

Samostalni seminarski rad u okviru kursa :
Istrazivanja Podataka 1

Gubitak klijenata banke

Marko Nikitović

mi20123@alas.matf.bg.ac.rs

Matematički fakultet, Univerzitet u Beogradu

1. Uvod

Analiza podataka o napuštanju pomaže bankama da bolje razumeju razloge za odlazak klijenata, što im omogućava da preduzmu efikasne korake za zadržavanje klijenata. Ovo može uključivati razvoj personalizovanih programa lojalnosti, prilagodbe ponuda i usluga, kao i unapređenje korisničke podrške. Kao što znamo, mnogo je skuplje privući novog klijenta nego zadržati postojećeg.

Za banke je od koristi znati šta navodi klijenta na odluku da napusti kompaniju. Upravljanje odlivom klijenata postaje ključna strategija za banke u savremenom bankarskom sektoru, jer im omogućava da ostanu konkurentne i ostvare dugoročni uspeh. Održavanje postojećih klijenata i smanjenje njihovog odliva može značajno doprineti finansijskoj stabilnosti i rastu banke, a analiza podataka igra ključnu ulogu u ostvarivanju ovih ciljeva.

Mi ćemo se ovim problem baviti na skupu podataka “Bank Customer Churn” preuzetog sa sajta *Kaggle*.

Na ovom skupu prikazaćemo nekoliko metoda za klasifikaciju i klasterovanje, a na kraju ćemo se pozabaviti i sa pravilima pridruživanja.

1.1. Eksplorativna analiza podataka

Prvo je potrebno upoznati se sa skupom podataka, skup sadrži 10000 instanci i inicijalno 18 atributa.

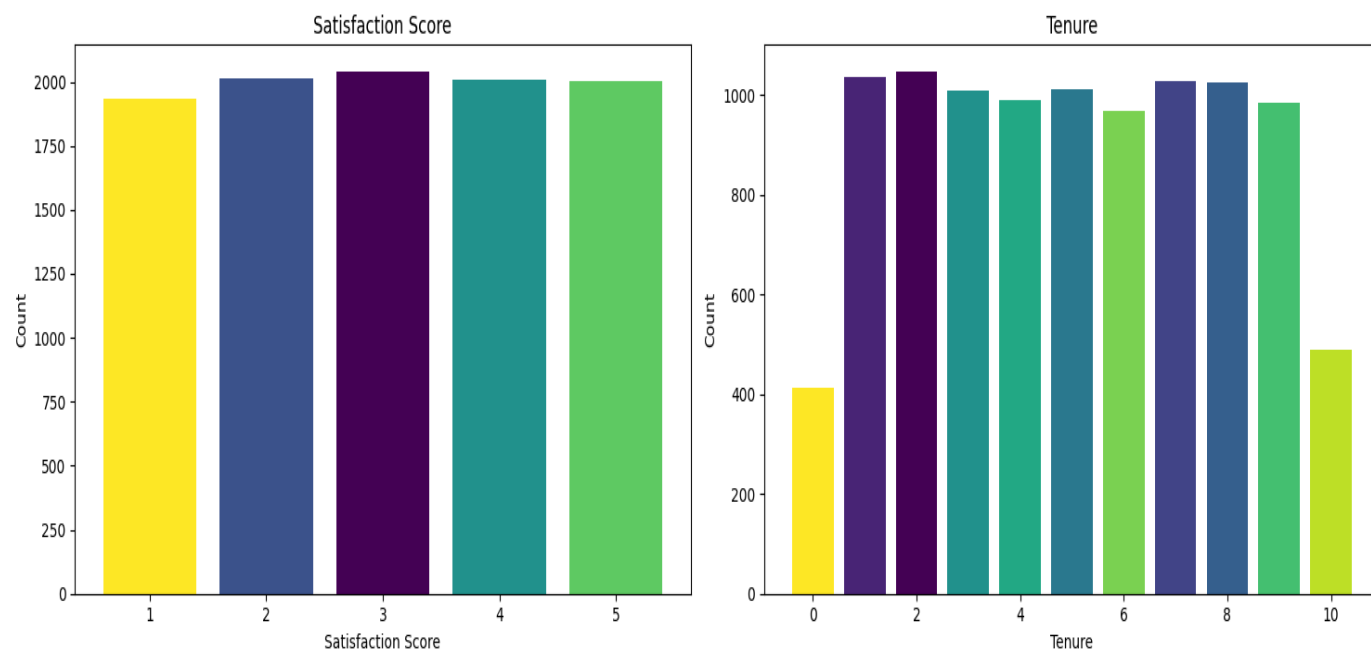
Svojstvo **Exited** predstavlja našu ciljnu promenljivu. Spisak atributa, njihovo značenje i tip :

- **RowNumber** – numerički
- **CustomerId** – identifikator klijenta, numerički atribut
- **Surname** – prezime klijenta, kategorički atribut
- **CreditScore** – je numerička ocena ili broj koji se koristi kako bi se procenila kreditna sposobnost ili kreditna istorija jednog pojedinca ili entiteta, numerički atribut
- **Geography** – država prebivališta klijenta, kategorički
- **Gender** – pol, binarni kategorički
- **Age** – godine, numerički atribut
- **Tenure** – referiše na broj godina koliko je klijent član banke, kategorički (ordinalni)
- **Balance** – stanje na računu klijenta, numerički
- **NumOfProducts** – predstavlja informaciju o broju finansijskih proizvoda ili usluga koje je klijent akvizirao ili trenutno koristi od strane banke, kategorički (ordinalni)
- **HasCrCard** – indikator da li klijent poseduje kreditnu karticu u ovoj banci, binarni
- **IsActiveMember** – da li je klijent trenutno aktivan i koristi banku za svoje finansijske potrebe, binarni
- **EstimatedSalary** – označava procenjeni ili očekivani prihod ili godišnju platu osobe koja se analizira, numerički

- **Exited** – ciljna promenljiva, indikator da li je klijent napustio banku, binarni
- **Complain** – da li je klijent imao žalbe ili ne, binarni
- **Satisfaction Score** – procena koliko je korisnik zadovoljan uslugama banke, numerički
- **Card Type** – tip kartice koju klijent poseduje, kategorički (ordinalni)
- **Points Earned** – broj poena koji je skupljen korišćenjem kreditne kartice, numerički

Atributi RowNumber, CustomerId, Surname nisu relevantni za dalju analizu, zato ih izbacujemo iz dalje obrade.

Figure 1.1.1 : Grafikoni apsolutnih frekvencija za redne (ordinalne) attribute



Na narednim slikama ce biti prikazano raspodele atributa po klasama, prvo za kategoricke attribute a zatim za numericke. Ovo uporedjivanje na moze dati znacajne informacije o vaznosti atributa prilikom klasifikacije podataka.

Distribution of categorical variables based on the value of the target variable

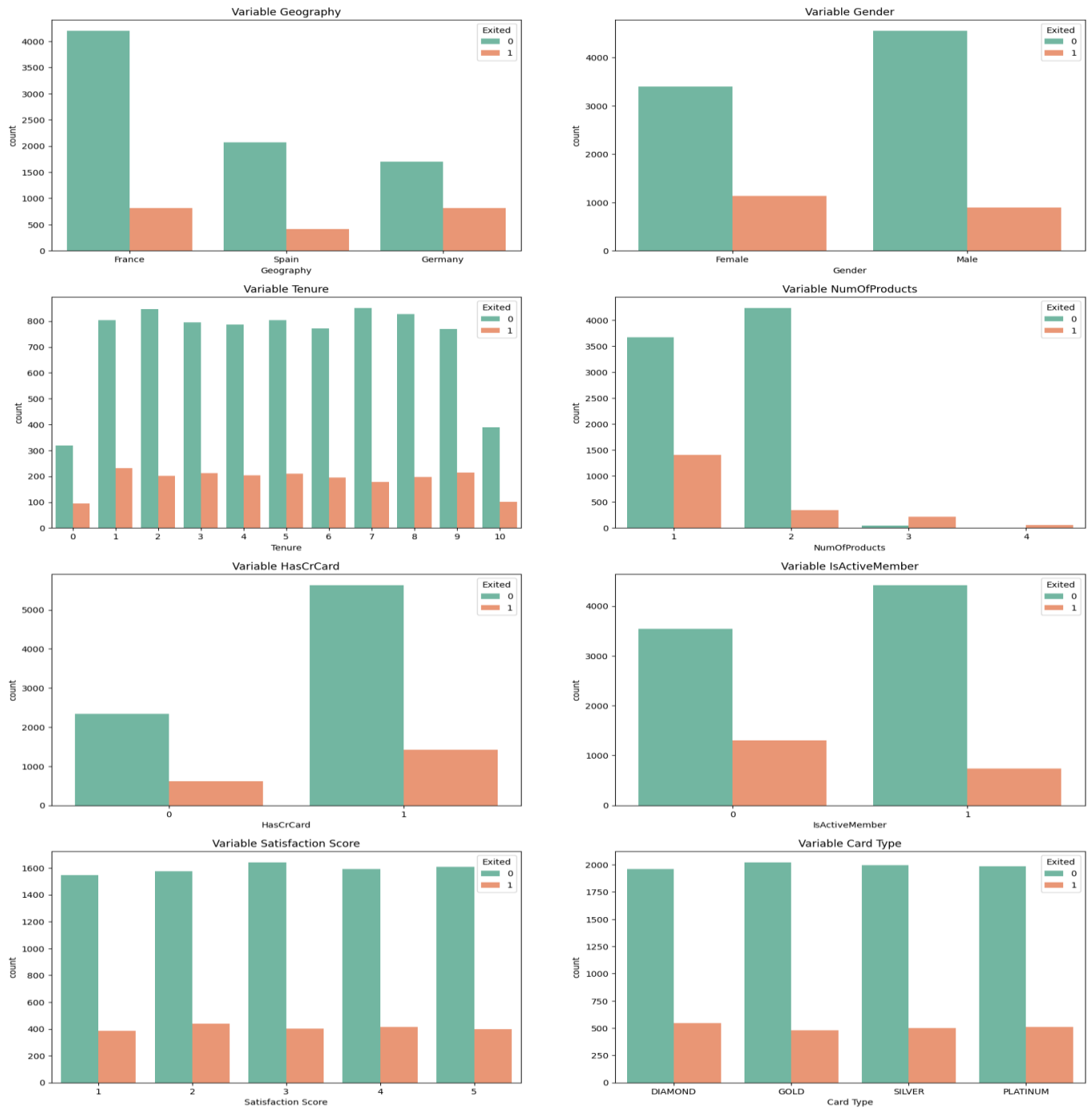


Figure 1.1.2: Raspodele kategorickih atributa po vrednosti ciljne promenljive

Za kategoricke attribute priblizno su podjednako raspodeljeni po ciljnoj promenljivoj osim atributa *NumOfProducts*, gde je razlika baš izražena.

