UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

# PRIPRAVA PODATKOV V PROCESU PODATKOVNEGA RUDARJENJA

Ljubljana, september 2013 ŽIGA VAUPOT

## IZJAVA O AVTORSTVU

Spodaj podpisani Žiga Vaupot, študent Ekonomske fakultete Univerze v Ljubljani, izjavljam, da sem avtor magistrskega dela Priprava podatkov v procesu podatkovnega rudarjenja, pripravljenega v sodelovanju s svetovalcem red. prof. dr. Jurijem Jakličem.

Izrecno izjavljam, da v skladu z določili Zakona o avtorskih in sorodnih pravicah (Ur. l. RS, št. 21/1995 s spremembami) dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

S svojim podpisom zagotavljam, da

- je predloženo besedilo rezultat izključno mojega lastnega raziskovalnega dela;
- je predloženo besedilo jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem
    - poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam v magistrskem delu, citirana oziroma navedena v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, in
    - pridobil vsa dovoljenja za uporabo avtorskih del, ki so v celoti (v pisni ali grafični obliki) uporabljena v tekstu, in sem to v besedilu tudi jasno zapisal;
- se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku (Ur. l. RS, št. 55/2008 s spremembami);
- se zavedam posledic, ki bi jih na osnovi predloženega magistrskega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom.

V Ljubljani, dne _____ Podpis avtorja:_____

# KAZALO

**SEZNAM SLIK**

**SEZNAM TABEL**

# UVOD

Z aplikacijami podatkovnega rudarjenja rešujemo poslovne probleme, kot so na primer razvrščanje komitentov banke, ki zaprošajo za posojila, v skupine glede na napovedana tveganja, ki so povezana z vračilom posojila. V zavarovalnicah uporabljajo podatkovno rudarjenje za identifikacijo prijavljenih škodnih primerov z namenom goljufije. Telekomunikacijska podjetja poskušajo napovedati, kateri od obstoječih kupcev bodo v naslednjih šestih mesecih nadgradili svoje storitve ali pa bodo z visoko verjetnostjo prešli h konkurenci. Trgovci analizirajo najpogostejše kombinacije izdelkov v nakupnih košaricah kupcev in na ta način snujejo svoje trženjske kampanje.

Podatkovno rudarjenje kot eno izmed področij poslovnega obveščanja v zadnjih letih s prehodom iz raziskovalnih laboratorijev v okvire standardnih aplikativnih programskih rešitev pridobiva na pomenu in uporabnosti. V svoji definiciji je podatkovno rudarjenje proces preiskovanja velikih baz podatkov s ciljem odkrivanja vzorcev in trendov, ki presegajo okvire enostavnega povpraševanja v bazah podatkov (Oracle, 2008, str. 1/1).

Ne glede na poslovne probleme, ki jih naslavljamo z metodami, funkcijami in algoritmi podatkovnega rudarjenja, je to predvsem bolj ali manj standarden proces, ki ga lahko opredelimo z več fazami (Shearer, 2000, str. 14):

1. raziskovanje in razumevanje poslovanja oziroma poslovnega problema,
2. spoznavanje s podatki,
3. priprava podatkov,
4. modeliranje,
5. vrednotenje modela in
6. uporaba.

Proces podatkovnega rudarjenja se v svoji definiciji in opredelitvi faz, ki ga sestavljajo, razlikuje od posameznega avtorja do avtorja oziroma ponudnika rešitev podatkovnega rudarjenja. Vendar ne glede na razlike v definiciji zavzemajo v projektih podatkovnega rudarjenja aktivnosti, povezane s pripravo podatkov, ključno mesto. Pyle (Pyle, 2003b, str. 366) navaja, da lahko priprava podatkov zajema tudi od 60 do 90 % vsega časa, potrebnega za izvedbo celotnega procesa podatkovnega rudarjenja. Hkrati pa dobra priprava podatkov prispeva kar od 75 do 90 % k uspešno izvedenemu projektu podatkovnega rudarjenja (Pyle, 2003b, str. 366). Nasprotno pa so slabi in nekonsistentni podatki odgovorni za neuspeh takšnega projekta (Pyle, 2003b, str. 366).

V fazi priprave podatkov se moramo tako vprašati, kateri podatki so ključni za izvedbo posameznega primera podatkovnega rudarjenja, kako te podatke pridobiti, kako jih organizirati in pripraviti, da bo podatkovno rudarjenje kar se da uspešno.

Soočamo se z različnimi, na prvi pogled precej enostavnimi, situacijami, kot so na primer (Larose, 2005, str. 27):

- podatek v okviru nekega atributa ne obstaja (na primer, ni letnice rojstva),
- obstaja podatek, ki izjemno odstopa od siceršnje zaloge vrednosti drugih podatkov v okviru istega atributa,
- podatki dveh atributov so visoko korelirani,
- podatki sicer obstajajo, vendar niso v obliki, ki je primerna za podatkovno rudarjenje,
- katere podatke sploh vzeti za izgradnjo modela, kateri so testni podatki.

Četudi so takšne in podobne situacije velikokrat na prvi pogled precej enostavne, se moramo vedno zavedati, da lahko kakršenkoli poseg ali neposeg v originalne podatke te popolnoma spremeni.

Pomen priprave podatkov za podatkovno rudarjenje opredeljujemo z več vidikov, med katerimi lahko navedemo vsaj tri (Zhang, Zhang & Yang, 2003, str. 377–378):

- Podatki v realnem svetu niso čisti, kar pomeni, da niso vedno popolni, vključujejo šum in so velikokrat nekonsistentni.
- Učinkovite aplikacije podatkovnega rudarjenja zahtevajo kakovostne podatke. Slednje lahko zagotavljamo s skrbnim izborom atributov v smislu njihove ustreznosti in zmanjšanja njihovega števila.
- Kakovostni podatki zagotavljajo kakovostne vzorce podatkov. Kakovost obstoječih podatkov v vzorcih lahko izboljšamo na več načinov. Med njimi so različni načini obnove nepopolnih podatkov, odprave napak, nenavadnih in izjemnih vrednosti.

Da bi bili podatki uporabni za izvedbo podatkovnega rudarjenja, jih moramo zato v fazi priprave prečistiti, ustrezno medsebojno povezati, zmanjšati njihovo število glede na njihov pomen, preoblikovati, preslikati, normalizirati (Han, Kamber & Pei, 2011, str. 103–107). Pri tem se moramo zavedati, da lahko napačno uporabljena strategija priprave podatkov poruši celoten vzorec podatkov in poda popolnoma napačne rezultate.

Nadomeščanje manjkajočega podatka je na primer na prvi pogled videti zelo enostavno. Manjkajočo vrednost atributa nadomestimo s povprečno vrednostjo vseh vrednosti atributa. S tem ohranimo povprečno vrednost stolpca. Po drugi strani pa vemo, da ima standardni odklon veliko večjo informacijsko vrednost od povprečne vrednosti. Mogoče je iz tega razloga primernejša strategija nadomeščanja vrednosti manjkajočega atributa, ki ohranja standardni odklon. Izbira strategije nadomeščanja manjkajočih vrednosti, ki jih izračunamo s pomočjo regresije, kot posledice korelacije z drugimi atributi vzorca podatkov, problem še poveča. (Pyle, 1999, str. 260–267).

Zgoraj navedene aktivnosti so praviloma potrebne zaradi tega, ker imajo različni algoritmi podatkovnega rudarjenja specifične zahteve glede podatkov, ki jih lahko pri vzpostavitvi modela sploh uporabijo. Nevronske mreže tako zahtevajo le numerične podatke, ki jih moramo pred izvedbo podatkovnega rudarjenja normalizirati, da bodo posamezni atributi obravnavani enakovredno in bo izvedba podatkovnega rudarjenja učinkovita. Če izvorni podatki niso numerični, jih moramo pred tem še dodatno preslikati v numerične vrednosti. Prav nasprotno pa nekateri drugi algoritmi, kot so na primer odločitvena drevesa, pričakujejo, da so vsi podatki, tudi numerični, v nominalni obliki (Pyle, 2003b, str. 369).

Podobno kot v omenjenih dveh primerih, se srečamo s specifičnimi zahtevami za pripravo podatkov pri drugih algoritmih podatkovnega rudarjenja. Tovrstne zahteve morajo biti izpolnjene in razrešene, preden se sploh soočimo z izgradnjo modelov in izvedbo same naloge podatkovnega rudarjenja.

Z veliko gotovostjo lahko zatrdimo, da če kje, potem za podatkovno rudarjenje velja pravilo »garbage in – garbage out«. Podatki, ki niso premišljeno pripravljeni, lahko namreč dajo zelo napačne rezultate od pričakovanih, kar se lahko neposredno odrazi na povečanju poslovnih tveganj in samem poslovanju podjetja.

Namen magistrskega dela je podrobneje raziskati problematiko priprave podatkov v procesu podatkovnega rudarjenja ter preveriti posamezne teoretične predpostavke na praktičnem primeru z realnimi podatki v informacijskem okolju, ki je v Sloveniji precej razširjeno.

Ključna vprašanja, na katera želim v okviru raziskovalne naloge odgovoriti, so:

- Kako se priprava podatkov umešča v proces podatkovnega rudarjenja in kateri so ključni postopki priprave podatkov v procesih podatkovnega rudarjenja?
- Katere tehnike in metode uporabljamo pri pripravi podatkov za doseganje čim boljših rezultatov aplikacij podatkovnega rudarjenja?
- Kakšne omejitve nam pri tem postavljajo programska orodja za podatkovno rudarjenje?
- Ali je mogoče na osnovi ugotovitev vzpostaviti enoten metodološki okvir priprave podatkov za podatkovno rudarjenje v izbranem informacijskem okolju?

Na ta način zbrane rezultate in ugotovitve povzemam v obliki predloga metodološkega okvirja za praktike, ki se že ukvarjajo ali se nameravajo ukvarjati s projekti podatkovnega rudarjenja, predvsem v Oracle okolju.

V okviru magistrske naloge sem na osnovi realnih podatkov založniško trgovskega podjetja izdelal model priprave podatkov za podatkovno rudarjenje. Tako pripravljene

podatke je mogoče nato uporabiti in dodatno prilogoditi za različne algoritme za izvedbo funkcij podatkovnega rudarjenja, kot so:

- segmentacija kupcev z razvrščanjem v gruče (angl. *clustering*),
- klasifikacija (angl. *classification*) za napovedovanje bodočega obnašanja kupcev glede na znano nakupno zgodovino,
- analiza nakupne košarice (angl. *market basket analysis*).

Za vse navedene funkcije podatkovnega rudarjenja je potrebna priprava podatkov, ki je izvedena na osnovi teoretičnih predpostavk, priporočil in dobre prakse, iz različnih virov in literature. Teoretičnih podlag in literature je na splošno temo podatkovnega rudarjenja načeloma precej, manj pa je literature na temo same priprave podatkov, ki je velikokrat vključena v širšo obravnavo področja podatkovnega rudarjenja in je zanjo v praksi dokazano, da ima ključno vlogo v procesu podatkovnega rudarjenja.

Za konkreten primer bodo ugotovitve iz naloge služile kot dobra osnova, kako se lotiti razvoja aplikacij podatkovnega rudarjenja in na kaj je potrebno še posebej paziti v fazi priprave podatkov.

Izhodišče za izdelavo naloge je obsežna strokovna literatura na področju podatkovnega rudarjenja. V večini primerov je priprava podatkov obravnavana le kot ena od faz v procesu podatkovnega rudarjenja, s čimer ji je praviloma posvečena manjša pozornost, kot bi bilo potrebno glede na njen pomen. Ne glede na to pa je v literaturi zaslediti veliko ugotovitev na osnovi teoretičnih razprav in dobrih praks, kako pravilno izpeljati pripravo podatkov za potrebe podatkovnega rudarjenja. V nalogi sem želel te ugotovitve povezati in jih predstaviti v obliki sinteze skupnih ugotovitev ter na njihovi osnovi vzpostaviti metodološki okvir, ki bo služil za njihovo preverbo na praktičnem primeru.

V praktičnem delu naloge sem za osnovo uporabil realne podatke knjižnega kluba. Te podatke sem na osnovi priporočil iz različnih virov in literature pripravil in organiziral tako, da so primerni za uporabo v aplikacijah podatkovnega rudarjenja. Posamezni koraki v pripravi podatkov so podrobno predstavljeni in ilustrirani s praktičnimi primeri.

Pri izvedbi naloge sem uporabil orodji za podatkovno rudarjenje v Oracle okolju, in sicer Oracle Data Miner, ki vključuje orodja za organizacijo procesa podatkovnega rudarjenja in priprave podatkov, ter Data Mining opcija v Oracle bazi podatkov, ki omogoča dejansko izvedbo posameznih funkcij podatkovnega rudarjenja.

Pri analizi podatkov sem si pomagal z odprtokodnim statističnim paketom R, pri čemer sem uporabil dodatek v Oracle bazi podatkov Oracle R Enterprise, ki omogoča izvedbe vseh statističnih analiz nad podatki neposredno v bazi podatkov.

V uvodu magistrskega dela opredeljujem podatkovno rudarjenje, kot so ga opredelili različni avtorji, in ga poskušam predstaviti z nekaterimi primeri. V literaturi je zaslediti veliko primerov uspešne uporabe, med katerimi sem izbral primere iz telekomunikacij, športa in prava. Primeri so predstavljeni v Prilogi 1.

V poglavju Proces podatkovnega rudarjenja je opredeljen proces podatkovnega rudarjenja. Procesni model CRISP-DM je danes najbolj razširjena opredelitev procesa podatkovnega rudarjenja, ki je v tem delu na kratko opisan.

V poglavju Priprava podatkov v okviru procesa podatkovnega rudarjenja so predstavljeni teoretični in praktični pristopi in tehnike v fazah spoznavanja in razumevanja podatkov ter njihove priprave. V okviru celotnega procesa se v nadaljevanju osredotočam predvsem na analizo in spoznavanje s podatki ter njihovo pripravo za podatkovno rudarjenje. Podatki so namreč le redkokdaj že pripravljeni v obliki, ki jo zahtevajo algoritmi podatkovnega rudarjenja. Zato ima v procesu podatkovnega rudarjenja priprava podatkov ključen pomen.

Ko govorimo o spoznavanju s podatki in njihovim razumevanjem, gre predvsem za pridobivanje podatkov ter njihovo opisovanje in raziskovanje. Pri slednjem izračunamo mere središčnosti ter mere razpršenosti in variabilnosti. Osnovne statistike si lahko predstavimo tudi v grafični obliki, saj velikokrat vizualna predstavitev pove več od številk.

V poglavju o pripravi podatkov so podrobneje opisane teoretične predpostavke s področja čiščenja podatkov, integracije podatkov, redukcije podatkov ter transformacije in diskretizacije podatkov.

V tretjem poglavju Priprava podatkov v Oracle okolju je predstavljena priprava podatkov za podatkovno rudarjenje na primeru knjižnega kluba. V okviru praktičnega primera se osredotočam na analizo izvornih podatkov, ki je izvedena s pomočjo statističnega orodja R (Oracle R Enterprise) in orodja za podatkovno rudarjenje (Oracle Data Miner), ki sem ga uporabil tudi pri vzpostavitvi procesa podatkovnega rudarjenja.

Zaključek dela predstavlja metodološki okvir priprave podatkov za podatkovno rudarjenje v Oracle okolju, ki temelji na ugotovitvah predstavljenega praktičnega primera.

# 1 PROCES PODATKOVNEGA RUDARJENJA

## 1.1 Podatkovno rudarjenje

Podatkovno rudarjenje je poslovni proces raziskovanja velikih količin podatkov z namenom odkrivanja smiselnih vzorcev in pravil (Linoff & Berry, 2011, str. 21). Pri tem ima ta definicija več elementov, od katerih je vsak ključen za podrobnejšo opredelitev podatkovnega rudarjenja.

Linoff in Berry (2011, str. 21) opredeljujeta podatkovno rudarjenje kot poslovni proces, ki je povezan z drugimi poslovnimi procesi v podjetju in je po svoji naravi stalen, nikoli zaključen proces. Podatkovno rudarjenje se pričenja s podatki, ki jih analiziramo, kar vodi v izvedbo aktivnosti, ki porodijo nove zahteve po podatkovnem rudarjenju. Za izboljšanje poslovanja podjetja nenehno zbirajo podatke, jih analizirajo in ukrepajo na osnovi rezultatov analize. Hkrati pa so strategije podatkovnega rudarjenja vključene v strategije trženja in razumevanja strank.

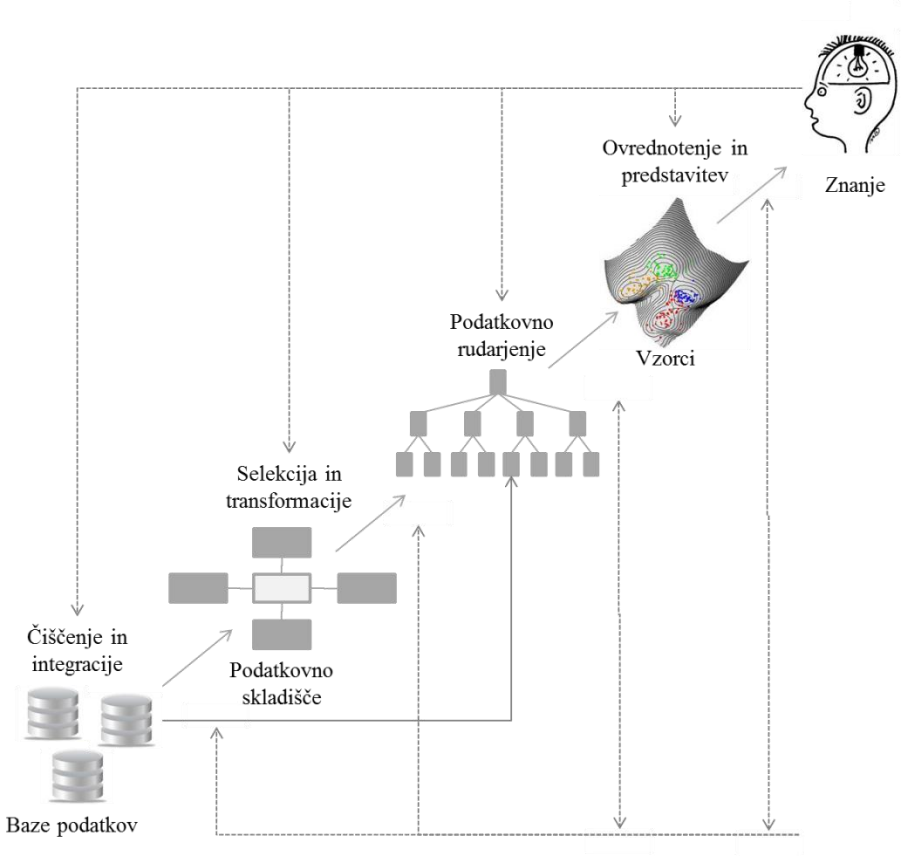
Dodatno ista avtorja navajata (Linoff & Berry, 2011, str. 22), da se tehnike podatkovnega rudarjenja niso bistveno spremenile od časov, ko so procesorske in pomnilniške omejitve računalnikov predstavljale veliko omejitev pri razvoju rešitev podatkovnega rudarjenja. Skozi desetletja so se računalniške kapacitete razvile in ne predstavljajo več bistvene omejitve za podatkovno rudarjenje. S tem pa so se količine podatkov, ki so bile pred 30 do 40 leti omejene na tabele z nekaj atributi in z nekaj sto zapisi, povečale za nekaj velikostnih razredov. Kot rečeno, so ostali algoritmi podatkovnega rudarjenja v osnovi povsem enaki.

Svojo definicijo podatkovnega rudarjenja Linoff in Berry (2011, str. 22–23) zaključujeta s smiselnimi vzorci in pravili, ki jih s podatkovnim rudarjenjem odkrivamo. Vzorcev v podatkih načeloma ni težko odkriti. Kar je pri odkrivanju vzorcev bistveno, je to, da je potrebno odkriti vzorce ali pravila, ki so pomembni za poslovanje. Se pravi, gre za podporo poslovnim aktivnostim, kot je recimo identifikacija tveganja poslovanja s posamezno stranko, napovedovanje najverjetnejših strank, ki se bodo odzvale na poslano ponudbo, ali izpis obvestila s priporočilom za nakup dodatnega izdelka v spletni trgovini glede na poznavanje nakupnih navad kupca in izdelkov, ki jih pravkar dodaja v nakupno košarico. Cilj takšnega odkrivanja vzorcev je v osredotočanju na kupce, ki bodo določene izdelke ali storitve najverjetneje potrebovali, pri čemer se poskuša prodajni proces čimbolj poenostaviti za vse, ki v njem sodelujejo.

Malce drugačen pristop k definiciji podatkovnega rudarjenja podajajo Han et al. (2011, str. 35–36), ko razširjajo vsebino podatkovnega rudarjenja z odkrivanjem znanja v podatkih (angl. *knowledge discovery from data*, v nadaljevanju KDD). V njihovi definiciji (glej

Sliko 1) je podatkovno rudarjenje le eden od korakov v procesu iskanja znanja v podatkih, s čimer opredeljujejo samo podatkovno rudarjenje podobno kot drugi avtorji, in sicer, da je podatkovno rudarjenje proces odkrivanja zanimivih vzorcev in znanja v velikih količinah podatkov.

*Slika 1: Podatkovno rudarjenje kot korak v procesu odkrivanja znanja*




*Vir: J. Han et al., Data Mining: Concepts and Techniques (3rd ed.), 2011, str. 36.*

Gartner Group (v Larose, 2005, str. 2) opredeljuje podatkovno rudarjenje kot proces odkrivanja smiselnih novih korelacij, vzorcev in trendov s presejanjem velikih količin podatkov, ki so shranjene v repozitorijih, z uporabo tehnologij za razpoznavanje vzorcev, kot tudi z uporabo statističnih in matematičnih tehnik. Cabena, Hadjinian, Stadler, Verhees in Zanasi (v Larose, 2005, str. 2) opredeljujejo podatkovno rudarjenje kot interdisciplinarno področje, ki združuje tehnike strojnega učenja, razpoznavanja vzorcev, statistike, baz podatkov in vizualizacije z namenom naslavljanja problema izluščenja informacij iz velikih baz podatkov.

Ker bodo v nadaljevanju predstavljeni primeri z uporabo Oracle tehnologije podatkovnega rudarjenja, naj navedem še Oraclovo opredelitev podatkovnega rudarjenja. Podatkovno rudarjenje (Oracle, 2008, str. 1/1) je proces preiskovanja velikih baz podatkov s ciljem odkrivanja vzorcev in trendov, ki presegajo okvire enostavne analize. Pri tem uporablja

podatkovno rudarjenje matematične algoritme za segmentacijo podatkov in ocenjevanje verjetnosti dogodkov v prihodnosti. Ključne lastnosti podatkovnega rudarjenja so po tej opredelitvi: avtomatično razpoznavanje vzorcev, predvidevanje najverjetnejših dogodkov, kreiranje informacij, potrebnih za odločitve, in osredotočenje na velike zbirke podatkov. Podatkovno rudarjenje podaja odgovore na vprašanja, na katera ne moremo odgovoriti z enostavnimi poizvedbami v bazi podatkov ali tehnikami poročanja.

Glede na vse navedeno lahko povzamemo navedbe in opredelimo podatkovno rudarjenje kot proces, ki ima cilj, da v velikih bazah podatkov s pomočjo statističnih, matematičnih in drugih metod identificira in odkrije vzorce in trende v podatkih, ki so pomembni za poslovanje podjetja.

## 1.2 Opredelitev procesa podatkovnega rudarjenja

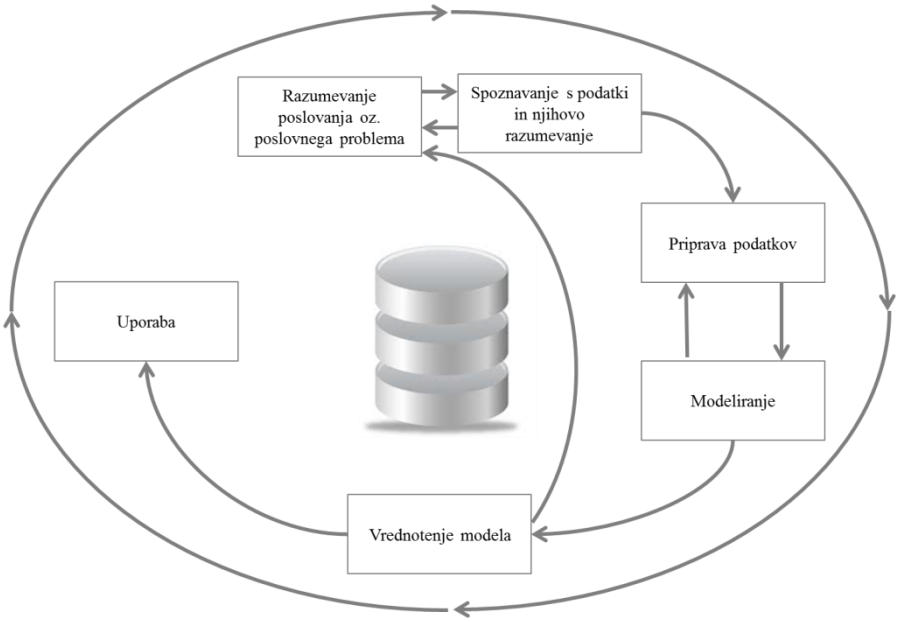
V drugi polovici devetdesetih let je razvoj aplikacij podatkovnega rudarjenja dosegel raven, na kateri se je vzpostavila potreba po standardizaciji procesa, ki bi podpiral razvoj tovrstnih rešitev (Shearer, 2000, str. 13). Nekatera vodilna podjetja, ki so razvijala tovrstne rešitve (Daimler-Benz, ISL, NCR in OHRA), so oblikovala standard in ga poimenovala CRISP-DM (CRoss-Industry Standard Process for Data Mining). Procesni model se je nato intenzivno razvijal v letih od 1997 do 1999 in v letu 2000 je bila objavljena prva različica standarda CRISP-DM 1.0. V letu 2006 je bila nato objavljena še druga različica standarda. Od takrat naprej konzorcij CRISP-DM SIG, ki ga je sestavljalo preko 300 podjetij, ni bil več aktiven oziroma je aktivnosti za nadaljnji razvoj prevzel IBM.

CRISP-DM je kot standard nevtralen v smislu uporabe v različnih industrijskih panogah. Prav tako je nevtralen do uporabljenih orodij in tehnologije ter podpira celoten proces podatkovnega rudarjenja, od identifikacije poslovne potrebe do uporabe. Ne glede na dejstvo, da je zdaj procesni model v »lasti« IBM-a in da ga neodvisni konzorcij ne razvija več, je to kljub vsemu eden od najbolj celovitih in uporabljenih procesnih modelov za razvoj aplikacij podatkovnega rudarjenja.

CRISP-DM predvideva 6 faz procesa podatkovnega rudarjenja (Shearer, 2000, str. 13–18; Chapman et al, 2000, str. 13–15), in sicer (glej Sliko 2):

1. razumevanje poslovanja oziroma poslovnega problema,
2. spoznavanje s podatki in njihovo razumevanje,
3. priprava podatkov,
4. modeliranje,
5. vrednotenje modela in
6. uporaba.

*Slika 2: Referenčni CRISP-DM procesni model*




*Vir: C. Shearer, The CRISP-DM Model: The New Blueprint for Data Mining, 2000, str. 14.*

### 1.2.1 Razumevanje poslovanja oziroma poslovnega problema

V prvi fazi procesa podatkovnega rudarjenja opredelimo poslovne cilje, ki jih želimo z razvojem rešitve doseči, in kriterije, ki določajo uspešnost ali neuspešnost v smislu doseganja postavljenih ciljev (Shearer, 2000, str. 14).

Med aktivnostmi, ki jih v tej osnovni pripravi izvajamo, je izdelava popisa vseh podatkovnih virov, ki jih bomo uporabili, ter opredelitev zahtev, predpostavk in omejitev. Prav tako poskusimo identificirati in oceniti tveganja, ter stroške in koristi, ki jih rešitev podatkovnega rudarjenja prinaša.

Operativno se proces podatkovnega rudarjenja opredeli kot projekt, za katerega izdelamo projektni načrt ter izberemo orodja in tehnike, s katerimi bomo razreševali problem podatkovnega rudarjenja.

### 1.2.2 Spoznavanje s podatki in njihovo razumevanje

Preden bomo pristopili k modeliranju podatkov, se moramo z njimi podrobneje spoznati in jih razumeti. Ta faza praviloma vključuje štiri korake, in sicer zbiranje, opisovanje, raziskovanje in preverjanje kakovosti podatkov (Shearer, 2000, str. 15).

V fazi spoznavanja s podatki je seveda prvi korak zbiranje podatkov, ki bodo vključeni v model podatkovnega rudarjenja. Zbrane podatke poskušamo inicialno prenesti v osnovno podatkovno strukturo, ki nam bo služila za izdelavo modela podatkovnega rudarjenja. V

primeru več podatkovnih virov se lahko že na tem mestu srečamo z vprašanjem integracije podatkov, ki je sicer predmet same priprave podatkov.

Ko imamo podatke zbrane, raziščemo njihove osnovne značilnosti, ki jih bomo podrobneje razčlenili v koraku raziskovanja podatkov. Z različnimi orodji za poizvedovanje, statistično analizo in vizualizacijo raziščemo distribucijske porazdelitve ključnih atributov, izdelamo enostavne statistike v okviru domene vrednosti posameznih atributov, kot so največja in najnižja vrednost, povprečna vrednost, število različnih vrednosti. Prav tako raziščemo, če obstajajo med posameznimi pari atributov korelacije. Na tej osnovi bomo izvajali v nadaljnjih fazah postopke čiščenja in konstrukcije podatkov.

Zbrane podatke nato analiziramo z namenom ugotavljanja njihove kakovosti. Neprimerno pripravljeni ini nekakovostni podatki se ne morejo uporabiti v modelu podatkovnega rudarjenja. Algoritmi enostavno ne bodo pravilno delovali. Osnovno ugotavljanje kakovosti podatkov vključuje analizo popolnosti podatkov, ugotavljanje obstoja neobstoječih ali izjemnih vrednosti v okviru posameznih atributov.

Faza spoznavanja s podatki je zelo povezana z naslednjo fazo procesa podatkovnega rudarjenja, tj. s fazo priprave podatkov. Zato ni izključeno, da bomo v nekem trenutku prešli neposredno v to fazo. Slednje je povezano tudi z orodji, ki jih pri tem uporabljamo.

### 1.2.3 Priprava podatkov

Faza priprave podatkov vključuje vse aktivnosti, ki jih je potrebno izpeljati, preden bomo podatke uporabili za izdelavo modelov podatkovnega rudarjenja. Aktivnosti priprave podatkov se lahko izvajajo večkrat, in nimajo vnaprej zapovedanega vrstnega reda. Med temi aktivnostmi so izbor tabel, zapisov, atributov zapisov, prečiščevanje podatkov v izbranih zapisih in potrebne transformacije podatkov (Shearer, 2000, str. 16).

Z vidika projekta podatkovnega rudarjenja traja priprava podatkov običajno od 50 do 70 % celotnega projekta (Shearer, 2000, str. 15). Pyle (2003b, str. 366) celo navaja, da lahko priprava podatkov zajema kar do 90 % vsega časa v takšnem projektu. Če upoštevamo, da Pyle združuje fazo razumevanja podatkov, ki po Shearerju (2000, str. 15) obsega od 20 do 30 % projekta, in fazo priprave podatkov v enotno fazo priprave podatkov, potem ta dejansko zajema od 60 do 90 % vseh aktivnosti v procesu podatkovnega rudarjenja, kar daje fazama razumevanja in priprave podatkov ključen pomen.

CRISP-DM predvideva v tej fazi pet korakov, in sicer: izbor, čiščenje, konstrukcijo, integracijo in formatiranje podatkov (Shearer, 2000, str. 16).

Pri izboru podatkov se mora poslovni analitik odločiti o podatkih, ki bodo vključeni v model podatkovnega rudarjenja. Kriterij izbire je odvisen predvsem od pomena, ki ga ima nek podatek za doseganje postavljenih ciljev, kot tudi od kakovosti in tehničnih omejitev, ki omejujejo razpoložljivost podatkov. Ko govorimo o izboru podatkov, imamo pri tem v mislih tako izbor posameznih atributov kot izbor posameznih zapisov tabele (Shearer, 2000, str. 16).

S čiščenjem podatkov želimo izboljšati kakovost podatkov, da dosežemo njihovo uporabnost za izvedbo posamezne funkcije podatkovnega rudarjenja. V primeru manjkajočih podatkov in drugih nečistoč v podatkih nekateri algoritmi ne bodo dali pričakovanih rezultatov ali pa sploh ne bodo pravilno delovali. Včasih se izkaže, da je potrebno konstruirati nove podatke, kar dosežemo s preslikavo posameznih atributov v nove, ki imajo večjo informacijsko vrednost za model podatkovnega rudarjenja (Shearer, 2000, str. 16).

Osnova za izdelavo podatkovnega modela podatkovnega rudarjenja je ena dvodimenzionalna tabela. Kadar imamo podatke v različnih tabelah (transakcijski podatki o prodaji so praviloma vedno v ločenih tabelah), moramo na osnovi podatkov v teh tabelah kreirati in integrirati nove zapise oziroma vrednosti atributov, ki so kasneje pridruženi osnovni tabeli, ki bo služila za osnovo modela podatkovnega rudarjenja (Shearer, 2000, str. 16–17).

Formatiranje podatkov se nanaša na transformacije, pri katerih gre predvsem za sintaktične modifikacije, ki ne spreminjajo pomena podatkov, so pa potrebne pri uporabi določenega algoritma ali orodja za podatkovno rudarjenje (Shearer, 2000, str. 17).

### 1.2.4 Modeliranje

V tej fazi se odločimo za uporabo funkcij podatkovnega rudarjenja. Za izvedbo posamezne funkcije je praviloma vedno mogoče izbirati med več različnimi algoritmi. Praviloma bomo različne algoritme tudi uporabili in se na koncu odločili za tistega, ki na osnovi testnih podatkov daje najboljše rezultate. Ob izgradnji modela z uporabo izbranega algoritma lahko z določanjem posameznih parametrov algoritma le-tega še dodatno optimiziramo (Shearer, 2000, str. 17).

V tej fazi bomo verjetno pripravili več zbirk podatkov, ki bodo namenjene izgradnji modela in njegovemu testiranju ter oceni kakovosti zgrajenega modela. Po izgradnji modela preverimo rezultate testiranja na testni podatkovni zbirki s poslovnimi analitiki oziroma strokovnjaki za področje, na katerem se izvajajo funkcije podatkovnega rudarjenja.

Zelo pogosto se v fazi modeliranja izpostavijo nova vprašanja glede kakovosti podatkov. Zato ni izključeno, da se iz faze modeliranja vrnemo v fazo priprave podatkov in da postopek priprave podatkov in modeliranja celo večkrat ponovimo, preden je model zrel za ovrednotenje in dokončno uporabo (Shearer, 2000, str. 17).

### 1.2.5 Vrednotenje modela

Faza vrednotenja modela je posvečena temeljiti oceni modela in pregledu posameznih korakov, ki so bili potrebni za izgradnjo modela. Pri tem je v ospredju vprašanje, ali smo s temi aktivnostmi dosegli zastavljene cilje oziroma ali smo karkoli pri tem spregledali (Shearer, 2000, str. 17).

V tej fazi so predvideni trije ključni koraki: ocena rezultatov, pregled procesa in določitev naslednjih korakov (Shearer, 2000, str. 18).

### 1.2.6 Uporaba

Izdelava modela seveda ne pomeni zaključka projekta podatkovnega rudarjenja. Rezultate podatkovnega rudarjenja, ki jih bomo dobili z uporabo modela na tretji zbirki podatkov (na primer možni kandidati za vključitev v trženjsko akcijo), moramo organizirati in predstaviti tako, da jih uporabnik lahko ustrezno uporabi. To pomeni, da se rezultate podatkovnega rudarjenja neposredno vključi v poslovne procese (Shearer, 2000, str. 18). Tak primer je recimo prilagoditev ponudbe vsaki stranki v spletni trgovini ali v telefonskem centru v realnem času, to je v trenutku, ko stranka odpre spletno stran oziroma je na telefonski liniji s prodajnim zastopnikom v klicnem centru.

### 1.2.7 Drugi procesni modeli podatkovnega rudarjenja

V literaturi in praksi obstaja kar nekaj opisanih modelov, ki pa so v osnovi standardnemu procesnemu modelu zelo podobni in se od njega razlikujejo le po organizaciji posameznih procesnih korakov.

Med popularnejšimi modeli lahko izpostavimo vsaj dva, in sicer SEMMA, ki ga uporablja SAS, in DMAIC 6-sigma pristop, ki se sicer primarno uporablja za industrijske aplikacije.

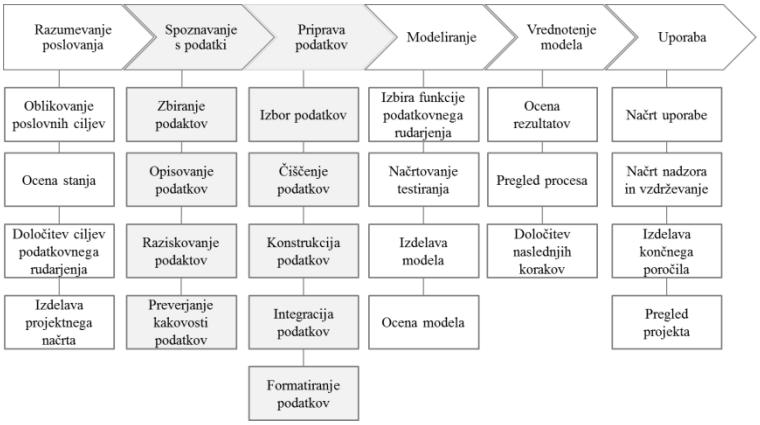
SEMMA predvideva 5 procesnih korakov, in sicer: vzorčenje, raziskovanje, manipuliranje, modeliranje in uporaba.

DMAIC pa predvideva naslednje korake: definicija, merjenje, analiza, izboljšave in kontrola.

# 2 PRIPRAVA PODATKOV V PROCESU PODATKOVNEGA RUDARJENJA

Priprava podatkov je v opisanem procesu podatkovnega rudarjenja samostojna faza, ki pa je zelo tesno povezana s predhodno fazo, tj. fazo spoznavanja s podatki in njihovim razumevanjem. Prav zaradi velike stopnje soodvisnosti med posameznima fazama in pomena kakovosti podatkov bom v nadaljevanju obravnaval in predstavil obe fazi (glej Sliko 3).

*Slika 3: Fazi spoznavanja in priprave podatkov v procesu podatkovnega rudarjenja*




*Vir: C. Shearer, The CRISP-DM Model: The New Blueprint for Data Mining, 2000, str. 14.*

## 2.1 Kakovost podatkov in pomen priprave podatkov

Vsi algoritmi podatkovnega rudarjenja delujejo na osnovi temeljne predpostavke, to je da so podatki, na osnovi katerih algoritmi zgradijo modele, ustrezne kakovosti. Z drugimi besedami to pomeni, da so lepo porazdeljeni, konsistentni, ne vsebujejo napak in ne vsebujejo manjkajočih vrednosti. Če ti pogoji niso izpolnjeni, lahko to vodi v prikrivanje uporabnih vzorcev, ki so skriti v podatkih. Prav tako so na ta način tudi sami rezultati podatkovnega rudarjenja slabše kakovosti (Zhang et al., 2003, str. 375).

Podatki so kakovostni tudi, če zadoščajo zahtevam nameravane uporabe. Na kakovost podatkov pa vpliva veliko dejavnikov, med njimi pravilnost, popolnost, konsistentnost, pravočasnost, verjetnost in razpoložljivost (Han et al., 2011, str. 103).

V velikih bazah podatkov se pogosto srečujemo s problemi nepravilnosti, nepopolnosti in nekonsistentnosti podatkov. Za netočnost podatkov, kar razumemo kot nepravilne vrednosti atributov, je veliko razlogov, od napak, ki nastanejo kot posledica napačnega vnosa podatkov v poslovno aplikacijo, do neodčitanja podatka na merilnem mestu. Za nepopolnost podatkov lahko prav tako naštejemo celo vrsto razlogov, med katerimi sta zelo pogosta dejanska nerazpoložljivost ali neobstoj podatka. V nekaterih primerih se

lahko dejansko zgodi, da se določen podatek ne zapiše, ker ne obstaja, ali pa je pri zapisovanju prišlo do napake. Pri podatkovnem rudarjenju se velikokrat srečamo s podatki o kupcih, med katerimi so podatki o njihovih naslovih. Naslovi in njihovo urejanje so tipičen primer, kako zelo hitro lahko podatki postanejo nekonsistentni, podvojeni in še kaj, če se ne urejajo dosledno in sistematično, kar pa se v praksi dogaja zelo pogosto.

Če v model podatkovnega rudarjenja vključujemo podatke na osnovi časovne serije, na primer dnevna prodaja po posameznih prodajnih mestih, je pravočasnost podatkov v smislu, da so datirani s pravilnim datumom transakcije, ključnega pomena za kakovost podatkov. Če pri zapisu podatkov pride do napačnega datuma transakcije iz kakršnegakoli razloga, podatki niso pravilni in nakupni vzorci, ki jih bo upošteval algoritem podatkovnega rudarjenja, ne bodo takšni, kot so v resnici.

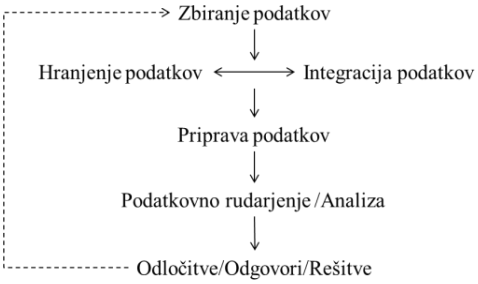
Han et al. (2011, str. 104) omenjajo še dva dejavnika, ki vplivata na kakovost podatkov, in sicer izpostavljata vprašanji, ali so podatki vredni zaupanja in ali so enostavno razumljivi.

Pomen priprave podatkov za podatkovno rudarjenje lahko zato opredelimo z več vidikov, med katerimi lahko navedemo vsaj tri (Zhang et al., 2003, str. 377–378):

- Podatki v realnem svetu niso čisti, kar pomeni, da niso vedno popolni, vključujejo šum in so velikokrat nekonsistentni.
- Učinkovite aplikacije podatkovnega rudarjenja zahtevajo kakovostne podatke. Slednje lahko zagotavljamo s skrbnim izborom atributov v smislu njihove ustreznosti in zmanjšanja njihovega števila.
- Kakovostni podatki zagotavljajo kakovostne vzorce podatkov. Kakovost obstoječih podatkov v vzorcih lahko izboljšamo na več načinov. Med njimi so različni načini obnove nepopolnih podatkov, odprava napak ter nenavadnih in izjemnih vrednosti.

Zagotavljanje kakovosti podatkov je kontinuiran proces (Dasu & Johnson, 2003, str. 100), ki se tesno prepleta s procesom podatkovnega rudarjenja (glej Sliko 4):

*Slika 4: Zagotavljanje kakovosti podatkov kot kontinuiran proces*




*Vir: T. Dasu in T. Johnson, Exploratory Data Mining and Data Cleaning, 2003, str. 101.*

## 2.2 Spoznavanje s podatki in njihovo razumevanje

V fazi spoznavanja in razumevanja podatkov si moramo odgovoriti na naslednja vprašanja (Nisbet, Elder & Miner, 2009, str. 51):

- Kako in kje najdemo podatke, ki jih potrebujemo za modeliranje?
- Kako bomo podatke povezali in integrirali, sploh, če jih najdemo v različnih podatkovnih virih?
- Kakšni so sploh ti podatki?
- Kako kakovostni so sploh ti podatki?

### 2.2.1 Podatki in njihovi atributi

Praviloma bomo podatke, ki jih bomo vključili v model, našli v zbirkah podatkov, kjer so zapisani v obliki podatkovnih zapisov. Vsak posamezen podatkovni zapis predstavlja preslikavo določenega objekta, entitete, iz realnega sveta, ki jo v okviru podatkovnega zapisa opišemo z *n*-tericami atributov. Na primer, hišo lahko opišemo z naslednjimi atributi: vrsta hiše, kvadratura, število nadstropij, leto izgradnje, velikost parcele.

Vsak atribut tako predstavlja posamezno lastnost objekta, ki ga opisujemo. Za podatkovno rudarjenje je pomemben tip atributa. Pyle (1999, str. 55) opredeljuje tipe atributov v smislu tipa uporabljene lestvice vrednosti. V tem smislu poznamo nominalne, ordinalne, numerične atribute na osnovi intervalne lestvice in numerične atribute na osnovi lestvice razmerij.

Nominalni atributi so opisni atributi, ki nimajo kakšne posebne vrednosti in bodo v modelu podatkovnega rudarjenja celo izpuščeni, saj so namenjeni predvsem identifikaciji in jih praviloma ne moremo razvrščati glede na njihovo vrednost.

Nominalne atribute imenujemo tudi kategorični, ker opisujejo določena stanja, ki jih je končno mnogo in pomenijo neko vrsto združevanja ali grupiranja. Lahko jih poimenujemo in/ali kategoriziramo, pri čemer velja, da med posameznimi kategorijami ne obstaja smiselna urejenost (Han et al., 2011, str. 68). Na primer, za atribut krvne skupine pacienta uporabljamo kategorije A, B, AB in 0, od katerih vsaka oznaka predstavlja posamezno krvno skupino, pri čemer med njimi ne obstaja nikakršen vrstni red ali druga urejenost. V podatkovnih modelih imamo zelo pogosto opraviti s poštnimi številkami. Tako poštni številki Trzina, 1236, in Slovenj Gradca, 2380, služita le za združevanje naslovov oseb ali podjetij v neko logično skupino, nikakor pa iz teh oznak ne izvemo ničesar o razmerjih med njima.

