;
- telgesid tuleb alati näidata täies ulatuses (vt joonis 3.15);
- graafikud tuleb hoida selged ja lihtsad (ilma „tulede ja viledeta“);
- andmete rõhutamiseks võiks kasutada värve, näiteks tavaliselt kuvatakse tulpdiagrammidel sama taseme kategooriad sama värviga (vt joonis 3.13).

Hea lisamaterjal (küll ainult inglise või vene keeles) erinevatest (ka keerukamatest) graafikutest on saadaval siin: *https://datavizcatalogue.com/*.

3.3.6 Interaktiivne andmete töölaud

Interaktiivne andmete töölaud (ingl k *data dashboard*) on visualiseerimise tööriist, mis kuvab andmeid ja mõõdikuid ühes kohas, võimaldades kasutajal jälgida, analüüsida ja esitada olulist teavet selgel ja arusaadaval viisil. See võimaldab ettevõtte seisul jooksvalt silma peal hoida ja vajaduse korral ka andmetesse sügavamale vaadata, et oleks lihtsam kasulikke järeldusi või otsuseid teha. Selleks visualiseeritakse ettevõtte jaoks olulisi näidikuid reaalajas. See tähendab, et kui andmed mingil ajahetkel muutuvad, siis muutub ka töölaua vaade. Neid vaadates saavad kasutajad korraga kätte kogu info, mida neil võib vaja minna edasiste otsuste tegemiseks. Lisaks, näiteks tootvas tööstuses võimaldavad interaktiivsed töölauad reaalajas jälgida, kas kõik protsessid toimivad nii, nagu on ette nähtud, ja ootamatutele tõrgetele kiirelt reageerida.

Interaktiivse töölaua koostamisel kasutatakse neidsamu põhilisi graafikutüüpe, nagu kirjeldasime eelmises alapeatükis (joonis 3.20), aga graafikutele on lisatud ka interaktiivsus. See tähendab, et on võimalik teha erinevaid andmeid filtreerivaid valikuid nagu ajaperiood või tootegrupp, mille tulemusena näidatavad graafikud muutuvad. Nii on võimalik andmete üldpildist sisse liikuda erinevatesse detailsusastmetesse, vastavalt sellele, mida loodud töölaud valida võimaldab. Tehniliselt võib töölaud olla väga keerukas – graafikute valimine, interaktiivsuse loomine, õigete andmete valimine, andmebaasiga sidumine jne – aga paljud ärianalüütika platvormid, näiteks Tableau ja Microsoft PowerBI, on nende loomise märksa lihtsamaks teinud.



Joonis 3.20. Kuvatõmmis andmete töölauast. *Töölaud koondab erinevad graafikud ja statistikuid (tulpdiagramm, joondiagramm, keskmised, koguarvud jne), et ühe vaatega anda andmetest kokkuvõttev ülevaade.*[8]

[8] Allikas: flickr.com, litsents PDM 1.0.

Vastavalt äriküsimustele ja kasutajatele jagatakse interaktiivsed töölauad kolmeks tüübiks: **operatsiooniline**, **strateegiline** ja **analüütiline töölaud** (joonis 3.21). Olenevalt töölaua eesmärgist ja kasutajast on oluline valida sobivad tunnused ja muud näitajad, mida töölaual kuvada. Ilmselt ei ole hea mõte üritada ühte kuva üle koormata kõigi näitajatega, et üks vaade sobiks nii juhtkonnale ettevõtte üldise seisu hindamiseks kui ka andmeanalüütikule müügitulu languse põhjuste otsimiseks.

Tüüp	Eesmärk	Kasutajad	Nõuded
Operatsiooniline	Ajaliselt kriitiline andmete hetkeseisu ülevaade	Valdkonnajuhid	Andmed ei pea olema väga täpsed, aga neid tuleb kuvada reaalajas
Strateegiline	Organisatsiooni KPI-de ülevaade	Juhtkond	Andmed peavad olema täpsed ja õigeaegselt kättesaadavad (kord kuus, kvartalis vms)
Analüütiline	Trendide ja uute teadmiste avastamine	Äri- ja andmeanalüütikud	Andmed peavad olema täpsed ja puhtad, ajalist nõuet ei ole

Tabel 3.5. Andmete töölaua tüübid ja nende omadused.

3.3.7 Kirjeldava analüüsi praktiline kasutamine ja kasulikkus

Kirjeldava analüüsi tulemused on äärmiselt väärtuslikud ja neid **saab kasutada kahel viisil**. Esiteks, kirjeldava analüüsi tulemused on olulised **sisendid andmete ettevalmistamise ja modelleerimise etapile**, aidates mõista, millist eeltöötlust andmed vajavad ja milliseid meetodeid tuleks kasutada mudeldamisel. Teiseks, kirjeldav analüüs **toetab äritegevuse operatiivset juhtimist, pakkudes aruandeid ja interaktiivseid töölaudu**, mis aitavad jälgida ja optimeerida igapäevaseid äriprotsesse. Mõlemad kasutusviisid on olulised, et tagada andmete tõhus kasutamine ja toetada otsuste tegemist.

Kirjeldava analüüsi tulemustel on määrav roll andmete ettevalmistamise ja mudeldamise etapis. See võimaldab mõista, milliste andmetega me tegeleme, millist eeltöötlust need vajavad ja milliseid mudeldamise meetodeid saab nende andmete puhul kasutada. Näiteks võib kirjeldav analüüs paljastada puuduvad väärtused, erindid või andmete jaotuse omadused, mis otseselt mõjutavad eeltöötluse meetodite, statistiliste testide või masinõppe mudelite valikut. Lisaks aitab see kindlaks teha, millised tunnused on kõige olulisemad ja kuidas neid tuleks mudeldamisel arvesse võtta. Seega, kirjeldav analüüs on esmane ja hädavajalik samm, mis teeb tõhusamaks edasise analüüsi ja mudeldamise.

Kirjeldava analüüsi tulemused ei ole olulised ainult CRISP-DM-i andmete ettevalmistamise ja mudeldamise etapis, vaid neid saab kasutada ka äritegevuste kontekstis otsuste langetamiseks. Otsuste vastuvõtmisel on kasulik, kui need tuginevad andmetele. Andmetele saab tugineda, kui need on hästi esitatud ja selgelt kirjeldustega kokku võetud. Kirjeldav analüüs on tõenäoliselt kõige tavapärasem ärianalüüsi lahendus, mille tulemuseks on aruanded või interaktiivsed töölauad. Aruanded kajastavad enamasti mingi fikseeritud hetke olukorda. Aga tihti on tarvis lahendusi, mis on

kasutusel operatiivinfo saamiseks, näiteks: milline on kaupade hetke laoseis, kuidas jaguneb elektritarbimine päeva jooksul, millised kliendid on võlgu ja kui palju. Paljud majandustarkvaralahenduste koosseisus olevad aruanded katavad just selle taseme analüüsivajadused. Kasutajatel on neid aruandeid lihtne kätte saada ja tarbida. Praktikas tähendab aruande koostamine seda, et sisestatakse valik parameetreid (ajaperiood, osakond, jne) ja vastu saadakse kirjeldav tabel või graafikud. Interaktiivne töölaud on aga nagu aruanne, millele on lisatud interaktiivsus (saab valida, mida näha soovitakse) ja soovi korral ka automaatne sidumine andmebaasiga – see tähendab, et kohe, kui andmed muutuvad, kajastub see ka interaktiivsel töölaual.

Kasutamise näide ja mõtlemisülesanne
Oletame, et Mari on ühe toidupoe omanik ja tahab optimeerida toodete valikut oma poes. On oluline, et poe riiulitel poleks palju tooteid, mis ei müü väga hästi, aga võtavad palju ruumi. See võib poe müügitulule väga halvasti mõjuda ja seda tuleks optimeerida. Et valikuid ja otsuseid teha, võiks kõigepealt näiteks uurida, mis üldse on poe kõige populaarsemad ja kõige ebapopulaarsemad tooted. Selleks saame kirjeldada ning visualiseerida müügitulu suurust tootekategooriate kaupa, näiteks lihatooted, kuivained ja piimatooted.
Siinkohal võite mõelda, kuidas saaks seda graafiliselt esitada. Millist graafikutüüpi te kasutaksite?
Mari on loonud oma poe müügiandmete uurimiseks interaktiivse andmete töölaua, kus on võimalik näha mitmesuguseid kokkuvõtvaid statistikuid tootekategooriate kaupa. Samuti saab seal sisse suumida kategooriatest toodete tasandile, et detailselt näha, milliste toodete müük läheb hästi ja millistel mitte nii hästi. Näiteks paistab sealt, et kuivainete müük edeneb väga hästi ja sügavamale sisse vaadates näeme, et eriti hästi müüvad hommikuhelbed. Kõige populaarsemad on Cini Minis ja Nesquiki šokolaadipallid, vähem populaarsed on klassikalised Kelloggsi maisihelbed. Selle ülevaate põhjal saab Mari juba otsustada, milliseid tooteid peaks rohkem tellima ja milliseid vähem. Mari saab rääkida oma töötajatega, et arutada, kas tellida tulevikus rohkem Cini Minis helbeid ja vähem maisihelbeid.

Praktiline näide
Andmete mõistmise peatüki praktilise näite leiate sellelt lingilt: Google Colabi vihik

Enesekontrolli küsimused

1) Mis on "tunnus" andmeteaduse kontekstis?
    a) Andmetabeli rida, mis esindab objekti.
    b) Andmetabeli veerg, mis esindab mõõdetud omadust.
    c) Andmetabeli pealkiri, mis selgitab andmete sisu.
    d) Arvutustabelis olev valem.

2) Mida tähendab "esinduslik andmestik"?
    a) Andmestik, mis sisaldab ainult suurimaid väärtusi.
    b) Andmestik, mille read on juhuslikult valitud ja peegeldavad päriselus leiduvaid proportsioone.
    c) Andmestik, mille veerud on kõik pidevad arvulised tunnused.
    d) Andmestik, mis sisaldab ainult täielikke andmeridu.

3) Miks peaks ühe andmetabeli veerg sisaldama ainult ühte tüüpi infot? Too näide, kuidas seda reeglit rikkudes võib tekkida probleem.

4) Kuidas tuleks Excelis puuduvad väärtused tähistada, et tarkvara saaks neid õigesti tõlgendada?

5) Kirjelda ühte meetodit anomaaliate tuvastamiseks andmetes.

4. Andmete ettevalmistamine

Andmete ettevalmistamine (ingl k *data preparation*) on protsess, mille käigus korrastatakse toorandmed nii, et neid saaks valitud andmeteaduse meetodeid kasutades analüüsida. See on tõenäoliselt kõige olulisem ja aeganõudvam osa andmeteaduse projektist. Hinnanguliselt võib see võtta 50–70% projektile kuluvast ajast. Head andmed on andmeteaduse alus, see teadmine on nii laialt levinud ja kinnistunud, et on jõudnud tihti kasutavatesse väljenditesse nagu „prügi sisse, prügi välja“ (ingl k *garbage in, garbage out*) ning „esmalt andmed“ (ingl k *data first*).

Eelnevas projekti sammus oleme me saanud hinnangud erinevate andmeallikate kvaliteedi, mahu, formaadi ja kättesaadavuse kohta. Selles etapis tuleb selle teadmise alusel valida andmed, mida kasutada, ja need mudeli jaoks sobivale kujule viia. Samuti on vaja eraldada mingi hulk testandmeid, millel mudelit hiljem hinnata. Täpsemalt selle kohta, miks ei tohi mudelit hinnata mudeli loomiseks kasutatavatel andmetel, saab lugeda peatükist 5.

4.1 Andmete valimine, loomine ja ühendamine

Potentsiaalselt on meil palju andmeallikaid, näiteks tabeleid või videokaamera salvestisi või klientide kõnesid klienditeenindusse. Et mitte kasutada kõige klassikalisemat näidet kahe tabeli kujul andmestiku ühendamisest, võtame mitmekesisema näitena telekomifirma klienditoele tehtud kõnede salvestised ja nende kohta teada oleva muu info. Kujutlegem, et meil on

- salvestatud kõned, pikkusega 5 sekundit kuni 15 minutit (pikkusi ja nende jaotust teame andmete mõistmise ja kirjeldamise etapist);
- tabeli kujul info nende kõnede kohta: kõne täpne aeg, kõne kestus, kõne üldine teema (kliendi nupuvajutuse alusel), kõne täpsem teema (nupuvajutus või klienditeenindaja korrigeeritud), kas kõne suunati tehnilisele toele, helistaja kliendi-ID, helistaja telefoninumber, klienditeenindaja ID, kas probleem sai lahendatud jne;
- klientide tabel, kus on kirjas kliendi aadress, vanus, sugu, telefoninumber, e-postiaadress jpm. Erakliendi ID on seotud füüsilise isikuga, seega on ID sama isiku jaoks püsiv läbi mitme lepingu;
- lepingute tabel, kus on kirjas info lepingute tüübi, kestuse ja hinna kohta, samuti kliendi ID, teenindaja ID (kes lepingu vormistas) jne. Lepingute tabelis on ka juba lõppenud lepingud. Ühel isikul võib olla mitu lepingut;
- ja kindlasti veel palju muud infot (näiteks meie firmalt renditud, ostetud, liisitud seadmete kohta).

Andmete kogumise, kvaliteedi ja muude varem tehtud raportite alusel teame, et kõned pärinevad viimasest kümnest aastast ja neid on kokku kümneid tuhandeid. Helistaja ID on olemas ainult juhtudel, kui seda oli vaja tuvastada. Teame ka, et klienditeenindajad ei sisesta mõnesekundiliste, sisuliselt mittetoimunud kõnede kohta palju infot. Kõigi kõnede kohta on sisestatud telefoninumber, kuid veerus esineb ka väärtusi „peidetud number“. Kõiki telefoninumbreid ei leia meie klientide telefoninumbrite nimekirjast.

Samuti pole kindel, et inimene helistab seoses enda lepinguga, mitte oma 90-aastase isa lepingu tõttu. Kõige sellega peame arvestama hiljem neid tunnuseid analüüsis kasutades ja tulemusi tõlgendades. Infot tunnuste olemuse, kogumisviisi jms kohta nimetatakse metainformatsiooniks ning see selgitatakse välja andmetega tutvumise faasis.

Kõigis andmeallikates on infot kokku väga palju. **Millist osa andmetest kasutada, oleneb analüüsi eesmärgist.** Kui soovime hinnata klienditeenindajate töökust ja suutlikkust klientide probleeme lahendada, piisab võib-olla ainult tabeli kujul andmetest kõnede kohta. Kui soovime teada, millised kliendid meile helistavad, võib vaja minna nii kõnede, klientide kui ka lepingute tabelit. Seega tuleb valida, millised tabelid ja **millised erinevates tabelites leiduvad tunnused** meie analüüsiks kasulikud võivad olla, ning kasutada edaspidises analüüsis ainult neid tunnuseid. On meil tegelikke helisalvestisi üldse analüüsiks vaja? Kasutamata jätmise aluseks võib olla ka puuduvate väärtuste hulk või märgendite madal kvaliteet (nt avastasime, et helistaja telefoninumber ei lähe tihti kokku kliendi ID kohta teada oleva telefoninumbriga, ja otsustasime, et see on liiga mürane infoallikas).

Kasutatavate tunnuste valimisele lisaks võib hoopis ka **luua uusi tunnuseid**. Tihti on veergude arvu vähendamiseks mõistlik mitme tunnuse põhjal mingi uus tunnus arvutada, mis võtaks ühe arvuga kokku kõik algsetes tunnustes sisalduva info. Näiteks võib pikkusest ja kaalust arvutada kehamassi indeksi või tuletada meie lepingute tabeli alusel teadmise kliendisuhte pikkusest teatud erakliendiga, mis võimaldab uurida, kas uutel klientidel on rohkem muresid ja kas need on lepinguga seotud või tehnilised. Samuti võivad uued tunnused tekkida ridade summeerimisel või keskmistamisel, näiteks kõigi sama kliendi tehtud tehingute koguväärtus või keskväärtused mingi ajaperioodi kohta. Et tundlikke andmeid peab kasutama võimalikult väikse täpsusega (see nõue tuleneb isikuandmete kaitse üldmäärusest, IKÜM, ingl k GDPR), tuleb täpsed vanused muuta vanusegruppideks ja aadressid piirkondadeks. Universaalseid vastuseid, milliseid tunnused eemaldada või juurde luua, pole ja tihti tulevad selles protsessis kasuks ka valdkonna teadmised. Võimaluse korral tuleks otsustamisse kaasata vastavad eksperdid, äriettevõttes näiteks müügijuhid.

Uusi tunnuseid võib luua ka tehniliste vahendite abil, näiteks struktureerimata andmetest mingit infot välja võttes. Tekstide analüüsimiseks on palju tööriistu, mis klassifitseerivad teksti sisu mingitesse klassidesse, näiteks kas see sisaldab solvavat sisu, kas see on positiivse või negatiivse tooniga, mis keeles see tekst on, mis on olulised märksõnad jne. Helisalvestiste puhul võime automaatselt kõne transkribeerida ja sama teha. Võime ka helikvaliteeti automaatselt hinnata, kõneleja sugu ja vanust kindlaks määrata.

Uusi tunnuseid võib otsida ka väljastpoolt ettevõtet. Näiteks võib suur hulk pöördumisi telekomifirma klienditeenindusse olla seotud tormikahjuga. Kui me püüame mõista, miks ja millal kliendid meie poole pöörduvad, oleks vaja seda arvesse võtta. Ilm võib mõjutada ka poodide müüki – inimesed võivad vältida tormiga kodunt välja minemist või just vastupidi, kehva ilmaga veeta rohkem aega kaubanduskeskuses.

Andmeid on vaja valida ka ridade kaupa – **milliseid näiteid kasutada**. Kui me soovime kasutada kõnede transkripte, võib olla vaja välja jätta **madala** heli**kvaliteediga** kõned, mille transkriptid on ebatäpsed. Võib-olla on vaja jätta kõrvale kõned, mis pole kliendi ID-ga seotud (**puuduvad andmed**). Võib-olla huvitavad meid kõned ainult teatud teemadel, seega valime ainult teatud read kõnede tabelis. Näiteks, võime analüüsida üksnes tehnilise probleemiga seotud kõnesid, et küsida, kui palju ja millised probleemid lahendas esmane klienditeenindaja ja millised suunati tehnilisele toele. Samuti võib kõrvale jätta liiga vanad kõned, liiga pikad või lühikesed kõned.

> Valimata jäetakse madala kvaliteediga või analüüsi jaoks mitteolulised tunnused.
>
> Valimata jäetakse madala tunnuste kvaliteediga või analüüsi jaoks mitteolulised näited.

Ka täielike ja kvaliteetsete tunnustega andmepunkte võib olla vajalik analüüsist kõrvale jätta, sest tihti on oluline andmete **esinduslikkus**. Kui meil on kehvem kõnede kvaliteet maapiirkondadest helistajate puhul või mingi vanusegrupi puhul (nt meie kõnetuvastus ei tööta hästi vanade inimeste häälel ja sõnavaral) ning paljud neist kõnedest on vaja kõrvale jätta, siis on andmestiku esinduslikkuse säilitamiseks vaja ka teistest gruppidest andmeid vähemaks võtta ehk alavalida (ingl k *subsampling*). Alternatiiv on puudulikud andmed uuesti, teise tööriistaga või käsitsi märgendada. Siiski, olenevalt eesmärgist ja analüüsi tüübist ei pruugi esinduslikkus olla oluline, näiteks võib analüüsi eesmärgiks olla just vähemusgruppidel ennustustäpsust suurendada (nt kui enamusgrupi jaoks on juba hea mudel olemas). Paljudel juhtudel pole mudeli loomiseks kasutatavate andmete puhul oluline esinduslikkus[9], vaid lihtsalt andmete mitmekesisus, kõikide võimalike erijuhtude kaetus näidetega.

Esinduslikkus on oluline pigem testandmete puhul. Ennustava analüüsi puhul tuleb osa andmeid mudelite, lahenduste arendamise faasist kõrvale jätta. Neid andmeid kasutatakse loodud lahenduste hindamiseks. Kõrvale jäetud andmed on mudeli jaoks sama uued kui näited, mida mudel hakkab nägema mudeli reaalelus kasutusele võtmisel. Seega on mudeli suutlikkus neil andmetel hea mõõdik mudeli tegeliku kasulikkuse kohta. Seda muidugi juhul, kui need testandmed pole mingil viisil kallutatud. Soovituslik ongi kasutada esinduslikku andmestikku, mis pärineb võimalikult hiljutisest ajaperioodist (sest näiteks klientide käitumine võib ajas muutuda). **Andmete jagamine mudeli loomise ja hindamise eri faasides kasutatavateks hulkadeks** on andmete valimise üks alamülesandeid.

Olles valinud (ning loonud) meid huvitavad tunnused ja näited erinevates tabelites, tuleb need tabelid ühendada. Sama sisuga tabelid saab üksteise järele tõsta (ingl k *appending*), lisades ühe tabeli read teise tabeli lõppu. Näiteks saab erinevate aastate tabelid sel viisil ühendada. Kahe erineva sisuga tabeli ühendamiseks peab leiduma nende tabelite jaoks ühine identifikaator (võti). See näitab, millised read tabelis B

[9] Esinduslikkus: näidete jaotus peab olema sarnane jaotusega, mida lahendus kohtab juurutamisel.

sisaldavad infot tabeli A mingi rea kohta. On võimalik, et tabelis B esineb sama võtit mitu korda, näiteks on sama kliendi ID-ga seotud mitu lepingut. Eri allikatest tulenevate andmete puhul võib tabelite ühendamiseks sobiv veerg ka üldse puududa, kuid võib olla võimalik vastav veerg mingist kolmandast andmeallikast juurde panna. Seega pole erinevate tabelite veergudes sisalduva info ühendamine ühte laia tabelisse üldsegi lihtne, vaid vajab tähelepanelikkust, et tulemus oleks ikka see, mida soovitakse.

Andmete valimise sammude tulemusena võiks meil olla üks suur tabel, mis sisaldab eri andmeallikatest kokku toodud infot uuritavate objektide kohta. See info peaks olema relevantne, mitteinformatiivsed tunnused on eemaldatud. Tihti on mitmes tabelis sama sisuga veerud, millest tuleks alles jätta ainult üks. Suuremate kvaliteediprobleemidega tunnused on sellest tabelist välja jäetud, samuti probleemsed ja üleliigsed näited. Selle suure tabeli read jaotatakse tihti veel ka erinevateks hulkadeks (treeninghulk, testhulk).

4.2 Andmete puhastamine

Andmete puhastamine võib toimuda üksikutes andmetabelites või juba ühendatud suures andmetabelis. Mingis mõttes on kergem töötada väiksema tabeliga. Teatud kvaliteediprobleemidest ei pruugi meil aga olla ülevaadet enne, kui tabelid ühendame. Kui näiteks soovime kõrvale jätta andmepunktid, millel on liiga palju puuduvaid tunnuste väärtusi, peame selle nägemiseks esmalt tabelid ühendama.

Peamised kaks tunnuste väärtuste tasemel esinevat andmekvaliteedi probleemi on **puuduvad andmed** ja **erindid**.

4.2.1 Puuduvad väärtused

Puuduvaid väärtusi on lihtne tuvastada, kui tabeli väli on lihtsalt tühi või on sisestatud märksõna „NA“. Keerulisem on juhul, kui puuduvaid väärtusi pole korrektselt ja ühtlase stiiliga sisestatud – näiteks on võimatu vahet teha nullil, mis tähistab nullväärtust, ja nullil, mis peaks tähendama „teadmata“.

Levinuim meetod puuduvate väärtuste probleemi lahendamiseks on lihtsalt tabelist eemaldada kõik read, kus mõni väärtus puudub. Kuid nii võib paljude tunnuste peale kokku juhtuda, et kaotame liiga palju andmeid. Mida rohkem näiteid jääb alles, seda parem edasise analüüsi jaoks. Teine variant on eemaldada kõik tunnused (veerud), millel on puuduvaid väärtusi. Aga ka see võib olla liiga range kriteerium ja viia liiga paljude oluliste veergude eemaldamiseni. Võib leida kompromissi nende kahe vahel – eemaldada nii ridu kui ka veerge, nii et võimalikult suur osa tabelist ikkagi alles jääks. Väheste andmete puhul on puuduvate väärtuste tõttu andmepunktide või tunnuste eemaldamisel suur mõju analüüsi kvaliteedile.

Puuduvaid väärtusi võib olla võimalik taastada, tagantjärele leida. Näiteks võib olemas olla mingi teine andmetabel, kus vajalik info on olemas kõigi uuritavate objektide (andmeridade) kohta. Samuti võib olla puuduv info peidus mõne teise tunnuse väärtuses, näiteks võib olla arve numbrist tuletatav tehingu toimumise umbkaudne aeg, pood, milles tehing toimus, vms. Puuduvad kategoorilise tunnuse väärtused saab

asendada väärtusega „teadmata“, kuid see võib mudeli jaoks ülesande keeruliseks teha, sest väärtusele „teadmata“ on vaja erinevate näidete puhul väga erinevalt reageerida. Arvulise tunnuse puhul on võimalik puuduv väärtus asendada mediaani, moodi või keskmisega. Jällegi on tegu veidi riskantse teguviisiga, sest me sisestame tabelisse mittetõeseid väärtusi. Seega tasub seda teha ainult teatud analüüside ja mudelitüüpide puhul. Viimases hädas, kui andmeid on ridade/veergude eemaldamiseks liiga vähe ja „teadmata“ või keskmine väärtus ei ole piisav, võib proovida luua ennustava mudeli, mis puuduvad väärtused teadaolevate tunnuste alusel ennustab. Ka see ennustus pole tõene, kuid loodetavasti pakub täpsemaid vastuseid kui keskmine. Samas, kui tunnuse C väärtus on A ja B alusel ennustatav, siis ehk polegi veergu C analüüsi kaasata vaja?

Enim levinud on siiski puuduvate väärtustega andmepunktide eemaldamine või probleemsete tunnuste eemaldamine.

4.2.2 Erindid

Erindid on üldise andmete loogika ja andmejaotusega mitte kokku minevad näited või üksikud tunnuseväärtused, mis on tihti ekslikud väärtused, kuid vahel võivadki olla lihtsalt haruldased päriselulised juhud. Ekslikud väärtused ei teki ainult inimeste vigadest, ka automaatselt mõõtmisi teostavad sensorid võivad rikki minna ja tabelisse valesid väärtusi sisestada.

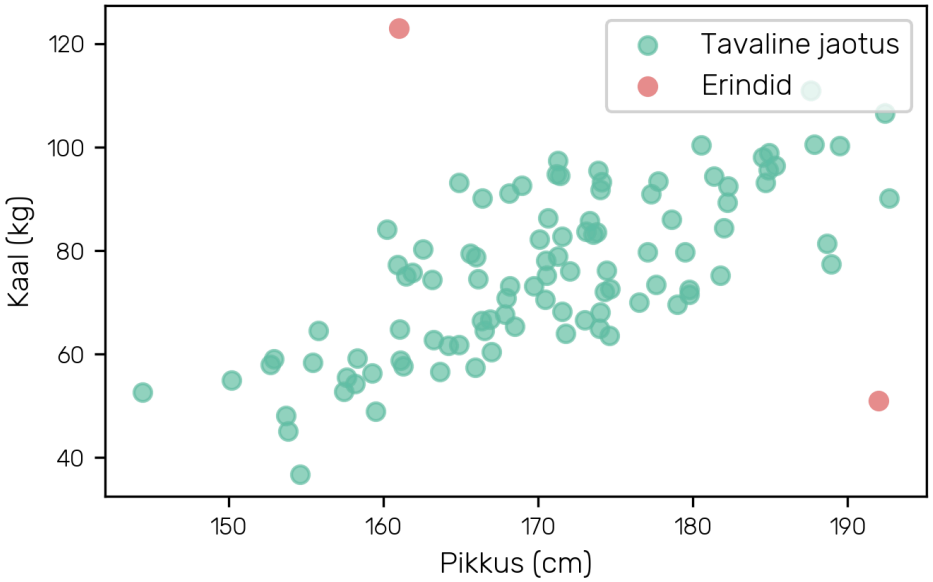
Erindi kohta võib öelda ka võõrväärtus või anomaalne väärtus. „Võõrväärtus“ ei ole tegelikult kuigi hea ja üldine termin, sest andmepunkti ei pruugi probleemseks teha ühe tunnuse väärtus, vaid mitme tunnuse väärtuste kombinatsioon. Ingliskeelne sõna *outlier* sisaldab veidi ruumilist ideed – andmepunkti, mis asub graafikul teistest eemal. Ka *erind* sisaldab mingi määral seda mõtet, nii et soovitame kasutada just seda terminit. Kui andmetabelite ühendamisel on midagi valesti läinud ja samale reale on kokku sattunud erinevate objektide tunnused, pole üksikud tunnuseväärtused erilised, kuid nende kombinatsioon on haruldane (joonis 4.1). Näitena oleme varem toonud pikkuste ja kaalude tabeli – 195 cm pikkus ja 50 kg kaal pole kumbki haruldased väärtused, kuid samal real esinevad need väga harva. Seega on tegu erindiga.

Siinkohal tuleb aga välja probleem erindite eemaldamisel – me ei tea, kas tegu on andmekvaliteedi probleemiga (viga sisestamisel või tabelite ühendamisel) või tegeliku haruldase näitega. Kui me liiga innukalt oma andmestikust erindeid eemaldame, kaotame ka tegelikult korrektseid andmeid ja seega olulist mitmekesisust, mis aitaks loodud mudelitel ka erijuhtudele üldistuda.

Valesti sisestatud või mõõdetud väärtuste tuvastamine on keeruline. Tõesti, kui keegi on pannud koma valesse kohta või unustanud nulli arvu lõppu panemata, siis võib tulemuseks olla niivõrd ekstreemne väärtus, et seda on võimalik tuvastada. Aga kui sisestaja vajutas kogemata klahvi „4“ asemel klahvi „5“, siis on tagantjärele viga tuvastada võimatu, sest tegu pole isegi erindiga. Andmestikku mingil määral võimalik „puhastada“, aga **andmestiku puhtus** on pigem peidetud, raskesti mõõdetav omadus. Kahtlaste väärtuste eemaldamine käib nii nagu puuduvate väärtuste puhulgi –

probleemsete ridade või veergude eemaldamise kaudu. Märgake, et ka nulliga märgitud puuduvad väärtused tuvastatakse tihti just erinditena.

Erindeid otsitakse tavaliselt andmetest tunnuse kaupa, eemaldades tavapärasest jaotusest kaugele jäävate väärtustega andmepunktid. Et avastada mitme tunnuse kombinatsioonina erilised näited, võib kasutada mõõtmelisuse vähendamise tehnikaid või anomaaliatuvastust. Nagu mainitud, tuleb erindite eemaldamisel hinnata, kas tegu on vigaste andmete või haruldaste juhtudega.



Joonis 4.1. Erindid. *Punased andmepunktid ei ole kummagi tunnuse väärtuse mõttes eraldiseisvalt väga haruldased, kuid kahe tunnuse väärtuste kombinatsioon on haruldane. Lähtekood.*

Erindite üheks allikaks on ühikute või mingi tunnuse arvutamise loogika muutus ajas või eri andmeallikate vahel. Sellisel juhul võib olla võimalik erineid parandada, mitte lihtsalt eemaldada. Näiteks võib olla kaal mingites tabelites mõõdetud grammides ja teistes kilogrammides. Tabeleid ühendades on vaja ühikud ühtlustada (oluline on teada iga tabeli iga veeru metainformatsiooni, ehk mida see veerg täpselt tähendab). Ka keeruliste, teiste tunnuste põhjal arvutatud tunnuste definitsioon võib ajas muutuda – näiteks võib taksoteenust vahendava firma hinnastamisega tegelev osakond otsustada teekonna keerukust uuel viisil hindama hakata. Neile võib tunduda, et see muutus mõjutab ainult nende tööprotsesse, kuid hiljem tellimuste tabeleid analüüsides tekib probleem – „Keerukus“ veerg ei ole erinevates ridades sama tähendusega, uued väärtused on varasematega võrreldes anomaalsed. Oluline on jällegi, et metaandmed sisaldaksid täpset ja uuendatud infot tunnuste taga peituvate valemite kohta, sel juhul on võimalik andmete kirjeldamise faasis väärtuste muutust ajas tuvastada, probleemi allikas kindlaks teha ning vanemad tabelid uue valemi järgi uuendada.

4.3 Andmete viimine vajalikule kujule

Viimase sammuna enne mudeldamiseni jõudmist on vaja viia andmed kujule, mida mudel eeldab. See tähendab teatud eeltöötlusi, mis on andmete mudeli jaoks mõistetavaks tegemiseks vajalikud.

- Kategoorilised väärtused nagu „Tallinn/Tartu/Pärnu/teadmata“ ei ole paljude mudelite jaoks mõistetavad, sest mudelid räägivad arvude keelt. „Tallinn“ ei ole arv, selle alusel ei saa ehitada arvutuste jada ehk masinõppemudelit. Seega viiakse kategoorilised muutujad tihti üks-mitmest-kujule (ingl k *one-hot*) ehk käsitletakse neid üksteist välistavate binaarsete tunnustena. Näiteks kui meie tabelis ongi variandid Tallinn, Tartu, Pärnu, teadmata, siis tekitame asukoha veeru asemel neli veergu, mis vastavad küsimusele „Kas asukoht on Tallinn?“ jne. Igal real saab korraga tõene olla ainult üks asukoht. Huvi korral saate kategooriliste andmete esitusest täpsemalt lugeda „Tehisintellekti algkursuse“ materjalide alapeatükist „Arvude keel“.

Kliendi_ID	Vanus	Linn = Tallinn	Linn = Tartu	Linn = Pärnu	Linn = teadmata
1	25	0	1	0	0
2	30	0	0	1	0
3	45	1	0	0	0
4	22	0	0	0	1

Tabel 4.1. Tunnus „linn“ üks-mitmest-kujul. *Ühest kategoorilisest tunnusest on saanud neli üksteist välistavat binaarset tunnust.*

- Teatud mudelid eeldavad, et kõik tunnused on standardjaotusega või vähemalt omavad väärtusi samas suurusjärgus. Näiteks klasterdamise puhul mõõdetakse andmepunktide vahelisi kaugusi ning tunnuse A väärtused 950 ja 1000 erinevad suhtelises mõttes vähem, kuid absoluutvea mõttes rohkem kui tunnuse B väärtused 1 ja 10. Mõõtes andmepunktide omavahelisi kaugusi, domineeriks tunnuse A väärtused liialt. Seega tunnused esmalt standarditakse ehk viiakse lõiku [0; 1] (näiteks min-max-skaleerimise abil).
- Mudelid, mis töötavad mitme sisendpildi (video) või aegridadega, võivad vajada andmestiku teatud kujule viimist. Kui iga andmepunkt koosneb tegelikult kümnest tunnusekomplektist erinevatel ajahetkedel, siis tekib küsimus, kas iga andmepunkt peaks olema eraldi tabel. Millisel viisil mudelit käivitav kood andmeid sisse eeldab saada, on väga lahenduse-spetsiifiline, see sõltub programmist, mida kasutatakse. Andmeteadlane peab olema kasutatavate tööriistade kasutusjuhendit lugenud ja viima andmestiku vajalikule kujule (sh näiteks kaustade korralduse ning failinimed).
- Kuupäevad võivad tekitada probleeme, sest mudel võib eeldada ameerikapärast MM/DD/YYYY formaati (kuu-päev-aasta). Nii treenitud mudel ei saa aga kindlasti aru eestikeelsetest väärtustest nagu „13. märts 2024“.
- Keerulisemate andmetüüpide, näiteks tekstide ja piltide, puhul võib vajalikule kujule viimine olla keerulisem. Näiteks võib olla vaja pildid õigele suurusele

skaleerida või nende heledust ühtlustada. Tekstide puhul võib mudelis olla kasutusel piiratud sõnavara ja sõnavarasse mitte mahtunud sõnad tuleb asendada UNK ehk teadmata sõnaga. Samuti tuleb tekst viia numbrite kujule, sest matemaatilised mudelid mõistavad ainult numbreid. Keele töötlemisest saab lähemalt lugeda peatükis 6.3.

See sobivale kujule viimise samm sõltub suuresti mudelist, mida kasutama hakatakse. Iga erineva mudelitüübi või teostatava analüüsi jaoks võib olla vajalik see samm eraldi läbi teha. Muidugi teevad mõned tööriistad neid samme automaatselt ise, näiteks võib paljudel juhtudel masinõppe mudelit loovale funktsioonile julgelt sisse anda tabeli kategooriliste väärtustega, sest mudeli jaoks näiteid ette valmistav koodijupp saab andmetüübist ise aru ja viib selle vajalikule kujule. Samal ajal tuvastaks see ka eestikeelsete kuunimedega kuupäevad kategooriliste tunnustena. On ka palju abistavaid funktsioone, mida võib kasutamiseks valida või mitte, näiteks andmete standardimise või tekstide eeltöötluse jaoks.

Koodinäide
Järgnevas koodivihikus näete andmete ettevalmistamise näidet päris andmetel, mis pärinevad suurelt apteegiketilt: link.

Enesekontrolli küsimused

1) Andmete kogumisele ja ettevalmistamisele kulub väga palju aega. Mis võib olla selle põhjus?

2) Tooge uusi näiteid (lisaks õpikus mainitule) andmete valimise kohta. Tooge näide juhust, kui me otsustame valida mitte kõik tabeli andmeread, kui me otsustame valida mitte kõik tabeli veerud, kui me otsustame kasutada ainult osa saadaval olevatest tabelitest.

3) Uute tunnuste loomine: millised kasulikke tunnuseid saab tuletada klienditeenindusse tehtud kõnede helisalvestuste põhjal? Milliseid teisi tööriistu saab selleks kasutada? Kas mingitel juhtudel võib olla otstarbekas kulutada aega ise kõnede läbi kuulamisele ja märkmete tegemisele?

4) Miks on andmete standardimine või skaleerimine oluline ja milliseid meetodeid selleks kasutatakse? Tooge näide mudelitüübist, mille puhul andmete mitteskaleerimine tekitaks probleeme.

5. Mudeldamine

Mudeli mõiste

EKI seletav sõnaraamat: mudel on originaalobjektiga kindlas vastavuses olev tehisobjekt, ka skeem, seoste matemaatiline kirjeldus vms.

Eesti entsüklopeedia: mudel on objekt, mis on kindlas vastavuses mingi teise objektiga (originaaliga) ja asendab seda tunnetusprotsessis. Mudeleid kasutatakse juhul, kui originaali otsene uurimine on raske või võimatu.

Vikipeedia: mudel on objekti struktuurselt sarnane esitus, analoog, mis asendab tunnetusprotsessis tavaliselt keerukamat objekti. Mudel võib olla abstraktne (näiteks matemaatiline mudel, joonis või sõnaline kirjeldus) või konkreetne ese.
Mudel võib olla hüpoteesi või teooria komponent, samuti võib see ise olla hüpotees või teooria, kui ta pretendeerib sellele, et vastab mõnele objektile reaalsuses. Mudel võib olla sõna, väide, mõttekonstruktsioon, asi, ese, teooria, hüpotees, formaliseeritud keel vms.

Enn Pärtel. „Füüsika mõisted gümnaasiumile“: mudel on keha või nähtuse koopia, milles on esile toodud antud uurimise seisukohast olulised omadused ja seosed ning välja on jäetud teisejärgulised omadused ja seosed.

Andmeteaduses on kuulus statistiku George Boxi tsitaat: „Kõik mudelid on valed, aga mõned on kasulikud.“ Mudelite ülaltoodud definitsioonidest selgub samuti kaks olulist omadust: mingi nähtuse lihtsustamine, mis teeb mudeli paratamatult mingil määral valeks, ja seejuures uurimisküsimuse jaoks oluliste omaduste säilitamine, mis teeb mudeli kasulikuks. Andmeteaduse kontekstis võib **mudelit** mõista kahel viisil.

1. Kirjeldava ja diagnostilise analüüsi ning hüpoteeside testimise kontekstis mõistame mudelina andmete kasulikku kirjeldust matemaatiliste valemite või jooniste abil. Näiteks andmete jaotuse kirjeldamine histogrammina pole täpne, sest punktide täpsed asukohad on asendatud vahemikuga. Seega oleme tegelikkusest loonud mingi lihtsustava mudeli, uskudes, et seda mudelit nähes on otsuste tegemiseks, teadmiste kätte saamiseks või muu eesmärgi jaoks oluline info alles. Näiteks ukse standardkõrguse määramiseks pole vaja teada kõikide inimeste pikkusi, piisab pikkuste jaotuse kasulikust kirjeldusest näiteks histogrammi või matemaatilise valemi (nt normaaljaotuse valemi) abil. Selle kasuliku kirjelduse põhjal saab järeldada ukse kõrguse, mille puhul 95% inimestest pead ära ei löö. Tihti on kirjeldava või diagnostilise analüüsi eesmärk mõõta või tõestada erinevate tunnuste omavaheline seos. Seoseid või nende puudumist ja aegridades esinevaid trende kirjeldatakse mingi valemiga ehk matemaatilise mudeliga. Mudelite eesmärk selles kontekstis on seletada olemasolevaid andmepunkte võimalikult hästi, järgides samal ajal Ockhami habemenoa printsiipi ja eelistades lihtsaid seletusi ehk lihtsaid mudeleid. Mingi muutuja jaotuse

mudeldamine võimaldab ka andmeid võrrelda, rakendades erinevaid statistilisi teste – hüpoteesid tõlgenduvad eeldatavateks andmejaotusteks, mudeliteks. Statistiliste testide rakendamine ja hüpoteeside testimine on seega CRISP-DM raamistiku kontekstis mudeldamine.

2. Ennustava analüüsi kontekstis mõistame mudelit kui mingit andmete tekke protsessi imiteerivat matemaatilist mudelit. Sellised mudelid väljastavad „ennustuse“ ehk proovivad genereerida võimalikult õige väärtusega märgendi. Päris maailmas eksisteerib mingi keeruline juhuslik protsess, millest kõik mõõdetud andmed tekivad ja mille toimimist ennustav mudel kuidagi kujutama peab, et täpseid ennustusi teha. Lihtsamad mudelid proovivad lihtsalt leida korrelatsioone ja seoseid ennustatava tunnuse ning muu teadaoleva info vahel. Keerulisemad mudelid võivad seesmiselt mingil viisil kujutada protsessi, mis nii ennustatava tunnuse kui ka teised teadaolevad tunnused tekitas. Näiteks võib homset temperatuuri ennustada viimase kolme päeva temperatuuri, tuulekiiruse ja sademete alusel, rõhudes mingite seaduspäralike trendide avastamisele nende arvude vahel, aga seda võib teha ka satelliidipiltide alusel kogu kontinendi ilmastikku mudeldades. Igal juhul on tegu mingi ebatäiusliku lihtsustatud vaatega ilma tegelikule tekkeprotsessile. Mudelite eesmärk selles kontekstis on teha korrektseid ennustusi veel nägemata andmepunktidel. Tihti pole oluline mudeli seesmist tööprotsessi mõista, tähtis on ainult selle täpsus, töökindlus. Seega Ockhami habemenoa printsiipi siin alati ei rakendata.

Andmete kirjeldamise peatükis me juba tutvusime kirjeldavate mudelitega, neid tol hetkel küll mudeliteks nimetamata. Selles peatükis tutvustame statistilise analüüsi meetodeid ja seejärel keskendume ennustavatele mudelitele, peamiselt masinõppele.

5.1 Statistiline analüüs

Viienda peatüki esimeses osas keskendume kolmele põhilisele statistilisele testile, mis on hädavajalikud andmetest tähendusrikka ülevaate saamiseks. Käesolev alapeatükk on mõeldud selleks, et sillutada tee teoreetilise statistika ja andmeteaduse praktiliste rakenduste vahel. Olgu tegemist kliendikäitumise analüüsimise, protsesside optimeerimise või tarkvaratoote uute funktsioonide testimisega, statistiline analüüs on andmepõhiste otsuste tegemise selgroog.

Paljudes rakendustes on tarvis teha vaatluste ehk kogutud andmete põhjal otsuseid kahe võimaliku hüpoteesi vahel, näiteks kas valmistatud detail vastab kvaliteedinõuetele või mitte. Vaatlusandmeid kasutatakse hüpoteesi kinnitamiseks või ümberlükkamiseks; näiteks võib küsida, kas tootmisest tulevad detailid on keskmiselt 10 mm pikad ning seeläbi vastavuses kvaliteedinõuetega. Sellistes olukordades saab otsuste tegemisel kasutada statistilisi teste, mis aitavad tuvastada olulisi erinevusi hüpoteeside vahel ja teha põhjendatud otsuseid kvaliteedi kohta, ehk kas mõõdetud väärtused vastavad kvaliteedinõuetele või viitavad kõrvalekalletele.

Igal testil on oma eesmärk, mis aitab mõista ja tõlgendada erinevat tüüpi andmeid ja suhteid. Statistilised testid võimaldavad hinnata tulemuste olulisust läbi statistilise olulisuse. Tulemused on statistiliselt olulised, kui need ei ole tekkinud juhuslike mõjutuste tulemusena, vaid nende taga on konkreetne põhjus. Kuigi statistiline olulisus on vajalik testi tulemuse tõsiselt võtmiseks, pole see praktikas alati piisav otsuste tegemiseks. Näiteks, kui mehed on nõus toote eest 5 senti rohkem maksma kui naised, ning see tulemus on statistiliselt oluline, siis pole see piisav toote hinnastamise muutmiseks, sest 5 senti on väga väike erinevus. Samamoodi pole mõistlik tooteid erinevalt hinnastada siis, kui uuringu järgi on mehed nõus 5 eurot rohkem maksma, kuid uuringus osalejate arv oli liiga väike (näiteks ainult 4 inimest), mistõttu pole tulemused statistiliselt olulised. Teisisõnu, erinevused võivad olla juhuslikud ja meil puuduvad piisavad numbrilised tõendid, et nullhüpotees kummutada.

Selles peatüki osas keskendume kolmele põhilisele statistilisele testile, mida iga andmeteadlane peaks valdama: t-test, χ^2-test (hii-ruut test) ja dispersioonanalüüs (ingl k *analysis of variance,* ANOVA).

Alustame **t-testist**, mis on hädavajalik kahe grupi keskmiste võrdlemiseks. See test on eriti kasulik, kui soovite hinnata sooritusnäitajaid enne ja pärast konkreetset muutust, näiteks analüüsida müügitulu enne ja pärast turunduskampaaniat, või võrrelda kahte erinevat kasutajaliidest. Käsitleme nii ühe valimi kui ka kahe valimi t-testi, varustades teid oskusega käsitleda sõltumatuid ja paaristatud valimeid.

Järgmisena uurime **hii-ruut test**i, mis on võimas vahend kategooriliste muutujate vaheliste suhete uurimiseks. Kui hindate, näiteks, turunduskampaania tõhusust erinevate demograafiliste gruppide seas või kasutajate eelistusi mitme tootekategooria puhul, aitab hii-ruut test kindlaks teha, kas täheldatud erinevused on statistiliselt olulised.

Statistilise analüüsi osa lõpetame dispersioonanalüüsi ehk **ANOVA**-ga, mis laiendab t-testi rohkem kui kahele grupile. ANOVA on hindamatu, kui on vaja testida väiteid, mis hõlmavad kolme või enamat gruppi. Näiteks olete arendanud mitu tarkvara versiooni, igaüks neist sisaldab uusi funktsioone. Kasutades ANOVA-t, saate võrrelda versioonide jõudlust, et teha kindlaks, millised uued funktsioonid töötavad kõige paremini. Või, näiteks, analüüsite erinevate tootegruppide puhul klientide rahulolu, kogudes ja võrreldes klientide tagasisidet iga tootegrupi kohta. Kasutades ANOVA-t, saate kindlaks teha, kas tootegruppide vahel on olemas statistiliselt oluline erinevus klientide tagasisides. See annab väärtuslikku teavet, millised tootegrupid vastavad kõige paremini klientide ootustele ja vajadustele, ning võimaldab ettevõttel teha informeeritud otsuseid tootearenduse ja turundusstrateegiate kohta.

Igas alajaotuses käsitleme mitte ainult teoreetilisi aspekte, vaid pakume ka praktilisi koodinäiteid ja juhtumiuuringuid. Need illustreerivad, kuidas rakendada teste reaalsetel andmetel ja kuidas tulemusi tõlgendada, et teha teadlikke otsuseid. Lisaks sisaldab iga jaotis praktilisi harjutusi ja näpunäiteid, mis aitavad teil neid statistilisi teste rakendada populaarsete andmeteaduse tööriistade ja tarkvaraga.

Selle alapeatüki õpiväljundina tekib teil kindel arusaam, millal ja kuidas rakendada neid põhilisi statistilisi teste oma andmeteadusprojektides. See teadmine tõhustab teie analüütilisi oskusi ja võimaldab teil läbi viia andmeanalüüse, mis võivad oluliselt mõjutada strateegiliste otsuste tegemist mistahes organisatsioonis.

Lisalugemist
Statistiline andmete analüüs on omaette ilus maailm. Statistiline andmete analüüs on omaette ilus maailm. Selles raamatus käsitleme seda pinnapealselt ja tutvustame teile kolme laialdasemalt kasutatavat testi. Statistiliste testide sügavamaks mõistmiseks soovitame raamatut Statistilise andmetöötluse algõpetus (Parring et al., 1997).

5.1.1 Sissejuhatus statistikaterminitesse

Enne kui sukeldume iga statistilise testi spetsiifikasse, on vajalik luua ühine arusaam põhilistest terminitest ja käsitusviisidest, mis on statistilise analüüsi aluseks. See mõistmine aitab statistilisi kontseptsioone tõhusalt rakendada erinevates andmeteaduse stsenaariumides.

Populatsioon on üldkogum, mida uuritakse ja mille kohta soovitakse järeldusi teha. Populatsioon hõlmab kõiki võimalikke andmepunkte (objekte või subjekte), mille kohta soovitakse saada informatsiooni, et lahendada püstitatud ülesannet. Näiteks võib populatsiooniks olla kõik ettevõtte töötajad, kõik kliendid või kõik toodetud detailid. Sageli on üldkogumi kõigi objektide või subjektide uurimine ajamahukas ning kallis, seetõttu on väga tavaline kasutada üldkogumist hästi valitud väiksemat alamhulka.

Valim on populatsioonist võetud alamhulk. Seda valikut kasutatakse kogu populatsiooni omaduste või parameetrite hindamiseks või järelduste tegemiseks. Andmeteaduses aitavad valimid testida hüpoteese ja mudeleid. Valimi moodustamisel on oluline tagada, et valim oleks esinduslik, et tulemused oleksid usaldusväärsed ja üldistatavad kogu populatsioonile.

Hüpoteeside testimine on meetod, mis kasutab valimi andmeid mingi populatsiooni omaduse kohta püstitatud statistiliste hüpoteeside kontrollimiseks. Peamine eesmärk on kindlaks teha, kas valimi andmetes on piisavalt tõendeid, et järeldada, kas teatud väide kehtib kogu populatsiooni jaoks.

Nullhüpotees (H_0) on tavaliselt kõige lihtsamat seletust pakkuv eeldus andmete kohta, näiteks, et uuritava teguri ja tulemuse vahel pole tuvastatavat seost. Seda eeldust peetakse tõeseks seni, kuni andmetest leitakse piisavalt alust selle ümberlükkamiseks. Näiteks t-testi korral tähendab nullhüpotees, et kahe rühma keskväärtused ei erine. Statistilise uuringu eesmärk on hinnata, kas saadud andmed toetavad nullhüpoteesi või annavad põhjuse see väide kummutada.

Sisukas hüpotees (H_a) on nullhüpoteesi alternatiiv, mis tüüpiliselt annab keerukama seletuse, näiteks et uuritava teguri ja tulemuse vahel on seos olemas. Sisukas hüpotees viitab sellele, et seose kohta täheldatud ilmingud andmetes ei ole juhuslikud ja on statistiliselt olulised. Näiteks t-testi korral võib olla sisukaks hüpoteesiks, et kahe rühma keskväärtused on erinevad, jättes täpsustamata, kui suur see erinevus on.

Teststatistik on arvutatud väärtus, mida kasutatakse statistiliste hüpoteeside testimisel, et otsustada, kas andmed annavad piisavalt tõendeid nullhüpoteesi kummutamiseks. Teststatistik koondab andmed üheks arvuks ning seda arvu võrreldakse ***kriitilise väärtusega.*** Kriitiline väärtus on piir, mille ületamisel loetakse tulemust statistiliselt oluliseks ja nullhüpotees kummutatakse ehk lükatakse tagasi; vastasel juhul jäädakse nullhüpoteesi juurde. Teststatistiku kriitiline väärtus sõltub valitud olulisuse nivoo väärtusest. Näiteks, kui nivooks on valitud tüüpiliselt 0.05, tähendab see, et 5% juhtudest võib teststatistiku väärtus ületada kriitilise väärtuse lihtsalt juhusliku varieeruvuse tõttu, isegi kui nullhüpotees on õige (seda tuntakse ka kui I liiki viga). Teststatistik järgib kindlat tõenäosusjaotust, näiteks normaaljaotust, t-jaotust, hii-ruut-jaotus, sõltuvalt kasutatavast statistilisest testist. See jaotus aitab määrata kriitilist väärtust ja tõlgendada, kui äärmuslik on täheldatud teststatistik nullhüpoteesi korral. Näiteks t-test kasutab t-jaotust keskmiste võrdlemiseks, samas kui hii-ruut-test kasutab hii-ruut-jaotust kategooriliste andmete analüüsiks. Teststatistiku ja selle jaotuse mõistmine on oluline, kuna see seob vaatlusandmed statistilise teooriaga, aidates hinnata, kas täheldatud seos on tõeline või juhuslik.

p-väärtus ehk olulisustõenäosus on hüpoteeside testimisel kasutatav mõõdik, mis aitab hinnata testi tulemuse statistilist olulisust. Mõõdiku väärtus on lõigus [0; 1] ja peegeldab seda, kui tõenäoline on vaatluse sedavõrd ekstreemne tulemus eeldusel, et nullhüpotees peab paika. See näitab, kui usutavalt võib saadud tulemus juhusest tuleneda: mida väiksem on *p*-väärtus, seda vähem tõenäoline on, et tulemus on tingitud juhusest. Seega, väike *p*-väärtus annab märku, et täheldatud efekt, erinevus või seos on tõenäoliselt tegelik. ***Olulisuse nivoo*** (tavaliselt 0,05) määratakse enne testi ja see toimib lävendina, mille alusel otsustatakse, kas *p*-väärtus on piisavalt väike, et lugeda tulemus statistiliselt oluliseks. Kui *p*-väärtus on väiksem kui olulisuse nivoo (nt $p < 0{,}05$), kummutame nullhüpoteesi.

Praktilises andmeanalüüsis peab aga meeles pidama, et statistiline olulisus pole alati ainuke oluline mõõdik, mille alusel teha lõplikke otsuseid andmete kohta. ***Efekti suuruse*** mõõdikuteks on nähtuse ulatuse kvantitatiivsed mõõdikud, mida kasutatakse statistilises testimises, et hinnata seose või erinevuse tugevust rühmade vahel. Siin mõistame efekti all nii põhjuslikke kui ka mittepõhjuslikke seoseid. Erinevalt *p*-väärtusest, mis näitab ainult efekti olemasolu, mõõdab efekti suurus, kui suur või tähenduslik see efekt (teisisõnu erinevus) on. Efekti suuruse ja *p*-väärtuse roll on komplementaarne. Oluline *p*-väärtus võib viidata ebaolulisele efektile ja suur efekt võib esineda isegi siis, kui *p*-väärtus ei ole statistiliselt oluline. Efekti suuruse arvutamine on oluline, sest see aitab mõista tulemuste praktilist tähtsust, mitte ainult statistilist olulisust. Efekti suuruse tõlgendamine on oluline leidude praktilise mõju mõistmiseks,

näiteks meditsiini- uuringutes, psühholoogias ja sotsiaalteadustes, kusjuures see tõlgendamine nõuab valdkonna-spetsiifilisi teadmisi ja konteksti arvestamist.

Illustreerime efekti suuruse tähtsust intuitiivselt näite kaudu. Oletame, et uurite uut ravimit vererõhu alandamiseks. Teil on kaks rühma: üks rühm võtab uut ravimit ja teine rühm platseebot. Paari kuu pärast on ravimirühma keskmine vererõhu langus 10 mmHg, platseeborühmas on see aga 1 mmHg. Oletame, et statistiline test näitab statistiliselt olulist vererõhu erinevust kahe rühma vahel, väga väikese *p*-väärtusega (näiteks 0,001). See tähendab, et puhtjuhuslikult oleks nii suurt erinevust väga väike tõenäosus kohata. Kuid *p*-väärtus ei ütle, kui suur on vererõhkude erinevus ning kui tähenduslik see vererõhu langetamise kontekstis on. Kui vererõhu langust 5 mmHg peetakse kliiniliselt oluliseks südamehaiguste riski vähendamisel, siis 9 mmHg langus näitab suurt efekti, mis tähendab, et uuel ravimil on oluline mõju. Vastupidi, kui uus ravim alandaks vererõhku ainult 2 mmHg (võrreldes platseeborühma 1 mmHg-ga), võib suure andmemahu korral *p*-väärtus siiski olla oluline, kuid efekt oleks väike ning seetõttu praktiliselt ebaoluline terviseriskide vähendamisel.

Seega aitab efekti suurus paremini mõista, kui märkimisväärne on tegelik kasu, mitte ainult seda, et erinevus on olemas.

Nende terminite mõistmine varustab teid vajalike teadmistega, et teostada hüpoteesi testimist ja tõlgendada tulemusi.

5.1.2 Kuidas valida ja läbi viia statistilist testi?

Statistilise testi valik ja läbiviimine hõlmab mitut sammu, alates andmete tüübi määramisest kuni tulemuste tõlgendamiseni *p*-väärtuse ja efekti suuruse põhjal.

Teste on statistikas väga palju, kuid laias laastus jagunevad need kaheks: parameetrilised ja mitteparameetrilised testid. Parameetrilised testid eeldavad, et andmed järgivad teatud jaotust (nt normaaljaotust), ja kasutavad arvutustes statistilisi parameetreid, nagu näiteks keskmine ja standardhälve. Näiteks t-test ja ANOVA on tüüpilised parameetrilised testid.

Igal parameetriliselt testil on oma eeldused ja kui need pole täidetud, siis ei saa testi tulemusi usaldusväärseteks pidada ja need võivad olla eksitavad. Sellisel juhul tuleb kasutada mitteparameetrilisi teste, mis ei eelda kindlat jaotust ega sõltu konkreetsetest parameetritest. Need testid põhinevad sagedustel või andmete järjestamisel, mistõttu on need kasulikud olukordades, kus andmed ei vasta parameetriliste testide nõuetele. Levinud mitteparameetrilised testid on näiteks Mann-Whitney U-test, Wilcoxoni test ja Kruskal-Wallise test. Selles raamatus me mitteparameetrilisi teste põhjalikumalt ei käsitle, kuid need võivad olla kasulikud, kui andmed ei vasta parameetriliste testide eeldustele. Lisainfot mitteparameetriliste testide kohta ja juhiseid nende rakendamiseks leiab paljudest statistika raamatutest (nt Parring *et al*., 1997) ning tarkvarade, näiteks R ja Pythoni teekidest (nt scipy.stats või statsmodels).

Allpool on algoritm, mis juhendab teid läbi selle protsessi.

Algoritm

Samm 1: Andmete tüübi ja omaduste kindlaksmääramine.

1. Tuvastage andmete tüüp:
    - nominaaltunnus või kategooriline tunnus (nt sugu, rühmad): kasutage näiteks hii-ruut testi;
    - pidev (nt pikkus, kaal): kaaluge näiteks t-testi või Wilcoxoni testi kasutamist.
2. Kontrollige andmete jaotust:
    - kui andmed on normaaljaotusega, siis teiste eelduste paikapidavusel võivad sobivad parameetrilised testid, näiteks t-test, ANOVA;
    - kui jaotustega seotud eeldused ei kehti, on parem kasutada mitteparameetrilisi teste, näiteks Mann-Whitney, Kruskal-Wallise või Wilcoxoni testi.
3. Kontrollige sõltuvuse olemust:
    - sõltumatud rühmad: rühmade vahel puudub otsene seos, st ühe rühma väärtused ei mõjuta teise rühma väärtusi. Näiteks, kui võrreldakse vererõhku meeste ja naiste vahel, on need sõltumatud rühmad, kuna ühe rühma tulemused ei sõltu teisest. Sellistel juhtudel tuleb kasutada sõltumatute rühmade jaoks sobivaid teste, näiteks kahe sõltumatu valimi t-testi;
    - sõltuvad rühmad: rühmade vahel on selge seos, st ühe rühma väärtused on seotud teise rühma väärtustega. Näiteks, kui mõõdetakse sama inimese vererõhku enne ja pärast ravi, on need mõõtmised omavahel seotud (paarisandmed). Sellistel juhtudel kasutatakse seotud andmetega teste, nagu paarisvõrdluse t-test, et arvestada rühmadevahelist seost.

Samm 2: Testi valik.

1. Üks arvuline tunnus (võrreldes mingi teadaoleva väärtusega):
    - ühe valimi t-test või mitteparameetriline alternatiiv.
2. Kaks või enam kategoorilist tunnust:
    - kasutage hii-ruut testi tunnuste vahelise seose olemasolu kontrollimiseks.
3. Võrdlus kahe rühma vahel (üks pidev ja üks binaarne tunnus):
    - sõltumatud rühmad (nt meeste ja naiste keskmise vererõhu võrdlus): kasuta kahe valimi t-testi normaaljaotuse korral; kui normaaljaotuse eeldus ei kehti, kasuta mitteparameetrilist Mann-Whitney testi;
    - sõltuvad rühmad (nt sama inimese vererõhk enne ja pärast ravi): kasuta seotud valimite t-testi; kui t-testi eeldused pole täidetud, vali Wilcoxoni test.

4. Üks pidev ja üks kategooriline tunnus (võrdlus rohkem kui kahe rühma vahel):
    - Kasuta ANOVA (parameetriline) või Kruskal-Wallis testi (mitteparameetriline).

Samm 3: Testi läbiviimine ja *p*-väärtuse arvutamine.

1. Arvutage teststatistik.
2. Leidke *p*-väärtus. Kui p-väärtus on väiksem kui valitud olulisuse nivoo (nt 0,05), siis on tulemus statistiliselt oluline ja nullhüpotees kummutatakse.

Samm 4: Efekti suuruse arvutamine.

1. Arvutage efekti suurus, et hinnata tulemuse praktilist tähtsust. Efekti suurus aitab hinnata, kas erinevus või seos on ka praktiliselt oluline, mitte ainult statistiliselt.

Samm 5: Tulemuste tõlgendamine.

1. Tulemuse statistiline olulisus:
    - kui *p*-väärtus < olulisuse nivoo, on tulemus statistiliselt oluline, st kummutatakse nullhüpotees.
    - Kui *p*-väärtus > olulisuse nivoo, siis meil pole piisavalt tõendusmaterjali, et kummutada nullhüpoteesi ja me jääme nullhüpoteesi juurde.
2. Tulemuse praktiline olulisus:
    - hinnake efekti suurust – suurem efekt näitab suuremat praktilist tähtsust.

See algoritm aitab valida õiget testi, arvestada *p*-väärtust ja efekti suurust ning tõlgendada tulemusi nii statistilisest kui ka praktilisest vaatenurgast.

5.1.3 T-test

T-test on statistiline test, mida kasutatakse kahe rühma keskväärtuste võrdlemiseks või ühe valimi keskväärtuse võrdlemiseks teadaoleva eeldatava väärtusega. T-test on oluline analüüsimeetod paljudes valdkondades, nagu turundus, finants, meditsiin jpt.

T-testi rakendamine

Alustame praktilise näitega, et kohe intuitiivselt illustreerida, kuidas t-test töötab ja kuidas seda kasutada. Praktilise tulemuse abil saab parema ettekujutuse testi rakendamisest ja tulemuste tõlgendamisest. Seejärel selgitame olulisemaid teoreetilisi üksikasju, et mõista sügavamalt, milliseid eeldusi t-test nõuab, kuidas see arvutatakse ja kuidas tulemusi õigesti tõlgendada.

Selles praktilises näites kasutame t-testi, et võrrelda kahe erineva turunduskampaania tulemusi. Ettevõte viis läbi kaks turunduskampaaniat, kasutades traditsioonilist reklaami (kampaania A) ja digitaalset reklaami (kampaania B). T-testi abil saame statistiliselt

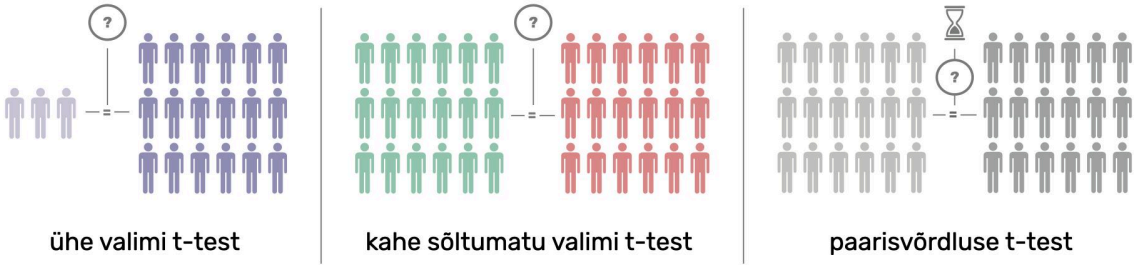
hinnata, kas kampaaniate tulemuste vahel on märkimisväärne erinevus. See analüüs aitab ettevõttel teha teadliku otsuse, millist turundusstrateegiat tulevikus eelistada.

Praktiline näide
Klõpsake sellel lingil, et avada Google Colaboratory vihik ja alustada praktilise tööga: Google Colaboratory vihik.

T-test põhineb t-statistikul, mille väärtus arvutatakse kasutades rühmade ehk teisisõnu valimite keskväärtust, standardhälvet ja suurust. See teststatistik mõõdab, kui hästi andmed toetavad või lükkavad ümber nullhüpoteesi, mille järgi rühmadel erinevusi pole. Kui t-statistik on arvutatud, võrreldakse seda t-jaotuse kriitilise väärtusega, et hinnata kui tõenoline on selline või veel ekstreemsem tulemus juhusliku varieeruvuse tõttu, eeldades, et nullhüpotees on tõene. T-testi sooritamiseks valitakse sobiv testi tüüp (nt ühe valimi t-test, kahe valimi t-test või seotud valimite t-test), lähtudes uuringu ülesehitusest, ja määratakse olulisuse nivoo (tavaliselt 0,05).

Millist t-testi kasutada?

T-teste on kolm põhitüüpi, igaüks neist täidab erinevat eesmärki (joonis 5.1). T-teste saab valida vastavalt uuritavale hüpoteesile ja andmetele. Allpool kirjeldame, kuidas võiks sõnastada need hüpoteesid iga t-testi tüübi jaoks ja kuidas arvutada t-statistikut.



***Joonis 5.1. T-testi kolm põhitüüpi**. Vasakpoolne joonise osa kujutab ühe valimi t-testi, mille eesmärk on võrrelda ühe valimi keskmist teadaoleva või hüpoteetilise grupi (populatsiooni) keskmisega. Joonise keskosa näitab kahe sõltumatu valimi t-testi, mille eesmärk on võrrelda kahe sõltumatu grupi keskmisi, et teha kindlaks, kas need erinevad oluliselt. Parempoolne joonise osa illustreerib paarisvõrdluse t-testi, mille eesmärk on võrrelda kahe seotud rühma keskmisi (nt sama grupi andmeid enne ja pärast ravimi kasutamist), et näha, kas esineb märkimisväärne muutus.*

T-testi tüübi valimisel peab arvestama kahe aspektiga: kas võrreldavad rühmad pärinevad ühest vaatluste populatsioonist või kahest eri populatsioonist ja kas soovite

testida erinevust kindlas suunas. Testiga on võimalik kahe grupi vahel mõõta järgmist efekti:

- erinevust teoreetilisest keskmisest;
- gruppide ooteväärtuste erinevust;
- enne ja pärast tehtud mõõtmiste süstemaatilisest erinevust.

Illustreerime iga t-testi tüübi rakendamist näite abil.

Ühe valimi t-testi kasutatakse kontrollimiseks, kas valimi põhjal erineb ühe rühma keskmine μ oluliselt mõnest teadaolevast või standardseks peetavast võrdlusväärtusest μ_0. Kuna tegelik keskmine μ on tundmatu, kasutatakse selle ligikaudseks hinnanguks valimi keskmist $\bar{x}$.

Näiteks, farmaatsiaettevõte on välja töötanud uue ravimi, mis on mõeldud vererõhu alandamiseks. Enne ulatuslikuma uuringu alustamist soovivad nad tagada, et ravim vähendab vererõhku oluliselt, võrreldes normaalse või standardse väärtusega (nt 120 mmHg). Antud olukorras saab kasutada ühe valimi t-testi pärast ravimi võtmist patsientide keskmise vererõhu võrdlemiseks väärtusega μ_0= 120 mmHg. See test aitab kindlaks teha, kas ravimil on statistiliselt oluline mõju vererõhu alandamisele.

Matemaatiliselt saab ühe valimi t-testi kirjeldada järgmiselt:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}},$$

kus

- $\bar{x}$ on valimi keskmine,
- μ_0 on võrdlusväärtus,
- *s* on valimi standardhälve,
- *n* on valimi suurus.

Ühe valimi t-testi hüpoteesid

- Nullhüpotees (H_0): rühma tegelik keskväärtus on võrdne teadaoleva võrdlusväärtusega. Matemaatiliselt võib seda väljendada järgmiselt:
  H_0: $\mu = \mu_0$, kus μ on rühma tegelik keskväärtus ja μ_0 on võrdlusväärtus.

- Sisukas hüpotees (H_1): rühma tegelik keskväärtus ei ole võrdne teadaoleva võrdlusväärtusega. See võib olla kahepoolne või ühepoolne, kui suund on määratud:
  H_1: $\mu \neq \mu_0$
  või ühepoolse testi puhul
  H_1: $\mu > \mu_0$ või H_1: $\mu < \mu_0$.

Kahe sõltumatu valimi t-testi kasutatakse kahe sõltumatu rühma keskväärtuste võrdlemiseks. Näiteks, ülikool soovib võrrelda suvise ettevalmistusprogrammi läbinud ja

mitte läbinud üliõpilaste akadeemilisi tulemusi. Eeldatakse, et ettevalmistusprogramm aitab üliõpilastel akadeemiliselt paremini toime tulla.

Kahe sõltumatu valimi t-test sobib hästi selle võrdluse jaoks. Test võrdleb suveprogrammi läbinud üliõpilaste rühma ja mitte läbinud üliõpilaste keskmisi akadeemilisi hindeid. See aitab hinnata programmi efektiivsust õpilaste akadeemiliste tulemuste parandamisel.

Matemaatiliselt saab kahe sõltumatu valimi t-testi kirja panna järgmiselt:

$$t = \frac{\bar{x}_2 - \bar{x}_1}{s_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

kus

- $\bar{x}_1$ ja $\bar{x}_2$ on vastavalt esimese ja teise rühma valimi keskmised,
- s_p on kahe valimi ühine standardhälve,
- n_1 ja n_2 on valimite suurused.

