Usporedba metoda za razvrstavanje oko središnje točke

Znaor, Filip

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

Sveučilišni studij

USPOREDBA METODA ZA RAZVRSTAVANJE OKO SREDIŠNJE TOČKE

Završni rad

Filip Znaor

Osijek, 2020.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 25.08.2020.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Filip Znaor
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	4460, 17.09.2019.
OIB studenta:	07398827546
Mentor:	Izv. prof. dr. sc. Ivica Lukić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Usporedba metoda za razvrstavanje oko središnje točke
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	25.08.2020.
Datum potvrde ocjene Odbora:	09.09.2020.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 09.09.2020.

Ime i prezime studenta:	Filip Znaor
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	4460, 17.09.2019.
Turnitin podudaranje [%]:	8

Ovom izjavom izjavljujem da je rad pod nazivom**: Usporedba metoda za razvrstavanje oko središnje točke**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Ivica Lukić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Prilikom analize bilo kakvog skupa podataka može se javiti potreba za razvrstavanjem (eng. *clustering*) tog skupa u manje podskupove, odnosno grupe ili grozdove na temelju sličnosti njihovih obilježja. Razvrstavanje objekata u grozdove omogućuje lakše uočavanje određenih sličnosti ili različitosti između objekata u skupu te bolje razumijevanje promatranih objekata, pogotovo u slučajevima kada nemamo prethodnih saznanja o svojstvima i međusobnim odnosima objekata u skupu. Kako je popularnost rudarenja podataka i sličnih računalno zahtjevnih procesa u zadnje vrijeme u porastu, postaje sve bitnije pronalaziti metode za razvrstavanje koje će biti učinkovite i precizne na vrlo velikim i vrlo složenim skupovima podataka.

Kako grozdovi mogu biti različitih oblika i svojstava, postoji veliki broj algoritama za razvrstavanje. U radu su razmatrane metode za razvrstavanje oko središnje točke, odnosno metode koje svaki grozd predstavljaju jednim središnjim vektorom te sve ostale objekte pridružuju grozdu čijem su središtu najbliži, odnosno najsličniji. Opisano je i implementirano nekoliko algoritama, uključujući algoritme za čvrsto razvrstavanje, kao i za neizrazito (eng. *fuzzy*) razvrstavanje.

U drugom poglavlju su opisana dosadašnja dostignuća i saznanja vezana uz razvrstavanje oko središnje točke. U trećem poglavlju je opisan problem razvrstavanja, mjere udaljenosti, algoritmi za razvrstavanje oko središnje točke te mjere kvalitete razvrstavanja. U četvrtom poglavlju opisano je ostvareno programsko rješenje te njegov način rada, a u petom poglavlju su prikazani i interpretirani rezultati eksperimentalne analize provedene pomoću ostvarenog programskog rješenja.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je napraviti aplikaciju koja će omogućiti usporedbu različitih metoda za razvrstavanje oko središnje točke. Aplikaciji se moraju moći zadati objekti i broj grozdova te ona treba mjeriti vrijeme razvrstavanja. Naposljetku treba testirati aplikaciju koristeći različite skupove podataka te analizirati i interpretirati dobivene rezultate.

2. PREGLED PODRUČJA TEME

Razvrstavanje objekata se koristi u mnogim znanstvenim disciplinama. Neki primjeri su: istraživanje tržišta (podjela potrošača u razrede ovisno o njihovim kupovnim navikama), astronomija (klasificiranje svemirskih tijela), psihijatrija (razvrstavanje pacijenata na temelju simptoma), meteorologija (analiziranje vremenskih pojava), arheologija (klasifikacija pronađenih artefakata), bioinformatika i genetika (pronalaženje grupa gena sa sličnim svojstvima) i mnogi drugi [1]. Također, razvrstavanje podataka ima bitnu ulogu u postupku rudarenja podataka gdje je potrebno napraviti kompaktni sažetak velikog skupa podataka kako bi se na njemu radile daljnje analize [2].

U radu su opisane klasične verzije nekih od popularnijih algoritama za razvrstavanje oko središnje točke. Međutim, za svaki od tih algoritama predložene su i implementirane brojne nadogradnje koje pospješuju točnost njihovih rezultata i smanjuju njihovo vrijeme izvođenja [3][4][5]. Osim opisanih algoritama postoje i brojni drugi algoritmi za razvrstavanje oko središnje točke, primjerice algoritam EM (eng. *Expectation-maximization algorithm),* algoritam x-means, algoritam kd-trees, algoritam k-medians, algoritam BFR i drugi.

Usporedba metoda za razvrstavanje oko središnje točke međusobno te s drugim metodama je provedena u brojnim radovima kao što su npr. [6] i [7]. Pri usporedbi algoritama naglasak je na proučavanju vremena izvođenja algoritma, otpornosti algoritma na šum u podatcima, rada algoritma s visokodimenzionalnim podatcima te oblikom grozdova koje algoritam može prepoznati.

Primjer aplikacije slične programskom rješenju opisanom u radu nalazi se na [8]. Aplikacija omogućuje zadavanje broja grozdova nad odabranim skupom podataka te vizualizaciju i analizu dobivenih rezultata razvrstavanja. Aplikacija također omogućuje detekciju podataka koji ne pripadaju niti jednom grozdu te skaliranje podataka što nije pokriveno u radu.

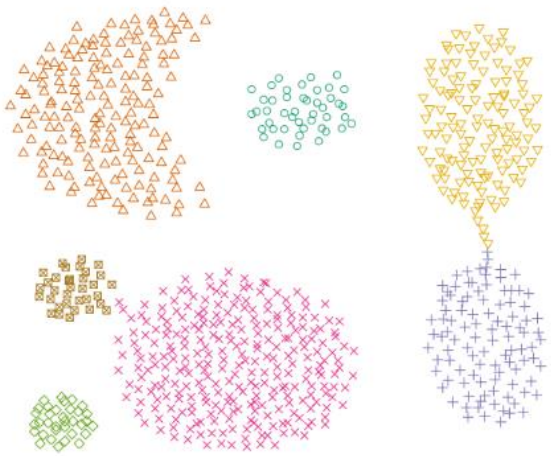
3. RAZVRSTAVANJE OBJEKATA

Razvrstavanje (eng. *clustering*) je metoda kreiranja grupa objekata, odnosno grozdova (eng. *clusters*) na način da su objekti unutar jednog grozda vrlo slični, a objekti unutar različitih grozdova znatno različiti [9]. Sličnost i različitost dvaju grozdova utvrđuju se pomoću mjera sličnosti i mjera različitosti (mjera udaljenosti). Kako bi se razvrstavanje uspješno provelo nad nekim skupom objekata, potrebno je definirati mjeru sličnosti i/ili različitosti između svakog para objekata u skupu.

Kriteriji određivanja kojem grozdu pripada pojedini objekt znatno se razlikuju ovisno o korištenoj metodi razvrstavanja te o svojstvima cjelokupnog skupa. Unatoč tome, postoje određeni kriteriji koje bi objekti unutar nekog grozda uvijek trebali zadovoljiti. Prema [10], svi objekti unutar jednog grozda bi trebali:

1. imati ista ili blisko povezana obilježja
2. biti malo udaljeni ili neznatno različiti
3. imati „kontakt“ ili „odnos“ s barem jednim drugim objektom u grozdu
4. biti vidljivo različiti od objekata u ostalih grozdovima

Veličina, oblik i gustoća grozda mogu znatno varirati te je na temelju svojstava grozdova potrebno odabrati odgovarajuću metodu razvrstavanja jer je svaka metoda prilagođena za razvrstavanje objekata u određen tip grozdova. Na slici 3.1. je vidljiv rezultat jednog razvrstanja skupa objekata u kojemu su dobiveni grozdovi različitih oblika, veličina i gustoća.



Slika 3.1. *Primjer razvrstanog skupa objekata* [11]

Problem razvrstavanja može se opisati matematički. Neka je zadan D skup od n objekata opisanih s d obilježja. Skup se tada može označiti s:

$$D = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x}_\mathrm{n}\} \tag{3-1}$$

gdje je $\boldsymbol{x_i}$ i-ti objekt skupa, odnosno:

$$\mathbf{x}_\mathrm{i} = (x_{i1}, x_{i2}, \dots, x_{id})^T \tag{3-2}$$

gdje je x_{ij} j-to obilježje i-tog vektora skupa. Broj obilježja d naziva se dimenzionalnost skupa [9]. Rezultat razvrstavanja skupa D u k grozdova je $k \times n$ matrica:

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k1} & u_{k2} & \cdots & u_{kn} \end{pmatrix} \tag{3-3}$$

gdje je n broj objekata u skupu, k broj grozdova, a u_{ji} pripadnost i-tog objekta j-tom grozdu. Ovisno o svojstvima matrice **U** razlikujemo čvrsto i neizrazito razvrstavanje. Kod čvrstog razvrstavanja svaki objekt pripada isključivo jednom grozdu. Pri tome vrijedi sljedeće:

$$u_{ji} \in \{0,1\}, \quad 1 \le j \le k, \quad 1 \le i \le n \tag{3-4}$$

$$\sum_{j=1}^{k} u_{ji} = 1\,, \quad 1 \le i \le n \tag{3-5}$$

$$\sum_{i=1}^{n} u_{ji} > 0, \quad 1 \le j \le k \tag{3-6}$$

Iz navedenih izraza je vidljivo da svaki objekt mora pripadati isključivo jednom grozdu što se označuje s vrijednošću $u_{ij} = 1$ dok se nepripadnost ostalim grozdovima označuje s vrijednošću 0. Također, svaki grozd mora sadržavati barem jedan objekt, odnosno ne smije postojati prazan grozd. Za razliku od čvrstog razvrstavanja, neizrazito razvrstavanje dopušta da jedan objekt istovremeno pripada većem broju grozdova, odnosno vrijedi:

$$u_{ji} \in [0,1], \quad 1 \le j \le k, \quad 1 \le i \le n \tag{3-7}$$

Jednadžba (3-7) pokazuje kako koeficijent pripadnosti objekta nekom grozdu kod neizrazitog razvrstavanja nije ograničen na 0 ili 1, tj. da objekti mogu pripadati većem broju grozdova s različitom vjerojatnošću pripadnosti. Kako matrica **U** kod neizrazitog razvrstavanja također

zadovoljava jednadžbe (3-5) i (3-6), osigurano je da zbroj vjerojatnosti pripadnosti jednog objekta svakom postojećem grozdu mora biti točno 1 te da ne smije postojati prazan grozd.

Razvrstavanje objekata se također može podijeliti na hijerarhijsko i particijsko. Hijerarhijsko razvrstavanje se može provoditi na dva načina. Prvi način započinje s pretpostavkom da je svaki objekt u zasebnom grozdu te zatim spaja najsličnije grozdove dok se ne dobije idealan broj grozdova (aglomeracijsko razvrstavanje) dok drugi način započinje s pretpostavkom da su svi objekti u jednom grozdu te zatim razdvaja objekte koji se najviše razlikuju u zasebne grozdove (razdvajajuće razvrstavanje). Za razliku od hijerarhijskog razvrstavanja, particijsko razvrstavanje odmah razdvaja početni skup objekata u k particija, odnosno grozdova, gdje je k unaprijed zadana vrijednost te se izvršavanjem algoritma nastoji postići što bolja raspodjela objekata u te grozdove. U ovome radu se razmatra isključivo particijsko razvrstavanje.

Razvrstavanje objekata je iterativan proces u kojem se u svakoj iteraciji nastoji poboljšati dosad postignut rezultat. Kako bi to bilo moguće, potrebno je definirati ciljnu funkciju koja će omogućiti usporedbu različitih rezultata grupiranja čime se cilj razvrstavanja svodi na pronalaženje ekstrema ciljne funkcije. Metoda računanja ciljne funkcije se razlikuje ovisno o korištenoj metodi razvrstavanja. Općenito se kao uvjet zaustavljanja procesa razvrstavanja uzima slučaj kada se ciljna funkcija ne mijenja između dviju uzastopnih iteracija ili slučaj kad je promjena ciljne funkcije između dviju uzastopnih iteracija manja od unaprijed zadane vrijednosti.

3.1. Mjere udaljenosti

Pri razvrstavanju objekata s brojčanim obilježjima potrebno je definirati kako će se računati udaljenosti između objekata. Izbor mjere udaljenosti ovisi o svojstvima skupa objekata i može znatno utjecati na rezultat razvrstavanja. U nastavku su opisane neke od najčešće korištenih mjera udaljenosti pri razvrstavanju objekata s brojčanim obilježjima.

3.1.1. Euklidska udaljenost

Euklidska udaljenost je najčešće korištena mjera udaljenosti u razvrstavanju objekata. Euklidska udaljenost dvaju d-dimenzionalnih objekata $\mathbf{x}$ i $\mathbf{y}$ računa se kao:

$$d_{euk}(\mathbf{x},\mathbf{y}) = \left[\sum_{j=1}^{d}\left(x_j - y_j\right)^2\right]^{\frac{1}{2}} = [(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T]^{\frac{1}{2}} \tag{3-8}$$

gdje su x_j i y_j j-ta obilježja objekata **x** i **y**. Uz euklidsku udaljenost, može se koristiti i kvadratna euklidska udaljenost koja se računa kao:

$$d_{kveuk}(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{d}\left(x_j - y_j\right)^2 = (\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T \tag{3-9}$$

U slučajevima kad je svejedno koristi li se euklidska ili kvadratna euklidska udaljenost efikasnije je koristiti kvadratnu euklidsku udaljenost jer se izbjegava računanje korijena [12].

