Analiza skupa podataka Loans Default Dataset

Stefan Kerkoč

Uvod

U ovom izveštaju prikazan je proces istraživanja podataka, preuzetih sa Kaggle web strane, a koji sadrže informacije o bankovnim kreditima. Cilj istraživanja je izrada modela koji predviđaju da li ce neko biti u mogućnosti da otplati kredit na osnovu 32 atributa.

Podacima možete pristupiti klikom na link

Ekspolativna analiza podataka

Pre pretprocesiranja, potrebno je da uradimo analizu podataka. Nažalost naši atributi dolaze bez pojašnjenja šta koji od njih znači. Ipak, za neke je moguće pretpostaviti, a kompletna lista atributa je sledeća:

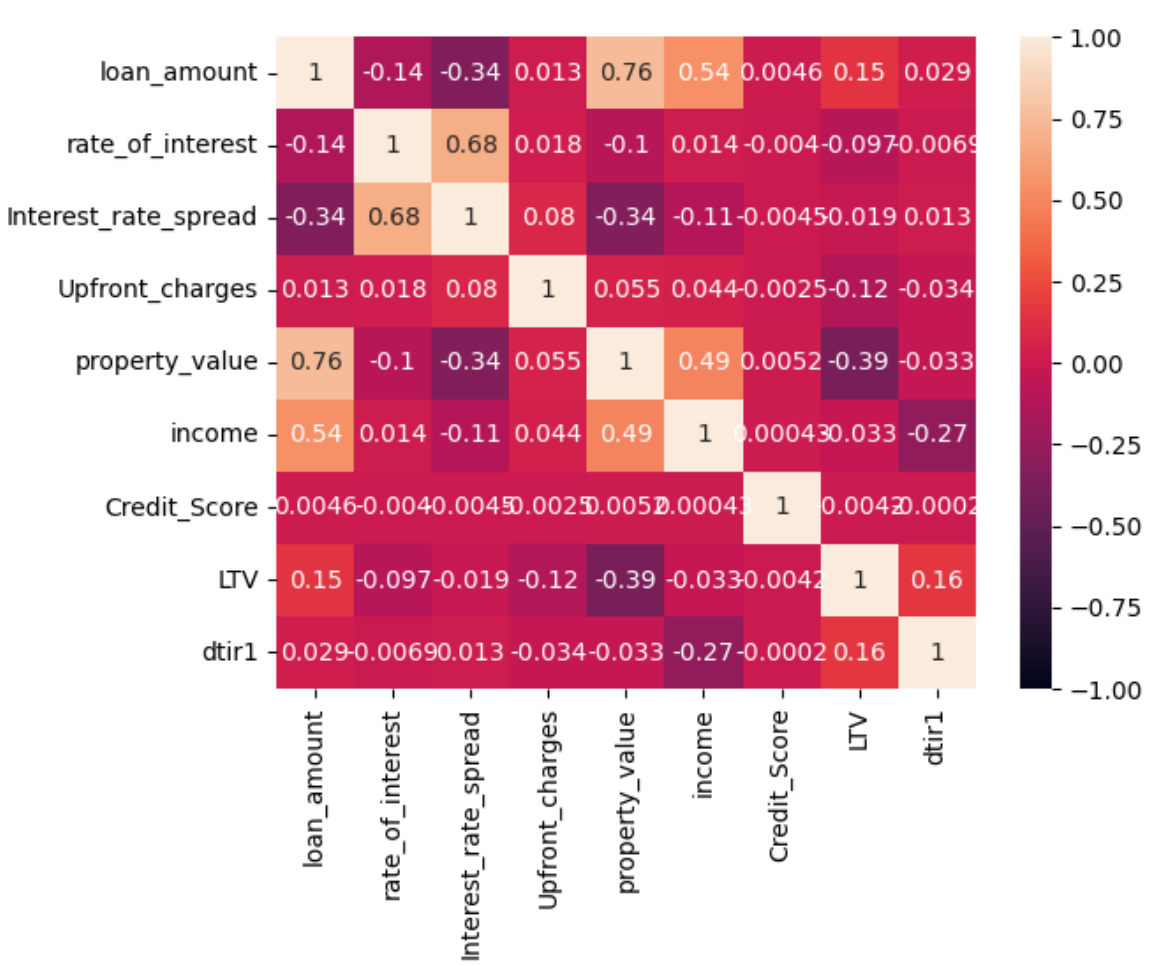
- loan_limit
- Gender
- approv_in_adv
- loan_type
- loan_purpose
- Credit_worthiness
- open:credit
- business_or_commercial
- loan_amount
- rate_of_interest
- Interest_rate_spread
- Upfront_charges
- term
- Neg_ammortization
- interest_only
- lump_sum_payment
- property_value
- construction_type
- occupancy_type
- Secured_by
- total_units
- income
- credit_type
- Credit_score
- co-applicant_credit_type
- age
- submission_of_application
- LTV
- Region
- Security_Type
- dtir1
- Status

Prva stvar koju treba da uradimo je da odredimo tip atributa, a to možemo uraditi tako što pogledamo pod kojim tipom podataka se čuvaji i, značajnije, broj jednistvenih vrednosti.

```
2. year has 1 unique values, which are [2019]
3. loan_limit has 3 unique values, which are ['cf' nan 'ncf']
4. Gender has 4 unique values, which are ['Sex Not Available' 'Male' 'Joint' 'Female']
5. approv_in_adv has 3 unique values, which are ['nopre' 'pre' nan]
6. loan_type has 3 unique values, which are ['type1' 'type2' 'type3']
7. loan_purpose has 5 unique values, which are ['p1' 'p4' 'p3' 'p2' nan]
8. Credit_Worthiness has 2 unique values, which are ['l1' 'l2']
9. open_credit has 2 unique values, which are ['nopc' 'opc']
10. business_or_commercial has 2 unique values, which are ['nob/c' 'b/c']
16. Neg_ammortization has 3 unique values, which are ['not_neg' 'neg_amm' nan]
17. interest_only has 2 unique values, which are ['not_int' 'int_only']
18. lump_sum_payment has 2 unique values, which are ['not_lpsm' 'lpsm']
20. construction_type has 2 unique values, which are ['sb' 'mh']
21. occupancy_type has 3 unique values, which are ['pr' 'sr' 'ir']
22. Secured_by has 2 unique values, which are ['home' 'land']
23. total_units has 4 unique values, which are ['1U' '2U' '3U' '4U']
25. credit_type has 4 unique values, which are ['EXP' 'EQUI' 'CRIF' 'CIB']
27. co-applicant_credit_type has 2 unique values, which are ['CIB' 'EXP']
28. age has 8 unique values, which are ['25-34' '55-64' '35-44' '45-54' '65-74' '>74' '<25' nan]
29. submission_of_application has 3 unique values, which are ['to_inst' 'not_inst' nan]
31. Region has 4 unique values, which are ['south' 'North' 'central' 'North-East']
32. Security_Type has 2 unique values, which are ['direct' 'Indriect']
33. Status has 2 unique values, which are [1 0]
```

Slika 1: broj jedinstvenih vrednosti

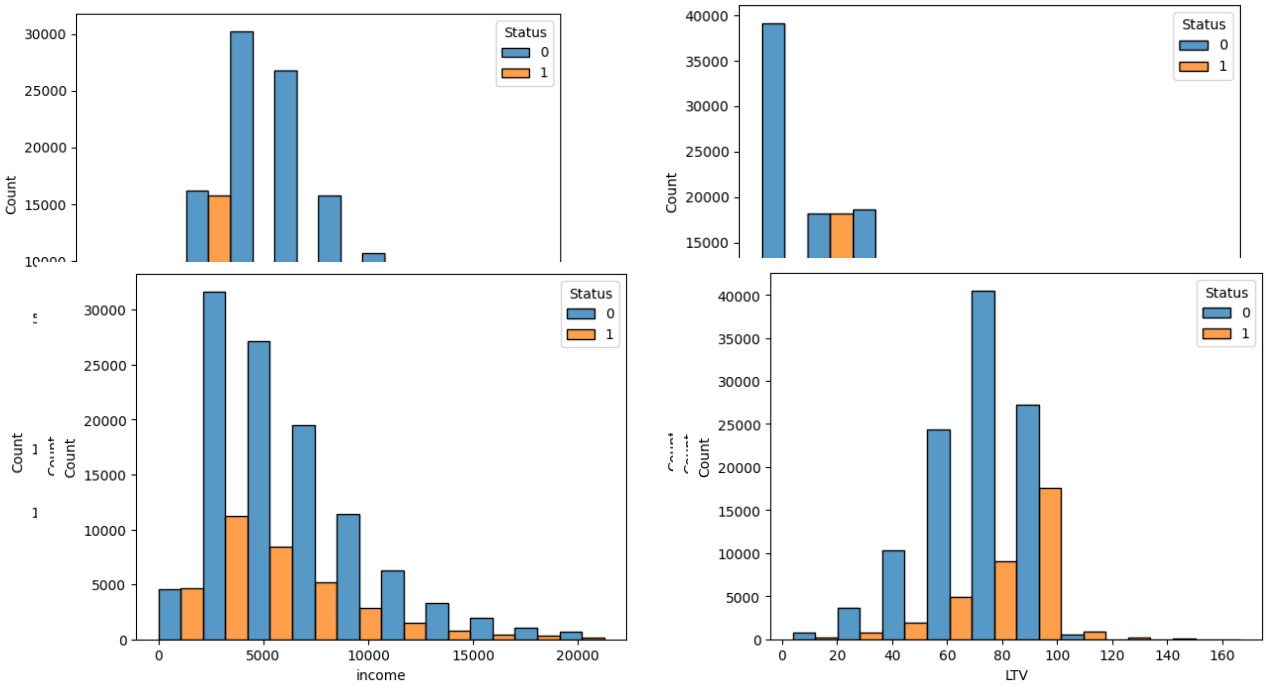
Nakon pretprocesiranja, o kojem će biti reči kasnije, možemo proveriti korelacije između ulaznih atributa

