Automatsko grupiranje podataka algoritmom k-means

Buršić, Siniša

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Preddiplomski studij

AUTOMATSKO GRUPIRANJE PODATAKA ALGORITMOM K-MEANS

Završni rad

Siniša Buršić

Osijek, 2019.

SADRŽAJ

1 UVOD

Od samih početaka ljudskog razvoja postojala je potreba za grupiranjem. Novo otkriveni materijali, pojmovi, fenomeni morali su se nekako razumjeti, objasniti. Sam tijek razumijevanja nekog novog pojma je ustvari usporedba tog pojma sa već poznatim pojmovima, a usporedba je na neki način grupiranje. Uspoređujući kamen sa komadom stakla i komadom plastike po njihovoj savitljivosti se može shvatiti kao grupiranje, gdje su dvije grupe savitljivi i nesavitljivi materijali i pitamo se kojoj grupi kamen pripada.

Može se reći da je grupiranje klasifikacija sličnih predmeta u različite grupe prema nekim obilježjima ili značajkama. Matematički, grupiranje predstavlja određivanje kojoj grupi pripada određeni podatak numeričkog tipa. Određuje se tako da podatci u grupama budu što kompaktniji (bliži jedni drugima). Kako bi se podaci grupirali u grupe potrebno je odrediti broj grupa. Nekada se broj grupa može logički odrediti (kao u gornjem primjeru savitljivosti materijala), a nekada se najbolji broj grupa određuje raznim matematičkim mjerenjima. U radu se prošlo kroz teoriju grupiranja, napravljeno je programsko rješenje pomoću kojeg se rješava problem grupiranja i određivanja broja grupa. Na kraju se analiziraju dobiveni rezultati programa.

U drugom poglavlju opisano je što predstavlja pojam grupiranja, odnosno grupiranje podataka, zadatci grupiranja podataka (problem pronalaska optimalnog broja grupa) i algoritmi grupiranja. Algoritam koji je detaljnije objašnjen je algoritam k-means. U poglavlju su navedene i neke primjene grupiranja, algoritma k-means. U Trećem poglavlju opisan je način rada ostvarenog programskog rješenja te način na koji se koristi. Četvrto poglavlje je eksperimentalna analiza ostvarena pomoću programskog rješenja. Njena svrha je prikaz i analiza automatskog grupiranja i vrednovanje particija uporabom tri kriterija.

2 GRUPIRANJE PODATAKA I ALGORITAM *K*-MEANS

Živimo u svijetu punom informacija i podataka. Svaki dan ljudi se susreću s različitim vrstama podataka koji dolaze iz svakakvih vrsta mjerenja, opažanja i pokusa. Podaci su nam potrebni za opis raznih pojava ili opažanja, kao primjerice, opis karakteristika živih bića, opis svojstava različitih procesa, prirodnih fenomena te za sažimanje rezultata određenih znanstvenih eksperimenata i slično. Isto tako podaci nam pružaju temelj za daljnju analizu, odluke i za razumijevanje svih vrsta objekata, pojava i problema. Zato je od velike važnosti da se taj veliki broj podataka može nekako klasificirati ili grupirati u skup kategorija ili grupa [1].

Podatci koji se grupiraju u iste grupe trebali bi imati slična svojstva na temelju nekih kriterija. Zapravo, kao jedna od najprimitivnijih aktivnosti ljudskih bića klasifikacija igra važnu i nezamjenjivu ulogu u povijesti ljudskog razvoja. Kako bi shvatili novi objekt ili razumjeli novu pojavu, ljudi uvijek pokušavaju identificirati opisne značajke tih objekata ili pojava te ih dalje usporediti sa značajkama ili svojstvima već poznatih na temelju njihovih sličnosti, odnosno različitosti. Primjerice, sve životinje su klasificirane u razne kategorije, neke od njih su kraljevstvo, tip, klasa, red koje su tako grupirane po svojim sličnostima. Samo imenovanje vrsta životinja je samo po sebi klasificiranje. Na slici 2.1 prikazane su grupirane smokve i jagode prema vrsti i boji.

Slika 2.1. Grupirano voće (http://www.ekocrep.eu/kategorija/marketing/page/2/)

Za grupiranje podataka tako postoje razni algoritmi koji pomažu pri grupiranju kada nije baš prirodno jasno koji podatak ide u koju grupu. Algoritmi u grupiranju podataka mogu se podijeliti na hijerarhijske algoritme i particijske algoritme. S hijerarhijskim algoritmima grupe se pronalaze upotrebom prijašnjih uspostavljenih grupa, tj. ondje gdje particijski algoritmi određuju sve grupe u jednom hodu. U particijskom grupiranju grupe su predstavljene središnjim vektorom, koji ne mora nužno biti član skupa podataka. Ako podatak pripada centroidu (najbliže tom centroidu) znači da je član grupe koja je predstavljena tim centroidom. Algoritam koji se detaljno obrađuje u ovom seminarskom radu, algoritam k-means jedan je od particijskih algoritama. Jednostavno rečeno, algoritam radi tako da na temelju zadanog broja grupa (k grupa) svakoj grupi se određuje njen centroid te se podatci pridružuju najbližem centroidu.

2.1 Grupiranje podataka

Kao što je već rečeno u prošlom poglavlju grupiranje podataka ima veliku važnost, jer su nam podaci bitni za analiziranje, opažanja, zaključivanje i slično. Za rješavanje problema grupiranja ne postoji neki specifičan algoritam koji može riješiti svaki problem. Postoje različiti algoritmi koji se značajno razlikuju, jer je različito razumijevanje pojma što je grupa te kako je najučinkovitije pronaći. Neke poznate definicije grupe su skupine s malim udaljenostima između članova, skupine s gustim područjima podataka ili intervala. Grupiranja se stoga može smatrati problemom višestrukog cilja. Važno je istražiti karakteristike problema da bi se odabrao najbolji algoritam grupiranja [1]. Problem grupiranja je iterativan proces otkrivanja najbolje optimizacije koji uključuje pokušavanje i greške.

Rečeno je da se pojam grupe može definirati na razne načine, ali zajedničko svakoj definiciji je da je to skupina podataka (podatkovnih objekata). Međutim, različiti istraživači upotrebljavaju različite modele grupa, a za svaki od tih modela mogu se ponovno dati različiti algoritmi. Različite definicije za pojam grupe značajno mijenjaju njena svojstva. Razumijevanje tih modela ključno je za razumijevanje razlika između različitih algoritama.

2.1.1 Opis problema grupiranja podataka

Problem grupiranja podataka možemo, prema [2], predstaviti formalno kako slijedi. Neka je A skup u kojem se nalazi m elemenata. Svaki element predstavljen je kao vektor $x=\{x_1,x_2, ..., x_d\}$, koji predstavlja podatak opisan s d značajki. Neka postoji i k ($1 \leq k \leq x$) nepraznih i disjunktnih podskupova $P_1,P_2,....,P_k$ za koje vrijedi da :

1.Unija svih podskupova daje skup A.

2.Presjek između svih podskupova mora biti prazan skup (disjunktni skupovi)

3.Svaki podskup mora sadržavati barem jedan element (neprazni skup)

Skup P naziva se particija skupa A. Elemente particije $P=\{P_1,P_2,...,P_k\}$ nazivamo grupe. Jednu particiju skupa A možemo označiti P(A;k).

Jedan od klasičnih problema grupiranja je problem, gdje je zadan skup A sa svim njegovim elementima te particija P sa brojem elemenata k (brojem grupa). Potrebno je odrediti elemente svake grupe $P_1,...,P_k$. Običan kriterij kojim se odabire kojoj grupi pripada koji element je prema udaljenosti. Bliski elementi pripadaju istoj grupi. U ovom radu koristi se isključivo Euklidska udaljenost (najkraći put između dvije točke u euklidskom prostoru). U particijskim algoritmima svaka grupa predstavljena je svojim centroidom C_k [1]. Centroid se dobiva računanjem srednje vrijednosti svih elemenata u grupi prema jednadžbi (2-1).

$$C_j = \frac{1}{|P_j|}\sum_{x_i \in P_j} x_i \qquad (2\text{-}1)$$

Isto tako problem grupiranja predstavlja odabir optimalne particije, gdje su zadani elementi skupa A. Ovdje treba uvesti kriterij da je bolja ona particija čiji su elementi kompaktniji i bolje razdvojeni. Elementi pojedine grupe moraju biti što kompaktniji, a grupe što razdvojenije. Jedan od kriterija koji pokazuje koja particija je bolja je rezidualni zbroj kvadrata koji mjeri udaljenost elemenata skupa [2]. Recimo da za elemente skupa A treba pronaći optimalnu particiju P_k. Rezidualni broj kvadrata dobiva se prema jednadžbi (2-2).

$$d_{LS} = \sum_{j=1}^{k}\sum_{x_i \in Pj} ||x_i - c_j||^2 \qquad (2\text{-}2)$$

2.2 Algoritam *k*-means

Već su navedeni i ukratko objašnjeni neki od algoritama grupiranja. Ovo poglavlje detaljnije objašnjava algoritam k-means. Algoritam k-means je metoda vektorske kvantizacije koja je popularna za grupiranje u rudarenju podataka. Algoritam k-means jedan je od particijskih algoritama. Tako K-means grupiranje ima za cilj podijeliti n podataka na k grupa u kojima se svaki podatak pridružuje najbližem centroidu, koji predstavlja grupu. Algoritam traži optimalnu particiju podataka pokušavajući pronaći što manji rezidualni zbroj kvadrata d_{LS} (jednadžba (2-2)) tokom svog iterativnog procesa. Broj k, koji govori u koliko grupa se podatci grupiraju, mora biti unaprijed poznat.

