TARTU ÜLIKOOL

Arvutiteaduse instituut

Informaatika õppekava

Uku Zingel

Õppijate tulemuste ennustamise tööriist

Bakalaureusetöö (9 EAP)

Juhendaja: Reimo Palm, PhD

Tartu 2023

Õppijate tulemuste ennustamise tööriist

Lühikokkuvõte:

Käesoleva bakalaureusetöö kirjutamise käigus valmis tööriist, mis võimaldab ennustada aine „Programmeerimine“ tudengite tulemusi. Tööriistaga saab ennustada õppija lõpptulemust aines igal hetkel semestri esimese kuueteistkümne nädala jooksul. Tööriist on põhiliselt mõeldud selle aine õppejõududele, kes saaksid selle abil juba varakult tuvastada tudengeid, kes võivad aine läbi kukkuda. Selle kasutamiseks tuleb hankida käesoleva aine hinnete ja Moodle’i logide failid. Tööriista loomiseks kasutati programmeerimiskeelt Python ja erinevaid masinõppe tehnikaid.

Võtmesõnad:

Masinõpe, ennustusmudelid, tööriist

CERCS: P175 Informaatika, süsteemiteooria. P176 Tehisintellekt

A tool for predicting the results of the students

Abstract:

A tool for predicting grades for students was created in the process of writing this Bachelor’s Thesis. The tool is designed to predict the performance of students in the course "Compute Programming". The tool can predict results for the first sixteen weeks of the course. It is primarily intended for instructors of this course, who can use it to identify students who may fail the course early on. To use it, the grade records and Moodle logs of this course need to be obtained. The tool was built using the Python language and various machine learning techniques.

Keywords:

Machine learning, prediction models, tool

CERCS: P175 Informatics, system theory. P176 Artificial Intelligence

Sisukord

Sissejuhatus

Tudengite tulemuste ennustamine on tegevus, mis toob kaasa kasu nii kursust läbi viivatele õppejõududele kui ka kursust võtvatele tudengitele. Õppejõud saavad niimoodi varakult leida tudengeid, kellel võib tekkida probleeme kursuse lõpetamisega ja neid varakult abistada. Tudengid ise saavad tulemuste ennustamise läbi paremini hinnata, mis seisus nad teatud ajahetkel on ning vajaduse korral rohkem kursuse läbimisse panustada.

Antud töö eesmärgiks on koostada mudel, mille abil saaks semestri jooksul igal ajahetkel ennustada õppija oodatavat lõpptulemust aines, ning realiseerida see programmina. Selle töö peamine eesmärk on, et saaks võimalikult varakult tuvastada õppijad, kellel on aines raskusi ja kes võivad olla väljalangemise ohus, et saaks neile õigel ajal abi anda. See tööriist oleks suunatud peamiselt aine õppejõududele ning peaks olema neile mugavalt kasutatav. See tööriist on mõeldud kasutamiseks esimese semestri aines "Programmeerimine" [12].

Bakalaureusetöö jaoks oli vaja ehitada ennustusmudel ja tööriist. Ennustamise jaoks on olemas kahte sorti andmeid. Esiteks on programmeerimise aine hinnete tabel, kus on olemas kõik kursuse jooksul saadud punktilised tulemused. Lisaks oli antud ka kursuse Moodle’i tegevuste logid. Lisaks toorandmetele olid kaasa antud masinõppe aines tehtud lõpuprojektide raames loodud mudelid.

Esimeses peatükis antakse ülevaade töö teoreetilisest taustast ja eelmainitud lõpuprojektidest. Teise peatükis räägitakse selle bakalaureusetööga seotud ja sarnastest töödest. Kolmandas peatükis kirjeldatakse tööriista nõudeid ja selle arendusprotsessi. Neljas peatükk koosneb tööriista validatsiooni kirjeldusest. Viimases peatükis tehakse analüüs ja koostatakse kokkuvõte.

1. Teoreetiline ülevaade

Tulemuste ennustamine on masinõppe ülesanne. Järgnevalt kirjeldatakse meetodeid, mille abil on sageli lahendatud sarnaseid ülesandeid, ja antakse ülevaade varasemalt loodud mudelitest.

1.1 Varem kasutatud mudelid

Masinõppe põhjal tehtavaid mudeleid, mis ennustavad õpilaste tulemusi, on ka varasemalt tehtud. Järgnevalt räägitakse veidi varem tehtud sarnaste ülesannete puhul rakendatud lähenemisviisidest.

1.1.1 Klassifitseerimine

Buenaño-Fernández jt [1] on kirjutanud, et klassifitseerimine (*classification*) on üks laialdasemalt kasutatav andmekaevetehnika ja seda tehnikat rakendatakse eelnevalt klassifitseeritud andmekirjetele, et töötada välja ennustav mudel, mida saab kasutada klassifitseerimata andmekirjete klassifitseerimiseks. Seda tehnikat saab rakendada näiteks otsustuspuu (*decision tree*) algoritmi abil. Protsess sisaldab kahte etappi: õppimine ja klassifitseerimine ise. Õppimisetapis analüüsitakse treeningandmestikku valitud klassifitseerimisalgoritmi abil. Nende sõnul on otsustuspuu algoritmi rakendamise peamine eelis see, et selle tulemusi saab hõlpsasti tõlgendada ja seletada tänu selle graafilisele esitusele, mis võtab kokku kaudsete otsustusreeglite mudeli (*implicit decision rules*). Ehk see näitab, milliste muutujate väärtuste järgi otsustatakse.

1.1.2 Kaasfiltreerimine

Iqbal jt [5] on kirjutanud, et üks sageli kasutatav masinõppe algoritm on kaasfiltreerimine (*collaborative filtering*). Hariduse kontekstis ennustavad kaasfiltreerimisalgoritmid keskmist hinnet, leides otsitava õpilasega sarnaseid õpilasi andmete hulgas. Selle meetodi puhul tehakse ennustused, valides ja koondades teiste õpilaste hinded. Ette on antud loend m õpilasest $S = \{s_1, s_2, ..., s_m\}$ ja n kursusest $C = \{c_1, c_2, ..., c\}$. Igal õpilasel on kursuste loend, mis näitab milliseid kursuseid on õpilane läbinud ja mis olid saavutatud tulemused. Läbi selle leitakse tema keskmine hinne. Kaasfiltreerimisalgoritmi ülesanne on leida õpilane, kelle keskmine hinne on sarnane mõne teise õpilasega. Seda meetodit rakendasid nad kolme sammuna. Esiteks algoritm mõõdab, kui sarnane on iga andmebaasis olev õpilane uuritava õpilasega, tehes seda sarnasusmaatriksi arvutamise abil. Teiseks leitakse kõige sarnasemad õpilased uuritava õpilasega k lähima naabri algoritmi abil. Lõpuks ennustatakse uuritava õpilase kursuse keskmine hinne, kasutades kõige sarnasemate õpilaste hinnete keskmise leidmist. See võib olla lihtsalt keskmise hinde leidmine,

kuid võib kasutada ka kaalutud keskmise hinde leidmist. Nende sõnul tähendab see seda, et sarnasemate õpilaste keskmisel hindel on rohkem kaalu, kui vähem sarnasematel õpilastel.

1.1.3 Maatriksi tegurdamine

Iqbal jt [5] on defineerinud, et maatriksi tegurdamine (*matrix factorization*) on maatriksi jagamine kaheks või enamaks maatriksiks niimoodi, et nende maatriksite korrutis võrdub esialgse maatriksiga. Maatrikstegurdamist kasutatakse varjatud tegurite avastamiseks ja puuduvate väärtuste ennustamiseks. Nende poolt defineeritud ülesande kontekstis käsitletakse õpilaste tulemuste ennustamise probleemi soovitussüsteemi (*recommender system*) probleemina, mille lahenduseks on sageli kasutatud singulaarset lahutust (*singular value decomposition*).

Iqbal jt [5] kirjutavad, et singulaarne lahutus on maatriksi tegurdamise tehnika, mis lagundab õpilaste ja kursuste maatriksi väiksemaks maatriksiks. Singulaarlahutust kasutatakse laialdaselt maatriksi R parima k-järgu lähenduse leidmiseks. Iga r-järku maatriksi saab taandada k-järku maatriksiks, kus k väärtus on väiksem kui r väärtus. Võttes maksimaalse sellise k väärtuse saadakse lähendus R_k, mis on saadud maatriksist R nii, et Frobeniuse norm on minimeeritud. Frobeniuse norm ($||R - R_k||_F$) on defineeritud kui summa elementide ruutudest, mis asuvad $R - R_k$-s. Nad kirjutavad ka, et keskmise hinde ennustamiseks kursusel eeldab singulaarne lahutus, et iga õpilase hinne koosneb kursuse erinevate varjatud tegurite eelistuste summast. Nende sõnul käib õpilase i hinde ennustamine kursuse j jaoks järgnevalt: vaja on võtta õpilaste tunnusmaatriksist võetud õpilase vektori i ja kursuse vektori j skalaarkorrutis.

Iqbal jt [5] kirjutavad, et singulaarse lahutuse probleem seisneb selles, et see ei ole efektiivne suurte ja hõredate andmekogumite puhul. Artiklis tehti ettepanek kasutada parima järgu k arvutamiseks juhuslikku gradiendi laskumise (*stochastic gradient descent*) algoritmi maatriksi lähendamiseks, kasutades ainult algse maatriksi teadaolevaid reitinguid. Nad kirjutavad, et juhuslik gradiendi laskumine on kumer optimeerimistehnika, mis saab kõige täpsemad väärtused nendest kahest maatriksist, mis saadakse algse maatriksi singulaarse lahutuse käigus singulaarse lahutuse teel. Juhusliku gradiendi laskumise algoritm koosneb kolmest etapist. Esiteks luuakse uuesti õpilaste ja kursuste maatriks, korrutades kaks madalamat järku maatriksit. Teiseks leitakse sihtmaatriksi ja genereeritud maatriksi erinevus. Lõpuks korrigeeritakse kahe madalamat järku maatriksi elementide väärtused, arvestades, kui suure osa nad moodustasid genereeritud maatriksi loomisel. See protsess kordub senikaua, kuni erinevus on väiksem kui varasemalt määratud lävi.

Nad kirjutavad ka, et vähendades õpilaste ja kursuste maatriksi dimensioonide arvu, muutub täitmise kiirus väiksemaks ja ennustuse täpsus suureneb. Dimensioonide vähendamine vähendab müra ja ülesobitamist (*overfitting*).

1.1.4 Tehisnärvivõrgud

Rastrollo-Guerrero jt [9] ütlevad, et tehisnärvivõrk (*artificial neural network*) koosneb omavahel tihedalt seotud üksuste komplektist, mida nimetatakse töötlemiselementideks (*processing elements*). Närvivõrgu struktuur ja funktsioon on inspireeritud bioloogilisest kesknärvisüsteemist, eriti ajust. Nad kirjutavad, et iga töötlemiselement on loodud jäljendama oma bioloogilist vastet neuronit, mis võtab vastu kaalutud sisendite komplekti ja reageerib vastava väljundiga.

Rastrollo-Guerrero jt [9] kirjutavad, et tehisnärvivõrke on kasutatud õpilaste tulemuste ennustamisel mitmetel eri viisidel. Näiteks edasisuunalisi (*feedforward*) tehisnärvivõrke saab õpetada ennustama hindamistestide hindeid, võttes arvesse kursuse osalisi hindeid. Tehisnärvivõrke õpetatakse välja hindama semestri õppeedukust, kasutades kumulatiivset keskmist hinnet. Mitte ainult hindamistulemused, vaid ka õpilastelt saadav lisateave võivad tehisnärvivõrkude ennustusi parandada. Nad kirjutavad, et seetõttu saab kasutada õpilaste põhiteavet koos kognitiivsete ja mittekognitiivsete meetoditega, et koostada mitmest tehisnärvivõrgust mudel, mis ennustaks õpilaste tulemuslikkust. Nad kirjutavad ka, et tehisnärvivõrke on ka kasutatud, et analüüsida akadeemilist tulemuslikkust mõjutavate kognitiivsete ja psühholoogiliste muutujate vahelist mittelineaarset seost, rühmitades õpilased tõhusalt erinevatesse kategooriatesse vastavalt nende eeldatavale sooritustasemele.

