# Uvod i opis problema

U poslu izdavanja kredita banke imaju veliki problem sa odobravanjem klijenata. Da bi banka bila konkurentna ona mora ponuditi dovoljno niske kamatne stope da ide u korak sa ostalim finansijskim institucijama, a opet veoma delikatno ostvariti profit. Odabir klijenata je ključan jer se procenjuje da jedan neuspeli platiša košta banku poslovanja između 7 do čak 13 uspešnih kredita. Dakle za samo jedan ne vraćen kredit banka treba da nadoknadi gubitke kroz 7+ uspešnih poslova. Stoga je možda i najbitniji posao banke da identifikuje dobre platiše u postupku odobravanja kredita.

U ovom zadatku se susrećemo sa L&T (Larsen & Toubro Limited) korporacijom koja se između ostalog bavi i finansiranjem kupovine vozila. Iz kompanije se nadaju da će našim angažovanjem dobiti pouzdaniji način predviđanja budućih klijenata i njihove mogućnosti da vrate kredit. Tom prilikom su nam isporučili podatke sa raznim atributima o klijentima (lični podaci, podaci iz banke, podaci iz kreditnog biroa itd.).

Uspešna realizacija ovog projekta bi osigurala da će veći broj klijenata koji mogu da otplate kredite biti odobreni, a oni koji nisu u stanju to da učine biti odbijeni. To vodi ka boljoj situaciji za sve učesnike: banku, odobrene i neodobrene klijente kao i društvo kao celinu sa boljom alokacijom novca.

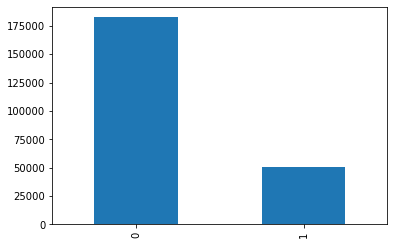
# Opis podataka

Set podataka koji dobijemo je jako obiman: sastoji se iz dva fajla: train i test. Razlika između njih je u veličini kao i u tome što target varijabla nedostaje u test podacima. Fokus će biti na train podacima.

Oblik train podataka se sastoji od 233 154 redova i 41 kolone (od kojih je jedna 'loan default' – target varijabla koja nam govori da li je taj klijent propustio da plati prvu mesečnu ratu kredita u predviđenom roku). Atribute možemo podeliki u nekoliko kategorija: podaci o klijentu, podaci o uzetom kreditu, informacije koje je dao banci, podaci iz kreditnog biroa, podaci o prethodno uzetim kreditima za dve vrste računa (primarni i sekundarni) kao i neki izračunati podaci o klijnetu (npr. broj kredita u poslednjih 6 meseci itd.).

## Target atribut – ‘loan\_default’

U samom target atributu postoji jasan disbalans koji se vidi sa slike. Ovaj disbalans će morati da se ispravi kako bi prediktivni modeli postizali dobre rezultate.



## Nedostajuće vrednosti

U našim podacima jedina varijabla koja ima nedostajuće vrednosti je ‘employment\_type’. Naime u tom atributu ima 7661 NA vrednosti, pored vrednosti ‘self\_employed’ i ‘salaried’. U razmatranju kako da se rešimo datih NA vrednosti odlučeno je da njima dodelimo vrednost ‘self\_employed’ kao dominantnu vrednost atributa. Razmatrano je i obacivanje datih 7661 redova ali radi očuvanja podataka nismo se odlučili za taj korak.

# Priprema podataka

Karakteristika ovog seta podataka je da je relativno obiman. Moramo ga inženjeringom atributa nekako načiniti manjim kako bi modelovanje oduzelo ne preveliki period vremena.

## Odbacivanje nerelevantnih atributa

U setu podataka imamo određene kolone koje se logički javljaju kao kandidati za odbacivanje. To su: 'uniqueid','branch\_id','supplier\_id','mobileno\_avl\_flag','current\_pincode\_id','employee\_code\_id','manufacturer\_id','state\_id'. Neki od ovih imaju jasan razlog za ne korišćenje jer imaju samo jednu vrednost, ali ono što im je zajedničko jeste da se ovi podaci ne koriste za determinaciju odobrenja kredita. Podaci kao što su broj filijale, id dilera automobila, poštanski broj, id zaposlenog ko je proknjižio podatke, marka vozila i savezna država ne spadaju podatke koje banke tipično koriste za obradu kredita te u duhu olakšvanja podataka ćemo ih zanemariti.