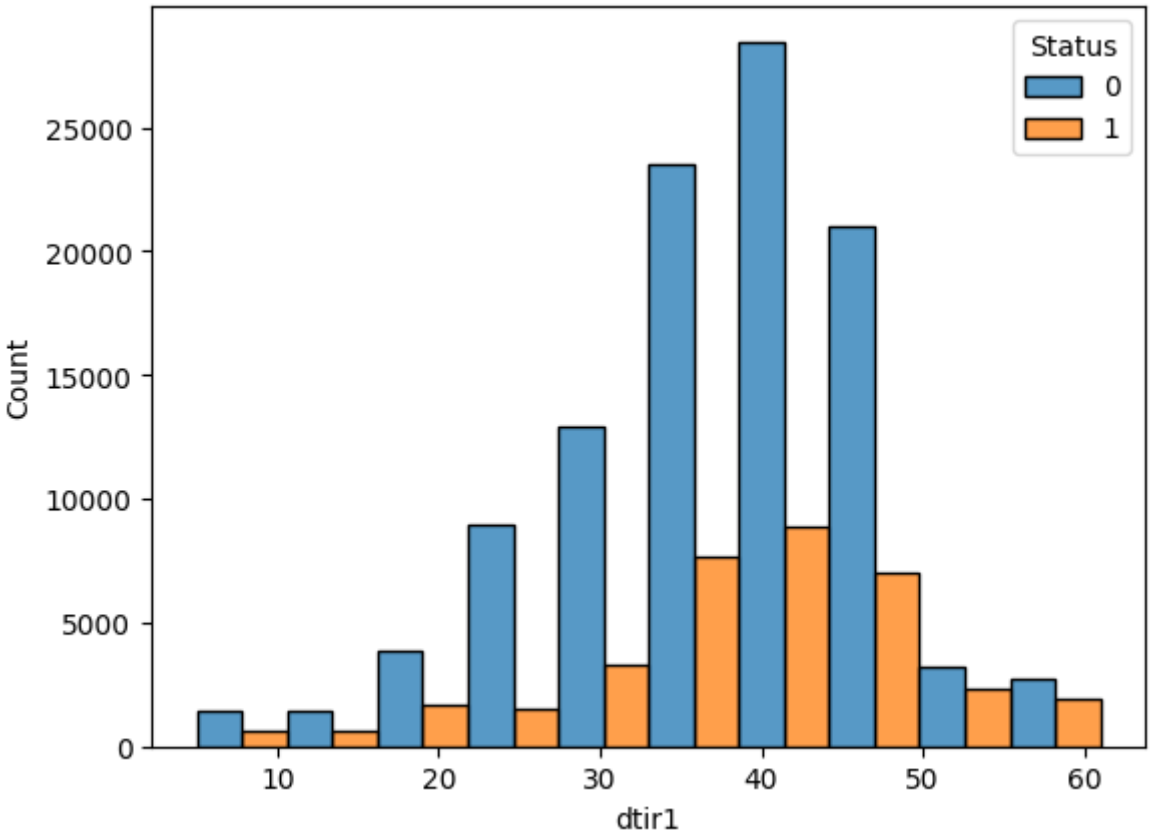


Slika 2: mapa korelacija

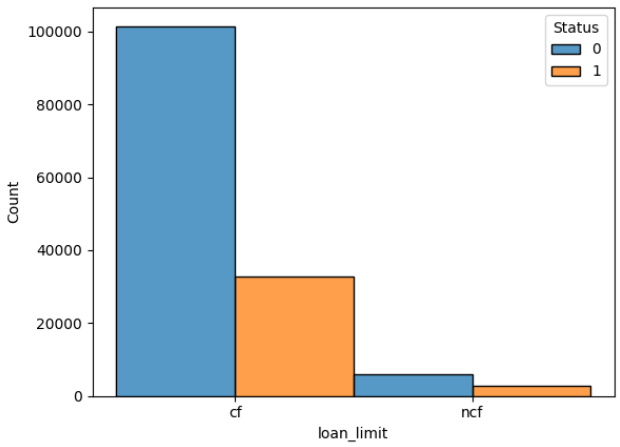
Vidimo da atributi property_value, income i loan_amount imaju jaku korelaciju, što važi i za atribute rate_of_interest i Interest_rate_spread. Ove činjenice su veoma logične i nisu iznenađujuće.

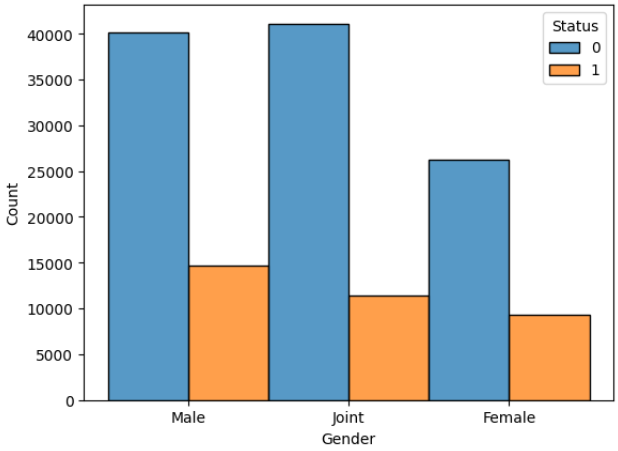
Dalje, želili bi da vidimo kako naš ciljni atribut zavisi od pojedinačnih ulaznih atributa.

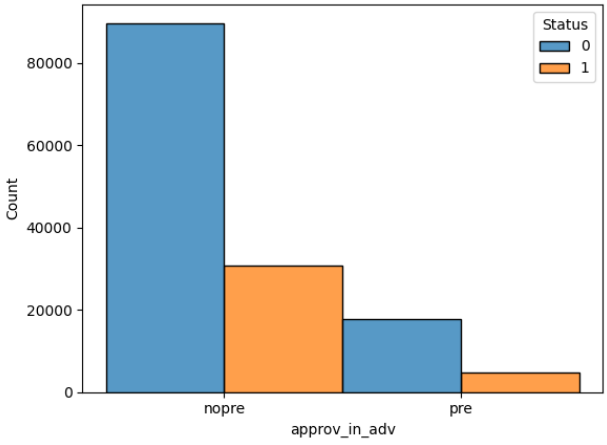


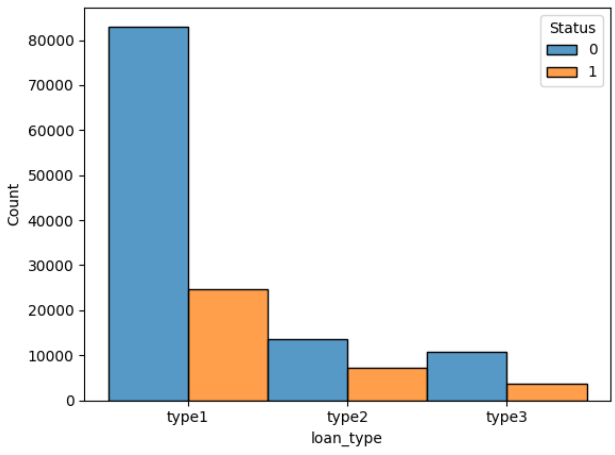


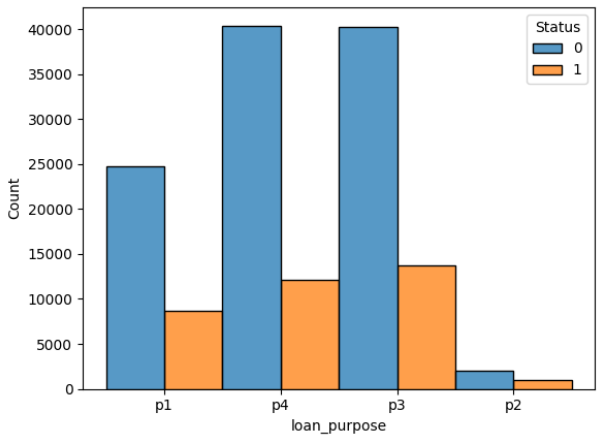
Slika 3: Zavisnost statusa od numeričkih atributa

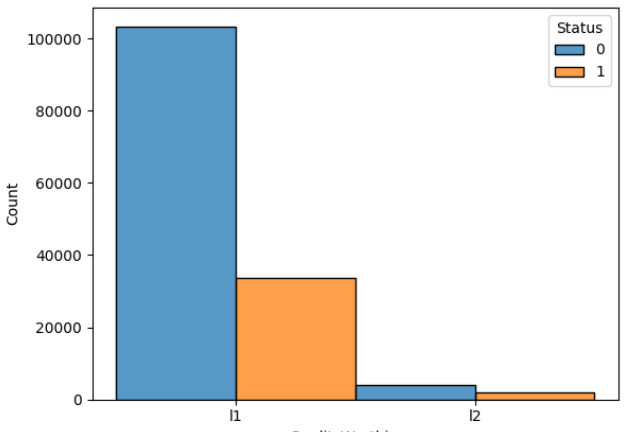


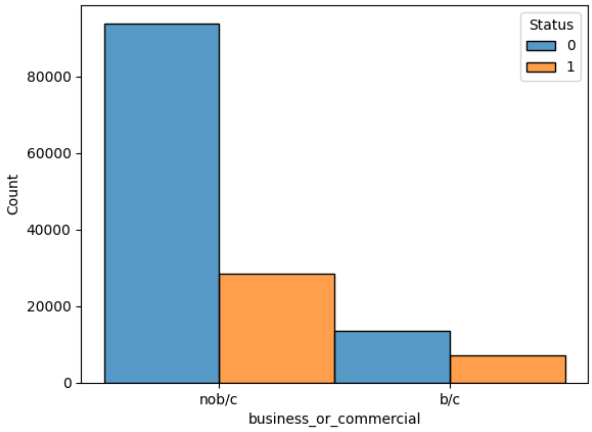


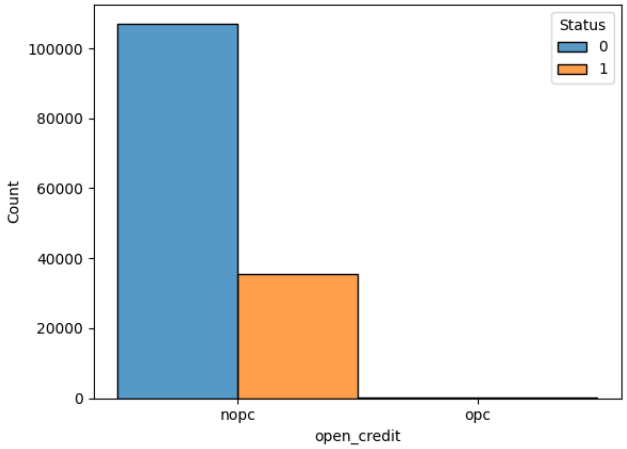


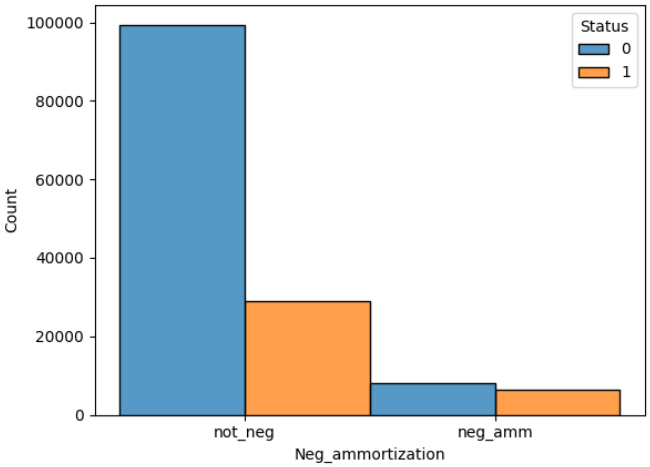


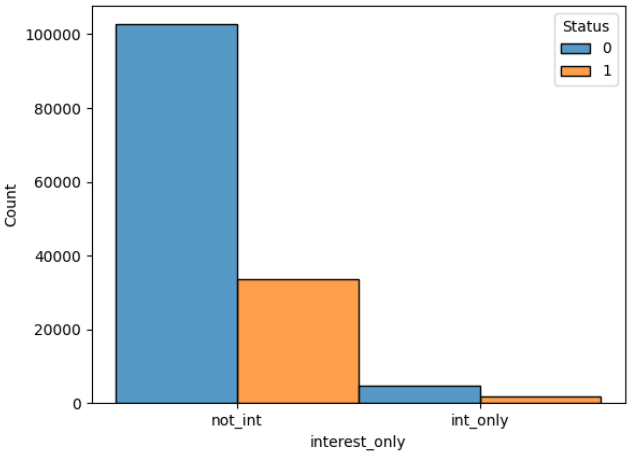

Status
0
1
Count
40000
35000
30000
25000
20000
15000
10000
5000
0
p1
p4
p3
p2
loan_purpose

