

“*Stroke prediction datasets*”

autor : Marija Papović

profesor: Nenad Mitić

asistent: Marija Erić

Avgust 15 , 2023

**Seminarski rad u okviru kursa Istraživanje podataka 1
na Matematičkom fakultetu u Beogradu**

Uvod

U ovom istraživanju rađeno je sa bazom podataka koji se mogu preuzeti na sledećem linku: [Stroke Prediction Dataset](#).

Reč je o medicinskim podacima nad kojima će raditi detaljnu analizu, a nakon toga ih sređivati i demonstrirati razne algoritme koje smo radili na pomenutom kursu.

Sa obzirom da je reč o medicinskim podacima i da već imamo šum u naše podatke, trudila sam se da dobijem onoliko dobre rezultete, koliko to oni dozvoljavaju, sa minimalnom izmenom početnih vrednosti kako ne bih unela dodatni šum.

Analiza skupa podataka, delimično preprocesiranje i vizuelizacija

Skup podataka sadrži 12 atributa:

- 1) id: int64 – Identifikator ispitanika
 - 2) gender: ["Male", "Female", "Other"] - pol
 - 3) age: float64 - godine
 - 4) hypertension: [0,1] - hipertenzija (Visok krvni pritisak)
 - 5) heart_disease: [0,1] – bolesti srca
 - 6) ever_married: ["No", "Yes"] – da li je ispatinak nekad bio u braku
 - 7) work_type:["Children", "Govt_jov", "Never_worked", "Private", "Self-employed"] – tip posla (“Deca”, “Posao u vladii”, “Nikad nije radio”, “Privatan posao”, “Samozaposlen”)
 - 8) Residence_type: ["Rural", "Urban"] – tip prebivališta
 - 9) avg_glucose_level: float64 – prosečan nivo glukoze
 - 10) bmi: float64 – indeks telesne mase
 - 11) smoking_status: ["formerly smoked", "never smoked", "smokes", "Unknown"]*
 - 12) stroke: [1,0] - udar
- * "Unknown" u smoking_status znači da se ispitanik nije izjasnio

Iz naziva atributa I vrednosti koje one uzimaju, možemo zaključiti o čemu se radi, šta opisuju I odmah možemo videti da imamo heterogenost među tipovima podataka.

U bazi imamo 5110 instanci, možemo zaključiti da će nam ciljni atribut biti stroke I pretpostaviti da nam atribut ID neće biti od neke koristi za naš dalji rad, što smo I proverili u kodu, pa ćemo ovaj atribut izbaciti.

Nedostajuće vrednosti

Među podacima imamo nedostajuće vrednosti, a to smo proverili pomoću naredne funkcije:

```
|: df.isna().sum()
|: id          0
|: gender      0
|: age         0
|: hypertension 0
|: heart_disease 0
|: ever_married 0
|: work_type    0
|: Residence_type 0
|: avg_glucose_level 0
|: bmi          201
|: smoking_status 0
|: stroke       0
|: dtype: int64

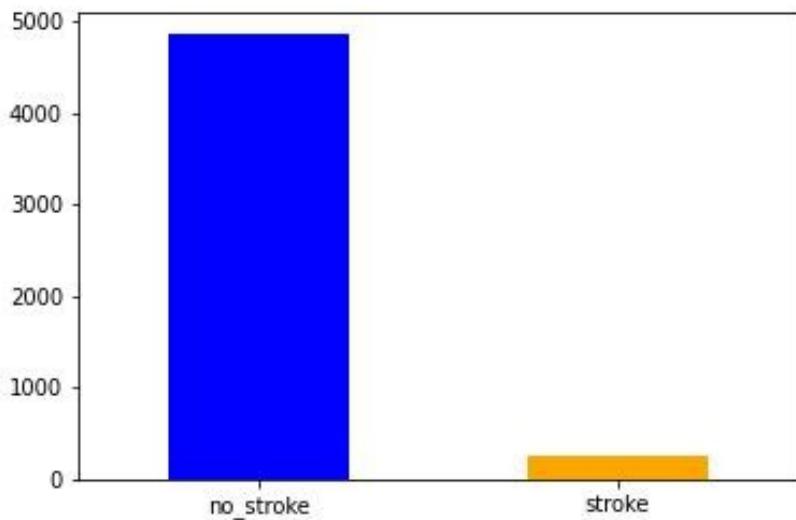
|: size_with_na = len(df)
|: size_without_na = len(df.dropna())
|
|: print(size_with_na , size_without_na , (size_without_na / len(df)) *100 )
|
5110 4909 96.0665362035225
```

Vidimo da atribut bmi ima 201 nedostajuću vrednost, iako to čini manje od 4% naših podataka nećemo ih ukloniti, jer će nam oni možda baš biti od koristi već ćemo iskoristiti funkciju koja postavlja nedostajuće vrednosti na mean vrednost atributa: **df['bmi'].fillna(df['bmi'].median(), inplace=True)**. Problem kod ovog pristupa je to što možemo imati potencijalno mnogo identičnih vrednosti, međutim na osnovu histograma koji se može naći u kodu videla sam da ovo neće mnogo izmeniti podatke.

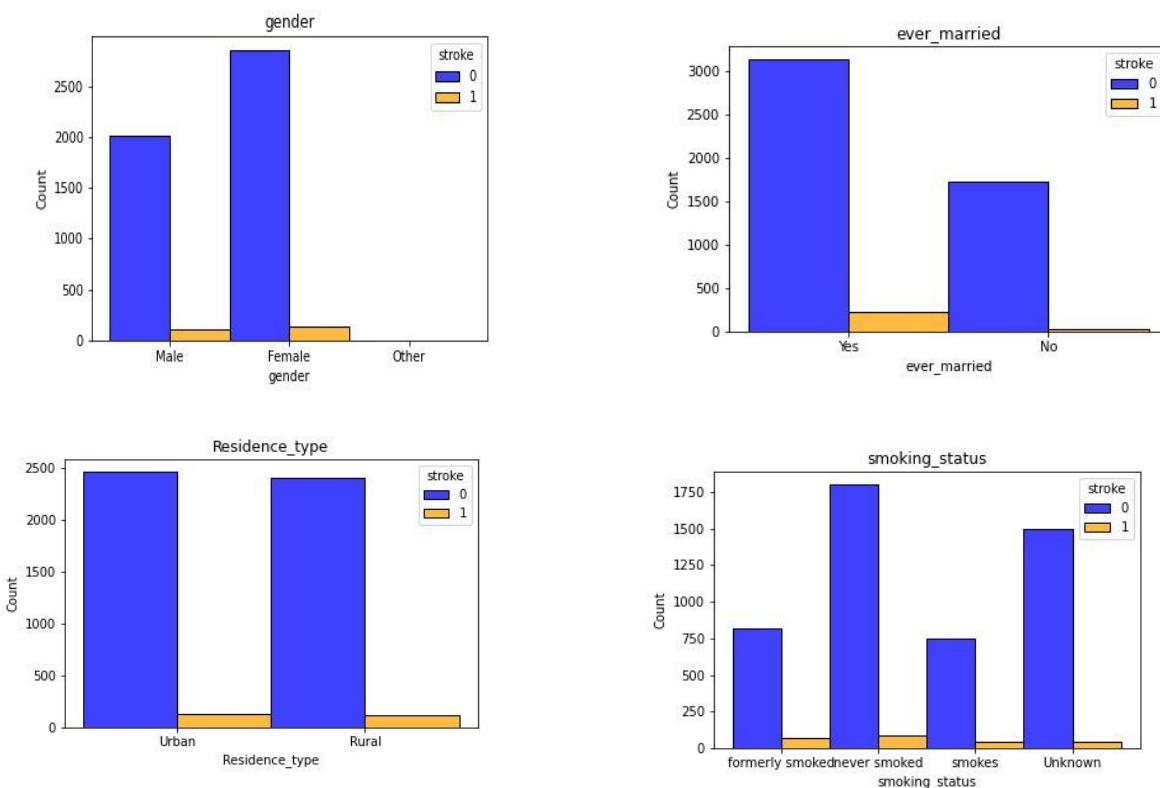
Ovime smo uklonili nedostajuće vrednosti I možemo ići dalje sa analizom.

