Orbit Classification For Prediction / NASA

Istraživanje podataka I

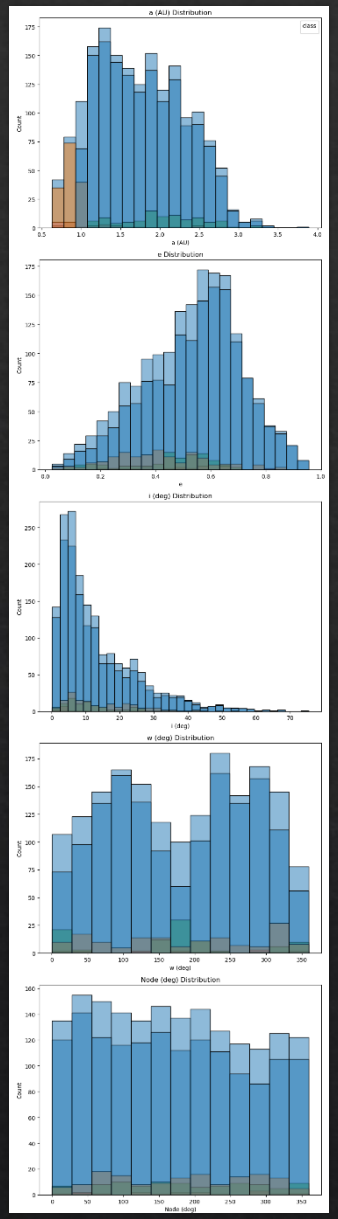
Maja Milenković

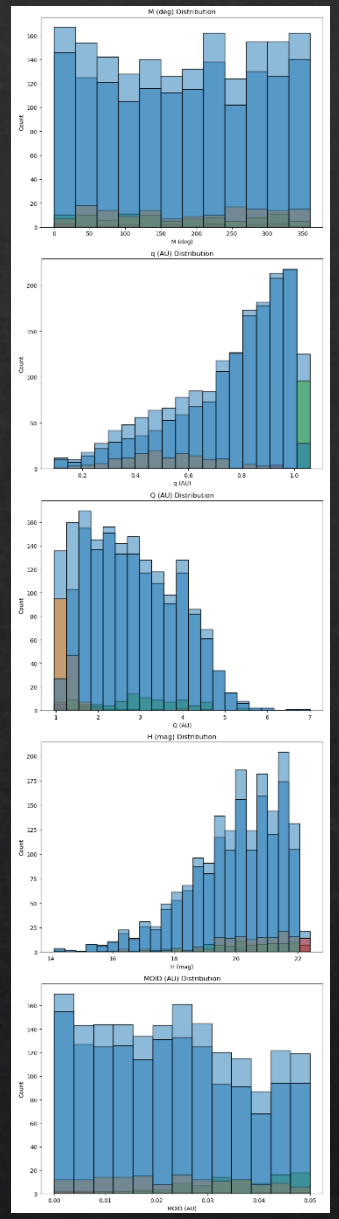
Opis skupa podataka

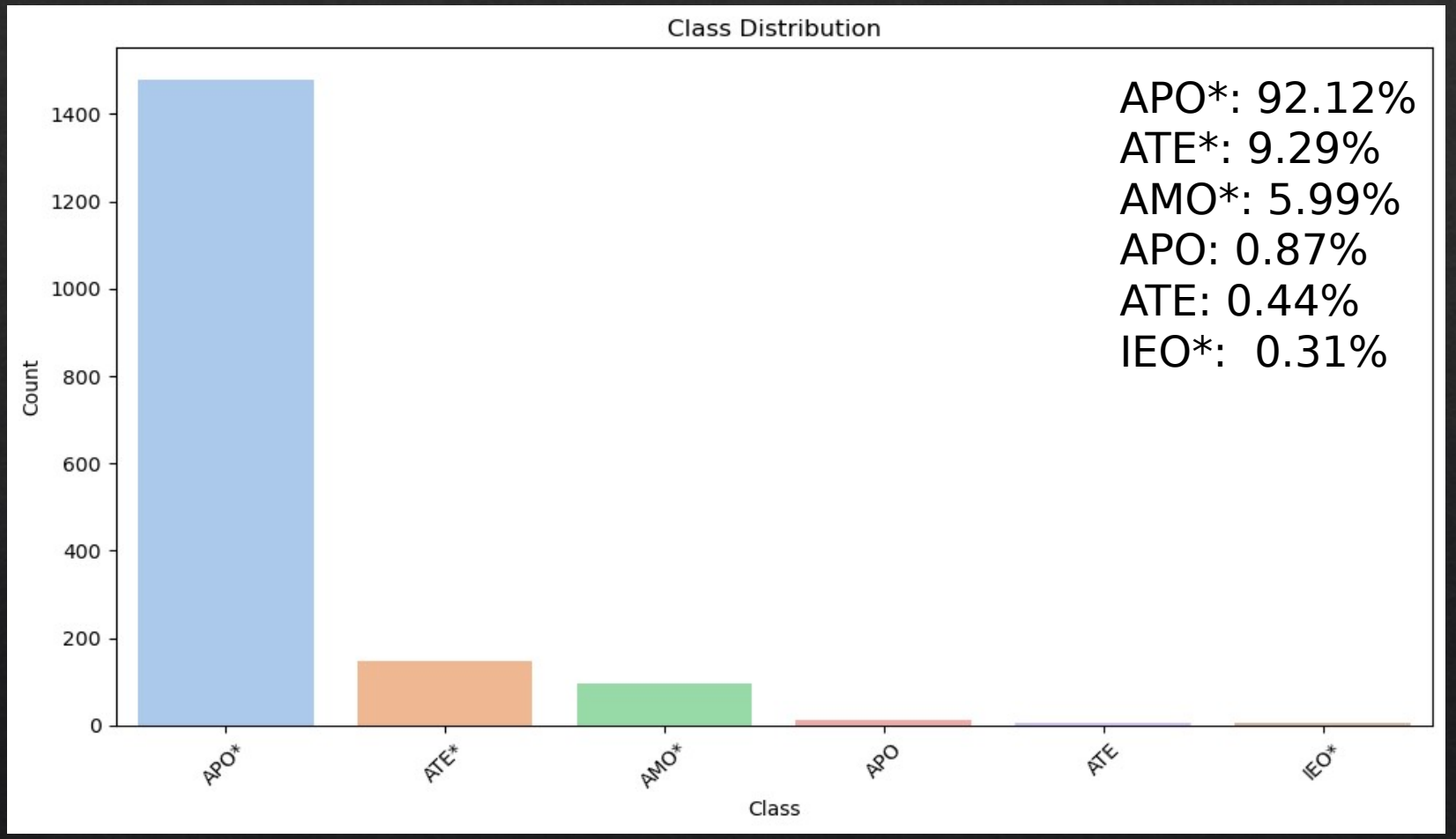
- 1748 instanci
- 11 numeričkih atributa i 1 kategorički

	a (AU)	e	i (deg)	w (deg)	Node (deg)	M (deg)	q (AU)	Q (AU)	P (yr)	H (mag)	MOID (AU)	class
0	1.078066	0.826854	22.825495	31.382966	88.010681	215.528772	0.1867	1.97	1.12	16.90	0.034507	APO*
1	1.245304	0.335342	13.337482	276.893024	337.207958	104.155607	0.8277	1.66	1.39	15.60	0.030669	APO*
2	1.470264	0.559922	6.352995	285.852564	35.736768	174.626213	0.6470	2.29	1.78	16.25	0.025795	APO*
3	1.776025	0.650141	39.832538	267.791993	356.903343	173.188556	0.6214	2.93	2.37	15.20	0.003551	APO*
4	1.874123	0.764602	1.326399	43.388048	349.694944	235.158622	0.4412	3.31	2.57	18.80	0.011645	APO*