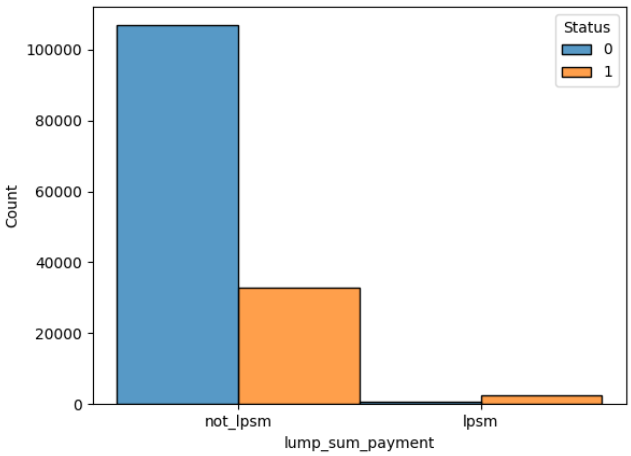

Status
0
1
Count
100000
80000
60000
40000
20000
0
l1
l2

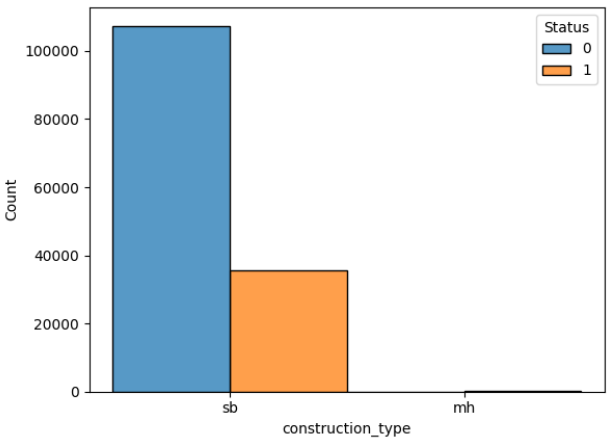

Status
0
1
Count
80000
60000
40000
20000
0
nob/c
b/c
business_or_commercial

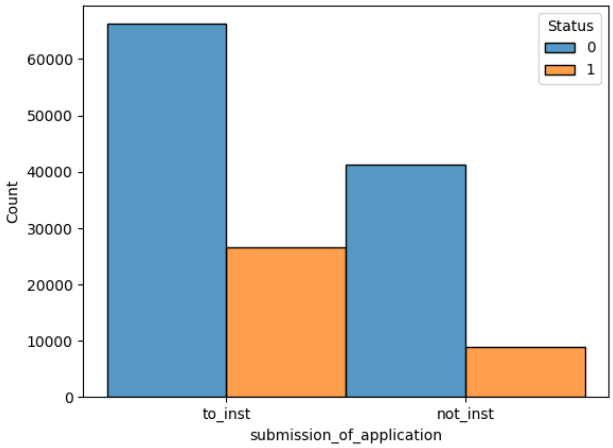

Status
0
1
Count
100000
80000
60000
40000
20000
0
nopc
opc
open_credit

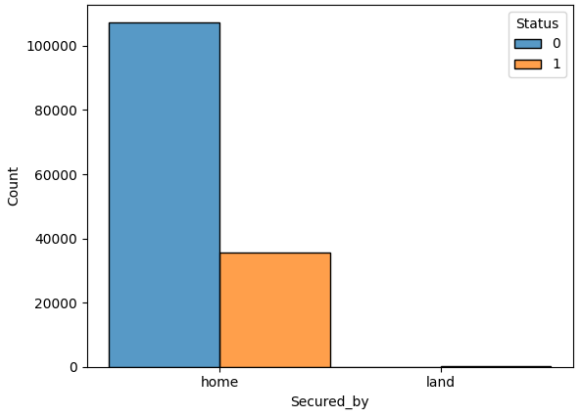

Status
0
1
Count
100000
80000
60000
40000
20000
0
not_neg
neg_amm
Neg_ammortization

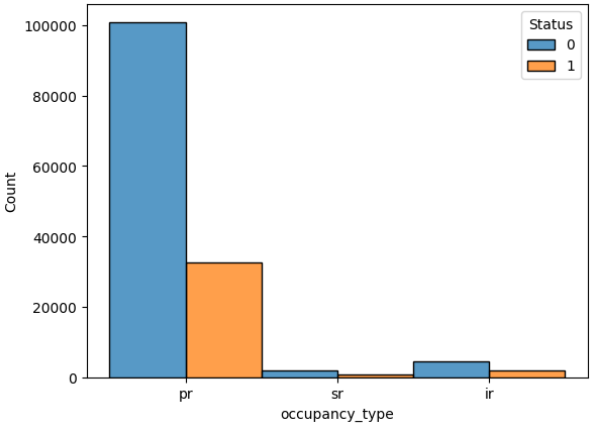

Status
0
1
Count
100000
80000
60000
40000
20000
0
not_int
int_only
interest_only

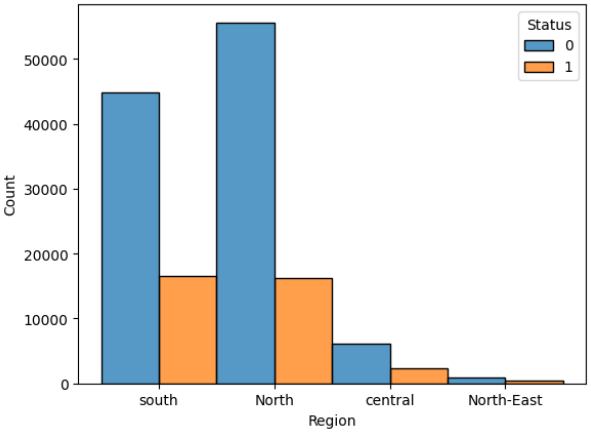

Status
0
1
Count
100000
80000
60000
40000
20000
0
not_lpsm
lpsm
lump_sum_payment

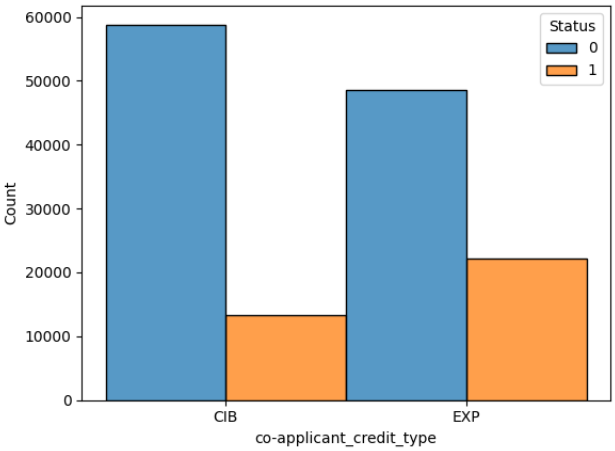

Status
0
1
Count
100000
80000
60000
40000
20000
0
sb
mh
construction_type

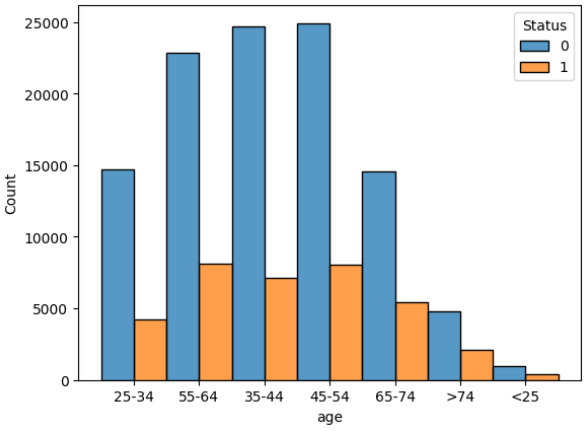

Status
0
1
Count
60000
50000
40000
30000
20000
10000
0
to_inst
not_inst
submission_of_application

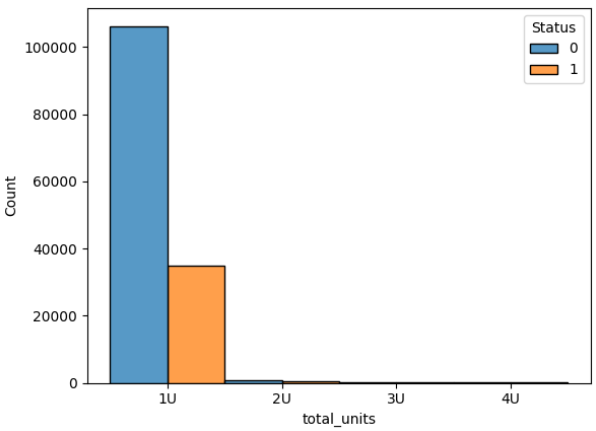

Status
0
1
Count
100000
80000
60000
40000
20000
0
home
land
Secured_by

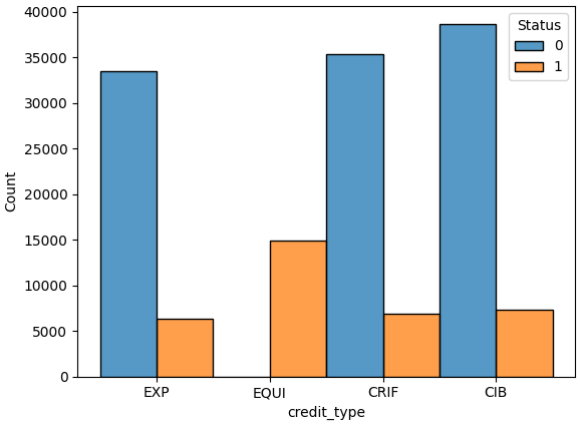

Status
0
1
Count
100000
80000
60000
40000
20000
0
pr
sr
ir
occupancy_type

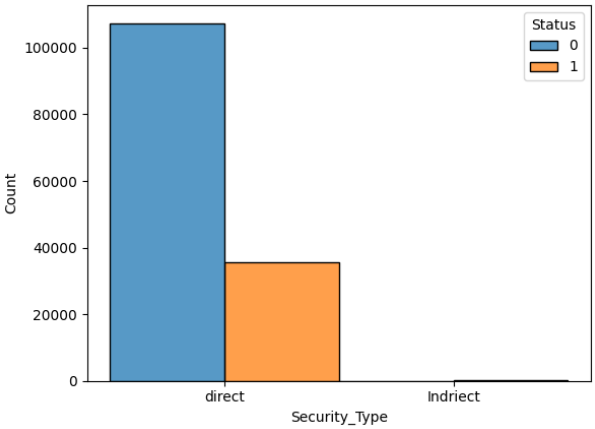

Status
0
1
Count
50000
40000
30000
20000
10000
0
south
North
central
North-East
Region