## Rešavanje datumskih promenljivih

Imamo dve datumske promenljive a to su ‘date\_of\_birth’ i ‘disbursalDate’ koji predstavljaju datum rođenja i datum isplate kredita na račun. Naše rešenje za ova dva atributa je da ih ukombinujemo u jedan atribut koji bi predstavljao godine klijenta na dan isplate kredita na račun, a to dobijemo pretvaranjem tih promenljivih u godine i njihovom razlikom. Time ove dve ‘object’ promenljive predvaramo u numeričku, dobijamo smisleniji atribut za rad i smanjujemo broj atributa za jednu kolonu.

## Rešavanje str atributa u formatu ‘1yrs 1mon’

U setu podataka susrećemo se sa dva atributa koja su izražena u formatu ‘1yrs 11mon’. To su atributi "average\_acct\_age" i "credit\_history\_length". Oni ozačavaju prosečno trajanje korisnikovog računa/naloga za otplatu, kao i vremenski period njehove kreditne istorije. Korišćenjem funkcije za manipulaciju stringova lako dobijamo numeričke vrednosti koje izražavaju ove atribute u količinu meseca umesto inicijalnog formata.

## Atribut ‘perform\_cns\_score\_description’

Ovaj atribut je specifičan jer je jedini zaista deskriptivni atribut sa većim brojem vrednosti. On predstavlja opisnu ocenu kreditnog rejtinga korisnika. Skalarno ta ocena ide alfabetnim redom od slova ‘A’ do slova ‘M’ uz koje ide kratka kategorizacija datih ocena u pogledu rizika uz veliki broj neocenjenih iz raznih razloga (njih ćemo posmatrati na isti način). Ispitivanjem datog atributa zaista uočavamo porast procenta loših platiša u skaldu sa ocenama ovog atributa, no zbog sporadičnog prekida rastućeg niza rešili smo da je logičnije da pretvorimo date skalabilne opisne vrednosti u numeričke u vrednostima od -1 do 4 (very high risk :0 , high risk : 1 , medium risk: 2, low risk : 3, very low risk: 4 i neocenjeni : -1).

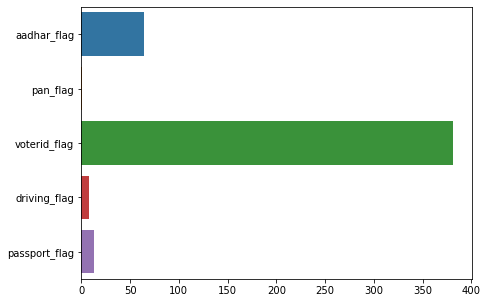
## Atribut ‘employment type’

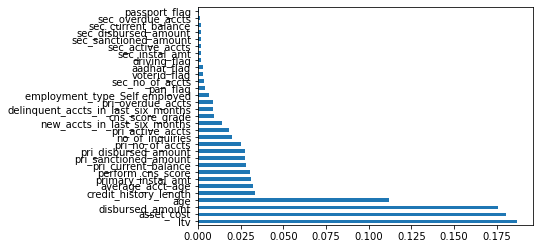
Rešavanjem NA vrednosti ovaj atribut postaje binaran sa string vrednostima ‘salaried’ i ‘self\_employed’. Ovo rešavamo na ustaljen način rada sa ovakvim atributima: pretvaramo ga u numerički atribut sa vrednostima 0 i 1 respektivno početnim vrednostima.

Ovime naš set podataka pretvaramo u potpuno numerički data set.

## Binarni atributi

Među našim podacima imamo podgrupu atributa koji se odnose na činjenice da li je klijent pružio određene lične informacije ili ne ('aadhar\_flag','pan\_flag','voterid\_flag','driving\_flag','passport\_flag'). Na prvi pogled deluje kao da ssu ovo jako nebitni podaci, ali korišćenjem feature\_imporances vidimo da neki od njih imaju uticaj na target varijablu. Tako da smo odlučili da ih ne izbacimo odmah