3.1.2. Manhattan udaljenost

Manhattan udaljenost je također poznata kao udaljenost gradskih blokova jer se često koristi za računanje udaljenosti unutar gradova, gdje nije moguće primijeniti euklidsku udaljenost. Manhattan udaljenost dvaju d-dimenzionalnih objekata **x** i **y** računa se kao:

$$d_{man}(\mathbf{x},\mathbf{y}) = \sum_{j=1}^{d}\left|x_j - y_j\right| \tag{3-10}$$

3.1.3. Čebiševljeva udaljenost

Čebiševljeva ili maksimalna udaljenost definirana je kao maksimalna vrijednost udaljenosti dvaju atributa objekata **x** i **y**. Računa se kao:

$$d_{max}(\mathbf{x},\mathbf{y}) = \max_{1\le j\le d}\left|x_j - y_j\right| \tag{3-11}$$

3.1.4. Minkowskijeva udaljenost

Minkowskijeva udaljenost predstavlja generalizaciju prethodno opisanih mjera udaljenosti. Minkowskijeva udaljenost dvaju d-dimenzionalnih objekata **x** i **y** računa se kao:

$$d_{min}(\mathbf{x},\mathbf{y}) = \left[\sum_{j=1}^{d}\left(x_j - y_j\right)^r\right]^{\frac{1}{r}},\ \ r \ge 1 \tag{3-12}$$

Može se primijetiti kako uvrštavanje vrijednosti 1, 2 i ∞ za parametar r redom daje jednadžbe za Manhattan, euklidsku i Čebiševljevu udaljenost. Minkowskijeva udaljenost funkcionira dobro kada su grozdovi kompaktni ili izolirani, dok u suprotnome atributi s većim vrijednostima dominiraju nad ostalim atributima te je potrebno skalirati atribute kako bi se to izbjeglo [9].

3.2. Razvrstavanje oko središnje točke

Metode razvrstavanja oko središnje točke definiraju svaki grozd preko jednog središnjeg objekta, često nazivanog centroidom, koji ne mora biti dio originalnog skupa te ostale objekte pridružuju grozdovima čijem su centroidu najsličniji. Kako je riječ o particijskom grupiranju, potrebno je unaprijed definirati broj grozdova k i ciljnu funkciju koja mjeri uspješnost razvrstavanja. Metode razvrstavanja oko središnje točke su vrlo učinkovite u razvrstavanju vrlo velikih skupova objekata i visokodimenzionalnih objekata. Međutim, kao rezultat daju grozdove konveksnih oblika što ih čini neprikladnima za razvrstavanje skupova objekata koji prirodno tvore grozdove nepravilnih oblika [9]. U nastavku su opisani neki od poznatijih algoritama za razvrstavanje oko srednje točke.

3.2.1. Algoritam k-means

Algoritam k-means je najšire korišten particijski algoritam za čvrsto razvrstavanje objekata. Cilj algoritma je minimizacija ciljne funkcije opisane jednadžbom (3-13).

$$E = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c_i}\|^2 \tag{3-13}$$

gdje je k broj grozdova i $\mathbf{c_i}$ centroid i-tog grozda C_i. Ciljna funkcija dana jednadžbom (3-13) naziva se sumom kvadratnih pogrešaka ili rezidualnom sumom kvadrata [2]. Centroidi se računaju na način opisan jednadžbom (3-14).

$$\mathbf{c_i} = \frac{\sum_{\mathbf{x} \in C_i} \mathbf{x}}{|C_i|} \tag{3-14}$$

gdje je $\mathbf{x}$ objekt koji pripada grozdu C_i, a $|C_i|$ broj objekata unutar i-tog grozda.

Algoritam k-means je iterativan algoritam te se sastoji od sljedećih koraka:

1. Iz skupa objekata $D = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_n}\}$ nasumično odaberi k objekata koji će biti početni centroidi. k se zadaje unaprijed i mora vrijediti $k \leq n$.
2. Pridruži svaki objekt iz D grozdu čijem je centroidu najbliži.
3. Koristeći jednadžbu (3-14) izračunaj nove vrijednosti centroida svakog grozda.
4. Ako je zadovoljen uvjet konvergencije završi algoritam, inače ponovi korake 2 i 3.

Kao uvjet konvergencije uzima se slučaj kada se nijedan centroid nije promijenio u koraku 3 ili slučaj kada je broj iteracija izvođenja algoritama dosegao prethodno zadanu vrijednost. Za

računanje udaljenosti može se uzeti proizvoljna mjera udaljenosti, no najčešće se koristi euklidska udaljenost [2].

Vremenska složenost algoritma je $O(inkd)$, gdje je i broj iteracija, n broj objekata, k broj grozdova i d dimenzionalnost objekata. Iako je algoritam relativno brz i jednostavan za implementirati, ima nekolicinu slabosti:

1. Dobiveni grozdovi su isključivo konveksnog oblika što ga čini nepogodnim za otkrivanje grozdova nepravilnih oblika.
2. Često se zaustavlja pri pronalasku lokalnog minimuma ciljne funkcije što daje neoptimalne rezultate razvrstavanja.
3. Rezultati razvrstavanja su uvelike ovisni o početnom odabiru centroida.
4. Ne radi dobro s visokodimenzionalnim podatcima [13].
5. Ograničen je na brojčane podatke.
6. Centroidi su osjetljivi na objekte unutar grozda koji znatno odstupaju od ostalih objekata unutar grozda.

Kako početni odabir centroida ima znatan utjecaj na konačne rezultate izvođenja algoritma, razvijeni su brojni algoritmi koji predlažu bolji način početnog odabira centroida. Jedan od takvih algoritama je k-means++ koji je opisan u nastavku.

3.2.2. Algoritam k-means++

Algoritam k-means++ predstavlja poboljšanje algoritma k-means, odnosno poboljšanje metode početnog odabira centroida. Za razliku od algoritma k-means koji početne centroide odabire nasumično, postupak početnog odabira centroida kod algoritma k-means++ je sljedeći: [14]

1. Iz skupa objekata $D = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x}_\mathrm{n}\}$ nasumično odaberi jedan objekt koji će predstavljati centroid prvog grozda.
2. Neka je $d(\mathbf{x})$ udaljenost objekta $\mathbf{x}$ iz skupa D do njemu najbližeg dosad odabranog centroida. Odaberi novi centroid birajući $\mathbf{x_j} \in D$ s vjerojatnošću opisanom jednadžbom (3-15).

$$P(\mathbf{c_i} = \mathbf{x_j}) = \frac{d^2(\mathbf{x_j})}{\sum_{\mathbf{x} \in D} d^2(\mathbf{x})} \tag{3-15}$$

gdje je $\mathbf{c_i}$ centroid i-tog grozda, odnosno centroid koji se trenutno određuje.

3. Ako je odabrano k centroida, završi postupak inicijalizacije i nastavi s običnim k-means algoritmom. U suprotnome, ponovi korak 2.

Kao što je vidljivo iz jednadžbe (3-15), algoritam pri odabiru novog centroida najveću vjerojatnost odabira daje objektima koji su najviše udaljeni od njima najbližeg centroida. Iako je postupak početnog odabira centroida sporiji nego kod algoritma k-means, ostatak algoritma u većini slučajeva konvergira znatno brže te je ukupno vrijeme izvođenja algoritma k-means++ često kraće od vremena izvođenja algoritma k-means. Također, k-means++ znatno poboljšava krajnji rezultat razvrstavanja objekata jer sofisticiraniji početni odabir centroida smanjuje vjerojatnost ranog zaustavljanja algoritma na lokalnom minimumu ciljne funkcije.

3.2.3. Algoritam k-medoids

Algoritam k-medoids predstavlja jednu od popularnijih varijacija algoritma k-means. Glavna razlika između tih dvaju algoritama je metoda izbora središta grozdova. Naime, dok algoritam k-means definira središte grozda kao aritmetičku sredinu svih objekata u grozdu, algoritam k-medoids bira središta grozdova isključivo iz zadanog skupa objekata. Takva središta grozdova nazivaju se medoidi. Takav način odabira središta grozdova čini algoritam k-medoids otpornijim na objekte koji znatno odstupaju od ostalih (eng. *outliers*) [2]. Algoritam minimizira ciljnu funkciju opisanu jednadžbom (3-16).

$$E = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{c_i}) \tag{3-16}$$

gdje je k broj grozdova, $\mathbf{c_i}$ medoid i-tog grozda, a $d(\mathbf{x}, \mathbf{c_i})$ proizvoljna mjera udaljenosti između objekta i medoida grozda u kojem se nalazi.

Jedna od najčešćih implementacija algoritma k-medoids je algoritam PAM (eng. *partitioning around medoids*). Koraci algoritma PAM su sljedeći:

1. Iz skupa objekata $D = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_n}\}$ odaberi k objekata koji će biti početni medoidi. k se zadaje unaprijed i mora vrijediti $k \leq n$.
2. Pridruži svaki objekt iz D grozdu čijem je medoidu najbliži na temelju proizvoljno odabrane mjere udaljenosti.
3. Za svaki grozd promatraj svaki objekt u skupu D (osim medoida grozda) te izračunaj koliko bi se promijenila ciljna funkcija kad bi taj objekt postao novi medoid grozda. Napravi

zamjenu objekta i medoida koja rezultira najvećim smanjenjem ciljne funkcije. Ako ne postoji zamjena koja smanjuje ciljnu funkciju, ne mijenjaj postojeće medoide.

4. Ako je zadovoljen uvjet konvergencije završi algoritam, inače ponovi korake 2 i 3.

Kao uvjet konvergencije uzima se slučaj kada se nijedan medoid nije promijenio u koraku 3 ili slučaj kada je broj iteracija izvođenja algoritama dosegao prethodno zadanu vrijednost.

Vremenska složenost algoritma je $O(ik(n-k)^2)$, gdje je i broj iteracija, n broj objekata i k broj grozdova, što ga čini nepovoljnim za korištenje na velikim skupovima objekata [15]. Kao i kod algoritma k-means, početni izbor medoida može utjecati na rezultate razvrstavanja i na trajanje izvođenja algoritma te se umjesto nasumičnog izbora preporuča korištenje neke naprednije metode kao što je k-means++ inicijalizacija. Također, algoritam pronalazi isključivo grozdove konveksnog oblika.

3.2.4. Algoritam fuzzy c-means

Algoritam fuzzy c-means (FCM) je particijski algoritam za neizrazito razvrstavanje. Za razliku od prethodno opisanih algoritama, FCM razvrstava objekte u više od jednog grozda tako što im dodjeljuje vrijednost pripadnosti grozdu $u_{ij} \in [0,1]$. Pri tome vrijede jednadžbe (3-5) i (3-6) koje osiguravaju da suma pripadnosti jednog objekta svim grozdovima mora biti jednaka 1 te da ne smije biti praznih grozdova. Što je vrijednost u_{ij} bliža 1, to je veća vjerojatnost da j-ti objekt pripada i-tom grozdu. Korištenje algoritma FCM je korisno kada se žele otkriti potencijalna preklapanja između grozdova. Algoritam nastoji minimizirati ciljnu funkciju opisanu jednadžbom (3-17).

$$E = \sum_{j=1}^{n} \sum_{k=1}^{i} u_{ij}^{m} \left\| \mathbf{x_j} - \mathbf{c_i} \right\|_{\mathbf{A}}^{2} \tag{3-17}$$

gdje je u_{ij} pripadnost j-tog objekta i-tom grozdu, m koeficijent neizrazitosti, k broj grozdova, n broj objekata, $\mathbf{c_i}$ centroid i-tog grozda, a $\left\| \mathbf{x_j} - \mathbf{c_i} \right\|_{\mathbf{A}}^{2}$ kvadratna udaljenost između j-tog objekta i centroida i-tog grozda u A-normi čiji je izračun opisan jednadžbom (3-18).

$$D_{ij\mathbf{A}}^{2} = \left\| \mathbf{x_j} - \mathbf{c_i} \right\|_{\mathrm{A}}^{2} = \left(\mathbf{x_j} - \mathbf{c_i} \right)^{T} \mathbf{A} \left(\mathbf{x_j} - \mathbf{c_i} \right) \tag{3-18}$$

Izbor matrice $\mathbf{A}$ određuje oblik grozdova koji će se dobiti izvođenjem algoritma. Na primjer, ako se uzme $\mathbf{A} = \mathbf{I}$, tada će udaljenost biti izračunata u euklidskoj normi te će algoritam pronalaziti

grozdove hipersfernog oblika [16]. Formule za izračun koeficijenta pripadnosti i centroida grozdova dane su jednadžbama (3-19) i (3-20).

$$u_{ij} = \sum_{l=1}^{k} \left(\left(\frac{D_{ij\mathbf{A}}}{D_{lj\mathbf{A}}} \right)^{\frac{2}{m-1}} \right)^{-1} \tag{3-19}$$

$$\mathbf{c_i} = \frac{\sum_{j=1}^{n} u_{ij}^{m} \mathbf{x_j}}{\sum_{j=1}^{n} u_{ij}^{m}} \tag{3-20}$$

FCM je iterativan algoritam te se sastoji od sljedećih koraka:

1. Nasumično inicijaliziraj matricu $\mathbf{U} = (u_{ij})$, $1 \leq i \leq k$, $1 \leq j \leq n$, gdje je k unaprijed zadan broj grozdova i n broj objekata u skupu D.
2. Izračunaj centroide grozdova koristeći jednadžbu (3-20).
3. Izračunaj koeficijente pripadnosti u_{ij} koristeći jednadžbu (3-19).
4. Usporedi $\mathbf{U}^{(t+1)}$ i $\mathbf{U}^{(t)}$, gdje je t broj iteracije.
5. Ako vrijedi $\left\| \mathbf{U}^{(t+1)} - \mathbf{U}^{(t)} \right\| < \varepsilon$ ili je dosegnut maksimalni broj iteracija završi algoritam. U suprotnome, vrati se na korak 2.

