UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

SEGMENTACIJA MARKETINŠKIH PODATAKA PRIMJENOM ALGORITAMA ZA KLASTERIZACIJU

- MASTER RAD -

Mentor:
Doc. dr Miloš Brajović

Student:
Lazar Trifunović

Podgorica, 2025.

UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET

Lazar Trifunović

SEGMENTACIJA MARKETINŠKIH PODATAKA PRIMJENOM ALGORITAMA ZA KLASTERIZACIJU

- MASTER RAD -

Podgorica, 2025.

PODACI I INFORMACIJE O MAGISTRANDU

Ime i prezime: **Lazar Trifunović**

Datum i mjesto rođenja: **29. januar 2001. godine, Berane**

Naziv završenog osnovnog studijskog programa i godina završetka studija: **Studije primijenjenog računarstva, 2022. godine**

INFORMACIJE O MASTER RADU

Studijski program: **Primijenjeno računarstvo**

Naslov rada: **Segmentacija marketinških podataka primjenom algoritama za klasterizaciju**

Fakultet/akademija na kojoj je rad odbranjen: **Elektrotehnički fakultet**

UDK, OCJENA I ODBRANA MASTER RADA

Datum prijave master rada: **13.02.2025.**

Datum sjednice vijeća na kojoj je prihvaćena tema: **20.02.2025.**

Mentor: **Doc. dr Miloš Brajović**

Komisija za ocjenu/odbranu rada:

1. **Prof. dr Vesna Popović-Bugarin**, predsjednik;
2. **Doc. dr Miloš Brajović**, mentor;
3. **Doc. dr Isidora Stanković**, član.

Datum odbrane: **25.12.2025.**

Izjava o autorstvu

Potpisani-a: Lazar Trifunović

Broj indeksa/upisa: 15/22

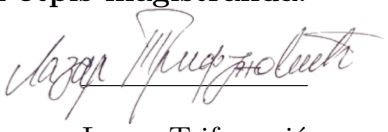
Izjavljujem

master rad po nazivom:

„Segmentacija marketinških podataka primjenom algoritama za klasterizaciju“

- rezultat sopstvenog istraživačkog rada,
- da predloženi master rad ni u cjelini ni u djelovima nije bio predložen za dobijanje bilo koje diplome prema strudijskim programima drugih ustanova visokog obrazovanja,
- da su rezultati korektno navedeni i
- da nijesam povrijedio/la autorska prava intelektualne svojine koja pripadaju trećim licima.

Potpis magistranda:



Lazar Trifunović

Podgorica, 24.11.2025. godine

Sažetak

Ovaj master rad se bavi segmentacijom marketinških podataka primjenom algoritama za klasterizaciju, sa ciljem identifikovanja značajnih grupa potrošača koje mogu doprinijeti donošenju informisanih marketinških odluka. U istraživanju su korišćena dva realna skupa podataka: „Customer Personality Analysis“, koji obuhvata demografske, socio-ekonomske i potrošačke osobine kupaca, i „Online Retail“, koji sadrži transakcione zapise elektronske trgovine. Prije klasterizacije, izvršeno je sveobuhvatno pretprocesiranje podataka, uključujući čišćenje skupa podataka, rješavanje problema nedostajućih vrijednosti, inženjering karakteristika, kodiranje kategorijskih karakteristika, uklanjanje ekstremnih vrijednosti, skaliranje i transformacije podataka. Dodatno, primijenjena je PCA analiza kako bi se smanjila dimenzionalnost podataka, poboljšala vizuelizacija i unaprijedila interpretacija rezultata klasterizacije.

U okviru rada analizirana su četiri algoritma za klasterizaciju: K-means, aglomerativni hijerarhijski algoritam, DBSCAN i spektralna klasterizacija. Ovi algoritmi izabrani su kako bi pokrili različite pristupe – zasnovane na centroidima, hijerarhijske, zasnovane na gustini i na spektru – čime je omogućen sveobuhvatan uvid u njihove performanse na podacima različite strukture. Procjena rezultata izvršena je pomoću kvantitativnih mjera, poput *Silhouette Score*-a i *Davies-Bouldin Index*-a, kao i kroz vizuelne interpretacije i analize dobijenih potrošačkih grupa.

Eksperimentalni rezultati pokazuju da struktura podataka značajno utiče na efikasnost algoritama. Na „Customer Personality Analysis“ skupu podataka najbolje rezultate je ostvario aglomerativni hijerarhijski algoritam, koji zbog svoje hijerarhijske prirode modeluje kompleksne odnose među atributima i formira stabilne i jasno razdvojene klastere. Na „Online Retail“ skupu podataka K-means je dao najbolje rezultate, zahvaljujući izraženoj numeričkoj strukturi podataka i mogućnosti algoritma da formira kompaktne potrošačke segmente.

Sveukupno, rezultati rada pokazuju da ne postoji univerzalni algoritam koji postiže najbolje rezultate na svim tipovima marketinških podataka. Umjesto toga, izbor metode treba prilagoditi prirodi podataka i cilju segmentacije. Zaključci dobijeni u radu predstavljaju vrijedne smjernice za primjenu klaster analize u marketingu, kao i osnovu za razvoj naprednih sistema personalizacije i ciljanja potrošača.

Ključne riječi: algoritmi za klasterizaciju, segmentacija marketinških podataka, analiza podataka, nenadgledano učenje, redukcija dimenzionalnosti.

Abstract

This master's thesis examines the segmentation of marketing data using clustering algorithms, with the aim of identifying meaningful consumer groups that can support informed marketing decision-making. The study utilizes two real-world datasets: the "Customer Personality Analysis" dataset, which includes demographic, socio-economic, and behavioral attributes of customers, and the "Online Retail" dataset, which contains transactional records from an e-commerce platform. Prior to clustering, extensive data preprocessing was carried out, including dataset cleaning, handling of missing values, feature engineering, categorical feature encoding, removal of extreme values, scaling, and data transformations. Additionally, PCA was applied to reduce data dimensionality, enhance visualization, and improve the interpretability of the clustering results.

Four clustering algorithms were analyzed in this research: K-means, agglomerative hierarchical clustering, DBSCAN, and spectral clustering. These algorithms were selected to represent different methodological approaches—centroid-based, hierarchical, density-based, and spectral—providing a comprehensive overview of their performance on datasets with varying structures. The evaluation of the clustering results was performed using quantitative metrics, such as the *Silhouette Score* and the *Davies–Bouldin Index*, as well as through visual inspection and analysis of the resulting consumer groups.

Experimental results demonstrate that the structure of the data has a significant impact on the effectiveness of the algorithms. In the "Customer Personality Analysis" dataset, the agglomerative hierarchical clustering algorithm achieved the best results, successfully modeling complex relationships between attributes and producing stable and clearly separated clusters due to its hierarchical nature. In contrast, on the "Online Retail" dataset, K-means demonstrated the best performance, owing to the strongly numerical structure of the data and the algorithm's ability to form compact and well-defined consumer segments.

Overall, the findings of this research indicate that there is no universal clustering algorithm that performs best across all types of marketing data. Instead, the choice of method should be adapted to the nature of the dataset and the goals of the segmentation task. The conclusions presented in this thesis offer valuable guidelines for the application of cluster analysis in marketing and provide a foundation for the development of advanced systems for consumer personalization and targeting.

Keywords: clustering algorithms, marketing data segmentation, data analysis, unsupervised learning, dimensionality reduction.

Sadržaj

Spisak slika

Spisak tabela

1 Uvod

U savremenom poslovnom okruženju, marketinške kampanje igraju ključnu ulogu. Marketinške kampanje generišu ogromnu količinu podataka o korisnicima, njihovom ponašanju, preferencijama i interakcijama. Ovi podaci se često prikupljaju korišćenjem različitih digitalnih kanala, poput društvenih mreža, specijalizovanih sajtova za prodaju i mobilnih aplikacija, te služe kao osnova za donošenje novih strateških odluka. Donošenjem strateških odluka treba preciznije targetirati određene grupe korisnika i, u skladu sa njihovim preferencijama, predložiti im različite proizvode. Međutim, bez odgovarajuće obrade i analize, ovakvi podaci često ostaju neiskorišćeni.

Klasterizacija predstavlja jednu od ključnih tehnika u analizi podataka, jer omogućava otkrivanje skrivenih obrazaca i grupisanje objekata na osnovu njihove međusobne sličnosti. Ova metoda posebno je korisna u marketingu, gdje pomaže marketinškim timovima da indetifikuju različite grupe korisnika i prilagode ponudu njihovim specifičnim potrebama. Na primjer, klasterizacija može otkriti grupe korisnika koje preferiraju luksuzne proizvode, kao i one koji su više usmjereni ka cjenovno pristupačnijim opcijama, čime se omogućava personalizacija marketinških aktivnosti. Pravilno primijenjena klaster analiza omogućava dublje razumijevanje korisničkog ponašanja i poboljšava efikasnost marketinške strategije [1]. Ovakav pristup doprinosi efikasnijem targetiranju kampanja, većem stepenu angažovanja korisnika i povećanju njihovog zadovoljstva.

Poseban izazov u segmentaciji marketinških podataka predstavlja izbor algoritma za klasterizaciju. Različiti algoritmi, kao što su K-means, DBSCAN (eng. *Density-Based Spatial Clustering of Applications with Noise*), hijerarhijska i spektralna klasterizacija, mogu dati različite rezultate čak i na istom skupu podataka. Njihove performanse mogu značajno varirati u zavisnosti od strukture podataka, dimenzionalnosti i prisustva šuma, što dodatno komplikuje proces odlučivanja. Ne postoji opšteprihvaćena strategija za odabir najpogodnijeg algoritma, pa se u praksi algoritam često bira bez jasnog utemeljenja, što može dovesti do nepouzdanih rezultata [2]. Određeni algoritmi su pogodniji za velike skupove podataka, dok su drugi pogodniji za manje i kompleksnije podatke [3]. Takođe, performanse algoritma zavise i od izbora osnovnih hiperparametara algoritma kao što su broj klastera, kod K-means, hijerarhijske i spektralne klasterizacije, i hiperparametara gustine kod DBSCAN-a. Pravilna podešavanja ovih hiperparametara mogu značajno uticati na kvalitet segmentacije, zbog čega se često koriste metode za odabir optimalnih vrijednosti tih hiperparametara [3, 4]. Imajući u vidu ove izazove, neophodno je sprovesti detaljnu analizu performansi različitih algoritama

kako bi se odabrao onaj koji postiže najbolje rezultate za određeni skup podataka.

Iako su algoritmi klasterizacije široko proučavani u oblasti mašinskog učenja, postoji ograničen broj radova koji detaljno upoređuju njihove performanse u marketinškim kampanjama na realnim skupovima podataka. Većina dosadašnjih istraživanja fokusira se na teorijske aspekte algoritama, dok primjena na konkretne marketinške podatke često izostaje. Takođe, postoje neslaganja među autorima u pogledu optimalne metodologije za segmentaciju podataka – dok jedni favorizuju metode poput K-means-a, drugi prednost daju algoritmima koji mogu raditi sa podacima nepravilnih oblika, poput DBSCAN-a. Na primjer, u radu [5] se ističe da je K-means najpopularniji algoritam zbog svoje efikasnosti i jednostavnosti, ali da njegova pretpostavka o sferičnim i klasterima koju su približno jednaki u veličini može ograničiti primjenu na složenije i nepravilno oblikovane podatke. S druge strane, rad [6] ističe da ne postoji univerzalan pristup koji funkcioniše najbolje u svim situacijama i da izbor algoritma zavisi od strukture podataka i ciljeva analize. Očekuje se da ovo istraživanje obogati postojeća znanja kroz empirijsku analizu algoritama klasterizacije primijenjenih u segmentaciji marketinških podataka.

Cilj ovog istraživanja je analiza i uporedna evaluacija efikasnosti različitih algoritama za klasterizaciju primijenjenih na podacima prikupljenim tokom marketinških kampanja. Istražiće se prednosti i nedostaci K-means-a, DBSCAN-a, hijerarhijske i spektralne klasterizacije, kako bi se ocijenila njihova sposobnost da efikasno i precizno grupišu podatke u klastere. Konkretno, istraživanje će se fokusirati na dva javno dostupna i popularna skupa podataka: „Customer Personality Analysis“ i „Online Retail“. Pored same analize performansi, biće razmatrani i faktori koji utiču na rezultate, kao što su skaliranje podataka, izbor optimalnih vrijednosti hiperparametara, kao i metoda za detekciju i eliminaciju izolovanih tačaka. Takođe, u cilju sveobuhvatne evaluacije, rezultati klasterizacije biće upoređeni korišćenjem različitih metrika koje omogućavaju kvantifikaciju kvaliteta dobijenih klastera.

Praktični dio ovog istraživanja, odnosno kompletna eksperimentalna evaluacija, sprovedeni su u Python okruženju. Programski jezik Python je odabran zbog svoje fleksibilnosti, bogate kolekcije biblioteka i široke primjene u oblasti analize podataka, što omogućava efikasno testiranje i poređenje performansi različitih algoritama na odabranim skupovima podataka. Biblioteke koje se koriste uključuju `Scikit-learn` za primjenu algoritama klasterizacije, `Pandas` i `NumPy` za manipulaciju i obradu podataka, kao i `Matplotlib` i `Seaborn` za vizuelizaciju podataka.

Rad je organizovan u nekoliko poglavlja. Nakon uvodnog dijela, rad će se baviti teorijskim osnovama klasterizacije koje uključuju metrike udaljenosti, pregled algoritama za klasterizaciju i metode za odabir optimalnih vrijednosti hiperparametara. Zatim, biće opisani koraci pretprocesiranja podataka, što uključuje inženjering karakteristika, rješavanje problema nedostajućih vrijednosti, detekciju i rješavanje problema *outlier*-a, kao i kodiranje kategorijskih podataka. Rješavanje ovih problema je izuzetno važno, jer sirovi podaci mogu da sadrže dosta nepravilnosti, pa su kao takvi neupotrebljivi u rješavanju razmatranih problema mašinskog

učenja. Takođe, biće obrađena i redukcija dimenzionalnosti, kao tehnika koja može značajno poboljšati performanse algoritama nenadgledanog učenja i omogućiti vizuelizaciju podataka. Kao glavni segment rada, koji sadrži ključne doprinose, biće opisana komparativna analiza algoritama za klasterizaciju. U ovom dijelu, biće detaljno predstavljeni rezultati klasterizacije korišćenjem različitih algoritama. Biće predstavljena komparativna analiza rezultata, te će se korišćenjem odgovarajućih metrika kvaliteta klasterizacije utvrditi algoritam koji postiže najbolje rezultate. Na kraju, u zaključku će biti predstavljeni ključni doprinosi rada i identifikovane teme budućih istraživanja.

2 Teorijske osnove klasterizacije

2.1 Uvod u mašinsko učenje

Mašinsko učenje (eng. *Machine Learning*) [7] je oblast u računarstvu koja se u najširem smislu bavi ekstrakcijom znanja iz podataka. Primjena mašinskog učenja je sveprisutna u svakodnevnom životu. Mnoge savremene web aplikacije koriste algoritme mašinskog učenja radi davanja različitih preporuka, poput preporuka za gledanje filmova, kupovinu proizvoda i slično. Složeni servisi, poput Facebook-a, Amazon-a ili Netflix-a, vrlo vjerovatno u svakom svom dijelu sadrže više modela mašinskog učenja.

Pored komercijalnih aplikacija, mašinsko učenje se često primjenjuje i u nauci. Neki od primjera upotrebe mašinskog učenja uključuju medicinu (rana dijagnoza bolesti, analiza medicinskih uzoraka, predikcija odgovora na terapiju itd), ekologiju, meteorologiju i proučavanje klimatskih promjena, psihologiju, društvene nauke (analiza ponašanja korisnika, obrada prirodnog jezika i sl), biologiju, matematiku i slično.

Mašinsko učenje je jedna od centralnih oblasti vještačke inteligencije, u kojoj se problemi rješavaju na osnovu istorijskih ili prethodnih primjera [8]. Za razliku od aplikacija vještačke inteligencije, mašinsko učenje podrazumijeva otkrivanje skrivenih obrazaca unutar podataka, koji se zatim koriste za klasifikaciju ili predikciju događaja vezanih za problem.

Metode mašinskog učenja se tradicionalno mogu podijeliti u dvije kategorije: učenje pod nadzorom (eng. *Supervised Learning*) i učenje bez nadzora (eng. *Unsupervised Learning*). Razlika između ove dvije kategorije leži u postojanju oznaka (labela) u skupu podataka za obuku.

Učenje pod nadzorom podrazumijeva postojanje skupa podataka sa unaprijed poznatim izlaznim atributom za svaki skup ulaznih atributa (parametara). To mogu biti, na primjer, oznake klasa kojima pripada posmatrani ulaz (problem klasifikacije), ili odgovarajuća izlazna vrijednost koja je vezana za odgovarajuću ulaznu vrijednost (problem regresije).

S druge strane, učenje bez nadzora se bavi prepoznavanjem obrazaca bez ciljnog atributa. To znači da se sve varijable koriste kao ulaz, pa se zbog tog pristupa ove tehnike dijele na: algoritme za klasterizaciju i algoritme za asocijativno rudarenje. Algoritmi učenja bez nadzora su posebno korisni za kreiranje oznaka unutar podataka koje se kasnije mogu koristiti za izvođenje zadataka za učenje pod nadzorom. Drugim riječima, algoritmi za klasterizaciju

identifikuju prirodne grupe u neoznačenim podacima i dodjeljuju oznake svakom podatku.

2.2 Klasterizacija u mašinskom učenju

Klasterizacija [9] je proces grupisanja skupa podataka tako da elementi unutar iste grupe (klastera) budu što sličniji jedni drugima, dok elementi iz različitih klastera treba da budu što različitiji. Klasterizacija je tehnika učenja bez nadzora, što znači da se grupe unaprijed ne znaju, već se otkrivaju automatski iz strukture podataka.

Ova tehnika može biti korisna za različite zadatke u mašinskom učenju, kao što su segmentacija slika, otkrivanje obrazaca u podacima, pretraživanje informacija i slično. Kod segmentacije podataka iz marketinških kampanja klasterizacija podataka se koristi radi grupisanja korisnika po sličnostima u ponašanju i potrošnji. U kontekstu segmentacije slika, klasterizacija vrši podjelu slike na djelove (regione) sa sličnim karakteristikama, kao što su boje i oblici. Ovo omogućava prepoznavanje objekata unutar slike. Klasterizacija se može koristiti i u svrhu grupisanja sličnih dokumenata, što je korisno u organizaciji informacija ili pretrazi.

Postoji više metoda klasterizacije podataka. Ove metode mogu se podijeliti u tri glavne grupe: particione metode, hijerarhijske metode i metode zasnovane na gustini podataka. Particione metode klasterizacije grupišu podatke na osnovu njihove udaljenosti. Najpopularniji algoritam iz ove grupe je K-means, dok su manje popularni K-medoids i K-medians. Hijerarhijska metoda klasterizacije podrazumijeva podjelu podataka na različite nivoe koji imaju oblik hijerarhije. Kod ove klasterizacije imamo dva načina podjele podataka:

1. odozdo nagore (eng. *Agglomerative*) - počinje sa svakom tačkom podataka kao zasebnim klasterom, a zatim postepeno spaja najbliže klastere;
2. odozgo nadolje (eng. *Divisive*) - počinje sa svim tačkama podataka u jednom klasteru, a zatim ih postepeno dijeli na manje klastere.

Za razliku od particionih i hijerarhijskih metoda, metode zasnovane na gustini podataka mogu formirati klastere proizvoljnih oblika. Ove metode u jedan klaster smještaju tačke koje se nalaze u reonu sa velikom gustinom. Algoritmi koji spadaju u ovu grupu metoda su DBSCAN i OPTICS (eng. *Ordering Points To Identify Clustering Structure*).

Svaki od ovih algoritama zahtijeva određene hiperparametre, koje je potrebno definisati prije pokretanja. K-means i hijerarhijska klasterizacija zahtijevaju definisanje broja klastera, dok DBSCAN zahtijeva izbor odgovarajućih hiperparametara za gustinu. Ovo može biti posebno izazovno, imajući u vidu da ovi hiperparametri mogu značajno uticati na konačne rezultate klasterizacije i performanse algoritama. Na primjer, izbor malog broja klastera bi mogao nepovoljno podijeliti podatke, dok izbor velikog broja klastera može smanjiti per-

formanse algoritma. Ovaj izazov se može prevazići korišćenjem različitih metoda za odabir optimalnih vrijednosti hiperparametara.

Uticaj šuma može biti značajan za performanse sistema. Šum (eng. *noise*), odnosno tačke koje značajno odstupaju od glavne strukture podataka, može izazvati probleme prilikom klasterizacije. Kod algoritama kao što je K-means, šum može pomjeriti centroide klastera ka nepovoljnim pozicijama. Pomeranje centroida ka nepovoljnim pozicijama može rezultirati neadekvatnom segmentacijom podataka – tačke koje bi prirodno pripadale jednom klasteru mogu biti dodijeljene drugom. S druge strane, ovaj problem je manje izražen kod algoritama zasnovanih na gustini podataka, kao što su DBSCAN i OPTICS. Ovi algoritmi su otporniji na šum jer ignorišu tačke koje ne pripadaju oblastima visoke gustine.

2.3 Metrike udaljenosti

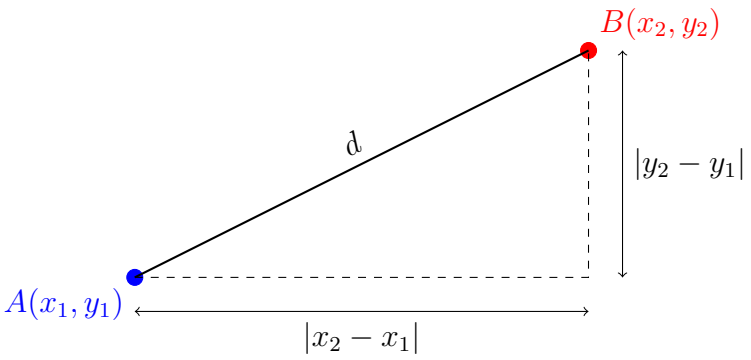
Metrike udaljenosti su jedan od neizbježnih alata u klasterizaciji. Pomoću metrika udaljenosti mjerimo sličnosti ili razlike između podataka, što je osnova grupisanja podataka u klastere. Odabir odgovarajuće metrike udaljenosti može značajno uticati na kvalitet klasterizacije, jer različite metrike mogu proizvesti različite rezultate čak i sa istim skupom podataka.

Metrike udaljenosti određuju način na koji se računa rastojanje između dvije tačke u prostoru karakteristika (eng. *features*), odnosno u prostoru u kojem je svaki podatak prestavljen kao vektor svojih atributa. U klasterizaciji, algoritmi koriste ove metrike da bi podatke dodijelili klasterima na osnovu njihove sličnosti. Najčešće korišćene metrike udaljenosti uključuju: Euklidovu distancu, Manhattan distancu, Minkowski distancu, maksimalnu udaljenost i slično. U nastavku će biti opisano nekoliko najpoznatijih metrika udaljenosti.

Euklidova distanca [10] je najčešće korišćena metrika u algoritmima klasterizacije, jer mjeri direktno rastojanje između tačaka u prostoru. Ona se koristi kada su podaci u koordinatnom prostoru sa pravolinijskim odnosima i kada je pogodno koristiti direktnu udaljenost između tačaka. Formula za Euklidovu distancu je:

$$d(\mathbf{x}, \mathbf{c}) = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2},$$

gdje je $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ tačka podataka, a $\mathbf{c} = [c_1, c_2, \ldots, c_n]$ centroid klastera.



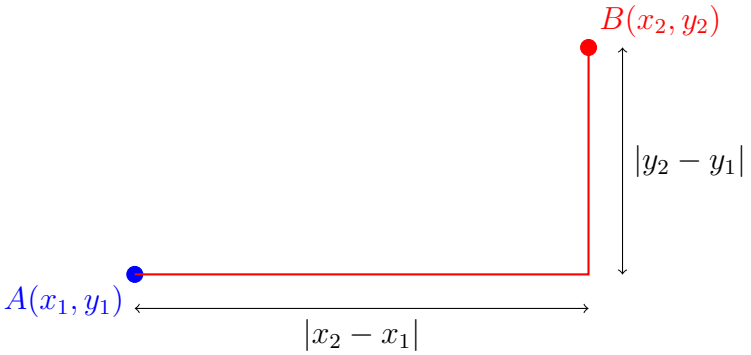
Slika 1: Ilustracija Euklidove distance

Na slici 1 ilustrovana je Euklidova distanca između tačaka A i B u dvodimenzionalnom prostoru, pri čemu je udaljenost između tačaka obilježena sa d.

Manhattan distanca (poznata kao ℓ_1-norma) [10] je metrika koja se koristi za izračunavanje udaljenosti tačaka u prostoru. Za razliku od Euklidove distance, Manhattan distanca mjeri udaljenost samo duž horizontalnih i vertikalnih pravaca (slično kretanju po gradskim ulicama). Ova osobina je posebno korisna kada je kretanje ograničeno na pravolinijske puteve duž koordinatnih osa. Formula za Manhattan distancu je:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|,$$

gdje je $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ druga tačka, od koje se mjeri rastojanje do tačke $\mathbf{x}$.

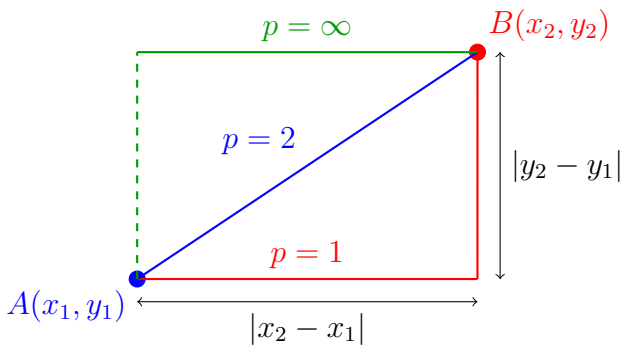


Slika 2: Ilustracija Manhattan distance

Slika 2 grafički prikazuje Manhattan distancu između dvije tačke u prostoru. Kao što je prethodno objašnjeno, Manhattan distanca mjeri rastojanje isključivo vertikalnim ili horizontalnim kretanjem.

Minkowski distanca [10] je generalizacija nekoliko drugih metrika, uključujući Euklidovu i Manhattan distancu. Ova metrika omogućava fleksibilnost u odabiru parametra p, koji određuje jačinu metrike. Kada je $p = 1$, Minkowski distanca postaje Manhattan distanca, dok za $p = 2$ postaje Euklidova distanca. Formula za Minkowki distancu je:

$$d(\mathbf{x}, \mathbf{c}) = \left(\sum_{i=1}^{n} |x_i - c_i|^p \right)^{1/p}.$$



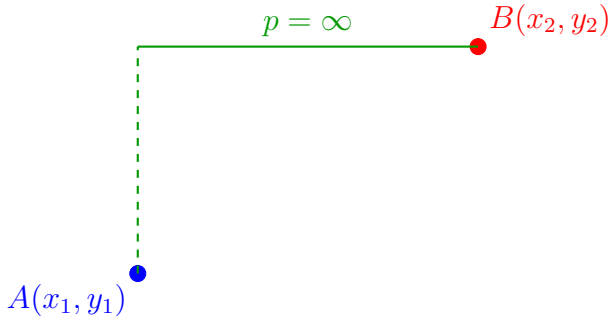
Slika 3: Ilustracija Minkowki distance za različite vrijednosti parametra p

Slika 3 ilustruje Minkowski distancu. Dakle, kada je parametar $p = 1$ imamo Manhattan distancu, odnosno kretanje u pravim pravcima. Takođe, kada je $p = 2$ riječ je o Euklidovoj distanci koja mjeri „pravu“ udaljenost između dvije tačke u prostoru. U slučaju kada je $p = \infty$, imamo Čebiševljevu distancu koja se računa kao maksimalna razlika između bilo koje dvije koordinate.

Maksimalna udaljenost (poznata i kao Čebišev distanca) [10] je posebna vrsta Minkowski distance koja se koristi kada je $p = \infty$. Ova metrika mjeri apsolutnu razliku između koordinata tačaka, odnosno u dimenziji u kojoj je razlika između tačaka najveća. Formula za ovu distancu je:

$$d(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq n} (|x_i - y_i|,$$

gdje $|x_i - y_i|$ predstavlja apsolutnu razliku između koordinata tačaka $\mathbf{x}$ i $\mathbf{y}$ u dimenziji i.



Slika 4: Ilustracija maksimalne distance

Slika 4 ilustruje maksimalnu distancu. Između dvije tačke prikazana je distanca koja se mjeri kao maksimalna vrijednost između razlika njihovih koordinata u svim dimenzijama. U dvodimenzionalnom prostoru, ova distanca se računa kao veća od razlike u horizontalnom i vertikalnom smjeru.

