Pronalaženje grupa, sličnih entiteta i čestih podskupova u analizi podataka

Bernt, Filip

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1542

PRONALAŽENJE GRUPA, SLIČNIH ENTITETA I ČESTIH PODSKUPOVA U ANALIZI PODATAKA

Filip Bernt

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1542

PRONALAŽENJE GRUPA, SLIČNIH ENTITETA I ČESTIH PODSKUPOVA U ANALIZI PODATAKA

Filip Bernt

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1542

Pristupnik:	**Filip Bernt (0036538399)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	doc. dr. sc. Adrian Satja Kurdija

Zadatak: **Pronalaženje grupa, sličnih entiteta i čestih podskupova u analizi podataka**

Opis zadatka:

Motivirati i opisati algoritme za analizu podataka s naglaskom na pronalaženje sličnih entiteta, pronalaženje čestih podskupova i grupiranje. Oblikovati, programski ostvariti i ispitati sustav za pronalaženje sličnih entiteta i čestih podskupova. Oblikovati, programski ostvariti i eksperimentalno usporediti različite algoritme grupiranja podataka. Eksperimente provesti na barem dvama skupovima podataka iz stvarnog svijeta, opisati i objasniti dobivene rezultate te izvesti zaključke. Uz rad je potrebno predati i dokumentirati izvorni kod, korištene skupove podataka te navesti korištenu literaturu i primljenu pomoć.

Rok za predaju rada: 14. lipnja 2024.

Zahvala

Zahvaljujem svojem mentoru doc. dr. sc. Adrianu Satji Kurdiji na stručnoj pomoći, podršci i korisnim savjetima tijekom izrade ovog završnog rada.

Sadržaj

Uvod

Strojno učenje je vrlo popularno u današnjici, ali je potrebno ispravno analizirati i prilagoditi podatke kako bi ono imalo smisla. Količina podataka koja se generira svaki dan je vrlo velika, ali uz pomoć različitih pristupa ih možemo prenamijeniti za korisnu svrhu. Postoje tri vrste strojnog učenja: nadzirano učenje, nenadzirano učenje i podržano učenje. Vrsta čiju podlogu ćemo proučavati u ovom radu je nenadzirano učenje koje se temelji na podacima koji su dani bez ciljne vrijednosti. Kako bi takvi podaci bili korisni unutar podataka treba ispravno pronaći pravilnosti.

Prvi smjer na koji ćemo se osvrnuti jest pronalaženje sličnih entiteta. Cilj ovog smjera nam je bio identificirati elemente koji imaju visoku sličnost temeljenu na parametrima i prikazati ih kao zasebne primjerke.

Drugi smjer koji smo proučavali je grupiranje (engl. *Clustering*). Ideja jest grupirati podatke u skupine na način da su elementi unutar iste skupine sličniji međusobno nego elementima iz drugih skupina.

Treći i posljednji smjer koji smo obradili je pronalaženje čestih podskupova (engl. *Frequent itemsets*). Traženje čestih podskupova je vrlo korisno unutar velikog skupa podataka jer možemo prikazati poveznicu između elemenata i predviđati buduće parove ili skupove elemenata.

Navedena tri smjera će se detaljno opisati u poglavljima rada i pozadina njihovog funkcioniranja.

Ove metode imaju široku primjenu u raznim područjima, uključujući poslovanje, medicinu, sociologiju i financije. Ovim radom, prikazat će se kako algoritmi mogu pomoći u razumijevanju složenih skupova podataka, te će se pružiti uvid u njihove primjene i učinkovitost u stvarnim scenarijima.

1. Pronalaženje sličnih entiteta

1.1. Općeniti opis pronalaženja sličnih entiteta

Temeljni problem rudarenja i analize podataka jest pronalaženje sličnih entiteta. Postupak pronalaska se temelji na identifikaciji elemenata unutar skupa koji međusobno dijele pojedine karakteristike ili atribute.

Konkretna primjena jest klasifikacija i regresija u područjima poput prepoznavanja obrazaca, prepoznavanja rukopisa i filtriranja neželjene pošte. Također se primjenjuje u sustavima za preporuke, gdje je glavna uloga algoritama prepoznavanje interesa korisnika.

1.2. Opis korištenog algoritma

Popularan algoritam u nadziranom i nenadziranom strojnom učenju jest k-NN. Navedeni algoritam se temelji na pronalasku najbližih susjeda određenog entiteta na temelju njegovih atributa. Odabir optimalne vrijednosti broja susjeda ključan je korak, jer prevelik broj može dovesti do pretjerane generalizacije, dok premalen može dovesti do prevelike osjetljivosti na šum u podacima. Prilikom primjene u nenadziranom učenju, može se koristiti za probleme grupiranja ili pronalaženje anomalija. Prilikom grupiranja algoritam može pomoći pri definiranju sličnih atributa, čime se formiraju skupine unutar podataka. Prije pseudokoda razjasnit ćemo osnovne pojmove:

- Učitavanje i prilagodba podataka – iz tablice odabiremo stupce koji nas zanimaju i ako postoje neodređene vrijednosti unutar stupca zamjenjujemo ih srednjom vrijednosti tog stupca. Provodimo standardizaciju podataka kako bi svi stupci bili na istoj skali (srednja vrijednost 0, standardna devijacija 1) što je važno za algoritme strojnog učenja kako bi ih ubrzali.
- Vrijednost k – predstavlja broj susjeda od svake točke
- Točka u podatcima ili na grafu predstavlja jedan vektor koji sadrži standardizirane numeričke podatke značajki, svaki vektor predstavlja točno jedan redak iz početne tablice podataka.
- Euklidska udaljenost – mjera udaljenosti detaljno opisana ispod pseudokoda

1.2.1. Pseudokod za algoritam k-NN:

```
1. Učitajte podatke i prilagodite za analizu
2. Odaberite vrijednost k
3. Za svaku točku u testnim podatcima:
   - pronađite euklidsku udaljenost od svih točaka u podacima
   - pohranite euklidske udaljenosti u listu i sortirajte ih
   - odaberite prvih k točaka
   - dodijelite testnu točku u skup s najbližim međusobnim udaljenostima
iz sortirane liste
4. Kraj
```

Najbitniji korak jest izračun udaljenosti između elemenata, te postoje različite opcije:

- Euklidska udaljenost
  - $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ (1)
  - Mjera udaljenosti u ravnom prostoru, koristi se kod pronalaženja sličnosti u računalnoj grafici i analizi podataka.
  - Ovu mjeru udaljenosti koristim u svojem programu.
- Manhattan udaljenost
  - $d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$ (2)
  - Mjeri udaljenost uzduž pravokutnih putanja, koristi se pri pomicanju u mreži pravokutnika.
- Jaccard udaljenost
  - $d(x,y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$ (3)
  - Mjerenje razlike između dvaju skupova, često se koristi u klasifikaciji teksta.
- Kosinusna udaljenost
  - $d(x,y) = 1 - \frac{x \cdot y}{\|x\|\|y\|}$ (4)

- Mjeri kut između dvaju vektora u prostoru, često korištena u analizi tekstova i sustavima za preporuke.

Glavni nedostatak k-NN algoritma je kompleksnost računanja na velikim skupovima podataka za svaki par entiteta. Kako bi poboljšali učinkovitost, postoje različite tehnike uz pomoć kojih možemo eliminirati neke nepotrebne izračune i skratiti vrijeme izvođenja. Prilikom izvedbe u Pythonu, uz *brute force* način obrade, postoje još dvije opcije za ubrzanje:

- K-D Tree
  - Primjena binarnog stabla za organizaciju točaka u k-dimenzionalnom prostoru. Rekurzivnom podjelom prostora prema dimenzijama se kreira struktura unutar koje je moguća učinkovita pretraga najbližih susjeda.
- Ball Tree
  - Primjenom hijerarhije kugli obuhvaćaju se točke koje međusobno dijele sličnosti, te se rekurzivno dijele na manje kugle sve do krajnjih listova. Ova tehnika je učinkovita za visokodimenzionalne skupove podataka kojima su rezultati neispravni oblici.

1.3. Alternativni algoritmi

Naravno postoje drugi algoritmi i metode koji su korisni za pronalazak sličnih entiteta, navest ću tri najpoznatije metode. Ovi algoritmi mogu biti vrlo korisni u različitim slučajevima pronalaženja sličnih entiteta unutar skupa podataka, no nisu bili primjenjivi kod mojih zadataka.

1.3.1. Locality-Sensitive Hashing

Algoritam Locality-Sensitive Hashing (LSH) koristi hash funkcije za grupiranje sličnih podataka te se koristi kada je brzina pretrage kritična, ali se time žrtvuje točnost.

1.3.2. Principal Component Analysis

Principal Component Analysis (PCA) je tehnika koja podatke transformira u manju dimenziju, a zadržava što veću točnost podataka. Često se koristi kao priprema podataka za optimalniju primjenu k-NN.

1.3.3. Self-Organising Map

Self-Organising Map (SOM) je vrsta neuronske mreže koja koristi nenadzirano učenje, te mapira višedimenzionalne podatke na nižu dimenziju poput 2D. Svrha smanjenja dimenzije jest lakša vizualizacija i grupiranje podataka.