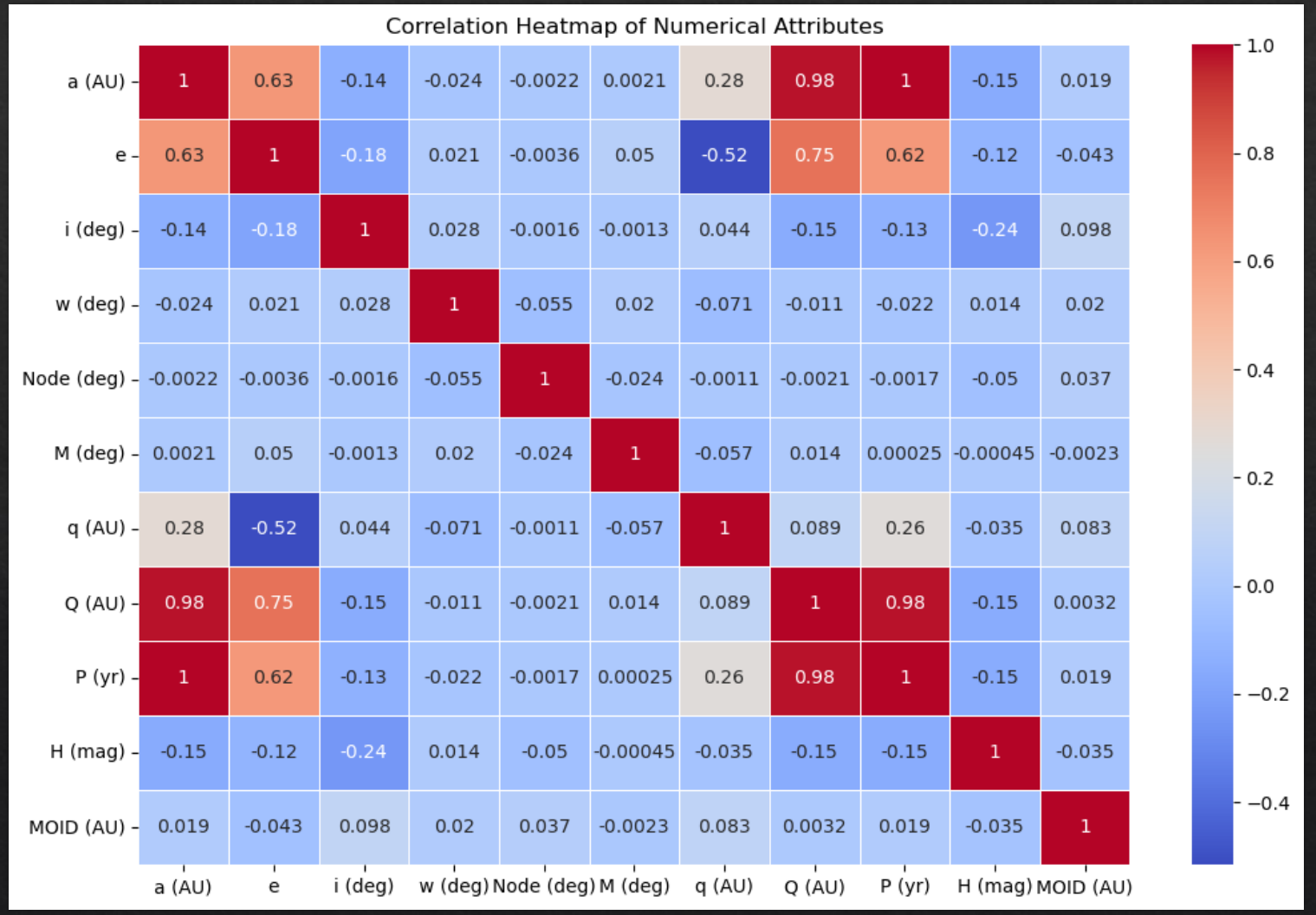
Histogrami atributa







Matrica korelacije

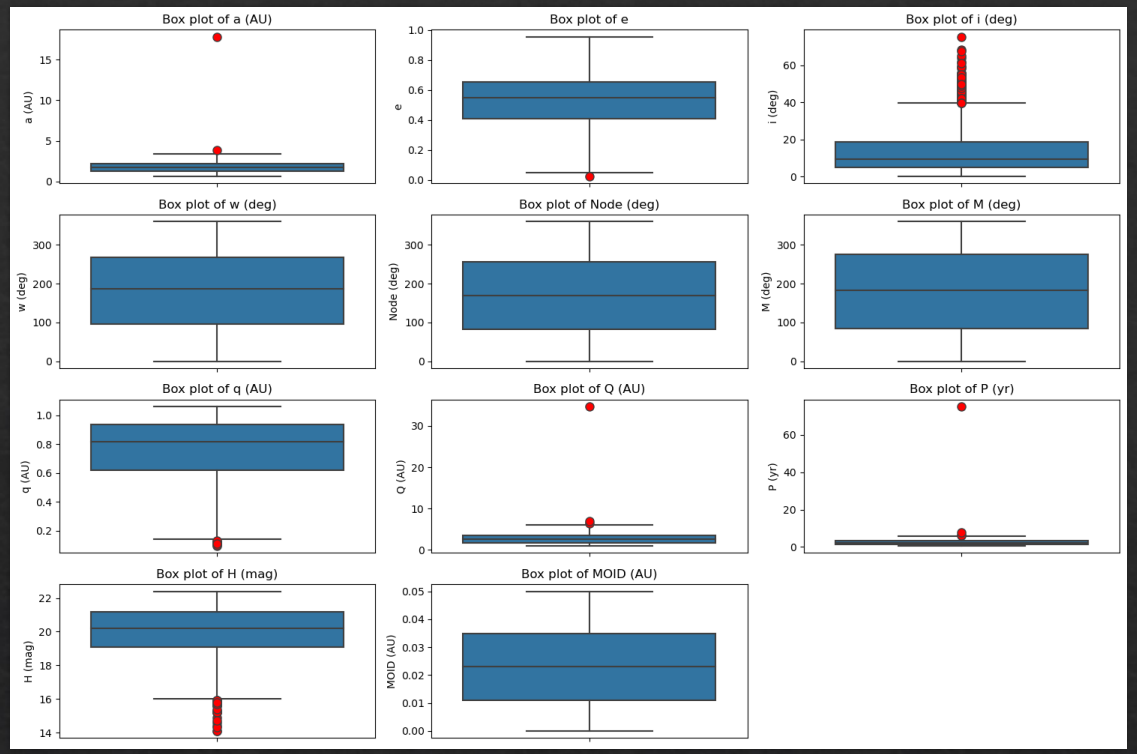


Nedostajući podaci i elementi van granica

```
data.isna().sum()
```

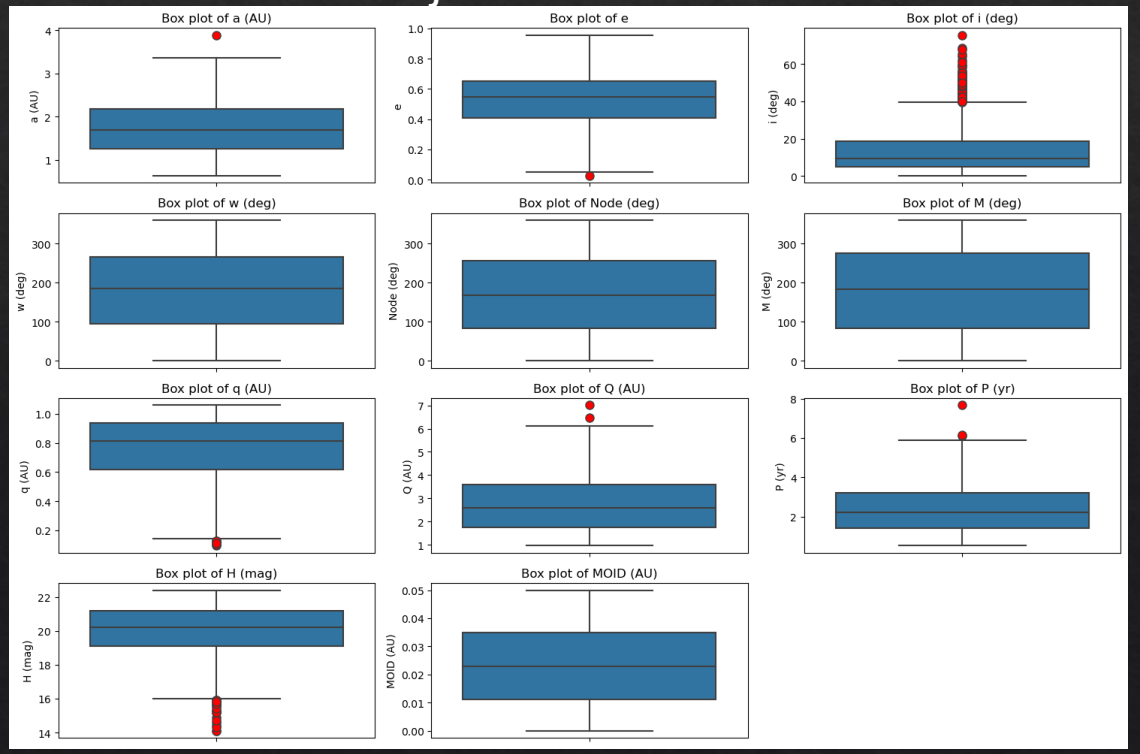
```
a (AU)          0
e               0
i (deg)         0
w (deg)         0
Node (deg)      0
M (deg)         0
q (AU)          0
Q (AU)          0
P (yr)          0
H (mag)         0
MOID (AU)       0
class           0
dtype: int64
```

	lower	min	num_lower	upper	max	num_upper	percantage
a (AU)	-0.104511	0.635223	0	3.546646	3.888719	1	0
e	0.041192	0.025425	1	1.020715	0.956042	0	0
i (deg)	-15.987849	0.146084	0	39.647466	75.412403	65	4
w (deg)	-160.797589	0.521838	0	523.014817	359.662669	0	0
Node (deg)	-178.426441	0.136042	0	517.340838	359.854602	0	0
M (deg)	-204.964619	0.052165	0	564.446926	359.825201	0	0
q (AU)	0.138375	0.092800	10	1.414575	1.060100	0	1
Q (AU)	-0.957500	0.960000	0	6.302500	7.010000	2	0
H (mag)	15.950000	14.100000	25	24.350000	22.400000	0	1
MOID (AU)	-0.024570	0.000010	0	0.070444	0.049987	0	0



Pre izbacivanja autlajera

Posle izbacivanja autlajera



Klasifikacija

Stabla odlučivanja

```
params = {'criterion': ['gini', 'entropy'],
          'max_depth': range(2,10),
          'min_samples_leaf' : [2,3,4,5],
         }
```

Parametri prosleđeni GridSearch-u

“criterion”: određuje kriterijum koji se koristi za merenje kvaliteta podela u svakom čvoru stabla. Dva najčešće korišćena kriterijuma su "gini" i "entropy"
“max_depth”: kontroliše maksimalnu dubinu stabla. Dubina stabla odnosi se na broj čvorova od korena do lista i time se može sprečiti preprilagođavanje modela na trening podacima
“min_samples_leaf”: određuje minimalni broj instanci koje moraju biti u svakom listu stable i može pomoći u sprečavanju preprilagođavanja

```
estimator = GridSearchCV(DecisionTreeClassifier(), param_grid = params, scoring = 'accuracy', cv = 4, verbose = 4)
```

```
estimator.fit(X_train, y_train)
```

```
Fitting 4 folds for each of 64 candidates, totalling 256 fits
```

```
estimator.best_estimator_
```

