Stav pôžičky na základe údajov o klientovi v prostredí zdielaných pôžičiek *

Richard Križan and Marek Krátky

Faculty of Informatics and Information Technologies,
Slovak University of Technology in Bratislava
xkrizanr@stuba.sk and xkratkym@stuba.sk
https://www.fiit.stuba.sk/

Abstrakt Hodnotenie spoľahlivosti klientov je typický problém v prostredí bánk. Náš problém je špecifickejší o to, že dáta nám poskytuje spoločnosť, ktorá sa zaoberá zdieľanými pôžičkami. Náš dataset sa dá popísať aj slovom big data. Na spracovanie dát sme preto použili špecifické metódy pre doménu big data. Natrénujeme model, ktorý bude klasifikovať stav pôžičky do 4 kategórií. Model overíme a otestujeme na malej ako aj na veľkej vzorke dát. Výslednú úspešnosť vyhodnotíme na základe metriky LogLoss.

Keywords: Artificial intelligence · Data Mining · Classification · Bank data · Feature selection · Feature extraction · Model training · LogLoss.

1 Analýza

V tejto sekcií popíšeme problém, opíšeme špecifiká dát, nasmerujeme na podobné práce a neskôr vysvetlíme, prečo je dôležité takýto problém riešiť.

1.1 Popis problému

Pôžičky sa stali nevyhnutnou súčaťou veľkého percenta svetovej populácie. Požičať niekomu peniaze však nie je udalosť, ktorá so sebou neprináša žiadne riziko. Každý uchádzač o pôžičku je do určitej miery rizikový z hľadiska schopnosti úplného či včasného splatenia pôžičky. Pre entitu, ktorá peniaze požičiava, môže byť preto informácia o schopnosti splatnosti pôžičky veľmi zaujímavá, pričom môže mať priamy vplyv na udelenie či neudelenie danej pôžičky. Na základe tejto informácie by taktiež bolo možné klientov kategorizovať na základe rizika, ktoré je asociované s nimi požadovanou pôžičkou. Z tohto dôvodu sme sa rozhodli zamerať na predikciu výsledku splatnosti jednotlivých pôžičiek na základe viacerých parametrov a informácií o daných pôžičkách.

* Semestrálna práca na predmet OZNAL šk. rok 2019/2020

1.2 Problémy v dátach

V rámci predspracovania dát sme uskutočnili nasledovné úpravy:

- úprava kategorických atribútov (napríklad encoding)
- zjednocovanie formátov (napríklad úprava formátu dátumu pôžičky)

Náš dataset obsahuje až 145 atribútov. Usudzujeme preto, že je pri trénovaní modelu vhodné zaoberať sa selekciou vhodných atribútov. Selekciou vhodných atribútov vieme zredukovať viacero negatívnych aspektov, ktoré sa v rozsiahlych datasetoch môžu vyskytovať.

- prekliatie dimenzionality - čím viac atribútov máme, tým viac dát potrebujeme na pokrytie priestoru možných hodnôt
- šum - použitie atribútov, ktoré majú v sebe veľa šumu, môže spôsobiť viacero problémov, ako napríklad pretrénovanie. Kvôli nízkej korelácii s predikovaným atribútom spôsobujú pretrénovanie modelu, ktorý sa naučí predikovať cieľový atribút na základe konkrétnych hodnôt týchto atribútov, čo v konečnom dôsledku znižuje schopnosť modelu predikovať cieľový atribút na dátach, ktoré vidí prvý krát výkon - atribúty, ktoré nepomáhajú zlepšiť predikčné schopnosti modelu stále využívajú dodatočné výpočtové prostriedky pri trénovaní, čo spôsobuje zbytočné zvýšenie výpočtovej zložitosti bez akejkoľvek pridanej hodnoty.

