UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

UPORABA STROJNEGA UČENJA ZA RAZVOJ MODELOV KREDITNEGA OCENJEVANJA

Ljubljana, junij 2020 URBAN BREZIC

IZJAVA O AVTORSTVU

Podpisani Urban Brezic, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Uporaba strojnega učenja za razvoj modelov kreditnega ocenjevanja, pripravljenega v sodelovanju s svetovalcem red. prof. dr. Jurijem Jakličem in sosvetovalcem red. prof. dr. Markom Košakom

I Z J A V L J A M

1. da sem predloženo delo pripravil samostojno;

2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;

3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;

4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;

5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;

6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;

7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;

8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;

9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;

10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne 5. 6. 2020 Podpis študenta: _____

KAZALO

KAZALO TABEL

KAZALO SLIK

SEZNAM KRATIC

angl. – angleško

AJPES – Agencija Republike Slovenije za javnopravne evidence in storitve

AUC – (angl. area under the curve); Območje pod krivuljo

CRISP-DM – (angl. cross-industry process for data mining); Standardizirana metodologija za izvedbo procesa podatkovnega rudarjenja

KNN – (angl. K-nearest neighbor); K-najbližjih sosedov

LN – Lažni negativi

LP – Lažni pozitivi

PCA – (angl. principal component analysis); Metoda glavnih komponent

PIT PD – (angl. point in time probability of default); Sistemi točk v času, točkovna ocena za dani parameter

PN – Pravi negativi

PP – Pravi pozitivi

ROC – (angl. Receiver Operating characteristic curve); Krivulja karakteristike delovanja sprejemnika

SMOTE – (angl. synthetic minority oversampling); Prevzorčenje s sintetičnimi manjšinskimi primeri

SVM – (angl. support vector machine); Metoda podpornih vektorjev

TTC PD – (angl. through the cycle probability of default); Sistemi skozi cikel, dolgoročna ocena vrednosti za dani parameter

UVOD

V zadnjih letih ima bančni sektor vedno večjo vlogo pri spodbujanju gospodarske rasti. Kreditno tveganje je med glavnimi tveganji, ki so jim izpostavljene banke, tako pri odobritvi kreditov kot tudi pri napovedovanju verjetnosti neplačila posojilojemalcev v primeru sklenitve kreditne pogodbe (Khemais, Nesrine & Mohamed, 2016). Da bi to tveganje nadzorovale in povečale svoj dobiček, komercialne banke po vsem svetu razvijajo različne analitične modele (Zhang, 2009). Primarni tehniki za to sta kreditno in vedenjsko ocenjevanje.

Kreditno ocenjevanje (angl. credit scoring) je tehnika, ki pomaga organizacijam pri razlikovanju med prosilci za posojilo na podlagi njihove kreditne sposobnosti (Lyn, 2000). Je možna nadgradnja točkovnih modelov, v katerih se različne spremenljivke ponderirajo na različne načine in rezultirajo v oceni (Jansma, 2018). Temelji na ugotavljanju verjetnosti neplačila na osnovi ugotovljene ocene. Kreditno ocenjevanje je statistična metoda za ocenjevanje verjetnosti dogodka neplačila (angl. default) posojilojemalca z uporabo zgodovinskih podatkov in statističnih metod za določanje enotnega kazalnika, ki loči med dobrimi in slabimi posojilojemalci (Khemais, Nesrine & Mohamed, 2016). Vedenjsko ocenjevanje pa odloča o tem, kako ravnati z obstoječimi strankami, na primer kakšne ukrepe bo banka sprejela, če bo stranka začela zaostajati z vračili (Li, 2012). Cilj obeh tehnik je zagotoviti pogodbeno dogovorjeni denarni tok in zmanjšati izgube zaradi slabih kreditnih odločitev (Hsieh, Lee & Lee, 2010).

Sistem kreditnega ocenjevanja razvrsti stranke v skupine glede na njihovo sposobnost vračanja kredita v določenem obdobju. Cilj tega sistema je, da pomaga pri razvoju procesa upravljanja kreditov ter da kreditnim analitikom zagotovi učinkovito orodje za določanje prednosti, slabosti, priložnosti in tveganj odobravanja kreditov (Hussein, 2011). Opredeljen je tudi način napovedovanja tveganj kreditojemalcev, ki upošteva trenutne gospodarske razmere in se tako samodejno prilagaja gospodarskim spremembam (Lyn, 2000). Uporablja se tudi za prepoznavanje najbolj dobičkonosnih strank.

Kreditno ocenjevanje uporablja kvantitativne meritve uspešnosti in značilnosti preteklih posojil za napoved prihodnje uspešnosti posojil s podobnimi značilnostmi (Khemais, Nesrine & Mohamed, 2016). Funkcija točkovanja temelji na metodi finančnih analiz razmerij, ki so prikazana kot kazalnik, ki lahko razlikuje med zdravimi in propadajočimi podjetji (Khemais, Nesrine & Mohamed, 2016). Boljša kot je funkcija za točkovanje, lažje lahko banka prepozna stranke, katerim se splača posojati. Vsaka manjša razlika v kakovosti modelov kreditnega ocenjevanja vodi do precejšnih razlik v uspešnosti kreditnih portfeljev. Poleg tega lahko banka z manjšimi izboljšavami v teh modelih ohrani konkurenčno prednost pred drugimi bankami (Dumitrescu, Hue, Hurlin & Tokpavi, 2018). Zanesljiv model

ocenjevanja mora temeljiti na strategiji tveganja in kreditni kulturi institucije (Hussein, 2011).

Ločimo med parametrskimi in neparametrskimi modeli. Parametrski oziroma statistični modeli izhajajo iz preverjanja hipoteze o pojasnjevalni moči neodvisnih spremenljivk za odvisno spremenljivko, ki je v našem primeru nastop dogodka neplačila. Neparametrski oziroma algoritemski modeli pa to povezovanje prepustijo algoritmom.

V večini bank za modeliranje sistemov kreditnega ocenjevanja uporabljajo parametrske modele, standardna statistična metoda pa je logistična regresija, saj z njo dobimo linearno kombinacijo spremenljivk z utežmi. Njen rezultat so točke in z njimi povezani razredi kreditne kvalitete (Breiman, 2001). Logistična regresija išče mejo linearne odločitve, predpostavka za to določitev pa je, da ima verjetnost neplačila logistično funkcionalno obliko z argumentom, ki je linearno povezan z napovednimi spremenljivkami (Hué, Hurlin & Tokpavi, 2017).

V zadnjih letih je zaradi razvoja novih tehnologij in večje možnosti zbiranja podatkov prišlo do ponovnega interesa za izboljševanje modelov za ocenjevanje verjetnosti dogodka neplačila. Zato se na področju financ vedno bolj uporabljajo neparametrski modeli oziroma tehnike strojnega učenja (angl. machine learning). V literaturi smo zasledili več trditev, da lahko z uporabo naprednih tehnik, kot so naključni gozdovi (angl. random forest) in nevronske mreže (angl. neural network), dosežemo boljše rezultate kot z logistično regresijo (Venter, 2016).

Glavni razlog uporabe naprednih tehnik je njihova zmožnost modeliranja zelo kompleksnih funkcij (Hussein, 2011). Splošna učinkovitost teh tehnik je boljša od drugih statističnih tehnik. Po drugi strani pa so lahko manj pomembne pri modelih za ocenjevanje kreditne sposobnosti, kjer nosilci odločanja potrebujejo preprosta in razumljiva pravila za napovedovaje (Dumitrescu, Hue, Hurlin & Tokpavi, 2018). Zaradi tega so tradicionalne statistične tehnike, kot sta linearna diskriminantna analiza in logistična regresija, v veliko primerih še vedno ustrezna izbira.

Na splošno ne obstaja ena specifična tehnika izdelave modela ocenjevanja kreditne sposobnosti, ki bi bila optimalna za vse podatkovne zbirke. Izbira modela je odvisna predvsem od podrobnosti problema, strukture in velikosti podatkov, uporabljenih spremenljivk, trga in mejne vrednosti (angl. cut-off point) (Hussein, 2011). Kot smo omenili, obstaja veliko potenciala, da s tehnikami strojnega učenja dobimo boljši model v primerjavi s statističnimi tehnikami. Ker pa obstajajo določene ovire, kot sta na primer kakovost podatkov in vprašanja, ali te tehnike vrnejo razumljive rezultate, je potrebno na konkretnem primeru preveriti in prikazati način uporabe.

Namen magistrske naloge je na konkretnem primeru preveriti možnost uporabe in preučiti ovire razvoja ocenjevalnih modelov, ki temeljijo na pristopih strojnega učenja. Poleg tega je

namen tudi izdelane modele med seboj primerjati in ugotoviti, kateri od teh modelov ima najboljšo napovedno moč ter kateri je najprimernejši za opredeljeni primer.

Za namen priprave magistrske naloge nam je izbrana banka omogočila sodelovanje in svetovanje v samostojnem projektu, kjer so nam dali na uporabo osnovno podatkovno zbirko, ki so jo že sami uporabili za namen modeliranja kreditnega ocenjevanja.

Osnovna tehnika razvijanja ocenjevalnega modela bo logistična regresija, ki se pogosto uporablja v bančništvu. Ta model bomo primerjali z nekaterimi drugimi modeli, ki temeljijo na novejših pristopih, kot so odločitvena drevesa, nevronske mreže, naključni gozdovi in tako dalje. Modele bomo primerjali na podlagi učinkovitosti in razumljivosti rezultatov. Pri primerjavi bo poudarek na vsebini, ocenjevalni modeli, ki jih bomo zasnovali v tem delu, pa ne bodo šli dejansko v uporabo.

Dodana vrednost raziskave bo oblikovanje in vrednotenje novih modelov ter njihova primerjava z modeli, ki jih trenutno uporablja izbrana banka. Izpostavili bomo morebitne ovire pri razvoju in uporabi teh modelov in tako pomagali izbrani banki pri odločitvi, ali je vredno vlagati v modeliranje s tehnikami strojnega učenja.

Raziskovalna vprašanja magistrskega dela so naslednja:

1. So modeli kreditnega ocenjevanja, ki temeljijo na pristopih strojnega učenja, lahko boljši od konvencionalnih, ki temeljijo na logistični regresiji, oziroma ali imajo boljšo **napovedno moč**?
2. Ali v izbranem primeru obstaja potencial za uporabo pristopov strojnega učenja za oblikovanje modela kreditnega ocenjevanja, s katerim napovemo verjetnost dogodka neplačila, in katere so ključne ovire pri tem?
3. Ali je vlaganje v razvoj novega lastnega modela kreditnega ocenjevanja, ki temelji na pristopih strojnega učenja, na dolgi rok v izbranem primeru smiselna investicija?

Cilj magistrske naloge je razvoj več modelov kreditnega ocenjevanja z različnimi tehnikami strojnega učenja, njihova primerjava ter preučevanje možnosti uporabe. S prvim modelom, ki bo oblikovan z logistično regresijo, bomo prišli do meritev napovedne moči za napoved dogodka neplačila. Oblikovali bomo še modele z različnimi pristopi strojnega učenja in nato primerjali rezultate vseh izdelanih modelov. Preučili bomo možnost uporabe ter možne ovire vseh modelov.

V prvem, teoretičnem delu magistrskega dela bo glavna metoda raziskovanja kritični pregled literature na temo modeliranja ocenjevalnih modelov. To bo podlaga za drugi, praktični del, ki bo vseboval pet korakov. Pri teh bomo sledili razvojnim stopnjam metodologije CRISP-DM (angl. cross-industry standard process for data mining). Prvi korak bo pridobitev in priprava internih finančnih podatkov iz bilanc podjetij, ki jih bomo uporabili pri oblikovanju modelov. Podatke bomo pridobili z oddelka za upravljanje s tveganji izbrane banke. Šlo bo za realne podatke podjetij, ki bodo za namen modeliranja in testiranja primerno

anonimizirani. Omejeni bodo na mala in srednja slovenska podjetja. Vsebovali bodo kazalnike oziroma indikatorje o poslovanju podjetij. Drugi korak bo vsebinska interpretacija kazalnikov ter čiščenje in priprava podatkov, da bodo primerni za uporabo v modelih. Določili bomo najpomembnejše kazalnike za napovedovanje dogodka neplačila. Tretji korak bo modeliranje ocenjevalnega modela z logistično regresijo. Ocene se bodo računale na trenutne stranke izbrane banke. Četrti korak bo oblikovanje modelov, ki bodo temeljili na strojnem učenju. Uporabili bomo tehnike strojnega učenja, za katere bomo v prvem delu magistrskega dela ocenili, da so za ta namen najbolj primerne. Zadnji, peti korak bo primerjava in ocena rezultatov. Primerjava modelov bo temeljila na smiselnosti analize in napovedni moči. Naše izdelane modele bomo lahko primerjali tudi z obstoječim bančnim. V sodelovanju z izbrano banko bomo s preverjanjem in intervjuji z uslužbenci iz banke primerjali rezultate oblikovanih modelov in na podlagi rezultatov odgovorili na raziskovalna vprašanja.

V prvem poglavju bomo opredelili kreditno ocenjevanje, opisali postopek razvoja in opredelili izzive, ki se pri tem pojavljajo. V drugem poglavju bomo definirali strojno učenje in nato opisali možnost njegove uporabe za kreditno ocenjevanje. Opisali bomo metodologijo za izvedbo procesa podatkovnega rudarjenja ter korake za pripravo podatkov. V tretjem poglavju bomo predstavili tehnike, ki jih bomo uporabili pri modeliranju, in predstavili lastnosti, njihove prednosti in slabosti, povezane s kreditnim ocenjevanjem. V četrtem poglavju bomo podrobno opisali podatkovno zbirko in vsebinsko interpretirali vsak posamezen kazalnik. Opisali bomo, kako smo očistili oziroma pripravili podatkovne zbirke za uporabo pri modeliranju. V petem poglavju bomo opisali postopek modeliranja z vsemi tehnikami in predstavili njihove rezultate. V šestem poglavju bomo povzeli ugotovitve.

1 MERJENJE KREDITNIH TVEGANJ

Kreditno ocenjevanje se je začelo razvijati v šestdesetih letih prejšnjega stoletja, ko se je začelo poslovanje s kreditnimi karticami in z naraščanjem popularnosti postopkov za samodejno odločanje (Brunel, 2016). Z razvojem poslovanja s kreditnimi karticami se je tehnologija široko uporabljala pri odločanju o posojilih, oblikovanju cen premoženja in upravljanju poslovnih posojil (Estrella, 2000). Igrala je zelo pomembno vlogo pri upravljanju s kreditnim tveganjem poslovnih bank (Hui, Li & Zongfang, 2013). Prvi model, ki je uporabljal finančna razmerja za prepoznavanje ali napovedovanje dogodka neplačila, je razvil Edward Altman leta 1968. Ta model je uporabljal linearno diskriminantno analizo in se še danes uporablja za primerjavo z drugimi modeli (Altman, 2018).

Zaradi povečevanja konkurence na področju vedenjskega ocenjevanja se v zadnjih letih otežuje privabljanje in zadrževanje dobičkonosnih strank z nizkim tveganjem. Ker se na splošno večina poslovnih prihodkov bank, včasih tudi do 90 odstotkov, ustvari z večkratnimi transakcijami obstoječih strank in ker se njihovo finančno stanje sčasoma spreminja, je zelo pomembno, da se jih redno spremlja (Hsieh, Lee & Lee, 2010). Z napovedovanjem bodoče

uspešnosti poslovanja modeli vedenjskega ocenjevanja finančnim institucijam omogočajo hitrejše in boljše odločanje za ohranitev kreditno sposobnih strank.

Določitev verjetnosti dogodka neplačila je v okviru kreditnega tveganja ena glavnih težav, s katerimi se morajo spoprijeti banke in druge kreditne družbe (Cao in drugi, 2009). Za to se uporabljajo modeli kreditnega ocenjevanja, ki so opredeljeni kot sistemi za podporo odločanju, ki pomagajo upravljavcem pri procesu sprejemanja odločitev o kreditnih naložbah in s tem zmanjševanju kreditnih izgub. Za izvajanje in razvoj modelov kreditnega ocenjevanja sta diskriminantna analiza in logistična regresija sprejeti kot standardni tehniki (Abid, Masmoudi & Zouari-Ghorbel, 2018).

1.1 Razvoj modelov kreditnega ocenjevanja

Razvoj sistema kreditnega ocenjevanja v bankah je z upravljavskega vidika zelo kompleksen postopek. V literaturi smo zasledili opis načrta, kako se po najboljših praksah to izvede (Caire & Kossmann, 2003). Pred oblikovanjem modela je najprej potrebno razumeti, kakšen sistem bi lahko deloval v izbranem primeru. Na podlagi tega se pripravi akcijski načrt, nakar se sestavi upravni odbor za razpravo o strateških in tehničnih težavah, ki bo ključnega pomena pri usmerjanju celotnega projekta. Po tem se začne oblikovanje in preizkušanje modelov. Temu sledi uvodno usposabljanje modela za določitev uspeha in priprava poročila za upravni odbor. Preden se odobri za formalno uporabo, je potrebno pripraviti postopke in zagotoviti, da je skladen s splošnimi pravili banke. Nazadnje se spremlja uspešnost modela in se ga po potrebi prilagaja.

Samo oblikovanje bančnih sistemov kreditnega ocenjevanja je tehnično najbolj zahteven korak. Postopek lahko razdelimo na naslednje module (Engelmann & Rauhmeier, 2011):

1. **Strojna ocena**: mehanski algoritem ustvari prvi predlog za predvideni dogodek neplačila posojilojemalca na podlagi njegovih bilančnih razmerij. Ta algoritem temelji na statističnih modelih.
2. **Strokovno vodene prilagoditve**: omogoča, da bančni strokovnjaki in analitiki prilagodijo kreditno oceno z določenimi podrobnostmi, ki niso dovolj izražene v prvem modulu. To se običajno izvede v standardizirani obliki z izbiro vnaprej določenih elementov in oceno njihove pomembnosti.
3. **Logika podpornikov**: posojilojemalcem, ki so na meji opredelitve kot slabi posojilojemalci, se zajame učinke morebitne podpore. Tu se uporabljajo potencialne ocene podpornikov, ki so skladne z vnaprej določenimi smernicami.

Po postopku teh treh modulov dobimo predlog definiranih pravil določanja ocene, ki pove, ali bo obravnavani posojilojemalec ocenjen kot dober ali slab. Ker ni mogoče predvideti vseh dogodkov, ki vplivajo na kreditno sposobnost posojilojemalca v postopku oblikovanja modela, sistemi kreditnega ocenjevanja omogočijo prilagoditev posameznih pravil ocene. Popravila morajo ponavadi biti dobro dokumentirana, utemeljena in odobrena (Engelmann

& Rauhmeier, 2011). Opisan postopek oblikovanja se lahko v praksi močno razlikuje. Moduli so izpuščeni, če so nepomembni, ali preveč stroškovno zahtevni glede na pričakovane koristi.

1.2 Izzivi pri razvoju

Cilj modelov kreditnega ocenjevanja je dobro ločevanje med potencialnimi plačniki in neplačniki. Stabilnost modela se kaže z zmožnostjo posploševanja na novih podatkovnih točkah in ne le na zgodovinskem naboru podatkov, na katerih je bil usposobljen (Phan, Hall & Whitson, 2016). Model in njegovi rezultati morajo biti dovolj intuitivni in razumljivi za uporabnika. Identificirali smo tri glavne izzive: neuravnoteženost porazdelitev, določitev časovnega okvira za merjenje verjetnosti neplačila ter zagotovitev razumljivosti povezav med spremenljivkami in rezultati.

1.2.1 Neuravnotežena porazdelitev razredov – problem pristranskosti

Eden glavnih izzivov pri razvoju modelov kreditnega ocenjevanja je **neuravnotežena porazdelitev razredov**. Ta se zgodi, ko število razreda dobrih posojilojemalcev na splošno močno presega število razreda neplačnikov. Pojav neravnovesja v razredih lahko najbolj vpliva na uspešnost klasičnih klasifikacijskih tehnik, ker domnevajo, da je porazdelitev razredov razmeroma uravnotežena in da ima imata razreda enake stroške napačnega razvrščanja. V realnosti pa razred neplačnikov pogosto predstavlja manj kot deset odstotkov celotne podatkovne zbirke (Garcia & Marques, 2012).

Poleg očitnega učinka, da se poveča negotovost ocenjevanja, pristranska vzorčna porazdelitev vodi do podcenjevanja resnične verjetnosti dogodka neplačila (Orth, 2011). V zadnjem desetletju je ta problem pritegnil zelo veliko pozornosti, tako na področju odkrivanja goljufivih finančnih dejavnosti kot tudi za napovedovanje kreditne sposobnosti novih prosilcev. Na področju kreditnega ocenjevanja se raziskave osredotočajo predvsem na analizo obnašanja običajnih modelov napovedovanja, ki kažejo, da se uspešnost napovedi manjšinskega razreda znatno zmanjša, ko se razmerje neravnovesja poveča (Garcia & Marques, 2012).

1.2.2 Izbira obravnavanega časovnega obdobja – problem filozofije merjenja verjetnosti neplačila

Bančni sistemi kreditnega ocenjevanja merijo verjetnost dogodka neplačila dolžnika v določenem časovnem obdobju. V praksi se modeli razlikujejo glede na obravnavano časovno obdobje. V tem kontekstu poznamo dve ocenjevalni filozofiji, in sicer (Mayer, Resch & Sauer, 2017):

1. **Sistemi točk v času** (angl. point in time probability of default, v nadaljevanju **PIT PD**): trenutna verjetnost neplačila, ki odraža vse trenutno razpoložljive informacije.
2. **Sistemi skozi cikel** (angl. through the cycle probability of default, v nadaljevanju **TTC PD**): dolgoročna verjetnost neplačil, kjer so ocene neodvisne od cikličnih sprememb makroekonomskih razmer.

Sistemi PIT PD merijo tveganje neplačila v kratkem obdobju, pogosto za eno leto ali manj. Temeljijo na vseh v trenutku merjenja razpoložljivih informacijah podjetja in vsebujejo finančne kazalnike podjetja, kot so merila likvidnosti, zadolženosti, denarnega toka, dobičkonosnosti itd. Zajemajo specifične učinke podjetja in odražajo tveganje podjetij (Ortl, 2016).

Sistemi TTC PD pa merijo tveganje neplačila v obdobju, ki je dovolj dolgo, da se učinki poslovnega cikla večinoma nevtralizirajo. Z makroekonomskimi spremenljivkami lahko model izboljšamo, ker so številni makroekonomski viri podatkov sodobnejši od finančnih razmerij posojilojemalcev (Ortl, 2016).

Z drugimi besedami, ocene po filozofiji PIT PD poskušajo oceniti trenutni položaj stranke ob upoštevanju cikličnih in trajnih učinkov, ocene po filozofiji TTC PD pa se osredotočajo predvsem na stalni del tveganja neplačila in so skoraj neodvisne od cikličnih sprememb kreditne sposobnosti stranke (Topp & Perl, 2010).

Makroekonomske spremenljivke vplivajo enako na vsa podjetja in s tem kažejo povprečno verjetnost dogodka neplačila, specifične spremenljivke podjetij pa so ključne za razlikovanje med kreditno sposobnostjo posameznih podjetij (Volk, 2012). Pristop TTC PD zagotavlja bolj stabilne in manj ciklične ocene, saj se pri pristopu PIT PD ocene v razburkanem obdobju močno razlikujejo (Ortl, 2016). Uporaba obeh vrst informacij za napoved dogodka neplačila je lahko odveč, saj so makroekonomske spremenljivke in finančna razmerja pogosto zelo povezana. Praksa preslikovanja zunanjih ocen TTC PD v notranje ocene PIT PD z namenom obogatitve ali potrjevanja bančne zbirke podatkov kreditnih ocen lahko vodi k sistematičnemu precenjevanju ali podcenjevanju verjetnosti dogodka neplačila (Topp & Perl, 2010).