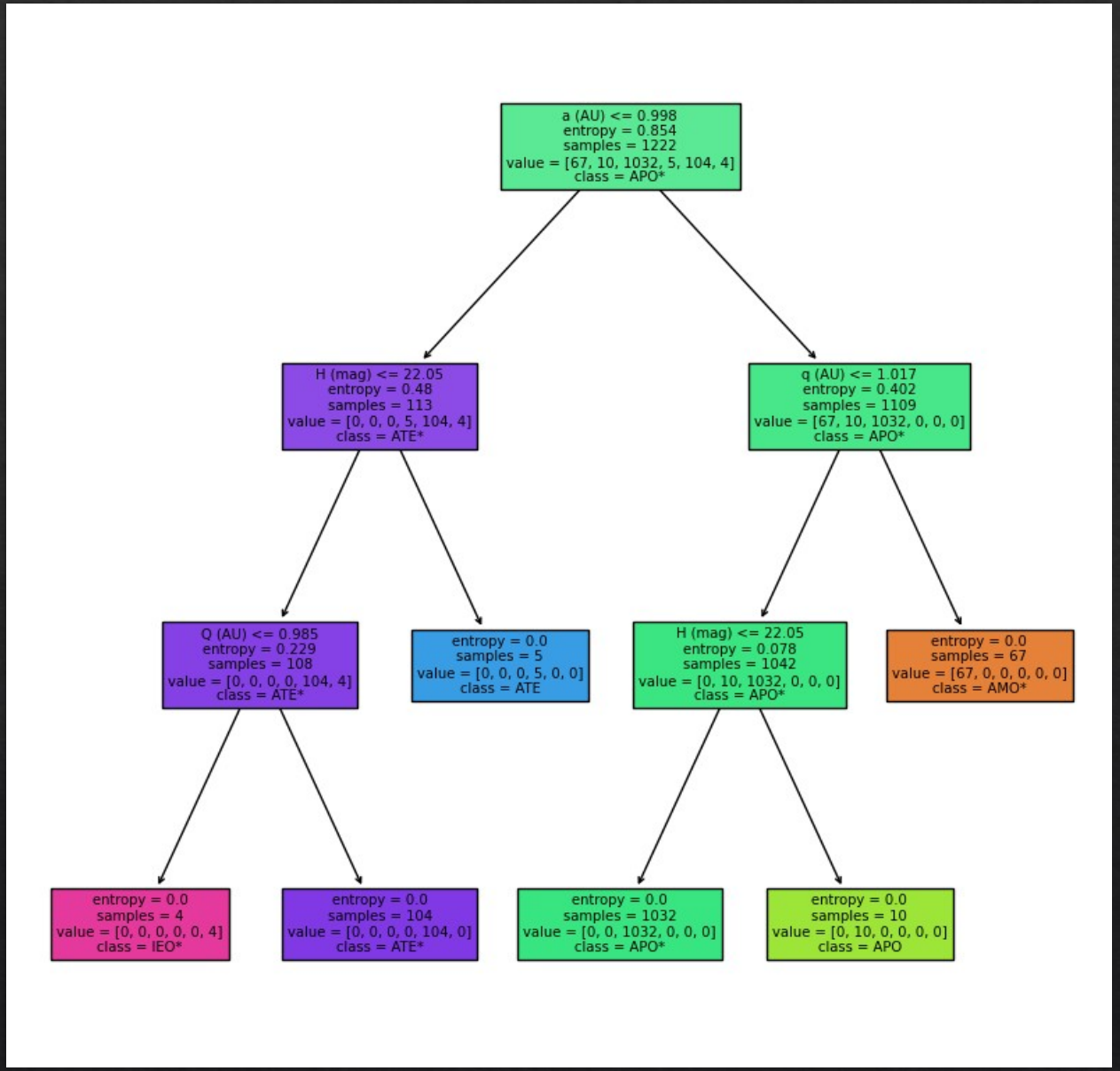
```
DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=2)
```

```
model.feature_importances_
```

```
array([0.52212145, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.36915616, 0.02280428, 0.08591811, 0.        ])
```

```
confusion_matrix(y_train, y_train_pred)
```

```
array([[  67,    0,    0,    0,    0,    0],
       [   0,   10,    0,    0,    0,    0],
       [   0,    0, 1032,    0,    0,    0],
       [   0,    0,    0,    5,    0,    0],
       [   0,    0,    0,    0,  104,    0],
       [   0,    0,    0,    0,    0,    4]])
```


a (AU) <= 0.998
entropy = 0.854
samples = 1222
value = [67, 10, 1032, 5, 104, 4]
class = APO*
H (mag) <= 22.05
entropy = 0.48
samples = 113
value = [0, 0, 0, 5, 104, 4]
class = ATE*
q (AU) <= 1.017
entropy = 0.402
samples = 1109
value = [67, 10, 1032, 0, 0, 0]
class = APO*
Q (AU) <= 0.985
entropy = 0.229
samples = 108
value = [0, 0, 0, 0, 104, 4]
class = ATE*
entropy = 0.0
samples = 5
value = [0, 0, 0, 5, 0, 0]
class = ATE
H (mag) <= 22.05
entropy = 0.078
samples = 1042
value = [0, 10, 1032, 0, 0, 0]
class = APO*
entropy = 0.0
samples = 67
value = [67, 0, 0, 0, 0, 0]
class = AMO*
entropy = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 4]
class = IEO*
entropy = 0.0
samples = 104
value = [0, 0, 0, 0, 104, 0]
class = ATE*
entropy = 0.0
samples = 1032
value = [0, 0, 1032, 0, 0, 0]
class = APO*
entropy = 0.0
samples = 10
value = [0, 10, 0, 0, 0, 0]
class = APO

- Kao napredniju tehniku koristimo RandomForest za poboljšanje performansi modela za klasifikaciju. Ova tehnika radi tako što kombinuje više stabala odlučivanja (Decision Trees) u ansambl, pri čemu svako stablo daje svoju predikciju. Random Forest koristi tehniku "bagging" (Bootstrap Aggregating) kako bi kreirao različite podskupove trening podataka za svako stablo.

```
random_forest_report(model, X_test, y_test)
```

```
Confusion Matrix:
[[ 29   0   0   0   0   0]
 [  0   4   0   0   0   0]
 [  1   0 443   0   0   0]
 [  0   0   0   2   0   0]
 [  0   0   1   0  44   0]
 [  0   0   0   0   0   1]]

Classification Report:
              precision    recall  f1-score   support

        AMO*       0.97      1.00      0.98        29
         APO       1.00      1.00      1.00         4
        APO*       1.00      1.00      1.00       444
         ATE       1.00      1.00      1.00         2
        ATE*       1.00      0.98      0.99        45
        IEO*       1.00      1.00      1.00         1

    accuracy                           1.00       525
   macro avg       0.99      1.00      0.99       525
weighted avg       1.00      1.00      1.00       525
```

```
Precision = True Positives / (True Positives + False Positives)
```

```
Recall = True Positives / (True Positives + False Negatives)
```

```
F1 Score = 2 * (Precision * Recall) / (Precision + Recall)
```

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

K-najbližih suseda

```
params = {
    'n_neighbors' : range(2,20),
    'weights' : ['uniform', 'distance'],
    'p' : [1,2]
}

estimator = GridSearchCV(KNeighborsClassifier(), params, cv = 4, verbose = 4)
```

```
estimator.best_estimator_
```

```
KNeighborsClassifier(n_neighbors=6, p=1, weights='distance')
```

“n_neighbors”:određuje broj najbližih suseda koji će se uzeti u obzir pri donošenju odluke o klasi novih instanci.
“weights”: Parametar koji kontroliše težine suseda. Može biti postavljen na "uniform" (svi susedi imaju istu težinu) ili "distance" (susedi imaju težine obrnuto proporcionalne rastojanju od nove instance).
“p”: Parametar koji određuje koji tip rastojanja se koristi. Ako je postavljen na 1, koristi se Manhattan rastojanje (L1 norma), dok se za vrednost 2 koristi Euklidsko rastojanje (L2 norma).

```
Confusion matrix:
[[  7   0  22   0   0   0]
 [  0   0   4   0   0   0]
 [  4   0 438   0   2   0]
 [  0   0   1   0   1   0]
 [  0   0  18   0  27   0]
 [  0   0   0   0   1   0]]
Accuracy: 0.8990476190476191
Precision: 0.8990476190476191
Recall: 0.8990476190476191
F1 score: 0.8990476190476191
```