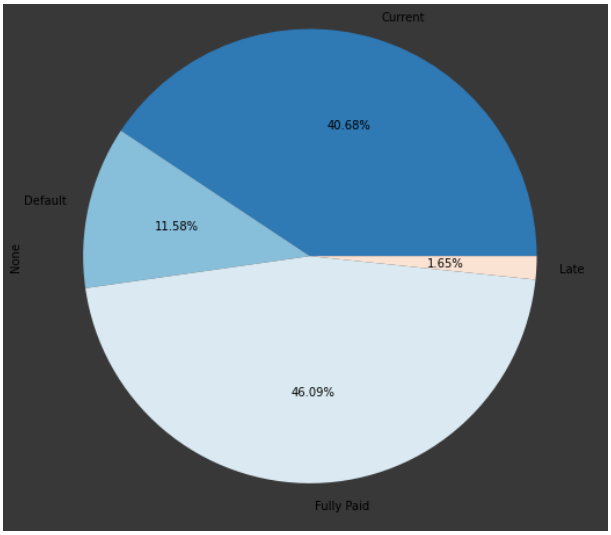
Najprv sme vykonali výber atribútov na základe manuálnej analýzy ich významu a hodnôt, ktoré dosahujú. Po odstránení prázdnych atribútov sme vybrali atribúty, ktoré na základe kontextu korelujú s atribútom, ktorý chceme predikovať.

V ďalšom kroku chceme použiť rekurzívnu elimináciu atribútov (popis sa nachádza pri danej funkcii), aby sme vedeli vybrané atribúty zoradiť podľa relevantnosti vzhľadom k predikcii cieľového atribútu 1.

1.3 Opis dát a ich charakteristiky

Dáta, na základe ktorých sme sa rozhodli trénovať náš predikčný model, pochádzajú z portálu kaggle. Názov datasetu je Lending Club Loan Data. Obsahuje až 2 260 668 pôžičiek so 145 atribútmi. Lending Club je spoločnosť poskytujúca zdielané pôžičky. V tabuľke č. 1 popisujeme výber dát zo zaujímavých stĺpcov.

V dátach sú odlíšené dva typy pôžičiek - joint a individual. Joint pôžička znamená, že ju berie spolu dva a viac ľudí, preto nás pri takýchto pôžičkách nezaujíma napríklad kreditné skóre či iné atribúty žiadateľa ako jednotlivca ale ako skupiny, ktorá berie pôžičku. Následne sme upravovali kategorické atribúty podľa vyššie definovaných funkcií alebo použitím našej vlastnej funkcie encodingu. Ako posledný krok sme zahodili všetky riadky, ktoré napriek uvedenej predpríprave obsahovali prázdne hodnoty. Túto akciu si môžeme dovoliť vzhľadom k množstvu záznamov v našom datasete (2 milióny).



Obr. 1. Rozdelenie atribútu loan_status

open_acc	Počet transakcií, kedy klient čerpal z debetu a sú stále nesplatené
installment	Mesačná splátka
total_acc	Počet transakcií, kedy klient čerpal z debetu a sú splatené
intRate	úrok
sub_grade	Loan Club priradené hodnotenie klienta
term	Počet splátok pôžičky
tot_cur_bal	Aktuálny stav všetkých účtov klienta
purpose	Účel pôžičky
funded_amnt	Pôžička ktorá bola schválená a je čerpaná
___-joint	Ak je stĺpec vyplnení, ide o pôžičku, ktorú požaduje viacero žiadateľov

Tabuľka 1. Opis zaujímavých stĺpcov v datasete.

1.4 Podobne riešené problémy

V tejto sekcií stručne popíšeme podobné riešenia, a ich prístup k problému.

Kaggle Kernel Kaggle kernel [1] obsahuje podrobnú exploratívnu analýzu nášho datasetu. Poskytuje taktiež informácie k doméne, z ktorej dáta pochádzajú. Pre porozumenie závislostiam medzi atribútmi je nevyhnutné porozumieť pôvodu a kontextu skúmaných dát. Analýza nám taktiež poskytla informácie o viacerých trendoch ako napríklad množstvo pôžičiek v určitom období či úspešnosť splatnosti pôžičiek vzhľadom na príjem osôb, ktoré si tieto pôžičky vyžiadali.

