Analiza skupa podataka
Body signal of smoking

(Projekat za predmet Istraživenje podataka 1)

Student: Ivana Nešković Asistent: Marija Erić Profesor: Nenad Mitić

Uvod

Ovaj rad prati istraživanje podataka iz skupa “Body signal of smoking” koji se mogu naći na linku:
https://www.kaggle.com/datasets/kukuroo3/body-signal-of-smoking

Sprovedena je detaljna analiza podataka, demonstrirana je primena različitih algoritama. Klasifikacija I klasterovanje su rađeni u Pythonu. Za klasifikaciju demonstrirani su algoritmi: Stabla odlučivanja I KNN(K Next Neighbours). Za klasterovanje su korišćeni algoritmi K-sredina I Gausov algoritam. Pravila pridruzivanja odrađena su u IBM SPSS Modeleru I korišćen je Apriori algoritam.

Kod ovog projekta može se naći na linku:
https://github.com/MATF-istrazivanje-podataka-1/2023_Data_Mining_Smoking_Dataset

Analiza skupa podataka

Cilj je da se na osnovu bio-signala utvrdi da li je osoba pušač.
Tačna identifikacija pušača i nepušača omogućava tačnije i preciznije proučavanje uticaja pušenja na zdravstvene ishode. Očigledno je važnije ispravno identfikovati pušače nego nepušače, jer su zdravstveni rizici povezani s pušenjem mnogo veći od rizika povezanih s nepušenjem.

Skup se sastoji od 55692 reda i narednih 27 atributa:

- `ID : indeks`
- `gender: pol`
- `age: godine`
- `height(cm): visina`
- `weight(kg): težina`
- `waist(cm) : obim struka`
- `eyesight(left): vid na levo oko`
- `eyesight(right): vid na desno oko`
- `hearing(left): sluh na levo uho`
- `hearing(right): sluh na desno uho`
- `systolic : Krvni pritisak`
- `relaxation : Krvni pritisak`
- `fasting blood sugar`
- `Cholesterol : ukupno`
- `triglyceride: trigliceridi`
- `HDL : tip holesterola`
- `LDL : tip holesterola`
- `hemoglobin`
- `Urine protein: protein u urinu`
- `serum creatinine: kreatinin u serumu`
- `AST : tip glutaminske oksalosirćetne transaminaze`
- `ALT : tip glutaminske oksalosirćetne transaminaze`
- `Gtp`
- `oral : Status usmenog ispitivanja`
- `dental caries: karijes zuba`
- `tartar : tartar status`
- `smoking: pušenje`

Da bismo se bolje upoznali sa skupom podataka, pre svega ćemo odrediti tipove atributa I broj jedinstvenih vrednosti za svaki od njih:

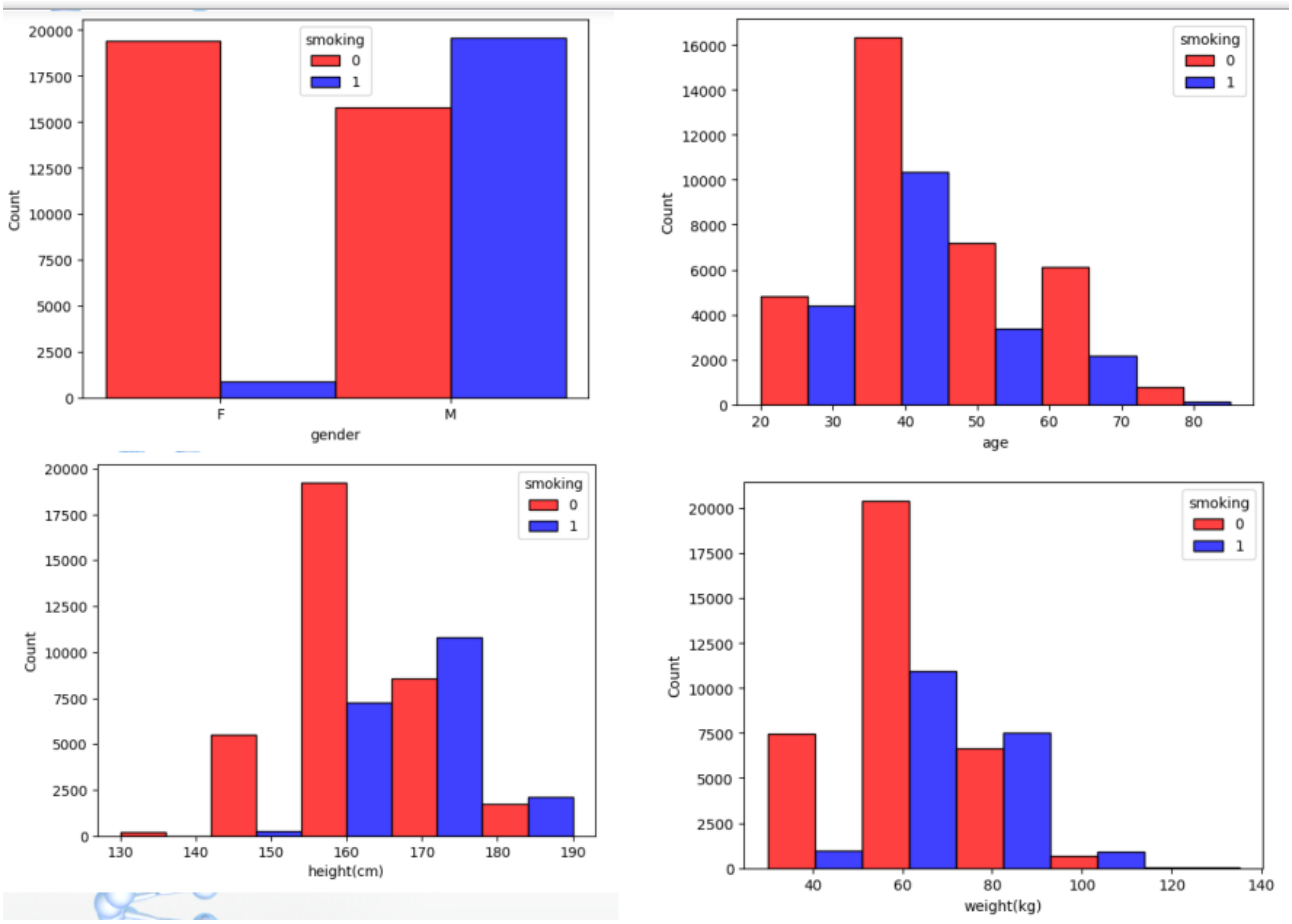
0 **ID** - int64 - ima različite vrednosti za sve instance

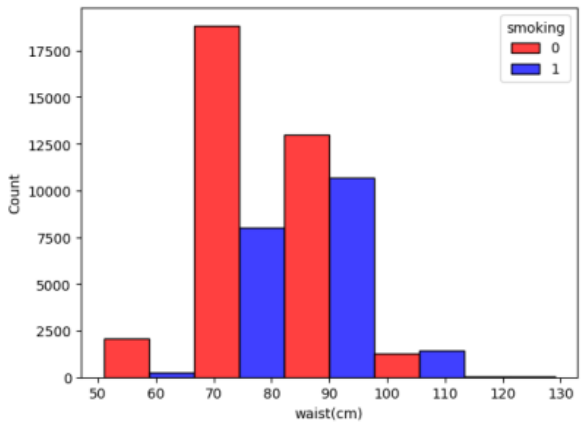
1 **gender** - object - ima 2 jedinstvenih vrednosti i to su: ['F' ,'M']
2 **age** - int64 - ima 14 jedinstvenih vrednosti i neke od njih su: [25,30,…,85]
3 **height(cm)** - int64 – 13 jedinstvenih vrednosti i neke od njih su: [135,140,…,185]
4 **weight(kg)** - int64 - ima 22 jedinstvenih vrednosti i neke od njih su: [30,35,…,135]
5 **waist(cm)** – float64 - ima 19 jedinstvenih vrednosti i to su: [1.2 0.8 1.5 1. 0.7 0.9 0.3 0.2 0.1 0.6 0.4 0.5 2. 9.9 1.3 1.6 1.9 1.1 1.8]
6 **eyesight(left)** - float64 - ima 17 jedinstvenih vrednosti i to su: [1. 0.6 0.8 1.5 1 .2 0.7 0.4 0.9 0.3 0.1 0.5 2. 9.9 0.2 1.3 1.6 1.1]
7 **eyesight(right)** -float64 - ima 17 jedinstvenih vrednosti i to su: [1. 0.6 0.8 1.5 1.2 0.7 0.4 0.9 0.3 0.1 0.5 2. 9.9 0.2 1.3 1.6 1.1]
8 **hearing(left)** - float64 - ima 2 jedinstvenih vrednosti i to su: [1. 2.]
9 **hearing(right)** – float64 - ima 2 jedinstvenih vrednosti i to su: [1. 2.]
10 **systolic** – float64 -ima 130 jedinstvenih vrednosti i neke od njih su: [114. 119. 138. 100. 120. 128. 116. 153. 115. 113. 130. 124. 95. 122. 102. 112. 126. 131. 127. 140. ...]
11 **relaxation** – float64 - ima 95 jedinstvenih vrednosti i neke od njih su:[73. 70. 86. 60. 74. 76. 82. 96. 64. 77. 78. 88. 52. 84. 90. 81. 80. 75. 83. ...]
12 **fasting blood sugar** – float64 - ima 276 jedinstvenih vrednosti i neke od njih su:[94. 130. 89. 96. 80. 95. 158. 86. 100. 101. 112. 81. 133. 72. 79. ...]
13 **Cholesterol** – float64 - ima 286 jedinstvenih vrednosti i neke od njih su: [215. 192. 242. 322. 184. 217. 226. 222. 210. 198. 178. 155. 269. 228. 227. 129. 191. 166. 186. 179. …]
14 **triglyceride** –float64 -ima 390 jedinstvenih vrednosti i neke od njih su su: [82. 115. 182. 254. 74. 199. 68. 269. 66. 147. 141. 197. 210. 47. …]
15 **HDL** – float64 - ima 126 jedinstvenih vrednosti i neke od njih su: [73. 42. 55. 45. 62. 48. 34. 43. 82. 39. 59. 88. 77. 71. …]
16 **LDL** – float64 - ima 289 jedinstvenih vrednosti i neke od njih su: [1.26e+02 1.27e+02 1.51e+02 2.26e+02 1.07e+02 1.29e+02 1.57e+02 1.34e+02 1.49e+02 7.30e+01 1.06e+02 7.70e+01 5.70e+01 …]
17 **hemoglobin** – float64 - ima 145 jedinstvenih vrednosti i neke od njih su: [12.9 12.7 15.8 14.7 12.5 16.2 17. 15. 13.7 16. 17.9 14.5 12.6 …]
18 **Urine protein** – float64 - ima 6 jedinstvenih vrednosti i to su: [1. 3. 2. 4. 5. 6.]
19 **serum creatinine**- float64 - ima 38 jedinstvenih vrednosti i to su: [0.7 0.6 1. 1.2 1.3 0.8 1.1 0.9 0.5 0.4 1.5 1.4 1.6 1.8 0.1 3. 1.9 10.3 5. 1.7 2. 3.3 0.3 7.5 7.4 2.6 2.5 0.2 9.9 2.2 2.1 6.4 3.2 3.4 11.6 2.3 5.9 10.]
20 **AST** – float64 - ima 219 jedinstvenih vrednosti i neke od njih su: [18. 22. 21. 19. 16. 38. 31. 26. 35. 34. 13. 20. 15. 37. 23. 17. 29. 42. …]
21 **ALT** -float64 - ima 245 jedinstvenih vrednosti i neke od njih su: [1.900e+01 1.600e+01 2.600e+01 1.400e+01 2.700e+01 7.100e+01 3.100e+01 2.400e+01 4.600e+01 6.900e+01 9.000e+00 …]