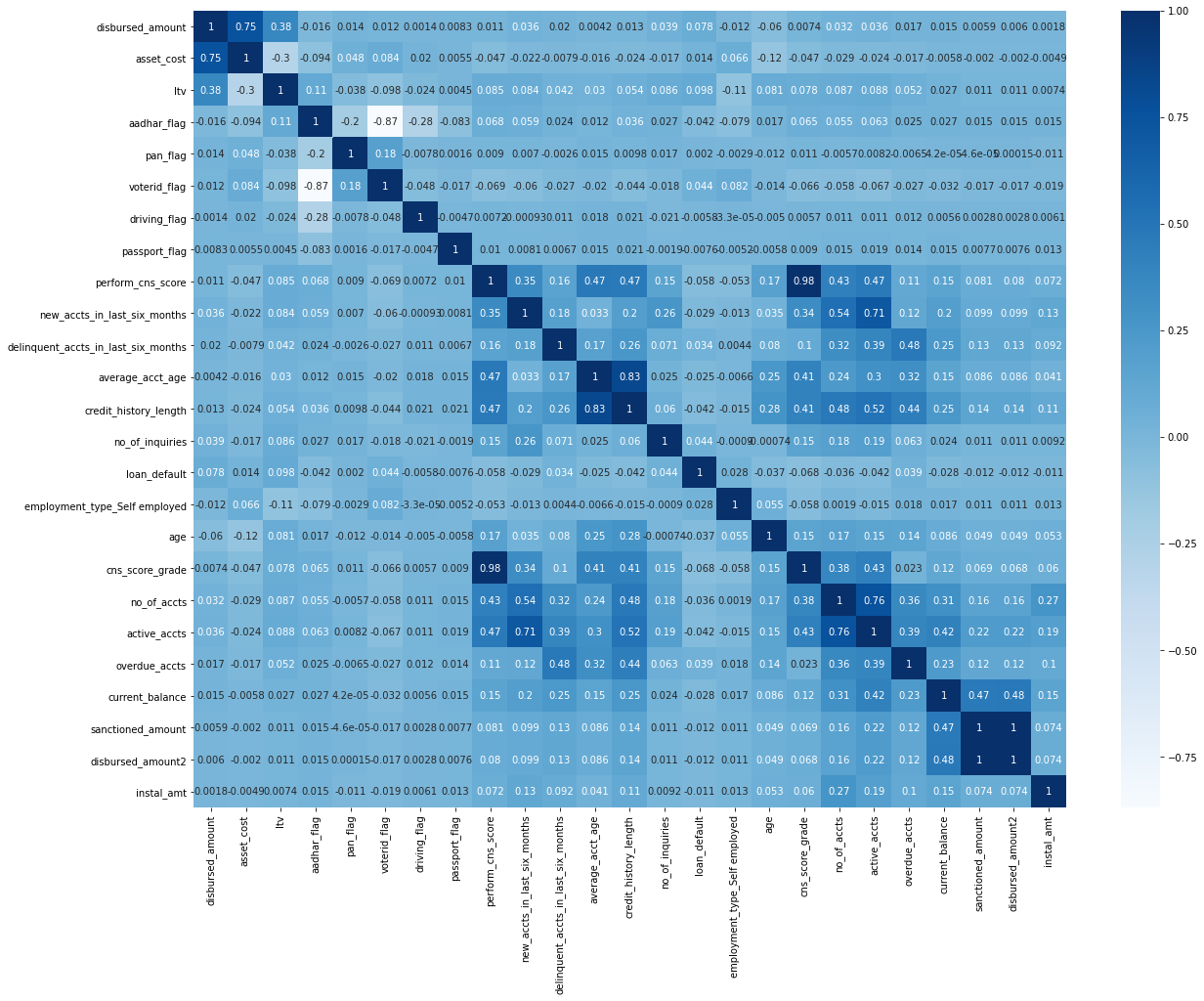




## Spajanje ‘primarnih’ i ‘sekundarnih’ atributima

U podacima su nam date grupe od po 6 atributa koje se odnose na potpuno istu stvar za različite vrste računa. U pitanju su primarni(računi za ličnu upotrebu klijenata) i sekundarni (računi koji nisu samo klijentovi ili za koje je klijen žirant). U smislu naših predikcija nema značajnog razloga za ovakvo odvajanje te ćemo od datih 12 atributa napraviti odgovarajućih 6. To još više dobija smisao ako vidimo da uticaj na izlaznu vrednost sekundarnih atributa je jako mala.

## Korelacije



Vidimo visoku korelaciju izmedju

sanctioned\_amount i disboursed\_amount = 1

cns\_score\_grade i perform\_cns\_score = 0.98

credit\_history\_lenght i avrage\_acct\_age = 0.83

no\_of\_accts i overdue\_accts = 0.76

active\_accts i new\_accts\_in\_the\_last\_six\_months = 0.71

Odbacićemo one atribute prva dva parova koji imaju manju feature importance (gore izracunata) jer su zaista gotovo 100% korelisani, a ostale cemo ostaviti.

## Autlajeri i normalizacija

## Sampling

Dati set podataka je jako veliki za efikasan rad sa bilo kojom metodom klasifikacije, a tek je situacija dramatična kada se zahteva štimovanje hiperparametara (GradiantBoosting je trajao 4h, KNN 50 min), Stoga zarad efikasnijeg korišćenja vremena uzelimo smo nasumično 10% sample na kome ćemo trenirati modele mašinskog učenja za klasifikaciju.

