SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 6491

# ANALIZA GRUPIRANJA PODATAKA UPORABOM PLATFORME ELKI

Mislav Križan

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 6491

# ANALIZA GRUPIRANJA PODATAKA UPORABOM PLATFORME ELKI

Mislav Križan

Zagreb, lipanj 2020.

**SVEUČILIŠTE U ZAGREBU**
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 13. ožujka 2020.

# ZAVRŠNI ZADATAK br. 6491

| | |
|---|---|
| Pristupnik: | **Mislav Križan (0036507558)** |
| Studij: | Računarstvo |
| Modul: | Računarska znanost |
| Mentor: | izv. prof. dr. sc. Alan Jović |

Zadatak: **Analiza grupiranja podataka uporabom platforme ELKI**

Opis zadatka:

Analiza grupiranja (engl. clustering) podataka tehnika je nenadziranog učenja kojoj je cilj ustanoviti postojanje dviju ili više grupa u podatcima koji nemaju određenu ciljnu kategoriju. Grupiranje podataka je značajno u mnogim područjima gdje nisu dostupne kvantifikacije neke ciljne funkcije. Za grupiranje podataka razvijeni su različiti algoritmi zasnovani na hijerarhijskoj udaljenosti između uzoraka, gustoći grupa, centroidima grupa i drugi. U ovom završnom radu potrebno je proučiti i opisati platformu ELKI za dubinsku analizu podataka. Platforma ELKI ima implementiranih više postupaka za analizu grupiranja podataka. Za potrebe završnog rada potrebno je na nekoliko slobodno dostupnih skupova podataka (npr. iz repozitorija UCI Irvine Machine Learning) provesti analizu grupiranja podataka koristeći platformu ELKI te kvalitativno i kvantitativno usporediti rezultate dobivene različitim implementiranim postupcima.

Rok za predaju rada: 12. lipnja 2020.

## Sadržaj

# Uvod

U današnjem dobu informacija, osnovni uvjet donošenja ispravnih zaključaka je raspolaganje valjanim i potrebnim informacijama. U rješavanju problema, izuzetno je važno razlikovanje bitnih od nebitnih informacija, uočavanje potrebnih od onih informacija kojima raspolažemo, kao i međusobna povezanost tih informacija.

Radi rješavanja takvih problema razvijeni su razni procesi dubinske analize podataka. Jedan od jako bitnih načina dubinske analize podataka je analiza grupiranja podataka (engl. *clustering analysis)*. Analiza grupiranja podataka je nenadzirana tehnika kojoj je zadatak ustanoviti postojanje dvije ili više grupa tako da se povezuju podaci temeljem njihove sličnosti u grupe s ciljem da su grupirani objekti više slični jedni drugima i više različiti onim objektima izvan grupe.

Postoje mnogi razlozi za korištenje analize grupiranja, poput statističke analize podataka, dolazak do novih spoznaja danog problema, itd. Grupiranje podataka nije jedan algoritam, već generalni problem koji treba riješiti. Postoje mnogi modeli koji su razvijeni za razvrstavanje podataka u smislene grupe, međutim ne postoji savršeni model za sve probleme. Svaki pojedinačni pristup problemu i njegovo rješavanje ima svoje prednosti i nedostatke. Za pravilan odabir pojedinog modela kod rješavanja specifičnog problema, bitno je znati njegovu funkcionalnost, te njegove jakosti i slabosti. U ovom radu će biti prikazane neke osnovne grupe, tj. modeli i na koji način oni prilaze danom problemu i kako ga pokušavaju riješiti. Također, biti će i prikazani i njihovi reprezentativni algoritmi te njihove mane i slabosti koristeći platformu ELKI.

# 1. Opis platforme ELKI

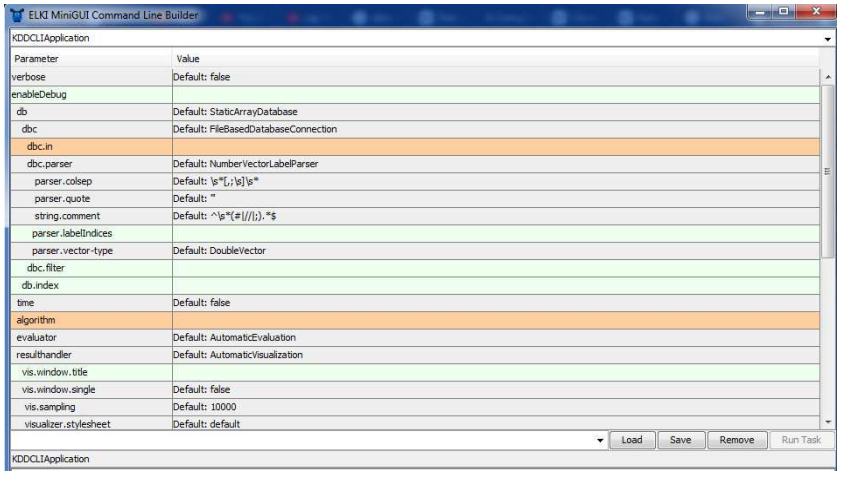
Okolina za razvijanje aplikacija za otkrivanje znanja u skupovima podataka (engl. *Knowledge Discovery in Datasets,* KDD) s osloncem na strukture indeksa pod nazivom ELKI (engl. *Environment for DeveLoping KDD-Applications Supported by Index-Structures*) je otvoreni softver za dubinsku analizu podataka pisan u Javi [1]. Glavni fokus u ELKI-ju je istraživanje pomoću razvijenih algoritama, gdje je naglasak na analizu grupiranja i detekciju stršećih vrijednosti.

Način pristupanju podataka unutar ELKI-ja je modeliran na sličan način kao model baze podataka. Podaci se spremaju u strukturu stupaca te zbog takvog oblika strukture određeni algoritmi se mogu lagano implementirati pa je pristup kolekcijama brz i učinkovit [1]. Također, ELKI koristi modele sučelja programskog jezika Java zbog čega je jednostavan za nadograditi.

ELKI dolazi i s već izgrađenom aplikacijom koja se može koristiti za dubinsku analizu podataka bez potrebe za poznavanjem jezika Java.

Samo korištenje već razvijene aplikacije se sastoji od dva koraka.

1. Odabir skupa podataka na kojem će se odvijati analiza i odabir algoritma koji će se koristiti za analizu podataka uz podešavanje parametara za pojedini algoritam, Slika 1.1 Prozor za odabir skupa podataka i načina analize.
2. Pokretanje analize s *run task*.




Slika 1.1 Prozor za odabir skupa podataka i načina analize [2]

# 2. Grupa algoritama grupiranja zasnovana na centroidima

U centroidno-orijentiranom grupiranju, grupe su predstavljene s vektorom središta [4].

Ovaj model gleda grupe kao eliptične skupine gdje je grupa određena sa središtem skupine i blizinom uzorka od središta.

Reprezentativni primjerak algoritama centroidnog grupiranja je algoritam k-srednjih vrijednosti (engl. *k-means*).

## 2.1. Algoritam *k-means*

Algoritam grupiranja kojemu je cilj minimizirati varijancu unutar pojedine grupe, tj. minimizirati kvadriranu euklidsku udaljenost između središta grupe i pojedinog uzorka unutar te grupe [5].

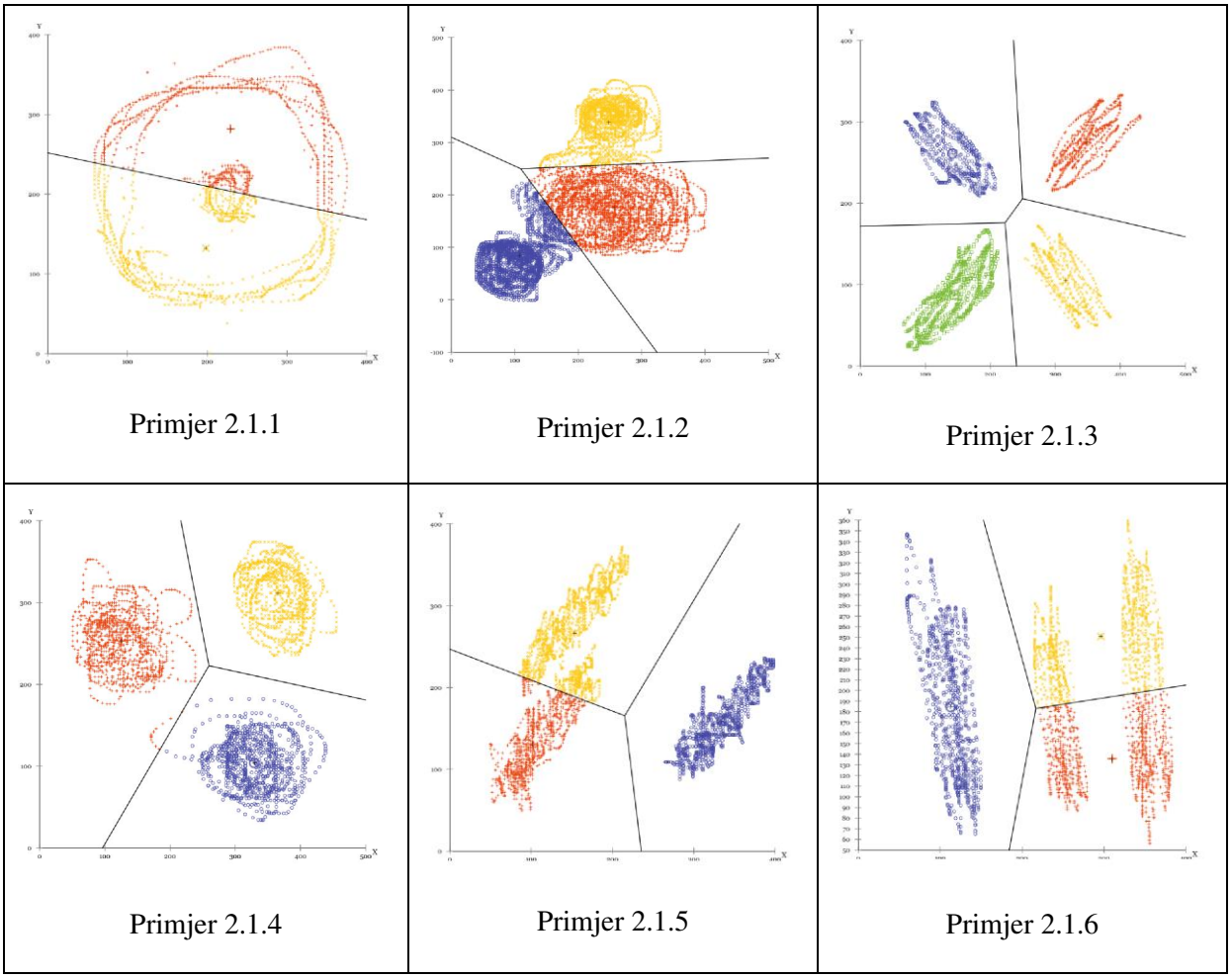
Problem minimizacije udaljenosti je dosta težak, međutim učinkoviti heuristički algoritmi za rješenje tog problema kao što je *k-means* jako brzo konvergiraju k rješenju. Ipak, *k-means* nije optimalan algoritam (rješenje se ne mora nužno nalaziti u globalnom optimumu)

Postupak algoritma *k-means*:

0. korisnik odredi parametar *k* (broj grupa u algoritmu);
1. određuju se početna središta grupa, nasumično ili na neki prethodno definiran način;
2. algoritam raspodjeljuje svaki uzorak u najbližu grupu;
3. izračunava se nova pozicija središta (centroida) preko svih uzoraka koji su dodijeljeni toj grupi;
4. ponavljaju se koraci 2 i 3 sve dok ne nastupi konvergencija k rješenju.

Platforma ELKI nudi različite implementacije algoritma *k-means* s manjim varijacijama koje se očituju u brzini, načinu biranja i pomicanja središta grupa, itd.

Neki od teorijskih primjera prikaza slučajeva kada je korisno, a kada ne koristiti algoritam *k-means* prikazano je Slika 2.1 Primjeri *k-means* algoritma.