1.2.3 Razumljivost delovanja modelov

Še en izziv, ki predstavlja veliko dilemo pri razvoju modelov kreditnega ocenjevanja, je **razumljivost** modelov. Nekateri modeli so lahko manj relevantni za kreditno ocenjevanje, saj odločevalci potrebujejo preprosta in razložljiva pravila za napovedovanje dogodka neplačila, česar pa nekateri modeli ne omogočajo.

Obrazložitev posameznih odločitev je kompleksnejša in ne takoj razvidna, kar pa je obvezno pri številnih panogah zaradi regulativnih zahtev, zakonskih razlogov, zaradi zmožnosti upravljanja s tveganji ali želje razumevanja posameznih odločitev (Molnar, 2020). Obstaja

vedno več zakonov, ki dajejo stranki pravico do obrazložitve odločitve v primeru negativne kreditne odločitve. Primer takega zakona je lani, februarja 2019, sprejela poljska vlada kot neposredno posledico izvajanja splošne uredbe varstva podatkov (angl. General Data Protection Regulation) v Evropski uniji (Klicki & Szymielewicz, 2019).

Razumljivost oziroma interpretativnost modelov predstavlja težavo še posebno pri modelih strojnega učenja, kar bomo podrobno razložili v naslednjem poglavju.

2 PRILOŽNOST UPORABE STROJNEGA UČENJA ZA MODELIRANJE

2.1 Strojno učenje

Strojno učenje je disciplina, ki se ukvarja s samodejnim prilagajanjem in učenjem programov iz podatkov ali preteklih izkušenj, ne da bi bili za to izrecno programirani (Mitchell, 1997). To dosežejo z algoritmi, ki določajo zaporedje navodil, ki vhodne informacije pretvorijo v izhodne. Ti algoritmi se uporabljajo za razlikovanje med pomembnimi in nepomembnimi vzorci v podatkih ter za sprejemanje boljših odločitev v prihodnosti na podlagi ponujenih podatkov (Kennedy, 2013). Glavni cilj strojnega učenja je izdelava in potrjevanje robustnih modelov, ki bodo sposobni obvladovati pomanjkljivosti v podatkih, kot so manjkajoče informacije in neuravnotežena porazdelitev razredov (Sadatrasoul, Gholamian, Siami & Hajimohammadi, 2013). V zadnjih letih je strojno učenje pridobilo široko pozornost in vse večjo priljubljenost za komercialno uporabo (Phan, Hall & Whitson, 2016).

Poznamo dve različni vrsti algoritmov strojnega učenja za odkrivanje vzorcev podatkov, ki vodijo do uporabnih učinkov, in sicer nadzorovano in nenadzorovano učenje (Guru99, brez datuma). Pri nadzorovanem učenju (angl. supervised learning) se algoritmi učijo iz označenih podatkov o usposabljanju in pomagajo predvideti rezultate, pri nenadzorovanem učenju (angl. unsupervised learning) pa se algoritmi ukvarjajo z neoznačenimi podatki in sami odkrivajo nove informacije. Nenadzorovani algoritmi učenja omogočajo izvajanje bolj zapletenih procesov obdelave v primerjavi z nadzorovanim učenjem.

Eden najpomembnejših primerov nadzorovanega učenja je razvrščanje ali **klasifikacija**. Opredelimo jo lahko kot metodo, ki elemente določenega niza podatkov razvrsti v skupine glede na njihove lastnosti. Razvrsti jih na podlagi diskriminantne funkcije, ki jo imenujemo tudi klasifikator ali model. Klasifikacija vključuje katerikoli kontekst, v katerem se na podlagi razpoložljivih informacij sprejme odločitev ali napoved (Keramati & Yousefi, 2011). Večina nadzorovanih algoritmov za klasifikacijo domneva uravnoteženo porazdelitev oznak razredov, s katerimi poskušajo čim bolj povečati skupno natančnost razreda med usposabljanjem (Kennedy, 2013). Najbolj uporabljene tehnike za opravljanje nalog kreditnega ocenjevanja izhajajo iz metode klasifikacije.

2.2 Uporaba strojnega učenja za modeliranje ocenjevalnih modelov

V zadnjih letih se aplikacije in metodologije za algoritmično modeliranje hitro razvijajo. Statistiki so že desetletja uporabljali algoritmično modeliranje. Razvoj algoritmičnih metod se je začel v skupnostih z drugih področij izven statistike. Do takrat je bil pri razvijanju modelov poudarek na podatkovnih modelih, te skupnosti pa so se osredotočile na same lastnosti algoritmov (Breiman, 2001), pri katerih je bil glavni kriterij napovedna natančnost. Tako sta se sredi osemdesetih let razvili dve novi tehniki na osnovi strojnega učenja, in sicer nevronske mreže in odločitvena drevesa (Breiman, 2001). Pri teh modelih je bistvo pridobiti koristne informacije o razmerju med odvisnimi in neodvisnimi spremenljivkami.

V zadnjih letih se je pojavil ponovni interes za reševanje poslovnih problemov z uporabo novih tehnik modeliranja, ki so se razvile na podlagi strojnega učenja in masovnih podatkov (angl. big data) (Sadatrasoul, Gholamian, Siami & Hajimohammadi, 2013). Čeprav so za te namene statistične tehnike že desetletja sprejete kot najprimernejša metoda, so te novejše tehnike vse bolj priljubljene zaradi učinkovitosti, natančnosti in sorazmerne preprostosti.

2.2.1 Uporaba strojnega učenja za kreditno ocenjevanje

Na temo primerjave statističnih tehnik s tehnikami strojnega učenja za kreditno ocenjevanje obstaja že ogromno raziskav. Te kažejo, da se pri nelinearni klasifikaciji vzorcev, pod katere se šteje tudi kreditno ocenjevanje, tehnike strojnega učenja lahko uporabljajo kot alternativne metode in da v veliko primerih vodijo do boljših rezultatov v primerjavi s tradicionalnimi statističnimi metodami (Tsai & Wu, 2008).

Za izdelavo modelov kreditnega ocenjevanja je kot standardna tehnika sprejeta logistična regresija in v manjši meri linearna diskriminantna analiza (Lyn, 2000). Težava pri uporabi teh statističnih metod za kreditno ocenjevanje je na primer ta, da predpostavljajo, da so vrednosti neodvisnih spremenljivk normalno porazdeljene, kar pa ne velja v skoraj nobenem realnem primeru (Wang, Hao, Ma & Jiang, 2011). Kljub temu da se dotičnemu problemu lahko izognemo z določenimi metodami priprave podatkov, to še vedno negativno vpliva na končni rezultat modelov.

V nasprotju s statističnimi metodami, metode strojnega učenja ne predvidevajo določene distribucije podatkov. Tehnike strojnega učenja samodejno pridobijo znanje iz vzorcev za usposabljanje (Koh, Tan & Goh, 2006). Vendar pa na splošno ni najboljše tehnike za oblikovanju modelov kreditnega ocenjevanja.

Rezultati empiričnih aplikacij so mešani (Sadatrasoul, Gholamian, Siami & Hajimohammadi, 2013). V pregledu raziskav na to temo smo opazili, da za kreditno ocenjevanje v veliko primerih najboljšo napovedno moč dosegajo nevronske mreže in metode podpornih vektorjev (Tsai & Wu, 2008). Katera vrsta modela je najboljša, je odvisno

od podrobnosti problema, strukture podatkov, zmožnosti ločevanja med razredi, uporabljenih lastnosti in cilja klasifikacije (Wang, Hao, Ma & Jiang, 2011).

2.2.2 Ansambelsko učenje

Čeprav v literaturi ni konsistentnih zaključkov o tem, katera tehnika je najboljša, smo opazili, da se pri novejših študijah veliko govori o izboljšanju obstoječih tehnik strojnega učenja s kombiniranjem več klasifikatorjev ali z t. i. ansambelskim učenjem (angl. ensamble learning) (Hung & Chen, 2009).

To je primer strojnega učenja, kjer se več različnih modelov usposablja za reševanje istega problema. Ansambelske metode poskušajo sestaviti nabor hipotez in jih kombinirati za reševanje danega problema. Sposobnost posploševanja ansambelskih metod je ponavadi veliko močnejša od sposobnosti posameznih modelov (Wang, Hao, Ma & Jiang, 2011).

2.2.3 Razumljivost modelov strojnega učenja

Večina algoritmov strojnega učenja daje rezultate tipa črne škatle (angl. black box), kar pomeni, da procesa ne moremo natančno razložiti in vidimo le vhodne in izhodne informacije (Phan, Hall & Whitson, 2016). Algoritmi strojnega učenja imajo velik potencial za izboljšanje izdelkov, procesov in raziskav, saj v primerjavi s statističnimi modeli upoštevajo večje število implicitnih interakcij spremenljivk. Vendar večja natančnost skoraj vedno pomeni manjšo **interpretativnost**. Enostavnejši modeli, kot so linearna regresija in odločitvena drevesa, na drugi strani zagotavljajo manj napovedne zmogljivosti, niso pa vedno sposobni obravnavati vseh kompleksnosti nabora podatkov (Hulstaert, 2019).

Obstaja več različnih tehnik, ki jih lahko uporabimo za izboljšanje interpretativnosti modelov strojnega učenja. Že leta se razvijajo razne verodostojne tehnike za treniranje interpretativnih modelov in pridobivanje vpogleda v vedenje in mehanizme modelov. Ena od teh so nadomestni modeli (angl. surrogate models), ki lahko zagotovijo vpogled tako v napovedi modela kot v njegove napake (Gill & Hall, 2019). Za razliko od navadnih modelov so ti usposabljani z uporabo napovedi rezultatov drugega, bolj kompleksnega modela. Nadomestni modeli pa niso sposobni popolnoma predstavljati funkcije osnovnega modela, niti niso sposobni zajeti zapletenih odnosov med spremenljivkami (Hulstaert, 2019). Služijo predvsem kot povzetek modela, vendar se nanje ne sme izključno zanašati.

Še eno orodje, ki se lahko uporabi za razlaganje modelov, je lastnost pomembnosti spremenljivk. Ta je na voljo za nekatere modele strojnega učenja, kot so nevronske mreže, naključni gozdovi in ansambelske metode, ki uporabljajo zviševanje gradientov (Hall, 2016).

2.3 Metodologija CRISP-DM

Standardizirana metodologija za izvedbo procesa podatkovnega rudarjenja (angl. cross-industry process for data mining, v nadaljevanju CRISP-DM) je metodologija, ki zagotavlja strukturiran pristop k načrtovanju, organiziranju in izvedbi projektov podatkovnega rudarjenja po vnaprej definiranih stopnjah ali fazah.

Definira okvir (angl. framework), ki omogoča iteracije skozi vse te stopnje, kar omogoča nenehno izboljševanje projekta, s tem da se lahko vrnemo na prejšnje stopnje in ponovno izvedemo določena dejanja (May, 2017).

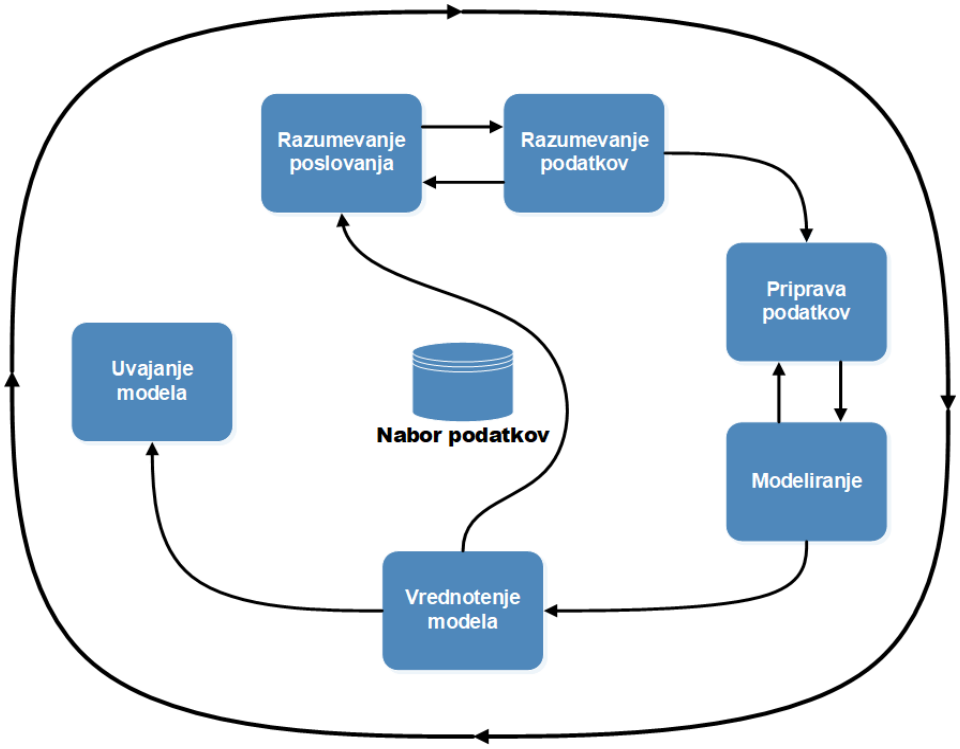
Metodologija je neodvisna od industrijskega sektorja, kjer jo želimo uporabiti, in od tehnologij, s katerimi želimo te projekte izvesti. Cilj te metodologije je, da pomaga narediti velike projekte podatkovnega rudarjenja cenejše, hitrejše, zanesljivejše in bolj obvladljive za vodenje. Sestavljena je iz šestih glavnih stopenj, in sicer (Wirth, 2000):

1. **Razumevanje poslovanja** (angl. business understanding): na začetku je pomembno, da razumemo kaj želimo doseči s poslovnega vidika, definiramo zahteve in vse to opredelimo kot projektni načrt, namenjen doseganju ciljev projekta podatkovnega rudarjenja. Namen te stopnje je, da odkrijemo vse pomembne dejavnike, ki lahko vplivajo na rezultat projekta.
2. **Razumevanje podatkov** (angl. data understanding): stopnja se začne z zbiranjem podatkov, nadaljuje pa s postopki za njihovo razumevanje in odkrivanje problemov.
3. **Priprava podatkov** (angl. data preparation): odločimo se, katere podatke bomo uporabili v analizi. Stopnja zajema vse dejavnosti za izgradnjo končnega nabora podatkov, ki bodo uporabljeni za izdelavo modelov. Postopki za pripravo podatkov so velikokrat izvedeni večkrat in nimajo določenega vrstnega reda.
4. **Modeliranje** (angl. modeling): na tej stopnji uporabimo različne tehnike modeliranja in kalibriramo njihove parametre, da dobimo čim boljše rezultate.
5. **Vrednotenje** (angl. evaluation): sledi pregled in ocenjevanje rezultatov z uporabo meril za poslovno uspešnost, ki smo jih definirali na začetku projekta.
6. **Uvajanje modela** (angl. deployment): izdelani modeli se uvedejo, vzdržujejo in spremljajo. Pred tem pa je pomembno, da izdelamo načrt uvajanja in ponovno preverimo če je potrebno katero od stopenj ponoviti.

Zaporedje stopenj ni pomembno in v večini projektov se po potrebi stalno premika med njimi. CRISP-DM model je zelo prilagodljiv in omogoča, da ustvarimo model rudarjenja podatkov, ki ustreza našim potrebam.

Model poteka po stopnjah metodologije in njihovo povezovanje je prikazano na sliki 1. S puščicami so prikazane najbolj pogoste odvisnosti med stopnjami.

Slika 1: Model CRISP-DM



Prirejeno po Salcedo & McCormick (2017).

2.4 Koraki priprave podatkov

Priprava podatkov je ključni korak za ustvarjanje dobrih modelov, ki dajejo smiselne rezultate (napovedi). Zaradi njene pomembnosti bomo v nadaljevanju tega poglavja obširno opredelili metode, ki smo jih uporabili za pripravo podatkov. Koraki so sledeči:

1. obravnavanje manjkajočih vrednosti,
2. odstranjevanje osamelcev,
3. normalizacija kazalnikov,
4. zmanjševanje dimenzionalnosti podatkovne zbirke,
5. ponovno vzorčenje in
6. prečno preverjanje podatkov.

2.4.1 Obravnavanje manjkajočih vrednosti

V podatkovni zbirki imamo manjkajoče vrednosti, kadar je v njenih spremenljivkah ena ali več vrednosti prazna oziroma nima shranjene nobene vrednosti podatkov. Večina postopkov modeliranja je občutljiva na manjkajoče vrednosti. Če uporabimo podatkovne zbirke z manjkajočimi podatki, bodo rezultati bistveno slabši, nekateri modeli pa sploh ne bodo vrnili rezultatov. Optimalne metode za obravnavo manjkajočih vrednosti ni, saj ima vsaka metoda določene slabosti.

Manjkajoče podatke lahko razvrstimo v tri kategorije, in sicer (Mack, Su & Westreich, 2018):

- **Podatki manjkajo popolnoma naključno** (angl. missing completely at random): predpostavlja, da so manjkajoče vrednosti neodvisne od ostalih podatkov.
- **Podatki manjkajo naključno** (angl. missing at random): predpostavlja, da lahko manjkajoče vrednosti predvidimo na podlagi ostalih podatkov.
- **Podatki manjkajo nenaključno** (angl. missing not at random): predpostavlja, da je razlog za manjkajoče vrednosti odvisen od samih neopaženih podatkov.

2.4.2 Odstranjevanje osamelcev

Osamelci (angl. outlier) so vzorci podatkov, ki imajo bistveno različne značilnosti od ostalih vzorcev. Zaznavanje in obravnava teh vrednosti ima lahko velik vpliv na uspešnost modelov. Hkrati pa lahko ravno te vrednosti vsebujejo pomembne vzorce, ki modelu pomagajo zaznati želeno vrednost, ki je v našem primeru dogodek neplačila. Kljub temu, da lahko z odstranjevanjem teh vrednosti izboljšamo rezultate modelov, moramo biti previdni, saj lahko odstranimo informacije o spremenljivosti (angl. variability), ki je značilna za področje raziskave (Frost, brez datuma).

2.4.3 Normalizacija spremenljivk

Normalizacija (angl. normalization) je tehnika priprave podatkov, ki spremeni obseg vrednosti podatkov v vrednosti med 0 in 1. Standardizacija (angl. standardization) pa spremeni obseg vrednosti podatkov tako, da imajo srednjo vrednost 0 in standardni odklon 1. Normalizacijo se uporablja, ko imajo podatki različne obsege v spremenljivkah in jih uporabljamo za algoritme, ki ne upoštevajo porazdelitve podatkov. To so večinoma vsi algoritmi, ki uporabljajo strojno učenje. Standardizacija pa se uporablja za algoritme, ki predpostavljajo, da so podatki standardno porazdeljeni, kot sta na primer linearna in logistična regresija (Lakshmanan, 2019).

2.4.4 Zmanjševanje dimenzionalnosti podatkovne zbirke

Pri strojnem učenju je pogost problem, da imamo preveč dejavnikov, na podlagi katerih izvedemo klasifikacijo. Večje kot je število dejavnikov, težje je vizualizirati podatke in iz njih narediti uporabne modele. Eden od teh dejavnikov so spremenljivke v podatkovni zbirki, ki jih imenujemo lastnosti. Velikokrat so lastnosti povezane in zelo podobne, iz česar sledi, da so pri modeliranju odveč. Da odpravimo ta problem, uporabimo tehniko zmanjševanja dimenzionalnosti podatkovne zbirke.

Zmanjševanje dimenzionalnosti (angl. dimensionality reduction) podatkovne zbirke je postopek zmanjšanja števila obravnavanih naključnih spremenljivk s ciljem pridobivanja

nabora najbolj pomembnih spremenljivk (Van Der Maaten, Postma & Van den Herik, 2009). Zmanjšana podatkovna zbirka mora čim bolje predstavljati dimenzionalnost podatkov originalnega nabora spremenljivk.

Poznamo dve vrsti zmanjšanja dimenzionalnosti, in sicer:

1. **Izbor lastnosti** (angl. feature selection): proces izbire najbolj relevantnih in izločanje manj pomembnih spremenljivk. S tem načinom ohranimo razlago ohranjenih spremenljivk, hkrati pa izgubimo vse prednosti, ki bi jih lahko prinesle izpuščene spremenljivke.
2. **Ekstrakcija lastnosti** (angl. feature extraction): proces združevanja spremenljivk v nove spremenljivke, ki še vedno natančno opisujejo prvotni nabor podatkov. Začetni nabor spremenljivk se zmanjša na bolj obvladljive skupine za obdelavo.

V literaturi smo zasledili trditev, da metode ekstrakcij lastnosti ne morejo ustrezno ravnati s kompleksnimi nelinearnimi podatki (Van Der Maaten, Postma & Van den Herik, 2009).

2.4.5 Ponovno vzorčenje

Za rešitev problema neuravnotežene porazdelitve razredov obstajajo različne metode vzorčenja, ki preuredijo podatke v uravnoteženo porazdelitev. Neuravnotežena klasifikacija predstavlja izziv za modeliranje s strojnim učenjem, saj je večina algoritmov strojnega učenja, uporabljenih za klasifikacijo, zasnovana na predpostavki, da je v vsakem razredu enako število primerov (Brownlee, 2020).

Z neuravnoteženim naborom podatkov algoritmi ne dobijo potrebnih informacij o manjšinskem razredu in ne morejo narediti natančne napovedi, saj so pristranski do večinskega razreda.

Z metodami ponovnega vzorčenja spremenimo izvirni nabor podatkov tako, da v novem naboru zagotovimo enak delež razredov odvisne spremenljivke. Nekatere od teh metod so:

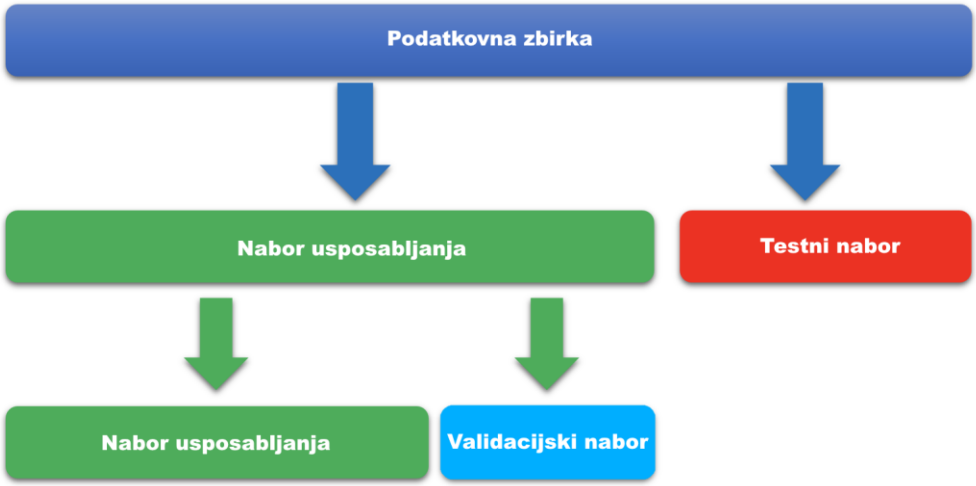
1. **Podvzorčenje** (angl. undersampling): v naboru podatkov zmanjša število opazovanj večinskega razreda tako, da se ujema z manjšinskim. Slabost te metode je, da z odstranjevanjem opazovanj izgubimo pomembne informacije, ki se nanašajo na večinski razred.
2. **Prevzorčenje** (angl. oversampling): replicira opažanja manjšinskega razreda, da uravnoteži nabor podatkov. Prednost te metode je, da z njo ne izgubimo informacij. Slabost te metode pa je, da s tem ko naboru podatkov doda podvojena opažanja, povzroči prenasičenost (angl. overfitting) podatkov.
3. **Sintetično generiranje podatkov** (angl. synthetic data generation): neuravnoteženost podatkov reši tako, da z različnimi metodami ustvari nove podatke in s tem ustvari ravnovesje med razredi.