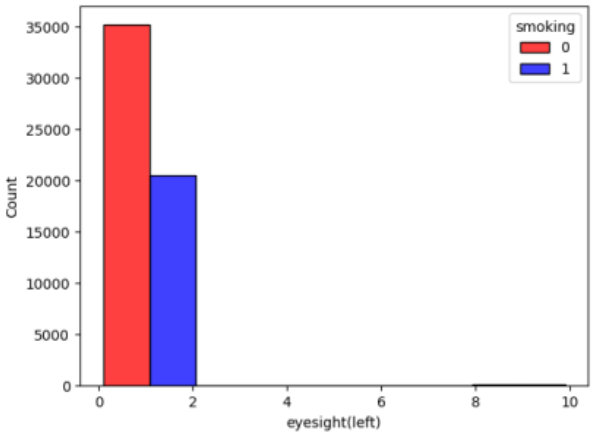
22 **Gtp** -float64 - ima 488 jedinstvenih vrednosti i neke od njih su: [27. 18.
22. 33. 39. 111. 14. 63. 37. 64. 83. 9. 19. 16. 25. …]
23 **oral** - object - ima 1 jedinstvenih vrednosti i to su: ['Y']
24 **dental caries** -int64 - ima 2 jedinstvenih vrednosti i to su: [0 1]
25 **tartar** - object - ima 2 jedinstvenih vrednosti i to su: ['Y' 'N']
26 **smoking** - int64 - ima 2 jedinstvenih vrednosti i to su: [0 1]

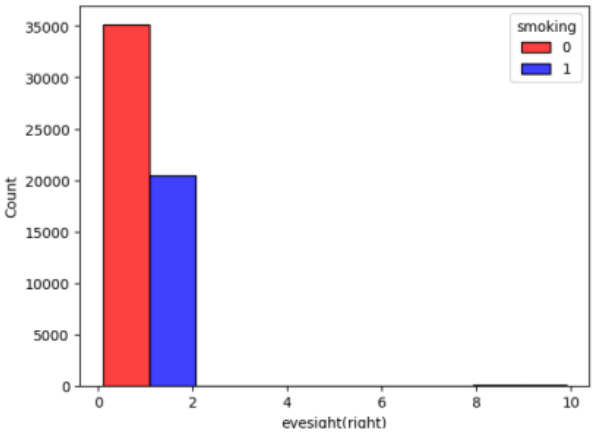
Atribut oral ima iste vrednosti za sve instance i neće nam dati nikakve korisne informacije pa ćemo ga izbaciti iz skupa. Slicno, atribut ID ima različite vrednosti za sve instance pa i njega izbacujemo iz skupa.

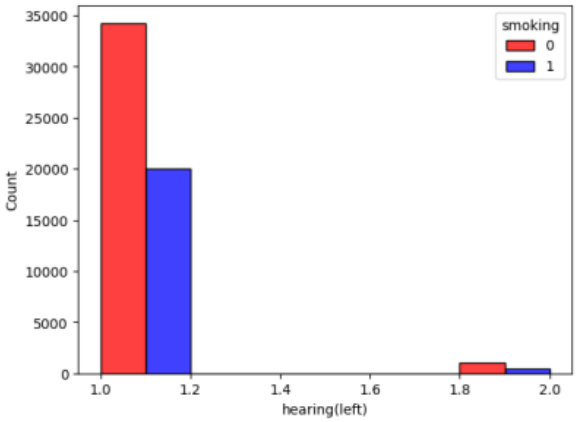
Naš ciljni atribut je smoking. On ima dve vrednosti: 0 označava nepušača, a 1 označava pušača, što znači da rešavamo problem binarne klasifikacije.
Sada ćemo pogledati kako pojedinačni ulazni atributi utiču na vrednost ciljne promenljive:

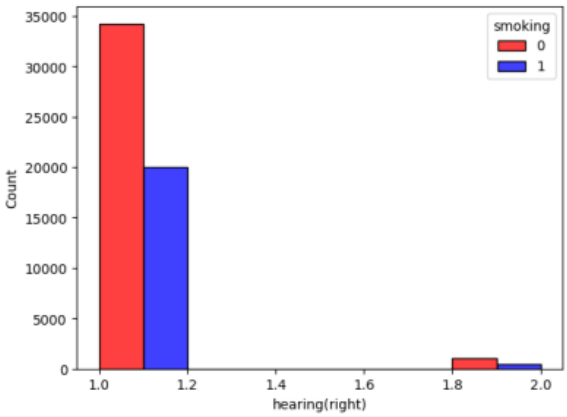


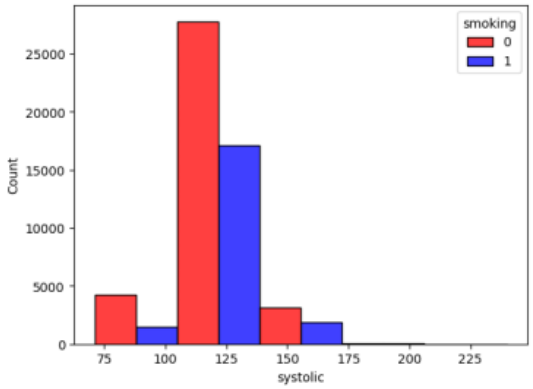

smoking
0
1
Count
waist(cm)

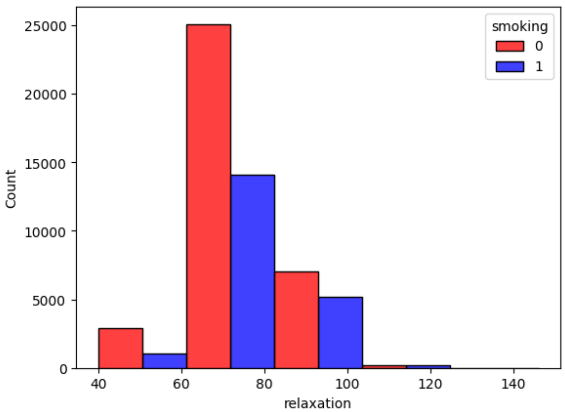

smoking
0
1
Count
eyesight(left)

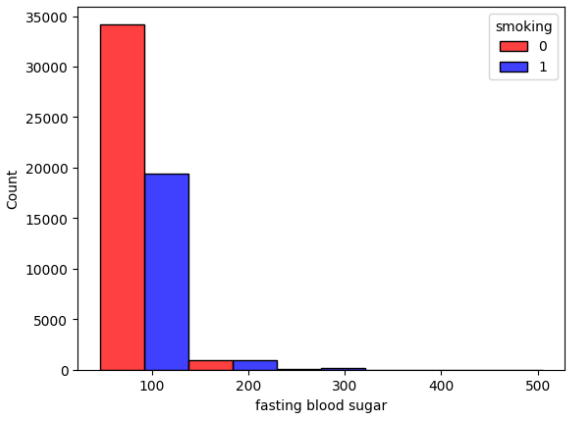

smoking
0
1
Count
eyesight(right)

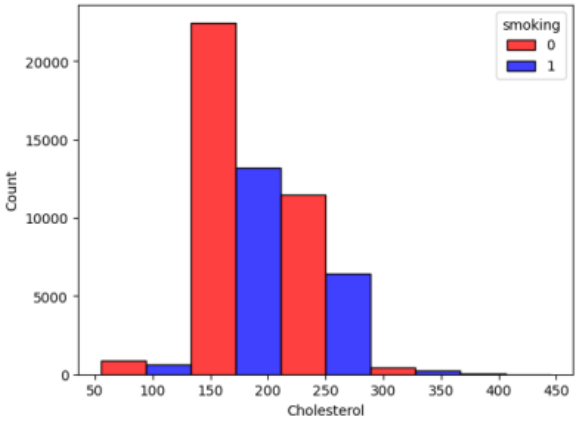

smoking
0
1
Count
hearing(left)

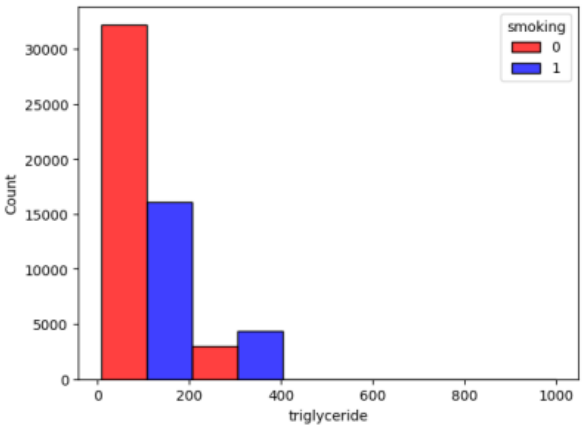

smoking
0
1
Count
hearing(right)