Algoritam k-means je iterativan , odnosno algoritam penjanja uzbrdo (engl. *hill climbing algorithm*) koji započinje sa nekim proizvoljnim rješenjem problema, što u algoritmu k-means predstavljaju prva odabrana mjesta centroida sa pridruženim podacima, a zatim se pokušava pronaći bolje rješenje unošenjem postupnih promjena. Ako se pronađe bolje rješenje, unosi se nova promjena, i tako dalje, sve dok se daljnja poboljšanja ne mogu pronaći. Sam proces rada algoritma može se predstaviti u 4 koraka:

1. Zadavanje broja k, koji govori u koliko grupa će se grupirati podatci. Broj grupa može se odabrati slučajnim odabirom ili odabirom na temelju nekih prethodnih znanja o podatcima koji se grupiraju. Zatim se slučajnim odabirom odabiru mjesta centroida pojedine grupe. Često se odabiru slučajna mjesta centroida na kojima se nalaze podatci s čim se osigurava da početni centroidi neće biti previše udaljeni od podataka. K centroida može se predstaviti vektorom

$$C = [c_1, c_2, \ldots, c_K]$$

2. Svaki podatak pridtružuje se njemu najbližem centroidu. Udaljenosti između centroida i podataka računamo pomoću formule za Euklidsku udaljenost. Formula (2-3) predstavlja pronalazak najbližeg centroida podatku x.

$$za\ svaki\ c_i\ do\ k \quad ako\ Udaljenost(x\ , c_i) < Udaljenost(x\ , c_{min})$$
$$onda \quad Udaljenost(x\ , c_{min}) = \sqrt{(x\ - c_i)^2} \qquad (2\text{-}3)$$

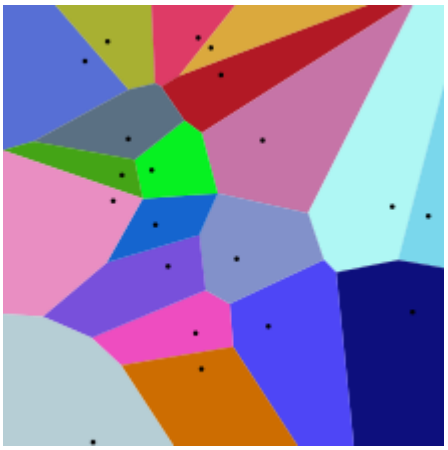
3. Računanje novih pozicija centroida svake grupe na temelju novih pridruživanja. Nove pozicije centroida su aritmetička sredina svih podataka u pojedinim grupama koja se računa jednadžbom (2-4), gdje N_i predstavlja broj podataka u grupi P_i.

$$c_i = \frac{1}{Ni} \sum_{x_j \in P_i} x_j \qquad (2\text{-}4)$$

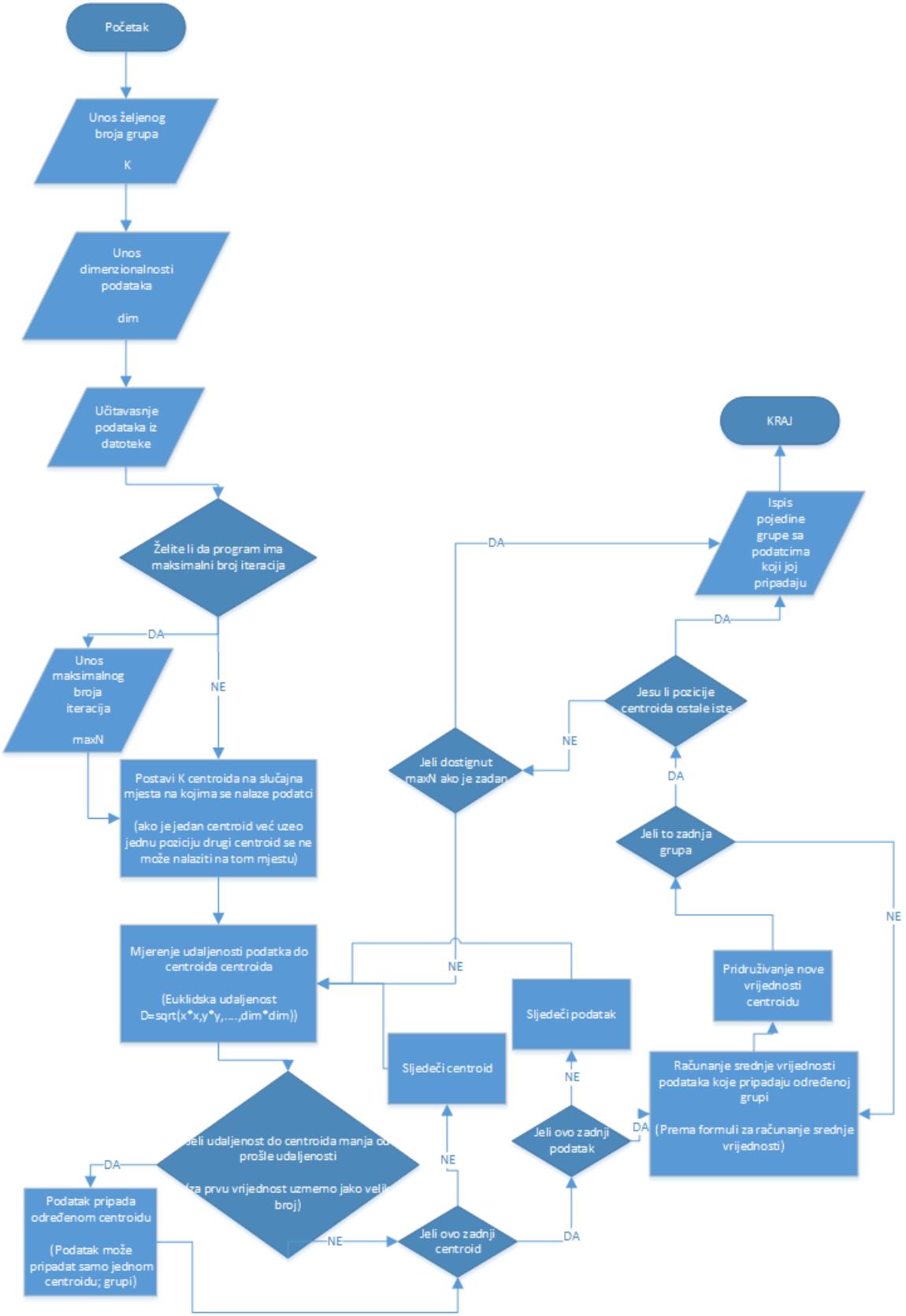
4. Ponavljanje koraka 2 i 3 dok god se pozicije centroida ne prestanu mijenjati. Može se se i za uvjet postaviti da maksimalni broj iteracija bude neki konačan broj [1].

Grupiranje u drugom koraku radi se na temelju najkraćih udaljenosti i prema tome to je Voronoijeva podjela [1]. U matematici Voronoijev dijagram je posebna vrsta razlaganja na

diskretne skupove objekata u prostoru, kao naš primjer gdje imamo skupove točaka u prostoru koji su određeni udaljenošću. U najjednostavnijem slučaju, zadan je skup točaka u ravnini koje su Voronoijeva područja, što su u ovom slučaju centroidi. Svaka ta točka (centroid) ima Voronoijevu ćeliju. Ćelija je prostor oko centroida koja se sastoji se od točaka najbližih tom centroidu. Na slici 2.2 prikazan je primjer Voronoijevog dijagrama, a na slici 2.3 prikazan je dijagram načina rada algoritma k-means.



Slika 2.2. 20 točaka sa njihovim voronoijevim ćelijama (https://en.wikipedia.org/wiki/Voronoi_diagram)



Slika 2.3. Dijagram načina rada algoritma k-means

2.2.1 Prednosti i nedostatci algoritma k-means

Algoritam k-means smatra se jednim od osnovnih algoritama u grupiranju podataka zbog svoje jednostavnosti implementacije i zato jer dobro radi za velik broj praktičnih problema [1]. Vremenska složenost algoritma je O(kndT) gdje k predstavlja broj grupa, n broj podataka, d broj značajki s kojima su opisani podaci i T broj iteracija. U većini slučajeva k,d i T su mnogo manji od n pa se može reći da je vremenska složenost O(n), linearna složenost (k-means je dobar izbor kada je mnogo podataka za grupiranje). Iako algoritam k-means ima nekih vrlo poželjnih svojstava isto tako ima dosta velikih nedostataka.

Neke od prednosti su da algoritam k-means ima jednostavan i shvatljiv princip rada koji se može opisati u četiri koraka [3]. Implementacija algoritma je isto tako dosta jednostavan problem. Lako se može unaprijediti i lako se naprave promjene u algoritmu u slučaju pogreške. Algoritam radi bolje na većim skupovima podataka i linearne je vremenske složenosti. Nakon grupiranja podatci u grupi su kompaktni i dobivena rješenja su točna.

Neki od nedostataka su da algoritam k-means ne može odrediti optimalan broj grupa, prije početka rada mora se zadati broj grupa. Jedno od većih problema algoritma je što nije konzistentan. Pri pokretanju algoritma više puta za isti skup podataka velika je vjerojatnost da će se pojaviti različita rješenja. Razlog zašto se to događa je jer se prva mjesta centroida biraju slučajno. Poredanost podataka u skupu isto zna utjecati na rezultat. Podatci moraju biti numeričkog tipa da bi algoritam radio [3]. Zbog ovih nedostataka postoje strategije i poboljšanja koja unaprijeđuju algoritam k-means.

2.2.2 Poboljšanja algoritma k-means

Početan odabir centroida u algoritmu k-meansu jedan je od većih problema tog algoritma, jer slučajnim odabirom rješenje grupiranja neće svaki puta biti isto. Isto tako velika je vjerojatnost da postoji i bolji način na koji se moglo krajnje grupirati podatke. Zato postoje razna poboljšanja algoritma koji osiguravaju bolji odabir početnih centroida. U nastavku su navedeni neki od poboljšanja.

