PONAVLJANJE IZ STROJNOG ZADNJI LAB

PROCJENA PARAMETARA

cilj: želimo na temelju uzorka koji imamo pogodit koji su parametri razdiobe prema kojim su ti podaci stvoreni. Npr. znamo da je bacanje novčića prema bernoullijevoj distribuciji, ali ne znamo parametar mi (vjerojatnost da se dogodi događaj *glava*). Pogledamo uzorak od 10 bacanja i na temelju broja dobivenih glava pokušamo pogodit koliki je bio mi koji je generirao te podatke.

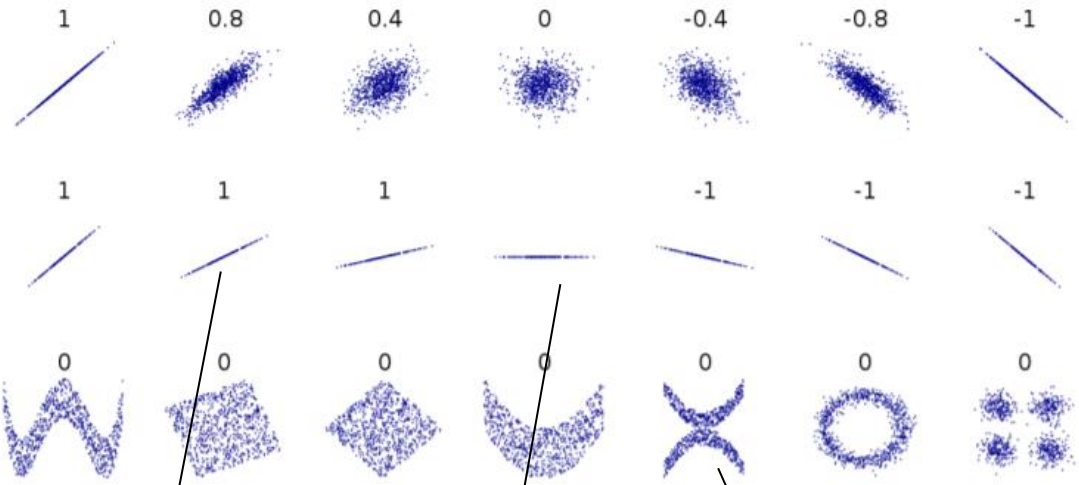
varijanca – koliko vrijendnosti varijable variraju oko očekivane vrijednosti

kovarijanca – u kojoj mjeri 2 slučajne varijable zajednički variraju oko svojih očekivanih vrijednosti

pearsonov koeficijent korelacije – koliko su 2 varijable međusobno **linearno** zavisne.

Za slučajne varijable X i Y za koje vrijedi $\mathrm{Var}(X) \neq 0$ i $\mathrm{Var}(Y) \neq 0$ definiran je **Pearsonov koeficijent korelacije**:

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}} = \frac{\sigma_{X,Y}}{\sigma_X \sigma_Y}$$

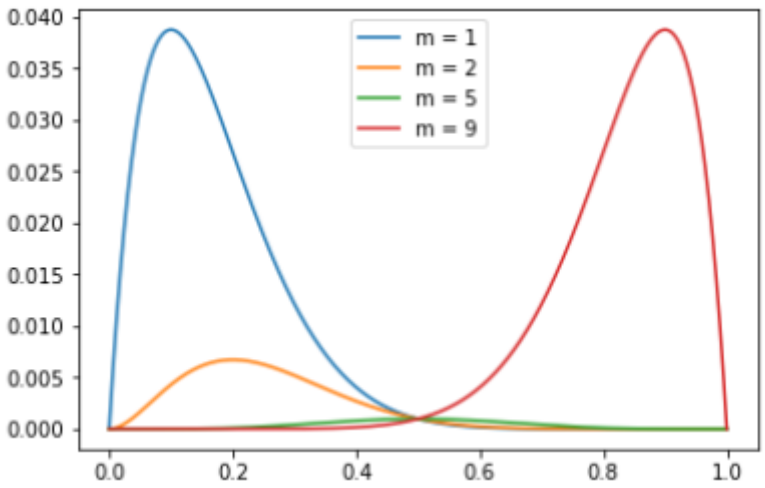


Savršeno linearno zavisni, ne mora bit y = x, nego y = kx

Nije definirano jer je y = konst.

