UNIVERZITET U BEOGRADU
MATEMATICKI FAKULTET

Customer Personality Analysis

Projekat iz Istrazivanja podataka 1[R274]

Stefan Mitrovic 350/2020

Avgust 2023

Sadrzaj

1. Uvod

U ovom projektu analiziramo podatke baze podataka “Customer Personality Analysis”. Baza podataka sadrzi informacije koje opisuju licnosti kupaca. Cilj ovog projekta je da predvidimo da li ce neki kupac prihvatiti ponudu tokom neke promocije koristeci razne tehnike i algoritme.

2. Opis podataka i neka svojstva

Nasa baza podataka sa kojom radim nije nesto velika ima svega 2240 instanci i 29 atributa. Atributi su podeljeni u 4 sekcije koje cine opis coveka, opis proizvoda, opis promocije i opis mesta na koji je izvrsena odredjena transakcija. Imamo 3 atributa koji su kategorickog tipa(Education, Marital_Status, Dt_Customer) a svi ostali su numerickog.

People

- ID:
- Year_Birth:
- Education:
- Marital_Status:
- Income:
- Kidhome:
- Teenhome:
- Dt_Customer:
- Recency:
- Complain:

Products

- MntWines:
- MntFruits:
- MntMeatProducts:
- MntFishProducts:
- MntSweetProducts:
- MntGoldProds:

Promotion

- NumDealsPurchases:
- AcceptedCmp1:
- AcceptedCmp2:
- AcceptedCmp3:
- AcceptedCmp4:
- AcceptedCmp5:
- Response:

Place

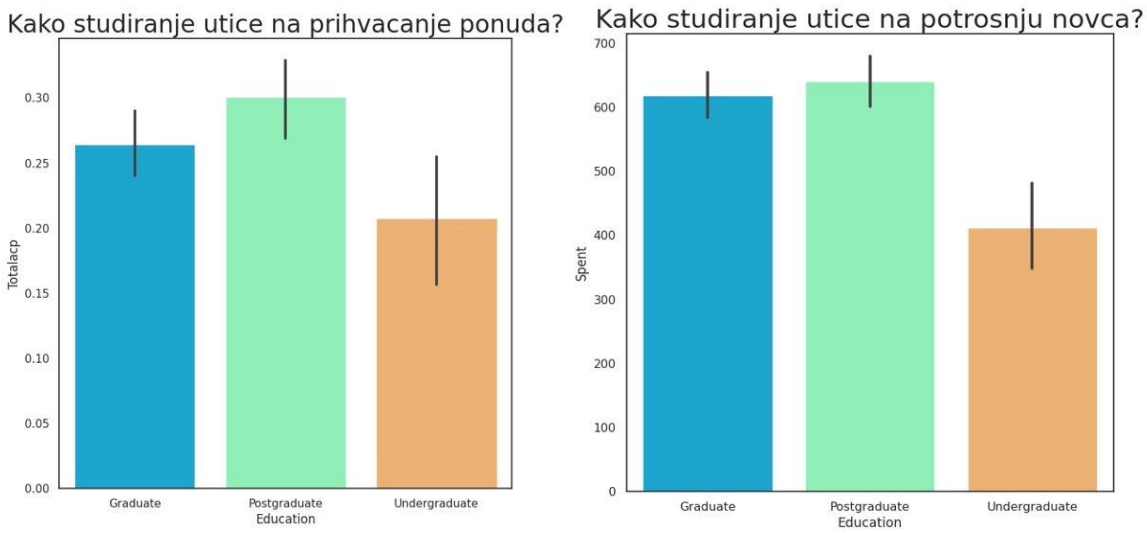
- NumWebPurchases:
- NumCatalogPurchases:
- NumStorePurchases:
- NumWebVisitsMonth:

[4]:

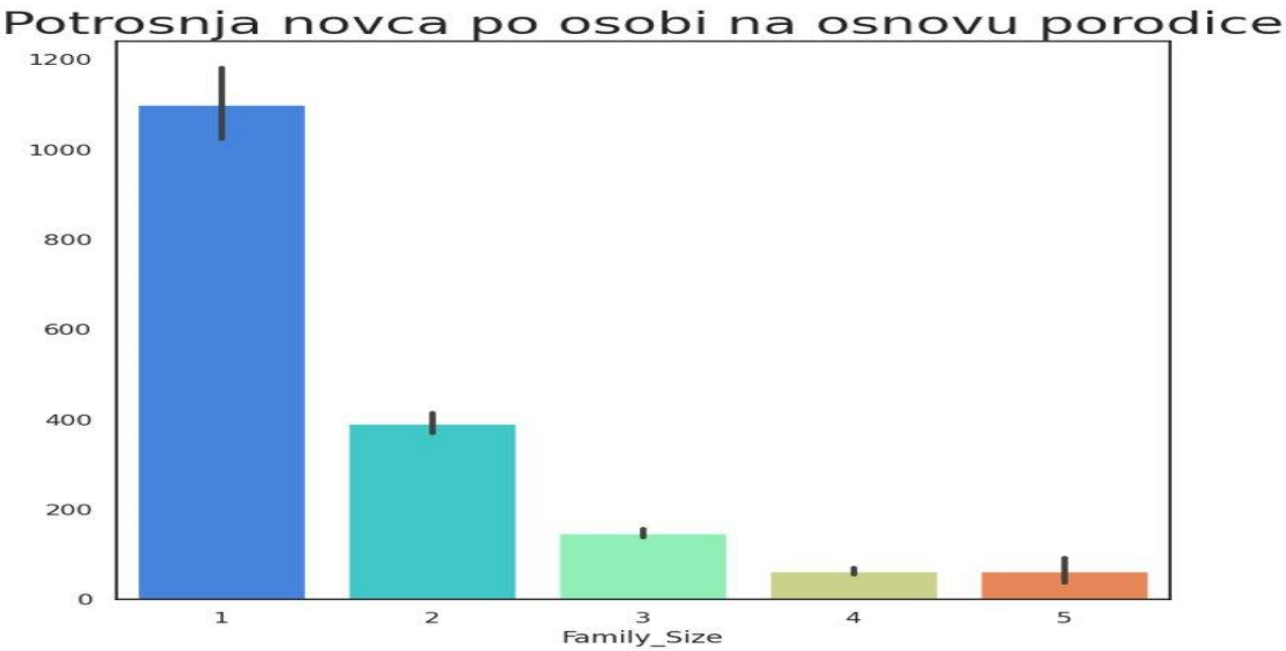
	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	AcceptedC
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	58	635	...	7	0	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	38	11	...	5	0	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	26	426	...	4	0	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	26	11	...	6	0	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	94	173	...	5	0	

