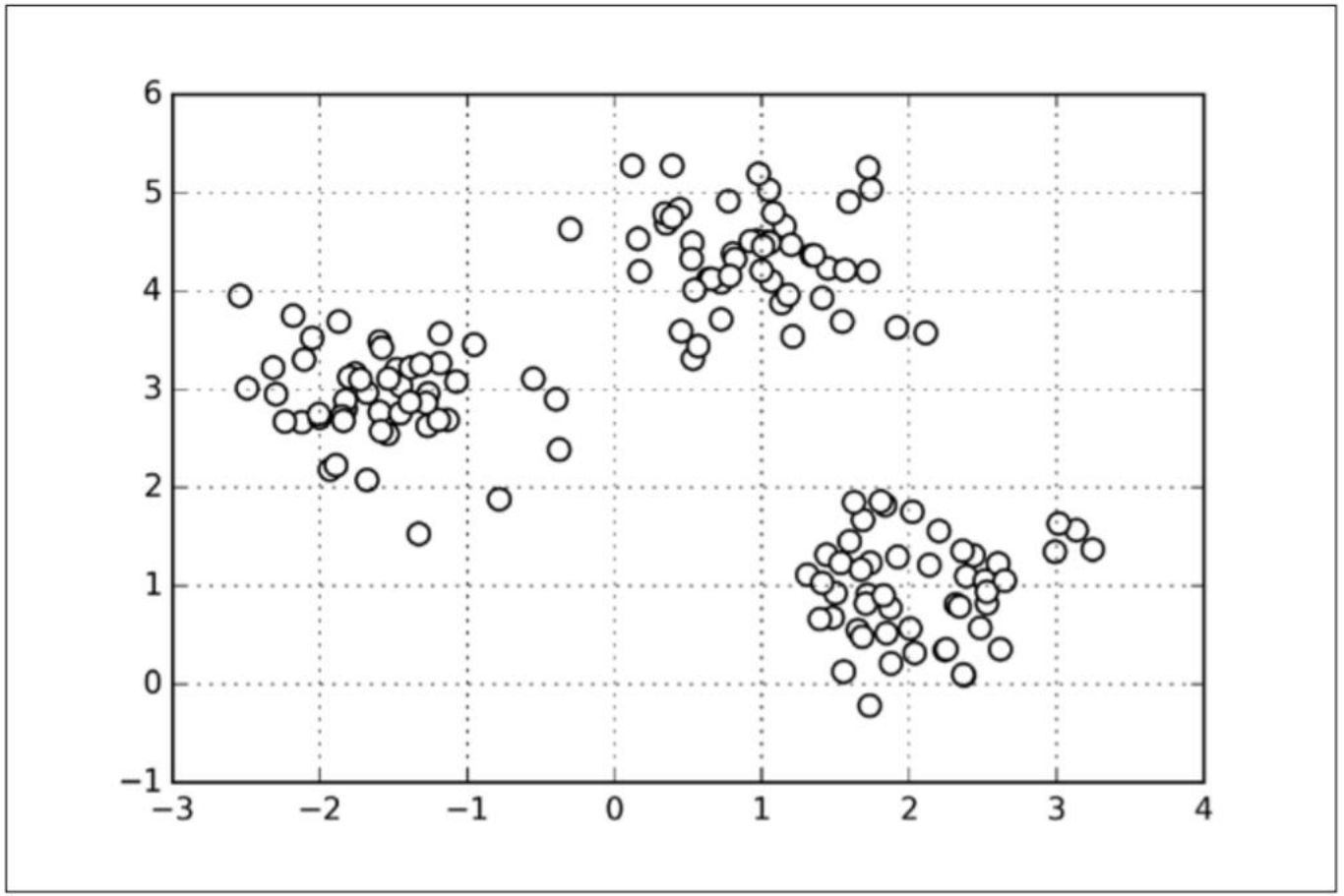
k-Srednjih vrednosti (k-Means)

