Univerzitet u Beogradu, Fakultet organizacionih nauka

Podrazumevano predviđanje kredita L&T vozila

Tim Kaizen - Anđela Čupković 83/19 I Nemanja Vukelić 32/19

1. Opis I razumevanje problema

Korišćeni dataset obuhvata podatke finansijskih institucija koje trpe značajne gubitke zbog kašnjenja povraćaja davanih kredita za vozila. To je dovelo do pooštravanja osiguranja kredita za vozila I povećane stope odbijanja kredita za vozila. Same institucije iskazuju potrebu za boljim modelom bodovanja kreditnog rizika. Tim Kaizen je unajmljen od strane finansijske institucije da tačno predvidi verovatnoću da zajmoprimac ne plati zajam za vozilo u jednakim mesečnim ratama do zadatog roka.

2. Opis I razumevanje podataka

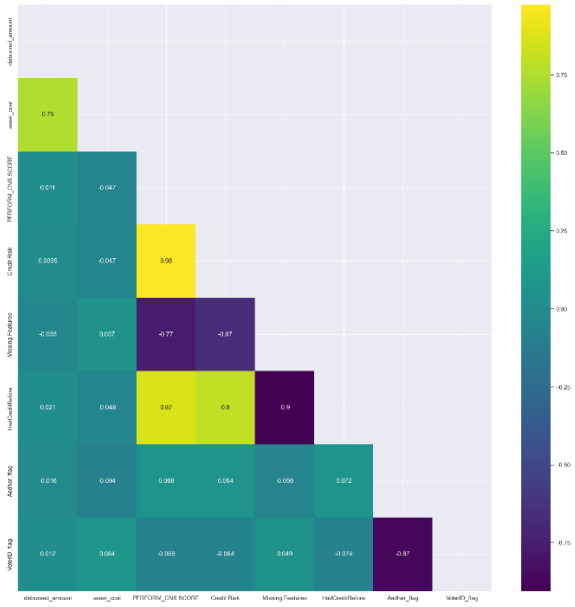
2.1. Opis atributa

Atribute koji sačinjavaju dataset koji smo dobili na korišćenje ugrubo možemo da podelimo u tri kategorije: informacije o zajmoprimcu (*Employment.Type, DisbursalDate, State_ID*…), informacije o kreditu (ltv) kao i podaci o istoriji biroa (*PERFORM_CNS.SCORE, AVERAGE.ACCT.AGE, PRI.CURRENT.BALANCE, PERFORM_CNS.SCORE.DESCRIPTION*…). To će nam osigurati da klijenti sposobni za otplatu kredita ne budu odbijeni i da se mogu prepoznati važne pravilnosti koje se mogu dalje koristiti za smanjivanje stopa neizvršenja obaveza. Korišćen set podataka sadrži 41 nezavisne varijable I ima 233154 instanci. Izlazna varijabla je *loan_default*, ona je binarna promenljiva, vrednost 0 označava da se predviđa da konkretna osoba neće vratiti kredit u zadatom roku dok vrednost 1 govori da hoće. Za tu promenljivu postoji disbalans u podacima. Naime, za 50611 ljudi se očekuje da neće vratiti kredit (što je 21,7% ukupnog broja instanci) dok se za 182543 očekuje da hoće (78,3%).

2.2. Vizualizacije podataka

Ispitali smo korelacije za sve atribute i najznačajnije izdvojili u poseban grafik. Na slici je prikazano sledeće:

- *disbursed_amount* i *asset_cost* 0.75 - jako pozitivna, izbacujemo *asset_cost* jer se u analizi prediktora pokazala kao lošija
- *PERFORM_CNS.SCORE* i *Credit risk* 0.98 -jaka pozitivna
- *PERFORM_CNS.SCORE* i *HadCreditBefore* 0.87 - jaka pozitivna
- *PERFORM_CNS.SCORE* i *Missing Features* -0.77 - jaka negativna
- *Missing Feature* i *HadCreditBefore* -0.9 - jaka negativna
- *Credit risk* i *HadCreditBefore* 0.8 - izbacujemo HadCreditBefore, jer ima dve jake pozitivne korelacije, a pritom je u analizi prediktora znacajno lošiji bio i od Missing Features i od Credit riska
- *Aadhar_flag* i *VoterID_flag* -0.87 - jaka negativna



