Master rad

Modeliranje kreditnog rizika primenom gradijentnog pojačavanja

Milica Sarić

Mentor:
Bojana Milošević

Matematički fakultet
Univerzitet u Beogradu

Sadržaj

```
O scoring, who art in regression,
Guessing be thy name.
Thy assumptions come,
Thy will be done in future as it was in the past,
Give us this day our expected bad rates,
And forgive us our lousy model weights,
As we forgive those who supply us with poor data.
Lead us not into write-offs,
And deliver us from the auditors.
For thine is the #NAME, the #DIV/0,
and the #VALUE!
Forever and ever, Amen.
```

Preuzeto iz [5]

1 Uvod

U ovom radu bavićemo se problemom modeliranja kreditnog rizika. Termin kreditni rizik obuhvata više stavki, a fokus u ovom radu će biti na oceni verovatnoće da klijent neće vratiti kredit. Određivanje kreditnog rejtinga izdvaja se kao posebna oblast (eng. credit scoring). Da bismo bliže razumeli šta je cilj ove oblasti, Anderson [5] predlaže da pogledamo poreklo reči. Kredit danas označava princip "kupi sada, plati kasnije", a sama reč potiče od latinske reči credo koja znači verovati. Dok reč rejting označava ocenu određenog kvaliteta radi uspostavljanja poretka.

U poglavlju Opis problema ćemo se upoznati sa ekonomskom pozadinom problema kreditnog rizika i zašto je danas toliko zastupljena. Pri radu sa podacima, često dosta više vremena oduzima sama priprema i čišćenje podataka od izbora modela. Poglavlje 3 je posvećeno ovoj temi. Nakon toga su opisani algoritmi koji se mogu koristiti za procenu verovatnoće difolta. Poglavlje 6 bavi se bitnom temom interpretabilnosti modela. Ispitujemo da li postoji način da se prevaziđe neinterpretablinost visoko prediktivnih modela. Poslednje poglavlje posvećeno je upoređivanju rezultata i upoznavanju sa različitim metrikama za evaluaciju modela.

Kod korišćen za dobijanje rezultata može se naći na sledećoj lokaciji https://github.com/milicasaric/Master-rad.

2 Opis problema

Zbog nestabilnosti finansijskog tržišta, polje kreditnog rizika značajno je unapređeno u prethodnim godinama. Banke mogu da pozajmljuju novac klijentima jedino ukoliko klijenti dovoljno veruju u stabilnost bankarskog sistema da bi ostavili depozit. Stoga je neophodno sprečiti kolaps bankarskog sistema, takođe, kolaps bankarskog sistema doveo bi do mnogo većih ekonomskih posledica. Zbog svega navedenog, bankarstvo je poznato kao visoko regulisana oblast. Uloga regulatora je da obezbedi stabilnost tržišta ograničavanjem rizika koje banke mogu da podnesu. Uvođenje Bazel 2 standarda 2004. godine omogućilo je bankama samostalno razvijanje modela za procenu kreditnog rizika.

Banke dobijaju keš iz različitih izvora. Prvi izvor je štednja, npr. penzioni fondovi, drugi izvor su deoničari ili investitori, oni kupuju vlašnistvo nad bankom, ako banka ima profit ulog im se vraća kroz dividende. Štednja i deoničari su pasiva banke, aktiva je novac koji banka dobija kroz investicije, investicija je pozajmljivanje novca kroz kredite. Drugi tip investicije su hartije od vrednosti kao sto su obveznice, akcije, opcije itd. Bitno je da ove investicije uvek nose rizik sa sobom. Događaj koji je nepovoljan po banku je da klijent ne vrati pozajmljeni novac. U tom slučaju govorimo o difoltu (eng. default), a klijenta nazivamo difolterom. Ako govorimo o tržištu hartija od vrednosti, banka ima rizik od kolabiranja tržišta tj. smanjenja vrednosti hartijama. Zbog socijalnog značaja banke moraju biti zaštićene od rizika kojima su izložene. Insolventnost banke se mora izbeći po svaku cenu i rizik koji banka preuzima na strani aktive mora biti kompenzovan na strani pasive. Stoga, banka mora imati dovoljno kapitala kao rezervu. Mora postojati direktna veza između kapitala i rizika.

Prvi bazelski standard je uveden 1988. godine sa ciljem da postavi minimalne regulatorne kapitalne zahteve da bi se obezbedilo da banka u svakom trenutku može da isplati depozite. Bazel 1 se najvecim delom fokusira na kreditni rizik i uvodi Kukov odnos (eng. Cook ratio) koji predstavlja odnos kapitalnog bafera i imovine ponderisane rizikom (RWA risk-weighted asset). Donja granica je postavljena na 8%. Drugim rečima kapital mora biti veći od 8% imovine ponderisane rizikom. Postoje 3 pristupa modeliranju kreditnog rizika:

- Standardizovani pristup;

- Osnovni pristup interne ocene rejtinga (eng. internal rating-based IRB);
- Napredni pristup interne ocene rejtinga.

Pristupi se razlikuju u pogledu sofisticiranosti i fleksibilnosti u proceni rizika. IRB pristup se zasniva na proceni 4 ključna parametra:

- PD - verovatnoća neizvršenja obaveza u narednih godinu dana, verovatnoća difolta (eng. probability of default);
- EAD - izloženost u trenutku difolta, tj. preostao dug, izražava se u valuti kredita (eng. exposure at default);
- LGD - gubitak u slučaju difolta, tj. odnos gubitka u odnosu na preostali dug, izražava se u procentima (eng. loss given default);
- M $f(M)$- ročnost kredita, koristi se za usklađivanje preostalog roka otplate kredita kod pravnih lica.

Cilj je proceniti očekivani gubitak po klijentu ECL (eng. expected client's loss). Da bi se to postiglo procenjuju se tri parametra PD (eng. probability of default), LGD (eng. loss given default) i EAD (eng. exposure at default).

Primer 1. *Neka je* $EAD = 10000$*,* $LGD = 20\%$*. To znači da će u slučaju difolta gubitak biti 2000 evra. Neka je verovatnoca difolta* $PD = 1\%$*, tada je očekivani gubitak* $ECL = 20$*. Drugim rečima, očekivani klijentski gubitak je proizvod 3 parametra:*

$$ECL = PD \cdot LGD \cdot EAD \cdot f(M). \tag{1}$$

Između navedenih parametara postoji pozitivna korelacija koja najčešće nije obuhvaćena modelima. Korelacija je obično izraženija u periodima ekonomske krize. Cena imovine opada pa je verovatnije da će klijenti odustati od plaćanja što dovodi do porasta i PD-a i LGD-a. EAD raste jer klijenti koriste sve dostupne linije kredita. EAD se najčešće razdvaja na povučen (eng. drawn) i nepovučen (eng. undrawn) iznos.

$$EAD = d \cdot EAD_d + u \cdot EAD_u. \tag{2}$$

Komponente EAD-a su iznosi u trenutku difolta i u trenutku godinu dana pre difolta, pa je formula sledeća:

$$EAD_d = \frac{min(d_t, d_{t-1})}{d_{t-1}}, \tag{3}$$

$$EAD_u = min(1, \frac{max(0, d_t - d_{t-1})}{u_{t-1}}). \tag{4}$$

Pri izračunavanju LGD-a, koriste se dva pristupa:

- korišćenjem podataka o tokovima novca nakon difolta;
- korišćenjem tržišnih podataka o vrednosnim papirima, ovaj pristup nije moguć kod fizičkih lica.

Poslednji parametar ročnost kredita se najčešće zanemaruje jer je uticaj najčešće vrlo mali, a deo rizika je prepoznat kroz EAD i LGD kalkulaciju. Teorijski gledano moguće je koristiti statističke modele za procenu EAD-a i LGD-a, ali zbog nedostatka pravovremenih podataka o naplaćenim iznosima nakon difolta, banke najčešće nisu u mogućnosti da razviju adekvatne modele [5].

U ovom radu će fokus biti na modeliranju verovatnoće difolta. Modeli za procenu verovatnoće difolta se razlikuju u pogledu segmenta na koji se primenjuju, tj. da li procenjuju difolt fizičkih lica, malih i srednjih preduzeća ili velikih kompanija. U zavisnosti od segmenta, različiti ulazni podaci se koriste i time svaki segment ima svoje specifične probleme. Za pravna lica, glavni izbor podataka su bilansi stanja i uspeha te stoga ne očekujemo greške uzrokovane manuelnim unošenjem podataka. Takođe, razlike u modelima dolaze i iz različitog broja podataka, kod fizičkih lica, očekivano, govorimo o mnogo većem broju opservacija. Kod velikih kompanija, osim što imamo dosta manje opservacija, proporcionalno imamo i manje difoltera, pa često možemo imati problem sa nedovoljno difoltera da bismo uopšte razvili model. S obzirom da radimo sa izrazito nebalansiranim skupom podataka, treba biti oprezan pri evaluaciji modela kako bismo bili sigurni da smo izabrali model koji zaista najbolje identifikuje difoltere.

Kod fizičkih lica možemo napraviti dalju podelu modela u zavisnosti od njihove istorije u banci. Ukoliko je fizičko lice već duže vreme kreditni klijent banke, banka o njemu ima dovoljno informacija na osnovu kojih može da proceni da li je klijent pogodan za još jedan kredit. U ovom slučaju govorimo

o bihevioralnom skoringu, što znači da koristimo podatke o stanju računa, kašnjenjima pri izmirenju obaveza itd. da bismo procenili rizik. Ako je pak reč o novom klijentu koji zahteva kredit, pri proceni njegovog rizika možemo se služiti jedino podacima iz aplikacione forme (broj godina, obrazovanje, dužina staža...) i podacima iz Kreditnog biroa. U ovom slučaju govorimo o aplikativnom modelu. Pri odobrenju novog kredita postojećem klijentu, u obzir ćemo uzeti i aplikativni i bihevioralni model, tako da je finalna predikcija rezultat kombinacije dva modela.

Za eksperimentalni deo korišćeni su podaci sa sajta Kaggle [3]. Korišćena su dva različita uzorka kako bi se uporedile performanse algoritma na različitim skupovima. Jedan uzorak ima dosta manji broj opservacija, što ga čini interesantnim za ispitivanje performansi algoritma.

Kompanija Home Credit je ustupila svoje podatke za Kaggle takmičenje gde je prvo mesto bilo nagrađeno sa čak 35000 dolara. Home Credit je kompanija sa sedištem u Holandiji koja se bavi kreditiranjem fizičkih lica. Najveći broj kredita se plasira putem kredita na licu mesta, tzv. POS krediti (eng. point of sale) [2]. U pitanju su krediti koji se odobravaju u prodavnici pri samoj kupovini. Ovakav način kreditiranja predstavlja najefikasniji način za akviziciju novih klijenata. Ovaj uzorak ima dosta više opservacija i podataka o klijentima pa će veći deo rada biti posvećen ovom uzorku.

Drugi uzorak je German Credit Data [1] koji se često koristi u statističkim radovima, a zasluge za ovaj uzorak pripadaju profesoru Hansu Hofmanu.

3 Pretpocesiranje podataka

Pretprocesiranje podataka je jedan u najbitnijih koraka u razvoju modela i često se spominje da priprema podataka oduzima 80% vremena. Osnovni koraci u pretprocesiranju, koji su nam neophodni da pripremimo modele koji su dalje opisani, su čišćenje, transformacija, integracija, redukcija podataka... U zavisnosti od skupa podataka i problema koji rešavamo biramo odgovarajuće tehnike pripreme podataka.

3.1 Kreiranje varijabli

Uzorak Home Credit se sastoji iz 307511 redova, gde svaki red predstavlja jedan kreditni zahtev. Podaci su podeljeni u 7 fajlova:

- *application_train* - podaci o apliciranju, sadrži i ciljnu varijablu sa vrednostima 0 ili 1;
- *bureau* - informacije iz Kreditnog biroa, podaci o klijentovim zaduženjima u drugim bankama;
- *burea_balance* - mesečni izvod iz Kreditnog biroa, klijentovo ponašanje iz meseca u mesec;
- *previous_application* - informacije o prethodnim apliciranjima klijenata u Home Credit, ukoliko ih je bilo;
- *pos_cash_balance* - mesečni podaci o prethodnim POS kreditima u Home Credit;
- *credit_card_balance* - mesečni podaci o prethodnim kreditnim karticama u Home Credit;
- *installments_payment* - informacije o mesečnim plaćanjima prethodnih kredita u Home Credit.