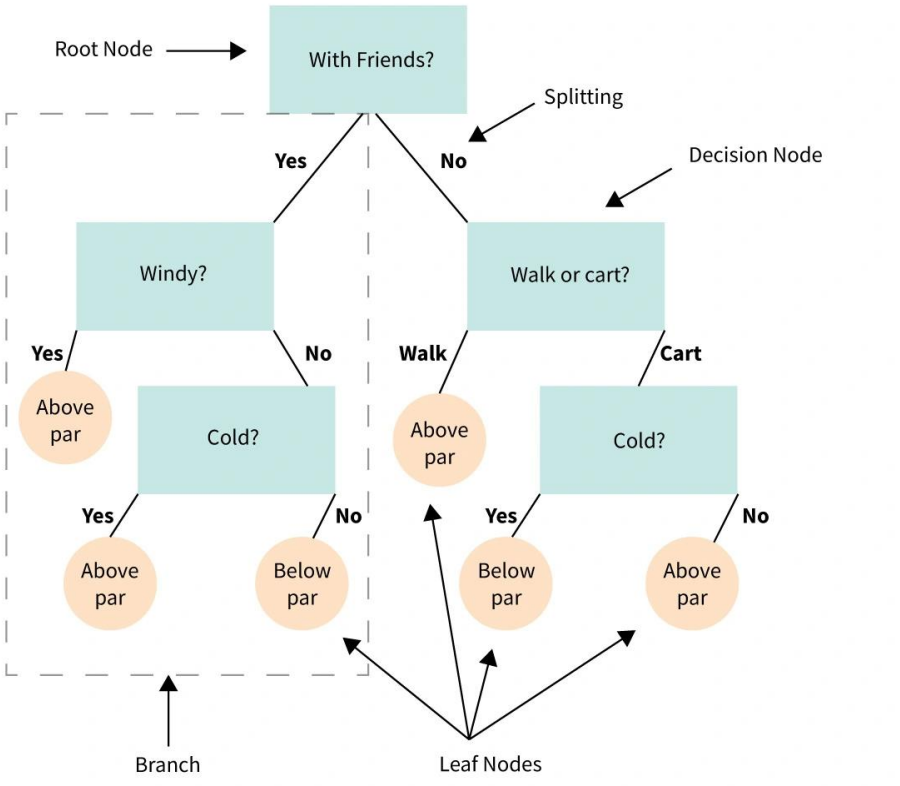
1.1.5 Juhusliku metsa algoritm

Juhusliku metsa (*random forest*) algoritm on järelevalvega masinõppe algoritm, mis on äärmiselt populaarne ja mida kasutatakse masinõppe klassifikatsiooni- ja regressiooniprobleemide lahendamiseks. Selle algoritmi mõistmiseks tuleb enne mõista, mis on otsustuspuu (*decision tree*) algoritm. [7]

Otsustuspuu meenutab puud. Puu alus on juurtipp (*root node*). Juurtipust kasvab välja otsustustippude (*decision node*) jada, mis kujutavad algoritmi poolt langetatavaid otsuseid. Otsustustippudest pärinevad lehetipud (*leaf node*), mis esindavad nende otsuste tagajärgi. Iga otsustustipp tähistab küsimust või jaotuspunkti ning otsustustippudest tulenevad lehetipud esindavad võimalikke vastuseid. Lehetipud võrsuvad otsustustippudest sarnaselt sellele, kuidas

leht võrsub puuoksal. Seetõttu nimetame iga otsustuspuu alamsektsiooni haruks (*branch*). Näide selle kohta on joonisel 1. [10]

Juhusliku metsa algoritm on olemuselt mets, mis koosneb paljudest otsustuspuudest. Mida rohkem puid, seda efektiivsem see on. Samamoodi, mida suurem on puude arv juhuslikus metsaalgoritmis, seda suurem on selle täpsus ja probleemide lahendamise võime. Juhusliku metsa algoritm sisaldab etteantud andmestiku erinevate alamhulkade kohta mitut otsustuspuud ja võtab nende prognoositavate tulemuste keskmise, et parandada täpsust. See põhineb ansambliõppe (*ensemble learning*) kontseptsioonil, mis on oma olemuselt mitme algoritmi kombineerimise protsess, et lahendada keeruline probleem ja parandada mudeli tööd ja täpsust. [7]



Joonis 1. Otsustuspuu näide.

1.2 Ülevaade lõppprojektidena tehtud mudelitest

2021. aastal oli õppijate tulemuste ennustamise mudeli teema pakutud ainetes "Masinõpe" [11] ja "Sissejuhatus andmeteadusesse" [13] projektiteemana ning 5 rühma realiseerisid selle peal erinevad masinõppemudelid. Järgnevalt antakse mudelite ja avastuste kohta ülevaade.

1.2.1 Rühm A3

Rühmal A3 oli kaks eesmärki, esiteks tuvastada õpilasi, kellel võib aine lõpetamisega tekkida raskusi, ja teiseks ennustada õppijate lõpptulemusi.

Esimese eesmärgi jaoks defineeriti, et tudeng on raskustes, kui ta esimese 6 hindelise ülesande eest saab kokku alla 50% võimalikest punktidest. Rühm A3 ei suutnud leida efektiivset meetodit, et tuvastada raskustes tudengeid.

Õppijate tulemuste ennustamiseks kasutas rühm A3 juhusliku metsa klassifitseerijat (*random forest classifier*). See realiseeriti andes igale õpilasele hinne skaalal A-F. Rühm A3 võttis oma mudeli sisendandmeteks videoloengu ülesannete punktide kogusumma ja kodutööde punktide kogusumma. Ennustati lõplikku hinnet. Mudel oli üsna kehv, kuna suutis ennustada ainult tulemusi A ja F. Lisaks sellele ei ennustanud mudel mitmete õpilaste puhul üldse lõpphinnet. Mudel saavutas täpsuse 0.44.

1.2.2 Rühm B1

Rühmal B1 oli kolm eesmärki. Esiteks identifitseerida tudengite käitumismustrite põhjal tüüpilisemad tudengiprofiilid, mille abil saaks tudengeid käitumise järgi rühmitada. Teine eesmärk oli ennustada tudengite lõpphindeid käitumise järgi. Viimaseks eesmärgiks oli leida raskustes tudengeid.

Rühm B1 jagas tudengid lõplikute tulemuste põhjal kuude rühma. Iga rühma iseloomustajaks sai lõpptulemus (A-F). Seejärel uuriti iga eri rühma käitumist Moodle’is, ja sealt saadi kätte mõned käitumismustrid. Näiteks avastati, et tudengite rühm tulemusega F veetis kõige rohkem aega foorumit vaadates.

Ennustamist tegi rühm B1 juhusliku metsa klassifitseerimisest (*random forest classifier*) kasutades. Rühm B1 ennustas lõpphinnet, mitte lõpptulemust. Andmeteks võeti esimese seitsme nädala testide ja kodutööde tulemused. Loodud mudel saavutas täpsuse umbes 0.5. Mudel oli natuke katki, kuna teatud õpilastele ei suutnud leida lõpphinnet.

Rühm B1 ei leidnud efektiivset meetodit, millega leida raskustes tudengeid.

1.2.3 Rühm D3

Rühmal D3 olid sarnased eesmärgid nagu rühmal B1. Esiteks ennustada tudengite lõplikke hindeid kursuse andmete põhjal, seejärel identifitseerida raskustes tudengid ja lõpuks leida tüüpilisemad tudengiprofiilid.

Rühm D3 võttis ennustamise sisendandmeteks kursuse hinded ja peamised logitegevused. Nad jagasid andmed ajaliselt nelja rühma: enne esimest kontrolltööd, suuremate tööde vaheline aeg, enne teist kontrolltööd ja enne eksamit. Rühm D3 kasutas ennustamiseks kuute erinevat mudelit: otsustuspuud (*decision tree*), juhusliku metsa (*random forest*), lineaarregressiooni (*linear regressioon*), harjaregressiooni (*ridge regression*), lassoregressiooni (*lasso regression*) ja kerge gradiendi võimendamise masinat (*LGBM*). Täpsuse hindamiseks kasutati keskmist absoluutset viga (*mean absolute error*) ehk näidati kui palju erines ennustatud tulemus keskmiselt õigest tulemusest. Tulemused on tabelis 1.

Rühm D3 kasutas tudengiprofiilide leidmiseks aprioorset algoritmi. Vaadati läbi logiandmed ja otsiti mustreid. See rühm leidis 4 profiili: kogenud, klikkijad, foorumikülastajad ja hinnetejälgijad. Kogenud olid tudengid, kes olid kursuse materjalidega varem tuttavad ja kes läbisid kursuse eeleksamiga. Klikkijad olid tudengid, kes vaatasid väga palju teste ja nende tulemusi. On võimalus et nad proovisid teste läbida toore jõuga. Foorumikülastajad osalesid aktiivselt foorumites, nad nii kirjutasid kui ka lugesid teiste vastuseid. Hinnetejälgijad olid tudengid kes väga tihti oma tulemusi ja hindeid vaatasid.

Raskustes tudengite leidmiseks kasutas rühm D3 ennustamisest saadud tulemusi. Vaadati esimese suurema tööni olevate andmete pealt tehtud ennustustulemusi ja kui ennustatav tulemus oli vähem kui 70% koguhindest, siis defineeriti tudeng raskustes olevana.

Tabel 1. Ennustatud tulemuste keskmine erinevus päristulemustest.

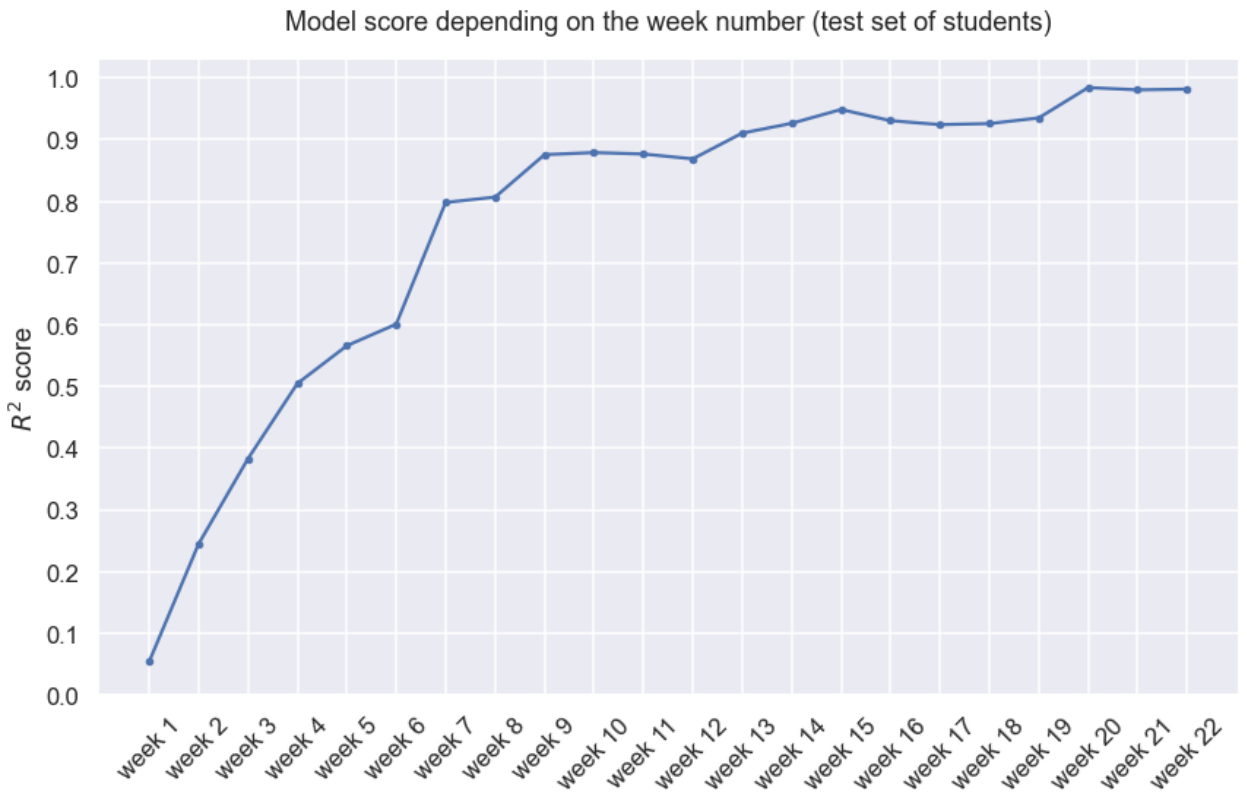
Treeningandmed kuni ajahetkeni	**Otsustuspuu**	**Juhuslik mets**	**Lineaarne regressioon**	**Harja-regressioon**	**Lasso-regressioon**	**Kerge gradiendi võimendamise masin**
Enne esimest kontrolltööd	10.4124	8.1639	13.746	13.3173	12.6226	11.1004
Kontrolltööde vahel	7.6226	6.1868	10.571	10.4345	9.4967	7.7310
Enne teist kontrolltööd	8.9442	5.1752	9.153	8.8521	8.2327	7.1456
Enne eksamit	1.6225	0.3940	1.2218e-13	0.0029	0.0043	0.9087

1.2.4 Rühm P05

Rühmal P05 oli kaks eesmärki, esiteks luua jälgimissüsteem, mille abil saaks ennustada tudengite hindeid suvalisel kursuse hetkel. Teiseks eesmärgiks oli identifitseerida Moodle'i logide abil aktiivsusmustreid.

