UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Grega Boštjančič

Podatkovno rudarjenje s sistemom Oracle Data Mining (11g)

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Matjaž Kukar

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00487/2009
Datum: 15.10.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GREGA BOŠTJANČIČ**

Naslov: **PODATKOVNO RUDARJENJE S SISTEMOM ORACLE DATA MINING (11G)**

ORACLE DATA MINING (11G)

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Kandidat naj v diplomskem delu podrobno opiše zgradbo, komponente in postopke, ki jih omogoča sistem Oracle Data Mining v različici 11g. Pri tem naj se osredotoči na podporo procesnemu standardu CRISP-DM in drugim standardom. V primerjavi z odprtokodnim sistemom Weka naj ovrednoti sistem Oracle Data Mining glede na različne kriterije, med drugim uporabniško izkušnjo, doseženo točnost zgrajenih modelov, hitrost in skalabilnost glede na velikost vhodne množice podatkov.

Mentor:
doc. dr. Matjaž Kukar

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:
prof. dr. Franc Solina

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani/-a Grega Boštjančič,

z vpisno številko 63050235,

sem avtor/-ica diplomskega dela z naslovom:

Podatkovno rudarjenje s sistemom Oracle Data Mining (11g)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom doc. dr. Matjaža Kukarja;
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 14.1.2010 Podpis avtorja: _____

Zahvala

Zahvaljujem se mentorju doc. dr. Matjažu Kukarju za ideje, kritike, spodbudo in pomoč pri izdelavi diplomskega dela. Zahvaljujem se tudi mami, bratu, sestri ter Zali za podporo v času nastajanja te naloge. Zahvalil bi se tudi stricu Andreju, ki me je kot prvi navdušil za svet računalništva ter mi nudil oporo skozi celoten študij.

Kazalo

Seznam uporabljenih kratic in simbolov:

Oznaka	Opis
CRISP-DM	Cross-Industry Standard Process for Data Mining
ODM	Oracle Data Miner
MDL	Minimum Description Length
SVM	Support Vector Machine
NMF	Non-negative Matrix Factorization
NB	Naive Bayes
RMSE	The Root Mean Squared Error
MAE	The Mean Absolute Error
GLM	Generalized Linear Models

Povzetek

Podatkovno rudarjenje postaja vedno bolj pomembno na veliko področjih našega življenja. Uporabljajo ga zdravniki pri ugotavljanju bolezenskega stanja, bankirji pri upravljanju s tveganjem in še bi lahko naštevali. Podatkovno rudarjenje je proces odkrivanja informacij, vzorcev in korelacij s preiskovanjem večjih ali manjših količin podatkov. Vsak proces rudarjenja mora biti, če želimo da je karseda uspešen, sestavljen iz točno določenih korakov oz nalog. Za dosego cilja se je ponavadi potrebno večkrat vrniti korak nazaj, da bi izboljšali prejšnje stanje. Ta proces je standardiziran s standardom CRISP-DM (CRoss-Industry Standard Process for Data Mining).

Podatkovno rudarjenje uporablja algoritme za odkrivanje vzorcev ter statistične in matematične tehnike. Zanj je narejenih veliko aplikacij, ki uporabljajo sofisticirano mešanico klasičnih in naprednih algoritmov kot so odločitveno drevo, naivni bayes, metoda podpornih vektorjev ipd.

V nalogi sem primerjal metode in skalabilnost v dveh aplikacijah za podatkovno rudarjenje. Prva je vključena v podatkovno bazo Oracle, Oracle Data Miner, druga, Weka, pa je samostojna, odprtokodna javanska aplikacija. Izkaže se, da je Oracle zmogljivejši pri delu z velikimi količinami podatkov, medtem ko je Weka boljša in natančnejša pri manjših množicah podatkov. Težava Weke je predvsem potratnost s sistemskimi viri, kar ji pri delu z velikimi podatkovnimi množicami onemogoča tekmovanje z Oraclom. Da bi zgradil model, pa more Oracle Data Miner žrtvovati točnost.

Ključne besede

Oracle, Oracle Data Miner, Weka, podatkovno rudarjenje, metode podatkovnega rudarjenja, CRISP-DM.

Abstract

Data mining is becoming more and more important on number of fields in our lives. It is used by doctors defining medical conditions, bankers dealing with risk management and so on. It is a process of discovering information, patterns and correlation with searching through bigger or smaller quantity of data. Every data mining process, if one wants it to be successful, must be composed from exact steps or tasks. To reach our goal, we often have to take a step back to improve the former state. This process is standardizated with standard CRISP-DM (CRoss-Industry Standard Process for Data Mining).

Data mining uses algorithms for pattern discovering as well as statistical and mathematical techniques. There are many applications which use sufisticated mix of classic and advanced algorihtms like decision tree, Naive Bayes, Support Vector Machines etc.

In this paper I compared methods and scalability in two applications for data mining. The first is included in Oracle database, Oracle Data Miner, whereas the second, Weka, is independent, opensource java application. It turns out that Oracle is better with working with larger datasets while Weka is more accurate with smaller datasets. Weka's problem is mainly her wastefulness with system sources which makes her incompetent at working with large datasets. Orcale, however, sacrifaces accuracy in order to be capable of always building a model.

Key words

Oracle, Oracle Data Miner, Weka, data mining, data mining algorithem’s, CRISP-DM.

1 Uvod

1.1 Opredelitev problema

Namen diplomskega dela je primerjati podatkovno rudarjenje v okolju komercialne podatkovne baze ORACLE z brezplačnim, odprtokodnim javanskim programom Weka. Zanima me standardiziran proces podatkovnega rudarjenja in kako je ta proces realiziran v Oraclu in Weki.

CRISP-DM (Cross-Industry Standard Process for Data Mining) model je standardiziral proces podatkovnega rudarjenja. S pomočjo tega modela lahko proces podatkovnega rudarjenja teče hitreje, ceneje, zanesljivejše in bolj vodljivo[2].

Kar se tiče točnosti rudarjenja, ima Weka dokazane nesporne kvalitete, vendar pa ima težave pri delovanju z velikimi množicami podatkov. Želel sem primerjati Oracle Data Miner (ODM) in Weko pri manjših podatkovnih množicah ter stopnjevati velikost le-te do tistega trenutka, ko Weka ne bo več sposobna opraviti svojega dela. Primerjava je smiselna iz večih pogledov. Prvi je točnost. Zanima me, kolikšna bo razlika med Oraclom in Weko. Zanima me tudi, kakšne so razlike v uporabniškem vmesniku, porabi sistemskih virov ter ceni.

1.2 Podatkovno rudarjenje v okolju Oracle Data Miner

Oracle je pridobil osnovo za podatkovno rudarjenje od podjetja Thinking Machines Corporation l.1999. Podjetje je razvilo Darwin, program za podatkovno rudarjenje. Uporabljal se je za marketing podatkovnih baz v financah in telekomunikacijski industriji. Darwin je analiziral ogromne množice strankinih transkcij, demografskih in ostalih podatkov, ki so pogosto dosegli na stotine milijonov zapisov[10].

Pri Oraclu so celoten program napisali na novo, tako da je sedaj del platforme podatkove baze. Podatkovno rudarjenje so prvič predstavili l.2002 v podatkovni bazi verzije 9iR2. Možno je le kot dodatek k Enterprise različici[10].

Orodje Oracle Data Miner vsebuje:

- 6 klasifikacijskih in regresijskih algoritmov
- 1 algoritem za zaznavo anomalij
- 1 algoritem za iskanje povezovalnih pravil
- 1 algoritem za določanje pomembnosti atributov
- 2 algoritma za iskanje vzorcev
- 1 algoritem za ekstrakcijo atributov

1.3 Podatkovno rudarjenje v okolju Weka

L. 1992 so Ianu Wittenu odobrili sredstva za izdelavo orodja za podatkovno rudarjenje. Čez eno leto so luč sveta ugledali uporabniški vmesnik in ostala infrastruktura. Na podlagi predloga Geoffa Holmesa so orodje poimenovali Weka (Waikato Environment for Knowledge Analysis). Andrew Donkin je ustvaril format ARFF. Prva interna verzija je izšla l. 1994 in je bila večinoma napisana v programskem jeziku C. Oktobra l. 1996 je bila izdana prva javna verzija Weke (v 2.1). V začetku l. 1997 so se odločili na novo napisati celoten program v programskem jeziku Java, kar jim je v verziji 3 uspelo dokončati do srede l. 1999[6].

Orodje Weka vsebuje:

- 49 orodij za pripravo podatkov
- 76 klasifikacijskih in regresijskih algoritmov
- 8 algoritmov združevanja v skupine
- 10 algoritmov za izbiro atributov
- 15 ocenjevalcev atributov
- 3 algoritme za iskanje povezaovalnih pravil

1.4 Uporaba rezultatov podatkovnega rudarjenja

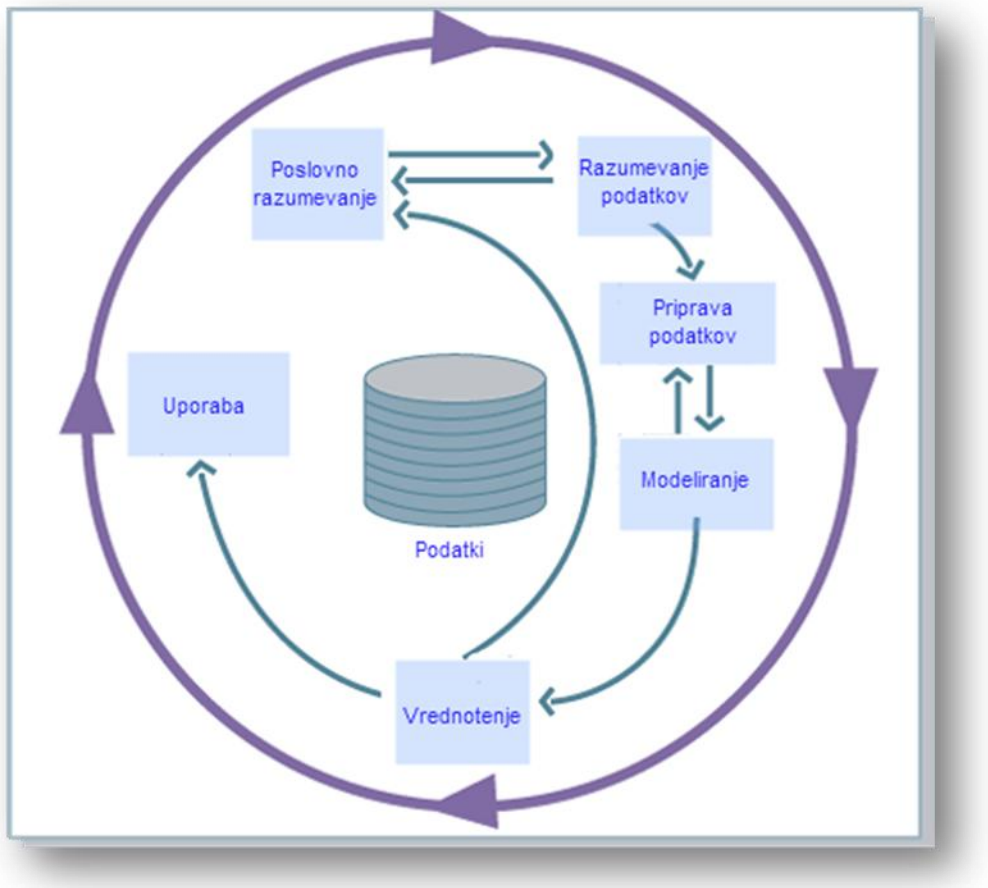
Podatkovno rudarjenje ni samo po sebi namen ampak je namenjeno uporabi. Modele lahko gradimo na enem sistemu ter uporabljamo na drugem. Standard PMML se uporablja za zapis modelov.