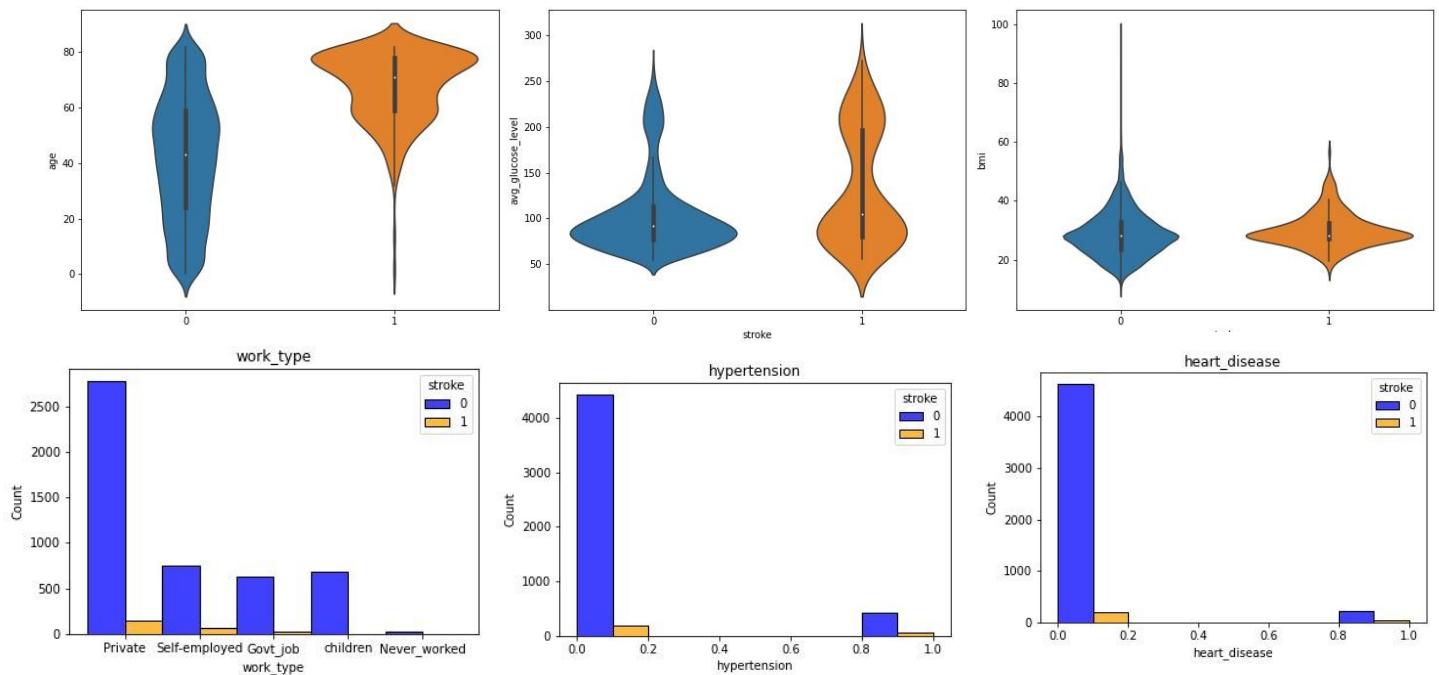
Vizualizacija ciljnog i ulaznih atributa i njihova međusobna zavisnost

Izvućićemo ciljni atribut koji je binarnog tipa i videti balansiranost klasa.



Vidimo veliku nebalansiranost medju podacima, medjutim tim problemom ćemo se baviti jasnije. Dalje, želimo da vidimo kako naš ciljni atribut zavisi od drugih atributa.





Na osnovu ove analize videli smo da vrednost other ne utiče ni malo na naše klase pa smo taj podatak izbacili iz skupa. Takođe smo dobili uvid u to koje vrednosti pojedinačnih atributa imaju najveći uticaj na naš ciljni atribut , a jasniju analizu možemo dobiti matricom korelacija.

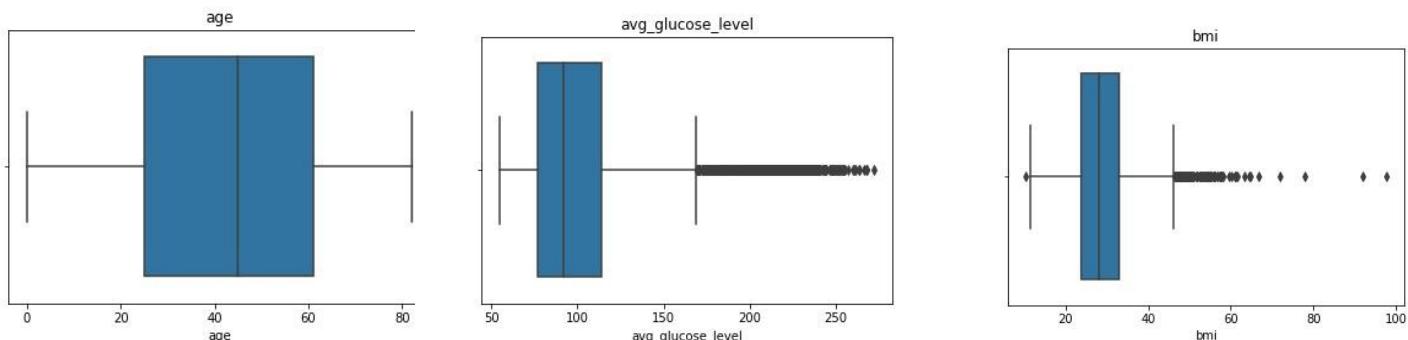


Iz ovoga možemo zaključiti da će nam godine najviše uticati na ciljni atribut, a da nam neki atributi neće biti od velikog značaja, kao što je Residence type, međutim pošto nemamo veliku međusobnu povezanost atributa nećemo izbacivati ni jedan atribut iz našeg skupa.

Elementi van granica

Pomoću kvantila smo proverili za naše neprekidne podatke da li imamo elemente van granica I dobili sledeće:

```
'age': 0, 'avg_glucose_level': 626, 'bmi': 122
```



Pomoću boxplota smo videli kako su naši podaci predstavljeni I zaključili da imamo veliki broj podataka sa malim vrednostima, a mali broj sa velikim, nemamo negativne podatke pa smo zaključili da koristimo log funkciju kako bi smanjili broj autlajera.

Nakon primene dobijamo sledeće: { 'avg_glucose_level': 380, 'bmi': 84 }

Nismo u potpunosti uklonili, ali smo dobili manji broj I ostavićemo tako da ne bi previše izmenili naše podatke.

Klasifikacija

U ovom poglavlju rešavamo problem binarne klasifikacije. Cilj klasifikacije će nam biti da imamo što manji broj False Negative instanci, sa obzirom da radimo na predviđanju moždanog udara.

Kako imamo veliku nebalansiranost, tačnost nam neće biti od značaja već ćemo posmatrati druge mere, kao što su recall i f1, a više o tome ću pričati kada budemo analizirali same algoritme.

Metode koje ćemo obraćivati su:

1. Stabla odlučivanja
2. Algoritam K najbližih suseda
3. Balansirana slučajna šuma

Preprocesiranje

Preprocesiranje je isto za sve gore navede metote.

Prvo sam učitala sredjeni skup podataka, zatim sam imenke kategoričke atributa pretvorila u numeričke.

Skup sam podelila na trening I test u razmeri 70:30, trening skup ćemo koristiti za pravljenje modela, a na test skupu ćemo videti kako naš model radi.

Nakon toga izvršila standradizaciju podataka jer će nam to trebati za KNN algoritam.

