Tehnika klasterovanja zasnovana na genetskim algoritmima

Seminarski rad u okviru kursa Računarska inteligencija
Matematički fakultet, Beograd

Radiša Mitrović, Milica Marić

Decembar 2021

Sadržaj

1 O problemu

Ovaj rad se bavi tehnikom klasterovanja zasnovanom na genetskim algoritmima.

1.1 Klasterovanje

Klasterovanje je podela objekata u grupe, takve da su objekti u jednoj grupi međusobno slični ili povezani, a objekti u različitim grupama međusobno različiti tj. nepovezani. Broj klastera zavisi od posmatranog kriterijuma [1]. Potrebno je odrediti meru sličnosti na osnovu koje se klasteri grupišu. Jedna od najčešće korišćenih mera sličnosti je Euklidsko rastojanje, pri čemu je sličnost veća što je rastojanje manje.
Postoji više tehnika klasterovanja. Najčešće korišćena je algoritam *K-means* [2].

1.2 Genetski algoritmi

Genetski algoritmi pripadaju tehnikama pretrage. Tačnije, pripadaju evolutivnim algoritmima koji su podgrana tehnika pretrage.
Genetski algoritmi su zasnovani na principu evolucije i genetike. Pronalaze približno optimalna rešenja za *ciljnu* tj. *fitnes funkciju* za optimizacione probleme [2]. Postupak pronalaženja rešenja se odvija kroz *generacije*, pri čemu u svakoj generaciji najčešće imamo isti broj jedinki. Početna generacija se obično sastoji od slučajno generisanih jedninki, ali može da bude rezultat neke druge optimizacione metode.
Funkcija prilagođenosti ili *funkcija kvaliteta* je funkcija koja računa kvalitet jedinke. Ta funkcija može, ali nije obavezno jednaka funkciji cilja.
Na osnovu vrednosti funkcije prilagođenosti se kroz proces koji se naziva *selekcija* biraju jednike koje će uticati na stvaranje novih jedinki ili *potomstva*. Nove jedinke se dobijaju *ukrštanjem* od roditelja tj. polaznih jedinki. Nakon operatora ukrštanja, primenjuje se operator *mutacije* koji sa obično veoma malom verovatnoćom menja deo jedinke. *Mutacija* se koristi da bi se obnovio izgubljeni genetski materijal i da bi se prečilo da jedinke postanu previše slične [3].

Osnovni koraci genetskog algoritma:

```
Begin
    1.  t = 0
    2.  Inicijalizuj populaciju P(t)
    3.  Izracunaj prilagodenost P(t)
    4.  t = t + 1
    5.  Ako je ispunjen kriterijum zaustavljanja, idi na korak 10
    6.  Izaberi P(t) iz P(t-1)
    7.  Izvrsi ukrstanje P(t)
    8.  Izvrsi mutaciju P(t)
    9.  Idi na korak 3
    10. Ispisi najbolji i zaustavi se
End
```

[2]

2 Opis rešenja problema

1. Implmentacija jedinke:

   - Jedinka u algoritmu je lista koja predstavlja pozicije k centroida u N dimenzionom prostoru:

     $$[(x_{11}, x_{12}, \cdots, x_{1N}), \cdots, (x_{k1}, x_{k2}, \cdots, x_{kN})]$$

   - Na primer, ako tacke pripadaju prostoru R^2 i ako je zadati broj klastera k jednak 3, onda je jedna jedinka oblika:

     $$[(x_1, y_1), (x_2, y_2), (x_3, y_3)]$$

2. Fitnes jedinke:

   - Neka su date tacke koje treba klasterovati $X1, X2, \cdots, Xm$ i neka je jedinka predstavljenja [(x11, x12, ..., x1N), ..., (xk1, xk2, ..., XkN)](k centroida u N dimenzionom prostoru). Neka je mera rastojanja koja se koristi kvadrat Euklidske norme tj. $dist(X, Y) = \sum_{i=0}^{N} (Xi - Yi)^2$.
   - Fitnes jedinke jeste suma kvadrata rastojanja tačaka do njihovih centroida tj.
   $$fitness = \sum_{i=1}^{k} \sum_{x \in C_i} dist(c_i, x)$$
   gde je x tačka u klasteru Ci a c_i centroida tog klastera.
   - U klasicnom Genetskom algoritmu jednika je kvalitetnija sto je njen fitnes **veci**. Kod nas u algoritmu je obrnuto, jedinka je kvalitetnija sto ima **manji** fitnes. Ovo smo mogli svesti na originalan nacin posmatranja jednostavno posmatrajuci fitnes kao $\frac{1}{SSE}$ umesto SSE. Medjutim, mi smo radi lakseg koriscenja fitnesa ostali na posmatranju SSE vrednosti.
   - Dakle, jedinka je bolja sto ima **manji** fitness odnosno SSE vrednost.

3. Klasa GACluster

   - Za koriscenje algoritma napravljena je klasa GACluster. Sam nacin koriscenja odnosno interfejs je napravljen tako da bude u skladu sa interfejsom klase KMeans iz biblioteke scikit-learn. Razlog za ovu odluku jeste to sto je biblioteka scikit-learn veoma dobro poznata onima koji se bave klasterovanjem. Dakle, nacin koriscenja klase GACluster i klase KMeans je maltene identican.
   - Prilikom poziva metoda fit koji vrsi klasterovanje tacaka kao pomoc je dodato i iscrtavanje najbolje jedinke nekon svakih 10 iteracija, ovime je omoguceno lepo vizuelno pracene Genetskog algoritma kroz iteracije. Kad se ovako vizuelno prate rezultati klasterovanja kroz iteracije lako se mogu korigovati parametri Genetskog algoritma kako bi se povecala efikasnost i kvalitet dobijenog resenja.
   - Sto se tice same efikasnosti algoritma u odnosu na KMeans algoritam, nas algoritam je sporiji sto je i ocekivano kada se radi sa Genetskim algoritmima. Takodje, na usporavanje uticu i iscrtavanja grafika.

