UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

DIPLOMSKO DELO

# PRIMERJALNA ANALIZA ODPRTOKODNIH PROGRAMOV ZA PODATKOVNO RUDARJENJE

Ljubljana, avgust 2013 ŽIGA VLAHUŠIĆ

## IZJAVA O AVTORSTVU

Spodaj podpisani Žiga Vlahušić, študent Ekonomske fakultete Univerze v Ljubljani, izjavljam, da sem avtor diplomskega dela z naslovom Primerjalna analiza odprtokodnih programov za podatkovno rudarjenje, pripravljenega v sodelovanju s svetovalcem dr. Jurijem Jakličem.

Izrecno izjavljam, da v skladu z določili Zakona o avtorski in sorodnih pravicah (Ur. l. RS, št. 21/1995 s spremembami) dovolim objavo zaključne strokovne naloge/diplomskega dela/specialističnega dela/magistrskega dela/doktorske disertacije 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 diplomskem 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 predložene zaključne diplomskega 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

## KAZALO SLIK



## KAZALO TABEL

# UVOD

Tri tisoč let pred našim štetjem so ob velikih rekah (Nil, Evfrat, Tigris) nastale prve visoke civilizacije. Značilnosti teh civilizacij so bile namakanje polj, urbano življenje, gradnja stavb in - kar je za nas najbolj pomembno - uporaba pisave. V času, ko človek še ni poznal papirja, so ljudje v mehko glino s trsnim pisalom vrezovali znamenja z dvema namenoma. Prvi namen je bil zbrati podatke, drugi pa s pomočjo analize ugotoviti, kako ti podatki vplivajo na njihova življenja in kako jih je mogoče s pomočjo pridobljenih informacij izboljšati. Podobno ravnamo še danes, ko zbiramo podatke in iz njih luščimo pomembne informacije. Danes temu procesu oz. tej tehniki obdelave podatkov rečemo podatkovno rudarjenje in je veliko obsežnejše kot je bilo pet tisoč let nazaj. Podatkovno rudarjenje je proces raziskovanja velike količine podatkov, da bi odkrili uporabne vzorce in pravila. Sami, surovi podatki nam v velikih količinah ne koristijo in izgubijo svoj pomen, s pomočjo pravih orodij pa je možno iz njih izvleči koristne informacije, ki nam bodo v prihodnosti služile pri boljšem in hitrejšem odločanju.

Odprta koda je izraz, pod katerega štejemo različna licencirana avtorska dela, za katera je značilno, da je koda v prosti uporabi, na voljo vsakomur, pod enakimi pogoji (ugodnosti in znanje). Nekateri avtorji odprto kodo razumejo kot enega izmed mnogih možnih pristopov načrtovanja, drugi pa jo razumejo kot bistven strateški element svojega delovanja (Chen, Yunming, Williams & Xu, 2007, str. 2-3). Izraz odprta koda je postal priljubljen z vzponom interneta, ta je namreč omogočil dostop do različnih modelov produkcije, komunikacijskih poti in interaktivnih skupnosti. Principi in prakse se po navadi nanašajo na razvijanje izvorne kode programov, ki so razpoložljivi za javno sodelovanje – odprto programje. Filozofija širjenja odprtokodnih rešitev običajno pomeni, da je rešitev/orodje delo skupnosti, ki ni nujno podprta s strani kakšne organizacije, temveč je skupek raziskovanj domačih in tujih razvijalcev na tem področju. Takšen način razvoja orodij/rešitev pomeni vključitev različnih praks, izkušenj in pogledov razvijalcev, osredotočenih na samo eno platformo. Odprtokodna orodja mogoče niso tako stabilna in vizualno privlačna kot komercialne različice, vendar ponujajo visoko uporabnost zaradi aktualnih tehnik raziskovanja ter prijaznih in zanimivih vmesnikov, sama odprtokodnost pa jim daje dodatno vrednost v fleksibilnosti razširljivosti in možnost implementacije različnih vrst podatkov (Zupan & Demšar, 2008, str. 38).

Trg programske opreme je v zadnjih 15. letih doživel velik razcvet. V tem tako kratkem času je bila ustvarjena takorekoč vsa programska oprema, ki je trenutno v uporabi. Zaradi načina licenciranja oz. prodaje te programske opreme je dejavnost dajala velike zaslužke. Z razvojem odprte kode so na trg stopili novi proizvodi, ki imajo drugačen način licenciranja in prostega razmnoževanja. Odprtokodne rešitve so danes stalno prisotne in zdi se, da uporabniki pogosto dajejo prednost odprtokodnim rešitvam v primerjavi s komercialnim rešitvami.

Pa je temu res tako? So odprtokodne rešitve bolj zmogljive, zanesljive, varne in bolj enostavne za uporabo? Prve odprtokodne rešitve za podatkovno rudarjenje so bile razvite v 90. letih prejšnjega stoletja in se do danes niso veliko spremenile. Seveda so jim bile dodane nove funkcionalnosti in spremenila se je njihova grafična podoba, vendar pa je namen samih programov ostal enak do danes. Za primer lahko vzamemo programsko rešitev WEKA, ki je »stara« že dobrih 20 let in jo še vedno uporablja kar osmina vseh svetovnih uporabnikov odprtokodnih rešitev za podatkovno rudarjenje (KDnuggets, 2013). Na podlagi teh vprašanj in ugotovitev sem se odločil za analizo odprtokodnih rešitev za podatkovno rudarjenje, ki so prosto dostopne na spletu.

Namen diplomske naloge je primerjalna analiza odprtokodnih orodij za podatkovno rudarjenje, ki bi potencialnim uporabnikom omogočila izbiro glede na njegove potrebe, obenem pa želim predstaviti tehnike podatkovnega rudarjenja kot tekmovalno prednost podjetij na različnih gospodarskih področjih.

Diplomska naloga je razdeljena na teoretični in empirični del. Prvi del temelji na preučevanju teorije podatkovnega rudarjenja, umeščenega v poslovni svet in na vplivu in pomenu odprtokodnih rešitev v svetu. V tem delu bom obravnaval tehnike in procese podatkovnega rudarjenja ter pomen in značilnosti odprtokodnih rešitev v poslovni in zasebni sferi. V drugem delu pa bom predstavil odprtokodna orodja za podatkovno rudarjenje, jih preizkusil in na podlagi preizkusov analiziral njihove značilnosti, uporabnosti, sposobnosti zajemanja podatkov, tehnike analize ter prednosti in slabosti. Končni rezultat torej ni »najboljše« orodje, pač pa ovrednotenje izbranih orodij po vnaprej izbranih kriterijih, ki omogočajo uporabniku, da si glede na njegova pričakovanja izbere tisto, ki mu v tistem trenutku ustreza. V sklepu bom analiziral dobljene rezultate.

# 1 POSLOVNA INTELIGENCA IN PODATKOVNO RUDARJENJE

Ko govorimo o informacijskih sistemih in njihovem razvoju, skozi leta izstopa predvsem ena stvar. To je izjemno hitra rast količine podatkov in njihova hramba ter obdelava. Analitika Grantz in Reinsel iz podjetja IDC (angl. *International Data Corporation*) ugotavljata, da je to šele začetek eksplozije, ki naj bi do leta 2020 količino podatkov povečala za 44 krat in sicer na 35 trilijonov gigabajtov (Grantz & Reinsel, 2010, str. 1). Za primerjavo: če bi vse podatke zapisali na nosilce medijev DVD, bi kup segal do pol poti do Marsa!

Organizacije, kot so na primer verige supermarketov, bank, zavarovalnic in konec koncev tudi bolnišnice, vsak dan proizvedejo velike količine podatkov, a se vseeno pogosto govori o tem, kako se organizacije »utapljajo« v podatkih, obenem pa so »lačne« kakovostnih informacij (Han & Kamber, 2006, str. 4). Informacije so temeljni gradnik znanja, istočasno pa so grajene na podatkih. Če ljudje ne posedujemo dobrih podatkov, iz njih ne bomo mogli izvleči koristnih informacij, kar pomeni, da bomo s svojim slabim ravnanjem škodovali

organizaciji. Velika količina podatkov ne pomeni nujno njihove visoke kakovosti, vrednosti in ustreznosti v procesu odločanja, zato je pomembno, da se dokopljemo do čim bolj koristnih podatkov. V samem procesu analize podatkov lahko odpade tudi do 95 % podatkov, saj se jih analizira samo majhen del, kljub temu pa vse te podatke shranjujemo na podatkovnih nosilcih, navkljub visokim stroškom hrambe. Večina podatkov po mnenju računalniških analitikov ni unikatna. Okoli 75% podatkov je podvojenih, kar z drugimi besedami pomeni, da edinstveno vsebino predstavlja le dobra četrtina vseh podatkov, ki jih hranimo (Grantz & Reinsel, 2010, str. 1).

## 1.1 POSLOVNA INTELIGENCA

Izraz poslovna inteligenca (angl. *Business Inteligence*) opisuje celoto metodologij in konceptov, ki služijo zbiranju, analizi, hrambi in dostopu do podatkov s pomočjo programske opreme za učinkovitejše poslovno odločanje. Tehnologija poslovne inteligence pomaga poslovnim subjektom izkoristiti priložnosti in vpeljati učinkovite strategije za odkrivanje konkurenčnih prednosti in dolgoročne stabilnosti podjetja (Parr Rud, 2009, str. 11-18). S pomočjo poslovne inteligence lahko odgovorimo na vprašanja, kot so: »Kje so ozka grla v vašem poslovnem procesu? Kakšni so trendi v vašem podjetju? Ali je vaša strateška usmeritev prava?«

Poslovna inteligenca zagotavlja orodje za analizo podatkov in njihovo pretvorbo v uporabne informacije. Omogoča hiter in enostaven dostop do podatkov ter izmenjavo informacij med zaposlenimi in vsebuje orodja za večdimenzionalni prikaz analiz in poizvedovanje na zahtevo. V podjetju Bilab (Poslovna inteligenca, 2013) ocenjujejo, da je glavni namen oz. korist poslovne inteligence preoblikovanje podatkov v informacije, ki se preoblikujejo v znanje, znanje pa v dobiček organizacije. Na Sliki 1 so prikazani gradniki poslovne inteligence, ki služijo kot osnova vsakega sistema poslovne inteligence.

*Slika 1: Vloga podatkovnega rudarjenja v procesu poslovne inteligence*


*Vir: O. P. Rud, Business Intelligence Success Factors: Tools for Aligning Your Business in the Global Economy, 2009, str. 54.*

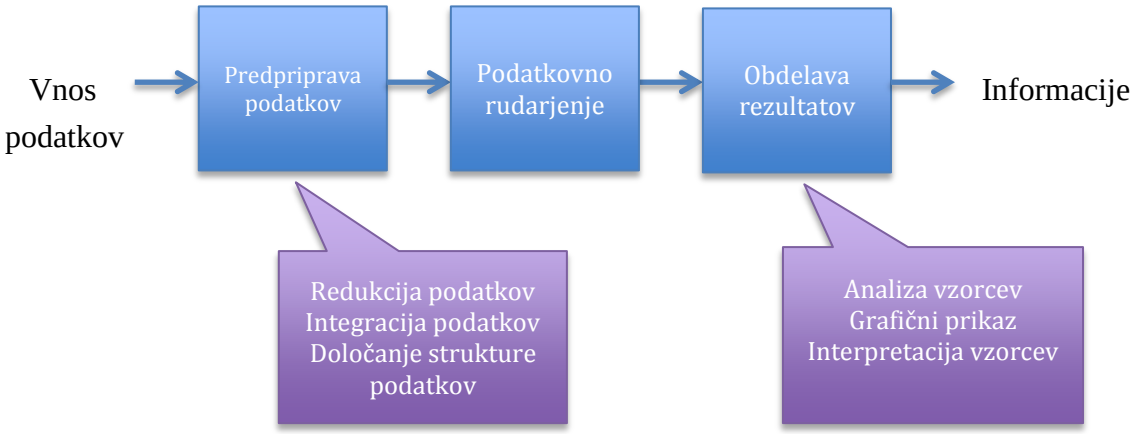
## 1.2 PODATKOVNO RUDARJENJE

Z razvojem tehnologije, računalnikov in interneta je organizacija podatkov postala lažje opravilo, in če želimo te podatke spremeniti v koristne, jih je treba pretvoriti v informacije in posledično znanje. Odkrivanje znanja iz podatkov (angl. *Knowledge Discovery from Data*) je proces odkrivanja doslej neznanega in potencialno uporabnega znanja iz podatkov. Del tega procesa, prikazanega na Sliki 2, je tako imenovano podatkovno rudarjenje (angl. *Data Mining*). Sam termin bi najlažje opisali kot proces odkrivanja koristnega znanja ali informacij iz velike količine podatkov (Berry & Linoff, 2000, str. 5-7).

Podatkovno rudarjenje je v osnovi ozko povezano s statističnimi pojmi in postopki. V večini postopkov podatkovnega rudarjenja obstaja štetje kupa podatkov in primerjava dobljenih rezultatov. Za razliko od statistike, ki je matematična veda in kot taka usmerjena v iskanje in verjetnostno vrednotenje odnosov, ki nastajajo med podatki, je podatkovno rudarjenje tehnična veda, ki izboljša proces podpori odločanja na strateško-poslovni ravni in nam daje vpogled v »skrite« podatke, prav tako pa nam rudarjenje po podatkih odkriva odnose, pravilnosti, trende, vzorce in ostale odnose med podatki. Iskanje znanja v podatkovnih bazah je netrivialen proces identifikacije veljavnih, neobičajnih, potencialno uporabnih in predvsem razumljivih vzorcev iz podatkov (Witten & Frank, 2005, str. 9). Izvir netrivialnosti je v neobstoju univerzalnega postopka rešitve, ki zahteva zapleteno iskanje in veliko izkušenj.

Obstaja veliko definicij, kaj podatkovno rudarjenje pravzaprav je. Avtorji knjige Introduction to DataMining (Tan, Steinbach & Kumar, 2006, str. 3) so podatkovno rudarjenje opredelili kot notranji proces pri odkrivanju znanja iz podatkov, ki pa je končni proces pri pretvorbi surovih podatkov v uporabne informacije.

*Slika 2: Proces odkrivanja znanja iz podatkovnih baz*




*Vir : P.N. Tan, M. Steinbach & V. Kumar, Introduction to DataMining (2nd ed.), 2006, str. 3.*

Definicija, ki meni najbolje opiše vse vidike podatkovnega rudarjenja, pa je: »Podatkovno rudarjenje je proces odkrivanja zanimivih in pomembnih vzorcev ter znanja iz ogromne količine podatkov, shranjenih v podatkovnih bazah, podatkovnih skladiščih in ostalih informacijskih odlagališčih.« (Han & Kamber 2006, str. 5)

## 1.3 UPORABA IN UPORABNOST PODATKOVNEGA RUDARJENJA

V poslovnem svetu se podatkovno rudarjenje uporablja v vseh panogah gospodarstva, medicine, politike in celo športa. Če smo sposobni neko informacijo obrniti sebi v prid, nam prinaša korist in prednost pred ostalimi organizacijami. Dobri menedžerji poslovnih subjektov bi za dobre informacije plačali velike vsote, vendar ni vsak od njih računalniški strokovnjak, zato še vedno potrebujemo analitike, ki dobljene rezultate predstavijo vodstvu organizacije na enostaven način. Podatkovno rudarjenje se danes uporablja v najbolj zahtevnih znanstvenih raziskavah in eksperimentih. Po drugi strani pa se ne zavedamo, da je podatkovno rudarjenje del našega vsakdana, saj ta proces poteka tudi med uporabo mobitela, interneta ali bančne kartice (Monk & Wagner, 2006, str. 13).