Kahe sõltumatu valimi t-testi hüpoteesid

- Nullhüpotees (H_0): kahe rühma keskväärtused on võrdsed.
  H_0: $\mu_1 = \mu_2$, kus μ_1 ja μ_2 on vastavalt esimese ja teise rühma keskväärtused.
- Sisukas hüpotees (H_1): kahe rühma keskväärtused ei ole võrdsed. Seda saab samuti sõnastada kahepoolse või ühepoolse testina:
  H_1: $\mu_1 \neq \mu_2$
  või ühepoolse testi puhul
  H_1: $\mu_1 > \mu_2$ või H_1: $\mu_1 < \mu_2$.

Paarisvõrdluse t-testi (sõltuvate vaatluste t-testi) kasutatakse sama rühma keskväärtuste võrdlemiseks eri aegadel, näiteks enne ja pärast ravi. Siinkohal on oluline märgata, et kahe rühma vaatlused pole üksteisest sõltumatud, vaid on üksteisega seotud, sest käivad samade objektide kohta. Näiteks, dietoloog hindab uue dieedikava efektiivsust kehakaalu vähendamisel. Ta kogub osalejate kaalu enne dieedi alustamist ja uuesti kolm kuud pärast dieedi järgimise algust.

Paarisvõrdluse t-test sobib selle ülesande lahendamiseks, kuna samu isikuid mõõdetakse enne ja pärast sekkumist, mis võimaldab võrrelda nende kaalu kaht eri mõõtmiskorda. See test kontrollib, kas dieediplaani ja osalejate kaalu vahel on seos selle perioodi jooksul.

Matemaatiliselt saab kahe sõltumatu valimi t-testi kirjeldada järgmiselt:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}},$$

kus

- $\bar{d}$ on paarisvõrdluse vaatluste erinevuste keskmine,
- s_d on erinevuste standardhälve.

Paarisvõrdluse t-testi hüpoteesid

- Nullhüpotees (H_0): paarisvõrdluse vaatluste keskväärtuste erinevus on null.

  H_0: $\mu_d = 0$, kus μ_d on paarisvõrdluse vaatluste keskväärtuste erinevus.
- Sisukas hüpotees (H_1): paarisvõrdluse vaatluste keskväärtuste erinevus ei ole null. See hüpotees käsitleb tavaliselt seda, kas sekkumine on seotud tulemusega:

  H_1: $\mu_d \neq 0$

  või see võib olla suunatud, kui ootus on konkreetne:

  H_1: $\mu_d > 0$ (näitab suurenemist) või H_1: $\mu_d < 0$ (näitab vähenemist).

T-testi eeldused

T-test on parameetriline test, mis tähendab, et testi tulemus on usaldusväärne, kui kehtivad testi eeldused. T-testi puhul on need eeldused järgmised:

1. **Vaatluste sõltumatus:**
   Iga andmepunkt peab olema sõltumatu ja mitte mõjutama teisi andmepunkte. Sõltumatute valimite korral kehtib see iga grupi sees, ja paarsusandmete puhul kehtib see iga paari kohta. Näiteks, kui uuritakse meeste ja naiste keskmist vererõhku, peab iga osaleja mõõtmine olema sõltumatu. Teisisõnu ei tohiks ühe osaleja vererõhumõõtmise tulemus sõltuda teise osaleja terviseseisundist ega mõõtmiste ajastamisest. Paarisvaatluste korral, kui mõõdetakse inimese vererõhku enne ja pärast uue ravimi manustamist, on iga paari mõõtmised seotud, kuna need pärinevad samalt inimeselt, kuid need mõõtmised ei tohi mõjutada teise paari tulemusi.
2. **Andmete normaaljaotus:**
   Andmed igas grupis või paaride erinevused peavad järgima normaaljaotust. See eeldus on eriti oluline väikeste valimite puhul, kuna kõrvalekalded normaaljaotusest võivad testi täpsust vähendada. Selle eelduse kontrollimiseks praktikas andmete mõistmise etapis võib kasutada jaotust visualiseerivaid graafikuid (näiteks histogramme) või statistilisi teste (näiteks Shapiro-Wilki või Kolmogorov-Smirnovi test).
3. **Ekstreemsete väärtuste puudumine:**
   Andmestikus ei tohiks esineda olulisi ekstreemseid väärtusi või teisisõnu erindeid (ingl k *outliers*), mis võivad tulemusi tugevalt mõjutada. Kui sellised väärtused on olemas, võib testi tulemuste tõlgendamine olla ebausaldusväärne.
4. **Hajuvuse homogeensus ehk homoskedastilisus** (sõltumatute valimite t-testi puhul):
   Kahte eraldi gruppi võrreldes eeldatakse, et rühmade hajuvus ehk standardhälve on sarnane. Kui rühmade hajuvused on erinevad, tekib heteroskedastilisus, mis võib viia ebatäpsete tulemusteni, kuna t-test eeldab võrdseid hajuvusi. Kui heteroskedastilisus on märkimisväärne, ei pruugi t-testi tulemused olla usaldusväärsed, ja tuleks kaaluda Welch'i t-testi kasutamist, kuna see ei eelda hajuvuste homogeensust.

See tähendab, et enne testi rakendamist peate veenduma, et need tingimused on täidetud. Juhul, kui eeldused pole täidetud, tuleb teha mitteparameetrilisi teste.

Enamik andmeteaduses kasutatavaid tarkvarasid (Python, R jne) sisaldavad t-testi funktsiooni. Funktsioon võtab ettevalmistatud andmed sisse ja arvutab t-statistiku. Seejärel võrreldakse seda kriitilise väärtusega ja arvutatakse *p*-väärtus. T-statistiku absoluutväärtus on suur, kui rühmade keskväärtuste erinevus on suur. Kui aga soovite hinnata t-testi tulemust ise käsitsi, võite oma arvutatud t-statistikut võrrelda kriitiliste väärtuste tabeli väärtustega (nt Parring *et al.*, 1997), et määrata, kas teie t-statistik on suurem kui juhuslikult eeldatav. Kui see vastab tõele, võite nullhüpoteesi kummutada ja järeldada, et kaks rühma on statistiliselt erinevad.

Praktikas testi tulemuste hindamiseks kasutatakse tihti *p*-väärtust ja efekti suurust. Hinnatakse, kas arvutatud *p*-väärtus on väiksem või võrdne 0,05 kriitilise nivoo väärtusega ja kas efekt on ülesande kontekstis piisavalt suur. Tulemus $p \leq 0,05$ viitab oluliste gruppide erinevusele ja tähendab, et nullhüpoteesi saab kummutada.

T-testi puhul on võimalik hinnata efekti suurust näiteks arvutades kahe rühma keskväärtuste $\bar{x}_1$ ja $\bar{x}_2$ erinevust, ehk $\bar{x}_2$ - $\bar{x}_1$. Kuna aga andmed võivad olla mõõdetud erinevates ühikutes või erinevatel skaaladel, siis on tulemuste võrdlemine raskendatud. Efekti suuruse mõõdik Coheni *d* võimaldab standardiseerida keskväärtuste erinevuse $\bar{x}_2$ - $\bar{x}_1$, arvestades andmete varieeruvust, ehk mõõta gruppide keskmiste omavahelist erinevust standardhälbe ühikutes. See muudab efekti suuruse tõlgendamise lihtsamaks ja võrreldavamaks.
Coheni *d* arvutamiseks kasutatakse valemit

$$d = \frac{\bar{x}_2 - \bar{x}_1}{s_p},$$

kus $\bar{x}_1$ ja $\bar{x}_2$ on vastavalt esimese ja teise rühma valimite keskmised ning s_p on kahe valimi ühine standardhälve.

Coheni *d* väärtuse tulemuse tõlgendamine võib kokkuvõtlikult olla järgmine:

- väike efekt : $d \approx 0,20$
- keskmine efekt : $d \approx 0,50$
- suur efekt : $d \approx 0,80$ või suurem

5.1.4 Hii-ruut test

Hii-ruut test, sageli esitatud kui χ^2, on statistiline meetod, mida kasutatakse selleks, et hinnata, kas kahe kategooriliste väärtustega tunnuse vahel on seos. See on eriti kasulik uurimisvaldkondades nagu tervishoid, turundus ja sotsiaalteadused, kus kategooriliste muutujate vaheliste suhete mõistmine on oluline.

Testi hüpoteesid võib sõnastada järgmiselt:

- nullhüpotees (H_0): tunnuste vahel puudub seos;

- sisukas hüpotees (H_1): tunnuste vahel on seos.

Hii-ruut test põhineb hii-ruut-statistikul, mille väärtus arvutatakse välja, kasutades vaatluste sagedusi. Test võrdleb sündmuste vaadeldud sagedusi nende teoreetiliste sagedustega, mida me ootaksime nullhüpoteesi kehtides. Selleks, et illustreerida hii-ruut-statistiku arvutamist, kasutame sagedustabelit (tabel 5.1)

Esimese muutuja tasemed

Teise muutuja tasemed

	Kategooria 1	Kategooria 2	…	Kategooria n	Kokku
Rühm 1	O_{11}	O_{12}	…	O_{1n}	Rea väärtuste summa $O_{1\cdot}$
Rühm 2	O_{21}	O_{22}	…	O_{2n}	Rea väärtuste summa $O_{2\cdot}$
…	…	…	…	…	…
Rühm k	O_{k1}	O_{k2}	…	O_{kn}	Rea väärtuste summa $O_{k\cdot}$
Summaarne väärtus	Veeru väärtuste summa $O_{\cdot 1}$	Veeru väärtuste summa $O_{\cdot 2}$	…	Veeru väärtuste summa $O_{\cdot n}$	Kogusumma O

Tabel 5.1. Sagedustabel. *Sagedustabel esitab andmeid kahe kategoorilise muutuja kohta. Iga veerg esindab ühe kategoorilise muutuja väärtusi, näiteks kategooria$_1$, kategooria$_2$, … , kategooria$_n$. Sagedustabeli iga rida esindab teise muutuja väärtusi, näiteks rühm$_1$, rühm$_2$, … ,rühm$_k$. Tabeli lahtrid näitavad vaadeldud sagedust, mida tähistatakse O_{ij}, ehk mitu korda iga ridade ja veergude kombinatsioon andmestikus esineb. Rea summaarne väärtus on ühe rea vaadeldud sageduste summa. Veeru summaarne väärtus on vaadeldud sageduste summa ühes veerus. Kogusumma $O_{\cdot}$ on tabelis esitatud vaadeldud sageduste kogusumma.*

Hii-ruut-statistik arvutatakse kasutades valemit:

$$\chi^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

kus O_{ij} on vaadeldud sagedus ja E_{ij} on oodatud sagedus nullhüpoteesi korral. Kuna nullhüpoteesi järgi on need kaks tunnust sõltumatud ehk ilma omavahelise seoseta, siis saab oodatava sageduse arvutada järgmiselt:

$$E_{ij} = \frac{O_{i\cdot} \cdot O_{\cdot j}}{O},$$

kus $O_{i\cdot}$ on rea *i* summaarne väärtus, $O_{\cdot j}$ on veeru *j* summaarne väärtus ning *O* on sageduste kogusumma.

Kuigi χ^2-test on laialdaselt kasutatav, ei ole see siiski täiesti eeldustevaba. Üks oluline eeldus on, et kõik oodatavad sagedused peavad olema suuremad kui 5. Kui see eeldus ei ole täidetud, võivad testi tulemused muutuda ebausaldusväärseks, kuna väiksed sagedused võivad moonutada χ^2-testi statistika ja *p*-väärtuse arvutust. Sellises olukorras tuleks kaaluda alternatiivseid meetodeid, nagu näiteks:

- Fisheri täpne test: sobib väikeste sageduste korral ja annab täpse p-väärtuse, kuid on väga arvutusmahukas suurte andmete korral.
- Andmete ümberkategooriseerimine: kategooriate ühendamine, et suurendada sagedusi.

Hindamaks, kas testi tulemus on statistiliselt oluline, kasutatakse hii-ruudu jaotuse tabelit, mille statistikud on juba varem koostanud (tabeliga saab tutvuda nt Parring *et al*., 1997), et leida kriitiline χ^2 väärtus soovitud olulisuse nivool. Analüüsi jooksul arvutatud χ^2 väärtust võrreldakse kriitilise väärtusega, mis leitakse jaotuse tabelist arvestades vabadusastmete arvu *k*-1, kus *k* on ridade ja veergude arvu miinimum (nt 3 × 4 tabelis on kolm rida ja neli veergu ning seega *k* = 3). Kui arvutatud χ^2 statistik on suurem kui kriitiline väärtus, kummutatakse nullhüpotees ja võetakse vastu sisukas hüpotees. Vastasel juhul püsib nullhüpotees.

Praktikas testi tulemuste hindamiseks kasutatakse tihti *p*-väärtust ning võrreldakse seda eelnevalt valitud olulisuse nivooga. Tulemus $p \leq 0,05$ viitab, et kategooriliste tunnuste vahel on statistiliselt oluline seos.

Kui hii-ruut testi *p*-väärtus näitab olulist seost, siis Craméri *V* väärtus aitab määrata, kui tugev see seos on.

Craméri *V* arvutamiseks kasutatakse valemit

$$V = \sqrt{\frac{\chi^2}{N(k-1)}},$$

kus:

- χ^2 on hii-ruut testi tulemus,
- *N* on valimi suurus ehk kõikide risttabelisse kantud vaatluste koguarv (tabelis 5.1 tähistatud kui O),
- *k* on ridade ja veergude arvu miinimum, samuti nagu hii-ruut-testi vabadusastmete arvutamisel.

Craméri *V* väärtusi tõlgendatakse järgmiselt:

- nõrk seos: *V* ≈ 0,1;
- keskmine seos: *V* ≈ 0,3;
- tugev seos: *V* ≈ 0,5 või suurem.

Seega aitab Craméri *V* mõista, kas leitud seos on nõrk, mõõdukas või tugev ning kas see on praktiliselt oluline lisaks statistilisele olulisusele.

Hii-ruut testi rakendamise praktiline näide

See näide illustreerib, kuidas ettevõtted saavad kasutada hii-ruut-testi, et teha teadlikke otsuseid turundusstrateegiate kohta, lähtudes demograafilistest eelistustest.

Oletame, et üks jaekaubandusettevõte on hiljuti turule toonud uue keskkonnasõbralike rõivaste tooteliini. Turundusmeeskond soovib uurida, kas tarbijate eelistus sellele uue liini suhtes on seotud vanusrühmaga. Selleks viisid nad läbi küsitlusuuringu klientide seas, kes külastasid poodi viimase kuu jooksul. Küsitluses paluti klientidel märkida, kas nad "Eelistavad keskkonnasõbralikku liini" või "Ei eelista keskkonnasõbralikku liini" ning küsiti ka nende vanust. Kliendid jaotati vastuste alusel kolme vanuserühma: "Alla 30", "30–50" ja "Üle 50". Kogutud andmed koondati sagedustabelisse 5.2, kus iga rida näitab vastuste arvu ("Eelistab" ja "Ei eelista") vastavalt iga vanuserühma lõikes. Tabelis esitatud andmed on vaadeldud sagedused, mille abil saab analüüsida vanuserühmade ja keskkonnasõbraliku rõivaste eelistuse vahelist võimalikku seost.

Vanuserühm	Eelistab keskkonnasõbralikku	Ei eelista keskkonnasõbralikku	Kokku
Alla 30	90	110	200
30–50	140	160	300
Üle 50	70	130	200
Kokku	300	400	700

Tabel 5.2. Näidisandmete sagedustabel.

Arvestades, et käsitsi arvutamine on aeganõudev ja võib tekkida näpuvigu, on tõhusam ja täpsem kasutada arvutuslikke meetodeid. Seepärast demonstreerime järgmistes osades arvutuslikku lähenemist selle probleemi lahendamisele, kasutades Pythoni keelt ja Google Colaboratory keskkonda.

Praktiline näide
Klõpsake siin, et avada Google Colaboratory vihik, kus saate näha ja käivitada koodi, mis viib läbi testimist koos selgitustega.

5.1.5 Dispersioonanalüüs

Dispersioonanalüüs (ingl k *analysis of variance*, ANOVA) on statistiline meetod, mida kasutatakse enam kui kahe üldkogumi ehk rühma keskväärtuste võrdlemiseks. Selle abil saame analüüsida näiteks erinevate raviviiside või toodete seost ühe või mitme

tunnusega ja teha järeldusi, kas vaatlustulemused erinevad juhuslikult või on seos tegelikult olemas.

ANOVA rakendamine praktikas

Alustame praktilise näitega, et anda esmane ülevaade ANOVA kasutusest ja rakendamisvõimalustest. Näite kaudu vaatleme, kuidas ANOVA abil saab hinnata erinevate rühmade keskmiste erinevusi. Pärast praktilise tulemuse mõistmist selgitame teoreetilist tausta põhjalikumalt, et saada aru, kuidas ANOVA meetod toimib, millised on eeldused ning kuidas tulemusi tõlgendada.

Näites kasutame ANOVA meetodit, et analüüsida kolme ravimi – ravim A, ravim B ja ravim C – mõju vererõhu langusele. Meie eesmärk on võrrelda rühmade keskväärtusi, et teha kindlaks, kas nende vahel on statistiliselt olulisi erinevusi. Kõrge vererõhk ehk hüpertensioon on tõsine terviseprobleem, mis suurendab südameinfarkti, insuldi ning südame- ja neerupuudulikkuse riski, mistõttu on tõhusate ravimeetodite hindamine ülitähtis. ANOVA meetodi abil saame võrrelda nende ravimite keskmist mõju vererõhule ning teha kindlaks, kas mõni neist on teistest oluliselt tõhusam, pakkudes seeläbi väärtuslikku teavet paremate ravimeetodite leidmiseks.

Praktiline näide
Klõpsake lingil, et avada Google Colaboratory vihik koos täieliku koodiga, mis viib läbi ANOVA analüüsi koos üksikasjalike selgitustega. See juhib teid samm-sammult läbi ANOVA rakendamise protsessi, sealhulgas andmete ettevalmistamise, mudeli koostamise ja tulemuste tõlgendamise.

ANOVA peamine eesmärk on testida nullhüpoteesi, mis väidab, et kõigi rühmade keskväärtused on võrdsed. Sisukas hüpotees väidab, et vähemalt ühe rühma keskväärtus erineb teistest.

Rühmade keskväärtuste võrdlemisel saab esitada kontrollimiseks järgmised hüpoteeside paarid:

- nullhüpotees (H_0): rühmade keskväärtused on võrdsed:
  H_0: $\mu_1 = \mu_2 = \ldots = \mu_k$, kus μ_1 , μ_2 , … , μ_k on esimese, teise, ..., *k-nda* rühma keskväärtused;
- sisukas hüpotees (H_1): on olemas vähemalt üks rühm, mille keskväärtus erineb teiste rühmade keskväärtustest:
  H_1: rühmade keskväärtuste seas leidub vähemalt üks paar erinevaid keskväärtusi ehk $\mu_i \neq \mu_j$ kus $i \neq j$.

ANOVA jaotab kogutud variatsiooni kaheks komponendiks:

- rühmadevaheline dispersioon (ingl k *between-group variance*) ehk variatsioon, mis on põhjustatud rühmade erinevusest;
- rühmasisene dispersioon (ingl k *within-group variance*) ehk variatsioon, mis on tingitud juhuslikest erinevustest rühma sees.

Rühmade erinevuse hindamiseks arvutab ANOVA *F*-statistiku, mis on rühmadevahelise dispersiooni ja rühmasisese dispersiooni suhe. Suur *F*-statistiku väärtus viitab sellele, et rühmadevahelised erinevused on suuremad kui rühmasisesed erinevused ja tõenäoliselt statistiliselt olulised.

F-statistik arvutatakse järgmiste valemite abil:

$$F = \frac{rühmadevaheline\ dispersioon}{rühmadesisene\ dispersioon}$$

$$rühmadevaheline\ dispersioon = \frac{\sum_{i=1}^{k} n_i(\overline{x_i} - \overline{x})^2}{k-1}$$

$$rühmadesisene\ dispersioon = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i} (x_{ij} - \overline{x_i})^2}{N-k}$$

kus:

- k on rühmade arv,
- n_i on vaatluste arv rühmas *i,*
- $\overline{x_i}$ on rühma *i* väärtuste keskmine,
- $\overline{x}$ on kõigi andmete keskmine.
- x_{ij} on *j*-nda vaatluse väärtus rühmas *i,*
- N on kõiki vaatluste koguarv.

ANOVA tulemust saab kirjeldada kahe väärtusega: *F*-statistiku ja vastava *p*-väärtusega. Seejuures *p*-väärtus näitab, kui tõenäoline on saada selline *F*-statistik (või suurem), eeldades, et nullhüpotees kehtib. Väike *p*-väärtus ehk väiksem kui määratud olulisuse tase (tavaliselt alla 0,05), viitab sellele, et nullhüpotees tuleks ümber lükata, kuna vähemalt ühe rühma keskväärtus on teistest statistiliselt erinev.

Kui ANOVA tulemus osutub statistiliselt oluliseks, viitab see rühmade vahelisele erinevusele. Kuid see ei anna teavet selle kohta, millised konkreetsed rühmad on üksteisest erinevad. Sellisel juhul on vaja läbi viia lisaanalüüs, mida nimetatakse *post-hoc* analüüsiks. *Post-hoc* analüüsi jooksul rakendatakse statistilisi teste nagu näiteks Tukey HSD test, et täpsustada, millised konkreetsed rühmad teistest erinevad.

ANOVA eeldused

ANOVA tulemuste usaldusväärsuse tagamiseks peavad olema täidetud järgmised eeldused.

1. **Vaatluste sõltumatus:**
   kõik vaatlused peavad olema sõltumatud nii rühmade sees kui ka rühmade vahel, st ühe vaatluse väärtus ei tohiks mõjutada teisi.
2. **Normaaljaotus:**
   andmed peavad olema normaaljaotusega iga rühma sees, seejuures pole see oluline marginaaljaotuse tasemel (ehk kõigi rühmade peale kokku).

Normaaljaotust saab hinnata visuaalselt (nt histogrammiga) või statistiliste testidega (nt Shapiro-Wilki test) iga rühma kohta. Suuremate valimite puhul on ANOVA robustne mõõdukate kõrvalekallete suhtes, kuid tõsised rikkumised võivad tulemuste usaldusväärsust mõjutada.

3. **Hajuvuse homogeensus:**
   rühmade vahel peaks hajuvus olema ligikaudu ühesugune, ehk dispersioon peaks igas rühmas olema ligikaudu sama.

Kuigi võrdne rühmade suurus ei ole ANOVA otsene eeldus, on see hea praktika, kuna see aitab vältida potentsiaalseid probleeme ja tagab usaldusväärsemad tulemused.

Kui need eeldused ei ole täidetud, tuleks kaaluda mitteparameetriliste alternatiivide kasutamist, näiteks Kruskal-Wallise testi, või andmete teisendamist, et parandada eelduste täitmist.

5.2 Mis on masinõpe?

Meie aju oskab hästi lahendada teatud meile lihtsana tunduvaid ülesandeid, mis tegelikult lihtsad ei ole. Me lahendame neid ülesandeid iga päev sadu kordi ilma ise nende keerukust märkamata. Näiteks on meie jaoks lihtne tuvastada oma nägemismeele (ja aju vastavate piirkondade) abil ekraanile kuvatud objekte. Looma liiki suudame hinnata ka siis, kui pole seda isendit või tõugu kunagi näinud, kasutades mingeid eelnevast kogemusest üldistatud reegleid. Samuti tunneme sõbra ära ka siis, kui ta on uue soengu saanud, kaalus juurde võtnud või kui me pole teda 10 aastat näinud. Me saame pingutuseta aru teise inimese kõnest, isegi mürarikkas keskkonnas, kus on tegelikult ka palju teisi helisid, mida kuulda võiks. Me langetame liigeldes ja igapäevaseid toimetusi tehes tohutul hulgal keerulisi otsuseid, et vältida objektidega ja teiste inimestega kokku põrkamist.

Kuigi nende ülesannete lahendamine ei valmista meie jaoks mingit probleemi, on meil peaaegu et võimatu formaalselt ehk algoritmiliselt (matemaatiliste või loogiliste sammudena) kirjeldada, kuidas me seda teeme. Tuvastamaks, et pildil on koer, mitte kass, millised arutlussammud toimuvad meie peas? Kui me suudaks need sammud kirja panna, suudaksime ka luua tehissüsteemi, mis kasse ja koeri eristaks. Samuti, kui autojuht suudaks kirja panna kogu oma otsustusprotsessi absoluutselt kõigis liiklussituatsioonides, suudaks me selle põhjal luua isejuhtiva masina. Praktikas näeme aga, et need protsessid (või mingid nende alamosad, näiteks liiklusolukorra tõlgendamine) toimuvad meie ajus liiga varjatult. Me ei tea, kuidas see toimub, ning seega me ei suuda neid samm-sammult kirja panna ja arvutisse programmeerida.

Masinõpe on väga oluline andmeteaduse valdkond, mis võimaldab arvuteid automaatselt ülesandeid täitma õpetada ilma neid sõnaselgelt programmeerimata. See tähendab, et inimene ei pea arvutile andma detailseid juhiseid, mida ja mis järjekorras programm või algoritm tegema peab, et õige väljundini (vastuseni, käitumiseni) jõuda. Piisab sellest, kui me näitame masinõppe algoritmile näiteid ehk sisestame andmed,

mille põhjal see saaks õppida. Masinõppe algoritmid on seega üldised **õppimise algoritmid**, mitte mingi kindla ülesande lahendamise algoritmid. Alles õppimise algoritmi töö tulemusena, kasutades mingit hulka andmeid näidetena, tekib püstitatud ülesannet lahendada suutev algoritm – **masinõppe mudel**. Selle õppimisprotsessi kohta öeldakse **mudeli treenimine** (ka optimeerimine). Seega kokkuvõtvalt: mingit ülesannet lahendada suutev mudel treenitakse olemasoleva andmekogumi põhjal, kasutades õppimisalgoritmi, mis on sobilik antud andmestiku analüüsiks.

Masinõppe mudel on matemaatiliste sammude (tehete, võrdluste) **loetelu, mille abil saab andmetest** (näidete tunnuste väärtustest ehk andmepunktidest) **teatud tüüpi mustreid avastada ja nende abil igale näitele sobiva väljundi arvutada**. Kindlaks määratud toimingute loetelu nimetataksegi algoritmiks ja selle põhjal saab luua arvutiprogrammi. Masinõppe mudelid on seega lihtsalt üks algoritmide alamliik. Mudeli põhjal loodud arvutiprogramm, saades sisendiks ükskõik millise **andmepunkti** (st tunnuste väärtuste komplekti), teeb need arvutuslikud sammud läbi ja tagastab mudeli väljundi selle andmepunkti jaoks. Väljund võib olenevalt ülesandest olla eri tüüpi, näiteks mingi arvulise mõõdu ennustus, mingisse klassi kuulumise tõenäosus vms.

Masinõppe meetodid ehk algoritmid erinevad üksteisest selle poolest, mil viisil nad andmetes mustreid avastavad, milliseid mustreid ära õppida suudavad ja millist tüüpi väljundeid tagastada oskavad. Erinevad algoritmid võivad samadel andmetel õppides jõuda väga erinevate tulemusteni. See võib tuleneda sellest, et üks algoritm on lihtsalt teisest parem, aga ka sellest, et üritatakse optimeerida erinevaid asju. Näiteks võib masinõppe süsteemi luues valida, kas püüame eksida võimalikult harva, võimalikult vähesel määral või, olenevalt ülesandest, püüame saavutada hoopis midagi muud, näiteks luua lihtsa ja arusaadava mudeli. Ennustades homset ilma, pole paari kraadi võrra eksimine tõenäoliselt väga suur probleem, ent ennustus võiks alati olla enam-vähem täpne. Ühe täielikult vale ennustusega võib inimese usalduse kaotada. Seega võib eksida tihti (lausa alati!), aga teha tohib ainult väikseid vigu. Seevastu ennustades jalgpallimängude võitjaid ja panustades neile raha, pole vahet, kui suure skooriga keegi võidab või kaotab. Oluline on ainult, kes võidab. Seega tuleb täpselt arvata võimalikult paljudel juhtudel ja eksimuste puhul pole eksimuse määr oluline. Tundlike andmetega töötades võib olla aga täpsusega sama oluline ka otsuste seletatavus ehk otsuse tegemise loogika mõistetavus inimese jaoks.

Masinõppe rakendamine tähendab, et andmeteadlane või masinõppe insener annab arvutile ette hulga andmeid, õppimisjuhised, kuidas nendest õppida, ja õpieesmärgi, mida saavutada. See toimub mitte sõnaliselt, vaid mingis programmeerimiskeeles muutujaid, valemeid ning algoritme defineerides või juba seadistatud algoritmide hulgast valides. Näiteks: „Kasutades andmestikku A, lineaarset mudelit ja gradientlaskumise õppimisalgoritmi, loo mudel, mis ennustab sisendtunnuste põhjal märgendi väärtust, minimeerides keskmist ruutviga näiteandmetel“ tuleb tõlgendada infoks, mille alusel arvuti mudeli valemi kindlaks määrab ja mudelit õpetama hakkab.

Kuigi see võib tunduda tehniliselt keerukas, on masinõppe rakendamine tehtud programmeerimiskeeltes nagu Python või R suhteliselt lihtsaks tänu standardsetele

masinõppe teekidele (ehk koodiraamatukogudele, pakettidele) nagu Scikit-Learn ja Caret. Enamasti ei pea andmeteadlane algoritme iga kord nullist arendama ja juurutama, vaid piisab mudeli tüübi ning üldiste omaduste (**hüperparameetrite**) määramisest, andmestiku ette andmisest, õppimisalgoritmi valimisest, õppimise kestuse või lõppemise kriteeriumide määramisest ja õppimisfunktsiooni käivitamisest. Kuigi palju algoritme on saadaval spetsiaalsetes teekides (joonis 5.3), on just sellest suurest hulgast andmetele ja küsimusele sobiva mudelitüübi valimine üks andmeteadlase olulisi oskusi, tema kunst.

Boosted Classification Trees	Model Rules	Ridge Regression with Variable Selection	Ridge Regression	Boosted Tree	Neural Networks with Feature Extraction	Rule-Based Classifier	Sparse Mixture Discriminant Analysis
Bagged AdaBoost	Model Averaged Naive Bayes Classifier	Fuzzy Rules Using Chi's Method	Robust Linear Model	C5.0	Principal Component Analysis	Partial Least Squares	Stabilized Nearest Neighbor Classifier
AdaBoost.M1	Mixture Discriminant Analysis	Fuzzy Rules with Weight Factor	Robust Mixture Discriminant Analysis	Cost-Sensitive C5.0	Penalized Discriminant Analysis	k-Nearest Neighbors	Sparse Linear Discriminant Analysis
AdaBoost Classification Trees	Maximum Uncertainty Linear Discriminant Analysis	Simplified TSK Fuzzy Rules	ROC-Based Classifier	Single C5.0 Ruleset	Penalized Discriminant Analysis	k-Nearest Neighbors	Spike and Slab Regression
Adaptive Mixture Discriminant Analysis	Multi-Layer Perceptron	Generalized Additive Model using Splines	Rotation Forest	Single C5.0 Tree	Penalized Linear Regression	Polynomial Kernel Regularized Least Squares	Sparse Partial Least Squares
Adaptive-Network-Based Fuzzy Inference System	Multi-Layer Perceptron, with multiple layers	Boosted Generalized Additive Model	Rotation Forest	Conditional Inference Random Forest	Penalized Linear Discriminant Analysis	Radial Basis Function Kernel Regularized Least Squares	Linear Discriminant Analysis with Stepwise Feature Selection
Model Averaged Neural Network	Multi-Layer Perceptron	Generalized Additive Model using LOESS	CART	CHi-squared Automated Interaction Detection	Penalized Logistic Regression	Least Angle Regression	Quadratic Discriminant Analysis with Stepwise Feature Selection
Model Averaged Neural Network	Multi-Layer Perceptron, multiple layers	Generalized Additive Model using Splines	CART	SIMCA	Partial Least Squares	Least Angle Regression	Supervised Principal Component Analysis
Naive Bayes Classifier with Attribute Weighting	Penalized Multinomial Regression	Gaussian Process	CART	Conditional Inference Tree	Partial Least Squares Generalized Linear Models	The lasso	Support Vector Machines with Boundrange String Kernel
Tree Augmented Naive Bayes Classifier with Attribute Weighting	Naive Bayes	Gaussian Process with Polynomial Kernel	Cost-Sensitive CART	Conditional Inference Tree	Ordered Logistic or Probit Regression	Linear Discriminant Analysis	Support Vector Machines with Exponential String Kernel
Bagged Model	Naive Bayes Classifier	Gaussian Process with Radial Basis Function Kernel	Quantile Regression with LASSO penalty	Cubist	Projection Pursuit Regression	Linear Discriminant Analysis	Support Vector Machines with Linear Kernel
Bagged MARS	Semi-Naive Structure Learner Wrapper	Stochastic Gradient Boosting	Non-Convex Penalized Quantile Regression	DeepBoost	Greedy Prototype Selection	Linear Regression with Backwards Selection	Support Vector Machines with Linear Kernel
Bagged MARS using gCV Pruning	Neural Network	Multivariate Adaptive Regression Splines	Regularized Random Forest	Dynamic Evolving Neural-Fuzzy Inference System	Knn regression via sklearn.neighbors.KNeighbors Regressor	Linear Regression with Forward Selection	Support Vector Machines with Polynomial Kernel
Bagged Flexible Discriminant Analysis	Neural Network	Fuzzy Rules via MOGUL	Regularized Random Forest	Stacked AutoEncoder Deep Neural Network	Quadratic Discriminant Analysis	Linear Regression with Stepwise Selection	Support Vector Machines with Radial Basis Function Kernel
Bagged FDA using gCV Pruning	Non-Negative Least Squares	Fuzzy Rules Using Genetic Cooperative-Competitive Learning	Robust Regularized Linear Discriminant Analysis	Linear Distance Weighted Discrimination	Robust Quadratic Discriminant Analysis	Robust Linear Discriminant Analysis	Support Vector Machines with Radial Basis Function Kernel
Bayesian Additive Regression Trees	Tree-Based Ensembles	Genetic Lateral Tuning and Rule Selection of Linguistic Fuzzy Systems	Robust SIMCA	Distance Weighted Discrimination with Polynomial Kernel	Quantile Random Forest	Linear Regression	Support Vector Machines with Radial Basis Function Kernel
Bayesian Generalized Linear Model	Oblique Trees	Fuzzy Rules via Thrift	Relevance Vector Machines with Linear Kernel	Distance Weighted Discrimination with Radial Basis Function Kernel	Quantile Regression Neural Network	Linear Regression with Stepwise Selection	Support Vector Machines with Class Weights
Self-Organizing Map	Single Rule Classification	Generalized Linear Model	Relevance Vector Machines with Polynomial Kernel	Multivariate Adaptive Regression Spline	Ensembles of Generalized Lienar Models	Logistic Model Trees	Support Vector Machines with Spectrum String Kernel
Binary Discriminant Analysis	Oblique Random Forest	Boosted Generalized Linear Model	Relevance Vector Machines with Radial Basis Function Kernel	Extreme Learning Machine	Random Forest	Localized Linear Discriminant Analysis	Tree Augmented Naive Bayes Classifier
Boosted Tree	Oblique Random Forest	glmnet	Subtractive Clustering and Fuzzy c-Means Rules	Elasticnet	Radial Basis Function Network	Bagged Logic Regression	Tree Augmented Naive Bayes Classifier Structure Learner Wrapper
The Bayesian lasso	Oblique Random Forest	Generalized Linear Model with Stepwise Feature Selection	Shrinkage Discriminant Analysis	Ensemble Partial Least Squares Regression with Feature Selection	Radial Basis Function Network	Boosted Logistic Regression	Bagged CART
Bayesian Ridge Regression (Model Averaged)	Oblique Random Forest	Generalized Partial Least Squares	Stepwise Diagonal Linear Discriminant Analysis	Ensemble Partial Least Squares Regression	Regularized Discriminant Analysis	Logic Regression	Variational Bayesian Multinomial Probit Regression

Joonis 5.2. Masinõppe mudelitüüpide näidisnimekiri. *Mudelitüübid, mis on saadaval Caret teegis R programmeerimiskeelele. See teek sisaldab üle 200 mudelitüübi.*[10]

Masinõppe meetodid võib jagada **juhendamata masinõppeks** (ingl k *unsupervised machine learning*), **juhendatud masinõppeks** (ingl k *supervised machine learning*) ja **stiimulõppeks** (ingl k *reinforcement learning*). Masinõppe meetodite jaotamine nendesse kategooriatesse põhineb sellel, kuidas ja mis eesmärgiga masinõppe mudel andmetest mingeid kasulikke seoseid otsib. Järgnevates peatükkides käsitleme neid kategooriaid põhjalikumalt.

[10] *Kuvatõmmis, joonise allikas: loeng „Juhendatud masinõpe bioinformaatikas", autor D. Fishman.*

5.3 Juhendamata õpe

Juhendamata õpe on protsess, kus algoritm õpib andmetest ilma seotud märgendita. See tähendab, et mudel ei ürita ennustada mingit konkreetset muutujat, vaid üritab otsida andmetest mingit peidetud struktuuri, mis seletaks või võtaks mingil viisil kokku andmetes peituvad reeglipärad. Sellisel juhul käsitleme me kõiki tunnuseid võrdsetena ega käsitle ühtegi neist tunnustest märgendina.

Juhendamata masinõppe põhiülesanne on leida andmetest mustreid ja gruppe, kasutades selleks vaatluste sarnasust üle mitme tunnuse (need objektid on sarnased) või tunnuste esinemise sarnasust üle mitme vaatluse (nende tunnuste esinemise vahel on seos). Sellised algoritmid on kasulikud, sest aitavad andmetest aru saada, **tekitada teadmisi**. See võimaldab näiteks palju tunnuseid mingi ühe kokkuvõtva tunnusega kokku võtta või palju vaatlusi ühte gruppi liigitada ning hiljem juba gruppide tasandil argumenteerida. Üldistused aitavad inimestel mõelda ja rohkem aspekte korraga arvesse võtta, sest inimese töömällu mahub korraga ainult 4–5 infokildu. Seega suure hulga info kasulikul viisil kokku võtmine aitab suurt kogust infot korraga tarbida, ajus töödelda, et rohkemaid aspekte arvesse võtta. Samuti on tunnuste arvu vähendamine kasulik paljude andmeteaduslike rakenduste jaoks, alates jooniste tegemisest (2D-joonisel saab kujutada ainult kahe tunnuse omavahelist seost) kuni juhendatud masinõppe mudeliteni, mida tutvustame järgmistes peatükkides.

Juhendamata õpe sarnaneb inimese alateadliku õppimisprotsessiga, kus inimene sordib objekte või sündmusi sarnasuse põhjal. Imikud suudavad õppida eristama ema ja isa nägu ning häält, ilma, et keegi neile „märgendi“ ette annaks. Samuti võib täiskasvanu aju tööl ette tulevaid juhtumeid sarnasuse alusel gruppidesse jagada või nende esinemises reeglipärasid avastada ilma, et keegi käsiks, või ilma, et sellest kasu tõuseks. Politseinikel võib kogemusest tekkida tunnetus, eelaimdus mingi olukorra olemuse kohta, mingisse juhtumite gruppi kuuluvuse kohta (see on kindlasti eriline juht, kus ei tohi lasta sel eelaimdusel fakte vaigistada). Toidukuller võib märgata, et just reedeti tellitakse mingil põhjusel pitsa kõrvale suurem pudel Coca-Colat. See tähelepanek ei too kullerile mingit kasu, vaevalt, et ta selliseid tellimusi vältima hakkab, sest pudel on raske. Ent kellelegi teisele, näiteks pitsarestoranile, võib see olla väga kasulik teave turunduse aspektist. Alljärgnevas räägimegi meetoditest, mis võimaldavad andmetest ilma mingeid eelaimdusi omamata reeglipärasid avastada.

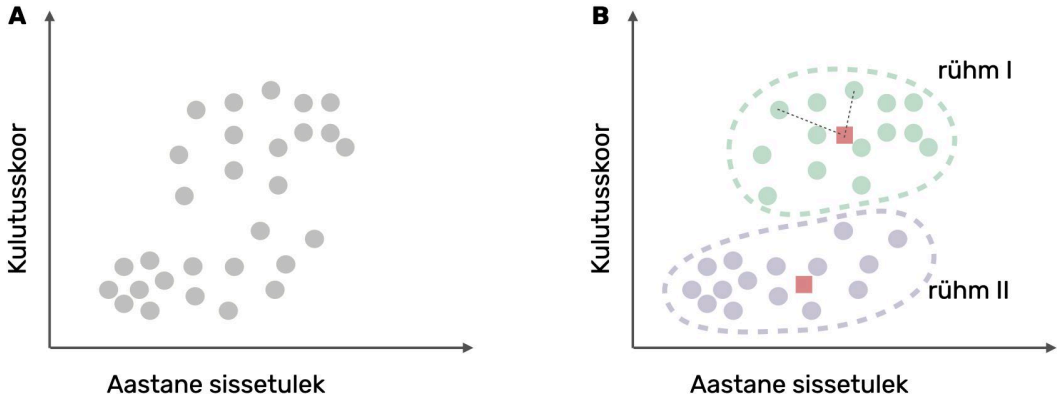
5.3.1 Klasterdamise meetodid

Juhendamata masinõppe ilmselt kõige tuntumaks rakenduseks on andmepunktide **klasterdamine** (ingl k *clustering*) ehk gruppideks jaotamine. Leides andmetest korduvaid mustreid, saame moodustada näidetest grupid, milles on koos mingil viisil omavahel sarnased, aga kõigist teistest näidetest piisavalt erinevad näited. Lastes algoritmil näiteid grupeerida, on võimalik avastada andmetes seni teadmata seoseid, mida inimene ise pole võimeline märkama ega defineerima. Kui andmestikus on sadu või tuhandeid tunnuseid ning tuhandeid või kümneid tuhandeid ridu, siis on keeruline lihtsalt peale vaadates hoomata, kas mingi hulk näiteid on omavahel sarnased ja eristuvad ülejäänud näidetest mingil viisil. Juhendamata masinõpe võimaldab

automatiseerida selliste peidetud gruppide avastamist. Neid gruppe nimetatakse masinõppe terminites **klastriteks** ja nende leidmist klasterdamiseks.

Klasterdamise kasutusjuhtu võib illustreerida järgmise näite abil. Kujutage ette, et olete poeomanik ja olete kogunud liikmekaartide kaudu mõned põhiandmed oma klientide kohta, näiteks kliendi ID, vanus, sugu, aastane sissetulek ja kulutusskoor. Kulutusskoor on tunnus, mis arvutatakse kliendi varasema käitumise ja ostuandmete põhjal. Teie soov on aru saada, milliseid kliente oleks loomulik kokku grupeerida, et turundusmeeskond saaks igale grupile erineva turundusstrateegia planeerida. Teil pole mingit eelnevat klientide liigitust, nii et iga kliendi kohta pole olemas mingit gruppi kuuluvuse märgendit. Samuti ei oska te aimata, mitu erinevat kliendirühma eksisteerib (joonis 5.3A).

Mitmesugused klasterdamismeetodid on saadaval näiteks Scikit-Learn klasterdamismeetodite teegi kaudu. Üsna sageli on esimesed klasterdamisalgoritmid, mida andmetel proovitakse, *k*-keskmiste ja hierarhiline klasterdamine. *K*-keskmiste klasterdamist (joonis 5.3B) kasutades otsib algoritm gruppidesse jaotumist, mille korral parameetrina ette antud arvu *k* klastrite puhul (joonisel esitletud juhul $k = 2$) on andmepunktide kaugus lähimast klastrikeskmest minimaalne. Masinõppimine seisnebki siinkohal klastrikeskmete asukoha optimeerimises nii, et treeningnäidete kaugus lähimast klastrikeskmest oleks keskmiselt võimalikult väike. Keerukamate algoritmidega saab korraga minimeerida nii näidete kaugust klastrikeskmest kui ka klastrikeskmete omavahelist kaugust. Samuti saab valida, mil viisil kaugus defineerida, sest Eukleidiline kaugus pole ainus võimalik kaugusmõõt.



Joonis 5.3. Kliendirühmade avastamine, kasutades juhendamata masinõpet. A *Joonis näitab algandmeid. Hallide täppidega on esitatud inimesed, kelle kohta on teada aastane sissetulek ja kulutusskoor.* ***B*** *Juhendamata õppe algoritm otsib algandmetest peidetud struktuuri ja leiab andmetest kaks kliendirühma. Punased kastikesed kujutavad klastrite keskmeid ja katkendlik must joon kujutab näidete kaugust klastri keskmest, mida* k*-keskmiste algoritm üritab minimeerida. Praktikas on iga kliendi kohta oluliselt rohkem tunnuseid kui ainult kulutusskoor ja aastane sissetulek, näiteks kliendi ID, vanus, sugu, kliendisuhte pikkus, ostusagedus jne, ning inimene ise sarnast analüüsi visuaalselt teha ei suuda.*

Mis on see, mis ühte klastrisse kuulujaid ühendab, on võimalik neid esindavate tunnuste väärtuste põhjal tagantjärele välja selgitada, et anda igale kliendiklastrile tähendus ja kirjeldus. Näiteks võivad ühe klastri moodustada juhuostjad, teise harva suuri oste tegevad kliendid, kolmanda iga päev väikseid oste tegevad kliendid jne. Teades ja

kirjeldades erinevaid klienditüüpe ja inimesi, kes sellesse tüüpi kuuluvad, saab ettevõte seada eesmärgiks näiteks juhuostjatest püsiklientide tegemise ning püsiklientide hoidmise.

Kui läheme klasterdamisest sammu kaugemale ja asume põhjalikumalt analüüsima, mis on ühist samasse klastrisse lahterdatud näidetel, siis teostame me **klastrite analüüsi.** Me avastame mingid tunnusekombinatsioonid, mis on klastri kõigil näidetel sarnased ja mis põhjustasidki nende samasse klastrisse sattumise. Need sarnasused võivad olla intuitiivsed seosed, mida me juba teame, kuid ka hoopis midagi meile etteaimamatut. Olles avastanud mingi seaduspära andmetes, saab seda mõtestada, et andmete olemust ja tekkepõhjust paremini mõista, reeglit täpsustada või mingil eesmärgil ära kasutada.

Mudeli definitsioon klasterdamise kontekstis
Klasterdamise mudel on matemaatiliste sammude loetelu, mille abil saab andmetes teatud tüüpi mustreid avastada ja nende abil igale näitele sobiva klastri numbri määrata.

5.3.2 Assotsiatsioonireeglite leidmine

Erinevalt klasterdamisest, mis otsib omavahel sarnaseid andmeridu, proovib **assotsiatsioonireeglite analüüs** (nimetatakse ka assotsiatsioonireeglite kaeveks) avastada peidetud seoseid tunnuste esinemise vahel. Otsitavad seosed on enamasti tugevad „kui → siis“ tüüpi seoseid kahe tunnuse esinemise või mingisse vahemikku kuulumise vahel. Näiteks „kui ostusumma on > 100 eurot, siis makstakse enamasti kaardiga“ või „kui asukoht on Tallinn, siis ei maksta enamasti sularahas“. Need reeglid on ühesuunalised ja vastassuunaline reegel ei pruugi paika pidada.

Suures, paljude tunnustega andmestikus on väga palju tunnuste omavahelisi kombinatsioone ja pidevate tunnuste puhul lõputu hulk tunnuste vahemikke, millest kõrgeid koosesinemise või koos mitte-esinemise sagedusi otsida. Seega piiratakse uuritavate seoste hulka, esmalt piirates reeglites osaleda võivate tunnuste/tunnusevahemike hulka esinemissageduse alusel (ingl k *support*, baas, alus). Pole mõtet otsida reeglit „ostusumma on lõigus [99 999; 100 000] eurot“ ja teiste tunnuste esinemise vahel, kui selliseid näiteid on andmestikus ainult üks. Seega uurime seoseid ainult tunnuste väärtuse vahel, mis esinevad piisavalt sageli, et olla huvitavad. Teine lävend seatakse reegli tugevusele (ingl k *confidence*, ka usaldustegur), näiteks soovime väljastada ainult koosesinemised, mis juhtuvad rohkem kui 80% juhtudest. Näiteks „kui summa on lõigus [9,99; 19,99], siis makstakse 81% juhtudest kaardiga“, tegeleb üsna sagedase juhuga ja on tugev. Reegleid võib ehitada ka keerulisemaid, näiteks „kui ostetakse jahu **ja** piima, siis 90% juhtudest ostetakse ka mune“ või „kui ostetakse margariini **või** võid, siis ostetakse ka teraviljatooteid“. Otsitavate reeglite keerulisus sõltub kasutatavast algoritmist, aga mida keerulisemaid kombinatsioone me otsime, seda eksponentsiaalselt rohkem võimalikke kombinatsioone on.

Kuigi paljud avastatud koosesinemise reeglid on väga intuitiivsed, võib avastada ka midagi väga ootamatut.

Näide ostukorvianalüüsi kontekstis
Ostukorvide järgi on võimalik leida seoseid erinevate kaupade ühte ostukorvi sattumiste kohta, et kirjeldada toodetevahelisi mustreid nende koosesinemise põhjal. Paljusid selliseid seoseid oskavad müügiinimesed juba intuitiivselt arvata, näiteks, et õlut ja küüslauguleiba ostetakse tihti koos (reeglid „kui õlu, siis küüslauguleib“ ja vastupidi). Andmeteaduse valdkonnas on kuulus aga assotsiatsioonireegleid otsides leitud seos, et tihti ostetakse mähkmeid ostes ka õlut (vastupidine seos paika ei pea). Assotsiatsioonireeglite analüüsi ülesandeks ongi leida analoogilisi seoseid andmetest – kui ostukorvis sisalduvad teatud kaubad, siis millised kaubad veel samas ostukorvis tõenäoliselt esinevad? Palju huvitavamaid ja ärilisest seisukohast paremini rakendatavaid tulemusi võib saada ka ostukorvide andmete uurimisel koos klientide demograafiliste andmetega. See võib viia ainult teatud gruppide puhul tugevate seoste avastamiseni, näiteks et just 25–30-aastane abielus mees ostab suure tõenäosusega koos mähkmetega ka õlut. Kokkuvõtteks: ka assotsiatsioonireeglid võimaldavad andmeid kirjeldada ja nendes sisalduvaid trende avastada, mida inimene ja lihtne statistika alati ei suuda.

Et üheks lävendiks, mille me reeglite otsimisel seame, on tunnuse väärtuse esinemissagedus, tekib loogiline probleem mitmekesiste andmestikega. Oleks loogilisem suur andmestik esmalt klasterdada ja sarnased andmeread eraldada. Ülal toodud „mähkmed → õlu“ reegel pole võib-olla piisavalt tugev, kui vaatame kogu ostuandmestikku, aga teatud kliendigruppides on.

Näide: klasterdamine + klastrite analüüs + assotsiatsioonireeglite analüüs
Vaatleme patsientide terviseandmete, diagnooside ajaloo analüüsimist. Me võime klasterdada patsiente neil esinenud haiguste ja nende esinemisaja või järjekorra alusel. Teeme seda näiteks kogu patsiendi eluea jooksul. Klasterdamise tulemusena liigitatakse patsiendid etteantud hulka klastritesse, mille olemust me uurime alles tagantjärele, pärast klasterdamist.

Vaadates (kirjeldades kirjeldava analüütika abil) igasse klastrisse sattunud isikute andmeid, võime anda klastritele nimed nagu „alati terved inimesed“, „varajaste ja rohkete sisehaigustega inimesed“, „peamiselt kopsuhaigustega, aga ilma muude haigusteta inimesed“ jne. Seda klastritele nime andmist nende sisu alusel võib pidada klasterdamise osaks ja sellel viisil tõlgendatud klastrikuuluvuste alusel saab juba üsna palju kasulikku teha.

Teatud klastreid veel põhjalikumalt uurides saame aga leida veel reeglipärasusi nagu „noorelt maksahaigusega intensiivravisse sattunud inimene (varajaste

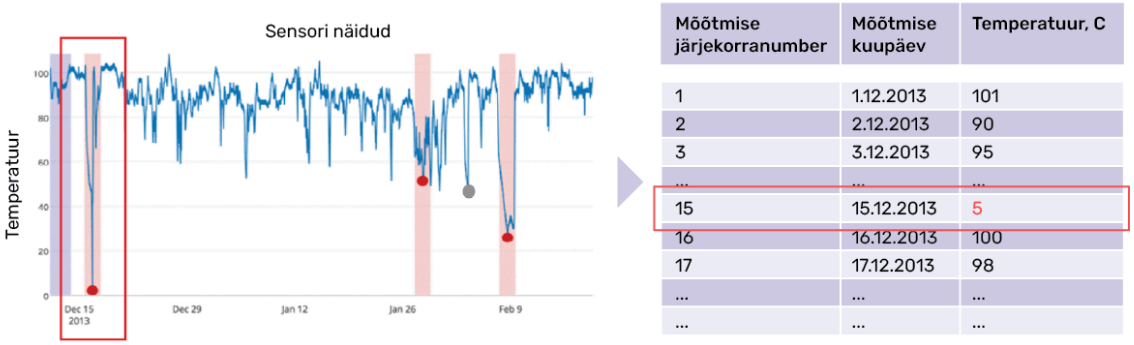
sisehaiguste klaster) enamasti satub sinna ka edaspidi korduvalt kuni surmani“ või „eelneva tõsise kopsuhaigusega inimestel viib Covid-19 suure tõenäosusega hospitaliseerimiseni“ (üldisel juhul võib-olla „Covid-19 → hospitaliseerimine“ ei ületaks tugevuse lävendit). Paljusid neist reeglitest me juba teame või võime aimata, aga automaatse tuvastuse käigus võib esile tulla reeglipärasid, mida keegi varem aimata ei osanud. Võib eksisteerida mingi aine vaeguse või ületarvitamisega seotud haiguste grupp, millest me veel teadlikud pole. Assotsiatsioonireegli tuvastamine nende haiguste esinemise vahel mingis patsiendigrupis ei anna meile veel vastust, mis on peidetud põhjus nende koosesinemise taga, aga tõstatab selle küsimuse. Muid andmestikke (vereanalüüsid) kasutusele võttes või teisi teaduslikke uuringuid kasutades on ehk võimalik need põhjused välja selgitada. Teaduses algab kõik alati õige küsimuse esitamisest, nii et mingi ootamatu assotsiatsiooni leidmine tõstatab põneva küsimuse, millele vastamiselt võib tekkida oluline ja praktiline teadmine.

5.3.3 Anomaaliate tuvastamine

Järgmine ülesannete klass, mida juhendamata õpe aitab lahendada, on **anomaaliate tuvastamine**, mis aitab leida ebatavalisi sündmusi või käitumisi. Anomaaliate ehk erindite tuvastamise eesmärgiga olete juba tutvunud alapeatükis 3.2. Siin peatükis keskendume sellele, kuidas juhendamata masinõppe meetodid saavad aidata anomaaliaid tuvastada.

Kõige levinumad ja lihtsamini tuvastatavad anomaaliad on võõrväärtused – mingi tunnuse väärtus on märgatavalt erinev teistest selle tunnuse väärtustest andmestikus (joonis 5.5). Sellised punktid tulevad esile karpdiagrammi joonistades või muul viisil mingi tunnuse väärtuste jaotust kirjeldades. Keerulisem on aga tuvastada anomaaliaid, mis seisnevad mitme tunnuste väärtuste haruldases kombinatsioonis – näiteks 190 cm pikkune ja 45 kg kaaluv inimene. Kumbki neist väärtustest eraldi pole kuigi haruldane, aga koos esinevad need väga harva. Ka selle kummalise näite võiksime me ise avastada, näiteks nende tunnuste põhjal hajuvusdiagrammi joonistades, aga paljude tunnustega andmestikus võib kõigi võimalike jooniste tähelepanelik läbivaatamine olla ajakulukas. Lisaks võib ette tulla anomaaliaid, mis seisnevad kolme või enama tunnuse väärtuste haruldases kombinatsioonis. Sellisel juhul ei ole inimesel endal enam väga lihtne neid anomaaliad andmetes näha, selle jaoks oleks vaja peas korraga kujutada kõiki teisi andmepunkte ning käesolevat andmepunkti nendega väga paljude nurkade alt võrrelda. Selliste anomaaliate automaatses leidmises seisnebki juhendamata õppe võlu selles

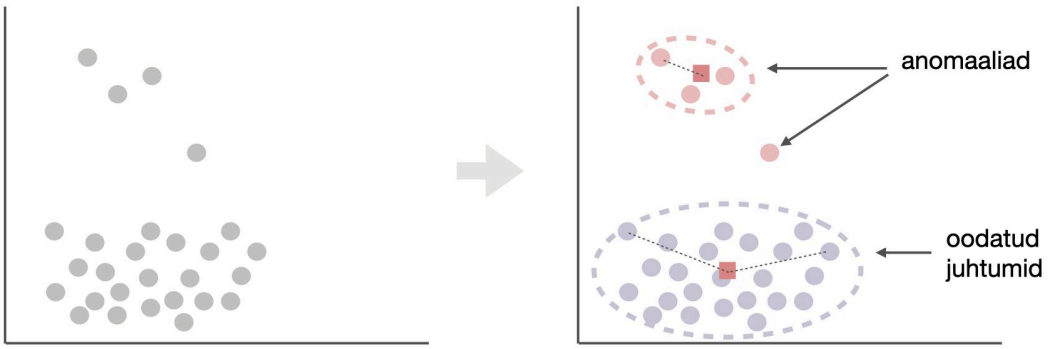
valdkonnas.



Mõõtmise järjekorranumber	Mõõtmise kuupäev	Temperatuur, C
1	1.12.2013	101
2	2.12.2013	90
3	3.12.2013	95
...	...	...
15	15.12.2013	5
16	16.12.2013	100
17	17.12.2013	98
...	...	...
...	...	...

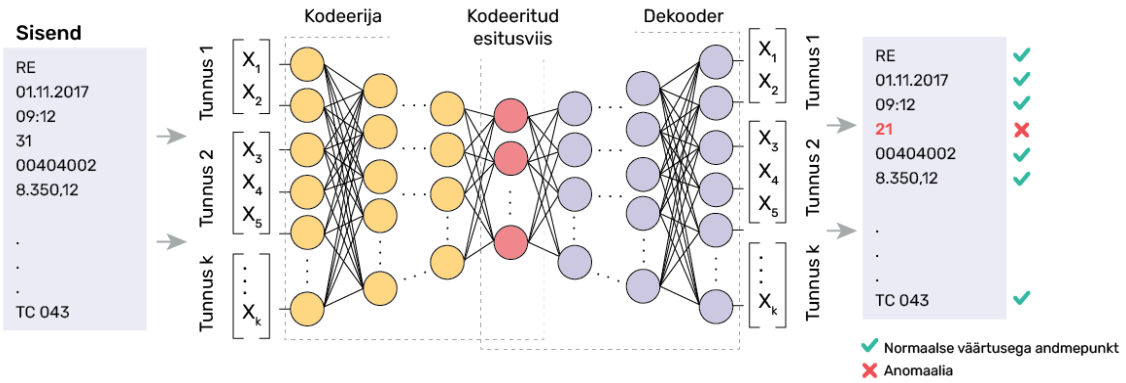
Joonis 5.4. Anomaaliate tuvastamine sensorite näitudest. *Kogutud temperatuurisensori andmed on esitatud joonisel ning tabelina, kus tunnusteks on mõõtmise kuupäev ja temperatuur ning vaatluse identifikaatoriks mõõtmise järjekorranumber. Punaste punktidega on esile toodud anomaalselt madalad sensori näidud. Kasutades juhendamata masinõppe algoritme nagu klasterdamine või autokooder, saab sellistest andmetest tuvastada kõrvalekaldeid.*

Näiteks võime klasterdamise tulemusena avastada näited, mis ei sobi väga hästi ühtegi klastrisse (joonis 5.5). Kujutage ette, et andmeid paberilt arvutisse sisestades on inimene poole rea pealt hakanud sisestama väärtusi järgmise rea vastavatest veergudest. Kui need kaks rida klaserdataks muidu klastritesse A ja B, siis selline pool-ja-pool andmerida võiks asuda nende kahe klastri vahel, aga mõlema keskmest üsna kaugel. Selle alusel, et see näide ühtegi klastrisse hästi ei sobi, tuleb selle andmerea erilisus esile ja meil on ehk võimalik seda vaadeldes mõista, et tegu on sisestamisveaga. Järsult („poolelt realt“) võib muutuda ka panga kliendi käitumine, kui kontole on ligipääsu saanud pettur. Sel juhul ei sobitu eelnev + praegune käitumine enam levinud käitumismustritega ja tekitab automaatse hoiatuse või konto ajutise blokeerimise.



Joonis 5.5. Anomaaliate tuvastamine klasterdamise abil. *Klasterdamist saab rakendada anomaaliate tuvastamiseks, eeldades, et normaalsed andmepunktid kuuluvad ühte või mitmesse klastrisse ja et anomaaliad ei kuulu nendega samadesse klastritesse. Näiteks* k*-keskmiste algoritm otsib algandmetest peidetud struktuuri ja leiab andmetest kaks klastrit. Punktid esindavad erinevaid andmepunkte. Punased kastikesed kujutavad klastrite keskmisi ja katkendlikud mustad jooned kujutavad andmepunktide kaugusi keskmisest, mida* k*-keskmiste algoritm üritab optimeerida. Kui andmepunkti kaugus normaalsete vaatluste klastri keskpunktist on liiga suur, võib see olla anomaalia. Kui kaugus on väike, lisatakse see klastrisse koos teiste tavapäraste punktidega.*

Sügavõppe valdkonna (vt pikemalt ptk 6) kiire areng on võimaldanud rakendada sügavaid tehisnärvivõrke ka anomaaliate tuvastamiseks. Selleks sobivad eriti hästi autokooderid (joonis 5.6). Autokooder koosneb kodeerijast ja dekooderist. Kodeerija kasutab sisendina toorandmeid ja annab väljundina nende andmete mingi esitusviisi ning dekooder kasutab kodeerija väljundit sisendina ja rekonstrueerib enda väljundina algsed sisestatud andmed. Autokooder õpib ja oskab rekonstrueerida treeningnäidete hulgas tavapäraseid andmeid. Ning just see võimaldab avastada anomaaliaid, sest anomaalseid andmeid mudel rekonstrueerida ei oska ning andmepunktide rekonstrueerimise viga ehk vahe rekonstrueeritud väljundi ja algse sisendi vahel on suurem.



Joonis 5.6. Anomaaliate tuvastus autokooderit kasutades. *Mudel kodeerib sisendi mingile väiksema dimensiooniga vektoresituse kujule ja proovib seda siis taasluua. Seda tehes peab mudel kasutama andmetes avastatud reeglipärasid, sest kodeerimisel surutakse info kokku vähemaks arvuks numbriteks, pole ruumi talletada kõiki detaile kõigi sisendtunnuste kohta. Kui andmetes oli müra või midagi muust andmestikust kõrvale kalduvat, ei suuda mudel seda täpselt rekonstrueerida, sest see ei järgi reeglipärasid. Siinses näites ei suutnud mudel rekonstrueerida neljanda tunnuse väärtust 31. Tegu on mudeli hinnangul anomaalse väärtusega. Kui keskmiselt üle kõigi tunnuste on rekonstruktsioon ebatäpne, on tegu anomaalse andmepunktiga.*[11]

5.3.4 Mõõtmelisuse vähendamine

Mõõtmelisuse vähendamine on paljudes andmeteaduse töövoogudes oluline samm, mille eesmärk on vähendada andmestike keerukust ilma probleemi kontekstis olulist informatsiooni kaotamata. See protsess on eriti oluline andmekogumite puhul, milles on palju tunnuseid. „Mõõtmelisuse needus“ (ingl k *curse of dimensionality*) võib muuta andmete analüüsimise keeruliseks ja aeganõudvaks, sest mida rohkem on mõõtmeid, seda kaugemal on keskmiselt üksteisest andmepunktid ning mudelitel võib olla keerulisem nende vahelisi seaduspärasid mõista. Rohkem tunnuseid tähendab ka seda, et on eksponentsiaalselt rohkem tunnuste kombinatsioone, mille hulgast mudel peaks leidma just need õiged, põhjuslikud seosed, mis andmete tekkeprotsessi või olemust kirjeldavad. Mõõtmelisuse vähendamise peamised eesmärgid ongi andmete visualiseerimise lihtsustamine, mudeli ülesobitamise (vt ptk 5.3.5) riski vähendamine ja arvutusliku efektiivsuse suurendamine, säilitades samal ajal võimalikult palju algsetes andmetes sisalduvat infot.

[11] Joonis on kohandatud, allikas: DeepAI, litsents: GPL-3.0.

Peakomponentide analüüs (ingl k *principal components analysis*, PCA) on üks enamkasutatavaid mõõtmelisuse vähendamise meetodeid. PCA leiab kõrgedimensionaalses tunnusteruumis suunad, milles andmete varieeruvus on suurim. Need suunad on kirjeldatud ühikvektoritega, mida nimetatakse peakomponentideks. Peakomponendid leitakse ükshaaval, iga järgnev peakomponent peab eelnevatega ristuma, samal ajal maksimeerides andmete varieeruvust enda defineeritud suunal. Seega tekib justkui uus ristteljestik, mille esimestel dimensioonidel on andmed rohkem välja venitatud ja varieeruvad ning viimastesse dimensioonidesse jääb pigem lihtsalt väikese amplituudiga müra. Projekteerides andmed valitud hulgale esimestest peakomponentidest, saame uue representatsiooni andmetest, mis on väiksema arvu tunnustega, kuid säilitab enamiku variatiivsusest. Vähemaid tunnuseid on lihtsam visualiseerida ja teatud masinõppe mudelitüüpide jaoks on väiksem arv sisendtunnuseid eelistatav.

Peale PCA on ka teisi mõõtmelisuse vähendamise tehnikad nagu **lineaarne diskriminantanalüüs** (LDA) ja **t-SNE** (*t-distributed Stochastic Neighbor Embedding*), mis panevad mõõtmelisuse vähendamisel rõhku teistsugustele aspektidele, mitte alles jääva variatiivsuse maksimeerimisele. LDA keskendub klassidevahelise varieeruvuse maksimeerimisele, muutes selle kasulikuks klassifitseerimisülesannete eeltöötlusena. T-SNE suudab kasutada ka mittelineaarseid seoseid tunnuste vahel. T-SNE püüab säilitada naabrus- ja lähedussuhted andmepunktide vahel – kui kaks punkti olid sarnased algsel, kõrgedimensionaalsel kujul, siis vähemate mõõtmetega samu andmeid kokku võttes peaks need punktid sarnaseks jääma. T-SNE väljundite visualiseerimine (näiteks 3D-punktipilvena) võimaldab andmepunktide sarnasusi mõista, aga algseid, rohkem kui kolme mõõtmega andmeid me visualiseerida ei oleks saanud. Faktoranalüüs (FA) püüab leida tunnustevahelisi seoseid (korrelatsioone) ja kirjeldada andmeid väiksema arvu muutujate, faktorite, abil. Rohkem teavet faktoranalüüsi kohta leiate siit.

Mõõtmelisuse vähendamise praktiliste rakenduste hulka kuuluvad näiteks genoomika, kus PCA võib aidata tuvastada geneetilisi mustreid, turundusanalüüs, kus LDA aitab eristada erinevate kliendisegmentide ostukäitumist, ning sotsiaalmeedia analüüs, kus T-SNE võimaldab visualiseerida kasutajatevahelisi suhteid ja peidetud gruppe suurtes (paljude tunnustega) andmehulkades.

Kokkuvõttes võimaldab mõõtmelisuse vähendamine andmeteadlastel ja analüütikutel tegeleda suurte andmekogumitega tõhusamalt. Sobiv mõõtmelisuse vähendamise meetod parandab analüüside kvaliteeti (eemaldades müra), mõistetavust (visualiseerimise abil) ja efektiivsust (vähem tunnuseid).

5.4 Juhendatud õpe

Juhendatud õpe on protsess, mis etteantud andmestiku alusel loob mudeli, mille eesmärk on ennustada iga andmepunkti kohta mingi(te) ettemääratud tunnus(t)e väärtus. Tunnust, mille ennustamine käsil on, nimetatakse **märgendiks** (tihti kasutatakse ka termineid väljund või silt, ingl k *output*, *label*, *target*). Eesmärk on suure hulga etteantud andmete ja märgendite põhjal ära õppida seosed ja mustrid, mis aitavad

korrektselt märgendeid ennustada. Nende mustrite alusel saab juba uute andmete jaoks ennustusi teha.

Mudeli definitsioon juhendatud masinõppe kontekstis
Juhendatud masinõppe mudel on matemaatiliste sammude loetelu, mille abil saab andmetest teatud tüüpi mustreid avastada ja nende abil igale andmereale sobiva märgendi väärtuse ennustada.

Näidisprobleem
Kokk loob uut supiretsepti, kasutades varem valitud koostisosi. Ta üritab leida maitseainete ja koostisosade hulkade õiget tasakaalu. Keetes teatud koostisega supi, annab ta seda oma restorani klientidele maitsta ja saab tagasisidena skoori, kui kõrgelt kliendid seda suppi keskmiselt hindavad. Eri koostistest ja skooridest loob ta andmestiku, kus read vastavad supivariantidele, tunnusteks on koostisosad, tunnuste väärtusteks kasutatud kogused ja märgendiks klientide hinnang. Juhendatud masinõppe abil saame selle andmestiku põhjal luua mudeli, mis võimaldab juba enne supi keetmist ennustada, millise hinnangu supp saab. Selle abil saab läbi katsetada lõputu hulga erinevaid retsepte ja valida parima.

Siinkohal kasutame terminit „mudel ennustab“, sest mudel võib olla ka väga halb ja kalkulatsioonide tulemusena tagastada väärtusi, mis ei sarnane absoluutselt tegelike märgenditega. Aga fakt, et me oleme mudeli matemaatiliselt kirja pannud, annab meile võimaluse mudeli eksimusi analüüsida. Võrreldes ennustusi ja tegelikke hinnanguid, mille kliendid eri koostisega suppidele andsid, saame hakata oma mudelit muutma nii, et ennustused läheksid täpsemaks. Seda mudeli muutmist nimetatakse mudeli optimeerimiseks ehk mudeli õpetamiseks. Muudatusi ei tehta juhuslikult, vaid õppimisalgoritmide abil, mis olenevalt mudeli tüübist võivad mudelis sisalduvaid matemaatilisi samme muuta, lisada või eemaldada. Õppimise tulemusena saame mudeli, mis ennustab märgendit täpsemini kui enne õppimist.

Vastavalt ennustatava märgendi tüübile võib juhendatud õppe jagada kahte gruppi: **klassifitseerimine** ja **regressioon**. Juhendatud õppe ülesandeid, kus märgend on kategooriline väärtus, nimetatakse klassifitseerimiseks. Näiteks, kui tahame inimese ajust mõõdetud aktiivsuse (nt elektroentsefalograafia, EEG) põhjal ennustada, millist tüüpi pilti (10 eri kategooriat) tema silmad parasjagu näevad. Teine juhendatud õppe tüüp on regressioon, kus märgendiks on mingi arvuline väärtus (nt hind, kogus, suurus, pikkus). Näiteks tahame kinnisvaraobjektide andmete (nt ruutmeetrid, kaugus kesklinnast, sisustuse olemasolu) põhjal ennustada kinnisvara lõplikku müügihinda, mis on arvuline väärtus. Ka keerulisemad ülesanded nagu pildi genereerimine sõnalise kirjelduse järgi, tõlkimine või valgu 3D-struktuuri ennustamine tõlgitakse matemaatiliselt kas klassifitseerimiseks (mis on järgmine sõnaosa?) või regressiooniks (mis väärtusega on iga piksel?).