Primjer 2.1.1 | Primjer 2.1.2 | Primjer 2.1.3

Primjer 2.1.4 | Primjer 2.1.5 | Primjer 2.1.6

Slika 2.1 Primjeri *k-means* algoritma

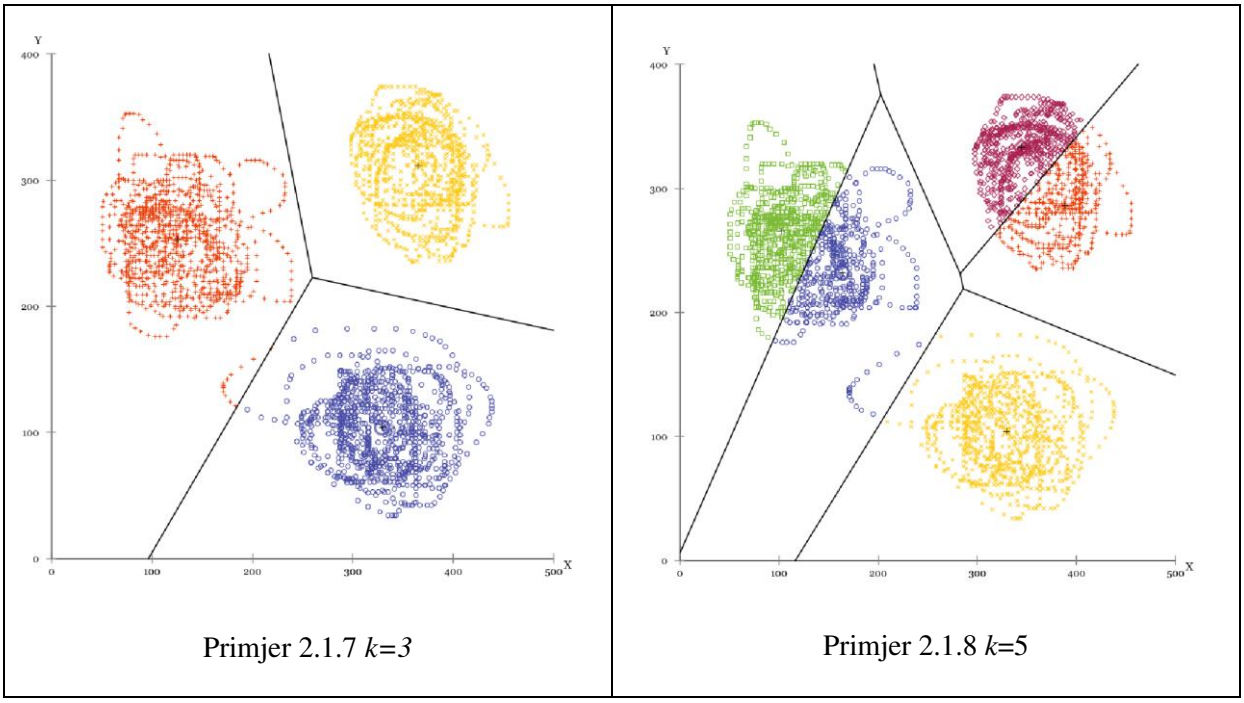
Primjer 2.1.1 pokazuje manu *k-meansa*, gdje on može raspodijeliti podatke na većinom konveksne grupe koje se po mogućnosti ne preklapaju, što čini prvi primjer gotovo nemogućim za raspodijeliti na dvije okom vidljive grupe, gdje se jedna nalazi unutar druge.

Primjer 2.1.6 pokazuje konveksnost algoritma *k-means* gdje, umjesto da se uzmu tri okom vidljive grupacije podataka, algoritam prepolovi dvije okom vidljive grupe.

U Primjer 2.1.2 vidi se mala razlika između okom vidljivih grupacija i *k-meansa*, gdje se granični dijelovi pridodaju krivoj grupi.

Najbolji primjer korištenja za *k-means* je Primjer 2.1.4, gdje su sve tri grupe dovoljno udaljene jedna od druge da *k-means* može lagano riješiti problem grupacije.

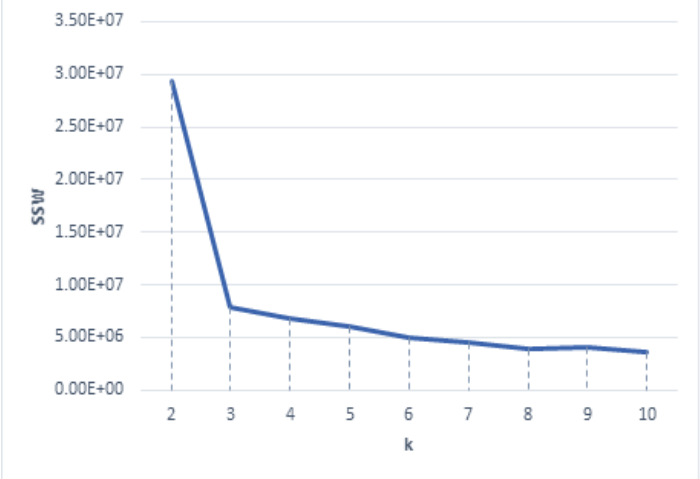
Međutim, tu dolazi problem određivanja broj grupa (parametra *k*). Na Slika 2.2 Usporedba parametra *k* može se vidjeti bitnost parametra *k*, gdje je na Primjer 2.1.7 *k=3* dok na Primjer 2.1.8 je *k=5* što rezultira dodatnim komplikacijama i dovodi do posve drukčijeg rješenja.




Primjer 2.1.7 *k=3*

Primjer 2.1.8 *k*=5

Slika 2.2 Usporedba parametra *k*




Slika 2.3 Graf varijance s obzirom na *k*

Jedno od predloženih rješenja kako odrediti parametar *k* je preko grafa varijance s obzirom na *k*, koristeći metodu lakta (lakat predstavlja dio grafa gdje nakon nekog većeg pada dolazi sljedeći pad koji je puno manji), Slika 2.3 Graf varijance s obzirom na *k* prikazuje graf varijance s obzirom na *k*.

Može se vidjeti da se „lakat“ pokazuje na

grafu za $k$=3, što znači da je to u ovom slučaju optimalan broj grupa.

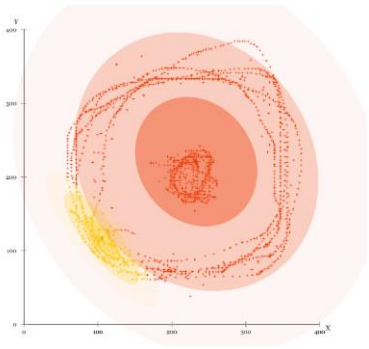
## 2.2. Algoritam EM

Algoritam očekivanja-maksimizacije (engl. *Expectation–maximization algorithm*) ili kraće EM-algoritam je u jednom obliku ekstenzija algoritma *k-means*, gdje su grupe modelirane normalnom (Gaussovom) raspodjelom, dok kod *k-meansa,* grupe su modelirane euklidskom udaljenošću od središta. Zbog toga, algoritmu je uz središte potrebna i kovarijanca koja opisuje elipsu grupe. Uzorci se dodjeljuju grupama temeljem vjerojatnosti pripadnosti pojedinoj grupi.

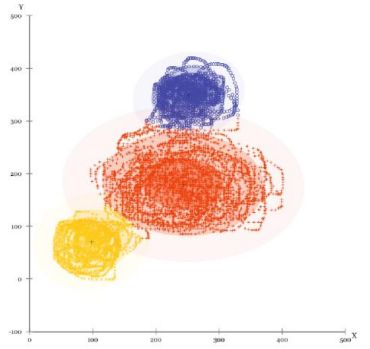
Postupak EM-algoritma:

1. postavljanje početnih vrijednosti parametara središta, kovarijance i veličine svake grupe;
2. za svaki uzorak u skupu podataka se izračunava vjerojatnost pripadnosti grupi;
3. normalizacija vjerojatnosti pripadnosti grupe preko skupa svih grupa (ukupna vjerojatnost da uzorak pripada nekoj grupi iznosit će jedan);
4. podešavanje vrijednosti parametara svake grupe temeljem vjerojatnosti da uzorak pripada grupi;
5. ponavljaju se koraci 1 do 4 sve dok ne nastupi konvergencija k rješenju;

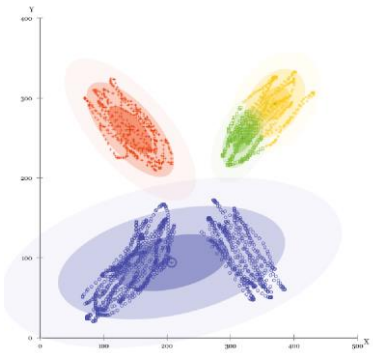
Poput *k-meansa*, algoritam EM će sigurno konvergirati, ali taj optimum ne mora biti globalni.



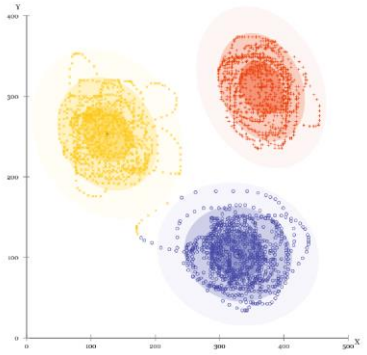
Primjer 2.2.1



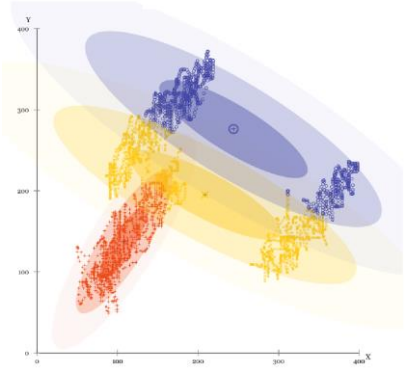
Primjer 2.2.2



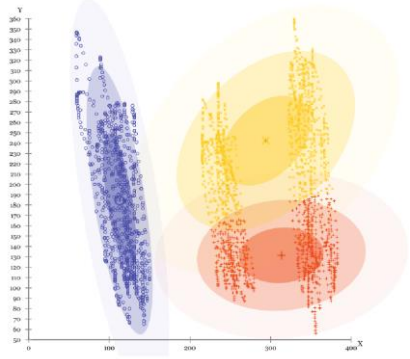
Primjer 2.2.3



Primjer 2.2.4



Primjer 2.2.5



Primjer 2.2.6

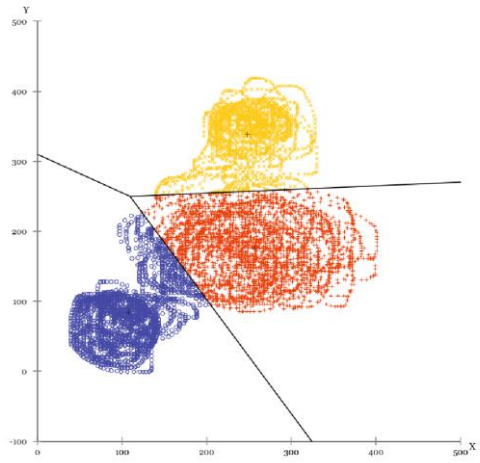
Slika 2.4 Primjeri *EM* algoritma

Na primjerima Primjer 2.2.1, Primjer 2.2.5 i Primjer 2.2.6 na Slika 2.4 Primjeri *EM* algoritma vidi se sličan problem kao i kod algoritma *k-means*, gdje vizualno očite grupe koje nisu eliptično oblikovane su stavljene u istu grupu.

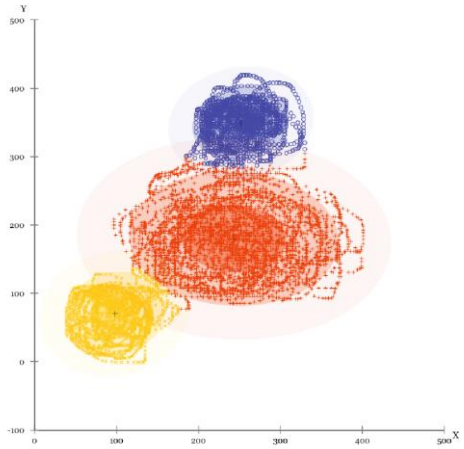
Kod Primjer 2.2.3 vidimo da je EM napravio puno lošiji posao od *k-meansa*, gdje je četiri jasne grupe loše grupirao.

