TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Egert Georg Teesaar

Sobiva headusmõõdu valimine binaarsete klassifitseerimisülesannete korral

Bakalaureusetöö (9 EAP)

Juhendaja: Mari-Liis Allikivi, MSc
Juhendaja: Meelis Kull, PhD

Tartu 2018

Sobiva headusmõõdu valimine binaarsete klassifitseerimisülesannete korral

Lühikokkuvõte:

Masinõpe on tehisintellekti üks suurimaid harusid, mille ideeks on imiteerida õppimisprotsessi, mida kõik elusorganismid kasutavad igapäevaste probleemidega toimetulemisel. See kasutab õppimiseks sarnaste ülesannete kohta olemasolevaid andmeid. Selle käigus üritab masinõppe algoritm tuvastada erinevaid mustreid, et treenida mudel, mis aitaks hiljem teha üldistusi samalaadsete probleemidega tegelemisel.

Üheks masinõppe suunaks on binaarklassifitseerimine. Binaarne klassifitseerimine tegeleb probleemidega, millele leidub ainult kaks võimalikku lahendit ehk klassi. Seega üritab binaarse klassifitseerimisülesande lahendamiseks treenitud mudel ennustada, kas vastav probleem kuulub klassi A või B. Seoses sellega aga tekib küsimus, kas olemasolev mudel on sobilik vastava probleemi lahendamiseks. Seda saab hinnata headusmõõtudega.

Käesolev uurimistöö tutvustab, kuidas valida sobivat headusmõõtu binaarsetele klassifitseerimisülesannetele. Töö toob välja erinevaid headusmõõte ning esitab küsimused, mis aitavad kindlaks teha klassifitseerimisprobleemi eesmärgi ning konteksti. Varasemalt on ilmunud mitmeid töid, mis annavad ülevaate erinevatest headusmõõtudest ning nende omadustest, kuid need nõuavad tööde lugejatelt juba teatud eelteadmisi ning jätavad tähelepanuta erinevate mõõtude kasutamisega kaasnevad riskid ning puudujäägid. Samuti ei too need välja kindlat juhist, kuidas mõõtu valida.

Seega on antud töö eesmärk aidata sobiva mõõduni jõuda ka inimestel, kellel puuduvad sügavamad teadmised masinõppest.

Võtmesõnad:

Binaarne klassifitseerimine, headusmõõdud, klassifitseerija hindamine

CERCS: P175 Informatics, systems theory

Choosing Appropriate Performance Measure for Binary Classification Problem

Abstract:

Machine learning is a big part of artificial intelligence which tries to imitate the learning process what every living organism uses in everyday life to deal with arisen problems. It uses available data about the problem of interest to learn and detect patterns so it can build a model which could be helpful in overcoming similar problems in the future.

One branch of machine learning is binary classification. It specializes in problems where there are only two possible outcomes also know as classes. Therefore the model which has been trained to solve such problems can only predict class A or class B. This in turn raises question how can one know if the given model is appropriate to deal with such problems. One way to evaluate this, is to use performance meaasures.

This thesis focuses on how to choose appropriate performance measure for binary classification problems. It brings out different measures and provides questions which try to discover the purpose of the model and the context in which it was trained. There have been published many works which give an insight into performance measures and their characteristics but they require the reader to already be familiar with the topic and therefore leave out risks and shortcomings of certain measures. They also don't provide a concrete manual on how to choose a performance measure.

This work tries to help people, who lack deeper knowledge about machine learning, to discover appropriate measure for the problem in hand.

Keywords:

Binary classification, performance measure, classifier evaluation

CERCS: P175 Informatics, systems theory

Sisukord

1 Sissejuhatus

Masinõpe on viimaste kümnendite jooksul laialdaselt populaarsust kogunud ning tohutuid edusaame teinud. Sellele üritatakse kasutust leida peaaegu igas võimalikus valdkonnas. Tänu masinõppeteekidele nagu näiteks Pyhtoni Scikit-Learn[6] ja Java WEKA[9] on võimalik kergesti treenida keerukaid masinõppe mudeleid isegi inimestel, kellel on väheseid kogemusi antud valdkonnas. See tähendab, et inimesed ei pea enam täpselt aru saama, kuidas masinõppe algoritmid töötavaid, oluline on vaid see, et need väljastaks soovitud tulemusi.

See aga tõstatab küsimuse, kuidas on võimalik kindlaks teha, et treenitud mudel tõesti soovitud tulemusi väljastas. Kuna iga probleem, mille lahendamisel masinõpet kasutatakse, on erinev, tuleb ka masinõppe mudeli hindamisel arvesse võtta algset eesmärki ning konteksti, millele mudel pidi vastuse leidma. Üheks võimalikuks mooduseks, kuidas olemasolevat mudelit hinnata, on kasutada spetsiaalseid mõõdikuid, mida kutsutakse "headusmõõtudeks". Nende all mõeldakse funktsioone, mis mudeli poolt ennustatud tulemusi võrdlevad soovitud tulemustega ning seejärel tagastavad skoori, mis näitab, kui hea nende arvates mudel on. Seega, kui sama mõõtu rakendada erinevate mudelite peal, saab saadud skoore omavahel võrrelda ning öelda, milline mudel parim on. Asja teeb keeruliseks see, et iga headusmõõt hindab erinevaid külgi mudelite juures. Mõned neist võtavad näiteks arvesse, kui kindel on mudel oma ennustustes, teised jällegi loevad kokku, mitu viga mudel tegi ning kus täpselt need vead esinesid.

Seda, kuidas valida kõige sobivamat headusmõõtu probleemile, püüab abistada käesolev uurimistöö. Täpsemalt keskendub antud uurimistöö binaarsetele klassifitseerimisprobleemidele, mis on üheks levinumaks masinõppe haruks. Binaarse klassifitseerimise korral tuleb masinõppe algoritmil määrata huvipakkuv objekt, kas ühte või teise klassi. Kuigi varasemalt on ilmunud mitmeid töid, mis tutvustavad erinevaid headusmõõte binaarsele klassifitseerimisele, ei ole sageli arvestatud inimestega, kellel puuduvad sügavad kogemused masinõppe valdkonnast. Selliste inimeste aitamiseks annab käesolev uurimistöö esimeses pooles ülevaate erinevatest headusmõõtudest ning nende omadustest, uurimistöö teises pooles aga esitab skeemi küsimustega, mille abil on võimalik valida sobiv headusmõõt binaarsele klassifitseerimisprobleemile.

Uurimistöö struktuur on jaotatud järgnevalt. Peatükk number kaks annab

ülevaate masinõppest taustast ning klassifitseerimisest. Kolmas peatükk räägib pikemalt klassifiteerija hindamisest ning aspektidest, mida tuleks hindamise juures arvesse võtta. Uurimistöö neljandas peatükis esitatakse headusmõõtude definitsioonid ning omadused koos näidetega, millal oleks mõistlik teatud headusmõõtu kasutada. Viimases viiendas peatükis esitatakse graaf koos küsimustega, mis jagavad peatükis 4 esitatud headusmõõdud konkreetsetesse harudesse.

1.1 Seotud tööd

Varasemalt on ilmunud mitmeid töid, mis annavad ülevaate erinevatest headusmõõtudest ning nende omadustest.

Näiteks on ilmunud raamat „Evaluating Learning Algorithms: A Classification Perspective“[3], mille eesmärk on aidata lugejal aru saada algoritmide hindamise vajalikkusest, kuid ei anna selgeid juhtnööre, millal teatuid mõõte tuleks rakendada, pigem keskendutakse mõõtude omadustele.

Ühtlasi on ilmunud artikkel „A systematic analysis of performance measures for classification tasks“[8], mis annab ülevaate 24-st headusmõõdust ning keskendub mõõtude vahelise erinevuse analüüsimisele. Täpsemalt analüüsib artikkel, kuidas reageerivad mõõdud segadusmaatriksis erinevate muudatuste tegemisele, mille visualiseerimiseks loob artikkel kokkuvõtlikku tabeli, kus on loetletud kaheksa liiki muudatust, mis segadusmaatriksiga tehti, ning tulemus, kas mõõt reageeris vastavale muutusele või püsis konstantsena. Segadusmaatriks on maatriks, kus saab kujutada klassifitseerija poolt ennustatud tulemusi testimisel kasutatavatele näidetele.

Lisaks on ilmunud artikkel „An experimental comparison of performance measures for classification"[2]. Artikkel tutvustab klassifitseerimisprobleemidele mõeldud 18 headusmõõtu, mida proovib kokku grupeerida, vastavalt sellele, kui sarnased mõõdud üksteisele on. Artikkel jõuab järeldusele, et paljud mõõdud ei ole üksteisest väga erinevad ning sageli pole vahet, millist neist klassifitseerija töö hindamisel kasutada, kuid samas toob välja ka mõõte, mille käitumine on teistest teatud tingimustel väga erinev.

Samuti on ilmunud artikkel „A Study on the Relationships of Classifier Performance Metrics"[7]. Artikkel on sarnane eelmisele selle poolest, et loetleb üles 22 headusmõõtu ning üritab kindlaks teha, millised neist on omavahel korreleeritud

ning millised mitte. Artikkel võimaldab kindlaks teha, milliseid headusmõõte oleks mõttekas koos kasutada ning milliseid mitte. Selleks toob põhjuse, et headusmõõdud, mis on üksteisega väga sarnaseid, ei anna piisavalt unikaalset informatsiooni klassifitseerija töö kohta, kui neid koos hindamisel rakendada.

