

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U
OSIJEKU ELEKTROTEHNIČKI FAKULTET**

Diplomski studij računarstva

Seminarski rad
Grupiranje podataka

Dumančić Andrija, DRB

Osijek, 15.09.2016

Sadržaj

1. Uvod	3
2. Metode grupiranja podataka	4
3. K-means (K srednjih vrijednosti)	10
3.1. Primjer pseudo koda K-means algoritma.....	11
3.2. Određivanje parametra modela	13
3.3 Primjer K-means algoritma	16
3.4. Specifičnosti K-means algoritma	18
4. Zaključak.....	21
5. Literatura.....	22

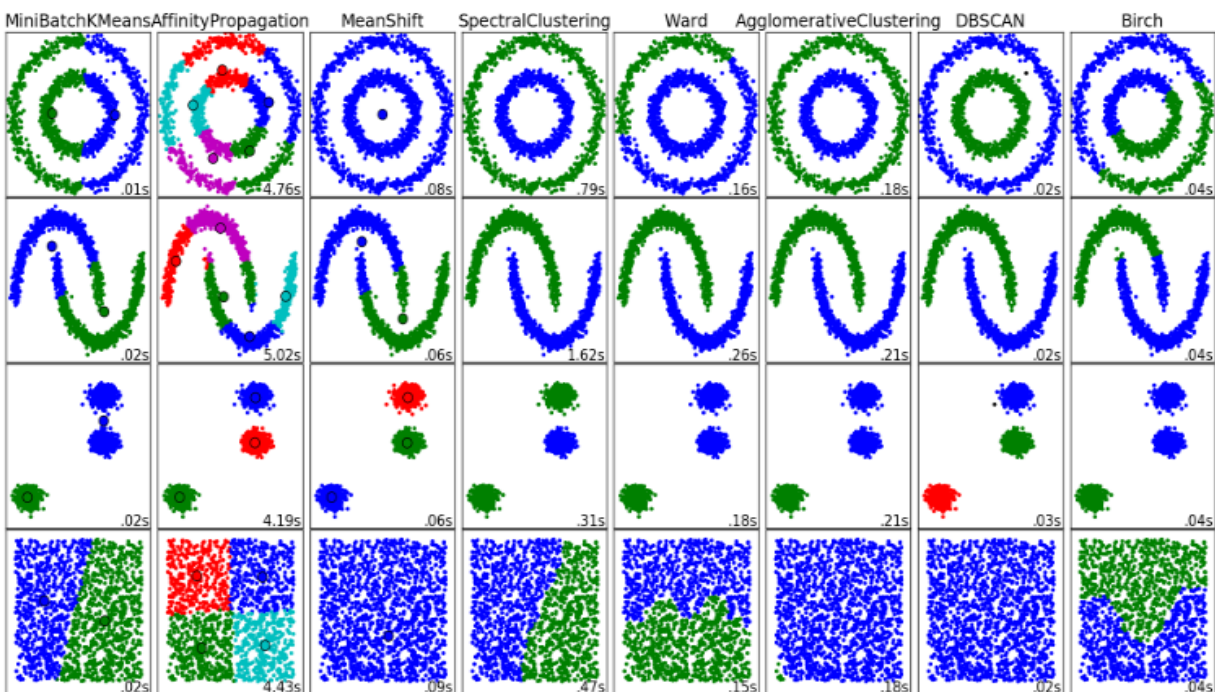
1. Uvod

Grupiranje podataka odnosno klasteriranje (engl. *clustering*) je zadatak kojim je potrebno grupirati podatke na način da su objekti istoj grupi (klasteri) više slični međusobno nego s onima iz druge skupine. To je glavna zadaća grupiranja podataka, a uobičajena tehnika u raznim primjenama. Tehnika se koristi u mnogim područjima poput strojnog učenja, raspoznavanja uzoraka, analize slike, pronalaženja informacija, bioinformatike, kompresije podataka te računalne grafike. Grupiranje nije specifičan algoritam, ali je zadatak koji treba riješiti. Rješenje se može pomoću različitih algoritama koji se znatno razlikuju u njihovom pojmu što čini jednu skupinu te na način kako ih učinkovito pronaći. Klasteri su skupine s malim udaljenostima među članovima (podacima) klastera, gusta područja prostora podataka, intervalne ili pojedine statističke distribucije. Grupiranje se može formulirati u obliku multi-objektivnog problema optimizacije. Klaster analiza je stoga najčešće iterativni proces u kojem se nastoji optimirati dani. Često je potrebno prije samog procesa grupiranja podataka provesti predobradbu raspoloživih podataka, a i vrijednosti različitih parametara postupka grananja je često potrebno mijenjati kako bi se postigao željeni rezultat. Klasteriranje se prvi puta spominje 1932. godine od strane Drivera i Kibera, a u psihologiju ju je uveo Zubin 1938. godine te 1939. godine Robert Tyron. Najpoznatija upotreba klaster analize se dogodila 1943. godine kada ju je Cattell koristio.