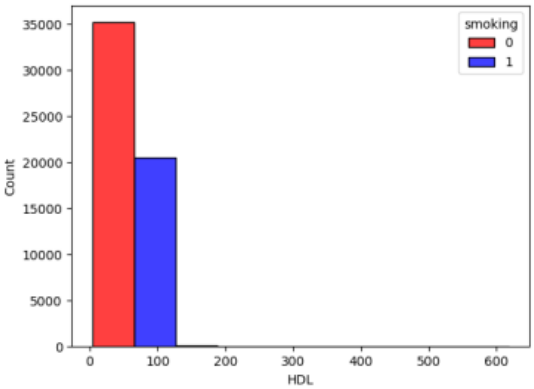

smoking
0
1
Count
systolic

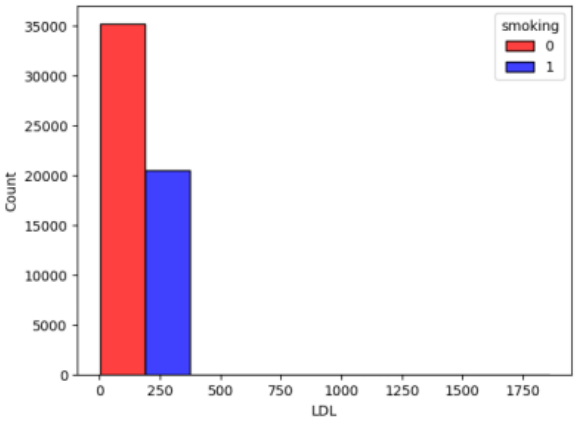

smoking
0
1
Count
relaxation

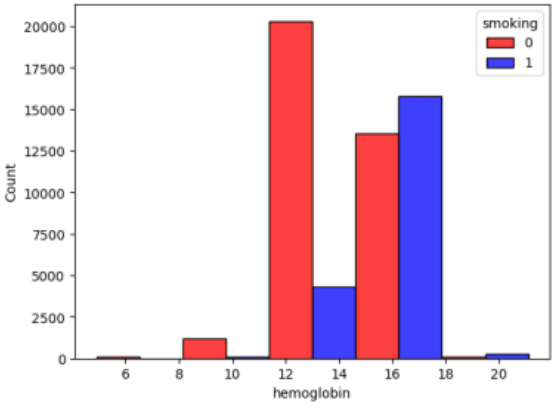

smoking
0
1
Count
fasting blood sugar

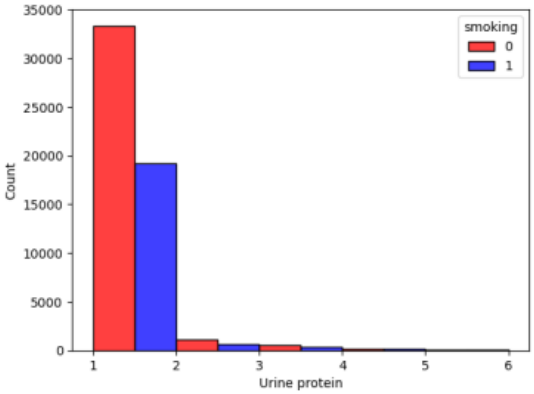

smoking
0
1
Count
Cholesterol

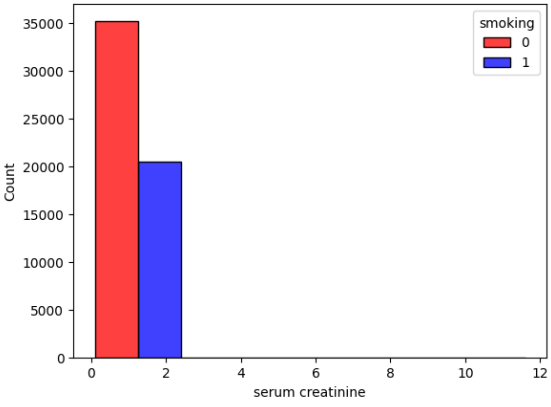

smoking
0
1
Count
triglyceride

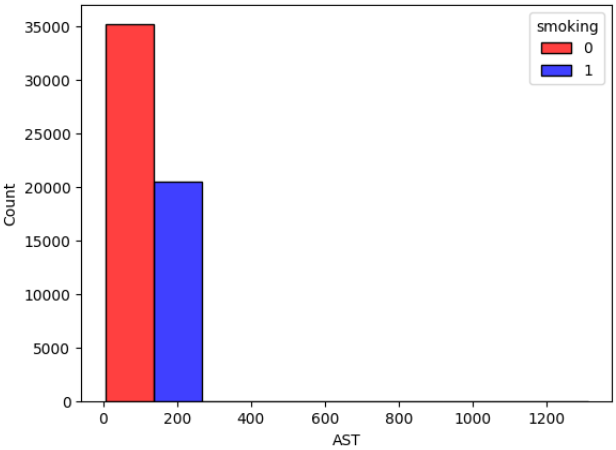

smoking
0
1
Count
HDL

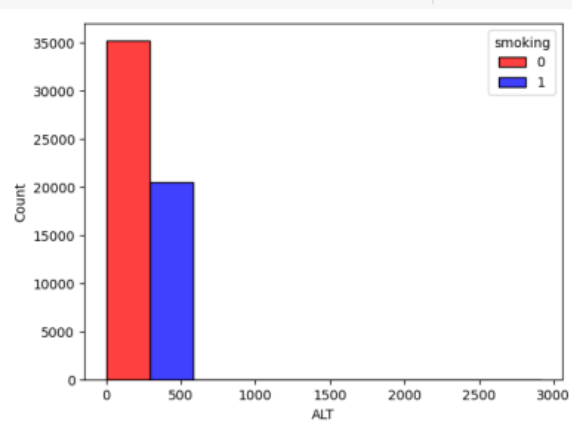

smoking
0
1
Count
LDL

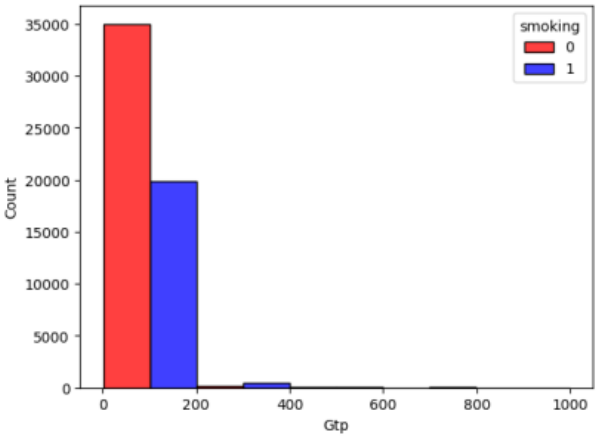

smoking
0
1
Count
hemoglobin

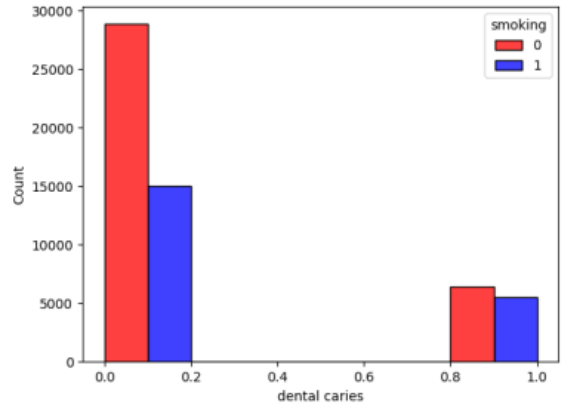

smoking
0
1
Count
Urine protein

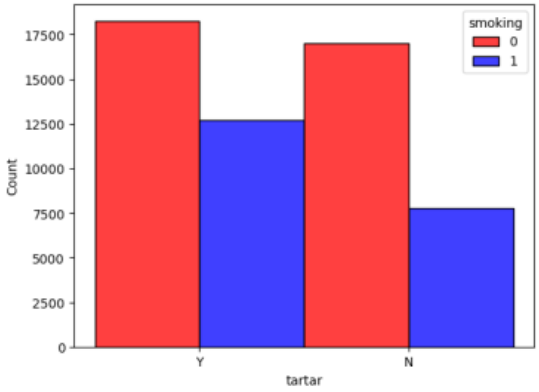

smoking
0
1
Count
serum creatinine

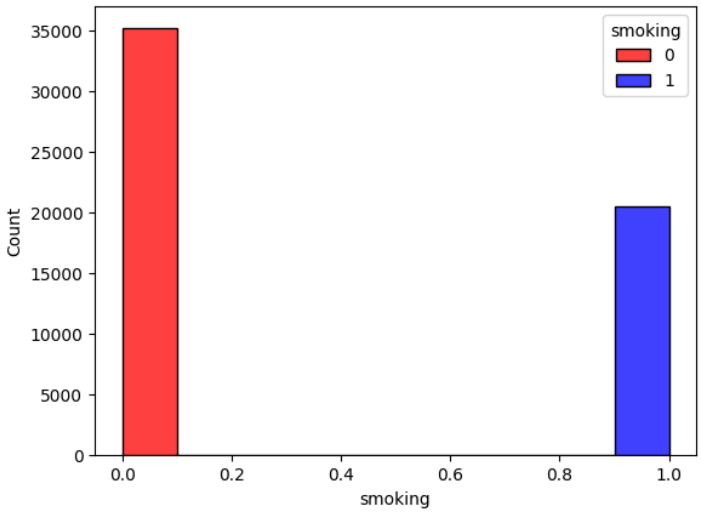

smoking
0
1
Count
AST











Zanimljivo je da među ženama ima znatno manje pušača.

Pretprocesiranje

Rad sa nedostajućim vrednostima

Zaključujemo da u skupu nema nedostajućih vrednosti.

Rad sa kategoričkim atributima

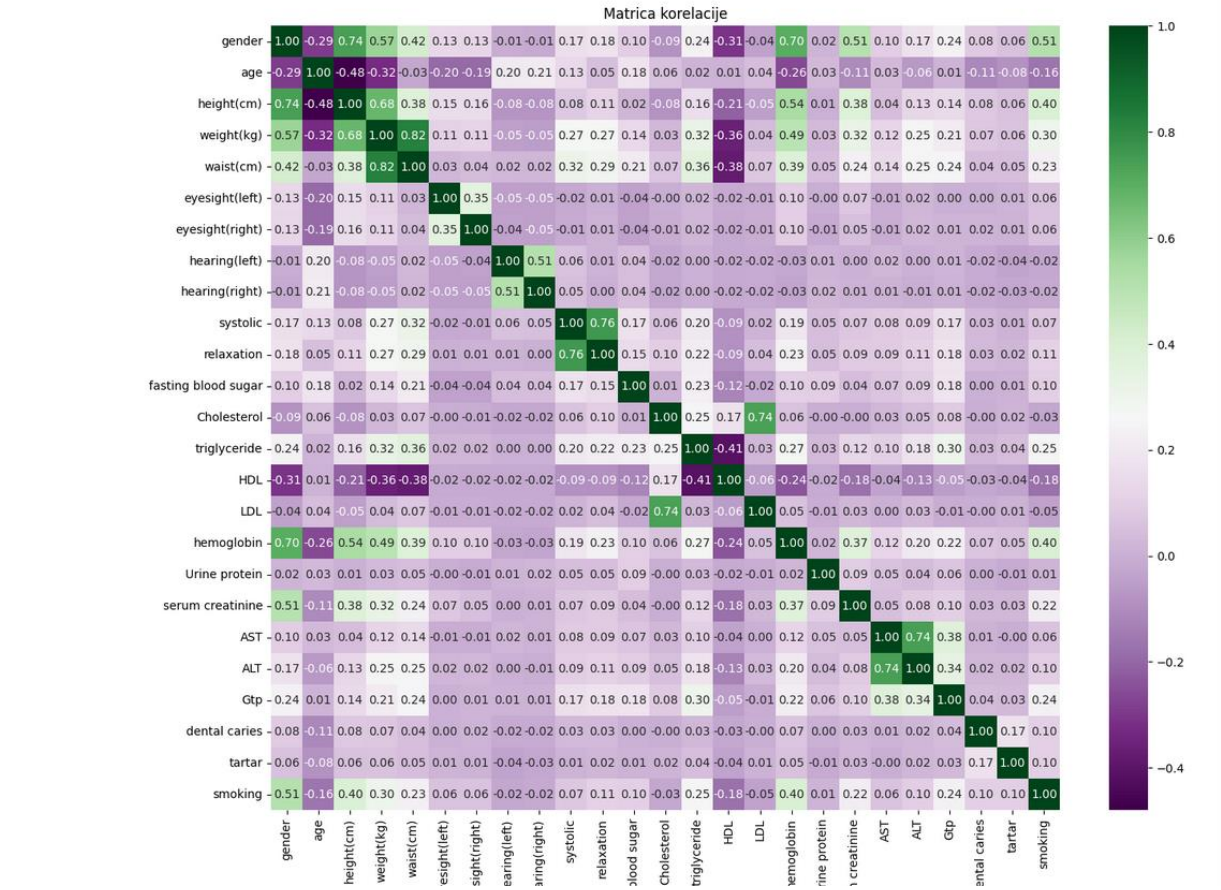
U skupu su nam preostala dva kategorička atributa gender I tartar koje ćemo preslikati u binarne koristeći funkciju LabelEncoder.

Odnosi između atributa u skupu I redukcija dimenzionalnosti

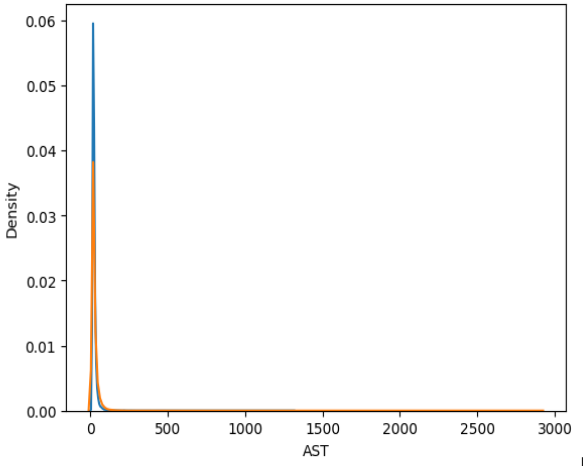
Kako je naš skup podataka veliki, teško je da uočimo odnose između atributa. Zato koristimo dijagram korelacije koji nam pruža vizuelni rezime.

Vidimo da postoje atributi koji su dosta povezani sa drugima. Neki od njih su:

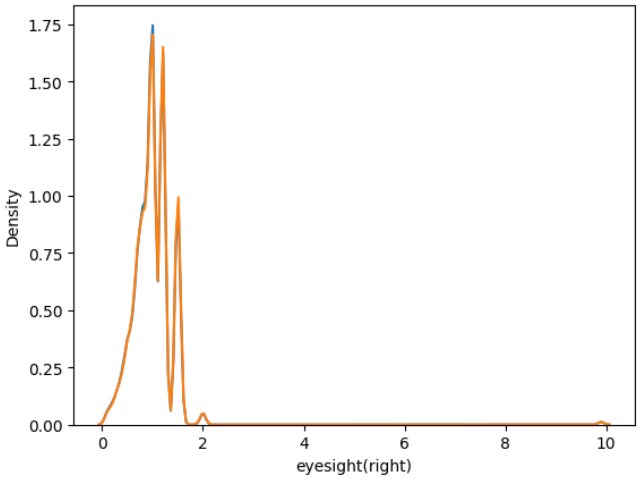
- `gender |-> height,hemoglobin`
- `height |-> gender,weight,hemoglobin,`
- `weight |-> gender,waist,hemoglobin`
- `eyesight_left |-> eyesight_right i obrnuto`
- `systolic |-> relaxation i obrnuto`
- `cholesterol |-> ldl i obrnuto`
- `ast |-> alt i obrnuto`



Nakon uočene visoke korelacije među atributima, hajde da uporedimo raspodele visokokorelisanih atributa kako bismo se uverili da nema velike razlike I da neke atribute možemo odbaciti.

