TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Tanel Kärvet 193312IABB

OPTIMAALSE MASINÕPPE VÕIMALUSE VALIMINE KINDLUSTUSSEKTORIS

Bakalaureusetöö

Juhendaja: Inna Švartsman

MSc

Kevin Lehtsalu

BSc

Tallinn 2022

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Tanel Kärvet

17.05.2022

Annotatsioon

Antud lõputöö eesmärgiks oli välja valida optimaalne masinõppe võimalus ettevõtetele, kes tegutsevad kindlustusvaldkonnas. Optimaalse võimaluse valimiseks analüüsiti nelja Pythoni masinõppe teeki: TensorFlow, PyCaret, Scikit-Learn ja fastai. Analüüs tehti litsentsi, andmete turvalisuse, dokumentatsiooni, õppimiskõvera, masinõppe koodi, masinõppe võimaluste ja tulemuste osas, mis kõik sisaldasid endas spetsiifilisemaid punkte.

Iga teegi analüüsimiseks loodi täpselt samasuguse andmekogumiku ja reeglite põhjal masinõppe projektid, mille lõppeesmärgiks oli leida kliendirisk, mida saaks kasutada kindlustussektoris hinnastamise mudeli parandamiseks. Näitajatele anti hinnangud, mida kasutati koos AHP-ga leitud näitajate osatähtsustega, et arvutada teekidele lõplik hinne.

Analüüsi ja leitud hinnete põhjal selgus, et kindlustussektoris on optimaalne kasutada TensorFlow ja PyCareti võimalusi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 44 leheküljel, 5 peatükki, 2 joonist, 1 tabel.

Abstract

Finding optimal machine learning approach in insurance industry

The aim of this thesis is to find optimal machine learning approach in insurance industry. To select optimal approach, four Python machine learning libraries were analyzed: TensorFlow, PyCaret, Scikit-Learn and fastai. The analysis was based on licence, data security, documentation, learning curve, machine learning code, machine learning opportunities and outcomes, all of which contain more detailed specific aspects that are explained in the thesis.

To analyze each selected library, a machine project with the same dataset and rules was created. The aim of previously named project was to find client risk which later could be used to improve insurance industry pricing models. All indicators were given estimates which were combined with the weights found by AHP. The sums of combinations were the final scores for the libraries.

Based on the analysis and final scores, most optimal libraries are TensorFlow and PyCaret.

The thesis is in Estonian and contains 44 pages of text, 5 chapters, 2 figures, 1 table.

Lühendite ja mõistete sõnastik

AHP	Inglise keeles: *Analytic Hierarchy Process*, eesti keeles: analüütiline hierarhia protsess
Aktuaar	Kindlustusmatemaatik
Boosting algoritm	Võimendustüüpi algoritm
CPU	Inglise keeles: *Central Processing Unit*, eesti keeles: protsessor
CR	Inglise keeles: *Consistency Ratio*, eesti keeles: kokkulangevuse suhtarv
GLM	Inglise keeles: *Generalized linear model*, eesti keeles: üldistatud lineaarne mudel
GPU	Inglise keeles: *Graphics Processing Unit*, eesti keeles: graafikakaart
Low-code	Vähese koodiga kirjutatav
Mudel	Valem, mida kasutatakse ülesandele tõenäolise vastuse leidmiseks
Teek	Inglise keeles: *Library*. Funktsioonide või meetodite kogum

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Kindlustussektoris töötab palju ettevõtteid, kes kasutavad toodete hinna kujundamiseks lineaarseid ning keskmise keerukusega mudeleid. Antud mudelid ei suuda reaalsust väga täpselt peegeldada, kuna need on loodud aktuaaride poolt neile teadaolevat äriloogikat silmas pidades [1]. Paratamatult ei ole inimene võimeline looma mudeleid nii paljude parameetritega nagu arvuti. Ebaefektiivsed hinnastamise mudelid võivad tekitada ettevõttele kahju nii raha kui ka kaotatud klientide näol [2]. Vaatamata sellele, et ettevõtted koguvad klientide kohta märkimisväärselt palju andmeid, pole suur osa ettevõtetest jõudnud masinõppeni [3].

Antud lõputöö eesmärk on välja valida masinõppe võimalus, mida käesoleval ajahetkel on kindlustussektoris tegutseval ettevõttel optimaalne kasutada. Optimaalse võimaluse valimiseks analüüsitakse erinevaid Pythoni masinõppe teeke. Analüüsimiseks valiti välja kaks suuremat ja tuntumat teeki ning kaks väiksemat ja uuemat teeki, mis veebist leitava informatsiooni põhjal on teinud masinõppe kasutamise võimalikult lihtsaks. Samuti on mõlemas suuruse kategoorias olemas üks teek, mis võimaldab kasutada süvaõppe algoritme. Antud kriteeriumitele vastasid ideaalselt TensorFlow, PyCaret, Scikit-Learn ja fastai mistõttu uuritakse antud töös just neid teeke.

Masinõpet on võimalik rakendada kindlustussektori väga paljudes erinevates protsessides. Sobivaima teegi leidmiseks testitakse teeke ühesuguse andmekogumikuga ettevõtte poolt pakutava toote ja seda kasutavate klientide kohta. Teekide juures vaadatakse järgmisi näitajad: litsents, andmete turvalisus, dokumentatsioon, õppimiskõver, masinõppe kood, erinevate algoritmide kasutamise võimalus ning tulemuste leidmise kiirus ja kvaliteet. Tulemuste analüüsimisel kasutatakse AHP-d (*Analytic Hierarchy Process*), mis võimaldab anda sobiva kaaluga hinnangu igale eelnevalt väljatoodud näitajale.

Lõputöö autor on töö jaganud kolme suuremasse osasse.

Metoodika peatüki objekti osas antakse esmalt täpsem ülevaade kindlustusettevõtete tööst, sealhulgas kuidas käib paljudes ettevõtetes tänapäeval hinnastamine, milliste probleemidega tuleb silmitsi seista ning miks soovitakse aina enam kasutada hinnastamisel võimalusi, mida pakub masinõpe. Seejärel antakse lühike ülevaade masinõppest ning selle erinevatest kasutusvõimalustest kindlustussektoris.

Sellele järgnevad ülevaated kõikidest töös kasutatud tööriistadest. Alapeatükk protsessist kirjeldab detailsemalt kõiki püstitatud alamülesandeid alates arenduskeskkonna seadistamisest, teekidega tutvumisest kuni arenduse ja analüüsini, mis kõik olid olulised vaheetapid enne püstitatud eesmärgini jõudmist.

Peatükk teekide ülevaatest kirjeldab esmalt kõiki näitajaid koos spetsiifilisemate punktidega, mida iga masinõppe teegi juures hinnatakse. Seejärel antakse ülevaade kõikide teekide kohta, tuginedes eelnevalt kirjeldatud punktidele.

Analüüsi ja tulemuste peatükis võrreldakse kõiki teeke samade punktide alusel, mida varasemalt kasutati ülevaate tegemisel. Iga teegi kõikidele näitajatele antakse hinnang skaalal 1–10. Peatüki lõpus arvutatakse teekide hinded kasutades näitajate hinnanguid ning neile vastavaid AHP abil leitud osakaalusid. Põhinedes arvutatud hinnetele tehakse lõplikud järeldused.

Antud tööl on väga suur potentsiaalne väärtus ka lõputöö väliselt. Analüüsi tulemustest lähtuvalt alustatakse ettevõttes esimese masinõppe projektiga, kus kasutatavaks teegiks on lõputöö käigus välja selgitatud optimaalne valik. Kasutades valitud Pythoni teeki, soovitakse leida võimalikult efektiivne mudel kliendiriski väljaselgitamiseks, mille põhjal arvutatakse välja kliendile hind.

2 Metoodika

Antud peatükk on jagatud kolme osasse. Esmalt antakse ülevaade kindlustussektorist, hinnastamisest tänapäeval ning masinõppe rakendamise võimalustest kindlustussektoris. Seejärel tutvustatakse lühidalt töös kasutatud tööriistu. Viimane alapeatükk kirjeldab lõputöö protsessi alates teekidega tutvumisest kuni teekidele hinnete andmiseni.

2.1 Objekt

Aktuaariteadustes on statistilise õppe mudeleid kasutatud juba pea 40 aastat. Algselt kasutati hinnastamisel lineaarseid mudeleid ja GLM-e (*Generalized linear model*). Aja möödudes, tehnika ja ka statistika valdkonna arenedes avanesid võimalused keerukamate ja paindlikumate mudelite kasutamiseks, mis olid märkimisväärselt efektiivsemad tavalistest lineaarsetest mudelitest. Vaatamata kõikidele uutele avanenud võimalustele, on suur osa kindlustussektoris töötavatest ettevõtetest implementeerimata jätnud uudseid ideid ja lahendusi [1].

Kindlustussektoris on kliendiriski hindamine väga kriitilise tähtsusega, kuna pakutava toote eest tuleb kliendile arve esitada enne reaalsete kulude tekkimist. Sobiva hinna kujundamiseks peavad ettevõtted suutma prognoosida, kui palju ja millises summas kahjunõudeid perioodil esitatakse. Ebaefektiivne mudel võib tähendada heale kliendile liiga kallist hinda ja halvale kliendile liiga soodsat hinda [2]. Hinna määramise mudel on kindlustusettevõtte kõige suurem ärisaladus ning edu võti.

Kuigi tehnoloogia on edasi arenenud, eelistatakse endiselt kasutada lineaarseid- ja logistilisi mudeleid, sest neid on lihtsam tõlgendada ja suuremal määral täidavad nad soovitud vajaduse. Antud mudelid jäävad tihtipeale kindlustusvaldkonna jaoks liiga pealiskaudseks, kuna nad ei peegelda päris elu eriti täpselt [4]. Lisaks sellele on GLM-id üsna ranged, millest tulenevalt on nendega tülikas luua mudeleid, kus tulemus sõltub erinevate tunnuste kombinatsioonidest [1].

Sõidukikindlustuse puhul on sageli üheks hinda mõjutavaks kriteeriumiks kliendi elukoht. Erinevatest uuringutest on selgunud, et kõige sagedamini juhtuvad õnnetused tihedalt asustatud piirkondades. Sellest tulenevalt on suurema rahvaarvuga piirkondades sõiduki kindlustamine kallim. Antud näitaja üldistab kliendiriski liiga palju, kuna kõrvale võib jääda palju statistiliselt olulisi tunnuseid [2].

Reaalsust kirjeldavad palju paremini mittelineaarsed mudelid. Parema mudeli saamiseks luuakse ise uusi tunnuseid juurde, sageli on need just muutuja x erineva astme polünoomid. See võib olla tülikas, kuna tunnuste loomine ja nende mudelisse lisamine tuleb ära teha käsitsi. Peale täienduste tegemist tuleb aktuaaril mudelit uuesti testida, mida tuleb aktuaaril taaskord ise teha [1].

Hinnastamisel puututakse kokku mitmete erinevate juriidiliste nõuetega, millest üheks suurimaks on see, et hinnad peavad olema eetilised ja mittediskrimineerivad. Euroopa Liit on kehtestanud reeglid, mis keelavad teatud isikuomaduste põhjal, nagu näiteks sugu ja rass, arvutada välja kliendispetsiifilist hinda [5].