Posebna oblika kategoričnih atributov so binarni atributi (Han et al., 2011, str. 69), ki imajo lahko le dve različni stanji oziroma vrednosti. Na primer, če želimo označiti, ali je bil izveden nakup izdelka, potem lahko pomeni vrednost »1«, da je stranka izdelek kupila, vrednost »0«, pa da izdelka ni kupila. Podobno lahko v binarni obliki označimo spol osebe. Kar je pomembno pri opazovanju binarnih atributov, je njihova simetričnost. Če so vrednosti atributa enakomerno razporejene, potem je binarni atribut simetričen (v primeru spola), sicer je nesimetričen. Primer nesimetričnega binarnega atributa je odzivnost na trženjsko akcijo, kjer je lahko pozitivni odziv (praviloma se označuje pomembnejša vrednost z »1«) na primer 5 %, medtem, ko se 95 % potencialnih kupcev na ponudbo ni odzvalo (vrednost »0«).

Za razliko od nominalnih atributov, ordinalni atributi predpostavljajo neko smiselno urejenost ali razvrstitev posameznih vrednosti v okviru takšnega atributa. Ordinalni atributi tako predstavljajo neko razvrščenost, ne povedo pa nič o razmerjih med posameznimi vrednostmi v okviru takšnega atributa.

Na primer, če želimo opisati zadovoljstvo strank z opravljeno storitvijo, bomo mogoče uporabili naslednjo lestvico vrednosti: »zelo nezadovoljen«, »nezadovoljen«, »tako–tako«, »zadovoljen«, »zelo zadovoljen«. Iz posameznih vrednosti atributa vemo, kako je bila stranka zadovoljna. Vemo tudi, da je bila »zelo zadovoljna« stranka bolj zadovoljna od le »zadovoljne« stranke, vendar ne vemo, koliko bolj je bila zadovoljna.

Zelo pogosto se s pomočjo diskretizacije atributov kreira ordinalne atribute na osnovi vrednosti numeričnega atributa, ki ga razdelimo na končno število razredov.

Numerični atributi predstavljajo mere, ki so opisane s celimi ali realnimi števili, in so po svoji naravi kvantitativni. V osnovi delimo numerične atribute (Pyle, 1999, str. 55–57; Han et al., 2011, str. 70–72) na intervalne atribute, ki temeljijo na intervalni lestvici (angl. *interval scale measurment*), in na razmernostne atribute, ki temeljijo na lestvici razmerij (angl. *ratio scale measurment*).

V prvem primeru gre za atribute, ki zavzemajo vrednosti v nekem intervalu in nimajo absolutne začetne vrednosti. Zaradi tega jih lahko primerjamo med seboj po velikosti, lahko kvantificiramo razliko med njimi, ne znamo pa opisati razmerij, ki veljajo med njimi. Celzijeva temperaturna lestvica ima tako negativne kot pozitivne vrednosti. Obstaja tudi vrednost nič. Popolnoma jasno je, da je 5˚C več od -5˚C ali 0˚C. Izračunamo lahko tudi, da je 5˚C za 10 stopinj več od -5˚C. Ne moremo pa povedati, kolikokrat je prva vrednost večja od druge oziroma tretje. Na enak način obravnavamo tudi atribute z datumskimi vrednostmi, na primer datum rojstva.

Če bomo hoteli te podatke med seboj tudi kvalitativno primerjati, jih moramo preslikati v zato primerno obliko, kar pomeni preslikavo v vrednosti, ki so razporejene v okviru lestvice na osnovi razmerij.

V tem drugem primeru imamo opraviti z atributi, ki imajo določeno absolutno začetno vrednost, in jih med seboj ne le primerjamo v smislu »večji–manjši«, temveč lahko povemo, v kakšnem razmerju (na primer večkratnik) je neka vrednost z drugo. Za razliko od Celzijeve lestvice ima Kelvinova lestvica absolutno ničlo (0˚K je enako -273.15˚C) in s tem lahko opredelimo tudi razmerje med 278,15˚K in 268,15˚K (5˚C in -5˚C). Na podoben način dobimo starostna razmerja, če namesto datuma rojstva uporabljamo starost osebe.

### 2.2.2 Pridobivanje podatkov

Pridobivanje podatkov praviloma nikdar ni enostavno. Na osnovi opredeljenega poslovnega problema je pogosto potrebno podatke pridobiti iz več virov, ki so na voljo podjetju. Pri tem lahko naletimo na celo vrsto težav. Podatki se namreč lahko nahajajo v različnih transakcijskih bazah podatkov, podatkovnih skladiščih, preglednicah, datotekah. Prvi izziv se tako postavlja sam po sebi: identifikacija potrebnih podatkov in zagotovitev dostopa do njih.

Za pridobivanje podatkov iz različnih virov lahko uporabimo programske jezike, kot je na primer SQL (angl. *Structured Query Language*), standardni programski jezik za dostop in delo z relacijskimi bazami podatkov. Seveda pa lahko uporabimo katero od uveljavljenih programskih orodij za ekstrakcijo, transformacijo in nalaganje podatkov (angl. *Extract-Transform-Load*, v nadaljevanju ETL orodja), kot sta na primer Oracle Data Integrator ali Informatica PowerCenter. Predvsem kadar imamo opravka z več različnimi podatkovnimi viri, se tovrstnim orodjem ne moremo izogniti.

Poleg tega, da z ETL orodji dostopamo do podatkov in jih prenašamo v naš model podatkovnega rudarjenja, imajo ta orodja vključene funkcionalnosti, ki nam pomagajo pri integraciji in transformaciji podatkov. Zato se lahko ta orodja uporablja tudi v nadaljnjih korakih priprave podatkov za podatkovno rudarjenje.

### 2.2.3 Opisovanje in raziskovanje podatkov

Preden pristopimo k pripravi podatkov za podatkovno rudarjenje, je pomembno, da se s podatki podrobno seznanimo. Pri tem se za osnovno spoznavanje podatkov praviloma uporablja opisno statistiko, s pomočjo katere odkrijemo osnovne lastnosti podatkov, hkrati pa identificiramo vrednosti, ki predstavljajo šum v podatkih, osamelce ali druge posebnosti v podatkih.

Seznanitev s podatki običajno pričnemo z izračunom najbolj osnovnih mer središčnosti. V vsakem vzorcu podatkov najprej pridobimo podatke o skupnem številu elementov vzorca, vsoti vrednosti, poiščemo največjo in najmanjšo vrednost, preštejemo pojavitve posameznih vrednosti in drugo. Iz teh osnovnih mer lahko izračunamo druge mere, ki že podrobneje opisujejo vzorec podatkov, kot so aritmetična sredina, mediana in modus. Že na osnovi teh mer in njihovih razmerij lahko sklepamo na simetričnost in asimetričnost porazdelitve.

Podrobnejše informacije o podatkih, ki jih opazujemo, dobimo s pomočjo izračuna dodatnih mer in opisujejo:

- Razpršenost in variabilnost: Za merjenje razpršenosti uporabljamo razpon, kvantile, kvartile, percentile in interkvantilni razmik. Navedene mere prikažemo na grafu kvantilov (angl. *box-and-whiskerplot*, *boxplot*), ki je zelo primeren tudi za identifikacijo osamelcev. Varianca in standardni odklon sta prav tako meri statistične razpršenosti, ki prikazujeta, kako so vrednosti vzorca podatkov razporejene okoli linije pričakovanih vrednosti. Nizka vrednost standardnega odklona pomeni, da so vrednosti vzorca bliže aritmetični sredini, kar pomeni manjšo razpršenost. V primeru višje vrednosti standardnega odklona je porazdelitev vrednosti bistveno bolj razpršena.
- Heterogenost: Standardnega odklona in variance ni mogoče izračunati za nenumerične vrednosti, zato uporabljamo mere heterogenosti (na primer Ginijev koeficient heterogenosti ali entropijo), ki merijo razpršenost tudi za tak tip vrednosti.
- Koncentriranost: Za razliko od skrajnih vrednosti, ki jih opazujemo pri heterogenosti, opazujemo pri koncentraciji vmesne vrednosti. Frekvenčna porazdelitev je maksimalno koncentrirana, ko ima ničelno heterogenost, in obratno. Mera, s katero merimo koncentriranost, je Ginijev koeficient koncentriranosti.
- Asimetričnost (angl. *skewness*).
- Sploščenost (angl. *kurtosis*).

Osnovne statistike podatkov lahko prikazujemo tudi s pomočjo grafov, med katerimi so najbolj razširjeni kvantilni graf, Q-Q graf, histogram in korelacijski grafikon.

Podrobnejši opis navedenih opisnih statistik se nahaja v Prilogi 2.

## 2.3 Ključne naloge v procesu priprave podatkov

### 2.3.1 Izbira podatkov

Orodja podatkovnega rudarjenja praviloma zahtevajo pripravo podatkov v obliki dvodimenzionalne tabele. Nekatera orodja sicer omogočajo neposredne poizvedbe nad bazami podatki ali podatkovnimi skladišči, a ne glede na to je rezultat teh poizvedb v

obliki tabele. Naloga analitika podatkovnega rudarjenja je določiti izvor podatkov, iz katerih bo zgrajena tabela za podatkovno rudarjenje (Pyle, 1999, str. 117).

Pri tem je potrebno identificirati vire podatkov. Podatki se lahko nahajajo v transakcijskem sistemu, v katerem so shranjeni podatki iz trgovskih blagajn, bančnih avtomatov ali kateregakoli drugega sistema, v katerem so shranjeni zapisi o posameznih transakcijah (Pyle, 1999, str. 118).

V tipičnem informacijskem sistemu telekomunikacijskega ali trgovskega podjetja se podatki o kupcih in poslovanju z njimi nahajajo velikokrat v različnih podsistemih, kot so na primer CRM informacijski podsistem, informacijski podsistem tehnične podpore, spletna trgovina, zaledni informacijski podsistem. Če želimo uporabiti podatke o kupcih, ki se nahajajo v teh sistemih, jih je potrebno prenesti v enotno zbirko podatkov.

Linoff in Berry (2011, str. 146–150) poskušata v zvezi z izbiro podatkov najti odgovore na naslednja vprašanja:

- Kateri podatki so na voljo?
- Koliko podatkov je dovolj?
- Koliko zgodovine je potrebno?
- Koliko spremenljivk potrebujemo?
- Kaj mora biti vsebovano v podatkih?

Najbolj idealno mesto za začetek ugotavljanja, kateri podatki so na voljo, je podatkovno skladišče, v katerem so podatki že prečiščeni in urejeni. V podatkovnem skladišču so shranjeni zgodovinski podatki, katerim se dnevno dodaja nove podatke. Ker so podatkovna skladišča namenjena podpori poslovnemu odločanju, so v njih podatki praviloma tudi že agregirani, kar je lahko pri podatkovnem rudarjenju zelo uporabno (Linoff & Berry, 2011, str. 147).

Edini problem s podatkovnimi skladišči je, da jih nimajo vsa podjetja, ali da obstaja več področnih podatkovnih skladišč (angl. *data marts*), ki medseboj niso povezana. Zato se iskanje primernih podatkov za podatkovno rudarjenje preusmeri na operativne informacijske podsisteme, ki so lahko prav tako nepovezani in razdrobljeni. Še večji problem pa predstavlja njihova zasnova, saj je osnovni cilj teh sistemov, da procesirajo transakcije hitro in učinkovito. Podatki so zato v takšni obliki, ki najbolje zagotavlja doseganje tega cilja, podatki starejši od nekaj let, pa so tudi velika redkost (Linoff & Berry, 2011, str. 147).

Koliko podatkov je potrebnih, je odvisno od uporabljenega algoritma podatkovnega rudarjenja, kompleksnosti samih podatkov in relativne frekvence možnih izidov. Pri

podatkovnem rudarjenju je »več boljše«. S tem pa sta povezana dva zadržka, in sicer gostota podatkov in čas, ki ga imamo na voljo za izvedbo podatkovnega rudarjenja. V primeru gostote podatkov, mora biti vzorec podatkov tako velik, da nobena ciljna vrednost v njem ne more biti preredka oziroma premalo zastopana. V primeru časa pa preveliki vzorci zahtevajo več časa za izvedbo procesa podatkovnega rudarjenja, in kot smo videli, gre za iterativen proces, ki na ta način postane precej dolgotrajen (Linoff & Berry, 2011, str. 148–149).

Zgodovinski podatki so pri podatkovnem rudarjenju vsekakor pomembni, saj na njihovi osnovi sklepa algoritem na možne izide v prihodnosti. Najprej se srečamo s problemom sezonskosti, zaradi česar je potrebno pripraviti takšen vzorec, ki zajame dovolj dogodkov, da je vpliv sezonskosti iz podatkov zaznati (Linoff & Berry, 2011, str. 149). Drug pomemben faktor, ki nas omejuje pri odločitvi o tem, koliko zgodovine je potrebno, so spremenljive razmere na trgu. To je posebej pomembno, če je nanje vplival kak zunanji dogodek, kot je na primer pojav ekonomske krize. Linoff in Berry (2011, str. 149) navajata, da je za večino aplikacij, ki imajo opraviti s kupci, dovolj že za obdobje dveh do treh let zgodovinskih podatkov. A tudi v tem primeru je dobro vedeti, kdaj se je razmerje s kupcem pričelo, kateri je bil začetni prodajni kanal, koliko je kupec na začetku plačeval za storitve ali izdelke (Linoff & Berry, 2011, str. 149).

Neizkušeni analitiki podatkovnega rudarjenja velikokrat kar sami izločijo marsikateri atribut z namenom izbrati skrbno nadzorovan nabor atributov, ki ga želijo pri podatkovnem rudarjenju uporabiti. S tem nezavedno storijo največjo možno napako, saj nihče ne more vnaprej napovedati, ali ne bo nek izločen atribut za izid podatkovnega rudarjenja pravzaprav ključen. V zvezi z vprašanjem, koliko spremenljivk potrebujemo, je tako edini pravilen pristop ta, da podatki sami razkrijejo, kateri so pomembni in kateri ne (Linoff & Berry, 2011, str. 149). Končni model bo verjetno res vzpostavljen nad majhnim številom atributov, ki pa bodo nastali v procesu priprave podatkov. To pa je že tema, ki jo obravnam v nadaljevanju.

Zadnje vprašanje se ukvarja s tem, kaj mora sploh biti vsebovano v podatkih. Če imamo na primer opraviti z napovedovanjem nekega ciljnega izida, potem je nujno, da imamo v osnovni množici podatkov zastopane vse možne ciljne izide. Če bi kateri od možnih izidov v osnovni učni množici manjkal, algoritem na osnovi zgodovinskih podatkov seveda ne bi mogel sklepati na tak izid in ga ne bi mogel v nobenem primeru napovedati (Linoff & Berry, 2011, str. 150).

Zanimiv problem se pojavi, če podatkov z vsemi izidi sploh ni na voljo. Na primer, neko podjetje lahko hrani podatke le o tem, da so se posamezni kupci pozitivno odzvali na ponudbo, nimajo pa podatkov, komu vse so poslali ponudbo. V takšnem primeru seveda napovednih modelov ne morem zgraditi (Linoff & Berry, 2011, str. 150).

Pyle diskutira še en vidik izbire podatkov. To je problem dostopnosti do podatkov, za kar navaja naslednje razloge (Pyle, 1999, str. 118–120):

- Pravni razlogi: nekateri podatki, ki jih neko podjetje sicer ima na voljo, le-teh ne sme uporabiti v komercialne namene zaradi zakonodaje o varovanju osebnih podatkov.
- Organizacijski razlogi: včasih so nekateri podatki dostopni le določenim oddelkom, medtem ko zaradi razlogov poslovne tajnosti isti podatki niso dostopni drugim nepooblaščenim osebam v podjetju.
- Umetne meje med oddelki podjetja: včasih si posamezni oddelki lastijo podatke in jih ne dajejo na razpolago drugim oddelkom. Razlogi za to so lahko različni, vendar je to kar pogost pojav.
- Format podatkov: danes mogoče to niti ni več tako velik problem, ker se podatki v informacijskih sistemih nahajajo v poenotenih podatkovnih zbirkah, ki uporabljajo standardne znakovne nabore (UTF), vendar se ponekod podatki še vedno nahajajo na različnih diskovnih poljih, trakovih, zapisani v »starih« podatkovnih formatih ASCII, EBCDIC in podobno.
- Dostopnost in povezljivost: ko izvajamo podatkovno rudarjenje, predpostavljamo, da so podatki v izvornih sistemih, s katerimi se lahko povezujemo, in so ves čas dostopni. Če so podatki shranjeni na magnetnem traku, le-ta ni nujno ves čas dostopen.
- Arhitekturni razlogi: več kot je v informacijskem sistemu zbirk podatkov, v katerih se nahajajo podatki, ki jih pri podatkovnem rudarjenju potrebujemo, več časa in truda bo potrebno vložiti v to, da se vse podatke prevede na »skupni imenovalec«. Pri tem gre predvsem za različne podatkovne tipe, ki so v uporabi v različnih podatkovnih zbirkah in ki v nekaterih podatkovnih zbirkah nimajo ustreznega enakega podatkovnega tipa.
- Čas: podatki, ki jih imamo v nekem trenutku na voljo, lahko izhajajo iz različnih časovnih obdobij in mogoče niso več aktualni v trenutku, ko izvajamo podatkovno rudarjenje. Slednje lahko vodi v nepravilne vhodne podatke, ki so časovno neusklajeni in ki nujno vodijo v napačne rezultate.

Problemov in razlogov zanje je lahko seveda še več, kar je odvisno od vsakega primera podatkovnega rudarjenja posebej. Analitik podatkovnega rudarjenja se mora tega dobro zavedati in to pri pripravi podatkov za podatkovno rudarjenje tudi upoštevati.

Pri pridobivanju podatkov je potrebno omeniti še nekaj; in sicer, da imajo podatki notranji in zunanji izvor (Pyle, 1999, str. 119). Med tem ko gre pri notranjih virih za podatke, ki so v samem podjetju, gre pri slednjem za to, da lahko določene podatke na trgu tudi kupimo.

### 2.3.2 Čiščenje podatkov

#### 2.3.2.1 Manjkajoče in neobstoječe vrednosti

V fazi čiščenja podatkov se pogosto srečamo s primeri, ko vrednost določenega atributa v zapisu manjka. Pri tem je potrebno razlikovati med manjkajočimi in neobstoječimi vrednostmi, čeprav jih praviloma obravnavamo na enak način. Manjkajoča vrednost atributa je vrednost, ki dejansko obstaja, vendar ni vpisana v podatkovnem zapisu. Ko govorimo o neobstoječih vrednostih, le-te v realnem svetu ne obstajajo in jih seveda ni bilo mogoče vključiti med podatke (Pyle, 1999, str. 62).

Manjkajoče vrednosti je potrebno nadomestiti iz več razlogov (Pyle, 1999, str. 257):

1. Nekateri algoritmi modeliranja ne znajo obravnavati podatkovnih zapisov, v katerih manjka ena od vrednosti. Zaradi tega ti algoritmi izločijo celoten podatkovni zapis iz množice zapisov.
2. Nekatera orodja avtomatično nadomeščajo manjkajoče vrednosti s privzetimi vrednostmi, pri čemer se lahko vzorec, če je ta metoda nadomeščanja neprimerna, bistveno spremeni.
3. Izvajalec podatkovnega modeliranja mora poznati lastnosti posamezne metode nadomeščanja in nadzorovati proces nadomeščanja.
4. Večina privzetih metod nadomeščanja zavrže informacije, ki so morebiti vključene v vzorcih podatkov, ki imajo manjkajoče vrednosti.

##### 2.3.2.1.1 Vzorci manjkajočih vrednosti

Pyle (1999, str. 258–260) opozarja še na en problem povezan z manjkajočimi vrednostmi. Gre za same vzorce podatkov, ki imajo manjkajoče vrednosti. Ne glede na to, da jih nadomestimo z drugimi vrednostmi, lahko s tem vseeno pokvarimo celoten vzorec podatkov, saj je dejstvo, da je nek podatek, ki manjka, lahko celo ključen podatek in del določenega vzorca v modelu. Zato je ob nadomeščanju manjkajočih ali neobstoječih vrednosti pomembno ohraniti tudi informacijo o tem dejstvu.

Zato je smiselno (Pyle, 1999, str. 259), da se za manjkajoče in/ali neobstoječe vrednosti vpelje dodaten atribut, s katerim označimo vzorec manjkajočih vrednosti. Primerna oblika za označevanje manjkajočih vrednosti je, da jih označimo z zastavicami, ki označujejo prisotnost vrednosti določenega atributa v vzorcu (na primer zastavica P za prisotnost, O za odsotnost). Takšen nov »zastavični« atribut vsebuje zastavico za vsak atribut vzorca. Kadar imamo v podatkovnem zapisu tri atribute, lahko označimo prisotnost manjkajočih vrednosti, kot je to prikazano v tabeli 1:

*Tabela 1: Predlog označevanja vzorca manjkajočih vrednosti*

| Številka vzorca | Vzorec | Številka vzorca | Vzorec |
|---|---|---|---|
| 1 | PPP | 5 | OPP |
| 2 | PPO | 6 | OPO |
| 3 | POP | 7 | OOP |
| 4 | POO | 8 | OOO |

*Vir: D. Pyle, Data Preparation for Data Mining,1999, str. 259.*

Vsaki kombinaciji vzorca manjkajočih vrednosti ustreza številka vzorca, ki se povečuje s povečevanjem števila atributov v modelu. V primeru modela z 10 atributi imamo tako na voljo 1.024 kombinacij, v primeru 100 atributov pa že kar neobvladljivih $2^{100}$ kombinacij, ki jim moramo pripisati enolične vrednosti. Vendar po drugi strani v praksi ni pričakovati takšne razporeditve, ki bi zajemala vse možne kombinacije, temveč le nekaj ponavljajočih vzorcev manjkajočih vrednosti. Prav tako je smiselno izraziti vzorce manjkajočih vrednosti z več atributi.

#### 2.3.2.1.2 Nadomeščanje manjkajočih vrednosti

Ko razrešimo problem glede vzorcev manjkajočih vrednosti, lahko pristopimo k samemu nadomeščanju teh vrednosti. Vendar se takoj pojavi vprašanje, s katero vrednostjo naj manjkajoče vrednosti sploh nadomestimo. Pri tem velja ključno vodilo, da se pri tem v vzorec ne vnaša dodatnih motenj. Hkrati je izračunavanje nove vrednosti lahko precej kompleksna računska operacija, ki v primeru zelo velikega vzorca podatkov zahteva zelo veliko procesorske moči, ki je mogoče sploh nimamo na voljo, ali pa nimamo na voljo časa za izvedbo teh računskih operacij. Zato se bomo v primeru splošne uporabe verjetno morali zateči v kompromis med iskanjem optimalne vrednosti, s katero bomo nadomeščali manjkajoče vrednosti, in dejanskimi zmožnostmi ter omejitvami pri izvajanju računskih operacij.

Za razrešitev problema manjkajočih in neobstoječih vrednosti, predlagajo Han et al. (2011, str. 108; Chakrabarti et al., 2009, str. 72) uporabo enega od naslednjih pristopov:

- Podatkovnega zapisa ne upoštevamo.
- Manjkajočo vrednost vpišemo ročno.
- Uporabimo splošno konstanto, katere vrednost je na primer neznana vrednost.
- Manjkajočo vrednost nadomestimo z eno od statističnih vrednosti atributa, kot je na primer povprečna vrednost ali mediana.
- Manjkajočo vrednost nadomestimo z eno od statističnih vrednosti, podatkovnih zapisov, ki imajo isto skupno značilnost, na primer pripadajo isti skupini kupcev.

- Manjkajočo vrednost nadomestimo z najbolj verjetno vrednostjo, na primer s pomočjo regresije.

Pri nadomeščanju manjkajočih vrednosti Pyle (1999, str. 261) govori tudi o ohranjanju razmerij med vrednostmi v okviru posameznega atributa in o ohranjanju razmerij med posameznimi atributi v okviru vzorca.

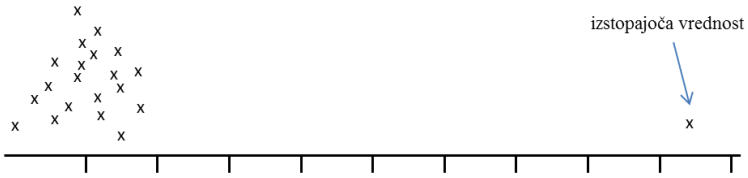
Če imamo atribut, v katerem so vrednosti 1, 2, 3, *y* in 5, pri čemer je *y* manjkajoča vrednost, potem se seveda postavlja vprašanje, kakšna naj bo ta vrednost. Osnovno vodilo pri pripravi podatkov za podatkovno rudarjenje je, da se pri tem poskuša čim bolj ohraniti značilnosti izvornih podatkov. V primeru manjkajočih vrednosti to pomeni, da vrednosti s katerimi nadomeščamo manjkajoče vrednosti ne pokvarijo obstoječega vzorca. A kaj to pravzaprav dejansko pomeni?

Če nadomeščamo manjkajočo vrednost z vrednostjo, ki ohranja povprečno vrednost atributa, potem je vrednost, ki jo iščemo 2,75. Če pa želimo ohraniti standardni odklon atributa, potem moramo *y* nadomestiti z 4,659. Se pravi, pri nadomeščanju moramo upoštevati razmerja med vrednostmi v okviru atributa in razmerja, v katerih je posamezen atribut z drugimi atributi v podatkovnem zapisu. Določitev tega, katera razmerja so pomembna in jih je potrebno ohraniti, določa, katere ocenjene vrednosti bomo uporabili pri nadomeščanju manjkajočih vrednosti.

#### 2.3.2.2 Izstopajoče vrednosti ali osamelci

Pri podatkovnem rudarjenju pogosto naletimo v vzorcih, ki imajo normalno porazdelitev, na eno (glej Sliko 5) ali več izstopajočih vrednosti, ki bistveno odstopajo od večine vrednosti v vzorcu. Takšne vrednosti imenujemo osamelci (angl. *outliers*).

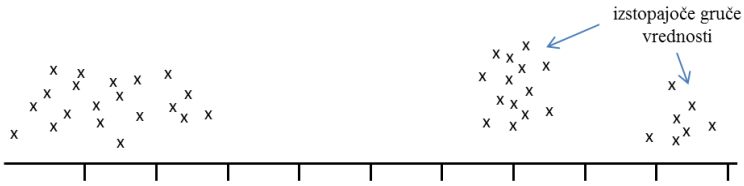
*Slika 5: Primer osamelca*




*Vir: D. Pyle, Data Preparation for Data Mining, 1999, str. 72.*

Drug primer izstopajočih vrednosti (glej Sliko 6) najdemo v vzorcih podatkov, v katerih so v podatkih povezani v gruče, med njimi pa so območja, v katerih se nahajajo posamezne vrednosti, ki ne pripadajo nobeni od gruč.

*Slika 6: Izstopajoče gruče osamelcev*




*Vir: D. Pyle, Data Preparation for Data Mining, 1999, str. 72.*

Izstopajoče vrednosti lahko pomembno vplivajo na vzorec podatkov, celo do te mere, da ta postane neuporaben. Pogosto se osamelci izkažejo kot napake, ki jih je vsaj v okviru vzorca mogoče odpraviti. Praviloma se jih obravnava na enak način kot manjkajoče vrednosti. Večji problem nastane, če se takšna izstopajoča vrednost ne more identificirati kot napaka, ali pa to dejansko sploh ni kot v primeru pologov ali dvigov sredstev s tekočega računa. Ti so običajno majhni, občasno pa kdo položi ali dvigne večjo vrednost, kar moramo pri analizi seveda upoštevati. Zato je potrebno, da se pojava izstopajočih vrednosti zavedamo in da ga ustrezno obravnavamo, tj. vrednosti vzorca transformiramo s pomočjo normalizacije in redistribucije.

### 2.3.3 Integracija podatkov

Integracija podatkov je korak, ki ga v pripravi podatkov za podatkovno rudarjenje ne moremo zaobiti ali izpustiti, saj je potrebno pripraviti eno samo tabelo, nad katero bomo izvajali operacije algoritma podatkovnega rudarjenja.

Integracija podatkov je praviloma zelo rutinska operacija, pri kateri lahko uporabimo številna programska orodja, kot so na primer ETL orodja. Pri tem vir podatkov praviloma ni le ena sama tabela v podatkovni zbirki, temveč je velikokrat potrebno združevati podatke iz različnih virov, kot so različne podatkovne zbirke, tekstovne datoteke ali vsaj različne tabele v okviru posamezne podatkovne zbirke. Pri tem naletimo na celo vrsto problemov, ki lahko vodijo do nekonsistentnosti podatkov, ki lahko vplivajo na učinkovitost podatkovnega rudarjenja.

Ključne probleme, na katere naletimo pri integraciji podatkov, lahko strnemo v naslednje skupine (Han et al., 2011, str. 114):

- problem identifikacije entitet,
- pojav redundance v podatkih,
- podvajanje vrednosti in
- razpoznavanje in razreševanje konfliktov povezanih z vrednostmi podatkov.

#### 2.3.3.1 Problem identifikacije entitet

Entitete realnega sveta so v podatkovnih zbirkah opisane s podatkovnimi strukturami, ki se od podatkovne zbirke do podatkovne zbirke razlikujejo. Recimo, da imamo opravka z dvema različnima podatkovnima zbirkama, v katerih se nahajajo podatki o kupcih. V prvi podatkovni zbirki se za identifikator kupca uporablja atribut ID_KUPCA. V drugi pa je uporabljen za isto entiteto identifikator atribut STEVILKA_STRANKE. Mogoče se razlikujeta tudi v tipu podatka. Kako naj na tej osnovi izvedemo integracijo podatkovnih shem? Ali gre sploh za isti atribut? V takšnih primerih je zelo smiselno dopolniti atribute z meta podatki, ki poleg imena atributa vsebujejo še pomen atributa, tip podatka, razpon dovoljenih vrednosti v okviru atributa, pravila za obravnavo neobstoječih, ničelnih ali NULL vrednosti (Han et al., 2011, str. 114).

Pri ugotavljanju ujemanja atributov iz različnih podatkovnih zbirk je potrebno posebej paziti na strukturo podatkov. Potrebno je zagotoviti, da se morebitne funkcijske odvisnosti in referenčne omejitve ohranijo tudi po izvedeni integraciji (Han et al., 2011, str. 114). Kot primer za to, navajajo Han et al. (2011, str. 114) obračun popusta, ki je lahko v enem sistemu obračunan na nivoju celotnega naročila, medtem, ko je v drugem sistemu obračunan na nivoju posamezne vrstice naročila. Če tega dejstva ne poznamo pred integracijo podatkov ali ga pri tem ne upoštevamo, potem lahko pride v ciljnem sistemu do napačnega izračuna popusta pri naročilu.

#### 2.3.3.2 Pojav redundance v podatkih

V primeru integracije več različnih virov podatkov pogosto pride do pojava redundance posameznih podatkov, ki je pravzaprav lahko prisotna tudi že v osnovni zbirki podatkov. Vsekakor je cilj, da poskusimo pojav redundance identificirati ter jo iz vzorca podatkov odstraniti.

Najpogostejši obliki redundance, s katerima se pri tem srečamo, sta (Han et al., 2011, str. 114) pojav istega atributa v različnih podatkovnih zbirkah ali tabelah, ki je različno poimenovan, in pojav atributa, ki je izveden iz drugih atributov, na primer v drugi tabeli.

Za podatkovno rudarjenje pojav redundance pomeni, da bo zaradi tega, ker so podatki bodisi podvojeni ali nekonsistentni, potrebno več časa in procesorskih kapacitet za izvedbo algoritmov podatkovnega rudarjenja. Zaradi velike soodvisnosti posameznih atributov pa lahko to tudi vpliva na njihovo izvedbo ter s tem na samo kakovost rezultatov podatkovnega rudarjenja.

Redundantne atribute odkrivamo s pomočjo korelacijske analize in analize kovariance. Korelacija dveh atributov meri odvisnost enega atributa od drugega, oziroma pove, kako se

vrednost enega atributa spreminja v odvisnosti od sprememb vrednosti drugega. Kovarianca pa predstavlja mero, ki pove, kako sta dve spremenljivki povezani.

Za ugotavljanje redundantnosti nominalnih podatkov uporabljamo test $\chi^2$(*hi-kvadrat*), medtem ko za numerične podatke uporabljamo korelacijski (Pearsonov) koeficient in kovarianco (Han et al., 2011, str. 114).

#### 2.3.3.2.1 Ugotavljanje redundantnosti nominalnih podatkov s pomočjo korelacijske analize

Redundantnost nominalnih podatkov ugotavljamo s pomočjo testa $\chi^2$. $\chi^2$ test statistične signifikantnosti je serija matematičnih formul, ki primerjajo opazovane frekvence nekega dogodka v vzorcu s frekvencami, katere bi pričakovali, če ne bi bilo nobene povezave med spremenljivkama v večji (vzorčeni) populaciji. $\chi^2$ test testira naše trenutne rezultate proti ničelni hipotezi in pove, ali so rezultati dovolj različni, da preidejo določeno verjetnost, da so nastali zaradi napačnega vzorčenja (Hi-kvadrat test, 2012).

Kadar imamo dva atributa A in B z vrednostmi $a_1, a_2, \ldots a_p$ in $b_1, b_2, \ldots b_r$, število posameznih skupnih pojavitev $(A_i, B_j)$ posameznih parov zapišemo v kontingenčno tabelo. Vrednost $\chi^2$ izračunamo s pomočjo formule:

$$\chi^2 = \sum_{i=1}^{p} \sum_{j=1}^{r} \frac{\left(o_{ij} - e_{ij}\right)^2}{e_{ij}},$$

pri čemer je $o_{ij}$ dejansko število pojavitev para vrednosti $\left(A_i, B_j\right)$, $e_{ij}$ pa pričakovano število pojavitev para vrednosti $\left(A_i, B_j\right)$. Pričakovano število para vrednosti $\left(A_i, B_j\right)$ je:

$$e_{ij} = \frac{\check{s}tevilo\ pojavitev(A = a_i) \times \check{s}tevilo\ pojavitev(B = b_j)}{n};$$

vrednost $n$ je število vseh parov, vrednost $\check{s}tevilo\ pojavitev(A = a_i)$ je število vseh parov, kjer je vrednost atributa $A$ enaka $a_i$ in vrednost $\check{s}tevilo\ pojavitev\left(B = b_j\right)$ je število vseh parov, kjer je vrednost atributa $B$ enaka $b_j$ (Han et al., 2011, str. 115).

Test $\chi^2$ temelji na hipotezi, da sta atributa A in B neodvisna oziroma med njima ni korelacije. Torej večja kot je vrednost $\chi^2$, bolj sta atributa v soodvisna.

#### 2.3.3.2.2 Ugotavljanje redundantnosti numeričnih podatkov s pomočjo korelacijske analize

V primeru, da imata atributa A in B numerične vrednost, izračunamo njuno soodvisnost s pomočjo korelacijskega koeficienta (tudi Pearsonovega korelacijskega koeficienta).

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B},$$

pri čemer je $n$ število vseh parov vrednosti, $a_i$ in $b_i$ sta vrednosti A in B para $i$, $\bar{A}$ in $\bar{B}$ sta povprečni vrednosti atributov A in B, $\sigma_A$ in $\sigma_B$ pa sta standardna odklona vrednosti atributov A in B (Han et al., 2011, str. 116).

Vrednosti $r_{A,B}$ se nahajajo v območju vrednosti $-1 \leq r_{A,B} \leq 1$. Če je $r_{A,B} > 0$, potem sta atributa A in B pozitivno korelirana. To pomeni, da če povečamo vrednost A, se poveča tudi vrednost B. Atributa A in B sta negativno korelirana, če se ob povečanju vrednosti atributa A vrednost atributa B zmanjša. V tem primeru je njun korelacijski koeficient negativen, $r_{A,B} < 0$. Večji kot je korelacijski koeficient, bolj sta atributa A in B korelirana. Če je $r_{A,B} = 0$, potem sta atributa A in B neodvisna (Han et al., 2011, str. 116).

Pri korelacijski analizi je pomembno upoštevati tudi, da korelacija ne predpostavlja recipročne vzročne povezanosti, kar z drugimi besedami povedano pomeni, da ne drži nujno, da je $r_{A,B} = r_{B,A}$. Na primer, število stalno prijavljenih prebivalcev na določenem območju lahko pomeni, da bo na tem območju prijavljenih tudi veliko vlomov v avtomobile. Po drugi strani pa veliko število prijavljenih vlomov v avtomobile še ne pomeni nujno, da na določenem območju stalno živi veliko število prebivalcev, na primer v finančnih četrtih velemest (Han et al., 2011, str. 116).

#### 2.3.3.2.3 Kovarianca in ugotavljanje redundantnosti atributov

Podobno kot s korelacijskim koeficientom, tudi s kovarianco merimo, kako se vrednosti dveh numeričnih atributov spreminjata v soodvisnosti.

V primeru dveh atributov A in B se vrednosti posameznih atributov pojavljajo v parih $\{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}$. Povprečni vrednosti atributov A in B imenujemo tudi pričakovani vrednosti $\bar{A}$ in $\bar{B}$:

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} \quad \text{in} \quad E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}.$$

Kovarianco med atributoma A in B izračunamo s formulo (Han, 2011, str. 116):

$$Cov(A,B) = E\left((A-\bar{A})-(B-\overline{B)}\right) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}.$$

Velja tudi naslednje (Han et al., 2011, str. 117):

$$r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B},$$

kjer sta $\sigma_A$ in $\sigma_B$ standardna odklona vrednosti atributov A in B.

Za izračun kovariance lahko uporabimo tudi naslednjo enačbo (Han et al., 2011, str. 117):

$$Cov(A,B) = E(AB) - \bar{A}\bar{B}.$$

#### 2.3.3.3 Podvajanje vrednosti

Poleg redundance med atributi je pomembno ugotoviti podvojene vrednosti v okviru vzorca podatkov po izvedeni integraciji podatkov iz različnih virov. Te pojave je potrebno ustrezno obravnavati, saj lahko dve ali več identičnih *n*-teric vrednosti opisuje isti dogodek (Han et al., 2011, str. 118).

Dodatno lahko na podvojenost zapisov, zaradi izboljšanja performans, vplivamo z denormalizacijo podatkov. Nekonsistentnost je v tem primeru lahko posledica napak pri vnosu in ažuriranju podatkov v izvorni zbirki podatkov. Nenazadnje je zaradi različnih podatkovnih struktur v različnih izvornih podatkovnih zbirkah v eni podatkovni zbirki podatek o kupcu zapisan neposredno med podatki naročila, v drugi pa se na podatke kupca samo sklicujemo z identifikacijsko številko, medtem ko so podatki o kupcu zapisani v tabeli kupcev. Pri tem lahko pride do podvojitve, tako da je podatek o naročilih nekonsistenten, saj je mogoče, da se podvojijo podatki o istem naročilu, ki pa ima različne podatke o kupcu (Han et al., 2011, str. 118).

#### 2.3.3.4 Razpoznavanje in razreševanje konfliktov, povezanih z vrednostmi podatkov

Vrednost podatka v okviru atributa je lahko narobe razumljena, če je njegova predstavitev različna v različnih podatkovnih virih. Tako imamo lahko povsem nekonsistentne podatke, če ne poznamo na primer merske enote, v kateri je določena vrednost zapisana ali če ne poznamo pretvornih pravil, kako pretvoriti vrednost enega atributa v drugega.

### 2.3.4 Redukcija podatkov

Ena od ključnih nalog v pripravi podatkov za podatkovno rudarjenje je odstranitev multikolinearnosti v podatkih (Larose, 2005, str. 84). Multikolinearnost lahko vodi v nestabilnost modelov in neskladne rezultate. V primeru uporabe multiple regresije lahko pride do signifikantne regresije na osnovi vhodnih spremenljivk, od katerih nobena ni signifikantna. Četudi ne pride do takšnega rezultata, obstaja možnost, da se posamezen vzorec podatkov v modelu preveč poudari, saj je zaradi velike koreliranosti dveh ali več podatkov, takšen vzorec predstavljen in upoštevan dvakrat ali celo večkrat.

Po drugi strani so lahko kompleksne analize podatkov in samo podatkovno rudarjenje na velikih količinah podatkov dolgotrajne ter zahtevajo veliko računalniških zmogljivosti, kar vodi v njihovo neuporabnost in neučinkovitost (Han v Chakrabarti et al., 2011, str. 84).

S pomočjo tehnik redukcije podatkov tako zmanjšamo količino podatkov, pri čemer dobimo manjše množice podatkov, ki so ohranile integriteto osnovne množice podatkov. Z drugimi besedami to pomeni, da pripravimo manjšo množico podatkov, ki bo še vedno zagotavljala enake rezultate kot osnovna množica. Han et al. (2011, str. 119) predlagajo naslednje možne strategije za redukcijo podatkov:

- Zmanjšanje dimenzionalnosti je proces zmanjšanja števila naključnih spremenljivk. Med tehnikami zmanjšanja dimenzij oziroma atributov navajajo Han et al. (2011, str. 119) diskretno valčno transformacijo (angl. *discrete wavelet transformation – DWT*) in analizo glavnih komponent (angl. *principal component analysis – PCA*). Tretja navedena tehnika (Han et al., 2011, str. 119) je izbor podmnožice atributov (angl. *atribut subset selection, feature subset selection*), pri kateri se identificirajo in odstranijo nepomembni, manj pomembni ali podvojeni atributi.
- Pri zmanjšanju številčnosti gre za zamenjavo količine osnovnih vrednosti z alternativno obliko predstavitve podatkov. Pri tem ločimo parametrske in neparametrske metode. Pri prvih gre za to, da se originalni podatki zamenjajo z modelom, katerega parametri opisujejo te podatke. Poleg tega se lahko shranijo še osamelci. Primera takšnega parametrskega modela sta regresijski in loglinearni model. Neparametrske metode, s katerimi lahko prikažemo zmanjšanje številčnosti, so prikazi s histogrami, gručenje, vzorčenje in agregacija podatkov v podatkovnih kockah.
- Kompresija podatkov je transformacija, pri kateri se zmanjša (kompresira) obseg originalnih podatkov. Pri tem ločimo dve vrsti kompresije podatkov. Če lahko po kompresiji podatkov te rekonstruiramo brez izgube, imenujemo takšno kompresijo podatkov brez izgub, sicer gre za kompresijo podatkov z izgubo. Pri kompresiji podatkov je potrebno upoštevati, da je s takšnimi podatki običajno težko kakorkoli manipulirati, zato smo pri analizi lahko omejeni. Po drugi strani pa lahko razumemo

tako zmanjšanje dimenzionalnosti kot zmanjšanje številčnosti kot obliko kompresije podatkov.

Možnosti za redukcijo podatkov v procesu priprave podatkov za podatkovno rudarjenje je še več. Ključno vodilo pri tej operaciji pa mora biti, da računalniški čas, porabljen za redukcijo podatkov, ne sme biti daljši od časa prihranjenega pri podatkovnem rudarjenju zaradi zmanjšanega obsega podatkov.

#### 2.3.4.1 Zmanjšanje dimenzionalnosti

##### 2.3.4.1.1 Diskretna valčna transformacija

Diskretna valčna transformacija je tehnika linearnega procesiranja signalov, ki preslika vektor $X$ v vektor valčnih koeficientov $X'$. Pri preslikavi se število elementov izvornega vektorja ohrani. Pri redukciji podatkov predstavlja vsaka *n*-terica podatkov en *n*-dimenzionalni vektor $X = (x_1, x_2, \dots x_n)$, kjer je $n$ število atributov podatkovnega zapisa (Han et al., 2011, str. 119).

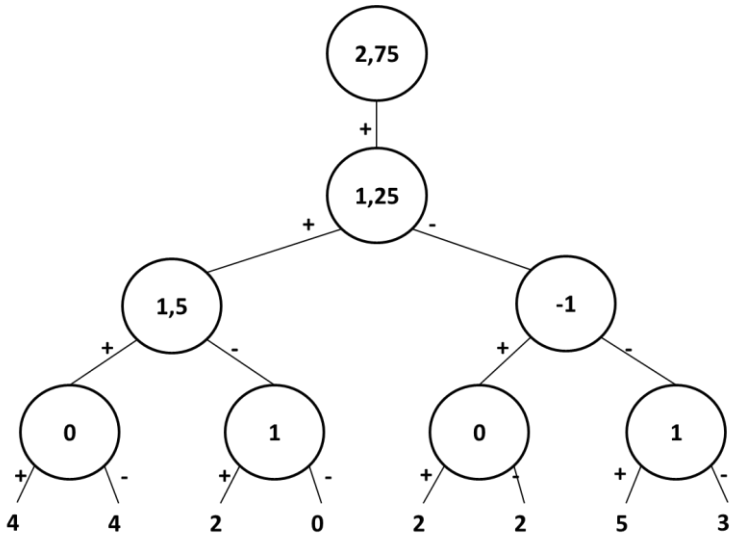
S samo transformacijo sicer števila dimenzij ne zmanjšamo, vendar lahko tako transformirane podatke odstranimo, tako da ohranimo le manjši del največjih vrednosti valčnih koeficientov. Vse druge, ki ne dosegajo določenega praga vrednosti preslikamo v vrednost 0. S tem se bistveno zmanjša število podatkov in posledično dimenzij, ki jih vključimo v postopke podatkovnega rudarjenja (Han et al., 2011, str. 119).

V praksi se uporablja več metod valčne transformacije, kot so na primer Haar-2, Daubechies-4 in Daubechies-6, v splošnem pa metode temeljijo na piramidnem algoritmu, ki v vsaki iteraciji prepolovi število podatkov (Han et al., 2011, str. 120).

Metoda z uporabo Haar-2 diskretne valčne transformacije (glej slika 7) poteka po naslednjem algoritmu (Han et al., 2011, str. 120):

1. Vektor, nad katerim izvajamo transformacijo, mora imeti število dimenzij oziroma atributov $L$, ki je potenca števila 2. V kolikor je število dimenzij v izvornih podatkih manj ($L \geq n$), se dodajo dodatni atributi z vrednostjo 0.
2. Vsaka transformacija vključuje dve funkciji. Prva je na primer vsota ali tehtano povprečje, ki je namenjena glajenju podatkov. Druga funkcija pa je tehtana razlika, ki izloči preveč podrobne značilnosti podatkov.
3. Obe funkciji se izračunata na parih vrednosti vektorja $X$, to je vrednostih $(x_{2i}, x_{2i+1})$. Kot rezultat dobimo dva vektorja dolžine $L/2$.
4. Postopek rekurzivno ponavljamo, dokler ni dolžina vektorja 2.
5. Izračunane vrednosti predstavljajo valčne koeficiente transformiranih podatkov.