Međutim na Slika 2.5 Usporedba algoritama *k-means* i *EM* je pokazan slučaj u kojem EM-algoritam daje puno bolje rješenje naspram algoritma *k-means*.

Pošto EM-algoritam radi temeljem vjerojatnosti i normalne raspodjele, EM granične slučajeve puno pravilnije i točnije grupira nego *k-means* (sSlika 2.5 Usporedba algoritama *k-means* i *EM*).




Primjer 2.2.7 *k-means*




Primjer 2.2.8 *EM*

Slika 2.5 Usporedba algoritama *k-means* i *EM*

# 3. Grupa algoritama grupiranja zasnovana na hijerarhiji

Hijerarhijska grupa algoritama pokušava povezati uzorke tako da izgradi hijerarhiju grupa od uzoraka [3].

Postoje dva tipa hijerarhijskog grupiranja [4]:

- aglomerativno grupiranje, ili grupiranje odozdo prema gore, gdje je inicijalno svaki uzorak smješten u svoju grupu te se dvije najbliže grupe spajaju u veću grupu i tako se kreće prema gore u hijerarhiji;
- razdvajajuće (engl. *divisive*) grupiranje, ili grupiranje odozgo prema dolje, gdje se inicijalno svi uzorci nalaze u jednoj grupi te se grupa/e rekurzivno razdvajaju na manje grupe i tako se spušta po hijerarhiji.

U ELKI-ju je fokus na implementacije aglomerativnog grupiranja, preciznije na algoritam aglomerativnog grupiranja i algoritam BIRCH.

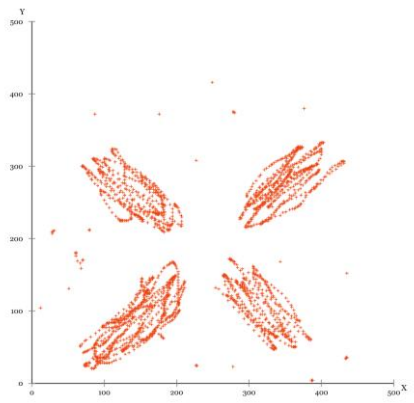
## 3.1. Aglomerativno grupiranje

Algoritam aglomerativnog grupiranja radi grupiranje tako da svaki uzorak započinje sa svojom grupom te se te grupe sukcesivno spajaju [6].

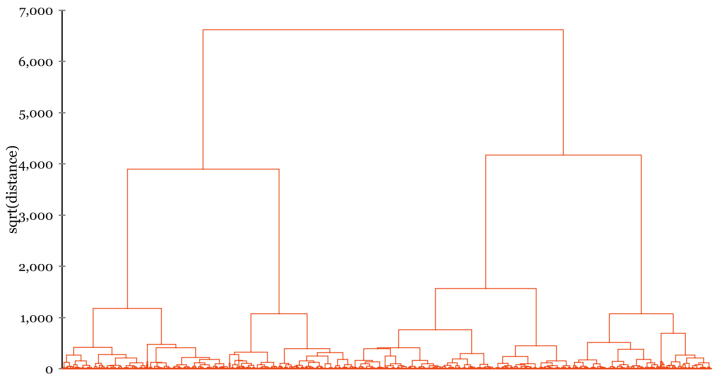
Postupak aglomerativnog grupiranja [7]:

1. svaki uzorak se označava kao individualna grupa;
2. uzimaju se dva uzorka određena temeljem udaljenosti i kriterija spajanja te se spajaju u jednu grupu;
3. korak 2 se ponavlja dok ne preostane jedna grupa;
4. grupe se ekstrahiraju.

Jedan način vizualizacije redoslijeda kojim je obavljeno grupiranje je dendrogram. Dendrogram je dijagram koji predstavlja drvo spajanja pojedinih podgrupa koje su odabrane temeljem kriterija spajanja. Visina dendrograma, tj. ordinata označava koliko je jedna podgrupa slična drugoj (što je veća visina to je jedna podgrupa manje sličnija drugoj), Slika 3.1 Dendrogram za Primjer 3.1.1.




Primjer 3.1.1



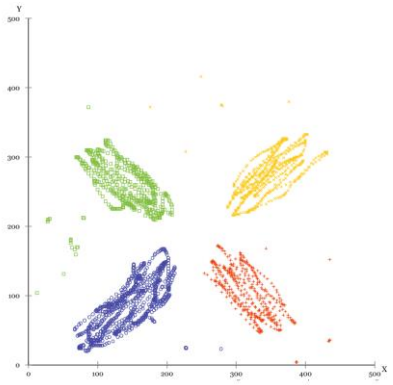

Slika 3.1 Dendrogram za Primjer 3.1.1

Slika 3.2 Primjer dendrograma unutar platforme ELKI

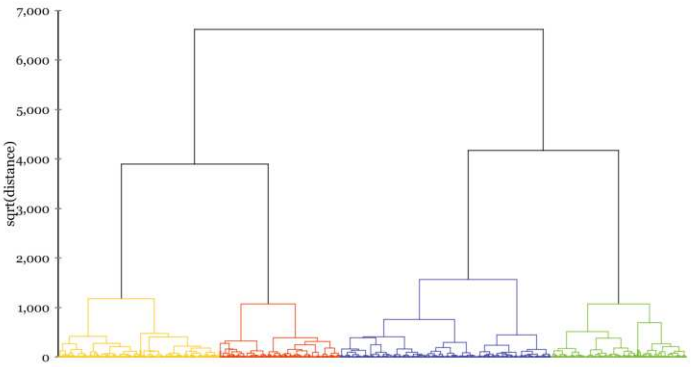
Zadnji korak je ekstrakcija grupa iz dendrograma. Postoji više načina kako izvući grupe iz dendrograma, jedan od načina je podrezivanje dendrograma na određenoj visini i sve podgrupe koje se nalaze ispod te visine postaju ciljne grupe. Također, postoji i podrezivanje dendrograma na točan broj grupa.

Vizualnom analizom dendrograma se može odrediti optimalan broj grupa tako da se dendrogram odreže tamo gdje je veliki skok udaljenosti, na Slika 3.1 Dendrogram za Primjer 3.1.1. može se vidjeti veliki skok udaljenosti na visini od 2000.

Dendogram sa Slika 3.1 Dendrogram za Primjer 3.1.1 odrezan na četiri grupe daje rezultat prikazan na Slika 3.4 Grupirani dendrogram.



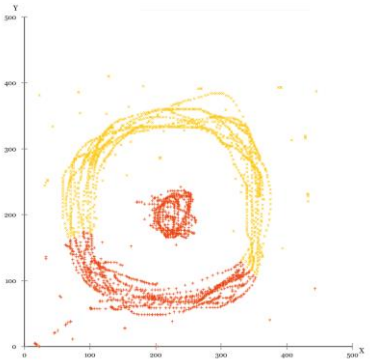
Primjer 3.1.2



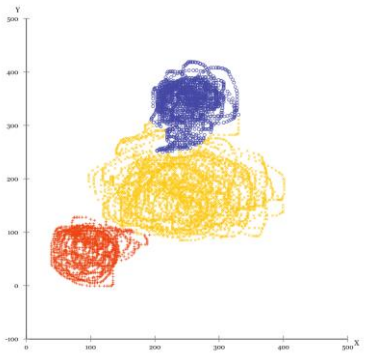

Slika 3.3 Grupirani dendrogram za Primjer 3.1.2

Slika 3.4 Grupirani dendrogram

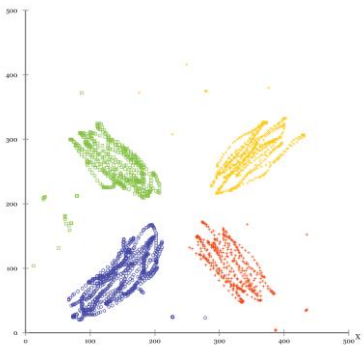
Primjenom sličnog načina određivanja grupa za ostale primjere dobiva se Slika 3.5 Primjeri aglomerativnog grupiranja.



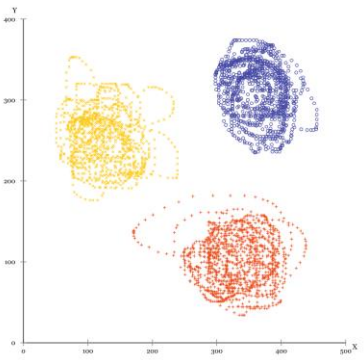
Primjer 3.1.3



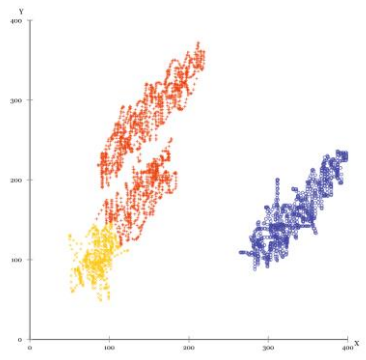
Primjer 3.1.4



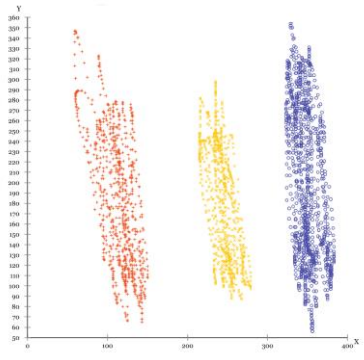
Primjer 3.1.5



Primjer 3.1.6



Primjer 3.1.7



Primjer 3.1.8

Slika 3.5 Primjeri aglomerativnog grupiranja

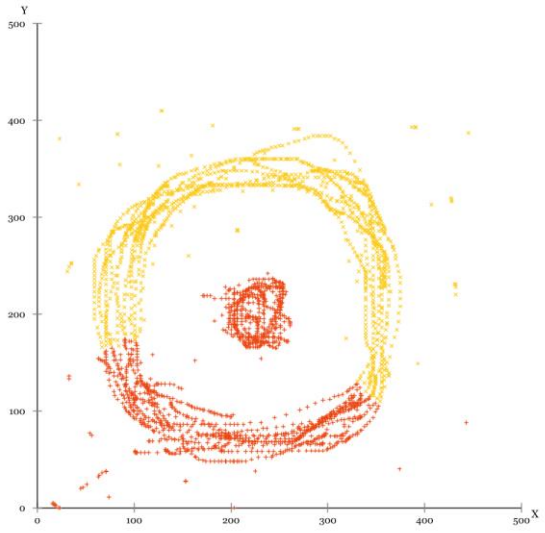
Aglomerativno grupiranje jako dobro rješava većinu slučajeva, što se može vidjeti skoro na svim primjerima osim na Primjer 3.1.3, gdje ima sličnu manu kao većina ostalih algoritma grupiranja jer jako teško rješava problem grupe unutar grupe te na primjeru 3.1.7 gdje su grupe dovoljno blizu da pojedini uzorci tih grupa se mogu povezati u istu grupu makar se vizualno nalaze u različitim grupama.

Složenost algoritma je O($n^2 \log(n)$) što je značajna mana aglomerativnog grupiranja, naročito na velikom skupu podataka [8].

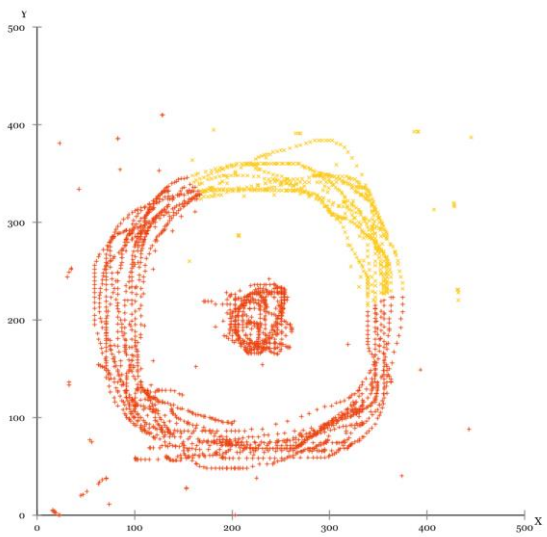
Za sve primjere na Slika 3.5 Primjeri aglomerativnog grupiranja korišten je kriterij povezivanja *Ward*, međutim postoje i drugi kriteriji koji mogu davati značajno drugačije rezultate.