2.4 Pregled algoritama za klasterizaciju

2.4.1 K-means algoritam

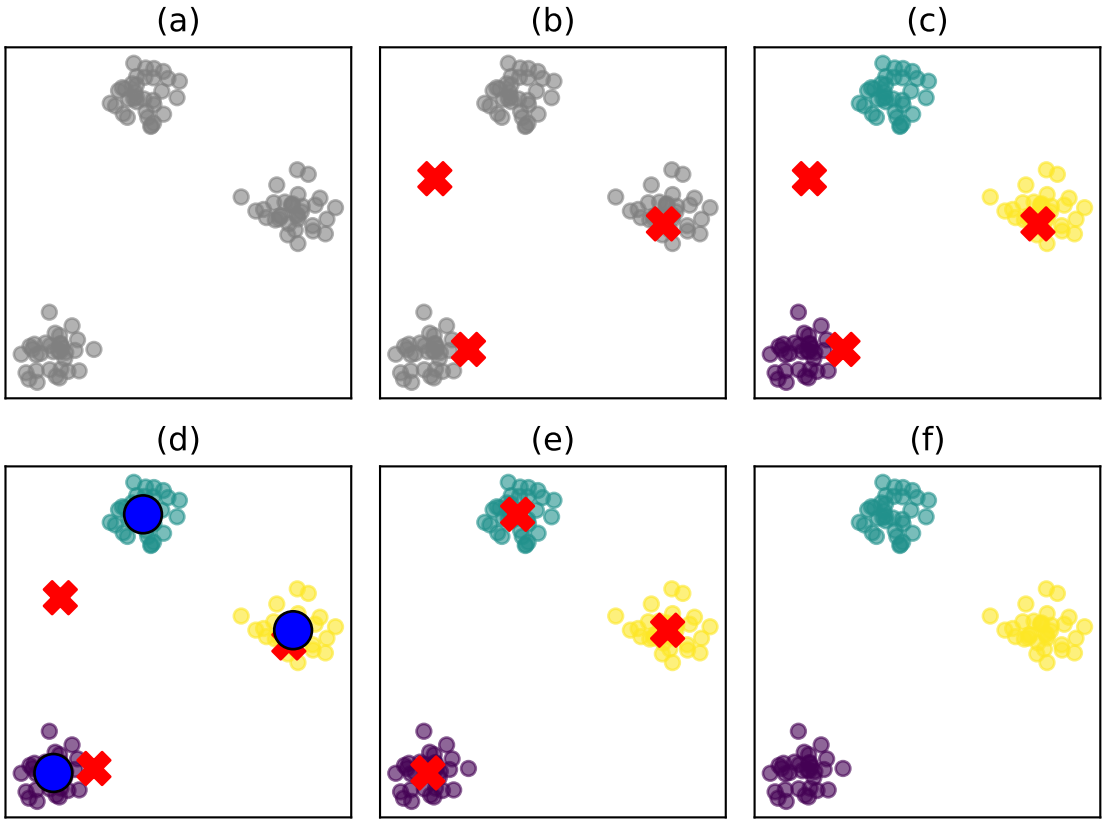
K-means algoritam [11] je jedan od najčešće korišćenih algoritama za klasterizaciju i pripada grupi particionih metoda. Algoritam počinje izborom K tačaka kao inicijalnih centroida, koje se obično proizvoljno pozicioniraju. Nakon ovog koraka, svaka tačka se dodjeljuje najbližem centroidu na osnovu metrike udaljenosti. Najčešće korišćena metrika udaljenosti u slučaju K-means algoritma je Euklidova distanca. Kada se klasteri formiraju, pozicije centroida svakog klastera se ažuriraju tako što se računa prosječna vrijednost svih tačaka koje pripadaju tom klasteru. Algoritam iterativno ponavlja ova dva koraka sve dok se ne postigne uslov konvergencije. Uslov konvergencije može biti ispunjen kada se:

- pozicije centroida više ne mijenjaju,
- broj tačaka u klasterima ne mijenja,
- dostigne unaprijed definisan broj iteracija, ili neki drugi uslov.

K-means je pohlepni algoritam koji garantuje konvergenciju ka lokalnom minimumu, što znači da ne mora uvijek naći globalni minimum (najbolje moguće rješenje), jer zavisi od početnih pozicija centroida. Ako su centroidi postavljeni blizu lokalnih minimuma, algoritam može ostati „zarobljen“ u tom regionu i pružiti neoptimalne rezultate. Pseudokod 1 daje pregled osnovnog K-means algoritma.

Pseudokod 1 K-means algoritam

1: Izaberi K tačaka kao početne centroide
2: **repeat**
3: Formiraj K klastera dodjeljujući svaku tačku njenom najbližem centroidu.
4: Ponovo izračunaj pozicije centroida svakog klastera.
5: **until** kriterijum konvergencije bude ispunjen.



Slika 5: Ilustracija izvršavanja K-means algoritma kroz nekoliko iteracija

Na slici 5 prikazan je iterativni proces rada K-means algoritma kroz nekoliko faza grupisanja podataka. Na slici 5 (a) prikazano je početno stanje, prije primjene algoritma, kada tačke još uvijek nisu dodijeljene klasterima. Slika 5 (b) prikazuje nasumično postavljanje centroida. Na slici 5 (c) prvi put dolazi do dodjeljivanja tačaka klasterima, na osnovu njihove udaljenosti od centroida. Slika 5 (d) predstavlja pozicije ažuriranih centroida (plavi krugovi) u odnosu na prethodne (crveni X-ovi). Slika 5 (e) ilustruje novo stanje klastera nakon dodjele tačaka na osnovu ažuriranih centroida. Nakon ovog koraka, smatra se da je postignut uslov konvergencije. Na kraju, slika 5 (f) prikazuje rezultujuće stanje nakon primjene K-means algoritma.

Jedan od glavnih ciljeva K-means algoritma je minimizacija sume kvadratnih grešaka (eng. *Sum of Squared Errors* - SSE). Suma kvadratnih grešaka je metrika koja se koristi za procjenu kvaliteta klasterizacije kod K-means algoritma. Minimizacijom SSE-a klasteri se oblikuju tako da tačke unutar svakog klastera budu što bliže svom centroidu. Ako je dat skup podataka $\mathbb{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ koji se sastoji od N tačaka, a klastere dobijene nakon primjene K-means algoritma označimo sa $\mathbb{C} = \{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K\}$, suma kvadratnih grešaka se računa pomoću sledeće formule:

$$SSE = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathbb{C}_k} ||\mathbf{x}_i - \mathbf{c}_k||^2,$$

gdje:

- $\mathbf{c}_k$ predstavlja centroid klastera $\mathbb{C}_k$;
- $\|\mathbf{x}_i - \mathbf{c}_k\|^2$ predstavlja kvadrat Euklidove distance između tačke $\mathbf{x}_i$ i centroida $\mathbf{c}_k$;
- prva suma računa grešku za svaki klaster $\mathbb{C}_k$;
- druga suma računa grešku za svaku tačku unutar datog klastera.

2.4.1.1 Faktori koji utiču na K-means algoritam

Ključni faktori koji utiču na performanse K-means algoritma su izbor početnih centroida i procjena broja klastera. Izbor početnih centroida značajno utiče na krajnji rezultat. Loše postavljeni centroidi će uzrokovati konvergenciju ka lošem lokalnom minimumu. Takođe, izbor premalo ili previše klastera može dovesti do nepravilne segmentacije podataka. Postoje različiti pristupi u rješavanju ovih problema, uključujući metode za višestruko inicijalizovanje centroida i korišćenje metoda, poput metode „lakta“, za određivanje optimalnog broja klastera.

Najčešće korišćene metode inicijalizacije su nasumična inicijalizacija i *k-means++* inicijalizacija:

1. Nasumična inicijalizacija je osnovna metoda inicijalizacije u K-means algoritmu koja se često koristi u praksi zbog svoje jednostavnosti i brzine. Ova metoda funkcioniše tako što se početni centroidi biraju nasumično iz skupa podataka. Svaka tačka može postati centroid, dok je broj centroida definisan brojem K, koji predstavlja broj klastera koje algoritam treba da formira. Nedostaci nasumične inicijalizacije su mogućnost lošeg rasporeda centroida, što može dovesti do loših rješenja. Takođe, ova metoda inicijalizacije moze rezultirati velikim brojem iteracija, što utiče na ukupnu efikasnost algoritma.

2. *k-means++* [12] inicijalizacija je poboljšana metoda inicijalizacije koja je razvijena kako bi se riješio problem loše inicijalizacije koji se javlja kod nasumične inicijalizacije. Glavni cilj ove metode je poboljšanje početne raspodjele centroida, čime se smanjuje vjerovatnoća da algoritam završi u lokalnom minimumu. Princip rada *k-means++* metode se zasniva na selekciji početnih centroida na način koji favorizuje tačke koje su daleko od već odabranih centroida. Proces počinje nasumičnim odabirom jednog od centroida. Nakon toga, za svaku tačku se računa vjerovatnoća da bude izabrana kao naredni centroid. Ova vjerovatnoća je veća za tačke koje su dalje od postojećih centroida. Prednosti ove metode inicijalizacije su: značajno smanjuje vjerovatnoću da K-means završi u lošem lokalnom minimumu, korišćenje bolje inicijalizacije obično dovodi do brže konvergencije algoritma, smanjujući broj potrebnih iteracija i smanjuje

varijaciju rezultata u odnosu na nasumičnu inicijalizaciju. Nedostaci su: zahtijeva dodatno vrijeme za izračunavanje udaljenosti za svaku tačku i nije garantovano da će uvijek pronaći globalni minimum.

Izbor optimalnog broja klastera je drugi najvažniji faktor koji utiče na K-means algoritam. Iako K-means algoritam nije u stanju da automatski odredi optimalan broj klastera, postoje različite metode koje pomažu u definisanju ovog hiperparametra. Za rješavanje ovog problema često se koriste metode kao što su metoda „lakta“ (eng. *Elbow Method*) i *Silhouette* analiza, koje će biti detaljno opisane u dijelu rada posvećenom metodama za odabir optimalnih vrijednosti hiperparametara.

2.4.1.2 Prednosti i nedostaci K-means algoritma

K-means algoritam je poznat po svojoj jednostavnosti i efikasnosti. Njegova intuitivna implementacija i brzina izvođenja, posebno na velikim skupovima podataka, doprinose njegovoj širokoj primjeni. Ovaj algoritam koristi se u mnogim domenima, od marketinga do biologije. Međutim, pored navedenih prednosti, algoritam ima i nekoliko značajnih nedostataka koje treba uzeti u obzir prilikom primjene.

Jedan od glavnih nedostataka K-means algoritma je njegova osjetljivost na inicijalizaciju centroida, što može uticati na konačni rezultat klasterizacije, kao što je prethodno opisano. Pored toga, potrebno je unaprijed odabrati broj klastera prije primjene K-means algoritma, što nije uvijek intuitivno. Poseban izazov predstavljaju tačke koje značajno odstupaju od ostalih, tj. izolovane tačke (eng. *outlier*). Ove tačke mogu dovesti do značajne promjene pozicija centroida i time narušiti kvalitet klasterizacije.

2.4.2 Aglomerativni hijerarhijski algoritam klasterizacije

Hijerahijski algoritmi klasterizacije [11] razvijeni su kako bi se prevazišli nedostaci particionih metoda. Hijerarhijske metode možemo podijeliti u dvije grupe: aglomerativne i divizivne. Aglomerativne metode počinju tako što svaku tačku dodijeljuju po jednom klasteru. Nakon toga, iterativno spajaju po dva klastera, čime prave hijerarhiju klastera od dna prema vrhu. Suprotno tome, divizivne metode počinju sa jednim velikim klasterom koji sadrži sve tačke. Divizivni algoritmi iterativno dijele veliki klaster na dva dijela, generišući hijerarhiju klastera od vrha prema dnu.

Hijerarhija klastera se može opisati terminologijom binarnih stabala. Korijen stabla predstavlja veliki klaster koji se dobija spajanjem manjih klastera i to čini vrh hijerarhije (nivo 0). Na svakom narednom nivou, čvorovi predstavljaju kako su se klasteri spajali tokom cijelog procesa. Osnova hijerarhije sastoji se od pojedinačnih tačaka koje čine listove stabla. Pomenuta hijerarhijska reprezentacija klastera se takođe naziva dendrogram. Osnovna prednost

hijerarhijskih metoda je to što omogućavaju da se na bilo kojem nivou hijerarhije „presječe“ dendrogram i dobije odgovarajući skup klastera.

Aglomerativni algoritam počinje tako što se svaka tačka podataka dodjeljuje po jednom klasteru. Koristeći određenu metriku udaljenosti, formira se matrica distanci koja pokazuje rastojanje između klastera. Najbliži klasteri se spajaju na svakom nivou, a matrica distanci se ažurira u skladu s tim. Ovaj postupak nastavlja se sve dok se ne dobije finalni veliki klaster koji sadrži sve tačke podataka. Veliki klaster predstavlja vrh dendrograma i označava završetak spajanja klastera. Pseudokod 2 daje pregled aglomerativnog hijerarhijskog algoritma.

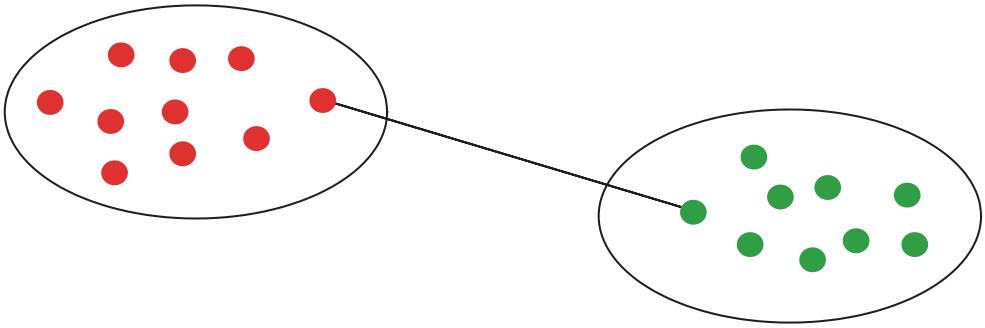
Pseudokod 2 Aglomerativni hijerahijski algoritam

1: Izračunati matricu distanci između svih tačaka podataka
2: **repeat**
3: Pronaći dva najbliža klastera i označiti ih kao C_a i C_b
4: Spojiti klastere kao $C_{a\cup b} = C_a \cup C_b$. Postaviti kardinalnost novog klastera kao $N_{a\cup b} = N_a + N_b$
5: Dodati novi red i novu kolonu koji sadrže rastojanja između novog klastera $C_{a\cup b}$ i preostalih klastera.
6: **until** ostao samo jedan veliki klaster.

U nastavku će biti opisane metode spajanja odnosno različite strategije za određivanje distanci ili sličnosti između klastera. Najpoznatije metode spajanja su: *single linkage* (najbliži susjed), *complete linkage* (najdalji susjed), *average linkage* (prosječna veza) i Vardova metoda (*Ward's method*).

2.4.2.1 Metode spajanja

Single-linkage [13] (slika 6) metoda je jedna od najednostavnijih metoda spajanja kod aglomerativnog hijerarhijskog algoritma klasterizacije. Ova metoda spaja tačke koje imaju najmanju međusobnu udaljenost, odnosno najveću sličnost, i tako formira prvi klaster. U narednom koraku se mogu dogoditi dva scenarija: ili se već formiranom klasteru od dvije tačke pridružuje i treća tačka, ili se dvije najbliže, još uvijek neklasterisane tačke, spajaju i dodaju u drugi klaster. Odluka zavisi od toga da li je udaljenost neke od neklasterisanih tačaka od prvog klastera manja od udaljenosti između dvije najbliže neklasterisane tačke. Proces se nastavlja dok sve tačke ne budu uključene u jedan jedinstveni klaster.



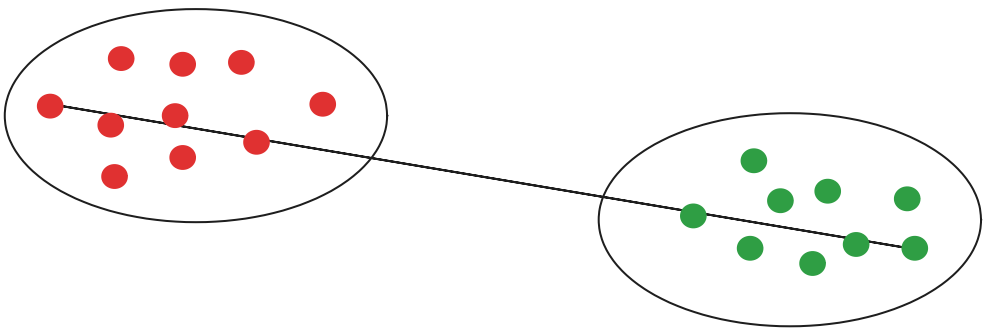
Slika 6: Ilustracija *single linkage* metode spajanja

Matematički, *single linkage* se izražava kao:

$$d_{\min}(\mathbb{C}_a, \mathbb{C}_b) = \min_{\mathbf{x}\in\mathbb{C}_a, \mathbf{y}\in\mathbb{C}_b} d(\mathbf{x}, \mathbf{y}),$$

gdje $\mathbb{C}_a$ i $\mathbb{C}_b$ predstavljaju klastere između kojih se računa rastojanje. Promjenljive $\mathbf{x}$ i $\mathbf{y}$ predstavljaju pojedinačne tačke podataka takve da $\mathbf{x} \in \mathbb{C}_a$ i $\mathbf{y} \in \mathbb{C}_b$, odnosno pripadaju klasterima $\mathbb{C}_a$ i $\mathbb{C}_b$. Rastojanje između ovih tačaka koristi se pri određivanju minimalnog rastojanja između dva klastera.

Za razliku od *single linkage*-a, *complete linkage* [13] (slika 7) koristi udaljenost između najudaljenijih tačaka. Ova metoda osigurava da su sve tačke unutar klastera na udaljenostima koje ne prelaze istu maksimalnu vrijednost. Postupak počinje pronalaskom dvije tačke koje su međusobno najsličnije, tj. imaju najmanju udaljenost, i te tačke se dodaju u prvi klaster. Zatim se računa udaljenost između novog klastera i svih ostalih tako što se uzima najveća udaljenost između bilo koje tačke iz tog klastera i neke druge tačke iz drugog klastera. Ova vrijednost se zatim upisuje u matricu distanci i koristi se kao novo međuklastersko rastojanje prilikom određivanja sledećeg para klastera koji će biti spojen. Matrica distanci se nakon svakog spajanja ažurira. Kompletan proces spajanja se završava kada sve tačke postanu dio jednog velikog klastera. Tokom procesa bilježi se koje su grupe spojene i pri kojim udaljenostima je došlo do spajanja.



Slika 7: Ilustracija *complete linkage* metode spajanja

Odgovarajuća jednačina koja se koristi kod *complete linkage*-a je sledeća:

$$d_{\max}(\mathbb{C}_a, \mathbb{C}_b) = \max_{\mathbf{x}\in\mathbb{C}_a, \mathbf{y}\in\mathbb{C}_b} d(\mathbf{x}, \mathbf{y}).$$

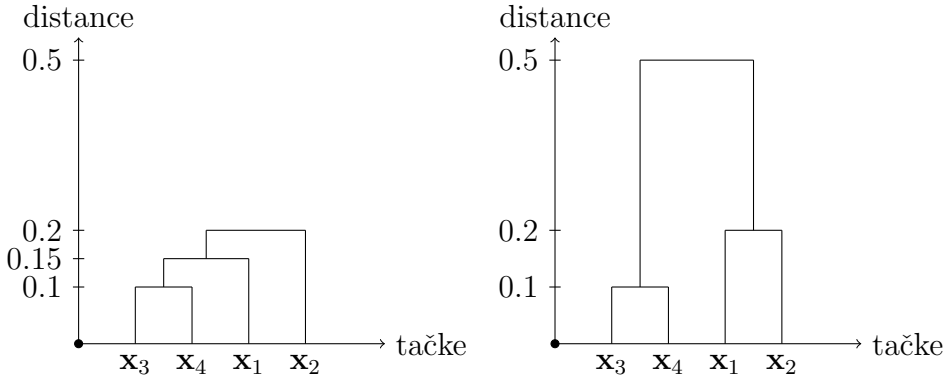
Ova formula se koristi za izračunavanje maksimalne udaljenosti između klastera $\mathbb{C}_a$ i $\mathbb{C}_b$. U okviru izraza, $\mathbf{x}$ i $\mathbf{y}$ predstavljaju pojedinačne tačke koje pripadaju klasterima $\mathbb{C}_a$ i $\mathbb{C}_b$, respektivno. Za svaki mogući par tačaka se izračunava međusobna udaljenost, nakon čega se uzima najveća vrijednost među njima.

U nastavku je opisan primjer koji pokazuje razlike u korišćenju *single linkage* i *complete linkage* metoda spajanja.

Neka je data matrica distanci $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} 0.0 & 0.20 & 0.15 & 0.30 \\ 0.20 & 0.0 & 0.40 & 0.50 \\ 0.15 & 0.40 & 0.0 & 0.10 \\ 0.30 & 0.50 & 0.10 & 0.0 \end{bmatrix},$$

koja prikazuje distance između tačaka $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ i $\mathbf{x}_4$ u proizvoljnom skupu podataka, pri čemu prvi red i prva kolona odgovaraju tački $\mathbf{x}_1$, drugi red i druga kolona tački $\mathbf{x}_2$ i tako dalje.



(a) *Single linkage* (b) *Complete linkage*

Slika 8: Ilustracija aglomerativnog algoritma klasterizacije. (a) Izgled dendrograma nakon primjene *single linkage* metode spajanja. (b) Izgled dendrograma nakon primjene *complete linkage* metode spajanja.

Slika 8 prikazuje odgovarajuća dva dendrograma dobijena korišćenjem *single linkage* i *complete linkage* metoda spajanja na osnovu prethodno definisane matrice distanci $\mathbf{D}$. Na prikazanim dendrogramima apscisna osa prikazuje tačke podataka, dok ordinatna predstavlja udaljenost na kojoj su tačke ili klasteri spojeni. Razlika u dendrogramima nastaje zbog specifičnih kriterijuma koje koriste ove dvije metode. Distanca između podataka se može računati pomoću različitih metrika udaljenosti, poput Euklidove distance, Manhattan distance ili druge. Proces spajanja klastera biće opisan u nekoliko koraka. U prvom koraku,

obije metode spajaju klastere $\mathbf{x}_3$ i $\mathbf{x}_4$, jer ti klasteri imaju najmanju distancu u matrici distanci. Nakon spajanja, drugi korak je ažuriranje matrica distanci za obije metode (tabele 1 i 2).

Tabela 1: Ažurirana matrica distanci za *single linkage* nakon prvog spajanja

	$\mathbf{x}_1$	$\mathbf{x}_2$	$(\mathbf{x}_3, \mathbf{x}_4)$
$\mathbf{x}_1$	0	0,20	$\min(d(\mathbf{x}_3, \mathbf{x}_1), d(\mathbf{x}_4, \mathbf{x}_1)) = 0,15$
$\mathbf{x}_2$	0,20	0	$\min(d(\mathbf{x}_3, \mathbf{x}_2), d(\mathbf{x}_4, \mathbf{x}_2)) = 0,40$
$(\mathbf{x}_3,\mathbf{x}_4)$	0,15	0,40	0

Tabela 2: Ažurirana matrica distanci za *complete linkage* nakon prvog spajanja

	$\mathbf{x}_1$	$\mathbf{x}_2$	$(\mathbf{x}_3, \mathbf{x}_4)$
$\mathbf{x}_1$	0	0,20	$\max(d(\mathbf{x}_3, \mathbf{x}_1), d(\mathbf{x}_4, \mathbf{x}_1)) = 0,30$
$\mathbf{x}_2$	0,20	0	$\max(d(\mathbf{x}_3, \mathbf{x}_2), d(\mathbf{x}_4, \mathbf{x}_2)) = 0,50$
$(\mathbf{x}_3,\mathbf{x}_4)$	0,30	0,50	0

Treći korak je spajanje klastera koji imaju najmanju distancu u ažuriranoj matrici distanci. Kod *single linkage* metode, spajaju se klasteri $(\mathbf{x}_3, \mathbf{x}_4)$ i $\mathbf{x}_1$, dok se kod *complete linkage* metode spajaju klasteri $\mathbf{x}_1$ i $\mathbf{x}_2$. Nakon novog spajanja klastera, u četvrtom koraku se ponovo ažuriraju matrice distanci (tabele 3 i 4).

Tabela 3: Ažurirana matrica distanci za *single linkage* nakon drugog spajanja

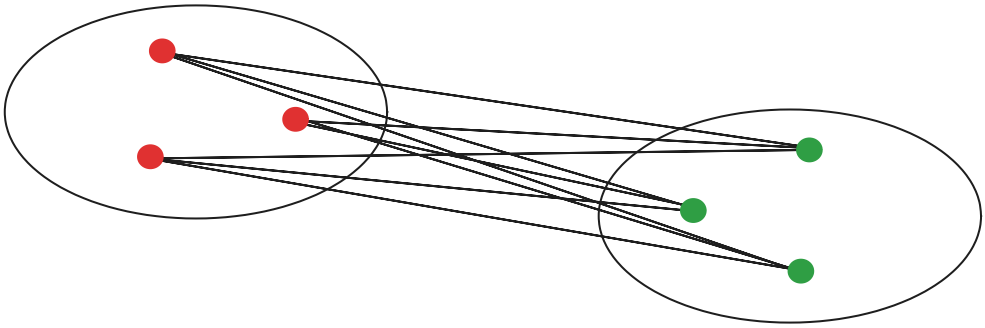
	$(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_1)$	$\mathbf{x}_2$
$(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_1)$	0	$\min(d(\mathbf{x}_3, \mathbf{x}_2), d(\mathbf{x}_4, \mathbf{x}_2), d(\mathbf{x}_1, \mathbf{x}_2)) = 0,20$
$\mathbf{x}_2$	0,20	0

Tabela 4: Ažurirana matrica distanci za *complete linkage* nakon drugog spajanja

	$(\mathbf{x}_3, \mathbf{x}_4)$	$(\mathbf{x}_1, \mathbf{x}_2)$
$(\mathbf{x}_3, \mathbf{x}_4)$	0	$\max(d(\mathbf{x}_3, \mathbf{x}_1), d(\mathbf{x}_3, \mathbf{x}_2), d(\mathbf{x}_4, \mathbf{x}_1), d(\mathbf{x}_4, \mathbf{x}_2)) = 0,50$
$(\mathbf{x}_1, \mathbf{x}_2)$	0,50	0

U poslednjem koraku, spajamo klastere $(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_1)$ i $\mathbf{x}_2$ kod *single linkage* metode, odnosno $(\mathbf{x}_3, \mathbf{x}_4)$ i $(\mathbf{x}_1, \mathbf{x}_2)$ kod *complete linkage* metode.

Average linkage [13] (slika 9) je još jedna poznata metoda spajanja. Prema ovoj metodi, udaljenost između dva klastera mjeri se kao prosječna udaljenost između svih parova tačaka, pri čemu svaka tačka iz jednog klastera ima svog odgovarajućeg para u drugom klasteru (slika 9).



Slika 9: Ilustracija *average linkage* metode spajanja

Matematički, za klastere $\mathbb{C}_a$ i $\mathbb{C}_b$ udaljenost se računa kao:

$$d_{avg}(\mathbb{C}_a, \mathbb{C}_b) = \frac{1}{|\mathbb{C}_a| \cdot |\mathbb{C}_b|} \sum_{\mathbf{x} \in \mathbb{C}_a} \sum_{\mathbf{y} \in \mathbb{C}_b} d(\mathbf{x}, \mathbf{y}).$$

Metoda *average linkage* predstavlja kompromis između *single linkage* i *complete linkage* metoda. Dok *single linkage* koristi samo najbliže tačke iz dva klastera, a *complete linkage* koristi najudaljenije, *average linkage* koristi sve tačke iz oba klastera, čime omogućava uravnoteženiji prikaz udaljenosti. Nedostatak ove metode je što zahtijeva više računarskih resursa, jer uključuje izračunavanje prosjeka svih parova tačaka između klastera, što može biti zahtjevno za velike skupove podataka.