4. Genetski algoritam:
   Sam genetski algoritam poziva funkcije za selekciju, mutaciju i ukrštanje.

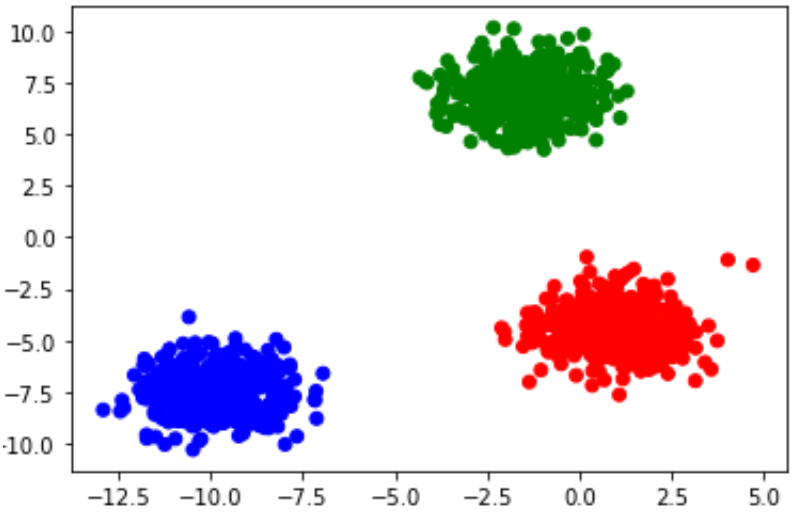
   - Selekcija može biti ruletska ili turnirska.
   Ruletska selekcija je implementirana tako što se računa fitnes svake jedinke u populaciji.Računaju se verovatnoće za svaku jedinku, proporcionalne fitnesu. Nakon toga, bira se pseudo-slučajan broj iz

inervala [0, 1). Ruletska selekcija vraća indeks one jednike čija je verovatnoča da bude izabrana veča ili jednaka pseudo-slučajnom broju. Kod *turnirske* selekcije imamo veličinu turnira koja predstavlja broj iteracija. U svakoj iteraciji se bira nasumičan indeks jedinke u populaciji. Na kraju se vrača onaj nasumičan indeks jedinke sa najboljim fitnesom tj. fitnesom koji ima najmanju vrednost.

- Ukrštanje je funkcija koja razmenjuje kod dva roditelja i od njih nastaju dva deteta. Vršeno je *jednopoziciono* ukrštanje. Za jedinku dužine n, generisan je broj u intervalu [0, n-1]. Deca se dobijaju tako što kod roditelja zameni mesta od generisanog broja.
- Svaka jedinka podleže mutaciji sa nekom verovatnoćom. Generiše se broj iz ranga [0.0, 1.0). Ukoliko je taj broj veći od verovatnoće mutacije, onda se mutacija neće izvršiti.
  Generiše se broj delta sa uniformnom raspodelom iz ranga (0, 1). Ako je vrednost na nekoj poziciji v, nakon mutacije ona postaje:
  $v \pm 2 * d * v$, za $v \neq 0$.
  $v \pm 2 * d$, za $v = 0$.