Podaci dolaze iz različitih izvora i nisu na istom nivou granularnosti. Neki podaci su na nivou klijenta (podaci Kreditnog biroa), a neki su na nivou zahteva (podaci o apliciranju), takođe, imamo i mesečne podatke o

klijentu. Da bismo mogli da koristimo podatke, prvo moramo da napravimo smislene varijable, tj. podatke moramo agregirati. Postoji više načina za agregaciju, možemo uzeti prosečnu vrednost, sumu, maksimalnu vrednost... Ne možemo, a priori, znati koja agregacija će nam dati najbolju vrednost, te ćemo napraviti više varijacija, a zatim daljim analizama doći do konačnih varijabli koje će ući u model. U tabelama *pos_cash_balance*, *credit_card_balance* i *installments_payment* podaci su na mesečnom nivou po zahtevima. Tako da ako je klijent u prošlosti imao dva kredita, za svaki mesec ćemo imati dva reda za jednog klijenta. Da bismo dobili jedinstvenu varijablu po klijentu, prvo moramo agregirati na nivou proizvoda, a zatim na nivou klijenta. U oba slučaja biramo koju agregatnu funkciju ćemo koristiti, tako da se broj potencijalnih varijabli uvećava eksponencijalno. Primera radi, posmatrajmo kolonu *AMT_DRAWINGS_ATM_CURRENT* u tabeli *credit_card_balance*, koja predstavlja iznos podignut na bankomatu u toku meseca. Ako koristimo srednju vrednost, dobijamo prosečan iznos podignut na bankomatu po kreditnoj kartici. Ukoliko klijent ima dve kreditne kartice, moramo primeniti još jedno agregiranje. Neka to ovaj put bude maksimum. Tada bi naša konačna varijabla bila maksimalna prosečna vrednost koju klijent podiže na bankomatu na mesečnom nivou. Nakon svih grupisanja dobijamo čak 1544 varijabli koje ćemo koristiti u daljim analizama.

3.2 Tretman nedostajućih vrednosti

Prilikom primene modela koji procenjuju kreditni rizik klijenata, nailazi se na razne probleme sa podacima. Razmotrimo primer fizičkih lica koji apliciraju za kreditnu karticu. Prilikom podnošenja zahteva njihovi podaci se u sistem najčešće unose manuelno. Ukoliko program koji službenici koriste ne uzima u obzir sve situacije zbog kojih može doći do netačnih podataka, vrlo je verovatno da će se u sistemu naći dosta netačnih podataka ili će dosta podataka faliti. Npr. službenik može uneti jednu nulu više pa bismo mogli da za varijablu broj godina imamo vrednost 300. Ovo je očigledno nemoguće i to bismo mogli da primetimo prilikom pretprocesiranja i da takve podatke uklonimo. Međutim, ukoliko bi bankarski službenik uneo jednu 0 više za varijablu plata, pa tako umesto 30000 dobijemo 300000, ne bismo sa sigurnošću mogli da tvrdimo da je taj podatak netačan. Takođe, ako nam

neki podatak nedostaje, u nekim situacijama je opravdano pretpostaviti da je nedostatak podataka informacija po sebi (npr. varijabla broj dece) i zameniti ga nulom, dok u drugim situacijama to nije moguće (npr. broj godina).

Ne postoji idealan način za tretman nedostajućih podataka, neke tehnike će se pokazati bolje pri određenim problemima, a neke lošije. Neki od načina na koji se tretiraju podaci koji nam nedostaju su brisanje redova, zamena srednjom vrednošću ili medijanom, procena podataka regresijom itd. Više o različitim načinima tretiranja nedostajućih podataka može se pročitati u [15]. Pristup koji izaberemo moramo primeniti i na budućim podacima, tj. pri apliciranju klijenata, te odbacivanje klijenta zbog nedostatka podatka nije prihvatljivo rešenje. Takođe, zbog nebalansiranosti podataka izbacivanjem opservacija možemo dodatno pogoršati odnos klasa. Sa druge strane, varijable koje imaju nizak procenat popunjenosti možemo već na početku analiziranja odbaciti i zadržati varijable koje nam daju više informacija. Pristup koji ćemo slediti u ovom radu je zamena nedostajućih vrednosti medijanom.

3.3 Problem nebalansiranosti podataka

Uzorak smatramo nebalansiranim ako klase nisu jednako raspodeljene. Nebalansirani uzorci se često sreću u telekomunikaciji (otkrivanje prevara), klasifikaciji teksta, prepoznavanje slika (npr. otkrivanje izlivanja nafte na satelitskim snimcima). Performanse algoritama se najčešće ocenjuju korišćenjem tačnosti (eng. accuracy). Međutim, ovakav način evaluacije algoritma nije prikladan kada se radi sa nebalansiranim skupom podataka ili kada posledice klasifikacije nisu iste (primer klasifikacija kancerogenih piksela). Postoje dva načina za tretman neizbalansiranog skupa podataka. Jedan je da se dodele različite težine klasama, a drugi je da se od polaznog uzorka napravi novi uzorak (eng. resampling) tako što se poveća udeo manjinske klase (eng. oversampling) ili tako što se smanji udeo većinske klase (eng. undersampling).

Kod kreditnog rizika se često susreće pojam portfolija sa malo difoltera (eng. low default portfolio). Ne postoji formalna definicija portfolija sa malo difoltera. Banka Engleske koristi broj 20 kao prag materijalnosti takvih portfolija (videti [6]). Ukoliko naš portfolio sadrži manje od 20 difoltera radimo sa portfoliom sa malo difoltera.

Poduzorkovanje (eng. undersampling) i preuzorkovanje (eng. oversam-

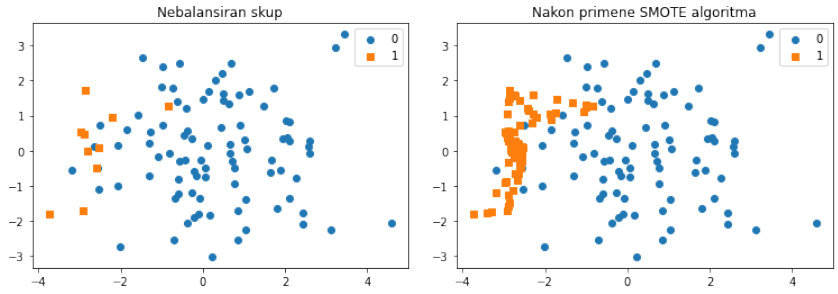
pling) su dva popularna načina za tretman portfolija sa malo difoltera. Oba se primenjuju na trening skupu, ali ne i na skupu za testiranje. Ideja poduzorkovanja je da se uklone dobri klijenti iz trening skupa, dok je ideja preuzorkovanja da se repliciraju loši klijenti. Cilj oba pristupa je da imamo manju razliku, recimo umesto 99% dobrih i 1% loših klijenata, imamo 90% napram 10%.

Algoritam preuzorkovanja koji se često koristi se naziva SMOTE algoritam (eng. the synthetic minority over sampling technique). Algoritam je osmislila grupa autora i objavila ga 2002.godine u radu [10]. Manjinska klasa se uvećava kreiranjem sintetičkih opservacija. U prvom koraku SMOTE algoritma, za svaku instancu manjinske klase određuje se k najbližih suseda u euklidskom smislu. Iz skupa svih instanci, biramo k koje imaju najmanje rastojanje od instance manjinske klase koju želimo da uvećamo. Kao meru rastojanja između opservacija x_i i x_0 koristimo:

$$d_i = \sqrt{\sum_{j=1}^{p}(x_{ij} - x_{0j})^2}, \tag{5}$$

gde je x_0 instanca manjinske klase, a x_i neka instanca iz skupa. U zavisnosti od odnosa klasa koji trenutno imamo i odnosa koji želimo da postignemo, biramo koliko najbližih suseda ćemo posmatrati. U slučaju da želimo da manjinsku klasu uvećamo za 200%, koristili bismo $k = 2$. Najčešće se koristi $k = 5$. Drugi korak je generisanje sintetičkih opservacija na sledeći način: uzima se razlika između varijabli instance manjinske klase i njenih najbližih suseda. Ova razlika se množi nasumičnim brojem između 0 i 1 i dodaje se u uzorak. Na ovaj način je dodat primer manjinske klase i time povećana frekvencija manjinske klase. Pre primene algoritma k najbližih suseda neophodno je sprovesti standardizaciju varijabli.

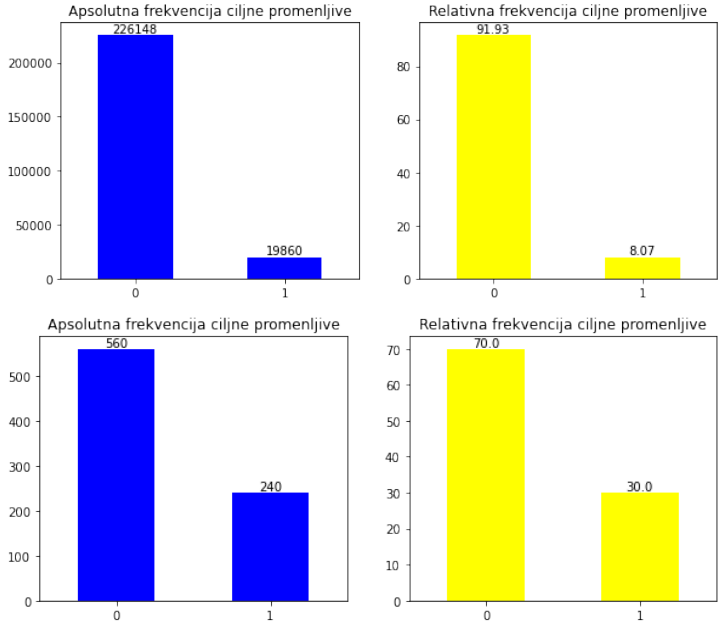
Gore opisani algoritam je primenljiv samo na numeričkim varijablama. Osmišljena je varijacija SMOTE algoritma koja se naziva SMOTE-NC, a koja se može koristiti na mešovitim skupovima [10]. Prvi korak je računanje medijane standardnih odstupanja svih numeričkih varijabli na manjinskoj klasi. Ako se nominalna varijabla razlikuje od svog suseda, tu razliku ćemo kazniti medijanom odstupanja na numeričkim varijablama. Nakon toga pravimo sintetičke opservacije na osnovu k najbližih suseda. Kod numeričkih varijabli sprovodimo isti postupak kao u slučaju osnovnog SMOTE algoritma, dok za kategoričke varijable sintetičku opservaciju popunjavamo većinskom vrednošću (modom) date varijable među najbližim susedima.



Slika 1: Primer SMOTE algoritma

Na slici 1 iznad vidimo primer nebalansiranog skupa sa leve strane, a sa desne strane skup nakon primene SMOTE algoritma, gde vidimo da je udeo manjinske klase znatno uvećan.

Pogledajmo odnos difoltera u našim uzorcima na slici 2.



Slika 2: Odnos klasa (gore Home Credit, dole German Credit)

Nakon primene SMOTE algoritma odnos klase je izjednačen.

3.4 Transformacije varijabli

Algoritam koji nam daje vrlo elegantno rešenje gore opisanog problema sa nedostajućim podacima, naziva se WoE (eng. weight of evidence) transformacija [27, 7]. Cilj ove transformacije je podela varijable na grupe, odnosno, podintervale, tako da razlika u procentima difoltera među grupama bude što vidljivija. Ovim pristupom rešili smo više problema poput nedostajućih podataka, ekstremnih vrednosti (odudarajuće vrednosti, eng. outliers), interpretablinosti modela.

WoE transformacija grupe i je sledeća:

$$WoE_i = ln \frac{\frac{dobar_i}{\sum_{i=1}^{n} dobar_i}}{\frac{los_i}{\sum_{i=1}^{n} los_i}} \tag{6}$$

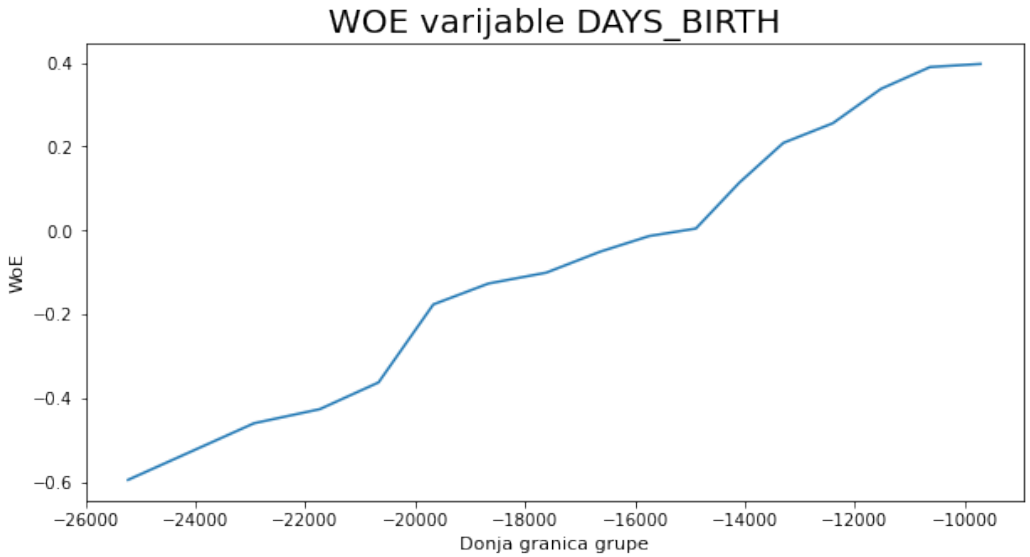
Pri čemu je vrednost *dobar* 1 ukoliko je klijent uredan, a 0 ako nije, a vrednost *los* 0 ukoliko je klijent uredan, a 0 inače. Drugim rečima, parametar WoE nam govori da li je u nekoj grupi veća relativna frekvencija dobrih klijenata ili frekvancija loših klijenata u odnosu na ceo uzorak. Pozitivne vrednosti označavaju da je veća relativna frekvencija dobrih klijenata, a negativne vrednosti ukazuju na veću frekvenciju loših klijenata. Vrednosti bliske nuli ne daju nam mnogo informacija o grupi.