Često korišteni kriteriji povezivanja:

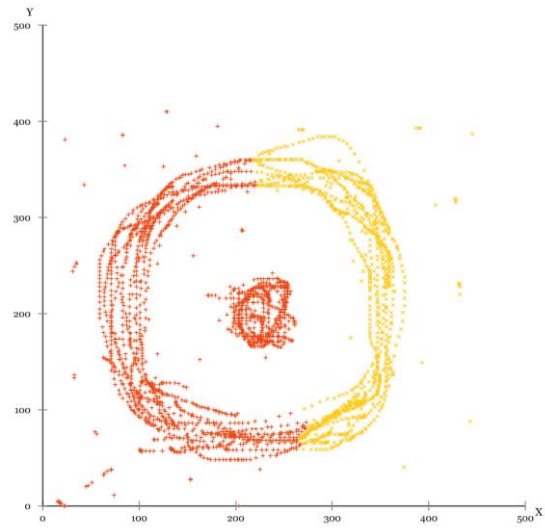
- metoda *Ward* – minimizira varijancu unutar svih grupa [12], Primjer 3.1.9;
- kompletna veza (engl. *complete linkage*) – minimizira maksimalnu udaljenost između dva para grupa [12], Primjer 3.1.11 Kompletna ;
- prosječna veza (engl. *average linkage*) – minimizira srednju udaljenost između svih parova grupa [12], Primjer 3.1.10 Prosječna ;
- pojedinačna veza (engl. *single linkage*) – minimizira udaljenost između dva najbliža para grupa [12], Primjer 3.1.12.



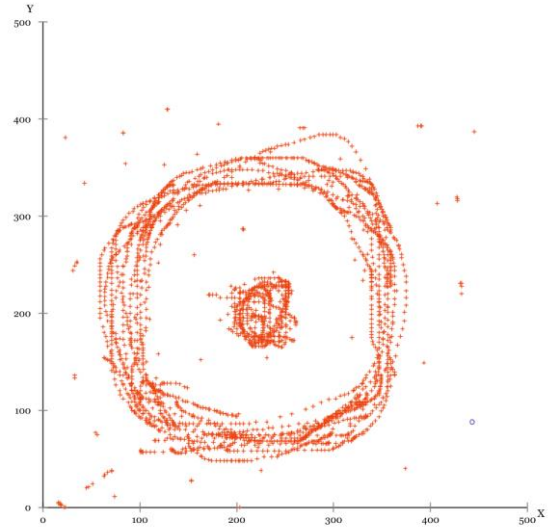

Primjer 3.1.9 *Ward* veza




Primjer 3.1.10 Prosječna veza




Primjer 3.1.11 Kompletna veza




Primjer 3.1.12 Pojedinačna veza

## 3.2. Algoritam BIRCH

Algoritam pod nazivom BIRCH (balansiran, iterativno smanjujući i grupiran korištenjem hijerarhija, engl. *balanced iterative reducing and clustering using hierarchies algorithm*) je algoritam koji grupira tehnikom hijerarhija na naročito velikom podatkovnom skupu.

Veliki problem prijašnjih algoritama je grupiranje preko velikog skupa podataka, što je onaj kod kojeg podaci ne bi stali u memoriju. Algoritam BIRCH rješava ovaj problem tako da derivira najbolje moguće podgrupe na temelju dostupnog memorijskog prostora i gradi drvo svojstava grupiranja (CF-drvo, engl. *clustering feature* (CF) *tree*).

Drvo svojstava grupiranja sadrži komprimirane početne podatke, gdje svaki čvor sadrži dovoljno informacija da bi se obavilo grupiranje, a drugi algoritmi mogu obaviti to grupiranje [12].

Postupak algoritma BIRCH [9]:

1. gradnja CF-drveta;
2. micanje nenormalnih čvorova kako bi CF-drvo dalo još bolje rezultate;
3. korištenje drugih algoritma grupiranja na izgrađeno CF-drvo (npr. algoritma *k-means*), što može rezultirati još boljim CF-drvom.

Glavni korak u algoritmu BIRCH je korak 1, dok ostali koraci služe za optimizaciju rezultata i oni su neobavezni.

Složenost algoritma je O($n$) što je značajna prednost algoritma BIRCH u odnosu na ostale algoritme grupiranja [9], međutim algoritam ne daje kvalitetne rezultate na skupu podataka s velikom dimenzionalnosti.

# 4. Grupa algoritama grupiranja zasnovana na gustoći

U ovoj grupi algoritama grupiranja, grupe su definirane kao područja veće gustoće nego ostatak podataka [4].

Uzorci koji se nalaze u slabo gustim područjima se smatraju šumom.

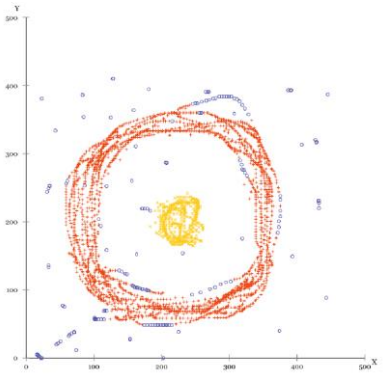
## 4.1. Algoritam DBSCAN

Prostorno grupiranje temeljeno na gustoći s primjenom šuma (engl. *density based spatial clustering of applications with noise*, DBSCAN) je algoritam grupiranja koji za dani broj točaka u prostoru, grupira one točke koje su usko spakirane. Isto tako označava i stršeće vrijednosti kao one koje se nalaze u područjima s malom gustoćom [10].

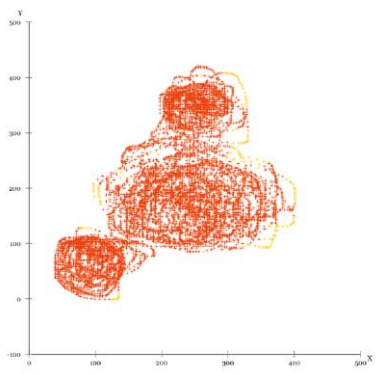
Postupak algoritma DBSCAN prikazan je u nastavku.

Algoritam prima dva parametra, $\varepsilon$ koji predstavlja najveću moguću udaljenost jednog uzorka do drugog, takvu da bi taj uzorak bio smatran kao susjedni uzorak prvom. Drugi parametar je najmanji broj uzoraka (engl. *minpts*) koji su potrebni za formaciju gustog područja. Algoritam radi tako da:

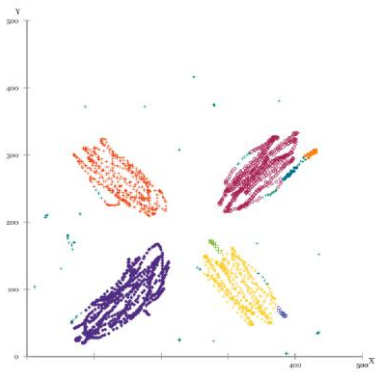
1. nasumično odabire točku koja još nije analizirana;
2. pronalazi susjede te točke uzimajući u obzir parametar $\varepsilon$ za odabir istih;
3. ako je broj pronađenih susjeda veći od minimalnog broj uzoraka za grupu, taj uzorak se smatra kao dio grupe inače se označava kao šum;
4. zatim prolazi kroz sve pronađene susjede i ako su oni dovoljno gusti, dodaju se u grupu ili se označavaju kao šum. Ovaj korak se ponavlja za svako $\varepsilon$-susjedstvo uzorka;
5. uzima se sljedeći neposjećeni uzorak i ponavljaju se koraci od 1 do 5 sve dok se ne posjete svi uzorci.



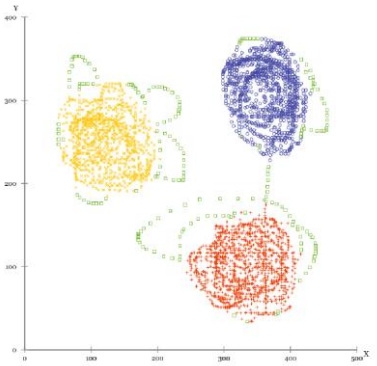
Primjer 4.1.1



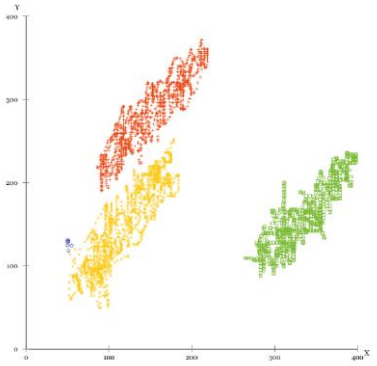
Primjer 4.1.2



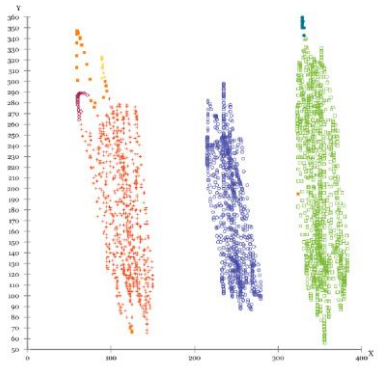
Primjer 4.1.3



Primjer 4.1.4



Primjer 4.1.5



Primjer 4.1.6

Slika 4.1 Primjeri DBSCAN algoritma

Na Primjer 4.1.1, Slika 4.1 Primjeri DBSCAN algoritma možemo vidjeti sposobnost algoritma da pronađe grupe koje se nalaze jedna unutar druge, dok god nisu međusobno povezane.

Isto tako vidimo na primjerima Primjer 4.1.3, Primjer 4.1.5 i Primjer 4.1.6 da je s lakoćom riješio čudno oblikovane grupe, dok su algoritmi zasnovani na centroidima i hijerarhiji ovdje griješili.

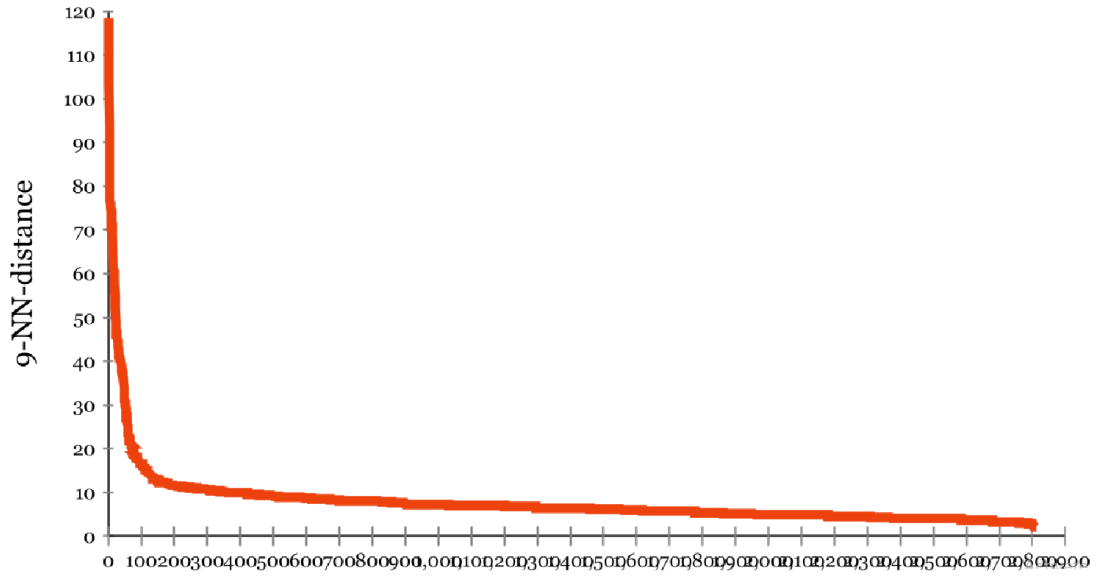
Primjeri Primjer 4.1.1 i Primjer 4.1.3 prikazuju sposobnost algoritma kod klasificiranja šuma, pod uvjetom da su vrijednosti šuma dovoljno dobro izolirane. Na Primjer 4.1.4 vidi se mana algoritma gdje jako teško rješava probleme kada grupe imaju različite gustoće ili se mijenja gustoća unutar jedne grupe zbog čega su granični slučajevi označeni kao šum.