TEHNIKA KLASTEROVANJA UZORAKA

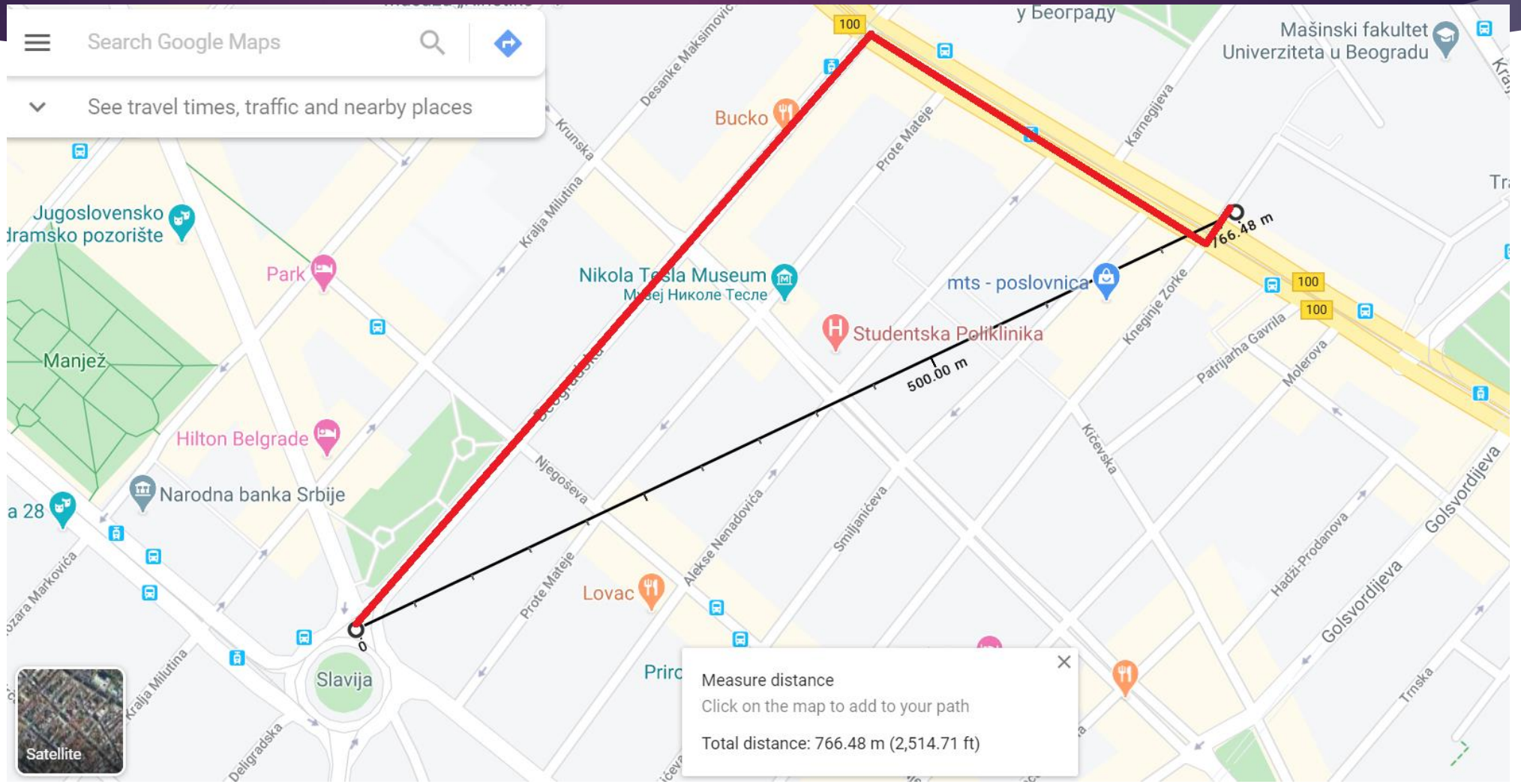
Klasterovanje



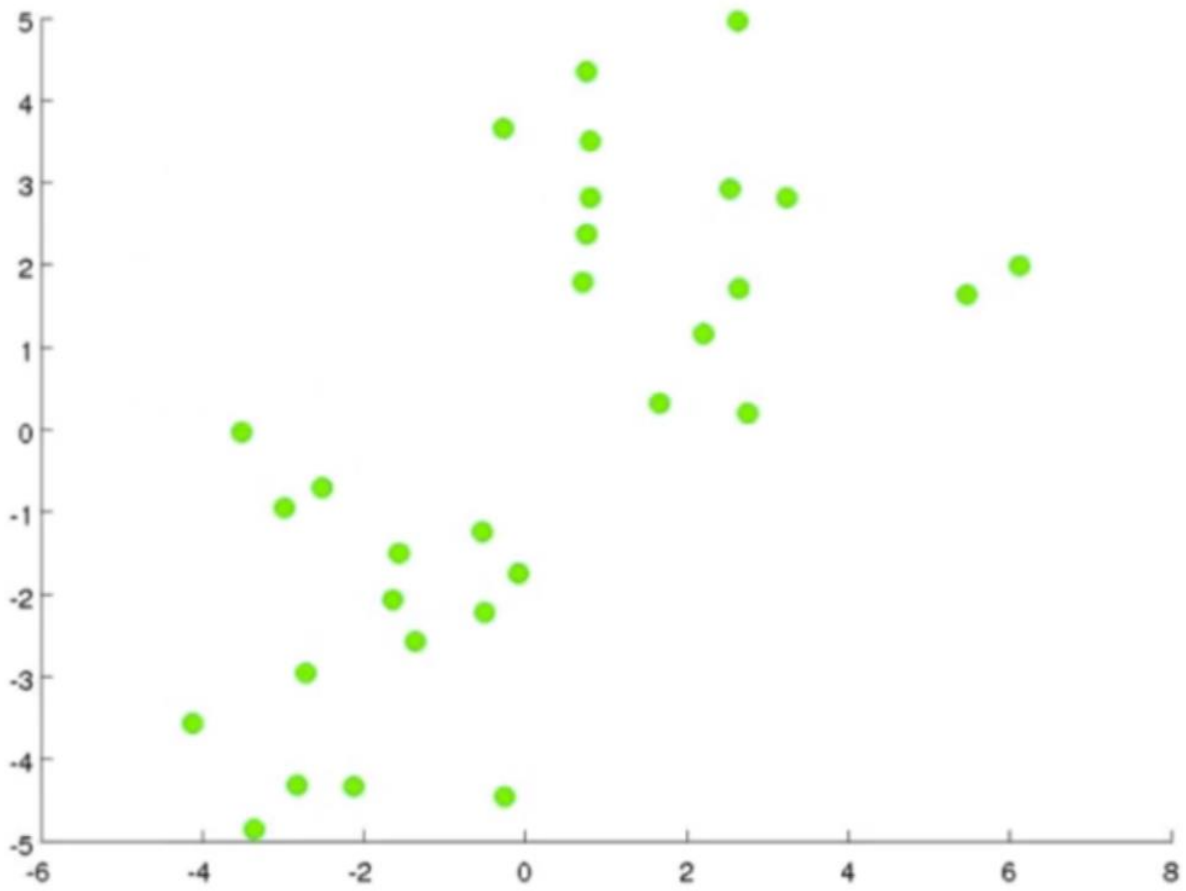
K-means

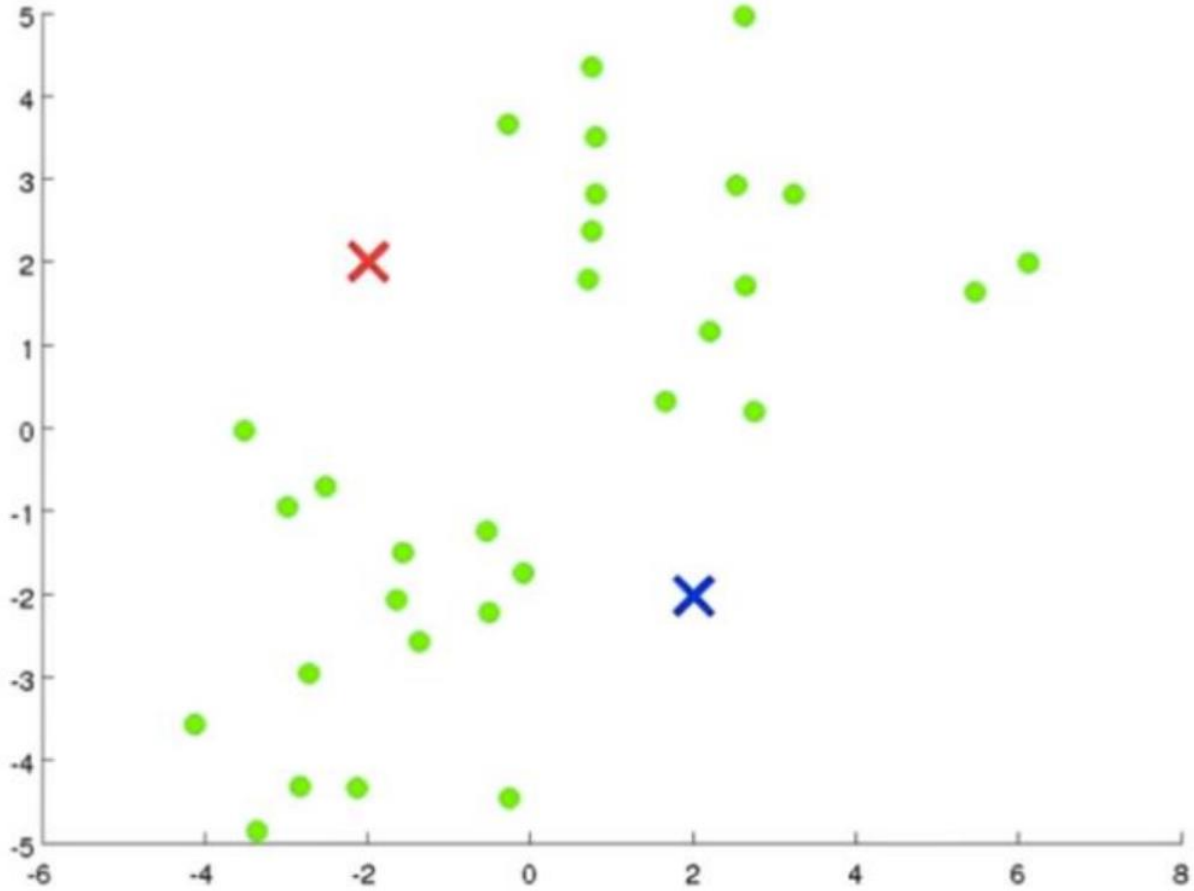
- Najpoznatiji i najjednostavniji algoritam klasterovanja (klasterizacije)
- Pripada tehnikama nenadgledanog učenja
- Sličnost instanci se može procenjivati primenom neke od mera za računanje:
  - Sličnosti (npr. kosinusna sličnost ili koeficijent korelacije)
  - Udaljenosti dve instance (npr. Euklidsko ili Menhetn rastojanje)