*Slika 7: Primer diskretne valčne transformacije*




S pomočjo diskretne valčne transformacije se vektor $X = (4, 4, 2, 0, 2, 2, 5, 3)$ v prikazanem primeru preslika v vektor $X' = (2.75, 1.25, 1.5, -1, 0, 1, 0, 1)$.

Ta tehnika se lahko enako učinkovito uporabi tudi pri čiščenju podatkov.

2.3.4.1.2 Analiza glavnih komponent

Analiza glavnih komponent je statistična tehnika, pri kateri analiziramo soodvisnost atributov z namenom zmanjšanja njihovega števila. Pri tem osnovni nabor atributov preslikamo v nov nabor atributov, ki jih imenujemo glavne komponente (Košmelj, 2007, str. 159).

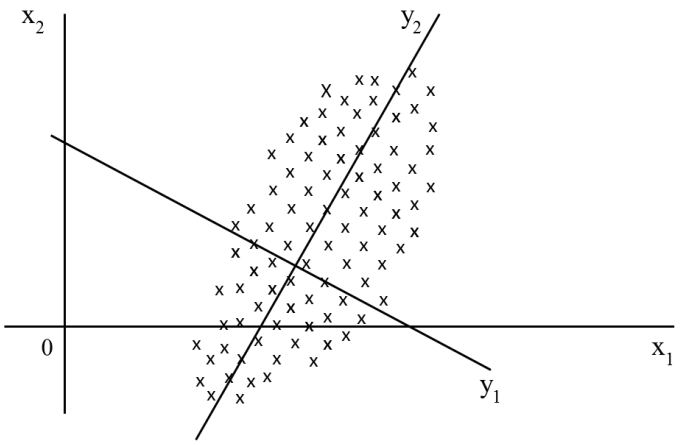
Glavnih komponent je toliko, kot je osnovnih atributov, in so med seboj neodvisne (pravokotne). Izražajo se kot linearna kombinacija osnovnih atributov in ohranjajo njihovo skupno variabilnost (Košmelj, 2007, str. 159).

Glavne komponente so urejene po pomembnosti, tj. padajoči velikosti variance, ki se izkazuje v tem, da prva glavna komponenta pojasnjuje kar se da velik del celotne variance osnovnih spremenljivk. Druga glavna komponenta je neodvisna od prve in pojasnjuje kar se da velik del še nepojasnjene variance. Tretja glavna komponenta je neodvisna od prve in od druge glavne komponente in pojasnjuje kar se da velik del še preostale nepojasnjene variance (Košmelj, 2007, str. 159).

Če so osnovne spremenljivke dovolj povezane, pojasnijo prve glavne komponente večji delež celotne variance. To pomeni, da lahko druge glavne komponente, ki sicer pojasnjujejo manjši del variance in so zaradi tega manj pomembne, zanemarimo. Bolj ko

so osnovni atributi med seboj povezani, bolj uspešna bo redukcija dimenzij. Kot mero povezanosti uporabimo koeficient kovariance oziroma korelacije, pri tem pa mora veljati, da je povezanost med spremenljivkami linearna (Košmelj, 2007, str. 159).

*Slika 8: Primer redukcija dimenzij z uporabo metode glavnih komponent*




*Vir: K. Košmelj, Metoda glavnih komponent: osnove in primer, Acta agriculturae Slovenica, 2007, str. 160.*

Primer (glej Sliko 8) prikazuje dvorazsežen prostor izhodiščnih atributov X1 in X2 ter preslikanih glavnih komponent Y1 in Y2. Iz diagrama je razvidno, da ker sta X1 in X2 zelo povezana, lahko Y1 uspešno nadomesti oba izhodiščna atributa X1 in X2. Dvorazsežni prostor s tem reduciramo v enorazsežnega, pri čemer je izguba informacije minimalna (Košmelj, 2007, str. 159).

Metodo glavnih komponent se lahko uporablja kot osnovo za multiplo regresijo (angl. *multiple regression*) in analizo gruč (angl. *cluster analysis*). V primerjavi z diskretno valčno transformacijo teži metoda glavnih komponent k boljšemu obravnavanju razpršenih podatkov, medtem ko je diskretna valčna transformacije bolj primerna za podatke z visoko stopnjo dimenzionalnosti (Han et al., 2011, str. 122).

#### 2.3.4.1.3 Izbor podmnožice atributov

Vzorci podatkov pogosto vsebujejo veliko količino atributov, ki so za izvedbo podatkovnega rudarjenja nepomembni ali redundantni. Na primer, pri prodaji zgoščenk je mogoče podatek, ali ima nekdo mobilni telefon, zelo nepomemben podatek, vsekakor manj pomemben, kot na primer starost ali glasbeni okus. Analitik z dobrim poznavanjem poslovnega področja bi sicer lahko izbral nekaj atributov, ki naj bi po njegovem mnenju bili pomembni za izvedbo podatkovnega rudarjenja. Vendar bi bil tak postopek v najboljšem primeru le časovno izjemno dolgotrajen, predvsem pa ne bi bil povsem prepričan, če ni izpustil katerega od pomembnejših atributov.

Po Han et al. (2011, str. 123) je cilj izbora podmnožice atributov poiskati minimalen nabor atributov, pri katerem bo verjetnostna porazdelitev podatkovnih razredov zmanjšanega nabora kar se da podobna verjetnostni porazdelitvi, če bi upoštevali vse atribute.

Problem vzorca podatkov z velikim številom atributov je v tem, da se lahko iskanje najboljše kombinacije atributov sprevrže v precej kompleksen in časovno potraten postopek (v primeru vzorca z $n$ atributi imamo $2^n$ kombinacij), ki lahko na koncu prinese več stroškov kot koristi. Zato se za iskanje najbolj optimalne kombinacije atributov uporablja požrešna metoda.

Požrešna metoda je ena od standardnih metod za načrtovanje algoritmov. Z njo rešujemo optimizacijske probleme. Pri požrešni metodi problem rešujemo kot končno zaporedje podproblemov. Pri vsakem koraku izberemo med delnimi rešitvami tisto, ki daje trenutno največji profit (Požrešna metoda, 2012).

Pri izboru najbolj optimalne kombinacije artiklov iščemo na vsakem koraku (podobno kot v primeru trgovskega potnika) najboljši atribut, ki ga ugotavljamo s testi statistične značilnosti, ki predpostavlja, da so atributi medsebojno neodvisni. Obstajajo tudi druge metode, ki temeljijo na ohranjanju informacijskega prispevka (angl. *information gain*) (Han, 2011, str. 123).

Najpogostejše tehnike preiskovanja po požrešni metodi so (Han et al., 2012, str. 124):

- Izbiranje naprej (angl. *stepwise forward selection*) je postopek, v katerem začnemo s prazno množico atributov, nato pa v vsakem koraku dodamo po en naključno izbran ali najbolje ocenjen atribut.
- Vzvratno odstranjevanje (angl. *stepwise backward elimination*) je obraten postopek od izbiranja naprej. To pomeni, da se postopek prične s polno množico atributov, nato pa se pri vsakem koraku izloči najslabše ocenjen atribut.
- Kombiniran pristop, pri katerem pričnemo s poljubno množico atributov, kateri dodajamo ali odstranjujemo atribute kot pri iskanju naprej oziroma vzvratnem iskanju.
- Pri indukciji z odločitvenimi drevesi kreira algoritem drevesno strukturo, v kateri vsako notranje vozlišče opisuje test nad posameznim atributom, vsak list drevesa pa predstavlja razred napovedi. Na vsakem vozlišču algoritem izbere najboljši atribut za razdelitev podatkov v razrede. V tem primeru se odločitveno drevo kreira na osnovi razpoložljivih podatkov. Vsi atributi, ki jih algoritem ne vključi v odločitveno drevo, za izbor niso pomembni in se jih lahko izpusti.

#### 2.3.4.2 Zmanjšanje številčnosti

##### 2.3.4.2.1 Regresijski in loglinearni model

Regresijski in loglinearni model sta primera parametrskih modelov za zmanjšanje številčnosti. V primeru linearne regresije modeliramo odvisno spremenljivko $y$ v odvisnosti od naključne spremenljivke $x$ (prediktor) z uporabo linearne enačbe:

$$y = ax + b.$$

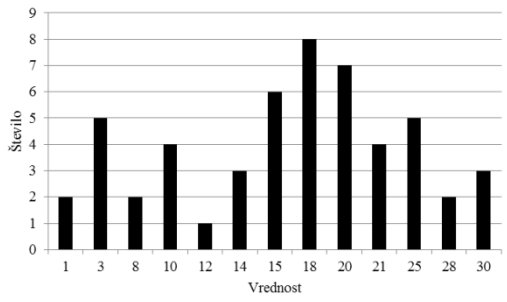
V kontekstu podatkovnega rudarjenja sta $x$ in $y$ vrednosti numeričnih atributov v bazi podatkov. Vrednosti $a$ in $b$, regresijska koeficienta, določata naklon premice in vrednost, kjer ta seka ordinatno os. Na ta način lahko izračunamo za vsako napovedno spremenljivko odvisno spremenljivko. Multipla linearna regresija je razširitev linearne regresije, v kateri vrednost odvisne spremenljivke določa več prediktorjev (Han v Chakrabrati et al., 2009, str. 92).

Loglinearni modeli omogočajo približno ocenitev diskretne večdimenzionalne verjetnostne porazdelitve. Pri zapisih *n*-teric podatkov lahko vsako *n*-terico razumemo kot točko v *n*-razsežnostnem prostoru. Loglinearne modele se lahko uporabi za oceno verjetnosti vsake točke v večdimenzionalnem prostoru za nabor diskretnih atributov, na osnovi manjše podmnožice atributov. Na ta način lahko večrazsežnostni prostor nadomestimo z manj razsežnim, kar z drugimi besedami pomeni, da zmanjšamo številčnost atributov (Han v Chakrabrati et al., 2009, str. 93).

##### 2.3.4.2.2 Uporaba histogramov za zmanjšanje številčnosti

Histograme se poleg grafičnega opisovanja podatkov uporablja tudi v primeru zmanjšanja številčnosti podatkov. Histogram atributa (glej Sliko 9) porazdeli vrednosti atributa v ločene podmnožice, grupe (angl. *buckets* ali *bins*).

*Slika 9: Histogram posameznih vrednosti*

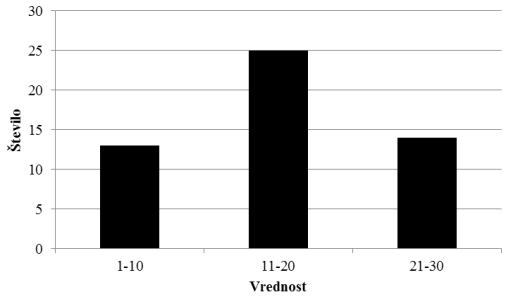



*Vir: S.Chakrabrati et al., Data Mining: Know it all, 2009, str. 94.*

Za razvrščanje v grupe se uporablja več pravil, med katerimi so najpogostejša naslednja (Han v Chakrabrati et al., 2009, str. 93–94):

- Razvrščanje na osnovi enake širine intervala (glej Sliko 10) pomeni, da celotno zalogo vrednosti atributa razdelimo na končno število intervalov enake širine.

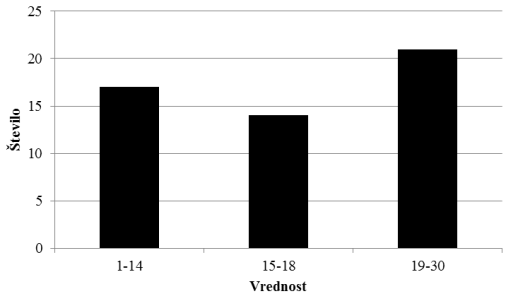
*Slika 10: Histogram z razvrščanjem na osnovi enake širine intervala*




*Vir: S.Chakrabrati et al., Data Mining: Know it all, 2009, str. 94.*

- Pri razvrščanju na osnovi enake frekvence (glej Sliko 11) se vrednosti atributa porazdelijo v grupe tako, da je v vsaki grupi približno enako število elementov.

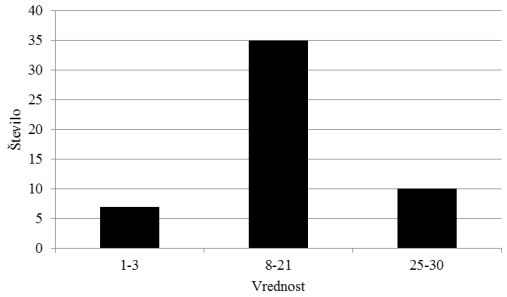
*Slika 11: Histogram z razvrščanjem na osnovi enake frekvence*




- V-optimalno razvrščanje. Če upoštevamo vse možne histograme za določeno število grup, je V-optimalni histogram tisti z najnižjo varianco. Varianca histograma je enaka tehtani vsoti izvornih vrednosti vsake grupe, pri čemer je utež enaka številu vrednosti v posamezni grupi.

- MaxDiff histogram (glej Sliko 12). Če uredimo podatke po velikosti in če jih želimo razdeliti na *N* grup, potem so meje posameznih grup določene tako, da je razlika robnih vrednosti posameznih grup največja.

*Slika 12: MaxDiff histogram*



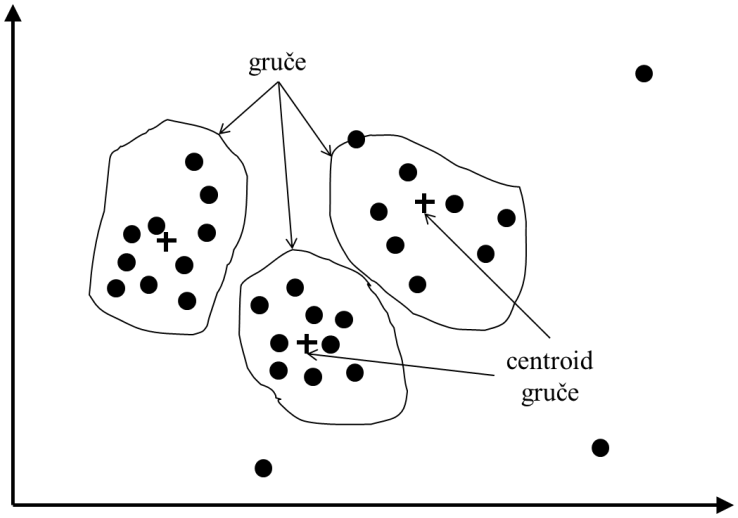

*Vir: S.Chakrabrati et al., Data Mining: Know it all, 2009, str. 94.*

V analizi podatkov se pogosto uporablja tudi večdimenzionalne histograme, ki pa se pokažejo še uporabni, če je analiza zajema do 5 atributov.

2.3.4.2.3 Gručenje

Pri gručenju se *n*-terice podatkov obravnavajo kot objekti, ki so v posamezne gruče razporejeni na osnovi medsebojnih podobnosti (glej Sliko 13). Podobnost se opredeljuje na osnovi bližine, ki se izračunava kot funkcija razdalje. Kakovost posamezne gruče se določi na osnovi premera, ki se izračuna na osnovi največje razdalje med dvema objektoma v gruči. Srednja razdalja je druga mera, ki določa kakovost gruče, izračuna pa se jo kot povprečno razdaljo vseh objektov v gruči od središčne točke (centroida) gruče.

*Slika 13: Gruče in centroidi gruč*




*Vir: S.Chakrabrati et al, Data Mining: Know it all, 2009, str. 75.*

V procesu podatkovnega rudarjenja je gručenje podatkov večkrat uporabno. V primeru zmanjšanja številčnosti se dejanske vrednosti podatkov zamenjajo z vrednostmi, ki predstavljajo posemezne gruče (Han v Chakrabarti et al., 2009, str. 95).

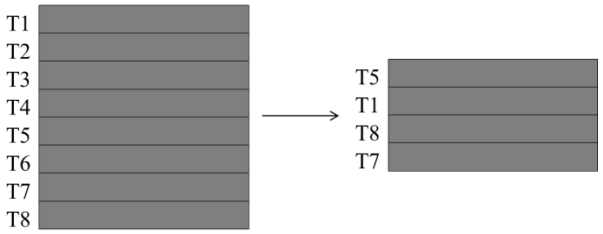
2.3.4.2.4 Vzorčenje

Vzorčenje se pogosto uporablja kot tehnika redukcije podatkov, saj omogoča, da predstavimo večje vzorce podatkov z manjšimi vzorci oziroma podmnožicami podatkov (Han et al., 2011, str. 127). Vendar pri vzorčenju ne gre le za zmanjšanje števila podatkov, temveč za to, da si izberemo takšno podmnožico, ki ustrezno predstavlja celoten vzorec. V nasprotnem primeru so lahko rezultati algoritma nepravilni in neuporabni.

Han et al. (2011, str. 127) navajajo nekaj najpogostejših metod vzorčenja:

- Enostavno naključno vzorčenje brez zamenjevanja (glej Sliko 14). Iz množice $N$ vzorcev naključno izberemo $s$ $n$-teric podatkov. Vsako izbrano $n$-terico, ko je izbrana, izločimo iz vzorca. Verjetnost, da iz množice izberemo posamezno $n$-terico, se z vsakim izborom povečuje $\frac{1}{N}, \frac{1}{N-1}, \ldots \frac{1}{N-s}$.

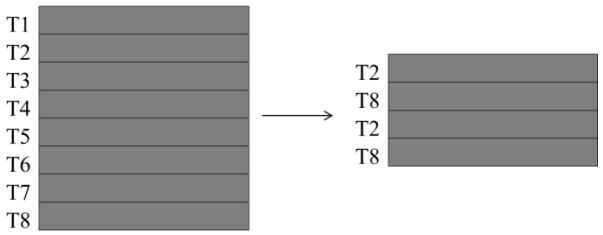
*Slika 14: Enostavno naključno vzorčenje brez zamenjevanja*




*Vir: J. Han et al., Data Mining: Concepts and Techniques (3rd ed.), 2011, str. 127.*

- Enostavno naključno vzorčenje z zamenjevanjem (glej Sliko 15). Tudi pri tej metodi vzorčenja gre za naključno vzorčenje, pri čemer se izbrana $n$-terica ne izloči iz vzorca. Verjetnost, da bo posamezna $n$-terica izbrana, je enaka $\frac{1}{N}$.

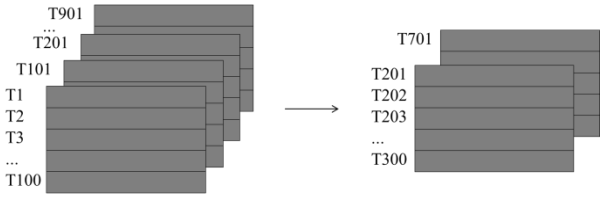
*Slika 15: Enostavno naključno vzorčenje z zamenjevanjem*




*Vir: J. Han et al, Data Mining: Concepts and Techniques (3rd ed.), 2011, str. 127.*

- Vzorčenje v skupinicah (glej Sliko 16). Pri tem vzorčenju celoten vzorec podatkov razdelimo na $S$ skupinic. Nato celotne skupinice izbiramo z uporabo ene od metod enostavnega naključnega vzorčenja.

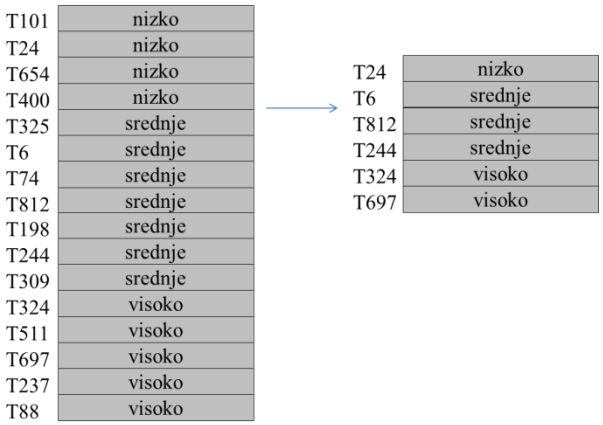
*Slika 16: Vzorčenje v skupinicah*




*Vir: J. Han et al., Data Mining: Concepts and Techniques (3rd ed.), 2011, str. 127.*

- Stratificirano vzorčenje (glej Sliko 17). Celoten vzorec podatkov v primeru stratificiranega vzorčenja razdelimo na medsebojno izključujoče se skupine podatkov, na primer, podatke o kupcih razdelimo na skupine glede na starost kupca. Tako pridobljene skupine *n*-teric podatkov vzorčimo z uporabo ene od metod enostavnega naključnega vzorčenja. Na ta način dobimo v reprezentativnem vzorcu tudi podatke, ki pripadajo skupinam z najmanj podatki.

*Slika 17: Stratificirano vzorčenje*




*Vir: J. Han et al., Data Mining: Concepts and Techniques (3rd ed.), 2011, str. 127.*

Z uporabo centralnega limitnega izreka, ki pravi, da je vsota (neodvisnih) vrednosti poljubno porazdeljene slučajne spremenljivke približno normalno porazdeljena oziroma čim več vrednosti seštejemo, tem bolj se porazdelitev vsote približuje normalni porazdelitvi (Batagelj, 2012), lahko določimo dovolj veliko velikost vzorca. Ta je lahko veliko manjša od velikosti celotnega vzorca podatkov $N$.

#### 2.3.4.2.5 Agregacija podatkov v podatkovnih kockah

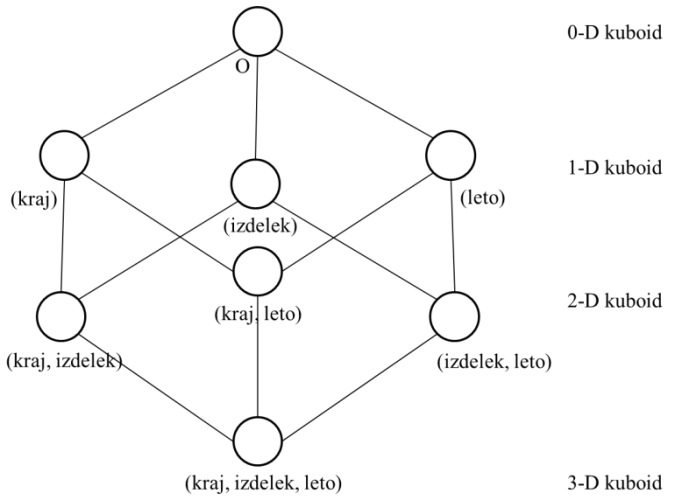
Numerični podatki, na primer o prodaji, so velikokrat na voljo v obliki transakcijskih podatkov. Takšni podatki velikokrat niso najbolj primerni za podatkovno rudarjenje. Po drugi strani so isti podatki, le da so agregirani na nekem višjem nivoju (na primer po

kupcih na nivoju skupine izdelkov in meseca) bolj primerni. Tehnologija za takšno agregacijo so večdimenzionalne podatkovne kocke.

Podatki so v večdimenzionalnih podatkovnih kockah shranjeni v obliki, ki je primerna za poslovno poročanje in v obliki agregatov na različnih ravneh. V okviru vsake kocke lahko v odvisnosti od upoštevanja podmnožice vseh dimenzij prikažemo kombinacijo agregiranih podatkov na določeni ravni – kuboidu. Če imamo v kocki $n$ dimenzij, je mogočih $2^n$ kuboidov, in če ima $i$-ta dimenzija $L_i$ nivojev, je mogočih $\prod_{i=1}^{n}(L_i + 1)$ kuboidov (Jaklič, 2010, str. 283).

Podatkovne kocke omogočajo zelo hiter dostop do že izračunanih podatkov, ki se nahajajo na različnih ravneh (glej Sliko 18). Vsaka celica v kocki ustreza določeni vrednosti v večdimenzionalnem prostoru. Za vsak atribut v kocki, praviloma obstaja konceptualna hierarhija, ki omogoča analizo podatkov na različnih nivojih abstrakcije. Vsak višji nivo abstrakcije v okviru konceptualne hierarhije vpliva na številčnost podatkov v okviru atributa, ki ga lahko vključimo v podatkovno rudarjenje. Praviloma pri pripravi podatkov podatkovnega rudarjenja uporabimo najmanjši možni kuboid (Han et al., 2011, str. 129).

*Slika 18: Ravni agregacije v podatkovni kocki*




*Vir: J. Jaklič, Poslovna inteligenca [prosojnice], 2010, str 283.*

### 2.3.5 Transformacija in diskretizacija podatkov

Pyle (Chakrabarti et al., 2009, str. 348) navaja tri tehnike transformacij in diskretizacij podatkov:

- razvrščanje v grupe,
- normalizacija obsega vrednosti in
- normalizacija distribucije vrednosti.

#### 2.3.5.1 Razvrščanje v grupe

Razvrščanje v grupe (angl. *binning*) je tehnika združevanja vrednosti numeričnih zveznih atributov v kategorične ali ordinalne, z namenom zmanjšanja variabilnosti v podatkih, pa tudi zaradi specifičnih zahtev posameznih algoritmov podatkovnega rudarjenja, ki zahtevajo kategorične ali ordinalne vrednosti atributov namesto numeričnih. Kot primera takšnih algoritmov lahko navedemo odločitvena drevesa ali naivne Bayesove mreže (Pyle, 1999, str. 103; Chakrabarti et al., 2009, str. 348).

Pri razvrščanju v grupe se celotna zaloga vrednosti razdeli na posamezne grupe, pri čemer se oznake posameznih grup uporabi kot nadomestke originalnih vrednosti. To ni posebno zahteven postopek, saj gre za način razvrščanja v grupe, ki ga uporabljamo v vsakodnevnem življenju. Na primer, temperaturo vode lahko označimo z naslednjimi grupami: vrelo, vroče, toplo, hladno in ledeno. Vsaka od navedenih oznak označuje temperature vode v določenem razponu. Kako določiti te razpone, je praviloma v domeni strokovne presoje, v kolikor pa to ni mogoče, se uporablja takšno razvrstitev v grupe tako, da imajo približno enako število članov (Pyle, 1999, str. 328).

Razvrščanje v grupe je predvsem intuitiven proces, pri katerem Pyle (Chakrabarti et al., 2009, str. 348) opozarja na dejstvo, da kadarkoli razvrstimo v grupe neko zvezno numerično vrednost, se v tem procesu informacijska vrednost te vrednosti nepovratno izgubi. A po drugi strani pomeni to kompromis z uporabnostjo te vrednosti v algoritmu podatkovnega rudarjenja. Z razvrščanjem v grupe se namreč nemalokrat odstrani tudi prevečkrat moteč šum v podatkih, ki se s tem zgladijo, model pa je zaradi tega bolj uporaben, kot bi bil sicer, če podatkov ne bi razvrstili v grupe in bi jih uporabili v originalni obliki.

Pri razvrščanju v grupe vidi Pyle (Chakrabarti et al., 2009, str. 348) dva problema:

1. Na koliko grup naj razdelimo celotno zalogo vrednosti vzorca?
2. Kako je najbolje dodeliti vrednosti posamezni grupi?

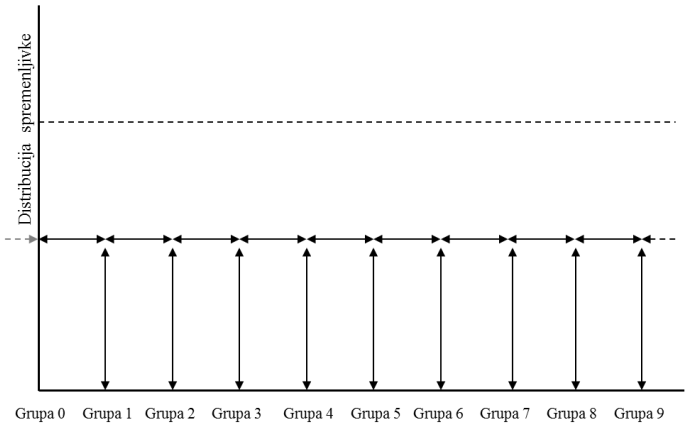
Najbolj splošni metodi razvrščanja v grupe, ki jih podpira tudi večina orodij za podatkovno rudarjenje, sta razvrščanje na osnovi enako velikih grup in na osnovi enake frekvence vrednosti.

##### 2.3.5.1.1 Razvrščanje na osnovi enakomernega obsega

Razvrščanje na osnovi enakomernega obsega pomeni, da celotno zalogo vrednosti atributa razdelimo na grupe z enakim intervalom vrednosti.

Če želimo celotno zalogo vrednosti atributa razdeliti na 10 grup, jo razdelimo tako, da v 8 grupah preslikamo vse znane vrednosti atributa, dve robni grupi pa sta namenjeni za vrednosti, ki bi morebiti kasneje izstopale iz učne množice vrednosti (glej Sliko 19).

*Slika 19: Razvrščanje v grupe na osnovi enakomernega obsega*




*Vir: S.Chakrabrati et al., Data Mining: Know it all, 2009, str. 350.*

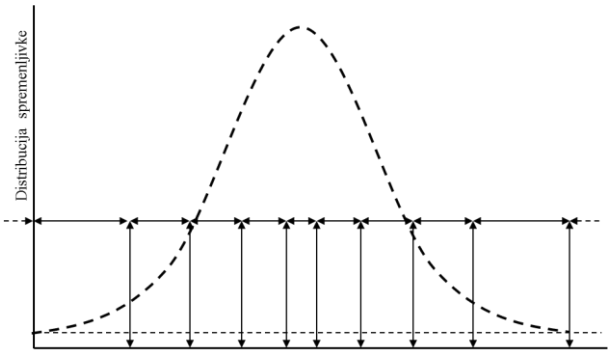
V grupe 1–8 se vrednosti razporedijo glede na mejne vrednosti grup, pri čemer vsaka grupa predstavlja 12,5 % vrednosti celotnega razpona. Grupi 0 in 9 sta namenjeni vrednostim, ki so bodisi manjše od najmanjše vrednosti v učni množici, bodisi so večje od največje vrednosti učne množice.

Takšna metoda razvrščanja v grupe deluje dobro , ko imamo opraviti z uniformno porazdelitvijo v učni množici, slabše pa se obnese v primeru, ko porazdelitev ni uniformna.

2.3.5.1.2 Razvrščanje na osnovi enakomerne porazdelitve

V primeru neuniformnih, na primer normalnih porazdelitev vrednosti, vrednosti niso enakomerno porazdeljene preko celotne zaloge vrednosti (glej Sliko 20).

*Slika 20: Razvrščanje na osnovi enakomerne porazdelitve*




*Vir: S.Chakrabrati et al, Data Mining: Know it all, 2009, str. 351.*

Kot je pričakovati, bodo zgoščene okrog povprečne vrednosti, pogostnost pojavljanja posameznih vrednosti pa pada z oddaljenostjo od povprečne vrednosti. Če bi uporabili razvrščanje na osnovi enakomernega obsega, bi se v grupo, ki bi vključevala povprečno vrednost, razvrstilo največ vrednosti, medtem ko bi se v grupe, ki bi bile najbolj oddaljene od povprečja, razporedilo le nekaj vrednosti (Chakrabarti et al., 2009, str. 350–351).

Z razvrščanjem vrednosti v grupe na osnovi enakomerne porazdelitve dosežemo, da ima vsaka grupa približno enako število elementov, kar se v praksi izkaže kot najbolj primerna oblika razvrščanja v grupe.

#### 2.3.5.1.3 Število grup pri razvrščanju v grupe

Ne glede na to, kakšno metodo razvrščanja v grupe izberemo, optimalnega števila grup ne moremo določiti s pomočjo kakšnega algoritma. Pyle (Chakrabarti et al., 2009, str. 351) navaja, da je običajno od 20 do 30 grup dobra štartna osnova, pri podrobnejši oceni potrebnega števila grup pa naj bo ključno vodilo, da je potrebno toliko grup, da bo lahko orodje, s katerim izvajamo podatkovno rudarjenje, razvilo dovolj kompleksne vzorce. Svetuje tudi (Chakrabarti et al., 2009, str. 351), da se primerno število grup poskuša ugotoviti s poskušanjem.

#### 2.3.5.1.4 Razvrščanje v grupe na osnovi vrednosti informacije

Pyle (Chakrabarti et al., 2009, str. 351) poleg opisanega navedenega nenadzorovanega razvrščanja opisuje še en način razvrščanja v grupe, ki temelji na nadzorovanem razvrščanju v grupe.

Izhodišče za takšno razvrščanje je vrednost informacije vsebovana v vhodni in izhodni množici podatkov. Vrednost informacije je mogoče izračunati s pomočjo teorije informacij. Predpostavka pri tem je, da je mogoče izdelati takšno strategijo razvrščanja v grupe, po kateri spremenljivka v izhodni množici ohrani kar se da veliko informacije spremenljivke v vhodni množici. Pyle (Chakrabarti et al., 2009, str. 352) navaja dve možni strategiji razvrščanja:

- razvrščanje z najmanjšo izgubo vrednosti informacije in
- razvrščanje z največjo pridobitvijo vrednosti informacije.

Obe strategiji sta izredno močni strategiji razvrščanja v grupe, vendar iz praktičnih razlogov nista preveč uporabni. Ob zelo veliki kompleksnosti izračunavanj je namreč potrebna zelo velika procesorska moč, zaradi česar tudi ni veliko orodij, ki bi omogočala takšno funkcionalnost.

#### 2.3.5.2 Normalizacija vrednosti

Normalizacija podatkov pride v poštev v primeru numeričnih atributov, saj z normalizacijo preslikamo vrednosti v območje med 0 in 1 oziroma med –1 in +1. To je potrebno zaradi tega, ker nekateri algoritmi, kot so na primer nevronske mreže, pričakujejo le vrednosti v tem območju. Z normalizacijo lahko razrešimo tudi probleme povezane z izstopajočimi vrednostmi.

Za normalizacijo podatkov lahko uporabimo več metod. Med pogosteje uporabljenimi metodami najdemo normalizacijo s povečevanjem decimalnih mest, Min-Max normalizacijo, normalizacijo na podlagi z-vrednosti in normalizacijo na podlagi softmax povečevanja.

##### 2.3.5.2.1 Normalizacija s povečevanjem števila decimalnih mest

Pri normalizaciji s povečevanjem števila decimalnih mest (angl. *decimal scaling)* so originalne vrednosti atributa *V* transformirane s premikom decimalne vejice, pri čemer je število decimalnih mest odvisno od največje absolutne vrednosti atributa *V* (Kantardžić, 2003; Han et al., 2011, str. 133). Z normalizacijo s povečevanjem decimalnih mest se preslikajo vrednosti v območje intervala [-1..1].

Transformirano vrednost $v_i'$ izračunamo z uporabo naslednje enačbe

$$v_i' = \frac{v_i}{10^k},$$

pri čemer velja, da je $k$ najmanjše naravno število, pri katerem še velja $max(|v_i'|) < 1$.

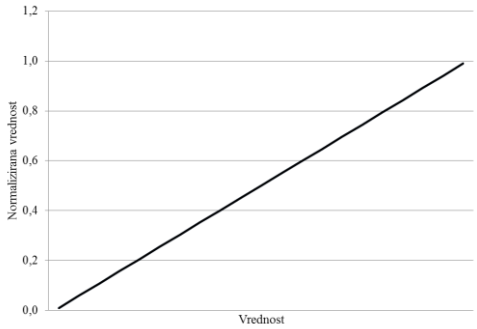
##### 2.3.5.2.2 Min-Max normalizacija

S pomočjo Min-Max normalizacije izvedemo linearno transformacijo nad osnovnimi podatki končne množice števil. Recimo, da imamo v zalogi vrednosti atributa *V* števila {$v_1$, $v_2$, ...$v_n$}, od katerih predstavlja $min_v$, najnižjo in $max_v$, najvišjo vrednost.

Min-Max normalizacija preslika vrednost $v_i$ v vrednost $v_i'$, ki leži v območju vrednosti $[min_v', max_v']$ (Han et al., 2011, str. 132):

$$v_i' = \frac{v_i - min_v}{max_v - min_v} * (max_{v'} - min_{v'}) + min_{v'}.$$

*Slika 21: Linearna preslikava vrednosti v normalizirano vrednost*




*Vir: D. Pyle, Data Preparation for Data Mining, 1999, str. 237.*

Min-Max normalizacija je linearna transformacija (glej Sliko 21), s katero ohranimo razmerja med števili v množici in ki omogoča enolično povratno transformacijo normaliziranih vrednosti v izvorne.

Vendar ta transformacija deluje le v primeru, da osnovni učni, vzorec vključuje tudi najvišjo in najnižjo možno vrednost, ki bo vključena v model podatkovnega rudarjenja. Če se v kasnejšem vzorcu pojavi vrednost $u$, za katero velja $u < min_v$ ali $u > max_v$, potem bo normalizirana vrednost izven intervala [0, 1]. V tem primeru bo v algoritmu podatkovnega rudarjenja, ki pričakuje le vrednosti med 0 in 1, prišlo do napake, ki lahko bistveno vpliva na rezultate podatkovnega rudarjenja.

2.3.5.2.3 Normalizacija na podlagi standardne oziroma z-vrednosti

Standardno ali z-vrednost (angl. *z-score*) $v_i'$ izračunamo s pomočjo povprečne vrednosti ($\bar{V}$) in standardnega odklona ($\sigma_v$) vrednosti atributa *V* (Han et al., 2011, str. 132):

$$v_i' = \frac{v_i - \bar{V}}{\sigma_v}.$$

V primerjavi z Min-Max normalizacijo je normalizacija na podlagi standardne vrednosti uporabnejša predvsem v primerih, ko najvišja in najnižja vrednost atributa ni znana, ali če vemo, da bodo nastopile izstopajoče vrednosti, ki lahko bistveno vplivajo na Min-Max normalizacijo.

2.3.5.2.4 Normalizacija na podlagi softmax povečevanja

Normalizacija na podlagi softmax povečevanja (angl. *softmax scaling)* temelji na hkratni uporabi linearne transformacije in transformacije z uporabo logistične funkcije ter v celoti rešuje problem izstopajočih vrednosti (Pyle, 1999, str. 237–242).

Po izvedeni normalizaciji morajo biti vse vrednosti v intervalu [0, 1]. Problema izstopajočih vrednosti lahko razrešimo na več načinov. Min-Max normalizacija omogoča preslikavo vrednosti, ki se lahko preslikajo izven zahtevanega intervala, na primer v -0,4 ali 1,3. V tem primeru se moramo odločiti, kaj storiti z vrednostmi izven območja normalizacije.

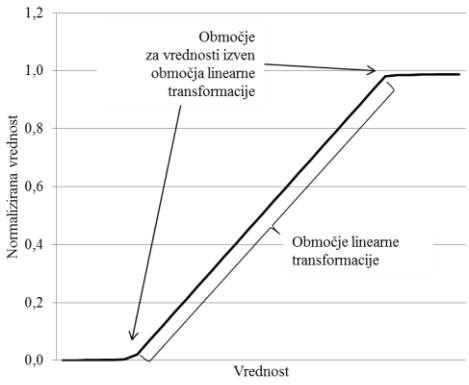
Najbolj enostavno bi bilo, da vrednosti območja normalizacije v modelu ne upoštevamo oziroma jih iz modela izločimo. Ta metoda je zagotovo najenostavnejša, a po drugi strani pomeni veliko tveganje, da bodo izločeni podatki bistveno vplivali na stopnjo zaupanja vzorca, ki naj bi predstavljal celotno populacijo ter posledično na model in rezultate procesa podatkovnega rudarjenja. Po drugi strani lahko z izločanjem podatkov nenamerno odstranimo obstoječe skrite vzorce v podatkih.

Druga možnost je, da se vrednostim izven območja normalizacije pripišejo vrednosti 0 in 1. Tudi v tem primeru imamo podobne probleme kot v primeru izločanja vrednosti.

Linearna preslikava tipa Min-Max nam z določeno stopnjo zaupanja zagotavlja, da bo v območje zaloge vrednosti [0, 1] preslikana večina vseh vrednosti atributa *V*. Končno območje normalizacije, vrednosti intervala [0, 1], bomo zato zmanjšali na obseg, ki ga opredeljuje stopnja zaupanja, ter v to zmanjšano območje linearno preslikali vse znane vrednosti atributa *V*.

Če je stopnja zaupanja 98 %, se območje intervala [0, 1] enakomerno zmanjša za 2 % (po 1 % na obeh straneh intervala), s čimer bodo vse vrednosti atributa V linearno preslikane v območje intervala [0,01, 0,99] (Pyle, 1999, str. 238).

*Slika 22: Prilagojena funkcija z upoštevanjem vrednosti izven območja transformacije*




*Vir: D. Pyle, Data preparation for Data Mining, 1999, str. 241.*

Vse izstopajoče vrednosti (glej Sliko 22), ki jih v trenutku izdelave modela podatkovnega rudarjenja ni v intervalu $[min_v, max_v]$**,** bomo preslikovali v območje vrednosti [0, 0,01] in

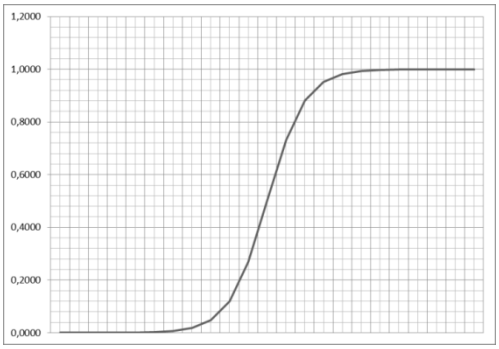
[0,99, 1]. Za te preslikave ne moremo več uporabiti enostavne linearne preslikave, temveč preslikavo, ki bo omogočila preslikavo še tako zelo izstopajočih vrednosti v vrednost, ki bo le malo odstopala od vrednosti 0,01 ali 0,99 in ne bo nikoli dosegla robnih vrednosti intervala [0, 0,01] oziroma [0,99, 1] (Pyle, 1999, str. 239–241).

Softmax povečevanje, ki temelji na uporabi logistične funkcije (Pyle, 1999, str. 253–257), zagotavlja potrebno funkcionalnost, ki preslika vrednosti [-∞,∞] v vrednosti intervala [0, 1]:

$$v_n = \frac{1}{1 + e^{-v_i}}.$$

Pri tem je $v_n$ normalizirana vrednost vrednosti $v_i$, krivulja (glej Sliko 23) pa ima obliko sploščenega S. Ključna značilnost softmax povečevanja je, da ne glede na vrednost $v_i$ v nobenem primeru preslikana vrednost ne doseže robnih vrednosti intervala [0, 1].

*Slika 23: Krivulja logistične funkcije*




*Vir: D. Pyle, Data preparation for Data Mining, 1999, str. 255.*

Logistična krivulja ima potrebno S obliko, vendar ne preko celotnega želenega območja. Prav tako ne moremo določiti območja, ki ga bomo pokrili z linearno preslikavo. Zato, in da bi dobili kar se da dobro preslikavo, nadomestimo vrednost $v_i$ v enačbi logistične funkcije z izrazom

$$v_t = \frac{(v_i - \bar{V})}{\lambda(\sigma_V / 2\pi)},$$

kjer $v_t$ v enačbi pomeni transformirano vrednost $v_i$, $\lambda$ pa odstotek linearnega dela krivulj, ki ga opredeljuje standardni odklon od povprečne vrednosti ($\pm 1\sigma_V$ pokriva 68 % celotne krivulje normalne porazdelitve, $\pm 2\sigma_V$ pokriva 95,5 %, $\pm 3\sigma_V$ 99,3 %, itn.) (Pyle, 1999, str. 253–257).

#### 2.3.5.3 Normalizacija porazdelitve

Ko govorimo o terminu normalizacija porazdelitve, to ne pomeni, da poskušamo neko določeno porazdelitev preslikati tako, da je po preslikavi podobna normalni porazdelitvi, temveč gre v vsebinskem pomenu za preslikavo, ki vrednosti vzorca preslika tako, da so kar se da enakomerno porazdeljene (Chakrabarti et al., 2009, str. 355).

Normalizacija distribucije praviloma velja le za numerične vrednosti. Sicer zelo poredko, pa vendar, se jo uporablja tudi v primeru kategoričnih in ordinalnih vrednosti (Chakrabarti et al., 2009, str. 355).

V praksi se precej uspešno izkaže razvrščanje na osnovi enakomernega obsega, če razdelimo celoten vzorec na zelo veliko število (na primer 101) gruč. V tem primeru vsaki od gruč dodelimo njen enakomerni del celotne zaloge vrednosti vzorca. Kadar so vrednosti vzorca med 0 in 1, razdelimo celoten vzorec na 101 gručo, ki jim dodelimo vrednosti 0, 0,01, 0,02, ..., 1,00 (Chakrabarti et al., 2009, str. 356).

Razvrščanje na osnovi enakomerne porazdelitve je ena od pogosteje uporabljenih tehnik za normalizacijo porazdelitve, saj ta tehnika enakomerno porazdeli vse vrednosti vzorca v gruče, ki vsebujejo (približno) enako število elementov.

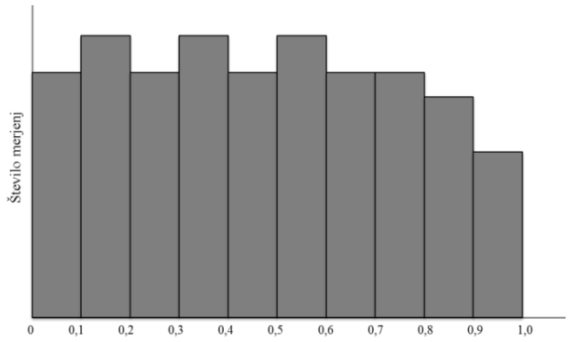
Primer: V vzorcu 1, 2, 3, 4, 5, 6, 7, 8, 9, 1000 imamo opraviti z vrednostmi v intervalu [1, 1000]. Prvih 9 vrednosti leži v okviru 1 % celotnega obsega intervala, medtem ko je vrednost 1000 izstopajoča vrednost, ki je lahko povsem realna vrednost. Če zdaj celoten vzorec razvrstimo na osnovi enakomernega obsega v 101 gručo, potem prva gruča vsebuje vse vrednosti, razen najvišje, ki bo razporejena v 101. gručo. Algoritmi podatkovnega rudarjenja bodo v tem primeru uporabili le dve vrednosti. Pri tem je vrednost informacije vsebovana v vrednostih od 1 do 9 izgubljena. Slednje se v primeru razporejanja v gruče na osnovi enakomerne porazdelitve ne more zgoditi (Chakrabarti et al., 2009, str. 356).