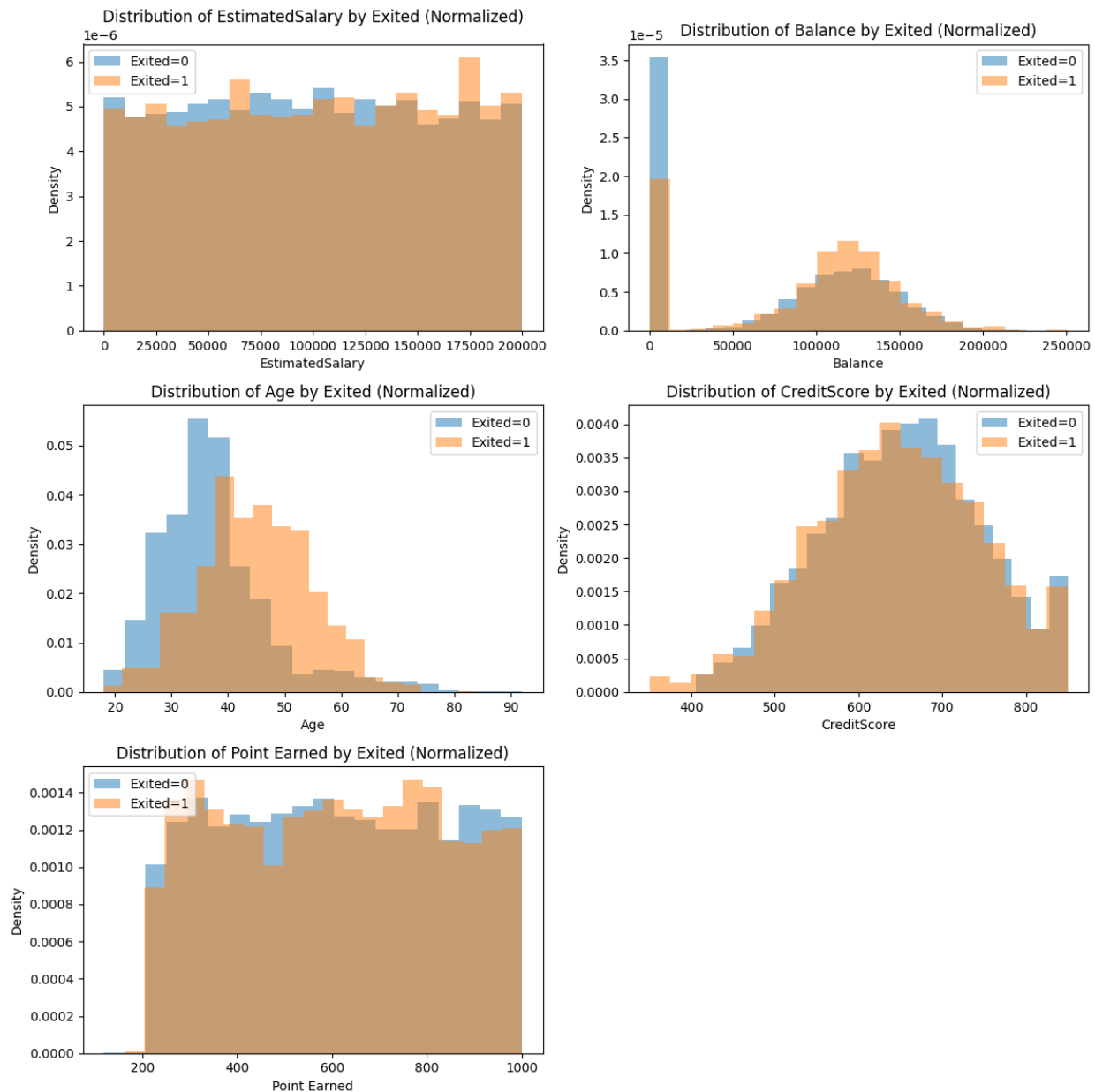


Figure 1.1.3: Raspodele numerickih atributa po vrednosti ciljne promenljive

Sa grafikona koji predstavlja raspodelu atributa Age, mozemo primetiti da postoji znacajnija razlika u raspodeli po klasama. Odnosno, da za Exited = 1 (klijenti koji su napustili) je vise pomerena u desno, odnosno da je cini starije stanovnistvo. Sto znaci, da mozemo da pretpostavimo da ce vecina modela za klasifikaciju iskoristiti upravo ovaj atribut za predvidjanje, kao i NumOfProducts.

1.2. Pretprocesiranje podataka

Matrica korelacije

Vrednosti u matrici korelacije variraju između -1 i 1. Vrednost -1 označava savršeno negativnu korelaciju, što znači da se promene u jednoj varijabli suprotno menjaju u odnosu na promene u drugoj. Vrednost 1 označava savršeno pozitivnu korelaciju, što znači da se promene u jednoj varijabli paralelno menjaju sa promenama u drugoj. Vrednost 0 ukazuje na potpunu odsutnost korelacije između varijabli.

Na sledecoj slici data je matrica korelacije naseg skupa podaka.

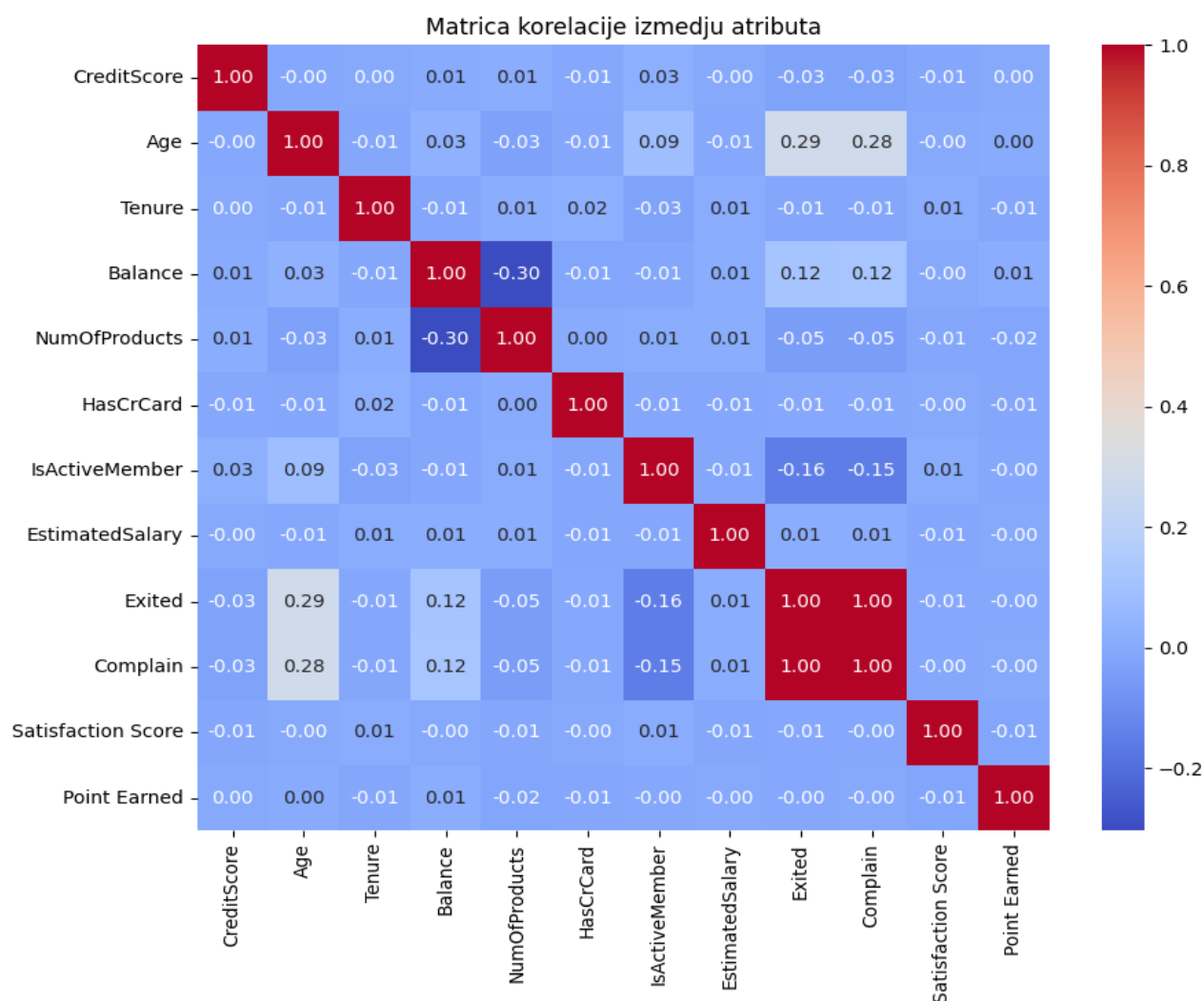


Figure 1.2.1: Matrica korelacije

Iz matice možemo primetiti da postoji apsolutna korelisanost izmedju atributa **Exited** I **Complain**, iz dalje analize izbacujemo podatke vezane za atribut **Complain**. Ukoliko to ne bismo uradili, naši modeli ne bi ništa naučili iz ostalih podataka, zato što bi se većina modela baziralo upravo na korelisanosti ove dve promenljive.

Nedostajuće vrednosti

Pošto veliki deo algoritama ne podržava rad sa nedostajućim vrednostima, pre bilo kakvog rada sa podacima trebalo bi se pozabaviti njima.

Na narednoj slici je prikazano da naši podaci ne sadrže nedostajuće vrednosti, pa deo analize vezano za njihovu nadoknadu preskačemo.

```
In [4]: data.isna().any()
```

Out[4]:	RowNumber	False
	CustomerId	False
	Surname	False
	CreditScore	False
	Geography	False
	Gender	False
	Age	False
	Tenure	False
	Balance	False
	NumOfProducts	False
	HasCrCard	False
	IsActiveMember	False
	EstimatedSalary	False
	Exited	False
	Complain	False
	Satisfaction Score	False
	Card Type	False
	Point Earned	False
	dtype:	bool

Figure 1.2.2: Nedostajuće vrednosti u skupu podataka

Balansiranost klasa

Ova pojam se odnosi na ravnotežu između broja instanci (ili primera) u svakoj klasi koju model treba da nauči prepoznati ili klasifikovati.

U našem slučaju reč je odnosu broja instanci između dve klase atributa **Exited** koj predviđamo. U nastavku je dat njihov odnos.

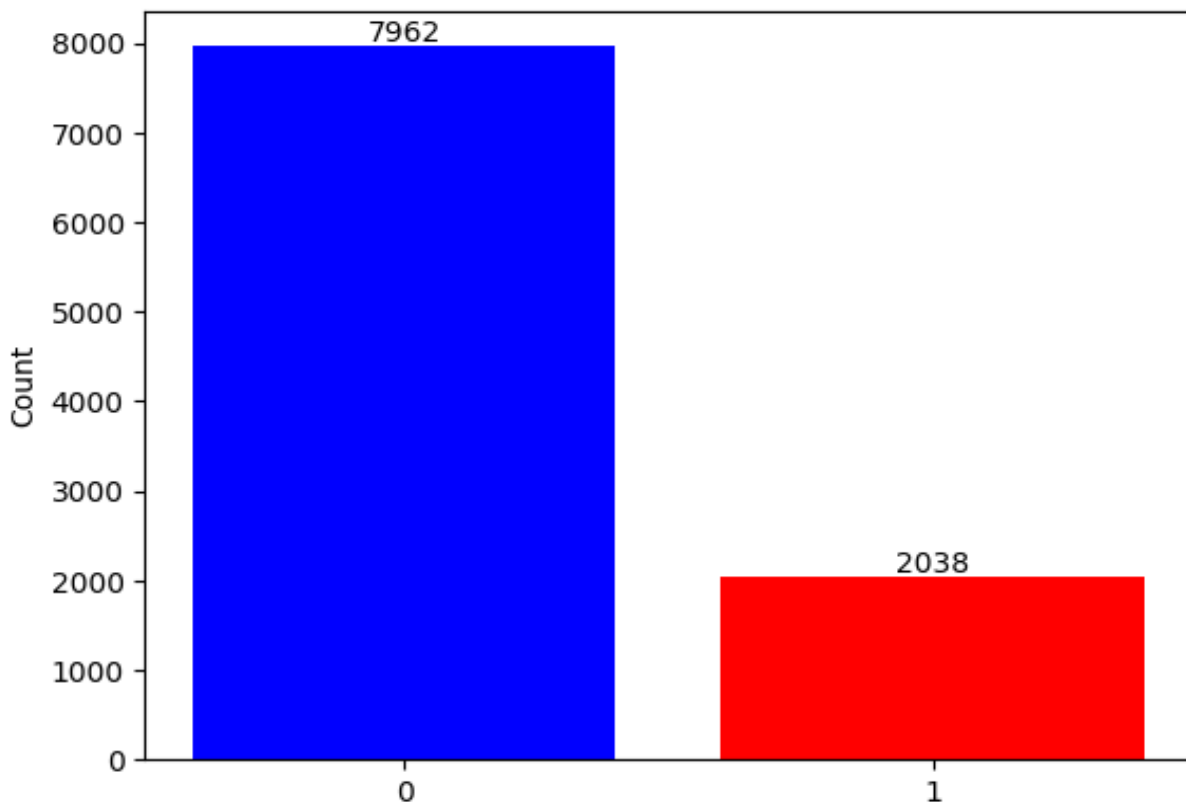


Figure 1.2.3: Odnos broja instanci *Exited*

Primetimo da postoji nebalansiranost između klasa, u odnosu 4:1. Ovaj problem se odnosi pre svega na klasifikaciju, i može uticati na kvalitet predviđanja manjinske klase.