Vardova metoda (eng. *Ward's method*) [13] (slika 10) predstavlja metodu spajanja koja formira raspodjelu klastera na način koji minimizira gubitak informacija pri svakom spajanju. Gubitak informacija se najčešće kvantifikuje korišćenjem kriterijuma zbira kvadratnih grešaka (eng. *Error Sum of Squares* – ESS). Za dati klaster $\mathbb{C}$, zbir kvadrata grešaka računa se kao:

$$\text{ESS}(\mathbb{C}) = \sum_{\mathbf{x} \in \mathbb{C}} \|\mathbf{x} - \boldsymbol{\mu}(\mathbb{C})\|^2 = \sum_{\mathbf{x} \in \mathbb{C}} \mathbf{x}^T \mathbf{x} - |\mathbb{C}|\, \boldsymbol{\mu}(\mathbb{C})^T \boldsymbol{\mu}(\mathbb{C}),$$

gdje je:

- $\mathbf{x}$ – tačka podataka koja pripada klasteru $\mathbb{C}$;
- $\mathbb{C}$ – klaster, skup tačaka koje su grupisane zajedno;
- $\boldsymbol{\mu}(\mathbb{C})$ – srednja vrijednost (centroid) svih tačaka u klasteru $\mathbb{C}$, izračunata kao

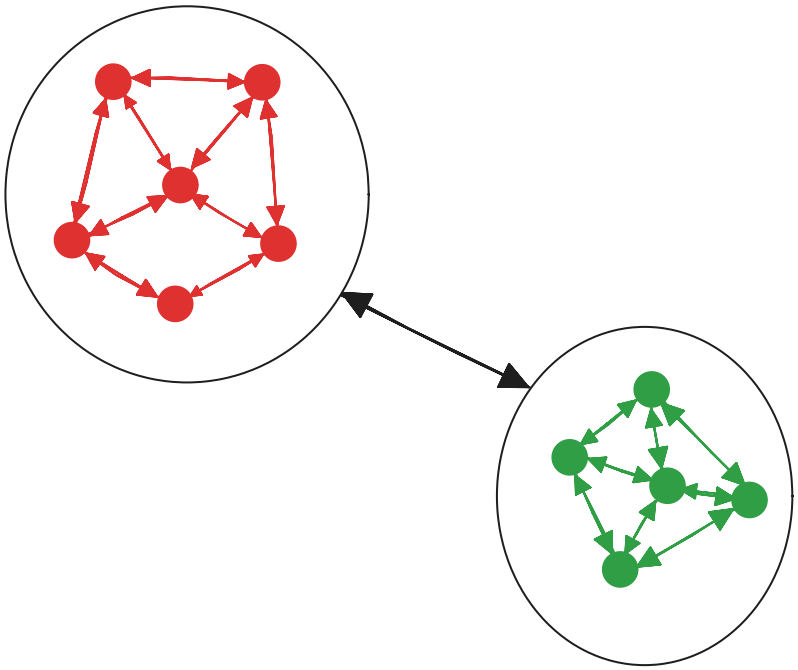
$$\boldsymbol{\mu}(\mathbb{C}) = \frac{1}{|\mathbb{C}|} \sum_{\mathbf{x} \in \mathbb{C}} \mathbf{x};$$

- $|\mathbb{C}|$ – broj tačaka u klasteru $\mathbb{C}$;
- $ESS(\mathbb{C})$ – suma kvadrata grešaka unutar klastera $\mathbb{C}$ koja prestavlja mjeru ukupne disperzije tačaka u odnosu na centroid klastera.

Pretpostavimo da na jednom nivou klasterizacije postoji k grupa $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_k$, tada se ukupni gubitak informacije izražava kao:

$$ESS = \sum_{i=1}^{k} ESS(\mathbb{C}_i),$$

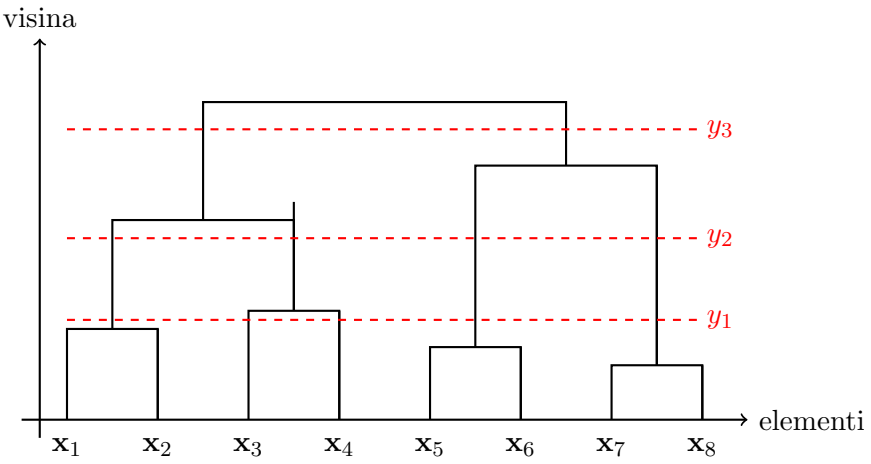
što predstavlja zbir unutar-klasterskih ESS vrijednosti. Na svakom koraku Vardove metode razmatraju se svi mogući parovi grupa i spajaju se one dvije čije spajanje izaziva najmanje povećanje gubitka informacija.



Slika 10: Ilustracija Vardove metode spajanja

2.4.2.2 „Sječenje“ dendrograma

„Sječenje“ dendrograma predstavlja jedan od najvažnijih koraka u procesu hijerarhijske klasterizacije. Kao što je ranije opisano, dendrogram je vizuelna reprezentacija hijerarhijske klasterizacije, gdje ordinatna osa predstavlja udaljenost na kojoj su klasteri spojeni, dok apscisna prikazuje podatke, odnosno klastere. Da bi se odredio željeni broj klastera, dendrogram se „siječe“ horizontalnom linijom. Visina se bira tako da odražava željeni nivo sličnosti između elemenata unutar klastera. Na primjer, ako se linija povuče nisko, dobijamo veći broj malih klastera, dok ako se linija povuče visoko dobija se manji broj većih klastera.



Slika 11: Prikaz dendrograma sa tri presječne visine y_1, y_2 i y_3.

Slika 11 prikazuje dendrogram koji je presječen pomoću tri horizontalne isprekidane crvene linije, u tri različita slučaja. U slučaju kada je linija obilježena sa y_1, dobijamo 5 manjih klastera. Zatim, u slučaju kada je linija obilježena sa y_2, imamo 4 klastera srednje veličine. Na kraju, linija obilježena sa y_3 siječe dendrogram na mjestu gdje imamo 2 velika klastera.

Iako je ovaj pristup jednostavan za implementaciju, često ne koristi pun potencijal hijerarhijske informacije sadržane u dendrogramu. „Sječenje“ dendrograma može dovesti do gubitka relevantnih informacija, jer zanemaruje dublje obrasce unutar podataka [14]. Umjesto toga, preporučljivo je korišćenje cijele hijerarhijske strukture za analizu, kao i metode za izbor optimalnog broja klastera zasnovane na metrikama gubitka koje kvantifikuju koliko informacija se gubi prilikom grupisanja.

2.4.2.3 Prednosti i nedostaci aglomerativnog hijerarhijskog algoritma klasterizacije

Jedna od glavnih prednosti aglomerativnog hijerarhijskog algoritma klasterizacije je njegova fleksibilnost, jer ne zahtijeva unaprijed definisan broj klastera. Ovo omogućava korisnicima da analizom dendrograma donesu odluku o broju klastera. Takođe, algoritam može biti primijenjen na različite skupove podataka, jer nije zavisan od unaprijed definisanih oblika klastera, što ga čini pogodnim za grupisanje podataka različitih geometrijskih oblika. Jednostavan je za implementaciju i može se koristiti u različitim scenarijima.

Pored navedenih prednosti, ovaj algoritam ima i određene nedostatke. Može biti računski „skup“, posebno u slučajevima velikih skupova podataka, zbog potrebe za izračunavanjem matrice distanci i njenim ažuriranjem na svakom koraku spajanja. Takođe, osjetljiv je na šum i izolovane tačke (eng. *ouliers*), jer spajanje klastera zavisi od sličnosti između njihovih članova, a *outlier*-i mogu značajno uticati na rezultate. Osim toga, algoritam može imati problema sa identifikacijom klastera u situacijama kada su klasteri vrlo različiti u veličini, jer se osnovne mjere sličnosti mogu koncentrisati na spajanje manjih, gustih klastera, zane-

marujući šire i rasprostranjenije grupe.

2.4.3 DBSCAN algoritam

DBSCAN (eng. *Density-Based Spatial Clustering of Applications with Noise*) [15] algoritam jedan je od najpoznatijih algoritama klasterizacije zasnovanih na gustini podataka. Metode zasnovane na gustini podataka identifikuju klastere na osnovu gustine tačaka podataka u prostoru. Za razliku od tradicionalnih metoda, kao što su particione i hijerarhijske metode koje pretpostavljaju sferni ili eliptični oblik klastera, algoritmi zasnovani na gustini mogu otkriti klastere proizvoljnih oblika.

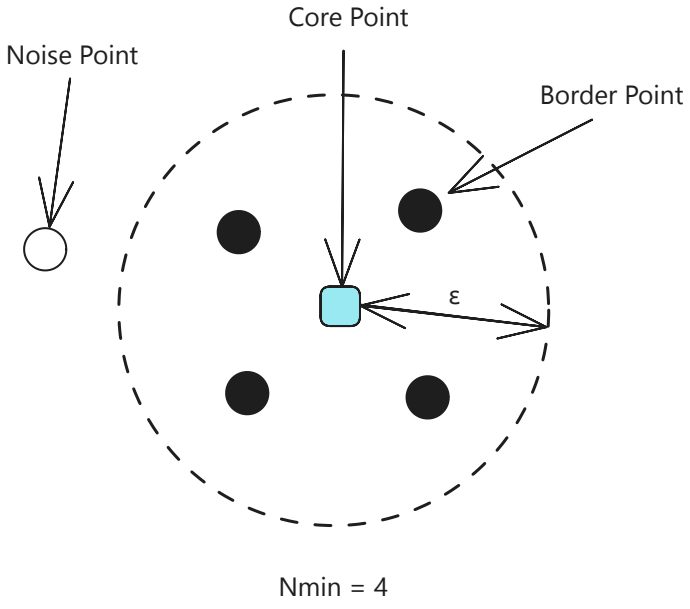
DBSCAN algoritam procjenjuje gustinu brojanjem tačaka unutar susjedstva definisanog fiksnim radijusom (ε) i smatra se da su dvije tačke povezane ukoliko se nalaze u međusobnim susjedstvima. Tačke se dijele u tri kategorije:

- *Core Point* - tačke oko kojih se formira krug prečnika ε, a koje sadrže najmanje N_{min} tačaka u okviru kruga, uključujući i nju samu.
- *Border* (*Non-Core*) *Point* - tačka koja se nalazi unutar kruga formiranog oko *Core Point*-a.
- *Noise* - tačka koja nije ni *Core Point* ni *Border Point* smatra se šumom i ne pripada nijednom klasteru.

Proces klasterizacije počinje nasumičnim odabirom tačke, za koju se provjerava da li ispunjava uslov da bude *Core Point* (da ima najmanje N_{min} tačaka u svom susjedstvu). Ako tačka zadovoljava taj uslov, postavlja se kao *Core Point* i dodaje u prvi klaster. Sve tačke unutar kruga koji je formiran oko *Core Point*-a (koje su gustinski dostupne, tj. *density-reachable*) se takođe dodaju u klaster. *Border Point*-i takođe postaju dio klastera, ali ga ne mogu proširiti jer nemaju dovoljno susjeda. Tačke koje nisu dodijelene nijednom klasteru smatraju se šumom. Ovaj proces se ponavlja sve dok se ne obrade sve tačke u skupu podataka. Na kraju, sve tačke su ili dodijeljene nekom klasteru, ili označene kao šum. Pregled DBSCAN algoritma dat je u Pseudokodu 3.

Pseudokod 3 DBSCAN algoritam za klasterizaciju

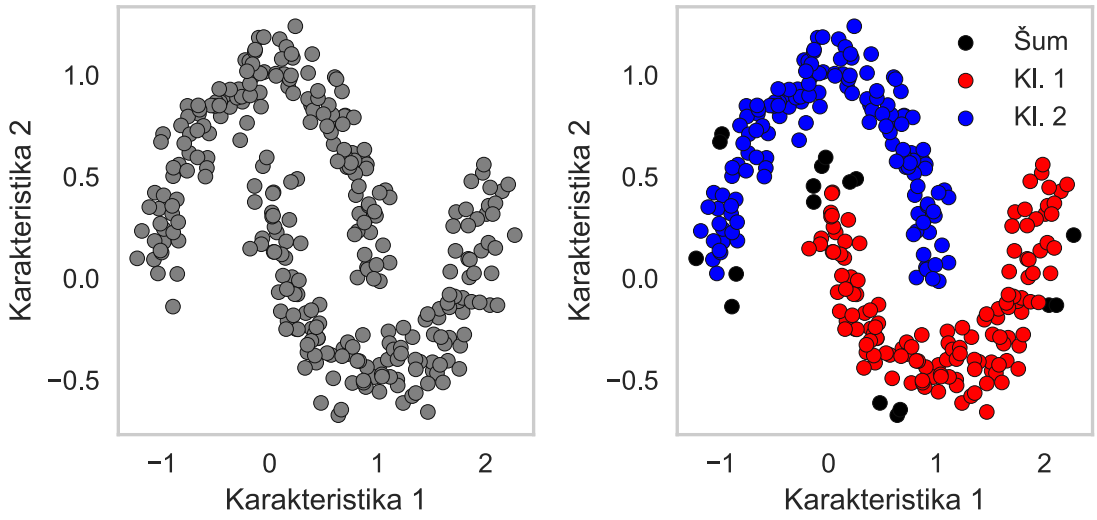
Ulaz: Skup podataka $\mathbb{D}$, radijus pretrage ε, minimalan broj tačaka u klasteru N_{min}
Izlaz: Skup klastera identifikovanih u podacima
for svaku tačku **p** u skupu podataka $\mathbb{D}$ **do**
    if p nije posjećena **then**
        Označi **p** kao posjećenu
        Pronađi sve susjedne tačke unutar radijusa ε
        if broj susjednih tačaka $\geq N_{min}$ **then**
            Kreiraj novi klaster i dodaj **p** u njega
            Rekurzivno proširi klaster dodavanjem susjednih tačaka koje ispunjavaju uslov gustine
        else
            Oznaci **p** kao šum (outlier)
        end if
    end if
end for
return skup pronađenih klastera



Slika 12: Ilustracija tačaka u DBSCAN-u

Slika 12 prikazuje tačke u okviru DBSCAN algoritma. Plavi kvadrat označava *Core Point* oko kojeg je formiran krug (kružna isprekidana linija). Crni puni krugovi označavaju *Border Point*-e, dok bijeli krug označava šum. U ovom primjeru, definisano je da je hiperparametar $N_{min} = 4$.

U nastavku će, za standardni testni skup podataka, biti prikazano kako DBSCAN vrši klasterizaciju.



Slika 13: Testni skup podataka prije i posle DBSCAN klasterizacije. Lijevi dijagram – originalni skup podataka. Desni dijagram – rezultujući skup podataka.

Na slici 13 predstavljen je primjer DBSCAN klasterizacije. Lijevi dijagram prikazuje vizuelizaciju originalnog skupa podataka, gdje imamo dvije odvojene grupe tačaka u obliku luka kao i nekoliko tačaka koje odstupaju od glavnih grupa. Desni dijagram prikazuje skup podataka nakon klasterizacije DBSCAN algoritmom. Crvenom bojom je označen prvi klaster, dok je plavom bojom označen drugi klaster. Crnom bojom je označen šum, tj. tačke koje nisu dodijeljene nijednom klasteru. U praksi, ovakve tačke se dodjeljuju klasteru koji je označen sa -1. Ovaj primjer potvrđuje da DBSCAN algoritam može detektovati klastere različitih oblika i izolovati šumove.

2.4.3.1 Prednosti i nedostaci DBSCAN algoritma

DBSCAN algoritam ima nekoliko prednosti koje ga čine pogodnim za specifične zadatke klasterizacije. Jedna od glavnih prednosti je njegova sposobnost da identifikuje klastere proizvoljnog oblika, što ga čini posebno pogodnim za skupove podataka sa nepravilno raspoređenim grupama podataka. Takođe, algoritam ne zahtijeva unaprijed definisan broj klastera, već samo hiperparametre ε i N_{min} za određivanje gustine. Još jedna značajna prednost je što DBSCAN može da izoluje šum u podacima. Ovo ga čini korisnim u situacijama kada je prisutna velika količina nepovezanih ili nevažnih podataka.

Međutim, DBSCAN ima i nedostatke. Efikasnost i tačnost ovog algoritma u velikoj mjeri zavise od hiperparametara ε i N_{min}. Ova dva hiperparametra se često moraju eksperimentalno prilagoditi. Algoritam može imati problema sa podacima različite gustine klastera, jer se tačke u rijetkim djelovima klastera mogu smatrati šumovima. Takođe, algoritam postaje računski zahtjevan kod skupova podataka sa visokom dimenzionalnošću, gdje je pretraživanje susjeda kompleksnije.

2.4.4 Spektralna klasterizacija

Tradicionalne metode, poput K-means algoritma, imaju ograničenje u pogledu oblika klastera koje mogu identifikovati. Ove metode obično rezultiraju klasterima sa jednostavnim konveksnim geometrijskim oblicima. Drugim riječima, svaka tačka unutar jednog klastera može biti povezana sa bilo kojom drugom tačkom unutar istog klastera, bez izlaska izvan granica klastera, što znači da su klasteri strogo definisani unutar jasno ograničenih prostora. Međutim, spektralna klasterizacija (eng. *Spectral Clustering*) [11] nudi mnogo fleksibilniji pristup. Ova metoda klasterizacije sposobna je da rješava složenije scenarije, uključujući one u kojima klasteri imaju nelinearne oblike, poput isprepletanih spirala ili drugih složenih struktura. Njena snaga leži u tome što ne postavlja pretpostavke o obliku klastera, čime omogućava identifikaciju skrivenih obrazaca u podacima koji bi tradicionalnim metodama bili nedostupni.

Ideja spektralne klasterizacije je da se koriste sopstveni vektori matrice sličnosti kako bi se odredile osnovne podjele podataka [16]. Postoje dvije varijacije algoritma za spektralnu klasterizaciju: jedna koja koristi normalizovani i druga koja koristi nenormalizovani Laplasijan grafa, što će biti objašnjeno u nastavku. Ovi algoritmi su uspješno primijenjeni u različitim oblastima poput segmentacije slika, analize teksta, obrade govora i opšte analize podataka.

Spektralna klasterizacija se može posmatrati kao algoritam sa tri koraka. Prvi korak je formiranje grafa sličnosti između tačaka podataka. Zatim se tačke podataka smještaju u prostor gdje su klasteri lakše uočljivi, koristeći sopstvene vektore Laplasijana grafa. I na kraju, koristi se klasični algoritam klasterizacije (kao što je K-means) kako bi se podijelili podaci.

2.4.4.1 Graf sličnosti

Neka skup podataka koji želimo da podijelimo u K grupa bude označen sa $\mathbb{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Da bismo izvršili spektralnu klasterizaciju, prvo moramo predstaviti ove podatke u formi neusmjerenog grafa sličnosti $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Svaka tačka podataka $\mathbf{x}_i$ predstavljena je kao čvor $\mathbf{v}_i$, a $\mathcal{E}$ predstavlja ivice (veze) između čvorova. Zatim možemo koristiti matricu susjedstva $\mathbf{A}$ dimenzija $n \times n$ da opišemo graf $\mathcal{G}$, gdje je $\mathbf{A}_{ij} = \{w_{ij}\}$, za $i, j = 1, 2, \ldots, n$, dok je w_{ij} težina ivice između dva čvora. Treba napomenuti da je $w_{ij} = 0$ kada čvorovi $\mathbf{v}_i$ i $\mathbf{v}_j$ nisu povezani. Pošto algoritam za spektralnu klasterizaciju ima cilj da podijeli čvorove tako da oni unutar istog klastera imaju visoku sličnost, a oni u različitim klasterima nisku sličnost, postaje ključno odabrati efikasan metod za konstrukciju takve matrice sličnosti. Postoje tri načina za konstrukciju matrice [17]:

1. Grafovi K-najbližih susjeda (eng. *K-nearest neighbors graphs*) – ideja je da su čvorovi $\mathbf{v}_i$ i $\mathbf{v}_j$ povezani ako je $\mathbf{v}_j$ među K-najbližim susjedima čvora $\mathbf{v}_i$, ili obratno. Udaljenost se računa na osnovu originalne reprezentacije tačaka podataka $\mathbf{x}_i$ i $\mathbf{x}_j$. Neki

primjeri udaljenosti uključuju Euklidovu, Manhattan i kosinusnu (eng. *cosine*) udaljenost. Dobijeni graf se obično naziva graf K-najbližih susjeda. Alternativa je da se $\mathbf{v}_i$ i $\mathbf{v}_j$ povežu kada su međusobno u njihovim susjedstvima. Ovaj graf se naziva graf međusobnih K-najbližih susjeda (eng. *mutual K-nearest neighbors graph*) ili simetrični graf K-najbližih susjeda (eng. *symmetric K-nearest neighbors graph*). U oba slučaja, nakon dodavanja ivica prema susjedstvu svakog čvora, možemo dodijeliti težine ivicama na osnovu sličnosti njihovih krajeva ili jednostavno koristiti težinu 0 kada postoji, i 1 kada ne postoji ivica između dva čvora.

2. ε-susjedski graf – u ovom grafu čvorovi su povezani samo kada je razdaljina između tačaka $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ manja od ε. Međutim, ova metoda često dovodi do grafova sa nespojenim komponentama ako ε nije pažljivo odabrano.

3. Potpuno povezani graf – ovdje jednostavno povezujemo sve tačke koje imaju pozitivnu sličnost, i težine ivica postavljamo na vrijednosti s_{ij}. Kako graf treba da predstavlja lokalne odnose susjedstva, ova konstrukcija je korisna samo ako funkcija sličnosti sama po sebi modeluje lokalna susjedstva. Primjer takve funkcije sličnosti je Gausova funkcija sličnosti.

$$s(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}),$$

gdje parametar σ kontroliše širinu susjedstva. Ovaj parametar ima sličnu ulogu kao hiperparamtar ε u slučaju ε-susjedskog grafa.

Izbor specifične metode za konstrukciju grafa sličnosti ponekad može biti složen i izazovan problem.

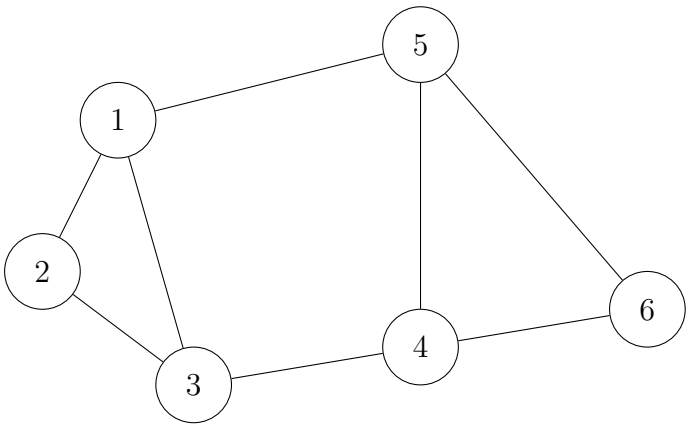
2.4.4.2 Nenormalizovana spektralna klasterizacija

Kada imamo graf sličnosti $\mathcal{G}$, glavni korak za spektralnu klasterizaciju je izračunavanje Laplasijan matrice grafa. U slučaju nenormalizovane spektralne klasterizacije, Laplasijan matrica grafa računa se kao:

$$\mathbf{L} = \mathbf{D} - \mathbf{A},$$

gdje je: $\mathbf{D}$ dijagonalna matrica (matrica stepeni), dok je $\mathbf{A}$ matrica susjedstva.

Prethodno smo, za matricu susjedstva $\mathbf{A}$, definisali nenegativnu težinu w_{ij} za svaki par čvorova $\mathbf{v}_i$ i $\mathbf{v}_j$ u neorijentisanom grafu sličnosti. Pošto je $\mathcal{G}$ neorijentisan graf, imamo da je $w_{ij} = w_{ji}$.



Slika 14: Neorijentisani graf sa 6 čvorova

Slika 14 prikazuje jednostavan neorijentisani graf sa 6 čvorova. Matrica susjedstva, za ovaj graf, je kvadratna matrica 6×6, gdje je svaki element matrice težina ivice između dva čvora. Za težine između čvorova uzeto je da je $w_{ij} = 0$ ako čvorovi nisu povezani i kada ne postoji petlja, odnosno kada se iz jednog čvora ne može stići u isti čvor u jednom koraku. S druge strane, $w_{ij} = 1$ kada postoji veza između dva čvora. Prvi red i prva kolona u matrici odnose se na čvor 1, drugi red i druga kolona na čvor 2, i tako dalje za svih šest čvorova. Matrica susjedstva za čvor sa slike 14 je sledeća:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Dijagonalna matrica $\mathbf{D}$ sadrži na glavnoj dijagonali odgovarajuće stepene čvorova grafa. Za svaki čvor, njegov stepen se računa kao suma svih težina ivica koje povezuju taj čvor sa drugim čvorovima u grafu. Matematički, elementi na glavnoj dijagonali matrice $\mathbf{D}$ se računaju na sledeći način:

$$d_{ij} = \sum_{j=1}^{n} w_{ij}, i = 1, 2, \ldots, n,$$

$$\mathbf{D} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

Za graf sa slike 14, je data dijagonalna matrica **D**. Kao što je objašnjeno, dijagonalna matrica po glavnoj dijagonali sadrži sumu težina ivica kojima je povezan posmatrani čvor, dok su ostale vrijednosti 0. Na primjer, čvor 1 povezan je sa čvorovima 2, 3 i 5, pa je njegov stepen 3, dok je čvor 2 povezan sa čvorovima 1 i 3, pa je njegov stepen 2 i tako dalje.

Laplasijan matrica grafa računa se tako što se od dijagonalne matrice oduzme matrica susjedstva. U opštem slučaju, vrijednosti ove matrice definišu se na sledeći način:

$$L_{ij} = \begin{cases} d_i, & \text{ako je } i = j \\ -w_{ij}, & \text{ako postoji ivica između } i \text{ i } j \\ 0, & \text{ako ne postoji ivica između } i \text{ i } j \end{cases}$$

$$\mathbf{L} = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}.$$

Matrica **L** predstavlja Laplasijan matricu grafa sa slike 14. Glavna dijagonala ove matrice sadrži stepene čvorova. Negativne vrijednosti u poljima matrice ukazuju na to da između dva čvora postoji ivica, dok nule označavaju da ta dva čvora nisu povezana.

Da bi se dobili sopstveni vektori i sopstvene vrijednosti matrice L, rješava se karakteristična jednačina $\mathbf{L} \cdot \mathbf{v} = \lambda \cdot \mathbf{v}$, gdje je $\mathbf{v}$ sopstveni vektor, a λ odgovarajuća sopstvena vrijednost. Rješavanjem ove jednačine dobijamo skup sopstvenih vrijednosti $\lambda_1, \lambda_2, \ldots, \lambda_n$ i odgovarajuće sopstvene vektore $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$. Sopstvene vrijednosti (eng. *eigenvalues*) su numeričke vrijednosti koje nam govore koliko je „važan“ ili „dominantan“ doprinos koji određeni sopstveni vektor ima u kontekstu grafa. Sopstveni vektori (eng. *eigenvectors*) pokazuju pravce ili obrasce u kojima je struktura grafa najizraženija, a komponente vektora često odražavaju važnost pojedinih čvorova.

Za potrebe spektralne klasterizacije koristi se nekoliko prvih sopstvenih vektora koji odgovaraju najmanjim sopstvenim vrijednostima, jer oni najbolje odražavaju strukturu grafa. Ti sopstveni vektori omogućavaju mapiranje podataka u prostor u kojem su klasteri lakše uočljivi. U praksi, različite numeričke metode se koriste za računanje sopstvenih vektora, posebno kada se radi o velikim skupovima podataka. Kada dobijemo ove sopstvene vektore, oni se mogu koristiti kao nisko-dimenzionalne reprezentacije podataka čime se omogućava primjena standardizovanih algoritama za klasterizaciju, poput K-means algoritma.

Pregled algoritma nenormalizovane spektralne klasterizacije dat je u Pseudokodu 4.

Pseudokod 4 Nenormalizovana spektralna klasterizacija

1: **Ulaz**: Matrica sličnosti $\mathbf{A}$, broj klastera K.
2: Izračunaj dijagonalnu matricu stepeni $\mathbf{D}$ gdje $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$.
3: Izračunaj nenormalizovani Laplasijan: $\mathbf{L} = \mathbf{D} - \mathbf{A}$.
4: Odredi K najmanjih sopstvenih vektora matrice $\mathbf{L}$ i formiraj matricu $\mathbf{Y}$ dimenzija $n \times K$.
5: Primijeni K-means algoritam na redove matrice $\mathbf{Y}$.
6: Dodeli oznake klastera originalnim podacima na osnovu K-means rezultata.
7: **Izlaz**: Oznake klastera za podatke.

2.4.4.3 Normalizovana spektralna klasterizacija

Normalizovana spektralna klasterizacija predstavlja varijantu spektralne klasterizacije koja koristi normalizovane verzije Laplasijan matrice grafa. Cilj normalizacije je prilagoditi analizu grafa kako bi se uzele u obzir varijacije u stepenima čvorova. Ovo je posebno važno za grafove sa čvorovima koji imaju vrlo različite stepene, jer nenormalizovana Laplasijan matrica može naglasiti čvorove visokog stepena, što otežava pravilnu detekciju klastera.