Masinõppe sissejuhatavas peatükis mainisime, et mudeleid on mitut tüüpi. Nad erinevad selle poolest, kui palju ja milliseid matemaatilisi samme nad sisaldada saavad ning seega milliseid mustreid avastada suudavad. Loetleme siin mõned tuntud mudelitüübid: lineaarne mudel, logistiline regressioon, otsustuspuu, otsustusmets, tugivektormasin ja tehisnärvivõrgud. Igaühel neist on teistest natuke erinev kuju ja õppimisloogika.

5.4.1. Regressioon

Järgnevalt vaatame mõnda mudelitüüpi, mis sobivad regressiooniülesannete lahendamiseks, see tähendab mingite teada olevate tunnuste ehk sisendtunnuste põhjal reaalarvulise märgendi ennustamiseks. Regressiooni puhul on õppimise eesmärk lihtsasti mõistetav – muudame mudelit nii, et erinevus väljundite ja tegelike märgendite vahel väheneks. Alustame kõige lihtsama mudelitüübiga, et mõista regressiooniülesande olemust.

Lineaarne regressioon

Lineaarne regressioon on üks enam kasutatavaid ennustava analüütika tehnikaid. Andmete kirjeldamise peatükis tutvustasime tunnustevahelise korrelatsiooni kasutamist muutujatevahelise seose uurimiseks ja lineaarsete seoste leidmiseks kas kirjeldaval või diagnostilisel eesmärgil. Kui korrelatsioonianalüüs võimaldab muutujatevahelist lineaarset seost leida, on võimalik kirjeldada ka vastav lineaarvõrrand, mis seob ühe muutuja teisega. Selle võrrandi abil saame iga muutuja ***x*** väärtuse korral arvutada ennustuse muutuja ***y*** väärtuse kohta. Nii saame ka veel teadmata või tuleviku väärtusi ennustada. Võib tekkida küsimus, kas korrelatsiooni leidmine ja regressioon on sama asi. Vastus on ei. Korrelatsioon lihtsalt mõõdab kahe muutuja vahelise lineaarse suhte tugevust, püüdmata nendevahelist seost valemiga defineerida, samuti ei sea korrelatsioon ühte tunnust sisendiks ja teist märgendiks. Lineaarne regressioon aga tagastab meile lineaarvõrrandi, mis ennustab sisendtunnuse järgi võimalikult täpselt väljundtunnust, leides selle jaoks optimaalsed **parameetrite väärtused**.

Lineaarne regressioon on juhendatud masinõppe meetod, mis kirjeldab suhet kahe muutuja vahel lihtsa matemaatilise võrrandi kujul:

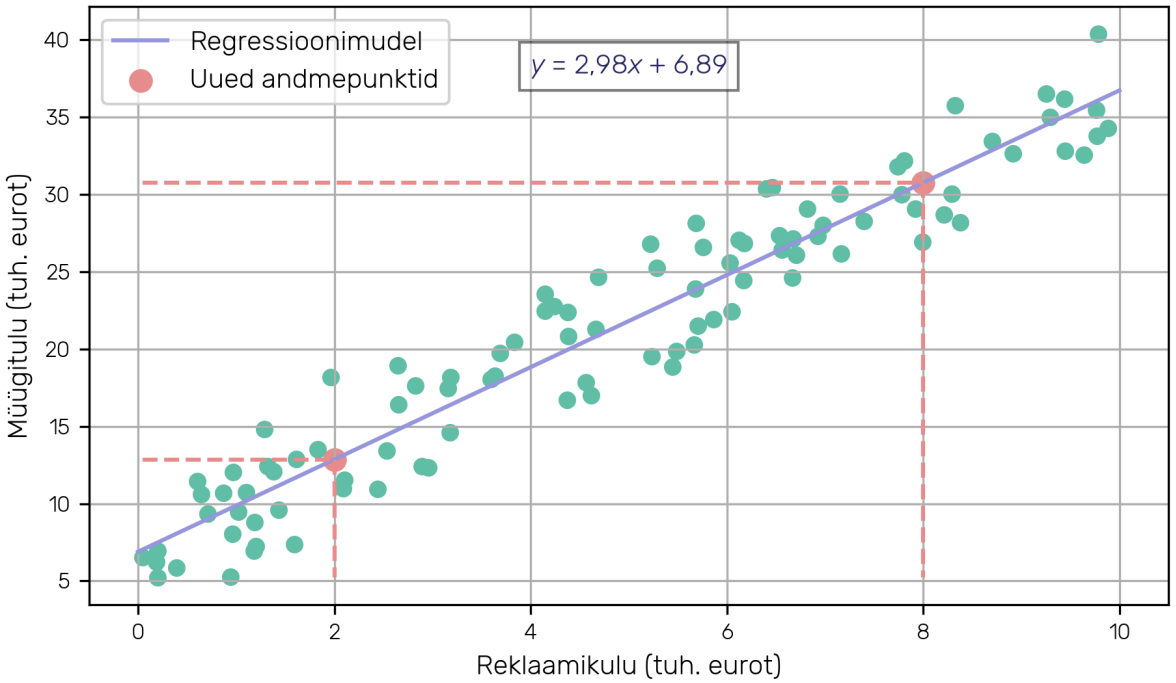
$$\hat{y} = b_0 + b_1 x.$$

Selles valemis esindab $\hat{y}$ mudeli väljundit ehk suurust, mida me tahame ennustada, ja x on ennustav ehk sõltumatu muutuja ehk masinõppe keeles sisendtunnus. Arvud b_0 ja b_1 on vastava mudeli koefitsiendid ehk parameetrid, mille kõige „kasulikumaid“ väärtusi õppimise ehk optimeerimise käigus otsitakse. „Kasulikumad“ parameetrite väärtused on enamasti defineeritud kui väärtused, mis optimeerimiseks kasutatud andmetel (treeningandmestikul) keskmiselt kõige väiksemaid ruutvigu annavad. Ruutviga on ennustuse ja tegeliku märgendi erinevuse ruut.

Lineaarse regressiooni intuitsiooni on lihtne illustreerida järgmise näite abil. Oletame, et olete poe omanik ja olete täheldanud, et kui kulutate rohkem reklaamile, siis tõenäoliselt

ka teenite rohkem. Samuti teate, näiteks kirjeldava analüüsi tulemusena, et reklaamikulu ja müügi vahel on positiivne korrelatsioon – kui üks kasvab, kasvab ka teine.

Mida te aga andmetele peale vaadates ei tea, on lineaarne võrrand, mis seob neid kahte muutujat ja aitaks teil reklaamikulu väärtust arvestades müügi suurust ennustada. Selle seose saamiseks võibki kasutada lineaarset regressiooni. Lineaarse regressiooni mudel meie näites on matemaatiline võrrand, kus sõltumatu muutuja x on reklaamikulu ja sõltuv muutuja y on müügitehingute summa. Meie eesmärk on treenida mudelit olemasolevate andmepunktide peal, ja sobitada joon, mis kõige paremini (kõige väiksemad ruutvead) esindaks seost meie andmetes. Lineaarse mudeli puhul on olemas vähimruutude meetod, mis lubab treeningandmete alusel parameetrite b_0 ja b_1 optimaalsed väärtused lihtsalt arvutada. Seega koosneb „treenimine“ lihtsalt ühest analüütilise valemi rakendamisest andmetele. Vähimruutude meetodi defineerime matemaatiliselt järgmises alapeatükis.



Joonis 5.7. Lineaarne regressioon. *Lineaarne seos müügitulu ja reklaamikulu kohta on esitatud rohelise joonena ning kirjeldatud matemaatilise võrrandina. Mudeli ennustused uute väärtuste puhul saab arvutada valemiga või leida graafiliselt (punased jooned). Lähtekood.*

Meie näites saame treenimise tulemusena, et võrrand on järgmine (joonis 5.7):

müügitehingute summa = 2,98 • reklaamikulu + 6,89.

Näiteks, kui me kasutame reklaamile ainult 180 eurot, siis mudel ennustab, et saame müügituluna 2,98 • 0,18 + 6,89 = 7,426 tuhat eurot. Siinkohal on olulised ühikud. Kõik summad on meil tuhandetes, seega on 180 eurot mudeli silmis 0,18 ja vabaliige 6,89 tähendab 6890 eurot.

Koodinäide
Sellise mudeli loomise koodinäite leiate siit Google Colab vihikust.

Sellist mudelitüüpi ei saa küll alati usaldada ning keerulistes probleemides sõltub ennustatav rohkematest tunnustest ja tihti ka keerulisemal viisil kui lineaarne seos. Siiski, paljudel juhtudel on selline lihtne ennustav mudel kasulikum kui mitte üldse mudelit omada ja seoseid kvantifitseerida. Paljud seosed maailmas ongi lineaarsed, näitena võite mõelda kasvõi füüsikast tuntud loodusseadustele.

Teised mudelitüübid

Regressioonanalüüsi meetodite mitmekesisus on aga palju laiem kui ainult lihtne lineaarne regressioon. Peaaegu kõiki tuntud mudelitüüpe (otsustuspuud, tugivektormasinad, tehisnärvivõrgud jne) saab kohandada teostama nii klassifikatsiooni kui ka regressiooni. Konkreetse meetodi valik sõltub püstitatud küsimusest, saadaolevate sisendandmete hulgast ja tüübist ning arvutusvõimsusest.

Näide mudelist, milles tehakse ennustus rohkema kui ainult ühe sõltumatu muutuja alusel, on mitmene lineaarne regressioon. See annab muutujaid (sisendtunnuseid) $x_1 \ldots x_n$ kasutades valemi:

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \ldots + b_nx_n.$$

Kõikide koefitsientide *b* väärtuste leidmiseks andmestiku *X* alusel saame kasutada vähimruutude meetodit, mille valemi siinkohal defineerime.

Vähimruutude meetod

$$b = (X^TX)^{-1}X^TY \quad ,$$

kus b = (b_0 , b_1 .. b_n) on empiiriliselt (andmete alusel) leitud koefitsente sisaldav vektor, mida me püüame leida. X on meie treeningandmestik, kus tunnuste väärtuste veergudele lisaks on esimese veeruna lisatud vaid number ühtesid sisaldav veerg, mis vastavad vabaliikme b_0 alatisele esinemisele valemis (valemis on justkui kirjas $b_0 \cdot 1$). Y on meie märgendite veerg. Valemis kasutatakse maatriksalgebra operatsioone transponeerimine, pöördmaatriksi leidmine ning maatrikskorrutis. Kui sõltumatuid muutujaid on 1 (ühene lineaarne regressioon), siis on treeningandmete maatsiksil vaid kaks veergu (ühtede veerg ning muutuja veerg) ning b vektoris vaid kaks elementi.

Lisaks lineaarse regressiooni variatsioonidele on trendide leidmiseks ja arvuliste märgendite ennustamiseks suurel hulgal erinevaid algoritme, näiteks otsustusmetsad ja tehisnärvivõrgud. Tavaliselt on masinõppe mudelite teekides nagu Scikit-Learn või Caret iga meetodi või mudeli jaoks saadaval kasutusjuhend selgitustega, kuidas ja milliste andmetega seda meetodit kasutada saab. Kogenud andmeteadlane oskab harilikult

valida antud ülesande jaoks optimaalse meetodi, arvestades andmete mahtu, eeldatavat seoste keerukust, saadavalolevat arvutusvõimsust, nõudeid seletatavusele jne. Käsitleme mitut meetodit pikemalt hilisemates peatükkides ja võimaluse korral mainime, kas neid saab ka regressiooni jaoks kohandada. Siinkohal oli meie eesmärk peamiselt näitlikustada regressiooni ideed.

5.4.2. Klassifitseerimine

Kategooriliste märgendite (näiteks objektitüüp, kodulinn, loomaliik) ennustamist nimetatakse **klassifitseerimiseks**. Klassifitseerimise ülesanne oleks näiteks ennustada, kas korter müüakse esimese kuue kuu jooksul. Vastus on kas „ei“ või „jah“ ehk kategooriline. Võimalik on luua ka mudeleid, mis lubavad vastata „ei“, „jah“ ning „ei tea“. Selles õpikus ei ole võimalik tutvustada kõiki võimalikke klassifikatsiooni teostada lubavaid mudeltiüüpe ja algoritme. Siiski, et anda teile aimu, kuidas klassifitseerimisalgoritmid täpsemalt töötavad, teeme pikemalt läbi kaks näidet kahe algoritmiga ja seejärel tutvustame lühidalt veel mõnda meetodit, mis lubavad teostada nii klassifikatsiooni kui ka regressiooni.

Logistiline regressioon

Logistiline regressioon on masinõppe meetod, mida kasutatakse binaarsete (jah/ei, tõene/väär) tulemuste ennustamiseks andmete põhjal. Erinevalt lineaarsest regressioonist, mis ennustab pidevat muutujat, on logistilise regressiooni eesmärk leida tõenäosus (väärtus 0 ja 1 vahel), et antud sisend kuulub positiivsesse klassi. Kuigi me räägime mudeli väljunditest tihti kui tõenäosustest, on need tegelikult ikkagi ainult mudeli ennustused tõenäosuse kohta. Halb mudel väljastab väga valesid tõenäosusi, mis tegelikkusele ei vasta.

Logistiline regressioon on keerulisem kui lineaarne regressioon, sest me soovime, et vastus jääks lõiku [0; 1] ja oleks tõlgendatav mingi sündmuse toimumise tõenäosusena. Seetõttu anname logistilise regressiooni jaoks kaks formulatsiooni (sama seose erinevat teisendust), millest esimene kujutab otseselt väljundtõenäosuse arvutamise valemit ja teine kajastab paremini selle mudeli sisemist loogikat.

Esimene esituskuju (väljundtõenäosuse arvutamise valem):

$$P(y = 1) = 1/(1 + e^{-(\beta_0 + \beta_1 x)}).$$

Näeme, et mudeli usk, et õige vastus on 1, on võrdne teatud valemi väljundiga. See valem saab omada väärtusi ühest nullini, sest eksponendiga valemiosa väärtus saab varieeruda nullist lõpmatuseni. Samuti märkame, et eksponendil asub lineaarse regressiooni valem, mis koosneb vabaliikmest β_0 ja sisendi x korrutisest koefitsiendiga β_1. Miinusmärgi tõttu eksponendil on kogu valemi väljundtõenäosus seda lähedasem arvule 1, mida suurem on selle lineaarse valemi väljund.

Teine esituskuju:

$$ln(\frac{P(y=1)}{1-P(y=1)}) = logit(P(y=1)) = \beta_0 + \beta_1 x$$

Selles esituses näeme, et tunnust *x* ning koefitsiente β_0 ja β_1 kombineeriv **lineaarse regressiooni** valem ütleb meile midagi sündmuse juhtumise ning mitte juhtumise tõenäosuste suhte kohta, täpsemini selle suhte logaritmi kohta. Seda juhtumise ja mittejuhtumise omavahelise suhte logaritmi nimetatakse tõenäosuse *P*(*y* = 1) logitiks. Et me lineaarse regressiooni valemiga ennustame logitit, annabki algoritmi nimeks logistiline regressioon. Seega me lahendame siinkohal tegelikult juba meile tuttavat lineaarse regressiooni ülesannet, ennustades tunnuse x põhjal antud näite positiivsesse klassi kuulumise logitit.

Omades treeningnäiteid, mille puhul me teame nii *x* kui *y* väärtusi, saame otsida parimaid β_0 ja β_1 väärtusi. Parimateks väärtusteks oleks väärtused, mida kasutades ennustaks mudel võimalikult suuri *P*(*y* = 1) tõenäosusi (seega ka võimalikult suurt logiti väärtust), kui märgend oligi 1, ning võimalikult väikseid väärtusi, kui märgend oli 0. Selliste optimaalsete parameetrite õppimiseks treeningandmete põhjal kasutatakse suurima tõepära meetodit (ingl k *maximum likelihood estimation*, MLE)[12].

Nagu lineaarse regressiooni puhul, on ka siin võimalik väljendada väljundi sõltumist mitmest sisendtunnusest, muutes regressioonivalemi mitmeseks:

$$ln(\frac{P(y=1)}{1-P(y=1)}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ... + \beta_n x_n .$$

Ka sisendite rohkemasse kui kahte (ei/jah) klassi liigitamine on võimalik, kasutades multinomiaalset logistilist regressiooni, mille käigus valitakse üks klass referentsklassiks ning optimeeritakse hulk logistilise regressiooni mudeleid iga ülejäänud klassi ja referentsklassi tõenäosuste suhet kirjeldama. Selline lähenemine võimaldab edukalt anda ka vastust „ei tea“ või „ei ole kindel“.

Toome lõpetuseks ka ühe näite logistilise regressiooni kasutusjuhust. Kujutlegem kindlustusettevõtet, mis soovib ennustada, millised kindlustatud sõidukid suurema tõenäosusega järgmise aasta jooksul liiklusõnnetuse põhjustavad. Sellise ennustuse jaoks on ettevõttel kogutud andmeid kindlustatud sõidukite kohta, sealhulgas vastutava kasutaja vanus, sugu, varasemate õnnetuste olemasolu ja sagedus, teiste kasutajate hulk, sõiduki vanus ja tüüp. Eelmise aasta kohta on ka teada, kas sõiduk põhjustas mõne õnnetuse. Seega on meil eelmise aasta kohta märgendatud andmestik. Võime neil andmetel luua logistilise regressiooni mudeli, mis ennustab õnnetuse tõenäosust **eelmisel aastal**. Kui usume, et sisendeid ja väljundeid siduvad seaduspärad pole aastaga muutunud, võime mudelit rakendada ning ennustusi teha ka tuleva aasta jaoks. Mudel oleks siis kujul

$$ln(\frac{P(y=1)}{1-P(y=1)}) = \beta_0 + \beta_1 vanus + \beta_2 sugu + \beta_3 avariisid + ... + \beta_n sõidukivanus$$

[12] Ka lineaarse mudeli loomist võib vaadelda kui tõepära maksimeerimist – me otsime mudelit, mis kõige tõenäolisemalt treeningandmed tekitas, tehes selle jaoks teatud eeldusi, näiteks et mudeli vead järgivad normaaljaotust.

ja P peegeldaks järgmise aasta jooksul õnnetusse sattumise tõenäosust. Selle alusel võiks kindlustusettevõte pakkuda kliendile kas kõrgemat või madalamat kindlustushinda.

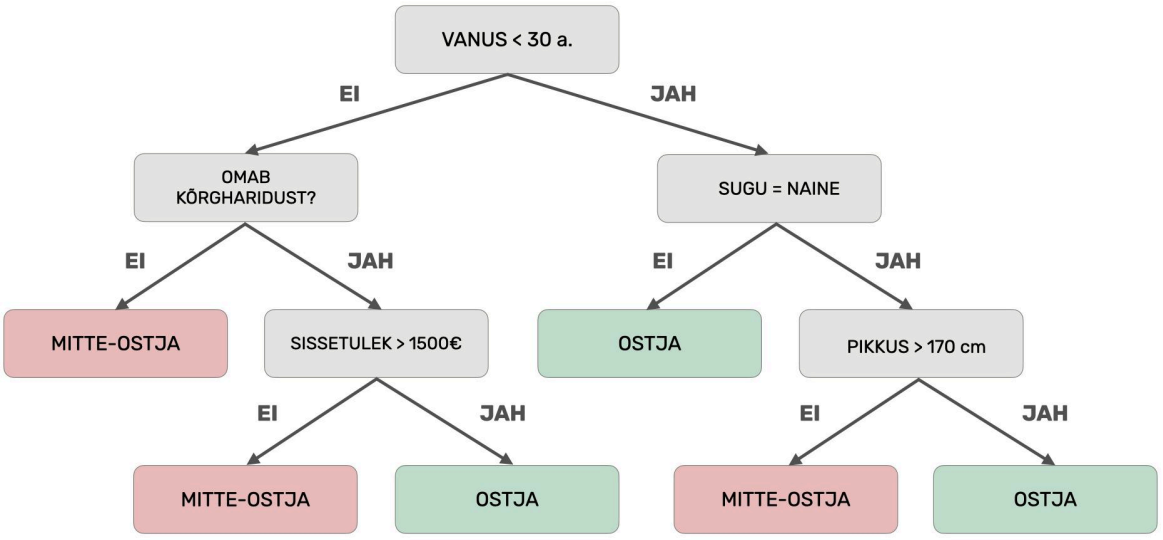
Praktiline näide
Koodinäite logistilise regressiooni mudeli loomisest sellise sisuga andmetel leiate siit: (Google Colaboratory vihik).

Otsustuspuud

Teise näite toome mudelitüübist nimega **otsustuspuu** (ingl k *decision tree*), sest see on ilmselt üks kõige intuitiivsemaid klassifitseerimise mudeleid. Otsustuspuu on järjestikustest otsustest koosnev „puu“ (joonis 5.8). Teisiti öeldes koosneb otsustuspuu järjestikest küsimustest, mida on vaja iga sisendi kohta esitada, et jõuda vastuseni ehk klassini. Küsimustele vastamist nimetatakse otsustamiseks. Need **puud koosnevad hargnemispunktidest** (teaduslikus keeles sõlmedest) ja **okstest** (teaduslikus keeles harudest). Hargnemispunktid vastavad „otsustele“ ehk kas-küsimustele. Igast hargnemispunktist väljub kaks oksa – „jah“-oks ja „ei“-oks.

Ütleme, et oleme kasutanud mingis programmeerimiskeeles olemas olevaid teeke (näiteks sklearn teek Pythonis) ja treeninud olemasolevatel klientide andmetel ühe otsustuspuu, et uute registreerunud klientide korral ennustada, kas nad jõuavad ostuni. See puu on loodud märgendatud andmestikul, algoritm on valinud küsimused just nii, et treenimisel kasutatud kliente (tunnuste-märgendi komplekte) võimalikult väheste küsimustega ja võimalikult täpselt liigitada. Selle jaoks valitakse igas hargnemispunktis just selline küsimus, mis „ei“ ja „jah“ vastuse saanud treeningnäitete grupid võimalikult klassipuhtaks saaks (Gini kriteerium, entroopia) või mingit muud mõõdetavat kriteeriumi maksimeeriks (nt entroopia kahanemine). Lisaks küsimuste valikukriteeriumi kindlaks määramisele on vaja ka otsustada, millisel hetkel uute harude lisamine lõpetada. Seda tehakse enamasti siis, kui harusse on jäänud vähem kui teatud hulk treeningnäiteid või kui ükskõik millise otsusega saavutataks liiga väike puhtuse kasv või entroopia vähenemine. Vastasel juhul võib lisada hargnemisi seni, kuni igasse harusse jääb ainult mõni näide. Sellisel juhul on puu treeningandmetele liialt kohanenud ega suuda üldistuda uutele näidetele (vt ptk 5.4.9). Olles need reeglite lisamise üldised parameetrid (ehk hüperparameetrid) kindlaks määranud, loob otsustuspuu algoritm meile ise treeningnäidete abil otsustuspuu.

Joonis 5.8 illustreerib selle otsustuspuu loogikat. Et klassifitseerida kliente kahte gruppi, „ostja“ ja „mitte-ostja“, tuleb vastata uue kliendi kohta nendele küsimustele, alustades ülemisest hargnemispunktist.



5.8. Otsustuspuu hindamaks, kas inimesest saab ostja.

Puud loetakse ülevalt alla – peame esitama esmalt kõige üleval oleva küsimuse ja liikuma vastavalt vastusele kas vasakule või paremale harusse, kuni jõuame vastuseni (ehk puu „lehte“, harusse, mida me väiksemates osadeks ei jaga). Esmalt peame küsima potentsiaalse kliendi vanust. Kui ta on vanem kui 30, viib see meid vasakpoolsesse harusse ja peame esitama järgmise küsimuse: kas ta omab kõrgharidust? Kui ei, siis on tegu inimesega, kes tõenäoliselt ei osta meie toodet (vasakpoolne pupane „mitte-ostja“ leht). Vastasel juhul on vaja teada, kas tema sissetulek on suurem kui 1500 eurot kuus. Kui vastus on jah, siis on tegu potentsiaalse ostjaga, kui ei, siis see inimene klassifitseerub „mitte-ostjaks“. Mõtteharjutusena võite proovida parempoolse haru samal viisil sõnadega lahti seletada.

Ülesanne
Proovige joonisel 5.8 toodud otsustuspuu alusel ise klassifitseerida uus andmepunkt: 31-aastane kõrgharidusega mees, kelle sissetulek on 1600 eurot kuus. Õige vastuse leiad siit.

5.4.3. Otsustusmets

Otsustusmets (ingl k *random forest*) on masinõppe mudel, mis koosneb mitmest otsustuspuust. Erinevalt üksikust otsustuspuust, kus otsuseid tehakse ühe puu struktuuri põhjal, võtab otsustusmets kokku mitme puu ennustused, et teha lõplik otsus. Otsustusmets on parema üldistusvõimega ning suudab mudeldada keerulisemaid seoseid sisendite ja väljundite vahel kui üksik otsustuspuu. Üksik puu suudab kujutada ainult piiratut hulka seaduspärasid ja need ei pruugi olla piisavad kõigi näidete korrektseks klassifitseerimiseks või võivad kogemata peegeldada hoopis treeningandmetes sisaldunud juhuslikku müra. Kui mets koosneb omavahel erinevaid otsuseid sisaldavatest puudest, mis kõik proovivad lahendada sama ülesannet, on puude keskmine vastus müra ja juhuslikkuse suhtes vähem tundlik ning mudel töökindlam

(miks mitme ennustaja vastuste keskmistamine annab parema tulemuse, vt täpsemalt peatükist „Ansamblimine“).

Otsustusmetsa loomisel treenitakse iga puu erineva juhusliku valikuga treeningnäidetest. Kasutatakse *bagging*-meetodit ehk asendamisega juhuslikku valikut, milles sama näide võib esineda valimis mitu korda. Seega, iga puu näeb loomise käigus erinevaid näiteid ja ongi seetõttu teistest erinev. Erinevad näited sisaldavad ka erinevat müra ja kõik puud ei õpi kogemata samu vigu tegema. Et puid omavahel veel erinevamaks teha ja sundida erinevaid puid kasutama erinevaid lahendusviise andmete klassidesse jaotamisel, kasutatakse tunnuste juhuslikku valikut igas hargnemispunktis. Mudel saab igas sõlmes optimaalse hargnemiskriteeriumi leidmisel kasutada ainult juhuslikult valitud alamhulka tunnustest. Nii näiteks ei saa kõik puud alata sama otsusega, sest kõik puud ei näegi optimaalse otsuse leidmisel samu tunnuseid. Juhusliku andmete ja tunnuste valiku tulemusena koosneb mets mitmekesisest hulgast otsustuspuudest. Ennustus tehakse kõigi puude ennustuste keskmistamise (regressiooni ülesannetes, väljund-tõenäosuste keskmistamine klassifikatsioonis) või enamushääletuse teel (klassifikatsiooni puhul). Üksiku puu ebatäpsused või kallutatus kaotavad keskmistades oma negatiivse mõju.

Otsustusmetsa üks eelis on võime töötada suure hulga sisendtunnustega, millest osa võivad olla kasutud ja mitteinformatiivsed. Iga hargnemiskriteeriumi valimisel võib mudel vähem informatiivsed tunnused lihtsalt kasutamata jätta, need küll aeglustavad ja segavad õppimist, kuid ei riku seda täielikult. Seega piltlikult öeldes võite te sisendiks anda kõik oma suure tabeli veerud, ilma tunnuste valimise sammu läbi tegemata. Eeliseks on ka selle algoritmi üsna suur töökindlus ja vähene tundlikkus hüperparameetrite suhtes. Kasutaja saab küll valida puude arvu, puu suurima lubatava sügavuse, hargnemispunktide edasisest jagamisest loobumise kriteeriumid, otsuste valimise kriteeriumid (Gini, entroopia) jpm, aga kogukond on leidnud viisid, kuidas neid väärtusi vaikimisi valida ja mudel pole neile liiga tundlik. Seega paljudel juhtudel polegi vaja neid väärtusi puutuda, saab kasutada masinõppe tarkvara vaikimisi pakutud väärtusi, välja arvatud ehk puude hulk, sest mida rohkem puid, seda kauem mudeli loomine võtab. Lihtsuse ja töökindluse tõttu on otsustusmetsad leidnud rakendust mitmesugustes valdkondades, alates krediidiriski hindamisest kuni meditsiinilise diagnoosi ja kliendisegmentatsioonini.

On selge, et sadadest või tuhandetest puudest koosnevat metsa pole võimalik graafiliselt kujutades hoomata ja tehtud ennustuste tagamaid lihtsasti mõista. Lahendustes, mille puhul on oluline väljundmudeli mõistetavus inimese jaoks, võib esmalt luua otsustusmetsa ning hakata seejärel järk-järgult kasutatavate tunnuste hulka, puude hulka ja puude suurust vähendama, et jõuda lõpuks ainult ühest või mõnest puust koosnevad metsani, mida inimene suudab mõista, kuid mis ikkagi saavutab kasuliku täpsuse.

Otsustusmetsa puhul on üsna intuitiivne viis, kuidas hinnata erinevate tunnuste olulisust lõplike ennustuste tegemisel: arvutatakse kokku, kui suure puhtuse kasvu (Gini puhtust kasutades) või entroopia vähenemise igal tunnusel baseeruvad otsused kõigis

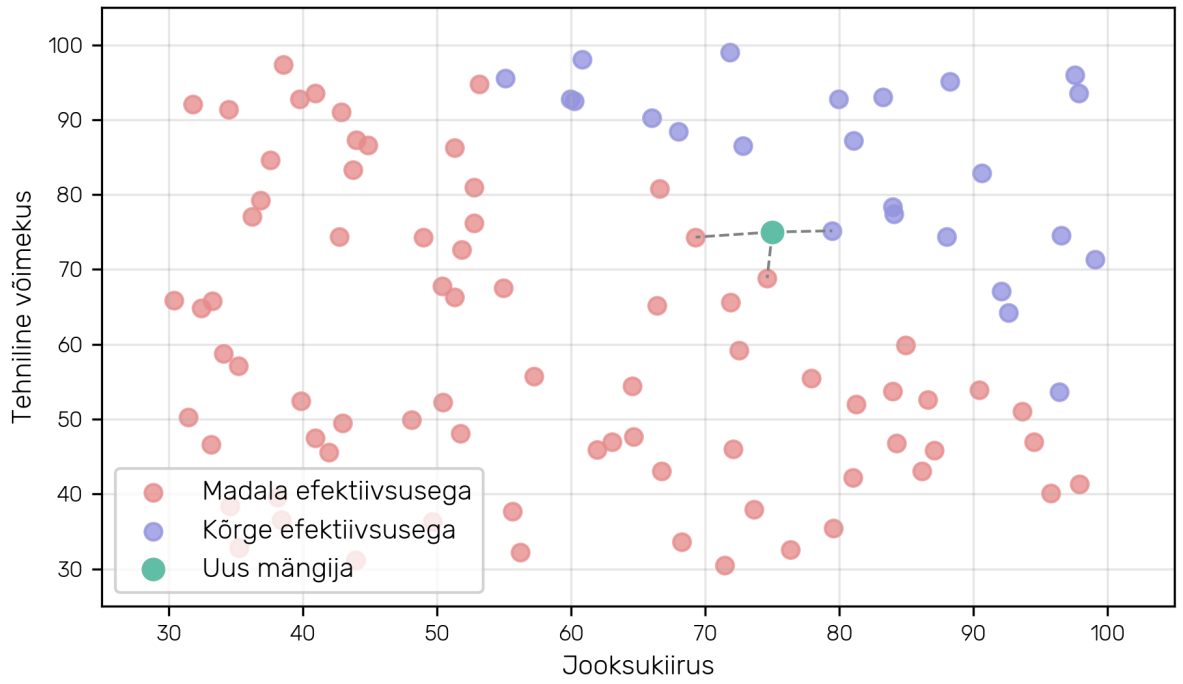
puudes kokku võetuna kaasa tõid. See võimaldab valida edasise analüüsi jaoks ainult informatiivsed tunnused või saavutada mingi üldise teadmise oma andmete kohta.

Otsustusmetsi saab kasutada mitte ainult klassifikatsiooni jaoks, lahendada saab ka regressiooniülesandeid. Sel juhul valitakse hargnemispunktides otsuseid selliselt, et igasse uude harusse edasi liikuvate näidete märgendite väärtused oleksid omavahel võimalikult sarnased, kasutades selle mõõdikuna tavaliselt keskmist ruutviga märgendite ja grupi keskmise vahel. Puude lehtedes tehtavad ennustused ongi treeningandmetes sellesse lehte jäänud näitete märgendite keskmine väärtus. Metsa ennustus on puude ennustuste aritmeetiline keskmine.

5.4.4. *K*-lähimad naabrid

K-lähima naabri (KNN) algoritm on masinõppe meetod, mis põhineb lihtsal ideel: me ei loo mudelit, vaid vaatame lihtsalt varasematest andmetest või andmebaasist otse järele, milliseid märgendeid sarnased näited on saanud. Eeldame, et sarnase märgendiga andmepunktid asuvad üksteise naabruses. Erinevalt teistest õppimisalgoritmidest, mis üritavad treeningfaasis mudeldada andmete tekkimise protsessi või tunnustevahelisi seoseid, kasutab KNN ennustuste tegemiseks treeningandmestikku oma algsel kujul. KNN liigitab iga uue andmepunkti selle *K* lähima naabri põhjal treeningandmete hulgas, kus *K* on kasutaja valitud täisarvuline hüperparameeter. Näiteks klassifitseerimisülesande puhul tagastatakse uuele andmepunktile ennustuseks enamasti lihtsalt kõige sagedamini esinev klass tema *K* lähima naabri seas (joonis 5.9). Alternatiivina võib lähemal asuvate naabrite märgendit tugevamalt arvesse võtta. Regressiooni puhul tagastatakse kas *K* lähima naabri keskmine märgendi väärtus, mediaanväärtus või mingil viisil kaalutud keskmine väärtus, võttes lähemal asuvate naabrite märgendit arvesse tugevama kaaluga.

KNN-i kasutamiseks on vaja valida arvesse võetavate naabrite arv ja kauguse definitsioon (eukleidiline, koosinuskaugus või muu), mis määratleb andmepunktide „läheduse“ kontseptsiooni. Teiseks on vajalik andmestiku eeltöötlus, näiteks omaduste standardimine (või muul viisil skaleerimine), et tagada kõigi omaduste võrdne arvesse võtmine. KNN ei saa väga hästi hakkama paljude sisendtunnustega, sest mida rohkem on dimensioone, seda kaugemal on keskmiselt üksteisest andmepunktid („mõõtmelisuse needus“, *curse of dimensionality*) ja on vaja väga palju treeningnäiteid, et igal võimalikul uuel näitel oleks mõnigi „lähedane“ naaber.



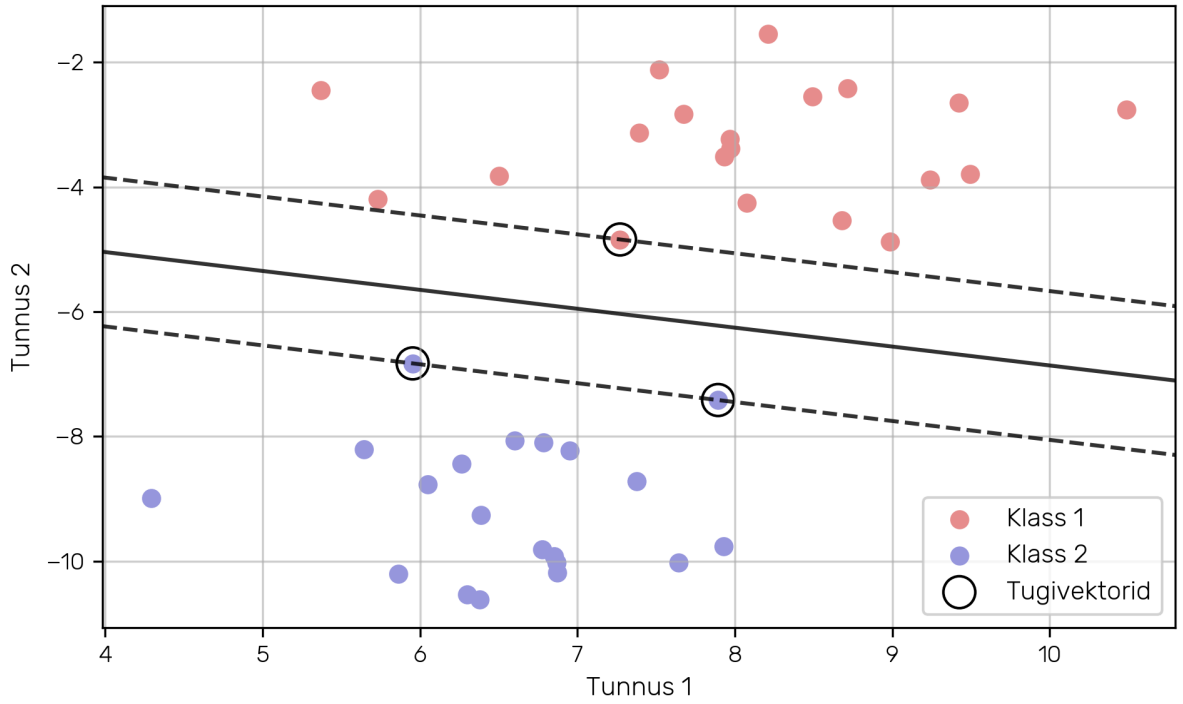
Joonis 5.9. 3-NN mudeli toimimise näide jalgpallurite efektiivsuse ennustamisel. *Teame iga mängija kohta tema jooksukiirust ja tehnilist võimekust. Varasemate mängude alusel oleme liigitanud hulga mängijaid (treeninghulga) kas väikese (punased punktid) või suure (sinised punktid) efektiivsusega mängijateks. Soovides uue mängija (roheline punkt) puhul ennustada, kas temast oleks platsil kasu, otsib 3-NN treeningandmestiku hulgast kolm tunnuste alusel sarnaseimat andmepunkti. Nende punktide märgendite alusel ennustatakse uue mängija kasulikkus. Lähtekood.*

Kokkuvõttes on KNN tänu oma lihtsusele paljudes masinõppeprojektides esimene algoritm, mida proovida. See ei anna küll enamasti väga häid tulemusi, aga seda on lihtne rakendada ja see annab aimu, kas andmetes üldse on reeglipära, mida keerulisema algoritmiga õppida. Samuti võib selle abil leida probleeme andmetes või ülesande püstituses – kui KNN ei anna paremat tulemust kui juhuslikult vastuseid andev mudel, siis puudub sisendtunnustes signaal märgendi kohta või on andmetega mingi probleem.

5.4.5. Tugivektormasinad

Tugivektormasinad (ingl k *support vector machines,* SVM) on mitmekülgsed masinõppe mudelid, mida kasutatakse nii klassifitseerimis- kui ka regressiooniülesannete (tugivektorregressioon, SVR) lahendamiseks. See algoritm oli 2000ndatel äärmiselt populaarne tänu võimele lahendada keerulisi klassifitseerimisprobleeme, kuid selle kasutus on tehisnärvivõrkude laialdase kasutuselevõtuga seoses viimasel kümnendil kahanenud. Erinevalt paljudest teistest masinõppemeetoditest, mis otsivad andmepunktide vahel seoseid, püüab SVM leida otsustuspiiri – hüpertasandi ehk kõrgedimensionaalse pinna, mis eraldab erinevate klasside andmepunkte (joonis 5.11). See tasand peaks võimalikult palju ühe klassi andmepunkte jätma endast ühele ja teise klassi punkte teisele poole, aga samal ajal maksimeeritakse ka andmepunktide kaugust

sellest pinnast. See aitab kaasa mudeli üldistusvõimele uute, nägemata andmete peal[13]. Andmepunktid, mis asuvad hüpertasandi lähedal ja mõjutavad selle kuju (tasand „hoiab neist teatud suunda“, et punkt jääks õigele poole, ning „hoiab neist ka eemale“), nimetatakse tugivektoriteks. Need andmepunktid määravad otsustuspiiri asukoha ja orientatsiooni, seega on mudeli nimi „tugivektormasin“.



Joonis 5.10. Tugivektormasin. *Tugivektormasin otsib otsustuspiiri, mis jätaks punasesse klassi kuuluvad punktid endast ühele ja sinisesse klassi kuuluvad punktid teisele poole. Kahemõõtmelise andmestiku puhul on see piir kujutatud sirgjoonena. Joone asendit mõjutavad ainult joonele lähedal asuvad punktid (joonisel ringiga märgitud). Lähtekood.*

SVM-i saab rakendada ka enama kui kahe klassi eristamiseks. Sel juhul on vaja iga klassi kohta leida üks-kõigi-vastu-otsustuspiir, mis eristab seda kõigist teistest klassidest, või otsustuspiir iga võimaliku klassipaari jaoks. SVM on väga tõhus suure mõõtmelisusega ruumides (palju sisendtunnuseid) ja seda saab kasutada ka siis, kui andmepunktid ei ole lineaarselt eraldatavad (joonega, tasandiga). Sellistes olukordades teisendatakse tugivektormasina treenimise eel esmalt algne tunnusteruum teatud kõrgema dimensionaalsusega ruumiks selliselt, et andmepunktid muutuksid uues ruumis paremini lineaarselt eraldatavaks. Selle teisenduse jaoks kasutatakse nn tuumafunktsiooni (ingl k *kernel, kernel trick*).

Tugivektorregressiooni (ingl k *support vector regression*, SVR) puhul ei piisa otsustuspiiri leidmisest, sest ennustada on vaja pidevaid väärtusi. SVR püüab hoopis leida funktsiooni, mis ennustaks treeningandmete märgendi väärtusi nii, et ennustuste vead oleksid väiksemad kui teatud eeldefineeritud väärtus (hüperparameeter). Samal ajal minimeeritakse mudeli keerukust, sest lihtsamad mudelid üldistuvad paremini. Vigu, mis on väiksemad kui eelseatud piir, enam pisemaks ei optimeerita, et mudel saaks

[13] Et otsustuspind on andmepunktidest võimalikult kaugel, teeb SVM-i vähem tundlikuks rünnakute ehk mudeli ära petmise suhtes. Ükski ühegi treeningnäitega väga sarnane sisend ei saa liigituda teise klassi.

keskenduda veel lahendamata suurematele vigadele. Keerulisemaid andmepunktide, mis funktsiooni kuju otseselt mõjutavad, nimetatakse tugivektoriteks.

SVM-i peamine eelis on võime töötada keerukate mustritega ja kõrge mõõtmelisusega andmekogumitega. Samal ajal võib SVM olla tundlik hüperparameetrite, näiteks marginaali suuruse (vähim kaugus otsustustasandi ja näidete vahel) ning tuumafunktsiooni valiku suhtes, mis nõuab andmeteadlaselt kogemust, hoolikat häälestamist, valideerimist.

Kokkuvõttes on tugivektormasinad üsnagi usaldusväärsed tööriistad, mis ei vaja loomiseks väga suurt treeningnäidete hulka. Need pakuvad lahendust mitmesugustele ülesannetele alates netikommentaaride klassifitseerimisest solvavateks ja mittesolvavateks kuni diagnoosi ennustamiseni meditsiinis. Kuigi otsustuspiirid pole alati lihtsa ja inimese jaoks mõistetava kujuga, on neid teatud juhtudel, eriti binaarse klassifikatsioon puhul, siiski võimalik interpreteerida.

5.4.6. Ansambelmeetodid

Ansambelmeetodid on nutikas tööriist, mis võimaldab parandada mudelite ennustustäpsust ja töökindlust. Nii nagu otsustusmets on täpsem ennustaja kui üksik otsustuspuu, võivad ka teised mudelite ansamblid saavutada parema üldistusvõime ja vähendada ülesobitumise riski.

Mudelite vigadesse panustavad kolm peamist allikat: kallutatus, dispersioon ja paratamatu müra. Kallutatus on mudelitüübi süsteemne võimetus teatud andmepunkte hästi ennustada, sest mudel ei suuda tegelikku andmetekke funktsiooni korrektselt kujutada või olemasolev andmestik ei luba seda teha. Dispersioon on mudelite õppimise tundlikkus mürale treeningandmetes, mis viib selleni, et õpitud mudeli ennustused testandmestiku näidete jaoks on juhuslikul määral ja juhuslikus suunas valed, sõltuvalt treeningandmestiku mürast, mis mudelit mõjutas. Paratamatu ja õppimatu müra on see osa testandmestiku märgendite väärtustest, mis ongi paratamatult juhuslik ning ennustamatu. Mitme ennustava mudeli kombineerimine ansamblisse võib vähendada mudeli vastuste kallutatust, dispersiooni või mõlemat.

Ansambelmeetodite juuridee on järgmine. Kui meil oleks viis 80% täpsusega mudelit, mis teevad kõik valideerimisandmestikul omavahel erinevaid vigu, siis oleks meil iga valideerimisnäite jaoks neli õiget ja üks vale ennustus. Lastes mudelitel enamushääletada, saaksime viie mudeli ansamblina palju võimekama ennustaja, kui seda on iga mudel üksikult. Selle idee keskmes on mudelite mitmekesisus. Kui kõik ansamblisse kuuluvad mudelid teeksid samu vigu, ei tooks nende kombineerimine mingit kasu. Seda mitmekesisust ei ole praktikas alati lihtne tekitada, mõned näited ongi oma olemuselt keerulisemad liigitada ja erinevad mudelid kipuvad ikkagi samadel näidetel eksima. Mitmekesisuse tekitamiseks võib näiteks teha ansambli iga mudeli eri tüüpi, treenida mudelid erineval alamhulgal andmetest või tunnustest või lisada juhuslikkust muul viisil.

Ansambelmeetodeid on peamiselt kolme tüüpi: võimendamine, *bagging* ja virnastamine.

- Võimendamine (ingl k *boosting*). Luuakse terve seeria ennustusmudeleid, kus iga järgnev mudel keskendub näidetele, mille eelmine mudel valesti klassifitseeris. Võimendamine parandab peamiselt mudeli kallutatust ja võimaldab õppida andma õigeid vastuseid ka haruldasemat tüüpi näidetele. Ansambel mitmest nõrgemast ennustusmudelist, mis on võimendamise abil üksteise järel õpetatud, võib olla märgatavalt võimekam ennustaja kui üks suur mudel või ansambel, mis koosneb mitmest samadel andmetel (ühtlase rõhuasetusega kõigile näidetele) loodud mudelist. Ansambli vastust ei saavutata võimendamise puhul lihtsa hääletuse teel, vaid igale mudelile määratakse mingi kaal, millega see ennustust mõjutab. Täpsed viisid, kuidas valesti ennustatud andmepunkte järgmise mudeli treenimisel prioriseeritakse ja kuidas mudelite tulemusi ennustamisel kombineeritakse, sõltuvad täpsest võimendamise algoritmist (nt AdaBoost, GradientBoosting).
- *Bagging* loob mitu sõltumatut mudelit, treenides igaühte veidi erineva andmekogumi peal, mis on saadud algsest andmestikust asendamisega juhusliku valimise teel. Mudelite ennustuste keskmistamine (regressiooni puhul) või enamushääletus (klassifitseerimisel) aitab tõhusalt ennustuste dispersiooni vähendada. Ka otsustusmetsad kasutavad *bagging*-meetodit, aga lisaks on igas hargnemispunktis otsuse valimisel kasutusel ainult juhuslik alamhulk tunnustest.
- Virnastamine (ingl k *stacking*) ühendab erinevate mudelite ennustused, kasutades teise taseme mudelit (metaõppijat, metamudelit). See teise taseme mudel õpib, kuidas kõige paremini kombineerida esimese tasandi mudelite ennustusi lõpliku ennustuse saamiseks. Selleks võib see mudel arvesse võtta ka andmepunktis sisalduvat infot, näiteks tuvastades, et mingit tüüpi andmetel tasub mudelit 4 kõige rohkem usaldada, mudelit 2 veidi vähem ning mudeleid 1 ja 3 pigem väga vähe. Virnastamise erijuhuks on mudelite segu (ingl k *mixture of experts*), mille puhul võetakse arvesse ainult ühe mudeli arvamust korraga. Luuakse väravmudel (ingl k *gating model*), mis valib, millist ekspertmudelit antud andmepunktil kasutada, ja teiste „ekspertide“ arvamus arvesse ei lähe.

Kokkuvõttes võib mudelite ansamblimine oluliselt parandada vastuste täpsust. Loogiliselt võttes on terve kogumi mudelite kasutamine arvutuslikult kulukam nii mudelite õpetamise kui kasutamise faasis. Siiski on saavutatud võit seda väärt. Populaarsel andmeteaduse võistluste korraldamise platvormil Kaggle osutuvad enamasti võitjaks just mudelite ansamblid. Üldjuhul aitab paljude mudelite ansamblimine tulemust võrreldes üksiku mudeliga vähemalt paari protsendi võrra parandada ja tihedas võistluses oleks rumal seda kasutamata jätta.

Metsad aastal 2024

Otsustusmetsad koosnevad puudest, mille oleme proovinud muuta mitmekesisteks, kasutades *bagging*-meetodit ja valides igas hargnemispunktis otsuseid ainult juhusliku alamhulga tunnuste seast. Olles avastanud võimendamis- ja virnastamismeetodid, oleks ju loogiline neid *bagging*-ideega kombineerida, et saada veel võimekam algoritm? Nii see tõesti on.

Kaggle'i andmeteaduse võistluste platvormil, kus erinevad organisatsioonid oma andmeid tuhandetel huvilistel analüüsida lasevad, on otsustuspuudel põhinevate algoritmide kasutamine tihti võidu võtmeks. Seda isegi ajastul, kui tehisnärvivõrgud trügivad kõikidesse valdkondadesse ja tunduvad alistamatud. Puud pole surnud, nende evolutsioon ja areng jätkub. Moodsad otsustuspuude metsad kasutavad õppimisse sisse ehitatud võimendamismeetodeid, mis teevad need konkurentsivõimeliseks suurte tehisnärvivõrkudega. Tuhat targal viisil üksteist võimendama õpetatud puud suudavad pakkuda konkurentsi ka suurtele ja keerulistele tehisnärvivõrkudele.

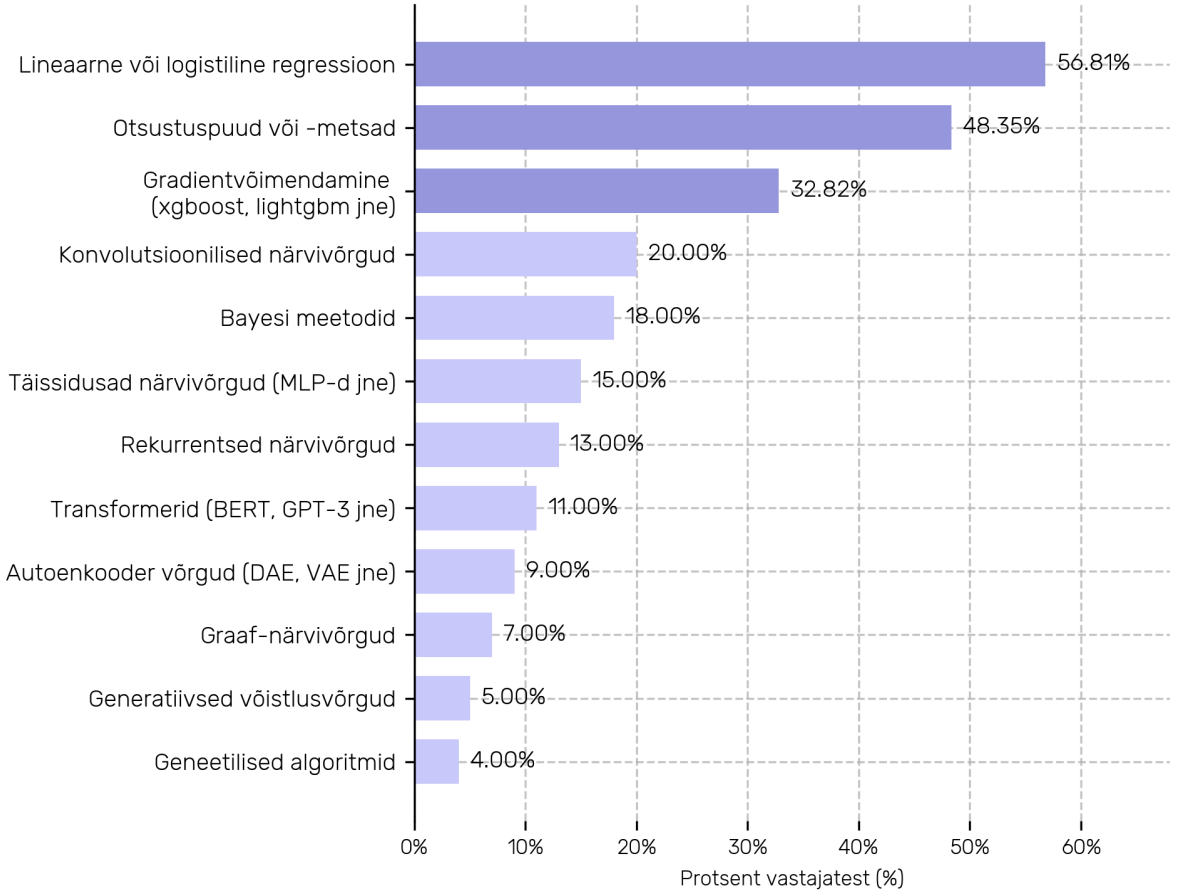
Järgnevalt tutvustame lühidalt mõnda populaarset puudel põhinevat algoritmi, mille rakendused leiate kindlasti ka oma masinõppe teegist (Pythonis sklearn, R-is Caret).

- ***Gradient boosting machines*** **(GBM)** ehk gradientvõimendamise masinad töötavad, lisades iteratiivselt uusi mudeleid, tavaliselt otsustuspuid, mis parandavad seni olemas oleva ansambli vigu. Erinevalt AdaBoostist (kõige klassikalisem võimendamisviis, mis suurendab seni ekslikult ennustustatud näidete kaalu järgneva mudeli treenimisel), õpib gradientvõimendamise puhul iga lisatav mudel seni olemas oleva mudeliansambli eksimusi parandama. Seega ansambli mudelite ennustusi ei keskmistata, vaid liidetakse, sest iga järgnev mudel täiendab senist mudelit, mitte ei tee iseseisvat ennustust. Ansambli eksimus (viga, kadu), mida me soovime minimeerida, on defineeritud mingi funktsiooni abil. Kaofunktsiooni gradient mudeli väljundite suhtes ütleb meile, mis suunas ja kui suurel määral mudeli väljundid muutuma peaksid, et vastused paremaks muutuksid. Igal andmepunktil arvutatud gradiendid läbi korrutatuna õpisammuga (kordaja, mis reguleerib iga üksiku võimendamissammu mõju) ongi ansamblisse lisatava mudeli ennustamise eesmärk (märgend, siht).
- **XGBoost** (*extreme gradient boosting*, ekstreemne gradientvõimendamine) on GBM-i edasiarendus. Sellele on sisse ehitatud meetodid otsustamaks, mida teha puuduvate väärtustega, ja suutlikkus tõhusalt kasutada olemasolevat riistvara (paralleelarvutused, vahemälu kasutamine), et mudel kiiremini õpetatud saaks. Need muudavad selle lahenduse kergemini kasutatavaks. Lisaks on arvesse võetud ohtu, et mudel ülesobitub, seatud teatud piirangud iga ansamblisse lisatava mudeli kaalule (vt regularisatsioon) ja iga mudeli loomiseks kasutatakse ristvalideerimist (vt ptk 5.3.10). Lisaks kasutatakse puude lõikamist – esialgu lubatakse puudel rohkem (sügavamaid, rohkemate otsustega) oksi kasvatada, hiljem eemaldatakse oksad, mis pole piisavalt kasulikud. Võib öelda, et see mudel kasutab kõiki võimalikke trikke, et saada hea ja üldistuv mudel ükskõik millistel andmetel (ka puuduvate väärtustega). XGBoost on äärmiselt edukas lähenemine ja on saavutanud palju võite andmestike analüüsimise võistlustel.
- **LightGBM** (kerge gradientvõimendamise mudel) on kohandatud suurte andmekogumite ja kõrge mõõtmelisusega andmete töötlemiseks. LightGBM kiirendab õppimist, valides igal õpisammul õppimiseks kõik suurte gradientidega andmepunktid ja ainult alamhulga ülejäänud andmetest. Puid kasvatatakse viisil, mis lubab okstel olla eri sügavusega. Samuti on leiutatud trikid hargnemispunktides otsuselävendite (ingl k *split points*) arvutuslikult efektiivsemaks leidmiseks ja kategooriliste sisendmuutujate tõhusamaks

kasutamiseks. Suurte andmemahtude puhul on tegu meetodiga, mida kindlasti proovida.

- **CatBoost** on Yandexi välja töötatud avatud lähtekoodiga algoritm, mis on kohandatud kategooriliste muutujatega töötamiseks. CatBoosti suudab automaatselt teisendada kategoorilised muutujad kasulikule esitusviisile (mitte enam üks-mitmest-esitusviis), lihtsustades seeläbi andmete ettevalmistamist. Ka CatBoostil on oma erilised viisid puid kasvatada ja ülesobitamisohtu vähendada, mis teevad selle piisavalt erinevaks teistest võimendamismudelitest, et ka seda mudelitüüpi oma andmetel katsetada. CatBoost on näidanud häid tulemusi väga eriilmelistel andmestikel eri valdkondadest.

Kui te loote ennustavat mudelit eesmärgiga saavutada maksimaalne ennustustäpsus, ei saa te võimendatud puudel põhinevatest algoritmidest mööda vaadata. Loetletud lähenemised on erinevad ja tulemuse maksimeerimiseks tasub proovida neid kõiki. Võimendatud puudest koosnevad mudelid olid aastal 2022 masinõppe enim kasutatavad „keerulised" mudelid (joonis 5.12). Ainult olemuselt lihtsad mudelid konkureerivad nendega kasutamissageduselt, tõenäoselt tänu oma tõlgendatavusele ja populaarsusele esimese baasmudelina, mida proovida.

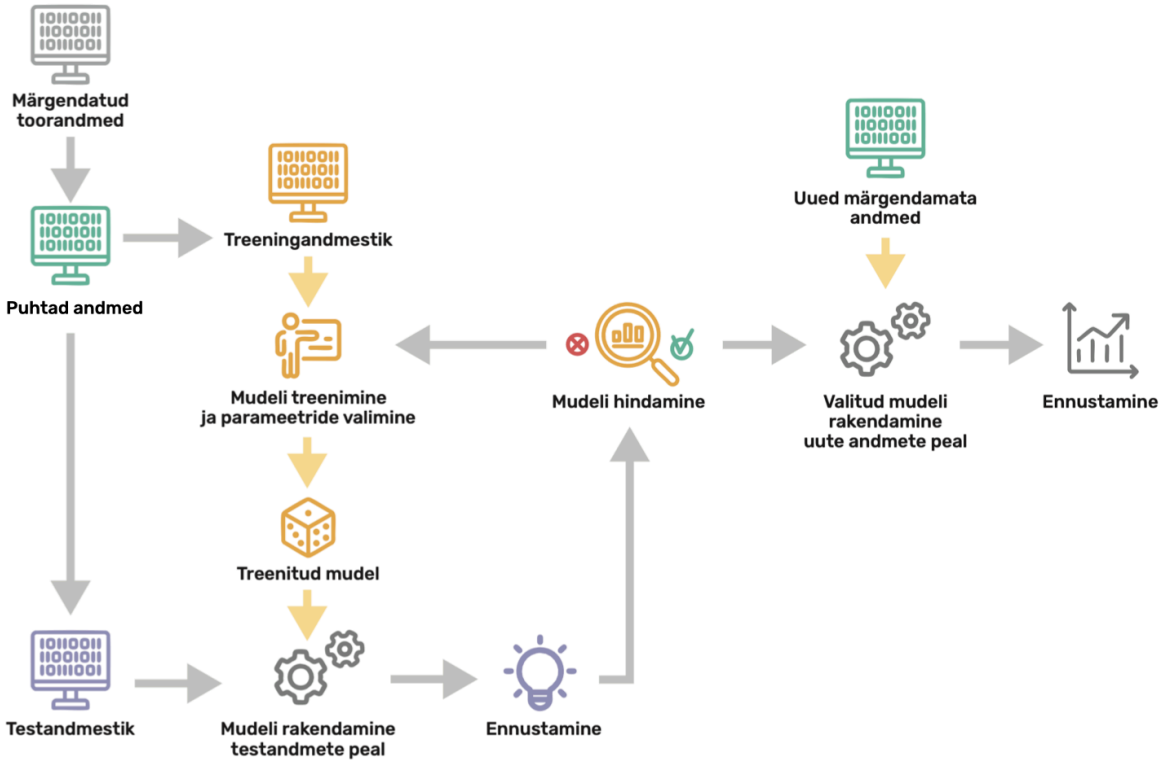


Joonis 5.11. ***Masinõppe meetodite kasutatavus.*** *Kaggle'i 2022. aastal korraldatud küsitluses osalenud andmeteadlaste vastused küsimusele „Milliseid mudelitüüpe te olete rakendanud?". Lineaarseid mudeleid ja otsustusmetsa on paljud vastanutest kindlasti kasutanud õpingute käigus või lihtsasti treenitava nõrga mudelina, millega järgnevaid mudeleid võrrelda. On märkimisväärne, et gradientvõimendamine on populaarsem kui esimene sügavõppe meetod*

(konvolutsioonilised tehisnärvivõrgud). Gradientvõimendamine on tugev meetod ja pigem kasutatakse seda ikka hea tulemuse saamiseks, mitte niisama võrdluseks. Lähtekood.[14]

5.4.7. Mudelite treenimise lihtsustatud töövoog

Juhendatud masinõppe mudeli loomise ja juurutamise üldist töövoogu saab kirjeldada joonisel 5.12 toodud skeemiga, mida hakkame nüüd etapi kaupa selgitama. Oluline on aga kogu asja juures märkida, et selle töövoo läbimine ei käi koskmudeli alusel, vaid esmalt tuleks kogu töövoog tööle saada ja siis vastavalt vajadusele erinevaid osi parandada.



Joonis 5.12. Juhendatud masinõppe lihtsustatud töövoog.

1. Kvaliteetsed andmed on hea mudeli loomise võtmekomponent, seega tuleb esimese etapina **toorandmed** enne nende kasutamist mudeli loomisel puhastada ja eeltöödelda.
2. Teine oluline etapp on **puhaste andmete** jaotamine **treeningandmestikuks** ja **testandmestikuks**. Tavaliselt kasutatakse treeningandmetena 80% olemasolevatest andmetest, kuid see sõltub muu hulgas olemasolevate andmete hulgast (vt nt ptk 5.3.10). Sellist jaotust on vaja, et treenimisprotsessi lõpus hinnata, kui hästi mudel seni nägemata andmetel ennustama õppis, ja vältida **ülesobitumist** (ingl k *overfitting,* vt ptk 5.3.9*)*. Ülesobitamine tähendab piltlikult öeldes juhtumit, kus mudel õpib treeningandmepunktid pähe ilma tegemata üldistusi, mis võimaldaksid ka uutele näidetele täpselt vastata.
3. Järgneb mudelitüübi valimine ja **mudeli treenimine**. Selles etapis õpib mudel andma treeningandmestikus olevatele sisenditele vastuseid, mis oleks võimalikult

[14] Joonis on tõlgitud, info pärineb: kaggle.com, litsents: Apache 2.0.

lähedased treeningandmestikus antud märgenditele. Mudel õpib, võrreldes oma ennustusi tegelike treeningandmetes olevate märgenditega ja **muutes ennast nii, et viga** (ennustatu ja tegeliku vahe) **väheneks**. Mudeli viis ennast muuta sõltub mudeli tüübist – milliseid matemaatilisi samme mudel sisaldab. Neid samme saab muuta, lisada ja eemaldada.

4. Treenitud mudeli võimalikult sõltumatu **hindamine** on äärmiselt oluline. Hindamisel võtame testandmestiku sisendtunnused, laseme mudelil ennustada märgendid ning arvutame ennustuste ja tegelike märgendite keskmise erinevuse (keskmise vea) või mõne muu headuse mõõdiku (vt ptk 5.3.8). Kasutades mudeli hindamiseks treenimisel mitte kasutatud näiteid, testandmestikku, anname mudelile üldistusvõimele ausa hinnangu. Kui meie testandmestik on esinduslik, siis töötab mudel sellel keskmiselt sama hästi kui andmetel, mida mudel hakkab töötlema lahenduse juurutamisel. Kui mudeli keskmine eksimus treeningandmestikul on märgatavalt väiksem veast testandmestikul, ütlemegi, et mudel on treeningandmetele ülesobitunud.
5. Olenevalt treeningandmete hulgast, mudelitüübist ja õppimiseks kasutatud reeglitest võib tõesti juhtuda, et ka pärast õppimist on mudeli ennustused testandmestikul üsna halvad. Peale ülesobitumise on võib põhjuseks olla, et on kasutatud valet mudelitüüpi või õppimisalgoritmi või et ongi võimatu antud tunnustest märgendit ennustada. Sel juhul on vaja minna tagasi ja valida uus lähenemine. Loodetavasti piisab mudelitüübi või selle seadistuse muutmisest ja uusi andmeid koguma ei pea. Igal juhul on mudeli loomine iteratiivne protsess, milles ei korrata ainult punkti 3, vaid kindlasti ka 4 ning tihti ka 1 ja 2.
6. Kui testandmestikul saavutatakse hea tulemus, võib mudeli kasutusele võtta. See tähendab mudeli rakendamist uutele andmetele, mille märgendit me veel ei tea, aga tahame ennustada. Kui testandmestikul olid mudeli ennustused head ja testandmestik pärines samast jaotusest, oli esinduslik, siis usaldame mudeli ennustusi piisavalt, et neile ärilisi või muid otsuseid rajada.

Me nimetame kirjeldatud töövoogu lihtsustatud töövooks, sest teatud juhtudel on vaja andmed hoopis kolmeks osaks jagada: treening-, valideerimis- ja testandmestikuks (tavaliselt vahekorras 80%, 10%, 10%). Kui mudelitüüpi ei valita eelneva teadmise põhjal, vaid katsetatakse suurt hulka mudelitüüpe ja nende seadistusi, siis on vaja treeningandmestikul loodud mudeleid hinnata ning neist hilisemaks rakendamiseks parim valida. Selle jaoks ongi eraldatud valideerimisandmestik – andmestik, millel me mudeleid otseselt ei treeni, vaid võrdleme mudeleid nende vahel valimise eesmärgil. Testandmestik peaks alati jääma mudeli loomise ja valimise protsessist täiest kõrvale, et pakkuda kogu protsessi viimase sammuna võimalikult ausat hinnangut mudeli võimekusele mudeli jaoks täiesti uutel andmetel.

5.4.8. Mudeli headuse mõõdikud

Nagu juhendatud õppe töövoos nähtud, soovime me enamasti **hinnata, kui hästi** meie **mudel uutel (seni nägemata) andmetel töötab**. Lõppkokkuvõttes huvitab meid just võimekus märgendeid korrektselt ennustada andmetel, mille märgendite väärtusi me veel ei tea, kuid teada tahame: keemilised ühendid, mille toksilisust me veel ei tea, patsiendid, kelle haigusriski soovime ennustada, uued kinnisvaraobjektid, mille lõplikku

müügihinda soovime ette aimata. Treeningandmete märgendeid me juba teame, nende uuesti ennustamine ei anna meile majanduslikku kasu ega muud teadmist. Me teame ka testandmestiku märgendeid, aga kui meie testandmestik on esinduslik, siis pole vahet, kas võtame uusi näiteid päriselust, kus me ei tea nende märgendeid, või võtame neid testandmestikust. Mudeli ennustustäpsus testandmetel on hea hinnang mudeli võimetele, enne kui seda reaalselt rakendama hakkame.

Mudelite hindamiseks on palju erinevaid mõõdikuid. Millist mõõdikut kasutada, sõltub esmalt sellest, kas tegu on klassifikatsiooni- või regressioonimudeliga. Teisena tuleb arvesse võtta, mida mudeliga pikas perspektiivis saavutada tahetakse, kuidas seda rakendada plaanitakse. Mudelid võivad teha eri tüüpi vigu ja päris elus võivad erinevad vead olla erineval määral ohtlikud või halvad.

Klassifikatsioon

Klassifikatsiooni puhul eristatakse tehtud mudeli õigeid vastuseid ja vigu omavahel. Teatud klassi vaatepunktist on tõsipositiivsed (TP) need näited, mis ennustati vaadeldavasse klassi ja märgendi järgi ka tegelikult kuulusid sinna. Valepositiivsed (VP) on ennustused, mille mudel tegi vaadeldavasse klassi, aga mis sinna tegelikult ei kuulunud. Tõsinegatiivsed (TN) on näited, mida mudel ei määranud vaadeldavasse klassi ja mis sinna ka ei kuulu. Valenegatiivsed (VN) on näited, mille mudel määras mõnda muusse klassi, kuid mis kuuluvad tegelikult vaadeldavasse klassi. Enamik levinud klassifikatsioonimudelite headuse mõõdikuid rajanevad nende nelja õnnestumise ja vea tüübi sageduste mõõtmisel.

On oluline märgata, et mudeli tegelikud väljundid on arvud, mitte kategoorilised klassiennustused. Klassifikatsioonimudeli puhul on neid arve sama palju kui omavahel eristatavaid klasse, need arvud on 0 ja 1 vahel ja neid tõlgendatakse mudeli enesekindlusena vastava klassi vastuseks ennustamisel. Mudeli ennustuseks loetakse enamasti kõrgeima enesekindluse saanud klass, aga on ka teisi variante. Binaarse klassifikatsiooni puhul pole harv see, et väljundi positiivseks ennustuseks lugemise lävend ei ole 0,5. Mitmese klassifikatsiooni puhul on võimalik defineerida lisaks vastus „mitte ükski klassidest“, kui ühelegi klassile vastav väljund seatud enesekindluse lävendit ei ületa. Seega sõltub lõppkokkuvõttes masinõppe mudeli TP, TN, VP ja VN arv lisaks optimeeritud parameetritega masinõppemudelile veel ka kasutaja valikutest mudeli rakendamisel. See kasutaja vabadus mudelit optimistlikumaks või rangemaks teha on väga vajalik, sest erinevates ülesannetes võivad olulisemad olla eri tüüpi õnnestumised või eksimused. Näiteks ennustades ravimikandidaatide toksilisust (binaarne, positiivne klass on „toksiline“), on loogiline seada lävend madalale, et turvaliseks (negatiivne klass) liigitataks ainult need ained, mille mittetoksilisuses on mudel väga kindel.

Veamaatriks on mudeli õnnestumiste ja eksimuste esitus tabeli kujul. See võtab kokku kõik ennustused, mida mudel tegi (kasutades seatud lävendit). Veamaatriksi diagonaalil asuvad täpsed ennustused (või nende proportsioon) ja ülejäänud väärtused aitavad mõista, milliste klasside vahel vigu tehti. See võib anda vihjeid probleemide kohta andmetes või mudelis. Siiski pole veamaatriks „mõõdik“, vaid vigade illustratsioon. Mõõdikult eeldame me mudeli headuse väljendamist ühe arvuna, mis võimaldaks näiteks

erinevaid mudeleid selgelt võrrelda või mõõdiku mingi väärtuse saavutamist projektiplaanis eesmärgiks seada.