Rühm P05 jagas andmed ümber nädalateks ja lõi iga tudengi iga nädala kohta põhjaliku andmestiku, mis koosnes hinnetest ja Moodle'i tegevustest. Kokku tuli sisendandmeteks iga tudengi iga nädala kohta 58 tunnust.

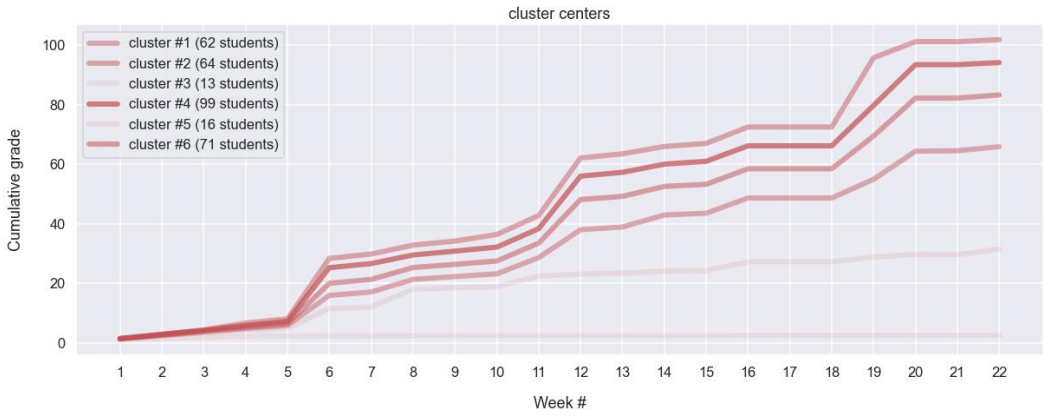
Ennustamiseks kasutas rühm P05 hulka regressioonimudeleid, mis töötasid igaüks ühe nädala andmetega. Seega sai dünaamiliselt valida, kui palju andmeid parasjagu mudelile anda. Mudelid söötsid järjest üksteisele andmeid ja lõpuks tehti ennustus. Ennustustulemused on näha joonisel 2.



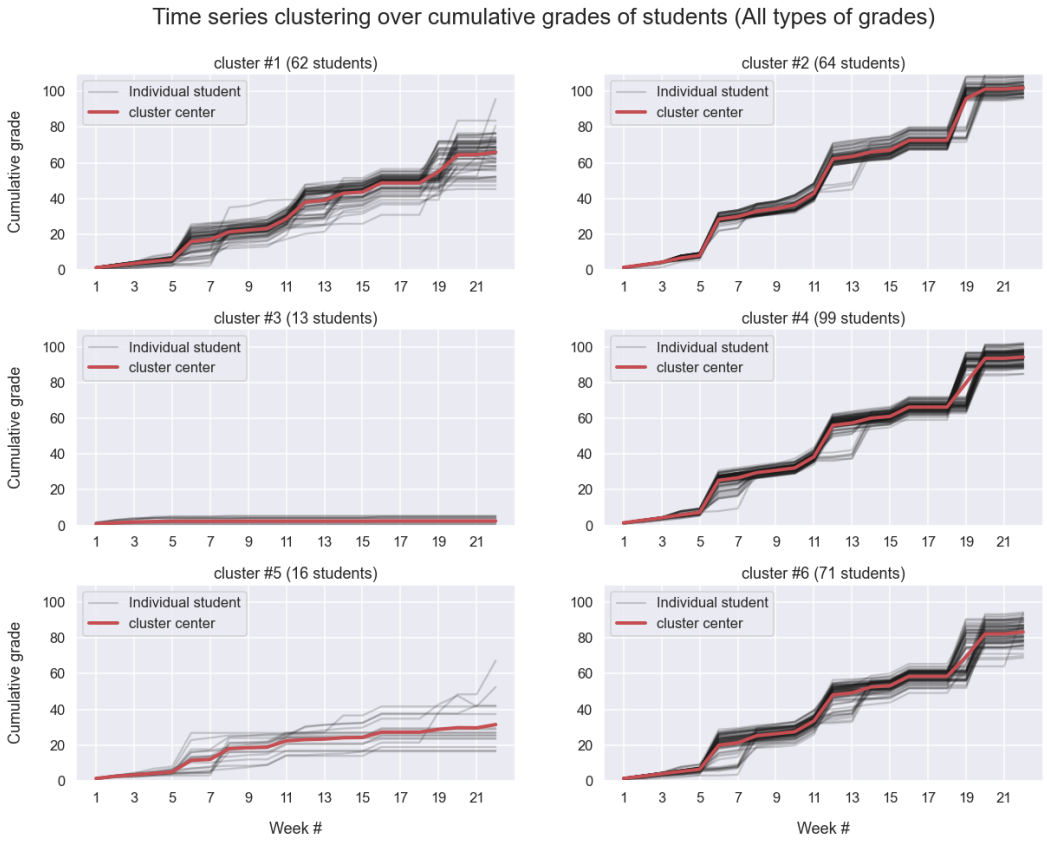
Joonis 2. Mudeli täpsus nädalate kaupa.

Nad leidsid, et on ilmne, et saadud hindeid kirjeldavad tunnused mängivad ennustamisel domineerivat rolli. Teisisõnu, nende tunnuste tähtsus on teiste tunnustega võrreldes kõrge. Tunnuste valikul märkasid nad aga, et mainitud tunnused ei ole esimese 1-6. õppenädala jooksul nii kasulikud. Tõepoolest, algusnädalatel on tudengitel viimastega võrreldes vähe punktisummat, seetõttu pole see ennustamisel nii kasulik. Selgus, et logide aktiivsusandmete lisamine suurendas mudeli täpsust esimestel õppenädalatel natuke alla poole võrra. Nende peamine eesmärk oli võimalikult kiiresti lõplikku hinnet täpselt ennustada, mis võimaldab varajast sekkumist. Tõepoolest, viimase nädala ennustamine on kasutu, sest kõik hinded on sel ajahetkel juba teada, kuid näitasid seda siiski täielikkuse huvides.

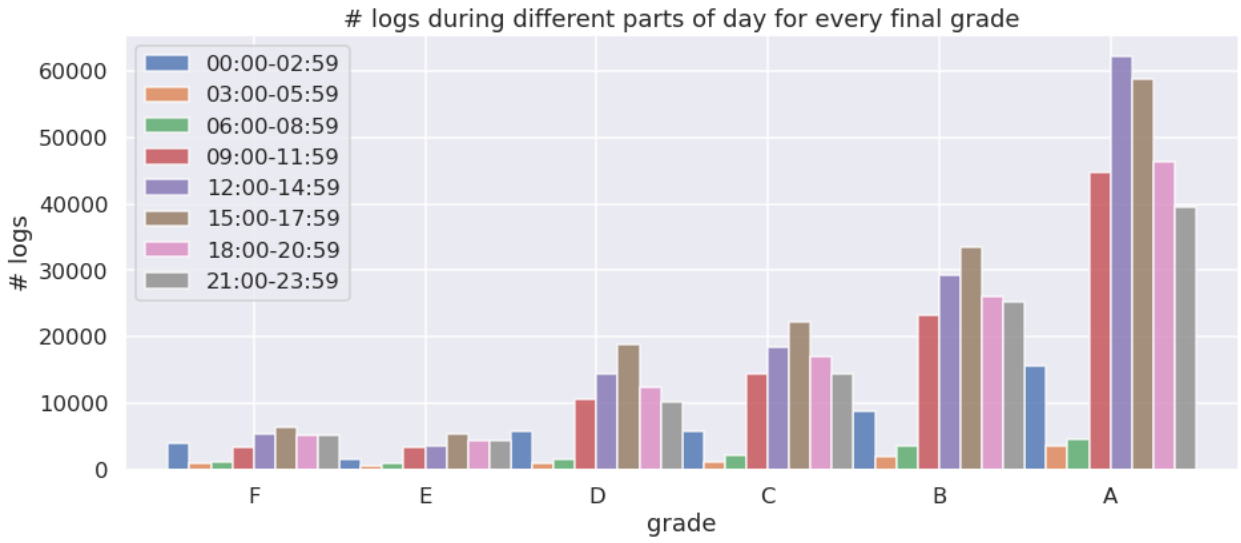
Levinumate aktiivsusmustrite leidmiseks kasutas rühm P05 EDA graafikuid ja klasterdamist. Mõned leitud mustrid ja tulemused on näha joonistel 3-6.



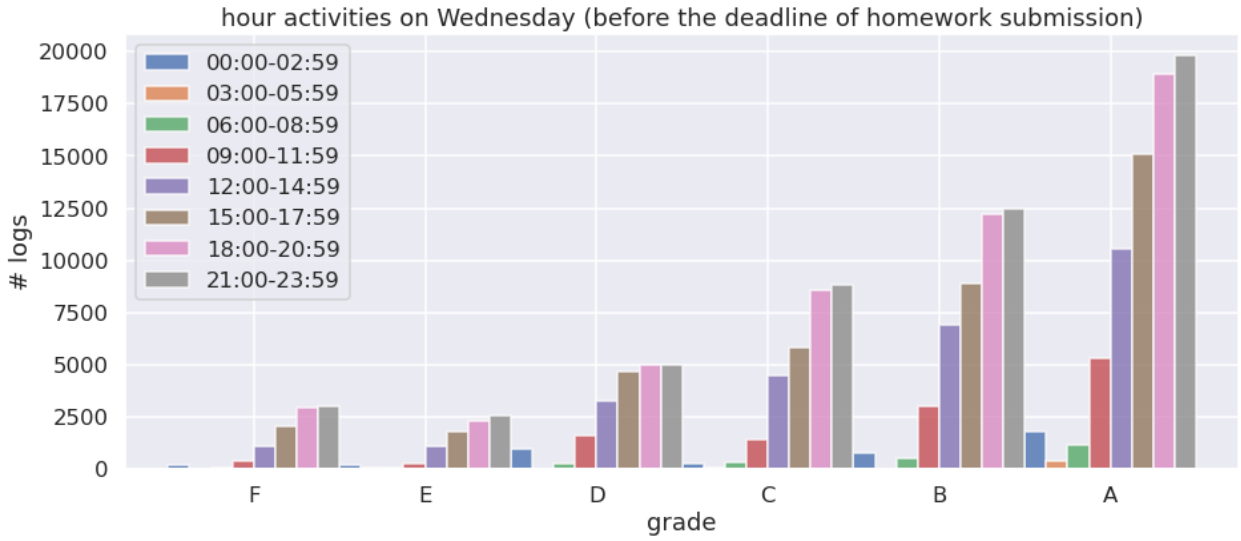
Joonis 3. Aktiivsusmustrite klasterdamise tulemused nädalate kaupa.