Spodaj je naštetih nekaj primerov, kjer lahko uporabimo podatkovno rudarjenje in zakaj (Pivk, 2001, str. 14):

- BANČNIŠTVO
  - Odkrivanje goljufij z iskanjem sumljivih transakcij subjektov
  - Odobritev posojil na banki, ko ugotovimo mejo posojila
  - Analiza investicij, napoved portfolja vračila investicije

- TRŽENJE
  - Iskanje dobičkonosnih strank
  - Izdelava profilov kupcev
  - Segmentacija kupcev (iskanje kupcev s podobnim vzorcem obnašanja)
  - Raziskava povezanosti različnih proizvodov
  - Direktni marketing
  - Stimulacija kupcev za nakup drugih izdelkov istega podjetja
  - Zadrževanje kupcev

- MEDICINA
  - Analiza vpliva zdravil
  - Analiza vzrokov bolezni
  - Napovedovanje širjenja bolezni

- POLITIKA
  - Analiza volitev iz vzorcev volivcev
  - Analiza družbene problematike

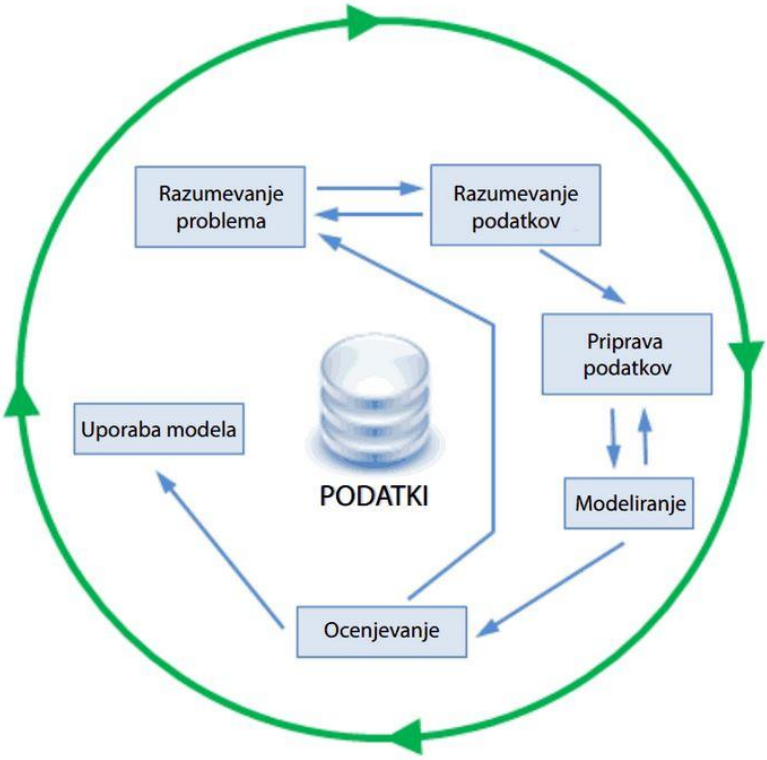
- PROIZVODNJA
  - Analiza proizvodnih procesov za ugotovitev problemov v proizvodnji
  - Analiza rezultatov poizkusa in ustvarjanje napovednih modelov

## 1.4 PROCES PODATKOVNEGA RUDARJENJA

V svetu obstaja veliko modelov, ki opisujejo proces podatkovnega rudarjenja. Nastali so v različnih časih iz različnih vzrokov in zato dajejo različne poglede na podatkovno rudarjenje. Eden od največkrat uporabljenih modelov je model CRISP-DM (Podatkovno rudarjenje, 2013), ki definira podatkovno rudarjenje kot iterativni proces, sestavljen iz več faz. Opisuje običajne pristope, ki jih koristijo strokovnjaki iz tega področja, da bi dobili odgovore na določene probleme.

Model CRISP (angl. *CRoss Industry Standard Process for Data Mining*) so leta 1997 razvili strokovnjaki iz različnih organizacij, različnih znanj in izkušenj, pod pokroviteljstvom Evropske unije (Shearer et al., 2000, str. 13). Sestavljen je iz šestih osnovnih korakov, ki so prikazani na Sliki 3. Prednost modela CRISP je njegova nevtralnost ne glede na orodje in panogo, v kateri ga bomo uporabili. CRISP daje stabilnost in smernice razvoja procesa podatkovnega rudarjenja. V nadaljevanju so razložene osnovne faze modela.

*Slika 3:Proces podatkovnega rudarjenja*




*Vir: P. Chapman et al.,CRISP-DM 1.0, 2000, str.13.*

- **Razumevanje problema**

Prva faza je usmerjena v razumevanje ciljev projekta in zahtev z vidika poslovanja.
Cilji se spremenijo v definicijo problema in so najpomembnejši korak za dosego le-teh.

- **Razumevanje podatkov**

Ta faza se začne ko začnemo z zbiranjem podatkov. Vse podatke je potrebno preučiti in pregledati glede na njihove značilnosti.

- **Priprava podatkov**

V tej fazi se lotimo čiščenja, bogatenja in transformacije podatkov. Ukvarjamo se s procesom ETL (angl. *Extract, Transform and Load*), kar pomeni da podatke izvlečemo, pretvorimo in naložimo, saj v veliki meri odločajo o kvaliteti končnega produkta analize.

- **Modeliranje**

Pod modeliranje spada izbira različnih tehnik modeliranja in njihovo uporabo glede na vhodne podatke. V tem procesu so vključeni tudi parametri uravnavanja parametrov. Na splošno obstaja več rešitev določenega problema, tako da ni univerzalnega orodja za rešitev, vendar pa določene metode zahtevajo podatke v določeni obliki, tako da je potrebno

vračanje na fazo priprave podatkov. Pred samim modeliranjem se tukaj odločimo, katerega izmed treh podatkovnih setov  podatkov bomo uporabili: Prva možnost je set za učenje, ki predstavlja podatke, na katerih gradimo model. Druga je ocenjevalni set, ki ga sestavljajo podatki, na katerih se optimizirajo parametri modela in tretja možnost testni set, ki je sestavljen iz podatkov, ki niso uporabljeni v gradnji modela in na katerih testiramo naš model. Če je podatkovni set majhen, obstajajo različne metode, ki omogočajo učinkovito izkoriščanje vseh podatkov.

- **Evalvacija modela**

Modele je potrebno detajlno pregledati in na koncu tudi razumeti, ali je njihov rezultat zadovoljiv in ali rešujejo poslovni problem. V tej fazi tudi ugotovimo, ali obstaja povezanost med podatki, zakonitost ali pa odgovori na vprašanja. Rezultat te faze procesa bo odločitev, ali bomo model uporabili.

- **Uporaba modela**

Proces podatkovnega rudarjenja se ne zaključi z uporabo modela. V tej fazi je potrebno rezultate predstaviti vodstvu, da jih bodo razumeli. Pogosto se namreč zgodi, da te faze ne izvaja tisti, ki je delal v procesu podatkovnega rudarjenja, temveč tisti, ki je analizo naročil.

# 2 NALOGE PODATKOVNEGA RUDARJENJA

Načeloma delimo naloge podatkovnega rudarjenja v dve kategoriji: napovedne naloge in opisovalne naloge. S pomočjo prvega se izdelajo modeli, nakar poizkušamo napovedati vrednost atributa na podlagi drugih, pri drugem pa je cilj opisati podatke in njihove lastnosti in s pomočjo tega najti vzorce v podatkih.  Glede na to, kakšne vzorce iščemo, lahko naloge podatkovnega rudarjenja razdelimo na opisovalne in napovedovalne (Tan, Steinbach & Kumar, 2006, str. 7). Znotraj obeh skupin pa ju delimo na naloge, opisovanje in zbiranje podatkov, analizo skupin in asociacij, klasifikacijo, napovedovanje in analizo odvisnosti.

## 2.1 OPISOVANJE IN ZBIRANJE PODATKOV

S pomočjo opisovanja in zbiranja podatkov dobimo natančen opis lastnosti podatkov. Opisovanje in zbiranje podatkov je lahko samostojen projekt v procesu podatkovnega rudarjenja. Na primer, prodajalec si želi pregledati prihodke vse trgovin, podatki pa naj bodo razdeljeni po kategorijah, kjer so razvidne spremembe glede na prejšnje obdobje. V skoraj vseh projektih podatkovnega rudarjenja je opisovanje in zbiranje podatkov eden izmed ciljev raziskave, največkrat je to na začetku raziskav, ko lahko s pomočjo analize razumemo naravo podatkov in podamo hipoteze o skritih informacijah. Zbiranje ima pomembno vlogo na koncu pri predstavitvi rezultatov. Večina informacijskih sistemov za podporo odločanja in informacijskih sistemov za menedžment je sposobnih opisovanja in zbiranja podatkov, vendar ne zmorejo naprednih tehnik modeliranja (Pyle, 2003, str. 97-99).

## 2.2 ANALIZA SKUPIN IN ASOCIACIJ

Analiza asociacij razdeli podatke v zanimive in logične podskupine ali razrede s podobnimi atributi. Primer nakupovalne košarice je najbolj znan primer uporabe segmentacije podatkov. Analitiki raziskujejo posebne skupine in podskupine, ki so pomembne za naš poslovni proces na podlagi znanja, pridobljenega na podlagi opisovanja in zbiranja podatkov. Obstajajo tudi avtomatski procesi analize skupin, ki lahko zaznajo odnose med podatki, čeprav so bile te prej nezaznavne. Po drugi strani pa obstaja možnost, da ne zmoremo ali ne znamo ustvariti skupin na podlagi velikega seta podatkov, saj so vrednosti preveč raznolike. Analiza asociacij je precejšen korak k rešitvi problema predvsem tam, kjer je velikost podatkov dovolj majhna za uporabo ali kjer je lažje odkriti homogene skupine elementov (North, 2012, str. 73).

Primer:
Avtomobilska hiša pri vsakem nakupu zbira podatke o svojih kupcih glede na njihove potrošniške navade. S pomočjo analize skupin lahko avtomobilska hiša razdeli svoje kupce v bolj razumljive skupine, ki jih nato analizira in priredi marketinško strategijo za vsako izmed njih.

## 2.3 KLASIFIKACIJA

Pri klasifikaciji predvidevamo, da obstaja nabor objektov brez razreda, ki so razdeljeni v različne razrede. Uporabljamo jo za napovedovanje diskretnih spremenljivk. Cilj je sestaviti model za klasifikacijo, ki bodo tistim podatkom, ki razreda še nimajo, le-tega določili (Tan, Steinbach & Kumar, 2006, str. 145). Največkrat uporabljamo tehniko klasifikacije pri napovednem modeliranju, ki napove vrednost napovedne spremenljivke.

Primer:
Banke, ki imajo na voljo informacije o plačilnih sposobnostih jemalcev kreditov. Z združitvijo finančnih informacij z ostalimi informacijami, kot so na primer spol, starost in dohodek, je mogoče razviti sistem, ki bo razvrščal nove komitente na dobre in slabe. Model klasifikacije lahko potem uporabimo pri oznaki novi potencialnih komitentov in na podlagi modela stranko potrdimo ali pa zavrnemo.

## 2.4 NAPOVEDOVANJE

Napovedovanje je zelo podobno klasifikaciji, ampak za razliko od nje spremenljivka ni diskretna, temveč zvezna. Cilj napovedovanja je najti številsko vrednost za atribute v prihodnosti in ugotoviti, na kakšen način bo napovedana vrednost čim manj odstopala od dejanske vrednosti (Pyle, 1999, str. 109-111).

Primer:
Letni prihodek v podjetju je povezan z atributi kot so recimo oglaševanje, stopnja inflacije, marketing in proizvodnja. S pomočjo teh vrednosti lahko v podjetju napovejo pričakovane prihodke za naslednje leto.

## 2.5 ANALIZA ODVISNOSTI

S pomočjo analize odvisnosti lahko poiščemo model, ki opisuje značajne odvisnosti med podatki ali dogodki. Analiza odvisnosti se lahko uporablja za napoved določene vrednosti podatkov glede na dano vrsto podatkov. Čeprav se lahko modeli odvisnosti uporabljajo kot napovedno modeliranje, se največkrat uporabljajo za razumevanje tipa odvisnosti med podatki. Analiza odvisnosti je v veliki meri povezana z napovedovanjem in klasifikacijo, kjer so odvisnosti implicitno izražene kot temelj gradnje napovednih modelov, pri zgradbi aplikacije pa odvisnosti v veliki meri sovpadajo s segmentacijo. (Shearer et al., 2000, str. 19-21). Na velikem naboru podatkov so odvisnosti pomembne zaradi različnih prekrivajočih se atributov. V takšnih primerih je priporočljivo narediti analizo odvisnosti na manjših podatkovnih setih.

Primer:
S pomočjo regresijske analize lahko finančni analitik najde pomembne povezave med celotno prodajo produkta in njegovo ceno, ter celotnimi stroški oglaševanja, namenjenega temu proizvodu. S tem znanjem lahko analitik doseže zaželeno prodajo, s tem ko spreminja ceno produkta in ceno njegovega oglaševanja.

# 3 TEHNIKE PODATKOVNEGA RUDARJENJA

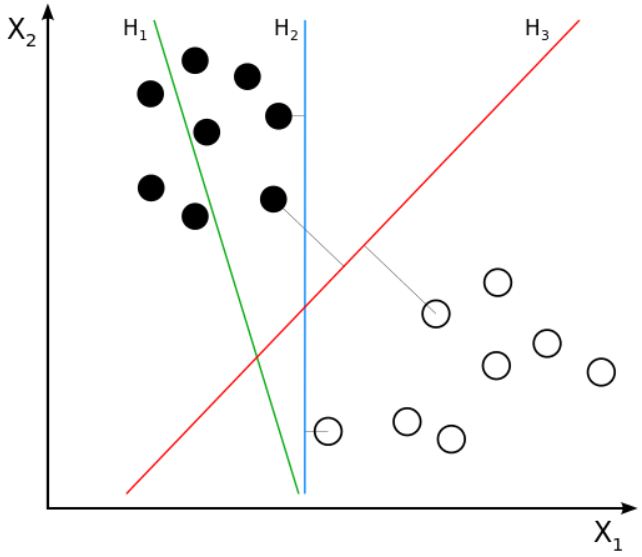
V tem poglavju bom na kratko opisal tehnike podatkovnega rudarjenja. Na seznamu ni vseh tehnik, osredotočil sem se na štiri napredne tehnike podatkovnega rudarjenja, za katere bi želel, da jih omogočajo tudi izbrani progami v praktičnem delu diplomske naloge. Razlog izbire naprednih tehnik podatkovnega rudarjenja se nahaja v dejstvu, da lahko s pomočjo le-teh dobimo drugačne in v večini primerov tudi bolj natančne rezultate, ter odkrijemo nekatere relacije, ki prej niso bile vidne.