Klass	0	1	2	3	4	5	6	7	8	9
0 Lennuk	**580**	5	4	0	1	2	0	0	6	2
1 Auto	3	**570**	0	0	0	0	0	0	2	25
2 Lind	4	0	**530**	25	12	15	8	2	4	0
3 Kass	0	0	18	**512**	8	40	10	5	2	5
4 Hirv	1	0	10	5	**550**	2	15	12	3	2
5 Koer	0	0	5	40	3	**510**	5	30	0	7
6 Konn	0	1	3	10	8	2	**570**	0	3	3
7 Hobune	0	0	0	5	10	35	0	**540**	0	10
8 Laev	7	15	2	0	2	0	1	0	**570**	3
9 Veok	2	25	0	2	0	1	0	5	3	**562**

Tabel 5.3. Veamaatriks piltide klassifitseerimisel kümnesse kategooriasse CIFAR10 andmestikul. *Read vastavad tegelikele märgneditele, iga rea summa on 600, sest igat klassi pilte oli testandmestikus 600. Veerud vastavad mudeli ennustustele. Iga kast loendab näiteid, mille puhul oli tõene rea nimele vastav märgend ja ennustus oli veerule vastav. Diagonaalil on korrektsed ennustused. Ülejäänud kastid annavad aimu, milliseid pilte mudel omavahel rohkem segi ajas, näiteks autosid ja veokeid.*

Tabelis 5.4 loetleme ja defineerime tuntuimad klassifikatsoonimudelite edukuse mõõdikud.

Analüüsi tüüp	Mõõdik	Seletus
Klassifikatsioon	Õigsus $\frac{TP+TN}{TP+TN+FP+FN}$, $\frac{\text{õigete ennustuste arv}}{\text{kõigi ennustuste arv}}$	Binaarses klassifikatsioonis lähevad õigete ennustustena arvesse nii TP-d kui ka TN-d. Mitmese klassifikatsiooni puhul panustab iga õige ennustus ühe tõsipositiivse murru lugejasse. Mitte-ennustatud valede klasside tõsinegatiivsust valemi lugejas arvesse ei võeta.
Binaarne klassifikatsioon	Täpsus $\frac{TP}{TP+VP}$	Õigete vastuse osakaal mudeli poolt positiivse klassina ennustatud näidete hulgas. Teisiti öelduna, millisel määral me saame usaldada mudeli positiivseid ennustusi.
Binaarne klassifikatsioon	Saagis $\frac{TP}{TP+VN}$	Mudeli poolt korrektselt klassifitseeritud positiivse klassi näidete osakaal kõikidest positiivse klassi näidetest. Teisiti öelduna, kui suure hulga positiivseid näiteid mudel tuvastada suudab.
Mitmene klassifikatsioon	Keskmine täpsus	Igale klassile arvutatakse täpsus alguses eraldi, käsitledes seda klassi positiivse klassina ja kõiki teisi klasse negatiivsena. Klasside täpsused keskmistatakse mingil viisil – kas klasside esinemissagedusi arvesse võttes või mitte.
Mitmene klassifikatsioon	Keskmine saagis	Igale klassile arvutatakse saagis eraldi, käsitledes seda klassi positiivse klassina ja kõiki teisi klasse negatiivsena. Klasside saagised keskmistatakse mingil viisil – kas klasside esinemissagedusi arvesse võttes või mitte.
Klassifikatsioon	F1-skoor $\frac{2 \cdot saagis \cdot täpsus}{saagis + täpsus}$	F1-skoor hindab mudeli kõrge täpsuse ja kõrge saagise koos saavutamise võimekust. Mängides läbi kõik võimalikud enesekindluse lävendid positiivse klassi ennustamiseks, võime me leida lävendi, mis annab kõrgeima F1-skoori ehk parima tasakaalu täpsuse ja saagise vahel. Max_F1 on levinud mudeli headuse mõõdik. Mitmese klassifikatsiooni puhul raporteeritakse klassideülene F1-skoor, mis võib vastavalt kasutusjuhule olla arvutatud kas klassijaotust arvesse võttes või mitte.
Binaarne klassifikatsioon	ROC kõvera alune pindala (AUROC)	Mängides läbi kõik võimalikud enesekindluse lävendid positiivse klassi ennustamiseks, saame igal väärtusel arvutada saagise (ehk tõsipositiivsete määra) ja valepositiivsete määra ehk osakaalu negatiivsetest näidetest, mis ekslikult positiivseks ennustati. Joonistades nende kahe mõõdiku väärtuste paarid (erinevatel lävenditel) graafikule, saame joone, mille alune pindala ongi meid huvitav mõõdik AUROC. Ideaalse mudeli AUROC on 1, juhuslikke vastuseid andva mudeli oma 0,5. Seda mõõdikut saab tõlgendada ka kui tõenäosust, et mudel annab juhusliku positiivse näite

		puhul kõrgema positiivse klassi enesekindluse kui juhusliku negatiivse näite puhul. Et AUROC kaalub kõiki võimalikke enesekindluse lävendeid ega ole tundlik klasside esinemissageduse suhtes andmestikus, on see levinud kui universaalne binaarse mudeli ennustusvõime hindamise mõõdik.
Binaarne klassifikatsioon	Täpsus-saagis kõvera alune pindala (AUC-PR)	Mängides läbi kõik võimalikud enesekindluse lävendid, saab joonistada täpsus-saagis-graafiku, mis kirjeldab nende kahe üsna intuitiivselt mõistetava ja laialt kasutatava mõõdiku omavahelist tasakaalu. Siiski märgake, et täpsus sõltub klasside esinemissagedusest andmestikus ja seega sõltub see mõõdik andmestikust, millel me seda arvutame. Saadud headuse hinnang pole universaalne, vaid sõltub testandmestikust. Kui aga testandmestiku klassijaotus vastab klassijaotusele mudeli hilisemal rakendamisel, võib AUC-PR olla isegi informatiivsem mõõdik mudeli hindamiseks meie spetsiifilise kasutusjuhu jaoks kui AUROC, mis on klassijaotusest sõltumatu.

Tabel 5.4. Klassifikatsiooni edukuse hindamise mõõdikud.

Täpsuse ja saagise puudus mõõdikuna on, et ekstreemsetel juhtudel saame need maksimaalselt heaks ka tegelikult mittekasuliku mudeli puhul. Kui kuulutada pimesi kõik näited positiivseks, saame ju 100% saagise, aga mingist täpsusest me enam rääkida ei saa. Samuti, kui märgime positiivseks ainult tühise osa näidetest, milles tõesti täiesti kindlad oleme, võime saada 100% täpsuse, aga olematu saagise. Seetõttu rakendatakse tihti mõõdikute kombinatsioone ja tasakaalukamaid mõõdikuid, nagu F1-skoor.

On oluline märgata, et mõned neist mõõdikutest on tundlikud mudeli hindamiseks kasutatava andmestiku klassijaotuse suhtes. Näiteks kui binaarse klassifikatsiooni andmestikus on väga vähe positiivseid ja palju negatiivseid näiteid, siis on oht ennustada arvuliselt üsna palju valepositiivseid võrreldes tõsipositiivsetega, tuues seega täpsuse madalale. Ka mudel, mis on positiivsetel näidetel 100% täpne ja ennustab ainult 1% negatiivsetest näidetest positiivseteks, võib kallutatud jaotusega andmestikul saada väga madala täpsuse. Seetõttu võibki olla huvipakkuv kasutada mõõdikuid, mis hindavad vigade määra, mitte arvu. Saagis on tõsipositiivsete määr ehk osakaal kõigist tegelikult sellesse klassi kuuluvatest näidetest. Saab defineerida ka valepositiivsete määra, tõsinegatiivsete määra ja valenegatiivsete määra. Need mõõdikud on suhtarvud ega sõltu esinemissagedusest. Kui me ei tea, milline saab olema klassijaotus mudeli rakendamisel, tasub hinnata mudeli headust ka klassijaotuse suhtes mittetundlike mõõdikute alusel. Kui aga on enam-vähem teada, millised andmed kasutamise käigus olema saavad, on täiesti mõistlik kasutada inimese jaoks kergemini tajutavat täpsuse mõõdikut.

Kokkuvõtteks: olenevalt mudeli lõplikust kasutusjuhust ja tingimustest peab andmeteadlane valima ka mudeli andmeteaduslikuks hindamiseks võimalikult sobiva mõõdiku. Kas tähtis on palju teatud tüüpi näiteid üles leida (aktsepteerides

valetuvastusi) või hoopis oma ennustustes võimalikult täpne olla (aktsepteerides valenegatiivseid ehk tuvastamata jätmisi)?

Regressioon

Regressiooni puhul on mudeli väljundiks reaalarv ja märgendiks samuti reaalarv, mingit lisakeerukust lävendite seadmise või ennustuste tõlgendamisel pole. Testandmetel tehtud ennustusi saab otseselt märgenditega võrrelda. Kõige intuitiivsem mõõdik selliste mudelite headuse hindamiseks on **keskmine absoluutviga**, mille tähendus on iseenesest mõistetav. Levinud on ka **keskmine ruutviga** ja selle ruutjuur ehk **ruutkeskmine viga**. Eksimuste ruutu võtmine tähendab, et suurtel eksimustel on keskmisele suurem mõju. Ruutkeskmise vea eelis keskmise ruutvea ees on, et see on samades ühikutes kui ennustatav väärtus ja lihtsamini tõlgendatav, selle suurust saab ennustatava tunnuse väärtustega võrrelda.

Levinud mõõdik on ka determinatsioonikordaja ehk R^2, mis võrdleb ruutvigade keskmist suurust märgendite dispersiooniga.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2},$$

kus y_i on andmepunkti *i* märgend, $\hat{y}_i$ on andmepunktil *i* tehtud ennustus ning $\bar{y}$ märgendite keskmine väärtus.

Determinatsioonikordaja justkui ütleb, kui palju on meie mudel parem nullmudelist, mis lihtsalt ennustaks keskmist väärtust. Öeldakse ka, et R^2 näitab, kui suure osa märgendites olevast varieeruvusest mudel ära seletada suudab. Seda mõõdikut kasutatakse enim lineaarsete mudelite hindamisel.

Projekti elutsükli vaade

Kui oleme mudelit hinnanud, kuid pole tulemusega rahul, peame pöörduma tagasi eelmise etapi juurde ja valima kas mingi teise masinõppe algoritmi või muutma mudeli parameetreid.

Mudelite arendamine ja optimeerimine on sageli ajamahukas, sest nõuab paljusid katsetusi ja kordusi. Vajadusel korratakse nii andmete kogumist (märgendatakse andmeid juurde), ettevalmistamist, mudelite loomist kui ka hindamist (paremal testandmestikul). Keerulisemates projektides (suured andmed ja mudelid) võib üks katsetus, andmestike kohandamine ja mudeli treenimine võtta päevi või isegi nädalaid. Näiteks võib juhendamata masinõpe osutuda ajakulukamaks kui juhendatud õpe, sest andmetes ei ole ette antud kindlat eesmärki (märgendit), mille suunas optimeerida. Sageli on vaja rohkem katsetusi, et avastada peidetud mustrid, mis lubavad andmeid kasulikul viisil struktureerida. Ka juhendatud õppe ja näiliselt lihtsamate algoritmide

puhul, nagu otsustusmetsad, tuleb arvestada korduvate katsetustega, et leida tõesti kõige parem seadistus, mis tulemust veel 0,5% täpsemaks teeks.

Kui me oleme oma mudeli andmeteadusliku hindamise tulemusega rahul, tuleb mudelit hinnata ka ärilise kasulikkuse aspektist (vt ptk 2.5 ja ptk 7). Seejärel võib **mudeli juurutada** ja hakata seda kasutama uute märgendamata andmete peal, et teha vajalikke ennustusi.

5.4.9. Ülesobitamine

Täpsust on muidugi võimalik mõõta ka treeningandmete põhjal. Me ei treeni (näiteks ei lisa otsustuspuule hargnemispunkte) mitte alati nii kaua, kuni meil on 100% täpsus treenimiseks kasutatud näidete alusel. Seega on mõttekas mõõta nii treeningtäpsust (täpsust treeningandmete põhjal) kui ka valideerimistäpsust. Kui need kaks arvu on väga erinevad, siis ütleme, et mudel on ülesobitunud.

Mudel võib olla ka alasobitunud (ingl k *underfitting*). See ei ole ülesobitumise otsene vastand ega tähenda, et valideerimistäpsus on treeningtäpsusest suurem (ka seda võib juhtuda, aga harva). See tähendab hoopis, et mudel pole suuteline isegi treeningandmete põhjal häid tulemusi saavutama. Selle põhjuseks võib olla näiteks asjaolu, et mudel pole piisavalt võimas, ei suuda piisavalt täpseid (piisavalt keerulisi) reegleid õppida. Võib ka juhtuda, et teatud algoritmid suudavad õppida ainult teatud tüüpi reegleid, näiteks suudavad leida ainult lineaarseid seoseid ega suuda tuvastada, kui märgend sõltub mingist tunnusest eksponentsiaalselt. Alasobitunud mudel annab ebatäpseid tulemusi nii treening- kui ka valideerimisnäidete pealt ja on seega kasutu. Me ei saa seda kasutada uute päriselust tulnud märgendamata andmete märgendamiseks. Mudeli alasobitumisele on lihtne vasturelv: tasub proovida võimsamat mudelitüüpi, mis suudab keerukamaid seoseid leida, ja treenida mudelit kauem. Ainuke mure ongi see, et liiga võimas mudel jällegi ülesobitub. Tuleb leida kompromiss mudeli võimekuses. Sobiva mudeli võimsus on seoses andmete hulgaga: vähestel andmetel suur mudel ülesobituks, ent suurtel andmetel lihtne mudel enam paremaks ei lähe, selle optimaalse kuju leidmiseks piisab vähemast. Tihti kasutatakse lähenemist, et kui mudeli täpsus treeningandmestiku kasvades enam ei parane, siis võiks liikuda keerulisema mudeli kasutamise juurde.

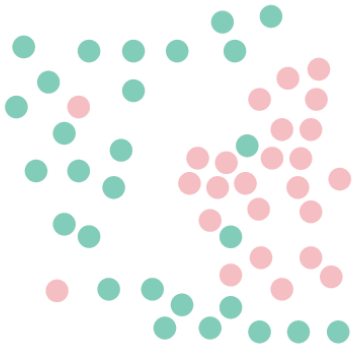
Nagu ennist mainisime, õpivad eri algoritmid erinevalt ja tuvastavad erinevaid seoseid. Seega on oluline ka algoritmi valik – mõni algoritm leiabki lihtsamini just need seosed, mis tegelikult andmetes peituvad. Milline algoritm millisele andmestikule sobib, pole midagi iseenesestmõistetavat ja on üsna tunnetuslik. Võiks öelda, et see on üks loomingulisi elemente andmeteadlase töös, tema kunst.

Ülesobitumise näide

Kasutame ülesobitumise näitena andmestikku, kus igal näitel on kaks arvulist tunnust. Kuna tegu pole reaalelulise andmestikuga, siis nimetame neid lihtsalt tunnuseks 1 ja tunnuseks 2. Märgend on kategooriline – kas „sinine“ või „punane“. Mudeli eesmärk on leida valem või reeglite süsteem (olenevalt mudelitüübist), mis võimaldaks tunnuste põhjal täpselt ennustada, millist värvi on punkt. Seda valemit või reeglistikku on võimalik

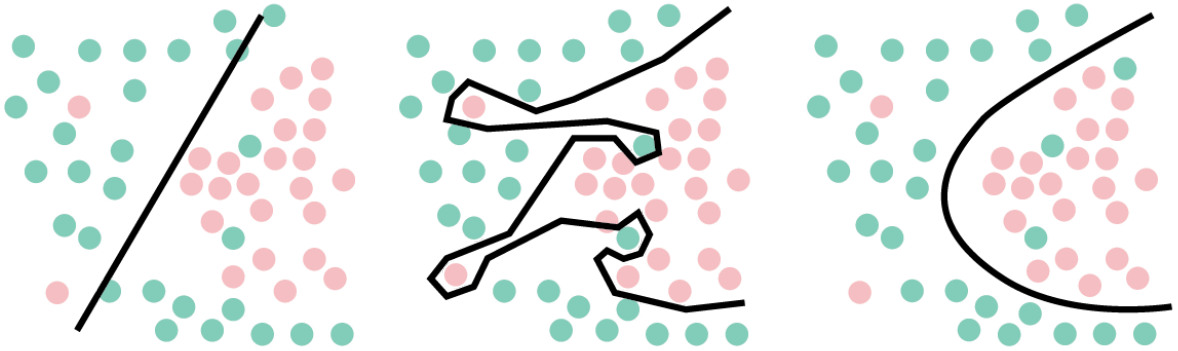
ka joonisel joonena kujutada, seega võime öelda, et otsime joont, millest ühele poole jäävad punased ja teisele poole sinised punktid. Oleme andmestiku juba jaganud treening- ja valideerimishulgaks.

Treenimiseks kasutatav andmestik on visualiseeritud joonisel 5.13. Selles andmestikus on mõned veidi imelikud punktid – punased siniste vahel ja vastupidi. Need võivad tuleneda valesti sisestatud tunnustest, valesti sisestatud märgenditest või lihtsalt andmed ongi sellised – alati polegi eri tüüpi objektid ideaalselt eristatavad.



Joonis 5.13. Treeningandmestik. *Iga punkt on üks näide, mille tunnuseks on tema asukoht ja märgendiks tema värv.*

Nende näidete põhjal on treenitud kolm mudelit (joonis 5.14). Esimene neist kuulub väga lihtsasse mudelitüüpi, mis õppimise käigus lihtsalt leiab parima sirgjoone, mille abil punkte eraldada. See mudel pole treeningpunktidel väga täpne, aga pole ka kohutav. Teine mudel on palju keerulisem mudelitüüp, mis suudab õppida keerulise murdjoone, mis eraldab treeningpunktid ideaalselt. Kolmas mudel on samuti üsna lihtne mudelitüüp, mis on võimeline õppima kõverjooni. See mudel ei erista treeningpunkte ideaalselt, aga leiab mõistliku kompromissi mudeli lihtsuse ja täpsuse vahel.

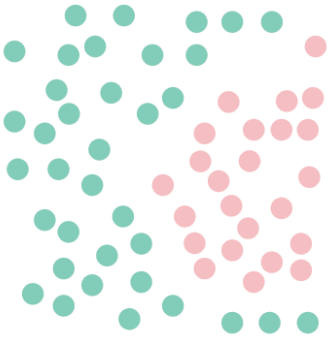


Joonis 5.14. Kolm treenitud mudelit. *Oleme treeninud kolm mudelit, mis on tüübilt erinevad. Vasakul on äärmiselt lihtne, keskel äärmiselt keerukas ja paremal keskmise keerukusega mudel.*

Milline mudel tundub teile valideerimisandmete põhjal kõige täpsem?

Vaatame nüüd nende mudelite täpsust valideerimisandmete järgi. Need andmed on illustreeritud joonisel 5.15. Seekord pole andmetes teistest omasugustest kõrvale kalduvaid punkte – see võis juhtuda juhuslikult (meie valideerimishulka sattusid

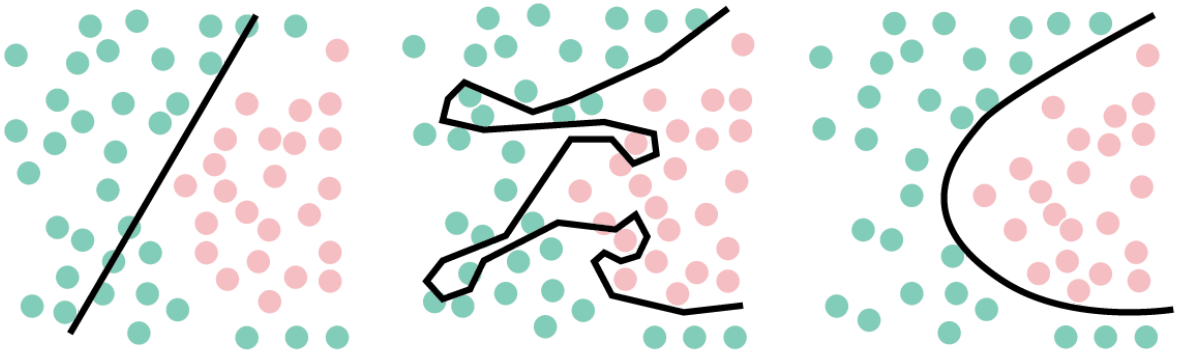
mitte-erilised punktid) või sellepärast, et andmeid märgendades oldi hoolikamad või need kontrolliti üle.



Joonis 5.15. Valideerimisandmed.

Rakendame neile andmetele kolme varem õpitud mudelit (joonis 5.16). Esimene mudel töötab umbes sama halvasti kui treeningandmetega. Tegu on alasobitunud mudeliga, mis lihtsalt ei suudagi klasside eraldamiseks vajalikku reeglit ära õppida. Teine mudel ei klassifitseeri enam punkte ideaalselt, nagu ta treeningandmetega seda tegi. Tegu on ülesobitunud mudeliga, mis õppis ära kõik treeningandmestiku nüansid, näiteks juhuslikult kõrvale kalduvad või valesti sisestatud andmed. Nagu nüüd näeme, oli see liig: keerukas murdjoon tekitab valideerimisnäidete põhjal vigu pigem juurde.

Viimane mudel on aga valideerimisandmete põhjal absoluutselt täpne. Treeningandmete põhjal täpsuses kompromisse tehes leidis see keskmise keerukusega mudel õige tasakaalu täpsuse ja lihtsuse vahel. Tihti ongi hea valideerimistäpsuse saavutamiseks püüda leida võimalikult lihtne mudel, mis saavutaks üsna hea täpsuse treeningnäidete põhjal. Mida lihtsam, seda tõenäolisemalt on mudel õige, nagu ütleb ka kuulus Ockhami habemenoa printsiip.



Joonis 5.16. Kolm mudelit valideerimisandmetele rakendatuna.

Selles ülesandes oli kolmanda mudeli võime õppida kõverjooni siniste ja punaste punktide eraldamiseks kasulik. Samas oleks ka keeruline mudel suutnud üsna täpselt vajalikku joont kujutada, kui see poleks liialt üksikuid punkte õppima hakanud. Siit peegeldub veel kaks reeglit, mis võivad aidata ülesobitumist vältida. Esiteks, kui kasutada mudelitüüpi, mis sobib just neile andmetele, on lootus õppides õige mudelini jõuda suurem. Teiseks, ka keeruliste mudelitega saab õppida ilma ülesobitumata – tuleb

kas vähem samme optimeerida, et mudel ei jõuaks kõigi detaile ära õppida, või rohkem andmeid kasutada, et üksikud mürased näited oma mõju kaotaksid. Enamiku õppimisalgoritmide puhul on võimalik enne otsustada, kui kaua ja kui põhjalikult treeningnäidete varal õppimine peaks toimuma.

Ülesobitumine on mingil määral paratamatu, sest lõppkokkuvõttes mudel ikkagi õpib ja saab kasutada ainult seoseid, mis on olemas treeningandmetes. Ka miljoni reaga treeningandmestik ei pruugi katta kõiki maailmas ette tulla võivaid olukordi ning alati võib valideerimisandmetes (ja mudelit päriselus rakendades) leida midagi uut ja mudeli jaoks ootamatut.

Lisamärkus: eluline paralleel ülesobitamise näitele
Kui punaste ja siniste täppide eraldamine jäi teie jaoks liiga abstraktseks, kujutlege, et punased punktid on päikselised päevad ja sinised on vihmased päevad. Tunnusteks on eelmise päeva temperatuur ja õhuniiskus. Seega püüame tänaste andmete järgi ennustada homset ilma. Ütleme, et mudel on treenides näinud haruldast näidet, kus 20,0-kraadisele ja 20,0% õhuniiskusega päevale järgneb suur sadu, aga on ka palju sarnaste ilmastikuoludega näiteid (temperatuur 19–21 kraadi, õhuniiskus 19–21%), millele järgneb hoopis kuiv päev. Mitte-ülesobitunud mudel üldistaks kõigi nende sarnaste näidete põhjal: pole vahet, kui mõni näitaja mõne kümnendiku võrra erineb. Küsides selliselt mudelilt, milline ilm järgneb 20,0-kraadilisele ja 20,0% õhuniiskusega päevale, vastaks see, et kuiv, sest vahemikus 19–21 kraadi see enamasti nii oli. Kuigi treeningandmetes oli näide sajusest päevast, ongi tegelikult palju tõenäosem, et sellisele ilmale järgneb kuiv päev.

Ülesobitunud mudel seevastu ei üldistaks, vaid ehitaks mingi keerulise reegli, mis võimaldaks 20,0 kraadi ja 2,.0% õhuniiskuse puhul vastata „sajab“, mis sest et kõik ümbritsevad, ainult veidike erinevad näited (näiteks 19,9 kraadi ja 20,1%) viitavad kuivale homsele. Seetõttu võrreldaksegi ülesobitunud mudelit treeningandmete pähe õppimisega, ilma üldistusi tegemata.

Lisamärkus: ülesobitamise kurb näide
Planeerides Fukushima tuumajaama, uuriti, mis on tõenäosus, et sealses piirkonnas juhtub maavärin magnituudiga 9. Tunnusteks võeti mingil ajaperioodil toimunud väiksemad maavärinad ja märgendina ennustati nende põhjal, mis on sellise seismilise aktiivsusega piirkonnas tõenäosus väga tugevaks maavärinaks, mille magnituud ületab 9. Kahjuks ülesobitati see mudel ja saadi (vale) vastus, et selline maavärin toimub korra 13 000 aasta jooksul. Tagantjärele tarkusena saame öelda, et mitte-ülesobitunud mudel oleks andnud vastuseks kord 300 aasta jooksul. Täpse mudeli puhul oleks sellise maavärina riski peetud piisavalt suureks, et tuumajaam veel tugevamaks ehitada. (Ingliskeelne allikas)

5.4.10. Ristvalideerimine

Sageli ei ole ette teada, milliste hüperparameetritega (mudeli tüüp, suurus) mudelid sobiks kõige paremini antud kontekstis. Samuti on sageli andmeid liiga vähe, näiteks kliinilisse uuringusse võib olla kaasatud ainult paarsada patsienti, kellele on tehtud mingi kulukas uuring. Siis oleks hea testida, milliste parameetritega mudelid töötaks tõestatult kõige paremini.

Ristvalideerimine on andmete kasutamise, mudelite loomise ning mudelite jõudluse hindamise viis, mille käigus jaotatakse andmed korduvalt eri viisidel treening- ja valideerimisandmestikuks. Andmete korduvalt jaotamise eesmärk on, et iga andmepunkt oleks ühel jaotamise kordadest valideerimisandmete hulgas ja me saaksime treeninghulka sattunud andmete põhjal loodud mudeli jõudlust sellel andmepunktil hinnata. Nii osaleb iga näide mingil jaotamise korral mudelite headuse hindamises ja kui andmestik sisaldab haruldasi näiteid, ei saa need juhuslikult valideerimisest või testimisest välja jääda. Samuti suureneb tulemuste statistiline tugevus, sest meil on rohkem valideerimistulemusi, mille põhjal statistikuid arvutada. Võrdluseks: kui meil on liiga väike testandmestik, siis võib juhtuda, et sellesse on sattunud ainult lihtsad näited ja me ülehindame mudeli jõudlust. Samuti ei ole näiteks kümnel testandmepunktil arvutatud statistika väga usaldusväärne. Ristvalideerimise puhul tuleb siiski märgata, et igal jaotamise korral luuakse uus, treeningandmete erinevuse tõttu teistsugune mudel ja kõik valideerimistulemused pole sama mudeli väljund. See piirab võimalikke statistilisi analüüse (kas kõik vea väärtused on samast jaotusest?) ja tulemuste tõlgendamist.

Peamiselt kasutatakse *K* korda ristvalideerimist (ingl k *K-fold cross-validation*), mille puhul jagatakse andmed juhuslikult *K* gruppi ja igal korral jäetakse üks neist gruppidest valideerimisandmeteks. Ülejäänul treenitakse mudel ning seda mudelit valideeritakse välja jäetud grupil ja arvutatakse vigade suurus. Kui kõik *K* korda on läbi tehtud, arvutatakse valideerimisvigade keskmine või muud statistikud, mis peegeldavad valitud lähenemise võimet üldistuda uutele andmetele. Siinkohal ei saa me rääkida ühe kindla mudeli üldistumisvõimest uutele andmetele, sest meil on *K* erinevat mudelit. Küll aga saame öelda, kas selline statistiline analüüs või mudeli tüüp ja treenimise protseduur viib hea üldistumisvõimega mudeliteni. Praktikas kasutatakse kõige enam viis korda ja kümme korda ristvalideerimist.

Kui ristvalideerimisel saab teada, milliste parameetritega mudelit ehitades saab parimaid tulemusi, võib nende valitud parameetritega veidi julgemalt ehitada uue mudeli, kasutades siis juba kõiki sisendandmeid korraga. Muidugi tuleb tulevikus jälgida, kas see treenitud mudel ka päriselt häid tulemusi annab.

Eriti väheste andmete korral või kui soovitakse hinnata mudeli tundlikkust üksikutele andmepunktidele, võib kasutada ka jäta-üks-välja-ristvalideerimist. Sel juhul korratakse analüüsi sama palju arv kordi, kui on andmestikus näiteid, sest igal korral jäetakse valideerimisandmestikku ainult üks näide. Nii kasutatakse treenimisel maksimaalsel hulgal andmeid, mis omakorda tähendab maksimaalselt head mudelit. Samas on kõigi korduste läbitegemise järel tulemuseks ikkagi üks valideerimisvea väärtus andmepunkti kohta ja statistiline tugevus lähenemise üldistusvõime kohta säilib. Puuduseks on

muidugi see, et tuleb treenida palju mudeleid ning see on arvutuslikult kulukas ja aeganõudev.

Ristvalideerimine hüperparameetrite ja mudelitüübi otsingul

Ideaaljuhul on meil lõpmata palju andmeid ning saame jagada oma andmestiku treening-, valideerimis- ja testandmeteks. Treeningandmestik on piisavalt suur, et mudelid näevad eri tüüpi näiteid ja õpivad hästi üldistuma. Otsides parimat mudelitüüpi ja hüperparameetreid, treenime paljusid (sadu) mudeleid. Valideerimisandmestik on piisav, et lubada neid erinevaid lähenemisviise statistilise olulisusega võrrelda ning meil parima mudelitüübi ja hüperparameetrid valida. Testandmestik on piisavalt suur, et näiteks statistilise olulisusega näidata, et loodud parim mudel töötab paremini kui mõni varem kasutusel olnud mudel.

Ent kui andmeid on vähe, on meie kolm andmestikku väikesed. Treenimisel ei õnnestu alati häid mudeleid luua, sest andmeid on vähe. Valideerimisel mudeleid omavahel võrreldes kasutame liiga vähe näiteid ja hindamistulemused on mürased. Võib juhtuda, et mõni mudel oli juhuslikult just neil vähestel valideerimisnäidetel täpne, kuid tegelikult väga hästi ei üldistu. Seega valime parimaks võib-olla tegelikult mitte kõige parema mudeli. Testandmestik on väike ega võimalda niikuinii midagi statistiliselt olulist väita.

Sel juhul kasutatakse hüperparameetrite otsingul ristvalideerimise abi. See võimaldab rohkematel andmepunktidel osaleda valideerimisel ja kui kasutada suurema kordade arvuga ristvalideerimist (või lausa jäta-üks-välja-ristvalideerimist), siis on ka iga mudel treenitud rohkematel andmetel. Sel juhul peaks ikkagi esmalt eraldama testandmestiku ja lõpliku hindamise tegema sel andmestikul. Aga millist meie K mudelist me siis testandmetel testima peaks? Võib testida kõiki ning raporteerida selliste mudelite keskmise edukuse ja selle variatiivsuse. Aga võib ka ristvalideerimise alusel leitud kõige paljulubavamaid hüperparameetreid kasutades treenida veel ühe, viimase mudeli kõigil treening- ja valideerimisandmetel. Sel mudelil on parim võimalus olla edukas, see on treenitud kõigil võimalikel andmetel leitud heade seadistustega. Siiski pole selle loomise käigus puututud testandmeid ja hindamistulemus on õiglane.

Kahjuks tuleb tunnistada, et praktikas jäetakse ristvalideerimist hüperparameetrite otsingul kasutades tihti testimine ära. Arvatakse justkui, et K korda valideerimine on piisav, andmaks hinnangut mudelite täpsusele uutel, seninägemata andmetel. Praktikas võib näha, et testandmetel on mudelite edukus keskmiselt siiski veidikese väiksem, võibolla paar protsenti. Põhjuseks on valikunihe – me valime hüperparameetrite otsingu käigus kõige madalamat valideerimisviga tegevat mudelitüüpi. Kui meil on kümme tegelikult sama head mudelitüüpi, aga omavahel veidi erinevad, siis valideerimisandmetel on mõnel neist juhuslikult parem tulemus kui teistel, sest valimisse sattusid selle mudeli jaoks sobivad näited. Väike valideerimisviga ei tulene ainult selle mudelitüübi ennustusvõimest uutel andmetel, vaid ka juhuslikust õnnest. See õnn ei kordu testandmetel ega päris elus ette tulevatel andmetel. Seega selline valideerimistäpsus ülehindab mudeli tegelikku võimekust.

5.5 Stiimulõpe

Stiimulõpe on protsess, kus, sarnaselt juhendamata õppega, õpib algoritm andmetest ilma seotud märgendita. Stiimulõppe korral saab algoritm positiivset või negatiivset tagasisidet vastavalt pakutud lahenduste edukusele. See on algoritmide klass, mis on mõeldud otsuste tegemise optimeerimiseks. Algoritm peab tegema otsuseid ja otsused kannavad tagajärgi, näiteks hoiavad isejuhtivat sõidukit teel või juhivad selle kraavi. Aja jooksul õpib stiimulõppe algoritm katse-eksitusmeetodil tegema järjest paremaid otsuseid, mis maksimeerivad kasu. Kasu võib olla defineeritud erinevalt, see võib tähendada, näiteks, õnnetusteta isejuhtimist, suuremat müügisummat, kasvavat klikkide arvu lehel jms. Tänapäevase stiimulõppe all mõeldakse enamasti sügavat stiimulõpet (ingl k *deep reinforcement learning*) ehk stiimulõppe mudeleid, mis kasutavad sügavaid tehisnärvivõrke. Sügav stiimulõppimine nõuab väga suurt hulka andmeid (katsetusi, korduseid) ja sellega on seotud palju tehnilisi probleeme, mis muudavad selle tehnoloogia rakendamise keeruliseks.

Stiimulõpe sarnaneb inimese õppimisprotsessiga, kus inimene õpib katse-eksitusmeetodit kasutades. Vead aitavad inimestel õppida, kuna nendega on seostatud karistus (nt kulu, ajakaotus, kahetsus, valu). Samuti saab õppida õnnestumistest, proovides neid edaspidi korrata. Seda tagasisidet kasutades õpib inimene, milline tema tegevus on edukas ja milline mitte. Üheks näiteks on jalgrattaga sõitma õppimine. Inimene üritab rattaga sõita, aga kukub ja saab haiget (negatiivne tagasiside), järelikult tegi ta midagi valesti ja peab oma tegevust muutma. Kui rattasõit õnnestub probleemideta (positiivne tagasiside), siis tuleb meelde jätta, millised liigutused viisid selle tegevuse õnnestumiseni.

Stiimulõppe abil loodud mudelid pole mingil viisil erilised ega erine juhendatud õppega loodud mudelitest. Need koosnevad ikkagi matemaatilistest sammudest ja mudelite kuju on ikka samasugune. Erinev on ainult mudelite sees olevate õpitavate parameetrite leidmise viis ehk õppimisalgoritm.

Stiimulõppe rakendused on ettevõtluses praegu piiratud, peamiselt suure hulga kulukate simulatsioonide vajamise tõttu – me ju ei taha, et mudel katsetab ja eksib päris elus. Siiski on maailma suurimad ettevõtted juba hakanud neid mudeleid juurutama. Näiteks, suuremad finantssektori ettevõtted on juba mõnda aega kasutanud kauplemise ja kapitali suurendamiseks masinõppe algoritme ja mõned neist, näiteks JPMorgan, proovivad ka stiimulõpet rakendada. Väidetavalt kasutab Google serverite jahutamise juhtimiseks stiimulõppel baseeruvat lahendust ja on olnud ka teadustöid arvutustööde optimaalse jaotamise kohta serverite vahel, kasutades stiimulõppega õpitud mudeleid.

Näide: stiimulõpe ettekirjutavas analüütikas
Stiimulõpe on juba oma olemuselt otsuseid langetav, sest stiimulõppe mudeleid optimeeritakse, lastes neil teha otsuseid ja andes seejärel neile tagasisidet (tasu, ingl. k. reward), kas valitud otsus osutus kasulikuks või kahjulikuks. Seega need mudelid ongi loodud ise otsustama, mitte andma inimesele mõistetavaid

väljundeid või ennustusi mingite aspektide kohta. Selliste süsteemide puhul, mis langetavad otsuseid ilma viisita nende otsuste tagamaid mõista, peab kasutaja aga olema üsna kindel mudeli usaldusväärsuses.

Seetõttu on stiimulõppel põhinevad lahendused veel üsna haruldased. Peamiselt IT-sektoris on siiski mõned paljulubavad rakendused. Näiteks arvutusressursside jagamine programmide vahel arvutusklastrites toimub tavajuhul inimese kindlaks määratud reeglite alusel, aga on näidatud, et stiimulõppel põhinev süsteem suudab töid serverite vahel paremini jagada. Piltlikult öeldes saab nii sama masinate hulgaga ja sama ajaga rohkem arvutusülesandeid lahendatud.

Enesekontrolli küsimused

1) Mida tähendab George Boxi kuulus tsitaat „Kõik mudelid on valed, aga mõned on kasulikud“? Mis mõttes on mudelid valed? Kuidas saab midagi valet olla kasulik?

2) Selgitage hii-ruut testi kasutamise eesmärki ja põhimõtet. Millistel andmetel seda kasutatakse? Mis on selle testi nullhüpotees?

3) Kirjeldage ANOVA eesmärki ja põhimõtet. Millised on ANOVA eelised ja piirangud võrreldes t-testiga?

4) Millised on juhendamata masinõppe peamised ülesanded ja kuidas need erinevad juhendatud masinõppe ülesannetest?

5) Olete ettevõtte andmeteadlane ja teie ülesanne on analüüsida klientide ostukäitumist, et leida mustreid ja teha ennustusi tulevaste ostude kohta. Tooge näited, milliste küsimuste puhul kasutaksite juhendamata ja milliste puhul juhendatud masinõpet. Kirjeldage, milliseid meetodeid täpselt kasutaksite ja miks.

6) Miks on oluline, et ansambelmudelis kasutatavad mudelid oleksid üksteisest erinevad? Tooge näide ansambelmeetodi rakendamisest ja analüüsige, kuidas mudelite mitmekesisus võib mõjutada lõpptulemuse täpsust.

7) Selgitage, miks on otsustusmets võimsam mudel kui lineaarne regressioonimudel. Milliseid keerulisemaid ülesandeid suudab otsustusmets lahendada, mida lineaarne mudel ei suuda? Tooge näiteid olukordadest, kus tuleks eelistada otsustusmetsa, ja põhjendage oma vastust.

8) Oletame, et treenisite masinõppe mudelit andmestikul, mis sisaldab klientide käitumisandmeid ja nende ostude ajalugu, et ennustada tulevasi oste. Treenimisetapis saavutas mudel väga suure täpsuse, aga kui hakkasite seda reaalses kasutuses rakendama, ei andnud see enam häid ennustusi. Selgitage, mis võiks olla selle probleemi põhjuseks.

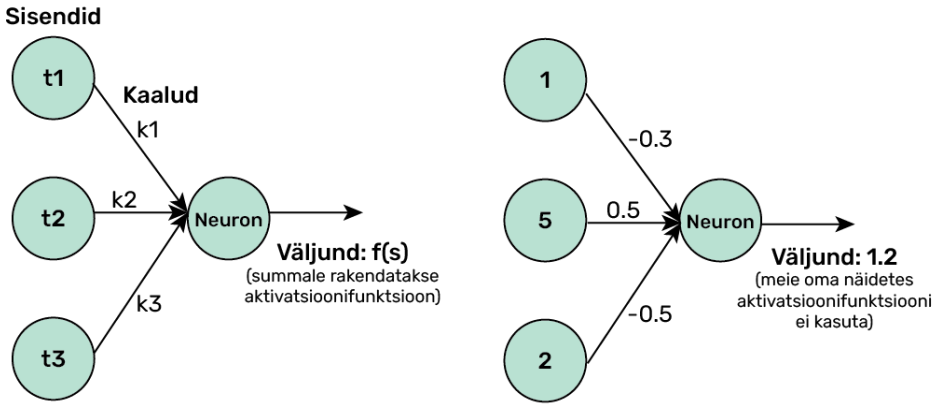
9) Pangad saavad tavaliselt uue laenu saamiseks sadu tuhandeid taotlusi. Iga taotlus sisaldab informatsiooni mitme tunnuse kohta, lihtsuse huvides kasutame selliseid tunnuseid nagu palk, vanus ja kas taotleja on väikelapse vanem. Oletame, et oleme loonud mudeli, mis ennustab, kas taotleja on usaldusväärne klient ehk kas ta maksab laenu tagasi. Oleme alltoodud tabelis kujutanud viit testandmepunkti. Nende märgendid on antud viimases veerus. Võrreldes ennustusi ja märgendeid (ehk tunnuse sihtväärtust), arvutage, mis on mudeli õigsus, täpsus ja saagis neil andmetel.

Palk	Kas on väikelapse vanem	Vanus	Mudeli ennustus	Õige märgend
4000	EI	31	JAH	JAH
1100	JAH	28	EI	EI
2500	EI	47	EI	JAH
1800	EI	56	JAH	JAH
1500	JAH	34	EI	JAH

6. Tehisnärvivõrgud ja sügavõpe

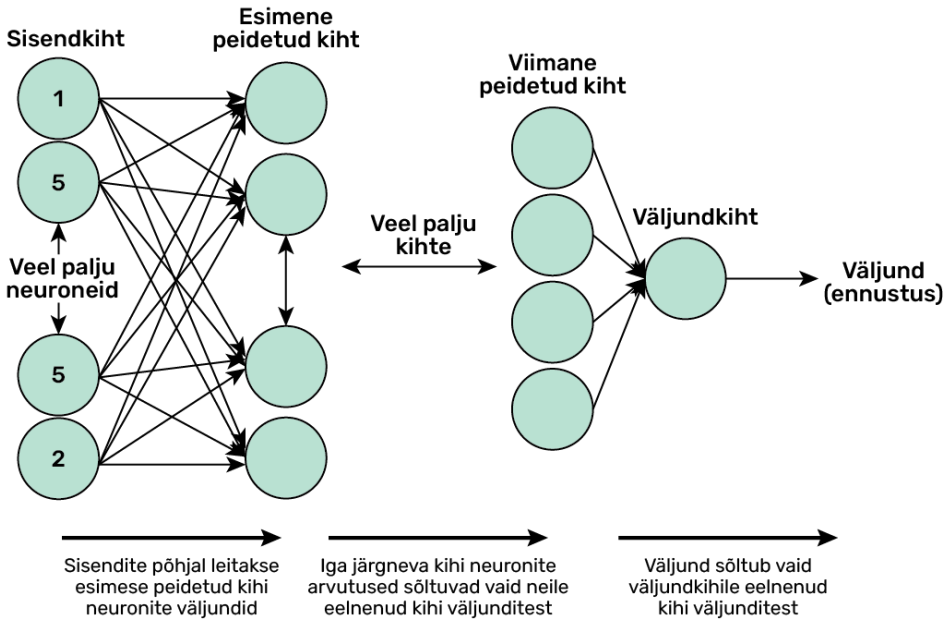
6.1 Sissejuhatus, põhitõed

Omaette masinõppe algoritmide grupi moodustavad **tehisnärvivõrgud** ehk **sügavõpe** (ingl k *artificial neural networks*, *deep learning*). Tehisnärvivõrgud on üks masinõppe mudelite tüüp, see tähendab, et ka nemad koosnevad lihtsalt arvutuslikest sammudest, mida tuleb üksteise järel teha. Tehisnärvivõrgud on inspiratsiooni saanud päris ajurakkude ehk neuronite võrgustikest ajus ja koosnevad samuti „neuronitest". Tehisneuronid (ingl k *artificial neurons*) püüavad imiteerida seda, kuidas päris neuronid infot töötlevad ehk kuidas neuronid üksteiselt signaale vastu võtavad ja neile reageerivad (joonis 6.1). Nagu päris neuronid, saavad ka tehisneuronid signaale teistelt neuronitelt ja reageerivad neile signaalidele teatud viisil.



Joonis 6.1. Tehisneuron. *Ühes tehisneuronis tehtavad arvutused.* ***Vasakul:*** *üldine definitsioon ja valemid. Neuron saab sisendiks teiste neuronite väljundid (t1, t2, t3). Need väljundid korrutatakse läbi kaaludega (k1, k2, k3) ja liidetakse. Saadud arvule rakendatakse veel teatud aktivatsioonifunktsioon f, mis muudab seda mingil viisil, kuid väljundiks on siiski lihtsalt üks reaalarv.* ***Paremal:*** *näide ühes neuronis tehtud arvutustest, kui neuroni sisenditeks on (1; 5; 2) ja kaaludeks (−0,3; 0,5; −0,5) ning aktivatsioonifunktsiooni ei kasutata.*

Tehisnärvivõrgu neuronid on alati, juba üle 50 aasta, üsna sarnased, uuendusi on tehtud ainult aktivatsioonifunktsiooni alal ja see võib erinevates võrkudes erinev olla. Peale neuronite arvu **erinevad tehisnärvivõrgud peamiselt neuronite omavaheliste ühenduste võrgustiku poolest**. Enamasti koosnevad tehisnärvivõrgud paljudest (tuhandetest sadade miljoniteni) tehisneuronitest, mis on organiseeritud eraldi **kihtidesse** (ingl k *layer*), mis on omavahel mingi loogika järgi ühendatud (joonis 6.2). Neid kihte võib ühes närvivõrgus olla tuhandeid, kuigi lihtsamate ülesannete lahendamiseks võib piisata mõnest kihist. Kihtide omavaheliste ühenduste võrgustikku nimetatakse **võrgu arhitektuuriks**. Terminid **sügavõpe** (ingl k *deep learning*) ja sügavad närvivõrgud tulevadki sellest, et närvivõrke liigitatakse muu hulgas ka nende kihtide arvu ehk **võrgu sügavuse** järgi.



Joonis 6.2. Tehisnärvivõrgu arhitektuur. *Joonisel on kujutatud kõige varasemat ja lihtsamat võrgu arhitektuuri – täissidusat võrku. Selles võrgus on neuronikihid reastatud üksteise järel. Iga kiht saab sisendeid ainult endale eelnevalt kihilt ja iga kihi neuronite väljundid on sisendiks järgmisele kihile. Seega, alustades sisendkihist, liiguvad arvutused alati väljundi suunas ja seda võrku liigitatakse seetõttu ka pärilevivõrguks (ingl k* feedforward neural network*). Lisaks on sellel võrgul omadus, et iga naaberkihi vahel on kõik-kõigile-ühendused neuronite vahel, mis annabki sellele nime täissidus võrk. Sama kihi neuronite vahel ühendusi ei ole.*

Õppimine tehisnärvivõrgus toimub ühendustugevuste ehk kaalude muutmise teel. Kõige levinum viis kaalude muutmiseks on **gradientlaskumise algoritm**. Iga kaalu jaoks arvutatakse ennustusvea tuletis selle kaalu suhtes ehk gradient, mis näitab, millises suunas seda kaalu muutma peaks, et ennustusviga väheneks. Seejärel muudetaksegi seda kaalu veidikene vea vähenemise suunas. Kaalude muutmine gradientide alusel on tehisnärvivõrkude optimeerimisel 99% juhtudest kasutatav optimeerimisviis. Võrgu arhitektuuri muutmine on haruldane õppimise viis, kuid ka sellel on omad rakendused – näiteks juba treenitud võrgult vähemtähtsaid neuroneid või ühendusi eemaldades saame väiksema ja arvutulikult tõhusama võrgu. Tehisnärvivõrkudes toimuvate arvutuste kohta saab lähemalt lugeda tehisintellekti algkursuse materjalidest.

Kuna õpitavad parameetrid on kaalud, võrreldakse mudelite omavahelist keerukust just õpitavate parameetrite arvu võrreldes. Sarnaste võrgutüüpide puhul, mida rohkem on õpitavaid parameetreid, seda suurem on teoreetiliselt mudeli võimekus mustreid õppida ja seega ka suurem oht ülesobituda. Kasutasin sõna „teoreetiliselt“, sest kuigi on tõestatud, et piisavalt suur tehisnärvivõrk suudaks õigeid kaale teades mistahes funktsiooni (nt päriselus toimuva andmetekke protsessi ideaalset mudelit) väga täpselt kujutada (vt teoreemi), pole olemas ideaalset õppimisalgoritmi, mis need ideaalsed kaalud alati leida suudaks (seda enam, et meil on piiratud arv näiteid ja tunnuseid). Rohkemate parameetritega, kuid ebaefektiivse arhitektuuriga (neuronite, kihtide ühenduvusmustri) mudel võib olla hoopiski nõrgem mustrite õppija kui mõni vähemate

õpitavate parameetritega, kuid sisendandmete ja ülesande jaoks sobiva arhitektuuriga mudel[15]. Seega on oluline valida oma andmete jaoks sobiv võrguarhitektuur.

Tehisnärvivõrkude arhitektuuride kogum on väga mitmekesine. Täissidusa võrgu ilmse puudusena võib märgata, et kõik-kõigile-ühendused lisavad võrku väga palju kaale, täpsemalt $N \times M$ kaalu iga kahe neuronikihi vahele, kus N ja M on kahe naaberkihi neuronite arv. Seega on loogiline idee luua võrke, mis pole täissidusad ja kus kihtide vahel on ainult osalised ühendused. Teine kõik-kõigile-ühenduste puudus on see, et need kombineerivad igas teise kihi neuronis kõiki sisendeid omavahel, kuigi mõned tunnused ei oma ilmselgelt koos mingit erilist tähendust ja oleks mõistlik kombineerida üksnes omavahel seotud tunnuseid. Näiteks, kui sisendiks on pildi pikseliväärtused, siis pole mõtet proovida omavahel kaugel asuvate pikslite väärtusi kombineerida ja sealt mingit teadmist otsida. Et vähendada parameetrite arvu ja omavahel kombineeritavate tunnuste hulka, on kasutusele võetud kihid, mille ühenduvus on lokaalne – iga järgmise kihi neuron näeb ainult teatud ala eelmise kihi väljunditest. Sellistest võrkudest räägime täpsemalt masinnägemise peatükis 6.2.

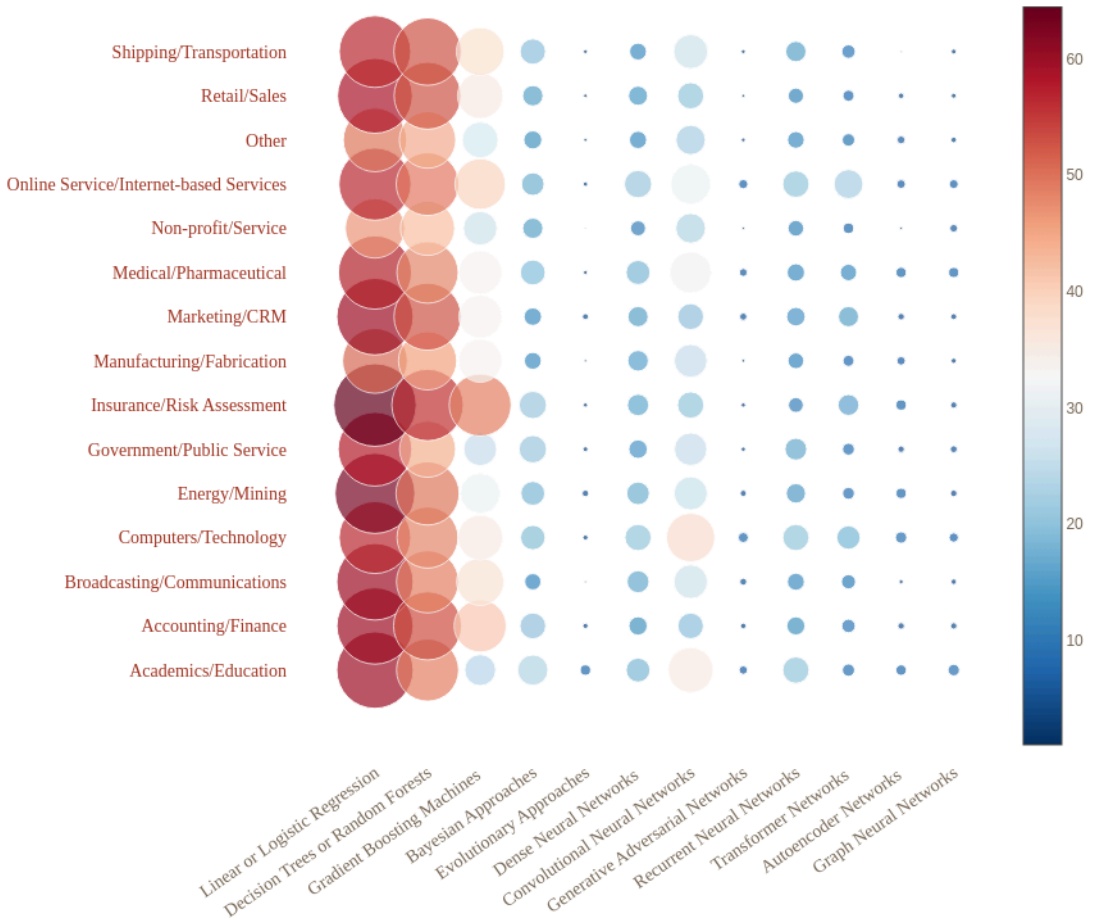
Kihtide vahel võib luua ka keerulisemaid kui üks-ühele-ühendusi, näiteks võib mingi kiht anda sisendit mitmele väljundkihile või saada sisendeid mitmelt eelmiselt kihilt (hargnemine, koondumine, *skip-connections*). Veel keerulisemad on **rekurrentsed võrgud**, mis pole enam pärilevivõrgud, vaid sisaldavad ühenduste tsükleid. Need võrgud on loodud andmejadade töötlemiseks üks jada element korraga, samal ajal eelmiste jada liikmete kohta infot meeles pidades. Viimase aja populaarseim võrgutüüp on aga **transformerid**, tehisnärvivõrkude arhitektuur, mis sisaldab tähelepanu kihte. Need võimaldavad igal kihil otsustada, millisele oma sisendite (eelmise kihi väljundite) osadele oma väljundeid arvutades kõige enam tähelepanu pöörata. Transformerite eelis on ka see, et neis sisalduvaid arvutusi on olemasoleval riistvaral väga efektiivne paralleelselt arvutada, tehes selle veel võimsama mudeli kättesaadavaks. Rekurrentseid võrke ja transformereid on kõige enam kasutatud keeletöötluses.

Tehisnärvivõrke saab kasutada nii juhendatud, juhendamata kui ka stiimulõppes, nii klassifikatsiooni kui ka regressiooni jaoks. Erinevatel eesmärkidel kasutades muutub põhimõtteliselt ainult see, mida me võrgu viimasest kihist välja saada soovime ja kuidas väljundi eksimust ehk kadu (ingl k *loss*) arvutame. Olles defineerinud kaofunktsiooni, saame seda gradientlaskumise abil minimeerida. Tehisnärvivõrkude puudus on, et nende loomiseks peab kasutatav treenimisandmestik tavaliselt olema väga suur (sõltuvalt õpitavate parameetrite arvust, tuhandetest sadade miljonite näideteni), kuid ka see probleem leeveneb tehnoloogia arenedes (vt ptk 6.4).

Sügavõpe kujundab innovatsiooni paljudes tööstusharudes. Seda rakendatakse nii ärilistes kui ka teaduslikes valdkondades, piltide, tekstide, kõne, keemiliste ühendite omaduste, genoomi, finantsturgude ja paljude muude andmeallikate töötlemiseks, mõistmiseks ning nende põhjal ennustuste tegemiseks. Juhendamata õppe

[15] Suurte võrkude korral leidub palju erinevaid parameetrite komplekte, mille korral treeningviga on minimaalne, ja kuna otsingumeetodid ei näe kaugemale treeningandmetest, on valik nende vahel juhuslik. Ainult vähesed neist aga üldistuvad ka uutele näidetele, seega lootus leida hea mudel on väike, välja arvatud juhul, kui mudeli arhitektuur seab mingeid muid piiranguid funktsioonidele, mida mudel õppida suudab.

tehisnärvivõrkudel baseeruvad kasutajate käitumismustrite mõistmise lahendused Netflixis ning Amazoni ja eBay veebipoodides. Närvivõrgud on kasutusel paljudes rakendustes (nt Google Assistant, Siri, Google Translate, Facebooki piltidel inimeste märgendamise rakendus) ja kõigis parimates isejuhtivate autode süsteemides. Sügavõppe algoritmid on andnud lubavaid tulemusi ka meditsiinis ja rahanduses. Siiski, joonisel 6.3 näeme, et sügavõppe revolutsioon pole jõudnud väga sügavale. Kuigi maailma tipptegijad ja rikkaimad korporatsioonid kasutavad tehisnärvivõrke, on enim kasutatavateks masinõppe algoritmideks maailmas ikka veel lihtsamad mudelitüübid. Võib minna aega, enne kui maailmas on piisaval hulgal sügavõppe eksperte ning piisavalt töökindlad arenduse töövood, tööriistad ja tavad, et ka keskmise suurusega ettevõtted saavad selle tehnoloogia juurutamist endale lubada. Siinkohal on aga veel kaks aspekti: tihti piisabki lihtsamast mudelitüübist ja tehisvõrk pole otstarbekas ning sageli puudub organisatsioonil võimekus ja valmidus selliseid mudeleid integreerida.



Joonis 6.3. Erinevate masinõppe mudelitüüpide levimus. *Joonis on loodud Kaggle'i kasutajate seas 2022. aastal tehtud uuringu alusel. Küsimused olid „mis valdkonnas te töötate?" ja „milliseid masinõppe algoritme te kasutanud olete?". Mullikese värv ja suurus peegeldab vastava mudeli kasutajate hulka sektorisse kuuluvate vastajate hulgas. Nagu näha, polnud tehisnärvivõrgud toona veel üheski valdkonnas kõige enam kasutatav mudelitüüp, kuid nende kasutamine on kasvutrendis.*[16]

[16] Allikas. Litsents: Apache 2.0.

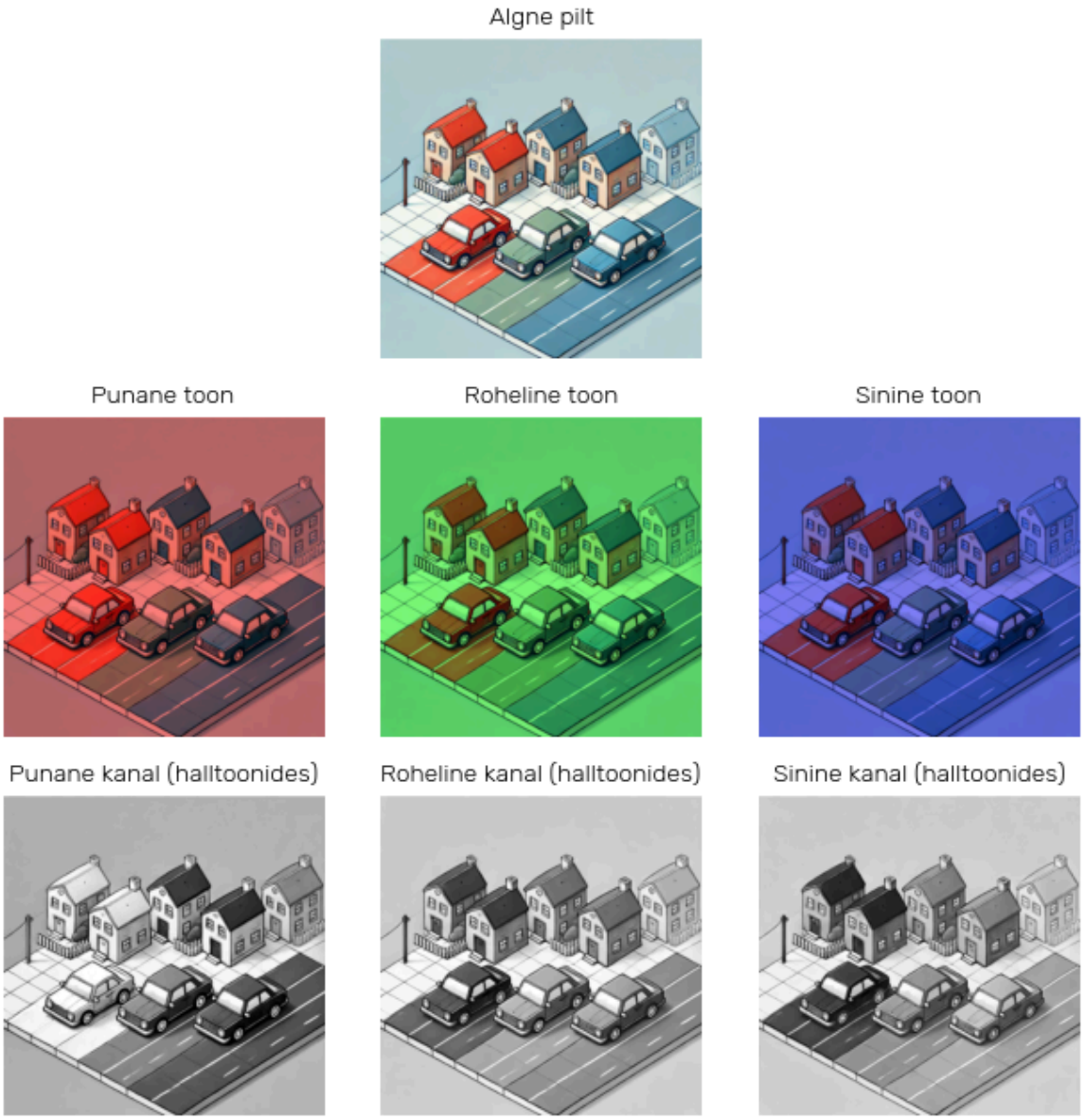
Kasulikud allikad mudelite ja tehnoloogia leidmiseks
Tehisnärvivõrkude ja teiste sügavõppe mudelite valik ning arendamine võib olla keeruline, kuna saadaval on palju erinevaid lahendusi ja arhitektuure. Õnneks on olemas hulk veebiplatvorme ja kogukonnapõhiseid ressursse, mis aitavad teadlastel ja arendajatel leida ning katsetada uusimaid mudeleid ja tehnoloogiaid.

- **Hugging Face** (huggingface.co) on aastal 2024 üks populaarsemaid platvorme, kus teadlased ja arendajad jagavad kasutamiseks valmis mudeleid erinevates valdkondades, sealhulgas loomuliku keele töötluses (NLP), masinnägemises ja helitöötluses. Platvorm võimaldab lihtsat ligipääsu mudelitele ning pakub tööriistu nende rakendamiseks ja peenhäälestamiseks.
- **Papers with Code** (paperswithcode.com) ühendab teadusartiklid ning nende juurde kuuluvad mudelid ja lähtekoodid, tehes teadussaavutuste praktilise katsetamise kergesti kättesaadavaks. See on kasulik platvorm, kui soovite rakendada uusimaid teadustöös esitatud lahendusi oma projektides.
- **TensorFlow Hub** (tfhub.dev) on Google'i loodud platvorm, kus jagatakse sügavõppe mudeleid ja komponente. See on kasulik ressurss TensorFlow tarkvara kasutavate mudelite leidmiseks.
- **Model Zoo** (mudelite loomaaed) erinevate raamistike jaoks, näiteks **PyTorch Model Zoo** ja **ONNX Model Zoo**, pakuvad laia valikut mudeleid, mida rakendada või oma ülesande jaoks peenhäälestada.

Nende ressursside kasutamine aitab kiirendada arendustööd, vähendades vajadust nullist mudeleid luua ja võimaldades kasutada juba tõestatud lahendusi.

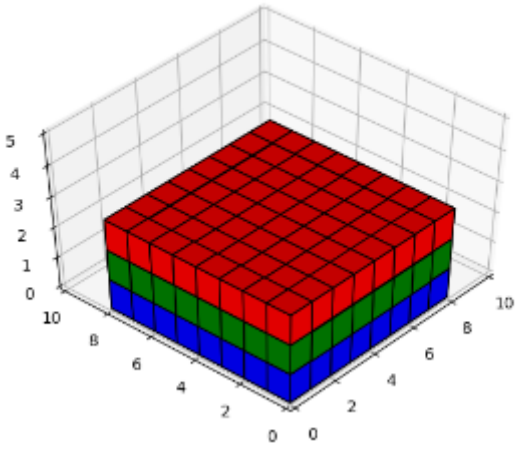
6.2 Masinnägemine

Pildid on üsna eriline andmeliik, sest need koosnevad väga suurest hulgast väga vähe informatiivsetest tunnustest. Värvilise pildi puhul on need tunnused iga piksli punasuse, roheluse ja sinisuse väärtused ehk RGB väärtused (joonis 6.4). Värvipildid on arvutis talletatud ja ekraanil esitatud tegelikult põhitoonide kombinatsioonina. Arvutis on justkui kolm mustvalget pilti, üks punase, üks rohelise ja üks sinise jaoks. Alles nende ühendamisel saadakse kokku õige värv (joonised 6.3 ja 6.4). Nii võib mõelda, et pilt on arvutis kui 3D-objekt, mis koosneb kolmest värvikihist.



Joonis 6.4. Värviline pilt koosneb arvuti jaoks kolmest põhivärvi pildist. Lähtekood.

Ühe teatud piksli punasuse väärtusest kindlasti ei piisa, et otsustada, kas pildil on kass või koer. Suvalise kümne piksli tooni (kokku 30 arvulist tunnust) puhul me tõenäoliselt samuti ostust teha ei saaks.



Joonis 6.5. Pilt on arvutis salvestatud kui kolmemõõtmeline objekt. *Iga piksli kohta on kolm väärtust, mis peegeldavad punase, sinise ja rohelise värvi osakaalu selle piksli värvis. Seega koosneb pilt tegelikult justkui kolmest üksteise otsa asetatud laius × kõrgus kihist (maatriksist), millest igaüks kirjeldab ühe värvi intensiivsust eri pikslites. Võrdluseks: mustvalge (ingl k* grayscale*) pilt koosneb ainult ühest maatriksist, mis kirjeldab pildi heledust eri punktides.*

Ülesanne: piltide kujutamise mõistmine

Punane

0	0	0	0
0	0	0	0
255	255	255	255

Roheline

0	0	0	0
0	0	0	0
255	255	255	255

Sinine

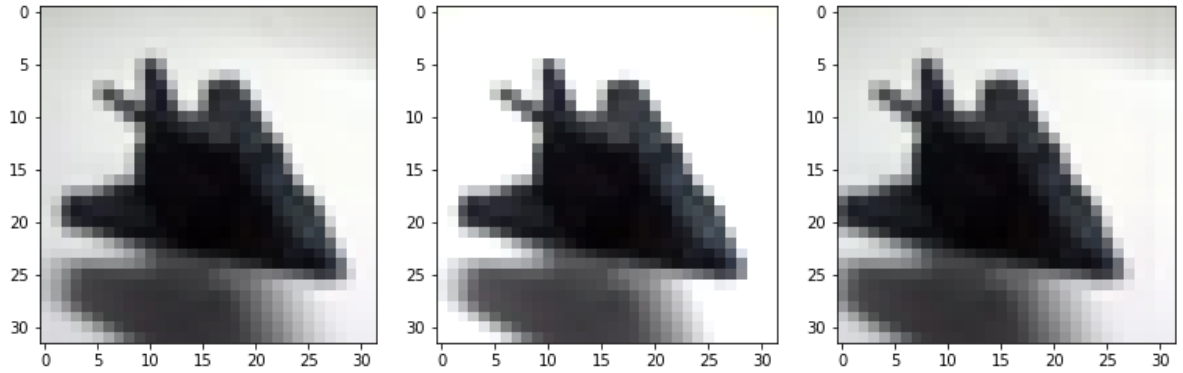
255	255	255	255
0	0	0	0
255	255	255	255

Joonis 6.6. Mis on pildil? *Esimene 3 × 4-maatriks vastab punastele, teine rohelistele ja kolmas sinistele komponentidele. Maatriksite taustavärv on illustratiivne. Vastuse leiate siit.*

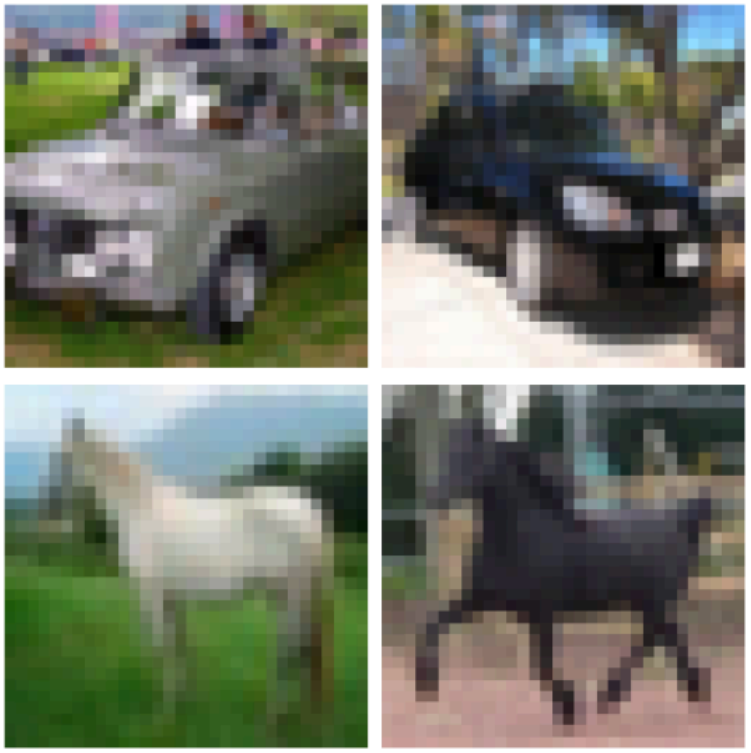
Kui paljude teiste andmeallikate puhul võib olla masinõppe peamine keerukus leida paljude sisendtunnuste hulgast tunnused või mõnest tunnusest koosnevad kombinatsioonid, mis ennustatava suurusega hästi korreleeruks, siis piltide puhul pole neid mõnda informatiivset tunnust olemas. Lihtsate seaduspärade (suurest hulgast variantidest) otsimise asemel on probleemi **keerukus paljude väheinformatiivsete tunnuste info kombineerimises**. Et tuvastada pildil jooni, tekstuure, värvi gradiente jne, on vaja leida väga paljudest tunnustest koosnevaid kombinatsioone. Juba 3 × 3 piksli suurune ala sisaldab 27 väärtust, mille võimalikke kombinatsioone on lugematu arv. Nagu eespool mainitud, õpivad tehisnärvivõrgud arvutama sisendtunnustest tuletatud tunnuseid, neist omakorda järgmisi tuletatud tunnuseid ja nii edasi, kuni meil on mingi hulk keerulisi, aga meie ülesande jaoks väga informatiivseid tunnuseid, mille alusel saab lõpliku vastuse anda. Seega on tehisnärvivõrkudel olemas eeldused paljude tunnuste nõrga informatsiooni kombineerimiseks. Probleemiks on peamiselt ülesobitumine – tuhandete tunnuste triljonite võimalike kombinatsioonide hulgas on neid, mis juhuslikult, kuid mitte põhjuslikult, treeningnäidete puhul märgendiga hästi korreleeruvad.