Joonis 4. Õpilaste klastrid eri aktiivsusmustrite kaupa.



Joonis 5. Logiaktiivsus erinevatel kellaaegadel hinnete kaupa.



Joonis 6. Logiaktiivsus kolmapäeviti enne kodutöö tähtaega.

Joonistelt 3-6 on näha nii mõndagi. Vaadates jooniseid 3 ja 4 näeme erinevaid klasterdamisest saadud õpilaste gruppe. See näitab mingil määral ära tekkinud rühmad, mille põhjalt saame minna edasi järgmiste jooniste juurde. Jooniselt 5 saame näha erinevate kellaaegade logiaktiivsust hinderühmade kaupa. Sealt on selgelt näha, et A saanud õpilased tegid kõikidel kellaaegadel keskmiselt rohkem Moodle'i tegevusi, kui kõik madalamate hinnetega õpilased. Sealt on ka näha, et B saanud õpilased olid aktiivsemad kui neist madalamate tulemustega õpilased. Viimaks on näha, et moodustus kaks gruppi, kes olid logiaktiivsuselt sarnased. Esimeses grupis olid C ja D

saanud õpilased ja teises grupis olid E ja F saanud õpilased. Joonis 6 näitab sarnast pilti joonisega 5. A saanud õpilased olid kõige aktiivsemad ja mida madalam oli hinne, seda madalam oli aktiivsus. Erandina olid F saanud õpilased aktiivsemad kui E saanud õpilased. Kokku näeme, et mida kõrgem oli hinne, seda rohkem tehti logitegevusi. See tähendab, et hiljem ennustusmudelit ehitades saame seda fakti enda kasuks kasutada.

1.2.5 Rühm P06

Rühmal P06 olid sarnased eesmärgid nagu teistel rühmadel. Esiteks tuvastada raskustes õpilased, teiseks ennustada tudengite lõpphindeid ja viimaseks leida levinumaid käitumismustreid.

Raskustes õpilaste tuvastamiseks kasutas rühm P06 kahte mudelit, mis ennustasid, kas tudeng on raskustes või mitte. Esimeses mudelis kasutati miinimumi maksimumi skaleerijat (*MinMaxScaler*), sünteetilise vähemuse ülesobitamise tehnikat (*Synthetic Minority Oversampling Technique; SMOTE*), juhusliku metsa (*random forest*) ja tugivektormasinat (*support vector machine*). Selle mudeliga saavutati raskustes tudengi ennustamise täpsus 91.18%. Teises mudelis kasutati miinimumi maksimumi skaleerijat (*MinMaxScaler*) juhusliku metsa (*random forest*), tugivektormasinat (*support vector machine*) ja klassi kaalu (*class weight*). Selle mudeliga saavutati kõigi nädalate andmete põhjal ennustamistäpsus 92.65%.

Rühm P06 kasutas tudengite hinnete ennustamiseks tugivektori regressiooni (*SVR*) mudelit. Sisendandmeteks olid ainult hinded, Moodle'i logisid see rühm hinde ennustamisel ei kasutanud. Mudel saavutas kaheteistkümne esimese nädala andmetega ennustamistäpsuse 76.55%. Mudel ennustas lõpphinnet, mitte punktilist lõpptulemust.

Levinumate käitumismustrite otsimisel leidis rühm P06, et kõige harvemad Moodle'is tehtavad tegevused olid erinevate objektide kustutamised. Selle alla lähevad nii foorumipostitused, kommentaarid kui ka arutelud. Kõige rohkem tehtavad tegevused olid kursuse eri osade vaatamised, olgu selleks kas siis testi tulemuse, kursuse või siis kodutöö esituse vaatamine.

2. Seotud ja sarnased tööd

Sarnaseid töid on nii Eestis kui maailmas ka varem tehtud. Järgnevalt kirjeldatakse natuke selliseid töid.

2.1 IKT eriala üliõpilaste varajase väljalangemise ennustamise veebirakenduse loomine R Shiny abil [6]

Tallinna Tehnikaülikoolis 2022. aastal tehtud lõputöö raames ehitati veebirakendus, mis suudaks ennustada tudengi väljalangemise tõenäosust. Valminud rakendust rakendatakse sisseastumise hetkel, et ennustada tudengi tõenäosust välja langeda. Töö ennustab kolme sisendi põhjal. Esiteks sisseastujatele tehtud küsitlus, teiseks sisseastuja isiklikud andmed, näiteks tema keskkooli lõputunnistuse keskmine hinne või sisseastumisvestluse punktide arv. Kolmas ennustusviis rakendus veidi hiljem. Sissejuhatavas aines oli vaja kirjutada essee. See essee ongi kolmas sisend. Antud töö sarnaneb käesoleva tööga, olles tööriist, millega ennustatakse õpilaste kohta. Käesolev töö keskendub tudengi ühe kursuse lõpptulemuse ennustamisele ja kasutab teistsuguseid sisendandmeid.

2.2 Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review [9]

See artikkel tutvustab uuringut õpilaste lõplike hinnete ennustamiseks, lähtudes nende tegevusest Moodle'i õppehaldussüsteemis ja Zoomi kaudu läbi viidud veebiloengutes osalemisest, kasutades statistilisi ja masinõppe tehnikaid. Hii-ruut-testi (*Chi-square test*) kasutatakse õpilaste lõpphinnete ja sündmuse konteksti (loengud, lähtekood, harjutused ja ülesanded) seose ning loengutes osalemise ja lõpptulemuste vahelise seose hindamiseks. Õpilaste lõplike hinnete ennustamiseks rakendatakse nelja masinõppe algoritmi (*Random Forest, XGBoost, KNN ja SVM*), jaotusega 70% treeningandmetele ja 30% testiandmetele. Antud töö sarnaneb käesoleva tööga, kuna ennustab samuti tudengite kursuse lõpptulemust. Lisaks kasutab see sarnaselt Moodle'i logisid. Erineb antud töö käesolevast tööst, kuna käesoleva töö eesmärk on praktilisem ja kasutab lisaks logidele ka punktilisi andmeid.

2.3 Õpisoorituse ennustamine Moodle'i logiandmete ja enesehinnanguliste õppimisega seotud psühholoogiliste tegurite põhjal [8]

Tallinna Tehnikaülikoolis tehti 2020. aastal magistritöö, mille raames ennustati mitme õppeaine raames tudengite lõpptulemusi. Ennustamismudeleid ehitati üles kolme komponendi abil. Esiteks

olid jooksvad tulemused: testid, kodutööd ja muud punktidega hinnatavad tulemused. Teiseks olid Moodle'i keskkonna aktiivsuslogid ja viimaseks olid kursuse jooksul tehtavate küsitluste vastused. Kõige vähem rõhku pandi nende kolme seast punktilistele tulemustele. Antud töö ja käesolev töö sarnanevad, kuna mõlemad ennustavad tudengi kursuse lõpphinnet ja kasutavad selleks osaliselt sarnaseid andmeid. Erinevus on selles, et käesolev töö ei süvene niivõrd psühholoogiste tegurite teemal, vaid põhieesmärk on luua praktiline tööriist.

2.4 Learners of an introductory programming MOOC: background variables, engagement patterns and performance [3]

Tartu Ülikoolis 2022. aastal kirjutati doktoritöö raames antud artikkel. See töö keskendub MOOC (*massive open online course*) uurimisele. MOOC-ide puhul on tähtis see, et need on väga paljudele inimestele avatud, mis tähendab, et seda satub õppima väga erineva taustaga inimesi. Antud töö keskendub just eri taustade uurimisele, ehk proovib näha, kas ja mis mõju avaldavad kursuse tulemusele eri taustad. Antud töö võrdleb järgmisena kursuse lõpetanuid ja mittelõpetanud tudengeid, et leida sealt mustreid. Antud töö puhul on andmete hulk tunduvalt suurem ja erinevam kui käesolevas töös. Lisaks on antud töö palju teoreetilisem, keskendutakse peamiselt mustrite leidmisele, käesoleva töö eesmärgiks on tööriist.

2.5 Application of Machine Learning in Predicting Performance for Computer Engineering Students: A Case Study [1]

Antud artiklis pakutakse välja metoodika, mille kohaselt viiakse algselt läbi andmete kogumise ja eeltöötlemise protsess ning seejärel teises etapis sarnaste õppeedukusmustritega üliõpilaste rühmitamine. Järgmises etapis valiti tuvastatud mustrite põhjal kõige sobivam juhendatud õppealgoritm ja seejärel viidi läbi katseprotsess. Lõpuks esitleti ja analüüsiti tulemusi. Tulemused näitasid masinõppetehnikate tõhusust õpilaste soorituste ennustamisel.

2.6 Machine Learning Based Student Grade Prediction: A Case Study [5]

Antud artiklis kasutatakse kaasfiltreerimist (*collective filtering*), maatriksi tegurdamist (*matrix factorization*) ja piiratud Boltzmanni masina (*restricted Boltzmann machine*) tehnikaid reaalmaailma andmete süstemaatiliseks analüüsimiseks. Hinnati bakalaureuseõppekava üliõpilaste õppeedukust elektrotehnika osakonnas. Leiti, et piiratud Boltzmanni masina tehnika oli olnud parem kui teised tehnikad, mida kasutatakse õpilaste tulemuste ennustamiseks konkreetsel kursusel.

2.7 Student Academic Performance Prediction by using Decision Tree Algorithm [4]

Antud töö uurib õpilase õppeedukust, kasutades otsustuspuu algoritmi. Simulatsiooni tulemused näitavad, et juhusliku metsa klassifitseerija (*random forest classifier*) näitas paremat tulemust kui võrdlevad otsustuspuu algoritmid (*decision tree algorithms*).

2.8 Multiclass Prediction Model for Student Grade Prediction Using Machine Learning [2]

Antud artiklis esitatakse masinõppetehnikate põhjalik analüüs, et ennustada esimese semestri kursustel õpilaste lõplikke hindeid. Antud artiklis tuuakse esile kaks meetodit. Esiteks võrreldi kuut tuntud masinõppetehnikat, nimelt otsustuspuud (*decision tree*), tugivektormasinat (*support vector machine*), naiivset Bayesi (*naive Bayes*), K-lähima naabri (*k nearest neighbors*), logistilise regressiooni (*logistical regression*) ja juhusliku metsa (*random forest*) algoritme, kasutades 1282 tegeliku õpilase kursuse hinnete andmeid. Teiseks pakuti välja mitmeklassiline ennustusmudel. Seda kasutati, et vähendada ülesobitamist (*overfitting*) ja valesti klassifitseerimise tulemusi. Need tulenesid sünteetilise vähemuse ülevalimise tehnika (SMOTE) kasutamisest.

3. Tööriista nõuded ja arendusprotsess

Antud peatükis tutvustatakse tööriista ja selle valmistusprotsessi.

3.1 Tööriista nõuded

Antud andmed on kahel kujul. Esiteks on hinded ja teiseks on logiandmed. Tööriist peab olema võimeline kasutama mõlemaid, et ennustada võimalikult täpselt tudengi lõpptulemust. Lisaks sellele peab tööriist toimima igal kursuse hetkel. Kui jätta välja eksamiperiood, on kursusel 16 nädala jagu tööd.