3 Rezultati

3.1 Lepo razdvojeni klasteri

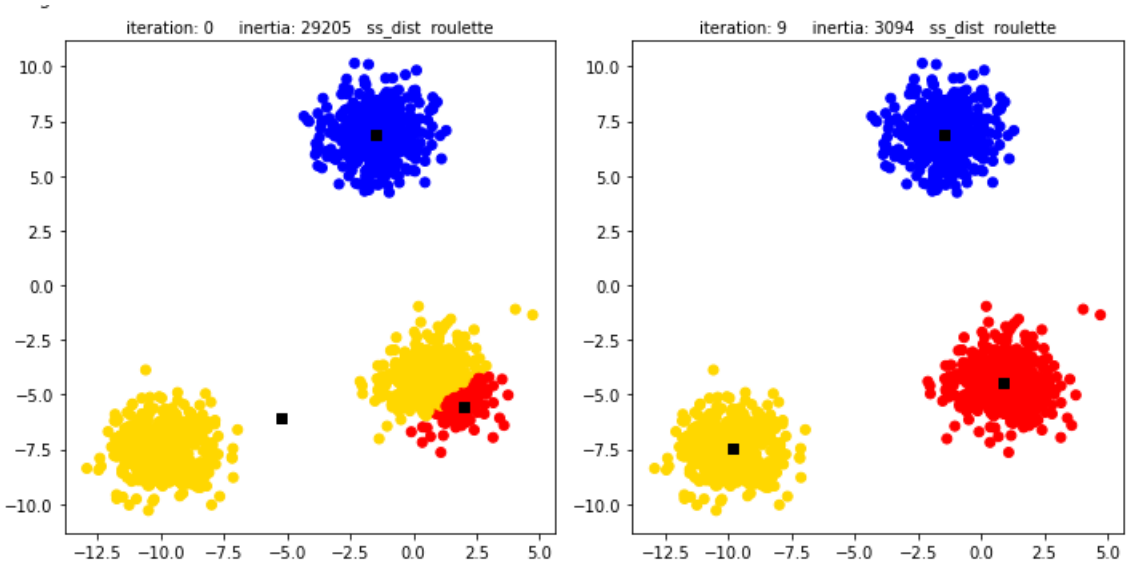


Slika 1: Izgled pre klasterovanja

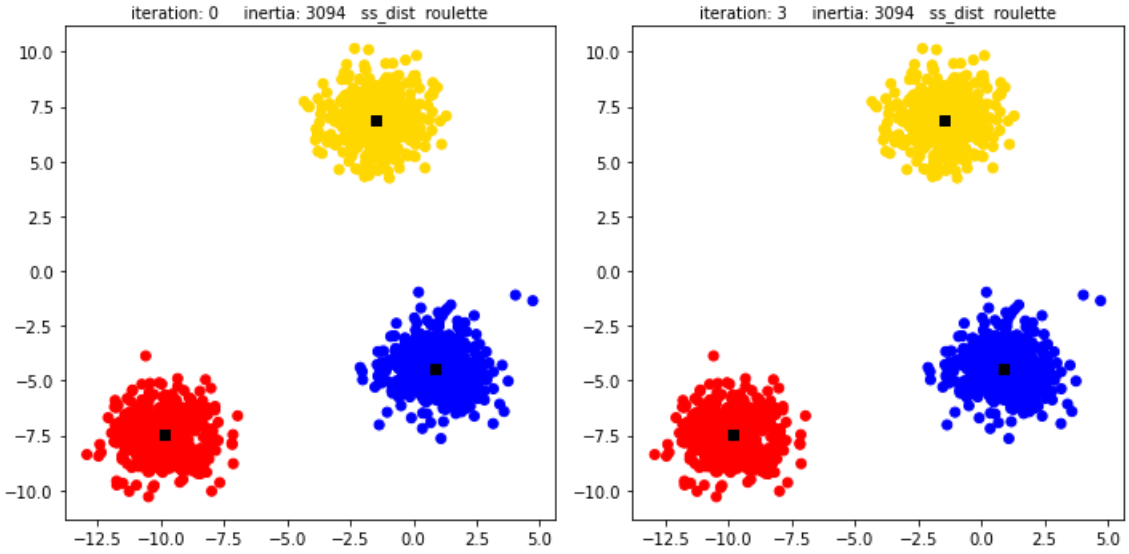
3.1.1 Genetski algoritam

max_iter	pop_size	category	t_size	mutation_rate	elitism_size	sse	silhouette_coef	K_Means_SSE	K_means silhouette
10	4	roulette	/	0.05	2	3095	0.838	3095	0.838
4	4	roulette	/	0.1	2	3095	0.838	3095	0.838
4	4	tournament	2	0.05	2	3095	0.838	3095	0.838
10	10	tournament	2	0.1	2	3095	0.838	3095	0.838

