

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Henri Lepik 203974IABM

**Andmeanalüüsi tehnikate rakendamine  
klassifitseerimaks detsentraliseeritud  
finantssüsteemi laene**

Magistritöö

Juhendaja: Innar Liiv  
Phd

Tallinn 2023

## **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: Henri Lepik

10.05.2023

## Annotatsioon

Selle töö eesmärgiks on luua lahendus, mis suudab klassifitseerida plokiahelas olevaid laene headeks ja halbadeks laenudeks kasutades masinõppe meetodeid. Töö eesmärgini jõudmiseks on kasutatud nelja erinevat meetodit: logistiline regressioon, otsustuspuu, juhuslik mets ning SVM. Andmestikuna on kasutatud Aave laenukeskkonna andmeid. Tulemuste osas on teostatud analüüs, milles leitakse kõige paremini toimiv mudel. Parim mudel on välja toodud kasutades täpsust, *recall*'i ja F1-skoori tulemusi.

Töö koosneb kirjanduslikust ülevaatest, detsentraliseeritud laenamise kirjeldusest ning laenude klassifitseerimise prototüübist. Vastavalt analüüsi järeldustele tuuakse välja ka soovitusi, mille alusel saaks loodud prototüüpi edasi arendada

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 46 leheküljel, 7 peatükki, 8 joonist, 10 tabelit.

## **Abstract**

### **Applying data analysis techniques to classify loans in the decentralized financial system**

The aim of this thesis is to create a solution that can classify decentralised loans in blockchain as good and bad loans using machine learning methods. Four different methods have been used to achieve this goal: logistic regression, decision tree, random forest and SVM. The used dataset for the prototype is the data from the Aave lending platform. An analysis has been carried out to find the best-performing model. The best model is presented using accuracy, recall, and F1 score results.

The output of the work consists of making a literary overview and describing decentralized lending, as well as creating a loan classification prototype. Based on the conclusions of the analysis, recommendations are also made for further development of the created prototype.

The thesis is in Estonian and contains 46 pages of text, 7 chapters, 8 figures, 10 tables.

## Lühendite ja mõistete sõnastik

AUC	Area under the ROC curve
DT	Decision Tree,
Eksimismatriks	Confusion matrix
Juhitud õpe	Supervised learning
Juhtimata õpe	Unsupervised learning
Kinnitusega õpe	Reinforcement learning
LR	Logistic regression
NN	Neural networks
Pooleldi juhitud õpe	Semi-supervised learning
RF	Random forest
ROC	Receiver operating characteristic curve
Eth	Ethereum
Aave	Aave
LT	Litiquitaon
HF	Health factor
P2P	Peer-to-peer
DeFi	Decentralised finance
DApps	Decentralised Application
SC	Smart Contract
P2P	peer-to-peer
FIAT raha	seaduslik maksevahend

## Sisukord

1	Sissejuhatus .....	10
1.1	Ülesande püstitus ja metoodika.....	11
1.2	Ülevaade tööst.....	11
2	Äriline taustinformatsioon ja kirjanduse ülevaade.....	13
2.1	Krediidiskoorring.....	13
2.2	Detsentraliseeritud finantssüsteem.....	14
2.3	DeFi laenamine .....	15
2.3.1	DeFi laenu platvormid.....	17
2.3.2	DeFi vs traditsiooniline laenamine.....	18
3	Töö tehniline taust.....	20
3.1	Etherum .....	20
3.2	Etherumi nutileping.....	21
3.3	Python.....	21
3.4	Jupyter Notebook .....	22
3.5	Kasutatud raamistikud ja teegid .....	22
3.5.1	Pandas .....	22
3.5.2	Numpy.....	22
3.5.3	Matplotlib.....	22
4	Teoreetilised aspektid.....	24
4.1	Masinõpe .....	24
4.2	Masinõppe algoritmid .....	26
4.2.1	Otsustuspuu .....	26
4.2.2	Juhuslik mets.....	26
4.2.3	SVM - Tugivektor masin .....	26
4.2.4	Logistiline regressioon .....	27
4.3	Masinõppe algoritmide võrdlus.....	27
4.4	Tulemuste mõõtmine.....	28
5	Metoodika .....	30
5.1	Eksperimendi disain.....	30

5.1.1 Andmete alla laadimine.....	32
5.2 Sisendparameetrid .....	35
6 Tulemused ja analüüs .....	40
6.1 Mudelite tulemused .....	40
6.1.1 Logistiline regressioon .....	40
6.1.2 Otsustuspuu .....	41
6.1.3 Juhuslik mets .....	41
6.1.4 SVM .....	42
6.1.5 ROC kõver .....	43
6.2 Järeldused ja analüüs .....	44
6.3 Võimalikud edasised arendused .....	45
7 Kokkuvõte .....	46
Kasutatud kirjandus.....	47
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	50

## Jooniste loetelu

Joonis 1 DeFI laenamise protsess [3].....	16
Joonis 2 Etherumi unikaalsed aadressid [2] .....	21
Joonis 3 Eksperimendi disain.....	30
Joonis 4 Eelnevate laenamiste arv.....	36
Joonis 5 Eelnevalt likvideeritud laenude arv.....	37
Joonis 6 Korrelatsioonimaatriks.....	38
Joonis 7 Likvideeritud ja mitte likvideeritud laenude osakaal.....	39
Joonis 8 ROC kõvera võrdlus .....	43



## Tabelite loetelu

Tabel 1 Laenu platvormid [4].....	18
Tabel 2 DeFI laenamine vs Traditsiooniline laenamine [4].....	18
Tabel 3 Masinõppe algoritmide võrdlus .....	28
Tabel 4 Veamaatriksi näide.....	28
Tabel 5 Ethereum full node tehnilised nõuded .....	33
Tabel 6 Logistilise regressiooni eksimismaatriks .....	40
Tabel 7 Otsustuspuu eksimismaatriks .....	41
Tabel 8 Juhuslik metsa eksimismaatriks .....	42
Tabel 9 SVM mudeli eksimismaatriks .....	42
Tabel 10 Mudelite tulemuste koondtabel .....	44

# 1 Sissejuhatus

Magistritöö töötab välja ja testib mudelit, mis uurib masinõppe meetodite rakendamist Ethereum plokihelal klassifitseerimaks Aave keskkonna laene. Töö on tehtud Etherumi plokihelal olevate Aave keskkonna andmete põhjal.

Magistritöö raames töötab autor läbi hetkel traditsioonilises panganduses kasutatavaid laenu hindamise mudeleid, ning toetudes kirjanduse analüüsile pakub välja parimad meetodid, millega jätkata detsentraliseeritud finantssüsteemides. Masinõppe algoritmide treenimiseks ja testimiseks on vajalik läbida järgnevad etapid. Esmalt tuleb määratleda töös aluseks olevad andmed ning parameetrid, mida on vaja tulemuse saavutamiseks. Järgmiseks etapiks on vaja valida, treenida ning seejärel testida valitud algoritme ning mõõta saavutatud tulemusi. Lõpptulemusena on võimalik välja tuua sobivaim töötav mudel ning seda rakendada.

Probleemi püstitamisel on töö autor seadnud järgimised uurimisküsimused, millele soovitakse töö käigus leida vastused:

- Millised mudelid eksisteerivad laenu klassifitseerimise ennustamiseks traditsioonilises finantssüsteemis?
- Kas masinõppe meetodeid rakendades on võimalik detsentraliseeritud finantssüsteemi laene klassifitseerida headeks ning halbadeks laenudeks?
- Milline meetod töötab kõige paremini laenu hindamiseks?

Töö erineb varasematelt tehtud töödest oma uudsuse poolest. Tänu plokihela ja DeFi kiirele arengule viimastel aastatel on teema kohta vähe teaduslikke uuringuid ning materjali, mistõttu saab seda tööd pidada uudseks.

Tulemuste hindamisel on rakendatud treening- ja testbaasidest juhuslikkuse printsiipi.

## **1.1 Ülesande püstitus ja metoodika**

Töö põhieesmärk on luua tehniline lahendus, mis suudab masinõppe meetodeid rakendades klassifitseerida detsentraliseeritud finantssüsteemis võetud laene. Plokiahela tehnoloogia kiire areng on toetanud ja võimaldanud detsentraliseeritud finantssüsteemide arengut, kus hajusate süsteemide kasutamisel jäetakse välja keskne ehk tsentraliseeritud võim ning kasutajad saavad iseseisvalt teha tehinguid. Samuti on masinõppe algoritmide arenguga finantssüsteemide probleemsete ja halbade laenude tuvastamine paranenud. Seetõttu otsustas autor valdkonnad ühendada ning sõnastas töö põhifookuseks Etherumi plokiahelal olevate Aave keskkonna andmete põhjal tuvastada laenude likvideerimisi.

Töö tulemusena annab autor ülevaate hetkeprobleemist ning analüüsib kirjanduslikele allikatele tuginedes olemasolevaid lahendusi. Töö teises osas tugineb autor kirjandusele ning kirjandusele tuginedes kirjeldab prototüüpi, mis suudab tuvastada likvideeritud laene ning tõestada hüpoteesi, et detsentraliseeritud finantssüsteemides on võimalik kasutaja rahakoti ning varasemate andmete põhjal ennustada kasutaja laenuvõimekust.

## **1.2 Ülevaade tööst**

Töö esimene peatükk annab ülevaate probleemistikust ning tööle püsitatud eesmärgist ja metoodikast .

Töö teine peatükk annab ülevaate traditsioonilise finantssüsteemi tausta ajaloost ja teeb ülevaate detsentraliseeritud finantssüsteemist. Samuti tutvustab autor töö teises peatükis varasemalt tehtud laenude klassifitseerimist puuduvaid töid.

Töö kolmas peatükk kirjeldab ja analüüsib töös kasutatud andmestikku, nende omadusi ja mahtu ning annab ülevaate töö tehnilisest taustast ja töövahenditest.

Töö neljas peatükk kirjeldab töös kasutatavaid masinõppe mudelite teoreetilisi aspekte ning kasutatavaid masinõppe algoritme.

Töö viies peatükk kirjeldab töös kasutatavat metoodikat tulemuste saavutamiseks. Toob välja eksperimendi kavandi, andmete kogumise loogika ja mudelis kasutatavad sisendparameetrid.

Töö kuues peatükk keskendub töö tulemustele ning järeldustele

Töö seitsmendas peatükis võetakse kokku töö eesmärgid ning olulisemad tulemused.

## 2 Äriline taustinformatsioon ja kirjanduse ülevaade

See peatükk annab ülevaate traditsioonilise finantsüsteemi arengust ning ajaloost ja teeb ülevaate detsentraliseeritud finantsüsteemi arengust ja hetkeseisust. Samuti annab autor ülevaate plokiahelal krediidiskooringu leidmist puudutavatest töödest ning olemasolevatest lahendustest.

