# Pronalaženje optimalnog broja kategorija za K-means algoritam strojnog učenja

**Kokot, Enrico**

Sveučilište Jurja Dobrile u Puli
Tehnički fakultet u Puli

**ENRICO KOKOT**

# PRONALAŽENJE OPTIMALNOG BROJA KATEGORIJA ZA K-MEANS ALGORITAM STROJNOG UČENJA

Završni rad

Pula, _____, 2022. godine

Sveučilište Jurja Dobrile u Puli
Tehnički fakultet u Puli

**ENRICO KOKOT**

# PRONALAŽENJE OPTIMALNOG BROJA KATEGORIJA ZA K-MEANS ALGORITAM STROJNOG UČENJA

Završni rad

**JMBAG: 0009068626, redoviti student**
**Studijski smjer: Sveučilišni preddiplomski studij računarstvo**

**Predmet: Sustavi temeljeni na znanju**
**Znanstveno područje: Tehničke znanosti**
**Znanstveno polje: Računarstvo**
**Znanstvena grana: Umjetna inteligencija**
**Mentor: doc. dr. sc. Nicoletta Saulig**

Pula, _____, 2022. godine

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika _____ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, 2022. godine

IZJAVA

o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom

_____

_____

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

## Sadržaj

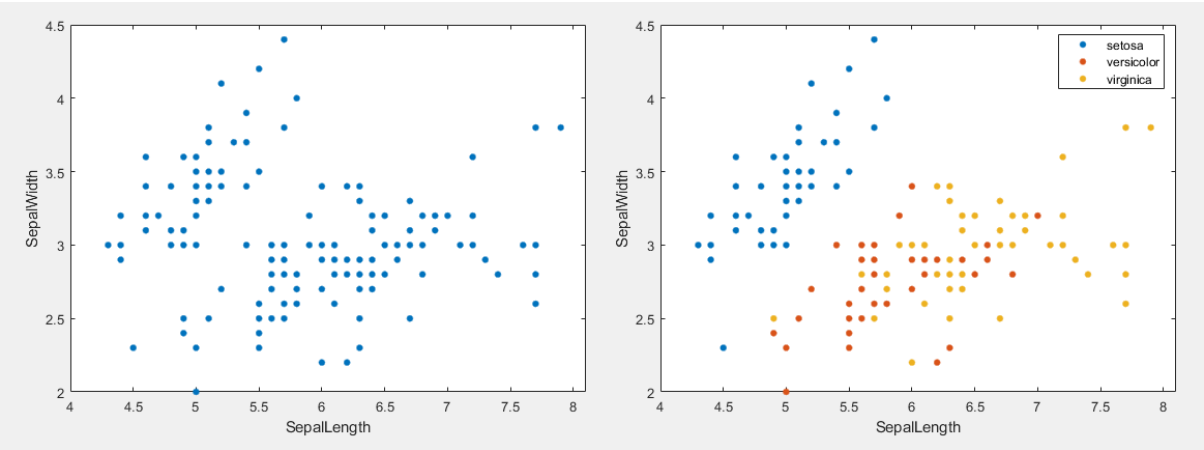
# 1. Uvod

Živimo u svijetu gdje se podaci konstantno stvaraju i čovječanstvo iz različitih razloga pokušava dobiti neke korisne informacije iz te silne količine podataka. Jedna od metoda koje se koriste sa tim ciljem pod nazivom strojnog učenja je klastering koji nam služi kako bismo unutar skupova podataka pronašli smislene cjeline. Klastering se može primjenjivati korištenjem različitih metoda od kojih ćemo se u ovom radu usredotočiti na k-means algoritam strojnog učenja. Ovaj algoritam sam po sebi očekuje, kao jedan od parametara za uspješno obavljanje svog posla, broj smislenih cjelina na koje želimo podijeliti dani skup podataka. Ovo ograničenje očekuje od korisnika da eksperimentira i time dopušta da se uvede greška temeljena na subjektivnoj procjeni. Zato se radi na tome da se uvedu algoritmi koji će moći samostalno odrediti broj kategorija na koje će se podaci podijeliti i time jednog dana izbjeći grešku ljudskog faktora te uspješno automatizirati cijeli proces. Neki od tih algoritama koji se koriste u platformi za programiranje i numeričko računanje Matlab biti će primarna okupacija u ovog teksta.

Ovaj će rad započeti sa opisom samog klasteringa i biti će predstavljene neke alternativne metode k-means algoritmu. Nakon toga će detaljnije biti opisan sami k-means algoritam. Ovo poglavlje će slijediti okvirno predstavljanje svih četiriju algoritama za pronalaženje optimalnog broja kategorija koji su dostupni u Matlab softverskom paketu. Nakon toga svaki će od algoritama biti posebno predstavljen, uz kod za njegovo ostvarivanje i rezultate primjene algoritma na dane skupove podataka. Za kraj bit će uspoređena uspješnost algoritama.

## 2. Klastering

Klastering za cilj ima iz danog skupa podataka dobiti klastere od kojih je svaki smislena cjelina. Preciznije, želja je da svaki skup bude što ˝sličniji samome sebi˝ uz uvjet da postoji neki maksimalni broj skupova na koje će skup biti podijeljen kako bi se izbjegla pogrešku da svaka točka predstavlja svoj zasebni skup [Guttag, 2016].




*Slika 1 Primjer klasteringa na skupu podataka latica perunika*

Klastering spada u skupinu algoritama strojnog učenja koji se provode bez nadzora i nastoji riješiti problem optimizacije. Formule koje se koriste su sljedeće [Guttag, 2017]:

$$variability(c) = \sum_{e \in c} distance(mean(c), e)^2$$

$$dissimilarity(C) = \sum_{c \in C} variability(c)$$

Za potrebe klasteringa se često koristi jedna od dvije metode: hijerarhijsko klasteriranje i k-means. Obe metode koriste pohlepne algoritme koji po svojoj prirodi mogu donijeti pogrešan zaključka iz razloga što se može dogoditi da se algoritam odluči za lokalno optimalno rješenje koje nije nužno globalno optimalno. [Russel i Norvig, 2009]

Neki faktori po kojima se ovi algoritmi razlikuju su sljedeći [Guttag, 2017]:

- Hijerarhijsko klasteriranje je fleksibilno, ali potencijalno sporo dok k-means brže dolazi do rješenja.
- Hijerarhijsko klasteriranje mjeri udaljenost između svake točke u skupu podataka dok k-means dobiva broj klastera koji se koristi za dodjeljivanje broja početnih vrijednosti iz skupa podataka.
- Hijerarhijsko klasteriranje je deterministično dok je k-means nedeterminističko. To znači da, dani neki input, output će uvijek biti isti ili, u obrnutom slučaju, ne

mora biti.