Teine piltide kui andmete keerukuse allikas on **tunnuste mitmetähenduslikkus ja tähenduse suhtelisus**. Pildil mingis asukohas asetsev piksel pole kuidagi fikseeritud või kindla tähendusega vaadeldava ja näiteks klassifitseeritava objekti suhtes. Pildil kujutatava looma silmad võivad asetseda eri kohtades, seega silmade värvis ja kujus sisalduv info on mudeli sisendis erinevatel piltidel eri tunnustes. Seega tuleb samalt pildiosalt otsida kõiki huvipakkuvaid mustreid (sest seal võib olla mis iganes) ja kõiki mustreid tuleb otsida kõigilt pildiosadelt (sest need võivad olla ükskõik kus). Samuti võib pilt olla erineva valgustatusega, suurendusega, udususega või erineva nurga alt tehtud ja silmad võivad olla erineva tooniga, suuremad, väiksemad või hoopiski pildilt puududa.

Sama sisu ja tähendusega stseeni või sama objekti pildil kujutades on võimalik saada väga erinevate tunnuste (RGB) väärtustega pildid. Seega pole tähenduselt ja inimese silm + aju süsteemi jaoks sarnased pildid tunnuste väärtusi omavahel võrreldes üldsegi sarnased. Meenutagem, et juhendamata õppes, klasterdamisel liigitasime me ühte gruppi omavahel Eukleidilise või koosinuskauguse alusel sarnased näited. Samuti kasutavad lähima naabri ja tugivektormasinate algoritmid eeldust, et sarnaste sisendtunnuste väärtustega näidetel on sarnased märgendid. Piltide puhul see eeldus paika ei pea. Joonisel 6.7 näete kolme pilti, mis on tunnuste väärtusi ükshaaval võrreldes ja erinevusi mingil viisil kokku võttes (Eukleidiline kaugus, erinevuste ruutude summa, ruutkeskmine viga vms) omavahel väga erinevad. Sisult on need pildid aga sarnased. Samuti võivad erineva sisuga, kuid samade toonidega pildid olla lihtsate kaugusemõõdikute järgi üsna sarnased (joonis 6.8).



Joonis 6.7. Sarnased pildid võivad olla arvuti jaoks erinevad. *Need pildid on tunnuste väärtusi ehk pikslite RGB väärtusi omavahel võrreldes väga erinevad. Esimese ja teise pildi vahel on heleduse erinevus, mis tekitab kõigi pikslite kõigis värviväärtustes märgatava nihke. Kolmas pilt on kaks pikslit vasakule nihutatud, tekitades kõigis tunnustes erinevused. Samal ajal on nende kolme pildi tähendus inimese jaoks ja tõlgendus erinevates võimalikes kontekstides väga sarnane.*



```
Valgete vaheline pikslite erinevuste summa on  156105
Mustade vaheline pikslite erinevuste summa on  170710
Hobuste vaheline pikslite erinevuste summa on  246457
 Autode vaheline pikslite erinevuste summa on  226288
```

Joonis 6.8. Pikslitevaheline kaugus. *Sama värvi hobune ja auto on pikslite väärtuste sarnasuse alusel omavahel lähemal kui eri värvi sama tüüpi objektid.*

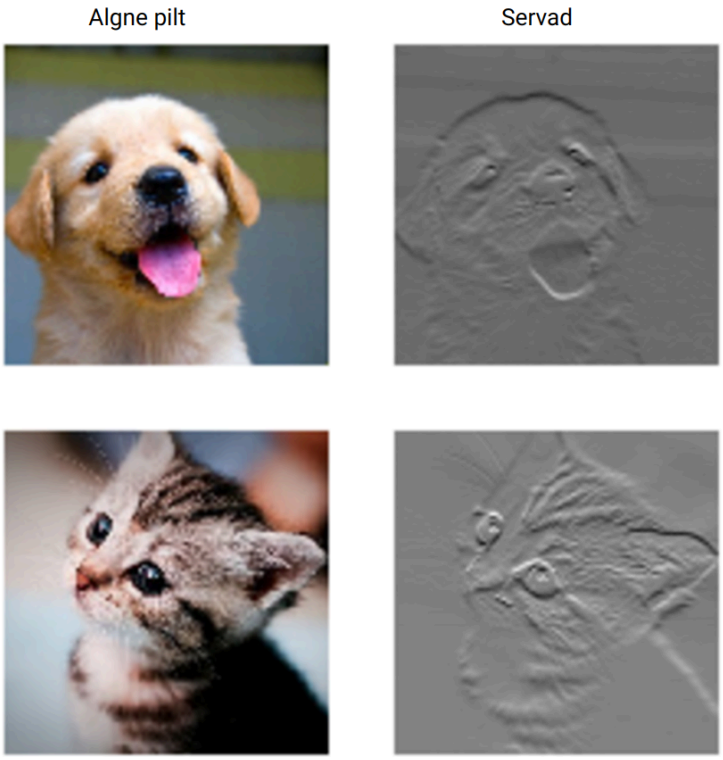
Selline sama tähendusega või sama märgendiga piltide tohutu mitmekesisus teeb masinnägemise keerukaks ja raskendas masinõppe kasutamist pildiandmetel mitme aastakümne vältel. Alles konvolutsiooniliste tehisnärvivõrkude jõulise esiletõusuga alates aastast 2012 on pildid muutunud analüüsitavaks ja kasulikuks sisendiks masinõppe jaoks.

6.2.1 Konvolutsioon

Eelnevast arutlusest ja pildinäidetest saime vihjeid selle kohta, mida tuleks teha, et pildilt teatud tüüpi objekte tuvastada. Muidugi mõistsid targad inimesed juba ammu enne tehisnärvivõrkude esiletõusu järgmist.

- On vaja otsida pildilt teatud ruumilisi mustreid, näiteks servi, täppe, tekstuure. See mõte sisaldab arusaama, et mingit kõrgemat tähendust omavad pildi puhul omavahel ruumiliselt lähedal asuvad tunnused, mitte suvalised tunnustepaarid, näiteks alumise vasaku nurga piksli rohelisus ja parema ülemise serva piksli punasus. Nii servad, täpid, tekstuurid, heleduse gradiendid, nurgad kui ka paljud muud võimalikud mustrid on **lokaalsed ruumilised tunnused**, mille esinemist mingis pildiosas saame sisendtunnuste ehk selle pildiosa RGB väärtuste alusel arvutada (joonis 6.9).
- Kõigilt pildiosadelt tuleb otsida samu tunnuseid, sest me ei tea, kuidas objektid pildil täpselt asetsevad. Sama funktsiooni (mingi mustri olemasolu mõõtmise)

rakendamist erinevatele sisendi osadele nimetatakse selles kontekstis konvolutsiooniks ja see annabki hiljem konvolutsioonilistele võrkudele nime.



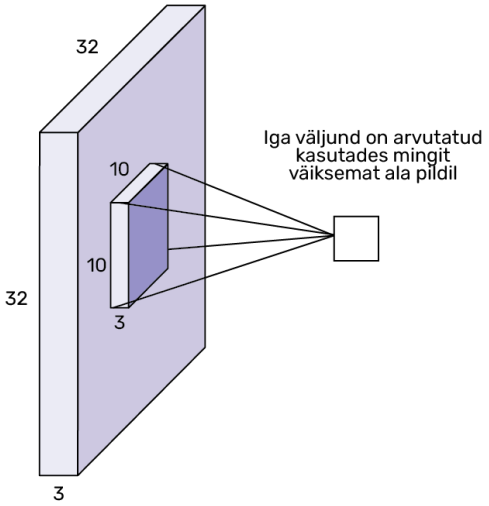
Joonis 6.9. Pildilt lokaalsete omaduste otsimine. *Näiteks võib otsida pildilt horisontaalseid jooni, kus ülevalt alla suunal toimub üleminek heledalt tumedale. See otsing annab keskmised väärtused alal, kus mingeid horisontaalseid üleminekuid pole, suured väärtused heledalt tumedaks üleminekute puhul ja väikesed tumedalt heledaks üleminekute puhul (otsitule vastupidine). Nendest tulemustest moodustub uus mustvalge pilt, kus heledana on suured, hallina keskmised ja tumedana väikesed väärtused.*

Siiski ei suudetud välja mõelda ruumiliste tunnuste komplekti, mille olemasolu või puudumine pildi eri osades oleks nii informatiivne, et selle alusel saaks fotosid sisu järgi klassifitseerida. Kõik muutus aastal 2012. Piltide tuhandesse eri klassi liigitamise võistlusel (ImageNet võistlus) osutus võitjaks tehisnärvivõrkudel baseeruv lahendus (Krizhevsky, 2012). See lahendus otsis samu lokaalseid ruumilisi mustreid erinevatelt pildiosadelt (juba teada olnud konvolutsiooniline lähenemine), kuid mudel valis õppimise käigus ise, milliseid mustreid tuvastada, et täpsus parim saaks. Lisaks oli oluline võrgu sügavus – juba leitud ruumiliste tunnuste omavahel kombineerimine keerulisemateks ruumilisteks tunnusteks. See mudel tegi testandmestikul 40% vähem vigu kui senised lahendused, mis otsitavaid ruumilisi tunnuseid mudelil endal õppida ei lasknud.

6.2.2 Konvolutsioonilised tehisnärvivõrgud

Tunnused, mida konvolutsiooniline võrk pildilt otsib, on defineeritud kolmemõõtmeliste maatriksite ehk tensoritena (nagu ka pilt). Need tensoreid nimetatakse filtriteks. Iga **filter on nagu väike pildike**, enamasti 3×3 kuni 7×7 pikslit suur ja sama sügav kui sisendpilt (üks kanal mustvalge ja kolm kanalit värvipildi puhul). Igat filtrit rakendatakse kõigile võimalikele pildiosadele, et saada selle **filtri aktivatsioon (sobivus või**

esindatus) selles pildiosas. Aktivatsioon arvutatakse filtri värviväärtuste ja filtriga sama suure pildiosa värviväärtuste skalaarkorrutisena (joonis 6.10). Filtrites sisalduvad väärtused on mudeli õpitavad parameetrid – iga filtri väljanägemine on õpitav.



Joonis 6.10. Filtri rakendamine pildiosale. *Filter koosneb 10 × 10 × 3 kaalust. Filtri kaalud ja pildiosa pikslite väärtused korrutatakse ja liidetakse (ehk leitakse skalaarkorrutis) ning tulemusena saadakse üks arv, mis mõõdab nende sarnasust.* N *filtrit kõigile erinevatele pildiosadele rakendades on väljundiks kolmemõõtmeline arvumassiiv, mille moodustavad* N *filtri aktivatsioonikaardid.*

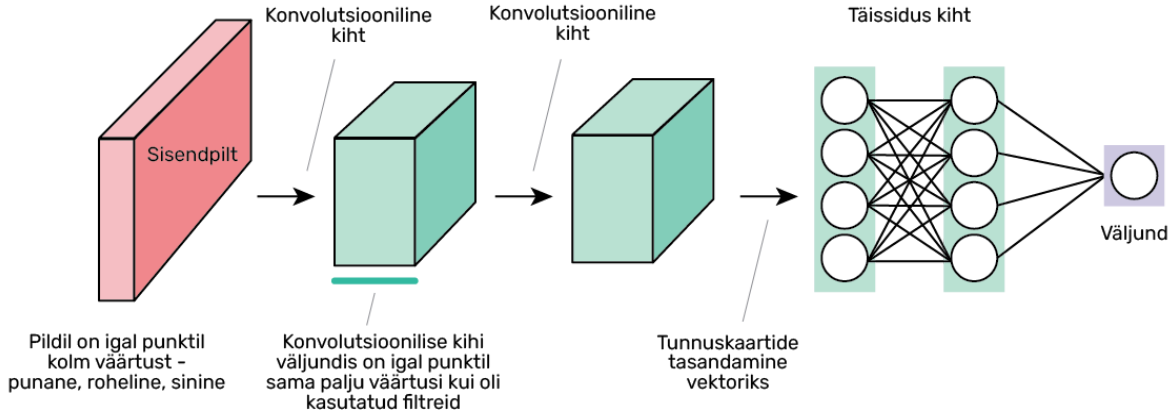
Filtrite kõigile pildiosadele rakendamise tulemusena tekivad aktivatsioonide maatriksid (TNV terminites **aktivatsioonikaardid** ehk **tunnuskaardid**) ehk uued mustvalged pildid, mis näitavad iga filtri kohta, kui filtriga sarnased erinevad pildiosad on. Rakendades *N* filtrit, saame *N* sellist pilti.

Tehisnärvivõrgu konvolutsiooniline kiht seisneb teatud hulga õpitavate väärtustega filtrite rakendamises pildile. Igat asukohta igal aktivatsioonikaardil tõlgendatakse neuronina, mis saab sisendeid ainult teatud pildiosast ja mille ühendustugevused nende sisenditega ongi filtri kaalud. Sama filtrit ehk samu ühendustugevusi (aga erinevate pildiosadega) kasutab terve aktivatsioonikaardi jagu neuroneid.

Tõlgendades neid aktivatsioonikaarte „kanalitena“ samuti, nagu me tõlgendame punase, rohelise ja sinise värvi esinemistugevuse maatrikseid värvikanalitena, märkame, et meil on tegelikult tekkinud uus pildisarnane kolmemõõtmeline andmemassiiv. Selle kaks mõõdet on ruumilised ja sama tähendusega kui pildi algsed teljed, kolmas mõõde tekib aga mitme aktivatsioonimaatriksi üksteise otsa ladumisest, nagu ennist tegime värvimaatriksitega. Kui me saime mingeid ruumilisi tunnuseid otsida kolme kanaliga piltidelt, siis saab seda teha ka *N* kanaliga piltidel, mille kanalid kujutavad eelmise konvolutsioonilise kihi filtrite aktivatsioonikaarte. Seega **saab ühe konvolutsioonilise kihi väljundile rakendada järgmist konvolutsioonilist kihti**. Teiseste filtrite rakendamine eelmise filtrihulga väljunditele tekitab sügavuse ja võimaldab teadmist

lihtsate ruumiliste tunnuste olemasolu kohta keerulisemal viisil kombineerida, keerulisi ruumilisi tunnuseid otsida.

Lihtsamal juhul koosnebki konvolutsiooniline tehisnärvivõrk reast konvolutsioonilistest kihtidest, mille eesmärk on pildilt kiht-kihilt aina keerulisemaid ruumilisi mustreid leida, ning täissidusatest kihtidest, mis viimase konvolutsioonilise kihi väljundina saadud tunnuskaartide väärtusi lõplikult kombineerivad ja vastuse annavad. Selline võrk on illustreeritud joonisel 6.11.



Joonis 6.11. Konvolutsiooniline tehisnärvivõrk. *Kahest konvolutsioonilisest kihist ja kahest täissidusast kihist koosnev konvolutsiooniline tehisnärvivõrk. Konvolutsiooniliste ja täissidusate kihtide vahel on tasandamisoperatsioon, mis lihtsalt muudab tunnuskaartide väärtused üheks pikaks vektoriks, sest täissidusad kihid ruumilist paigutust arvesse ei võta.*[17]

Oleme defineerinud, et masinõppe mudel on lihtsalt üks matemaatiliste toimingute jada, mis sisendtunnuste väärtuste põhjal väljundi arvutab. Konvolutsiooniliste tehisnärvivõrkude puhul võib tunduda, et see arvutuste jada ehk valem on küll õige keeruliseks läinud. Siiski koosneb see ainult sisendite alamosa valimisest, korrutamisest, liitmisest ja aktivatsioonifunktsiooni rakendamisest (millest me seni siin lihtsuse mõttes rääkinud pole). Et need on kõik iseenesest lihtsad matemaatilised toimingud, pole üllatav, et saame arvutada eksimuse tuletist neis tehetes osalevate parameetrite, sealhulgas filtrites sisalduvate kaalude, suhtes. Kui teame vea tuletisi õpitavate parameetrite suhtes, saame kasutada gradientlaskumise algoritmi ja muuta oma mudeli parameetreid vähehaaval vastusteid täpsemaks muutvas suunas. Nii õpib konvolutsiooniline võrk märgendatud andmestiku alusel, milliseid ruumilisi mustreid sisendpildilt esimese konvolutsioonilise kihi filtrite abil tuvastada, milliseid esimeses kihis leitud ruumiliste tunnuste mustreid teise konvolutsioonilise kihi abil otsida jne.

Olles tutvunud konvolutsiooniliste võrkude põhiideedega, tutvustame järgnevates alapeatükkides masinnägemise alaülesandeid, mida nende abil lahendatud on. Paljudel juhtudel on neid ülesandeid lahendavate võrkude loomisel kasutatud ka siinkohal mitte mainitud trikke (ingl k *dropout*, *layer norm*, *learning rate scheduling* jne) ning keerulisi tehisnärvivõrgu kihitüüpe (*pooling layers, deconvolution layers, pyramid networks* jne), mille kõigi tutvustamiseks oleks vaja eraldi õpikut (nt (Zhang, 2021)). Tehisnärvivõrke

[17] Joonis on kohandatud, allikas: CS231n, Litsents: MIT.

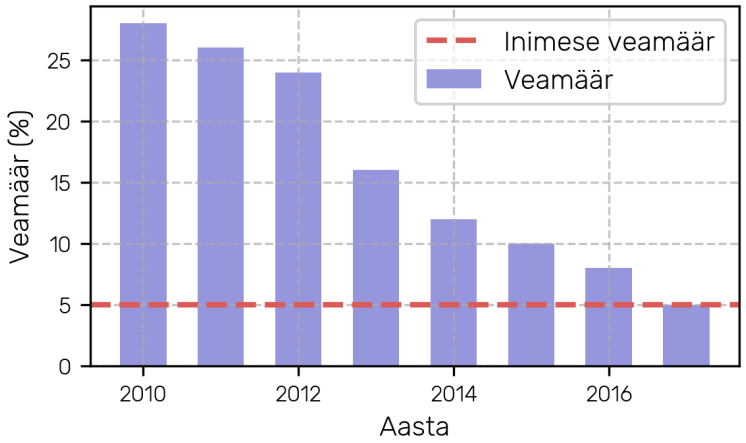
kasutavatel praktikutel on kasulik neid uuendusi ja kihte vähemalt nende võimekuste ja mõju tasandil tunda, et mõista, kuidas mingi tehisnärvivõrk infot töötleb ning kas see mingi antud lahenduse jaoks mõistlik on. Samas on kindlasti võimalik ka võrke edukalt rakendada täiesti mustade kastidena, uskudes nende võimetesse ainult teiste ülesannete lahendamise õnnestumiste alusel.

Ajaloolise märkusena tuleks mainida, et kuigi konvolutsiooniliste tehisnärvivõrkude laialdane kasutuselevõtt sai alguse AlexNeti (Krizhevsky, 2012) triumfiga 2012. aastal ImageNet andmestiku 224 × 224 × 3 dimensiooniga fotodel, leiutas need tegelikult juba aastal 1989 Yann LeCun 32 × 32 suurusega mustvalgete piltide töötlemiseks (LeCun, 1989). Täpsemalt kasutati neid numbrite 0–9 ära tundmiseks postikoodide automaatseks lugemiseks postisaadetistel. Juba 1990ndatel kasutati neid ka optilises märgituvastuses laiemalt (ka tähestiku tähed) (LeCun, 1998).

6.2.3 Masinnägemise tuntuimad ülesanded

Masinõppe lahendused, mis suudavad piltide pikslivääртustes laiali olevat infot stseeni kohta üles leida, on loomulikult osutunud äärmiselt kasulikuks. Joonis 6.12 kujutab esimest nelja ülesannet, mida hakati esimese viie aasta jooksul pärast konvolutsiooniliste võrkude esiletõusu aktiivselt uurima.

- **Piltide klassifitseerimine ja objektituvastus**
  Enamasti vastatakse küsimusele „mis on peamine objekt pildil?“ ehk vastuseks on üks objektitüüp. Tegu on tüüpilise klassifikatsiooniga, kus vastuseks on üks klass mitmest võimalikust (tegelikult väljastab mudel küll iga klassi kohta tõenäosuse, millest suurima valime). Sellist mudelit on võimalik panna vastama ka kolme kõige tõenäolisemat klassi või saame rakendada seda pildi eri osadele, et proovida leida mitut erinevat objekti. Selle mudelitüübi tuntuim näide ongi AlexNet, mis liigitas väikese dimensiooniga fotosid tuhandesse klassi. Keerulisemate ja võimekamate tehisnärvivõrgu arhitektuuride hulgast on objektituvastuses laialdaselt kasutatud ResNet-tüüpi võrgud ning alates 2020. aastast ka visuaalsed transformerid.



Joonis 6.12. Parima mudeli klassifitseerimistäpsus ImageNet miljoni pildi ja tuhande klassiga andmestikul. *Aastal 2012 võitis võistluse esmakordselt konvolutsiooniline tehisnärvivõrk, aastal 2015 ületati inimtäpsus, aastal 2018 võistlust enam ei korraldatud, sest ülesannet põhimõtteliselt enam paremini lahendada ei saa. Lähtekood.*[18]

[18] Info allikas: (Langlotz, 2019), litsents CC 4.0.

Fotode liigitamine klassidesse võib olla teatud juhtudel praktiliselt kasulik, näiteks binaarse klassifikatsiooni ülesannetes „kas parkimiskoht on vaba?“, „kas foorituli on roheline?“, „kas pilt sisaldab alastust või vägivalda?“, „kas valvekaamera pildil on mõni inimene?“, „kas fotol oleval tootel on kvaliteediprobleem?“. Ka näotuvastus liigitub objektituvastuseks, milles eristatavateks klassideks on erinevad inimesed. Praktiliselt rakendatavad on ka näiteks numbri- ja märgituvastus (tekstituvastus tähthaaval). Kui meil on mingi numbrite/tähtede lokaliseerija, mis suuremalt pildilt sümbolid ühekaupa välja lõikab, siis saab tõesti esitada küsimuse „mis on peamine sümbol pildil?“. Kuna tekst asub enamasti sirgel real, on tekstist üksikute tähtede välja lõikamine tihti üsnagi lahendatav. Standarditud vormidel on aga lausa iga tähe jaoks asukohad ette määratud. Ka paljudel muudel juhtudel (nt kvaliteedikontroll toomisliinil) on kaamera asend vaadeldava objekti suhtes fikseeritud, klassifitseerimist vajava pildiosa saab automaatselt pildilt välja lõigata ja mudelile sisse anda.

- **Lokaliseerimine ehk „mis objektid on pildil ja kus?“**
  Märgake, et lokaliseerimisülesanne sisaldab peaaegu alati ka lokaliseeritud objekti tüübi tuvastamist. On ka erijuhte, mis ainult lokaliseerivad või esimese (eraldi kasutatava) sammuna lokaliseerivad potentsiaalsed objektid. Vastusena oodatakse kõiki tuvastatavatesse klassidesse kuuluvate objektide asukohti kastikestena[19], mis objekti ümbritsevad, ja tavaliselt ka objektitüüpi. „Kus ja mis tüüpi liiklejad asuvad pildil?“ on kasulik küsimus isejuhtivate autode kontekstis, et valida ohutu sõidutrajektoor ning kiirus. „Kus asub pakendil kuupäev?“ on kasulik eeltöötluse samm optilise märgituvastuse jaoks kvaliteedikontrollis, lõikamaks välja meid huvitava ala suuremalt pildilt.

Joonis 6.13. Neli võimalikku pilditöötlusülesannet. (a) *Lihtsalt klassifitseerimine: mis objektid on pildil?* ***(b)*** *Lokaliseerimine ja klassifitseerimine: mis objektid on pildil ja kus?* ***(c)*** *Semantiline segmenteerimine: milline piksel kuulub millist tüüpi objektile?* ***(d)*** *Individuaalne segmenteerimine: millised pikslid kuuluvad igale objektile?* [20]

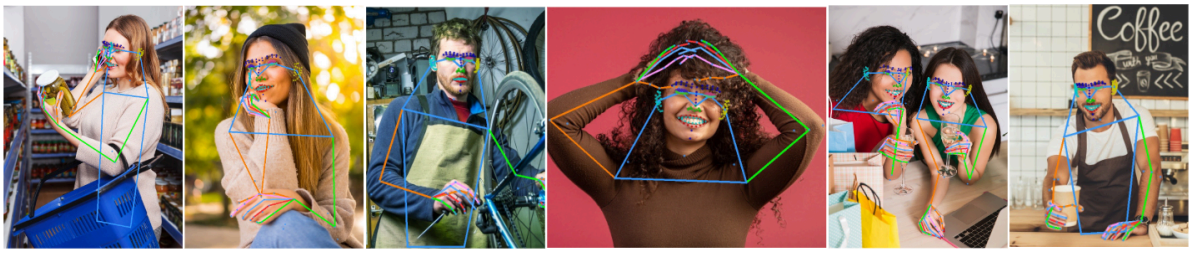
[19] Mudel võiks ennustada ka objekti ümbritseva ovaali või muu kujundi, aga kaste on mugav välja lõigata ja uue pildina salvestada.
[20] Kohandatud, allikas: Microsoft COCO andmestik.

Tihti polegi oluline, kus objektid asuvad, vaid kui palju neid on. Selle jaoks on ikkagi mõistlik objektid tuvastada, võimalikud topelttuvastused asukohtade kattuvuse alusel välja filtreerida ja siis objektid kokku lugeda. „Mitu inimest on videopildil?“ aitab jälgida jalakäijate liikumist linnaplaneerimise eesmärgil. „Mitu rakku on pildil?“ aitab mikroskoobipilte analüüsida ja võimalike ravimite tõhusust või toksilisust automaatselt kvantifitseerida.
Kuulsaim objektide lokaliseerimise ja identifitseerimise mudelitüüp on You Only Look Once ehk YOLO (Redmon, 2016), millest on praeguseks avaldatud vähemalt kaheksa versiooni.

- **Semantiline segmenteerimine vastab iga pildi piksli kohta, millist tüüpi objektile see piksel kuulub**
  See on täpsem asukoha määratlus kui lokaliseerimine, mis andis objekti ümbritseva kasti asukohta. Sellise kasti raamidesse võib jääda ka teist tüüpi objekte, mis klassifitseerimist segavad. Samuti võib olla mudelil keeruline objektidena mõista vaid osaliselt nähtaval olevaid objekte. Iga piksel aga kuulub paratamatult ainult ühele objektile ja vastab täpselt ühele klassile. Segmenteerimine võimaldab aimu saada objekti asendist, poosist, samuti mõõta objektide täpset suurust, pindala. Siiski pange tähele, et semantilise segmenteerimise mudel ise ei jaga samasse klassi segmenteeritud pildialasid individuaalseteks objektideks, seda tuleb teha mingi muu mudeli või reeglistiku alusel. Segmenteerimise abil võib meditsiinilistel piltidel erineva läbipaistva tooniga pildil ära värvida erinevad koed, et lihtsustada arsti jaoks pildi tõlgendamist. Põllumajanduses võib aerofoto alusel automaatselt ära märkida taimehaiguse tõttu välimust muutnud põlluosad ja hinnata nende pindala. Keskkonna jälgimise rakenduste hulgas on näiteks üleujutuste ja metsatulekahjude ulatuse automaatne hindamine aero- või satellidipildi põhjal. Ravimiarenduses võib rakkude kokku lugemise asemel ka nende kogupindala mõõta, et hinnata ravimi mõju rakkude kasvule.
  Segmenteerimise jaoks on laialdaselt kasutatud U-Net tüüpi võrgud (Ronneberger, 2015), mis esmalt loovad pildist konvolutsiooni abil kompaktse vektoresituse ja siis taasloovad algse pildiga sama suurusega segmentatsioonipildi.
- **Individuaalne segmenteerimine lokaliseerib erinevad objektid piksli täpsusega ehk segmenteerib iga objekti eraldi**
  See on täpseim võimalik iga objekti lokaliseerimine pildil. Iga objekti võib pildilt välja lõigata ja mõnele muule taustale kleepida, näiteks fotomontaaži eesmärgil. Iga objekti pindala ja kuju saab eraldi hinnata. „Kui suured ja millise kujuga on pigmendimuutused nahal?“ võib olla kasulik, ennustamaks nahavähi riski. Sarnaselt lokaliseerimisega võimaldab individuaalne segmenteerimine loendamist, näiteks „mitu lammast on pildil?“.

Lokaliseerimise ja individuaalse segmenteerimise tulemusel leitud objekte võib läbi mitme kaadri ehk mitme ajahetke jälgida, et kirjeldada nende liikumist ja näiteks hinnata nende asukohta tulevikus (nt „kas jalakäija astub ülekäigurajale?“). Samuti võib olla kasulik hinnata mingite objektide suuruse muutust ajas.

Objektide lokaliseerimise lisandina on tekkinud valdkond nimega **asendituvastus**. Näiteks isejuhtimises ei piisa inimese järgneva käitumise ennustamiseks tema asukohast, vaid on vaja teada ka tema kehaasendit (joonis 6.14). Selle jaoks defineeritakse kehal võtmepunktid (ingl k *keypoints*), mille asukohad treeningandmestikus ära märgitakse ja mille asukohta uutel andmetel mudel ennustama peab (lisaks asukohakastile või selle asemel). Teades kehaasendit mitmel järjestikusel videokaadril, on inimese liikumist lihtsam mõista ja ennustada, kui ainult asukohakastikeste muutumist või selleks inimeseks segmenteeritud ala muutumist jälgides. Inimese kehaasendid on olulised ka masinate ja inimeste interaktsiooni puhul, et masin mõistaks, mida inimene soovib või plaanib. Ka teiste objektide asendid võivad olla olulised, mitte ainult inimkeha. Isejuhtival sõidukil oleks kasulik teada, millises suunas teatud pildiosas tuvastatud teine sõiduk asetseb. Tööstuses vajavad robotkäpad täpset infot haaratava objekti asendi kohta. Ka näo miimikat saab teatud võtmepunktide asukohtade kaudu kirjeldada, et näiteks videokõnes osalejate rahulolu hinnata. Defineerides võtmepunktid kätel ja nende asendit jälgides on võimalik püüda viipekeelt automaatselt tuvastada ehk sõnadesse tõlkida (Rastgoo, 2021).



Joonis 6.14. Asendituvastus Sapiens tehisnärvivõrgu abil.[21]

Seoses tekstitöötluse arenguga on saanud võimalikuks ka **pildil oleva info lausega kirjeldamine**. Selle jaoks on vaja mudelit, mis koosneb kahest osast. Esimene võrgu osa saab sisendiks pildi ja loob sellest mingi vektoresituse (ingl k *embedding*), mis pildil oleva info mingil kasulikul viisil kokku võtab. Teine võrgu pool genereerib selle vektori alusel lause, mis pilti kirjeldada võiks. Selline ülesanne on veidi halvasti püstitatud, sest samale pildile võib tegelikult vastuseks anda erinevaid lauseid. Matemaatiliselt on funktsioon defineeritud kui seos, mis annab igale sisendile üheselt defineeritud vastuse, mitme võimaliku vastuse andmist pole ette nähtud. Ka tehisnärvivõrkude iga õppimise samm maksimeerib just märgendina näidatud lause vastusena genereerimise tõenäosust sisendina saadud pildi alusel. Siiski, kuna meie väljundiks on tõenäosused, mitte diskreetsed sõnad, on võimalik, et mudel õpib, et mitu sarnase sisuga lauset on üsna võrdselt tõenäolised. Piltide kirjeldamise praktilised rakendused on näiteks fotodele automaatselt allkirjade pakkumine sotsiaalmeediaplatvormidel ja uudisteportaalides fotode kirjeldamine vaegnägijate jaoks. Kuid töötatakse ka näiteks meditsiiniliste piltide (tomograafia, röntgen) põhjal automaatsete raportite koostamise suunal (Xu, 2023).

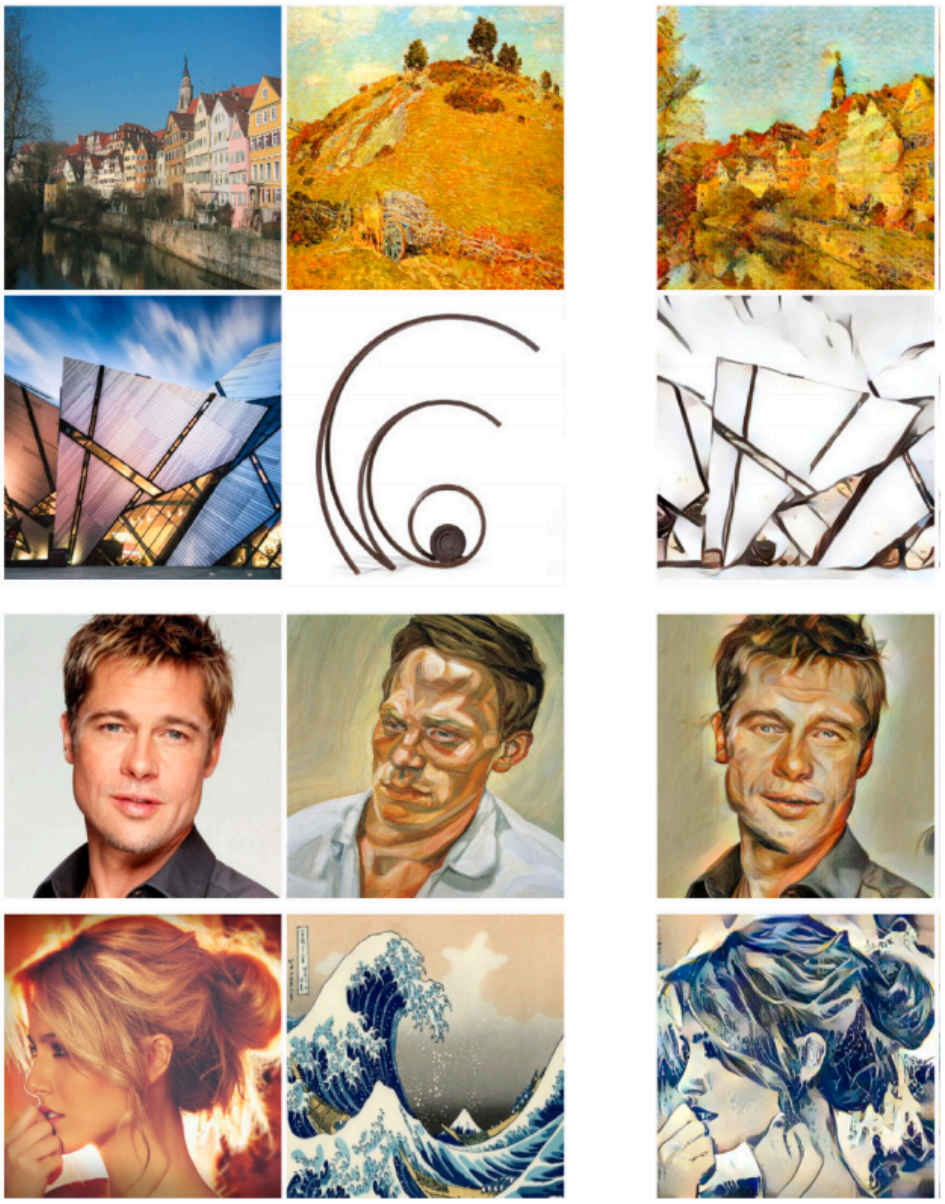
[21] Allikas: (Khirodkar, 2024), litsents CC BY-NC-SA 4.0.

Pilditöötluse loogiline edasiarendus on **videotöötlus**. Videos toimuva kohta võib järeldada midagi ka üksikutele kaadritele rakendatud töötluse väljundite, näiteks objektide asukohakastide alusel. Siiski võib olla tõhusam anda mudelile sisendiks mitu kaadrit ja lasta mudelil ise õppides otsustada, kuidas neis sisalduvat infot kombineerida. Pildid kas laotakse üksteise otsa, tekitades tensori, millel on rohkem kui kolm kanalit, või töödeldakse esmalt ühekaupa, luues igaühest kompaktse vektoresituse, mida siis edasi kombineerima hakatakse. Videotöötluse rakendusi on lõputult palju. Huvitava näitena võib tuua videointervjuu salvestuse alusel inimese isiksuse ennustamise, et palgata ainult teatud isikuomadustega töötajaid. See on eetiliselt väga piiripealne ja diskrimineerimisohuga lahendus, aga kümnete tuhandete töötajatega korporatsioonid, mis saavad sadu tuhandeid taotlusi oma vabadele töökohtadele, on kuuldavasti selliseid lahendusi katsetanud. Kindlasti aitab see vähendada personaliosakonna koormust, kui teatud kandidaadid automaatse lahendusega kohe välistada, aga kas näiteks introvertide diskrimineerimine ja automaatne välistamine oleks aktsepteeritav?

6.2.4 Piltide genereerimine

Märgakem, et segmenteerimisülesannete puhul saab mudeli väljundit kujutada algse pildiga sama suure pildina, mille värvid näitavad pikslite kuuluvust mingile objektile või mingisse klassi. Kui me midagi sellist mudeli väljundina genereerida suudame, siis tekib loomulikult ka küsimus: kas me suudaksime mingi sisendi alusel ka loomulikuna näivaid pilte luua?

Tegemist on siis masinnägemise pöördülesandega: saades sisendiks objektitüübi, loo selle objekti koha näidispilt. Päriselt masinnägemise alla sellist ülesannet me pigem ikkagi ei liigitaks, sest sisendiks pole pilt ja tegu pole nägemisega ega pildi sisu mõistmisega. Siiski mainime seda ülesannet, sest see on muutunud ühiskondlikult oluliseks või isegi ohtlikuks. Tänapäeval on võimalik muuta olemasolevate piltide stiili või sisu (joonis 6.14, stiiliülekanne, ingl k *neural style transfer,* (Park, 2019)), neilt objekte eemaldada ja asemele genereerida loomulikuna näiv taust (Liu, 2018) (joonis 6.15) või sõnalise päringu alusel soovitud sisuga pilt luua (Rombach, 2022). Sellised ülesanded on matemaatiliselt keerulised defineerida, sest samale sisendile võib vastata lõputu hulk pilte, on palju võimalikke õigeid väljundeid. On oluline, et mudel ei looks sõna „koer“ alusel pilti, millel on poolenisti valge, poolenisti pruun, poolenisti väike lühikarvaline, poolenisti suur ja pikakarvaline koer, sest kõik keed variandid ju sõnaga „koer“ sobivad. Pilt peab olema ruumiliselt koherentne, ei tohi sisaldada kolme silma ega seitset käppa. Samuti ei tohi sellel olla isegi lokaalseid artefakte, eksimusi, mis kohe silma torkavad.



***Joonis 6.15. Närvivõrkudel põhinev stiiliülekanne**. Esimese veeru pildi sisule on rakendatud teise veeru pildi stiili. Tulemuseks on kolmanda veeru pilt. Keda selline lahendus lummab, võib Tartu Ülikooli Delta õppehoonest leida ekraani, mis sel viisil reaalajas veebikaamera pilti stiliseerib. Astudes veebikaamera vaatevälja, näeb kasutaja ekraanil iseennast maalituna kuulsate maalide stiilis.*[22]

Seetõttu on piltide genereerimisel olnud populaarne genereerivate võistlusvõrkude lähenemine (Goodfellow, 2014), mis loob korraga pilte genereeriva võrgu ning pilte kunstlikeks ja päris piltideks ennustada prooviva võrgu, mis omavahel võistlevad. Genereeriv võrk püüab luua nii loomulikke pilte, et kunstlikke pilte tuvastav võrk neid päris piltidest eristada ei suudaks. Siiski on viimastel aastatel edukamaks osutunud difusioonil ja transformer-arhitektuuril baseeruvad lahendused nagu StableDiffusion (Robach, 2022) ning DALL-E (Ramesh, 2022).

[22] Allikas: (Park, 2018), MIT litsents.

Joonis 6.16. Puuduva pildiosa genereerimine. *Kas soovite kogemata pildile jäänud isiku eemaldada? Pole probleemi. Märkige ära eemaldamist vajav osa, segmentatsioonivõrk tuvastab objekti piirjooned ja genereeriv võrk täidab puuduva pildiosa.*[23]

Lisaks piltide või pildiosade loomisele saab pilt pildiks lähenemisega luua ka lahendusi, mis pildi resolutsiooni suurendavad (ingl k *image superresolution*) või pildilt müra (peegeldused, „punased silmad“, skannitud piltidel rebenemis- või murdmisjooned) eemaldavad. Selle jaoks on vaja ainult kvaliteetseid pilte, mida kasutatakse väljundina. Sisend luuakse neid pilte lihtsate arvutuslike meetoditega „rikkudes“, kas suurust vähendades või müra lisades. Võrk õpib rikutud algseid pilte taastama ja see oskus üldistub loodetavasti ka uutele, seni nägemata piltidele, millel on mitte meie tekitatud, vaid loomulik müra.

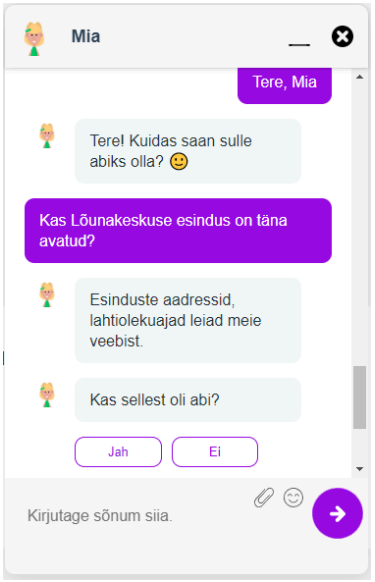
6.3 Loomuliku teksti töötlus

Enamik meist on kokku puutunud rämpsposti filtriga, mis tuvastab massiliselt ja sisutult saadetud reklaami meie e-kirjade hulgast (isegi kui see filter töötab märkamatult taustal). Kindlasti on kõigile tuttav ka automaattõlge, kasvõi näiteks Tartu Ülikooli Neurotõlge või ka Google Translate[24], mis tõlgib tekste automaatselt rohkem kui 100 keele vahel. Jah, muidugi teeb see kohati naljakaid või ootamatuid vigu, eriti kui tõlkida eesti keelde, kuid siiski suudab see enamasti sisendteksti sisu väljundis kajastada. Mõnda aega tagasi lisas Telia oma veebilehele juturoboti nimega Mia (vt joonis 6.17), mis suudab pidada kasutajatega vestlust ja vastata tüüppäringutele. Selliseid juturoboteid võib pidada eelkäiaks tänapäeva keelemudelitele nagu ChatGPT.

Keeletehnoloogia ülesanded võib jagada kahte gruppi: lõppkasutaja rakendused ja komponendid. Lõppkasutaja rakendustest ongi kasu tavakasutajaile, näiteks masintõlge, dialoogisüsteemid ehk juturobotid, dikteerimine ehk kõnetuvastus. Komponendid on aga sellised ülesanded, mille väljundist ei ole otse lõppkasutajaile kasu, kuid mida on tihti vaja kasutada vahesammuna lõppkasutaja rakenduste juures, näiteks sõnaliikide tuvastamine, sõnade ja lausete struktuuri analüüs, nimede ja teiste nimeüksuste leidmine tekstist, tekstide liigitamine.

Kõige olulisem on mõista, et arvuti ei saa keelest aru nii nagu meie, inimesed. Kõik keeletehnoloogilised rakendused ja lahendused on kitsa tehisintellekti näited (ingl k *artificial narrow intelligence*, ANI), mis tegelevad ainult ühe konkreetselt piiritletud ülesandega. Näiteks oskab Google Translate ainult tõlkida ega saa seejuures aru, kas tekstis on midagi naljakat, solvavat või ohtlikku. Ka rämpsposti filter ei saa aru, et kirja

[23] Allikas: GitHub/Inpaint-Anything, litsents.
[24] https://translate.ut.ee, https://translate.google.com.



Joonis 6.17. Juturobot Mia vastab inimkeeles esitatud küsimustele.

tekstis pakutakse midagi osta vms, see ainult liigitab kirja teksti kasulikuks või rämpspostiks. Põhjus on selle juures lihtne: ei keeleteadlased ega teised inimesed tea, mida tähendab „keelest aru saamine", ja seetõttu ei oska keegi sellist arusaamist formaalselt, arvutile arusaadavalt kirjeldada. Sellepärast peamegi piirduma üksikute keelenähtuste umbkaudse lahendamisega.

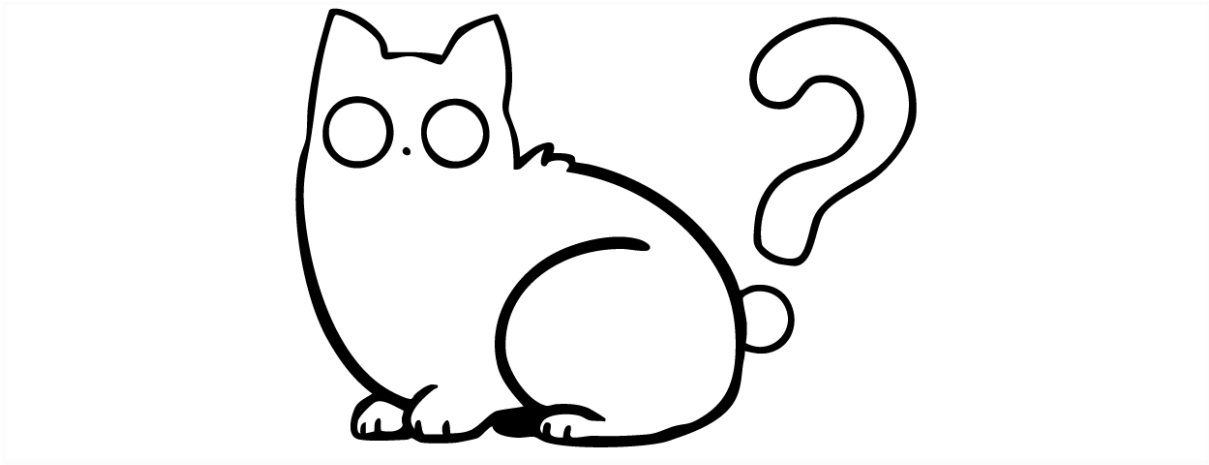
6.3.1 Miks on keel raske?

Mitmes peatükis oleme rääkinud sellest, et andmeteaduse rakendused eksivad vahel. Siin püüame selgitada, miks just keelega töötades ei tea me, kuidas keele kasutamist tehisaru abil 100% õigesti imiteerida. Teiste sõnadega: miks teevad keeletehnoloogialised rakendused vigu palju sagedamini kui inimesed.

Üks suur keele keerulisuse allikas on **mitmetähenduslikkus**. Nii sõnu kui ka terveid fraase, lauseid ja grammatilisi konstruktsioone võib sageli tõlgendada mitmel eri viisil. Vaatame näiteid joonisel 6.18.

Joonis 6.18. Näited mitmetähenduslikest sõnadest. *Keel võib tähendada nii kehaosa, suhtlusvahendit kui ka pinguldatud häälestatavat heliallikat. Jooksime võib olla tegusõna „jooksma" mineviku mitmuse 1. isiku vorm või ka „jooma" tingivas kõneviisis mitmuse 1. isiku vorm.*

Need on näited mitmetähenduslikest sõnadest, kuid ka fraasi või lause tasemel esineb samasugust mitmesust: näiteks võib tõlgendada fraasi „professori nahast portfell“ nii nahast portfellina, mis kuulub professorile, kui ka portfellina, mis on tehtud professori nahast, mille kohta ärme näidispilti siia õpikusse lisame. Veel üks näide, seekord päriselust: Tartu turu juures oli mõned aastad tagasi kuulutus tekstiga „kadunud valge sabaga kass“. Kas tegu on kassiga, kellel on kadunud valge saba, või on kadunud valge kass, kelle üks eripära on see, et tal on saba, või on hoopis kadunud kass, kelle saba on valge, ja ülejäänud kassi värvust ei ole mainitud? Inimesele tundub tihti ilmselge, milline tõlgendus on õige, kuid see info ei sisaldu otse sõnades endis.



Joonis 6.19. „Kadunud valge sabaga kass“. *Kas selles lauses on tegu kassiga, kellel on kadunud valge saba, või on kadunud valge kass, kelle üks eripära on see, et tal on saba, või on hoopis kadunud kass, kelle saba oli valge, ja ülejäänud kassi värvust ei ole mainitud?*

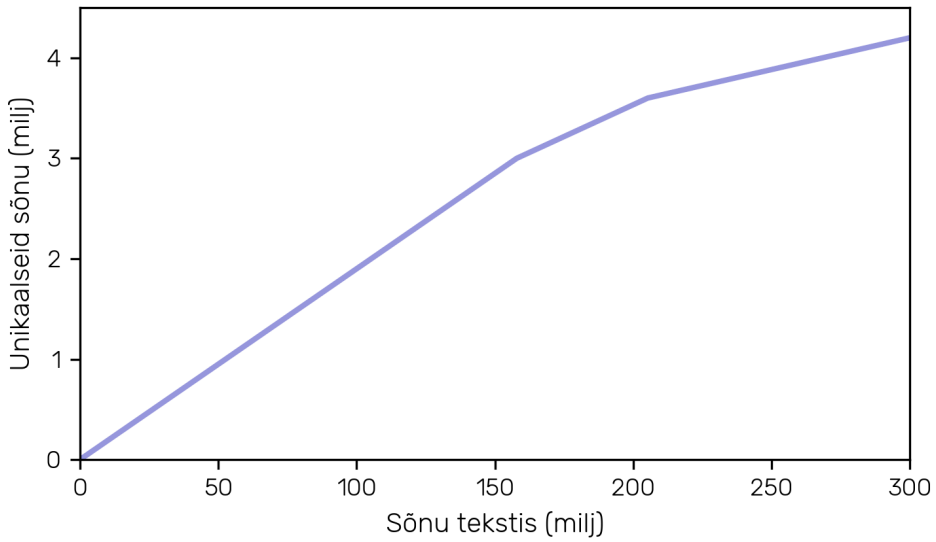
Selliseid mitmesuse näiteid esineb igas keeles ja mitmel tasandil ning see teeb keelega töötamise nii raskeks kui ka põnevaks. Inimesed lahendavad seda mitmesust, kasutades konteksti. Näiteks, kui lause sisaldab lisaks sõnale „keel“ teisi sõnu nagu „rääkima“ või „tõlkima“, siis on tõenäoline, et see „keel“ tähendab just suhtlusvahendit, mitte pinguldatud häälestatavat heliallikat muusikainstrumendil või liikuvat elundit suuõõne põhjas. Samuti on paljud mitmetähenduslikkuse näited sellised, kus ainult üks tõlgendamise variant on mõistlik. Näiteks, arvestades meie kultuurilist konteksti, on ilmselgelt aru saada, mida tähendab „professori nahast portfell“. Need kaks mitmesuse lahendamise allikat (mõistlikkus ja konteksti arvestamine) muidugi ei lahenda kõiki juhtumeid. Lisaks on need automaatsete lahenduste jaoks praktiliselt kasutud:

- arvutiprogrammid ei tea, mis on mõistlik. Lähim asendus mõistlikkusele on antud juhul see, kui sage mõni keeleline näide või selle tõlgendamisviis keeleandmetes on;
- konteksti arvestamine on arvutiprogrammide jaoks väga raske. Keeletehnoloogiliste ülesannete lahendamine juba omaette nõuab keerulisi lähenemisi ja selleks, et asju mitte veel keerulisemaks ajada, jäetakse kontekstiga arvestamine enamasti mudelitest välja.

Teine põhjus, miks keeletöötlus on keeruline, seisneb selles, et ükski inimkeel ei koosne sõnade või mõistete lõplikust nimekirjast.

- Keele kasutamise käigus ja erinevate tegurite mõjul keel muutub: sõnad kaovad sellest või tekivad juurde, ka keele reeglid muutuvad. Näiteks sada aastat tagasi ei oleks Eestis keegi aru saanud sõnadest „meem“ ja „arvuti“ ning 19. sajandi lõpus oli õige öelda „läksivad“, mitte „läksid“.
- Ühe keele sees peitub tegelikult palju erialakeeli ja žargoone. On raske piiritleda, millal lõpeb üks keel ja algab mõni selle dialekt või isegi omaette keel.
- Paljudes keeltes on olemas viljakas uute sõnade moodustamise mehhanism. Eesti keeles on selleks mehhanismiks liitsõnad: näiteks sõna „elevandilondikondiüdi“ on võib-olla kohmakas, kuid keeleliselt täiesti korrektne. Teine näide: kauplust, kus müüakse katuseid, võib nimetada „peavarjupoeks“, mis on eesti keelt rääkivatele inimestele arusaadav, kuid näiteks Google'i otsing leiab sellele 0 vastet. Lisaks sellele on eesti keeles rikas morfoloogia ja igal nimisõnal on kuni 30 erinevat vormi (14 käänet + lühike sisseütlev = 15 vormi, ainsuses ja mitmuses = 15 × 2). Tegusõnadel on samuti palju vorme, mis tähendab, et isegi tuttavad sõnad esinevad eri kujudel.

Kõik see aga tähendab, et me ei saa kunagi andmetest näha või kirja panna ühe keele kõiki võimalikke sõnu, grammatilisi konstruktsioone ega fraase. Ükskõik kui pika sõnastiku me koostame ja ükskõik kui suurt keeleandmestikku kasutame, peab ikka arvestama, et jääb selliseid sõnavorme, sõnu ja muid keelelisi elemente, mida see ei sisalda. Seda võib illustreerida graafikuga (vt joonis 6.20), mis näitab tekstiandmestiku suurust ja selles sisalduvate unikaalsete sõnavormide arvu: kui unikaalsed sõnad saaksid mingil hetkel otsa, siis läheks see kõver paremal pool horisontaalseks, see aga tõuseb kuni graafiku lõpuni välja.



Joonis 6.20: Unikaalsete sõnade arv. *Eestikeelse teksti suuruse ja unikaalsete sõnade (ehk sõnastiku suuruse) suhe. Peamine järeldus selle graafiku põhjal on: mida suurem on tekst, seda rohkem on erinevaid sõnu (sõnavorme), ehk uued sõnad ei „saa kunagi otsa“. Lähtekood.*

Loodetavasti on meil nüüdseks õnnestunud teid veenda selles, et keel on nii keeruline kui ka kindlasti põnev, ja ka selles, et juba praegusel tasemel võivad keeletehnoloogilised rakendused kasulikud olla.

6.3.2 Keeletöötluse rakendused

Teksti liigitamine

Teksti liigitamine on ülesannete hulk, mille alla mahuvad kõik keeletehnoloogia ülesanded, kus mõnda lauset või teksti tuleb klassifitseerida ühte mitmest klassist. Selle alla kuuluvad sellised lahendused nagu rämpsposti tuvastamine (liigitamine rämpspostiks ja mitte-rämpspostiks), meelsusanalüüs (liigitamine positiivseks, negatiivseks ja neutraalseks tekstiks), samuti žanri, teema või teksti tüübi tuvastamine (nt uudised, teaduslikud, juriidilised, tehnilised tekstid).

Tekste on pikka aega liigitatud klassikalise masinõppe abil, kasutades klassifitseerimismeetodeid. See masinõppel põhinev lähenemine seisneb sisendteksti kirjeldamises mingite tunnuste abil ja seejärel klassifitseerija treenimises.

Joonis 6.21. Erineva otstarbega tekstid sisaldavad erinevaid sõnu. *Vasakul enamesinenud sõnad lehelt armastuskirjad.com ja paremal sõnad põhikooli riiklikust õppekavast.*

Üks universaalne tunnuste tüüp, mida tekstide klassifitseerimisel tihti kasutatakse, kannab nime sõnahulgad (ingl k *bag-of-words*). See lähenemine seisneb selles, et iga treenimisandmestikus esinev sõna moodustab binaarse tunnuse: kas see sõna esineb selles konkreetses tekstis (või lauses) või mitte. Tulemuseks on suur hulk binaarseid tunnuseid, mis jätavad arvesse võtmata sõnade järjekorra ja esinemissageduse, ning alles jääb ainult info iga sõna olemasolu kohta. Kuigi juba intuitiivselt võib aimata, et see lähenemine on pigem naiivne, võimaldab see siiski leida selliseid võtmesõnu, mis osutavad kindlalt konkreetsele liigile või klassile.

Teksti ID	kallis	ilus	õpilane	hinne	patsient	ravikuur
tekst_1	1	1	0	0	0	0
tekst_2	0	0	1	1	0	0
tekst_3	1	0	0	1	1	1

Tabel 6.1. Tekstide kujutamine sõnahulkade kujul *ehk märkides 1 või 0 abil, kas sõna esineb tekstis või ei. Ilmselt pole keeruline selliste tunnuste järgi arvata, milline kolmest tekstist on pärit lehelt armastuskirjad.com, milline põhikooli riiklikust õppekavast ja milline arstiteaduse õppekavast.*

Targem on muidugi arvesse võtta mitte kõiki sõnu, vaid ainult neid, mille kohta me teame, et need aitavad teksti liike eristada. Üks tüüpiline võte, mida tehakse sõnahulga vähendamiseks, on mitte-sisuliste sõnade (sidesõnad, eessõnad, asesõnad jms) eemaldamine. See on mõistlik, kuna neid sõnu esineb tekstis olenemata selle liigist ja seetõttu ei anna nad infot selle kohta, mis liiki see tekst võiks olla.

Hiljuti on sügavõppepõhised lähenemised samuti toonud kvaliteetseid lahendusi tekstide liigitamise ülesannetele. Seejuures õpetatakse närvivõrke algusest lõpuni (ingl k *end-to-end*) sellisel meetodil, et teksti sõnad lähevad otse närvivõrgu sisendiks, ilma neid käsitsi tunnusteks (sõnahulkadeks vms) teisendamata.

Masintõlge

Automaatne tõlkimine inimkeelte vahel ehk masintõlge on üks esimesi keeletehnoloogia ülesandeid, mida arvutite abil üritati lahendada ja kus esimesi katseid tehti juba 1950ndate alguses. Selle aja jooksul on püütud masintõlget lahendada reeglipõhise lähenemisega (valdavalt kuni 1980ndateni), siis statistiliste mudelitega (lähenemine, mis tegi masintõlke kasutatavaks praktikas ja domineeris umbes 2014.–2016. aastani) ning hiljuti sügavõppe abil neuromasintõlkena. Siin kirjeldame pealiskaudselt just neuromasintõlget.

Neuromasintõlkes kasutatavat lähenemist nimetatakse järjendite teisendamiseks (ingl k *sequence-to-sequence* ehk *seq2seq*) ja see ei ole ainult tõlkimisele spetsiifiline idee. See lähenemine toetab suvaliste järjendite (nt lausete) teisendamist teisteks järjenditeks (nt teise keele lauseteks). Juhul, kui andmed sisaldavad tõlkenäiteid (lause ja selle tõlge), õpib taoline süsteem tõlkimist. Kui andmestik koosneb hoopis vigaste lausete parandamise näidetest (vigane lause ja parandatud lause), lause lihtsustamisest või ümber sõnastatud lausetest, õpib süsteem autokorrektuuri, lihtsustamist või parafraaside genereerimist.

Kõigepealt vaatleme tõenäosusliku teksti genereerimist närvivõrgu abil sihtkeeles, ilma lähtekeele sisendit vaatamata. Korraldame genereerimist „vasakult paremale“: kõigepealt ennustab närvivõrk lause esimese sõna tõenäosusjaotust ja selle abil võime välja valida lause esimene sõna. Seejärel ennustab närvivõrk teise sõna tõenäosusjaotust, mis on

tingitud esimesest sõnast. Kolmanda sõna tõenäosusjaotus on tingitud esimesest kahest sõnast jne. Sellist genereerimist nimetatakse autoregressiivseks.

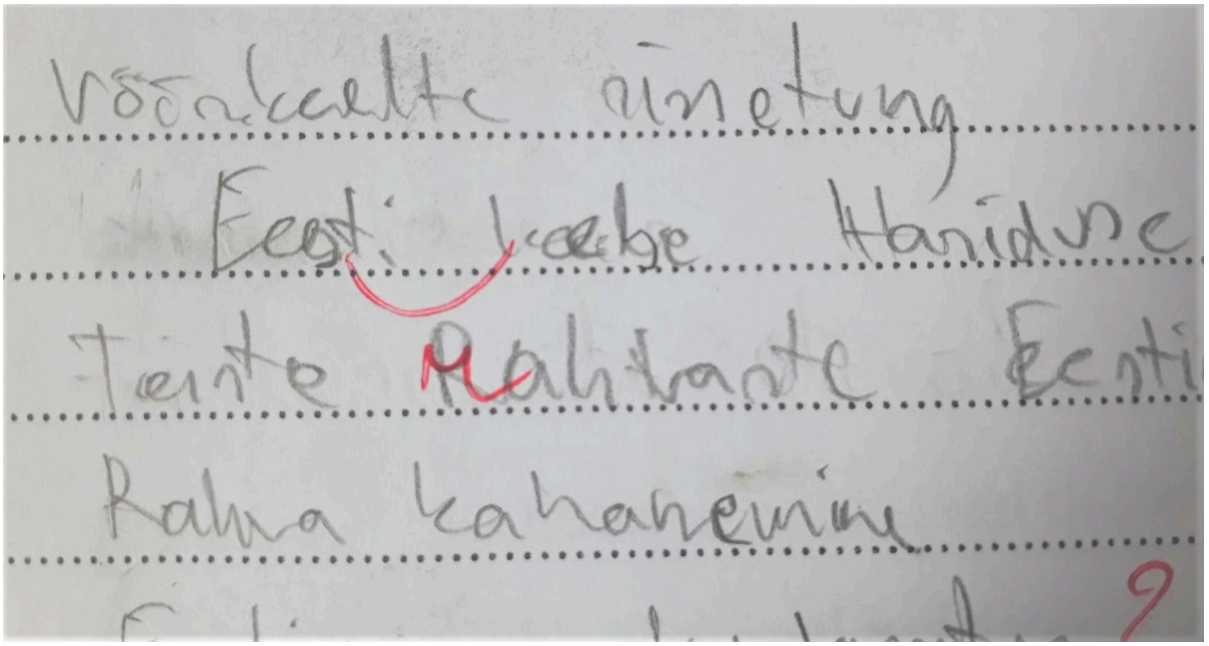
Näiteks, otsustatakse juhuslikult/tõenäosuslikult, mis võiks olla esimene sõna. Olgu selleks „kui“. Edasi, teades, et esimene sõna oli „kui“, genereerime järgmise sõna, milleks olgu näiteks nimi „Arno“. Teades, et lause algus on „kui Arno“, genereerime järgmise sõna „isaga“ jne.

Sellise lause jätkamisega saab inimene hästi hakkama: kerge on jätkata lauset „kas te soovite teed või …?“ sõnaga „kohvi“ (isegi kui „õlut“ oleks samuti grammatiliselt ja tähenduse poolest täiesti aktsepteeritav variant). Samuti on kerge tuvastada, et lause algust „lugupeetud …“ võib jätkata mitmel viisil, nagu „daamid ja härrad“, „vanemad“, „tudengid“, „reisijad“ vms, aga tõenäoliselt mitte „kurgid“ või „siniseks“. Samamoodi õpib tehisnärvivõrk juba olemas olevale lausealgusele järgneda võivate sõnade jaotust, et genereeritud lause saaks grammatiliselt korrektne ja sorav.

Et aga juhuslikult genereeritud lause asemel genereerida mingile sisendlausele vastav tõlge, peame lisama genereerimisingimuste hulka ühe väga olulise lisaklausli. Me soovime genereerida järgmise sõna väljundlauses, arvestades nii seni genereeritud väljundiga (just nagu enne) kui ka kogu meie sisendlausega (tõlkimist vajava lausega sisendkeeles). Tõlkides inglise keelest eesti keelde, olgu sisendiks näiteks „*dear travellers*“ ja juba genereeritud osaliseks väljundiks „lugupeetud“. Võttes arvesse nii eesti keele genereerimise reegleid (sõnale „lugupeetud“ järgnevad tõenäoliselt teatud sõnad) kui ka sisendlauset, on lihtne jätkata lauset kui „lugupeetud reisijad“.

Seq2seq meetodite ehitus:

- sisendi sõnu töötleb kodeerija (ingl k *encoder*), mis leiab igale sõnale vektoresituse, mis sõltub nii sõnast endast kui ka ülejäänud lause sisust – et arvestada konteksti. Nii on näiteks sõna „sai“ vektorid erinevad, olenevalt sellest, kas tegu on tegusõna (nt *ta sai pähe*) või nimisõnaga (nt *sai on laua peal*);
- väljundit genereerib dekooder (ingl k *decoder*), mis läheneb genereerimisele autoregressiivselt ehk iga järgmise sõna genereerimine tugineb juba genereeritud sõnadele;
- dekooderi ja kodeerija vahel on nn tähelepanumehhanism, mis annab dekoodrile ligipääsu terve sisendlause vektoritele, aga arvestab ka seda, millised sisendi elemendid on vajalikud, et järgmist väljundsõna genereerida;
- seda, milliste vektoritega esitada sisendit ja juba genereeritud väljundit ning kuidas otsustada, millised sisendsõnad on vajalikud milliste väljundsõnade genereerimiseks, õpib tehisnärvivõrk korraga, algusest lõpuni (ehk *end-to-end*-stiilis).



Joonis 6.22. Kirjavigade parandamine. *Põhikooliõpilase õigekirjavigu parandab tema eesti keele õpetaja.*

Viimasel ajal domineerib grammatiliste vigade parandamises samasugune lähenemine nagu masintõlke puhul: näiteks *seq2seq*, täpsemalt tähelepanumehhanismiga kodeerija-dekooder-tüüpi tehisnärvivõrk. Selle ülesande eripära on aga selles, et sisendiks on vaja lauseid, mis võivad sisaldada grammatilisi vigu, ja väljundiks on samas keeles laused, kus need vead on parandatud. Põhiline erinevus masintõlkest seisneb selles, et tõlgitakse maailmas massiliselt, mis tähendab, et tõlkenäiteid, mida saab treeningandmetena taaskasutada, on palju rohkem. Grammatilisi vigu aga parandatakse oluliselt vähem. Konkreetselt eesti keele puhul on veaparandustega andmestik väga väike (vähem kui 30 000 lauset) ja väiksematel keeltel (võru, komi jt) sellist andmestiku üldse ei olegi.

Võib muidugi proovida „tõlkida“ eesti keelest eesti keelde; seda saab ise katsetada TÜ tõlkedemo veebilehel (vt joonis 6.23). Selle lahenduse põhimõte seisneb selles, et tõlkimise jaoks kasutatakse mitmekeelset tehisnärvivõrku, mida õpetati inglise-eesti, eesti-inglise, vene-eesti, eesti-läti ja teiste keelepaaride tõlkenäidete abil. Tulemusena aga oskab tõlkesüsteem ka teisendada eestikeelseid sisendsõnu vektoriteks (kodeerija abil) ja nende järgi genereerida eestikeelset väljundit. Genereerimist on dekooder õppinud korrektsete tekstide põhjal ja seega genereerib enamasti korrektseid lauseid, ilma vigadeta.

Muidugi ei ole see lahendus ideaalne ja on vigu, mille parandamisega see lahendus hakkama ei saa. Parima tulemuse saamiseks kasutatakse sünteetilisi andmeid tehisvigadega:

- võetakse jälle appi inimvigadega andmestik, kuid selleks, et õpetada vigade genereerijat: sisendiks on korrektne ja väljundiks vigadega lause;

- seda rakendatakse suurele hulgale korrektsetele lausetele (nt raamatutest, Vikipeediast või niisama veebist), et lisada sinna vigu. Tulemuseks võib tekkida mitu miljonit lausepaari, kus on sünteetiliselt lisatud vigadega laused ja lähtelaused;
- nüüd võib paari järjekorra ümber pöörata ning kasutada tehisvigadega lauset sisendina ja korrektset lauset väljundina. Parima tulemuse saavutamiseks kombineeritakse sünteetilist andmestiku lähteandmestikuga, mis sisaldab inimeste vigu. Katsed näitavad, et selline lähenemine annab isegi paremaid tulemusi, kui ChatGPT-l teksti vigu parandada palumine.

Joonis 6.23. Neurotõlge. *TÜ tõlkedemo lehel saab tõlkida vigast eestikeelset teksti korrektsesse eesti keelde. Näiteks kui sisestada „me parandab vead", on väljundiks „me parandame vigu".*

6.4 Alusmudelid

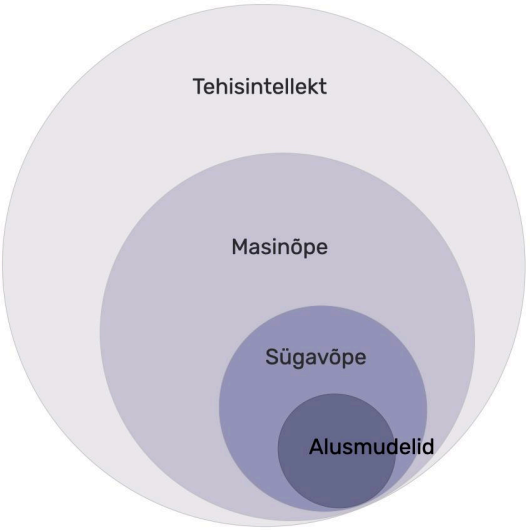
Alusmudelitest (ingl k *foundation models*) on saanud tänapäeva andmeteaduse maailma oluline osa, millele tuginevad paljud edasijõudnud rakendused ja teenused erinevates valdkondades.

Termini „alusmudel" pakkus välja Stanfordi Ülikooli inimkeskse tehisintellekti instituudi (ingl k Stanford University Institute for Human-Centered Artificial Intelligence[25], HAI) alusmudelite uurimiskeskus (ingl k Center for Research on Foundation Models[26], CRFM) 2021. aastal. Alusmudeli definitsioon esitati CRFM-i artiklis (Bommasani, 2021).

[25] https://hai.stanford.edu/.
[26] https://hai.stanford.edu/news/introducing-center-research-foundation-models-crfm.

Definitsioon
Alusmudel on mudel, mis on treenitud laiaulatuslike andmete peal, kasutades suures mahus isejuhendatud õpet, ja mida saab kohandada, näiteks peenhäälestamise kaudu, erinevateks edasisteks ülesanneteks.

Andmeteaduse vaatenurgast kuuluvad alusmudelid sügavõppe mudelite klassi (joonis 6.24). Traditsioonilised sügavõppe mudelid on sageli treenitud konkreetsete ülesannete jaoks. Need mudelid ei pruugi olla sama mitmekülgsed kui alusmudelid ja vajavad sageli rohkem kohandatud andmeid ja treenimist konkreetsete probleemide lahendamiseks. Alusmudelid on suured miljardite parameetritega sügavõppe mudelid, mis põhinevad sügavatel närvivõrkudel ja isejuhendatud õppel. Viimase kolme aasta jooksul on alusmudelid laiendanud meie arusaamist sellest, mis on võimalik. Näiteks mudel GPT-3, mis oli populaarse ChatGPT[27] rakenduse aluseks, sisaldab 175 miljardit parameetrit. Kasutajaliidese kaudu saab anda mudelile loomulikus keeles käske ja see suudab täita laia valikut ülesandeid. Mudel saab lahendada isegi paljusid ülesandeid, mille jaoks ta ei ole spetsiaalselt treenitud, nagu näiteks teksti tõlkimine.