1.4.1 PMML

Data Mining Group (DMG) je neodvisen, tržno naravnan konzorcij, ki razvija standarde v podatkovnem rudarjenju. Eden izmed njih je Predictive Model Markup Language (PMML). PMML je vodilen standard za statistične modele in modele podatkovnega rudarjenja. Podprt je z strani več kot 20 prodajalcev in organizacij. S pomočjo PMML-ja je enostavno izdelati model na enem sistemu, z eno aplikacijo, in ga uporabiti na drugem, z drugo aplikacijo. PMML uporablja označevalni jezik XML za predstavitev modelov[4]. Pomemben je za prenos modelov iz testnih sistemov na realne, ki niso nujno enaki.

2 Proces podatkovnega rudarjenja po modelu CRISP-DM

Model CRISP-DM so leta 1996 razvili analitiki iz DaimlerChrysler-ja, SPSS-a in NCR-ja. Njihov cilj je bil sestaviti standarden in objektiven model. Bil je delno financiran s strani evropske unije, preko programa ESPRIT. Uporaben je na različnih industrijskih področjih in vseskozi ohranja objektivnost do orodij za podatkovno rudarjenje. Model sestoji iz šestih faz. Večkrat se zgodi, da se je za doseganje izboljšav potrebno vračati na predhodne faze[1].



Slika 2-1 Model CRISP-DM

2.1 1. faza: Poslovno razumevanje

Določitev poslovnih ciljev

Prvi cilj pri analiziranju podatkov je njihovo celotno razumevanje iz poslovne perspektive, kaj želimo doseči. Pogosto imamo več ciljev in omejitev, ki jih je potrebno pravilno uravnovesiti. Cilj je odkriti pomembne faktorje že na začektu. Posledica zanemarjanja tega koraka je lahko potrošnja veliko časa in truda za iskanje pravih odgovorov na napačna vprašanja.

Določiti moramo kriterije, s katerimi bomo določili ali so rezultati uspešni oz uporabni. To je lahko zelo specifično in objektivno kot naprimer zmanjšanje vrtinčenja (churn) strank (odhod k tekmecu)

Ocena situacije

Ta korak vključuje bolj podrobno iskanje dejstev o virih, omejitvah, predpostavkah in ostalih faktorjih. V prejšnjem koraku smo želeli hitro do jedra problema, sedaj pa nas zanimajo vse podrobnosti.

Sestaviti je potrebno seznam predpostavk, ki smo jih naredili v tem projektu. To so lahko predpostavke o podatkih, ki jih je možno preveriti med projektom, lahko pa so to tudi nepreverljive poslovne predpostavke, na katerih stoji projekt. Urediti je potrebno tudi omejitve projekta. Te so lahko pomanjkanje virov za izvedbo določenih nalog v projektu ali pa pravne oz. etične v zvezi z uporabo podatkov za izvedbo podatkovnega rudarjenja.

Določitev ciljev podatkovnega rudarjenja

Poslovni cilj je cilj v poslovni terminologiji. Cilj podatkovnega rudarjenja je tehnične narave. Na primer poslovni cilj je lahko “Povečati kataloško prodajo obstoječim kupcem”, medtem ko bi bil cilj podatkovnega rudarjenja “Napovedati, koliko stvari bo stranka kupila glede na njene značilnosti v preteklosti in glede na ceno predmeta”. Opisati moramo predvidene učinke projekta, ki omogočajo dosego poslovnih ciljev. Definirati moramo kriterije za uspešen izid projekta v tehničnem smislu, na primer določen nivo točnosti pri napovedovanju. Kot pri poslovnih ciljih je nujno le-te opisati subjektivno. V tem primeru moramo zapisati, kdo jih je oblikoval.

Izdelava načrta projekta

Opisati moramo načrt, kako doseči cilje podatkovnega rudarjenja in poslovne cilje. Načrt mora biti sestavljen iz pričakovanih korakov, ki jih bomo izvedli med projektom, vključno z izbiro orodij in tehnik.

2.2 2. faza: Razumevanje podatkov

Zbiranje začetnih podatkov

Zbrati moramo podatke, ki smo jih navedli kot vir. Začetno zbiranje vključuje tudi uporabo specifičnih orodij za razumevanje podatkov. Navesti moramo podatkovne množice, ki smo jih zbrali, metode, s katerimi smo jih pridobili, in morebitne težave pri tem koraku.

Opis podatkov

Opisati moramo podatke, ki smo jih zbrali. To vključuje format podatkov, velikost le-teh, število polj v zapisu, unikatne atribute in ostale morebitne lastnosti podatkov, ki smo jih odkrili.

Raziskava podatkov

V tem koraku uporabljamo vprašanja povezana z podatkovnim rudarjenjem, ki zadevajo SQL povpraševanje (SQL Query), vizualizacijo in poročila. Poročila vsebujejo distribucijo ključev, relacijo med pari atributov in rezultate enostavne agregacije.

Preverba kvalitete podatkov

Potrebno je raziskati kvaliteto podatkov in si postavljati vprašanja, kot so: so ti podatki popolni (Pokrivajo vse potrebne primere?), ali so pravilni, ali v njih manjkajo vrednosti (prazna polja) ter, kako so predstavljene prazne vrednosti, kje se pojavljajo in kako pogosta so.

2.3 3. faza: Priprava podatkov

V tej fazi nastane podatkovna množica, ki jo bomo v tem projektu uporabili za modeliranje ali analiziranje.

Izbira podatkov

Izbrati moramo podatke, ki jih bomo uporabili za analizo. Kriterij vključuje relevantnost za cilje podatkovnega rudarjenja, kvaliteto in tehnične omejitve, kot so velikost podatkov ali tipi podatkov. Vključuje tudi izbiro atributov ter izbiro vrstic v tabeli. Naredimo seznam podatkov, ki bodo vključeni oz izključeni in navedemo razloge za te odločitve.

Čiščenje podatkov

Glede na tehniko analize podatkov je potrebno povišati kvaliteto podatkov. To lahko vključuje izbiro čistih podatkovnih podmnožic, vstavljanje ustreznih pomot ali bolj prizadevne tehnike, kot je ocenitev manjkajočih podatkov z modeliranjem. Opisati moramo, katere odločitve smo sprejeli, in katera dejanja smo izvedli zaradi povečanja kvalitete podatkov. Potrebno je upoštevati transformacije podatkov zaradi čiščenja in morebiten vpliv le-teh na rezultate analiz.

Oblikovanje podatkov

V tem koraku oblikujemo podatke z operacijami, kot so izdelava izpeljanih atributov, celotnih zapisov ali preoblikovanje obstoječih atributov. Izpeljani atributi so novi atributi, ki so oblikovani iz enega ali več obstoječih atributov v istem zapisu. Primer: površina = dolžina * širina.

Opisati moramo ustvarjanje novih zapisov. Naprimer: ustvari zapise za stranke ki niso nakupovale v zadnjem letu. Ni bilo razloga, da bi take podatke imeli v surovih podatkih, zaradi modeliranja pa bi bilo koristno imeti tudi take zapise.

Sestavljanje podatkov

To je uporaba metod, kjer združujemo informacije iz večih tabel ali zapisov in ustvarimo nove zapise ali vrednosti. Združevanje tabel se sklicuje na spajanje dveh ali več tabel, ki imajo različne informacije o istih objektih. Primer: veriga trgovin ima eno tabelo z informacijami o splošnih karakteristikah posamezne trgovine, drugo s seštevki prodaj in tretjo z demografskimi podatki okolja trgovine. Vsaka od teh tabel vsebuje en zapis za vsako trgovino. Z združevanjem polj iz vhodnih tabel te tabele lahko združimo skupaj v novo tabelo z enim zapisom za vsako trgovino.

Formatiranje podatkov

Formatiranje preoblikovanj se primarno nanaša na sintaktične spremembe podatkov, ki ne spremenijo njihovega pomena, a so lahko potrebne za uporabo modelirnega orodja. Nekaj orodij ima zahteve po posebnem vrstnem redu atributov, kot npr. da je prvi ključ zadnji pa razredni atribut. Navadno so zapisi urejeni po določenem zaporedju, toda algoritem zahteva povsem drugačno ureditev. Na primer, ko uporabljamo nevronske mreže, je na splošno najbolje, da so zapisi predstavljeni v naključnem redu. Možna pa je tudi odstranitev vejic iz .csv datoteke ipd.

2.4 4. faza: modeliranje

Izbira tehnike modeliranja

Prvi korak v fazi modeliranja je izbira dejanske tehnike katero bomo uporabili za modeliranje. Čeprav ste že lahko izbrali orodje v fazi poslovnega razumevanja, se tokratna izbira nanaša na bolj specifično izbiro, kot npr. Odločitveno drevo z C4.5 (razvil ga je Ross Quinlan)[3]. Če uporabljamo več tehnik, je potrebno ta korak ponoviti za vsako tehniko[1].

Izgradnja testnega načrta

Preden dejansko zgradimo model, moramo najprej narediti proceduro oz. mehanizem, ki bo testiral kvaliteto modela. Na primer, v nadzorovani nalogi podatkovnega rudarjenja kot je klasifikacija, je pogosta uporaba razmerja napak za merjenje kvalitete modela. Zato ločimo podatkovno množico na učni in testni del. Na učni množici podatkov zgradimo, na testni množici pa preverimo kakovost.

Grajenje modela

Na pripravljeni podatkovni množici za kreiranje enega ali več modelov poženemo modelirno orodje. Pri vsakem modelirnem orodju je pogosto veliko število parametrov, ki jih lahko prilagodimo, preden orodje zgradi model.