Što se efikasnosti algoritama tiče, vremenska složenost jedne iteracije hijerarhijskog klasteriranja, osim nekih posebnih slučajeva, gdje je *N* broj točaka je [Guttag, 2017]:

$$O(N^3)$$

S druge strane, vremenska složenost jedne iteracije k-means algoritma gdje je *k* broj klastera, *N* broj „točaka“ i *d* vrijeme potrebno za računanje udaljenosti između dvaju točaka je [Guttag, 2017]:

$$O(k * N * d)$$

## 3. K-means

K-means algoritam za klasteriranje je iterativni algoritam koji razdvaja podatka koji funkcionira na temelju dodjeljivanja n promatranja točno jednome od k klasa koje su definirane njihovim težištem, gdje je k određen prije nego što se algoritam pokrene [Guttag, 2016].

Koraci koje algoritam poduzima su sljedeći:

1. Odabiremo k početnih težišta klasa.
2. Računa se udaljenost svake točke promatranja od svake točke težišta.
3. Pokreće se jedna od sljedećih operacija:
    a. Skupno ažuriranje – svaka točka promatranja se dodjeljuje klasi čija je točka težišta najbliža
    b. Mrežno ažuriranje – za svaku točka promatranja se provjerava da li bi promjena njene pripadnosti od jedne u drugu klasu rezultiralo smanjenjem iznosa sume kvadrirane udaljenosti točke opažanja od točke težišta klase unutar klastera.
4. Izračunava se prosječna vrijednost točaka promatranja svakog klastera kako bi se dobilo novih k točaka težišta klastera.
5. Koraci 2, 3 i 4 se ponavljaju sve dok se točke promatranja ne prestanu dodjeljivati novim klasterima ili algoritam dostigne maksimalni broj iteracija.

Matlab koristi specifičnu verziju k-means algoritma naziva k-means++ koja koristi heuristiku za pronalaženje sjemena težišta. Ova verzija dobiva na brzini i kvaliteti krajnjeg rezultata.

Sjeme biva birano, dano k, na sljedeći način:

1. Nasumično se odabire jedna od točaka promatranja. Ta točka se smatra početnim težištem i naziva se $c_1$.
2. Računa se udaljenost svake točke od $c_1$. Udaljenost između $c_j$ i promatranja m se zapisuje kao $d(x_m, c_j)$.
3. Nasumično se odabire sljedeće težište nazvano $c_2$ sa sljedećom vjerojatnošću:

$$\frac{d^2(x_m, c_1)}{\sum_{j=1}^{n} d^2(x_j, c_1)}$$

4. Kako bi se odabralo sljedeće težište j se koristi sljedeća procedura:
    a. Računa se udaljenost svake točke promatranja od svakog težišta i svaka točka promatranja se dodjeljuje klasteru sa njoj najbližim težištem.

b. Za m=1,…,n i p=1,…,j-1, nasumično se odabire težište j sa sljedećom vjerojatnošću gdje vrijedi da je $C_p$ skup svih promatranja koja su najbliža težištu $c_p$ i $x_m$ pripada $C_p$-u. Točnije, odabire se svako sljedeće težište sa vjerojatnošću koja je proporcionalna udaljenosti od samog sebe do najbližeg težišta koje smo već odabrali.

$$\frac{d^2(x_m, c_p)}{\sum_{\{h;x_h \in C_p\}}^{n} d^2(x_h, c_p)}$$

5. Ponavlja se korak 4 sve dok nije odabrano k težišta.[1]

[1] MathWorks, 2013.“k-means clustering“, Pristupljeno: 16. rujan 2022. https://www.mathworks.com/help/stats/kmeans.html

## 4. Pronalaženje optimalnog broja kategorija

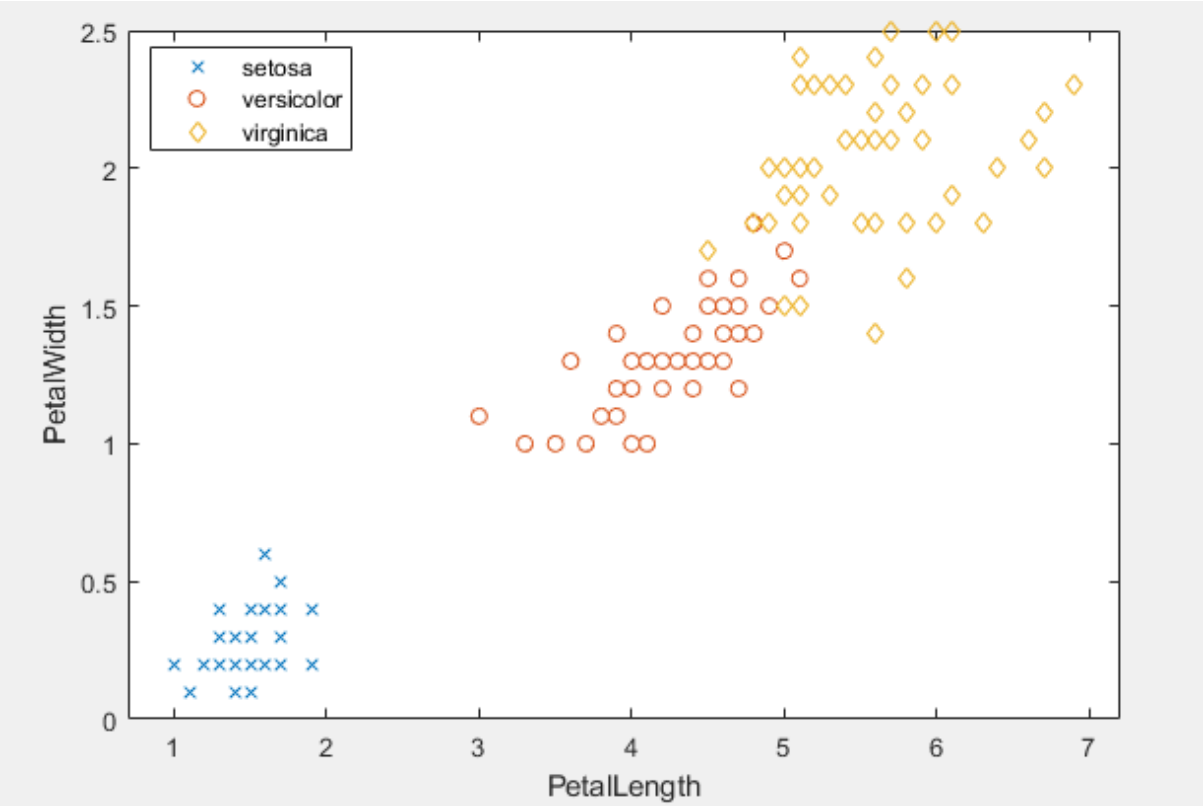
U svrhu testiranja algoritama biti će korištena dva različita skupa podataka kako bi se dobio bolji uvid u jake i slabe strane ponuđenih algoritama.