## 3.1 METODA PODPORNIH VEKTORJEV

Metoda podpornih vektorjev je opisana kot metoda razvrščanja in linearne regresije. Izhodišče za nastanek SVM je množica učnih predmetov, za katere vemo, kateremu razredu pripadajo. Vsak predmet predstavimo z vektorjem v vektorskem prostoru. Naloga SVM je poiskati v tem več dimenzionalnem prostoru hiperravnino, ki ločuje predmete iz različnih razredov. Razdaljo vektorjev, ki ležijo najbližje hiperravnini, pri tem maksimiramo. To

široko prazno območje med razredi nam kasneje omogoča, da lahko čim bolj zanesljivo razvrščamo tudi predmete, ki niso povsem enaki učnim predmetom. (Cortes & Vapnik, 1995, str. 273-275) Na Sliki 4 je prikazana optimalna postavitev hiperravnine, kjer premica $H_1$ ne ločuje razredov, premica $H_2$ jih, ampak le z majhno razliko, $H_3$ pa ju loči optimalno.

*Slika 4: Optimalna postavitev hiperravnine*




Vir: *I. H. Witten & E. Frank, Data mining: practical machine learning tools and techniques, 2005, str. 216.*

Metoda podpornih vektorjev je danes zelo cenjena med strokovnjaki, saj daje podroben opis dobljenega modela. Dosti ljudi se za to metodo ne odloči, saj zahteva veliko predznanja in časa, za svoje delovanje pa potrebuje najnovejšo strojno opremo, saj procesi zahtevajo veliko procesorske moči. Še ena slabost SVM je, da jo je možno uporabiti v modelih z le dvema razredoma in je potrebno uporabiti posebne algoritme, ki pretvorijo več razredne modele v več modelov z dvema razredoma. Metoda podpornih vektorjev se uporablja na več področjih, vključno z digitalno prepoznavo pisave in govora ter prepoznavo objektov in likov (Han & Kamber, 2006, str. 337).

## 3.2 BAYESOV KLASIFIKATOR

Bayesov klasifikator se uporablja za izračun pogojne verjetnosti za vsak razred pri danih vrednostih atributov. Klasifikator, ki bo točno izračunal pogojno verjetnost razredov, je zelo težko izračunati, zato se poslužujemo vpeljave določenih predpostavk za izračun približkov verjetnosti. Naivni Bayesov klasifikator predpostavlja, da je vpliv atributov razreda neodvisen od drugih vrednosti razreda, kar omogoči, da učna množica zadošča za dobro oceno potrebnih verjetnosti za izračun verjetnosti končnega razreda. Splošna verzija naivnega Bayesovega klasifikatorja so Bayesove mreže, ki za razliko od naivnega klasifikatorja upoštevajo tudi odvisnost med atributi (Kantardzic, 2003, str. 147).

## 3.3 METODA BOOSTING

Metoda Boosting je algoritem za strojno učenje, ki sta ga vpeljala Shapire in Freund. Cilj metode je zmanjšati pristranskost in nadzorovano učenje in se navezuje na vprašanje Kearnsa iz leta 1988, ali lahko iz skupine šibkih »učencev« dobimo enega močnega »učenca«. Šibek »učenec« je definiran kot klasifikator, ki je le v majhni korelaciji z načinom klasifikacije. Obratno je močni »učenec« klasifikator, ki je zelo povezan z načinom klasifikacije (Freund & Schapire, 1996, str. 148-156). Da bi najbolje razumeli, kako metoda deluje, poglejmo primer. Predpostavimo, da smo bolni in imamo več simptomov bolezni. Namesto da se posvetujemo samo z enim zdravnikom, se posvetujemo z večimi, nato pa vsako diagnozo številčno ovrednotimo glede na točnost diagnoz, ki so jih ti zdravniki postavili v preteklosti. Končna diagnoza je kombinacija ovrednotenih diagnoz. Tako nekako deluje metoda boosting (Han & Kamber, 2006, str. 368).

## 3.4 K-MEANS

K-Means ali algoritem srednjih K vrednosti spada pod naloge analize skupin. Najboljše se ta algoritem obnese, ko imamo na voljo numerične vrednosti podatkov. Glede na vrednosti, ki jih ima vsaka spremenljivka, se vsak zapis prikaže na večdimenzionalnem prostoru, kjer se znotraj tega prostora ustvarjajo naravne skupine. Skupine definira majhna oddaljenost med podobnimi elementi skupine in velika oddaljenost med pripadniki različnih skupin. Da pa bi takšna kalkulacija oddaljenosti elementov imela smisel, je potrebno spremenljivke normalizirati. Proces se začne z izbiro centroida (središče skupine), ki je izbran naključno. V naslednjih korakih se središča pomikajo na točke v večdimenzionalnem prostoru, ki največkrat ne predstavljajo niti enega zapisa, temveč se izračunajo kot sredina skupin. Medtem ko centroidi menjajo svoj položaj, skozi proces iteracije pridemo do končnega modela. Včasih je rezultat odvisen od izbire začetnih centroidov, zato se postopek ponovi z različnimi začetnimi pogoji. (Tan, Steinbach & Kumar, 2006, str. 497)

Primer osnovnega algoritma srednjih K vrednosti:

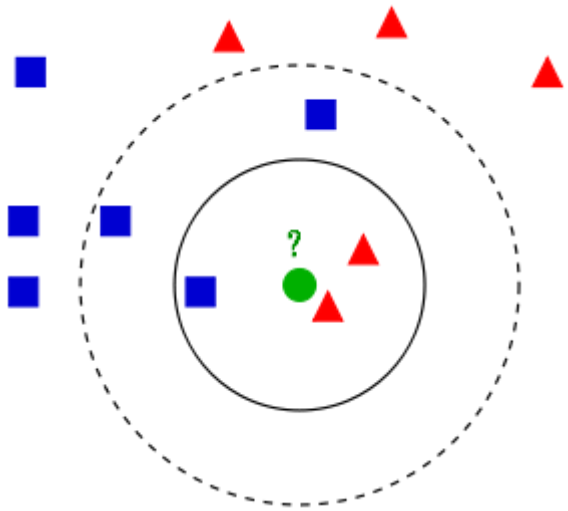
| |
|---|
| 1. Izbira K točk kot začetne centroide |
| 2. Ponovi operacijo |
| 3. Ustvarjanje K skupin in dodeljevanje točk k najbližjim centroidom |
| 4. Preračunavanje centroidov za vsako skupino |
| 5. Ponavljanje postopka, dokler se centroidi več ne spremenijo |

## 3.5 NAJBLIŽJI SOSED

Tehnika najbližjega soseda (angl. *Nearest Neighbour*) je metoda klasifikacije in je ena najstarejših tehnik podatkovnega rudarjenja. Tehnika je nastala že leta 1950, vendar se je začela bolj uporabljati šele po letu 1960, ko se je povečala procesorska moč računalniških sistemov. Princip tehnike temelji na učenju sorodnosti elementov in bližini zapisa v množici, kakor je prikazano na Sliki 5. Napoved spremenljivke pri izbrani opazovani enoti se nanaša na pretekle podatke in izmed podobnih izbere tisto, ki ji je po značilnosti najbližja (Han & Kamber, 2006, str. 348).

*Primer: Testni primerek sadja (zeleni krog) pripada bodisi prvemu razredu, tj. jabolkom (modri kvadrat) bodisi drugemu razredu, tj. hruškam(rdeči trikotniki). Če je k=3 (polna krožnica), potem zeleni krog pripada razredu rdečih trikotnikov, saj sta v krogu dva trikotnika in le eden kvadrat. V primeru k=5 (črtkana krožnica), pa bo zeleni krog pripadal razredu kvadratov, saj so znotraj krožnice trije in od teh le dva trikotnika.*

*Slika 5: Preprost primer prikaza klasifikacije kNN*




Vir: P.N. *Tan, M. Steinbach & V. Kumar, Introduction to Data Mining, 2006, str. 225*.

# 4 EVOLUCIJA ODPRTOKODNIH PROGRAMOV ZA PODATKOVNO RUDARJENJE

Zgodnji modeli in programi za strojno učenje iz začetka 80. let so bili večinoma osnovani na prvih operacijskih sistemih UNIX in DOS, kar je pomenilo, da so bili zagnani neposredno iz ukazne vrstice. Uporabnik je moral vnesti ime zagonske/vhodne datoteke in pa parameter za izvedbo algoritma. V tem času še ni bilo razvitih veliko rešitev in je bilo v skupnosti najbolj popularno orodje C 4.5, v katerega je bilo denimo možno vnesti razčlenjene zagonske datoteke za testiranje modela, njegova pomanjkljivost pa je bila, da ni zmogel postopka

vzorčne evalvacije algoritma. Večina takratnih rešitev za podatkovno rudarjenje je analizirala predvsem podatke in zbirke podatkov iz medicine, vključno s tistimi za odkrivanje in napovedi rakavih obolenj (Zupan & Demšar, 2008, str. 38). Napovedni modeli iz medicine so potrebovali (poleg teh analiz) tudi dodatne zmožnosti vnosa drugih parametrov, zato niso bili dokončno integrirani v zgodnje programske rešitve podatkovnega rudarjenja. Razvijalci so takrat večinoma programirali v programskem jeziku Perl (angl. *Practical Extraction and Report Language*), ki si je sintakso sposodil pri C-ju, Unixu in drugih, pri čemer so želeli uvesti posamezne postopke vzorčenja in jih nato izvesti za izdelavo napovednih modelov. Za primerjavo različnih tipov algoritmov je bilo treba modificirati podatke za vsak algoritem posebej, na koncu procesa pa dobljene besedilne datoteke tako razčleniti, da so bile uporabne za izračun vrednosti.

Zgoraj navedeno pomeni, da je bilo potrebno implementaciji postopkov nameniti veliko programiranja in obdelave besedilnih datotek, kar je od uporabnika zahtevalo veliko časa. Kot alternativo je več skupin razvijalcev ponudilo programske rešitve, ki bi bile zmožne obdelati različne tipe vhodnih datotek ter hkrati ponuditi dovolj dobro oceno rezultatov in poročanja. Rezultat sodelovanja skupin razvijalcev je bil MLC ++ (angl. *Machine Learning in C++*), rešitev, z vmesnikom z ukazno vrstico, ki je vključevala vse takratne standard tehnike analize programov za strojno učenje. Razvijalci niso bili navdušeni nad vmesniki z ukazno vrstico, omejeno obdelavo podatkov in prikazom le-teh, neatraktivno analizo s tekstovnim prikazom in dokaj slabo zmogljivostjo, zato jih je želja po razvoju gnala naprej. Želeli so ustvariti programsko rešitev, ki bi omogočala vizualizacijo (prikaz) podatkov in sposobnost enostavnega upravljanja programa.

Tako so kmalu nastale prve rešitve z grafičnim vmesnikom, podporo različnim operacijskim sistemom, možnostjo razširljivosti in bolj naprednim, večdimenzionalnim prikazom podatkov dobljenih analiz. Clementine (www.spss.com/clementine) je bil prvi izmed prihajajočih popularnih komercialnih rešitev za podatkovno rudarjenje, ki je uvedel uporabniški nadzor nad razvojem smeri analize z vključitvijo različnih tehnik podatkovnega rudarjenja v ločenih komponentah, ki jih je bilo mogoče potem združiti v celoto pri izdelavi določene vrste analize podatkov.

Večina današnjih programskih rešitev za podatkovno rudarjenje uporablja podobne pristope, saj so enostavni za uporabo in prilagodljivi, s tem pa tudi privlačni za uporabnike, ki niso računalniški strokovnjaki. Prilagodljivost in razširljivost programov za analizo se je oblikovala iz vidika zmožnosti uporabe izvorne kode: za razvoj ali za razširitev algoritmov. Za primer vzemimo program WEKA, popularno programsko rešitev, ki nam, v kolikor se spoznamo na programiranje v Javi in imamo zadostno znanje zgradbe programa, ponuja obilo obstoječih funkcij in razredov, ki jih je možno enostavno razširiti.

Drugačen pristop so recimo vzeli razvijalci drugih programskih rešitev, tudi razvijalci programa R, danes najbolj poznanega in razširjenega odprtokodnega programa za statistiko in analizo. Namesto da bi razširili funkcije programa s pomočjo kode, napisane v programskem jeziku C, ki je obenem tudi programski jezik jedra programa, so se raje osredotočili na razširljivost v obliki na novo spisanega skriptnega jezika, ki skozi vmesnik poganja enake ukaze kot programski jezik C (Zupan & Demšar, 2008, str. 40). Večina razširitev je vstavljena v skripte, ki ne zahtevajo prevajalnika v drug programski jezik ali uporabe drugačnega okolja za razvoj.

Prednost takšnega razvoja programskih rešitev je hitrost (računske funkcije so vključene v enostavno napisane programske jezike in so nam na voljo preko skriptnega jezika), prilagodljivost (skripte lahko vsebujejo funkcije originalnega programa in funkcije skriptnega jezika) in pa razširljivost (nadgradnja programa za podatkovno rudarjenje skozi uporabo drugih rešitev, ki komunicirajo z določenim skriptnim jezikom). Vse našteto je morda težko razumljivo in zahtevno za uporabo za tiste z manj računalniškega znanja oz. za tiste, ki se bolje znajdejo v programih z grafičnimi uporabniškimi vmesniki. Skriptni jezik in uporaba skript pa sta kljub temu danes bistvenega pomena za razvoj novih tehnološki rešitev in sta ključ uspeha številnih programskih rešitev.

## 4.1 PODATKOVNO RUDARJENJE Z ODPRTOKODNIMI REŠITVAMI

V primerjavi s komercialnimi programskimi rešitvami imajo odprtokodni programi kar nekaj slabosti. Največkrat za razvojem programov stoji skupina zanesenjakov, ki v rešitve vključuje aktualne algoritme za analizo podatkov, ki niso najbolj stabilni. Prav tako programi s »trgovinskih polic« pridejo v paketu s podporo upravljanja različnih podatkovnih baz, medtem ko imajo odprtokodni programi na voljo vtičnike, ki uporabniku omogočajo poizvedbe v standardnih podatkovnih bazah. Postopek je večinoma dolgotrajen, zato se mnogi raje odločijo za nakup komercialne rešitve. Takšne pomanjkljivosti oziroma slabosti pa so zanemarljive v primerjavi s prednostmi, ki jih ponujajo odprtokodni programi za podatkovno rudarjenje.