Još jedna mana je prikazana na Primjer 4.1.2 gdje je algoritam sve uzorke svrstao pod jednu grupu jer su sve tri okom vidljive grupe međusobno gusto povezane.

Značajna prednost algoritma DBSCAN u odnosu na druge algoritme grupiranja je što mu nije potrebno odrediti parametar koji određuje količinu grupa, ali cijena toga je što se moraju odrediti druga dva parametara koja označavaju gustoću koja određuje grupe.

Parametar koji označava najmanji broj točaka unutar grupe (engl. *minpts*) se može odrediti temeljem broja dimenzija skupa podataka, u pravilu najmanji broj točaka se može postaviti na vrijednost dva puta veću od dimenzija skupa podataka. Također, *minpts* = 1 nema smisla jer bi onda svaka točka pripadala svojoj grupi, dok će *minpts* = 2 davati iste rezultate kao hijerarhijsko grupiranje, s dendrogramom odrezanim na vrijednosti $\varepsilon$.

Kod određivanja vrijednosti $\varepsilon$ mora se pripaziti, jer za male vrijednosti $\varepsilon$ većina podataka neće biti svrstana u grupe, dok za preveliku vrijednost $\varepsilon$ većina podataka će biti u istoj grupi. Jedan od načina kako odrediti $\varepsilon$ je preko grafa *k*-najbližih susjeda (engl. *k nearest neighbor graph)* s *k* postavljenim na (*minpts*-1*)*, gdje je vrijednost $\varepsilon$ jednaka poziciji lakta, a Slika 4.2. Graf najbližih susjeda za Primjer 4.1.12 prikazuje graf najbližih susjeda za Primjer 4.1.1 gdje je *k*=9 i željeni $\varepsilon$ između 10 i 20.



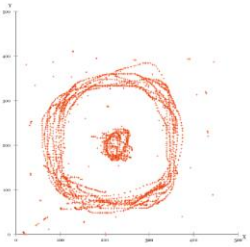

Slika 4.2. Graf najbližih susjeda za Primjer 4.1.1

## 4.2. Algoritam OPTICS

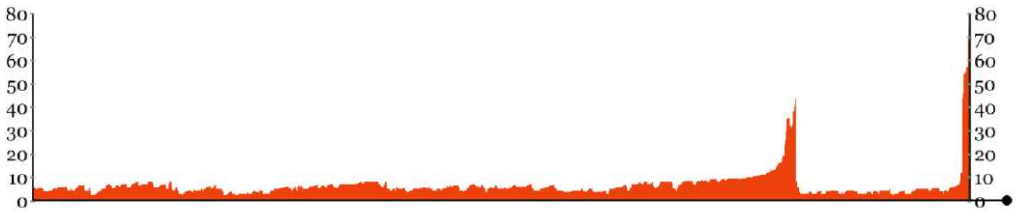
Algoritam određivanja točaka za identifikaciju strukture grupiranja (engl. *ordering points to identify the clustering structure*, OPTICS) je algoritam koji je jako sličan algoritmu DBSCAN, ali se obraća jednom od glavnog nedostatka DBSCAN-algoritma, a to je pronalazak značajnih grupa s različitom gustoćom [11]. OPTICS je na neki način generalizacija DBSCAN-a gdje umjesto da parametar $\varepsilon$ predstavlja jednu nezamjenjivu vrijednost, $\varepsilon$ predstavlja opseg vrijednosti [12].

OPTICS poput DBSCAN-a prima dva parametara, $\varepsilon$ i najmanji broj uzoraka za grupaciju (*minpts*). Parametar $\varepsilon$ i ovdje predstavlja maksimalnu udaljenost, ali ta udaljenost služi za brze izvođenje programa, jer algoritam neće ispitivati povezanost uzoraka dalje od te vrijednosti. Ta vrijednost teorijski nije potrebna i može biti postavljena na maksimalnu udaljenost unutar skupa podataka.

Algoritam OPTICS stvara graf pristupačnosti (engl. *reachability plot*) iz kojeg se mogu izvući grupe.



Primjer 4.2.1

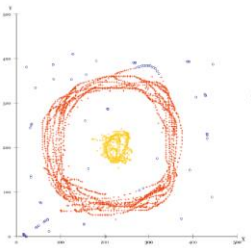



Slika 4.2 Graf pristupačnosti za Primjer 4.2.1

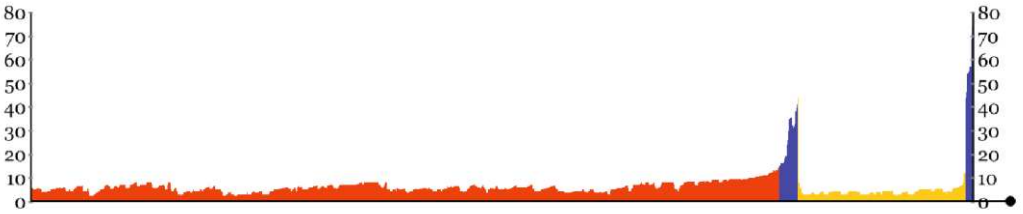
Na Slika 4.2 Graf pristupačnosti za Primjer 4.2.1 se može vidjeti izgled grafa pristupačnosti za Primjer 4.2.1, os apscise predstavlja uzorke u redoslijedu koji je odredio algoritam OPTICS, dok se na osi ordinate nalaze vrijednosti pristupačnosti za pojedini uzorak. Uzorci koji pripadaju nekoj grupi se nalaze u dolini grafa, dok se na brjegovima nalaze stršeće vrijednosti.

Kako bi izvukli grupe iz ovog grafa nakon vizualne analize, potrebno je odabrati vrijednost na osi ordinata koja će predstavljati vrijednost $\varepsilon$.

Za vrijednost $\varepsilon$=15, rezultat grupiranja je prikazan na Slika 4.3 Grupiran graf pristupačnosti s $\varepsilon$=15 (plava boja predstavlja šum, a crvena i žuta dvije grupe).



Primjer 4.2.2




Slika 4.3 Grupiran graf pristupačnosti s $\varepsilon$=15

# 5. Evaluacija performansa algoritama grupiranja

Postoje mnogi načini evaluacije kvalitete obavljenog grupiranja. Većina metoda evaluacije se sastoje od različitih kombinacija točnih pozitiva (TP, engl. *true positive*), točnih negativa (TP, engl. *true negative*), netočnih pozitiva (TP, engl. *false positive*) i netočnih negativa (TP, engl. *false negative*).

Unutar ELKI-ja evaluacija se provodi automatski nakon pokretanja analize. Evaluacija se izračunava temeljem samog skupa za učenje, uspoređivanjem dobivenih grupa s prije označenim ciljnim grupama.

Tablica 5.1 Evaluacijske metode unutar ELKI-ja prikazane su neke evaluacijske metode implementirane unutar ELKI-ja.

Tablica 5.1 Evaluacijske metode unutar ELKI-ja [13][3]

| Ime | Formula |
|---|---|
| Preciznost | $\frac{TP}{TP+FP}$ |
| Opoziv (engl. *recall*) | $\frac{TP}{TP+FN}$ |
| Jaccardov indikator | $\frac{TP}{TP+FP+FN}$ |
| F1-mjera | $\frac{2TP}{2TP+FP+FN}$ |
| Randov indeks | $\frac{TP+TN}{TP+FP+FN+TN}$ |
| Fowlkes-Mallowsov indeks | $\sqrt{\frac{TP}{TP+FP}\cdot\frac{TP}{TP+FN}}$ |

# 6. Provedba procesa grupiranja na skupu podataka

## 6.1. Opis skupa podataka

Skup podataka na kojem će se provesti analiza grupiranja je preuzet s repozitorija UCI Irvine machine learning. Podaci predstavljaju količinu učenja glavnog i njemu sličnih predmeta te uspješnost na ispitima glavnog i njemu sličnih predmeta. Podaci se sastoje od pet atributa, svi su numeričke vrijednosti [14]:

- STG – količina vremena ispitanika utrošena za učenje glavnog predmeta;
- SCG – količina repeticije materijala glavnog predmeta;
- STR – količina učenja predmeta sličnih glavnome;
- LPR – uspješnost ispitanika na ispitima sličnih predmeta;
- PEG – uspješnost ispitanika na ispitu glavnog predmeta.

Skup podataka na kojemu će provesti analiza se sastoji od 258 primjera.

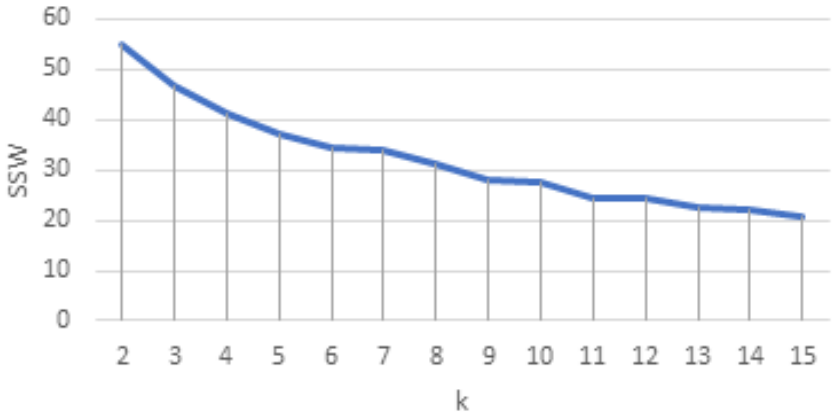
Cilj analize je grupirati ispitanike temeljem njihovog ukupnog znanja i evaluirati kvalitetu dobivenih grupa temeljem ručno određenih klasa.

Kod evaluacije algoritama bit će korištene sve mjere navedene u poglavlju 5.

## 6.2. Algoritam *K-means*

Prvi korak algoritma *k-means* je određivanje optimalnog broja grupa. Za određivanje parametra *k* radimo graf varijance s obzirom na parametar *k*, Slika 6.1 Graf varijance unutar grupe (SSW, engl. *sum of squares within clusters*) s obzirom na *k*. Platforma ELKI ne nudi izravan izračun ovoga grafa. Međutim, provedbom algoritma s različitim vrijednostima za *k* i računanjem ukupne varijance temeljem dobivenih rezultata, može se izgraditi dani graf.

Vrijednost parametra *k* određena je laktom grafa sa Slika 6.1 Graf varijance unutar grupe (SSW, engl. *sum of squares within clusters*) s obzirom na *k*. Lakat na slici nije očit, ali nešto nalik njemu može se vidjeti za *k*=6.