### 2.1 Krediidiskooring

Krediidiskoori saab defineerida olukorras kui statistilistes mudelites kasutatakse asjakohaseid andmeid arvulisteks mõõtmeks, mis annavad tulemuseks kliendi usaldusväärset. Vajadus saada subjektiivset krediidireitingut mis oleks samal ajal objektiivne, kiire, järjepidev ning oleks aluseks adekvaatsele otsusele, oli peamine põhjuseks, mis pani aluseks krediidiskooringu tehnoloogilisele arengule. [1]

Krediidiskoori eeliseks on võime klassifitseerida laenuvõtjad headeks ja halbadeks laenuvõtjateks. Kirjanduse alusel saab välja tuua, et krediidivõimekus on hinnang laenuvõtjale tema praeguste ja tulevaste kohustuste võimalikuks täitmiseks. Laenumakse õigeaegne mittetasumine või tasumata jätmine on risk nii laenuandjale, kui ka laenusajale. Laenuandja juhul puudub konkreetne teadmine selles osas, kas väljastatud laen laekub tagasi ning laenuvõtjale tekib risk tulevaste uute laenude mittesaamise osas. [2]

Lisaks vähendab krediidiskoor diskrimineerimist, kuna krediidiskoori mudel võimaldab tarbija krediidivõimekust objektiivsemalt analüüsida ning aitab vältida olukordi, kus analüütik või kindlustusandja subjektiivse otsuse langetaks. Seega pakub krediidiskoorimine krediidi väljastajatele võimaluse paremini keskenduda krediidiriskiga seotud teabele. Mudelid koosnevad ainult teabest, mis on olemuselt mittediskrimineeriv ja see on aja jooksul aidanud paremini ette ennustada makse sooritamise võimekust. [1]

## 2.2 Detsentraliseeritud finantssüsteem

Detsentraliseeritud finantssüsteem on süsteem, mis on ehitatud ploki ahelale. Selle tööpõhimõtte toetub loogikale, et hajutatud pearaamatul tehingute tegemiseks pole vaja kolmandat isikut. DeFi suurimaks erinevuseks on vahendajate puudumine süsteemis. Enamik DeFi tehingutest on realiseeritud Ethereumi ploki ahelal. Detsentraliseeritud finantssüsteemi peamiseks eelisteks peetaksegi võimalust kõigil turu osalistel osaleda süsteemi töös. Detsentraliseeritud finantssüsteemi haldab pearaamat kuhu salvestatakse kõik tehingud ning mis on omakorda kõigile osapooltele kättesaadavad teha. Lisaks võimaldab detsentraliseeritud süsteem teha tehinguid ilma tsentraliseeritud osapoolteta. [3]

Põhiliseks vastuargumendiks ploki ahela puhul peetakse asjaolu, et ploki ahela tehingud on anonüümsed, mis ilma kolmanda osapoole volituseta tõstab riski rahapesu ja maksudest kõrvalehoidmise osas. Siiski tasub meeles pidada, et ploki ahela tehingud on jälitatavad ja kontrollitavad ning kõik tehingud on kõigile nähtavad. Seega on kahtlaseid, tehinguid ning osapooli võimalik jälgida, ilma et kontrollorganid või muud asutused tehingute üle kontrolli omaksid. [3]

Detsentraliseeritud finantssüsteem põhineb üldiselt tokenitel. Tokenid on digitaalse varatüübi esindajad, mille väärtuse saab üle kanda ploki ahela andmebaasi. Kõik detsentraliseeritud finantssüsteemi kasutajad hoiavad ja haldavad oma digitaalset vara rahakotis. Rahakotid on DeFi vaste traditsioonilise finantssüsteemi poolt kasutatavatele pangakontodele. [6]

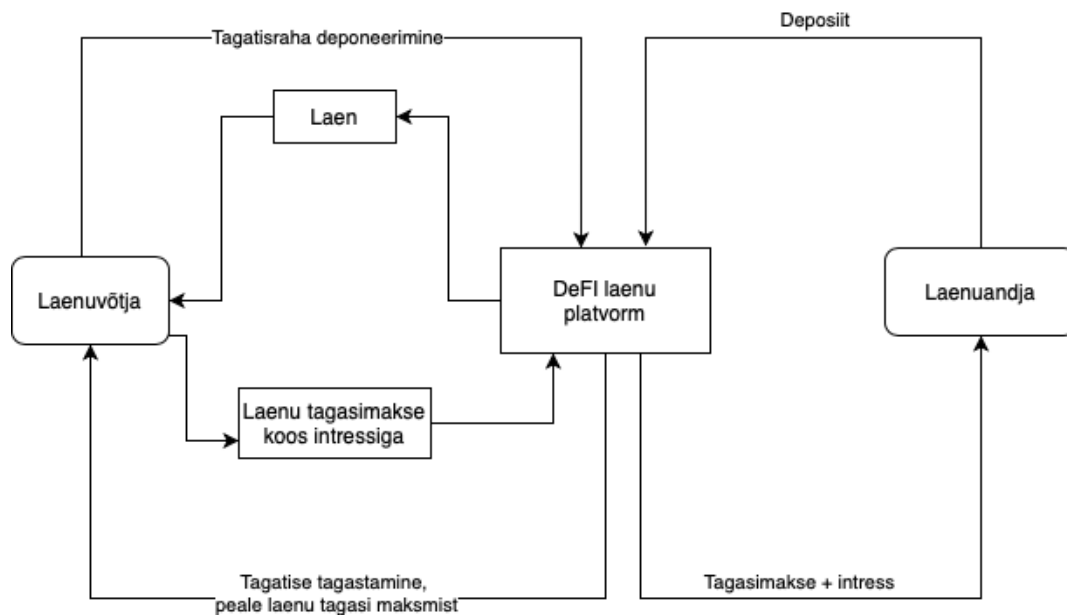
Detsentraliseeritud finantssüsteemi jaoks on olulisemaiks väljakutseks pakkuda turvalist ja usaldusväärset alternatiivi praegusele finantssüsteemile. DeFi platvormid on sõltuvad ploki ahela tehnoloogiast, mis võib olla riskantne, kuna eksisteerib võimalus häkkerirünnakule, mis võib kahjustada kogu platvormi. Siiski võib sama risk tekkida ka panganduses, kus häkkerid võivad samuti rünnata pangatarkvara. Teine väljakutse, mis tekib detsentraliseeritud finantsmaailmas, on regulatiivse raamistiku loomine ja regulatiivsete asutuste kohanemine uue tehnoloogiaga. [4] Üldiselt võib traditsiooniline süsteem vajada mitmeid vahendajaid ja makse turvalisuse aspektid on ühendatud komisjonitasudega, mis muudab maksete pakkumise kallimaks. Maksete töötlemine võtab tavaliselt aega mitmest tunnist kuni kolme päevani, kuid pangad saavad neid

makseid vajadusel kiiremini pakkuda. Plokiahelatehnoloogia võib vastata või ületada pankade võimeid mitmel viisil. [7]

Turu volatiilsus on detsentraliseeritud laenamisele üheks väljakutseks. Tagatise väärtuse langus alla laenu algse väärtuse, kutsub esile likvideerimismehhanismi, mis 2022. aastal kahekordistus, kuna laene ei pakutud stabiilsete krüptorahade abil. Stabiilsed krüptorahad vastavad tavaliselt kindla FIAT valuuta väärtusele, milleks enamasti on USD väärtusele. [35]

## 2.3 DeFi laenamine

Detsentraliseeritud finantssüsteemi oluliseks aluseks saab pidada hajutatud finantseerimise idee kasutuselevõttu, mille eesmärk on pakkuda krediidi- ja hüpoteegihaldusplatvormide loomist ning laenuprotokollide väljatöötamist. Laenamine on DeFis olulisel kohal ning DeFi laenamine on võimalik läbi järgmiste laenuprotokollide: Compound Finance [33], Aave [38] ja MakerDao [37]. Laenuprotokollid võimaldavad kasutajatel laenata raha oma hoiuste tagatisel või teenida intressitulu hoiustatud rahalt, mida laenatakse süsteemisiseselt edasi teistele kasutajatele. Protokollid pakuvad õiglast vahetuskurssi laenatud raha vahetamiseks näiteks USA dollarite vastu. DeFi laenamise ja laenude protsess põhineb tavaliselt hajutatud platvormidel nagu Compound või Aave. Plokiahelasse on loodud ka kesksed ettevõtted nagu BlockFi [39] või Celsius [40], mis on täielikult tsentraliseeritud ja ei erine märkimisväärselt traditsioonilistest pankadest. Need lahendused töötavad sarnaselt traditsioonilistele pankadele sama põhimõtte järgi, kuid kasutavad krüptovaluutasid traditsiooniliste FIAT valuutade asemel. Näiteks juhib juhtiv tsentraliseeritud laenuplatvorm BlockFi hoiuste vara ning annab laene välja oma äranägemise järgi. Selliste "krüptopankade" põhimõte eitab osaliselt krüptovaluutade eesmärki ja kaotab enamiku plokiahela tehnoloogia laenamise ning kasutamise eelistest. DeFi laenamine plokiahela kaudu võib olla laenuturul läbimurdeks ja kasutada täielikult plokiahela tehnoloogiat. DeFi laenamine annab võimaluse inimestele teenida tulu nende krüptovaluutade varadest. Laenamise platvormid võimaldavad kasutajatel turule panna oma hoiused krüptotookenite kujul. Seejärel saab hoiuseid kasutada tagatisena FIAT raha või krüptovaluuta laenamiseks või laenude väljastamiseks teistele kasutajatele nutilepingute abil [34]



Joonis 1 DeFI laenamise protsess [4]

Laenuplatvormide põhimõte on järgnev: kõigepealt kogub protokoll kasutajate raha turult kokku ning kasutab seda seejärel laenude väljastamiseks teistele kasutajatele, raha laenamine ning tingimused on kokkulepitud ja kirjutatud eelnevalt programmeeritud nutilepingutesse. [4]

Laenuandja jaoks on protsess kirjeldatav järgnevalt: [4]

1. Kasutaja valib välja platvormi
2. Kasutaja seejärel deponeerib raha valitud platvormile, mille tingimused on kirjeldatud nutilepingusse.
3. Kui deponeeritud raha võetakse teiste kasutajate poolt kasutusele siis väljastatakse laenu andjale nutilepingu alusel valitud keskkonna tasu, ehk intress. Näitena saab tuua, et Aave puhul tasutakse intress aTokenites, MakerDao keskkonnas aga Dai-na.
4. Deponeeritud varad, mille kasutaja on platvormile deponeerinud, saab igal ajal välja võtta.

Laenuvõtja jaoks on protsess kirjeldatav järgnevalt: [4]

1. Kasutaja valib välja platvormi.
2. Kasutaja tasub tagatisraha, mis tavaliselt on suurem kui laen ise. Näiteks Compoundi platvormil on minimaalne tagatisraha 133%. laenusummast Ülemäärane tagatisraha on vajalik kaitseks krüptovarade äärmiselt kõrge



volatiilsuse vastu. Kõrge tagatisraha põhjuseks on panna tagatisraha müüki enne laenu väärtuse langemist allapoole laenu enda tegelikust väärtust.

3. Kui tagatisraha on kasutaja poolt fikseeritud ja platvormile lukustatud, saab laenaja soovitud vahendid krüptovaluutas.
4. Laenu tähtaja lõpus peab laenaja selle tagasi maksma. Kui seda ei juhtu, siis pannakse tagatisraha oksjonile.

### 2.3.1 DeFi laenu platvormid

Detsentraliseeritud finantssüsteemi suurimad laenuga tegelevad keskkonnad on välja toodud tabelis 1. Selles tabelis on kajastatud viie suurema platvormi nimed, platvormi asutamise aasta, millises infrastruktuuris ja ökosüsteemis antud keskkond tegutseb ja töötab, milline on keskkonnas minimaalne tagatise kordaja laenuvõtjalt, erinevad krüptorahad ja vääringud, mida see keskkond kasutab ja toetab, ning millist valuutat on võimalik anda tagatiseks ning muutujana on välja toodud kogu lukustatud varade väärtuse muutus USDs.