5 rows × 29 columns

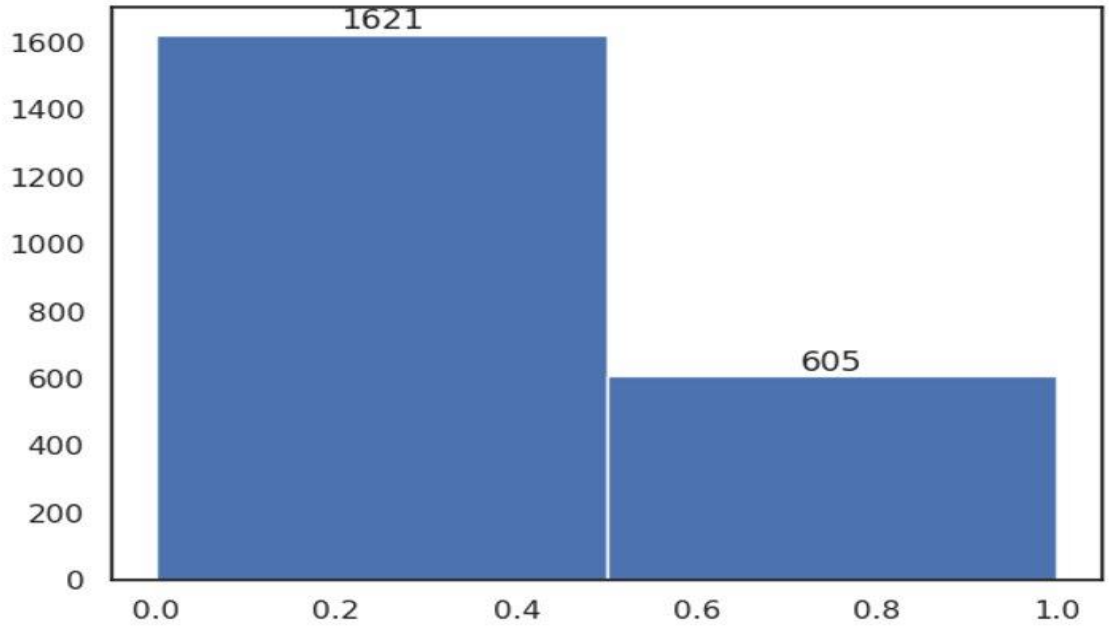
Ilustrovan prikaz nekih od atributa.



Na slikama iznad vidimo da su grafici skoro identicni, i da studenti koji nisu jos zavrsili osnovne studije pretezno malo trose a samim tim sansa da prihvate neku od ponuda iz promocije je dosta manja u odnostu na ljude koji su diplomirali ili su na visim studijama. U prevodu ako covek voli da trosi pare sanse su da prihvati ponudu iz promocije vece.



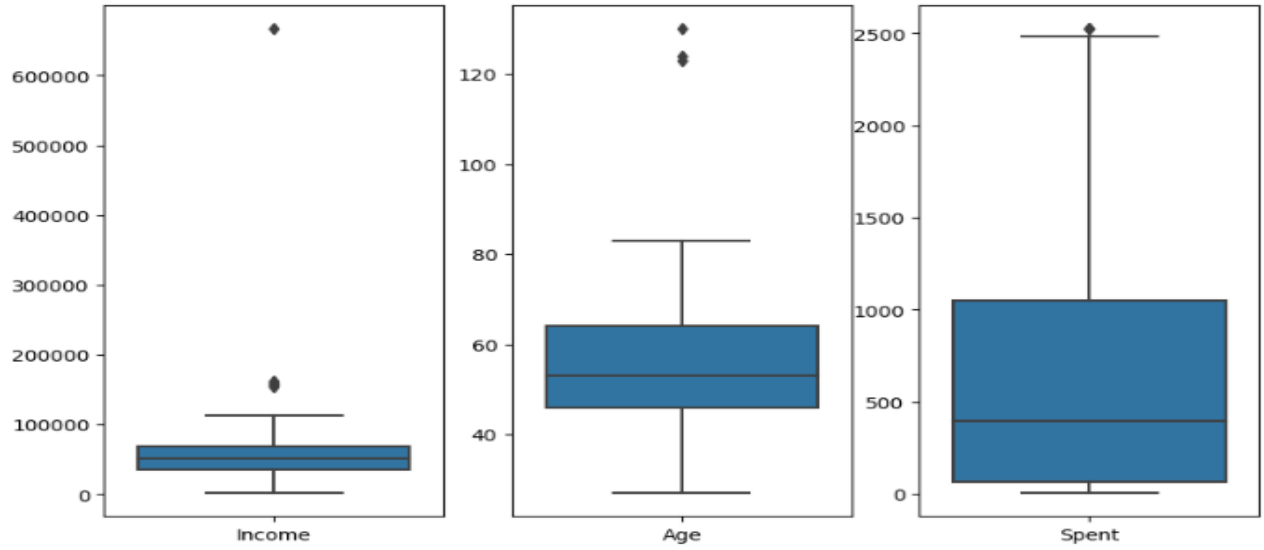
Kad vec pricamo o potrsnji novca na slici iznad vidimo koliko covek trosi na osnovu porodice u kojoj zivi. Vidimo da najvise trose osobe koje su same dok porodice sa vise clanova 3-5 dosta manje jer imaju decu, a deca nemaju prihode. Samim tim zakljucujemo da ce najpre osoba koja zivi sama prihvatiti neku ponudu iz promocije.



Na ovoj slici mozemo videti broj ljudi grupisan po tome da li su ikada prihvatili neko od ponuda iz promocije. Oko 75% ljudi nije nikada prihvatilo ponudu dok 605 jeste.

3. Priprema podataka za dalju obradu.

U daljim koracima vrsimo pretprocesiranje podataka. U prvom koraku prvo proveravamo da li postoje neke nedostajuce vrednosto (NULL). Izvrsavanjem komande *data.isna().any()* dobijamo da atribut “Income” sadrzi NULL vrednosti. Posto ih ima samo 14 komada mozemo obrisati te instance jer nece umati uticaj kasnije. Dalje sto nam sledi da proverimo je da li mozda postoje neki elementi koji su van gradnice.



Na slici vidimo da postoje takvi elementi I njih cemo takodje ukloniti. Sledece sto radimo je da izmenimo malo atribute npr imamo atribute “Kidhome” i “Teenhome” koji su jako slicni i ima smisla napraviti novi atribut “Children” koji ce obuhvatati predhodna dva atributa. Takodje pravimo atribut “Age” za godine, “Spent” koliko je neko ukupno potrosio para na proizvode, “Living_With” sa koliko ljudi zivi, “Family_Size” kolika mu je porodica, “Is_Parent” da li je roditelj, “NumTotalPurchases” sabracemo sve kupovine u jedan atribut , “Totalacp” svako prihvacanje ponude bice 1 inace 0, “Education” napravicemo 3 grupe radi lakseg snalazenja i takodje skracujemo imena proizvoda. Nakon toga brisemo ne potrebne atribute (Response','Complain',"Marital_Status", "Dt_Customer", "Z_CostContact", "Z_Revenue", "Year_Birth","ID","Kidhome","Teenhome",'AcceptedCmp1','AcceptedCmp2','AcceptedCmp3','AcceptedCmp4','AcceptedCmp5','NumWebPurchases','NumCatalogPurchases','NumStorePurchases','NumDealsPurchases'). Sada resavamo problem kategorickih promenljivih preko LabelEncodera. Pretvara kategoricke vrednosti u numericke. Potrebno je podeliti podatke na ostale i na ciljnu promenljivu u nasem slucaju je to atribut “Totalacp”. Da bi mogli da radimo klasifikaciju neophodno je da podatke podelimo na trening i test skup zatim da odradimo normalizaciju podataka koristeci funkcije MinMaxScaler time dobijamo da nas skup podataka nece imati

elemente van granica. Sada je sve spremno za klasifikaciju podatke cemo cuvati preko funkcije dump iz biblioteke joblib. Kod podataka za klasterovanje nije potrebno deljenje na trening i test vec samo da ih skaliramo zatim cuvamo kao kod klasifikacije.