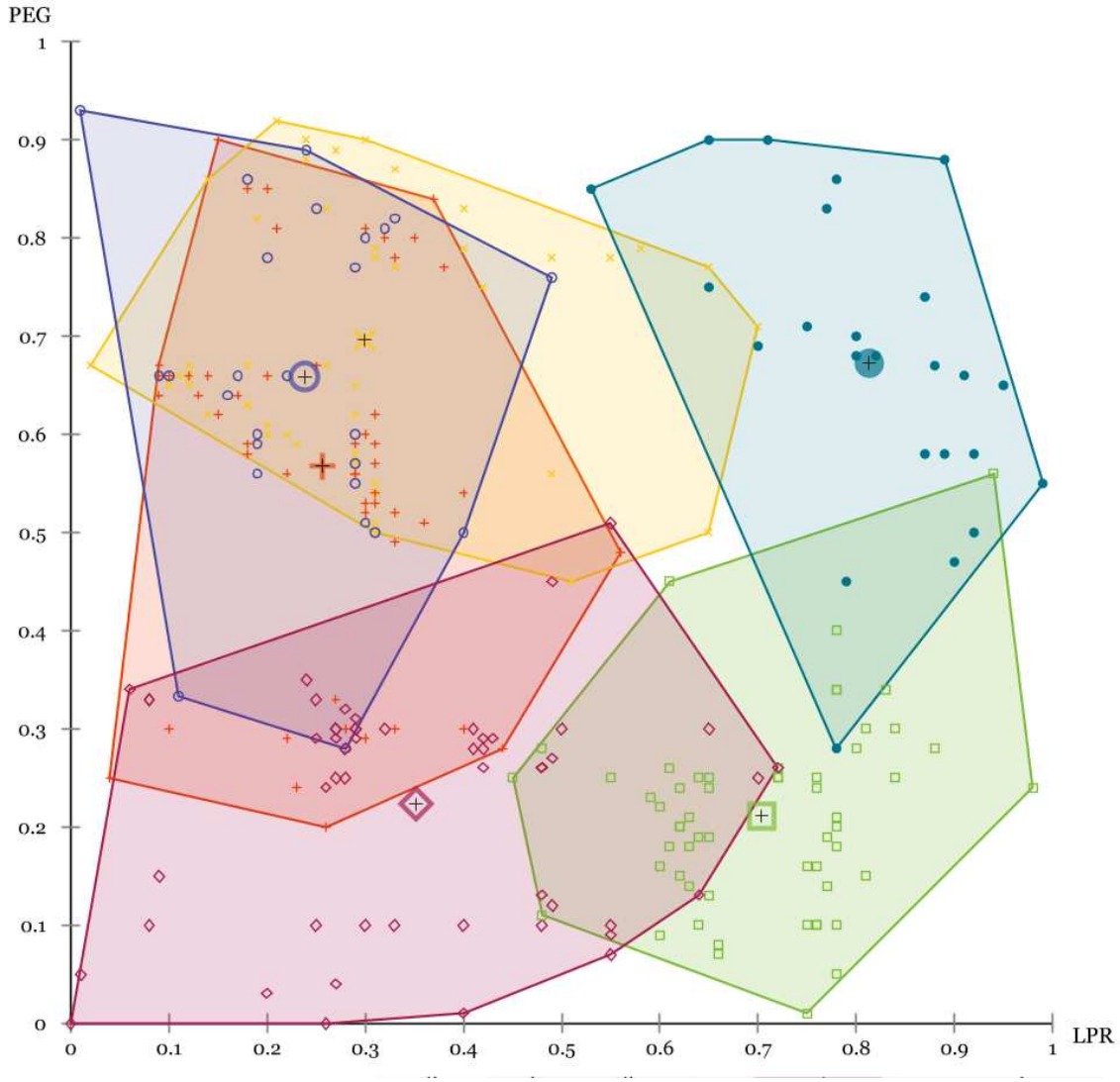

Slika 6.1 Graf varijance unutar grupe (SSW, engl. *sum of squares within clusters*) s obzirom na *k*

Za konkretnu implementaciju algoritma *k-means* unutar platforme ELKI, bit će korišten algoritam *KMeansSort*. Unos izračunatog parametra *k* zajedno s odabranim algoritmom, prikazano je na Slika 6.2 Početni parametri algoritma *k-means* . Parametri za inicijalizaciju početnih središta grupa postavljeni su na nasumični odabir vrijednosti, dok će za računanje udaljenosti biti korištena euklidska udaljenost.

| algorithm | clustering.kmeans.KMeansSort |
|---|---|
| kmeans.k | 6 |
| kmeans.initialization | Default: RandomlyChosenInitialMeans |
| kmeans.seed | Default: global random |
| algorithm.distancefunction | Default: minkowski.SquaredEuclideanDistanceFunction |
| kmeans.maxiter | Default: 0 |

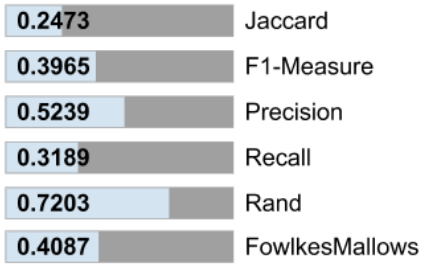
Slika 6.2 Početni parametri algoritma *k-means*

Slika 6.3 Prikaz dobivenih grupa algoritma *k-means*, grafom PEG-a (uspješnost na testu glavnog predmeta) i LPR-a (uspješnost na sličnom predmetu) prikazuje dobivene grupacije s obzirom na uspješnost učenika na glavnom ispitu i ispitu sličnog predmeta. Može se vidjeti da crvena grupa prikazuje studente koji su loše prošli na oba ispita, dok svijetlo plava prikazuje učenike koji su na oba ispita prošli dobro.




Slika 6.3 Prikaz dobivenih grupa algoritma *k-means*, grafom PEG-a (uspješnost na testu glavnog predmeta) i LPR-a (uspješnost na sličnom predmetu)

Evaluacija algoritma *k-means* putem platforme ELKI se obavlja automatski, te je prikazana na Slika 6.4 Evaluacijske mjere za algoritam *k-means* .




Slika 6.4 Evaluacijske mjere za algoritam *k-means*

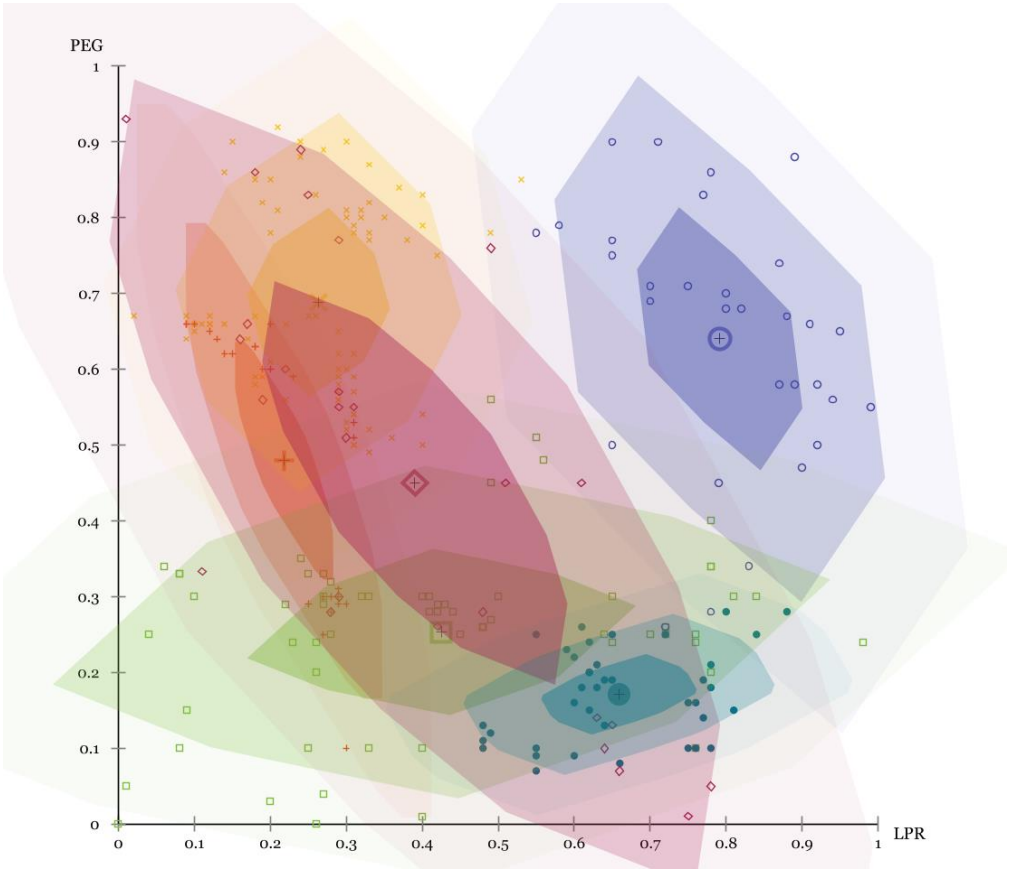
## 6.3. EM-algoritam

Početni parametar za broj grupa preuzet je iz algoritma *k-means* te su nasumično odabrane početne vrijednosti za središta grupa. Slika 6.5 Početni parametri EM-algoritma prikazuje početne parametre EM-algoritma unutar platforme ELKI.

| algorithm | clustering.em.EM |
|---|---|
| em.k | 6 |
| em.model | Default: MultivariateGaussianModelFactory |
| em.centers | Default: RandomlyChosenInitialMeans |
| kmeans.seed | Default: global random |
| em.delta | Default: 1.0E-7 |
| kmeans.maxiter | |
| em.map.prior | |

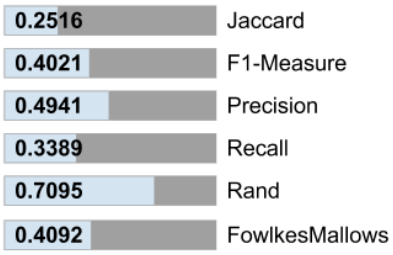
Slika 6.5 Početni parametri EM-algoritma

Na Slika 6.6 Prikaz dobivenih grupa EM-algoritma su vidljive grupe koje je odabrao EM-algoritam, mogu se vidjeti neke sličnosti s algoritmom *k-means*, poput grupe označene tamno plavom bojom koja predstavlja dobre rezultate na oba ispita.




Slika 6.6 Prikaz dobivenih grupa EM-algoritma

Također, Slika 6.7 Evaluacijske mjere EM-algoritma se vidi da EM-algoritam ima jako slične evaluacijske vrijednosti kao algoritam *k-means*.




Slika 6.7 Evaluacijske mjere EM-algoritma

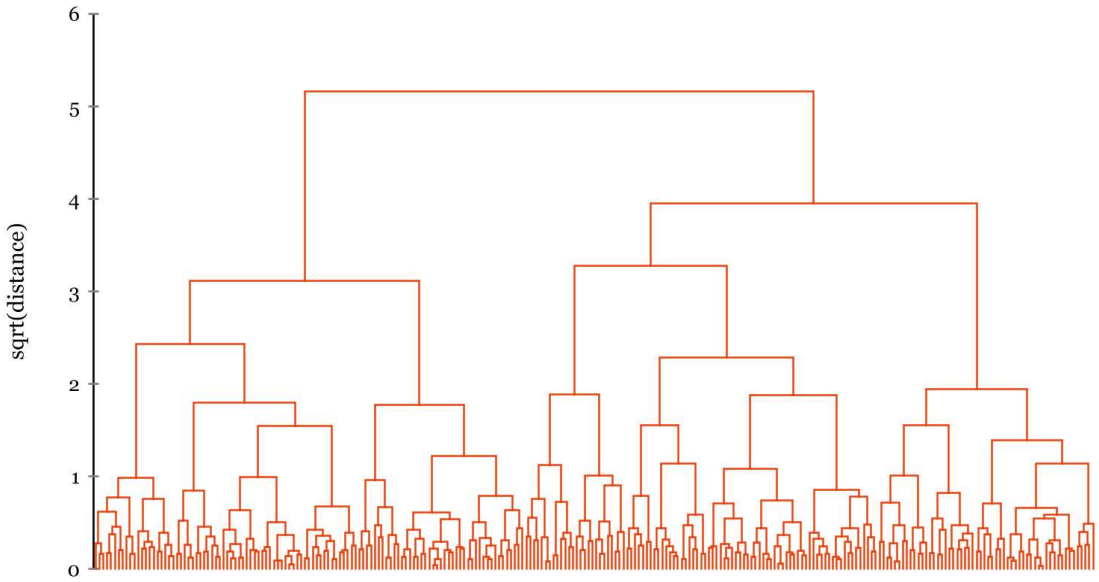
## 6.4. Aglomerativno grupiranje

Unutar ELKI-ja za obavljanje aglomerativnog grupiranja koristiti će se algoritam AGNES, s Wardovom vezom, Slika 6.8 Početni parametri aglomerativnog grupiranja.

| | |
|---|---|
| algorithm | clustering.hierarchical.AGNES |
| algorithm.distancefunction | Default: minkowski.SquaredEuclideanDistanceFunction |
| hierarchical.linkage | Default: WardLinkage |

Slika 6.8 Početni parametri aglomerativnog grupiranja

Dobiveni dendrogram prikazan je Slika 6.9 Dendrogram aglomerativnog grupiranja, vizualnom inspekcijom može se vidjeti veliki skok u udaljenosti na vrijednostima iza tri, što bi dani skup podataka razdvojilo na pet grupa.