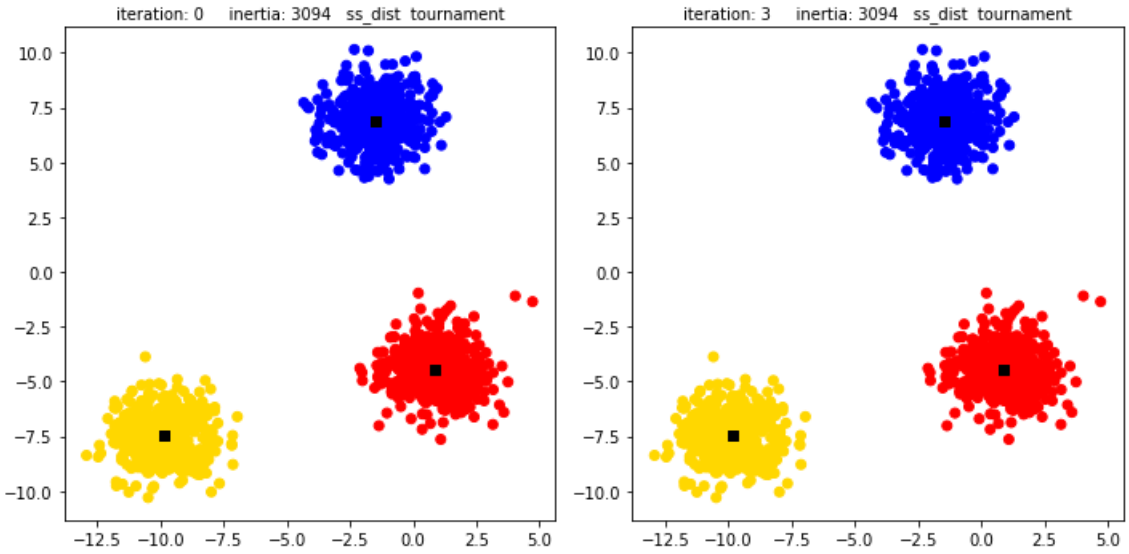
Slika 2: Tabela sa rezultatima



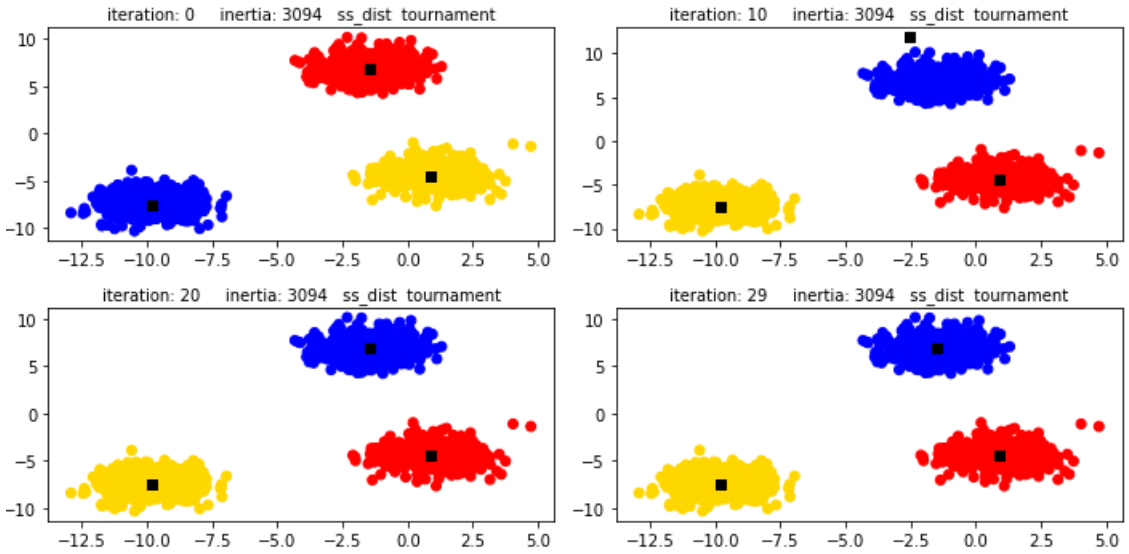
Slika 3: Prvo testiranje



Slika 4: Drugo testiranje

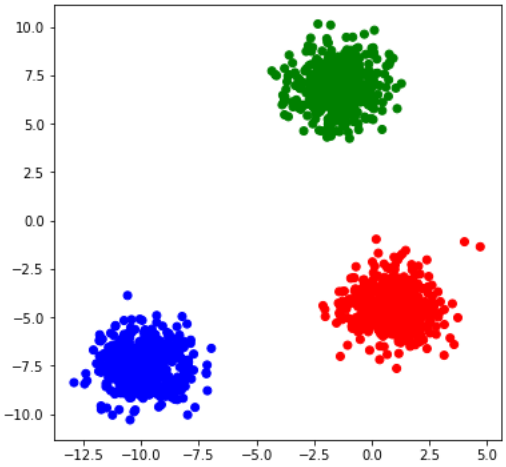


Slika 5: Treće testiranje



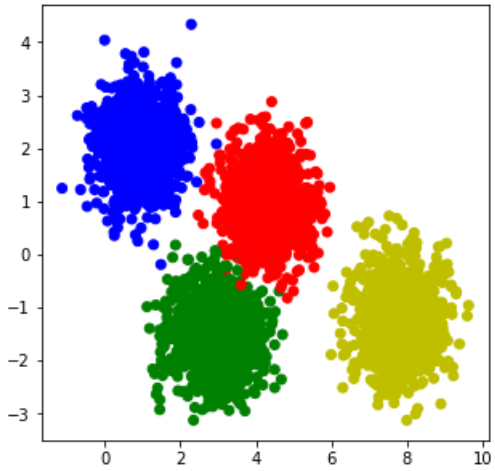
Slika 6: Četvrto testiranje

3.1.2 KMeans



Slika 7: KMeans

3.2 Gusti klasteri



Slika 8: Izgled pre klasterovanja

3.2.1 Genetski algoritam

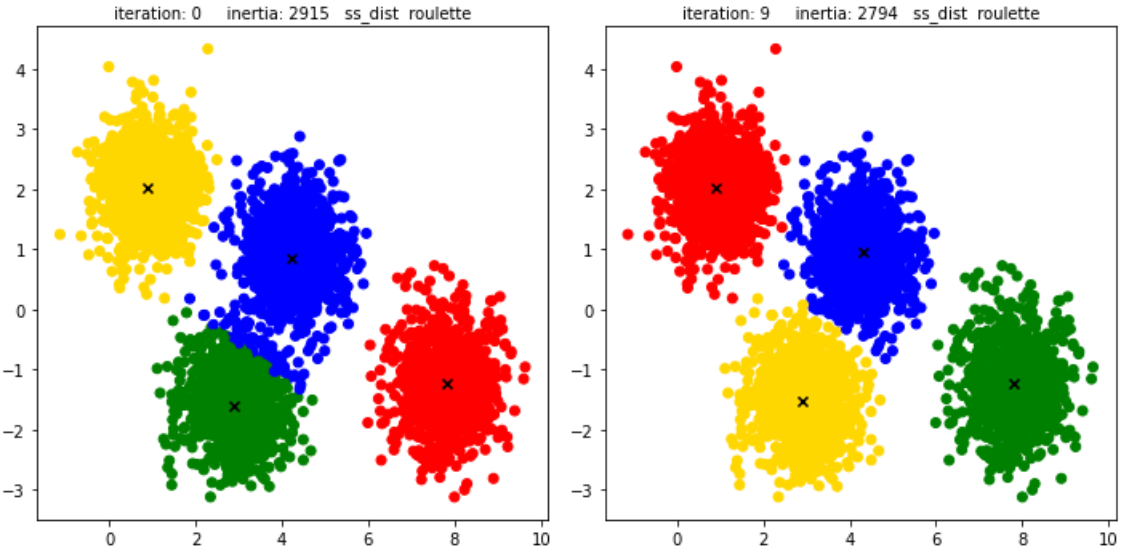
Rezultati - najbolji koji su dobijeni za 4 klastera

max_iter	pop_size	category	t_size	mutation_rate	elitism_size	sse	silhouette_coef	K_Means_SSE	K_means silhouette
10	10	roulette	/	0.05	2	2794	0.67	2794	0.67
10	10	tournament	3	0.05	2	2794	0.67	2794	0.67
20	10	tournament	4	0.05	2	2794	0.67	2794	0.67
10	10	tournament	4	0.2	2	2794	0.67	2794	0.67
5	4	tournament	2	0.1	2	2794	0.67	2794	0.67