Še en problem, ki se lahko pojavi pri vzorčenju, je negotovost parametrov. Negotovost parametrov se zgodi, ko ne vemo natančnih ali najboljših vrednosti v populaciji. V tem primeru lahko z vzorčenjem le dobro ugibamo. Prisotna je le v majhnih podatkovnih zbirkah (Liu, 2011). V našem primeru bi bila negotovost parametrov problem, če bi imela ena spremenljivka pod 200 vrednosti.

2.4.6 Prečno preverjanje podatkov

Prečno preverjanje podatkov (angl. cross validation) je metoda preverjanja pravilnosti podatkov, ki rezervira določen del podatkovne zbirke, na katerem ne usposablja modelov. Na koncu modeliranja se na tem rezerviranem delu preveri napovedna moč modela – imenujemo ga testni nabor podatkov (angl. testing set). (Ray, 2018). Razdelitev podatkovne zbirke je prikazana na sliki 2.

Slika 2: Razdelitev podatkovne zbirke



Prirejeno po Brownlee (2020).

Ostali podatki se uporabljajo za usposabljanje modela in ga imenujemo podatkovni nabor usposabljanja (angl. training set). Testni nabor podatkov se uporablja, da bolje razumemo vhodne podatke, za oceno različnih modelov in za nastavitev parametrov izbranega modela. Uporabi se ga tudi za usposabljanje končnega modela z vsemi razpoložljivimi podatki, s katerim bi napovedovali nove primere (Brownlee, 2020). Manjši del testnega nabora podatkov se rezervira za validacijo. Ta del se imenuje validacijski nabor podatkov (angl. validation set) in se ga uporablja za nepristransko oceno modela, ki se prilega na podatkovni nabor usposabljanja med nastavljanjem parametrov modela.

S prečnim preverjanjem podatkov se izognemo problemu prenasičenosti (angl. overfitting), ki se zgodi, če prečno preverjanje izvedemo napačno. Prenasičenost pomeni, da izdelani model deluje zelo dobro na podatkovnem naboru usposabljanja, hkrati pa zelo slabo na testnem naboru (Drakos, 2018).

3 PREDSTAVITEV UPORABLJENIH TEHNIK

Glavna tehnika modeliranja, ki jo bomo primerjali z ostalimi, je logistična regresija, ki se pogosto uporablja pri kreditnem ocenjevanju in ker vodi do rezultatov, ki so enostavni za interpretacijo vsem deležnikom procesa. V primerjalni analizi širokega nabora klasifikacijskih algoritmov z nabori podatkov, namenjenim kreditnem ocenjevanju, je bilo ugotovljeno, da v povprečju logistična regresija deluje dobro (Lessmann, Baesens, Seow & Thomas, 2015). Na temo primerjanja različnih metodologij za kreditno ocenjevanje je bilo izvedenih že zelo veliko raziskav. Zanimanje za izboljšanje rezultatov se je preselilo z logistične regresije na modernejše napredne tehnike. Ugotovitve kažejo, da imajo lahko modeli, kot so naključni gozdovi in nevronske mreže, boljšo napovedno moč kot logistična regresija. V novejših raziskavah se za metodološko naprednejše šteje modele, ki imajo še boljšo napovedno moč kot naključni gozdovi (Lessmann, Baesens, Seow & Thomas, 2015).

Zaradi navedenih razlogov bomo za razvoj modelov kreditnega ocenjevanja uporabili na področju strojnega učenja najpopularnejše tehnike in sicer: K-najbližjih sosedov, odločitvena drevesa, naključni gozdovi, nevronske mreže in metode podpornih vektorjev. Na koncu bomo uporabili še tehniko zviševanja gradientov (angl. gradient boosting), ki se je v zadnjih letih še posebej izkazala za izrazito učinkovito in zmožno točnih napovedi (Petropoulos, Siakoulis, Stavroulakis & Klamargias, 2018).

3.1 Logistična regresija

Logistična regresija (angl. logistic regression) je oblika linearne regresije (Keramati & Yousefi, 2011). Čeprav ima v imenu besedo regresija, je to algoritem, ki se uporablja za reševanje problemov klasifikacije. Modeliranje z logistično regresijo se pogosto uporablja za analizo multivariatnih podatkov in je ena ključnih tehnik za računanje diskriminantne funkcije med dvema razredoma, oziroma pri binarni klasifikaciji, kar je značilnost modeliranja kreditnega ocenjevanja (Sarlija, Benšić & Bohacek, 2006).

Teoretično je dokazano, da v večini primerov kaže boljše posploševalno vedenje kot regresija z metodo najmanjših kvadratov in je bolj odporna na odstopanja kot multivariatna Gaussova porazdelitev (Phan, Hall & Whitson, 2016). Linearna logistična regresija je postala standardna tehnika modeliranja, s katero pridobimo enostavno berljiv model, vendar je omejena na domnevo linearnega razmerja med finančnimi razmerji in diskriminatorno oceno. Kljub temu ne zahteva normalno porazdeljenih spremenljivk za napovedovanje, kar je velika prednost te tehnike.

Razlog, da se logistična regresija še vedno tako pogosto uporablja za kreditno ocenjevanje je ta, da lahko pri njej razmeroma enostavno izvajamo preverjanje rezultatov (Lund, 2015). Ker je postopek izdelave zelo pregleden, lahko modeli zlahka izpolnjujejo vse regulativne zahteve (Salvaire, 2019). Možno je na primer preveriti koeficient posamezne spremenljivke in preučiti smiselnost rezultata ter tako zlahka odkriti težave.

3.2 K-najbližjih sosedov

K-najbližjih sosedov (angl. K-nearest neighbor, v nadaljevanju KNN) je klasifikacijska tehnika, ki temelji na učenju po podobnosti (Keramati & Yousefi, 2011). Ta tehnika spada v kategorijo neparametričnih metod razvrščanja, kar pomeni, da nima nobenih predpostavk o osnovni distribuciji podatkov, kar je zelo koristno, saj v resničnem svetu večina podatkov ne upošteva značilnih teoretičnih predpostavk.

KNN običajno temelji na evklidski razdalji med naborom testiranja in določenimi vzorci nabora usposabljanja. Algoritem KNN deluje tako, da kadarkoli potrebuje napoved nove točke, se iz podatkov nabora usposabljanja izberejo njeni najbližji sosedje. Napoved nove točke je povprečje vrednosti njegovih »k« najbližjih sosedov (Mukid, Widiharih, Prahutama & Rusgiyono, 2017). Izbira vrednosti spremenljivke »k« je zelo pomembna, saj ta določa kompromis med varianco in pristranskostjo ocene. Nižja vrednost »k« za napoved povzroči nizko pristranskost in visoko varianco, kar pomeni, da lahko model pri usposabljanju kaže zelo dobre rezultate, pri testiranju pa bistveno slabše. Po drugi strani pa višja vrednost spremenljivke »k« lahko zmanjša varianco napovedi, hkrati pa povzroči visoko pristranskost, kar pomeni, da je velika možnost, da vrne napačno napoved (Bagherpour, 2017). Ob primerni izbiri vrednosti te spremenljivke in pri nelinearni meji odločitve lahko pričakujemo boljše rezultate modela v primerjavi z logistično regresijo (Abdelmoula, 2015).

Tehnika KNN se zaradi preprostosti interpretiranja rezultatov zelo pogosto uporablja za reševanje težav s klasifikacijo, kot je kreditno ocenjevanje (Mukid, Widiharih, Prahutama & Rusgiyono, 2017). Poleg tega je eden najpreprostejših algoritmov za razvrščanje in se zaradi tega pogosto uporablja kot merilo za primerjavo z bolj zapletenimi klasifikatorji.

3.3 Odločitvena drevesa

Odločitvena drevesa (angl. decision trees) so prikaz odločitev in njihovih možnih posledic, hkrati pa tehnika za napovedno regresijo in klasifikacijo. Drevesa so modeli, ki so sestavljeni iz niza pogojev za razvrščanje primerov v dve ali več različnih skupin.

V drevesnih strukturah so listna vozlišča, ki predstavljajo posamezne klasifikacije. V začetnem, zgornjem delu drevesa je glavno vozlišče, iz katerega izhajajo notranja vozlišča, pri katerih se opravi test posameznega atributa ali vhodne spremenljivke. Vsaka veja, ki sledi glavnemu vozlišču, vodi do rezultata testa. Notranja vozlišča predstavljajo trenutne atribute napovedovanj, veje pa predstavljajo povezave atributov (spremenljivk), ki vodijo do končnih klasifikacij. (Zhang, Zhou, Leung & Zheng, 2010). Vsaka t. i. korenina v drevesu predstavlja eno odločitveno pravilo (Šmid, 2002). Nova opazovanja se obdelujejo po drevesu v skladu s pravili odločitve, dokler ni doseženo končno vozlišče, ki nato predstavlja klasifikacijo tega opažanja (Engelmann & Rauhmeier, 2011). Odločitvena drevesa se uporabljajo za klasifikacijo, v kolikor je odvisna spremenljivka kvantitativno diskretna ali kvalitativna (Keramati & Yousefi, 2011).

Ta pristop je postal zelo priljubljena tehnika za razvoj modelov kreditnega ocenjevanja, saj so odločitvena drevesa preprosta za interpretacijo in jih je mogoče enostavno vizualizirati. Slabost te tehnike je, da ne dobimo linearne funkcije ocene, kot jo dobimo pri logistični regresiji, kar odločevalci in regulatorji pogosto pričakujejo.

3.4 Naključni gozdovi

Naključni gozdovi (angl. random forests) so tehnika strojnega učenja, ki se lahko uporablja tako za regresijo kot tudi za klasifikacijo. Spada pod ansambelske metode (angl. ensemble method), kar pomeni, da združuje skupino šibkih modelov in tvori en sam močan model. Tehnika združuje sklop posameznih klasifikacijskih in regresijskih dreves ter tako tvori gozd (Mageto, Mwalili & Waititu, 2015). Za razvrstitev novega predmeta na podlagi atributov vsako drevo poda klasifikacijo. Gozd izbere klasifikacijo, ki ima največ glasov in v primeru regresije vzame povprečje izhodnih vrednosti različnih dreves (Lateef, 2019). Odločitve o delitvi, sprejete na začetku algoritma, določajo pravila delitve v nastajajočih vozliščih. Napovedi posameznih dreves lahko torej kažejo veliko variabilnost, hkrati pa so občutljivi na majhne spremembe pri usposabljanju modela. Za odpravo te težave metode za napovedovanje uporabljajo povprečje skupine dreves. Zaradi zmanjšanja odstopanj napovedi je mogoče natančnost modela izboljšati (Kraus, 2014).

Naključni gozdovi so odlična alternativa tradicionalnim tehnikam kreditnega ocenjevanja, saj ponujajo boljši vpogled v interakcije spremenljivk (Sharma, 2009). Ena največjih prednosti naključnih gozdov je možnost razvrstitve po pomembnosti, ki oceni napovedno vrednost posameznih spremenljivk tako, da spreminja spremenljivko in opazuje, kako to vpliva na uspešnost modela (Breiman, 1999). Poleg tega, da je tehnika preprosta za uporabo, je na splošno prepoznana po natančnosti in zmožnosti soočanja z majhnimi velikostmi vzorcev (Roy, 2016). Algoritem naključnih gozdov deluje dobro tudi brez predhodne obdelave podatkov, zlasti pripisovanja manjkajočih vrednosti (Grennepois, 2018). Naključni gozdovi ne povzročijo prenasičenosti podatkov, saj so zgrajeni iz vzorčnih testov za vsak podmodel (Hué, Hurlin & Tokpavi, 2017). Kot veliko tehnik strojnega učenja pa so naključni gozdovi črna škatla.

3.5 Nevronske mreže

Nevronske mreže (angl. neural networks) so algoritemski postopek za pretvorbo vhodnih informacij v želene izhodne z uporabo močno povezanih omrežij, ki so sestavljena iz sorazmerno preprostih procesnih elementov (Hsieh, Lee & Lee, 2010). So skupek tehnik, namenjenih reševanju številnih različnih problemov, od klasifikacije do modeliranja in optimizacije. Vsak posamezen sistem nevronske mreže sestavlja veliko število medsebojno povezanih in vplivajočih procesnih enot, ki temeljijo na nevrobioloških modelih (Šušteršič, Mramor & Zupan, 2009). Sestavljeni so iz številnih vozlišč, ki pošljejo izhodne informacije če dobijo določeno vhodno informacijo iz drugih vozlišč, na katera so povezana. Končno

omrežje dobimo s prilagoditvijo povezav med vhodnimi in izhodnimi informacijami ter morebitnimi vmesnimi vozlišči (Engelmann & Rauhmeier, 2011). Plast, ki sprejema vhodne informacije, se imenuje vhodna ali prva plast. Končna plast, ki zagotavlja ciljni izhodni signal ali odgovor, se imenuje izhodna plast. Vse plasti med tema dvema slojema se imenujejo skrite plasti (Šušteršič, Mramor & Zupan, 2009). Vedno pogosteje se uporabljajo za modeliranje procesov zaradi pripadajočih spominskih značilnosti in zmožnosti posploševanja (Hsieh, Lee & Lee, 2010). Njihova glavna prednost je, da se hitro prilagodijo na nove informacije, slabost pa, da so tudi nevronske mreže črna škatla.

Pri kreditnem ocenjevanju nevronske mreže dajejo na splošno boljše rezultate od logistične regresije. V primeru, da je velikost vzorcev majhna, lahko logistična regresija ustvari bolj uspešne modele zaradi manjšega števila parametrov, ki zahtevajo oceno (Nurlybayeva & Balakayeva, 2013). Nevronske mreže imajo prednost, ko je nabor podatkov zelo velik, kadar ni korelacije med vhodnimi informacijami in odvisno spremenljivko ali kadar je ekonomska razlaga končnega modela manj pomembna (Tsai & Wu, 2008).

3.6 Metode podpornih vektorjev

Metoda podpornih vektorjev (angl. support vector machine, v nadaljevanju SVM) je tehnika za reševanje problema binarne klasifikacije, priljubljena zaradi dobre zmožnosti posploševanja (Hsieh, Lee & Lee, 2010). SMV preslika vhodne spremenljivke v prostor z več dimenzijami s pomočjo nelinearnega preslikovanja. Linearno ločljiva opazovanja ločimo s hiperravnino (angl. hyperplane) in izberemo tisto hiperravnino, ki maksimira razdaljo od opazovanj (Haltuf, 2014). Podporni vektorji skušajo najti optimalno ločevalno hiperravnino med razredi tako, da maksimirajo prostor med njimi. V tem prostoru se točke, ki ležijo na mejah, imenujejo podporni vektorji, sredina prostora pa optimalna ločitvena hiperravnina. Ta lastnost maksimiranja podpornih vektorjev je razlog za klasifikacijsko učinkovitost te tehnike (Harris, 2015).

Je zelo popularna tehnika za kreditno ocenjevanje, pri praktični uporabi pa jo omejuje dejstvo, da nima možnosti pojasnjevanja rezultatov (Gestel in drugi, 2005). Pri tehniki SVM razmerja med neodvisno in odvisno spremenljivko ni mogoče neposredno razložiti (Han, Han & Zhao, 2013).

Zaradi učinkovitosti in popularnosti te tehnike za klasifikacijo se v zadnjih letih vedno bolj nadgrajuje (Goh, 2019). Pogosto se uporabljajo hibridni SVM modeli, kjer se poleg SVM algoritma uporablja še dodatna tehnika za pomoč pri klasifikaciji. V literaturi smo zasledili, da so enorazredne metode klasifikacije (angl. one-class classification) bolj primerne za kreditno ocenjevanje zaradi tega, ker se v zasnovi izognejo težavi neuravnotežene porazdelitve razredov. Zato smo se odločili poskusiti tudi **enorazredno metodo podpornih vektorjev** (angl. one-class support vector machines), ki temelji na enorazredni klasifikaciji. To je metodologija, ki temelji na enem samem razredu primerov za prepoznavanje običajnega ali pričakovanega vedenja ciljnega razreda (Kennedy, 2013). Enorazredna

metoda podpornih vektorjev je poseben primer SVM, kjer se model najprej usposobi s podatki in nato, ko vidi novo podatkovno točko, lahko ugotovi, ali je ta dovolj blizu podatkom, s katerimi je bil model usposabljan. V raziskavah se je izkazala kot bolj primerna in natančna tehnika v primerjavi s klasično metodo podpornih vektorjev (Hwang, 2018).

3.7 LightGBM

LightGBM (Light Gradient Boosting Machine) je visokozmogljivo ogrodje za zviševanje gradientov (angl. gradient boosting), ki se uporablja kot tehnika za regresijo, klasifikacijo in druge vrste napovedovanja (Morde, 2019). Spada med ansambelske metode. Za model se lahko uporabljajo različne tehnike, kot na primer odločitvena drevesa in naključni gozdovi. Algoritmi povečanja gradientov se začnejo z usposabljanjem odločitvenega drevesa, v katerem je vsakemu opazovanju dodeljena enaka teža (Singh, 2018). Nato dodaja nova drevesa, ki se prilegajo spremenjeni različici izvirnega nabora podatkov. Po oceni prvega drevesa se poveča pomembnost tistih opazovanj, ki jih je težje razvrstiti. Nadaljnja drevesa pomagajo razvrstiti opažanja, ki jih prejšnja drevesa niso dobro razvrstila. Napoved končnega modela je torej tehtana vsota napovedi prejšnjih odločitvenih dreves.

Največja motivacija za uporabo zviševanja gradientov je ta, da omogoča optimizacijo uporabniško določene stroškovne funkcije (angl. cost function) namesto funkcije izgube (angl. loss function) (Singh, 2018). Naprednejše tehnike zviševanja gradientov, kot je XGBoost, so se v zadnjih letih izkazale kot zelo učinkovite in uspešne za reševanje klasifikacijskih problemov, tudi na področju kreditnega ocenjevanja (Cao, He, Chen & Zhang, 2018). Kar loči LightGBM od ostalih tovrstnih tehnik je, da uporablja algoritme na osnovi histograma in izvirno tehniko vzorčenja s filtriranjem podatkovnih primerov za iskanje delitvene vrednosti (Microsoft Corporation, brez datuma). To pospeši usposabljanje modela in minimizira porabo računalniškega spomina. Še ena prednost teh tehnik je, da niso povsem črna škatla, saj ponujajo omejen vpogled v način usposabljanja modela.

4 ANALIZA IZBRANEGA PRIMERA

4.1 Opis primera izbrane banke

Izbrana banka se je odločila, da bo izdelala nov model za kreditno ocenjevanje. Do sedaj je uporabljala model, ki upošteva podatke tudi iz drugih podružnic iz različnih držav, nov model pa bo uporabljal le lokalne podatke. Razvila je nov model, pri čemer je uporabila logistično regresijo. Uporabili so podatkovno zbirko z enim kazalnikom na skupino, podatke pa so pripravili na lasten način. Končni model je bil tehtano povprečje dveh modelov, pri katerem je en uporabljal finančne podatke, drugi pa nefinančne. Večjo utež oziroma pomembnost je imel finančni model.

Zaradi zahteve razumljivosti modela so bili za modeliranje omejeni na tehniki logistične regresije in odločitvenih dreves. Naprednejše modele, ki uporabljajo strojno učenje, bi bili pripravljeni uporabljati za referenco poleg osnovnega modela in za sistem zgodnjega opozarjanja (angl. early warning system), ki aktivno spremlja stanje podjetij, s katerimi poslujejo, in jih opozori, če se njihovo finančno stanje poslabša. Izbrano banko je predvsem zanimalo ali bi za ta namen s tehnikami strojnega učenja dosegli boljše rezultate kot s konvencionalnimi metodami.

4.2 Priprava podatkov in vsebinska interpretacija kazalnikov

Za namen te raziskave nam je izbrana banka zagotovila podatkovno zbirko, ki jo sama uporablja za modeliranje kreditnega ocenjevanja. Zbirka vsebuje anonimizirane podatke o vseh podjetjih, s katerimi banka posluje. Podatkovno zbirko med drugimi sestavljajo naslednje skupine podatkov:

- **Finančni kazalniki**: podatki o poslovanju podjetij, pridobljeni iz Agencije Republike Slovenije za javnopravne evidence in storitve (v nadaljevanju AJPES). Vsebujejo dve skupini po 177 kazalnikov, in sicer ena skupina za podatke tekočega leta (pridobljeni maja 2019) in druga skupina za podatke enega leta nazaj.
- **Izračunani kazalniki**: 38 kazalnikov uspešnosti podjetij, izračunanih na podlagi finančnih kazalnikov.
- **Vrsta podjetja**: opisuje za kakšno vrsto podjetja gre. Med njimi so mala in srednja podjetja, velika podjetja in območni organi.
- **Datum pridobitve**: pove kdaj so bili podatki v tej zbirki pridobljeni.
- **Podatki o dogodku neplačila**: kazalnik, ki pove ali je podjetje v zadnjem letu bilo označeno kot neplačnik.
- **Datum dogodka neplačila.**

Kazalnik s podatki o neplačilu (kazalnik default) se bo pri modeliranju uporabljal kot neodvisna spremenljivka. Ima tri vrednosti, in sicer:

- **0**, če podjetje v zadnjem letu ni bilo označeno kot neplačnik
- **1**, če je podjetje v zadnjem letu bilo označeno kot neplačnik
- **99**, če podjetje ni nikoli bilo označeno kot neplačnik

Celotna podatkovna zbirka, ki smo jo dobili od izbrane banke, je skupaj vsebovala 419 stolpcev in 46327 vrstic. Vsaka vrstica predstavlja informacije za podjetje, ki je imelo v zadnjem letu sklenjen kredit z izbrano banko. Za eno podjetje je v podatkovni zbirki lahko več vrstic, če so s tem podjetjem poslovali več let. Osredotočili smo se na mala in srednja podjetja, ki imajo od dva do petdeset milijonov letnega prometa in tako celotno podatkovno zbirko zmanjšali na 9137 vrstic. 38 izračunanih kazalnikov se bo uporabljalo za modeliranje ocenjevalnih modelov. Kazalniki se za namen strojnega učenja imenujejo lastnosti ali

atributi, v statističnem jeziku pa jih imenujemo neodvisne spremenljivke. Za poenostavljanje modeliranja v kazalniku default smo vse vrednosti 99 spremenili v 0.

Na podlagi teorije, ki smo jo predstavili v podpoglavju 1.2.2, smo se odločili, da podatkovni zbirki ne bomo dodajali makroekonomskih spremenljivk. Iz tega sledi, da bodo naši modeli sledili filozofiji sistemov točk v času (PIT PD).

4.2.1 Pomen posameznih kazalnikov

V podatkovni zbirki imamo 38 finančnih kazalnikov, izračunanih na podlagi AJPES kazalnikov. Glede na finančne značilnosti, ki jih predstavljajo, so razdeljeni na šest skupin:

- **skupina A**, ki predstavlja likvidnost,
- **skupina B**, ki predstavlja upravljanje premoženja in obveznosti,
- **skupina C**, ki predstavlja razmerja med dolgovi in lastnim kapitalom,
- **skupina D**, ki predstavlja pokritost dolgov,
- **skupina E**, ki predstavlja trende poslovanja in
- **skupina F**, ki predstavlja donosnost.

Vse skupine uporabljajo informacije kazalnikov za podatke tekočega leta, razen skupina E, ki upošteva še podatke enega leta nazaj. Imena skupin ter način izračuna posameznega kazalnika so predstavljena v tabeli 1.