Andmeid kogutakse sageli mittestruktureeritud kujul, näiteks vabatekstina. Selleks, et aktuaarid andmete põhjal paremaid mudeleid koostada saaks, tuleb andmed esmalt viia struktureeritud kujule, et oleks võimalik kasutada lähenemisi, mida on järgitud juba pikki aastaid. Andmete üldistamine tähendab aga seda, et potentsiaalselt läheb mingi osa vajalikust infost kaduma, mis omakorda avaldab mõju uuele loodavale mudelile [1].

Suuremad muutused kindlustussektori toodete hinnastamisel on hakanud toimuma just viimastel aastatel. Populaarsuse kasvu illustreerib väga hästi teaduslike artiklite avaldamiste statistika, kus käsitletakse hinnastamist kasutades masinõpet [1]. Suurenenud huvi masinõppe vastu on peamiselt tingitud arvutusvõimsusest, mis on aastast aastasse läinud aina soodsamaks. Lisaks sellele mõistetakse, et nullist analüüsi tegemise asemel on mõistlikum ja ressursiefektiivsem kasutada erinevate masinõppe võimaluste abi [6]. Kindlustamine on muutunud oluliselt personaalsemaks ning konkurentsis püsimiseks on oluline kasutusele võtta võimalused, mida pakub masinõpe [1].

Masinõpe on üks osa tehisintellektist, mis aitab arvutil mõista ja käituda sarnaselt inimestele. Erinevalt traditsioonilisest arendusest, kus pannakse paika reeglid, mille järgi süsteem toimetab, kasutatakse masinõppes andmete põhjal treenimist ehk otsuste

tegemist ilma, et see oleks inimese poolt ette programmeeritud [7]. Oma töös kasutab masinõpe erinevaid algoritme ja närvivõrke, mis võimaldab arvuti muuta ajapikku oluliselt võimekamaks ning tema poolt leitud tulemused täpsemaks. Algoritm on kui reeglite kogum, mille järgi tuleb ülesannet täita ning närvivõrk on algoritmide kogum, mida kasutatakse inimese aju jäljendamiseks [8].

Sisendandmetena kasutatakse masinõppes sageli erinevaid tekstilisi andmeid, graafikuid, pilte ning helisalvestisi. Peale sisendite ette andmist algab protsess, mis otsib andmetest erinevaid seoseid. Leitud seoste põhjal saab arvuti hiljem prognoosida, milline peaks olema etteantud andmete põhjal ülesande tulemus. Sõltuvalt sisendandmete tüübist on võimalik valida kolme erineva masinõppe liigi vahel, mida projektis kasutada: juhitud-, juhtimata- ja stiimulõpe [7].

Kergemate ülesannete puhul, kus reegleid on vähe ning kõiki seoseid lihtne hallata, võib piirduda traditsioonilise arendusega ning tihtipeale masinõppe kasutuselevõtt oleks liigne ressursi kulutamine. Kindlustussektoris sõltub hind väga paljudest erinevatest kriteeriumitest, mistõttu reegleid traditsioonilise arenduse jaoks on pea võimatu paika panna, veel vähem on võimalik lõpuks kõiki seoseid mõista. Masinõppe jaoks piisab vaid ühekordsest koodi kirjutamisest, mida lisanduvate andmete korral on võimalik uuesti käivitada, et saada veelgi täpsem mudel. Lisaks sellele on võimalik leida sisendandmete vahel seoseid, mida inimene ei pruugiks ise muidu tähele panna. See on vägagi oluline võimekus, tänu millele saab kliendile pakkuda võimalikult õiglast hinda [9].

Masinõppe kasutamisel kindlustussektoris tuleb arvestada mitmete faktoritega, mis võivad raskendada areneva tehnoloogia integreerimist olemasolevate äriprotsessidega. Hinnastamisel soovitatakse kasutada peamiselt võimendustüüpi ehk *boosting* algoritme, mis kombineerivad nõrgalt seletavaid mudeleid, et luua tugevamaid mudeleid, mis suudavad tulemusi täpsemini ennustada [10]. Algoritmid nende mudelite taga on tihtipeale väga keerulised. Keerukus avaldub ka tulemustes, mida võrreldes lineaarsete mudelitega on oluliselt raskem tõlgendada. Lisaks sellele nõuavad võimendustüüpi algoritmid andmekogumike analüüsimiseks rohkem ajalisi- ja arvutuslikke ressursse [11]. Probleeme võivad tekitada ka eelpool mainitud õigusaktidest tulenevad piirangud, mis takistavad piisavalt efektiivse ja reaalsust hästi seletava mudeli kasutamist.

Tänapäeva erinevad seadmed, sealhulgas ka sõidukid, koguvad suurtes kogustes erikujulisi asukoha ja liikumisega seonduvaid andmeid, mis annavad hea ülevaate kliendi sõiduharjumustest. Sellise detailsusega andmed aitaksid mudeleid oluliselt parandada, kuid asukoha- ning liikumisandmete kasutamine võib rikkuda inimeste privaatsusõiguseid. Lisaks sellele genereeritakse antud andmeid järjepidevalt suurtes kogustes juurde, mis alguses mõjutab mälumahtu ning hiljem arvutuslikku kiirust [12].

Arvestades asjaolu, et suur osa ettevõtetest on turul olnud juba aastaid, siis üldjuhul koguneb selle aja jooksul klientide kohta väga suures koguses erinevaid andmeid. Paari aasta vanuste andmete olemasolu võib olla juba piisav, et leida andmetest peidetud seoseid, mille abil on võimalik märkimisväärselt parandada erinevate kindlustussektori toodete hinnastamise mudeleid või teisi sektorispetsiifilisi protsesse [3].

Masinõpet on võimalik rakendada erinevate kindlustusvaldkonda puudutavate ülesannete juures nagu näiteks hinnastamine, ideaalse kliendi leidmine parema hinna pakkumiseks, kliendile sobivaima toote pakkumine, kindlustuspettuste avastamine. Sõltuvalt ülesandest ja sellele kohaldatavatest nõuetest tuleb valida mudeli loomiseks sobiv algoritm, mis sobib olemasolevate andmete, lubatava õppimisviisi ja eeldatava lõpptulemuse kujuga [13]. Masinõpet võib kasutada ka ainult andmetest seoste leidmiseks. Seoseid saavad aktuaarid edasi analüüsida ning sobivuse korral lisada olemasolevatesse mudelitesse.

2.2 Tööriistad

Python on maailma üks enim kasutatavaim programmeerimiskeel, mida sageli kasutatakse veebilehtede ja tarkvara arendamiseks, protsesside automatiseerimiseks, andmeanalüüsi läbiviimiseks ning andmete visualiseerimiseks. Kuna Python on üsna algajasõbralik, siis on see väga laialdaselt kasutusel ka inimeste seas, kes pole elukutselt arendajad [14]. Suure kasutajaskonna olemasolu on viinud selleni, et kõikidele kasutajatele on kättesaadav väga suur arv erinevaid teeke, sealhulgas masinõppe teeke, mis lihtsustavad ja kiirendavad arendusprotsessi veelgi enam. Pythonit eelistatakse masinõppe jaoks, kuna see võimaldab visualiseerida andmeid erineval kujul graafikutena [15].

Jupyter Notebook on veebirakendus, mis võimaldab kirjutada ja jagada koodi rohkem kui 40 keeles, sealhulgas Pythonis, R-is ja Julias. Tänu lihtsale kasutajaliidesele ja

võimalusele koodi käivitada osade kaupa, on Jupyter Notebook üks eelistatumaid arenduskeskkondi, mida kasutada masinõppeks [16].

AHP ehk analüütiline hierarhia protsess on keeruliste otsuste analüüsimise ja organiseerimise meetod, mis kasutab nii matemaatikat kui ka psühholoogiat. AHP koosneb kolmest suuremast osast: eesmärgist või lahendatavast probleemist, kõikidest alternatiividest ning kriteeriumitest, mille põhjal alternatiive hinnatakse. Vaatamata sellele, et meetod loodi juba 1970ndatel, kasutatakse seda endiselt väga suure hulga inimeste poolt. AHP-d eelistatakse, kuna see ei nõua, et kõik otsustuskriteeriumid oleksid sama kaaluga, vaid võimaldab kasutajal määrata iga kriteeriumi tähtsuse teiste tunnuste suhtes [17].

TensorFlow on Google Brain teami poolt loodud avatud lähtekoodiga platvorm masinõppe mudelite loomiseks ja kasutamiseks. Platvorm koosneb mitmekülgsetest tööriistadest, mis on mõeldud arendajatele, ettevõtetele, teadlastele kui ka lihtsalt huvilistele, kes soovivad luua masinõppel töötavaid lahendusi nii serveritele, pilve- kui ka mobiilisüsteemidele, brauseritele ja paljudele teistele JavaScriptil põhinevatele platvormidele. Google kasutab TensorFlow'd paljudes enda poolt pakutavates toodetes nagu näiteks Gmail, Translate, Android ja YouTube [16]. Antud töös kasutatakse võimendustüüpi mudelite treenimiseks TensorFlow Decision Forests teeki ning süvaõppe võimaluste analüüsimiseks TensorFlow Core teeki, mida edaspidi käsitletakse ühiselt kui TensorFlow'd.

PyCaret on *low-code*, avatud lähtekoodiga Pythoni masinõppe teek, millega on võimalik suurel hulgal automatiseerida masinõppe töövoogu. PyCaret ühendab paari väga olulist masinõppe raamistikku ja teeki nagu näiteks Scikit-Learn, LightGBM ja Optuna. Teek loodi eesmärgiga vähendada aega, mis kulub andmete testimise ja masinõppe algoritmide hindamise ning võrdlemise peale [18].

Scikit-Learn on avatud lähtekoodiga masinõppe teek, mis on loodud NumPy, SciPy ja Cythoni baasil. Teek sisaldab suurel hulgal erinevaid masinõppe algoritme, mudeleid ja tööriistu ning toetab nii juhendatud kui ka juhendamata õpet [19].

Fastai on süvaõppe teek, mida kasutades on võimalik vähese vaevaga jõuda soovitud tulemusteni. Teegi autorite eesmärk oli luua teek, mis oleks paindlik, kiire ja võimalikult lihtne arendajale kasutada, ilma, et tuleks üheski kriteeriumis järeleandmisi teha. Fastai

on ehitatud PyTorchi baasil ning üritab võimendada viimase paindlikkust veelgi enam [20].

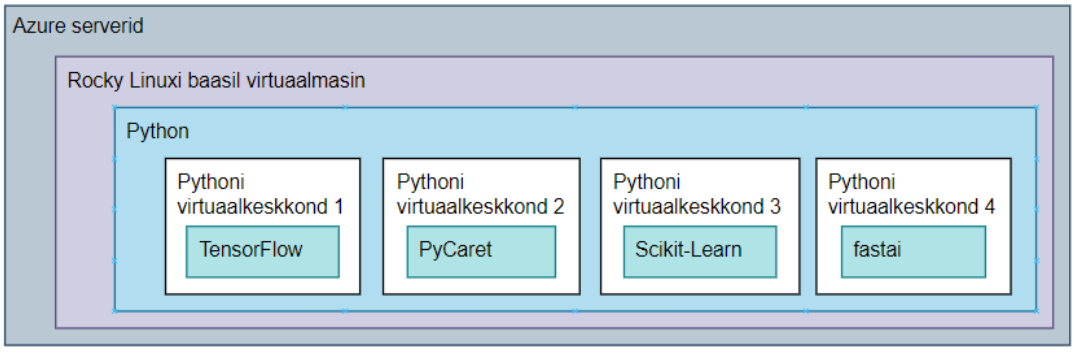
Pandas on Pythoni avatud lähtekoodiga andmeanalüüsi teek, mis võimaldab kasutada erinevaid keerukaid andmestruktuure. Teek sisaldab mitmeid meetodeid, mis lihtsustavad struktureeritud andmetega töötamist. Pandas loodi, et parandada aegridade, andmestruktuuride ja andmebaasidega seotud puudujääke, mis esinesid paljudes teistes andmeanalüüsi tööriistades [21]. Antud töös kasutatakse teeki juhul, kui väljavalitud masinõppe teek ei võimalda andmete importimist, töötlemist või esmast analüüsimist.