Prvi će biti Fisherov skup podataka iz 1963. godine o dužinama i širinama latica i čašičnih listića 50 cvjetova koji pripadaju trima vrstama perunika sljedećih naziva: Setosa, Versicolor i Virginica.

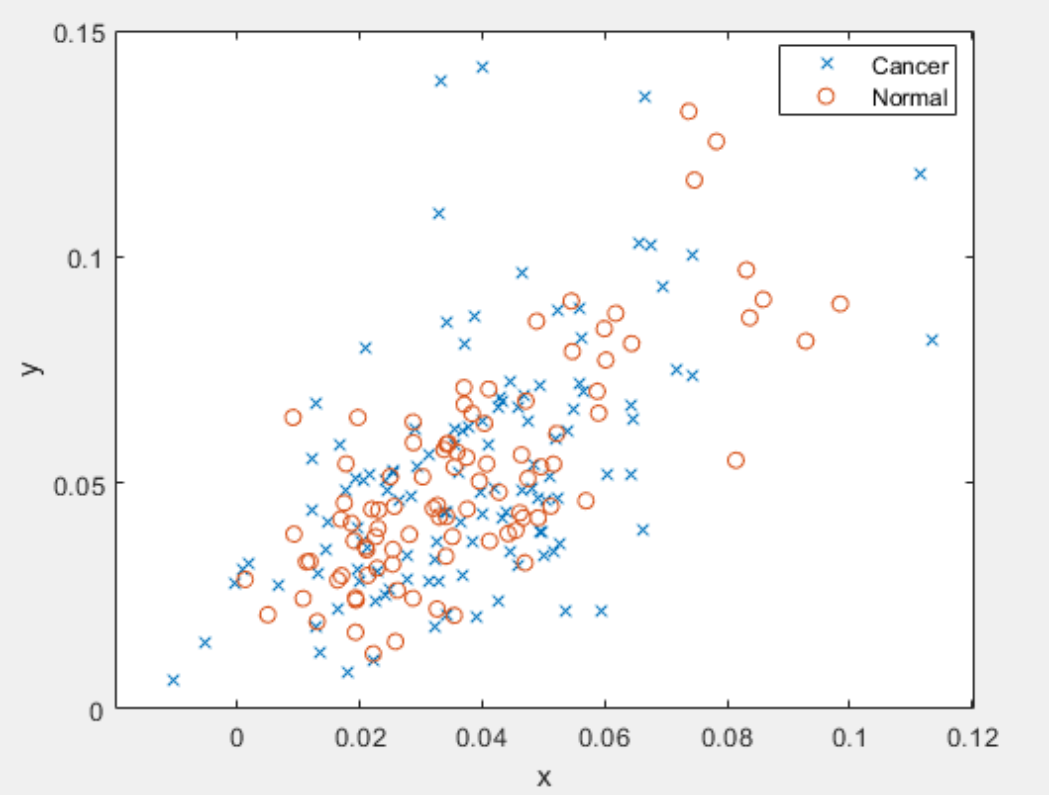
Drugi skup podataka sadrži 160 pacijenata i za svakog 4000 vrijednosti odabranih bio markera koji ukazuju na prisutnost raka jajnika kod određenog pacijenta te su pacijenti podijeljeni na dvije skupine: oni kojima je dijagnosticiran rak jajnika i oni kojima nije.

Biti će predstavljeno stvarno rješenje i nakon toga za svaki usporediti njegovo rješenje sa onim točnim.

Na sljedećim grafovima su prikazani skupovi podataka podijeljeni u njihove stvarne kategorije dok će se u sljedećim poglavljima svaki od skupova podataka realizirati svaki od algoritama za utvrđivanje optimalnog broja kategorija i usporediti rezultate algoritama sa stvarnim rezultatima. Postotak točnosti k-means algoritma je utvrđen usporedbom liste stvarnih kategorija poredanih u retka sa listom koju je generirao algoritam.




*Slika 2 Stvarna kategorizacija vrsta perunika s obzirom na duljinu latica*




*Slika 3 Stvarna kategorizacija pacijenata s obzirom na dijagnozu*

Koraci koje je potrebno poduzeti u Matlab-u kako bi se došlo do istih rezultata su sljedeći:

1. Učitati željeni skup podataka iz Matlab-ove riznice dostupnih skupova.

```
load fisheriris
```

2. Postaviti generirane nasumične varijable na početnu vrijednost kako bi rezultati mogli biti reproducirani unatoč nasumičnoj prirodi k-means algoritma.

```
rng("default")
```

3. Pomoću funkcije *evalclusters* dobiti optimalni broj klastera sukladno kriterijima odabranog algoritma zajedno sa kategoriziranim skupom podataka. Funkcija prima pet argumenta koji su redom:
    a. Skup podataka koji biva analiziran.
    b. Algoritam koji se primjenjuje u svrhu kategorizacije.
    c. Algoritam koji se primjenjuje u svrhu pronalaženja optimalnog broja kategorija na koje podijeliti dani skup podataka.
    d. Ime argumenta funkcije koji ćemo sljedeći specificirati.
    e. Popis brojeva koje ćemo testirati kao potencijalni broj kategorija.

```
evaluation = evalclusters(meas, "kmeans", algorithm, "Klist", 1:6)
```

4. Prikazati rezultate grafički postupkom po želji.

### 4.1. Calinski-Harabasz algoritam

Ponekad zvan i kriterij omjera varijance (eng. variance ratio criterion, VRC). Dobro definirani klasteri imaju veliku varijancu između klastera i malu varijancu unutar klastera. Što je veći omjer VRC, to je bolja odvojenost podataka. Za određivanje optimalnog broja klastera, potrebno je maksimizirati VRC s obzirom na broj klastera. Optimalan broj klastera odgovara rješenju s najvećom vrijednošću Calinski-Harabasz indeksa.