Stabla odlučivanja

Prvo sam formirala stablo sa samo jednim parametrom a to je random_state=42, I iscrtala drvo kako bih imala uvid u maksimalnu dubinu stabla I u to kako to drvo izgleda I ispitala kako atributi utiču na naše stablo odlučivanja.

Classification report for model DecisionTreeClassifier on trening data

	pre	rec	spe	f1	geo	iba
0	1.00	1.00	1.00	1.00	1.00	1.00
1	1.00	1.00	1.00	1.00	1.00	1.00
avg / total	1.00	1.00	1.00	1.00	1.00	1.00

Confusion matrix for model DecisionTreeClassifier on trening data

	0	1
0	3402	0
1	0	174

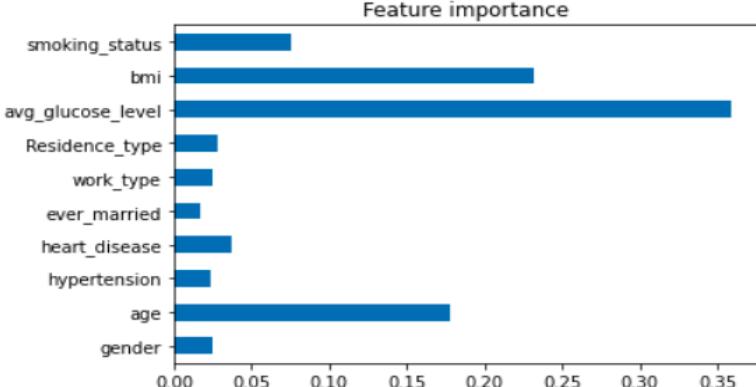
Classification report for model DecisionTreeClassifier on test data

	pre	rec	spe	f1	geo	iba
0	0.96	0.95	0.20	0.95	0.44	0.20
1	0.16	0.20	0.95	0.18	0.44	0.18
avg / total	0.92	0.91	0.24	0.91	0.44	0.20

Confusion matrix for model DecisionTreeClassifier on test data

	0	1
0	1380	78
1	60	15

Feature importance



Sledeće šta sam uradila jeste to da sam pomoću GridSearchCV trenirala model sa sledećim parametrima :

Naš model se preprilagodio I daje odlične rezultate na trening skupu sa FN I FP instanci jednakim 0, međutim ovo je ono što smo dobili nad test podacima:

Na test podacima vidimo da je broj FN instanci preveliki ,veći od TP I da nam ovaj model ne daje dobre rezultate.

Takođe vidimo da nam svi atributi učestvuju u pravljenju stabla a da nam numerički imaju najveći uticaj.

Maksimalna dubina stabla nam je 16 pa ćemo pokušati da smanjenjem dubine I podešavanjem parametara rešimo preprilagođenost.

```
'criterion': ['gini', 'entropy'],
'max_depth': [10,11,12,13,14,15],
class_weight' : [None , {0:1 ,1:23}, {0:1,1:30},{0:1,1:40} , {0:1 ,1: 50}, {0:1 , 1:60}]
```

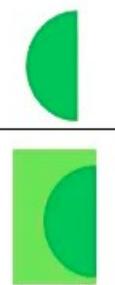
Za GridSearchCV sam stavila da scoring bude recall.

Ovime sam htela da sprečim preprilagođenost, a ponoću class_weight pokušam da dam veći značaj klasi 1 I takođe smanjim broj FN instanci.

Dobili smo bolji model sa sledećim parametrima: {'class_weight': {0: 1, 1: 60}, 'criterion': 'gini', 'max_depth': 10}

Classification report for model DecisionTreeClassifier on training data						
	pre	rec	spe	f1	geo	iba
0	1.00	0.74	1.00	0.85	0.86	0.72
1	0.17	1.00	0.74	0.29	0.86	0.76
avg / total	0.96	0.76	0.99	0.83	0.86	0.73

Confusion matrix for model DecisionTreeClassifier on training data		
0	1	
0	2529	873
1	0	174

$$\text{Recall} = \frac{\text{TP}}{\text{(Sensitivity)}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$


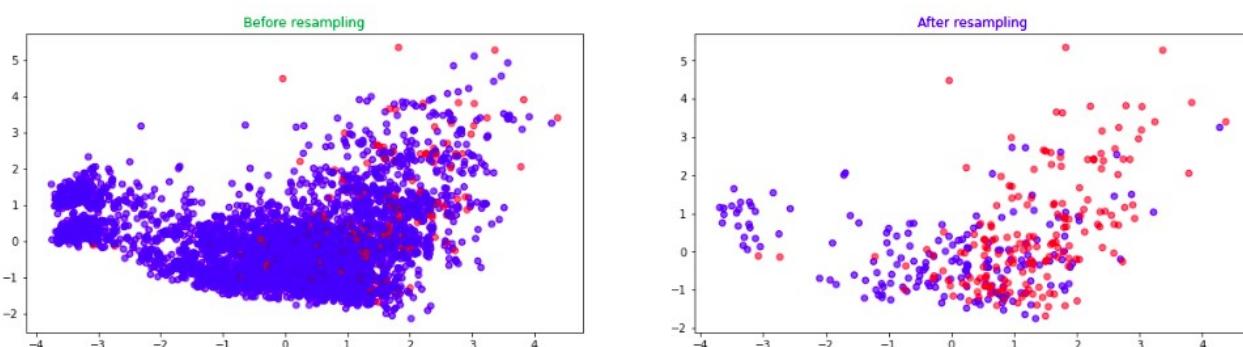
Fraction of positives predicted correctly

Classification report for model DecisionTreeClassifier on test data						
	pre	rec	spe	f1	geo	iba
0	0.98	0.73	0.71	0.84	0.72	
1	0.12	0.71	0.73	0.21	0.72	
avg / total	0.94	0.73	0.71	0.81	0.72	

Confusion matrix for model DecisionTreeClassifier on test data		
0	1	
0	1071	387
1	22	53

Ovaj model daje solidne rezultate, vidimo da smo smanjili broj FN sa 60 na 22 I povećali broj TP, međutim povećali smo broj FP instanci.

Sada ću primeniti jednu od tehnika koja uklanja problem nebalansiranosti. Ja sam izabrala RandomUnderSampler tehniku, a motivacija za to mi je bila to što ova tehnika ne dira manjinsku klasu I ne unosi dodatni šum u podatke.



```
'class_weight': {0: 1, 1: 40}, 'criterion': 'gini', 'max_depth': 10}
Classification report for model DecisionTreeClassifier on training data
```

	pre	rec	spe	f1	geo	iba
0	1.00	0.76	1.00	0.87	0.87	0.75
1	0.81	1.00	0.76	0.89	0.87	0.78
avg / total	0.90	0.88	0.88	0.88	0.87	0.76

```
Confusion matrix for model DecisionTreeClassifier on training data
```

	0	1
0	133	41
1	0	174

Nakod toga sam trenirala sa istip parametrima samo nad balansiranim podacima I dobila znatno bolje rezultate.Smanjili smo btoj FN instanci, preciznost je ostala ista dok se recall smanjio, zbog manjeg broja TN instanci.

```
Classification report for model DecisionTreeClassifier on test data
```

	pre	rec	spe	f1	geo
0	0	0.99	0.59	0.83	0.74
1	1	0.09	0.83	0.59	0.17
avg / total	0.94	0.60	0.82	0.71	0.70

```
Confusion matrix for model DecisionTreeClassifier on test data
```

	0	1
0	858	600
1	13	62

```
Classification report for model Pipeline on trening data
```

	pre	rec	spe	f1
0	1.00	0.50	1.00	0.67
1	0.67	1.00	0.50	0.80
avg / total	0.83	0.75	0.75	0.73

```
Confusion matrix for model Pipeline on trening data
```

	0	1
0	87	87
1	0	174

```
Classification report for model Pipeline on test data
```

	pre	rec	spe	f1
0	0.99	0.39	0.96	0.56
1	0.08	0.96	0.39	0.14
avg / total	0.95	0.42	0.93	0.54

```
Confusion matrix for model Pipeline on test data
```

	0	1
0	574	884
1	3	72

Poslednji pokušaj za stabla odlučivanja je korišćenje PCA I primena na redukovane podatke, ovo nam je I dalo najbolji rezultat na test skupu što se tiće FN instanci.