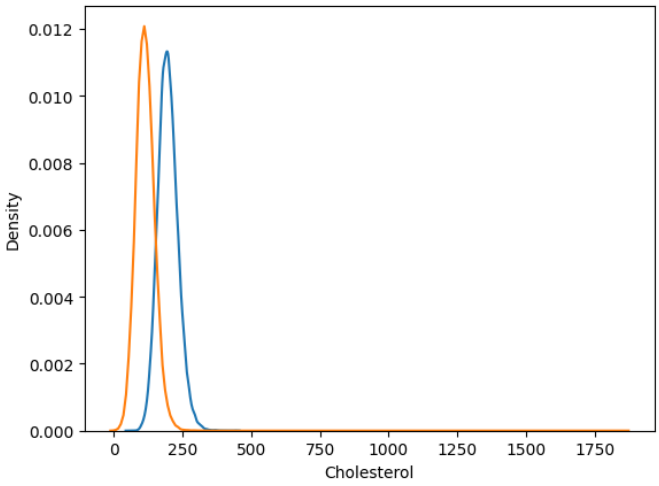


AST - ALT

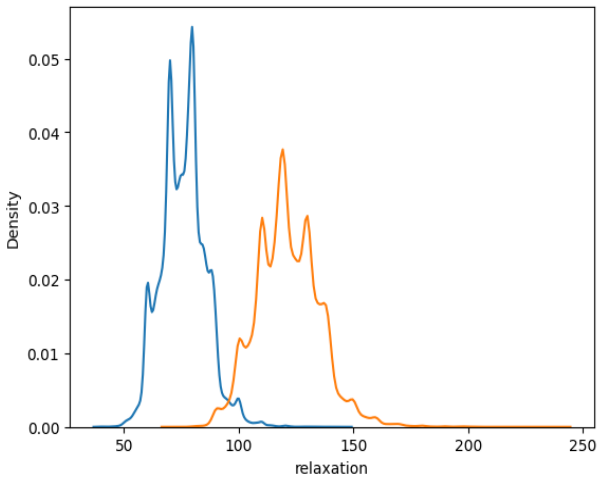


Eyesight (right) – eyesight (left)

Kao što vidimo raspodele su identične. Iz skupa uklanjamo ALT I eyesight(right).



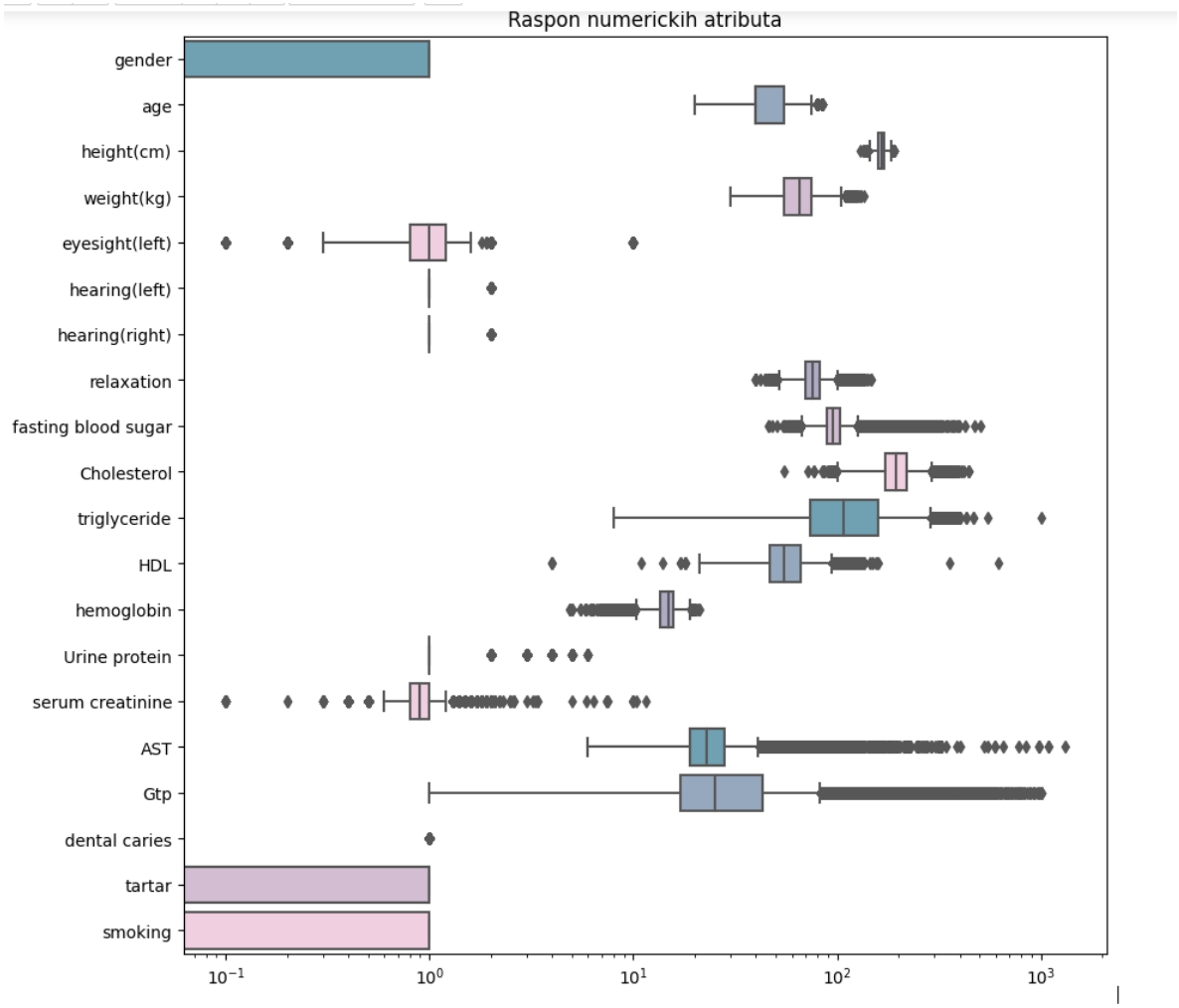
Cholesterol – LDL



relaxation – systolic

Vidimo da su raspodele za parove atributa Cholesterol – LDL, relaxation – systolic takođe jako slične pa uklanjamo I atribute LDL I systolic iz skupa.

Iz skupa podataka ćemo izbaciti I atribut waist(cm), jer je obim struka dosta uslovljen težinom, a I obim struka intuitivno neće uticati na to da li je osoba pušač ili nije.



Možemo zaključiti da su podaci u razlicitim opsezima tako da bismo morali da ih normalizujemo ukoliko model to zahteva.
Takođe treba da odradimo detekciju elemenata van granica, ali to ćemo nakon podele na test I trening skup, jer bismo u suprotnom kompromitovali naš test skup.

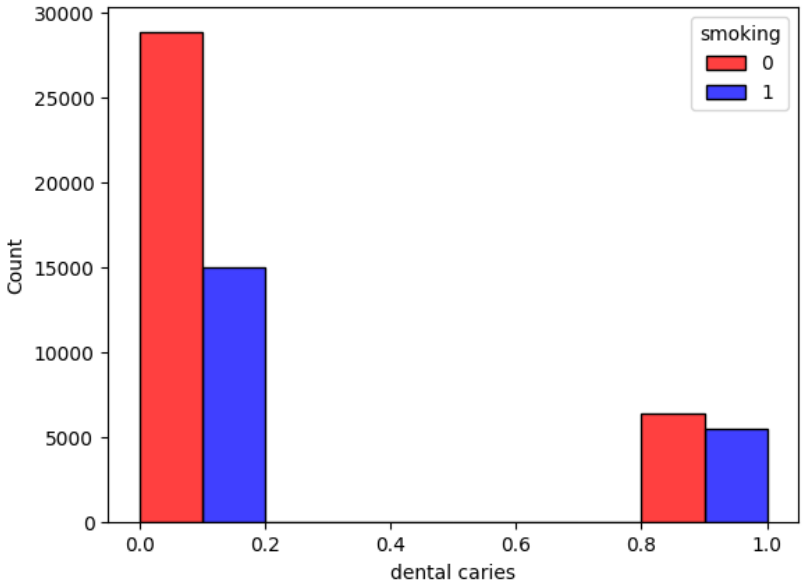
Podela na ulazne i ciljne atribute

Informacije o tome da li je osoba pušač se nalaze u koloni smoking što predstavlja ciljni atrinut. Ostali atributi se koriste za predikciju - ulazni atributi.

Prilikom podele, za test skup ćemo uzeti 30% podataka iz skupa, a ostalih 70% ćemo koristiti za trening skup.

Proveravanje vrednosti van granica

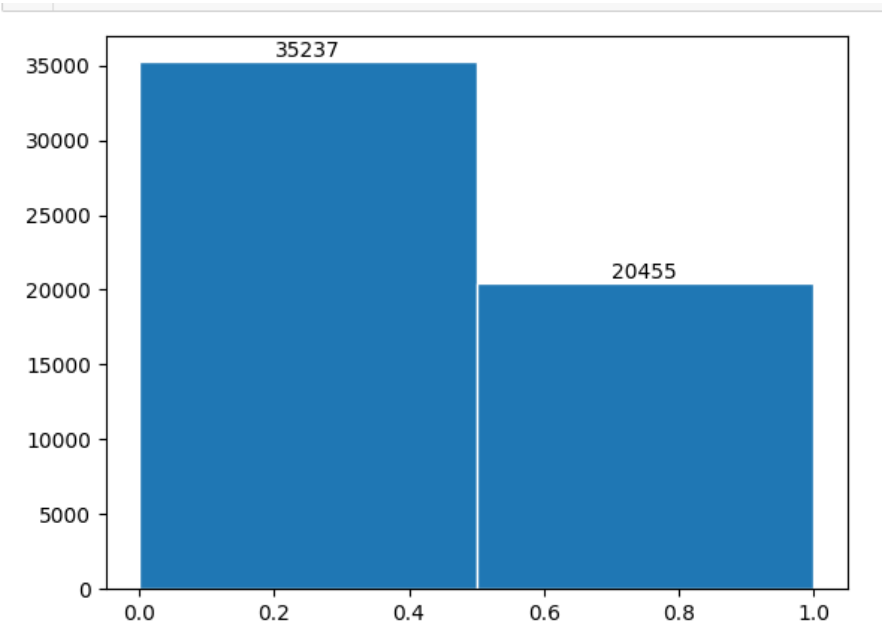
Koristeći statičku metodu za identifikaciju elemenata van granica – IQR, možemo zaključiti da u većini atributa nemamo elemente van granica ili ih imamo u zanemarljivo malom procentu. Atribut dental caries ima 21% elemenata van granica. Kako je to atribut koji uzima samo vrednosti 0 i 1 , postojanje elemenata van granica nam govori da postoji veliki broj 0 ili 1. Pogledajmo ponovo kako atribut dental caries utiče na ciljni atribut:

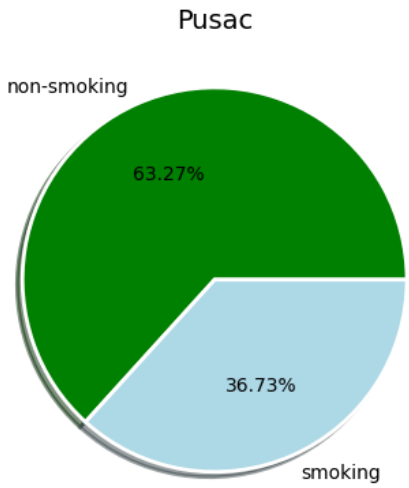


Možemo videti da je prisutna nebalansiranost klasa. Kasnije ćemo pre treniranja modela iskoristiti neku od tehnika za balansiranje klasa.

Provera balansiranosti ciljne klase

Na osnovu histograma vidimo da se u skupu nalazi 35237 (63.27%) osoba koje nisu pušači i 20455 (36.73%) osoba koje jesu pušači. Primećujemo blagu nebalansiranost. Pošto imamo blagu nebalansiranost koristićemo f1-score kao meru evaluacije.