Loetletud artiklid toovad välja küll mitmeid erinevaid headusmõõte ning nende omadusi, kuid on rohkem mõeldud inimestele, kes on masinõppes juba üpris kogenud ning on vastavate mõõtudega juba tuttavad. Seega nõuavad artiklid lugejatelt juba teatud eelteadmisi ning jätavad tähelepanuta erinevate mõõtude kasutamisega kaasnevad riskid ning puudujäägid. Erinevalt artiklitest ei tee raamat „Evaluating Learning Algorithms A Classification Perspective“ selliseid eeldusi lugejate osas ning vajab vaid mõningaid eelteadmisi masinõppest, kuid sarnaselt artiklitele puuduvad selged juhised, kuidas valida sobivat headusmõõtu. Raamatu eesmärk on viia lugejad kurssi sellega, mida tuleks mudelite hindamisel arvesse võtta ning seletab lahti erinevate headusmõõtude head küljed ning puudused, kuid jätab lugeja ülesandeks ära otsustada, millist headusmõõtu oleks mõistlikum kasutada erinevates kontekstides.[3, p7]

2 Masinõpe

Masinõpe on tehisintellekti üks suurimaid harusid. Selle ideeks on imiteerida õppimisprotsessi, mida kõik elusorganismid kasutavad igapäevaste probleemidega toimetulemisel. Sellist õppimisviisi iseloomustavad võime kohaneda ning teha kogetu põhjal üldistusi, samuti tuleb kasuks ka informatsiooni talletamine, sest antud juhtum võib abiks osutuda sarnaste probleemide lahendamisel tulevikus. Võtmesõnaks eelmises lauses on üldistamine. Oluline on aru saada, et vastused küsimustele, kuidas tekkivatest probleemidest üle saada, ei ole unikaalsed, vaid samasid lahenduskäike saab ka rakendada juhtumitel, mis erinevad algsest teatud määral. Seega ei ole tarvilik teada igat võimalikku versiooni olukorrast, et viimasega edukalt toime tulla. Peamine erinevus masinõppe ning õppimisviisi vahel, mida näiteks inimesed kasutavad, on loogika ning arutlusvõime, miks mõni nähtus teatud viisil käitus. Kuigi need on elutähtsad omadused elusmõistuse juures, masinõpe nendele ei toetu[3, p4].

Masinõppe implementeerimine koosneb S. Marslandi [4, p10-11] sõnul järgnevatest etappidest. Esimeseks sammuks on andmete kogumine ja ettevalmistamine. Eesmärgiks on hankida võimalikult palju andmeid huvipakkuva probleemi kohta ning viia need sobivale kujule. Tihtipeale tähendab viimane üksikute puuduvate väärtuste asendamist andmehulgas või väärtuste viimist numbrilisele kujule, millega masinad opereerivad. Peale töötlemist saab andmeid kutsuda juba treeningandmeteks.

Järgneval sammul tuleb valida nendest teatud tunnused, mida masinad kasutavad hiljem mudeli treenimiseks. Oluline on tuvastada omadused, mis kirjeldavad kõige paremini käesolevat juhtumit ning mille põhjal oleks hiljem võimalik teha järeldusi. See on ka masinõppe üks võtmekohtadest, sest sellest sõltub, kui hästi töötab treenitav mudel tulevikus sarnaste probleemide käsitlemisel.

Peale omaduste tuvastamist jõutakse järgmise olulise sammuni, milleks on mudeli treenimiseks sobiva algoritmi välja valimine. Algoritmi valimisel tuleb mõelda, millisel kujul soovitakse ennustusi saada. Samuti valmis mõelda, kuidas saaks hiljem testida, kas treenitav mudel on piisavalt hea.

Peale seda etappi on kõik ettevalmistused tehtud ning saab asuda mudel treenimise kallale. Mudelina võib ette kujutada funktsiooni, mis teatud sisendile väljastab sobiva väljundi. Treenimise peale kuluv aeg sõltub kõigest eelnevast: treeningand-

mete suurusest, omaduste arvust ning treenitavast algoritmist.

Viimaseks sammuks masinõppe implementeerimise juures on tulemuste hindamine. Kuidas olla kindel, kas eelnevate etappide juures tehti ikka kõik õiged otsused? Sellel peatub antud uurimistöö hiljem.

2.1 Masinõppe tüübid

Masinõpe õppeviisid jagunevad peamiselt kaheks: juhendatud ning juhendamata õpe.

Esimese viisi korral valitakse välja omadus või omadused, mida tahetakse ennustada, ning antakse treeningandmed mudeli treenimiseks ette koos vastavate omaduste väärtustega teatud kindlatel juhtudel. Nende põhjal saab mudel järeldada, et kui sisendi fikseeritud väärtustel on vastuseks taoline väärtus, siis järelikult on ka tulevikus sarnaste sisendite puhkudel vastus kas sama või ligilähedane. Antud õppimisviisi kaks kõige populaarsemat suunda on: klassifitseerimine ning regressioon. Esimese puhul on ennustatavateks väärtusteks lõplik arv klasse, kuid teise puhul kuuluvad vastused reaalarvude hulka ehk võimalikke väärtusi ennustatavale omadusele on lõpmatu arv[4, p6]. Klassifitseerimisprobleemiks on näiteks inimese veregrupi ennustamine, sest olemasolevaid veregruppe on kindel arv. Regressiooniprobleemiks on aga näiteks inimese pikkuse ennustamine. Olgugi, et inimesed ei saa kasvada lõpmatult pikaks, asub ennustatav suurus reaalarvuskaalal.

Juhendamata õpe on õppimisviis, kus sarnaselt eelmiselegi valitakse välja ennustatavad omadused, kuid sel puhul ei anta koos treeningandmetega kaasa õigeid vastuseid. See tähendab, et mudel peab ise pakutud juhtumite puhul kindlaks tegema, mis vastavate väljundite väärtuseks võiks olla. Selle jaoks tuleb mudelil juhtumeid sarnaste omaduste järgi gruppideks liigitada[4, p6].

Juhendamata õppega oleks tegu siis, kui käskida mudelil jälgida masinatel leiduvate rataste arvu, ning see seejärel paigutada tänavale, kus see erinevate masinatega kokku puutub. Tulemuseks on see, et mudel moodustab grupid autodest, ratastest, bussidest ning veokitest. Oluline punkt ülesande juures on see, et mudelile ei öeldud, et kaherattalised sõidukid on rattad, neljarattalised on autod, jne. Mudel moodustas ise grupid vastavalt sõidukite juures leitud tunnustele.

Antud õppimisliiki saab rakendada juhtudel, kui õigete vastuste kogumine on kas liiga ressursirohke või isegi võimatu. Seda tuleb ette tekstigrupeeringul, kus

tuleb tihtipeale jaotada suur hulk tekste erinevatesse liikidesse, nt teadusartikliteks, ilukirjanduseks, uudisteks, jne. Eelnevatel ei ole sageli selgelt märgitud, kuhu gruppi nad täpselt kuuluvad, seega peab mudel seda ise otsustama[4, p281].

2.2 Klassifitseerimine

Nagu juba kord öeldud kuulub klassifitseerimine juhendatud õppe alla, kus ennustatav väärtus kuulub lõpliku arvu lahendite hulka. See tähendab, et tahtmise korral saab kõik lahendid üles loetleda. Klassifitseerimine jaguneb neljaks alamliigiks: binaar- (*binary*), mitmeklass- (*multi-class*) , mitmemärgend- (*multi-label*) ning hierarhiliseks klassifitseerimisene[8].

Binaarklassifitseerimine on neist levinuim tänu selle lihtsa kontseptsiooni pärast: antud meetodiga üritatakse ennustada lahendit juhtumi ühele omadusele ehk aspektile ning see väärtus saab kuuluda ainult ühte kahest võimalikust klassist. Mitmeklassi puhul on lahendite hulgaks lõplik arv klasse, kuid üldjoontes sarnane binaarse meetodiga. Mitmemärgendklassifitseerimine erineb kahest eelnevast sellepoolest, et seal proovitakse leida lahendusi mitmele omadusele korraga. Ennustatavate lahendite hulk on lõplik. Eelnimetatud kolme ühiseks jooneks on tõsiasi, et meetodite puhul ei kattu omavahel ennustatavad klassid. Hierarhilise klassifitseerimise puhul antud tingimus ei kehti, kuna seal on iga eelnev järgneva klassi alamklassiks. Seega on klassid omavahel hierarhilises suhtes ning klassifitseerija eesmärgiks on määrata olemasolev näide võimalikult kõrgele hierarhilises puus[8].

Skoorivad/tõenäosuslikud klassifitseerijad

Lisaks eelnevale võivad klassifitseerijad erineda üksteisest sellepoolest, mis tüüpi vastuseid need väljastavad. Enamik klassifitseerijaid väljastavad iga näite jaoks fikseeritud klassi ning seega saab klassifitseerija konkreetse näite juures pakkuda, kas 100% mööda või on ennustus täiesti ideaalne. Sõltub sellest, kas klassifitseerija ennustas vastava juhu õigesti, või mitte[3, p75].

Peale selle on olemas ka skoorivad ning tõenäosuslikud klassifitseerijad. Need väljastavad pakutava skoori, kui kindlad nad on näite kuulumises mingisse kindlasse klassi[3, p75]. Näiteks, kui tõenäosuslik klassifitseerija peaks tuvastama, kas teatud objekti värvus on valge või must, siis võiks klassifitseerija kergelt hallika tooni juures pakkuda, et tema arvates on 90% tõenäosus, et objekt on valget värvi.

3 Klassifitseerija töö hindamine

Peale mudeli treenimist tekib küsimus, kui hästi toimib saadud klassifitseerija uute näidete peal. Tavaliselt eraldatakse treeningandmetest enne mudeli treenimist hulk andmeid. Eraldatud näiteid kutsutakse testandmeteks ning neid kasutatakse hiljem klassifitseerija poolt tehtud töö hindamisel. Testandmed peaksid võimalikult hästi jäljendama juhtusid, millel plaanitakse tulevikus klassifitseerijat kasutama hakata.