Algoritam k-means uobičajeno daje bolje rezultate ako su početni centroidi što udaljeniji jedni od drugih, nije dobro kad su skupljeni zajedno. Jedna od metoda za pronalaženje početnih pozicija centroida je k-means++. Ukratko koraci metode su:

1. Prvi centroid izabire se slučajno (na mjesto nekog podatka)
2. Računa se udaljenost D između centroida i njima najbližim podatcima
3. Novi centroid postavlja se na mjesto od nekog podatka koji je proporcionalan s vjerojatnošću D^2

4. Ponavljaju se drugi i treći korak dok se ne postave k centroida [4].

U algoritmu k-means potrebno je prije samog grupiranja odabrati broj grupa. To zna biti problem ako sam korisnik ne zna koliko grupa želi imati. Jedan od algoritama koji pomaže pri tome je ISODATA (eng. *Iterative Self - Organizing Data Analysis Technique*) algoritam koji dinamički procjenjuje broj K. Razlika od algoritma k-means je što se pri radu prilagođava broj grupa spajanjem jedne grupe s drugom ili razdvajanje jedne grupe u više grupa ovisno o nekim unaprijed zadanim pravilima. Grupa se razdvaja u dvije ako standardna devijacija u grupi prelazi predefinirani prag ili ako grupa ima više podataka nego je to dozvoljeno. Dvije grupe se spajaju ako je broj podataka u grupi manji nego što je predefinirano ili ako su dva centroida bliže nego je dozvoljeno. Broj grupa k koji se zada u ISODATA algoritmu nije konačan broj grupa nego broj grupa od kojeg se kreće algoritam [5].

2.3 Relativni indeksi za vrednovanje particija

Vrednovanje grupiranja predstavlja ocjenjivanje koliko je dobar rezultat grupiranja. Vrednovanjem grupiranja možemo usporediti algoritme grupiranja te odrediti koji algoritam je bolji u kojem slučaju. Postoji unutarnja, vanjska i relativna validacija grupiranja [1]. Vanjska validacije bazira se na nekoj već zadanoj strukturi koja je odraz već poznatih informacija o podatcima. Unutarnja validacija ne ovisi o vanjskim informacijama (o prijašnjem znanju o podatcima) nego se struktura grupe analizira direktno.

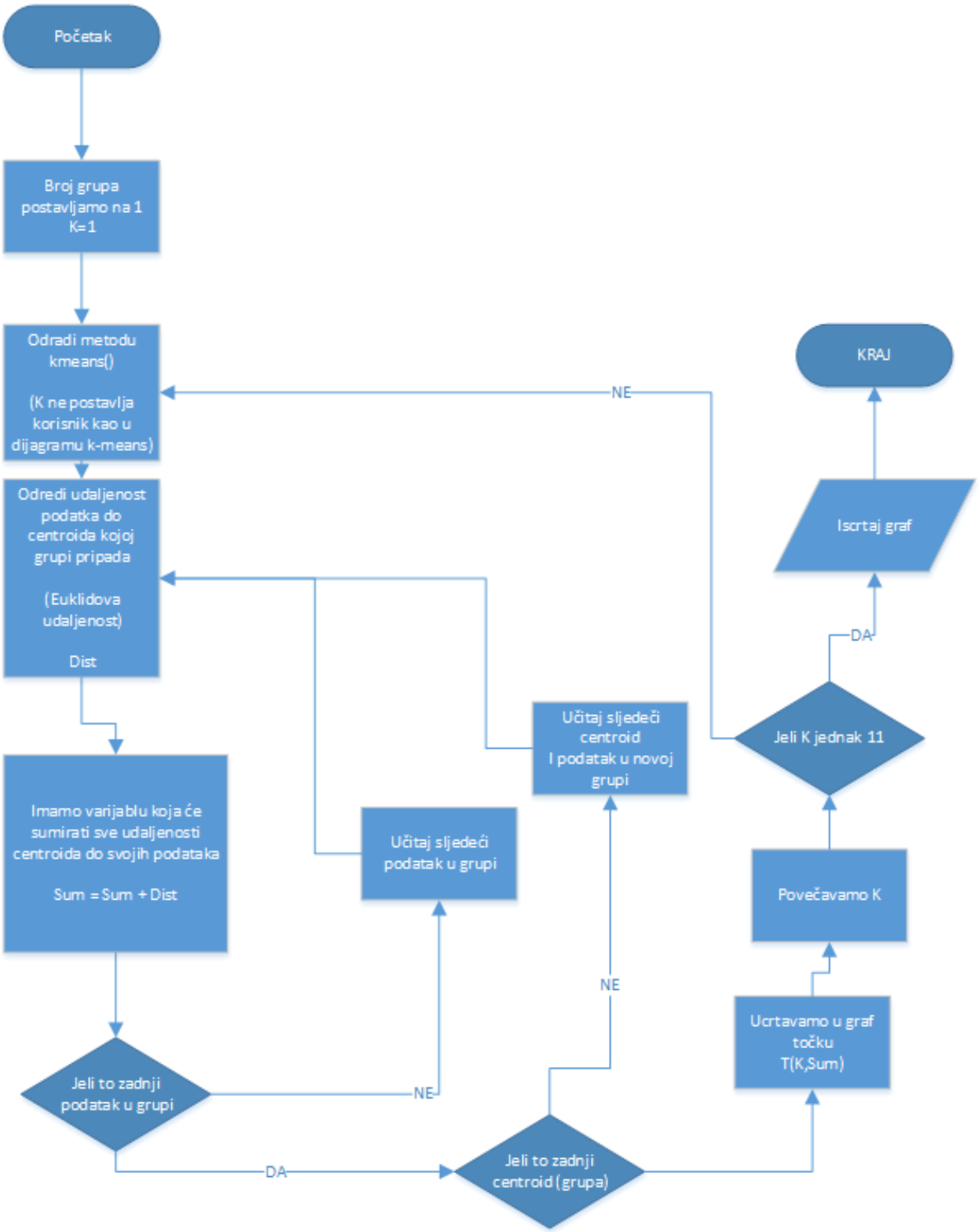
Relativna validacija, za razliku od unutarnje i vanjske ne zahtjeva nikakva statistička testiranja nego se uspoređuju sami rezultati dobiveni pomoću različitih algoritama ili pomoću jednog, ali sa izmijenjenim ulaznim parametrima. Algoritmu k-means se kao ulazni parametar mora upisati broj grupa k. Ako particija ima prevelik broj grupa, zakomplicirati će se struktura grupa i teže je interpretirati i analizirati takve rezultate, a ako particija ima premali broj grupa može doći do gubitka informacija. Zato je dobro znati je li odabrani broj grupa prikladan za određeni skup.

Jedan od jednostavnijih načina kako odrediti je li k prikladan je vizualizacijom rezultata. Ako su podatci jednodimenzionalni ili dvodimenzionalni grupirani podatci mogu se prikazati u ravnini. Vizualizacijom se može ugrubo odrediti je li odabrani k dobar i ako nije može sse i odrediti novi k, ali problem je ako se podatci ne mogu vizualno prikazati (više od dvije dimenzije). Vizualizacija može pomoći samo u posebnim slučajevima, gdje se podatci mogu lijepo prikazati, isto tako vizualizacijom odluka o dobrom k može biti subjektivna.

U algoritmima koji moraju kao ulaz dati broj grupa k može se napraviti slijed grupiranja sa particijama gdje je broj grupa k od k_{min} do k_{max}. Između tih particija se onda može procijeniti koja particija je bolja. Jedan od načina procjene je pomoću funkcije cilja, metoda lakta (engl. *elbow method*) [6]. Funkcija cilja pokazuje ukupno rasipanje podataka svake grupe te ukupno rasipanje particije do njihovih centroida. Funkcija cilja računa se prema jednadžbi (2-5).

$$F_{LS} = \sum_{j=1}^{k} \sum_{x \in P_j} ||c_j - x||^2 \qquad (2\text{-}5)$$

Povećanjem broja grupa vrijednost funkcije cilja monotono opada. Zato se kao optimalnu particiju uzima ona particija za koju vrijednost funkcije cilja naglo opada. Ako se to prikaže na grafu ovisnosti funkcije cilja o broju grupa vidi se da graf izgleda poput lakta, zato naziv metoda lakta. Naravno taj kriterij nije uvijek prikladan, ali zajedno s nekim drugim uvjetima može ukazati na traženu particiju s prikladnim ili čak optimalnim brojem grupa [2]. Na slici 2.4 prikazan je dijagram gdje je korištena metoda lakta za pronalazak optimalne particije, gdje su podatci grupirani algoritmom k-means.



Slika 2.4. Dijagram prikaza rada metode lakta

Sljedeći način procjene optimalne particije su relativni indeksi. Relativni indeksi kombiniraju informacije o particijama kao što su kompaktnost podataka unutar grupe, razdvojenost pojedinih grupa, uzimaju u obzir faktore kao što su kvadratna pogreška, geometrijska i statistička svojstva podataka, broj podataka, broj grupa i slično [1]. Optimalnu particiju dobiva se tako da se računa pojedini relativni indeks za svaku particiju i ona koja ima najbolju vrijednost (ovisno o relativnom indeksu) smatra se optimalnom. Postoji preko 30 vrsta indeksa. Neki od njih su Dunn indeks, Calinski-Harabasz indeks, Davies-Bouldin indeks, C indeks, GDI indeks, Ball-Hall indeks i Banfield-Raftery indeks [7]. Ne postoji idealan indeks koji je uvijek najbolji, sve ovisi o raznim uvjetima, broju podataka, njihovim pozicijama itd. Dva često korištena indeksa su Davies-Bouldin i Calinski-Harabasz indeks.

2.3.1 Calinski-Harabasz indeks

CH indeks predložili su T.Calinski i J.Harabasz 1974. godine [2]. Indeks je definiran tako da interno kompaktnija particija čije se grupe dobro međusobno razdvojene imaju veću CH vrijednost. Što znači da se particija s najvećom CH vrijednošću smatra optimalnom. CH indeks računa se prema jednadžbi (2-6) gdje n predstavlja ukupni broj podataka, F_{LS} funkciju cilja, a G dualnu funkciju.

$$CH(k) = \frac{G(P_i)/(k-1)}{F_{LS}(P_i)/(n-k)} \qquad (2\text{-}6)$$

Funkcija cilja prikazuje ukupno rasipanje elemenata svih grupa do njihovih centroida. Što je F_{LS} manji to je rasipanje manje, što znači da su grupe kompaktnije. Vrijednost dualne funkcije pokazuje ukupnu težinsku razdvojenost centroida. Dualna funkcija računa se prema jednadžbi (2-7) gdje n_j predstavlja broj podataka u grupi. Što je vrijednost funkcije G veća to su centroidi skupova c_j udaljeniji od globalnog centroida c [2].

$$G = \sum_{j=1}^{k} n_j \left|\left|c_j - c\right|\right|^2 \qquad (2\text{-}7)$$

2.3.2 Davies-Bouldin indeks

Davies i Bouldin predložili su DB indeks 1972. godine [2]. Indeks je definiran tako da interno kompaktnija particija čije grupe su međusobno bolje razdvojene ima manju DB vrijednost. Što znači da se particija s najmanjom DB vrijednošću smatra optimalnom. Neka je

točka c centaroid svih podataka skupa A. Pomoću centroida računamo varijancu (prosječna suma kvadratnih odstupanja) skupa jednadžbom (2-8).