Rastojanje

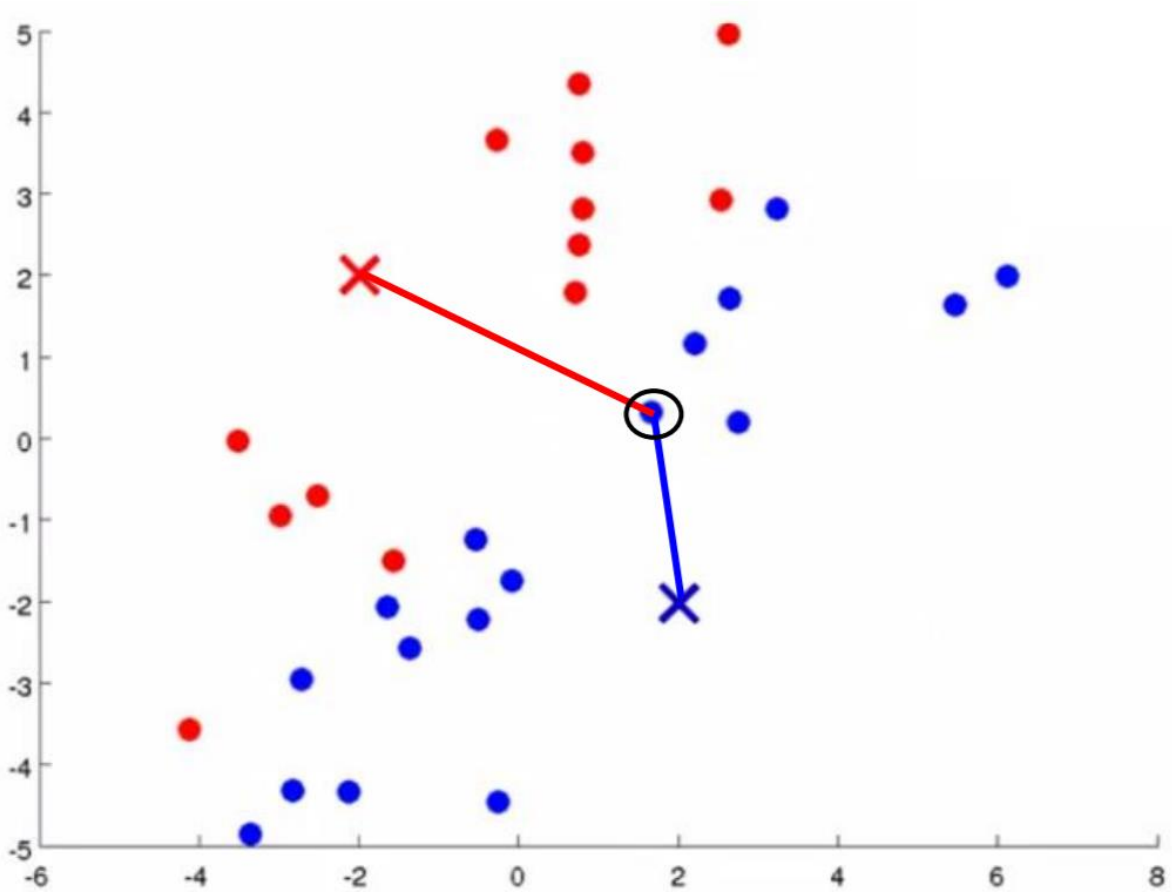

Search Google Maps
See travel times, traffic and nearby places
Bucko
Jugoslovensko dramsko pozorište
Park
Manjež
Hilton Belgrade
Narodna banka Srbije
Nikola Tesla Museum
Музеј Николе Тесле
Studentska Poliklinika
mts - poslovnica
Mašinski fakultet Univerziteta u Beogradu
Lovac
Slavija
Satellite
766.48 m
500.00 m
Measure distance
Click on the map to add to your path
Total distance: 766.48 m (2,514.71 ft)

