Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Matemaatika ja statistika instituut

Johanna Õun

Raviarvete tagasilükkamise põhjuste tuvastamine

Matemaatilise statistika eriala
Bakalaureusetöö (9 EAP)

Juhendaja Sven Laur

Tartu 2020

Raviarvete tagasilükkamise põhjuste tuvastamine

Bakalaureusetöö

Johanna Õun

Lühikokkuvõte

Bakalaureusetöös uuritakse, milliste reeglite alusel lükkab Eesti Haigekassa raviarveid tagasi. Töö esimeses peatükis tehakse teoreetiline ülevaade klassifitseerimispuudest ning mudeli treenimisest. Teises peatükis antakse ülevaade Eesti Haigekassa andmestikust ning otsustusprotsessist. Kolmandas peatükis antakse ülevaade tagasilükatud ning hüvitatud arvete jaotusest. Neljas peatükk keskendub tagasilükkamise põhjuste leidmisele.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika.

Märksõnad: Klassifitseerimispuu, täpsus, saagis, F_1-skoor.

The identification of reasons for rejection of medical bills

Bachelor thesis

Johanna Õun

Abstract

This Bachelor's thesis focuses on the rules for which the Estonian Health Insurance Fund refuses payment for medical bills. In the first chapter, the author gives a theoretical overview of the classification trees and the creation of the method used in this thesis. The second chapter gives an overview of the decision process of the Estonian Health Insurance Fund and the data behind it. The third chapter focuses on the balance between accepted and rejected medical bills by the Estonian Health Insurance Fund. The final chapter is used for finding the reasons for the rejection of the medical bills. .

CERCS research specialisation: P160 Statistics, operations research,

programming, financial and actuarial mathematics.

Key Words: Classification tree, precision, recall, F_1-score.

Sisukord

Sissejuhatus **5**

1 Metoodika **6**

1.1 Otsustuspuud 6

1.2 Klassifitseermispuud 6

1.3 ID3 algoritm 7

1.4 Teised otsustuspuu treenimise algoritmid 8

1.5 Pakett *DecisionTreeClassifier* 8

1.6 Andmestiku tasakaalustamine 9

1.7 Ristvalideerimine 9

1.8 Mudeli hindamine 10

2 Eesti Haigekassa raviarved **11**

2.1 Otsustusprotsess 11

2.2 Andmestiku ülevaade 11

3 Tagasilükatud raviarvete esmane analüüs **13**

3.1 Tunnuste jaotus tagasilükatud arvete korral 14

3.2 Tunnuste jaotus kõikide arvete korral 17

4 Tagasilükkamise põhjuste leidmine **18**

4.1 Enim esinenud järjendid 19

4.1.1 Võrdsete väärtustega lehtede uurimine 23

4.1.2 Mudeli kontroll testandmestikul 24

4.1.3 Treeningandmete ja testandmete tulemuste võrdlus 25

4.2 100 kuni 1000 korda esinenud järjendid 27

Kokkuvõte **32**

Kasutatud kirjandus **33**

Lisad **34**

Lisa 1. Teise lehe tingimuste põhjal kõikidest andmetest ning tunnustest loodud klassifitseerimispuu. 34

Lisa 2. Enim esinenud järjendite andmestikul treenitud klassifitseerimispuu. 35

Lisa 3. Kasutatud kood parima puu sügavuse leidmiseks ristvalideerimise meetodil. 37

Sissejuhatus

Eesti Haigekassa on Eestis riiklikku ravikindlustust haldav organisatsioon. Üheks haigekassa ülesandeks on raviasutusele osutatud tervishoiuteenuste eest tasumine. Tervishoiuteenuse osutajad (TTO-d) esitavad haigekassale arved esitatud teenuste eest. Arves on ära näidatud patsiendi info, diagnoos, TTO kohane informatsioon ning teenuse maksumus. Igal aastal esineb paar tuhat korda, mil haigekassa hüvitab arve maksumuse osaliselt või jätab täielikult hüvitamata. Reeglid, mille põhjal arve hüvitamise otsus vastu võetakse, pole kunagi selgelt ära dokumenteeritud.

Selles bakalaureusetöös tegeletakse antud reeglite tuvastamisega kasutades haigekassa raviarveid aastatest 2010-2018. Meetodina kasutatakse klassifitseerimispuid, mis on loodud kasutades teegi Scikit-learn paketti DecisionTree-Classifier ning on visualiseeritud GraphViz paketi abil.

Töö põhiosa on jaotatud nelja peatükki. Esimeses peatükis antakse teoreetiline ülevaade klassifitseerimispuudest. Teises peatükis antakse lühiülevaade haigekassa otsustusprotsessist ning kasutatud andmestikust. Kolmas peatükk keskendub raviarveid iseloomustavate tunnuste jaotuste uurimisele. Neljas peatükk on jagatud kaheks. Esimeses osas rakendatakse klassifitseerimispuu algoritmi enim esinenud ning teises osas 100 kuni 1000 korda esinenud järjenditega arvetele.

Töö on kirjutatud küljendussüsteemi LaTeX liideses Overleaf. Analüüs on läbiviidud keskkonnas Jupyter, kasutades programmeerimiskeelt Python, andmebaasisüsteemi PostgreSQL ja masinõppe teeki Scikit-learn.

Bakalaureusetöö autor tänab Sven Lauri panustatud aja ning paranduste eest.

1 Metoodika

1.1 Otsustuspuud

Järgnev alapeatükk põhineb allikal [1].

Otsustuspuid kasutatakse regressioon- ja klassifitseerimisülesannete lahendamiseks. Tulemuseks on funktsioon, mida on kujutatud puuna ning mis koosneb tippudest ja harudest. Iga tipp kujutab kindlat tingimust ning sellest langev haru tingimusest saadud väärtust. Otsustuspuu algab algustipust ehk juurest ning selle juures olevast tingimusest. Tulemuse põhjal hargneb juur edasi harudeks ning tippudeks, kuni jõutakse lõpptipu ehk leheni. Leht kujutab saadud lõplikku väärtust või klassifikatsiooni.

1.2 Klassifitseermispuud

Järgnevas peatükis on kasutatud allikat [2].

Klassifitseerimispuu eesmärk on vaatluste klassifitseerimine, milleks jagatakse vaatlused uuritavate tunnuste $X_1, X_2, ..., X_p$ põhjal J erinevasse mitte lõikuvasse klassi $R_1, R_2, ..., R_J$. Puu koostamiseks valitakse tunnus X_j ja lõikepunkt s nii, et tekkivad piirkonnad $R_{j_1} = \{X|X_j < s\}$ ja $R_{j_2} = \{X|X_j \geq s\}$ annaksid suurima kasu. Diskreetse järjestatava tunnuse X_j korral kodeeritakse selle väärtused järjestusele vastavalt järjestikeks numbriteks ning kategoorilise X_j korral luuakse väärtustest fiktiivsed tunnused (*dummy variables*). Kasu mõõtmiseks kasutatakse, kas klassifitseerimisviga, Gini indeksit või entroopiat.

Klassifitseerimisviga (*classification error rate*) näitab piirkonna R_j vaatluste osakaalu, mis ei kuulu selle piirkonna enim levinud klassi ning on leitud järgnevalt:

$$E = 1 - \max_k(\hat{p}_{mk})$$

Suurus $\hat{p}_{mk}$ näitab piirkonna R_m vaatluste arvu, mis kuulvad k klassi ehk klassi, mis ei ole piirkonna R_m enim levinud klass.

Lisaks kasutatakse Gini indeksit (*Gini index*), mis mõõdab varieeruvust üle K klassi:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}).$$

Gini indeks tuleb väike, kui $\hat{p}_{mk}$ väärtused on nulli ja ühe lähedased. Seetõttu nimetatakse Gini indeksit lehe puhtuse näitajaks.

Veel on võimalik kasu kirjeldamiseks mõõta entroopiat (*entropy*):

$$H = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk},$$

kus $0 \leq \hat{p}_{mk} \leq 1$, millest tuleneb, et $0 \leq -\hat{p}_{mk} \log \hat{p}_{mk}$.

1.3 ID3 algoritm

Järgnev alapeatükk põhineb allikal [1].

ID3 on otsustuspuu algoritm, mille töö põhimõte on leida igas tipus kõige kasumlikum tunnus, mida testida. Kasumlikust mõõdetakse, kas informatsiooni kasu (*information gain*) või entroopia järgi ning tunnused võivad olla nii pidevad kui ka diskreetsed. Tunnuse A valimisel vaatluste S seas on informatsiooni kasu defineeritud järgnevalt:

$$Kasu(S, A) = H(S) - \sum_{v \in vaartused(A)} \frac{|S_v|}{|S|} \cdot H(S_v),$$

kus $vaartused(A)$ on tunnuse A kõik võimalikud väärtused, H tähistab entroopiat ning $S_v = \{s \in S | A(s) = v\}$. Protsessi korratakse igas tipus, kuni jõutakse, kas olukorrani, kus kõik tunnused on juba ära kasutatud või kõik vaatlused antud tipus on entroopia poolest nullid.

1.4 Teised otsustuspuu treenimise algoritmid

Lisaks ID3 algoritmile on kasutusel veel palju teisi, näiteks C4.5 (kõige populaarsem praktikas), C5 ja CART.

C4.5 on variant ID3 algoritmist, mis suudab kasutada ennustamiseks nii pidevaid kui diskreetseid tunnuseid. Sarnaselt ID3 algoritmile kasutab C4.5 entroopiat, kuid vastupidiselt ID3-le töötab C4.5 algoritm ka puuduvate väärtuste korral [3].

C5 algoritm on C4.5 algoritmi järglane, mis võimaldab luua väiksemaid ning lihtsamaid otsustuspuid. Näiteks suudab C5 puu treenimisel erinevate kategooriate mittekorduvaid tingimusi omavahel kombineerida [4].