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} ||c - x_i||^2 \qquad (2\text{-}8)$$

Iz statistike je poznato je da se u krugu K(c,σ) sa središtem u točki c i radijusom σ nalazi oko 68% točaka skupa. Za dva centroida c_1 i c_2 u skupu A, možemo reći da su njihovi krugovi K1(c_1, σ_1) i K2(c_2, σ_2) razdvojeni (ako nema presjeka između dva kruga) ako vrijedi da

$$\frac{\sigma_1+\sigma_2}{||c_1-c_2||} < 1, \qquad (2\text{-}9)$$

gdje σ predstavlja standardnu devijaciju

Promatranjem u optimalnoj particiji odnos jedne grupe s ostalim grupama veličinom

$$D_j = max_{s\neq j}\frac{\sigma_j+\sigma_s}{||c_j+c_s||} \qquad (2\text{-}10)$$

dobili smo maksimalno preklapanje promatrane grupe s jednom od ostalih grupa. Tako možemo dobiti prosjek maksimalnih preklapanja svake grupe preko jednadžbe (2-11).

$$\frac{1}{k}(D_1 + D_2 + \cdots + D_k) \qquad (2\text{-}11)$$

Prosjek predstavlja mjeru kompaktnosti i vanjske razdvojenosti. Kada je vrijednost manja particija je kompaktnija i grupe su međusobno razdvojenije.
DB indeks za traženje optimalne particije računa se jednadžbom (2-12) [2].

$$DB(k) = \frac{1}{k}\sum_{j=1}^{k} max_{s\neq j}\frac{\sigma_j+\sigma_s}{||c_j-c_s||} \qquad (2\text{-}12)$$

2.4 Automatsko određivanje prikladnog broja grupa

U nekim slučajevima odabir prikladnog broja grupa odabire se samom prirodom problema. Primjer takvog problema grupiranja je grupiranje gljiva po njihovoj jestivosti. Tada bi broj grupa bio tri, za jestive, nejestive i otrovne gljive. Ali već je poznato da nekada nije tako lako odrediti broj grupa. Ako u zadanom skupu podataka nisu poznate nikakve informacije za taj skup tada kada tražimo prikladan broj grupa tražimo da podatci u grupi budu što kompaktniji, a grupe što međusobno udaljenije.

Ovisno o algoritmu grupiranja postoje razni načini kako odrediti takvu particiju. Za algoritam k-means (gdje broj grupa mora biti unaprijed zadan) jedan od najboljih načina su već spomenuti indikatori zvani indeksi. Grupiraju se podatci za niz vrijednosti k te se za svaki računaju indeksi. Optimalna particija biti je ona za koju indeks ima najbolju vrijednost. Problem kod indeksa je što iako su indeksi dobri za određivanje broja grupa nije svaki indeks dobar u svakoj situaciji. Zato je dobar način da se dobivena optimalna particija ne bazira samo na jedan indeks nego na više njih. Validacija grupiranja smatra se jednom od kompliciranijih dijelova grupiranja podataka, ali iznimno je važna [8].

2.5 Primjena grupiranja podataka

Primjenu grupiranja moguće je pronaći u svim područjima znanstvenih i primijenjenih istraživanja. U ekonomiji grupiranje služi u klasifikaciji nabavljača (za određivanje povoljnih nabava), u marketingu gdje se grupiraju kupci sa sličnim zanimanjima u svrhu efikasnog reklamiranja. Klasifikacija životinja u (u koljena, razrede, porodice) primjer je grupiranja u biologiji. Za segmentaciju CT (*engl. Computed tomography*) i PET (*engl.positron emission tomography*) slika primjenjuje se grupiranje u medicini. U prometu grupiranje se primjenjuje u identifikaciji prometnih čepova [2].

2.5.1 Primjena algoritma k-means

Jedan konkretan problem koji se rješava grupiranjem je identificiranje spam poruka. Spamom može doći i do krađe identiteta. Poruke koje se nalaze u spam folderu su poruke koje je algoritam identificirao kao spam. Algoritam koji se pokazao kao efektivan algoritam za pronalazak spam mailova je algoritam k-means [9]. Algoritam gleda različite odlomke poruke (zaglavlje, naslov, tekst poruke), riječi iz odlomaka onda filtrira kroz spam filter koji određuje rang korisnosti pojedinih odlomaka i riječi. Još neki zanimljivi problemi koje se mogu riješiti algoritmom k-means su : Profiliranje sumnjivaca, optimizacija trgovine isporuke (optimalan broj stanica isporuke), segmentacija korisnika u marketingu i segmentacija slika [9].

3 OSTVARENO PROGRAMSKO RJEŠENJE

Ostvareno programsko rješenje napravljeno je u besplatnom open source IDE-u (engl. *integrated development environment*), SharpDevelop-u. SharpDevelop dizajniran je kao besplatna alternativa Microsoft Visual Studia. Ono što je korišteno iz SharpDevelopa za ostvareno programsko rješenje je WPF (engl. *Windows Presentation Foundation*) što je jedan od koncepata iz .NET razvojne cjeline.

WPF je grafički podsustav (sličan Windows formama) razvijen od strane Microsofta za izradu aplikacija (Windows desktop aplikacija). Koristi XAML (engl. *Extensible Application Markup Language*) za izradu i dizajn interface-a i programski jezik C# za programiranje funkcionalnosti programa. Zadatak programskog rješenja je grupiranje podataka algoritmom k-means za određeni broj grupa, te određivanje koji broj grupa je optimalan.

3.1 Način rada programskog rješenja

Način rada programskog rješenja može se podijeliti na četiri dijela. Prvi dio predstavlja učitavanje tekstualne datoteke iz koje se čitaju pojedini elementi, određuje se koliko podataka ima u datoteci i dimenzija podataka (broj značajki koje ih opisuju). Te informacije potrebne su u drugom koraku. Drugi dio predstavlja izvršavanje algoritma k-means (grupiranje podataka) za broj grupa od k_{min} do k_{max}. Prije samog algoritma moraju se odrediti minimalni i maksimalni k. Izvršavanjem ili izvođenjem algoritma k-means može se obaviti treći korak programa. Treći dio predstavlja računanje funkcije cilja, CH i DB indeksa te prikaz vrijednosti pojedinih na grafu za različite vrijednosti k, odnosno različite brojeve grupa. Pomoću tih grafova dobija se predodžba koji je optimalni broj grupa za grupiranje tih podataka. U četvrtom koraku iz izračunatog intervala k može se prikazati koji elementi pripadaju kojoj grupi. U sljedeća četiri potpoglavlja objašnjena su ta četiri dijela programa.

3.1.1 Učitavanje podataka

Prije samog grupiranja te određivanja optimalnog broja grupa prvo se trebaju učitati podatci s kojima će se obavljati navedene procedure. Učitavanjem tekstualnih datoteka određenog tipa čitaju se podatci. U datoteci svaki podatak mora biti u novom redu, a dimenziju podatka prikazuje se razmakom. Kao decimalni separator može se koristiti decimalna točka i zarez.

Ako nije odabrana dobra tekstualna datoteka ili nije izabrana ni jedna datoteka dolazi do greške i treba se ponovno odabrati datoteka. Odabirom datoteke pravog tipa u određene varijable sprema se broj podataka i dimenzija podataka. Sami podatci se prvo spremaju u *string*, a kasnije nakon što se prođu provjere, u 2D matricu tipa *double* koja ima broj redaka kao i broj podataka, a stupaca kao dimenzija podataka.

3.1.2 Algoritam k-means

Drugi dio programa je grupiranje podataka. Glavni zadatak programskog rješenja je pokušati odrediti optimalnu particiju za podatke koji se odrede u prvom dijelu programa. Ako se želi pronaći optimalna particija pomoću algoritma k-means mora se grupirati podatke za cijeli skup grupa te za svako grupiranje odrediti indekse koji kasnije pomažu za pronalazak optimalne particije. Zato je potrebno grupirati podatke. Prije samog grupiranja treba odrediti skup k-ova iz kojih se kasnije traži optimalna particija. Upisivanjem k_{min} i k_{max} dobiva se skup k-ova [k_{min},k_{max}] . U programu je ograničeno da k_{min} mora biti barem 2, a k_{max} ne smije biti veći od 50. Isto tako k_{min} mora biti manji od k_{max}, te oni moraju biti cijeli brojevi. Ako je neki od uvjeta narušen pojavljuje se poruka.

Sljedeći korak je sam algoritam k-means. U poglavlju 2.2 već su navedeni koraci algoritma k-means, isto tako na slici 2.3 prikazan je dijagram toka. Prvi korak je postavljanje centroida na slučajna mjesta. U programskom rješenju centroidi su postavljeni na slučajna mjesta na kojima se nalaze podatci uz uvjet da se dva centroida ne smiju nalaziti na istom mjestu. Centroidi se spremaju u novoj matrici. Zatim se pridružuju podatci sebi najbližim centroidima. U 1D matricu koja ima elemenata kao i broj podataka (gdje svaki element predstavlja jedan podatak) upisujemo kojoj grupi pripada koji podatak. Na primjer ako podatak 4 pripada grupi 2: dataPointIsPartOfTheGroup[3] = 1 (kreće se od indeksa nula).

Zatim se računaju nova mjesta centroida na temelju pridruživanja podataka prema jednadžbi (2-4). Imamo još jednu varijablu za spremanje novih centroida s razlogom da se stara mjesta mogu usporediti s novima u sljedećem koraku. Zadnja dva koraka se ponavljaju dok se ne ispune jedno od dva uvjeta. Prvi uvjet je da se prestanu mijenjati pozicije centroida, a drugi uvjet je da se dosegne maksimalni broj iteracija koji se može podesiti na početku algoritma. Maksimalni broj iteracija mora biti u intervalu od 1 do 100. Upisom krivog broja pojavljuje se poruka koja javlja da se upiše točno. Broji se koliko puta centroidi ostaju isti, ako dvaput zaredom ostanu isti algoritam se završava. Algoritam k-means vrti se za svaki broj grupa k u intervalu koji je već prije određen.