[1] https://www.kaggle.com/janiobachmann/lending-club-risk-analysis-and-metrics

Rough sets and logistic regression analysis for loan payment Publikácia [1] sa zaoberá predikciou rizikovosti vystavenia pôžičky na základe schopnosti jej splatnosti. Uvádza a porovnáva rough set prístup a logistickú regresiu. LR bola evaluovaná chi-squared testom. Záver - testované modely LR neboli dostatočne dobré v klasifikácii splatených a nesplatených pôžičiek podľa hodnôt významnosti LR koeficientov s hladinou významnosti 0,05.

Prediction of loan risk using naive bayes and support vector machine V článku [2] sme sa dočítali o typoch pôžičiek a procese, na základe ktorého dochádza k evaluácii žiadateľa o pôžičku. Autori použitím modelov NB (Naive Bayes) a SVM (Support Vector Machine) predikujú z datasetu s 21 atribútmi risk pôžičiek. Modely boli porovnané na základe metriky presnosti. NB dosiahol 77% presnosť, zatiaľ čo SVM dosiahol 79%.

Developing prediction model of loan risk V článku [3] sme sa dočítali, že pri evaluácii vhodnosti udelenia pôžičky sa banka pozerá na "5 Cčok Character (Credit History), Cash Flow, Collateral, Capitalization a Conditions. V článku boli použité tri modely - j48, bayesNet a naiveBayes. Model j48 mal najlepšiu accuracy a nízky mean absolute error.

Decision Tree Classification for Identifying Risky Bank Loans V publikácii [4] popisujú využitie decision tree algoritmu pri identifikácii riskantných bankových pôžičiek. Algoritmus taktiež využíva "divide and conquer"metódu a entropiu. Presnosť predickie je 63%. Porovnali sme najdôležitejšie atribúty z článku s našimi najdôležitejšími atribútmi a zhodujú sa nám najmä: funded amount (koľko z požičky bolo preplatené), purpose (účel pôžičky), aktuálny stav žiadateľa na účte, dĺžka aktuálneho zamestnania, credit history (suma debentých a kreditkých tranzakcií).

1.5 Navrhované metódy vyhodnocovania

Našou úlohou bude vyhododnotiť schopnosť klienta splatiť pôžičku. Túto informáciu nesie náš dataset v atribúte loan_class, ktorý je kategorický a obsahuje celkom 4 kategórie: "Late", "Default", "Fully paid", "Current".

Keďže ide o kategorický atribút, zvolili sme štandardné metriky vyhodnocovania ako napr.:

- Precision-recall
- Log Loss

2 Opis metód

V tejto sekcií si stručne popíšeme metódy data mining vhodné pre náš problém.

2.1 Predspracovanie

Normalizácia dát Je dôležitou súčasťou dátovej analytiky, zabezpečuje nám jednoduchšiu prácu s dátami a lepšiu výpočtovú efektivitu. Často používanou metódou pre chemické údaje je napríklad zmenšovanie údajov do rovnakého rozsahu pre jednotlivé atribúty.

2.2 One hot encoding

Je proces, ktorým dochádza ku konverzií kategorických atribútov na formu, ktorú je možné použiť pri aplikácií machine learning algoritmov. Náš one-hot encoding 2 konvertuje kategorické atribúty na číselné, jedinou výnimkou sa stali stĺpce s dátumom. Tie konvertujeme na číselnú hodnotu, ktorá značí počet dní do dátumu 1.1.2020.

```
def text_to_int(frame,name):
  unique_values = frame.unique();
  counter = 1
  naming = {}
  naming["name"] = name
  for i in unique_values:
    frame = frame.replace(i,int(counter))
    naming[counter] = i
    counter = counter + 1
  naming_dict.append(naming)
  return frame
```