4. Klasifikacija

Za pocetak krenucemo sa Decision Tree algoritnom. Radi tako što pravi drvo sačinjeno od niza odluka ili uslova kako bi se donela konačna predikcija. Osnovni koraci kod algoritma su : **Izbor atributa za podelu, Podela skupa podataka, Rekurzivni proces, Listovi drveta, Predviđanje. Nakon izgradnje stabla,** primeri se klasifikuju putujući od korena do listova, gde svaki list predstavlja konačnu klasifikaciju primera.

```
Train data:

Confusion matrix:
 [[1210    1]
 [   3  457]]
Accuracy score:  0.997606223818073
Precision score:  0.9978165938864629
Recall score:  0.9934782608695653
F1 score:  0.9956427015250545

--------------------------------------

Test data:

Confusion matrix:
 [[337  73]
 [ 64  84]]
Accuracy score:  0.7544802867383512
Precision score:  0.535031847133758
Recall score:  0.5675675675675675
F1 score:  0.5508196721311476
```

Na slici vidimo da je preciznost na trening skupu ~0.99 dok je na test skupu ~0.75 samim tim je doslo do preprilagodjavanja i potrebno je podesiti hiperparametre. Za podesavanje koristimo GridSearchCV koji ce odabrati najbolje rezultate na osnovu unakrsne validacije.

```
Train data:

Confusion matrix:
 [[1188   23]
 [ 276  184]]
Accuracy score:  0.8210652304009575
Precision score:  0.8888888888888888
Recall score:  0.4
F1 score:  0.5517241379310346

----------------------------------------

Test data:

Confusion matrix:
 [[389  21]
 [102  46]]
Accuracy score:  0.7795698924731183
Precision score:  0.6865671641791045
Recall score:  0.3108108108108108
F1 score:  0.42790697674418604
```

Rezultati govore da nam unakrsna validacija nije pomogla pa idemo na naredni algoritam, a to je isto Decision Tree ali sa parametrom class_weight='balanced'.

```
Train data:

Confusion matrix:
 [[1207    4]
 [   0  460]]
Accuracy score:  0.997606223818073
Precision score:  0.9913793103448276
Recall score:  1.0
F1 score:  0.9956709956709957

----------------------------------------

Test data:

Confusion matrix:
 [[339  71]
 [ 76  72]]
Accuracy score:  0.7365591397849462
Precision score:  0.5034965034965035
Recall score:  0.4864864864864865
F1 score:  0.49484536082474234
```

Ovaj algoritam je jos slabiji u odnosu na pocetni vidimo da je preciznost na test skupu ovde ~0.73.

Sledeci algoritam je Random Forest. Random Forest (Slučajna šuma) je ensemble algoritam koji se bazira na konceptu Decision Tree algoritma. Ensemble algoritmi kombinuju više modela kako bi se poboljšala sposobnost generalizacije i smanjio efekat preprilagođavanja (overfitting). Random Forest je jedan od najpopularnijih ensemble algoritama i koristi se za klasifikaciju.

```
Train data:

Confusion matrix:
 [[1210    1]
 [   3  457]]
Accuracy score:  0.997606223818073
Precision score:  0.9978165938864629
Recall score:  0.9934782608695653
F1 score:  0.9956427015250545

---------------------------------------

Test data:

Confusion matrix:
 [[379  31]
 [ 78  70]]
Accuracy score:  0.8046594982078853
Precision score:  0.693069306930693
Recall score:  0.47297297297297
F1 score:  0.5622489959839357
```

Na slici iznad vidimo da je takodje doslo do preprilagodjavanja jer nam je preciznost na trening skupu gotovo ista kao kod Decision Tree, a test je ~0.8 sto je dosta bolje od ranijih algoritma. Pokusacemo sa unakrsnom validacijom.

```
Train data:

Confusion matrix:
 [[1209    2]
 [   4  456]]
Accuracy score:  0.9964093357271095
Precision score:  0.9956331877729258
Recall score:  0.991304347826087
F1 score:  0.9934640522875816

---------------------------------------

Test data:

Confusion matrix:
 [[378  32]
 [ 75  73]]
Accuracy score:  0.8082437275985663
Precision score:  0.6952380952380952
Recall score:  0.49324324324324326
F1 score:  0.5770750988142292
```

Vidimo da nam unakrsna validacija nije nista pomogla I da smo skoro dobili identicne rezultate kao i Random Forest.

Pokusacemo sada sa GradientBoosting. Gradient Boosting je još jedan moćan ensemble algoritam u mašinskom učenju, koji takođe spada u kategoriju boosting algoritama. Ovaj algoritam se koristi za rešavanje problema klasifikacije, regresije i drugih problema predviđanja, a često daje veoma dobre rezultate i sa složenim skupovima podataka.

```
Train data:

Confusion matrix:
 [[1197   14]
 [ 169  291]]
Accuracy score:  0.8904847396768402
Precision score:  0.9540983606557377
Recall score:  0.6326086956521739
F1 score:  0.7607843137254902

--------------------------------------

Test data:

Confusion matrix:
 [[379  31]
 [ 74  74]]
Accuracy score:  0.8118279569892473
Precision score:  0.7047619047619048
Recall score:  0.5
F1 score:  0.5849802371541502
```

Ovde vidimo prvi put da nije doslo do preprilagodjivanja i da je preciznost ~0.89, a na test skupu ~0.81. Probacemo sa hiperparametrima

```
Train data:

Confusion matrix:
 [[1209    2]
 [ 101  359]]
Accuracy score:  0.93836026331538
Precision score:  0.9944598337950139
Recall score:  0.7804347826086957
F1 score:  0.874543239951279

--------------------------------------

Test data:

Confusion matrix:
 [[376  34]
 [ 73  75]]
Accuracy score:  0.8082437275985663
Precision score:  0.6880733944954128
Recall score:  0.5067567567567568
F1 score:  0.5836575875486382
```

Nazalost ni ovde nismo dobili bolje rezultate te se okrecemo jos jednom algoritnu.

Zadnji algoritam koji sam testirao je KNN. Osnovna ideja iza KNN algoritma je da predviđanje (klasa ili vrednost) za određeni uzorak podataka zavisi od klasa/vrednosti njegovih "k najbližih suseda" u trening skupu.

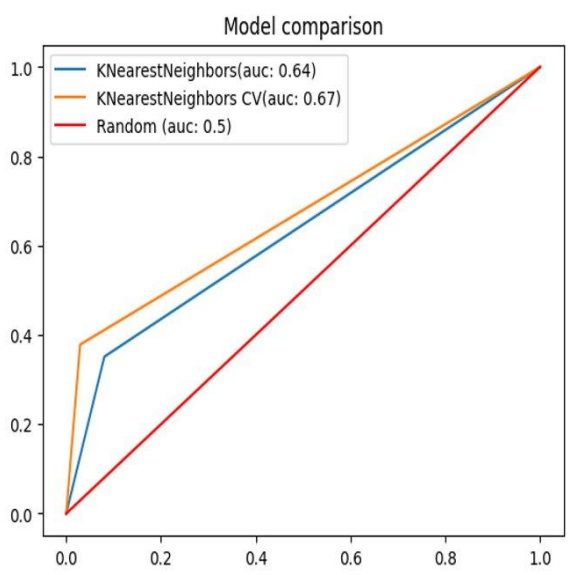
```
Train data:

Confusion matrix:
 [[1143   68]
 [ 209  251]]
Accuracy score:  0.834230999401556
Precision score:  0.786833855799373
Recall score:  0.5456521739130434
F1 score:  0.644415917843389

--------------------------------------

Test data:

Confusion matrix:
 [[377  33]
 [ 96  52]]
Accuracy score:  0.7688172043010753
Precision score:  0.611764705882353
Recall score:  0.35135135135135137
F1 score:  0.4463519313304722
```