Formula za računanje kriterija omjera varijance gdje $SS_B$ predstavlja ukupnu varijancu između klastera, $SS_W$ ukupnu varijancu unutar klastera, $N$ broj opažanja i $k$ odabrani broj klastera je sljedeća:

$$VRC_k = \frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)}$$

Za računanje ukupne varijance između klastera se koristi sljedeća formula gdje je $k$ broj klastera, $n_i$ broj opažanja unutar klastera, $m_i$ težište klastera $i$, $m$ je ukupna srednja vrijednost uzorka podataka dok izraz $\|m_i - m\|^2$ predstavlja $L^2$ normu (euklidsku udaljenost) između dva vektora.

$$SS_B = \sum_{i=1}^{k} n_i \|m_i - m\|^2$$

Za računanje ukupne varijance unutar klastera se koristi sljedeća formula gdje je $k$ broj klastera, $c_i$ $i$-ti klaster, $m_i$ težište klastera $i$ dok izraz $\|x - m_i\|^2$ predstavlja $L^2$ normu (euklidsku udaljenost) između dva vektora.[2]

$$SS_W = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - m_i\|^2$$

Matlab funkcija za izračunavanje vrijednosti Calinski-Harabasz indeksa se sastoji od sljedećih linija koda:

```
function CH = getCH(this, centroids, Ni, SUMD, NC)
      SSW = sum(SUMD,1);
      SSB = (pdist2(centroids,this.GlobalMean)).^2;
      SSB = sum(Ni.*SSB);
      CH =(SSB/(NC-1))/(SSW/(this.NumObservations-NC));
end
```

Slijedi objašnjenje svake linije koda:

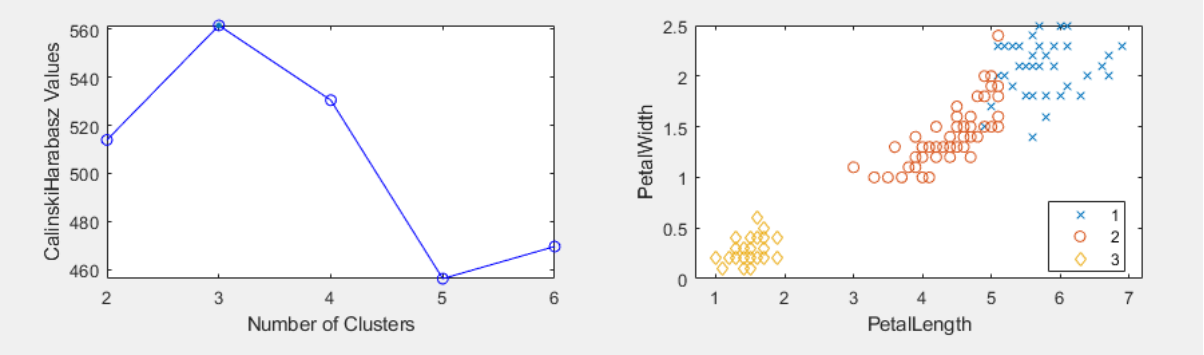
1. Funkcija kao argumente prima sljedećih 5 parametra:

[2] MathWorks. 2013. ″Calinski-Harabasz criterion clustering evaluation″ Pristupljeno: 16. kolovoz 2022. https://www.mathworks.com/help/stats/clustering.evaluation.calinskiharabaszevaluation.html

a. *this* predstavlja objekt koji je rezultat primjene k-means metode na dani skup podataka
b. *centroids* predstavlja redom težišta svakog od klastera
c. *Ni* predstavlja poredanu listu ukupnog broja točaka u svakom od klastera
d. *SUMD* predstavlja poredanu listu suma euklidskih udaljenosti između točaka danog klastera
e. *NC* predstavlja ukupni broj klastera u cjelobrojnom obliku

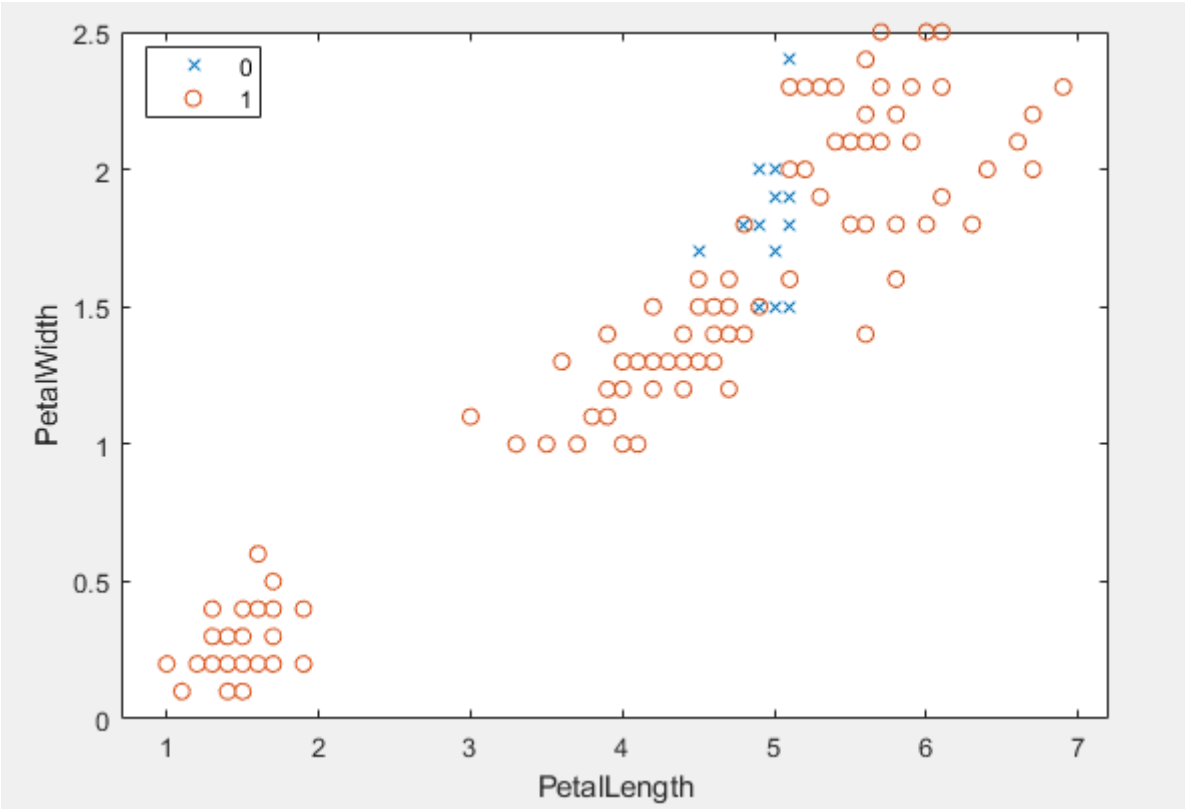
2. Ukupna varijanca unutar svakog od klastera se dobiva tako što se sve udaljenosti među točkama svakog od klastera sumiraju.
3. Prvi korak za dobivanje ukupne varijance među klasterima je računanje udaljenosti svakog težišta od globalne srednje vrijednosti i kvadriranje dobivenih rezultata.
4. Drugi korak za dobivanje ukupne varijance među klasterima je umnažanje svake točke koja je dio skupa podataka sa gore dobivenim vrijednostima. Potom se dobivena lista sumira u jedinstvenu vrijednost.
5. Prethodno izračunate varijable se uvrštavaju u formulu za računanje kriterija omjera varijance.