Sama implementacija algoritma se sastoji u tome da prvo sortiramo varijablu. Nakon što smo je sortirali, varijablu ćemo podeliti u n grupa iste veličine. Sada možemo svakoj grupi dodeliti vrednost WoE primenom formule. Pored numeričkih varijabli, WoE transformacija se vrlo lako može primeniti i na kategoričke varijable bez prethodne transformacije. Svaka kategorija je grupa za sebe (ili u slučaju dosta različitih kategorija, možemo grupisati više kategorija u jednu grupu) i dobija svoju WoE vrednost. Na ovaj način je tretman nedostajućih vrednosti olakšan, jer oni mogu biti grupa za sebe. Takođe, grupisanjem vrednosti, ekstremne vrednosti gube na značaju i ne zahtevaju poseban tretman.

Primenom WoE transformacije olakšava se i interpretacija modela. Nakon transformacije koristi se samo izračunata WoE vrednost, što znači da varijabla ima onoliko različitih vrednosti za koliko grupa smo se odlučili.

Sada kada smo se odlučili za WoE transformaciju treba izabrati varijable koje su pogodne za model. Ranije smo pomenuli važnost ekonomskog smisla modela, s tim u vezi, jedan od zahteva je da varijabla nakon WoE transformacije ima monotonu vezu sa stopom difolta. Na taj način je verovatnije da će se

eksperti složiti sa izborom varijable jer možemo na jednostavan način izvući ekonomski smisao varijable. Pogledajmo kako izgleda raspodela varijable nakon primene WoE transformacije. Na slici 3 je varijabla $DAYS\_BIRTH$, na x-osi je donja granica grupe, a na y-osi je WoE vrednost za tu grupu. Vidimo da sa porastom vrednosti varijable, raste i WoE vrednost, pa možemo zaključiti da je varijabla monotona.



Slika 3: WOE varijable DAYS_BIRTH

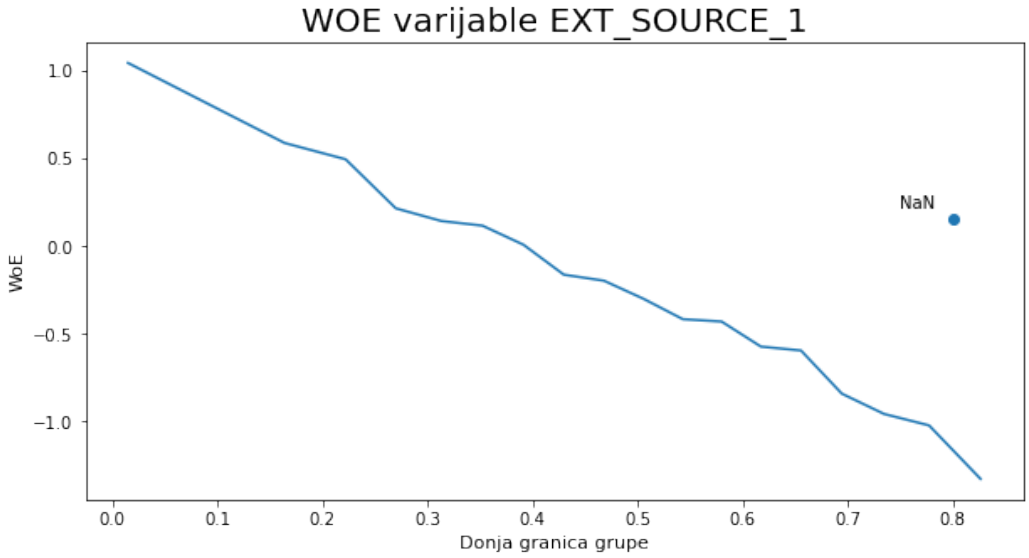
Pogledajmo primer još jedne varijable na slici 4, u pitanju je eksterni skor. I ona je većim delom monotona, međutim, WoE vrednost poslednje grupe ne prati trend prethodnih grupa. Ipak, ovo ne predstavlja problem, jer se u toj grupi nalaze samo vrednosti NaN, pa je tako i ova varijabla monotona.

Uvedena je i kvantitativna mera koja nam govori o prediktivnosti varijable u slučaju kada koristimo WoE transformaciju. Reč je informativnoj vrednosti varijable (information value). Računa se po sledećoj formuli:

$$IV = \sum_{i=1}^{n} \left(\frac{dobar_i}{\sum_{i=1}^{n} dobar_i} - \frac{los_i}{\sum_{i=1}^{n} los_i} \right) * WoE_i \tag{7}$$

Uspostavljena je sledeća skala za procenu prediktivnosti varijable na osnovu informativne vrednosti [5]:

- $IV \leq 0.05$: neprediktivna varijabla;



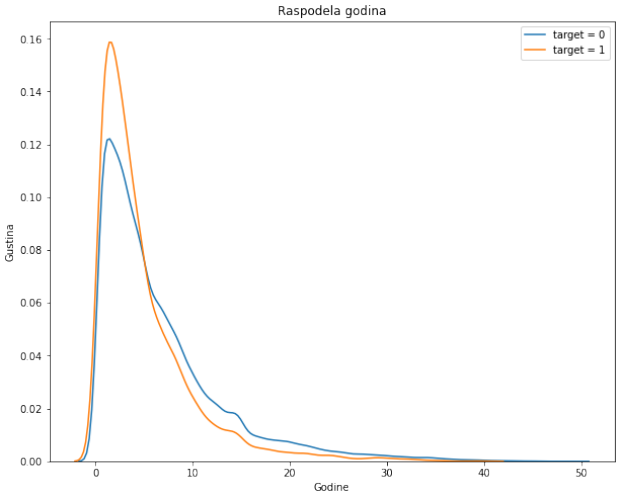
Slika 4: WOE varijable EXT_SOURCE_1

- $0.05 \leq IV \leq 0.1$: nisko prediktivna varijabla;
- $0.1 \leq IV \leq 0.25$: srednje prediktivna varijabla;
- $IV \geq 0.25$: visoko prediktivna varijabla.

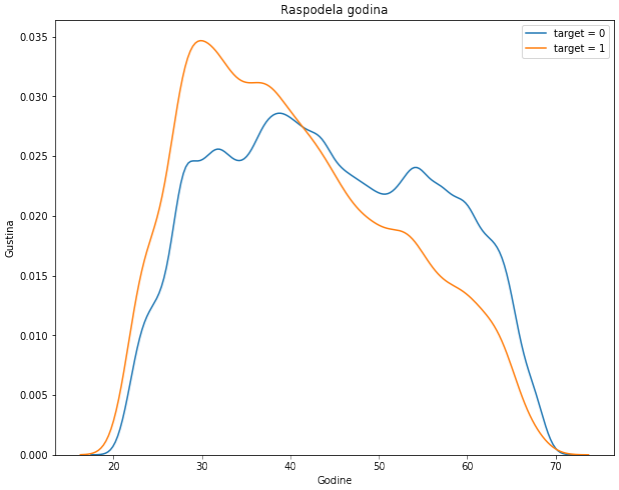
Grafički možemo ispitati smislenost varijable, time što ćemo nacrtati gustinu raspodele u obe klase. Ukoliko bi gustine bile iste to bi značilo da varijabla ne pruža nikakvu diskriminaciju klasa i ne bi imalo smisla koristiti je.

Na slici 5 je varijabla koja predstavlja dužinu staža, vidimo da difoteri mahom imaju kraći staž.

Nešto drugačija raspodela je kod starosti klijenta na slici 6, ali zaključak je sličan, a to je da su mlađe osobe, a posledično osobe sa kraćim stažem, dosta rizičniji klijenti.



Slika 5: Gustina raspodele varijable DAYS_EMPLOYED po klasama



Slika 6: Gustina raspodele varijable DAYS_BIRTH po klasama

3.5 Multikolinearnost

Kako smo kreirali veliki broj varijabli primenama različitih funkcija agregiranja, očekivano je da će dosta varijabli biti dosta međusobno korelisano. Da bismo izbegli probleme koji mogu nastati usled prisustva multikolinearnosti, izvršićemo selekciju varijabli i sačuvati manji skup varijabli koje međusobno nisu korelisane. Želimo da sačuvamo one varijable koje najviše doprinose modelu. Kao prvi korak izračunate su F-statistike za sve varijable i zadržano 25% najboljih. Nakon toga su izbačene visoko korelisane varijable.

Postojanje multikolinearnosti se lako detektuje korišćenjem koeficijenata tolerancije i VIF (faktor inflacije varijanse eng. variance inflation factor). Tolerancija varijable se definiše kao procenat varijanse koji se ne može objasniti ostalim prediktorima. VIF definišemo kao recipročnu vrednost tolerancije.

$$VIF = \frac{1}{1 - R^2} \tag{8}$$

R^2 predstavlja proporciju disperzije zavisne promenljive koja je objašnjena nezavisnom promenljivom. Smatra se da vrednost VIF koeficijenta iznad 10 ukazuje na postojanje multikolinearnosti [26]. Kod kategoričkih varijabli korišćen je χ^2-test za ispitivanje postojanja statističke povezanosti među varijablama.

4 Pregled metoda koji se mogu koristiti za klasifikaciju

U ovom poglavlju ćemo se upoznati sa metodama koje se najčešće koriste kod problema klasifikacije. Počinjemo od najjednostavnijeg modela logističke regresije. Zatim uvodimo pojam stabla odlučivanja i slučajne šume, da bismo na kraju definisali gradijentno pojačavanje koje zauzima centralno mesto u ovom radu.

4.1 Logistička regresija

Primena logističke regresije je u današnje vreme vrlo rasprostranjena, od biologije, medicine, lingvistike, pa do finansija i kriminologije. Interpretablinost i jednostavno treniranje modela su ono što čini logističku regresiju pogodnom za primene u različitim industrijama.

Da bismo definisali logističku regresiju moramo poći od linearne regresije oblika:

$$E(Y|(x_1,\ldots,x_p)) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p, \tag{9}$$

pri čemu je p broj varijabli (prediktora), a $E(Y|(x_1,\ldots,x_p))$ očekivanje zavisne varijable Y pri datim vrednostima prediktora $(x_1,\ldots,x_p)$. Na osnovu ove formule, $E(Y|(x_1,\ldots,x_p))$ može uzeti bilo koju vrednost izmedju $-\infty$ i $+\infty$. Kod problema klasifikacije klijenata koji neće izvršiti svoje obaveze na vreme (klijent će difoltirati $y = 1$) i urednih klijenata ($y = 0$) moguća su samo dva ishoda, te je zavisna promenljiva y binarna. Da bismo uprostili notaciju, koristićemo funkciju $\pi(x_1,\ldots,x_p) = E(Y|(x_1,\ldots,x_p))$ koja predstavlja uslovnu verovatnoću za $Y = 1$ pri vrednostima prediktora $(x_1,\ldots,x_p)$. Transformacija koju ćemo koristiti je sledeća:

$$\pi(x_1,\ldots,x_p) = \frac{e^{\beta_0+\beta_1 x+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x+\cdots+\beta_p x_p}} \tag{10}$$

Transformacija funkcije $\pi(x_1,\ldots,x_p)$ je centralno mesto u izvođenju logističke regresije. Do nje se dolazi uvođenjem funkcije $g(x)$ koja se naziva *logit* transformacija:

$$g(\boldsymbol{x}) = \ln\frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})} = \beta_0 + \beta_1 x + \cdots + \beta_p x_p, \tag{11}$$

gde je $\boldsymbol{x}$ vektor jedne opservacije sa vrednostima prediktora $(x_1, x_2, \ldots, x_p)$. Funkcija $g(x)$ ima osobine modela linearne regresije, a to je da je linearna po parametrima, može biti neprekidna i $g(x)$ je iz opsega $-\infty$ do ∞ u zavisnosti od vrednosti x. Zbog navedenih osobina ova funkcija je pogodna za korišćenje. Kod linearne regresije pretpostavljamo da se opservacija zavisne promenljive može izraziti na sledeći način;

$$Y|(x_1, \ldots, x_p) = E(Y|(x_1, \ldots, x_p)) + \epsilon \tag{12}$$

Gde ϵ predstavlja odstupanje opservacije od srednje vrednosti uslovnog očekivanja. Možemo predstaviti vrednost zavisne promenljive kao $y = \pi(\boldsymbol{x}) + \epsilon$. Ako je $y = 1$, onda je $\epsilon = 1 - \pi(\boldsymbol{x})$ sa verovatnoćom $\pi(\boldsymbol{x})$, dok u slučaju kada je $y = 0$, tada je $\epsilon = -\pi(\boldsymbol{x})$, sa verovatnoćom $1 - \pi(x)$. Dakle, ϵ ima očekivanje 0 i disperziju $\pi(\boldsymbol{x})(1 - \pi(\boldsymbol{x}))$.