Matplotlib on Pythoni teek, mis võimaldab andmeid visualiseerida väga paljude erinevate diagrammitüüpidena. Antud töös kasutatakse Matplotlibi koos masinõppe teekidega, kui välja valitud teek pole eraldi loonud võimalusi andmete visualiseerimiseks [22].

2.3 Protsess

Töö jagati lõppeesmärgi saavutamiseks väiksemateks alamülesanneteks. Esmalt tuli tutvuda välja valitud Pythoni masinõppe teekide litsentside, teenusetingimuste ja privaatsuseeskirjadega, et veenduda kliendiandmete turvalisuses ning leida üles kindlustussektoris tegutsevale ettevõttele rakenduvad piirangud.

Sellele järgnes masinõppe keskkonna valmis seadmine (Joonis 1). Kogu arendus toimus masinõppe jaoks eraldatud Azure'i virtuaalmasinal, mis kasutas operatsioonisüsteemina Rocky Linuxit. Virtuaalmasin vastas mitmetele standarditele, sealhulgas ISO 27001 ja SOC 2 [23]. Virtuaalmasina siseselt implementeeriti kõik organisatsiooni reeglitest tulenevad turvanõuded. Arenduskeskkonnana kasutati veebipõhist Jupyter Notebooki.



Joonis 1. Masinõppe keskkonna arhitektuur.

Virtuaalmasina siseselt loodi iga teegi jaoks eraldi virtuaalkeskkonnad. Antud protsess oli oluline, kuna valitud teegid kasutasid sageli samasuguseid, kuid erinevate versioonidega abistavaid teeke ning versioonide erinevus võis põhjustada potentsiaalseid tõrkeid teekide töös. Peale teekide installeerimist vastavatesse virtuaalkeskkondadesse, blokeeriti kõik väljaminevad ühendused, et veenduda andmete turvalisuses.

Andmete eeltöötluse etapis tutvuti esmalt etteantud andmekogumikega, et mõista iga veeru tähendust. Andmete puhastamiseks kasutati võimaluse korral väljavalitud teegi vastavaid meetodeid. Sobivate meetodite puudumisel kasutati alternatiivsete teekidena Pandast ja Matplotlibi. Eeltöötlus sisaldas endas näiteks erinevate andmekogumite importimist ja nende ühendamist, puuduvate väärtustega tegelemist, klassifitseerimist ning normaliseerimist. Enne mudeli treenimist jagati andmed 80/20 osakaalu järgi osadeks ehk 80% tervest andmekogumist kasutati treenimiseks ning 20% treenitud mudeli testimiseks.

Mudelite treenimiseks kasutati nii võimendustüüpi algoritme kui ka süvaõppe meetodeid. Lisaks sellele treeniti mudeleid erinevate seadistustega, et jõuda võimalikult efektiivse tulemuseni. Mudeleid võrreldi erinevate näitajate alusel ning parima tulemusega mudeli näitajad salvestati, et neid hiljem võrrelda teiste teekide tulemustega. Leitud tulemuste kvaliteeti ja asjakohasust valideeriti koostöös kindlustusettevõttes töötava aktuaariga ning puuduste korral täiendati mudeleid.

Selleks, et kõiki kriteeriumeid, sealhulgas ka suhtelisi näitajaid adekvaatselt hinnata, tuli esmalt leida kõikide näitajate olulisus. Antud töö raames leiti, et kõige otstarbekam on kasutada Dr. Klaus D. Goepeli Exceli AHP malli, kuna see võimaldab leida nii arvuliste- kui ka suhteliste näitajate olulisused.

Esmalt pandi paika kriteeriumid, mille alusel teeke hinnatakse. Hindamisel võeti kasutusele üheksa kriteeriumit: dokumentatsiooni struktuur, dokumentatsiooni sisu, õppevõimalused teegi autorite poolt, õppevõimalused internetis, meetodite täielikkus, meetodite keerukus, masinõppe võimalused, tulemuste leidmise kiirus ning tulemuste kvaliteet. Valitud kriteeriumite sisu on detailsemalt kirjeldatud alapeatükis 3.1. Hindamisel ei võetud arvesse teegi litsentsi, kuna välja valiti vaid teegid, mida on lubatud kasutada ka ärilistel eesmärkidel. Samuti jäi AHP-ga hindamiselt välja andmete

turvalisus, kuna teeke testiti turvalises keskkonnas, kus kogu võrguliiklus sai toimuda vaid turvalises võrgus olevate seadmete vahel.

Peale kriteeriumite valimist tuli määrata kõikide kriteeriumite omavathelised suhted – tuli otsustada, kumb näitaja on olulisem ja anda hinnang skaalal 1–9. Antud skaalal märgib 1, et kriteeriumid on sama tähtsad ning 9, et olulisemaks valitud kriteerium on teisest igal võimalikul viisil tähtsam. Peale hinnangute andmist tõi tabel esile kõik sellised võrdlused, mis tuli vastuolu tõttu uuesti hinnata. Lõpuks, kui kõik oli tasakaalus, kuvas tabel kõikide kriteeriumite osakaalud. Samuti veenduti, et peale tasakaalus olemist oleks ka CR (*Consistency Ratio*) lubatud piirides. CR näitab, kas kriteeriumid on tõeselt hinnatud ning heaks peetakse olukorda, kus antud näitaja on väiksem kui 10% [24].

Iga teegi juures hinnati kõiki kriteeriumeid 10 palli skaalal. Teegi lõpliku hinde leidmiseks korrutati kriteeriumite kaalud ja vastavad hinnangud 10 palli skaalal ning liideti korrutised kokku. Leitud summad võimaldasid järjestada kõiki väljavalitud masinõppe teeke. Antud järjestuse põhjal tehti lõplikud järeldused ning anti soovitus optimaalse masinõppe teegi osas, mida käesoleval ajahetkel on kindlustussektoris töötaval ettevõttel mõistlik enda protsessides rakendada.

3 Teekide ülevaade

Antud peatükk annab ülevaate kõikidest töös kasutatavatest Pythoni masinõppe teekidest. Iga teegi kohta tuuakse välja millise litsentsiga on teek kaitstud, kas on viiteid andmete kogumise kohta, milline on teegi dokumentatsioon ning milliseid võimalusi on loodud õppimise lihtsustamiseks. Arendusspetsiifilisemalt tuuakse välja teegi täielikkus, keerukus, võimalused ning tulemuste leidmise kiirus ja kvaliteet samal andmestikul.

Esmalt selgitatakse punkte, mille alusel teeke analüüsitakse ning seejärel tuuakse välja kõikide välja valitud teekide ülevaated.

3.1 Analüüsitavad punktid

Järgnev alapeatükk toob välja peamised näitajad koos selgitustega, mille abil selgitatakse välja teek, mida on otstarbekas kindlustussektoris kasutada.

Litsents – Töös kasutatakse vaid avatud lähtekoodiga Pythoni teeke, mis on kaitstud lubavat tüüpi litsentsidega, peamiselt seetõttu, et nende kasutamisel on kasutajal rohkem õiguseid. Hoolimata sellest, et teegid on tasuta, tuleb täpsemalt uurida iga teegi litsentsi, et saada ülevaade, millised piirangud kehtivad ettevõtetele ning millistel tingimustel on lubatud teeki kasutada.

Andmete turvalisus – Kogu arendustegevus käib töö raames virtuaalmasinal, millel pärast kõikide analüüsiks vajalike komponentide lisamist puudub ligipääs internetile. Vaatamata sellele, peab töö autor oluliseks uurida, kas on viiteid selle kohta, et väljavalitud teeke kasutades saadetakse arendaja teadmata andmeid kolmandale osapoolele. Kindlustussektoris on kategooriliselt välistatud, et kliendiandmetele pääsevad ligi isikud, kellel pole õigust neid näha. Piirang kehtib olenemata sellest, kas andmeid kogutakse vaid analüüsimiseks, et teeki parandada või muul klienti kahjustaval viisil.

Dokumentatsioon – Dokumentatsiooni struktuuri punktis kontrollitakse selle olemasolu, üleüldise struktuuri loogilisust ning otsimise võimalust. Dokumentatsiooni sisu osas

analüüsitakse meetodite spetsiifilisemalt dokumentatsiooni mõistetavust – kas on olemas kirjeldus meetodite ülesannetest, millisel kujul on sisend- ja väljundparameetrid ning kas dokumentatsioon sisaldab koodi näidiseid meetodite või funktsioonide kasutamisest.

Õppimiskõver – Esmalt vaadatakse, kas teegi autorid on loonud õppimist toetavaid õppeprogramme ning millisel kujul need on – kas juhendid on interaktiivsed või on tegu tavalise tekstiga. Lisaks sellele vaadatakse milline on juhendite üleüldine struktuur – kas õpetamist alustatakse algusest, kui sujuvad on üleminekud ühelt teemalt teisele ning kas õppeprogrammid sisaldavad piisavalt näidiseid koos kirjeldustega. Interneti õppimisvõimaluste osas uuritakse, kui palju on internetis teegi õppimist toetavaid materjale nii raamatute, videote kui ka foorumipostituste näol. Teegi spetsiifilisi raamatuid otsitakse O'Reilly õppeplatvormilt, mis sisaldab kümneid tuhandeid IT sektorit puudutavaid raamatuid. Videote otsimiseks kasutatakse Google poolt loodud vastavat otsingut, mis koondab kokku veebis leitavad otsingusõnaga seonduvad videod. Foorumipostituste kohta uuritakse Stack Overflow'st, mis on peamiselt arendajatele suunatud platvorm IT teemaliste küsimuste küsimiseks.

Masinõppe kood – Meetodite täielikkuse ja keerukuse analüüsimise abil soovitakse hinnata masinõppe projekti üleüldist keerukust. Hindamiseks luuakse projekt, mille tulemusena valmib mudel, mis suudab ennustada sisendparameetrite abil kliendiriski, mida aktuaarid saavad kasutada hinnastamise mudeli täpsustamisel. Täielikkuse ja keerukuse analüüsimisel vaadatakse üsna sarnaseid punkte, kuid erinevatelt külgedelt.

Täielikkusega soovitakse hinnata, kas teek sisaldab terveks projektiks vajaminevaid võimalusi, või tuleb projektis kasutada abistavaid teeke. Uuritakse võimalusi andmete importimiseks, kuvamiseks, puhastamiseks, klassifitseerimiseks, tulemuste graafikul kuvamiseks ning mudelite treenimiseks, testimiseks ja võrdlemiseks.

Keerukus näitab seda, kui keeruline on kõiki projekti osasid luua olemasolevate meetoditega. Uuritakse kui palju teevad meetodid taustal arendaja eest analüüsi tööd ära, sealhulgas võrdlevad erinevaid algoritme ja soovitavad kõige otstarbekamat mudelit.