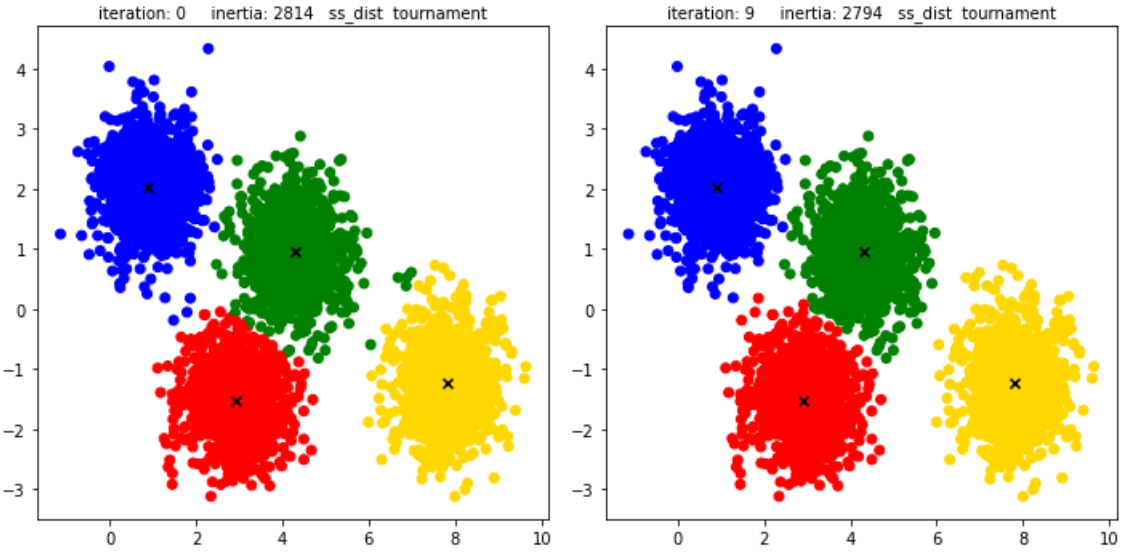
Rezultati - najbolji koji su dobijeni za 3 klastera

max_iter	pop_size	category	t_size	mutation_rate	elitism_size	sse	silhouette_coef	K_Means_SSE	K_means silhouette
10	4	tournament	2	0.05	2	6826	0.6	6826	0.6