Obrada elemenata van granica

Predstavlja važan deo tokom pretprocesiranja podataka, jer elementi van granica mogu da utiču na brojne algoritme klasifikacije i klasterovanja (npr. Algoritam K-Nearest Neighbours).

Analiza elemenata van granica u našem skupu podataka data je u nastavku.

	lower	min	num_lower	upper	max	num_upper	percentage
EstimatedSalary	-96577.09625	11.58	0	296967.45375	199992.48	0	0
Balance	-191466.36000	0.00	0	319110.60000	250898.09	0	0
Age	14.00000	18.00	0	62.00000	92.00	359	4
CreditScore	383.00000	350.00	15	919.00000	850.00	0	0
Point Earned	-176.50000	119.00	0	1387.50000	1000.00	0	0

Figure 1.2.4: Udeo elemenata van granica u numeričkim atributima

Samo atribut *Age* sadrži elemente van granica, I to 4% što ne predstavlja puno, I nećemo modifikovati dato svojstvo.

Računali smo gornju I donju granicu, koristeći sledeće formule: $upper = Q3 + (1.5 * IQR)$ I $lower = Q1 - (1.5 * IQR)$. Gde Q3 I Q1 predstavljaju redom, 25 % I 75% kvantile.

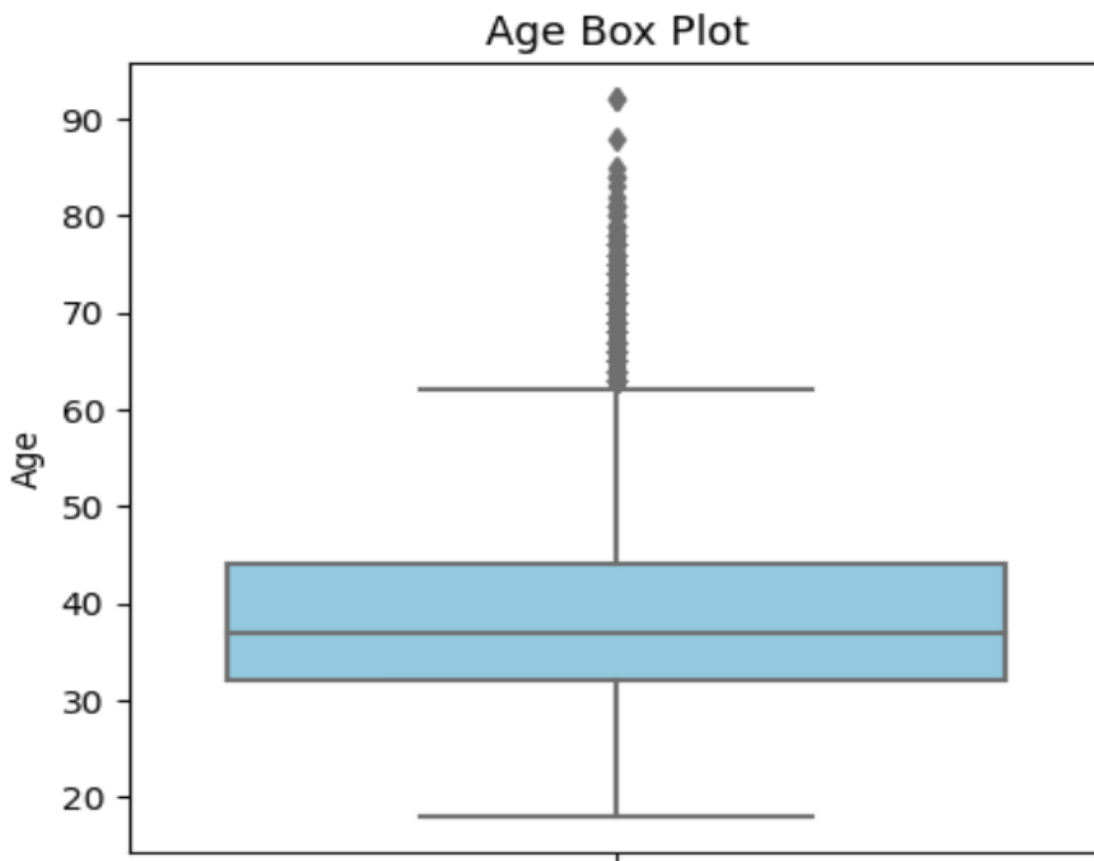


Figure 1.2.5: Box plot za atribut Age I prikaz elemenata van granica

Binarizacija kategoričkih atributa

Na samom kraju pretprocesiranja podataka, potrebno je izvršiti binarizaciju kategoričkih atributa.

Binarizacija (ili kodiranje) kategoričkih atributa je proces prevođenja tih atributa u numeričke vrednosti kako bi se omogućila upotreba tih atributa u modelima mašinskog učenja. Na uštrb povećvanja dimenzionalnosti podataka.

U našem slučaju primenili smo binarizaciju (primenom OneHotLabel Encoder-a) na atirbute Gender, Geography, Card Type.

Ovim postupkom povećali smo dimenzionalnost podataka za 6 dimenzija.

2. Klasifikacija

Klasifikacija je važan koncept u mašinskom učenju i statistici, a označava proces dodeljivanja klasa tj kategorija objektima na osnovu njihovih karakteristika ili osobina. Predstavlja metodu nadgledanog učenja.

Kokrento u našem slučaju mi predviđamo atribut **Exited**, koji može imati vrednosti 0 ili 1, na osnovu informacija o klijentu.

Primenićemo algoritme: stabla odlučivanja I SVM (metoda potpornih vektora).

Pre daljeg rada podelili smo podatke na test I trening skup, I izvršili skaliranje podataka koristeći MinMaxScaler, koji se podatke dovodi na opseg $[0,1]$. Skaliranje je neophodno uraditi, pre svega jer ćemo u nastavku koristiti SMOTE za balansiranje klasa.

2.1. Stabla odlučivanja

Ovaj algoritam donosi odluke na osnovu niza uslova ili atributa, gradeći hijerarhijsko stablo koje se sastoji od čvorova i grana. Svaki čvor u stablu predstavlja testiranje određenog atributa, dok grane vode do podstabala ili ishoda.

```

Classification report for model DecisionTreeClassifier on training data
-----
              precision    recall  f1-score   support

     0           1.00       1.00       1.00       5971
     1           1.00       1.00       1.00       1529

 accuracy          1.00
 macro avg          1.00
 weighted avg       1.00

-----
Confusion matrix for model DecisionTreeClassifier on training data
-----
      B      M
B  5971      0
M      0  1529

-----
F1 score:  1.0
Accuracy score:  1.0

```

Figure 2.1.1: Rezultati na trening skupu

```

Classification report for model DecisionTreeClassifier on test data
-----
              precision    recall  f1-score   support

     0           0.88       0.87       0.87       1991
     1           0.51       0.52       0.51        509

 accuracy          0.80
 macro avg          0.69
 weighted avg       0.80

-----
Confusion matrix for model DecisionTreeClassifier on test data
-----
      B      M
B  1731    260
M   243    266

-----
F1 score:  0.5140096618357488
Accuracy score:  0.7988

```

Figure 2.1.2: Rezultati na test skupu

Možemo приметити да је дошло до прилагођавања модела тренинг скупу податак. Ограничићемо параметре и извршити GridSearch да би нашли најбољи модел, метрика коју користимо за претрагу је f1 скор, cv параметар је постављен на 3.

GridSearchCV

```
params = {'criterion': ['gini', 'entropy'],
          'max_depth': [2,4,6,8,10],
          'class_weight':[
              {1:1, 0:1},
              {1:2, 0:1},
              {1:3, 0:1}
          ]
        }
```

Figure 2.1.3: Parametri koji se prosledjuju GridSeachCV

Nakon primene GridSearch-a dobijamo da su najbolji paramteri:

```
{'class_weight': {1: 2, 0: 1}, 'criterion': 'gini',
'max_depth': 6}
```

Classification report for model DecisionTreeClassifier on test data

	precision	recall	f1-score	support
0	0.89	0.91	0.90	1991
1	0.62	0.57	0.59	509
accuracy			0.84	2500
macro avg	0.76	0.74	0.75	2500
weighted avg	0.84	0.84	0.84	2500

Confusion matrix for model DecisionTreeClassifier on test data

	B	M
B	1812	179
M	219	290

F1 score: 0.5930470347648262
Accuracy score: 0.8408

Figure 2.1.4: Rezultati na test skupu nakon primene GridSearch-a

Mozemo primetiti da su nam se povećali kako f1 skor, tako i preciznost modela.

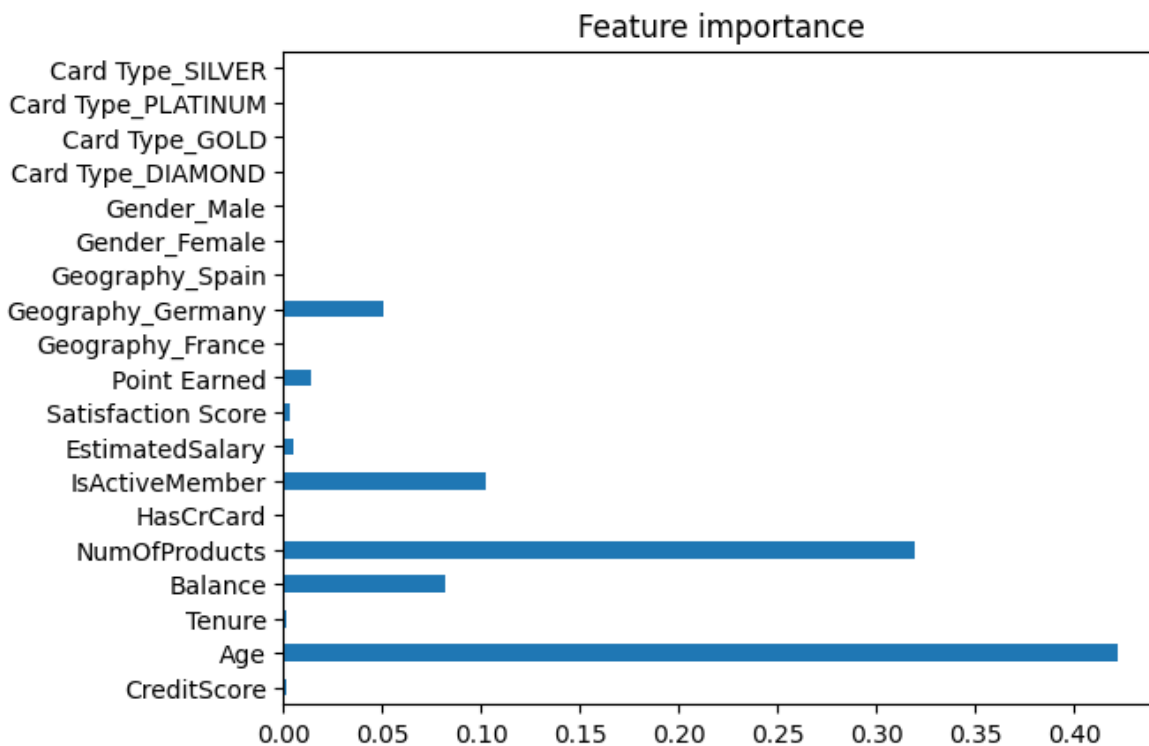


Figure 2.1.5: Značaj atributa za DecisionTree model

Primetimo da upravo važi ono što smo zaključili tokom inicijalnog analiziranja podataka.