Vidimo da se broj FP instanci znatno povećao I smanjio broj TN instanci pa je I recall znatno manji.

Balansirana slučajna šuma

Za rešavanje nebalansiranosti možemo koristiti I balansirane slučajne šume.

To je jedna od ansabl metoda I očekujemo da će dati dobre rezultete jer on kombinuje skup modela koji rešavaju određeni problem.

Balansirane slučajne šume vrše slučajan under-sampling svakog uzorka koji se koristi za formiranje slučajne šume.

Prvo smo napravili model bez podešavanja parametra I dobili sledeće rezultate.

Classification report for model BalancedRandomForestClassifier on trening data						
	pre	rec	spe	f1	geo	iba
0	1.00	0.71	1.00	0.83	0.84	0.69
1	0.15	1.00	0.71	0.26	0.84	0.73
avg / total	0.96	0.73	0.99	0.80	0.84	0.69

Confusion matrix for model BalancedRandomForestClassifier on trening data						
	0	1				
0	2425	977				
1	0	174				

Classification report for model BalancedRandomForestClassifier on test data						
	pre	rec	spe	f1	geo	iba
0	0.99	0.70	0.85	0.82	0.77	0.58
1	0.13	0.85	0.70	0.22	0.77	0.60
avg / total	0.95	0.70	0.85	0.79	0.77	0.59

Confusion matrix for model BalancedRandomForestClassifier on test data						
	0	1				
0	1014	444				
1	11	64				

Dobijamo solidne rezultate I na trening I na tetst skupu, a sada ćemo pokušati da podešavanjem parametara dobijemo još bolje.

Sa parametrima koji su dati na slici dobijamo bolji model, tj model koji daje manji broj FN instanci, preciznost se nije smanjila, dok su se ukupne vrednosti drugih mera malo smanjile.

Classification report for model BalancedRandomForestClassifier on trening data							Classification report for model BalancedRandomForestClassifier on test data							
	pre	rec	spe	f1	geo	iba		pre	rec	spe	f1	geo	iba	sup
0	1.00	0.56	1.00	0.72	0.75	0.53	⋮	0	1.00	0.56	0.95	0.71	0.73	0.51
1	0.10	1.00	0.56	0.19	0.75	0.58	⋮	1	0.10	0.95	0.56	0.18	0.73	0.55
avg / total	0.96	0.58	0.98	0.69	0.75	0.54	⋮	avg / total	0.95	0.57	0.93	0.69	0.73	0.51

Confusion matrix for model BalancedRandomForestClassifier on trening data						
	0	1				
0	1903	1499				
1	0	174				

Confusion matrix for model BalancedRandomForestClassifier on test data						
	0	1				
0	810	648				
1	4	71				

{'class_weight': {0: 1, 1: 40}, 'criterion': 'gini', 'max_depth': 10, 'n_estimators': 120}
--

K najbližih suseda

Što se tiče ovog algoritma pokušali smo da napravimo dobar model sa podešavanjem parametara I korišćenjem PCA radi smanjenja dimenzionalnosti podataka. Trenirali smo podatke nad kojim je primenjena RandomUnderSampler tehnika I dobili sledeće rezultate.

Nismo zadovoljni modelom koji smo dobili jer imamo veliki broj FN instanci, a pritom se nije smanjio ni broj FP.

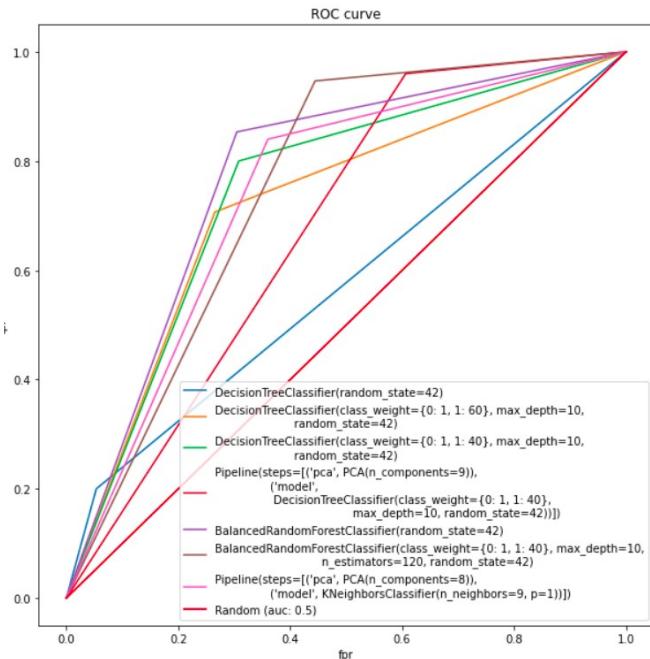
Parametri koji su korišćeni kod pravljenja modela su sledeći:

```
'pca_n_components' : [ 3,4,5,6,7,8,9],  
'model_n_neighbors' : list(np.arange(2, 15, 1)),  
'model_weights' : ['uniform', 'distance'],  
'model_p' : [1,2]
```

```
{'model_n_neighbors': 9, 'model_p': 1, 'model_weights': 'uniform', 'pca_n_components': 8}  
Classification report for model Pipeline on trening data  
-----  
          pre    rec    spe     f1     geo     iba    sup  
0      0.81   0.72   0.83   0.77   0.78   0.60   174  
1      0.75   0.83   0.72   0.79   0.78   0.61   174  
avg / total  0.78   0.78   0.78   0.78   0.78   0.60   348  
-----  
Confusion matrix for model Pipeline on trening data  
-----  
      0   1  
0  126  48  
1   29  145  
-----  
Classification report for model Pipeline on test data  
-----  
          pre    rec    spe     f1     geo     iba    sup  
0      0.99   0.64   0.84   0.78   0.73   0.53  1458  
1      0.11   0.84   0.64   0.19   0.73   0.55    75  
avg / total  0.94   0.65   0.83   0.75   0.73   0.53  1533  
-----  
Confusion matrix for model Pipeline on test data  
-----  
      0   1  
0  933  525  
1   12   63
```

Poređenje modela

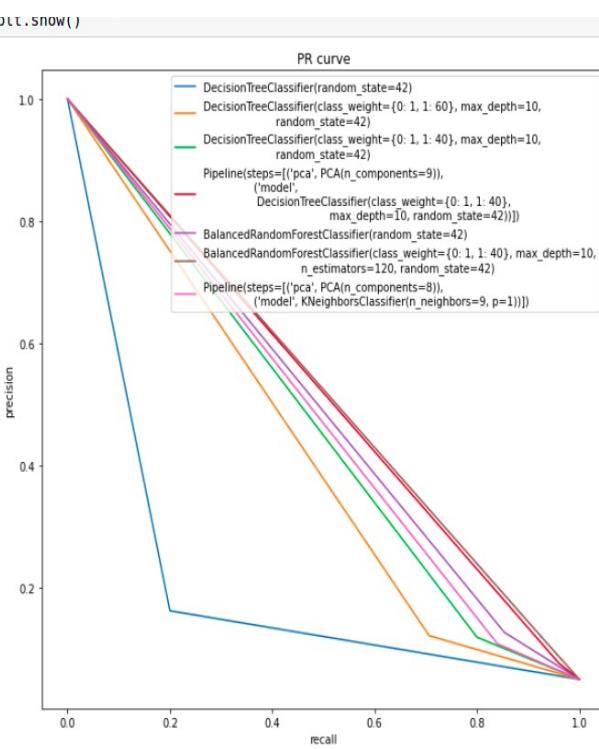
Za poređenje dobijenih modela koristila sam PR krivu, ROC krivu koja daje lažno optimistične podatke, a pratili smo I F1 score I recall score.