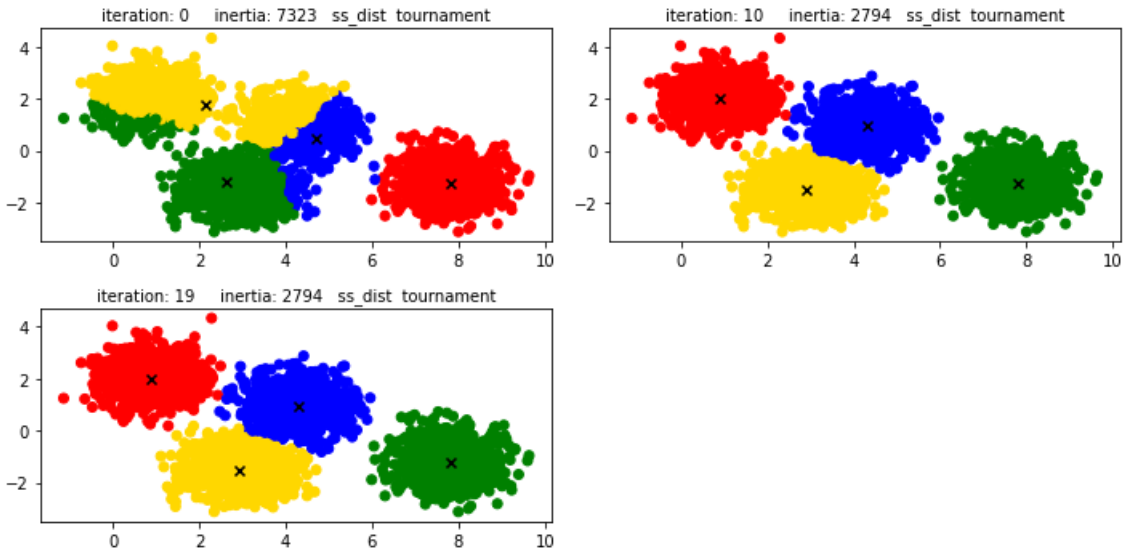
Slika 9: Tabele sa rezultatima



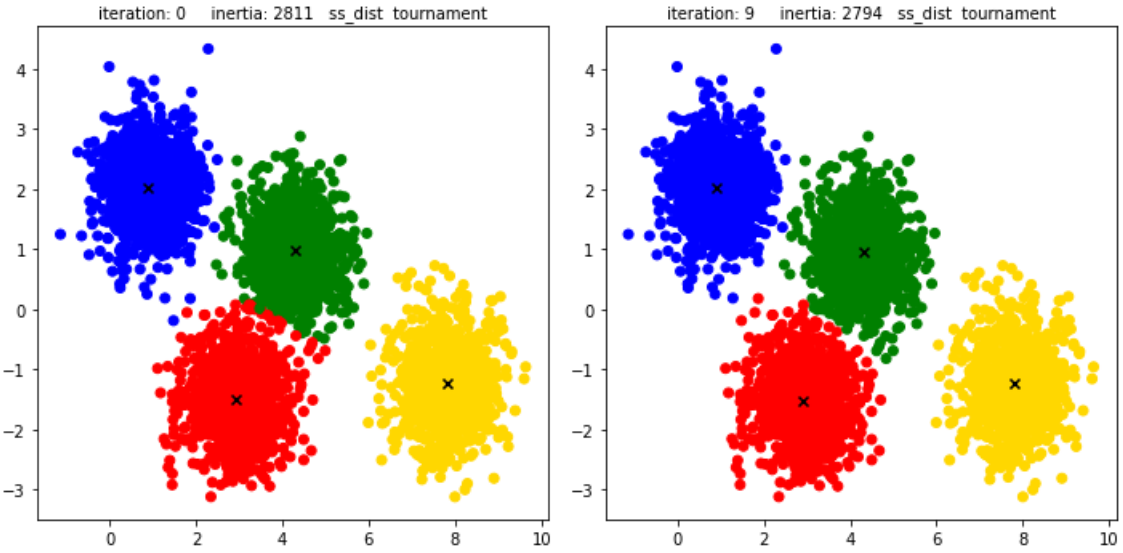
Slika 10: Prvo testiranje



Slika 11: Drugo testiranje

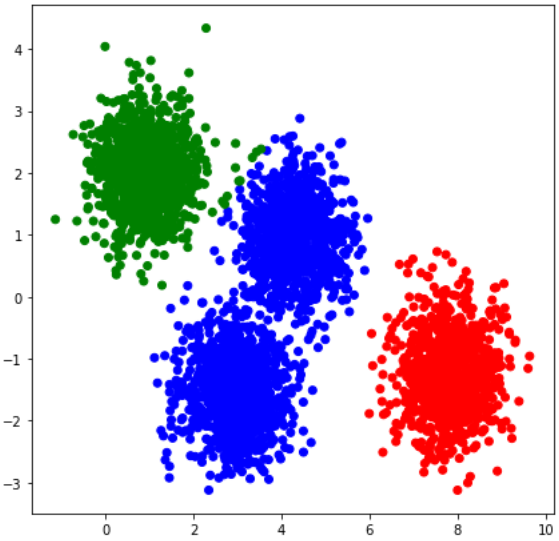


Slika 12: Treće testiranje



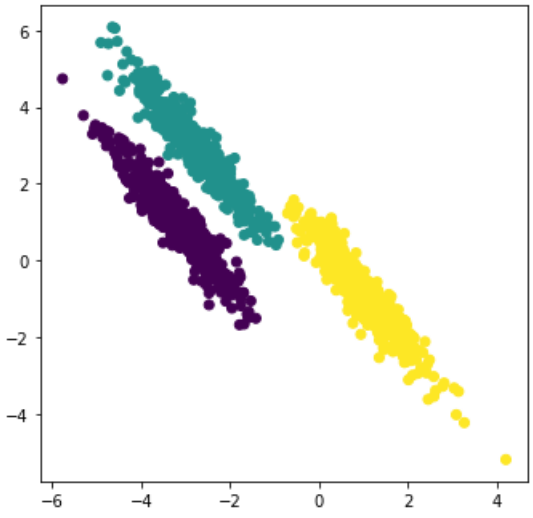
Slika 13: Četvrto testiranje

3.2.2 KMeans



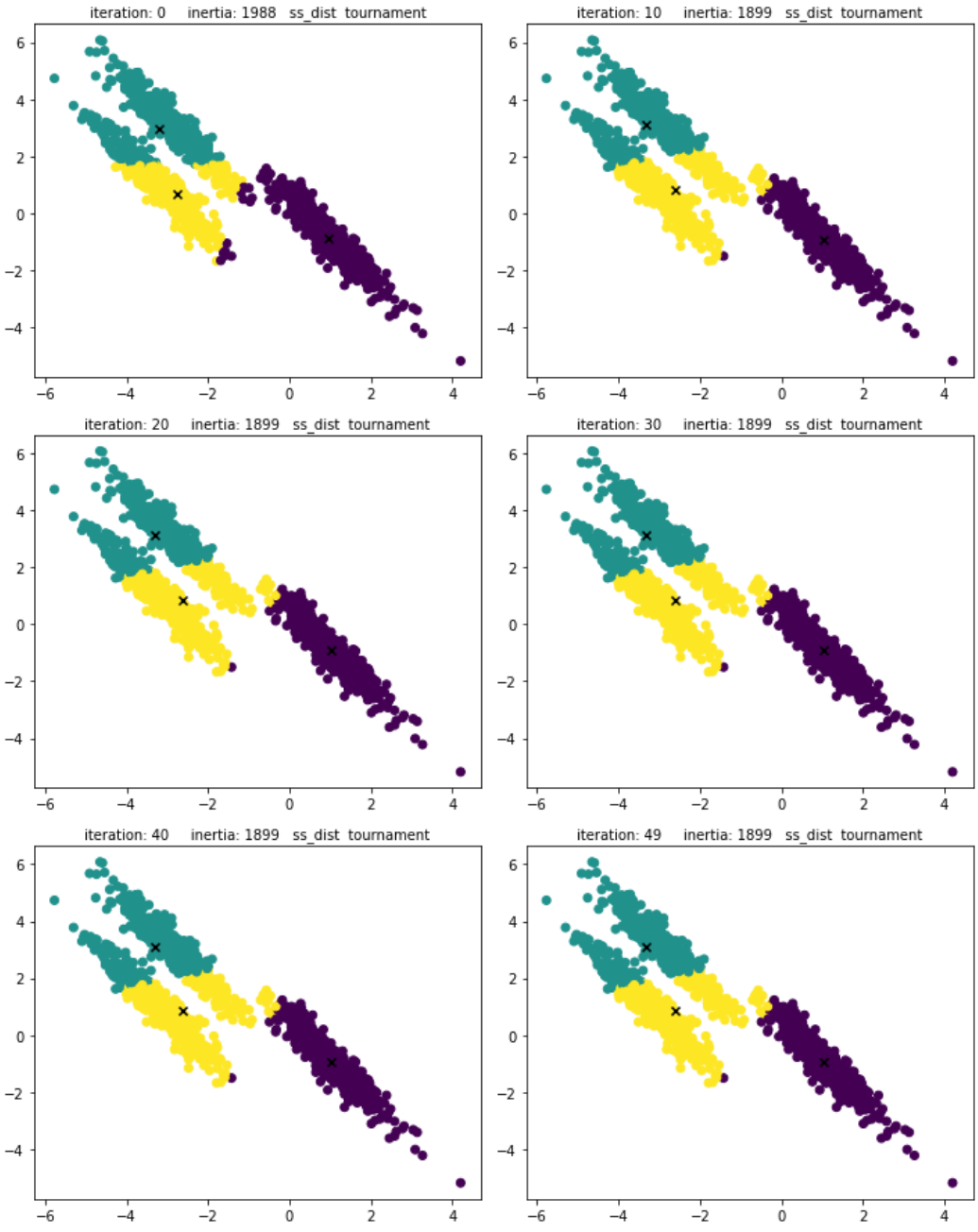
Slika 14: KMeans

3.3 Neglobularni klasteri



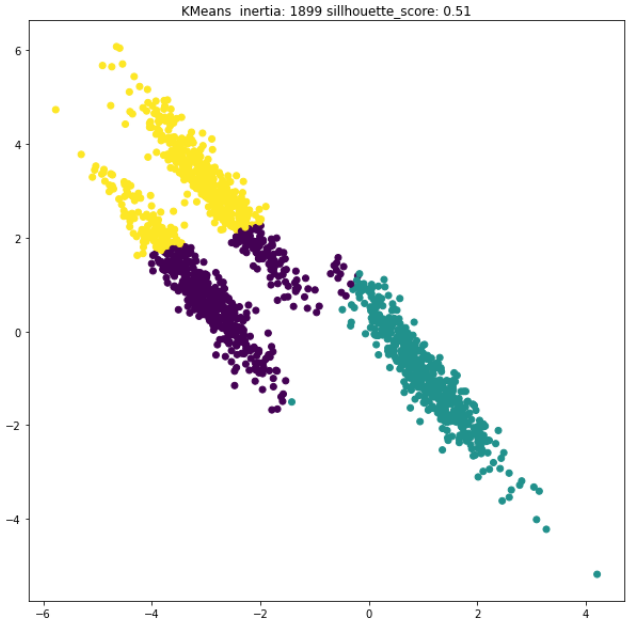
Slika 15: Izgled pre klasterovanja

3.3.1 Genetski algoritam



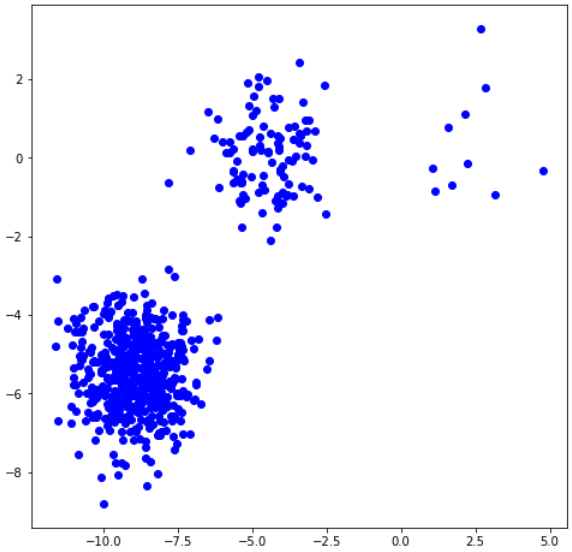
Slika 16: ProcesKlasterovanja

3.3.2 KMeans



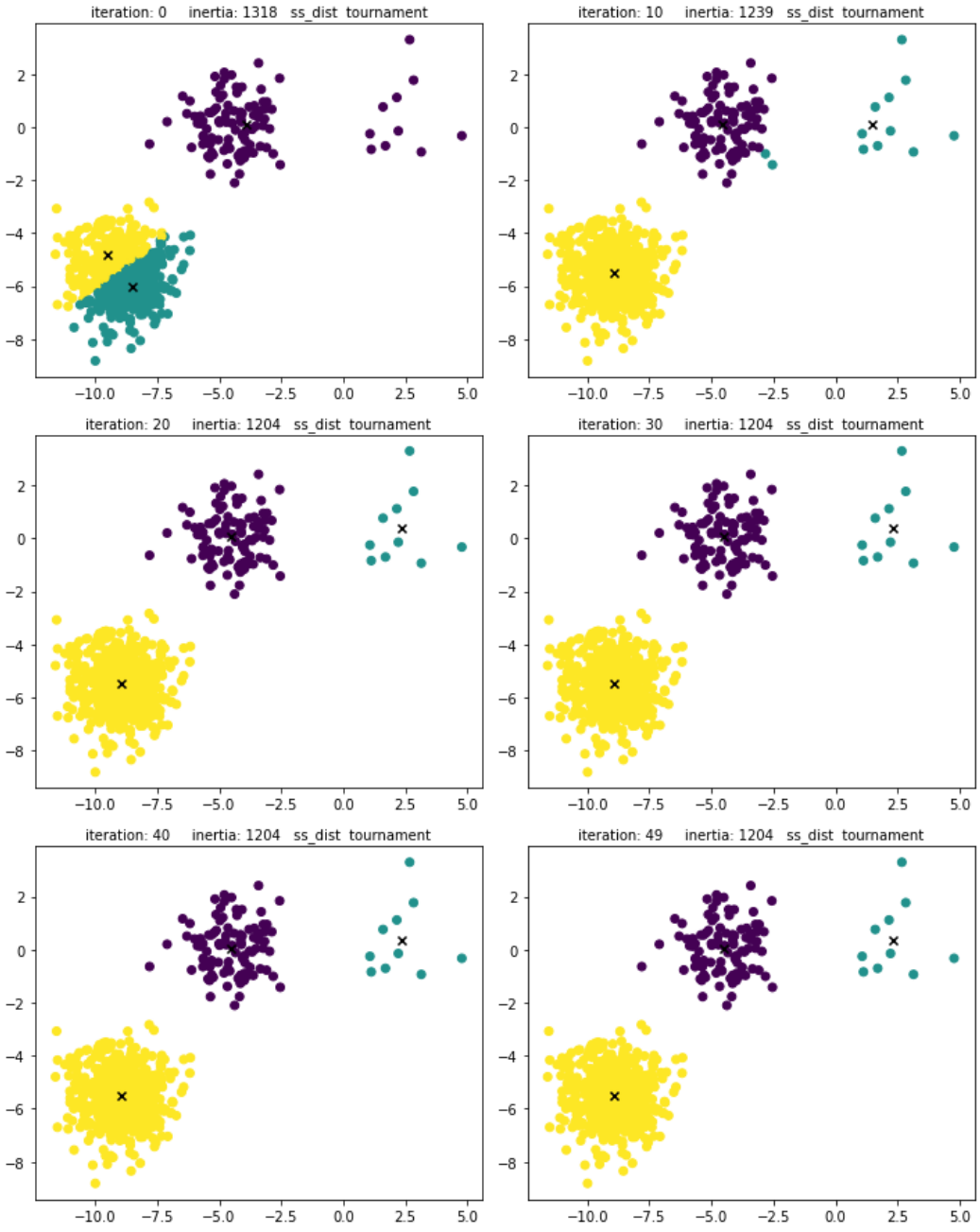
Slika 17: KMeans

3.4 Klasteri različitih gustina



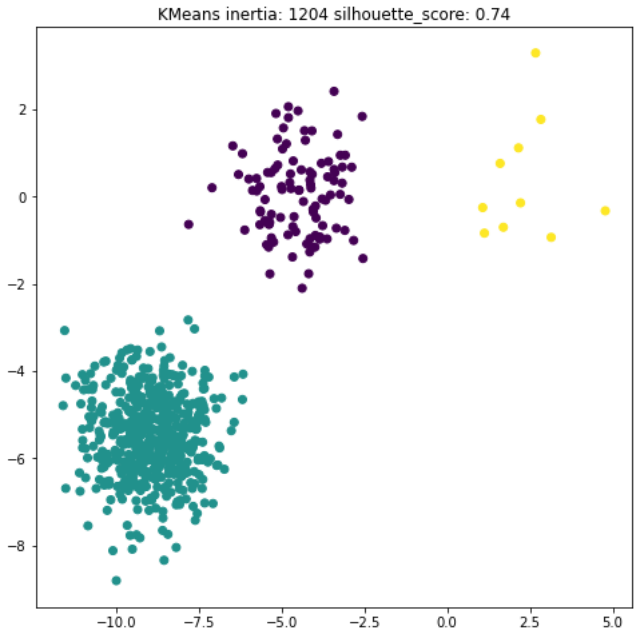
Slika 18: Izgled pre klasterovanja

3.4.1 Genetski algoritam



Slika 19: ProcesKlasterovanja

3.4.2 KMeans



Slika 20: KMeans

3.5 Ostali testovi

Testiranje algoritma rađeno je na skupu Iris, kao i na skupu test delu skupa Human Activity Recognition with Smartphones.