Decision tree of depth 6 with 52 nodes

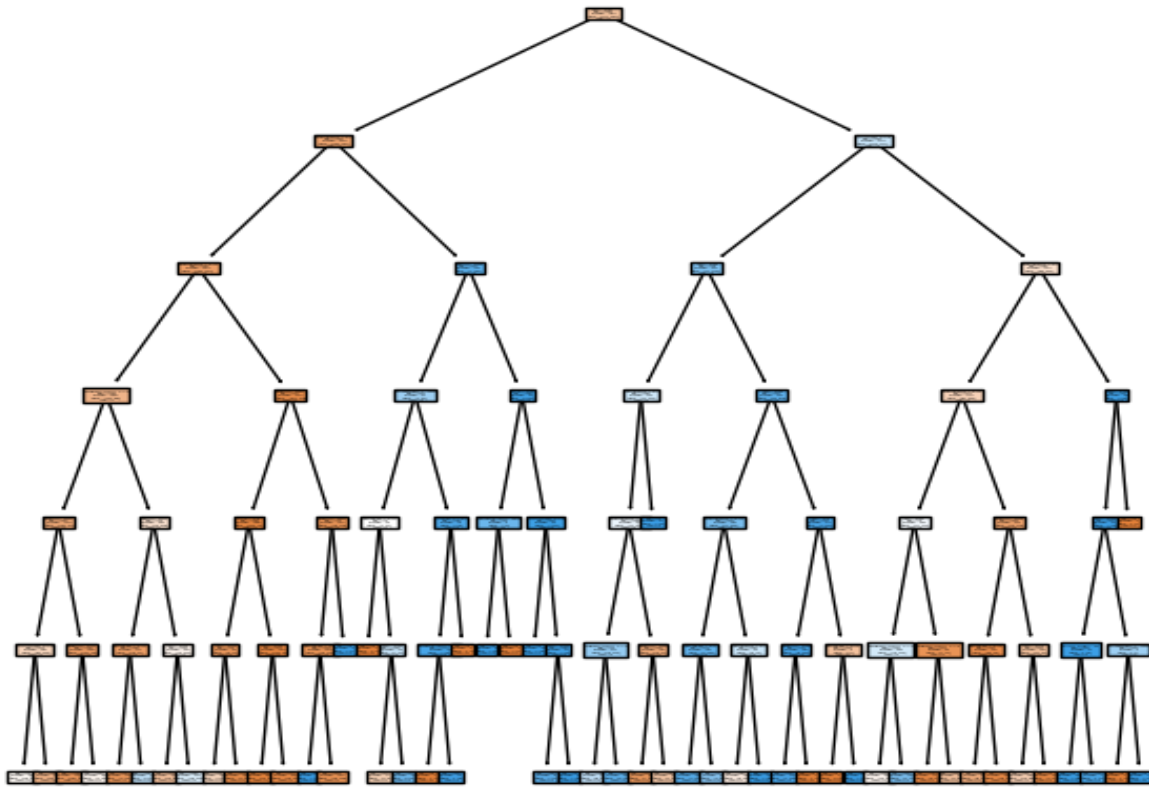


Figure 2.1.5: Struktura drveta odlučivanja, koju smo dobili treniranjem modela

Slučajne šume

Slučajna šuma (Random Forest) je ansambl algoritam u mašinskom učenju koji se koristi za klasifikaciju i regresiju. Funkcioniše tako što kombinuje mnogo stabala odlučivanja kako bi se poboljšala tačnost i smanjio rizik od preprilagođavanja (overfitting).

Classification report for model RandomForestClassifier on test data

	precision	recall	f1-score	support
0	0.88	0.97	0.92	1991
1	0.79	0.46	0.58	509
accuracy			0.87	2500
macro avg	0.83	0.72	0.75	2500
weighted avg	0.86	0.87	0.85	2500

Confusion matrix for model RandomForestClassifier on test data

	B	M
B	1928	63
M	274	235

F1 score: 0.5824039653035936

Accuracy score: 0.8652

Figure 2.1.6: Rezultati RandomForrest

Primenicemo GridSearch da bi poboljšali model, sa sledećim parametrima :

```
parameters = {'max_depth': [2,4,6,8,10],
              'criterion': ['entropy', 'gini', 'log_loss'],
              'n_estimators': [50, 100, 200, 300]}
}
```

Classification report for model RandomForestClassifier on test data

	precision	recall	f1-score	support
0	0.89	0.95	0.92	1991
1	0.72	0.53	0.61	509
accuracy			0.86	2500
macro avg	0.80	0.74	0.76	2500
weighted avg	0.85	0.86	0.85	2500

Confusion matrix for model RandomForestClassifier on test data

	B	M
B	1886	105
M	240	269

F1 score: 0.609286523216308

Accuracy score: 0.862

Figure 2.1.6: GridSearchCv primenjen na RandomForrest

Možemo primetiti da negativnu klasu modeli dosta bolje pogodjaju, to je posledica nebalansiranosti klasa u skupu podataka. Problem nebalansiranosti probaćemo da rešimo primenom OverSampling tehnike (prilikom UnderSampling gubimo ogromnu količinu informacija, dok je OverSampling dao empirijski bolje rezultate od Over-Under sampling-a na ovom skupu podaka).

Koristićemo SMOTE algoritam, nakon kojeg ćemo opet izvršiti treniranje.

Classification report for model RandomForestClassifier on training data

	precision	recall	f1-score	support
0	0.91	0.88	0.89	1991
1	0.58	0.66	0.62	509
accuracy			0.83	2500
macro avg	0.75	0.77	0.76	2500
weighted avg	0.84	0.83	0.84	2500

Confusion matrix for model RandomForestClassifier on training data

	B	M
B	1747	244
M	172	337

F1 score: 0.618348623853211

Accuracy score: 0.8336

Figure 2.1.7: Primena RandomForresta nakon balansiranja

Mozemo primetiti da nam je se povecao recall na uštrb preciznosti i da su rezultati priblizno isti.

2.2. SVM (metoda potpornih vektora)

SVM (Support Vector Machine) funkcioniše tako da pronađe hiperravan (u dvodimenzionalnom prostoru to je prava linija, a u višedimenzionalnom prostoru to je ravna površina) koja najbolje razdvaja dve klase podataka u skupu za učenje. Cilj je postići maksimalnu marginu između te hiperravni i najbližih tačaka oba razreda.

SVM može koristiti različite kernel funkcije da preslika podatke u višedimenzionalni prostor kako bi se omogućilo bolje razdvajanje ako je problem nelinearan. Kerneli su funkcije koje omogućavaju transformaciju podataka u prostor više dimenzija.

Osetljiv je na neskaliране podatke, pošto računa rastojanja između tačaka kako bi pronašao optimalnu razdvajajuću ravan. Zato pre primene samog algoritma izvršićemo skalariranje koristeći StandarScaler.

Da bi podesili hiperparametre modela koristićemo GridSearch. Parametri :

```
params = [  
    {  
        'kernel': ['linear'],  
        'C': [0.01, 0.1, 1, 10],  
    },  
    {  
        'kernel': ['rbf'],  
        'C': [0.01, 0.1, 1, 10],  
        'gamma': [0.01, 0.1, 1, 10],  
    },  
]
```

```

Classification report for model SVC on test data
-----
              precision    recall  f1-score   support

     0       0.89        0.92        0.91        1991
     1       0.64        0.56        0.60         509

 accuracy          0.85        2500
 macro avg         0.77        0.74        0.75        2500
 weighted avg      0.84        0.85        0.84        2500

-----
Confusion matrix for model SVC on test data
-----
      B      M
B  1831    160
M   222    287

-----
F1 score:  0.600418410041841
Accuracy score:  0.8472

```

Figure 2.2.1: Rezultati za SVM korišćenjem GridSearch-a

HistGradientBoostingClassifier

GradientBoostingClassifier je algoritam nadgledanog učenja koji spada u kategoriju ansambl modela.

Ansambl modeli kombinuju više osnovnih modela kako bi poboljšali prediktivnu snagu. Ključna ideja iza Gradient Boostinga je da se svaki sledeći osnovni model fokusira na ispravljanje grešaka koje su prethodni modeli napravili.

HistGradientBoostingClassifier je varijanta GradientBoostingClassifier-a koja je optimizovana za brzinu i efikasnost.

Ključna ideja histograma je grupiranje podataka u diskretne intervale, tzv. "korpe" (bins), i računanje statistika unutar svake korpe.

Pošto ćemo pokušati da optimizujemo algoritam na velikom broju parametara, umesto GridSearch-a koristićemo RandomizedSearchCv zbog nedostatka hardversih resursa.

Za razliku od GridSearch-a on nasumično izabira određeni broj kombinacija hiperparametara iz hiperparametarskog prostora. To znači da nema garancije da će pronaći najbolju kombinaciju hiperparametara, ali je verovatno da će naći dovoljno dobru kombinaciju za mnogo manje vremena nego GridSearch.

Prostor hiperparametara :

```

param_grid = {
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 4, 5],
    'max_iter': [100, 200, 300],
    'max_leaf_nodes': [15, 31, 63],
    'min_samples_leaf': [1, 2, 4],
    'l2_regularization': [0.0, 0.1, 0.2],
    'max_bins': [50, 100, 255]
}

```

	precision	recall	f1-score	support
0	0.91	0.91	0.91	1991
1	0.64	0.63	0.64	509
accuracy			0.85	2500
macro avg	0.77	0.77	0.77	2500
weighted avg	0.85	0.85	0.85	2500

Confusion matrix for model HistGradientBoostingClassifier

	B	M
B	1811	180
M	188	321

F1 score: 0.6356435643564358
Accuracy score: 0.8528

Figure 2.2.2: HistGradientBoostingClassifier-a

```

Classification report for model HistGradientBoostingClassifier on test data
-----
              precision    recall  f1-score   support

     0           0.90       0.91       0.91       1991
     1           0.63       0.62       0.63        509

 accuracy          0.85       0.85       0.85       2500
 macro avg         0.77       0.76       0.77       2500
 weighted avg      0.85       0.85       0.85       2500

-----
Confusion matrix for model HistGradientBoostingClassifier on test data
-----
      B      M
B  1806   185
M   192   317

-----
F1 score:  0.6271018793273986
Accuracy score:  0.8492

```

Figure 2.2.3: HistGradientBoostingClassifier I SMOTE balansiranje

2.3. Poredjenje modela

Za poredjenje modela koristimo ROC krivu i metriku AUC.