3.1.3 Funkcija cilja, CH i DB indeks

U sljedećem dijelu programa za svako grupiranje iz intervala određuje se funkcija cilja te CH i DB indeks. Nakon njihovog određivanja crtaju se grafovi za svaki od njih. Čim se završi grupiranje za jedan k iz intervala računaju se indeksi za tu particiju. Funkciju cilja računa se tako da se za svaki podatak pita koje je grupe te se računa rasipanje tog podatka od centroida te grupe. Graf funkcije cilja dobivamo kako je prikazano na slici 2.4.

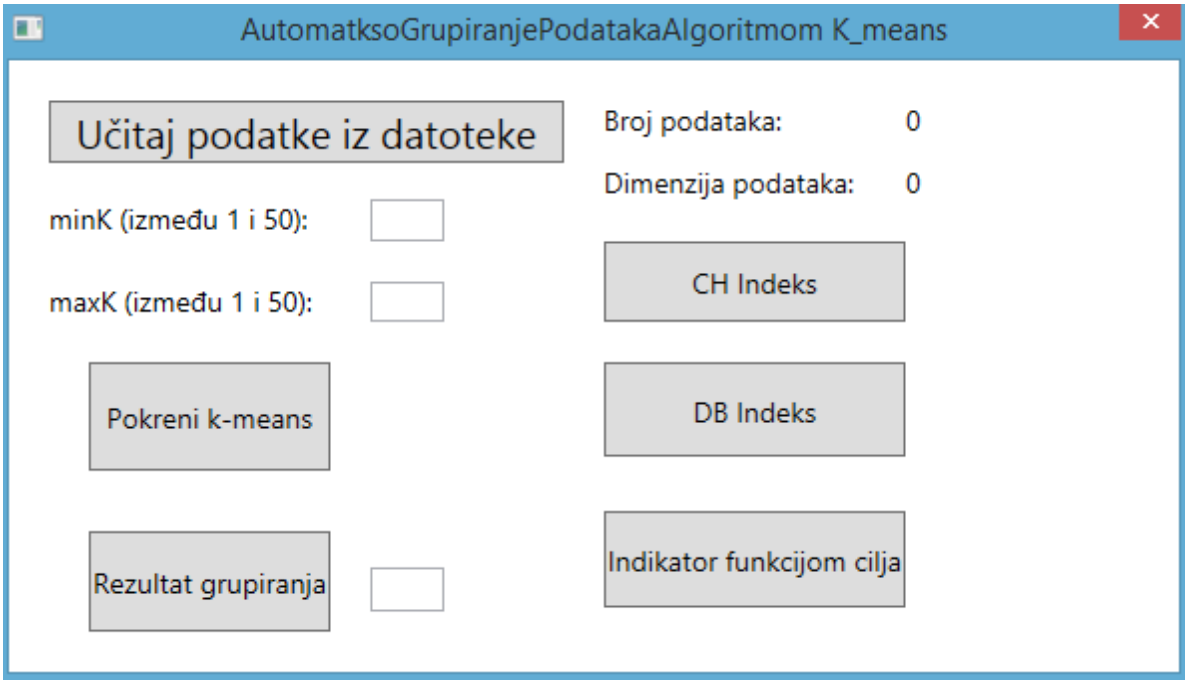
Za računanje Calinski-Harabasz indeksa potrebna je već izračunata funkcija cilja i dualna funkcija koja se računa prema jednadžbi (2-7) . Za računanje dualne funkcije računa se rasipanje svakog centroida do glavnog centroida (srednja vrijednost svih podataka). CH indeks računa se prema jednadžbi (2-6). Za računanje DB indeksa prvo treba izračunati standardnu devijaciju svake grupe s jednadžbom (2-8). Vrijednosti svakog od indeksa za svaki k iz intervala sprema se u svoj niz. Iz vrijednosti tih nizova crtaju se onda grafovi.

3.1.4 Prikaz rezultata grupiranja

Četvrti korak programskog rješenja predstavlja tekstualni prikaz svake grupe odabrane particije sa podatcima pridruženih pojedinoj grupi. Ako su podatci jednodimenzionalni ili dvodimenzionalni mogu se prikazati grafički, ako particija ima manje od 11 grupa. Pošto nisu spremljeni podatci za svako grupiranje u ovom koraku podatci se ponovno grupiraju što znači da će za pojavljivati različita rješenja.

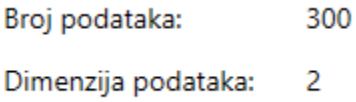
3.2 Prikaz i način uporabe programskog rješenja

U poglavlju 3.1 ukratko je opisan način rada programskog rješenja. Ovo poglavlje bazira se na prikazu i načinu uporabe programa. Nakon što se pokrene program pojavljuje se korisničko sučelje kao na slici 3.1. Prvi korak je učitavanje podataka iz datoteke. Klikom na gumb otvara se prozor gdje možemo odabrati datoteku.



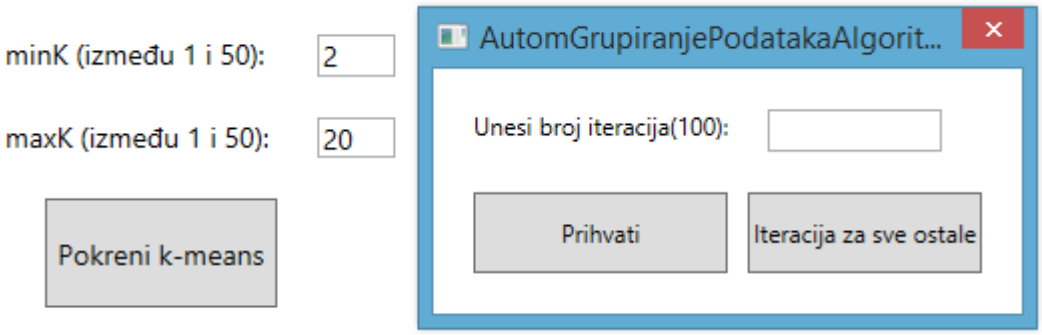
Slika 3.1. Glavno sučelje programa

Ako se ne odabere tekstualna datoteka dobrog tipa ili se uopće ne izabere pojavljuje se poruka upozorenja. Odabirom dobre datoteke na glavnom sučelju mijenja se broj podataka i dimenzija kao na slici 3.2. Za izvršenje algoritma k-means moraju se upisati minK, maxK te učitati podatci.



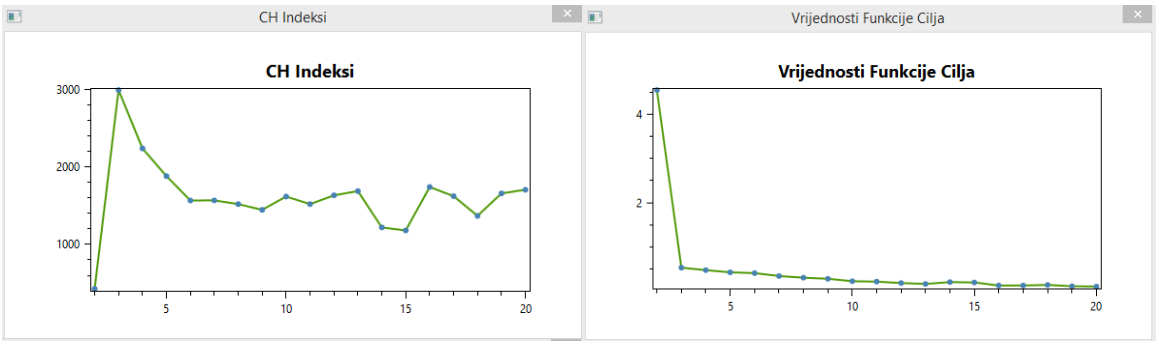
Slika 3.2. Izmijenjen broj podataka i dimenzija podataka

Upisom dobrih graničnih k-ova i pritiskom gumba za pokretanje algoritma, za svaki k iz intervala pojavljuje se poruka za unos broja iteracija kao na slici 3.3. Pritiskom na tipku iteracija za sve ostale sljedeći broj iteracija uvažuje se za sve sljedeće vrijednosti k. Početni broj iteracija je 100. Dozvoljeni broj iteracija je od 1 do 100.

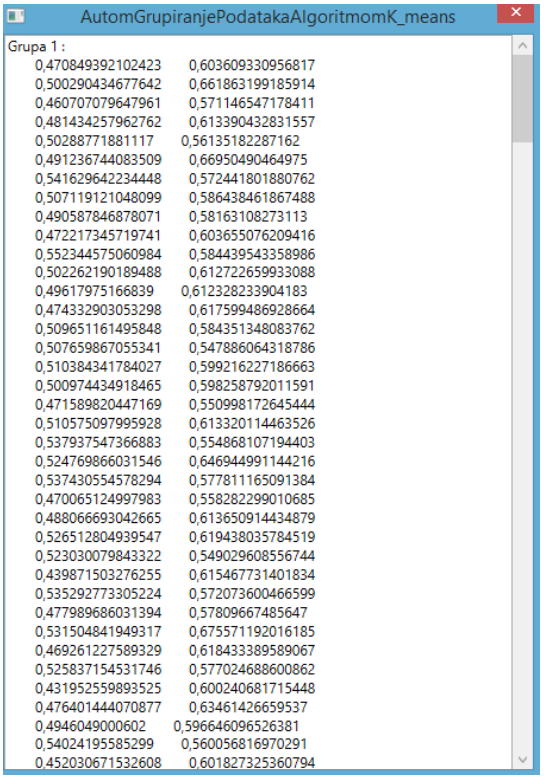


Slika 3.3. Upis broja iteracija

Nakon završetka svih grupiranja mogu se klikom na gumb otvoriti grafovi, graf funkcije cilja, graf CH i DB indeksa. Ako se grafovi pokušaju otvoriti prije grupiranja pojavljuje se poruka koja javlja da prvo mora grupiranje biti izvršeno. Prije nego se otvori graf CH indeksa i DB indeksa pojavljuje se poruka koja govori koliki je optimalni k prema tom indeksu (CH – što veća vrijednost to bolji ; DB – što manja vrijednost to bolji). Na slici 3.4 prikazani su primjeri grafa vrijednosti funkcije cilja i graf CH indeksa koji ukazuju na to da je optimalna particija gdje je k=3. Na slici 3.5 pokazano je kako se prikazuju grupirani podatci.