Da bismo dobili model logističke regresije koji ćemo koristiti, neophodno je da ocenimo nepoznate parametre $\beta_0, \beta_1, \ldots, \beta_p$. Pretpostavimo da imamo n nezavisnih parova opservacija $(\boldsymbol{x_i}, y_i), i = 1 \ldots n$, gde y_i predstavlja vrednost zavisne binomne promenljive, a $\boldsymbol{x_i}$ je vrednost nezavisne promenljive za i-tu opservaciju. Kod linearne regresije, metod koji se uobičajeno koristi je metod najmanjih kvadrata. Tražene parametre $\beta_0, \beta_1, \ldots, \beta_p$ biramo tako da minimizuju sumu kvadrata odstupanja stvarnih vrednosti od vrednosti predviđenih na osnovu modela. U slučaju logističke regresije koristimo metod maksimalne verodostojnosti. Traženi parametri su oni koji maksimizuju funkciju verostojnosti, tj. najbolje odgovaraju dostupnim podacima.

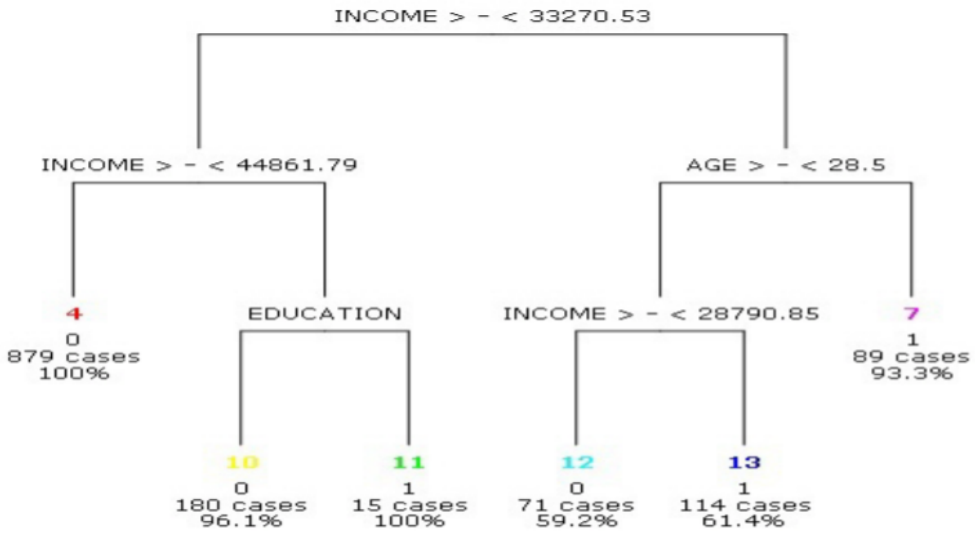
Da bismo ocenili maksimum funkcije verodostojnosti neophodno je koristiti numeričke metode poput Njutn-Rafsonovog algoritma [21].

Nakon što smo dobili verovatnoću događaja $y = 1$, $\pi(\boldsymbol{x})$, možemo izvršiti klasifikaciju. Podrazumevana granica je 0.5, pa sve opservacije sa verovatnoćom iznad 0.5 klasifikujemo kao rizične klijente. Kako radimo sa nebalansiranim skupom podataka i važnije nam je da uočimo potencijalnog difoltera nego da nerizičnog klijenta proglasimo rizičnim, ova granica se može smanjiti i definisati internim procesima unutar banke.

Detaljnije informacije o logističkoj regresiji mogu se naći u sledećim knjigama [20, 19].

4.2 Stablo odlučivanja

Model stabla odlučivanja se zasniva na deljenju prostora varijabli na regione, gde svaki region predstavlja jedan jednostavan model. U slučaju regresionih stabala izlaz modela je konstanta koja se računa kao srednja vrednost opservacija iz trening skupa koje se nalaze u tom regionu. U našem slučaju ispitujemo performanse klasifikacionog stabla pa je izlaz modela udeo difoltera u određenom regionu.



Slika 7: Primer stabla odlučivanja preuzet iz [13]

Da bismo napravili stablo odlučivanja, potrebno je da odredimo prediktore koji će se nalaziti u čvorovima stabla i vrednosti prediktora koje će razdvajati prostor na levo i desno podstablo. Kada smo izabrali koren stabla, izbor se rekurzivno nastavlja na levom i desnom podstablu. Čvorovi koji se dalje ne granaju se nazivaju listovi, a dužina maksimalne grane od korena do lista se drugačije naziva dubina stabla. Skup pravila koji nas je doveo do korena do nekog lista određuje jedan region. Sa slike 7 vidimo jedan region označen brojem 7 gde je $income < 33270.53$ i $age < 28.5$. U ovaj region upada ukupno 89 klijenata od čega je 93.3% difoltiralo.

Da bismo napravili stablo odlučivanja, potrebno je da ispratimo sledeća dva koraka:

- Prostor prediktora $(X_1, \ldots, X_n)$ delimo na J različitih, disjunknih regiona $R_1, \ldots, R_J$.
- Za svaku opservaciju koja upadne u region R_j predviđamo istu vrednost, koja je udeo difoltera onih opservacija iz trening skupa koje upadaju u R_j.

Kao meru čistoće (eng. purity) odnosno sposobnosti razvrstavanja klasa koriste se Đinijev indeks, koji u slučaju binarne klasifikacije uzima sledeći oblik:

$$Gini = 2p(1-p), \tag{13}$$

pri čemu p predstavlja udeo difoltera u listu stabla. Kao alternativa Đinijevom indeksu koristi se entropija:

$$H = -plogp - (1-p)log(1-p). \tag{14}$$

Idealan slučaj bi bio da stablo potpuno razdvaja difoltere od nedifoltera, tada bi verovatnoća bila 1, a vrednost entropije 0. Maksimalna vrednost entropije se postiže pri verovatnoći $p = \frac{1}{2}$, što znači da stablo pruža vrlo slabo razdvajanje klasa.

Cilj je da pronađemo regione $R_1, \ldots, R_J$ koji će minimizovati entropiju

$$-\hat{p}_j log\hat{p}_j - (1-\hat{p}_j)log(1-\hat{p}_j) \tag{15}$$

gde $\hat{p}_j$ predstavlja udeo difoltera u odnosu na sve opservacije koje upadaju u region R_j. Algoritam treba da izabere varijable koje će deliti prostor, kao i tačke koje će biti temena regiona. Koristimo rekurzivni pristup da bismo dobili regione, tj. prvo biramo varijablu X_j koja najbolje razdvaja difoltere od nedifoltera. Ta varijabla će biti u korenu stabla. Pored varijable bitno je i naći vrednost s te varijable koja će razdvajati difoltere od nedifoltera, nazovimo je tačka preseka. Time smo dobili dva regiona R_1 i R_2 koja izgledaju ovako:

$$R_1(j,s) = \{X | X_j \leq s\}, \tag{16}$$

$$R_2(j,s) = \{X | X_j > s\} \tag{17}$$

Ovi regioni predstavljaju prostor prediktora X gde je varijabla X_j manja ili veća od vrednosti s. Postupak nastavljamo rekurzivno na regionima R_1 i R_2.

Nakon što smo podelili prostor na J regiona $R_1, R_2, ..., R_J$ i rezultat modela u svakom regionu je konstanta c_j, tada stablo možemo formalno zapisati na sledeći način:

$$f(x) = \sum_{j=1}^{J} p_j I\{x \in R_j\} \tag{18}$$

Ocena verovatnoće p_j je $\hat{p}_j$ srednja vrednost svih y_i koji pripadaju regionu R_j:

$$\hat{p}_j = \frac{\sum_{x_i \in R_i} y_i}{|\{x_i \in R_i\}|}. \tag{19}$$

Sada se postavlja pitanje kako pronaći najbolju podelu prostora. Razmotrićemo podelu varijable j kojoj je s tačka preseka. Ranije smo utvrdili podelu na dva regiona:

$$R_1(j, s) = \{X | X_j \leq s\} \tag{20}$$

$$R_2(j, s) = \{X | X_j > s\} \tag{21}$$

Tražimo varijablu j i tačku preseka s minimizacijom sledećeg izraza:

$$\min_{j,s}[-\hat{p}_j \log \hat{p}_j - (1 - \hat{p}_j) \log(1 - \hat{p}_j))] \tag{22}$$

Za svaku varijablu može se vrlo lako doći do tačke preseka, te prolaženjem kroz sve varijable, određivanje para (j, s) je moguće. Proces se dalje nastavlja za svaki od dva regiona.

Kada se proces zaustavlja? Odnosno, koliko veliko stablo treba da bude? Ukoliko je stablo preveliko može doći do preprilagođavanja (eng. overfitting), dok prerano zaustavljanje može imati slabu prediktivnost kao posledicu. Jedan način regularizacije stabla je da formiramo veliko stablo T_0 određene dubine m. Zatim ćemo stablo T_0 "orezati" na sledeći način.

Definišemo stablo T koje je podskup stabla T_0. Neka $|T|$ predstavlja broj listova stabla T. Neka su:

$$N_m = |\{x_i \in R_m\}|, \tag{23}$$

$$\hat{p}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i, \tag{24}$$

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (-\hat{p}_m log \hat{p}_m - (1 - \hat{p}_m) log(1 - \hat{p}_m)), \tag{25}$$

definišemo funkciju:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|. \tag{26}$$

Cilj je da, za svako α, nađemo stablo $T_\alpha \subseteq T_0$ koje minimizuje $C_\alpha(T)$. Regularizacioni parametar $\alpha \geq 0$ balansira između dubine stabla i prilagođavanja podacima. Veće vrednosti α daju manje stablo, za $\alpha = 0$ dobili bismo celo stablo T_0.

Za svako α postoji jedinstveno najmanje podstablo T_α koje minimizuje $C_\alpha(T)$, detalji dostupni u [18]. Da bismo pronašli T_α izbacujemo čvor koji dovodi do najmanjeg povećanja $\sum_{m=1}^{|T|} N_m Q_m(T)$ i nastavljamo dok nam ne ostane stablo sa samo jednim čvorom. Ovo nam daje niz podstabala i može se dokazati da ovaj niz mora sadržati T_α. Procena parametra α se vrši unakrsnom validacijom.

Stabla su posebno popularna u medicini jer oponašaju način razmišljanja doktora, gde su dijagnoze poređane od verovatnijih do manje verovatnih. Prednosti stabla su interpretabilnost, ne zahtevaju poseban tretman kategoričkih varijabli i nisu osetljivi na ekstremne vrednosti. Mana je što se lako preprilagođavaju i osetljivi su na podatke na kojima su istrenirani. U narednom poglavlju predstavljamo načine na koje se to može izbeći.

4.3 Ansambli

Nestabilnost stabla odlučivanja može se prevazići korišćenjem ansambla. Ideja ansambla je da umesto jednog modela koristimo više njih, a zatim ukombinujemo njihove rezultate. Modeli koje koristimo su jednostavni algoritmi poput stabla odlučivanja i oni se nazivaju slabi učenici (eng. weak learners). Njihovim kombinovanjem dobijamo konačni model koji je jak učenik. Razlikujemo dve vrste ansambla: pojačavanje (eng. boosting) i prosta agregacija (eng. bagging).