Platvorm	Asutatud	Ploki ahela võrk	Minimaalne tagatise kordaja	Tokenid	Lukustatud varade väärtus 01.05.2023 (TVL)
Compound	2018	Ethereum	1.33	ETH, DAI, USDT, REP, WBTC, BAT, ZRX	1.34 Miljardit USD
Maker Dao	2017	Ethereum	1.5	DAI Tagatiseks saab anda: ETH BAT REP OMG FNT DGD	7.4 Miljardit USD
Aave	2020	Ethereum	1.33	16 valuutat: 13 saab kasutada tagatiseks	5.39 Miljardit USD

Venus	2020	Binance smart chain	2	XVS, USDC USDT BUSD, BNB BTCB, ETC etc	851.8 Miljonit USD
Alpaca finance	2021	Binance smart chain	2	APLACA, BNB, BUSD, USDT BTCB, ETC	310.3 Miljonit USD

Tabel 1 Laenu platvormid [4]

### 2.3.2 DeFi vs traditsiooniline laenamine

Detsentraliseeritud finantssüsteemi laenamine on alternatiiv traditsioonilistele pankadele, krediidivahendajatele ja eralaenajatele. Peamine erinevus traditsioonilise ja hajutatud laenamise vahel seisneb tsentraliseeritud võimu puudumises ja minimaalses vahendajate arvus, kus detsentraliseeritud süsteemis on vahendajate osakaal minimaalne. Lisaks süsteemi osapoolte arvukusele saab välja tuua järgnevad erinevused: [4]

	<b>DeFi laenamine</b>	<b>Traditsiooniline laenamine</b>
Ligipääsetavus	Piirangud puuduvad	Reguleeritud seadusandlusega.
KYC/AML	Hetkel vastavad nõuded puuduvad	KYC /AML on kohustuslik vastavalt seadustele ja regulatsioonidele
Kontroll varade üle	Platvormil puudub kontroll kasutajate varade üle	Klientide raha on hoiustatud vastava institutsiooni kontol
Kontroll	Puudub kontrolliorgan	Erinevad organisatsioonid kes kontrollivad vastavust seadustele, kokkulepetele ja nõuetele.
Kiirus	Kiire protsess, mis on täielikult automatiseeritud,	Aeglane, Palju erinevaid osapooli ning nõudeid, mis vajab täitmist
Platvormid	Compound, Aave, Maker	Pangad, BlockFI, Nexo, Celsisu

Tabel 2 DeFi laenamine vs Traditsiooniline laenamine [4]

Tabelis 2 on välja toodud erinevad aspektid ning erinevused traditsioonilise süsteemi ja detsentraliseeritud süsteemi laenamisest. Esiteks saab igaüks võtta DeFi-platvormil laenu, kui tal on õige kogus krüptovaluutat tagatisraha andmiseks. Kusjuures tsentraliseeritud platvormidel sõltub laenu andmine moderaatori otsusest. Teiseks, DeFi-platvormidel puudub juurdepääs kasutajate rahadele. [4]

Kokkuvõtvalt saab DeFi laenamise puhul väita, et laenud teostatakse anonüümsete laenuvõtjate ja laenuandjate vahel. Intressimäärad määratakse likviidsuse ja nõudluse alusel, mitte laenuvõtja krediitvõimekuse alusel. Tänu automatiseeritusele, reitingute ja õiguskaitse puudumisele ning kõrgele volatiilsusele tähendab see laenude kõrget tagatisnõuet [7]. Selline laenamine erineb traditsioonilisest laenamisest, kus laenuandjad koguvad oma klientide kohta teavet ja kasutavad seda krediitdiskorimiskes läbi masinõppe algoritmide. Misläbi saavad laenuandjad paremini hinnata laenuvõtja riskitaset ning võimekust ja seejärel laenata paremate tingimuste alusel. [8]

### 3 Töö tehniline taust

Töö kolmas peatükk kirjeldab ja analüüsib töös kasutatud andmestikku, nende omadusi ja mahtu ning annab ülevaate töö tehnilisest taustast ja kasutatud tööriistadest.

#### 3.1 Ethereum

Ethereum on hajusvõrk arvutitest, mis töötavad tarkvaraga, mis suudab kinnitada plokkide ja tehingute andmeid. Tarkvararakendust ehk klienti, tuleb käivitada arvutil, et muuta see Ethereumis sõlmeks. Ethereum (ETH) on plokkiahela tehnoloogial põhinev avatud tarkvara platvorm, mis võimaldab arendajatel ehitada ja turule lasta detsentraliseeritud krüptovaluutasid ja projekte, ilma oma plokkiahela ehitamise vajaduseta. [9]

Ethereum on Bitcoinil järel oma turupositsioonilt teine krüptovaluutal. Ethereum ei kujuta endast ainult krüptovaluutana kasutamise võimalust, vaid võimaldab ka uute rakenduste senisest kiiremat arendamisprotsessi ning kasutuselevõttu, pakkudes lahendusi erinevatele tööstusharudele. [9]

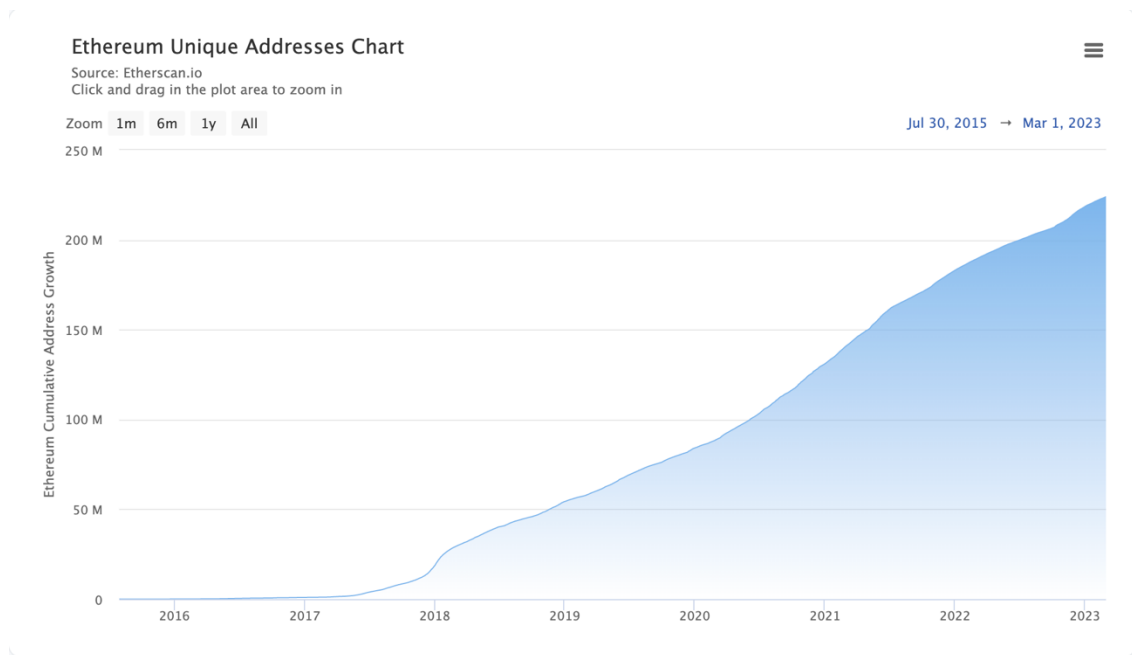
Plokkiahel on digitaalne tehinguraamat, mis salvestab ja kontrollib süstemaatiliselt plokkiahelal tehtavaid tehinguid. Seda kasutatakse Bitcoinis tehingute jälgimiseks ja kontrollimiseks. Kuna ülemaailmsed P2P (inglise k. peer-to-peer) võrgustikud hoiavad plokkiahelat toimimas, on see rikkumatu. Võrku lisatakse pidevalt uusi tehinguid sisaldavaid plokkide, mille õigsust kinnitatakse iga järgmise ploki lisamisel.

Ethereum keskendub rakenduste programmeerimiskoodi käitamisele. Rakenduste arendajad kasutavad seda teenuste ja tehingutasude maksmiseks Ethereumis võrgus.

Ethereum on plokkiahelapõhine detsentraliseeritud platvorm, millele saab ehitada detsentraliseeritud rakendusi (ehk Dapps). Oluline on välja tuua, et plokkiahel on andmebaas, millel puudub keskne server, mis jälgiks iga tehingut. Valdav osa krüpt vahetustest ja detsentraliseeritud projektidest toimivad osaliselt plokkiahela rakendustel.

Etherumi arengut silmaspidades saab välja tuua avatud unikaalsete rahakottide arvu kasvu. Etherumi areng algas aastal 2015 ning sellest ajast on toimunud kiire rahakottide arvu kasv. Jooniselt 2 on välja toodud Etherumi keskkonna areng ja kasvu. Alates 2015

aastast protokoll loomisest kuni tänaseni on kasvanud unikaalsete aadresside arv üle kahe saja miljoni. Joonis 2 mis, kujutab Ethereum unikaalsete aadresside arvu kasvu. [10]



Joonis 2 Etherumi unikaalsed aadressid [10]

### 3.2 Etherumi nutileping

Nutilepingud (inglise k. Smart Contracts) on võimelised automaatselt täitma etteantud tingimuste täitmist. Selle asemel, et saada vajalik nõusolek ühelt keskselt asutuselt, näiteks pangalt, on need lepingud iseseisvad ning suudavad vastu võtta otsuseid reeglite põhjal, mis on nutilepingusse algselt kirja pandud. See mitte ainult ei muuda kogu protsessi tõhusamaks, vaid muudab selle ka õiglasemaks ja objektiivsemaks. Nutilepingud võivad automatiseerida erinevaid ülesandeid, ilma et oleks vaja vahendajaid. Kõik nutilepingu tingimused on sätestatud lepingu osapoolte vahel.

Solidity on objektorienteeritud kõrgetasemeline keel nutikate lepingute rakendamiseks. Solidity abil saab luua lepinguid selliste kasutusviiside jaoks nagu hääletamine, ühisrahastamine, pimeoktsjonid ja mitme allkirjaga rahakotid [11]

### 3.3 Python

Python on üldotstarbeline, objektorienteeritud ja lihtne vabavaraline programmeerimiskeel, mille lõi Guid Van Rossum (Holland) 1991. aastal. Python leiab laialdast kasutamist erinevat liiki tarkvara loomisel, muuhulgas ka veebirakenduste

juures. Kasutamise ulatuselt on ta võrreldav PHP ja Visual Basicuga. Neist kõrgemal on vaid sellised keeled nagu Java ja C-pere keeled (C, C++, C#), mis on eeskätt süsteemprogrammeerimise keeled.[12]

### **3.4 Jupyter Notebook**

Jupyter Notebook on brauseris töötav rakendus, mis võimaldab Python'il suhelda kasutajaliidese vahendusel. Jupyter Notebooki eelisteks saab pidada [13]:

- Interaktiivselt käivitavad lühikesed koodijupid.
- Kogu sisend-väljund hoitakse töölehel, kõiki eelnevaid sisestusi saab redigeerida ja vastavaid arvutusi korrata.
- Koodijuppide vahele saab lisada erineva kujundusega teksti, valemeid ja jooniseid.

Jupyter Notebook sobib arvutuste ja koodi eksperimenteerimiseks ning samal ajal ka dokumentatsiooni kirjutamiseks.[13]

### **3.5 Kasutatud raamistikud ja teegid**

#### **3.5.1 Pandas**

Pandas on Pythoni teek, mis võimaldab paindlikke ning kiireid andmestruktuure Pandas'e peamine eesmärk on muuta struktureeritud andmetega töötamine lihtsamaks ning kiiremaks. Pandas'e andmestruktuurid imiteerivad maatrikseid, tabelleid ja vektoreid [14]

#### **3.5.2 Numpy**

Numpy on Pythoni teek, mis võimaldab töötada mitmemõõtmeliste massiividega, koos suure hulga matemaatiliste funktsioonidega, mida saab omakorda massiividel rakendada. Numpy võimaldab ka andmebaasi integratsiooni.[15]

#### **3.5.3 Matplotlib**

Matplotlib on Pythoni teeks, mida kasutatakse andmete visualiseerimiseks. Matplotlib pakub laia valikut tööriistu ja funktsioone, et luua staatilisi, animatsioonidega ja interaktiivseid visualiseeringuid. Matplotlibit saab kasutada erinevat tüüpi

visualiseeringute loomiseks, sealhulgas jooniste, hajusdiagrammide, tulpdigrammide, histogrammide, 3D-jooniste ja muude jaoks. Matplotlibi kasutatakse laialdaselt erinevates valdkondades, nagu teaduslikud uuringud, inseneritöö, rahandus ja andmeteadus. [16]

## 4 Teoreetilised aspektid

See peatükk annab ülevaate masinõppest üldiselt ning töös kasutatavatest algoritmidest ning meetoditest.