AUC (Area Under the Curve) je statistička metrika koja se često koristi za evaluaciju performansi binarnih klasifikacionih modela. AUC meri površinu ispod ROC krive (Receiver Operating Characteristic curve), a ROC kriva prikazuje odnos između stope istinito pozitivnih (True Positive Rate, TPR) i stope lažno pozitivnih (False Positive Rate, FPR) za različite pragove klasifikacije.

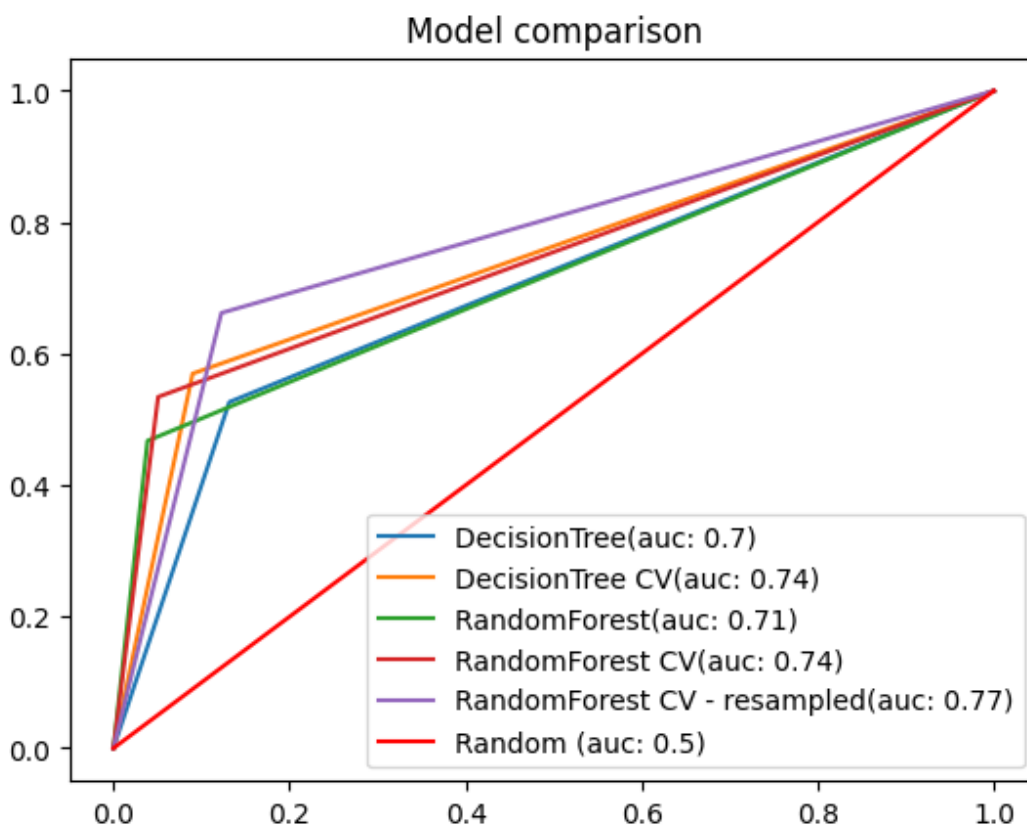


Figure 2.3.1: ROC kriva za prvi skup modela

Po vrednosti AUC, možemo zaključiti da je najbolje rezultate dao RandomForest nakon primene SMOTE algoritma.

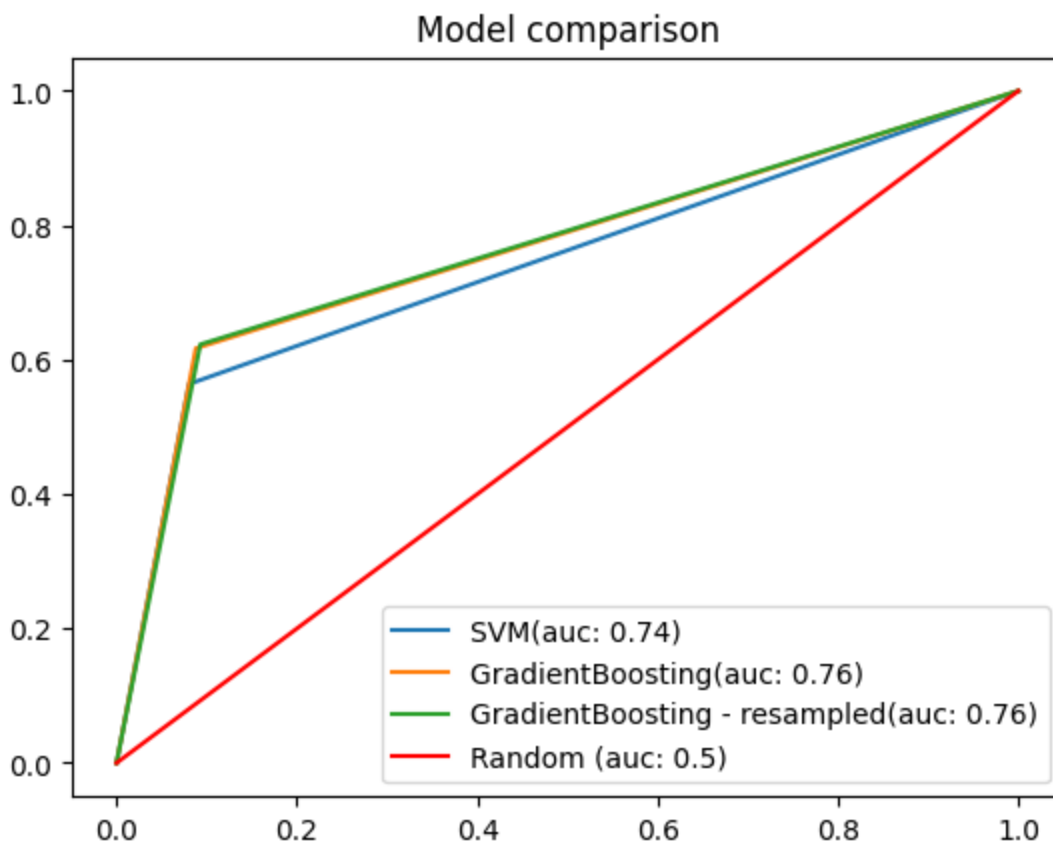


Figure 2.3.2 : ROC kriva za drugi skup modela

Iz ovog skupa modela, najbolje se pokazao GradientBoostingClassifier, koji je približnog kvaliteta kao najbolji model iz prvog skupa.

3. Klasterovanje

Klasterovanje je tehnika nenadgledanog učenja u oblasti analize podataka koja se koristi za grupisanje sličnih podataka zajedno u klastere ili grupe.

Glavni cilj klasterovanja je otkriti prirodne strukture u podacima, tako da slični podaci budu u istom klasteru, dok se različiti podaci nalaze u različitim klasterima.

Koristićemo metode: K-sredina I hijerarhijsko klasterovanje.

Pre daljeg rada, a pošto se algoritmi baziraju na razdaljinama, potrebno je izvršiti skaliranje podataka što smo mi I učinili, koristeći StandarScaler.

3.1. K – sredina

Glavni cilj K-sredina je minimizacija unutarklasterne varijanse, tj. razdaljine između tačaka unutar istog klastera. Algoritam pokušava pronaći K klastera tako da suma kvadrata udaljenosti između svake tačke i centra njenog klastera bude minimizovana.

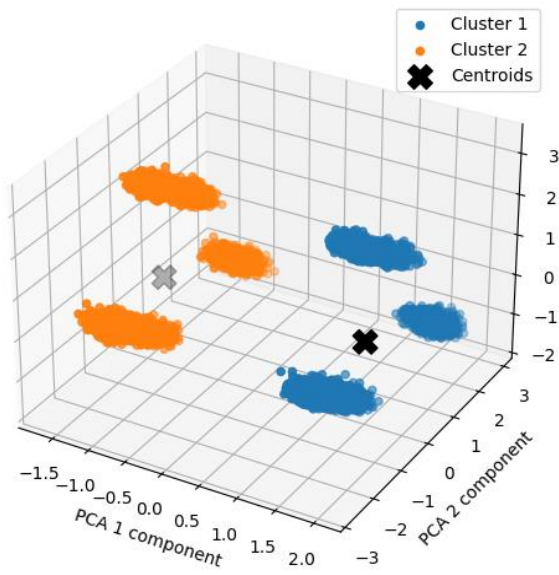
Algoritam počinje slučajnim izborom K početnih centara klastera. Zatim se iterativno izvršava dva koraka: prvo, svaka tačka se dodeljuje klasteru čijem centru je najbliža, a zatim se centri klastera ponovno izračunavaju kao srednje vrednosti svih tačaka u svakom klasteru.

PCA redukcija

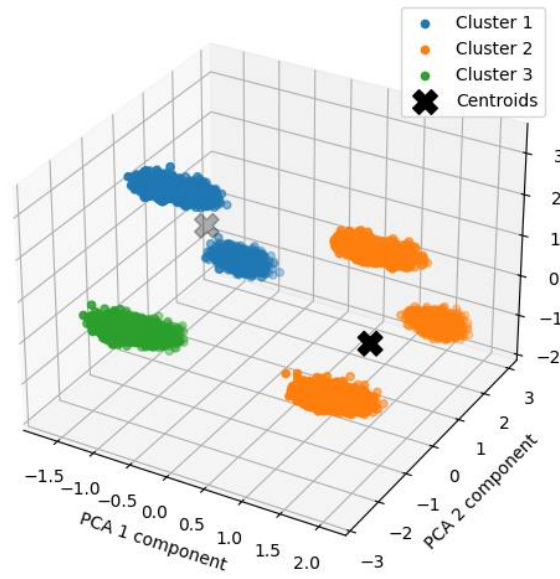
Skup podataka je visoko dimenziolan, tačnije imamo 18 dimenzija, potrebno je izvršiti dimenzionalnu redukciju skupa podataka pre svega zbog problema: “prokletstva dimenzionosti” , poboljšanja intepretacije, smanjenje šuma a I smanjenje računarskog opterećenja.

Nakon primene PCA metode naš skup sada ima 3 nova atributa, uz ukupno objašnjenu varijansu od 28.33 %.