Masinõppe võimalused – Uuritakse, kas ja milliseid masinõppe algoritme teek sisaldab ja võimaldab kasutada. Antud osas uuritakse vaid hinnastamise teooria poolt soovitatud võimendustüüpi algoritme. Lisaks sellele rakendatakse võimaluse korral andmetele

süvaõppe algoritme, et testida, kas teegid on võimelised leidma peidetud seoseid andmete vahel.

Tulemused – Tulemuste hindamisel uuritakse nii tulemuste leidmise kiirust kui ka nende kvaliteeti. Kiiruse abil selgitatakse välja aeg, mis kulub mudeli treenimiseks ja testimiseks. Tulemuste kvaliteedi hindamisel analüüsitakse koos aktuaariga lõplikku mudelit, et veenduda selle õigsuses, välja valitud parameetrite loogilisuses ning seletatavuses. Tulemusi ja treenimiseks kulunud aega antud alapeatükkides eraldi ei kajastata, küll aga on antud näitajate võrdlus erinevate teekide suhtes välja toodud peatükis 4.2.

3.2 TensorFlow

Litsents – Teek on kaitstud Apache 2.0 litsentsiga, mille erinevusi teiste litsentsidega käsitletakse peatükis 4.2.

Andmete turvalisus – Puuduvad viited andmete kogumise kohta nii internetis leiduva info kui ka võrgulogide järgi.

Dokumentatsioon – Väga sisukas ja loogilise ülesehitusega dokumentatsioon, kus külje peal on vertikaalne navigeerimismenüü koos filtreerimise võimalusega. Kogu dokument on jagatud TensorFlow poolt pakutavate moodulite kaupa osadeks ja iga osa alt leiab kõik moodulit sisaldavad meetodid. Otsing toimib kiiresti ning peale iga uue tähemärgi sisestamist kuvatakse vaid need moodulid ja meetodid, mis sisaldavad otsitavat märksõna.

Iga meetodi juures on lühikirjeldus meetodi tööst, milliseid andmeid on vaja sisendina, mis tüüpi väljund tagastatakse ning mis tüüpi vead võivad esineda. Lisaks sellele on keerukamate ja rohkemate sisenditega meetodite juures näidised meetodi kasutamisest ning märgitud millises juhendis on seda kasutatud, mis võimaldab funktsiooni sisu veelgi paremini mõista. TensorFlow sisaldab ka mitmeid erinevaid eksperimentaalseid meetodeid ehk meetodeid, mis on veel testimisjärgus, kuid on kasutajatele juba kättesaadavaks tehtud. Lisaks sellele on neil meetodeid, mille tööpõhimõtteid on mõnest teisest teegist laenatud, kuid mille kasutamist on oluliselt lihtsustatud. Parema võrdlusmomendi tekitamiseks viitavad selliste meetodite juures teegi autorid asukohale algse teegi dokumentatsioonis.

Õppimiskõver – TensorFlow on loonud nii algajatele kui ka edasijõudnutele õppeprogramme, mida on võimalik läbida nii enda arvutis kui ka kasutades Google Colabi. Algajatele mõeldud õpetused alustavad masinõppe õpetamisega päris algusest ehk teekide importimisest, andmete importimisest, -töötlemisest ja -kuvamisest, ning juhendi lõpuks valmib esimene väiksem projekt. Kõik mudelid, mida juhendis kasutatakse, viitavad vastavale kohale dokumentatsioonis, et kasutaja saaks võimalikult lihtsalt infot juurde otsida. Lisaks sellele on eraldi välja toodud ka põhiinfo TensorFlow'st, mida loojate arvates kasutajad peaksid teadma. See sisaldab infot teegi erinevate muutujate, graafikute, funktsioonide, moodulite, kihtide ja mudelite treenimise kohta. Vaatamata rohketele algajatele mõeldud juhenditele, on kohati välja toodud koodi näidised puudulikud ning täpsustusi vajavad. Lisaks sellele on loodud eraldi leht, kus on kiirviited kõikidele traditsioonilise masinõppe projekti olulistele osadele. Samalt lehelt leiab TensorFlow poolsed soovitused erinevate teekide ja tööriistade osas, mida soovitatakse koos teegi endaga kasutada, et muuta masinõppe mudelid põhjalikumaks ning masinõppe projektide protsess efektiivsemaks.

TensorFlow on üsna aktiivne ka sotsiaalmeedias. Nad peavad blogi, kus räägitakse kõikidest uuendustest ja nende lisamisest olemasolevatesse projektidesse, uutest toodetest ning ettevõtetest, kes kasutavad TensorFlow'd. Lisaks sellele omavad nad YouTube kanalit, kuhu postitatakse vähemalt kord nädalas õpetus- või tutvustusvideo, mis võib kasulik olla nii algajale kui ka edasijõudnud masinõppe arendajale või huvilisele. TensorFlow kohta on internetis kümneid videoid, kus lahendatakse erinevaid elu- ja sõidukikindlustuse probleeme kasutades teegi võimekusi. Stack Overflow's on TensorFlow'ga seonduvaid küsimusi ligi 76 000 ning nendest pea veerand on ilma vastuseta. Sarnaselt videotele ja foorumipostitustele, on TensorFlow'd kajastatud paljudes raamatutes. Teeki on mainitud rohkem kui tuhandes O'Reillyse raamatus.

Masinõppe kood – Teegil on olemas meetod, mis võimaldab importida erinevate valdkondade andmekogumeid. Vabalt valitud andmete importimiseks pole ühtegi võimalust loodud, sellest tulenevalt kasutatakse andmete importimisel Pandase teeki.

TensorFlow sisaldab mõningaid universaalseid meetodeid andmete puhastamiseks, mille toimimist saab arendaja erinevate sisenditega vastavalt projektile seadistada. Puhastamise funktsionaalsusi on võimalik rakendada nii tekstilistel-, numbrilistel-, kategoorilistel- kui ka graafilistel andmetel. Lisaks sellele on teegi autorid loonud TensorFlow jaoks eraldi

TensorFlow Transform teegi, mis on spetsiaalselt mõeldud andmete eeltöötlemiseks. Lõputöö raames uuritakse vaid masinõppeks välja valitud teekide võimalusi ning vajaduse korral kasutatakse kõikide teekide jaoks samasuguseid abistavaid teeke, mistõttu Transform teek jääb antud töö skoobist välja ning andmete töötlemiseks kasutatakse täiendavalt veel Pandase teeki. Andmete graafikutel kuvamiseks pole samuti eraldi meetodeid loodud, mistõttu kasutatakse selleks Matplotlib teegi võimalusi.

Enne mudeli treenimist tuleb arendajal esmalt väärtustada treenimise ja testimise andmekogumid kasutades Pandase vastavat meetodit. Lisaks sellele peab arendaja käsitsi eemaldama nii treening- kui ka test andmekogumite korral tulba, mille tulemusi soovitakse leida. Treeningandmete graafikute ja statistika uurimiseks kasutatakse abistavaid Pandase ja Matplotlibi teeke. Mudelite treenimiseks ja -parandamiseks on loodud üsna lihtsad meetodid, mis vajavad toimimiseks minimaalseid sisendeid, kuid soovi korral saab antud meetodite tööd vägagi detailselt seadistada. Erinevate algoritmide abil loodavate mudelite võrdlemiseks ning parima valimiseks otseselt abistavaid meetodeid loodud ei ole. Samas üksikute mudelite treenimisel on võimalik määrata, et protsess jääks pooleli, kui mudel otseselt ei parane.

Sarnaselt eelnevaga, on ka süvaõppe rakendamiseks loodud üsna lihtsalt mõistetavad meetodid. Nende kasutamisel saab samuti piirduda vaid minimaalsete sisendandmetega ning olenevalt masinõppe projekti keerukusest on võimalik mõjutada kogu protsessi tööd väga spetsiifiliste sisenditega. Selle jaoks on loodud palju kihte, optimiseerimise võimalusi, eri tüüpi funktsioone ja palju muud. Antud võimaluste sobivust ülesande tüübiga otseselt ei kontrollita, mistõttu peab arendaja olema teadlik erinevatest sisenditest ja nende kasutamise kohtadest. Lisaks sellele tuleb ka süvaõppe rakendamise korral mudelite võrdlemiseks arendajal mitmete sisenditega erinevaid mudeleid luua ning analüüsida nende näitajaid, et teha lõplik valik, milliseid sisendeid ja millist mudelit protsessides kasutada.

Masinõppe võimalused – Teegi autorid on loonud võimendustüüpi algoritmi, mille põhimõtteid on laenatud traditsioonilisest GBM-ist. Antud algoritmi on märkimisväärselt täiendatud ning autorite sõnul on TensorFlow võimendamise algoritm kiirem kui GBM [25]. Teegist leiab erinevaid võimalusi, mille abil saab arendaja luua väga spetsiifilisi süvaõppe mudeleid vastavalt projekti tingimustele.

3.3 PyCaret

Litsents – Teek on kaitstud MIT litsentsiga, mida käsitletakse täpsemalt peatükis 4.2.

Andmete turvalisus – Puuduvad viited selle kohta, et teek saadab arendaja teadmata andmeid kolmandatele osapooltele.

Dokumentatsioon – Dokumentatsioon on jagatud teegi moodulite kaupa osadeks, mis on ligipääsetavad navigeerimismenüüst. Moodulis olevad meetodid pole jagatud eraldi lehtedele, vaid on lisatud üksteise järele koos meetodeid kirjeldavate alapeatükkidega. Dokumentatsioonist märksõnade otsimine võib olla tülikas, kuna sageli kuvatakse tulemustes alapeatükid ning versioonide uuendused läbisegi.

Kõiki teegi poolt pakutavaid võimalusi on kirjeldatud väga arusaadavalt ning võimaluse korral on illustreeritud ka piltidega. Meetodite juures on välja toodud mitmeid koodi näiteid, mis on jagatud loogilistesse osadesse ning eraldatud kommentaaridega. Kõik sisendparameetrid sisaldavad lühikirjeldust parameetri rollist vastavas meetodis, vaikimisi väärtusi, tüüpe ning kõiki väärtusi, mida on võimalik omistada. Osade meetodite kirjeldustes viidatakse teistele meetoditele, kuid, et neid lähemalt vaadata, tuleb arendajal dokumendist ise vastav koht üles otsida, kuna kiirviide nendeni puudub.

Õppimiskõver – PyCaret on loonud iga teegi mooduli kohta vastavalt arendaja tasemele mitmeid juhendeid, mis sisaldavad andmete importimist ja töötlemist, mudelite loomist, parandamist, võrdlemist, graafikute kuvamist, andmete analüüsimist ning tulemuste ennustamist. Kasutatavaid meetodeid on arendajatele üldjuhul arusaadavalt selgitatud ning osaliselt viidatakse juhendite siseselt ka teegi dokumentatsioonile. Paljud algoritmid ning statistika spetsiifilised lähenemised, mida juhendites on kasutatud, viitavad erinevatele välistele allikatele, kus teemasid on detailsemalt kirjeldatud.

Esmastes juhendites rakendatud meetodid kasutavad üldjuhul vaid minimaalseid sisendandmeid, mida meetodi toimimiseks on vaja. Iga järgneva juhendiga tutvustatakse aina rohkem võimalusi, tänu millele võivad muutuda mudelid veelgi täpsemaks ning paljudel juhtudel tulemuste leidmine efektiivsemaks. Mõned juhendid viitavad mooduli edasi õppimiseks järgmise keerukusega juhenditele, mis paraku paljudel juhtudel pole valminud. Arvestades asjaolu, et selliste juhendite viimased uuendused on olnud aasta 2020 lõpus, siis pole kindel, kas lähiajal keerukamad ja lubatud detailsemad juhendid

kasutajateni jõuavad. Lisaks sellele tuletatakse suurema raskusastme juhendites meelde pea kõiki algteadmisi teegi kohta. Selle asemel oleks oluliselt otstarbekam tutvustada juhendis kas erinevaid meetodeid ning nende võimalikke parameetreid või üldse jätta keerukamad juhendid lühemad ja konkreetsemad.