CART (*Classification And Regression Tree*) algoritm kasutab kasu mõõtmiseks Gini indeksit ning suudab ennustada ka pidevaid tunnuseid [3].

1.5 Pakett *DecisionTreeClassifier*

Järnev alapeatükk põhineb allikal [5], kui ei ole viidatud teisiti.
Kategoorilise tunnuse ennustamiseks saab kasutada Scikit-learn-i paketti *DecisionTreeClassifier*, mis kasutab puu loomiseks optimiseeritud CART algoritmi. Kategooriliste tunnuste kasutamiseks tuleb nende tasemed eelnevalt eraldi fiktiivseteks tunnusteks viia, kus üks fiktiivne tunnus näitab ühe kategooria taseme väärtuse olemasolu (väärtus 1) või puudumist (väärtus 0) [6]. Puu võtab sisendiks uuritavate tunnuste vektorid $x_i \in R^n$, kus $i = \{1, .., l\}$ ja ennustatava klassi väärtuste vektori $y \in R^\ell$, kus uuritavad tunnused on pidevad või binaarsed. Tipus m iga võimaliku tükelduse $\theta = (j, t_m)$ korral jaotatakse andmehulk Q osadeks $Q_{vasak}(\theta)$ ja $Q_{parem}(\theta)$ järgnevalt:

$$Q_{vasak}(\theta) = (x, y) \mid x_j \leq t_m$$

$$Q_{parem}(\theta) = Q \setminus Q_{vasak}(\theta),$$

kus j tähistab tunnust ning t_m selle läviväärtust. Tipu tükelduseks valitakse θ, mille korral on

$$G(Q, \theta) = \frac{n_{vasak}}{N_m} \cdot H(Q_{vasak}(\theta)) + \frac{n_{parem}}{N_m} \cdot H(Q_{parem}(\theta))$$

minimaalne, kus H tähistab entroopiat ning N_m tipus m olevate vaatluste arvu. Samamoodi jätkatakse järgmistes tippudes, kuni on jõutud kas maksimaalse lubatud puu sügavuseni, tipus on vähem vaatlusi kui lubatud või on jõutud olukorda, kus tipus on ainult üks vaatlus.

1.6 Andmestiku tasakaalustamine

Alapeatükis on kasutatud allikat [3].

Andmestiku tasakaalustatakse, kui ennustatava tunnuse väärtuste jaotus on liialt erinev. Tasakaalustamiseks saab kasutada alavalimist (*undersampling*) või ülevalimist (*oversampling*). Käesolevas bakalaureusetöös kasutati efektiivsuse huvides treeningvalimi alavalimist, mille jaoks leitakse tunnuse enamus- ja vähemusgrupp ning võetakse juhuslikult enamusgrupist tagasipanekuta vaatlusi nii, et neid oleks võrdselt vähemusgrupiga.

1.7 Ristvalideerimine

Järgnevas alapeatükis on kasutatud allikat [2], kui ei ole märgitud teisiti.

Mudeli täpsuse hindamiseks kasutatakse k-kordset ristvalideerimist (*k-fold cross-validation*), mille käigus jagatakse juhuslikult kõik vaatlused k võrdseks osaks (enamasti $k = 10$). Esimest osa kasutatakse mudeli valideerimiseks ning ülejäänud $k-1$ osa mudeli sobitamiseks. Protsessi korratakse k korda, mil igal korral leitakse valesti klassifitseeritud vaatluste arv ning lõpliku mudeli veaks võetakse nende keskmine:

$$mudeliviga_{(k)} = \frac{1}{k} \sum_{i=1}^{k} I(y_i \neq \hat{y}_i) \ ,$$

kus $I(y_i \neq \hat{y}_i)$ on võrdne nulli või ühega, vastavalt kas vaatlus on õigesti või valesti klassifitseeritud. Ristvalideerimisest on mitmeid modifikatsioone, millest üks on *stratified k-fold cross-validation*, kus andmestiku osadeks jaotamisel jäetakse tunnuste jaotus osade vahel võimalikult võrdseks. Käesolevas bakalaureusetöös kasutati *stratified* 10-kordset ristvalideerimist Scikit-learn-i paketi „StratifiedKFold“ [1] abil.

1.8 Mudeli hindamine

Selles peatükis on kasutatud allikat [3].

Mudeli ennustamisvõime hindamiseks kasutatakse eksimismaatriksit (*confusion matrix*), kui prognoositakse binaarset tunnust (tabel 1).

Tabel 1. Eksimismaatriks binaarse prognoositava tunnuse korral

Prognoos/Tegelik	Sündmus toimus	Sündmus ei toimunud
Sündmus toimus	õiged positiivsed	valepositiivsed
Sündmus ei toimunud	valenegatiivsed	õiged negatiivsed

Eksimismaatriksi põhjal leitakse mudeli täpsus (*precision*) ja saagis (*recall*) järgnevalt:

$$\text{täpsus} = \frac{\text{õiged positiivsed}}{\text{õiged positiivsed} + \text{valepositiivsed}}$$

$$\text{saagis} = \frac{\text{õiged positiivsed}}{\text{õiged positiivsed} + \text{valenegatiivsed}}$$

[1] Pakett "StratifiedKFold" `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html`

Täpsuse ja saagise põhjal leitakse mudeli F_1-skoor:

$$F_1\text{-skoor} = 2 \cdot \frac{\text{täpsus} \cdot \text{saagis}}{\text{täpsus} + \text{saagis}} \ ,$$

mis näitab täpsuse ja saagise harmoonilist keskmist.

Lisaks saab mudelit hinnata ka klassifitseerimistäpsuse (*classification accuracy*) kaudu:

$$\text{klassifitseerimistäpsus} = \frac{\text{õiged positiivsed} + \text{õiged negatiivsed}}{\text{kõigi vaatluste arv}}$$

Kõikide nimetatud suuruste väärtused jäävad vahemikku $[0, 1]$ ning mida lähemal on väärtus ühele, seda parem on mudel.

2 Eesti Haigekassa raviarved

2.1 Otsustusprotsess

Selles alapeatükis on kasutatud allikat [7]. TTO esitab haigekassale tervishoiuteenuse osutamise eest arve, mis võib koosneda ühest või enamast reast, kus iga rida kujutab patsiendile osutatud tervishoiuteenust. Haigekassa otsustab iga rea kohta, kas sellel kirjeldatud tervishoiuteenus hüvitatakse TTO-le või mitte. Haigekassa on kirja pannud põhilised reeglid, mille põhjal käib otsustusprotsess. Näiteks ei tohi arve olla vigaselt täidetud, iga TTO tohib osutada ainult kindlaid tervishoiuteenuseid, jne.

2.2 Andmestiku ülevaade

Töös kasutatud andmestikuks on haigekassalt saadud raviarved aastatest 2010-2018, kus iga arverea kohta oli näha otsus selle hüvitamise kohta (tunnus *tagasinõue*). Lisaks kasutati kahte iseloomustavate tunnustega andmestikku, millest üks kirjeldas TTO-d ning teine osutatud teenust. Samuti oli teada patsiendi sugu ning

vanusegrupp. Kõik andmestikud olid omavahel seotavad arve ID ja aasta kaudu. Kokku kasutati andmestikest järgnevaid tunnuseid:

- aasta,
- arve ID,
- arve tüüp,
- elukohakood,
- EMO arve,
- põhidiagnoos,
- põhieriala,
- ravi pikkus,
- ravitüüp,
- saatja eriala,
- sugu,
- tagasinõue,
- teenuse kategooria,
- teenuse kood,
- TTO asukoht,
- TTO-kood,
- TTO-tüüp,
- vanusegrupp,
- väljakirjutamise staatus,
- vältimatu abi arve.

Kokku oli kasutatud andmestikus üle 11 miljoni rea, mille aastate kaupa jagunemine on näidatud tabelis 2.

Tabel 2. Tagasilükatud ja hüvitatud arvete hulk ning tagasilükatud arvete protsent aasta kaupa

Aasta	Hüvitatud	Täielikult tagasilükatud	Osaliselt tagasilükatud	Kokku tagasilükatud	Tagasilükatud arvete %
2010	1 414 351	204	86	290	0,0205
2011	1 512 076	549	119	668	0,0442
2012	1 469 141	666	155	821	0,0559
2013	1 471 430	624	147	771	0,0524
2014	525 312	833	451	1 284	0,2438
2015	1 479 758	1 484	1 132	2 616	0,1765
2016	1 459 507	1 308	234	1 542	0,1055
2017	1 324 468	11 440	64	11 504	0,1134
2018	1 247 588	375	38	413	0,0331

3 Tagasilükatud raviarvete esmane analüüs

Analüüsi alustati raviarvete uurimisega, kus vähemalt ühel real oli toimunud tagasilükkamine ehk positiivne tulemus. Kõikidest sellistest raviarvetest kokku moodustati positiivsete näidetega andmestik. Loodi uus binaarne tunnus *kõik_tagasi*, mis näitas, kas rida kuulus arvesse, kus kõik read olid tagasilükatud (väärtus 1) või mitte (väärtus 0), et võrrelda teiste tunnuste jaotust kahes eraldi grupis. Jaotuste erinevus näitaks, millised tunnused võivad tagasilükkamist kõige rohkem mõjutada. Positiivsete näidete hulgas oli täielikult tagasilükatud ridu ligi 3 korda vähem kui osaliselt tagasilükatud ridu (tabel 3).

Tabel 3. Osaliselt ja täielikult tagasilükatud raviarvete jaotus.