Zavisni, ali ne linearno zavisni

Funkcija izglednosti – vjerojatnost da s nekim parametrima izučemo neki uzorak



Ako gledamo plavu krivulju da u 10 bacanja izvučemo 1 glavu mi može bit svakakav, ali najveća je vjerojatnost da je mi upravo 0.1, kada je mi 0.1 vjerojatnost da izvučemo 1/10 glava je najveća. Kad je mi = 1 vjerojatnost da izvučemo 1 glavu je 0, jer takav mi daje uvijek glave.

Ovo nije funkcija gustoće i integral ispod ove krivulje nije 1.

MLE – parametri koji maksimiziraju log izglednost. Logaritam iz funkcije izglednosti je tu samo zbog lakše matematike, u stvarnosti gledaš za koji parametar (kod nas za koji mi) dobiješ najveću izglednost.

Na plavoj krivulji bi ML procjena bila 0.1, na crvenoj (u 10 bacanja 9 glava) je to mi = 0.9

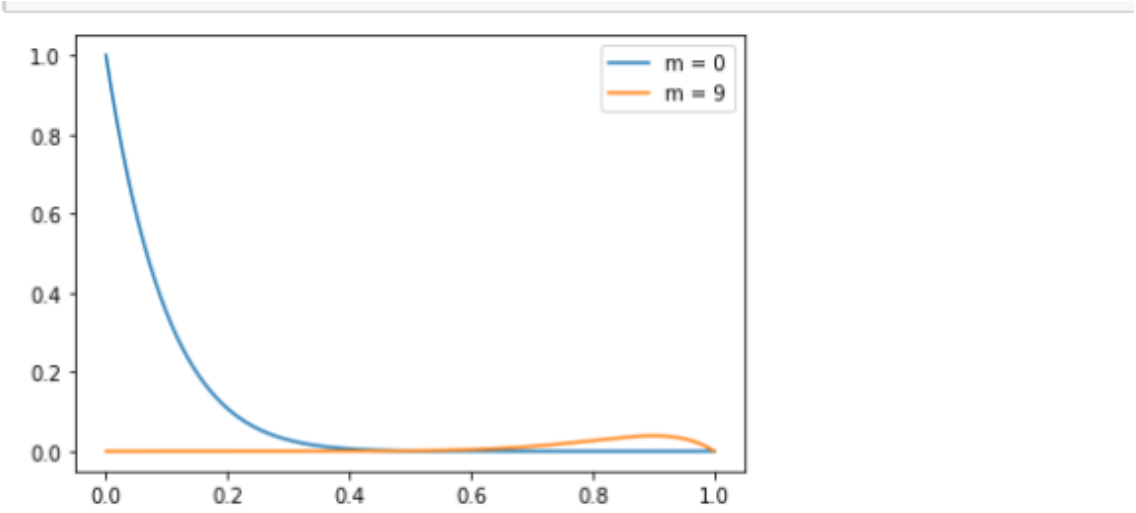
Zbilja vrijedi da je ML procjena za Bernoullijevu varijablu uvijek m/N, tj. E(MLE) = E(X) = mi

Zbog ovog je to nepristran procjenitelj. Ja ne razumijem baš što to znači.

Za multinolijevu varijablu je ML procjena Nk/N što je isto samo s više varijabli

Za gaussa je malo kompliciranija formula

Bitno za MLE je što **nema ugrađeno apriorno znanje**. To znači da se mi pravimo da ne znamo ništa o novčićima ni kako rade, samo uzimamo u obzir naš uzorak i to je to. Zbog toga je MLE podložan na prenaučenost.



Q: Koja je ML-procjena za μ i što je problem s takvom procjenom u ovome slučaju?

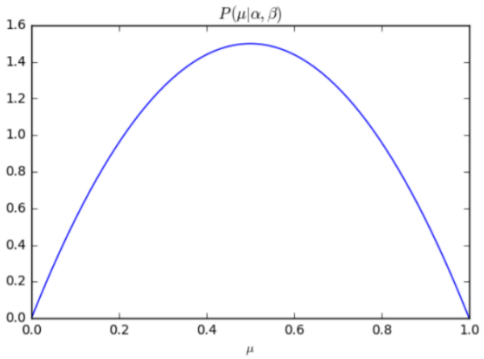
Kod m=0 (Od 10 bačenih novčića dobili 0 glava) je ML procjena 0 (tu je max plave krivulje). To bi značilo da je najizglednije da imamo novčić koji ima šansu 0 za glave i 1 za pisma. To je kao malo suludo jer na temelju svega što znamo o novčićima nema šanse da je vjeojatnost baš 0. Ali MLE **nema ugrađeno apriorno znanje** pa je podložan takvim ludim situacijama. Ovdje se dogodila prenaučenost. Imamo jako malo podataka i jednostavno se tu i tamo dogodi niz od 10 pisma. Kada bismo u MLE ugradili nekako naše znanje o novčiću (i činjenicu da novčići imaju tendenciju bit 50%-50%) onda bi nam procjena parametara bila malo točnija.