Vrijednost ε zadaje se prije izvođenja algoritma. Za usporedbu matrice $\mathbf{U}$ u dvama uzastopnim iteracijama može se koristiti proizvoljna matrična norma [16]. Vremenska složenost algoritma je $O(ink^2d)$, gdje je i broj iteracija, n broj objekata, k broj grozdova i d dimenzionalnost objekata.

Koeficijent neizrazitosti m ima značajan utjecaj na krajnji rezultat razvrstavanja algoritmom FCM. Naime, riječ je o realnom koeficijentu koji može poprimati vrijednosti iz intervala $[1, \infty\rangle$. Za vrijednost $m = 1$ dobiva se čvrsto razvrstavanje, dok rastom vrijednosti koeficijenta m granice između grozdova postaju sve manje definirane te se dobivaju sve veća preklapanja između grozdova. Za većinu podataka vrijednost $m \in [1.5,3]$ daje dobre rezultate [16]. Kao i prethodno opisani algoritmi, FCM ne radi dobro s visokodimenzionalnim podatcima, pogotovo u slučaju neprikladno odabranog koeficijenta m. Jedan od prijedloga izbjegavanja navedenog problema je određivanje koeficijenta m po jednadžbi (3-21).

$$m = 1 + \frac{2}{d} \tag{3-21}$$

gdje je d dimenzionalnost podataka. Pri radu s visokodimenzionalnim podacima m će biti sve bliži vrijednosti 1 te će neizrazitost dobivenih grozdova biti sve manja. Međutim, ako je željeni rezultat

razvrstavanje podataka neovisno o njihovoj neizrazitosti, tada je određivanje koeficijenta m pomoću jednadžbe (3-21) dobar izbor [17].

3.2.5. Algoritam uk-means

Algoritam uk-means koristi se za razvrstavanje objekata koji sadrže nesigurnost. Nesigurnost objekata može biti uzrokovana brojnim razlozima kao što su pogreške u mjerenju, pogreške u uzorkovanju, zastarjeli izvori podataka i druge pogreške [18]. Primjer situacije u kojoj se dobivaju objekti s nesigurnošću je praćenje lokacije vozila. Zbog vremenskog kašnjenja i mogućih pogrešaka u mjerenju, točna lokacija vozila u određenom trenutku ne može biti jednoznačno određena, već se mora procijeniti s određenom razinom nesigurnosti. Iz tog razloga se objekti s nesigurnošću ne predstavljaju jednom točkom u prostoru, nego se predstavljaju minimalnim područjem nesigurnosti koje se zadaje odgovarajućom funkcijom gustoće vjerojatnosti [19].

Algoritam uk-means minimizira očekivanu sumu kvadratnih pogrešaka $E(SSE)$ opisanu jednadžbom (3-22) [18].

$$E\left(\sum_{j=1}^{k}\sum_{i\in C_j}\left\|\mathbf{c_j}-\mathbf{x_i}\right\|^2\right)=\sum_{j=1}^{k}\sum_{i\in C_j}\int\left\|\mathbf{c_j}-\mathbf{x_i}\right\|^2 f(\mathbf{x_i})d\mathbf{x_i} \tag{3-22}$$

gdje je $\boldsymbol{c_j}$ središte j-tog grozda, $\mathbf{x_i}$ i-ti objekt iz promatranog skupa objekata, C_j j-ti grozd, k broj grozdova, $f(\mathbf{x_i})$ funkcija gustoće vjerojatnosti i-tog objekta i $\left\|\mathbf{c_j}-\mathbf{x_i}\right\|^2$ kvadratna udaljenost središta j-tog grozda i i-tog objekta. Središta grozdova računaju se prema jednadžbi (3-23).

$$\mathbf{c_j}=E\left(\frac{1}{|C_j|}\sum_{i\in C_j}\mathbf{x_i}\right)=\frac{1}{|C_j|}\sum_{i\in C_j}\int\mathbf{x_i}f(\mathbf{x_i})d\mathbf{x_i} \tag{3-23}$$

gdje je $|C_j|$ broj objekata unutar j-tog grozda.

Koraci algoritma uk-means su sljedeći:

1. Odredi početne vrijednost središta svih k grozdova.
2. Pridruži svaki objekt $\mathbf{x_i}$ grozdu C_j za koji je $E(\left\|\mathbf{c_j}-\mathbf{x_i}\right\|)$ minimalan.
3. Izračunaj nova središta grozdova koristeći jednadžbu (3-23).
4. Ako je zadovoljen uvjet konvergencije završi algoritam, inače ponovi korake 2 i 3.

Proces izračuna $E\left(\left\|\mathbf{c_j} - \mathbf{x_i}\right\|\right)$ u 3. koraku je često zahtjevan. Različiti oblici područja nesigurnosti i funkcije gustoće vjerojatnosti mogu zahtijevati korištenje numeričkih metoda integracije te se zbog jednostavnosti preporučuje korištenje kvadratne očekivane udaljenosti $E\left(\left\|\mathbf{c_j} - \mathbf{x_i}\right\|^2\right)$ [18]. Navedeni algoritam je primjenjiv za svaki oblik područja nesigurnosti i funkcije gustoće vjerojatnosti [18].

3.3. Mjere kvalitete razvrstavanja

Kako bi usporedba dvaju algoritama za razvrstavanje bila moguća, potrebno je definirati mjere koje će govoriti koliko su kvalitetno objekti razvrstani u grozdove. Pri tome je bitno znati jesu li pravi, odnosno očekivani rezultati grupiranja poznati. U slučaju da jesu, ocjena kvalitete razvrstavanja provodi se na temelju usporedbe dobivenih i očekivanih rezultata koristeći tzv. ekstrinzične mjere. U suprotnome je potrebno promatrati koliko su dobiveni grozdovi kompaktni i dobro razdvojeni koristeći tzv. intrinzične mjere. Kako je razvrstavanje objekata najčešće korišteno kao oblik nenadziranog učenja, češće se koriste intrinzične mjere. U nastavku su opisane neke od poznatijih mjera kvalitete razvrstavanja objekata. Sve opisane mjere primjenjuju se na rezultate čvrstog grupiranja, no mogu se primijeniti i na rezultate neizrazitog grupiranja ukoliko svaki objekt pridružimo isključivo grozdu za koji mu je koeficijent pripadnosti u_{ij} najveći.

3.3.1. Calinski-Harabaszov indeks

Calinski-Harabaszov (CH) indeks je intrinzična mjera kvalitete razvrstavanja. Izračun CH indeksa opisan je jednadžbom (3-24).

$$CH = \frac{SS_B}{SS_W} \times \frac{n-k}{k-1} \tag{3-24}$$

gdje je n broj objekata, a k broj grozdova. SS_B je ukupna varijanca između grozdova koja se računa prema jednadžbi (3-25).

$$SS_B = \sum_{i=1}^{k} |C_i| \|\mathbf{c_i} - \mathbf{m}\|^2 \tag{3-25}$$

gdje je $|C_i|$ broj elemenata unutar i-tog grozda, $\mathbf{m}$ aritmetička sredina svih objekata u skupu, a $\|\mathbf{c_i} - \mathbf{m}\|^2$ kvadratna euklidska udaljenost središta i-tog grozda od objekta $\mathbf{m}$. SS_W u jednadžbi (3-24) predstavlja ukupnu varijancu objekata unutar grozda čiji je izračun opisan jednadžbom (3-26).

$$SS_W = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c_i}\|^2 \tag{3-26}$$

gdje je C_i i-ti grozd, a $\mathbf{c}_\mathrm{i}$ njegovo središte.

Što su grozdovi gušći i bolje razdvojeni, to je CH indeks veći. Prednost korištenja CH indeksa je njegov brz izračun. Međutim, CH indeks je općenito viši za grozdove konveksnog oblika što može dati krivu ocjenu kvalitete razvrstavanja ako su grozdovi nepravilnih oblika [20].

3.3.2. Davies-Bouldinov indeks

Davies-Bouldinov (DB) indeks je intrinzična mjera kvalitete razvrstavanja. Predstavlja prosječnu sličnost između svakog grozda $C_i, i = 1, \dots, k$ i njemu najsličnijeg grozda C_j [20]. Izračun DB indeksa opisan je jednadžbom (3-27).

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij} \tag{3-27}$$

gdje je k broj grozdova, a R_{ij} mjera sličnosti i-tog i j-tog grozda opisana jednadžbom (3-28).

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{3-28}$$

gdje je s_i prosječna udaljenost svakog objekta i-tog grozda od njegovog središta, a d_{ij} udaljenost između središta i-tog i j-tog grozda.

Što su grozdovi bolje razdvojeni, to je DB indeks manji. Najmanja moguća vrijednost DB indeksa je 0. Kao i CH indeks, DB indeks općenito daje bolje rezultate za grozdove konveksnog oblika [20].

3.3.3. Koeficijent siluete

Koeficijent siluete je intrinzična mjera kvalitete razvrstavanja. Računa se pojedinačno za svaki objekt u skupu, a za koeficijent siluete cijelog skupa uzima se aritmetička sredina dobivenih koeficijenata siluete svakog objekta u skupu. Izračun koeficijenta siluete jednog objekta skupa opisan je jednadžbom (3-29).

$$s = \frac{b - a}{\max(a, b)} \tag{3-29}$$

gdje je a prosječna udaljenost objekta od svih ostalih objekata u njegovom grozdu, a b prosječna udaljenost objekta od svih objekata u najbližem grozdu (ne brojeći onaj u kojem se nalazi). Najbliži grozd se pronalazi na temelju prosječne udaljenosti objekta od svih objekata unutar određenog grozda.

Koeficijent siluete može poprimiti vrijednosti između -1 i 1, pri čemu veća vrijednost označuje kvalitetnije razvrstavanje [21]. Kako je za izračun koeficijenta siluete cijelog skupa potrebno izračunati koeficijent siluete svakog objekta u skupu, računanje koeficijenta siluete nije prikladno za vrlo velike skupove objekata. Također, kao i prethodno navedene mjere, koeficijent siluete daje bolje rezultate za grozdove konveksnog oblika [20].

3.3.4. Randov indeks

Randov indeks je ekstrinzična mjera kvalitete razvrstavanja te se može koristiti za direktnu usporedbu dvaju dobivenih rezultata razvrstavanja ili za usporedbu rezultata razvrstavanja sa stvarnim vrijednostima ako su one poznate.