Odprtokodne programske rešitve obsegajo računalniške programe, ki uporabniku dovoljujejo uporabo, razmnoževanje, razširjanje, razumevanje, spreminjanje in izboljševanje programa. Prednost uporabe takšnih rešitev je večja varnost programov in manjša ranljivost za viruse in varnostne luknje, verjetno najpomembnejše pa je to, da ni stroškov pri nakupu programske opreme (Free Software Foundation: *The GNU Project*, 2013). Ponujajo najnovejše eksperimentalne tehnike, algoritme in grafični prikaz, vključujejo prototipne funkcije in so lahko rešitev tam, kjer je komercialno orodje brez odgovora. Za tem največkrat stoji velika in raznolika skupnost, ki s temi programi dela na dnevni ravni. Nabor števila algoritmov je pri njih velik, kar nakazuje na zmožnost reševanja velikega števila problemov, s katerimi se srečujemo. Poslovni analitiki najdejo v teh rešitvah veliko uporabnost predvsem v primeru

razširljivosti programov, kot je na primer prost dostop do izvorne kode in njenih komponent ter možnost spajanja programske opreme z drugimi programskimi rešitvami za analizo podatkov, s sodobnimi skriptnimi jeziki pa pridobijo podporo pri takšni ad-hoc integraciji. (Zupan & Demšar, 2008, str. 40) Navodila in dokumentacija pri odprtokodnih rešitvah seveda niso tako dodelana kot pri komercialnih, vendar so prosto dostopna na internetu v več oblikah, se stalno nadgrajujejo in imajo v večini primerov dodane vaje in rešitve, ki so jih napisali uporabniki in zanesenjaki. Na koncu je tu še podpora uporabniku, ki je drugačna, kot jo poznamo iz sveta komercialne programske opreme.

Uporabniki komercialnih rešitev so odvisni od službe za pomoč uporabnikom, medtem ko pri odprtokodnih programski rešitvah pomoč temelji na principu vzajemnosti, kjer si vsi uporabniki pomagajo med seboj. Takšno sodelovanje je največkrat vidno pri programskih rešitvah z veliko aktivnimi uporabniki programa v obliki spletnih forumov, klepetalnic, listah elektronske pošte, spremljanju napak in hroščev v obliki obveščanja in povratnih informacijah razvijalcem, ter konec koncev tudi pri podpori novim uporabnikom.

Z vključitvijo vseh teh prednosti v uporabniške vmesnike in orodij za poročanje, implementacijo najnovejših/aktualnih postopkov analize podatkov in večanjem števila uporabnikov lahko zaključimo, da odprtokodna orodja za podatkovno rudarjenje postajajo uporabna alternativa in zamenjava komercialnim programskim rešitvam.

## 4.2 POMEMBNE LASTNOSTI ODPRTOKODNIH PROGRAMOV ZA PODATKOVNO RUDARJENJE

V zadnjih desetletjih so odprtokodni izdelki, kot na primer GNU/Linux, Apache, BSD (angl. *Berekley Software Development*), mySQL in OpenOffice, poželi velik uspeh in s tem jasno demonstrirali svoje sposobnosti, ki so v rangu komercialnih rešitev ali pa še boljše. Finančno so bolj dostopne, s čimer omogočajo podjetjem, da investicije namenjene v nakup programske opreme vložijo v razvoj in izobraževanje človeškega kapitala (Chen, Yunming, Williams & Xu, 2007, str. 2).

Odprtokodni programi za podatkovno rudarjenje se med seboj razlikujejo po uporabi in obliki, prikazu podatkov na zaslonu, številnih funkcijah, implementaciji, načinu vnosa podatkov itd. Skoraj vsa so razširljiva, kar pomaga predvsem programerjem (razvoj komponent po meri) in tistim, ki analizirajo podatke, manj pa običajnemu uporabniku. Kljub temu, da so orodja razvita za namen podatkovnega rudarjenja, se med seboj razlikujejo v različnih pogledih. Večina jih je na voljo na internetu v lično zapakiranih paketih, vključujejo pa velik nabor komponent za analizo podatkov.

Da bi lažje razumeli značilnosti in razlike med njimi ter jih na koncu lahko ovrednotili, avtorji izpostavljajo pomembne lastnosti, ki naj bi jih imela vsaka programska rešitev za podatkovno rudarjenje (Chen, Yunming, Williams & Xu, 2007, str. 4-5), posledično tudi odprtokodna orodja:

- **Zmožnost dostopa do različnih podatkovnih virov**

Tukaj so vključeni vsi podatki iz podatkovnih baz, podatkovnih skladišč in drugih virov. Sistem mora omogočiti enostavno uvažanje različnih tipov podatkov.

- **Zmožnost dobre predpriprave podatkov**

Priprava podatkov zahteva dobršen del časa v procesu, ki bi ga lahko porabili v procesu podatkovnega rudarjenja. Dobra in kvalitetna predpriprava podatkov je lahko ključnega pomena za končni rezultat procesa. Programska rešitev naj bi torej ponujala hitro, učinkovito in enostavno pripravo podatkov, vključno z možnostjo poizvedbe, izbiro primera, podskupine itn.

- **Integracija različnih tehnik podatkovnega rudarjenja**

Različni problemi odpirajo pot različnim tehnikam in rešitvam. V osnovi ne obstaja najboljša tehnika podatkovnega rudarjenja, ki bi ustrezala vsem problemom, na katere naletimo tekom procesa podatkovnega rudarjenja. Programska rešitev združuje večino tehnik od predpriprave do algoritmov uporabljenih v procesu, hkrati pa nam na enostaven način omogoča dostop do tehnik, ne glede na kompleksnost problema, s katerim se srečujemo. V to kategorijo sodijo razvrstitvena pravila in drevesa za odločanje, Support Vector Machines, Bayesov klasifikator itd.

- **Zmožnost delovanja z velikimi nabori podatkov**

Komercialne različice orodij za podatkovno rudarjenje, kot na primer SAS Enterprise, so zmožne upravljati z velikim naborom podatkov in podatkovnih baz, zato je pomembno, da takšne lastnosti posedujejo tudi odprtokodne rešitve.

- **Dober grafični prikaz in predstavitev podatkov in modelov**

Ličen grafični vmesnik, enostavno izbiranje tehnik in algoritmov ter raznolik prikaz rezultatov so ključnega pomena, saj omogočajo začetnikom, naprednim uporabnikom in strokovnjakom, da se v aplikaciji znajdejo in jo enostavno uporabljajo. Dandanes nihče ne želi vnašati ukazov v ukazno vrstico.

- **Razširljivost programske opreme**

S prihodom novih tehnik in algoritmov v podatkovnem rudarjenju je pomembno, da programska rešitev ponuja ogrodje za nadgrajevanje s sodobnimi tehnikami podatkovnega

rudarjenja. Pomembno je, da rešitev ponudi dovolj enostavno arhitekturo, izvorno kodo in enostavno implementacijo zunanjih gradnikov, vtičnikov.

- **Zmožnost dobrega poročanja in grafičnega prikaza rezultatov**

Možnost uporabe orodja, ki bo shranilo trenutne rezultate analize in z njo povezana poročila, vključujoč komentarje v pregledni obliki in možnost priklica pripadajoče analize za nadaljnje raziskovanje.

- **Interoperabilnost z drugimi sistemi**

Interoperabilnost je definirana kot zmožnost informacijskih sistemov in poslovnih procesov, ki jih ti sistemi podpirajo, da izmenjujejo podatke, informacije, modele. Dobra rešitev bo ponudila uporabo osnovnih standardov, kot sta CWM (angl. *Common Warehouse Model)* in PMML (angl. *Predictive Model Markup Language*).

- **Aktivna razvijalska skupnost**

Aktivna skupnost skrbi, da programske rešitve delujejo brezhibno in omogoča vsem uporabnikom vzdrževanje in posodobitve, hkrati pa tudi pomoč in podporo pri težavah in razvoju novih razširitev.

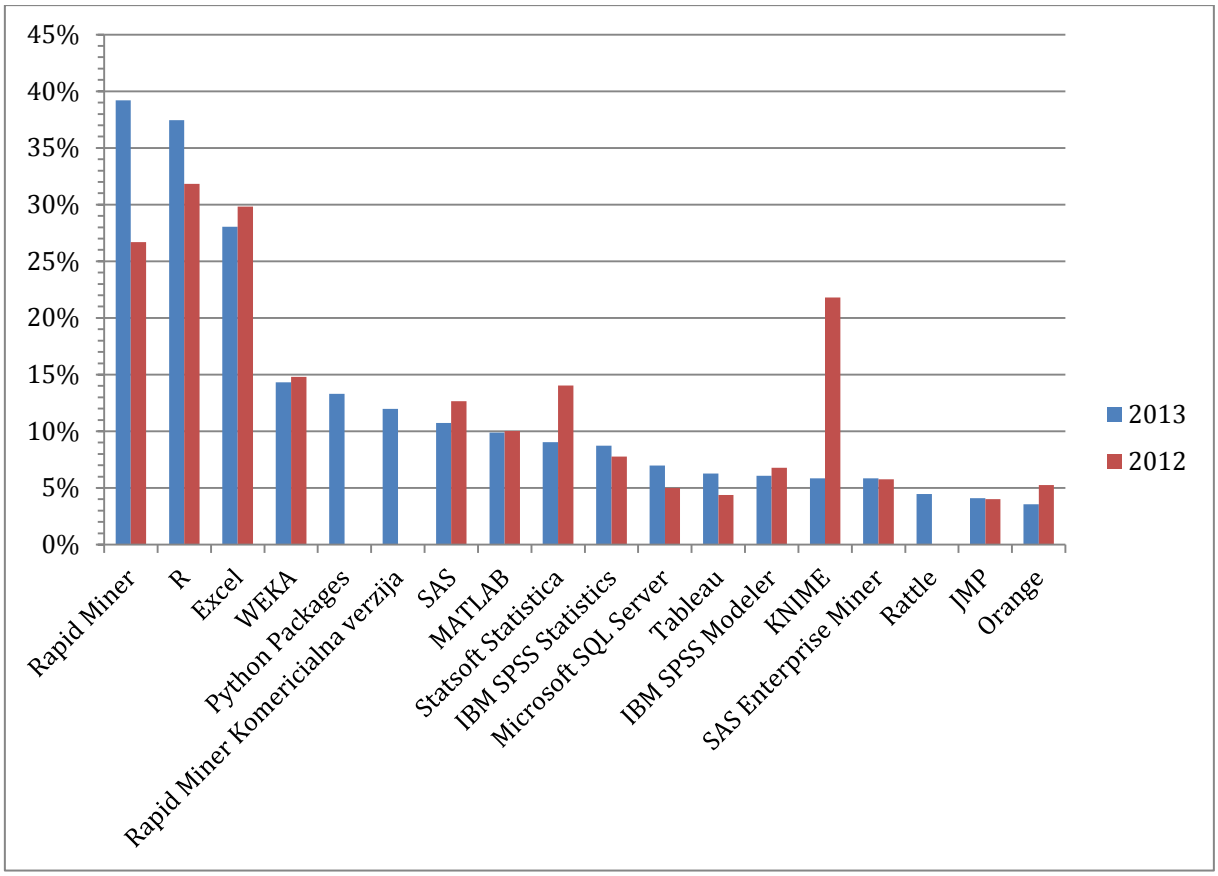
# 5 ODPRTOKODNA ORODJA ZA PODATKOVNO RUDARJNEJE

V nadaljevanju je predstavljenih in analiziranih šest odprtokodnih programskih rešitev za podatkovno rudarjenje, ki so na voljo na internetu. Vse rešitve je mogoče brez posebnega truda shraniti na trdi disk, saj so na voljo v brezplačni verziji. Na seznamu so največje in najbolj priljubljene rešitve kot npr. WEKA, RapidMiner in R. Seznam takšnih programov je zelo dolg zaradi velikega števila skupnosti, ki ustvarjajo in programirajo na dnevni ravni, a smo s pomočjo spletne strani KDnuggets, ki jo ustvarja skupnost za podatkovno rudarjenje, izbrali le najbolj uporabljene spletne rešitve v preteklem letu. Na Sliki 6 je anketa spletne strani KDnuggets, kjer so uporabnike povprašali, katere programske rešitve na področju analize podatkov in podatkovnega rudarjenja so uporabili v preteklem letu. V anketi je sodelovalo več kot 1800 uporabnikov. Rezultati so presenetljivo pokazali, da sta največkrat uporabljeni rešitvi RapidMiner in pa R (KDnuggets, 2013) izmed prvih petih izbir pa je kar štiri odprtokodnih, kar nakazuje porast odprtokodnih rešitev za podatkovno rudarjenje.

Ločnica med odprtokodnimi rešitvami in komercialnimi se vsako leto zmanjšuje, saj proizvajalca rešitev, kot sta KNIME in RapidMiner, ponujata tudi komercialne rešitve, ki pa se ne razlikujejo preveč od odprtokodne rešitve. Kot zanimivost lahko navedemo, da je 29 % uporabnikov uporabili samo komercialne programske rešitve in ne brezplačnih, 30 % uporabnikov pa samo brezplačne. Nekatere rešitve imajo opazen napredek, deloma zaradi razvoja in deloma zaradi namere prodajalcev, da spodbudijo uporabnike. Visoko na lestvici so tudi razširitve s pomočjo skriptnega jezika Phyton, ki je vse bolj dostopen običajnim

uporabnikom, saj programiranje v temu programskemu jeziku ne zahteva pretiranega predznanja.