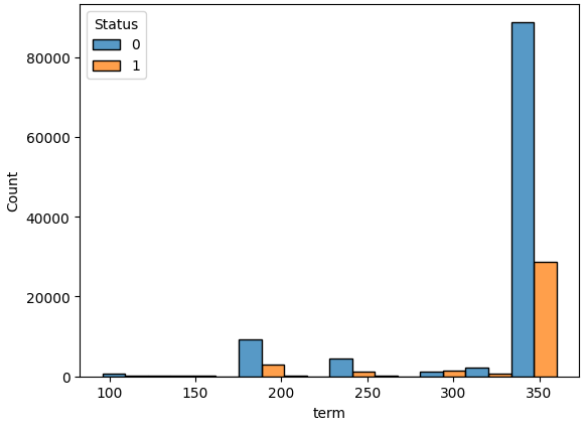

Status
0
1
Count
60000
50000
40000
30000
20000
10000
0
CIB
EXP
co-applicant_credit_type


Status
0
1
Count
25000
20000
15000
10000
5000
0
25-34
55-64
35-44
45-54
65-74
>74
<25
age


Status
0
1
Count
100000
80000
60000
40000
20000
0
1U
2U
3U
4U
total_units


Status
0
1
Count
40000
35000
30000
25000
20000
15000
10000
5000
0
EXP
EQUI
CRIF
CIB
credit_type





Slika 4: Zavisnost statusa od kategoričkih atributa

Pretprocesiranje

Prvi korak pri pretprocesiranju je da rešimo problem nedostajućih vrednosti. Prvo možemo da nađemo broj i procenat nedostajućih vrednosti po koloni. Nakon toga tražimo korelaciju između nedostajućih polja sa slicnim procentima. Dolazimo do zaključka da za atribute LTV i property_value važi da ako jednog nema, nema ni drugog, a veoma jako povezanost u nedostatku podataka primećujemo u poljima Interest_rate_spread, Upfront_charges i rate_of_interest, gde u svim redovima u kojima nedostaje poslednji, nedostaju i druga dva, koji fale u malo(<3%) većem broju redova. Kolona Gender koja ima sličan procenat nedostajućih vrednosti ne pokazuje jaku korelaciju u nedostajanju sa ova 3 atributa.

```
loan_limit : 3344 (2.249%)
Gender : 37659 (25.331%)
approv_in_adv : 908 (0.611%)
loan_purpose : 134 (0.09%)
rate_of_interest : 36439 (24.51%)
Interest_rate_spread : 36639 (24.645%)
Upfront_charges : 39642 (26.664%)
term : 41 (0.028%)
Neg_ammortization : 121 (0.081%)
property_value : 15098 (10.155%)
income : 9150 (6.155%)
age : 200 (0.135%)
submission_of_application : 200 (0.135%)
LTV : 15098 (10.155%)
dtir1 : 24121 (16.225%)
```

Slika 5: Broj i procenat nedostajućih vrednosti

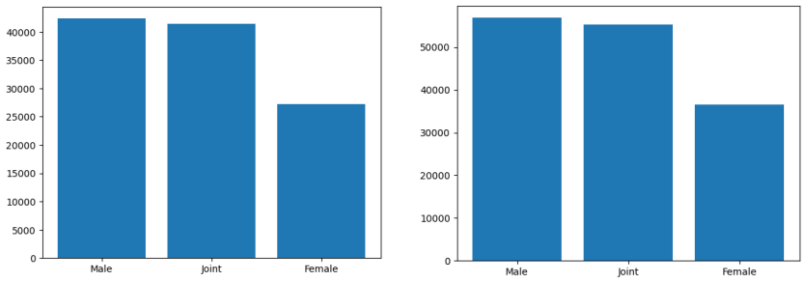
```
15098
36439
```

Slika 6: Broj nedostajućih vrednosti zajedničkih za LTV i property_value, odnosno rate_of_interest, Interest_rate_spread i Upfront charges

```
10567
```

, Slika 7: Broj nedostajućih vrednosti Izajedničkih zaGender i gorepomenuta tri atributa

Gender je kateogrička promenljiva, pa nedostajuće vrednosti možemo popuniti koristeći bfill ili ffill. Iscrtavanjem grafa sa brojem vrednosti po kolonama pre i posle popunjavanja grubo proverimo da li je odnos ocuvan.



Slika 8: Broj vrenosti po kolonama pre i posle popunjavanja nedostajućih vrednosti za atribut Gender

Za ostale kategoričke atribute možemo koristiti metodu ffill i bfill zajedno za popunjavanje nedostajućih vrednosti, i ne moramo da proveravamo odnos nakon popunjavanja jer je njihov procenat nedostajućih vrednosti mali, pa bilo kakvo popunjavanje neće pokvariti odnos.

Za numeričke podatke koristimo KNNImputer. On radi tako što za red r1 koji ima nedostajuće vrednosti nađe k redova koji su mu najsličniji po kolonama koje nemaju nedostajuće vrednosti, a u njima ima nešto upisano u kolonama koje su prazne u redu r1, a zatim u prazna polja r1 upiše interpoliranu vrednost atributa iz tih k redova.

Nakon ovoga, izbacimo outlier-e.

```
5757  redova izbaceno
4.0 % originalnog dataset-a
```

Slika 9: Broj i procenat izbašenih redova

Sada vršimo kodiranje kategoričkih atributa, i to dummy kodiranje.

Nakon toga, spremni smo da za klasifikaciju podelimo podatke u train I test skup u odnosu 70/30, dok za klasterovanje skaliramo podatke koristecći StandardScaler.

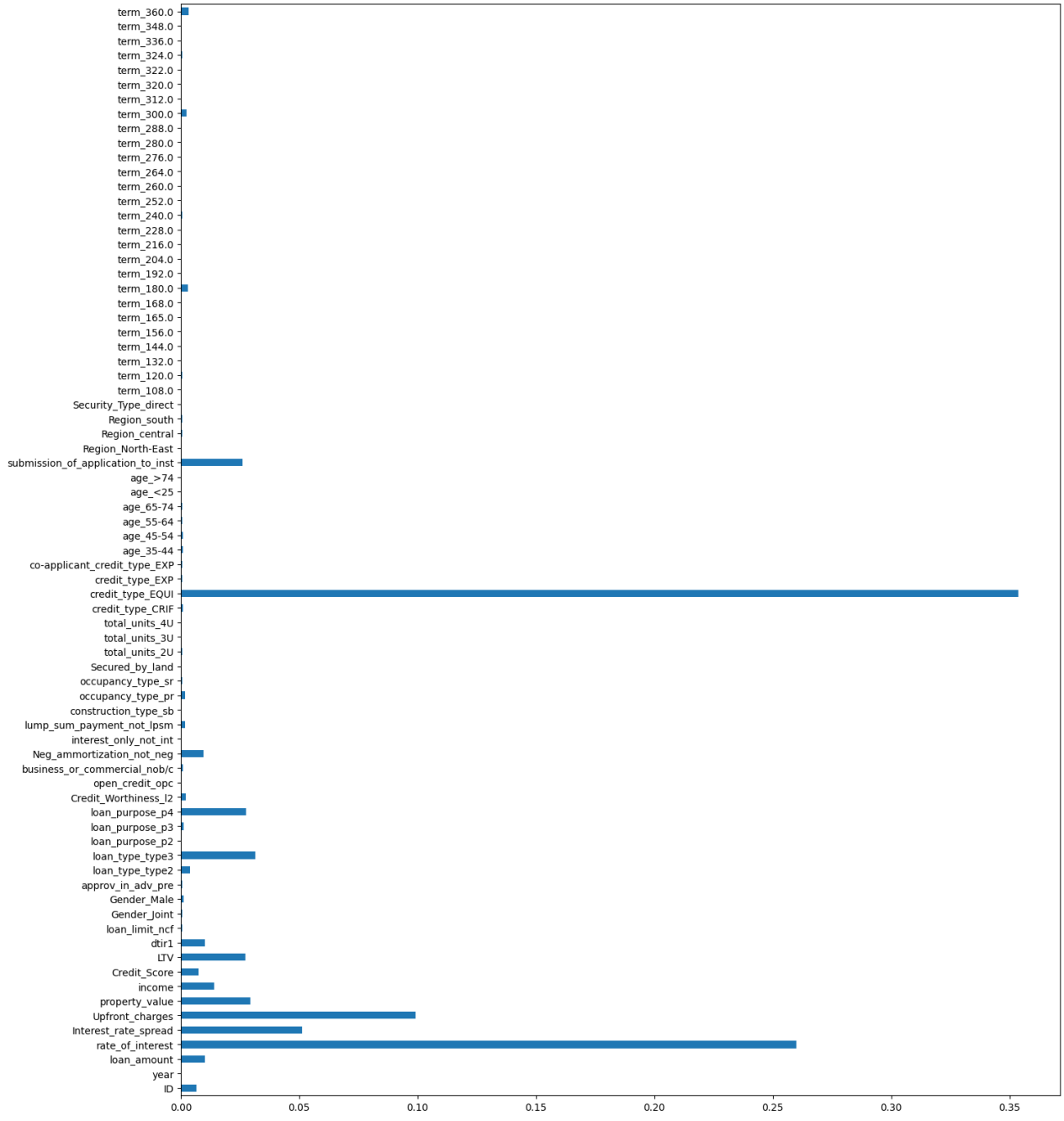
Klasifikacija

Prvi algoritam koji koristimo je Decision Tree. Iako se model prepirlagodjava nad trening podacima, i kada se ne podese hiper parametri, a i kada se podese koristeći GridSearchCV, daje odlične rezultate nad trening skupom. Sa ~96% tačnošću i ~92 f1 merom kada se ne podese hiper parametri i ~97% tačnošću i ~94% f1 merom kada se podese, ovo će biti naš najbolji model.

```
Train:
Confusion matrix:
 [[75208     0]
 [    0 24831]]
Accuracy:  1.0
Precision:  1.0
Recall:  1.0
F1:  1.0
Test:
Confusion matrix:
 [[31384   848]
 [  760  9882]]
Accuracy:  0.9624947520641881
Precision:  0.9209692451071761
Recall:  0.9285848524713399
F1:  0.9247613700168444
```

```
Train:
Confusion matrix:
 [[75208     0]
 [    0 24831]]
Accuracy:  1.0
Precision:  1.0
Recall:  1.0
F1:  1.0
Test:
Confusion matrix:
 [[31554   678]
 [  692  9950]]
Accuracy:  0.9680459019452349
Precision:  0.936206247647723
Recall:  0.9349746288291675
F1:  0.9355900329102022
```