Rezultat primjene algoritma na dane skupove podataka je sljedeći:




*Slika 4 Rezultat primjene Calinski-Harabasz algoritma na Iris skup podataka*

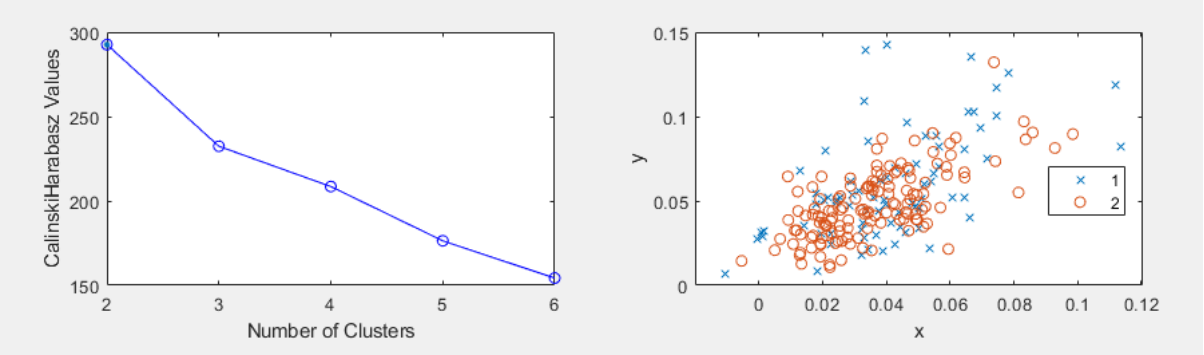
Algoritam je podijelio skup podataka na stvarni broj kategorija i velika većina članova je točno raspoređena u pripadajuće kategorije iako točnost još uvijek nije potpuna. Ona je na razini 89.33% što čini 16 točaka dodijeljenima pogrešnim kategorijama iz skupa 150 ponuđenih točaka.




*Slika 5 Graf točnosti kategorizacije točaka Iris skupa podataka*

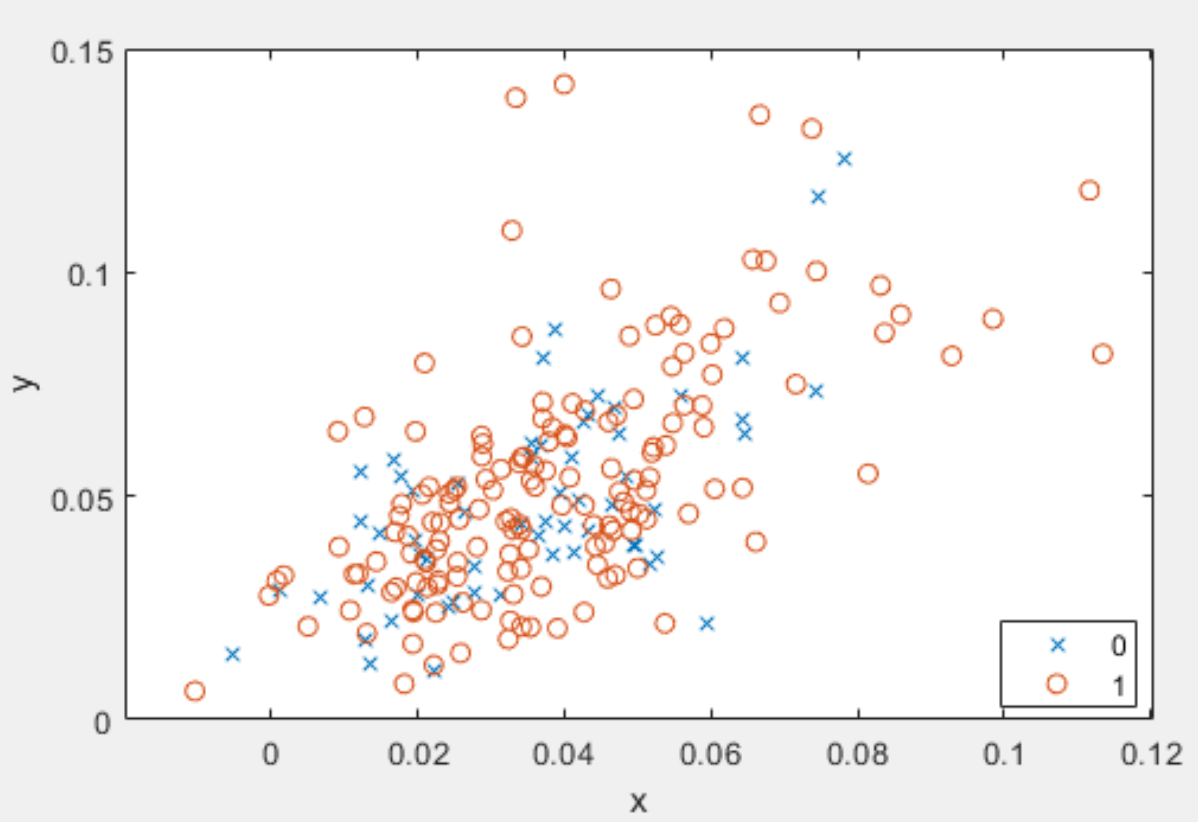
Na prethodnoj slici se može detaljnije sagledati koje su točke kategorizirane točno, a koje nisu. Jedinica predstavlja pravilno kategorizirane točke dok nula predstavlja nepravilno kategorizirane točke.

Na skupu podataka pacijenata sa rakom jajnika je rezultat bio sljedeći:




*Slika 6 Rezultat primjene Calinski-Harabasz algoritma na skup podataka raka jajnika*

Algoritam je uspio podijeliti skup podataka na dvije kategorije kako je bilo i pravilno. Točnost k-means algoritma je u ovom slučaju dostigla 72.22%.




*Slika 7 Graf točnosti kategorizacije točaka skupa podataka raka jajnika*

### 4.2. Davies-Bouldin algoritam

Davies-Bouldin kriterij se temelji na omjeru udaljenosti unutar klastera i između klastera. Optimalno rješenje klasteriranja ima najmanju vrijednost Davies-Bouldin indeksa.