Obr. 2. Implementácia funckie One hot encodingu

2.3 Data mining metódy

Cieľom strojového učenia je použitie takých počítačových systémov, ktoré sa počas svojho použitia dokážu zlepšovať vo vykonávaní ich špecifickej úlohy. Tieto algoritmy často používajú trénovacie dáta na základe ktorých vykonávajú odhady alebo rozhodnutia. V našom prípade budeme vykonávať redukciu dimenzionality či výber čŕt, za účelom zníženia výpočtovej zložitosti nášho problému. Nakoľko

sledujeme vzťahy, ku ktorým máme vstupy aj výstupy, budeme používať učenie s učiteľom.

Učenie s učiteľom je typom implementácie, ktorá na základe už vyriešeného problému hľadá úpravou parametrov najlepšie riešenie pre daný problém a následne toto riešenie testuje aplikáciou v neznámom prostredí.

Binárne rozhodovacie stromy Táto metóda sa používa najmä na modelovanie predikcií. Základom sú vetvy tvorené pozorovaním objektu a listy reprezentujúce závery o hodnote objektu. Delíme ich na klasifikačné a regresné. Klasifikačné majú za úlohu v prípade rozdelenia do tried identifikovať na základe údajov, do ktorej triedy patria dáta. Regresné aproximujú výslednú hodnotu, čo by v našom prípade znamenalo aproximáciu hodnoty spektrálnej veličiny na základe ostatných atribútov.

Random forest Náhodný les je metóda, ktorá býva často použitá na klasifikáciu alebo regresiu. Je zdokonalením binárnych rozhodovacích stromov, pretože rieši chybu, ktorá sa pri ich použití často vyskytne - algoritmus binárnych rozhodovacích stromov si pri trénovaní vytvorí príliš silnú väzbu na trénovací dataset. Cieľom algoritmu náhodného lesu je znížiť koreláciu pridaním faktoru náhodnosti.

Random feature elimination Táto metóda zabezpečuje na základe natrénovaného klasifikátora 3 postupné (rekurzívne) odstraňovanie najmenej potrebných vlastností na základe kritérií, ktoré sme vybrali. Vlastnosti môžu byť odstraňované po jednej alebo po skupinkách za cenu možnosti degradácie výkonnosti.

```
def RFE(X, target):
  #rfc = LogisticRegression(random_state=101)
  rfc = RandomForestClassifier(random_state=101)
  rfecv = RFECV(estimator=rfc, step=1, cv=StratifiedKFold(10), scoring='accuracy', verbose=1)
  rfecv.fit(X, target)


  dset = {}
  attr = X.columns
  return rfecv.estimator_.feature_importances_
```

Obr. 3. Implementácia RFE s využitím random forestu ako klasifikátora.

Kfold Je algoritmus, ktorý nám pomáha rozdeliť dataset na train/test. Algoritmus nám rozdelí dáta na 3 rozdelenia. Tieto rozdelenia musia byť minimálne dve, keďže na trénovanie slúži k-1 rozdelení a na validáciu k-te rozdelenie. Pomocou algoritmu Kfold vieme zabezpečiť taktiež trénovanie dát na vždy inej vzorke datasetu. Pri zvolení väčšieho množstva dát vieme model natrénovať aj na viacerých rozdeleniach.