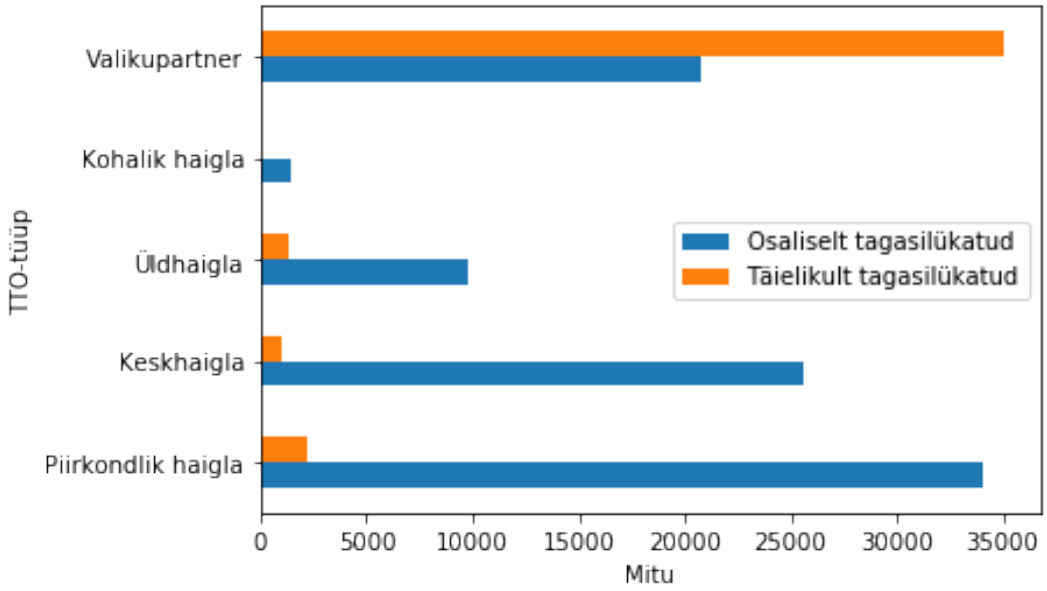
Kõik_tagasi	Mitu
Osaliselt tagasilükatud	91 630
Täielikult tagasilükatud	39 560

3.1 Tunnuste jaotus tagasilükatud arvete korral

TTO puhul oli üheks tähtsaimaks tunnuseks TTO-tüüp, kuna paljud haigekassa poolt kirjapandud reeglid näitasid, milliseid tervishoiuteenuseid võib keegi osutada, et kõik oleks reeglitekohane. Eestis on TTO-d jagatud tüübilt 5 grupiks, kus

- piirkondlikud haiglad on SA Põhja-Eesti Regionaalhaigla, SA Tartu Ülikooli Kliinikum ja SA Tallinna Lastehaigla,
- keskhaiglad on AS Ida-Tallinna Keskhaigla, AS Lääne-Tallinna Keskhaigla, SA Ida-Viru Keskhaigla ja SA Pärnu haigla,
- üldhaiglad on erinevate maakondade haiglad,
- kohalikud haiglad on üldhaiglatest väiksemad lokaalse tähtsusega haiglad

ning kõik ülejäänud TTO-d kuuluvad valikupartnerite alla [8]. Valikupartnerite korral on täielikult tagasilükatud raviarvete hulk kõikide positiivsete näidete seas võrreldes teiste TTO-tüüpidega mitmekordselt suurem (tabel 4), mis võib tähendada, et nad pole haigekassa poolt kehtestatud reeglitega niivõrd hästi tuttavad või on nende hulgas palju uusi TTO-sid, kes on arve edastamisel haigekassale millegagi eksinud.



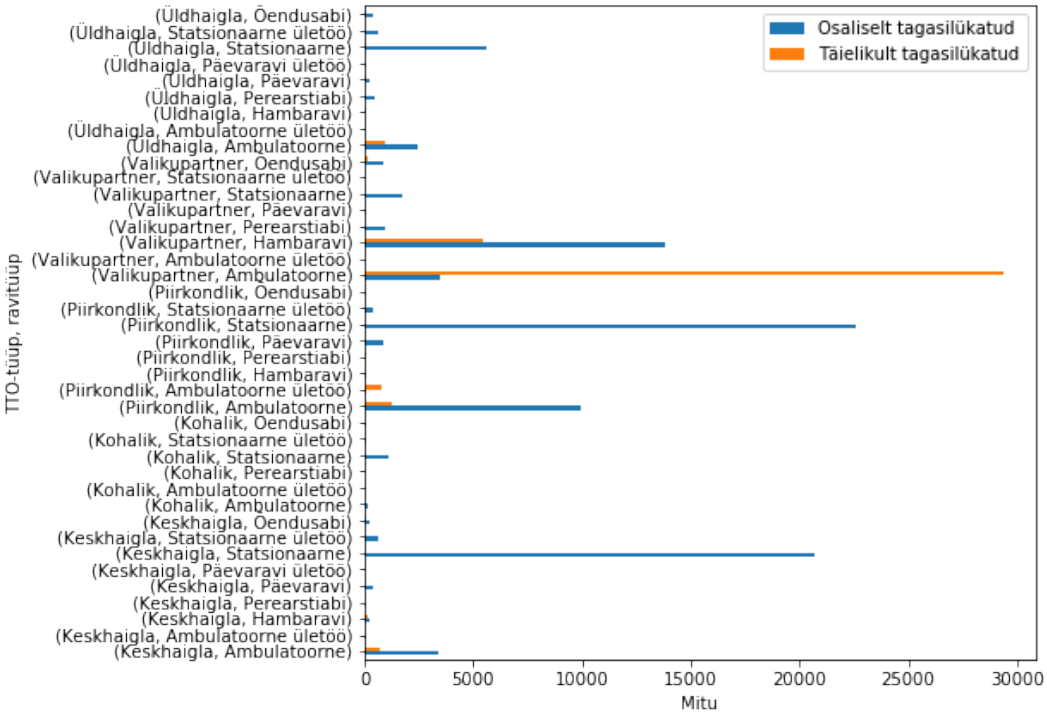
Joonis 1. TTO-tüübi jaotus sõltuvalt, kas arve lükati osaliselt või täielikult tagasi

Jooniselt 1 on näha, et kokkuvõttes on enim tagasilükatuid arveid valikupartnerite ja piirkondlike haiglate hulgas, millele järgnevad keskhaiglad, mis on oodatav, kuna nimetatud TTO-tüüpe on TTO-de hulgas kõige enam (tabel 4).

Tabel 4. Täielikult tagasilükatud arvete protsent kõikidest arvetest, kus mõnel real on tagasilükkamine toimunud TTO-tüübi kaupa ning TTO-tüübi esinemiste arv

TTO-tüüp	Mitu	Täielikult tagasilükatud arvete % kõikidest arvetest, kus mingi tagasilükkamine on toimunud
Piirkondlik haigla	3 058 331	6,10%
Keskhaigla	3 517 892	3,81%
Üldhaigla	1 547 962	11,55%
Kohalik haigla	55 894	2,51%
Valikupartner	2 970 048	62,74%

Joonis 2 näitab, et erinevate TTO-tüüpide ja ravitüübi korral on täielikult ja osaliselt tagasilükatud arvete jagunemine erinev, mis viitab sellele, et need tunnused võivad aidata reeglite tuvastamist ning tuleb puusse juurde võtta.

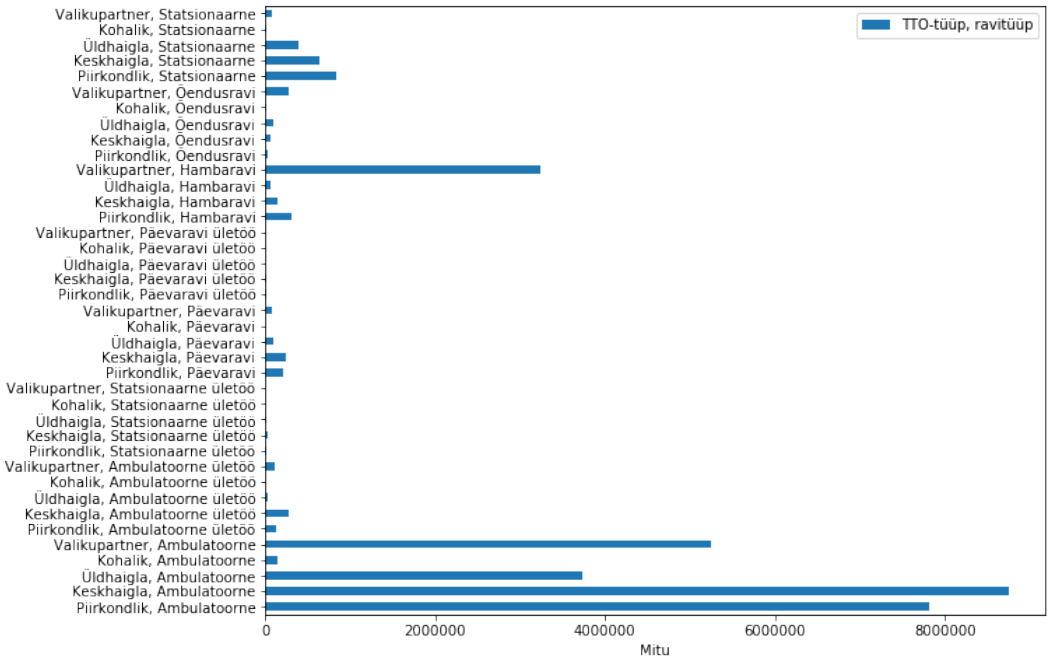


Joonis 2. TTO-tüübi ja ravitüübi jaotus sõltuvalt tunnusest *kõik_tagasi*

3.2 Tunnuste jaotus kõikide arvete korral

Haigekassa kogu andmebaasi tehti päring, et võrrelda, kas arvete jaotus, kus mingil real oli tagasilükkamine toimunud, erines kõikide arvete jaotusest. Vaadati samuti arveid aastatest 2010-2018. Selgus, et arvete jaotus TTO-tüübi ja ravitüübi kaupa on tagasilükatud arvete ning kõikide arvete seas erinev (joonis 3). Kui tagasilükatud arvete seas oli enim esinenud valikupartnerite ambulatoorseid arveid, siis kõikide andmete korral oli enim keskhaigla ambulatoorseid arveid, millele järgnesid piirkondlike haiglate ambulatoorsed arved. Jooniselt 2 nähtub, et tagasilükkamisi esines palju piirkondlike haiglate ning keskhaiglate statsionaarsete arvete seas, kuid joonisel 3 on näha, et neid arveid on võrreldes teistega siiski tunduvalt vähem. See võib tähendada olla põhjustatud sellest, et statsionaarset ravi (ravi haiglas ööbimisega) osutatakse vähem kui ambulatoorset ravi (arsti vastuvõtt), kuid selle