Klassifitseerimisalgoritmi headuse hindamiseks on mitu moodust. See uurimistöö peatub pikemalt klassifitseerija headusmõõtudel. Kuna igat klassifitseerijat treenitakse erinevatel eesmärkidel, on mõistetav, et neid ei saa ühe mõõdikuga hinnata. Seega tuleb tuvastada mõõt, mis arvestab probleemi ning konteksti, kus klassifitseerijat rakendadakse.

3.1 Segadusmaatriks

Nagu eelnevalt juba mainitud kasutatakse klassifitseerija headuse hindamiseks testandmeid. Selleks rakendatakse klassifitseerijat testhulga iga näite peal ning võrreldakse, kas ennustatav väärtus vastab tegelikkusele. Saadud tulemusi saab kujutada segadusmaatriksiga, kus ülemisel horisontaalsel real on toodud kõik tegelikud klassid ning vasakul vertikaalsel veerul kõik klassifitseerija poolt pakutud klassid ennustatavale omadusele. Seega on tegemist $l \times l$ ruutmaatriksiga C, kus l tähistab ennustatava omaduse kõigi võimalike lahendite arvu. Iga maatriksi element C_{ij} näitab, kui mitu klass i elementi määras klassifitseerija klassi j.[3, p77-78]

Kõige levinumaks formaadiks segadusmaatriksile on binaarses klassifitseerimises kasutatav 2×2 segadusmaatriks, kus tänu kahele võimalikule klassile, saab ühe võtta positiivseks ning teise negatiivseks klassiks. Vastava segadusmaatriksi väljade nimetused on seega järgnevad: element c_{11} tähistab lahtrit 'õige-positiivne' (ingl *True Positive (TP)*), c_{12} lahtrit "vale-negatiivne"(ingl *False Negative (FN)*), c_{21} lahtrit "vale-postiivne"(ingl *False Positive (FP)*) ning c_{22} lahtrit "õige-negatiivne"(ingl *True Negative (TN)*). Nagu võib märgata, on klassifitseerija ennustanud õigesti kõik väärtused, mis jäävad segadusmaatriksi peadiagonaalile, kuid on mööda pannud väärtustega, mis peadiagonaalist välja jäävad.

Binaarse segadusmaatriksi kujutus:

	Ennustatud: JAH	Ennustatud: EI
Tegelik: JAH	TP	FN
Tegelik: EI	FP	TN

Segadusmaatriksi saab koostada ka ülesannetele, kus võimalikke klasse on rohkem kui kaks. Näiteks probleemi korral, kus tuleb ära arvata, mis aastaaega langeb inimese sünnipäev, on võimalikeks klassideks: kevad, suvi, sügis, talv. Antud probleemile vastav ruutmaatriks on mõõtmetega 4×4, kus ülemine horisontaalne rida ning vasak vertikaalne veerg omavad väärtuseid: kevad, suvi, sügis, talv. Seega element c_{11} näitaks mitmele kevadel sündinud inimesele ennustas klassifitseerija sünniajaks "kevad", element c_{31} näitaks mitmele kevadel sündinud inimesele pakkus klassifitseerija aastaajaks "sügis"ning element c_{13} näitaks omakorda mitmele sügisese sünniajaga inimesele omistas klassifitseerija aastaajaks "kevad".

Tänu sellele, et segadusmaatriksiga saab kujutada peaaegu kõigi klassifitseerijate tulemusi, kasutavad enamus headusmõõte, kas kõiki või mingit osa segadusmaatriksi väljadest oma valemites.

3.2 Vigade hinnad klassifitseerimisel

Enne klassifitseerija hindamise juurde asumist tuleks välja selgitada, kas ennustatava omaduse võimalikud klassid on omavahel võrdse tähtsusega või leidub mõni klass või klassid, mille valesti klassifitseerimisel on kahju oluliselt suurem kui teiste klasside valesti klassifitseerimisel.

Näide: Vähihaigete riskigrupi tuvastamine Näiteks meditsiinis kasutatakse masinõpet vähihaigete riskigrupi tuvastamiseks. See loob olukorra, kus haige patsiendi klassifitseerimisel mitte haigeks võib omada surmavaid tagajärgi, samas kui mitte haige klassifitseerimisel haigeks nii tõsiseid tagajärgi ei omaks. [3, p88-89] Antud näite puhul omasid erinevaid hindu segadusmaatriksi vale-positiivsete ning

vale-negatiivsete väljad. Samas tuleb märkida, et eelneva näite puhul on väga raske kindlaks teha konkreetset vea hinda, sest kuigi inimese elu loetakse n-ö hindamatuks, omab vähiravi see-eest päris kopsakat hinda ning kõigi patsientide igaks juhuks haigeks liigitamine viiks ravipakkuja kiiresti pankroti. Järelikult tuleb leida tasakaal positiivse klassi elemendi valesti klassifitseerimisel negatiivsesse klassi ning negatiivse klassi elemendi klassifitseerimisel postiivsesse klassi kaasnevate kulude vahel.

Olukorrad, kus vigade tähtsus on klasside lõikes sama, iseloomustavad probleeme, kus pole oluline, kumb klass võtta negatiivseks ning kumb positiivseks.

Näide: Inimese soo ennustamine Näiteks tuleb meditsiinis ennustada haigeprofiili täitmisel vastava inimese sugu. Seega disainitakse mudel, mis ennustab inimese kaalu ja pikkuse järgi, kas tegu on mehe või naisega. Raviasutus ootab mudelilt, et see klassifitseeriks õigesti võimalikult palju inimesi. See tähendab, et tehtud viga on sama kulukas ning pole vahet, kas mees määratakse profiilis naiseks või hoopis naine meheks. .

3.3 Klassijaotus

Reaalse elu probleemidel tuleb tihti ette, et ühe klassi elemendid on tunduvalt rohkem esindatud, kui mõne teise klassi isendid, mis tähendab, et andmed ei ole tasakaalus. See tähendab, et enne headusmõõdu valimist tuleb mõelda, kuidas andmed on jaotunud mudeli treenimishetkel ning kas hiljem püsib klassijaotus sama, kui mudel päriselt käiku lastakse. Ilma klasside omavahelist suhet arvesse võtmata, võib klassifitseerija testandmetel näidata küll suurepärast efektiivsust, kuid pärast ühe klassi isendite drastilist juurde tulekut, totaalselt mööda hakata panema.

Näide: Banaani värvuse ennustamine Näiteks, kui võtta treenimiseks andmehulk, kus 99% on toored banaanid, mille värvus on roheline, ning järelejäänud 1% on küpsenud banaanid, mille värvus on kollane, ning seejärel disainida klassifitseerija, mis ennustab puuvilja värvi, siis võib tunduda hea mõte ennustada koguaeg värvi "roheline", sest nii liigitakse õigesti 99% juhtudest 100st. Viga tuleb nähtavale alles mõne aja möödudes, kui kõik toored banaanid on saanud küpsenuteks. Seega on andmehulgas 100% kollase värvusega puuvilju, kuid varasemalt disainitud mudel ennustab ikka kõigile värvuseks roheline, mille tulemusel eksib klassifitseerija kõigil

juhtudel. Sellise olukorra vältimiseks tuleks sarnastel juhtudel kasutada algoritmi hindamiseks headusmõõte, mis ei sõltu klassijaotusest ning mis ei muutu ühe klassi isendite juurde lisamisel.

Samas ülesannetel, kus klasside vaheline suhe jääb konstantseks, sõltumata ajast, ei tohiks kasutada eelmises lauses kirjeldatud headusmõõte.

Näide: Ilma ennustamine kõrbes Näiteks kui peaks prognoosima järgneva päeva ilma kõrbes, kus aastaringselt 99 % päevadest on päikseline ning 1% päevadest sajune ilm ning selleks disainida klassifitseerija, mis ennustaks 70% päikselised ilmad ning 70% sajused ilmad õigesti, siis jääks see soorituselt alla algoritmile, mis pakuks igapäev ilmaks päikeselise ilma. Olgugi, et esimene mudel paistab targem kui teine, teeb see kokkuvõttes rohkem klassifitseerimisvigu ainuüksi sellepärast, et "päikselise ilma"klass on palju suurem.

Järelikult tuleks klassijaotusele immuunseid headusmõõte kasutada, kui klasside vaheline suhe ei püsi sama, ning ülejäänud ajal pöörduda mõõtude poole, mis kasutavad ära ka klassijaotust.

4 Headusmõõtud

Järgnevas peatükis tuuakse välja klassifitseerija hindamisel kasutatavaid headusmõõte. Mõõdud jagunevad tüübi poolest kaheks: segadusmaatriksipõhised mõõdud ning skoorivad/tõenäosuslikud mõõdud. Segadusmaatriksipõhised mõõdud, mis vaatluse alla võetakse, on täiskulu (ingl *Total Cost*), täpsus (ingl *Accuracy*), veamäär (ingl *Error Rate*), geomeetriline keskmine (ingl *Geometric Mean*), tasakaalustatud täpsus (ingl *Balanced Accuracy*), F-mõõt (ingl *F measure*), täpsus saagisel (ingl *Prec @ Rec*), täpsus top K'l *Prec @ K* ning skoorivatest ja tõenäosuslikest mõõtudest on vaatluse all logaritmiline kadu (ingl *LogLoss*), keskmine ruutviga (ingl *Mean Squared Error*) ning ROCi alune pindala (ingl *Area Under the ROC Curve (AUC)*). Loetletud mõõdud on populaarsed binaarsete klassifitseerimisprobleemide hindamisel ning iga mõõt on teistest suuremal või väiksemal määral erinev. Põhjus, miks just need mõõdud käesolevasse uurimistöösse sisse on toodud, on see, et nendega annab hinnata mudeleid võimalikult paljude erinevate nurkade alt. Iga headusmõõdu juures on välja toodud vastava mõõdu definitsioon, omadused ning lisatud ka näide, kus seda mõõtu oleks sobilik kasutada.