Dvije varijante normalizovane Laplasijan matrice su: simetrična normalizovana Laplasijan matrica ($\mathbf{L}_{\text{sym}}$) i normalizovana Laplasijan matrica za slučajni hod ($\mathbf{L}_{\text{rw}}$). Simetrična normalizovana Laplasijan matrica i normalizovana Laplasijan matrica za slučajni hod su definisane kao:

$$\mathbf{L}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}},$$

$$\mathbf{L}_{\text{rw}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}.$$

Simetrična normalizovana Laplasijan matrica koristi se u situacijama kada simetrija olakšava analizu, posebno pri rješavanju standardnog sopstvenog problema. Ova matrica je pogodna za većinu spektralnih klasterizacija.

Pristup slučajnog hoda (eng. *Random-walk*) u klasterizaciji koristi pristup nasumičnog obilaska grafa, gdje se čvorovi prikazuju kao stanja, a prelazi između njih kao vjerovatnoće kretanja. Ovaj metod omogućava efikasno otkrivanje strukture podataka kroz analizu putanja definisanih vjerovatnoćama, što vodi ka stabilnijem i preciznijem grupisanju elemenata.

U radu [18], predložen je unaprijeđeni algoritam spektralne klasterizacije koji koristi *random walk* za bolju ekstrakciju grupa u grafu. Algoritam poboljšava konvergenciju i robusnost, balansirajući lokalne i globalne informacije o strukturi podataka, čime postiže kvalitetnije rezultate u odnosu na klasične metode spektralne klasterizacije.

Pregled algoritma normalizovane spektralne klasterizacije dat je u Pseudokodu 5.

Pseudokod 5 Normalizovana spektralna klasterizacija

1: **Ulaz**: Matrica sličnosti $\mathbf{A}$, broj klastera $\mathbf{K}$.
2: Izračunati dijagonalnu matricu stepeni $\mathbf{D}$ gdje $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$.
3: Izračunati jednu od normalizovanih Laplasijan matrica:
- Simetrična: $\mathbf{L}_{\text{sym}} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{A})\mathbf{D}^{-1/2}$.
- Random-walk: $\mathbf{L}_{\text{rw}} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A})$.

4: Odrediti K najmanjih sopstvenih vektora matrice $\mathbf{L}_{\text{sym}}$ ili $\mathbf{L}_{\text{rw}}$ i formirati matricu $\mathbf{Y}$ dimenzija $n \times K$.
5: Normalizovati redove matrice $\mathbf{Y}$ tako da svaki red bude jedinični vektor.
6: Primijeniti K-means algoritam na redove matrice $\mathbf{Y}$.
7: Dodijeliti oznake klastera originalnim podacima na osnovu K-means rezultata.
8: **Izlaz**: Oznake klastera za podatke.

2.4.4.4 Prednosti i nedostaci spektralne klasterizacije

Spektralna klasterizacija ima brojne prednosti koje je čine pogodnom za rješavanje složenih problema. Omogućava otkrivanje nelinearnih i složenih struktura u podacima, što je posebno korisno kada klasteri nisu sferično raspoređeni. Takođe, pruža fleksibilnost u definisanju grafova sličnosti, omogućavajući prilagođavanje različitim aplikacijama i vrstama podataka. Osim toga, spektralna klasterizacija je manje osjetljiva na početne vrijednosti i pruža robustnost u prisustvu šuma i izolovanih tačaka. Transformacija podataka u prostor sopstvenih vektora smanjuje dimenzionalnost, omogućava vizuelizaciju i primjenu tradicionalnih algoritama poput K-means-a.

Međutim, postoje i određeni nedostaci koji ograničavaju njenu primjenu. Glavni izazov je visoka računska složenost, jer izračunavanje sopstvenih vektora postaje problematično za velike grafove. Rezultati zavise od izbora hiperparametara, poput broja klastera i definicije sličnosti, što može otežati postizanje optimalnih performansi. Spektralna klasterizacija takođe zahtijeva da broj klastera bude unaprijed poznat i može biti osjetljiva na loše definisane grafove sličnosti. Iako je efikasna za manje i srednje skupove podataka, skaliranje na velike grafove ostaje izazov. Osim toga, interpretacija sopstvenih vektora i njihovih sopstvenih vrijednosti često nije intuitivna.

Važno je napomenuti da teorijska analiza pokazuje razlike u pouzdanosti između normalizovane i nenormalizovane varijante spektralne klasterizacije. Normalizovana varijanta pokazuje konzistentnost u veoma opštim uslovima. Nasuprot tome, nenormalizovana spektralna klasterizacija je konzistentna samo pod specifičnim, strogo definisanim uslovima koji često nisu ispunjeni u stvarnim primjerima. Zbog toga normalizovana spektralna klasterizacija bi trebalo da bude preferirana u praksi [19].

2.5 Metode za odabir optimalnih vrijednosti hiperparametara

U klasterizaciji, odabir optimalnih vrijednosti hiperparametara ključan je korak za postizanje kvalitetnih i interpretabilnih rezultata. Različiti algoritmi zahtijevaju definisanje hiperparametara koji utiču na kvalitet klasterizacije i performanse algoritama. Zbog toga je često izazovno odabrati odgovarajuću vrijednost. Različite metode mogu pomoći u odabiru ovih vrijednosti kako bi rezultati što bolje odražavali stvarne odnose među podacima.

Algoritmi poput K-means, hijerarhijske i spektralne klasterizacije zahtijevaju određivanje broja klastera. Za ovaj problem često se koriste metode poput metode „lakta“ (eng. *Elbow Method*), *Silhouette* analize i *Davies-Bouldin Index*-a. Ove metode procjenjuju kvalitet klastera na osnovu njihove unutrašnje kohezije i međusobne separacije. U algoritmima zasnovanim na gustini, poput DBSCAN-a, odabir hiperparametara ε (radijus susjedstva) i N_{min} (minimalni broj tačaka u susjedstvu) igra ključnu ulogu u detekciji različitih oblika i izolaciju šuma. Optimalne vrijednosti ovih hiperparametara zavise od strukture podataka i često se određuju eksperimentalnim putem.

2.5.1 Metoda „lakta“, *Silhouette Score* i *Davies-Bouldin Index*

Metoda „lakta“ [20] se koristi za određivanje optimalnog broja klastera posmatranjem promjene rezultata pri različitim vrijednostima broja klastera. Ova metoda funcioniše tako što se za svaku vrijednost broja klastera (K) računa kohezija unutar klastera, koja se obično izražava kao suma kvadrata udaljenosti svake tačke u klasteru i centroida tog klastera (WCSS – *Within-Cluster Sum of Squares*). Formula za izračunavanje WCSS-a je sledeća:

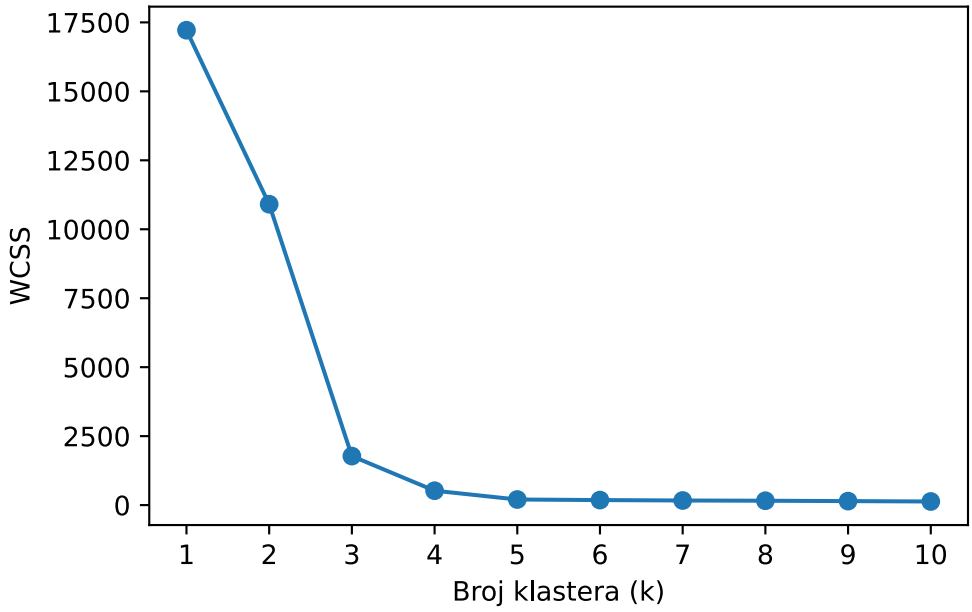
$$WCSS = \sum_{i=1}^{n} \sum_{k=1}^{K} \|\mathbf{x}_i - \mathbf{c}_k\|^2,$$

gdje je:

- $\mathbf{x}_i$ tačka podataka,
- $\mathbf{c}_k$ centroid klastera kojem je tačka pripala,
- $\|\mathbf{x}_i - \mathbf{c}_k\|^2$ kvadratna udaljenost između tačaka $\mathbf{x}_i$ i $\mathbf{c}_k$.

Rezultati se predstavljaju pomoću grafika, gdje apscisna osa sadrži različite brojeve klastera, a ordinatna osa vrijednosti WCSS-a. Na grafiku se formira kriva koja opada kako se broj klastera povećava. Optimalan broj klastera identifikuje se na mjestu gdje se kriva „lomi“ ili formira „lakat“, odakle metoda i dobija naziv. Ova tačka predstavlja trenutak kada

dodavanje novih klastera ne dovodi do značajnog smanjenja WCSS-a, čime se postiže ravnoteža između broja klastera i kvaliteta grupisanja. U nastavku će biti prikazan grafik koji predstavlja rezultat korišćenja metode „lakta“.



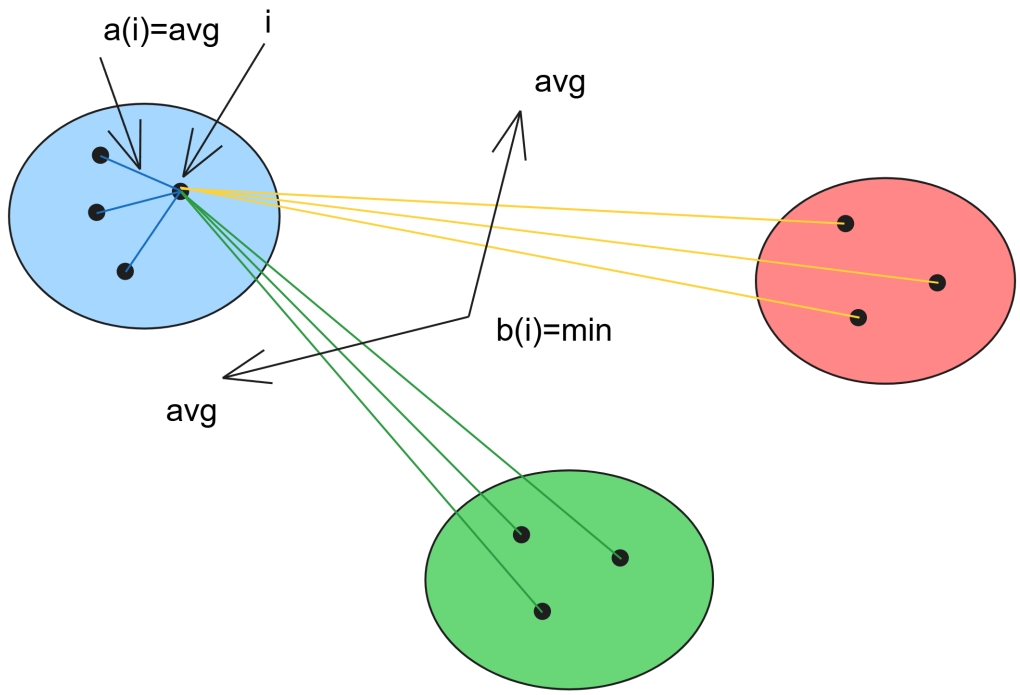
Slika 15: Prikaz rezultata korišćenja metode „lakta“

Na slici 15 prikazan je rezultat korišćenja metode „lakta“. Kao što je prethodno objašnjeno, po x-osi su prikazane vrijednosti broja klastera, dok su na y-osa prikazane vrijednosti WCSS-a. U ovom primjeru, optimalan broj klastera je 3, jer na toj tački pad vrijednosti WCSS-a usporava i postaje stabilniji.

Silhouette metoda [20] se koristi za procjenu kvaliteta klastera, odnosno za procjenu koliko je dobro određeni objekat smješten u klaster. Ova metoda je kombinacija kohezije i separacije. Kohezija mjeri koliko su objekti unutar istog klastera slični, dok separacija mjeri koliko su klasteri međusobno udaljeni. Faze proračuna *Silhouette* koeficijenta su:

- Računanje prosječne udaljenosti između tačke i i svih drugih tačaka u istom klasteru $(a(i))$,
- Računanje prosječne udaljenosti između tačke i i svih drugih tačaka u najbližem susjednom klasteru pri čemu se uzima najmanja vrijednost $(b(i))$,
- *Silhouette* koeficijent se računa pomoću formule:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}.$$



Slika 16: Prikaz računanja *Silhouette* koeficijenta

Slika 16 ilustruje računanje koeficijenta siluete za tačku i koja pripada određenom klasteru. Crnim punim krugovima označene su tačke podataka. Plave linije označavaju prosječnu udaljenost tačke i u odnosu na sve ostale tačke unutar plavog klastera (*intra-cluster cohesion*). Zelene i žute linije označavaju minimalnu prosječnu udaljenost između tačke i i svih tačaka u bilo kojem drugom klasteru (zelenom ili crvenom), a ovo se naziva *inter-cluster cohesion*.

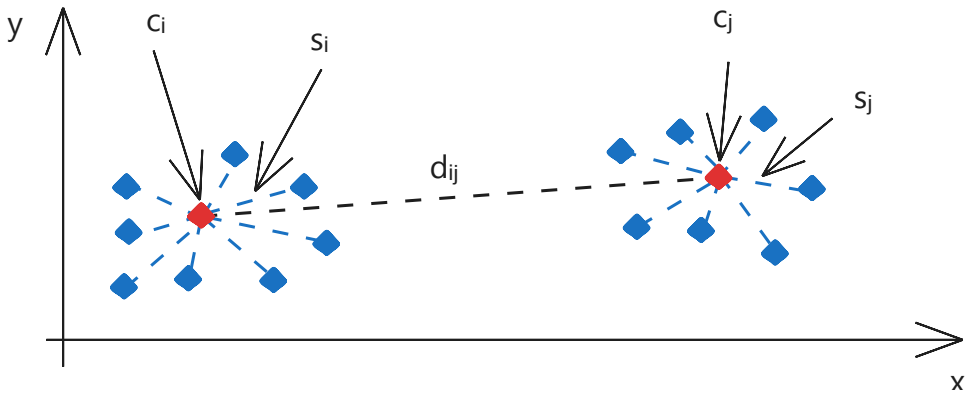
Koeficijent siluete se kreće od -1 do 1. Vrijednost blizu +1 znači da je objekat dobro uklopljen u svoj klaster, sa malim odstupanjima od drugih klastera. Vrijednost 0 označava da se objekat nalazi na granici između dva klastera. Negativne vrijednosti ukazuju na to da objekat vjerovatno nije dodijeljen pravom klasteru. Optimalan broj klastera se dobija eksperimentalno, tj. uzima se onaj broj za koji se dobije najveći koeficijent siluete.

Davies-Bouldin Index (DBI) [21] predstavlja jedan od najpoznatijih internih pokazatelja kvaliteta klasterizacije. Razvijen je sa ciljem da kvantifikuje koliko su formirani klasteri kompaktni i međusobno dobro razdvojeni. Niže vrijednosti *Davies-Bouldin Index*-a ukazuju na bolju klasterizaciju, jer sugerišu da su elementi unutar svakog klastera slični jedni drugima, dok su različiti od elemenata u drugim klasterima. Računanje indeksa podrazumijeva više koraka: najprije se za svaki klaster izračunava prosječna udaljenost tačaka unutar njega od njegovog centroida, a zatim se za svaki par klastera računa rastojanje između njihovih centroida. Na osnovu toga se računa sličnost između svakog klastera i svih ostalih, a zatim se za svaki klaster uzima maksimalna vrijednost te sličnosti. Konačni DBI predstavlja srednju vrijednost tih maksimalnih sličnosti za sve klastere. Matematički, *Davies-Bouldin Index* je

definisan kao:

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{S_i + S_j}{d_{ij}}\right),$$

gdje je S_i prosječna udaljenost unutar klastera i, a d_{ij} udaljenost između centroida klastera i i j.



Slika 17: Ilustracija *Davies-Bouldin Index*-a

Slika 17 ilustruje *Davies-Bouldin Index*. Na dvodimenzionalnom grafiku predstavljena su dva različita klastera podataka. Tačke podataka predstavljene su plavim rombovima, dok su centroidi klastera predstavljeni crvenim rombovima. Za dva klastera, sa centroidima c_i i c_j, prikazane su sledeće komponente za razumijevanje *Davies-Bouldin Index*-a:

1. Intra-klasterska disperzija (s_i i s_j) – isprekidane plave linije povezuju centroid svakog klastera sa ostalim tačkama unutar tog klastera. Linije označene kao s_i i s_j simbolizuju mjeru prosješne udaljenosti tačaka unutar klastera od njihovog centra. Kraće linije ukazuju na kompaktniji klaster.

2. Inter-klasterska udaljenost (d_{ij}) – crna isprekidana linija označava udaljenost između centroida dva različita klastera (c_i i c_j). Ova udaljenost je označena kao d_{ij}. Veća vrijednost d_{ij} ukazuje na bolje razdvajanje između klastera.

Vrijednosti *Davies-Bouldin Index*-a se kreću od 0 do $+\infty$, pri čemu niže vrijednosti označavaju bolju klasterizaciju, dok veće vrijednosti označavaju lošiju klasterizaciju. Imajući u vidu prethodno navedeno, cilj je minimizacija vrijednosti *Davies-Bouldin Index*-a.

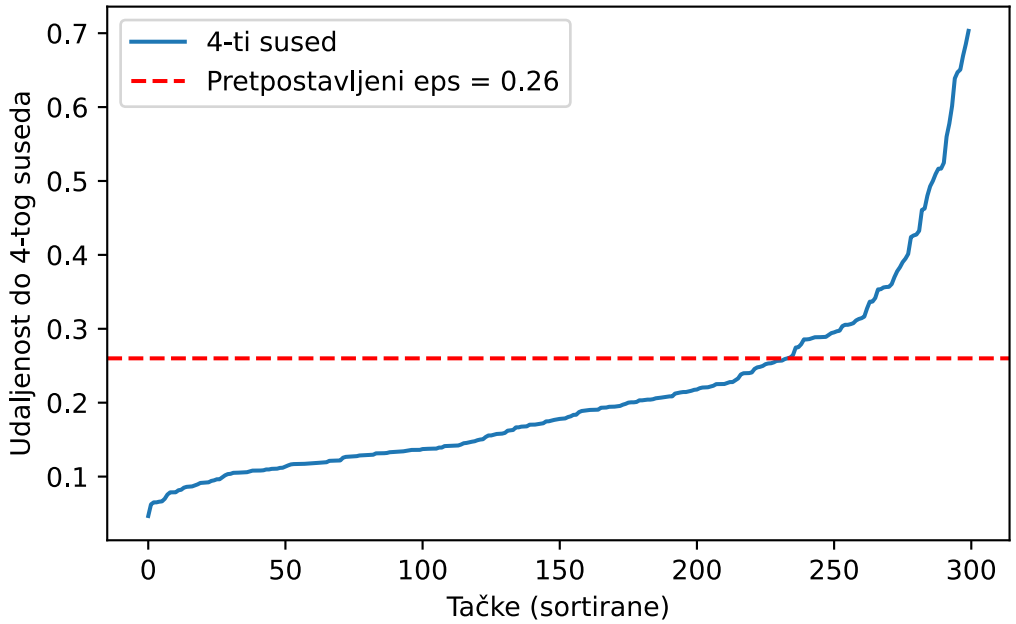
Primjena metode „lakta“, *Silhouette* koeficijenta i *Davies-Bouldin Index*-a nije ograničena samo na particione algoritme, već može pružiti korisne rezultate i kod hijerarhijskih i spektralnih metoda. Kombinacija ovih metoda uz analizu dendrograma će pružiti dobre rezultate za hijerarhijsku klasterizaciju.

Iako metoda „lakta“, *Silhouette Score* i *Davies-Bouldin Index* daju korisne informacije za odabir broja klastera, broj klastera ne treba određivati isključivo na osnovu ovih metoda.

Važno je pronaći kompromisno rješenje, koje će dati zadovoljavajuće rezultate u metričkim evaluacijama, ali istovremeno i omogućiti smislenu i praktično primjenljivu analizu podataka u kontekstu konkretnog domena.

2.5.2 Hiperparametri ε i N_{min} kod DBSCAN algoritma

Hiperparametar ε kod DBSCAN algoritma predstavlja maksimalnu udaljenost između tačaka koje se smatraju susjedima unutar istog klastera. Odabir optimalne vrijednosti ε hiperparametra [15] je obavezan kod ovog algoritma i važan jer od njega zavisi uspješna klasterizacija podataka. Za odabir vrijednosti ovog hiperparametra najčešće se koristi funkcija *k-dist*, koja za svaku tačku u skupu podataka računa udaljenost do njene k-te najbliže susjedne tačke. Sortiranjem ovih tačaka prema njihovim *k-dist* vrijednostima u opadajućem redosledu dobija se graf koji predstavlja raspodjelu gustine tačaka. Na osnovu ovog grafa može se odrediti optimalna vrijednost hiperparametra ε. Na grafu se obično vidi „dolina“, koja označava prelaz između gustih i rijetkih područja. Vrijednost *k-dist* ove tačke postavlja se kao ε, čime se definiše maksimalna udaljenost unutar koje se tačke smatraju susjedima.



Slika 18: Primjer grafika k-distanci

Slika 18 prikazuje primjer grafika k-distanci. Po apscisnoj osi prikazuje se indeks tačaka u skupu podataka, dok su na ordinatnoj prikazane udaljenosti. Grafika prikazuje kako se udaljenosti mijenjaju povećanjem indeksa tačaka u skupu. Isprekidana crvena linija označava vrijednost ε hiperparametra, gdje dolazi do oštrog skoka udaljenosti. Ovaj skok signalizira prelaz između gustih i rijetkih oblasti u skupu podataka.

N_{min} je hiperparametar koji određuje minimalni broj tačaka potrebnih u okolini da bi se tačka smatrala *Core Point*-om. Optimalna vrijednost N_{min} obično zavisi od broja dimenzija

podataka i gustine skupa. Preporučuje se da N_{min} bude bar $D+1$, gdje je D broj dimenzija podataka. Pored ovog heurističkog pravila, N_{min} se često određuje eksperimentalno, testiranjem različitih vrijednosti u kombinaciji sa hiperparametrom ε, kako bi se postigla optimalna detekcija klastera. Dakle, N_{min} se obično bira u rasponu od 4 do 5, iako tačna vrijednost može zavisiti od specifičnosti podataka.

3 Pretprocesiranje i redukcija dimenzionalnosti

Ova sekcija istraživanja bavi se praktičnim koracima vezanim za fazu pretprocesiranja, sa ciljem da se obezbijedi struktuisan i detaljan prikaz pripreme podataka prije same primjene algoritama. Ovaj pristup doprinosi povećanju pouzdanosti i interpretabilnosti rezultata klasterizacije.

Sekcija započinje opisom i porijeklom podataka koji se koriste u istraživanju. Zatim su opisani ključni koraci pripreme podataka, što uključuje inženjering karakteristika, rješavanje problema nedostajućih vrijednosti i izolovanih tačaka (eng. *outliers*), kodiranje kategorijskih promjenljivih i skaliranje podataka. Posebna pažnja posvećena je redukciji dimenzionalnosti, tehnici koja omogućava smanjenje složenosti skupa podataka bez gubitka ključnih karakteristika. Ova metoda je važna ne samo za pojednostavljivanje analize podataka, već i za vizuelni prikaz rezultata klasterizacije.

Za sprovođenje praktičnog dijela ovog istraživanja korišćen je programski jezik Python. Python je odabran zbog bogate kolekcije biblioteka za mašinsko učenje, analizu podataka i klasterizaciju. Ključne biblioteke uključuju: `Pandas`, `NumPy`, `Scikit-learn` (`sklearn`), `Matplotlib` i `Seaborn`.

3.1 Opis i izvor podataka

Prvi skup podataka koji se koristi u ovom istraživanju naziva se „Customer Personality Analysis“. Ovo je javni i vrlo popularan skup podataka za analizu podataka. Preuzet je sa sajta Kaggle [1], koji je jedan od najpopularnijih sajtova za nauku o podacima (eng. *Data Science*) i mašinsko učenje.

„Customer personality analysis“ [22] skup podataka sadrži informacije o kupcima, proizvodima i promocijama. Analizom ovog skupa podataka može se doći do razumijevanja ponašanja korisnika, njihovih preferencija i odgovora na marketinške akcije. Takođe, može se pomoći preduzećima u personalizaciji marketinških strategija, optimizaciji ponude proizvoda, unapređenju zadovoljstva korisnika i povećanju prodaje i profitabilnosti. U nastavku su prikazane karakteristike ovog skupa podataka i njihov opis:

1. `ID` – jedinstveni indentifikator korisnika;

[1] https://www.kaggle.com/

2. `Year_Birth` – godina rođenja;
3. `Education` – nivo obrazovanja;
4. `Marital_Status` – bračni status;
5. `Income` – ukupna primanja domaćinstva na godišnjem nivou;
6. `Kidhome` – broj djece u domaćinstvu;
7. `Teenhome` – broj tinejdžera u domaćinstvu;
8. `Dt_Customer` – datum kada je korisnik registrovan u bazu podataka;
9. `Recency` – broj dana od poslednje kupovine korisnika;
10. `Complain` – 1 ako je korisnik imao žalbu u poslednje 2 godine, 0 u suprotnom;
11. `MntWines` – iznos potrošen na vino u poslednje 2 godine;
12. `MntFruits` – iznos potrošen na voće u poslednje 2 godine;
13. `MntMeatProducts` – iznos potrošen na meso u poslednje 2 godine;
14. `MntFishProducts` – iznos potrošen na ribu u poslednje 2 godine;
15. `MntSweetProducts` – iznos potrošen na slatkiše u poslednje 2 godine;
16. `MntGoldProds` – iznos potrošen na zlato u poslednje 2 godine;
17. `NumDealsPurchases` – Broj izvršenih kupovina sa popustom;
18. `AcceptedCmp1` – 1 ako je korisnik prihvatio ponudu u 1. kampanji, 0 u suprotnom;
19. `AcceptedCmp2` – 1 ako je korisnik prihvatio ponudu u 2. kampanji, 0 u suprotnom;
20. `AcceptedCmp3` – 1 ako je korisnik prihvatio ponudu u 3. kampanji, 0 u suprotnom;
21. `AcceptedCmp4` – 1 ako je korisnik prihvatio ponudu u 4. kampanji, 0 u suprotnom;
22. `AcceptedCmp5` – 1 ako je korisnik prihvatio ponudu u 5. kampanji, 0 u suprotnom;
23. `Response` – 1 ako je korisnik prihvatio ponudu u poslednjoj kampanji, 0 u suprotnom.

U tabeli 5 je prikazano prvih pet redova skupa podataka nakon učitavanja.

Tabela 5: Prikaz „Customer Personality Analysis“ skupa podataka nakon učitavanja

	ID	Year_Birth	Education	Marital_Status	Income	...	Response
0	5524	1957	Graduation	Single	58138.0	...	1
1	2174	1954	Graduation	Single	46344.0	...	0
2	4141	1965	Graduation	Together	71613.0	...	0
3	6182	1984	Graduation	Together	26646.0	...	0
4	5324	1981	PhD	Married	58293.0	...	0

Drugi skup podataka koji se koristi u istraživanju je „Online Retail“ skup podataka [23]. Ovaj popularni skup podataka preuzet je sa UCI Machine Learning Repository [2] sajta, koji je jedna od najpoznatijih *online* resursa za istraživače u oblasti mašinskog učenja i analize podataka.

„Online Retail“ skup podataka sadrži transakcije elektronske trgovine jedne kompanije sa sjedištem u Velikoj Britaniji. Podaci obuhvataju period između decembra 2010. i decembra 2011. godine, uključujući kupovine koje su obavili klijenti iz različitih zemalja. Ovaj skup podataka se često koristi za segmentaciju podataka, predikciju ponašanja kupaca i analizu prodaje, kao i za razvoj modela preporuka u oblasti e-trgovine. Opis karakteristika ovog skupa podataka je sledeći:

1. `InvoiceNo` – jedinstveni broj fakture za svaku transakciju,
2. `StockCode` – jedinstveni broj proizvoda,
3. `Description` – opis proizvoda,
4. `Quantity` – količina kupljenog proizvoda,
5. `InvoiceDate` – datum i vrijeme fakture,
6. `UnitPrice` – cijena jedne jedinice proizvoda,
7. `CustomerID` – identifikacioni broj korisnika,
8. `Country` – zemlja u kojoj korisnik živi.

U tabeli 6 prikazano je prvih pet redova skupa podataka nakon učitavanja.