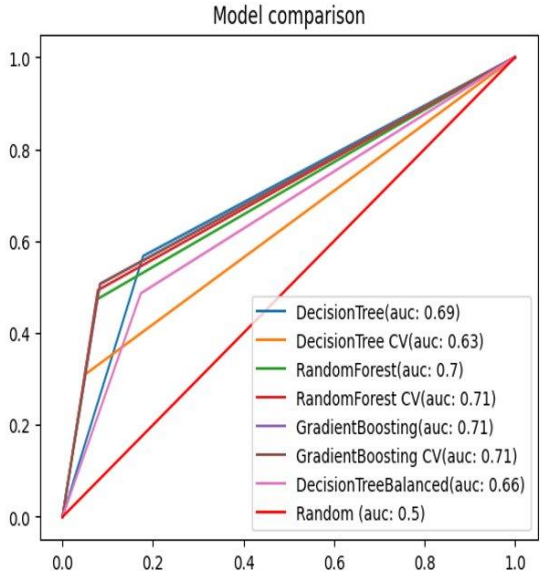
I ovde vidimo da nemamo problem preprilagodajvanja trening skup ~0.83, a test skup ~0.75. Probacemo opet sam hiperparametrima.

```
Train data:

Confusion matrix:
 [[1210    1]
 [   3  457]]
Accuracy score:  0.997606223818073
Precision score:  0.9978165938864629
Recall score:  0.9934782608695653
F1 score:  0.9956427015250545

--------------------------------------

Test data:

Confusion matrix:
 [[398  12]
 [ 92  56]]
Accuracy score:  0.8136200716845878
Precision score:  0.8235294117647058
Recall score:  0.3783783783783784
F1 score:  0.5185185185185186
```

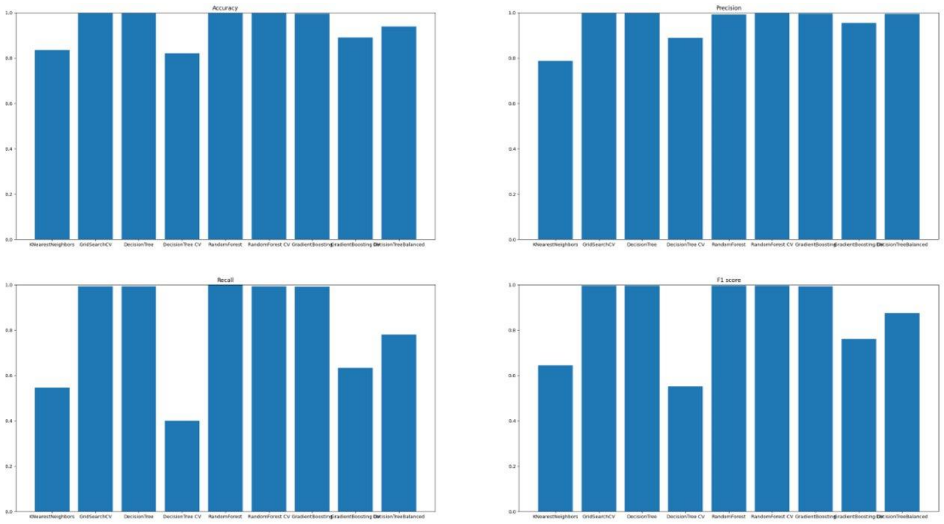
Za n=25 koje CV izabrao kao najbolji dobijamo vec pomenito preprilagodjavanje... ali i mali napredak jer nam ovde test skup dostize cak ~0.81.

Poredjenje izabranih modela. Receiver Operating Characteristic (ROC) kriva je grafički prikaz performansi klasifikacionog modela, posebno binarne klasifikacije. ROC kriva ilustruje odnos između stope lažno pozitivnih (False Positive Rate, FPR) i stope istinski pozitivnih (True Positive Rate, TPR), uz različite pragove za donošenje odluke. Idealna ROC kriva ide prema gornjem levom uglu, što znači visok TPR i nizak FPR sto kod nas nije bas slucaj.

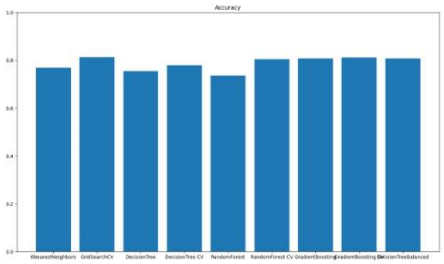


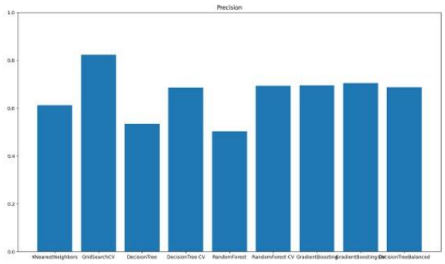


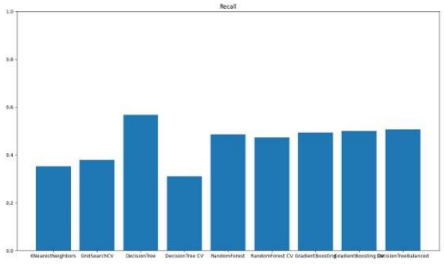
Sada mozemo da uporedimo i razlicitu metriku izmedju svih algoritama, a to je tacnost, preciznost, odziv i f1 score. Na trening skupu vidimo da jedino KNN i Decision Tree CV daju lose rezultate dok su ostali dobri. To nam je manje bitno, vec nas zanima sta kazu podaci za test skup.

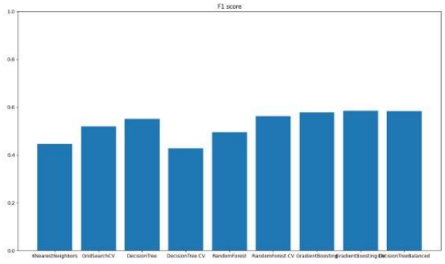


Na test skupu vidimo da se ne izdvaja nijedan algoritam I da sku skoro svi dali lose rezultate pa cak i kada sam uveo unakrsnu validaciju


Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
KNearestNeighbors
GridSearchCV
DecisionTree
DecisionTree CV
RandomForest
RandomForest CV
GradientBoosting
GradientBoosting
DecisionTreeBalanced


Precision
1.0
0.8
0.6
0.4
0.2
0.0
KNearestNeighbors
GridSearchCV
DecisionTree
DecisionTree CV
RandomForest
RandomForest CV
GradientBoosting
GradientBoosting
DecisionTreeBalanced


Recall
1.0
0.8
0.6
0.4
0.2
0.0
KNearestNeighbors
GridSearchCV
DecisionTree
DecisionTree CV
RandomForest
RandomForest CV
GradientBoosting
GradientBoosting
DecisionTreeBalanced