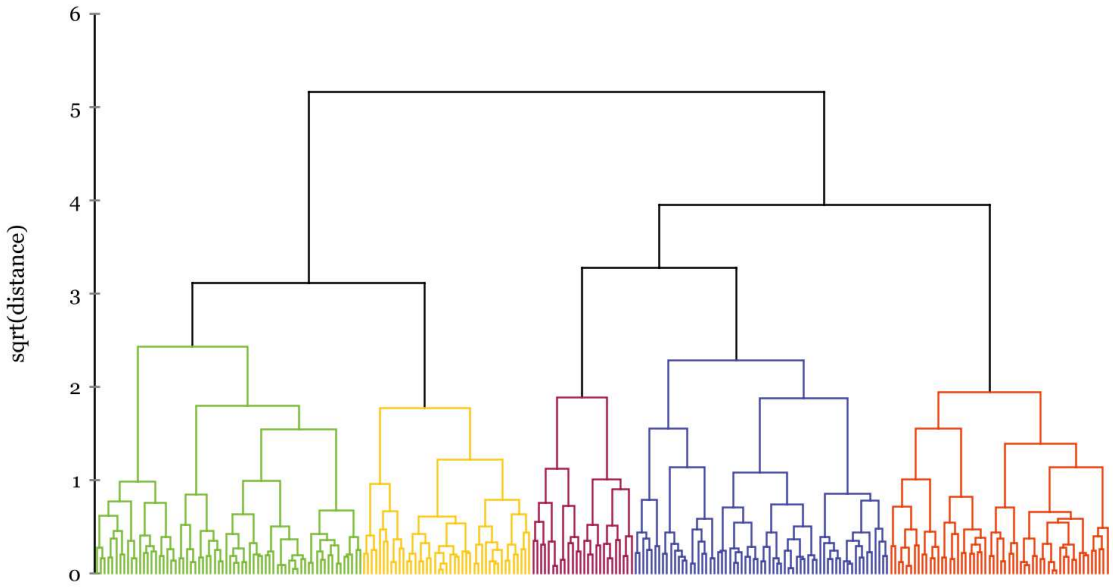

Slika 6.9 Dendrogram aglomerativnog grupiranja

Ekstrakcija grupa putem platforme ELKI, može se obaviti tako da se pokrene ekstrakcijski algoritam temeljem broja grupa (u ovom slučaju 5) i nakon toga algoritam na kojemu će se odraditi ekstrakcija, Slika 6.10 Početni parametri ekstrakcije.

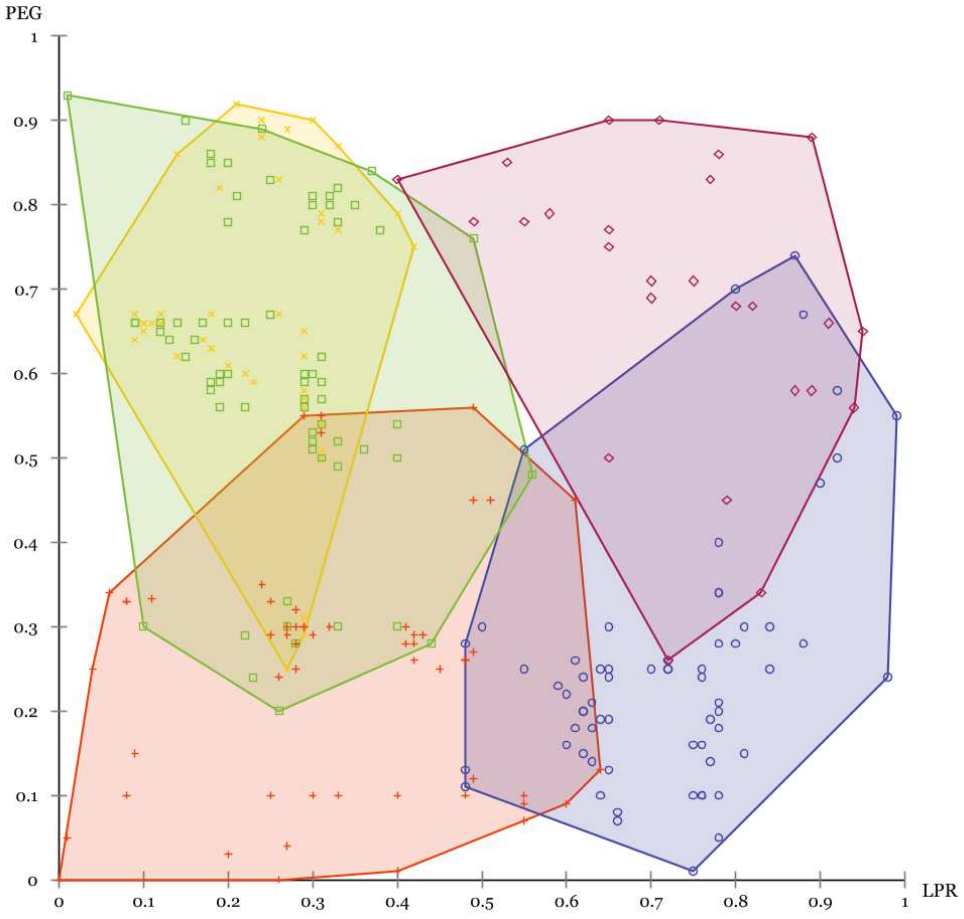
| | |
|---|---|
| algorithm | clustering.hierarchical.extraction.CutDendrogramByNumberOfClusters |
| algorithm | AGNES |
| algorithm.distancefunction | Default: minkowski.SquaredEuclideanDistanceFunction |
| hierarchical.linkage | Default: WardLinkage |
| hierarchical.hierarchy | Default: false |
| hierarchical.minclusters | 5 |

Slika 6.10 Početni parametri ekstrakcije

Rezultat ekstrakcije temeljem pet grupa je dendrogram prikazan na Slika 6.11 Dendrogram aglomerativnog grupiranja nakon ekstrakcije. i vizualni prikaz grupacija sa Slika 6.12 Prikaz dobivenih grupa aglomerativnog grupiranja.

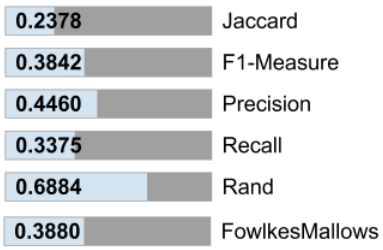



Slika 6.11 Dendrogram aglomerativnog grupiranja nakon ekstrakcije




Slika 6.12 Prikaz dobivenih grupa aglomerativnog grupiranja

Provedbom evaluacije dobivena je Slika 6.13 Evaluacijske mjere EM-algoritma, koja prikazuje da je algoritam aglomerativnog grupiranja u svim mjerama lošiji u odnosu na algoritme centroidnog pristupa.




Slika 6.13 Evaluacijske mjere EM-algoritma

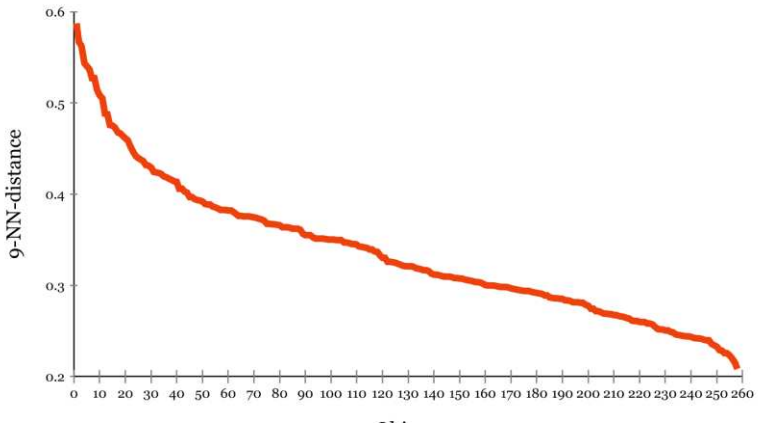
## 6.5. Algoritmi DBSCAN i OPTICS

Prvi korak, kao i kod svakog navedenog algoritma je određivanje početnih parametara, u ovom slučaju to su parametri $\varepsilon$ i *minpts*. Parametar *minpts* temeljem pravila navedenog kod opisa samog algoritma, će biti postavljen na dva puta veću vrijednost od broja dimenzija (atributa), što je u ovom slučaju 10. Parametar $\varepsilon$ će biti određen putem metode lakta grafa KNN s $k = 9$ (*minpts* – 1), Slika 6.14 Početni parametri KNN algoritma.

| algorithm | KNNDistancesSampler |
|---|---|
| algorithm.distancefunction | Default: minkowski.EuclideanDistanceFunction |
| knndistanceorder.k | 9 |

Slika 6.14 Početni parametri KNN algoritma

Dobiveni graf prikazan je na Slika 6.15 KNN graf za *k=9*. Najbliži oblik lakta je vidljiv oko vrijednosti 0.4, čime je određen parametar $\varepsilon$.




Slika 6.15 KNN graf za *k=9*

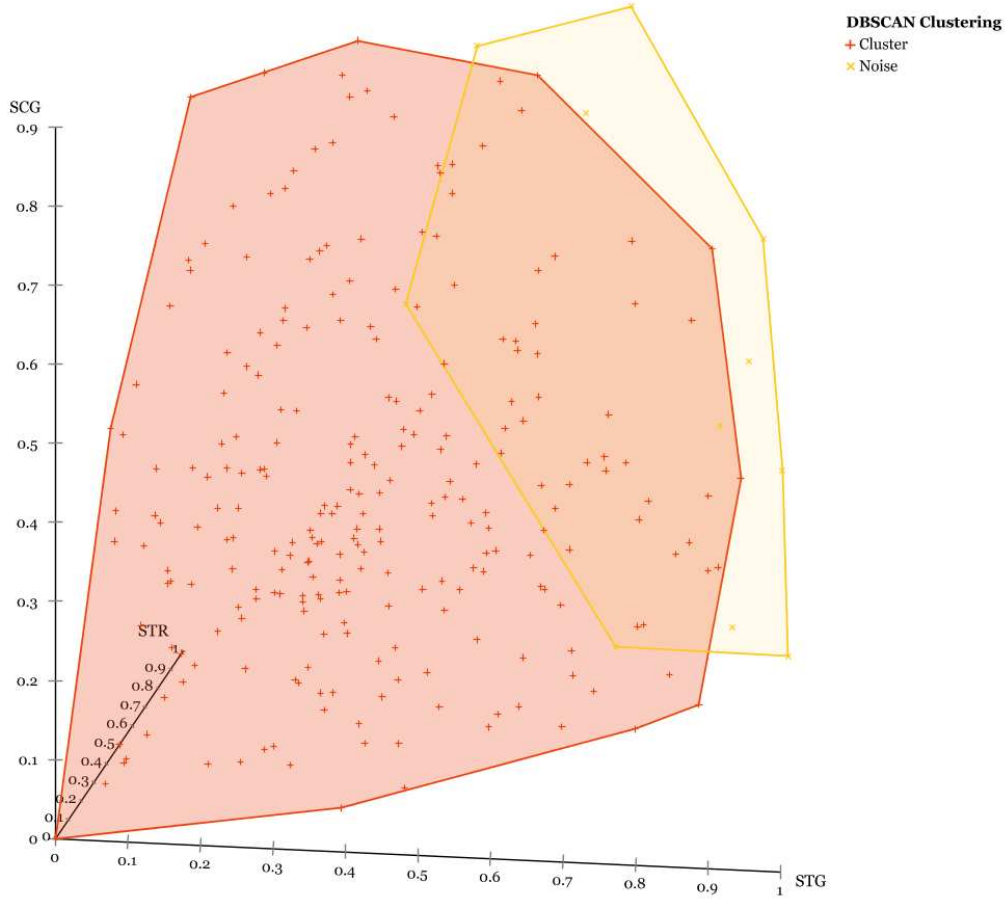
Slika 6.16 Početni parametri DBSCAN algoritma su prikazani izračunati početni parametri algoritma DBSCAN, postavljeni unutar ELKI platforme.

| algorithm | clustering.DBSCAN |
|---|---|
| algorithm.distancefunction | Default: minkowski.EuclideanDistanceFunction |
| dbscan.epsilon | 0.4 |
| dbscan.minpts | 10 |

Slika 6.16 Početni parametri DBSCAN algoritma

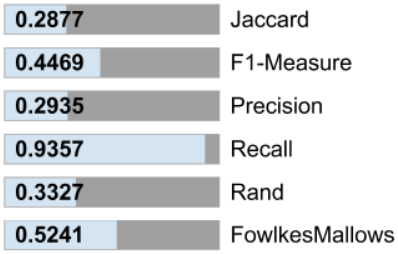
Rezultat grupiranja je prikazan Slika 6.17 Grupe dobivene DBSCAN algoritmom, gdje se vidi je algoritam napravio jako loš posao i pronašao je samo jednu grupu (crveno) a ostatak

je označio šumom (žuto). Ako se parametar $\varepsilon$ poveća trenutni šum će postati dio grupe. Međutim, ako se parametar $\varepsilon$ smanji, grupa će se početi smanjivat i više uzoraka će biti označeno kao šum. Razlog ovoga je podjednaka gustoća svih podataka.