Slika 3.4. Prijmer grafa vrijednosti funkcija cilja i CH indeksa



Slika 3.5. Neki elementi iz prve grupe

4 EKSPERIMENTALNA ANALIZA

U eksperimentalnoj analizi koristiti se šest skupova. Tri sintetička skupa i tri skupa sa stvarnim podacima. Stvarni podatci su dobiveni nekim izravnim mjerenjima. Sintetički skupovi su skupovi dobiveni pomoću normalne razdiobe. Karakteristike pojedinog sintetičkog skupa prikazane su u tablici 4.1, s karakteristike realnih skupova u tablici 4.2.

Tablica 4.1. Karakteristike sintetičkih skupova podataka

Ime	Skup 1	Skup 2	Skup 3
Dimenzionalnost	2	2	2
Broj podataka	300	625	750
Stvarni broj grupa	3	5	7

Prvi korišteni skup sa stvarnim podatcima je jedan od najpoznatijih skupova podataka korišten u statističkoj klasifikaciji. Skup ima tri klase podataka ; tipa cvijeta irisa (*Iris-setosa*, *Iris-versicolor* i *Iris-virginica*). Prvih 50 elemenata skupa predstavlja cvijet *Iris-setosa*, sljedećih 50 *Iris-versicolor* i zadnjih 50 *Iris-virginica*. Svaki element sastoji se od četiri atributa. Prva dva atributa predstavljaju izmjerenu duljinu i širinu čašićnog listića, a druga dva atributa duljinu i širinu latica. Mjerna jedinica svakog atributa je centimetar [10].

Podatci drugog skupa su rezultati kemijske analize vina uzgajanih u Italiji. Sva vina uzgajana su u istoj regiji, ali postoje tri različite sorte vina. Jedan podatak predstavljen je s 13 atributa. Atributi redom predstavljaju : postotak alkohola, jabučna kiselina, pepeo, alkalnost pepela, magnezij, ukupni fenoli, flavonoidi, neflavanoidni fenoli, *proanthocyanidins*, intenzitet boje, nijansa, OD280/OD315 razrijeđenih vina, pročin. Prvih 59 podataka predstavlja prvu sortu, sljedećih 71 predstavlja drugu sortu i zadnjih 48 podataka u skupu predstavlja treću sortu [11].

Treći skup podataka je iz američke službe za forenzičke znanosti. U skupu postoji šest tipova stakla. Svaki podatak opisan je sa deset atributa, prvi atribut predstavlja ID pa je u modificiranom skupu izbačen, tako da ustvari ima devet atributa. Ti atributi redom predstavljaju: indeks loma, natrij, magnezij, aluminij, silicij, kalij, kalcij, barij i željezo (jedinica mjerenja je težinski postotak u odgovarajućem oksidu). Podatci svake klase su poslagane redom : prva klasa ima 70 podataka, druga 76, treća 17, četvrta 13, peta 9 i šesta 29 [12].

Tablica 4.2. Karakteristike stvarnih skupova podataka

Ime:	cvijet	vino	staklo
Dimenzionalnost:	4	13	9
Broj podataka:	150	178	214
Stvarni broj grupa:	3	3	6

4.1 Postavke eksperimenta

U analizi svih šest skupova Kmin je postavljen na dva, Kmax na deset i maksimalni broj iteracija je svaki put 100. Rezultati za jedan skup podataka su grafovi za svaki od indeksa, zaključak koji je optimalni broj grupa prema tom indeksu, te samo grupiranje podataka (rezultat grupiranja za optimalnu grupu). Rezultati se vrednuju po rezultatu relativnih indeksa i funkcije cilja (koja particija je optimalna prema indeksu), koliko blizu su njihovi rezultati od stvarnog broja grupa. U eksperimentu se koristiti Calinski-Harabasz indeks, Davies-Bouldin indeks i vrijednost funkcije cilja, gdje je korištena metoda lakta za određivanje optimalne particije. Stvarni skupovi modificirani su tako da su se podatci prebacili u format koji program može pročitati.

4.2 Rezultati

Prvo su prikazani rezultati za sintetičke skupove, tablično su prikazani rezultati, grafovi indeksa, te komentari rezultata.

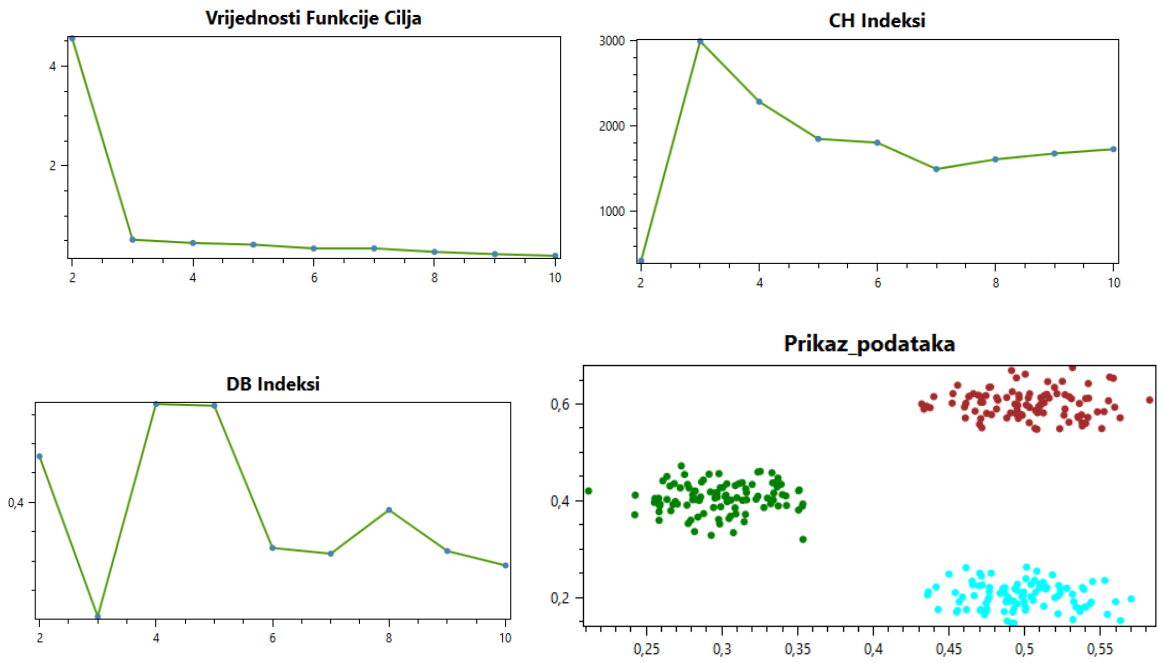
Tablica 4.3. Rezultati optimalne particije prema određenom indeksu sintetičkih skupova

	Optimalna particija dobivena metodom lakta	Optimalna particija dobivena CH indeksima	Optimalna particija dobivena DB indeksima	Pravi broj grupa
Skup 1	3	3	3	3
Skup 2	5	5	5	5
Skup 3	6-8	7	10	7

Iz tablice 4.3 s rezultatima može se vidjeti da su rezultati relativno blizu pravom broju grupa. Sami rezultati algoritma k-means ovise o početno slučajno odabranim mjestima centroida, pa neće svaki puta grafovi biti isti. Prikazani su grafovi koji su se najčešće pojavljivali.

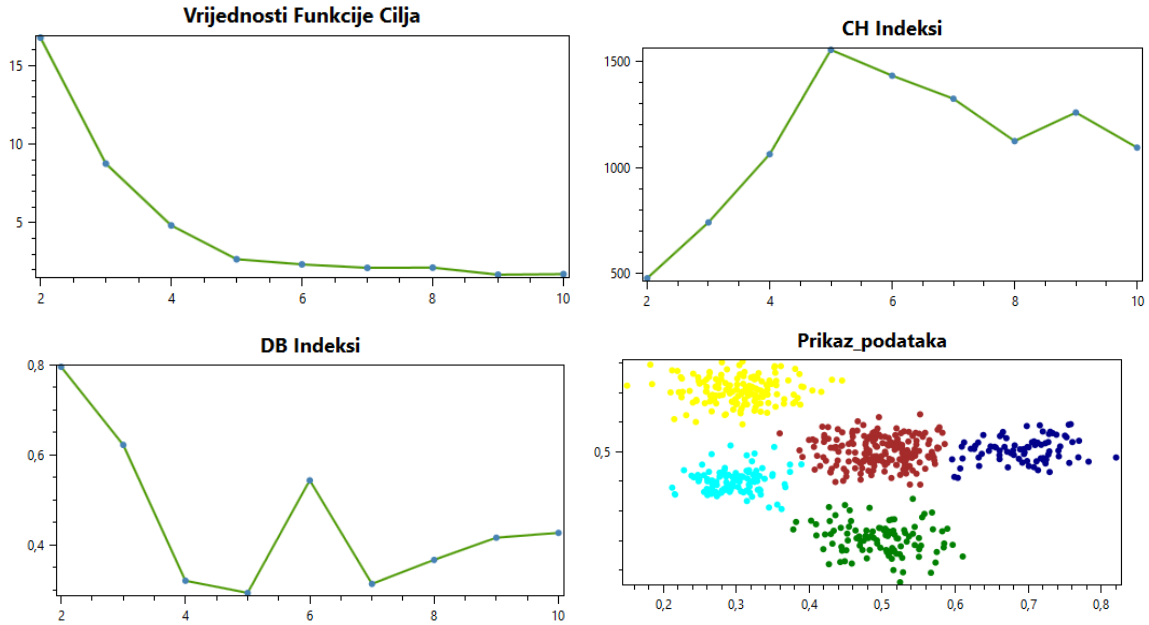
U svakom pokušaju pronalaska optimalne particije za prvu datoteku graf funkcije cilja i CH graf uvijek su ukazivali da je optimalna particija baš tri. Kod DB grafa znalo se dogoditi

da je najmanja vrijednost DB indeksa na k=4 ili k=5. Na slici 4.1 prikazani su najčešći grafovi indeksa, funkcije cilja i grupirani podatci Skupa 1.