4.1 Segadusmaatriksipõhised mõõdud

Segadusmaatriksipõhisesed mõõdud on antud alampeatükis omakorda jaotatud kaheks: "Mõõdud, mis kasutavad pooli segadusmaatriksi lahtreid"ning "Mõõdud, mis kasutavad rohkem kui kahte segadusmaatriksi välja". Esimesi mõõte üksikuna võtta pole enamikes olukordades mõtet. Näiteks on esitatud mõõt saagis (ingl *Recall*), mis näitab, kui suure osa positiivse klassi isenditest suutis klassifitseerija õigesti klassifitseerida. Seda on mõttetu üksikuna võtta, sest alati saab disainida klassifitseerija, mis ennustab isenditele alati positiivset klassi. Seeläbi püütakse kinni küll kõik positiivse klassiga isendid ehk maksimeeritakse õige-positiivsete suhe, kuid koos kõigi positiivsete näidetega loetakse positiivseks ka kõik negatiivse klassiga juhud.

Samalaadselt saab teha ka mõõduga täpsus (ingl *Precision*), mis näitab, kui suur osa klassifitseerija poolt positiivseks loetud näidetest ka päriselt positiivsesse klassi kuulusid. Selle maksimeerimiseks peaks õpetama klassifitseerijale väga täpselt, milline peab positiivne väärtus välja nägema. Sel viisil saab kannatada mudeli üldistusvõime, sest mudel hakkaks ennustama positiivseks ainult näiteid, mida ta

treenimisel korra juba näinud on, seega on pealtnäha küll mudeli töö väga hea, sest kõik väärtused, mis positiivseks loeti, olid ka päriselt positiivsed, kuid seda tehes on märkamata jäänud palju teisi päriselt positiivseid näiteid, mis nii täpselt positiivse klassi kirjeldusele ei vastanud.

Samas, kui samalaadseid mõõte omavahel kombineerida ning seega võtta vaatluse alla rohkem kui kaks segadusmaatriksi lahtrit, on saadud mõõdud juba päris informatiivsed.

Mõõdud, mis kasutavad pooli segadusmaatriksi lahtreid:

Õige-Positiivsete suhe (ingl *True-Positive Rate* (TPR)) ehk saagis (ingl ***Recall***): õigesti klassifitseeritud positiivsed väärtused jagatud kõik päriselt positiivsed väärtused ehk protsent, kui palju positiivseid väärtuseid klassifitseerija kinni püüda suutis. Jätab tähelepanuta TN ning FP ehk kõik teise(negatiivse) klassi isendid. Seega mõõt ei sõltu sellest, kas klassid on tasakaalus või mitte.[3, p94-95]

$$TPR = \frac{TP}{TP + FN}$$

Vale-Positiivsete suhe (ingl *False-Positive Rate* (FPR)): valesti klassifitseeritud negatiivsed väärtused jagatud kõik päriselt negatiivsed väärtused ehk protsent, kui palju klassifitseerija negatiivse klassi isenditest valesti klassifitseeris[3, p94-95]. Jätab tähelepanuta TP ja FN ehk kõik positiivse klassi isendid. Seega mõõt ei sõltu sellest, kas klassid on tasakaalus või mitte.

$$FPR = \frac{FP}{FP + TN}$$

Õige-Negatiivsete suhe (ingl *True-Negative Rate* (TNR)) : õigesti klassifitseeritud negatiivsed väärtused jagatud kõik päriselt negatiivsed väärtused ehk protsent, kui palju negatiivseid väärtuseid klassifitseerija poolt kinni püüti[3, p95-96]. Jätab tähelepanuta TP ning FN ehk kõik teise positiivse klassi isendid. Seega mõõt ei sõltu sellest, kas klassid on tasakaalus või mitte.

$$TNR = \frac{TN}{TN + FP}$$

Vale-Negatiivsete suhe (ingl *False-Negative Rate* (FNR)): valesti klassifitseeritud positiivsed väärtused jagatud kõik päriselt positiivsed väärtused ehk

protsent, kui palju klassifitseerija positiivse klassi isenditest valesti klassifitseeris[3, p95-96]. Jätab tähelepanuta TN ja FP ehk kõik negatiivse klassi isendid. Seega mõõt ei sõltu sellest, kas klassid on tasakaalus või mitte.

$$FNR = \frac{FN}{FN + TP}$$

Positiivse klassi ennustusvõime (ingl *Positive-Predictive Value* (PPV)) ehk täpsus (ingl ***Precision***): õigesti klassifitseeritud positiivsed väärtused jagatud kõik väärtused, mis klassifitseerija positiivsesse klassi liigitas ehk protsent, kui suur osa klassifitseerija poolt positiivsesse klassi liigitatud isenditest päriselt sinna klassi kuulus[3, p99-100]. Jätab tähelepanuta TN ja FN.

$$PPV = \frac{TP}{TP + FP}$$

Negatiivse klassi ennustusvõime (ingl *Negative-Predictive Value* (NPV)): õigesti klassifitseeritud negatiivsed väärtused jagatud kõik väärtused, mis klassifitseerija negatiivsesse klassi liigitas ehk protsent, kui suur osa klassifitseerija poolt negatiivsesse klassi liigitatud isenditest päriselt sinna klassi kuulus[3, p99-100]. Jätab tähelepanuta TP ja FP.

$$NPV = \frac{TN}{TN + FN}$$

Mõõdud, mis kasutavad rohkem kui kahte segadusmaatriksi välja

Täiskulu (ingl *Total Cost*) on mõõt, mis liidab kokku kõik klassifitseerimisvigade hinnad.
Valemis esinevate tähtede tähendused: N on näidete arv, k_i on näitega i kaasnev tulu (kulu).
Ideaalne kasutusjuhtum: Täiskulu on sobivaks headusmõõduks juhtudel, kui klassifitseerimisvigade hinnad on täpselt teada ning on vaja leida minimaalne kulu. Näiteks saab võtta poeketi, mis peab koostama poodi müügile mineva inventuuri ning selleks disainib klassifikaatori, mis prognoosiks igale tootele, kas sellele on turgu või ei ole.

Seega, kui võrduks FN vastava näite puhul olukorraga, kui klassifitseerija teeb ennustuse, et vastava toote vastu huvi ei ole, kuid tegelikult oleks see maha müüdud.

See tähendab, et klassifitseerimisviga oleks FN korral võrdne toote müügist saamatu jäänud tuluga. Kui klassifitseerija aga arvab, et toode läheb müügiks, kuid see jääb hoopis poodi seisma, siis oleks tegu FP väljaga ning kahju oleks võrdne vastava toote tootmiskuluga. Seega võimaldab täiskulu kõiki tulusid ja kulusid kokku liites valida mudeli, mis aitab poeketil hoida kokku kulusid.

Täpsus (ingl *Accuracy*) ja veamäär (ingl *Error Rate*) hindavad mudeli tööd võttes arvesse kõiki segadusmaatriksi lahtreid. Veamäär on valesti klassifitseeritud näidete arvu ja kogu näidete arvu suhe ning täpsus (ingl *Accuracy*) on õigesti klassifitseeritud näidete ning kõigi näidete arvu suhe[3, p86-88].

$$Acc = \frac{TP + TN}{N}$$

$$Err = 1 - Acc$$

Täpsuse eesmärgiks on tuvastada mudelid, mis teevad kõige vähem vigu, olenemata sellest, kui kulukad tehtud vead on või mis on klassijaotus andmetes. See tähendab, et kui andmetes on positiivse klassi isendeid 90 tükki ning negatiivse klassi isendeid 10 tükki ning disainida kaks mudelit, millest esimene ennustab 100% positiivse klassi ning 0% negatiivse klassi isendeid õigesti, tehes nii kokku 10 viga ($1.0 \times 90 + 0.0 \times 10 = 90$), ning teine mudel, mis ennustab näiteks 80% positiivse ning 80% negatiivse klassi isendeid õigesti, tehes kokku 80 viga ($0.80 \times 90 + 0.80 \times 10 = 80$), siis eelistaks täpsus esimest mudelit, sest see teeb kokkuvõttes vähem vigu.
Ideaalne kasutusjuhtum: Täpsus sobiks ideaalseks headusmõõduks probleemile, kus loeb ainult klassifitseerija poolt tehtud vigade arv ning kus klassifitseerija otsustega kaasnevad kulud ja tulud on võrdsed. Sellisted tingimused olid täidetud varasemalt toodud näidetes "Ilma ennustamine kõrbes"ning "Inimese soo ennustamine". Mõlema näite puhul on teada, et klassijaotus oluliselt muutuda ei saa aja jooksul, sest kõrbes on aastaringselt vihmaste ning päikeseliste arvude suhe sama, samuti on läbi ajaloo püsinud meeste ning naiste arv maailmas omavahel proportsioonis.

Ühtlasi on klassifitseerimisviga erinevatel klassidel sama, sest kummaski näites ei ole eelistatud ühte klassi rohkem kui teist.

Seega on täpsus heaks mõõduks, sest võtab arvesse klassijaotust, kuid jätab erinevad klassifitseerimisvead tähelepanuta.