[2] https://archive.ics.uci.edu/

Tabela 6: Prikaz „Online Retail“ skupa podataka nakon učitavanja

	InvoiceNo	StockCode	...	CustomerID	Country
0	536365	85123A	...	17850.0	United Kingdom
1	536365	71053	...	17850.0	United Kingdom
2	536365	84006B	...	17850.0	United Kingdom
3	536365	84029G	...	17850.0	United Kingdom
4	536365	84029E	...	17850.0	United Kingdom

3.2 Pretprocesiranje podataka

Pretprocesiranje podataka [24] je važan korak u mašinskom učenju, gdje se sirovi skup podataka transformiše u korisniji i format koji je pogodniji za primjenu algoritama za klasterizaciju. Ovaj proces uključuje čišćenje, transformaciju i organizaciju podataka sa ciljem njihove pripreme za odgovarajući algoritam mašinskog učenja. Pretprocesiranje podataka je neophodno jer sirovi podaci često sadrže nepravilnosti poput nedostajućih vrijednosti, šuma, redundantnih karakteristika, neusklađenih formata ili nebalansiranih distribucija, što može negativno uticati na performanse algoritama. Ukoliko se ovi problemi ne riješe, može doći do smanjenja tačnosti modela, dužeg vremena izvršavanja algoritma i smanjenja efikasnosti.

3.2.1 Inženjering karakteristika

Inženjering karakteristika (eng. *Feature Engineering*) predstavlja proces u kojem se sirovi podaci transformišu tako da budu pogodniji za korišćenje od strane algoritma, čime se povećava tačnost modela. U suštini, inženjering karakteristika omogućava da se iz postojećih karakteristika naprave nove, relevantnije karakteristike koje mogu značajno unaprijediti sposobnost modela da prepozna obrasce [25]. Ovo uključuje kreiranje novih karakteristika, transformaciju postojećih karakteristika i uklanjanje irelevantnih ili redundantnih karakteristika.

Da bi primijenili inženjering karakteristika moramo biti upoznati sa osnovnim informacijama vezanim sa skup podataka. U nastavku je dat rezultat korišćenja `info` metode iz Python-ove biblioteke `Pandas` za „Customer Personality Analysis“ skup podataka.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
#   Column               Non-Null Count  Dtype
---  ------               --------------  -----
0   ID                   2240 non-null   int64
1   Year_Birth           2240 non-null   int64
```

```
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Dt_Customer          2240 non-null   object
 8   Recency              2240 non-null   int64
 9   MntWines             2240 non-null   int64
 10  MntFruits            2240 non-null   int64
 11  MntMeatProducts      2240 non-null   int64
 12  MntFishProducts      2240 non-null   int64
 13  MntSweetProducts     2240 non-null   int64
 14  MntGoldProds         2240 non-null   int64
 15  NumDealsPurchases    2240 non-null   int64
 16  NumWebPurchases      2240 non-null   int64
 17  NumCatalogPurchases  2240 non-null   int64
 18  NumStorePurchases    2240 non-null   int64
 19  NumWebVisitsMonth    2240 non-null   int64
 20  AcceptedCmp3         2240 non-null   int64
 21  AcceptedCmp4         2240 non-null   int64
 22  AcceptedCmp5         2240 non-null   int64
 23  AcceptedCmp1         2240 non-null   int64
 24  AcceptedCmp2         2240 non-null   int64
 25  Complain             2240 non-null   int64
 26  Z_CostContact        2240 non-null   int64
 27  Z_Revenue            2240 non-null   int64
 28  Response             2240 non-null   int64
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

Rezultat korišćenja `info` metode prikazuje nazive svih karakteristika sa tipovima podataka. Neke od ovih karakteristika koristićemo za inženjering karakteristika.

Prva karakteristika koja je interesantna u kontekstu inženjeringa karakteristika je `Year_Birth`. Ova karakteristika prikazuje godinu rođenja korisnika. Radi pojednostavljenja, kreiraćemo novu kolonu *Age* koja će, na osnovu podataka iz `Year_Birth` kolone, sadržati godine starosti korisnika. Karakteristiku `Age` računamo tako što od tekuće godine oduzmemo godinu rođenja korisnika.

Dalje, istražićemo kolonu `Education` i provjeriti da li kod nje ima prostora za pojednostavljenje. Kolona `Education` sadrži podatke o nivou obrazovanja korisnika.

Tabela 7: Prikaz kategorija karakteristike `Education`

Kategorija	Broj redova
Graduation	1116
PhD	481
Master	365
2n Cycle	200
Basic	54

Tabela 7 pokazuje da karateristika `Education` sadrži kategorije: *Graduation*, *PhD*, *Master*, *2n Cycle* i *Basic*. Termini *Basic* i *2n Cycle* koriste za korisnike koji nisu fakultetski obrazovani. Za korisnike sa *Graduation* nivoom obrazovanja smatra se da su završili osnovne studije. I na kraju, kategorije *Master* i *PhD* označavaju da korisnik ima završen određeni stepen postdiplomskih studija. U skladu sa tim, navedene kategorije ćemo smjestiti u tri nove, i to: *Undergraduate* (*Basic* i *2n Cycle*), *Graduate* (*Graduation*) i *Postgraduate* (*Master* i *PhD*).

Sledeća karakteristika je `Marital_Status`, koja pokazuje bračno stanje korisnika. Kategorije ove karakteristike prikazane su u tabeli 8.

Tabela 8: Prikaz kategorija karakteristike `Marital_Status`

Kategorija	Broj redova
Married	857
Together	573
Single	471
Divorced	232
Widow	76
Alone	3
Absurd	2
YOLO	2

Tabela 8 pokazuje da karakteristika ima sledeće kategorije: *Married* (oženjen/udata), *Together* (ima partnera/ku ali nisu u braku), *Single* (nema partnera/ku), *Divorced* (razveden/a), *Widow* (udovica), *Alone* (nema partnera/ku), *Absurd* (nema partnera/ku), *YOLO* (nema partnera/ku). Na osnovu ovih kategorija napravićemo karakteristiku `Living_With` koja će imati dvije grupe: *Partner* (korisnici koji imaju partnera/ku) i *Alone* (korisnici koji nemaju partnera/ku).

Karakteristike `Kidhome` i `Teenhome` prikazuju broj djece i tinejdžera u domaćinstvu. Na osnovu podataka iz ovih promjenljivih možemo kreirati novu karakteristiku `Total_Children`, koja će prikazivati ukupan broj djece u domaćinstvu. Takođe, kreiraćemo i karakteristiku `Is_Parent` koja će prikazivati da li je korisnik roditelj ili ne.

Dodatno, na osnovu karakteristika `Total_Children` i `Living_With` možemo izračunati ukupan broj članova domaćinstva. Novu karakteristiku nazvaćemo `Family_Size` i računaćemo je tako što broj djece (`Total_Children`) saberemo sa odgovarajućim numeričkim ekvivalentima kategorija karakteristike `Living_With` (1 za *Alone*, 2 za *Partner*).

Na ispisu dobijenom korišćenjem `info` metode vidimo da je karakteristika `Dt_Customer` tipa *object*. Ova karakteristika prikazuje datum kada je korisnik registrovan u bazi podataka firme. Tip *object* mora se zamijeniti tipom *datetime*, kako bi podaci bili čuvani u pravilnom formatu. Na osnovu izmijenjene karakteristike, možemo izračunati broj dana od kada je korisnik upisan u bazu podataka firme. Ove vrijednosti računamo u odnosu na najskorije upisanog korisnika. Nova karakteristika zvaće se `Customer_For`.

Sledeće karakteristike koje su interesatne u kontekstu inženjeringa karakteristika su: `MntWines`, `MntFruits`, `MntMeatProducts`, `MntFishProducts`, `MntSweetProducts` i `MntGoldProds`. Pomenute karakteristike prikazuju potrošnju korisnika na odgovarajuću kategoriju proizvoda. Pomoću njih možemo izračunati ukupnu potrošnju korisnika, tj. kreirati karakteristiku `Total_Spent`. Takođe, radi jasnoće možemo skratiti nazive ovih karakteristika, pa će novi nazivi biti: `Wines`, `Fruits`, `Meat`, `Fish`, `Sweets` i `Gold`.

Na samom kraju inženjeringa karakteristika, brišemo redudantne i irelevantne karakteristike, kao što su: `Marital_Status`, `Dt_Customer`, `Z_CostContant`, `Z_Revenue`, `Year_Birth` i `ID`. Izgled podataka nakon inženjeringa karakteristika prikazan je u tabeli 9.

Tabela 9: Prikaz „Customer Personality Analysis“ skupa podataka nakon inženjeringa karakteristika

	Education	Income	Kidhome	Teenhome	...	Age	Total_Spent
0	Graduate	58138.0	0	0	...	64	1617
1	Graduate	46344.0	1	1	...	67	27
2	Graduate	71613.0	0	0	...	56	776
3	Graduate	26646.0	1	0	...	37	53
4	Postgraduate	58293.0	1	0	...	40	422

Sada prelazimo na „Online Retail“ skup podataka. Prije svega, prikazaćemo osnovne informacije ovog skupa podataka.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
#   Column       Non-Null Count   Dtype
---  ------       --------------   -----
0   InvoiceNo    541909 non-null  object
1   StockCode    541909 non-null  object
2   Description  540455 non-null  object
3   Quantity     541909 non-null  int64
4   InvoiceDate  541909 non-null  object
5   UnitPrice    541909 non-null  float64
6   CustomerID   406829 non-null  float64
7   Country      541909 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

Iz ovog skupa podataka nam neće biti potrebne sve karakteristike, već će podaci biti analizirani pomoću RFM (eng. *Recency*, *Frequency*, *Monetary*) analize. Dakle, podaci će biti analizirani na osnovu tri faktora:

1. *Recency* - broj dana od poslednje kupovine korisnika,
2. *Frequency* - broj transakcija korisnika i
3. *Monetary* - ukupan iznos koji je korisnik potrošio.

Prvo, računamo *Monetary* faktor. Kreiraćemo novi skup podataka *rfm_m* i u njemu karakteristiku `Amount`, koja se računa kao proizvod broja jedinica i cijene jedne jedinice proizvoda. Nakon toga, grupisanjem po `CustomerID`-u sumiramo sve iznose koje je korisnik potrošio po transakcijama. Na taj način dobijamo ukupan iznos koji je korisnik potrošio. Izgled novokreiranog skupa podataka prikazan je u tabeli 10.

Tabela 10: Izgled *rfm_m* skupa podataka

	CustomerID	Amount
0	12346.0	0.00
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40

Zatim, računamo *Frequency* faktor. Ovaj faktor računamo tako što za svakog korisnika brojimo fakture koje je on napravio. Takođe, rezultat ovog računanja smještamo u novi skup podataka *rfm_f*. Izgled ovog skupa podataka prikazan je u tabeli 11.

Tabela 11: Izgled *rfm_f* skupa podataka

	CustomerID	Frequency
0	12346.0	2
1	12347.0	182
2	12348.0	31
3	12349.0	73
4	12350.0	17

Sledeće što računamo je faktor *Recency*. Za faktor *Recency* treba nam karakteristika `InvoiceDate` iz originalnog skupa podataka. Ova karakteristika je sačuvana u nepravilnom formatu, pa ćemo tip *object* zamijeniti tipom *datetime*. Dalje, neophodno je izvući „maksimalni“ datum, tj. datum poslednje transakcije (od svih transakcija), kako bismo utvrdili broj dana od poslednje kupovine. Da bismo utvrdili broj dana od poslednje kupovine korisnika, u originalnom skupu podataka kreiraćemo karakteristiku `Diff`, koja će predstavljati razliku između datuma poslednje transakcije za cio skup podataka i pojedinačnih datuma transakcije svakog korisnika. Konačno, kreiramo novi skup podataka *rfm_r* gdje ćemo broj dana od poslednje transakcije korisnika računati tako što za svakog korisnika izvučemo minimalnu vrijednost karakteristike `Diff`. Izgled *rfm_r* skupa podataka prikazan je u tabeli 12.

Tabela 12: Izgled *rfm_r* skupa podataka

	CustomerID	Diff
0	12346.0	325 days 02:33:00
1	12347.0	1 days 20:58:00
2	12348.0	74 days 23:37:00
3	12349.0	18 days 02:59:00
4	12350.0	309 days 20:49:00

Kao što se vidi u tabeli 12, karakteristika `Diff` novog skupa podataka sadrži vrijednosti koje uključuju broj dana i vrijeme od poslednje transakcije korisnika. S obzirom na to da nam je potreban samo broj dana, uklonićemo dio sa vremenom, pa će skup podataka izgledati kao u tabeli 13.

Tabela 13: Konačan izgled *rfm_r* skupa podataka

	CustomerID	Diff
0	12346.0	325
1	12347.0	1
2	12348.0	74
3	12349.0	18
4	12350.0	309

Nakon što smo izračunali faktore *Recency*, *Frequency* i *Monetary*, te kreirali tri skupa podataka za ove faktore, sledeći korak je spajanje ta tri skupa podataka u jedan skup podataka. Ovaj skup podataka ćemo koristiti za dalje operacije pretprocesiranja podataka. Rezultat spajanja skupova podataka je *rfm* skup podataka sa karakteristikama: `CustomerID`, `Amount`, `Frequency` i `Recency`, čiji je izgled prikazan u tabeli 14.

Tabela 14: Izgled *rfm* skupa podataka

	CustomerID	Amount	Frequency	Recency
0	12346.0	0.00	2	325
1	12347.0	4310.00	182	1
2	12348.0	1797.24	31	74
3	12349.0	1757.55	73	18
4	12350.0	334.40	17	309

3.2.2 Rješavanje problema nedostajućih vrijednosti

Mnogi skupovi podataka sadrže nepostojeće vrijednosti [24]. Nepostojeća vrijednost je vrijednost atributa koja nije unesena ili je izgubljena tokom procesa bilježenja. Postoje različiti razlozi za prisustvo nepostojećih vrijednosti, kao što su ručni unos podataka i greške kod automatskih unosa podataka. Glavni problemi koje proizvode nedostajuće vrijednosti su gubitak efikasnosti modela i komplikacije u analizi podataka. Obrada nedostajućih vrijednosti se najčešće sprovodi na dva načina:

- Brisanje redova sa nedostajućim vrijednostima - iako je jednostavan, ovaj metod može dovesti do gubitka značajnog dijela podataka, što smanjuje efikasnost analize. Ovaj način je praktičan samo kada skup podataka sadrži relativno mali broj redova sa nedostajućim vrijednostima.
- Imputacija nedostajućih vrijednosti - obuhvata niz procedura koje imaju za cilj da popune nedostajuće vrijednosti procijenjenim vrijednostima. Po pravilu, karakteristike u skupu podataka nisu nezavisne jedne od drugih. Prepoznavanjem odnosa među karakteristikama moguće je odrediti vrijednosti koje nedostaju. Ovaj pristup omogućava očuvanje veličine skupa podataka i smanjenje gubitka informacija.

Za slučaj „Customer Personality Analysis“ skupa podataka, ranije prikazani ispis dobijen kao rezultat `info` metode pokazuje da karakteristika `Income` umjesto 2240 vrijednosti, sadrži 2216 vrijednosti. Ovo ukazuje na postojanje 24 nedostajuće vrijednosti u ovoj karakteristici. Ovu informaciju možemo dobiti i sledećim kodom: `data.isnull().sum()` koji kao rezultat daje sledeći ispis:

```
ID                   0
```

```
Year_Birth             0
Education              0
Marital_Status         0
Income                24
Kidhome                0
Teenhome               0
Dt_Customer            0
Recency                0
MntWines               0
MntFruits              0
MntMeatProducts        0
MntFishProducts        0
MntSweetProducts       0
MntGoldProds           0
NumDealsPurchases      0
NumWebPurchases        0
NumCatalogPurchases    0
NumStorePurchases      0
NumWebVisitsMonth      0
AcceptedCmp3           0
AcceptedCmp4           0
AcceptedCmp5           0
AcceptedCmp1           0
AcceptedCmp2           0
Complain               0
Z_CostContact          0
Z_Revenue              0
Response               0
dtype: int64
```

Obzirom da skup podataka ne sadrži veliki broj nedostajućih vrijednosti, izvršićemo njihovu obradu na prvi način, tj. uklonićemo redove sa nedostajućim vrijednostima.

Rješavanje problema sa nedostajućim vrijednostima u slučaju „Online Retail“ skupa podataka vrši se prije izračunavanja RFM faktora kako bi se osigurala tačnost i pouzdanost analize. Provjeravamo da li u originalnom skupu podataka postoje redovi sa nedostajućim vrijednostima i brišemo ih.