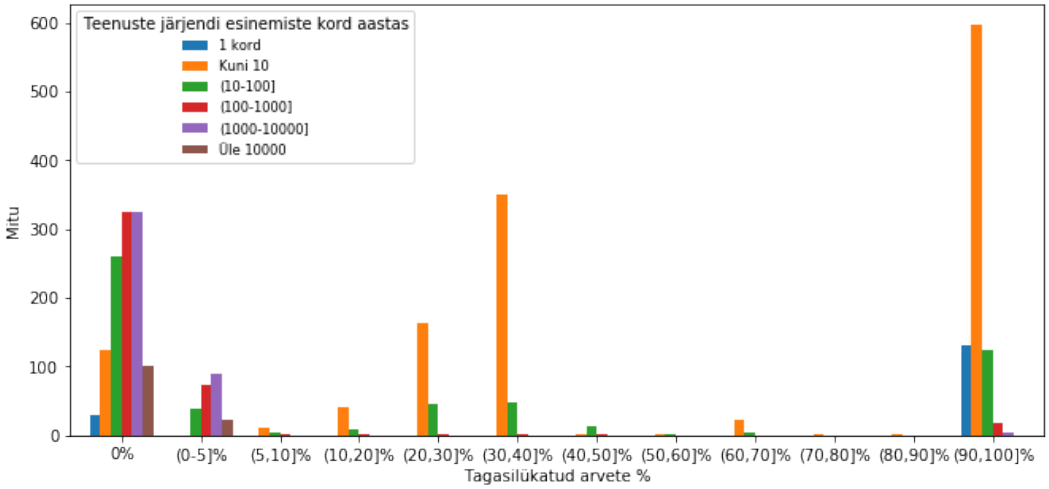
maksumus võib olla suurem ning hüvitamise tingimused karmimad.



Joonis 3. TTO-tüübi ja ravitüübi jaotus kõikide arvete korral

4 Tagasilükkamise põhjuste leidmine

Iga arve kohta leiti teenuste hulk ehk teenuste järjend, mis sisaldas kõiki teenuse koode ning koguseid, mida antud raviarve kajastas. Leides palju andmestikus igat järjendit esines ning rakendades sellele otsustuspuud, saab leida, mis tunnused on tagasilükkamist mõjutanud. On võimalik, et haigekassa on aastate jooksul mõningaid reegleid muutnud, mistõttu leiti kõik järjendite kordused aasta kaupa. Järjendid jagati nende esinemise arvu järgi kuueks rühmaks, millele leiti tagasilükkamiste %. Joonis 4 näitab gruppide tagasilükkamiste protsentuaalset jaotust.

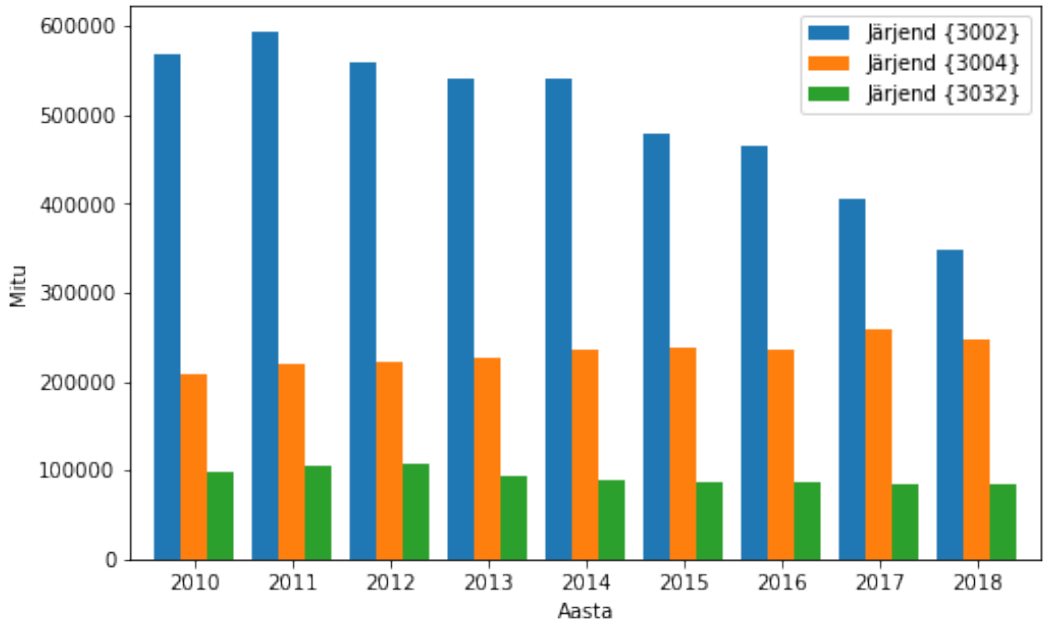


Joonis 4. Näidatud protsendil tagasilükatud arvete arv, kus teenuste järjendid on grupeeritud vastavalt nende esinemiste arvule aastas

Lisaks grupeeriti olemasolevad põhidiagnoosid RHK (*Rahvusvaheline haiguste klassifikatsioon*) järgi diagnoosigruppidesse [9].

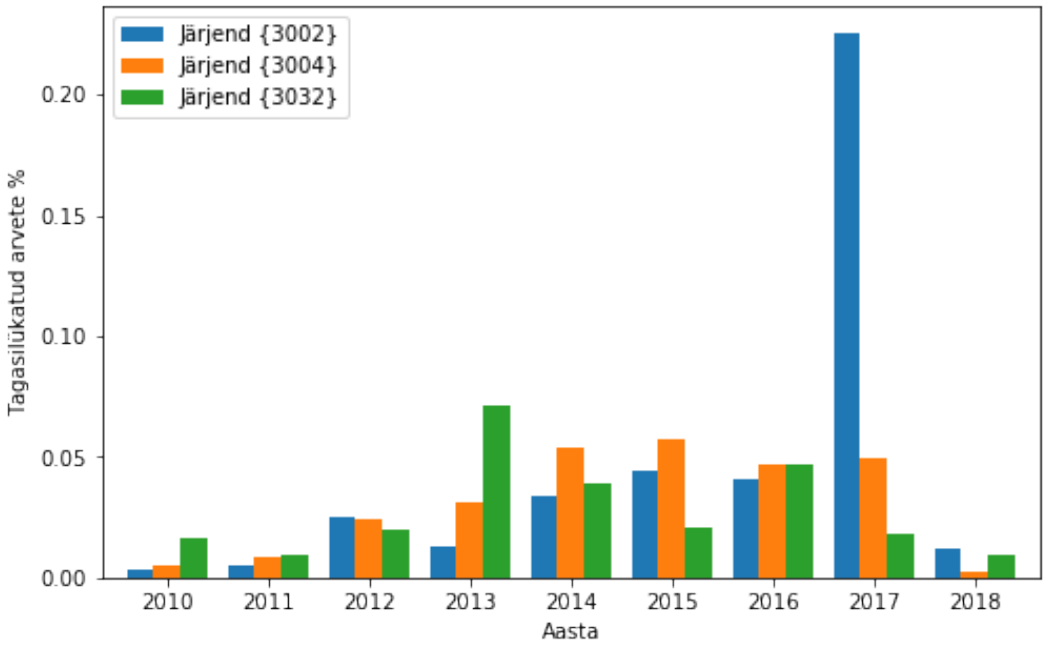
4.1 Enim esinenud järjendid

Järjendite esinemiste arvust selgus, et iga aasta oli kõige enam esinenud raviarveid, kus teenuseks oli eriarsti esmane vastuvõtt (kood 3002), eriarsti korduv vastuvõtt (kood 3004) ning psühhiaatri vastuvõtt aktiivravi perioodis (kood 3032) (joonis 5).



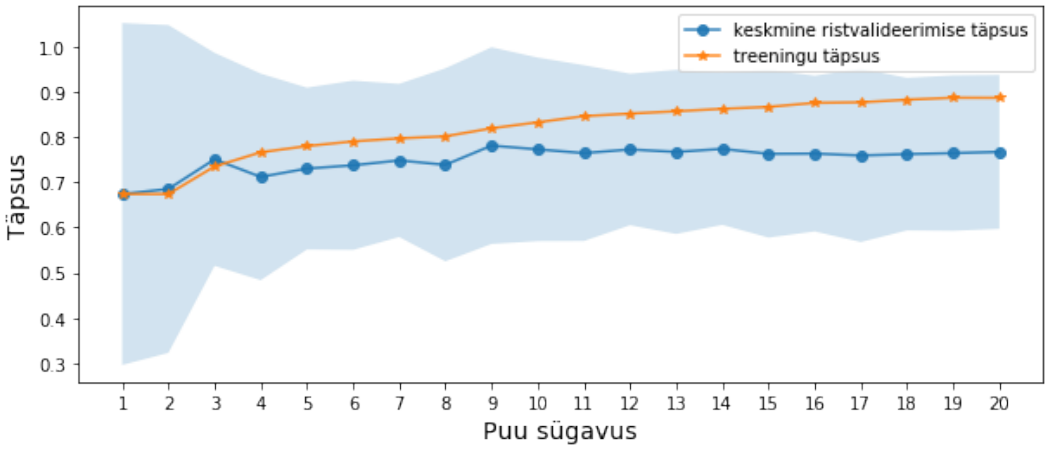
Joonis 5. Kolme kõige enam esinenud järjendi esinemite arv aastate kaupa.