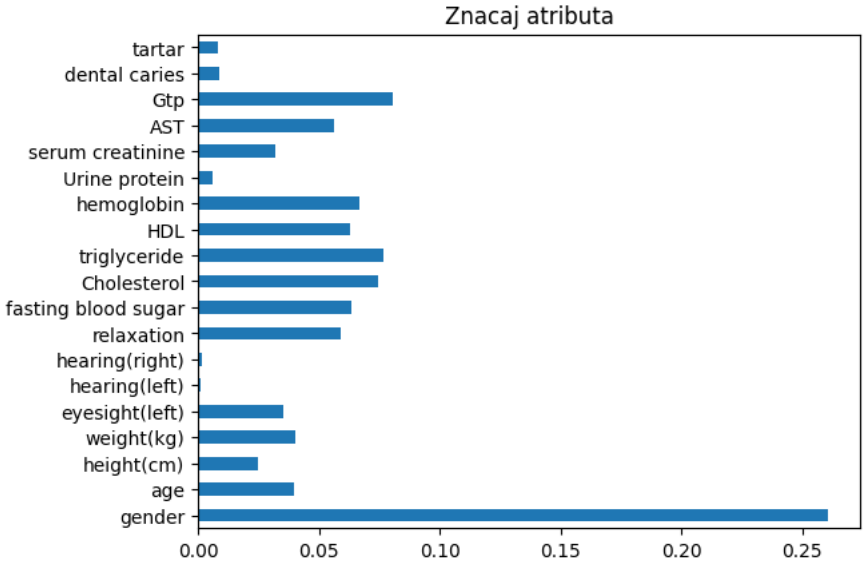
Klassifikacija

1 Decision Tree

Osnovna ideja stabla odlučivanja je da podeli skup podataka na sve manje i manje podskupove na osnovu atributa, tako da svaki podskup sadrži što sličnije primere. Ovi podskupovi se nazivaju čvorovima i formiraju hijerarhijsku strukturu, pri čemu svaki čvor predstavlja testiranje jednog atributa. Krajnji čvorovi, nazivaju se listovima, predstavljaju klasifikacijsku odluku.

S obzirom da je prisutna nebalansiranost klasa prilikom treniranja modela postavićemo parametar class_weight= 'balanced'.
Na početku kada istreniramo ovaj model dolazi do preprilagođavanja (f1-score na trening skupu je 100%, a na test skupu 75.5%).

Pogledajmo atribute koji su bili od značaja:



Vidimo da nam hearing(right) i hearing(left) imaju vrlo mali značaj prilikom formiranja stabla. Odmah za njima je i atribut Urine proteine. Pol ima najveći uticaj, ali to je i očekivano, jer smo na početku videli da među ženskom populacijom ima dosta manje pušača.

Podešavanjem hiper parametara uz korišćenje unakrsne validacije dobijamo da je najbolji kriterijum za merenje kvaliteta modela gini, a najbolja dubina stabla 10.

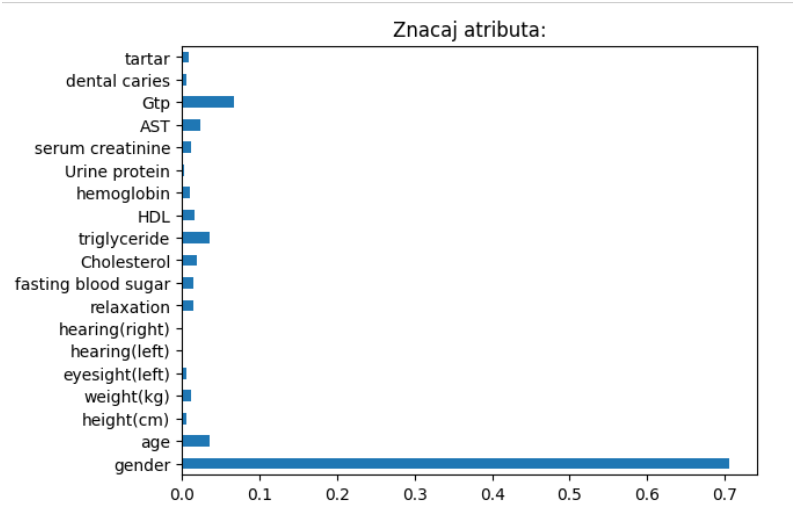
Ovaj model daje nam 73.5% kao najbolji f1 score.
Pogledajmo izvestaj klasifikacije, kao I atribute koji su bili od značaja:

```
: 1 report(estimator.best_estimator_, X_train, Y_train)
```

```
Izvestaj o klasifikaciji za modelDecisionTreeClassifier nad training podacima
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.91      0.71      0.80     24666
           1       0.64      0.89      0.74     14318

    accuracy                           0.78     38984
   macro avg       0.78      0.80      0.77     38984
weighted avg       0.81      0.78      0.78     38984

--------------------------------------------------------------------------------
Matrica konfuzije za model DecisionTreeClassifier nad training podacima
--------------------------------------------------------------------------------
       1      0
1  17541   7125
0   1644  12674
--------------------------------------------------------------------------------
```

```
: 1 report(estimator.best_estimator_, X_test, Y_test, "test")
```

```
Izvestaj o klasifikaciji za modelDecisionTreeClassifier nad test podacima
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.88      0.68      0.77     10571
           1       0.60      0.84      0.70      6137

    accuracy                           0.74     16708
   macro avg       0.74      0.76      0.73     16708
weighted avg       0.78      0.74      0.74     16708

--------------------------------------------------------------------------------
Matrica konfuzije za model DecisionTreeClassifier nad test podacima
--------------------------------------------------------------------------------
      1     0
1  7184  3387
0   977  5160
--------------------------------------------------------------------------------
```



Kao što možemo da vidimo, ubedljivo najveći značaj prilikom formiranja stabla imao je pol. Zatim Gtp, godine I trigliceridi. Sluh na levo I desno uho nije imao značaja, što je I očekivano.
Međutim, iznenađujuće je što karijes nema veći uticaj na klasifikaciju pušača.

2 *Random Forest*

Osnovna ideja Random Forest-a je kreiranje velikog broja stabala odlučivanja i kombinovanje njihovih predviđanja kako bi se donela finalna odluka.
Prilikom treniranja ovog modela postavili smo parametar class_weight= ‘balanced’.

Pre nego što smo podesili parametre došlo je do preprilagođavanja modela.

Sa podešenim parametrima pottižemo f1 score od 80.5% na test podacima.

```
: 1 report(rfc_cv, X_train, Y_train, "trening")
```

```
Izvestaj o klasifikaciji za modelGridSearchCV nad trening podacima
--------------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     24666
           1       0.94      1.00      0.97     14318

    accuracy                           0.97     38984
   macro avg       0.97      0.98      0.97     38984
weighted avg       0.98      0.97      0.97     38984

--------------------------------------------------------------------
Matrica konfuzije za model GridSearchCV nad trening podacima
--------------------------------------------------------------------
       1      0
1  23701    965
0     27  14291
--------------------------------------------------------------------
```

```
: 1 report(rfc_cv, X_test, Y_test, "test")
```

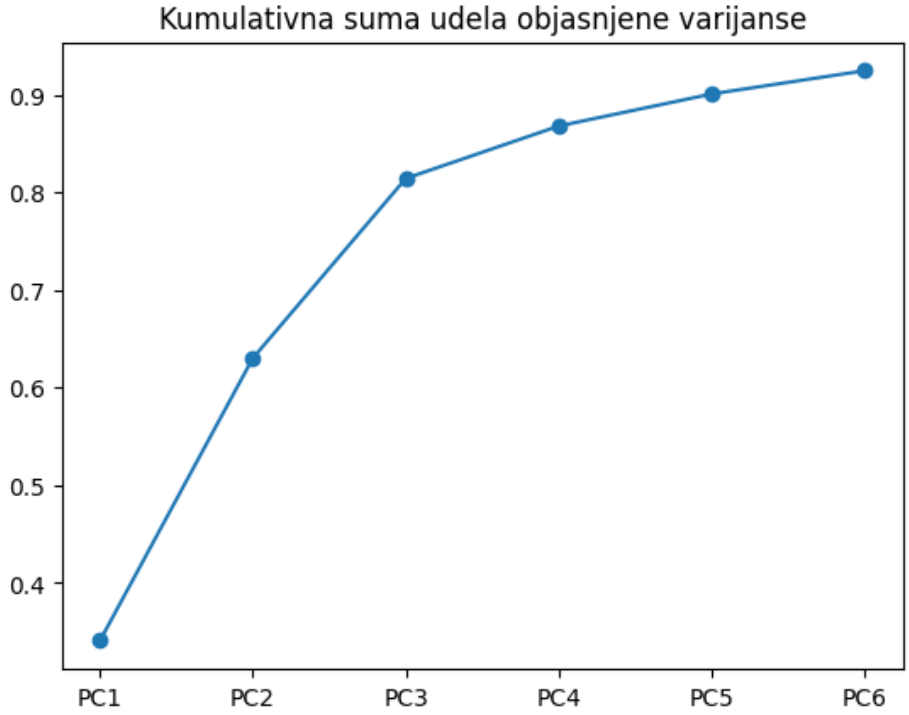
```
Izvestaj o klasifikaciji za modelGridSearchCV nad test podacima
--------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.90      0.80      0.84     10571
           1       0.71      0.84      0.77      6137

    accuracy                           0.81     16708
   macro avg       0.80      0.82      0.81     16708
weighted avg       0.83      0.81      0.82     16708

--------------------------------------------------------------------
Matrica konfuzije za model GridSearchCV nad test podacima
--------------------------------------------------------------------
      1     0
1  8442  2129
0   985  5152
--------------------------------------------------------------------
```

3 KNN

Osnovna ideja KNN algoritma je da za novi ulazni podatak (instancu) odredi njegovu klasu ili vrednost ciljne promenljive na osnovu "k" najbližih suseda iz trening skupa podataka. Računa se udaljenost između novog ulaznog podatka i svih trening instanci koristeći neku metriku udaljenosti (npr. Euklidsku udaljenost). Udaljeni susedi se sortiraju prema rastućem redosledu udaljenosti. Bira se prvih "k" najbližih suseda. Na osnovu klase najbližih suseda, novi ulazni podatak se dodeljuje klasi koja ima najviše suseda.

Pre treniranja modela iskoristićemo MinMaxScaler da normalizujemo podatke kako bismo osigurali ravnotežu između atributa I sprečili da atributi sa većim rasponom dominiraju prilikom računanja rastojanja.

Naš skup atributra nije mnogo veliki pa nam PCA neće doneti značajno poboljšanje, ali ćemo ga primeniti radi demonstracije.



Prvih 6 komponenti opisuju više od 90% skupa.

Pomoću RandomOverSampler-a smo izbalansirali klase.

Nakon treniranja modela dobijamo sledeći izvestaj klasifikacije:

```
report(knn_balans, X_resampled, Y_resampled)
```

```
Izvestaj o klasifikaciji za modelKNeighborsClassifier nad training podacima
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.89      0.79      0.84     24666
           1       0.81      0.91      0.86     24666

    accuracy                           0.85     49332
   macro avg       0.85      0.85      0.85     49332
weighted avg       0.85      0.85      0.85     49332

--------------------------------------------------------------------------------
Matrica konfuzije za model KNeighborsClassifier nad training podacima
--------------------------------------------------------------------------------
       1      0
1  19551   5115
0   2332  22334
--------------------------------------------------------------------------------
```

```
report(knn_balans, X_test, Y_test, "test")
```

```
Izvestaj o klasifikaciji za modelKNeighborsClassifier nad test podacima
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.83      0.72      0.77     10571
           1       0.61      0.75      0.67      6137

    accuracy                           0.73     16708
   macro avg       0.72      0.73      0.72     16708
weighted avg       0.75      0.73      0.73     16708

--------------------------------------------------------------------------------
Matrica konfuzije za model KNeighborsClassifier nad test podacima
--------------------------------------------------------------------------------
      1     0
1  7593  2978
0  1564  4573
--------------------------------------------------------------------------------
```

f1 score na trening skupu je 84%, a na test skupu 76.5%.

Pokušajmo sa podešavanjem hiper parametara.