Slika 10: Model DecisionTree – mere



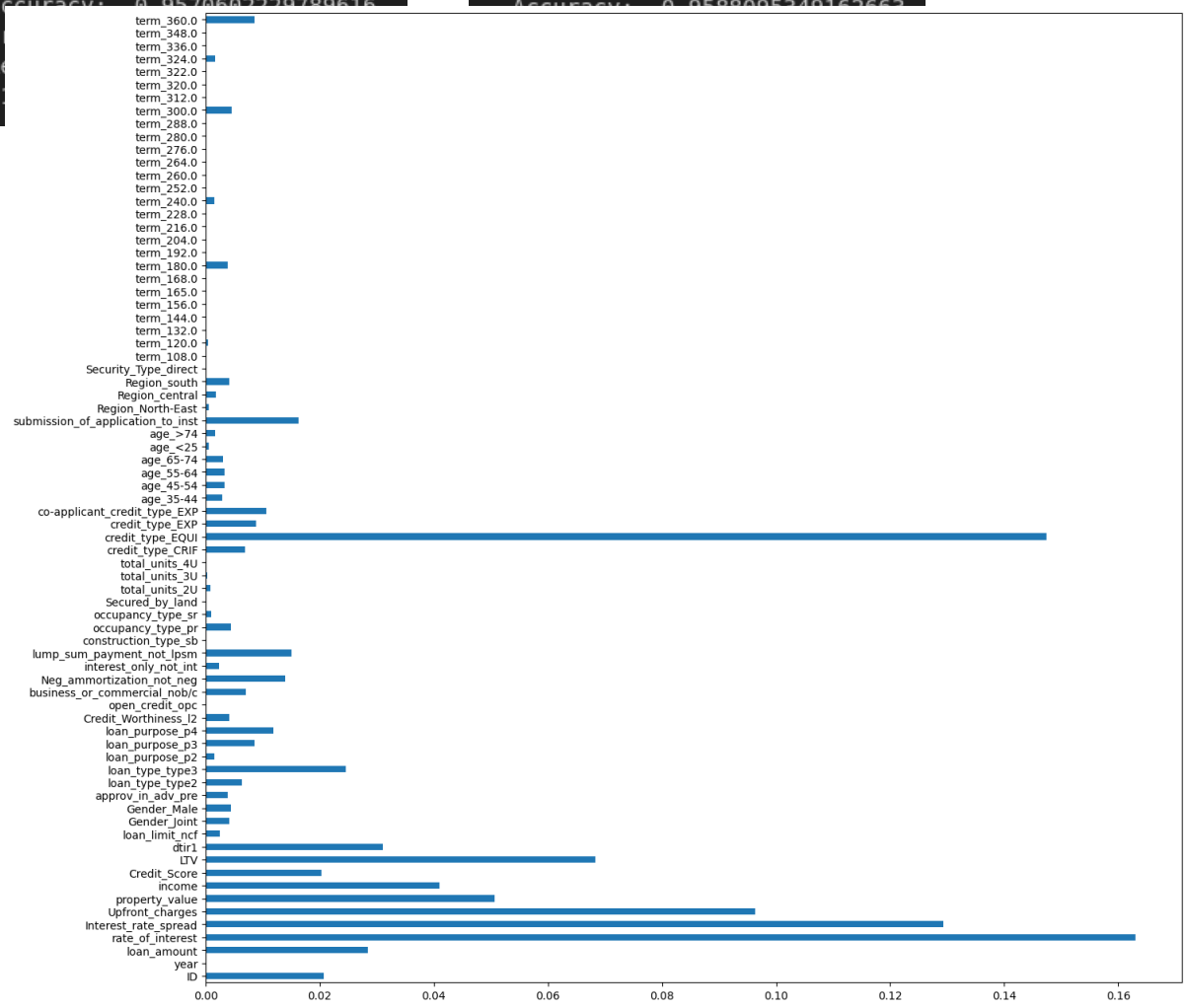
Slika 11: Graf feature_importance-a za DecisionTree model

RandomForrest algoritam nam iznenađujuće daje gore rezultate u svemu osim u preciznošću od običnog DecisionTree algoritma.

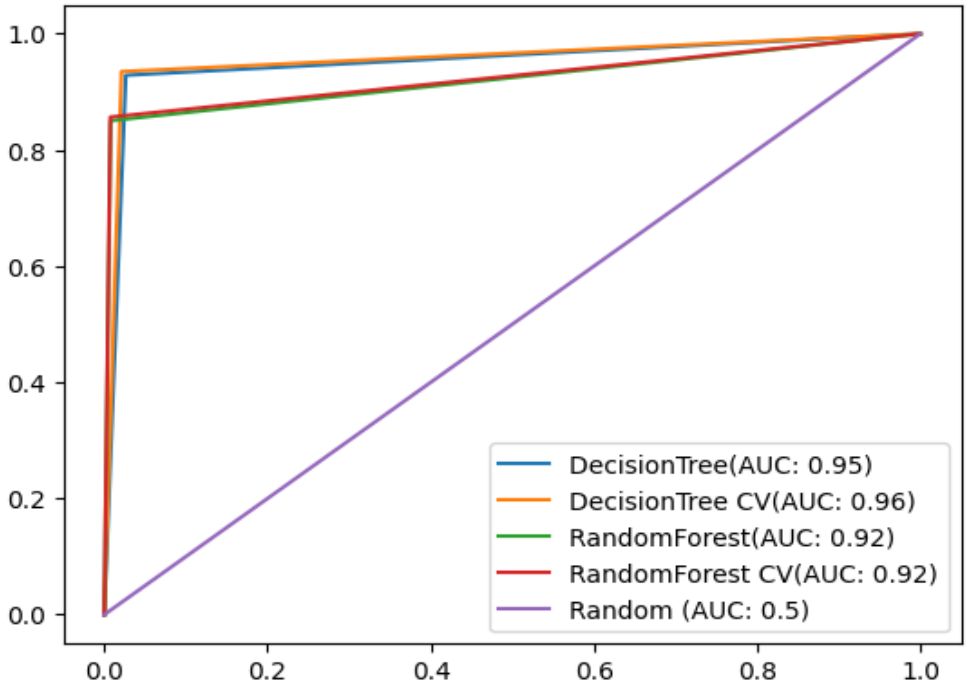
```
Train:
Confusion matrix:
 [[75208     0]
 [    0 24831]]
Accuracy:  1.0
Precision:  1.0
Recall:  1.0
F1:  1.0
Test:
Confusion matrix:
 [[31983   249]
 [ 1592  9050]]
```

```
Train:
Confusion matrix:
 [[75208     0]
 [    0 24831]]
Accuracy:  1.0
Precision:  1.0
Recall:  1.0
F1:  1.0
Test:
Confusion matrix:
 [[31992   240]
 [ 1526  9116]]
```

Slika 12: Model Random Forrest – mere



Slika 13: Graf feature_importance-a za RandomForrest

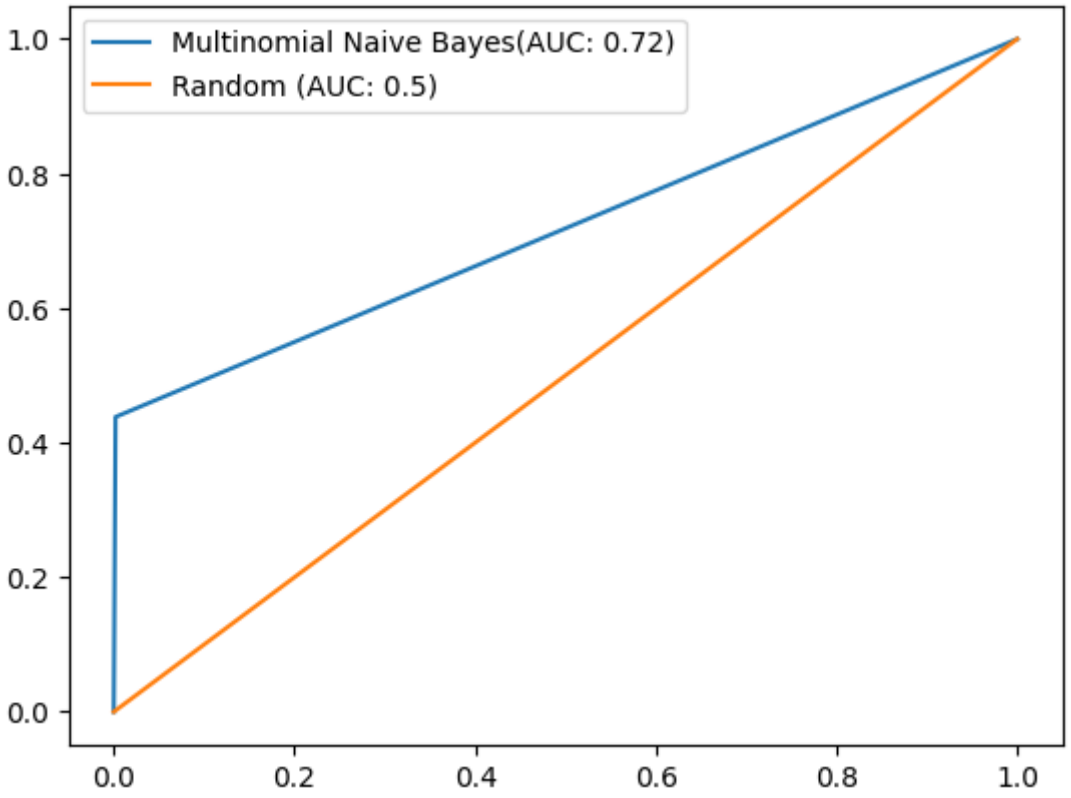


Slika 14: ROC krive i AUC za DecisionTree i RandomForrest

Multinomijalni naivni Bajesova algoritam nam ocekivano ne daje dobre rezultate, s obzirom na prirodu skupa podataka.

```
Train:
Confusion matrix:
 [[75066   142]
 [13944 10887]]
Accuracy:  0.8591949139835464
Precision:  0.987124852661166
Recall:  0.4384438806330796
F1:  0.6071946458449526
Test:
Confusion matrix:
 [[32169    63]
 [ 5978  4664]]
Accuracy:  0.8590987544899006
Precision:  0.9866723080177703
Recall:  0.438263484307461
F1:  0.6069360400806818
```