PyCareti kohta on internetis mitmeid videoid, kus tutvustatakse, millistel viisidel on võimalik teeki kasutada kindlustussektoris. Lisaks sellele omavad nad YouTube kanalit, kuhu postitatakse videoid nii teegi kasutamisest koos teiste platvormidega kui ka teegi uuendusi ning võimalusi. Samasid teemasid, mis videotes, käsitletakse ka teegi ametlikus blogis. Stack Overflow's on veidi üle 100 PyCareti teemalise küsimuse, millest umbes pooled pole saanud mitte ühtegi kommentaari teiste foorumi kasutajate käest. O'Reillyst leiab vaid 7 raamatut, kus on vähemalt korra mainitud sõna PyCaret, kuid mis kõik ei kasuta oma näidetes PyCareti võimalusi.

Masinõppe kood – Teegil on meetod andmete importimiseks, kuid see on eelkõige mõeldud ainult PyCareti tundma õppimiseks, kuna seda saab kasutada vaid andmekogumike importimiseks, mis on teegi autorite poolt kättesaadavaks tehtud. Vabalt valitud andmete importimise võimalusi otseselt loodud ei ole, mistõttu töös kasutatakse selleks Pandase teegi võimalusi.

Andmete puhastamiseks on loodud küllaltki universaalne meetod, mida kasutajal on võimalik väga täpselt oma soovide järgi seadistada. Antud üherealine meetod tegeleb näiteks puuduvate väärtustega, andmete normaliseerimise ja -klassifitseerimisega. Meetod rakendab andmetele erinevaid algoritme, mis aitavad selgeks teha, kas andmed on vaja jagada erinevatesse kategooriatesse või mitte. Kui on vaja, siis tehakse see taustal automaatselt ära. Peale analüüsimist kuvatakse kasutajale tabel, kus on kirjas tulba pealkiri ja selle tüüp. Kasutajal tuleb kinnitada, et automaatselt tehtud järeldused on õiged. Vigade korral on võimalik meetodi lõpuleviimine katkestada ning teha käsitsi muudatused, mille osas meetod eksis. Meetodi töö lõppedes kuvatakse kasutajale tabel kõikide esmaste analüüsi ja seadistuste tulemustest, mis võimaldab taaskord veenduda, et kõik sai õigesti seadistatud.

Mudelite treenimiseks ja sobivaima valimiseks kasutatakse ühte ühist meetodit. Kui kasutaja ei määra teisiti, kasutab teek mudeli treenimisel vaikimisi kõiki algoritme. Olukorras, kus on teada, millised algoritmid on andmekogumiku korral tõenäoliselt kõige

otstarbekamad, siis ressursi kokkuhoiu mõttes saab arendaja mudeleid treenida vaid valitud algoritmidega. Mudeleid võrreldakse automaatselt erinevate näitajate järgi ning kasutajale kuvatakse parim algoritm ja mudel kindla näitaja korral. Võrdlusel kasutatavaid põhimõtteid on võimalik muuta meetodi sisendandmete muutmisega.

Mudelite testimise jaoks jagatakse juba andmete puhastamise meetodis kogu andmekogumik vaikimisi 70/30 jaotuse põhjal osadeks, kuid arendajal on võimalik seda soovi korral muuta.

Tulemuste kuvamiseks on loodud üherealisi meetodeid. Ühe meetodiga kuvatakse kasutaja poolt soovitud graafik, teise meetodiga käivitatakse kasutajaliides, kus on võimalik nuppudele vajutamisega vahetada graafikuid ja võrrelda nende sisu.

Masinõppe võimalused – Teegi autorid pole ise uusi algoritme loodud, vaid on koondanud kokku tuntuimad algoritmid, mida on võimalik eri tüüpi ülesannete lahendamiseks kasutada. Teek võimaldab kasutada mitmeid erinevaid võimendustüüpi algoritme, sealhulgas ka XGBoosti ja GBMi. Süvaõppe võimalusi teegil hetkel ei ole, kuid autorid on vihjanud, et tulevikus võib see muutuda [18].

3.4 Scikit-Learn

Litsents – Teek on kaitstud BSD – 3 litsentsiga, mida käsitletakse täpsemalt peatükis 4.2.

Andmete turvalisus – Internetis leiduva info ja võrgulogide põhjal pole alust arvata, et teek koguks andmeid.

Dokumentatsioon – Kodulehel olev dokumentatsioon ei ole eriti kasutajasõbralik – puudub kiirmenüü ning see teeb peatükkide vahel navigeerimise tülikaks. Otsingu tulemusi ei kuvata peale iga tähemärgi sisestamist ning tulemused pole jagatud moodulite kaupa kategooriatesse, vaid on järjestatud peatüki nime järgi tähestikulisse järjekorda.

Kõik meetodid mida teek võimaldab kasutada, sisaldavad detailset infot otstarbe, sisendparameetrite ja väljundi kohta. Algoritmide kohta, mida teek võimaldab kasutada, on loodud erinevaid näidiskoode, kuhu on lisatud algoritmi kirjeldus ja ligikaudne tööaeg. Lisaks sellele on dokumentatsioonis ära märgitud kõik uuendustega lisandunud võimalused ja tehtud muudatused. Kui meetodi kasutamine eeldab, et andmetega on

eelnevalt läbi tehtud mingi protsess, siis kuvatakse meetodi lõpus meeldetuletav kiri ning viide antud protsessile.

Õppimiskõver – Scikit-Learn on loonud juhendi, mis võtab kokku kõik põhilised võimalused, mida antud teegiga on võimalik teha. Teegi algajatele mõeldud juhendid eeldavad minimaalseid teadmisi masinõppe kohta. Õpetatakse projekti jaoks probleemi püstitamist, sobiva masinõppe viisi valimist, andmete importimist, töötlemist, kuvamist, mudeli treenimist, tulemuste ennustamist ning erinevate algoritmide rakendamist andmetele. Osad kasutatavad meetodid ja spetsiifilisemad sõnad viitavad kas vastavale kohale dokumentatsioonis või täpsustavale artiklile internetis. Meetodid, kus tuleb sisendparameetreid valida sõltuvalt projekti tüübist ning andmekogumiku sisust, viitavad abistavatele juhenditele, kus õpetatakse sobivaima sisendi valimist.

Juhendid on piisavalt informatiivsed ning annavad lihtsas ja arusaadavas keeles ülevaate sellest, miks mingid teegi poolt pakutavad võimalused on vajalikud. Lisaks sellele on kodulehel välja toodud palju erinevaid videoid, mis aitavad alustada Scikit-Learniga, millest paraku enamus on üle 10 aasta vanad ega sisalda uuemaid teegi võimalusi.

Internetis leidub mitmeid videoid, kus teeki kasutatakse erinevate kindlustustoodete hinnastamiseks. Scikit-Learni teemalisi foorumipostitusi on üle 25 000, millest vähem kui viiendik on endiselt ilma vastuseta. Raamatuid, kus teeki on vähemalt ühe korra käsitletud, leidub O'Reillys üle 900.

Masinõppe kood – Teegis on meetod, mis võimaldab importida erinevat tüüpi andmekogumeid, mida dokumentatsioonis ja juhendites kasutatakse Scikit-Learni tutvustamiseks. Täiendavalt on loodud meetodid erinevat tüüpi andmekogumite genereerimiseks. Lokaalsete failide importimiseks on samuti olemas meetod, kuid kuna teek kuvab andmeid massiivi kujul, mida inimesel võib olla ebamugav lugeda, siis parema loetavuse ning töö efektiivsuse parandamise põhjustel kasutatakse nii importimiseks kui ka andmete kuvamiseks Pandast.

Andmete puhastamiseks on loodud mitmeid erinevaid meetodeid, mida arendaja peab ise andmekogumiku peal rakendama. Meetodite rakendamise vajadus sõltub andmetest, kuid seda, millal mingeid meetodeid kasutada tuleb, peab arendaja ise teadma. Andmete õigete tüüpide määramiseks on loodud meetod, mis vajab minimaalse sisendina vaid andmekogumit. Meetodil on täiendavalt veel mitmeid parameetreid, mis võimaldavad

rohkem kontrollida protsessi käiku. Kuigi suur osa tööst tehakse automaatselt ära, peab arendaja veenduma antud muudatuste õigsuses ning vajadusel tegema parandusi.

Teek võimaldab visualiseerida vaid mudeli treenimisega seonduvaid näitajaid. Imporditud andmete graafikutel kuvamiseks võimalused puuduvad, mistõttu kasutatakse abistava teegina Matplotlibi.

Enne andmete treenimist tuleb arendajal käsitsi jagada kogu andmekogum test- ja treeningandmeteks, sealjuures tuleb mõlemast andmekogumist eemaldada tulp, mille väärtusi soovitakse ennustada. Mudelite treenimiseks ja testimiseks on loodud üsna kergesti mõistetavad ja minimaalset tööd vajavad meetodid. Soovi korral on arendajal võimalik muuta seadistusi, mille alusel meetodid toimivad. Samuti leidub teegis mitmeid mudelite parandamiseks loodud meetodeid, mida on võimalik igas olukorras rakendada, kuid mis võivad teha mudelid märkimisväärselt halvemaks, kui arendaja ei rakenda neid õiges olukorras. Antud meetodid võivad arendaja soovi korral lõpetada treenimisprotsessi, kui mudel määratud aja jooksul ei parane piisavalt palju.

Tulemuste võrdlemine tuleb väga suures osas ära teha arendaja poolt, kes peab iga mudeli jaoks leidma mudeli täpsust iseloomustavad näitajad. Antud näitajate leidmist lihtsustavad erinevad meetodid, kuid tulemuste võrdlus ning nende põhjal sobivaima mudeli valik tuleb arendajal ise teha.

Masinõppe võimalused – Teek võimaldab kasutada kümneid juhitud ja juhtimata õppe algoritme, sealhulgas ka mitmeid võimendustüüpi algoritme. Nende seas on algoritme, mida teegi autorid on ise loonud, kui ka neid, mis on kellegi teise poolt loodud, kuid Scikit-Learni poolt täiendatud. Teegiga on võimalik luua eraldi närvivõrgu mudeleid, kuid dokumentatsioonis on välja toodud, et seda võimalust pole soovitatud suuremate projektide korral rakendada, kuna teek ei toeta GPU (*Graphics Processing Unit*) kasutamist, mis süvaõppe korral on oluliselt efektiivsem kui CPU (*Central Processing Unit*) kasutamine [26].

3.5 Fastai

Litsents – Teek on kaitstud Apache 2.0 litsentsiga, mida käsitletakse peatükis 4.2.

Andmete turvalisus – Ei leidu infot selle kohta, et teek saadaks omavoliliselt andmeid kolmandatele osapooltele.