Ocenitev modela

Inženir podatkovnega rudarjenja modele interpretira glede na domeno svojega znanja ter glede na merila uspešnosti podatkovnega rudarjenja in želenega testnega načrta. Medtem ko inženir podatkovnega rudarjenja sodi uspešnost modela bolj tehnično, mora poslovni analitik oceniti uspešnost v poslovnem kontekstu. Inženir razvrsti zgrajene modele glede na ocenjevalne kriterije.

.

2.5 5. faza: Vrednotenje

Rezultati vrednotenja

V prejšnjih korakih vrednotenja smo se ukvarjali s faktorji kot so točnost in splošnost modela. Ta korak ocenjuje stopnjo ujemanja modela s poslovnimi cilji in išče poslovne razloge zaradi katerih bi lahko bil model pomanjkljiv. Druga možnost vrednotenja je testiranje modela na testnih aplikacijah v resnični aplikaciji, v kolikor časovne in finančne omejitve to dopuščajo. Po vrednotenju modela glede na poslovne kriterije tisti modeli, ki dosegajo merila, postanejo priznani.

Pregled procesa

V tem trenutku imamo model, ki izgleda zadovoljiv glede na poslovne potrebe. Sedaj je potreben še globlji pregled, da bi ugotovili, če smo spregledali kakšen pomemben faktor. Ta pregled vključuje tudi jamstvo kakovosti npr. Ali smo pravilno zgradili model? Ali smo uporablili samo tiste atribute, ki jih smemo?

Določitev naslednjega koraka

Glede na rezultate ocenjevanja in pregleda procesa se je potrebno odločiti ali projekt zaključimo in preidemo na uporabo ali se vrnemo korak ali več nazaj v prejšnje faze. To vključuje analizo virov in proračuna.

2.6 6. faza: Uporaba

Načrtovanje uporabe

Da bi uporabili rezultate podatkovnega rudarjenja v poslovnem okolju, ta korak glede na rezultate ocenjevanja naredi strategijo za uporabo. Tu nastanejo potrebni koraki in način, kako jih je potrebno izvesti.

Načrtovanje kontroliranja in vzdrževanja

Kontroliranje in vzdrževanja sta zelo pomembni točki, še posebej, če rezultat podatkovnega rudarjenja postane del vsakodnevne uporabe v podjetju in njegovem okolju. Skrbna priprava strategije vzdrževanja pripomore k izognitvi zamudni, nepravilni uporabi rezultatov.

Izdelava končnega poročila

Na koncu projekta je potrebno napisati končno poročilo. Glede na načrt uporabe je lahko to poročilo le pregled projekta in izkušnje z njim, ali pa končna in obširna predstavitev rezultatov podatkovnega rudarjenja. Ponavadi je na koncu projekta sklican sestanek, kjer so rezultati predstavljeni stranki oz naročniku.

Pregled projekta

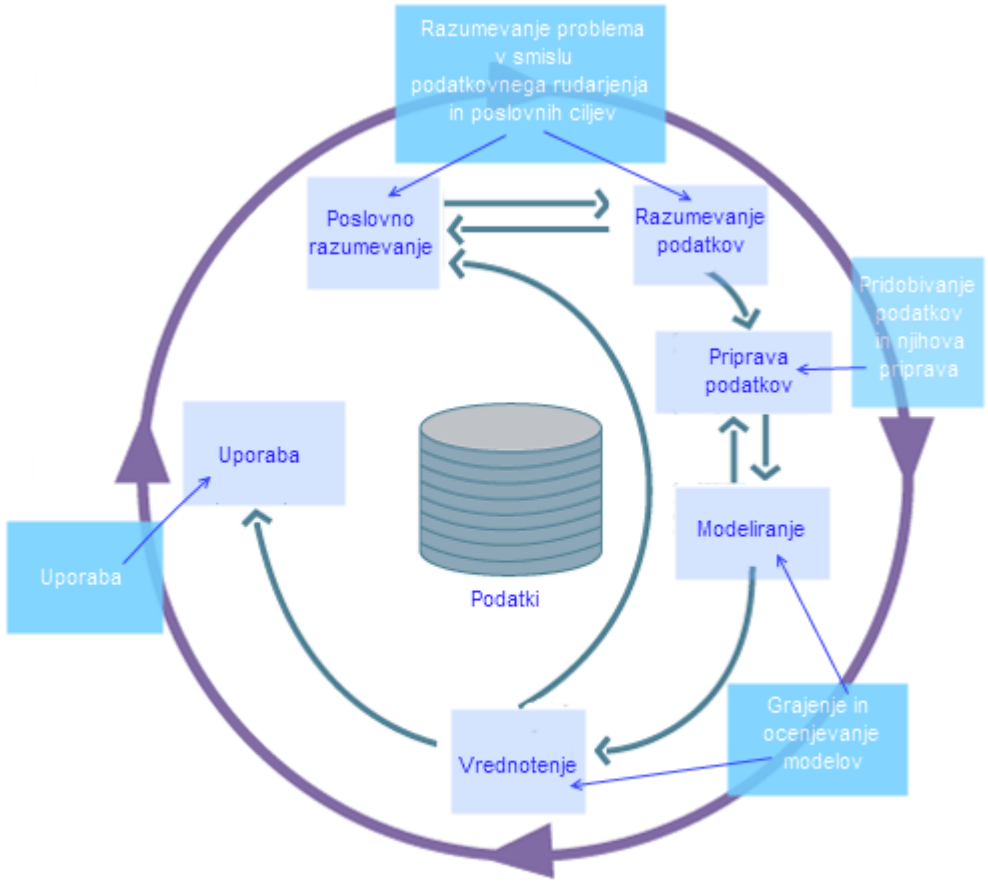
V zadnjem koraku zadnje faze ocenimo, kaj je uspelo in kaj ne, kaj smo naredili dobro in kaj bi lahko izboljšali. Izdelamo dokumentacijo z vsemi pomembnimi izkušnjami, ki smo jih zbrali v tem projektu: na primer pasti, zavajajoče pristope ali namige za izbiro najbolše tehnike v podobnih situacijah.

3 Proces podatkovnega rudarjenja v okolju Oracle

3.1 Umestitev Oracle Data Miner-ja v model CRIP-DM

Oraclovo orodje Oracle Data Miner nam predstavi štiri, spodaj opisane faze[7]:

1. Razumevanje problema v smislu podatkovnega rudarjenja in poslovnih ciljev
2. Pridobivanje podatkov in njihova priprava
3. Grajenje in ocenjevanje modelov
4. Uporaba



Slika 3-1 Umestitev procesa ODM v model CRISP-DM

Pri Oraclu so, tam kjer se jim je to zdelo potrebno, združili določene faze. Potek je podoben kot pri modelu CRIP-DM, vendar bolj poenostavljen, ne tako natančno definiran.

CRISP-DM	ORACLE
Poslovno razumevanje	Razumevanje problema v smislu podatkovnega rudarjenja in poslovnih ciljev
Razumevanje podatkov	
Priprava podatkov	Pridobivanje podatkov in njihova priprava
Modeliranje	Grajenje in ocenjevanje modelov
Vrednotenje	
Uporaba	Uporaba

Table 3-1 Umestitev ODM-ja v model CRISP-DM

3.2 1.faza: Razumevanje problema v smislu podatkovnega rudarjenja in poslovnih ciljev

Poslovni problem mora biti dobro definiran in določen v smislu funkcionalnosti podatkovnega rudarjenja. Na primer, prodaja na drobno, telefonske družbe, finančne institucije in druga podjetja se zanimajo za »vrtinčenje« (churn), kar pomeni, da prej njim zveste stranke prestopijo h konkurenčnemu prodajalcu.

Izjava " Želim uporabiti podatkovno rudarjenje za rešitev svojega problema vrtinčenja" je preveč nejasna. Iz poslovnega vidika je realnost, da je veliko težje in dražje poskušati pridobiti nazaj prebeglo stranko, kot preprečiti zvesti, da odide. Nadalje, možna je nezainteresiranost za zadrževanje malo vrednostne stranke. S tega vidika podatkovnega rudarjenja, je zato problematično napovedati, za katere stranke je prebeg najbolj verjeten in tudi, katere od teh so potencialno visoke vrednosti.

Zato potrebujemo jasno definicijo, kaj je stranka z nizko vrednostjo in kaj vrtinčenje oz prestop. Oboje sta poslovni odločitvi in sta v nekaterih primerih lahko zelo težki. Banka ve, kdaj je stranka zaprla svoj bančni račun, toda kako naj podjetje na drobno ve, kdaj ga je stranka zamenjala za drugo? Mogoče je to lahko določeno, ko se kar naenkrat naročila dramatično znižajo.

Predpostavimo, da so te poslovne definicije določene. Nato lahko določimo problem: »Potrebujem seznam strank z visoko vrednostjo, pri katerih je verjetnost, da bodo prestopile, največja in ponuditi tem strankam nekaj spodbudnega ter preprečiti prestop. « Definicija, kaj je »največja verjetnost« bo ostala odprta, dokler ne vidimo rezultatov, ki jih priskrbi Oracle Data Mining.

1.faza v Oracle Data Minerju

ODM ni namenjen za pomoč v prvi fazi procesa. Tu je potrebno znanje iz problemske domene za posameznike, ki ta problem rešujejo.

3.3 2. faza: Pridobivanje podatkov in njihova priprava

Splošno pravilo v podatkovnem rudarjenju je zbrati čim več informacij o posamezni enoti in potem prepustiti algoritmom podatkovnega rudarjenja, da nam pokažejo, če je potrebno kakšno filtriranje.

Še posebej pomembno je, da ne eliminiramo določenih atributov samo zato, ker se nam zdi, da niso pomembni.

Naj ODM algoritmi odločijo o njihovi pomembnosti. Nadalje, ker je cilj zgraditi profil obnašanja, ki ga lahko apliciramo katerikoli enoti, bi morali specifične atribute, kot so ime, naslov, telefonska številka itd. eliminirati (Vendar so atributi, ki nakazujejo na lokacijo posamezne enote kot je poštna številka, lahko pomembni)

2.faza v Oracle Data Minerju

ODM nam v drugi fazi ponuja nekaj metod za pripravo podatkov. Izbiramo lahko med naslednjimi:

- Pomembnost atributov (Minimum Description Length)
- Odkrivanje anomalij (One-Class Support Vector Machine)
- Združevanje v skupine (K-Means, O-Cluster)
- Povezovalna pravila (Apriori)
- Ekstrakcija atributov (Non-negative Matrix Factorization)

Uporaba le-teh je razmeroma enostavna. Uporabnik si najprej izbere željeno tabelo kot podatkovni vir. V meniju izbere »Activity« in nato »Build«. V meniju ima sedaj na voljo vse tehnike, tako za pripravo podatkov kot za strojno učenje. Uporabnik mora po izbiri metode izbrati razredni atribut, morebitni zunanji vir podatkov ter nastavitve pri grajenju modela. Ko je model zgrajen, ga lahko z izbiro v meniju »activity« ter »apply« apliciramo na želeno podatkovno množico. Po pregledu rezultatov se odločimo o morebitnih ukrepih glede le-te.