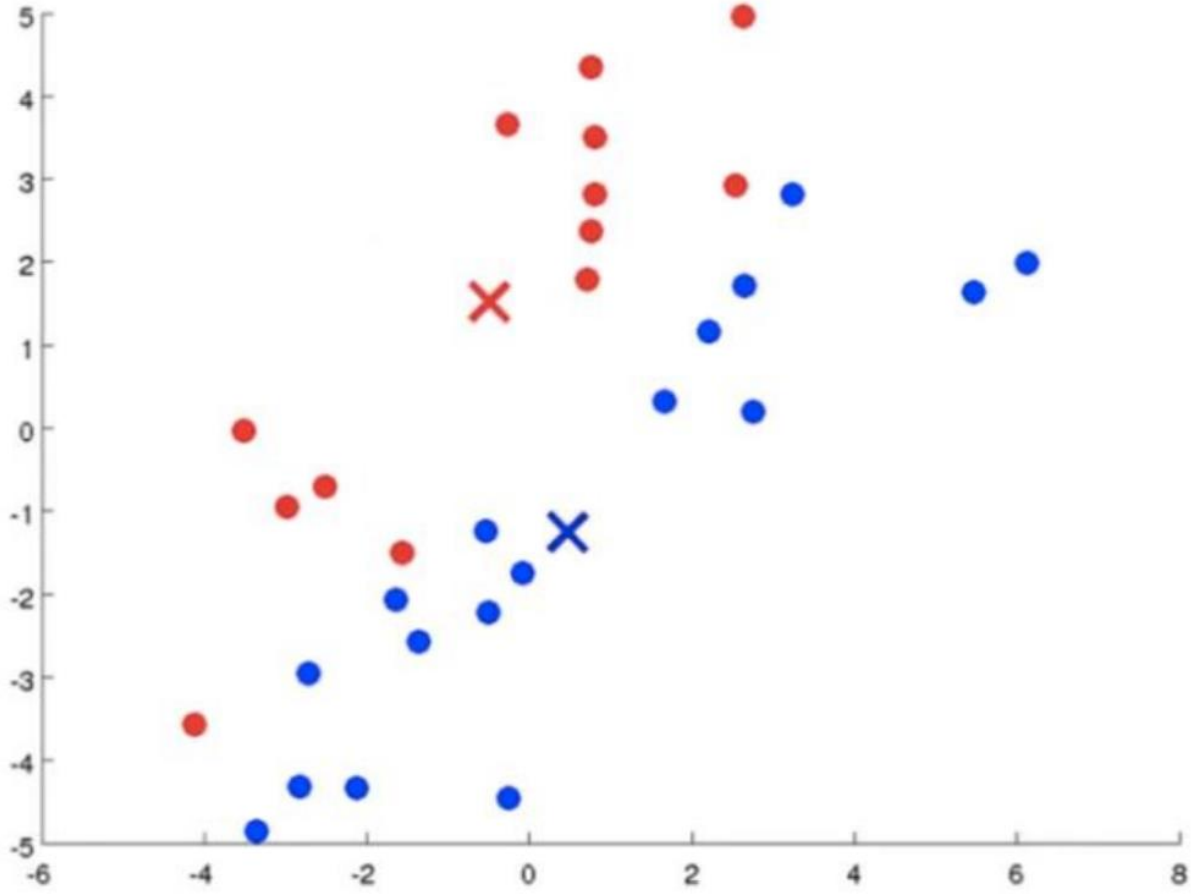
K-means (1) - Inicijalno



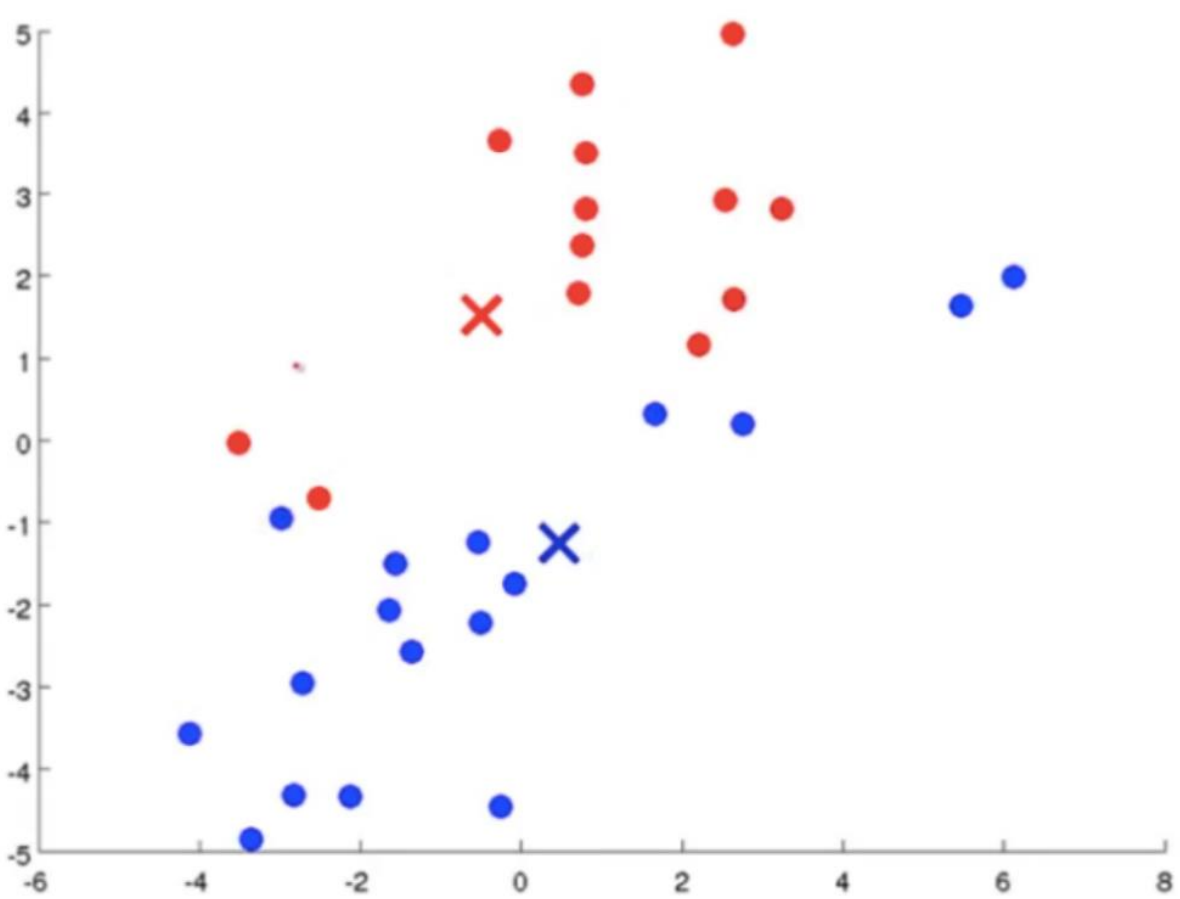


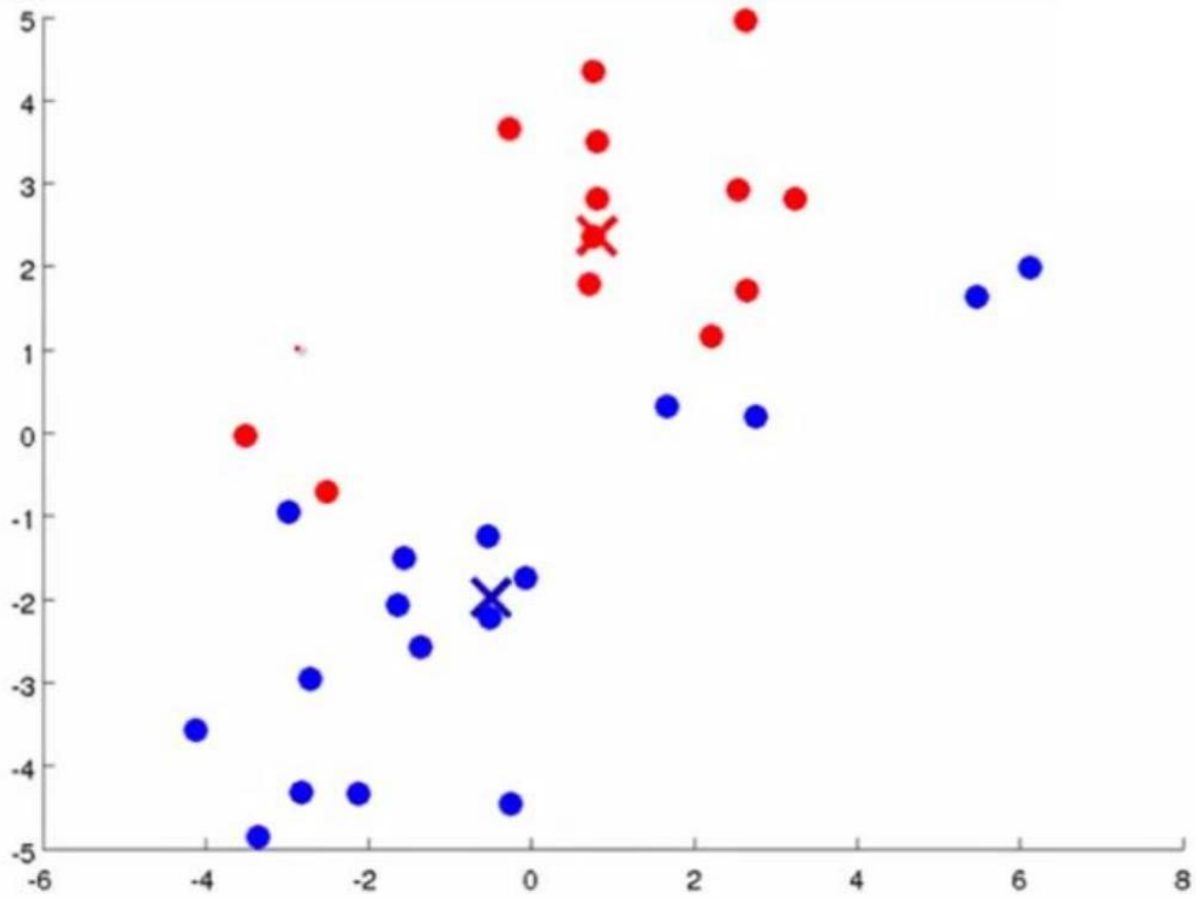
K-means (2) – Iteracija br. 1



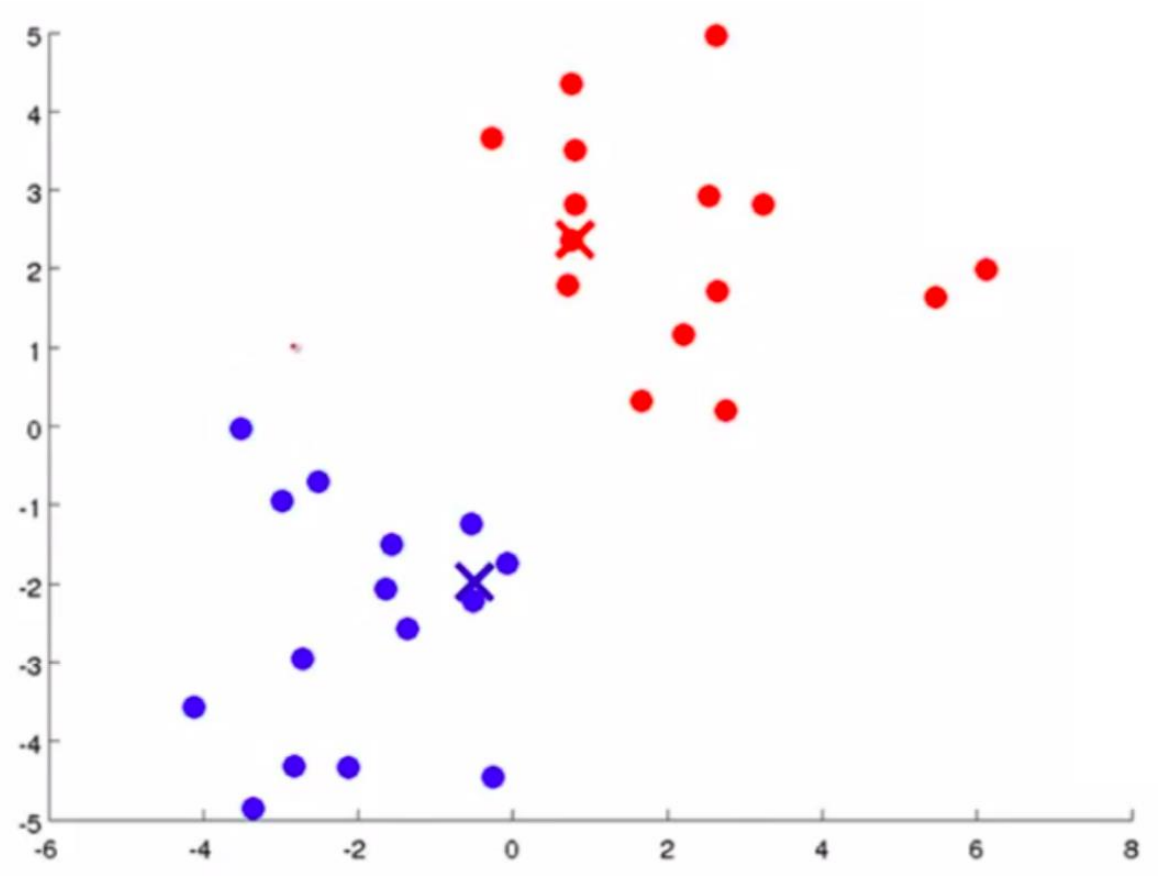


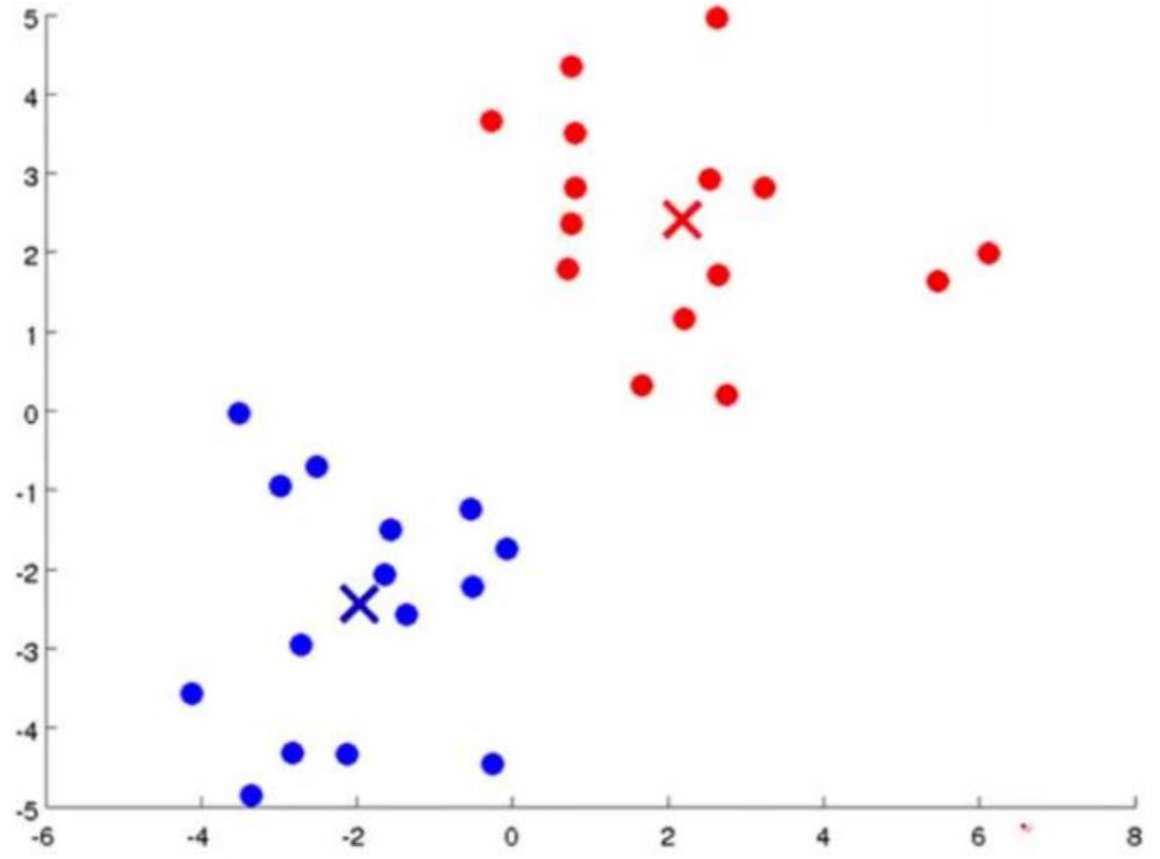
K-means (3) – Iteracija br. 2



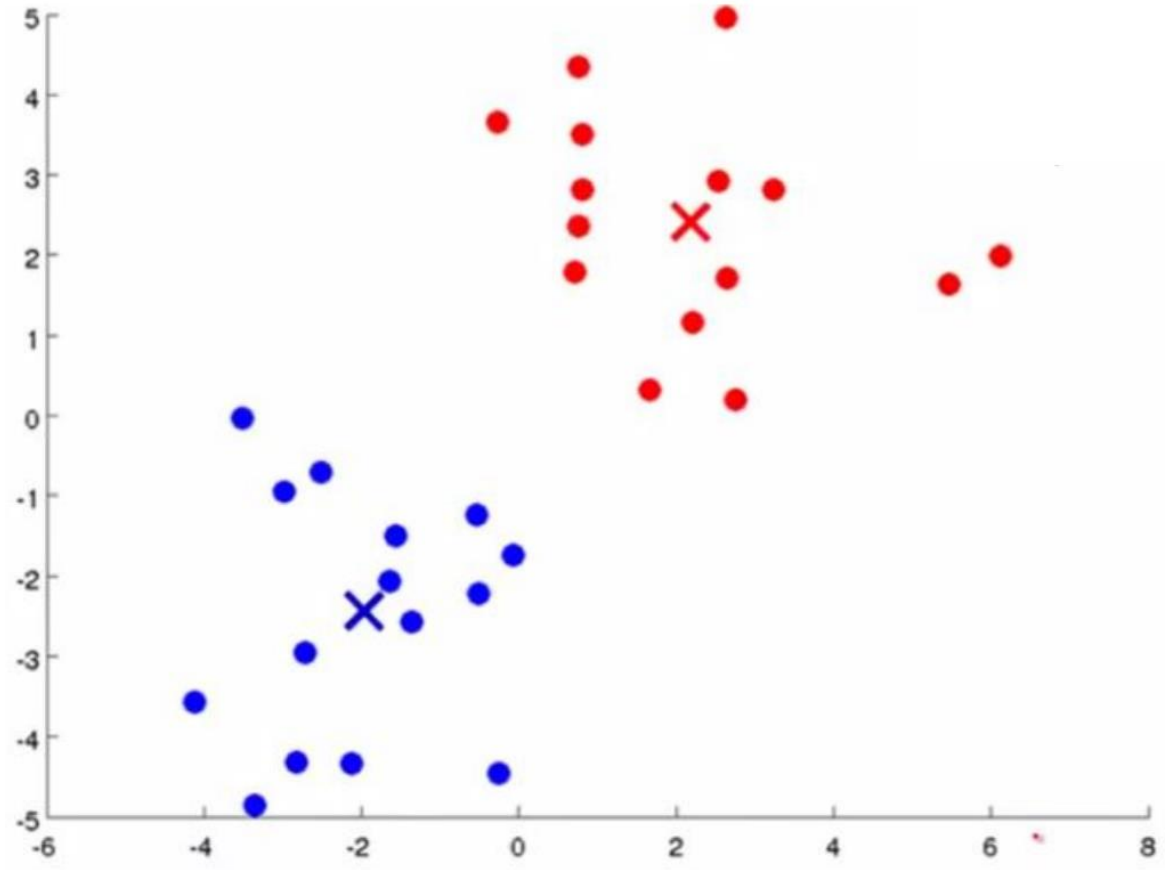


K-means (4) – Iteracija br. 3





K-means (5) - Završetak



K-means - Ulazni podaci

- Ulaz:
  - K – broj klastera
  - skup za trening sa m uzoraka (instanci), a svaki uzorak u skupu je vektor opisan sa n atributa (x1, x2, ... Xn)
  - Opciono: maksimalan broj iteracija koji se izvršava (max)

K-means – Koraci algoritma

- 1) Slučajan odabir K težišta klastera (centroida)
- 2) Grupisanje po klasterima - svaki uzorak (instancu) dodeljujemo najbližem centroidu (najbliže težište na osnovu odabrane metrike)
- 3) Pomeranje težišta (centroida) ka centru svih uzoraka u klasteru – za svaki klaster se izračuna novo težište uzimajući prosek instanci koje su dodeljene tom klasteru
- Koraci 2) i 3) se ponavljaju sve dok algoritam ne konvergira ili broj iteracija ne dostigne MAX

K-means - Funkcija koštanja/distorzije

- Smisao K-means algoritma je minimizacija funkcije koštanja J (eng. cost function):

$$J\left(c^{(1)}, \dots, c^{(m)}, \mu_{(1)}, \dots, \mu_K\right) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2$$

$x^{(i)}$ – i-ta instanca u skupu podataka za trening, i = 1,...m
$c^{(i)}$ – indeks klastera u koji je instanca $x^{(i)}$ trenutno raspoređena
μ_j – težište klastera j, j = 1,... K, gde je K ukupan broj klastera
$\mu_{c(i)}$ – težište klastera u koji je instanca $x^{(i)}$ trenutno raspoređena