2. Metode grupiranja podataka

Postoje razne metode grupiranja_[1] poput:

- K-Means_[2]
- Affinity propagation_[3]
- Mean-shift_[4]
- Spectral clustering_[5]
- Ward hierarchical clustering_[6]
- Agglomerative clustering
- DBSCAN_[7]
- Gaussian mixtures
- Birch_[8]

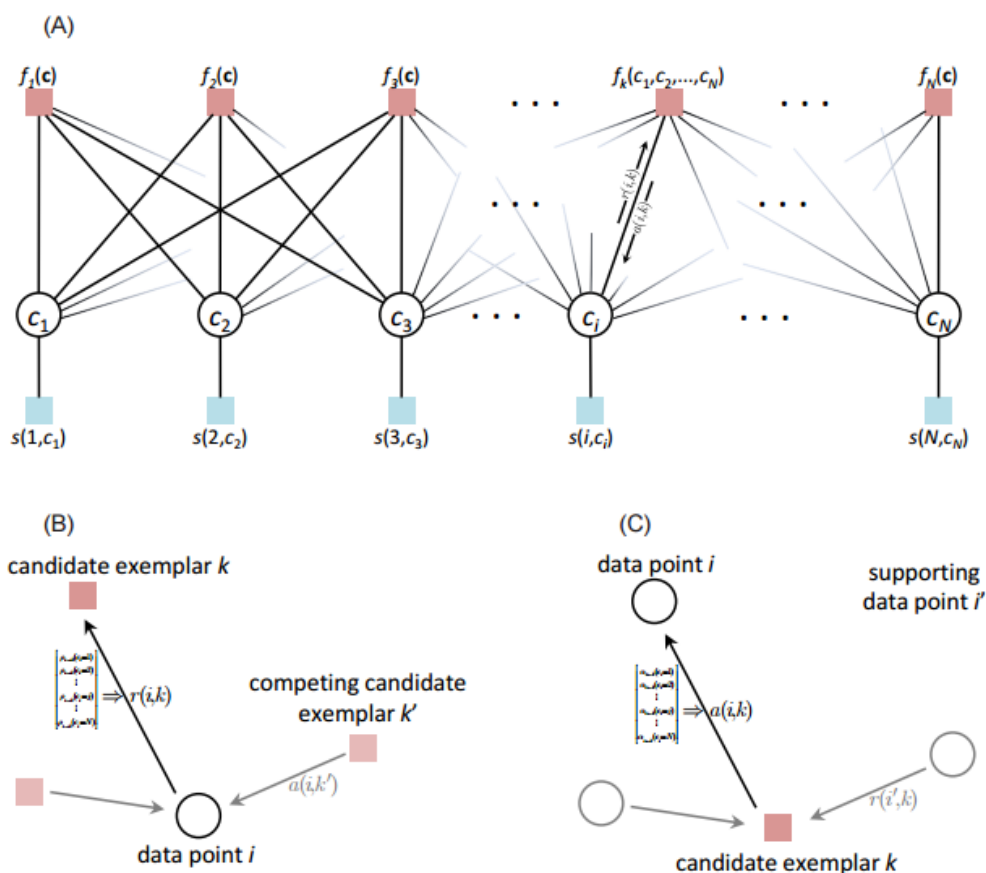


Sl. 2.1. Primjer grupiranja dvodimenzionalnih podataka različitim metodama

Vidi se da metode spectral clustering, agglomerative clustering i pravilno grupiraju podatke dok kod ostalih metoda to nije slučaj.

Affinity propagation [3]

Algoritam na ulazu uzima skup realnih sličnosti između centra podataka $\{s(i, k)\}$, gdje svaka sličnost $s(i, k)$ pokazuje koliko je dobar centar podataka s indeksom k te koliko je pogodan da bude primjer za točke koje prikazuju i . Svaka točka je u paru sa promjenjivim čvorom. Vrijednosti $C_i = k$ te $6 = k$ ukazuju na to da nam se podaci dodjeljuju u skupinu sa točkom k kao njegov uzorak.

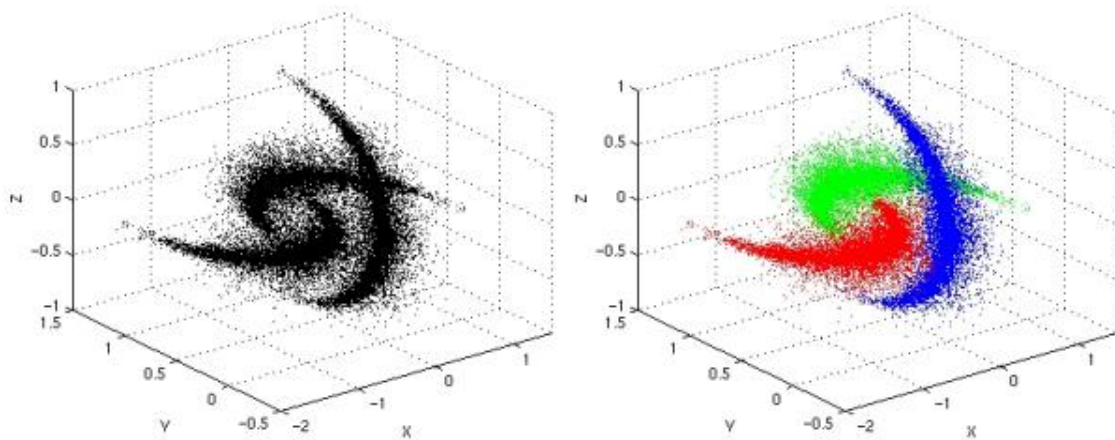


Sl. 2.2. Primjer na bazi klaster algoritma koji obavlja širenje na grafu

Primjer na bazi klastera algoritam koji obavlja širenje na grafu faktor koji je prikazan u (A). Prenose se dvije vrste poruka u grafikonu odgovornosti (B) s promjenjivim čvorovima. Dostupnost se prenosi iz funkcijskih čvorova na promjenjive čvorove (C), koji se očituju s primjerima kandidata podatkovnih točaka.