ROC kriva nam govori koliki nam je True Positive Rate I False Positive Rate. Na osnovu ovoga možemo zaključiti da nam prvi model daje najgore rezultate, što se i očekivalo, a da se najbolje pokazao model gde smo vršili RandomUnderSampler tehniku + PCA + DecisionTreeClasifier , kao i Balansirana slučajna šuma sa podešavanjem parametara

Što se tiče PR krive, ona nam daje realniju sliku naših modela.

Možemo zaključiti da ni jedan od naših modela nije idealan i da ne bi mogli da ih koristimo u realnim situacijama, ali opet možemo zaključiti kao kod ROC krive da se ova dva modela najbolje ponašaju, kao i da nam senzitivnost ima dobre vrednosti kod većinu modela.



Ovde možemo da vidimo da sa povećanjem recall score (udziv) umanjujemo F1 score, jer utičemo negativno na preciznost, pa treba naći balans I izabratи model kod koga I senzitivnost I F1 score imaju prihvatljive vrednosti.

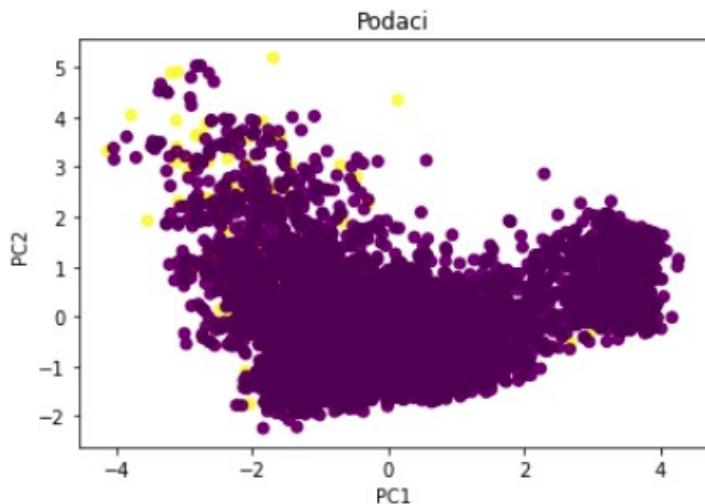
```
DecisionTreeClassifier(random_state=42)
F1 score :
0.9099804305283757
recall score :
0.2
DecisionTreeClassifier(class_weight={0: 1, 1: 60}, max_depth=10,
                      random_state=42)
F1 score :
0.7332028701891716
recall score :
0.7066666666666667
DecisionTreeClassifier(class_weight={0: 1, 1: 40}, max_depth=10,
                      random_state=42)
F1 score :
0.6973255055446836
recall score :
0.8
Pipeline(steps=[('pca', PCA(n_components=9)),
                ('model',
                 DecisionTreeClassifier(class_weight={0: 1, 1: 40},
                                         max_depth=10, random_state=42))])
F1 score :
0.42139595564253096
recall score :
0.96
BalancedRandomForestClassifier(random_state=42)
F1 score :
0.7031963470319634
recall score :
0.8533333333333334
BalancedRandomForestClassifier(class_weight={0: 1, 1: 40}, max_depth=10,
                               n_estimators=120, random_state=42)
F1 score :
0.5746901500326158
recall score :
0.9466666666666667
Pipeline(steps=[('pca', PCA(n_components=8)),
                ('model', KNeighborsClassifier(n_neighbors=9, p=1))])
F1 score :
0.649706457925636
recall score :
0.84
```

Klasterovanje

Što se tiče klasterovanja ne možemo puno očekivati sa obzirom da je naš skup za klasifikaciju I znamo da nam ciljni atribut ima dve klase, a takođe nam je I velika nebalansiranost I podaci nam nisu lepo rasporedjeni.

Ono što će ovde uraditi jeste da pokušam da demonstriramo neke od algoritama I da vidimo koji će se najbolje pokazati.

Prvo ćemo vizualizovati naše podatke da vidimo kako to izgleda, primenićemo PCA kako bi smanjili dimenzionalnost I predstavili podatke u 2D.



Na osnovu ovog prikaza možemo pretpostaviti da nam ni jedan algoritam neće dati sjajne rezultate, ali svakako ćemo pokušati.

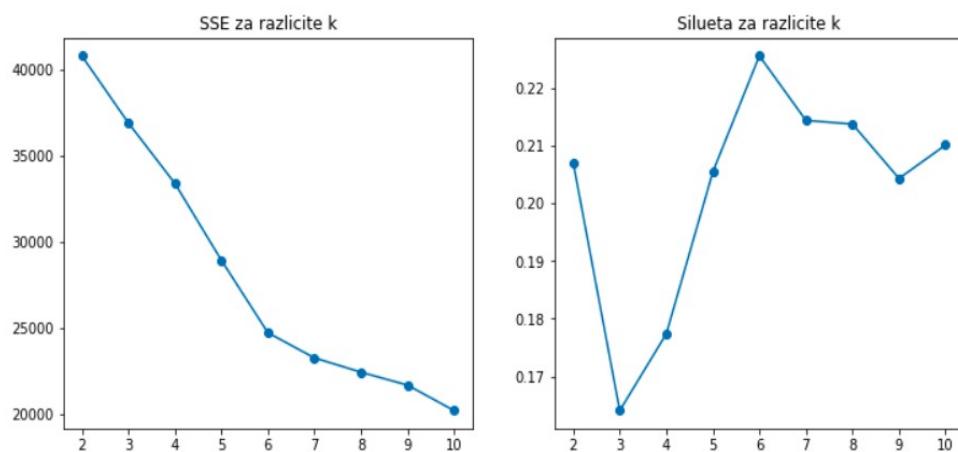
Preprocesiranje za klasterovanje

Što se tiče preprocesiranja, smanjili smo broj autlajera jer oni utiču na klasterovanje I izvršili normalizaciju korišćenjem StandardScaler()-a jer radimo sa distancama I bitno nam je da sve vrednosti atributa budu u istom opsegu.

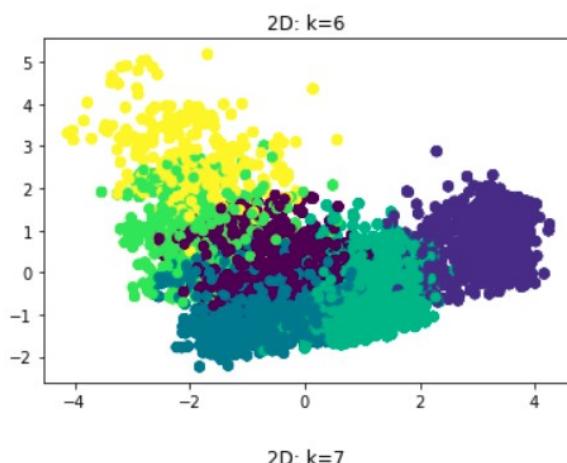
Ovde nema potrebe podeliti podatke na trening I test, jer nemamo tačno zadato rešenje problema i nemamo šta da testiramo.

K-means

Pokretali smo K-means više puta sa različitim brojem K I pratili SSE (srednje kvadratna greška) kao I silhouette score. Stavili smo da se izvrši 500 iteracija, a za inicijalno formiranje centroida smo stavili ‘k-means++’ tehniku, jer ona koristi naprednije tehnike za biranje centroida I nadamo se da će nam dati bolje rezultate.