Njega definiramo sljedećom formulom gdje $D_{i,j}$ predstavlja omjer udaljenost unutar naspram između klastera:

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{j\neq i}\{D_{i,j}\}$$

Formula za računanje omjera udaljenost unutar naspram između klastera gdje je $\bar{d}_i$ prosječna udaljenost između svake točke u *i*-tom klasteru i težište *i*-tog klastera, $\bar{d}_j$ predstavlja prosječnu udaljenost između svake točke *j*-tog klastera i težište *j*-tog klastera dok $d_{i,j}$ predstavlja euklidsku udaljenost između težišta *i*-tog i *j*-tog klastera je sljedeća:

$$D_{i,j} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{i,j}}$$

Maksimalna vrijednost $D_{i,j}$ predstavlja najgori slučaj omjera između klastera za klaster

*i*. Optimalno rješenje klasteriranja ima najmanju vrijednost Davies-Bouldin indeksa.[3]

Metoda prevedena u Matlab kod ima sljedeći oblik:

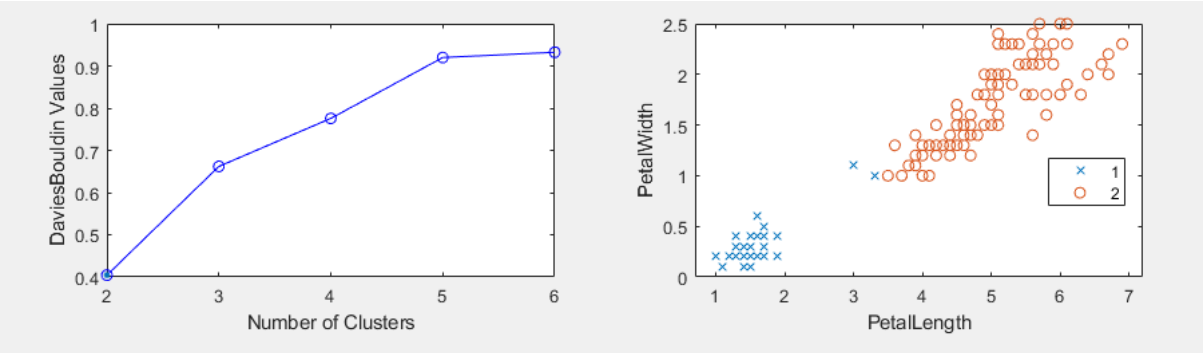
```
function DB = getDB(this,index)
    clusts = unique(index);
    num = length(clusts);
    if num == 1
        DB = nan;
        return;
    end
    centroids = NaN(num,size(this.PrivX,2));

    aveWithinD= zeros(num,1);
    for i = 1:num
        members = (index == clusts(i));
        if any(members)
            centroids(i,:) = mean(this.PrivX(members,:),1) ;
            aveWithinD(i)= mean(pdist2(this.PrivX(members,:),centroids(i,:)));
        end
    end

    interD = pdist(centroids,'euclidean');
    R = zeros(num);
    for i = 1:num
        for j=i+1:num %j>i
             R(i,j)= (aveWithinD(i)+aveWithinD(j))/ interD((i-1)*(num-i/2)+j-i);
        end
    end
    R=R+R';

    RI = max(R,[],1);
    DB = mean(RI);
end
```

U slučaju primjene algoritma na Iris skup podataka rezultat je bio sljedeći:

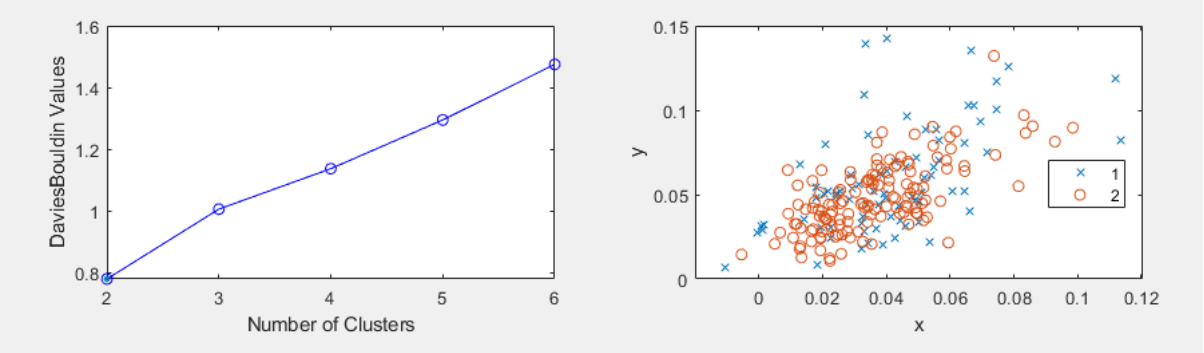



*Slika 8 Rezultat primjene Davies-Bouldin algoritma na Iris skup podataka*

Algoritam je skup podataka podijelio na dvije kategorije i time pogrešno procijenio.

Primjena algoritma na skup podataka raka jajnika je dala pravilnu podjelu:

[3] MathWorks. 2013. ″Davies-Bouldin criterion clustering evaluation″ Pristupljeno: 16. kolovoz 2022. https://www.mathworks.com/help/stats/clustering.evaluation.daviesbouldinevaluation.html




*Slika 9 Rezultat primjene Davies-Bouldin algoritma na skup podataka raka jajnika*

Graf točnosti više nije potrebno prikazivati s obzirom da se odabirom istog broja kategorija primjenjuje k-means metoda sa jednakim parametrima kod svih pokušaja.

### 4.3. Gap algoritam

Vrijednost razmaka (eng. gap value) se definira sljedećom formulom za koju vrijedi da *n* predstavlja broj promatranja, *k* je broj klastera koje promatramo i $W_k$ predstavlja objedinjenu mjeru disperzije unutar klastera:

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

Formula za računanje objedinjene mjere disperzije unutar klastera gdje $n_r$ predstavlja broj točaka u klasteru *r* dok $D_r$ predstavlja sumu udaljenosti između svih točaka u klasteru *r* je sljedeća:

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

Očekivana vrijednost izraza $E_n^*\{\log(W_k)\}$ se određuje Monte Carlo uzorkovanjem iz referentne distribucije, a $\log(W_k)$ se izračunava iz uzorka. Vrijednost razmaka definirana je čak i za rješenja klasteriranja koja sadrže samo jedan klaster i može se koristiti s bilo kojom metrikom udaljenosti. Međutim, kriterij razmaka je računalno skuplji od ostalih kriterija za ocjenu klasteriranja, jer se algoritam klasteriranja mora primijeniti na referentne podatke za svako predloženo rješenje klasteriranja.[4]

Gap metoda u slučaju Matlab koda na najvišoj razini poprima sljedeći oblik:

```
function [this,IDX] = getGapValue(this,j)
    if ~this.ValidResult(j)
        IDX = [];
        this.CriterionValues(j) = nan;
        this.LogW(j) = nan;
        return;
```