Tabela 1: Pomen posameznih kazalnikov

Skupina	Število kazalnika	Način izračuna kazalnika
A	**1**	Obratni kapital / kratkoročna sredstva
	2	Kratkoročni koeficient
	3	Pospešen koeficient
B	**4**	Skupna prometna sredstva
	5	Kratkoročne poslovne terjatve / povprečna mesečna prodaja
	6	Zaloge / povprečna mesečna prodaja
	7	Zaloge + terjatve / povprečna mesečna prodaja
	8	Kratkoročne obveznosti / povprečna mesečna prodaja
	9	Kratkoročne finančne obveznosti / povprečna mesečna prodaja
	10	Kratkoročne operativne obveznosti / povprečna mesečna prodaja
	11	Kratkoročne operativne obveznosti + nastali stroški in odloženi prihodki / povprečna mesečna prodaja
	12	Neto dolg / povprečna mesečna prodaja
	13	Opredmetena osnovna sredstva / skupna sredstva
	14	Trenutna sredstva / skupna sredstva
	15	(Opredmetena osnovna sredstva + dolgoročne naložbe + dolgoročne terjatve) / celotna sredstva

se nadaljuje

Tabela 1: Pomen posameznih kazalnikov

Skupina	Število kazalnika	Način izračuna kazalnika
C	**16**	Lastniški kapital / skupne obveznosti
	17	(Lastniški kapital + dolgoročne rezervacije) / skupna sredstva
	18	Dolgovi / Lastniški kapital
	19	Neto dolgovi / materialna sredstva
	20	(Lastniški kapital + dolgoročne rezervacije + dolgoročni dolgovi) / skupne obveznosti
D	**21**	Odhodki za obresti / dobiček iz poslovanja
	22	Finančni odhodki / dobiček iz poslovanja pred amortizacijo
	23	Dolgovi / dobiček iz poslovanja pred amortizacijo
	24	Dolgovi / bruto denarni tokovi
	25	Dolgoročni dolgovi / bruto denarni tokovi
E	**26**	Indeks povečanja prodaje
	27	Indeks rasti bruto operativnega donosa
	28	Naraščanje dobička iz poslovanja pred amortizacijo
	29	Spremenjen delež stroškov blaga, materiala in storitev v čistem prihodku od prodaje
	30	Trend vzvodov
	31	Trend razmerja med dolgovi in lastnim kapitalom
	32	Trend donosa na sredstva
F	**33**	Stopnja dobička iz poslovanja pred amortizacijo
	34	Dobiček iz poslovanja / čisti dobiček prodaje
	35	Čisti dobiček / čisti dobiček prodaje
	36	Donos na kapital
	37	Donos na sredstva
	38	Dobiček iz poslovanja / skupna sredstva

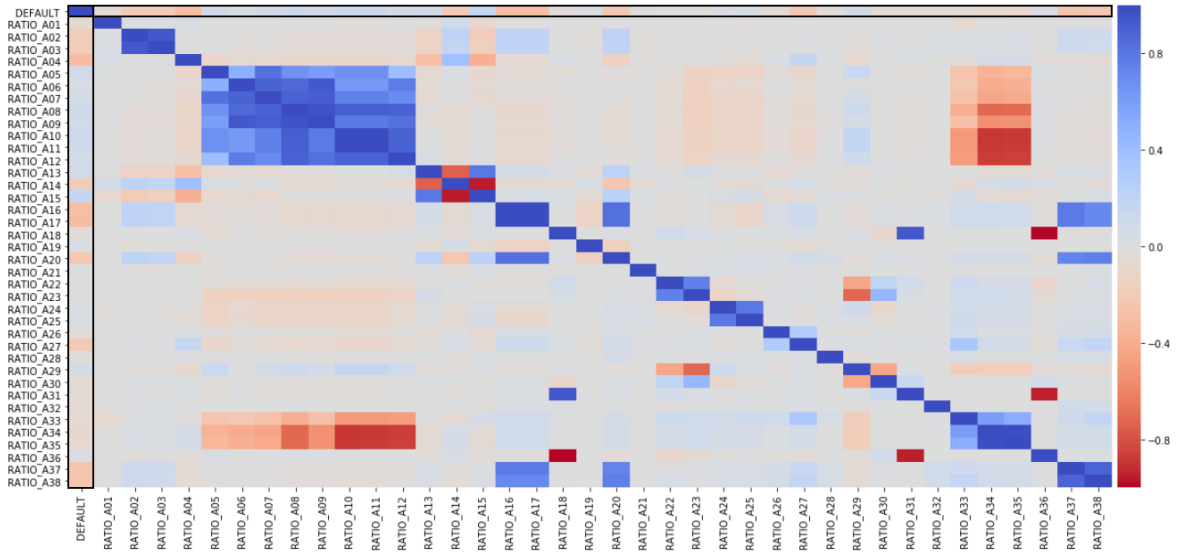
Vir: Lastno delo.

4.2.2 Statistična analiza podatkov

Pred obdelavo (čiščenjem) podatkov smo naredili raziskovalno analizo podatkov (angl. exploratory data analysis). S pomočjo statistične analize in grafičnih predstavitev smo tako odkrili vzorce, nepravilnosti in distribucijo podatkov.

Korelacijo posameznih kazalnikov smo prikazali s korelacijsko matriko (angl. correlation matrix). Vsaka celica v tej matriki prikazuje korelacijske koeficiente med dvema kazalnikoma. Koeficient korelacije se meri na lestvici od -1 do 1, vrednost 0 pa pomeni, da ni povezave med njima (Lu, 2019). Barve v matriki predstavljajo statistično pomembnost. Pozitiven korelacijski koeficient pomeni pozitivno razmerje med obema kazalnikoma, obratno pa velja za negativen koeficient. Glavna diagonala, ki poteka od zgornje leve do spodnje desne strani kaže, da se vsak kazalnik vedno popolnoma ujema sam s seboj. Korelacijska matrika naše osnovne podatkovne zbirke je prikazana na sliki 3.

Slika 3: Korelacijska matrika osnovne podatkovne zbirke



Vir: Lastno delo.

Kazalniki so na matriki označeni kot »Ratio«. Najbolj zanimivi sta vrstici, odebeljeni s črno obrobo, ki predstavljata korelacijo posameznega kazalnika s kazalnikom default. Tukaj vidimo, da večina kazalnikov zelo šibko korelira z našo odvisno spremenljivko (default), veliko tudi negativno. Opazimo lahko tudi, kako so kazalniki porazdeljeni na skupine, kar se kaže s tem, da je več kazalnikov zapored visoko koreliranih. Tak primer je najbolj očiten od 5 do 12. Povečana različica slike je prikazana v prilogi 1.

Statistični povzetek naše podatkovne zbirke je prikazan v tabeli 2.

Tabela 2: Opisna statistika podatkovne zbirke

	Število vrstic	Povprečna vrednost	Standardni odklon	Minimalna vrednost	Maksimalna vrednost
Kazalnik 01	7692	-2.13899	91.40891	-5,466.31915	1.52125
Kazalnik 02	7692	2.48315	29.44158	-448.26691	2,424.00000
Kazalnik 03	7692	1.94458	28.94233	-408.22755	2,424.00000
Kazalnik 04	7700	1.70965	2.92070	0	215.89798
Kazalnik 05	7638	52.50706	3,962.69092	0	346,164.24000
Kazalnik 06	7638	23.55593	1,754.92272	0	153,207.96000
Kazalnik 07	7638	76.06299	5,716.20529	0	499,372.20000
Kazalnik 08	7638	145.37848	11,387.61689	0	994,989.36000
Kazalnik 09	7638	102.49955	8,386.47335	0	732,880.44000
Kazalnik 10	7638	42.87866	3,003.57089	0	262,108.92000
Kazalnik 11	7638	43.05248	3,003.60898	0	262,108.92000

se nadaljuje

Tabela 2: Opisna statistika podatkovne zbirke

	Število vrstic	Povprečna vrednost	Standardni odklon	Minimalna vrednost	Maksimalna vrednost
Kazalnik 12	7638	46.11788	3,310.88731	-3,817.19747	288,126.96000
Kazalnik 13	7700	0.33399	0.24514	0	1
Kazalnik 14	7700	0.56228	0.25127	0	1
Kazalnik 15	7700	0.40564	0.25442	0	1
Kazalnik 16	7700	0.31025	3.06450	-250.31616	1
Kazalnik 17	7700	0.32719	3.06501	-250.31616	1
Kazalnik 18	7696	-0.33366	181.14612	-15,814.26667	851.74520
Kazalnik 19	7630	23.11872	2,060.59483	-4,579.18789	179,773.33330
Kazalnik 20	7700	0.49167	3.06138	-250.31616	1
Kazalnik 21	7676	-1.37196	127.12957	-11,074.36111	118.45349
Kazalnik 22	7676	-1.17171	126.85972	-11,074.36111	848.26605
Kazalnik 23	7676	-0.16072	174.23662	-9,767.22222	3,169.28571
Kazalnik 24	7676	4.07876	62.15196	-2,773.12500	2,978.89182
Kazalnik 25	7676	2.68311	43.67025	-1,089.47075	2,978.89182
Kazalnik 26	8989	100.94613	471.53378	0	39,635.77778
Kazalnik 28	7547	-0.57094	120.17623	-8,949.87847	1,522.90416
Kazalnik 29	7546	1.06646	1.80552	0	129.66549
Kazalnik 30	6961	3.89543	116.23170	-849.40350	8,493.07544
Kazalnik 31	6982	3.20426	109.11001	-1,723.00915	7,322.99691
Kazalnik 32	7611	-2.82120	199.38743	-15,642.40798	4,224.03147
Kazalnik 33	7638	-0.02440	7.56527	-643.36000	10.89087
Kazalnik 34	7638	-0.20018	11.24640	-948.08000	10.47901
Kazalnik 35	7638	-0.30464	15.78626	-1,330.15000	3.77594
Kazalnik 36	7696	0.23235	18.54467	-163.57564	1,605.93333
Kazalnik 37	7700	0.02004	0.38163	-9.42578	23.63601
Kazalnik 38	7700	0.03332	0.30562	-18.00908	1.74460

Vir: Lastno delo.

V stolpcu »število vrstic« je zapisano, koliko vrednosti ima posamezni kazalnik oziroma za koliko podjetij imamo na voljo izračun obravnavanega kazalnika. Tukaj smo opazili, da noben kazalnik nima vseh 9137 vrednosti, kar pomeni, da je v podatkovni zbirki ogromno manjkajočih vrednosti. Naslednji stolpec je povprečna vrednost posameznega kazalnika. Tu smo videli, da imajo kazalniki zelo različne obsege vrednosti. V naslednjem stolpcu je zapisan standardni odklon, ki pove kako razpršene so vrednosti okoli aritmetične sredine posameznega kazalnika. Na koncu imamo še minimalno in maksimalno vrednost kazalnikov, kjer ponovno vidimo veliko raznolikost. Ponovno opazimo, da je obseg vrednosti zelo različen in da je v podatkih veliko ekstremnih vrednosti. To je na primer

očitno pri kazalniku 32, kjer minimalna vrednost odstopa za 78 standardnih odklonov. Vrednosti kazalnikov niso enakomerno porazdeljene. Izjeme so kazalniki 13, 14 in 15.

4.3 Priprava podatkov

Pri pripravi oziroma čiščenju podatkov smo sledili korakom, kot so bili opredeljeni v poglavju 2.4.

4.3.1 Obravnavanje manjkajočih vrednosti

V dani podatkovni zbirki imamo veliko manjkajočih podatkov ali praznih vrednosti. Če odštejemo zapise, ki vsebujejo prazne vrednosti, nam jih od 9137 ostane le še 6898. Pri razvoju modelov z uporabo strojnega učenja je pri veliko tehnikah predpogoj, da podatkovna zbirka ne vsebuje manjkajočih podatkov. Splošno gledano ni najboljšega načina za reševanje problema manjkajočih vrednosti, pomembno pa je, da razumemo področje, s katerega izhajajo podatki ter da se odločimo za najprimernejšo rešitev. Preden smo se odločili, kako se soočati s tem problemom, smo morali najprej preučiti, za kakšno obliko praznih vrednosti gre.

Najlažji način, da ugotovimo, za katero kategorijo gre v naši podatkovni zbirki, je, da razumemo postopek zbiranja podatkov. V našem primeru so kazalniki rezultat izračunov raznih AJPES podatkov. Kljub temu da smo imeli na voljo informacijo, kako je izračunan posamezen kazalnik, jih nismo mogli naknadno izračunati, ker so pri teh zapisih v večini manjkali tudi AJPES podatki. Pridobivanje AJPES podatkov za posamezni zapis tudi ni bil mogoč, saj so zapisi anonimizirani. Iz tega lahko sklepamo, da manjkajoči podatki v našem primeru manjkajo popolnoma naključno.

Za potrditev te domneve lahko naredimo tudi statistični test, ki se imenuje »Little's MCAR test«. S to metodo lahko potrdimo predpostavko, da podatki manjkajo popolnoma naključno, če nam vrne p vrednost večjo od 0,05 (Li, 2013). Za izvedbo tega testa smo uporabili R knjižnico »LittleMCAR«. Rezultat testa je bila p vrednost 0,527, kar pomeni, da lahko potrdimo, da podatki manjkajo na naključen način. V primeru, da bi bili naši podatki v drugih dveh kategorijah, bi bilo najbolj smiselno izločiti vse zapise, ki vsebujejo prazne vrednosti. Ker podatki v našem primeru manjkajo na naključen način, pa lahko uporabimo metode vnašanja praznih vrednosti.

Za vnašanje vrednosti smo uporabili metodo »Hmisc«. Ta metoda samodejno prepozna vrste spremenljivk in uporabi prediktivno ujemanje srednjih vrednosti in vzorčno zankanje (angl. bootstrapping) za pripis manjkajočih vrednosti (University of Pennsylvania, brez datuma). Funkcija, ki jo uporabi za napovedovanje novih vrednosti, izdela več modelov na danih podatkih. Za vsako spremenljivko izračuna vrednost R kvadrat, ki nakazuje, kako

verodostojne so nove vrednosti v podatkovni zbirki. Za posamezne spremenljivke nato izbere model, ki ima najboljšo vrednost R kvadrat.

Našo podatkovno zbirko smo razdelili v dve skupini, tako da so bili v prvem delu vsi zapisi z vrednostjo default 0 in v drugi vsi zapisi s to vrednostjo 1. Za ta način smo se odločili, ker smo želeli, da bo metoda pri izračunu novih vrednosti imela na razpolago le vrednosti iz posamezne skupine.

Za izvedbo vnašanja novih vrednosti smo uporabili R knjižnico »Hmisc«. Vrednosti R kvadrat, ki smo jih dobili v posamezni skupini naše podatkovne zbirke za prvih in zadnjih pet kazalnikov (v tabeli označeni kot K in število kazalnika), so prikazane v tabeli 3.

Tabela 3: R kvadrat vrednosti podatkovne zbirke

Skupina 1: zapisi z vrednostjo default 0									
K1	K2	K3	K4	K5	K34	K35	K36	K37	K38
0,801	0,955	0,998	0,731	1,000	0,993	0,942	0,395	0,926	0,905
Skupina 2: zapisi z vrednostjo default 1									
K1	K2	K3	K4	K5	K34	K35	K36	K37	K38
0,235	0,784	0,905	0,620	1,000	0,996	1,000	1,000	0,958	0,938

Vir: Lastno delo.

Kot smo prej omenili, R kvadrat vrednosti povejo, kako verodostojne so nove vrednosti, ki so bile vnesene namesto praznih vrednosti v podatkovni zbirki. Vrednosti so v razponu med 0 in 1 – vrednost 1 pomeni, da je vnesena vrednost verodostojna, medtem ko vrednost 0 pomeni, da je nova vrednost le ugibanje brez povezave z ostalimi vrednostmi. V našem primeru je kar nekaj R kvadrat vrednosti bistveno pod 0,5. Na podlagi teh vrednosti, in še nekaterih drugih kriterijev, se bomo v naslednjem poglavju odločili, katere kazalnike bomo iz podatkovne zbirke izločili.

4.3.2 Odstranjevanje osamelcev

Metoda, ki smo jo uporabili za odkrivanje osamelcev, temelji na algoritmu »Isolation Forest«. Klasični pristopi odkrivanja osamelcev pripravijo oris primerov in označijo tiste, ki izstopajo. Za razliko od teh pristopov pa algoritem »Isolation Forest« upošteva vse spremenljivke v podatkovni zbirki in izrecno izloči osamelce (Liu, Ting & Zhou, 2009).

Deluje na podoben način kot odločitvena drevesa, ki razdeljuje podatke na več podskupin na podlagi informacijskega pribitka (angl. information gain) (Jain, 2017). Za razliko od odločitvenih dreves pa algoritem »Isolation Forest« razdeljuje podatke naključno. Osamelce izloči tako, da v podatkovni zbirki naključno izbere kazalnik in nato izbere ločilno vrednost med največjo in najmanjšo vrednostjo. Izbira ločilne vrednosti je odvisna od količine časa,

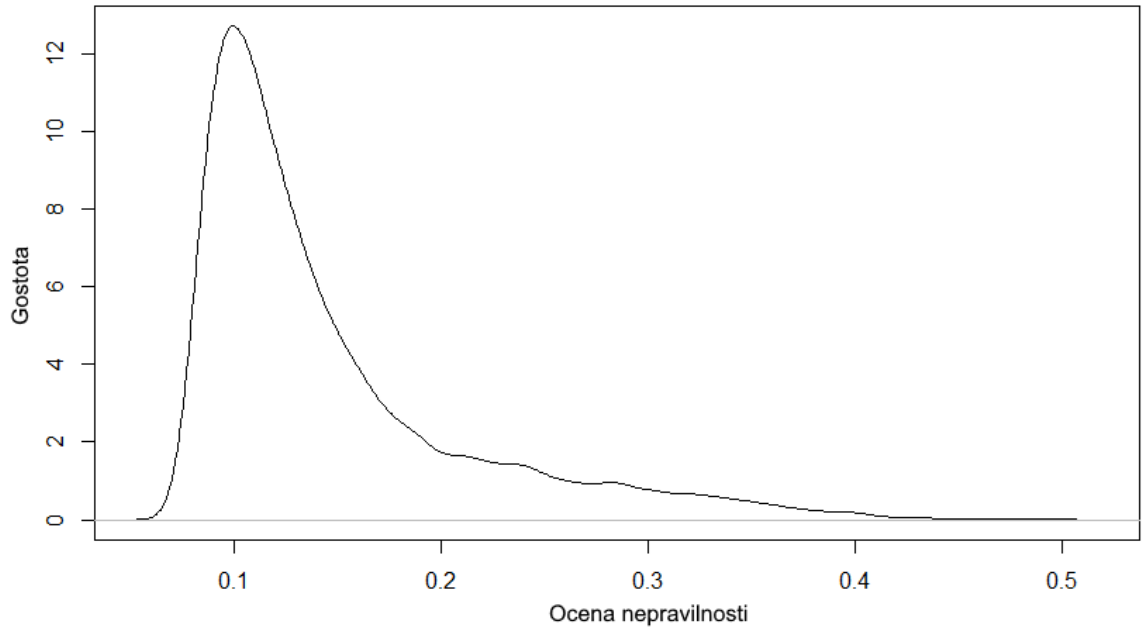
ki jo potrebuje, da loči te dve vrednosti (Sundar, 2019). Osamelci so v tem kontekstu manj pogosti in bolj oddaljeni kot navadne vrednosti.

Vsaka vrstica podatkov, oziroma sklop kazalnikov dobi oceno nepravilnosti (angl. anomaly score), na podlagi katere se lahko odločimo naslednje (Lewinson, 2018):

- Ocena blizu 1 nakazuje, da je vrednost osamelec
- Ocena, manjša od 0,5 nakazuje, da je vrednost normalna
- Če so vse ocene blizu 0,5, pomeni, da podatkovna zbirka nima jasno razločljivih osamelcev.

Na naših podatkih, z vnesenimi praznimi vrednostmi, smo naredili model Isolation Forest, za kar smo uporabili R knjižnico »Solitude«. Porazdelitev ocen nepravilnosti na naših podatkih je razvidna na sliki 4.

Slika 4: Porazdelitev ocen nepravilnosti na naših podatkih



Vir: Lastno delo.

Večina zapisov ima vrednosti ocen nepravilnosti približno 0,1, kar pomeni da ima jasno razberljive osamelce. Nad mejno vrednostjo 0,44 smo zaznali 27 zapisov. Za to ločilno vrednost smo se odločili, ker smo že pri njeni manjši spremembi zaznali veliko več zapisov.

V tej fazi smo se morali odločiti, kaj bomo z osamelci naredili. Te vrednosti v našem primeru niso rezultat napačnih izračunov, ampak le predstavljajo primere, ki zelo odstopajo od povprečne vrednosti ostalih. Po drugi strani pa nas zanima, v kolikšni meri vplivajo na naše rezultate. Ker nimamo utemeljitve za odstranitev teh vrednosti, bomo uporabili pristop, kjer bomo naredili analizo modelov enkrat z vključitvijo teh vrednosti in enkrat brez, ter rezultate primerjali.

Večina tehnik, ki smo jih uporabili za izdelovanje naših modelov, je odporna na osamelce, kar pomeni, da njihovo odstranjevanje ne bi bistveno vplivalo na njihove končne rezultate. Od uporabljenih tehnik za modeliranje je na osamelce v teoriji najbolj občutljiva logistična regresija (Baesens, Roesch & Scheule, 2016).

4.3.3 Normalizacija kazalnikov

Podatki v naši podatkovni zbirki imajo zelo različne obsege vrednosti in zato potrebujejo normalizacijo ali standardizacijo. Kot smo opisali v poglavju 2.4.3, je odločitev, katero od teh metod bomo uporabili, odvisna od modela. Za logistično regresijo in enorazredno metodo podpornih vektorjev smo uporabili standardizirane podatke, za ostale modele pa normalizirane.

Za ta dva postopka smo uporabili R knjižnico »Effectsize«. Za standardizacijo smo uporabili konvencionalni algoritem, za normalizacijo pa smo uporabili metodo »Z-score« (angl. Z-score normalization). To metodo smo izbrali, ker ni občutljiva na osamelce. Kljub temu da smo v prejšnjem poglavju odstranili nekaj teh vrednosti, naša podatkovna zbirka še vedno ni normalno porazdeljena in še vedno vsebuje veliko osamelcev. Klasične metode normaliziranja izračunajo nove vrednosti na podlagi minimalnih in maksimalnih vrednosti (na primer normalizacija min-max), Z-score normalizacija pa izračuna nove vrednosti na podlagi srednje vrednosti in standardnega odklona (Codecademy, brez datuma). Z uporabo tega načina izračuna podatkovna zbirka ohrani oddaljenosti med normalnimi vrednostmi in osamelci, kar je pomembno za modele, ki uporabljajo strojno učenje, kot na primer KNN in nevronske mreže.

V prejšnjem podpoglavju smo uporabili metodo odstranjevanja vrednosti, ki ni občutljiva na normalizacijo, kar pomeni, da odločitev, ali smo vrednosti odstranili pred ali po normalizaciji, ni pomembna, saj bosta rezultata na koncu enaka.

4.3.4 Zmanjševanje dimenzionalnosti podatkovne zbirke

Kot smo omenili v prejšnjem poglavju, imamo v podatkovni zbirki 38 kazalnikov, razdeljenih na 6 skupin, ki predstavljajo finančne položaje podjetij. Kazalniki v posameznih skupinah so si v večini primerov zelo podobni. To se še najbolj vidi v drugi skupini med kazalniki od 5 do 12, ki so izračunani na podlagi podobnih AJPES kazalnikov. Odločili smo se, da bomo pri izdelavi modelov preskusili obe metodi zmanjševanja dimenzionalnosti podatkov, ki smo jih opisali v poglavju 2.4.4, ter primerjali rezultate.