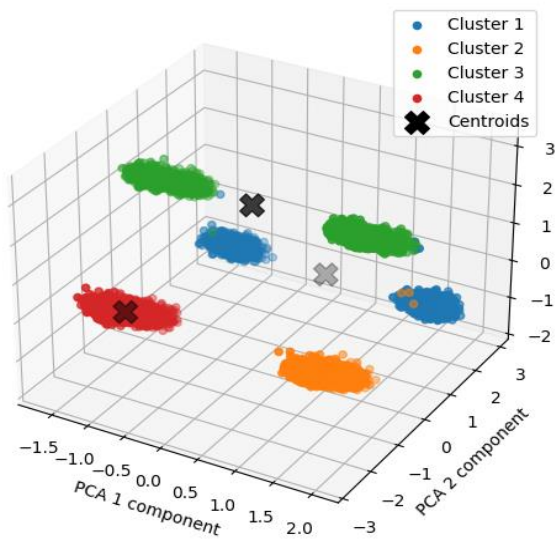
K means for 2



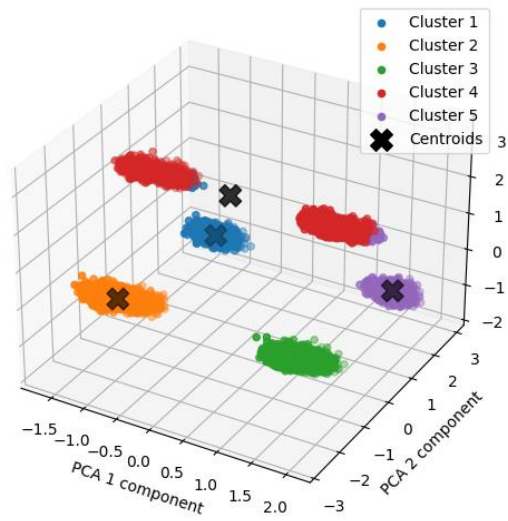
K means for 3



K means for 4



K means for 5



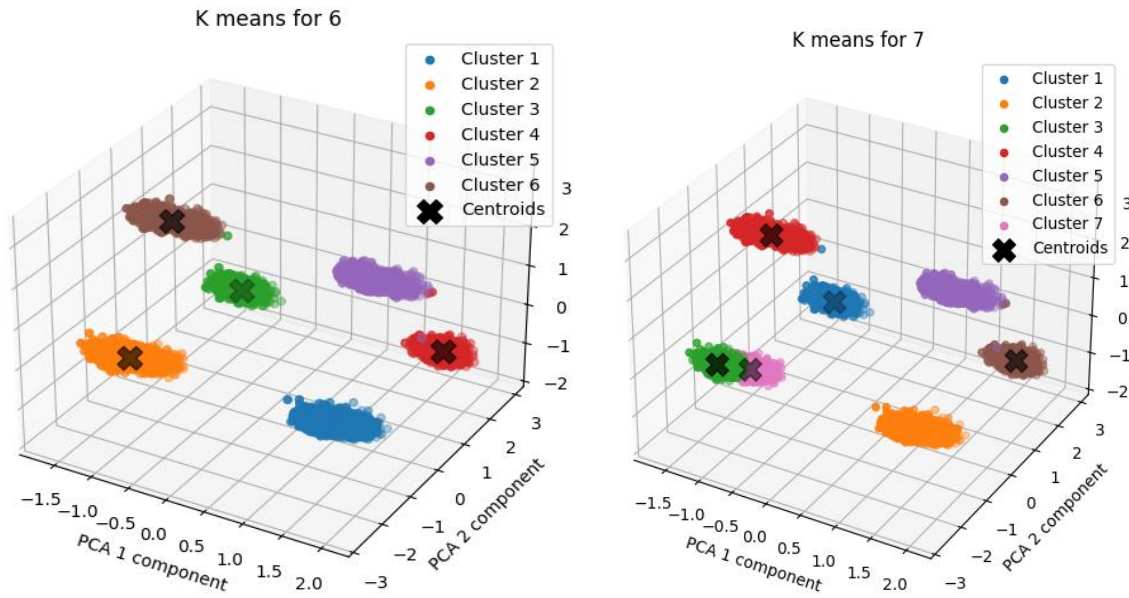


Figure 3.1.1: Klasteri i centroid za različite vrednosti K

Možemo odmah primetiti da se jasno izdvaja 6 klastera. To ćemo utvrditi na osnovu SSE (suma kvadratnih gresaka) i koeficijenta senki.



Figure 3.1.2: Koeficijenti senki



Figure 3.1.3: SSE za različite vrednosti K

Na osnovu grafika možemo utvrditi isto ono što smo I sa same vizualizacije klastera.

t-SNE dimenzionalna redukcija

Predstavlja drugu metodu dimenzionalne redukcije podataka. Glavne razlike između PCA i t-SNE predstavljaju.

Tip transformacije:

PCA je linearna transformacija podataka. To znači da novi atributi (glavne komponente) koje se generišu kao rezultat PCA-a su linearna kombinacija originalnih značajki.

t-SNE je nelinearna transformacija podataka. Ona očuvava sličnosti između tačaka u originalnom prostoru tako da slične tačke budu bliže jedna drugoj u smanjenom prostoru, ali to ne čini linearno.

Očuvanje varijacije:

PCA se fokusira na očuvanje maksimalne varijacije u podacima. Prve glavne komponente zadržavaju veći deo varijacije u podacima.

t-SNE se ne fokusira na očuvanje varijacije. Umesto toga, t-SNE se fokusira na očuvanje sličnosti između tačaka, čime se stvaraju gusti klasteri.

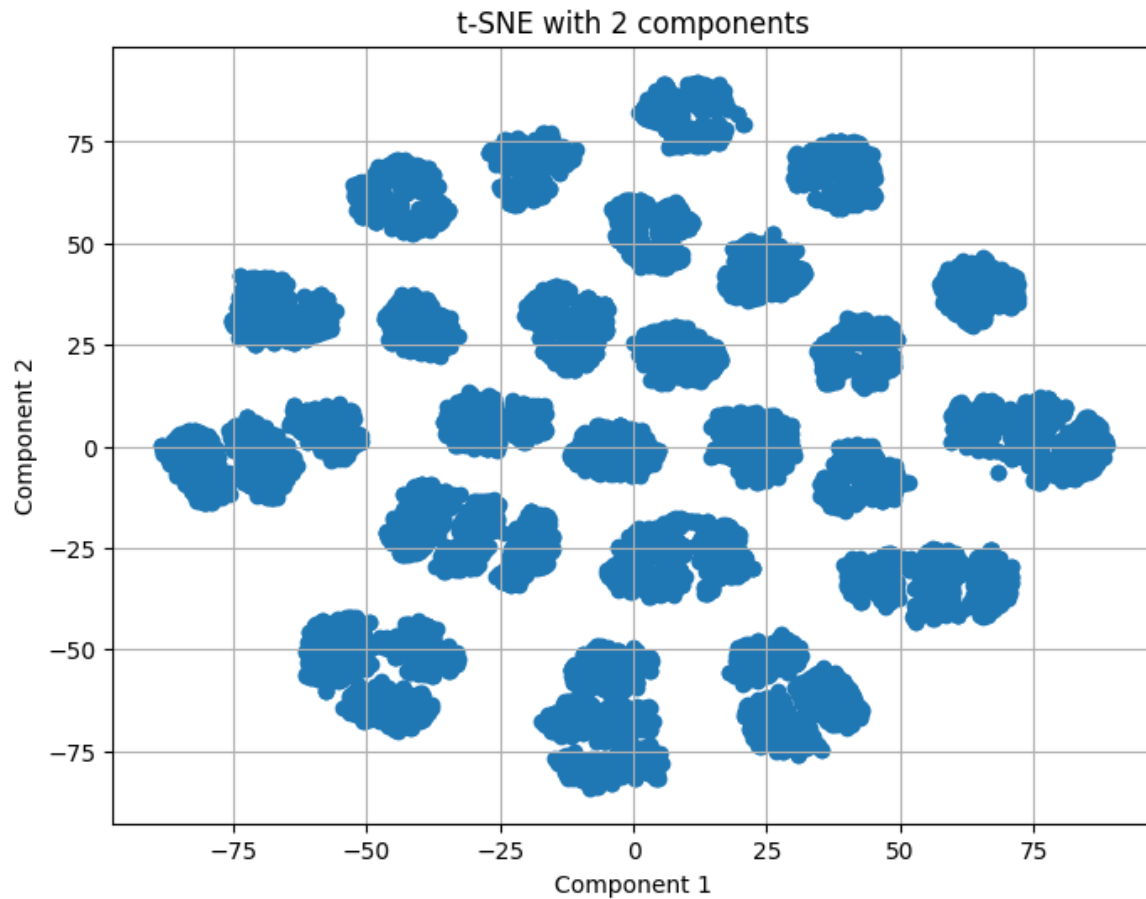


Figure 3.1.3: podaci nakon primene t-SNE

Možemo primetiti da je se opet iskristalisao određeni broj grupa što može biti posledica velikog broja kategoričkih atributa.

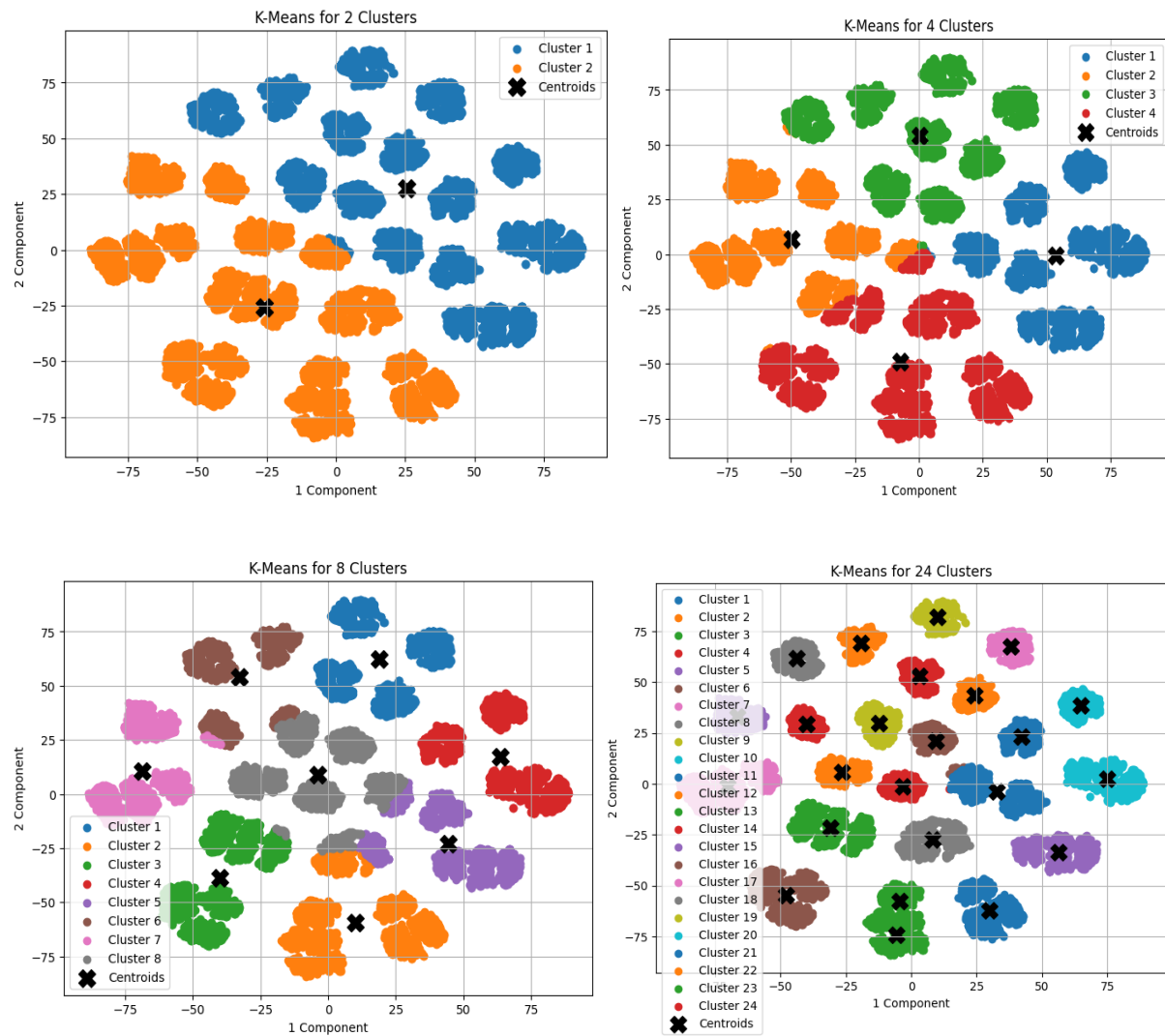


Figure 3.1.4: K - means za različite vrednosti parametra K, t-SNE

U nastavku će biti dati I grafici SSE I koeficijenta senki.