Koristimo tehniku kombinovanog oversampling-a i undersampling-a, SMOTE-ENN

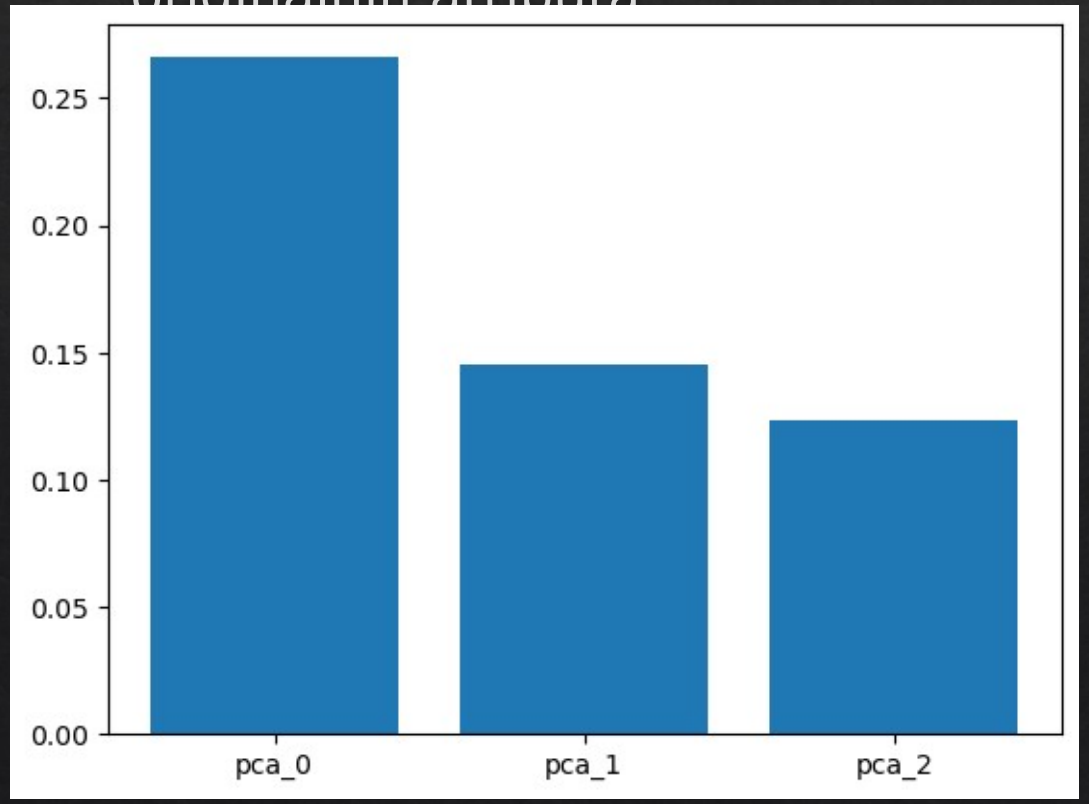
```
Confusion matrix:
[[ 12   0  16   0   1   0]
 [  0   0   4   0   0   0]
 [ 31   7 390   1  15   0]
 [  0   0   1   0   1   0]
 [  0   0  13   3  26   3]
 [  0   0   0   0   0   1]]
Accuracy: 0.8171428571428572
Precision: 0.8171428571428572
Recall: 0.8171428571428572
F1 score: 0.8171428571428572
```

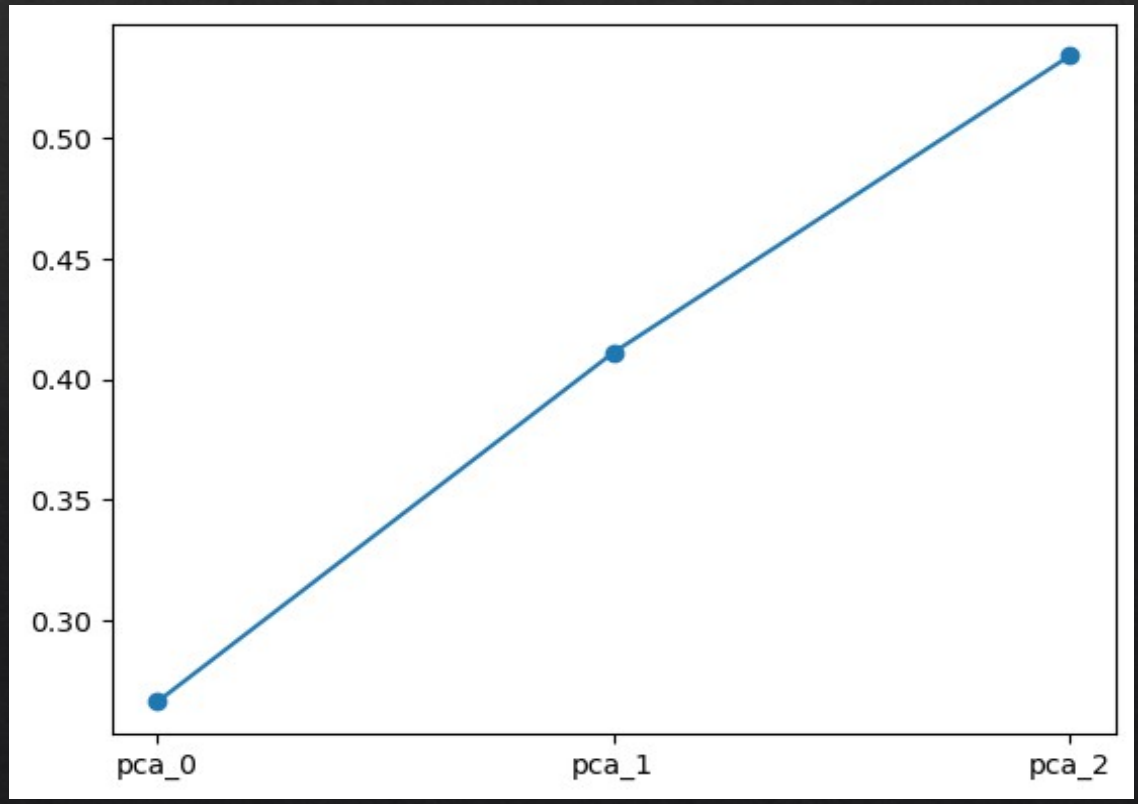
- Kao ansambl tehniku za unapređenje KNN možemo iskoristiti BaggingClassifier. Ovaj ansambl algoritam funkcioniše tako što generiše više podskupova (bootstrap uzoraka) od trening podataka, a zatim trenira više kopija osnovnog algoritma (u ovom slučaju, KNN) na svakom od ovih podskupova.

```
Confusion matrix:
[[  22    0   45    0    0    0]
 [   0    0    9    0    1    0]
 [   0    0 1026    0    6    0]
 [   0    0    2    0    3    0]
 [   0    0   32    0   72    0]
 [   0    0    2    0    2    0]]
Accuracy: 0.9165302782324058
Precision: 0.9165302782324058
Recall: 0.9165302782324058
F1 score: 0.9165302782324058
--------------------------------
Confusion matrix:
[[  3   0  26   0   0   0]
 [  0   0   4   0   0   0]
 [  3   0 437   0   4   0]
 [  0   0   1   0   1   0]
 [  0   0  29   0  16   0]
 [  0   0   0   0   1   0]]
Accuracy: 0.8685714285714285
Precision: 0.8685714285714285
Recall: 0.8685714285714285
F1 score: 0.8685714285714285
```

Analiza glavnih komponenti(PCA)

- PCA radi tako što transformiše originalne atribute u novi skup nezavisnih atributa, nazvane glavne komponente, koje su linearna kombinacija originalnih atributa



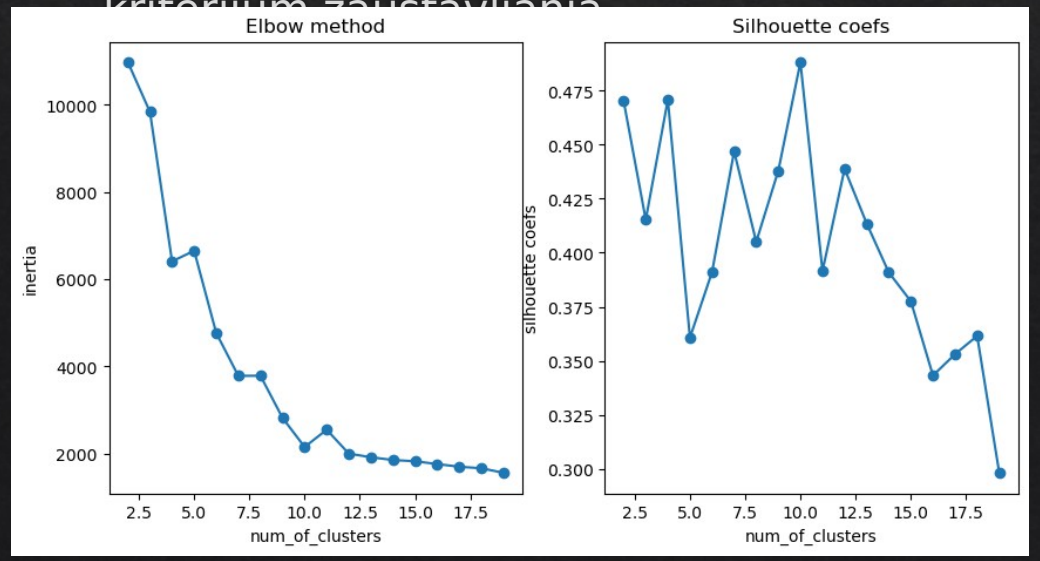


K-sredina

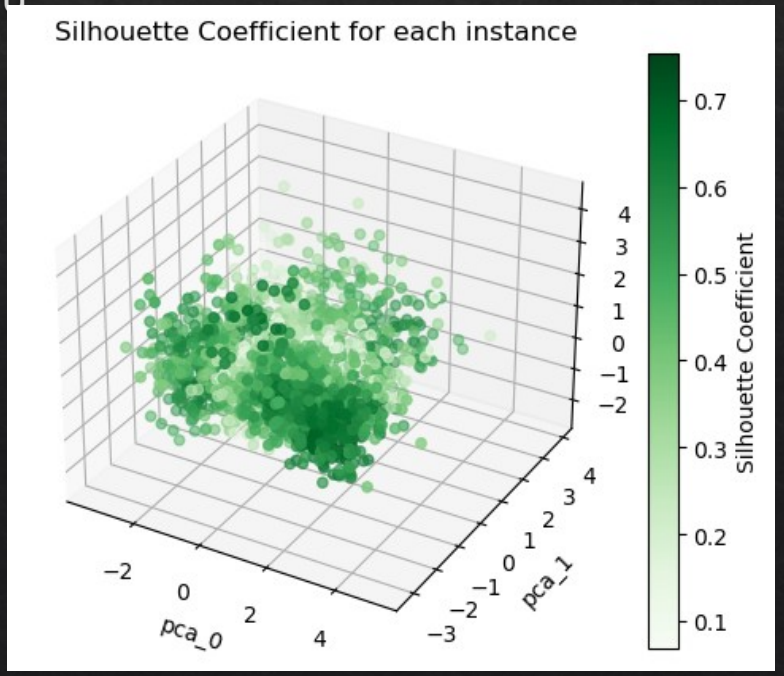
Algoritam K-sredina (K-means) je algoritam za klasterovanje koji funkcioniše tako što deli skup podataka u K klastera, gde je K unapred definisani broj klastera.