3. Priprema podataka

3.1. Nedostajuće vrednosti

Kako bismo mogli da počnemo sa predviđanjima i primenom modela potrebno je prvo pripremiti podatke. Što se nedostajućih vrednosti tiče, samo promenljiva *Employment.Type* je imala 7661 nedotajućih vrednosti a kako je to kategorička promenljiva zamenili smo ih sa najučestalijom vrednošću datog atributa (Self employed).

3.2. Transformacija podataka

U cilju pogodnijih podataka za analizu bilo je potrebno transformisati određene kateogrije. Pre svega pod tim mislimo na pretvaranje kategoričkih promenljivih u numeričke što smo učinili sa promenljivom *Employment.Type*. Koristeći funkciju get_dummies dobili smo dva nova atributa – *Salaried* i *Self employed*. Sudeći po tome da ako instanca ima jednu vrednost u jednoj od ovih atributa da to znači da će vrednost drugog biti 0 uklonili smo atribut Salaried zbog bolje preglednosti svih atributa i nepotrebnog gomilanja istih.

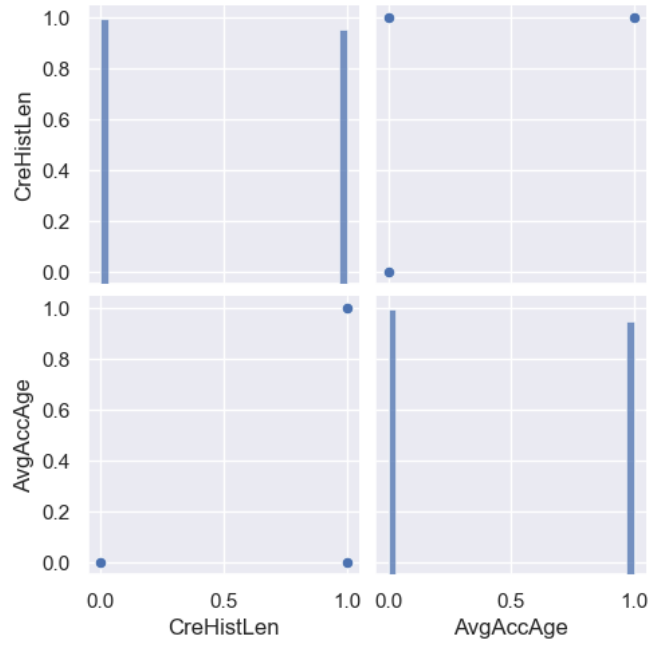
3.3. Izvođenje novih atributa

Atribut *Date.of.Birth* smo zamenili atributom *Age_of_customer* jer nam je informacija koju nam je taj atribut davao bila jasnija kroz broj godina umesto godine rođenja. Takođe smo definisali još jednu funkciju preko koje smo atribute *AVERAGE.ACCT.AGE* i *CREDIT.HISTORY.LENGTH* prebacili iz jedinice godina u jedinicu mesec.

Atribut *PERFORM_CNS.SCORE.DESCRIPTION* je sadržao veliki broj kategorija čiji podaci nam nisu dostupni zbog čega smo uradili preimenovanje dostupnih deskripcija. Naime, cilj nam je bio da svaka od kategorija bude karakteristična po određenom karakteru, tako da smo umesto opisa koji se nalaze u koloni *PERFORM_CNS.SCORE.DESCRIPTION*, kreirali novi atribut credit_risk_grade. Sledeći korak u adekvatnijoj interpretaciji datog atributa je bila novokreirana varijabla Credit Risk, koja je za svaku vrednost atributa credit_risk_grade imala odgovarajuću numeričku interpretaciju. Kako Credit Risk na pojednostavljen način odražava ono što želimo da postignemo, a to je rang kreditnog rizika, ostavljamo samu tu varijablu, dok prethodne dve uklanjamo iz data seta.