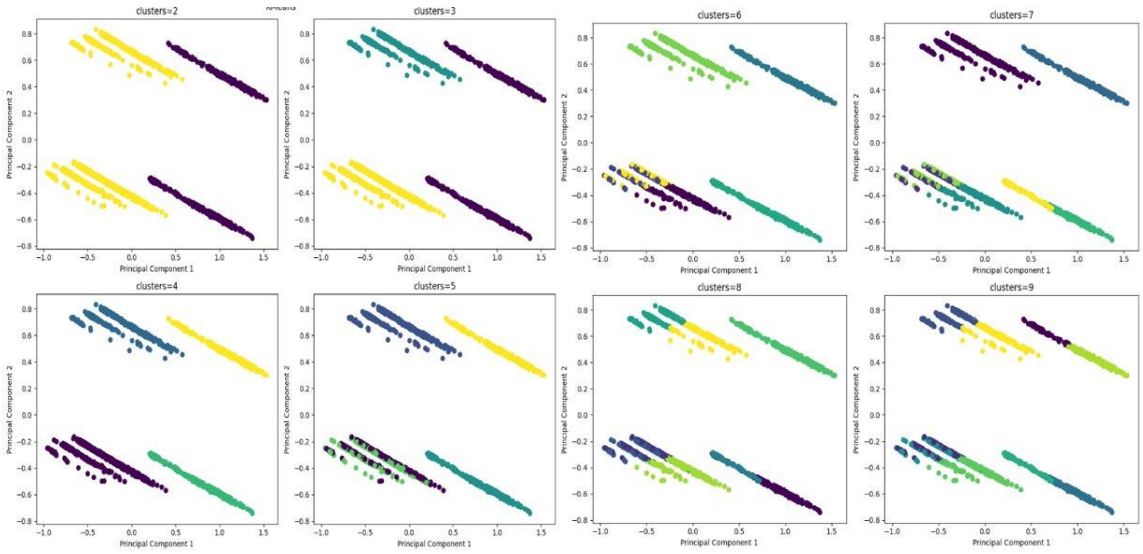

F1 score
1.0
0.8
0.6
0.4
0.2
0.0
KNearestNeighbors
GridSearchCV
DecisionTree
DecisionTree CV
RandomForest
RandomForest CV
GradientBoosting
GradientBoosting
DecisionTreeBalanced

5. Klasterovanje

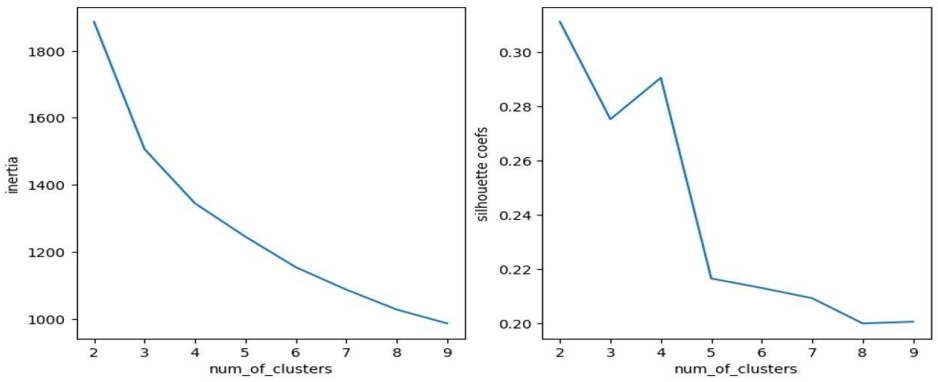
K-means je algoritam za klasterovanje, tj. grupisanje sličnih instanci podataka u različite grupe ili klase. Ovaj algoritam pripada kategoriji nenadgledanog učenja, gde model ne koristi oznake klasa prilikom učenja, već se trudi da grupiše podatke na osnovu sličnosti.

Osnovna ideja K-means algoritma je da grupiše podatke tako da minimizuje varijabilnost unutar klastera i maksimizuje varijabilnost između klastera.

Najpre cemo nase podatke smestiti u 2D uz pomoc PCA radi lakse vizualizacije algoritma.



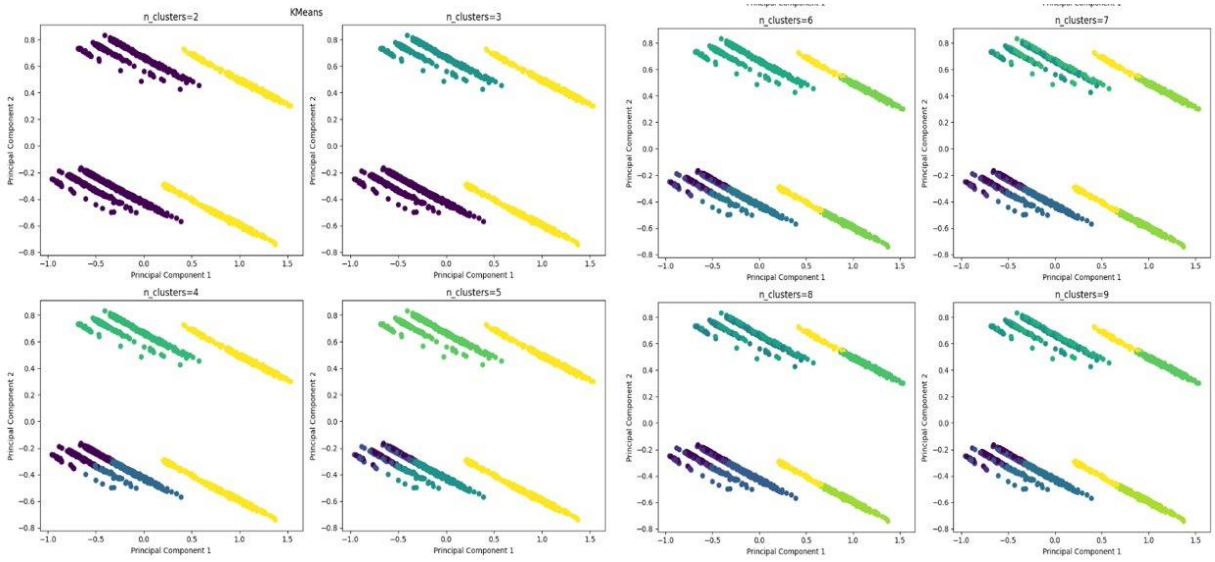
Na slici iznad vidimo izvrsavanje kmeans algoritma za vrednosti k=10



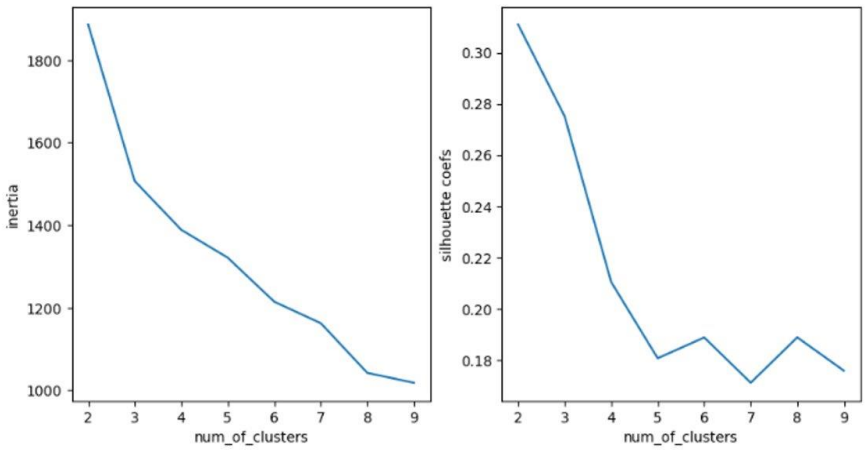
Dok na ovoj slici ovde sa leve strane imamo inerciju po klasterima odnosno sa desne strane silhouette_score.