3.4 3.faza: Grajenje in ocenjevanje modelov

Med grajenjem in testiranjem modela vodič delovanja Oracle Data Miner-ja avtomatizira veliko težavnih nalog. Težko je vnaprej vedeti, kateri od algoritmov bo najbolje rešil problem, zato ponavadi zgradimo in testiramo več modelov.

Noben model ni popoln in iskanje najboljšega modela ni nujno vprašanje, kateri model ima največjo točnost, temveč bolj vprašanje, kateri tipi napak so sprejemljivi v okviru ciljev, ki smo si jih zadali.

Na primer, banka, ki uporablja podatkovno rudarjenje za napovedovanje tveganja kreditiranja v procesu prošnje posameznega kandidata, želi minimizirati napake, kjer odobri kredit, a stranka dolga ne vrne, ker je taka napaka za banko lahko zelo draga.

Po drugi strani pa bo banka tolerirala določeno število napak pri napovedi, da določena stranka predstavlja tveganje, saj v resnici banke to ne stane veliko (čeprav nekaj malega potencialnega dobička le izgubi).

3.faza v Oracle Data Minerju

V tretji fazi imamo pri ODM-ju na voljo štiri klasifikacijske in dve regresijski metodi za grajenje modelov. Za večrazredne klasifikacijske probleme imamo na voljo algoritme :

- Naivni bayes (Naive Bayes)
- Odločitveno drevo (Decision tree)
- Metoda podpornih vektorjev (Support Vector Machine)

medtem ko Logistična regresija (Logistic Regression) lahko rešuje le binarne probleme.

Za reševanje regresijskih problemov sta v ODM-ju implementirani naslednji metodi:

- Vekčratna regresija (Multiple Regression)
- Metoda podpornih vektorjev

Postopek v ODM-ju je enak kot v prejšni fazi, kjer smo pripravili podatke. Ponvadi zgradimo več modelov.

3.5 4.faza: Uporaba

Oracle Data Mining izdela uporabne rezultate, toda temu ni tako, če jih čimprej ne posredujemo v prave roke. Če nadaljujemo z bančnim problemom vrtinčenja, ko je ODM napovedovalni model apliciran na strankino bazo podatkov, da ustvari urejen niz tistih, ki bodo zamenjali banko, je ustvarjena tabela in napolnjena s StrankaID/napoved/Verjetnost. Rezultati so na voljo z uporabo običajnih metod pri delu s tabelo v podatkovni bazi. Podatke lahko izvozimo tudi v Excel ali Oracle Discover.

4.faza v Oracle Data Minerju

V prejšnji fazi smo zgradili enega ali več modelov. Po pregledu rezultatov ponavadi najboljšega apliciramo na realne podatke. To storimo z izbiro »Activity« ter »Apply«. Izberemo si podatkovno množico, na katero želimo aplicirati naš model, ter pregledamo rezultate. Po potrebi se vračamo fazo ali več nazaj ter tako izboljšujemo model.

Prenos modelov s pomočjo standara PMML

Oracle omogoča, da izvozimo model klasifikacijskega drevesa v XML formatu, ki ustreza standardu PMML[7]. Weka pa omogoča izvažanje regresije (Regression), splošne regresije (General Regression) in nevronskih mrež (Neural Networks), od leta 1999 pa tudi RuleSetModel in TreeModel[11].

4 Metode za pripravo in vizualizacijo podatkov

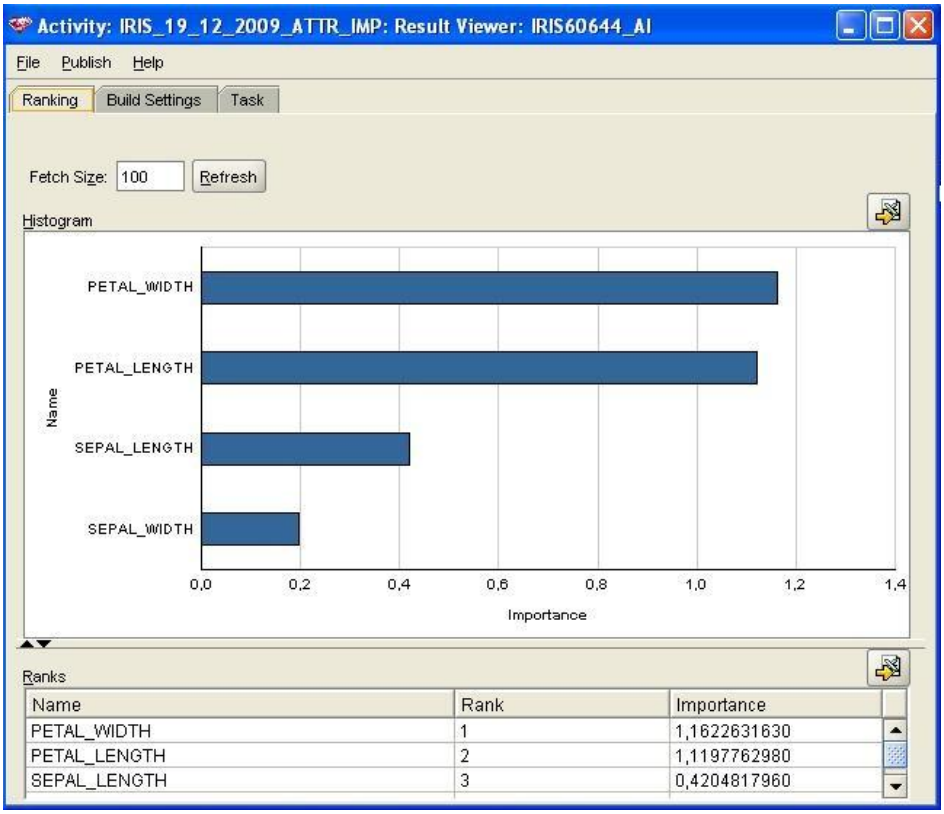
4.1 Pomembnost atributov: Minimalna dolžina opisa

To poglavje opisuje minimalno dolžino opisa (Minimum Description Length MDL), nadzorovano tehniko za izračunavanje pomembnosti atributov[7].

O MDL

MDL je informativno-teoretični model, ki deluje na osnovi izbiranja. MDL predvideva, da je najpreprostejša, najkompaktnejša predstavitev podatkov istočasno tudi najboljša in najverjetnejša pojasnitev podatkov. MDL način se uporablja za grajenje Oracle Data Mining modelov pomembnosti atributov.

MDL obravnava vsak atribut kot napovedovalni model ciljnega razreda. Posamezni napovedni modeli so primerjani in razvrščeni glede na MDL matriko (kompresija v bitih). MDL v izogib preveč ujemajočih rezultatih kaznuje kompleksnost modela. Je načelen pristop, ki upošteva kompleksnost napovednikov (kot modelov), da so primerjave pravične.



Slika 4-1 Rezultat iskanja pomembnosti atributov z MDL

4.2 Odkrivanje anomalij: Enorazredna Metoda podpornih vektorjev

ODM podpira zaznavanje anomalij in kot nenadzorovano funkcijo rudarjenja za zaznavanje redkih primerov v podatkih. Opisana je enorazredna metoda podpornih vektorjev (One-Class Support Vector Machine SVM)

O odkrivanju anomalij

Cilj zaznavanja anomalij je prepoznavanje primerov, ki so nenavadni glede na ostale v podatkovni množici. Odkrivanje anomalij je pomembno orodje pri odkrivanju prevar, vdorov v omrežje in drugih redkih dogodkov ali napak, ki so lahko zelo pomembni, a jih je težko odkriti.

Odkrivanje anomalij lahko uporabimo za reševanje naslednjih problemov:

- Organi pregona zberejo podatke o nezakonitih aktivnostih, a nič o zakonitih aktivnostih. Kaj nas lahko opozori na sumljive aktivnosti? Podatki so vsi istega razreda. Ni nasprotnih primerov.
- Zavarovalna agencija obdeluje milijone zavarovalniških zahtevkov, vedoč, da je zelo majhno število tistih goljufivih. Kako lahko prepoznamo goljufive zahtevke? Podatki o zahtevkih vključujejo zelo majhno število nasprotnih primerov. To so odstopanja.

Enorazredna klasifikacija

Odkrivanje anomalij je oblika klasifikacije. Implementira se kot enorazredna klasifikacija, saj je v učni množici podatkov zastopan samo en razred. Model odkrivanja anomalij predvidi ali je podatkovna točka navadna za dano distribucijo ali ne. Nenavadna podatkovna točka je lahko odstopanje ali primer razreda, ki ga prej nismo videli.

Običajno je klasifikacijski model naučen na množici podatkov, ki vključuje tako primere kot protiprimere za vsak razred, tako da se model lahko nauči razlikovati med obema.

Enorazredni klasifikator razvije profil, ki na splošno opiše tipične primere v testni množici podatkov. Odstopanje od profila se smatra za anomalijo. Na enorazredne klasifikatorje se pogosto navezuje kot na pozitivne varnostne modele, ker iščejo »dobra« vedenja in predvidevajo, da so ostala vedenja »slaba«.

Opomba:

Reševanje problema enorazredne klasifikacije je lahko zahtevno. Točnost enorazrednih klasifikatorjev navadno ni primerljiva s standardnimi klasifikatorji, zgrajenimi z vsebinsko pomembnimi protiprimeri.

Cilj zaznavanja anomalij je priskrbeti uporabne informacije za tisto, o čemer prej bilo na voljo nikakršnih informacij. Vendar, če je dovolj »redkih« primerov, tako da bi lahko razslojeno vzorčenje zagotovilo učno množico podatkov z dovolj protiprimeri za standarden klasifikacijski model, potem je to na splošno boljša izbira.

Zaznavanje anomalij za enorazredne podatke

Pri enorazrednih podatkih imajo vsi primeri isto klasifikacijo. Protiprimere, primere drugih razredov, je kdaj težko podrobno označiti ali pa je njihovo zbiranje drago. Na primer, v klasifikaciji tekstovnega dokumenta je lahko klasificiranje dokumenta pod dano temo preprosto. Vendar pa je lahko prostor dokumentov izven te teme velik in raznolik. Zatorej, podrobna označba ostalih tipov dokumentov kot protiprimerov morda ni izvedljiva. Zaznavanje anomalij se lahko uporabi z namenom, da se najdejo nenavadni primeri ali določen tip dokumentov.