Mean- shift_[4]

Mean-shift je ne parametarska tehnika za analizu gdje se locira maksimum funkcije gustoće. To je algoritam koji se temelji na centroidu, radi na način da se ažuriraju podaci za centroide koji imaju srednju vrijednost unutar određenog područja. Kandidati se filtriraju u fazi nakon obrade kako bi se uklonili duplikati za konačni skup centroida. Vidljivo je da metoda ne pretpostavlja neki konkretni oblik klastera već može prepoznati i dosta složene klasterne. Za razliku od drugih algoritama ovaj je algoritam neovisan alat pogodan za stvarne analize podataka. Ne uzima se nikakav određen oblik unaprijed na podacima klastera te se postupak oslanja samo na parametar propusnosti.

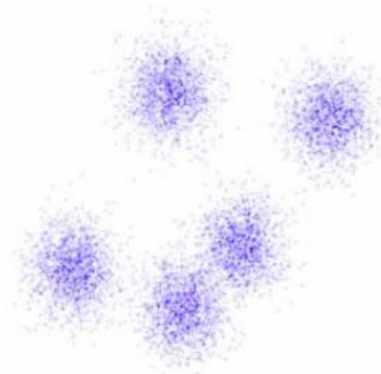


Sl. 2.3. Sintetički primjer tri ne-linearno odvojiva klastera

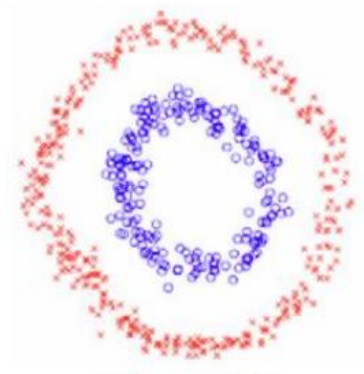
Spectral clustering^[5]

Cilj prilikom realizacije ovog algoritma je okupiti podatke koji su spojeni, ali ne i nužno kopaktni ili grupirani unutar granice.

- Compactness, e.g., k-means, mixture models
- Connectivity, e.g., spectral clustering



Compactness



Connectivity

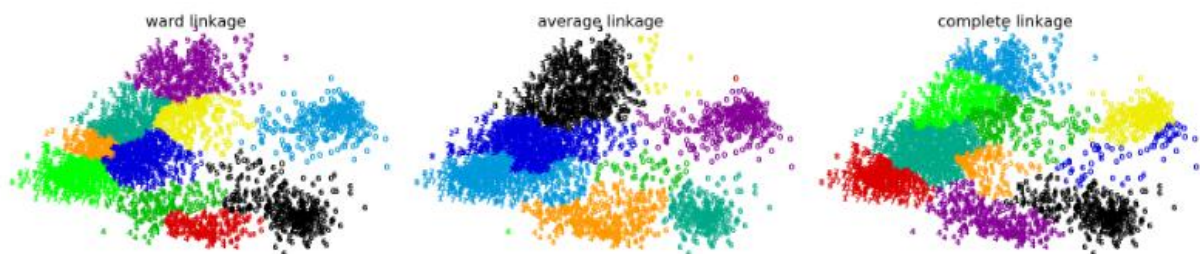
Sl. 2.4. S lijeve strane je prikaz K-means algoritma, dok se sa lijeve nalazi povezanost spektralnim klastiranjem

Hierarchical clustering

Hijerarhijsko klasteriranje_[6] je opća skupina klaster algoritma koja gradi ugniježdene klastere sukcesivnim spajanjem ili razdvajanjem dviju grupa podataka (mjerni uzorak). Tijek grupiranja podataka ovom metodom moguće je prikazati u obliku stabla. Korijen stabla je jedinstveni klaster koji sadrži sve uzorke, listovi čine nakupine od samo jednog uzorka.

Kriterij povezanosti određuje princip koji se koristi za strategiju spajanja:

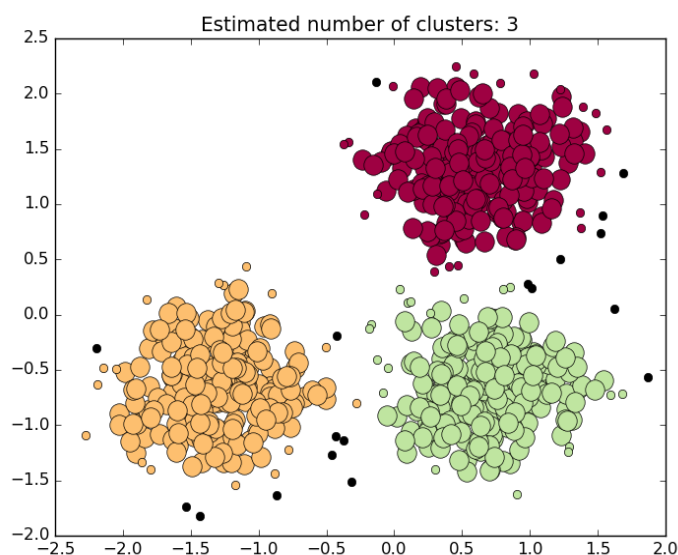
- Ward minimizira zbroj kvadrata unutar svih klastera.
- Maksimalna ili potpuna veza smanjuje maksimalnu udaljenost između klastera.
- Prosjek povezanosti smanjuje prosječnu udaljenost između svih parova klastera.