Ako je zadan skup objekata $D = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x}_\mathrm{n}\}$ i ako su $P = \{C_1, C_2, \dots, C_{k_1}\}$ i $P' = \{C'_1, C'_2, \dots, C'_{k_2}\}$ dva rezultata razvrstavanja skupa D u k_1, odnosno k_2 grozdova, tada je izračun Randovog indeksa opisan jednadžbom (3-30) [9].

$$R = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}} \tag{3-30}$$

gdje je n broj objekata i:

- a broj parova objekata koji se nalaze u istom grozdu u P i u istom grozdu u P'
- b broj parova objekata koji se nalaze u različitim grozdovima u P i u različitim grozdovima u P'
- c broj parova objekata koji se nalaze u istom grozdu u P i u različitim grozdovima u P'
- d broj parova objekata koji se nalaze u različitim grozdovima u P i u istom grozdu u P'

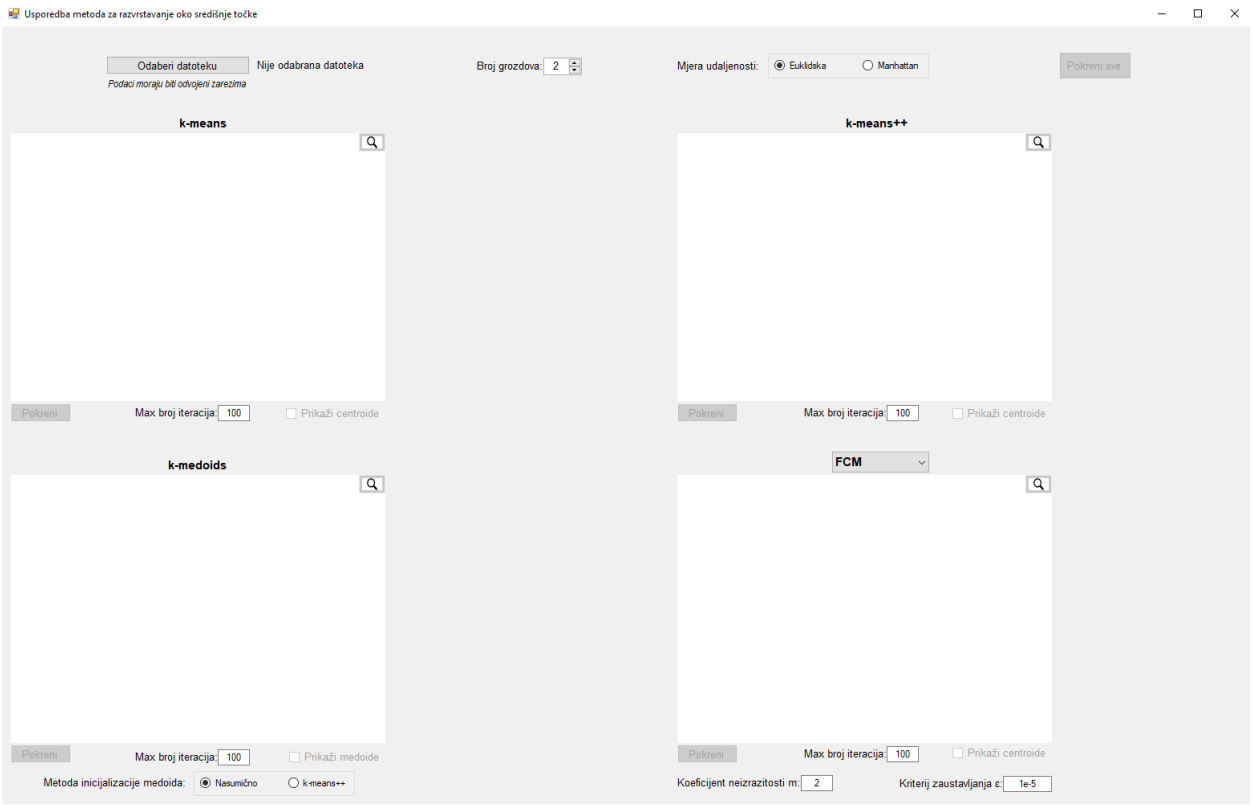
Kako nazivnik u jednadžbi (3-30) sadrži sve moguće kombinacije parova objekata u skupu D, on se jednostavnije može zapisati kao $\binom{n}{2}$.

Randov indeks može poprimiti vrijednosti između 0 i 1 pri čemu viša vrijednost ukazuje na veću sličnost između rezultata grupiranja. Glavni nedostatak Randovog indeksa je njegovo zanemarivanje mogućnosti da se preklapanja rezultata dogode slučajno što predstavlja problem pri

analizi rezultata razvrstavanja u kojemu se broj grozdova i broj objekata ne razlikuje znatno. Iz tog razloga se često koristi prilagođeni Randov indeks [2].

4. OSTVARENO PROGRAMSKO RJEŠENJE

Ostvareno programsko rješenje izrađeno je u programskom jeziku C# koristeći razvojno okruženje Microsoft Visual Studio 2017, a za dizajn sučelja programskog rješenja korištene su Windows forme.



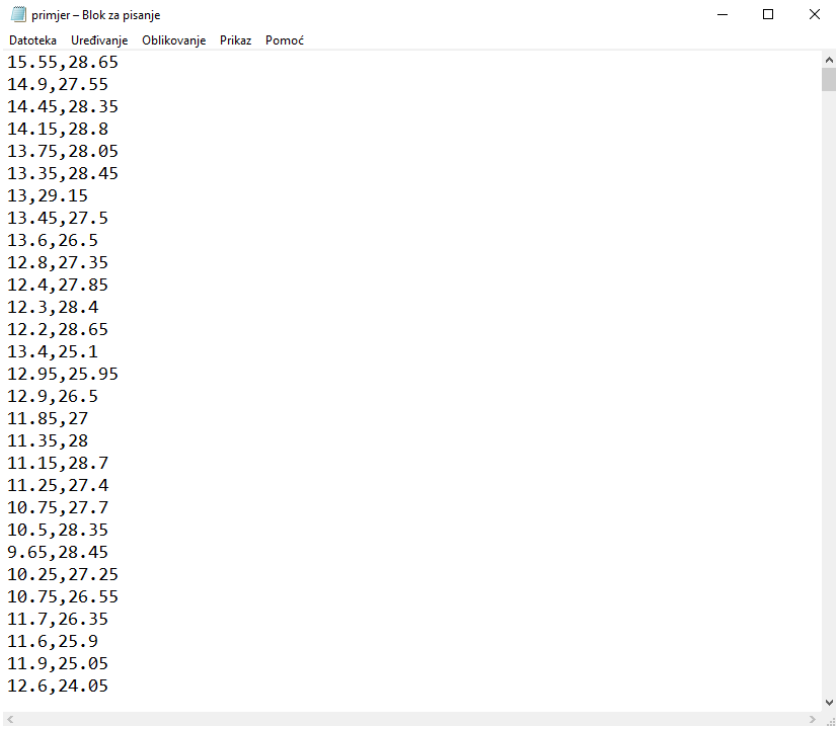
Slika 4.1. *Grafičko sučelje programskog rješenja*

Na slici 4.1. je prikazano grafičko sučelje programskog rješenja. Kako su implementirane 4 metode za razvrstavanje objekata, sučelje je podijeljeno u 4 dijela kako bi se rezultati razvrstavanja koristeći različite metode mogli lako uspoređivati. Dok su grafikoni za k-means, k-means++ i k-medoids strogo vezani uz te metode, donji desni grafikon omogućuje biranje bilo koje od implementiranih metoda kako bi se omogućio ispravan rad aplikacije u slučaju dodavanja dodatnih metoda.

4.1. Učitavanje objekata

Da bi korisniku bilo omogućeno pokrenuti izvođenje metoda za razvrstavanje, potrebno je učitati objekte iz datoteke. To je omogućeno putem gumba „Odaberi datoteku“ koji otvara dijalog za odabir datoteke iz koje će se učitati objekti za razvrstavanje. Korisniku je omogućen odabir *.txt* i *.csv* datoteka pri čemu se očekuje da će podatci biti odvojeni zarezima te da će svaki objekt biti

u zasebnom retku. Nakon što korisnik odabere datoteku, provjerava se ispravnost učitanih podataka, odnosno ima li u datoteci barem jedan objekt, jesu li svi objekti jednake dimenzionalnosti te jesu li svi objekti isključivo brojčani. U slučaju da neki od uvjeta nije zadovoljen ispisuje se odgovarajuća poruka o grešci. Na slici 4.2. je prikazan primjer ispravno formatirane datoteke.



Slika 4.2. *Primjer ispravno formatirane datoteke s objektima*

4.2. Implementirane metode

U programskom rješenju implementirane su 4 metode za razvrstavanje objekata oko središnje točke: k-means, k-means++, k-medoids i fuzzy c-means. Omogućen je rad isključivo s brojčanim podatcima. Za svaku od navedenih metoda korisniku je omogućen odabir maksimalnog broja iteracija (nužno veći od 0) kao i broja grozdova (nužno veći od 0 i manji ili jednak broju objekata, maksimalna vrijednost 50). Ako je unesen maksimalan broj iteracija ili broj grozdova neispravan ispisuje se poruka o pogrešci te se ne započinje izvršavanje algoritma. Klikom na tipku „Pokreni“ ispod određenog grafikona pokreće se izvršavanje odgovarajuće metode, dok se klikom na tipku „Pokreni sve“ simultano pokreću sve metode. Nije moguće ponovno pokrenuti metodu dok nije završilo njezino trenutno izvođenje. Korisniku je također omogućen odabir euklidske ili Manhattan udaljenosti za mjeru udaljenosti između objekata.

Algoritam k-means implementiran je na način opisan u poglavlju 3.2.1. Za odabir početnih centroida korištena je metoda odabira *k* nasumičnih objekata, gdje je *k* broj grozdova. Kao mjera

udaljenosti koristi se ona mjera koju je korisnik odabrao prije pokretanja algoritma, a centroid grozda se računa kao aritmetička sredina svih objekata u grozdu. Algoritam se zaustavlja ako je broj iteracija dosegao prethodno zadanu maksimalnu vrijednost ili ako nijedan objekt nije promijenio grozd u trenutnoj iteraciji.

Algoritam k-means++ implementiran je na isti način kao i algoritam k-means osim u postupku odabira početnih centroida. Taj postupak je implementiran na temelju jednadžbe (3-15), a metoda za odabir početnih centroida prikazana je na slici 4.3.

```
public Cluster[] initializeClusterCentres(Point[] points, int clusterCount)
{
    Cluster[] clusters = new Cluster[clusterCount];
    Random rnd = new Random();
    Point firstCentroid = points[rnd.Next(points.Length)];
    clusters[0] = new Cluster(0, new Point(firstCentroid));
    for (int i = 1; i < clusterCount; i++)
    {
        double[] minSquaredDistances = getSmallestSquaredDistances(points, clusters, i);
        double minSquaredDistanceSum = getSumOfMinSquaredDistances(minSquaredDistances);
        double seed = rnd.NextDouble() * minSquaredDistanceSum;
        double sum = 0;
        for (int j = 0; j < minSquaredDistances.Length; j++)
        {
            sum += minSquaredDistances[j];
            if (seed < sum)
            {
                Point newCentroid = points[j];
                clusters[i] = new Cluster(i, new Point(newCentroid));
                break;
            }
        }
    }
    return clusters;
}
```

Slika 4.3. *Metoda za odabir početnih centroida u algoritmu k-means++*

Klasa *Cluster* predstavlja jedan grozd te je definirana jedinstvenim indeksom i trenutnim centroidom, a klasa *Point* predstavlja n-dimenzionalnu točku, odnosno objekt iz skupa. Funkcija *getSmallestSquaredDistances()* vraća polje u kojem i-ti element predstavlja kvadratnu udaljenost i-tog objekta od njemu najbližeg centroida, a funkcija *getSumOfMinSquaredDistances()* vraća sumu kvadratnih udaljenosti svih objekata u skupu od njima najbližeg centroida.

Algoritam k-medoids implementiran je po uzoru na PAM algoritam opisan u poglavlju 3.2.3. Pri pokretanju algoritma korisniku je omogućen i odabir metode odabira početnih medoida (nasumično ili k-means++ algoritmom inicijalizacije). Korak pronalaženja novih medoida i njihove zamjene s dosadašnjim medoidima implementiran je po uzoru na algoritam opisan u [22,

str. 7] koji poboljšava vremensku složenost algoritma PAM za faktor $O(k)$, gdje je k broj grozdova. Implementacija navedene metode prikazana je na slici 4.4.

```
private bool swap()
{
    double bestCostChange = 0;
    int candidatePoint = 0, candidateCluster = 0;
    for (int i = 0; i < points.Length; i++)
    {
        if (isMedoid(points[i])) continue;
        double[] clusterCostChanges = new double[clusterCount];
        for (int j = 0; j < clusterCount; j++)
        {
            clusterCostChanges[j] = -1 * distancesToNearestMedoid[i];
        }
        for (int j = 0; j < points.Length; j++)
        {
            if (i == j) continue;
            double distance = distances[i][j];
            double distanceToNearestMedoid = distancesToNearestMedoid[j], distanceToSecondNearestMedoid = distancesToSecondNearestMedoid[j];
            clusterCostChanges[results[j]] += Math.Min(distance, distanceToSecondNearestMedoid) - distanceToNearestMedoid;
            if (distance < distanceToNearestMedoid)
            {
                for (int k = 0; k < clusterCount; k++)
                {
                    if (k == results[j]) continue;
                    clusterCostChanges[k] += distance - distanceToNearestMedoid;
                }
            }
        }
        double minCostChange = clusterCostChanges[0];
        int minCostChangeCluster = 0;
        for (int j = 1; j < clusterCount; j++)
        {
            if (clusterCostChanges[j] < minCostChange)
            {
                minCostChange = clusterCostChanges[j];
                minCostChangeCluster = j;
            }
        }
        if (minCostChange < bestCostChange)
        {
            bestCostChange = minCostChange;
            candidatePoint = i;
            candidateCluster = minCostChangeCluster;
        }
    }
    if (bestCostChange < 0)
    {
        clusters[candidateCluster].centroid = points[candidatePoint];
        return true;
    }
    return false;
}
```

Slika 4.4. *Metoda za zamjenu medoida u algoritmu k-medoids*

Metoda *swap()* traži zamjenu postojećeg medoida s objektom koji nije medoid koja će rezultirati najvećim smanjenjem ciljne funkcije opisane jednadžbom (3-16). To postiže tako što za svaki objekt koji nije medoid razmatra koliko bi se promijenila ciljna funkcija ako bi taj objekt postao medoidom određenog grozda te pamti za koji grozd bi nastalo najveće smanjenje ciljne funkcije. Nakon što prođe kroz sve objekte izvršava se zamjena za koju je utvrđeno da će dovesti do najvećeg smanjenja ciljne funkcije, a ako nijedna zamjena ne rezultira smanjenjem ciljne funkcije, prekida se izvršavanje algoritma k-medoids. Kako bi se smanjilo vrijeme izvršavanja funkcije *swap()*, potrebno je unaprijed izračunati i pohraniti udaljenosti između objekata te udaljenosti objekata od najbližeg i drugog najbližeg medoida. U metodi na slici 4.4. to su polja *distances*, *distancesToNearestMedoid* i *distancesToSecondNearestMedoid*.