Antud järjendite hulgas oli väga vähe tagasilükkamisi, 2700 tagasilükkamist 7 miljoni kohta, mistõttu saab puud rakendades hea ülevaate, mis põhjustel tagasilükkamine oli toimunud (joonis 6). Kuna nimetatud järjendid olid iga aasta kohta enim esinenud, siis ei vaadatud järjendit kui teenuse koodi ning selle kogust, vaid võeti kõikidest andmetest välja raviarved, mis koosnesid ühest reast ning teenus oli kas 3002, 3004 või 3032.



Joonis 6. Kolme kõige enam esinenud järjendi tagasilükatud arvete protsent aastate kaupa.

Enim informatsiooni andvate tunnuste leidmiseks ning optimaalseima puu sügavuse valimiseks viidi klassifitseerimispuu algoritmil läbi 10-kordne *stratified* ristvalideerimine, milleks kasutati näitekoodi. Leiti, et parim puu sügavus valitud tunnustega on 9, kuna see andis parima täpsuse ristvalideerimise korral (joonis 7).



Joonis 7. Treening andmestiku täpsus ning keskmine ristvalideerimise täpsus ja selle 95% usaldusvahemik puu sügavuse kaupa

Tabel 5 näitab tulemusi, mis on saadud tasakaalustatud treening andmete korral. Algoritmi klassifitseerimistäpsuseks tuli

$$\frac{2068 + 2388}{2068 + 361 + 681 + 2388} \cdot 100\% \approx 81,05\% \ ,$$

täpsuseks

$$\frac{2068}{2068 + 361} \approx 0,8514 \ ,$$

saagiseks

$$\frac{2068}{2068 + 681} \approx 0,7523$$

ning F_1-skooriks

$$2 \cdot \frac{0,8514 \cdot 0,7523}{0,8514 + 0,7523} = 0,7988 \ .$$

Tabel 5. Prognoosi ja tegelike tulemuste sagedustabel treeningandmetelt

Prognoos/Tegelik	Lükati tagasi	Ei lükatud tagasi
Lükati tagasi	2 068	361
Ei lükatud tagasi	681	2 388

Tulemuseks saadud puul esines mitmeid lehti, mille vaatluste lõplik protsentuaalne jagunemine oli ligikaudu $[50\%, 50\%]$ (lisa 1). See viitab, et nende vaatluste kohta on puudu vajalik meta informatsioon, et teha õige otsus, kas rida lükatakse tagasi või mitte.

4.1.1 Võrdsete väärtustega lehtede uurimine

Tasakaalustatud treeningandmete seas tuli nähtavale 6 lehte, milles jagunesid alles jäänud vaatlused võrdselt mõlemasse klassi. Iga lehe korral võeti tingimused, mille alusel antud lehte jõuti ning võeti kogu andmestikust välja kõik vaatlused, mis sinna kuulusid. Igalt saadud valimilt treeniti uus puu, kasutades kõiki olemasolevaid tunnuseid.

Kaks kirjeldatud kriteeriumitele vastavat lehte olid tekkinud tingimustest, et TTO asukoha kood ei ole '0000' (asukoht puudub) ega '0387'(Lasnamäe linnaosa), TTO ei ole tüübilt üldhaigla ega valikupartner, ravitüüp ei ole ambulatoorne ületöö, teenus ei ole eriarsti korduv vastuvõtt ning saatja eriala on anestesioloogia. Esimesel lehel oli lisaks tingimus, et elukohakood on '37' ning teisel, et elukohakood ei ole '37'. Mõlema valimi korral treeniti puu, kus lehes ei olnud täpsustatud minimaalset vaatluste arvu ning puu sügavust ei olnud piiratud.

Esimese lehe puhul, ei suutnud klassifitseerimispuu antud tunnuste korral õigesti prognoosida, millised vaatlused on tagasilükatud ning teise lehe korral suudeti õigesti klassifitseerida 3 vaatlust 8-st (tabel 6). See näitab, et tagasilükkamist on mõjutanud mõni tunnus, mida töö koostajale pole antud. Lisaks esines puudel endiselt lehti, kus vaatlused jagunesid protsentuaalselt võrdseteks osadeks (lisa 2). Seetõttu võttis töö kirjutaja seisukoha, et antud lehtede puhul ei oleks algses puus sügavuse suurendamine mõttekas, kuna olulist infot see juurde ei annaks.

Tabel 6. Esimese ja teise lehe prognoosi ja tegelike tulemuste sagedustabel

Esimene leht		
Prognoos/Tegelik	Lükati tagasi	Ei lükatud tagasi
Lükati tagasi	0	0
Ei lükatud tagasi	14	45 026

Teine leht		
Prognoos/Tegelik	Lükati tagasi	Ei lükatud tagasi
Lükati tagasi	0	3
Ei lükatud tagasi	8	35 038

Sarnased tulemused tulid ka teiste lehtede korral, kuid lisaks tuli puu lõplik sügavus mitmekordselt suurem võrreldes eelneva kahe lehega. Kuna uuritud lehti ei suudetud ka ülejäänud olemasolevate tunnustega õigesti klassifitseerida ning jätkuvalt leidus ligikaudu [50%,50%] jaotusega lehti, siis ei ole töö kirjutaja meelest ka nende lehtede korral antud tunnuste korral puu edasine loomine mõttekas. Mudeli parandamiseks oleks vaja kasutada lisatunnuseid, mida antud töös ei uuritud.

4.1.2 Mudeli kontroll testandmestikul

Sama mudelit rakendati testimiseks tervest enim esinenud järjendite andmestikust juhuslikult võetud valimil, milles oli $2,8$ miljonit vaatlust (tabel 7). Saadud mudeli klassifitseerimistäpsuseks tuli 99.97%,

$$\frac{383 + 2798846}{383 + 10 + 761 + 2798846} \cdot 100\% \approx 99,97\% \ ,$$

täpsuseks

$$\frac{383}{383 + 10} \approx 0,9746 \ ,$$

saagiseks

$$\frac{383}{383+761} \approx 0,3348$$

ning F_1-skooriks

$$2 \cdot \frac{0,9746 \cdot 0,3348}{0,9746+0,3348} = 0,4984 \ .$$

Madal saagis näitab, et leitud mudel on range ning loeb paljud arved hüvitatuteks, kuigi tegelikult on need tagasilükatud. Seetõttu tuli ka F_1-skoor madal. Kui võtta juurde tunnuseid, mida käesoleva bakalaureusetöö kirjutajal ei ole, siis oleks ilmselt tulemus parem.

Tabel 7. Prognoosi ja tegelike tulemuste sagedustabel kogu enim esinenud järjendite andmestikust võetud 2,8 miljonilise valimi korral

Prognoos/Tegelik	Lükati tagasi	Ei lükatud tagasi
Lükati tagasi	383	10
Ei lükatud tagasi	761	2 798 846

Suur klassifitseerimistäpsus antud valimi korral näitab, et andmestikus on kahetiste väärtustega puu lehtede kogu osakaal palju väiksem, kui tasakaalustatud valimil. Saadud puu osa, kus on näidatud tagasilükkamist põhjustavaid tingimusi on toodud lisas 3.

4.1.3 Treeningandmete ja testandmete tulemuste võrdlus

Tabelist 8 on näha, et kuigi testandmestikul loodud puu klassifitseerimistäpsus on suurem kui treeningandmestiku korral, siis F_1-skooride võrdlusest selgub, et treeningandmestiku korral on mudel parem. Testandmestiku korral tuli saagis madal, mistõttu on ka mudeli F_1-skoor väikene. See näitab, et testandmestiku korral hinnatakse tagasilükatud arved tihti hüvitatuteks, kuid tunnuseid, mille tõttu on arved

hinnatud tagasilükatuteks, saab pidada olulisteks mõjutajateks arve tagasilükkamise otsusel.

Tabel 8. Treening- ja testandmetel saadud mudeli näitajate võrdlus

	Treeningandmestik	Testandmestik
Klassifitseerimistäpsus	81,05%	99,97%
Täpsus	0,8514	0,9746
Saagis	0,7523	0,3348
F_1-skoor	0,7988	0,4984

Puu treenimiseks kasutati tunnuseid

- aasta,
- arve tüüp,
- diagnoosigrupp
- elukohakood,
- EMO arve,
- peaerala,
- ravitüüp,
- saatja eriala,
- sugu,
- tagasinõue,
- teenuse kood,
- TTO-tüüp,
- TTO asukoht,
- väljakirjutamise staatus,
- vältimatu abi arve.