Sl. 2.5. Razne vrste povezivanja opcija klastera kod 2D postavljanja skupa podataka

DBSCAN_[7]

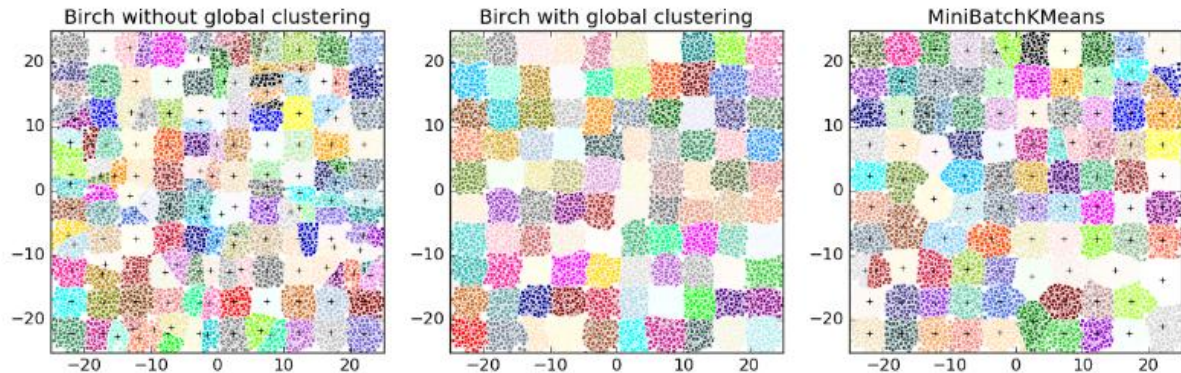
Ovaj algoritam gleda klastere kao područja visoke gustoće odvojenih od područja sa niskom gustoćom. Zbog tog općenitog pogleda, klasteri pronađeni pomoću DBSCAN algoritma mogu biti bilo kakvog oblika. Središnja komponenta DBSCAN algoritma je koncept osnovnih uzoraka, a to su uzorci koji se nalaze u području visoke gustoće. Klaster je skup osnovnih uzoraka koji se nalaze jedan do drugoga. Ovaj algoritam ima dva parametra, a to su `min_samples` koji je minimalni broj podataka koji mora biti na manjoj udaljenosti od `eps` u okolini nekog drugog podatka da bi se taj podatak smatrao osnovnim uzorkom.



Sl. 2.6. Uzorci visoke gustoće klastera spremnih za proširivanje

Birch_[8]

Birch izgrađuje stablo nazvano CFT (Characteristic Feature Tree) za dobivene podatke. Algoritam je bez nadzorna vrsta grupiranja podataka koji se koristi kod hijerarhijskih klastera tijekom velikih skupina podataka



Sl. 2.7. Usporedba Birch i Mini K-means algoritma

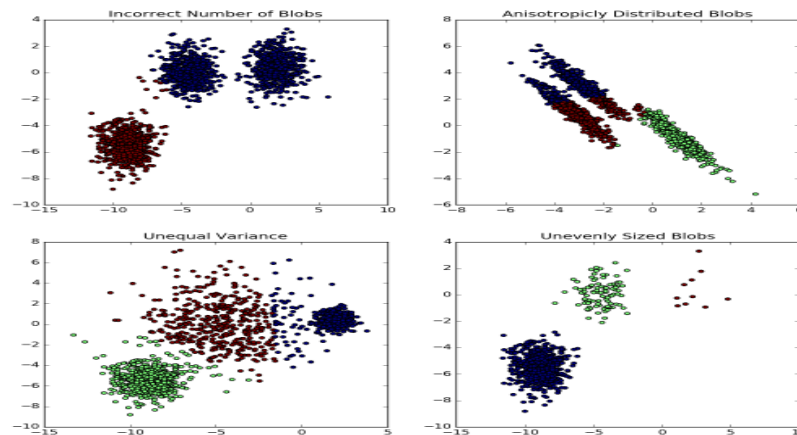
3. K-means (K srednjih vrijednosti)

Podaci pomoću algoritma K srednjih vrijednosti_[3] pokušavaju razdvojiti uzorke u n grupa jednakih varijanci, umanjuje se kriterij poznat kao inercija ili unutar klasterni zbroj kvadrata. Ovaj algoritam zahtjeva da se zada određeni broj klastera. Ovaj algoritam zahtjeva veliki broj uzoraka i korišten je na području velikog raspona u mnogim različitim područjima. Algoritam K srednjih vrijednosti dijeli skup od N uzoraka X u K disjunktne klastera C , svaki je opisan sa srednjom vrijednosti centra klastera. Algoritam K srednjih vrijednosti odabire centroid koji minimiziraju inerciju ili unutar klasterni zbroj kvadrata.

$$\sum_{i=0}^n \min(\|x_j - \mu_i\|^2)$$

K srednjih vrijednosti se još nazivaju i kao Lloyd algoritam. To je algoritam od tri koraka. Prilikom prvog koraka odabiru se inicijalni centroidi te se uz najosnovnije metode odabiru uzorci k iz skupa podataka X . Nakon inicijalizacije, K srednjih vrijednosti sastoji se od petlje između

druga dva koraka. Prvi korak dodjeljuje svaki uzorak najbližem centru. Drugi korak stvara nove centroide uzimanjem srednjih vrijednosti svih uzoraka dodijeljenih svakom prethodnom težištu. Razlika između starih i novih centroida se računa i algoritam ponavlja posljednja dva koraka sve dok je ta vrijednost manja od praga. Nakon toga centroidi ažuriraju prosjek svakom segmentu. Algoritam ponavlja sve dok jedan od kriterija zaustavljanja nije ispunjen. Algoritam se zaustavlja kada je relativno smanjenje funkcije cilja između ponavljanja manja od zadane tolerancije vrijednosti.