## Balansiranje podataka

Kao što smo napomenuli značajno je skup podataka u disbalasnu. Pokušavši da radimo sa ne balansiranim podacima dobijali smo jako loše i jednostrane rezultate. Balansiranje smo izvršili koristeći funkciju SMOTE over samplinga nad našem skraćenom skupu podataka.

# 3) Pravljenje modela i predikcija

## KNN

Bez posebnog štelovanja hiperparametara (samo ručno)korišćenjem algoritma knn dobili smo solidne rezultate. Odlična je stvar što je recall jako visok. Ako prihvatimo pretpostavku da je False negative mnogo skuplja greška nego FP onda KNN najbolje radi od naredna 3 algoritma. Korišćenjem GridSearchCv-a smo znatno unapredili metrike izborom hiperparametara.

**1)K Nearest Neighbour(weights='distance', n\_neighbors=15, n\_jobs=4)**

accuracy\_score 74.04456227660159 precision\_score 67.98894398610508  
recall\_score 91.6530150922303 f1\_score 78.0665562072336

roc\_auc\_score 73.9035162041238  
confusion\_matrix  
 0 1  
0 1013 791  
1 153 1680

**2)KNeighborsClassifier(weights='distance', metric='manhattan', n\_neighbors=3, n\_jobs=4)**

accuracy\_score 79.68105581523234 precision\_score 73.53700516351118  
recall\_score 93.23513366066557 f1\_score 82.22275679576619

roc\_auc\_score 79.57211228487824  
confusion\_matrix  
 0 1  
0 1189 615  
1 124 1709

## GradientBoostingClassifier

Bez posbnog nameštanja algoritama smo dobili jako dobre rezultate korišćenjem Gradient Boosting klasifikatora (1) dobili smo solidne rezultate uz visoku preciznost ali malo niži recall. Korišćenjem GridSearchCV dobili smo optimalne vrednosti hiperparametara i blago poboljšali rezultate pretrage.

**1)GradientBoostingClassifier(n\_estimators=100, min\_samples\_split=5, max\_depth=15)**

accuracy\_score 83.91531481990651 precision\_score 87.77239709443099  
recall\_score 79.10529187124932 f1\_score 83.21377331420372  
  
roc\_auc\_score 83.95397631256479  
confusion\_matrix:  
 0 1  
0 1602 202  
1 383 1450

**2)GradientBoostingClassifier(n\_estimators=100, min\_samples\_split=2, max\_depth=15)**

accuracy\_score 84.05279076161672 precision\_score 87.80929390464695  
recall\_score 79.37806873977087 f1\_score 83.38108882521492

roc\_auc\_score 84.09036474682556  
confusion\_matrix  
 0 1  
0 1602 202  
1 378 1455

## Random Forrest

**1)RandomForestClassifier(n\_estimators = 100, max\_depth=15, min\_samples\_split=5, min\_samples\_leaf=3)**

accuracy\_score 73.46714324993127 precision\_score 70.0  
recall\_score 80.6036892118502 f1\_score 74.92855287087555  
roc\_auc\_score 73.58106538514588  
confusion\_matrix  
 0 1  
0 1230 618  
1 347 1442

**2)RandomForestClassifier(n\_estimators = 100, max\_depth=25, min\_samples\_split=3, min\_samples\_leaf=3, max\_features= 4)**

accuracy\_score 77.75639263128953 precision\_score 75.33609100310238  
recall\_score 81.44214645053103 f1\_score 78.27021219446682  
roc\_auc\_score 77.81522906942135  
confusion\_matrix  
 0 1  
0 1371 477  
1 332 1457

## Feature\_importances

Obzirom da smo se feature inženjeringom bavili značajno u pripremi podataka došli smo do situacije da samo izbacivanje atributa ‘passport\_flag’ daje neznatno bolje rezultate. To smo i učinili ali promena u rezultatima je ostala praktično ne promenjena.

# 4) Zaključak

Iako je bilo zadovoljstvo raditi na ovako izazovnom projektu za naš nivo znanja susretali smo se sa nekoliko prepreka: vreme, nivo znanja i računarska moć se jasno ističu.   
Neke ideje koje smo smatrali da smo možda mogli da ostvarimo su da pokušamo nešto pametnije da uradimo sa geografskim podacima. Naime, atribut ‘current\_pincode’ koji govori o poštanskom broju klijenta mislimo da može da se poveže sa ekonomskim statusom. Ukoliko bismo mogli da nađemo podatke o socio-geografskoim karakteristikama idije mogli bismo da klasterujemo ove podatke u nekoliko klastera koji bi davali značajne informacije o socio-ekonomskom statusu korisnika. To bi gotovo sigurno bio veoma značajan atribut u našem zadatku.