4.3.4.1 Izbor lastnosti

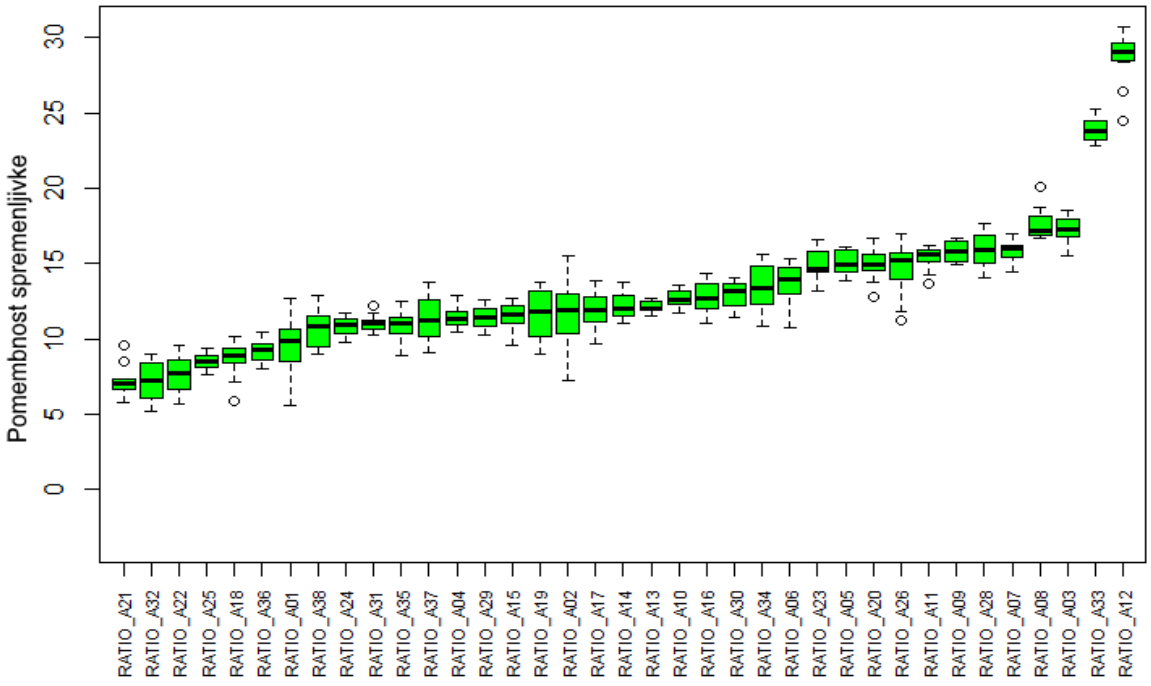
Za izbor lastnosti smo uporabili tri kriterije, in sicer:

1. R kvadrat vrednosti, ki nakazujejo verodostojnost na novo vnesenih vrednosti,
2. pomembnost spremenljivk na podlagi algoritma Boruta in
3. pomembnost spremenljivk na podlagi tehnike XGBoost.

Boruta je algoritem za določanje pomembnosti spremenljivk. Deluje tako, da podvoji podatkovno zbirko in premeša vrednosti v vsakem stolpcu. Nato na podatkih izvede poljubno klasifikacijsko tehniko. Pomembnost spremenljivk meri tako, da primerja izvirne spremenljivke z naključnimi – kot pomembne se štejejo le tiste, ki imajo višjo vrednost pomembnosti kot naključno generirane spremenljivke (Bhalla, 2017).

Boruta lahko deluje s katerokoli klasifikacijsko tehniko, v našem primeru smo uporabili naključne gozdove. Za izvedbo algoritma na naši podatkovni zbirki smo uporabili R knjižnico »Boruta«. Pomembnost posameznih spremenljivk oziroma kazalnikov, kjer je odvisna spremenljivka default, je prikazana s škatlami z brki (angl. box plot) na sliki 5.

Slika 5: Pomembnost spremenljivk na podlagi algoritma Boruta

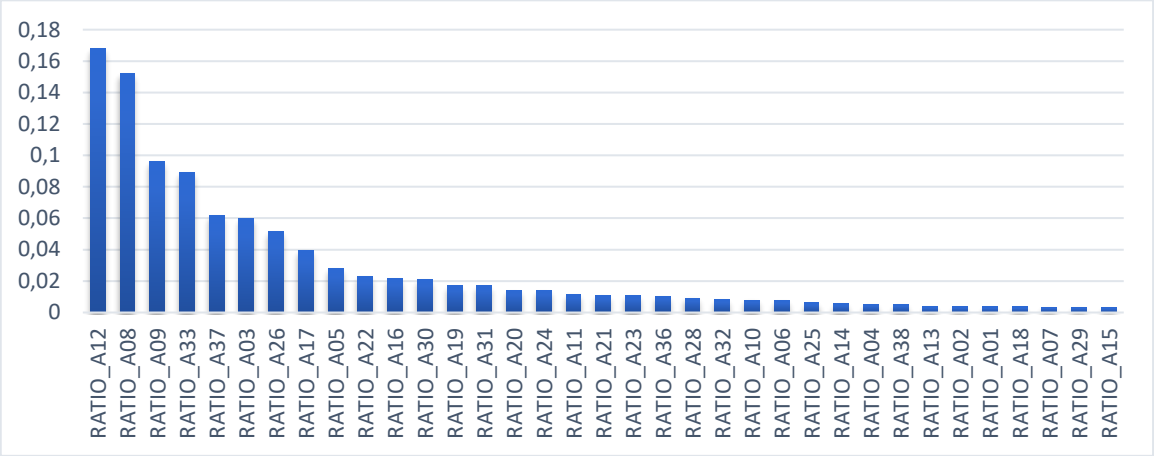


Vir: Lastno delo.

Spremenljivki 12 in 33 bistveno odstopata, ostale so pa na približno isti ravni.

Da ne bomo spremenljivk odstranili pristransko, smo uporabili še eno metodo, ki uporablja drugačen pristop za zaznavanje pomembnosti, to je metoda zviševanja gradientov tehnike XGBoost. Pomembnost spremenljivke izračuna na podlagi odločitvenih dreves, in sicer tako, da izračuna vpliv ločne vrednosti spremenljivke na posamezno drevo (Brownlee, 2020), kar se v algoritmu izraža z merilom »gain«. Pomembnost posameznih spremenljivk oziroma kazalnikov, kjer je odvisna spremenljivka ponovno default, je prikazana na sliki 6.

Slika 6: Pomembnost spremenljivk na podlagi tehnike XGBoost



Vir: Lastno delo.

Kot je razvidno s slike 6, imata kazalnika 12 in 8 bistveno prednost pred ostalimi. Razlike med rezultati algoritma Boruta in tehnike XGBoost so kar očitne.

Primerjali smo rezultate teh dveh metod in pri tem upoštevali R kvadrat vrednosti na novo vnesenih vrednosti ter se nato za vsako posamezno spremenljivko odločili, če jo bomo obdržali v končni podatkovni zbirki. Pri izbiri je imela največjo težo vrednost R kvadrat – če je bila ta nizka, smo spremenljivko odstranili, tudi če je imela sprejemljivo pomembnost z ostalima metodama.

Na koncu smo odstranili 12 kazalnikov: 1, 4, 18, 19, 21, 22, 26, 27, 28, 31, 2, 36.

Na podlagi teh rezultatov smo izdelali tudi podatkovno zbirko, kjer celo skupino predstavlja le en kazalnik, glavni razlog za to pa je, da ta način uporabljajo v izbrani banki. To podatkovno zbirko sestavljajo naslednji kazalniki:

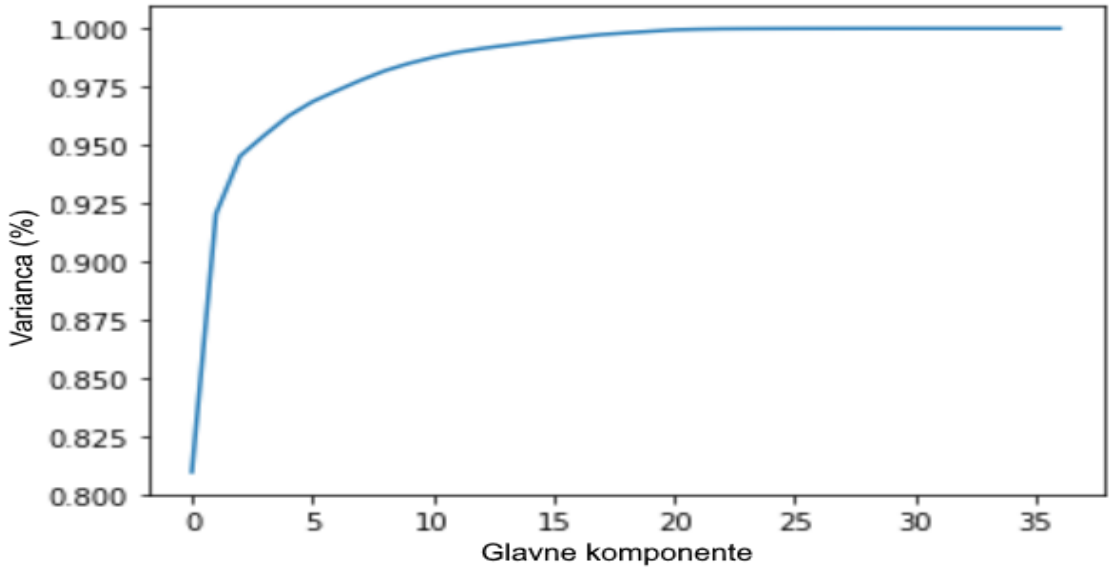
- Skupina 1: kazalnik 3
- Skupina 2: kazalnik 12
- Skupina 3: kazalnik 16
- Skupina 4: kazalnik 24
- Skupina 5: kazalnik 30
- Skupina 6: kazalnik 33

4.3.4.2 Ekstrakcija lastnosti

Za ekstrakcijo lastnosti smo uporabili metodo glavnih komponent (angl. principal component analysis, v nadaljevanju PCA). S to metodo smo združili naše spremenljivke v nov nabor linearno ne koreliranih spremenljivk, ki se imenujejo glavne komponente (Brems, 2017). Glavne komponente so neodvisne ena od druge. Nove spremenljivke, glavne

komponente, povejo, kako dobro so ohranile informacije, ki jih je imela prvotna podatkovna zbirka – to se izraža z vrednostjo skupna varianca, ki je vsota varianc posameznih glavnih komponent (Šušteršič, Mramor & Zupan, 2009). Metodo smo izvedli na naši normalizirani podatkovni zbirki in dobili graf varianc, ki je prikazan na sliki 7.

Slika 7: Graf varianc glavnih komponent



Vir: Lastno delo.

Z grafa smo razbrali, da z izbiro približno osemnajstih komponent ohranimo skoraj vso varianco podatkov. Tako smo se odločili za ločilno vrednost in ustvarili novo podatkovno zbirko z osemnajstimi kazalniki.

Na koncu procesa izbire kazalnikov smo tako imeli tri različne podatkovne zbirke:

1. podatkovno zbirko z odstranjenimi kazalniki,
2. podatkovno zbirko, ki je bila preoblikovana s PCA, in
3. podatkovno zbirko z enim kazalnikom na skupino.

4.3.5 Ponovno vzorčenje

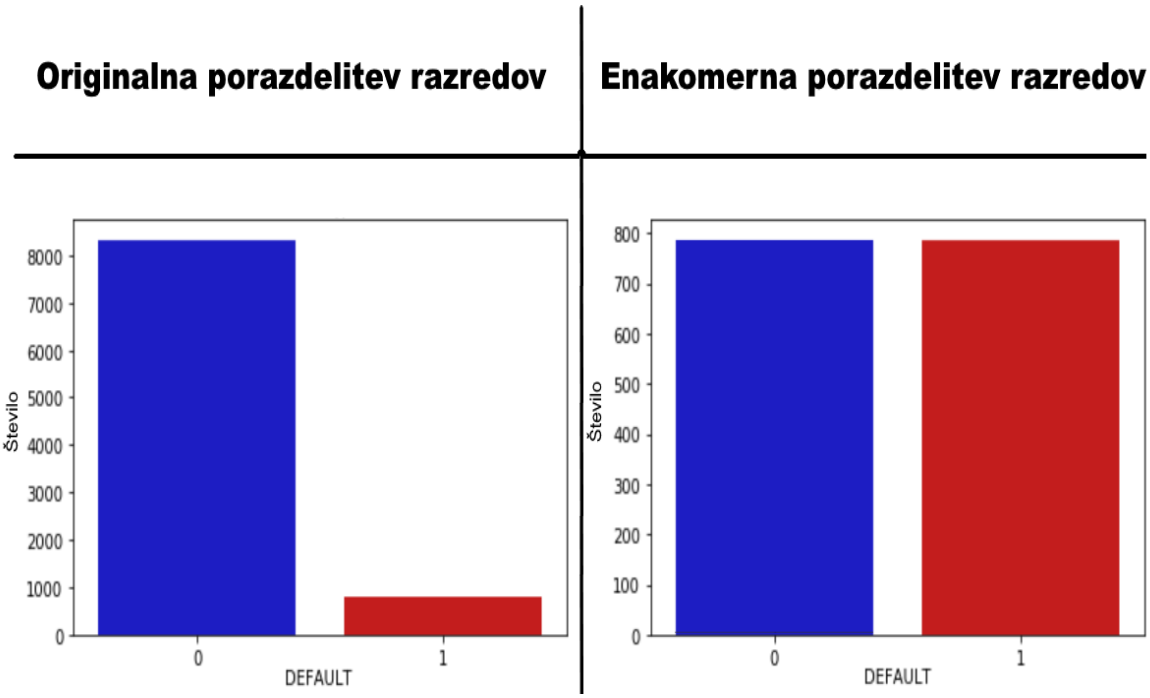
Naša podatkovna zbirka je zelo neuravnotežena, saj večina zapisov ne predstavlja dogodka neplačila. 787 zapisov oz. 8,64 odstotkov podatkovne zbirke, ima vrednost default 1, ostalih 8323 primerov, oz. 91,36 odstotkov, pa vrednost 0.

Če bi za ustvarjanje modelov uporabili podatkovno zbirko v takšni obliki, bi naši razvrstitveni modeli predvidevali, da se dogodek neplačila skoraj nikoli ne zgodi. V izogib temu problemu smo ustvarili dva nova vzorca podatkov z metodami, ki smo jih opisali v poglavju 2.4.5. Uporabili smo podvzorčenje in sintetično generiranje podatkov.

4.3.5.1 *Podvzorčenje*

Za podvzorčenje smo uporabili metodo naključnega podvzorčenja (angl. random undersampling). Ta metoda naključno izbere opažanja iz večinskega razreda in jih briše, dokler se število teh opažanj ne ujema s številom opažanj manjšinskega razreda. Podatkovna zbirka pred tem postopkom in po postopku je prikazana na sliki 8.

Slika 8: Originalna porazdelitev razredov in porazdelitev po podvzorčenju



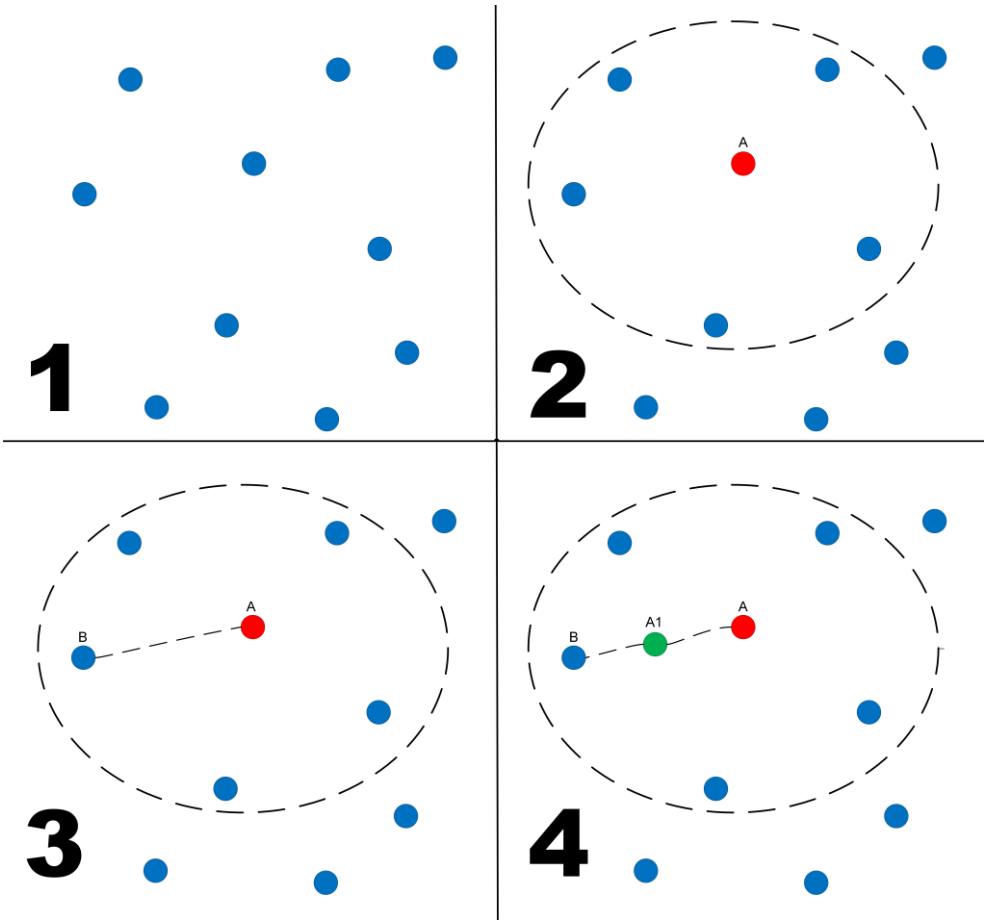
Vir: Lastno delo.

4.3.5.2 *Sintetično generiranje podatkov*

Za sintetično generiranje podatkov smo uporabili tehniko prevzorčenja s sintetičnimi manjšinskimi primeri (angl. synthetic minority oversampling, v nadaljevanju SMOTE). To je pristop, ki ustvari sintetične vzorce manjšinskega razreda, zaradi česar postane porazdelitev razredov bolj uravnotežena (Bagherpour, 2017). Sintetični so zato, ker so na novo ustvarjeni iz obstoječih primerov manjšinskega razreda.

Poenostavljen primer procesa generiranja novega sintetičnega vzorca je prikazan na sliki 9 v štirih korakih. V prvem koraku je prikazan del podatkovne zbirke. Za ustvarjanje novega sintetičnega vzorca SMOTE najprej naključno izbere primer manjšinskega razreda »A« (v koraku 2 na sliki označen z rdečo barvo) in poišče njegove k-najbližje sosede (na sliki obkroženi z črtkasto elipso). Sintetični vzorec se ustvari tako, da izbere enega izmed k-najbližjih sosedov »B«, ter naključno poveže »A« in »B« in med njima naredi nov primer (sintetični vzorec) »A1«, kar je razvidno v koraku 3 in 4 (He & Ma, 2013).

Slika 9: Proces generiranja novega sintetičnega vzorca



Prirejeno po Pozzolo, Caelen, Waterschoot & Bontempi (2015).

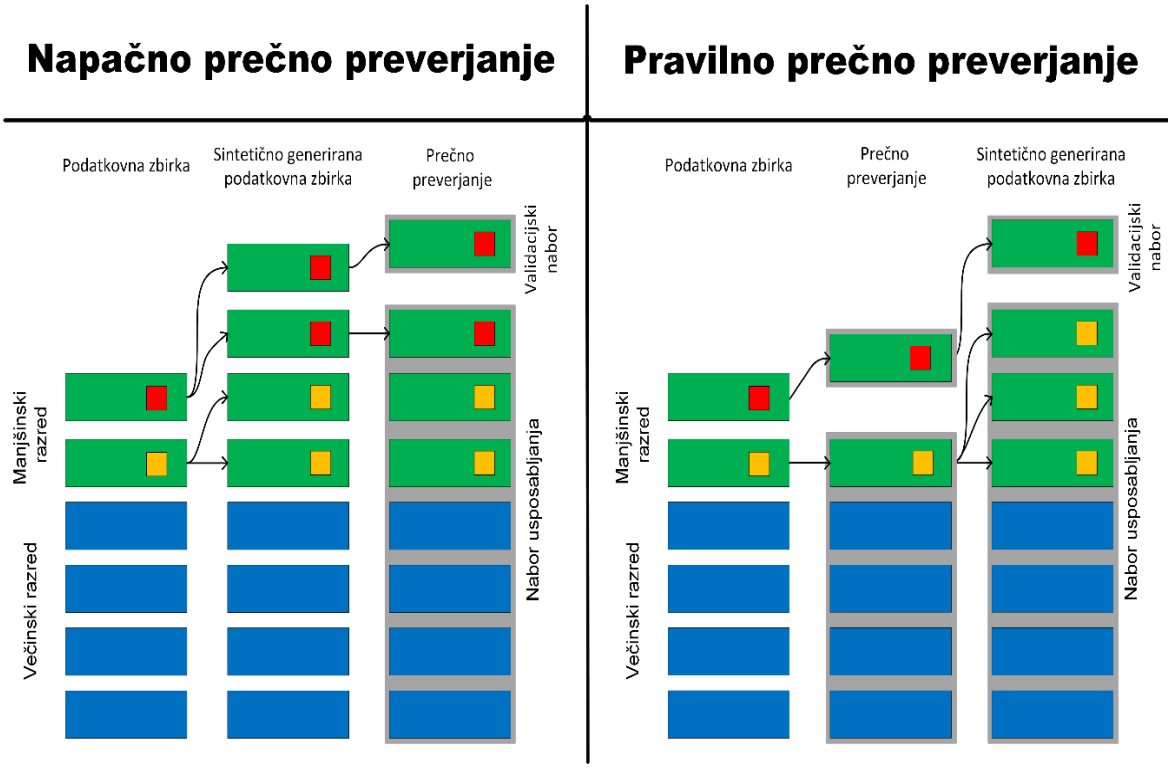
Tako imamo na koncu procesa dve novi podatkovni zbirki: eno podvzorčeno in eno s sintetično generiranimi vzorci.

4.3.6 Prečno preverjanje podatkov

V vseh naših podatkovnih zbirkah smo izvedli prečno preverjanje podatkov tako, da smo jih razdelili na nabor usposabljanja in nabor testiranja. Za nabor testiranja smo namenili 30 odstotkov podatkovne zbirke, ker smo na ta način dobili bistveno boljše rezultate kot z 20 odstotki podatkovne zbirke pri začetnem testiranju modelov. Eno petino nabora testiranja smo namenili validacijskemu naboru.

Da smo se izognili prenasičenosti, smo morali podatkovno zbirko razdeliti pred izvajanjem sintetičnega generiranja podatkov. Če bi razdelitev naredili za tem postopkom, bi lahko imeli povsem ista opažanja v naboru usposabljanja in testiranja. Omenjeni problem in pravilen postopek sta prikazana na sliki 10.

Slika 10: Pravilen postopek prečnega preverjanja



Prirejeno po Altini (2015).

Na levi strani je prikazan napačen postopek prečnega preverjanja. V začetni podatkovni zbirki imamo dva vzorca podatkov. Pri sintetičnem generiranju jih podvojimo in nato izvedemo prečno preverjanje. Iz tega sledi, da ima nabor usposabljanja in validacijski nabor ista vzorca podatkov, kar povzroči prenasičenost. Na desni strani pa je prikazan pravilen postopek prečnega preverjanja. Tu najprej izvedemo prečno preverjanje in tako namenimo en vzorec za validacijo, preostali vzorec s sintetičnim generiranjem pa podvojimo. Tako v tem primeru ne uporabljamo istih podatkov za usposabljanje in validacijo.