MLE je ok samo ako imamo jako jako puno podataka, ako u milijun puta padne pismo onda stvarno možemo računat da novčiću glava ne radi.

Rješenje na to što MLE **nema ugrađeno apriorno znanje** je MAP

MAP – procjena maksimalne aposteriorne vrijednosti

Ovdje pokušavamo ugraditi naše znanje o novčiću. To činimo tako da definiramo gustoću vjerojatnosti za mi. Dakle mi kažemo hey bok mi je najčešće kod novčića 0.5 i rjeđe 0.3, a nikad 0.0:

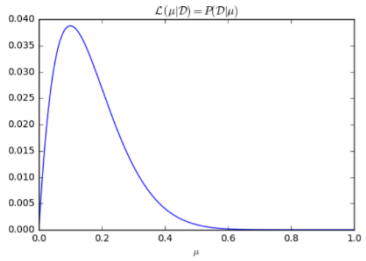


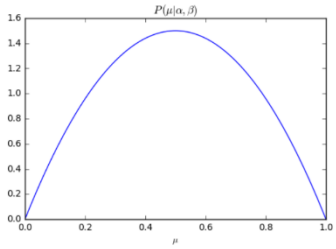
Molim te uzmi mi to u obzir sljedeći put kad budeš pokušavao pogodit koji su parametri najizgledniji za moj uzorak.

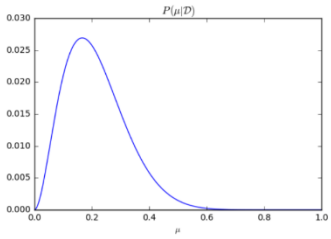
Dakle osim što sad znamo koji uzorak je došao, znamo i koji mi-jevi su izgledniji, pa kad to oboje uzmemo u obzir izaberemo mi za koji je najizglednije da je proizveo ovakav uzorak.

Da te dve stvari iskombiniramo koristimo Bayesovo pravilo.

Na kraju se ispostavi sljedeće: *Kombinacija se ostvaruje jednostavnim množenjem tih dviju vjerojatnosti. Ako neke vrijednosti za parametar ϑ imaju malu apriornu vjerojatnost, onda će i njihova aposterirorna vjerojatnost biti smanjena!*







Zadnja slika (MAP) se dobiva kombinacijom funkcije izglednosti za taj uzorak i funkcije gustoće vjerojatnosti za parametar (u našem slučaju mi). Vidimo da je ova malo desnije nego prva. To nam je super, pomaknuli smo je u smjeru u kojem mi vjerujemo da novčići rade!

Sljedeće što želimo je birati tu apriornu distribuciju pametno – na način da umnožak nje i distribucije od primjera bude opet neka poznata distribucija. Najbolje bi bilo da ta nova aposteriorna distribucija bude identična kao i naša apriorna → to nam omogućuje da radimo pojedinačno (online) učenje.

Naime, ako je aposteriorna distribucija istoga tipa kao apriorna distribucija, onda, kada izračunamo aposteriornu distribuciju, možemo je u idućoj iteraciji, kada dodu novi podatci, koristiti kao novu apriornu distribuciju

Konjugatne distribucije – aposteriorna i apriorna vjerojatnost odabrane tako da su to iste vrste distribucija

Aposteriorna $p(\boldsymbol{\theta}\|\mathcal{D})$	Izglednost $p(\mathcal{D}\|\boldsymbol{\theta})$	Apriorna $p(\boldsymbol{\theta})$
Beta	Bernoulli	Beta
Dirichlet	Multinuli	Dirichlet
Normal	Normal	Normal
Multivariate normal	Multivariate normal	Multivariate normal