Dokumentatsioon – Dokumentatsioon on jagatud moodulite kaupa osadeks. Menüüs on välja toodud kõik moodulid ning nende alamkategooriad, kuid meetodite nägemiseks tuleb kategooriad eraldi avada. Keeruline on vahet teha, kus algab ja lõpeb funktsionaalsuste kirjeldus. Meetodite parameetrid on kirja pandud lühenditega ning ilma täpsustavate kommentaarideta, mis raskendavad mõistmist. Paljudel juhtudel puuduvad dokumentatsioonis meetodi kasutamise näited. Dokumentatsioonis sobiva mõiste otsimine on ebamugav, kuna otsingu tulemustes kuvatakse pooltel juhtudel esmalt Google reklaame ja alles peale seda kohad dokumentatsioonis, kus leidub otsingusõna. Kogu dokumentatsiooni on autorid loonud ka Google Colabi, kus arendajad saavad proovida erinevate meetodite tööd enda poolt valitud andmetega.

Õppimiskõver – Autorid on loonud lühikese juhendi teegi peamistest võimalustest ja moodulitest. Lisaks sellele on võimalik kodulehel õppida teegi kasutamist pea 20 juhendi abil, mis on jagatud nii raskusastmete kui ka moodulite kaupa eraldi kategooriatesse. Õpetusi on võimalik läbida ka Google Colabis. Juhendid on üsna kergesti mõistetavad, kood on jagatud väiksemateks osadeks ning iga osa funktsionaalsus on tehtud õppijale arusaadavalt selgeks. Paraku on mitmed juhendid uuendamata nii dokumentatsioonis kui ka Google Colabis, millest tulenevalt kui neid värskeima ning soovitusliku teegi versiooniga kasutada, tekib nii moodulite importimise kui ka meetodite kasutamisega probleeme. Lisaks sellele kasutatakse osades juhendites abistavaid teeke, mille importimist pole kuvatud ning arendaja peab mõistma, millist teeki kasutati. Vaatamata sellele, et peamiselt kasutati vaid Pandast ja Matplotlibi, mille importimisel kasutatakse üldjuhul igas masinõppe projektis samasuguseid lühendeid, võib masinõppe valdkonnaga alles tutvuval arendajal tekkida alguses raskuseid selle mõistmisega.

Õpetuste seast leiab mitmeid fastaile migreerimise juhendeid, mis aitavad suurema vaevata võtta mõne teise teegi asemel kasutusele fastai ilma, et tuleks koodis väga suuri muudatusi teha. Kõik meetodid, mida õpetuses käsitleti esimest korda, viitasid vastavale kohale dokumentatsioonis.

Fastai kohta leidub vaid üksikuid kindlustussektorit puudutavaid videoid. Üleüldiseid videoid teegi kohta on rohkem ning lisaks sellele kajastab üks teegi loojatest enda

YouTube kanalil väga detailselt teegi võimalusi. Raamatuid, kus teeki käsitletakse, on O'Reilly platvormil 73. Stack Overflow's on fastai kohta 381 küsimust, millest pea kolmandik on vastuseta.

Masinõppe kood – Teek sisaldab meetodit, millega on võimalik importida kümneid veebis olevaid andmekogumeid, mis on mõeldud erinevat tüüpi masinõppe projektide jaoks. Lokaalsete andmekogumite importimiseks pole ühtegi meetodit loodud, mistõttu kasutatakse importimiseks Pandase meetodeid. Tabeli kujul andmete graafikul kuvamiseks võimalused puuduvad, mistõttu kasutatakse selle jaoks Matplotlibi meetodeid.

Teegi dokumentatsioonis ja juhendites kasutatakse üsna sageli Pandase võimekusi. Kuna fastai ja Pandase andmeraamistikud on veidi erinevad, siis arendajate töö lihtsustamiseks on loodud abistav meetod, mis konverteerib Pandase tabeli fastai tabeliks. Sama meetodiga on võimalik arendajal ära teha ka kogu andmete eeltöötlus. Meetod sisaldab võimalusi täpsustada tulpasid kategooriliste ja numbriliste andmetega ning määrata millise tulba väärtusi soovitakse hiljem leida. Samuti on antud meetodiga võimalik määrata seadistused puuduvate väärtuste haldamise, andmete normaalkujule viimise ning test- ja treeningandmete osakaalu osas. Ilma eelpool välja toodud üherealise andmete eeltöötluse meetodita peaks arendaja kirjutama kümneid ridu koodi ning võrdlema erinevaid andmeid, et jõuda samasuguse tulemuseni.

Peale andmete töötlust on võimalik rakendada süvaõppe meetodit, mille toimimiseks piisab vaid puhastatud tabeli kujul andmetest. Kuna iga projekt ja kasutatavad andmed on erinevad, saab arendaja mõjutada meetodi toimimist mitmete erinevate sisenditega. Teek võimaldab lisada süvaõppele omaseid kihte, määrata näitaja mille alusel valitakse parim mudel, otsustada milliseid funktsioone tuleb mudeli treenimise ajal käivitada ning millistel tingimustel tuleb treenimine pooleli jätta. Lisaks sellele on teegis olemas palju võimalusi, millega on võimalik masinõppe protsessi veelgi detailsemalt seadistada. Loodud mudelite kohta on võimalik leida väga palju mudelit iseloomustavaid näitajaid, kuid teegis pole otseselt meetodeid, mis võtaksid arvesse antud näitajaid ning otsustaksid, milline mudel on kõige täpsem. Vaatamata sellele, et teek ei võimalda tabeli kujul andmeid graafikul kuvada, on loodud meetodid, mis võimaldavad vaadata andmete treenimisega seonduvaid graafikuid.

Masinõppe võimalused – Tegu on süvaõppe teegiga ning ei sisalda otseselt võimendustüüpi algoritme, kuid teegi autorid on loonud meetodi ja juhendi, mille abil on võimalik muuta fastai spetsiifilise struktuuriga tabel tagasi Pandase struktuuriga tabeliks. Antud tabel on omakorda sobiv sisend teiste teekide võimendustüüpi algoritmidele. Süvaõppe võimaluste poolest on tegu sisuka teegiga – süvaõpet on võimalik rakendada nii tekstiliste, tabulaarsete kui ka graafiliste andmete jaoks.

4 Analüüs ja tulemused

Antud peatükk on jagatud kolme osasse. Esmalt tuuakse välja AHP kriteeriumite järjestus koos positsiooni täpsustava kommentaariga. Seejärel võrreldakse omavahel teeke ning tuuakse välja peamised tähelepanekud. Lõpetuseks võetakse kõik tulemused kokku ning tehakse lõplikud järeldused teekide osas, mida oleks optimaalne kindlustussektoris kasutada.

4.1 AHP kriteeriumite tulemused

AHP tulemusena selgus, et tähtsamad on näitajad, mis on olulised pigem pikas plaanis (Joonis 2). Näitajate võrdlemise tulemusena saavutati CR-iks 5,6%, mis jääb antud näitaja lubatavuse piiridesse. Kõige olulisemaks osutus tulemuste kvaliteet, mis annab üle veerandi punktidest, mis on äärmiselt oluline, kuna just tulemused on need, mille põhjal on võimalik olemasolevaid mudeleid parandada.

	Criterion	Comment	Weights
1	Crit1	Meetodite täielikkus	9,2%
2	Crit2	Meetodite keerukus	5,6%
3	Crit3	Masinõppe võimalused	22,4%
4	Crit4	Tulemuste leidmise kiirus	10,7%
5	Crit5	Dokumentatsiooni struktuur	2,1%
6	Crit6	Dokumentatsiooni sisu	14,7%
7	Crit7	Autorite Õppimisvõimalused	2,8%
8	Crit8	Interneti Õppimisvõimaluse	5,4%
9	Crit9	Tulemuste kvaliteet	27,1%

Joonis 2. Kuvatõmmis AHP tulemustest.

Kvaliteedile järgnes erinevate algoritmide olemasolu, mis andis kogupunktidest veidi vähem kui veerandi. Spetsiifilised võimendustüüpi algoritmid on just need, mis mitmete allikate järgi aitavad praeguse tehnoloogilise arenguga jõuda kõige paremate tulemusteni hinnastamisel, mistõttu näitaja suur osakaal lõplikul hindamisel on igati mõistetav. Viimasest väiksema tähtsusega on dokumentatsiooni sisu, mida arendaja kasutab suure tõenäosusega pea igas projekti faasis, mistõttu on oluline, et ametlikus dokumentatsioonis on piisavalt sisukad ja hästi illustreeritud näited teegi võimaluste osas.

Tulemuste leidmise kiirus on projekti juures olulise tähtsusega, kuid kiiremini leitud vastusest ei ole otseselt kasu, kui kiirema vastuse saamiseks kasutatakse ebasobivaid meetodeid, mis viivad valede või tõlgendamatute tulemusteni. Kiiruse ja kvaliteedi vahel tuleb leida optimaalne seos ning pikas perspektiivis on otstarbekam, kui see on rohkem kvaliteedi poole kallutatud.

Kõikide vajalikke meetodite olemasolu teegis on oluline, et arendaja saaks kindel olla teegi kõikide funktsionaalsuste toimimises. Samas, näitaja pole eriti kriitilise tähtsusega olukorras, kus projekti lisatakse juurde mõni sagedasti kasutatav abistav teek, mille sobivust algse teegiga on võimalik internetist kontrollida.

Sellele järgnesid kriteeriumid, mis kõik on üldjuhul olulised lühiperioodil, kui arendaja pole kursis teegi ja selle võimalustega. Meetodite keerukus sai hindamisel üsna madala osatähtsuse, seda peamiselt seetõttu, et kui meetodi kasutamine ja selle sisu selgeks teha, siis üldjuhul ei teki meetodi kasutamisega hiljem probleeme. Keerukusele järgnesid õppimisvõimalused nii internetis kui ka autorite poolt, mille madala osatähtsuse põhjus sarnaneb suuresti meetodite keerukuse põhjusele. Dokumendi sisust on õppevõimalused vähemtähtsamad peamiselt seetõttu, et sisuka dokumentatsiooni olemasolu korral vaatab arendaja sageli ametlikku dokumentatsiooni ja sealt vastuse mitte leidmise korral kasutab alternatiivseid allikaid. Kõige väiksem osatähtsus on dokumentatsiooni struktuuril, mis taaskord on oluline teegiga tutvumisel, kuid mis pikas perspektiivis omab väiksemat rolli.

4.2 Teekide näitajate võrdlemine

Järgnevalt tuuakse välja peamised tähelepanekud kõikide punktide osas, mida käsitleti peatükis 3.1.

Litsents – Tuntuimad lubavad litsentsid on MIT, BSD ja Apache, mis kõik on antud töös esindatud. Antud litsentsid on üsna suurel määral oma põhimõtete osas sarnased. MIT litsents lubab kasutajal lähtekoodiga teha peaaegu kõike, sealhulgas muuta, täiendada, levitada ja müüa eeldusel, et säilitatakse algne autoriõiguste teatis. BSD ja MIT peamiseks erinevuseks on see, et BSD litsentsiga koodi muutes ja levitades ei tohi ilma kirjaliku nõusolekuta viidata esialgse projekti osapooltele. Apache litsentsis on kõik põhilised lubavate litsentside põhimõtted, kuid lisaks sellele peab kasutaja teavitama autoreid, kui ta on lähtekoodile teinud suuremaid muudatusi [27].

Arvestades asjaolu, et antud töö raames kasutati kõiki teeke sellistena nagu nad on, ühtegi muudatust lähtekoodi ei tehtud, hinnastamine toimub vaid ettevõtte siseselt ning vabavara litsentsiga kaitstud koodi ei kasutata projektides, mida otseselt kliendile müüakse, siis antud töös võib nendel tingimustel lugeda analüüsimiseks välja valitud teegid võrdseteks.