2. Pronalaženje grupa

2.1. Općeniti opis grupiranja

Grupiranje ili *clustering* je tehnika nenadziranog učenja koja se koristi za organizaciju skupa podataka u podskupove tako da su podatci unutar iste grupe slični jedni drugima, dok su slabo povezani s podatcima iz ostalih grupa. Cilj je prepoznati strukturu unutar podataka i pronaći obrasce među podatcima koji nemaju ciljnu vrijednost.

Najbitnija primjena grupiranja je segmentaciji tržišta, gdje pomaže u identificiranju različitih grupa kupaca sličnih karakteristika za ciljanje kampanja. Također se primjenjuje u analizi bioloških podataka, kao što je grupiranje gena sličnih izraza za bolje razumijevanje bolesti.

2.2. Strategije grupiranja

Strategije grupiranja možemo podijeliti na dvije glavne skupine koje slijede temeljno različite puteve, algoritmi koji prema podatcima određuju broj grupa i algoritmi koji moraju imati zadani broj grupa.

2.2.1. Hijerarhijsko grupiranje

Hijerarhijsko grupiranje je metoda koja formira grupe kroz hijerarhijsku strukturu. Postoje dva glavna pristupa: aglomerativno i razdvajajuće. Aglomerativno grupiranje započinje sa svakom točkom kao zasebnom grupom, a zatim iterativno spaja najbliže grupe prema određenoj mjeri udaljenosti dok svi podaci ne budu u jednoj grupi. Razdvajajuće grupiranje radi suprotno, započinjući s jednom grupom koja sadrži sve točke i postupno ih dijeli dok svaka točka ne postane zasebna grupa. Ključna prednost ove strategije je što ne zahtijeva unaprijed određen broja grupa, za razliku od grupiranja s dodjelom točaka.

2.2.2. Grupiranje s dodjelom točaka

Grupiranje s dodjelom točaka je strategija koja se temelji na tome da se svaka točka u skupu podataka dodijeli u podskup na temelju nekog kriterija sličnosti. Broj grupa je unaprijed određen i ova strategija obično uključuje postupno dodjeljivanje svake točke u grupu u koju

najbolje odgovara. To može uključivati iterativni postupak dodjeljivanja točaka grupama na temelju neke mjere udaljenosti, kao što je udaljenost od središta grupe.

2.3. Opis korištenog algoritma

k-Means je popularan i jednostavan algoritam za grupiranje podataka koji spada u strategiju grupiranja s dodjelom točaka. Cilj k-Means algoritma je podijeliti skup podataka u k grupa tako da svaki podatak pripada grupi s najbližim srednjim vrijednostima (centroidima). Algoritam se temelji na iteracijama, što znači da se smanjuje zbroj kvadrata udaljenosti između podataka i njihovih centroida.

Prije pseudokoda razjasnit ćemo osnovne pojmove:

- k – broj grupa na koji ćemo razdijeliti podatke
- učitavanje podataka – isto kao i za prethodni algoritam moramo provesti standardizaciju podataka za svaki stupac kako bismo doveli sve podatke na istu skalu i izbjegli pristranosti za različite mjere.
- centroid – središte grupe, srednja vrijednost svih elemenata grupe
- Točka u podatcima ili na grafu (uključujuću centroide) predstavlja jedan vektor koji sadrži standardizirane numeričke podatke značajki, svaki vektor predstavlja točno jedan redak iz početne tablice podataka.

2.3.1. Pseudokod za algoritam k-Means

1. Odaberite k i učitajte podatke
2. Postavite centroide (c_1, c_2,...c_k) nasumično
3. Ponavljajte korake 4 i 5 dok ne dođe do konvergencije ili do kraja zadanog broja iteracija
4. Za svaku točku x_i :
   - pronađite najbliži centroid (c_1, c_2,...c_k)
   - dodijelite točku toj grupi
5. Za svaku grupu j = 1 .. k:
   - novi centroid = prosjek svih točaka dodijeljenih toj grupi
6. Kraj

Tijekom ovih koraka algoritam konvergira prema optimalnim grupama, smanjujući razliku unutar grupa. Na kraju, dobivene grupe predstavljaju raspodjelu podataka prema njihovim prirodnim grupama ili strukturama.

K-Means algoritam je osjetljiv na početno nasumično odabran broj grupa, što može dovesti do različitih rezultata prilikom svakog izvođenja algoritma. Zato ćemo koristiti Elbow metodu uz pomoć koje dolazimo do optimalnog broja grupa za određeni skup podataka.

2.4. Opis metoda

Elbow metoda je jednostavna tehnika koja se koristi za određivanje optimalnog broja grupa *k* u k-Means algoritmu. Ova metoda se temelji na smanjenju varijance unutar svake grupe dok ne dođemo do kraja zadanog raspona. Ideja je pronaći „lakat“ na grafu koji prikazuje odnos između broja grupa i varijance unutar te grupe. „Lakat“ je prijelomna točka na grafu gdje dodavanje dodatnih grupa više ne donosi značajno smanjenje varijance, čime možemo zaključiti da smo došli do optimalnog broja grupa.

Postupak izvođenja koraka Elbow metode:

- Pokretanje k-Means algoritma za različite vrijednosti *k* unutar nekog raspona
- Izračunavanje varijance unutar grupa za svaki *k*
- Crtanje grafa s izračunatim varijancama i brojem grupa *k*
- Prepoznavanje točke „lakta“ na grafu i odabir te vrijednosti kao optimalnu za *k*

Ova metoda pruža jednostavan način odabira optimalnog broja grupa u k-Means algoritmu, olakšavajući korisnicima da izbjegnu prekomjerno ili nedovoljno grupiranje pri analizi podataka.

3. Pronalaženje čestih podskupova

3.1. Općeniti opis pronalaženja čestih podskupova

Pronalaženje čestih podskupova je ključna aktivnost u analizi pravilnosti podataka. Ova tehnika istražuje predmete koji se često pojavljuju zajedno unutar skupa podataka. Glavna prednost su pravila asocijacije koja nam govore kako su neki skupovi međusobno povezani.

Primjene su raznolike i obuhvaćaju područja kao što su analiza tržišta, preporuke proizvoda i medicinska dijagnostika. Ovakav pristup pomaže u otkrivanju uzoraka, što može optimizirati rad u poslovnom i medicinskom okruženju. Jednostavan primjer je lokalna trgovina, pronalaženjem čestih podskupova vlasnik može otkriti koje stavke se često kupuju zajedno, što omogućuje bolje upravljanje zalihama i povećanje prodaje.

3.2. Opis korištenog algoritma

Algoritam koji sam koristio za pronalazak čestih podskupova, pod nazivom Apriori, je jedan od najpoznatijih i najučinkovitijih u području analize podataka za ovaj slučaj. Naziv algoritma potječe iz latinskog i znači „ono što dolazi prije“, čime zaključujemo da se koristi za pretpostavke uz pomoć kojih možemo doći do konkretnih odluka koje mogu biti korisne za naše poslovanje ili primjenu u svakodnevnom životu.

Prije opisa pseudokoda definirat ćemo neke osnovne pojmove:

- izgled pravila asocijacije:

$$A \rightarrow \mathrm{B}$$

  - A – predstavlja antecedent (prethodnik, uzrok)
  - B – predstavlja consequent (posljedica)
  - Primjer: Ako je netko kupio jabuke (A), tada će kupiti i naranče (B).
- minimalna podrška – predstavlja minimalno pojavljivanje uzroka A i posljedice B unutar skupa, dodatno objašnjenje pojma podrška navedeno je ispod (Support).
- transakcija – predstavlja skupove elemenata ili stavki koje su zajedno prisutne u svakom retku podataka. Ovaj pristup je čest u analizi podataka kod kupovine, gdje svaka transakcija predstavlja kupnju koja sadrži više stavki.

- učitavanje skupa podataka – svaki redak pohranjujemo u obliku transakcije, zbog transformacije u binarni format. Transformiranje podataka u binarni format omogućuje da u redcima matrice pohranjujemo transakcije, a u stupcima definiramo značajke. Zatim za svako polje matrice postavljamo broj 1 ili 0. Ako je određena značajka prisutna u transakciji postavljamo broj 1, a ako nije postavljamo broj 0. Na ovaj način smo postigli vrlo jednostavno korištenje skupa podataka za algoritam Apriori jer lako može iščitati koje značajke su prisutne ili ne za svaku transakciju.
- jedinstvene stavke – one stavke koje se nalaze unutar transakcija ali ih bilježimo samo jednom, npr.

  ['jabuka', 'banana']
  ['banana', 'naranča']
  ['jabuka', 'naranča', 'grožđe']
  ['banana', 'grožđe']

  Jedinstvene stavke ovih transakcija su: {'jabuka', 'banana', 'naranča', 'grožđe'}
- kandidati – Nakon pronalaska jedinstvenih stavki biramo elemente te liste. Na početku su kandidati jednoelementni, ali kasnije kreacijom čestih podskupova broj elemenata unutar kandidata raste.
- česti podskupovi – predstavljaju kombinaciju prethodno navedenih kandidata koji se dovoljno puta pojavljuju zajedno, npr. {'jabuka', 'banana'}, {'mlijeko', 'naranča'}, {'{'jabuka', 'banana'}', {'naranča'}}.
- generiranje pravila asocijacije – nakon što smo pronašli česte podskupove iz njih možemo točno izvesti pravila, npr. {'jabuka', 'banana'} i {'mlijeko', 'naranča'}.