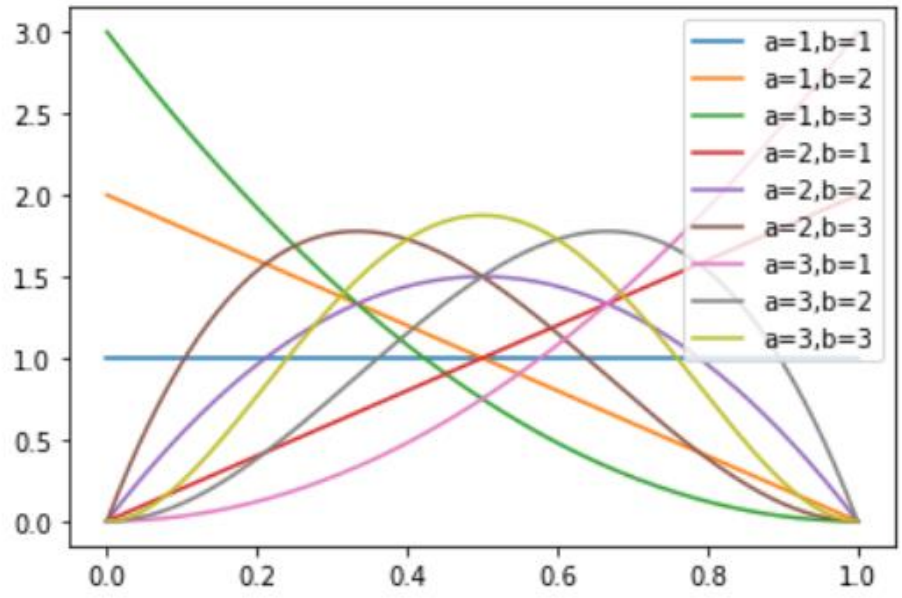
Beta-Bernoullijev model

Funkcija gustoće beta-distribucije: $$p(\mu|\alpha,\beta) = \frac{1}{B(\alpha,\beta)}\mu^{\alpha-1}(1-\mu)^{\beta-1}$$

Gdje je B tzv. beta-funkcija – normalizirajuća konstanta koja osigurava da je integral funkcije gustoće jednak 1

- Što su alfa i beta veći, više gustoće vjerojatnosti guramo oko 0.5
- Alfa < beta → maksimizator < 0.5
- Alfa > beta → maksimizator > 0.5
- Alfa = beta = 1 → uniformna distribucija, odnosno

- nemamo nikakvoga apriornog znanja o vrijednosti parametara
- **neinformativna apriorna distribucija**



Pazi!

mi – parametar: njega želimo *procijeniti*
alfa, beta – hiperparametri: njih definiramo *unaprijed*

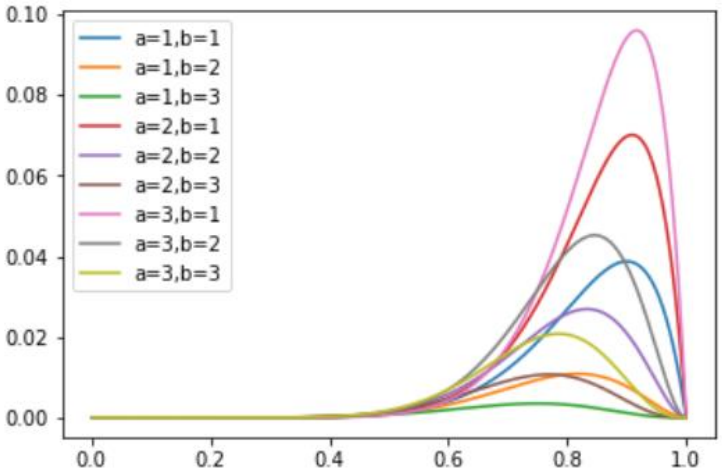
Kad pomnožimo apriornu distribuciju (koja je beta-distribucija) s Bernoullijevom varijablom dobit ćemo aposteriornu beta distribuciju
Ako iz toga uzmemo argmax dobit ćemo MAP procjenitelj $= \frac{m + \alpha - 1}{\alpha + N + \beta - 2}$
Vidi se da MAP postane MLE (m/N) ako uzmemo alfa = beta = 1

Najbolja stvar kod MAP procjenitelja jest što ako ima jako puno podataka (veliki N) više se vjeruje podacima, a ako ima jako malo podataka onda se više vjeruje apriornom znanju

Q: Koje parametere biste odabrali za modeliranje apriornog znanja o parametru μ za novčić za koji mislite da je "donekle pravedan, ali malo češće pada na glavu"? Koje biste parametre odabrali za novčić za koji držite da je posve pravedan? Zašto uopće koristimo beta-distribuciju, a ne neku drugu?