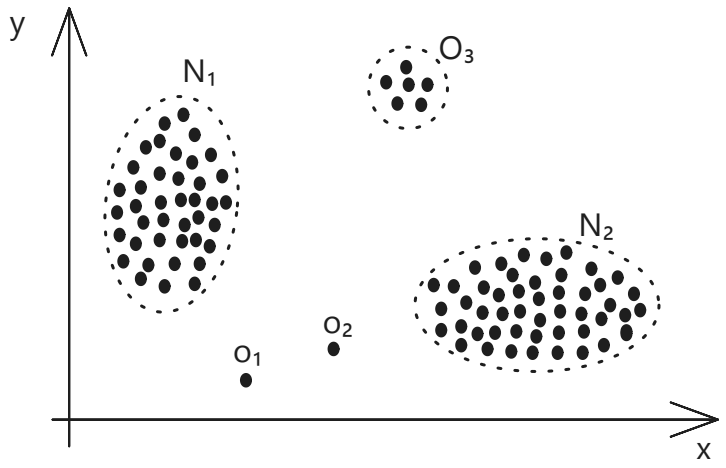
```
InvoiceNo          0
StockCode          0
Description     1454
```

```
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

Očigledno, gornji ispis pokazuje da karakteristika `Description` ima 1454, dok karakteristika `CustomerID` sadrži 135.080 redova sa nedostajućim vrijednostima. Kao i u prethodnom slučaju, i ovdje brišemo redove sa nedostajućim vrijednostima.

3.2.3 Detekcija i rješavanje problema *outlier*-a

Izolovane tačke u podacima (eng. *outliers*) [26] su obrasci koji se ne uklapaju u dobro definisano poimanje normalnog ponašanja. Slika 19 prikazuje primjer outlier-a u jednostavnom dvodimenzionalnom skupu podataka.



Slika 19: Primjer outlier-a

Slika 19 prikazuje jednostavan dvodimenzionalni skup podataka. Podaci imaju dva normalna regiona N_1 i N_2, jer većina posmatranih tačaka leži u tim regionima. Tačke koje su dovoljno udaljene od tih regiona, npr. tačke o_1 i o_2, kao i tačke u regionu O_3, predstavljaju izolovane tačke, odnosno *outlier*-e.

Apstraktno govoreći, *outlier*-i su obrasci koji odstupaju od normalnog ponašanja, koji se u svom najjednostavnijem obliku može predstaviti kao region, pri čemu se sve normalne posmatrane vrijednosti vizuelizuju kao dio tog normalnog regiona, dok se ostatak smatra izolovanim podacima. Iako ovaj pristup djeluje jednostavno, može biti izazovan iz više razloga.

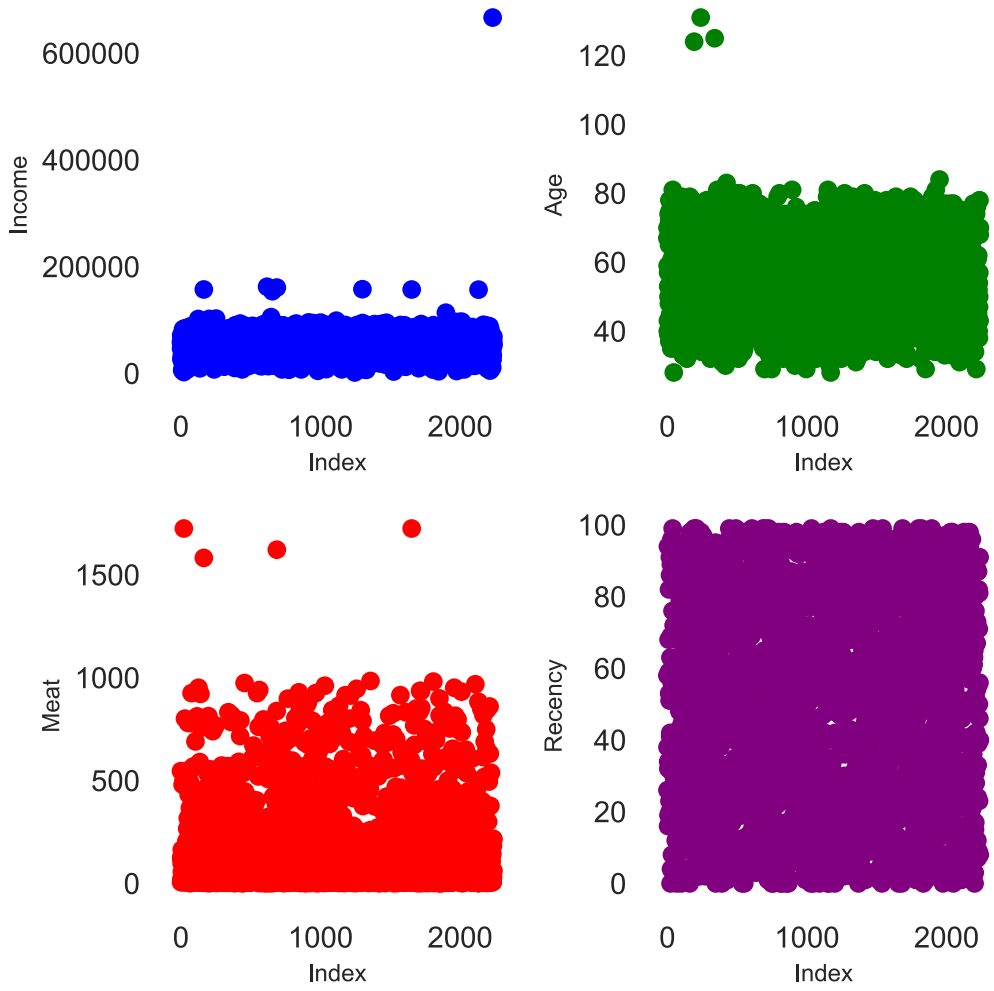
Vrlo je teško definisati normalno ponašanje ili normalni region. Neke od poteškoća su sledeće:

- Obuhvatanje svakog mogućeg normalnog ponašanja unutar regiona.
- Neprecizna granica između normalnog i izolovanog ponašanja, jer ponekad izolovana posmatranja koja se nalaze blizu granice mogu zapravo biti normalna, i obrnuto.
- Različita shvatanja *outlier*-a u različitim aplikativnim domenima čine primjenu tehnike razvijene u jednom domenu na drugi vrlo teškom.

Postoji veliki broj tehnika za detekciju *outlier*-a. Neke od najčešće primijenjenih tehnika detekcije izolovanih tačaka su:

- *Z-Score* – mjerenje broja standardnih devijacija koje su potrebne da bi se posmatrana vrijednosti razlikovala od prosjeka.
- *IQR* (*Interquartile Range*) – detekcija *outlier*-a na osnovu udaljenost od prvog ($Q1$) i trećeg ($Q3$) kvartila. Izolovane tačke se obično definišu kao vrijednosti koje leže izvan opsega od $1,5 \cdot IQR$ ispod $Q1$ ili $1,5 \cdot IQR$ iznad $Q3$.
- *k-Nearest Neighbors* (KNN) – identifikacija izolovanih tačaka na osnovu udaljenosti između tačke i njenih K-najbližih susjeda.

Vizuelizovaćemo određene karakteristike „Customer Personality Analysis“ skupa podataka za koje ćemo utvrditi postojanje izolovanih tačaka. Na primjer, grafički će biti prikazane karakteristike `Income`, `Age`, `Meat` i `Recency` (slika 20).



Slika 20: Grafički prikaz odabranih karakteristika „Customer Personality Analysis“ skupa podataka

Slika 20 pokazuje da karakteristika `Income` ima jednu tačku koja ekstremno odstupa od ostalih vrijednosti. Dok se većina primanja kreću najviše do 100.000, postoji jedna tačka koja pokazuje primanja veća od 600.000. Takođe, karakteristika `Age` sadrži tri tačke koje pokazuju vrijednosti preko 120, što su ekstremno visoke vrijednosti ako uzmemo u obzir da se radi o godinama starosti, dok se ostale vrijednosti kreću od 20 do 80. Karakteristika `Meat` sadrži 4 tačke koje se mogu smatrati potencijalnim *outlier*-ima, dok karakteristika `Recency` najvjerovatnije nema *outlier*-a.

Za potvrdu prisustva i uklanjanje izolovanih tačaka u ovom radu biće korišćena IQR tehnika. Otkrivanje izolovanih tačaka pomoću IQR tehnike zasniva se na razumijevanju statističkog ponašanja podataka unutar opsega definisanog kvartilima. Vrijednost kvartila za odgovarajuće karakteristike skupa podataka može se dobiti korišćenjem metode `describe` biblioteke `Pandas`. Tabela 15 prikazuje opisnu statistiku skupa podataka, koja se dobija kao rezultat korišćenja `describe` metode.

Tabela 15: Opisna statistika „Customer Personality Analysis“ skupa podataka

	Income	Kidhome	...	Family_Size	Is_Parent
count	2216.000000	2216.000000	...	2216.000000	2216.000000
mean	52247.251354	0.441787	...	2.592509	0.714350
std	25173.076661	0.536896	...	0.905722	0.451825
min	1730.000000	0.000000	...	1.000000	0.000000
25%	35303.000000	0.000000	...	2.000000	0.000000
50%	51381.500000	0.000000	...	3.000000	1.000000
75%	68522.000000	1.000000	...	3.000000	1.000000
max	666666.000000	2.000000	...	5.000000	1.000000

Kod IQR, koriste se prvi kvartil ($Q1$ – 25% vrijednosti podataka) i treći kvartil ($Q3$ – 75% vrijednosti podataka). Prvo se računa interkvartilni opseg (IQR), koji predstavlja područje u kojem leži srednjih 50% podataka. Ova vrijednost je razlika trećeg i prvog kvartila, tj. računa se po formuli $IQR = Q3 - Q1$. Zatim se definiše donja i gornja granica opsega i sve tačke koje izlaze iz tog opsega se smatraju izolovanim tačkama. Odgovarajuće formule za donju i gornju granicu su:

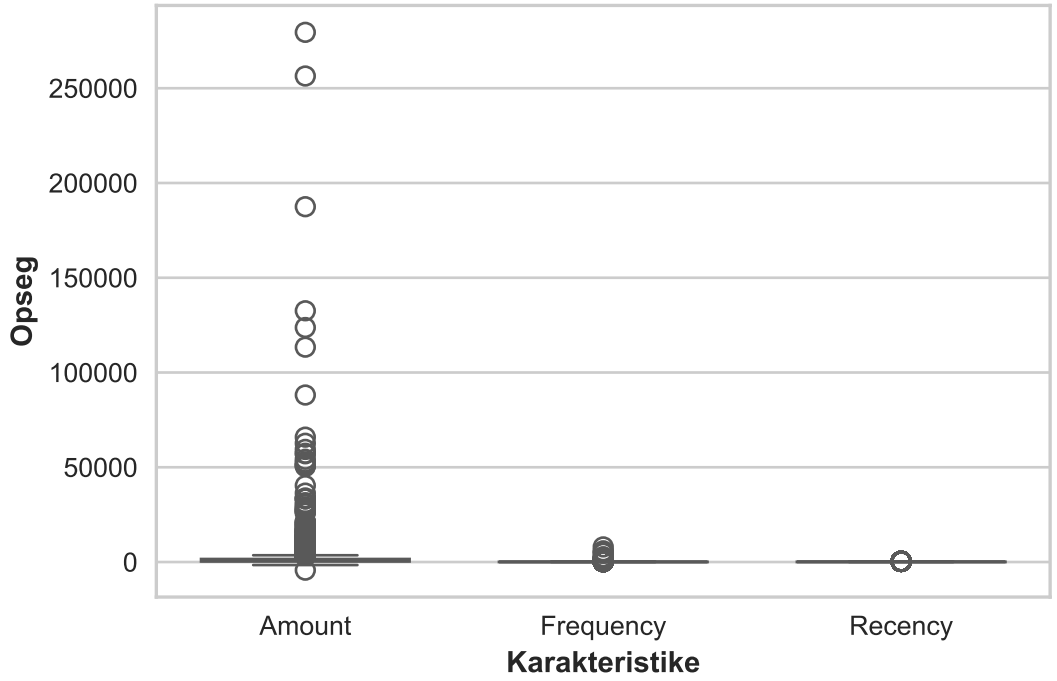
$$\text{Donja granica} = Q1 - k \cdot IQR$$

$$\text{Gornja granica} = Q3 + k \cdot IQR$$

Parametar k definiše koliko daleko od interkvartilnog opsega može biti neka tačka a da se ne smatra *oulier*-om. Standardna vrijednost za ovaj parametar je $k = 1,5$, što omogućava detekciju umjerenih izolovanih tačaka. Za blaži kriterijum koji identifikuje samo ekstremne vrijednosti uzima se da je $k = 3,0$. Ukoliko je $k < 1,5$, tada se povećava osjetljivost na izolovane tačke, pa i manja odstupanja mogu biti označena kao izolovane tačke. U slučaju „Customer Personality Analysis“ skupa podataka uzećemo da je $k = 3,0$, odnosno uklonićemo samo vrijednosti koje ekstremno odstupaju od ostalih.

Pored ovog načina uklanjanja *oulier*-a, mogli smo primijeniti i jednostavniji način. Ovaj način definiše da se brišu sve vrijednosti karakteristike `Age` koje su veće od 90 i sve vrijednosti karakteristike `Income` veće od 600.000. Oba pristupa daju identičan rezultat, tj. nakon brisanja izolovanih tačaka, ukupan broj tačaka iznosi: 2212.

Kod drugog skupa podataka *rfm* koji smo dobili pomoću „Online Retail“ skupa podataka, takođe koristimo IQR tehniku za detekciju i uklanjanje izolovanih tačaka. Vizuelizovaćemo karakteristike `Amount`, `Frequency` i `Recency` kako bismo vizuelno detektovali prisustvo izolovanih tačaka (slika 21). Ovog puta ćemo podatke vizuelizovati pomoću metode `boxplot` biblioteke `Seaborn`.



Slika 21: Grafički prikaz izabranih karakteristika *rfm* skupa podataka

`Boxplot` sa slike 21 jasno ukazuje na izolovane tačke, koje se nalaze iznad ili ispod podebljane horizontalne linije. Karakteristika `Amount` ima veliki broj izolovanih tačaka, koje se protežu na vrijednostima koje su značajno veće u odnosu na ostale (čak i preko 250.000). `Frequency` karakteristika ima manje izolovanih tačaka u poređenju sa `Amount` karakteristikom, ali su prisutne tačke koje značajno odstupaju od većine. Na kraju, `Recency` karakteristika ima vrlo malo izolovanih tačaka.

Kao i kod „Customer Personality Analysis“, izolovane tačke u *rfm* skupu podataka biće detektovane i uklonjenje korišćenjem IQR tehnike.

3.2.4 Kodiranje kategorijskih promjenljivih

Skupovi podataka, pored numeričkih karakteristika, mogu sadržati i nenumeričke, tj. kategorijske karakteristike. Za razliku od numeričkih karakteristika, koje kvantitativno opisuju podatke, kategorijske karakteristike se odnose na kategorije, klase ili oznake koje nemaju numeričku vrijednost. Tipični primjeri kategorijskih promjenljivih su pol („muški“ i „ženski“), boja („crvena“, „plava“ i „zelena“) ili status („aktivan“ i „neaktivan“).

Ovakve karakteristike igraju značajnu ulogu u mnogim primjenama algoritama mašinskog učenja. Međutim, pošto većina algoritama radi sa numeričkim vrijednostima, kategorijske promjenljive predstavljaju izazov u procesu analize podataka. Izostavljanje karakteristika koje ne sadrže numeričke vrijednosti može imati veoma negativan uticaj na rezultate algoritama, odnosno na klasterizaciju. Zbog toga, vrijednosti kategorijskih karakteristika se,

korišćenjem odgovarajućih tehnika, moraju pretvoriti u numeričke. Postoji veliki broj tehnika kodiranja kategorijskih promjenljivih [27], a u nastavku će biti opisane tri najčešće korišćene: *label encoding*, *ordinal encoding* i *one-hot encoding*.

Label encoding [27] tehnika kodiranja kategorijskih podataka podrazumijeva da se svaka jedinstvena tekstualna vrijednost pretvara u cjelobrojnu vrijednost na osnovu redosleda. Naime, ako je skup jedinstvenih vrijednosti u jednom polju sačinjen od n elemenata $\mathbb{A} = \{a_0, a_1, a_2, \ldots, a_{k-1}\}$, tada se zamjena u tekstualnim podacima u ovom polju postiže korišćenjem sledeće formule:

$$LEnc(b_i) = \begin{cases} 0, & b_i = a_0; \\ 1, & b_i = a_1; \\ 2, & b_i = a_2; \\ \vdots & \vdots \\ k-1, & b_i = a_{k-1} \end{cases}$$

gdje i ide od 1 do n.

Na slici 22 prikazan je primjer kodiranja kategorijske karakteristike jednostavnog skupa podataka korišćenjem *label encoding* metode.

Originalni podaci

`Tim`	`Poeni`
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

$\Rightarrow$

Kodirani podaci

`Tim`	`Poeni`
0	25
0	12
1	15
1	14
1	19
1	23
2	25
2	29

Slika 22: Primjer kodiranja *label encoding* tehnikom

Primjer sa slike 22 prikazuje izgled jednostavnog skupa podataka čija je karakteristika `Tim` kodirana `label encoding` tehnikom. Kategorije A, B i C kodirane su brojevima 0, 1 i 2, respektivno. Iako se ova tehnika smatra najjednostavnijom i najrazumljivijom metodom kodiranja kategorijskih promjenljivih, ova metoda se obično koristi kada kategorijski podaci u jednom polju imaju samo dvije jedinstvene vrijednosti. Dakle, ova metoda je efikasnija kada je $k = 2$, jer se u ovoj metodi tekstualne vrijednosti zamjenjuju numeričkim vrijednostima bez obraćanja pažnje na bilo kakav nivo. Ova metoda kodiranja se u Python-u može koristiti pomoću modula `sklearn.preprocessing` i klase `LabelEncoder`.

Metoda *ordinal encoding* [27] slična je prethodnoj metodi. Glavna razlika između njih je u tome što se kod metode *label encoding* zamjena tekstualnih podataka cjelobrojnim vrijednostima vrši prema njihovom redosledu pojavljivanja, dok se kod ordinalnog kodiranja zamjena vrši prema njihovoj semantičkoj vrijednosti. Kategorijski podaci sa većom semantičkom vrijednošću zamjenjuju se većim cjelobrojnim vrijednostima, dok se podaci sa manjom semantičkom vrijednošću zamjenjuju manjim cjelobrojnim vrijednostima.

Tabela 16 prikazuje primjer kodiranja *ordinal encoding* tehnikom.

Tabela 16: Primjer kodiranja *ordinal encoding* tehnikom

Originalni podaci	**Podaci nakon kodiranja**
Slabo	0
Dobro	1
Vrlo dobro	2
Odlično	3

Tabela 16 prikazuje originalne podatke i podatke nakon kodiranja *ordinal encoding* tehnikom. Prvobitna karakteristika sadrži kategorije *Slabo*, *Dobro*, *Vrlo dobro* i *Odlično*. Najmanju semantičku vrijednost ima kategorija *Slabo* i ona je kodirana vrijednošću 0. Kategorija *Odlično* ima najveću semantičku vrijednost, pa je kodirana vrijednošću 3. Kategorije *Dobro* i *Vrlo dobro* su po simantičkoj vrijednosti između *Slabo* i *Odlično* pa su kodirane vrijednostima 1 i 2, respektivno. I ova metoda se u Python-u koristi preko modula `sklearn.preprocessing`, dok je odgovarajuća klasa `OrdinalEncoder`.

Primjena gore pomenutih metoda na nominalne podatke možda neće biti efikasna. Ove metode, kao što je već pomenuto, dodjeljuju numeričke vrijednosti različitih veličina nominalnim vrijednostima istog nivoa, što može dovesti do grešaka. U ovom slučaju, *one-hot encoding* [27] se smatra efikasnom metodom. U ovoj metodi, svaka jedinstvena vrijednost se dodaje u skup podataka kao posebno polje (karakteristika). To znači da, ako postoji k jedinstvenih vrijednosti u jednom tekstualnom polju u skupu podataka, k novih kolona se dodaje u skup podataka. Ćelije (b_{ij}) u tim kolonama se popunjavaju vrijednostima 0 ili 1. Da li će vrijednost biti 0 ili 1, određuje se prema sledećoj formuli:

$$b_{ij} = \begin{cases} 0, & b_{ij} \neq a_j \\ 1, & b_{ij} = a_j \end{cases}$$

gdje a_j predstavlja jedinstvene vrijednosti u tekstualnim poljima, dok se j kreće od 0 do $k-1$, i se kreće od 1 do n. Nedostatak ove metode je što se, ako postoji veliki broj jedinstvenih vrijednosti, automatski povećava dimenzionalnost podataka. Zbog ovoga se ova metoda preporučuje za slučajeve kada postoji manji broj jedinstvenih vrijednosti.

Na slici 23 dat je primjer kodiranja *one-hot encoding* tehnikom.

Originalni podaci

id	boja
1	crvena
2	plava
3	zelena
4	plava

$\Rightarrow$

Podaci nakon kodiranja

id	boja_crvena	boja_plava	boja_zelena
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Slika 23: Primjer kodiranja *one-hot encoding* tehnikom

Slika 23 ilustruje kodiranje kolone `boja` iz jednostavnog skupa podataka. Kolona `boja` sastoji se od kategorija *crvena*, *plava* i *zelena*. Za svaku od ovih kategorija se, primjenom *one-hot encoding* tehnike, kreiraju posebne kolone `boja_crvena`, `boja_plava` i `boja_zelena`, dok se kolona `boja` briše. Ukoliko je red u originalnom skupu podataka sadržao vrijednost *crvena* u koloni `boja`, vrijednost kolone `boja_crvena` će u tom redu biti 1, dok će ostale vrijednosti tog reda biti 0. Istim postupkom se dobijaju vrijednosti u kolonama `boja_plava` i `boja_zelena`. Ova tehnika implementirana je u okviru `sklearn.preprocessing` modula pomoću klase `OneHotEncoder`.

U „Customer Personality Analysis“ skupu podataka imamo dvije kategorijske karakteristike: `Education` sa kategorijama *Undergraduate*, *Graduate* i *Postgraduate*, i `Living_With` sa kategorijama *Alone* i *Partner*. Da bismo utvrdili koje su promjenljive kategorijske možemo istražiti skup podataka pomoću metode *info* i naći one karakteristike čiji je tip *object*. Kodiranje karakteristike `Education` ćemo izvršiti *ordinal encoding* metodom jer postoji jasan redosled kategorija, i taj redosled je bitan za analizu podataka. Suprotno tome, kategorije *Alone* i *Partner* karakteristike `Living_With` nemaju prirodan redosled već jednostavno predstavljaju dvije različite kategorije, pa ćemo ih kodirati *label encoding* metodom.

Skup podataka *rfm* ne sadrži kategorijske promjenljive pa primjena ovih metoda nije potrebna.

3.2.5 Skaliranje podataka

Kada je riječ o algoritmima mašinskog učenja, uključujući i algoritme za klasterizaciju, ako su vrijednosti karakteristika bliže jedna drugoj, veća je šansa da algoritam bude dobro obučen [28]. U suprotnom, ako su podaci ili vrijednosti daleko jedni od drugih, proces obrade će trajati duže, a tačnost će biti smanjena. Kao rezultat toga, ako podaci sadrže tačke koje su na bilo koji način udaljene, njihovo približavanje će se vršiti metodom skaliranja podataka. Dvije glavne metode skaliranja podataka su normalizacija i standardizacija.

U statistici, normalizacija je postupak skaliranja u kojem nastojimo da sve tačke podataka smjestimo u opseg od 0 do 1, tako da budu bliže jedna drugoj. Normalizacija je relativno uobičajena metoda skaliranja podataka. U ovom postupku skaliranja, najmanja vrijednost bilo koje karakteristike se transformiše u 0, dok se najveća vrijednost karakteristike pretvara u

1. U suštini, normalizacija dijeli razliku između bilo koje vrijednosti i minimalne vrijednosti sa razlikom između maksimalne i minimalne vrijednosti. Odgovarajuća formula za normalizaciju podataka je:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}},$$

gdje je:

- X originalna vrijednost podataka,
- X_{min} minimalna vrijednost u skupu podataka,
- X_{max} maksimalna vrijednost u skupu podataka,
- X_{norm} normalizovana vrijednost.

Normalizacija je implementirana u sklopu Python-ove klase `MinMaxScaler` iz modula `sklearn.preprocessing`.

Metoda standardizacije temelji se na ideji da se podaci centriraju oko srednje vrijednosti svih podataka prikazanih u nekoj karakteristici sa standardnom devijacijom 1. Srednja vrijednost podataka biće 0, a standardna devijacija 1. Srednja vrijednost smatra se prosječnom vrijednošću bilo kog razmatranog dijela skupa brojeva, dok je standardna devijacija mjera disperzije u odnosu na srednju vrijednost podataka u statistici. Kao rezultat toga, podaci se ponovo skaliraju kako bi bili u obliku krive nakon skaliranja. Formula za standardizaciju je sledeća:

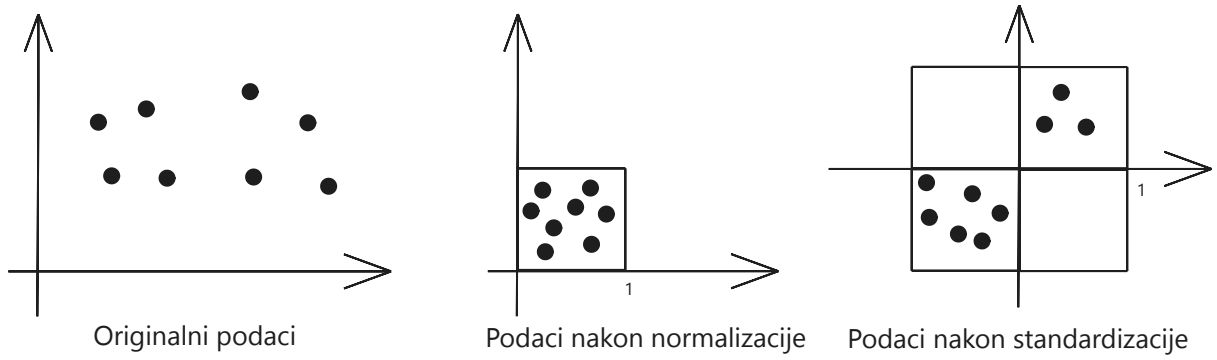
$$z = \frac{x - \mu}{\sigma},$$

gdje je:

- z standardizovana vrijednost,
- x originalna vrijednost podataka,
- μ srednja vrijednost skupa podataka,
- σ standardna devijacija skupa podataka.

Standardizacija se u Python programskom jeziku može koristiti pomoću `sklearn.preprocessing` modula i klase `StandardScaler`.

Postavlja se pitanje kada koristiti normalizaciju, a kada standardizaciju. Normalizacija se koristi kada je distribucija podataka nepoznata ili kada podaci nemaju Gausovu (normalnu) distribuciju. S druge strane, standardizacija se koristi kada podaci imaju normalnu distribuciju.



Slika 24: Izgled podataka nakon normalizacije i standardizacije

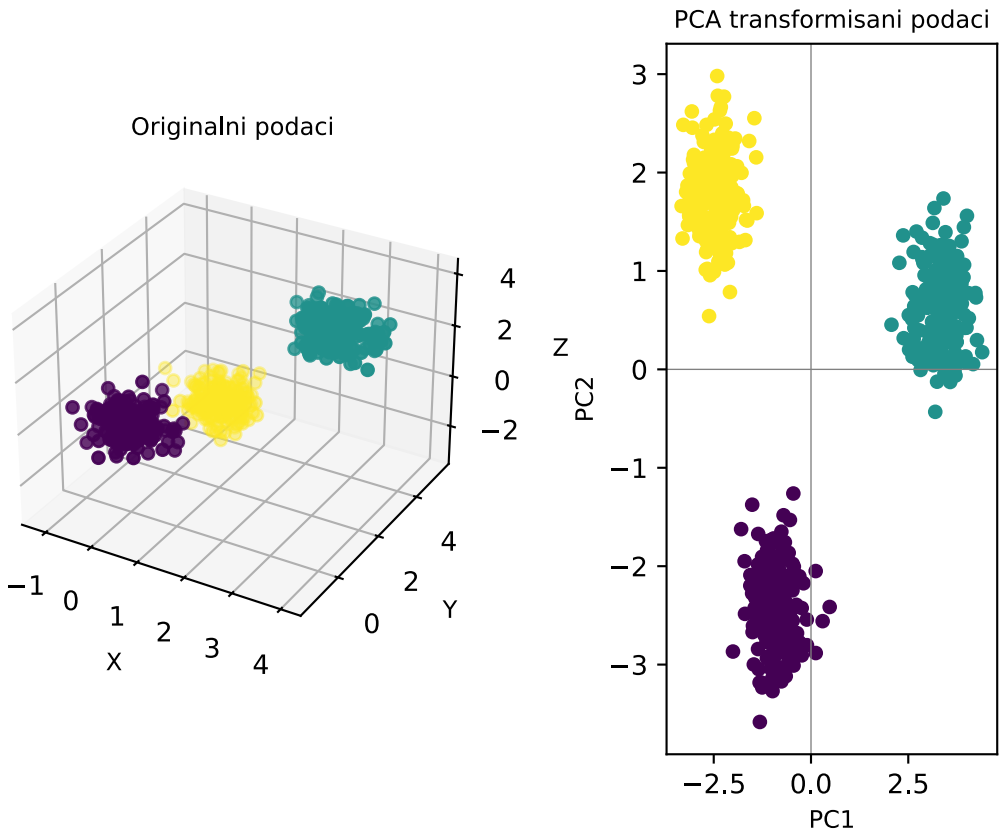
Slika 24 sastoji od tri dijela. Prvi dio prikazuje sirove podatke, drugi prikazuje podatke nakon normalizacije i treći dio prikazuje podatke nakon standardizacije. Kod drugog dijela, sve tačke se nalaze unutar crnog kvadrata, čime se podaci skaliraju na interval od 0 do 1. Treći dio prikazuje standardizaciju. Tačke su ponovo ograničene unutar crnog kvadrata, ali sada je kvadrat centriran oko koordinatnog početka (0, 0).

3.3 Redukcija dimenzionalnosti

Skupovi podataka sa velikim brojem karakteristikama nazivaju se podacima visoke dimenzionalnosti [29]. Kompleksni podaci sadrže mnogo korisnih informacija. Međutim, mnogo vremena za računanje i prostora za skladištenje se troši na obradu ovakvih podataka. Efektivne informacije sakrivene su u kompleksnim podacima, što otežava otkrivanje suštinskih karakteristika podataka. Za rješavanje ovog problema koristi se redukcija dimenzionalnosti.

Osnovni princip redukcije dimenzionalnosti karakteristika je mapiranje uzorka podataka iz prostora visoke dimenzionalnosti u relativno niže dimenzionalnosti. Osnovni zadatak je pronaći to mapiranje i dobiti efektivnu strukturu niske dimenzionalnosti koja sadrži skrivene informacije iz originalnih podataka visoke dimenzionalnosti.

Najčešće korišćena tehnika redukcije dimenzionalnosti je PCA (eng. *Principal Component Analysis*), koja transformiše podatke u nekoliko glavnih komponenti koje najbolje predstavljaju originalne informacije. Ove komponente su nezavisne i omogućavaju jednostavniju analizu podataka. PCA analizira kako da se iz originalnih podataka pronađu najvažnije komponente, a da se pri tome očuva što više informacija. Na osnovu matrice kovarijanse podataka, PCA koristi sopstvene vektore da stvori potprostor sa manjim brojem dimenzija, što smanjuje složenost podataka. Međutim, PCA pretpostavlja da su podaci linearni ili približno linearni, čime se povećava efektivnost u očuvanju ključnih informacija iz originalnih podataka.



Slika 25: Izgled proizvoljnog skupa podataka prije i poslije redukcije dimenzionalnosti

Na lijevom dijelu slike 25 prikazan je trodimenzionalni prostor sa tri jasno odvojena klastera, pri čemu se separabilnost ostvaruje kombinacijom sve tri dimenzije. Iako se klasteri u 3D prostoru mogu uočiti, njihova vizuelizacije je ograničena perspektivom i položajem posmatrača, što otežava intuitivno razumijevanje strukture podataka. Na desnoj strani prikazan je rezultat primjene metode glavnih komponenti (PCA), kojom je izvorni trodimenzionalni skup transformisan u dvodimenzionalni prostor zadržavajući najveći dio ukupne varijanse podataka. U ovom primjeru, prve dvije komponente objašnjavaju najveći dio informacija, što rezultira jasnom i preglednom separacijom klastera. Ovakva transformacija omogućava da se skrivena struktura podataka mnogo lakše uoči, a klasteri postaju vizuelno u potpunosti razdvojeni bez gubitka suštinske relacije. Prikaz jasno ilustruje glavnu prednost PCA analize – sposobnost da redukuje dimenzionalnost, ukloni redudantnost i sadrži najinformativnije karakteristike podataka, čineći proces klasterizacije efikasnijim i razumljivijim.