Pri prostoj agregaciji formiraju se modeli na različitim uzorcima. Konačna odluka je srednja vrednost odluka pojedinačnih modela u slučaju regresije ili većinske odluke u slučaju klasifikacije.

Kod pojačavanja, umesto nezavisnih modela pravimo niz sekvencijalnih modela tako da svaki naredni model teži da poboljša nedostatke prethodnog.

Pri pravljenju svakog narednog modela opservacijama dodeljujemo koeficijente tako da one opservacije koje su pogrešno klasifikovane u prethodnom koraku imaju veću težinu. Opisani algoritam su predstavili Frojnd i Šapire 1996. godine u radu [16].

4.4 Slučajna šuma

Slučajna šuma je jedan od najpoznatijih primera ansambla. Algoritam slučajne šume nastaje tako što od početnog uzorka napravimo više podskupova u smisli varijabli i opservacija koje ga čine. Na svakom uzorku se formira jedno stablo. Slučajna šuma pripada ansamblima proste agregacije te je finalna odluka rezultat glasanja u slučaju klasifikacije. Svako stablo daje jedan glas, a većina glasova određuje konačni izlaz modela. U slučaju regresije izlaz modela je srednja vrednost ocena pojedinačnih stabala.

Stabla odlučivanja su modeli sa visokom disperzijom. To znači da ako podelimo uzorak na dve polovine i treniramo stablo odlučivanja nad njima, dobijena stabla mogu biti veoma različita. Kako bi se smanjila disperzija, koristi se metod butstrap agregacije (eng. bootstrap). Za dati skup od n nezavisnih opservacija $Z_1, \ldots, Z_n$, pri čemu svaka ima disperziju σ^2, disperzija srednje vrednosti $\overline{Z}$ je data sa σ^2/n. Drugim rečima, uprosečavanje opservacija smanjuje disperziju. Dakle, način da poboljšamo prediktivnost modela je da napravimo različite podskupove populacije, a zatim na svakom podskupu treniramo model i na kraju uprosečimo dobijene rezultate [20]. Formalno zapisano, pravimo modele $\hat{f}^1(x), \hat{f}^2(x), ..., \hat{f^B}(x)$ na B odvojenih trening skupova. Njihovim uprosečavanjem dobijamo jedan model sa manjom disperzijom:

$$\hat{f_{avg}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x). \tag{27}$$

Čest problem je nedovoljan broj opservacija u trening skupu pa da ne bismo dodatno umanjivali veličinu trening skupa koristimo butstrap pristup. To znači da iz istog skupa biramo sa ponavljanjem B različitih podskupova. Broj stabala B predstavlja parametar, međutim, korišćenje velike vrednosti parametra B ne vodi preprilagođavanju. Grafički se može ispitati nakon koje vrednosti parametra B je greška predviđanja stabilna i izabrati tu vrednost B.

Postupak opisan iznad je univerzalan za sve modele proste agregacije. Kod slučajnih šuma napravljena je dodatna izmena koja unapređuje algoritam proste agregacije. Prethodno smo formirali stabla na nasumično generisanim podskupovima, sada, dodatno, biramo i nasumičan podskup od m prediktora iz skupa svih p prediktora. Najčešće biramo m tako da $m \approx \sqrt{p}$. Time smo obezbedili da, ukoliko u skupu imamo jednog ili više prediktora koji imaju značajno veću prediktivnu moć od ostalih prediktora, izbegnemo situaciju gde imamo dosta sličnih stabala odlučivanja. Nasumičnim biranjem prediktora dajemo šansu i ostalim prediktorima da budu deo stabla odlučivanja.

5 Gradijentno pojačavanje

Kao i kod drugih ansambla, ideja pojačavanja je da kombinuje slabe učenike poput stabla odlučivanja u jakog učenika. Pod slabim učenikom podrazumevamo model čija je prediktivnost bliska nasumičnom pogađanju. Kombinovanjem slabih učenika dobijamo model dosta bolje prediktivne moći koji nazivamo jakim učenikom. Formira se serija stabala odlučivanja koji zajedno predstavljaju prediktivni model. Svako novo stablo se trenira na greškama prethodnog učenika, tako da poboljšava prediktivnost.

Algoritam gradijentnog pojačavanja je prvi izložio Fridman u svom radu [17]. Ideja gradijentnog pojačavanja je da korišćenjem negativnog gradijenta unapredi ansambl.

Kod statističkog učenja, cilj je da korišćenjem vrednosti prediktora $\boldsymbol{x} = \{(x_1, \ldots, x_n)\}$ na trening skupu i vrednosti ciljne promenljive y dobijemo aproksimaciju $\hat{F}$ funkcije F^* koja minimizuje funkciju greške $L(y, F(\boldsymbol{x}))$, gde je $F(\boldsymbol{x})$ iz skupa dopustivih funkcija.

$$F^* = \underset{F}{\operatorname{argmin}} E_{y,\boldsymbol{x}} L(y, F(x)) = \underset{F}{\operatorname{argmin}} E_{\boldsymbol{x}}[E_{y,\boldsymbol{x}} L(y, F(\boldsymbol{x}))|\boldsymbol{x}]. \tag{28}$$

Često se $F(\boldsymbol{x})$ ograniči da bude iz parametrizovane klase funkcija $F(\boldsymbol{x}, \boldsymbol{P})$, gde je $\boldsymbol{P} = \{P_1, P_2, \ldots\}$ konačan skup parametara. U našem slučaju, fokus je na aditivnoj formi

$$F(\boldsymbol{x}; \{\beta_m; \boldsymbol{a}_m\}_1^M) = \sum_{m=1}^{M} \beta_m h(\boldsymbol{x}; \boldsymbol{a}_m) \tag{29}$$

Funkcija $h(\boldsymbol{x}; \boldsymbol{a})$ je najčešće jednostavna parametarska funkcija ulaznih podataka $\boldsymbol{x}$ sa parametrima $\boldsymbol{a} = \{a_1, a_2, \ldots\}$.

Nama je posebno interesantan slučaj kada je $h(\boldsymbol{x}; \boldsymbol{a})$ stablo odlučivanja. Paramateri $\boldsymbol{a}_m$ nekog stabla su varijable u čvorovima, tačke preseka i vrednosti u listovima stabla.

Kako bismo odredili parametrizovanu funkciju $F(\boldsymbol{x})$, sa problema funkcionalne optimizacije prelazimo na parametarsku optimizaciju:

$$\boldsymbol{P}^* = \underset{F}{\operatorname{argmin}} \phi(\boldsymbol{P}), \tag{30}$$

gde je

$$\phi(\boldsymbol{P}) = E_{y,\boldsymbol{x}} L(y, F(\boldsymbol{x}; \boldsymbol{P})). \tag{31}$$

Nakon toga dobijamo

$$F^*(\boldsymbol{x}) = F(\boldsymbol{x}; \boldsymbol{P^*}). \tag{32}$$

Da bismo rešili problem 30, rešenje izražavamo u obliku:

$$\boldsymbol{P^*} = \sum_{m=1}^{M} \boldsymbol{p}_m, \tag{33}$$

gde $\boldsymbol{p}_0$ predstavlja inicijalnu pretpostavku, a $\{\boldsymbol{p}_m\}_1^M$ su sukcesivni inkrementi koji se računaju na osnovu prethodnih koraka.

Strmi spust (eng. steepest decscent) je jedna od najčešće korišćenih numeričkih metoda za minimizaciju. Inkrementi $\{\boldsymbol{p}_m\}_1^M$ se definišu na sledeći način. Prvo se izračuna gradijent kao:

$$\boldsymbol{g}_m = \{g_{jm}\} = [\frac{\partial \phi(\boldsymbol{P})}{\partial P_j}]_{\boldsymbol{P}=\boldsymbol{P}_{m-1}}, \tag{34}$$

gde

$$\boldsymbol{P}_{m-1} = \sum_{i=0}^{m-1} \boldsymbol{p}_i. \tag{35}$$

U ovom koraku računamo p_m

$$\boldsymbol{p}_m = -\rho_m \boldsymbol{g}_m, \tag{36}$$

gde je

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \phi(\boldsymbol{P}_{m-1} - \rho \boldsymbol{g}_m) \tag{37}$$

Negativan gradijent $-\boldsymbol{g}_m$ definiše strmi spust, a ρ_m se naziva linija pretrage duž tog pravca.

Sada možemo primeniti numeričku optimizaciju u prostoru funkcija. Smatramo da je vrednost $F(\boldsymbol{x})$ u svakoj tački $\boldsymbol{x}$ parametar i težimo da minimizujemo:

$$\phi(F) = E_{y,\boldsymbol{x}} L(y, F(\boldsymbol{x})) = E_{\boldsymbol{x}}[E_{y,\boldsymbol{x}} L(y, F(\boldsymbol{x}))|\boldsymbol{x}] \tag{38}$$

ili ekvivalentno

$$\phi(F(\boldsymbol{x})) = E_y[L(y, F(\boldsymbol{x}))|\boldsymbol{x}] \tag{39}$$

za svako $\boldsymbol{x}$. Dalje dobijamo rešenje u obliku

$$F^*(\boldsymbol{x}) = \sum_{m=0}^{M} f_m(\boldsymbol{x}), \tag{40}$$

gde je $f_0(\boldsymbol{x})$ inicijalna pretpostavka i $\{f_m\}_1^M$ su inkrementalne funkcije definisane optimizacionim metodom. Za strmi spust:

$$f_m(\boldsymbol{x}) = -\rho_m g_m(\boldsymbol{x}), \tag{41}$$

gde je

$$g_m(\boldsymbol{x}) = [\frac{\partial \phi(F(\boldsymbol{x}))}{\partial F(\boldsymbol{x})}]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})} = [\frac{\partial E_{y,\boldsymbol{x}} L(y, F(\boldsymbol{x}))|\boldsymbol{x}]}{\partial F(\boldsymbol{x})}]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})}, \tag{42}$$

i

$$F_{m-1}(\boldsymbol{x}) = \sum_{i=0}^{m-1} f_i(\boldsymbol{x}). \tag{43}$$

Pod pretpostavkom regularnosti da smemo da zamenimo mesta integralu i diferencijalu, ovo postaje:

$$g_m(\boldsymbol{x}) = E_y[\frac{\partial L(y, F(x))}{\partial F(x)}|\boldsymbol{x}]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})}, \tag{44}$$

gde je

$$\rho_m = \underset{\rho}{\operatorname{argmin}} E_{y,\boldsymbol{x}} L(y, F_{m-1}(\boldsymbol{x}) - \rho g_m). \tag{45}$$

Kada bi nam cilj bio da minimizujemo gresku na trening skupu, korišćenje strmog spusta bi bilo prihvatljivo rešenje. Međutim, gradijent je definisan samo na tačkama $\boldsymbol{x}_i$ iz trening skupa, dok je nas cilj da dobijemo $\mathbf{F}^*$ koje će imati dobar performans i na podacima van trening skupa. Jedan način je da pretpostavimo parametarsku formu poput 29, a zatim primenimo parametarsku optimizaciju

$$\{\beta_m; \boldsymbol{a}_m\}_1^M = \underset{\{\beta'_m; \boldsymbol{a}'_m\}_1^M}{\operatorname{argmin}} \sum_{i=1}^{M} L(y_i, \sum_{m=1}^{M} \beta'_m h(\boldsymbol{x}_i; \boldsymbol{a}'_m)). \tag{46}$$

U situacijama kada to nije moguće, možemo pokušati "pohlepni fazni" (eng. greedy stagewise) pristup. Za $m = 1, 2, \ldots, M$

$$(\beta_m; \boldsymbol{a}_m) = \underset{\{\beta;\boldsymbol{a}\}}{\operatorname{argmin}} \sum_{i=1}^{M} L(y_i, F_{m-1}(\boldsymbol{x}_i) + \beta h(\boldsymbol{x}_i; \boldsymbol{a})), \tag{47}$$

a zatim

$$F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \beta_m h(\boldsymbol{x}; \boldsymbol{a}_m)). \tag{48}$$