  {'jabuka', 'banana'} i {'mlijeko', 'naranča'}, iz ovoga slijedi:

  Ako netko kupi jabuku, onda će vjerojatno kupiti bananu. (jabuka – > banana)
  Ako netko kupi mlijeko, onda će vjerojatno kupiti naranču. (mlijeko – > naranča)

- minimalna sigurnost – predstavlja granicu kroz koju propuštamo pravila uzimajući u obzir njihovu vjerodostojnost. Pojam sigurnosti je detaljno opisan u poglavlju ispod pseudokoda (Confidence).

3.2.1. Pseudokod za algoritam Apriori

```
1. Definirajte minimalnu podršku (min_support) i učitajte skup podataka

2. Identificirajte sve jedinstvene stavke u transakcijama skupa

3. Generirajte početne kandidate i izračunajte njihovu podršku unutar
transakcija
```

4. Zadržite samo one kandidate čija je podrška veća od minimalne

5. Ponavljajte korake 6-8 dok ne pronađete sve česte skupove ili dok ne možete generirati nove kandidate

6. Za svaki novi skup kandidata:

   - generirajte kandidate (kombinacije prethodnih čestih skupova)
   - izračunajte podršku za nove kandidate

7. Zadržite samo one kandidate čija je podrška veća od minimalne

8. Dodajte nove česte skupove u konačni skup

9. Generirajte pravila asocijacije iz konačnog skupa

10. Filtrirajte pravila na temelju minimalne sigurnosti

11. Kraj

3.3. Ključni pojmovi Apriori algoritma

3.3.1. Antecedent

Antecedent (prethodnici) su elementi ili skupovi elemenata koje pronalazimo na lijevoj strani pravila asocijacije, te predstavljaju uvjet koji se mora zadovoljiti kako vi se ostvarila desna strana pravila (posljedica).

3.3.2. Consequent

Consequent (posljedica) predstavlja element ili skup elemenata koji se nalaze na desnoj strani pravila asocijacije i predstavljaju rezultat ako se prethodnik pojavio.

3.3.3. Support

Support (podrška) je pojam koji opisuje koliko puta se pojavljuju elementi iz pravila u cijelom skupu podataka. Takav omjer je vrlo koristan u pronalasku čestih podskupova jer uz pomoć minimalne granice možemo eliminirati pravila koja se rijetko pojavljuju i nisu korisna. Izgled formule za računanje podrške:

$$support(A) = \frac{A}{skup\ podataka}$$

3.3.4. Confidence

Confidence (sigurnost) opisuje kolika je vjerojatnost da će posljedica (B) biti ostvarena ako se dogodio uzrok (A). Sigurnost računamo kao omjer dvije podrške, u brojnik stavljamo podršku od uzroka i posljedice, a u nazivnik podršku samo uzroka. Ako je omjer bliži broju jedan možemo zaključiti da je povezanost između uzorka i posljedice visoka.

Izgled formule za računanje sigurnosti:

$$confidence(A \rightarrow B) = \frac{support(A \cup B)}{support(A)}$$

3.3.5. Lift

Lift (podizanje) opisuje povezanost između uzroka i posljedice ali u oba smjera, negativnom i pozitivnom. Ako je omjer veći od jedan tada je povezanost pozitivna, a ako je omjer manji od jedan tada ukazuje na negativnu povezanost. Omjer sadrži sigurnost pravila u brojniku i podršku posljedice u nazivniku čime pokazuje kolika je vjerojatnost pojavljivanja posljedice (B), ako se pojavio uzrok (A) naspram kolika je pojavljivanje posljedice (B) unutar cijelog skupa podataka.

Izgled formule:

$$lift(A \rightarrow B) = \frac{confidence(A \rightarrow B)}{support(B)}$$

4. Eksperimenti

4.1. Opis skupova podataka

Skupove podataka koje sam analizirao su preuzeti s kagglea, a njihove teme su prodaja automobila u 2022. godini (statistika za SAD) i statistika NBA igrača tijekom povijesti. Nakon analize tih skupova sam izabrao još jedan, specifično za traženje čestih podskupova kod kupovanja namirnica u dućanu.

4.1.1. Opis skupa podataka vezanog za automobile

Tablica se sastoji od 16 stupaca i otprilike 24000 redaka jedinstvenih vrijednosti. Ovakva tablica je vrlo korisna za analizu tržišta, uočavanje obrazaca, stvaranje predviđanja i optimizaciju sustava prodaje.

Zaglavlje skupa podataka vezanog za automobile Car_Sales.csv se sastoji od 16 parametara koji su:

- Car_id – jedinstveni identifikacijski broj automobila
- Date – datum prodaje
- Customer Name – ime i prezime kupca
- Gender – spol kupca
- Annual Income – godišnji prihod kupca
- Dealer_Name – naziv autokuće
- Company – proizvođač automobila
- Model – model automobila
- Engine – vrsta motora
- Transmission – vrsta mjenjača
- Color – boja automobila
- Price – cijena automobila
- Dealer_No – identifikacijski broj prodavača
- Body Style – tip šasije
- Phone – telefonski broj kupca
- Dealer_Region – područje na kojoj se nalazi autokuća

4.1.2. Opis skupa podataka vezanog za košarkaše

Tablica ovog skupa podataka se sastoji od 35 stupaca i otprilike 32000 redaka jedinstvenih vrijednosti statistike NBA i ABA igrača od 1947. godine do danas. Stranica na kaggleu s koje sam preuzeo skup podataka koji mi se činio najzanimljiviji Player_Totals.csv, sadrži još mnogobrojne zanimljive .csv datoteke koje su klasificirane za uspješnost ekipa, rezultate utakmica, sezone, igrače i nagrade. Ovakav tip podataka je vrlo zanimljiv za analizu i pronalazak čestih podskupova.

Zaglavlje skupa podataka Player_Totals.csv se sastoji od 35 parametara koji su:

- seas_id i season – identifikator sezone za svakog igrača i godina sezone
- player_id i player – identifikator igrača i njegovo puno ime
- birth_year – godina rođenja
- pos – pozicija na terenu
- age i experience – starost igrača i broj sezona u NBA/ABA
- lg i tm – liga u kojoj igrač igra i tim za koji igra
- g i gs – broj utakmica odigranih u sezoni i broj započetih utakmica
- mp – ukupan broj odigranih minuta
- fg, fga i fg_percent – broj ubačenih šuteva, broj pokušaja i uspješnost šuta
- x3p, x3pa, x3p_percent – broj pogođenih trica, broj pokušanih i uspješnost
- x2p, x2pa, x2p_percent – broj pogođenih dvica, broj pokušanih i uspješnost
- e_fg_percent – efikasna uspješnost šuta
- ft, fta i ft_percent – broj pogođenih slobodnih bacanja, broj pokušanih uspješnost
- orb, drb i trb – broj napadačkih, obrambenih i ukupni broj skokova
- ast – broj asistencija
- stl – broj ukradenih lopti
- blk – broj blokiranih šuteva
- tov – broj izgubljenih lopti
- pf – ukupan broj prekršaja
- pts – ukupan broj poena

4.1.3. Opis skupa podataka vezanog za namirnice

Skup podataka s namirnicama je jednostavan skup, koji sadrži otprilike 10000 redaka i 32 stupca. Svaki redak predstavlja jednu kupljenu košaricu, a svaki stupac jedan proizvod. Namjerno sam izabrao ovaj skup podataka jer vrlo dobro pokazuje ovisnost pojedinačnih i skupa namirnica.

Zaglavlje skupa podataka Grocery_Products_Purchase.csv:

- Product 1 - Product 32 – prikazuje košarice s maksimalnih 32 proizvoda

4.2. Rezultati k-NN

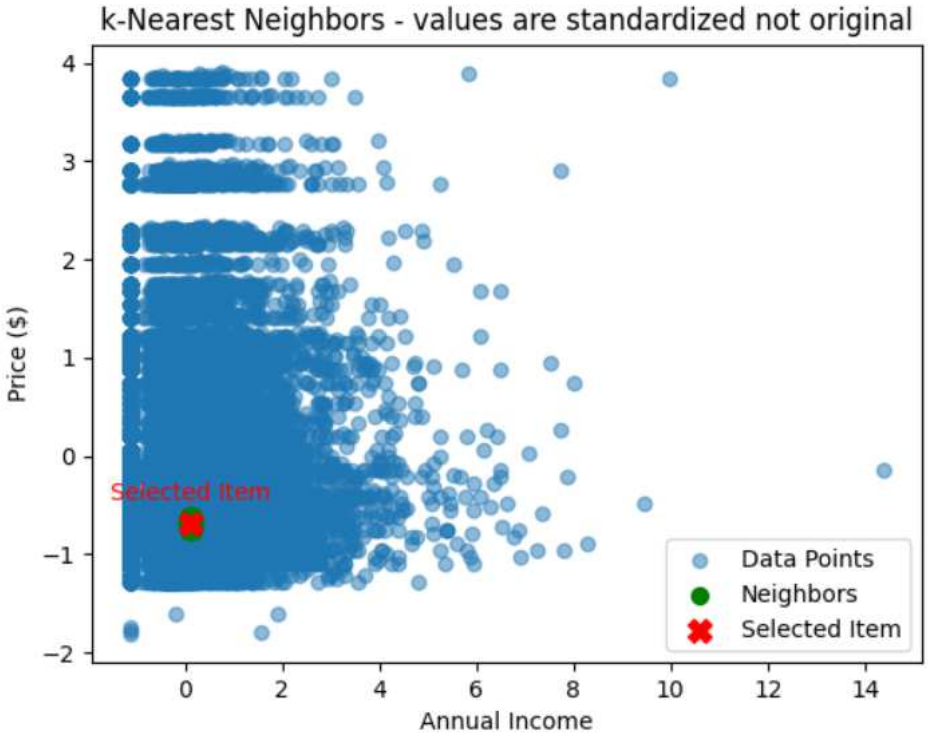
Cilj korištenja k-NN algoritma na skupu podataka automobila jest proučiti gustoću susjeda nasumično odabrane prodaje automobila kako bi se identificirala područja s visokom koncentracijom prodaje ili nepokrivena tržišta. Očekivani rezultat je jednostavno uočavanje područja s visokom potražnjom za automobilima u određenom cjenovnom rangu, što može biti korisno za planiranje strategija, lokaciju prodajnih mjesta ili razvoj novih proizvoda u automobilskoj industriji.