Zaznavanje anomalij za najdbo odstopanj

Odstopanja so primeri, ki so nenavadni, ker padejo izven distribucije, ki se smatra za normalno za (dane) podatke. Na primer, popis podatkov lahko prikaže srednji (mediano) 90.000 EUR in povprečen prihodek gospodinjstva 100.000 EUR, čeprav je en prihodek enega ali dveh gospodinjstev 250.000 EUR. Ti primeri bodo najverjetneje označeni kot odstopanja.

Razdalja med centrom normalne distribucije nam pove, kako tipična je dana točka v razmerju do distribucije podatkov. Vsak primer se lahko razvrsti glede na verjetnost, da je tipičen ali netipičen (atipičen).

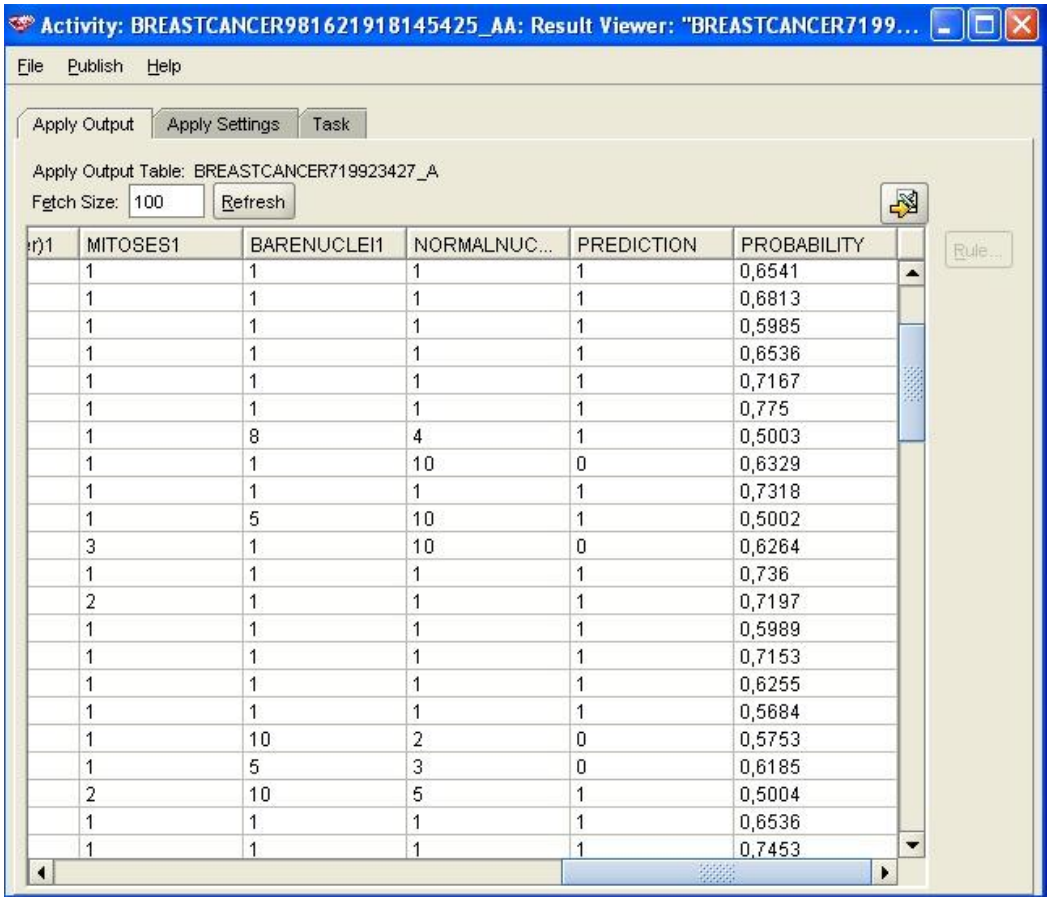
Prisotnost odstopanj ima lahko škodljiv vpliv na številne oblike podatkovnega rudarjenja. Zaznava anomalij se lahko uporabi za zaznavo odstopanj pred rudarjenjem podatkov/začetkom podatkovnega rudarjenja.

Enorazredna Metoda Podpornih Vektorjev (Support vector machine SVM)

Oracle Data Mining uporablja SVM kot enorazredni kvalifikator za zaznavo anomalij. Kadar se SVM uporablja za zaznavo anomalij, njegov algoritem ne uporablja ciljne spremenljivke.

Enorazredni SVM modeli, ko so aplicirani, ustvarijo napoved ali verjetnost za vsak primer v podatkih. Če je napoved 1, je primer obravnavan kot tipičen. Če je napoved 0, je primer obravnavan kot nenavaden. To vedenje odraža dejstvo, da je model naučen z normalno množico podatkov.

Natančno lahko določiš odstotek podatkov za katere pričakuješ, da bodo nenavadni z gradbeno funkcijo SVMS_OUTLIER_RATE. Če veste da je določen delež vaše populacije število sumljivih primerov, potem lahko nastavite stopnjo odstopanj temu odstotku primerno. Ko bo model apliciran na splošno populacijo, bo identificira temu primerno število »redkih« primerov. V osnovnih nastavitvah je to 10%, kar je verjetno visoko za veliko problemov zaznave anomalij.



r)1	MITOSES1	BARENUCLEI1	NORMALNUC...	PREDICTION	PROBABILITY
	1	1	1	1	0,6541
	1	1	1	1	0,6813
	1	1	1	1	0,5985
	1	1	1	1	0,6536
	1	1	1	1	0,7167
	1	1	1	1	0,775
	1	8	4	1	0,5003
	1	1	10	0	0,6329
	1	1	1	1	0,7318
	1	5	10	1	0,5002
	3	1	10	0	0,6264
	1	1	1	1	0,736
	2	1	1	1	0,7197
	1	1	1	1	0,5989
	1	1	1	1	0,7153
	1	1	1	1	0,6255
	1	1	1	1	0,5684
	1	10	2	0	0,5753
	1	5	3	0	0,6185
	2	10	5	1	0,5004
	1	1	1	1	0,6536
	1	1	1	1	0,7453

Slika 4-2 Rezultat zaznavanja anomalij z enorazrednim SVM-jem

4.3 Razvrščanje v skupine: O-Cluster

Orthogonal Partitioning Clustering (O-Cluster) je Oraclov zakonsko zaščiten algoritem za razvrščanje v skupine.

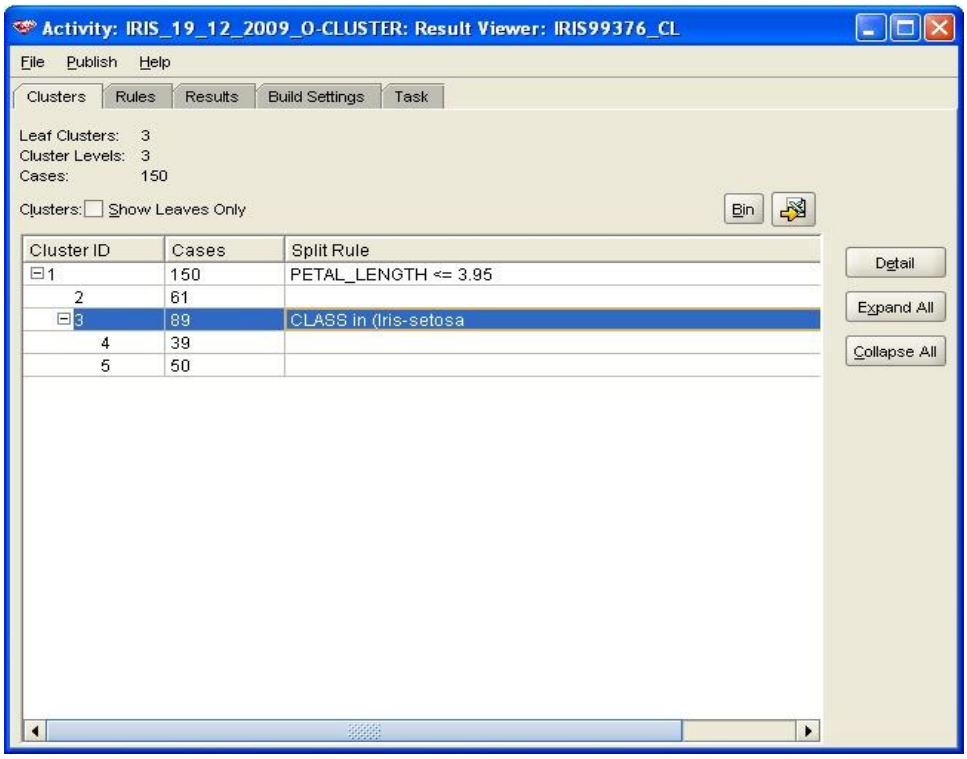
O O-Cluster-ju

O-Cluster algoritem na osnovi mreže ustvari hiearhičen model. V prostoru vhodnih atributov ustvari vzporedne osi, pregrade. Algoritem deluje rekurzivno. Končna hiearhična struktura predstavlja neenakomerno mrežo, ki obloži prostor atributov v skupine. Končne skupine definirajo goste površine v prostoru atributov.

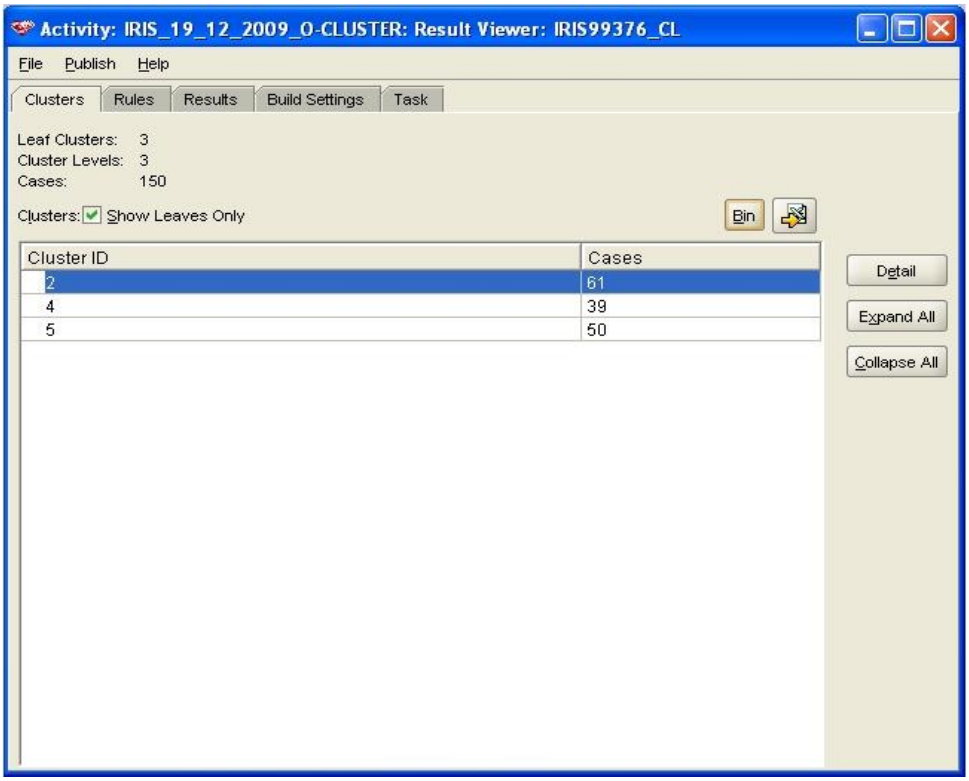
Skupine so opisane z intervali na atributnih oseh, ustreznega centroida in histograma. Parameter občutljivosti določi osnoven nivo specifične teže. Samo površine z najvišjo točko specifične teže nad osnovnim nivojem lahko istovetimo z skupinami.