5 MODELI IN NJIHOVO VREDNOTENJE

Pri modeliranju vseh modelov bomo testirali različne podatkovne zbirke in predstavili rezultat tiste kombinacije, ki nam je dala najboljši rezultat. Izbirali bomo med:

1. podatkovno zbirko z osamelci,
2. podatkovno zbirko brez osamelcev,
3. standardizirano podatkovno zbirko,
4. normalizirano podatkovno zbirko,
5. podatkovno zbirko z odstranjenimi odvečnimi kazalniki,
6. podatkovno zbirko, ki je bila preoblikovana s PCA,
7. podatkovno zbirko z enim kazalnikom na skupino,

8. podvzorčeno podatkovno zbirko in
9. podatkovno zbirko s sintetično generiranimi vzorci.

Za ocenjevanje učinkovitosti napovedovanja naših modelov smo lahko izbirali med različnimi meritvami vrednotenja (angl. evaluation metric). Ker vsaka od teh meritev predstavi drugačno plat učinkovitosti napovedovanja modela, je bilo zelo pomembno, da smo jih med seboj ločili.

Glavna meritev, na podlagi katere lahko izračunamo tudi vse ostale, je **matrika zamenjave** (angl. confusion matrix), ki prikazuje vrste pravilnih in napačnih napovedi. Izračun matrike zamenjav nam nudi predstavo o tem, kaj model napove točno in kakšne vrste napak dela (Shung, 2018). Razdeljena je na štiri dele, od katerih vsak del predstavlja rezultate napovedi modela. Primer matrike, ki se bo uporabljala v našem primeru, je prikazan v tabeli 4.

Tabela 4: Primer matrike zamenjave

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**Pravi negativi** (PN)	**Lažni negativi** (LN)
	Default: 1	**Lažni pozitivi** (LP)	**Pravi pozitivi** (PP)

Vir: Lastno delo.

Na koncu modeliranja modele testiramo s testnim naborom podatkov. Model napove rezultate odvisne spremenljivke, ki se primerjajo z dejansko vrednostjo. V našem primeru posamezne vrednosti pomenijo naslednje:

- **Pravi pozitivi (PP)**: število primerov, za katere je model pravilno napovedal, da so dogodek neplačila.
- **Pravi negativi (PN)**: število primerov, za katere je model pravilno napovedal, da niso dogodek neplačila.
- **Lažni pozitivi (LP)**: število primerov, za katere je model napačno napovedal, da so dogodek neplačila.
- **Lažni negativi (LN)**: število primerov, za katere je model napačno napovedal, da niso dogodek neplačila.

Pri kreditnem ocenjevanju je najbolj pomembno, da čim bolj zmanjšamo število lažnih pozitivov (Matheson, 2018). Primer lažnega pozitiva je, da model označi prosilca za posojilo kot neplačnika, tudi če bi bil dejansko uspešen.

Na podlagi matrike zamenjave lahko izpeljemo druge meritve (Oppermann, 2019):

- **Natančnost** (angl. precision) nam pove razmerje pravih pozitivnih primerov, ki jih je model pravilno napovedal. Nizka natančnost kaže, da je veliko rezultatov lažno pozitivnih.

- **Priklic** (angl. recall) nam pove, kako dobro model prepoznava prave pozitivne rezultate. Nizek priklic kaže, da je veliko vrednosti lažno negativnih.
- **Metrika F1** (angl. F1 score) je tehtano povprečje meritev natančnosti in priklica. Dobra vrednost te meritve nakazuje, da sta obe vrednosti visoki.

Za vsak model bomo prikazali napovedno moč s temi tremi meritvami za obe vrednosti razreda in matriko zamenjave. V našem primeru bo pri ocenjevanju imela največjo težo meritev metrika F1, saj želimo prepoznati čim več pravih pozitivnih rezultatov, hkrati pa so naše podatkovne zbirke neuravnotežene. Pri predstavitvi meritev bomo tudi povedali, kolikšno **število** posameznih vrednosti razreda (default 0 in 1) je bilo upoštevanih.

V določenih primerih bomo uporabili meritev **točnost** (angl. accuracy), ki na splošno pove, kako pogosto model pravilno napoveduje. Je vsota vseh pravilnih napovedi (pravih negativov in pravih pozitivov), deljeno s skupnim številom napovedi. Točnost je dobra meritev, v kolikor je vrednost lažnih pozitivov in lažnih negativov skoraj enaka. V primerih, ko imamo neuravnoteženo porazdelitev razredov, je ta meritev zelo zavajajoča in je zato ne bomo uporabili za prikaz napovedne moči modelov.

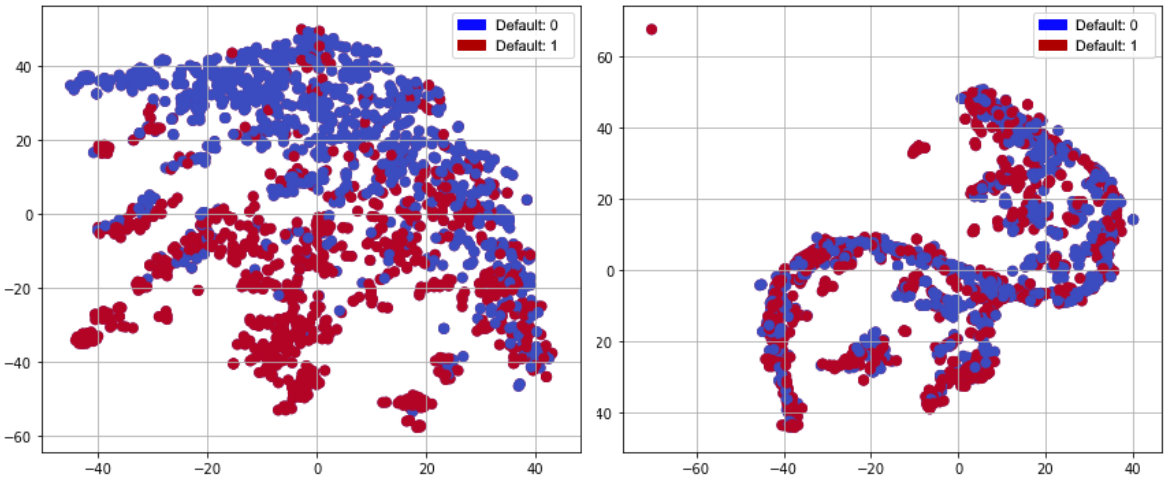
Kot metodo za ocenjevanje modelov bomo uporabili tudi krivuljo karakteristike delovanja sprejemnika (angl. Receiver Operating characteristic curve, v nadaljevanju ROC). Ta nam prikaže, kako dobro lahko model razlikuje med pozitivnim in negativnim razredom (Oppermann, 2019). Zmogljivost ločevanja se opredeli z meritvijo, ki se imenuje območje pod krivuljo (angl. area under the curve, v nadaljevanju AUC). AUC 1 pomeni, da je model popoln in da vedno naredi pravilno razvrstitev, medtem ko AUC pod 0,5 pomeni, da model ne zmore razlikovati med pozitivnim in negativnim razredom.

5.1 Modeliranje ocenjevalnega modela z logistično regresijo

Za modeliranje logistične regresije smo uporabili standardizirano podatkovno zbirko brez osamelcev. Glavna primerjava je potekala med podatkovno zbirko z odstranjenimi odvečnimi kazalniki in zbirko, preoblikovano z metodo glavnih komponent.

Da smo dobili občutek, katera bo bolj primerna za modeliranje, smo uporabili tehniko za vizualizacijo podatkov t-SNE (angl. Distributed Stohastic Neighbor Embedding). S to tehniko prikažemo, kako je razred dogodka neplačila (default) porazdeljen v prostoru. Bolj sta vrednosti ločeni, lažje jih bo napovedni model ločil. S slike 11 je razvidno, da so vrednosti razreda 0 in 1 veliko bolje ločene za podatkovno zbirko z odstranjenimi odvečnimi kazalniki (levi del slike). Iz tega sklepamo, da bodo imeli modeli, ki bodo uporabili to podatkovno zbirko, boljšo napovedno moč. Rezultat je prikazan na sliki 11, in sicer za podatkovno zbirko z odstranjenimi odvečnimi kazalniki na levi in za podatkovno zbirko, preoblikovano z metodo glavnih komponent na desni.

Slika 11: t-SNE vizualizacija porazdelitve razreda default



Vir: Lastno delo.

Za modeliranje logistične regresije in več drugih modelov strojnega učenja smo uporabili Python knjižnico Scikit-Learn. Scikit-learn ponuja vrsto nadzorovanih in nenadzorovanih algoritmov učenja, osredotoča pa se na modeliranje podatkov (Brownlee, 2020).

V modelu logistične regresije se verjetnosti, ki opisujejo možne izide posameznega primera, modelirajo z uporabo logistične funkcije. Če je izhodna vrednost te funkcije višja kot 0,5, rezultat razvrstimo kot 1, kar predstavlja dogodek neplačila. Logistična funkcija za podatkovno zbirko z odstranjenimi odvečnimi kazalniki, kjer β_0 do β_{26} predstavlja koeficiente, x_1 do x_{26} pa kazalnike iz podatkovne zbirke, je prikazana v enačbi (1).

$$P(Y) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_{26} x_{26})}} \quad (1)$$

Natančnost modelov smo izboljševali z iskanjem optimalnih parametrov. To smo naredili z mrežnim iskanjem (angl. grid search), ki preveri vse možne parametre za klasifikatorje in pove, kateri dajo najboljši napovedni rezultat. Uporabili smo »GridSearchCV«, ki je funkcija znotraj knjižnice Scikit-Learn. Pri izdelavi modela je za naslednje parametre izbrala naslednje optimalne rezultate (https://scikit-learn.org/):

- **solver**: izbira algoritma, ki se bo uporabil za oceno parametrov – lbfgs,
- **class_weight**: uteži, oziroma pomembnost vrednosti odvisne spremenljivke – 1 za razred 0 in 15 za razred 1 (dogodki neplačil),
- **penalty**: metoda regularizacije ali penalizacijska funkcija – L2,
- **C**: vrednost obratnega vpliva regularizacije – 1.0,
- **fit_intercept**: določi, če se odločitveni funkciji doda konstanto (konstantno vrednost) – True,
- **dual**: dvojna ali enojna formulacija podatkov – False, se pravi enojna formulacija,
- **max_iter**: maksimalno število iteracij oziroma poskusov iskanja rezultatov – 100 in

- **tol**: toleranca modela za ustavljanje – 0.0001**.**

Za oceno parametrov je bil uporabljen optimizacijski algoritem Broyden–Fletcher–Goldfarb–Shanno (LBFGS), ki je način iskanja lokalnega minimuma objektivne funkcije z uporabo vrednosti objektivnih funkcij in naklona ciljne funkcije (Nocedal & Wright, 2006). Metoda regularizacije, ki se je uporabila za preprečitev preprileganja podatkov, se imenuje L2. Ta deluje tako, da penalizira oziroma kaznuje velike uteži v naši stroškovni funkciji in tako zniža odstopanja modela (Lau, Gonzalez & Nolan, brez datuma).

Pri uporabi metod ponovnega vzorčenja smo dobili najboljše rezultate s sintetično generiranimi vzorci v primerjavi s podvzorčenimi. Rezultati modela logistične regresije za različni podatkovni zbirki sta prikazani v tabelah 5 in 6.

Tabela 5: Rezultati za logistično regresijo s podatkovno zbirko z odstranjenimi odvečnimi kazalniki

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**206** (PN)	**22** (LN)
	Default: 1	**65** (LP)	**180** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,76	0,91	0,83	232
Default: 1	0,89	0,73	0,80	241
Povprečje	0,83	0,82	0,82	473

Vir: Lastno delo.

Tabela 6: Rezultati za logistično regresijo s podatkovno zbirko, preoblikovano z metodo glavnih komponent

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**200** (PN)	**43** (LN)
	Default: 1	**60** (LP)	**173** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,77	0,82	0,79	239
Default: 1	0,80	0,75	0,77	237
Povprečje	0,79	0,78	0,82	476

Vir: Lastno delo.

Pri rezultatih s podatkovno zbirko z odstranjenimi odvečnimi kazalniki smo dobili kar dober rezultat z zelo nizkim številom lažnih negativov. Po drugi strani pa je število lažnih pozitivov kar visoko. Še ena slabost je, da se natančnost in priklic razredov 0 in 1 za kar precej razlikuje. Tudi če je povprečje meritev relativno dobro, tega ne smemo zanemariti.

Rezultati s podatkovno zbirko, preoblikovano z metodo glavnih komponent, so bili za vse meritve slabši v primerjavi s prejšnjo podatkovno zbirko. Rezultati meritev za razred pa so precej podobni in bolj uravnovešeni.

Kar se tiče primerjave načina ponovnega vzorčenja, je bila točnost modelov v obeh primerih boljša s sintetičnimi manjšinskimi primeri. Primerjava je prikazana v tabeli 7.

Tabela 7: Primerjava točnosti za različni metodi ponovnega vzorčenja

Metoda ponovnega vzorčenja	**Točnost za podatkovno zbirko**	
	Brez odvečnih kazalnikov	Metoda glavnih komponent
Naključno podvzorčenje	0,816	0,784
Prevzorčenje s sintetičnimi manjšinskimi primeri	0,858	0,833

Vir: Lastno delo.

Za model logistične regresije smo poskusili uporabiti osnovno podatkovno zbirko brez kakršnekoli obdelave podatkov. Rezultati so bili presenetljivo le malo nižji od zgoraj predstavljenih. Za vse zgoraj navedene modele smo tudi preiskali, kakšen vpliv na rezultate ima vključevanje osamelcev v podatkovne zbirke. Za vse primere so se rezultati poslabšali za približno en odstotek. To je skladno s teorijo, ki pravi da parametrični klasifikatorji, kot je logistična regresija, ponavadi izgubijo napovedno moč z obstoječimi osamelci (Paleologo, Elisseeff & Antonini, 2010).

5.2 Modeliranje ocenjevalnega modela s tehnikami strojnega učenja

Za modeliranje s tehnikami strojnega učenja smo uporabili normalizirano podatkovno zbirko brez odstranjevanja osamelcev. Pri vseh teh modelih je odstranjevanje osamelcev negativno vplivalo na napovedno moč. Glavna primerjava je v vseh primerih potekala med podvzorčeno podatkovno zbirko in podatkovno zbirko s sintetično generiranimi vzorci. Modeli, za katerega smo uporabili podatkovno zbirko z enim kazalnikom na skupino, je imel bistveno slabšo napovedno moč v primerjavi z ostalimi.

5.2.1 Modeliranje ocenjevalnega modela z k-najbližjih sosedov

Za modeliranje k-najbližjih sosedov smo uporabili Scikit-Learn. Z mrežnim iskanjem smo za naslednje parametre izbrali naslednje optimalne rezultate (https://scikit-learn.org/):

- **algorithm**: algoritem, ki se uporablja za izračun najbližjih sosedov – BallTree,
- **leaf_size**: velikost lista modela, ki je uporabljen v algoritmu – 30,
- **metrics**: vrsta meritve razdalje za drevo modela – minkowski,
- **p**: parameter moči za metriko minkowski – 2,

- **n_jobs**: število vzporednih opravil za iskanje k sosedov – 1,
- **n_neighbors**: število sosedov, ki jih algoritem uporabi za poizvedbe – 3 in
- **weights**: teža posameznega soseda, ki se uporablja pri napovedovanju – uniform.

Uporabljen algoritem »BallTree« je posebna struktura podatkov, ki razdeli podatkovne točke v ugnezdeni niz in tako potrebuje le del podatkov za izdelavo modela (Knighten, 2019). Uporabljena vrsta metrike »minkowski« in p vrednost 2 pomeni, da razdalja za drevo modela temelji na evklidski razdalji. Teža »uniform« pomeni, da so v modelu vsi sosedi enako oteženi.

Pri uporabi metod ponovnega vzorčenja smo dobili najboljše rezultate z naključnim podvzorčenjem. Rezultati modela k-najbližjih sosedov so prikazani v tabeli 8.

Tabela 8: Rezultati za k-najbližjih sosedov z naključno podvzorčenimi podatki

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**208** (PN)	**20** (LN)
	Default: 1	**50** (LP)	**195** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,81	0,91	0,86	233
Default: 1	0,91	0,79	0,85	240
Povprečje	0,86	0,85	0,85	473

Vir: Lastno delo.

V matriki zamenjave smo dobili precej dobre rezultate z nizkim številom lažnih negativov in malo višjim številom lažnih pozitivov. Meritev natančnosti je nekoliko višja za razred 1, kar je pri kreditnem ocenjevanju zelo pozitivna lastnost. Pri priklicu pa velja obratno in sicer, da je za razred 0 meritev bistveno višja.

Klasifikator k-najbližjih sosedov zahteva uravnoteženi razred za optimalno delovanje in se odloča na podlagi celotnega nabora podatkov (Mukid, Widiharih, Prahutama & Rusgiyono, 2017). Zaradi tega je bilo očitno, da bomo najboljše rezultate dosegli z podvzorčeno podatkovno zbirko. Odstranjevanje osamelcev ni imelo vpliva na rezultate modelov.

5.2.2 Modeliranje ocenjevalnega modela z odločitvenimi drevesi

Za modeliranje z odločitvenimi drevesi smo uporabili Scikit-Learn. Z mrežnim iskanjem smo za naslednje parametre izbrali naslednje optimalne rezultate (https://scikit-learn.org/):

- **class_weight**: uteži razredov – balanced,
- **criterion**: funkcija, ki se uporabi za merjenje kakovosti delitve vozlišč – gini,
- **max_depth**: maksimalna globina posameznega drevesa – int,

- **max_features**: število značilnosti, ki jih model upošteva pri delitvi – int,
- **max_leaf_nodes**: največje število vozlišč – None, se pravi neomejeno,
- **min_samples_leaf**: najmanjše število vzorcev, potrebnih za listno vozlišče – 1,
- **min_samples_split**: najmanjše število vzorcev, potrebnih za razdelitev notranjega vozlišča – 2,
- **min_weight_fraction_leaf**: najmanjši tehtani delež vsote uteži vseh vzorcev – 0,0 in
- **splitter**: strategija, ki se uporablja za izbiro delitve na vsakem vozlišču – best.

Funkcija za delitev kakovosti vozlišč temelji na Gini nečistoči (angl. Gini impurity). Vozlišča se širijo, dokler vsi listi ne dosegajo maksimalne globine drevesa ali dokler vsi listi ne dosegajo vrednost »min_samples_split«. Strategija »best« deluje tako, da v vsakem primeru izbere najboljšo delitev.

Pri uporabi metod ponovnega vzorčenja smo dobili najboljše rezultate z naključnim podvzorčenjem. Rezultati modela odločitvenih dreves so prikazani v tabeli 9.

Tabela 9: Rezultati za odločitvena drevesa z naključno podvzorčenimi podatki

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**180** (PN)	**55** (LN)
	Default: 1	**28** (LP)	**210** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,86	0,76	0,81	232
Default: 1	0,79	0,88	0,84	241
Povprečje	0,83	0,82	0,82	473

Vir: Lastno delo.

Matrika zamenjave je precej neuravnovešena, kar se kaže v tem, da je vrednost lažnih negativov različna in dosti večja od vrednosti lažnih pozitivov. Meritve so na splošno precej dobre. Pri vseh treh meritvah je majhna razlika med vrednostmi razreda. Povprečja meritev s podatkovno zbirko s sintetično generiranimi vzorci so bila identična, le metrika F1 je bila za eno točko nižja.

5.2.3 Modeliranje ocenjevalnega modela z naključnimi gozdovi

Za modeliranje z naključnimi gozdovi smo uporabili Scikit-Learn. Z mrežnim iskanjem smo za naslednje parametre izbrali naslednje optimalne rezultate (https://scikit-learn.org/):

- **bootstrap**: določi, ali se pri gradnji dreves uporablja vzorčno zankanje – True,
- **class_weight**: uteži oziroma pomembnost vrednosti odvisne spremenljivke – 1 za razred 0 in 12 za razred 1 (dogodki neplačil),
- **criterion**: funkcija, ki se uporabi za merjenje kakovosti delitve – entropy,

- **max_depth**: maksimalna globina dreves – 8,
- **max_features**: število značilnosti, ki jih model upošteva pri iskanju najboljše razdelitve – log2,
- **max_leaf_nodes**: največje število vozlišč – None,
- **min_samples_leaf**: najmanjše število vzorcev, potrebnih za listno vozlišče – 10,
- **min_samples_split**: najmanjše število vzorcev, potrebnih za delitev notranjega vozlišča – 2,
- **min_weight_fraction_leaf**: najmanjši tehtani delež vsote uteži – 0,0 in
- **n_estimators**: število dreves v gozdu – 30.

Funkcija za delitev kakovosti vozlišč za posamezno drevo temelji na informacijskem pribitku. Pri iskanju števila značilnosti za iskanje najboljše razdelitve je optimalna logaritmirana vrednost števila spremenljivk.

Pri uporabi metod ponovnega vzorčenja smo dobili najboljše rezultate z naključnim podvzorčenjem. Rezultati modela naključnih gozdov so prikazani v tabeli 10.

Tabela 10: Rezultati za naključne gozdove z naključno podvzorčenimi podatki

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**199** (PN)	**34** (LN)
	Default: 1	**33** (LP)	**207** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,86	0,85	0,86	232
Default: 1	0,86	0,86	0,86	241
Povprečje	0,86	0,86	0,86	473

Vir: Lastno delo.

Matrika zamenjave je zelo uravnovešena, saj sta vrednosti lažnih pozitivov in lažnih negativov skoraj enaki. Meritve uspešnosti napovedi so bile visoke in skoraj vse identične, kar je zelo pozitivna lastnost modela. Model, pri katerem smo uporabili podatkovno zbirko s sintetično generiranimi vzorci, je imel vrednost vseh meritev 85, in je bil tako slabši od modela z naključno podvzorčenimi podatki.

5.2.4 Modeliranje ocenjevalnega modela z metodami podpornih vektorjev

Primerjali smo osnovno metodo podpornih vektorjev in enorazredno metodo podpornih vektorjev. Za modeliranje obeh smo uporabili Scikit-Learn. Za metodo podpornih vektorjev smo z mrežnim iskanjem za naslednje parametre izbrali naslednje optimalne rezultate (https://scikit-learn.org/):

- **C**: parameter regularizacije – 1,

- **class_weight**: uteži oziroma pomembnost vrednosti odvisne spremenljivke – None, se pravi 1 za obe,
- **decision_function_shape**: oblika odločitvene funkcije – ovr,
- **kernel**: določa vrsto funkcije za določitev meje med razredi (jedro), uporabljene v algoritmu – rbf in
- **tol**: toleranca za kriterij ustavljanja funkcije – 0.001.

Funkcija, ki določa mejo odločitve med razredi ali jedri, je osnovana na funkcija radialne osnove (angl. radial basis function, v nadaljevanju RBF). Odločitvena funkcija deluje po principu »One-vs.-rest«, kar pomeni, da usposablja en klasifikator na razred.

Pri uporabi metod ponovnega vzorčenja smo dobili najboljše rezultate z naključnim podvzorčenjem. Rezultati modela naključnih gozdov so prikazani v tabeli 11.

Tabela 11: Rezultati za metodo podpornih vektorjev z naključno podvzorčenimi podatki

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**150** (PN)	**81** (LN)
	Default: 1	**22** (LP)	**220** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,87	0,65	0,75	232
Default: 1	0,73	0,91	0,81	241
Povprečje	0,80	0,78	0,78	473

Vir: Lastno delo.