Joonis 6.24. Tehisintellekti mudelite klassid. *Joonisel on kujutatud kontseptuaalne skeem, mis näitab tehisintellekti, masinõppe, sügavõppe ja alusmudelite seoseid. Masinõpe on tehisintellekti alamvaldkond, mis keskendub algoritmidele ja mudelitele, mis õpivad olemasolevatest andmetest. Sügavõpe on masinõppe spetsiifiline alamvaldkond, mis kasutab sügavaid tehisnärvivõrke suuremahuliste ja keerukate andmete analüüsimiseks. Alusmudelid on sügavõppe mudelid, mis on treenitud laiaulatuslike andmete peal ja mida saab kohandada erinevate ülesannete lahendamiseks.*

6.4.1 Alusmudelite võimsus ja mitmekülgsus

Alusmudelite võimsus ja mitmekülgsus tuginevad viiele olulisele käsitlusviisile: siirdeõpe, isejuhendatud õpe, arvutusvõimsus, mudeli võimekas arhitektuur ning suuremahulised mitmekülgsed treeningandmed. **Siirdeõpe** võimaldab ühe ülesande lahendamisel

[27] https://openai.com/index/chatgpt/.

saadud teadmisi n-ö üle kanda teise ülesande lahendamiseks. Sügavõppe mudeli loomise kontekstis see tähendab, et me saame mudelit treenida esmalt ühel ülesandel, mille jaoks meil on palju andmeid, näiteks ennustada tekstis puuduvat sõna, ja hiljem kohandada mudeli teise ülesande lahendamiseks, näiteks teksti meelsuse ennustamiseks. Siirdeõpe ongi see, mis võimaldab alusmudelitel nii hästi hakkama saada väga paljude erinevate ülesannetega. Siirdeõpe, kus kasutatakse märgendatud andmeid eeltreenimise faasis, oli populaarne mudelite arendamise viis vähemalt viimased kümme aastat. Kuid andmete märgendamine ehk annoteerimine on kallis ja aeganõudev. Enamik praeguseid alusmudeleid kasutavat **isejuhendatud õpet** (ingl k *self-learning, self-supervised learning*). Mudelid peavad õppima andmete mustritest ja genereerima väljundeid selle põhjal ilma seotud märgenditeta.

Alusmudelite treenimine vajab suuri arvutusressursse ja pilvepõhist taristut, nagu näiteks Microsoft Azure, AWS või Google Cloud. Arvutusjõu suurenemine, eriti GPU protsessorite jõudluse ja mälu kasv, võimaldab treenida suuri ja keerukaid mudeleid.

Keerukate alusmudelite **transformer-arhitektuur** on alusmudelite selgroog, mis võimaldab tõhusat andmetöötlust. Transformer-arhitektuuri tutvustas 2017. aastal teadustöö „Attention is All You Need“ (Vaswani jt, 2017) ja see on põhjalikult mõjutanud sügavõppe valdkonda. Transformeri keskseks ideeks on **enesetähelepanu** (ingl k *self-attention*) mehhanism, mis võimaldab mudelil keskenduda sisendi olulistele osadele. See tähendab, et mudel on võimeline mõistma konteksti ja seoseid andmetes. Erinevalt varasematest, näiteks RNN- ja LSTM-arhitektuuriga, mudelitest võimaldab transformer-arhitektuur ka paremat paralleeltöötlust, mis teeb treenimise palju kiiremaks ja tõhusamaks. Transformer-arhitektuuri saab kohandada paljude erinevate ülesannete jaoks, sealhulgas loomuliku keele töötlus, teksti genereerimine, küsimustele vastamine, pilditöötlus.

Viimase koostisosana, suuremahulised ja mitmekülgsed treeningandmed annavad mudelitele vajalikke teadmisi, et lahendada laia valikut ülesandeid tõhusalt ja täpselt.

6.4.2 Alusmudelite kategooriad

Alusmudelid hõlmavad laia valikut tehnoloogiat, mis on kohandatud erinevate andmetüüpide ja ülesannete jaoks, sealhulgas suured keelemudelid tekstitöötlusega seotud ülesannete jaoks, masinnägemise mudelid piltide ja videote analüüsiks, multimodaalsed mudelid mitut tüüpi andmete integreerimiseks ning kõnemudelid kõnetöötluseks.

Alusmudelid saab jaotada mitmesse kategooriasse, olenevalt nende rakendusalast ja kasutusviisist.

- **Suured keelemudelid** (ingl k *large language models*, LLMs) keskenduvad tekstitöötlusele (sh loomuliku keele töötlusele) ning on seotud ülesannetega nagu teksti genereerimine, tõlkimine, kokkuvõtete tegemine ja küsimustele vastamine. Sellesse kategooriasse kuuluvad näiteks mudelid GPT-4, Claude 3, Llama 3, BERT.

- **Masinnägemise alusmudelid** keskenduvad piltide ja videote töötlusele, sealhulgas objektituvastusele, objektide klassifitseerimisele ja segmenteerimisele. Sellesse kategooriasse kuuluvad näiteks mudelid ViT, DETR, SAM.
- **Multimodaalsed mudelid** suudavad töödelda ja integreerida mitut tüüpi andmeid, näiteks teksti, pilte ja heli, võimaldades lahendada keerukamaid ülesandeid, mis nõuavad erinevate andmetüüpide mõistmist ja kombineerimist. Sellesse kategooriasse kuuluvad näiteks Google'i Gemini mudel ja OpenAI Clip.
- **Kõnetöötluse mudelid** keskenduvad kõne tuvastusele, sünteesile ja töötlusele. Sellesse kategooriasse kuuluvad näiteks Wave2Vec ja Whisper.

Järgmistes alapeatükkides tutvustame alusmudeleid kahes andmeteaduse rakendusvaldkonnas: masinnägemine ja loomuliku keele töötlus, kus räägime suurtest keelemudelitest.

6.4.3 Alusmudelid masinnägemises

Masinnägemise ülesande keerukus seisneb pildilt mingi otsuse tegemiseks vajaliku info leidmises. Pildid on mitmekesised ja inimesed ei suutnud ise välja mõelda piisavalt hästi üldistuvat ruumiliste tunnuste kogumit, mille pildilt leidmine annaks piisavalt selgeid vihjeid pildi sisu kohta, et selle alusel otsus teha (vt ptk 6.2). Isegi ainult piltide klassifitseerimiseks ei piisanud inimeste leiutatud tunnustest ja arenguhüpe toimus ise ülesande lahendamise jaoks kasulikke ruumilisi tunnuseid õppivate konvolutsiooniliste võrkude kasutuselevõtuga. Siiski, **võrgud õpivad leidma tunnuseid, mis on kasulikud lahendatava ülesande jaoks**, kuid optimaalsed ruumilised tunnused näotuvastuse ja liiklusstseenide segmenteerimise jaoks pole samad. Ühte ülesannet lahendama õpetatud võrk ei anna meile ikkagi universaalseid tunnuseid, mida piltidelt otsima peaks. Kas see tähendab, et iga veidigi uudse ülesande jaoks peab koguma suure andmestiku, et treenida sadade kihtidega võimas nägemismudel?

Õnneks on erinevatel piltidel siiski üht-teist sarnast ja sarnastel andmetel treenitud mudel on kasulik treenimise alguspunkt. Uuel ülesandel treenitava mudeli kaalud algväärtustatakse mõne teise, sarnasel ülesandel treenitud mudeli kaaludega ja õppimist alustatakse sellest seisust. Sellist õpitud ühendustugevuste taaskasutamist nimetatakse **siirdeõppeks** (ingl k *transfer learning*) ja mudeli kohandamist uuele ülesandele **peenhäälestamiseks** ehk **täppistreenimiseks** (ingl k *finetuning*). Sel viisil piisab hea mudeli loomiseks vähemast hulgast märgendatud näidetest, kuid täppistreenimine on ettevaatlikkust nõudev protsess. Kuna meie ülesandespetsiifilised andmestikud on tõenäoliselt väikesed, on oht üsna võimekad mudelid ülesobitada. Samuti on oht eelmistel andmetel õpitu unustada või esimestes (sisendile lähemates, sügavates) kihtides toimuv sassi ajada (ingl k *catastrophic forgetting*). Õigeid meetodeid kasutades on aga tegu väga kasuliku tehnikaga, mis säästab palju ressursse.

Lisaks on võimalik panna mudeleid lahendama paljusid ülesandeid korraga. Üldistatuna võib ette kujutada, et selline võrk koosneb ülesandeagnostilisest enkooderist (võrgu selgroog, ingl k *backbone*) ja paljudest ülesandespetsiifilistest peadest (ingl k *output heads*). Ükskõik millist paljudest masinnägemise ülesannetest see võrk parasjagu

lahendab, kasutatakse alati enkooderit sisendpildi viimiseks vektoresitusele (ingl k *embedding*). Iga „pea“ on mõnekihiline tehisnärvivõrk, mis selle vektoresituse alusel lahendab ühte kindlat ülesannet, näiteks lokaliseerimine või segmentatsioon. Väga paljudel eriilmelistel (värvilised ja mustvalged fotod, maalid, joonistused, meditsiinilised pildid jne) ning erinevaid ülesandeid sisaldavatel andmestikel treenitud visuaalne enkooder ongi **universaalne pildilt oluliste tunnuste otsija**.

Mitme ülesande jaoks kasulikke vektoresitusi võivad mudelid õppida ka muul viisil, näiteks isejuhendatud õppe abil (ingl k *self-supervised learning*), eesmärk on aga sama: luua universaalne piltide esitusviis vektorina. Uuemad universaalsed pildi vektorkujule eeltöötlejad on treenitud sadadel miljonitel märgendamata piltidel just isejuhendatud õppe abil. See õppimisviis kasutab erinevatest andmestikest pärit pilte märgenditena ja loob sisendid nende baasil. Tuntuimad kolm viisi on pildilt osa kustutada ja võrgul see taastada lasta, pilt mustvalgeks teha ja võrgul see taas värvida lasta ning pilt juppideks lõigata ja võrgul see pusle uuesti kokku panna lasta[28]. See on ainuke viis tekitada nii suurte mudelite loomiseks piisavalt treeningandmeid, sest maailmas pole olemas märgendatud pildiandmestikke, mis sisaldaksid miljardeid pilte. Nende ülesannete lahendamiseks väga eriilmelistel piltidel (mida iganes internetist leida võib) peab võrk piltide olemust, sisu ja loogikat üsna hästi mõistma. Vektoresitused, mille võrk enda sees igast pildist loob, peavad olema inforikkad, et kõiki neid ülesandeid lahendada. Neid samu inforikkaid vektoresitusi saab kasutada ka teiste küsimuste esitamiseks nende samade piltide kohta.

Definitsioon
Masinnägemise alusmudel on tehisnärvivõrk, mis saab sisendiks pildi ja tagastab sellest tuletatud väljundi, mis on kasulik paljude erinevate ülesannete lahendamise sisendina.

Kasulik võib olla ka mitte pildi vektoresitus, vaid näiteks segmentatsioon või muu mõistetavam väljund. Seega liigituvad ka erinevatele andmetele (portreefotod, aerofotod, joonistused jne) üldistuvad või uuele ülesandele väga kergesti ümber kohandatavad võrgud alusmudeliteks. Võrk, mis segmenteerib ükskõik mis tüüpi objekte (aimates, kus mingi objekt algab ja lõpeb, ilma selle objekti tüüpi teadmata), liigitub alusmudeliks (vt Segment anything model (Kirillov, 2023)). Võrk, mis lubab pildilt sõnalise päringu alusel ükskõik milliseid objekte lokaliseerida, liigitub samuti alusmudeliks (OWLv2 mudel (Minderer, 2022)). Üldistumist uutele kasutujuhtudele ilma peenhäälestamise ja märgendatud sisendita nimetatakse nullsammuga üldistumiseks (ingl k *zero-shot generalization*). Alusmudeliks võib liigituda ka objektituvastuse võrk, mida saab panna uut tüüpi objekte tuvastama vaid mõne näite abil. Sel juhul on tegu mõne sammuga üldistumisega (ingl k *few-shot generalization*), sest mudel vajab kohanemiseks mõnda näidet.

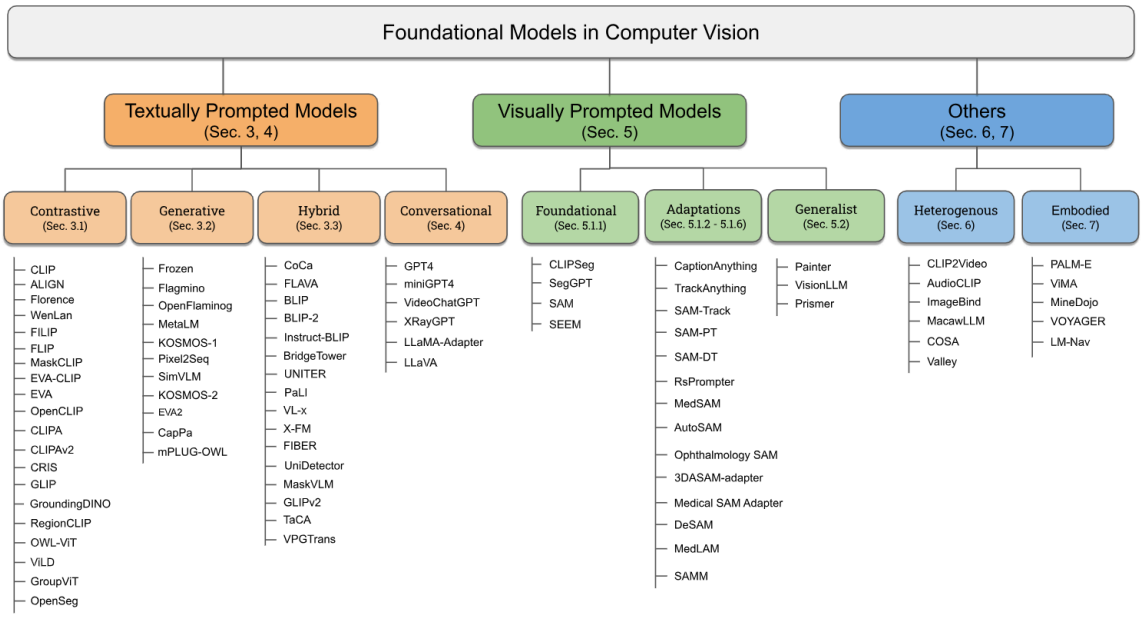
[28] Pange tähele, et siin on inspiratsiooni saadud väga edukaks osutunud keelemudelitest, sh keeletöötluse alusmudelitest, mis klassikaliselt õpivad lauses järgmist sõna, puuduvat sõna või järgmist lauset ennustama.

Tabelis 6.2 loetleme mõned aastal 2024 ajakohased mudelid, mida kasutatakse siirdeõppes või alusmudelitena. On oluline märgata, et need on üldised mudelid, mis peaksid hakkama saama igasuguste andmetega. Igas valdkonnas (teksti- ja näotuvastus, liiklusstseenide segmenteerimine jne) avaldatakse igal aastal sadu mudeleid (teadustööde raames), mis on spetsialiseerunud ühele ülesandele ja võivad veelgi paremaid tulemusi anda.

Alusmudel	Arendaja	Sisend	Väljund	Kohandamine	Litsents
CLIP	Open AI	Pilt, lause	Vektoresitused	Nullsamm	Vabavara
DinoV2	Meta AI	Pilt	Vektoresitus	Nullsamm	Vabavara
ImageBind	Meta AI	Pilt, heli, tekst, termokaamera pilt jne	Vektoresitused	Nullsamm	Mitteäriline litsents
RAM	Erinevad asutused	Pilt	Tuvastatud objektid, märksõnad	Nullsamm	Vabavara
SAM, SEEM	Meta AI	Pilt, päring erinevates formaatides	Segementatsioon	Nullsamm	Vabavara
OWL v2	Google	Pilt, sõnaline päring	Lokaliseerimine	Nullsamm	Vabavara
YOLO v8	Ultralytics	Pilt või video	Klassifitseerimine, lokaliseerimine, segmenteerimine, asendutuvastus, objektide jälgimine üle kaadrite	Vajab täppistreenimist märgendatud andmetel	Vabavara
ConvNeXt	Meta AI	Pilt	Klassifitseerimine, lokaliseerimine, segmenteerimine	Vajab täppistreenimist märgendatud andmetel	Vabavara

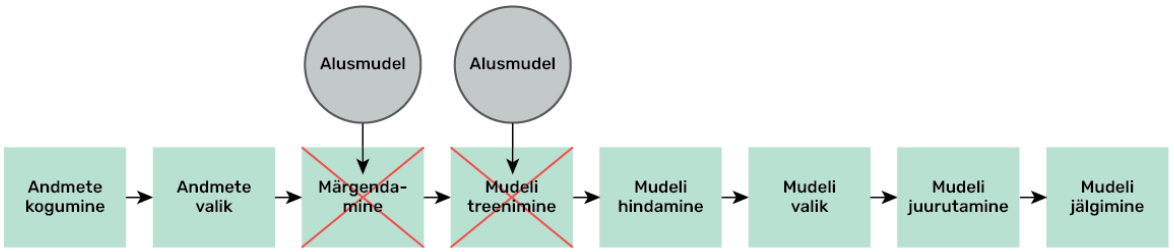
Tabel 6.2. Olulised masinnägemise mudelid 2024. aasta alguse seisuga. *Alusmudelite väljund on vektoresitus. Kui alusmudel on mitmemodaalne, on väljund vektoresitus iga sisendi kohta. Need vektoresitused on sama sisu korral sarnased – seega fraas „pilt, millel on must pikakarvaline koer“ annab Eukleidilise kauguse mõttes sarnase vektori kui pilt, millel selline sisu on, ja helifail, kus selline lause öeldakse.*

Et mõista alusmudelite alal toimuvat võidujooksu ja tohutult kiiret arengut, lisame veel ühe joonise. Joonis 6.25 kujutab masinnägemise alusmudelite nomenklatuuri ja näiteid igast kategooriast. Joonisel on kokku mitukümmend mudelinime. Alusmudeleid kasutama hakates on vaja endale selgeks teha, mis on iga mudeli eelised ja täpne võimekus, et sellest suurest hulgast sobiv valida.



Joonis 6.25. Masinnägemise alusmudelite mitmekesisus. *Potentsiaalselt kasulikke mudeleid on palju, aga igaüks neist on veidi erineval viisil ja erinevatel andmetel loodud. Aimata, milline neist üldistuks kõige paremini just teid huvitavale ülesandele, pole lihtne.*[29]

On ilmselge, et niivõrd üldiste tööriistade ilmumine masinnägemise valdkonda muudab seda, kuidas tööd tehakse. Paljudel juhtudel polegi enam vaja mudelit treenida või on vaja treenida ainult mudeli „pea“ ehk mõned kihid, mis alusmudeli loodud vektoresituse alusel otsuse teeksid. See tähendab, et märgendatud andmeid koguma ei pea või on neid vaja väga palju vähem. Piltlikult öeldes oskab mudel juba näha, me peame märgendatud andmete abil ainult täpse ülesande selgeks tegema (või ise käsitsi reeglistiku kirjutama, mis vektoresitusi edasi töötleb ja vastuse annab). Joonisel 6.26 on kujutatud üks võimalik masinnägemise töövoog, kus andmete ettevalmistamise faasi peamise ülesandena on kujutatud andmete annoteerimist ehk märgendamist, mis on tõesti ajakulukas ülesanne, eriti segmenteerimise puhul. Nii annoteerimise kui ka mudeli loomise (tüübi valmine, optimeerimine) saab lihtsalt vahele jätta, kui meil on olemas sobivat ülesannet lahendav alusmudel[30]. Masinnägemise lahenduste arendamine läheb seeläbi märgatavalt lihtsamaks ja rohkematele inimestele kättesaadavaks.



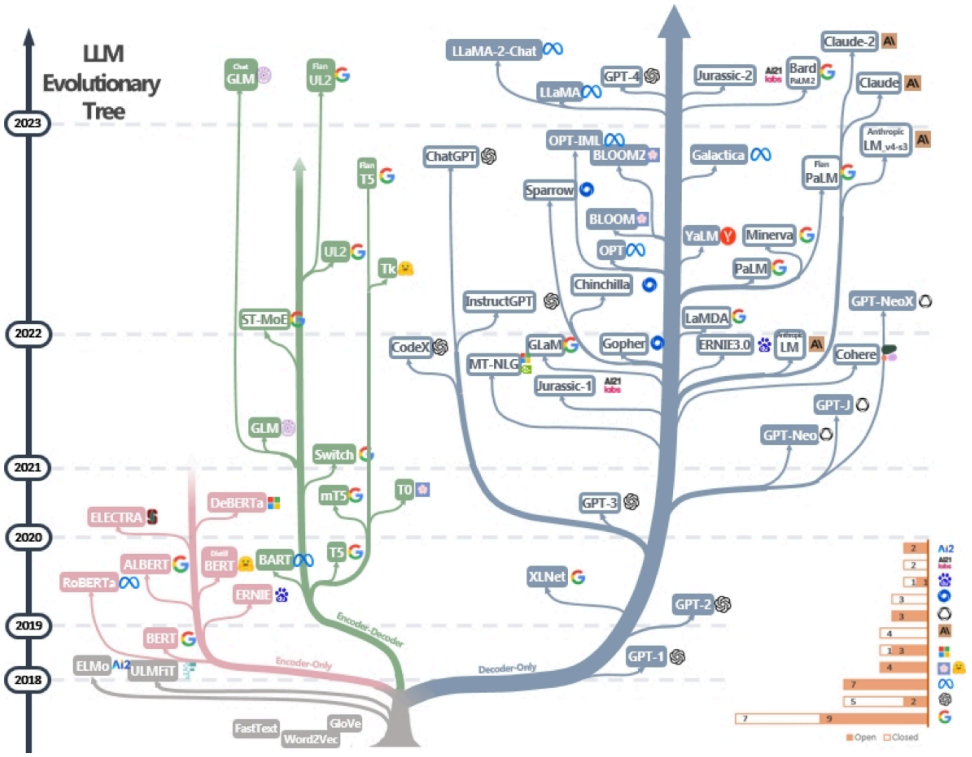
Joonis 6.26. Masinnägemise töövoog on muutumas. *Paljude ülesannete puhul pole vaja oma märgendatud andmestikku luua ja oma mudelit treenida.*

[29] *Kuvatõmmis, allikas: (Awais et. al., 2023), litsents: CC BY 4.0.*
[30] Kui sobivat alusmudelit, mis täpselt õiget ülesannet lahendaks, pole, siis on ikkagi võimalik andmed koguda ja mudel peenhäälestada oma ülesannet lahendama. Sel juhul on märgendamine ja treenimine taas töövoo osad, aga piisab väiksemast hulgast andmetest.

6.4.4 Alusmudelid keeletöötluses

Alusmudelid on generatiivsed tehisintellekti mudelid, mis võimaldavad masinatel õppida ja erinevate ülesannetega kohaneda. Need mudelid on treenitud suurel hulgal andmetel ja neid saab rakendada mitmesuguses kontekstis.

Üks oluline ja laialdaselt kasutatud alusmudelite kategooria on **suured keelemudelid** (ingl k *large language models*, LLM), mis keskenduvad keele töötlemisele (joonis 6.27). Suured keelemudelid on arenenud tehisnärvivõrgud, mis on treenitud suure hulga tekstiandmete peal. Nad suudavad genereerida arusaadavaid lauseid loomulikus keeles nagu inimene. Seeläbi jäljendavad suured keelemudelid inimintellekti. Neid nimetatakse suurteks keelemudeliteks väga suure parameetrite arvu tõttu. Nad suudavad mõista, analüüsida ja genereerida inimkeelt, mis muudab nad väärtuslikuks mitmesugustes rakendustes. Näiteks kasutatakse keelemudeleid tekstide tõlkimisel, küsimustele vastamise ülesannete lahendamisel, teksti loomisel ja tekstianalüüsis. Tihti on suured keelemudelid vestlusrobotite nagu ChatGPT[31] või virtuaalsete assistentide nagu Microsoft Copilot[32] lahutamatu osa.



Joonis 6.27. Suurte keelemudelite evolutsiooni skeem. *Joonis näitab suurte keelemudelite arengut ja omavahelisi seoseid.*[33]

Suured keelemudelid, nagu Open AI GPT-4, esindavad olulist edasiminekut keelemudelite valdkonnas. Need mudelid on treenitud kasutama tohutuid andmekogumeid ja keerukaid algoritme, mis võimaldavad neil paremini mõista semantilist tähendust ja genereerida lauseid loomulikus keeles. See areng on oluliselt

[31] https://openai.com/index/chatgpt/.
[32] https://copilot.microsoft.com/.
[33] Kuvatõmmis, allikas: (Yang et. al., 2024), litsents CC BY 4.0.

suurendanud selliste mudelite kasutusvõimalusi ja tõhusust, muutes need hädavajalikuks tööriistaks tänapäevastes tehisintellekti rakendustes.

Tänu edusammudele tehisintellekti valdkonnas ja populaarsele vestlusrobotile ChatGPT on suured keelemudelid, nagu GPT-3.5 Turbo[34] ning GPT-4[35], saanud palju tähelepanu, kuid tegelikult on suuri keelemudeleid, nii avatud lähtekoodiga kui ka kommertsiaalseid, palju rohkem (joonis 6.27, tabel 6.3).

Suured keelemudelid viitavad peamiselt transformer-arhitektuuril (Vaswani jt, 2017) põhinevatele tehisnärvivõrkude keelemudelitele, mis sisaldavad kümneid kuni sadu miljardeid parameetreid ja on eeltreenitud tohutul hulgal tekstiandmetel.

Tabel 6.3 annab ülevaate suurtest keelemudelitest ja nende peamistest omadustest.

Type	Model Name	#Parameters	Release	Base Models	Open Source	#Tokens	Training dataset
Encoder-Only	BERT	110M, 340M	2018	-	✓	137B	BooksCorpus, English Wikipedia
	RoBERTa	355M	2019	-	✓	2.2T	BooksCorpus, English Wikipedia, CC-NEWS, STORIES (a subset of Common Crawl), Reddit
	ALBERT	12M, 18M, 60M, 235M	2019	-	✓	137B	BooksCorpus, English Wikipedia
	DeBERTa	-	2020	-	✓	-	BooksCorpus, English Wikipedia, STORIES, Reddit content
	XLNet	110M, 340M	2019	-	✓	32.89B	BooksCorpus, English Wikipedia, Giga5, Common Crawl, ClueWeb 2012-B
Decoder-only	GPT-1	120M	2018	-	✓	1.3B	BooksCorpus
	GPT-2	1.5B	2019	-	✓	10B	Reddit outbound
Encoder-Decoder	T5 (Base)	223M	2019	-	✓	156B	Common Crawl
	MT5 (Base)	300M	2020	-	✓	-	New Common Crawl-based dataset in 101 languages (m Common Crawl)
	BART (Base)	139M	2019	-	✓	-	Corrupting text
GPT Family	GPT-3	125M, 350M, 760M, 1.3B, 2.7B, 6.7B, 13B, 175B	2020		×	300B	Common Crawl (filtered), WebText2, Books1, Books2, Wikipedia
	CODEX	12B	2021	GPT	✓	-	Public GitHub software repositories
	WebGPT	760M, 13B, 175B	2021	GPT-3	×	-	ELI5
	GPT-4	1.76T	2023	-	×	13T	-
LLaMA Family	LLaMA1	7B, 13B, 33B, 65B	2023	-	✓	1T, 1.4T	Online sources
	LLaMA2	7B, 13B, 34B, 70B	2023	-	✓	2T	Online sources
	Alpaca	7B	2023	LLaMA1	✓	-	GPT-3.5
	Vicuna-13B	13B	2023	LLaMA1	✓	-	GPT-3.5
	Koala	13B	2023	LLaMA	✓	-	Dialogue data
	Mistral-7B	7.3B	2023		✓	-	-
	Code Llama	34	2023	LLaMA2	✓	500B	Publicly available code
	LongLLaMA	3B, 7B	2023	OpenLLaMA	✓	1T	-
	LLaMA-Pro-8B	8.3B	2024	LLaMA2-7B	✓	80B	Code and math corpora
	TinyLlama-1.1B	1.1B	2024	LLaMA1.1B	✓	3T	SlimPajama, Starcoderdata
PaLM Family	PaLM	8B, 62B, 540B	2022	-	×	780B	Web documents, books, Wikipedia, conversations, GitHub code
	U-PaLM	8B, 62B, 540B	2022	-	×	1.3B	Web documents, books, Wikipedia, conversations, GitHub code
	PaLM-2	340B	2023	-	✓	3.6T	Web documents, books, code, mathematics, conversational data
	Med-PaLM	540B	2022	PaLM	×	780B	HealthSearchQA, MedicationQA, LiveQA
	Med-PaLM 2	-	2023	PaLM 2	×	-	MedQA, MedMCQA, HealthSearchQA, LiveQA, MedicationQA
Other Popular LLMs	FLAN	137B	2021	LaMDA-PT	✓	-	Web documents, code, dialog data, Wikipedia
	Gopher	280B	2021	-	×	300B	MassiveText
	ERNIE 4.0	10B	2023	-	×	4TB	Chinese text
	Retro	7.5B	2021	-	×	600B	MassiveText
	LaMDA	137B	2022	-	×	168B	public dialog data and web documents
	ChinChilla	70B	2022	-	×	1.4T	MassiveText
	Galactia-120B	120B	2022	-		450B	
	CodeGen	16.1B	2022	-	✓	-	THE PILE, BIGQUERY, BIGPYTHON
	BLOOM	176B	2022	-	✓	366B	ROOTS
	Zephyr	7.24B	2023	Mistral-7B	✓	800B	Synthetic data
	Grok-0	33B	2023	-	×	-	Online source
	ORCA-2	13B	2023	LLaMA2	-	2001B	-
	StartCoder	15.5B	2023	-	✓	35B	GitHub
	MPT	7B	2023	-	✓	1T	RedPajama, m Common Crawl, S2ORC, Common Crawl
	Mixtral-8x7B	46.7B	2023	-	✓	-	Instruction dataset
	Falcon 180B	180B	2023	-	✓	3.5T	RefinedWeb
	Gemini	1.8B, 3.25B	2023		✓	-	Web documents, books, and code, image data, audio data, video data
	DeepSeek-Coder	1.3B, 6.7B, 33B	2024	-	✓	2T	GitHub's Markdown and StackExchange
	DocLLM	1B,7B	2024	-	×	2T	IIT-CDIP Test Collection 1.0, DocBank

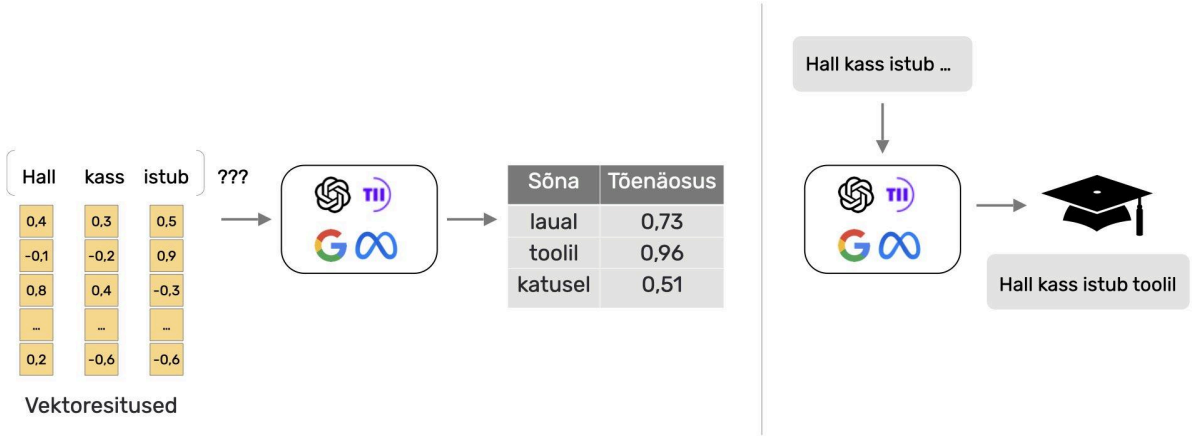
Tabel 6.3. Populaarsed keelemudelid 2024. aastal. [36]

[34] https://platform.openai.com/docs/models/gpt-3-5#gpt-3-5-turbo.
[35] https://platform.openai.com/docs/models/gpt-3-5#gpt-4-turbo-and-gpt-4.
[36] Kuvatõmmis, allikas: Minaee et al, 2024, litsents CC BY 4.0.

Toimimispõhimõtted

Suured keelemudelid õpivad statistilisi seoseid keele sõnade ja fraaside vahel. See saavutatakse mudeli treenimisega, kasutades massiivseid teksti- ja koodiandmestikke, sealhulgas raamatuid, artikleid, koodirepositooriume ja sotsiaalmeediapostitusi. Kui mudel on need seosed ära õppinud, suudab see genereerida uut teksti. Selleks alustab mudel sisendtekstiga, näiteks mõne sõna või lausega. Seejärel kasutab mudel varem õpitud statistilisi seoseid, et ennustada lauses järgmist sõna, ja jätkab selle tegemist, kuni on genereerinud kõik uued laused. Sisendteksti, juhiseid, küsimusi ja muud tekstilist konteksti edastatakse mudelile **viiba** ehk tervikliku mudeli sisendi kaudu ning see tekst mõjutab genereeritud väljundit. Seega, iga kord, kui te oma sisendteksti viibas veidi muudate, võite saada erineva vastuse. Viiba loomine (ingl k *prompt engineering*) tähendab põhimõtteliselt seda, kuidas te muudate oma juhiseid ja sisendteksti, et genereerida uus vastus.

Nagu teame, ei räägi arvutid ja mudelid inimkeelt, vaid nad mõistavad numbreid. Seega, et mõista teksti ja tabada selle semantilist tähendust, peavad sõnad olema esitatud arvuliste vektoritena. Suured keelemudelid teisendavad esmalt sõna millekski, mida nimetatakse vektoresituseks (ingl k *embedding*). Intuitiivselt võime öelda, et vektoresitused on arvulised kõrgemõõtmelised vektorid ehk sõnade arvuline esitus, mis aitavad suurtel keelemudelitel arvutada järgmise sõna statistilist tõenäosust. Võtame näidisena lause „*Hall kass istub …*“. Me tahame täita lünga ja ennustada järgmist sõna ehk mille peal kass istus. Iga sõna esitatakse kõrgemõõtmelise vektorina ja arvutatakse järgmise sõna tõenäosus. Selles näides on järgmised võimalikud sõnad *laual*, *toolil* ja *katusel*. Sõnal *toolil* on kõige suurem tõenäosus ja seega valitakse see järgmiseks sõnaks.



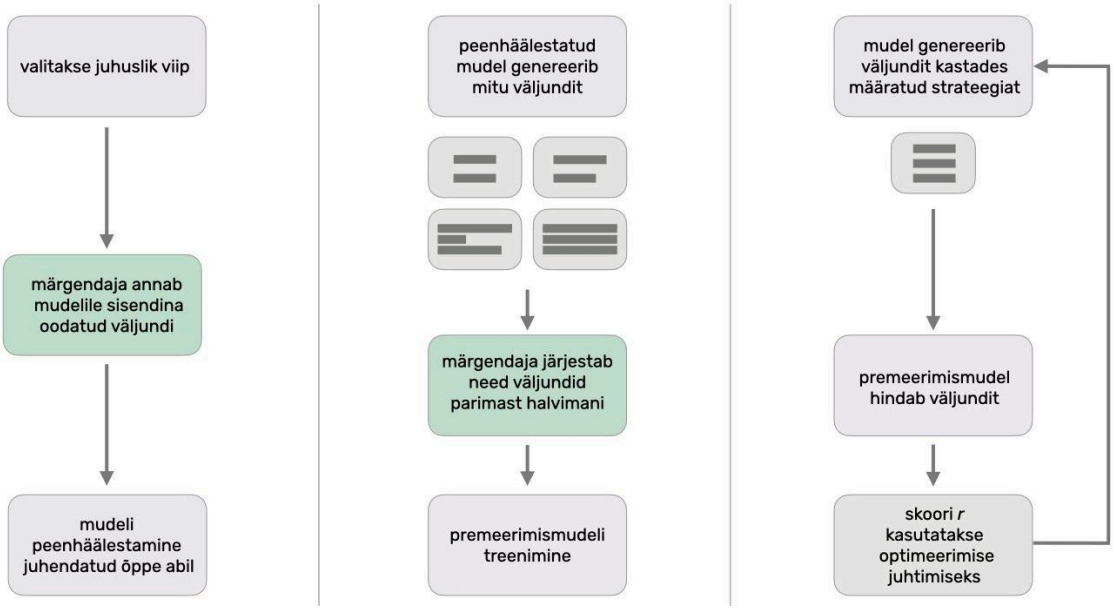
Joonis 6.28. Suurte keelemudelite tööpõhimõte. *Vasakpoolne joonise osa näitab sõnade teisendamist vektoresitusteks. Mudel kasutab vektoresitusi, et ennustada järgmist sõna, mis sobiks antud konteksti. Näiteks lauses „*Hall kass istub*…“ mudel arvutab tõenäosused erinevate võimalike järgnevate sõnade jaoks, nagu „*laual*“, „*toolil*“ või „*katusel*“. Parempoolne joonise osa demonstreerib, kuidas kasutatakse treenitud mudelit, et genereerida terviklikke tekste.*

Eelised

Suured keelemudelid ei ole võimekad ainult suuruse pärast, vaid näitavad sügavamat semantilist keele mõistmist ja paremat teksti genereerimise võimet võrreldes väiksemate keelemudelitega. Suured keelemudelid suudavad

- õppida kontekstist ehk õppida uut ülesannet lahendama vähesest hulgast viipa lisatud näidetest, mis esitatakse päringu tegemise ajal;
- lahendada uut tüüpi ülesandeid pärast juhiste kohandamist ilma uute näideteta;
- lahendada keerulisi ülesandeid, jagades need vahepealseteks arutlusetappideks, kasutades viiba mõistmiseks näiteks mõttekäigu ahela (ingl k *chain-of-thought*) tehnikat.

LLM-e saab täiendada ka väliste teadmiste ja tööriistade abil, võimaldades neil tõhusalt suhelda kasutajate ja keskkonnaga ning pidevalt ennast täiustada, kasutades suhtlusest saadud tagasisidet, näiteks stiimulõppe kaudu koos inimtagasisidega (ingl k *reinforcement learning from human feedback*, RLHF, joonis 6.29).



Joonis 6.29 Suure keelemudeli treenimine ja peenhäälestamine, kasutades inimtagasisidet. *Esiteks kogutakse näidisandmestik. Selleks valitakse viipade andmebaasist juhuslik viip. Seejärel annab märgendaja mudelile sisendi ja näitab, milline peaks olema õige vastus. Seda näidisandmestikku kasutatakse GPT-3.5 mudeli peenhäälestamiseks juhendatud õppe abil, et mudel õpiks soovitud käitumist. Teises etapis kogutakse võrdlusandmeid. Selleks valitakse viip ja genereeritakse mitu mudeli väljundit. Märgendaja järjestab need väljundid parimast halvimani. Genereeritud andmeid kasutatakse premeerimismudeli treenimiseks, mis hindab, kui hästi mudeli väljundid vastavad soovitud käitumisele. Kolmas etapp keskendub optimeerimisele.*[37]

Genereeritud väljundi hindamine

LLM-ide jõudluse ja vastuse kvaliteedi hindamiseks on mitu mõõdikut. Mudelite hindamisel tuleb arvestada ka probleeme, nagu eelarvamused, õiglus ja tõlgendatavus.

[37] *Kuvatõmmis, allikas: https://openai.com/blog/chatgpt.*

Jõudluse ja tõhususe optimeerimise tehnikad aitavad tagada, et mudelid toimivad parimal võimalikul viisil. Mudeli headuse hindamise mõõdikud võib jagada deterministlikeks ja mudelipõhisteks. Traditsiooniliselt loomuliku keele töötluse valdkonnas kasutatavad deterministlikud mõõdikud on näiteks BLEU[38], ROUGE[39] ja METEOR[40]. Mudelipõhised mõõdikud kaasavad teksti kvaliteedi hindamisel suuri keelemudeleid või muid masinõppe mudeleid. Mudelipõhised mõõdikud võimaldavad hinnata kahe teksti sarnasust, võrreldes vastavate tekstide vektoresitusi koosinussarnasuse abil. RAGAS[41] raamistik on üks headest näidetest, mis hõlmab generatiivse hindamise kasutamist.

Piirangud, probleemid ja regulatsioonid

Suured keelemudelid on väga võimekad, aga neil on ka omad piirangud, mida on vaja siin mainida. Üks tuntumaid probleeme on see, et keelemudelite teadmised on piiratud sündmustega, mis on toimunud selleks hetkeks, kui mudeli treenimine toimus. Seega ei saa mudelile esitada küsimusi lähiaja sündmuste ja värskeima informatsiooni kohta. Peenhäälestamistehnikad võimaldavad kohandada suuri keelemudeleid konkreetsete ülesannete jaoks. Peenhäälestamiseks on saadaval tööriistad ja raamatukogud, nagu HuggingFace[42], TensorFlow[43] ja PyTorch[44].

Samuti võivad mudelil puududa spetsiifilise valdkonna, näiteks meditsiini, teadmised. See tähendab, et enne mudeli kasutamist selles valdkonnas oleks vaja mudelit kohandada valdkonna teadmistega.

Kolmas probleem on seotud andmete privaatsusega. Suurte keelemudelite loojad võivad kasutada kasutaja esitatud informatsiooni treenimise eesmärgil. See ei ole aga mõeldav tundlike andmete puhul, sest võib rikkuda andmete privaatsust.

Neljas probleem on hallutsinatsioonid – mudel võib väga enesekindlalt esitada valesid vastuseid.

Nende probleemide lahendamiseks on olemas kaks lähenemist: mudeli peenhäälestamine, mis võib aga olla väga kallis, ja mudeli vastuse genereerimine, kasutades ainult valitud allikatest pärinevat informatsiooni. Sellised süsteemid sisuliselt täiendavad mudelit otsingufaasiga ja neid nimetatakse allikapõhisteks teksti loomise süsteemideks (ingl k *retrieval-augmented generation*, RAG[45]).

Arvestades neid probleeme, on oht, et keegi kasutab tehisintellekti valesti ja tekitab kahju. On vajadus selgete õiguslike raamistike järele, mis tagavad ohutuse, selgitatavuse ja vastutuse AI-süsteemide kasutamisel. Üks olulisemaid regulatsioone on Euroopa Liidu

[38] https://aclanthology.org/P02-1040.pdf.
[39] https://aclanthology.org/W04-1013.pdf.
[40] https://aclanthology.org/W05-0909.pdf.
[41] https://docs.ragas.io/en/stable/.
[42] https://huggingface.co/.
[43] https://www.tensorflow.org/.
[44] https://pytorch.org/.
[45] https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

tehisintellekti määrus, tuntud kui **AI Act**[46,47], mis võeti vastu 2024. aastal. AI Act seab nõuded tehisintellekti kasutusele ja jagab AI-süsteemid nelja riskikategooriasse: minimaalne, piiratud, kõrge ja vastuvõetamatu risk. Kõrge riskiga AI-süsteemid, näiteks need, mida kasutatakse tervishoius, hariduses ja õiguskaitses, peavad vastama rangetele selgitatavuse ehk läbipaistvuse, turvalisuse ja andmekaitse nõuetele.

AI Acti eesmärk on hoida ära tehnoloogiline diskrimineerimine ja muude riskide ilmnemine, tagades, et innovatsioon toimub eetilistes ja ühiskonnale ohututes piirides. Peale AI Acti on teisigi üle maailma tunnustatud standardeid, nagu OECD tehisintellekti põhimõtted ja UNESCO AI eetika soovitused, kuid need on ainult soovitused, mitte seadused.

Regulatsioonide tundmine ja järgimine on oluline nii arendajatele kui ka rakendajatele, sest need piiravad, millist tüüpi lahendused on mingis valdkonnas lubatud või vajavad lisajärelevalvet. Trahvid võivad kõrge riskiga valdkondades ulatuda 30 miljoni euroni või 6%-ni ettevõtte aastasest üleilmsest käibest, olenevalt sellest, kumb summa on suurem. Lisaks võib EL rikkumiste jätkumisel keelata ettevõtte tegevuse EL-i majandusruumis.

Enesekontrolli küsimused

1) Mis on tehisnärvivõrgu põhiline tööüksus ning kuidas see toimib?

2) Miks nimetatakse teatud tüüpi närvivõrke sügavateks närvivõrkudeks?

3) Kuidas toimub õppimine tehisnärvivõrgus ning milline on kõige levinum optimeerimisviis?

4) Mis on pildi RGB väärtused? Kuidas moodustub RGB väärtustest pilt?

5) Miks on pildid keeruline andmetüüp. Milline tehnika, lähenemine on leiutatud, et pilte ikkagi efektiivselt töödelda?

6) Miks on keel tehisintellektile keeruline, võrreldes näiteks tabeliandmetega?

7) Mis on alusmudel ja kuidas see erineb tavalisest sügavõppe mudelist?

8) Mida tähendavad nullsammuga ja mõne sammuga üldistumine ja miks need alusmudelite puhul olulised?

[46] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689&qid=1734269475974.
[47] https://artificialintelligenceact.eu/.

7. Tulemuste hindamine

Kõige tavalisemas koosolekuruumis kohtuvad andmeteadusmeeskond ja ettevõtte tegevdirektor. Andmeteadusmeeskond on loonud süsteemi, mis suudab üsna täpselt ennustada, kas kasutaja tagastab laenu või mitte. Nende lahendus paistab silma igast küljest: õigsus, saagis, täpsus, ROC-kõver – esitlus on täis arve, mis on 0,90-st ülespoole. Ühesõnaga, kõik viitab sellele, et mudel tõesti suudab üsna hästi modelleerida andmestikus sisalduvat varieeruvust. Ent vaatamata graafikutele on tegevdirektoril ikkagi küsimus: „Aga mida need arvud tähendavad ärilises mõttes?“ Mida ta üldse selle all mõtleb?

Tegevdirektoril on peas mitmesuguseid muresid (andmeteaduse meeskonna kaitseks: selline oli ta juba uksest sisse astudes). Kas mudel suudab võrdväärselt hästi ennustada kõikides laenusegmentides, kaasa arvatud kõige suuremad ja rahalises mõttes riskantsemad laenud? Kui kõige paremaks mudeliks osutus musta kasti tüüpi mudel, kas juriidilised reeglid üldse lubavad sellist mudelit kasutusse võtta? Ja kui lubavad, siis kui palju raha me kaotaksime, kui võtaksime kõige parema ja keerulisema mudeli asemel kasutusse natuke halvema, aga lihtsamini mõistetava mudeli? Ega mudeli ennustused diskrimineeri teatud kasutajate segmenti? Kas praegune protsess, kus kõige suuremaid laene kontrollib inimene, tasub asendada selle uue suurepärase mudeliga või on see rahaliselt otstarbetu?

Selles peatükis räägime, kuidas saadud lahendust analüüsida, et pärast analüüsi oleks tegevdirektoril vähem selliseid muresid.

7.1 Baasmudel

Baasmudel (ingl k *baseline*) on heuristiline lahendus või masinõpe mudel, millega võrreldakse uut lahendust. Baasmudel ei ole alati lineaarne regressioon (või muu tehniliselt lihtne lahendus). Näiteks, teadusartiklides on baasmudeliks tipptasemel (ingl k *state of the art,* SOTA) lahendus probleemile, millele üritatakse pakkuda veelgi parem lahendus. Baasmudelil on mitu eesmärki:

- hinnata probleemi keerukust ja madalaimat ennustustäpsuse taset, millest alates keerulisemal mudelil on üldse mõtet;
- suunata keerulisema mudeli disaini sinna, kus baasmudel hakkama ei saa;
- hinnata andmekvaliteeti. Lihtne interpreteeritav baasmudel võib paljastada andmetega seotud probleeme;
- kiire prototüüpimine.

Baasmudelid jagunevad kolme tüüpi: heuristiline lahendus, masinõpe mudel ja inimtase (ingl k *human performance baseline*).

7.1.1 Heuristiline lahendus

Enne masinõpe lahenduse loomist tasub endalt küsida, kas probleemi saaks lahendada ka ilma masinõppeta. Sageli on ka see võimalik ja selline lahendus peitub teatud

kuldreeglite ehk heuristikate kasutamises. Harva aga juhtub ka, et heuristiline lahendus osutub masinõpe lahendusest hoopiski kasulikumaks, ja siis on hea, et sellise lahenduse sobivust kontrolliti, enne kui asuti ehitama keerulist mudelit.

Võtame näiteks kaupluse, mille juhil on vaja iga nädal otsustada, milliseid tooteid juurde tellida. Heuristiliselt lähenedes võiks juht kehtestada mingi lihtsa reegli, põhinedes ajaloolistel müügiandmetel. Näiteks, ta võib otsustada tellida kõiki tooteid, mille laovaru väheneb alla teatud läve, näiteks 120% keskmisest nädala müügimahust. Masinõppe mudeli loomine võib siin olla keeruline, sest mõne väiksemamahulise toote puhul ei pruugi olla piisavalt treeningandmeid.

7.1.2 Masinõppe baasmudel

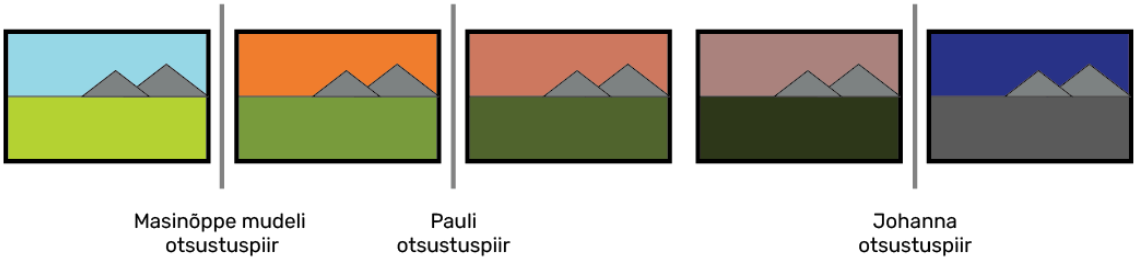
Masinõppe populaarseimad baasmudelid on lineaarne ja logistiline regressioon. Aga võib ka luua ennustused ilma mudelit kasutamata.

- Enamusklassi ennustamine ehk ennustada alati seda klassi, mida andmestikus on kõige rohkem. See on eriti kasulik juhul, kui andmestiku klassijaotus on väga kallutatud.
- Juhuslik ennustus (ingl k *random baseline*). Kasutatakse siis, kui on kahtlus, kas andmetes üldse sisaldub piisavas koguses kasulikku infot, et saaks midagigi ennustada. Hilisema lahenduse juhusliku baasmudeliga võrdlemisel öeldakse tavaliselt, et mudel oli nii palju protsente parem kui juhuslik ennustus.

7.1.3 Inimtase

Kui eelmised kaks tüüpi baasmudeleid luuakse selleks, et hinnata miinimumtaset, mille lahendus peaks kindlasti saavutama, siis inimtase näitab, vastupidi, saavutatavat maksimumi. Seda kasutatakse tavaliselt struktureerimata andmetel lahendatavate probleemide korral nagu masinnägemine, masinkuulmine või loomuliku teksti töötlus, et aru saada, kas on veel ruumi tulemuse parandamiseks. Kui inimene ei suuda aru saada tema emakeeles öeldud fraasist, siis on tavaliselt vastuvõetav, kui ka algoritm seda ei suuda.

Inimtaseme hindamisel on oluline meeles pidada inimeste erimeelsusi. Võtame näiteks ülesande, kus kaamerarežiimi automaatseks reguleerimiseks tahetakse klassifitseerida öösel ja päeval tehtud pilte. Kas öö algab päikeseloojangu hetkega või siis, kui taevas on täielikult tume ja sellele ilmuvad tähed (joonis 7.1)? Kui andmetes on see piir selgelt tõmmatud, aga inimtaseme hindamisel on jäetud inimeste äranägemisele, siis võib ekslikult tunduda, et mudel on inimtasemest tugevam.



Joonis 7.1. Inimeste otsustuspiirid erinevad.

Lõksud
Andmeteadlasel on kiusatus mitte peatuda piisavalt baasmudeli loomisel, kuna tundub, et see jääb niikuinii kasutamata ja selle ainuke otstarve on näidata keerulisema mudeli võimekust. Tegelikult on mõttekas kuulutada aega baasmudeli hüperparameetrite otsinguks ja regulariseerimiseks, et luua kõige tugevam võimalik baasmudel. Siis on keerulisel mudelil, millega võistelda, ja saab puhta südamega vastata, et keerulisema mudeli juurutamine on otstarbekam, sest see on tõesti seda väärt.

7.2 Vigade analüüs

Peatükis 5 (mudeldamine) rääkisime sellest, kuidas hinnata mudeli täpsust. Optimeerisime mudeli nii, et võimalikult väike protsent andmeridu oleks ennustatud valesti, aga isegi väga heal mudelil see väike protsent jääb ning nende vigade iseloom võib olla erinevatel andmete alamgruppidel erinev, sõltuda andmetes sisalduvast mürast, mõjutada mudeli õiglust ja ohutust.

7.2.1 Tabeliandmete vigade analüüs

Tuleme tagasi näite juurde, mida kasutasime peatüki alguses: laenu tagastamise tõenäosuse ennustamine. Mudeli treeningandmetes on sellised tunnused nagu võla suurus, panga ajalugu, laenaja vanus, sugu, töökoht jne. Meie ülesanne on leida, kas mõnes segmendis on mudeli täpsus oluliselt väiksem. Näiteks, kontrollime, kas mudel on sama täpne nii väikelaenude kui ka suurte laenude korral. Tabelist 7.1 näeme, et mudeli täpsus kahaneb laenusumma suurenedes. Laenude puhul, mille suurus on üle miljoni euro, on täpsus ainult 0,56, mis tähendab, et mudel tagastab 44% valepositiivseid tulemusi ehk 44% laenudest, mille mudel heaks kiitis, ei tagastata.

Joonis 7.1. Mudeli täpsuse võrdlemine laenusumma kategooriate lõikes. *Kõrgeim täpsus on alla 5000 eurostel laenudel.*

Kui on selge, millistes kategooriates mudel kõige rohkem vigu teeb, võib pöörduda mudeli seletatavuse meetodite juurde uurimaks, miks mudel neid vigu tegi. Näiteks, võib selguda, et mudel otsustas suure laenu väljastada sellepärast, et laenutaotleja palk oli suur. Väikese laenu puhul see on garantiiks, et inimene tagastab laenu, suure laenu puhul mitte nii väga. Kui suurte laenude osakaal on väike, võib aidata andmete tasakaalustamine selle kategooria võrra, ja treeningandmestiku loomisel andmete stratifitseerimine ehk klasside võrra tasakaalustatud treeningandmestiku loomine.

7.2.2 Multimeediaandmete vigade analüüs

Kui tegu on mudeliga, mille sisendiks on pildid, kõne või videod, ole sellistel andmetel sageli olemas piisavalt detailset taustainfo märgistust, mida võiks vigade analüüsis näidete gruppide loomisel kasutada. Mõnikord aitavad selgust luua metaandmed. Näiteks, kui hindame kõnetuvastussüsteemi kvaliteeti, siis võime arvutada õigsust kõneleja järgi (kes räägib), asukoha järgi (kus heli salvestati), taustamüra taseme ja tüübi järgi. Siis võib selguda, et kõnetuvastus ei toimi hästi kooliealiste laste kõnel või restorani müraga. Kui sobivaid metaandmeid ehk taustainfot iga andmepunkti kohta ei ole, tuleb neid luua käsitsi, kuulates ja märgistades neid andmepunkte, kus mudel on teinud vigu.

7.3 Mudeli seletatavus

On mitu põhjust, miks võib olla oluline mõista, kuidas masinõppe mudel otsuseid teeb.

- **Usaldusväärsus** ja **läbipaistvus** on eriti olulised tundlikes valdkondades, nagu tervishoid, õigusteadus, pangandus. Näiteks, patsiendi läbivaatusel üldise vereanalüüsi ja teiste andmete põhjal kahtlustas mudel süvaveenitromboosi ja soovitas lisauuringuid. Arstile on oluline teada, millised näitajad vereanalüüsis on põhjustanud sellise kahtluse, et ka ise lisauuringu vajaduse üle otsustada. Patsiendile võib olla oluline teada, et üks olulistest tunnustest, mida mudel kasutas, oli suitsetamine.
- **Õiglus** (ingl k *fairness*): mudelid võivad õppida eelarvamusi nendest andmetest, mille peal neid treenitakse. Näiteks, 2014. aasta uuringus leiti, et mudel, mida kasutati kurjategija riskihindamiseks, oli kallutatud rassi suhtes ja hindas mustanahalisi inimesi peaaegu kaks korda tõenäolisemalt tulevasteks kurjategijateks, ka arvestades teisi tegureid nagu kriminaalajalugu, vanus ja sugu (Hong, 2019).
- **Seaduslikkus**: tehisintellekti süsteemidele kehtivad omad reeglid ja eeskirjad, mille järgi peab olema tagatud nende ohutus, inimeste privaatsuse säilitamine ja eetiline käitumine ning teatud juhtudel ka läbipaistvus. Üks selliseid seadusi on 13. märtsil 2024 Euroopa Liidus vastu võetud tehisintellekti määrus (ingl k *AI Act*).
- **Mudeli täpsuse parandamine**: seletatavuse tehnikad võivad aidata täpselt kindlaks teha, kus on mudeli või andmestiku kitsaskohad, ja aidata neid parandada.

Masinõppe mudelid jaotuvad kahte tüüpi: **oma olemuselt seletatavad** mudelid (ingl k *inherently explainable*) ja **musta kasti** tüüpi mudelid (ingl k *black box models*). Peatükis 5 on tutvustatud enamasti esimesse kategooriasse kuuluvad mudelid, välja arvatud tugivektormasinad. Peatükis 6 tutvusite klassikaliste musta kasti mudelitega –

tehisnärvivõrkudega. Mõlemat tüüpi mudelite jaoks on välja töötatud tehnikad, mis aitavad aru saada, mis teadmine on väljendatud mudeli parameetrides. Enamik tehnikaid on loodud juhendatud õppe mudelite jaoks.

7.3.1 Seletatavad mudelid

Lineaarne ja logistiline regressioon on klassikalised algoritmid, mis on tõhusad ja lihtsad, ei nõua väga palju andmeid ning mille koefitsiendid on lihtsasti seletatavad. Nendel mudelitel on aga tõsine puudujääk: neid saab rakendada ainult lihtsamatele probleemidele, kus andmetes väljendatud seosed on lineaarsed ja mitte väga keerulised.

Tuleme tagasi näite juurde peatükis 5.3.1, kus reklaamikulu põhjal ennustatakse müügitulu. Tabelis 7.1 on näidatud regressioonimudeli koefitsiendid ja hinnangu standardvead (ingl k *standard error of the estimate*).

Tunnus	Tüüp	Koefitsient	Hinnangu standardviga
Vabaliige	–	6,89	2,3
Televisiooni reklaamikulu	Arvuline tunnus	1,5	0,3
Sotsiaalmeedia reklaamikulu	Arvuline tunnus	2	0,75
On madalhooaeg	Kategooriline tunnus	–3,5	2,1

Tabel 7.1. Regressioonimudeli koefitsiendid ja hinnangu standardvead.

Meenutame, et sõltuv muutuja oli väljendatud tuhandetes eurodes. Tabelist näeme, et televisiooni reklaamikulu suurendamine 1000 euro võrra suurendaks müügitulu 1500 euro võrra, kõiki teisi muutujaid samaks jättes. Sotsiaalmeedial on rohkem mõju: müügitulu suureneb kaks korda rohkem võrreldes sotsiaalmeedia reklaamikuluga. Kolmas tunnus on kategooriline ja selle koefitsient ütleb meile, et kui on madalhooaeg, siis müügitulu väheneb 3500 euro võrra.

Kui tegu on logistilise regressiooniga, siis seos koefitsiendi ja ennustuse vahel ei ole enam lineaarne, kuna me kasutasime ennustuse tõenäosuse arvutamiseks eksponentfunktsiooni. Selleks, et seosed oleksid lineaarselt interpreteeritavad, peame rakendama eksponentfunktsiooni ka koefitsientidele. (Molnar, 2022)

Otsustuspuud lubavad võtta kahest maailmast parima – nad lubavad seletatavust ja oskavad hakkama saada ka situatsioonis, kus seosed on mittelineaarsed. Väiksed otsustuspuud ei nõua spetsiaalseid seletamise tehnikaid, vaid on otseselt mõistetavad kui reeglid. Suuremad puud ja metsad, kus sama tunnus võib esineda tuhandetes hargnemispunktides, arvutavad iga **tunnuse tähtsuse** (ingl k *feature importance*), läbides kõik otsustussõlmed, kus seda kasutatakse, ja mõõtes, kui palju tunnus on võrreldes eelmise sõlmega vähendanud kas dispersiooni või Gini indeksit.

7.3.2 Globaalsed ja lokaalsed meetodid

Kui mudel on musta kasti tüüpi, tuleb kasutada teistsuguseid seletatavuse tehnikaid. Need jagunevad kahte tüüpi: globaalsed mudeliagnostilised ja lokaalsed mudeliagnostilised tehnikad. Mudeliagnostilisus tähendab, et tehnika sobib ükskõik milliste mudelite, sealhulgas närvivõrkude, otsustusmetsade, tugivektormasinate jne, otsuste seletamiseks. Globaalsed meetodid püüavad kirjeldada mudeli ennustusi tervel andmestikul keskmiselt. Lokaalsed meetodid, millest tuntuim on **LIME** (ingl k *local interpretable model agnostic explanations*) meetod, seletavad üksikuid ennustusi ehk mudeli loogikat ühe andmepunkti korral.

Erinevate meetoditega tutvumiseks soovime kasutada eespool viidatud raamatut „Interpretable Machine Learning".

7.3.3 Isejuhendatud mudelite seletamine

Isejuhendatud õpe on üks populaarsemaid tänapäeva tehnikaid, mis võimaldab mudelite treenimist hiigelsuurtel märgendamata andmetel. Seda kasutatakse suurte keelemudelite loomisel (GPT, Gemini, Claude mudelid jne), masinnägemise ja masinkuulamise ülesannetel. Peatükis 7.3.2 kirjeldatud tehnikad selliste mudelite seletamiseks ei sobi.

Üks viis, mida kasutatakse, et aru saada, mida on selline mudel õppinud, on hinnata mudeli vektoresitustes (ingl k *embedding*) sisalduvat informatsiooni. Selleks kasutatakse siirdeõpet mõnele juhendatud õppe ülesandele (ingl k *probing*)[48]. Näiteks, kui keelemudel õppis isejuhendatud õppe käigus muu hulgas sõnaliike eristama, õpib sellise mudeli vektoresituste peal treenitud klassifikaatormudel väikese vaevaga sõnaliike ennustama, sest see teadmine oli mudelis juba mingil kujul olemas. Kui sõnaliikide ennustamine hea täpsusega ei õnnestu, tähendab see, et sellist informatsiooni lihtsasti kättesaadaval kujul nendes vektoresitustes ei sisaldu ja järelikult mudel seda oma otsuste tegemise loogikas ei kasuta.

7.4 Mudeli headuse mõõdikute kasutamine

Peatükis 5 rääkisime mudelite headuse hindamisest ja mõõdikutest, mida selleks kasutatakse. Õppisite, mis on õigsus, saagis, täpsus, keskmine ruutviga. Millal sobib kasutada neid mõõdikuid?

Vaatame ennustusmudelit, mis suudab ~98% täpsusega ennustada, kas inimene on punapea või mitte. Selle tööpõhimõte on väga lihtne. Kui inimene elab Punapea jõe kaldal (Saaremaal Metskülas), siis pakume, et ta on punapea. Kui ta elab ükskõik kus mujal Eestis, näiteks Mustvees või Valgemetsas, siis meie mudel on kindel, et selline inimene ei saa olla punapea. Kogume meie andmed tabelisse. Lihtsuse mõttes teeme, nagu Eestis elaks täpselt 1 360 000 inimest.

[48] Lisainfot sel teemal võib leida Stanfordi ülikooli lehelt: link.

	On punapea	Ei ole punapea	Kokku
Metskülast pärit	2	105	107
Mujalt Eestist pärit	27 198	1 332 695	1 359 893
Kokku	27 200	1 332 800	

Tabel 7.2. Eesti elanikke jaotus kahe tunnuse järgi (kas elab Metskülas, kas on punapea).

Arvutame meie mudeli õigsuse. Kaks inimest, kes elavad Punapea jõe kaldal ja on punapead, on meie tõsipositiivsed, 1 332 695 inimest, kes seal ei ela ega ole ka punapead, on meie tõsinegatiivsed väärtused. Veel on meil 105 valepositiivset ja 27 200 valenegatiivset väärtust. Mudeli õigsus on seega ~98%[49].

Ilmselgelt võib sellel mudelil selline õigsus olla ainult selle pärast, et positiivse klassi väärtuseid on meie andmestikus vähe. Maailmas ongi punapäid vähe. Selliseid probleeme, kus üks klass on alaesindatud, on reaalelus väga palju. Enamik kõnesid, mis tehakse mobiilivõrgus, ei ole petukõned. Enamik pangaülekandeid ei soorita rahapesu. Enamikul inimestel ei ole rinnavähki. Selliste ülesannete jaoks, kus andmestiku jaotus klasside vahel ei ole ühtlane, ongi vaja kasutada **täpsust**, **saagist** ja **ROC-kõveraid**. Arvutame meie punapeade tuvastamismudeli täpsuse ja saagise.

$$\text{Täpsus} = TP / (TP + VP) = 2 / (2 + 105) = 0{,}019 \text{ ehk } 1{,}9\%.$$

Täpsus näitab, milline protsent mudeli tagastatud positiivsetest ennustustest on tõesti positiivsed näited.

$$\text{Saagis} = TP / (TP + VN) = 2 / (2 + 27\,198) = 0{,}00007 \text{ ehk } 0{,}007\%.$$

Saagis näitab, kui paljud positiivse klassi näidetest me oleme üles leidnud. Milline täpsus ja saagis on kasutaja jaoks vastuvõetav, sõltub probleemist.

Ülesanne iseseisvaks mõtlemiseks
Mõelge, kas nende probleemide korral on olulisem täpsus või saagis:

1) mudel tuvastab lahkuvaid kliente, et neile saaks helistada ja neid ümber veenda;
2) sõjaväe radar tuvastab vaenlase droone, et sihtida nende pihta rakette;
3) meditsiiniline test tuvastab, kas inimesel on vähk;
4) pangaülekannetel tuvastatakse pettust.

Ühed võimalikud vastused leiate siit: link.

[49] Õigsus = TP + TN / (TP + TN + VP + VN). Et meil on väga suur hulk tõsinegatiivseid ehk mitte-punapäid mujal Eestis, on õigsus suur.

7.5 Tervikliku andmeteaduslahenduse hindamine

Tulemuse hindamise etapis olete läbinud kõik CRISP-DM projekti etapid. Nüüd on aeg tulla tagasi äriliste kriteeriumite ja nõuete juurde, mis olid seatud esimeses etapis, ja vastata järgmistele küsimustele.

1. Kuidas mõjutab projektis saavutatud tulemus põhilisi äri võtmenäitajaid? Näiteks, kui projekti käigus oli loodud töölaud finantsosakonnale, kas see parandab osakonna tõhusust? Kui projekti käigus oli loodud soovitussüsteem, kas selle soovitused suurendavad ristmüüki?
2. Mis on juurutatud lahenduse jooksev kulu? Siin on vajalik tasuvusanalüüs, mis arvestab sellega, kui palju rahalist kasu lahendus toob.
3. Mis on ebaõnnestunud ja miks ning kuidas saaks järgmisel korral neid vigu vältida?
4. Mis on hästi õnnestunud ja kas saaks seda rakendada ka mujal?

7.5.1 Investeeringutasuvus

Praktikas andmeteaduse ja tehisintellekti projektide tulemuste hindamine on puudulik, kui ei hinnata investeeringutasuvust ehk investeeringute tootlust (ingl k *return of investments,* ROI). ROI mõõdab projekti investeeritud ressurssidest saadavat kasu. Mida suurem on investeeringu tasuvus, seda suurem on saadud tulu võrreldes tehtud kulutustega. ROI abil saab hinnata, kui tulemuslik on investeering olnud, ning tagada ressursside tõhusa kasutamise ja väärtuse loomise. Tehisintellekti ja andmeteaduse projektide kontekstis kajastab ROI rahalist tulu, efektiivsuse paranemist ja strateegilist väärtust, mida lahendused pakuvad. ROI-d võib arvutada, kasutades üldist valemit, mida saab kohandada iga ROI kategooria jaoks:

ROI = (investeeringu tulu – investeeringu kulu) / investeeringu kulu × 100%.

Miks arvutada ROI-d? ROI aitab

- **põhjendada investeeringuid** – ROI näitab tehisintellekti projektide väärtust äriga tegelevatele osapooltele ja aitab tagada tulevast rahastust;
- **kooskõlastada projekti mõju ärieesmärkidega** – ROI näitab, kas projekt on kooskõlas protsesside tõhususe, tulu kasvu, kulude optimeerimise ja tööjõu koormuse leevendamise eesmärkidega;
- **võrrelda projektide tulemusi,** et tuvastada suurima mõjuga lahendused.

Kõige paremad ROI-d andmeteaduse ja tehisintellekti projektide jaoks võib genereeritud väärtuse alusel jaotada järgmisteks gruppideks:

- **äritegevuste tõhusus** – kui palju lahendus aitab vähendada vigu ja parandada kvaliteeti tänu tõhustatud äriprotsessidele.
- **tulu kasv ja kulude optimeerimine** – kui palju ressursse lahendus aitab säästa (vähendada raiskamist) või kui palju kasvab müügitulu tänu lahenduse kasutamisele.

- **tööjõu koormuse leevendamine** – kui palju aega lahendus aitab säästa tänu korduvate ülesannete automatiseerimisele.
- **efektiivsus ja kiirus** – kui palju lahendus kiirendab otsuste tegemist ja ülesannete täitmist.

Oluline tegur ROI analüüsis on ka lahenduse kulu. Tüüpiline andmeteaduse või tehisintellekti projekti kulu koosneb järgmistest kuludest.

- **Taristukulu**
  - Pilvelahendused (nt AWS, Azure, Google Cloud) hõlmavad tavaliselt tasu arvutusvõimsuse, teenuste (mudelite API-d, konteinerite äpid, virtuaalsed masinad jpm), salvestusruumi ja andmeedastuse eest. Pilvepõhine AI-lahendus võib maksta 1000 eurot kuus.
  - Lokaalsed lahendused nõuavad investeeringuid serveritesse, võrgustikku ja hooldusesse. Lokaalse taristu loomise algmaksumus võib olla 50 000 eurot, millele lisandub 5000-eurone aastane hoolduskulu.
- **Arenduskulu**
  - Arendajate palk, töövahendid jpm. Näiteks kolmest insenerist koosnev meeskond kuue kuu jooksul palgaga 3000 eurot kuus maksab kokku 54 000 eurot.
- **Jooksev äritegevuse kulu**
  - Pidev tugi, monitoorimine ja uuendused.

Lihtsustatud näide ROI arvutamisest andmeteaduse projektis #1
Pood kaotab kliente pärast esimest ostu. Nende klientide hoidmine on tähtis, kuna uute klientide hankimine on kallis. Andmeteaduse meeskond pakub välja masinõppe mudeli, mis tuvastab riskigrupis olevad kliendid, ja pood saadab selle alusel neile isikustatud pakkumisi, et motiveerida neid korduvoste tegema. Ettevõte soovib teada saada, kas tema jaoks on kasumlik masinõppe mudeli arendusesse investeerida. Selle jaoks arvutatakse ROI-d. Kui ROI on positiivne, soovib ettevõte mudelit arendada ja selle kasutusele võtta.