Opisani primer kaže na to, da brez uporabe razporejanja v gruče ali kakšne druge strategije normalizacije porazdelitve, numerično občutljivi algoritmi ne bodo razlikovali vhodnega vzorca od vzorca, ki vsebuje le nekaj vrednosti (Chakrabarti et al., 2009, str. 357).

Kot tretjo možnost omenja Pyle (Chakrabarti et al., 2009, str. 356), zvezno preslikavo vrednosti katerekoli porazdelitve, ki te vrednosti preslika v vrednosti enakomerne porazdelitve preko celotne zaloge vrednosti vzorca. Vsaka vrednost se preslika v drugo vrednost, ne v gručo, ki bi si jo delila z drugimi vrednostmi. Pri tem se informacija posamezne vrednosti ne izgublja kot v primeru gruč, kjer vsi elementi gruče privzamejo novo vrednost, ki je ista za vse elemente posamezne gruče.

Primer (Pyle, 1999, str. 246): Histogram na primeru vzorca podatkov prikazuje porazdelitev (glej Sliko 24), v kateri vsak stolpec predstavlja 10 % celotnega intervala zaloge vrednosti atributa in prikazuje število pojavitev vrednosti v posameznem intervalu.

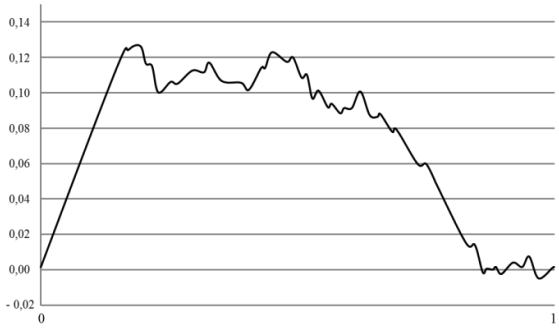
*Slika 24: Histogram distribucije vrednosti vzorca*




*Vir: D. Pyle, Data preparation for Data Mining, 1999, str. 246.*

Da dobimo enakomerno zvezno porazdelitev preko celotnega intervala zaloge vrednosti, je potrebno vsako vrednost preslikati s pomočjo premika vrednosti v novo vrednost. Spodnji diagram (glej Sliko 25) prikazuje primer premikov za vsako od posameznih vrednosti v vzorcu.

*Slika 25: Premiki vrednosti vzorca zaradi normalizacije distribucije vzorca*

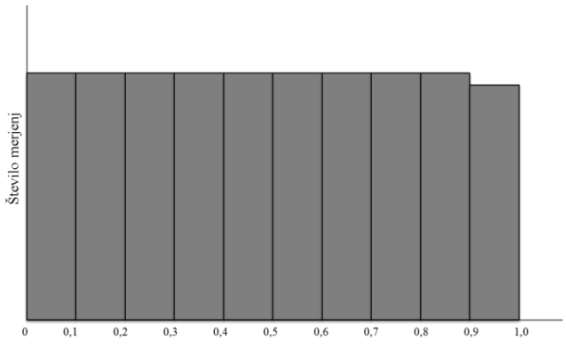



*Vir: D. Pyle, Data preparation for Data Mining, 1999, str. 247.*

Nekatere vrednosti se bodo tako premaknile proti vrednosti 0, druge proti vrednosti 1. V prikazanem primeru bodo skoraj vse vrednosti povečane, s čimer dobimo novo porazdelitev, ki je prikazana na Sliki 26.

Porazdelitev vrednosti atributa po redistribuciji je skoraj popolnoma pravokotna, v vsakem intervalu porazdelitve je skoraj enako število elementov.

*Slika 26: Histogram normalizirane porazdelitve vrednosti vzorca*




*Vir: D. Pyle, Data preparation for Data Mining, 1999, str. 247.*

# 3 PRIPRAVA PODATKOV V ORACLE OKOLJU

V splošnem velja, da večina orodij za podatkovno rudarjenje, in tu Oracle orodja niso prav nobena posebnost, pričakuje, da so podatki pripravljeni v obliki, ki je primerna za izvedbo podatkovnega rudarjenja (Linoff & Berry, 2011, str. 557):

- Vsi podatki morajo biti v eni tabeli.
- Vsaka vrstica v tabeli ustreza entiteti iz realnega svet (na primer kupec), ki je pomembna za poslovanje.
- Stolpce z eno vrednostjo se zanemari.
- Stolpce z vsemi različnimi vrednostmi se zanemari, četudi je informacija izvedena iz drugih stolpcev.
- Vse sopomenske stolpce se odstrani.
- Za napovedne modele je potrebno identificirati ciljni stolpec.

## 3.1 Funkcije in algoritmi podatkovnega rudarjenja

Ko govorimo o podatkovnem rudarjenju, je potrebno razločevati med dvema pojmoma, ki se ju pogosto zamenjuje, in sicer med funkcijami in algoritmi podatkovnega rudarjenja. Ko govorimo o funkcijah podatkovnega rudarjenja, te pomenijo skupek problemov, ki jih želimo rešiti s pomočjo algoritmov podatkovnega rudarjenja. Ko vzpostavljamo model podatkovnega rudarjenja, se moramo najprej odločiti za funkcijo podatkovnega rudarjenja, šele nato izberemo ustrezen algoritem, ki implementira izbrano funkcijo, razen če ta že izvorno ne opredeljuje samega algoritma (Oracle, 2008, str. 4/1–9/8).

Podatkovno rudarjenje v Oracle okolju omogoča izvedbo naslednjih funkcij (Oracle, 2008, str. 4/1–9/8):

- Regresija (angl. *regression*),

- Klasifikacija (angl. c*lassification*),
- Detekcija anomalij (angl. *anomaly detection*),
- Gručenje (angl. *clustering*),
- Asociacije (angl. *association*),
- Izbor in ekstrakcija lastnosti (angl. *feature selection and extraction*)

Izvedbo zgoraj navedenih funkcij podpirajo naslednji algoritmi (Oracle, 2008, str. 10/1–18/6):

- Apriori,
- Odločitvena drevesa (angl. *Decision trees – DT*),
- Posplošeni linearni model (angl. *Generalized Linear Models – GLM*),
- Razvrščanje z voditelji (angl. *k-means*),
- Minimalna opisna razdalja (angl. *Minimum Description Length – MDL*),
- Naivni Bayesov algoritem, (angl. *Näive Bayes – NB*),
- Nenegativna matrična faktorizacija (angl. *Non-Negative Matrix Factorization – NNMF*),
- O-Cluster,
- Metoda podpornih vektorjev (angl. *Support Vector Machines – SVM*).

## 3.2 Posebnosti priprave podatkov v Oracle okolju

Do sedaj smo že ugotovili, da je priprava podatkov eden od ključnih korakov v procesu podatkovnega rudarjenja. Kot je bilo že opisano v predhodnih poglavjih, je potrebno podatke dobro pregledati, jih ustrezno prečistiti in transformirati ter jih pripraviti glede na predvideno uporabo specifičnega algoritma. Pri tem se analitik podatkovnega rudarjenja sreča z nekaterimi zahtevami in posebnostmi priprave podatkov v Oracle okolju.

### 3.2.1 Osnovne zahteve

#### 3.2.1.1 Tabela primerov

Osnovna zahteva pri pripravi podatkov za podatkovno rudarjenje v Oracle okolju je, da se podatke organizira v eni sami tabeli (Oracle, 2008, str. 19/2), ki se nahaja v Oracle bazi podatkov, kjer se tudi sicer izvaja celoten proces podatkovnega rudarjenja.

Vsak zapis v tabeli imenujemo primer (angl. *case*), ki ga enolično označuje unikatni identifikator.

#### 3.2.1.2 Tipi podatkov

Oracle Data Mining podpira naslednje podatkovne tipe (Oracle, 2010 , str. 3/2):

- VARCHAR2,
- CHAR,
- NUMBER,
- FLOAT,
- DM_NESTED_CATEGORICALS,
- DM_NESTED_NUMERICALS.

Vse druge morebitne podatkovne tipe, ki jih potrebujemo, je potrebno prevesti v enega od zgoraj navedenih tipov.

Najbolj pogost podatkovni tip, ki ni podprt, je datumski (DATE, TIMESTAMP). Če želimo tak tip podatka uporabiti, ga bomo v večini primerov preslikali v numerični tip podatka (NUMBER), včasih pa ga je smiselno, v odvisnosti od zahtev podatkovnega rudarjenja, preslikati v nominalni (kategoričen) tip podatka (VARCHAR2) (Oracle, 2008, str. 19/2 – 19/3).

Drug primer, ko bomo uporabili preslikavo v drug podatkovni tip, je primer poštne številke. Če je poštna številka v izvornih podatkih zapisana kot numeričen podatek, je smiselno preslikati to numerično vrednost v nominalno. Vsebina podatka namreč ne predpostavlja medsebojne urejenosti, zato sklepamo, da gre v tem primeru za kategoričen atribut (Oracle, 2008, str. 19/2).

#### 3.2.1.3 Potrebne zbirke podatkov

Za izdelavo klasifikacijskega ali regresijskega modela podatkovnega rudarjenja potrebujemo dve tabeli primerov, eno za kreiranje modela, drugo za njegovo testiranje. Običajno obe tabeli primerov dobimo iz osnovne zbirke, ki jo razdelimo s pomočjo vzorčenja. Na primer 60 % primerov se uporabi za izgradnjo modela, 40 % pa za njegovo testiranje. Modeli, ki implementirajo druge funkcije podatkovnega rudarjenja, ne potrebujejo posebnih testnih podatkov. Ne glede na to je potrebna še ena zbirka podatkov, na kateri apliciramo model podatkovnega rudarjenja. Pri tem je pomembno, da imajo vse potrebne zbirke podatkov iste atribute in so pripravljene na isti način (Oracle, 2010, str. 3/2).

### 3.2.2 Gnezdeni podatki

Algoritmi pričakujejo, da so podatki organizirani v eni sami tabeli primerov, kjer vsak zapis predstavlja samostojen primer. V primeru, ko imamo med izvornimi podatki na voljo tudi transakcijske podatke, kot so na primer naročila z vsemi naročenimi izdelki nekega kupca, imamo med dvema entitetama (kupec in naročilo) opraviti z relacijo ena-več (angl. *one-to-many*).

Ko uporabljamo Oracle Data Mining, imamo možnost, da za te transakcije ne uporabimo operacije pivotiranja tabel, s katero bi obstoječe podatke iz tabele transakcij agregrirali, jih nato transformirali ter jih prikazali kot dodatne stolpce v zapisu posameznega primera, temveč lahko uporabimo možnost gnezdenja (angl. *Nested data*).

Z gnezdenjem lahko v dodaten stolpec tabele primerov zapišemo gnezdeno tabelo kot tabelo parov atribut – vrednost. Oracle Data Mining interpretira vsako vrstico gnezdene tabele kot ločen atribut osnovne tabele primerov.

Gnezdene podatke lahko uporabimo v primeru naslednjih algoritmov (glej Tabelo 2):

*Tabela 2: Gnezdeni podatki in algoritmi podatkovnega rudarjenja*

| **Algoritem** | **Funkcija** |
|---|---|
| Apriori | Asociativna pravila |
| Posplošeni linearni model | Klasifikacija, regresija |
| Razvrščanje z voditelji | Gručenje |
| Minimalna opisna razdalja | Pomembnost atributov |
| Naivni Bayesov algoritem | Klasifikacija |
| Nenegativna matrična faktorizacija | Ekstrakcija lastnosti |
| Metoda podpornih vektorjev | Klasifikacija, regresija, detekcija anomalij |

*Vir: Oracle, Oracle Data Mining Application Developer's Guide, 11g Release 2 (11.2), 2010, str. 3/8.*

Gnezdeni podatki so lahko dveh podatkovnih tipov, od katerih je en namenjen numeričnim, drugi pa kategoričnim atributom:

- DM_NESTED_NUMERICALS
- DM_NESTED_CATEGORICALS

#### 3.2.2.1 Primer transformacij v primeru gnezdene tabele

Gnezdenje si najlažje predstavljamo na primeru tabele transakcij (glej Sliko 27), ki jo je potrebno pripraviti za podatkovno rudarjenje. Tabela transakcij ima naslednjo vsebino:

*Slika 27: Primer tabele transakcij*

```
KUPEC ID     IZDELEK      VREDNOST NAKUPA
-----------------------------------------------
1001         Izdelek A    100
1002         Izdelek B    200
1001         Izdelek B    200
1003         Izdelek A    100
1003         Izdelek C    300
```

KUPEC ID nastopa kot identifikator tabele primerov. Med entiteto kupec in entiteto nakup obstaja relacija »ena – več«, kar pomeni, da so podatki o nakupih kandidati za gnezdeno tabelo. Tabela primerov z gnezdenimi podatki bi bila videti, kot je prikazano na spodnji sliki (glej Sliko 28):

*Slika 28: Tabela primerov z gnezdenimi podatki*

```
KUPEC ID     NAKUPI
-----------------------------------------------
1001         (Izdelek A, 100)
             (Izdelek B, 200)
1002         (Izdelek B, 200)
1003         (Izdelek A, 100)
             (Izdelek C, 300)
```

Kljub temu, da bodo podatki pripravljeni zelo enostavno, bo Oracle Data Mining ob izvedbi algoritmov podatkovnega rudarjenja, dejansko izvedel transformacijo in celotno gnezdeno tabelo pivotiral na naslednji način (glej Sliko 29):

*Slika 29: Izvedena transformacija nad gnezdenem stolpcem*

```
KUPEC ID     NAKUPI.Izdelek A    NAKUPI.Izdelek B    Nakupi.Izdelek C
-------------------------------------------------------------------------
1001         100                 200                 0
1002         0                   200                 0
1003         100                 0                   300
```

Z drugimi besedami, operacijo, ki bi jo morali v vsakem primeru narediti ročno, Oracle Data Mining izvede avtomatično.

Edina izjema, pri kateri ne bomo uporabili gnezdenih stolpcev, je v asociativnem modelu, na primer pri analizi tržne košarice.

### 3.2.3 Tri ključne transformacije

Oracle Data Mining podpira tri ključne transformacije, ki se uporabljajo pri vseh algoritmih podatkovnega rudarjenja, in sicer:

- razvrščanje v grupe,

- normalizacijo podatkov in
- obravnavo osamelcev.

Razvijalec modela lahko seveda razvije svoje lastne programske funkcije (s pomočjo programskih jezikov PL/SQL ali Java), ki jih vključi v model, ki ga razvija.

#### 3.2.3.1 Razvrščanje v grupe

Pri razvrščanju v grupe se moramo zavedati, da manj grup praviloma vodi v bolj kompaktne modele podatkovnega rudarjenja, ki jih je mogoče hitreje zgraditi. Hkrati pa to pomeni možno izgubo informacije, kar smo že obravnavali v poglavju Transformacija in diskretizacija podatkov.

Prav tako vpliva na model pravilna izbira meja med posameznimi grupami. Oracle Data Mining ima implementirane naslednje štiri metode razvrščanja v grupe (Oracle, 2008, str. 19/9):

- Razvrščanje na osnovi N najpogostejših pojavitev se uporablja za razvrščanje nominalnih atributov. Pred razvrščanjem se odločimo za število grup, v katere bomo razvrščali podatke. V prvo grupo bomo nato razvrstili vrednost, ki se največkrat pojavi, v drugo tisto, ki ima drugo najvišje število pojavitev, in tako naprej do zadnje grupe, v katero razporedimo vse preostale vrednosti.
- Pri nadzorovanem razvrščanju v grupe se za določitev meja med posameznimi grupami uporablja določeno karakteristiko podatkov. Na osnovi enega prediktorja se zgradi odločitveno drevo, s pomočjo katerega se identificira meje med grupami. Ta metoda je uporabna za razvrščanje tako numeričnih, kot tudi nominalnih podatkov.
- Razvrščanje v grupe na osnovi enakomernega obsega smo že obravnavali v poglavju Razvrščanje v grupe, kjer smo obravnavali ključne naloge v procesu priprave podatkov. Na osnovi intervala vrednosti posameznega atributa, se vse vrednosti atributa razporedijo v intervale enakega razpona. Koliko bo grup, lahko opredelimo z vnosom parametra za število grup, ali pa se število grup izračuna avtomatično. To razvrščanje se uporablja le v primeru numeričnih podatkov in pri ugotavljanju osamelcev.
- Razvrščanje na osnovi kvantilov smo obravnavali v poglavju Razvrščanje v grupe,, ko smo opisali razvrščanje v grupe na osnovi enakomerne porazdelitve.

#### 3.2.3.2 Normalizacija podatkov

Oracle Data Mining omogoča normalizacijo podatkov s pomočjo treh metod normalizacije, in sicer (Oracle, 2008, str. 19/9):

- Min-Max normalizacija,
- normalizacija na osnovi absolutnih vrednosti (angl. *scale normalization*) in
- normalizacija na podlagi standardne oziroma z-vrednosti.

Min-Max normalizacijo in normalizacijo na podlagi standardne oziroma z-vrednosti smo obravnavali v poglavju Normalizacija vrednosti. Normalizacija na osnovi absolutnih vrednosti, prav tako kot Min-Max normalizacija, uporablja najmanjšo in najvišjo vrednost, vendar se vse vrednosti preslikajo v absolutne vrednosti. Njihov razpon opredeljuje vrednost $max\{abs(max), abs(min)\}$.

#### 3.2.3.3 Obravnava osamelcev

Osamelci so v nalogi že obravnavani v poglavju Izstopajoče vrednosti ali osamelci, ko smo govorili o čiščenju podatkov. Osamelci namreč, če jih ne obravnavamo pravilno, bistveno vplivajo na vzorce podatkov in na transformacije, ki jih izvajamo nad njimi, na primer pri Min-Max normalizaciji ali razvrščanju v grupe na osnovi enakomernega obsega.

Oracle Data Mining obravnava osamelce na dva načina (Oracle, 2008, str. 19/9–19/10), in sicer s preslikovanjem robnih vrednosti (angl. *winsorization*) in z obrezovanjem (angl. *trimming*).

Z metodo preslikovanja robnih vrednosti se vse robne vrednosti preslikajo v neko določeno vrednost. Tako se v primeru 90 % preslikave robnih vrednosti, najnižjih 5 % vrednosti preslika v najvišjo vrednost 5. percentila, in najvišjih 5 % v najnižjo vrednost 95. percentila (Oracle, 2008, str. 19/10). Pri obrezovanju se robne vrednosti preslikajo v vrednost NULL in se nato v algoritmih obravnavajo kot manjkajoče vrednosti (Oracle, 2008, str. 19/10).

### 3.2.4 Manjkajoči podatki

Oracle Data Mining razlikuje med »redkimi« podatki in podatki, ki vsebujejo »naključne manjkajoče vrednosti«. Osnovna razlika med tema dvema vrstama podatkov je, da podatki z naključnimi manjkajočimi vrednostmi dejansko niso znani, pri redkih podatkih pa gre za vrednosti, za katere predpostavimo, da so znane, vendar v podatkovni zbirki niso zajete.

Tipičen primer redkih podatkov je nakupna košarica, kjer je med stotinami izdelkov, le nekaj izdelkov vključenih v transakcijo, ki prikazuje posamezen primer nakupne košarice. Izdelkov, ki jih ni v konkretni nakupni košarici, se v okviru te transakcije ne zapisuje v podatkovno zbirko. Vemo pa, da je vrednost nakupa oziroma količina izdelkov, ki jih nismo dali v nakupno košarico enaka 0 (Oracle, 2010, str. 3/11).

Oracle Data Mining obravnava manjkajoče podatke (Oracle, 2010, str. 3/11) kot:

- manjkajoče (angl. *missing*), pri katerih gre za dejansko manjkajoče vrednosti v posameznih stolpcih enostavnih podatkovnih tipov (negnezdeni stolpci), ki se obravnavajo kot naključno manjkajoči, in
- redke (angl. *sparse*), pri katerih gre za vrednosti, ki manjkajo v gnezdenih stolpcih.

#### 3.2.4.1 Primer manjkajočih podatkov v tabelah kupcev in naročil

V primeru knjižnega kluba, ki je obravnavan podrobneje v nadaljevanju, sta uporabljeni dve tabeli, in sicer SRC_KUPCI, ki predstavlja tabelo kupcev in njihovih podatkov, in SRC_NAROCILA, ki hrani podatke o vseh transakcijah, ki so jih izvedli člani kluba.

Tabela SRC_KUPCI je osnovna tabela primera in bo, ko bo pripravljena za podatkovno rudarjenje, vključevala gnezden stolpec, ki bo vključeval podatke iz tabele SRC_NAROCILA.

Tabela SRC_KUPCI vsebuje naključno manjkajoče podatke, kar vidimo že iz enostavnega izpisa podatkov (glej Sliko 30).

*Slika 30: Delni izpis tabele SRC_CLANI*

```
> select * from SRC_CLANI;
CLAN   FIZOSEBA      STEVCLAN     NACPRI NACIZP LETOZAC      OBDZAC LETOZADZEL
       OBDZADZEL     POSTA  PTTNAZIV      LETOROJ      TOSTEL
6123293      104638 2      1      B      2001   3      2011   1      8000   NOVO
MESTO  973    41
6970057      104644 1      7      -      2006   4      null   null   2380   SLOVENJ
GR     974    2
7613763      104662 1      10     -      2011   4      null   null   3255   BUČE
       971    3
4194734      104674 1      6      P      1992   2      1997   4      1360   VRHNIKA
       967    -
3471497      104688 1      9      P      1990   1      2000   3      1410   ZAGORJE
OB     null   3
4797049      104689 1      5      P      1993   2      1998   4      1000   LJUBLJANA
       null   1
1010420      104704 1      5      -      1981   3      null   null   1351   BREZOVICA
       956    1
6105985      104735 1      7      S      2002   2      2002   2      2223   JUROVSKI
D      null   2
1833136      104760 1      5      P      1976   1      2000   4      6250   ILIRSKA
BI     944    -
6970743      104789 1      7      M      2006   4      2012   2      1000   LJUBLJANA
       null   41
6482046      104795 2      1      null   2005   1      null   null   8275   ŠKOCJAN
       964    31
```

Vrednosti NULL se pojavljajo naključno, v različnih stolpcih in med njihovim pojavljanjem ni nobene povezave.

V primeru tabele naročil si lahko ogledamo vsako posamezno transakcijo posebej. Na primer transakcija z oznako VRSTANAR = 'N31' in NAROCIL = '5094537' (glej Sliko 31) ima vključene tri izdelke s količinami, ki so različne od 0.

*Slika 31: Primer transakcije naročila*

```
> select * from SRC_NAROCILA where VRSTANAR = 'N31' and NAROCIL='5094537';
CLAN   VRSTANAR      NAROCIL       ARTIKEL       KOLICINA      DATUMNAK      NACINPRID
       LETONAK       OBDNAK STORFAK       GRUPAART      GRUPANAK      VRSTAKL1
       VRSTAKL2      VRSTAKL3      PCENAKK
5963236       N31     5094537       97896101175751      05.07.12      30      2012   3
       -      512     24      1      3      4      18,35
5963236       N31     5094537       97896101165781      05.07.12      30      2012   3
       -      266     25      1      4      5      23,95
5963236       N31     5094537       97886111539401      05.07.12      30      2012   3
       -      233     23      1      1      3      15,95
```

Število vseh artiklov, ki se pojavljajo v tabeli SRC_NAROCILA, je 10.581 (glej Slikoa 32).

*Slika 32: Število različnih naročenih izdelkov*

```
> select count(distinct ARTIKEL) from SRC_NAROCILA;
10581
```

Slednje pomeni, da bi morali transakcijo, da bi bila pravilno zapisana (za potrebe podatkovnega rudarjenja), zapisati s 10.581 vrsticami, v katerih bi za vsak izdelek, razen za zgornje tri, zapisali v polje KOLICINA vrednost 0. To je seveda nesmotrno in vseh vrednosti ne pišemo. Imamo pa opraviti seveda z redkimi manjkajočimi vrednostmi.

#### 3.2.4.2 Oracle Data Mining in obravnava manjkajočih vrednosti

Kako Oracle Data Mining obravnava manjkajoče vrednosti, je odvisno od uporabljenega algoritma in od tega, ali imamo opraviti z numeričnimi ali nominalnimi podatki oziroma ali gre za redke ali naključno manjkajoče podatke (Oracle, 2010, str. 40/3-12).

Avtomatična obravnava manjkajočih vrednosti je prikazana v Tabeli 3.

V primeru, ko avtomatične transformacije ne ustrezajo potrebam modela, jih je potrebno ustrezno nadomestiti z drugimi transformacijami. Tako lahko, kadar želimo obravnavati naključno manjkajoče podatke kot redke, uporabimo na primer SQL funkcijo NVL, ki vrednosti NULL preslika v vrednost 'NA'. Ker je atribut s tem prevzel neko določeno vrednost, podatek ne bo obravnavan kot naključno manjkajoč (Oracle, 2010, str. 3/12).

V nasprotnem primeru, ko želimo obravnavati redke manjkajoče vrednosti kot naključno manjkajoče, je potrebno transformirati gnezdene podatke v ločene stolpce, v katere se nato manjkajoče vrednosti zapišejo kot NULL (Oracle, 2010, str. 3/12).

*Tabela 3: Obravnava manjkajočih vrednosti v Oracle Data Miningu*

| Manjkajoči podatki | SVM, NMF, k-Means, GLM | NB, MDL, DT, OC | Apriori |
|---|---|---|---|
| **Numerični, naključno manjkajoči** | Manjkajoče vrednosti se nadomesti s povprečno vrednostjo. | Algoritem sam obravnava manjkajoče vrednosti kot naključno manjkajoče. Posebna predpriprava za to ni potrebna. | Algoritem obravnava naključno manjkajoče vrednosti kot redke. |
| **Nominalni, naključno manjkajoči** | Manjkajoče vrednosti se nadomesti z modusom. | Algoritem sam obravnava manjkajoče vrednosti kot naključno manjkajoče. Posebna predpriprava za to ni potrebna. | Algoritem obravnava naključno manjkajoče vrednosti kot redke. |
| **Numerični, redki** | Manjkajoče vrednosti se nadomesti z vrednostjo 0. | DT in OC ne podpirata gnezdenih podatkov, zato tudi ne obravnavata redkih podatkov, NB in MDL nadomestita manjkajoče podatke z 0. | Algoritem sam obravnava manjkajoče podatke kot redke. Posebna predpriprava za to ni potrebna. |
| **Nominalni, redki** | Manjkajoče vrednosti se nadomestijo z ničelnim vektorjem. | DT in OC ne podpirata gnezdenih podatkov, zato tudi ne obravnavata redkih podatkov, NB in MDL nadomestita manjkajoče kategorične podatke s posebno vrednostjo DM$SPARSE. | Algoritem sam obravnava manjkajoče podatke kot redke. Posebna predpriprava za to ni potrebna. |

*Vir: Oracle, Oracle Data Mining Application Developer's Guide, 11g Release 2 (11.2), 2010, str. 3/13.*

### 3.2.5 Avtomatična in vgrajena priprava podatkov

Pri pripravi podatkov se v odvisnosti od uporabljene funkcije podatkovnega rudarjenja lahko pripravlja različne množice podatkov (učni vzorec, testni vzorec, dejanski vzorec podatkov, nad katerim izvedemo podatkovno rudarjenje) in na vseh je potrebna izvedba istih transformacij. Različna orodja podatkovnega rudarjenja (ne samo Oracle Data Miner, ki ga za potrebe naloge uporabljamo za izvedbo praktičnega primera) omogočajo uporabo

funkcionalnosti, ki omogočajo poenostavitev izvedbe posameznih korakov procesa podatkovnega rudarjenja, predvsem v fazi priprave podatkov. Uporabniki te funkcionalnosti seveda z veseljem uporabimo, vendar je ključno, da se zmožnosti in omejitve teh orodij zavedamo in jih razumemo.

Oracle Data Miner omogoča razvijalcu aplikacije podatkovnega rudarjenja, da pri pripravi podatkov uporablja naslednje funkcionalnosti (Oracle, 2008, str. 19/1–19/2):

- Vgrajena priprava podatkov (angl. *Embedded data preparation*) omogoča, da vse transformacije, ki so bile razvite v okviru vzpostavljanja modela, vključimo v samo izvedbo postopkov procesa. Na ta način se opredeli pravila preslikav na enem mestu.
- Avtomatična priprava podatkov (angl. *Automatic data preparation*). Če je vklopljen način priprave podatkov z avtomatično pripravo podatkov, Oracle Data Miner avtomatično izvede pripravo podatkov, ki jih zahteva algoritem. Procedure za avtomatično pripravo se vključi v proces priprave podatkov skupaj z drugimi procedurami, ki smo jih sicer drugače pripravili.
- Orodja za razvoj lastnih procedur za transformacijo. S pomočjo teh orodij lahko razvijalec razvije svoje lastne procedure za transformacijo podatkov.
- Avtomatična obravnava manjkajočih in neobstoječih podatkov omogoča konsistentno obravnavo teh vrednosti v odvisnosti od uporabljenega algoritma.
- Transparentnost je pomembna lastnost orodja Oracle Data Miner. Ko gradimo model podatkovnega rudarjenja, prihaja do preslikav in transformacij podatkov. Zaradi specifičnosti posameznih algoritmov ti podatki niso več identični originalnim podatkom. Prav transparentnost pa omogoča, da tudi podrobnosti podatkovnega modela dejansko vidimo skozi originalne vrednosti atributov.

## 3.3 Priprava podatkov za podatkovno rudarjenje na primeru knjižnega kluba

Teoretične predpostavke, opisane v predhodnih poglavjih, bomo v nadaljevanju preizkusili na praktičnem primeru knjižnega kluba, ki trži in prodaja knjige preko svoje mreže prodajaln – klubskih centrov, klicnega centra, mreže prodajnih zastopnikov in spleta.

Priprava podatkov bo izvedena na zbirki podatkov, ki vsebuje podatke o članstvu in prodaji od nastanka knjižnega kluba. S pripravljenimi podatki je mogoče pripraviti modele za napovedovanje prekinitve članstva, razvrščanje članov kluba v gruče, analizirati nakupno košarico. Z dodatnimi podatki, kot je na primer odzivanje na trženjske akcije, pa bi lahko napovedni model izboljšali tako, da bi predvideli, kateri člani se bodo na določeno akcijo bolje odzivali od drugih.

Knjižni klub posluje v skoraj nespremenjeni obliki od začetka 70-ih let prejšnjega stoletja. Deluje po principu članstva, ki zahteva od članov, da vsako četrtletje kupijo vsaj eno knjigo iz kataloga. Poleg kataloga, v katerem so vsako četrtletje predstavljene knjige in drugi izdelki, v podjetju pripravljajo posebne akcije (na primer rojstnodnevne ugodnosti), dodatne ugodnosti za nakup dodatnih knjig v četrtletju in akcije pridobivanja novih članov, ki stalno potekajo preko različnih prodajnih kanalov, to je zastopnikov, klubskih centrov, klicnega centra, spletne trgovine. V zadnjih nekaj letih se v podjetju soočajo z upadom članstva in posledično z upadom prodaje. Podatkovno rudarjenje in izvedba trženjskih akcij z njegovo pomočjo je ena od možnih izboljšav v poslovanju.

Pri izvedbi trženjskih akcij je v pomoč izbiri naslovnikov, ki jim bodo poslali neko ponudbo, na voljo precej standardizirana aplikacija za pripravo akcij, ki je del zalednega informacijskega sistema. Dejansko gre za aplikacijo, v katero se na nekaj ekranskih maskah vnese kriterij izbora. Kriterij izbora se shrani v obliki zahtevka za izvedbo paketne obdelave, ki se izvede ponoči. V kolikor v podjetju presodijo, da so dobili slab izbor, ves postopek ponovijo naslednji dan.

Cilj uvedbe podatkovnega rudarjenja je tako med drugim odkrivati nove kandidate za trženjske akcije, saj je ciljna skupina pri uporabi zaledne aplikacije velikokrat ista ali vsaj zelo podobna. Prav tako nas na primer zanima, kdo od obstoječih članov kluba bo glede na njegove osebne podatke in nakupno zgodovino, njegovo in njemu podobnih članov, v prihodnjem obdobju najverjetneje izstopil iz kluba. Prav tako lahko poiščemo izdelke, ki jih člani pogosto kupujejo v spletni trgovini v okviru iste transakcije, in na tej osnovi oblikujemo spletno ponudbo tako, da bodo ti izdelki prikazani skupaj. Zanima nas tudi, katere knjige bodo posamezni člani najverjetneje kupili oziroma kako prilagoditi ponudbo vsakemu članu knjižnega kluba posebej. Mogoče bodo to slednje želeli uporabiti že v času klica člana v klubski klicni center.

Pri tem so na voljo vsi podatki, ki jih o kupcih imajo v podjetju in so shranjeni v enotnem zalednem sistemu. Četudi so v uporabi različni prodajni kanali, so v zalednem sistemu že zbrani vsi podatki o prodaji članom knjižnega kluba. Na ta način odpade potreba po iskanju in združevanju podatkov iz posameznih poslovnih aplikacij.

### 3.3.1 Analiza podatkov

#### 3.3.1.1 Pridobivanje podatkov

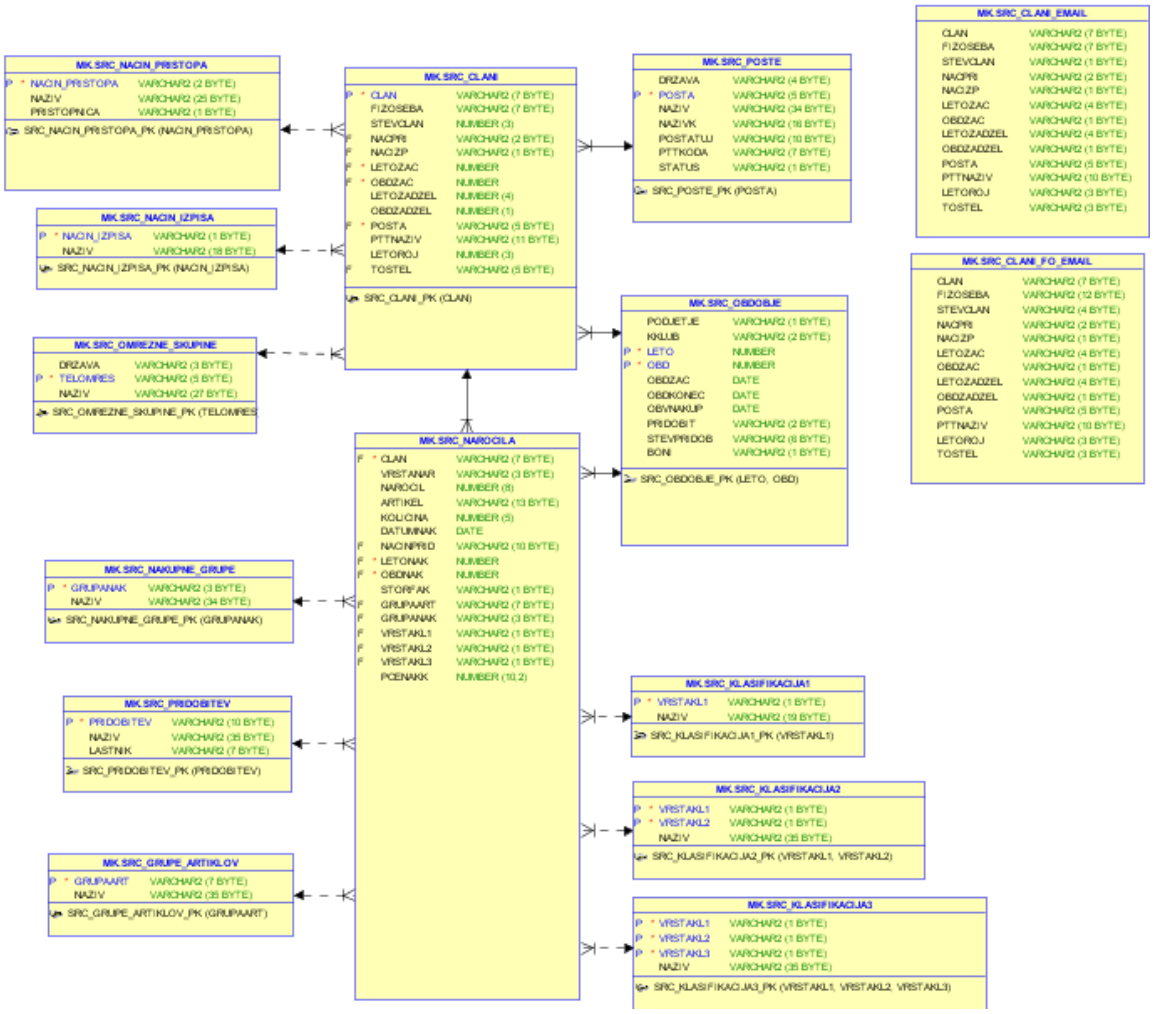
V primeru knjižnega kluba se izvorni podatki nahajajo v transakcijskem sistemu prodaje v tabelah, ki vsebujejo podatke o kupcih, prodajnih naročilih in izdelkih. V transakcijskem modelu so podatki o prodaji že združeni in na voljo v eni sami tabeli, čeprav se za različne

prodajne poti uporablja različne poslovne aplikacije (na primer blagajniški program, spletna trgovina, klicni center).

Tabele so iz izvornega sistema prenesene v podatkovno bazo, kjer bomo izvajali podatkovno rudarjenje, z ETL orodjem Oracle Warehouse Builder v izvorni obliki, brez transformacij in integracij, ki so sicer v takšnih projektih potrebne.

Podatkovni model, ki ponazarja tabele (seznam vseh tabel se nahaja v Prilogi 3) in relacije med njimi po prenosu v shemo, ki jo bomo uporabili za podatkovno rudarjenje, lahko prikažemo z naslednjim entitetno-relacijskim (E-R) diagramom (glej Sliko 33):

*Slika 33: E–R diagram podatkovnega modela izvornih tabel*




### 3.3.1.2 Raziskovanje podatkov

Ključni tabeli za raziskovanje podatkov v primeru knjižnega kluba sta SRC_CLANI in SRC_NAROCILA. Druge tabele so predvsem šifranti in jih potrebujemo predvsem za

preslikavo šifer v opise. Podatki v šifrantih so praviloma enoznačni, zato sta predmet raziskovanja podatkov omenjeni dve tabeli.

#### 3.3.1.2.1 Podatki o članih knjižnega kluba – analiza tabele SRC_CLANI

Za začetek analize izdelamo nekaj osnovnih statistik in pregledamo porazdelitve vrednosti posameznih atributov. Za to je mogoče uporabiti funkcijo Explore Data, ki je na voljo v orodju Oracle Data Miner. S tem orodjem izdelamo enostaven model, kjer nad izvorno tabelo podatkov izvedemo funkcijo Explore Data (glej Sliko 34).

*Slika 34: Data mining model – Explore SRC_CLANI*


Explore Data prikaže osnovne opisne statistike podatkov v izvorni tabeli (glej Sliko 35):

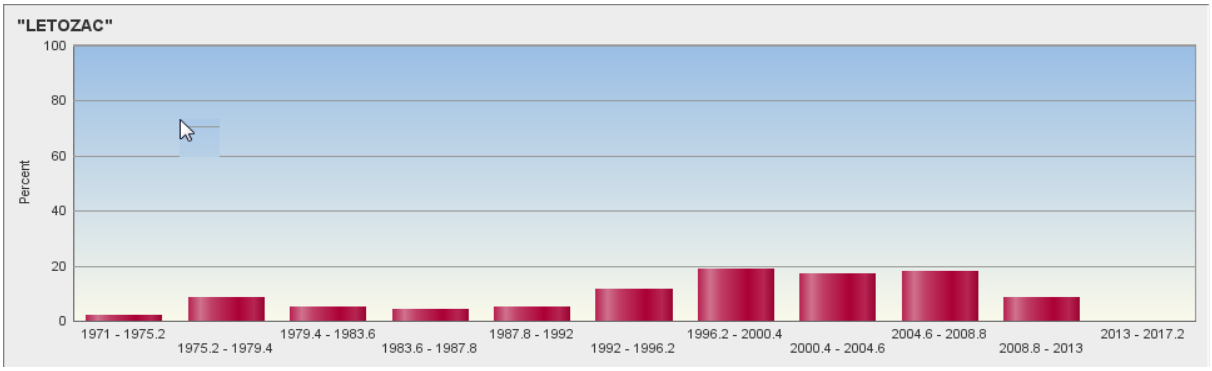
- tip podatka,
- odstotek NULL vrednosti,
- število in odstotek različnih vrednosti,
- modus, srednja vrednost, mediana,
- najmanjša in največja vrednost,
- standardni odklon in varianca,
- asimetričnost in sploščenost.

*Slika 35: Rezultat funkcije Explore Data za tabelo SRC_CLANI*

| Name | Histogram | Data Type | Percent NULLs | Distinct Values | Distinct Percent | Mode | Average | Median | Min Value | Max Value | Standard Deviation | Variance | Skewness | Kurtosis |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLAN | | VARCHAR2 | 0 | 328.319 | 96,0739 | 4.788.329 | | | | | | | | |
| FIZOSEBA | | VARCHAR2 | 0 | 318.877 | 93,3109 | 2.708.730 | | | | | | | | |
| LETOROJ | | NUMBER | 11,1958 | 111 | 0,0366 | | 943,9295 | 970 | 0 | 999 | 151,2245 | 22.868,8433 | -5,9933 | 34,303 |
| LETOZAC | | NUMBER | 0 | 43 | 0,0126 | | 1.996,8954 | 1.999 | 1.971 | 2.013 | 10,2365 | 104,7865 | -0,7277 | -0,5112 |
| LETOZADZEL | | NUMBER | 15,253 | 35 | 0,0121 | | 2.003,4294 | 2.004 | 1.979 | 2.022 | 5,4667 | 29,8844 | -0,504 | -0,0554 |
| NACIZP | | VARCHAR2 | 3,4784 | 15 | 0,0045 | P | | | | | | | | |
| NACPRI | | VARCHAR2 | 0 | 14 | 0,0041 | 7 | | | | | | | | |
| OBDZAC | | NUMBER | 0 | 4 | 0,0012 | | 2,2127 | 2 | 1 | 4 | 1,1268 | 1,2696 | 0,3416 | -1,2978 |
| OBDZADZEL | | NUMBER | 15,253 | 4 | 0,0014 | | 2,6044 | 3 | 1 | 4 | 1,1802 | 1,3928 | -0,103 | -1,4915 |
| POSTA | | VARCHAR2 | 0 | 632 | 0,1849 | 1.000 | | | | | | | | |
| PTTNAZIV | | VARCHAR2 | 0 | 598 | 0,175 | LJUBLJANA | | | | | | | | |

Za vsak atribut lahko prikažemo histogram s porazdelitvijo (glej Sliko 36), pri čemer so posamezne vrednosti razvrščene v grupe na osnovi enakega intervala:

*Slika 36: Histogram porazdelitve za atribut LETOZAC*




Dodatno se z uporabo *Group by* funkcije generira multivariantno analizo v obliki histograma. Prikazane statistike je mogoče prikazati za vse podatke ali za vzorec (na primer 2000 zapisov) podatkov.

Podrobnejšo analizo podatkov izvedemo z uporabo statističnega orodja R, ki ga je mogoče z dodatnim paketom Oracle R Enterprise (ORE) uporabiti neposredno nad podatki v bazi podatkov.

Tabela Analiza tabele SRC_CLANI, v kateri so navedene osnovne statistike atributov tabele SRC_CLANI, se nahaja v Prilogi 4.

#### 3.3.1.2.1.1 Atribut CLAN

Atribut CLAN vsebuje podatek o članski številki člana knjižnega kluba.

S funkcijo summary() izpišemo frekvenčno porazdelitev vrednosti za atribut CLAN (glej Sliko 37):

*Slika 37: Osnovni podatki porazdelitve atributa CLAN*

```
> clani <- SRC_CLANI
> summary(clani$CLAN,20)
4788329 5147608 1550466 2477073 3079845 3818333 4719993 4954426 4959029 5044474
      8       8       7       7       7       7       7       7       7       7
5214515 5326954  544593 5562871 5652730 5677976 5871595 1051630 1070481 (Other)
      7       7       7       7       7       7       7       6       6  341603
```

Iz podatkov je razvidno, da se posamezne članske številke večkrat ponovijo. Razlog je v tem, da se nekateri člani izpišejo in se čez čas ponovno včlanijo. Podatek o tem, kolikokrat se je nekdo ponovno včlanil v klub, je sicer vsebovan v atributu STEVCLAN.

V algoritmih podatkovnega rudarjenja bo ta atribut uporabljen kot identifikacijski atribut za razvrščanje v grupe.

#### 3.3.1.2.1.2 Atribut FIZOSEBA

Atribut FIZOSEBA vsebuje podatek o identifikacijski številki kupca podjetja. Gre namreč za to, da v informacijskem sistemu podjetja obstajajo tudi kupci, ki niso člani knjižnega kluba. V primeru, da bi imeli na voljo podatke tudi o drugih naročilih (ne samo o naročilih knjižnega kluba), bi z uporabo tega podatka lahko povezali tudi ostala naročila člana knjižnega kluba, ki so bila sicer izvedena izven knjižnega kluba. Ti podatki za potrebe te naloge žal niso na voljo.

*Slika 38: Osnovni podatki porazdelitve atributa FIZOSEBA*

```
> clani <- SRC_CLANI
> summary(clani$FIZOSEBA,20)
2708730 3089283 3543891 4051776 1260033 1534125 1701231 1793468 2381335 2652207
      8       8       8       8       7       7       7       7       7       7
3148055 3443080 3533335 3709974 3841092  405473 4207341 5084587 5313861 (Other)
      7       7       7       7       7       7       7       7       7  341599
```

Tudi FIZOSEBA se pojavlja več kot enkrat (glej Sliko 38). Vsak član, ki ima identifikacijsko številko, lahko postane član več kot enkrat tako, da obnovi svoje članstvo z uporabo že obstoječe članske številke, včasih pa se mu dodeli novo člansko številko.

Za trenutne potrebe podatek FIZOSEBA ni potreben in se ga lahko izpusti.

#### 3.3.1.2.1.3 Atribut STEVCLAN

STEVCLAN je numerični atribut, ki hrani informacijo o tem, kolikokrat je posamezen član knjižnega kluba obnovil svoje članstvo.