Slika 6.17 Grupe dobivene DBSCAN algoritmom

U evaluaciji algoritma (Slika 6.18), se vidi jako velika vrijednost *recall* mjere, kojoj je uzrok pronalazak samo jedne grupe.




Slika 6.18 Evaluacijske mjere DBSCAN algoritma

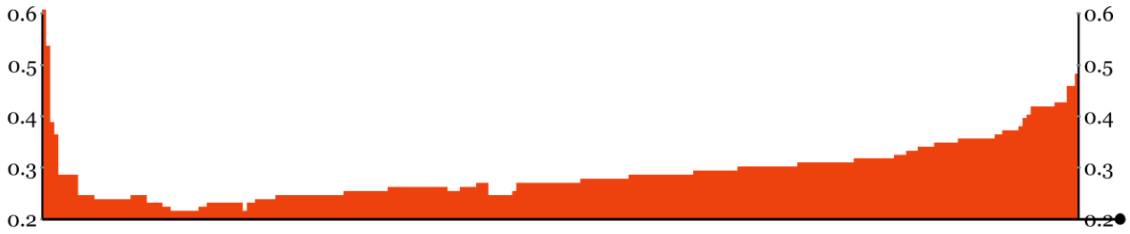
Za algoritam OPTICS su na sličan način kao i kod algoritma DBSCAN određeni početni parametri, s time da vrijednost $\varepsilon$ neće biti postavljena, što znači da će se pretraga napraviti

kroz sve moguće udaljenosti. Početni parametri unutar EKLI-ja su prikazani Slika 6.19 Početni parametri algoritma OPTICS .

| algorithm | clustering.optics.OPTICSHeap |
|---|---|
| algorithm.distancefunction | Default: minkowski.EuclideanDistanceFunction |
| optics.epsilon | |
| optics.minpts | 10 |

Slika 6.19 Početni parametri algoritma OPTICS

Dobiveni graf pristupačnosti prikazan je Slika 6.20 Graf gdje se vidi da je većina podataka slične gustoće, zbog čega je algoritam DBSCAN davao loše rezultate. Može se vidjeti da za bilo koji $\varepsilon$, rezultat će uvijek biti jedna grupa i šum. Nema više brjegova i dolova koji bi označavali različite grupe, zbog čega će ekstrakcija grupa dati isti rezultat kao i algoritam DBSCAN.

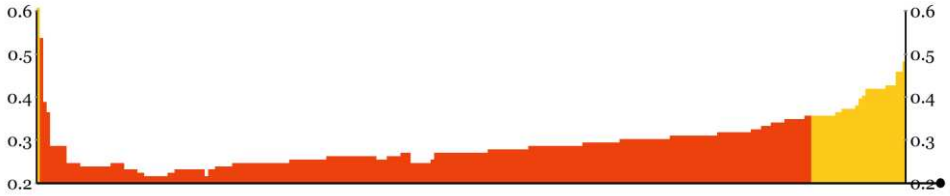



Slika 6.20 Graf pristupačnosti

Primjer ekstrakcije grupa za algoritam OPTICS unutar ELKI-ja, može se izvesti pozivanjem samog algoritma OPTICS i zatim algoritma DBSCAN sa određenim $\varepsilon$, Slika 6.21 Parametri za ekstrakciju grupa algoritma OPTICS. Odijeljeni graf pristupačnosti je prikazan na Slika 6.22 Graf pristupačnosti s određenim grupama gdje je rezultat jedna grupa i šum.

| algorithm | clustering.optics.OPTICSHeap,clustering.DBSCAN |
|---|---|
| algorithm.distancefunction | Default: minkowski.EuclideanDistanceFunction |
| optics.epsilon | |
| optics.minpts | 10 |
| algorithm.distancefunction | Default: minkowski.EuclideanDistanceFunction |
| dbscan.epsilon | 0.35 |
| dbscan.minpts | 10 |

Slika 6.21 Parametri za ekstrakciju grupa algoritma OPTICS




Slika 6.22 Graf pristupačnosti s određenim grupama

## 6.6. Pregled evaluacije algoritama grupiranja

Tablica 6.1 Tablica vrijednosti evaluacijskih mjera pojedinih algoritama prikazane su evaluacijske mjere pojedinog algoritma. Vidljivo je da *k-means*, EM-algoritam i aglomerativno grupiranje imaju slične vrijednosti. Međutim, uzimajući u obzir sve vrijednosti, najbolji algoritam za ovaj skup podatka bi bio algoritam *k-means*. Također treba uzeti u obzir da ovdje algoritmi temeljeni na gustoći (DBSCAN i OPTICS) ne mogu biti korišteni, jer sve podatke grupiraju u jednu grupu, čime sama analiza grupiranja nema smisla, jer ona zahtijeva minimalno dvije grupe.

Tablica 6.1 Tablica vrijednosti evaluacijskih mjera pojedinih algoritama

| **Evaluacijska mjera** | ***k-means*** | **EM-algoritam** | **Aglomerativno grupiranje** | **DBSCAN i OPTICS** |
|---|---|---|---|---|
| Preciznost | 0,5239 | 0,4941 | 0,4460 | 0,2935 |
| Opoziv | 0,3189 | 0,3389 | 0,3375 | 0,9357 |
| Jaccardov indikator | 0,2473 | 0,2516 | 0,2378 | 0,2877 |
| F1-mjera | 0,3965 | 0,4021 | 0,3842 | 0,4469 |
| Randov indeks | 0,7203 | 0,7095 | 0,6884 | 0,3327 |
| Fowlkes-Mallowsov indeks | 0,4087 | 0,4092 | 0,3880 | 0,5241 |

# Zaključak

Analiza grupiranja podataka je veliko područje istraživanja te postoje mnogi načini rješavanja tog zadatka s pregršt raznih modela, tehnika i algoritama koji pokazuju svoje prednosti u nekim slučajevima, a i svoje nedostatke u usporedbi s drugim modelima.

Cilj ovog rada je bio prikazati neke osnovne modele i algoritme grupiranja podataka. Na jednostavnim primjerima pokazane su prednosti, nedostaci i sami način rada pojedinog modela i njegovih reprezentativnih algoritama.

Ne postoji univerzalni model koji se može iskoristiti za sve slučajeve. Svaki slučaj zahtijeva drugačiji način sagledavanja problema te, često, korištenje jednog algoritma neće biti dovoljno. Zbog toga ključno je poznavati način rada pojedinog algoritma. Takvim pristupom ćemo pravilno odabrati i primijeniti pojedini model u svakoj pojedinačnoj situaciji.

Analiza grupiranja je, kao i sama znanstvena metoda, mukotrpan ali bitan proces koja zahtjeva znanje o samom području koje se analizira, te konstantno podešavanje za postizanje boljeg rezultata.

# Literatura


[1] Wikipedia. ELKI. Poveznica: https://en.wikipedia.org/wiki/ELKI/; pristupljeno 20. svibnja 2020.

[2] Erich Schubert and Arthur Zimek. ELKI: *A large open-source library for data analysis*. ELKI *Release* 0.7.5 "Heidelberg". CoRR arXiv 1902.03616. Poveznica: https://elki-project.github.io/; pristupljeno 20. svibnja 2020.

[3] Dongkuan Xu, Yingjie Tian, *A Comprehensive Survey of Clustering Algorithms*, Ann. Data. Sci. (2015) 2(2):165–193, DOI: 10.1007/s40745-015-0040-1. Poveznica: https://link.springer.com/content/pdf/10.1007/s40745-015-0040-1.pdf; pristupljeno 20. svibnja 2020.

[4] Wikipedia, *Cluster analysis*. Poveznica: https://en.wikipedia.org/wiki/Cluster_analysis/; pristupljeno 22. svibnja 2020.

[5] Wikipedia, *K-means clustering*. Poveznica: https://en.wikipedia.org/wiki/K-means_clustering/; pristupljeno 22. svibnja 2020.

[6] Wikipedia, *Hierarchical clustering*. Poveznica: https://en.wikipedia.org/wiki/Hierarchical_clustering; pristupljeno 25. svibnja 2020.

[7] Alboukadel Kassambara, *Agglomerative Hierarchical Clustering*. Poveznica: https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/; pristupljeno 25. svibnja 2020.

[8] Cory Maklin, *Hierarchical Agglomerative Clustering Algorithm Example In Python*. Poveznica: https://towardsdatascience.com/machine-learning-algorithms-part-12-hierarchical-agglomerative-clustering-example-in-python-1e18e0075019/; pristupljeno 25. svibnja 2020.

[9] Pawan Jain, *BIRCH Clustering Clearly Explained*. Poveznica: https://towardsdatascience.com/birch-clustering-clearly-explained-ffd75f07e5ed/; pristupljeno 26. svibnja 2020.

[10] Wikipedia, DBSCAN. Poveznica: https://en.wikipedia.org/wiki/DBSCAN/; pristupljeno 28. svibnja 2020.

[11] Wikipedia, OPTICS *algorithm*. Poveznica: https://en.wikipedia.org/wiki/OPTICS_algorithm/; pristupljeno 28. svibnja 2020.

[12] Scikit. *Clustering*. Poveznica: https://scikit-learn.org/stable/modules/clustering.html; pristupljeno 20.svibnja 2020.

[13] Wikipedia, *Sensitivity and specificity*. Poveznica: https://en.wikipedia.org/wiki/Sensitivity_and_specificity pristupljeno 6.lipnja 2020.

[14] H. T. Kahraman, Sagiroglu, S., Colak, I., *Developing intuitive knowledge classifier and modeling of users' domain dependent data in web,*
*Knowledge Based Systems*, vol. 37, pp. 283-295, 2013.

# Sažetak

Naslov: Analiza grupiranja podataka uporabom platforme ELKI

Sažetak:

Analiza grupiranja podataka je tehnika nenadziranog učenja kojoj je cilj ustanoviti postojanje dviju ili više grupa u podatcima koji nemaju ciljnu kategoriju. Postoje razni algoritmi, tj. grupe algoritama koji služe za provedbu grupiranja. U ovom radu, pomoću platforme ELKI, analizirane su grupe algoritama zasnovane na centroidima, hijerarhiji i gustoći. Pokazan je način rada, prednosti i mane pojedinih algoritama, radi kvalitetne primjene algoritama u različitim slučajevima jer ne postoji jedan algoritam koji se može iskoristiti za svaki problem.

Ključne riječi: Analiza grupiranja, centroid, hijerarhijsko grupiranje, gustoća, ELKI

# Summary

Title: Clustering analysis using ELKI platform

Summary:

Clustering analysis is an unsupervised learning method, which goal is to establish two or more clusters in data which originally do not have a goal category. There are many algorithms or groups of algorithms which have been developed for clustering analysis. In this paper using ELKI platform, groups based on centroid, hierarchical distance and density have been analyzed. The way that algorithm operates, its advantages and disadvantages are shown here, for the sake of understanding when to use which algorithm for better quality analysis, because there is not one algorithm which can be used for any problem.

Keywords: clustering analysis, centroid, hierarchical clustering, density, ELKI

# Skraćenice

- ELKI *Environment for DeveLoping KDD-Applications Supported by Index-Structures* Okolina za razvijanje KDD-aplikacija s osloncem na strukture indeksa
- EM *Expectation–maximization* Očekivanje-maksimizacija
- BIRCH *balanced iterative reducing and clustering using hierarchies algorithm* Balansiran iterativno smanjujući i grupiranje korištenjem hijerarhija algoritam
- CF *clustering feature* svojstava grupiranja
- DBSCAN *Density based spatial clustering of applications with noise* prostorno grupiranje temeljeno na gustoći s aplikacijom šuma
- OPTICS *Ordering points to identify the clustering structure* određivanje točaka za identifikaciju strukture grupiranja