Minimizacija funkcije koštanja

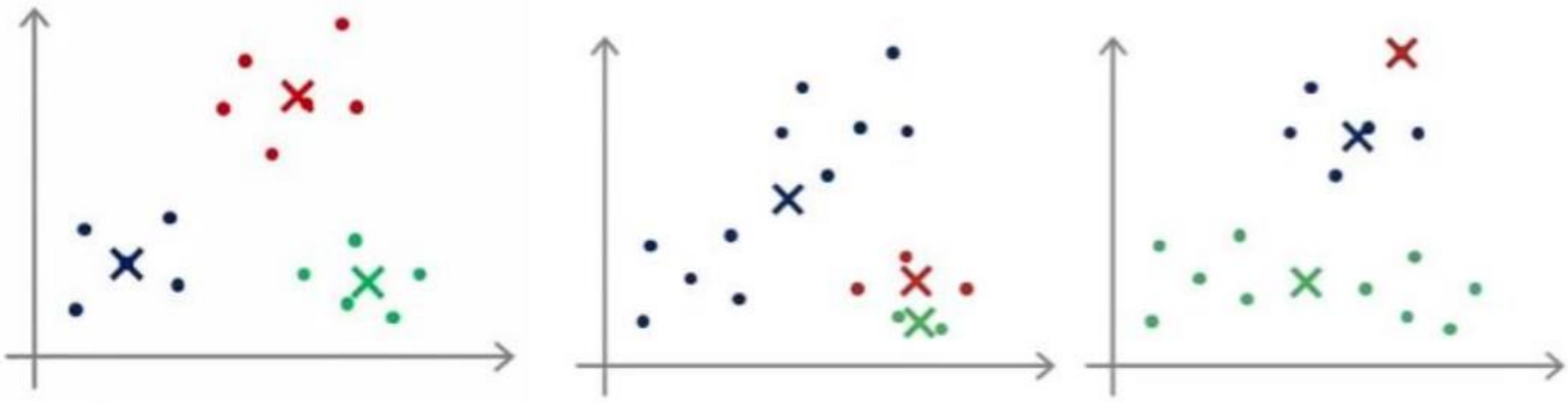
- Minimizacija funkcije koštanja J kroz K-means algoritam:
  - Faza Grupisanja po klasterima (korak 2) minimizuje J po parametrima c^1, c^2,... c^m, držeći μ_1, μ_2,... μ_k fiksnim
  - Faza Pomeranja težišta minimizuje J po parametrima μ_1, μ_2,... μ_k, držeći c^1, c^2,... c^m, fiksnim

K-means evaluacija

- Da li imamo tačno rešenje?
- Kriterijumi za procenu kvaliteta kreiranih klastera
  - Međusobna udaljenost težišta
  - Standarda devijacija pojedinačnih instanci u odnosu na težište
  - Suma kvadrata unutar klastera

Problem inicijalnog izbora težišta

- U zavisnosti od inicijalnog izbora težišta:
  - K-means algoritam može konvergirati brže ili sporije
  - Može se doći do lokalnog minimuma i time imamo loše rešenje (lokalni minimum funkcije koštanja)



K-means - višestruka nasumična inic.

- Omogućava da se izbegne ulazak u lokalni minimum
- Sastoji se u sledećim koracima:
  for i = 1 to N { // N obicno ima vrednosti izmedju 50 i 1000
  Nasumicno odabrati inicijalni skup težišta
  Izvršiti K-means algoritam
  Izračunati funkciju koštanja
  }
  Izabrati instancu algoritma koja daje najmanju vrednost funkcije koštanja.
- Ovaj pristup daje dobre rezultate ukoliko je broj klastera relativno mali (između 2 i 10), za veći broj klastera ne treba da se koristi.

K-means ++ algoritam

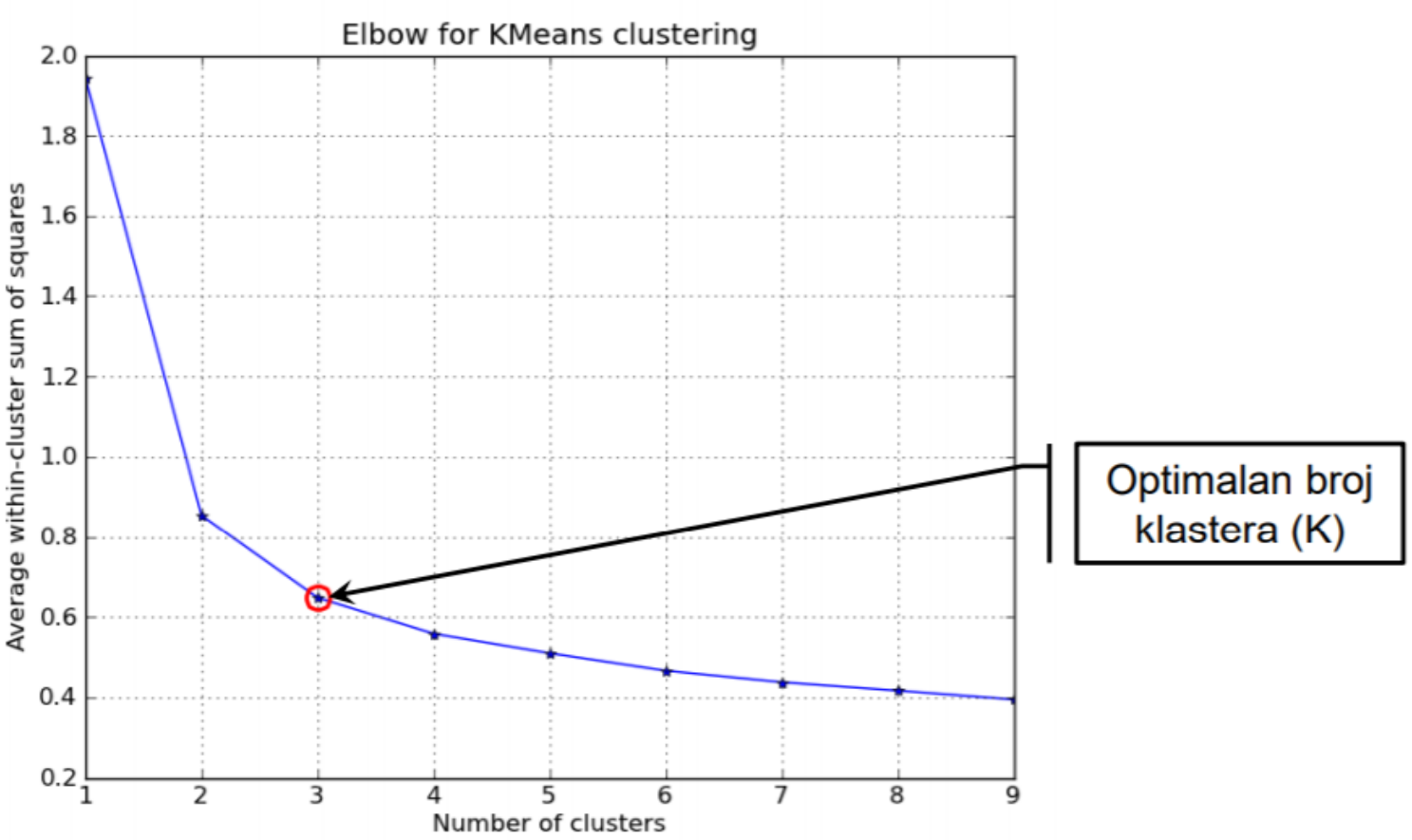
- Alternativno rešenje

1. Initialize an empty set $\mathbf{M}$ to store the k centroids being selected.
2. Randomly choose the first centroid $\boldsymbol{\mu}^{(j)}$ from the input samples and assign it to $\mathbf{M}$.
3. For each sample $\boldsymbol{x}^{(i)}$ that is not in $\mathbf{M}$, find the minimum squared distance $d\left(\boldsymbol{x}^{(i)}, \mathbf{M}\right)^2$ to any of the centroids in $\mathbf{M}$.
4. To randomly select the next centroid $\boldsymbol{\mu}^{(\mathrm{p})}$, use a weighted probability distribution equal to $\frac{d\left(\boldsymbol{\mu}^{(\mathrm{p})}, \mathbf{M}\right)^2}{\sum_i d\left(\boldsymbol{x}^{(i)}, \mathbf{M}\right)^2}$.
5. Repeat steps 2 and 3 until k centroids are chosen.
6. Proceed with the classic k-means algorithm.