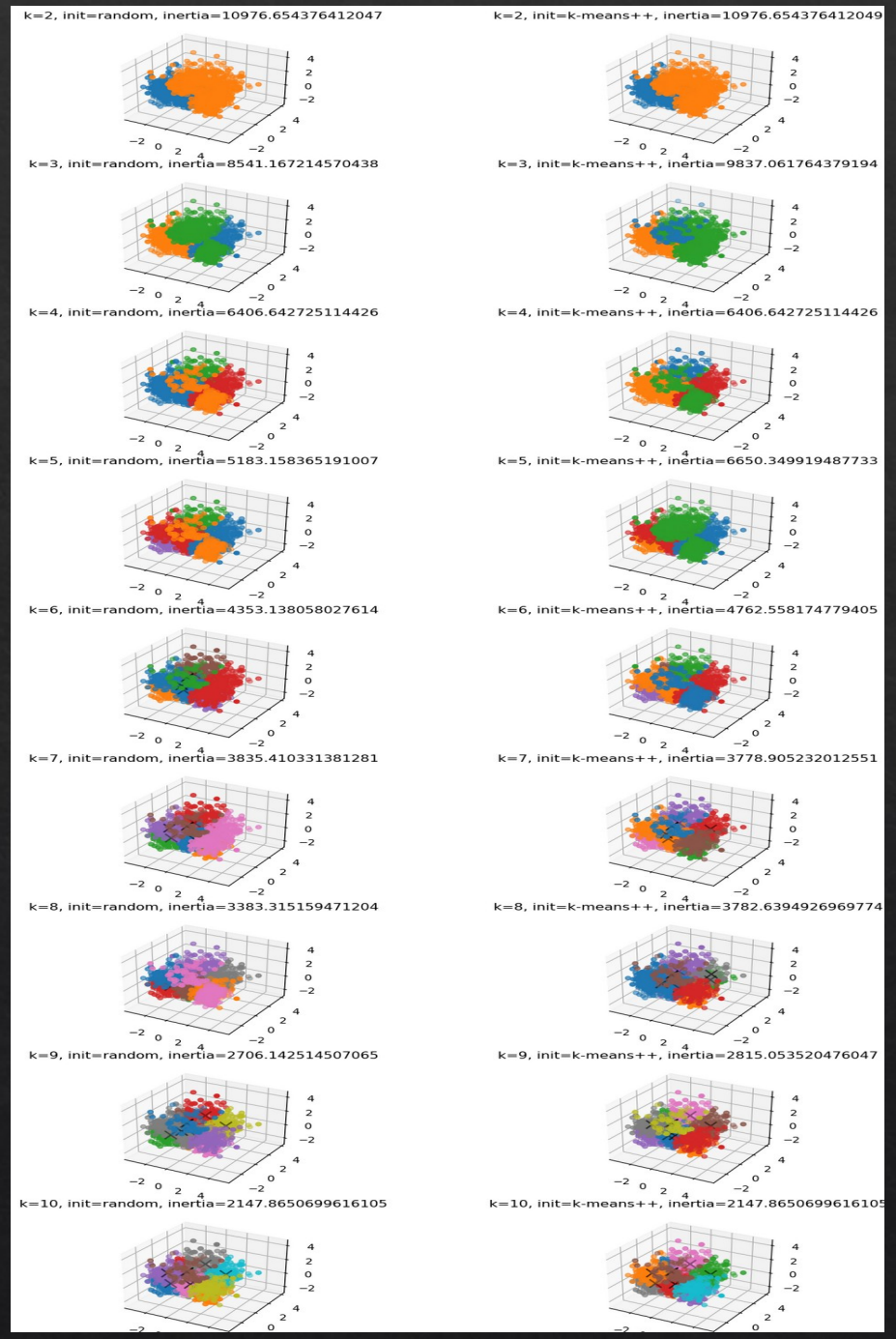
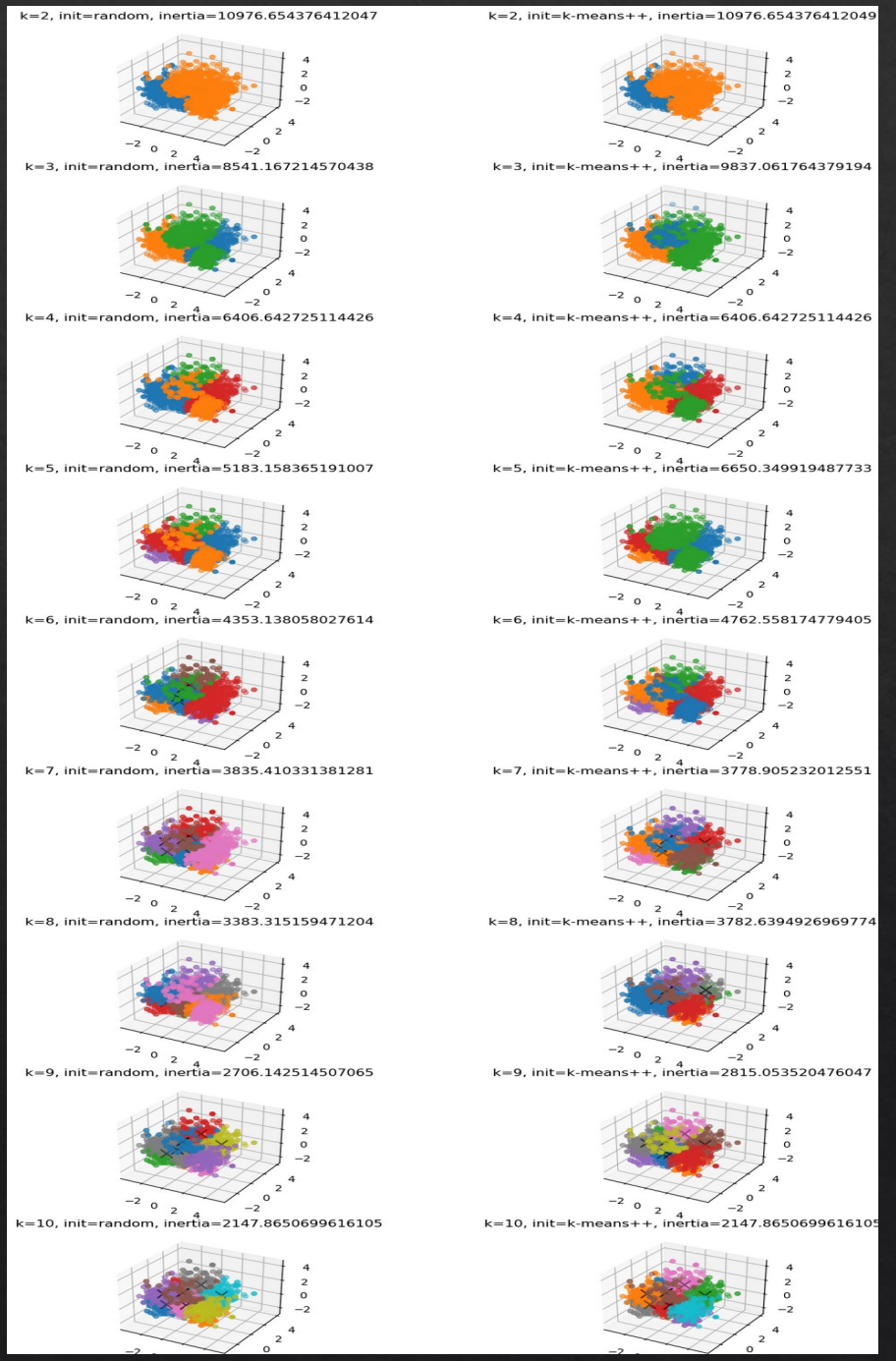
Iterativno izvršava sledeće korake:

- Svaka tačka se dodeljuje najbližem centru, formirajući time klasterove
- Centri se ponovo računaju kao srednje vrednosti tačaka u svakom klasteru
- Postupak se ponavlja dok se centri ne stabilizuju ili dok ne bude ispunjen kriterijum zaustavljanja