Vidimo da je silhouette_score najbolji pri pocetku rada algortima za vrednost k=2.

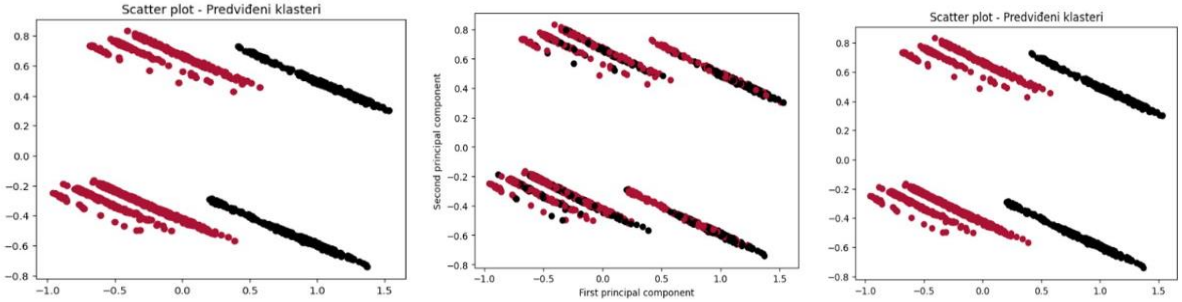
Bisecting K-means je varijacija K-means algoritma koja se koristi za klasterovanje podataka. Ovaj algoritam radi tako što iterativno deli postojeće klasterove na dva manja klastera sve dok se ne postigne željeni broj klastera. Umesto da pravi klaster po klaster, Bisecting K-means deli postojeći klaster koji ima najveću varijabilnost ili grešku.



Na slici iznad vidimo izvrsavanje bisecting kmeans algoritma za vrednosti k=10



Ovde vidimo isto da klaster broj 2 je takodje najbolji, a sve losije rezultate za vise klastera.



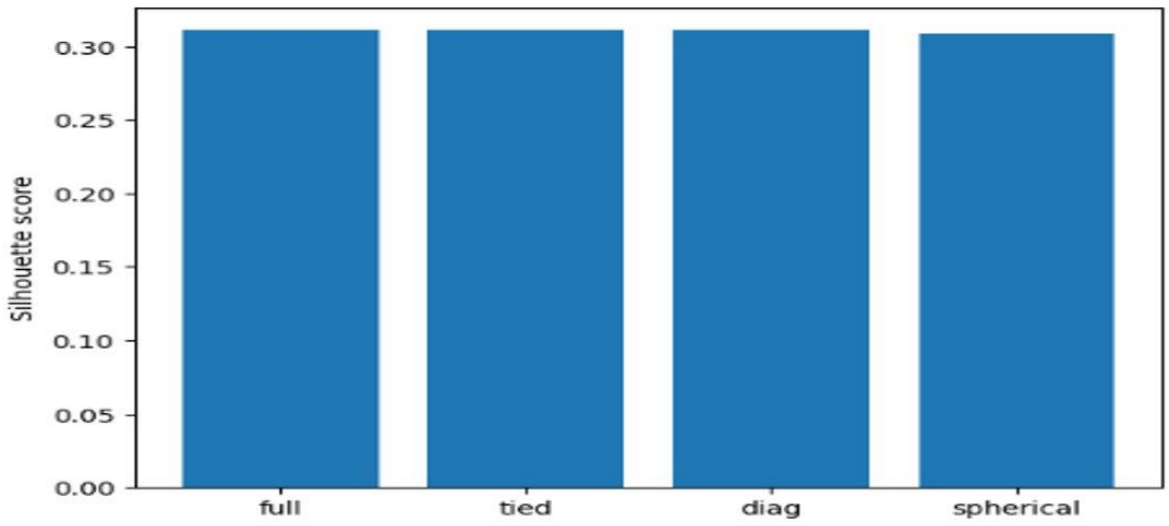
Na slici broj 1 i 3 mozemo videti rad algoritma za k=2 i primecujemo da su rezultati identicni stoga mozemo slobodno odbaciti Bisecting kmeans, dok na slici broj 2 su nasi podaci koji su rasoredjeni i vidimo da bas nisu povezi vec su razbacani.

Gaussian Mixture Model (GMM) je statistički model koji se koristi za modeliranje verovatnoćne raspodele podataka. GMM se često koristi u klasterskoj analizi, ali i za modeliranje verovatnoćne raspodele kontinualnih podataka.

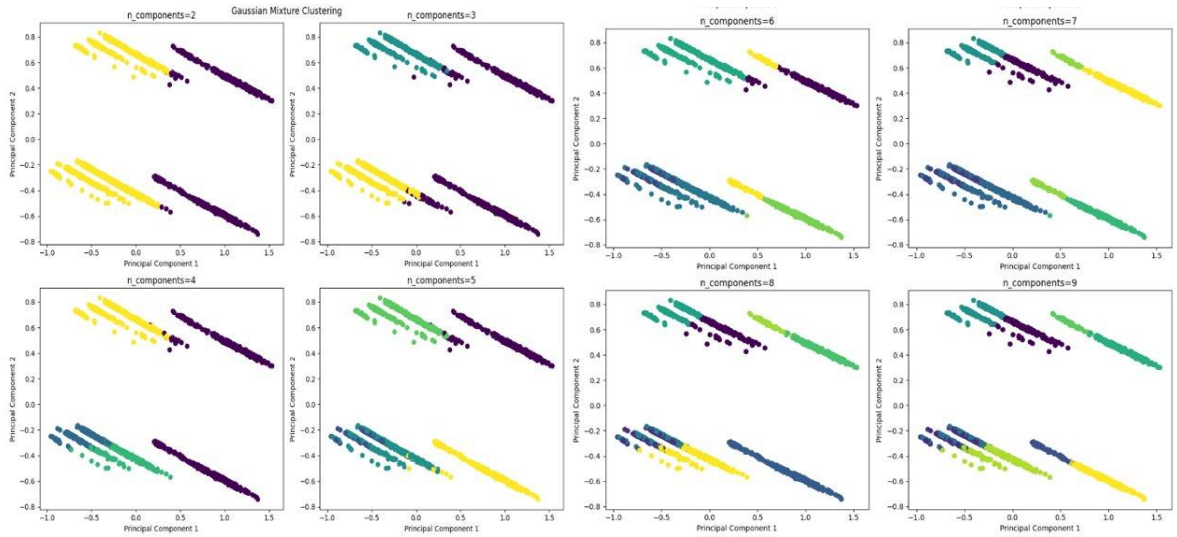
Osnovna ideja iza Gaussian Mixture Modela je da podaci potiču iz kombinacije više Gausovih raspodela (normalnih raspodela). Svaka Gausova komponenta predstavlja jedan klaster u podacima. GMM se sastoji od nekoliko komponenata, a svaka komponenta opisuje normalnu raspodelu sa određenim parametrima: srednja vrednost (mean) i standardna devijacija (variance).