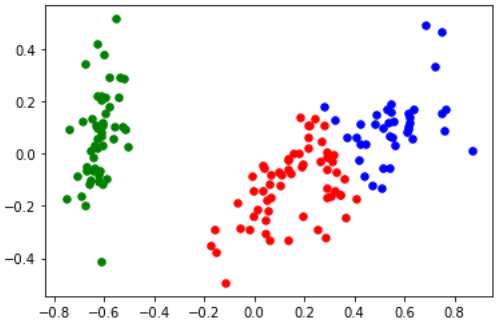
3.5.1 Iris

Sledeća tabela prikazuje rezultate genetskog algoritma:

max_iter	pop_size	category	t_size	mutation_rate	elitism_size	sse	silhouette_coef	K_Means_SSE	K_means silhouette
10	4	roulette	/	0.05	2	14	0.63	6.98	0.797
4	4	roulette	/	0.1	2	7	0.5	6.98	0.797
4	4	tournament	2	0.05	2	6.98	0.504	6.98	0.797
10	10	tournament	2	0.1	2	12.12	0.63	6.98	0.797

Slika 21: Rezultati genetskog algoritma

Algoritam KMeans daje sledeće rezultate:




```
model.inertia_
```

```
6.982216473785234
```

```
silhouette_score(x_pca, model.labels_)
```

```
0.7976752476198908
```

Slika 22: KMeans - Iris

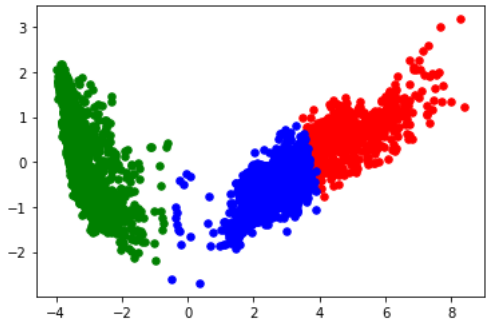
3.5.2 Human Activity Recognition with Smartphones

Sledeća tabela prikazuje rezultate genetskog algoritma:

max_iter	pop_size	category	t_size	mutation_rate	elitism_size	sse	silhouette_coef	K_Means_SSE	K_means silhouette
10	4	roulette	/	0.05	2	22641	0.498	19748	0.387
4	4	roulette	/	0.1	2	18080	0.2	19748	0.387
4	4	tournament	2	0.05	2	19757	0.359	19748	0.387
10	10	tournament	2	0.1	2	15688	0.092	19748	0.387

Slika 23: Rezultati genetskog algoritma

Algoritam KMeans daje sledeće rezultate:



```
model.inertia_
```

```
19748.68758511227
```

```
silhouette_score(x_pca, model.labels_)
```

```
0.38725388086910795
```

Slika 24: KMeans

4 Zaključak

Algoritam radi bar približno dobro kao i originalni KMeans algoritam. Kada su klasteri lepo razdvojeni, algoritam je u stanju da brzo nađe optimalno rešenje. Algoritam ne radi lepo za globularne klastere. Algoritam je u stanju da obradi klastere različitih gustina.

Literatura

[1] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. *Introduction to Data Mining, 2nd Edition.*

[2] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Genetic algorithm-based clustering technique.

[3] Predrag Janičić and Mladen Nikolić. *Veštačka inteligencija.*