3.2 Mudelite uurimine

Enne tööriista loomisega peale hakkamist oli esimene samm vaadata korralikult üle eelmise aasta lõppprojektidena tehtud mudelid ja otsustada, kas ja kui palju võtta neist inspiratsiooni. Kokku oli antud töö loomiseks antud 5 erineva rühma tööd. Põhjalikumalt on nende rühmade töö kirjeldatud alapeatükis 1.2. Rühma A3 tööst ei olnud abi, mudel ennustas lõpptulemuste asemel lõpphindeid ja ka seda mitte väga hästi. Mudel suutis ennustada ainult A või F tulemust. Rühmal B01 olid sarnased probleemid, ennustati jälle lõpphinnet, kuid seda mitte väga täpselt ja mõningate tudengite puhul ei suutnud mudel üldse ennustada. Rühm P06 oli täpsem kui eelnevad kaks, kuid sügavama vaatluse põhjal tuvastati, et see on vähem täpne kui järgnevad kaks rühma. Nende kolme rühma tööst ei võetud väga palju inspiratsiooni.

Kõige rohkem võttis töös kasutatud mudel inspiratsiooni P05 ja D3 rühma mudelitest. Mõlemad mudelid ennustasid lõpptulemust regressioonülesandena. Rühma D3 mudel kasutas viit erinevat algoritmi ja siis võrdles tulemusi. Selle rühma kõige täpsem kasutatud algoritm oli juhusliku metsa algoritm (*random forest*). Kuid sügavama uurimise järeldusena oli näha, et selle rühma tulemused olid paremad, kui nad tegelikud oleksid pidanud olema, kuna logifailides olevate andmete töötluses oli tehtud mõned vead. Näiteks kasutati varastel nädalatel lõpunädalate kodutööde esitamise logisid, kuigi antud tööni oli mitmeid nädalaid jäänud. Lisaks oli ka hinnete töötlus tehtud kergekäeliselt, näiteks lisaülesannete töötlus oli üles ehitatud valesti. Selle rühma tööst sai käesolev töö inspiratsiooniks natuke andmete töötlust ja kasutatava algoritmi, milleks oli juhusliku metsa algoritm.

P05 rühma mudel oli ehitatud järgnevalt: üksteise otsa kuhjatud regressioonalgoritmid, millest igaüks tegeles ühe nädala informatsiooniga ja andis tulemuse teisele edasi. Mudel oli natuke

vähem täpne, kui D3 mudel, kuid seal oli logiandmete töötlus korrektne. Selle mudeli peamine kasutusprobleem oli see, et mudel oli üles ehitatud koguandmestiku põhjal. See suutis koguandmestiku põhjal anda edasi iganädalasi tulemusi, aga seda dünaamiliselt kasutada oli keeruline, mistõttu ei sobi see nii hästi selle tööriista jaoks.

Andmete töötlus neil kahel rühmal erines. Rühm D3 jättis hinnete tabeli enam-vähem samale kujule nagu see oli originaalselt, eraldades ja puhastades vajamineva. Erinevate ajahetke jaoks loodi erinevad failid. Rühm P05 koondas kõik andmed nädalate kaupa samasse faili, kus oli iga tudengi jaoks iga nädala kohta oma rida. Igal real oli näidatud nädala jooksul saadud punktid ja neid söödeti nädal nädala kaupa mudelisse.

3.3 Andmete töötlus

Peale mudelitega tutvumist oli järgmine samm andmetöötlus valmis teha. Eelmiste projektide andmetöötluse eeliseks oli see, et oli vaja tegeleda ainult ühe aasta andmetega, käesoleva töö jaoks kasutati kolme erineva aasta andmeid. Andmed pärinesid aastatest 2020-2022. Antud aine ehk „Programmeerimine“ [12] varieerub aastati natukene, mistõttu oli vaja andmetöötluseks esiteks saada andmed samale kujule ja teiseks ehitada selle töötlusprotsessi sisse hulk veakontrolle, et tulevaste andmete töötlus läheks samuti probleemideta.

3.3.1 Hinnete töötlus

Esimesena keskenduti selle tööriista loomisel hinnete töötlemisele. Hinnete töötlus ei nõudnud väga palju vaeva, kuni oli tegemist ühe aasta andmetega, siis oli ainult vajalik puhastada välja hulk mittevajalikke andmeid. Teiste aastate andmete lisamisel olid aastate andmed erinevad. Kas puudusid mõned tulbad, olid mingid tulbad kuskil juures või olid tulpade nimetused eri aastatel erinevad. See tähendas, et tuli eri aastate hinded ühildada samasugusteks. Peale selle protsessi lõpetamist tuli jagada hinded nädalate kaupa failidesse. Kokku jagati andmed kuueteistkümnesse eri faili. Näide kolmanda nädala andmetest on joonisel 8.

3.3.2 Logifailide töötlus

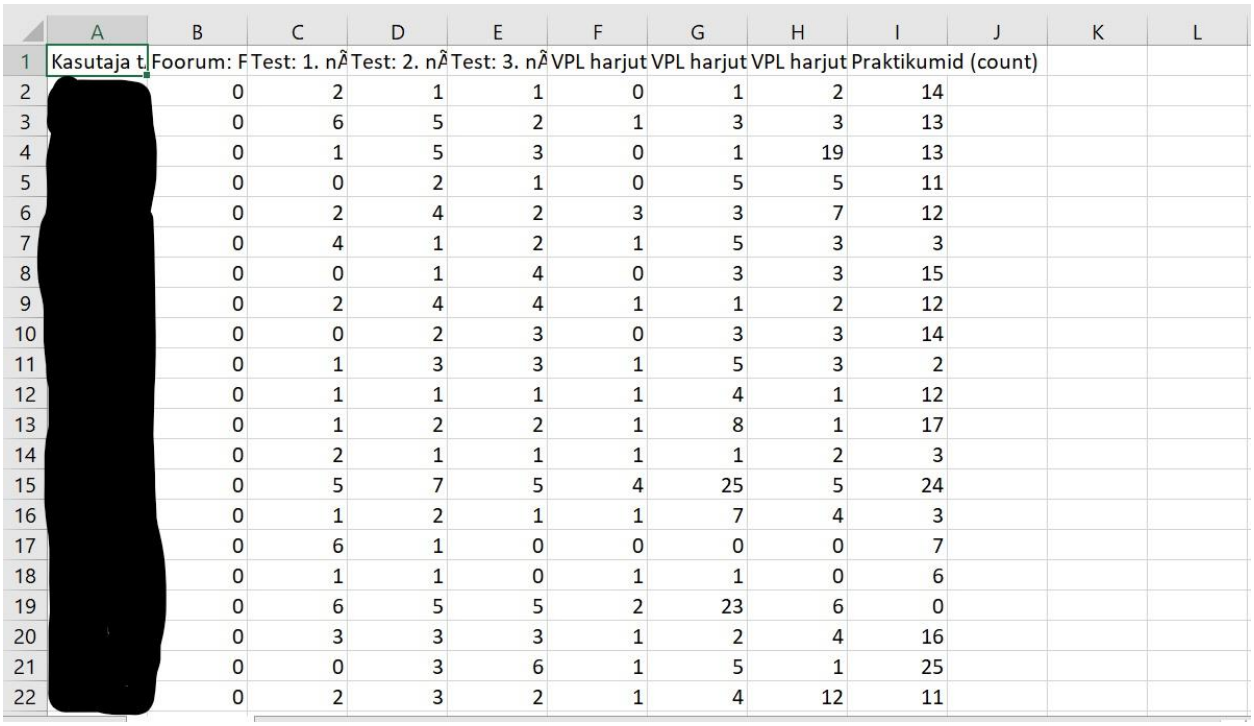
Teiseks keskenduti logifailide töötlemisele. Logifailide töötlemine oli keerulisem kui hinnete töötlemine. Esiteks oli logifailides tunduvalt rohkem informatsiooni kui hinnete tabelis. Tuli kõik see sisse lugeda kasutatavale kujule. Lisaks tuli välja sorteerida mittevajalik informatsioon. Esiteks oli logifailides lisaks tudengitele ka õppejõudude tegevused, mis tuli eemaldada. Teiseks oli vaja leida, milliseid logiandmeid kasutada. Seda tehti katse-eksituse meetodil, kus vaadati, millised

logitegevused mõjutasid mudeli täpsust ja kuidas. Kui kõik logiandmed alles jäid, muutus mudel ebatäpsemaks, kuna andmeid oli kas liiga palju või läks see andmestik tasakaalust välja. Kui logiandmeid oli tunduvalt rohkem, kui hinnete andmeid, siis oli hindetulpade kaal mudelis väiksem. Lõpptulemusena jäeti alles foorumi tegevused, testide esitamine, kodutööde esitamine, lisatööde esitamine ja praktikumides osalemise arv. Seda tehti sellepärast, kuna nende tegevuste korduv tegemine andis vajaminevat informatsiooni. Välja jäeti näiteks erinevate testide, tööde ja muude asjade vaatamise tegevused, kuna need kas ei muutnud mudelit täpsemaks või siis muutsid mudeli täpsust halvemaks. Peale seda, kui oli välja sõelutud vajaminev informatsioon, jagati tulemused nädalate kaupa kuueteistkümnesse faili. Ka siin oli asi keerulisem kui hinnete puhul, kuna ei saanud lihtsalt öelda, et kolmanda testi tulemus läheb kolmandasse faili, vaid teatud tegevusi sai teha ka kas nädal enne või nädal hiljem, või üldse mingil muul ajal. Õnneks oli logitegevustele kaasa pandud tegevuse ajahetk. Läbi selle sai jaotatud logid nädalatele. Näide kolmanda nädala andmetest on joonisel 9.

Viimase etapina kombineeriti logiandmed hindeliste andmetega ja tulemuseks saadi kuusteist faili, mis kõik sümboliseerisid ühte nädalat kursusel.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Nimi	Test:1. nÃ	Test:2. nÃ	Test:3. nÃ	VPL harjut	VPL harjut	VPL harjut	VPL harjut	VPL harjut	VPL harjut	VPL harjut	Ãœlesann	Test:Eelek	Ãœlesann	Kogutulemus
2	0	[redacted]	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	109.4
3	1		0.48	0.46	0.5	0.5	0.5	0	0.5	0.5	0.5	0	0	0	0	97.6
4	2		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	95.68
5	3		0.5	0.45	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	73.13
6	4		0.48	0.38	0.35	0	0.3	0.5	0.5	0.5	0.5	0.5	0	0	0	42.23
7	5		0.48	0.44	0.47	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	84.92
8	6		0.45	0.46	0.48	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	92.32
9	7		0.5	0.5	0.35	0.5	0.5	0	0.5	0.5	0	0	1	1	91	91
10	8		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	105.4
11	9		0.5	0.48	0.41	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0	85.5
12	10		0	0.5	0.26	0.5	0	0.5	0.5	0.5	0.5	0	0	0	0	29.51
13	11		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	1	65	98.5
14	12		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	75.96
15	13		0.5	0	0	0	0	0	0	0	0	0	0	0	0	0.5
16	14		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	100.4
17	15		0.43	0.45	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	94.9
18	16		0.5	0.48	0.36	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	93.87
19	17		0.5	0.48	0.46	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	103.8
20	18		0.48	0.3	0.45	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	77.59
21	19		0.42	0.45	0.35	0.5	0.4	0.5	0.5	0.5	0	0.5	0	0	0	93.33

Joonis 8. Kolmanda nädala hinded.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Kasutaja t	Foorum: F	Test: 1. nÂ	Test: 2. nÂ	Test: 3. nÂ	VPL harjut	VPL harjut	VPL harjut	Praktikumid (count)			
2	[redacted]	0	2	1	1	0	1	2	14			
3		0	6	5	2	1	3	3	13			
4		0	1	5	3	0	1	19	13			
5		0	0	2	1	0	5	5	11			
6		0	2	4	2	3	3	7	12			
7		0	4	1	2	1	5	3	3			
8		0	0	1	4	0	3	3	15			
9		0	2	4	4	1	1	2	12			
10		0	0	2	3	0	3	3	14			
11		0	1	3	3	1	5	3	2			
12		0	1	1	1	1	4	1	12			
13		0	1	2	2	1	8	1	17			
14		0	2	1	1	1	1	2	3			
15		0	5	7	5	4	25	5	24			
16		0	1	2	1	1	7	4	3			
17		0	6	1	0	0	0	0	7			
18		0	1	1	0	1	1	0	6			
19		0	6	5	5	2	23	6	0			
20		0	3	3	3	1	2	4	16			
21		0	0	3	6	1	5	1	25			
22		0	2	3	2	1	4	12	11			

Joonis 9. Kolmanda nädala logiandmed. Iga tulp näitab, kui palju kordi on tehtud mingit Moodle'i tegevust.