### 4.1 Masinõpe

Arvutiga probleemide lahendamine eeldab sobivate algoritmide kasutamist ning kirjeldamist, mille käigus tehakse arvutile selgeks täpsed juhised ning tingimused mida ja millises järjekorras tuleb teha, et saada sisendist soovitud väljund. Osadel juhtudel on inimesel keeruline täpselt probleemi ja algoritmi kirjutada. Olukorras, kus on olemas suur hulk infot ja andmeid ning me teame, mis mustrit me soovime andmetest otsida, kuid kõikide reeglite täpne kirjapanek oleks väga ajakulukas ning keeruline ülesanne, on probleemide lahendamisel kasulik kasutada masinõppe mudeleid. Selliste probleemide puhul soovitakse, et arvuti oleks ise võimeline looma algoritmi, mis oskab piisava täpsusega olemasolevatest näidisandmetest eraldada probleemi lahendamiseks vajaliku info. Masinõpe (*ML, machine learning*) on osaks tehisintellektis (*AI, artificial intelligence*), mis on võimeline keskkonna muutusega kohanema ning seeläbi omab ka õppimisvõimet.[17]

Masinõppes aluseks on võetud eelnevalt defineeritud muutujad ja mudeli õppimine toimub eelnevalt defineeritud näidisandmete põhjal. Mudel on võimeline ennustama eelnevate andmete ja seoste põhjal.[18]

Masinõppe algoritmide saab grupeerida ja eristada järgnevalt:

- Juhitud õpe (*supervised learning*) – juhitud õppe puhul on sisendiks määratud treeningandmed, mille tulemusel mudel õpib tegema tulemuste osas õigeid ennustusi.[41]
- Juhtimata õpe (*unsupervised learning*) – juhtimata õppe puhul sisendandmed ei ole tulemuse osas märgistatud ja masinõppe mudel peab leidma andmetest sarnaseid struktuure või looma üldistamiseks reegleid[ 41].



- Pooleldi juhitud õpe (*semi-supervised learning*) – pooleldi juhitud õppe puhul on sisendandmed segu kahes eelmisest meetodist, juhitud ja juhtimata õppest ehk märgistatud ja märgistamata näidetest, kus mudel otsib andmetest sarnasusi ning seejärel peab olema suuteline tegema ennustusi [41].
- Kinnitusega õppimine (*reinforcement learning*) – kinnitusega õppe puhul toimub algoritmi õppe katse-eksitus meetodi põhjal. Kus õppimise käigus peab mudel tegema järeldusi ning seejärel leidma millised tegevused toovad parima tulemuse. [41]

Masinõppe arengu käigus on loodud erinevaid algoritme ning loodud algoritmid jaotatakse toimimismehhanismi sarnasuse põhjal. Jaotamise võimalusi on mitmeid ning üks masinõppe algoritm võib definitsiooni kohaselt kuuluda mitmesse erinevasse kategooriasse. Brownle'e on defineerinud ja toob välja erinevad kategooriad masinõppe algoritmidest. [41]

- Juhtumil põhinevad algoritmid (k-lähima naabri algoritm (kNN), ise organiseeruv kaart (SOM), kohapeal kaalutud õppimine (LWL), õppevektori kvantimine (LVQ)) [18]
- Seaduspärasuse algoritmid (Ridge regressioon, elastne võrk, LASSO) [41]
- Otsustuspuu algoritmid (tingimuslikud otsustuspuud, Hii-ruudu automaatne vastastikmõju avastamine (CHAID), üheastmeline otsustuspuu jne) [41]
- Bayesi algoritmid (Naiivne Bayes, Gaussi naiivne Bayes, Bayesi võrk (BN), keskmist ühest sõltuvust ennustavad (AODE)) [41]
- Klasterdamise algoritmid (k-keskmine, k-mediaan, eelduste maksimeerimise algoritm (EM), hierarhiline klasterdamine) [41]
- Tehislikud närvivõrkude algoritmid [41]

Järgmistes peatükkidest teeb töö autor ülevaate algoritmidest, mida on kasutatud töö praktilises osas.

## 4.2 Masinõppe algoritmid

Vastavate tööpõhimõtetega masinõppe algoritme kasutatakse erinevate probleemide lahendamiseks. Järgnevalt on välja toodud masinõppe algoritmid, mida on kasutatud finantssektoris krediidiskooringu tuvastamisel nii panganduses, kui P2P (*peer-to-peer*) laenude hindamiseks.

### 4.2.1 Otsustuspuu

Otsustuspuu on üks osa masinõppe algoritmidest ja andmekaevest. Otsustuspuu on puud meenutav skeem, mis kujutab otsuseid ja nende võimalikke tagajärgi. Eesmärgiks on tulemuste klassifitseerimine või ennustamine etteantud tunnuste põhjal. Otsustuspuu eeliseks peetakse otsustuspuu lihtsust ja arusaadavust. Otsustuspuu miinuseks võib pidada, et see ei ole alati parima täpsusega. Otsustuspuu puhul saabki saavutada kompromissi mudeli täpsuse ja mudeli arusaadavuses osas. [18]

Otsustuspuu struktuur on traditsiooniline, kus aluseks on üks sõlm, mis seejärel jaguneb erinevateks harudeks (ehk oksteks), mis omakorda tipnevad sõlmedega, millest igauks võib edasi hargneda või lõppeda lehega. Iga sõlme juures on küsimus, mis määrab edasise hargnemise järgmise leheni. [18]

### 4.2.2 Juhuslik mets

Juhusliku metsa tööpõhimõtete puhul on tegemist juhusliku valikuga vaatlustest ja parameetritest. Juhuslikku metsa kasutatakse tihti esimese mudelina tulemuste saamiseks ning tulemuste tõlgendamiseks tänu juhusliku metsa eelistele. Peamine eelis on andmete eeltöötlemine ehk parem sõltumatus andmete muutmisest ja andmetes sisalduvast müra kuna mudel ei nõua suurt andmete eeltöötlemist ning andmeid ei pea eraldi hakkama normaliseerima. Lisaks on juhusliku metsa puhul eeliseks parem toimetulek tasakaalustamata andmetega, mida tihti peale võib kohata näiteks halbade laenud tuvastamisel. [18]

### 4.2.3 SVM - Tugivektor masin

SVM ehk Tugivektor masin (*support vector machine*) on masinõppe tööriist, mis kuulub juhitud õppe meetodite hulka. SVM'i kasutatakse klassifitseerimiseks, regressiooniks ja ebanormaalsuste tuvastamiseks. SVM'i otsib vektoreid, mis on asuvad klasside vaatluste

äärealadel. Äärealade abil on võimalik eristada klasse üksteisest. Klasside vahele jäävat ala kutsutakse marginaaliks. [20]

#### 4.2.4 Logistiline regressioon

Logistiline regressioon (ingl. *logistic regression*) või üldisemalt logistiline mudel ehk logit-mudel prognoosib uuritava sündmuse toimumise tõenäosust ja selle muutumist sõltuvalt pideva argumenttunnuse väärtuse muutumisest. Logit-mudeli uuritaval tunnusel võib olla ainult kaks võimalikku väärtust: jah/ei, on/ei ole, esineb/ei esine.[18]

### 4.3 Masinõppe algoritmide võrdlus

Järgnevas tabelis on välja toodud meetodite kokkuvõtlik ülevaade ning algoritmide tugevused ja nõrkused. [20]

Meetod	Tugevused	Nõrkused
Logistiline regressioon	Lihtne rakendada. Kiire treenida, sobib väiksemate klassifitseerimis ülesannete lahendamiseks.	Madalam klassifitseerimise täpsus, ning üle treenimise oht.
Otsustuspuu	Lihtne rakendada, ei nõua suurt arvutusvõimsust omab reaalajas kasutamise potentsiaali	Nõuab pidevat taas treenimist
Juhuslik mets	Sõltumatus ekstreemsetest vaatlustest ning ülesobitamisest.  Ei vaja andmete eeltöötlemist ja	nõuab rohkem arvutusressurssi

	parameetrite eelvaliku teostamist	
--	-----------------------------------	--

Tabel 3 Masinõppe algoritmide võrdlus [19]

#### 4.4 Tulemuste mõõtmine

Sagedusmaatriks on üheks levinud meetodiks valideerimaks ja võrdlemaks mudelite tulemusi. Segadusmaatriksi abil saab välja tuua erinevaid mõõdikuid, nagu täpsus, *recall* ja F1-skoor. Järgnevas tabelis 4 on välja toodud veamaatriks, kus on visualiseeritud tõesed ja valed ennustused. [22]

Ennustatud klass	Tegelik klass	
	Tõene	Vale
Tõene	Tõene Positiivne (TP)	Vale positiivne (VP)
Vale	Vale negatiivne (VN)	Tõene negatiivne (TN)

Tabel 4 Veamaatriksi näide

Tõese positiivsega (TP) on tegu siis kui ennustus ja tegelik väärtus on tõesed.

Tõese negatiivsega (TN) on tegu siis kui ennustus ja tegelik väärtus on valed.

Vale negatiivsega (FN) on tegu siis kui ennustus on vale, kuid tegelik väärtus on tõene

Vale positiivsega (FP) on tegu siis kui ennustus on tõene, kuid tegelik on vale.

F1 – skoor (*F1 – score*) valem (1) näitab skaalal 0 – 1, keskmist täpsuse ja *recall*'i suhet. Mida kõrgem on tulemuse seda parem ja edukam on mudel. Väärtus saadakse läbi keskmise *precision*'i ja *recall*'i suhte.

$$F1 \text{ skoor} = \frac{2*PPV*TPR}{PPV+TPR} = \frac{2*TP}{2TP+FP+FN} \quad (1)$$

*Precision* ehk PPV on mõõdik, mille väärtus kuvab positiivsete väärtuste osakaalu kõikide positiivselt ennustatud väärtuse vastu valem (2). Vastused on 0-1, kus siis suurem väärtus näitab suuremat ja paremat täpsust.

$$Precision (PPV) = \frac{TP}{FP+TP} \quad (2)$$

*Recall* ehk TPR on mõõdik, mille väärtuse tulemus jääb 0 – 1 ning suuremad ja kõrgemad väärtused näitavad paremat tulemust valem (3).

$$Recall (TPR) = \frac{TP}{FN+TP} \quad (3)$$

Täpsus ehk ACC on mõõdik, valem (4) mille väärtus jääb 0-1 vahele ning mida suurem ja kõrgem on väärtus seda paremat tulemust näitab mudel. Väärtus kujutab endast õigesti klassifitseeritud väärtuste osakaalu kõigi väärtuste suhtes.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

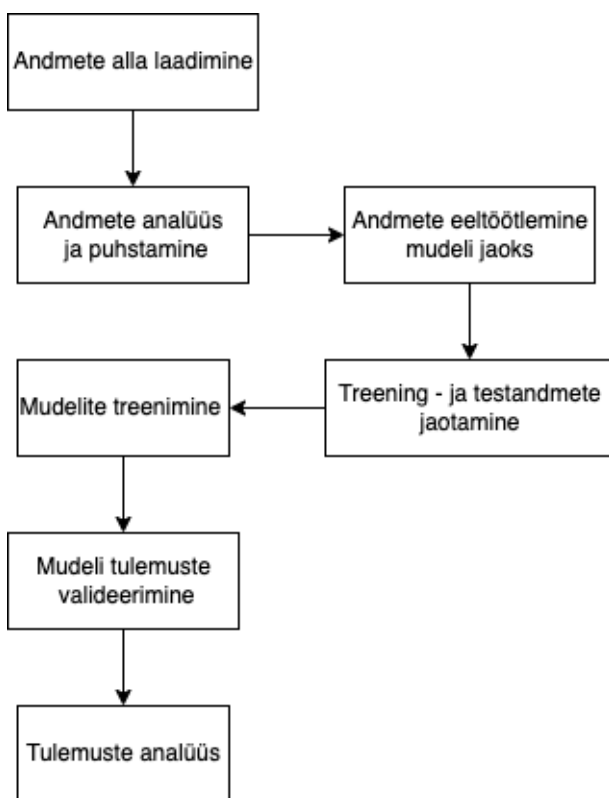
Visualiseerimaks ning kujutamaks mudeleid ühe graafikuna kasutab töö autor ROC graafikut. ROC graafik kaardistab õige positiivset määra vastu vale positiivset määra ning AUC-d ehk kurvi alla jäävat ala kasutatakse kui standardset mõõdikut, hindamaks klassifikatsiooni tulemuslikkust [23]

## 5 Metoodika

Selles peatükis toob autor välja, millistest osadest eksperiment koosneb, kuidas eksperiment on läbi viidud, millised olid andmed ja millised on eksperimentis saadud järeldused. Lisaks on kirjeldatud tulemuste valideerimise meetodid, mille alusel on võimalik teha edasist analüüsi.