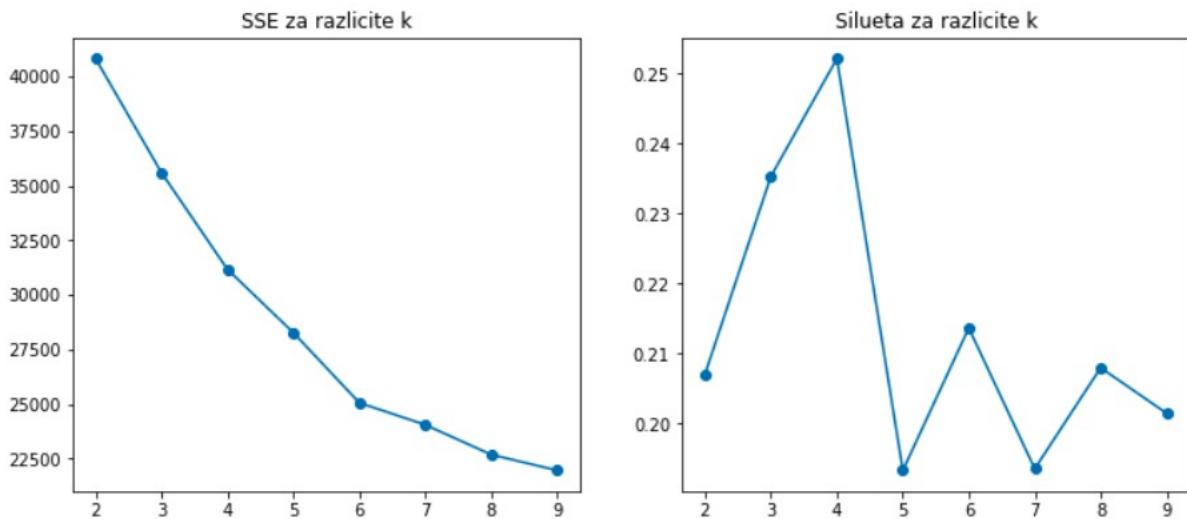
Na osnovu metrika za procenu kvaliteta K-meansa trebalo bi da izaverano podelu na 76klastera, što je suprotno onome što mi znamo, ali ipak ćemo prikazati kako bi ta podela izgledala.



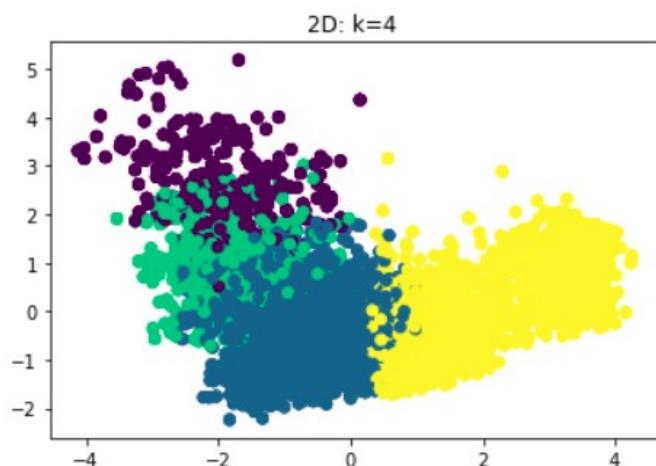
Bisecting Kmeans

Bisecting K-means algoritam se zasniva na ideji: Prvo podeliti instance u 2 klastera, zatim izabrati jedan od postojećih i podeliti ga na 2 klastera. Proces se ponavlja dok se ne formira klastera.

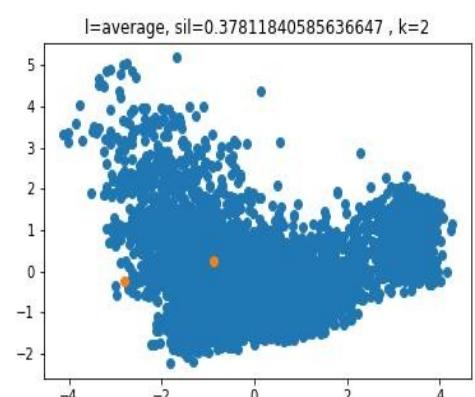
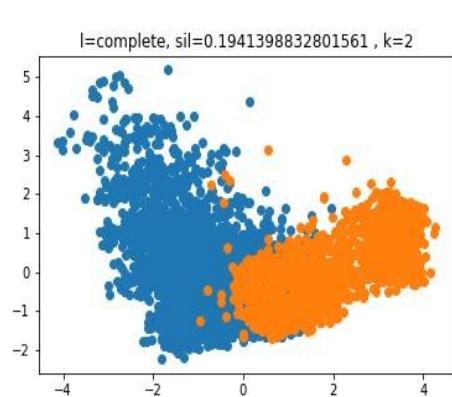
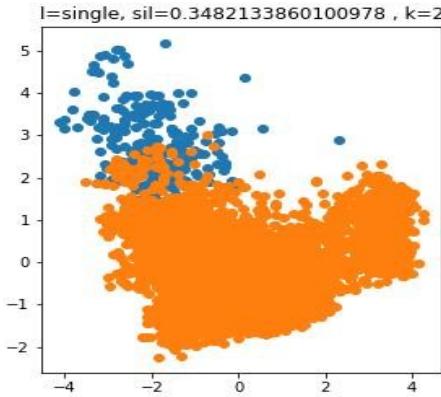
Izbor klastera za podelu se može izvršiti na više načina: možemo izabrati najveći klaster, klaster sa najvećom SSE ili kriterijum koji se zasniva na SSE i veličini.
Mi smo izabrali da se deli klaster sa najvećim SSE.



Ovde vidimo da je silueta najveća za $K=4$, takođe vidimo da se vrednost siluete malo povećala u odnosu na vrednosti kod predhodnog algoritma, ali će nam greška biti veća.



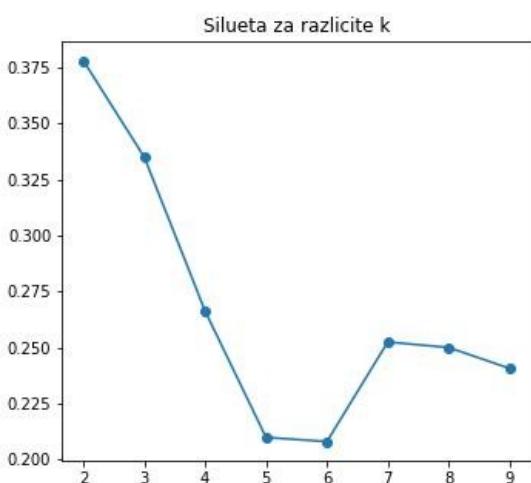
Hijerarhijsko klasterovanje



Kod hijerarhijskog klasterovanja korišćenjem “AgglomerativeClustering” funkciju, dobijamo klaster skroz drugačijeg izgleda. Ako fiksiramo k na 2,a menjamo metodu po kojoj se vrši spajanje klastera, dobijemo klastera kao na slici.

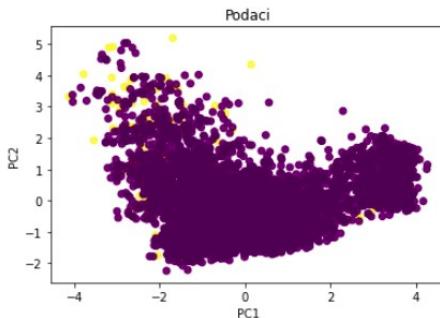
Interesantno je to što kada koristimo tehniku koja spaja klastere po najdaljoj udaljenosti izmedju elemenata dobijemo da su više podataka u klasi 1, nego u 0, što nam se do sada nije dešavalo.

Mi ćemo izabrati metodu koja kosisti prosečno rastojanje, jer nam je tu I najveći siluet koeficijent, a I najviše liči na naš ciljni skup vrednosti.