- Malo češće pada na glavu → želimo da bude malo nadesno → želimo da alfa > beta → npr. a=3 b=2
- Posve pravedan → želimo da alfa = beta i u strojnom učenju se najčešće uzima vrijednost 2
- Beta distribuciju koristimo zato što je ona konjugatna Bernoullijevoj izglednosti. Ako to vrijedi, onda će apriorna i aposteriorna distribucija biti iste – u ovom slučaju obje *beta-distribucije*. Ako su iste onda možemo raditi online učenje. Sa svakim novim primjerom mi aposteriorna distribucija postane apriorna i idemo dalje.

Definirajte funkciju za izračun zajedničke vjerojatnosti $P(\mu, D) = P(D|\mu) \cdot P(\mu|\alpha, \beta)$ te prikažite tu funkciju za $N = 10$ i $m = 9$ i nekolicinu kombinacija parametara α i β.



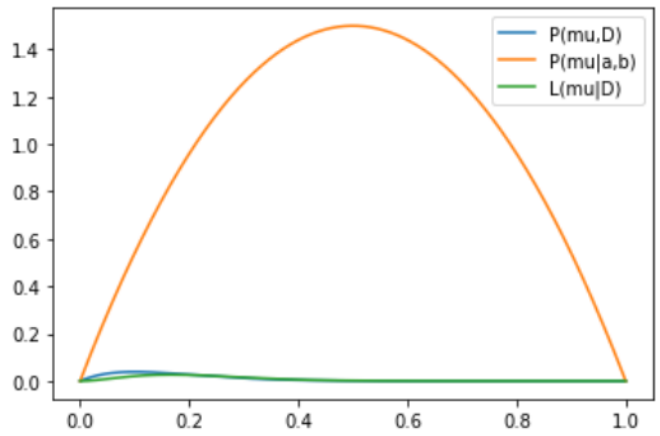
Q: Koje vrijednosti odgovaraju MAP-procjeni za μ? Usporedite ih sa ML-procjenama.

MAP procjene su maksimumi ovih funkcija.

ML procjena je max plave krivulje jer se MAP za alfa = beta = 1 svodi na MLE

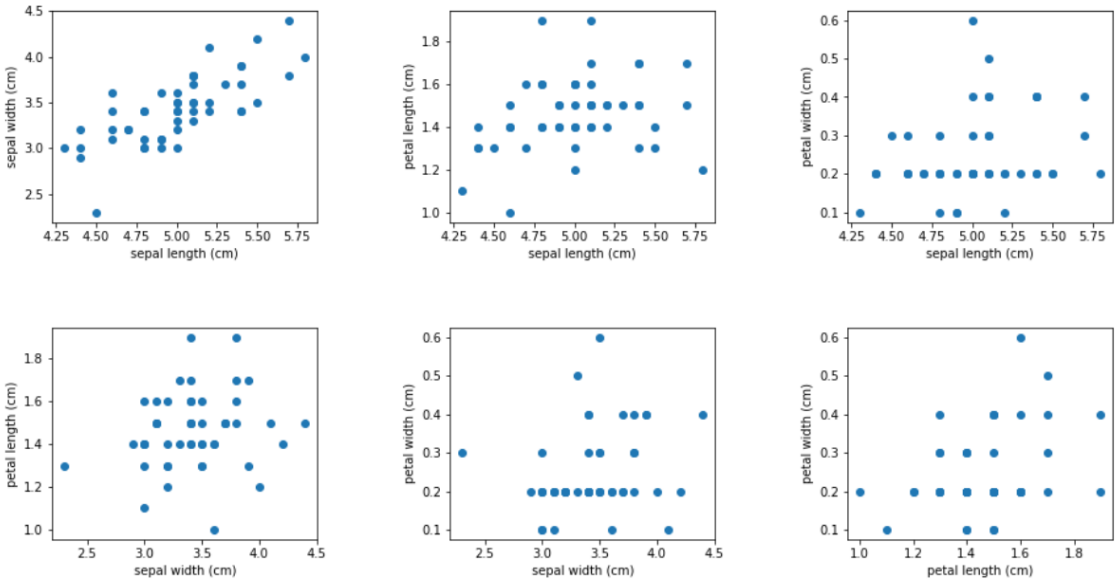
Što je alfa *više veći* od beta to cijela krivulja ide više desno, i obrnuto.