V matriki zamenjave opazimo, da je zelo veliko število lažnih negativov. Meritve so precej nizke, hkrati pa so zelo različne med razredi. Model, pri katerem smo uporabili podatkovno zbirko s sintetično generiranimi vzorci, je bil nekoliko slabši.

Za enorazredno metodo podpornih vektorjev smo za razliko od osnovnega modela uporabili standardizirane podatke. Z mrežnim iskanjem smo za naslednje parametre izbrali naslednje optimalne rezultate (https://scikit-learn.org/):

- **kernel**: določa vrsto jedra uporabljenega v algoritmu – rbf,
- **gamma**: koeficient za jedro – 0,1,
- **nu**: zgornja meja deleža napak pri usposabljanju – 0,5,
- **shrinking**: določi ali se pri računanju uporabi hevristično krčenje – True in
- **tol**: toleranca za kriterij ustavljanja funkcije – 0.001.

Uporabljena je bila ista funkcija za določitev meje med razredi kot pri osnovni metodi. Meritve natančnosti enorazredne metode podpornih vektorjev so prikazane v tabeli 12.

Tabela 12: Meritve natančnosti za enorazredne metode podpornih vektorjev z naključno podvzorčenimi podatki

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,92	0,47	0,62	232
Default: 1	0,65	0,96	0,78	241
Povprečje	0,78	0,72	0,70	473

Vir: Lastno delo.

Pri meritvah takoj vidimo, da so slabše od osnovnega modela in na splošno zelo slabe. Za razred 0 je priklic pod 0,5, kar je velik znak, da takšnega modela ne bi bilo smiselno uporabiti. Presenetljivo je, da je za razred 0 natančnost in za razred 1 priklic tako visok.

5.2.5 Modeliranje ocenjevalnega modela z nevronskimi mrežami

Za modeliranje z nevronskimi mrežami smo uporabili Keras, ki je Python knjižnica za razvoj in ocenjevanje modelov globokega učenja (angl. deep learning) (https://keras.io/). Ta uporablja ogrodje za strojno učenje TensorFlow in omogoča definiranje ter usposabljanje modelov nevronskih mrež.

Prvi korak je bil oblikovanje modela. Povzetek predstavitve našega modela s podatkovno zbirko z naključnim podvzorčenjem je prikazan na sliki 12.

Slika 12: Povzetek modela z nevronskimi mrežami

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 26)                702
_________________________________________________________________
dense_2 (Dense)              (None, 32)                864
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 66
=================================================================
```

Vir: Lastno delo.

Model je razdeljen na tri globoko povezane plasti nevronske mreže, imenovane »Dense«. Za vsako plast se uporabi določeno število vhodnih podatkov iz nabora usposabljanja. Model smo nato usposabljali s parametri in vrednostmi:

- **validation_split**: del nabora usposabljanja, ki se uporablja za validacijo – 0,2,
- **batch_size**: število vzorcev za posamezno posodobitev gradienta – 25,
- **epochs**: kolikokrat bo algoritem učenja procesiral celoten nabor podatkov – 20 in

- **shuffle**: ali naj se podatki premešajo pred vsakim procesiranjem – True.

Algoritem učenja se je dvajsetkrat procesiral za pridobivanje končnega modela. Za vsako procesiranje nabora podatkov (imenovano »epoch«) se izpiše uspešnost modela, ki se z vsakim procesiranjem izboljša. Najpomembnejša meritev v tem izpisu je »val_acc«, ki predstavlja točnost napovedi modela z validacijskim naborom podatkov. Del izpisa za naš model je predstavljen na sliki 13.

Slika 13: Del izpisa postopka procesiranja modela z nevronskimi mrežami