Redukcija dimenzionalnosti značajna je i za vizuelizaciju podataka. U situacijama kada postoji veliki broj dimenzija, njihovo interpretiranje i posmatranje postaje teško ili nemoguće, jer ne možemo vizuelizovati podatke koji imaju više od tri dimenzije. Dakle, redukcija dimenzionalnosti može smanjiti skup podataka na dvije ili tri dimenzije i omogućiti vizuelizaciju.

PCA tehnika redukcije dimenzionalnosti se u Python programskom jeziku može koristiti učitavanjem klase `PCA` iz modula `sklearn.decomposition`.

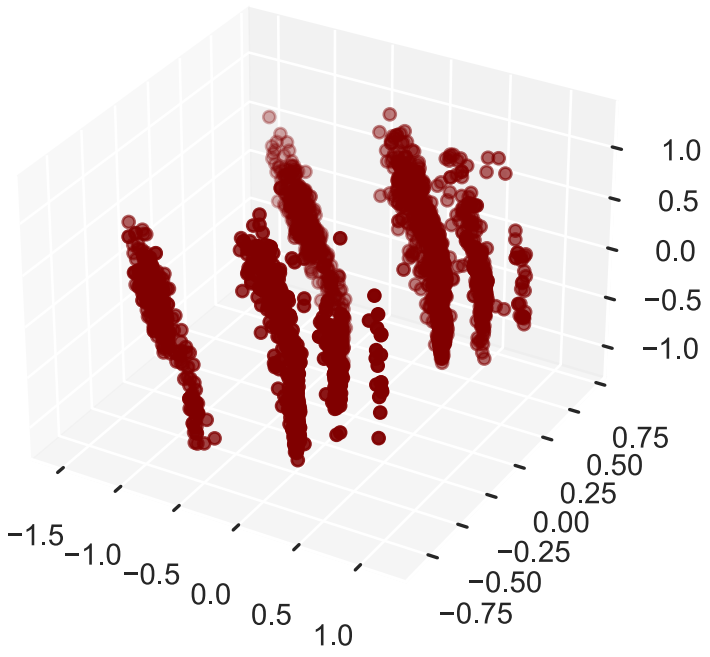
Za „Customer Personality Analysis“ skup podataka korišćen u ovom radu, izvršićemo redukciju dimenzionalnosti skupa podataka na tri komponente, tj. tri dimenzije. To znači da će novi skup podataka imati tri kolone sa nazivima `col1`, `col2` i `col3`. Primjenom PCA tehnike na posmatranom skupu podataka sačuvano je 64,6% ukupne varijanse. Ovakav nivo objašnjene varijanse smatra se zadovoljavajućim i očekivanim za marketinške i demografske podatke, gdje je informacija prirodno raspoređena preko preko većeg broja slabije korelisanih atributa. Iako ne postoji jedna dominantna dimenzija koja bi objasnila većinu varijanse, PCA uspijeva da sačuva najveći dio strukture podataka, što omogućava kvalitetnu trodimenzionalnu vizuelizaciju i predstavlja dobru osnovu za efikasnu primjenu algoritama za klasterizaciju.

Za *rfm* skup podataka nije potrebno vršiti redukciju dimenzionalnosti, jer se procesom pripreme podataka uklanjaju nepotrebne karakteristike, uključujući i `CustomerID`, koja se koristi samo za identifikaciju kupaca, a nije relevantna za analizu. U skupu podataka ostaju samo ključni faktori za analizu: `Amount`, `Frequency` i `Recency`. Ove karakteristike pružaju dovoljno informacija za segmentaciju podataka i dalju analizu. Dakle, skup podataka sadrži samo tri karakteristike što ga već čini skupom podataka niske dimenzionalnosti, te nema potrebe za dodatnom redukcijom dimenzionalnosti. Ipak, u okviru eksperimentalnog dijela, biće sprovedena redukcija dimenzionalnosti sa tri na dvije dimenzije radi vizuelizacije, provjere rezultata i potencijalnog poboljšanja kvaliteta klasterizacije. Pokazalo se, da je smanjenjem dimenzionalnosti sa tri na dvije dimenzije, očuvano čak 96% ukupne varijanse podataka. Ovim se potvrđuje da je redukcija dimenzionalnosti u ovom slučaju izuzetno efikasna i da dobijene komponente dobro reprezentuju stvarnu strukturu podataka, što klasterizaciju čini stabilnijom i interpretaciju pouzdanijom.

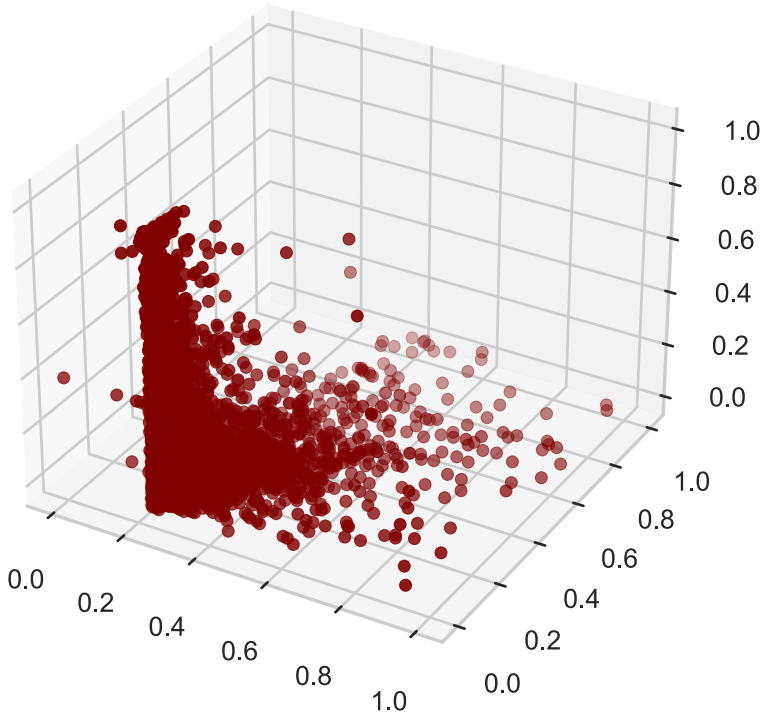
4 Eksperimentalna analiza i rezultati

U ovom poglavlju predstavljeni su rezultati klasterizacije na marketinškim podacima koji su prethodno prošli kroz faze pretprocesiranja i redukcije dimenzionalnosti. Detaljno su opisane implementacije odabranih algoritama, sa posebnim fokusom na izbor vrijednosti hiperparametara. Takođe, analizirane su metričke mjere kvaliteta klasterizacije za svaki algoritam, uz odgovarajuće vizuelizacije rezultata.

Prije primjene algoritama za klasterizaciju, podaci su prikazani na slikama 26 („Customer Personality Analysis“) i 27 („Online Retail“) radi bolje interpretacije i analize njihove strukture. Vizuelizacija omogućava uvid u potencijalne klastere i olakšava poređenje dobijenih rezultata nakon grupisanja podataka.



Slika 26: Projekcija skupa podataka „Customer Personality Analysis“ u prostoru redukovane dimenzionalnosti definisanim sa tri glavne komponente `col1` (PC1), `col2` (PC2) i `col3` (PC3) dobijene PCA metodom



Slika 27: Projekcija skupa podataka „Online Retail“ u trodimenzionalnom prostoru ključnih karakteristika za analizu: `Amount` (x-osa), `Frequency` (y-osa) i `Recency` (z-osa).

4.1 Implementacija algoritama za klasterizaciju

Za eksperimentalnu analizu odabrani su sledeći algoritmi za klasterizaciju: K-means, aglomerativni hijerarhijski, DBSCAN i spektralna klasterizacija. Implementacija je izvedena u programskom jeziku Python korišćenjem biblioteka `scikit-learn` (`sklearn`) i `NumPy`, uz dodatne biblioteke za vizuelizaciju: `Matplotlib` i `Seaborn`.

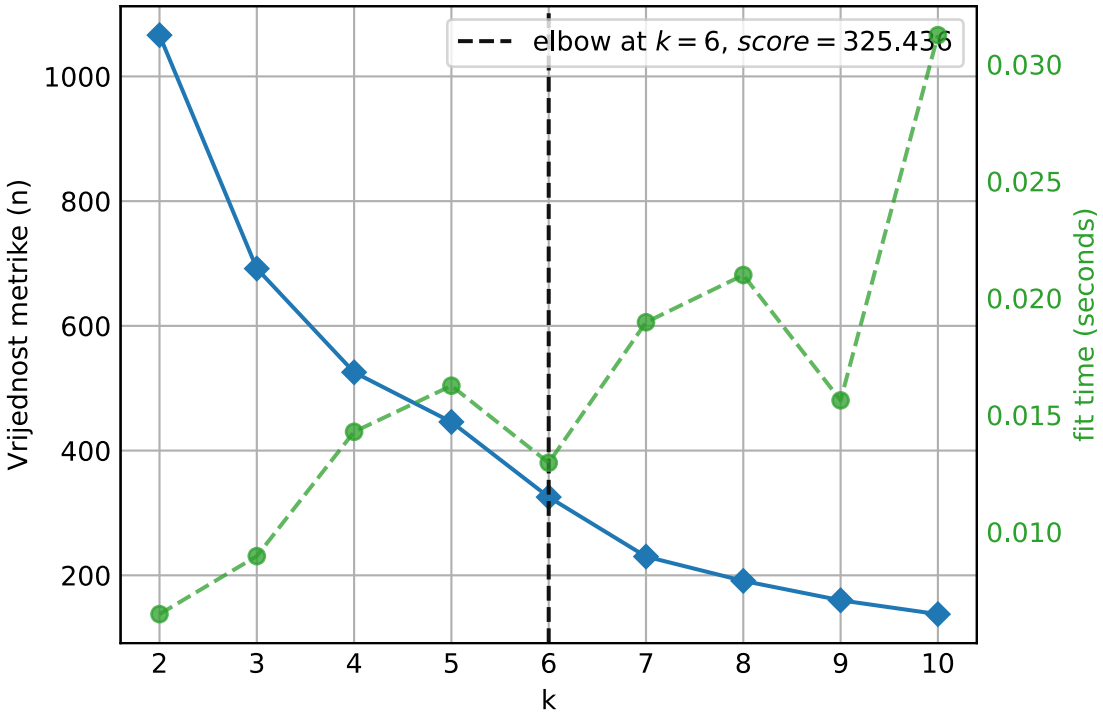
4.1.1 K-means algoritam

K-means je jedan od najkorišćenijiih algoritama za klasterizaciju, koji grupiše podatke tako da su elementi unutar jednog klastera što sličniji, dok su elementi unutar različitih klastera što udaljeniji. Ovaj algoritam implementiran je u Python programskom jeziku korišćenjem modula `sklearn.cluster` i klase `KMeans`.

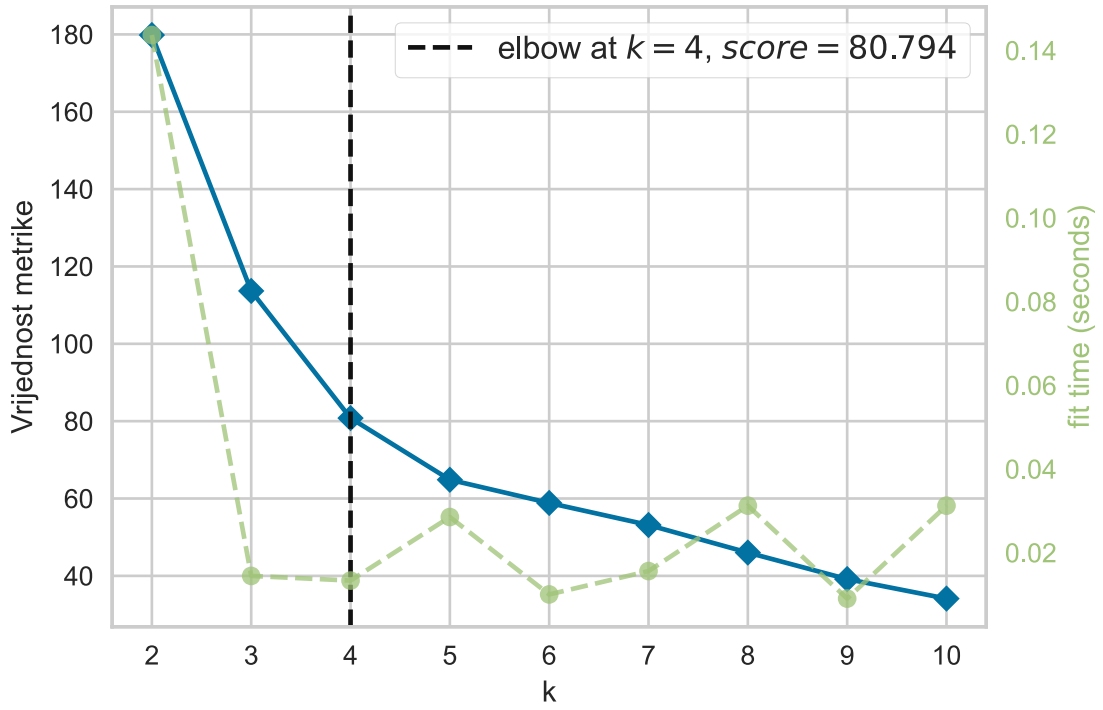
Jedan od ključnih izazova u K-means klasterizaciji je određivanje odgovarajućeg broja klastera K. U tu svrhu primijenjena je metoda „lakta“, koja analizira unutar-klastersku sumu kvadrata (WCSS – *Within-Cluster Sum of Squares*) za različite vrijednosti K. WCSS se računa kao zbir kvadrata udaljenosti svake tačke od centroida klastera kojem pripada.

S obzirom na to da korišćenjem običnog grafa metode „lakta“ nije uvijek lako odrediti mjesto gdje se formira „lakat“, u ovom istraživanju se koristi `KElbowVisualizer` klasa

modula `yellowbrick.cluster`. Ovaj pristup je odabran jer `KElbowVisualizer` na grafu prikazuje vertikalnu isprekidanu liniju koja označava odgovarajući broj klastera. Rezultati primjene `KElbowVisualizer` metode za skupove „Customer Personality Analysis“ i „Online Retail“ prikazani su na slikama 28 i 29, respektivno.



Slika 28: Prikaz `KElbowVisualizer`-a za „Customer Personality Analysis“



Slika 29: Prikaz `KElbowVisualizer`-a za „Online Retail“

Takođe, pored metode „lakta“ za odabir odgovarajućeg broja klastera korišćena je i *Silhouette Score* metrika. Ovo je popularna metrika za procjenu kvaliteta klasterizacije, koja se može koristiti i u svrhu odabira broja klastera. Funkcioniše tako što se odgovarajući algoritam pokreće više puta sa različitim vrijednostima broja klastera (najčešće od 2 do 10), dok se za svaki broj klastera štampa *Silhouette* koeficijent. Cilj je dobiti vrijednost koja je što bliža +1. Tabele 17 i 18 prikazuju rezultate korišćenja *Silhouette Score* metrike za odabir odgovarajućeg broja klastera kod skupova „Customer Personality Analysis“ i „Online Retail“.

Tabela 17: Prikaz rezultata *Silhouette Score*-a za odabir broja klastera kod „Customer Personality Analysis“ skupa podataka

Broj klastera	*Silhouette Score*
2	0.4524
3	0.4778
4	0.4623
5	0.4720
6	0.5342
7	0.5364
8	0.5165
9	0.5099
10	0.4885

Tabela 18: Prikaz rezultata *Silhouette Score*-a za odabir broja klastera kod „Online Retail“ skupa podataka

Broj klastera	*Silhouette Score*
2	0.5893
3	0.5466
4	0.5019
5	0.4481
6	0.4039
7	0.4049
8	0.3901
9	0.3892
10	0.3838

Za „Customer Personality Analysis“ skup podataka metode „lakta“ i *Silhouette Score* daju različite rezultate. Metoda „lakta“ predlaže 6 kao odgovarajući broj klastera, dok *Silhouette Score* pokazuje 7. S obzirom na to da *Silhouette Score* pokazuje veoma malu razliku u rezultatima za 6 i 7 klastera, odabran je broj 6 kao odgovarajući broj klastera.

Kod „Online Retail“ skupa podataka situacija je nešto složenija. Metoda „lakta“ pokazuje 4, dok *Silhouette Score* pokazuje 2 kao odgovarajući broj klastera. Kao kompromisno rješenje, postavićemo K na 3, jer je to vrijednost koja se po rezultatima nalazi između vrijednosti 2 i 4.

Još jedan pristup koji je doveo do značajno boljih rezultata je redukcija dimenzionalnosti kod „Online Retail“ sa tri na dvije dimezije. Ovakva transformacija često rezultira prostorom u kojem su klasteri izraženiji i jasnije razdvojeni, što algoritmima za klasterizaciju olakšava prepoznavanje strukture podataka. Važno je napomenuti da se ovaj postupak može sprovesti bez značajnog gubitka informacija – u konkretnom slučaju redukcijom na dvije dimenzije sačuvano je čak 96% ukupne varijanse, što potvrđuje da transformisani podaci i dalje odražavaju originalnu strukturu podataka. Redukcija dimezionalnosti sa tri na dvije dimenzije kog ovog skupa podataka biće primijenjena i kod ostalih algoritama za klasterizaciju.

Kada su u pitanju ostali hiperparametri algoritma, pokazalo se da njihove podrazumijevane vrijednosti daju najbolje rezultate. Izuzetak je hiperparametar `n_init` čija je podrazumijevana vrijednost 1 za `init = 'k-means++'` (metoda inicijalizacije). Ovaj hiperparametar je postavljen na 10 što znači da će se algoritam pokrenuti 10 puta sa različitim inicijalizacijama i odabraće se najbolje rješenje. Hiperparametar `n_init` promijenjen je za skup podataka „Customer Personality Analysis“.

4.1.2 Aglomerativni hijerarhijski algoritam

Aglomerativni hijerarhijski algoritam pripada hijerarhijskoj grupi metoda za klasterizaciju. Na početku svaka tačka je prikazana kao zaseban klaster. Najbliži klasteri se iterativno spajaju pomoću metoda za spajanje, sve dok se ne dobije jedan veliki klaster koji sadrži sve tačke podataka. Aglomerativni hijerarhijski algoritam za klasterizaciju implementiran je u okviru `AgglomerativeClustering` klase iz modula `sklearn.cluster`.

Ovaj algoritam ne zahtijeva unaprijed definisan broj klastera. Međutim, često se dešava da podaci ne budu dobro podijeljeni ukoliko se eksplicitno ne odredi broj klastera. Broj klastera kod ovog algoritma se, pored metoda „lakta“ i *Silhouette Score*-a, može odrediti i analizom dendrograma. S obzirom na to da su podaci pripremljeni na isti način za svaki algoritam, i ovdje ćemo postaviti broj klastera na 6 za „Customer Personality Analysis“ skup podataka, i na 3 za „Online Retail“ skup podataka.

Eksperimentisanjem sa drugim hiperparametrima došlo je do poboljšanja rezultata. Konkretno, promjene vrijednosti hiperparametara `metric` (metrika udaljenosti) i `linkage` (metoda spajanja) dovele su do boljih rezultata.

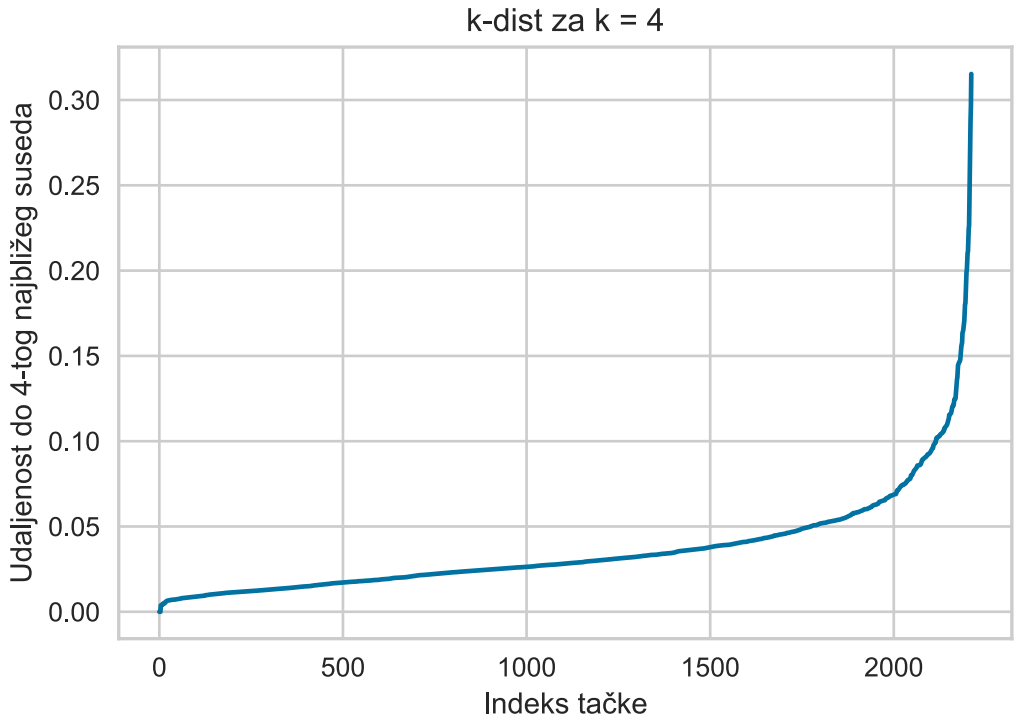
Kod „Customer Personality Analysis“ skupa podataka, pokazalo se da se promjenom hiperparametra `metric`, čija je podrazumijevana vrijednost `euclidean`, na `manhattan` (Manhattan distanca) dobijaju bolji rezultati. Manhattan distanca mjeri udaljenost između tačaka duž koordinatnih osa, što je korisno kada su klasteri nepravilnog oblika. Isto tako, posta-

vljanjem hiperparametra `linkage` na `average`, dobijeni su boji rezultati. *Average linkage* metoda spajanja izračunava prosječno rastojanje između svih tačaka u dva klastera, što omogućava uravnoteženo grupisanje podataka.

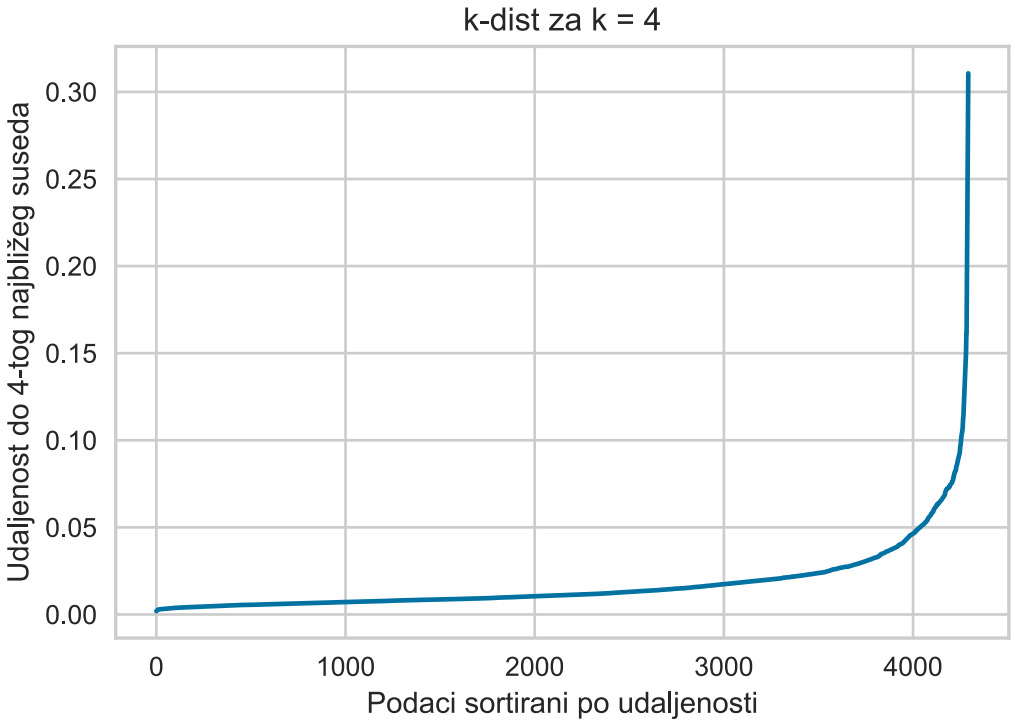
4.1.3 DBSCAN algoritam

DBSCAN algoritam pripada grupi algoritama zasnovanih na gustini podataka. Klasteri se formiraju oko *Core Point*-a, koji imaju najmanje N_{min} susjeda unutar radijusa ε. Susjedne tačke se pridružuju klasteru, dok se rijetko naseljene oblasti označavaju kao šum. DBSCAN je posebno koristan za otkrivanje klastera nepravilnog oblika i nije potrebno unaprijed zadati broj klastera. Međutim, efikasnost ovog algoritma zavisi od pravilnog izbora hiperparametara ε i N_{min}, a može imati poteškoća sa podacima neujednačene gustine. Algoritam je implementiran u Python programskom jeziku pomoću klase `DBSCAN` iz modula `sklearn.cluster`.

Prvi od dva pristupa u određivanju hiperparametra ε je grafik k-distanci. Grafik k-distanci prikazuje rastojanja od k-tog najbližeg susjeda za svaku tačku u skupu podataka, gdje su rastojanja sortirana u rastućem poretku. Na grafiku se obično traži „lakat“ - nagla promjena gdje rastojanja značajno rastu. Ova tačka označava granicu između gustih regiona (klastera) i rijetkih regiona (šuma), pa se njena vrijednost koristi kao predlog za ε hiperparametar. Grafici k-distanci za dva skupa podataka prikazani su na slikama 30 i 31.



Slika 30: Prikaz k-dist grafika za „Customer Personality Analysis“ skup podataka



Slika 31: Prikaz k-dist grafika za „Online Retail“ skup podataka

Slika 30 prikazuje da je optimalna vrijednost hiperparametra ε između 0,05 i 0,10. Međutim, odabirom ovakve vrijednosti ε hiperparametra, uz N_{min} postavljen na 4, klasterizacijom je dobijen veliki broj klastera. Razlog ovome je što je vrijednost ε previše mala, pa je omogućila prepoznavanje više gustih regiona kao odvojenih klastera, umjesto da ih grupiše u veće cjeline. Da bi se riješio ovaj problem, testirane su veće vrijednosti hiperparametra ε korišćenjem *Silhouette Score*-a i kao bolja vrijednost odabrana je 0,25. Ova vrijednost jasno pokazuje 4 dobro odvojena klastera kod „Customer Personality Analysis“ skupa podataka.

Za N_{min}, kod ovog skupa podataka, odabrana je vrijednost 4. Vrijednost je odabrana korišćenjem pravila: $N_{min} \geq dimenzionalnost\_podataka + 1$.

Situacija se prilično komplikuje kod drugog skupa podataka. Na slici 27 prikazana je grafička reprezentacija ovog skupa, iz koje se jasno vidi da su podaci dominantno grupisani u jednu gustu grupu. Ovo može predstavljati problem za DBSCAN algoritam, jer će vjerovatno prepoznati jedan veliki klaster, dok će preostale tačke, koje nisu dio te guste grupe, biti klasifikovane kao šum.

Na slici 31 prikazan je k-dist graf koji pokazuje da se optimalna vrijednost hiperparametra ε kreće od 0,04 do 0,08. Međutim, odabir bilo koje vrijednosti hiperparametara ε i N_{min} dovodi do problema koji je prethodno opisan, tj. do problema loše klasterizacije. Zbog prethodno navedenog, već sada se može zaključiti da DBSCAN nije dobar izbor za ovaj skup podataka.

4.1.4 Spektralna klasterizacija

Algoritam spektralne klasterizacije podatke dijeli tako što koristi prvih nekoliko sopstvenih vektora Laplasijan matrice grafa konstruisanog iz podataka. Prvo se formira graf sličnosti na osnovu međusobnih udaljenosti tačaka, zatim se računa Laplasijan matrica tog grafa. Nakon toga, određuje se nekoliko njenih sopstvenih vektora, koji se koriste za redukciju dimezionalnosti i transformaciju podataka u prostor pogodniji za primjenu klasičnih algoritama za klasterizaciju, poput K-means algoritma.

U Python programskom jeziku koristi se normalizovana spektralna klasterizacija, odnosno spektralna klasterizacija sa normalizovanom Laplasijan matricom, jer daje bolje performanse i stabilnije rezultate, posebno kada podaci imaju varijabilne gustine klastera. Normalizovana spektralna klasterizacija implementirana je u okviru klase `SpectralClustering` modula `sklearn.cluster`.

Kao i kod K-means metode, spektralna klasterizacija traži definisanje broja klastera. Koristeći se metodom „lakta“ i *Silhouette Score* metrikom, odabran je broj 6 kao odgovarajući broj klastera za „Customer Personality Analysis“ skup podataka.

Pored broja klastera, analizirani su rezultati dobijeni različitim konfiguracijama ostalih hiperparametara. Poboljšani rezultati dobijeni su podešavanjem hiperparametra `gamma` na 1,5, dok je podrazumijavana vrijednost 1,0. Hiperparametar `gamma` određuje koliko brzo opada vrijednost funkcije sličnosti sa povećanjem udaljenosti između podataka. Ukoliko je vrijednost ovog hiperparametra preniska, udaljenije tačke mogu imati značajnu sličnost, što može dovesti do spajanja nepovezanih grupa. Povećanjem hiperparametra `gamma` na 1,5, sličnost između udaljenih tačaka se smanjuje, čime se bolje naglašavaju lokalne strukture u podacima. To omogućava jasnije razdvajanje klastera i precizniju segmentaciju podataka, što je dovelo do poboljšanih rezultata klasterizacije.

Za „Online Retail“ skup podataka, odabran je broj 3 kao odgovarajući broj klastera. Podsjećanja radi, broj 3 je odabran kao kompromisno rješenje za ovaj skup podataka, jer su metode „lakta“ i *Silhouette Score* pokazivale vrijednosti 4 i 2, respektivno. Eksperimentisanjem se pokazalo da promjene drugih hiperparametara ne daju bolje rezultate.

4.2 Evaluacija rezultata klasterizacije

U ovoj sekciji analiziraju se rezultati klasterizacije primjenom različitih evaluacionih metrika i vizuelizacija. Korišćene su metričke mjere poput *Silhouette Score*-a i *Davies-Bouldin Index*-a kako bi se kvantitativno ocijenio kvalitet formiranih klastera. Pored numeričkih pokazatelja, rezultati su predstavljeni i vizuelno, pomoću 3D grafika. Cilj ove analize je objektivno sagledavanje sposobnosti svakog algoritma da prepozna strukturu podataka, bez interpretacije značaja pojedinih klastera, što će biti detaljnije razmotreno u sledećoj sekciji.

4.2.1 Kvantitativna evaluacija klastera

Silhouette Score (SS) metrika mjeri kvalitet klasterizacije tako što za svaku tačku računa prosječnu udaljenost do tačaka unutar svog klastera i prosječnu udaljenost do najbližeg susjednog klastera. Vrijednost *Silhouette Score*-a su u opsegu od -1 do +1, gdje važi:

- Vrijednost blizu +1 – tačka je dobro uklopljena u svoj klaster,
- Vrijednost blizu 0 – tačka je na granici između klastera,
- Vrijednost blizu -1 – tačka je vjerovatno dodijeljena pogrešnom klasteru.

Veći prosječan *Silhouette Score* za cijeli skup podataka znači bolju klasterizaciju.

Davies-Bouldin Index (DBI) mjeri kvalitet klasterizacije na osnovu sličnosti između klastera. Niže vrijednosti ove metrike ukazuju na bolje razdvojene i kompaktne klastere. Ova metrika se računa kao prosjek najgorih slučajeva preklapanja između klastera, pri čemu se za svaki klaster određuje odnos njegove unutrašnje raspršenosti i udaljenosti do najbližeg klastera. Vrijednosti se kreću od 0 do $+\infty$, pri čemu su niže vrijednosti bolje.

Primjenom različitih algoritama za klasterizaciju dobijeni su različiti rezultati metričkih mjera kvaliteta. Rezultati za oba skupa podataka prikazani su u tabelama 19 i 20.

Tabela 19: Rezultati metričkih mjera kvaliteta klasterizacije za „Customer Personality Analysis“ skup podataka

	Silhouette Score	*Davies-Bouldin Index*
K-means	0,74	0,40
Aglomerativni hijerarhijski	0,74	0,33
DBSCAN	0,67	0,43
Spektralna klasterizacija	0,75	0,38

Tabela 20: Rezultati metričkih mjera kvaliteta klasterizacije za „Online Retail“ skup podataka.

	Silhouette Score	*Davies-Bouldin Index*
K-means	0,85	0,25
Aglomerativni hijerarhijski	0,83	0,27
Spektralna klasterizacija	0,83	0,25

Analizirajući rezultate metričkih mjera kvaliteta, može se zaključiti da različiti algoritmi imaju različite performanse u zavisnosti od metrike. Tabela 19 pokazuje da je za K-means rezultat SS metrike 0,74, što ukazuje na dobro formirane klastere, dok njegov DBI iznosi 0,40, što znači da postoji određeni nivo preklapanja među klasterima. Aglomerativni hijerarhijski

algoritam daje identičan rezultat za SS, ali je njegov DBI niži i iznosi 0,33, što sugeriše da su klasteri najbolje razdvojeni korišćenjem ovog algoritma. Najslabije rezultate postiže DBSCAN algoritam sa SS od 0,67 i DBI od 0,43. Spektralna klasterizacija ostvaruje najbolji SS od 0,75, što znači da su klasteri dobro formirani, ali njen DBI je 0,38, što je bolje od K-means i DBSCAN-a, ali slabije od aglomerativnog hijerarhijskog algoritma.

Za konkretan skup podataka, svi algoritmi su postigli vrlo dobre rezultate. Jasno definisani i kompaktni klasteri vjerovatno su rezultat pravilnog izbora tehnika pretprocesiranja podataka i dobro podešenih hiperparametara algoritama. Naime, primjenom drugih pristupa (poput standardizacije umjesto normalizacije) postignuti su značajno lošiji rezultati. Konačno, na osnovu obije metrike, može se zaključiti da aglomerativni hijerarhijski algoritam postiže najbolje rezultate za „Customer Personality Analysis“ skup podataka.

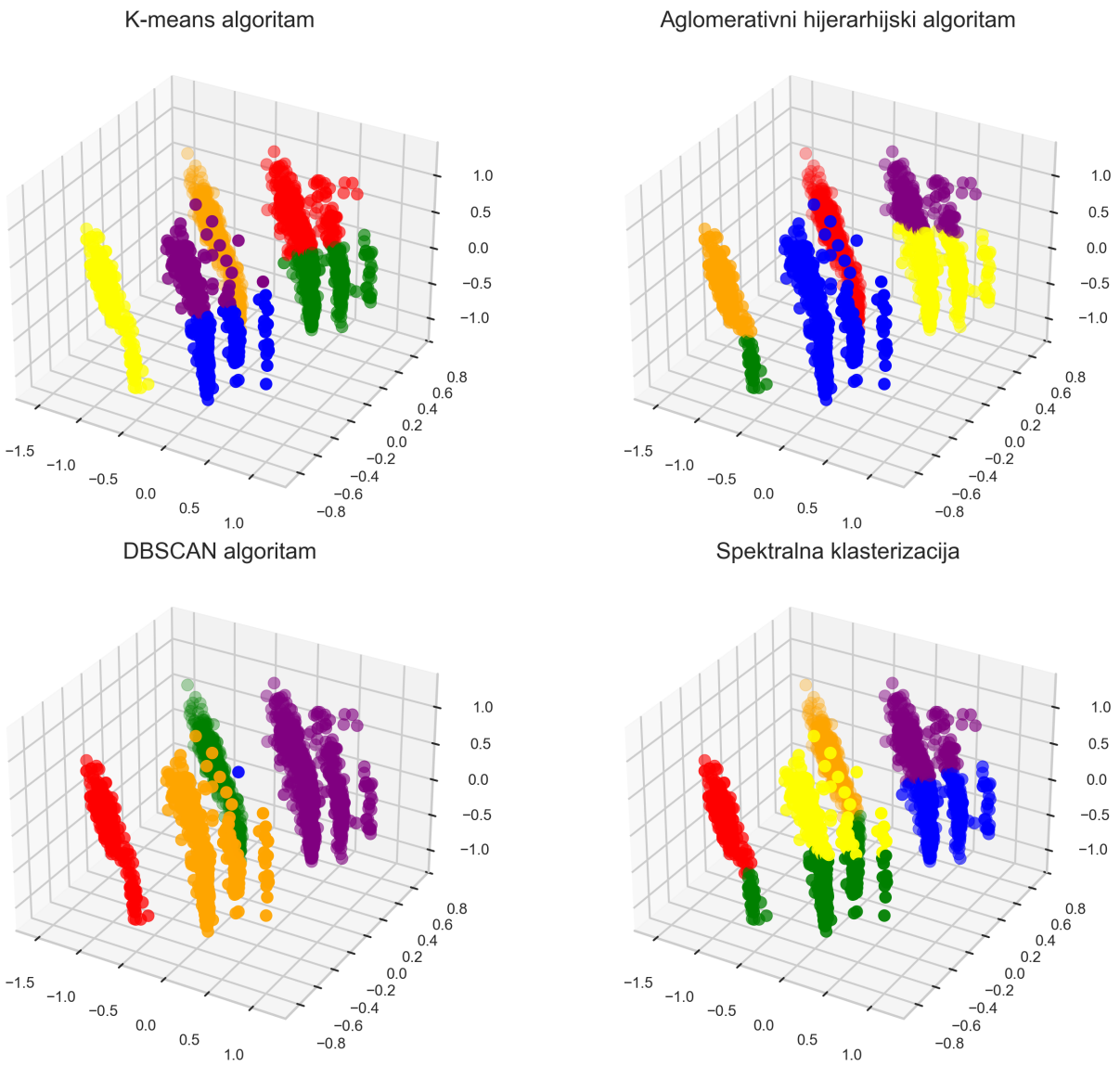
Tabela 20 prikazuje rezultate metričkih mjera kvaliteta klasterizacije kod skupa podataka „Online Retail“. K-means algoritam je postigao najbolje rezultate kod ovog skupa podataka. Sa *Silhouette Score*-om od 0,85 i *Davies-Bouldin Index*-om od 0,25, formirao je jasno razdvojene i kompaktne klastere. Neznatno slabije rezultate postigli su aglomerativni hijerarhijski algoritam (SS – 0,83 i DBI – 0,27) i spektralna klasterizacija (SS – 0,83 i DBI – 0,25).

Klasterizacija navedenog skupa podataka ocijenjena je kao vrlo uspješna. Jedan od ključnih razloga ovako uspješne klasterizacije je redukcija dimenzionalnosti. Eksperimenti su pokazali da je kvalitet klasterizacije podataka u trodimenzionalnom prostoru značajno lošiji u poređenju sa slučajem kada su podaci redukovani na dvije dimenzije, što je u skladu sa prethodno obrazloženim razlozima.

Kako je ranije pretpostavljeno, DBSCAN je nepovoljno podijelio podatke. Uzimajući u obzir visoku gustinu podataka, formirao je dva klastera, od kojih je jedan označen kao -1, tj. u tom klasteru se nalaze tačke označene kao šum. Zbog prethodno navedenog, nije bilo moguće izračunati SS i DBI, te se zbog toga rezultati ne nalaze u tabeli 20. DBSCAN više neće biti uključen u analizu za konkretan skup podataka.

4.2.2 Vizuelna analiza rezultata klasterizacije