Iako se čini da dolazi to preprilagođavanja, to nije slučaj. Pošto je za najbolji model izabran parametar weight='distance', dakle udaljenost suseda od instance utiče na klasifikaciju (sto je sused blizi instanci koju klasifikujemo, njegov "glas" ima veću težinu). Kada evaluiramo model nad trening skupom, sve instance za koje tražimo susede, imaju suseda na rastojanju 0 (to su one same) i onda njihova klasa uvek u potpunosti određuje klasu instance koju klasifikujemo. Intuitivno, ako je weights='distance' i u skupu za trening postoji identična instanca onoj koju klasifikujemo, onda će novoj instanci biti dodeljena ista klasa. A kada pokušamo sa evaluacijom na test skupu, sada instance koje klasifikujemo nemaju susede na razdaljini 0, pa mnogo više suseda utiče na klasifikaciju (pa je i tačnost manja).

```
1 report(knn_cv.best_estimator_, X_train, Y_train)
```

```
Izvestaj o klasifikaciji za modelKNeighborsClassifier nad training podacima
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     24666
           1       1.00      1.00      1.00     24666

    accuracy                           1.00     49332
   macro avg       1.00      1.00      1.00     49332
weighted avg       1.00      1.00      1.00     49332

--------------------------------------------------------------------------------
Matrica konfuzije za model KNeighborsClassifier nad training podacima
--------------------------------------------------------------------------------
       1      0
1  24666      0
0      0  24666
--------------------------------------------------------------------------------
```

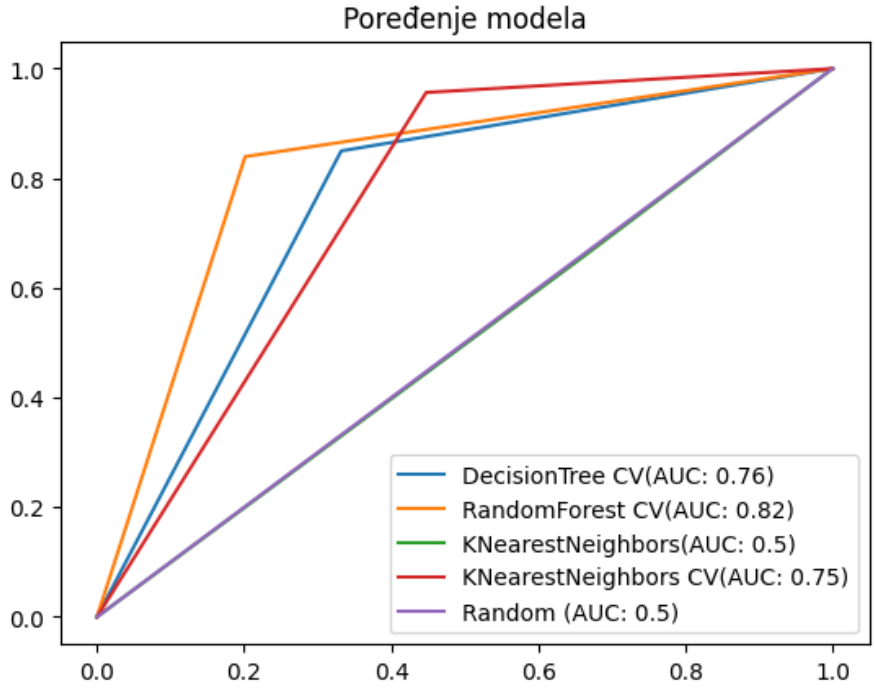
```
1 report(knn_cv.best_estimator_, X_test, Y_test, "test")
```

```
Izvestaj o klasifikaciji za modelKNeighborsClassifier nad test podacima
--------------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.94      0.74      0.83     10571
           1       0.67      0.91      0.77      6137

    accuracy                           0.80     16708
   macro avg       0.80      0.83      0.80     16708
weighted avg       0.84      0.80      0.81     16708

--------------------------------------------------------------------------------
Matrica konfuzije za model KNeighborsClassifier nad test podacima
--------------------------------------------------------------------------------
      1     0
1  7820  2751
0   525  5612
--------------------------------------------------------------------------------
```

Uspeli smo da dobijemo f1-score na test skupu 80%.

Poređenje klasifikacionih modela

Za poređenje modela koristićemo ROC krivu. Na x-osi su predstavljene FPR (lažno pozitivne) , a na y-osi TPR(stvarno pozitivne). Što je AUC (površina ispod krive) veća model je bolji.



Najbolji model je Random Forest sa podešenim parametrima, sa kojim smo uspeli da ostvarimo f1 score 80.5%, dok ništa mnogo za njim ne zaostaje ni KNN sa podešenim parametrima gde smo postigli f1 score 80%.

Klasterovanje

Klasterovanje je tehnika bez nadgledanja. Stoga, ne postoji koncept treninga i testiranja u tradicionalnom smislu kao kod nadgledanog učenja.
Klasterovanje se primarno koristi kako bi se otkrile prirodne strukture i veze unutar skupa podataka. To može pomoći u razumevanju skupa podataka ili pronalaženju grupa sličnih podataka. Evaluacija klasterovanja obično se obavlja na osnovu unutrašnjih metrika koje mere kompaktnost klastera i razdvajanje među njima.

Izdvojićemo ciljni atribut smoking.

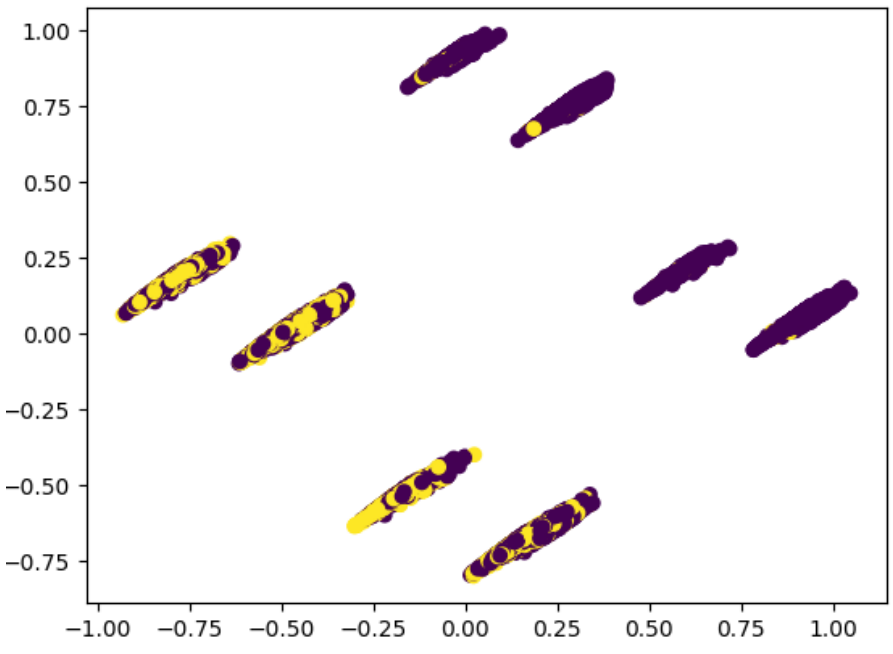
I ako smo ranije videli da prvih 6 komponetata u PCA najbolje opisuje naš skup (preko 90%), za vizuelizaciju klasterovanja uzimaćemo prve dve komponente jer je praktičnije I izvršavanje algoritama biće efikasnije.

Pre primene PCA normalizovaćemo podatke pomoću MinMaxScaler-a kako neka svojstva ne bi bila prenaglašena.

1 K-Means

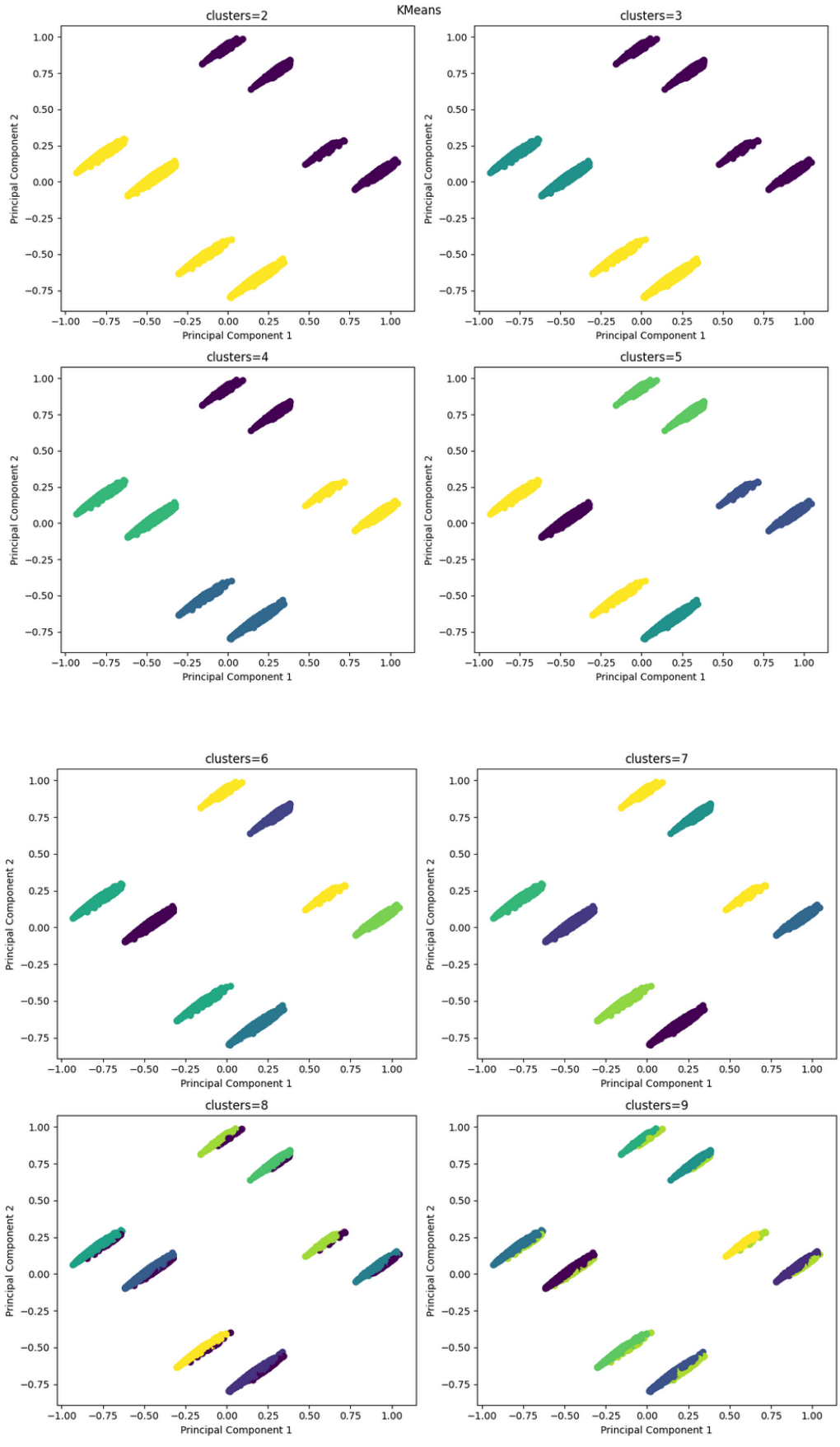
Glavna ideja K-Means algoritma je da podeli podatke u klaster tako da se minimizuje suma kvadrata udaljenosti između podataka i centara klastera.

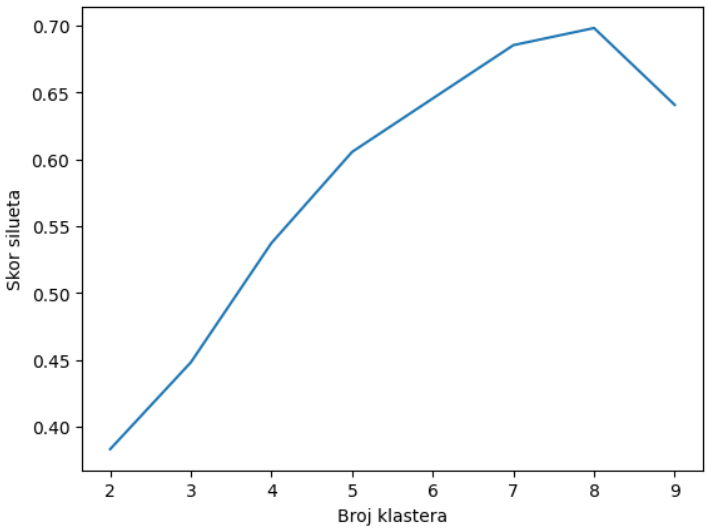
Pogledajmo kako u startu izgledaju naši podaci na raspršenom grafiku u dvodimenzionalnom prostoru:

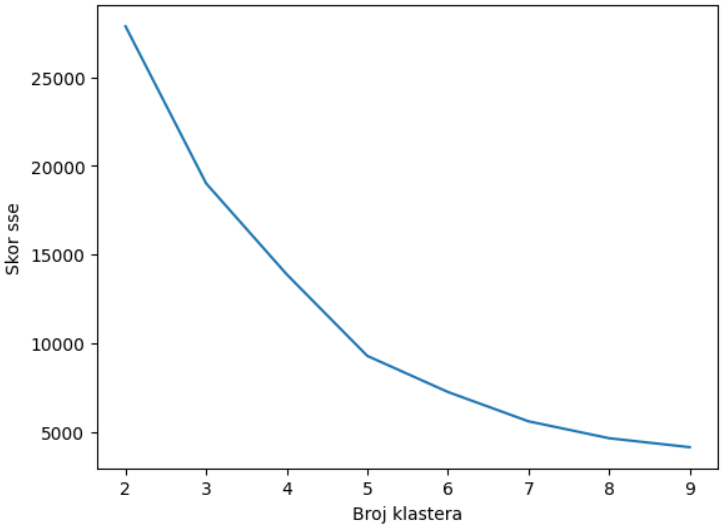


| Algoritam K-

Means zahteva da se unapred definiše broj klastera. Zbog toga ćemo definisati broj klastera u opsegu od 2 do 10 I testirati algoritam, pri čemu ćemo pamtiti inerciju I siluetu kao mere koje ćemo koristiti za ocenu rada algoritma.