Sl. 3.1. Primjeri rješenja K-means algoritma

3.1. Primjer pseudo koda K-means algoritma

Primjer:

Let n be the number of clusters you want

Let S be the set of feature vectors ($|S|$ is the size of the set)

Let A be the set of associated clusters for each feature vector

Let $\text{sim}(x,y)$ be the similarity function

Let $c[n]$ be the vectors for our clusters

Init:

Let $S' = S$

//choose n random vectors to start our clusters,

for $i=1$ to n

$j = \text{rand}(|S'|)$

$c[n] = S'[j]$

```
    S' = S' - {c[n]} //remove that vector from S' so we can't choose it  
again end
```

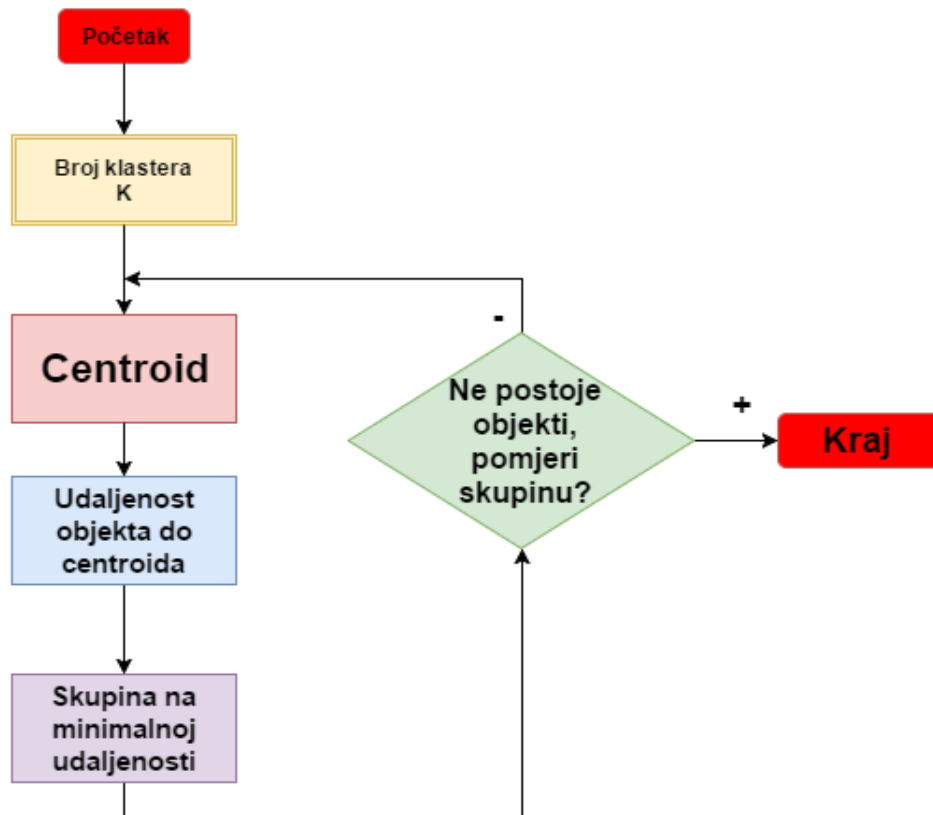
```
    //assign initial clusters  
    for i=1 to |S|  
        A[i] = argmax(j = 1 to n) { sim(S[i], c[j]) }  
    end
```

Run:

```
    Let change = true  
    while change  
        change = false //assume there is no change  
  
        //reassign feature vectors to clusters  
        for i = 1 to |S|  
            a = argmax(j = 1 to n) { sim(S[i], c[j]) }  
            if a != A[i]  
                A[i] = a  
                change = true //a vector changed affiliations -- so we need to  
                //recompute our cluster vectors and run again  
            end  
        end  
    end  
  
    //recalculate cluster locations if a change occurred  
    if change  
        for i = 1 to n  
            mean, count = 0  
            for j = 1 to |S|  
                if A[j] == i  
                    mean = mean + S[j]  
                    count = count + 1  
                end  
            end  
            c[i] = mean/count  
        end  
    end  
end
```

3.2. Određivanje parametra modela tj. centroida

Primjer_[12] određivanja centroida u K-means algoritmu na osnovu sljedećeg skupa podataka koji se sastoji od dvije varijable (A i B) za sedam različitih osoba.



Sl. 3.2. Blok dijagram K- means algoritma

SUBJECT	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Tab 3.1

Podaci su grupirani u dva klastera. Prvi korak je pronalazak razumne inicijalne particije, neka vrijednost A i B za dvije osove koje su najdalje (koristimo Euklidovo mjerenje udaljenosti) te definiramo inicijalna klaster sredstva.

	Individual	Mean vector (centroid)
Group 1	1	(1.0 1.0)
Group 2	4	(5.0 7.0)

Tab 3.2

Preostali pojedinci se ispituju u nizu te su smješteni do klastera koji su najbliži, u smislu Euklidove udaljenosti. Srednji vektor se izračunava svaki puta kada se dodaje novi član. To dovodi do sljedećeg niza koraka.