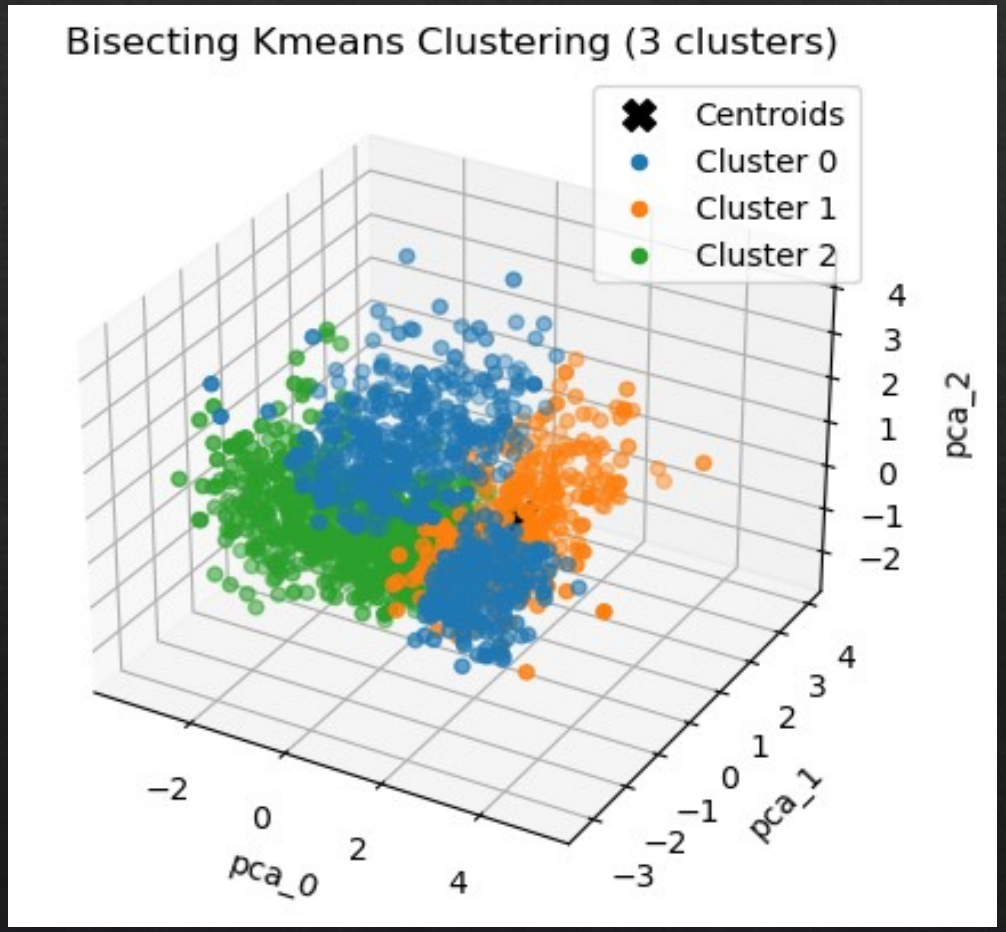
SSE (Sum of Squared Errors): meri sumu kvadratnih udaljenosti svake tačke od njenog najbližeg centra klastera
Silhouette koeficijent: meri koliko je svaka tačka slična tačkama u svom klasteru u poređenju sa tačkama u drugim klasterima




k=2, init=random, inertia=10976.654376412047
k=2, init=k-means++, inertia=10976.654376412049
k=3, init=random, inertia=8541.167214570438
k=3, init=k-means++, inertia=9837.061764379194
k=4, init=random, inertia=6406.642725114426
k=4, init=k-means++, inertia=6406.642725114426
k=5, init=random, inertia=5183.158365191007
k=5, init=k-means++, inertia=6650.349919487733
k=6, init=random, inertia=4353.138058027614
k=6, init=k-means++, inertia=4762.558174779405
k=7, init=random, inertia=3835.410331381281
k=7, init=k-means++, inertia=3778.905232012551
k=8, init=random, inertia=3383.315159471204
k=8, init=k-means++, inertia=3782.6394926969774
k=9, init=random, inertia=2706.142514507065
k=9, init=k-means++, inertia=2815.053520476047
k=10, init=random, inertia=2147.8650699616105
k=10, init=k-means++, inertia=2147.8650699616105

k=2, init=random, inertia=10976.654376412047
k=2, init=k-means++, inertia=10976.654376412049
k=3, init=random, inertia=8541.167214570438
k=3, init=k-means++, inertia=9837.061764379194
k=4, init=random, inertia=6406.642725114426
k=4, init=k-means++, inertia=6406.642725114426
k=5, init=random, inertia=5183.158365191007
k=5, init=k-means++, inertia=6650.349919487733
k=6, init=random, inertia=4353.138058027614
k=6, init=k-means++, inertia=4762.558174779405
k=7, init=random, inertia=3835.410331381281
k=7, init=k-means++, inertia=3778.905232012551
k=8, init=random, inertia=3383.315159471204
k=8, init=k-means++, inertia=3782.6394926969774
k=9, init=random, inertia=2706.142514507065
k=9, init=k-means++, inertia=2815.053520476047
k=10, init=random, inertia=2147.8650699616105
k=10, init=k-means++, inertia=2147.8650699616105

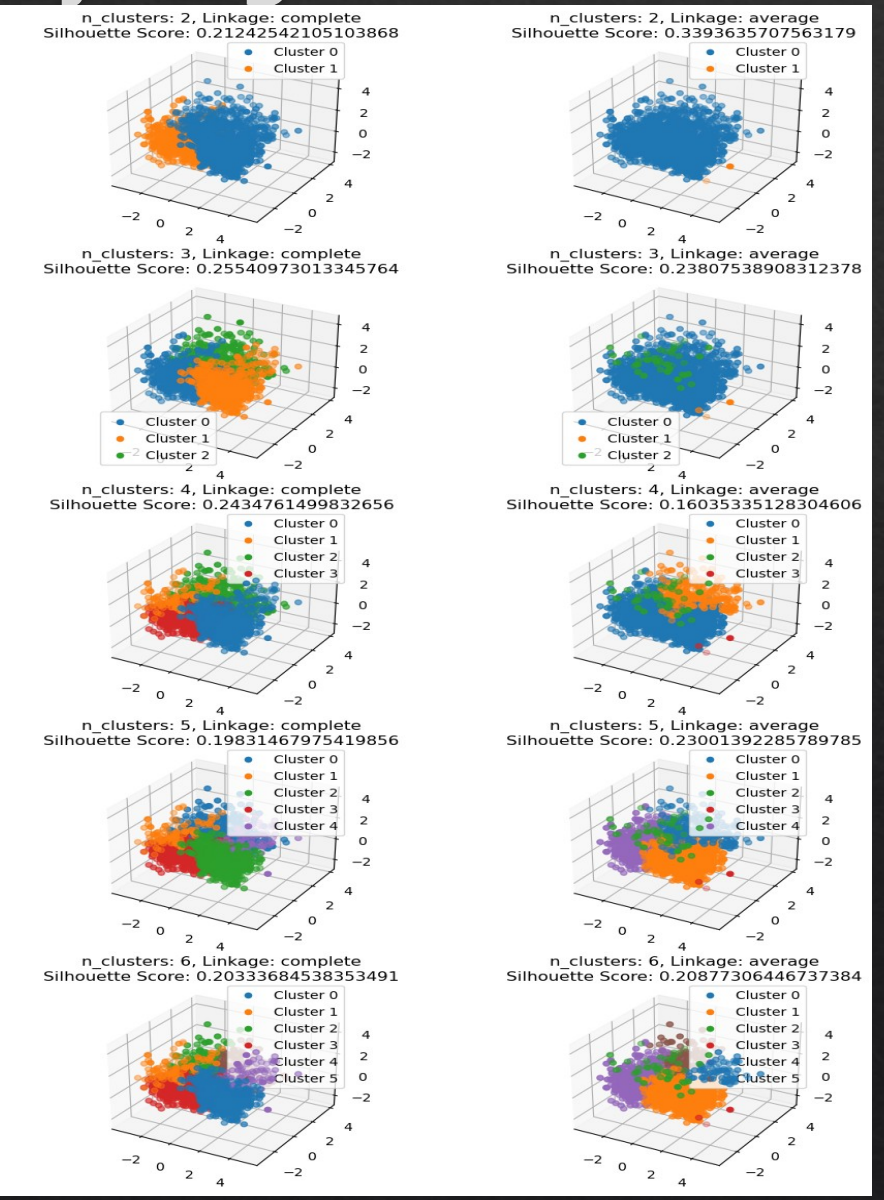
Bisekting K-Means

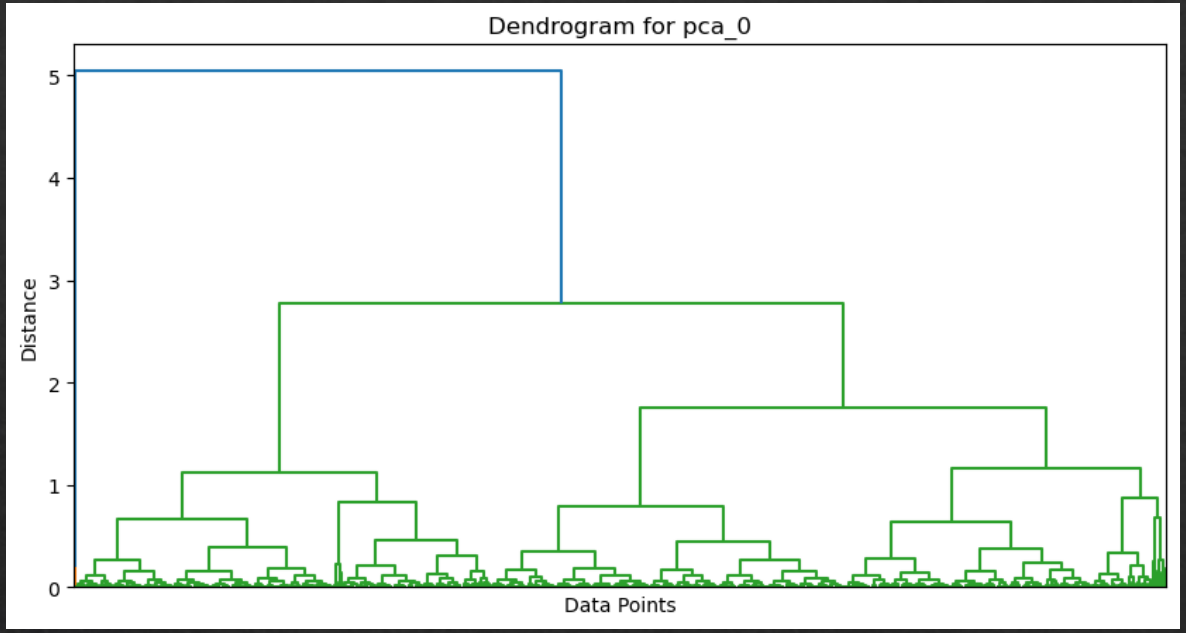
- Bisekting K-Means (Bisektni K-sredina) je varijacija algoritma K-sredina koja se koristi za podelu podataka na K klastera putem hijerarhijskog pristupa. Ovaj algoritam počinje sa jednim klasterom koji obuhvata sve tačke i zatim iterativno deli klaster na dva manja klastera tako da se minimizira suma kvadratnih udaljenosti unutar svakog podklastera.