Slika 15: Model MultinomialNB – mere



Slika 16: ROC kriva i AUC za MultinomialNB

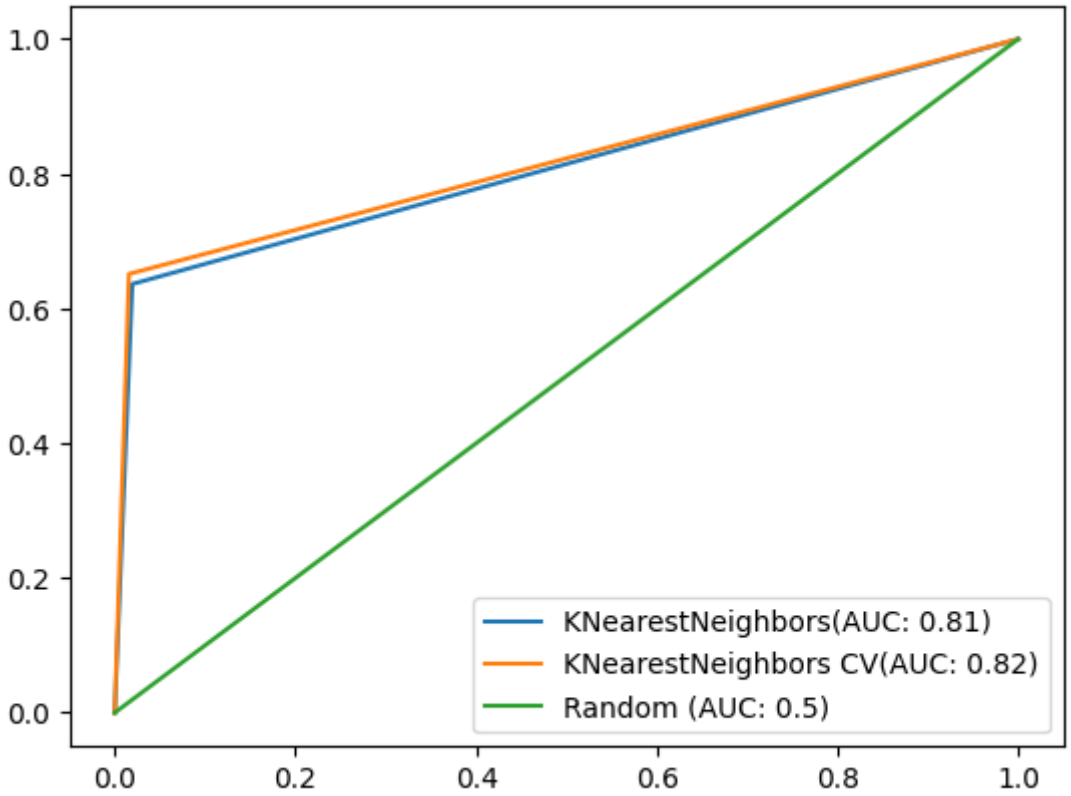
KNN algoritam nam daje dobru tačnost i preciznost međutim odziv je loš. Kako bi zadržali dobro, a poboljšali loše, kao za scoring hiperparametar postavljamo f1_scorer, međutim ne vidimo preterano veliko poboljšanje

```
Train:
Confusion matrix:
 [[74414   794]
 [ 6961 17870]]
Accuracy score:  0.9224802327092434
Precision score:  0.9574582083154737
Recall score:  0.7196649349603318
F1 score:  0.8217036440970226
Test:
Confusion matrix:
 [[31607   625]
 [ 3870  6772]]
Accuracy score:  0.8951579045575407
Precision score:  0.9155062863322969
Recall score:  0.6363465514001128
F1 score:  0.7508176728200011
```

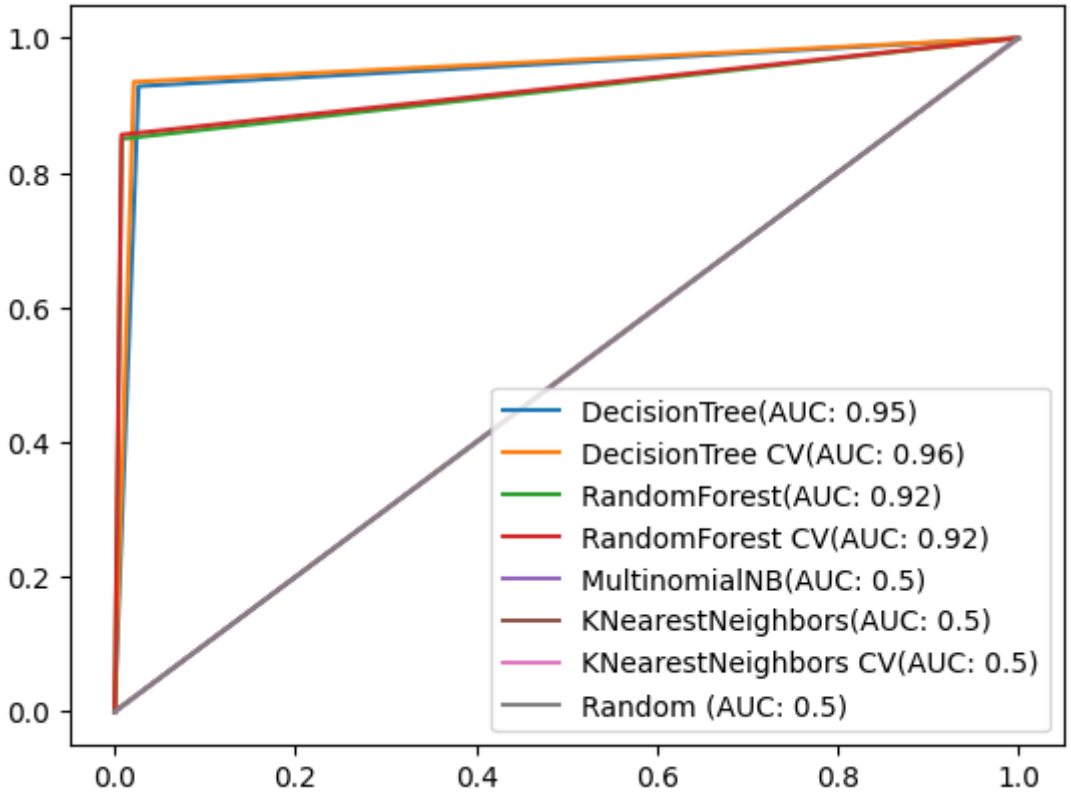
```
Train:
Confusion matrix:
 [[75208     0]
 [    0 24831]]
Accuracy score:  1.0
Precision score:  1.0
Recall score:  1.0
F1 score:  1.0
Test:
Confusion matrix:
 [[31727   505]
 [ 3711  6931]]
Accuracy score:  0.901665344964314
Precision score:  0.9320871436256052
Recall score:  0.6512873520015035
F1 score:  0.7667883615444187
```

Slika 17: Model KNN – mere

Slika 18: ROC kriva i AUC za KNN



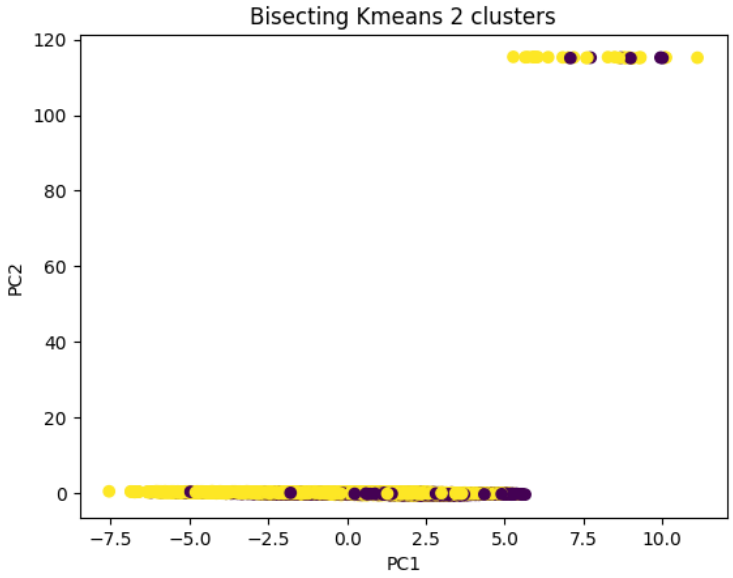
Poređenje modela



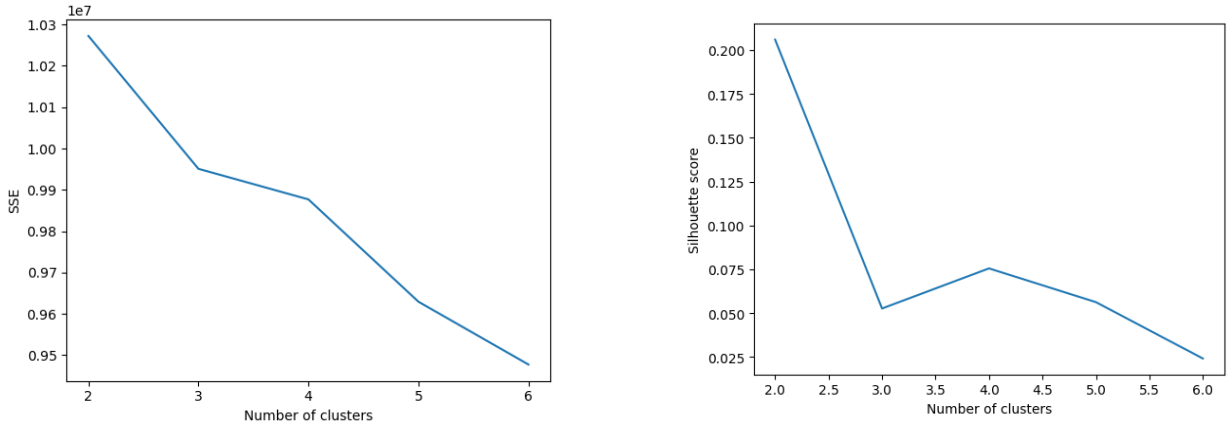
Slika 19: Poređenje modela klasifikacije

Klasterovanje

Kako bi mogli da vizualizujemo klasterovanje prvo ćemo primeniti PCA nad celim X skupom svesti skup na 2 atributa. Prvi algoritam za klasterovanje koji primenjujemo je Bisecting Kmeans, čije centre ćemo kasnije iskoristiti za običan Kmeans algoritam. Međutim imamo problem, jer nam grafici izgledaju ovako (Silhoutte score i SSE nam ne idu u korist takođe):