Algoritem razvrščanja v skupine z metodo voditeljev (k-means) obloži prostor tudi če naravna skupina ne obstaja. Na primer, če je predel enotne specifične teže, ga k-means obloži n skupin (kjer je n določen s strain uporabnika). O-Cluster loči površine z visoko specifično težo, tako da doda ostre ravnine preko površin z nizko specifično težo. O-Cluster potrebuje večmodalne histograme (vrhove in doline). Če ima površina projekcijo z enotno ali monotono specifično težo, ga O-Cluster ne bo razdelil.

Skupine, ki jih odkrije O-Cluster, se uporabljajo pri izgradnji Bayesovega verjetnostnega modela, ki je potem uporabljen pri apliciranju modela za dodelitev podatkovnih točk skupinam. Zgrajen verjetnostni model je mešanica modela, kjer so križane komponente predstavljene s produktom neodvisnih normalnih distribucij za numerične attribute in večnominalnih distribucij za nominalne attribute.



Slika 4-3 Rezultat razvrščanja v skupine z metodo O-CLUSTER



Slika 4-4 Skupine z metodo O-Cluster

4.4 Razvrščanje v skupine: k-Means

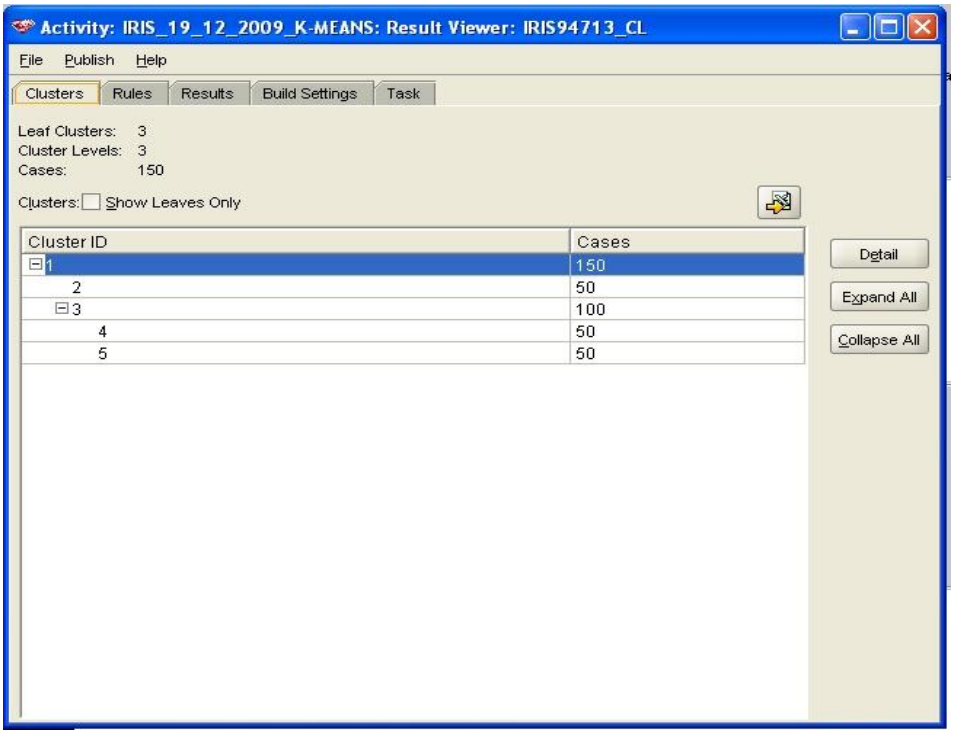
O k-Means

K-Means je algoritem, ki temelji na razdalji med skupinami. Podatke razdeli v prej definirano število skupin (predpostavimo, da je dovolj različnih primerov). Algoritem se sklicuje na razdaljo ter tako meri podobnost med podatkovnimi točkami. Razdalja je lahko Evklidova, kosinusna ali Hitra kosinusna. Podatkovne točko so dodeljene najbližji skupini gelede na uporabljeno razdaljo.

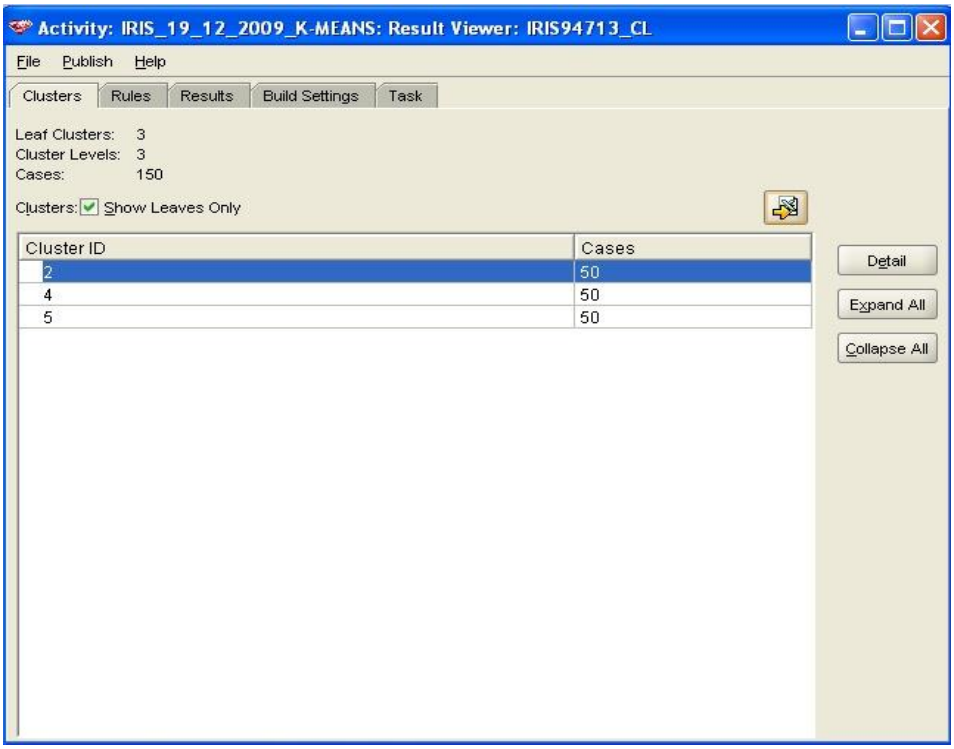
Oracle Data Mining implementira izboljšano verzijo algoritma z naslednjimi lastnostmi:

- Algoritem zgradi model na hierarhičen način, od zgoraj navzdol, ter uporablja binarno delitev in čiščeneje vseh vozlišč na koncu. V tem smislu je algoritem podoben razpolovitvenemu algoritmu k-means. Centoridi notranjih vozlišč v hiearhiji so posodobljeni in tako odsevajo spremembe, medtem ko se drevo razvija. Vrne se celotno drevo.
- Algoritem gradi drevo eno vozlišče za drugim (neuravnotežen pristop). Glede na uporabnikove nastavitve je vozlišče z najvišjo varianco razpolovljeno, da bi povečali velikost drevesa dokler željeno število skupin ni doseženo. Maksimalno število skupin je določeno v nastavitvah gradnje drevesa za modele razvrščanja v skupine
- Algoritem ponuja verjetnostno preizkušanje in določitev podatkov v skupine
- Algoritem za vsako skupino vrne centroid (prototip skupine), histograme (enega za vsak atribut) in pravilo, ki opisuje hiperpredel, ki obsega večino podatkov v skupini. Centroid opiše način za nominalne atribute ali sredino vrednosti in variance za numerične atribute.

Oraclov k-means se izogne potrebi po izgradnji več modelov in priskrbi skupine, ki so dosledno boljše od tradicionalne k-means metode.



Slika 4-5 Rezultat razvrščanja v skupine z metodo K-Means



Slika 4-6 Skupine z metodo K-Means

4.5 Povezovalna pravila: Apriori

Apriori je algoritem, ki ga uporablja Oracle Data Mining za izračunavanje povezovalnih pravil.

O Apriori-ju

Problem pri iskanju povezovalnih pravil je lahko razdeljen na dva podproblema:

- Najti vse kombinacije vseh predmetov v množici transakcij, ki se pojavijo z določeno minimalno frekvenco. Te kombinacije imenujemo frekvenčna množica atributov.
- Izračunati pravila, ki izražajo verjeten so-obstoj atributov v frekvenčni množici.

Apriori izračuna verjetnost, da bo atribut prisoten v frekvenčni množici, če so prisotni kateri od drugih atributov.

Rudarjenje povezovalnih pravil (Association rule mining) je priporočljivo za iskanje povezav, ki vključujejo redke podatke v domenah problemov z velikim številom atributov. Apriori odkriva vzorce s frekventnostjo višjo od praga minimalne podpore (minimal support threshold). Zatorej mora, da najdemo povezave, ki vključujejo redke dogodke, algoritem delovati z zelo nizkimi minimalnimi podpornimi vrednostmi. Vendar lahko pri tem pride do kombinatorične eksplozije števila oštevilčenih frekvenčnih množic in za n atributov dobimo 2^n podmnožic, predvsem v primerih večjega števila atributov. To lahko močno podaljša čas izvajanja. Klasifikacija ali zaznava anomalij je lahko primernejša za odkrivanje redkih dogodkov, ko imajo podatki večje število atributov.

Pravila povezovanja

Algoritem apriori izračuna pravila, ki izražajo verjetnostna razmerja med atributi in pogostimi frekvenčnimi množicami. Na primer, pravilo, ki izvira iz frekvenčnih množic, ki vključuje A, B ali C lahko trdi da, če sta A in B vključena v transakcijo, potem je verjetno, da je C ravno tako vključen.