### 5.1 Eksperimendi disain

Eksperiment on jagatud järgmisteks etappideks, mis on autori poolt kirjeldatud alltoodud sammudest.



Joonis 3 Eksperimendi sammud

Joonisel 3 on visualiseeritud töö käigus loodud eksperimendi tegevuskäiku.

#### 1. Andmete alla laadimine

Eksperimendi aluseks on andmete kogumine ning ettevalmistamine järgmisteks sammudeks. Eksperimendiks eesmärk on koguda andmed Etherumi ploki ahelast ning

seeläbi tuvastada laenudega seotud tehingud. Seoses plokiahela töö põhimõtetega on otsitavad andmed, kättesaadavad kõigile soovijatele. Seoses andmete avalikult kättesaadavusega on otsitavate andmete saamiseks erinevaid võimalikke meetodeid, ning andmekogumeid. Võimalik on kasutada suuremaid eelnevalt alla laaditud ja ettevalmistatud andmehulkasid, mis on arendajate kommuunile kasutamiseks antud. Need andmed on oma olemuselt plokiahelast eraldatud andmed, ning neid enam pidevalt ei uuendata. Teine võimalus on kasutada erinevaid teenusepakkujaid, kes pakuvad vastava tasu eest - kas piiramatus koguses või piiratud koguses otsitavaid andmeid läbi erinevate API- teenuste. Kolmas võimalus on kasutada teenusepakkujate piiratud andmemahuga, kuid see eest tasuta versiooni andmete saamiseks. Prototüübi arendades ja testides läheneb töö autor kolmandast lahendusest ehk kasutab piiratud andmetega tasuta versiooni. Lahenduse eeliseks on finantsiliselt odavam prototüübi ja idee testimine ning arendamine. Lahendust kasutades on piiranguks kasutatavate andmete maht ning piiratud muutujate valik.

## 2. Andmete analüüs ja puhastamine

Kui andmed on eksperimendiks saadud, siis tuleb andmed kõigepealt puhastada, struktureerida ning ette valmistada treenimiseks, testimiseks ning analüüsimiseks. Selles sammus saab andmetega tutvuda, need puhastada, ning vajadusel tasakaalustada.

## 3. Treening ja testandmete jaotamine

Peale andmete eeltöötlemist ja puhastamist jagas autor andmed treeningandmeteks ning testandmeteks. Misjärel jätkati mudeli treenimise ja testimisega

## 4. Mudeli treenimine

Treeningandmete peal rakendati edasi eelnevalt valitud ja välja toodud algoritme ning peale andmete treenimist testiti tulemusi juba seejärel testandmete peale.

## 5. Tulemuste analüüs

Peale mudeli treenimist ja testimist, on saadud arvulised mõõdikud, mille alusel võrrelda meetodite tulemusi ning hinnata, milline algoritm ja meetod on sobivaim probleemi lahendamiseks.

### 5.1.1 Andmete alla laadimine

Andmekogumine on üks olulisemaid osasid masinõppes, kuna see määrab suuresti töö väljundi ja kvaliteedi: milliseid andmeid kasutati; kust saadi andmed; kui hästi on mudel treenitud; kui täpseid ning häid tulemusi mudel annab. Tänu plokiahela tehnoloogiale, mis kasutab privaatset hajutatud pearaamatut andmete ning tehingute salvestamiseks, on kasutajaspetsiifiliste andmete leidmine palju raskem võrreldes tsentraliseeritud süsteemidega, kus andmed on ühe tsentraalse pakkuja käes. Meetodeid, kuidas üldandmeid plokiahelast on võimalik saada erinevad - igal ühel oma tugevused ja nõrkused. Järgnevalt toob töö autor välja võimalikud erinevad versioonid andmete saamiseks.

- Ethereum *full node* alla laadimine

Etherumil on erinevat tüüpi sõlmi, mis tarbivad andmeid erinevalt. Kliendid saavad käivitada kolme erinevat tüüpi sõlmesid: kerge sõlm (*light node*), täissõlm (*full node*) ja arhiivsõlm (*archive node*). On ka erinevaid sünkroniseerimisstrateegiaid, mis võimaldavad kiiremat sünkroniseerimisega. Sünkroniseerimine viitab sellele, kui kiiresti saab Ethereumi oleku kohta kõige värskemad teavet. Täissõlm (*Full node*) säilitab täieliku plokiahela andmestiku, osaleb ploki valideerimises, kontrollib kõiki plokkide ja olekuid ning teenindab võrku ja pakub andmeid päringu korral. [24]

Täissõlme jooksumiseks on põhipiirang peamiselt kettamaht. Ethereumi plokiahela sünkroonimine nõuab väga palju andmevahetust ja vajab palju ruumi. Andmebaasi suurus ja algse sünkroniseerimise kiirus sõltuvad valitud kliendist, selle konfiguratsioonist ja sünkroniseerimisstrateegiast. Tuleb veenduda, et interneti-ühendust ei piiraks ribalaius. Soovitatav on kasutada piiramatu ribalaiusega ühendust, kuna algne sünkroniseerimine ja võrgule edastatud andmed võivad ületada limiidi. Kõik sõlmed toetavad peamisi operatsioonisüsteeme – Linux, MacOS, Windowsi ehk saab käivitada sõlmi tavalistel töölaua- või serverimasinatel sobiva operatsioonisüsteemiga (OS) [24]



<b>Miimumnõuded</b>	<b>Soovitatud nõuded</b>
CPU 2+ tuumaga	Kiire CPU 4+ tuumaga
8 GB RAM	16 GB+ RAM
2TB SSD	Kiire SSD 2+ TB
10+ Mbit/s ribaläius	25+ Mbit/s ribaläius

Tabel 5 Ethereum full node tehnilised nõuded

Tulenevalt tehnilistest nõuetest ja piirangutest ei kasuta töö autor Etherumi terve ahela alla laadimist ning kasutamist.

Alternatiivina on võimalik andmed laadida, kas serverisse või siis kasutada erinevaid pilve teenuse pakkujaid, nagu Amazon Web Service või Google Cloud Platform andmete haldamiseks. See lahendus töö autori poolt välistati seoses finantsilistest piirangutest

Etherscani kasutamine - Etherscan on enimkasutatud ploki ahela uurimisvahend ja analüüsplatvorm Ethereumile. Etherscan loodi ja käivitati aastal 2015 ning see on üks varaseimaid ja üks pikema ajalooga sõltumatuid projekte, mis loodi Etherumi ja selle kogukonna jaoks, et pakkuda võrdset juurdepääsu ploki ahela andmetele.[25]

Etherscani kasutamine võimaldab esiteks läbi erinevate API-de leida endale sobivaid andmeid. Teine võimalus on pärida endale sobiva ajavahemiku põhjal olevaid tehinguid, ning tõmmata vastav andmefail alla. Mõlemal puhul on andme mahu piirangud. Saada piisavas koguses andmeid edasiseks analüüsiks on võimalik teha premium konto, mida kasutades oleks võimalik saada suurema mahu eest andmeid. Tasuta andmete pärimise mahulistest piirangutest tulenevalt ei kasutanud töö autor valitud meetodikat. [25]

- Aave V2 Health Factor Dataset

Aave V2 tervisefaktori andmestik sisaldab Aave V2 tokeni konto tasemel võlga, tagatise ja likvideerimise väärtusi koos seotud alusvara hindadega. Kus on toodud väärtused koos tagatise ja võla väärtustega, arvutatud likvideerimise väärtused ning sellest tulenevalt arvutatud Aave V2 tervisemuutuja. [18]

Andmestiku eelisteks saab pidada eelnevalt ettevalmistatud ja puhastatud andmekogu, kus on välja toodud tehingu põhiselt, tehingu toimumise aeg, kuhu plokki tehing kuulub, Aave keskkonnas tehtavate valuutade põhine laenuinfo ning konto tervise näitaja olemasolu.

Aave Health Factor on numbriline väärtus, mis kajastab hoiustatud varade ja nende alusväärtuste suhet. Mida kõrgem on muutuja seda turvalisem on laenu positsioon ning seda väiksem on tõenäosus, et laen likvideeritakse. Põhjust, millest alates laen likvideeritakse, ehk kui laen on alakapitaliseeritud peetakse alla ühe ehk kui tervisefaktor langeb alla ühe, siis Aave nutilepingu kohaselt laen likvideeritakse vältimaks suuremaid kaotusi.[26]

See andmestik on hea näide, võimalikest andmest, mida saaks võtta kasutusele edasisel töö. Edasisel andmete uurimisel leidis töö autor, et andmed saadi andmestiku kasutades GraphQLi, millest räägitakse järgmises punktis. Võttes arvesse, et pärida, enda eksperimendiks sobivad andmed, otse allikast loobus töö autor andmestikku kasutamisest. Teiseks miks andmestiku ei kasutatud on saada paremad tulemused järgnevas töö sammudes. Mis võimaldab pärida samu andmeid erinevatest keskkondadest läbi Graphi, ning võimaldab jätkata toote ehitamist läbi Graphi.

- Graph kasutamine

Graph on hajusprotokoll indekseerimiseks ja päringute esitamiseks andmete plokiahelatest. Graph võimaldab andmeid saada lihtsalt ja konkreetselt plokiahelast.

GraphQL on Graphi aluseks olev päringukeel. Erinevus tavaliste RESTful API-päringute ja GraphQL-päringute vahel seisneb selles, et traditsioonilised API-d nõuavad arendajatelt kasutajatele konkreetsete andmete tagastamiseks spetsiifiliste otspunktide loomist. Kui kasutaja vajab rohkem teavet, võib tal vaja mitut API-päringut, mõnikord sadu API-päringuid. Graphi (mis kasutab GraphQL-i) i abil on võimalik pärida andmeid, mis on otseselt raskesti päritavad. Graphi võrgustik on hajutatud indekseerimisprotokoll plokiahela andmete organiseerimiseks. Rakendused kasutavad GraphQL-i, et saada võrgustikust andmeid. Graphi abil saab ehitada serverivabu rakendusi, mis töötavad täielikult avalikul infrastruktuuril.[29]

Töö autor valis andmete pärimiseks GraphQLi, kuna sobivate andmete pärimisloogikat saab rakendada sarnaste laenuplatvormide peale ning võimaldaks erinevate laenu platvormide tulemusi vajadusel rakendada teistel keskkondadel.

## 5.2 Sisendparameetrid

Analüüsiks on kasutatud järgmiseid sisendparameetreid klassifitseerimaks laenu.