	Cluster 1		Cluster 2	
	Individual	Mean vector (centroid)	Individual	Mean vector (centroid)
1	1	(1.0 1.0)	4	(5.0 7.0)
2	1, 2	(1.2 1.5)	4	(5.0 7.0)
3	1, 2, 3	(1.8 2.3)	4	(5.0 7.0)
4	1, 2, 3	(1.8 2.3)	4, 5	(4.2 6.0)
5	1, 2, 3	(1.8 2.3)	4, 5, 6	(4.3 5.7)
6	1, 2, 3	(1.8 2.3)	4, 5, 6, 7	(5.1 5.4)

Tab 3.3

Nakon što je inicijalna particija promjenjena, dva klastera u ovoj fazi dobijaju sljedeće karakteristike:

	Individual	Mean vector (centroid)
Cluster 1	1, 2, 3	(1.8 2.3)
Cluster 2	4, 5, 6, 7	(4.1 5.4)

Tab 3.4

Nakon ovog koraka još ne možemo sa sigurnošću reći koji pojedinac je dodjeljen pravom klasteru. Nastavljamo uspoređivati svaku udaljenost pojedinca sa njegovim klasterom te sa suprotnim klasterom te pronalazimo sljedeće rješenje.

Individual	Distance to mean (centroid) of Cluster 1	Distance to mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Tab 3.5

Vidimo da je pojedinac tri najbliži suprotnom klasteru (Klaster 2) nego njegovom klasteru (Klasteru 1). Drugim riječima, svaka udaljenost pojedinca za njego klaster bi trebala biti manja nego udaljenost do suprotnog klastera (što nije slučaj sa pojedincem 3). Pojedincu tri se dodaje nova lokacija u klaster 2 što dovodi do novog konačnog rješenja.

	Individual	Mean vector (centroid)
Cluster 1	1, 2	(1.3 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9 5.1)

Tab 3.6

3.3 Primjer K-means algoritma

K srednjih vrijednosti je izveden u pythonu 2.7. pomoću sljedećeg koda koji se nalazi ispod:

```
"""
@author: Andrija
"""

import numpy as np
import pylab as plt

mean = [0, 0]
cov = [[320, 10], [200, 100]]
mean2 = [0, 1]
cov2 = [[100, 20], [100, 400]]
mean3 = [0, 1]
cov3 = [[300, 35], [5, 100]]

K = 3
maxIters = 10

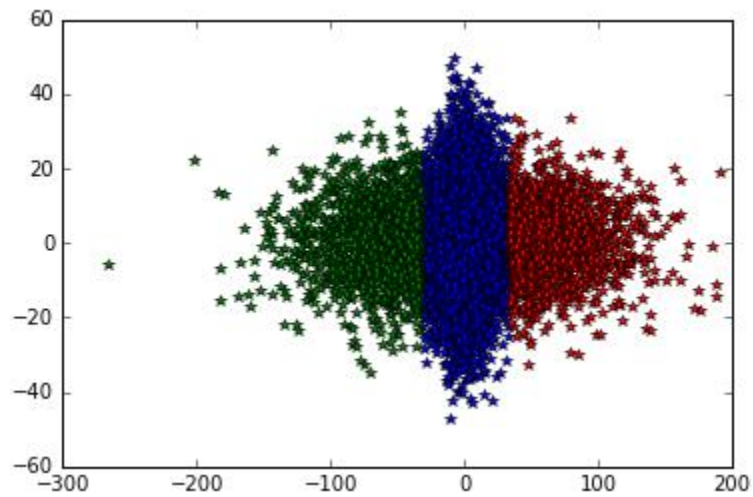
data1 = np.random.multivariate_normal(mean, cov, 50)
data2 = np.random.multivariate_normal(mean2, cov2, 5000)
data3 = np.random.multivariate_normal(mean2, cov3, 5000)
X = np.vstack((data1, np.vstack((data2, data3))))
np.random.shuffle(X)

centroids = X[np.random.choice(np.arange(len(X)), K), :]
for i in range(maxIters):
    C = np.array([np.argmin([np.dot(x_i - y_k, x_i - y_k) for y_k in centroids])
                  for x_i in X])
    centroids = [X[C == k].mean(axis=0) for k in range(K)]
    centroids = np.array(centroids)

plt.ion()
plt.cla()
plt.plot(X[C == 0, 0], X[C == 0, 1], '*b',
X[C == 1, 0], X[C == 1, 1], '*r',
X[C == 2, 0], X[C == 2, 1], '*g')
plt.draw()
plt.ioff()
plt.show()
```

Listing 3.1. Primjer grupiranja podataka u Pythonu

Rezultat koji smo dobili prilikom realizacije gore navedenog koda:



Promjenom dolje navedenih vrijednosti se dobije i drugačija realizacija K srednjih vrijednosti:

$\text{mean} = [0, 0]$

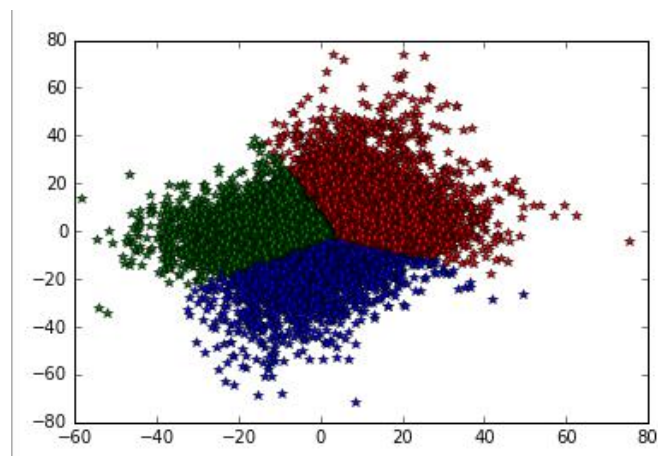
$\text{cov} = [[320, 10], [200, 100]]$

$\text{mean2} = [0, 1]$

$\text{cov2} = [[100, 20], [100, 400]]$

$\text{mean3} = [0, 1]$

$\text{cov3} = [[300, 35], [5, 100]]$



3.4. Specifičnosti K-means algoritma

- Potrebno je unaprijed zadati broj klastera i njihove početne vrijednosti

mean = [0, 1]

cov = [[320,200], [200, 100]]