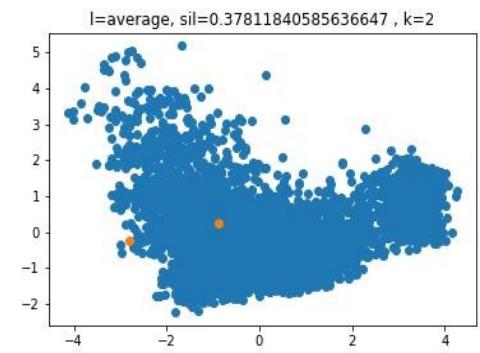
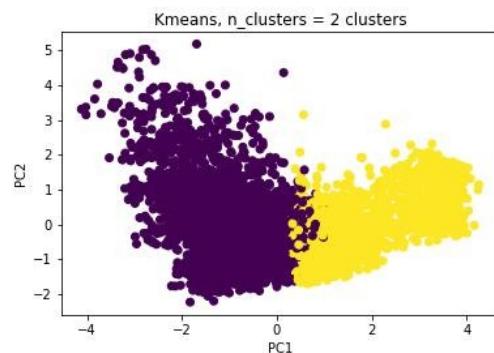
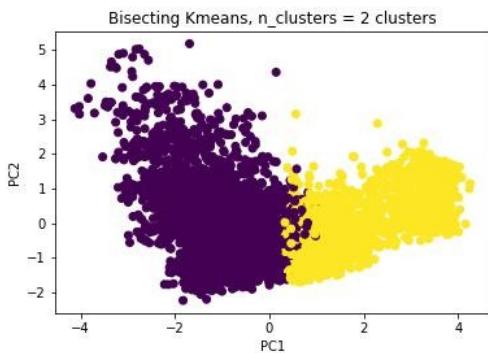


Ako koristimo “avarage”, a menjamo k, dobijamo ovakav grafik za siluetu koja nam govori da je najbolje izabrat 2 klastera.

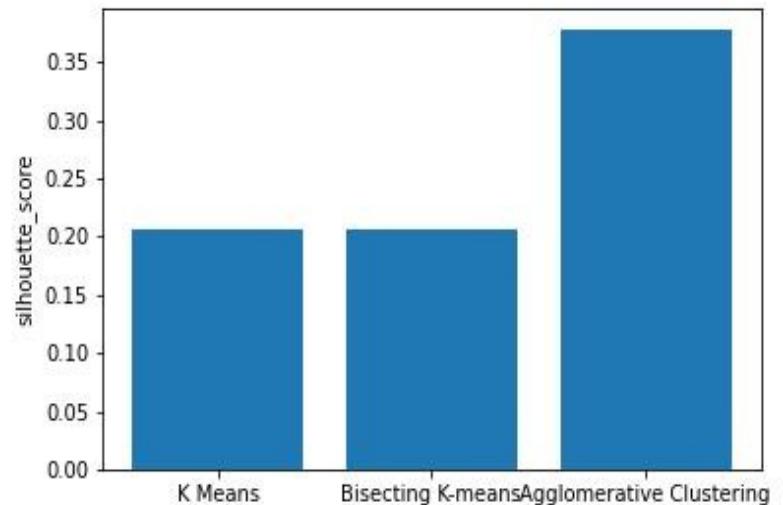
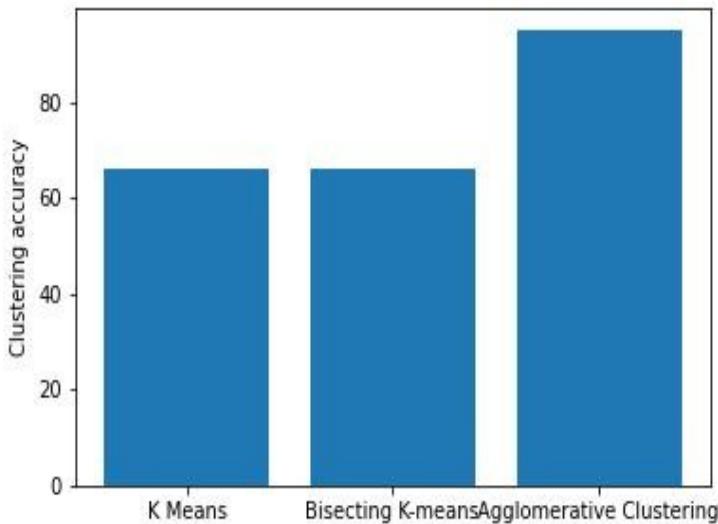
Poređenje modela



Kao što sam već napomenula, ne možemo očekivati neke sjajne rezultate, pogovoto jer je velika nebalansiranost, a podaci su nam jako gusto rasporedjeni. Vidmo da nam K-means I Bisecting K-means daju gotovo identične klasterove, dok se hijerarhijsko klasterovanje razlikuje.



Ukoliko jedan klaster koji nam algoritmi vrate posmatramo kao klasu 0, a drugi kao 1, možemo da napravimo funkciju koja nam pokazuje koliki je procenat poklapanja. Procenat poklapanja ne daje toliko loše rezultate, pogotovo AgglomerativeClustering model, ali to je zbog toga što većina podataka pripada klasi 0 pa su oni donekle dobro poklopljeni, te ne treba da nas ova statistika zavarava da je naše klasterovanje uspešno.



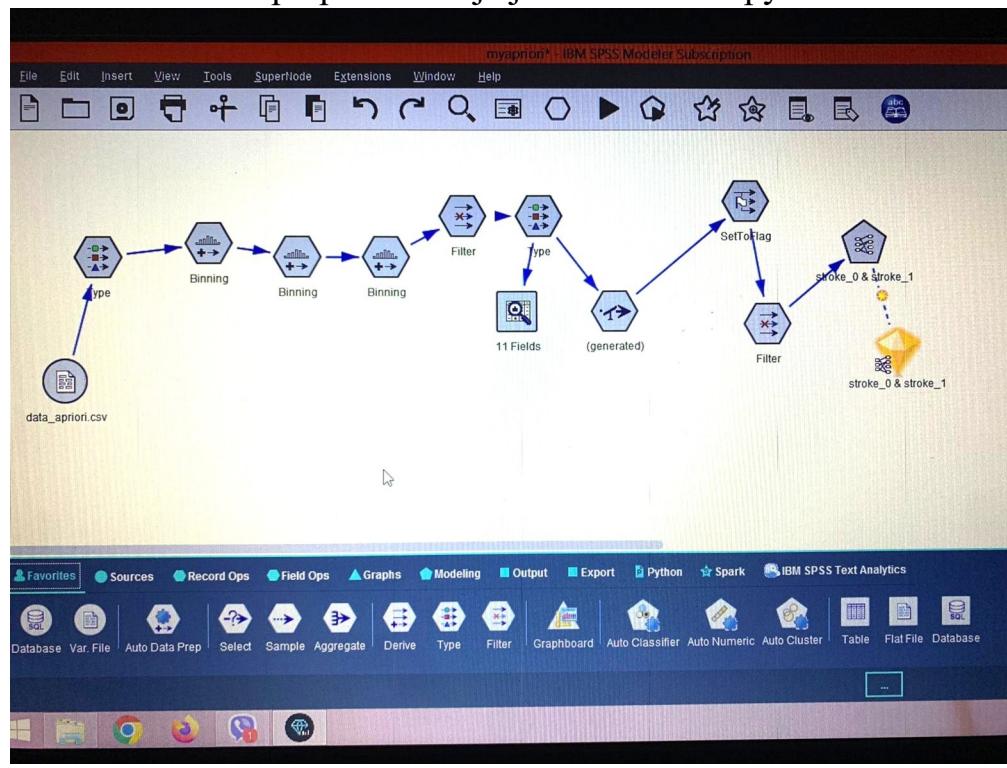
Pravila pridruživanja

Pravila pridruživanja koristimo kako bi uočili neke pravilnosti, tj grupacije vrednosti atributa koje se javljaju u paru.

Pravila su nam oblika telo → glava, gde ćemo mi za telo izabratati naš ciljni atribut I videti najčešće parove koje se javljaju kada nam je Stroke 1 I kad nam je Stroke 0. Značenje je sledeće, ako se u transakciji pojave stavke iz tela, verovatno će se pojaviti I stavke iz glave.