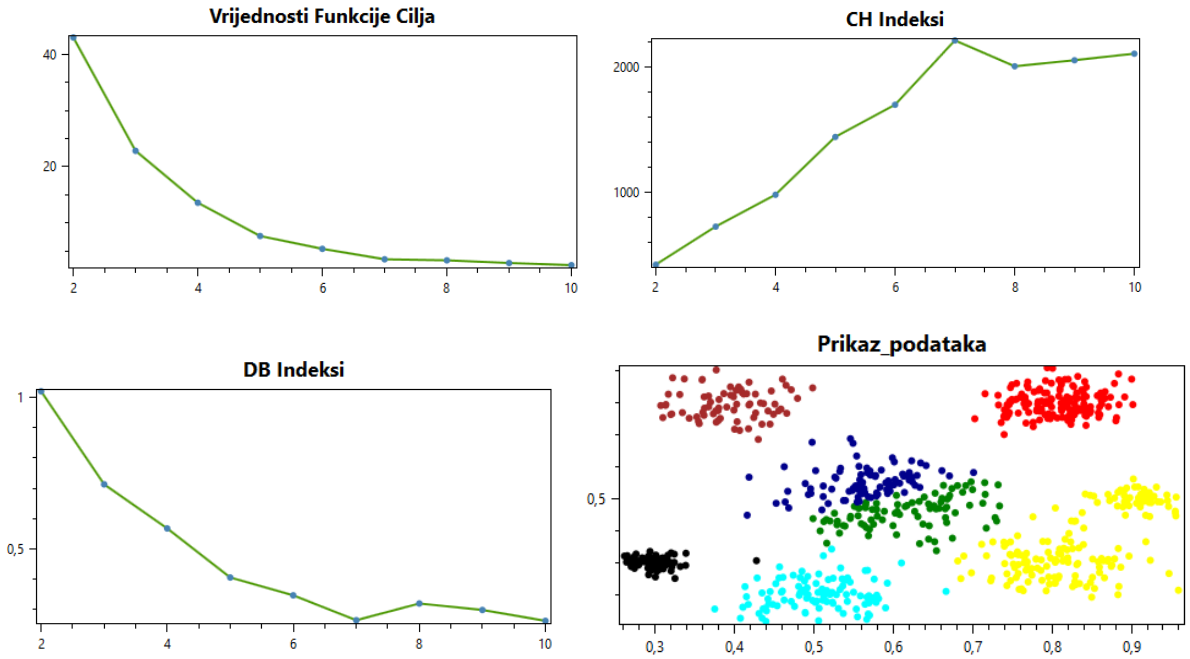
Slika 4.1. Grafovi indeksa i prikaz podataka za Skup 1

Za drugu datoteku graf funkcije cilja i CH graf isto uvijek ukazuju na pravi broj grupa kao optimalnu particiju, k=5. Kod DB grafa često se pokazalo da je optimalna particija 7 ili 8, iako je za k=5 uvijek bila dosta mala vrijednost DB indeksa. Na slici 4.2 prikazani su najčešći grafovi indeksa, funkcije cilja i grupirani podatci Skupa 2.



Slika 4.2. Grafovi indeksa i prikaz podataka za Skup 2

Za treću datoteku par puta se dogodilo da se prema CH grafu pokazalo da je k=9 optimalna particija. Graf funkcije cilja bio je svaki puta sličan, ali mjesto gdje funkcija cilja počne naglo opadati (koji ukazuje gdje je optimalna particija) nije toliko očit kao za prošle dvije datoteke. DB graf često je ukazivao na 8 i 7 kao optimalni broj grupa. Vrijednost DB indeksa za k=7 uvijek je bila relativno mala. Na slici 4.3 prikazani su najčešći grafovi indeksa, funkcije cilja i grupirani podatci Skupa 3.



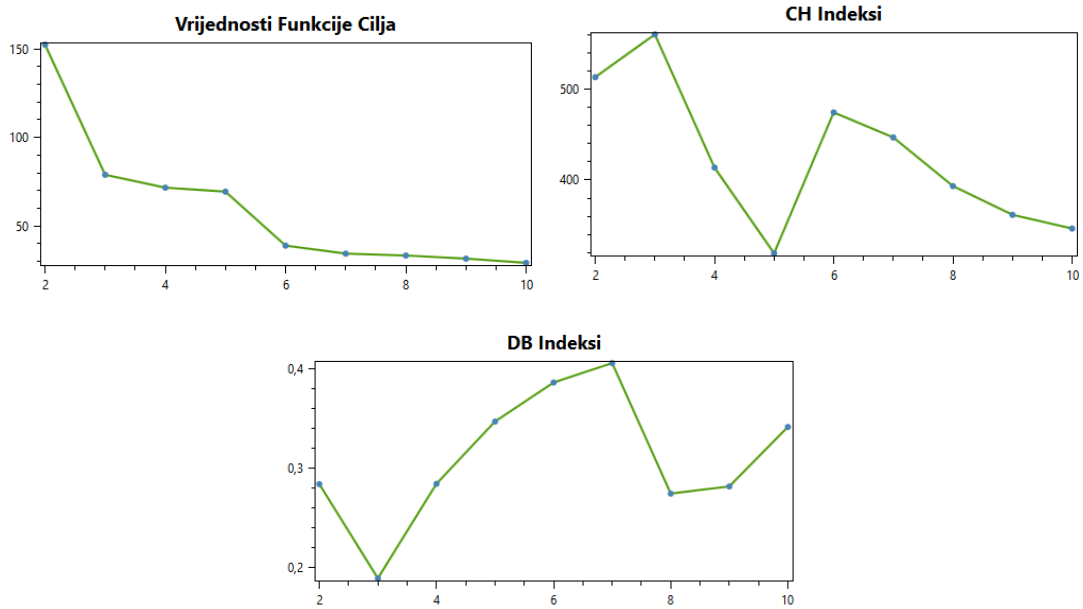
Slika 4.3. Grafovi indeksa za Skup 3

Sve u svemu rezultati traženja optimalne particije za sintetičke skupove ispali su relativno dobri, iako pogotovo za DB indeks, rezultati nisu uvijek konzistentni. U tablici 4.4 prikazani su rezultati traženja optimalne particije za stvarne skupove.

Tablica 4.4. Rezultati traženja optimalne particije prema određenom indeksu stvarnih skupova

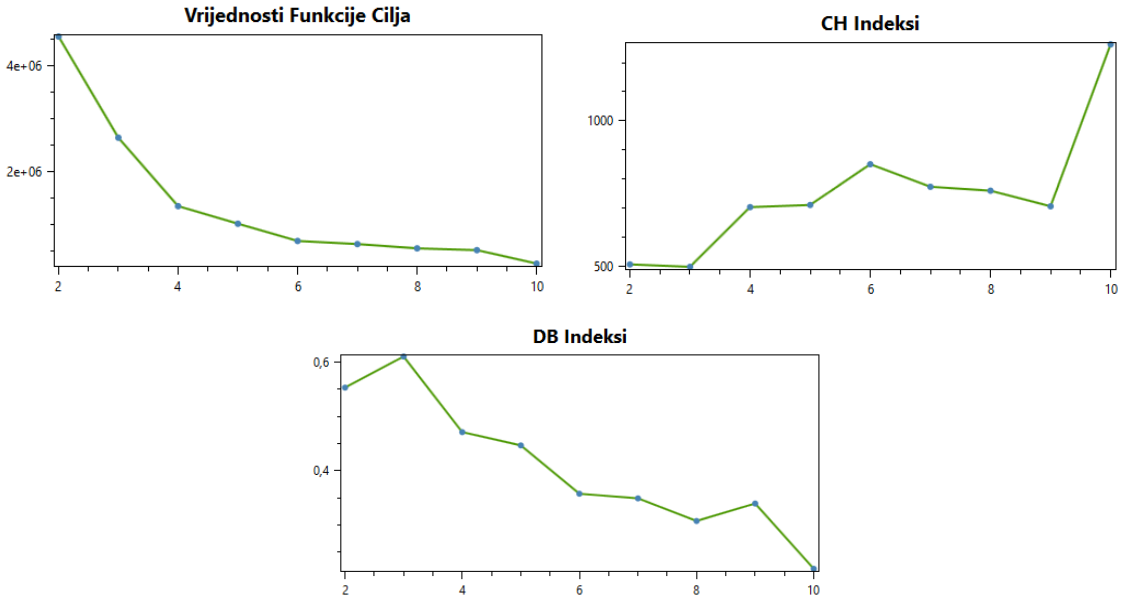
	Optimalna particija dobivena metodom lakta	Optimalna particija dobivena CH indeksima	Optimalna particija dobivena DB indeksima	Pravi broj grupa
cvijet	3 ili 6	3	3	3
vino	7 do 10	10	10	3
staklo	9 ili 5	9	2	6

Za prvu datoteku graf vrijednosti funkcija cilja u svakom pokušaju pronalaska optimalne particije izgledao je slično. U nekim slučajevima je uočljivije da je mjesto gdje funkcija cilja najviše opala na k=3. U grafovima CH indeksa CH vrijednost je većinom bila najveća za k=3. U par slučajeva ispalo je da je k=2 optimalna particija. DB indeks opet nije bio konzistentan, često je najmanja vrijednost bila za k=8 ili k=2. Grupirani podatci za k=3 većinom pripadaju pravoj grupi, oko 10-ak elemenata znaju biti u krivoj grupi. Na slici 4.4 prikazani su najčešći grafovi indeksa i funkcije cilja za skup cvijet.