Prilikom analize izlazne promenljive *loan_default* i novokreirane varijable *Credit Risk* možemo da uočimo da klijenti imaju vrednost 0 iz dva potpuno drugačija razloga - ili je izlazna promenljiva 0 kao rezultat nepostojanja kreditne istorije ili posedovanja kreditne istorije u kojoj je zabeleženo da zaduženje nije otplaćeno. Zato uvodimo novi atribut *Missing Features* kako bismo omogućili modelu da razgraniči te dve grupe ljudi. Klijenti koji nemaju kreditnu istoriju imaće veću vrednost ovog atributa, jer će im za više atributa faliti vrednost.

Upoređivanjem broja vrednosti za atribute *AVERAGE.ACCT.AGE* i *CREDIT.HISTORY.LENGTH* možemo da primetimo da i jedna i druga promenljiva za gotovo 50% opservacija imaju trajanje od 0 godina i 0 meseci, te je smisleno uvesti novu promenljivu koja će nam dati informaciju o tome da li je neko pre imao kredit ili ne. Proveravamo korelisanost ova dva atributa da bismo u korist novokreiranog izbacili navedene. Prvo smo uveli binarne promenljive *AvgAccAge* I *CreHistLen* koje sui male vrednost 0 ukoliko je 0 bio broj godina I meseci, a ukoliko nije se unosila vrednost 1 u pomenute parametre. Zatim smo prešli na korelaciju.




```
[[1.         0.99789092]
 [0.99789092 1.        ]]
```

Uviđamo da su date varijable u potpunosti kolerisane, te pravimo novi atribut *HadCreditBefore* i izbacujemo ova dva kao i atribute od kojih smo i počeli sa ovim.

3.4. Analiza outlier-a

Napravili smo funkciju za pronalaženje outlier-a koja radi po principu box plot-a samo finalni rezultat piše u vidu teksta i brojeva umesto grafikona. Pomoću nje smo ispitali sve atribute i ustanovili da 15 od 23 instance ,koje smo imali u tom momentu nakon odrađenih prethodnih koraka i koraka 5, sadrži outlier-e. Time smo ustanovili da treba biti posebno pažljiv prilikom pokretanja modela koji su osetljivi na ekstremne vrednosti.

4. Treniranje više algoritama I interpretacija dobijenih rezultata

Od algoritama smo hteli da koristimo stablo odlučivanja, naivnog Bajesa, logističku regresiju I KNN. Pokušali smo prvo da normalizujemo raspodele vrednosti svih atributa, ali nam je to rezultiralo gorim performansama. Najbolji krajnje performanse nam je dala selekcija 20 najznačanijih atributa. Normalizacija bi nam bila neophodni korak pre primene knn-a ali smo odlučili da ovaj algoritam ne razradimo dodadno zbog prevelikog broja opservacija i atributa i malih brzina. Dodatno smo uradili Feautre selection dodavanjem VarianceTreshold-a čime smo izvršili selekciju atributa na način da smo zadali prag (threshold) varijanse na 0.05, pri čemu svi atributi čija varijansa ne prelazi datu granicu neće biti validni za dalju analizu i stoga će biti izostvaljeni iz skupa podataka. Poredili smo putem kros-validacije rezultate modela pre I posle uvođenja pomenutog feature selection-a I većinom su rezultati bili slični ili bolji. Najveći pomak smo imali kod logističke regresije gde se roc_auc sa 49,63% popeo na 52,183%.

roc_auc je kod naivnog Bajesa imao rezultate 56,876% I 57,766%. Pokušali smo da poboljšamo performanse koristeći Wrapper feature selector. Nakon toga rezultat je bio 58,583% što, iako nije značajno poboljšanje, nam govori da smo jako dobro odradili osnovnu pripremu podataka. Sa ovom selekcijom atributa ćemo dalje raditi naivnog Bajesa dok ćemo ostale modele raditi sa sređenim podacima pre ovog koraka.