Postoje 4 razlicite vrste kovarijanse koje se mogu koristiti:

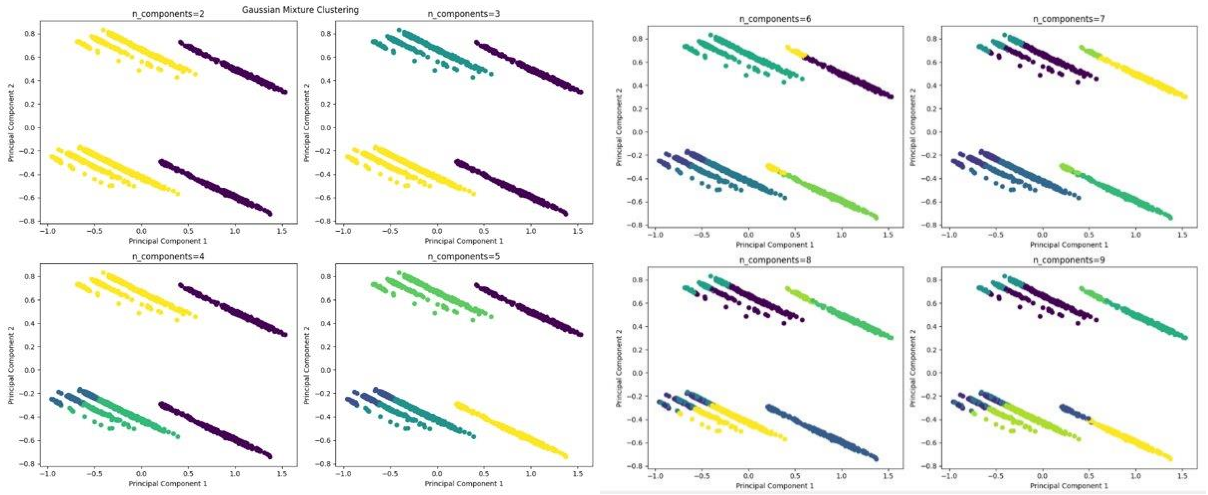
- 'full': svaka komponenta ima svoju opštu matricu
- 'tied': sve komponente dele istu opštu matricu kovarijanse
- 'diag': svaka komponenta ima svoju dijagonalnu matricu kovarijanse
- 'spherical': svaka komponenta ima svoju pojedinačnu varijansu.



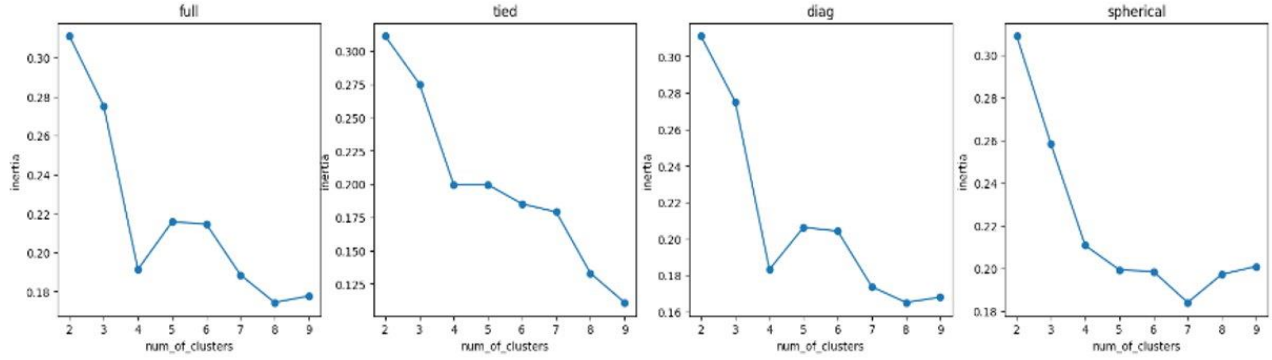
Na ovoj slici vidimo da su skoro sve 4 identicne jedino spherical malo odudara od ostalih.



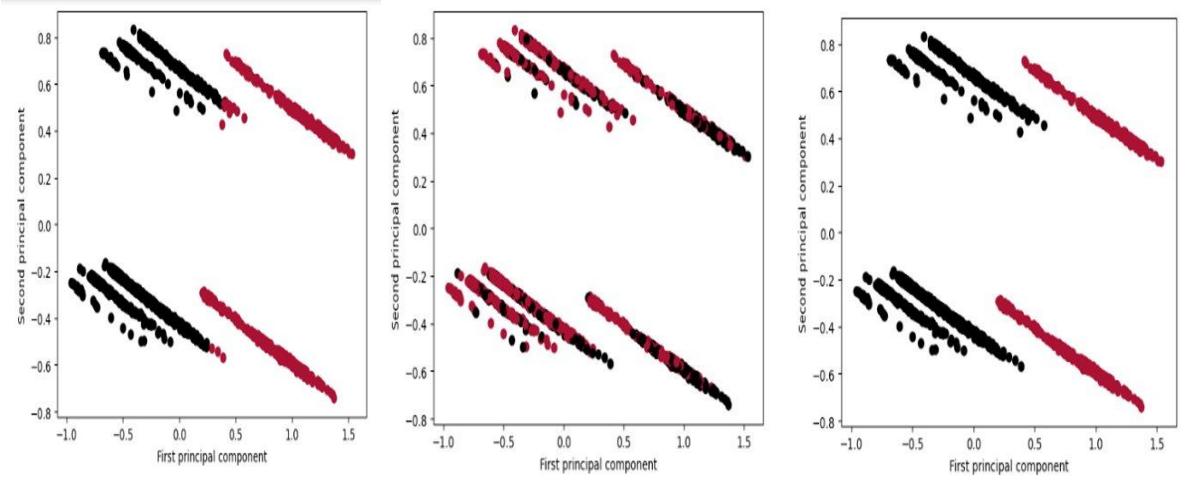
Ovde vidimo princip rada GMM algoritma sphrecal varijanse za vrednosti k=10



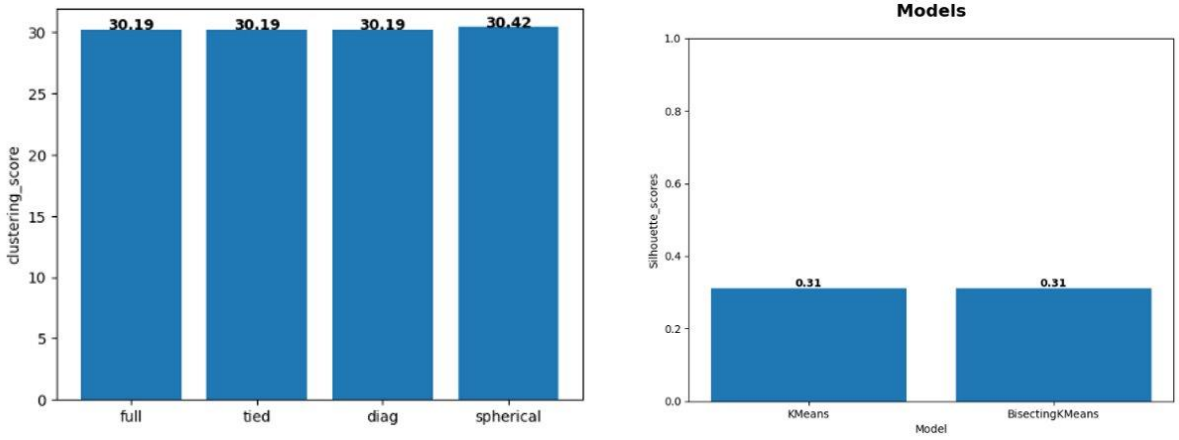
Ovde vidimo princip rada GMM algoritma tied full i diag varijanse za vrednosti k=10



Na ovoj slici vidimo opet da je najbolje da se izabere kada je k=2 za klaster jer je tada najveca vrednost siluete.