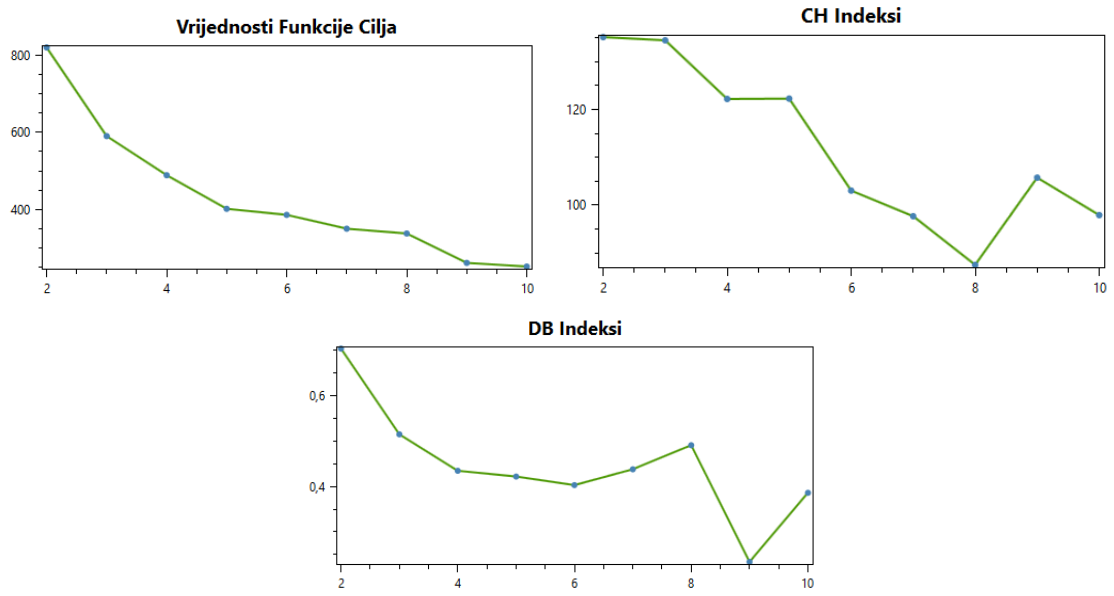
Slika 4.4. Grafovi indeksa za skup cvijet

U drugoj datoteci svaki graf pokazuje skroz različite vrijednosti za optimalnu particiju od stvarnog broja grupa. Metoda lakta u svakom pokušaju pokazuje da je optimalna particija između 7 i 10. Graf CH i DB indeksa pokazuju da je optimalna particija devet ili deset u skoro svakom pokušaju. Rezultati su krivi od stvarnog broja grupa, ali su konzistentni. Na slici 4.5 prikazani su najčešći grafovi indeksa i funkcije cilja za skup vino.



Slika 4.5. Grafovi indeksa za skup vino

Za treću datoteku, kao i za drugu ni jedan graf kao optimalnu particiju nije pokazivao stvarnu vrijednost grupa. Iz grafa vrijednosti funkcije cilja dosta je teško za iščitati optimalnu grupu, ali je u većini pokušaja negdje između 6 i 10. U grafu CH indeksa najveća vrijednost je većinom za k=2, iako se pojavila par puta i na k=5 i k=6. Vrijednost DB indeksa je najmanja za k=8,9 ili 10. Na slici 4.6 prikazani su najčešći grafovi indeksa i funkcije cilja za skup staklo.



Slika 4.6. Grafovi indekse za skup staklo

Iz dobivenih rezultata vidi se da grupiranjem algoritmom k-means i korištenjem indeksa i funkcije cilja dobivena optimalna particija ne bude pravi broj grupa ako su grupe neujednačene (ako jedna grupa ima mnogo više elemenata od drugih). U tri sintetička skupa svaka grupa imala je sličan broj podataka, pa su stvarni brojevi grupa bili relativno slični broju k za koje su indeksi i funkcija cilja pokazali kao optimalnu particiju. Moguće je da na rješenje utjecala i činjenica da su podatci poredani u datotekama.

5 ZAKLJUČAK

U završnom radu zadatak je bio upoznati se s problemima grupiranja. U svrhu toga objašnjeni su problemi grupiranja, problem pronalaska optimalne particije, algoritmi grupiranja i navedene su neke primjene grupiranja. Ostvareno je programsko rješenje koje je grupiralo podatke za različite brojeve grupa te je odredilo koje od tih grupiranja je optimalno. Za grupiranje je korišten algoritam k-means, a za određivanje optimalne particije korištena su 2 relativna indeksa i metoda lakta. Određivanje pomoću metode lakta pokazalo se kao dobra metoda u skoro svakoj situaciji, ali problem je što je nekada dosta teško za odrediti mjesto gdje vrijednost funkcije cilja naglo opada. Određivanje optimalne pomoću DB indeksa pokazalo se kao dobra metoda u nekim situacijama, problem je što u ovakvoj implementaciji s algoritmom k-means koji slučajnim odabirom odabire prva mjesta centroida ova metoda nije bila konzistentna. Određivanje pomoću CH indeksa pokazala se kao najbolja metoda od tri korištene. Rezultati su bili relativno konzistentni uzimajući u obzir da se za grupiranje koristi algoritam k-means. Moguće dorade programskog rješenja mogle bi biti da se nadogradi algoritam k-means, na primjer da se prva mjesta centroida ne odabiru slučajno i da algoritam može raditi i sa tekstualnim podatcima, a ne samo numeričkim. Isto tako moglo bi se unaprijediti da se mogu pročitati podatci spremljeni u drugom formatu, na primjer podatci spremljeni u tablice. Dosta bi jednostavno bilo i dodati još relativnih indeksa za traženje optimalne particije. Za mogući budući rad mogli bi se iskoristiti i drugi algoritmi grupiranja za usporedbu s algoritmom k-means ili grupiranje podataka za neku konkretnu primjenu, kao na primjer segmentacija slike. Problem grupiranja će zasigurno i u budućnosti biti relevantan problem. S razvojem novih tehnologija i sve većim brojem podataka (npr. internet) potrebno će biti modificirati i poboljšavati već postojeća rješenja u svrhu brzine i efikasnosti.

LITERATURA

[1] R. Xu and D. C. Wunsch, Clustering. Hoboken, New Jersey: John Wiley & Sons Inc., 2009.

[2] R.Scitovski , M.Briš Alić, Grupiranje podataka, Sveučilište Josipa Jurja Strossmayera u Osijeku , Ekonomski fakultet u Osijeku , 2016.

[3] Pros and Cons of K-means Clustering, https://www.prosancons.com/education/pros-and-cons-of-k-means-clustering/, pristupljeno 14.8.2019.

[4] R. Smith, Stack Exchange, https://datascience.stackexchange.com/questions/5656/k-means-what-are-some-good- ways-to-choose-an-efficient-set-of-initial-centroids, pristupljeno 15.8.2019.

[5] Unsupervised Classification algorithms, http://www.wu.ece.ufl.edu/books/EE/communications/UnsupervisedClassification.html, pristupljeno 16.8.2019.

[6] Determining The Optimal Number Of Clusters: 3 Must Know Methods, https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/#elbow-method, pristupljeno 15.8.2019.

[7] Bernard Desgraupes, Clustering Indices, University Paris Ouest Lab Modal'X, 2017.

[8] K.Jain and C.Dubes, Algorithms for clustering data, Michigan State University, 1988.

[9] 7 Innovative Uses of Clustering Algorithms in the Real World, https://datafloq.com/read/7-innovative-uses-of-clustering-algorithms/6224, pristupljeno 20.8.2019.

[10] R.A. Fisher, Iris Data Set, UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Iris], pristupljeno 5.9. 2019.

[11] Wine data sets, UCI Machine Learning Repository, [https://archive.ics.uci.edu/ml/datasets/Wine], pristupljeno 5.9.2019.

[12] B.German, Glass Identification Data Set, [https://archive.ics.uci.edu/ml/datasets/Glass+Identification], pristupljeno 5.9.2019.

SIMBOLI I OZNAKE

k – broj grupa

c - centroid

d_{LS} - rezidualni broj kvadrata

F_{LS} - funkcija cilja

G - dualna funkcija

CH - Calinski-Harabasz

DB - Davies-Bouldin

IDE - *integrated development environment*

WPF - *Windows Presentation Foundation*

XAML - *Extensible Application Markup Language*

SAŽETAK

U radu je objašnjen pojam grupiranja, navedeni su problemi u klaster analizi od kojih su naglašeni problem samog grupiranja i problem pronalaska optimalnog broja grupa. Navedeni su neke vrste i algoritmi grupiranja od kojih je algoritam k-means mnogo detaljnije objašnjen. Za metodu pronalaska optimalne particije navedeni su relativni indeksi, CH indeks, DB indeks te metoda lakta. Objašnjen je rad programskog rješenja koji koristi algoritam k-means i već navedene indekse. Objašnjen i je način kako koristiti program. Analizirana su rješenja dobivena u programu za šest odabranih skupova.

Ključne riječi: algoritam k-means, Calinski-Harabasz indeks, Davies-Bouldin indeks, grupiranje, grupiranje podataka, metoda lakta

ABSTRACT

In the final paper the term clustering is explained in detail. Some basic problems of cluster analysis are mentioned, the problem of clustering a set of data and a problem of finding the optimal number of groups for clustering are the two problems that are highlighted. There is also a mention of some types and algorithms for clustering from which k-means algorithm was explained in more detail. CH and DB indices are explained, two methods for finding an optimal partition, as well as elbow method. The way how the realized software solution works as well as how to use it is also explained. Six datasets are selected and are used in the program, solutions given were analyzed.

Keywords: k-means algorithm, Calinski-Harabaz index, Davies-Bouldin index, clustering, cluster analysis, elbow method

ŽIVOTOPIS

Siniša Buršić rođen je u Puli 14. rujna 1996. Prvih pet razreda osnovne škole završio je u Vrsaru u Osnovnoj školi Vladimira nazora. 2008. Preselio se u Vukovar gdje je završio zadnje tri godine osnovnog obrazovanja u Osnovnoj školi Dragutina Tadijanovića. U Vukovaru je završio i srednju školu, Tehničku školu Nikole Tesle. Od 2015. Pohađa Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, preddiplomski studij računarstva. Jako dobro se snalazi na računalu. Zna dobro osnove programiranja, objektno orijentiranog programiranja, web dizajna. Jako dobro se služi engleskim jezikom u pisanju i govoru.

PRILOZI

P1 Završni rad u DOCX i PDF formatu nalazi se na CD-u

P2 Ostvareno programsko rješenje nalazi se na CD-u

P3 Skupovi korišteni u analizi nalaze se na CD-u