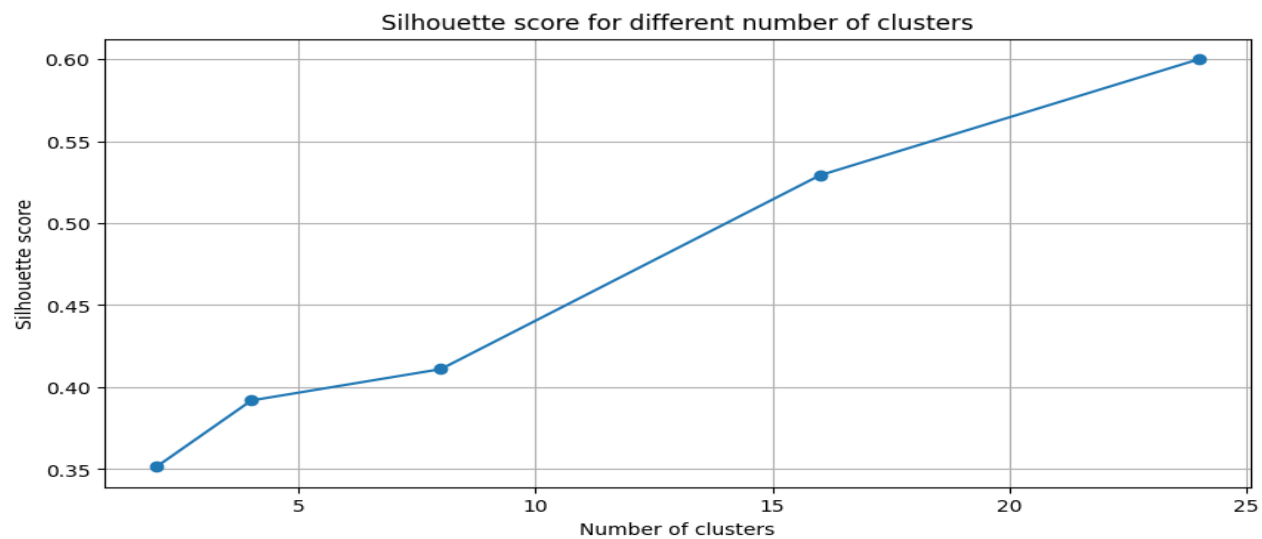


Figure 3.1.5: Koefcijenti senki



Figure 3.1.6: SSE

Ono što smo mogli utvrditi I samim prebrojavanjem klastera, a što nam potvrđuju I grafici iznad, da je optimlan broj klastera oko 24.

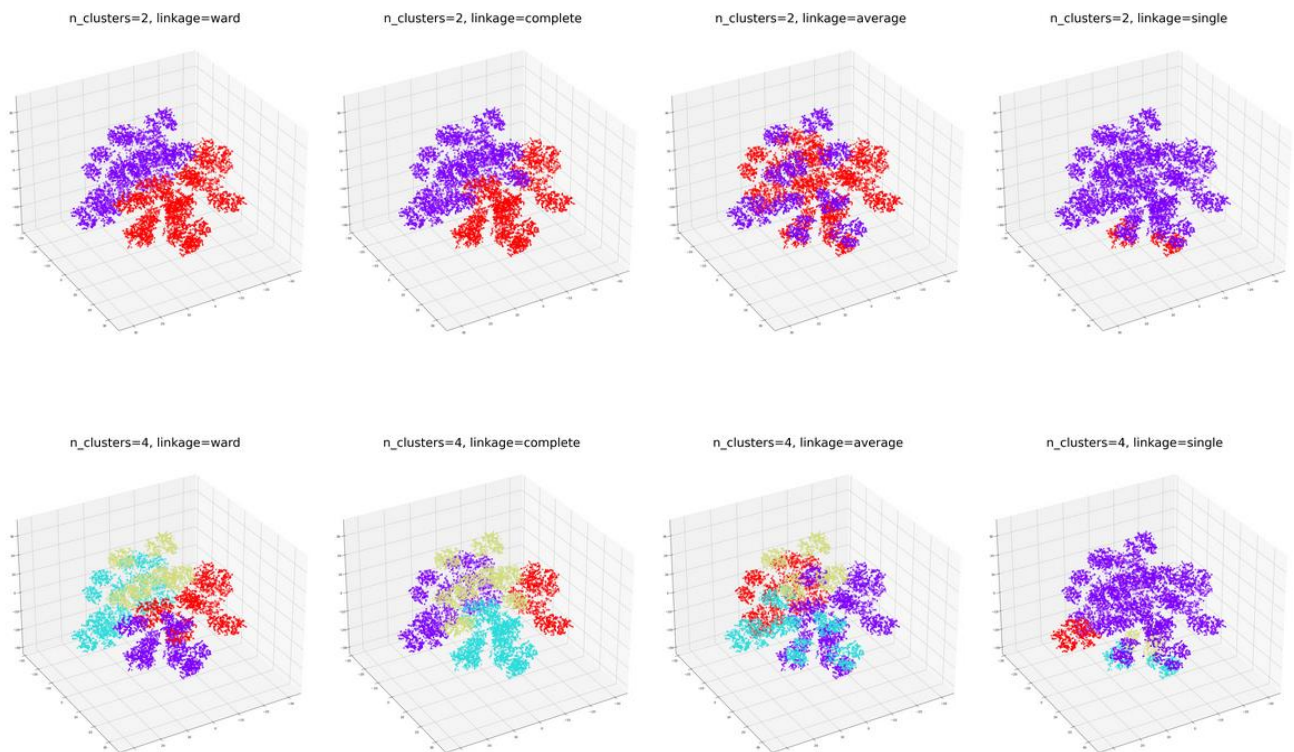
3.2. Hijerarhijsko - sakupljajuće klasterovanje

Hijerarhijsko sakupljajuće klasterovanje je tehnika klasterovanja koja ima za cilj grupisanje podataka u hijerarhijsku strukturu klastera.

Ova tehnika počinje sa svakim podatkom kao pojedinačnim klasterom i postepeno spaja (aglomeriše) slične klastera kako bi se formirala hijerarhijska stabla klastera. Ovo stablo se često prikazuje u obliku dendrograma.

Pre samog nastavka izvršili smo dimenzionu redukciju pomoću metode t-SNE, pošto smo prilikom rada sa PCA već pronašli optimalan broj klastera.

U potrazi za najboljom strategijom isprobavamo *ward*, *single*, *complete* i *average* sa različitim vrednostima hiper – parametra K.



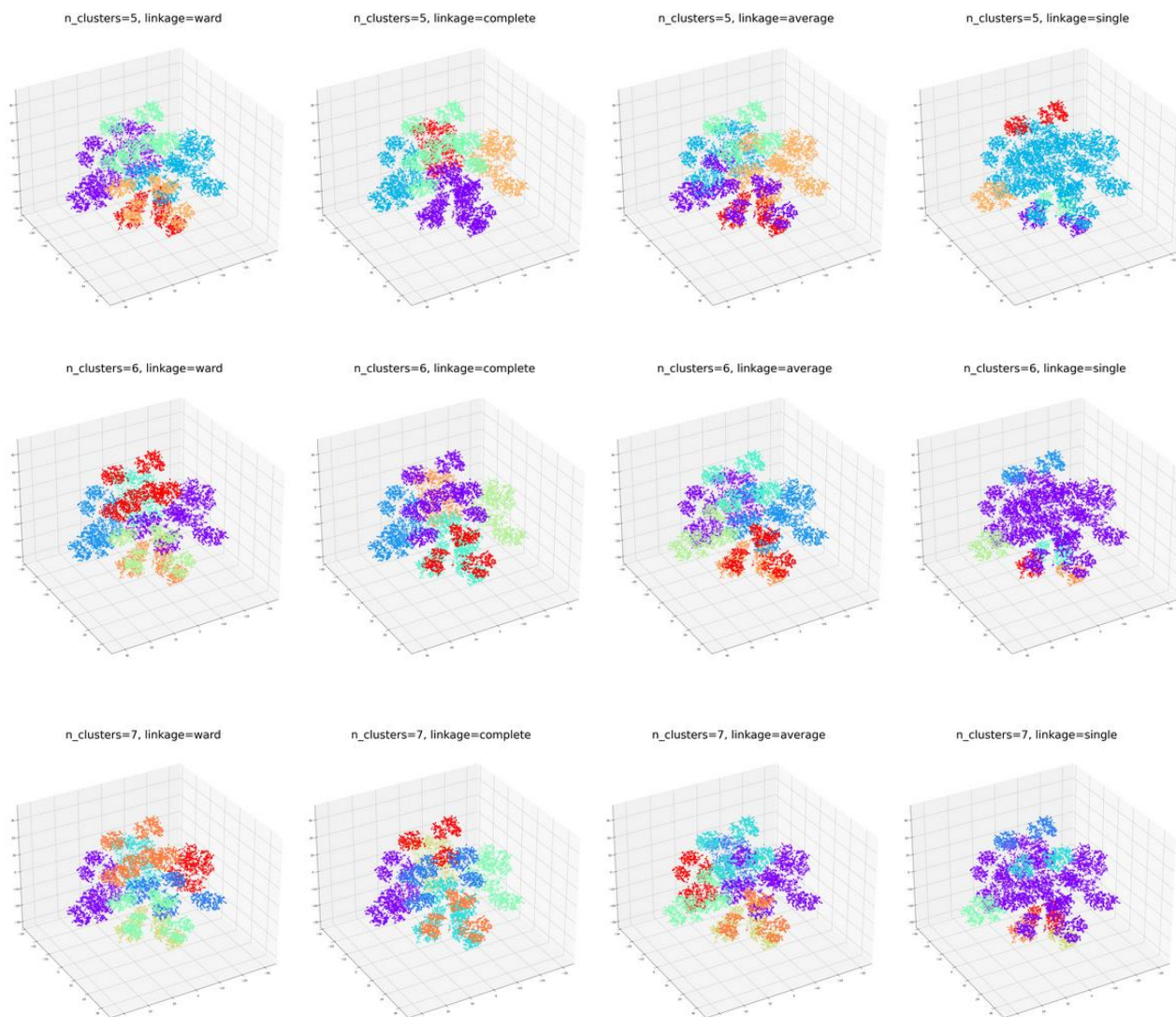


Figure 3.2.1: Hijearhijsko klasterovanje

Poredeći koeficiente senki, možemo zaključiti da je od svih strategija najbolja ward sa 7 klastera.