5. Selekcija atributa I interpretacija dobijenih rezultata

Samom analizom opisa svakog atributa u dataset-u smo zaključili da atributi *UniqueID*, nije relevantan za analizu tako da smo ga izbacili iz daljeg opticaja.

Sledeća metoda koju smo primenili jeste pregled broja različitih vrednosti koje su se našle u svakoj koloni. Time smo videli da atribut DisbursalDate (identifikator godine isplate) ima samo jednu vrednost – 2018 na osnovu čega ne možemo izvršiti diferencijaciju klijenata zbog čega smo I taj atribut izbacili.

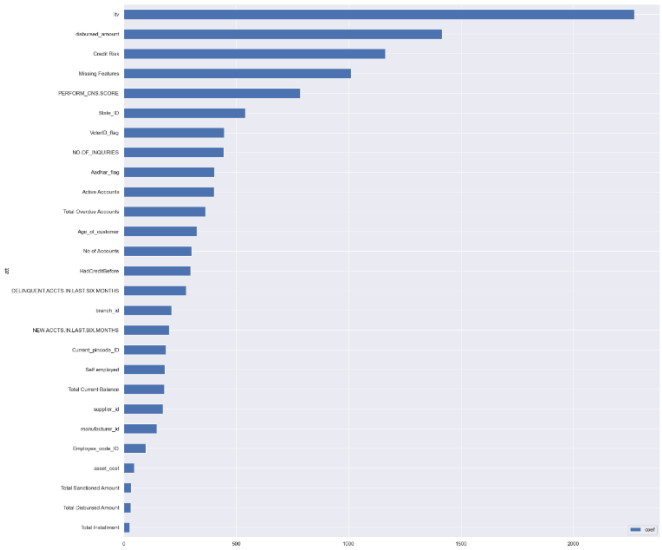
Potom smo pregledali sve kolone I izvršili normalizacije njihovih vrednosti. Time smo utvrdili da sve opservacije atributa *MobileNo_Avl_Flag*

imaju samo jednu vrednost zbog čega je i taj atribut izbačen.

5.1. Provera važnosti numeričkih prediktora

Pomoću SelectKBest-a smo Ispitali u kojoj meri je svaki od atributa korelisan sa izlaznom promenljivom i poređali ih po važnosti.

Na osnovu dobijenog grafa videli smo da su svi podaci o sekundarnim nalozima lošiji prediktori, pa smo onda njih kombinovali zajedno sa informacijama o primarnim nalozima. Nakon toga smo ponovili grafik (na sledećoj strani), videli da su prediktori ovoga puta bolji ali I da imamo atribute *disbursed_amount* i *Total Disbursed Amount* koje nose istu informaciju. Izbacili smo *Total Disbursed Amount* jer je lošiji prediktor



6. Optimizacija parametara I interpretacija dobijenih rezultata

6.1. Randomized I Grid Search

Korišćenjem neiscrpne pretrage, čiji je cilj da se pronađu optimalne vrednosti hiperparametara, ali tako da se ne pokušaju sve kombinacije već samo pojedine dobili smo da će naš model najbolje performanse postići ako su hiperparametri C: 0.3935 i penalty: l2. Kad uporedimo perforamanse sa početnim modelom logističke regresije vidimo da se one jesu malo poboljšale, roc_auc sa 0.5 na 0.52.

Korišćenjem GRID pretrage koja isprobava sve kominbacije hiperparametara koje smo zadali, dobili smo da će naš model najbolje performanse postići ukoliko su hiperparametri max_depth: 29, max_features: 17, min_samples_leaf: 26}. Kada uporedimo performanse optimizovanog modela sa početnim vidimo da se roc_auc poboljšala sa 0.87 na 0.96 i accuracy sa 0.85 na 0.92

6.2. Ansambl algoritmi

Napravili smo bagging ansambl za 5 istih modela tree1, razlika je u tome što svaki model uči na drugačijem podskupu podataka, pri čemu su modeli specijalizovani za određene podskupove - svaki će malo drugačije glasati jer ima različito iskustvo, to jest podatke na kojima se uči. Zahvaljujući tome smo uspeli da i već prethodno odlične performanse postanu još bolje. Sad ćemo videti da li na ovaj način možemo poboljšati rad algoritma koji nije imao tako dobre performanse.