*Slika 6: Največkrat uporabljene programske rešitve za podatkovno rudarjenje v letih 2013 in 2012*

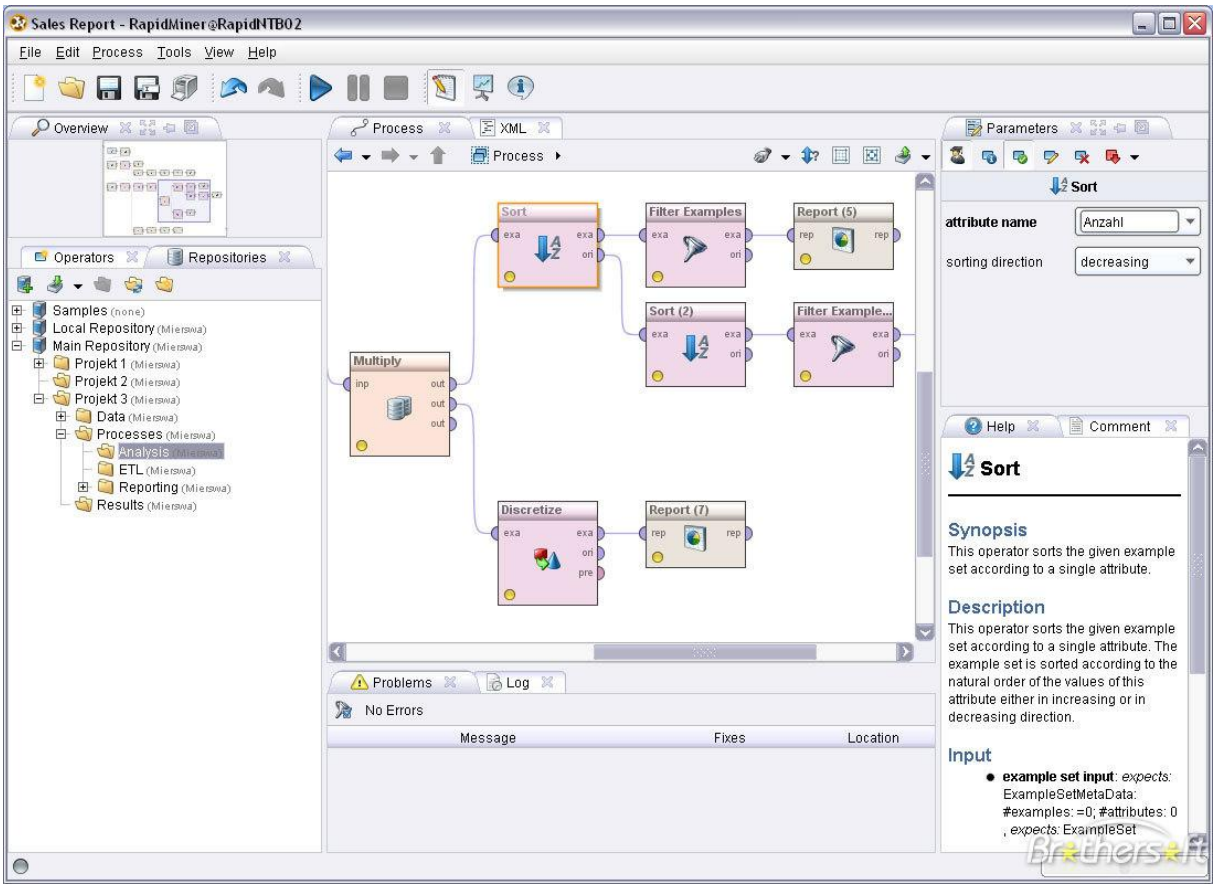



*Vir : KDnuggets, Annual Software Poll, 2013.*

## 5.1 RAPIDMINER

Vodilna programska rešitev na področju podatkovnega rudarjenja po mnenju uporabnikov širom sveta (KDnuggets, 2013), prej imenovana YALE (angl. *Yet Another Learning Enviroment*), je na voljo kot samostojna rešitev za analizo podatkov, lahko pa jo tudi integriramo v lastne rešitve. Že ko pridemo na spletno stran programa, v zgornjem kotu zasledimo moto »Open source software for big data analytics – no programming required«, kar v prevodu pomeni »Odprtokodna programska rešitev za velike analize – predznanje v programiranju ni potrebno«. S tem nam dajo nemški razvijalci jasno vedeti, da je program namenjen vsem uporabnikom in ne le računalniškim strokovnjakom. Tako kot orodje WEKA je program RapidMiner napisan v programskem jeziku JAVA. Grafični vmesnik je možno nastaviti na dva pogleda, in sicer kot oblikovni pogled (kjer podatke in funkcije vnašamo s tehniko »povleci in spusti«, ter jih nato medsebojno povezujemo) in pogled rezultatov analiz (omogoča prikaz analiz in podatkov izbranih v oblikovnem pogledu).

*Slika 7: Zaslonska maska programa RapidMiner*



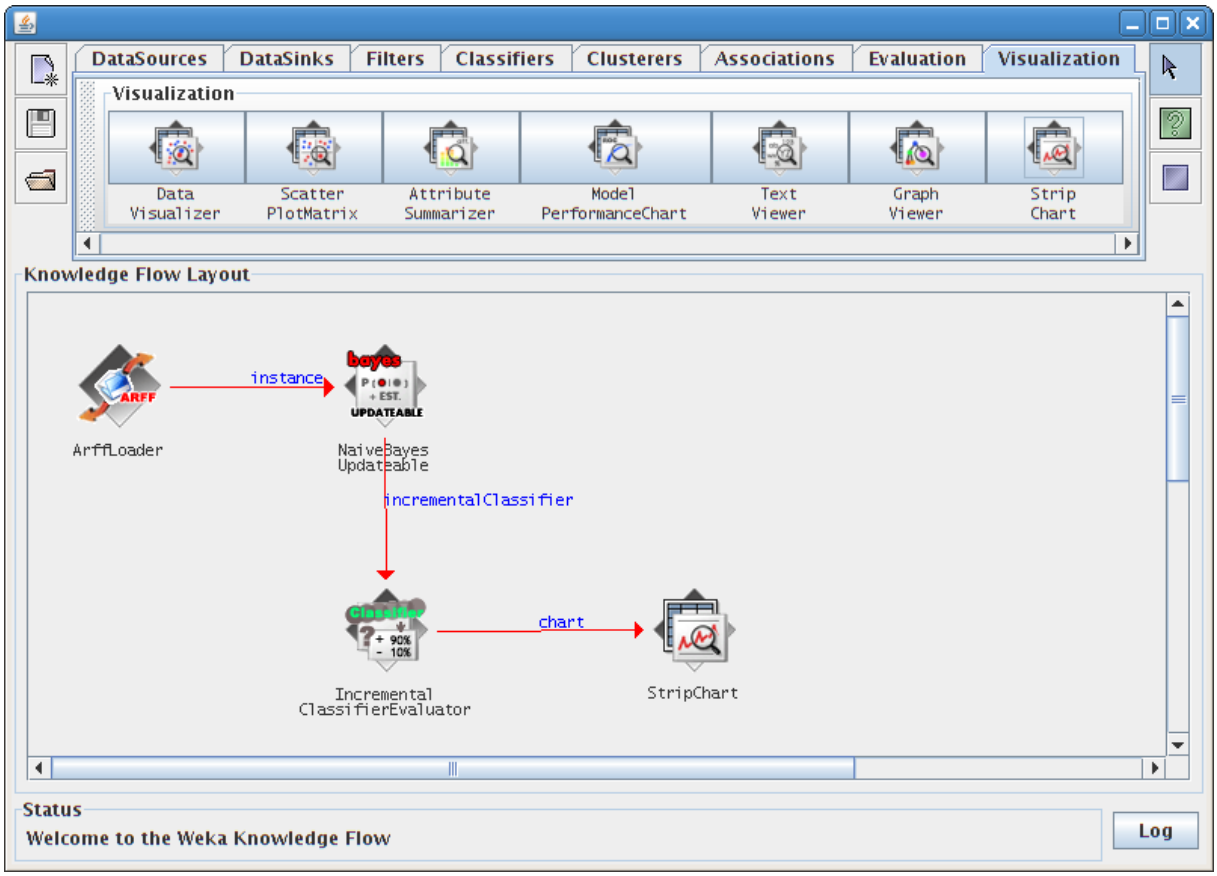

Med delom lahko pregledujemo in urejamo procese, ki so zapisani v obliki XML, gradimo pa s pomočjo operatorjev, ki se nahajajo na levi strani oblikovnega pogleda. Operatorjev je preko 500 (operatorji za strojno učenje, operatorji za uvoz in izvoz podatkov, operatorji za nadzor procesa itd.), kar je lahko tudi slabost, saj nekateri med njimi ne naredijo bistvene razlike, poleg tega pa bi se začetniki lažje znašli ob manjšem številu le-teh, poznavalci pa so zmožni sestaviti precej bolj kompleksne procese. Rapidminer je zmožen predstaviti rezultate na več načinov, najbolj pa izstopa grafični prikaz, saj ponuja veliko različnih grafov. V primerjavi z drugimi okolji programskih rešitev je vmesnik RapidMinerja bolj intuitiven, kar kaže strma krivulja učenja pri uporabnikih z omejenim znanjem računalništva in programiranja.

## 5.2 WEKA

WEKA je aplikacija, zasnovana v programskem jeziku JAVA, (Waikato Environment for Knowledge Analysis), razvita pa na Univerzi v Waikitu. Podpira strojno učenje, kakor tudi vse mogoče funkcije in algoritme podatkovnega rudarjenja. Aplikacija je sestavljena iz štirih uporabniških vmesnikov: Explorer (raziskovalec z možnostjo generiranja podatkov, vnosa podatkov in podatkovnih baz), Experimenter (za izvajanje eksperimentov, analizo

algoritmov in vodenje statističnih testov), Knowledge Flow (za postopno učenje in grafično sestavo algoritmov, ki jih izbiramo iz vmesnika) ter Simple CGI (enostavna ukazna vrstica, ki omogoča izvedbo WEKA ukazov, primerna za napredne in izkušene uporabnike).

*Slika 8: Zaslonska maska WEKA Knowledge Flow*




Program predpostavlja, da so podatki shranjeni v eni sami besedilni datoteki, kjer je vsaka relacija zapisana z določenim številom lastnosti. Na primer:

- *@relation weather*
- *@attribute outlook {sunny, overcast, rainy}*
- *@attribute temperature real*
- *@attribute humidity real*
- *@attribute windy {TRUE, FALSE}*
- *@attribute play {yes, no}*

Vsi podatki so torej zapisani v eni sami besedilni datoteki v formatu ARFF (Attribute Relationship File Format), ki je sestavljena iz glave in podatkovnega dela, kjer glava določa ime relacije, v našem primeru so to vrstice označene z @. Spodaj pa sledijo podatki kot npr.:

- *sunny,85,85,FALSE,no*
- *sunny,80,90,TRUE,no*

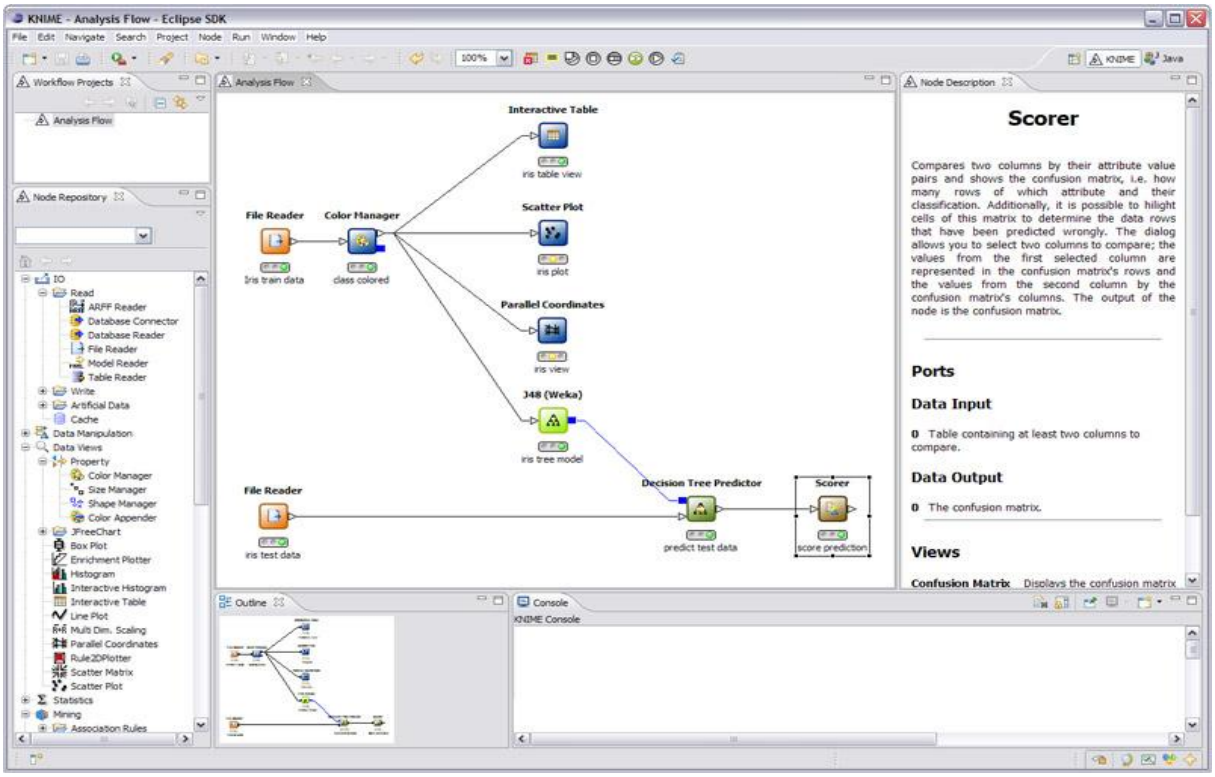
- *overcast,83,86,FALSE,yes*
- *rainy,70,96,FALSE,yes*

Za začetnike je dovolj le uporaba vmesnika Knowledge Flow, saj ponuja precej enostavne funkcije podatkovnega rudarjenja. V primerjavi s programsko rešitvijo R se WEKA slabše obnese v prvinah statistične analize, njene prednosti pa so v močnih algoritmih za strojno učenje.

## 5.3 KNIME

Konstanz Info Miner ali KNIME je tako kot RapidMiner delo nemških programerjev, razvit na Univerzi v Konstanzu. Orodje je osnovano na okolju IBM Eclipse – to je okolje za razvoj programskih rešitev in vtičnikov, napisano v programskem jeziku JAVA, zaradi česar je program (kot tudi WEKA in RapidMiner) lahko razširljiv in z uporabo vtičnikov pridobi dodatno funkcionalnost. Osnovni zaslon grafičnega vmesnika je razdeljen na več delov, največji del pa tvori urejevalnik toka podatkov. Uporabnik tako s tehniko »povleci in spusti« vleče želene module iz stolpca gradnikov na levi v urejevalnik toka podatkov in s tem tvori proces. Vhodi in izhodi modulov so barvno označeni, tako da nam je prihranjeno ugibanje o tem, kateri del se povezuje z drugim. Pod vsakim modulom je majhen semafor, kjer je predstavljeno delovanje tega modula. Če je semafor rdeč, potem modul ni pravilno spojen in ni pripravljen, pri rumeni luči imamo sicer stvar nastavljeno, vendar se zaradi določene težave še ni izvedla, zelena pa pomeni, da je zadeva pripravljena.