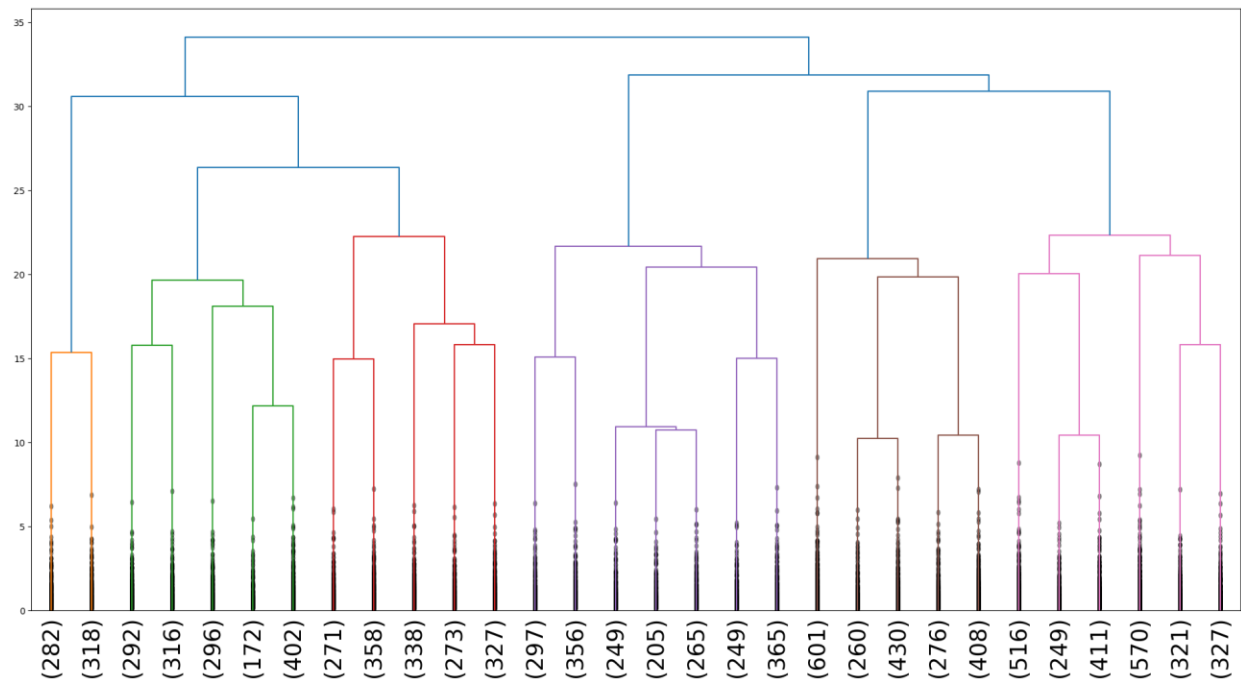


Figure 3.2.2: Dendrogram – strategija ward

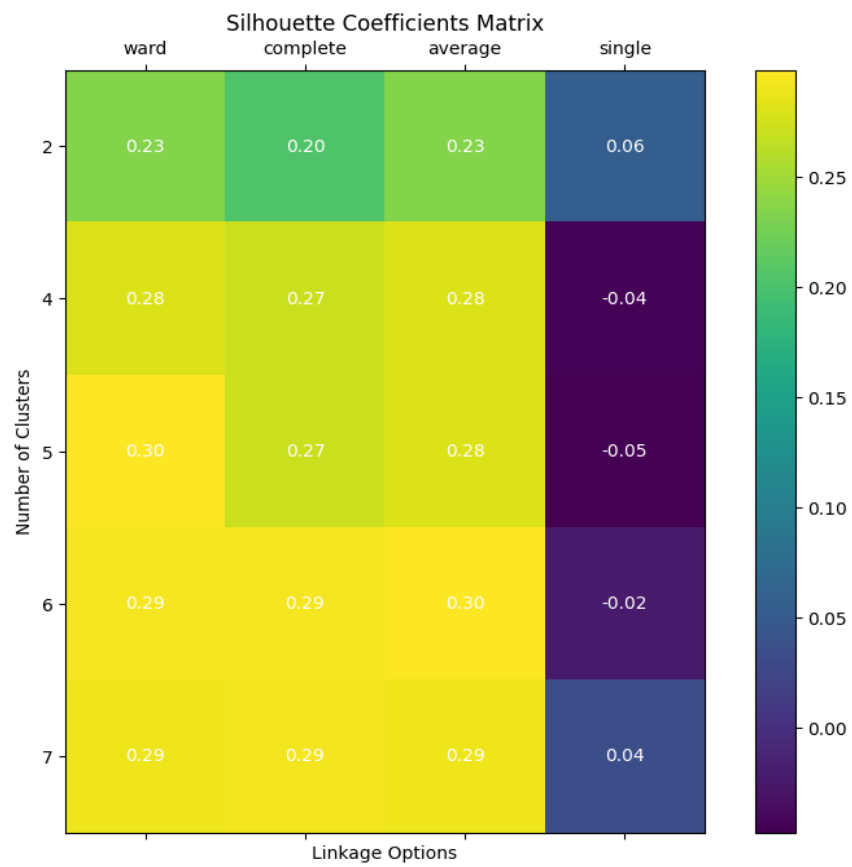


Figure 3.2.3: Matrica koeficijenata senki

4. Pravila pridruživanja

Služe za otkrivanje zanimljivih povezanosti izmedju podataka. Na našem skupu podataka primenili smo Apriori algoritam, korišćenjem alata *SPSS Modeler*.

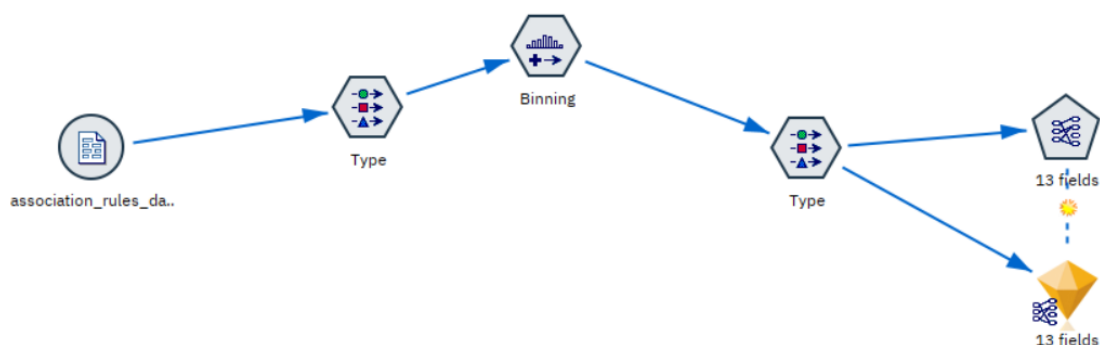


Figure 4.1: Korišćeni pipeline u alatu SPSS

Koristili smo čvor *Binning* koji nam je izvršio diskretizaciju numeričkih atributa atributa, a za sama pravila pridruživanja korišćen je čvor *Apriori*.

Bin	Lower	Upper
1	≥ 350	< 450
2	≥ 450	< 550
3	≥ 550	< 650
4	≥ 650	< 750
5	≥ 750	≤ 850

CreditScore bins

Bin	Lower	Upper
1	≥ 18	< 32.8
2	≥ 32.8	< 47.6
3	≥ 47.6	< 62.4
4	≥ 62.4	< 77.2
5	≥ 77.2	≤ 92

Age Bins

Bin	Lower	Upper
1	>= 0	< 50179.618
2	>= 50179.618	< 100359.236
3	>= 100359.236	< 150538.854
4	>= 150538.854	< 200718.472
5	>= 200718.472	<= 250898.09

Bin	Lower	Upper
1	>= 11.58	< 40007.76
2	>= 40007.76	< 80003.94
3	>= 80003.94	< 120000.12
4	>= 120000.12	< 159996.3
5	>= 159996.3	<= 199992.48

Balance bins

Bin	Lower	Upper
1	>= 119	< 295.2
2	>= 295.2	< 471.4
3	>= 471.4	< 647.6
4	>= 647.6	< 823.8
5	>= 823.8	<= 1000

EstimatedSalary bins

PointEarned bins

Koristićemo metriku *Lift* da bi izabrali zanimljiva pravila.

Uslovi koji pravila moraju da ispunjavaju :

Minimum antecedent support (%):

Minimum rule confidence (%):

Maximum number of antecedents:

Najzanimljivija pravila koja smo dobili ukoliko se kao Consequent (desna strana pravila) nalazi ciljni atribut.

Consequent	Antecedent	Support %	Confidence %	Lift
Exited = Exited	Age_BIN = 3 Gender = Female IsActiveMember = 0 NumOfProducts = 1	2.11	86.256	4.232
Exited = Exited	Age_BIN = 3 IsActiveMember = 0 NumOfProducts = 1	4.1	83.171	4.081
Exited = Exited	Age_BIN = 3 Geography = Germany IsActiveMember = 0	2.28	82.895	4.067
Exited = Exited	NumOfProducts = 3	2.66	82.707	4.058
Exited = Exited	Age_BIN = 3 IsActiveMember = 0 NumOfProducts = 1 HasCrCard = 1	2.93	82.594	4.053
Exited = Exited	Age_BIN = 3 Gender = Female IsActiveMember = 0 HasCrCard = 1	2.2	80.0	3.925
Exited = Not Exited	Point Earned_BIN = 4 Balance_BIN = 1 NumOfProducts = 2 Geography = France Gender = Male	2.05	100.0	1.256
Exited = Not Exited	Age_BIN = 1 HasCrCard = 0	2.44	99.59	1.251

Figure 4.2: Pravila koja smo dobili Apriori algoritmom, sortirano po Liftu opadajuće

Možemo primetiti da je za prva šest pravila Lift poprilično visok, ova pravila nam takodje mogu dati značajan uvid u to koji klijenti zapravo napuštaju banku, takodje vidimo da je *support* približno jednak 2%, što predstavlja poprilično usku ciljnu grupu.

Promenom broja kategorija na koliko delimo numeričke attribute možemo dobiti nešto drugačija pravila, koja se takođe mogu iskoristiti.

Consequent	Antecedent	Support %	Confidence %	Lift
IsActiveMember	Age_BIN = 4 Exited	2.63	95.057	1.845
IsActiveMember	Age_BIN = 4	3.35	82.687	1.605
IsActiveMember	Age_BIN = 4 HasCrCard	2.4	82.083	1.594
IsActiveMember	Age_BIN = 3 Balance_BIN = 1 Exited	2.36	80.508	1.563
IsActiveMember	Age_BIN = 3 Geography = France HasCrCard Exited	2.13	80.282	1.559

Figure 4.3: Pravila gde nije sa desne strane Exited, pouzdanost veća od 80%, sortirano po Liftu

Pravila koja smo dobili a koja na ne impliciraju Exited, za njih možemo primetiti da imaju nešto niži lift I da desnu stranu pravila čini IsActiveMember, sva ostala pravila imaju dosta niži ovaj parametar.

5. Zaključak

Bankarska industrija je dinamična i konkurentna sfera poslovanja. Banke moraju pažljivo pratiti tržište, tehnološke trendove i regulatorne promene kako bi se prilagodile i ostvarile uspeh u ovom okruženju. Razumevanje konkurencije i tržišnih faktora ključno je za dugoročni uspeh u bankarskoj industriji.

Iz same početne analize mogli smo zaključiti da će atributi imati *Age* i *NumOfProducts* imati veliki značaj u predviđanju, što smo kasnije mogli da vidimo i kod stabala odlučivanja, ali i kod pravila pridruživanja (možemo primetiti da se nalaze u svim pravilima i to sa leve strane).

Od modela klasifikacije najbolje su se pokazali HistGradientBoosting i slučajne šume, koji su ansambl metodi.

Što se tiče klasterovanja ono nije bilo previše izazovno, jer i sami podaci su pre svega namenjeni za binarnu klasifikaciju. Ali mogli smo primetiti da se jasno izdvajaju klasteri.

Na kraju smo dobili i određeni broj zanimljivih pravila pridruživanja, koji nam mogu dati uvid koje grupacije su sklone napuštanju, i u budućnosti unapređivanje usluga bazirati na njima.