Geomeetriline keskmine (ingl *Geometric mean*)[3, p100-101] ja keskmine täpsus(ingl *Balanced Accuray*(AUC_b)[8]) on mõõdud, mis võtavad arvesse klassifitseerija soorituse nii positiivsetel kui ka negatiivsetel klassidel, põimides TPR ja TNR.

$$Gmean_1 = \sqrt{TPR \times TNR}$$

$$AUC_b = \frac{TPR + TNR}{2}$$

Mõlemad mõõdud kasutavad kõiki segadusmaatriksi välju. Kuna TPR ja TNR ei sõltu klassijaotusest, ei sõltu sellest ka $Gmean_1$ ning AUC_b.
Ideaalne kasutusjuhtum: $Gmean_1$ ning AUC_b sobiks ideaalseteks headusmõõtudeks mudelile, mis üritab prognoosida ilma sõltumatult asukohast. Varasemalt toodud näite "Ilma ennustamine kõrbes"puhul on selgelt teada, milline on aastaringselt vihmaste ning päikeseliste ilmade suhe, kuid kui kasutada sama klassifikaatorit suvaliselt valitud kohas maailmas, oleks koguaeg ilmaks "päikeseline"pakkuda väga julge otsus. Seega on vaja uut klassifikaatorit, mis ei sõltuks klassijaotusest. Seal tulebki kasuks antud mõõtude immuunsus klasside proportsioonile treeningandmetes.

F-mõõt (ingl *F measure*) on mõõt, mis kombineerib PPV ja TPR ühte skalaarsesse väärtusesse ning selle käigus saab lisada kummalegi headusmõõdule rohkem kaalu teise ees. Selleks saab omistada α-le kõiki reaalarvulisi väärtusi, väljaarvatud 0[3, p103].

$$F_\alpha = \frac{(1+\alpha)[PPV \times TPR]}{[\alpha \times PPV] + TPR}$$

Raskendavaks aspektiks on kaalu valimine kummagi mõõdu jaoks, kuna reaalses elus pole neid sageli välja toodud. Mõõt jätab täielikult välja TN väärtused[3, p104].
Ideaalne kasutusjuhtum: F-mõõt sobiks headusmõõduks juhtudele, kus on PPV ja TPR ühtviisi olulised. Seda saab kasutada näiteks kontrollimiseks, kui aktuaalsed on otsingumootorite poolt tagastatavad veebilehed seoses kasutaja sisestatud

otsinguga. F-mõõt sobib siia hästi, sest jätab tähelepanuta TN välja segadusmaatriksis, mis antud probleemi korral tähendaks kõiki veebilehti, mida otsingumootor otsingutulemuste sisse ei kaasaud ning mille vastu ka kasutajal huvi puudub. Samas laseb F-mõõt omistada erinevad kaalud lehtedele, mille vastu kasutaja küll huvi tundis, kuid mida otsingumootor otsustas mitte kasutajale näidata, ning lehtedele, mis küll otsingule tagastati, kuid mida kasutaja näha ei tahtnud.

Täpsus top K'l (ingl *Prec @ K*) on mõõt, mis näitab PPV väärtus hetkel, kui täpselt K väärtust on ennustatud klassifitseerija poolt positiivseks ($K = TP + FP$)[5].
Ideaalne näide: Sageli ei ole päriseluliste probleemide lahendamiseks saadaval lõpmatult ressursse. Sellistes olukordades tuleb välja selgitada piirarv K positiivse klassi isendite jaoks, millest üle ei tohi minna, ning seejärel disainida klassifitseerija, mis määraks kuni K isendit positiivsesse klassi, omades sealjuures võimalikult head PPV väärtust.

Sarnaseks probleemiks on näiteks stipendiumi jagamine parimatele tudengitele. Selleks, et keegi stipendiumiväärilistest õppuritest sellest ilma ei jääks, võiks idee poolest väljastada selle igaks juhuks kõigile, kuid kahjuks pole see ülikooli jaoks rahaliselt võimalik. Seega tuleb luua mudel, mis annaks kindlale arvule tudengitest stipendiumi ja teeks seda võimalikult ausalt. Klassifitseerija efektiivsuse kontrollimiseks tuleks sel juhul kasutada headusmõõtu Precision @ Top K.

Täpsus saagisel (ingl *Prec @ Rec*) on mõõt, mis näitab PPV väärtust, kui TPR on fikseeritud.
Ideaalne kasutusjuhtum: *Prec @ Rec* sobiks mõõduks klassifitseerijale, mis üritab tuvastada, kas inimesel on D-vitamiini puudus või mitte. Antud klassifitseerija on sarnane mudelile, mida kasutati varasemas näites "Vähihaigete riskigrupi tuvastamine", ainukeseks erinevuseks probleemide vahel on see, et kui vähiravi kõrvalmõjud ei sõltu sellest, kas inimene on haige või terve, siis D-vitamiini manustamisel võib võimaliku üledoosioht sõltuda sellest, kas inimesel on D-vitamiini vaegus või ei ole. Seega on TP ning FP erineva kahjuga. Sellises olukorras saab *Prec @ Rec* ära fikseerida TPR, kui on vaja ära tuvastada näiteks 90% vitamiinivaegusega inimesi ning seejuures vaadata, et mudel ilma vaeguseta inimesi tablettide manustamisele ei suunaks.

4.2 Skoorivad mõõdud

ROC kõver on moodustunud joon graafikul, kus horisontaaltelg tähistab klassifitseerija FPR ja vertikaaltelg TPR. Kuna FPR ja TPR väärtused kuuluvad vahemikku [0,1], siis on ka graafil mõlema telje maksimaalseteks väärtusteks 1. Punktis (0,0) asub klassifitseerija, mis klassifitseerib iga väärtuse negatiivseks ning punktis (1,1) paikneb klassifitseerija, mis liigitab iga näite treeninghulgas positiivsesse klassi. Punktid, mis asuvad diagonaalil, sümboliseerivad klassifitseerijaid, mille töö kvaliteet ei ole parem juhuslikust klassidesse määramisest. Mida rohkem vasakul ja üleval punkt paikneb, seda parem on klassifitseerija[1].

AUC on pindala, mis jääb ROC kõvera alla [1]. Iga punkt ROC kõveral sümboliseerib TPR ja FPR väärtust teatud lävendi juures, seega mida kõrgemal vasakus punktid paiknevad, seda suurem on kõvera alla jääv pindala. Ühtlasi tähendab suur pindala seda, et klassifitseerija suudab hästi eristada positiivseid ning negatiivseid näiteid iga seatud lävendi juures.
Ideaalne kasutusjuhtum: *AUC* sobiks headusmõõduks mudelile, mille eesmärgiks on reastada andmepunktid sobivuse järjekorras. Näiteks ettevõte, mis üritab tööle võtta uusi töötajaid, saaks CVde läbitöötamisel kasutada mudelit, mis ennekõike otsustaks, kas inimene sobib tööle või ei sobi. Sealjuures oleks kasulik, kui mudel oskaks koostada ka vastava pingerea, mille alusel ettevõte saaks inimesi hakata intervjuudele kutsuma. Lisaks järjestuse pakkumisele erineb see *Prec @ K* mõõdu juures toodud stipendiumite jagamise näitest sellepoolest, et ei ole kehtestatud kindlat piirarvu töötajate vastuvõtmise osas. See tuleb kasuks, kui ettevõte ei tea, mitu uut töötajat neil konkreetse ülesande juurde on vaja ning pole välistatud juhtum, kus nad avastavad, et esimene kandidaadist piisab kõigi otsitavate ametipositsioonide täitmiseks.

Logaritmiline kadu (ingl *LogLoss*) ja keskmine ruutviga (ingl *Mean Squared Error*(MSE) on mõõdud, mis hindavad kui head on klassifitseerija poolt pakutud tõenäosused ning kui suured on kõrvalekalded päris klassidest, et element tõesti sinna klassi kuulub[2].

$$LogL = \frac{-\sum_{j=1}^{2}\sum_{i=1}^{m}(f(i,j)log_2 p(i,j))}{m}$$

$$MSE = \frac{\sum_{j=1}^{2}\sum_{i=1}^{m}(f(i,j) - p(i,j))^2}{2m}$$

Valemites esinevate tähtede tähistused: m kõigi andmepunktide arv, $f(i,j)$ tõenäosus, et testhulga i-ndas element kuulub klassi j, kuid tavaliselt võetakse seda lihtsalt funktsioonina, mis väljastab väärtuse {0,1}, $p(i,j)$ tõenäosus, mille klassifitseerija omistas näitele i, et see kuulub klassi j.