Podataka nema puno pa alfa i beta dosta utječu na izgled krivulja



Narančasto je beta → ona je takva jer alfa=beta=2, puno gustoće vjerojatnosti guramo oko 0.5

Za ostalo nez, nije baš neki graf



Iz ove slike se vidi da prvi graf actually ima neku korelaciju između značajki [0,1] pa ćemo to kasnije i dokazat pri izračunu pearsonovog koeficijenta korelacije.

```
mi mle značajke 0.: 5.006
sigma mle značajke 0.: 0.12176400000000002
log izglednost značajke 0.: -18.305163312803863
----------------------------------------------------------------------
mi mle značajke 1.: 3.428
sigma mle značajke 1.: 0.14081600000000002
log izglednost značajke 1.: -21.939396526466616
----------------------------------------------------------------------
mi mle značajke 2.: 1.4620000000000002
sigma mle značajke 2.: 0.029555999999999995
log izglednost značajke 2.: 17.08978609115976
----------------------------------------------------------------------
mi mle značajke 3.: 0.24599999999999997
sigma mle značajke 3.: 0.010883999999999998
log izglednost značajke 3.: 42.0646097912948
----------------------------------------------------------------------
```

Mi mle je u biti prosjek

Sigma mle je prosječno kvadratno odstupanje od toga.

Log izglednost nez zakaj smo izračunali, al oki doki

```
Koef. korelacije između značajki 0 i 1 je 0.7425466856651596
Koef. korelacije između značajki 0 i 2 je 0.26717575886875716
Koef. korelacije između značajki 0 i 3 je 0.27809835293596963
Koef. korelacije između značajki 1 i 2 je 0.17769996678227068
Koef. korelacije između značajki 1 i 3 je 0.23275201136287935
Koef. korelacije između značajki 2 i 3 je 0.33163004080411873
```

Evo vidite da je 0 i 1 najveća linearna korelacija!

Kovarijacijska matrica – unutra su kovarijance svih parova značajki.
Simetrična je jer Cov(x,y) = Cov(y,x)
Na dijagonali su varijance jer Cov(x, x) = Var(x)

Još fun factova o njoj:

(1) Ako su varijable $X_1, \ldots, X_n$ međusobno linearno nezavisne, onda $\mathrm{Cov}(X_i, X_j) = 0$ za $i \neq j$ i kovarijacijska je matrica **dijagonalna matrica**, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_i^2)$;

(2) Ako nezavisne varijable $X_1, \ldots, X_n$ imaju jednaku varijancu, onda $\sigma_i^2 = \sigma^2$, pa kovarijacijska matrica degenerira u $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, gdje je $\mathbf{I}$ jedinična matrica. Takav slučaj nazivamo **izotropnom kovarijancom**.

Još je i pozitivno semidefinitna → ne mora nužno imati inverz: akko je neki element na dijagonali jednak nuli ili ako postoji linearna zavisnot redaka/stupaca u matrici dizajna.

```
mi mle   : 0.15378469387755095
sigma mle: 0.06181794737817571
N = 50
     srednja  : 0.0007689234693877 49
     kvadratna: 1.4518483236151334e-06
N = 25
     srednja  : 0.0018493333333333304
     kvadratna: 7.680071499999981e-06
N = 12
     srednja  : 0.0023989898989899023
     kvadratna: 1.293849160799922e-05
```

//todo skripta

GRUPIRANJE – PRVI DIO

k-sredina

deterministički? Da, ali…

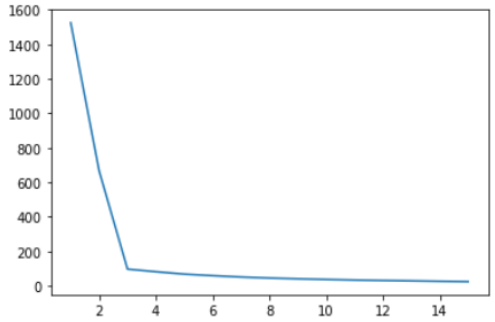
ovisi o načinu na koji biramo početne centre, ako ih biramo nekako deterministički algoritam će biti deterministički (za iste ulaze davat iste izlaze). Također, mora se pripaziti da se pametno razriješi izjednačenje udaljenosti dva primjera od centroida → inače se može upast u beskonačnu petlju.