Pretpostavimo da je za određenu funkciju greške $L(y, F)$ ili učenika $h(\boldsymbol{x}; \boldsymbol{a})$ teško odrediti rešenje problema 47. Pri bilo kom aproksimatoru $F_{m-1}(\boldsymbol{x})$, funkcija $\beta_m h(\boldsymbol{x}; \boldsymbol{a}_m)$ može se smatrati najboljim pohlepnim korakom ka $F^*(\boldsymbol{x})$ dato formulom 28, sa ograničenjem da pravac $h(\boldsymbol{x}; \boldsymbol{a}_m)$ bude član parametrizovane klase funkcija $h(\boldsymbol{x}; \boldsymbol{a})$. Dakle, može se smatrati za korak strmog spusta 41. Negativan gradijentni spust daje najbolji pravac strmom spustu $-\boldsymbol{g}_m = \{-g_m(\boldsymbol{x}_i)\}_1^N$ u N-dimenzionom prostoru podataka za $F_{m-1}(\boldsymbol{x})$. Međutim, ovaj gradijent je definisan samo u tačkama $\{\boldsymbol{x}_i\}_1^N$ i ne može se proširiti na ostale vrednosti $\boldsymbol{x}$. Jedna mogućnost za generalizaciju je da izaberemo članove parametrizovane klase $h(\boldsymbol{x}; \boldsymbol{a}_m)$ koja daje $\boldsymbol{h}_m = \{h(\boldsymbol{x}_i; \boldsymbol{a}_m)\}_1^N$ najpribližnije paralelnom sa $-\boldsymbol{g}_m \in R^N$. To je $h(\boldsymbol{x}; \boldsymbol{a}_m)$ koja ima najveći koeficijent korelacije sa $-g_m(\boldsymbol{x})$ na trening skupu. Može se dobiti iz rešenja:

$$\boldsymbol{a}_m = \underset{\{\beta;\boldsymbol{a}\}}{\operatorname{argmin}} \sum_{i=1}^{M} (-g_{m-1}(\boldsymbol{x}_i) - \beta h(\boldsymbol{x}_i; \boldsymbol{a}))^2, \tag{49}$$

Ograničeni negativni gradijent $h(\boldsymbol{x}; \boldsymbol{a}_m)$ koristimo umesto $-g_m(\boldsymbol{x})$. Linija pretrage je dalje jednaka:

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{m-1}(\boldsymbol{x}_i) + \rho h(\boldsymbol{x}; \boldsymbol{a}_m)) \tag{50}$$

ažuriramo aproksimaciju

$$F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \rho_m h(\boldsymbol{x}_i; \boldsymbol{a}_m)). \tag{51}$$

Ideja je da umesto da sprovodimo komplikovanu optimizaciju 47, sprovedemo optimizaciju metodom najmanjih kvadrata.

Sada možemo zapisati konkretne korake algoritma

1. $F_0(\boldsymbol{x}) = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \rho)$

2. For $m = 1$ to M do:

3. $\tilde{y}_i = -[\frac{\partial L(y,F(\boldsymbol{x}))}{\partial F(\boldsymbol{x})}]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})}, i = 1, \ldots, N$

4. $\boldsymbol{a}_m = \underset{\{\beta;a\}}{\operatorname{argmin}} \sum_{i=1}^{N} (-\tilde{y}_i(\boldsymbol{x}_i) - \beta h(\boldsymbol{x}_i; \boldsymbol{a}))^2$

5. $\rho_m = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{m-1}(\boldsymbol{x}_i) + \rho h(\boldsymbol{x}_i; \boldsymbol{a}_m))$

6. $F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \rho_m h(\boldsymbol{x}; \boldsymbol{a}_m))$

7. endFor

Pogledajmo dalje kako izgleda algoritam u slučaju klasifikacije. Funkcija greške ima sledeći oblik

$$L(y, F) = \log(1 + exp(-2yF)), y \in \{-1, 1\}, \tag{52}$$

gde je

$$F(\boldsymbol{x}) = \frac{1}{2} \log[\frac{P(y = 1|\boldsymbol{x})}{P(y = -1|\boldsymbol{x})}] \tag{53}$$

Negativni gradijent je:

$$\tilde{y}_i = -[\frac{\partial L(y, F(\boldsymbol{x}))}{\partial F(\boldsymbol{x})}]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})} = \frac{2y_i}{1 + exp(2y_i F_{m-1}(\boldsymbol{x}_i))}. \tag{54}$$

Linija pretrage postaje

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^{N} \log(1 + exp(-2y_i(F_{m-1}(\boldsymbol{x}_i) + \gamma))). \tag{55}$$

Vrednosti u čvorovima su rešenje sledećeg problema:

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{\boldsymbol{x}_i \in R_{jm}} \log(1 + exp(-2y_i(F_{m-1}(\boldsymbol{x}_i) + \gamma))). \tag{56}$$

Korišćenjem Njutn-Rafsonovog algoritma dobija se:

$$\gamma_{jm} = \frac{\sum_{\boldsymbol{x}_i \in R_{jm}} \tilde{y}_i}{\sum_{\boldsymbol{x}_i \in R_{jm}} |\tilde{y}_i|(2 - |\tilde{y}_i|)}. \tag{57}$$

1. $F_0(\boldsymbol{x}) = \frac{1}{2}\log\frac{1+\overline{y}}{1-\overline{y}}$
2. For $m = 1$ to M do:
3. $\tilde{y}_i = \frac{2y_i}{1+exp(2y_i F_{m-1}(x_i))}, i = 1, \ldots, N$
4. $\{R_{jm}\}_1^J = J\{\tilde{y}_i, \boldsymbol{x}_i\}_1^N$
5. $\gamma_{jm} = \frac{\sum_{\boldsymbol{x}_i \in R_{jm}} \tilde{y}_i}{\sum_{\boldsymbol{x}_i \in R_{jm}} |\tilde{y}_i|(2-|\tilde{y}_i|)}$
6. $F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \gamma_{jm} I\{\boldsymbol{x} \in R_{jm}\}$
7. endFor

Kada F_M prevedemo u verovatnoću dobijamo finalni izlaz modela:

$$p(y = 1|x) = \frac{1}{1 + e^{-2F_M}} \tag{58}$$

$$p(y = -1|x) = \frac{1}{1 + e^{2F_M}} \tag{59}$$

5.1 Ekstremno gradijentno pojačavanje

Ekstremno gradijentno pojačavanje (eng. XGBoost) su razvili Čen i Gerstin 2014. godine [11]. XGBoost je napredna implementacija algoritma gradijentnog pojačavanja, nudi veću efikasnost, tačnost i skalabilnost naspram jednostavnih algoritama agregacije. Optimizacija se vrši na proširenom skupu metaparametara što povećava fleksibilnost. XGBoost je stekao veliku popularnost i privukao pažnju brojnim pobedama na Kaggle izazovima.

XGBoost ima ugrađen u sebi nekoliko optimizacija koje ubrzavaju obučavanje algoritma. To su:

- Približni pohlepni algoritam - Jedan od ključnih problema pri obučavanju stabla je pronalaženje najbolje tačke podele. Da bi se odredila najbolja podela algoritam prolazi kroz sve moguće podele za sve varijable. Pre toga algoritam prvo sortira podatke po varijablama. Ovo se naziva tačni pohlepni algoritam (eng. exact greedy algorithm). Ovaj algoritam je vrlo moćan, ali ukoliko radimo sa velikim

skupom podataka, nemoguće je sprovesti efikasnu podelu. Kao alternativu tačnom pohlepnom algoritmu, koristimo njegovu aproksimaciju. Umesto isprobavanja svih mogućih podela varijable, ekstremno gradijentno pojačavanje koristi kvantile varijable za brže nalaženje najbolje podele.

- Korišćenje blokova za paralelno učenje - Najveći deo vremena oduzima sortiranje varijabli. Da bi se smanjilo vreme potrošeno na sortiranje, podaci se čuvaju u blokovima. Podaci u svakom bloku su sortirani po varijabli, time je ubrzano nalaženje najbolje podele. Računanje statistika po kolonama se može obavljati paralelno.

- Prepoznavanje probleme u podacima poput - Pod ovim podrazumevamo detekciju nedostajućih vrednosti, čestih pojavljivanja nula ili prisustva prethodnih transformacija varijabli poput enkodiranja. Uvodi se podrazumevana grana, tako da u slučaju da nedostaje vrednost neke varijable, instanca će slediti podrazumevani pravac.

- Korišćenje procesorske keš memorije za čuvanje gradijenata, što ubrzava kalkulaciju.

6 Interpretabilnost modela

Interpretabilnost modela je od velikog značaja u oblasti kreditnog rizika. Zbog visoke regulisanosti oblasti u kojoj se primenjuje, modeli koji se koriste moraju biti etički, transparentni i u skladu sa Zakonom o zaštiti podataka o ličnosti (eng. GDPR) [4]. Modeli moraju biti odobreni od strane eksperata iz oblasti bankarstva koji najčešće nemaju statističko znanje. Da bi se model odobrio, potrebno je jasno predstaviti koje varijable ga čine i kako one doprinose krajnjem rezultatu. Na primer model u sebi sadrži varijablu broj godina i model prepoznaje mlađe klijente kao rizičnije, to je u skladu sa dugogodišnjim ekspertskim iskustvom u odobravanju kredita i ova varijabla je pogodna za model. Ukoliko bismo koristili neuronsku mrežu i znali samo koje varijable ulaze u model, ali ne i kako one utiču na krajnji rezultat, susreli bismo se sa dosta problema u predstavljanju modela ekspertima i tumačenju ishoda modela. Netransparentan model može dovesti do različitih zloupotreba. Poslednjih godina, često se diskutuje o tome da li statistički modeli diskriminišu pojedine grupe ljudi [9].Kancelarija američkog predsednika objavila je izveštaj koji ističe opasnosti automatskog donošenja odluka u različitim industrijama među kojima je i određivanje kreditnog rejtinga [24]. Sve više radova diskutuje o pravičnosti modela i statističkim načinima da se izmeri koliko je model fer [14, 22].

Kako bismo prevazišli problem interpretabilnosti modela, koristićemo Šeplijevu[1] vrednost. Šeplijeva vrednost je koncept iz koalicione teorije igara. Odgovara srednjoj vrednosti marginalnih doprinosa igrača. Prednost Šeplijeve vrednosti je što se može iskoristiti za merenje doprinosa svake varijable bez obzira koji model koristimo. Pristup se oslanja na Šeplijev okvir koji dozvoljava procenu Šeplijeve vrednosti izražavajući predikcije kao linearne kombinacije binarnih varijabli koje daju informaciju za svaku varijablu da li je uključena u model ili ne. Primena Šeplijeve vrednosti u oblasti kreditnog rizika ispitana je u radu [8].

Formalno, funkcija $g(x')$ koji objašnjava predikcije $f(x)$ konstruiše se modelom aditivnih rangiranja koja razlaže predikcije u linearnu funkciju bi-

[1]Lojd Stovel Šepli (1923-2016) američki matematičar i dobitnik Nobelove nagrade za ekonomiju, bavio se oblastima primene matematike u ekonomiji i teorijom igara.

narnih varijabli $z' \in \{0,1\}^M$ i vrednosti $\phi_i \in R$:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i. \tag{60}$$

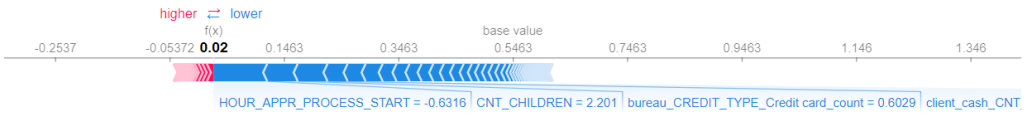
Drugim rečima, $g'(z') \approx f(h_x(z'))$ je lokalna aproksimacija predikcija gde lokalna funkcija $h_x(x') = x$ preslikava uprošćenu varijablu x' u x,$z' \approx x$ i M je broj selektovanih varijabli.

Lundberg i Li [23] su dokazali da se jedini model aditivnih rangiranja, koji zadovoljava svojstva lokalne tačnosti, doslednosti i tretmana 0, dobija dodeljivanjem varijabli x'_i vrednosti ϕ_i, koja se naziva Šeplijeva vrednost definisana je kao:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z \setminus i)] \tag{61}$$

gde je f trenirani model, x vektor ulaznih podataka , x' vektor izabranih varijabli. Vrednost $f_x(z') - f_x(z \setminus i)$ je doprinos varijable i i izražava za svaku predikciju devijaciju Šeplijeve vrednosti od srednje vrednosti.