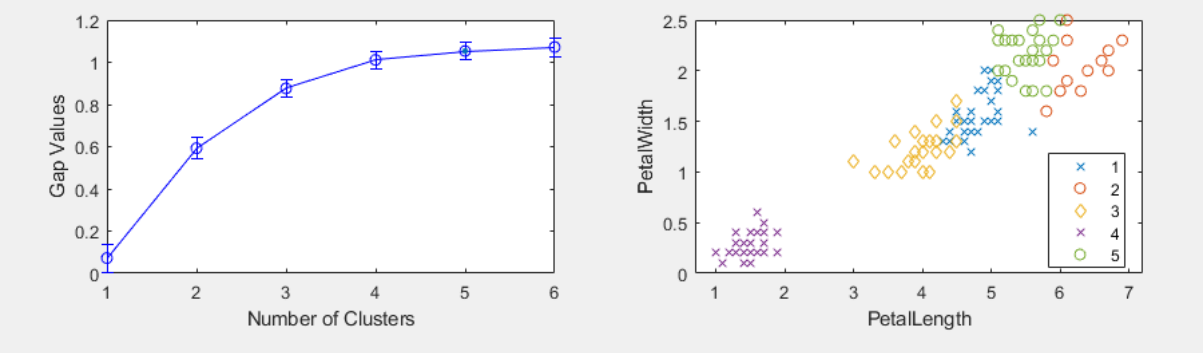
[4] MathWorks. 2013. ″Gap criterion clustering evaluation″ Pristupljeno: 16. kolovoz 2022. https://www.mathworks.com/help/stats/clustering.evaluation.gapevaluation.html

```
    end

    NC = this.InspectedK(j);
    if ischar(this.ClusteringFunction) && this.FunLoc==1 && this.distLoc==1
        [IDX, ~, SUMD] = kmeans(this.PrivX,NC,'rep',5,'empty','singleton');
        this.LogW(j) = log(sum(SUMD,1));
    else
        IDX = this.evalFun(NC);
        if ~isempty(IDX)
        this.LogW(j) = getLogW(this.PrivX,IDX,this.Distance,this.distLoc==1);
        else
            this.LogW(j)= nan;
            this.ValidResult(j) = false;
        end
    end

    this.CriterionValues(j) = this.ExpectedLogW(j)-this.LogW(j);
end
```

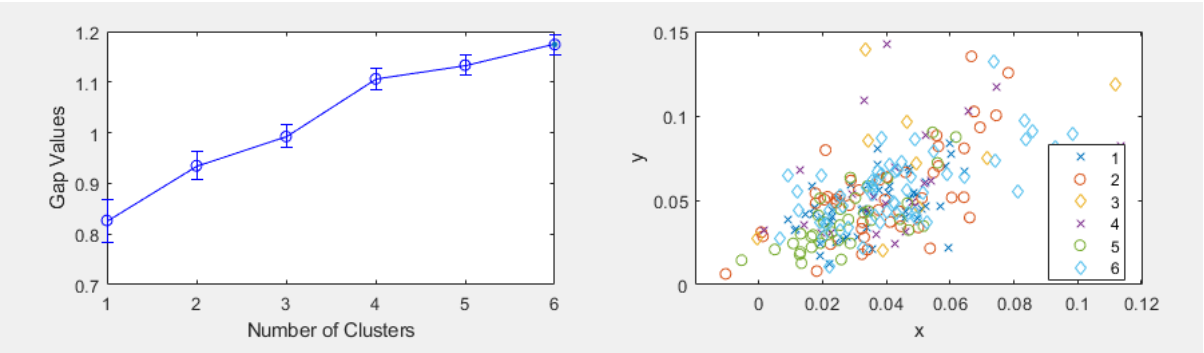
Primjena algoritma na Iris skup podataka je rezultirala sljedećim:




*Slika 10 Rezultat primjene Gap algoritma na Iris skup podataka*

Algoritam je skup podataka podijelio na pet kategorija i time pogriješio.

Primjena algoritma na skup podataka raka jajnika je ponovno rezultirala odabirom velikog broja kategorija što nije rezultiralo točnim odgovorom:




*Slika 11 Rezultat primjene Gap algoritma na skup podataka raka jajnika*

### 4.4. Silhouette algoritam

Vrijednost siluete za svaku točku mjera je koliko je ta točka slična drugim točkama u istom klasteru, u usporedbi s točkama u drugim klasterima.

Vrijednost siluete $s_i$ za *i*-tu točku se definira sljedećom formulom gdje je $a_i$ prosječna udaljenost *i*-te točke od drugih točaka u istom klasteru u kojem se nalazi *i* dok je $b_i$ najmanja prosječna udaljenost *i*-te točke od točaka u nekom drugom klasteru, minimiziranih preko klastera:

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$

Vrijednosti siluete su u rasponu od –1 do 1. Visoka vrijednost siluete označava da je točka dobro usklađena s vlastitim klasterom, a loše s drugim klasterima. Ako većina točaka ima visoku vrijednost siluete, tada je rješenje grupiranja prikladno. Ako mnogo točaka ima nisku ili negativnu vrijednost siluete, tada rješenje klasteriranja može imati previše ili premalo klastera. Vrijednosti siluete kao kriterij procjene klasteriranja se može koristiti s bilo kojom metrikom udaljenosti.[5]

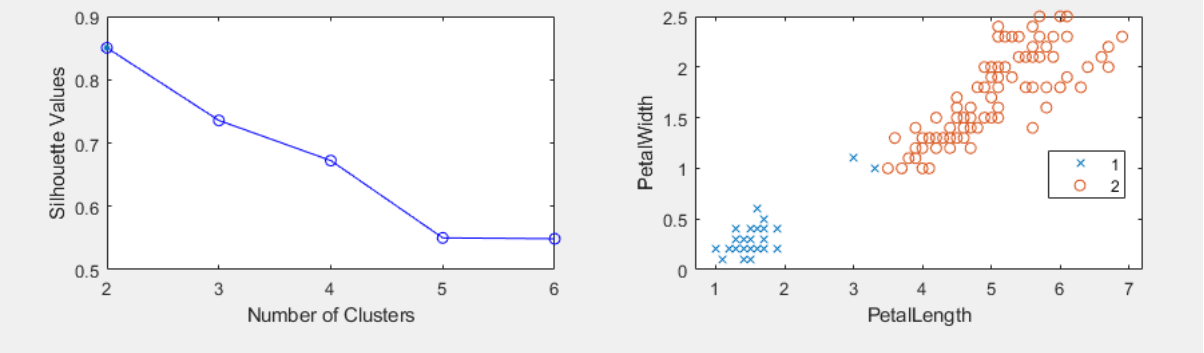
Matlab kod za odabrani algoritam je sljedeći:

```
function [distance,distLoc] = checkDistance(distance,nIn)
    if ischar(distance)
        distList = {'sqEuclidean' 'Euclidean' 'cityblock'  'cosine' 'correlation'
'Hamming', 'Jaccard'};
        [distance,distLoc] = internal.stats.getParamVal(distance,distList,'distance');
    elseif isnumeric(distance)

        if (size(distance,1) == 1) && (size(distance,2) == .5*nIn*(nIn-1))
            if any(distance < 0)
error(message('stats:clustering:evaluation:SilhouetteEvaluation:NegDistanceValues'));
            end
        else
            error(message('stats:silhouette:DistanceMatrixNotUpperTri'));
        end
        distLoc = 0;
elseif isa(distance,'function_handle')
       distLoc = 0;
else
    error(message('stats:clustering:evaluation:SilhouetteEvaluation:BadDist'));
end
```