*Slika 39: Osnovni podatki porazdelitve atributa STEVCLAN*

```
> clani <- SRC_CLANI
> summary(clani$STEVCLAN,20)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.049   1.000   8.000
```

Iz zgornjih podatkov (glej Sliko 39) na primer izvemo, da obstajajo člani, ki so dosedaj že sedemkrat izstopili iz kluba, in so že osmič obnovili svoje članstvo, kar smo sicer že ugotovili v analizi atributov CLAN in FIZOSEBA.

#### 3.3.1.2.1.4 Atribut NACPRI

Atribut NACPRI podaja informacijo o načinu pristopa člana v knjižni klub. Pomen posameznih kod je pojasnjen v tabeli SRC_NACIN_PRISTOPA, katere opis se nahaja v Prilogi 5.

S funkcijo summary() lahko ugotovimo frekvenčno porazdelitev atributa NACPRI (glej Sliko 40).

*Slika 40: Osnovni podatki porazdelitve atributa NACPRI*

```
> clani <- SRC_CLANI
> summary(clani$NACPRI)
     7      5      9      4      6      1      2     10     12      3     11
183252  68045  41306  17288  14614   7791   5628   1250   1036    728    671
    13      8      0
   115      9      3
```

Zanimiva informacija, ki jo razberemo iz zgornjih podatkov je, da so kar 53,62 % vseh članov knjižnega kluba pridobili zastopniki, da jih malo več kot 19 % v klubu pridobijo z akcijami ter da jih malo več kot 12 % pridobijo v klubskih centrih. Drugi kanali, vključno s telefonskim pridobivanjem, so očitno manj uspešni in predstavljajo le 5 % ali manj.

3.3.1.2.1.5 Atribut NACIZP

Atribut NACIZP podaja informacijo o načinu izpisa člana iz knjižnega kluba. Pomen posameznih kod je pojasnjen v tabeli SRC_NACIN_IZSTOPA, katere opis se nahaja v Prilogi 6.

S funkcijo summary() lahko ugotovimo frekvenčno porazdelitev atributa NACIZP (glej Sliko 41).

*Slika 41: Osnovni podatki porazdelitve atributa NACIZP*

```
> clani <- SRC_CLANI
> summary(clani$NACIZP)
     P      -      S      E      X      T      M      K      V      B      I
195162  40259  27313  23041  14967  13579   7642   3534   2121   1126    505
     A      O      p     ?P   NA's
   314    282      3      1  11887
```

Iz podatkov atributa NACIZP razberemo tudi, ali je nek član društva še aktiven ter njihovo skupno število. Aktivni člani imajo vrednost atributa NACPRI prazno (blank) ali NULL.

Iz podatkov, ki jih vrne funkcija summary(), razberemo, da je trenutno aktivnih 52.146 članov, kar predstavlja 15,26 % vseh vpisanih članov knjižnega kluba od leta 1971 (vseh članov, vključno z obnovitvami, je bilo dosedaj 341.736).

Za analizo je zanimiv še podatek, ki ga vrne funkcija summary() za člane, ki so se v klub včlanili po letu 2006 (glej Sliko 42).

*Slika 42: Osnovni statistike atributa CLANI ob pogoju LETOZAC >= 2006*

```
> clani2006 <- clani[clani$LETOZAC >= 2006]
> summary(clani2006$CLAN,1)
(Other)
  75733
> summary(clani2006$NACIZP)
    P     -     E     X     T     S     M     V     B     K     A     O     I
24993 17698 11947  7080  3290  2988  2497   856   686   621   163   107     6
 NA's
 2801
```

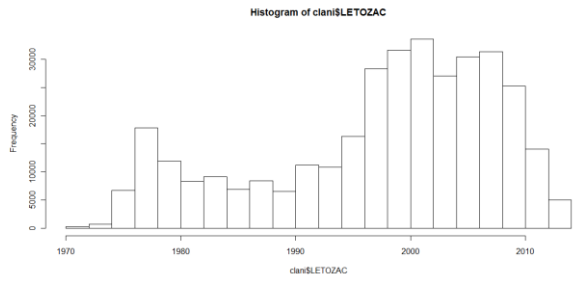
Od vseh 75.733 članov, ki so se včlanili v knjižni klub po letu 2006, jih je aktivnih še 20.499 (vrednosti »–« in »NA«) oziroma 27,07 %. Verjetno kar precejšen razlog za zaskrbljenost.

### 3.3.1.2.1.6 Atributa LETOZAC in OBDZAC

Atributa LETOZAC in OBDZAC vsebujeta podatek o letu in kvartalu začetka članstva v knjižnem klubu.

*Slika 43: Statistike in histogram vrednosti atributa LETOZAC*

```
> summary(clani$LETOZAC)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1971    1991    1999    1997    2005    2013

> hist(clani$LETOZAC)
```

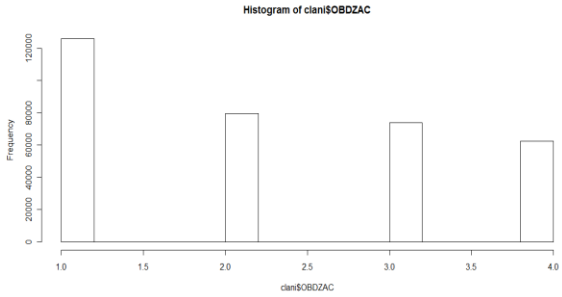



Iz histograma (glej Sliko 43) je razvidno, da je klub zelo uspešno pridobival nove člane v drugi polovici 70-ih, potem pa se je število novih članov ustalilo pri 10.000 letno. Tako je ostalo do leta 1994, ko se je letno število novih članov pričelo znatno povečevati. Število novih članov po letu 2010 je padlo na raven izpred leta 1994. Podatki, ki jih imamo na voljo, so iz sredine leta 2012, zato podatek za to leto ni popoln, a vseeno je videti, da je bilo to leto še slabše, kot prejšnje.

Analiza vrednosti podatkov atributa OBDZAC (glej Sliko 44) kaže, da je za pridobivanje novih članov najpomembnejši prvi kvartal (kar še potrjuje slabo napoved za leto 2012):

*Slika 44: Statistike in histogram vrednosti atributa OBDZAC*

```
> summary(clani$OBDZAC)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   2.213   3.000   4.000
> hist(clani$OBDZAC)
```




### 3.3.1.2.1.7 Atributa LETOZADZEL in OBDZADZEL

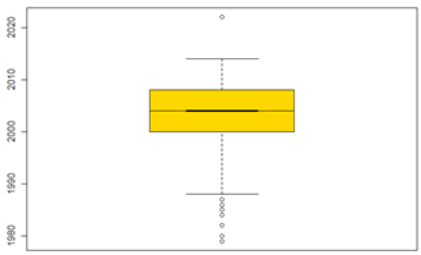
Atributa LETOZADZEL in OBDZADZEL določata, kdaj je članstvo v knjižnem klubu prenehalo oziroma kdaj se glede na pogodbo o članstvu izteče. Če je član še aktiven, potem je v zaledni aplikaciji v veljavi poslovno pravilo, da ima v tem primeru atribut LETOZADZEL vrednost NULL, prav tako tudi OBDZADZEL (glej Sliko 45).

*Slika 45: Osnovne statistike atributov LETOZADZEL in OBDZADZEL*

```
> summary(clani$LETOZADZEL)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   1979    2000    2004    2003    2008    2022   52120
> summary(clani$OBDZADZEL)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
    1.000     2.000     3.000     2.604     4.000     4.000 52120.000
```

Atribut LETOZADZEL ima najvišjo vrednost 2022 (glej Sliki 45 in 46). To je zaradi tega, ker člani ob vpisu v klub potrdijo članstvo v klubu za določeno obdobje. Tako vrednosti v prihodnosti niso vedno nepravilne, čeprav gre v primeru vrednosti 2022 najbrž za napako pri vnosu podatka. Lahko pa gre tudi za osamelce, kar lahko hitro preverimo:

*Slika 46: Boxplot diagram atributa LETOZADZEL*



Iz diagrama (glej Sliko 46) je razvidno, da je vrednost, ki leži nad vrednostjo 2014, samo ena. Iz tega sledi, da je pri vrednosti 2022 skoraj zagotovo prišlo do tipkarske napake.

#### 3.3.1.2.1.8 Atributa POSTA in PTTNAZIV

S summary() (glej Sliko 47) lahko hitro pridobimo osnovno informacijo o porazdeliti obeh atributov:

*Slika 47: Osnovni podatki porazdelitve atributov POSTA in PTTNAZIV*

```
> summary(clani$POSTA, 20)
   1000    2000    3000    4000    8000    3320    6000    2250    1420    1230
  38188   20121    9427    8288    6424    6422    5161    4603    3946    3854
   9000    1241    4220    5000    4270    2310    2380    1330    4290 (Other)
   3631    3573    3511    3104    2995    2493    2418    2385    2268  208924

> summary(clani$PTTNAZIV, 20)
 LJUBLJANA    MARIBOR      CELJE      KRANJ LJUBLJANA- NOVO MESTO    VELENJE
     38197      20121       9427       8288       6878       6424       6422
KOPER-CAPO       PTUJ   TRBOVLJE    DOMZALE MURSKA SOB     KAMNIK SKOFJA LOK
      5161       4603       3946       3853       3630       3573       3511
NOVA GORIC   JESENICE  SLOVENSKA SLOVENJ GR    KOCEVJE    (Other)
      3104       2995       2493       2418       2385     204307
```

Atributa POSTA in PTTNAZIV sta nominalna in po vsej verjetnosti soodvisna, kar preverimo s pomočjo testa $\chi^2$ (glej Sliko 48). Visoka soodvisnost je hkrati tudi kriterij za ugotavljanje redundantnosti.

*Slika 48: Preverba soodvisnosti med atributoma POSTA in PTTNAZIV*

```
> tabela <- table(clani$POSTA, clani$PTTNAZIV)
> chisq.test(tabela)

        Pearson's Chi-squared test

data:  tabela
X-squared = 200501263, df = 376076, p-value < 2.2e-16
```

Vrednost X-squared je zelo visoka, število prostorskih stopenj prav tako, p-vrednost pa je skoraj nič, kar pomeni visoko stopnjo soodvisnosti med atributoma. Zato se v nadaljnjih korakih priprave podatkov, enega od obeh podatkov lahko izpusti (na primer PTTNAZIV).

#### 3.3.1.2.1.9 Atribut LETOROJ

LETOROJ je atribut, ki vsebuje podatke o letu rojstva člana knjižnega kluba.

Funkcija summary() (glej Sliko 47) pokaže, da so podatki atributa LETOROJ vsaj malo nenavadni. Najnižja vrednost je 0, najvišja 999, povprečje pa je pri 943,9.

*Slika 49: Osnovni podatki porazdelitve atributov LETOROJ*

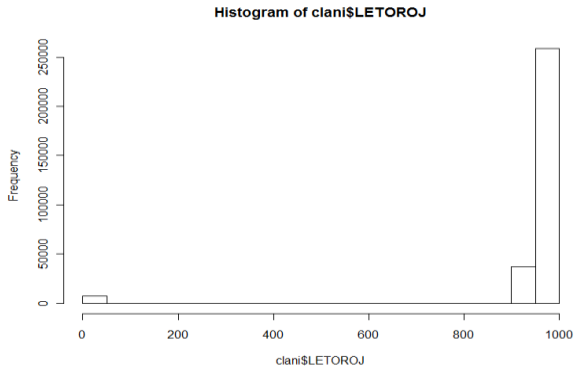
```
> summary(clani$LETOROJ)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
    0.0   957.0   970.0   943.9   979.0   999.0 38260.0
```

Trimestna vrednost letnice se pojavlja zato, ker v izvornem sistemu v podatku LETOROJ ni predvidena tisočica, tudi tam je namreč letnica zapisana s trimestno številko.

Prikaz porazdelitve (glej Sliko 48) vrednosti atributa LETOROJ razkrije še eno nelogičnost:

*Slika 50: Histogram porazdelitve atributa LETOROJ*

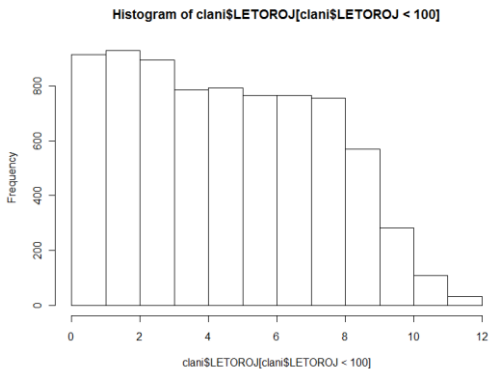
```
> hist(clani$LETOROJ)
```




Očitno so med podatki tudi vrednosti, ki so večje od 0 in manjše od 100. Ena od možnih razlag je, da je mogoče med člani nekdo, ki je rojen leta 2000, vendar se je vrednost LETOROJ vpisala kot »000«. Če to ni tako, potem se bomo morali kasneje odločiti, kaj narediti s temi vrednostmi. Iz histograma (glej Sliko 51) je razvidno, da bi lahko imeli člane, ki so dejansko rojeni med letoma 2000 in 2012, čeprav je malo nenavadno, da se je v knjižni klub včlanil nekdo, ki je danes star na primer 2 leti.

*Slika 51: Histogram vrednosti atributa LETOROJ < 100*

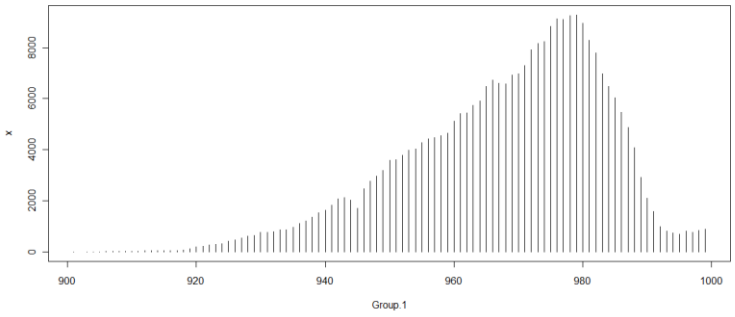
```
> hist(clani$LETOROJ[clani$LETOROJ < 100])
```



Podobno lahko naredimo analizo za vrednosti atributa LETOROJ (glej Sliko 52), ko so te večje od 900:

*Slika 52: Histogram vrednosti atributa LETOROJ > 900*

```
> letoroj900 <- clani$LETOROJ[clani$LETOROJ >= 900]
> letoroj900.agg <- aggregate(letoroj900, by = list(letoroj900), FUN = length)
> plot(letoroj900.agg, type = "h")
```




Tu so vrednosti vsekakor bolj logične. Praktično enak rezultat nam daje analiza z boxplot grafom, ki se nahaja v Prilogi 7.

3.3.1.2.1.10 Atribut TOSTEL

Atribut TOSTEL je zelo podoben atributu POSTA. Nekaj vrednosti ima zelo veliko pojavitev, medtem, ko se nekatere vrednosti pojavljajo zelo redko (glej Sliko 53):

*Slika 53: Osnovni podatki porazdelitve atributov TOSTEL*

```
> summary(clani$TOSTEL)
    2     1     -    41     3    31     5    40     4     7    51    70   599
54044 52269 37111 36046 32533 29516 24885 22258 21325 18172  8066  2681   932
   30   590   591    64   592     8   597    68    81   838
  770   623   270   103    75    28    15     8     3     3
```

3.3.1.2.2 Podatki o naročilih - analiza tabele SRC_NAROCILA

Podobno kot pri podatkih o članih knjižnega kluba kreiramo enostaven model (54) podatkovnega rudarjenja z namenom analize podatkov v tabeli SRC_NAROCILA.

*Slika 54: Data mining model – Explore SRC_NAROCILA*

Osnovne statistike atributov tabele SRC_NAROCILA se nahajajo v tabeli Analiza tabele SRC_NAROCILA. Tabela se nahaja v Prilogi 8.

#### 3.3.1.2.2.1 Atribut NAROCIL

Atribut NAROCIL je identifikacijska številka naročila, ki je bilo vneseno v zaledno aplikacijo. Za potrebe podatkovnega rudarjenja ta atribut ni posebej uporaben (razen v primeru analize nakupne košarice), saj imajo (skoraj) vsi zapisi različne vrednosti in ga lahko zaradi tega izpustimo.

#### 3.3.1.2.2.2 Atribut VRSTANAR

Atribut VRSTANAR ima samo dve vrednosti, N31 in N32, prodajni kanal, preko katerega je bilo prodajno naročilo vneseno v sistem. S funkcijo aggregate() ugotovimo (55), da je skoraj 60 % vseh naročil vnesenih v klubskem centru, ostala naročila so bila vnešena v sistem preko drugih kanalov. Vsebina atributa je predstavljena v tabeli, ki se nahaja v Prilogi 9.

*Slika 55: Osnovni podatki porazdelitve atributa VRSTANAR*

```
> n.vrstanar <- aggregate(SRC_NAROCILA$VRSTANAR, by =
list(SRC_NAROCILA$VRSTANAR), FUN = length)
> n.vrstanar
  Group.1       x
1     N31 1587572
2     N32 2100708
```

#### 3.3.1.2.2.3 Atribut NACINPRID

Atribut NACINPRID predstavlja informacijo o načinu, kako je bilo naročilo pridobljeno. Vsebina tega atributa je podrobneje pojasnjena v tabeli SRC_PRIDOBITEV, ki se nahaja v Prilogi 10.

Porazdelitev atributa NACINPRID je prikazana na Sliki 56:

*Slika 56: Osnovni podatki porazdelitve atributov NACINPRID*

```
> n.nacinprid <- SRC_NAROCILA$NACINPRID
> summary(n.nacinprid)
     33      30      37      35      34      36      40      32      38      46
2100829  598152  341287  277407  130331  114089   35165   29034   28653   17318
     39      31      44       -      41
   8471    7331      75      69      69
```

Na prvi pogled izstopa število vrednosti nakupov v klubskem centru, ki je zelo podobno številu pojavitev vrednosti N32 atributa VRSTANAR. Njuno soodvisnost bi lahko

preverili v nadaljevanju pri redukciji podatkov modela. Pričakovati je seveda visoko stopnjo soodvisnosti in s tem možnost, da atribut VRSTANAR izločimo.

#### 3.3.1.2.2.4 Atributi DATUMNAK, LETONAK in OBDNAK

Vsi trije atributi označujejo časovno dimenzijo nakupa, ki jo lahko predstavimo s hierarhijo LETONAK (leto) – OBDNAK (kvartal) – DATUMNAK (datum). LETONAK in OBDNAK sta atributa, ki označujeta leto in obdobje (kvartal) nakupa ter sta v hierarhiji neposredno nadrejena atributu DATUMNAK.

DATUMNAK verjetno predstavlja preveč podroben nivo informacije, da bi ga lahko uporabili v algoritmih podatkovnega rudarjenja, a ga lahko vseeno koristno uporabimo, če ga prevedemo v podatek, ki označuje pravočasnost nakupa v skladu s klubskimi pravili. Ta pravila določajo, da mora vsak član vsak kvartal opraviti vsaj en nakup. Kot poseben mejnik se uporablja 20. dan drugega meseca v kvartalu, ko se izvaja akcija opominjanja članov, ki še niso izpolnili klubske obveznosti v kvartalu. Na ta način lahko vpeljemo dodatno informacijo o tem, kako pravočasno posamezni člani opravljajo svojo obveznost do knjižnega kluba oziroma jih je treba k temu posebej spodbujati.

#### 3.3.1.2.2.5 Atribut CLAN

CLAN je atribut, ki služi za identifikacijo kupca in se uporablja za integracijo s tabelo SRC_CLANI. Za potrebe priprave podatkov za izvedbo podatkovnega rudarjenja je atribut načeloma brez posebne vrednosti in ga bomo v algoritmih podatkovnega rudarjenja izpustili.

#### 3.3.1.2.2.6 Atribut ARTIKEL

Atribut ARTIKEL vsebuje informacijo o kodi izdelka na naročilu. Po svoji naravi je ta atribut nominalni atribut, ki služi predvsem za identifikacijo izdelka. V primeru analize prodajne košarice (angl. *market basket analysis*) je to ključni atribut modela.

*Slika 57: Osnovni podatki porazdelitve atributa ARTIKEL*

```
> n.artikel <- SRC_NAROCILA$ARTIKEL
> head(n.artikel)
[1] 3830008669650 9788611162751 3850125708537 9788611164304 9789612314361
[6] 3831017600658
10581 Levels: 0000077200167 0000772007269 0008521124113 0008521313456 ...
MKZ0000001547
```

Iz te enostavne statistike (57) vidimo, da je vseh artiklov na naročilih 10.581, kar je razvidno tudi iz tabele Analiza tabele SRC_NAROCILA, ki se nahaja v Prilogi 8.

#### 3.3.1.2.2.7 Atribut GRUPAART

GRUPAART je prvi od treh načinov klasifikacije artiklov. Atribut ima naslednjo porazdelitev (58):

*Slika 58: Osnovni podatki porazdelitve atributa GRUPAART*

```
> n.grupaart <- SRC_NAROCILA$GRUPAART
> summary(n.grupaart, 20)
    258     293     270     500     288     325     508     259     267     330
213
 331057  322934  215238  212487  200144  173726  157027  151257  127809  121470
117084
    277     201     203     246     296     278     266     268 (Other)
 101575   93016   81875   77212   75632   68559   62783   60517  936878
```

Šifrant vsebuje 137 (59) grup artiklov, pri čemer te niso organizirane hierarhično.

*Slika 59: Osnovni podatki porazdelitve atributa GRUPART - nadaljevanje*

```
> head(n.grupaart)
[1] 325 208 288 241 500 325
137 Levels: 201 202 203 204 205 206 207 208 209 210 211 213 214 215 216 217 218
... 999
```

#### 3.3.1.2.2.8 Atribut GRUPANAK

Drugi atribut, s katerim v knjižnem klubu klasificirajo artikle, je nakupna grupa GRUPANAK.

*Slika 60: Osnovni podatki porazdelitve atributa GRUPANAK*

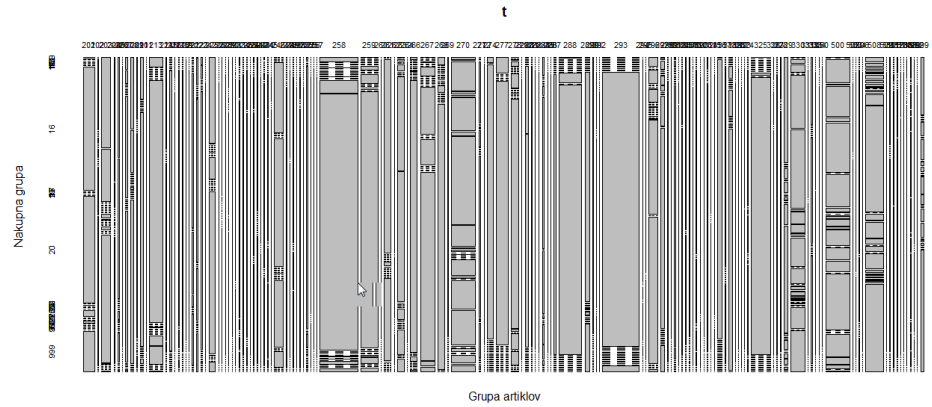
```
> n.grupanak <- SRC_NAROCILA$GRUPANAK
> summary(n.grupanak, 20)
     20      22      24       2      10      15      26       8       3     999
16
 483051  423978  361977  249911  184479  174075  165993  161566  151170  132547
131406
      -      29      25      17      18      27      23      13 (Other)
 129698  116060  115414   76439   71457   70920   67763   56975  363401

> head(n.grupanak,20)
 [1] 22 3  22 1  15 22 27 22 29 3  3  3  2  2  3  22 2  26 3  2
31 Levels: - 1 10 11 12 13 14 15 16 17 18 19 2 20 21 22 23 24 25 26 27 28 29 3 4
5 ... 999
```

Ta šifrant ima 31 različnih vrednosti (60).

Atributa GRUPANAK in GRUPAART sta oba nominalna atributa, zato lahko preverimo njuno soodvisnost s pomočjo grafa (61):

*Slika 61: Korelacija med atributoma GRUPAART in GRUPANAK*

```
> ga <- STG_NAROCILA$GRUPAART
> gn <- STG_NAROCILA$GRUPANAK
> t <- table(ga,gn)
> plot(t, xlab = "Grupa artiklov", ylab = "Nakupna grupa")
```




Graf (61) ne kaže na soodvisnost med atributoma GRUPANAK in GRUPAART. Zato kasneje enega ali drugega atributa ne bomo mogli kar tako izpustiti.

### 3.3.1.2.2.9 Atributi VRSTAKL1, VRSTAKL2 in VRSTAKL3

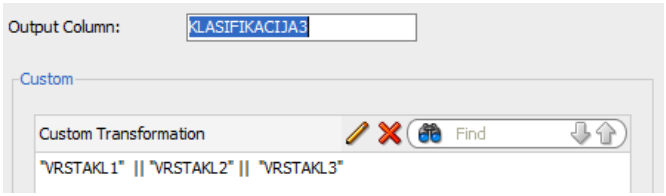
Klasifikacija atributov je tretji način razvrščanja artiklov. Med seboj so povezani v hierarhični strukturi VRSTAKL1 – VRSTAKL2 – VRSTAKL3. Podrobnejši opis vsebine posameznih atributov se nahajava v prilogah (Priloga 11, Priloga 12 in Priloga 13).

Iz vsebine tabel je razvidno, da šele kombinacija atributov VRSTAKL1, VRSTAKL2 in VRSTAKL3 enoznačno določa vrsto klasifikacije. Sami atributi so medsebojno povsem neodvisni, vendar če jih obravnavamo v kombinacijah, dobimo enoznačno opredeljene klasifikacije izdelkov, ki so glede na nivo hierarhije naslednji:

- KLASIFIKACIJA1 = VRSTAKL1,
- KLASIFIKACIJA2 = VRSTAKL1 || VRSTAKL2 in
- KLASIFIKACIJA3 = VRSTAKL1 || VRSTAKL2 || VRSTAKL3.

Za nadaljnjo analizo podatkov moramo najprej narediti zgoraj opisane transformacije podatkov na osnovni tabeli SRC_NAROCILA (glej Sliko 62):

*Slika 62: Kreiranje novega atributa KLASIFIKACIJA3*

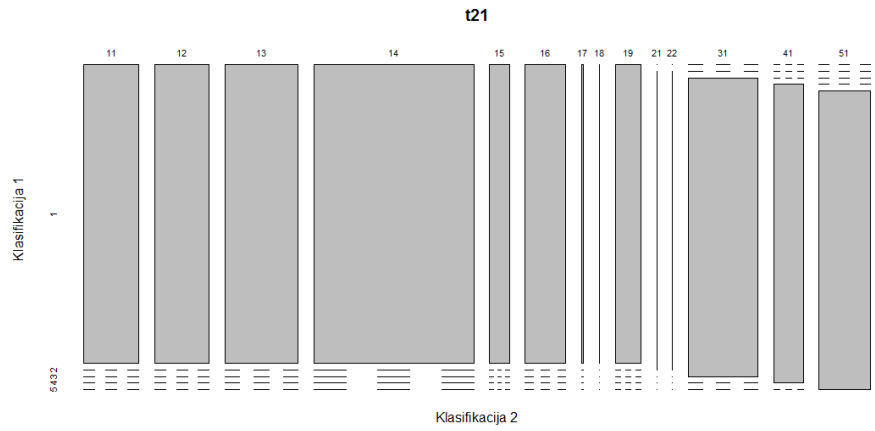


Ko kreiramo vse tri nove atribute KLASIFIKACIJA1, KLASIFIKACIJA2 in KLASIFIKACIJA3, lahko preverimo njihovo soodvisnost.

Graf (glej Sliko 63) prikazuje močno koreliranost atributov KLASIFIKACIJA1 in KLASIFIKACIJA2:

*Slika 63: Korelacija med atributoma KLASIFIKACIJA2 in KLASIFIKACIJA1*

```
> k1 <- STG_NAROCILA$KLASIFIKACIJA1
> k2 <- STG_NAROCILA$KLASIFIKACIJA2
> t21 <- table(k2,k1)
> t21
    k1
k2         1       2       3       4       5
  11  352545       0       0       0       0
  12  344395       0       0       0       0
  13  463772       0       0       0       0
  14 1014275       0       0       0       0
  15  125531       0       0       0       0
  16  256132       0       0       0       0
  17   14539       0       0       0       0
  18    2403       0       0       0       0
  19  162638       0       0       0       0
  21       0      62       0       0       0
  22       0     244       0       0       0
  31       0       0  441536       0       0
  41       0       0       0  187022       0
  51       0       0       0       0  323186

> plot(t21, type="p", xlab='Klasifikacija 2', ylab='Klasifikacija 1')
```




Tudi test $\chi^2$ (64) nam pokaže korelacijo med obema atributoma:

*Slika 64: Preverba korelacije s testom $\chi^2$*

```
> chisq.test(t21)

        Pearson's Chi-squared test

data:  t21
X-squared = 14753120, df = 52, p-value < 2.2e-16
```

Podoben rezultat dobimo, če testiramo soodvisnost z atributom KLASIFIKACIJA3.

Glede na ugotovitve lahko zaključimo, da ne bo velike škode, če bomo pri pripravi podatkov izpustili atributa KLASIFIKACIJA1 in KLASIFIKACIJA2.

#### 3.3.1.2.2.10 Atribut KOLICINA in PCENAKK

Atributa označujeta količino in prodajno ceno izdelka na naročilu. V nadaljevanju bomo preverili možnosti nadomeščanja atributov z drugimi atributi. Ta dva atributa sta kandidata za takšno zamenjavo, saj bi lahko vpeljali atribut vrednost naročila, s čimer ta dva atributa postaneta odveč.

#### 3.3.1.2.2.11 Atribut STORFAK

Atribut STORFAK označuje, da je bilo naročilo stornirano. Za potrebe demonstracije priprave podatkov za podatkovno rudarjenje bo ta podatek izpuščen.

### 3.3.2 Priprava podatkov

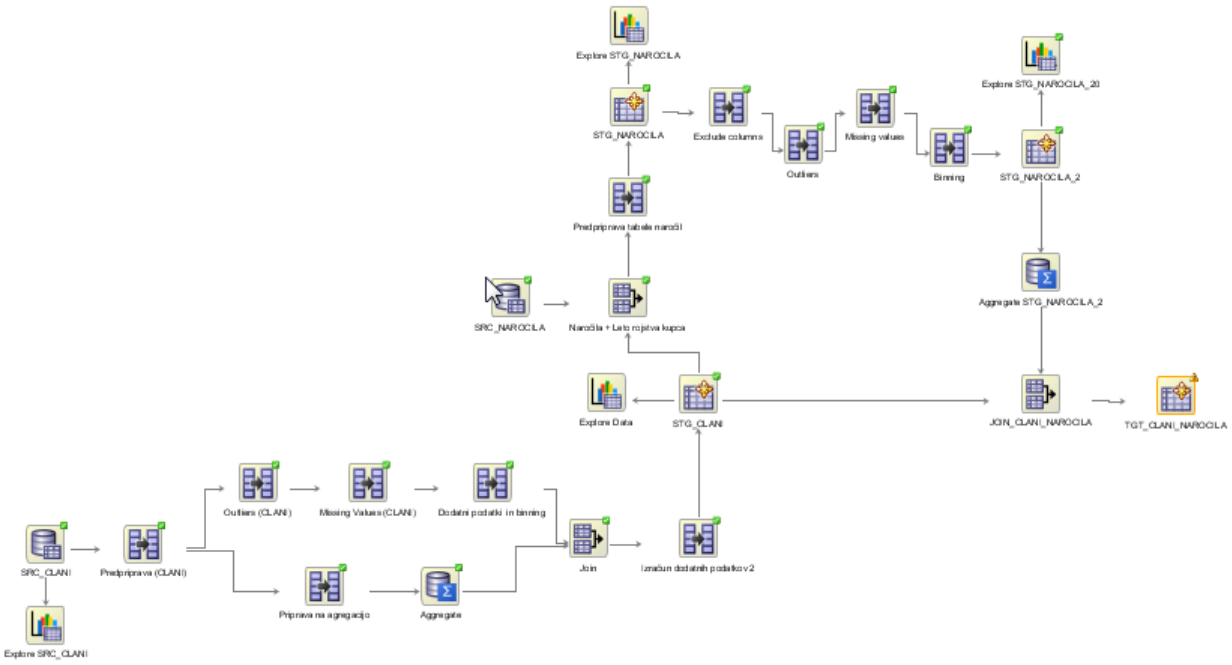
#### 3.3.2.1 Proces priprave podatkov

Proces priprave podatkov (glej Sliko 65) je zaporedje korakov, v katerih izvajamo posamezne operacije nad podatki. Za njegovo izdelavo uporabimo orodje Oracle Data Miner, s katerim lahko grafično opredelimo celoten proces in posamezne korake opisno opredelimo, pri čemer ni nujno poznavanje SQL ukaznega jezika, v katerem so sicer generirane vse potrebne procedure.

V prvem delu procesa poteka priprava podatkov članov kluba, v nadaljevanju pa se posebej pripravi podatke o naročilih, ki se jih v zadnjem koraku združi s podatki o članstvu tako, da obstaja za vsakega člana knjižnega kluba en sam zapis, ki vključuje tako njegove osebne podatke, kot podatke o njegovih nakupih.

Glede na potrebe in zahteve posameznih algoritmov podatkovnega rudarjenja sledijo tej osnovni pripravi še dodatni koraki, kot je na primer priprava nabora podatkov za vzorčenje, priprava testne množice podatkov in podobno.

*Slika 65: Proces priprave podatkov v primeru knjižnega kluba*




### 3.3.2.2 Priprava podatkov o članih

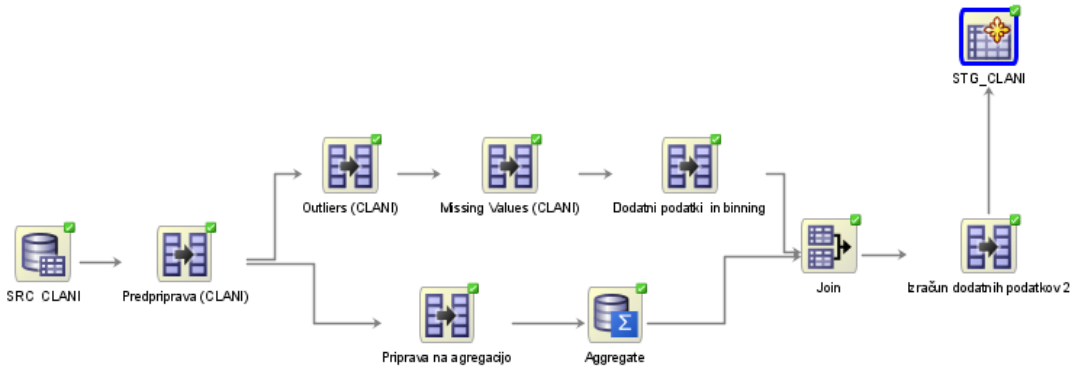
Proces priprave podatkov o članih (glej Sliko 66) vključuje kar nekaj korakov transformacij, v katerih so uporabljene naslednje operacije:

- kreiranje novih atributov,
- agregacija,
- identifikacija izstopajočih vrednosti in njihovo nadomeščanje,
- nadomeščanje manjkajočih vrednosti,
- razvrščanje v grupe.

Vir podatkov za proces priprave podatkov o članih, kjer se proces priprave začenja, je tabela SRC_CLANI, ki je analizirana v prejšnjih poglavjih.

Proces priprave podatkov o članih se zaključuje z zapisom podatkov v začasno tabelo STG_CLANI, ki jo lahko pred nadaljevanjem procesa priprave podatkov analiziramo podobno kot SRC_CLANI in na osnovi ugotovitev analize izvedemo dodatne transformacije podatkov, če so te potrebne.

*Slika 66: Priprava podatkov o članih*




Za potrebe te naloge predpostavimo, da je ta del podatkov primerno pripravljen.

#### 3.3.2.2.1 Izločitev nepotrebnih atributov

Za nekatere podatke smo v fazi analize podatkov ugotovili, da jih ne bomo potrebovali, zato jih izločimo takoj na začetku. Gre za atribute FIZOSEBA, OBDZAC, OBDZADZEL in PTTNAZIV. Po potrebi bodo izločeni v kasnejših korakih tudi nekateri drugi atributi.

#### 3.3.2.2.2 Enostavna transformacija atributa NACIZP

V zalogi vrednosti atributa NACIZP je tudi vrednost NULL, ki v tem primeru pomeni manjkajočo vrednost.

Glede na posebnosti, kako obravnava Oracle Data Miner manjkajoče podatke, lahko označimo manjkajoče podatke atributa kot nominalne, naključno manjkajoče. Neobstoj tega podatka pravzaprav pomeni, da imamo opraviti s še aktivnim članom knjižnega kluba. Zato je vprašanje, če je pravilno, da odločanje o tem, kako naj se nadomesti manjkajoči podatek, prepustimo algoritmu. Verjetno ne. Pravzaprav obstaja pravilo, s pomočjo katerega lahko dejansko ugotovimo, ali je nekdo še vedno aktiven član ali ne.

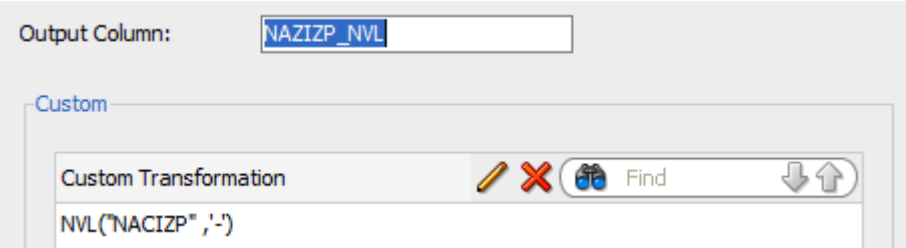
V zalogi vrednosti atributa NACIZP je tudi vrednost '–'. Tudi ta vrednost označuje, da je nek član še vedno aktiven. Skupaj je tako aktivnih članov še 52.146 (glej Sliko 67):

*Slika 67: Število aktivnih članov društva*

```
> select nacizp, count(clan) from src_clani where nacizp IN ('-') or nacizp IS
NULL group by nacizp order by count(clan) DESC;
NACIZP        COUNT(CLAN)
-             40259
(null)        11887

> select count(clan) from src_clani where nacizp IN ('-') or nacizp IS NULL;
COUNT(CLAN)
52146
```

Vrednosti NULL zato v koraku Predpriprava (CLANI) nadomestimo z vrednostjo '–' (glej Sliko 68).

*Slika 68: Nadomeščanje vrednosti NULL v primeru atributa NACIZP_NVL*

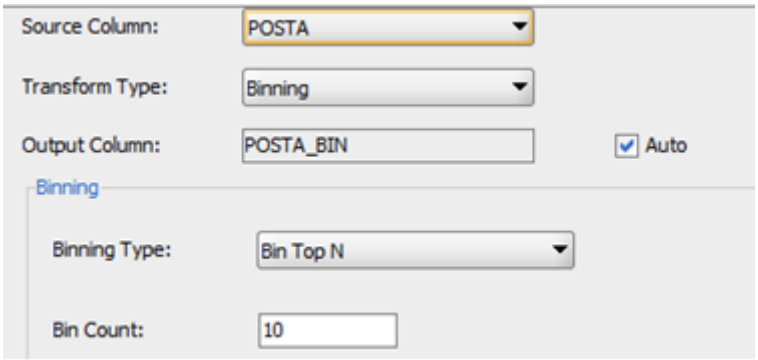



3.3.2.2.3 Razvrščanje v grupe na primeru atributov POSTA, TOSTEL in NACPRI

Zaloga vrednosti atributa POSTA predstavlja 632 različnih vrednosti. Od tega jih je kar 481 takšnih, ki imajo manj kot 500 zapisov, in od teh jih je celo kar 121 takšnih, ki imajo le en zapis, kar za algoritem podatkovnega rudarjenja nima praktično nobene vrednosti.

Problem prevelike razdrobljenosti atributa POSTA lahko razrešimo z razvrščanjem v grupe. Za razvrstitev v grupe uporabimo operacijo za razvrščanje v grupe na osnovi 10 najpogostejših grup (glej Sliko 69).

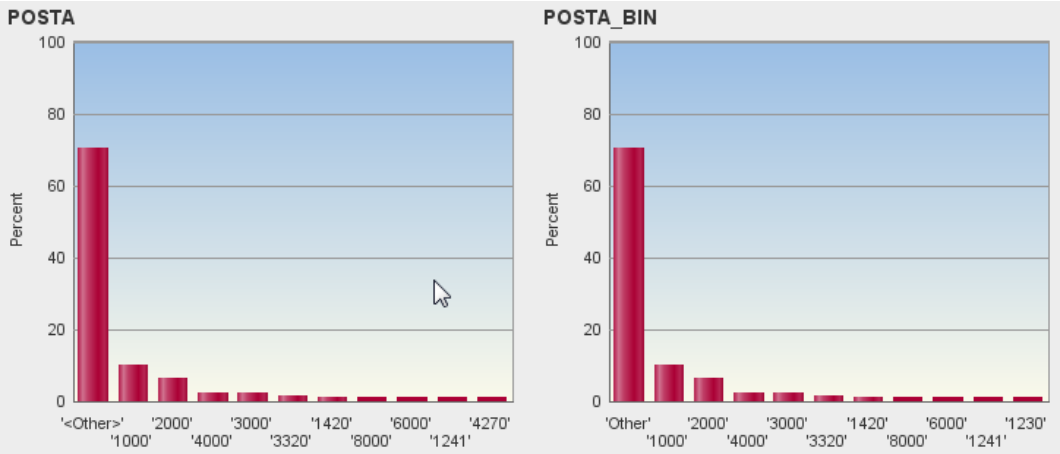
*Slika 69: Razvrščanje v grupe na primeru atributa POSTA*




Rezultat razvrščanja v grupe je nov atribut POSTA_BIN (glej Sliko 70).

Glede na število pojavitev posamezne vrednosti atributa POSTA_BIN je bilo kreiranih 10 grup in še dodatna grupa 'Other' (v primeru porazdelitve atributa POSTA pomeni '<Other>' le navidezno grupo, v katero so razvrščene vse ostale pošte), v katero se razvrstijo vse ostale vrednosti atributa POSTA, ki niso razvrščene v eno od 10 najpogostejših grup.

*Slika 70: Primerjava porazdelitev atributov POSTA in POSTA_BIN*

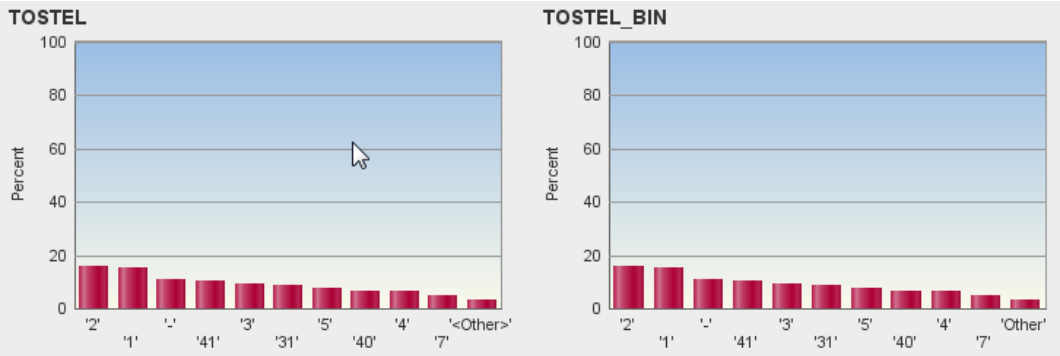



Atribut POSTA je s kreiranjem atributa POSTA_BIN odveč, zato ga izločimo iz nadaljnje obravnave.

Zelo podobne statistike, kot jih ima atribut POSTA, so statistike atributa TOSTEL. Tudi slednji ima v nekaterih grupah zelo malo primerov pojavitev vrednosti atributa. Na primer, omrežna številka 590 ima le 623 pojavitev.

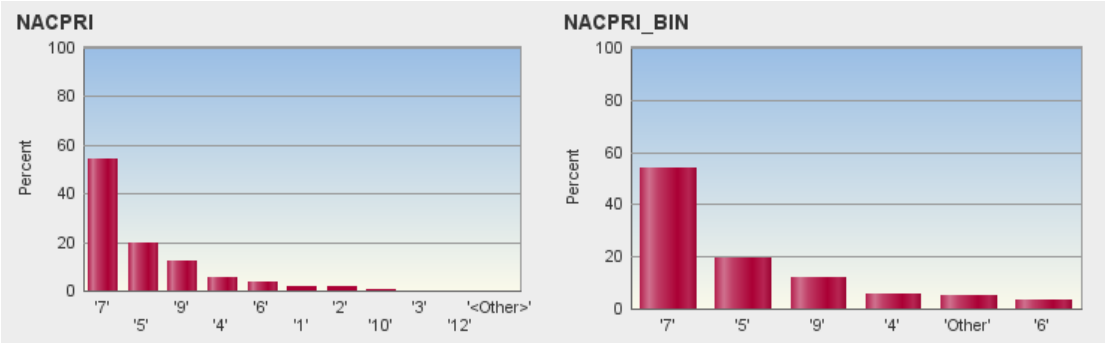
Podobno, kot smo kreirali nov atribut POSTA_BIN, kreiramo nov atribut TOSTEL_BIN, za kar lahko prav tako uporabimo razvrščanje v grupe na osnovi 10 najpogostejših grup (glej Sliko 71):

*Slika 71: Razvrščanje v grupe na osnovi 10 najpogostejših pojavitev atributa TOSTEL*




Tretji atribut, za katerega uporabimo transformacijo razvrščanja v grupe, je atribut NACPRI, ki ga razdelimo na 5 grup in dodatno grupo 'Other' (glej Sliko 72):

*Slika 72: Porazdelitev novega atributa NACPRI_BIN v primerjavi z atributom NACPRI*



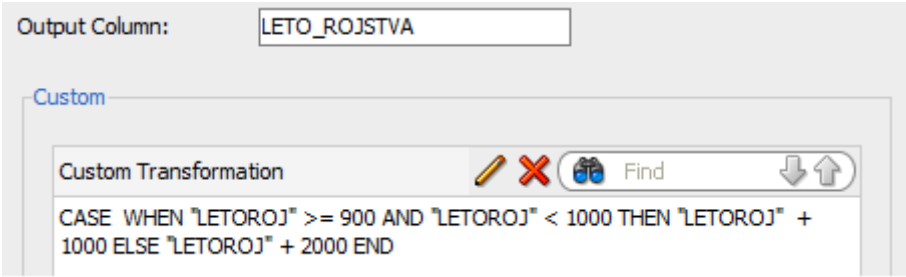

3.3.2.2.4 Transformacija atributa LETOROJ

Transformacija podatkov o letu rojstva poteka v več korakih. Najprej prevedemo vse vrednosti v zalogi vrednosti atributa LETOROJ na vrednosti, ki so smiselne. Nato se znebimo izstopajočih vrednosti ter na koncu nadomestimo vse manjkajoče vrednosti z ustreznimi vrednostmi.