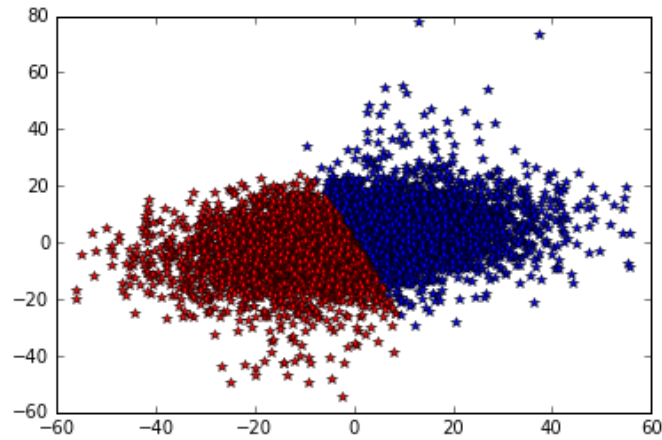
mean2 = [0, 1]

cov2 = [[100, 120], [100, 400]]

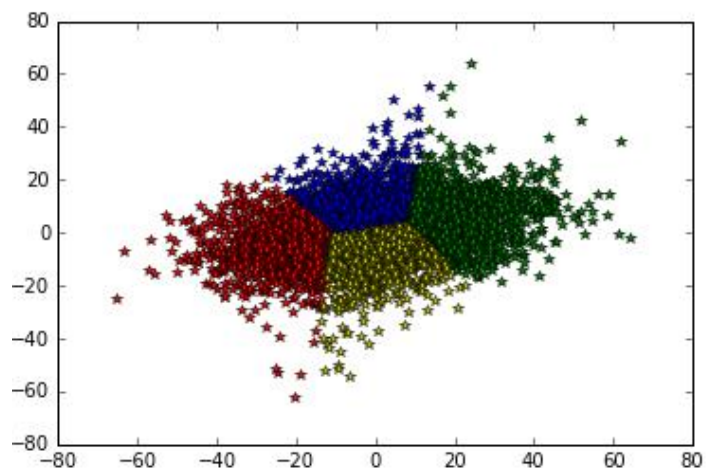
mean3 = [0, 1]

cov3 = [[300, 35], [100, 100]]

K = 2 => *Određivanje broja klastera*



Rješenje sa 2 klastera

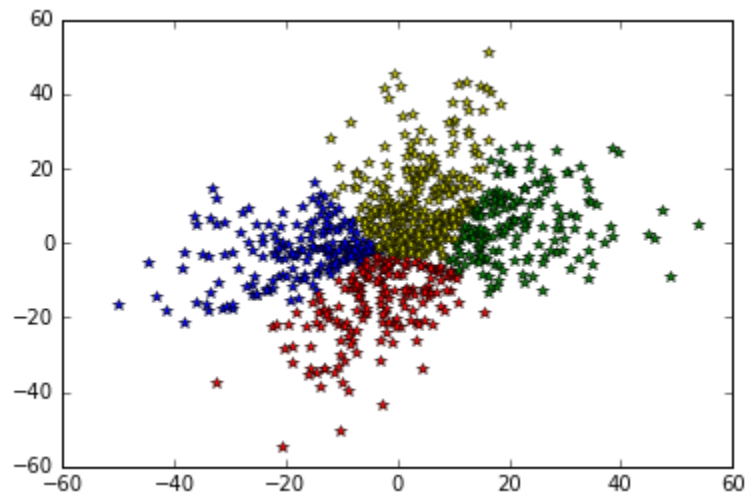


Rješenje sa 4 klastera

- Određivanje gustoće K-means klastera

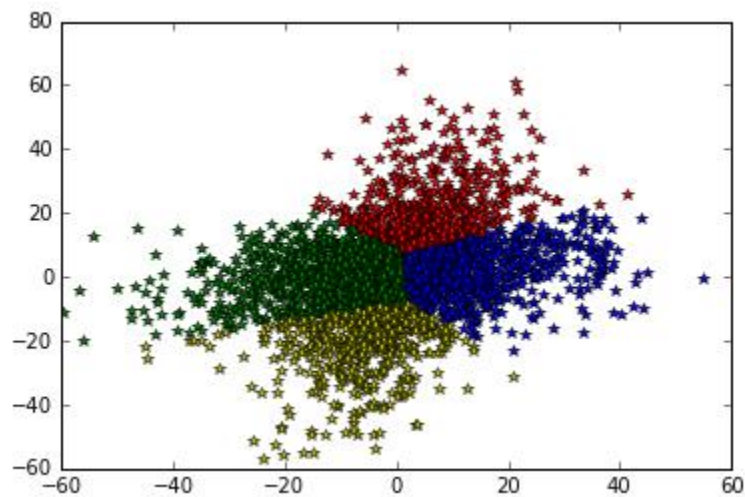
- Rješenje sa parametrima data 1 (50), data2 (300), data 3(500)

```
data1 = np.random.multivariate_normal(mean, cov, 50)
data2 = np.random.multivariate_normal(mean2, cov2, 300)
data3 = np.random.multivariate_normal(mean2, cov3, 500)
X = np.vstack((data1, np.vstack((data2, data3))))
np.random.shuffle(X)
```