```
Train on 880 samples, validate on 221 samples
Epoch 1/20
 - 1s - loss: 2.6320 - acc: 0.5580 - val_loss: 1.7823 - val_acc: 0.5566
Epoch 2/20
 - 0s - loss: 1.1873 - acc: 0.7068 - val_loss: 1.0114 - val_acc: 0.7511
                                    ⋮
Epoch 19/20
 - 0s - loss: 0.4334 - acc: 0.8534 - val_loss: 0.7664 - val_acc: 0.8190
Epoch 20/20
 - 0s - loss: 0.4014 - acc: 0.8477 - val_loss: 0.7355 - val_acc: 0.8235
```

Vir: Lastno delo.

Na sliki vidimo, da se je bilo za nabor usposabljanja namenjenih 880 primerov, za validacijski nabor pa 221. S procesiranjem se je točnost izboljšala iz 0,56 na 0,82 skozi dvajset iteracij.

Pri uporabi metod ponovnega vzorčenja smo dobili najboljše rezultate z naključnim podvzorčenjem. Rezultati modela nevronskih mrež so prikazani v tabeli 13.

Tabela 13: Rezultati za nevronske mreže z naključno podvzorčenimi podatki

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**206** (PN)	**26** (LN)
	Default: 1	**47** (LP)	**194** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,81	0,89	0,85	232
Default: 1	0,88	0,80	0,84	241
Povprečje	0,85	0,85	0,85	473

Vir: Lastno delo.

Matrika zamenjave je precej slaba, saj je število lažnih pozitivov bistveno večje kot lažnih negativov. Kljub temu so meritve uspešnosti v povprečju dobre. Model, pri katerem smo

uporabili podatkovno zbirko s sintetično generiranimi vzorci, nam je vrnil zelo slabe rezultate, pri katerih ni niti ene vrednosti pravilno uvrstil kot pravi pozitiv.

5.2.6 Modeliranje ocenjevalnega modela z LightGBM

Za modeliranje z LightGBM smo uporabili Python knjižnico LightGBM (Microsoft Corporation, brez datuma). Z mrežnim iskanjem smo za naslednje parametre izbrali naslednje optimalne rezultate:

- **boosting_type**: metoda zviševanja gradientov – gdbt,
- **class_weight**: pomembnost vrednosti odvisne spremenljivke – None, enakomerno,
- **colsample_bytree**: vzorčenje stolpcev na drevo – 0,6002251666834131,
- **importance_type**: definira, kako se izračuna pomembnost – split,
- **learning_rate**: množenje na vsakem ponovnem zvišanju – 0,2,
- **max_depth**: omeji največjo globino dreves – 7,
- **metric**: metrika, ki se uporabi v naboru ocenjevanja – None, nobena metrika,
- **min_child_samples**: najmanjše število podatkov, potrebnih v listu drevesa – 230,
- **min_child_weight**: najmanjša vsota, potrebna v listu – 10,0,
- **n_estimators**: število dreves v modelu – 500,
- **num_leaves**: največje število listov za posamezno drevo – 7,
- **reg_alpha**: izraz na utež za regularizacijo L1 – 50,
- **reg_lambda**: izraz na utež za regularizacijo L2 – 5 in
- **subsample**: razmerje posameznega primera za usposabljanje – 0,9508421672126002.

Metoda zviševanja gradientov »gdbt« pomeni, da se za model uporabljajo odločitvena drevesa. Izračun pomembnosti temelji na številu ponovitev oziroma pove, kolikokrat je vsaka spremenljivka uporabljena v modelu.

Pri uporabi metod ponovnega vzorčenja smo dobili najboljše rezultate z naključnim podvzorčenjem. Rezultati modela nevronskih mrež so prikazani v tabeli 14.

Tabela 14: Rezultati za nevronske mreže z naključno podvzorčenimi podatki

		Dejanski razred	
		Default: 0	Default: 1
Napovedan razred	Default: 0	**208** (PN)	**33** (LN)
	Default: 1	**28** (LP)	**204** (PP)

	Natančnost	**Priklic**	**Metrika F1**	**Število**
Default: 0	0,88	0,86	0,87	237
Default: 1	0,86	0,88	0,87	236
Povprečje	0,87	0,87	0,87	473

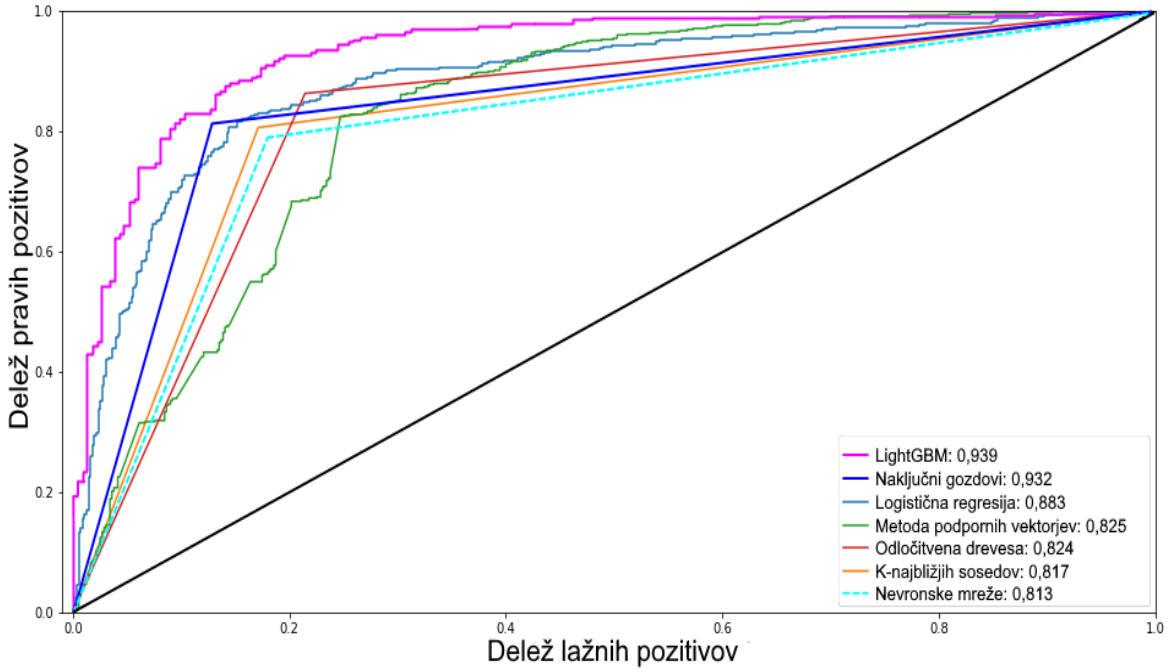
Vir: Lastno delo.

Matrika zamenjave je uravnovešena z nizkimi vrednostmi lažnih pozitivov in lažnih negativov. Meritve uspešnosti napovedi so zelo visoke in uravnovešene. Model, pri katerem smo uporabili podatkovno zbirko s sintetično generiranimi vzorci, nam je vrnil bistveno slabše rezultate.

5.3 Krivulje ROC

Za najboljši model v svoji kategoriji smo naredili grafični prikaz krivulj ROC. Skupek teh krivulj je predstavljen na sliki 14.

Slika 14: Krivulje ROC vseh najboljših modelov



Vir: Lastno delo.

V spodnjem desnem kotu je legenda, ki pove, kakšne barve je krivulja ROC posameznih modelov. Poleg tega je v legendi zabeležen še AUC rezultat posamezne krivulje. Razlog, da so nekatere krivulje oblikovane iz lomljene črte, druge pa so veliko bolj razgibane, je v tem, da v Scikit-Learn nekatere funkcije, ki smo jih uporabili, nimajo možnosti izrisa funkcije odločanja ali pa sam model tega ne omogoča.

Že na prvi pogled je razvidno, da so krivulje za modele LightGBM, naključne gozdove in logistično regresijo boljše, oziroma, da dobro razlikujejo med pozitivnim in negativnim razredom, saj imajo že v začetnem delu krivulje dobro razmerje pozitivov. Krivulja modela metode podpornih vektorjev ima v zadnjem delu zelo dobro razmerje, celo primerljivo z modelom LightGBM, vendar ima v prvem delu med vsemi modeli najslabše razmerje, kar je posledica visokega števila lažnih negativov.

6 UGOTOVITVE IN DISKUSIJA

6.1 Povzetek rezultatov

Povzetek vseh rezultatov je predstavljen v tabeli 15.

Tabela 15: Povzetek rezultatov vseh najboljših modelov

	Povprečja meritev			
Model	**Natančnost**	**Priklic**	**Metrika F1**	**ROC AUC**
Logistična regresija	0,83*	0,82*	0,82	0,883
K-najbližjih sosedov	0,86*	0,85*	0,85	0,817
Odločitvena drevesa	0,83*	0,82*	0,82	0,824
Naključni gozdovi	0,86	0,86	0,86	0,932
Metoda podpornih vektorjev	0,80*	0,78*	0,78	0,825
Nevronske mreže	0,85*	0,85*	0,85	0,813
LightGBM	**0,87**	**0,87**	**0,87**	**0,939**

Vir: Lastno delo.

V tabeli je poleg meritve znak *, če je imela meritev velike razlike med vrednostmi 1 in 0 v odvisni spremenljivki default. Odebeljeni so najboljši rezultati v posamezni kategoriji. Edina dva modela, ki sta imela popolno uravnovešeni vrednosti meritev, sta modela naključnih gozdov in LightGBM. Hkrati sta ta dva modela izmed vseh tudi daleč najboljša.

Če primerjamo model **logistične regresije** in model **odločitvenih dreves**, ki sta **najlažje interpretativna modela**, vidimo, da so rezultati presenetljivo podobni. Ob bolj podrobni preučitvi rezultatov obeh modelov pa je razvidno, da ima vsak od njiju različne prednosti in slabosti, in sicer:

- Model logistične regresije ima nizko število lažnih negativov in visoko število **lažnih pozitivov**, model odločitvenih dreves pa obratno.
- Model logistične regresije ima boljše rezultate meritev za default vrednost 0, model odločitvenih dreves pa boljše meritve za default vrednost 1.

Glede na to, da je pri kreditnem ocenjevanju pomembno, da čim bolj zmanjšamo število lažnih pozitivov, je tukaj boljši model odločitvenih dreves. Poleg tega je pri kreditnem ocenjevanju ključna napoved dogodka neplačila (vrednosti 1 za default) in ima tudi v tem smislu prednost model odločitvenih dreves. Iz tega sledi, da je izmed teh dveh modelov za kreditno ocenjevanje bolj primeren model odločitvenih dreves.

Rezultati za model **nevronskih mrež** so bili na prvi pogled dobri, vendar je bila v meritvah razlika med vrednostmi razreda kar precejšnja. Model je vrnil tudi kar precej **visoko število lažnih pozitivov**, kar se vidi tudi v krivulji ROC, ki je presenetljivo slaba. Zelo podobno je veljalo za model **K-najbližjih sosedov**, ki je pa bil v povprečju le nekoliko boljši od modela nevronskih mrež.

Rezultati za **model podpornih vektorjev** so bili na splošno slabši, kot smo pričakovali. Kljub temu da je model vrnil relativno **nizko število lažnih pozitivov,** so bili rezultati meritev slabi. Pri uporabi te metode je izziv, kako izbrati optimalno podmnožico vhodnih spremenljivk, saj je model nanje zelo občutljiv. Če bi želeli izboljšati rezultat za ta model, bi morali izračunati pomembnost kazalnikov izključno za to metodo, vendar je velika verjetnost, da tudi s tem ne bi mogli prekašati ostalih modelov.

Kot smo izpostavili v poglavju 2.4.4, metode ekstrakcij lastnosti ne morejo ustrezno ravnati s kompleksnimi nelinearnimi podatki. To pojasni, zakaj nismo z uporabo podatkovne zbirke, ki je bila preoblikovana s PCA, v nobenem primeru dobili boljšega rezultata v primerjavi s podatkovnimi zbirkami, kjer smo uporabili metode izbora lastnosti.

Pri večini modelov smo dobili najboljše rezultate z naključno podvzorčeno normalizirano podatkovno zbirko z odstranjenimi kazalniki. Opisna statistika te podatkovne zbirke je prikazana v prilogi 2. Izključevanje osamelcev je na napovedno moč imelo pozitiven vpliv le za model logistične regresije.

6.2 Diskusija raziskovalnih vprašanj

Na raziskovalna vprašanja magistrskega dela, ki smo jih zastavili v uvodu, smo prišli do naslednjih odgovorov:

1. **So modeli kreditnega ocenjevanja, ki temeljijo na pristopih strojnega učenja, lahko boljši od konvencionalnih, ki temeljijo na logistični regresiji, oziroma ali imajo boljšo napovedno moč?**

Z nekaterimi modeli strojnega učenja smo dobili bistveno boljšo napovedno moč v primerjavi z modelom logistične regresije. Tukaj bi posebno izpostavili modela z uporabo naključnih gozdov in LightGBM, ki sta z vsakega vidika vrnila boljše rezultate. Zaradi podobnih slabosti, kot jih je imel model logistične regresije, ne bi priporočili uporabe modelov nevronskih mrež in K-najbližjih sosedov, kljub temu da sta imela v določenih primerih boljše rezultate.

Izbrana banka nam je zaupala, da ima njihov model na izbranem testnem naboru meritev točnosti 0,74. V našem primeru za meritev uspešnosti modelov nismo uporabili merila točnosti zaradi razlogov, ki smo jih opisali v začetku 5. poglavja, z našim testnim naborom pa je model LightGBM dosegel točnost 0,87, model naključnih gozdov pa 0,86. Pri teh

rezultatih se moramo zavedati, da sta bila testna nabora podatkov različna in da je model izbrane banke tehtano povprečje še enega modela, ki je uporabil nefinančne podatke.

2. **Ali v izbranem primeru obstaja potencial za uporabo pristopov strojnega učenja za oblikovanje modela kreditnega ocenjevanja, s katerim napovemo verjetnost dogodka neplačila, in katere so ključne ovire pri tem?**

Ključna ovira pri uporabi pristopov strojnega učenja za oblikovanje kreditnega ocenjevanja je razumljivost oziroma interpretativnost. Ker se model, ki ga uporablja izbrana banka uporablja neposredno za odobravanje kreditov, jim ni dovoljeno uporabiti tehnik, ki so težje interpretativne. Zaradi tega večina metod, ki smo jih predstavili ni uporabnih. Izjema je model odločitvenih dreves, ki nam je z vidika uporabe za kreditno ocenjevanje dal boljše rezultate v primerjavi z modelom logistične regresije, iz česar sledi, da za njegovo uporabo vsekakor obstaja potencial.

Tudi za paralelno uporabo, torej za referenco poleg osnovnega modela, in za sistem zgodnjega opozarjanja brez dvoma obstaja potencial uporabe pristopov strojnega učenja, še posebej za tiste modele, ki so nam vrnili bistveno boljše rezultate.

3. **Ali je vlaganje v razvoj novega lastnega modela kreditnega ocenjevanja, ki temelji na pristopih strojnega učenja, na dolgi rok v izbranem primeru smiselna investicija?**

Oblikovanje modela, ki je v naši raziskavi pokazal najboljše rezultate – LightGBM in modela odločitvenih dreves bi skupaj z mrežnim iskanjem najprimernejših parametrov trajalo le nekaj minut tudi s celotno podatkovno zbirko. Dejstvo je, da se je naš način priprave podatkov v primerjavi z bančnim zelo razlikoval. V primeru, da bi se banka odločila za razvoj modela LightGBM ali odločitvenih dreves, bi morala vložiti čas v ponovno pripravo podatkov za optimalne rezultate. Rezultati modela pa bi bili zelo verjetno že malo boljši s podatkovno zbirko, ki jo uporabljajo za trenutni model. Že z izborom kazalnikov, ki bi bili prilagojeni pomembnosti modela LightGBM, bi lahko ta rezultat bistveno izboljšali. V pomoč bi jim lahko bil tudi način priprave podatkov, opisan v magistrskem delu.

Menim, da bi lahko banka z minimalno časovno investicijo oblikovala modela LightGBM ali odločitvenih dreves, ki bi imela zelo dobro napovedno moč. Zaradi tega bi bilo vlaganje v razvoj novega modela smiselno že kratkoročno.

SKLEP

V okviru magistrskega dela smo opredelili način razvoja modelov kreditnega ocenjevanja z uporabo strojnega učenja in vse izzive, ki se pri tem pojavijo. Ugotovili smo, da je pri razvoju modelov s strojnim učenjem zelo pomembna izbira primernega načina priprave podatkov, pri kateri imamo ogromno možnosti, vsaka od teh pa ima svoje prednosti in slabosti.

Raziskali smo posamezne tehnike modeliranja s strojnim učenjem in preučili smiselnost uporabe posamezne metode za kreditno ocenjevanje.

Za razvoj modelov smo na podlagi preučevanja literature in primera izbrane banke izbrali najsodobnejše in po našem mnenju najprimernejše metode priprave podatkov in tako pripravili več podatkovnih zbirk, ki smo jih uporabili za oblikovanje modelov. Odločili smo se, da bodo naši modeli sledili filozofiji sistemov točk v času, pri razvoju pa smo sledili razvojnim stopnjam metodologije CRISP-DM. Razvili smo model z logistično regresijo in šest modelov z uporabo tehnik strojnega učenja, pri vseh pa smo uporabili identične meritve uspešnosti. Za razvite modele smo primerjali rezultate in preučili smiselnost uporabe posameznih modelov za kreditno ocenjevanje in za opredeljen primer izbrane banke.

Modeli, ki so imeli boljšo napovedno moč v primerjavi z modelom logistične regresije, so bili modeli odločitvenih dreves, naključnih gozdov in LightGBM. Zaradi zahtev razumljivosti modelov v izbrani banki bi lahko model logistične regresije nadomestil le model odločitvenih dreves, ostala dva pa bi bila primerna za paralelno uporabo ali za sistem zgodnjega opozarjanja.

Ugotovili smo, da v opredeljenem primeru obstaja možnost uporabe modelov, ki uporabljajo strojno učenje. Razvili smo več modelov, preučili možnost njihove uporabe in ovire ter tako dosegli zastavljene cilje magistrskega dela. Rezultati in ugotovitve so se v veliki meri ujemali s preučenimi raziskavami. Slednje so pokazale, da v veliko primerih s tehnikami, kot so nevronske mreže, metoda podpornih vektorjev in naključni gozdovi, dosežemo boljšo napovedno moč v primerjavi z logistično regresijo. Ugotovili smo, da v našem primeru ne bi bila primerna uporaba metode podpornih vektorjev in da kljub temu, da je imel model nevronskih mrež na prvi pogled boljšo napovedno moč, ne bi bil primeren za namen kreditnega ocenjevanja.

Ključni prispevek magistrskega dela je ugotovitev, da v danem primeru izbrane banke obstaja potencial uporabe strojnega učenja za modeliranje kreditnega ocenjevanja in dokaz, da je to smiselna investicija. Empirično smo dokazali, da lahko s podatkovno zbirko, ki jo banka trenutno uporablja za modeliranje, in z uporabo strojnega učenja naredimo model, ki ima boljšo napovedno moč kot logistična regresija, ki je trenutno uporabljena tehnika modeliranja v banki.

Ovira pri delu je bila izbira metode priprave podatkov, saj je imela zelo velik vpliv na končne rezultate modelov. Preizkusili smo različne kombinacije metod, vendar se zavedamo, da je tukaj še veliko možnosti za izboljšavo. Omejitev dela je bila, da nismo imeli možnosti na enak način testirati izdelanih modelov z modelom izbrane banke. Prav tako nam ni bila na voljo podatkovna zbirka, ki bi vsebovala podatke novih primerov, s katero bi lahko potrdili uspešnost modelov v realnem primeru. Možnost za nadaljnje delo bi bila dodatna raziskava na področju priprave podatkov, s katero bi lahko prišli do podatkovnih zbirk, ki bi bile bolj primerne za izbran primer. Nadalje bi lahko preučili tudi možnost uporabe nadomestnih

modelov za izboljšanje interpretativnosti. Rezultati magistrskega dela ponujajo priložnost izbrani banki, da bi razvila in uvedla model z uporabo tehnike odločitvenih dreves in tako natančno ugotovila, za koliko bi se izboljšala napovedna moč.

LITERATURA IN VIRI

1. Abdelmoula, A. K. (2015). Bank Credit Risk Analysis with K-Nearest-Neighbor Classifier: Case of Tunisian Banks. *Journal of Accounting and Management Information Systems, 14,* 79-106.
2. Abid, L., Masmoudi, A. & Zouari-Ghorbel, S. (2018). The Consumer Loan's Payment Default Predictive Model: an Application of the Logistic Regression and the Discriminant Analysis in a Tunisian Commercial Bank. *Journal of the Knowledge Economy, 9,* 948-962.
3. Altini, M. (2015, 17. august). *Dealing with imbalanced data: undersampling, oversampling and proper cross-validation* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation
4. Altman, E. I. (2018). A Fifty-Year Retrospective on Credit Risk Models, the Altman Z - Score Family of Models and their Applications to Financial Markets and Managerial Strategies. *Journal of Credit Risk, 14*(4).
5. Baesens, B., Roesch, D. & Scheule, H. (2016). *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS.* New Jersey: Wiley.
6. Bagherpour, A. (2017). *Predicting Mortgage Loan Default with Machine Learning Methods.* Riverside: University of California.
7. Bhalla, D. (2017, 1. julij). *Select Important Variables using Boruta Algorithm* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://www.datasciencecentral.com/profiles/blogs/select-important-variables-using-boruta-algorithm
8. Breiman, L. (1999). *Random Forests - Random Features.* Berkeley: University of California.
9. Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science, 16*(3), 199-215.
10. Brems, M. (2017, 17. april). *A One-Stop Shop for Principal Component Analysis* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c
11. Brownlee, J. (2020). Machine Learning Mastery. *Imbalanced Classification with Python.* Vermont: Machine Learning Mastery.
12. Brunel, V. (2016, 8. november). Lifetime PD Analytics for Credit Portfolios: A Survey. *SSRN Electronic Journal*.
13. Caire, D. & Kossmann, R. (2003, februar). *Credit Scoring: Is It Right for Your Bank?* Pridobljeno 10. aprila 2020 iz http://www.microfinance.com/DeanCaire/Caire-Is-Credit-Scoring-Right-for-Your-Bank.pdf

14. Cao, A., He, H., Chen, Z. & Zhang, W. (2018). Performance Evaluation of Machine Learning Approaches for Credit Scoring. *International Journal of Economics, Finance and Management Sciences, 6,* 255-260.
15. Cao, R., Vilar, J. M., Rivera, A. E., Veraverbeke, N., Boucher, J.-P. & Beran, J. (2009). Modelling consumer credit risk via survival analysis. *Statistics and Operations Research Transactions*.
16. Codecademy. (brez datuma). *Normalization.* Pridobljeno 10. aprila 2020 iz https://www.codecademy.com/articles/normalization
17. Drakos, G. (2018, 16. august). *Cross-Validation* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://medium.com/@george.drakos62/cross-validation-70289113a072
18. Dumitrescu, E., Hue, S., Hurlin, C. & Tokpavi, S. (2018). Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects.
19. Engelmann, B. & Rauhmeier, R. (2011). *The Basel II Risk Parameters: Estimation, Validation, Stress Testing - with Applications to Loan Risk Management.* New York: Springer-Verlag Berlin Heidelberg.
20. Estrella, A. (2000). *Credit Ratings And Complementary Sources Of Credit Quality Information.* Basel: Basel Committee On Banking Supervisionworking Papers.
21. Frost, J. (brez datuma). *Guidelines for Removing and Handling Outliers in Data*. Pridobljeno 10. aprila 2020 iz https://statisticsbyjim.com/basics/remove-outliers/
22. Garcia, V. & Marques, A. I. (2012). Improving Risk Predictions by Preprocessing Imbalanced Credit Data. *Neural Information Processing*, 68-75.
23. Gestel, T. V., Baesens, B., Dijcke, P. V., Suykens, J. A., Garcia, J. & Alderweireld, T. (2005). Linear and Non-linear Credit Scoring by Combining Logistic Regression and Support Vector Machines. *Journal of Credit Risk, 1*(4).
24. Gill, N. & Hall, P. (2019). *An Introduction to Machine Learning Interpretability.* Kalifornija: O'Reilly Media, Inc.
25. Goh, R. Y. (2019). Credit Scoring: A Review on Support Vector Machines and Metaheuristic Approaches. *Advances in Operations Research, 2019*(2), 1-30.
26. Grennepois, N. (2018, september). *Using Random Forest for credit risk models.* Pridobljeno 10. aprila 2020 iz https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/financial-services/deloitte-nl-fsi-using-random-forest-for-credit-risk-models.pdf
27. Guru99. (brez datuma). *Supervised vs Unsupervised Learning: Key Differences*. Pridobljeno 10. aprila 2020 iz https://www.guru99.com/supervised-vs-unsupervised-learning.html
28. Hall, P. (2016, 11. februar). *Predictive modeling: Striking a balance between accuracy and interpretability*. Pridobljeno 10. aprila 2020 iz https://www.oreilly.com/content/predictive-modeling-striking-a-balance-between-accuracy-and-interpretability/
29. Haltuf, M. (2014). *Support Vector Machines for Credit Scoring* (magistrsko delo). Praga: University of Economics, Prague.
30. Han, L., Han, L. & Zhao, H. (2013). Orthogonal support vector machine for credit scoring. *Engineering Applications of Artificial Intelligence, 26*(2), 848-862.

31. Harris, T. (2015). Credit scoring using the clustered support vector machine. *Expert Systems with Applications, 42*(2), 741-750.
32. He, H. & Ma, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications.* New Jersey: Wiley.
33. Hsieh, H.-I., Lee, T.-P. & Lee, T.-S. (2010). Data Mining in Building Behavioral Scoring Models. *International Conference on Computational Intelligence and Software Engineering*.
34. Hué, S., Hurlin, C. & Tokpavi, S. (2017). *Machine Learning for Credit Scoring: Improving Logistic Regression with Non Linear Decision Tree Effects.* Paris: Paris Nanterre University.
35. Hui, L., Li, S. & Zongfang, Z. (2013). The Model and Empirical Research of Application Scoring based on Data Mining Methods. *Procedia Computer Science, 17*, 911-918.
36. Hulstaert, L. (2019, 14. marec). *Black-box vs. white-box models* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://towardsdatascience.com/machine-learning-interpretability-techniques-662c723454f3
37. Hung, C. & Chen, J.-H. (2009). A selective ensemble based on expected probabilities for bankruptcy prediction. *Expert Systems with Applications, 36*(3), 5297-5303.
38. Hussein, A. A. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature. *Intelligent Systems in Accounting Finance & Management, 18*(2-3), 59-88.
39. Hwang, Y. H. (2018). *C# Machine Learning Projects.* Birmingham: Packt Publishing.
40. Jain, R. (20. 3 2017). *Decision Tree. It begins here* [objava na blogu]. Pridobljeno 10. april 2020 iz https://medium.com/@rishabhjain_22692/decision-trees-it-begins-here-93ff54ef134
41. Jansma, M. (2018, 28. februar). *What are scoring models and how do they come about?* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://www.berkeleybridge.com/blog/what-are-the-scoring-models-and-how-do-they-come-about/
42. Kennedy, K. (2013). *Credit Scoring Using Machine Learning* (doktorska desertacija). Dublin: Technological University Dublin.
43. Keramati, A. & Yousefi, N. (2011). A Proposed Classification of Data Mining Techniques in Credit Scoring. *International Conference on Industrial Engineering and Operations Management* (str. 22-24). Kaula Lumpur.
44. Khemais, Z., Nesrine, D. & Mohamed, M. (2016). Credit Scoring and Default Risk Prediction: A Comparative Study between Discriminant Analysis & Logistic Regression. *International Journal of Economics and Finance*.
45. Klicki, W. & Szymielewicz, K. (2019, 12. junij). *The right to explanation of creditworthiness assessment*. Pridobljeno 10. aprila 2020 iz https://en.panoptykon.org/right-to-explanation
46. Knighten, J. (2019, 26. marec). *KNN and BallTree Overview*. Pridobljeno 10. aprila 2020 iz https://github.com/JKnighten/k-nearest-neighbors/wiki/KNN-and-BallTree-Overview

47. Koh, H. C., Tan, W. C. & Goh, C. P. (2006). A Two-step Method to Construct Credit Scoring Models with Data Mining Techniques. *International Journal of Business and Information*, *1*(1), 96-118.
48. Kraus, A. (2014). *Recent Methods from Statistics andMachine Learning for Credit Scoring* (doktorska desertacija). München: Ludwig-Maximilians-Universität München.
49. Lakshmanan, S. (2019, 17. maj). *How, When and Why Should You Normalize / Standardize / Rescale Your Data?* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://medium.com/@swethalakshmanan14/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff
50. Lateef, Z. (2019, 22. maj). *A Comprehensive Guide To Random Forest In R* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://www.edureka.co/blog/random-forest-classifier/
51. Lau, S., Gonzalez, J. & Nolan, D. (brez datuma). *Principles and Techniques of Data Science*. Pridobljeno 10. aprila 2020 iz https://www.textbook.ds100.org/
52. Lessmann, S., Baesens, B., Seow, H.-V. & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research, 247*(1), 124-136.
53. Lewinson, E. (2018, 2. julij). *Outlier Detection with Isolation Forest* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e
54. Li, C. (2013). *Little's Test of Missing Completely at Random.* Evanston: Northwestern University.
55. Li, X.-L. & Zhong, Y. (2012). An Overview of Personal Credit Scoring: Techniques and Future Work. *International Journal of Intelligence Science, 2*(4), 181-189.
56. Liu, C. Y. (2011). *Parameter Uncertainty in Credit Risk Portfolio Models* (magistrsko delo). Amsterdam: Vrije Universiteit Amsterdam.
57. Liu, F. T., Ting, K. M. & Zhou, Z.-H. (2009). *Isolation Forest.* Victoria: Gippsland School of Information TechnologyMonash University.
58. Lu, B. (2019, 24. april). *Demystifying the Correlation Matrix* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://www.datadriveninvestor.com/2019/04/24/demystifying-the-correlation-matrix/
59. Lund, A. (2015). *Two-Stage Logistic Regression Models forImproved Credit Scoring* (magistrsko delo). Stockholm: Royal Institute of Technology.
60. Lyn, T. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*(2), 149-172.
61. Mack, C., Su, Z. & Westreich, D. (2018). *Managing Missing Data in Patient Registries: Addendum to Registries for Evaluating Patient Outcomes: A User's Guide.* Rockville: Agency for Healthcare Research and Quality.
62. Mageto, D. K., Mwalili, S. M. & Waititu, A. G. (2015). Modelling of Credit Risk: Random Forests versus Cox Proportional Hazard Regression. *American Journal of Theoretical and Applied Statistics*, *4*(4), 247-253.

63. Matheson, R. (2018, 20. september). *Reducing false positives in credit card fraud detection*. Pridobljeno 20 aprila 2020 iz http://news.mit.edu/2018/machine-learning-financial-credit-card-fraud-0920
64. May, J. (2017, 23. maj). *The CRISP-DM Methodology*. Pridobljeno 10. aprila 2020 iz https://datashoptalk.com/crisp-dm-methodology/
65. Mayer, M., Resch, F. & Sauer, S. (2017). Validating Point-inTime vs . Through-the-Cycle Credit Rating Systems.
66. Microsoft Corporation. (brez datuma). *LightGBM's documentation*. Pridobljeno 10. aprila 2020 iz https://lightgbm.readthedocs.io/en/latest/
67. Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill Education.
68. Molnar, C. (2020, 7. april). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* Pridobljeno 10. aprila 2020 iz https://christophm.github.io/interpretable-ml-book/
69. Morde, V. (2019, 8. april). *XGBoost Algorithm: Long May She Reign!* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
70. Mukid, M. A., Widiharih, T., Prahutama, A. & Rusgiyono, A. (2017). Credit scoring analysis using weighted k nearest neighbor. *Journal of Physics: Conference Series*.
71. Nocedal, J. & Wright, S. (2006). *Numerical Optimization.* New York: Springer-Verlag.
72. Nurlybayeva, K. & Balakayeva, G. (2013). Algorithmic Scoring Models. *Applied Mathematical Sciences, 7*(12), 571-586.
73. Oppermann, A. (2019, 6. oktober). *Evaluation Metrics in Data Science and Machine Learning*. Pridobljeno 10. aprila 2020 iz https://www.deeplearning-academy.com/p/ai-wiki-evaluation-metrics-in-data-science
74. Orth, W. (2011). Default Probability Estimation in Small Samples - With an Application to Sovereign Bonds. *Discussion Papers in Statistics and Econometrics*.
75. Ortl, A. (2016). *Practical aspects of developing and validating corporate probability of default model and the use of machine learning* (magistrsko delo). Ljubljana: Ekonomska fakulteta.
76. Paleologo, G., Elisseeff, A. & Antonini, G. (2010). Subagging for credit scoring models. *European Journal of Operational Research, 201*(2), 490-499.
77. Petropoulos, A., Siakoulis, V., Stavroulakis, E. & Klamargias, A. (2018). *A Robust Machine Learning approach for credit risk analysis of large loan level datasets using deep learning and extreme gradient boosting.* Basel: Irving Fisher Committee on central bank statistics.
78. Phan, W., Hall, P. & Whitson, K. (2016). *The Evolution of Analytics: Opportunities and Challenges for Machine Learning in Business.* Kalifornija: O'Reilly Media, Inc.
79. Pozzolo, A. D., Caelen, O., Waterschoot, S. & Bontempi, G. (2015, 20. januar). *Racing for unbalanced methods selection.* Pridobljeno 10. aprila 2020 iz https://www.slideshare.net/dalpozz/racing-for-unbalanced-methods-selection
80. Ray, S. (2018, 3. maj). *Improve Your Model Performance using Cross Validation* [objava na blogu]. Pridobljeno 10. aprila 2020 iz

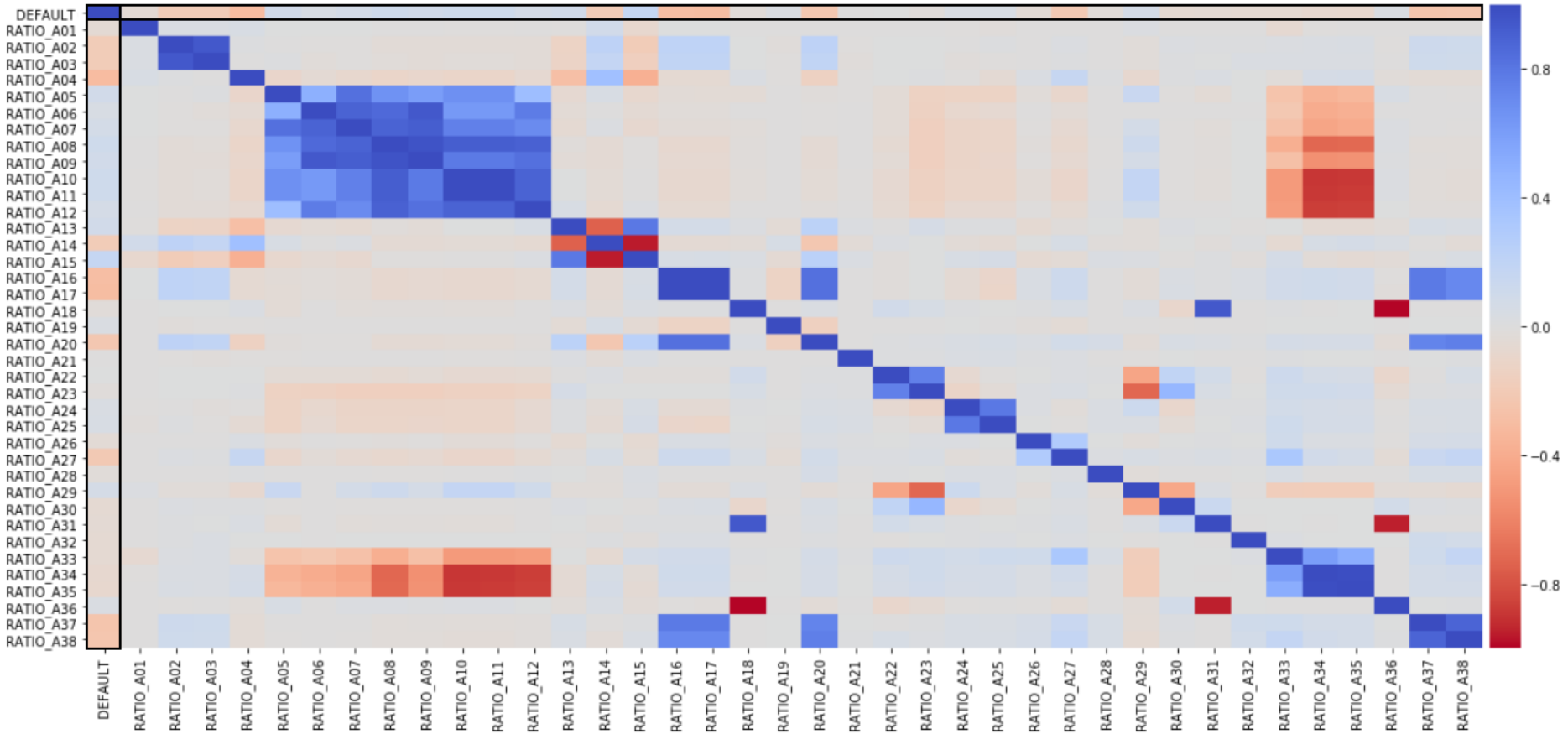
https://www.analyticsvidhya.com/blog/2018/05/improve-model-performance-cross-validation-in-python-r/
81. Roy, H. S. (2016). *Consumer Credit Scoring Using Logistic Regression and Random Forest.* Berunanpukuria: West Bengal State University.
82. Sadatrasoul, S. M., Gholamian, M., Siami, M. & Hajimohammadi, Z. (2013). Credit scoring in banks and financial institutions via data mining techniques: A literature review. *Jounral of AI and Data Mining, 1*(2), 119-129.
83. Salcedo, J. & McCormick, K. (2017). *IBM SPSS Modeler Essentials.* Packt.
84. Salvaire, P. (2019). *Explaining the predictions of a boosted tree algorithm : application to credit scoring* (magistrsko delo). Lizbona: NOVA IMS.
85. Sarlija, N., Benšić, M. & Bohacek, Z. (2006). *Customer revolving credit – how the economic conditions make a difference.* Osijek.
86. Sharma, D. (2009). *Guide to Credit Scoring in R.* Pridobljeno 10. aprila 2020 iz https://cran.r-project.org/doc/contrib/Sharma-CreditScoring.pdf
87. Shung, K. P. (2018, 15. marec). *Accuracy, Precision, Recall or F1?* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
88. Singh, H. (2018, 3. november). *Understanding Gradient Boosting Machines*[objava na blogu]. Pridobljeno 10. aprila 2020 iz https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab
89. Sundar, K. (2019, 3. julij). *Anomaly Detection Using Isolation Forest* [objava na blogu]. Pridobljeno 10. aprila 2020 iz https://lambda.grofers.com/anomaly-detection-using-isolation-forest-80b3a3d1a9d8
90. Šmid, M. (2002). *Uporaba metod za odkrivanje znanj iz podatkov v bančnem trženju* (diplomsko delo). Ljubljana: Fakulteta Za Računalništvo In Informatiko.
91. Šušteršič, M., Mramor, D. & Zupan, J. (2009). Consumer credit scoring models with limited data. *Expert Systems with Applications, 36*(3), 4736-4744.
92. Thomas, L. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*(2), 149-172.
93. Topp, R. & Perl, R. (2010). Through-the-Cycle Ratings Versus Point-in-Time Ratings and Implications of the Mapping Between Both Rating Types. *Finance Markets Institutions & Instruments, 19*(1).
94. Tsai, C.-F. & Wu, J.-W. (2008). Using neural network ensembles for bankruptcyprediction and credit scoring. *Expert Systems with Applications, 34*(4), 2639–2649.
95. University of Pennsylvania. (brez datuma). *Multiple Imputation using Additive Regression, Bootstrapping, and Predictive Mean Matching*. Pridobljeno 10. aprila 2020 iz: http://finzi.psych.upenn.edu/R/library/Hmisc/html/aregImpute.html
96. Van Der Maaten, L., Postma, E. & Van den Herik, J. (2009). Dimensionality reduction: a comparative review.
97. Venter, E. S. (2016). *Probability of Default Calibration for Low Default Portfolios: Revisiting the Bayesian Approach.* Matieland: University of Stellenbosch.

98. Volk, M. (2012). Estimating probability of default and comparing it to credit rating classification by banks. *Economic And Business Review, 14*(4), 299-320.
99. Wang, G., Hao, J., Ma, J. & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications, 38*(1), 223-230.
100. Wirth , R. (2000). CRISP-DM: Towards a standard process model for data mining. V *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining* (str. 29-39). London: Springer-Verlag.
101. Zhang, A. (2009). *Statistical Methods in Credit Risk Modeling* (doktorska desertacija). Michigan: The University of Michigan.
102. Zhang, D., Zhou, X., Leung, S. C. & Zheng, J. (2010). Vertical bagging decision trees model for credit scoring. *Expert Systems with Applications, 37*(12), 7838–7843.

PRILOGE

Priloga 1: Povečana korelacijska matrika osnovne podatkovne zbirke

Slika 1: Korelacijska matrika osnovne podatkovne zbirke



Vir: Lastno delo.

Priloga 2: Opisna statistika prečiščene podatkovne zbirke

Tabela 1: Opisna statistika prečiščene podatkovne zbirke

	Število vrstic	Mediana	Standardni odklon	Minimalna vrednost	Maksimalna vrednost
Kazalnik 02	1574	-0.17073	1	-1.26442	18.78884
Kazalnik 03	1574	-0.17122	1	-1.32459	13.16932
Kazalnik 05	1574	-0.08289	1	-0.10751	37.62661
Kazalnik 06	1574	-0.13461	1	-0.18332	25.02133
Kazalnik 07	1574	-0.09509	1	-0.13151	36.73457
Kazalnik 08	1574	-0.14922	1	-0.17872	19.49783
Kazalnik 09	1574	-0.18678	1	-0.21344	21.36755
Kazalnik 10	1574	-0.09621	1	-0.11513	22.09447
Kazalnik 11	1574	-0.09644	1	-0.11665	22.09900
Kazalnik 12	1574	-0.09517	1	-1.31956	32.63638
Kazalnik 13	1574	-0.10282	1	-1.24354	2.46175
Kazalnik 14	1574	0.03499	1	-2.02139	1.70441
Kazalnik 15	1574	-0.05162	1	-1.57100	2.12925
Kazalnik 16	1574	0.0715	1	-26.45041	0.31488
Kazalnik 17	1574	0.07406	1	-26.45044	0.31019
Kazalnik 20	1574	0.08857	1	-36.26910	0.29581
Kazalnik 23	1574	0.05237	1	-18.94366	6.21064
Kazalnik 24	1574	-0.03	1	-14.92137	6.80211
Kazalnik 25	1574	-0.05443	1	-14.15370	5.40898
Kazalnik 29	1574	-0.16407	1	-0.40578	10.33996
Kazalnik 30	1574	0.0071	1	-4.85840	12.51650
Kazalnik 33	1574	0.0782	1	-21.53644	1.66716
Kazalnik 34	1574	0.19782	1	-13.12349	4.37566
Kazalnik 35	1574	0.14671	1	-36.20644	0.78140
Kazalnik 37	1574	0.23844	1	-10.64270	1.48929
Kazalnik 38	1574	0.21177	1	-11.36806	1.48863

Vir: Lastno delo.