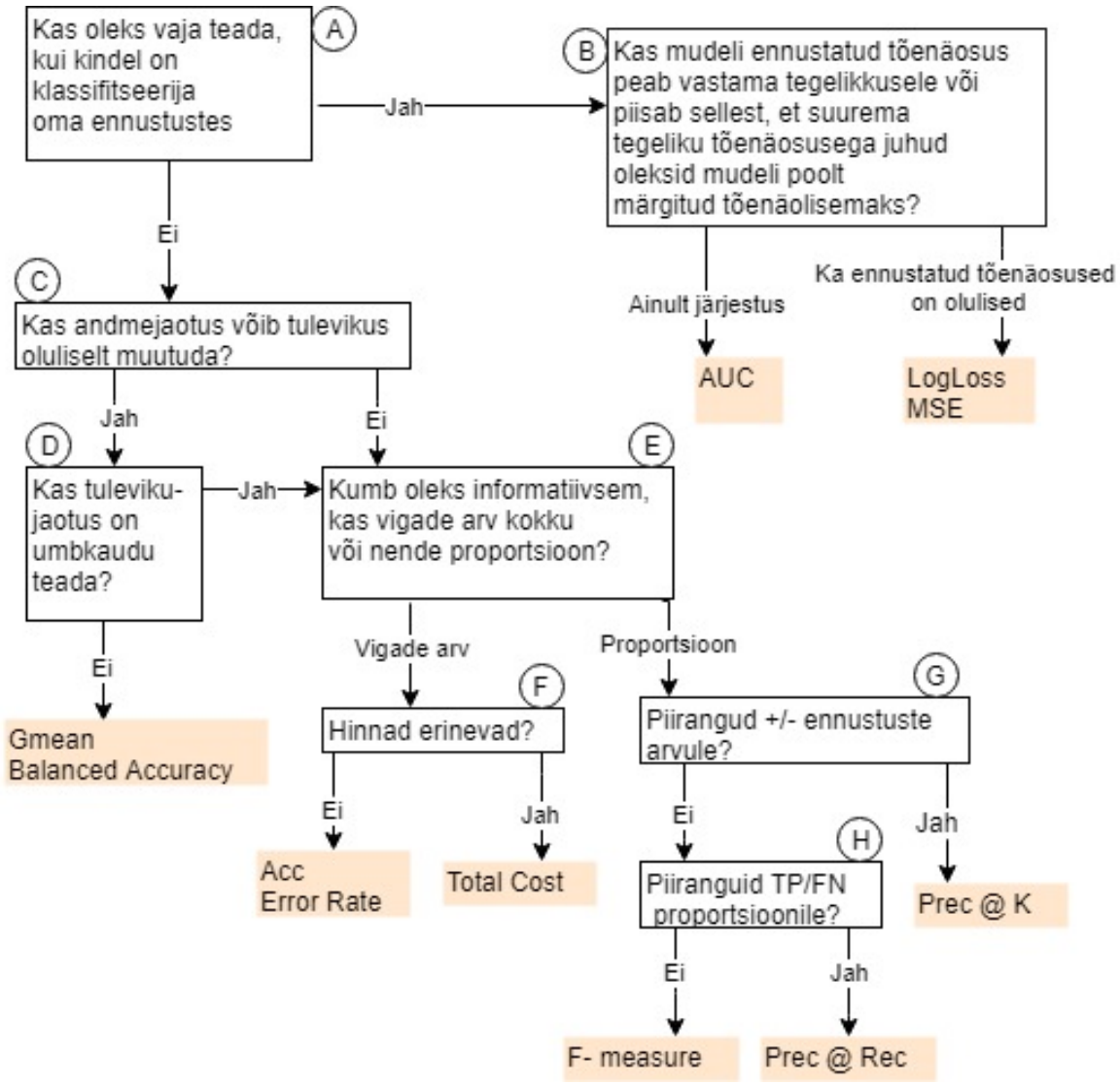
Ideaalne kasutusjuhtum: *LogLoss* ja *MSE* sobiks headusmõõduks varasemalt toodud näitele "Vähihaigete riskigrupi tuvastamine". Lisaks sellele, et klassifitseerimisvead omavad erinevaid tagajärgi, tuleks mudeli töö hindamisel ühtlasi välja selgitada, kui kindel see oma otsustes on. Eelmises lõigus välja toodud järjestav headusmõõt *AUC* küll kontrolliks, et inimesed oleksid järjestatud kõige tervemast inimesest kõige haigemani, kuid see annaks veel vähe informatsiooni selle kohta, kui kindel mudel oma ennustustes on. Seal tulevadki mängu *LogLoss*-i ja *MSE*-i omadused kontrollida ka mudeli pakutud skoori kõrvalekallet tegelikest tõenäosustest. Tänu neile on võimalik kindlaks teha, kui tõenäoline on klassifitseerija arvates, et inimesel esineb vähk või mitte.

Seega, kui nimetatud headusmõõtudega võrrelda kahte mudelit, mis mõlemad ennustasid tervele inimesele vähki, üks neist 51% ning teine 97% tõenäosusega, siis saaks teha mõningaid järeldusi kummagi klassifitseerija töö kohta. Kui esimese arvates on tegu piirijuhuga, mis tähendab, et mudel usub õige pisut rohkem inimese kuulumist riskigruppi, kui seda, et inimene on terve, siis teise klassifitseerija ennustuste kohaselt on inimene kindlasti vähihaige. Järelikult on teise mudeliga kindlasti midagi viga, kui selliseid katastroofilisi möödapanekuid esineb, kuid esimene mudel ei pruugi üldse halvasti töötada. Samuti pakuvad ennustatud tõenäosused tuge arstile. Nähes, et mudeli arvates on patsient haige 97% tõenäosusega saab arst ka ise hinnata patsiendi seisundit ning lõpliku otsuse ise langetada.

5 Headusmõõdu valimine

Nüüdseks on tutvustatud erinevaid headusmõõte ning toodud näiteid, kus neid saaks kasutada, kuid siiski ei pruugi olla selge, kuidas huvipakkuvale probleemile kõige sobivam headusmõõt leida. Selleks on koostatud käesoleva uurimistöö raames graaf, mis peaks juhatama lugeja soovitava mõõduni. Peatüki teises pooles on lahti seletatud graafi tippudes paiknevad küsimused. Lisaks on toodud välja aspektid, mille graaf jätab tähelepanuta ning üks näide, kuidas graafi kasutatakse.

5.1 Graaf sobiva headusmõõdu valimiseks



(A) Kas oleks vaja teada, kui kindel on klassifitseerija oma ennustustes?

Kuigi esmapilgul võib tunduda, et alati on kasulikum teada saada, kui kindel klassifitseerija oma ennustustes on, siis tuleks ennekõike mõelda järgnevate küsimuste peale.

Kas eesmärgiks on lihtsalt juhud klassidesse jaotada ning usaldada täielikult mudeli langetatud otsuseid või soovitakse mudelit kasutada ainult toena otsuste tegemisel? Kas inimmõistus tuleks paremini toime klasside ennustamisega? Kui suured on tagajärjed, mis mudeli ennustustega kaasnevad? Kas juhud peaksid olema järjekorras, vastavalt selle, kui hästi nad sobivad ennustatavasse klassi?

Esiteks võiks mõelda, kas inimene suudaks klassifitseerijast paremini ülesandega hakkama saada.

Kui jah, siis on võimalik mudelit kasutada ainult toena otsuste langetamisel, kus lõplikku otsuse teeb ikkagi inimene. Samas, see võib osutuda väga ajakulukaks, kui mudelit rakendada suurte andmehulkade peal. Seega tuleb mõelda, kui tõsiste tagajärgedega võivad olla klassifitseerimisvead. Väikeste vigade peale pole mõtet palju inimressurssi raisata, kuid suuremate kulude korral on sageli kasulikum otsustusprotsessi kaasata ka inimene.

Võttes näiteks kaks varasemalt mainitud mudelit: esimene pidi prognoosima, kas inimesel võib esineda vähk ning teine arvama ära, millised veebilehed kasutajale huvi võiks pakkuda. Mõlema mudeli puhul tegeletakse suurte andmehulkadega, kuid haige inimese klassifitseerimine terveks on palju tõsisemate tagajärgedega, kui juhtum, kus kasutajale tagastati otsingumootori poolt vale veebileht. Mõlema probleemi korral suudaks inimene ilmselt mudelist parema otsuse langetada. Samas aeg, mis on kulutatud elude päästmise peale, on palju väärtuslikum, kui see, mis kuluks inimesel sobivate veebilehtede otsimise peale.

Seega tuleks skooriva ja tõenäosuslikke mõõte kaaluda siis, kui viimane otsustusõigus tahetakse ikka jätta inimesele. See võib olla tingitud probleemi suurtest tagajärgedest, mida annab vältida, kui inimene lisaks oma panuse klassi määramisse.

Samas juhtudel, kus mudel on võimeline palju paremini klasse ennustama, kui inimene, võib sellele anda täielikult otsustusõiguse. Selliseid masinõppe probleeme saab hästi hinnata segadusmaatriksipõhiste mõõtudega, sest teadmisega, kui kindel mudel oma ennustustes on, pole midagi peale hakata.

(B) Kas mudeli ennustatud tõenäosus peab vastama tegelikkusele või

piisab sellest, et suurema tegeliku tõenäosusega juhud oleksid mudeli poolt märgitud tõenäolisemaks?

Kas oluline on ainult järjekord, kui hästi juhud sobivad ennustatavasse klassi või loeb ka ennustatud tõenäosus?

Varasemalt oli toodud kaks näidet, kus ühes tuli firmal palgata uusi töötajaid ning taheti tugevamad kandidaadid varem tööintervjuule kutsuda ning teine näide, kus mudel pidi leidma protsendi, kui suure tõenäosusega inimesel võib leiduda vähk.

Kuigi esimesel juhul oleks võib-olla ka kasulik, kui mudel oskaks öelda kandidaadi sobivuse töökohale, ei muudaks see firma strateegiat kandidaatide värbamisel. Reaalsus on see, et firma kutsub pingerea alusel kandideerijaid senikaua tööintervjuudele, kuni tööjõu probleem on lahendatud. Teisel juhul aga on tõenäosus olulise tähtsusega, sest ainult järjekorra koostamisel kõige tervematest kõige haigemani ei omaks suurt mõtet, sest kõikide vähihaigetega tuleks tegeleda ning järjekord ei annaks arstile aimu, kes on haige ning kes mitte.

Seega piisab esimese näitega sarnaste probleemide korral ainult järjestavast mõõdust, nagu näiteks AUC, kuid tõsisemate juhtudel tuleks hinnata ka ennustatud tõenäosust, milleks sobivad LogLoss ning MSE.

(C) Kas andmejaotus tulevikus võib oluliselt muutuda? (D) Kas tulevikujaotus on teada?