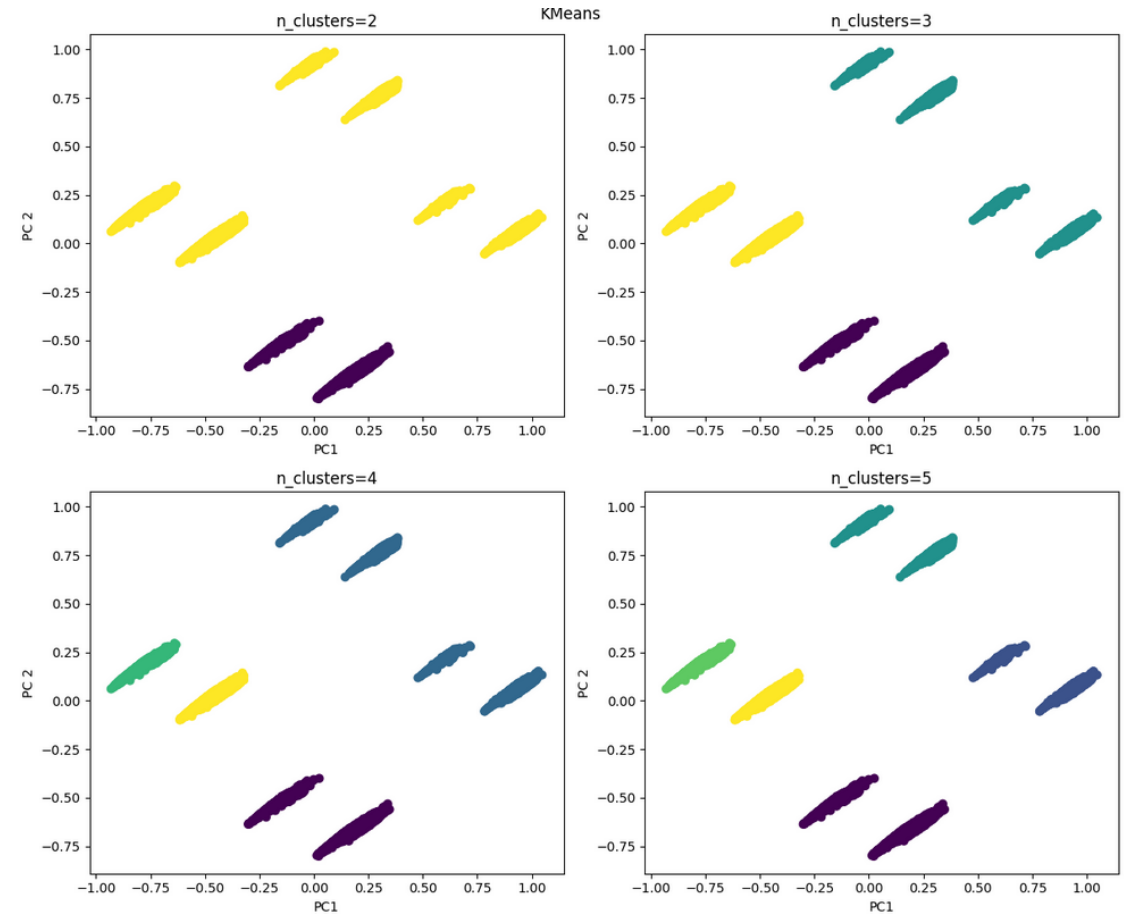
Niža vrednost SSE ukazuje na to da su tačke unutar klastera bliže centrima klastera, što bi bilo poželjno.
Viša vrednost siluete ukazuje na to da su tačke unutar klastera bliže jedna drugoj, a dalje od tačaka u drugim klasterima, što takođe želimo.
Pri upoređivanju ovih vrednosti, težimo da izaberemo broj klastera koji balansira smanjenje SSE i povećanje vrednosti siluete.

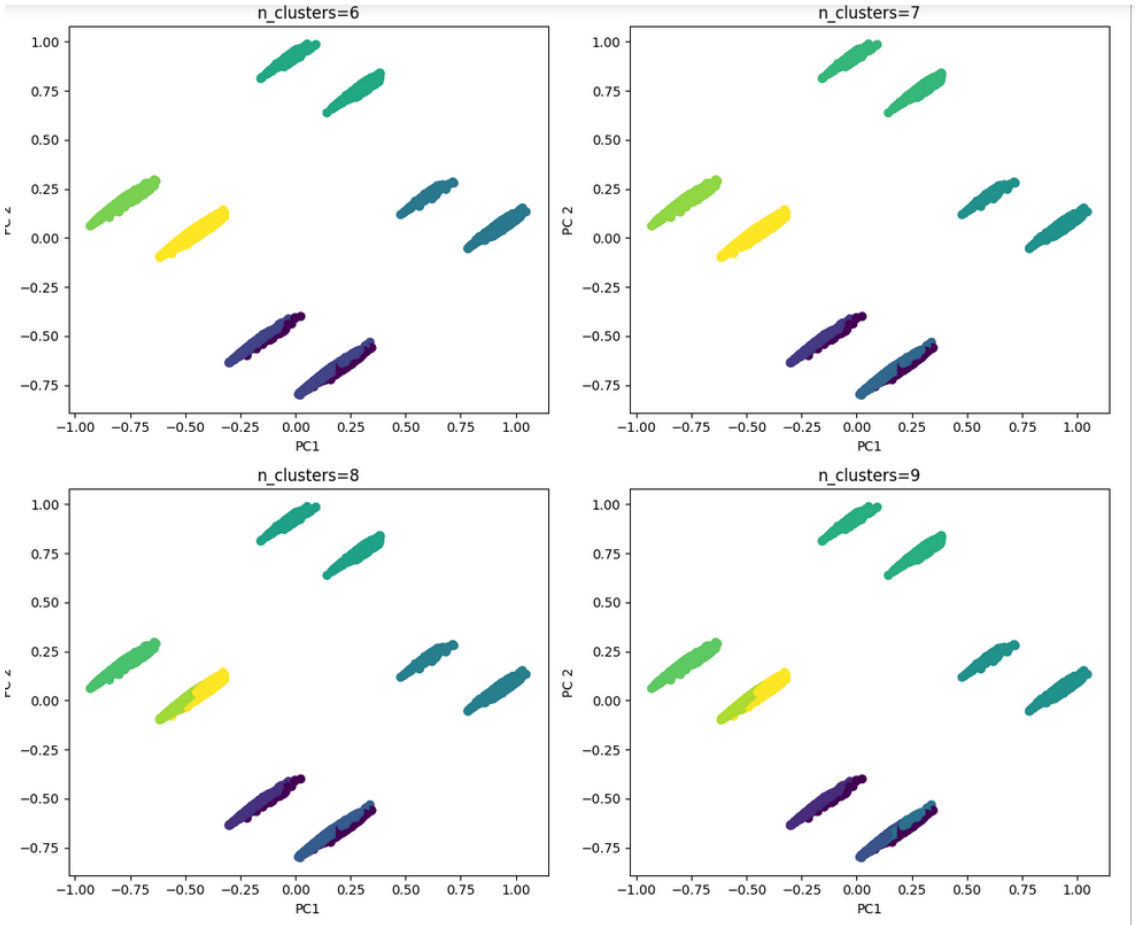
Tačka na grafiku gde varijabilnost prestane da se značajno smanjuje naziva se "lakat" tačka i predstavlja optimalan broj klastera. Na osnovu tog pravila I toga što je silueta skor najveći za 8 klastera I iznosi 69.8%, izdvojićemo taj model kao najbolji.

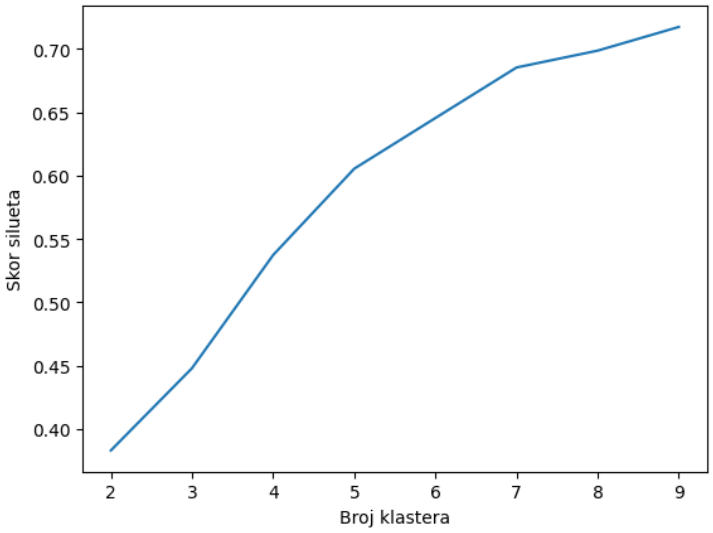
2 *Bisecting K-Means*

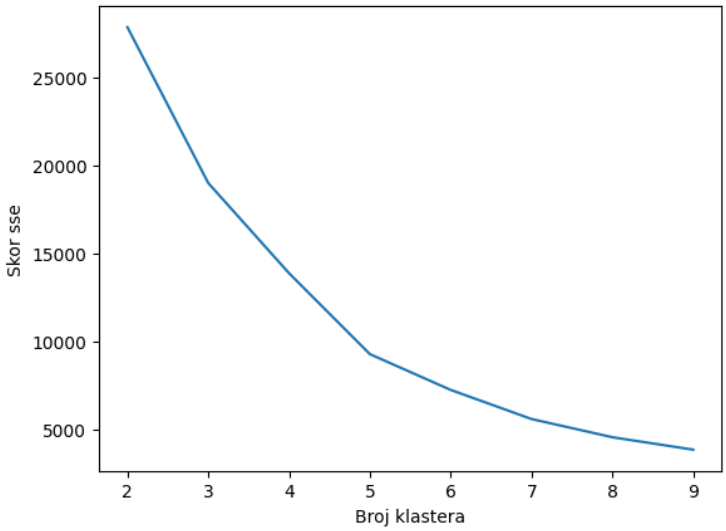
Bisecting K-Means je tehnika klasterovanja koja se bazira na iterativnom deljenju klastera kako bi se formirali podklasteri. Ova tehnika je varijacija klasičnog K-Means algoritma i koristi se za podelu klastera na više manjih klastera. Funkcioniše tako što se početno klasterovanje vrši nad celim skupom podataka koristeći K-Means algoritam sa određenim brojem klastera K. Nakon što se formira početni klaster, klaster sa najvećom varijansom (tj. najvećim rasponom podataka) se bira za deljenje. Taj klaster se deli na dva podklastera. Postupak se ponavlja iterativno na svakom podklasteru dok se ne dostigne određeni broj klastera ili dok se ispuni određeni uslov zaustavljanja.