3.4 Mudeli valmistamine

Peale andmete töötlust oli järgmine tööriista loomise etapp luua kasutamiseks sobiv mudel. Lõpliku mudeli loomiseks katsetati läbi hulk erinevaid andmeteaduse algoritme ja tööriistu.

3.4.1 Lõplik mudel

Lõpliku mudeli loomiseks katsetati läbi hulk erinevaid andmeteaduse algoritme ja tööriistu. Täpsemalt räägin ebaõnnestunud mudelitest järgmises jaotises. Lõplik algoritm, mis valiti selle tööriista jaoks, oli juhusliku metsa regressiooni algoritm. See sai valitud, kuna oli kõikidest katsetatud algoritmidest kõige täpsem ning suutis tulemusi ennustada korrektses vahemikus. Selle all mõeldakse, et paljud teised algoritmid ennustasid kas negatiivseid või võimatult kõrgeid tulemusi. Mõnedel tuli ennustusele miinimumpiir, näiteks kümme punkti, millest allapoole algoritm ei suutnud ennustada. Juhusliku metsa algoritm ennustas õigetes piirides ja veel tähtsam, suutis ennustada kõikidest mudelitest kõige täpsemini just alla 50-punktilisi tulemusi, mille leidmine ongi selle töö eesmärk. Niiviisi suudetakse juba varakult tuvastada raskustes tudengeid. Mudeli ennustustäpsus nädalate kaupa on toodud välja tabelis 2.

Tabel 2. Tööriista mudeli ennustustulemused testandmetel nädalate kaupa. Teises veerus olev arv näitab keskmist erinevust ennustuse ja päristulemuse vahel.

Nädal	**Täpsus**
1	18.11846981170675
2	16.513179984170346
3	14.50134473416065
4	13.052693078336194
5	12.124284326949198
6	11.66847223485097
7	8.995354989728636
8	8.230722480341985
9	7.809084116755009
10	7.514920445309388
11	7.3898614046795895
12	7.3160562470288015
13	6.385306120309559
14	6.308029895607933
15	6.300183367230095
16	5.951970964332596

3.4.2 Katsetused teiste mudelitega

Enne kui valiti lõplik mudel, prooviti läbi hulk erinevaid algoritme ja masinõppe tööriistu. Alguses oli plaan kasutada lõplikus tööriistas kahte parimat mudelit, kuid kahjuks osutusid kõik ülejäänud algoritmid sobimatuks.

Esimene algoritm, mida prooviti kasutada, oli lineaarregressiooni (*linear regression*) algoritm. See algoritm saavutas küll paremuselt teise täpsuse kasutatud algoritmide seas, kuid sel oli kaks peamist probleemi. Esiteks oli selle algoritmi poolt ennustatud tulemustel miinimumpiir umbes nelja punkti juures. See ei tundu küll palju, kuid kui ekraanil on kahe mudeli tulemused, siis see võib anda kasutajale vastakaid tulemusi. Teiseks läks ennustatud tulemused algoritmi punktiskaalast välja. Punktiskaala on nullist saja kümneni, kuid algoritm ennustas mõnel juhul üle 120. Maksimaalne ennustatav tulemus, mida nähti oli 160 lähedal.

Teine algoritm, mida prooviti kasutada, oli otsustuspuu algoritm (*decision tree*). Kuna juhusliku metsa algoritm koosneb mitmetest otsustuspuudest, siis oli lootus, et see on efektiivne. Kahjuks oli tulemus halb. Algoritmi poolt ennustatud tulemustel oli jällegi miinimumpiir, milleks oli seekord arv viieteistkümne lähistel. Erand sellele reeglile oli see, et ennustas paari õpilase jaoks negatiivseid tulemusi.

Järgmiseks katsetati lasso- ja harjaalgoritme. Need olid omavahel üsna sarnased selle poolest, et ennustasid üsna palju tulemusi negatiivsete punktidega. Seetõttu ei peetud kumbagi sobivaks.

Prooviti kasutada ka XGB-regressiooni (*XGBregressor*), mida kasutas näiteks rühm P05. Nende tehtud mudel nõudis väga erineval kujul andmeid, mistõttu ajapuuduse tõttu ei jõutud seda algoritmi väga hoolikalt proovida. Seega kahjuks seda algoritmi ei kasutatud.

Prooviti kasutada ka tehisnärvivõrku (*artificial neural network*), kuid see andis halvemaid tulemusi kui mitu muud mudelit. Spekuleeriti, et see on suure tõenäosusega põhjustatud kas treeningandmete ebasobivusest tehisnärvivõrkude jaoks. Ehk andmed on mingis aspektis tasakaalust väljas. Teine võimalus on see, et treeningandmeid pole piisavalt palju, et tehisnärvivõrku korralikult välja õpetada. Seega ei kasutatud ka tehisnärvivõrke.

Lõpuks otsustati, et ei kasutata ühtegi siin peatükis mainitud algoritmi, kuna kõikidel oli probleeme ennustamisega, mis oleks suure tõenäosusega toonud kaasa kasutajale segaduse tekitamise.

3.5 Kasutajaliidese loomine

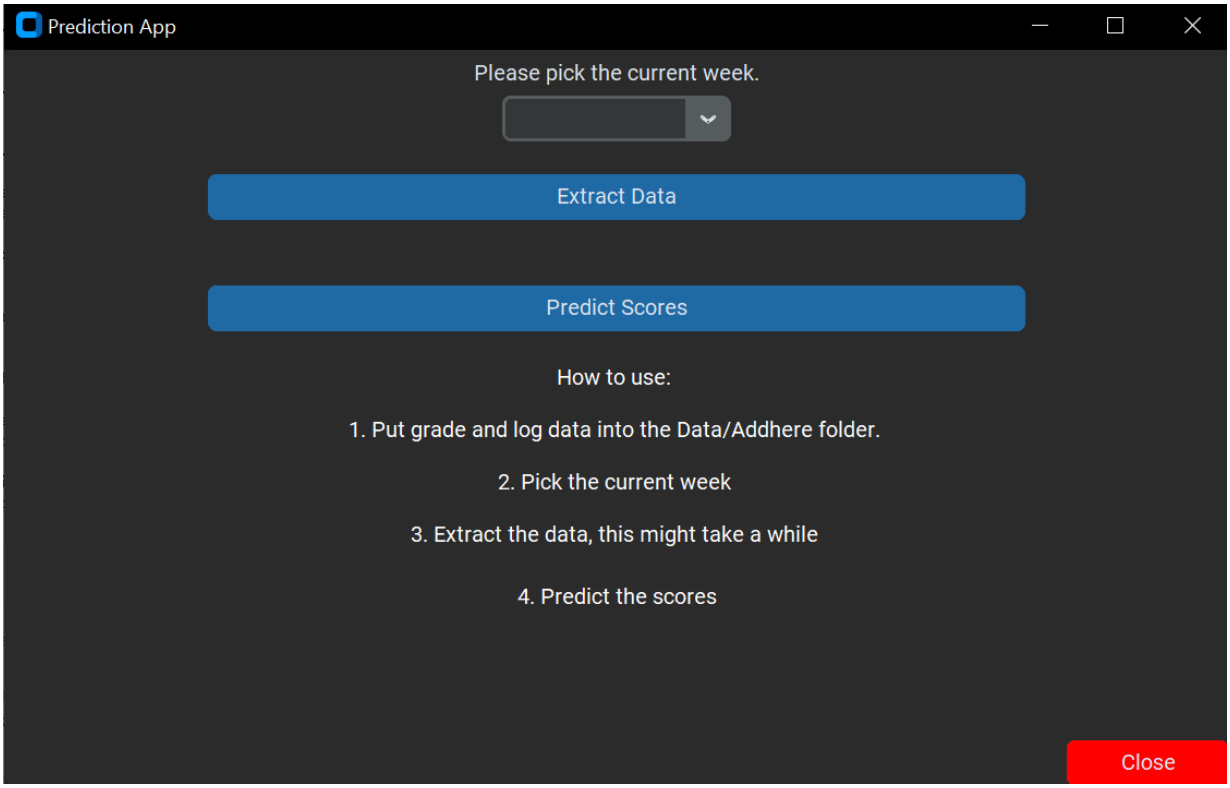
Selleks etapiks oli kõik funktsionaalne osa tööriistast valmis saanud. Andmed olid töödeldud ja mudel oli valmis. Viimaseks sammuks oli luua tööriistale kasutajaliides. Selle loomisel lähtuti loogikast, et mida kergem ja arusaadavam on selle kasutamine, seda parem. Ei tahetud luua midagi erakordselt keerulist, vaid eesmärgiks oli pigem lihtne ja kasutatav tööriist.

Esiteks loodi esimene prototüüp. See ei näinud välja kuigi ilus ja seal ei töötanud kõik funktsioonid, aga see oli samm edasi. Seejärel keskenduti funktsionaalsuse töölesaamise peale. Tööriista kõige lihtsam funktsioon on jooksva nädala valimine. Kasutaja valib nädala, et tööriist teaks, mis nädala andmete põhjal uusi andmeid töödelda ja milliste andmetega neid pärast võrrelda. Tööriistas on kaks peamist funktsiooni. Esiteks on andmete töötlus. Selle alustamiseks on vaja tööriista „Addhere“ nimelisse kausta lisada kõik ennustamiseks vajalikud andmed. Nendeks on

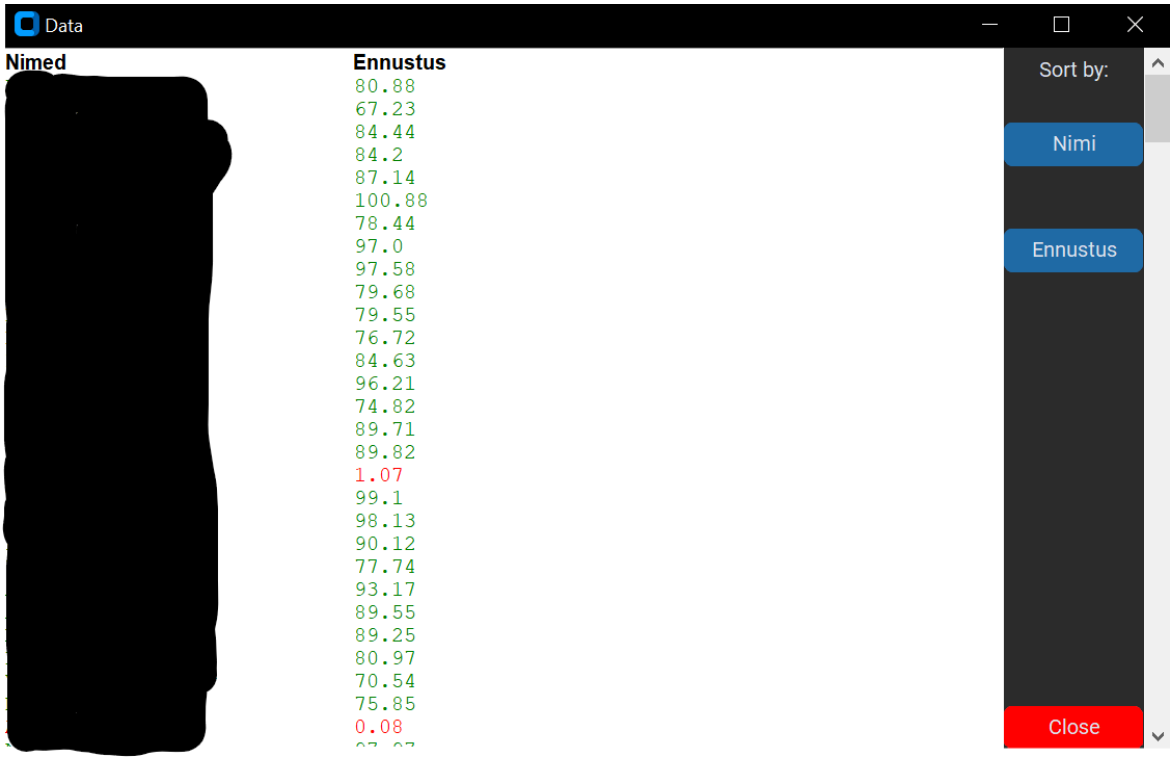
hinnete ja logide failid. Järgmisena tuleb vajutada nupule „Extract Data“ ja andmetöötlus algab. Andmetöötluse lõpptulemuseks on fail nimega „data.csv“, kus on kõik ennustamiseks vajalik.