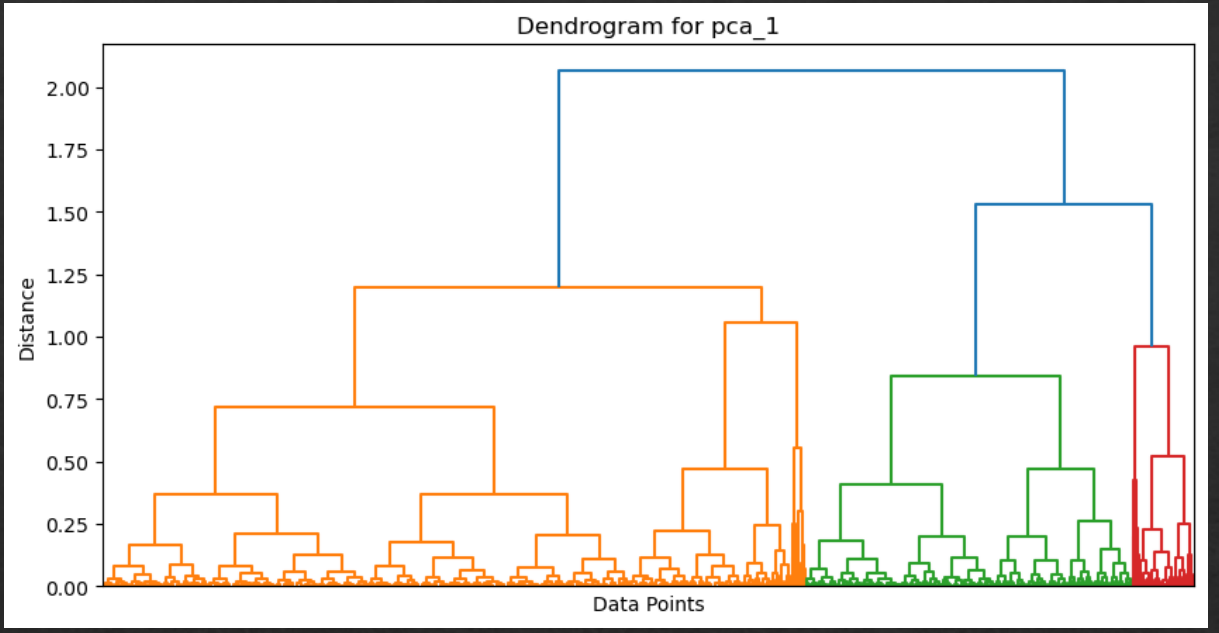


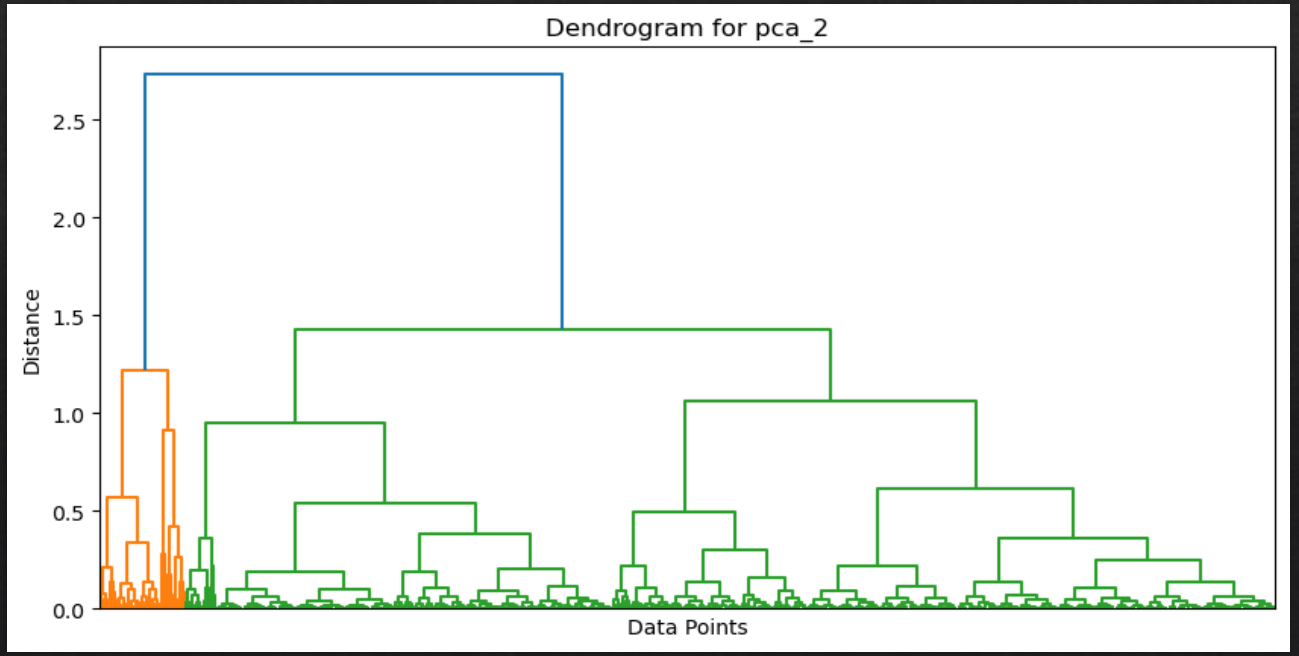
Algoritam sakupljajućeg hijerarhijskog klasterovanja

- Algoritam sakupljajućeg hijerarhijskog klasterovanja je metoda klasterovanja koja počinje sa svakom tačkom kao zasebnim klasterom i iterativno spaja najbliže klasterove kako bi se formirali hijerarhijski klasteri




Dendrogram for pca_0
Distance
Data Points
0
1
2
3
4
5

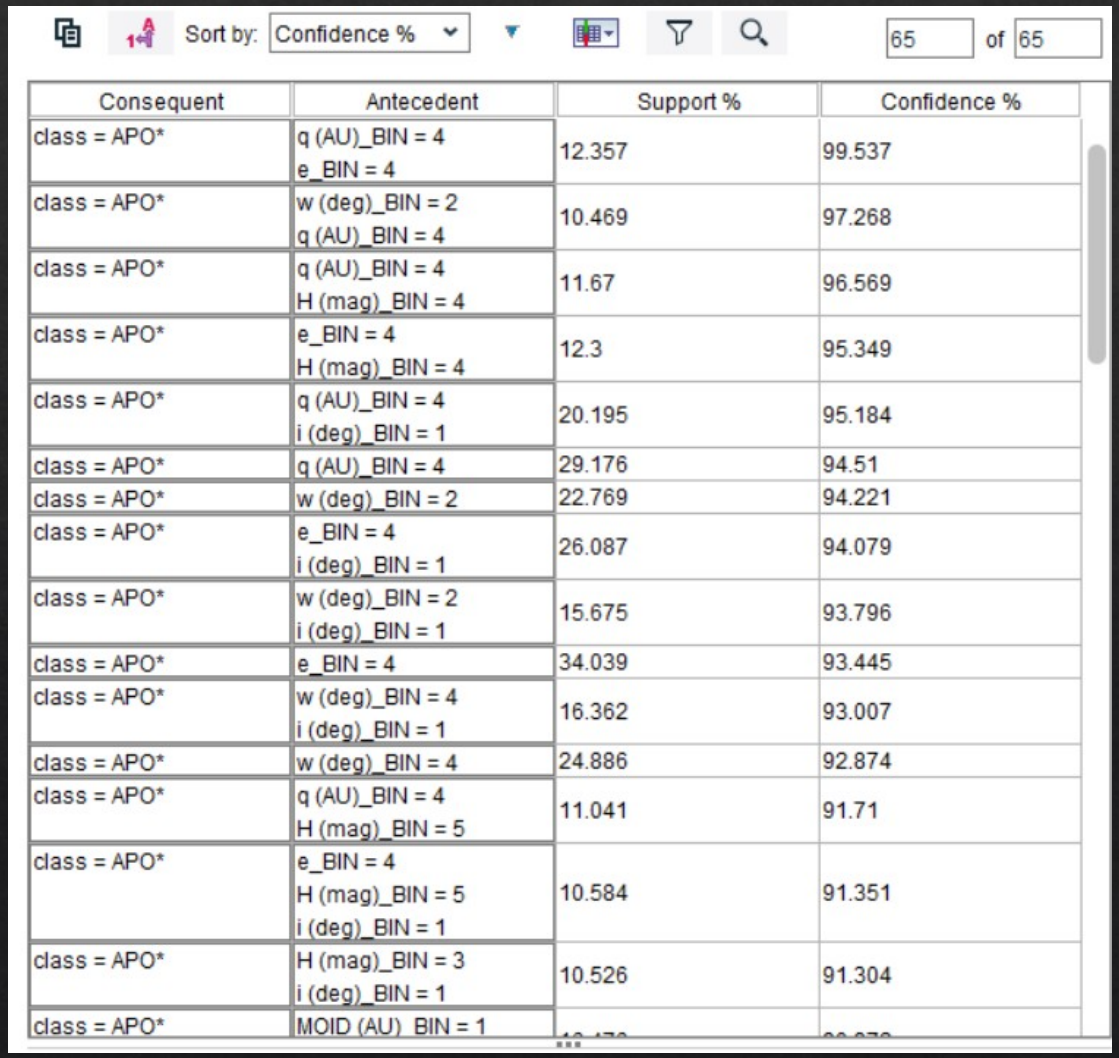

Dendrogram for pca_1
Distance
Data Points
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Dendrogram for pca_2
Distance
Data Points
0.0
0.5
1.0
1.5
2.0
2.5