Testirajmo algoritam za različit broj klastera:


KMeans
n_clusters=2
n_clusters=3
n_clusters=4
n_clusters=5
PC 2
PC1


n_clusters=6
n_clusters=7
n_clusters=8
n_clusters=9
PC 2
PC1

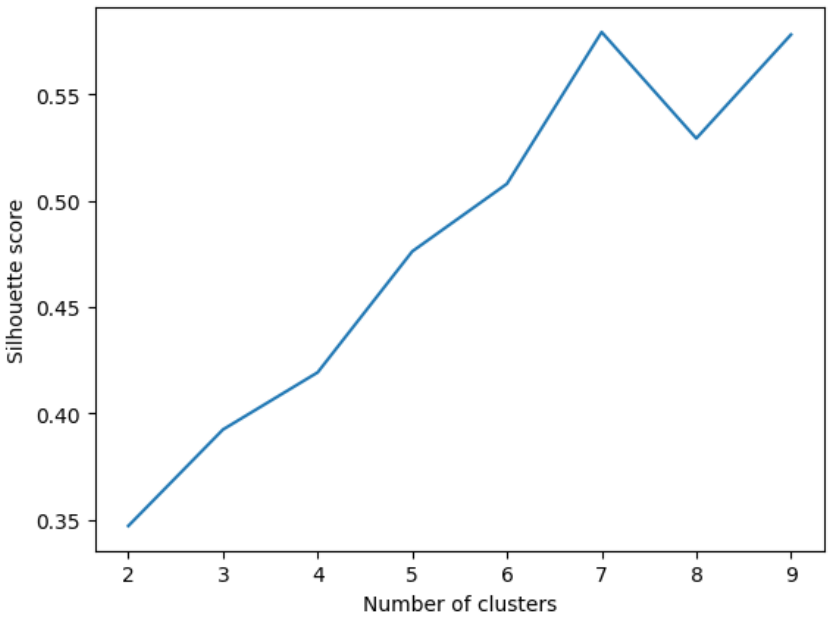




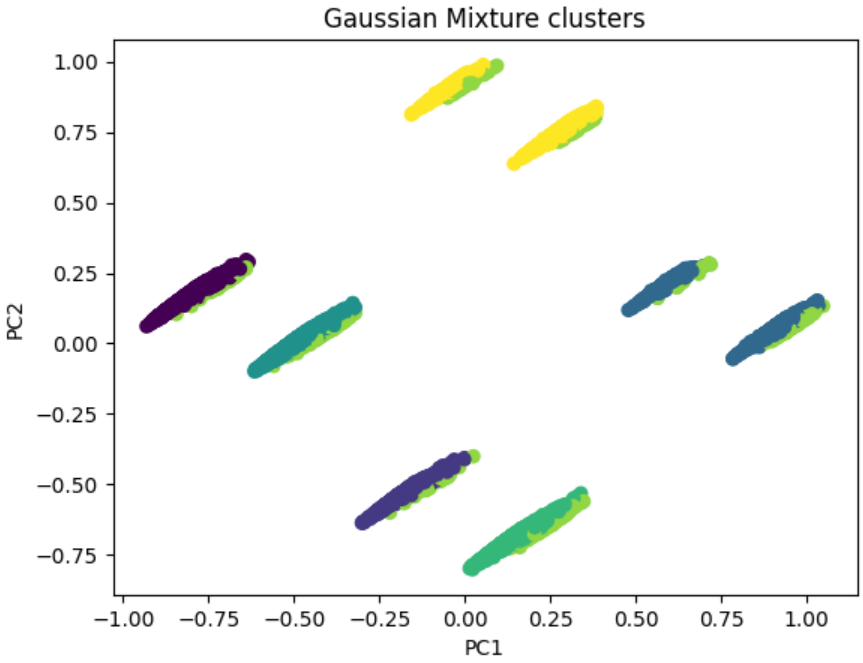
Sada deluje da je optimalan broj klastera 5. Zaista, ako istreniramo model za 5 I za 8 klastera dobijamo da je silueta skor za 5 klastera 58%, dok je za 8 klastera 50%. Izdvojićemo model sa 5 klastera kao najbolji.

3 *Gausian Mixture*

Ovaj algoritam koristi Gasovu raspodelu da odredi verovatnoću da određena tačka pripada određenom klasteru. Omogućava meku dodelu klastera, gde svaki podatak može imati verovatnoću pripadnosti više klastera.
Ponovo ćemo testirati algoritam za različit broj klastera I pamtiti silueta koeficijent.

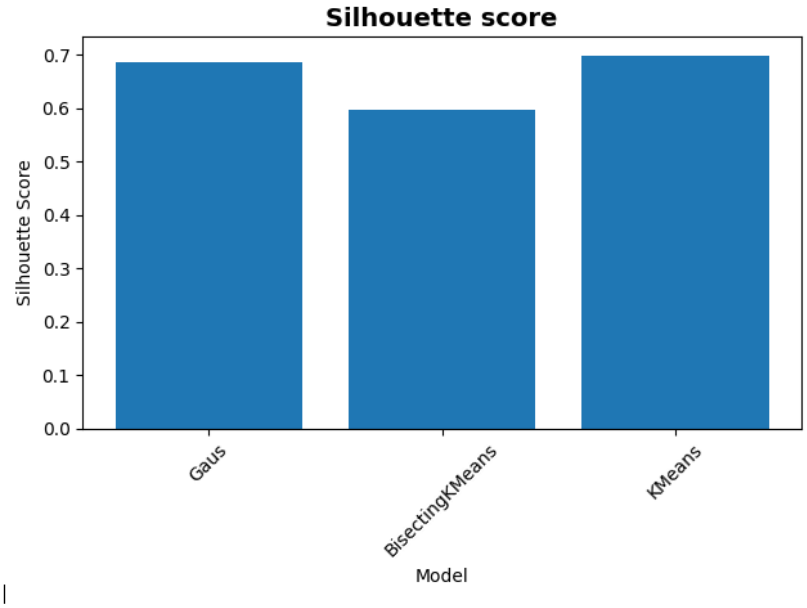


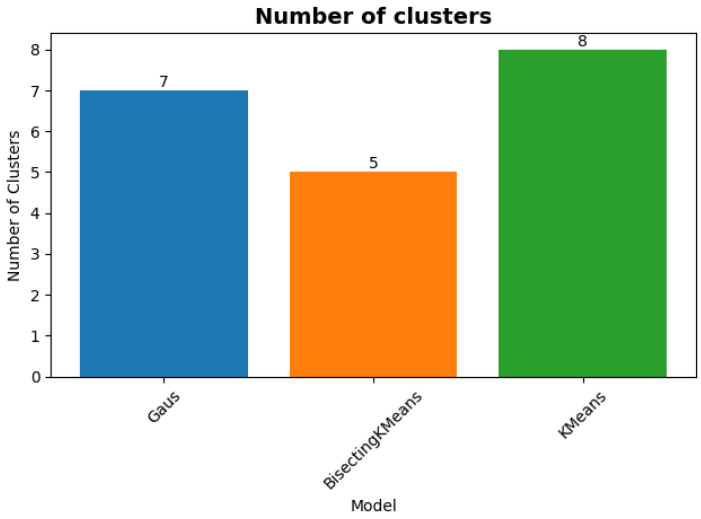
Kao što možemo da vidimo silueta skor je najveći za 7 klastera, pa taj model uzimamo kao njabolji. Silueta skor je 55%.



Poređenje modela klasterovanja

Uporedićemo modele prema najboljoj oceni siluete.





Vidimo da Kmeans daje za nijansu bolje rezultate od Gausa, dok Bisecting Kmeans daje nešto losiji rezultat.

Kako ciljni atribut smoking ima dve klase, istreniraćemo svaki model za dva klastera I ARI metrikom ćemo proveriti sličnost klasterovanih I stvarih klasa.

Adjusted Rand Index (ARI): Ova metrika meri sličnost između klasterovanja i stvarnih klasa, uzimajući u obzir nasumične promene. Veće vrednosti ukazuju na bolje podudaranje.

- Vrijednost bliska 1 ukazuje na savršeno podudaranje između klasterovanja i stvarnih klasa.
- Vrijednost bliska 0 ukazuje na slučajno klasterovanje, gde se klasteri ne podudaraju sa stvarnim klasama više nego što bi se to očekivalo nasumično.
- Vrijednost bliska -1 ukazuje na suprotno klasteriranje u odnosu na stvarne klase.

Dobijamo vrednost 0.16, što znači da se klasteri ne podudaraju sa stvarnim klasama više nego sto bi se to očekivao nasumično.

Pravila pridruživanja

Određivanje prvila pridruživanja je proces u kome se za dati skup transakcija pronalaze pravila koja predviđaju pojavljivanje stavke na osnovu pojavljivanja ostalih stavki u transakcijama.

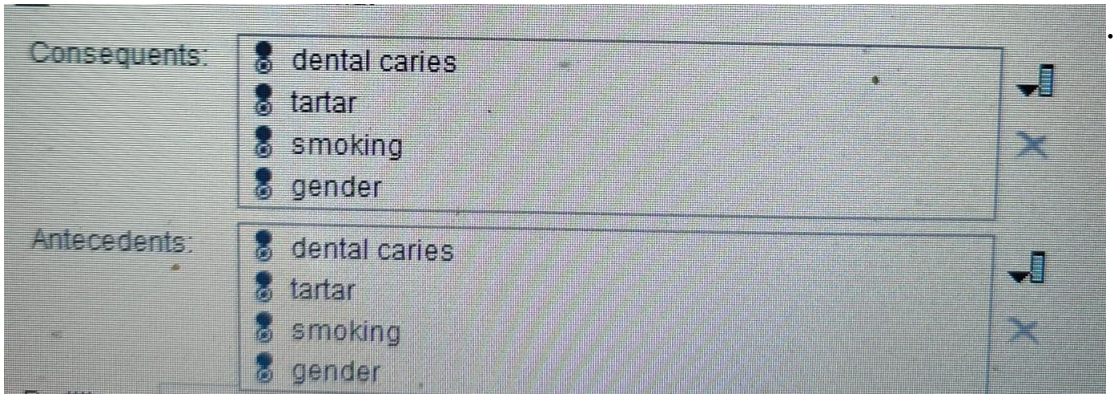
U IBM SPSS Modeler-u implementiraćemo jedan od najpoznatijih algoritama za izdvajanje pravila pridruživanja – Apriori algoritam.

Prvi korak je da učitamo podatke I odradimo binovanje, kako bismo dobili pogodan format za dalji rad.

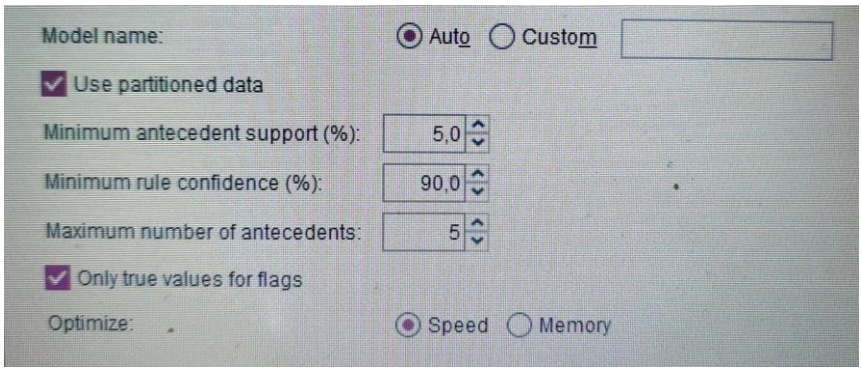
Pogledajmo kako naši podaci izgledaju:

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
gender		Flag	–	–	–	–	–	2	55692
age		Continuous	20	85	44.183	12.071	0.268	–	55692
weight(kg)		Continuous	30	135	65.865	12.820	0.534	–	55692
relaxation		Continuous	40.000	146.000	76.005	9.679	0.395	–	55692
Cholesterol		Continuous	55.000	445.000	196.901	36.298	0.392	–	55692
dental caries		Flag	0	1	–	–	–	2	55692
smoking		Flag	0	1	–	–	–	2	55692

Pokrećemo algoritam sa sledećim opcijama:



Atribute dental caries, tartar, smoking I gender dodaćemo u obe liste, što znači da mogu da se pojave I u telu I u glavi pravila.
Pri podešavanju algoritma postavili smo minimalnu podršku na 4% što označava koliko se često pravilo pojavljuje, minimalnu pouzdanost na 90%.



Izlaz Apriori algoritma za ove parametre:

Consequent	Antecedent	Support %	Confidence %
gender	dental caries smoking	9,887	96,204
gender	dental caries smoking tartar	7,443	96,116
gender	smoking	36,729	95,801
gender	smoking tartar	22,768	95,702

Interesantno, ako je osoba muskog pola velika je verovatnoća da će biti pušač.

Zaključak

Kada je u pitanju klasifikacija, najbolje rezultate daje algoritam Slučajnih šuma sa podešenim parametrima gde postižemo f1 score 80.5%.

Kod klasterovanja dobijamo poprilično loše rezultate.

Pravila pridruzivanja nam nisu dala mnogo, mada možemo videti da na to da li je osoba pušač dosta utiče kog je pola.