Esimese sammuna arvutatakse kogu investeeringu kulu, mis selles projektis koosneb taristu- ja arenduskulust.

- Taristukulu: andmete töötlemise tööriistad ja pilvearvutus mudeli arendamiseks ja kasutuselevõtuks maksavad 5000 eurot.
- Arenduskulu: projekt nõuab kahe andmeteadlase tööd kahe kuu jooksul, kummagi kulu on 10 000 eurot, 2 × 10 000 eurot = 20 000 eurot.
- **Kogu investeeringu kulu** = 20 000 eurot + 5000 eurot = 25 000 eurot.

Teise etapina võib arvutada potentsiaalse tulu.

- Oletame, et aastane tulu ühe kliendi kohta on 100 eurot. Praegune kliendihoidmise määr on 50% (1000 kliendist). Eeldame, et mudel

suurendab kliendihoidmise määra 50%-lt 60%-le. See tähendab, et püsima jääb 100 lisaklienti, seega lisandunud tulu hoitud klientidelt on 100 × 100 eurot = 10 000 eurot.
- Oletame, et see tulu püsib igal aastal ja mudel toimib edukalt kolm aastat. **Kogu investeeringu tulu** = 10 000 eurot × 3 = 30 000 eurot.

Kolmanda sammuna saab arvutada ROI.
- **ROI** = (30 000 – 25 000) / 25 000 × 100 = 20%.

Neljas etapp on tõlgendamine.
- 20% ROI tähendab, et iga investeeritud 1 euro kohta teenib ettevõte kolme aastaga tagasi 1,20 eurot. See näitab, et projekt on kasumlik, kuna see loob rohkem väärtust, kui kulutab investeeringuteks.

Keskendudes konkreetsetele tulemustele, nagu näiteks hoitud klientide tekitatud tulu, seob ROI mudeli tehnilise edu ettevõtte ärilise eesmärgiga ja muudab projekti väärtuse hindamise lihtsamaks.

Lihtsustatud näide ROI arvutamisest andmeteaduse projektis #2
ROI-d saab hinnata mitte ainult rahalise väärtuse, vaid ka säästetud aja ja teiste oluliste mõõdikute kaudu, olenevalt projekti olemusest. Masinõppe mudelite kasutamine automatiseerimislahendustes on suurepärane näide, kuidas investeeringut õigustada ja selle väärtust mõõta.

Näiteks, võtame olukorra, kus käsitsi andmete sisestamine võtab 1000 tundi kuus, andmesisestaja tasu on 20 eurot tunnis, mis teeb kokku 20 000 eurot kuus. Kui sama töö masinõppe mudeli abil automatiseeritakse, vähendatakse töökoormust nii, et andmesisestus vajab edaspidi ainult 100 tundi kuus. Sellisel juhul saavutatakse märkimisväärne aja ja raha kokkuhoid.

Masinõppe mudel töötab selleks, et analüüsida ja klassifitseerida andmeid automaatselt, kasutades enne treenitud algoritmi. Näiteks võib mudel andmeid valideerida ja korrastada ilma inimese sekkumiseta, jättes töötajale üksnes keerulisemad ülesanded. See võimaldab töötajatel keskenduda väärtuslikumatele ja loovamatele ülesannetele.

Säästetud aja ja tööjõukulu arvutused näitavad järgmisi tulemusi:
- säästud aeg = (1000 – 100) / 1000 × 100 = 90%;
- tööjõukulu vähenemine = (20 000 – 100 × 20) / 20 000 × 100 = 90%.

See tähendab, et masinõppe mudeli rakendamine vähendab nii tööks kuluvat aega kui ka tööjõukulu **90% võrra**, mis demonstreerib investeeringu tugevat tasuvust ja lahenduse lisandväärtust ettevõttele. Peale selle parandab mudel andmete töötlemise täpsust ja kiirust, aidates teha äriprotsesse tõhusamaks.

Enesekontrolli küsimused

1) Vaatame järgmist päriselu situatsiooni. Andmeteadusmeeskond arendas lahkuva kliendi ennustamise süsteemi. Iga kord, kui klient helistas kõnekeskusesse, näidati kõnekeskuse operaatorile, kas see klient tõenäoliselt lahkub lähima kolme kuu jooksul või mitte. Süsteem on sellisel kujul töötanud pool aastat, aga klientide lahkumine ei ole vähenenud. Miks nii?

2) Ettevõtte ABC Music müüb muusikat reklaami jaoks. Selleks oleks mugav, kui kasutaja saaks otsida muusikat emotsiooni järgi – kas muusika on kurb või rõõmus. ABC Musicu andmeteadlane on loonud sellise mudeli, aga tal ei õnnestu kuidagi tõsta süsteemi õigsust üle 70%. Milline oluline etapp võiks siin olla abiks, et välja selgitada, mis toimub?

3) Teie kontoris otsustati kasutada ukse avamiseks ise arendatud näotuvastussüsteemi. Paraku andis süsteem üsna tihti valenegatiivseid tulemusi (ei lasknud töötajaid sisse ja töötajad pidid kasutama võtit) ning ükskord lasi kontorisse tuvi, kes tänaval mööda lendas. Millise veaanalüüsi te teeksite?

8. Juurutamine

Olete järginud kõiki masinõppe häid tavasid ning nüüd on teie käes testitud ja töötav masinõpe mudel, mis on valmis tooma praktilist kasu teie tegevusvaldkonnas. Kas olete jõudnud teekonna lõppu või hoopis asute teekonna alguses? Sageli arvatakse ekslikult, et mudeli juurutamine on puhtalt inseneeriaülesanne, milles olemasolev mudel sobitakse kokku ülejäänud IT-süsteemiga. Tegelikult on juurutamisprotsessis omajagu ka andmeteaduslikke probleeme, nagu **andmete kõrvalekalle** (ingl k *data drift*), **masinõppesüsteemi käitumise jälgimine** (ingl k *monitoring*) ja **andmete õiglus** (ingl k *data fairness*). Juurutamist käsitleb selline teadmusala nagu **MLOps** (ingl k *machine learning operations*).

Masinõppe lahenduste kasutuselevõtt erineb oluliselt tavalise tarkvaratoote omast. Tavaline tarkvara käitub deterministlikult ehk ettemääratult. Igal käivitamisel me teame täpselt, mis juhtub. Mudel aga tagastab ennustusi vastavalt oma sisendile ja juurutatud süsteemis sisend varieerub.

8.1 Masinõppe töövoog

Selleks, et tarkvarasüsteemis toimida, vajab masinõppe mudel enda ümber suurt kogust koodi, mis korjab kokku, töötleb ja puhastab andmeid, liidab andmeallikaid ja teeb tunnuste eeltöötluse, võtab vastu päringuid, tagastab vastuseid, jälgib süsteemi toimimist. Selgub, et mudeldamine ja selle jaoks loodud kood on tegelikult väga väike osa kogu süsteemist. Kogu töövoog (ingl k *pipeline*) on tavaliselt korraldatud suunatud atsüklilise graafina. Graafis sisalduvate funktsioonide jooksutamiseks, paralleliseerimiseks, logimiseks ja jälgimiseks kasutatakse tarkvara, mis on välja töötatud töövoogude korraldamiseks. Tabelis 8.1 on esitatud selle valdkonna populaarsemad teegid.

Tarkvara	Esimene avatud lähtekoodiga versioon	Populaarsus (GitHub repo tärnides)	Keerukus	Programmeerimiskeel
Apache Airflow	2014	34 500 ☆	Väga keeruline	Python
Luigi	2012	17 300 ☆	Väga lihtne	Python
MLFlow	2018	17 300 ☆	Lihtne	Python
Prefect	2018	14 600 ☆	Väga lihtne	Python
Argo	2017	14 000 ☆	Keskmine	YAML
Kubeflow	2018	13 700 ☆	Keskmine	Python

Tabel 8.1 Töövoogude tarkvara.

Apache Airflow on kõige levinum tarkvara ja selles on kõige rohkem võimalusi, aga seda on kõige keerulisem kasutusele võtta (järsk õppimiskõver). Luigi ja Prefect on Pythoni arendajate jaoks kõige lihtsamad kasutusele võtta, nad on minimalistliku disainiga ja selge loogikaga. MLFlow ja Kubeflow keskenduvad palju rohkem masinõppe mudelite haldamisele kui andmetega seotud töövoogudele. Argo ja Kubeflow on mõlemad Kubernetese konteineritega tegelevad tarkvarad.

Töövoo üldised etapid on järgmised.

1. Sisendandmete eri allikatest (andmebaasid ja andmelaod, veebiteenused, andmevood) kokku korjamine. Keerulisematel juhtudel tegeleb sellega andmeinsener.
2. Sisendandmete liitmine.
3. Sisendandmete kvaliteedikontroll. Peame tagama, et mudel saab just selliseid andmeid, nagu ta ootab, täiesti teistsugustele andmetele mudel ei üldistu. Selleks salvestame enne mudeli treeningandmete omadusi (nt iga tunnuse keskmise ja standardhälbe arvuliste tunnuste puhul). Pikemalt tutvume sellega monitooringu peatükis.
4. Tunnuste transformeerimine identselt sellega, kuidas see oli tehtud treenimise jaoks (kategooriliste tunnuste kodeerimine, tunnuste teisendamine, standardimine, transformeerimine jne).
5. Serialiseeritud mudeli väljakutsumine. Mudeli serialiseerimine on mudeli seisu (parameetrite ja hüperparameetrite) salvestamine treenimise lõpu hetkel.
6. Tulemuse tagastamine. Näiteks klassifitseerimise tulemus võib olla tagastatud nii tõenäosustena kui ka klassina, olenevalt vajadusest.
7. Mõõdikute salvestamine monitoorimiseks (nt, mis tulemus tagastati, kui kiiresti, mis päringu peale).

Lõksud
Tasub mõelda, mis moel võivad juurutatud mudeli sisendandmed muutuda, et süsteemi sisse ehitada vastavad kontrollid. Näiteks, masinnägemise mudel teostab toote kvaliteedikontrolli. Kas võib juhtuda, et keegi kustutab valguse selles ruumis, kus seade asub, ja mudeli sisendpildid muutuvad mustaks? Kas võib juhtuda, et nihutatakse tööpinki ja toode on kaadris teises kohas, kui ta oli treeningandmestikus?

8.2 Kasutuselevõtt

Paljud probleemid uue andmeteaduslahendusega tulevad esile ainult siis, kui see on juurutatud. Pärast uue soovitussüsteemi juurutamist näidati kõigile kasutajatele, vaatamata sellele, kas nad olid eakad naised, noored emad või teismelised, sama soovitust: ostke kuuldeaparaat. Põhjus oli selles, et kasutajaliidese arendamisel oli kohahoidjana kasutatud fikseeritud juhuslikku toodet andmebaasist, see koodiosa oli eemaldamata jäänud ja mudeli tagastatud tulemused ei jõudnudki tegelikult lõppkasutajani.

Selleks, et testida oma toodet väiksel hulgal kasutajatest, on mitu varianti.

- **Varjurežiim** (ingl k *shadow mode deployment*) – töövoog on sisse lülitatud ja mudel ennustab reaalsetel andmetel, aga ennustusi ei kasutata pärissüsteemis, vaid salvestatakse analüüsimise eesmärgil. Kui on olemas mudeli eelmine versioon, võib tulemusi võrrelda selle mudeli tulemustega, kui seda ei ole, siis, näiteks, võib inimene kontrollida tulemust käsitsi.
- **Kanaarilinnu režiim** (ingl k *canary mode deployment* või *silent live*) – ainult väike protsent päringutest suunatakse uue mudeli juurde (väiksele protsendile kasutajatest näidatakse uut lahendust).
- **Siniroheline juurutamine** (ingl k *blue-green deployment*). Kui on olemas mingi eelmine lahendus, siis osale kasutajatele näidatakse vana ja osale uut lahendust (sinine on vana ja roheline on uus).

8.3 Latentsus, jõudlus ja läbilaskevõime

Masinõppeinseneril on vaja saada ärilt vastused paljudele küsimustele, enne kui ta suudab otsustada, kus ja mis kujul on mudelit kõige parem juurutada. Kõige olulisemad küsimused on järgmised.

1. Kui suurt latentsust saab endale lubada? Mis on pikim lubatud lõppkasutaja ooteaeg? Mõnikord ei ole latentsus üldse probleem. Ennustamist saab jooksutada skriptiga kord päevas või isegi kord nädalas ja tulemused ootavad, kuni neid vaja läheb. Selline on, näiteks, lahkuva kasutaja ennustamine. Mõnikord aga on suurim lubatav latentsus millisekundites (nt, soovitused peavad ilmuma enne, kui kasutaja lõpetab trükkimise).
2. Kui suur peab olema läbilaskevõime ehk mitu päringut sekundis peab server olema võimeline töötlema? See parameeter on oluline siis, kui teenust kasutavad korraga mitu kasutajat.
3. Kui palju me tohime teenuse peale kulutada? Võimsam tarkvara ja kiirem reageerimine on kallid. Vastuse sellele küsimusele annab tasuvusanalüüs.
4. Kas lahendus töötab pilvearvuti peal või kasutaja seadme peal (ingl k *edge deployment*). Pilvelahendus nõuab päringute saatmist interneti teel, mis multimeediaandmete korral võib olla nii aeglane kui ka kallis. Kasutaja seadmel aga on piiratud arvutusressurss.

Paraku saab keerulise andmeteadusülesanne korral valida ainult kaks kolmest – kas mudel ennustab kiiresti, odavalt või täpselt.

8.4 Monitoorimine

Andmeteaduslahenduse õige toimimine sõltub nii sisendandmete kvaliteedist kui ka toetavast taristust. Monitoorimistöölaud peab näitama nii lahenduse enda latentsust kui ka nende ressursside latentsust, millest lahendus sõltub, ja masinõppe mudeli korral ka mõõdikute muutust ajas.

Masinõppe mudelid kirjeldavad andmetes sisalduvaid seoseid maailmas, mis on pidevas muutumises, ja andmeteaduslahendused teenindavad kasutajate vajadusi, mis samuti pidevalt muutuvad. Ärid laienevad teistesse riikidesse, konkurendid võtavad ära või vabastavad nišše, hooajalisus ja mood toovad muutusi. Masinõppe mudeli korral peab monitoorima kahe probleemi esinemist andmetes: andmetriiv (ingl k *data drift*) ja kontseptsioonitriiv (ingl k *concept drift*).

Andmetriivi korral muutub sõltuvate muutujate (sisendandmete) jaotus. Näiteks, inflatsiooni tagajärjel tõusevad hinnad. Kontseptsioonitriivi puhul muutub sõltuvate ja sõltumatute muutujate suhe. Näiteks, COVID-19 pandeemia sundis inimesi muutma oma ostmisharjumusi. Paljud inimesed, kes ei olnud kunagi veebipoodidest midagi ostnud, hakkasid seda tegema. Krediitkaardipettuste tuvastamise mudelid tagastasid selle tagajärjel palju valepositiivseid tulemusi.

Selleks, et tuvastada probleeme andmejaotuse muutusega, on vaja salvestada oma mudeli treeningandmete jaotused (keskmine, standardhälve) ja andmetüüp (nt, kas oodatakse ujukomaarvu või täisarvu). Uute andmete kogumisel võrreldakse algset ja uut jaotust, näiteks, kasutades Kullbacki-Leibleri lahknevust (ingl k *Kullback-Leibler divergence*). Arvutatud mõõdikutest suure muutuse avastamiseks võib kasutada näiteks Chebyshevi kaugust, mis defineerib kahe vektori vahelise kauguse kui suurima kauguse nende koordinaatide vahel.

Vaatame Bolti inseneri jagatud näidet andmejaotuse muutuse kohta. Uus äsja juurutatud petturluse avastamise mudel hakkas tuvastama kolm korda rohkem petturlust. Nädal varem kanaarilinnu režiimis käitus mudel normaalselt. Põhjus peitus ühe tunnuse arvutamise koodi muutumises ehk ühe sisendtunnuse jaotuse muutuses. Koodi, mis arvutas sisendtunnust „kliendiga seotud teiste klientide arv“, muudeti just mudeli juurutamise päeval.

Soovitused

1) Selleks, et tuvastada andmetriivi, võib ka ennustada tunnuste põhjal ajahetke, millest andmepunkt pärineb. Tunnused, mis on selle ennustuse jaoks olulised, võivad olla ajas muutunud.
2) Testige tunnuste efekti monotoonsust – kui mingite tunnuste kohta on teada, et nende mõju on tugevas korrelatsioonis väljundiga, siis testige seda juurutamise ajal. Kui tunnuse arvutamise loogika on muutunud, siis see võib seal peegelduda.
3) Jälgige üksikjuhtumeid, mille ennustus on teistest väga erinev.

Järgmises tabelis on toodud monitoorimistöölaua mõõdikud, mille hulgast tuleb valida enda projekti jaoks asjakohased.

Infrastruktuur	Andmekvaliteet	Väljund
• Operatiivmälu hõivatus • Arvutusressurss • Latentsus • Läbilaskevõime • Serveri koormus • Internetiühenduse kiirus	• Puuduvate väärtuste arv • Sisendi pikkus • Tunnuse keskmine ja andmetüüp • Sensorite seisund • Kas kasutajad kordavad sama päringut (võib viidata sellele, et nad ei saa vastust)	• Ennustuste jaotus • Kui sageli tagastatakse null väärtust • Läbiklikkimse määr *(*ingl k *click-through rate, CTR)*

Tabel 8.2. Monitoorimistöölaua meetrikad.

On oluline silmas pidada, et mudeli juurutamisega mudeli elu alles algab ja andmeteadlase ülesanne on nüüd jälgida, kuidas mudel käitub reaalsetel andmetel, et seda vajaduse korral parandada.

Praktiline näide
Lahenduse juurutamise praktiline näide Google Colabis: link.

Enesekontrolli küsimused

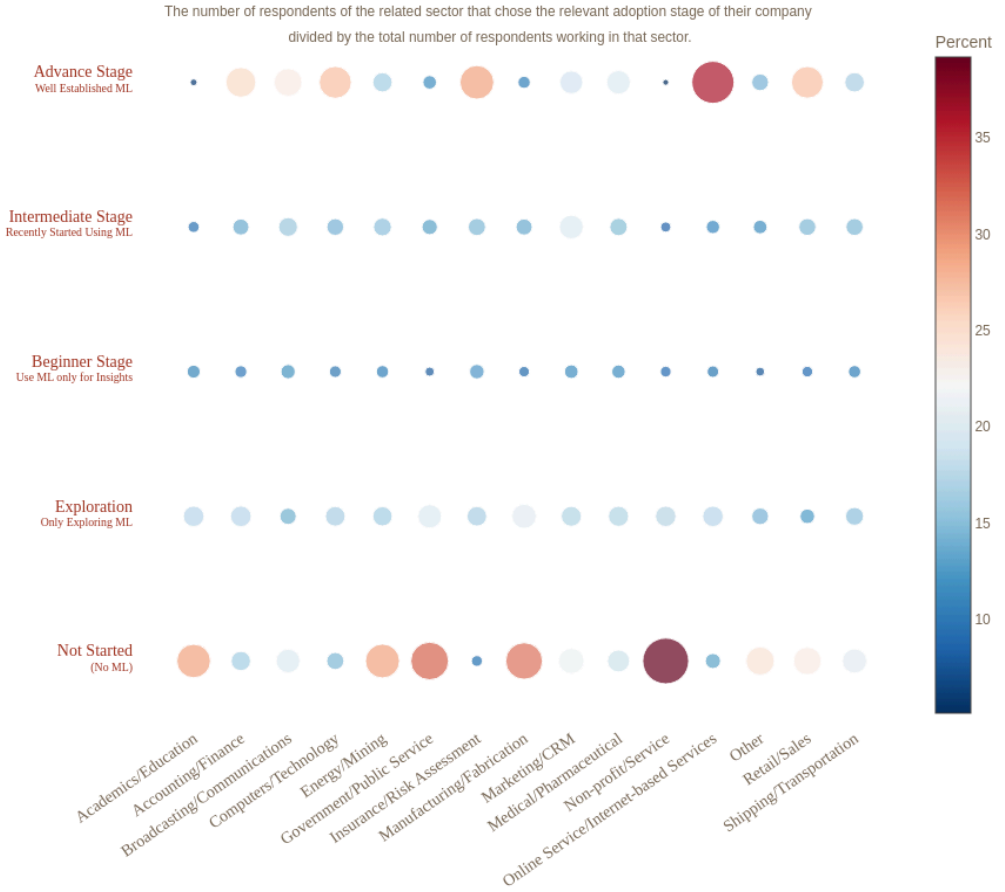
1) Aastal 2008 lubas Netflix anda miljon dollarit sellele, kes oskab parandada Netflixi soovitussüsteemi 10% võrra. Aastal 2009 maksid nad miljon dollarit võitjale. Lahendust ei võetud aga kunagi kasutusele. Miks?

2) Puidutehas toodab toormaterjalist ehituslaudu. Selleks, et määrata puidutüki kvaliteeti, kasutatakse masinnägemissüsteemi, mis peab klassifitseerima iga puidutüki esimesse (ilma oksakohtadeta, sile, ilus), teise (mõni oksakoht, aga mitte väga suur, sile) või kolmandasse klassi (palju oksakohti, võib esineda mõni muu defekt). Milliseid mõõdikuid te kasutaksite sellise süsteemi monitoorimiseks?

3) Milline režiim mõjutab äri rohkem, kas varjurežiim või kanaarilinnu režiim?

9. Rakenduslikud näited

Selles peatükis näitlikustame valdkondi, milles andmeteadust saab rakendada. Andmeteaduse rakendusvaldkondi on lõputult ning siinkohal jäävad pikemalt käsitlemata olulised valdkonnad nagu avalik sektor, inseneeria, ehitus jne. Teame, et kõigis eluvaldkondades kasutatakse andmeteadust paremate otsuste tegemisel. Siiski pole kõik andmeteaduse meetodid erinevatel rakendusaladel samaväärselt levinud. Teatud juhtudel piisabki otsuste tegemiseks mõnest arvust ja graafikust (kirjeldav analüüs), lisaks pole teatud juhtudel musta kasti tüüpi keerulised masinõppe lahendused aktsepteeritavad. Masinõppe levimust eri eluvaldkondades kirjeldab joonis 9.1. Andmeid andmeteaduse kasutatavuse kohta laiemalt ei ole kahjuks lihtne leida.



Joonis 9.1. Masinõppe levimus valdkonniti. *Kaggle 2022. aasta uuringu alusel. Teatud valdkondades ei luba regulatsioonid seletamatute otsustega masinõppemudeleid või on olemasolevad meetodid juba piisavalt edukad, näiteks kindlustuse ja riskide hindamise valdkonnas. Avalik sektor, kolmas sektor ja haridusvaldkond on samuti masinõppe kasutuselevõtul aeglasemad.* [50]

Järgnevates peatükkides kasutame õppevahendina muu hulgas taas koodinäiteid Google Colabi keskkonnas, mis illustreerivad andmeteaduse meetodite rakendamist tänapäevaste koodiraamatukogude abil. Näited on üles ehitatud CRISP-DM (ingl k *cross-industry standard process for data mining*) raamistiku järgi, kattes lühidalt kõik projekti sammud.

[50] Kuvatõmmis, allikas: kaggle.com, litsents: Apache 2.0.

9.1 Kaubandus

9.1.1 Soovitussüsteemid

Soovitussüsteemid on tänapäeva veebilehekülgede üks nähtamatu, aga üldlevinud koostisosa. Kaubandus, uudisteportaalid ja muidugi sotsiaalmeedia – kõik kasutavad soovitussüsteeme, et suurendada ristmüüki ning meelitada kasutajat leheküljel rohkem aega veetma. McKinsey[51] andmetel tuleb 35% Amazoni veebipoe müügist soovitussüsteemide pakutud toodetest.

Peatükis 5 tutvusite populaarsemate juhendatud ja juhendamata õppe mudelitega. Soovitussüsteemide kohta võiks öelda, et nad kuuluvad juhendatud õpe alla, kuna nende loomisel kasutatakse kasutaja genereeritud andmeid (ostud, vaated, klikid, hindamised), aga klassikalised juhendatud õppe mudelid, millega oleme tutvunud, selleks ülesandeks üldjuhul ei sobi, kuna klasside arv oleks liiga suur. Selle asemel kasutatakse teisi meetodeid: maatriksi faktoriseerimine, LambdaMART, RBM (ingl k *restricted Boltzmann machines*), millest esimene on kõige populaarsem.

Andmed, mida kasutatakse, võivad olla kahte tüüpi:

- **otsesed hinnangud** (ingl k *explicit ratings*) – kasutaja tagasiside on väljendatud otseselt – kasutaja on jätnud arvustuse, hinnangu, vajutanud „meeldib";
- **kaudsed hinnangud** (ingl k *implicit ratings*) – kasutaja tegevus, mida me võime interpreteerida kui huvi – sirvimisajalugu, ostuajalugu, tootelehel veedetud aeg, reklaamidel tehtud klikid.

Peale selle võib soovitussüsteeme jagada veel kahte tüüpi:

- **kaasfiltreerimine** (ingl k *collaborative filtering*) – nende süsteemide lähtepunkt on otseste või kaudsete hinnangute maatriks, mille põhjal keskenduvad seda tüüpi mudelid sellele, et leida kasutajaid, kes käituvad sarnaselt (tunnevad huvi sarnaste toodete vastu), või tooteid, mis käituvad sarnaselt (neid ostetakse sageli koos);
- **sisupõhine soovitus, filtreerimine** (ingl k *content-based recommendation, content-based filtering*) – need süsteemid leiavad sarnaseid tooteid nende sisu, olemuse järgi (kirjeldus, muu teada olev seotud info, märgendid). Näiteks, kui tegu on filmidega, võib sisupõhine soovitussüsteem kasutada filmi peaosatäitjaid, väljalaskeaastat, žanrit, et leida sarnaseid filme, mida soovitada.

Koostööpõhise filtreerimise süsteemid ei tööta juhul, kui toode on uus ja keegi ei ole sellele hinnangut andnud. Sellist probleemi nimetatakse külmaks stardiks (ingl k *cold start*). Sellise probleemi lahendamiseks võib kasutada sisupõhist soovitust.

Praktiline näide soovitussüsteemi loomisest
Soovitussüsteemi loomise praktiline näide Google Colabis: link.

[51] MacKenzie, I., Meyer, C.,Noble, C. (2013). How Retailers Can Keep Up with Consumers. *McKinsey & Company*.

9.1.2 Müügi prognoosimine

Müügi prognoosimise käigus hinnatakse tulevast müüki ajalooliste andmete ja turuanalüüsi põhjal. Täpne müügiprognoosimine on oluline mitmel põhjusel. Esiteks aitab see säilitada optimaalset laoseisu, vähendades hoiustamiskulu ja vältides laovaru lõppemist. Teiseks aitab see tõhusalt jaotada ressursse, nagu tööjõud, eelarve ja turundustegevus. Müügiprognoosimine on ka finantsjuhtimise oluline komponent, sisend eelarvestamisele, finantsplaneerimisele. See aitab tuvastada riske ja võimalusi, võimaldades ettevõtetel olla riskijuhtimisel proaktiivne.

Lühiajaline prognoosimine

Lühiajaline prognoosimine hõlmab tavaliselt mõne päeva kuni mõne kuu pikkust perioodi. Seda tüüpi prognoosimist kasutatakse peamiselt varude haldamiseks, personali planeerimiseks ja igapäevasteks toiminguteks. Jaemüügisektoris on lühiajaline prognoosimine ülitähtis müügikasvu prognoosimiseks pühade ja eriliste sündmuste ajal, et tagada tööjõu ning laovaru olemasolu. Lühiajalise prognoosimise kõige levinumad meetodid hõlmavad aegrea analüüsi tehnikaid, nagu liikuva keskmise arvutamise meetodid (sh eksponentsiaalse silumisega), mis aitab andmete kõikumisi siluda, ARIMA (autoregressiivse integreeritud liikuva keskmise) mudelid, mis tuvastavad lühiajalisi regulaarseid tõuse ja langusi (nt nädala jooksul), ning STL (ingl k *seasonal-trend decomposition using LOESS)*, mis proovib tuvastada eraldi nii hooajalisi muutusi kui ka andmetes esinevat trendi.

Keskmise pikkusega prognoosimine

Keskmise pikkusega prognoosimine hõlmab mõne kuu kuni paari aasta pikkust perioodi. Seda kasutatakse eelarvestamiseks, finantsplaneerimiseks ja turundusstrateegiateks. Tootmissektoris on keskmise pikkusega prognoosimine hädavajalik nõudluse prognoosimiseks ja tootmisplaanide optimeerimiseks. Keskmise pikkusega prognoosimise puhul kasutatakse muu hulgas masinõppe regressioonimeetodeid nagu lineaarne regressioon, otsustusmetsad ja tehisnärvivõrgud. Need meetodid aitavad tuvastada müügi ja seda mõjutavate tegurite, näiteks hinna ja kampaaniate, vahelisi seoseid. Ka aegridade analüüsi meetodid nagu SARIMA (hooajaline, ingl k *seasonal* ARIMA) ja ARIMAX, mis võtab peale müügiväärtuste aegrea arvesse ka sõltumatuid muutujaid. Lisaks on võimalik kasutada majandusteooriat arvesse võtvaid mudeleid, ökonomeetrilisi mudeleid, mis arvestavad näiteks keskmise palga tõusu, intressimäärasid, majanduskasvu.

Pikaajaline prognoosimine

Pikaajaline prognoosimine hõlmab mitme aasta pikkust perioodi ning seda kasutatakse strateegiliseks planeerimiseks, turuosa laiendamiseks ja investeerimisotsusteks. Finantssektoris on pikaajaline prognoosimine oluline tulude prognoosimisel ja riskide hindamisel. Masinõppe mudeleid nagu otsustusmetsad ja närvivõrgud kasutatakse pikaajalise prognoosimise puhul üha enam. Need mudelid suudavad töödelda suuri andmekogumeid ja keerukaid mustreid, pakkudes sügavamat ülevaadet tulevastest müügitrendidest ja aidates ettevõtetel teha teadlikke strateegilisi otsuseid. Samuti on kasutusel ökonomeetrilised lähenemised. Et tulevik pole kunagi kindel, võib olla kasulik

analüüsida mitut stsenaariumi, seades teatud parameetrid või eeldused erinevatele väärtustele.

Müügi prognoosimise tööriistad ja tarkvara

Müüki saab prognoosida nii tabelrakenduste, spetsialiseeritud tarkvara kui ka programmeerimispõhiste lähenemistega. Tabelitarkvara nagu Microsoft Excel ja Google Sheets kasutatakse tavaliselt põhiprognoosideks nagu keskmiste ja trendide arvutamine, kuid seal leidub ka keerulisemaid funktsioone. Näiteks Excel pakub sisseehitatud liikuvate keskmiste ja eksponentsiaalse silumise funktsioone ning regressioonanalüüsi. Spetsialiseeritud prognoositarkvara võib ühenduda ERP (ingl k *enterprise resource planing*) ja CRM (ingl k *customer relationship management*) tarkvaraga ning pakub automaatseid analüüse ja tulemusi. Nende lahenduste hulka kuuluvad SAP Integrated Business Planning, Oracle Demand Planning Cloud, Anaplan jne.

Keerukamate ja kohandatud prognoosimisvajaduste jaoks on programmeerimiskeeled, nagu Python ja R, mille teegid võimaldavad arendada keerukaid prognoosimismudeleid, näiteks aegridade analüüsi mudeleid ning ennustavaid mudeleid, mis on kohandatud konkreetsetele ärivajadustele. Need programmeerimispõhised lähenemised pakuvad suuremat paindlikkust ja täpsust.

Praktiline näide müügi prognoosimise kohta
Müügi prognoosimise praktiline näide baseerub Kaggle andmeteaduse võistluste platvormil avaldatud andmestikul Rossmani apteegiketi müügitulemuste kohta. Vaadake CRISP-DM ülesehitusega näidisprojekti siit: link.

9.2 Tootmine

9.2.1 Kvaliteedikontroll

Kvaliteet tööstustootmises viitab toote või teenuse vastavusele määratletud standarditele ja kliendi ootustele. See hõlmab mitmesuguseid omadusi, nagu vastupidavus, funktsionaalsus, töökindlus ja esteetika. Kvaliteetsed tooted täidavad oma ettenähtud eesmärgi ja pakuvad positiivset kasutajakogemust, aidates kaasa klientide rahulolule ja brändilojaalsusele.

Kvaliteet on tööstusliku edu nurgakivi mitmel põhjusel.

- **Kliendirahulolu:** kvaliteetsed tooted, mis vastavad kliendi ootustele või ületavad neid, toovad kaasa suurema rahulolu ja kordusostud.
- **Konkurentsieelis:** ettevõtted, mis on tuntud oma kvaliteedi poolest, saavad endale tihti lubada kõrgemaid hindu. Tootemargi hea maine tagab nõudluse pikema aja jooksul.
- **Kulude vähendamine:** defektide vähendamine ja tootmise järjepidevuse tagamine vähendab jäätmeid ja uuesti töötlemise kulu, samuti tagastatud toodete asendamise kulu ning leppetrahve.

- **Nõuetekohasus:** kvaliteedistandardite järgimine on sageli vajalik tööstusharu regulatsioonide järgimiseks, õiguslike karistuste vältimiseks ja ka edasimüüjate nõuete täitmiseks.

Samal ajal pole üldsegi üheselt selge, mida kvaliteet tähendab. Erinevate toodete puhul hindame me erinevaid asju ja eri tüüpi klientide jaoks on olulised erinevad aspektid. Tihti hinnatakse mitte-funktsionaalseid omadusi ja paljud olulised omadused on peidetud. Kurikuulus on näide, et tohutus koguses puuvilju visatakse minema, sest need ei vasta „kvaliteedile" oma välimuse, kuju või värvi poolest. Mõne tööstusettevõtte jaoks aga ei pruugi sissetulevate materjalide välimus oluline olla. Peamised kvaliteedi aspektid, mida saavutada püütakse, on

- **jõudlus** – kui hästi toode oma ettenähtud funktsiooni täidab;
- **lisaomadused, kohandatavus** – toode võib täita peale oma põhifunktsiooni veel mitut funktsiooni;
- **töökindlus** – jõudluse järjepidevus aja jooksul;
- **vastupidavus** – toote eluiga ja kulumiskindlus, kas toodet saab parandada;
- **vastavus** – regulatiivsete standardite ja spetsifikatsioonide järgimine;
- **esteetika** – toote välimus, kasutusmugavus ja üldiselt tunne, mis kasutajal toodet kasutades tekib, ehk tajutud kvaliteet. Tajutud kvaliteeti võivad mõjutada tootemark, turundus ja kasutajakogemus.

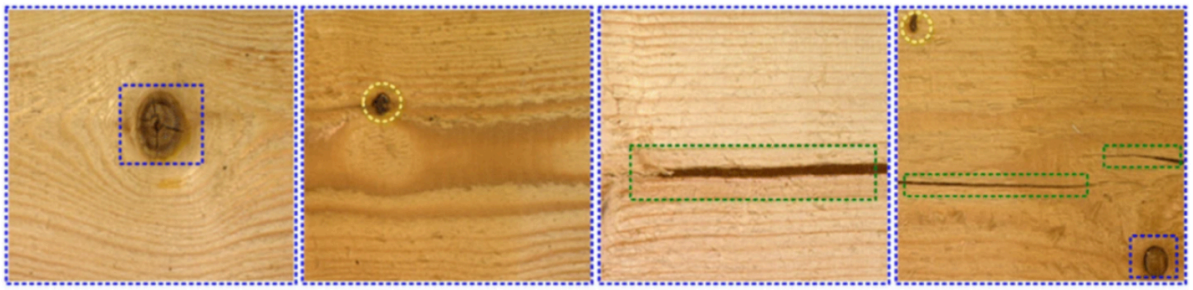
Eri sihtrühmade puhul on olulised erinevad aspektid. Jaeklient ei ole ehk võimeline hindama toote töökindlust, sest sujuv jõudluse vähenemine võib jääda märkamata. Oma andmeid analüüsiv ettevõte võiks aga oma tööriistade jõudluse vähenemist märgata.

Nii nagu andmeteaduses, on ka tootmises sisendid ülimalt olulised.

Sisendite kvaliteedikontroll

Kui tarnija seda teinud pole või me teda ei usalda, on mõistlik kontrollida oma tegevuse sisendite kvaliteeti. Samuti on võimalik, et materjalid on kahjustada saanud transpordi või ladustamise käigus.

Metalli, puidu, kanga, plastplaatide jpt sisendmaterjalide puhul on võimalik teha visuaalne kvaliteedikontroll, et tuvastada plekke, auke, kriimustusi, mõrasid, paindeid jms. Toorme kontrollimine töötaja poolt on aeganõudev ja mittetäielik – tooret on liiga palju, et seda kõike tähelepanelikult üle vaadata. Seega arendataksegi masinnägemisel põhinevad toorme kvaliteedi kontrollimise lahendusi. Siinkohal aga märgake, et masinnägemist saab kasutada ka muude seadmetega omandatud piltidel, näiteks röntgenipiltidel. Kui pragunenud või oksakohtadega laud ei jõua saepingile, säästame me töötaja aega. Samuti saame kvantitatiivse hinnangu toorme kvaliteedile ja võimaluse vajaduse korral tarnijat vahetada või kaebuse esitada.



a) oksakoht b) ussiauk d) pragu d) kolm defekti

Joonis 9.2. Tootmist häirivate või lõpptoote kvaliteeti alandavate omaduste tuvastamine saematerjalil. *Oksakohti, ussiauke ja pragusid on võimalik tuvastada rohkem kui 90% täpsusega.* [52]

Toorme puhul on tihti oluline selle mõõtmete nõuetekohasus. Juhuslikel sisenditel tehtud mõõtmiste alusel jälgitakse statistilisi väärtusi (keskmine, standardhälve, miinimum, maksimum). Nende statistiliselt olulisel nihkumisel aja jooksul peab olema mingi põhjus.

Pooltoote kvaliteedikontroll

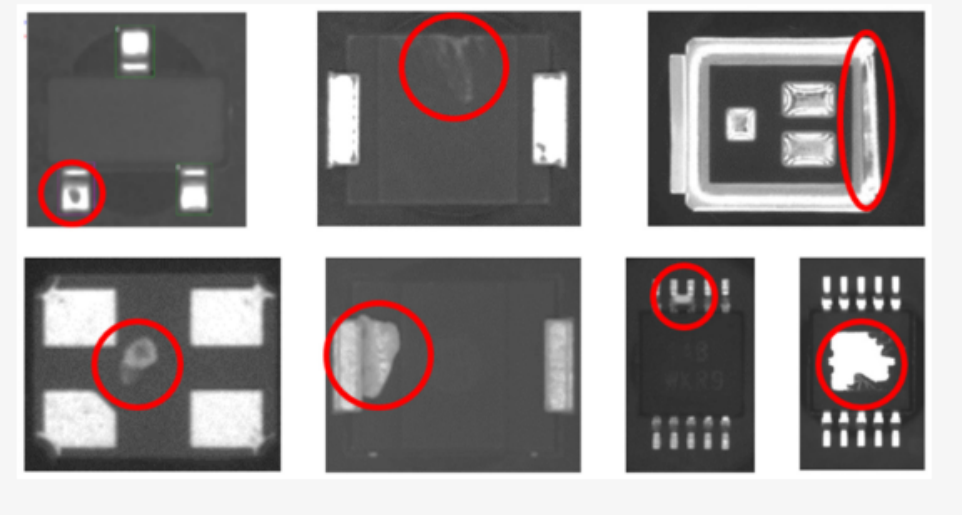
Nii nagu katkisest või määrdunud toormest pole mõtet hakata toodet tegema, pole mõtet tegevust toote kallal jätkata, kui tootmise käigus tekib probleem. Näiteks autotööstuses keevitavad robotid kokku suuri detaile – kui mõni keevituskoht on ebaühtlane, viltu või muul viisil ebakvaliteetne, pole mõtet seda detaili edasi värvimisfaasi saata. Defektid võivad tekkida kuumutamisel (põletusjäljed), toote vale asendi tõttu värvimisel, aga ka juhuslikult liinil teiste toodete vastu hõõrudes või kriimustades. Mida varem defekt avastatakse ja vigane toode liinilt eemaldatakse, seda parem. Siingi võib kasutada masinnägemise lahendusi.

Lõpptoote kvaliteedikontroll

Märgake, et pakendatud lõpptoote puhul pole mõnda defekti võib-olla enam näha. Siiski tekib just pakendamisel veel lisadefekte, viltu kleebitud või puuduvaid silte, rebenenud või mõranenud pakendeid. Seega võib kvaliteeti kontrollida nii enne kui ka pärast pakendamist. Pakendi kvaliteet on eriti tähtis näiteks ravimite ja toidukaupade puhul.

Lõpptoodangu kvaliteedikontrolli klassikaline näide on elektrooniliste trükkplaatide tootmisel, kus jootmisvigade, kriimustuste jms tuvastamiseks on kasutusel masinnägemisel baseeruvad lahendused. Selliste toodete puhul on see ainus võimalus, sest mahud on suured, detailid väikesed ja inimese tehtav kvaliteedikontroll võtaks palju aega.

[52] *Kuvatõmmis, allikas: (Cui, et. al., 2023), link litsentsile.*



Joonis 9.3. Masinnägemine aitab elektroonikatööstuses defekte tuvastada.[53]

Masinnägemine on olulisel kohal ka põllumajandussaaduste ja toiduainetööstuse toodete kvaliteedikontrollis. Sellealane praktiline näide: tootmisdefektide tuvastamise peatüki lõpust leiate koodinäite defektsete toodete tuvastamisest, iseseiva tööna võite luua kõlblikke ja mädanenud õunu klassifitseeriva mudeli.

Toodete kvaliteedi jälgimine nende kasutuse jooksul

Tootja võib saada toote kvaliteedi kohta infot ka pärast toote müümist. Kogudes andmeid raporteeritud probleemide kohta, aga ka jälgides klientide ostukäitumist (uute varuosade ostmise sagedust), saame teha järeldusi näiteks oma toodete vastupidavuse kohta. Tootearvustusi ja kasutajate kommentaare keeletöötlusvahendite ning statistiliste meetodite abil analüüsides võib avastada meile veel seni teadmata vigu, mis juhtuvad harva, kuid õõnestavad tootemargi mainet. Laiatarbeelektroonika tootjad analüüsivad oma toodetele antud kommnetaare erinevatel platvormidel, et tüüpvigu tuvastata, nende tekkepõhjus firma sees välja uurida ja protsesse parandada.

Praktiline näide: tootmisdefektide tuvastus
Tootmisdefektide tuvastamine on ülitähtis kvaliteedikontrolli osa, mis aitab ettevõtetel säästa aega ja ressursse. Automaatne defektide tuvastus, kasutades tehisintellekti, võimaldab kiirelt ja täpselt eristada terveid ja defektseid tooteid, tagades seeläbi kõrgema kvaliteedi ning vähendades inimeste manuaalse töö vajadust. Lingil olevas Google Colabi vihikus õpite, kuidas luua mudelit automaatseks defektide tuvastamiseks: link.

[53] Allikas: (Weiss, et. al., 2024), litsents: CC BY 4.0.

9.3 Suurte keelemudelite rakendused

9.3.1 Generatiivse tehisintellekti ja suurte keelemudelite arenguhüpe

Viimastel aastatel on suured keelemudelid (ingl k *large language model*, LLM), mis kuuluvad generatiivse tehisintellekti valdkonda, teinud suure arenguhüppe, mis on kasvatanud ühiskonna teadlikkust ja huvi nende kasutamise vastu. LLM-id suudavad luua arusaadavaid ja inimlikega sarnanevaid kirjalikke lauseid, matkides seeläbi inimintellekti. OpenAI[54] loodud LLM-idele tuginev populaarne juturobot ChatGPT[55] on andnud inimestele võimaluse kogeda, kui mugav on selle tööriista abil otsida avalikult kättesaadavat informatsiooni. Sellest inspireerituna ootavad nii kasutajad kui ka organisatsioonid sarnast mugavust vastuste leidmisel oma organisatsiooni privaatsetest dokumentidest. Andmeteadlased ja tarkvaraarendajad on hakanud looma selliseid LLM-idel põhinevaid lahendusi, millest üks järjest populaarsem arhitektuuriline lähenemine on allikapõhine genereerimine (ingl k *retrieval-augmented generation*, RAG).

LLM-idel põhinevad süsteemid suudavad töödelda ja mõista tohutul hulgal struktureerimata vaba teksti andmeid, võimaldades organisatsioonidel kiiresti saada vajalikku informatsiooni. Selliste süsteemide abil saavad ettevõtted, näiteks, automatiseerida rutiinseid ülesandeid, tõhustada otsustusprotsesse ja parandada kliendikogemust. Näiteks suudavad LLM-idega varustatud vestlusrobotid hallata kliendipäringuid, isikupärastada suhtlust ja vähendada inimtöötajate töökoormust. Generatiivne tehisintellekt annab töötajatele ka võimaluse võtta kokku pikki dokumente, luua sisu ja intuitiivselt pääseda juurde andmebaasidele. Ettevõtete jaoks tähendab see suuremat tootlikkust, väiksemat tegevuskulu ja konkurentsieelist turumuutustega kohanemisel.

Ent kuigi need mudelid paistavad silma keele mõistmise poolest, on neil piirangud, nagu näiteks aegunud teadmised, valdkonnaspetsiifiliste teadmiste puudumine, andmete privaatsusega seotud probleemid ja mõnikord hallutsinatsioonid. Keelemudelite piiranguid käsitlesime üksikasjalikumalt peatükis 6.4.4. Siin tulebki mängu allikapõhine genereerimine, millest võiks mõelda nagu „ChatGPT-st oma dokumentidega“.

9.3.2 Allikapõhise genereerimise tutvustus

Allikapõhine genereerimine on hübriidsüsteem, mis ühendab suurte keelemudelite võimekuse ja välistest andmebaasidest või dokumentidest relevantse teabe hankimise võimaluse. Kujutage ette, et esitasite virtuaalsele assistendile keerulise küsimuse, nagu „Mis on ettevõtte eelmise kvartali kogutulu Euroopa regioonis sisemise aruandluse järgi?“. Ainult LLM-i kasutamine võib anda vale või ebatäpse vastuse, sest mudel ei ole sellist informatsiooni varem õppinud.

Allikapõhine genereerimise süsteem lahendab selle probleemi, tuues esile olulised algdokumendid ja teabekillud usaldusväärsetest allikatest ning genereerides seejärel täpse ja kontekstipõhise vastuse. RAG-süsteem on nagu personaalne assistent, kes

[54] https://openai.com/about/.
[55] https://openai.com/index/chatgpt/.

mitte ainult vastab küsimustele, vaid toob ka vastuseid toetavad dokumendid, raamatud või artiklid. Lihtsamalt öeldes parandab RAG generatiivse tehisintellekti vastuste täpsust ja asjakohasust, lisades neile reaalajas juurdepääsu ettevõtte või avaliku sektori organisatsiooni teadmistebaasile.

See lähenemine on eriti oluline organisatsioonidele, kus täpsus ja valdkonnapõhisus on kriitilise tähtsusega. RAG-süsteemid leiavad rakendust paljudes valdkondades, pakkudes tõhusaid lahendusi, näiteks, järgmistele keerulistele probleemidele.

- **Tervishoius** aitavad RAG-süsteemid kaasa tervisealase kirjaoskuse arendamisele, pakkudes lihtsat juurdepääsu usaldusväärsele ja täpsele teabele. RAG-süsteemid aitavad leida usaldusväärset tervisealast teavet, vastates küsimustele lihtsas ja arusaadavas keeles, tuginedes usaldusväärsetele allikatele. See aitab võidelda valeinformatsiooni vastu ja toetab paremate tervisetulemuste saavutamist.
- **Õigusvaldkonnas** võimaldavad RAG-süsteemid vastata keerukatele õiguslikele küsimustele, viidates samal ajal asjakohastele õigusaktidele.
- **Klienditoes** võimaldavad RAG-süsteemid kiirelt ja tõhusalt töödelda suures koguses kliendipäringuid, vähendades inimoperaatorite koormust ja parandades kliendikogemust.
- **Ettevõtete teabe haldamisel ja otsingus** annavad RAG-süsteemid töötajatele võimaluse küsida andmehoidlatest teavet loomulikus keeles, muutes informatsiooni leidmist kiiremaks ja lihtsamaks.

9.3.3 RAG-süsteemi arhitektuur ja tööpõhimõte

RAG-süsteemi arhitektuur on oma olemuselt üsna lihtne, kuid samal ajal väga tõhus. RAG-süsteem koosneb kolmest peamisest komponendist, mis on illustreeritud joonisel 9.4: dokumentide sisestamine, otsing ja vastuse genereerimine. **Dokumentide sisestamise töövoog** valmistab ette ja salvestab dokumente vektorandmebaasi, **otsingusüsteem** leiab asjakohase teabe ning **generatiivne mudel** (ehk LLM) loob saadud konteksti põhjal vastuse. Koos tagavad need komponendid sujuva protsessi andmete ettevalmistamisest asjakohalise vastuse loomiseni.

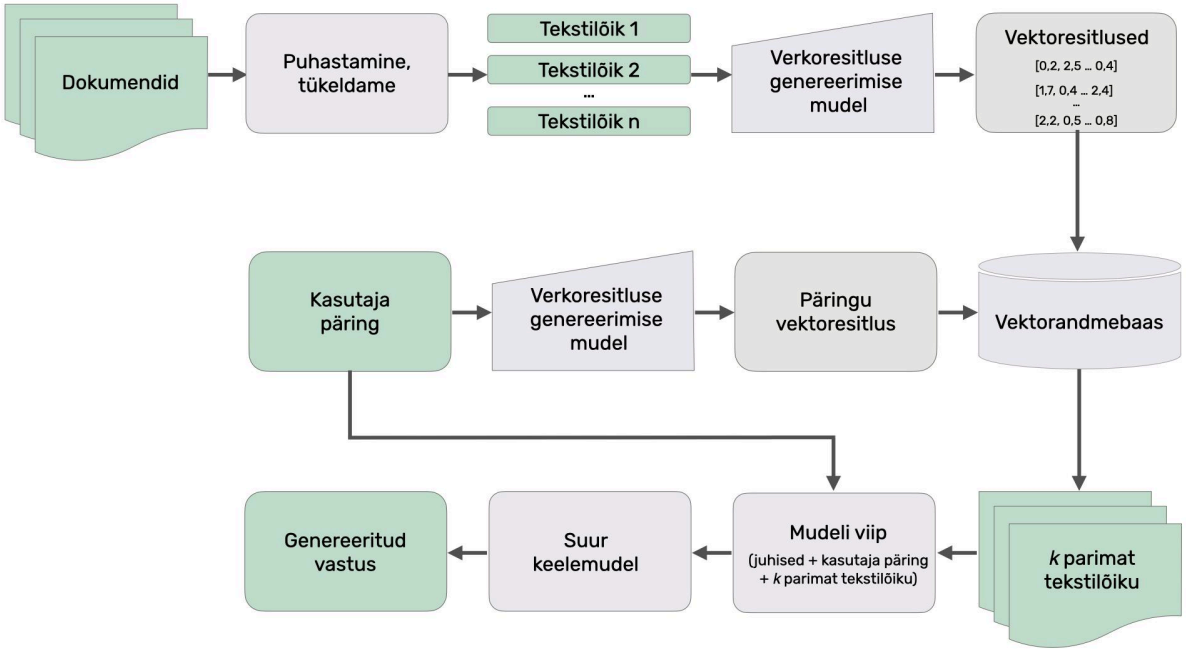
Andmete sisestamise töövoog

Andmete sisestamise töövoog valmistab ette teadmistebaasi, millele RAG-süsteem tugineb täpsete vastuste genereerimiseks. See protsess hõlmab järgmisi samme.

- **Dokumentide laadimine ja teksti ekstraheerimine.** Süsteemi laetakse üles algandmed, näiteks PDF-failid, ettekanded, veebilehed või struktureeritud andmed – tabelid, Exceli failid jpm.
- **Eeltöötlus.** Tekst puhastatakse, tükeldatakse ja jagatakse väiksemateks lõikudeks (ingl k *chunk*) või segmentideks. Teksti jaotamine väiksemateks osadeks tagab, et otsingusüsteem saab teavet tõhusalt töödelda ja asjakohast informatsiooni leida, ilma et andmemaht oleks liiga suur.
- **Vektoresituste genereerimine.** Iga tekstilõik teisendatakse tihedaks vektoriks, mida nimetatakse vektoresituseks (ingl k *embedding*). Need vektoresitused

püüavad teksti semantilist tähendust, võimaldades tõhusat sarnasusepõhist otsingut. Vektoresituste loomisel kasutatakse spetsiaalseid mudeleid, näiteks OpenAI text-embedding-ada-002[56]. Vektoresituste genereerimise mudel mängib olulist rolli, määrates, kui hästi süsteem suudab haarata semantilisi seoseid.

- **Indekseerimine.** Loodud vektoresitused salvestatakse vektorandmebaasi (nt Pinecone[57], ChromaDB[58]). Indekseerimine võimaldab kiiret sarnasusepõhist otsingut. Peale tekstilõikude saab vektorandmebaasi salvestada ka metaandmeid, mis rikastavad otsingut ja konteksti. Metaandmed võivad sisaldada viiteid allikatele, kasutaja rolliga seotud informatsiooni, mis, näiteks, võimaldab rollipõhiste õiguste rakendamist, ja muid seotud andmeid.



***Joonis 9.4. Allikapõhise genereerimise süsteemi ülesehitus**. RAG-süsteemid jagavad ülesanded kolme peamisse etappi: andmete sisestamine, teabe otsimine ja vastuse genereerimine. **Dokumentide sisestamine.** Süsteem alustab dokumentide kogumi sisestamisega. Need dokumendid jagatakse väiksemateks tekstilõikudeks, muudetakse tihedateks vektoresitusteks ja salvestatakse vektorandmebaasi. **Asjakohase teabe otsing.** Kui kasutaja esitab päringu, genereerib süsteem päringu jaoks vektoresituse ja otsib vektorandmebaasist sellega kõige sarnasemaid tekstilõike. See sarnasuse otsing tagab, et isegi suurte andmekogumite korral leitakse ainult kõige asjakohasem teave. **Vastuse genereerimine.** Leitud tekstilõigud ühendatakse kasutaja päringuga, et luua päringupõhine sisend. Seda sisendit töötleb vastuse genereerimise mudel, mis sünteesib vastuse. Mudel integreerib teabe leitud lõikudest ning koostab sidusa ja kasutajasõbraliku vastuse.*

Otsingusüsteem

Otsingusüsteem tuvastab kasutaja päringule vastamiseks kõige asjakohasemad teabekillud. See hõlmab järgmisi samme.

[56] https://openai.com/index/new-and-improved-embedding-model/.
[57] https://www.pinecone.io/.
[58] https://www.trychroma.com/.

- **Päringu vektoresituse genereerimine.** Kui kasutaja esitab päringu, teisendatakse see vektoresituseks, kasutades sama mudelit, mida kasutati andmete sisestamise etapis.
- **Sarnasuse otsing.** Kasutaja päringu vektoresitust võrreldakse vektorandmebaasis salvestatud vektoresitustega, et tuvastada semantiliselt sarnased tekstilõigud. Süsteem kasutab tavaliselt **ligikaudset *k*-lähima naabri algoritmi** (ingl k *approximate k-nearest neighbor,* ANN), et tuvastada kõige asjakohasemad tekstilõigud, mis vastavad kasutaja päringu vektoresitusele. See meetod tagab kiire ja tõhusa otsingu isegi suurte andmestike puhul, ohverdades väikese osa täpsusest kiiruse nimel. Tavaliselt valitakse otsingu tulemusel *k* kõige paremat tekstilõiku, mis edastatakse järgmisesse töötlusetappi. Lisaks võib olla rakendatud **tekstilõikude ümberjärjestamise** (ingl k *reranking*) samm, kus leitud lõike hinnatakse uuesti vastavalt nende sobivusele konteksti ja kasutaja päringu täpsusele. See samm võib kasutada keerukamaid meetodeid, nagu täpsemad keelemudelid või kohandatud hindamisfunktsioonid, et tagada, et lõplikult valitud kontekst on sobivaim vastuse genereerimiseks.

Vastuse genereerimine

Leitud tekstilõigud edastatakse vastust genereerivale mudelile (tavaliselt suur keelemudel, nt GPT4[59], LLama 3.3[60], Claude Sonet 3.5[61]), et koostada lõplik, kontekstitundlik vastus.

- **Viiba loomine.** Leitud tekstilõigud ühendatakse kasutaja päringuga, et moodustada sisend vastust genereerivale mudelile. Tüüpiline mudeli viiba (ingl k *prompt*) sisend võib olla selline:

  > Juhised mudelile: [Instruktsioon 1, Instruktsioon 2, …]
  >
  > Kontekst: [Tekstilõik 1, Tekstilõik 2, ...]
  >
  > Küsimus: [Kasutaja päring]

  See struktuur tagab, et vastust genereerival mudelil on vastuse loomiseks olemas kogu vajalik kontekst.
- **Vastuse genereerimine.** Kasutades viiba edastatud sisendit, loob mudel asjakohase vastuse, mis põhineb allikatest leitud informatsioonil. Oluline on, et vastuse genereerimise mudel oleks suunatud kasutama ainult antud konteksti, et vältida mudeli sisemiste teadmiste kasutamist. Mudeli sisemised teadmised võivad viia eksitavate või ebatäpsete vastusteni. Kasutades ainult valitud tekstilõike, väheneb valede ja väljamõeldud vastuste tõenäosus, tagades täpsemad ja usaldusväärsemad vastused. Koos genereeritud vastusega edastatakse sageli ka **viited originaaldokumentidele või algallikatele**, kust tekstilõigud pärinevad. See aitab kasutajal kinnitada vastuse täpsust ja saada vajaduse korral rohkem kontekstiteavet. Viidete lisamine suurendab läbipaistvust ja usaldusväärsust, võimaldades kasutajal hõlpsasti kontrollida kasutatud allikaid.

[59] https://openai.com/index/gpt-4/.
[60] https://www.llama.com/.
[61] https://www.anthropic.com/news/claude-3-5-sonnet.

Need kolm komponenti töötavad koos, et tagada tõhus ja kontekstitundlik lahendus, mis suudab pakkuda kasutajale täpseid ja asjakohaseid vastuseid.

9.3.4 RAG-süsteemide rakendused eri valdkondades

RAG-süsteemid avavad organisatsioonidele laia valiku võimalusi, täiustades nende suutlikkust andmetega suhelda viisil, mis varem oli mõeldamatu. Tihti on RAG-süsteemid kasutajatele kättesaadavad vestlusroboti kasutajaliidese kaudu. Vestlusrobot võib olla eraldi tarkvara, mis on integreeritud firma veebiportaali assistendina, lisatud veebilehele või integreeritud sisemisse suhtlusrakendusse, nagu näiteks Microsoft Teams[62]. Mõned peamised rakendused võib jaotada järgmiselt.

- **Küsimustele vastamine**. RAG-süsteemid ei tee ainult märksõnapõhiseid otsinguid, vaid suudavad mõista keerulisi loomulikus keeles päringuid ja leida kontekstuaalselt asjakohaseid dokumente või vastuseid. Näiteks töötaja, kes küsib midagi viimase aasta tarkvaraprojektide kohta, saab mitte ainult vastavad dokumendid, vaid ka asjakohase kokkuvõtte.
- **Dokumendiga vestlemine**. RAG võimaldab kasutajatel dokumentidega vestelda. Sellisel juhul võib kasutaja vestlusrobotisse üles laadida oma faili või valida faili organisatsiooni süsteemist kättesaadavate failide hulgast, kasutades filtreid. Kasutaja saab esitada konkreetseid küsimusi dokumendi kohta ja rakendus vastab vastavalt sellele, mis on dokumendi sisus. Näiteks võib õigusmeeskond kasutada vestlusrobotit, et küsida lepingute või õigusaktide kohta ja kiiresti leida teavet spetsiifiliste klauslite kohta.
- **Dokumendi loomine**. Kombineerides genereerimisvõimekuse ja otsinguvõimaluse, saab RAG-süsteem aidata dokumentide koostamisel. Olgu tegemist aruannete, äripakkumiste või personaliseeritud turundusmaterjalide loomisega, allikapõhine genereerimise süsteem tagab, et genereeritud sisu põhineb asja- ja ajakohasel teabel.

RAG-süsteemid on teinud suuri edusamme paljudes valdkondades, pakkudes märkimisväärseid rakendusi tervishoius, valitsuses ja panganduses.

Tervishoiuvaldkonna näide
Üleriigilistes tervisekriisides, nagu COVID-19 pandeemia, kasvab vajadus kiire, professionaalse ja usaldusväärse nõustamise järele. Kui COVID-19 tabas Saksamaad, oli Alam-Saksi liidumaa tervishoiuameti (saksa k Niedersächsisches Landesgesundheitsamt, NLGA) infoliin üle koormatud, sest kodanikud otsisid teavet uue haiguse kohta. Isegi hooajaliste haiguspuhangute korral põhjustab päringute arvu järsk kasv tervishoiuasutuste sidekanalitele suurt koormust. Selle probleemi lahendamiseks otsustas NLGA kasutada tehisintellekti, et muuta suhtlus kodanikega tõhusamaks ja tulevikukindlaks. Selle otsuse tulemusena töötati välja vestlusrobot Nova[63], mis on oma olemuselt allikapõhine genereerimise süsteem. Nova paigutati NLGA veebilehele, et vastata tervisealastele küsimustele,

[62] https://www.microsoft.com/en-us/microsoft-teams/group-chat-software.
[63] https://www.nlga.niedersachsen.de/chatbot.

nagu näiteks „Millist ohtu puugid endast kujutavad?“, „Kas gripi vastu vaktsineeritakse?“ või „Ma soovin lasta oma joogivett kontrollida. Kuidas seda teha?“. Kodanikud saavad küsida COVID-19, leetrite ja muude murede kohta. Vastused saadakse mõne sekundi jooksul. Vestlusrobot on kättesaadav 24 tundi ööpäevas ja suhtleb mitmes keeles. Nova kasutab professionaalselt kureeritud andmebaasi, mida haldab täielikult NLGA. Terviseeksperdid kiidavad teabe enne heaks, vaatavad seda regulaarselt üle ja ajakohastavad. Süsteem järgib rangeid andmekaitse-eeskirju, vältides andmete väärkasutamist ja tagades kasutaja privaatsuse. Vestlusrobot ei salvesta ega töötle isikuandmeid. Nova suudab kiiresti omandada uut teavet ja kohandada oma vastuseid, muutes selle tulevaste tervisekriiside puhuks tõhusaks lahenduseks.

Õigusvaldkonna näide
Eesti parlament ehk Riigikogu vastutab seaduste vastuvõtmise, õigusaktide ratifitseerimise ja riigi toimimise tagamise eest. Õigusaktide koostamine ja läbivaatamine on Riigikogu üks kõige olulisemaid ja aeganõudvamaid ülesandeid. Riigikogu Kantselei pakub parlamendi komisjonidele ja töötajatele õiguslikku tuge, tagades, et parlamendi arutelud ja otsused oleksid teadmistepõhised. Traditsiooniliselt on Kantselei õigusosakond käsitsi töödelnud suurt hulka infopäringuid, analüüsinud seaduseelnõusid ja vastanud küsimustele. Dokumentide (sh seaduseelnõud, seletuskirjad, Euroopa Liidu dokumendid ja avalikud arhiivid) mahu kasvuga muutus see protsess liiga töömahukaks ja ebaefektiivseks. Kantselei vajas lahendust, mis suudaks tõhustada teabeotsingut, parandada täpsust ja pakkuda kasutajasõbralikku kogemust. Kantselei tegevuse toetamiseks töötati välja tehisintellekti-põhine assistent. Rakendus otsib infot Eesti ja Euroopa Liidu õigusaktidest, varasematest ametlikest päringutest ning muudest avalikest õigusdokumentidest, töötleb saabunud päringuid, leiab vastavad vastused ja esitab need koos viidetega algsetele õigusdokumentidele. See kiirendab otsuste tegemist ja tagab, et kõik vajalikud andmed on seadusandjatele kergesti kättesaadavad.

9.3.4 Praktiline osa. Loome ise RAG-süsteemi

Selles õpiku osas asume tegutsema ja sukeldume põnevasse ülesandesse ehk loome ise nullist allikapõhise genereerimise süsteemi küsimustele vastamiseks.

RAG-süsteemid on tänapäevaste tehisintellekti rakenduste esirinnas, kombineerides teabe otsingu ja generatiivsete mudelite tugevused, et pakkuda väga asjakohaseid ja täpseid vastuseid. Selle asemel, et toetuda üksnes eeltreenitud keelemudelitele, otsib RAG-süsteem dünaamiliselt kõige olulisema teabe teadmistebaasist, tagades, et vastused on nii täpsed kui ka konteksti arvestavad.

Selles osas õpite

- **looma dokumentide sisestamise töövoogu**, eeltöödeldes dokumente ja luues teadmistebaasi;
- **rakendama semantilist otsingumehhanismi**, et tõhusalt leida asjakohast teavet;
- **integreerima generatiivse mudeli**, mis sünteesib vastuseid leitud teabe põhjal.

NB! RAG-süsteemide arendamine võib vajada süsteemi optimeerimist, automatiseeritud vastuse hindamist ja kaitsereeglite (ingl k *AI guardrails*) rakendamist, et maandada generatiivse tehisintellekti kasutamisega seotud riske[64]. Need teemad on tihti kasutusjuhu-spetsiifilised. Loodame, et lugeja on nüüd nendest teemadest piisavalt huvitatud ja suudab iseseisvalt oma teadmisi laiendada. Näiteks uurige LLM Guard[65] ja Nvidia NeMo[66] tarkvara, mis pakuvad rakendustele kaitsemehhanisme. Need on kaks tööriista, mis aitavad tagada tehisintellekti süsteemide turvalisust ja usaldusväärset kasutamist.

Selle praktilise osa lõpus mõistate RAG-süsteemi komponente ning teil on toimiv lahendus, mida saate laiendada ja täiustada vastavalt erinevatele kasutusjuhtudele. Olgu teie eesmärgiks klientide päringutele vastamine või isikliku assistendi loomine, allpool toodud praktiline näide varustab teid vajalike tööriistade ja teadmistega, et alustada.

Praktiline näide allikapõhise süsteemi loomisest
Küsimustele vastamise süsteemi loomise praktiline näide Google Colabis: link.

[64] https://genai.owasp.org/llm-top-10/.
[65] https://llm-guard.com/.
[66] https://www.nvidia.com/en-eu/ai/?ncid=no-ncid.

Lisamaterjalid

Selle õppematerjali esimeses osas räägime üldiselt andmeteadusega seotud töörollidest ja nende erinevatest ülesannetest ettevõtetes. Kursuse lõpetame andmete käsitlemisega kaasnevate regulatsioonide ja võimalike eetikaprobleemide ülevaatega.

Töörollid andmeteaduse projektides

Selle kursuse jooksul oleme näinud, et andmeteadus ei ole üks konkreetselt piiritletud valdkond, see on eri valdkondade teadmiste rakendamise kombinatsioon. Andmeteadus hõlmab kogu tegevust, mis aitab andmete põhjal kasulikke otsuseid teha. Me teame, et andmeteadlased tegelevad andmetega, aga milliseid ülesandeid siis andmeteadusega mingilgi määral seotud töötajad tegelikult lahendavad?

Üks võimalik andmeteadusega seotud töörollide nimekiri on järgmine:

- **andmeinsener** – puhastab, ühendab ja organiseerib eri allikatest pärit andmeid ning sisestab need ühtsetesse andmeladudesse või andmebaasidesse;
- **andmearhitekt** – kujundab, loob ja haldab ettevõtte andmearhitektuuri;
- **andmeanalüütik** – töötleb ja kasutab suuri andmekogusid, et tuvastada sealseid peidetud trende, mille põhjal saab teha strateegiliste äriotsuste langetamiseks kasulikke sisulisi järeldusi;
- **äriteabe spetsialist / ärianalüütik** (ingl k *business intelligence analyst*) – sarnaselt andmeanalüütikuga tuvastab andmetest trende, et aidata ettevõtte väärtust suurendada;
- **andmeteadlane** – kavandab ja viib ellu andmete modelleerimise protsesse, et luua uusi algoritme ja ennustavaid (masinõppe) mudeleid.

Täielikku ametinimetuste ja rollide nimekirja on suhteliselt võimatu tuua. Olenevalt ettevõtte suurusest võivad vastavad tööülesanded ja ametinimetused olla erinevad. Näiteks tunduvad andmeanalüütiku ja äriteabe analüütiku rollid sisuliselt väga sarnased, aga sõltuvalt ettevõtte struktuurist võib sisult samu ülesandeid täitva inimese ametinimetus olla erinev. Suuremates ettevõtetes on tavaliselt ülesanded konkreetsemalt jagatud ja inimene, kes tegeleb analüüsiga, ei pea tegelema andmete kogumise ja haldamisega. Väiksemates ettevõtetes ei ole sageli võimalik iga ülesande jaoks eraldi inimest palgata ja võib juhtuda, et sealne andmeanalüütik peab lisaks analüüsile tegelema veel väga paljude muude andmehaldusega seotud probleemidega. Lisaks võib andmeteadlase sünonüümina (või täiendusena) kohata nimetusi nagu masinõppe teadlane või masinõppe insener. Tüüpiliselt palgatakse andmeteaduse meeskonda luues esimese töötajana andmeinsener või andmeanalüütik.

Siit nimekirjast on näha ka üks võimalik eristus andmeteadlase ja analüütiku vahel. Nimelt tegeleb analüütik pigem andmete uurimisega ning nendest trendide tuvastamise ja raporteerimisega, mis on sisuliselt kirjeldava ja diagnostilise analüüsi meetodite rakendamine. Andmeteadlane tegeleb rohkem nende meetoditega, mis kuuluvad ennustava ja ettekirjutava analüüsi valdkonda. Eesti andmeteaduse kogukonnas

(https://datasci.ee/) on räägitud andmeteadlaste tüüpide kontseptsioonist. Seal tutvustatakse ideed, mis sobitub ka ülaltoodud andmeteadlase ja andmeanalüütiku rollide eristusega. Täpsemalt eristatakse **A-tüüpi** ja **B-tüüpi** andmeteadlasi. A-tüüpi (nagu ingl k ***a****nalysis*) andmeteadlaste hulka kuuluvad analüütikud, kes teevad ühekordseid analüüse, kasutavad valdkonnateadmisi ja tegelevad ettevõttesisese raporteerimisega. B-tüüpi andmeteadlased (nagu ingl k ***b****uilding*) ehitavad automaatseid süsteeme, enamasti millegi ennustamiseks. B-tüüpi andmeteadlane kasutab rohkem masinõppe ja tehisintellekti meetodeid ning oskab ka tarkvaraarendust. Näiteks, A-tüüpi andmeteadlane leiab ettevõtte jaoks kõige väärtuslikumad kliendid ja kirjeldab nende profiili ning B-tüüpi andmeteadlane ehitab süsteemi, mis ennustab ettevõtte e-poe külastajate arvu järgmise nädala jooksul tunniajase täpsusega.

Tutvume nüüd erinevate töörollidega lähemalt.