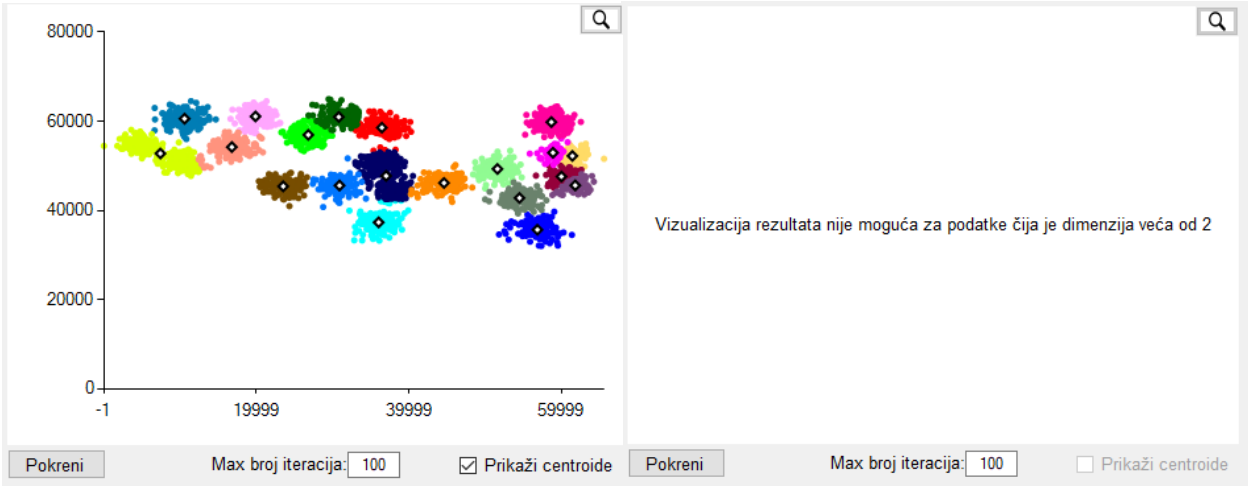
Algoritam fuzzy c-means implementiran je na način opisan u poglavlju 3.2.4. Pri pokretanju algoritma korisniku je omogućen odabir koeficijenta neizrazitosti m (nužno veći ili jednak 1) i kriterija zaustavljanja ε (nužno veći ili jednak 0). Ako je unesena vrijednost koeficijenta neizrazitosti ili kriterija zaustavljanja neispravna, ispisuje se poruka o pogrešci. Kao matrica **A** u jednadžbi (3-18) odabrana je jedinična matrica. Matrica pripadnosti **U** inicijalizira se nasumično, no pritom se vodi računa da vrijede jednadžbe (3-4), (3-5) i (3-6). Koeficijenti pripadnosti u_{ij} i centroidi grozdova računaju se prema jednadžbama (3-19) i (3-20), a kao mjera za usporedbu matrice **U** u uzastopnim iteracijama radi utvrđivanja konvergencije koristi se matrična ∞-norma opisana jednadžbom (4-1).

$$\|\mathbf{A}\|_{\infty} = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}| \tag{4-1}$$

gdje je m broj redaka, a n broj stupaca matrice.

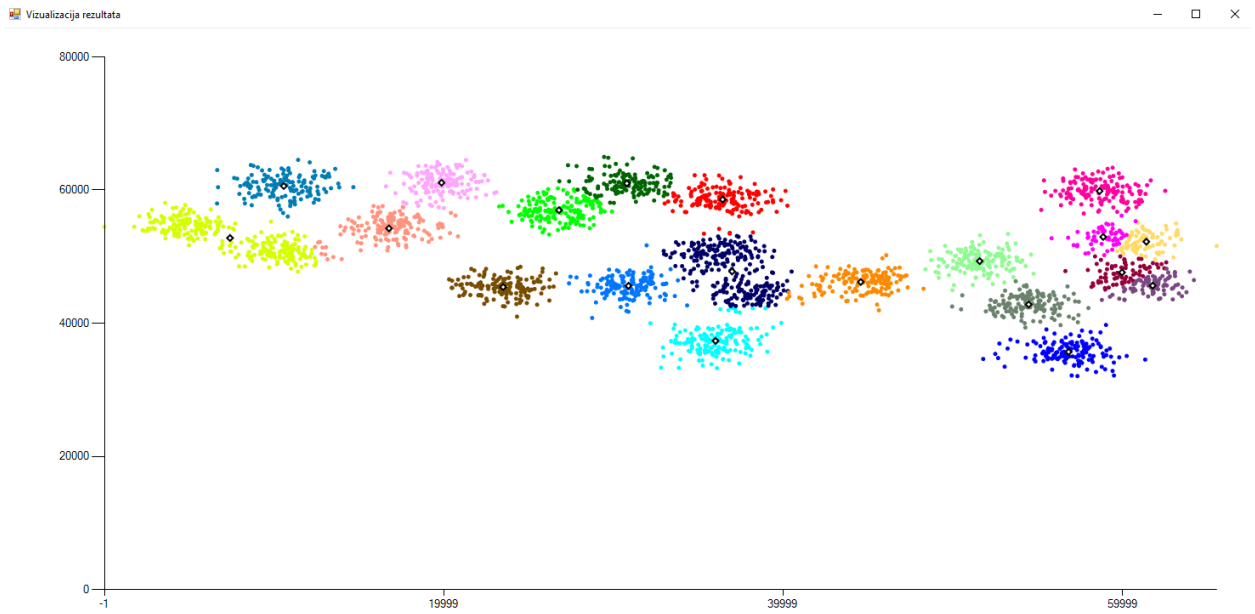
4.3. Vizualizacija rezultata

Nakon što je gotovo izvođenje metode za razvrstavanje, prikazuje se vizualizacija dobivenih rezultata. Vizualizacija se ostvaruje na grafikonu na kojem su objekti prikazani kao točke, a boja točke određena je grozdom kojem ta točka, odnosno objekt, pripada. Vizualizacija rezultata je moguća ako je dimenzionalnost podataka manja ili jednaka 2, a u suprotnome se na grafikon ispisuje odgovarajuća poruka. Ako se rezultati mogu vizualizirati, korisnik može birati hoće li na grafikonu biti prikazani i centroidi grozdova. Centroidi su prikazani kao dijamanti s crnim obrubom. Primjer vizualizacije rezultata razvrstavanja vidljiv je na slici 4.5.



Slika 4.5. *Vizualizacija rezultata grupiranja za dvodimenzionalne podatke i za podatke dimenzionalnosti veće od 2*

Klikom na ikonu povećala u gornjem desnom kutu grafikona prikazuju se rezultati grupiranja u zasebnoj Windows formi. Ta opcija je omogućena isključivo u slučaju kada je dimenzionalnost podataka manja ili jednaka 2. Primjer vizualizacije rezultata u zasebnoj Windows formi prikazan je na slici 4.6.

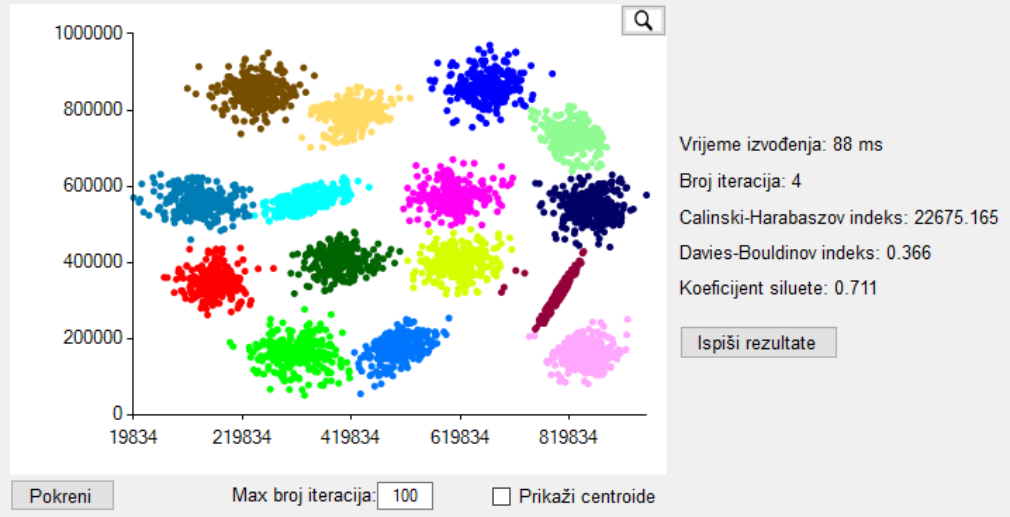


Slika 4.6. *Vizualizacija rezultata grupiranja u zasebnoj Windows formi*

Pri vizualizaciji rezultata neizrazitog grupiranja svaki objekt se prikazuje kao član onoga grozda za koji je koeficijent pripadnosti u_{ij} najveći.

4.4. Podatci o rezultatima razvrstavanja

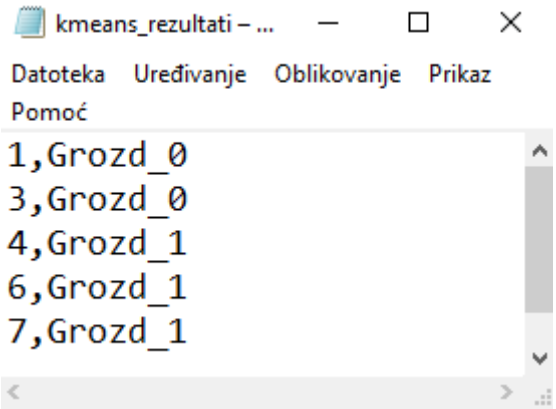
Osim vizualizacije samih rezultata razvrstavanja, nakon izvršavanja metode za razvrstavanje ispisuju se i drugi podatci o dobivenim rezultatima, odnosno broj iteracija, vrijeme izvođenja u milisekundama, Calinski-Harabaszov indeks, Davies-Bouldinov indeks i koeficijent siluete. Izračun navedenih mjera kvalitete razvrstavanja ostvaren je na načine opisane u poglavlju 3.3. Nije moguće ponovno pokrenuti metodu za razvrstavanje dok nisu izračunati svi podatci vezani uz prethodno izvođenje metode. Primjer prikaza dobivenih podataka je prikazan na slici 4.7.



Slika 4.7. *Primjer prikaza podataka o rezultatima razvrstavanja*

4.5. Ispis rezultata

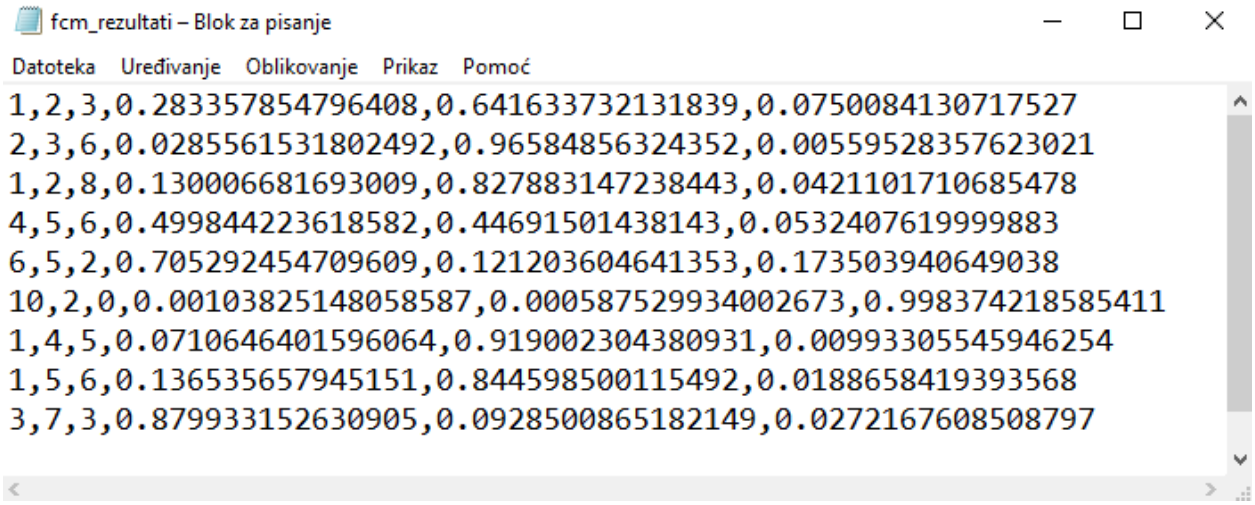
Ako se rezultati razvrstavanja žele spremiti radi daljnje analize ili korištenja u drugim programima, potrebno je kliknuti na gumb „Ispiši rezultate“ koji će stvoriti novu datoteku te u nju ispisati dobivene rezultate razvrstavanja. Ako je korišten algoritam čvrstog razvrstavanja, uz vrijednosti atributa svakog objekta dodaje se oznaka „Grozd_broj_grozda“, gdje broj grozda označava indeks grozda kojem objekt pripada. Primjer ispisa rezultata čvrstog razvrstavanja prikazan je na slici 4.8.