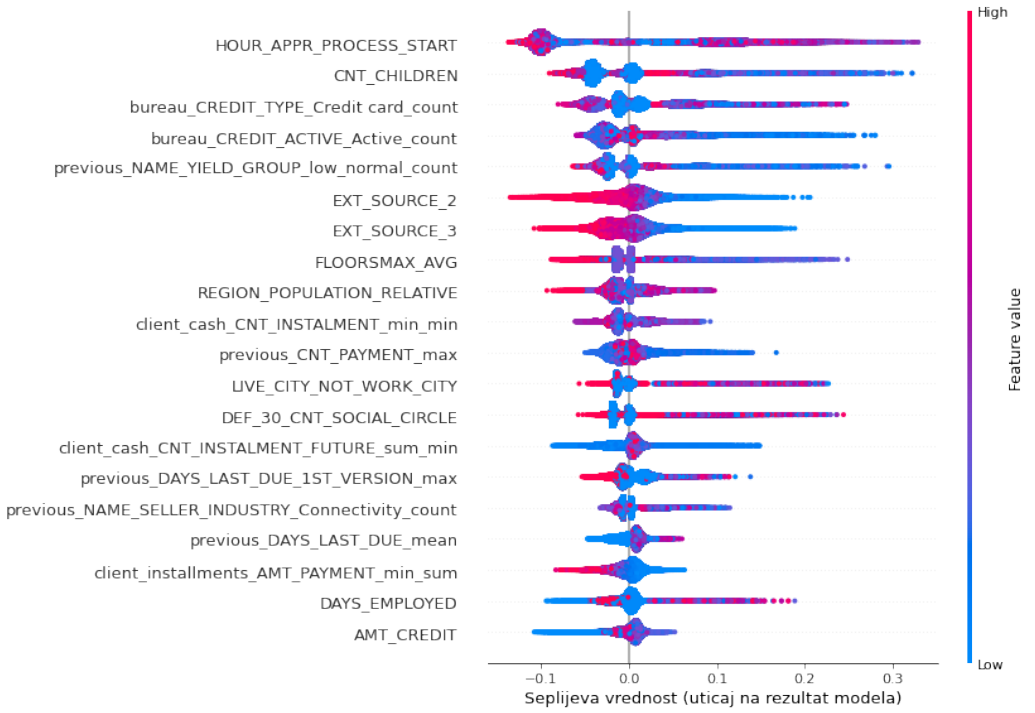
Drugim rečima, Šeplijeva vrednost predstavlja jedinstvenu veličinu koja može da konstruiše model koji lokalno linearno aproksimira originalni model za specifični ulazni podatak x (lokalna tačnost). Ima svojstvo da kad god je varijabla 0, Šeplijeva vrednost je isto 0 i ako je u drugom modelu veći doprinos varijable i Šeplijeva vrednost je veća (doslednost).



Slika 8: Šeplijeva vrednost

Na slici iznad je primer kako vizuelno možemo prikazati izlaz modela za jednog klijenta. Plavom bojom su označene varijable koje poboljšavaju kreditni rejting, a crvenom one koje ga snižavaju. Varijable su poređane po doprinosu, tako da su u sredini one koje najviše poboljšavaju, odnosno snižavaju rejting, a kako se udaljavamo od sredine varijable sve manje doprinose konačnom rezultatu. Sa slike možemo i uporediti dve varijable različite boje posmatranjem veličine strelica, pa tako možemo zaključiti da li jedna

varijabla više poboljšava rejting nego što ga druga snižava ili, u slučaju da su strelice iste veličine, zaključak bi bio da se te dve varijable međusobno potiru.



Slika 9: Šeplijeva vrednost

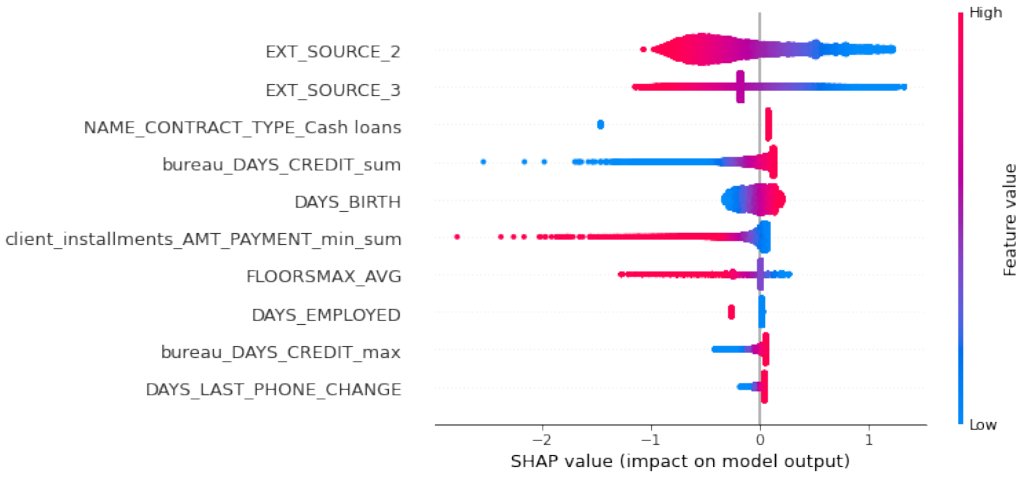
Prethodna slika nam je davala interpretaciju rezultata jednog klijenta, dok nam prikaz ispod daje tumačenje kompletnog modela. Varijable su poređane po značaju odozgo na dole, skala sa desne strane pokazuje vrednost varijable, dok na $x - osi$ možemo da pročitamo kako su vrednosti varijable raspodeljene po klijentima. Npr. varijabla iznos kredita koja je poslednja po značaju jasno pokazuje da niske tj. plave vrednosti varijable ukazuju na nisku verovatnoću difolta. Varijable EXT_SOURCE_2 i EXT_SOURCE_3 jasno razgraničavaju dobre od loših klijenata, a prisetimo se da su ove varijable imale visoku informativnu vrednost.

Šeplijeva vrednost je od posebnog značaja kod neinterpretabilnih modela poput slučajnih šuma, ipak ništa nas ne sprečava da postupak primenimo na logističku regresiju koju već znamo da interpretiramo. Time možemo uporediti dobijene rezultate sa postojećim ishodom logističke regresije i testirati

da li se interpretacije poklapaju.

	coef	std err	z	P>|z|	[0.025	0.975]
EXT_SOURCE_2	-0.5116	0.003	-147.918	0.000	-0.518	-0.505
EXT_SOURCE_3	-0.4818	0.004	-133.566	0.000	-0.489	-0.475
DAYS_BIRTH	0.1013	0.005	22.403	0.000	0.092	0.110
NAME_CONTRACT_TYPE_Cash loans	-0.2289	0.004	-64.194	0.000	-0.236	-0.222
client_installments_AMT_PAYMENT_min_sum	-0.1538	0.004	-37.362	0.000	-0.162	-0.146
bureau_DAYS_CREDIT_max	0.0722	0.004	19.052	0.000	0.065	0.080
bureau_DAYS_CREDIT_sum	0.1885	0.004	50.482	0.000	0.181	0.196
DAYS_LAST_PHONE_CHANGE	0.0220	0.004	5.880	0.000	0.015	0.029
DAYS_EMPLOYED	-0.1034	0.005	-22.180	0.000	-0.113	-0.094
FLOORSMAX_AVG	-0.1606	0.004	-40.969	0.000	-0.168	-0.153

Slika 10: Rezultati logističke regresije



Slika 11: Šeplijeva vrednost kod logističke regresije

Varijable su prethodno standardizovane pa je osnovu vrednosti koeficijenta logističke regresije jasno koje varijable imaju najveći uticaj. Izuzetak su dummy varijable koje interpretiramo drugačije. Na slici 10 najveći apsolutni koeficijent ima varijabla EXT_SOURCE_2, a negativni koeficijent ukazuje da klijenti sa većim vrednostima varijable imaju manju verovatnoću difolta. Uporedimo sada Šeplijevu vrednost za ovu varijablu sa slike 11. Horizontalna linija ukazuje koje vrednosti varijable imaju veću verovatnoću difolta. Što se više pomeramo u desno, povećava se verovatnoća difolta. Na vertikalnoj liniji je označeno da su niže vrednosti varijable označene plavom, a veće crvenom bojom. Konačno možemo zaključiti da manje vrednosti varijable EXT_SOURCE_2 dovode do veće verovatnoće difolta. Ovo se poklapa sa dobijenim rezultatima logističke regresije.

Na slici 11 varijable su poređane po značaju u modelu, upoređivanjem sa apsolutnim koeficijentima logističke regresije sa slike 10 vidimo da se dobija približno isti zaključak. Još jedan način da uporedimo rezultate je upoređivanjem znaka koeficijenata sa rasporedom boja na slici 11. Varijable kod kojih je plava vrednost na desnoj strani označavaju da niže vrednosti varijable imaju veću verovatnoću difolta, te varijable imaju negativan koeficijent. Obrnuto, varijable kod kojih veće vrednosti imaju veću verovatnoću difolta imaju pozitivan koeficijent.

7 Evaluacija modela

Kao što smo napomenuli ranije, radimo sa nebalansiranim skupom podataka. Zbog toga moramo biti izuzetno pažljivi pri evaluaciji modela. Tradicionalne mere poput tačnosti (eng. accuracy) ne daju nam pravi uvid u prediktivnu moć modela. Kod problema binarne klasifikacije, uvek možemo početi od matrice konfuzije. Imamo četiri mogućnosti kod klasifikacije:

- TP (eng. true positive) - predvideli smo da klijent neće difoltirati i zaista nije difoltirao
- TN (eng. true negative) - predvideli smo da će klijent difoltirati i to se desilo
- FP (eng. false positive) - predvideli smo da klijent neće difoltirati, međutim klijent je ipak difoltirao
- FN (eng. false negative) - predvideli smo da će klijent difoltirati, ali se to nije ostvarilo

Na osnovu ovih brojeva možemo izračunati više mera prediktivnosti modela:

- Tačnost (eng. accuracy) = $\frac{TP+TN}{P+N}$, gde P predstavlja ukupan broj nedifoltera, a N ukupan broj difoltera, daje nam procenat tačno klasifikovanih instanci.
- Preciznost (eng. precision) = $\frac{TP}{TP+FP}$, računa se u odnosu na ciljnu klasu i daje nam procenat tačno klasifikovanih instanca klase od interesa u odnosu na sve instance koje smo klasifikovali kao pripadnike ciljne klase. U našem slučaju želimo da prepoznamo što više difoltera, pa bismo preciznost računali kao odnos prepoznatih difoltera i svih opservacija za koje smo rekli da su difolteri.
- Odziv (eng. recall) = $\frac{TP}{P}$, računa se u odnosu na ciljnu klasu i daje nam procenat tačno klasifikovanih instanci klase od interesa.
- F_1 mera = $2\frac{preciznost \cdot odziv}{preciznost+odziv}$ predstavlja harmonijsku sredinu preciznosti i odziva.

Da bismo dobili matricu konfuzije, potrebno je da definišemo graničnu vrednost, te da zatim sve opservacije sa verovatnoćom iznad granične proglasimo difolterima. Ukoliko bi nam klase bile podjednako važne, koristili bismo 0.5 kao granicu, međutim, često nam je bitnije da na vreme uočimo sve opservacije jedne klase, čak i po cenu toga da dosta opservacija druge klase pogrešno klasifikujemo. Ovakav pristup se koristi u medicini, prepoznavanju prevara, pa i u kreditnom riziku. Želimo meru koja nam ukazuje da li model dodeljuje nižu verovatnoću nedifolterima, a višu difolterima, bez obzira na graničnu vrednost koju izaberemo. To nam pokazuje ROC kriva [12] (eng. Receiver Operating Characteristic), a konkretna mera prediktivnosti se naziva AUC (eng. area under curve) i računa se kao površina ispod ROC krive. Da bismo definisali ROC krivu, potrebno je da definišemo njene ose. Na jednoj osi će biti odziv pozitivne klase, naziva se još i osetljivost (eng. sensitivity).

$$osetljivost = \frac{TP}{TP + FN} \tag{62}$$

Na drugoj osi će biti specifičnost (eng. specificity).

$$1 - specificnost = \frac{FP}{TN + FP} \tag{63}$$

Osetljivost se drugačije naziva i udeo tačno klasifikovanih pozitivnih opservacija (eng. true positive rate), a 1-specifičnost udeo netačno klasifikovanih pozitivnih opservacija. U graničnim slučajevima, ako bismo sve opservacije proglasili difolterima, osetljivost bi bila 0.

Skup podataka je podeljen na 80% podataka koji su deo trening skupa i 20% upada u test skup. Kod algoritama slučajne šume i gradijentnog pojačavanja postoji dosta hiperparametara čijim podešavanjima možemo postići bolje rezultate. Hiperparametre biramo metodom unakrsne validacije na trening skupu i tu konfiguraciju dalje koristimo. Za dobijanje najbolje konfiguracije korišćen je algoritam $RandomSearchCV$. Zbog prevelikog broja kombinacija hiperparametara nije bilo izvodljivo koristiti $GridSearchCV$ algoritam.