Konvergira? Da

Zato što je broj konfiguracija konačan: broj svih mogućih grupacija ikad (što je K^N) kao i odabir početnih centara. Optimizacijski je postupak definiran tako da se kriterijska funkcija J nužno smanjuje kroz iteracije. Zbog te dve stvari znamo da algoritam konvergira

Optimalno rješenje? Ne

Pronalazi se lokalno optimalno rješenje, a jel to lokalno ujedno i globalno ovisi o početnom odabiru centara.



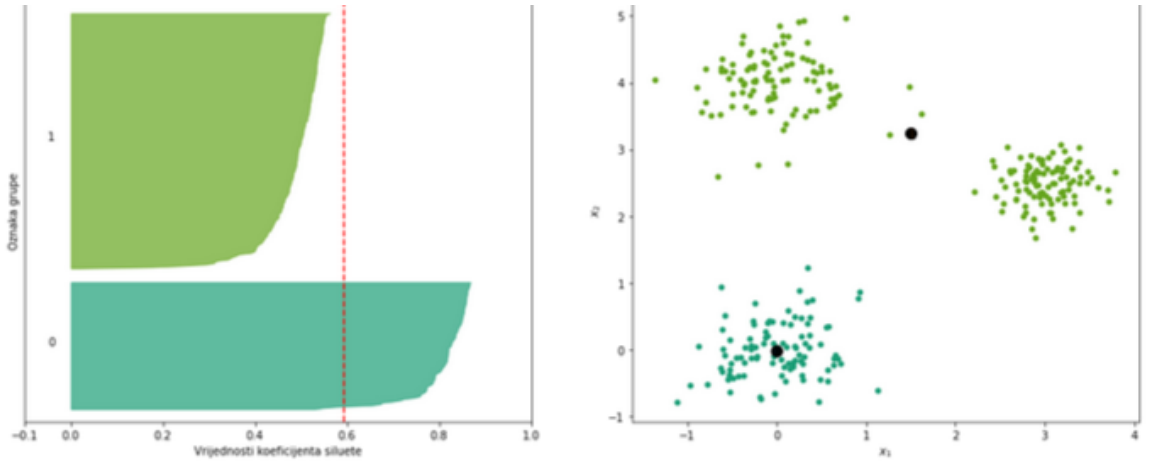
Q: Koju biste vrijednost hiperparametra K izabrali na temelju ovog grafa? Zašto? Je li taj odabir optimalan? Kako to znate?
Q: Je li ova metoda robusna?
Q: Možemo li izabrati onaj K koji minimizira pogrešku J? Objasnite.

Na slici je ovisnost kriterijske funkcije J (pogreška – udaljenost primjera od centroida njegove grupe) o broju grupa K.

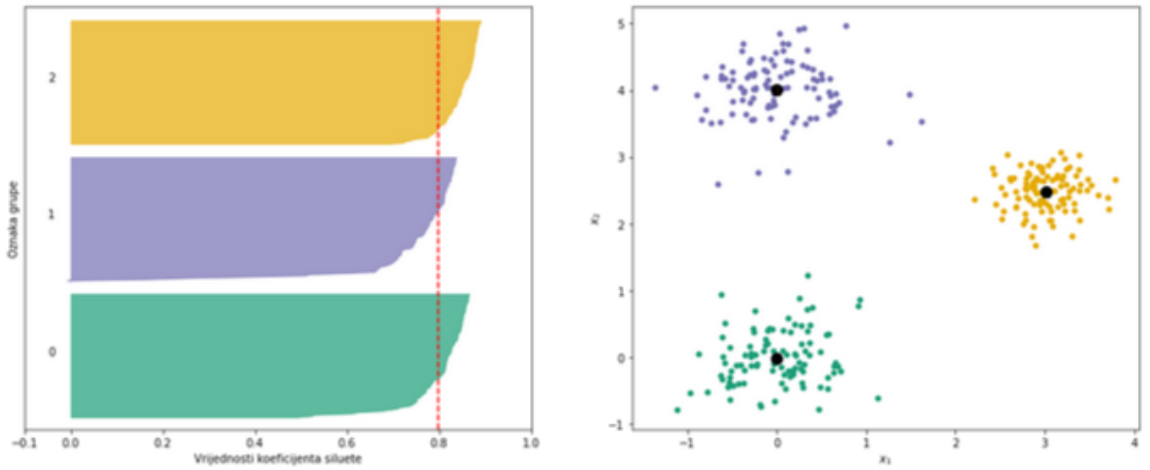
1. Izabrala bi K = 3. To je zato što nakon K = 3 J počinje stagnirati dakle povećanjem broja grupa ne mijenjamo puno grešku ali zato povećavamo složenost, znači u nekom smislu prenaučavamo. Želimo onaj K tik prije stangacije, zadnju grupaciju koja stvarno napravi razliku.
2. Taj odabir nije optimalan zato što K-means nije deterministički kad koristimo nasumično odabrane centre. Bili bismo sigurniji kada bismo jako puno puta izvrtili algoritam i nacrtali prosječan J.
3. Je li ova metoda robusna?????

```
Q: Je li ova metoda robusna?
A: Ne nužno. Ako imamo dosta snažnih outliera, oni mogu znatno
povećati iznos kriterijske funkcije te će zahtijevati dodatne grupe
(veći parametar K), iako u stvarnosti ne postoji potreba za tim.
```