Pravilo povezovanja pravi, da atribut ali skupina atributov namiguje na prisotnost drugega atributa z enako verjetnostjo. V nasprotju z odločitvenim drevesom, ki napove ciljno spremenljivko, povezovalna pravila preprosto izražajo povezavo.

Predhodnik in posledica

Komponenta ČE povezovalnega pravila je poznana kot predhodnik. Komponenta POTEM je znana kot posledica. Predhodnik in posledica sta ločeni; nimata nobenega skupnega atributa.

Oracle Data Mining podpira povezovalna pravila, ki imajo enega ali več atributov v predhodniku in samo en atribut v posledici.

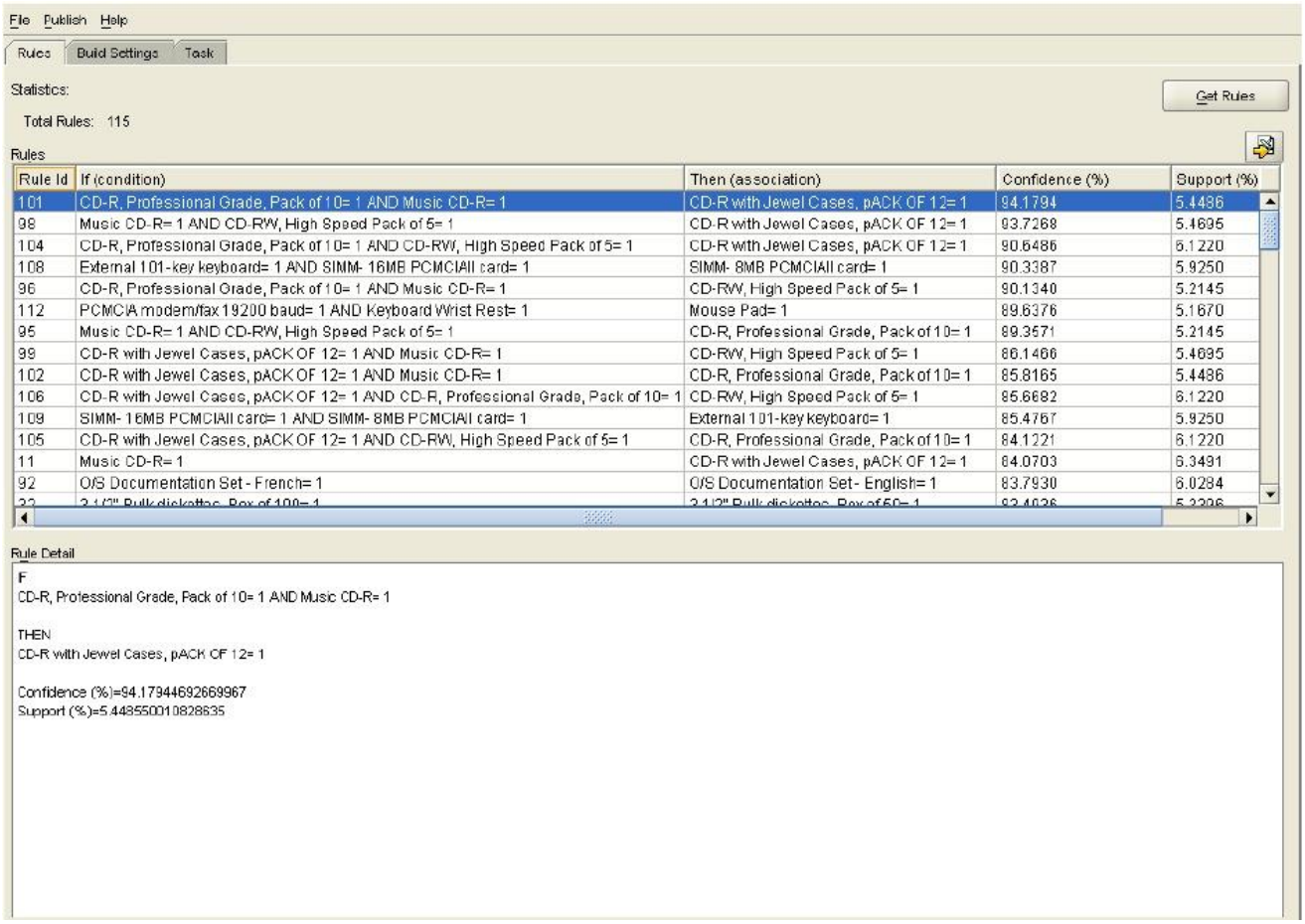
Zaupanje in podpora

Pravila imajo povezovalno zaupanje, ki je pogojna verjetnost, da se bo posledica pojavila, če se bo pojavil predhodnik. ASSO_MIN_CONFIDENCE funkcija določa minimalno zaupanje za pravila. Ta model izključi vsa pravila z vrednostjo, nižjo od zahtevanega minimuma.

Podpora pravila pokaže, kako pogosto se atributi v pravilu pojavijo skupaj. Na primer atribut A in atribut B se pojavita skupaj v 50% transakcij, zato imata sledeči pravili vsak podporo 50%.

- A implicira B
- B implicira A

Podpora je razmerje med transakcijami, ki vključujejo vse atribute v predhodniku in posledici in številu vseh transakcij. Podporo lahko izrazimo kot verjetnost. Podpora (A implicira B) = P (A,B)



Slika 4-7 Rezultat iskanja pravil z metodo Apriori

4.6 Ekstrakcija atributov: Nenegativna matrična faktorizacija

Nenegativna matrična faktorizacija (Non-negative Matrix Factorization NMF) je nenadzorovan algoritem, ki ga uporablja Oracle Data Mining za ekstrakcijo atributov (Feature extraction).

Ekstrakcija atributov

Ekstrakcija atributov je postopek zmanjševanja števila atributov. V nasprotju z izbiro lastnosti, ki razvršča atribute glede na njihovo napovedno pomembnost, ekstrakcija atributov dejansko pretvori atribute. Pretvorjeni atributi ali lastnosti so linearne kombinacije originalnih atributov. Posledica procesa ekstrakcije atributov je veliko manjša in bogatejša množica atributov. Največje možno število lastnosti je nadzorovano s FEAT_NUM_FEATURES nastavitvijo, zgrajeno za modele ekstrakcijo atributov.

Modeli, zgrajeni na podlagi ekstrakcije atributov, so lahko bolj kvalitetni, ker so podatki opisani z manjšim številom bolj pomembnih lastnosti.

Ekstrakcija atributov projecira podatkovno množico z večje dimenzionalnostjo na manjše število dimenzij. Kot tak je uporaben za vizualizacijo podatkov, saj je, zmanjšana na dve ali tri dimenzije, kompleksna podatkovna množica lahko učinkovito vizualizirana.

Nekatere aplikacije ekstrakcije atributov so latentna semantična analiza, stiskanje podatkov, dekompozicija podatkov, projekcija in prepoznavanje vzorcev. Ekstrakcija atributov se prav tako lahko uporablja za povečanje hitrosti in učinkovitosti nadzorovanega učenja.

Ekstrakcija atributov se lahko uporablja za izločevanje tistih zbirk podatkov, kjer so dokumenti predstavljeni z množico ključnih besed in njihovih pogostosti. Vsaka tema (lastnost) je predstavljena s kombinacijo ključnih besed. Dokumenti v zbirki so lahko predstavljeni v smislu odkritih tem.

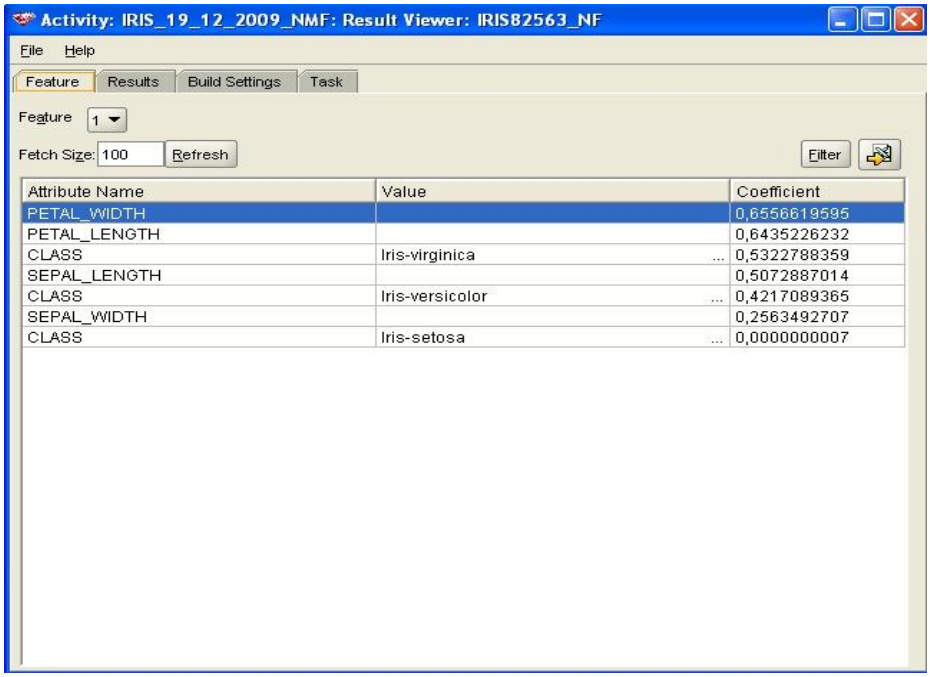
O NMF

NMF je vrhunski algoritem za ekstrakcijo atributov. Uporaben je, kadar imamo veliko atributov in so le-ti dvoumni. Z združevanjem atributov lahko NMF ustvari pomenljive vzorce ali teme.

NMF je pogosto uporaben v tekstovnem rudarjenju. V tekstovnem dokumentu se ista beseda lahko pojavi na različnih mestih z različnimi pomeni. Z združevanjem atributov NMF vpelje kontekst, ki je bistven za napovedovalno moč.

Kako deluje?

NMF razčleni multivariabilne podatke s kreiranjem določenega števila atributov, ki ga je določil uporabnik. Vsak atribut je linearna kombinacija originalnege množice atributov. Koeficienti teh linearnih kombinacij so nenegativni. NMF razčleni podatkovno matriko V v produkt dveh matrik nižjega ranga W in H tako, da je V približno enaka W * H. NMF uporablja iterativno proceduro za spremembo začetnih vrednosti W in H tako, da se produkt približa V. Procedura se konča, ko približna vrednost napake konvergira ali, ko je število iteracij doseglo določeno število. Ko apliciramo model, NMF model preslika originalne podatke v novo množico atributov, ki jo je odkril model.