Hiperparametri slučajne šume:

- $n\_estimators$ - broj stabala, podrazumevana vrednost je 100, a skup na kojem ćemo tražiti najbolje rešenje je $\{100, 500, 1000\}$;
- max_depth - dubina stabla, ispitane vrednosti iz skupa $\{3, 4, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120\}$;

- *min_sample_split* - predstavlja minimalan broj opservacija potreban da se nastavi grananje stabla, podrazumevana vrednost je 2, a testirane vredosti su $\{2, 6, 10\}$;
- *min_samples_leaf* - označava minimalan broj opservacija koje se mogu naći u listu stabla, najbolji parametar tražimo među $\{1, 3, 4\}$, a podrazumevana vrednost je 1;
- *max_sample* - odnosi se na udeo opservacija koje se prosleđuju pri kreiranju svakog novog stabla odlučivanja, ispitane vrednosti $\{0.6, 0.8, 1\}$;
- $max\_features$ - predstavlja broj varijabli koje se prosleđuju pri pravljenju novog stabla odlučivanja, uzima vrednosti *sqrt* ili $log2$, podrazumevana vrednost je *sqrt*;
- *bootstrap* - označava da li se podaci generišu *bootstrap* metodom ili ne, moguće vrednosti True i False.

Parametar	Home Credit	German Credit
$n\_estimators$	1000	500
min_sample_split	6	10
$min\_samples\_leaf$	3	1
max_sample	0.8	0.8
$max\_features$	sqrt	log2
max_depth	110	80
bootstrap	False	True

Tabela 1: Hiperparametri slučajne šume

Hiperparametri ekstremnog gradijentnog pojačavanja:

- $n\_estimators$ - broj stabala, podrazumevana vrednost je 100, a skup na kojem ćemo tražiti najbolje rešenje je $\{100, 500, 1000\}$;
- *eta* - koristi se da bi se usporio proces i time sprečilo preprilagođavanje, predstavlja parametar kojim se množe težine novih stabala odlučivanja, može uzeti vrednosti iz intervala $[0, 1]$, a podrazumevana vrednost je 0.3, ispitane su vrednosti iz skupa $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.5\}$;

- *gamma* - predstavlja minimalno smanjenje greške potrebno da se nastavi grananje stabla, veće vrednosti ukazuju na konzervativniji algoritam, podrazumevana vrednost je 0, testirane su vrednosti $\{0.1, 0.5, 1, 1.5, 2, 3, 5\}$;

- *max_depth* - dubina stabla, ispitane vrednosti iz skupa $\{5, 6, 7\}$, podrazumevana vrednost je 3;

- *min_child_weight* - predstavlja minimalnu sumu težina svih opservacija u čvoru stabla, veće vrednosti sprečavaju prilagođavanje modela specifičnim pojavama, testirane vrednosti $\{1, 2, 3, 5, 10, 20\}$, podrazumevana vrednost je 1;

- *max_delta_step* - kontroliše težine listova, ovaj parametar nije uvek neophodan, ali može pomoći u slučaju nebalansiranih klasa, podrazumevana vrednost je 0, a tražimo najbolju vrednost na skupu $\{1, 3, 5, 10\}$;

- *subsample* - označava udeo opservacija pri treniranju svakog novog stabla odlučivanja, podrazumevana vrednost je 1, a ispitane vrednosti $\{0.6, 0.8, 1\}$;

- *colsample_bytree* - odnosi se na udeo varijabli koje ćemo razmatrati pri pravljenju stabla odlučivanja, ispitane vrednosti $\{0.6, 0.8, 1\}$, a podrazumevana vrednost je 1;

- *lambda* - regularizacioni parametar, ispitane vrednosti $\{0.01, 0.1, 1, 10\}$, podrazumevana vrednost je 1;

- *tree_method* - označava algoritam korišćen za dobijanje stabla, vrednost *exact* označava da je korišćen tačni pohlepni algoritam, a *auto* da algoritam automatski bira da li se koristi tačni ili približni pohlepni algoritam u zavisnosti od podataka;

U tabeli su prikazani rezultati najbolje konfiguracije evaluirane na test skupu. Pri obučavanju modela dodat je *early_stopping* parametar koji sprečava preprilagođavanje time što zaustavlja obučavanje modela ako se rezultat na test skupu ne poboljša posle zadatog broja iteracija.

Parametar	Home Credit	German Credit
n_estimators	500	500
eta	0.1	0.05
gamma	2	1.5
max_depth	6	6
min_child_weight	3	1
max_delta_step	5	10
subsample	0.8	0.8
colsample_bytree	0.6	0.6
lambda	1	1
tree_method	auto	auto

Tabela 2: Hiperparametri ekstremnog gradijentnog pojačavanja

Model	AUC	Preciznost	Odziv	F1
Logistička regresija	0.74	0.16	0.66	0.26
Logistička regresija (WoE)	0.67	0.15	0.67	0.25
Slučajna šuma	0.73	0.27	0.03	0.06
XGBoost	0.75	0.49	0.05	0.08

Tabela 3: Rezultati na Home Credit uzorku

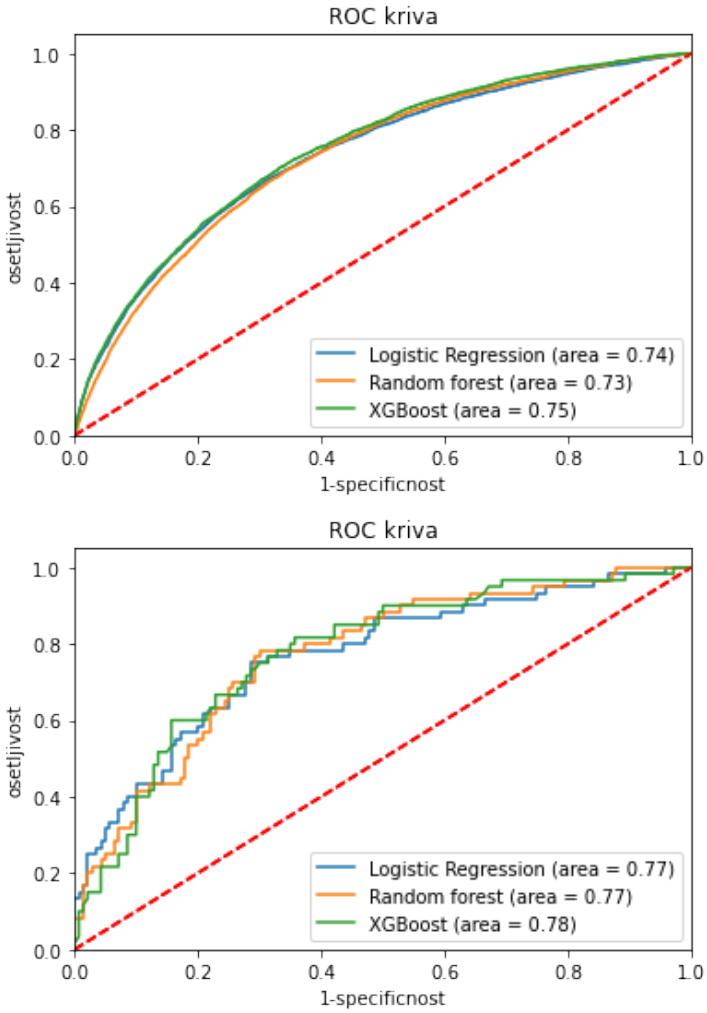
Iz tabele vidimo da na oba skupu ekstremno gradijentno pojačavanje postiže najbolje rezultate po AUC kriterijumu. Nešto lošije rezultate postižu slučajna šuma i logistička regresija. Logističke regresije sa primenom WoE transformacije ima nešto manju površinu ispod ROC krive u odnosu na logističku regresiju bez primene WoE transformacije, dok su po ostalim parametrima dosta slične. Takođe slične performanse pokazuju i slučajna šuma i XGBoost. Primećujemo da je kod logističke regresije na Home Credit uzorku odziv nešto veći od preciznosti, dok je kod ansambla obrnuto. Logistička regresija je mnogo više opservacija klasifikovala kao difoltere pa je time i više pogodila, dok su ansambli bili dosta oprezniji pri označavanju klijenta kao difoltera, ali ipak pogodili približno polovinu od tih kojima su

Model	AUC	Preciznost	Odziv	F1
Logistička regresija	0.77	0.51	0.70	0.59
Logistička regresija (WoE)	0.73	0.53	0.75	0.62
Slučajna šuma	0.77	0.57	0.65	0.60
XGBoost	0.78	0.53	0.73	0.62

Tabela 4: Rezultati na German Credit uzorku

dodelili visoku verovatnoću difolta. Naravno, jasno je da su rezultati dobijeni pod pretpostavkom balansiranosti klasa, pa je tako granica 0.5 za klasifikovanje klijenta kao difoltera. U realnom scenariju, već pri verovatnoći od 0.05 ili 0.1 bismo klijenta smatrali za visoko rizičnog. Zbog toga nam je AUC najbitnija mera prediktivnosti modela.

Na slici ispod 12 prikazano je poređenje ROC krivih za tri različita modela.



Slika 12: ROC kriva (gore Home Credit, dole German Credit)

8 Zaključak

Kroz ovaj rad upoznali smo se sa problemom kreditnog rizika. Ispitali smo različite algoritme za rad sa binarnom klasifikacijom i videli da je popularnost ekstremnog gradijentnog pojačavanja opravdana. Takođe, uočili smo da neinterpretabilnost modela više nije prepreka za primenu naprednijih algoritama i da postoje značajni pomaci u nauci u pravcu raskrinkavanja takozvanih crnih kutija (eng. black box). Postoji još dosta algoritama čije performanse se mogu ispitati na problemu predviđanja difolta, međutim, algoritmi predstavljeni u ovom radu daju okvir različitih pristupa datom problemu. U daljem istraživanju, mogu se ispitati performanse neuronskih mreža kao što je to urađeno u radu [25].

Literatura

[1] German credit data. https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data).

[2] Homecredit. https://www.homecredit.net/about-us/our-products.aspx.

[3] Kaggle. https://www.kaggle.com/.

[4] Zakon o zaštiti podataka o ličnosti. https://www.minrzs.gov.rs/sites/default/files/2018-11/Zakon%20o%20zastiti%20podataka%20o%20licnosti.pdf.

[5] Raymond Anderson. *The credit scoring toolkit: theory and practice for retail credit risk management and decision automation*. Oxford University Press, 2007.

[6] Prudential Regulation Authority. Supervisory statement 13/13 credit risk: internal ratings based approaches. *London: Bank of England*, 2013.

[7] Katarzyna Bijak and Lyn C Thomas. Does segmentation always improve model performance in credit scoring? *Expert Systems with Applications*, 39(3):2433–2442, 2012.

[8] Niklas Bussmann, Paolo Giudici, Dimitri Marinelli, and Jochen Papenbrock. Explainable machine learning in credit risk management. *Computational Economics*, 57(1):203–216, 2021.

[9] Anupam Chander. The racist algorithm? *Michigan Law Review*, 115(6):1023–1045, 2017.

[10] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[11] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[12] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[13] Elena Dumitrescu, Sullivan Hue, Christophe Hurlin, and Sessi Tokpavi. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, 2021.

[14] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[15] Craig K Enders. *Applied missing data analysis*. Guilford press, 2010.

[16] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[17] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[18] Kamil A Grajski, Leo Breiman, Gonzalo Viana Di Prisco, and Walter J Freeman. Classification of eeg spatial patterns with a tree-structured methodology: Cart. *IEEE transactions on biomedical engineering*, (12):1076–1086, 1986.

[19] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[20] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[21] Robert I Jennrich and Stephen M Robinson. A newton-raphson algorithm for maximum likelihood factor analysis. *Psychometrika*, 34(1):111–123, 1969.

[22] Nikita Kozodoi, Johannes Jacob, and Stefan Lessmann. Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*, 297(3):1083–1094, 2022.

[23] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[24] Executive Office of the President. Big data: A report on algorithmic systems, opportunity, and civil rights, 2016.

[25] Salomey Osei, Jules Sadefo Kamdem, Berthine Nyunga Mpinda, and Jeremiah Fadugba. Accuracies of some learning or scoring models for credit risk measurement. 2021.

[26] NAMR Senaviratna, TMJA Cooray, et al. Diagnosing multicollinearity of logistic regression model. *Asian Journal of Probability and Statistics*, 5(2):1–9, 2019.

[27] Lyn C Thomas. *Consumer credit models: pricing, profit and portfolios: pricing, profit and portfolios*. OUP Oxford, 2009.

Biografija autora

Milica Sarić je rođena 11.01.1995. u Valjevu gde je završila Osnovnu školu ,,Andra Savčić", a potom Valjevsku gimnaziju, specijalizovano-matematičko odeljenje. Matematički fakultet, smer Statistika, aktuarska i finansijska matematika, je upisala 2014. godine i diplomirala 2018. Karijeru je započela u Unicredit banci kao saradnik za modeliranje kreditnog rizika. Trenutno je zaposlena u kompaniji Robert Bosch gde uspešno primenjuje znanje stečeno na Matematičkom fakultetu.