Cilj primjene algoritma na skupu podataka košarkaša jest za veći broj susjeda (100-200) pronaći u kojoj statistici šutiranje možemo pronaći najveću razliku. Očekivanje jest da će se najveća razlika pojavljivati u slučaju gađanja trica ili slobodnih bacanja. Ovim pristupom možemo pronaći igrače koji su podcijenjeni zbog generalizacije statistike pucanja na koš i pomoći trenerskom timu izabrati najbolje igrače za specifične situacije.

4.2.1. Rezultati susjeda kod prodaje automobila

Motivacija za pronalazak susjeda pri analizi podataka prodaje automobila jest uzeti u obzir cijenu proizvoda i godišnji prihod kupca i analizirati navedene podatke. Nakon vizualizacije na grafu i odabira par nasumičnih kupaca (Slika 4.1)(Slika 4.2) vrlo jasno možemo zapaziti različiti raspon gustoće susjeda. Analizom grafa uočavamo kako se gustoća točaka susjeda smanjuje što više raste cijena automobila ili raste godišnji prihod kupca. Time možemo zaključiti da su automobili u nižem i srednjem cjenovnom rangu daleko najprodavaniji kod svih slojeva društva, ali se gustoća postupno smanjuje zbog manjeg broja imućne populacije. Automobili u višem cjenovnom rangu se prodaju u manjem broju, ali se također prodaju uspješno kod kupaca niže kupovne moći što govori da prodavači u autokućama ne smiju imati predrasude kako bi bili uspješni trgovci.

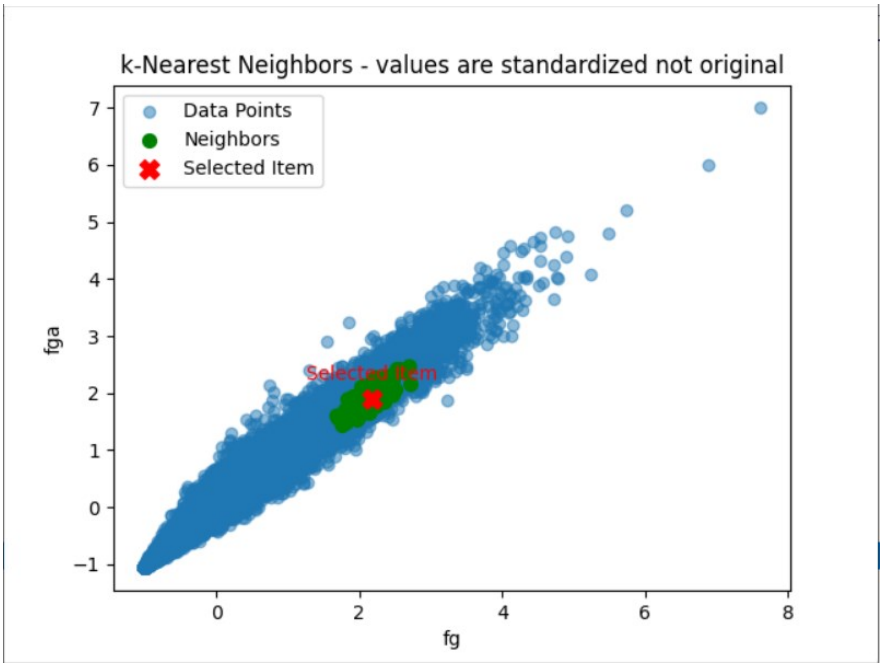
Slika 4.1 Prikaz susjeda nasumičnog kupca automobila



Slika 4.2 Prikaz susjeda drugog nasumičnog kupca automobila

4.2.2. Rezultati susjeda kod košarkaša

Kao primjer korištenja k-NN algoritma sam odabrao sve statistike za šutiranje košarkaša i odabrao prikaz 100 najbližih susjeda za svaki element. Tablični prikaz (Tablica 4.1) prikazuje podatke za susjede nasumično odabranog igrača, te možemo uočiti da su razlike između statistika vrlo niske osim za trice. Time potvrđujemo da je šut za tri poena najzahtjevnija opcija za pogoditi i stoga najteža stavka za pronaći pravilnost na velikom skupu podataka. Na grafičkom prikazu (Slika 4.3) se vrlo jasno uočava grupa susjeda naspram ostalih igrača unutar skupa podataka.



Slika 4.3 Grafički prikaz susjeda košarkaša

Tablica 4.1 Prikaz susjeda košarkaša (šutiranje)

fg	fga	fg percent	x3p	x3pa	x3p_percent	x2p	x2pa	x2p_percent	e _fg_percent	ft	fta	ft percent
591	1157	0.511	2.0	14.0	0.143	589	1143	0.515	0.512	317	403	0.787
599	1218	0.492	2.0	16.0	0.125	597	1202	0.497	0.493	310	378	0.82
598	1188	0.503	8.0	43.0	0.186	590	1145	0.515	0.507	300	396	0.758
584	1154	0.506	7.0	33.0	0.212	577	1121	0.515	0.509	298	377	0.79
630	1218	0.517	1.0	7.0	0.143	629	1211	0.519	0.518	279	359	0.777
619	1211	0.511	4.0	24.0	0.167	615	1187	0.518	0.513	273	350	0.78
613	1201	0.51	1.0	15.0	0.067	612	1186	0.516	0.511	298	392	0.76
604	1242	0.486	3.0	22.0	0.136	601	1220	0.493	0.488	284	361	0.787
591	1224	0.483	6.0	37.0	0.162	585	1187	0.493	0.485	292	363	0.804
602	1114	0.54	1.0	12.0	0.083	601	1102	0.545	0.541	324	402	0.806
583	1229	0.474	4.0	32.0	0.125	579	1197	0.484	0.476	321	396	0.811
611	1225	0.499	3.0	15.0	0.2	608	1210	0.502	0.5	291	396	0.735
644	1187	0.543	3.0	20.0	0.15	641	1167	0.549	0.544	339	428	0.792
610	1236	0.494	3.0	14.0	0.214	607	1222	0.497	0.495	338	442	0.765
530	1072	0.494	3.0	17.0	0.176	527	1055	0.5	0.496	350	440	0.795
561	1082	0.518	2.0	11.0	0.182	559	1071	0.522	0.519	271	374	0.725
517	1030	0.502	3.0	21.0	0.143	514	1009	0.509	0.503	328	398	0.824
546	1025	0.533	3.0	19.0	0.158	543	1006	0.54	0.534	341	410	0.832
540	1090	0.495	4.0	23.0	0.174	536	1067	0.502	0.497	364	462	0.788
578	1173	0.493	4.0	16.0	0.25	574	1157	0.496	0.494	302	398	0.759
569	1153	0.493	4.0	19.0	0.211	565	1134	0.498	0.495	364	428	0.85
598	1219	0.491	1.0	13.0	0.077	597	1206	0.495	0.491	281	340	0.826
627	1275	0.492	1.0	12.0	0.083	626	1263	0.496	0.492	294	403	0.73
605	1283	0.472	9.0	51.0	0.176	596	1232	0.484	0.475	329	423	0.778
616	1258	0.49	5.0	22.0	0.227	611	1236	0.494	0.492	339	409	0.829

4.3. Rezultati k-Means

Cilj korištenja k-Means algoritma na skupu podataka automobila jest identificirati karakteristične grupe automobila na temelju njihovih atributa kako bi se bolje razumjela struktura tržišta. Očekivani rezultat je jasna segmentacija automobila prema njihovim specifičnim karakteristikama poput cijene, što može pomoći u boljem ostvarenju prodaja.

Cilj primjene algoritma na skupu podataka košarkaša jest grupiranje igrača u kategorije na temelju njihovih statistika pucanja na koš. Očekivanje je da će kroz grupiranje igrača prema njihovim performansama u različitim aspektima šutiranja, treneri moći bolje prilagoditi taktiku igre, identificirati snage i slabosti tima i donositi odluke o rotaciji igrača na terenu.