Slika 4.8. *Primjer ispisa rezultata čvrstog razvrstavanja*

Ako je korišten algoritam fuzzy c-means, odnosno algoritam neizrazitog razvrstavanja, uz vrijednosti atributa svakog objekta dodaje se k vrijednosti koje predstavljaju koeficijente pripadnosti tog objekta svakom od k grozdova, pri čemu je prva vrijednost pripadnost objekta

prvom grozdu, druga vrijednost pripadnost objekta drugom grozdu itd. Primjer ispisa rezultata neizrazitog razvrstavanja prikazan je na slici 4.9.



Slika 4.9. *Primjer ispisa rezultata neizrazitog razvrstavanja (k=3)*

5. EKSPERIMENTALNA ANALIZA

Programsko rješenje opisano u prethodnom poglavlju korišteno je za analizu i usporedbu metoda za razvrstavanje oko središnje točke na različitim skupovima podataka.

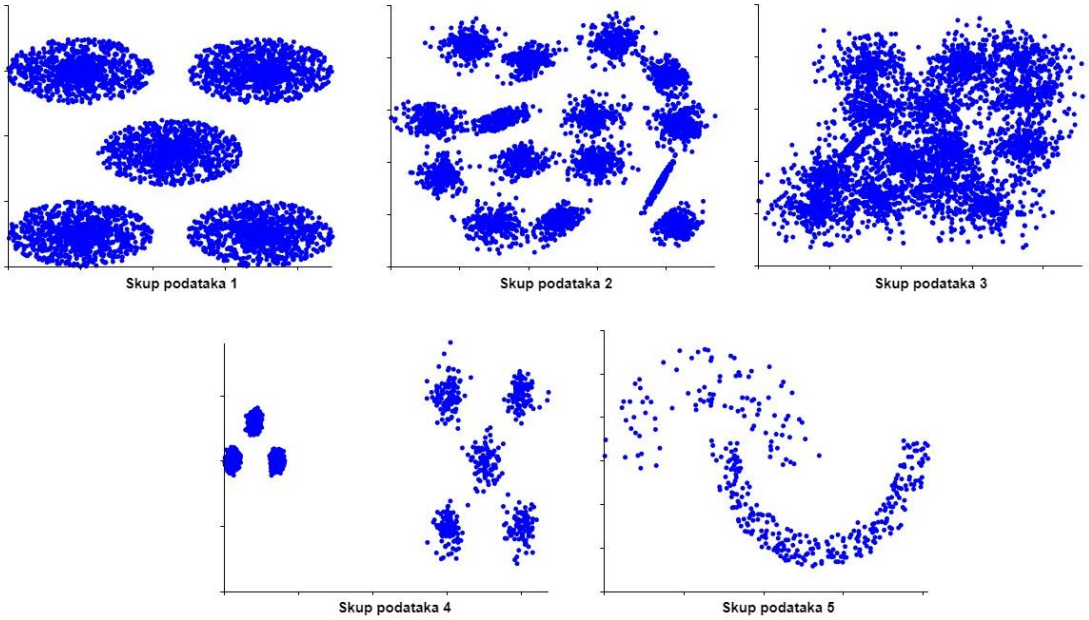
5.1. Korišteni skupovi podataka

U svrhu eksperimentalne analize je korišteno 7 skupova podataka različitih dimenzionalnosti, broja objekata, broja grozdova i oblika grozdova. Detalji o karakteristikama korištenih skupova podataka vidljivi su u tablici 5.1.

Tablica 5.1. *Korišteni skupovi podataka*

Naziv	**Dimenzionalnost**	**Broj objekata**	**Broj grozdova**	**Opis**
Skup podataka 1	2	5000	5	Dobro odvojeni grozdovi identičnog konveksnog oblika
Skup podataka 2	2	5000	15	Dobro odvojeni grozdovi različitih konveksnih oblika
Skup podataka 3	2	5000	15	Grozdovi s vrlo visokom razinom preklapanja
Skup podataka 4	2	6500	8	Grozdovi znatno različitih gustoća
Skup podataka 5	2	373	2	Grozdovi nekonveksnih oblika
Skup podataka 6	64	1024	16	Dobro odvojeni grozdovi hipersfernih oblika
Iris	4	150	3	Stvarni podaci, grozdovi različitih oblika

Prvih 6 skupova podataka sadrže sintetički generirane podatke, dok skup podataka *Iris* sadrži podatke o 50 uzoraka svake od triju vrsta cvijeta irisa (*Iris setosa*, *Iris virginica* i *Iris versicolor*), odnosno duljinu i širinu latice i čašičnog listića svakog uzorka. Svi skupovi podataka osim skupa podataka 1 preuzeti su s [23]. Na slici 5.1. prikazane su vizualizacije svih dvodimenzionalnih skupova podataka korištenih u eksperimentalnoj analizi.



Slika 5.1. *Dvodimenzionalni skupovi podataka korišteni u eksperimentalnoj analizi*

5.2. Postavke eksperimenta

Algoritmi korišteni u eksperimentalnoj analizi su algoritmi k-means, k-means++, k-medoids (s nasumičnom i k-means++ inicijalizacijom) i FCM. Svaki algoritam je pokretan 10 puta na pojedinom skupu podataka kako bi se utvrdilo koliko često daje optimalno rješenje te kolika su odstupanja između rezultata izvođenja istog algoritma na istom skupu podataka. Za svaki algoritam i svaki skup podataka zabilježeni su podaci o najuspješnijem izvođenju te podaci o prosječnom rezultatu svih 10 izvođenja. Mjereni podaci su broj iteracija, vrijeme izvođenja, Calinski-Harabaszov indeks, Davies-Bouldinov indeks i koeficijent siluete. Za skup podataka *Iris* je izračunat Randov indeks kako je riječ o skupu podataka za koji su poznate stvarne vrijednosti rezultata. Pri svim izvođenjima je korištena isključivo euklidska udaljenost, broj iteracija ograničen je na maksimalno 200, a pri izvođenju algoritma FCM koeficijent neizrazitosti i kriterij zaustavljanja su postavljeni na 2 (ako nije drugačije navedeno) i 10^{-5}.

5.3. Ostvareni rezultati

Rezultati su navođeni i opisivani redoslijedom skupova podataka iz tablice 5.1.

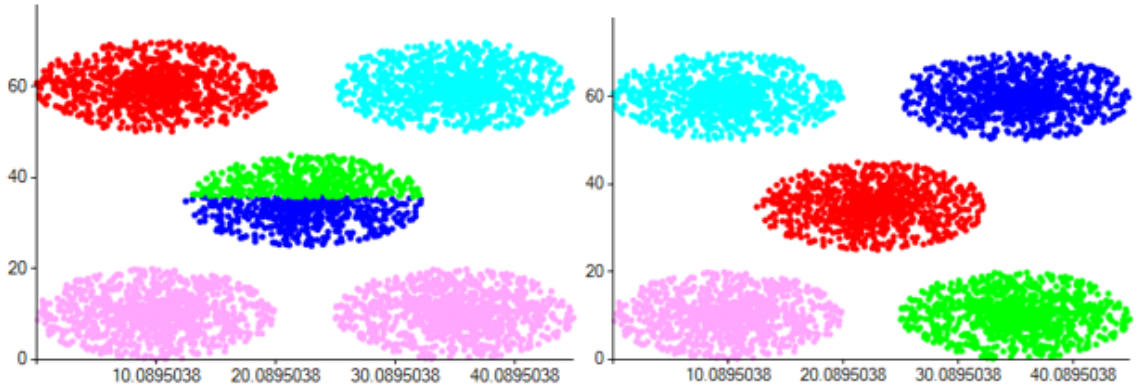
Tablica 5.2. *Najuspješnija izvođenja algoritama na skupu podataka 1*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	2	1 ms	23003,19	0,396	0,696
k-means++	2	1 ms	23003,19	0,396	0,696
k-medoids (nasumična inicijalizacija)	6	808 ms	23011,45	0,395	0,696
k-medoids (k-means++ inicijalizacija)	6	793 ms	23011,45	0,395	0,696
FCM	21	209 ms	23181,18	0,393	0,696

Tablica 5.3. *Prosječni rezultati izvođenja algoritama na skupu podataka 1*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	13	2,9 ms	16786,76	0,592	0,616
k-means++	5,5	1,8 ms	21451,94	0,445	0,676
k-medoids (nasumična inicijalizacija)	6,7	917,8 ms	23011,45	0,395	0,696
k-medoids (k-means++ inicijalizacija)	6,2	839,3 ms	23011,45	0,395	0,696
FCM	21,9	219,4 ms	23181,18	0,393	0,696

Tablice 5.2 i 5.3. prikazuju rezultate postignute na skupu podataka 1. Vidljivo je kako su svi algoritmi uspjeli postići točan rezultat. Međutim, dok su obje varijante algoritma k-medoids i algoritam FCM u svakom izvođenju postigli točan rezultat, algoritam k-means postigao ga je u samo 6 izvođenja, a algoritam k-means++ u 9 izvođenja. Razlog tomu je loš odabir početnih centroida koji je izraženiji kod algoritma k-means koji koristi nasumičan odabir. Primjer netočnog i točnog razvrstavanja dobivenog algoritmom k-means prikazan je na slici 5.2. Visoka vremenska složenost algoritma k-medoids s obzirom na broj objekata čini ga znatno sporijim od ostalih algoritama. Također, vidljivo je kako metoda inicijalizacije medoida nije imala utjecaj na rezultat algoritma k-medoids, ali je korištenje k-means++ inicijalizacije dovelo do smanjenja prosječnog broja iteracija, a time i vremena izvođenja. Algoritam FCM daje najbolje položaje središta grozdova, a time i najbolje vrijednosti DB i CH indeksa.



Slika 5.2. *Primjer netočnog i točnog razvrstavanja objekata iz skupa podataka 1*

Tablica 5.4. *Najuspješnija izvođenja algoritama na skupu podataka 2*

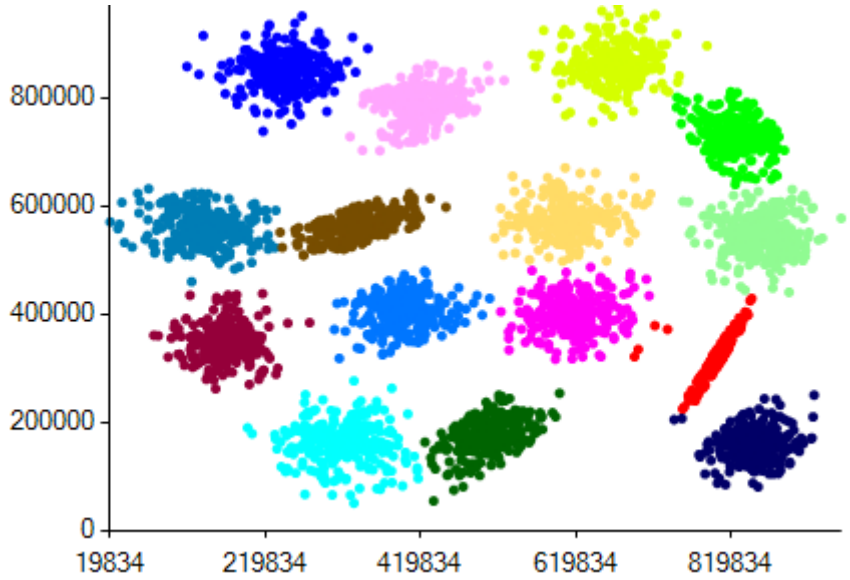
Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	10	10 ms	15144,67	0,483	0,661
k-means++	4	5 ms	22675,14	0,367	0,711
k-medoids (nasumična inicijalizacija)	19	2173 ms	22694,97	0,366	0,711
k-medoids (k-means++ inicijalizacija)	17	1920 ms	22694,97	0,366	0,711
FCM	30	2027 ms	22716,72	0,366	0,711

Tablica 5.5. *Prosječni rezultati izvođenja algoritama na skupu podataka 2*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	22,7	9,6 ms	10748,09	0,627	0,601
k-means++	10,2	7,9 ms	15204,45	0,496	0,656
k-medoids (nasumična inicijalizacija)	22,4	2566,1 ms	22694,97	0,366	0,711
k-medoids (k-means++ inicijalizacija)	19	2156,5 ms	22694,97	0,366	0,711
FCM	88,5	5993,9 ms	16337,75	0,451	0,676

Tablice 5.4. i 5.5. prikazuju rezultate postignute na skupu podataka 2. Povećanje broja grozdova negativno je utjecalo na algoritme k-means, k-means++ i FCM, čija se stopa postizanja optimalnog rezultata znatno smanjila. Algoritam k-means niti jednom nije uspio postići optimalni rezultat, dok su algoritmi k-means++ i FCM to uspjeli 2, odnosno 4 puta. Obje verzije algoritma k-medoids ponovno su postigle optimalni rezultat u svih 10 izvođenja, a k-means++ inicijalizacija se opet pokazala bržom. Povećanje broja grozdova znatno smanjuje vjerojatnost idealne inicijalizacije središta grozdova što rezultira krivo pozicioniranim središtima grozdova koja ne mogu postupno doći na idealnu lokaciju zbog utjecaja grozdova koji se nalaze između njih. Algoritam k-medoids zaobilazi taj problem korištenjem metode najpovoljnije zamjene medoida nekim drugim objektom, time dovodeći medoide na idealne pozicije u vrlo malom broju iteracija. Primjetljivo je usporenje algoritma FCM u odnosu na algoritam k-medoids što je posljedica njegove kvadratne vremenske složenosti s obzirom na broj grozdova.