Teine tööriista peafunktsioon on ennustamine ise. Kui andmed on töödeldud ja õiges kaustas, siis tuleb vajutada nupule „Predict Scores“. Seejärel algab ennustamise protsess. Töödeldud andmed ja jooksva nädala andmed loetakse sisse vastavalt treening- ja testandmetena. Seejärel tehakse veakontroll, et võrrelda andmehulkade tabeleid ja vaadata kas seal on tulbanimed samad. Kui ei ole, siis tehakse korda. Seejärel avab tööriist uue lehe, kus on tabeli kujul esitatud kõik andmetest pärit õppijate nimed ja nende ennustatud tulemused. Tulemused on rohelist värvi, kui ennustus on üle kuuekümne punkti ja punased kui alla selle. See aitab luua kasutajale visuaalse abi, et tuvastada kergemini raskustes õpilasi. Tabelit saab sorteerida nii nime kui ka ennustuse põhjal.

Joonisel 8 on näidatud, kuidas näeb välja tööriista avaleht ja joonisel 9 on näidatud, kuidas näeb välja tööriista tulemuste leht.



Joonis 8. Tööriista avaleht

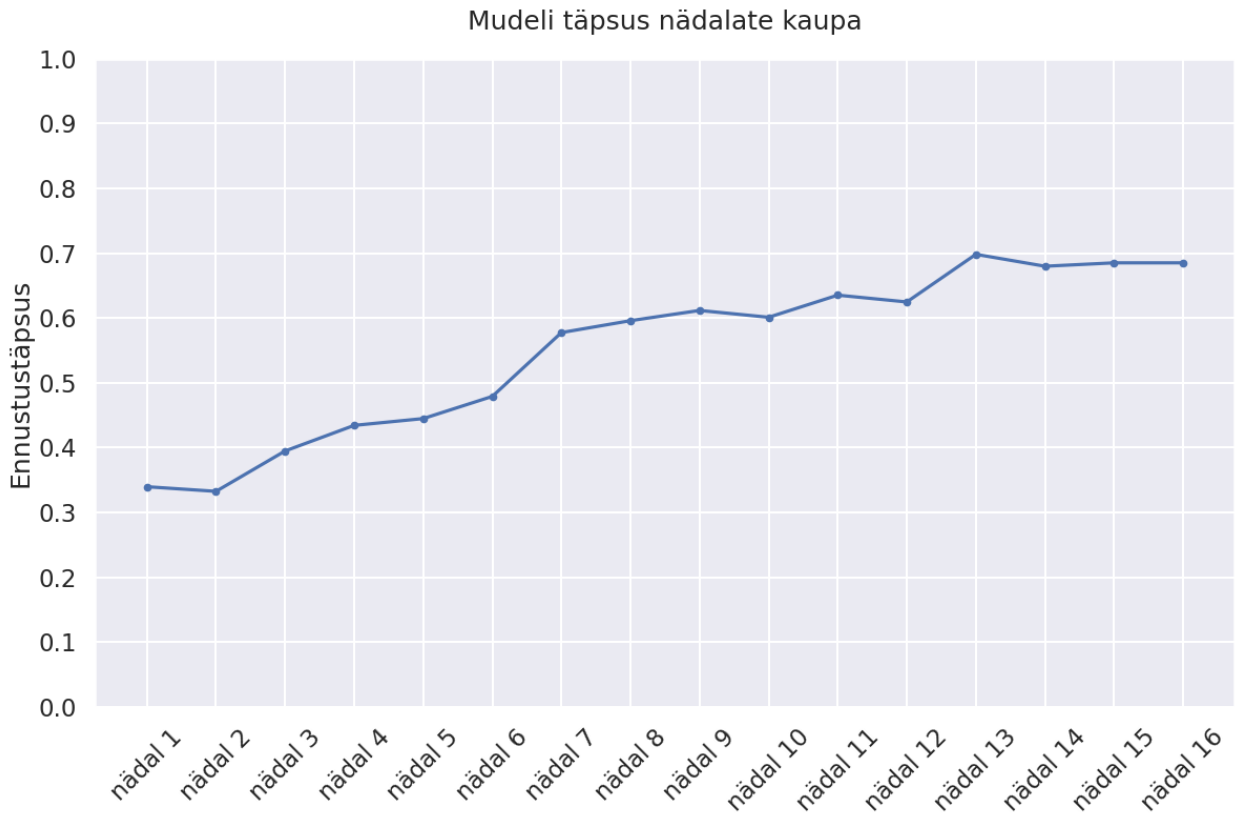


Joonis 9. Tööriista tulemuste leht.

4. Tööriista validatsioon

Järgnevalt tegeleme tööriista validatsiooniga.

Tööriista täielikult valideerida ei ole hetkel võimalik. Seda sellepärast, et tööriista ennustava mudeli loomiseks kasutati kolme aasta andmeid. Kui seda korrektselt valideerida, siis tuleks võtta uuel aastal uued andmed ja kasutada neid testandmetena. Kuna see pole võimalik, siis validatsiooni jaoks jagati olemasolevad kolme aasta treeningandmed kaheks. 33% andmetest said testandmed ja ülejäänud andmeid kasutati treeninguks. Tabelis 2 on näidatud sellise mudeli ennustustäpsust, kasutades mõõteühikuks keskmist täpsuse viga (*mean accuracy error*). See näitab, kui palju keskmiselt erines ennustatav tulemus päristulemusest. Joonisel 10 on näidatud ennustusmudeli täpsus, kui kasutada kogutulemuse asemel ainult hindeid skaalal A-F. Kuna mudel oli ülesse ehitatud regressioonile, mis ennustab numbrilist tulemust, siis see tulemus on madalam. Kui oleks ehitatud klassifitseerimismudel, siis oleks see tulemus teine.

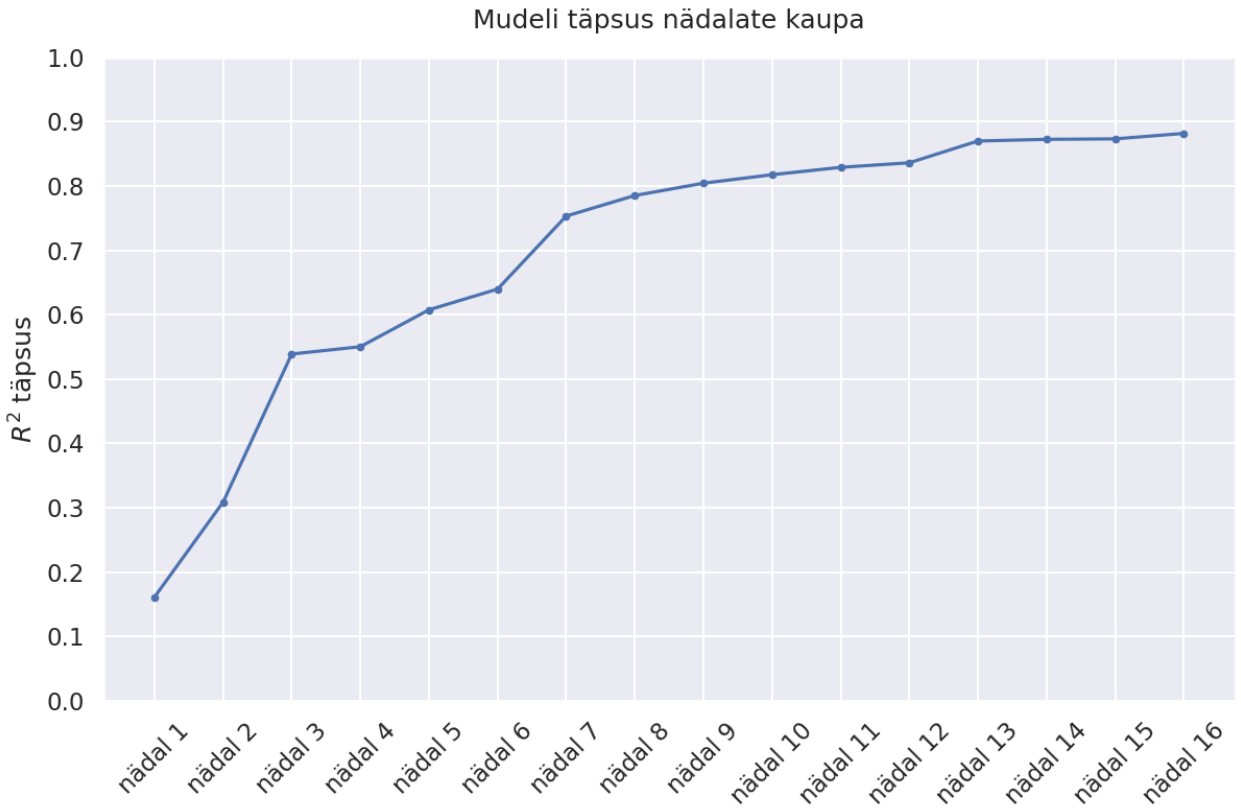


Joonis 10. Tööriista ennustusmudeli täpsus kasutades võrdlemiseks hindeid skaalas A-F.

4.1 Mudeli tulemuste võrdlus

Võrdleme mudeli ennustusi peatükis 1.2 kirjeldatud mudelitega.

Rühma P05 ennustustäpsus on näidatud joonisel 2, kus mõõdeti mudeli täpsust kasutades R^2 täpsust (*R^2 score*). Tööriista ennustusmudeli R^2 täpsust on näha joonisel 11. Rühma P05 mudel oli hilisematel nädalatel natuke täpsem kui see mudel. Samas oli selle tööriista mudel varasematel nädalatel täpsem. Kuna töö eesmärgiks oli võimalikult varakult leida raskustes tudengeid, siis võib öelda, et see mudel on selle ülesande jaoks efektiivsem.



Joonis 11. Tööriista ennustusmudeli R^2 täpsus nädalate kaupa.

Rühm D3 kasutas oma ennustusmudeli hindamiseks keskmist täpsusviga, tulemused on tabelis 1. Neid tulemusi saab kergesti võrrelda selle töö mudeliga, mille peamine hindamisviis oli samuti keskmine täpsusviga, mille tulemused on tabelis 2. Esimene otsavaatamine neile tulemustele näitab, et rühma D3 juhusliku metsa algoritmiga mudel oli täpsem kui selle töö mudel. Siinkohal tuleks taaskord aga mainida, et rühm D3 tegi andmetöötluses vigu, mille tulemusel oli nende mudelil rohkem logi- ja tavaandmeid. Seetõttu on ka nende mudel täpsem kui see peaks olema. Kokkuvõtvalt võib öelda, et mudelid on umbes sama täpsed, kuid selle tööriista mudeli loomisel

oli kolm korda rohkem algandmeid. Kuna mõlemad mudelid kasutasid sama algoritmi, siis tekitab see mulje, et lisatud andmed ei paranda mudelite efektiivsust.

Ülejäänud rühmadelt ei võetud mudeli loomiseks inspiratsiooni, seega ei süveneta nende võrdlusesse selles töös sügavalt.