Preprocesiranje

Delimično preprocesiranje je izvršeno u Jupyter Notebook-u, u delu Analiza I



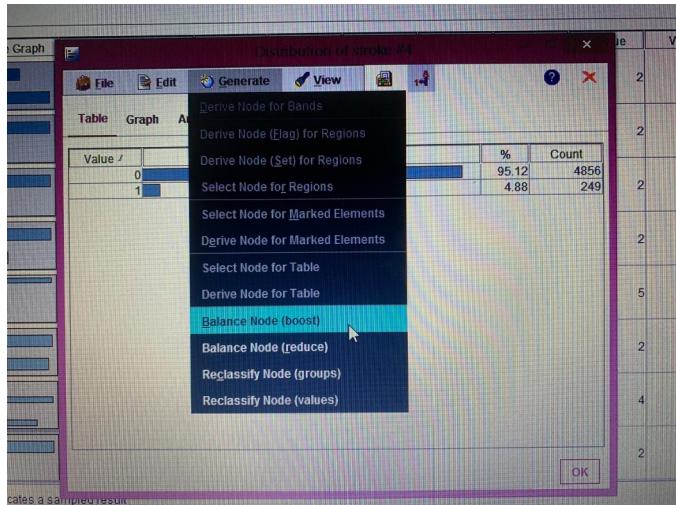
preprocesiranje I te podatke smo uneli, međutim ovde je potrebno dodatno izmeniti podatke kako bi primenili Apriori algoritam.

(Age, BMI, avg_glucose_level).

Sve atribute smo podelili na 4 jednakaka intervala, pomoću opcije **Binning**.

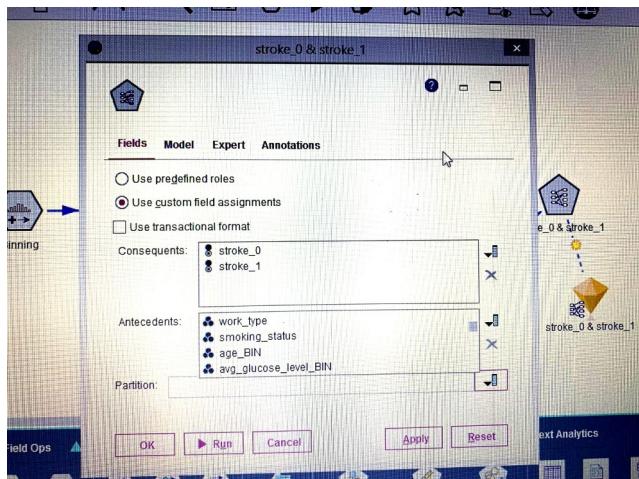
Nakon učitavanja vrednosti I postavljanje njihovih uloga na Both, izvršili smo binovanje neprekidnih atributa

Nakon toga smo izvršili balansiranje klasa naše ciljne promenjive, koristeći opciju Balance Node(boost).



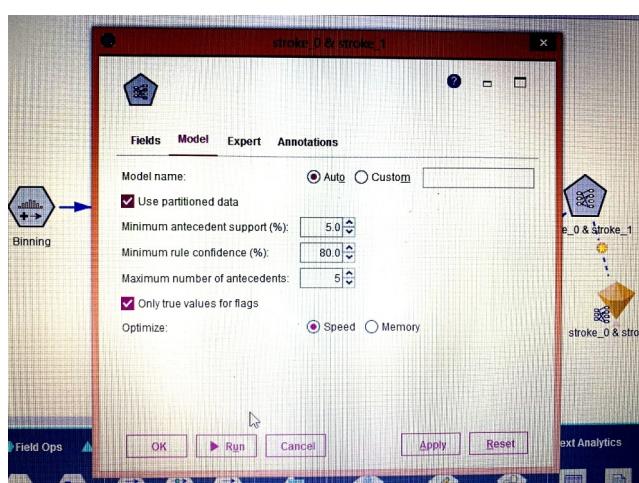
Apriori u SPSS Modeler-u

Nakon preprocesiranja, pokrenuli smo apriori algoritam sa sledećim opcijama:



U listu Consequents se dodaju stavke koje mogu da se pojave u glavi pravila, a u listu Antecedents se dodaju stavke koje mogu da se pojave u telu pravila, mi smo u glavi stavili Stroke ,a u telu sve ostale attribute.

Za minimalnu podršku tela pravila smo stavili 5% , dok smo za pouzdanost pravila stavili 80% Only true values for flags - uzeti u obzir samo vrednosti tacno u binarnim atributima, to smo takođe izabrali.



Consequent	Antecedent	Support %	Confidence %	Lift
stroke_1	near_disease_0 heart_disease_1 Residence_type_0 bmi_BIN = 3 ever_married_0	5.204	84.752	1.696
stroke_0	ever_married_1 avg_glucose_level_BIN = 2 Residence_type_1	5.286	84.795	1.695
stroke_0	age_BIN = 2 bmi_BIN = 3 ever_married_0 hypertension_0	5.348	84.778	1.694
stroke_0	ever_married_1 Residence_type_1	10.479	84.759	1.694
stroke_0	age_BIN = 2 bmi_BIN = 3 ever_married_0 hypertension_0 heart_disease_0	5.337	84.749	1.694
stroke_1	avg_glucose_level_BIN = 4 age_BIN = 4 Residence_type_0 ever_married_0	7.573	84.626	1.694

Dobijam pravila u obliku u kom sam želela da budu, gde mi je u glavi pravila Stroke, a u telu vrednosti ostalih atributa. Vidimo da smo dobili 379 pravila, a da među njima ima I onih kod kojih je predviđen moždani udar I onih kod kojih nije.

Evo I objašnjenje nekih od pravila:

Ako je vrednost za stroke 0, onda je verovatno u pitanju čovek koji ima izmedju 20 I 40 godina, nije oženjen, nema hipertenziju (visok krvi pritisak), bmi mu je srednje vrednosti I nema bolesti srca.

Ako je vrednost za stroke 1, onda je verovatno u pitanju čovek koji ima visok prosečni nivo glukoze, ima preko 60 godina, živi u urbanoj sredini I nije oženjen.

Model Settings Summary Annotations

- Collapse All + Expand All

Analysis

- Number of Rules: 379
- Number of Valid Transactions: 9,705
- Minimum Support: 5.008%
- Maximum Support: 20.948%
- Minimum Confidence: 80.034%
- Maximum Confidence: 100.0%
- Minimum Lift: 1.602%
- Maximum Lift: 1.999%
- Minimum Deployability: 0.0%
- Maximum Deployability: 3.586%
- Minimum Rule Support: 4.019%
- Maximum Rule Support: 17.362%

Fields

Build Settings

Training Summary

Ovde možemo videti vrednosti nekih od statistika koji su nam od značaja.

Najveći lift nam je 1.999, a najmanji 1.602, što nam sugeriše da su nam sva pravila koja smo izdvojili značajna. Lift uzima u obzir I pouzdanosti pravila,a I podršku desne strane pravila, te nam daje bolje uvid o značajnosti pravila od obične podrške I pouzdanosti.

Zaključak

Na osnovu dosadašnjeg rada mogu zaključiti da na dobijanje modela najviše utiče postavljanje odgovarajućih hiperparametara, odabir odgovarajućih mera koje pratimo, kao i preprocesiranje podataka.

Treba biti oprezan pri radu sa medicinskim podacima i dobro analizirati ulazne informacije, te bi stručnjak u ovoj oblasti mogao bolje proceniti da li nam je možda još neki deo informacija bio suvišan i olakšao nam poso kada bi ga izbacili.

Po mom mišljenju ni jedan od dobijenih modela za klasifikaciju ne bi mogao da se koristi u praksi, iako smo dobili mali broj lažno negativnih instanci, to nam je uticalo na povećanje lažno pozitivnih, što nam takođe ne ide u korist.

Takođe, ono što se može zaključiti jeste da vizuelizacija znatno olakšava razumevanje podataka i pomaže u donošenju odluka o odabiru najboljih modela.