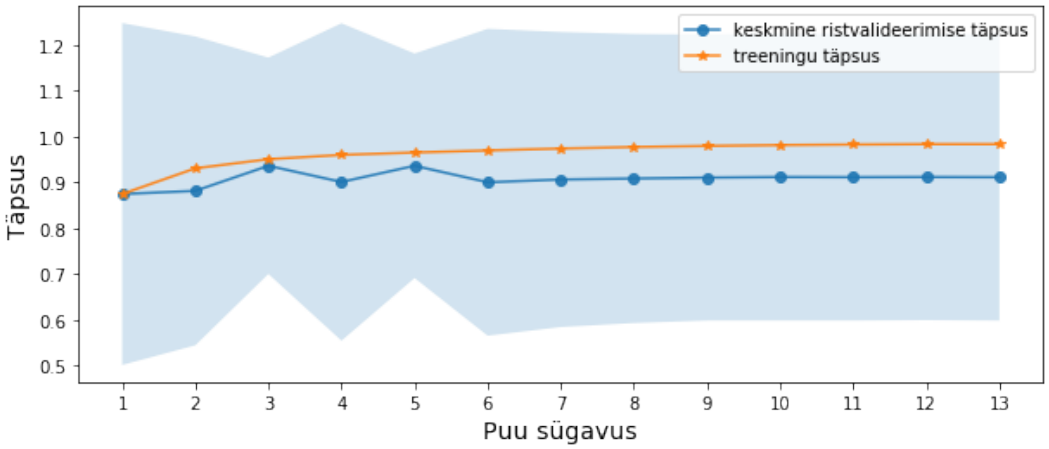
Kuna kõik kasutatud tunnused olid kategoorilised, siis enne puu treenimist loodi nende põhjal uued fiktiivsed $(0, 1)$ tunnused, kus 1 märgib tõest rida ning 0 väära. Näiteks, tunnusest aasta moodustati eraldi tunnused *aasta_2010*, *aasta_2011*, jne, ning kui arve aastaks oli 2010, siis tunnuse *aasta_2010* väärtus selle rea korral oli 1.

Kokkuvõttes ei suudetud enim esinenud teenuste järjendite korral kindlat reeglistikku välja selgitada, kuid nii treening- kui ka testandmestikul on korduvaid tunnuseid, mis tagasilükkamist põhjustavad. Treeningandmestikust saadud puu parempoolseimast harust on näha, et lükatakse tagasi arved, kus TTO asukoht on '0387' (Lasnamäe linnaosa) ning aasta on 2017 (lisa 1). Testandmestiku korral saadi sarnane reeglistik, kus on lisaks tingimus, et saatja eriala ei tohi olla E240 oftalmoloogia (lisa 3). Nagu eelnevalt mainitud, ei ole kõiki olemasolevaid tunnuseid selle bakalaureusetöö käigus kasutatud, mistõttu ei saa kindlalt öelda, et antud tingimustel lükatakse arve tagasi.

4.2 100 kuni 1000 korda esinenud järjendid

Treeniti eraldi mudel nende arvete kohta, mille järjend oli kogu andmestikul kordunud 100 kuni 1000 korda, kuna nii mitu korda esinenud järjendite seas oli veel juhtumeid, kus mõne järjendi tagasilükkamise protsent jäi 90 kuni 100% juurde (joonis 4). Vastupidiselt eelmises peatükis tehtule vaadati siin peatükis teenuste järjendit tervikuna. Iga järjendi kohta vaadatakse järjendis olevaid teenuseid ning teisi tunnuseid iga arverea kohta eraldi ehk arvestatud pole, kas arves on üks või mitu rida.

Ristvalideerimise abil treeniti tasakaalustatud andmetel mudel, mis andis parima täpsuse sügavusel 3 (joonis 8).



Joonis 8. Treening andmestiku täpsus ning keskmine ristvalideerimise täpsus ja selle 95% usaldusvahemik puu sügavuse kaupa

Tabelis 9 on näha treeningandmetelt saadud prognooside ja tegelike tulemuste erinevused. Mudeli täpsuseks tuli $95,07\%$ ning F_1-skooriks $0,9485$. Kõrge F_1-skoor näitab, et mudel on paremini kooskõlas, kui eelmises peatükis loodud mudel.

Tabel 9. Prognoosi ja tegelike tulemuste sagedustabel tasakaalustatud treeningandmetelt

Prognoos/Tegelik	Lükati tagasi	Ei lükatud tagasi
Lükati tagasi	2 719	24
Ei lükatud tagasi	271	2 966

Jooniselt 9 on näha, et saadud puus puuduvad lehed, kus vaatlused jaguneksid ligikaudu võrdseteks osadeks. Selleks, et jõuda puus vasakult esimesse lehte, kus on ainukesed mitte tagasilükatud vaatlused, tuleb läbida 3 vahetippu. Kuna me kasutame kõikide tunnuste jaoks *dummy* tunnuseid, siis antud lehte jõudmiseks peavad olema täidetud järgmised tingimused:

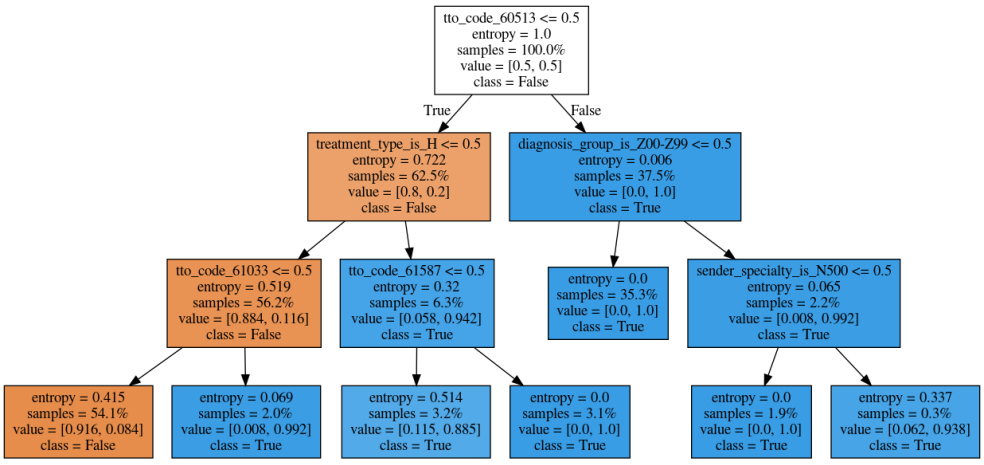
- TTO-kood ei ole ’60513’,

- ravitüüp ei ole hambaravi (kood ’H’),
- TTO-kood ei ole ’61033’.

Seega, 100 kuni 1000 korda esinenud teenuste järjendite korral saame antud puu põhjal öelda, et tagasi ei lükata arveid, kus TTO-ks ei ole XXX (kood ’60513’) ega XXX (kood ’61033’) ning ravitüüp ei ole hambaravi. Puult on näha, et 100 kuni 1000 korda esinenud teenuste järjendite korral lükatakse tagasi

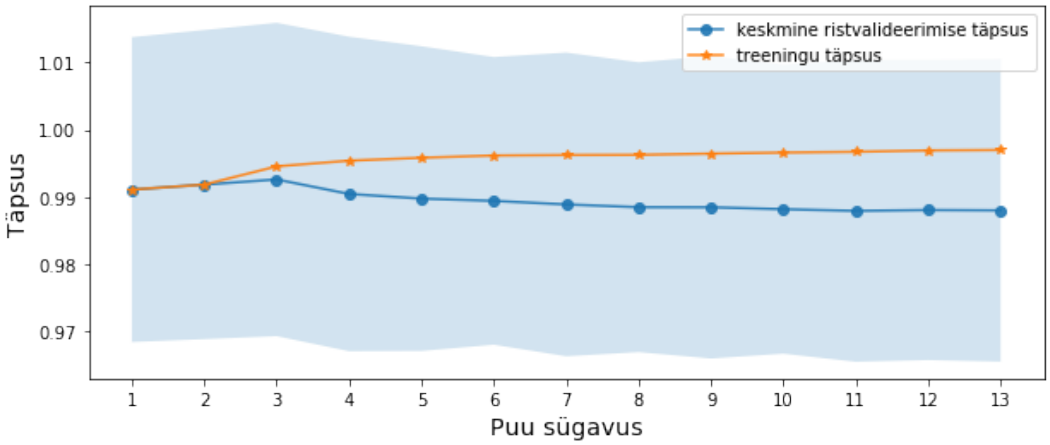
- TTO ’61033’ raviarved, kus ravitüübiks ei ole hambaravi,
- TTO ’61587’ arved, kus ravitüübiks on hambaravi,
- kõik TTO ’60513’ arved.

Lehtede tumeda värvi põhjal on näha, et saadud klassifikatsioon on üsnagi täpne. Lisaks kinnitab seda suhteliselt madal entroopia.



Joonis 9. 100 kuni 1000 korda esinenud teenuste järjendite tasakaalustatud andmestikul loodud klassifitseerimispuu

Sama mudelit rakendati kõikidele arvetele, mille teenuste järjend oli 100 kuni 1000 korda esinenud järjendite hulgas. Ristvalideerimine näitas, et optimaalseimaks puu sügavuseks oli samuti 3 (joonis 10).



Joonis 10. Terve 100 kuni 1000 korda esinenud järjendite andmestiku täpsus ning keskmine ristvalideerimise täpsus ja selle 95% usaldusvahemik puu sügavuse kaupa

Terve andmestiku korral saadud tulemused on toodud tabelis 10, kust saadi klassifitseerimistäpsuseks $99,46\%$ ning F_1-skooriks $0,9063$. Tulemused on ligikaudu samad, mis tasakaalustatud andmestiku korral.