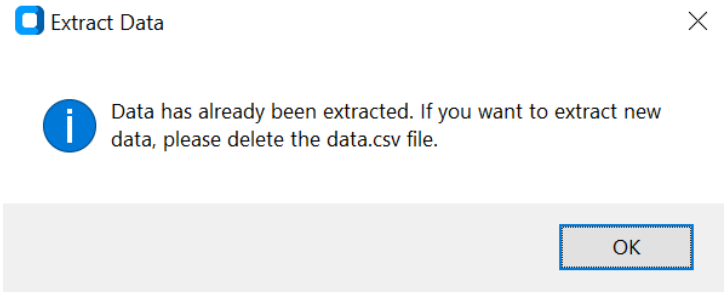
4.2 Kasutajaliidese validatsioon

Kasutajaliidese loomisel oli eesmärgiks luua lihtne ja mugav tööriist, mida oleks kerge kasutada. Selle kasutajaliidese on heaks kiitnud „Programmeerimise“ [12] aine õppejõud.

4.2.1 Avalehe validatsioon

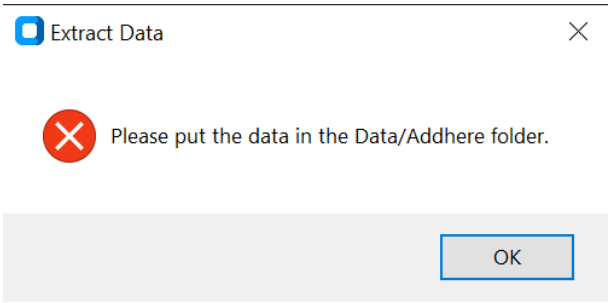
Avalehel on kolm funktsionaalset osa. Avalehte võib näha joonisel 8. Esiteks on võimalik valida nädal, mis on üsna selge. Järgmiseks on lehel kaks nuppu.

Esimene nupp on „Extract Data“. Vajutades sellele nupule juhtub üks kolmest stsenaariumist. Esimene võimalus on see, et kaustas „Addhere“ on juba olemas töödeldud andmed, mille peale programm teavitab kasutajat. Kuna andmete töötlus on ajamahukas protsess, siis aitab see protsess kaitsta kasutajat ajamahuka vea eest. Teavitus on näha joonisel 12.



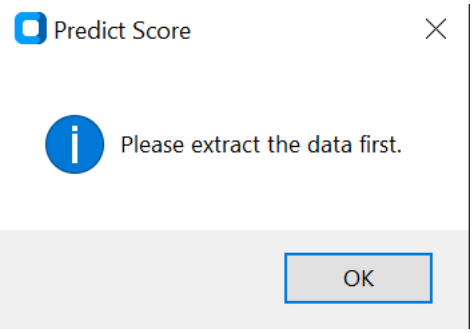
Joonis 12. Teavitus juba töödeldud andmetest nupule „Extract Data“ vajutamisel.

Teine võimalus on see, et kaustas pole töötlemiseks vajalikke andmeid, ka sel juhul teavitab programm kasutajat. Teavitus on näha joonisel 13. Kolmandal juhul alustab programm andmete töötlust.



Joonis 13. Teavitus andmete puudumise kohta nupule „Extract Data“ vajutamisel.

Teine nupp on „Predict Scores“ nupp. Vajutades sellele nupule juhtub üks kahest stsenaariumist. Kui andmed on töödeldud ja õiges kaustas, siis programm ennustab tulemused ja läheb järgmisele lehele. Kui aga andmeid ei ole kaustas ja seda nuppu vajutada, siis programm teavitab kasutajat. Teavitust on näha joonisel 14.



Joonis 14. Teavitus andmete puudumise kohta nupule „Predict Scores“ vajutamisel.

4.2.2 Tulemuste lehe validatsioon

Tulemuste lehel on näha tudengite nimed ja tulemused. Ainus funktsionaalsus on nimekirja sorteerimine, mis töötab korrektselt. Tulemuste lehte on võimalik näha joonisel 9.

5. Kokkuvõte ja analüüs

Selles peatükis arutletakse natuke tehtud töö ja selle käigus leitud huvitavate avastuste üle.

5.1 Lisatud andmete mõju mudelile

Tööriista validatsiooni ajal tuli välja huvitav fakt. Kui võrrelda tööriista mudelit lõpuprojektide mudelitega, avastati, et tööriista ennustusmudel ei olnud väga palju täpsem, kui nendel. Mis teeb asja huvitavaks, on see, et selle tööriista loomisel oli treeningandmeid kolm korda rohkem. Sellest võib järeldada üht kahest. Esiteks, kas uued treeningandmed olid väga sarnased eelnevatele treeningandmetele ja seetõttu nende lisamine ei parandanud mudeli täpsust märkimisväärselt.

Teine variant on see, et antud ülesanne, tudengi lõpptulemuse ennustamine, ei saagi olla mingist piirist täpsem ükskõik kui mitme aasta andmeid lisada. See variant tähendaks, et tudengi hinnetest ja logitegevustest saab mingisuguse arusaama, kui hoolikas tudeng on, kuid see oleks ka kõik. Nende andmetega saab teha mingisuguse ennustuse, kuid mida varasem nädal, seda vähem andmed tegelikult lõpptulemust mõjutavad. Seda näitab ka fakt, et kui tudeng jätaks kõik paari esimese nädala hindelised ülesanded tegemata, siis mudel ennustaks tema läbikukkumist. Kuid tegelikult on väga võimalik läbida kursus ka ainult hilisemate tulemuste põhjalt. Kuna aine „Programmeerimine“ [12] on esimesele kursusele mõeldud aine, siis ei ole seal läbitavad teemad eriti keerulised. See tähendab, et tudeng suudab igal kursuse hetkel end kokku võtta ja natuke rohkem pingutades oma tulemust märkimisväärselt parandada.

5.2 Tööriista puudused ja edasiareng

Loodud tööriistal on üks peamine puudus, milleks on tulevikukindluse puudus. Tööriista loomisel töötati kolme aasta andmetega. Selle käigus nähti, et kõigi kolme aasta andmed erinesid üksteisest erinevatel määradel. See tähendab, et arenduse käigus tuli kõigi aastate andmed viia samale kujule. Andmete töötlusesse on ehitatud hulk veakontrolle ja andmete ühildamist, kuid kui andmed peaksid suuremal määral muutuma, siis oleks vaja manuaalselt andmete töötlust ümber teha. See oleks aga kasutajale tüütu.

Antud tööriista on võimalik edasi arendada mitmel viisil. Mudeli loomisel otsiti juhusliku metsa algoritmile (*random forest regressor*) lisaks mõnda muud algoritmi, mis suudaks samuti edukalt tulemusi ennustada. Kui tahta seda tööriista edasi arendada, siis kõige mõistlikum oleks esiteks leida mõni algoritm või mudel, mis suudaks anda sama täpseid või täpsemaid tulemusi, kui

juhuliku metsa algoritm. See oleks kasulik, kuna andes kasutajale kahe erineva algoritmiga saadud tulemused, annab see parema usalduskindluse. Kui kahe algoritmi tulemused on sarnased, siis kasutaja teab, et tulemus on usaldusväärne. Kui need suurel määral erinevad, siis saab kasutaja seda tulemust vaadata suurema skepsisega.

5.3 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua ennustusmudel, millega saaks ennustada programmeerimise aine tudengite lõpptulemusi. Mudel pidi olema võimeline ennustama tulemusi dünaamiliselt igal kursuse hetkel ja saavutama võimalikult varakult võimalikult hea täpsuse. Samuti oli eesmärgiks ehitada selle mudeli peale üles kasutajaliides, millega kasutaja saaks mugavalt ja lihtsalt ennustada tulemusi.

Käesoleva bakalaureusetöö käigus valmis ennustusmudel. Mudeli algoritmiks valiti juhusliku metsa algoritm (*random forest regressioon*). Mudel realiseeriti nädalate kaupa niimoodi, et mudel suudaks ennustada kuueteistkümnel kursuse esimesel nädalal tudengite lõpptulemusi. Lisaks ehitati kasutajaliides, mille abil on kerge töödelda uusi andmeid ja ennustada tudengite tulemusi.

Töö tulemusena said püstitatud eesmärgid täidetud ning valminud tööriista abil on kindlasti võimalik programmeerimise aine õppejõududel tuvastada juba varakult tudengeid, kel võib esineda aine lõpetamisega raskusi.

Kasutatud allikad

[1] Diego Buenaño-Fernández, David Gil and Sergio Luján-Mora. 2019. Application of Machine Learning in Predicting Performance for Computer Engineering Students: A Case Study. https://www.mdpi.com/2071-1050/11/10/2833 (07.05.2023)

[2] Siti Dianah Abdul Bujang, Ali Selamat, Roliana Ibrahim, Ondrej Krejcar. 2021. Multiclass Prediction Model for Student Grade Prediction Using Machine Learning. https://ieeexplore.ieee.org/document/9468629 (07.05.2023)

[3] Lidia Feklistova. 2022. Learners of an introductory programming MOOC: background variables, engagement patterns and performance. https://dspace.ut.ee/handle/10062/88052 (07.05.2023)

[4] Raza Hasan, Sellappan Palaniappan, Abdul Rafiez Abdul Raziff, Salman Mahmood and Kamal Uddin Sarker. 2018. Student Academic Performance Prediction by using Decision Tree Algorithm. https://ieeexplore.ieee.org/abstract/document/8510600 (07.05.2023)

[5] Zafar Iqbal, Junaid Qadir, Adnan Noor Mian, and Faisal Kamiran. 2017. Machine Learning Based Student Grade Prediction: A Case Study. https://arxiv.org/abs/1708.08744 (07.05.2023)

[6] Ksenia Koroljova. 2022. IKT eriala üliõpilaste varajase väljalangemise ennustamise veebirakenduse loomine R Shiny abil. https://digikogu.taltech.ee/et/Item/7701f103-53e5-4419-9a9e-5bf3290ce0c6 (07.05.2023)

[7] MasterDataScience with etX. 2023. What Is a Decision Tree? https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/#:~:text=A%20decision%20tree%20is%20a,that%20contains%20the%20desired%20categorization. (07.05.2023)

[8] Heleriin Ots. 2020. Õpisoorituse ennustamine Moodle'i logiandmete ja enesehinnanguliste õppimisega seotud psühholoogiliste tegurite põhjal. https://digikogu.taltech.ee/et/Item/2ddcb69a-27d9-492c-8c51-886ad60e3478 (07.05.2023)

[8] Juan L. Rastrollo-Guerrero, Juan A. Gómez-Pulido and Arturo Durán-Domínguez. 2020. Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review. https://www.mdpi.com/2076-3417/10/3/1042 (07.05.2023)

[10] Simplilearn. 2023. Random Forest Algorithm. https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm (07.05.2023)

[11] Tartu Ülikooli kursus "Masinõpe (MTAT.03.227, 6 EAP)", 2021 sügis. https://ois2.ut.ee/#/courses/MTAT.03.227/version/ac4dc403-b0fc-2c2c-fa58-621da6b4f38b/details (07.05.2023)

[12] Tartu Ülikooli kursus “Programmeerimine“ (LTAT.03.001, 6 EAP). https://ois2.ut.ee/#/courses/LTAT.03.001/version/a9559a3a-2ad1-2829-52ff-dc8a9ea5ad79/details (07.05.2023)

[13] Tartu Ülikooli kursus "Sissejuhatus andmeteadusesse (LTAT.02.002, 6 EAP)", 2021 sügis. https://ois2.ut.ee/#/courses/LTAT.02.002/version/cbeed4b1-2ada-bcee-7752-8e122b3d0de6/details (07.05.2023)

Lisad

I. Valminud tööriista lähtekood

Lähtekood asub lingil: https://github.com/uku20/Balakaureuse-Tooriist.git

See on privaatne, nii et kui tahta juurdepääsu, palun kirjutada meilile ukuzingel@gmail.com.

Tööriista installeerimisjuhend on lingil olevas failis nimega „README.md“.

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Uku Zingel (sünnikuupäev: 12.11.2000),

1. Annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Õppijate tulemuste ennustamise tööriist“,

mille juhendaja on Reimo Palm, reprodutseerimiseks eesmärgiga seda säilitada,

sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks

Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative

Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost

reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja

kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega

isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Uku Zingel

Tartus, 09.05.2023