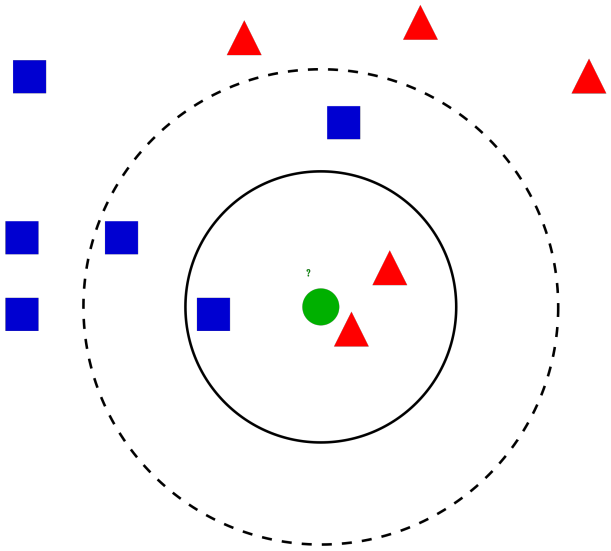
t-SNE Sa radí do skupiny algoritmov na redukciu dimenzionality. Obvykle sa používa na lepšie porozumenie mnoho-dimenzionálnym dátam tak, že ich zobrazí do málo-dimenzionálneho priestoru (obvykle 2D alebo 3D). Jeho použitie je pri našich dátach výhodné preto, že zachovávava nelineárne závislosti medzi atribútmi. To znamená, že nám zachová viac dát ako napr. algoritmus PCA, ktorý zisťuje len lineárne závislosti a snaží sa na zaklade nich zachovať črty dát.

Naive Bayes Je pravdepodobnostný algoritmus, ktorý sa obvykle používa pri klasifikačných problémoch. Využíva najzákladnejšiu pravdepodobnostnú vedomosť a vychádza z naivného predpokladu, že všetky atribúty sú nezávislé. Zvolili sme ho ako základ pre jeden z našich modelov z dôvodu, že napriek svojej jednoduchosti a intuitívnosti dosahuje v mnohých situáciach veľmi dobré výsledky. ďalším dôvodom, prečo sme ho zvolili je jedna z jeho silných stránok - Naivný Bayes dobre zvláda prácu s mnoho-dimenzionálnymi dátami a s veľkým počtom záznamov v dátach.

Support Vector Machine SVM sa sa v ML používa pri modeloch využívajúcich učenie s učiteľom. SVM trénovací algoritmus zostavuje model, ktorý priraďuje nové príklady do jednej alebo druhej kategórie - jedná sa teda o nepravdepodobnostný lineárny klasifikátor.SVM dokáže vykonať aj nelineárnu klasifikáciu tak, že implicitne namapuje svoje vstupy do mnoho-dimenzionálnych priestorov atribútov.

K nearest neighbour Algoritmus KNN je typickým algoritmom strojového učenia, učenia s učiteľom,ktorý slúži na klasifikáciu. Algoritmus hľadá najväčší počet susedov vyskytujúci sa v vzdialenosti K od neklasifikovaného bod a pridelí mu ich triedu. Algoritmus potrebuje trénovacie dáta, na základe ktorých vyhodnotí klasifikáciu predikovaných objektov. Na zistenie vzdialenosti od neklasifikovaného bodu typicky algoritmus využíva euklidovskú vzdialenosť 4.

Logistická regresia Jedným z najzákladnejších modelov určených na klasifikáciu tried kategorického atribútu je logistická regresia. Nemala by preto chýbať v našej sade modelov. Je taktiež ľahko pochopiteľná a interpretovateľná. Vďaka tomu, že sa jedná o veľmi známy základný model, viacero knižníc obsahuje jeho implementáciu, zahŕňajúc taktiež vysokú flexibilitu v rámci práce s rôznymi parametrami, čo nám umožňuje jednoduchú implementáciu optimalizácie tohto modelu.



Obr. 4. Typický príklad modelu KNN

2.4 Vyhodnocovanie a metriky

Accuracy Jedna zo základných metrík na evaluáciu úspešnosti modelu. Reprezentuje, aké percento predikcií z celkového množstva bolo správnych. Sama o sebe nemusí byť dostačujúca, je preto vhodné doplniť jej použitie ďalšími metrikami.

LogLoss Log loss funkcia kvantifikuje presnosť klasifikátora penalizáciou nesprávnych klasifikácií. Čím väčšiu presnosť model má, tým má log loss menšiu hodnotu. Táto metrika je zaujímavá svojou silnou penalizáciou klasifikátorov, ktoré sú si veľmi isté nesprávnou klasifikáciou.