Tabel 10. Prognoosi ja tegelike tulemuste sagedustabel terve 100 kuni 1000 korda esinenud teenuste järjendite andmestiku korral

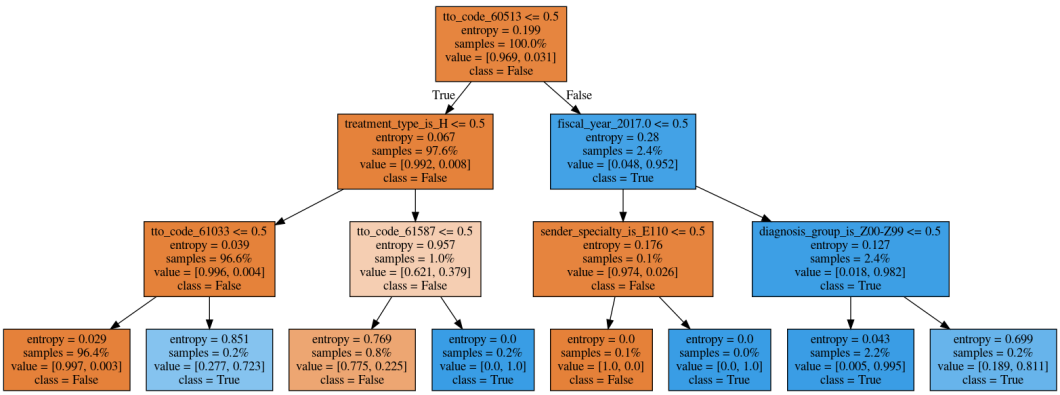
Prognoos/Tegelik	Lükati tagasi	Ei lükatud tagasi
Lükati tagasi	2 549	86
Ei lükatud tagasi	441	93 831

Saadud puu on toodud joonisel 11, kus on näha, et võrreldes tasakaalustatud andmetel saadud puuga, on lehed vähem puhtad ning klassifikatsioon ei ole niivõrd tugev. Siiski säilisid treeningandmetelt saadud reeglid, et tagasi lükatakse

- TTO '61033' arved, kus ravitüübiks ei ole hambaravi,

- TTO ’61587’ arved, kus ravitüübiks on hambaravi.

Kui tasakaalustatud andmetelt saadi tingimus, et tagasi on lükatud kõik TTO ’60513’ arved, siis jooniselt 11 selgub, et tagasi lükatakse kõik nende arved, välja arvatud juhul, kui saatja eriala ei ole dermatoveneroloogia (kood E110).



Joonis 11. 100 kuni 1000 korda esinenud teenuste järjendite kogu andmestikul loodud klassifitseerimispuu

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli tuvastada reeglid, mille alusel haigekassa otsustab arve tagasilükata või hüvitada.

Bakalaureusetöös anti ülevaade klassifitseerimispuudest ja haigekassa otsustusprotsessist. Töö praktilises pooles rakendati klassifitseerimispuu algoritmi arvetele, mille teenuste järjendid olid andmestikus enim esinenud ning arvetele, mille järjendit oli esinenud 100 kuni 1000 korda.

Töö tulemusena selgus, et enim esinenud teenuste järjendeid kasutades oli puuduvaid tunnuseid, mis mõjutavad arve tagasilükkamist. Lisaks selgusid TTO-d, kelle raviarved lükati 100 kuni 1000 korda esinenud teenuste järjendite korral tagasi.

Tööd on võimalik edasi arendada, kuna selgus, et käesoleva bakalaureusetöö jaoks ei olnud antud tunnused piisavad, sest puu ei suutnud kindlaid otsuseid paljudel juhtudel teha.

Kasutatud kirjandus

[1] Mitchell, T. (1997). *Machine learning*. The McGraw-Hill Companies, Inc, lk 52–60.

[2] James, G., Witten, D., Hastie, T. ja Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. New York: Springer, lk 85, 130, 176-186, 306-312. http://faculty.marshall.usc.edu/gareth-james/ISL/ISLR\%20Seventh\%20Printing.pdf

[3] Kelleher, J., Mac Namee, B. ja D'Arcy, A. (2015) *Fundamentals of machine learning for predictive data analytics*. The MIT Press, lk 99, 402-416.

[4] Kuhn, M., Johnson, K. (2013) *Applied Predictive Modeling*. Springer, lk 394. Kasutatud 17.05.2020. doi: 10.1007/978-1-4614-6849-3

[5] Scikit-learn. *1.10. Decision Trees*. Kasutatud 07.05.2020. https://scikit-learn.org/stable/modules/tree.html

[6] Tooding, L-M. *Regressioonimudelid*. Kasutatud 19.05.2020. http://samm.ut.ee/regressioonanalyys

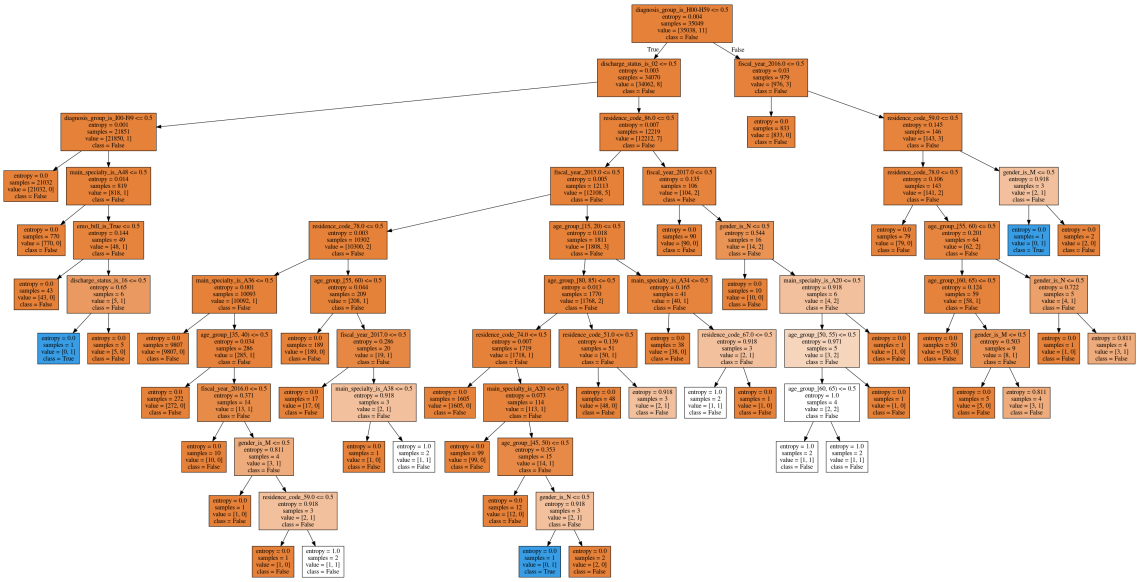
[7] Eesti Haigekassa. *Raviarvete ja lepingute andmevahetusteenused*. Kasutatud 08.05.2020. https://www.haigekassa.ee/sites/default/files/IT_juhised/EHK_RTA_teenused_v3.9vv.pdf

[8] Vabariigi Valitsus. *Haiglavõrgu arengukava*. Kasutatud 09.05.2020. https://www.riigiteataja.ee/akt/104042018005

[9] Tartu Ülikooli psühhiaatriakliinik. *Rahvusvaheline haiguste klassifikatsioon - RHK-10*. Kasutatud 18.05.2020. https://www.kliinikum.ee/psyhhiaatriakliinik/lisad/ravi/RHK/RHK10-FR17.htm

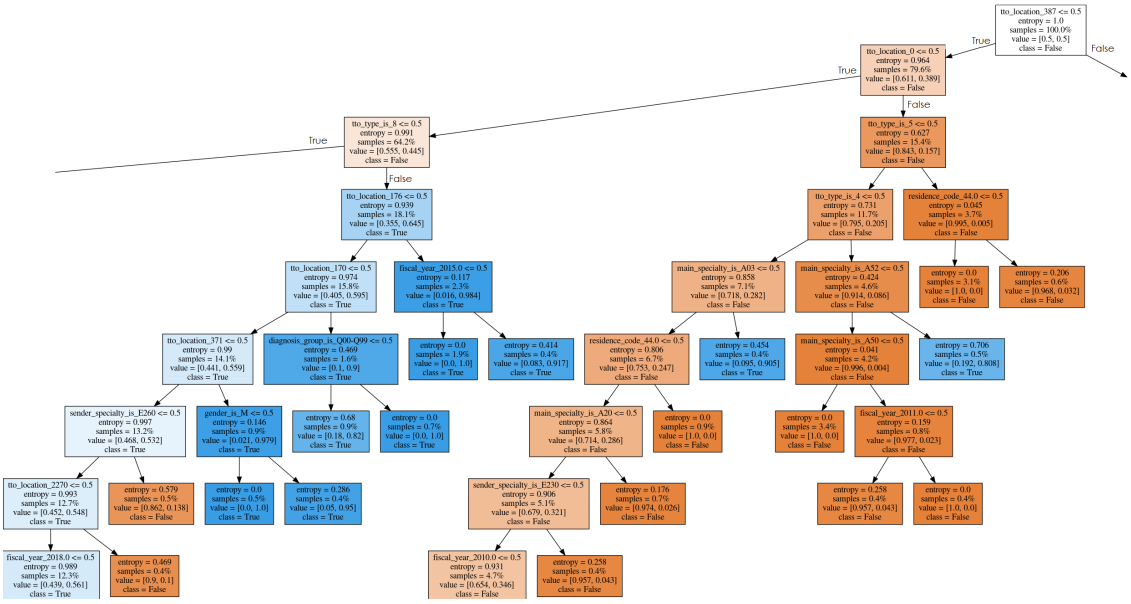
Lisad

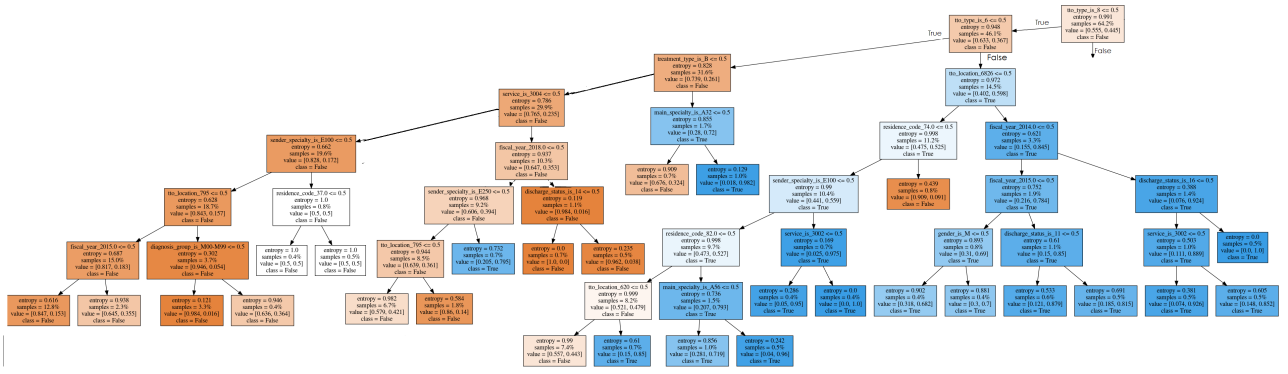
Lisa 1. Teise lehe tingimuste põhjal kõikidest andmetest ning tunnustest loodud klassifitseerimispuu.