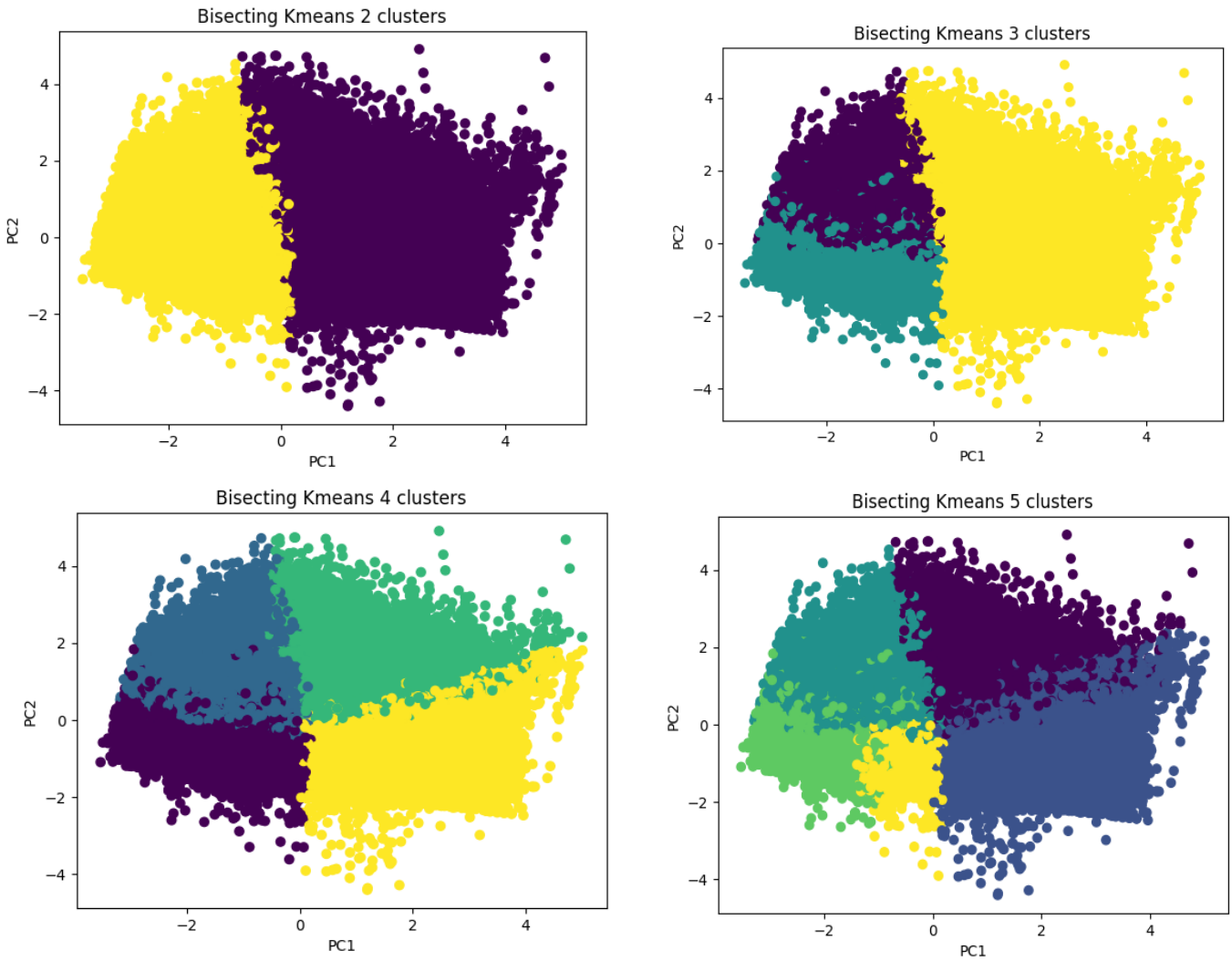


Slika 20: Bisecting Kmeans greška



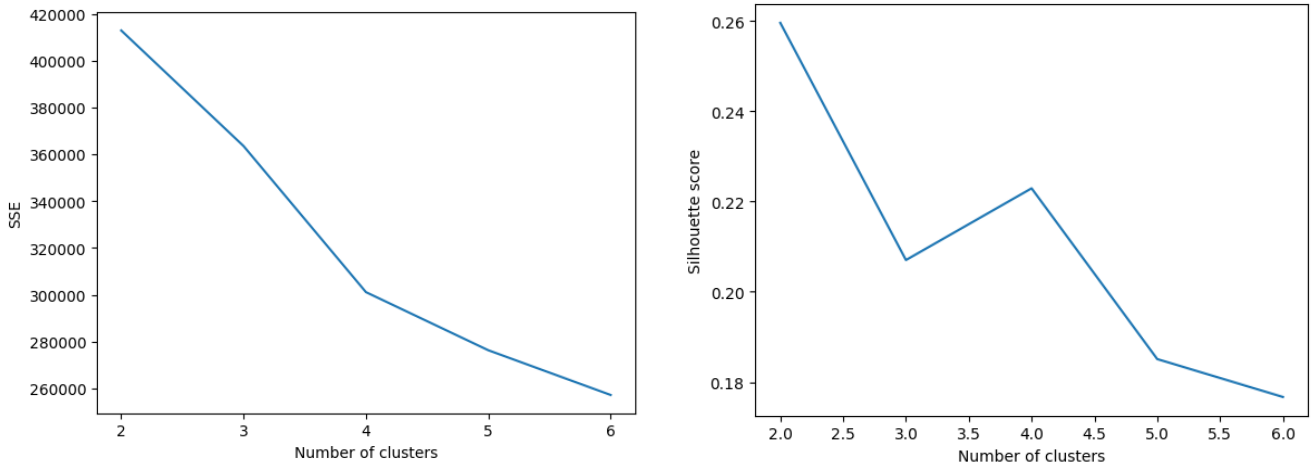
Slika 21: Bisecting Kmeans silhoutte score i SSE – greška

Kako bi ovo ispravili PCA primenjujemo samo na 4 kolone iz X skupa, i to 4 numeričke koje su imale najveći feature_importance u RandomForrest algoritmu.



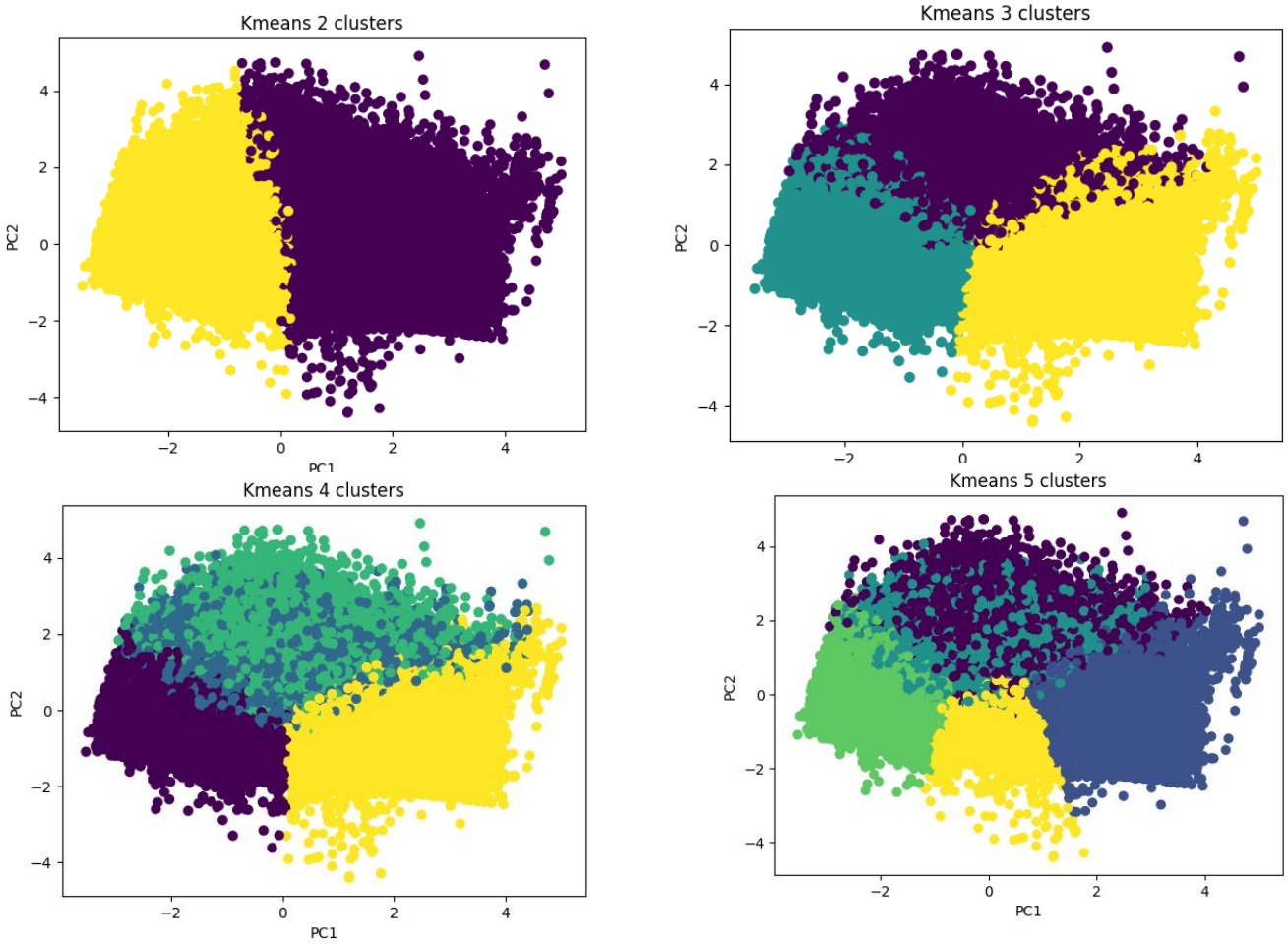
Slika 22: Bisecting Kmeans

Sillhoute score i SSE nam sugerišu da je najbolje podeliti na 4 klastera, što je suprotno od onoga što mi znamo. Probajmo sad Kmeans.

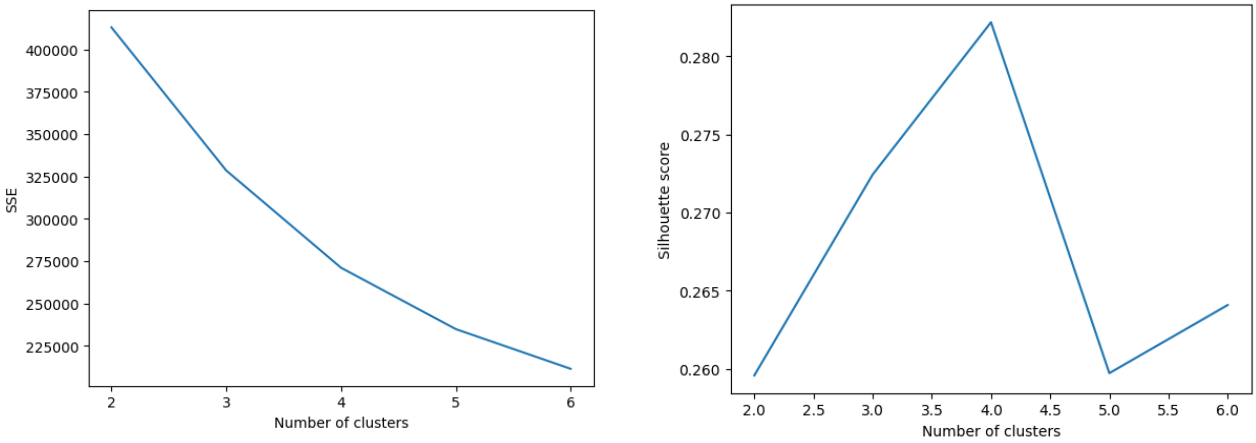


Slika 23: Bisecting Kmeans – Silhoutte score i SSE

Kmeans algoritam nam sugerise isto, još jasnije.