V poglavju Atribut LETOROJ smo ugotovili, da so vrednosti tega atributa iz tehničnih razlogov zapisane brez tisočice, kar ima za posledico porazdelitev večine vrednosti atributa v intervalu [900, 999] in gručo osamelcev v intervalu [0, 12]. Zato je potrebno najprej prevesti vrednosti LETOROJ na skupni imenovalec, to je letnico rojstva, ki vsebuje tudi tisočice.

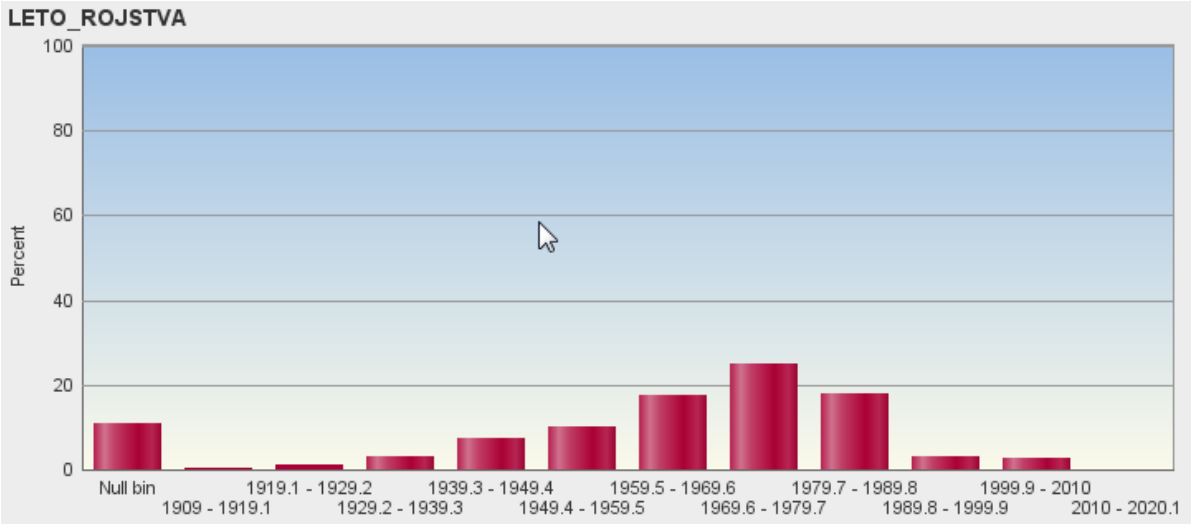
V koraku Predpriprava (CLANI) zato kreiramo nov atribut LETO_ROJSTVA, ki prikazuje vrednost v pričakovani zalogi vrednosti od leta 1900 naprej (glej Sliko 73).

*Slika 73: Transformacija atributa LETOROJ*




Nov atribut ima naslednjo porazdelitev (glej Sliko 74):

*Slika 74: Histogram porazdelitve novega atributa LETO_ROJSTVA*

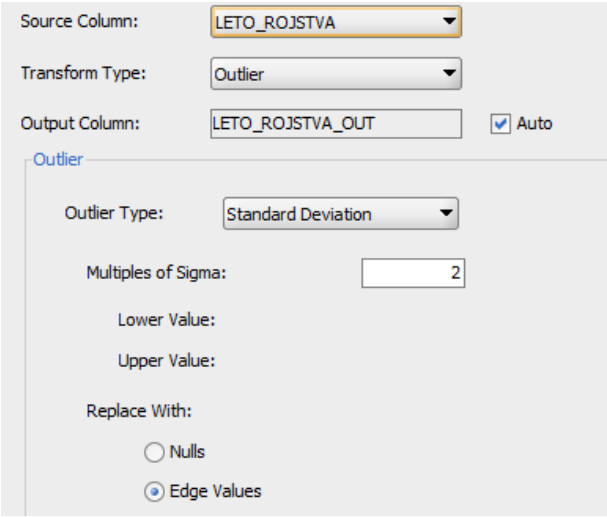



Iz histograma (glej Sliko 74) razberemo, da nov atribut LETO_ROJSTVA še ni povsem pripravljen za podatkovno rudarjenje, saj so nekatere letnice rojstva višje od trenutne letnice (2013), hkrati pa ima več kot 10 % pojavitev zapisa vrednosti NULL.

V tem trenutku atribut LETOROJ ni več potreben in ga izločimo iz nadaljnje obravnave.

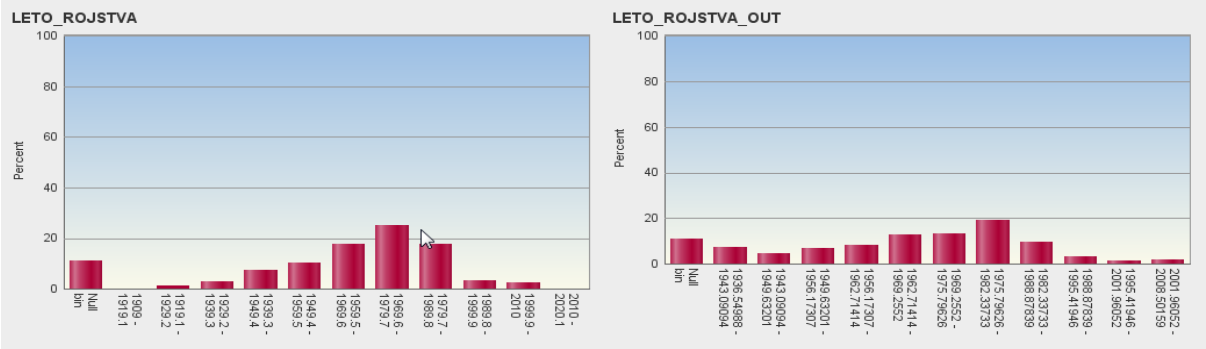
V prvem naslednjem koraku, Outliers (CLANI), bomo izločili izstopajoče vrednosti. V ta namen kreiramo nov atribut LETO_ROJSTVA_OUT (glej Sliko 75), nad katerim izvedemo transformacijo, pri kateri vrednosti izven intervala $[\bar{x} - 2\sigma, \bar{x} + 2\sigma]$, nadomestimo z robnimi vrednostmi tega intervala.

*Slika 75: Zamenjava izstopajočih vrednosti z robnimi vrednostmi porazdelitve*



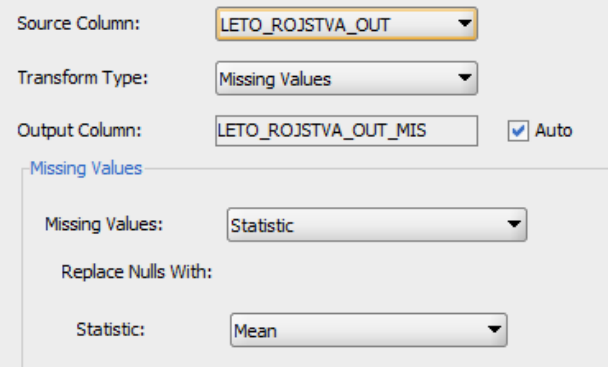

Atribut LETO_ROJSTVA_OUT ima zdaj bolj realno porazdelitev vrednosti (glej Sliko 76):

*Slika 76: Porazdelitev vrednosti atributa LETO_ROJSTVA pred in po transformaciji*



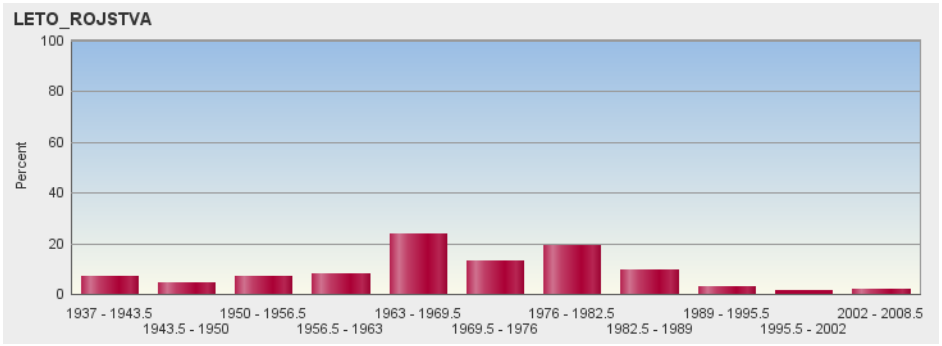

V zadnjem koraku, Missing Values (CLANI), nadomestimo še manjkajoče vrednosti *NULL* z ustreznimi vrednostmi. V našem primeru uporabimo povprečno vrednost atributa LETO_ROJSTVA_OUT (glej Sliko 77).

*Slika 77: Nadomeščanje manjkajočih vrednosti s povprečno vrednostjo*




Na osnovi atributa LETO_ROJSTVA_OUT kreiramo nov atribut LETO_ROJSTVA_OUT_ MIS, katerega opisne statistike se nahajajo v Prilogi 14.

*Slika 78: Porazdelitev atributa LETO_ROJSTVA*



V zadnjem koraku vrednosti LETO_ROJSTVA_OUT_MIS še zaokrožimo (korak Dodatni podatki in Binning). Končni novi atribut LETO_ROJSTVA ima tako naslednjo porazdelitev prikazno na Sliki 78.

#### 3.3.2.2.5 Dilema: leto rojstva ali starost?

V primeru letnic, ne samo leto rojstva, imamo zanimivo dilemo. Kateri podatek pravzaprav potrebujemo: leto rojstva ali starost člana kupca? Ali se bodo na posebno ponudbo nove »kuharice« odzvale članice kluba, ki so rojene med leti 1953 in 1963, ali tiste, ki so danes stare med 50 in 60 let?

Če gledamo na podatke brez zgodovinske perspektive, potem je najbrž vseeno. Če pa upoštevamo podatke o preteklih nakupih, potem so se pred desetimi leti na identično posebno ponudbo dobro odzivale ženske v starostni skupini od 50 do 60 let, vendar so bile rojene med leti 1943 in 1953. Če gradimo napovedni model odzivanja na posebno ponudbo, nam pri uporabi atributa LETO_ROJSTVA, ta ne da prave informacije, saj bi morali po logiki dobrega odziva, nasloviti skupino žensk, ki so danes stare od 60 do 70 let. Če uporabimo atribut STAROST, nam lahko napovedni model napove, da moramo ponudbo poslati ženskam, ki so danes stare od 50 do 60 let.

Podobno razmišljanje lahko prenesemo na starost, ko se nekdo odloči za vpis v knjižni klub in pri kateri starosti se člani najpogosteje odločajo za izpis.

Prav tako je najbrž dobro vedeti, v katerem starostnem obdobju kupujejo člani več in kdaj ne. Na primer, verjetno je veliko višja verjetnost, da člani kluba kupujejo otroško literaturo v starosti od 25 do 35 let, saj je to obdobje, ko imajo majhne otroke. Da v tem primeru poznamo leto njihovega rojstva, nam ne pomaga prav nič, saj se starostna skupina spreminja vsako leto.

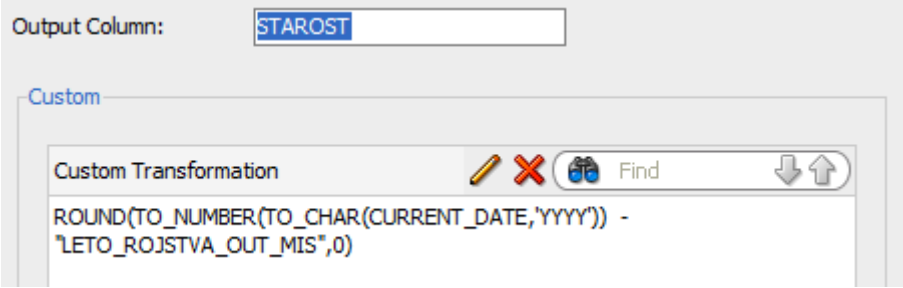
Zato je smiselno v naš model vpeljati podatek o starosti, ko je prišlo do določenega dogodka, saj nam ta podatek daje možnost primerjave preko različnih časovnih obdobij.

Seveda to ne pomeni, da se podatek LETO_ROJSTVA enostavno izpusti. V primeru knjig, ki se recimo nanašajo na točno določeno obdobje (na primer knjige o predsedniku Titu), je povsem mogoče, da jih kupujejo v večjem številu predvsem člani kluba, ki so rojeni v obdobju pred letom 1980. In ti so vsako leto starejši, kar pomeni, da si s podatkom STAROST ne moremo veliko pomagati.

#### 3.3.2.2.6 Nov atribut STAROST

STAROST je nov atribut, ki ga kreiramo (glej Sliko 79) kot razliko med tekočim letom in letom rojstva člana kluba:

*Slika 79: Izračun novega atributa STAROST*




Nov atribut ima naslednjo porazdelitev (glej Sliko 80).

*Slika 80: Histogram porazdelitve novega atributa STAROST*

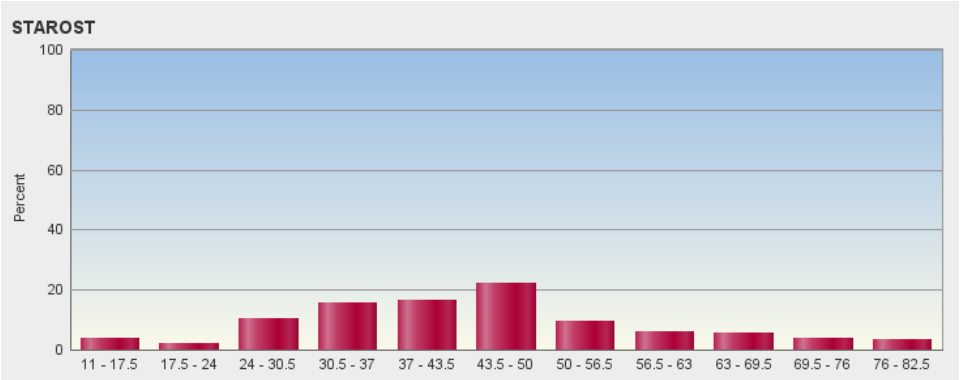
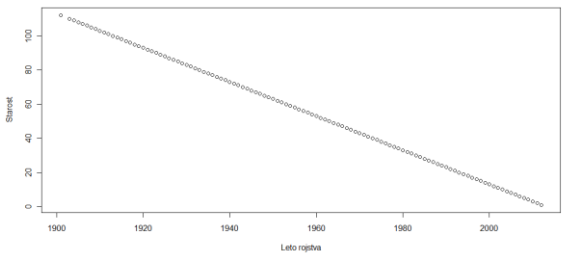



Tabela z opisnimi statistikami se nahaja v Prilogi 15.

*Slika 81: Negativna koreliranost med atributoma LETO_ROJSTVA in STAROST*




LETO_ROJSTVA in STAROST sta seveda negativno korelirana. Višja, ko je vrednost atributa LETO_ROJSTVA, manjša je vrednost atributa STAROST, kar je razvidno tudi iz zgornjega grafa (glej Sliko 81).

V poglavju Redukcija podatkov smo prav s pomočjo korelacije identificirali odvečne atribute, da bi zmanjšali kompleksnost modela, ki ga pripravljamo za podatkovno rudarjenje. Zdaj smo v model vpeljali nov atribut, ki je hkrati močno koreliran z že obstoječim. Vendar je zaradi poslovnih razlogov njegova uvedba smiselna, saj lahko na njegovi osnovi dobimo informacije o tem, v kateri starosti člani kupujejo neko določeno skupino izdelkov.

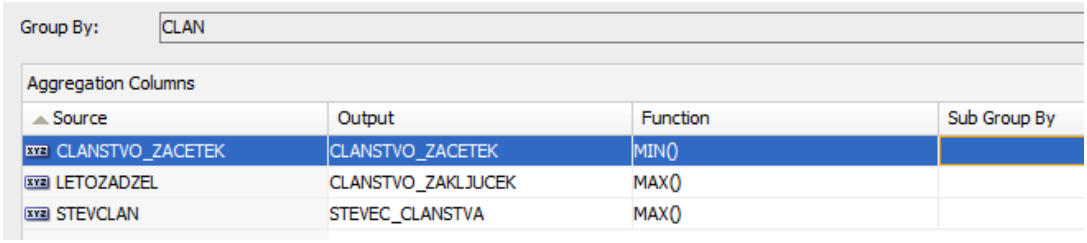
#### 3.3.2.2.7 Primer agregacije in novi atributi o članstvu

V množici podatkov imamo lahko v primeru večkratnega izpisa in ponovnega včlanjevanja istega kupca, več zapisov o njem. V našem primeru nas zanima le letnica prvega vpisa oziroma zadnjega izpisa, če ta obstaja, ter skupno število ponovnih včlanitev.

S pomočjo agregatnih funkcij MIN() in MAX() lahko izračunamo (glej Sliko 82):

- kdaj je nekdo prvič vstopil v knjižni klub,
- kdaj se je zadnjič izpisal (ali pa je še aktiven) in
- kolikokrat je bil včlanjen.

*Slika 82: Primer uporabe agregacije podatkov*




Na osnovi teh novih atributov lahko kreiramo še druge, kot je na primer trajanje članstva, starost ob prvi vključitvi in starost ob zadnjem izstopu iz kluba, če član ni več aktiven član kluba. V primeru, da je član še aktiven, ima atribut CLANSTVO_ZAKLJUCEK vrednost NULL in iz njega izpeljemo nov atribut AKTIVEN_CLAN.

V kombinaciji z atributom LETO_ROJSTVA je mogoče izračunati starost člana ob vpisu in izpisu iz kluba, če ni več aktiven član (glej Tabelo 4).

*Tabela 4: Novi atributi o članih na osnovi agregiranih podatkov*

| Atribut | Formula |
|---|---|
| AKTIVEN_CLAN | CASE WHEN "CLANSTVO_ZAKLJUCEK" <= TO_NUMBER(TO_CHAR(CURRENT_DATE,'YYYY')) THEN 'N' ELSE 'Y' END |

se nadaljuje

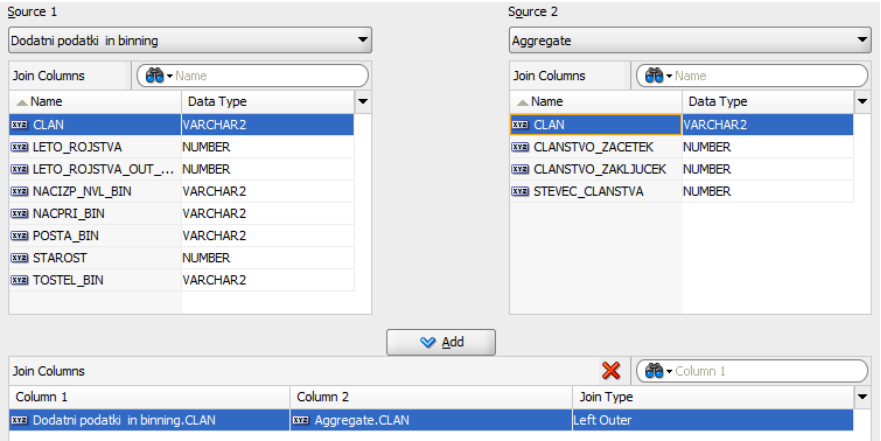
nadaljevanje

| Atribut | Formula |
|---|---|
| CLANSTVO_TRAJANJE | CASE WHEN "CLANSTVO_ZAKLJUCEK" >= "CLANSTVO_ZACETEK" THEN (CASE WHEN "CLANSTVO_ZAKLJUCEK" <= TO_NUMBER(TO_CHAR(CURRENT_DATE,'YYYY')) THEN "CLANSTVO_ZAKLJUCEK" - "CLANSTVO_ZACETEK" ELSE TO_NUMBER(TO_CHAR(CURRENT_DATE,'YYYY')) - "CLANSTVO_ZACETEK" END) ELSE 0 END |
| STAROST_ZACETEK | "CLANSTVO_ZACETEK" - "LETO_ROJSTVA" |
| STAROST_ZAKLJUCEK | "CLANSTVO_ZAKLJUCEK" - "LETO_ROJSTVA" |

#### 3.3.2.2.8 Združevanje podatkov (operacija JOIN)

V modelu imamo kar nekaj primerov operacije JOIN, pri kateri na osnovi istega atributa združujemo različne množice podatkov. V prvem delu, kjer imamo primer priprave podatkov o članih je prva pojavitev te operacije v koraku JOIN (glej Sliko 83).

*Slika 83: Primer združevanja podatkov z operacijo JOIN*




Na podoben način je izpeljano združevanje podatkov tudi v drugih primerih združevanja podatkov v modelu. Atribut, ki je ves čas »vodilni atribut združevanja«, je atribut CLAN.

#### 3.3.2.3 Priprava podatkov o naročilih in združitev podatkov s kupci

Izhodišče priprave podatkov o naročilih (glej Sliko 84) sta dve tabeli, in sicer:

- tabela naročil SRC_NAROCILA, ki smo jo analizirali v poglavju, Analiza podatkov, kjer smo analizirali podatke o naročilih, in
- pripravljena tabela kupcev STG_CLANI, ki smo jo pripravili v poglavju Priprava podatkov o kupcih.

*Slika 84: Proces priprave podatkov o naročilih in združitev podatkov s kupci*

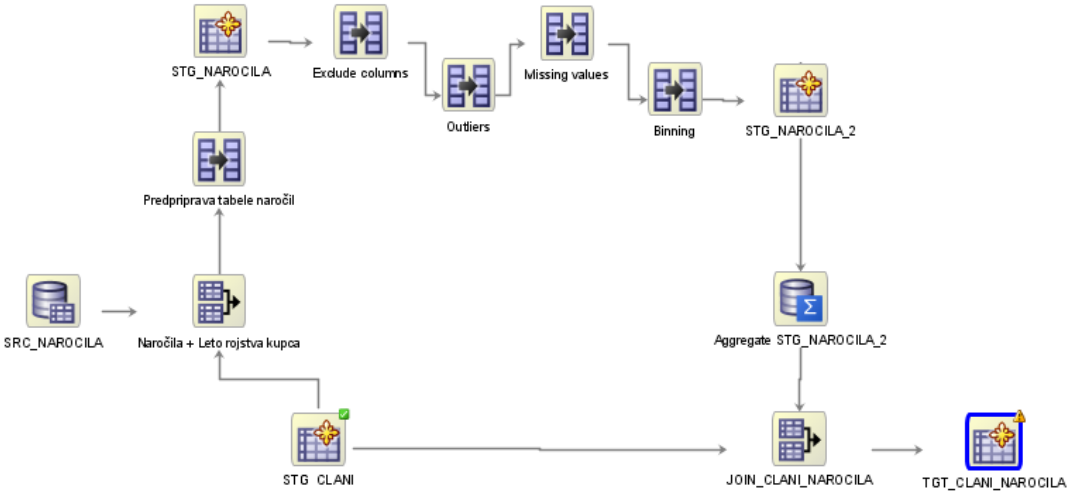



Tabela STG_CLANI se v proces vključuje na začetku, kjer podatkom o naročilih (iz tabele SRC_NAROCILA) pridružimo podatek o letu rojstva kupca, ki je potreben pri izračunavanju starosti kupca ob nakupu.

Na koncu procesa priprave podatkov o naročilih združimo podatke o članih in naročilih v tabelo TGT_CLANI_NAROCILA, s čimer se proces priprave osnovne zbirke podatkov zaključi.

Priprava podatkov o naročilih poteka podobno kot priprava podatkov o članih skozi naslednje korake:

- predpriprava tabele naročil,
- priprava podatkov iz transakcij,
- priprava gnezdenih stolpcev.

#### 3.3.2.3.1 Predpriprava tabele naročil

V tem koraku izločimo nekatere nepotrebne atribute, ki smo jih že identificirali za izločitev v fazi analize podatkov naročil.

Prav tako se v tem koraku izvede nekatere osnovne transformacije podatkov in kreira nove atribute (glej Tabelo 5):

*Tabela 5: Novi atributi o naročilih na osnovi agregiranih podatkov*

| Atribut | Formula |
|---|---|
| KLASIFIKACIJA1 | "VRSTAKL1" |
| KLASIFIKACIJA2 | "VRSTAKL1"\|\| "VRSTAKL2" |
| KLASIFIKACIJA3 | "VRSTAKL1"\|\| "VRSTAKL2"\|\|"VRSTAKL1"\|\| "VRSTAKL3" |
| KVARTAL_NAKUPA | "LETONAK" \|\| '-Q' \|\| "OBDNAK" |
| STAROST_KUPCA | "LETONAK" - "LETO_ROJSTVA" |
| VREDNOST_NAKUPA | "KOLICINA" * "PCENAKK" |

3.3.2.3.2 Koraki: Exclude columns, Outliers, Missing values, Binning

Korak *Exclude columns* sledi zapisu podatkov v vmesno tabelo STG_NAROCILA. Predpripravljene podatke bi morali namreč pred nadaljevanjem analizirati, kot smo to sicer že storili v poglavju Podatki o naročilih – analiza tabele SRC_NAROCILA, ko smo še raziskovali podatke o naročilih.

Tako je analiza na primer pokazala, da obstaja med atributi KLASIFIKACIJA1, KLASIFIKACIJA2 in KLASIFIKACIJA3 velika soodvisnost, zato lahko prva dva atributa izpustimo in ohranimo le atribut KLASIFIKACIJA3.

Podobno ne potrebujemo več atributov LETO_ROJSTVA in LETONAK, ki smo ju uporabili za izračun starosti kupca (STAROST_KUPCA).

Nadaljnji koraki Outliers, Missing values in Binning so identični korakom, ki smo jih uporabili pri atributu STAROST, zato jih na tem mestu ne bi ponavljali. Rezultat teh aktivnosti je nova vmesna tabela STG_NAROCILA_2, katere podrobnejši opis se nahaja v Prilogi 16.

3.3.2.3.3 Uporaba gnezdenih stolpcev na primeru naročil

Osnova za pripravo podatkov o naročilih je vmesna tabela STG_NAROCILA_2. Podatke o naročilih moramo povezati s podatki iz osnovne tabele o članih tako, da jih zapišemo v gnezdene stolpce osnovne tabele.

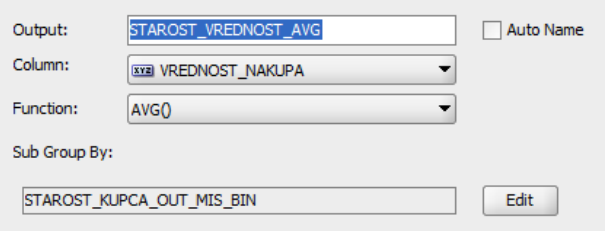
Vendar celotne tabele naročil ne moremo kar enostavno zapisati v osnovno tabelo članov kot gnezden stolpec, razen če bomo uporabili algoritem Apriori, kjer analiziramo le tabelo naročil in tabela članov niti ni potrebna.

Podatke je potrebno pripraviti v obliki paric (atribut, vrednost), ki jih identificiramo z enoznačnim identifikatorjem. V našem primeru je to atribut CLAN. Tako želimo na primer za vsakega posameznega kupca izvedeti:

- vsoto vrednosti nakupov kupca glede na starostno skupino, ki ji je v času nakupa kupec pripadal,
- povprečno vrednost nakupov glede na klasifikacijo izdelkov, ki jih je kupec kupil,
- število nakupov posameznega kupca glede na vrsto naročila in kvartal nakupa.

Pripravo gnezdenih stolpcev omogoča funkcija agregacije (glej Sliko 85). V modelu je to korak Aggregate STG_NAROCILA_2.

*Slika 85: Primer uporabe agregatne funkcije za pripravo gnezdenega stolpca*




Po izvedbi agregacije bo nov stolpec STAROST_VREDNOST_AVG imel za kupca 1186964 naslednje vrednosti (glej Sliko 86):

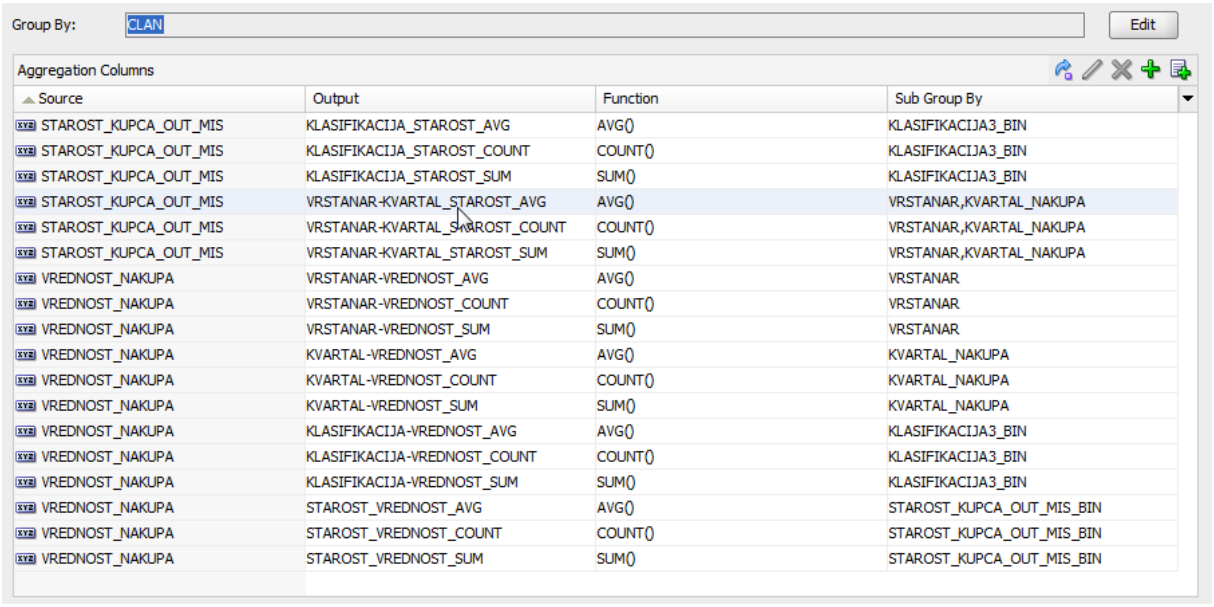
*Slika 86: Primer agregacije v gnezdenem stolpcu*

```
> select STAROST_VRENOST_AVG from temp_table where clan = '1186964';

SYS.DM_NESTED_NUMERICALS
(SYS.DM_NESTED_NUMERICAL(36 - 40, 20.776875),
SYS.DM_NESTED_NUMERICAL(40 - 44, 13.0017647),
SYS.DM_NESTED_NUMERICAL(44 - 52, 15.14167))
```

Naključno izbrani kupec ima nakupe v treh starostnih razredih: '36–40', '40–44' in '44–52' s povprečnimi vrednostmi nakupov 20,78 EUR, 13,00 EUR in 15,14 EUR. Takšen zapis je generiran za vse člane kluba.

Na podoben način opredelimo še ostale parice atribut – vrednost (glej Sliko 87).

*Slika 87: Primer paric (atribut, vrednost) kot osnova za gnezdene stolpce*




| Source | Output | Function | Sub Group By |
|---|---|---|---|
| STAROST_KUPCA_OUT_MIS | KLASIFIKACIJA_STAROST_AVG | AVG() | KLASIFIKACIJA3_BIN |
| STAROST_KUPCA_OUT_MIS | KLASIFIKACIJA_STAROST_COUNT | COUNT() | KLASIFIKACIJA3_BIN |
| STAROST_KUPCA_OUT_MIS | KLASIFIKACIJA_STAROST_SUM | SUM() | KLASIFIKACIJA3_BIN |
| STAROST_KUPCA_OUT_MIS | VRSTANAR-KVARTAL_STAROST_AVG | AVG() | VRSTANAR,KVARTAL_NAKUPA |
| STAROST_KUPCA_OUT_MIS | VRSTANAR-KVARTAL_STAROST_COUNT | COUNT() | VRSTANAR,KVARTAL_NAKUPA |
| STAROST_KUPCA_OUT_MIS | VRSTANAR-KVARTAL_STAROST_SUM | SUM() | VRSTANAR,KVARTAL_NAKUPA |
| VREDNOST_NAKUPA | VRSTANAR-VREDNOST_AVG | AVG() | VRSTANAR |
| VREDNOST_NAKUPA | VRSTANAR-VREDNOST_COUNT | COUNT() | VRSTANAR |
| VREDNOST_NAKUPA | VRSTANAR-VREDNOST_SUM | SUM() | VRSTANAR |
| VREDNOST_NAKUPA | KVARTAL-VREDNOST_AVG | AVG() | KVARTAL_NAKUPA |
| VREDNOST_NAKUPA | KVARTAL-VREDNOST_COUNT | COUNT() | KVARTAL_NAKUPA |
| VREDNOST_NAKUPA | KVARTAL-VREDNOST_SUM | SUM() | KVARTAL_NAKUPA |
| VREDNOST_NAKUPA | KLASIFIKACIJA-VREDNOST_AVG | AVG() | KLASIFIKACIJA3_BIN |
| VREDNOST_NAKUPA | KLASIFIKACIJA-VREDNOST_COUNT | COUNT() | KLASIFIKACIJA3_BIN |
| VREDNOST_NAKUPA | KLASIFIKACIJA-VREDNOST_SUM | SUM() | KLASIFIKACIJA3_BIN |
| VREDNOST_NAKUPA | STAROST_VREDNOST_AVG | AVG() | STAROST_KUPCA_OUT_MIS_BIN |
| VREDNOST_NAKUPA | STAROST_VREDNOST_COUNT | COUNT() | STAROST_KUPCA_OUT_MIS_BIN |
| VREDNOST_NAKUPA | STAROST_VREDNOST_SUM | SUM() | STAROST_KUPCA_OUT_MIS_BIN |

### 3.3.3 Posebnosti priprave podatkov glede na algoritem podatkovnega rudarjenja

#### 3.3.3.1 Priprava podatkov pri izdelavi modela za segmentacijo podatkov

Oracle Data Mining uporablja za segmentacijo podatkov (angl. *clustering*) algoritem razvrščanja z voditelji (angl. *k-means*), ki porazdeljuje podatke v poljubno število gruč na osnovi razdalje do centroida gruče.

Avtomatična priprava podatkov izvede normalizacijo podatkov z upoštevanjem izstopajočih vrednosti (Oracle, 2008, str. 13/2).

V primeru manjkajočih vrednosti pri atributih z enostavnimi podatkovnimi tipi (negnezdeni stolpci), algoritem privzame te kot naključno manjkajoče. Nominalne vrednosti bodo tako zamenjane z modusom, numerične pa s povprečno vrednostjo. V gnezdenih stolpcih algoritem interpretira te manjkajoče vrednosti kot redke. To pomeni, da jih bo v primeru numeričnih atributov nadomestil z ničlami, v primeru nominalnih pa bo algoritem manjkajoče vrednosti zamenjal z ničelnim vektorjem (Oracle, 2008, str. 13/2).

Če ne uporabljamo avtomatične priprave podatkov, je potrebno upoštevati, da lahko izstopajoče vrednosti v primeru razvrščanja v grupe na osnovi enakomernega obsega povzročijo oblikovanje grup, ki se po vsebini ne razlikujejo bistveno (Oracle, 2008, str. 13/2). Z drugimi besedami, oblikovane bodo grupe, ki bodo imele zelo podobne centroide, porazdelitev in pravila razvrščanja.

V naši pripravi podatkov smo poskrbeli, da v podatkih nimamo manjkajočih vrednosti, medtem ko bo v izdelavi modela za segmentacijo uporabljena avtomatična priprava podatkov, ki bo izvedla normalizacijo podatkov.

#### 3.3.3.2 Priprava podatkov pri izdelavi modela za klasifikacijo

Klasifikacija je funkcija podatkovnega rudarjenja, ki priredi posameznemu primeru iz zbirke podatkov njegovo ciljno kategorijo ali razred. Cilj klasifikacije je čim natančneje napovedati ciljni razred za vsak primer iz zbirke primerov (Oracle, 2008, str. 5/1). V primeru knjižnega kluba je to lahko napoved članov, ki bodo v prihodnosti izstopili iz kluba ali napoved odzivanja članov na prodajno akcijo.

Ko pripravljamo podatke za klasifikacijo, moramo upoštevati, da je za izdelavo potrebno pripraviti najmanj tri zbirke podatkov, in sicer:

- zbirko podatkov, ki jo uporabimo za izdelavo napovednega modela,
- zbirko podatkov, ki jo uporabimo za test napovednega modela in
- zbirko podatkov, ki jo bomo aplicirali na napovednem modelu in prišli do napovedi.

Pomembno je, da so vse tri zbirke pripravljene na isti način. Običajno sta prvi dve zbirki derivata iste osnovne zbirke, ki jo z vzorčenjem razdelimo na dva dela, in potem prvi del uporabimo za izdelavo, drugi pa za testiranje kreiranega modela.

Oracle Data Mining podpira štiri algoritme za izvedbo klasifikacije (Oracle, 2008, str. 5/13), in sicer:

- odločitvena drevesa (angl. *Decision trees – DT*),
- posplošeni linearni model (angl. *Generalized Linear Models – GLM*),
- naivni Bayesov algoritem, (angl. *Näive Bayes – NB*),
- metoda podpornih vektorjev (angl. *Support Vector Machines – SVM*).

Vsak od algoritmov ima svoje zahteve in predpostavke. Na tem mestu se bomo osredotočili na to, kaj je potrebno upoštevati pri uporabi algoritma metode podpornih vektorjev.

Algoritem metode podpornih vektorjev deluje izključno na numeričnih atributih. V primeru nominalnih atributov algoritem avtomatično preslika tak atribut v več binarnih, katerih kombinacija predstavlja ustrezno preslikano vrednost nominalnega atributa. Vsak binarni atribut ima vrednost lahko le 0 ali 1 (Oracle, 2008, str. 18/4).

V primeru manjkajočih vrednosti pri atributih z enostavnimi podatkovnimi tipi (negnezdeni stolpci), algoritem metode podpornih vektorjev privzame te kot naključno

manjkajoče. Nominalne vrednosti bodo tako zamenjane z modusom, numerične pa s povprečno vrednostjo. V gnezdenih stolpcih algoritem interpretira te manjkajoče vrednosti kot redke, kar pomeni, da jih bo nadomestil z ničlami v primeru numeričnih atributov, v primeru nominalnih pa bodo manjkajoče vrednosti zamenjane z ničelnim vektorjem (Oracle, 2008, str. 18/4).

Posebnost algoritma metode podpornih vektorjev je, da zahteva vse podatke v normalizirani obliki v intervalu [0,1]. Normalizacijo numeričnih atributov je mogoče izpeljati »ročno« v okviru predhodne priprave podatkov ali pa prepustiti njeno izvedbo algoritmu. Avtomatična priprava podatkov v primeru algoritma metode podpornih vektorjev uporablja Min-Max normalizacijo (Oracle, 2008, str. 18/4).

#### 3.3.3.3 Priprava podatkov pri izdelavi modela za analizo nakupne košarice

Asociacije so funkcija podatkovnega rudarjenja, ki jo uporabljamo za odkrivanje verjetnosti, da se določeni izdelki hkrati pojavijo v isti transakciji. Razmerja med tako pojavljajočimi izdelki so izražena v obliki asociacijskih pravil (Oracle, 2008, str. 8/1).

Na primer, asociacijsko pravilo v primeru knjižnega kluba je, da bo pri nakupu knjige A kupec z verjetnostjo P v okviru istega nakupa kupil tudi knjigo B. Takšno modeliranje imenujemo tudi analiza nakupne košarice, ki se izkaže za zelo koristno v primeru direktnega trženja, prodajnih akcij, odkrivanja prodajnih trendov, oblikovanja prodajnih mest, dizajna prodajnih katalogov in spletnih trgovin (Oracle, 2008, str. 8/1).

Asociacijski modeli so načrtovani, da uporabljajo transakcijske podatke. Vrednosti NULL v transakcijskih podatkih se obravnava kot znane, vendar ne nastopajo v posamezni transakciji. Zaradi tega so transakcijski podatki v svoji osnovi redki, saj na primer v eni sami transakciji nastopa le majhen delež vseh možnih izdelkov (Oracle, 2008, str. 10/5).

V asociacijskih modelih se odsvetuje razvrščanje v grupe na osnovi enakomernega obsega, saj lahko pojav osamelcev, povzroči koncentracijo podatkov v le nekaj gručah, v izjemnih primerih celo v le eni sami gruči, kar bistveno vpliva na učinkovitost algoritma.

## 3.4 Metodološki okvir priprave podatkov v Oracle okolju

Pred opredelitvijo metodološkega okvira priprave podatkov v Oracle okolju, je potrebno poudariti, da pri procesu podatkovnega rudarjenja nikoli ne gre za enosmeren proces, ki se začne z definicijo problema, raziskovanjem in pripravo podatkov, izdelavo modela podatkovnega rudarjenja ter se konča z uporabo tega modela na množici novih podatkov. V osnovi je to cikličen proces, ki se ves čas vrača na preverjanje, ali smo s trenutno rešitvijo rešili poslovni problem, ter na iskanje možnosti, kako trenutno rešitev izboljšati.

Oracle definira proces podatkovnega rudarjenja (glej Prilogo 17) nekoliko drugače kot CRISP-DM, vendar gre v osnovi za vsebinsko podobno opredelitev, ki predvideva štiri faze (Oracle, 2008, str. 1/5):

- opredelitev poslovnega problema,
- zbiranje podatkov in priprava,
- izdelava modela in vrednotenje ter
- uporaba znanja.

Pri pripravi podatkov za podatkovno rudarjenje mora biti v ospredju vedno cilj, ki ga želimo z izvedbo podatkovnega rudarjenja doseči. Razreševanje poslovnega problema narekuje izbiro funkcije in algoritma podatkovnega rudarjenja, ki posledično opredeljuje poslovne in tehnične zahteve priprave podatkov. Ti morajo biti pripravljeni v obliki, ki je bila praktično prikazana, in v kateri vsak zapis v tabeli primerov opisuje eno entiteto (osebo, dogodek, ...). Vsebino zapisov tabele primerov lahko opredelimo do neke mere, toda dokler ni narejena podrobna analiza izvornih podatkov, ne moremo povsem vedeti, ali bomo želeni format podatkov dejansko lahko pripravili.

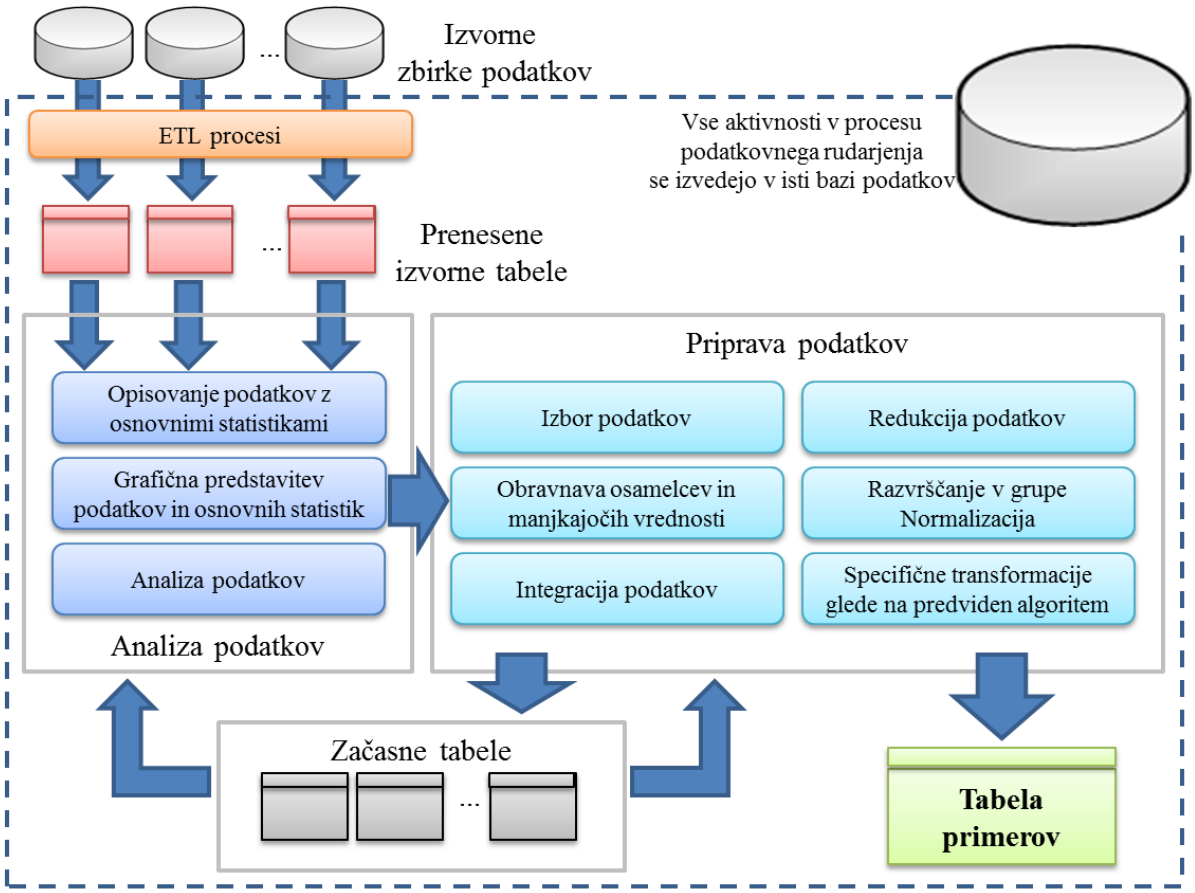
Ne glede na pričakovanja je potrebno identificirati možne vire podatkov. Običajno analize identificiranih virov podatkov ne izvajamo v izvornih sistemih, temveč je smiselno prenesti tabele, v katerih pričakujemo podatke za izvedbo podatkovnega rudarjenja, v eno samo podatkovno shemo, kjer poteka opisovanje in spoznavanje s podatki. V Oraclovem okolju je to podatkovna shema v bazi podatkov, kjer imamo nameščeni dodatni opciji Oracle baze podatkov Data Mining in Oracle R Enterprise. Za prenos iz izvornih baz podatkov v ciljno, uporabimo Oracle Data Integrator, ali kakšno drugo ETL orodje. Omenjeno orodje je primerno predvsem zaradi dejstva, da je po eni strani neodvisno od tehnologije baz podatkov, v katerih se izvorni podatki nahajajo, po drugi strani pa ima zelo širok nabor možnosti za povezovanje z različnimi bazami podatkov. Na ta način se izognemo uporabi večih orodij pri pridobivanju podatkov. Seveda pa so nam na voljo tudi druge možnosti in orodja.