Primjenom algoritma na Iris skup podataka smo dobili sljedeći rezultat:

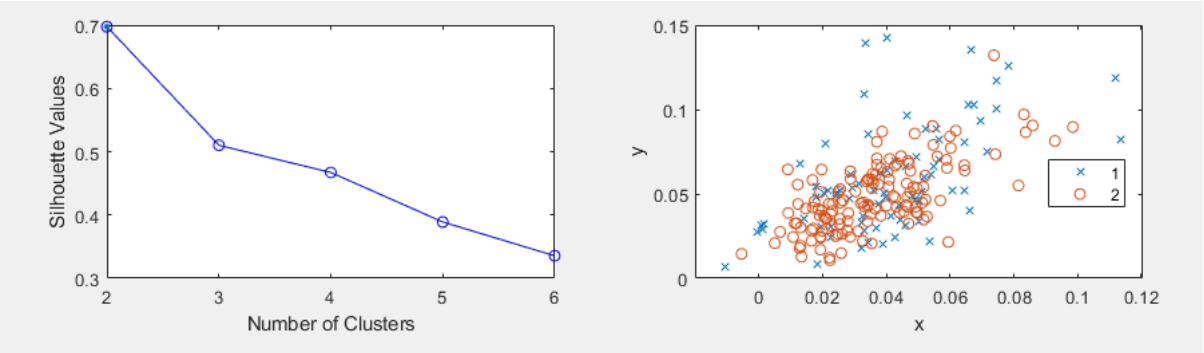
[5] MathWorks. 2013. ˝Silhouette criterion clustering evaluation˝ 16. kolovoz 2022. https://www.mathworks.com/help/stats/clustering.evaluation.silhouetteevaluation.html




*Slika 12 Rezultat primjene Silhouette algoritma na Iris skup podataka*

Ovaj algoritam je odredio da je optimalni broj kategorija unutar ovog skupa podataka dva i time pogriješio.

Primjena algoritma na skup podataka raka jajnika rezultira točnom podjelom:




*Slika 13 Rezultat primjene Silhouette algoritma na skup podataka raka jajnika*

# 5. Zaključak

Dobiveni rezultati se mogu preslikati u sljedeću tablicu gdje brojevi u ćelijama predstavljaju broj kategorija koje je dani algoritam za taj redak presudio kao optimalni:

*Tablica 1 Usporedba procjena optimalnog broja kategorija*

| | Iris | Rak jajnika |
|---|---|---|
| Calinski-Harabasz | 3 | 2 |
| Davies-Bouldin | 2 | 2 |
| Gap | 5 | 6 |
| Silhouette | 2 | 2 |
| Stvarni broj kategorija | 3 | 2 |

Iz dobivenih rezultata se mogu izvući sljedeći zaključci:

- Calinski-Harabasz algoritam se dokazao kao najuspješniji s obzirom da je za oba skupa podataka dao točnu procjenu broja kategorija.
- Gap algoritam se pokazao kao algoritam sa najvećom pogreškom u procjeni broja kategorija i njegovi su rezultati težili najvećem broju kategorija od ponuđenih.
- Davies-Bouldin i Silhouette algoritmi su demonstrirali procjene koje su težile manjem broju kategorija od ponuđenih što se dokazalo kao optimalna formula za skup podataka podijeljen na dvije kategorije, ali ne i na onaj podijeljen u tri kategorije.

## 6. Literatura


- Gutttag, John. 2016. Introduction to Computation and Programming using Python. Cambridge: The MIT Press.
- Guttag, John. 2017. „12. Clustering“, Objavljeno Svibanj 19, 2017. YouTube, 50 min. https://www.youtube.com/watch?v=esmzYhuFnds
- Russel, Stuart J. i Norvig Peter. 2009. Artificial Intelligence: A Modern Approach. Boston: Prentice Hall.
- MathWorks Help Center, posljednja izmjena 2022. https://www.mathworks.com/help/ (pristupljeno 16. rujna 2022.)

## 7. Popis slika

# 8. Popis tablica

## 9. Sažetak

Ovaj rad je započeo sa opisom klasteringa i upoznao čitatelja sa dva pristupa rješavanju problema u obliku algoritama koji su bili k-means i hijerarhijsko klasteriranje. U nastavku se tekst pozornije posvetio k-means algoritmu te su opisani rad tog specifičnog algoritma i napomenuta je činjenica da algoritam zahtjeva prethodno poznavanje broja skupina na koje će se dani skup podataka podijeliti. U sljedećem poglavlju su bila predstavljena dva skupa podataka koji su bili korišteni u svrhu utvrđivanja efikasnosti algoritama za pronalaženje optimalnog broja kategorija za k-means algoritam. Taj dio je slijedilo detaljnije predstavljanje svakog od četiri odabrana algoritma na razini ideje i matematičkih formula te njihovo testiranje na odabranim skupovima podataka. Na kraju je uspoređena točnost algoritama pri odabiru optimalnog broja kategorija.

**Ključne riječi:** klasteriranje, k-means, algoritmi, točnost

## 10. Abstract

This paper addresses optimization of clustering techniques, considering two clustering approaches; the k-means algorithm and hierarchical clustering. Following that topic, this work focuses in particular on the k-means algorithm, the description of its inner mechanisms and the fact that the algorithm requires prior knowledge of the number of groups into which the given set of data will be divided. Namely, two sets of data are presented in order to determine the efficiency of different algorithms for determining the optimal number of categories for the k-means algorithm. Four algorithms are compared by their theoretic background, and tested on selected data sets. Finally, the accuracy of the algorithms, when choosing the optimal number of categories, is compared.

**Keywords:** clustering, k-means, algorithms, accuracy