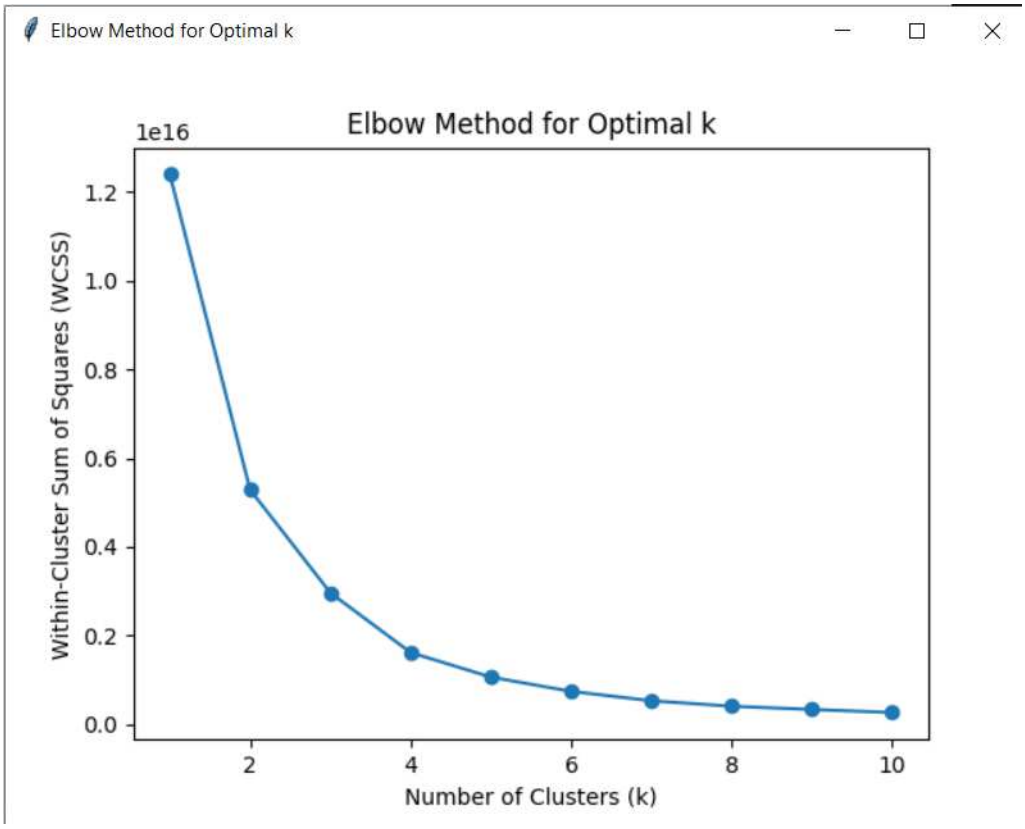
Za razliku od prethodnom algoritma u ovoj strategiji ne biramo neki element i grupu oko njega, već na temelju zadanog broja grupa određujemo pravilne skupove jednako udaljene od centroida koji sadrži srednju vrijednost te skupine.

4.3.1. Rezultati grupiranja kod prodaje automobila

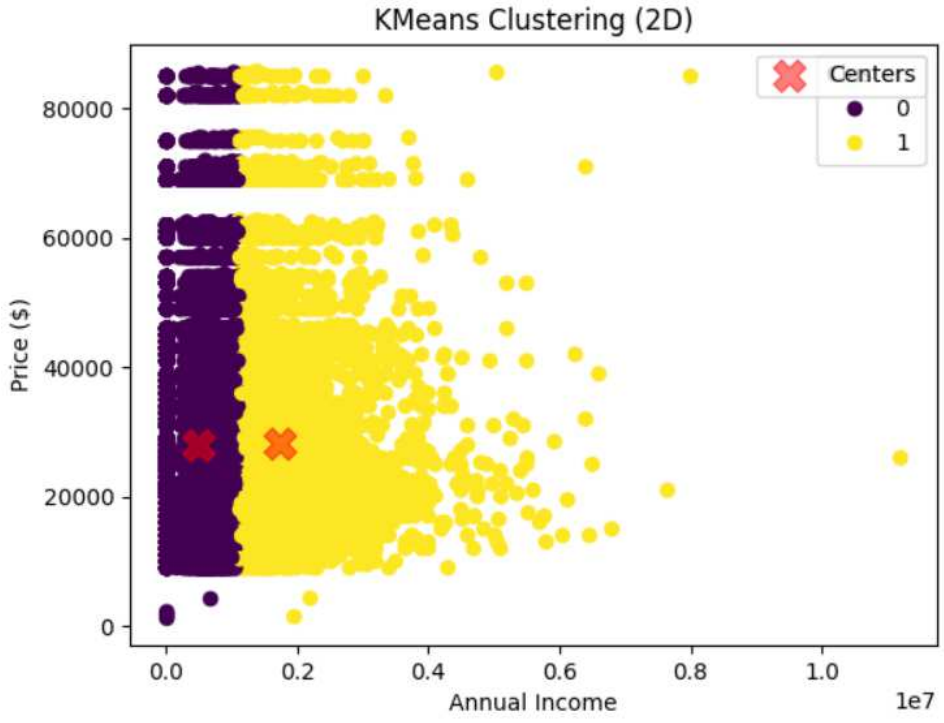
Grupiranje na skupu podataka automobila sm opet proveo na temelju kupovne moći i cijene proizvoda. Prvo sam proveo Elbow metodu kako bi odredio optimalan broj grupa na temelju parametara i skupa podataka i rezultat je bio između 2 i 3 ali smo zaokružili na nižu opciju zbog razlike vrijednosti (Slika 4.4). Zatim sam na temelju broja grupa proveo k-Means algoritam koji je očigledno prikazao razlike u grupama na temelju kupovne moći (Slika 4.5). Nakon analize grafa i koordinata središta (Tablica 4.2**Error! Reference source not found.**) zasebnih grupa očitavamo da su automobili u srednjem cjenovnom rangu jednako popularni iako je godišnja zarada druge grupe tri puta veća od prve. Govori nam da je idealna cijena vozila u SAD-u oko 30000 dolara što je vrlo koristan podatak za prodavače jer time znaju kojom brojkom mogu „namamiti“ kupce u oglasnicima i reklamama da posjete njihove autokuće.

Tablica 4.2 Koordinate središta grupa prodaje automobila

Cluster	Annual Income	Price($)
Cluster 1	499948.13	28068.83
Cluster 2	1729784.54	28148.43



Slika 4.4 Elbow metoda na skupu automobila



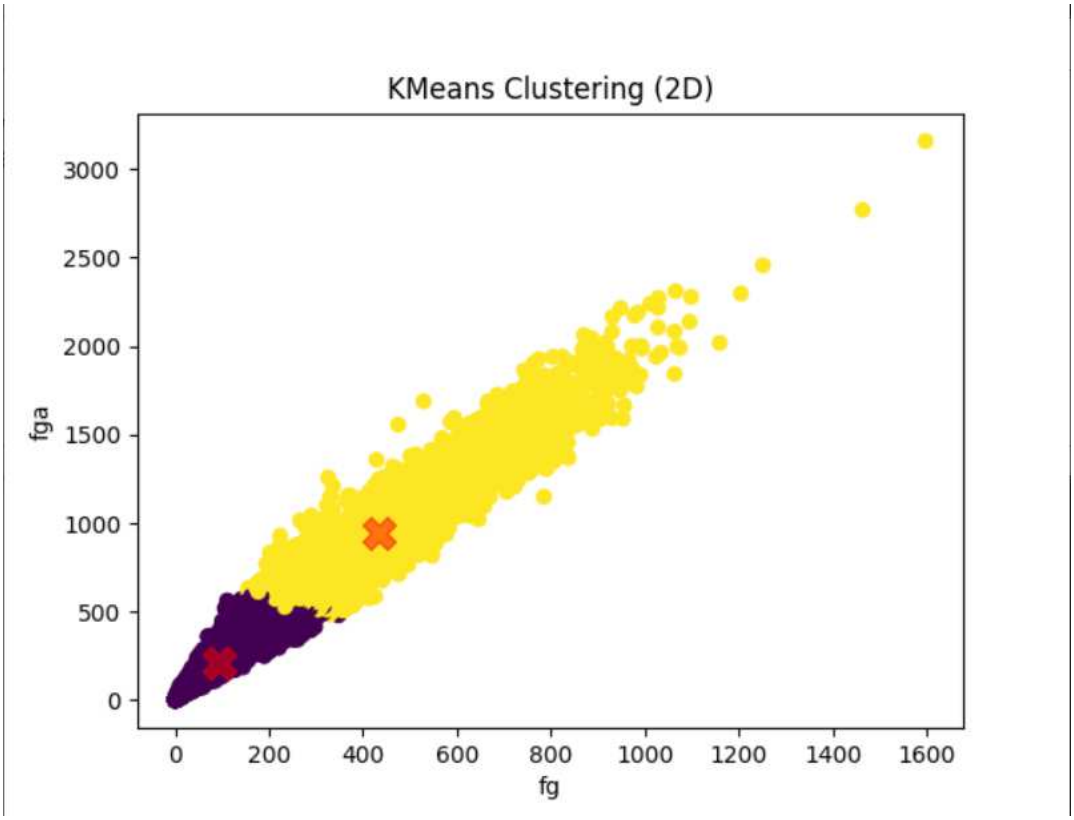
Slika 4.5 Prikaz podjele na grupe kupovne moći

4.3.2. Rezultati grupiranja kod košarkaša

Grupiranje na skupu podataka košarkaša sam opet primijenio na statistici šutiranja. Problem kod grafičkog prikaza jest odabir 13 parametara efikasnosti, a prikaz bi tada trebao biti u istom broju dimenzija. Stoga sam uzeo samo prvi par parametara za prikaz na 2D grafu (pokušaj prikaza za tri varijable je već vrlo nepregledan), i prikazao same grupe i njihova središta (Tablica 4.3). Postoje preklapanja grupa jer samim prikazom u dvije dimenzije nismo eliminirali utjecaj ostalih parametara na izgled grupe (Slika 4.6). Zaključak ovih rezultata ukazuje na važnost uzimanja u obzir više parametara prilikom grupiranja košarkaša radi preciznijeg razumijevanja njihovih stilova igre i potencijalnih doprinosa timu. Iako centar prve skupine ukazuje da zabijaju puno manje poena, njihova efikasnost je usporediva s igračima koji započinju utakmicu i igraju više minuta te stoga imaju više šuteva i poena.