*Slika 9: Zaslonska maska programa KNIME*

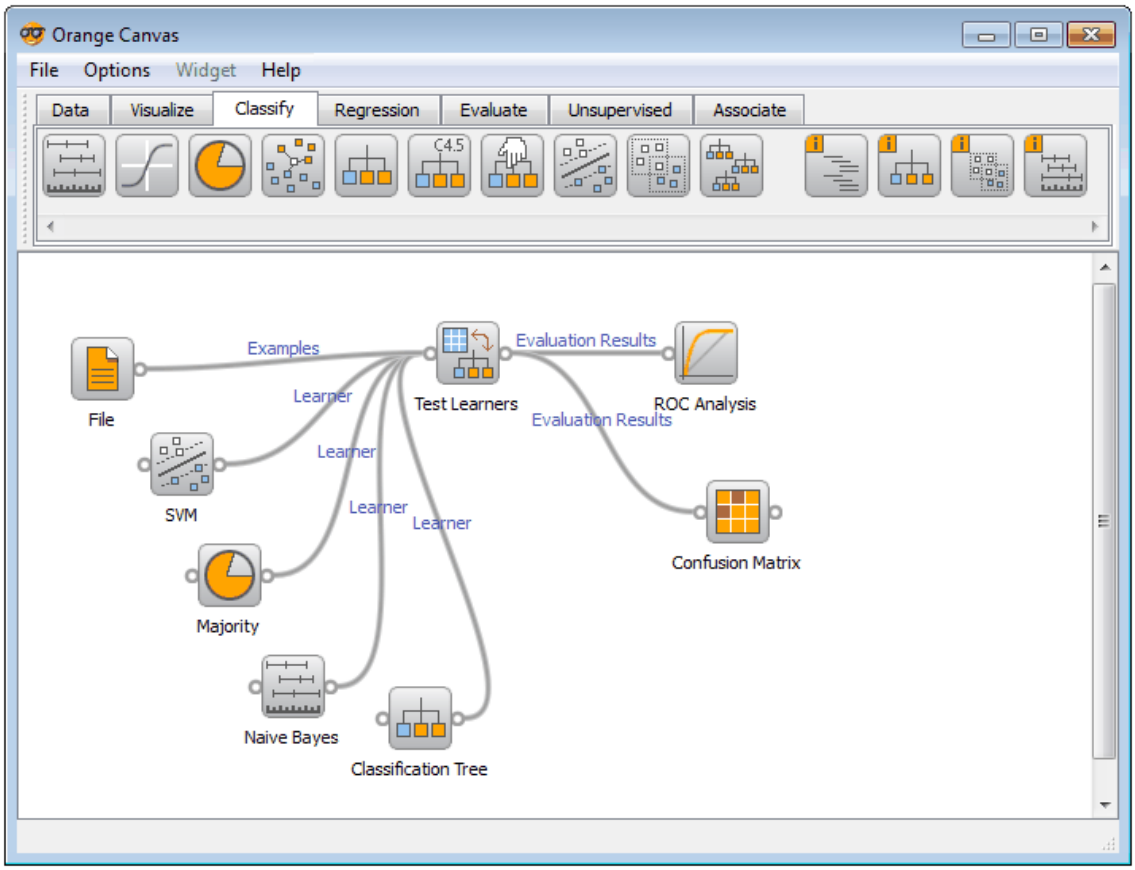


V osnovni verziji tako dobimo prek 100 različnih tipov modulov, vsak od njih pa ima določeno le eno funkcijo, kot so npr. branje podatkov, filtriranje podatkov, modeliranje, dopolnjevanje manjkajoče vrednosti, izvajanje skripte itd. Orodje KNIME je mogoče razširiti z različnimi moduli, ki omogočajo integracijo algoritmov iz programov WEKA ter R in so na voljo na njihovi spletni strani, lahko pa modul sprogramiramo tudi sami preko čarovnika, ki se nahaja v aplikaciji. KNIME je v zadnjem letu izgubil dosti uporabnikov, glede na lansko leto skoraj 15 odstotkov, razloge pa gre pripisati napredku in enostavnejši uporabi drugih odprtokodnih rešitev, kot sta R in RapidMiner.

## 5.4 ORANGE

Orange je programska rešitev za strojno učenje in podatkovno rudarjenje, narejena po podobnih načelih kot WEKA in KNIME. Kot zanimivost lahko omenimo, da je nastala v Sloveniji, točneje v Laboratoriju za umetno inteligenco na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Delo se izvaja v grafičnem okolju imenovano Orange Canvas, kjer na platno (ang. *Canvas*) »slikamo« s pomočjo grafičnih gradnikov (ang. *Widgets*), ki jih izbiramo v orodni vrstici.

*Slika 10: Zaslonska maska programa Orange*



Jedro programa je napisano v programskem jeziku C++, gradnike pa je mogoče razvijati in implementirati s pomočjo uporabe skript in skriptnega vmesnika, zasnovanega v programskem jeziku Phyton.

*Slika 11: Primer prikaza vnosa podatkov s pomočjo skriptnega jezika*

```
1   import orange
2   data = orange.ExampleTable("lenses")
3   print "Attributes:",
4   for i in data.domain.attributes:
5       print i.name,
6   print
7   print "Class:", data.domain.classVar.name
8
9   print "First 5 data items:"
10  for i in range(5):
11     print data[i]
```

Vsak od gradnikov je namenjen za izvedbo enega procesa, seveda pa jih tisti z znanjem programiranja skriptnega jezika lahko po želji nadgradijo ali spremenijo. Zaenkrat je na njihovi spletni strani (na voljo več kot sto gradnikov, ki pokrivajo večino statističnih analiz, vmes pa so na voljo tudi specializirani vtičniki za področje bioinformatike. Poleg prijaznosti in enostavnosti uporabe aplikacije Orange nam orodje ponuja izredno veliko grafičnih prikazov: od diagrama razpršitve, dreves, vročinskih map nevronskih mrež, linearnih projekcij itd. Slabi strani aplikacije sta statistična plat (saj ne ponuja naprednih funkcij ter gradnikov za statistično analizo) in omejenost pri izvažanju grafičnih prikazov podatkov in modelov.

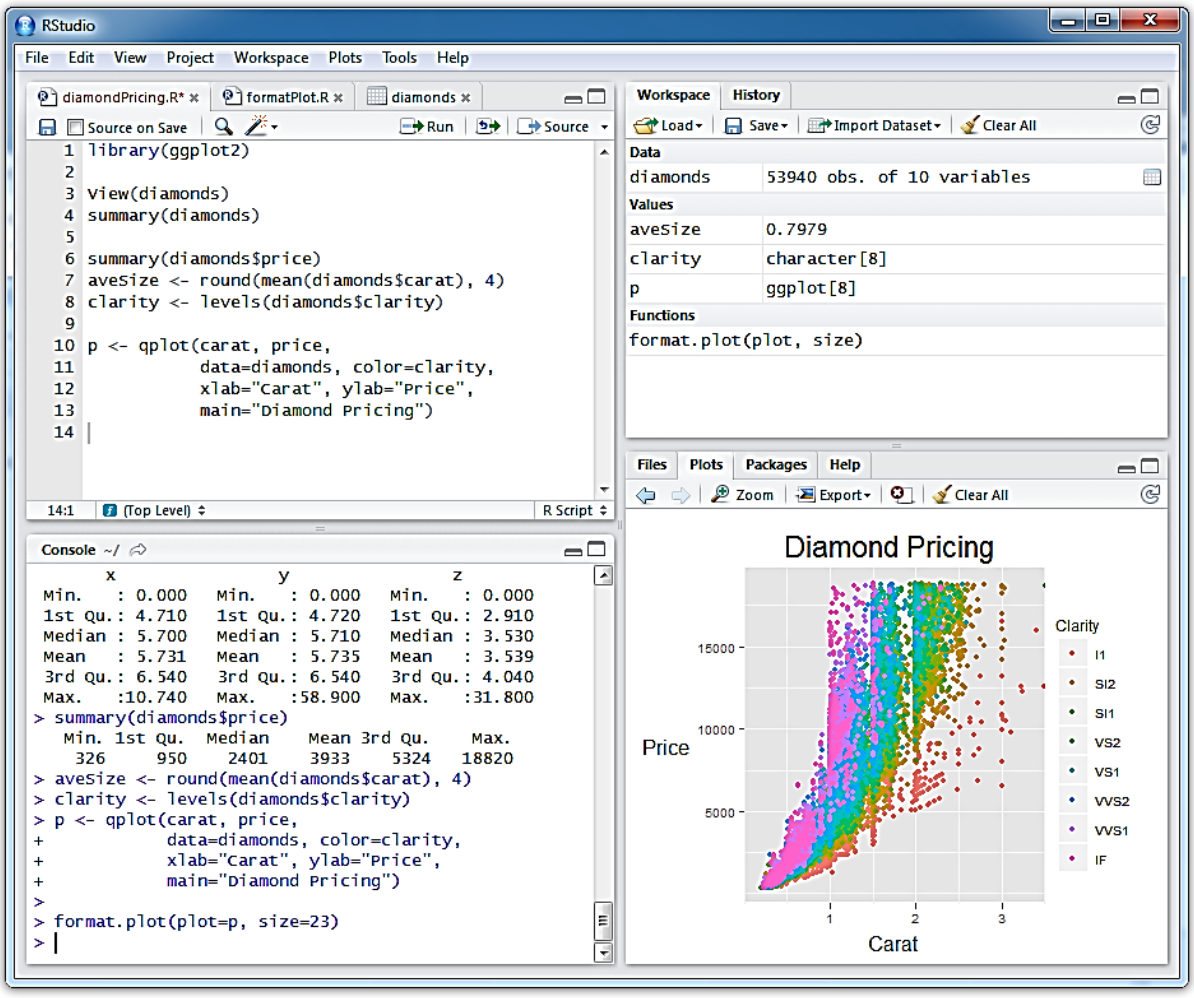
## 5.5 R

R je programski jezik, kakor tudi okolje za statistično obdelavo podatkov in podatkovno rudarjenje. Je najbolj pogost med razvijalci orodij za statistično analizo in zelo popularen pri uporabnikih zaradi svoje enostavnosti. Primer uporabe programskega jezika R je razvidna sintaksa (skladnja) jezika:

```
> x <- c(1,2,3,4,5,6)        # Ustvarimo skupino števil (vector))
> y <- x^2                   # Kvadriranje elementov x
> print(y)                   # Natisni (vector) y
[1]  1  4  9 16 25 36
> mean(y)                    # Izračunaj povprečje (aritmetična sredina) iz (vector) y;
rezultat je skalar
[1] 15.16667
> var(y)                     # Izračunaj standardni odklon
[1] 178.9667
```

Z rdečo barvo so označena števila oz. rezultati, ki jih dobimo z izvajanjem funkcij, te pa so označene z modro barvo. Z zeleno so označeni komentarji in nimajo vpliva pri izvedbi procesa. Orodje R torej vključuje zelo širok nabor statističnih funkcij, napovednega modeliranja in grafičnega prikaza. Aplikacija je razširljiva z več kot 4000 vtičniki, ki so na voljo na njihovi spletni strani, obsegajo pa skoraj vsa mogoča področja od športa, medicine, statistike, informatike, ekonomije itd. Vmesnik je narejen iz ukazne vrstice, kamor vpisujemo ukaze. Jasen postopek, kako do rezultata, je za marsikoga prednost, za večino brez izkušenj v programiranju pa ta lastnost ne pomeni veliko. Vendar pa je za takšne uporabnike na voljo veliko razširitev, ki s pomočjo grafičnega vmesnika naredijo program prijazen in enostavnejši za uporabo. Primer takšne razširitve je R Studio grafični vmesnik s funkcionalnostjo programskega jezik R. Kot samostojni program najdemo tudi aplikacijo Rattle, ki kot razširljiva zadeva programa R omogoča dostop do večine funkcij podatkovne analize in modeliranja.

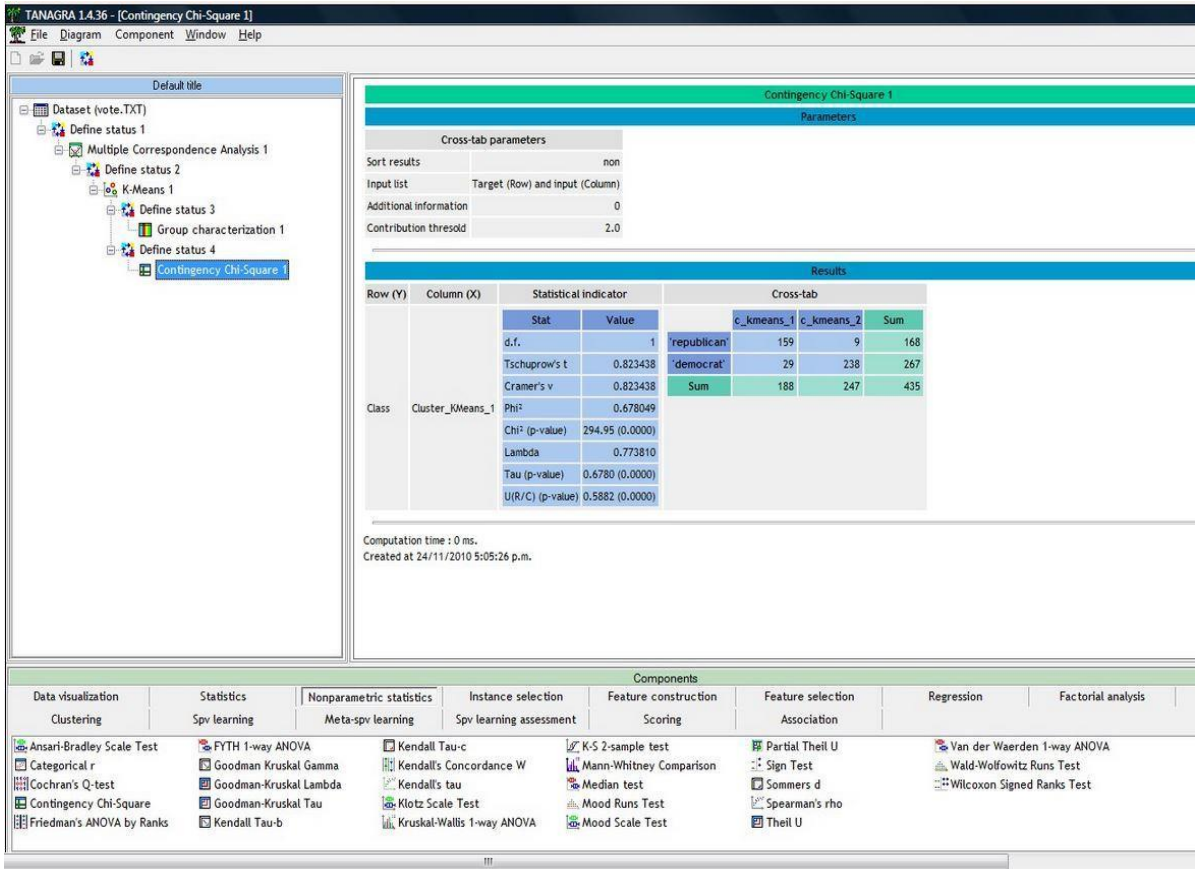
*Slika 12: Zaslonska maska programa R Studio*



## 5.6 TANAGRA

Tanagra je program za podatkovno rudarjenje razvit na Univerzi v Lyonu. Arhitektura programa je napisana v jeziku C++, kar omogoča razvoj razširitev, vendar je potrebno znanje programiranja, kar je za začetnika zahtevno opravilo. Grafični vmesnik programa je razdeljen na tri dele, in sicer imamo na levi strani delotok, kamor se razvrščajo funkcije ena na drugo, spodaj so komponente za izvajanje procesov, na sredini pa se nam prikazujejo rezultati oz. ustvarjajo forme za vnos podatkov. Prednost takšnega prikaza je v enostavnem povezovanju operacij. Vmesnik je podobno kot pri RapidMinerju in KNIME narejen tako, da enostavno s tehniko »vleci in spusti« vlečemo funkcije iz polja komponent v diagram toka, ki nam kaže nabor procesov za izvedbo. Tanagra ima zelo močno podporo statističnim analizam, enako zanimiv pa je tudi nabor ostalih lastnosti in funkcij, ki jih ponuja aplikacija. Predstavitev modelov za strojno učenje je večinoma negrafična, ampak za razliko od WEKA in Orange vključuje več statističnih prvin in meritev. Med slabše lastnosti štejemo zmožnost delovanja le na enem operacijskem sistemu (Windows) in slab grafični prikaz podatkov in modelov. Podatke analiz komponent sicer TANAGRA prikaže v elegantno oblikovani HTML datoteki.