K-means: Kako odrediti K?

- Kako odrediti broj klastera K?
  - Ukoliko imamo znanje o oblasti/pojavi koju podaci opisuju
    - Pretpostaviti broj K na osnovu domenskog znanja
    - Testirati model sa K-1, K i K+1 klastera i uporediti grešku
  - Ukoliko ne posedujemo znanje o oblasti/pojavi koju podaci opisuju
    - Krenuti od malog broja iteracija i u više iteracija testirati model uvek sa jednim klasterom više
    - U svakoj iterarciji uporediti grešku (nekom metodom) tekućeg i prethodnog modela i kad smanjenje greške postane zanemarljivo, prekinuti postupak

Primer kada nemamo znanje o domenu



Hard vs Soft klasterovanje

- Hard klasterovanje – grupa algoritama gde svaki uzorak (instanca) u skupu je dodeljena tačno jednom klasteru
- Soft (Fuzzy) klasterovanje – uzorak dodeljen više klastera (primer: fuzzy C-means, ili soft k-means)

$$\begin{bmatrix} \mu^{(1)} \to 0 \\ \mu^{(2)} \to 1 \\ \mu^{(3)} \to 0 \end{bmatrix} \qquad \begin{bmatrix} \mu^{(1)} \to 0.1 \\ \mu^{(2)} \to 0.85 \\ \mu^{(3)} \to 0.05 \end{bmatrix}$$

Expectation-Maximization algoritam

- EM algoritam spada u grupu probabilističke klasterizacija
- Ideja: uzorci nam ne pripadaju jednom i samo jednom klasteru, već svaki uzorak sa određenom verovatnoćom pripada svakom od klastera
- Ova vrsta klasterizacije podrazumeva:
  - Svaki klaster je opisan određenom verovatnoćom (jedna raspodela zajednička za sve atribute kojima su uzorci opisani ili više raspodela, po jedna za svaki od atributa)
  - Postoji i raspodela verovatnoća koja karakteriše pripadnost klasteru.

Probabilističko klasterovanje

- Svi uzorci su opisani jednim numeričkim atributom koji ima Normalnu raspodelu u svim klasterima (ukupno K klastera)
- Svaki klaster Ci ima specifične vrednosti parametara Normalne raspodele – srednje vrednosti i standardne devijacije
- p_i predstavlja prior probability i-tog klastera C_i, odnosno verovatnoću da nasumice odabrana instanca pripada klasteru C_i
- Dobili smo skup uzoraka (instanci) za koje znamo da dolaze iz opisanih K klastera, ali ne znamo iz kojih, niti znamo parametre modela (μ_i,σ_i,ρ_i, gde je i = 1,...K)
- Za svaku instancu treba odrediti verovatnoću pripadnosti svakom od K klastera

EM algoritam

- Opisani problem se može rešiti primenom postupka koji predstavlja generalizaciju K-means algoritma:
- 1) Inicijalno, definisati broj klastera (k) i nasumice izabrati vrednosti parametara modela (μi, σi, pi, $i=1,k$)
- 2) Za date vrednosti parametara, za svaku instancu iz dataset-a, izračunati verovatnoću pripadanja svakom od klastera
- 3) Na osnovu izračunatih verovatnoća pripadnosti klasterima, odrediti nove vrednosti parametara modela
- Iterativno ponavljati korake 2) i 3) dok vrednosti parametara ne počnu da konvergiraju

Koraci kod EM algoritma (1)

- Sastoji se iz:
- Korak **Inicijalizacije (I)** – središta klastera na osnovu inicijalno postavljenih vrednosti parametara modela
- **E (expectation) korak** – u ovom koraku podrazumevamo da znamo vrednosti parametara modela i na osnovu njih, za svaku instancu, računamo verovatnoću pripadanja svakom od klastera
- **M (maximization) korak** – na osnovu datih instanci, računamo (ponovo) vrednosti parametara modela; maksimizacija se odnosi na usklađivanje (parametara) modela sa datim podacima

Koraci kod EM algoritma (2)

E korak

- Za svaku instancu iz skupa podataka x_j (j=1,n) računamo verovatnoću pripadanja i-tom klasteru C_i (i=1,k):

  $e_{i\,j} = p_i * P(x_j|C_i)$

- P(xj|Ci) se računa primenom funkcije Normalne raspodele
  f(x; μ,σ)

M korak

- Određuju se nove vrednosti parametara modela:
  - prior probability $$p_i = \sum_j \frac{e_{ij}}{n}$$
  - srednja vrednost $$\mu_i = \frac{\sum_j e_{ij} * x_j}{\sum_j e_{ij}}$$
  - varijansa $$\sigma_i^2 = \frac{\sum_j e_{ij} * (x_j - \mu_i)^2}{\sum_j e_{ij}}$$

Koraci kod EM algoritma (3)

Konvergencija

- Koraci algoritma se ponavljaju sve dok ima značajne promene (tj. porasta) loga ukupne verovatnoće modela (overall log-likelihood):

$$\log P(x) = \log \sum_i (p_i * P(x|C_i))$$

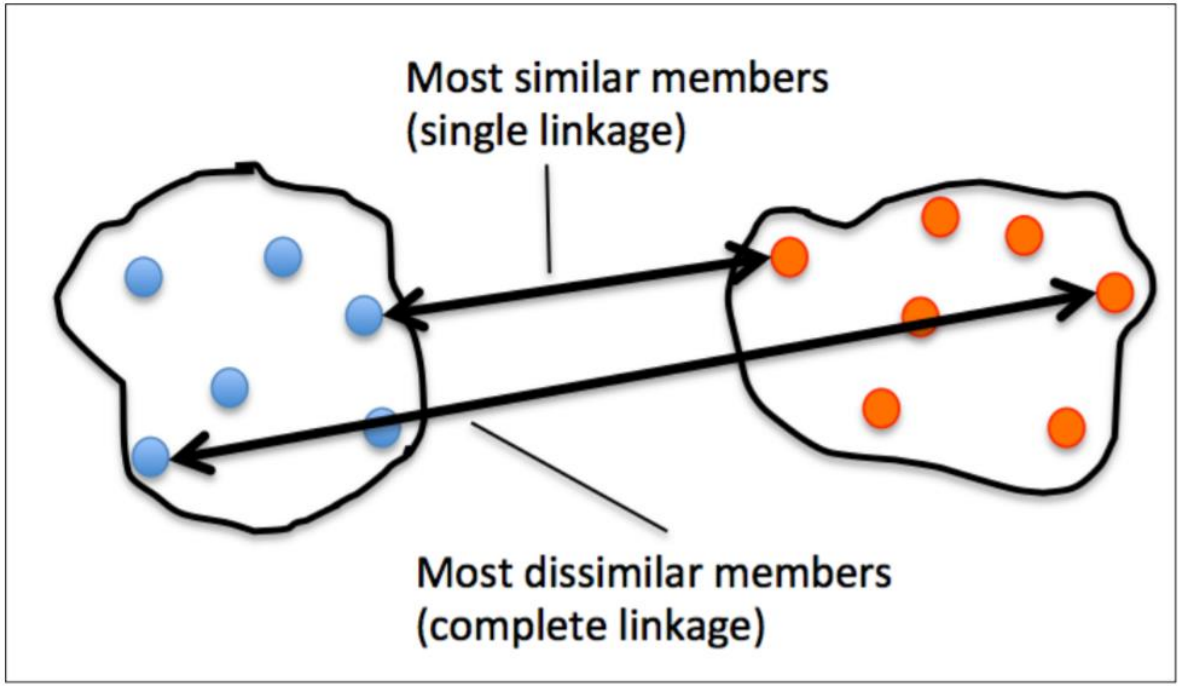
- Tipično, ova vrednost intenzivno raste tokom prvih nekoliko iteracija algoritma, a zatim vrlo brzo dolazi u stanje gde skoro da nema promene