Pri pridobivanju podatkov iz izvornih sistemov je priporočljivo, da se podatke prenese v enotno podatkovno shemo nespremenjene. Vse transformacije izvajamo v fazi priprave podatkov. Če bi izvajali transformacije že v sami fazi pridobivanja, kar ETL orodja seveda omogočajo, bi podatke lahko nevede pokvarili.

Ko so podatki preneseni v enotno podatkovno zbirko, se z vsako tabelo primerov seznanimo in podatke podrobneje raziščemo. Pri tem imamo možnost uporabe različnih orodij, Oracle Data Miner in Oracle R Enterprise, da za vse atribute v teh tabelah pripravimo osnovne opisne statistike oziroma tudi zahtevnejše statistične analize, kot je na primer izračun soodvisnosti kategoričnih atributov s testom $\chi^2$, ko ugotavljamo možnosti

redukcije podatkov. Omenjeni dve orodji omogočata, da se vse analize izvedejo v bazi podatkov, zaradi česar se izognemo izvozu podatkov iz baze podatkov, vse skupaj pa se lahko zaradi zmožnosti baze podatkov izvede hitreje.

*Slika 88: Metodološki okvir priprave podatkov v Oracle okolju*




Priprava podatkov poteka z definiranjem poteka dela (angl. *workflow*) v orodju Oracle Data Miner. Posamezni koraki poteka dela izvajajo posamezne transformacije (na primer nadomeščanje manjkajočih vrednosti, agregacija, obravnava izstopajočih vrednosti, redukcija podatkov) in operacije (na primer združevanje podatkov, filtriranje, vzorčenje), pri čemer uporabnik ne potrebuje nujno znanja iz programiranja, saj vse potrebne ukaze nastavlja s parametri. V procesu je mogoče po vsakem izvedenem koraku kreirati vmesno tabelo, ki jo lahko v vsakem trenutku analiziramo z orodji, ki smo jih spoznali že v fazi spoznavanja s podatki. Končni cilj priprave je tabela primerov, v kateri vsak zapis v tabeli predstavlja natanko en primer. Tako imamo, kot v našem primeru knjižnega kluba, za vsakega člana kluba, aktivnega in neaktivnega, kreiran točno po en zapis, ki vsebuje tako njegove osebne in članske podatke, kot tudi zgodovino njegovih nakupov.

Oracle Data Mining s svojimi orodji načeloma podpira vse faze procesa podatkovnega rudarjenja. Orodja veliko omogočajo in so dobro prilagojena uporabniku. Vgrajene

funkcionalnosti pa lahko po drugi strani zavajajo, saj lahko neizkušen uporabnik dobi vtis, da je mogoče preskočiti nekatere teoretične predpostavke, ki so opisane v predhodnih poglavjih. Funkcionalnost, kot je na primer avtomatična priprava podatkov, lahko zavede uporabnika, da priprava podatkov morda sploh ni potrebna, saj imajo algoritmi vgrajene funkcionalnosti, kot je nadomeščanje manjkajočih vrednosti, obravnava osamelcev ali normalizacija. V resnici to ne drži povsem, kar potrjuje predstavljen primer knjižnega kluba. V teoretičnem delu naloge smo v poglavju Normalizacija vrednosti obravnavali različne načine normalizacije. Videli smo, da algoritem pri napovednem modelu uporabi Min-Max normalizacijo, pri čemer smo tudi prikazali, da je na primer normalizacija na osnovi standardne oziroma z-vrednosti, učinkovitejša.

Podobno je potrebno odgovoriti na vprašanje, ali je uporaba povprečne vrednosti, ki jo največkrat uporablja algoritem avtomatske priprave v primeru nadomeščanja manjkajočih vrednosti, najbolj pravilna. V teoretičnem delu smo v poglavju Manjkajoče in neobstoječe vrednosti namreč diskutirali, ali ni mogoče v tem primeru bolje uporabiti vrednost, ki ohranja standardni odklon in ne povprečne vrednosti atributa.

Tu se ponuja sklep, da so funkcionalnosti uporabljenih orodij namenjene predvsem olajšanju in poenostavljanju posameznih korakov v fazi priprave podatkov, nikakor pa njihovi odstranitvi, ter da je potreben skrben premislek, ko te funkcionalnosti orodja uporabljamo.

Prav tako sem prišel do ugotovitve, da za izvedbo faze spoznavanja s podatki in njihove analize niso dovolj zgolj opisne statistike, ki jih omogoča osnovno orodje Oracle Data Miner. Za boljše spoznavanje s podatki so potrebna dodatna statistična orodja, ki omogočajo podrobnejšo analizo podatkov. Za njeno izvedbo je v Oraclovem okolju najbolj primerna uporaba Oraclove implementacije standardnega statističnega paketa R v Oracle bazi podatkov, Oracle R Enterprise (ORE). Zelo dobra lastnost te implementacije je dejstvo, da se vse statistične operacije izvajajo v bazi podatkov, s čimer je možna obravnava veliko večje količine podatkov kot pri klasični različici statističnega paketa R. Prav tako ni potrebe po izvozu podatkov v datotečni sistem. V kombinaciji z ORE je Oraclova rešitev povsem primerljiva z drugimi platformami podatkovnega rudarjenja, kot sta SAS ali SPSS.

## SKLEP

Pyle (Pyle, 2003b, str. 366) je zapisal, da lahko priprava podatkov zajema od 60 do 90 % vsega časa, potrebnega za izvedbo celotnega procesa podatkovnega rudarjenja. Na podlagi teoretičnih predpostavk, ki so bila osnova tudi za praktičen primer priprave podatkov, lahko potrdim zgornjo navedbo. Dejansko lahko priprava podatkov traja večino vsega časa, ki ga porabimo za izvedbo podatkovnega rudarjenja. Ključno vprašanje, ki se pri tem

postavlja, je, kdaj vemo, da so podatki dovolj dobro pripravljeni. V dostopni literaturi odgovora na to vprašanje nisem našel, zato je odgovor nanj najbrž povezan z izkušnjami analitika, ki izvaja proces podatkovnega rudarjenja, ki je po svoji naravi cikličen in stremi k nenehnemu izboljševanju. Slednje je vgrajeno tudi v vse referenčne modele procesa podatkovnega rudarjenja, ki jih navajam. Nenazadnje so lahko rezultati izvedbe aktivnosti, ki jih narekuje rezultat podatkovnega rudarjenja (na primer izvedba trženjske akcije na osnovi seznama potencialnih kupcev, ki je rezultat napovedi klasifikacijskega modela podatkovnega rudarjenja), avtomatično dodaten vir podatkov za nov cikel podatkovnega rudarjenja.

V uvodu sem si postavil nekaj ključnih vprašanj, na katera sem iskal odgovore v okviru raziskovalne naloge, in sicer:

1. Kako se priprava podatkov umešča v proces podatkovnega rudarjenja in kateri so ključni postopki priprave podatkov v procesih podatkovnega rudarjenja?

Priprava podatkov je ključen korak v vseh standardnih procesnih modelih, ki opredeljujejo podatkovno rudarjenje, kar potrjujeta tako teoretični pregled modelov kot tudi praktični primer. Preden se lotimo same priprave podatkov je potrebno povsem razumeti poslovni problem, ki ga želimo rešiti s podatkovnim rudarjenjem, pridobiti potrebne podatke ter se z njimi podrobno seznaniti. S tem si vzpostavimo ustrezno podlago za izvedbo priprave podatkov, v okviru katere se podatki prečistijo, integrirajo, reducirajo in transformirajo v obliko, primerno za izvedbo algoritmov podatkovnega rudarjenja. Glede na izbiro algoritma nato podatke še dodatno prilagodimo glede na njegove zahteve.

2. Katere tehnike in metode uporabljamo pri pripravi podatkov za doseganje čim boljših rezultatov aplikacij podatkovnega rudarjenja?

V nalogi so opisane nekatere tehnike in metode, ki jih lahko uporabimo pri pripravi podatkov in so tudi sicer dostopne v literaturi. Prepričan sem, da njihov teoretični izbor in pregled nikakor ni končen in da je mogoče uporabiti še druge, predvsem zahtevnejše tehnike in bolj specializirane metode. Hkrati sem prepričan, da opisane tehnike in metode podajajo dovolj dobro osnovo za kasnejšo preverbo v praksi, sploh v kombinaciji z orodji, ki sem jih v praktičnem primeru uporabil.

3. Kakšne omejitve nam pri tem postavljajo programska orodja za podatkovno rudarjenje?

Uporabljena so orodja, ki niso postavljena v okvire raziskovalnih laboratorijev, temveč so v tem trenutku na voljo na trgu in jih ima možnost uporabiti skoraj vsakdo. Pri tem znanje programiranja ni nujno potrebno, seveda pa to ne pomeni, da je reševanje poslovnih problemov s pomočjo podatkovnega rudarjenja enostavno. Na osnovi spoznanj, ki sem jih

pridobil pri pripravi te naloge, ocenjujem, da je potrebno široko poznavanje poslovnih problemov, potrebna so znanja iz statistike in podatkovnega modeliranja, potrebno je dobro poznavanje principov podatkovnega rudarjenja ter predvsem izkušnje, ki jih praktik pridobiva ravno na primerih, ki jih rešuje.

4. Ali je mogoče na osnovi ugotovitev vzpostaviti enoten metodološki okvir priprave podatkov za podatkovno rudarjenje v izbranem informacijskem okolju?

Na osnovi dostopne literature in praktičnega primera sem poskusil opredeliti metodološki okvir priprave podatkov v Oracle okolju. Drugih platform podatkovnega rudarjenja v nalogi nisem obravnaval, tako da težko podam oceno, da je metodološki okvir primeren tudi za ta okolja. Slednje verjetno velja predvsem za tehnični del izvedbe, saj ima vsaka platforma svoje prednosti in slabosti, predvsem v zmožnostih orodij za izvedbo posameznih operacij priprave podatkov.

V nalogi je opisan le del problemov, ki jih rešujemo s podatkovnim rudarjenjem. Že danes - zagotovo pa je tudi v prihodnosti pričakovati porast uporabe tehnik in metod podatkovnega rudarjenja - se odpira cela vrsta poslovnih problemov, ki jih popularno imenujemo »Big Data«, od natančnega napovedovanja volilnih izidov do analize tekstovnih podatkov, ki jih pridobivamo iz socialnih omrežij. Pred nami so ogromne količine podatkov. Če jih želimo učinkovito uporabljati, je izjemnega pomena, da jih odlično poznamo. Šele na osnovi odličnega poznavanja podatkov, jih lahko pripravimo tako, da bomo s pomočjo modelov podatkovnega rudarjenja dobivali kar se da dobre rezultate, na primer za vprašanje, koga moramo posebej naslavljati in s čim, da bo nek kandidat na volitvah dobil največje število glasov, da bomo pravočasno zaznali trende na spletu, ki se tičejo vprašanj varnosti, ali panenazadnje, če hočete, katera dva izdelka moramo postaviti skupaj na polico, da bomo kar se da optimizirali prodajo obeh.

# LITERATURA IN VIRI


1. Ayres, I. (2007). *Super Crunchers*. London: John Murray.
2. Bramer, M. (2007). *Principles of Data Mining* (11–20). London: Springer-Verlag London Limited.
3. Batagelj, V. (b.l.). *Centralni limitni izrek*. Najdeno 12. maja 2012 na spletnem naslovu http://vlado.fmf.uni-lj.si/educa/dmfa/OZ97/CLT/cenlimiz.htm
4. Chakrabarti, S., Cox, E., Frank, E., Güting, R. H., Han, J., Jiang, X., Kamber, M., Lightstone, S. S., Nadeau, T. P., Neapolitan, R. E., Pyle, D., Refaat, M., Schneider, M., Teorey, T. J., & Witten, I. H. (2009). *Data Mining: Know it all* (37–109). Burlington: Morgan Kaufmann Publishers.
5. Dasu, T. & Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning*. Haboken: John Wiley & Sons, Inc.
6. Davenport, T. H. & Harris, J. G. (2007). *Competing on Analytics – The Science of Winning*. Boston: Harvard Business School Press.
7. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996, november). The KDD Process for extracting useful knowledge from volumes of data. *Communications of the ACM. 39*(11), 27–34.
8. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, Fall 1996, 37–54.
9. Giudici, P., & Figini, S. (2009). *Applied Data Mining: Statistical Methods for Business and Industry* (2nd ed.). Chichester: John Wiley & Sons Ltd.
10. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). San Francisco: Morgan Kaufmann Publishers, Inc.
11. Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining*. Cambridge: MIT Press.
12. Hi-kvadrat test. (b.l.) V e-studij.si. Najdeno 28. aprila 2012 na spletnem naslovu http://www.e-studij.si/Test_hi-kvadrat
13. Jermyn, P., Dixon, M., & Read, B.J. (1999, 2.–3. november). 12th ERCIM Workshop on Database Research. *Preparing Clean Views of Data for Data Mining*. Najdeno aprila 2011 na spletnem naslovu http://www.ercim.eu/publication/ws-proceedings/ 12th-EDRG/EDRG 12/ JeDiRe.pdf
14. Jaklič, J. (2010, oktober). *Poslovna inteligenca* [prosojnice]. Najdeno 12. maja 2012 na spletnem naslovu http://miha.ef.uni-lj.si/_dokumenti3plus2/196150/pi-1011.pdf
15. Jurišič, J. (2011, 18. oktober). *Analiza glavnih komponent*. Najdeno 12. maja 2012 na spletnem naslovu http://matematika-racunalnistvo.fnm.uni-mb.si/stat/MA%20Izredni/ 3.%20Analiza% 20glavnih%20komponent.pdf
16. Kantardžić, M. (2003). *Data Mining—Concepts, Models, Methods, and Algorithms*. Hoboken: John Wiley & Sons, Inc.
17. Kimball, R. (1996, september). DBMS Online – Data Warehouse Architect. *Dealing with Dirty Data - The science of maintaining clean data in your warehouse, and why*

*nobody talks about it*. Najdeno novembra 2011 na spletnem naslovu http://www.kimballgroup.com/ html/ articles_search/articles1996/9609d14.html.
18. Kimball, R. (1997a, oktober). DBMS Online – Data Warehouse Architect. *Digging into Data Mining – Your Data Warehouse is Your Data Mining Platform*. Najdeno 26. februarja 2012 na spletnem naslovu http://www.kimballgroup.com/html/articles_ search/articles1997/9710d05.html
19. Kimball, R. (1997b, november). DBMS Online – Data Warehouse Architect. *Preparing For Data Mining*. Najdeno 26. februarja 2012 na spletnem naslovu http://www. kimballgroup.com/html/articles_search/articles1997/9711d05.html
20. Košmelj, K. (2007). Metoda glavnih komponent: osnove in primer. *Acta agriculturae Slovenica, 89(1)*. Avgust 2007, 159–172.
21. Larose, D. T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. Hoboken: John Wiley & Sons.
22. Lewis, M. (2003). *Moneyball – The Art of Winning an Unfair Game*. New York: W.W. Norton & Company Inc.
23. Linoff, G. S., & Berry, M. J. A. (2011). *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management* (3rd ed.). Indianapolis: Wiley Publishing, Inc.
24. Maimon, O. (ur.) & Rokach, L. (ur.) (2010). *Data Mining and Knowledge discovery Handbook* (2nd ed.), 19–132. New York: Springer Science+Business Media.
25. McCue, C. (2007). *Data mining and predictive analysis: intelligence gathering and crime analysis.* Oxford: Butterworth-Heinemann.
26. Nisbet, R., Elder, J., & Miner G. (2009). *Handbook of statistical analysis and data mining applications*. Burlington: Elsevier Inc.
27. Oracle. (2006, april). *Oracle Data Mining 10.2.0.1 Tutorial*. Najdeno novembra 2011 na spletnem naslovu http://download.oracle.com/odm/odminer/odminer_tutorial_ 10201.zip.
28. Oracle. (2008). *Oracle Data Mining Concepts, 11g Release 1 (11.1)*. San Francisco: Oracle Corporation.
29. Oracle. (2009, julij). *Oracle Retail Data Model – Reference 10g Release 2* (str. 6/1–6/46). San Francisco: Oracle Corporation.
30. Oracle. (2010). *Oracle Data Mining Application Developer's Guide, 11g Release 2 (11.2).* Najdeno novembra 2011 na spletnem naslovu http://www.oracle.com/ technetwork/database/options/odm/odmtelcowhitepaper-26595.pdf.
31. Oracle. (2011a, februar). *Oracle Data Mining 11g Release 2: Mining Star Schemas – A Telco Churn Case Study.* An Oracle White Paper. San Francisco: Oracle Corporation.
32. Oracle. (2011b, marec). *Oracle Communications Data Model – Reference 11g Release 2* (str. 10/1–10/14). San Francisco: Oracle Corporation.
33. Oracle. (2011c, 11. marec). *Oracle Data Mining 11g Release 2 OBE Series*. Najdeno novembra 2011 na spletnem naslovu strani: http://apex.oracle.com/pls/apex/f?p=44785: 24:1885469748235176::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5272,29.

34. Parr Rud, O. (2000). *Data Mining Cookbook: Modeling Data for Marketing, Risk, and Customer Relationship Management*. New York: John Wiley & Sons, Inc.
35. Pearson, R. K. (2005). *Mining Imperfect Data: Dealing with Contamination and Incompete Records.* Philadelphia: Society for Industrial and Applied Mathematics.
36. Piatetsky-Shapiro, G., & Parker, G. (2006, 7. junij) *Data Mining Course, Module 12: Data Preparation for Knowledge Discovery*. Najdeno novembra 2011 na spletnem naslovu http://www.kdnuggets.com/data_mining_course/index.html.
37. Požrešna metoda. (b.l.). Najdeno 28. aprila 2012 na spletnem naslovu http://wiki.fmf.uni-lj.si/wiki/Požrešna_metoda.
38. Pyle, D. (1999). *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc.
39. Pyle, D. (2003a). *Business Modeling and Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc.
40. Pyle, D. (2003b). Data Collection, Preparation, Quality, and Visualization. Ye, Nong (ur.). *The handbook of data mining* (str. 365–391). Mahwah: Lawrence Erlbaum Associates, Inc., Publishers.
41. Refaat, M. (2007). *Data preparation for data mining using SAS*. San Francisco: Elsevier Inc.
42. Richeldi, M., & Perrucci, A. (2002, 17.december). *Churn Analysis Case Study*. Najdeno novembra 2012 na spletnem naslovu http://www-ai.cs.uni-dortmund.de/ DOKUMENTE/richeldi_perrucci_20 02b.pdf
43. Ruger, T.W., Kim, P.T., Martin, A.D., & Quinn, K.M. (2002, oktober). The Supreme Court Forecasting Project: Legal and Political Science Approaches to Predicting Supreme Court Decisionmaking. *Columbia Law Review, 104.* 1150–1210. Najdeno decembra 2011 na spletnem naslovu http://wusct.wustl.edu
44. Ruger, T. W., Kim, P.T., Martin, A.D., &Quinn, K.M. (2004, december). Competing Approaches to Predicting Supreme Court Decision Making. *Perspectives on Politics Symposium. 2*(4), 761–767. Najdeno decembra 2011 na spletnem naslovu http://wusct.wustl.edu
45. Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing, 3*(2), 13–22.
46. Teetor, P. (2011). *R Cookbook*. Sebastopol: O'Reiley Media.
47. Thornthwaite, W. (2005, 1. oktober). Informationweek. *Get Started With Data Mining Now - Are you missing valuable data mining opportunities?* Najdeno novembra 2012 na spletnem naslovu http://www.kimballgroup.com/html/articles_search/ articles%202005/ 0510IE.html?articleID=171000647
48. Two Crows Corporation. (2005). *Introduction to Data Mining and Knowledge Discovery, Third Edition* (3rd ed.). Potomac: Two Crows Corporation.
49. Witten, I. H. and Frank, E. (2005). *Data Mining: practical machine learning tools and techniques* (str. 285–344). San Francisco: Morgan Kaufmann Publishers.

50. Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., & Steinberg, D. (2007, 7. december). *Top 10 algorithms in data mining*. Najdeno novembra 2011 na spletnem naslovu http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf
51. Yau, C. (b.l.). *Elementary statistics with R*. Najdeno 28. septembra 2012 na spletnem naslovu http://www.r-tutor.com/elementary-statistics.
52. Zhang, C., Zhang, S., & Yang, Q. (2003). Data Preperation for Data Mining. *Applied Artificial Intelligence, 17*, 375–381.

# PRILOGE

**KAZALO PRILOG**

**Priloga 1: Nekateri primeri podatkovnega rudarjenja**

## 1 Primer podatkovnega rudarjenja v telekomunikacijah

Telekomunikacije so industrijska panoga, v kateri se podatkovno rudarjenje, predvsem na področju prodaje in trženja storitev uporabnikom, zelo učinkovito uporablja. Osnova za to je predvsem izjemno velika količina podatkov, ki jih imajo telekomunikacijska podjetja o uporabnikih svojih storitev, in storitvah, ki jih ti uporabniki uporabljajo (telefonski klici v različna omrežja, poslana in prejeta SMS sporočila, uporaba podatkovnih komunikacij za prenos podatkov, uporaba različnih mobilnih storitev, itn.).

Orodja za poizvedovanje v bazah podatkov in aplikacije poslovnega obveščanja za izvedbo učinkovitih trženjskih in prodajnih akcij zaradi velikih količin podatkov enostavno niso dovolj. Za potrebe napovedovanja, kateri uporabniki storitev bodo v naslednjem kratkoročnem obdobju prenehali uporabljati storitve podjetja in bodo prešli k drugim ponudnikom telekomunikacijskih storitev, se uporablja rešitve, razvite s pomočjo tehnik napredne statistične analize in metod podatkovnega rudarjenja, ki omogočajo prav takšno napovedovanje.

Telekomunikacijska podjetja razvijajo modele podatkovega rudarjenja za podporo trženja in prodaje svojih storitev za različne namene (Oracle, 2011b, str. 10/1), kot so na primer:

- napovedovanje prehoda h konkurenci,
- profiliranje uporabnikov storitev,
- identifikacija faktorjev, ki vplivajo na problem prestopa h konkurenci ali višino porabe posameznega uporabnika,
- identifikacija priložnosti navzkrižne prodaje,
- napovedovanje vrednosti življenske dobe uporabnikov (angl. *life time value*) storitev.

S pomočjo modela, ki ga zgradimo za napovedovanje prehoda h konkurenci, poskušamo identificirati značilnosti uporabnikov storitev, za katere se predvideva, da bodo prenehali uporabljati storitve podjetja in prešli h konkurenci. Rezultat podatkovnega rudarjenja je napoved verjetnosti, da bo uporabnik storitev prešel h konkurenci v naslednjih tednih ali mesecih. Model se gradi na osnovi podatkov o uporabniku storitev, kot so demografski podatki, podatki o uporabljenih storitvah, kakovosti teh storitev in drugi podatki, ki jih imamo o uporabnikih storitev. S pomočjo znanja, ki ga ponudnikih komunikacijskih storitev na ta način pridobijo, lahko pripravijo posebne trženjske programe, da dobičkonosne stranke zadržijo kot uporabnike svojih storitev.

Za izdelavo modelov za napovedovanje prehoda h konkurenci se uporablja klasifikacijske algoritme, kot sta na primer odločitvena drevesa (angl. *decision trees*) in metoda podpornih vektorjev (angl. *support vector machines*) (Oracle, 2011b, str. 10/5–10/10).

Drugačen poslovni problem naslavlja profiliranje uporabnikov storitev, pri katerem gre za razvrščanje uporabnikov storitev v segmente ali gruče na osnovi ključnih demografskih podatkov, vzorcev uporabe storitev in podatkov o storitvah, ki so jih uporabniki uporabljali v preteklosti. Na ta način identificiramo najbolj značilne lastnosti vsake posamezne skupine uporabnikov, ki jim glede na te značilnosti tržniki oblikujejo posebne ponudbe. Na primer, mobilni operater bo za določen segment, na primer »mladi«, ugotovil značilnost, da uporabniki storitev v tem segmentu uporabljajo bistveno več SMS sporočil, kot uporabniki v drugih segmentih. Za takšen segment uporabnikov bodo zato verjetno pripravili storitveni paket, ki bo atraktiven predvsem iz vidika kratkih sporočil.

Pri izdelavi modela podatkovnega rudarjenja, ki omogoča profiliranje uporabnikov storitev, se uporablja algoritme razvrščanja v segmente, kot je na primer razvrščanje z voditelji (angl. *k-means clustering*) (Oracle, 2011b, str. 10/10).

## 2 Primer podatkovnega rudarjenja v profesionalnem športu

Profesionalni šport je zagotovo eno od najbolj zanimivih področij uporabe analitičnih in statističnih metod ter metod podatkovnega rudarjenja. Četudi so si posamezne športne panoge medseboj različne, so si podobne glede količine podatkov, s katerimi razpolagajo, in večinoma tudi glede tega, da v njih tekmujejo nadarjeni, a dragi športniki. Tako kot podjetja, tudi športna moštva poskušajo optimizirati svoje vire in prevsem želijo v svojem delovanju uspeti, to je zmagati (Davenport, 2007, str. 17).

Oakland Athletics, profesionalno baseball moštvo iz lige American League World Series (Lewis, 2002, str. 4–6), je bilo eno prvih, ki je opustilo dotedanji način rekrutiranja mladih igralcev iz srednješolskih lig in lig nižjega ranga. Slednje je do takrat potekalo izključno preko mreže skavtov, ki so s svojo »strokovnostjo« izbirali in predlagali nove igralce za moštvo. Glavni manager moštva je leta 2002 najel analitike, ki so s pomočjo statističnih modelov predlagali igralce, ki jih skavti niti slučajno niso imeli na svojih seznamih. S pomočjo napovednih modelov, zgrajenih na statističnih podatkih igralcev v ligi, so namreč uspeli iz velikih količine podatkov o igralcih izluščiti vzorce in napovedi, na osnovi katerih so za posamezna igralna mesta predlagali presenetljive »idealne« igralce. Takšni predlogi so med skavti povzročili šok in ogorčenje, vendar so se kasneje izkazali kot tekmovalno uspešni in predvsem stroškovno zelo učinkoviti. Oakland Athletics je namreč ekipa z letnim proračunom, ki je med nižjimi v državni ligi in ki je kar nekajkrat manjši od proračunov najbogatejših klubov. Na osnovi napovednega modela so bili predlagani igralci

seveda bistveno cenejši od favoritov letnega izbora. Oakland Athletics so se v naslednjih dveh letih kot zmagovalci zahodne divizije uvrstili v finale v American League World Series (Major League Baseball je razdeljen na dve ligi American League in National League, pri čemer državnega prvaka odloči finale med zmagovalcema obeh lig), v naslednjih dveh letih pa so dvakrat zaporedoma v svoji diviziji osvojili drugo mesto.

Oakland Athletics pa ni bilo edino moštvo v ameriških baseball ligah, ki je za potrebe rekrutiranja igralcev uporabilo napovedne modele na osnovi statističnih podatkov igralcev (Davenport, 2007, str. 18). Baseball moštvo Boston Red Socks je leta 2004, po 68 letih, osvojilo prvenstvo Major League Baseball, leto pred tem pa so igrali v finalu American League World Series. Red Socks so pri implementaciji analitičnih modelov šli še korak dlje od moštva Oakland Athletics. Ne le, da so uporabili napovedne modele pri rekrutaciji igralcev, temveč so uporabili analitiko tudi za pomoč odločitvam pri vodenju igre v času tekme. Davenport (2007, str. 18) opisuje primer v finalu American League World Series leta 2003, ko so Red Socks igrali proti ekipi New York Yankees. V zadnji, peti tekmi je v sedmem delu igre (baseball ima 9 delov igre) metal igralec, za katerega so analitiki moštva napovedali, da bo po sedmem delu igre popustil in ne bo več enako učinkovit, kot v prvih sedmih delih tekme, zaradi česar je obstajala velika verjetnost, da bo izgubil proti odbijalcu, ki mu je takrat stal nasproti. Vodja ekipe se je vseeno odločil, da napovedi analitikov ne bo upošteval in metalec je nadaljeval z igro. Red Socks so, tako kot so napovedali analitiki, izgubili.

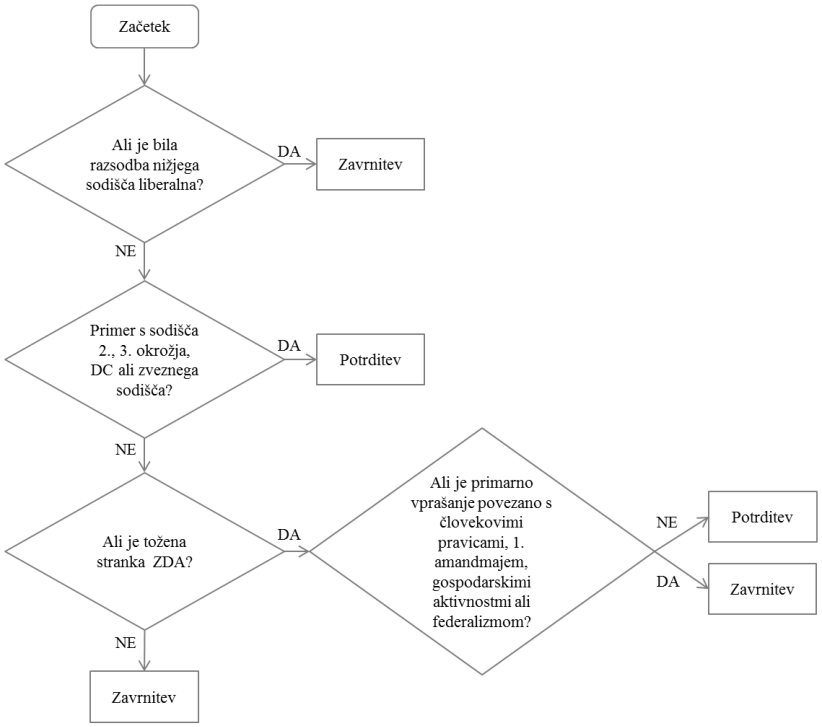
Nogometni klub AC Milan uporablja metode podatkovnega rudarjenja za izdelavo modelov za napovedovanje poškodb igralcev in s tem njihovo preprečevanje. Napovedne modele gradijo na osnovi fizioloških, ortopedskih in psiholoških podatkov, ki jih imajo zbrane iz različnih virov. Na tej osnovi poskušajo ugotoviti dejavnike tveganja poškodbe posameznega igralca v moštvu in se pravočasno preventivno odzvati na takšne napovedi (Davenport, 2007, str. 20).

## 3 Primer podatkovnega rudarjenja v pravu

Ayres (2007, str.104–108) opisuje primer napovedovanja razsodb Vrhovnega sodišča ZDA. V poskusu (Ruger et al, 2004; Ruger et al, 2002), ki so ga opravili v ZDA leta 2002, je bila postavljena teza, da lahko na osnovi poznavanja preteklih razsodb Vrhovnega sodišča ZDA s pomočjo modela podatkovnega rudarjenja natančneje napovedo razsodbo kot skupina 83 vrhunskih pravnih strokovnjakov, med katerimi so bili pravniki, ki so v preteklosti že delovali v okviru Vrhovnega sodišča ZDA, profesorji prava in celo dekani najuglednejših ameriških pravnih fakultet.

Vrhovno sodišče ZDA sestavlja 9 vrhovnih sodnikov, za vsakega od katerih je bilo izdelano odločitveno drevo na osnovi podatkov njihovih razsodb v 628 primerih v preteklosti. Za testiranje klasifikacijskega modela so uporabili 68 primerov, ki jih je vrhovno sodišče ZDA obravnavalo oktobra 2002.

*Slika 1: Model klasifikacijskega odločitvenega drevesa za vrhovno sodnico O'Connor*




*Vir: T. W. Ruger et al., Competing Approaches to Predicting Supreme Court Decision Making, 2004, str. 762.*

Na osnovi posameznih odločitvenih dreves (glej Sliko 1) so ugotovili, da je za napovedni model dovolj upoštevati le 6 podatkov:

- izvorno okrožno sodišče,
- področje pravnega primera,
- tip vlagatelja tožbe (na primer vlada ZDA, delodajalci),
- tip tožene stranke,
- ideološka usmeritev sodišča, ki je razsojalo na nižji stopnji (liberalno/konzervativno),
- ali je vlagatelj trdil, da je razsodba sodišča na nižji stopnji neustavna.

Ugotovitve so bile za avtorje raziskave (Ruger et al., 2004, str. 765) presenetljive. Njihov model se je namreč izkazal za natančnejšega v napovedi skupne razsodbe vrhovnega sodišča (75 % pravilnih napovedi) od napovedi pravnih strokovnjakov (59 %). Slednji so

sicer za malenkost bolje napovedali razsodbe posameznih sodnikov (67,9 % pravilnih napovedi razsodb strokovnjakov v primerjavi s 66,7 % pravilnih napovedi modela), pri čemer je bil model natančnejši pri ključnih treh vrhovnih sodnikih.

Za praktično uporabo opisani primer žal ne temelji na dovolj veliki količini podatkov in primerov. Prav tako niso bile predstavljene morebitne ponovitve raziskave, da bi lahko z večjo gotovostjo trdili, da ugotovitve dejansko držijo. V vsakem primeru pa je predstavljen zanimiv primer uporabe podatkovnega rudarjenja.

**Priloga 2: Opisne statistike**

## 1 Mere središčnosti

V vsakem vzorcu podatkov najprej pridobimo podatke o skupnem številu elementov vzorca, vsoti vrednosti, poiščemo največjo in najmanjšo vrednost, preštejemo pojavitve posameznih vrednosti in drugo. Iz teh osnovnih mer lahko izračunamo druge mere, ki že podrobneje opisujejo vzorec podatkov.

### 1.1 Aritmetična sredina

Ena najosnovnejših mer merjenja središčnih tendenc v podatkih je aritmetična sredina oziroma povprečna vrednost. Za vzorec $N$ podatkov $x_1$, $x_2$, ... $x_N$ izračunamo povprečno vrednost

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + \cdots + x_N}{N}.$$

V relacijskih bazah podatkov temu ustreza vgrajena agregatna funkcija $AVG()$, mogoče pa jo je izračunati s pomočjo funkcij $SUM()/COUNT()$. Pri uporabi statističnih aplikacij (na primer *R*) se povprečna vrednost izračuna s pomočjo funkcije $mean()$.

V nekaterih primerih so vrednosti $x_i$ povezane z utežmi $w_i$. Vrednost $w_i$ podaja informacijo o pomenu ali frekvenci pojavljanja posamezne vrednosti v vzorcu. V tem primeru je tehtana povprečna vrednost

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_N x_N}{w_1 + w_2 + \cdots + w_N}.$$

### 1.2 Mediana

Čeprav je aritmetična sredina ena od najbolj uporabnih mer središčnosti, ima ključno pomanjkljivost, in sicer, da je zelo občutljiva na pojav ekstremnih vrednosti (na primer osamelcev) ali pojav nesimetričnosti v vzorcih.

V primeru ekstremnih vrednosti lahko takšne vrednosti v izračunu omejene povprečne vrednosti izpustimo in tako izločimo njihov vpliv na druge podatke v vzorcu. Seveda v takšnem izračunu ne smemo izpustiti prevelikega dela vzorca (na primer 20 %). Odstotek vzorca, ki ga izločimo zaradi svoje ekstremne vrednosti je manjši (na primer 1–2 %).

Podoben problem imamo s povprečno vrednostjo v primeru asimetrične porazdelitve podatkov.

Za takšne porazdelitve je primernejša mera srednja vrednost ali mediana, saj asimetrije in osamelci manj vplivajo na njeno vrednost. Mediano izračunamo na naslednji način:

$$\tilde{x} = \begin{cases} x_{\frac{N+1}{2}} & N\ je\ liho\ število \\ \frac{1}{2}\left(x_{\frac{N}{2}} + x_{\frac{N}{2}+1}\right) & N\ je\ sodo\ število \end{cases}$$

Mediana je holistična mera, kar pomeni, da jo lahko izračunamo le na celotnem vzorcu podatkov. Slednje lahko v zelo velikih vzorcih pomeni precejšen problem, saj zahteva veliko več računalniških zmogljivosti.

Mediano zelo velikega podatkovnega vzorca pa lahko vseeno ocenimo. Predpostavimo, da je celoten vzorec podatkov razdeljen v razrede, ki so omejeni z enakomernimi intervali, in da so vrednosti $x_i$ razporejene v posamezne razrede glede na njihove vrednosti. Predpostavimo, da je število posameznih vrednosti v razredu, tj. frekvenca intervala, znana. Pri izračunu mediane potrebujemo medialni interval, tj. interval, v katerem se nahaja mediana frekvenčne porazdelitve. Na tej osnovi izračunamo mediano:

$$\tilde{x} = Y_{min} + d\ \frac{\frac{N}{2} - \sum f_i}{f_m},$$

pri čemer so $Y_{min}$ spodnja meja medialnega intervala, $d$ širina medialnega intervala, $\sum f_i$ kumulativna frekvenca do medialnega intervala in $f_m$ frekvenca medialnega intervala.

## 1.3 Modus

Modus je vrednost podatka, ki se v celotnem vzorcu podatkov najpogosteje pojavlja. Ker gre za vrednost pojavitve, se modus lahko uporabi tako v kvantitativnih kot v kvalitativnih vzorcih.

Če ima nek vzorec en modus, potem pravimo, da je porazdelitev unimodalna, v primeru dveh vrednosti, ki se pojavljata najpogosteje, pravimo, da je takšna porazdelitev bimodalna in tako naprej. V splošnem pa pravimo, da je porazdelitev multimodalna, če je vrednosti z enako frekvenco pojavitve v vzorcu več.

Podobno kot pri mediani lahko modus izračunamo tudi v primeru frekvenčne porazdelitve. V tem primeru priprada modus frekvenčnemu intervalu z največjo frekvenco – modalnemu intervalu:

$$M = Y_{min} + d\ \frac{f_M - f_{-1}}{2f_M - f_{-1} - f_{+1}},$$

pri čemer je $Y_{min}$ spodnja meja modalnega intervala, $d$ širina modalnega intervala, $f_M$ frekvenca modalnega intervala, $f_{-1}\ in\ f_{+1}$ frekvenca predhodnega in naslednjega modalnega intervala.

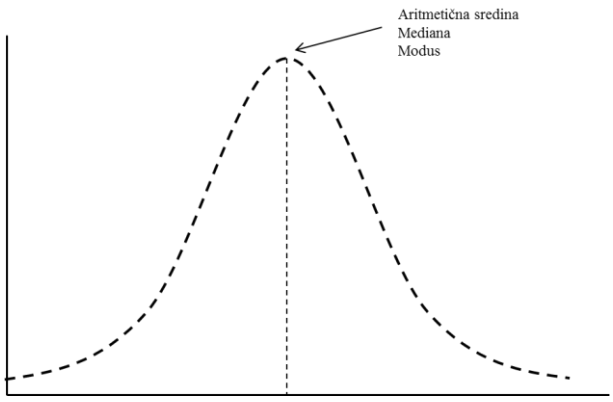
## 1.4 Odnos med aritmetično sredino, mediano in modusom

Za unimodalno numerično porazdelitev, ki je zmerno asimetrična, obstaja naslednji empirični odnos med aritmetično sredino, mediano in modusom:

$$\bar{x} - M\ \approx 3\ \times (\bar{x} - \tilde{x}).$$

V primeru popolnoma simetrične porazdelitve podatkov imajo aritmetična sredina, mediana in modus isto vrednost (glej Sliko 2).

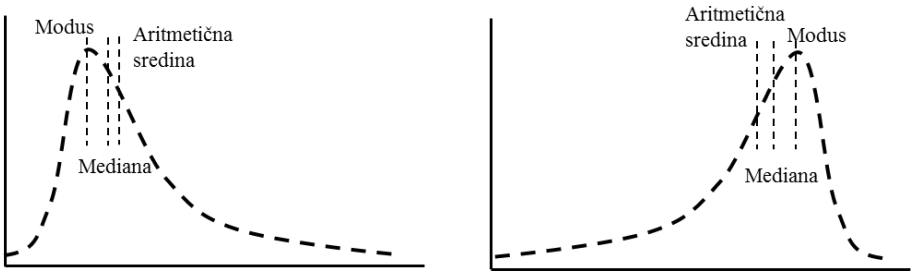
*Slika 2: Simetrična porazdelitev*




*Vir: S.Chakrabarti et al., Data Mining: Know it all, 2009, str. 64*

V realnem svetu je popolnoma simetrična porazdelitev izjemno redka. Namesto tega je bodisi pozitivno asimetrična, bodisi negativno asimetrična (glej Sliko 3).

*Slika 3: Pozitivno asimetrična in negativno asimetrična porazdelitev*




*Vir: S.Chakrabarti et al., Data Mining: Know it all, 2009, str. 64*

# 2 Mere razpršenosti in variabilnosti

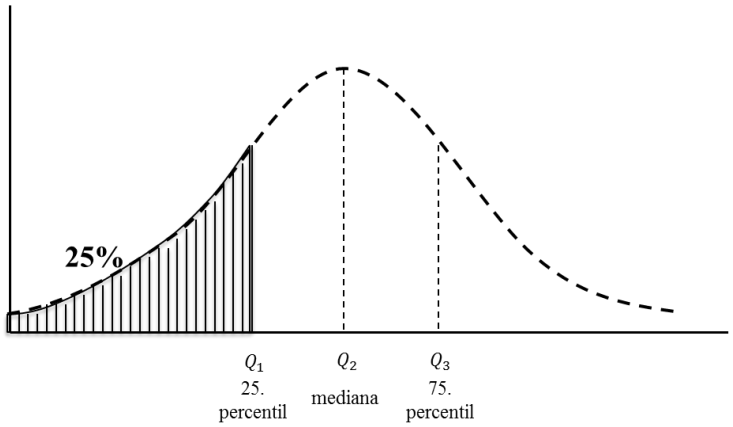
## 2.1 Razpon, kvartili, interkvartilni razmik

Za merjenje razpršenosti uporabljamo razpon, kvantile, kvartile, percentile in interkvantilni razmik. Navedene mere prikažemo na grafu kvantilov (angl. *box-and-whiskerplot, boxplot*), ki je zelo primeren tudi za identifikacijo osamelcev.

Razpon je razlika med najvišjo in najnižjo vrednostjo opazovanega vzorca podatkov.

*n*-ti percentil po velikosti urejenega vzorca podatkov je vrednost $x_i$, pri kateri velja, da je *n* odstotkov vseh vrednosti vzorca manjših ali enakih $x_i$. Mediana je tako na primer 50. percentil. Poleg mediane se pogosto uporabljajo kvartili. Prvi kvartil predstavlja vrednost 25. percentila, tretji kvartil pa vrednost 75. percentila. Kvartili podajajo informacijo o središču, razpršenosti in obliki porazdelitve (glej Sliko 4).

*Slika 4: Razdelitev porazdelitve na kvartile*




*Vir: J. Han et al., Data Mining: Concepts and Techniques (3rd ed.), 2011, str. 76*

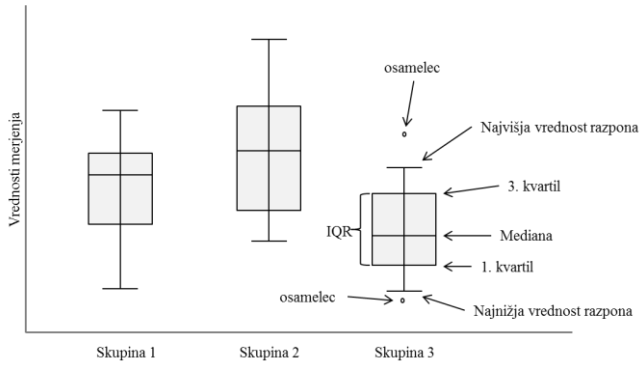
Interkvartilni razmak je razdalja, ki meri razpršenost v osrednji polovici podatkov:

$$IQR = Q_3 - Q_1.$$

Če opazujemo katerokoli od navedenih vrednosti samostojno, ne bomo imeli podrobne informacije o simetričnosti/nesimetričnosti porazdelitve, zato je smiselno opazovati vse vrednosti, tj. prvi kvartil ($Q_1$), tretji kvartil ($Q_3$), mediano ($\tilde{x}$), razpon ($x_{max} - x_{min}$) in interkvartilni razmik ($IQR$), skupaj.

Navedene vrednosti prikažemo na grafu kvantilov (glej Sliko 5):

*Slika 5: Prikaz petih mer razpršenosti na grafu kvantilov*




Za osamelec velja, da je od prvega kvantila oddaljen za 1,5 večkratnik interkvartilnega razmika.

## 2.2 Varianca in standardni odklon

Varianca in standardni odklon sta prav tako meri statistične razpršenosti, ki pa prikazujeta, kako so vrednosti vzorca podatkov razporejene okoli linije pričakovanih vrednosti. Nizka vrednost standardnega odklona pomeni, da so vrednosti vzorca bliže aritmetični sredini, kar pomeni manjšo razpršenost. V primeru višje vrednosti standardnega odklona je porazdelitev vrednosti bistveno bolj razpršena.