Tablica 4.3 Prikaz koordinata središta grupa

fg	fga	fg_p	x3p	x3pa	x3p_p	x2p	x2pa	x2p_p	e_fg_p	ft	fta	ft_p
93.00	213.22	0.42	18.79	53.75	0.259	78.60	171.95	0.441	0.446	44.16	61.43	0.704
435.15	942.24	0.46	43.78	123.95	0.264	398.44	838.41	0.475	0.480	233.34	304.5	0.765



Slika 4.6 Grafički prikaz grupa košarkaša

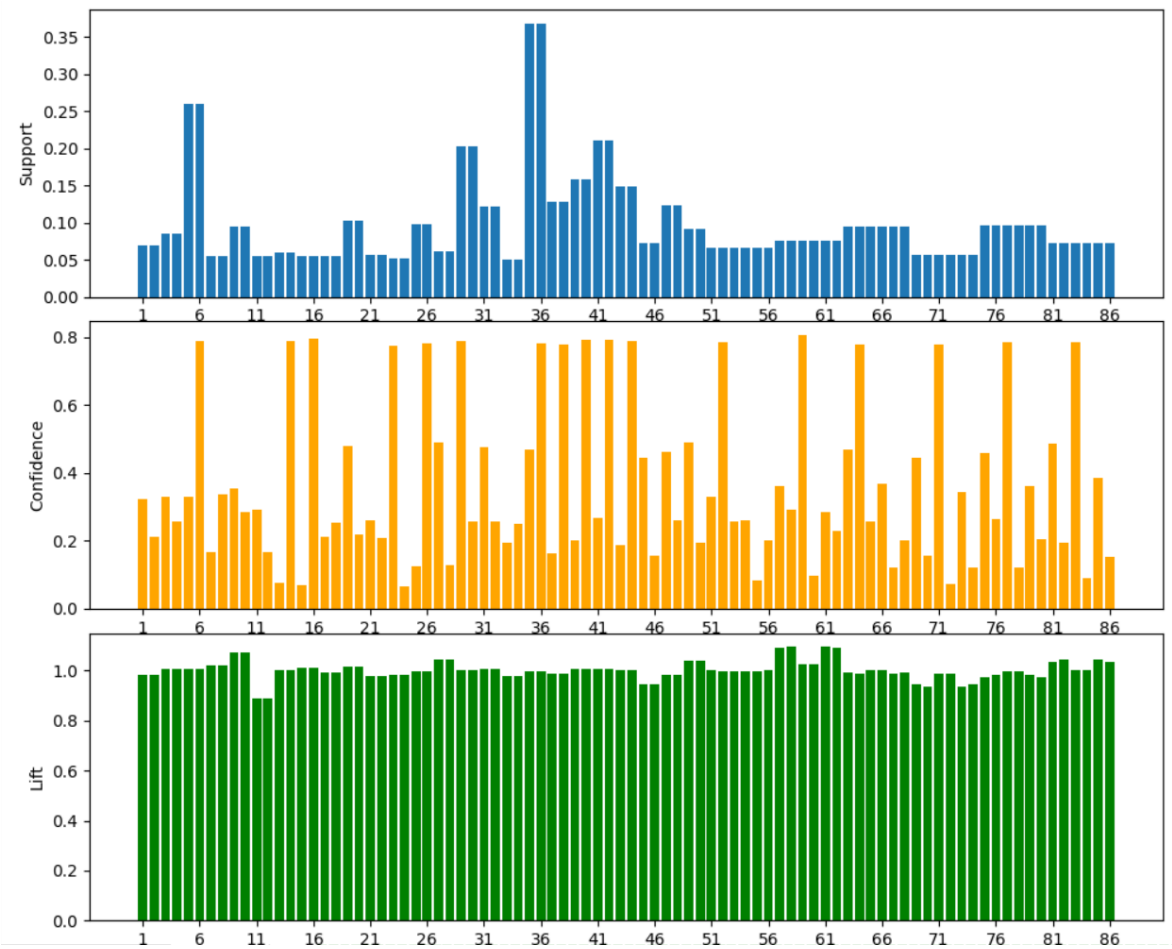
4.4. Rezultati Apriori

Cilj primjene algoritma Apriori na skupu podataka automobila jest prepoznati podskupove u obliku kombinacija karakteristika automobila. Očekivani rezultat je jasno uočavanje kombinacija specifičnih značajki vozila koje se često pojavljuju zajedno. Poznato je da su u SAD-u popularna vozila poput terenaca ili džipova stoga očekujem da će biti uključeni u rezultatu.

Skup podataka s kupljenim namirnicama je savršen slučaj za primjenu algoritma Apriori. Pronalaskom podskupova možemo pronaći presjek koji sadrži neophodne proizvode koje kupci najčešće biraju. Rezultati će najvjerojatnije sadržavati vodu, kruh, mlijeko, voće ili povrće jer iz mojeg osobnog iskustva su to proizvodi koji se pri skoro svakoj kupovini pronađu u košarici kupaca.

4.4.1. Rezultati čestih podskupova kod automobila

Analiza čestih podskupova kod skupa podataka prodaje automobila je vrlo korisna jer se mogu iščitati zanimljiva pravila asocijacije (Tablica 4.4). Takve ishode možemo iskoristiti u analizi tržišta u unaprijediti prodaju. Nakon iscrtavanja grafa za parametre (spol, model, kompanija, oblik šasije i boja) (Slika 4.7) najzanimljiviji podatci su da muškarci i žene biraju boje automobila u istom redoslijedu bijela, crna, crvena, ali su muškarci puno veći kupci u omjeru 3:1. Tri najpopularnije marke su Chevrolet, Dodge i Ford, a najzastupljeniji oblik šasije SUV, zatim blizu Hatchback i Sedan. Moj zaključak jest da je najbolja opcija prodavati vozila tipa šasije SUV u crnoj boji jer je to najpopularnija kombinacija kod muškaraca koji upravljaju većinom tržišta (Tablica 4.5).