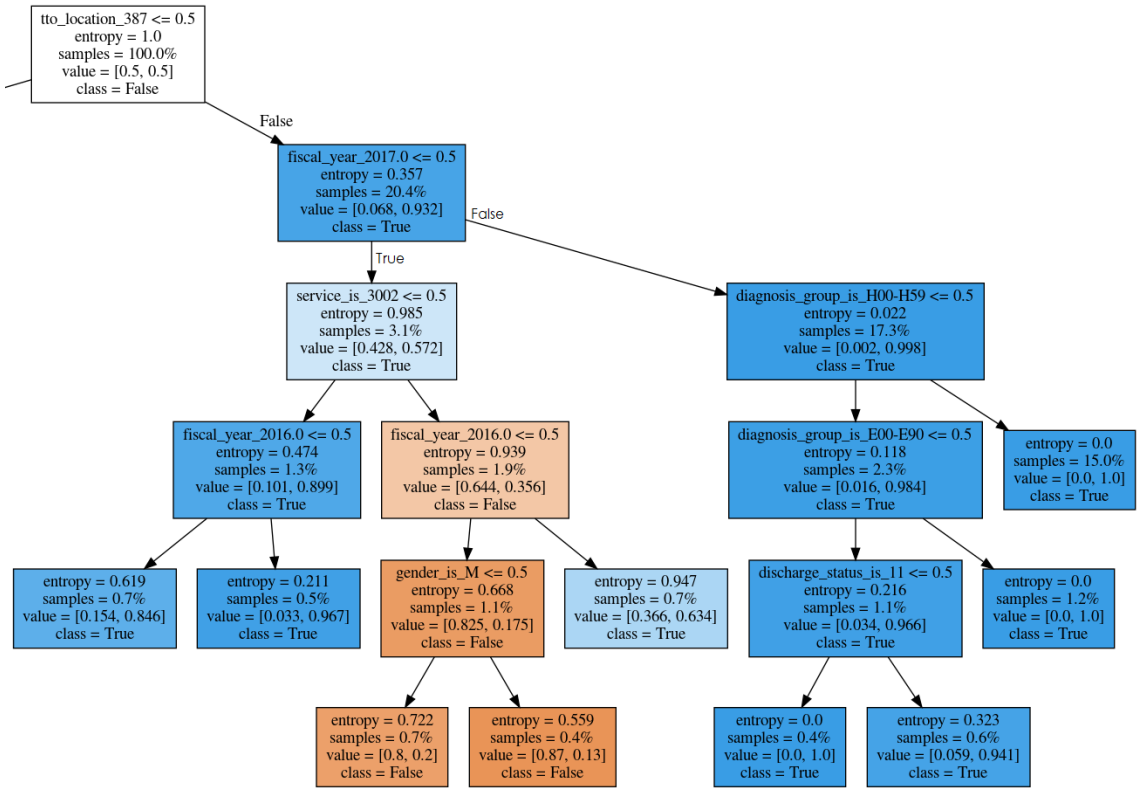


Lisa 2. Enim esinenud järjendite andmestikul treenitud klassifitseerimispuu.

Puu on lõigatud loetavuse jaoks tükkideks. Ülemine joonis näitab puu algust ning vasakpoolset (*True*) haru. Teine joonis kujutab vasaku haru jätku ning kolmas joonis juurtipust paremale hargnevat puud.






tto_location_387 <= 0.5
entropy = 1.0
samples = 100.0%
value = [0.5, 0.5]
class = False
False
fiscal_year_2017.0 <= 0.5
entropy = 0.357
samples = 20.4%
value = [0.068, 0.932]
class = True
False
True
service_is_3002 <= 0.5
entropy = 0.985
samples = 3.1%
value = [0.428, 0.572]
class = True
diagnosis_group_is_H00-H59 <= 0.5
entropy = 0.022
samples = 17.3%
value = [0.002, 0.998]
class = True
fiscal_year_2016.0 <= 0.5
entropy = 0.474
samples = 1.3%
value = [0.101, 0.899]
class = True
fiscal_year_2016.0 <= 0.5
entropy = 0.939
samples = 1.9%
value = [0.644, 0.356]
class = False
diagnosis_group_is_E00-E90 <= 0.5
entropy = 0.118
samples = 2.3%
value = [0.016, 0.984]
class = True
entropy = 0.0
samples = 15.0%
value = [0.0, 1.0]
class = True
entropy = 0.619
samples = 0.7%
value = [0.154, 0.846]
class = True
entropy = 0.211
samples = 0.5%
value = [0.033, 0.967]
class = True
gender_is_M <= 0.5
entropy = 0.668
samples = 1.1%
value = [0.825, 0.175]
class = False
entropy = 0.947
samples = 0.7%
value = [0.366, 0.634]
class = True
discharge_status_is_11 <= 0.5
entropy = 0.216
samples = 1.1%
value = [0.034, 0.966]
class = True
entropy = 0.0
samples = 1.2%
value = [0.0, 1.0]
class = True
entropy = 0.722
samples = 0.7%
value = [0.8, 0.2]
class = False
entropy = 0.559
samples = 0.4%
value = [0.87, 0.13]
class = False
entropy = 0.0
samples = 0.4%
value = [0.0, 1.0]
class = True
entropy = 0.323
samples = 0.6%
value = [0.059, 0.941]
class = True

Lisa 3. Kasutatud kood parima puu sügavuse leidmiseks ristvalideerimise meetodil.

Kohandatud *stratified* 10-kordse ristvalideerimise läbiviimiseks.

https://towardsdatascience.com/how-to-find-decision-tree-depth-vi a-cross-validation-2bf143f0f3d6

```
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import precision_recall_fscore_support
from sklearn.model_selection import StratifiedKFold, KFold
from sklearn.metrics import average_precision_score
from sklearn.metrics import recall_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
import numpy as np
import matplotlib.pyplot as plt

#funktsioon puu treenimiseks erinevatel sygavustel kasutades
#ristvalideerimist
def run_cross_validation_on_trees(X, y, tree_depths,
#cv=StratifiedKFold(10), scoring='accuracy'):
    cv_scores_list = []
    cv_scores_std = []
    cv_scores_mean = []
    accuracy_scores = []
    for depth in tree_depths:
```

```
            tree_model = DecisionTreeClassifier(criterion = "entropy",
            random_state = 100,
                 min_samples_leaf = 1, max_depth=depth)
            cv_scores = cross_val_score(tree_model, X, y, cv=cv,
            scoring=scoring)
            cv_scores_list.append(cv_scores)
            cv_scores_mean.append(cv_scores.mean())
            cv_scores_std.append(cv_scores.std())
            accuracy_scores.append(tree_model.fit(X, y).score(X, y))
        cv_scores_mean = np.array(cv_scores_mean)
        cv_scores_std = np.array(cv_scores_std)
        accuracy_scores = np.array(accuracy_scores)
        return cv_scores_mean, cv_scores_std, accuracy_scores

# joonis
def plot_cross_validation_on_trees(depths, cv_scores_mean,
cv_scores_std, accuracy_scores):
    fig, ax = plt.subplots(1,1, figsize=(10,4))
    ax.plot(depths, cv_scores_mean, '-o', label='keskmine
␣␣␣␣ristvalideerimise␣tapsus', alpha=0.9)
    ax.fill_between(depths, cv_scores_mean-1.96*cv_scores_std,
    cv_scores_mean+1.96*cv_scores_std, alpha=0.2)
    #ylim = plt.ylim()
    ax.plot(depths, accuracy_scores, '-*', label='treeningu␣tapsus',
    alpha=0.9)
    ax.set_xlabel('Puu␣sygavus', fontsize=14)
    ax.set_ylabel('Tapsus', fontsize=14)
    #ax.set_ylim(ylim)
    ax.set_xticks(depths)
```

```
    ax.legend()

X = dum_df2[features]
y = dum_df2["valid_reclaim"]
cv_scores_list = []
cv_scores_sd = []
cv_scores_mean = []
accuracy_scores = []

rskf = StratifiedKFold(n_splits=10, shuffle=True)

#Puu sygavustel 1 kuni 13
sm_tree_depths = range(1,14)
sm_cv_scores_mean, sm_cv_scores_std, sm_accuracy_scores =
run_cross_validation_on_trees(X, y, sm_tree_depths)

#Graafik
plot_cross_validation_on_trees(sm_tree_depths, sm_cv_scores_mean,
sm_cv_scores_std, sm_accuracy_scores)

idx_max = sm_cv_scores_mean.argmax()
sm_best_tree_depth = sm_tree_depths[idx_max]
sm_best_tree_cv_score = sm_cv_scores_mean[idx_max]
sm_best_tree_cv_score_std = sm_cv_scores_std[idx_max]
print('The␣depth-{}␣tree␣achieves␣the␣best␣mean␣cross-validation
accuracy␣{}␣+/-␣{}%␣on␣training␣dataset'.format(
      sm_best_tree_depth, round(sm_best_tree_cv_score*100,5),
      round(sm_best_tree_cv_score_std*100, 5)))
print(sm_cv_scores_mean)
```

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Johanna Õun,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Raviarvete tagasilükkamise põhjuste tuvastamine“, mille juhendaja on Sven Laur, 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.

Johanna Õun

19.05.2020