Na slici 5.3. je prikazan optimalni rezultat razvrstavanja objekata iz skupa podataka 2. Iako je dobiveni rezultat najbolji rezultat koji se može dobiti korištenim algoritmima, primjetljivo je kako postoji nekolicina krivo razvrstanih objekata u okolini grozda obojanog crvenog bojom koji oblikom najviše odstupa od ostalih grozdova.



Slika 5.3. *Optimalno razvrstavanje objekata iz skupa podataka 2 korištenim algoritmima*

Tablica 5.6. *Najuspješnija izvođenja algoritama na skupu podataka 3*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	16	7 ms	7889,5	0,645	0,492
k-means++	19	11 ms	7889,403	0,645	0,492
k-medoids (nasumična inicijalizacija)	22	2750 ms	7889,019	0,645	0,492
k-medoids (k-means++ inicijalizacija)	27	3391 ms	7889,019	0,645	0,492
FCM	47	3189 ms	7777,032	0,652	0,492

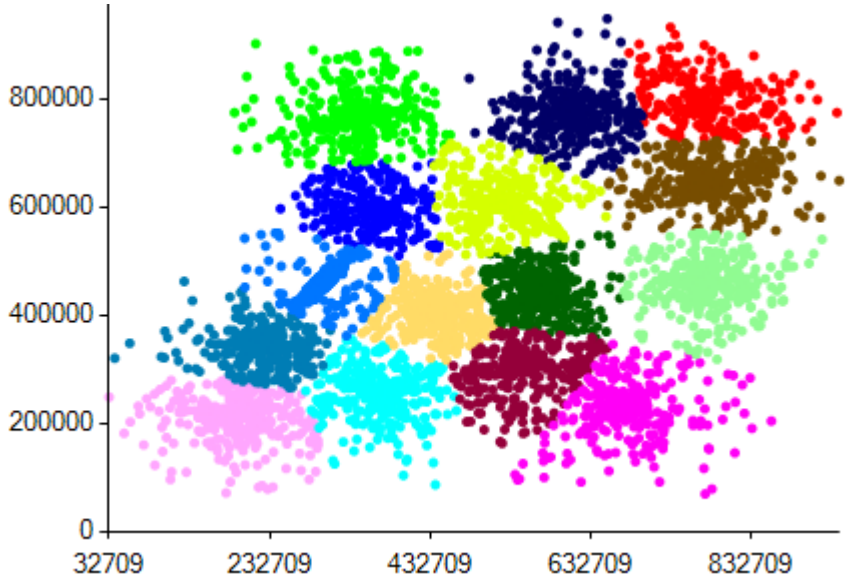
Tablica 5.7. *Prosječni rezultati izvođenja algoritama na skupu podataka 3*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	25	11,1 ms	6812,436	0,744	0,458
k-means++	22,8	13,8 ms	6941,461	0,722	0,466
k-medoids (nasumična inicijalizacija)	29,8	3761,5 ms	7889,019	0,645	0,492
k-medoids (k-means++ inicijalizacija)	30,9	3896,8 ms	7889,019	0,645	0,492
FCM	144,3	9857,7 ms	7022,572	0,732	0,468

Tablice 5.6. i 5.7. prikazuju rezultate postignute na skupu podataka 3. Skup podataka 3 ima jednak broj objekata i grozdova kao skup podataka 2, no sadrži veliku količinu preklapanja između grozdova. Dodavanje preklapanja između grozdova znatno je poboljšalo rezultate algoritma k-means. Naime, u 2 izvođenja postiže optimalan rezultat dok u ostalim izvođenjima daje rezultate koji ne odstupaju od optimuma u velikoj mjeri te su vrlo blizu rezultata algoritma k-means++. Rezultati ostalih algoritama nisu se znatno promijenili u odnosu na skup podataka 2. Iz navedenoga se može zaključiti kako veća količina preklapanja između grozdova smanjuje nedostatak nasumične inicijalizacije središta grozdova. Osim na rezultatima algoritma k-means, to je vidljivo i na rezultatima dviju varijanti algoritma k-medoids, gdje je nasumična inicijalizacija po prvi put postigla bolje rezultate (po pitanju brzine) nego k-means++ inicijalizacija.

Rezultati algoritma FCM su daleko najkorisniji za ovakav tip skupa podataka. Analizom koeficijenata pripadnosti u_{ij} može se odrediti koji objekti pripadaju većem broju grozdova. Također, moguće je filtriranje objekata koji pripadaju samo jednom grozdu ako se promatraju samo oni objekti kojima je najveći koeficijent pripadnosti veći od zadane vrijednosti.

Na slici 5.4. je prikazan optimalni rezultat razvrstavanja objekata iz skupa podataka 3.



Slika 5.4. *Optimalno razvrstavanje objekata iz skupa podataka 3 korištenim algoritmima*

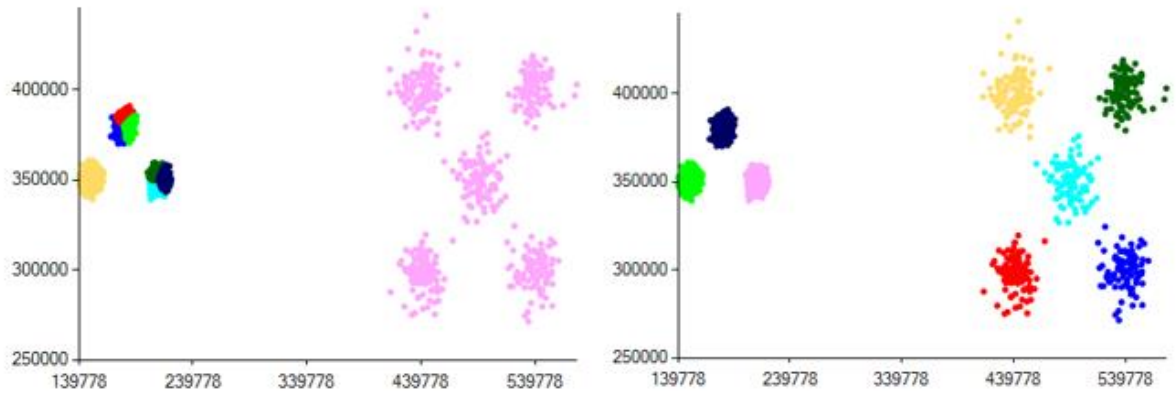
Tablica 5.8. *Najuspješnija izvođenja algoritama na skupu podataka 4*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	23	7 ms	29521,0	0,874	0,496
k-means++	3	2 ms	221461,0	0,291	0,858
k-medoids (nasumična inicijalizacija)	9	2133 ms	220822,6	0,292	0,858
k-medoids (k-means++ inicijalizacija)	9	2014 ms	220822,6	0,292	0,858
FCM	116	3341 ms	97024,58	0,603	0,678

Tablica 5.9. *Prosječni rezultati izvođenja algoritama na skupu podataka 4*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	39,5	14 ms	23514,87	0,831	0,509
k-means++	25,2	9,2 ms	123759,8	0,565	0,730
k-medoids (nasumična inicijalizacija)	10,2	2346,7 ms	220822,6	0,292	0,858
k-medoids (k-means++ inicijalizacija)	9,6	2187,6 ms	220822,6	0,292	0,858
FCM	191,6	5520,1 ms	40085,74	0,976	0,527

Tablice 5.8. i 5.9. prikazuju rezultate postignute na skupu podataka 4. Skup podataka se sastoji od triju grozdova koji se sastoje od 2000 objekata i 5 većih grozdova koji se sastoje od 100 objekata. Kako je vjerojatnost da nasumično odabrani objekt pripada jednom od tih 5 grozdova jednaka 7,7%, algoritmi koji koriste nasumičnu inicijalizaciju središta grozdova nemaju gotovo nikakvu vjerojatnost uspjeha (uz izuzetak algoritma k-medoids). To je vidljivo na rezultatima algoritama k-means i FCM koji niti jednom ne uspijevaju točno razvrstati objekte, već 5 rjeđe popunjenih grozdova najčešće identificiraju kao jedan grozd. Algoritam k-means++ je u 2 izvođenja postigao točno rješenje te je u ostalim izvođenjima uspio prepoznati barem 5 od 8 grozdova. Algoritam k-medoids je u svim izvođenjima postigao točan rezultat neovisno o metodi inicijalizacije medoida. Primjer netočnog i točnog razvrstavanja prikazan je na slici 5.5.



Slika 5.5. *Primjer netočnog i točnog razvrstavanja objekata iz skupa podataka 4*

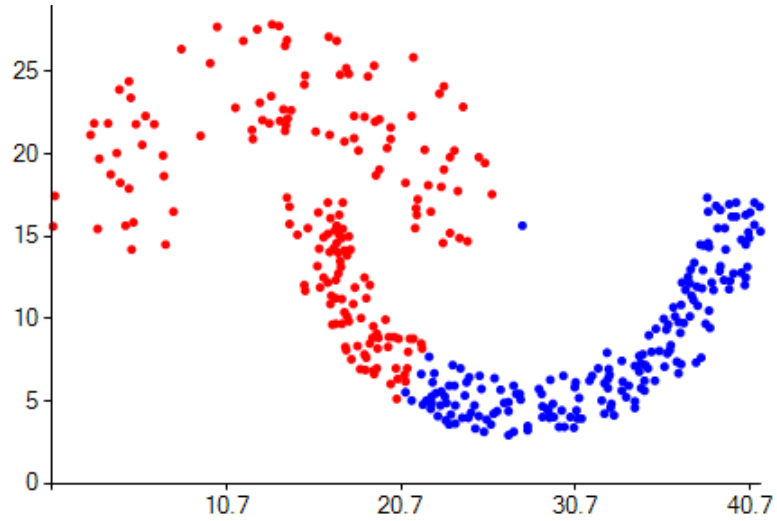
Tablica 5.10. *Najuspješnija izvođenja algoritama na skupu podataka 5*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	8	~0 ms	503,457	0,782	0,495
k-means++	7	~0 ms	503,457	0,782	0,495
k-medoids (nasumična inicijalizacija)	3	2 ms	537,688	0,754	0,495
k-medoids (k-means++ inicijalizacija)	3	2 ms	537,688	0,754	0,495
FCM	28	5 ms	520,08	0,771	0,495

Tablica 5.11. *Prosječni rezultati izvođenja algoritama na skupu podataka 5*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	8,5	0,1 ms	503,417	0,783	0,495
k-means++	9,5	0,1 ms	503,406	0,783	0,495
k-medoids (nasumična inicijalizacija)	4,1	3,1 ms	537,688	0,754	0,495
k-medoids (k-means++ inicijalizacija)	3,8	2,6 ms	537,688	0,754	0,495
FCM	33	6,1 ms	520,008	0,771	0,495

Tablice 5.10. i 5.11. prikazuju rezultate postignute na skupu podataka 5. Skup se sastoji od dvaju grozdova nekonveksnih oblika. Niti jedan algoritam nije uspio točno razvrstati objekte iz skupa. Svi algoritmi su dali identičan rezultat prikazan na slici 5.6. Za razvrstavanje ovakvog tipa skupa podataka potrebno je koristiti metodu za razvrstavanje po gustoći. Također, pri razvrstavanju ovakvog tipa skupa podataka CH indeks, DB indeks i koeficijent siluete nisu valjane mjere kvalitete razvrstavanja.



Slika 5.6. *Rezultat razvrstavanja skupa podataka 5*

Tablica 5.12. *Najuspješnija izvođenja algoritama na skupu podataka 6*

Algoritam	**Broj iteracija**	**Vrijeme izvođenja**	**CH indeks**	**DB indeks**	**Koef. siluete**
k-means	3	4 ms	1387,59	0,539	0,851
k-means++	2	12 ms	54451,23	0,050	0,966
k-medoids (nasumična inicijalizacija)	16	110 ms	52507,62	0,050	0,966
k-medoids (k-means++ inicijalizacija)	16	119 ms	52507,62	0,050	0,966
FCM ($m = 2$)	25	3273 ms	17,713	10995759	0,395
FCM ($m = 1{,}03125$)	4	515 ms	54451,23	0,050	0,966

Tablica 5.13. *Prosječni rezultati izvođenja algoritama na skupu podataka 6*

Algoritam	Broj iteracija	Vrijeme izvođenja	CH indeks	DB indeks	Koef. siluete
k-means	3,9	10,3 ms	470,019	1,061	0,662
k-means++	2	12,3 ms	54451,23	0,050	0,966
k-medoids (nasumična inicijalizacija)	17,7	123,6 ms	52507,62	0,050	0,966
k-medoids (k-means++ inicijalizacija)	16,8	125,2 ms	52507,62	0,050	0,966
FCM ($m = 2$)	25,9	3316,9 ms	8,369	10487118	0,213
FCM ($m = 1{,}03125$)	12	1520 ms	17092,43	0,510	0,899

Tablice 5.12. i 5.13. prikazuju rezultate postignute na skupu podataka 6. Riječ je o 64-dimenzionalnom skupu podataka koji se sastoji od 16 dobro odvojenih grozdova pravilnih (hipersfernih) oblika. Veliki broj grozdova i nedostatak njihovog preklapanja ponovno je rezultirao lošom učinkovitošću algoritma k-means, koji niti u jednom izvođenju nije uspio postići točno rješenje. Algoritmi k-means++ i k-medoids postigli su točno rješenje u svakom izvođenju. Iz navedenog je vidljivo kako prethodno postignuta opažanja za algoritme k-means, k-means++ i k-medoids vrijede i za visokodimenzionalne skupove podataka pod uvjetom da su grozdovi hipersfernih oblika, što je rijetko slučaj jer postoji velika vjerojatnost da određene dimenzije imaju veći utjecaj na svojstva skupa od drugih. Algoritam FCM daje izuzetno loše rezultate ako se koristi vrijednost koeficijenta neizrazitosti $m = 2$. Međutim, ako koeficijent neizrazitosti izračunamo jednadžbom (3-21), dobivamo vrijednost $m = 1{,}03125$ koja daje bolje rezultate, odnosno postiže točno rješenje u 3 izvođenja. Negativna posljedica korištenja male vrijednosti koeficijenta neizrazitosti je smanjena mogućnost algoritma FCM da pronađe preklapanja između grozdova.