Slika 24: Kmeans



Slika 25: Kmeans SSE i Silhoutte score

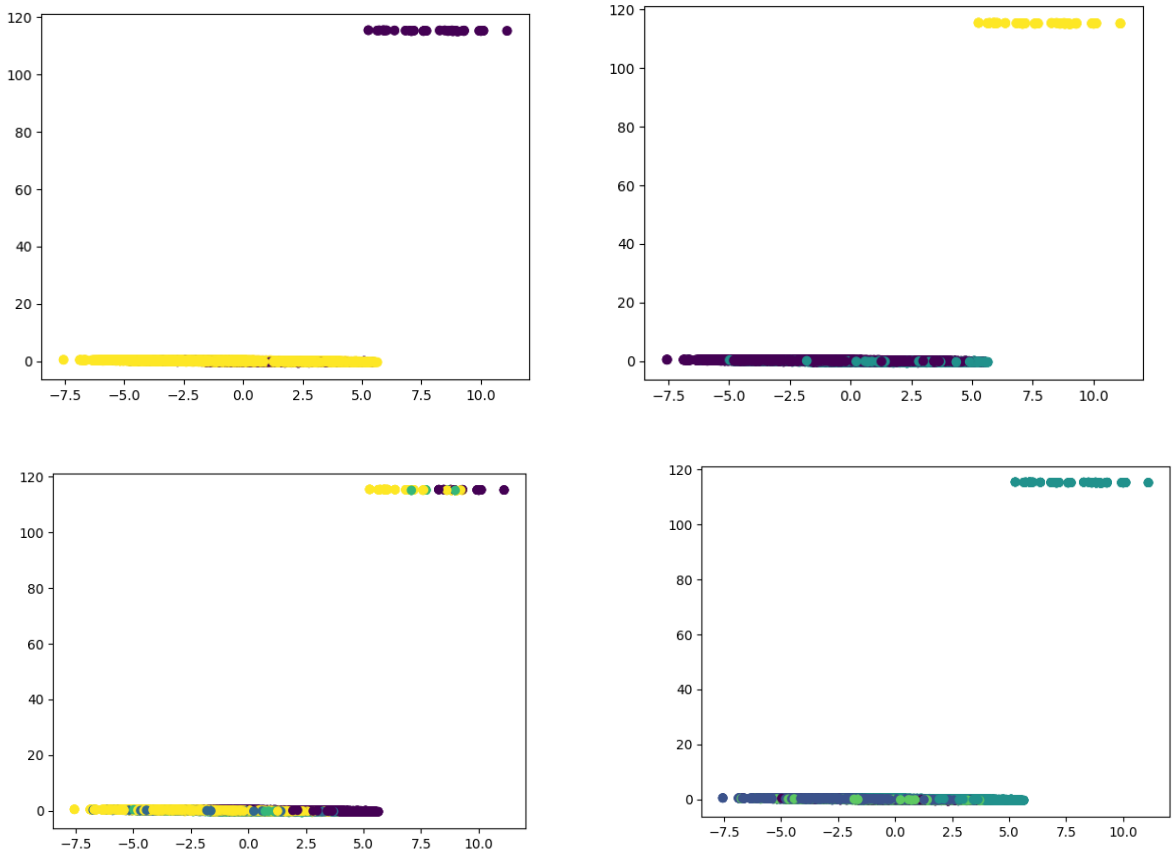
Dalje, koristimo Gaussian Mixture model. On koristi Gausovu raspodelu i parametar Pi za odredjivanje verovatnoce da odredjena tacka pripada odredjenom klasteru. Gausova raspodela je

parametrizovana prosekom i matricom kovarijanse za dati broj klastera, pa se zavreme treniranja ti parametri, kao I Pi parametar optimizuju.

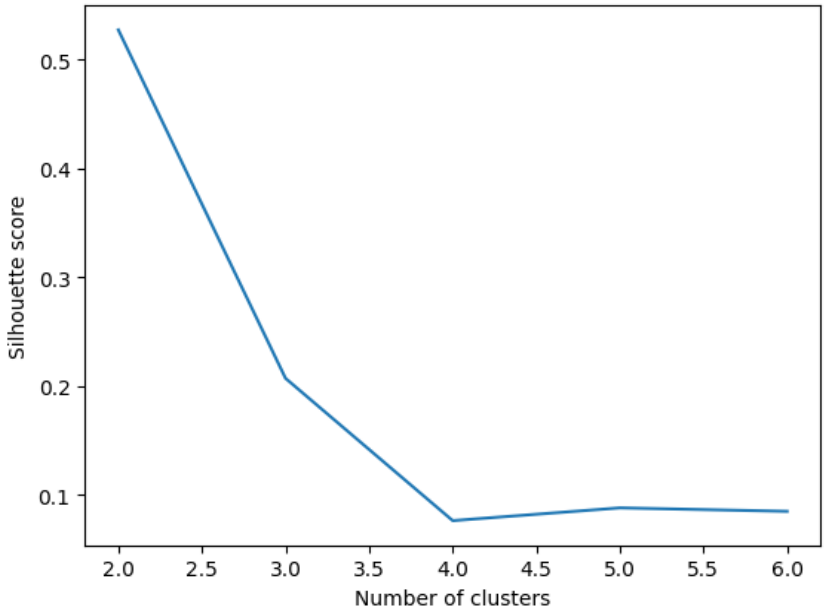
$$G(X|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}(X-\mu)^T \mathbf{\Sigma}^{-1}(X-\mu)\right)$$

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N} \gamma_k(x_n) \qquad \gamma_k(X) = p(k|X).$$

Slika 26: Formule za Gausovu distribuciju i Pi parametar



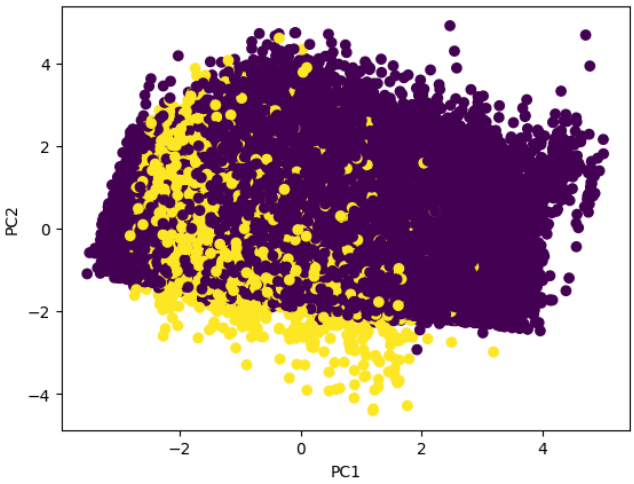
Slika 27: Gausian Mixture

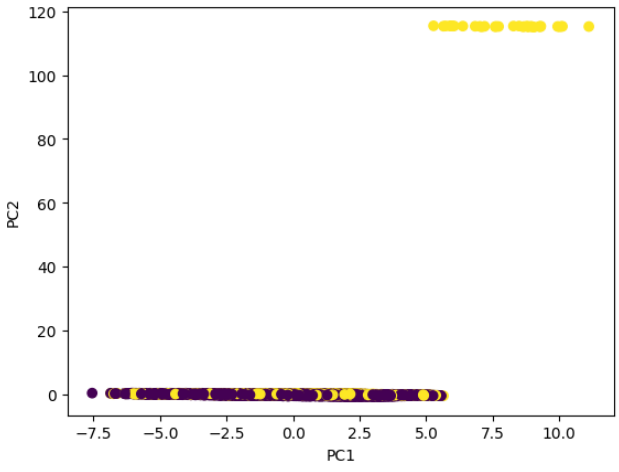


Slika 28: Silhoutte score za Gausian Mixture

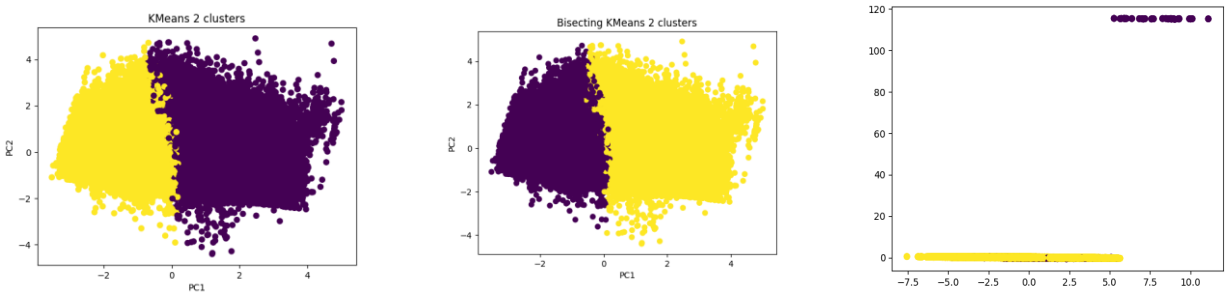
Poređenje modela

Kako klaster stvarnih podataka ne liči na naše klastere 2, ne možemo da očekujemo mnogo dobre rezultate. Dobijamo tačnost od ~65%, odnosno ~73%, što nam ne znači mnogo s obzirom da je jedna klasa otprilike duplo veća od druge. Očekivano je da klasterovanje ne daje dobre rezultate jer je u pitanju skup za klasifikaciju





Slika 29: Stvarno podaci - klasteri



Slika 30: Pogadjanje kod klasterovanja

Pravila pridruživanja

Od pravila pridruživanja ne možemo mnogo očekivati, s obzirom da nam značenje kolona nije objašnjeno, a pritom je u pitanju skup za klasifikaciju. Najbolje što možemo dobiti je šta to sve povlači zajednički kredit. Koristli smo Apriori algoritam.

Sort by: Lift 312511 of 312511

Consequent	Antecedent	Support %	Confidence %	Lift
Gender = Joint	submission_of_a... co-applicant_credi... Status = 0 approv_in_adv = n... business_or_com...	11,7	87,174	3,131
Gender = Joint	submission_of_a... co-applicant_credi... Status = 0 approv_in_adv = n... total_units = 1U	12,734	87,122	3,129
Gender = Joint	submission_of_a... co-applicant_credi... Status = 0 business_or_com... total_units = 1U	13,062	87,106	3,128
Gender = Joint	submission_of_a... co-applicant_credi... Status = 0 approv_in_adv = n... Neg_ammortizatio...	12,084	87,059	3,126
Gender = Joint	submission_of_a... co-applicant_credi... Status = 0 business_or_com... Neg_ammortizatio...	12,49	87,011	3,125

Slika 31: Pravila pridruživanja

Zaključak

Predviđanje da li će neko moći da otplati kredit ne može biti perfektno, međutim možemo napraviti veoma dobro predviđanje. Ljudska priroda, koliko god nepredvidiva, otplaćivanje kerdita je prioritet gotovo svima, što nama omogućava pravljenje dobrog modela.