*Slika 13: Zaslonska maska programa TANAGRA*



# 6 PRIMERJALNA ANALIZA ORODIJ ZA PODATKOVNO RUDARJENJE

Po namestitvi, pregledu in uporabi šestih programskih rešitev za podatkovno rudarjenje pridemo do dela diplomske naloge, kjer bom ovrednotil programe, ki smo jih uporabljali. S pomočjo podobnih analiz, objavljenih na internetu (*Data mining tools comparison - Summary*, Wei, 2008, *A Survey of Open Source Data Mining Systems*, Chen, Yunming Williams & Xu, 2007 in *Open Source Data Mining* Software Evaluation, Informatics Research and Developement Unit, 2010) sem izbral deset značilnosti s štirih različnih vidikov in jih tudi ovrednotil z utežmi. Uteži so bile izbrane uravnoteženo, in sicer glede na podobne raziskave z interneta (Chen, Yunming, Williams & Xu, 2007, str. 9) in iz mojega stališča do uporabe programskih rešitev v praksi. Numerično ovrednotenje vsake uteži je subjektivne narave, saj so bile izbrane za potrebe analize, ki je temeljila predvsem na zmogljivosti programskih rešitev v namen poslovnega odločanja. Nekdo, ki bi recimo potreboval rezultate analize na kakšnem drugem področju, bi verjetno uteži ovrednotil po svojem prepričanju oz. izbral tiste vidike, ki so zanj najpomembnejši. V Tabeli 1 je prikazana porazdelitev uteži glede na štiri vidike.

*Tabela 1: Porazdelitev uteži med štiri vidike ocenjevanja programskih rešitev*

| Vidik | Lastnosti | Uteži (izraženo v %) |
|---|---|---|
| ***Osnovne lastnosti*** | Aktivnost | 7,0 |
| | Podpora operacijskim sistemom | 5,0 |
| | Podpora uporabnikom | 9,0 |
| | Kompleksnost namestitve | 3,0 |
| ***Podatkovni viri*** | Možnost vnosa različnih vrst datotek | 9,0 |
| | Možnost uporabe velikih podatkovnih setov | 10,0 |
| ***Funkcionalnost*** | Predpriprava podatkov | 13,0 |
| | Funkcije podatkovnega rudarjenja | 13,0 |
| | Prikaz podatkov in modelov | 10,0 |
| ***Uporabnost*** | Človeška interakcija | 5,0 |
| | Interoperabilnost | 5,0 |
| | Razširljivost | 10,0 |

Po pripravi uteži sem programe namestil na svoj računalnik in jih skladno z zgoraj navedenimi lastnosti tudi preizkusil s pomočjo testnega podatkovnega seta s strani http://www.knime.org/downloads/datasets. Pri analizi sem šest programskih rešitev za podatkovno rudarjenje ocenjeval glede na štiri različne vidike. To so: osnovne lastnosti, podatkovni viri, funkcionalnost in uporabnost. Vse izmed njih sem razčlenil na nekaj podlastnosti, tako da sem lahko čim bolj porazdelil uteži. Spodaj so razloženi vsi štirje vidiki, k vsakemu pa so priložene tudi tabele, kjer so zbrani rezultati za vsakega od vidikov.

## 6.1 OSNOVNE LASTNOSTI

Značilnosti programskih rešitev so aktivnost, licence, programski jezik in kompatibilnost z drugimi operacijskimi sistemi. Aktivnost programa razumemo kot pogostost posodabljanja rešitve in količino časa, ki je minil od zadnjega popravka. Programski jezik razumemo kot strukturo ogrodja programa, kompatibilnost pa kot zmožnost delovanja na različnih platformah, kot so Windows, Mac in Linux. Naj še dodam, da so rešitve, ki delujejo na sistemih Linux, združljive z sistemom Unix in njegovimi različicami.

*Tabela 2: Osnovne lastnosti izbranih rešitev za podatkovno rudarjenje*

| Produkt | Dejavnost | Programski jezik | Podpora Linux | Podpora Windows | Podpora MAC |
|---|---|---|---|---|---|
| **KNIME** | Visoka | Java | DA | DA | DA |
| **WEKA** | Visoka | Java | DA | DA | DA |
| **RAPIDMINER** | Visoka | Java | DA | DA | DA |
| **TANAGRA** | Nizka | C++ | NE | DA | NE |
| **R** | Visoka | R | DA | DA | DA |
| **ORANGE** | Visoka | C++ Phyton | DA | DA | DA |

## 6.2 PODATKOVNI VIRI

S področja ekonomije, medicine, bioinformatike, farmacije in športa prihajajo različni tipi podatkov iz različnih virov. Zmožnost dostopa do različnih tipov podatkov je velika prednost pri izbiri rešitve za podatkovno rudarjenje. V Tabeli 3 so našteti različni podatkovni viri, na katere vsakodnevno naletimo, in vnešeno je še, katere od naštetih je možno uvoziti v program ter z njim upravljati.

*Tabela 3: Sposobnost uvoza različnih podatkovnih virov rešitev za podatkovno rudarjenje*

| Produkt | SQL Server | My SQL | Oracle | Access | ODBC | JDBC | ARFF | CSV | Excel |
|---|---|---|---|---|---|---|---|---|---|
| **KNIME** | NE | DA | NE | DA | DA | DA | DA | DA | DA |
| **WEKA** | DA | DA | DA | DA | DA | DA | DA | DA | DA |
| **RAPIDMINER** | DA | DA | DA | DA | DA | DA | DA | DA | DA |
| **TANAGRA** | DA | DA | NE | NE | DA | NE | DA | DA | DA |
| **R** | DA | DA | DA | DA | DA | DA | DA | DA | DA |
| **ORANGE** | DA | DA | NE | DA | DA | NE | DA | DA | DA |

## 6.3 FUNKCIONALNOST ODPRTOKODNIH REŠITEV ZA PODATKOVNO RUDARJENJE

Zmožnost reševanja različnih problemov in težav podatkovnega rudarjenja je ključnega pomena za vsakega raziskovalca. Funkcionalnost je razdeljena na predpripravo podatkov, osnovne funkcije podatkovnega rudarjenja (klasifikacija in napovedovanje, analiza skupin, analiza asociacij, KMeans), napredne funkcije podatkovnega rudarjenja (SVM, Bayes, Boosting, naključna drevesa), ovrednotenje rezultatov ter prikaz podatkov in modelov v GUI (angl. *Graphic User Interface*). Pri predpripravi in prikazu podatkov ter modelov sem program ovrednotil s številkami od 1 do 5, kjer 1 pomeni najslabši in 5 najboljši program. Vsa orodja ponujajo osnovne funkcije podatkovnega rudarjenja, razlike so le v predstavitvi in zmožnostih prikaza.

*Tabela 4: Funkcionalnost rešitev za podatkovno rudarjenje*

| **Produkt** | **Predpripr ava podat kov** | **Bayes** | **Odločit vena dreves a** | **Nnet** | **SVM** | **Boosting** | **K-means** | **Prikaz podat kov in model ov** |
|---|---|---|---|---|---|---|---|---|
| **KNIME** | 5 | DA | DA | DA | DA | DA | DA | 5 |
| **WEKA** | 4 | DA | DA | DA | DA | DA | DA | 4 |
| **RAPIDMIN ER** | 5 | DA | DA | DA | DA | DA | DA | 5 |
| **TANAGRA** | 4 | DA | DA | DA | DA | DA | DA | 3 |
| **R** | 3 | DA | DA | DA | DA | DA | DA | 4 |
| **ORANGE** | 5 | DA | DA | DA | DA | DA | DA | 5 |

## 6.4 UPORABNOST ODPRTOKODNIH REŠITEV ZA PODATKOVNO RUDARJENJE

Uporabnost opisuje, kako na najbolj enostaven način uporabljati programske rešitve pri reševanju poslovnih problemov in biti pri tem čim bolj uspešen. Med vsemi dejavniki, ki vplivajo na proces podatkovnega rudarjenja, izpostavljamo najpomembnejše tri in sicer človeško interakcijo s sistemom, interoperabilnost in razširljivost programske opreme.

Človeška interakcija kaže, koliko mora uporabnik posegati v sam proces podatkovnega rudarjenja. Samostojno pomeni, da sistem ne potrebuje nameščanja vseh parametrov, temveč uporabnik vstavi samo želene podatke in programska rešitev samostojno izvede postopek.

Če je postopek voden, pomeni, da nam orodje pomaga, oziroma nam je na voljo v pomoč pri celotnem procesu, Ročno pa pomeni, da v procesu podatkovnega rudarjenja sistem ne ponuja nobene pomoči oz vodenja.

Interoperabilnost je definirana kot zmožnost informacijskih sistemov in poslovnih procesov, ki jih ti sistemi podpirajo, da izmenjujejo podatke, informacije, modele, pri podatkovnem rudarjenju pa to pomeni, da lahko sistemi podpirajo in pa so kompatibilni z modeli v PMML (angl. *Predictive Model Markup Language*), kar omogoča standarden način predstavitve modela za analizo podatkov.

Pri razširljivosti sem gledal na to, kako je določena programska rešitev razširljiva in kakšen so te možnosti. *Z odlično* sem ocenil tiste, ki ponujajo velik nabor vtičnikov in knjižnic, ki jih je mogoče tudi samostojno dodajati ali sprogramirati. *Preprosto* pomeni, da je bil ta kriterij zmanjšan na minimum, z izrazom *nerazširljivo* pa sem hotel izpostaviti, da nam programska rešitev ne ponuja nikakršnih možnosti razširitve programske opreme, vendar mi ga sploh ni bilo treba uporabiti.

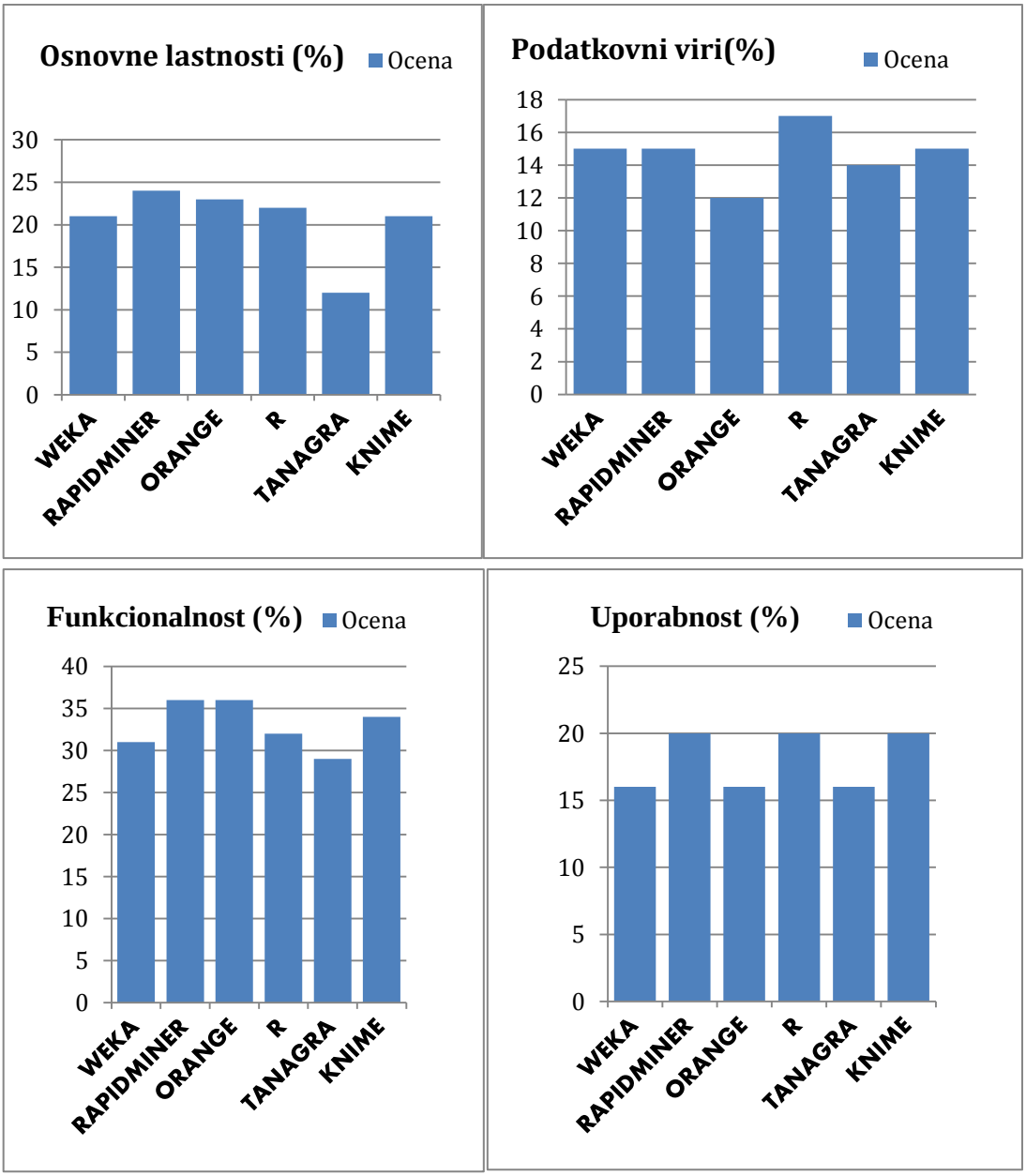
*Tabela 5: Uporabnost rešitev za podatkovno rudarjenje*

| **Produkt** | **Človeška interakcija** | **Interoperabilnost** | **Razširljivost** |
|---|---|---|---|
| **KNIME** | Ročno | PMML | Odlično |
| **WEKA** | Ročno | NE | Odlično |
| **RAPIDMINER** | Ročno | PMML | Odlično |
| **TANAGRA** | Ročno | NE | Preprosto |
| **R** | Ročno | PMML | Odlično |
| **ORANGE** | Ročno | NE | Odlično |

## 6.5 OCENA PROGRAMSKIH REŠITEV ZA PODATKOVNO RUDARJENJE

Na podlagi rezultatov analize in testa sem izračunal ocene za vsako programsko rešitev in vsak vidik posebej. Spodaj na sliki 14 so prikazani rezultati analize:

*Slika 14: Rezultati posameznih analiz štirih vidikov ocenjevanja programskih rešitev*




Pri osnovnih lastnostih se najbolje odreže program RapidMiner, takoj za njim pa sledijo vse ostale rešitve z izjemo Tanagre, ki jo nezmožnost delovanja na drugih operacijskih sistemih potisne v ozadje. Med podatkovnimi viri vodi programska rešitev R, saj je z njo možno obdelati skoraj vse znane tipe podatkov, ob tem pa je zmožna analize velikih podatkovnih setov in baz. Pri funkcionalnosti ni velikega odstopanja med posameznimi rešitvami, kot najbolj funkcionalni pa sem ocenil RapidMiner in Orange. Zaradi slabega grafičnega prikaza in predpriprave podatkov na je zadnjem mestu ponovno Tanagra. Zadnji vidik je vidik

uporabnosti, kjer so rešitve razdeljene v dve skupini. Slabše se odrežejo Tanagra, Orange in WEKA, ostali pa imajo višjo oceno predvsem zaradi podpore PMML.