Na slici 1 sphrecal varijansa i na slici 3 ostale 3 varijanse dok su nasi podaci na slici 2.

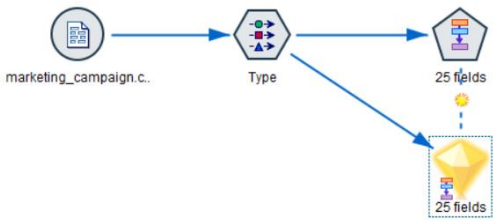


Na kraju ako uporedimo rezultate koje smo dobili vidimo da su full tied diag stvarno jednaki, da je spherical malo bolji od njih i su kmeans ili bisecing kmeans najbolji ali imaju los siluet score.

Ostaje da uporedimo procenat preklapanja a on je ~70% za kmeans i ~30% za GMM varijanse, te je kmeans(bisecing kmeans) najbolji iako je imao maltene isti silutni score kao i GMM.

6. Pravila pridruzivanja

Association rules:



Neke od zanimljivih pravila dobijenih association rules

Most Interesting Rules by Lift

Rank	Rule ID	Condition	Prediction	Sorted By Lift	Other Evaluation Statistics			
					Condition Support (%)	Confidence (%)	Rule Support (%)	Deployability (%)
1	513	299 ≤ MntWines < 597 MntFruits ≤ 40 MntMeatProducts ≤ 345 4 ≤ NumWebVisitsMonth < 8 AcceptedCmp1 ≤ 0	5 ≤ NumWebPurchases < 11	3.36	5.85	87.02	5.09	0.76
2	514	299 ≤ MntWines < 597 MntMeatProducts ≤ 345 MntSweetProducts ≤ 53 4 ≤ NumWebVisitsMonth < 8 AcceptedCmp3 ≤ 0	5 ≤ NumWebPurchases < 11	3.36	5.85	87.02	5.09	0.76
3	515	299 ≤ MntWines < 597 MntMeatProducts ≤ 345 MntSweetProducts ≤ 53 4 ≤ NumWebVisitsMonth < 8 AcceptedCmp1 ≤ 0	5 ≤ NumWebPurchases < 11	3.36	6.16	86.96	5.36	0.80
4	519	299 ≤ MntWines < 597 MntFruits ≤ 40 MntMeatProducts ≤ 345 4 ≤ NumWebVisitsMonth < 8	5 ≤ NumWebPurchases < 11	3.35	6.12	86.86	5.31	0.80
5	521	299 ≤ MntWines < 597 MntFruits ≤ 40 MntMeatProducts ≤ 345 4 ≤ NumWebVisitsMonth < 8 AcceptedCmp5 ≤ 0	5 ≤ NumWebPurchases < 11	3.35	6.12	86.86	5.31	0.80

Neka pravila koja se nalaze u segmentu onih sortiranih po liftu

Most Interesting Rules by Rule Support

Rank	Rule ID	Condition	Prediction	Sorted By Rule Support(%)	Other Evaluation Statistics			
					Condition Support (%)	Confidence (%)	Lift	Deployability (%)
1	850	Year_Birth > 1,975 MntWines ≤ 299 MntFruits ≤ 40 4 ≤ NumWebVisitsMonth < 8 AcceptedCmp4 ≤ 0	1 ≤ Kidhome < 1	12.10	14.55	83.13	2.07	2.46
2	851	Year_Birth > 1,975 MntWines ≤ 299 MntFruits ≤ 40 4 ≤ NumWebVisitsMonth < 8 AcceptedCmp1 ≤ 0	1 ≤ Kidhome < 1	12.10	14.55	83.13	2.07	2.46
3	894	Year_Birth > 1,975 MntWines ≤ 299 MntFruits ≤ 40 4 ≤ NumWebVisitsMonth < 8	1 ≤ Kidhome < 1	12.10	14.60	82.87	2.06	2.50
4	895	Year_Birth > 1,975 MntWines ≤ 299 MntFruits ≤ 40 MntMeatProducts ≤ 345 4 ≤ NumWebVisitsMonth < 8	1 ≤ Kidhome < 1	12.10	14.60	82.87	2.06	2.50
5	896	Year_Birth > 1,975 MntWines ≤ 299 MntFruits ≤ 40 NumCatalogPurchases ≤ 6 4 ≤ NumWebVisitsMonth < 8	1 ≤ Kidhome < 1	12.10	14.60	82.87	2.06	2.50

Neka pravila koja se nalaze u segmentu onih sortiranih po podrsci

Most Interesting Rules by Confidence

Rank	Rule ID	Condition	Prediction	Sorted By Confidence(%)	Other Evaluation Statistics			
					Condition Support (%)	Rule Support (%)	Lift	Deployability (%)
1	1	1,955 ≤ Year_Birth < 1,975 Kidhome ≤ 0 3 ≤ NumDealsPurchases < 6 AcceptedCmp5 ≤ 0	1 ≤ Teenhome < 1	97.44	5.22	5.09	2.12	0.13
2	2	Kidhome ≤ 0 Teenhome ≤ 0 NumWebPurchases ≤ 5 6 ≤ NumCatalogPurchases < 11 Response ≤ 0	NumWebVisitsMonth ≤ 4	97.39	5.13	5.00	2.76	0.13
3	3	1,955 ≤ Year_Birth < 1,975 Kidhome ≤ 0 3 ≤ NumDealsPurchases < 6	1 ≤ Teenhome < 1	96.67	5.36	5.18	2.10	0.18
4	4	Kidhome ≤ 0 Teenhome ≤ 0 299 ≤ MntWines < 597 NumWebPurchases ≤ 5	NumWebVisitsMonth ≤ 4	96.67	5.36	5.18	2.74	0.18
5	5	Kidhome ≤ 0 Teenhome ≤ 0 299 ≤ MntWines < 597 NumDealsPurchases ≤ 3 NumWebPurchases ≤ 5	NumWebVisitsMonth ≤ 4	96.67	5.36	5.18	2.74	0.18

Neka pravila koja se nalaze u segmentu onih sortiranih po pouzdanosti

7. Zakljucak

Zaključak ovog projekta ukazuje na izazove i ograničenja u pokušaju predviđanja da li će neka osoba prihvatiti ponudu iz kataloga korišćenjem programiranog modela. Iako smo primenili određene tehnike mašinskog učenja i analizirali relevantne podatke, rezultati nisu ispunili očekivanja.

Ovo ukazuje na složenost prirode ljudskih odluka i ponašanja. Faktori koji utiču na donošenje odluka mogu biti mnogo dublji i komplikovaniji nego što su mogli biti obuhvaćeni u modelu. Nedostatak preciznosti može takođe proizaći iz nedovoljne količine ili kvaliteta podataka na kojima je model treniran.

Iako rezultati nisu bili zadovoljavajući, ovaj projekat pruža dragoceno iskustvo u razumevanju složenosti predviđanja ljudskih ponašanja kroz tehnologiju. Učimo da je važno prepoznati granice tehnologije i pristupiti takvim problemima s realističnim očekivanjima.

8. Literatura

https://scikit-learn.org/stable/index.html

https://github.com/MATF-istrazivanje-podataka-1/materijali_2022-2023/tree/main

https://chat.openai.com