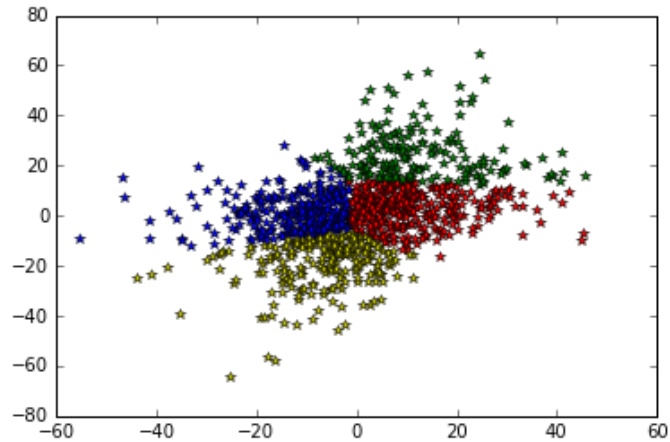


- Rješenje sa parametrima data 1 (50), data2 (1300), data 3(1500)

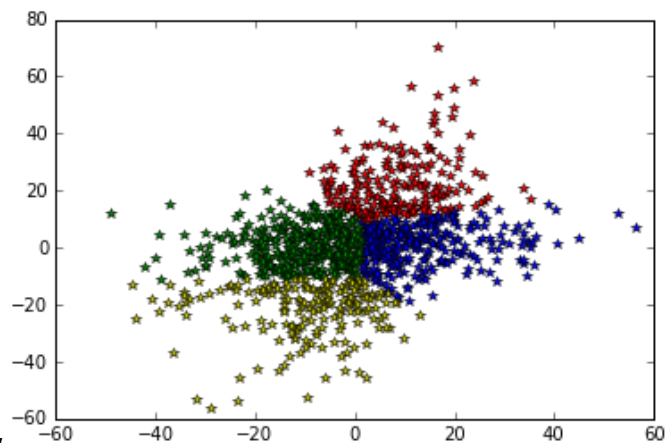
```
data1 = np.random.multivariate_normal(mean, cov, 50)
data2 = np.random.multivariate_normal(mean2, cov2, 1300)
data3 = np.random.multivariate_normal(mean2, cov3, 1500)
```



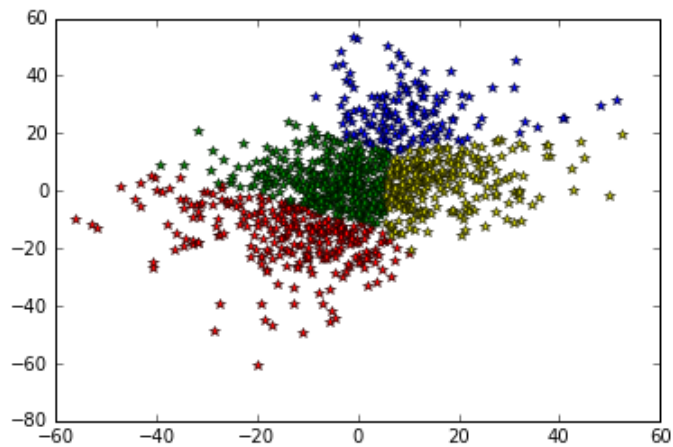
- *Rješenje nije uvijek isto*
-Rješenje prilikom izvršavanja K-means algoritma nam nije uvijek isto kao što vidimo u primjeru ispod gdje su parametri uvijek bili jednaki.



Prilikom prvog pokretanja



Prilikom drugog pokretanja



Prilikom trećeg pokretanja

4. Zaključak

Izrada projektnog zadatka mi je pomogla u upoznavanju pythona kojeg nisam koristio u velikoj mjeri. Prilikom ovog zadatka susreo sam se sa mnogim problemima poput same realizacije koda ali sam i naučio mnoge vrste grupiranja podataka među kojima sam se najviše posvetio K-means algoritmu. Prilikom izvršavanja K means algoritma najbolji pregled sam imao kad su postojale male količine uzoraka jer tada sam dobio preglednija rješenja iako se sa velikim količinama uzoraka dobijaju bolja rješenja. Jedna od stvari koje su bitne kod K-means algoritma je ta što je potrebno unaprijed zadati broj klastera koji će se realizirati. Također, prilikom svakog izvršavanja algoritma dobili smo drugačiji izgled rješenja iako su parametri ostajali isti što znači da nikad nećemo dobiti isto rješenje.

5. Literatura

- [1] <http://scikit-learn.org/stable/modules/>
Pristupio: 15.09.2016
- [2] <http://scikit-learn.org/stable/modules/clustering.html#k-means>
Pristupio: 15.09.2016
- [3] <http://scikit-learn.org/stable/modules/clustering.html#affinity-propagation>
Pristupio: 15.09.2016
- [4] <http://scikit-learn.org/stable/modules/clustering.html#mean-shift>
Pristupio: 15.09.2016
- [5] <http://scikit-learn.org/stable/modules/clustering.html#spectral-clustering>
Pristupio: 15.09.2016
- [6] <http://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering>
Pristupio: 15.09.2016
- [7] <http://scikit-learn.org/stable/modules/clustering.html#dbscan>
Pristupio: 15.09.2016
- [8] <http://scikit-learn.org/stable/modules/clustering.html#birch>
Pristupio: 15.09.2016
- [9] https://en.wikipedia.org/wiki/K-means_clustering
Pristupio: 15.09.2016
- [10] https://en.wikipedia.org/wiki/Affinity_propagation
Pristupio: 15.09.2016
- [11] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.150.5946&rep=rep1&type=pdf>
Pristupio: 26.09.2016
- [12] <http://mnemstudio.org/clustering-k-means-example-1.htm>
Pristupio: 26.09.2016