Sisendparameetrite saamiseks on kasutatud Graph teenust. Graph siis pärib andmed Aave V2 alamgraphist [24]. Päringu tegemise loogika on järgnev: päritakse kasutajad, kes on eelnevalt laenanud vähemalt ühe korra. Laenamiste korduste arv on defineeritud läbi (*borrowedReservesCount\_gte*: 1) ning antud kasutajate kohta päritakse laenamise ajaloo informatsioon, ning likvideerimiste ajalugu. [28]

*borrowRate* – kasutaja laenamise sagedus

- *id*
- *borroRateMode* –1 stabiilne, fikseeritud , 2 muutuv
- *amount* – laenatud summ (wei)

*liquidationCallHistory* – likvideerimiste ajalugu

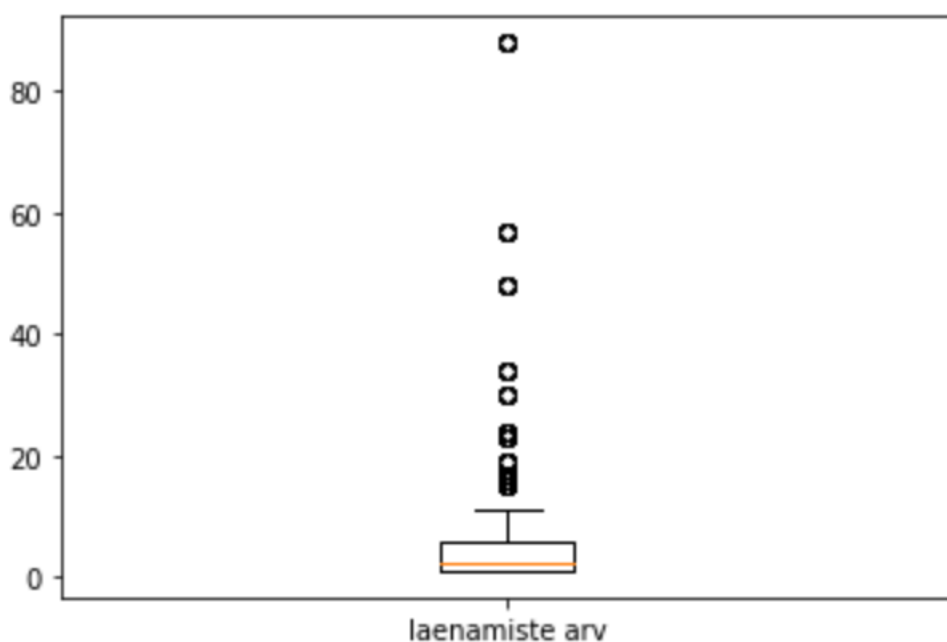
- *id*
- *CollateralAmount* – tagatise väärtus

Andmetest saab tuletada edasi järgmised muutujad, mida saab kasutada laenu klassifitseerimiskes.

- *Rate\_borrowed\_avg* = kasutaja laenamise sagedus / laenamiste arv
- *amount\_borrowed\_avg* = keskmine laenatud summa
- *borrowed\_count* – laenamise korduste arv
- *default\_count* – eelnevalt likvideeritud laenamiste arv
- *default* – likvideeritud laen, muutuja mida uuritakse

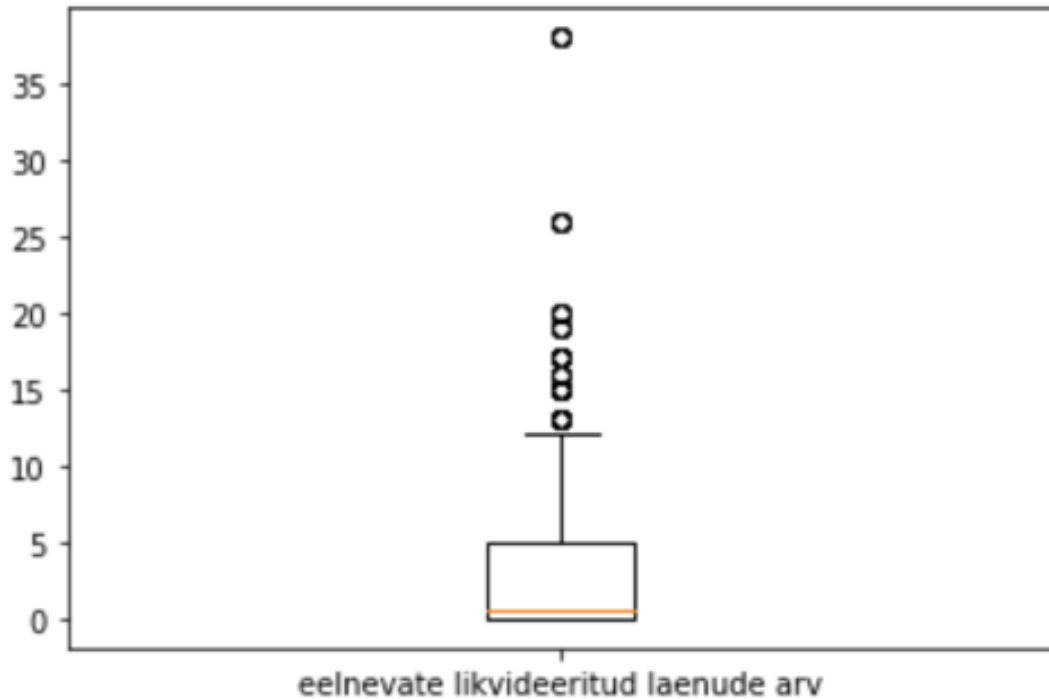
Päringu vastus kirjutatakse seejärel faili, mille alusel saab andmete kallal töötamist ning mudeli ehitamist jätkata.

Peale andmete laadimist ning kirjeldamist keskendub töö autor eraldi järgnevate andmete uurimisse. Esimesena vaadeldakse *borrowed\_count* ehk eelnevate laenamiste arvu summat, mitu korda on konkreetne rahakott teinud laenu tehinguid antud keskkonnas. Laenamiste arvu visualiseerimiseks kasutab töö autor *boxplot* joonist. Joonisel 4 on visualiseeritud kastina laenamiste arvu, kus kasti alumine äär kujutab endast muutuja esimest kvartiili ehk 25%. Väärtuse keskmine on kuvatud oranži joonega kastil. Kasti ülemine äär on ülemine kvartiil ehk 75%. Punktidega on välja toodud äärmuslikud tulemused.



Joonis 4 Eelnevate laenamiste arv

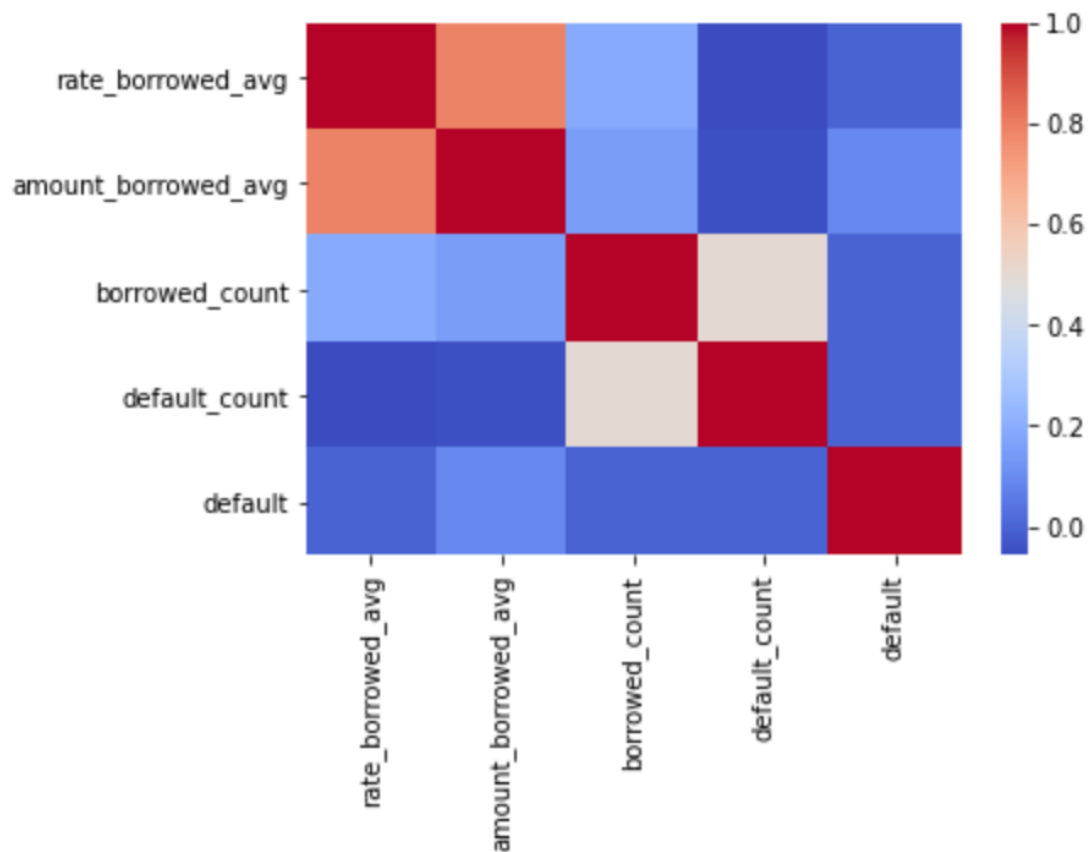
Meid huvitavad andmed, kus saame välja tuua erinevad äärmused andmestikust, kus siis on olemas kontosid, kus on maksimaalselt tehtud näiteks 88 laenamist, kui keskmine laenamist arv on 6.2.



Joonis 5 Eelnevate likvideeritud laenude arv

Joonisel 5 on kirjeldatud eelnevalt likvideeritud laenude arvu *boxplot*'i abil. Eelnevalt likvideeritud laenude arvu keskmine väärtus on 2.8. Jooniselt joondub välja väärtuse ekstreemsused, ehk leidub kasutajaid, kellel on likvideeritud laenude arv 38. Kolmanda kvartiili laenamiste arvud ehk 75% väärtus on 5.

Enne andmete kasutamist mudelites toob autor läbi visuaalse tabeli välja ka korrelatsioonimaatriksi läbi visuaalse tabeli. Korrelatsioonimaatriks on tabel, mis näitab muutujate vahelisi korrelatsioonikordajaid. Tulemusi mõõdetakse vahemikus 0-1, ning mida kõrgem ja suurem on väärtus seda tugevam on seos kahe muutuja vahel (joonis 6). [31] Lisaks saab andmetest välja tuua ka, eelnevad läbikukkunud laenud, kus siis maksimaalne laenude arv on 38, ning keskmine on 5.8

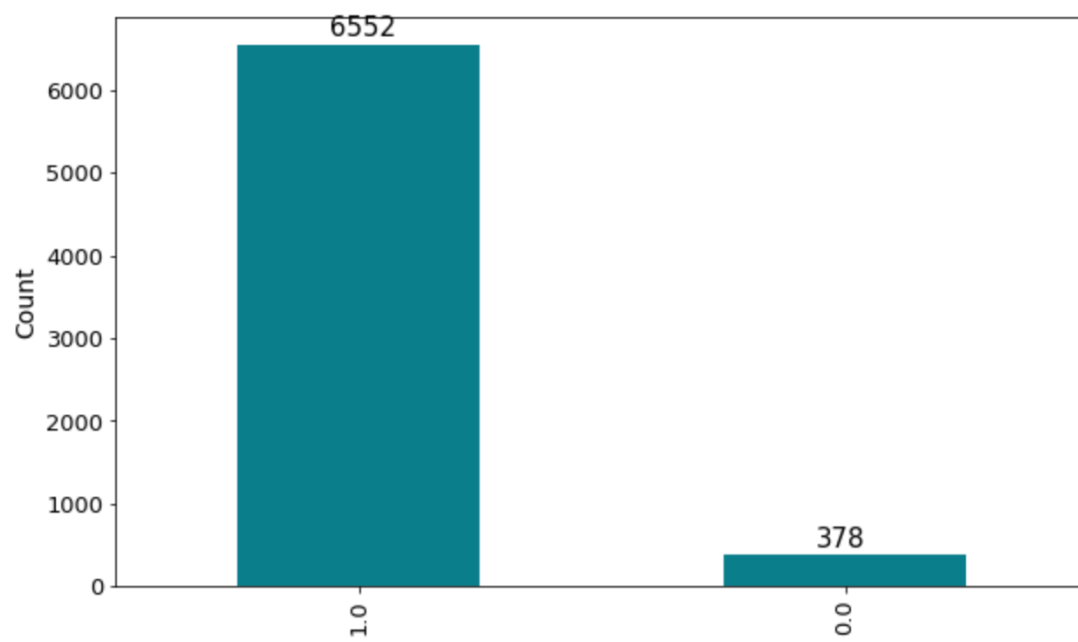


Joonis 6 Korrelatsioonimaatriks

Korrelatsioonimaatriksist saab välja lugeda et suurim seos muutujate vahel on keskmise laenamise sagedusel ja keskmisel laenatud summal. Muude muutujate vahel tugev korrelatsioon puudub.