Organizacija klastera kao hijerarhijsko stablo

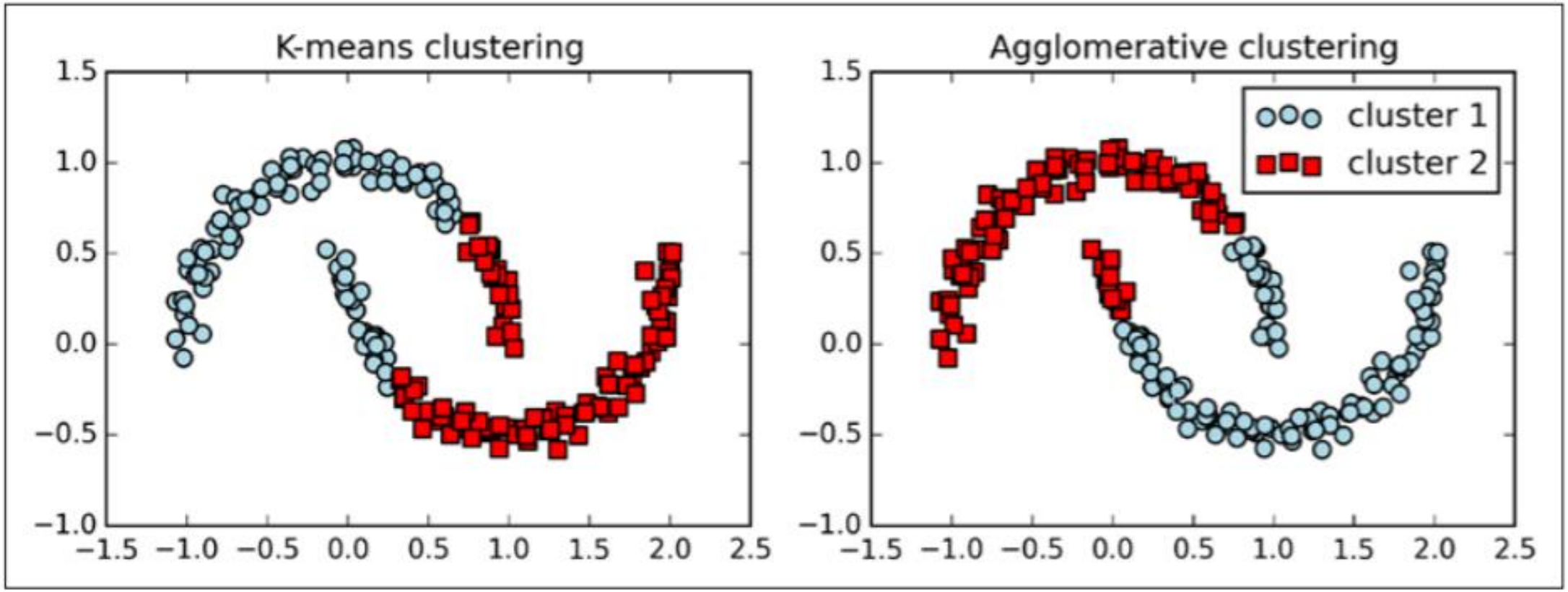
- Alternativni pristup klasterovanju zasnovanom na prototipu je hijerarhijsko klasterovanje
- Prednosti ovog pristupa:
  - crtanje Dendrograma (vizuelizacija hijerarhijskog klasterovanja)
  - ne treba da preciziramo broj klastera unapred
- Dva osnovna tipa:
  - Aglomerativno klasterovanje
  - Podeljeno hijerarhijsko klasterovanje

Aglomerativno klasterovanje

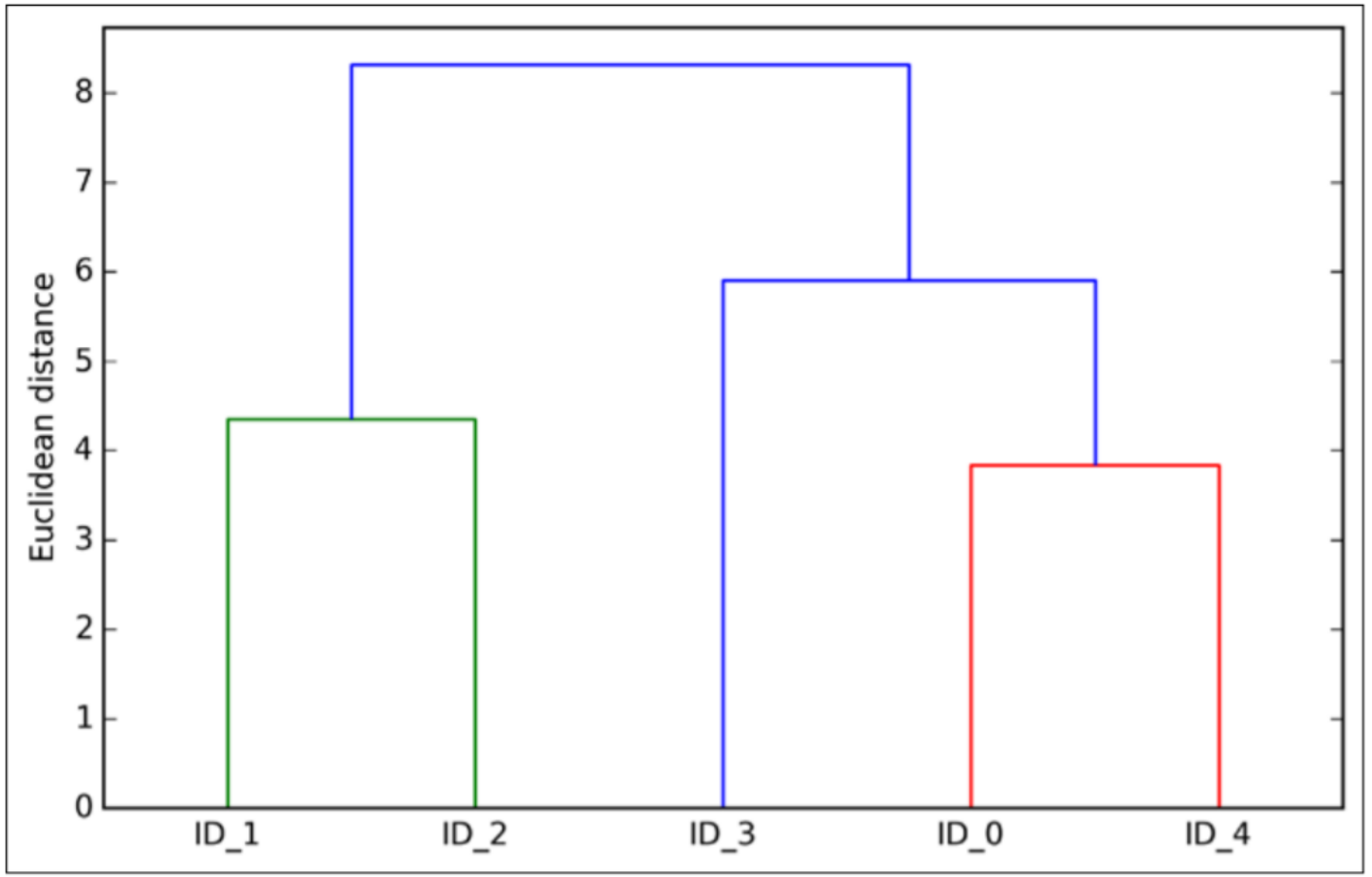
- Dva standardna algoritma:
  - algoritam sa jednostrukom vezom (eng. single linkage)
  - algoritam sa potpunom vezom (eng. complete linkage)
- Algoritam sa jednostrukom vezom: izračunavamo udaljenosti između najsličnijih članova za svaki par klastera i spajamo dva klastera za koje je udaljenost između najsličnijih članova najmanja
- Algoritam sa potpunom vezom: ne uspoređujemo najsličnije članove u svakom paru klastera, već uspoređujemo najviše različitih članova kako bismo obavili spajanje



Izlaz: dijagram sa klasterima



Izlaz: dendrogram



Izlaz: heat map