Andmete turvalisus – Kõikide teekide osas uuriti nii kodulehtedelt kui ka üleüldiselt internetist viiteid andmete kogumise kohta. Mitte ühegi teegi kohta sellealaseid vihjeid internetist ei leidunud. Andmete mitte kogumist kinnitavad ka võrgulogid, mille järgi väljaminev liikluse maht oli olematu suurusega võrreldes projektides kasutatavate sisendandemete mahuga.

Dokumentatsioon – Kõikidel töös analüüsitud masinõppe teekidel oli loodud dokumentatsioon, mis sisaldas ülevaadet pakutavatest võimalustest. Struktuuri osas eristusid märkimisväärselt TensorFlow ja PyCareti dokumendid. Need on suhteliselt lihtsa ja mugava ülesehitusega ning kasutajate jaoks on lihtsustatud ka erinevate osade vahel liikumine. Teiste teekide dokumentatsioonidest on võimalik leida kõik vajalikud kohad üles, kuid see võib võtta veidi rohkem aega. Peamisteks põhjusteks on see, et dokumendis liikumine on kohati raskendatud ning otsinguvõimaluse kasutamisel pole tulemused mõistlikult struktureeritud. Positiivses mõttes eristus teistest fastai, mille dokumentatsioonis olevad koodi osad olid kõik lisatud ka Google Colabi, mis lihtsustab uute koodiosade testimist.

Hoolimata sellest, et nii TensorFlow kui ka Scikit-Learni pakuvad teistest teekidest oluliselt rohkem funktsionaalsuseid, on nad suutnud oma dokumentatsiooni teistest sisukamana hoida. Detailselt on kirjeldatud nii erinevate meetodite sisu kui ka parameetreid. Lisaks sellele kuvavad antud teegid üsna sageli viiteid nii dokumentatsioonisisestele kui ka -välistele allikatele, mis aitavad funktsionaalsuseid paremini mõista. Veidi väiksema, kuid siiski piisava detailsusega on PyCareti dokumentatsioon. Kõik siiani välja toodud teegid on loonud piisavalt näidiseid, et muuta funktsionaalsuste mõistmine kasutaja jaoks võimalikult lihtsaks. Fastai dokumentatsioon on sisu poolest teistest oluliselt nõrgem. Paljudel juhtudel on meetodite ja ka parameetrite kirjeldused pealiskaudsed ning lugejale pole kuvatud, kuidas meetodeid peab kasutama.

Õppimiskõver – Iga teegi õpetamiseks on lähenetud suhteliselt sarnaselt. Loodud on lühikesed ülevaated peamistest võimalustest ning mitmed sisukamad juhendid erinevate

moodulite kohta. Juhendite struktuur on samuti suhteliselt sarnane. Kõikidel juhtudel õpetatakse uusi teemasid lühikese projekti abil, mis põhineb mõnel veebis kättesaadaval andmekogumikul. Juhendi lõpuks valmib projekt, milles leitud mudeliga on võimalik ennustada tulemusi. Teekide õpetused on üldiselt kergelt mõistetavad ning annavad hea ülevaate, kuidas teeki enda projektides rakendada. Uute terminite või meetodite mainimisel juhendis, viitavad kõik sageli ka vastavale kohale dokumentatsioonis või internetis, et kasutaja saaks soovi korral lihtsalt ja kiirelt täpsustavat informatsiooni.

Nii TensorFlow kui ka fastai juhendid on kättesaadavad ka Google Colabis, kus kogu kood on ette kirjutatud ning kasutaja saab valida, kas käivitab koodi osade kaupa või muudab sisendeid vastavalt soovile.

Paraku on iga analüüsitava teegi õpetuste juures midagi probleemset. TensorFlow kasutab kohati pikki koodi osasid ilma täpsustuseta, PyCaret on lubanud aastaid juhendeid, mida pole endiselt loodud, Scikit-Learn soovitab õppimiseks videoid, mis on suhteliselt vanad ja kasutavad aegunud lähenemisi ning mõned fastai juhendid on uuendamata, mistõttu pole kõiki juhendeid võimalik täielikult järgida.

Õppimist soosivate võimaluste hulk internetis on teekide lõikes väga erinev. TensorFlow ja Scikit-Learn on teistest nii raamatute, videote kui ka foorumipostituste arvu poolest ülekaalukalt ees. Samasugune seaduspärasus kehtib ka veebis leitavate materjalide kohta, kus teeki kasutatakse erinevates kindlustussektori projektides. Teegi paremaks reklaamimiseks ja uute lähenemiste tutvustamiseks on nii TensorFlow kui ka PyCaret loonud YouTube kanali ja blogi, kuhu postitatakse aktiivselt uut sisu. Scikit-Learn ja fastai kasutavad samuti antud platvorme, kuid postitusi tehakse sinna väga harva.

Masinõppe kood – Meetodite täielikkust uurides joonistub välja sarnane muster – teekide autorid on pigem keskendunud erinevate masinõppe võimaluste pakkumisele. Lihtsamate toimingute jaoks, nagu näiteks andmete importimine ja -visualiseerimine, pole loodud eraldi meetodeid ning pigem soovitatakse kasutada abistavaid teeke, mis on spetsiaalselt selleks otstarbeks loodud. Teiste andmepuhastuse toimingute jaoks on igas teegis olemas mitmeid meetodeid. Samuti on kõigil olemas vajalikud meetodid mudelite treenimiseks ning testimiseks. Paraku puudus kõikidel teekidel peale PyCareti võimalus visualiseerida loodava mudeliga seotud andmeid.

Teegi keerukuse osas eristuvad enim PyCaret ja fastai, mille autorid on suutnud luua teegi, järgides enda eesmärki muuta masinõppe mudelite loomine kasutaja jaoks võimalikult lihtsaks. Mõlema teegi korral toimub andmete eeltöötlus suurel määral taustal, kuid arendajale on jäetud võimalus muuta selle protsessi toimimist. Nii TensorFlow'd kui ka Scikit-Learni kasutades peab arendaja omama rohkem teadmisi andmepuhastuse ning statistika osas.

Mudelite treenimisel ja testimisel on teegid üsna võrdsed. Teegid, mis võimaldavad kasutada olemasolevaid algoritme, vajavad mõlemaks toiminguks vaid ühte üsna lihtsalt kasutatavat meetodit. Hoolimata sellest, et meetodid on tehtud väga lihtsaks, saab keerukamate projektide korral muuta meetodite kasutamise suure hulga sisendparameetritega vägagi detailseks. Süvaõppeks mõeldud teegid on treenimisel ja testimisel oluliselt keerukamad, nõuavad rohkem arendaja poolseid seadistusi ning valdkonnaspetsiifilisi teadmisi. Kõikide teekide treenimise meetodid võimaldavad jätta treenimise pooleli olukorras, kus mudeli täpsus ei parane piisavalt palju erinevate treeningtsüklite jooksul.

Mudelite analüüsimises on taaskord PyCaret teistest oluliselt võimekam. Arendaja peab vaid teadma, mis näitajate alusel soovitakse võrdlusi näha. Teisi teeke kasutades tuleb analüüsimiseks luua eraldi mudelid erinevate algoritmide ja sisendandmete korral, leida mudelite statistilised näitajad ning võrrelda leitud näitajaid.

Masinõppe võimalused – Teeke analüüsides selgus, et kõik teegid võimaldavad kasutada mudelite loomiseks kas võimendustüüpi algoritme või süvaõpet, millest järeldub, et kõik teegid omavad piisavalt funktsionaalsusi, mida saab kasutada eri tüüpi kindlustussektori ülesannete jaoks. Kõige universaalsemaks teegiks võib pidada TensorFlow'd, kuna see võimaldab kasutada nii süvaõpet kui ka võimendustüüpi algoritme. Välja valitud teekidest võimaldavad PyCaret ja Scikit-Learn kasutada teistest rohkem erinevaid võimendamise algoritme. Antud teegid ei võimalda küll töö kirjutamise hetkel süvaõpet rakendada, kuid on mingil määral alustanud selle võimaluse teeki lisamisega. Fastaid on antud hetkel võimalik kasutada vaid projektides, kus soovitakse rakendada süvaõpet.

Tulemused – Mitmete mudeli treenimiste ja parandamiste tulemusena selgus, et töös kasutatud andmekogumiku korral on mudeli treenimine ajaliselt kõige efektiivsem

kasutades PyCaretit. Sellest pea veerandi võrra aeglasemad olid fastai ja TensorFlow süvaõppe võimalus. Teistest märkimisväärselt aeglasemad, kuid ligilähedase treenimise ajaga olid Scikit-Learn ning TensorFlow võimendustüüpi algoritme kasutada võimaldav teek.

Mudelite parameetrid ning nende kaalud olid üldjuhul loogilised ja sarnased, kuid vaatamata sellele tekitasid mõned tulemused kahtlusi, mistõttu vajavad need aktuaari poolt edasist analüüsimist. Varasemalt mitte nähtud andmekogumiku tulemusi suutis kõige täpsemalt ennustada võimendustüüpi TensorFlow mudel. Sarnase täpsusega oli ka Scikit-Learni abil loodud mudel. Viimasest ligi 7% suuremate vigadega olid PyCareti loodud mudeli ennustused. Kõige ebatäpsemad olid mõlemad süvaõppe meetodeid kasutanud mudelid.

4.3 Tulemuste kokkuvõte

Iga teegi juures hinnati üheksat kriteeriumit 10 palli skaalal (Tabel 1). Antud hinnangud korrutati läbi vastava kriteeriumi osatähtsusega ning korrutised liideti kokku. Leitud summa on teegile antav hinne, mille põhjal otsustatakse, millist teeki on kõige optimaalsem kasutada kindlustussektoris.

Tabel 1. Teekide kriteeriumite hinded 10 palli skaalal.

Tähis	**Selgitus**	**TensorFlow**	**PyCaret**	**Scikit-Learn**	**fastai**
Crit1	Meetodite täielikkus	8	9	8	8
Crit2	Meetodite keerukus	7	10	7	8
Crit3	Masinõppe võimalused	10	9	9	8
Crit4	Tulemuste leidmise kiirus	7	10	4	7
Crit5	Dokumentatsiooni struktuur	10	8	5	6
Crit6	Dokumentatsiooni sisu	10	9	10	5
Crit7	Autorite Õppimisvõimalused	8	8	9	8
Crit8	Interneti Õppimisvõimaluse	10	8	10	10
Crit9	Tulemuste kvaliteet	9	7	8	5
	Kokku	9,00	8,52	8,09	6,70

Valem (1) järgi arvutati välja iga teegi hinne, mis maksimaalselt võis olla 10.

$$y = 0{,}09Crit1 + 0{,}06Crit2 + 0{,}22Crit3 + 0{,}11Crit4 + 0{,}02Crit5 + 0{,}15Crit6 + 0{,}03Crit7 + 0{,}05Crit8 + 0{,}27Crit9 \quad (1)$$

Tehtud analüüsi ja leitud hinnete põhjal selgus, et kõige optimaalsem on kindlustussektoris kasutada TensorFlow'd**,** mis sai hindeks 9,09 punkti maksimaalsest 10-st punktist. Antud teek sisaldab suurel hulgal erinevaid võimalusi, tänu millele on seda võimalik rakendada väga paljudes erineva sisuga kindlustusvaldkonna projektides. Lisaks sellele on teeki võimalik ühendada kõikide teiste TensorFlow platvormi poolt pakutavate võimalustega. See võimalus korvaks antud töö raames kaotatud punkte, tänu millele tagaks antud kombinatsioon kogu vajaliku funktsionaalsuse.