Andmeinsener

Võttes arvesse, et andmed on andmeteaduse projektide aluseks, on oluline tagada, et andmed on korrektselt hoiustatud ning kõik vajalikud andmed on olemas ja ligipääsetavad. **Andmeinsener** mõtleb välja, kuidas andmeid koguda, organiseerida ja hallata. Tema kontrolli all on andmete liikumine ja ta loob andmete hoiustamiseks arhitektuure – spetsiaalseid andmehaldussüsteeme, andmebaase ja nendega seotud taristut. Andmeinseneri kohustuseks on tagada ka andmetele ligipääs ja andmete töötlemise võimalused, et andmeanalüütikud saaksid oma tööd teha. Ligipääsu puhul tuleb silmas pidada ärisaladuste ja isikuandmete kaitset – hallata, kes saavad üldse mingitele andmetele ligi, ja dokumenteerida (logida) andmete kasutamist.

Suurte andmete hoiustamine on keeruline, sest kõik andmed ei mahu ühte tabelisse. Kõikide andmetega tabel oleks vahel mitmesaja gigabaidi või lausa mitme terabaidi suurune. Nii suured failid mahuvad küll kõvakettale, aga arvuti töömällu (RAM) need ei mahu ja nendega on võimatu edasi töötada. Seetõttu on vaja paigutada andmed eraldi failidesse või andmebaasi tabelitesse, mis on mingil viisil omavahel taas-seotavad. Näiteks, selle asemel, et omada tabelit, kus on veergudeks nii ostu aeg, ostetud toote omadused (mitu veergu) ja kliendi andmed (mitu veergu), võib andmed paigutada kolme faili või andmebaasi tabelisse:

1) klientide andmete tabel, kus iga kliendi kohta on üks rida ja igale kliendile on määratud unikaalne ID;
2) toodete andmete tabel, kus iga toote kohta on üks rida ja igale tootele on määratud unikaalne ID;
3) ostude tabel, kus on kirjas ostu aeg, ostu koht, kliendi ID ja toote ID (ainult neli veergu).

Andmebaaside haldamiseks on andmeinseneril vaja teadmisi andmebaasidest ja programmeerimiskeeltest, käsurea kasutamisest jne. Kõrgemale tasemele jõudmiseks tuleb arvesse võtta muu hulgas ka seda, milliseid ülesandeid saab protsessorite või arvutite-serverite vahel ära jagada, et neid paralleelselt sooritada. Kuigi nende lahenduste loomine ja haldamine on üldiselt andmeinseneri töö, peavad edasiseks analüüsiks ka andmeteadlane ja analüütik neid lahendusi kasutada oskama.

Kui ettevõttes on peale andmeinseneri ka **andmearhitekt**, siis tema on see, kes defineerib, kuidas tuleb andmeid koguda, talletada ja jagada. Sellisel juhul tegeleb andmeinsener nende andmearhitekti kehtestatud nõuete rakendamisega ja loodud lahenduste haldamisega. **Andmeinsener ja andmearhitekt vastutavad tegevuse eest, mis tehakse enne andmete töötlemist ja kasutamist.**

Andmeanalüütik

Kui andmed on kogutud ja struktureeritud, saavad andmeanalüütik ja andmeteadlane oma ülesandeid täita. Alles andmeanalüüsiga alustavas ettevõttes on tõenäoliselt esmalt vaja ainult **andmeanalüütikut**. Kuigi, olenevalt ettevõttest, võib andmeanalüütikutel olla palju erinevaid ülesandeid ja vastutusalasid, mõeldakse andmeanalüütikust kui avastajast ja uurijast. Andmeanalüütiku kompetentsiprofiili kohta võite pikemalt lugeda Kutsekoja koostatud dokumendist (link).

Heal analüütikul on uudishimu ja oskused, et andmeid eri vaatenurkadest uurida. Selleks, et andmetest trende tuvastada, tegeleb ta ka andmete puhastamise ja mitmel viisil teisendamisega. Andmeanalüütik võib ettevõttele leida uusi võimalusi, mida uurida, või valdkondi, mille uurimiseks oleks tarvis rohkem andmeid koguda, et neist hiljem ettevõttele ka kasu oleks. Peale andmekaeve peab analüütik oskama ka oma tulemusi selgelt esitada ja raporteerida. Analüütik peaks oskama visualiseerida ja kasutada tööriistu, mis võimaldavad luua andmete interaktiivseid töölaudu. Sealjuures ei tohiks visualisatsioonid ja raportid olla arusaadavad ainult oma meeskonnakaaslastele, vaid ka inimestele väljastpoolt, näiteks valdkondade juhtidele ja turundustiimile.

Kui andmeanalüütik viib läbi analüüse ja koostab raporteid, siis temalt ei oodata tingimata, et ta oskaks leitud tulemusi tõlgendada ettevõtte tegevuse kontekstis. **Ärianalüütik** on andmeanalüütikute erijuht, kelle peamine eesmärk on andmete kasutamine, et vastata äriküsimustele ja anda ettevõttele soovitusi tuleviku tegevuse kohta. Ärianalüütikust võib mõelda ka kui andmeanalüütikust, kes on spetsialiseerunud äri valdkonnas. Ärianalüütik osaleb rohkem ka andmetest leitud trendide rakendamises äris ja võimalike strateegiate planeerimises.

Andmeteadlane

Andmeanalüütiku töö on anda edasi informatsiooni, mida andmed kajastavad, ja raporteerida nähtud faktidest, mitte statistilistest ebamäärasustest. Kui andmeanalüütik arendab tulemuslikkuse mõõdikuid, raporteerib olemasolevat ja edastab need tähelepanekud teistele, siis **andmeteadlane** keskendub pigem sellele, et need tähelepanekud oleksid ka statistiliselt olulised ehk mitte juhuslikud. Andmeteadlane peab olema võimeline andmeid vaatama samamoodi, nagu seda teeb andmeanalüütik, aga lisaks oskama rakendada statistilisi meetodeid, et eristada päris seost mürast, juhuslikkusest või peidetud probleemist andmetes. Ta peab ära tundma, milliseid küsimusi tasub edasi uurida ja kuidas neile küsimustele vastata, kogudes lisaandmeid või viies läbi erinevaid eksperimente. Tema ülesanne on mõista, kuidas toimub katseplaanide koostamine, ja sealjuures ära tunda sageli esinevaid vigu. Sellised

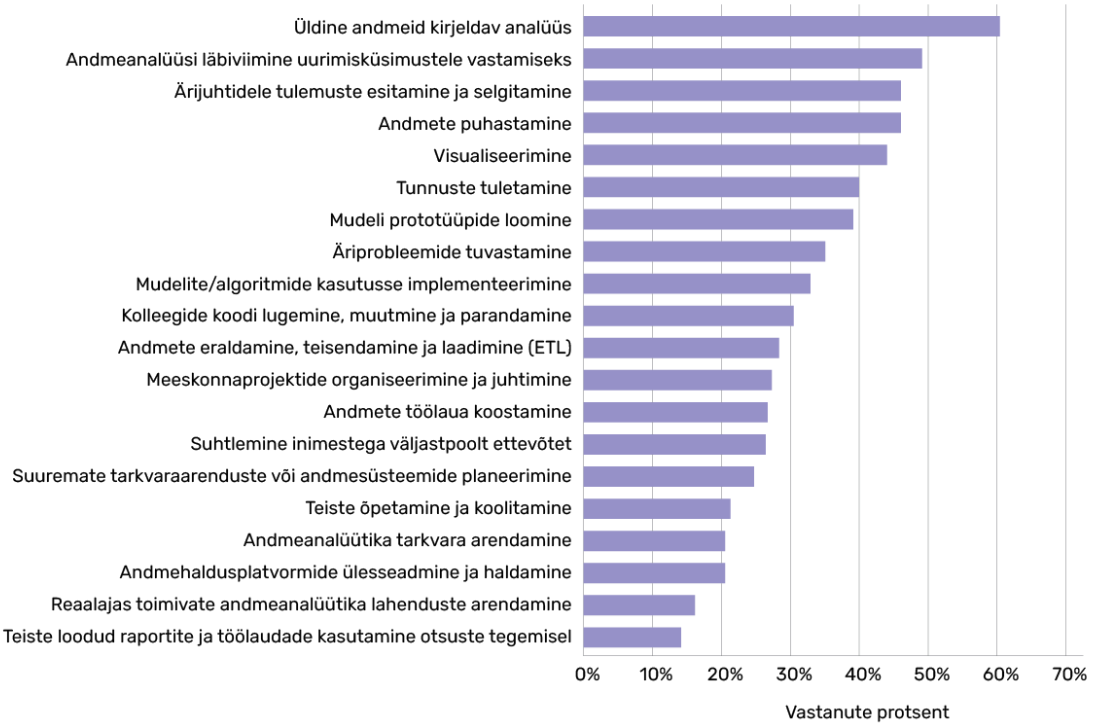
teadmised võimaldavad andmeteadlasel minna analüüsist samm edasi, lihtsast korrelatsioonist põhjuslike seosteni.

Üldiselt on andmeteadlastel oskus rakendada masinõppe meetodeid. Masinõpet saab kasutada koos teiste statistiliste meetoditega, et liikuda kirjeldavast analüüsist edasi ennustavasse analüüsi, kus ennustatakse mingeid tulevikusündmusi või tulemusi. Kuid masinõpe nõuab kasulike ennustuste tegemiseks sageli palju andmeid. Seega on masinõpet rakendava andmeteadlase üks olulisemaid oskusi ära tunda, millistel algoritmidel on suurim võimalus olla kasulik projektis, millega ta parajasti tegeleb.

Kui vaadata töökuulutusi, on selge, et tegelikult oodatakse andmeteadlaselt väga mitmekülgseid oskusi. Neist põhilised on näiteks

- programmeerimine;
- matemaatika ja statistika;
- visualiseerimine;
- andmete haldamine ja andmebaasid;
- probleemide ja tulemuste kommunikeerimine;
- andmete põhjal kasulike ennustuste tegemine (masinõpe).

Meediafirma O'Reilly viis 2017. aastal läbi uuringu, mille raames küsitleti 359 Euroopa andmeteadlast, kelle vastuste põhjal koostati nende tööülesannete, tehnoloogia ja palkade kohta ülevaateraport. Selles raportis toodi välja ka andmeteadlaste tööülesanded, mille oleme joonisel L.1 eesti keelde tõlkinud. On näha, et need ülesanded vastavad ka ülalmainitud oskusteootustele.



Joonis L.1. Andmeteadlaste tööülesanded. *Graafik loodud O'Reilly raporti alusel.*

Seega tundub, et andmeteadlase töö vastab korraga justkui mitmele töörollile: projektijuht, tarkvaraarendaja, andmeanalüütik, andmeinsener, masinõppe insener jne. Seda võib ühelt inimeselt olla väga palju nõutud ja sellega tuleks ka andmeteadlast palgates arvestada.

Andmeteaduse regulatsioonid ja eetika

Andmeteadus võib olla väga võimas tööriist lahendamaks mitmesuguseid probleeme, aga sellega võivad kaasneda ka oma ohud. Klientide-töötajate nimede, isikukoodide, kontaktandmete jms kogumise ja töötlemisega tuleb olla hoolikas, sest nende valedesse kätesse sattumine tekitab ohu eraelu puutumatusele, mis on põhiseaduslik õigus. Siiski pole isikuandmed ainult kontaktandmed, nimi ja muud inimesele mõistetavad tulbad tabelis. Näiteks ka inimeste Facebooki meeldib-nupu vajutuste andmed võivad olla isikuandmed, sest neist on andmeteaduse mudelitega võimalik küllaltki täpselt ennustada näiteks inimeste

- poliitilisi ja usulisi vaateid;
- perekonnaseisu, rasedust;
- seksuaalset sättumust;
- intelligentsust ja isiksuseomadusi;
- sõltuvust tekitavate ainete, nagu alkohol, narkootikumid ja sigaretid, tarvitamine.

Sellise info kätte saamise võimalus muudab ka selle andmestiku isikuandmeid sisaldavaks. Seega tekib küsimus, et **mis on isikuandmed ja mis pole**. Ka ekspertidel pole ühest vastust, kõik sõltub asjaoludest. Siiski on Andmekaitse Inspektsiooni (AKI) veebilehelt võimalik leida järgmised selgitused isikuandmete liikide kohta.

Definitsioonid
Tavalised isikuandmed on teave inimese ehk füüsilise isiku (andmesubjekti) kohta, millega saab teda **otse või kaudselt tuvastada**. Tuleb arvestada, et isikuandmed on ainult füüsilisel isikul – juriidilisel isikul puudub eraelu ja seega ka isikuandmed. Tavaliste isikuandmete alla kuuluvad näiteks nimi, isikukood, asukohateave, võrguidentifikaatorid (tunnused, mis sidevõrgus aitavad viia konkreetse isikuni), samuti füüsilised, füsioloogilised, geneetilised, vaimsed, majanduslikud, kultuurilised ning mistahes muud tuvastamist võimaldavad tunnused ja nende **kombinatsioonid**.

Eriliiki isikuandmete alla kuulub enamik selliseid andmeid, mis varasema sõnastuse järgi olid Eestis käsitletavad kui **delikaatsed isikuandmed**. Nendeks on isikuandmed, millest ilmneb rassiline või etniline päritolu, poliitilised vaated, usulised või filosoofilised veendumused või ametiühingusse kuulumine, füüsilise isiku kordumatuks tuvastamiseks kasutatavad biomeetrilised andmed (ennekõike sõrmejälje-, peopesajälje- ja silmaiirisekujutised), terviseandmed ning andmed füüsilise isiku seksuaalelu ja seksuaalse sättumuse kohta.

Tundlikud isikuandmed ei ole üldmääruses eraldi loetletud, kuid on määratletavad kui isiku privaatelule suuremat ohtu valmistavad andmed, mis ei kuulu eriliiki isikuandmete loetellu. Tundlikeks loetakse samuti neid andmed, mille avaldamisega kaasneb oht elule, tervisele, identiteedivargusele ning võib kaasneda varaline või mainekahju vms. Näiteks on tundlike andmetena käsitletav sotsiaalabi saamine, samuti kriminaal- ja väärteomenetlusega seotud andmed, makseteenustega seotud andmed pankades, krediitkaardi andmed, digitaalsed usaldusteenuse andmed digiallkirjastamiseks, mitteavalik teave inimese varandusliku seisu kohta, sõnumisaladusega kaetud sideandmed, reaalajas asukohatuvastuse andmed, krediidireiting jm profileerimine, millel on õiguslik tagajärg või oluline mõju. Paljudel juhtudel osutub tundlikuks inimese kohta käiv info, mis on avaliku teabe seaduse alusel juurdepääsupiiranguline teave.

Isikuandmete kogumine ja nendega töötamine on riiklikult ja Euroopa Liidu tasemel reguleeritud kõigile liikmesriikidele otse rakenduva üldmäärusega (ingl k *the EU general data protection regulation*, GDPR), mida järgnevas alapeatükis tutvustame. Selle määruse ja muude Eestis rakenduvate reeglite kohta saab põhjalikult lugeda Andmekaitse Inspektsiooni koostatud isikuandmete töötleja üldjuhendist (link), mis on ka suuresti siinse materjali aluseks.

Isikuandmete kaitse üldmäärus

Isikuandmete kaitse peamiseks juriidiliseks aluseks on põhiseaduslik õigus eraelu puutumatusele. Selle õiguse kaitse on tihti keerulisem kui mõne muu põhiõiguse kaitse.

- Kui võtame mõneks ajaks kelleltki ära liikumisvabaduse, saame selle hiljem täies mahus taastada. Privaatsust taastada on raske – see tähendaks, et privaatset infot teada saanud inimesed peavad selle käsu peale unustama.
- Ka suures mahus isikuandmete töötlemisel säilib probleem: neid analüüsitakse infotehnoloogiliste lahendustega, mis aga tavajuhul tähendab, et andmeid on lihtne kopeerida ja mitte-sihtotstarbeliselt kasutada. Kui andmed ei aegu, võib riive toimuda kaua aega pärast sihtotstarbelist analüüsi.

Seega on isikuandmete kaitsmine nii seaduslikult kui ka tehniliselt keeruline ülesanne. Seetõttu on Euroopa Liit ühiselt töötanud välja keskse regulatsiooni, mis kehtib kõigis liikmesriikides. Ka Eestis baseerub isikuandmete kaitsmine suuresti EL-ülesel **isikuandmete kaitse üldmäärusel** (IKÜM). On ka teisi, kohalikke seadusi, mis reguleerivad andmete kasutust näiteks riigiasutustes, finantsettevõtetes ja otseturunduses, aga peamised eraettevõtteid puudutavad põhimõtted pärinevad siiski IKÜM-ist. Seetõttu tutvustame selle määruse põhiideid siinkohal lähemalt.

Esmalt tuleks küsida, milliseid andmeid üldse kaitsma peab. Kõik andmed pole isikuandmed ja tavalisi andmeid kaitsma ei pea – ilma või aktsiahindade kohta võib iga firma talletada andmeid ükskõik mis mahus ja viisil ning neid vabalt avaldada. Lisaks isikuandmete kirjeldustest (vt ülal) välja jäävatele andmetele on veel juhte, kus andmekaitsemäärus ei rakendu.

Isikuandmete kaitse reeglid **ei kehti**, kui

1) andmeid kogutakse juriidilise isiku või asutuse kohta (ka juriidiline isik on isik, aga tema andmeid ei kaitsta);
2) teave ei võimalda inimest **mõistlike pingutustega**[67] tuvastada;
3) isikuandmeid ei töödelda automatiseeritult ja neist ei tehta ka füüsilist andmekogumit. Oma peas võib kõike meelde jätta. Kui aga juba luua firmasiseseks kasutuseks klientide andmetega vihik, on tegu struktureeritud andmekoguga ja tuleb IKÜM-i reegleid järgida;
4) andmeid kogutakse isiklikuks kasutuseks, näiteks pulmakülaliste nimekiri ja kontaktinfo. Seda nimekirja hiljem ettevõtte huvides kasutada ei tohi, sest see pole enam isiklik kasutus;
5) andmeid kasutatakse EL-i liikmesriigi või EL-i ühise julgeoleku eesmärkidel.

Kui isikuandmete kasutus ei mahu ühegi nimetatud reegli alla, aga firma soovib ikkagi isikuandmeid kasutada, peab selleks olema **õiguslik alus**.

1. On olemas **nõusolek**. Nõusolek peab olema spetsiifiline – mitte üldiselt isikuandmete töötlemiseks, vaid peab olema selgelt defineeritud töötlemise eesmärk. **Nõusolek ei tohi olla kohustuslik millegi muu saamiseks**, näiteks „soodustus kehtib ainult neile klientidele, kes annavad oma e-postiaadressi". Nõusolekuvormi vaikevalikud peavad olema need, mis kõige rohkem privaatsust tagavad.
2. Isikuandmete töötlust **on vaja lepingu sõlmimiseks või täitmiseks**.
3. Isikuandmete töötlust **on vaja seaduste täitmiseks**, näiteks raamatupidamises arvete kohta aruannete esitamiseks, maksude maksmiseks.
4. Erijuhud avaliku sektori jaoks. Soovi korral vaadake isikuandmete töötleja üldjuhendist (link).

Seega on andmeid koguda ja töödelda lubatud küll, aga see ei tohi toimuda lihtsalt niisama, ilma põhjuseta. Järgida tuleb **eesmärgipärasuse ja minimaalsuse printsiipe**. Andmeid tohib koguda ainult mingi selgelt defineeritud **eesmärgi** saavutamiseks, mida ei ole muul viisil võimalik saavutada. See eesmärk peab tulenema firma põhitegevusest. Koguda tohib ainult **minimaalne** hulk andmeid, mis seda eesmärki saavutada lubavad.

Ka oma isikuandmete töötlemiseks **nõusolekut** andes jääb isikuandmete omanikuks alati ikkagi isik ise. Riigiasutus või ettevõte võib omada õigust tema andmeid **töödelda** ehk koguda ja kasutada[68]. Andmete omanikuna on inimesel loogiliselt üsna palju õigusi nendega seoses. Üldjuhul, v.a teatud riigiasutuste andmed, on inimesel õigus

1) olla unustatud – isikul on võimalik nõuda, et ettevõte kustutaks kõik tema andmed;
2) oma andmetele ligi pääseda – isikuandmed peavad olema isikutele ligipääsetavad ja neid peab olema võimalik isikule väljastada kergesti loetaval kujul;

[67] Mis on mõistlik pingutus, on tihti vaieldav. See, mis on firma tavatöötaja jaoks suur pingutus, võib kogenud andmeteadlase jaoks olla mõne minuti töö. Näiteid selle reegli toimimise kohta leiate ka isikuandmete töötleja üldjuhendist. Selle punkti täitmise eesmärgil rakendatakse ka andmete anonüümimist.

[68] **Definitsioon: isikuandmete töötlemine** on andmetega tehtav mistahes toiming – kogumine, korrastamine, säilitamine, muutmine, lugemine, kasutamine, edastamine, ühendamine, kustutamine jne.

3) nõuda ebaõigete andmete parandamist – andmetöötlejal on kohustus parandada ebaõiged isikuandmed ning tagada andmete kvaliteet ja õigsus;
4) oma andmete kasutamist piirata – isikul on õigus nõuda, et tema andmeid mingitel juhtudel ei kasutataks, aga need võivad jääda ettevõtte andmebaasi;
5) andmeid üle kanda – inimesel on õigus nõuda, et teda puudutavad andmed edastatakse näiteks konkureerivale ettevõttele. Seejuures tuleb seda teha loetavas formaadis.

Kui ettevõte töötleb ja kasutab isikuandmeid, peab ta nende õigustega arvestama. Kui klient soovib oma andmeid kustutada, ise näha või teisele ettevõttele edastada, siis satub firma, mis on andmeid küll kogunud, aga kaootiliselt organiseerinud, hätta. Sel juhul on vaja eraldi vaeva näha, et kõik isikut puudutavad andmed eri failidest kokku koguda, Exceli kommentaarid eraldi veergudeks teha jne. Nii unustamise, parandamise kui ka päringutele vastamise võimaldamiseks on mõistlik, et kui andmeid juba koguma hakatakse, siis korrastatult – nii jõuame tagasi andmete kogumise ja korraldamise hea tava olulisuse juurde.

Andmete anonüümimine

Et töödelda andmeid ilma kellegi privaatsust rikkumata, kasutatakse **anonüümimis-, pseudonüümimis- ja krüptimismeetodeid**. Üks eesmärke võib olla täita ülal mainitud tingimust, et isikuandmete kaitse reeglid ei kehti, kui teave ei võimalda inimest **mõistlike pingutustega** tuvastada (vt IKÜM-i rakendusala nimekirja ülal). Tõesti, täielikult anonüümseid andmeid ei ole vaja kaitsta. Siiski on anonüümimise ja pseudonüümimise puhul tihti tegu pigem turvameetmega, et isikuandmed valedesse kätesse ei jõuaks ja ettevõte menetluse alla ei satuks. See tähendab, et nende meetodite rakendamine ei pruugi alati täielikult vabastada IKÜM-i reeglite täitmisest, vaid pigem vähendab riske.

Definitsioonid
Pseudonüümimine tähendab äratuntavate isikuandmete asendamist varjunimede, numbrikoodide ja muude tunnustega, mida asjassepuutumatud isikud ei oska ära arvata.

Krüptimine tähendab edastatava või hoitava teabe sisu kättesaamatuks muutmist. Lihtsaim viis selleks on ID-kaardiga krüptimine, krüptitud faile saavad avada ainult kindla isikukoodiga ID-kaardi valdajad.

Anonüümimine tähendab, et teabest kaotatakse kõik jäljed, mis võiksid viia tuvastatavate isikuteni. Kui pseudonüümimine ja krüptimine on tagasipööratav umbisikustamine, siis anonüümimine tähendab tagasipööramatut ehk lõplikku umbisikustamist.

Deanonüümimine on mitte piisaval määral anonüümitud andmete uuesti isikutega sidumine, näiteks kasutades teisi sarnaseid andmestike, mis on isikustatud ja mille võrdlemine anonüümitud andmetega võimaldab isikuid taas-identifitseerida.

Selle asemel, et kasutada andmetabelites inimeste identifitseerimiseks nime ja isikukoodi, saab igale inimesele määrata **pseudonüümitud identifikaatori**, näiteks formaadis „klient_143“. Kuskil võib olla olemas ka tabel, mis seob identifikaatori päris nime ja isikukoodiga, aga seda tabelit saab eraldi kaitsta ning sellele ligipääsu piirata seda mingi kindla isiku valdusesse andes või IT-lahendusi kasutades. IKÜM-i reeglites on muu hulgas ka kirjas, et isikuandmetele ligipääs peab olema **logitud** – igast vaatamisest peab jääma logisse märge. Seega identifikaatoreid isikuandmetega siduva tabeli kasutamist tuleb logida. Kui ülejäänud kogutud andmete tabel aga sisaldab ainult identifikaatorit, siis selle kasutus on vähemalt teoreetiliselt logimisest vaba.

Vahemärkus
Isikuandmete töötlemist on vaja **logida** – igast andmete vaatamisest peab jääma märge. Keegi ei tohi pääseda andmetele ligi omakasupüüdlikel või pahatahtlikel eesmärkidel ega pelgast uudishimust. Seega ei saa isikuandmeid hoida lihtsalt tavalises Exceli tabelis, kuhu ei jää jälge, kui töötaja seda vaatab. Isikuandmeid omav firma peab rakendama asjakohaseid tehnilisi ja korralduslikke meetmeid, et logimine tagada. Samuti lasub ettevõttel kohustus tuvastada rikkumised ja neist teavitada.

Kahjuks säilib alati oht, et keegi suudab pseudonüümid lahti mõtestada ja isikutega taas-siduda, seega on mõistlik ka pseudonüümitud tabelit kaitsta. Tehniliselt saavutab pseudonüümimine tihti lihtsalt selle, et **andmetes ühe indiviidi tuvastamine on keerukam, aga mitte võimatu**. Näiteks, sidudes pseudonüümitud andmestiku mõne muu andmestikuga, võib nende koosmõjul olla isik taas tuvastatav. Vahel võib piisata avalikult saada olevast lisateabest (vt allolevaid näiteid 1–2). Seaduses pole selgelt defineeritud, kelle jaoks ja kui suur peab olema pingutus inimese tuvastamiseks andmetest, et andmeid saaks tõesti anonüümseks lugeda ja need vabaneksid IKÜM-i nõuetest. See võib tekitada juriidilisi vaidlusi. Seetõttu ongi targem käsitleda ka pseudonüümsete identifikaatoritega andmestikke isikuandmetena ja neid vastavalt kaitsta.

Näide 1
Inimeste liikumisi (nt asukohti mobiilimasti täpsusega) sisaldav andmestik võib olla töödeldud nii, et selles ei sisaldu isiku nimi, telefoninumber ega muu isikut defineeriv teave, vaid ainult isikule omistatud identifikaator ja asukohad, kus ta viibinud on. Ka neis kohtades viibimise ajad võivad olla täielikult andmetest eemaldatud. Siiski näitab sellist tüüpi andmetel tehtud uuring, et 95% inimestest on võimalik teistest eristada nelja asukoha põhjal, mida nad mingi aja jooksul külastanud on. Seega, kui me teame isiku aadressi, töökohta, maakodu asukohta ja näiteks spordiklubi, mida ta külastab, siis suudame ta sellest andmestikust väga tõenäoliselt üles leida. Seeläbi saame teada ka kõik ülejäänud kohad, kus ta viibinud on, ja rikume tema eraelu puutumatust.

Näide 2
Samuti on tähelepanuväärne näiliku anonüümsuse näide see, et andmestikus, mis ei sisalda nime ega isikukoodi, aga sisaldab sugu, postiindeksit ja sünniaega, on võimalik väga suure tõenäosusega inimene unikaalselt tuvastada (USA andmetel põhinevas näites 87% juhtudest). See tähendab, et te võite sellest andmestikust üles leida oma sõbra, kelle sünniaega ja postiindeksit te teate, ning seeläbi näha ka kõiki teisi temaga seotud andmeid, mis selles andmestikus on.

On raske öelda, kas näidetes toodud juhtudel oli tegu rohkema kui „mõistliku pingutusega“ või peaks neid andmestikke kaitsma kui isikuandmeid. Mõistliku pingutuse suurus sõltub muu hulgas ka otsija osavusest (kas ta oskab otsingut automatiseerida), nii et sellise andmestiku osavale andmeteadlasele andmine on isikuandmete lekitamine, tavainimesele andmine aga mitte. Igal juhul võib öelda, et sellised andmestikud pole täielikult anonüümsed, neid peaks nimetama pseudonüümseteks. Neid sõnu kasutatakse tavakeeles kahjuks tihti sünonüümidena ja see võib tekitada segadust, kas IKÜM-i nõuded rakenduvad.

Täielik anonüümsus on tehniliselt keeruline ja pigem teoreetiline eesmärk. Võtame näiteks Tartu rattaringluse andmed, mis oma originaalkujul sisaldavad kliendi andmeid ja aegrida iga kliendi tehtud rattasõitudest. Asendades nimed pseudonüümidega, võib neist andmetest teatud inimese ikkagi tuvastada, kui teame, kust kuhu ja mis kellaaegadel ta tavaliselt sõidab. Täielikult anonüümseks saab andmestik alles siis, kui me lõikame aegread katki ja kaotame info sama kliendi tehtud sõitudest. Nii jääb meile alles ainult info, kust kuhu sõideti, ja mitte mingit märget, kes sõitis. Sellised andmed on anonüümsed, IKÜM-i reeglitest vabad, aga ikkagi kasulikud – näiteks uurimaks linnakodanike liikumistrajektoore ja -aegu ning planeerimaks rataste paigutust peatustesse. Samas ei võimalda see enam tundma õppida kliente.

Lõppkokkuvõttes on üsna keeruline andmeid täielikult anonüümida, sest selle saavutamiseks on tihti vaja viia nad kujule, mis pole enam teatud analüüsideks kasulik. Targem on isikuandmeid lihtsalt kaitsta, piirates neile ligipääsu, logides nende kasutamist ja järgides kõiki muid IKÜM-is ette kirjutatud reegleid. Veel parem on võimaluse korral koguda ainult tavaandmeid ning neid mitte üldse isikustada – võibolla polegi ostuajaloo uurimiseks vaja teada kliendi nime, aadressi ja paljut muud. Ehk piisab lihtsalt kliendikaardi jälgimisest inimese jälgimise asemel?

Privaatsuse riivamise riski vähendamisele tuleb mõelda kogu projekti jooksul, meie kursuse mõistes kõigil CRISP-DM-i etappidel. See on oluline firma mainele ja ka sellepärast, et trahvid suurte lekete korral võivad olla märkimisväärsed.

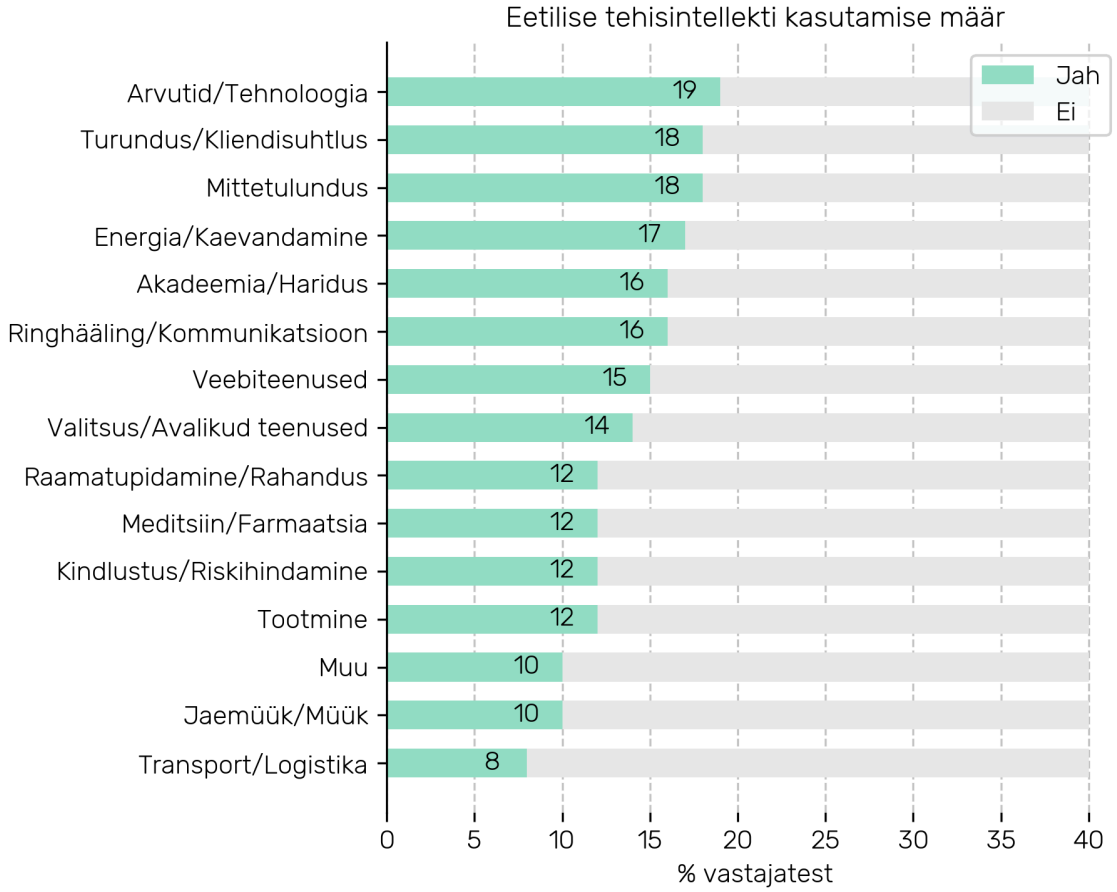
Riske võib vähendada näiteks järgmistel viisidel.

1. Vähem andmeid kogudes. Näiteks, mitte koguda klientide kontaktandmeid, mitte talletada hoones sees uksekaartidega uste avamise andmeid. Ka info regulaarne kustutamine vähendab lekke korral tekkivat kahju.

2. Rakendades meetodeid, mis vähendavad infotöötlemise vajadust. Kui saame andmetest vajaliku vastuse vähem ja väiksemaid andmestikke kombineerides, spetsiifilisemalt pärides, peame töötlema vähem infot vähemate isikute kohta.
3. Andmeid ebatäpsemalt talletades. Aadressi asemel võib piisata linnaosast. Sünnipäeva asemel võib piisata vanusevahemikust. Näiteks „25–30-aastane mees Nõmmelt“ on piisav täpsus paljude analüüside jaoks, aga ei võimalda isikut identifitseerida.

Eetika

Andmeteaduse arenguga seotud eetilised probleemid on sarnased nendega, mis on üldiselt seotud tehnoloogia kasutamise ja arendamisega. See tähendab, et tehnoloogia kasutamine on eetiline, kui selle kasutamise eesmärk on eetiline, ja mitte-eetiline, kui selle eesmärk on ebaeetiline. Kui juturoboteid kasutatakse, et levitada vihakõnet, siis on tegu ebaeetilise kasutuse, mitte ebaeetilise tehnoloogiaga. Seega ei ole olemas andmeteaduse-spetsiifilist reeglit, mis eetikat kuidagi eraldi reguleeriks. Seetõttu ei proovi me siin õpetada, mis on eetiline, vaid lihtsalt tuua näiteid lahendustest, mille eetilisus on kaheldav.



Joonis L.2. Eetiline tehisintellekt. *Jaatavate vastuste määr masinõppe kasutajate hulgas vastates küsimusele, kas kasutate vastutustundliku või eetilise tehisintellektiga seotud tooteid või prakitkaid. Eetilise tehisintellekti tehnoloogia rakendamine polnud 2022. aasta seisuga veel väga levinud. Lähtekood.* [69]

[69] Info allikas: Kaggle 2022 survey, litsents: Apache 2.0).

Diskrimineerimise oht andmeteaduses
See, et andmeteaduslikud mudelid on loodud erapooletute algoritmide poolt, ei tähenda, et mudelite tulemused, näiteks nende ennustused, oleksid alati õiged või õiglased. Kui mudel on loodud, kasutades andmeid, mis peidavad endas diskrimineerimist, õpib ka mudel diskrimineerima.

Näide 1
Pank tahab asendada laenu andmise otsust tegevad inimesed masinõppe mudeliga, mis treenitakse imiteerima nende samade inimeste otsuseid (sisend: info inimese kohta, märgend: jah/ei anname laenu). Sel juhul õpib mudel ära mitte ainult üldise otsuste tegemise loogika, vaid ka kõik seniste otsustajate kallutatused. Näiteks, et välismaalastele laenu pigem ei anta, kuigi põhjust polegi. Seega pole andmepõhine lähenemine vaikimisi kuidagi ausam või õiglasem. Et mudel ei hakkaks kedagi diskrimineerima, tuleb analüüsi hoolikalt planeerida ja treeningandmete kvaliteeti hinnata.

Näide 2
Ebatäielikel andmetel loodud mudel võib viia diskrimineerimiseni. Kui kindlustusfirmadel oleks ligipääs vastavatele andmetele ja neil oleks lubatud arvesse võtta iga isiku detailseid terviseandmeid, rahvust või rassi, võiks see viia mõne isiku jaoks ebaausalt kõrge kindlustusmakseni. Mudel suudaks õppida näiteks, et maksahaiguste diagnoosiga inimesed on keskmiselt riskantsemad liiklus- ja kodukindlustuse kliendid, ning nõuaks neilt kõrgemat makset. Selliselt loodud mudeli sisendid pole aga ilmselt täielikud ega erista alkoholismist põhjustatud maksahaigusi teistest maksahaigustest. Sellist mudelit kasutades võivad paljud tegelikult vähese riskiga inimesed ebaausalt kannatada. Sama juhtub rahvust või rassi arvesse võttes – indiviid saab kõrgema või madalama makse, olenemata tema tegelikust käitumisest, sest seda mudel arvesse võtta ei suuda. Lisaks, kuna andmed võivad olla valed või mürased, tehtaks vahel kindlustusmakse suuruse otsus hoopis valedel alustel.

Samuti võib eetilisuse küsimus tekkida „piiri pidamise“ kontekstis. Mis hetkel on kasumi optimeerimine olulisem näiteks klientide tervisest? Kui ristmüügi käigus soovitatakse ebatervislikke toiduained (nt maiustusi), siis kas see on eetiline? Jah, inimene tihti klõpsab sellel šokolaadil ja toob poele kasumit. Õnneks on alkoholi ja tubakareklaam eraldi reguleeritud ning võib loota, et e-toidupood ei paku koolaga kaasa rummi või närimiskummiga sigarette, kuigi soovitussüsteem on tuvastanud, et neid ostetakse tihti koos. Eetiliselt eriti kaheldavaks muutub näiteks kaloririkaste toodete pakkumine juhul, kui pakkumised on isikustatud. Sel juhul võib soovitussüsteem võimendada juba olemasolevat halba harjumust, seda teadlikult teatud kliendile meelde tuletades (ja mitte suvalistele teistele klientidele). Kindlasti ei tähenda see, et poed peaksid ristmüügi

lõpetama, aga ehk oleks võimalik klientidele anda suurem valikuvabadus ja kontroll neile soovitatavate toodete üle.

Viga esindamises
Masinõppe süsteem teeb vea esindamises (ingl k *representational harm*), kui ta võimendab või peegeldab negatiivset stereotüüpi teatud rühmade kohta. Näiteks, 2015. aastal klassifitseerisid Google Photos teenuse algoritmid mustanahalisi inimesi kui gorillasid. Loomuliku teksti genereerimise algoritmid võimendavad neid stereotüüpe, mis olid väljendatud andmestikes, mille peal neid treeniti.

Toote täpsuse ebaproportsionaalsus
Toote täpsuse ebaproportsionaalsus (ingl k *disproportionate product failure*) tähendab, et mudeli tõhusus on ebaühtlane sõltuvalt sellest, kes teenust kasutab. Andmeteadusel põhinevad teenused kipuvad kõige täpsemaid ja kasulikumaid tulemusi andma ainult teatud grupile ühiskonnast.

Näide
Näotuvastustehnoloogiad, sealhulgas telefoni avamiseks, töötasid oma esimeses versioonis kõige paremini valgete keskealiste meeste jaoks. Põhjuseks oli see, et andmestik, mille abiga vastav lahendus loodi, sisaldas kõige rohkem just selle ühiskonnagrupi liikmete näopilte. See aga tähendas, et noore tüdruku või mustanahalise vanatädi õunamärgiga telefon ei avanenud tema nägu nähes või veelgi hullem – avanes teiste samasse gruppi kuuluvate inimeste nägude peale. Sama lugu on ka kõnetuvastuse täpsusega – see sõltub hääle tämbrist ja aktsendist. Lõppkokkuvõttes viib selline erisus selleni, et erinevatel ühiskonna- ja demograafilistel gruppidel on erinev ligipääs tehnoloogiale. Õnneks võtavad suurfirmad ja riigid probleemi tõsiselt ning näiteks ülal mainitud näotuvastuslahendus treeniti uuesti, kasutades seekord juba erinevamaid näopilte sisaldavat andmestikku.

.

Enesekontrolli küsimuste vastused

Peatükk 1

1) **Milline järgmistest väidetest on tõene andmete kohta?**
   B) Digiteeritud andmed võivad sisaldada pilte, tekste ja helifaile.

2) **Andmeteaduse peamine eesmärk on andmete põhjal _____.**
   C) paremate otsuste tegemine

3) **Miks on andmete kvaliteet andmeteaduse protsessis oluline ja kuidas "prügi sisse, prügi välja" (garbage in, garbage out) põhimõte seda selgitab? Milliseid samme saab astuda, et vältida andmete kvaliteediprobleeme?**
   Andmete kvaliteet on oluline, sest vigased või puudulikud andmed viivad ebausaldusväärsete analüüsitulemusteni, mida illustreerib põhimõte "prügi sisse, prügi välja".

   Selle vältimiseks tuleb võtta järgmised sammud:
   - Koguda andmeid hoolikalt ja struktureeritult
   - Kontrollida andmete täpsust ja täielikkust.
   - Puhastada vigased või puudulikud väärtused.

Peatükk 2

1) **Järjesta CRISP-DM-i sammud**. Õige järjekord: d, a, b, c

2) **Vii kokku tegevus ja CRISP-DM-i samm**. Õige vastavus: a-h, b-f, c-e, d-g

3) **Lünktekst**: *CRISP-DM* on valdkondadeülene andmekaeve standardprotsess.

4) **Andmete tulemuste hindamise etapis peavad osalema**
   C) nii andmeteadlane kui ka äripool

5) **Andmete kvaliteedi kontrollimiseks tuleb kindlaks teha,**
   a) kas andmed on täielikud
   b) kas andmed on õiged
   c) kuidas tähistatakse andmetes esinevaid puuduvaid väärtusi

6) **Mida järgnevatest tehakse CRISP-DM-i andmete ettevalmistamise etapil?**
   c) andmete puhastamine
   d) mudeli sisendiks sobiva tabeli koostamine

Peatükk 3

1) **Mis on "tunnus" andmeteaduse kontekstis?**
   b) Andmetabeli veerg, mis esindab mõõdetud omadust.

2) **Mida tähendab "esinduslik andmestik"?**
   B) Andmestik, mille read on juhuslikult valitud ja peegeldavad päriselus leiduvaid proportsioone.

3) **Miks peaks ühe andmetabeli veerg sisaldama ainult ühte tüüpi infot? Too näide, kuidas seda reeglit rikkudes võib tekkida probleem.**
   Kui veerg sisaldab korraga kogust ja ühikut (nt "1,5 kg"), ei saa seda kasutada matemaatiliste arvutuste jaoks, sest väärtusi pole võimalik otse võrrelda.

4) **Kuidas tuleks Excelis puuduvad väärtused tähistada, et tarkvara saaks neid õigesti tõlgendada?**
   Siin on mõned levinud meetodid:
   - **Tühjad lahtrid.** Paljud analüüsitarkvarad tõlgendavad tühje lahtrid automaatselt puuduvate väärtustena.
   - **"NA" või "NaN".** Sisestage puuduvate väärtuste asemele "NA" (inglise keeles *Not Available* või *not a number*).

   Oluline on **vältida arvuliste väärtuste** nagu "0", "-999" või "9999" kasutamist puuduvate väärtuste tähistamiseks, kuna need võivad analüüsis ekslikult käsitletud saada tegelike mõõtmistulemustena ja moonutada tulemusi. Ühtse ja läbimõeldud puuduvate väärtuste märkimisega tagate andmete kvaliteedi ning usaldusväärsed analüüsi tulemused.

5) **Kirjelda ühte meetodit anomaaliate tuvastamiseks andmetes.**
   Anomaaliate ehk erandlike andmepunktide tuvastamiseks on mitmeid meetodeid. Siin on mõned neist:
   Karpdiagrammi kasutamine:

   - Karpdiagramm visualiseerib andmete jaotust, näidates mediaani, kvartiile ja võimalikke erindeid. Joonistage karpdiagramm huvipakkuva tunnuse kohta. Erandid ilmuvad punktidena, mis asuvad karpdiagrammi "vurrudest" väljaspool.

   Hajuvusdiagrammi kasutamine:

   - Kujutage andmete kahe tunnuse vaheline seos punktidena graafikul. Tuvastage punktid, mis asuvad selgelt eemal üldisest mustrist või klastrist. Võimaldab tuvastada anomaaliad, mis seisnevad kahe tunnuse haruldases kombinatsioonis.

   Valitud meetod peaks sõltuma andmete tüübist, jaotusest ja analüüsi eesmärgist.

Peatükk 4

1) **Andmete kogumisele ja ettevalmistamisele kulub väga palju aega. Mis võib olla selle põhjus?**

- Andmete ebatäiuslikkus: andmetes võivad olla puuduvad väärtused, duplikaadid või vead, mis vajavad käsitsi puhastamist ja korrastamist.
- Erinevad andmeallikad: andmeid võib tulla mitmest allikast (nt andmebaasid, CSV-failid, välised API-d) ning need tuleb esmalt integreerida ja ühildada, mis on ajamahukas.
- Andmete vormingute erinevus: erinevad andmeformaadid (struktureeritud ja struktureerimata andmed) nõuavad teisendamist, et neid saaks mudelis kasutada.
- Andmete mitmekesisus: suur andmete mitmekesisus (nt tekst, pildid, arvud) nõuab erinevaid töötlusmeetodeid ja tehnoloogiat, mis teeb ettevalmistamise aeganõudvaks.
- Äriprobleemile sobivuse tagamine: Andmeid tuleb pidevalt kontrollida ja kohandada, et need vastaksid lahendatavale äriprobleemile.

2) **Tooge uusi näiteid (lisaks õpikus mainitule) andmete valimise kohta. Tooge näide juhust, kui me otsustame valida mitte kõik tabeli andmeread, kui me otsustame valida mitte kõik tabeli veerud, kui me otsustame kasutada ainult osa saadaval olevatest tabelitest.**
   Näide, kui otsustame valida mitte kõik tabeli andmeread:
   kui analüüsitakse kliendikäitumist teatud ajavahemiku jooksul (nt viimasel kuuel kuul), valitakse ainult selle perioodi andmeread ja varasemad andmed jäetakse välja, et analüüs keskenduks hetkel olulisele trendile.

   Näide, kui otsustame valida mitte kõik tabeli veerud:
   kui analüüsime klienditehinguid ja soovime keskenduda ainult tehingu maksumusele ja kuupäevale, võime jätta välja veerud nagu „kliendi nimi“ ja „kliendi vanus“, mis pole selle analüüsi jaoks olulised.

   Näide, kui otsustame kasutada ainult osa saadaval olevatest tabelitest:
   ettevõtte andmebaasis võib olla mitu seotud tabelit (nt kliendid, tellimused, tooted), aga kui keskendutakse ainult toodete populaarsuse analüüsile, võidakse kasutada ainult tooteid ja tellimusi puudutavaid tabeleid, jättes klienditabelid välja.

3) **Uute tunnuste loomine: millised kasulikke tunnuseid saab tuletada klienditeenindusse tehtud kõnede helisalvestuste põhjal? Milliseid teisi tööriistu saab selleks kasutada? Kas mingitel juhtudel võib olla otstarbekas kulutada aega ise kõnede läbi kuulamisele ja märkmete tegemisele?**
   Tunnuste loomine helisalvestuste põhjal:
   - kõne kestus: kõne pikkuse mõõtmine, mis võib anda teavet probleemi keerukuse või kliendirahulolu kohta (lühikesed kõned viitavad rahulolevatele klientidele, pikemad keerulistele probleemidele);
   - emotsioonide tuvastamine: kõne toonianalüüs, kus tuletatakse, kas kõne oli positiivne, negatiivne või neutraalne, kasutades kõnetuvastustehnoloogiat;

- sagedased sõnad või fraasid: kõnelemiskõne analüüs, mis tuvastab, millised märksõnad või fraasid esinevad sageli, et leida korduvaid probleeme või kliendi vajadusi.
- Tööriistad tunnuste loomiseks: NLP tööriistad, Sentiment Analysis, Speech-to-text tarkvara

Kas ise kõnede kuulamine on otstarbekas? Kui kõnesid on vähe ja nende sisu on keeruline või oleneb kontekstist (nt väga tehnilised või delikaatsed kõned), võib olla mõistlik kuulata kõnesid käsitsi, et paremini mõista kliendiprobleeme ja konteksti.

4) **Miks on andmete standardimine või skaleerimine oluline ja milliseid meetodeid selleks kasutatakse? Tooge näide mudelitüübist, mille puhul andmete mitteskaleerimine tekitaks probleeme.**
Andmete standardimine ja skaleerimine on oluline, et tunnuste väärtused oleksid samas suurusjärgus, mis aitab vältida teatud tunnuste domineerimist mudelite treenimisel. Tavaliselt kasutatakse standardimist (tunnuste viimine normaaljaotusele) või min-max-skaleerimist (väärtuste viimine lõiku [0; 1]). Näiteks lähimate naabrite meetod ja klasterdamine põhinevad kauguse mõõdu arvutamisel, mida sama protsendi suurused muutused väikestes arvudes ja suurtes arvudes mõjutavad erinevalt.

Peatükk 5

1) **Mida tähendab George Boxi kuulus tsitaat „Kõik mudelid on valed, aga mõned on kasulikud“? Mis mõttes on mudelid valed? Kuidas saab midagi valet olla kasulik?**
Boxi tsitaat rõhutab, et mudelid on lihtsustused ja seetõttu paratamatult mingil määral ebatäpsed, kuid nende kasulikkus seisneb oluliste omaduste säilitamises uurimisküsimuse jaoks.

2) **Selgitage hii-ruut testi kasutamise eesmärki ja põhimõtet. Millistel andmetel seda kasutatakse? Mis on selle testi nullhüpotees?**
Hii-ruut testi kasutatakse kategooriliste muutujate vahelise seose uurimiseks, võrreldes vaadeldud sagedusi teoreetiliste sagedustega, et hinnata, kas täheldatud erinevused on statistiliselt olulised.

3) **Kirjeldage ANOVA eesmärki ja põhimõtet. Millised on ANOVA eelised ja piirangud võrreldes t-testiga?**
ANOVA eesmärk on võrrelda mitme rühma keskväärtusi, et teha kindlaks, kas nende vahel on statistiliselt olulisi erinevusi. ANOVA eeliseks on, et see võimaldab võrrelda rohkem kui kahte rühma korraga, vältides mitme t-testi tegemisega seotud vea suurenemist. Piiranguks on aga vajadus eeldada normaaljaotust ja rühmade homogeenset dispersiooni.

4) **Millised on juhendamata masinõppe peamised ülesanded ja kuidas need erinevad juhendatud masinõppe ülesannetest?**
Peamised juhendamata masinõppe ülesanded on klasterdamine, assotsiatsioonireeglite leidmine ja anomaaliate tuvastamine, mis keskenduvad andmetest mustrite ja struktuuride avastamisele ilma eelneva märgendita. Juhendatud masinõpe keskendub märgistatud andmetega seoste õppimisele ja ennustuste tegemisele.

5) **Oletame, et olete ettevõtte andmeteadlane ja teie ülesanne on analüüsida klientide ostukäitumist, et leida mustreid ja teha ennustusi tulevaste ostude kohta. Tooge näited, milliste küsimuste puhul kasutaksite juhendamata ja milliste puhul juhendatud masinõpet. Kirjeldage, milliseid meetodeid täpselt kasutaksite ja miks.**
Kui eesmärk on leida mustreid klientide ostukäitumises ilma eelteadmisteta kliendigruppide kohta, kasutaksin juhendamata masinõpet. Näiteks võiksin kasutada klasterdamist (*k*-keskmised, hierarhiline klasterdamine), et jaotada kliendid nende ostukäitumise järgi segmentidesse. See aitab mõista, millised kliendid on sarnased ja kuidas neid paremini sihtida turunduskampaaniatega.

Kui eesmärk on teha konkreetseid ennustusi, näiteks ennustada, millised kliendid tõenäoliselt teevad järgmise kuu jooksul suure ostu, kasutaksin juhendatud masinõpet. Sel juhul võiksin rakendada logistilist regressiooni või otsustusmetsa meetodit, kasutades ajaloolisi ostuandmeid ja kliendiomadusi (nt vanus, sissetulek, varasemad ostud) märgenditena, et treenida mudelit tulevaste ostude ennustamiseks.

6) **Miks on oluline, et ansambelmudelis kasutatavad mudelid oleksid üksteisest erinevad? Tooge näide ansambelmeetodi rakendamisest ja analüüsige, kuidas mudelite mitmekesisus võib mõjutada lõpptulemuse täpsust.**
Ansambelmeetodid masinõppes põhinevad mitme mudeli kombineerimisel, et parandada ennustuste täpsust ja üldistust. Ansamblis kasutatavad mudelid peavad üksteisest erinema, sest erinevate mudelite kombineerimine aitab vähendada iga üksiku mudeli spetsiifilisi vigu ja suurendab mudelite mitmekesisuse kaudu üldist täpsust. Näiteks võib *bagging*-meetod (nt otsustusmets) kombineerida mitu otsustuspuud, mis on treenitud erinevate andmealamhulkadega, et vähendada üleõppimise riski ja suurendada täpsust. Kui kõik mudelid oleksid sarnased, ei tuleneks nende kombineerimisest mingit kasu.

7) **Selgitage, miks on otsustusmets võimsam mudel kui lineaarne regressioonimudel. Milliseid keerulisemaid ülesandeid suudab otsustusmets lahendada, mida lineaarne mudel ei suuda? Tooge näiteid olukordadest, kus tuleks eelistada otsustusmetsa, ja põhjendage oma vastust.**
Erinevalt lineaarsest regressioonist suudab otsustusmets jäädvustada keerukamaid andmemustreid ja mittelineaarseid seoseid. Lineaarne mudel eeldab, et sõltuv muutuja ja sõltumatud muutujad on omavahel lineaarsetes

suhetes, mis piirab mudeli võimekust lahendada keerulisi ülesandeid. Otsustusmets koosneb mitmest otsustuspuust, mis koosnevad suurest hulgast hargnemispunktidest. See võimaldab õppida keerulisi ja mittelineaarseid seoseid andmetes. Otsustusmets suudab leida suure hulga tunnuste hulgast informatiivsed tunnused ja on mürale vähem tundlik.

8) **Oletame, et treenisite masinõppe mudelit andmestikul, mis sisaldab klientide käitumisandmeid ja nende ostude ajalugu, et ennustada tulevasi oste. Treenimisetapis saavutas mudel väga suure täpsuse, aga kui hakkasite seda reaalses kasutuses rakendama, ei andnud see enam häid ennustusi. Selgitage, mis võiks olla selle probleemi põhjuseks.**
Üks põhjus, miks mudel ei andnud reaalses kasutuses häid ennustusi, võib olla üleõppimine ehk ülesobitumine. See tähendab, et mudel õppis liiga hästi ära treeningandmetele spetsiifilised mustrid, kuid ei suuda üldistuda uutele andmetele, mida see treeningu ajal ei näinud. Teine võimalik põhjus on treeningandmete ja reaalse maailma andmete erinevused, näiteks võivad erineda andmete ajakohasus, kvaliteet või omadused.

9) **Pangad saavad tavaliselt uue laenu saamiseks sadu tuhandeid taotlusi. Iga taotlus sisaldab informatsiooni mitme tunnuse kohta, lihtsuse huvides kasutame selliseid tunnuseid nagu palk, vanus ja kas taotleja on väikelapse vanem. Oletame, et oleme loonud mudeli, mis ennustab, kas taotleja on usaldusväärne klient ehk kas ta maksab laenu tagasi. Oleme alltoodud tabelis kujutanud viit testandmepunkti. Nende märgendid on antud viimases veerus. Võrreldes ennustusi ja märgendeid (ehk tunnuse sihtväärtust), arvutage, mis on mudeli õigsus, täpsus ja saagis neil andmetel.**

Palk	Kas on väikelapse vanem	Vanus	Mudeli ennustus	Õigemärgend
4000	EI	31	JAH	JAH
1100	JAH	28	EI	EI
2500	EI	47	EI	JAH
1800	EI	56	JAH	JAH
1500	JAH	34	EI	JAH

Õigsus on täpsete vastuste osakaal, siin 3/5. Täpsus on positiivse klassi näidete hulk positiivse ennustuse saanud näidete hulgas, siin 2/2. Saagis on positiivsena tuvastatud näidete hulk kõigi positiivsete näidete hulgas, siin 2/4.

Peatükk 6

1) **Mis on tehisnärvivõrgu põhiline tööüksus ning kuidas see toimib?**
   Tehisnärvivõrgu põhiline tööüksus on tehisneuron. Tehisneuron saab sisendeid teistelt neuronitelt, korrutab need läbi vastavate kaaludega, liidab tulemused ja rakendab aktivatsioonifunktsiooni, mis muudab tulemuse üheks reaalarvuks.

2) **Miks nimetatakse teatud tüüpi närvivõrke sügavateks närvivõrkudeks?**
   Närvivõrke nimetatakse sügavateks, kui neil on palju kihte (tuhandeteni), mis suurendab nende võimet keerulisi mustreid õppida. Sügavus viitab võrgus olevate kihtide arvule.

3) **Kuidas toimub õppimine tehisnärvivõrgus ning milline on kõige levinum optimeerimisviis?**
   Õppimine toimub ühendustugevuste ehk kaalude muutmise teel. Kõige levinum optimeerimisviis on gradientlaskumise algoritm, kus iga kaalu korrigeeritakse eksimuse vähendamise suunas.

4) **Mis on pildi RGB väärtused? Kuidas moodustub RGB väärtustest pilt?**
   RGB väärtused määravad iga piksli punase, rohelise ja sinise värvi intensiivsuse.

5) **Miks on pildid keeruline andmetüüp. Milline tehnika, lähenemine on leiutatud, et pilte ikkagi efektiivselt töödelda?**
   Suurimad väljakutsed piltide puhul on tunnuste mitmetähenduslikkus (sama piksli asukoht võib ühel pildil olla koera nina, teisel pildil koera kõrv), piltide mitmekesisus (sama objektitüüp võib olla eri värvi, asukohas ja suurusega) ja tunnuste suur arv (miljonid pikslivääртused), mis võib viia ülesobitumiseni. Lahenduseks on olnud konvolutsiooniliste tehisnärvivõrkude kasutuselevõtt ja suurte treeningandmestike kasutamine.

6) **Miks on keel tehisintellektile keeruline, võrreldes näiteks tabeliandmetega?**
   Keele keerukus tuleneb mitmetähenduslikkusest, sõnade ja fraaside kontekstist sõltumisest ning sellest, et keele reeglid ja sõnavara on pidevas muutumises.

7) **Mis on alusmudel ja kuidas see erineb tavalisest sügavõppe mudelist?**
   Alusmudel on treenitud laiaulatuslikel andmetel, kasutades tihti isejuhendatud õpet. Erinevalt tavalisest sügavõppe mudelist on see mitmekülgsem ja kohandatav erinevateks ülesanneteks.

8) **Mida tähendavad nullsammuga ja mõne sammuga üldistumine ja miks need alusmudelite puhul olulised?**
   Nullsammuga üldistumine tähendab, et mudel suudab lahendada uusi ülesandeid ilma täiendava treeninguta, kasutades juba mudelisse talletatud teadmisi. See võimaldab mudelit kiirelt ja tõhusalt rakendada. Mõne sammuga üldistumine tähendab, et mudeli kohandamiseks uue ülesande lahendamiseks piisab vaid mõnest näitest.

Peatükk 7

1) **Vaatame järgmist päriselu situatsiooni. Andmeteadusmeeskond arendas lahkuva kliendi ennustamise süsteemi. Iga kord, kui klient helistas kõnekeskusesse, näidati kõnekeskuse operaatorile, kas see klient tõenäoliselt lahkub lähima kolme kuu jooksul või mitte. Süsteem on sellisel kujul töötanud pool aastat, aga klientide lahkumine ei ole vähenenud. Miks nii?**
   Selleks, et kliendid ei lahkuks, ei piisa ainult hästi töötavast lahkuvate klientide ennustamise mudelist. On vaja teada, kuidas kliendi lahkumist ära hoida. Käesolevas näites näidati kõnekeskuse operaatorile, kas klient tõenäoliselt on lahkumas. Samas, kui operaator ei tea, mida selle teadmisega peale hakata, siis mudel ei toimi. Koos lahkuvate klientide ennustamise mudeliga on vaja testida ka meetmeid, kuidas klientide lahkumist ära hoida. Näiteks võib küsida, millega kliendid rahul ei ole, kõrvaldada probleeme või pakkuda neile paremaid tingimusi. Erinevate lahenduste testimisel võib andmeteadus samuti kasuks tulla.

2) **Ettevõtte ABC Music müüb muusikat reklaami jaoks. Selleks oleks mugav, kui kasutaja saaks otsida muusikat emotsiooni järgi – kas muusika on kurb või rõõmus. ABC Musicu andmeteadlane on loonud sellise mudeli, aga tal ei õnnestu kuidagi tõsta süsteemi õigsust üle 70%. Milline oluline etapp võiks siin olla abiks, et välja selgitada, mis toimub?**
   Siin võiks kasutada baasmudelit, et aru saada ülesande keerukusest. Väga hästi sobiks selleks inimeste küsitlemine – kui hästi suudab inimene tuvastada muusikast emotsiooni. Mõned ülesanded ongi nii subjektiivsed, et 100% täpsust ei ole võimalik saavutada.

3) **Teie kontoris otsustati kasutada ukse avamiseks ise arendatud näotuvastussüsteemi. Paraku andis süsteem üsna tihti valenegatiivseid tulemusi (ei lasknud töötajaid sisse ja töötajad pidid kasutama võtit) ning ükskord lasi kontorisse tuvi, kes tänaval mööda lendas. Millise veaanalüüsi te teeksite?**
   Võib alustada sellest, et logida kõik süsteemi ennustused koos tingimustega – kes siseneb, mis kellaajal ja kas väljas on juba pime. Seejärel võib hakata märgistama kogutud pilte: kas taustal oli veel inimesi, milline oli inimese sugu, rass ja vanus. Seejärel saab analüüsida, kuidas süsteem toimib nende kategooriate lõikes.

Peatükk 8

1) **Aastal 2008 lubas Netflix anda miljon dollarit sellele, kes oskab parandada Netflixi soovitussüsteemi 10% võrra. Aastal 2009 maksid nad miljon dollarit võitjale. Lahendust ei võetud aga kunagi kasutusele. Miks?**
   Lahenduseks oli niivõrd keeruline ansambelsüsteem, et selle juurutamine ja hooldamine oleksid liiga kallid, et seda õigustada.

2) **Puidutehas toodab toormaterjalist ehituslaudu. Selleks, et määrata puidutüki kvaliteeti, kasutatakse masinnägemissüsteemi, mis peab**

klassifitseerima iga puidutüki esimesse (ilma oksakohtadeta, sile, ilus), teise (mõni oksakoht, aga mitte väga suur, sile) või kolmandasse klassi (palju oksakohti, võib esineda mõni muu defekt). Milliseid mõõdikuid te kasutaksite sellise süsteemi monitoorimiseks?
Kõigepealt võiks monitoorida, mis protsent ennustustest kuulub esimese, teise ja kolmanda klassi puitu. Ilmselt sõltub see ka puidu partiist, kuid on kahtlane, kui varem kuulus esimesse klassi 20% puidust, ja mõnes partiis tõuseb see järsku 50%-ni. Võiks salvestada sisendpildi parameetreid, nagu piksli heledus ja pildi suurus. See aitaks tuvastada probleeme valgustuse või näiteks kaamera nihkega. Loomulikult tuleks monitoorida ka latentsust ja serveri koormust.

3. Milline režiim mõjutab äri rohkem, kas varjurežiim või kanaarilinnu režiim?
Kanaarilinnu režiim, kuna ennustusi päriselt kasutatakse süsteemis.

Kasutatud allikad

Aruvee, E., Engstrand, U., & Olsson, U. (2000). Matemaatilise statistika põhimõisted bioloogilistele erialadele: Mathematical statistics with biological applications, in Estonian.

Bland, M. (2015). *An introduction to medical statistics*. Oxford university press.

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.".

Grus, J. (2019). *Data science from scratch: first principles with python*. O'Reilly Media.

Hastie, T. (2009). The elements of statistical learning: data mining, inference, and prediction.

Kull, M. (2022). Andmeteaduse meetodid LTAT.02.006 kursuse õppematerjalid. Tartu Ülikool

Kull, M, Aljanaki, A. (2022). Sissejuhatus andmeteadusesse LTAT.02.002 kursuse õppematerjalid. Tartu Ülikool

Käärik, E. (2013). E-kursuse "Andmeanalüüs II" õppematerjalid. Tartu Ülikool. https://dspace.ut.ee/server/api/core/bitstreams/76953a28-1622-440b-b286-889dcbb7d68e/content (kasutatud 31.05.2024)

Lember, J. (2021). Tehisõpe 1: Loengukonspekt. Tartu Ülikool. https://courses.ms.ut.ee/MTMS.02.046/2021_spring/uploads/Main/tehis%C3%B5pe21.pdf (kasutatud 31.05.2024)

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Mun, J. (2008). *Advanced analytical models: over 800 models and 300 applications from the basel II accord to Wall Street and beyond* (Vol. 419). John Wiley & Sons.

Parring, A. M., Vähi, M., & Käärik, E. (2000). *Statistilise andmetöötluse algõpetus*. Tartu Ülikooli Kirjastus/Tartu University Press.

Provost, F. (2013). *Data Science for Business: What you need to know about data mining and data-analytic thinking* (Vol. 355). O'Reilly Media, Inc.

Raag, M. (2020). Epidemioloogia ja biostatistika ARTH.03.006 kursuse materjalid. Tartu Ülikool. https://kodu.ut.ee/~maitraag/epi/_book/index.html (kasutatud 31.05.2024)

Sügis, E., Tampuu, A. (2021). "Andmeteaduse võimalused äriettevõttes" LTAT.02.020 kursuse õppematerjalid. Tartu Ülikool. https://courses.cs.ut.ee/2021/atva/fall (kasutatud 08.08.2023)

Sügis, E., Tampuu, A., Kolberg, L, (2021). MOOC Andmeteaduse võimalused äriettevõttes kursuse õppematerjalid. Tartu Ülikool. https://courses.cs.ut.ee/2021/atva/fall (kasutatud 08.08.2023)

Sügis, E., Tampuu, A. jt (2021). MOOC "Tehisintellekti Algkursus" õppematerjalid. Tartu Ülikool. https://courses.cs.ut.ee/2022/Tehisintellekti_algkursus (kasutatud 07.08.2023)

Tooding, L-M. (2014). Faktoranalüüs. K. Rootalu, V. Kalmus, A. Masso, ja T. Vihalemm (toim), Sotsiaalse analüüsi meetodite ja metodoloogia. õpibaas.https://samm.ut.ee/faktoranalyys/ (kasutatud 07.08.2023)

Vähi, M. (2021). Tartu Ülikooli kursus Andmeanalüüs 2 (MTMS.01.007) kursuse õppematerjalid. https://ois2.ut.ee/#/courses/MTMS.01.007/details (kasutatud 08.07.2023)

Viited teadusartiklitele

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283).

Banerjee, S., & Lavie, A. (2005, June). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization* (pp. 65-72).

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Hong, J. W., & Williams, D. (2019). Racism, responsibility and autonomy in HCI: Testing perceptions of an AI agent. *Computers in Human Behavior*, *100*, 79-84.

Khan, W., Daud, A., Khan, K., Muhammad, S., & Haq, R. (2023). Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends. *Natural Language Processing Journal*, 100026.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... & Girshick, R. (2023). Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4015-4026).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459-9474.

Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74-81).

Liu, G., Reda, F. A., Shih, K. J., Wang, T. C., Tao, A., & Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 85-100)

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., ... & Houlsby, N. (2022, October). Simple open-vocabulary object detection. In *European Conference on Computer Vision* (pp. 728-755). Cham: Springer Nature Switzerland.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, *1*(2), 3.

Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, *164*, 113794.

Redmon, J. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684-10695).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* (pp. 234-241). Springer International Publishing.

Sullivan, G. M., & Feinn, R. (2012). Using effect size—or why the P value is not enough. *Journal of graduate medical education*, *4*(3), 279-282.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.

Tukey, J. W. (1949). Comparing individual means in the analysis of variance. *Biometrics*, 99-114.

Xu, L., Tang, Q., Lv, J., Zheng, B., Zeng, X., & Li, W. (2023). Deep image captioning: A review of methods, trends and future challenges. *Neurocomputing*, *546*, 126287.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., ... & Hu, X. (2024). Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, *18*(6), 1-32.

Muud allikad

Eesti Keele Instituut, Sõnaveeb. https://sonaveeb.ee/ (kasutatud 31.05.2024)

HuggingFace repositoorium. https://huggingface.co/ (kasutatud 31.05.2024)

IBM Corporation. (2021) IBM SPSS Modeler CRISP-DM Guide. https://www.ibm.com/docs/en/spss-modeler/saas?topic=guide-introduction-crisp-dm

Isikuandmete kaitse üldmäärus (GDPR) 2016/679 EUR-Lex. (2016) https://eur-lex.europa.eu/ET/legal-content/summary/general-data-protection-regulation-gdpr.html (kasutatud 31.05.2024)

MacKenzie, I., Meyer, C.,Noble, C. (2013). How Retailers Can Keep Up with Consumers. *McKinsey & Company*.https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers (kasutatud 31.05.2024)

Microsoft. (2024). GitHub Copilot, kasutatud https://copilot.microsoft.com/ (kasutatud 31.05.2024)

Pytorch teek. https://pytorch.org/ (kasutatud 31.05.2024)

RAGAS teek. https://docs.ragas.io/en/stable/ (kasutatud 31.05.2024)

TensorFlow teek. https://www.tensorflow.org/ (kasutatud 31.05.2024)

The Data Visualisation Catalogue. https://datavizcatalogue.com/ (kasutatud 31.05.2024)

Viited jooniste alllikatele

Awais, M., Naseer, M., Khan, S., Anwer, R. M., Cholakkal, H., Shah, M., ... & Khan, F. S. (2023). Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*.

Cui, Y., Lu, S., & Liu, S. (2023). Real-time detection of wood defects based on SPP-improved YOLO algorithm. *Multimedia Tools and Applications*, *82*(14), 21031-21044.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Stanford University. CS231n: Convolutional Neural Networks for Visual Recognition. Retrieved from https://cs231n.github.io/convolutional-networks/

Weiss, E., Caplan, S., Horn, K., & Sharabi, M. (2024). Real-Time Defect Detection in Electronic Components during Assembly through Deep Learning. *Electronics*, *13*(8), 1551.
Park, D. Y., & Lee, K. H. (2019). Arbitrary style transfer with style-attentional networks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5880-5888).