Kas ühe klassi isendeid võib mingil perioodil tulla massiliselt juurde või hoopis väheneda, samal ajal kui teise klassi isendite arv püsib üldjoontes sama? Kas on teada, kus klassis muudatus esineb ning kui suur muudatus on?

Varasemalt on toodud näited mudelitest, kus ühel tuli ennustada, kas päikeselist või sajust ilma kõrbes ning teisel mudelil tuli täita sama ülesanne, kuid asukoht, kus seda läbi tuli viia, ei olnud teada.

Esimese näite puhul on teada, et kõrbe püsib aastaringselt päikeseline ning aegajalt esineb mõningaid vihmahooge, mis lubab klassifitseerijal ennustada koguaeg päikeselist ilma. Samuti, kui laiendada mudeli teguala kõrbe kõrval asuvatele veidi niiskemate alade peale, teatakse, mis ilmamuutused sellega kaasnevad ning saab modifitseerida mudelit vastavalt.

Samas mudel, mis peaks hakkama saama kogu maailmas, ei saa selle peale lootma jääda, et ilmad aastaringselt samad püsivad. Ühel hetkel võidakse seda kasutada vihmametsades, kus olenevalt perioodist võib koguaeg sadada, samas võib

seda kasutada ka kõrbes, sarnaselt esimesele klassifitseerijalegi. Seega ei saa teine mudel üldse kindel olla, mis teda ees ootab.

Seega tuleks teise mudeli hindamisel kasutada headusmõõte, mis ei võta üldse klassijaotust arvesse. Sellised mõõdud on näiteks $Gmean_1$ ning AUC_b. Samas esimese mudeli puhul ei tohiks klassijaotust tähelepanuta jätta ning tuleb suunduda mõõtude peale, mis klassijaotust oma valemites sisse toovad. Sel juhul tuleks rakendada valitud mõõtu nii hetkelise kui ka arvatava tulevikujaotuse peal.

(E) Kumb oleks informatiivsem, kas vigade arv kokku või nende proportsioon?

Kas eesmärgiks on teha võimalikult vähe vigu või on ka oluline, kus need vead tehakse? Kas on teada täpsed, mis on hind ühe klassifitseerimisvea tegemisel?

Võttes näiteks kaks probleemi. Esimeses tuleb hinnata võistluse finišiprotokollis inimese pikkuse ja kaalu põhjal, kas ta on mees või naine, et oleks võimalik teha meestele ning naistele eraldi arvestus. Teises aga tuleb vastavalt kasutaja sisestatud otsingusõnale tagastada talle huvipakkuvad veebilehed. Esimese näite korral, kui rakendada mõõtu täpsus (ingl *Accuracy*), ütleks see üpris palju mudeli töö kohta, kui suure osa see meestest ning naistest õigesti suutis tuvastada. Teise näite puhul, aga oleks sellest vähe kasu, kui õigesti tehtud ennustuste protsent tagastada, sest kui otsingumootor kasutajale mitte midagi ei tagastaks, saavutatakse juba ligi 100. protsendiline efektiivsus, sest enamik lehekülgi, mis tagastamata jäi, ei oleks kasutajale vähimatki huvi pakkunud. Seega tuleb täpsus (ingl *Accuracy*) valida juhul, kus mõlemad klassid on sama olulised ning pole vahet, kus vead tehakse, tähtis on ainult see, et neid võimalikult vähe on.

Teiseks tuleb mõelda, kas on teada täpselt ühe andmepunkti valestiklassifitseerimisel tuleneva vea hind.

Näiteks probleemis, kus pidi prognoosima, millised tooted on võimalik klientidele maha müüa ja millised mitte, oli väga täpselt teada toote tegemise peale minev kulu ning toote müügist saamata jäänud tulu. Seega saab iga mudeli korral kokku arvutada, mis täpselt nende vead kokku maksavad, ning eelistada mudelit, mis kõige väiksema kuluga on. Selleks sobib mõõt täiskulu (ingl *Total Cost*).

Samas vigadega kaasnevad hinnad ei ole alati teada. Sel juhul tasuks vaadata proportsionaalselt, millised vead esinevad. Eelnevalt toodud otsingumootori näite juures tähendaks see seda, et saab valida mõõdu juures, kumb on kulukam, kas kasutajale jätta teatud veebilehed tagastamata või kui kasutajale tagastatakse

veebilehed, mille vastu tal huvi puudub. Seega tuleb sarnaste ülesannete juures valida proportsioone hindavad mõõdud. Nendeks on antud uurimistöös F-mõõt (ingl *F measure*), täpsus saagisel (ingl *Prec @ Rec*) ja täpsus top K'l (ingl *Prec @ K*).

(F) Kas klassifitseerimisvigade hinnad on erinevad?

Kas ühe klassi valestiklassifitseerimisel kaasnev kulu on suurem, kui teise klassi valestiklassifitseerimisel kaasnev kulu.

Eelneva küsimuse juures polnud finišiprotokolli koostamisel vahet, kas mees ennustati naiseks või naine meheks, viga oli sama. Samas toodete müügipopulaarsuse hindamisel sõltus klassifitseerimisviga tootele valmista kulu ning müügist saadava tulu suurusest.

Seega sobib mõõt täpsus (ingl *Accuracy*), kui klassiftiseerimishinnad on samad, ning mõõtu täiskulu (ingl *Total Cost*), kui hinnad on täpselt teada ning erinevad.

(G) Piirangud +/- ennustuse arvule?

Kas klassifitseerija käsutuses on lõpmatud ressursid või on isendi määramisel ühte klassi kaasnev tulu piisavalt suur, et on kapitali ainult teatud hulga selliste ennustuste tegemiseks?

Mõõt täpsus top K'l (ingl *Prec @ K*) tegeleb just selliste probleemidega, kus ressursid on piiratud. Näitena oli toodud stipendiumi määramise probleem, kus ülikool sai premeerida ainult kindlat arvu õpilasi.

Samas raviasutuse poolt kasutatav mudel, mis pidi tuvastama inimesed, kellel on D-vitamiini defitsiit, võivad põhimõtteliselt positiivseid ennustusi teha niipalju kui tahab, sest D-vitamiini eest peavad maksma inimesed ise. Samuti ei oldud seatud piiranguid otsingumootori näitel, kui palju ta võib veebilehti kasutajale esitada. Seega, kui ressursid ei ole piiratud, tuleks kaaluda mõõte F-mõõt (ingl *F measure*) ning täpsus saagisel (ingl *Prec @ Rec*).

(H) Piirangud õige-positiivsete (TP) või vale-positiivsete (FP) proportsioonile?

Kas klassifitseerijale on vaja seada piirang, et see tuvastaks kindla osa kõikidest positiivsetest näidetest õigesti? Kas on umbmääraselt teada, kui tõsised on klassifitseerimisvead?

Eelnevalt oli nimetatud mudel, mis proovis kindlaks teha kas inimesel on D-vitamiini vaegus või mitte. Tänu mõõdule täpsus saagisel (ingl *Prec @ Rec*) saab määrata, et klassifitseerija peab vähemalt 90% D-vitamiini vaegusega inimesi

õigesti tuvasta ning alles siis keskenduma sellele, ega liiga paljudele ilma defitsiidita inimestele põhjuseta D-vitamiini välja ei kirjutatud.

Mõõt F-mõõt (ingl *F measure*) aga aitaks leida konkreetse tasakaalu proportsioonide vahel, kellele D-vitamiini ei oleks tohtinud anda ning kellele D-vitamiin jäi andmata. Mõõdu F-mõõt (ingl *F measure*) kasutamise teeb raskeks see, et kaalu määramisel peab suhteliselt täpselt teadma, kui tõsised on mõlemad vead.

Seega, kui klassifitseerimisvigadega kaasnev kahju täpselt teada ei ole, aga on teadmine, et ühe klassi isendeid tuleks võimalikult palju õigesti klassifitseerida, siis tuleks valida mõõt täpsus saagisel (ingl *Prec @ Rec*). Samas, kui vigade suurus on umbmääraselt teada, saab leida optimaalse tasakaalu TPR ja PPV vahel, kui kasutada mõõtu F-mõõt (ingl *F measure*).

5.2 Graafi kasutamise näide

Antud peatükk demonstreerib, kuidas näeb välja graafi kasutamine, kui selle abil üritada valida sobivat headusmõõtu populaarsele suhtlusrakendusele *Tinder*. Tegu on rakendusega, mis annab kasutajale ette teiste inimeste pilte ning kasutaja peab otsustama, kas inimene meeldib talle või mitte. Seega tuleks *Tinder*il kasutada klassifitseerimist, mis tuvastab ära inimesed, kes võiksid konkreetsele kasutajale meeldida. Sellega saab parandada rakenduse kasutusmugavust, mis tähendab, et kasutajale ei peaks esitama inimeste profiile, kellest ta absoluutselt huvitatud ei oleks.

Alustuseks tuleks mõõdu valimist alustada küsimusest tipust (A) ehk kui kindel peaks klassifitseerija oma ennustustes olema.

Tuleb mõelda, kas *Tinder*il oleks kasulik, kui inimeste profiilid oleksid sobivuse järjekorras, alustades inimesest, kes tõenäoliselt kasutajale kõige rohkem meeldib ning lõpetades inimestega, kellest kasutaja tahaks eemale hoiduda. Kuna *Tinder* teenib raha ka reklaamide näitamisega, siis on oluline, et inimesed veedaksid võimalikult palju aega rakenduses. Kui aga kasutajad kõige sobilikumad inimesed kohe üles leiaksid, oleks nende veedetud aeg rakenduses väga lühike.