Teisele kohale jäänud PyCaret on teek, mis oma kogufunktsionaalsuse poolest on veidi vähem võimekam kui TensorFlow. Arvestades asjaolu, et teek muudab masinõppe rakendamise erinevates projektides väga lihtsaks, kiireks ning samaaegselt eeldab minimaalset arenduskogemust, siis tegu on teegiga, mis sobib ideaalselt kasutamiseks ka kindlustusvaldkonnas töötavatele ettevõtetele. Teeki sobib kasutada nii ettevõttel, mis soovib teha esmaseid katsetusi masinõppe kasutamisel, et veenduda masinõppe rakendamise sobivuses, kui ka ettevõttel, mis soovib teegi abil leitavaid mudeleid integreerida juba olemasolevate süsteemidega.

Töös mitte optimaalseks osutunud Scikit-Learnil ja fastail olid suuremad puudused rohkemates näitajates. Mõlema teegi korral esinesid puudujäägid mitmetes dokumentatsiooni osades. Samuti oli antud teekidel tulemuste leidmise kiirus või kvaliteet oluliselt halvem kui teistel teekidel. Puudused kiiruses ja kvaliteedis võivad esineda vaid töös testimiseks kasutatud andmekogumikul ning sellest tulenevalt ei saa üheselt väita, et Scikit-Learn ja fastai ei sobi kindlustussektori ettevõtetele.

Masinõppe poolt leitavad tulemused on suhtelised ning mudelite treenimise kiirus kiiruse ja kvaliteet võivad varieeruda väga palju ka sama tüüpi masinõppe ülesannete korral [28]. Samuti võib kiirust mõjutada väga suurel määral see, kas treenitakse CPU või GPU peal. Süvaõppe korral GPU kasutamine võib tulemuste leidmise kiirust muuta mitmeid kordi kiiremaks [29].

Kui tulemustest eemaldada nii mudeli treenimise kiirus kui ka kvaliteet, mis võib sõltuvalt ülesandest suuresti varieeruda, siis muutuvad hinded sarnasemaks, kuid esialgne järjestus säilib. Sellest tulenevalt loetakse endiselt optimaalseteks valikuteks vaid TensorFlow'd ja PyCaretit.

5 Kokkuvõte

Kindlustussektoris pakutavad tooted muutuvad iga aastaga aina enam personaalsemaks, kuid suur osa antud valdkonna ettevõtetest kasutab siiani traditsioonilisi meetodeid. Nende lähenemiste tõttu on järjest keerulisem klientidele võimalikult õiglase hinnaga toodet pakkuda, mis on äärmiselt oluline, et püsida konkurentsivõimelisena.

Antud lõputöö eesmärk oli välja valida masinõppe võimalus, mida käesoleval ajahetkel on kindlustussektori ettevõtetel hinnastamisel kõige optimaalsem kasutada. Analüüsimiseks valiti välja kaks suuremat ja kaks väiksemat Pythoni masinõppe teeki. Iga teegiga loodi masinõppe projekt, mille lõppeesmärgiks oli välja selgitada kliendirisk, mida on omakorda võimalik kasutada hinnastamise mudeli täpsemaks muutmisel. Iga teegi juures hinnati üheksat erinevat näitajat, mille tulemusi kombineeriti AHP tulemustega, et panna paika teekide järjestus, mille põhjal sai teha lõpliku valiku.

Suurimad erinevused teekide juures esinesid dokumentatsiooni ja tulemuste osas. Teekide analüüsi ja hindamise tulemusena selgus, et kindlustussektoris töötaval ettevõttel on kõige optimaalsem kasutada TensorFlow võimalusi. Antud teegil esines analüüsitud näitajate osas kõige vähem puudujääke ning on funktsionaalsuste poolest kõige võimekam. Samuti saab optimaalseks valikuks pidada veidi vähem punkte saanud PyCaretit, mis tänu oma lihtsusele ja kiiretele esmastele tulemustele võimaldab rakendada masinõpet ka ilma varasemate kogemusteta.

Arvestades asjaolu, et iga kindlustussektori projekt ja sellega seonduvad andmed on unikaalsed, siis ühte kõige optimaalsemat masinõppe teeki, mis sobiks kõigile samal ajal, pole võimalik soovitada. Antud valik tuleks ettevõttel teha lähtuvalt enda eesmärkidest ja võimalustest masinõppe rakendamise osas.

Kasutatud kirjandus

[1] C. Blier-Wong, H. Cossette, L. Lamontagne, and E. Marceau, “Machine learning in P&C insurance: A review for pricing and reserving,” *Risks*, vol. 9, no. 1, p. 4, Dec. 2020, doi: 10.3390/risks9010004.

[2] M. Hanafy and R. Ming, “Machine learning approaches for auto insurance big data,” *Risks*, vol. 9, no. 2, p. 42, Feb. 2021, doi: 10.3390/risks9020042.

[3] S. Abdelhadi, K. Elbahnasy, and M. Abdelsalam, “A proposed model to predict auto insurance claims using machine learning techniques,” *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 22, pp. 3428–3437, Nov. 2020. [Online]. Available: https://www.jatit.org/volumes/Vol98No22/8Vol98No22.pdf

[4] L. Guelman, “Gradient boosting trees for auto insurance loss cost modeling and prediction,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3659–3667, Feb. 2012, doi: 10.1016/j.eswa.2011.09.058.

[5] R. Xenidis and L. Senden, “EU non-discrimination law in the era of artificial intelligence: Mapping the challenges of algorithmic discrimination,” *Raphaële Xenidis and Linda Senden,‘EU non-discrimination law in the era of artificial intelligence: Mapping the challenges of algorithmic discrimination’in Ulf Bernitz et al (eds), General Principles of EU law and the EU Digital Order (Kluwer Law International, 2020)*, pp. 151–182, Sep. 2019. Accessed: May 08, 2022. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3529524

[6] K. Kuo and D. Lupton, “Towards Explainability of Machine Learning Models in Insurance Pricing,” *arXiv*, 2020, *Preprints*, doi: 10.48550/arxiv.2003.10674.

[7] J. Selig, “What Is Machine Learning? A Definition,” expert.ai. https://www.expert.ai/blog/machine-learning-definition/ (accessed Mar. 27, 2022).

[8] J. Riley, “AI and Machine Learning Usage in Actuarial Science,” B.S. thesis, Williams Honors College, Honors Research Projects, 2022. [Online]. Available: https://ideaexchange.uakron.edu/honors_research_projects/1081

[9] L. Moroney, *AI and Machine Learning for Coders: A Programmer’s Guide to Artificial Intelligence*, 1st ed. O’Reilly Media, 2020. Accessed: Mar. 05, 2022. [Online]. Available: https://www.oreilly.com/library/view/ai-and-machine/9781492078180/

[10] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, “Extreme gradient boosting machine learning algorithm for safe auto insurance operations,” *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Sep. 2019, pp. 1–5, doi: 10.1109/ICVES.2019.8906396.

[11] G. A. Spedicato, C. Dutang, and L. Petrini, “Machine Learning Methods to Perform Pricing Optimization. A Comparison with Standard GLMs,” *Variance*, vol. 12, no. 1, pp. 69–89, Dec. 2018. Accessed: Apr. 12, 2022. [Online]. Available: https://www.researchgate.net/publication/323202093

[12] P. Embrechts and M. V. Wüthrich, “Recent challenges in actuarial science,” *Annu. Rev. Stat. Appl.*, vol. 9, no. 1, pp. 119–140, Mar. 2022, doi: 10.1146/annurev-statistics-040120-030244.

[13] Y. Grize, W. Fischer, and C. Lützelschwab, “Machine learning applications in nonlife insurance,” *Appl. Stochastic Models Bus. Ind.*, vol. 36, no. 4, pp. 523–537, Jul. 2020, doi: 10.1002/asmb.2543.

[14] Coursera, “What Is Python Used For? A Beginner’s Guide,” coursera.org. https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python (accessed Mar. 27, 2022).

[15] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition*, 3rd ed. Packt Publishing, 2019. Accessed: Mar. 27, 2022. [Online]. Available: https://learning.oreilly.com/library/view/python-machine-learning/9781789955750/

[16] N. Silaparasetty, *Machine Learning Concepts with Python and the Jupyter Notebook Environment: Using Tensorflow 2.0*. Berkeley, CA: Apress, 2020. Accessed: Mar. 8, 2022. [Online]. Available: https://learning.oreilly.com/library/view/machine-learning-concepts/9781484259672/

[17] P. Juneja, “What is Analytical Hierarchy Process (AHP) and How to Use it?,” https://www.managementstudyguide.com/analytical-hierarchy-process.htm (accessed Mar. 27, 2022).

[18] M. Ali, “PyCaret: An open source, low-code machine learning library in Python,” pycaret.gitbook.io. https://pycaret.gitbook.io/docs/ (accessed Mar. 27, 2022).

[19] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research : JMLR*, vol. 12, pp. 2825–2830, 2011. Accessed: Mar. 27, 2022. [Online]. Available: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?ref=https://githubhelp.com

[20] “The fastai deep learning library,” github.com. https://github.com/fastai/fastai (accessed Mar. 27, 2022).

[21] W. Mckinney, *Python for data analysis: data wrangling with Pandas, NumPy, and IPython*, 2nd ed. O’Reilly Media, Inc., 2017. Accessed: Mar. 08, 2022. [Online]. Available: https://learning.oreilly.com/library/view/python-for-data/9781491957653/

[22] “Matplotlib: Visualization with Python,” matplotlib.org. https://matplotlib.org/ (accessed Apr. 05, 2022).

[23] “Virtual Machines (VMs) for Linux and Windows,” azure.microsoft.com. https://azure.microsoft.com/en-us/services/virtual-machines/ (accessed May 15, 2022).

[24] BPMSG, “AHP – High Consistency Ratio,” bpmsg.com. https://bpmsg.com/ahp-high-consistency-ratio/ (accessed Apr. 25, 2022).

[25] N. Ponomareva *et al.*, “TF boosted trees: A scalable tensorflow based framework for gradient boosting,” *Machine learning and knowledge discovery in databases*, vol. 10536, Y. Altun, K. Das, T. Mielikäinen, D. Malerba, J. Stefanowski, J. Read, M. Žitnik, M. Ceci, and S. Džeroski, Eds. Cham: Springer International Publishing, 2017, pp. 423–427, doi: 10.1007/978-3-319-71273-4_44.

[26] “Neural network models (supervised),” scikit-learn.org. https://scikit-learn.org/stable/modules/neural_networks_supervised.html (accessed May 09, 2022).

[27] FOSSA Editorial Team, “All About Permissive Licenses,” fossa.com. https://fossa.com/blog/all-about-permissive-licenses/ (accessed Apr. 02, 2022).

[28] P. Płoński, “Tensorflow vs Scikit-learn,” mljar.com. https://mljar.com/blog/tensorflow-vs-scikit-learn/ (accessed May 12, 2022).

[29] S. I. Baykal, D. Bulut, and O. K. Sahingoz, “Comparing deep learning performance on BigData by using CPUs and GPUs,” *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, pp. 1–6, Apr. 2018, doi: 10.1109/EBBT.2018.8391429.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Tanel Kärvet

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Optimaalse masinõppe võimaluse valimine kindlustussektoris“, mille juhendajad on Inna Švartsman ja Kevin Lehtsalu
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

17.05.2022

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.