Peale andmete puhastamist saab otsitavad väärtused ehk siis halvad laenud, mis on läinud likvideerimisele. Head laenud, mida ei ole likvideeritud, grupeerida väärtuste põhjal ning visualiseerida. Jooniselt 7 on näha, et mitte likvideeritud laene on hetkel 6552 ning likvideeritud laenude hulk on 378.

## Laenude jaotus



Joonis 7 Likvideeritud ja mitte likvideeritud laenude osakaal

## 6 Tulemused ja analüüs

Selles peatükis on analüüsitud töös saadud mudelite tulemusi ning välja toodud mudelite põhjal tulemustest tehtavad järeldused. Peatükis vastab töö autor töös püstitatud küsimustele ning toob välja võimalikud edasised arendused.

### 6.1 Mudelite tulemused

Alljärgnevalt on välja toodud iga meetodi tulemused. Mudelite tulemuste saamiseks vaadeldi muutuja  $Y$  ehk *default* väärtusi, kus siis 1 kajastab head laenu ning 0 kajastab siis halba ning likvideeritud laenu.

Töö autor kirjeldab meetodi täpsust, AUC skoori, F1- skoori ning *Recall* tulemust.

#### 6.1.1 Logistiline regressioon

Logistiline regressioon prognoosib uuritava sündmuse toimumise tõenäosust ja selle muutumist sõltuvalt pidevate argumentide tunnuste muutumisest. Logistilise regressiooni läbiviimisel kasutati juba eelnevalt jaotatud treening- ja valideerimisandmeid.

- Mudeli täpsuseks ehk *accuracy* on 0.95 ehk 95%
- Mudeli F1 – skoor on 0.972 ehk 97.2%
- Mudeli *recall* 0.947 ehk 94.7%

Logistilise regressiooni tulemusel saadi eksimismatriks, millelt on näha katse oodatud ja tegelikke tulemusi katsel. Matriksi tulemusest on näha, et 1386-st korrast arvati 1315 korda õigesti ning 71 korral valesti.

Eksimismatriks	0	1
0	74	2
1	69	1241

Tabel 6 Logistilise regressiooni eksimismatriks



### 6.1.2 Otsustuspuu

Otsustuspuu tulemusel saadi eksimismaatriks, millelt on näha oodatud ja tegelikke tulemusi katsel. Maatriksi tulemusest on näha, et 1386st korrast arvati 1318 korda õigesti ning 68 korral valesti.

Eksimismaatriks	0	1
0	54	22
1	46	1264

Tabel 7 Otsustuspuu eksimismaatriks

Otsustuspuu mudeli tulemused:

- Mudeli täpsuseks ehk *accuracy* on 0.95 ehk 95%
- Mudeli F1 – skoor on 0.973 ehk 97.3%
- Mudeli *recall* on 0.965 ehk 96.5%

### 6.1.3 Juhuslik mets

Juhusliku metsa tulemuse saamiseks tuli sarnaselt logistilise regressiooniga vaadelda muutujate Y ehk *default* väärtusi, kus siis 1 kajastab head laenu ning 0 kajastab siis halba ning likvideeritud laenu. Juhusliku metsa läbiviimisel kasutati juba eelnevalt jaotatud treening- ja valideerimisandmeid.

- Mudeli täpsuseks ehk *accuracy* on 0.96 ehk 96%
- Mudeli F1 – skoor on 0.976 ehk 97.6%
- Mudeli *recall* 0.968 ehk 96.8%

Juhusliku metsa mudeli tulemused näitavad kõrget täpsust, täpsuse skoor on 0,96, mis näitab, et 96% mudeli poolt tehtud ennustustest olid õiged. F1-skoor, mis on kaalutud keskmine täpsusest ja tundlikkusest, on 0,976, mis näitab, et mudelil on hea üldine jõudlus nii tõeliste positiivsete kui ka tõeliste negatiivsete ennustuste tegemisel. *Recall* skoor 0,968 näitab, et mudel tuvastas 96,8% positiivsetest juhtudest andmestikus.

Kokkuvõttes viitavad need tulemused sellele, et juhusliku metsa mudel on selle konkreetse klassifikatsiooni ülesandes lahendamise täpne. Siiski nagu igasuguse mudeli puhul on oluline hoolikalt hinnata selle jõudlust konkreetsetes probleemides, mida lahendada püütakse.

Juhusliku metsa algoritmi kasutamise tulemusel saadi eksimismatriks, millelt on näha katse oodatud ja tegelikke tulemusi. Matriksi tulemusest on näha, et 1386st korrast arvati 1269 korda õigesti ning 61 korral valesti.

Eksimismatriks	0	1
0	56	20
1	41	1269

Tabel 8 Juhuslik metsa eksimismatriks

#### 6.1.4 SVM

SVM mudeli tulemusel saadi eksimismatriks, millelt on näha katse oodatud ja tegelikke tulemusi. Matriksi tulemusest on näha, et 1386st-st korrast arvati 1310 korda õigesti ning 76 korral valesti.

Eksimismatriks	0	1
0	0	76
1	0	1310

Tabel 9 SVM mudeli eksimismatriks

SVM ehk tugivektor mudeli tulemusteks saadi

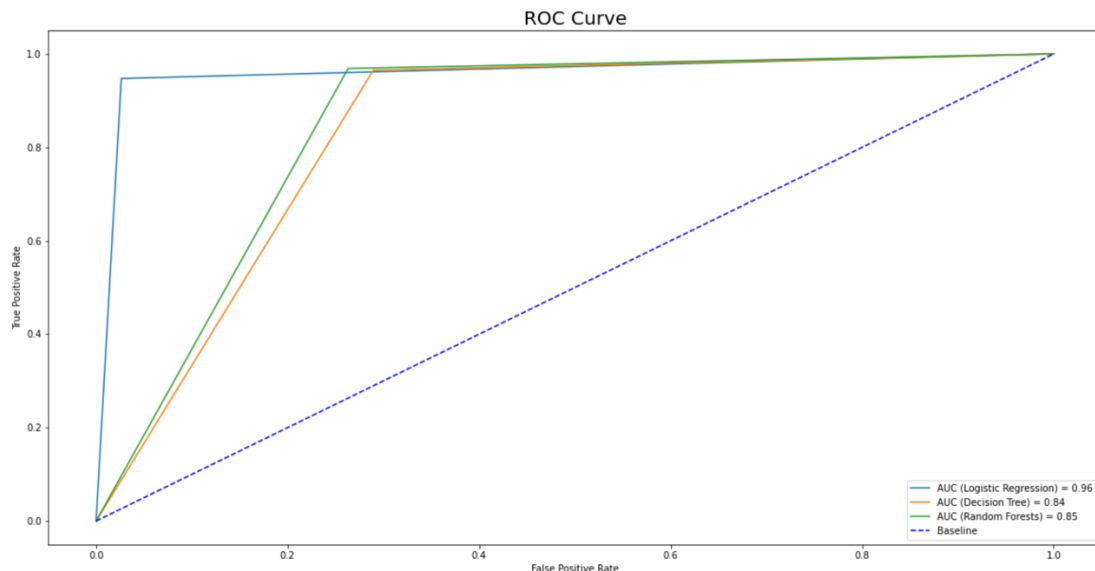
- Mudeli täpsuseks ehk *accuracy* on 0.94 ehk 94%
- Mudeli F1 – skoor on 0.971 ehk 97.1%
- Mudeli *recall* 1.0 ehk 100%

SVM mudeli tulemused näitavad kõrget täpsustaset, täpsuse skooriks on 0,94, mis tähendab, et mudel tegi õigeid ennustusi 94% juhtudest. F1-skoor, mis on täpsuse ja

meeldejätmise kaalutud keskmine, on 0,971, mis näitab, et mudelil on hea üldine jõudlus nii tõeliste positiivsete kui ka negatiivsete juhtumite ennustamisel. *Recall* skoor 1,0 näitab, et mudel tuvastas kõik positiivsed juhtumid andmestikus, mis on paljudes klassifitseerimisülesannetes soovitud tulemus.

Kokkuvõttes viitavad need tulemused sellele, et SVM mudel on tugev esitaja selles konkreetse klassifitseerimise ülesandes ja võib olla väärtuslik tööriist ennustuste tegemiseks sarnastes kontekstides. Siiski on oluline meeles pidada, et mudeli jõudlus sõltub ainult treenimisandmete kvaliteedist ja konkreetsetest tingimustest, mille alusel seda testiti. Seetõttu on oluline hinnata hoolikalt iga mudeli jõudlust konkreetse probleemi kontekstis, mida ta lahendada kavatseb.

### 6.1.5 ROC kõver



Joonis 8 ROC kõvera võrdlus

Joonis 8 kuvab kõigi 4 erineva mudeli tulemusi ROC kõveral. Logistilise regressiooni, juhusliku metsa, otsustuspuu ja SVM mudeli võrdlemiseks on klassifitseerijad kokku võetud ühe mõõdikuga ehk siis AUC, mis on võrdne täpsete laenu tulemuste avastamise tõenäosusega. Sellest tulenevalt saab joonist hästi kasutada ennustusliku täpsuse mõõdikuna. Esimene mudel ehk siis logistiline regressioon, mis on sinist värvi, saavutas parima tulemuse. Logistilise regressiooni mudel saavutas AUC väärtuseks 0.96.

Juhuslik mets, mis saavutas teise parima tulemuse, mis on rohelist värvi, saavutas AUC väärtuseks 0.85. Teine mudel tulemuselt oli *Decision Tree* ehk siis otsustuspuu, mis on oranži värvi, saavutas AUC väärtuseks 0.84. Eraldi on joonisel välja toodud sinise

punktiiriga baasmäär, mis kujutab endast juhuslikkuse alusel jagatud tulemust. Kõikide mudelite puhul saab kokkuvõtvalt öelda, et mudelid suudavad klassifitseerida kumba gruppi kuulub antud laen, tulenevalt kasutaja eelnevast tehingute ajaloost ja andmetest.

## 6.2 Järeldused ja analüüs

Eksperimendi ja prototüübi tulemused olid ootuspärased. Suurimaks piiranguks ja raskuseks oli eksperimendi puhul piisavate andmete saamine ning nende põhjal prototüübi loomine. Töö autor leidis parima andmete saamise vahendiks GraphQLi. Töö autor toob välja andmete pärimisega seotud mahulistest piirangutes, mis ei võimaldanud testida mudeleid suuremate andme mahtudega, kuid esialgse prototüübi ja mudeli tegemiseks sai töö autor piisava andmestiku. Prototüüpiga suutis töö autor tõesta erinevate masinõppe meetodite täpsust ja tõhusust klassifitseerimaks plokiahela laene. Tabelis 10 on välja toodud eksperimendis rakendatud mudelid koos täpsuse, F1 skoori ning *recall*'iga Leidmaks parim ning täpseim mudel.

Meetod	Täpsus	F1-skoor	<i>Recall</i>
Logistiline regressioon	95%	97.2%	94.7%
Otsustuspuu	95%	97.3%	96.5%
Juhuslik mets	96%	97.6%	96.8%
SVM	94%	97.1%	100%

Tabel 10 Mudelite tulemuste koondtabel

Võrreldes nelja erinevat klassifikatsioonimeetodi tulemust, võime näha, et kõik mudelid saavutasid kõrge täpsuse taseme vahemikus 94% kuni 96%. Juhusliku metsa mudelil oli kõrgeim täpsuse skoor 96%. Otsustuspuul ja logistilisel regressioonil olid mõlemal 95% täpsuse skoor. Madalaim täpsuse skoor oli SVM mudelil, väärtusega 94%.