Varianco izračunamo s pomočjo formule:

$$V = \sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N} = \left(\frac{1}{N}\sum_{i=1}^{N} {x_i}^2\right) - \bar{x}^2.$$

Standardni odklon je enak kvadratnemu korenu variance:

$$\sigma = \sqrt{V} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}} = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N} {x_i}^2\right) - \bar{x}^2}.$$

Osnovni značilnosti standardnega odklona kot mere razpršenosti sta:

- $\sigma$ meri razpršenost okoli aritmetične sredine in se jo uporablja, če je aritmetična sredina izbrana kot središčna mera,
- $\sigma = 0$, če ni nobene razpršenosti, torej v primeru, ko imajo vse vrednosti vzorca isto vrednost.

Varianca in standardni odklon sta algebraični vrednosti, ker jih je mogoče izračunati na osnovi osnovnih distribucijskih mer, to je mer, kot so $N$ (število vseh elementov vzorca), $\sum x_i$ (vsota vseh vrednosti vzorca) in $\sum x_i^2$ (vsota kvadratov vrednosti vzorca). Vse te vrednosti je mogoče izračunati na katerikoli podmnožici vzorca podatkov.

## 2.3 Mere heterogenosti

Standardnega odklona in variance ni možno izračunati za nenumerične vrednosti, zato je potrebno vpeljati mero, ki bo merila razpršenost tudi za tak tip vrednosti. Za vsako od posameznih vrednosti lahko ugotovimo pogostost njenega pojavaljanja, kar zapišemo v tabeli frekvečne porazdelitve atributa, prikazano z absolutnimi in relativnimi vrednostmi (Guidici & Figini, 2009, str. 16):

| Nivo vrednost | Absolutna Frekvenca | Relativna frekvenca |
|---|---|---|
| $x_1^*$ | $n_1$ | $p_1$ |
| $x_2^*$ | $n_2$ | $p_2$ |
| ... | ... | ... |
| $x_k^*$ | $n_k$ | $p_k$ |

V praksi obstajata dve skrajni vrednosti:

- Ničelna heterogenost, ko imamo v atributu vse vrednosti na enem nivoju X. V tem primeru je $p_i$ = 1 za ta i-ti nivo vrednosti, za vse ostale vrednosti pa je enak 0.

- Maksimalna heterogenost, ko imamo opravka z enakomerno porazdelitvijo je in velja $p_i = \frac{1}{k}$, za vse $i = 1,2,\dots k$.

Koeficient, ki meri heterogenost, bo zaradi tega imel minimum v primeru ničelne heterogenosti in maksimu v primeru maksimalne heterogenosti.

Ginijev koeficient heterogenosti (angl. *Gini index of heterogeniety*) izračunamo:

$$G = 1 - \sum_{i=1}^{k} p_i^2.$$

Ginijev koeficient heterogenosti je enak 0, v primeru popolne homogenosti, v primeru maksimalne heteregonesti pa je enak $1 - 1/k$. Ginijev koeficient heterogenosti lahko normaliziramo, da dobimo relativni koeficient heterogenosti, ki ima vrednosti med 0 in 1:

$$G' = \frac{G}{(k-1)/k}.$$

Drugi koeficient, s katerim lahko prav tako izmerimo heterogenost, je entropija, ki jo izračunamo:

$$E = -\sum_{i=1}^{k} p_i \log p_i,$$

oziroma normalizirani relativni koeficient heterogenosti:

$$E' = \frac{E}{\log k}.$$

## 2.4 Mere koncentracije

Koncentracija je zelo povezana s heterogenostjo. V bistvu je frekvenčna porazdelitev maksimalno koncentrirana, ko ima ničelno heterogenost in obratno. Za razliko od skrajnih vrednosti pa je zanimivo opazovati vmesne vrednosti.

Recimo, da imamo vzorec s prihodki *N* posameznikov, ki ga lahko uredimo po velikosti $0 \le x_1 \le x_2 \le \cdots \le x_N$. Naj bo $N\bar{x} = \sum x_i$ skupna vrednost prihodkov. Pri tem imamo opraviti z dvema skrajnima vrednostima:

- $x_1 = x_2 = \cdots = x_N = \bar{x}$, kar ustreza minimalni stopnji koncentracije (enak prihodek preko *N* enot);
- $x_1 = x_2 = \cdots = x_{N-1} = 0, x_N = N\bar{x}$, kar ustreza maksimalni stopnji koncentracije (le ena enota ima celoten prihodek).

Za opredelitev koeficienta koncentracije najprej vpeljemo dve spremenljivki:

$$F_i = \frac{i}{N} \;\; ; \;\; i = 1,2, \dots N,$$

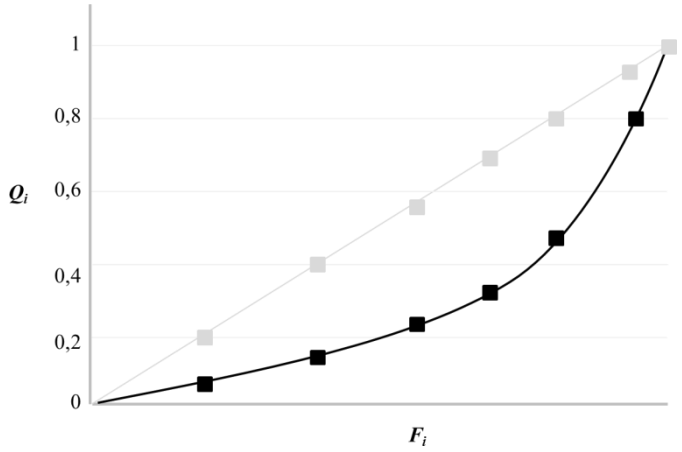
$$Q_i = \frac{x_1 + x_2 + \cdots + x_i}{N\bar{x}} = \frac{\sum_{j=0}^{i} x_j}{N\bar{x}} \quad ; \quad i = 1,2, \dots N.$$

Velja tudi naslednje:

$$0 \leq F_i \leq 1; \; 0 \leq Q_i \leq 1;$$
$$Q_i \leq F_i;$$
$$F_N = Q_N = 1.$$

Če privzamemo, da je $F_0 = Q_0 = 0$, in izračunamo vse preostale pare (0,0), $(F_1, Q_1)$, $(F_2, Q_2)$, ..., $(F_{N-1}, Q_{N-1})$, (1,1), lahko narišemo graf (glej Sliko 6), ki prikazuje koncentracijsko krivuljo:

*Slika 6: Koncentracijska krivulja*




Vir: P. Guidici & S. Figini, Applied Data Mining: Statistical Methods for Business and Industry (2nd ed.), 2009, *str. 19.*

Ginijev koeficient koncetracije je osnovan na razliki $(F_i - Q_i)$, pri čemer velja:

- pri minimalni koncentraciji je $(F_i - Q_i) = 0, i = 1,2, \dots N,$

- pri maksimalni koncentraciji je $(F_i - Q_i) = F_i, i = 1,2, \dots, N-1$, in $(F_N - Q_N) = 0$,
- v splošnem je $0 < (F_i - Q_i) < F_i, i = 1,2, \dots N-1$, pri čemer se razlika povečuje s približevanjem maksimumu koncentracije.

Koeficient koncentracije je tako razmerje med vrednostmi $\sum_{i=1}^{N-1}(F_i - Q_i)$ in njihovim maksimumom $\sum_{i=1}^{N-1} F_i$:

$$R = \frac{\sum_{i=1}^{N-1}(F_i - Q_i)}{\sum_{i=1}^{N-1} F_i},$$

pri čemer R pomeni vrednost 0 najmanjšo in 1 največjo koncentracijo.

## 2.5 Mere asimetrije

Asimetričnost (angl. *skewness*) lahko prikažemo na grafu kvantilov, kjer velja, da je mediana enako oddaljeno od prvega ($Q_1$) in tretjega kvantila ($Q_3$). V primeru zamaknjenosti v desno je razdalja med mediano in $Q_3$ večja kot razdalja med mediano in $Q_1$.

Koeficient asimetričnosti porazdelitve se izračuna kot funkcija tretjega centralnega momenta porazdelitve in standarnega odklona, in sicer:

$$\gamma = \frac{\mu_3}{\sigma^3},$$

pri čemer je $\mu_3$ tretji centralni moment in je enak $\mu_3 = \frac{\sum(x_i - \bar{x})^3}{N}$.

Če velja $\gamma = 0$, potem je porazdelitev simetrična. Če velja $\gamma < 0$, potem je porazdelitev negativno simetrična, v primeru $\gamma > 0$ pa je porazdelitev pozitivno simetrična (Guidici & Figini, 2009, str. 20).

## 2.6 Mere sploščenosti

Koeficient sploščenosti (angl. *index of kurtosis*) je mera, ki meri oddaljenost opazovane porazdelitve od normalne porazdelitve:

$$\beta = \frac{\mu_4}{{\mu_2}^2},$$

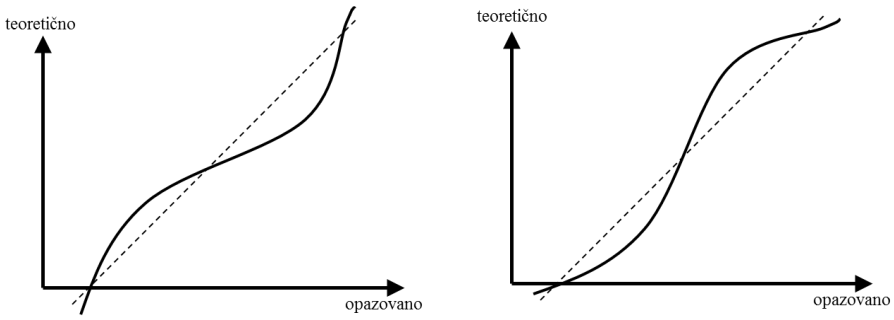
pri čemer sta

$$\mu_4 = \frac{\sum(x_i - \bar{x})^4}{N} \text{ in } \mu_2 = \frac{\sum(x_i - \bar{x})^2}{N}.$$

Porazdelitev je normalna, če je vredost koeficienta sploščenosti enaka 3, $\beta = 3$. V primeru, da velja $\beta > 3$, je porazdelitev koničasta ali hiponormalna, če pa je $\beta < 3$, je porazdeltitev sploščena ali hipernormalna (Guidici & Figini, 2009, str. 21).

Asimetrije in sploščenosti lahko ponazarjamo tudi z grafičnimi orodji. Eno takšnih je Q-Q graf. Pri tem grafu primerjamo kvantile opazovanega atributa s teoretičnimi v normalni porazdelitvi. Če se vrednosti kvantilov opazovanega atributa tesno približajo vrednostim, ki jih opredeljuje premica pod kotom 45˚, je porazdelitev normalna. Sicer imamo lahko enega od naslednjih primerov (Guidici & Figini, 2009, str. 21), ki sta prikazana na Sliki 7:

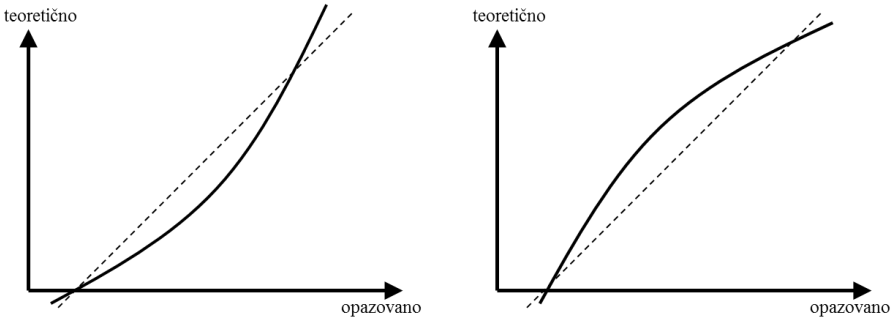
*Slika 7: Primer hiponormalne in hipernormalne porazdelitve*




*Vir: P. Guidici & S. Figini, Applied Data Mining: Statistical Methods for Business and Industry (2nd ed.), 2009, str. 22.*

Podobno kot sploščenost lahko prikažemo tudi simetričnost v primerjavi z normalno porazdelitvijo:

*Slika 8: Primer pozitivno in negativne asimetrične porazdelitve na Q-Q grafu*




*Vir: P. Giudici in S. Figini, Applied Data Mining: Statistical Methods for Business and Industry (2nd ed.)., 2009, str. 22.*

# 3 Grafična predstavitev osnovnih statistik podatkov

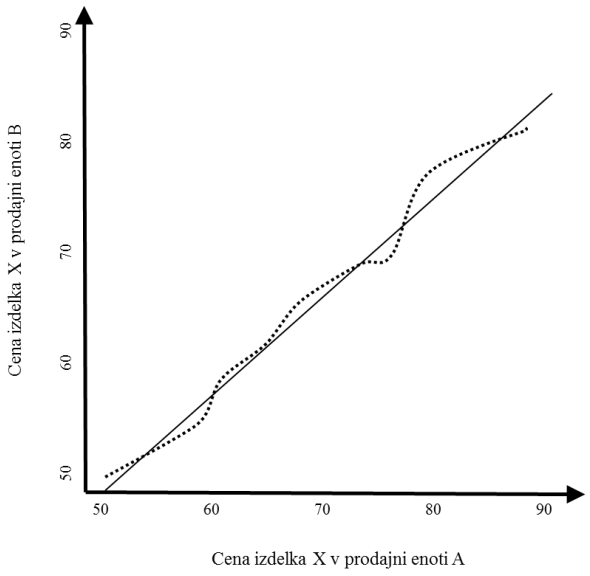
Osnovne statistike podatkov lahko prikazujemo tudi s pomočjo grafov. Najbolj razširjeni so kvantilni graf, Q-Q graf, histogram in korelacijski grafikon.

## 3.1 Kvantilni graf, Q-Q graf

Kvantilni graf prikazuje vse vrednosti izbranega atributa, s čimer podaja takojšen vpogled v njegovo porazdelitev in morebitne odstopanja. Poleg podatkov o vrednostih atributa, so na istem grafu prikazani podatki o kvantilih (Han et al, 2011, str. 79).

Posebna oblika kvantilnega grafa je Q-Q graf (glej primer na Sliki 9), ki prikazuje kvantile porazdelitve enega atributa v primerjavi z ustreznimi kvantili porazdelitve drugega atributa.

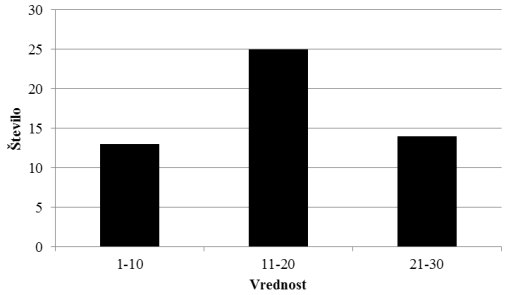
*Slika 9: Primer Q-Q grafa*




## 3.2 Histogram ali stolpični diagram

Histogram (glej Slika 10) se uporablja za prikaz porazdelitve vrednosti atributa. Če je atribut nominalen, stolpec na diagramu predstavlja vrednost atributa, njegova višina pa predstavlja število pojavitev tega atributa v vzorcu (Han et al, 2011, str. 81).

*Slika 10: Primer stolpičnega diagrama*

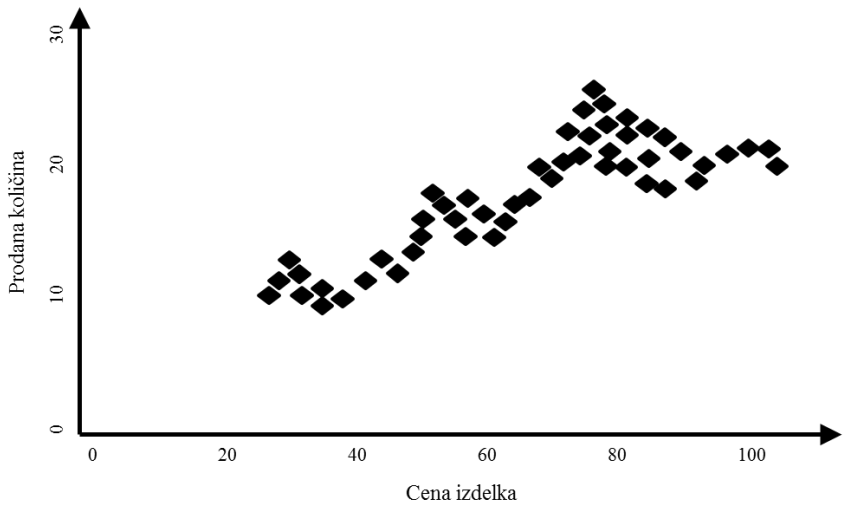



V primeru numeričnega atributa se celoten interval vseh vrednosti atributa razdeli na ločene intervale izbranega razpona. Običajno so ti enakomerni, razpon pa se lahko določi tudi glede na nek drug kriterij, kot je na primer število primerov v izbranem razponu. Ne glede na izbrane razpone število pojavitev vrednosti vsakega posameznega intervala določa višino stoplca (Han et al, 2011, str. 81).

## 3.3 Korelacijski grafikon

Korelacijski grafikon (angl. *scatter plot*) je eden najučinkovitejših načinov ugotavljanja obstoja povezav, odkrivanja vzorcev in trenda med dvema numeričnima atributoma. Na grafu predstavlja vsaka točka par koordinat, ki ju tvorita vrednosti obeh atributov. S korelacijskim grafikonom zelo enostavno identificiramo pojav gruč, izstopajočih vrednosti ali morebitne korelacije med atributoma (Han et al, 2011, str. 82), kar prikazuje Slika 11:

*Slika 11: Primer korelacijskega grafikona*



**Priloga 3: Seznam izvornih tabel primera**

*Tabela 1: Seznam izvornih tabel*[*]

| Ime tabele | Vsebina |
|---|---|
| SRC_CLANI | Tabela članov knjižnega kluba. |
| SRC_CLANI_EMAIL | Tabela članov knjižnega kluba, ki imajo email naslov. |
| SRC_CLANI_FO_EMAIL | Tabela članov knjižnega kluba, fizičnih oseb, ki imajo email naslov (tabela članov knjižnega kluba se razlikuje od tabele fizičnih oseb – vsebuje namreč podatke, ki so specifični za člane kluba). |
| SRC_GRUPE_ARTIKLOV | Tabela grup artiklov vsebuje šifrant, ki predstavlja enega od treh načinov razvrščanja izdelkov. |
| SRC_KLASIFIKACIJA1 | Klasifikacija izdelkov je eden od treh načinov razvrščanja izdelkov, pri čemer je to edini hierarhični način razvrščanja izdelkov.<br>Tabela KLASIFIKACIJA1 predstavlja najvišji nivo hierarhije. |
| SRC_KLASIFIKACIJA2 | Klasifikacija izdelkov je eden od treh načinov razvrščanja izdelkov, pri čemer je to edini hierarhični način razvrščanja izdelkov.<br>Tabela KLASIFIKACIJA2 predstavlja srednji nivo hierarhije. |
| SRC_KLASIFIKACIJA3 | Klasifikacija izdelkov je eden od treh načinov razvrščanja izdelkov, pri čemer je to edini hierarhični način razvrščanja izdelkov.<br>Tabela KLASIFIKACIJA3 predstavlja najnižji nivo hierarhije. |
| SRC_NACIN_IZPISA | Tabela s šifrantom možnih načinov izpisa iz knjižnega kluba. |
| SRC_NACIN_PRISTOPA | Tabela s šifrantom možnih načinov pristopov v knjižni klub. |
| SRC_NAKUPNE_GRUPE | Tabela nakupnih grup vsebuje šifrant, ki predstavlja enega od treh načinov razvrstitev izdelkov. |
| SRC_NAROCILA | Tabela naročil. Tabela ima takšno nenormalizirano obliko že v transakcijskem sistemu, kar sicer ni najbolj običajno. |
| SRC_OBDOBJE | Tabela obdobij. Gre za kvartale poslovanja knjižnega kluba, vse transakcije se namreč zbirajo na nivoju kvartala. |
| SRC_OMREZNE_SKUPINE | Tabela omrežnih telefonskih skupin. |
| SRC_POSTE | Tabela pošt in poštnih številk. |

**Legenda:** [*] Tabele so v podatkovni model, ki je kreiran za potrebe podatkovnega rudarjenja, že prenesene iz izvornega sistema.

**Priloga 4: Analiza tabele SRC_CLANI**

*Tabela 2: Analiza tabele SRC_CLANI*

| Atribut | Tip podatka | % Null vrednosti | Število različnih vrednosti | % različnih vrednosti | Modus | Aritmetična sredina | Mediana | MIN | MAX | Standardni odklon | Varianca | Simetričnost | Sploščenost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLAN | VARCHAR2 | 0 | 328.319 | 96,0739 | 4788329 | | | | | | | | |
| FIZOSEBA | VARCHAR2 | 0 | 318.877 | 93,3109 | 2708730 | | | | | | | | |
| LETOROJ | NUMBER | 11,1958 | 111 | 0,0366 | | 943,9295 | 970 | 0 | 999 | 151,2245 | 22.868,8433 | -5,9933 | 34,303 |
| LETOZAC | NUMBER | 0 | 43 | 0,0126 | | 1.996,90 | 1.999 | 1971 | 2013 | 10,2365 | 104,7854 | -0,7277 | -0,5112 |
| LETOZADZEL | NUMBER | 15,253 | 35 | 0,0121 | | 2.003,43 | 2.004 | 1979 | 2022 | 5,4667 | 29,8844 | -0,504 | -0,0054 |
| NACIZP | VARCHAR2 | 3,4784 | 15 | 0,0045 | P | | | | | | | | |
| NACPRI | VARCHAR2 | 0 | 14 | 0,0041 | 7 | | | | | | | | |
| OBDZAC | NUMBER | 0 | 4 | 0,0012 | | 2,2127 | 2 | 1 | 4 | 1,1268 | 1,2696 | 0,3416 | -1,2978 |
| OBDZADZEL | NUMBER | 15,253 | 4 | 0,0014 | | 2,6044 | 3 | 1 | 4 | 1,1802 | 1,3928 | -0,103 | -1,4915 |
| POSTA | VARCHAR2 | 0 | 632 | 0,1849 | 1000 | | | | | | | | |
| PTTNAZIV | VARCHAR2 | 0 | 598 | 0,175 | LJUBLJANA | | | | | | | | |
| STEVCLAN | NUMBER | 0 | 8 | 0,0023 | | 1,0489 | 1 | 1 | 8 | | | | |
| TOSTEL | VARCHAR2 | 0 | 23 | 0,0067 | 2 | | | | | | | | |

**Priloga 5: Tabela SRC_NACIN_PRISTOPA**

*Tabela 3: Tabela SRC_NACIN_PRISTOPA*

| NACIN_PRISTOPA | NAZIV | PRISTOPNICA |
|---|---|---|
| 0 | OŽIVITEV | 1 |
| 1 | OBNOVITEV | 0 |
| 2 | PODALJŠEVANJE | 0 |
| 3 | PRENOS ČLANSTVA | 0 |
| 4 | PRIJATELJSKO PRIDOBIVANJE | 1 |
| 5 | AKCIJA | 1 |
| 6 | TELEFONSKO | 0 |
| 7 | ZASTOPNIK | 1 |
| 8 | OGLAS | 1 |
| 9 | KLUBSKI CENTER | 1 |
| 10 | INTERNET | 1 |
| 11 | PRISTOP PO VZORU prodajnega kanala A* | 1 |
| 12 | ZAPOSLENI | 0 |
| 13 | PRESTOP IZ knjižnega kluba B* | 1 |

**Legenda:** * Ime prodajnega kanala A in knjižnega kluba B je simbolično.

**Priloga 6: Tabela SRC_NACIN_IZPISA**

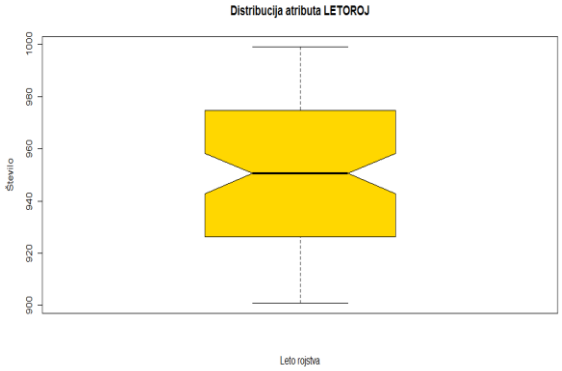
*Tabela 4: Tabela SRC_NACIN_IZPISA*

| NACIN_IZPISA | NAZIV |
|---|---|
| A | ANKETA |
| B | DOGOVOR OB VPISU |
| E | EL. POŠTA |
| I | TOŽBE-NEIZTOŽLJIVI |
| K | KLUBSKI CENTER |
| M | AVTOMATSKO |
| O | OSEBNO |
| P | PISNO |
| S | TS-LISTA |
| T | TELEFONSKO |

**Priloga 7: Podrobnejša analiza atributa LETOROJ > 900**

```
> boxplot(letoroj900.agg$Group.1, notch=TRUE, col="gold", main="Distribucija atributa LETOROJ", xlab="Leto rojstva", ylab="Število", plot=TRUE)
```

*Slika 12: Boxplot graf za vrednosti atributa LETOROJ > 900*




Grafični prikaz lahko nadomestimo tudi s posameznimi statistikami iz zgornjega grafa:

```
> boxplot(letoroj900.agg$x, notch=TRUE, col="gold", main="Distribucija atributa LETOROJ", xlab="Leto rojstva", ylab="Število", plot=FALSE)

$stats
       [,1]
[1,] 901.00              # najnižja vrednost vzorca
[2,] 926.25              # spodnja vrednost razpona (1.kvartil)
[3,] 950.50              # mediana
[4,] 974.75              # zgornja vrednost razpona (3.kvartil)
[5,] 999.00              # največja vrednost vzorca
attr(,"class")

"ore.integer"

$n
[1] 98                   # število različnih vrednosti atributa LETOROJ

$conf
         [,1]
[1,] 942.7592            #  spodnja  vrednost  področja  5%  intervala  zaupanja (notch)
[2,] 958.2408            #  zgornja  vrednost  področja  5%  intervala  zaupanja (notch)

$out
numeric(0)               # število osamelcev

$group
numeric(0)

$names
[1] ""
```

**Priloga 8: Analiza tabele SRC_NAROCILA**

*Tabela 5: Analiza tabele SRC_NAROCILA*

| Atribut | Tip podatka | % Null vrednosti | Število različnih vrednosti | % različnih vrednosti | Modus | Srednja vrednosti | Mediana | MIN | MAX | Standardni odklon | Varianca | Simetričnost | Sploščenost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARTIKEL | VARCHAR2 | 0 | 10581 | 0,286882 | 3831022439328 | | | | | | | | |
| CLAN | VARCHAR2 | 0 | 154173 | 4,180079 | 5968003 | | | | | | | | |
| DATUMNAK | DATE | 0 | 2328 | 0,063119 | | 23.12.08 | 04.12.08 | 03.01.06 | 05.07.12 | | | | |
| GRUPAART | VARCHAR2 | 0 | 137 | 0,003714 | 258 | | | | | | | | |
| GRUPANAK | VARCHAR2 | 0 | 31 | 0,000841 | 20 | | | | | | | | |
| KOLICINA | NUMBER | 0 | 66 | 0,001789 | | 1,02834 | 1 | -1 | 15000 | 9,772213 | 95,49614 | 1119,969 | 1571152 |
| LETONAK | NUMBER | 0 | 13 | 0,000352 | | 2008,484 | 2008 | 2000 | 2012 | 1,813301 | 3,288062 | 0,183532 | -1,07559 |
| NACINPRID | VARCHAR2 | 0 | 15 | 0,000407 | 33 | | | | | | | | |
| NAROCIL | NUMBER | 0 | 1204299 | 32,65205 | | 1695172 | 345356,5 | 1 | 5094537 | 1875124 | 3,52E+12 | 0,457835 | -1,5668 |
| OBDNAK | NUMBER | 0 | 4 | 0,000108 | | 2,526852 | 3 | 1 | 4 | 1,142692 | 1,305745 | -0,0246 | -1,41525 |
| PCENAKK | NUMBER | 0 | 1208 | 0,032752 | | 19,96814 | 15,95 | 0 | 1650 | 19,57409 | 383,1451 | 4,161834 | 130,4148 |
| STORFAK | VARCHAR2 | 0 | 3 | 8,13E-05 | - | | | | | | | | |
| VRSTAKL1 | VARCHAR2 | 0 | 5 | 0,000136 | 1 | | | | | | | | |
| VRSTAKL2 | VARCHAR2 | 0 | 9 | 0,000244 | 1 | | | | | | | | |
| VRSTAKL3 | VARCHAR2 | 0 | 8 | 0,000217 | 3 | | | | | | | | |
| VRSTANAR | VARCHAR2 | 0 | 2 | 5,42E-05 | N32 | | | | | | | | |

**Priloga 9: Tabela pomenov vrednosti atributa VRSTANAR**

*Tabela 6: Tabela pomenov vrednosti atributa VRSTANAR*

| VRSTANAR | Pomen – prodajni kanal |
|---|---|
| N31 | Prodajno naročilo je bilo vneseno v zaledni prodajni sistem, kar pomeni vsa naročila razen vnosa na blagajni v klubskem centru. |
| N32 | Prodajno naročilo je bilo vneseno na blagajni v klubskem centru. |

**Priloga 10: Tabela vrednosti atributa NACINPRID**

*Tabela 7: Tabela vrednosti atributa NACINPRID*

| NACINPRID | Vrednost |
|---|---|
| 30 | KLICNI CENTER |
| 31 | TONSKO NAROČANJE |
| 32 | AVTOMATSKI ODZIVNIK |
| 33 | KLUBSKI CENTER-NAKUP |
| 34 | INTERNET NAROČILO |
| 35 | TELEFONSKO NAROČILO |
| 36 | NAROČILA PREKO PTT |
| 37 | OBDELAVA KNJIGE OBDOBJA |
| 38 | ZASTOPNIŠKA PRODAJA |
| 39 | OPOMINI |
| 40 | E-MAIL NAROČIL |
| 41 | REDNI KATALOG knjižnega kluba |
| 42 | REDNI KATALOG drugega knjižnega kluba |
| 43 | AKCIJSKI KATALOG drugega knjižnega kluba |
| 44 | AKCIJSKI KATALOG knjižnega kluba |
| 45 | IZTERJAVA-DVIG V klubskem centru |
| 46 | SMS NAROČILA |
| 47 | Knjižni klub-NAROČANJE NOVITET |
| 48 | Knjižni klub 48 |
| 49 | Knjižni klub 49 |

**Priloga 11: Tabela vrednosti atributa VRSTAKL1**

*Tabela 8: Tabela vrednosti atributa VRSTAKL1*

| VRSTAKL1 | Vrednost |
|---|---|
| 1 | KNJIGE |
| 2 | REVIJE |
| 3 | AVDIO, VIDEO, CD |
| 4 | GALANTERIJA |
| 5 | PRODAJNI PRIPOMOČKI |
| 6 | ŠPORT |
| 7 | CENTER XXX |
| 9 | NENAČRTOVANO |

**Priloga 12: Tabela vrednosti atributa VRSTAKL2**

*Tabela 9: Tabela vrednosti atributa VRSTAKL2*

| VRSTAKL1 | VRSTAKL2 | Vrednost |
|---|---|---|
| 1 | 1 | LEPOSLOVJE ZA OTROKE IN MLADINO |
| 1 | 2 | PRIROČNIKI ZA OTROKE IN MLADINO |
| 1 | 3 | LEPOSLOVJE |
| 1 | 4 | PRIROČNIKI |
| 1 | 5 | DRUŽBOSLOVJE |
| 1 | 6 | ENCIKLOPEDIJE, LEKSIKONI, JEZIKOSL. |
| 1 | 7 | UČBENIKI |
| 1 | 8 | KNJIGE V TUJIH JEZIKIH |
| 1 | 9 | DARILNI PROGRAM |
| 2 | 1 | REVIJE |
| 2 | 2 | REVIJA RD |
| 3 | 1 | AVDIO, VIDEO, CD |
| 4 | 1 | GALANTERIJA |
| 5 | 1 | PRODAJNI PRIPOMOČKI |
| 6 | 1 | OKS |
| 7 | 1 | KNJ.IN DOD.GRADIVA ZA UČENJE ANGLEŠ |
| 7 | 2 | KNJ.IN DOD.GRADIVA ZA UČENJE NEMŠČI |
| 7 | 3 | KNJ.IN DOD.GRADIVA ZA UČENJE OST.J. |
| 7 | 4 | KARTE IN VODNIKI V TUJIH JEZIKIH |
| 7 | 5 | LEPOSLOVJE V TUJIH JEZIKIH |
| 9 | 1 | NENAČRTOVANO |

**Priloga 13: Tabela vrednosti atributa VRSTAKL3**

*Tabela 10: Tabela vrednosti atributa VRSTAKL3*

| VRSTAKL1 | VRSTAKL2 | VRSTAKL3 | Vrednost |
|---|---|---|---|
| 1 | 1 | 1 | SLIKANICE |
| 1 | 1 | 2 | LEPOSLOVJE ZA OTROKE |
| 1 | 1 | 3 | LEPOSLOVJE ZA MLADINO |
| 1 | 2 | 1 | POBARVANKE, KARTONKE, ... |
| 1 | 2 | 2 | PRIROČNIKI ZA OTROKE |
| 1 | 2 | 3 | PRIROČNIKI ZA MLADINO |
| 1 | 3 | 1 | IZVIRNO LEPOSLOVJE |
| 1 | 3 | 2 | PREVODNO LEPOSLOVJE |
| 1 | 3 | 3 | ŽEPNICE |
| 1 | 3 | 4 | PREVODNO LEPOSLOVJE ŽANRSKO |
| 1 | 3 | 5 | PREVODNO LEPOSLOVJE LITERATURA |
| 1 | 4 | 1 | KNJIGE ZA POSLOVNEŽE |
| 1 | 4 | 2 | NARAVA, VRTNARJENJE |
| 1 | 4 | 3 | KUHARICE |
| 1 | 4 | 4 | DOM, HOBIJI IN ZABAVE |
| 1 | 4 | 5 | PSIHOLOGIJA, OSEBNI RAZVOJ, VZGOJA |
| 1 | 4 | 6 | ZDRAVO ŽIVLJENJE, MEDICINA, NEGA |
| 1 | 4 | 7 | POLJUDNA ZNANOST |
| 1 | 4 | 8 | TURIZEM, POTOPISI, MONOGRAFIJE |
| 1 | 5 | 1 | ZGODOVINA |
| 1 | 5 | 2 | UMETNOST IN GLASBA |
| 1 | 5 | 3 | PUBLICISTIKA |
| 1 | 6 | 1 | ENCIKLOPEDIJA SLOVENIJE |
| 1 | 6 | 2 | ENCIKLOPEDIJE, LEKSIKONI |
| 1 | 6 | 3 | JEZIKOSLOVNI TEČAJI, KNJIGE |
| 1 | 6 | 4 | SLOVARJI |
| 1 | 6 | 5 | ATLASI |
| 1 | 6 | 6 | REVIJA A |
| 1 | 7 | 1 | UČB., OBVEZNI-8L.OŠ |
| 1 | 7 | 2 | UČB., DODATNI-8L.OŠ |

se nadaljuje

nadaljevanje

| VRSTAKL1 | VRSTAKL2 | VRSTAKL3 | Vrednost |
|---|---|---|---|
| 1 | 7 | 3 | UČB., OBVEZNI-9L.OŠ |
| 1 | 7 | 4 | UČB., DODATNI-9L.OŠ |
| 1 | 7 | 5 | UČB., OBVEZNI-SŠ |
| 1 | 7 | 6 | UČB., DODATNI-SŠ |
| 1 | 7 | 7 | ATLASI ZA ŠOLO |
| 1 | 7 | 8 | ZEMLJEVIDI |
| 1 | 8 | 1 | KNJIGE-KNJIGE V HRVAŠKEM JEZIKU |
| 1 | 8 | 2 | KNJIGE-TUJI JEZIKI |
| 1 | 8 | 3 | KNJIGE-KNJIGE V MAKEDONSKEM JEZIKU |
| 1 | 8 | 4 | KNJIGE-KNJIGE V SRBSKEM JEZIKU |
| 1 | 8 | 5 | KNJIGE-KNJIGE ZA BOSANSKI TRG |
| 1 | 8 | 6 | KNJIGE-KNJIGE ZA BOLGARSKI TRG |
| 1 | 8 | 7 | KNJIGE-KNJIGE ZA ROMUNSKI TRG |
| 1 | 9 | 1 | DARILNE KNJIŽICE |
| 2 | 1 | A | STARŠ-SI (SOS) |
| 2 | 1 | B | MINI MOJ PLANET |
| 2 | 1 | C | CICIZABAVNIK |
| 2 | 1 | 1 | CICIBAN |
| 2 | 1 | 2 | GEA |
| 2 | 1 | 3 | PIL - PLUS |
| 2 | 1 | 4 | CICIDO |
| 2 | 1 | 5 | DINO (ALBERT) |
| 2 | 1 | 6 | PIL |
| 2 | 1 | 7 | POLIGLOT (TELEBAJSKI) |
| 2 | 1 | 8 | MOJ PLANET |
| 2 | 1 | 9 | ENIGMA |
| 2 | 2 | 1 | REVIJA A |
| 3 | 1 | 1 | AVDIO KASETE |
| 3 | 1 | 2 | VIDEO KASETE |
| 3 | 1 | 3 | CD PLOŠČE |
| 3 | 1 | 4 | CD ROM |
| 3 | 1 | 5 | DVD |
| 4 | 1 | 1 | KOLEDARJI |
| 4 | 1 | 2 | UKINJENO |

se nadaljuje

nadaljevanje

| VRSTAKL1 | VRSTAKL2 | VRSTAKL3 | Vrednost |
|---|---|---|---|
| 4 | 1 | 3 | RAZGLEDNICE, ČESTITKE |
| 4 | 1 | 4 | IGRAČE, DRUŽABNE IGRE |
| 4 | 1 | 5 | ŠOLSKE POTREBŠČINE |
| 4 | 1 | 6 | DRUGA GALANTERIJA |
| 4 | 1 | 7 | SVET ŽELJA |
| 5 | 1 | 1 | TISKOVINE, KATALOGI |
| 5 | 1 | 2 | STOJALA |
| 5 | 1 | 3 | IZDELKI ZA POSP. PRODAJE |
| 5 | 1 | 4 | DARILA ZA KUPCE |
| 5 | 1 | 5 | DRUGO |
| 6 | 1 | 1 | OKS |
| 7 | 1 | 1 | OUPELT-KNJ. IN DOD. GRADIVA ANG. J. OUP |
| 7 | 1 | 2 | OUPACA-DRUGE KNJ. IN DOD. GRAD. OUP |
| 7 | 1 | 3 | CUPELT-KNJ. IN DOD. GRADIVA ANG. J. CUP |
| 7 | 1 | 4 | CUPACA-DRUGE KNJ. IN DOD. GRAD. CUP |
| 7 | 1 | 5 | MCMELT-KNJ. IN DOD. GRAD. ANG. J. MACMIL |
| 7 | 1 | 6 | ELTD-KNJ .IN DOD. GRAD. DRUGIH ZALOZB |
| 7 | 2 | 1 | DAFL-KNJ. IN DOD. GRAD. NEM. J.-LANGENS |
| 7 | 2 | 2 | DAFH-KNJ. IN DOD. GRAD. NEM. J.-HUEBER |
| 7 | 2 | 3 | DAFD-KNJ. IN DOD. GRAD. NEM. J. DR. ZALOZ |
| 7 | 3 | 1 | ITA-KNJ. IN DOD. GRAD. ITALIJANŠČINA |
| 7 | 3 | 2 | SPAN-KNJ. IN DOD. GRAD. ŠPANŠČINA |
| 7 | 3 | 3 | FRAN-KNJ. IN DOD. GRAD. FRANCOŠČINA |

se nadaljajuje

nadaljevanje

| VRSTAKL1 | VRSTAKL2 | VRSTAKL3 | Vrednost |
|---|---|---|---|
| 7 | 3 | 4 | DRUGO-VSE K<br>NJ. IN DOD. GRADIVA OSTALO |
| 7 | 4 | 1 | TURISTIKA-KARTE IN VODNIKI ANG. ZALO |
| 7 | 4 | 2 | TURISTIKAD-KARTE IN VODNIKI DRUGIH |
| 7 | 5 | 1 | LEPOA-LEPOSLOVJE V ANGLEŠKEM JEZIKU |
| 7 | 5 | 2 | LEPON-LEPOSLOVJE V NEMŠKEM JEZIKU |
| 7 | 5 | 3 | LEPOS-LEPOSLOVJE V ŠPANSKEM JEZIKU |
| 7 | 5 | 4 | LEPOSD-LEPOSLOVJE V DRUGIH JEZIKIH |
| 9 | 1 | 1 | NENAČRTOVANO |

**Priloga 14: Transformacija atributa LETOROJ in statistike vmesnih atributov**

*Tabela 11: Statistika novega atributa LETO_ROJSTVA*

| | |
|---|---|
| Odstotek NULL vrednosti | 11,0945 |
| Različnih vrednosti | 93 |
| Modus | |
| Povprečna vrednost | 1.969,2552 |
| Mediana | 1.972 |
| Najnižja vrednost | 1909 |
| Najvišja vrednost | 2010 |
| Standardni odklon | 16,3527 |
| Varianca | 267,4095 |
| Simetričnost | -0,4307 |
| Sploščenost | 0,1949 |

*Tabela 12: Statistika transformiranega atributa LETO_ROJSTVA_OUT_MIS*

| | |
|---|---|
| Odstotek NULL vrednosti | 0 |
| Različnih vrednosti | 68 |
| Modus | |
| Povprečna vrednost | 1.969,4283 |
| Mediana | 1.969,4283 |
| Najnižja vrednost | 1936.5498 |
| Najvišja vrednost | 2001.9605 |
| Standardni odklon | 14,6779 |
| Varianca | 215,4405 |
| Simetričnost | -0,315 |
| Sploščenost | -0,0903 |

**Priloga 15: Opisne statistike atributa STAROST**

*Tabela 13: Statistika novega atributa STAROST*

| | |
|---|---|
| Odstotek NULL vrednosti | 0 |
| Različnih vrednosti | 66 |
| Modus | |
| Povprečna vrednost | 43,6047 |
| Mediana | 44 |
| Najnižja vrednost | 11 |
| Najvišja vrednost | 76 |
| Standardni odklon | 14,6498 |
| Varianca | 214,6172 |
| Simetričnost | 0,2953 |
| Sploščenost | -0,1097 |

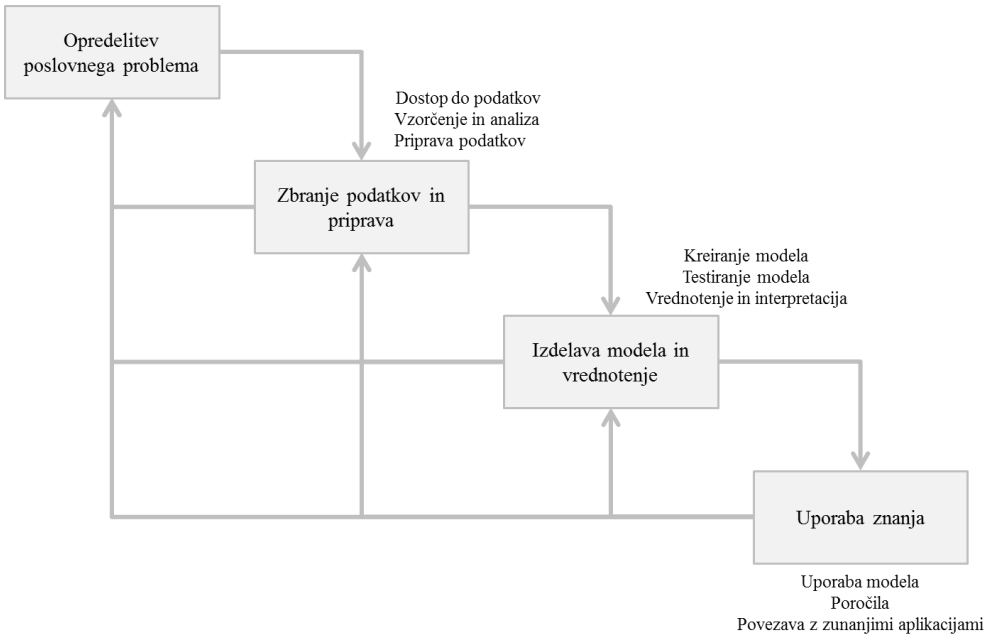
**Priloga 16: Analiza tabele STG_NAROCILA_2**

*Tabela 14: Analiza tabele STG_NAROCILA_2*

| Atribut | Tip podatka | % Null vred. | Število različnih vrednosti | % različnih vrednosti | Modus | Srednja vrednost | Mediana | MIN | MAX | Stand. odklon | Var. | Simetr. | Sloščen. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLAN | VARCHAR2 | 0 | 1.955 | 98,6875 | 1572304 | | | | | | | | |
| GRUPAART_BIN | VARCHAR2 | 0 | 11 | 0,5553 | Other | | | | | | | | |
| GRUPANAK_BIN | VARCHAR2 | 0 | 11 | 0,5553 | Other | | | | | | | | |
| KLASIFIKACIJA3_BIN | VARCHAR2 | 0 | 21 | 1,0601 | Other | | | | | | | | |
| KVARTAL_NAKUPA | VARCHAR2 | 0 | 27 | 1,3629 | 2007-Q4 | | | | | | | | |
| NACINPRID | VARCHAR2 | 0 | 12 | 0,6058 | 33 | | | | | | | | |
| STAROST_KUPCA_OUT_MIS | NUMBER | 0 | 61 | 3,0793 | | 41,1514 | 39 | 15 | 75 | 15,0548 | 226,6478 | 0,2972 | -0,6828 |
| STAROST_KUPCA_OUT_MIS_BIN | VARCHAR2 | 0 | 9 | 0,4543 | 30 - 36 | | | | | | | | |
| VREDNOST_NAKUPA | NUMBER | 0 | 307 | 15,4972 | | 20,0369 | 15,95 | 0 | 159.5 | 19,0807 | 364,0743 | 2,4313 | 8,2237 |
| VRSTANAR | VARCHAR2 | 0 | 2 | 0,101 | N32 | | | | | | | | |

## Priloga 17: Kontinuiran proces podatkovnega rudarjenja po Oraclu

*Slika 13: Kontinuiran proces podatkovnega rudarjenja*




*Vir: Oracle, Oracle Data Mining Concepts, 11g Release 1 (11.1), 2008, str. 1/5*