Slika 4.7 Grafički prikaz mjera asocijacije

Tablica 4.4 Primjer nekih pravila asocijacije na tržištu

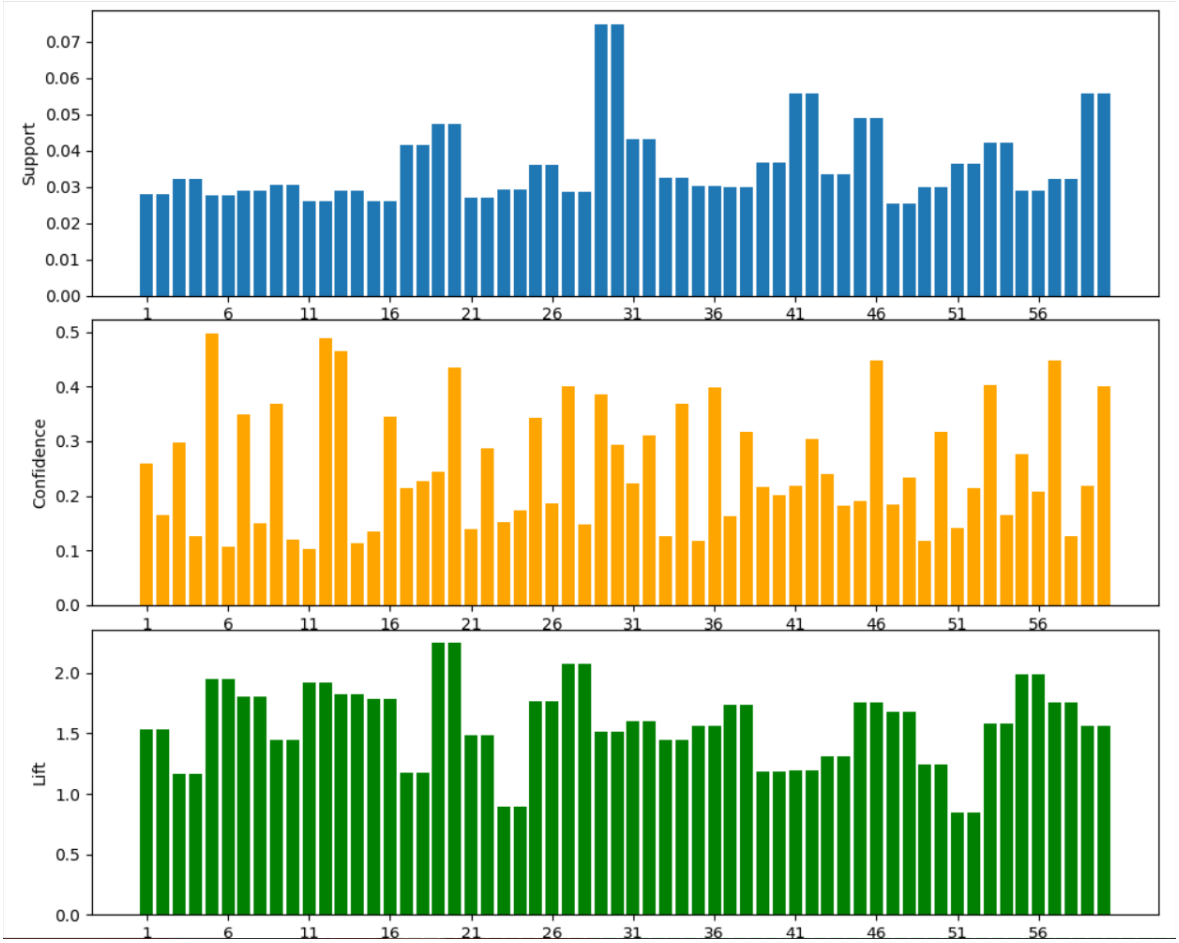
Rule	Antecedents	Consequence	Support	Confidence	Lift
Rule 23	other vegetables	soda	0.0293	0.1513	0.8945
Rule 24	soda	other vegetables	0.0293	0.1731	0.8945
Rule 25	tropical fruit	other vegetables	0.0359	0.3421	1.7678
Rule 26	other vegetables	tropical fruit	0.0359	0.1855	1.7678
Rule 27	whipped/sour crear	other vegetables	0.0287	0.4011	2.0731
Rule 28	other vegetables	whipped/sour cream	0.0287	0.1482	2.0731
Rule 29	other vegetables	whole milk	0.0748	0.3868	1.5136
Rule 30	whole milk	other vegetables	0.0748	0.2929	1.5136
Rule 31	other vegetables	yogurt	0.0431	0.2228	1.6007
Rule 32	yogurt	other vegetables	0.0431	0.3097	1.6007
Rule 33	whole milk	pastry	0.0324	0.1269	1.445
Rule 34	pastry	whole milk	0.0324	0.3692	1.445
Rule 35	whole milk	pip fruit	0.0301	0.1178	1.557
Rule 36	pip fruit	whole milk	0.0301	0.3978	1.557
Rule 37	rolls/buns	sausage	0.0298	0.163	1.7355
Rule 38	sausage	rolls/buns	0.0298	0.3171	1.7355
Rule 39	soda	rolls/buns	0.0367	0.2169	1.1874
Rule 40	rolls/buns	soda	0.0367	0.2009	1.1874
Rule 41	whole milk	rolls/buns	0.0556	0.2177	1.1913
Rule 42	rolls/buns	whole milk	0.0556	0.3044	1.1913
Rule 43	yogurt	rolls/buns	0.0334	0.2396	1.3113
Rule 44	rolls/buns	yogurt	0.0334	0.1825	1.3113
Rule 45	whole milk	root vegetables	0.0489	0.1914	1.756

Tablica 4.5 Prikaz najboljih ponuda za muškarce

Rule	Antecedens	Consequence	Support	Confidence	Lift
Rule 59	SUV, Black	Male	0.0758	0.8056	1.0245
Rule 16	Dodge	Male	0.0555	0.7941	1.0099
Rule 42	SUV	Male	0.2108	0.7906	1.0054
Rule 40	Red	Male	0.1584	0.7899	10.045
Rule 6	Black	Male	0.2595	0.7896	10.042
Rule 44	Sedan	Male	0.1482	0.7894	1.004
Rule 29	Hatchback	Male	0.202	0.7882	1.0024
Rule 14	Chevrolet	Male	0.0599	0.7867	10.005
Rule 83	Sedan, Pale White	Male	0.0722	0.7862	0.9999
Rule 52	Hatchback, Black	Male	0.0665	0.7855	0.9989
Rule 77	SUV, Pale White	Male	0.0966	0.7841	0.9972
Rule 36	Pale White	Male	0.3684	0.7825	0.9951
Rule 26	Hardtop	Male	0.0972	0.7822	0.9948
Rule 71	Passenger, Pale White	Male	0.057	0.7774	0.9886
Rule 64	Hatchback, Pale White	Male	0.0945	0.7773	0.9885
Rule 38	Passenger	Male	0.1281	0.7762	0.9871

4.4.2. Rezultati čestih podskupova kod namirnica

Ovaj skup podataka je obrađen samo u ovom poglavlju jer je posebno koristan u svakodnevnom životu i jer ga uz pomoć Apriori algoritma možemo vrlo jednostavno prikazati. Odabrao sam opciju da kupac odabire prosječno 12 proizvoda i proučio tablicu kako bi doznao koje proizvode ljudi najčešće kupuju zajedno. Grafički rezultati (Slika 4.8) ukazuju na pravila gdje je sigurnost najviša i da su najpopularniji proizvodi povrće, sir, vrhnje, jaja, voda, voće ili maslac u kombinaciji s mlijekom (pravila pod zadanim brojem iz tablice) (Tablica 4.6) Time se lako uočava da je mlijeko najpopularniji proizvod pri odlasku u dućan, jer u 30-40% slučajeva kupac će ga uzeti nakon iznad navedenih proizvoda.



Slika 4.8 Grafički prikaz pravila asocijacije namirnica

Tablica 4.6 Pravila asocijacije namirnica

Rule	Antecedents	Consequence	Support	Confidence	Lift
Rule 5	butter	whole milk	0.0276	0.4972	1.9461
Rule 12	curd	whole milk	0.026	0.4895	1.9157
Rule 14	domestic eggs	whole milk	0.029	0.4649	1.8196
Rule 45	root vegetables	whole milk	0.0489	0.4487	1.756
Rule 58	whipped/sour cream	whole milk	0.032	0.4481	1.7536
Rule 19	root vegetables	other vegetables	0.0474	0.4347	2.2466
Rule 53	tropical fruit	whole milk	0.0423	0.4031	1.5776
Rule 28	whipped/sour cream	other vegetables	0.0287	0.4011	2.0731
Rule 59	yogurt	whole milk	0.0557	0.4003	1.5666
Rule 36	pip fruit	whole milk	0.0301	0.3978	1.557
Rule 29	other vegetables	whole milk	0.0748	0.3868	1.5136
Rule 33	pastry	whole milk	0.0324	0.3692	1.445
Rule 9	citrus fruit	whole milk	0.0305	0.3686	1.4424
Rule 8	citrus fruit	other vegetables	0.0289	0.3489	1.8031
Rule 16	pip fruit	other vegetables	0.0261	0.3454	1.7852
Rule 26	tropical fruit	other vegetables	0.0359	0.3421	1.7678

4.5. Diskusija

Svi algoritmi korišteni na prikazanim eksperimentima imaju svoje specifične prednosti i mane koje treba napomenuti.

k-NN je prilično jednostavan i intuitivan za korištenje, ali njegova mana ako radimo na vrlo velikim skupovima podataka i koristi puno računalnih resursa. Također može biti osjetljiv pri izračunu međusobnih udaljenosti entiteta ovisno o izabranoj metodi koja sa svojom kompleksnosti može usporiti sustav i pogoršati rezultate. Problem s ograničenim resursima sam osjetio kod različitog unosa susjeda, performanse u rasponu od 10 do 30 susjeda je izvođenje u milisekundama, u rasponu od 100 do 300 jest par sekundi dok za raspone od 1000 naviše se bližimo minuti.

k-Means je učinkovit u grupiranju podataka na osnovi središta srednje vrijednosti, ali zahtijeva pretpostavku o broju grupa i može biti osjetljiv na nasumični početni položaj središta te na podatke s različitim veličinama i gustoćama grupa. Njegova primjena može zahtijevati višestruko pokretanje s različitim početnim točkama zbog postizanja stabilnih rezultata. Naspram algoritma k-NN uz pomoć k-Means vrlo brzo možemo odrediti podjelu većih grupa i smatram da je to njegova idealna primjena, dok bi za k-NN bila idealna primjena na manjim i specifičnim skupinama. Time sam potvrdio podatke s interneta koji govore da je baš taj algoritam najčešće korišten pri grupiranju zbog jednostavnog razumijevanja i implementacije, te visoke efikasnosti [12].

Apriori je koristan za otkrivanje čestih uzoraka u velikom skupu podataka, ali može biti ograničen brojem transakcija i zahtijevati prilagodbu pragova podrške i sigurnosti ovisno o domeni podataka. Njegova kompleksnost raste s brojem jedinstvenih stavki u podatcima, što može utjecati na brzinu izvršenja. Eksperimenti provedeni u radu su sadržavali manji broj jedinstvenih stavki i zato je vrijeme izvođenja bilo nisko i prikaz podskupova precizan, ali sam također isprobao algoritam kada je broj jedinstvenih stavki velik i primijetio očigledan pad performansi (čekanje rezultata po nekoliko minuta, visoko zagrijavanje računala) i nepreglednost rezultata (česti podskupovi sadrže puno zasebnih elemenata iz kojih je teško izlučiti one bitne).