2.5 Optimalizáca modelu

Optimalizáciu parametrov sme uskutočnili na našom modeli logisickej regresie, pretože dosahoval zo všetkých základných modelov najlepšie výsledky (metriky accuracy a log loss). Implementácia optimalizácie bola realizovaná prístupom grid search s využitím krížovej validácie. ukážka parametrov:

```
param_grid = [              {
                "penalty" : ["l2","none"],
                "solver" : ["lbfgs","newton-cg","sag","saga"],
                "max_iter" : [100,1000]                 }
]
```

Zdôvodnenie výberu parametrov Podľa dokumentácie popisujúcej nami použitú implementáciu logistickej regresie sme sa rozhodli použiť z algoritmov určených na riešenie optimalizačného problému algoritmy "lbfgs","newton-cg","sagä "saga", pretože sú použiteľné na klasifikáciu kategorického atribútu s viac než dvoma triedami (v našom prípade štyri). Tieto algoritmy podporujú len L2 regularizáciu, preto testujeme dva varianty - L2 regularizáciu a bez penalizácie. Počet iterácii ako aj hodnoty parametru C, ktorý slúži na ovplyvnenie sily regularizácie sú zvolené intuitívne a na základe štúdia literatúry danej domény. Realizácie optimalizácie parametrov priniesla pozitívny výsledok.

3 Experimenty

V následujúcej sekcií si popíšeme všetky naše experimenty s modelmi. Popíšeme rozdiely výsledkov medzi jednotlivými modelmi a celý proces od načítania dát cez predspracovanie až po vyhodnocovanie a optimalizáciu modelu. Poradie následujúcich sekcií bude preto korešpondovať s reálnym workflowom.

3.1 Načítanie dát

Dáta sme obdržali vo formáte CSV, ktorý sme načítali do pamäte. Dáta obsahovali 145 stĺpcov a viac ako 2 milióny riadkov.

3.2 Predspracovanie

Počas predspracovania sme si vytvorili vlastnú funkciu na one-hot-encoding kategorických atribútov. Taktiež sme počas predspracovania použili normalizáciu na vychýlené atribúty ako napr. out_prncp, dti a iné. Počas predspracovania sme taktiež prekonvertovali stĺpce s dátumom na počet dní po fixný dátum (1.1.2020) [2]. Počas predspracovania sme taktiež zahodili stĺpce, ktoré boli plné hodnôt NaN a unknown. Po ukončení predspracovania mal náš dataset 45 relevantných stĺpcov. Vytvorili sme si tiež dataframy X,y pre budúcu prácu s algoritmami machine learningu.

3.3 Train - Test split

Na rozdelenie dát na trenovaciu a testovaciu časť sme použili funkciu train_test _split. Dataset sme rozdelili v pomere 60-40 trénovacia ku testovacej časti. Zároveň sme pri tejto funkcii využili jej možnosť parametru random_state, ktorý sme nastavili na 42 (štandardne používaná hodnota), čo nám zabezpečilo náhodný výber riadkov pri spustení tejto funkcie.

[2] Všetky naše dátumy boli skôr.

3.4 Random forest ako podporný algoritmus pre feature selection

Využili sme natívnu funkcionalitu algoritmu random forest. Táto funkcionalita poskytuje po natrénovaní modelu hodnotu dôležitosti atribútov v percentách.

```
Variable: out_prncp             Importance: 0.42
Variable: datetime              Importance: 0.25
Variable: application_type      Importance: 0.1
Variable: term                  Importance: 0.08
Variable: int_rate              Importance: 0.04
Variable: loan_amnt             Importance: 0.03
Variable: funded_amnt           Importance: 0.03
Variable: grade                 Importance: 0.02
Variable: sub_grade             Importance: 0.01
Variable: verification_status   Importance: 0.01
Variable: total_acc             Importance: 0.01
Variable: annual_inc            Importance: 0.0
Variable: delinq_2yrs           Importance: 0.0
....                            .....
....                            .....
```