Teiseks tuleks mõelda, kas mudelit kasutatakse toena või lastakse ise kõik otsused teha. Kuna profiilide hulk, mis tuleks läbi töötada on väga suur ning klassifitseerimisvigadel ei ole tõsiseid tagajärgi, siis ei ole mõtet inimressurssi raisata. See tähendab, et mudelil lastakse kõik otsused ise teha.

Seega ei ole oluline, kui kindel mudel oma ennustustes on ning tuleks kõrvale jätta skoorivad ja tõenäosuslikud mõõdud ning suunduda segadusmaatriksipõhiste mõõtude juurde.

Tipus (C) on küsimus, kas andmejaotus võib muutuda ja kuhu poole. See tähendaks *Tinderi* puhul seda, kas järsult võib juurde tulla kasutajale (mitte) meeldivate inimeste profiile. Kuna tegu on asukoha põhise rakendusega, mis võtab vaatluse alla profiilid, mis asuvad kasutajale lähedal, siis on oht, et kui kasutaja reisile läheb, võib ta sattuda kohta, kus ükski inimene talle ei pruugi meeldida. Seega tuleks sellel puhul kasutada mõõtu AUC_b või $Gmean_1$, mis ei võta arvesse klassijaotust ning arvutab keskmise, kui hästi suutis proportsionaalselt mudel tuvastada huvipakkuvaid ning eemaletõukavaid profiile.

Samas, kui *Tinder*i uuringutest selguks, et igas piirkonnas oleks kasutajale meeldivaid ning mitte meeldivaid inimesi, siis saab kaaluda ka klassijaotust arvesse võtvaid mõõte. Tipus (E) tuleb vastata küsimusele, kumb on informatiivsem, kas vigade arv või proportsioon. Kuna kasutaja profiile on väga palju, siis nagu 'otsingumootori' näite juures, ei ütleks ka ilmselt *Tinder*i puhul midagi protsent, kui palju profiile suutis mudel õigesti klassifitseerida. Samuti ei ole täpselt teada, mis on valeklassifitseerimise hind.

Seega tuleks liikuda edasi proportsioone arvesse võtvate mõõtude juurde. Tipus (G) küsitakse, kas on seatud piiranguid, palju võib profiile kasutajale meeldivaks lugeda. Vastus sellele on, et piiranguid ei ole. Profiili lugemine "meeldivaks"ei maksma *Tinder*ile midagi ning mida rohkem nad "meeldivaid"profiile tuvastavad seda kauem veedavad ka inimesed aega nende rakendustes ning see tähendab *Tinder*ile suuremat tulu.

Seega saab liikuda tipu (H) juurde. Tipp teeb kindlaks, kas on vaja seada piiranguid TP ja FN proportsioonile. TP ja FN piirangud tähendaks, et *Tinder* peaks eelistama mudeleid, mis on mingi osa päriselt meeldivaid inimesi õigesti ära tuvastanud ning alles seejuures valima mudeli, mis on seda teinud kõige efektiivsemalt ehk mis on lugenud kõige vähem eemaletõukavaid inimesi "meeldivateks". Seda, kas piirangut on vaja, saab testida, kui kehtestada nõudmine, et 80% meeldivatest inimestest tuleb õigesti klassifitseerida, ning selleks disainida kaks mudelit. Üks neist tuvastab 100st meeldivast inimesest 80 positiivset ära, täites sellega kehtestatud piirangu. Samas loeb selle käigus ka 120 ebameeldivat inimest positiivseks. Teine aga tuvastab 100st meeldivast inimesest ära 70, kuid seal juures loeb ainult 5

ebameeldivat inimest "meeldivaks". Tuleb välja, et kuigi piirang soosiks esimest mudelit, töötab teine mudel palju paremini. Seega näeb, et piirangut ei ole vaja seada ning järele jääb mõõt F-mõõt (ingl *F measure*).

5.3 Puudused graafi juures

Kui tulevikujaotus pole täpselt teada, siis ei saa graafi poolt pakutavate mõõtudega arvesse võtta juhte, kus klassifitseerimisvead on erinevad. Näiteks tahaks päikese-energiale pühenduv firma prognoosida, kui palju tuleb kasu erinevates maailma paikades päikesepaneelide üles seadmine. Neile tähendavad päikeselised ilmad tulu ning vihmased ilmad kulu. Juhul, kui klassifitseerija ennustab päikeselise ilma vihmaseks, jääks saamata paneelide poolt toodetud energia, vastupidise ennustuse korral seisaks paneelid lihtsalt vihma käes ning nende ostmisele kuluv raha oleks kulutatud ilma asjata. Tähtsama klassi tuvastamiseks tuleks võtta ühe päikeselise päeva pealt saadav tulu ning paneelide ostmisele minev kulu ning seeläbi vaadata, kumba ilma on olulisem õigesti ennustada. Toodud näitele oleks vaja mõõtu, mis ei võtaks arvesse klassijaotust, kuid mis seaks ühele klassile suurema tähtsuse. Graafi poolt toodud mõõdud $Gmean_1$ ning AUC_b täidavad sellest ainult esimest tingimust.

Samuti ei leidu graafis tõenäosuslikke mõõte, mis ei sõltuks klassijaotusest. See tähendab seda, et mudel, mille ülesandeks on inimesi klassifitseerida, kas haigeks või terveks, võib tagastada suurepäraseid tulemusi näiteks külmetushaiguste määramisel, kuid katastroofilisi tulemusi nakkushaigustel. *LogLossi* ning *MSE* andmetel töötaks mudel rahuldavalt, kui külmetusega inimesi on ülekaalus ning haiguste proportsioon püsiks muutumatu. Juhul, kui nakkushaiguste levikus toimuks aga äkiline puhang, suureneks nende haiguste proportsioon, millel mudeli töö nii hea ei olnud ning mudeli kasutegur väheneks märgatavalt. Seda aga ei osanud ette näha *LogLoss* ning *MSE*, sest need võtsid arvesse ainult hetkelist klassijaotust, kus külmetushaigusega inimesi oli ülekaalus.

6 Kokkuvõte

Käesolev töö keskendus ainult binaarsele klassifitseerimisprobleemide hindamisele. Kõigepealt andis töö ülevaate masinõppest ning klassifitseerimisest. Seejärel peatus pikemalt klassifitseerija hindamise juures, kus rääkis pikemalt segadusmaatriksist ning kuidas klassijaotus ning erinevad vigade hinnad võivad klassifitseerija hindamise taktikat muuta. Järgnevas peatükis tõi töö välja klassifiteerija hindamise juures kasutatavate headusmõõtude definitsioonid koos näidetega, kus oleks kasulik konkreetseid mõõte kasutada. Uurimistöö tuum asus peatükis "Headusmõõdu valimine", kus esitati graaf, milles tippudeks olid küsimused, mida headusmõõdu valimisel tuleks küsida ning rippuvateks tippudeks headusmõõdud, milleni jõuti seoses küsimuste vastatuga.

Segadusmaatriksipõhistest headusmõõtudest võeti vaatluse alla täiskulu (ingl *Total Cost*), täpsus (ingl *Accuracy*), veamäär (ingl *Error Rate*), geomeetriline keskmine (ingl *Geometric Mean*), tasakaalustatud täpsus (ingl *Balanced Accuracy*), F-mõõt (ingl *F measure*), täpsus saagisel (ingl *Prec @ Rec*), täpsus top K'l *Prec @ K* ning skoorivatest ja tõenäosuslikest mõõtudest on vaatluse all logaritmiline kadu (ingl *LogLoss*), keskmine ruutviga (ingl *Mean Squared Error*) ning ROCi alune pindala (ingl *Area Under the ROC Curve (AUC)*). Nende valimisega püüti katta võimalikult palju erinevaid külgi, mida klassifitseerija töö juures oleks võimalik hinnata.

Uurimistöö jõudis järeldusele, et headusmõõdu valimisel oleks vaja kaaluda, kas klassifitseerimisel on oluline, kui kindel on klassifitseerija oma ennustustes. Ühtlasi tuleks mõelda, milliste andmehulkade peal hiljem klassifitseerijat plaanitakse kasutama hakata ning kas andmejaotus nendes hulkades võib drastiliselt muutuda. Kuna osad mõõdud hindavad klassifitseerija töö juures üldist vigade arvu ning vigadega kaasnevat kulu, aga teist tüüpi mõõdud hindavad hoopis erinevaid proportsioone vigade juures, siis tuleks mõõt valida vastavalt sellele, kumb nimetatud tüüpidest ütleb klassifitseerija töö efektiivsuse kohta rohkem. Samuti saab erinevate mõõtudega arvesse võtta vigade erinevat hinda ning seada piiranguid klassifitseerija tööle, näiteks kui palju see võib teatud klassi elemente määrata.

Tööd saaks laiendada veel teistele klassifitseerimisliikidele nagu näiteks juhud, kus ennustatavaid klasse on rohkem kui kaks. Samuti võib headusmõõtude valimist aidata soodustada regressioonil.

Viidatud kirjandus

[1] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[2] César Ferri, José Hernández-Orallo, and R Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27–38, 2009.

[3] Nathalie Japkowicz and Mohak Shah. *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

[4] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2015.

[5] Charles Parker. An analysis of performance measures for binary classifiers. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 517–526. IEEE, 2011.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Naeem Seliya, Taghi M Khoshgoftaar, and Jason Van Hulse. A study on the relationships of classifier performance metrics. In *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*, pages 59–66. IEEE, 2009.

[8] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[9] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Egert Georg Teesaar**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

   Tüübituletus neljandat järku loogikavalemitele

   mille juhendajad on Mari-Liis Allikivi ja Meelis Kull

   1.1 reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

   1.2 üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 14.05.2018