Korištene algoritme odlikuju mogućnosti pružanja dubokih uvida u podatke, uz izazove u efikasnosti i skalabilnosti ovisno o metodama i vrsti podataka.

Zaključak

Završetkom ovog rada smatram da sam vrlo dobro usvojio vrste analize podataka kojima je cilj grupirati i olakšati budući rad s njima. Skupovi podataka mogu biti nezgrapni za korištenje i na neki način ih moramo prilagoditi kako bi bili uporabljivi. Zato sam u svojem programu ostvario prilaganje bilo koje .csv datoteke kako bi omogućio univerzalnu primjenu za sve skupove podataka. Stilovi obrade podataka opisani u radu su na prvi pogled međusobno vrlo slični, ali ipak imaju svoje razlike i usmjerene primjene.

Prvi smjer koji je najopćenitiji i može se primijeniti na širok izbor područja se naziva pronalaženje sličnih entiteta. Tijekom upoznavanja s logikom pretrage i samim algoritmom k-NN, vrlo jasno sam uočio primjenu ovakve analize podataka. Ako imamo neki skup podataka unutar kojeg želimo pronaći točno određenu skupinu sličnih entiteta koja nas zanima ovakav način pretrage je definitivno jedan od najkorisnijih i najučinkovitijih. Sljedeće smo objasnili pojam grupiranja i njegovu ulogu u rudarenju podataka i strojnom učenju. Pomoću grupiranja, za razliku od prethodne metode, možemo nepregledni skup podataka razlomiti na zasebne cjeline, unutar kojih možemo pronaći pravilnosti jer su vrijednosti u grupama bliske. Posljednji način obrade, koji je meni bio najzanimljiviji, jest pronalaženje čestih podskupova pomoću Apriori algoritma. Nakon prolaska kroz podatke lako se uočavaju najbitniji elementi zadanog skupa jer se najčešće pojavljuju i uz pomoć takvih elemenata možemo stvarati buduća predviđanja. Osim očigledne primjere u poslovnom svijetu gdje je optimizacija prodaje i razlika u cijenama ključna, ovakva analiza je primjenjiva u medicini gdje traženjem određenih simptoma bolesti ili genetskih markera za predodređene bolesti.

Programska podrška ostvarena uz ovaj rad ima prostora za dodatni razvoj dodavanjem još algoritama ili optimizacijom i usavršavanjem postojećih. Usmjerenje rudarenja podataka u svrhe strojnog učenja je u današnjici najpopularniji smjer u IT sektoru, te se iz dana u dan otkrivaju nove tehnike i metode optimizacije u analizi podataka. Zato smatram da je ovaj rad vrlo relevantan jer obrađuje početne korake, bez kojih nije moguće razumjeti grananje ove široke teme.

Literatura

[1] Leskovec J., Rajaraman A., Ullman J. D. *Mining of Massive Datasets*. 2014.

[2] *K-Nearest Neighbour (KNN) Algorithm*, Geeks for geeks, (2024, siječanj). Poveznica: https://www.geeksforgeeks.org/k-nearest-neighbours/; pristupljeno 25. svibnja 2024.

[3] *NBA Stats (1947-present),* Kaggle, (2024, travanj). Poveznica: https://www.kaggle.com/datasets/sumitrodatta/nba-aba-baa-stats/data; pristupljeno 18. travnja 2024.

[4] *Grocery Store Dataset*, Kaggle, (2020, veljača). Poveznica: https://www.kaggle.com/code/ekrembayar/apriori-association-rules-grocery-store/input?select=Grocery+Products+Purchase.csv; pristupljeno 2. lipnja 2024.

[5] *Car Sales Report*, Kaggle, (2024, siječanj). Poveznica: https://www.kaggle.com/datasets/missionjee/car-sales-report/data; pristupljeno 25. travnja 2024.

[6] Bašić D. B., Šnajder J., *Uvod u umjetnu inteligenciju – prezentacija Strojno učenje*, FER ZEMRIS, 2020.

[7] Ng A., *Association Rules and the Apriori Algorithm*, Kdnuggets, (2016, travanj), Poveznica: https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html; pristupljeno 27. svibnja 2024.

[8] *Nearest Neighbours*, scikit-learn, Poveznica: https://scikit-learn.org/stable/modules/neighbors.html; pristupljeno 25. svibnja 2024.

[9] Gandhi R., *K Nearest Neighbours — Introduction to Machine Learning Algorithms*, Medium, (2018, lipanj), Poveznica: https://towardsdatascience.com/k-nearest-neighbours-introduction-to-machine-learning-algorithms-18e7ce3d802a; pristupljeno: 3. lipnja 2024.

[10] Gandhi R., *K-Means Clustering — Introduction to Machine Learning Algorithms*, Medium, (2018, lipanj), Poveznica: https://towardsdatascience.com/k-means-clustering-introduction-to-machine-learning-algorithms-c96bf0d5d57a; pristupljeno: 3. lipnja 2024.

[11] Chonyy, *Apriori — Association Rule Mining In-depth Explanation and Python Implementation*, (2020, listopad), Poveznica: https://towardsdatascience.com/apriori-association-rule-mining-explanation-and-python-implementation-290b42afdfc6; pristupljeno: 4. lipnja 2024.

[12] *Why do we use k-means instead of other algorithms?,* (2019, siječanj), Poveznica: https://stats.stackexchange.com/questions/58855/why-do-we-use-k-means-instead-of-other-algorithms; pristupljeno: 9. lipnja 2024.

Sažetak

Rad pod nazivom „Pronalaženje grupa, sličnih entiteta i čestih podskupova u analizi podataka“ obrađuje točno tri navedena načina za proučavanje skupova podataka. Algoritmi korišteni su k-NN, k-Means i Apriori. Prvi navedeni algoritam se koristi pri pronalasku sličnih entiteta, drugi za podjelu skupa podataka u grupe, a posljednji za uočavanje podskupova u obliku obrazaca na temelju kojih možemo raditi predikcije. Programska podrška je ostvarena u jeziku Python uz korištenje već ugrađenih modula i algoritama iz prikladnih paketa. Rezultati ovakvih načina analize se mogu navesti kao vrlo korisni u danjoj obradi i primjeni u IT sektoru, ali i u zdravstvenom sustavu, proizvodnji, financijama i mnogim drugim.

Summary

The paper titled "Finding clusters, similar items, and frequent itemsets in data mining" addresses exactly the three stated ways to study datasets. Algorithms used in this case are k-NN, k-Means and Apriori. The first listed algorithm is used to find similar entities, the second to divide a dataset into groups, and the third to observe subsets in a form of patterns on the basis of which we can make predictions. Software is implemented in the Python language with the use of already built-in modules and algorithms from suitable packages. The results of such methods of analysis can be cited as very useful in today's application in the IT sector, but also in the health system, production, finance and many others.

Skraćenice

k-NN	*k - Nearest Neighbours*	algoritam najbližih susjeda
k-Means	*k - Means Clustering*	algoritam srednje vrijednosti

Privitak

Instalacija programske podrške

Instalacija potrebnih paketa za pokretanje programa u Pythonu:

```
pip install pandas scikit-learn matplotlib mlxtend
```

Također preuzeti .csv datoteke koje želite analizirati.

Upute za korištenje programske podrške

Navigacija do direktorija u kojem ste preuzeli DataAnalyzer.py

Pokretanje programa uz pomoć naredbenog retka i naredbe:

```
python DataAnalyzer.py
```

Nakon pokretanja programa pritiskom na gumb *Select CSV File*, otvara se prozor u kojem odabirete željenu .csv datoteku za analizu. Zatim se otvara izbornik na kojem se nude tri vrste analize gdje možete odabrati željenu.

Odabirom na opciju *Finding Similar Items* i pritiskom gumba *Next,* korisniku se nudi opcija odabira parametra na temelju kojih će se tražiti sličnost. Nakon odabira parametara i unosa broja susjeda (veličina skupine sličnih entiteta) i pritiska gumba *Run Analysis* pokreće se analiza po parametrima i otvara prozor s pronađenim enitetima. Duplim pritiskom na redak koji korisnik želi proučiti, otvara se novi prozor s detaljnim prikazom entiteta.

Odabirom na opciju *Clustering* i pritiskom gumba *Next*, korisniku se nudi opcija odabira parametara na temelju kojih će se tražiti sličnost. Nakon odabira parametara korisnik ima opciju pritiska gumba *Run Elbow Method* koja će izračunati optimalan broj grupa za tražene parametre. Osim toga korisnik po svojoj volji može unijeti broj grupa koje želi vidjeti. Pritiskom na gumb *Run Analysis* prikazuje se tablica s pozicijom centra svake od grupa, te ako je korisnik odabrao dva ili tri parametra prikazuje se vizualizacija grupa.

Odabirom na opciju *Frequent Items* i pritiskom gumba *Next*, korisniku se nudi opcija odabira parametara na temelju kojih će se tražiti sličnost. Nakon odabira parametara i pritiska gumba *Run Analysis*, otvara se novi prozor s prikazom tablice u kojoj se nalaze podatci.