U odnosu na stablo odlučivanja sa dubinom 3, čiji roc_auc je iznosio 0.8681, Random Forest je dao bolje rezultate. U odnosu na stablo kom nismo podešavali hipeparametre, Random Forest je dao za nijansu bolje rezultate. Što se tiče stabla sa podešenim optimalnim hiperparametrima, čiji roc_auc iznosi 0.9589, možemo da zapazimo da je Random Forest dao nešto lošije rezultate. Međutim, možemo da povećamo broj stabala koji čine šumu, budući da je pretreniranje kao takvo retko jer povećanjem broja modela mi zapravo umnožavamo podatke.

U pogledu roc_auc imamo bolje rezultate primenom Ada Boost-a, međutim kod accuracy imamo za nijansu lošije rezultate. Upoređivali smo sa modelom kod koga su podešeni optimalni hiperparametri. Accuracy kod modela nb je iznosio 0.7828774106277975, a roc_auc 0.5776622578682757, tako da ne možemo primetiti poboljšanja u peformansama kada je u pitanju

upotreba Ada Boost-a pri čemu modeli koji u njemu učestvuju predstavljaju Naivnog Bajesa.

Takođe smo primenili Gradient boosting, rezultati su bili odlični ali ne na nivou Random Forest-a. Potom smo utvrdili važnost svih od preostalih atributa i uvideli da atributi Age_of_customer, manufacturer_id, Aadhar_flag, Employee_code_ID ne igraju značajnu ulogu u predviđanju zbog čega smo ih izbacili i pokrenuli model bez njih. Performanse se jesu malo poboljšale.

7. Zaključak I ideje koje bismo realizovali da smo imali više vremena

7.1. Model koji bismo koristili

Na osnovu ove analize može se preporučiti korišćenje Random Forest modela u praktičnoj upotrebi zato što je pokazao najbolje performanse. To ukazuje na činjenicu da je model sposoban da pravilno klasifikuje instance i da ima dobro balansiranu predikciju između obe klase. Takodje, na osnovu rezultata možemo videti da je Random Forest prilično stabilan model i bez normalizacije i sa izborom atributa različitim metodama selekcije, kao i optimizacijama parametara. To je posebno važno u praktičnoj primeni gde se podaci mogu menajti tokom vremena. Takođe, Random Forest je sposoban da se nosi sa različitim vrstama podataka, uključujući i neuravnotežene skupove.

7.2. Da smo imali više vremena...

... poradili bismo na poboljšanju objašnjivosti i pravednosti modela.

Sudeći po tome da se Random Forest pokazao kao najbolji algoritam, u polju performansi, a kako je on kompleksan ansambl algoritam verovatno da nekad ne bi bilo jasno kako je došao do neke odluke. Takođe bi se moglo desiti da napravi grešku u svojoj odluci ali da se to ne primeti zbog nerazumevanja modela; model treba da bude jasan ljudima koji će ga koristiti a ne samo tehničkim ekspertima, u suprotnom može doći do toga da korisnici prestanu da mu veruju. Ovo bismo rešili radeći na post-hoc modelu koji bi služio da shvati random forest algoritam i potom objasni kako se donela neka odluka. Prvo bismo radili na globalnoj slici a potom na lokalnoj.

Zatim, vrlo je bitno uočiti da li je model, učeći se na dataset-u u kom se sadrže prethodno donošene odluke o predviđanju toga da li će neko vratiti kredit ili ne, ne stekne pristrasnost prema određenim podacima. Ako i dođe do toga želimo da pronađemo uzrok pristrasnosti kako bismo ga rešili (i ujedno kako ne bismo dobili neku kaznu). Jedan od načina da rešimo pravednost jeste otežavanje (i oslabljivanje) određenih instanci ali to možemo uraditi podešavanjem konkretnog parametra u sklopu Random Forest-a. Ukoliko bismo nastavili da radimo na ovome opredili bismo se za drugi način a to je dodavanje ograničenja na pravednost (constraint optimization).