# 7 ANALIZA REZULTATOV

Končni rezultat, prikazan na sliki 15, kaže na izenačenost programskih rešitev, odstopa samo Tanagra, saj ne ponuja tako veliko možnosti analize podatkov kot ostala orodja, kar pomeni, da se program ne razvija naprej, temveč stagnira. Najbolje sem ocenil rešitev RapidMiner, saj s svojim naborom funkcij in odlično vizualizacijo presega vse ostale rešitve. Le malenkostno zaostajata KNIME in R, za njima pa Orange in WEKA. Pri izbiri programa se moramo vprašati torej kakšni so cilji projekta in pa kakšno je naše predznanje o podatkovnem rudarjenju ter programiranju. Za rudarjenje kemijskih in farmacevtskih podatkov je najbolj priporočljivo orodje Knime, dočim je program R nadmočen pri statistični analizi. Navezujoč na naštete podatke in analizo, spodaj v Tabelah 6 in 7 navajam dobre in slabe lastnosti odprtokodnih programskih rešitev za podatkovno rudarjenje.

*Slika 15: Končni rezultat analize odprtokodnih rešitev za podatkovno rudarjenje*

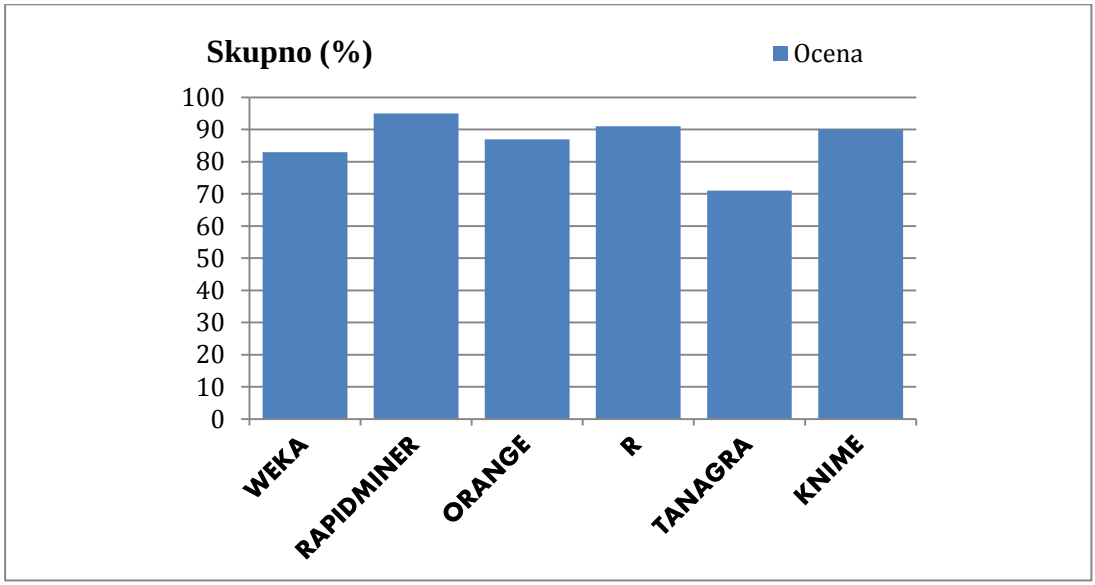


*Tabela 6: Dobre lastnosti odprtokodnih rešitev za podatkovno rudarjenje*

| **DOBRE LASTNOSTI** | |
|---|---|
| **Podpora in kompatibilnost z operacijskimi sistemi** | Vse rešitve podpirajo Linux, Windows in Mac platformo, z izjemo TANAGRE, ki deluje le na operacijskem sistemu Windows. |
| **Človeška interakcija** | Vse rešitve ponujajo podporo in pomoč pri procesu, vse rešitve razen Rattle in WEKA pa poleg tega ponujajo možnost uporabe povleci in spusti načrtovanje procesa, ki ga prikazujejo s pomočjo delotokov. |
| **Obilje funkcij in algoritmov** | Vsaka programska rešitev izmed zgornjih ima integrirano veliko število algoritmov za predpripravo podatkov in modelov. Nekatere izmed njih ponujajo celo več kot 100 algoritmov, kar je lahko tudi dvorezen meč, saj lahko s preobiljem zmede povprečnega uporabnika. |
| **Večkratna uporaba izvedenih testov in primerov** | Vse rešitve omogočajo večkratno uporabo primerov. Rattle in pa KNIME omogočata izvoz modelov v PMML, kar pomeni, da so kompatibilne z ostalimi PMML podprtimi aplikacijami. |
| **Enostavna razširljivost** | Zmožnost enostavne razširitve je prisotna pri vseh programskih rešitvah, nekateri jih ponujajo kot skripte, drugi kot vtičnike in gradnike. Za nekatere je potrebno zanje programskih jezikov in programiranja, druge je mogoče z nekaj kliki vključiti v program. |

*Tabela 7: Slabe lastnosti odprtokodnih rešitev za podatkovno rudarjenje*

| **SLABE LASTNOSTI** | |
|---|---|
| **Podpora raznolikim tipom podatkov** | Nekatere rešitve so omejene in zmožne uvoza le nekaj vrst podatkov, obstajajo pa na internetu na voljo tudi namenski prevajalniki, ki zmorejo prevesti podatke iz enega formata v drugi, vendar je postopek zamuden, zlasti pri velikih količinah podatkov. |
| **Manjše težave pri obdelavi velikih podatkovnih baz** | V današnjem svetu odprtokodne rešitve s težavo shajajo s takšnimi nalogami, saj človek vsak dan ustvari na milijone informacij in podatkov z nekaj milijoni vrstic in stolpcev. Večina jih deluje odlično pri manjših količinah podatkov in prikazu algoritmov, vendar se omejitve pokažejo pri poslovni rabi. Sam pri analizi nisem naletel na ta problem, saj je bil testni podatkovni set majhen, vendar pa podobna analize iz interneta (Chen, Yunming, Williams, & Xu, 2007, str. 9) izpostavlja problem pri obdelavi velikih podatkovnih setov. Možno je sicer podatkovne baze razbiti na manjše dele, vendar smo prikrajšani za celotno izkušnjo. Največkrat je težava pri uporabnikih, saj ne premore vsak aktualne strojne opreme, ki je zmožna poganjati velike podatkovne zbirke. |
| **Podpora v obliki dokumentacije in storitev** | Po pregledu vseh rešitev sem ugotovil, da večina programskih rešitev ne premore podpore v obliki papirne dokumentacije oz. jo ponuja zelo malo, kar omejuje uporabnike, ki bi želeli hitro razumeti delovanje programa. Druga stvar je razmeroma slaba podpora proizvajalcev, saj temelji na dobri volji skupnosti in uporabnikov. Nekateri sicer ponujajo pomoč, vendar je ta plačljiva. |

# SKLEP

Odprtokodne programske rešitve so naredile velik korak naprej v primerjavi s tem, kar so omogočale v preteklosti. Funkcionalnost in zanesljivost sta le dva od kriterijev, zaradi katerih lahko odprtokodne rešitve postavljamo ob bok komercialnim. Ponujajo sodobne in očem prijetne grafične vmesnike, ponujajo uporabne in interaktivne rešitve, podpirajo razširljivost s pomočjo spreminjanja izvorne kode ali vtičnikov in gradnikov. V diplomski nalogi sem analiziral šest v preteklem letu najpogosteje uporabljenih odprtokodnih orodij za podatkovno rudarjenje. Skozi analizo smo ugotovili, da vsa orodja ponujajo vizualno programiranje v grafičnih vmesnikih ali pa s pomočjo skriptnega jezika. Večina orodij znotraj programov je dobro označena in ponuja odlične nadomestne rešitve za komercialne programe. Stopnja implementacije je seveda drugačna od programa do programa, vendar vsi razvijalci ponujajo rešitve na mnoga vprašanja, povezana s poslovnim svetom.

Kot pričakovano, ima vsaka od analiziranih programskih rešitev svoje prednosti in slabosti. Govoriti o zmagovalcu je seveda brezpredmetno, saj ne obstaja najboljše orodje za podatkovno rudarjenje in bi bila sodba subjektivne narave. Naš cilj je bil približati odprtokodne rešitve vsem zainteresiranim, uporabnik bo na koncu izbral tisto programsko rešitev, ki jo bo v tistem času najbolj potreboval. Iz vidika poslovnega sveta bi priporočal rešitvi RapidMiner in pa KNIME, tistim bolj veščim v programiranju pa nemara celo R, saj s pomočjo nekaterih dodatkov in grafičnega vmesnika ponuja zelo veliko možnosti obdelave podatkov in prikaza rezultatov. Vse programske rešitve podpirajo proces podatkovnega rudarjenja na visoki ravni z veliko možnostmi analize in prikaza podatkov, nekatere imajo prednost v statistični analizi, druge v grafičnem prikazu. Zaman boste iskali programsko rešitev, ki ne bo zadovoljila vaših zahtevam. Sama navodila za uporabo so malce oskubljena, vendar lahko z malo raziskovanja po namenskih forumih, skupinah za podporo uporabnikom in z izmenjavo idej vsak problem pripeljemo do rešitve. Odzivnost skupnosti je dokaj hitra in zanesljiva. Čeprav ljudje nimajo finančne koristi od tega, vedno z veseljem priskočijo na pomoč.

Število odprtokodnih rešitev raste iz dneva v dan, vse ne bodo uspešne na dolgi rok, a vendarle jih je na voljo dosti, varne so za uporabo in v preteklosti so se že dokazale na različnih področjih gospodarstva, medicine, športa itd. Ljudje se danes zavedamo konceptov odprtokodnosti in njenega vpliva na razvoj programskih rešitev. Skupaj z orodji, ki se vsakodnevno razvijajo in nadgrajujejo, pa odprtokodne rešitve omogočajo tudi odlično okolje za skupinski razvoj skupnosti, kjer se izmenjujejo ideje, metode, algoritmi in njihova implementacija. Primer spletnih brskalnikov nakazuje, da odprtokodne rešitve zaradi novih idej postajajo nosilci razvoja in v vsakodnevnem življenju pogosto zamenjujejo komercialne programske rešitve vsem uporabnikom.

# LITERATURA IN VIRI


1. Bilab d.o.o. (2013). Poslovna inteligenca. Najdeno 19. maja 2013 na spletnem naslovu http://www.bilab.si/?show=content&id=10&men=14&oce=13&oce=13

2. Berry, M.,& Linoff, G.(2000). *Mastering Data Mining: The Art and Science of Customer Relationship Management.* New York: John Wiley & Sons, Inc.

3. Cortes, C.,& Vapnik, V. (1995). *Machine Learning*. Boston: Kluwer Academic Publishers

4. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R., (2000). *CRISP-DM 1.0*. Chicago: NCR Systems Engineering

5. Chen, X., Yunming, Y., Williams, G., & Xu, X. (2007*).* A Survey of Open Source Data Mining Systems. Najdeno 19. maja 2013 na spletnem naslovu http://link.springer.com/chapter/10.1007%2F978-3-540-77018-3_2#page-1

6. Freund, Y., & Schapire, R. (1996). *The Strenght of Weak Learnability*. Boston: Kluwer Academic Publishers

7. Grantz, J., & Reinsel, D. (2010). Extracting Value from Chaos. Najdeno 19. maja 2013 na spletnem naslovu http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf

8. Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Technigues* (2. izd.)., Amsterdam: Morgan Kaufmann

9. Informatics Research and Developement Unit. (report 2010) Open Source Data Mining Software Evaluation. Najdeno 19. maja 2013 na spletnem naslovu http://www.phiresearchlab.org/downloads/OpenSourceDataMining.pdf

10. Kantardzic, M. (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. New York: John Wiley & Sons.

11. KDnuggets. (Julij 2013). KDnuggets Annual Software Poll: RapidMiner and R vie for first place. Najdeno 19. maja 2013 na spletnem naslovu http://www.kdnuggets.com/2013/06/kdnuggets-annual-software-poll-rapidminer-r-vie-for-first-place.html

12. *Knime*.(2013). Najdeno 19. maja 2013 na spletnem naslovu http://www.knime.org/

13. Monk, E., & Wagner, B. (2006). *Concepts in Enterprise Resource Planning.* (2nd ed.). Boston, MA: Thomson Course Technology.

14. North, M. (2012). *Data mining for the masses.* Washington: A global text project book.

15. *Orange – Data Mining Fruitful & Fun*. (2013). Najdeno 19. maja 2013 na spletnem naslovu http://orange.biolab.si/

16. Pyle, D. (1999). *Data preparation for data mining*. San Francisco: Morgan Kaufmann Publishers.

17. Pyle, D. (2003). *Business Modelling and Data Mining*. San Francisco: Morgan Kaufmann Publishers.

18. Pivk, A.(2001). *Rudarjenje podatkov*. Ljubljana: Inštitut Jožefa Štefana

19. Podatkovno rudarjenje (b.l.) V *Wikipedia*. Najdeno 19. maja 2013 na spletnem naslovu http://en.wikipedia.org/wiki/Data_mining

20. *Rapid -I-*.(2013). Najdeno 19. maja 2013 na spletnem naslovu http://rapid-i.com/content/view/181/190/

21. Rud, O. P. (2001). *Data mining cookbook: Modeling data for marketing, risk, and customer relationship management.* New York: John Wiley & Sons, Inc.

22. Rud, O. P. (2009). *Business Intelligence Success Factors: Tools for Aligning Your Business in the Global Economy*. New York: John Wiley & Sons, Inc.

23. Shearer, C., Hammer, K., Adelman, S., Hardlein, S., & Grecich, D. (2000). The DataWarehousing Institute, *Journal of DataWarehousing*, 5(4), 19-21.

24. Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining* (2. izd.). Boston: Addison-Wesley.

25. *Tanagra – A free DATA MINIG software for teaching and research*. (2013). Najdeno 19. maja 2013 na spletnem naslovu http://eric.univ- lyon2.fr/~ricco/tanagra/en/tanagra.html

26. *The GNU Project.*(2013). Najdeno 19. maja 2013 na spletnem naslovu http://www.gnu.org

27. *The Comprehensive R Archive Network*.(2013). Najdeno 19. maja 2013 na spletnem naslovu http://cran.rstudio.com/bin/windows

28. *Weka*.(2013). Najdeno 19. maja 2013 na spletnem naslovu http://www.cs.waikato.ac.nz/~ml/weka/

29. Witten, I. H., & Frank, E. (2005). *Data mining: practical machine learning tools and techniques* (2. izd.). Amsterdam: Morgan Kaufmann.

30. Zupan B., & Demšar J. (marec 2008). Open-source tools for data mining. Najdeno 19. maja 2013 na spletnem naslovu http://www.slideshare.net/Tommy96/opensource-tools-for-data-mining