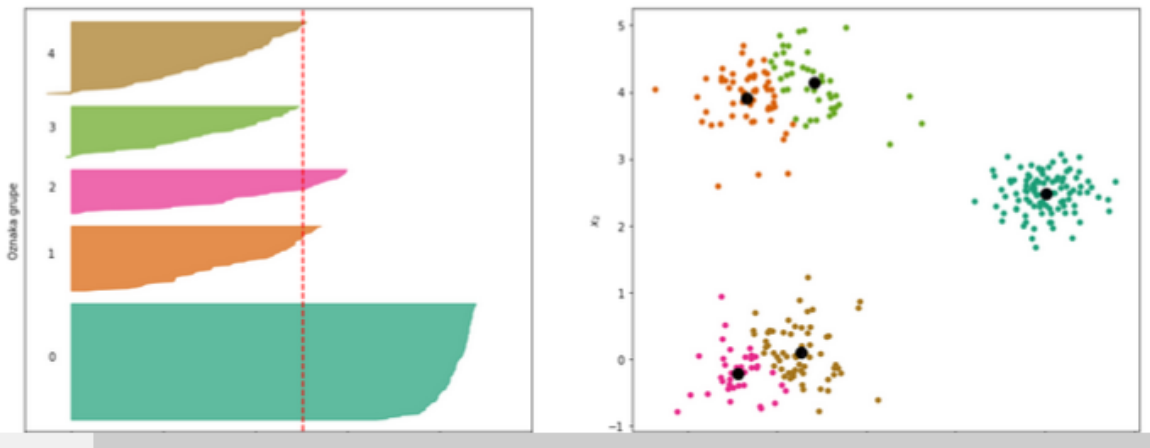
4. Ne smijemo izabrati K koji minimizira pogrešku J. Pogreška J pada kako ima više grupa, i J je minimalna (0) ako je svaki primjer sam svoj centar, jer su onda svi primjeri najbliže centru, a to ne želimo.




```
For n_clusters = 3 The average silhouette_score is : 0.7975462212061406
```




```
For n_clusters = 5 The average silhouette_score is : 0.5052371766008248
```



Silueta: Za svaki primjer se gleda koliko je prosječno udaljen od svih primjera iz svoje grupe minus od svih primjera iz najbliže susjedne grupe. Rezultat +1 znači da je puno puno bliže svojima nego tuđima, =0 da je

na pola puta, a -1 (kod k-means se ne može dogoditi) da je puno puno bliže tuđoj grupi. Tak se nacrtaju svi ti rezultati i gledaju se siluete.

Problematične grupacije su one gdje je silueta za cijelu grupu manja od prosječne vrijednosti za sve siluete. Također, ne valjaju one siluete koje imaju veliku varijancu (npr roza). Te grupe nisu prirodne.

Q: Kako biste se gledajući ove slike odlučili za K?
Q: Koji su problemi ovog pristupa?

1. K = 3 zato što su sve siluete iznad prosjeka i dodatna olakotna okolnost je što su sve grupe podjednake veličine, to je uvijek lijepo.
2. Nemam bolju ideju nego: silueta neće dobro raditi na linearno neodvojivim grupama. Npr. ako imamo koncentrične kružnice ko u zadatku dolje, čak i da savršeno grupiramo rezultat će bit bliže 0 jer će svaki primjer bit podjednako udaljen od svoje i susjedne grupe. Ne moraju ni bit linearno neodvojivi, dovoljno je da jedna grupa skoro okružuje drugu kao mjesec! Hihi mjesec