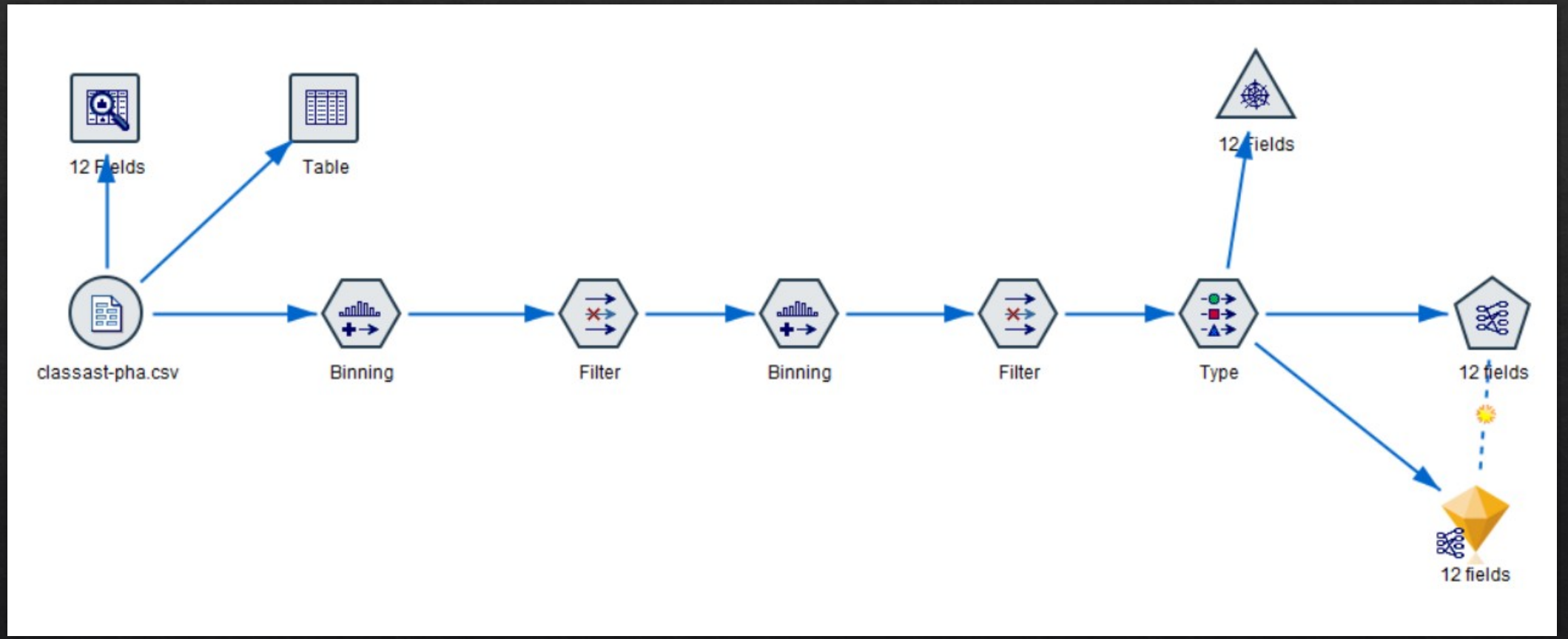
Pravila pridruživanja

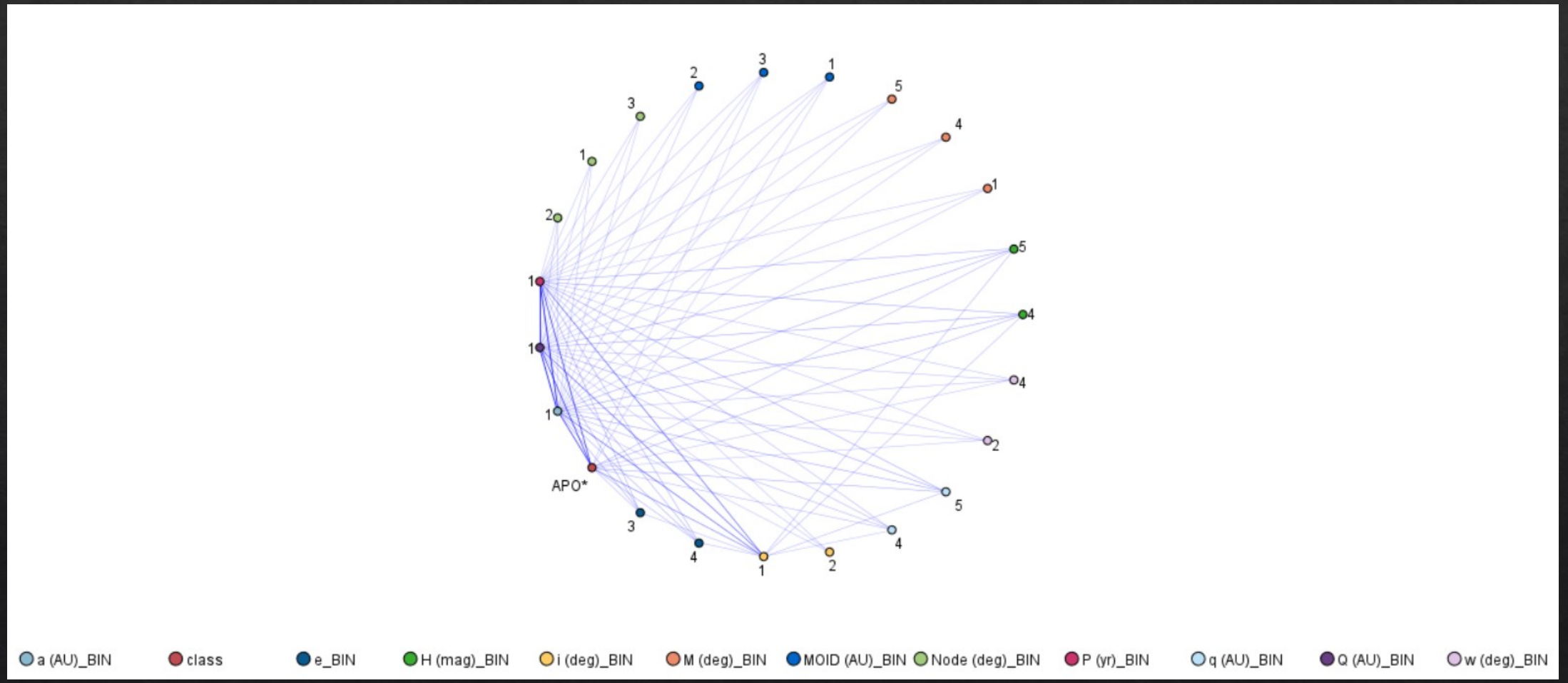
Primenjuje se kako bi se otkrili korisni obrasci i veze između različitih atributa u velikim skupovima podataka. Apriori algoritam radi tako što identifikuje česte asocijacije među stavkama i stvara "ako-onda" pravila koja opisuju te veze.

- **Pouzdanost** (eng. confidence): Ova metrika meri koliko često se pravilo stvarno ostvaruje u stvarnim podacima. Izražava se kao verovatnoća da će "onda" deo pravila biti tačan, ako je "ako" deo tačan.
- **Podrška** (eng. support): Ova metrika meri koliko često se pravilo pojavljuje u celokupnom datasetu.



Consequent	Antecedent	Support %	Confidence %
class = APO*	q (AU)_BIN = 4 e_BIN = 4	12.357	99.537
class = APO*	w (deg)_BIN = 2 q (AU)_BIN = 4	10.469	97.268
class = APO*	q (AU)_BIN = 4 H (mag)_BIN = 4	11.67	96.569
class = APO*	e_BIN = 4 H (mag)_BIN = 4	12.3	95.349
class = APO*	q (AU)_BIN = 4 i (deg)_BIN = 1	20.195	95.184
class = APO*	q (AU)_BIN = 4	29.176	94.51
class = APO*	w (deg)_BIN = 2	22.769	94.221
class = APO*	e_BIN = 4 i (deg)_BIN = 1	26.087	94.079
class = APO*	w (deg)_BIN = 2 i (deg)_BIN = 1	15.675	93.796
class = APO*	e_BIN = 4	34.039	93.445
class = APO*	w (deg)_BIN = 4 i (deg)_BIN = 1	16.362	93.007
class = APO*	w (deg)_BIN = 4	24.886	92.874
class = APO*	q (AU)_BIN = 4 H (mag)_BIN = 5	11.041	91.71
class = APO*	e_BIN = 4 H (mag)_BIN = 5 i (deg)_BIN = 1	10.584	91.351
class = APO*	H (mag)_BIN = 3 i (deg)_BIN = 1	10.526	91.304
class = APO*	MOID (AU)_BIN = 1		


12 Fields
Table
12 Fields
classast-pha.csv
Binning
Filter
Binning
Filter
Type
12 fields
12 fields


APO*
a (AU)_BIN
class
e_BIN
H (mag)_BIN
i (deg)_BIN
M (deg)_BIN
MOID (AU)_BIN
Node (deg)_BIN
P (yr)_BIN
q (AU)_BIN
Q (AU)_BIN
w (deg)_BIN