Tablica 5.14. *Najuspješnija izvođenja algoritama na skupu podataka Iris*

Algoritam	Broj iteracija	Vrijeme izvođenja	Randov indeks
k-means	4	~0 ms	0,854
k-means++	8	1 ms	0,852
k-medoids (nasumična inicijalizacija)	5	2 ms	0,868
k-medoids (k-means++ inicijalizacija)	5	1 ms	0,868
FCM ($m = 2$)	56	18 ms	0,844
FCM ($m = 1{,}5$)	21	7 ms	0,837

Tablica 5.15. *Prosječni rezultati izvođenja algoritama na skupu podataka Iris*

Algoritam	Broj iteracija	Vrijeme izvođenja	Randov indeks
k-means	8,9	0,4 ms	0,836
k-means++	9,3	0,6 ms	0,832
k-medoids (nasumična inicijalizacija)	5,6	1,5 ms	0,844
k-medoids (k-means++ inicijalizacija)	5,9	1,6 ms	0,852
FCM ($m = 2$)	54,5	17,9 ms	0,840
FCM ($m = 1{,}5$)	54,1	18,1 ms	0,835

Tablice 5.14. i 5.15. prikazuju rezultate postignute na skupu podataka *Iris*. Svi algoritmi daju podjednake rezultate, no obje verzije algoritma k-medoids daju najbolje rezultate postižući Randov indeks od 0,868 (90% točno razvrstanih objekata). Krivo razvrstani objekti posljedica su preklapanja između grozdova koji predstavljaju vrste *Iris virginica* i *Iris versicolor*, dok su svi uzorci vrste *Iris setosa* ispravno razvrstani. Za razliku od prethodnog skupa podataka, korištenje jednadžbe (3-21) za određivanje koeficijenta neizrazitosti algoritma FCM nije poboljšalo rezultate u odnosu na koeficijent neizrazitosti $m = 2$.

6. ZAKLJUČAK

Zadatak završnog rada bio je napraviti aplikaciju za razvrstavanje objekata koristeći metode za razvrstavanje oko središnje točke te upotrijebiti tu aplikaciju za usporedbu i analizu različitih metoda razvrstavanja. U radu je opisan pojam grupiranja, pojedine metode za razvrstavanje oko središnje točke, mjere kvalitete razvrstavanja te ostvareno programsko rješenje. Testirani su algoritmi k-means, k-means++, k-medoids i fuzzy c-means.

Algoritam k-means je vrlo koristan zbog lagane implementacije i velike brzine, no daje dobre rezultate samo na skupovima podataka koji se sastoje od malog broja grozdova te na skupovima podataka čiji grozdovi imaju veliku razinu preklapanja. Algoritam k-means++ pokazao se kao dobra nadogradnja algoritma k-means koja postiže dobre rezultate i na skupovima podataka s većim brojem grozdova te nerijetko zahtjeva manji broj iteracija i rezultira kraćim vremenom izvođenja zbog boljeg izbora početnih centroida. Jedan od nedostataka algoritama k-means i k-means++ je potreba za višestrukim izvođenjima algoritma na istom skupu podataka kako bi se dobilo optimalno rješenje. Brzina izvođenja i točnost razvrstavanja čine algoritam k-means++ najboljim izborom za razvrstavanje velikih skupova podataka. Algoritam k-medoids se pokazao uvjerljivo najpreciznijim od testiranih algoritama, pronalazeći optimalna rješenja najčešće od svih algoritama. Također, gotovo sva izvođenja algoritma k-medoids na istom skupu podataka dala su isti rezultat, što uklanja potrebu za brojnim ponovnim izvođenjima algoritma. Metoda inicijalizacije medoida nije imala utjecaj na rezultate razvrstavanja algoritmom k-medoids, ali je utjecala na vrijeme izvođenja. Glavni nedostatak algoritma k-medoids je njegova vremenska složenost koja ga čini neprikladnim za velike skupove podataka. Algoritam fuzzy c-means je po pitanju točnosti davao slične rezultate kao algoritam k-means++. Međutim, njegova glavna svrha je korištenje na skupovima podataka čiji se grozdovi preklapaju jer omogućuje otkrivanje objekata koji pripadaju većem broju grozdova. Posljedica detaljnijeg izračuna je visoka vremenska složenost s obzirom na broj grozdova. Jedno moguće poboljšanje algoritma fuzzy c-means je naprednija metoda inicijalizacije centroida. Glavni nedostatak algoritma fuzzy c-means su njegove slabe performanse na visokodimenzionalnim skupovima podataka.

Nijedan od testiranih algoritama nije uspio pravilno razvrstati objekte koji tvore grozdove nekonveksnih oblika. To je nedostatak svih algoritama za razvrstavanje oko središnje točke te se u tim slučajima preporučuje koristiti algoritam za razvrstavanje po gustoći ili neki drugi tip algoritma za razvrstavanje. Iz istog razloga se ne preporučuje korištenje algoritama za

razvrstavanje oko središnje točke na visokodimenzionalnim skupovima podataka osim ako je unaprijed poznato da su grozdovi hipersfernog oblika.

Programsko rješenje moglo bi se unaprijediti dodavanjem drugih algoritama za razvrstavanje oko središnje točke, dodatnih mjera kvalitete razvrstavanja, dodatnih mjera udaljenosti, podrške za dodatne tipove datoteka s podacima, skaliranja podataka te detekcije objekata koji ne pripadaju niti jednom grozdu. Postojeće programsko rješenje može se upotrijebiti za daljnje usporedbe i analize algoritama za razvrstavanje oko središnje točke.

LITERATURA

[1] B. Everitt, S. Landau, M. Leese, D. Stahl, Cluster Analysis, 5th Edition, John Wiley & Sons, Chichester, 2011.

[2] C. Aggarwal, C. Reddy, Data Clustering, Algorithms and Applications, Chapman & Hall, Boca Raton, FL, 2013.

[3] S. Belhaouari, S. Ahmed, S. Mansour, Optimized K-Means Algorithm, Mathematical Problem in Engineering, Vol. 2014, pp. 1-14, September 2014.

[4] E.Schubert, P. Rousseeuw, Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms, Cornell University, Ithaca, NY, 2018.

[5] X. Wang, Y. Wang, L. Wang, Improving fuzzy c-means clustering based on feature-weight learning, Pattern Recognition Letters, Vol. 25, No. 10, pp. 1123-1132, July 2004.

[6] B. Zhang, Comparison of the Performance of Center-Based Clustering Algorithms, PAKDD 2003: Advances in Knowledge Discovery and Data Mining, pp. 63-74, 2003.

[7] A. Nayyar, V. Puri, Comprehensive Analysis & Performance Comparison of Clustering Algorithms for Big Dana, Review of Computer Engineering Research, Vol. 4, No. 2, pp. 54-80, October 2017.

[8] Outlier App: An Interactive Visualization of Outlier Algorithms, datascience+, dostupno na: https://datascienceplus.com/outlier-app-an-interactive-visualization-of-outlier-algorithms/ [26.6.2020.]

[9] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, 2007.

[10] H. Bock, Probabilistic aspects in cluster analysis, Springer-Verlag, Augsburg, 1989.

[11] M. Gagolewski, dostupno na: https://www.gagolewski.com/resources/data/clustering/sipu_o_plot-2.png, [23.6.2020.]

[12] Euclidean and Euclidean Squared Distance Metric, Improved Outcomes Software, dostupno na: http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm, [23.6.2020.]

[13] D. Keim, A. Hinneburg, Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In Proceedings of the 25th international

conference on very large data bases (VLDB ’99), Morgan Kaufmann, San Francisco, CA, 1999.

[14] D. Arthur, S. Vassilvitskii, k-means++: The Advantages of Careful Seeding, In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, str. 1027-1035, 2007.

[15] M. Tiwari, R. Singh, Comparative Investigation of K-Means and K-Medoid Algorithm on Iris Data, International Journal of Engineering Research and Development, No. 4, Vol. 8, pp. 69-72, November 2012.

[16] J. Bezdek, R. Ehrlich, W. Full, FCM: The Fuzzy c-means Clustering Algorithm, Computers & Geosciences, Vol. 10, No. 2-3, pp. 191-203, 1984.

[17] R. Winkler, F. Klawonn, R. Kruse, Fuzzy C-Means in High Dimensional Spaces, International Journal of Fuzzy System Applications, Vol. 11, June 2010.

[18] M. Chau, R. Cheng, B. Kao, J. Ng, Uncertain data mining: An example in clustering location data, In PAKDD Singapore 2006, pp. 199–204, 2006.

[19] L. Xiao, E. Hung, An Efficient Distance Calculation Method for Uncertain Objects, Computational Intelligence and Data Mining, CIDM, 2007.

[20] Clustering performance evaluation, scikit-learn, dostupno na: https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation [25.6.2020.]

[21] Silhouette plot, Mathworks, dostupno na: https://www.mathworks.com/help/stats/silhouette.html [25.6.2020.]

[22] E. Schubert, P.J. Rousseeuw, Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms, Similarity Search and Applications, Springer International Publishing, 11807, pp. 171-187

[23] Clustering basic benchmark, University of Eastern Finland, dostupno na: http://cs.joensuu.fi/sipu/datasets/ [23.8.2020.]

SAŽETAK

U završnom radu je opisan pojam razvrstavanja objekata te vrste metoda za razvrstavanje objekata s naglaskom na razvrstavanje oko središnje točke. Opisane su neke od češće korištenih metoda razvrstavanja objekata oko središnje točke. Navedene su i objašnjene mjere udaljenosti za objekte s brojčanim obilježjima te mjere kvalitete razvrstavanja objekata. Dan je opis ostvarenog programskog rješenja koje omogućuje izvođenje i usporedbu metoda za razvrstavanje oko središnje točke. Algoritmi k-means, k-means++, k-medoids i fuzzy c-means analizirani su i uspoređeni koristeći ostvareno programsko rješenje na raznim skupovima podataka. Algoritam k-medoids pokazao se kao najbolji algoritam za razvrstavanje malih skupova podataka, algoritam k-means++ kao najbolji algoritam za razvrstavanje velikih skupova podataka zbog male vremenske složenosti, a algoritam fuzzy c-means kao najprikladniji algoritam za razvrstavanje niskodimenzionalnih skupova podataka s preklapanjem između grozdova.

Ključne riječi: fuzzy c-means, k-means, k-means++, k-medoids, razvrstavanje objekata

ABSTRACT

COMPARISON OF CENTER-BASED CLUSTERING METHODS

The final paper describes data clustering and different types of data clustering methods with an emphasis on center-based data clustering. Descriptions of some of the most frequently used center-based data clustering methods are given. Distance metrics for numerical data and assessment metrics for data clustering are listed and explained. A description of the realized software solution that allows execution and comparison of center-based data clustering methods is given. The algorithms which were analyzed and compared using the software solution on various datasets are k-means, k-means++, k-medoids and fuzzy c-means. The k-medoids algorithm proved to be the best algorithm for clustering small datasets, the k-means++ algorithm proved to be the best algorithm for clustering large datasets due to its low time complexity, and the fuzzy c-means algorithm proved to be the most suitable algorithm for clustering low-dimensional datasets with overlap between clusters.

Keywords: data clustering, fuzzy c-means, k-means, k-means++, k-medoids

ŽIVOTOPIS

Filip Znaor rođen je 4. ožujka 1998. godine u Osijeku gdje je pohađao Osnovnu školu Jagode Truhelke. 2013. godine upisuje III. gimnaziju Osijek koju završava 2017. godine. Iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek. Tijekom prve godine studija pohađa preddiplomski studij elektrotehnike nakon čega prelazi na preddiplomski studij računarstva na kojem je trenutno student treće godine.

_____

Filip Znaor

PRILOZI

Na CD-u:

1. Završni rad u *.docx* i *.pdf* formatu
2. Izvorni kod ostvarenog programskog rješenja
3. Skupovi podataka korišteni u eksperimentalnoj analizi