F1-skoori osas, saavutasid kõik mudelid kõrged tulemused vahemikust 97,1% kuni 97,6%. Otsustuspuu ja juhusliku metsa mudelid saavutasid kõrgeima F1-skoori vastavalt 97,3% ja 97,6%.

*Recall* tulemused varieerusid mudelite vahel laiemalt, kus SVM mudel saavutas täiusliku meeldejätmise skoori 100%, samal ajal kui otsustuspuu mudelil oli madalaim meeldejätmise skoor 96,5%.

Kokkuvõttes näitavad tulemused, et kõik neli mudelit esinesid hästi, kus juhusliku metsa ja otsustuspuu mudelid näitasid pisut kõrgemat esinemist F1-skoori ja *recall*'i osas võrreldes logistilise regressiooni ja SVM mudelitega.

### **6.3 Võimalikud edasised arendused**

Töö raames uuriti kasutaja rahakoti laenuvõimekust läbi järgnevate muutujate: keskmine laenamise sagedus, keskmine laenatud summa, laenamiste kord ning eelnevalt likvideeritud laenude arvu. Võimalikuks edasiarenduseks võiks töö autori hinnangul luua sarnase mudeli, mis võtaks arvesse erinevate krüptovaluutade andmeid, millega on tehtud tehinguid.

Teine võimalik edasise arendusena oleks üle kanda Aave süsteemis loodud mudel ja lahendus ning kasutada seda teiste laenu platvormide ja keskkondade peal, mis tegutsevad ja töötavad Ethereum infrastruktuuris olgu selleks Compound, Maker. Selle edasiarendusel oleks vajalik täiendada ja muuta andmete eeltötlust ning andmete alla laadimist ning vastavalt platvormidele ning reeglitele defineerida mudeli parameetreid.

## 7 Kokkuvõte

Töö eesmärgiks oli luua prototüüp, mis oleks masinõppe meetodeid rakendades võimeline efektiivselt klassifitseerima detsentraliseeritud finantssüsteemis võetud laene. Töö esimese väljundina teostas töö autor kirjanduslik ülevaade probleemist ning uuris detsentraliseeritud finantsüsteemi laenamist. Töö käigus uuriti Ethereumi plokiahelas oleva Aave keskkonna tehingute andmeid. Töö tulemusena analüüsiti erinevaid andmete allalaadimise versioone prototüübi tegemiseks, ning leiti, et parim lahendus oleks andmed pärida läbi GraphQL'i, et klassifitseerida plokiahela laene. Töö käigus uuriti nelja erinevat mudelit, logistilist regressiooni, otsustuspuud, juhusliku metsa ja SVM meetodit. Kõigi nelja mudeli tulemused olid kõrged, kuid läbivalt parima tulemuse andis juhuslik mets, millel oli kõrgeim täpsus, F1-skoor.

Saadud tulemused näitavad, et kasutaja rahakotti andmete pealt on võimalik ennustada laenu likvideerimise tõenäosust. Tänu DeFi automatiseeritusele, reitingute puudumisele ning kõrgele volatiilsusele tähendab DeFi laenamine, laenamist kõrgete tagatisnõuete vastu, mis muub laenamise hetkel ebaefektiivseks võrreldes traditsioonilise laenamisega, kuid kasutades masinõppe algoritme on võimalik muuta laenamine efektiivsemaks.

Vastavalt töös saadud tulemustele leiab autor, et rahakoti andmete põhjal on võimalik ennustada kas võetud laen läheb likvideerimisele või mitte. Töö edasiarendusteks saab lisada erinevate keskkondade ja platvormide andmeid, parandamaks mudelite kvaliteet.

## Kasutatud kirjandus

- [1] R. A. Anderson, *The Credit Scoring Toolkit, Theory and Practice for Retail Credit Risk Management and Decision Automation*, New York: Oxford University Press Inc, 2017.
- [2] A. Samreen, „Design and Development of Credit Scoring Model for the Commercial banks of Pakistan. Forecasting creditworthiness of individual borrowers,“ *International Journal of Business and Social Science* , kd. 3, nr 17, 2012.
- [3] D. Gogel, „Defi Beyond the Hype,“ May 2021. [Võrgumaterjal]. Available: <https://wifpr.wharton.upenn.edu/wp-content/uploads/2021/05/DeFi-Beyond-the-Hype.pdf>. [Kasutatud 23 Aprill 2023].
- [4] A. Takyar, „Leewayhertz,“ [Võrgumaterjal]. Available: <https://www.leewayhertz.com/how-defi-lending-works/>. [Kasutatud 20 Aprill 2023].
- [5] R. d. Isidro, 14 Märts 2023. [Võrgumaterjal]. Available: <https://www.globalxetfs.com/aave-the-basics/>. [Kasutatud 20 Aprill 2023].
- [6] Y. Musienko, 2 March 2023. [Võrgumaterjal]. Available: <https://merehead.com/blog/defi-lending-work/>. [Kasutatud 2 Aprill 2023].
- [7] J. B. Marc Truchet, „DECENTRALIZED FINANCE (DeFi): OPPORTUNITIES, CHALLENGES AND POLICY IMPLICATIONS,“ Bandman Advisors, 2022.
- [8] S. D. W. H. A. S. Sirio Aramonte, „DeFI lending: intermediation without information?,“ BIS, 2022.
- [9] Ethereum, [Võrgumaterjal]. Available: <https://ethereum.org/en/developers/docs/nodes-and-clients/>. [Kasutatud 20 Aprill 2023].
- [10] Etherscan, [Võrgumaterjal]. Available: <https://etherscan.io/chart/address>. [Kasutatud 18 aprill 2023].
- [11] Solidity, [Võrgumaterjal]. Available: <https://docs.soliditylang.org/en/v0.8.17/>.
- [12] Taltech informaatikainstituut, „Tutvumine Pythoniga,“ [Võrgumaterjal]. Available: [http://scratch.ttu.ee/failid/Python\\_sisse.pdf](http://scratch.ttu.ee/failid/Python_sisse.pdf).
- [13] Jupyter, [Võrgumaterjal]. Available: <https://jupyter.org/about>.
- [14] Pandas - Python software foundation, [Võrgumaterjal]. Available: <https://pypi.org/project/pandas/>. [Kasutatud 10 Aprill 2023].
- [15] „Numpy,“ [Võrgumaterjal]. Available: <https://pypi.org/project/numpy/>. [Kasutatud 10 Aprill 2023].
- [16] Matplotlib, [Võrgumaterjal]. Available: <https://matplotlib.org/>. [Kasutatud 10 Aprill 2023].
- [17] E. Alpaydın, *Introduction to Machine Learning, Second Edition*, London: The MIT Press Cambridge, 2010.

- [18] J. Brownee, „A Tour of Machine Learning Algorithms, Machine learning mastery,“ 12 August 2019. [Võrgumaterjal]. Available: <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>. [Kasutatud 03 Märts 2023].
- [19] G. Williams, Data Mining with Rattle and R, New York: Springer, 2011.
- [20] V. V. Cortes Corinna, Support-vector networks, Machine Learning, 1995.
- [21] F. Malik, „Machine learning algorithms comparison,“ 28 August 2018. [Võrgumaterjal]. Available: <https://medium.com/fintechexplained/machine-learning-algorithm-comparison-f14ce372b855>. [Kasutatud 26 Aprill 2023].
- [22] A. Tharwat, „Classification assessment methods,“ *Applied Computing and Informatics*, kd. 17, nr 1, pp. 2-3, 2018.
- [23] T. Fawcett, „An introduction to ROC analysis,“ 19 December 2005. [Võrgumaterjal]. Available: <https://people.inf.elte.hu/kiss/11dwhdm/roc.pdf>. [Kasutatud 14 Aprill 2023].
- [24] C. Smith, „Ethereum Node Requirements,“ 14 aprill 2023. [Võrgumaterjal]. Available: <https://ethereum.org/en/developers/docs/nodes-and-clients/run-a-node/>. [Kasutatud 20 Aprill 2023].
- [25] Etherscan, [Võrgumaterjal]. Available: <https://etherscan.io/aboutus>. [Kasutatud 20 Aprill 2023].
- [26] CredProtocol, [Võrgumaterjal]. Available: <https://github.com/credprotocol/Aave-V2-Health-Factor-Dataset>. [Kasutatud 10 Märts 2023].
- [27] Aave, [Võrgumaterjal]. Available: <https://docs.aave.com/faq/borrowing>. [Kasutatud 20 Aprill 2023].
- [28] Graph, [Võrgumaterjal]. Available: <https://thegraph.com/docs/en/network/overview/>. [Kasutatud 20 Aprill 2023].
- [29] Aave, [Võrgumaterjal]. Available: <https://github.com/aave/protocol-subgraphs#usage>. [Kasutatud 10 Aprill 2023].
- [30] Aave, „Lending pool,“ [Võrgumaterjal]. Available: <https://docs.aave.com/developers/v/2.0/the-core-protocol/lendingpool/ilendingpool>. [Kasutatud 20 Märts 2023].
- [31] G. mastrandrea, „Correlation Matrix, demystified,“ 6 juuli 2022. [Võrgumaterjal]. Available: <https://towardsdatascience.com/correlation-matrix-demystified-3ae3405c86c1>. [Kasutatud 26 aprill 2023].
- [32] E. Frangella ja L. Herskind, 27 Jaanuar 2022. [Võrgumaterjal]. Available: [https://github.com/aave/aave-v3-core/blob/master/techpaper/Aave\\_V3\\_Technical\\_Paper.pdf](https://github.com/aave/aave-v3-core/blob/master/techpaper/Aave_V3_Technical_Paper.pdf). [Kasutatud 17 Aprill 2023].
- [33] CredProtoco, 14 Juuli 2022. [Võrgumaterjal]. Available: <https://github.com/credprotocol/Aave-V2-Health-Factor-Dataset>. [Kasutatud 16 Märts 2023].
- [34] Deutsche Bundersbank, „Crypto tokens and decentralised financial applications,“ 31 2021 July. [Võrgumaterjal]. Available: <https://www.bundesbank.de/resource/blob/873132/74fc658db07569219ff74f4e4e55c36f/mL/2021-07-kryptotoken-data.pdf>. [Kasutatud 23 Aprill 2023].
- [35] O. Kharif, Bloomberg, 20 Juuni 2022. [Võrgumaterjal]. Available: <https://www.bloomberg.com/news/articles/2022-06-19/mounting-crypto->



- liquidation-risks-cause-defi-to-go-to-extremes?leadSource=uverify%20wall.  
[Kasutatud 23 Aprill 2023].
- [36] Compound, „Compound,“ [Võrgumaterjal]. Available:  
<https://docs.compound.finance/>. [Kasutatud 20 Aprill 2023].
- [37] MakerDao, „MakerDao,“ [Võrgumaterjal]. Available:  
<https://docs.makerdao.com/>. [Kasutatud 20 Aprill 2023].
- [38] Aave, [Võrgumaterjal]. Available: <https://docs.aave.com/hub/>. [Kasutatud 20 Aprill 2023].
- [39] Blockfi, [Võrgumaterjal]. Available: <https://blockfi.com/crypto-loans/>.  
[Kasutatud 20 Aprill 2023].
- [40] Celsius, [Võrgumaterjal]. Available: <https://celsius.network/crypto-loans>.  
[Kasutatud 20 Aprill 2023].
- [41] I. Arandi, Masinõpe. [Võrgumaterjal] <https://masinope.ee/masinoppimine/> .  
[Kasutatud 02.Mai.2023]

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Henri Lepik

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Andmeanalüüsi tehnikate rakendamine klassifitseerimaks deentraliseeritud finantsüsteemi laene“ mille juhendaja on Innar Liiv
  - 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.

10.05.2023

---

<sup>1</sup> 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 loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.