3.5 RFE ako algoritmus feature selection

Na základe dôležitosti atribútov podľa random forestu sme vybrali všetky, ktorých dôležitosť bola nad 4%. Tieto sme pre overenie podrobili algoritmu RFE, ktorý nás uistil, že výber podľa random forestu bol správny a náš model silno závisí iba od troch z nich.

```
["out_prncp","issue_d","application_type"]
[0.801, 0.190, 0.009]
```

3.6 Model Naive Bayes

Výsledky nášho modelu Naive Bayes boli:
LogLoss Gaussian Naive Bayes: 0.38210967485340414
Gaussian Naive Bayes model accuracy(in %): 86.85349729937202

3.7 Model Logistic regression

Výsledky nášho modelu Logistic regression boli:
LogLoss log regression: 0.37991533267352295
Log regression model accuracy(in %): 85.8578852839606

3.8 Model Random Forest

Výsledky nášho modelu Random forest boli:
LogLoss randomForest: 0.5118281593258275
Accuracy randomForest: 0.8687185622339785

3.9 Model KNN

Výsledky nášho modelu KNN boli:
LogLoss KNN: 2.9115753872722414
KNN model accuracy(in %): 82.14428197693331

3.10 Finálny model Logistic regression

Ako najlepší model sme preto vybrali logistic regression pre multi-class problem. Tento model sme ďalej optimalizovali. Optimalizáciou sa nám podarilo implementovať model logistickej regresie s logloss 0,36 a presnosťou 86,9%. Model sme optimalizovali pomocou nasledovných parametrov.

```
Best estimator LogisticRegression(C=1, class_weight=None,
    dual=False,
    fit_intercept=True,
    intercept_scaling=1, l1_ratio=None, max_iter=100,
    multi_class='auto', n_jobs=None, penalty='none',
    random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
    warm_start=False)
```

Model sa preto od najlepšieho neoptimalizovaného zlepšil o 0,01 v metrike logloss a o 1% v metrike accuracy. Metrika logloss bola pre nás rozhodujúcov pri volbe parametrov a finálneho modelu. Dosiahli sme výsledky:
Log regression model accuracy(in %): 86.87190666000231
LogLoss log regression: 0.36933641978268184

4 Záver

Semestrálnu prácu na projekte hodnotíme ako úspešnú. Implementovali sme viacero modelov a následne preskúmali a porovnali ako ich výhody a nevýhody, tak aj úspešnosť predikcií, ktoré sme podrobnejšie predstavili v článku. Podarilo sa nám splniť podmienky projektu a dospeli sme k relevantnému optimalizovanému modelu. Veríme, že náš model je dostatočne znovu použiteľný, aby mohol byť použitý v praxi pre klienta poskytujúceho zdieľané pôžičky. Toto tvrdenie vieme podložiť implementáciou opatrení voči pretrénovaniu modelu, ako aj implementáciou optimalizácie tohto modelu. Pri porovnaní s článkom [2], kde najlepšie modely dosahujú pri riešení podobného problému presnosti 77% (Naive bayes model) a 79% (Support vector machine), sa nášmu optimalizovanému modelu, ktorý využíva logistickú regresiu podarilo dosiahnuť výrazne lepší výsledok - 86.87%.

Literatúra

1. Ruzgar, Bahadtin & Ruzgar, Nursel. (2014). Rough sets and logistic regression analysis for loan payment. 2.
2. VIMALA, S.; SHARMILI, K. C. Prediction of loan risk using naive bayes and support vector machine. In: Int Conf Adv Comput Technol (ICACT). 2018. p. 110-113.
3. HAMID, A. Jafar; AHMED, Tarig Mohammed. Developing prediction model of loan risk in banks using data mining. Machine Learning and Applications: An International Journal (MLAIJ), 2016, 3.1.
4. Decision Tree Classification for Identifying Risky Bank Loans