Vizuelna analiza rezultata klasterizacije omogućava intuitivno sagledavanje formiranih klastera i njihove međusobne separacije. Grafički prikazi, poput 3D *scatter* grafika, pomažu u procjeni kvaliteta klasterizacije i identifikaciji potencijalnih problema, poput preklapanja klastera ili prisustva šuma. U ovom dijelu biće prikazane vizuelizacije rezultata dobijenih primjenom odabranih algoritama, uz analizu njihove strukture. Cilj ove analize je da se prikaže kako različiti algoritmi grupišu iste podatke na različite načine, uzimajući u obzir kriterijume kao što su udaljenost, gustina i povezanost među tačkama.



Slika 32: Trodimenzionalna reprezentacija rezultata klasterizacije nad „Customer Personality Analysis“ skupom podataka. Ose predstavljaju prve tri glavne komponente: `col1`, `col2` i `col3`.

Na slici 32 vizuelizovani su rezultati primjene odabranih algoritama na „Customer Personality Analysis“ skupu podataka. Svaki od 3D grafika prikazuje rezultat korišćenja različite metode klasterizacije. Različite boje označavaju različite klastere koje su agloritmi prepoznali, dok tačke predstavljaju kupce. Kratka analiza vizuelizacije podataka je data u nastavku.

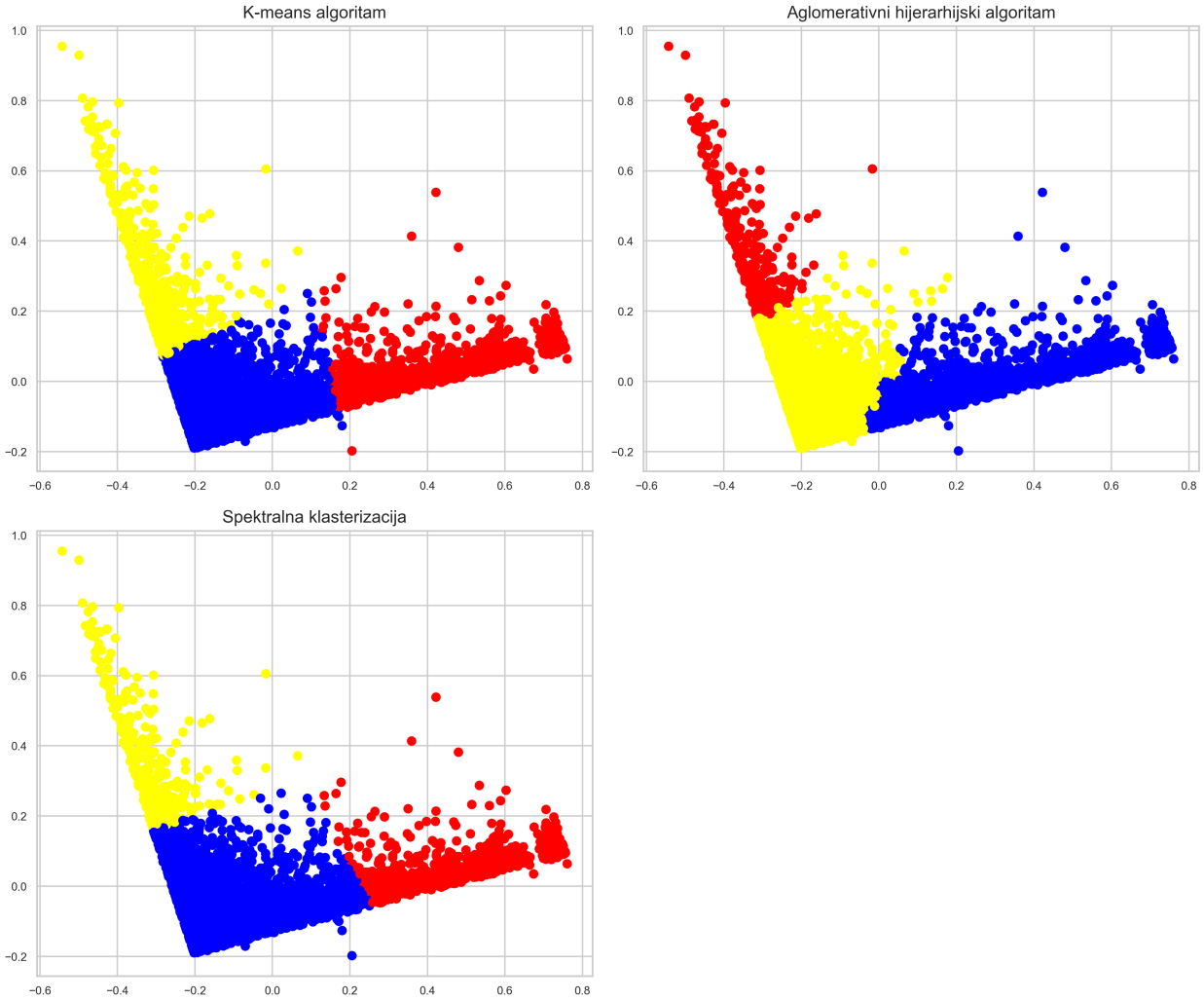
Rezultat primjene K-means algoritma prikazan je u gornjem lijevom uglu slike. Klasteri su jasno odvojeni i gotovo pravilno oblikovani, što je karakteristično za K-means, jer najbolje funkcioniše kada su klasteri sferični i jednako raspoređeni. Sa slike se može zaključiti da je algoritam efikasno podijelio podatke.

U gornjem desnom uglu prikazan je rezultat klasterizacije aglomerativnim hijerarhijskim algoritmom. Vizuelni rezultat podsjeća na K-means, ali se primjećuju određene razlike u veličini i distribuciji klastera. Na primjer, neki klasteri su manji ili drugačije raspoređeni. Ova

metoda često može bolje uočiti skrivene obrasce unutar podataka jer omogućava formiranje klastera različitih veličina i oblika.

U donjem lijevom uglu prikazan je rezultat primjene DBSCAN algoritma. Podaci su podijeljeni u pet klastera, pri čemu je jedan klaster određen za šum. Uočava se jasna razlika u raspodjeli podataka po klasterima u odnosu na K-means i aglomerativni hijerarhijski algoritam – DBSCAN algoritam nudi fleksibilniji i realističniji prikaz klasne strukture, ali njegova osjetljivost na ε i N_{min} hiperparametre zahtijeva pažljivu kalkulaciju. Iako je DBSCAN imao lošije rezultate sa slike se vidi da je on vjerovatno najrealnije podijelio podatke.

Na kraju, donji desni prikaz predstavlja rezultat spektralne klasterizacije. Klasteri su dobro definisani i jasno odvojeni, a primjećuje se da su neke tačke koje su kod prethodnih metoda bile blizu granica ovdje bolje svrstane. Ponovo imamo klasterizaciju sličnu onoj kod K-means i aglomerativnog hijerarhijskog algoritma.



Slika 33: Dvodimenzionalna reprezentacija rezultata klasterizacije nad „Online Retail“ skupom podataka. Ose predstavljaju dvije glavne komponente dobijene redukcijom dimenzionalnosti sa tri na dvije dimenzije, i to: `col1` i `col2`.

Na slici 33 prikazani su 2D *scatter* dijagrami koji prikazuju rezultate klasterizacije za „Online Retail“ skup podataka.

Na gornjem lijevom dijagramu, K-means algoritam grupisao je podatke u tri klastera. Zbog prirode K-means algoritma da pronalazi sferične klastere bazirane na minimizaciji WCSS-a, primjetno je da su klasteri prilično kompaktni i dobro razdvojeni. Granice između klastera su relativno jasne sa minimalnim preklapanjem, što ukazuje na dobro definisane i odvojene grupe unutar podataka.

Gornji desni dijagram predstavlja rezultate klasterizacije aglomerativnih hijerarhijskim algoritmom. Vizuelno, distribucija klastera je gotovo identična onoj kod K-means algoritma. Ovo sugeriše da oba algoritma identifikuju istu fundamentalnu strukturu u podacima. Klasteri su, takođe, dobro razdvojeni i kompaktni.

Rezultati spektralne klasterizacije prikazani su na donjem lijevom dijagramu. Klasteri su, ponovo, vrlo slično raspoređeni kao kod prethodna dva algoritma.

Sva tri algoritma su izuzetno dosljedno identifikovala tri dobro definisana i vizuelno jasno odvojena klastera u ovom skupu podataka. Konzistentnost rezultata različitih algoritama ukazuje na robustnu i jasnu segmentaciju. Ovo je snažan pokazatelj da su pronađeni segmetni prirodno prisutni u podacima i da su vjerovatno veoma relevantni za marketinške svrhe. S obzirom na jasnu separaciju, očekuje se da će i profilisanje ovih segmenata otkriti značajne razlike u ponašanju i karakteristikama kupaca, što će omogućiti kreiranje ciljanih marketinških strategija.

4.3 Interpretacija i profilisanje klastera

Ovaj dio rada posvećen je analizi i interpretaciji dobijenih klastera, što je važan korak u prevođenju rezultata klasterizacije u razumljive marketinške uvide. Cilj je da, na osnovu prethodno sprovedene kvantitativne i vizuelne evaluacije, produbimo razumijevanje strukture tržišta i ponašanja kupaca. Kroz profilisanje svakog definisanog klastera, istraživaće se njegove specifične karakteristike, kao što su demografski podaci, kupovne navike i slično. Konačno, na osnovu detaljnog uvida u svaki segment, biće formulisane konkretne marketinške implikacije i preporuke za ciljane strategije, čime se obezbjeđuje primjenljivost rezultata u realnom poslovnom okruženju.

Nakon sprovedene klasterizacije marketinških podataka primjenom četiri algoritma, uočeno je da se, uprkos razlikama u broju i raspodjeli tačaka po klasterima, pojedini potrošački segmenti stabilno pojavljuju bez obzira na metodu klasterizacije. Drugim riječima, klasteri koji pripadaju različitim algoritmima često grupišu slične tipove korisnika, iako se tehnički razlikuju po indeksima, granicama i broju članova.

U nastavku su opisani karakteristični segmenti korisnika, identifikovani na osnovu uvida iz svih algoritama. Uz svaki segment je naznačeno u kojim se klasterima (po algoritmu) taj segment pojavljuje.

„Customer Personality Analysis“ skup podataka analiziran je na osnovu karakteristika

`Age` (godine starosti kupca), `Total_Children` (ukupno djece u domaćinstvu), `Family_Size` (broj članova domaćinstva), `Is_Parent` (da li je kupac roditelj ili ne), `Education` (nivo obrazovanja) i `Living_With` (da li ima partnera/ku). Analizom ovog skupa dobijeni su sledeći segmenti:

- **Klaster A** (prisutan kao: klaster 0 – K-means, klaster 0 – AHC, klaster 1 – DBSCAN, klaster 5 – spektralna klasterizacija):
  - **Bračni status**: nemaju partnera/ku;
  - **Roditeljstvo**: roditelji su;
  - **Broj članova domaćinstva**: uglavnom 2 ali postoji i određeni broj sa 3 člana domaćinstva;
  - **Ukupan broj djece u domaćinstvu**: uglavnom 1, ali postoji i određeni broj sa 2;
  - **Potrošnja**: niska;
  - **Prihodi**: niski;
  - **Potrošnja na meso, vino, voće, ribu, slatkiše i zlato**: niska.
- **Klaster B** (prisutan kao: klaster 1 – K-means, klaster 5 – AHC, klaster 3 – DBSCAN, klaster 4 – spektralna klasterizacija):
  - **Bračni status**: imaju partnera/ku;
  - **Roditeljstvo**: roditelji su;
  - **Broj članova domaćinstva**: 3;
  - **Ukupan broj djece u domaćinstvu**: 1;
  - **Potrošnja**: srednja;
  - **Prihodi**: srednji;
  - **Potrošnja na meso, voće, ribu i slatkiše**: srednja;
  - **Potrošnja na vino i zlato**: visoka.
- **Klaster C** (prisutan kao: klaster 2 – K-means, klaster 3 – AHC, klaster 0 – DBSCAN, klaster 3 – spektralna klasterizacija):
  - **Bračni status**: nemaju partnera/ku;
  - **Roditeljstvo**: nisu roditelji;
  - **Broj članova domaćinstva**: 1;
  - **Ukupan broj djece u domaćinstvu**: 0;
  - **Potrošnja**: visoka;

  - **Prihodi**: srednji do visoki;
  - **Potrošnja na meso, vino, voće, ribu, slatkiše i zlato**: srednja do visoka.

- **Klaster D** (prisutan kao: klaster 3 – K-means, klaster 1 – AHC, klaster 2 – DBSCAN, klaster 1 – spektralna klasterizacija):
  - **Bračni status**: imaju partnera/ku;
  - **Roditeljstvo**: nisu roditelji;
  - **Broj članova domaćinstva**: 2;
  - **Ukupan broj djece u domaćinstvu**: 0;
  - **Potrošnja**: visoka;
  - **Prihodi**: srednji do visoki;
  - **Potrošnja na meso, vino, voće, ribu, slatkiše i zlato**: srednja do visoka.

- **Klaster E** (prisutan kao: klaster 4 – K-means, klaster 2 – AHC, klaster 3 – DBSCAN, klaster 0 – spektralna klasterizacija):
  - **Bračni status**: imaju partnera/ku;
  - **Roditeljstvo**: roditelji su;
  - **Broj članova domaćinstva**: 3 do 4;
  - **Ukupan broj djece u domaćinstvu**: 1 do 2;
  - **Potrošnja**: niska;
  - **Prihodi**: srednji do visoki;
  - **Potrošnja na meso, vino, voće, ribu, slatkiše i zlato**: niska.

- **Klaster F** (prisutan kao: klaster 5 – K-means, klaster 0 – AHC, klaster 1 – DBSCAN, klaster 2 – spektralna klasterizacija):
  - **Bračni status**: nemaju partnera/ku;
  - **Roditeljstvo**: roditelji su;
  - **Broj članova domaćinstva**: 2;
  - **Ukupan broj djece u domaćinstvu**: 1;
  - **Potrošnja**: srednja;
  - **Prihodi**: srednji;
  - **Potrošnja na meso, voće, ribu i slatkiše**: srednja;
  - **Potrošnja na vino i zlato**: visoka.

Dobijeni klasteri otkrivaju jasno odvojene segmente korisnika, sa značajnim implikacijama za ciljanje marketinških kampanja. Klaster A obuhvata korisnike sa niskim prihodima i ograničenom potrošnjom, koji su izuzetno cjenovno osjetljivi. Efikasne strategije za ovaj segment uključuju promotivne akcije i pristupačne proizvode. Klaster B čine stabilne porodice sa srednjim prihodima, koje pokazuju veća izdvajanja za vino i zlato. Ovaj segment je pogodan za ponude koje kombinuju funkcionalnost i dozu luksuza. Klaster C okuplja korisnike bez partnera i bez djece, sa srednjim do visokim prihodima i snažnijim potrošačkim obrascima, posebno u luksuznim kategorijama. Klaster D sadrži partnere bez djece, koji raspolažu značajnim prihodima i pokazuju visok nivo potrošnje – predstavljaju odličnu priliku za proizvode vezane za iskustva, putovanja i kvalitetan životni stil. Klaster E je specifičan po tome što se radi o porodicama sa srednjim do visokim prihodima, ali izuzetno niskom potrošnjom – mogu predstavljati potencijal za edukativne kampanje, popuste i druge podsticaje u kupovini. Na kraju, klaster F pokazuje usmjeren nivo prihoda i potrošnje, ali je naglašeno i povećano trošenje u pojedinim kategorijama (vino i zlato), što sugeriše na potencijalno aspirativni potrošački stil. Ovaj segment bi mogao pozitivno reagovati na selektivne ponude i proizvode koji kombinuju kvalitet i imidž.

Za „Online Retail“ priprema i klasterizacija podataka zasniva se na tri faktora RFM (*Recency*, *Frequency* i *Monetary*). Dakle, klasteri će biti profilisani na osnovu broja dana od poslednje kupovine korisnika (*Recency*), broja transakcija po korisniku (*Frequency*) i ukupnog potrošenog iznosa (*Monetary*). Analizom rezultata klasterizacije ovog skupa podataka otkriveni su sledeći segmenti:

- **Klaster A** (prisutan kao: 0 – K-means, 0 – AHC, 1 – spektralna klasterizacija)
  - ***Recency***: nisu kupovali određeno vrijeme;
  - ***Frequency***: povremeno kupuju;
  - ***Amount***: srednja potrošnja.

- **Klaster B** (prisutan kao: 1 – K-means, 2 – AHC, 0 – spektralna klasterizacija)
  - ***Recency***: skoro su kupovali;
  - ***Frequency***: često kupuju;
  - ***Amount***: visoka potrošnja.

- **Klaster C** (prisutan kao: 2 – K-means, 1 – AHC, 2 – spektralna klasterizacija)
  - ***Recency***: dugo nisu kupovali;
  - ***Frequency***: rijetko kupuju;
  - ***Amount***: niska potrošnja.

U okviru sprovedene klaster analize, identifikovana su tri segmenta korisnika: Klaster A, Klaster B i Klaster C. Ovi klasteri se značajno razlikuju prema svojim potrošačkim navikama izraženim kroz RFM metrike. Klaster A obuhvata korisnike koji nisu kupovali u skorije vrijeme, ali još uvijek povremeno obavljaju kupovine i ostvaruju srednju potrošnju. Ova grupa predstavlja solidnu osnovu za marketinške aktivnosti usmjerene ka reaktivaciji i podsticaju lojalnosti kroz ciljanje personalizovanim promocijama koje podstiču učestaliju kupovinu. Klaster B se izdvaja kao najvrjedniji segment – korisnici iz ove grupe su skoro kupovali, često obavljaju transakcije i troše značajne iznose. Marketinški napori prema ovom segmentu treba da budu fokusirani na zadržavanje i povećanje vrijednosti kroz ekskluzivne ponude, nadogradnju prodaje, kao i posebne pogodnosti u okviru programa za lojalne korisnike. S druge strane, klaster C okuplja korisnike koji nisu dugo bili aktivni, rijetko kupuju i ostvaruju nisku potrošnju. Budući da postoji mala vjerovatnoća za značajan povratak ovih korisnika, preporučuje se sprovođenje kampanja koje će ih ponovo zainteresovati sa jasno izraženim benefitima, kao i istraživanje razloga njihove neaktivnosti putem anketa.

5 Zaključak

U ovom radu sprovedena je klasterizacija nad dva skupa podataka: „Customer Personality Analysis“ i „Online Retail“, sa ciljem segmentacije korisnika na osnovu njihovih kupovnih navika i ponašanja. Korišćeni su različiti algoritmi klasterizacije, uključujući K-means, aglomerativni hijerarhijski algoritam, DBSCAN i spektralnu klasterizaciju.

Rezultati su pokazali da uspješnost algoritama u velikoj mjeri zavisi od prirode i strukture podataka. Za skup „Customer Personality Analysis“, aglomerativni hijerarhijski algoritam se pokazao kao najefikasniji u formiranju jasno diferenciranih i poslovno interpretabilnih klastera, pružajući uvid u različite tipove potrošača. Nasuprot tome, K-means algoritam je dao najstabilnije i najlogičnije klastere kod „Online Retail“ skupa, čime je omogućio efikasnu identifikaciju lojalnih, povremenih i neaktivnih kupaca.

Na osnovu dobijenih klastera formulisane su i marketinške preporuke usmjerene ka personalizaciji pristupa korisnicima – od zadržavanja najvrjednijih klijenata do reaktivacije neaktivnih. Ovi nalazi potvrđuju da primjena odgovarajućih metoda klasterizacije, uz jasno definisane ciljeve, može značajno doprinijeti unapređenju poslovne strategije i donošenju odluka zasnovanih na podacima.

Jedna od ključnih doprinosa uspješnosti klaster analize u ovom radu bio je i pravilan izbor tehnika za pretprocesiranje podataka i redukciju dimenzionalnosti. Normalizacija numeričkih vrijednosti, uklanjanje *outlier*-a i brisanje redova sa nepostojećim vrijednostima omogućili su algoritmima da otkriju stvarne obrasce u ponašanju korisnika. Korišćenje PCA tehnike za redukciju dimezionalnosti dodatno je olakšalo vizuelizaciju podataka i omogućilo bolju separaciju klastera u nižedimenzionalnom prostoru. Kao rezultat toga, svi primijenjeni algoritmi za klasterizaciju proizveli su slične i konzistentne segmente, što povećava pouzdanost i interpretabilnost dobijenih rezultata.

U budućim istraživanjima preporučuje se korišćenje dodatnih izvora (korisničke ankete, ponašanja na veb sajtovima ili aktivnost na društvenim mrežama), kao i primjena naprednijih metoda poput metoda zasnovanih na dubokom učenju. Posebno je preporučljivo sprovesti analizu koja bi pratila promjene u ponašanju korisnika kroz vrijeme i efekte primijenjenih marketinških strategija po klasterima, kako bi se dodatno potvdila njihova poslovna vrijednost i relevantnost.

Literatura

[1] Tsiptsis KK, Chorianopoulos A. Data Mining Techniques in CRM: Inside Customer Segmentation. Chichester: Wiley; 2009.

[2] Ben-David S. Clustering – What Both Theoreticians and Practitioners are Doing Wrong. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). 2018 Jul 2–7; New Orleans.

[3] Rodríguez A, Laio A, Ochoa S. Clustering algorithms: a comparative approach. PLOS ONE. 2019;14(3).

[4] Fan X, Yue Y, Sarkar P, Wang YXR. On hyperparameter tuning in general clustering problems. In: Proceedings of the 37th International Conference on Machine Learning (ICML); 2020 Jul 13–18; Virtual Conference. PMLR; 2020.

[5] Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognit Lett. 2010;31(8).

[6] Xu R, Wunsch D. Survey of clustering algorithms. IEEE Trans Neural Netw. 2005 May;16(3).

[7] Müller AC, Guido S. Introduction to machine learning with Python: a guide for data scientists. Sebastopol: O'Reilly Media; 2016.

[8] Berry MW, Mohamed A, Yap BW, editors. Supervised and unsupervised learning for data science. Cham: Springer; 2021.

[9] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Comput Surv. 1999;31(3).

[10] Kumar V, Chhabra JK, Kumar D. Performance evaluation of distance metrics in the clustering algorithms. INFOCOMP J Comput Sci. 2014;13(1).

[11] Aggarwal CC, Reddy CK, editors. Data clustering: algorithms and applications. Boca Raton: CRC Press; 2014.

[12] Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. 2007.

[13] Yahia R, Dhieb N, Ben Messaoud M, Ghézala HB. Overview of agglomerative hierarchical clustering (AHC). In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE; 2020.

[14] Cabezas LMC, Izbicki R, Stern RB. Hierarchical clustering: Visualization, feature importance and model selection. Appl Soft Comput. 2023 Jul;141:110303.

[15] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96); 1996.

[16] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems. 2002;14

[17] Von Luxburg U. A tutorial on spectral clustering. Stat Comput. 2007 Dec;17(4):395–416.

[18] Zhang X, You Q. An improved spectral clustering algorithm based on random walk. Front Comput Sci China. 2011;5(3).

[19] Von Luxburg U, Belkin M, Bousquet O. Consistency of spectral clustering. Ann Stat. 2008;36(2):555–586.

[20] Setiady DA, Leong H. Implementation of K-Means algorithm Elbow method and Silhouette coefficient for rainfall classification. Proxies. 2020;4(1).

[21] Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell. 1979 Apr;1(2).

[22] Acheme ID, Enoyoze E. Customer personality analysis and clustering for targeted marketing. Int J Sci Res Arch. 2024;12(01).

[23] John JM, Shobayo O, Ogunleye B. An exploration of clustering algorithms for customer segmentation in the UK retail market. Analytics. 2024;2(4):42

[24] García S, Luengo J, Herrera F. Data preprocessing in data mining. Cham: Springer; 2015.

[25] Heaton J. An empirical analysis of feature engineering for predictive modeling. Proceedings of the SoutheastCon 2016; 2016 Mar 17-20; Norfolk, VA, USA.

[26] Singh K, Upadhyaya S. Outlier detection: applications and techniques. IJCSI Int J Comput Sci Issues. 2012;9(1):3.

[27] Bolikulov F, Nasimov R, Rashidov A, Akhmedov F, Cho YI. Effective methods of categorical data encoding for artificial intelligence algorithms. Mathematics. 2024;12(16).

[28] Sharma V. A study on data scaling methods for machine learning. Int J Glob Acad Sci Res. 2022;1(1).

[29] Jia W, Sun M, Lian J, Hou S. Feature dimensionality reduction: a review. Complex Intell Syst. 2022;8.

Izjava o istovjetnosti štampane i elektronske verzije master rada

Ime i prezime autora: Lazar Trifunović

Broj indeksa/upisa: 15/22

Naslov rada: Segmentacija marketinških podataka primjenom algoritama za klasterizaciju

Mentor: doc. dr Miloš Brajović

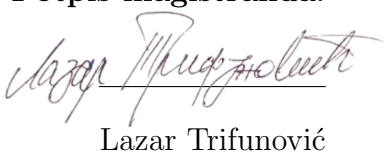
Potpisani/a: Lazar Trifunović

Izjavljujem

da je štampana verzija mog master rada istovjetna elektronskoj verziji koju sam predao/la za objavljivanje u Digitalni arhiv Univerziteta Crne Gore.

Istovremeno izjavljujem da dozvoljavam objavljivanje mojih ličnih podataka u vezi sa dobijanjem akademskog naziva master nauka, kao što su ime i prezime, godina i mjesto rođenja, naslov master rada i datum odbrane rada.

Potpis magistranda:



Lazar Trifunović

Podgorica, 24.11.2025. godine

Izjava o korišćenju

Ovlašćujem Univerzitetsku biblioteku da u Digitalnom arhivu Univerziteta Crne Gore pohrani moj master rad pod nazivom:

„Segmentacija marketinških podataka primjenom algoritama za klasterizaciju“

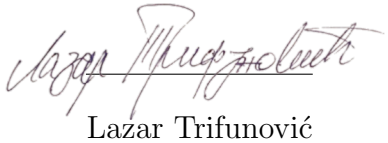
koji je moje autorsko djelo.

Master rad sa svim prilozima predao/la sam u elektronskom formatu pogodnom za trajno arhiviranje.

Moj master rad pohranjen u Digitalnom arhivu Univerziteta Crne Gore mogu da koriste svi koji poštuju odredbe sadržane u odabranom tipu licence Kreativne zajednice (Creative Commons) za koju sam se odlučio/la.

1. Autorstvo
2. Autorstvo - nekomercijalno
3. Autorstvo - nekomercijalno - bez prerade
4. Autorstvo - nekomercijalno - dijeliti pod istim uslovima
5. Autorstvo - bez prerade
6. Autorstvo - dijeliti pod istim uslovima

Potpis magistranda:



Lazar Trifunović

Podgorica, 24.11.2025. godine