Slika 4-8 Rezultat po ekstrakciji atributov z metodo NMF

5 Metode za strojno učenje (avtomatsko modeliranje) podatkov

5.1 Klasifikacija: Logistična regresija

Logistična regresija spada med posplošene linearne modele (Generalized Linear Models GLM), ki vključujejo in razširjajo razred linearnih modelov.

Linearni modeli predpostavljajo določene omejitve, med katerimi je najpomembnejša ta, da je ciljna spremenljivka (odvisna spremenljivka y) normalno porazdeljena, če obstaja vrednost napovednikov s stalno varianco, ne glede na predvideno odzivno vrednost. Prednost linearnih modelov in njihovih omejitev vključuje računsko preprostost, interpretativno oblika modela in sposobnost izračuna nekaterih diagnostičnih informacij glede kvalitete prilagoditve.

Posplošeni linearni modeli sprostijo tiste omejitve, ki so v praksi pogosto prekršene. Na primer, binarni (ja/ne ali 0/1) odzivi nimajo enake variance skozi razrede. Še več, vsota izrazov v linearnem modelu ima navadno lahko zelo velik razpon in vključuje zelo negativne in zelo pozitivne vrednosti. Za primer binarnega odziva bi radi, da bi bila verjetnost odziva v razponu [0,1].

Posplošeni linearni modeli prilagodijo odzive, ki kršijo napovedi linearnih modelov, in sicer na dva načina: s povezovalno funkcijo in funkcijo variance. Povezovalna funkcija pretvori ciljni razpon za potencialno –neskončnost na + neskončnost tako, da se lahko ohrani preprosta oblika linearnih modelov. Funkcija variance izraža variance kot funkcijo predvidenega odziva, tako da prilagodi odzive z nestalnimi variancami (kot na primer binarni odzivi).

Oracle Data Mining vključuje dva najbolj popularna člana GLM družine modelov (linearna regresija, logistična regresija).

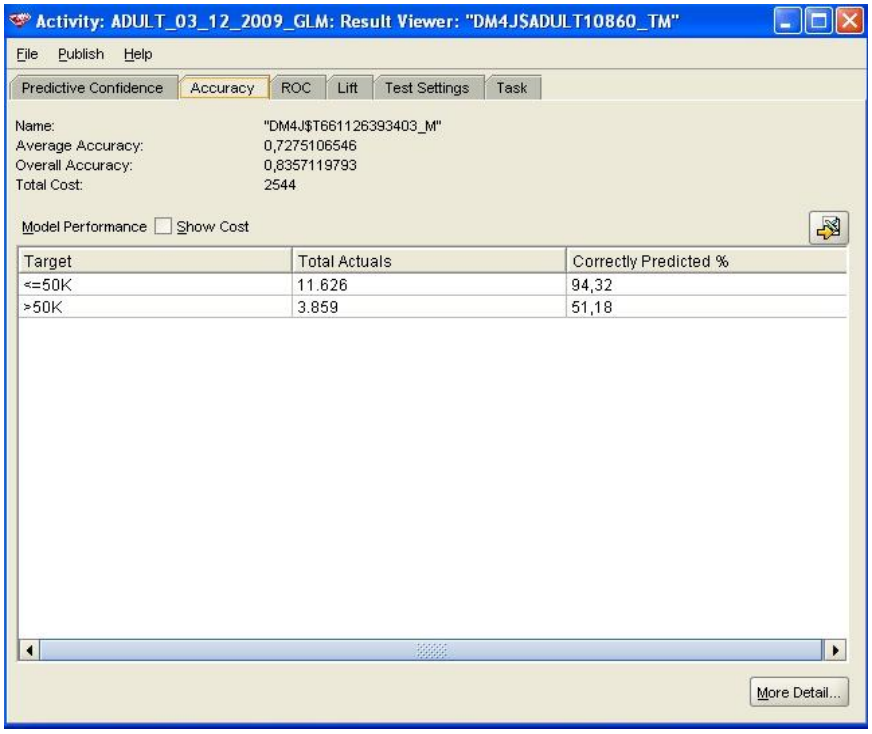
GLM v Oracle Data Mining

GLM je parametrična tehnika modeliranja. Parametrični modeli predvidijo, kako se bodo podatki distribuirali/porazdeljevali. Ko so prilagojeni, so lahko parametrični modeli bolj učinkoviti od neparametričnih modelov. Izziv pri izdelavi modelov te vrste vključuje ocenjevanje obsega prilagoditev. To je razlog, da je kvalitetna diagnostika ključ do razvijanja kvalitetnih parametričnih modelov.

Oracle Data Mining GLM je posebej prilagojen za upravljanje s podatki z veliko atributi. Algoritem lahko zgradi ali zadane kvalitetne modele, ki uporabljajo tako rekoč neomejeno število napovednikov (atributov). Edine omejitve so tiste, ki jih narekujejo viri sistema.

GLM lahko predvidi meje zaupanja. Za najboljšo napoved ocene in verjetnosti (samo klasifikacija) za vsako vrstico, GLM označi interval znotraj katerega bo napoved (regresija) ali verjetnost (klasifikacija) ležala. Širina intervala je odvisna od natančnosti modela in nivoja zaupanja, ki ga določi uporabnik.

Nivo zaupnosti je merilo kako gotov je model, da bo resnična vrednost ležala znotraj intervala, ki ga je izračunal model. Popularna izbira za nivo zaupanja je 95%. Na primer, model lahko predvidi, da je prihodek zaposlenega 115.000 EUR in da si lahko 95 % prepričan, da leži med 90.000 EUR in 160.000 EUR. Oracle Data Mining podpira 95% zaupanje po osnovnih nastavitvah, vendar se lahko vrednost prilagaja.



Slika 5-1 Rezultat klasifikacije z pomočjo logistične regresije

5.2 Klasifikacija: Naivni Bayes

Uvod

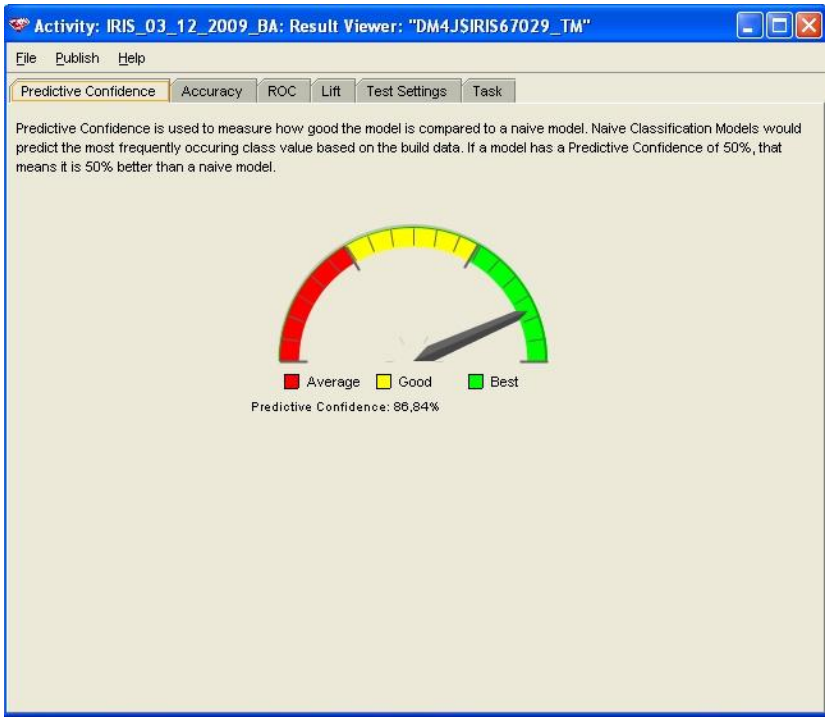
Metoda Naivni Bayes (NB) je ena najpogosteje uporabljenih metod strojnega učenja. Njene prednosti so enostavnost uporabe, hitrost in robustnost. Metoda ima dve večji pomanjkljivosti, ki sta naivnost (predpostavka o neodvisnosti atributov) in težave pri uporabi zveznih atributov. Metoda temelji na oceni pogojnih verjetnosti, ki jih izračunamo za nominalne atribute. Pri zveznih atributih pa se večinoma uporablja kategorizacija (diskretizacija) zveznih atributov.

NB nudi hitro grajenje modela in zadetke za relativno majhne obsege podatkov. Za svoje napovedovanje uporablja Bayesov teorem, ki izpelje pogojno verjetnost iz naslednje formule:

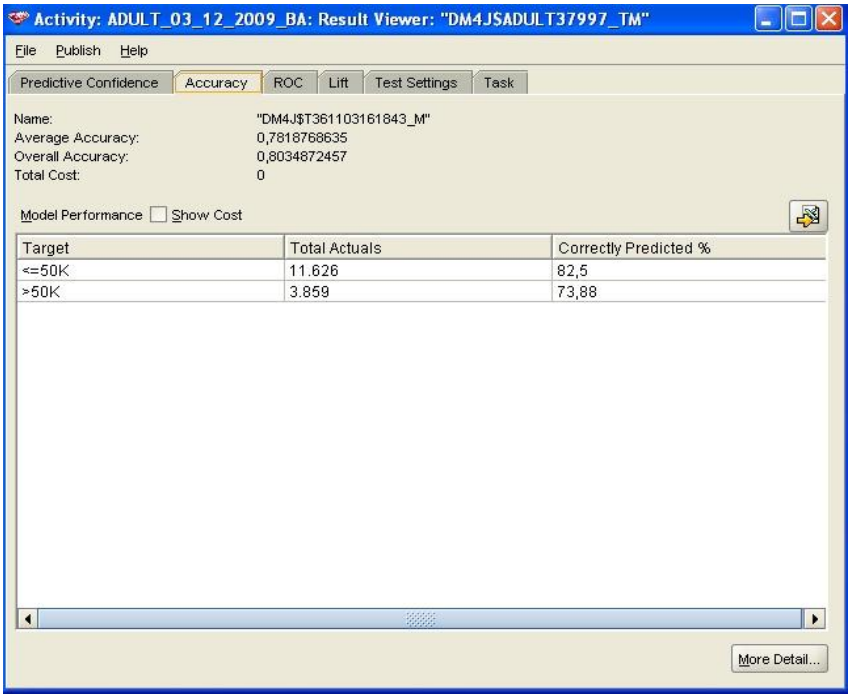
$P(A \mid B) = (P(B \mid A)\, P(A))/P(B)$

Verjetnost dogodka A ob pogoju, da se je zgodil dogodek B, je enaka verjetnosti dogodka B, ob pogoju da se je zgodil dogodek A, pomnoženo z verjetnostjo dogodka A in deljeno z verjetnostjo dogodka B.

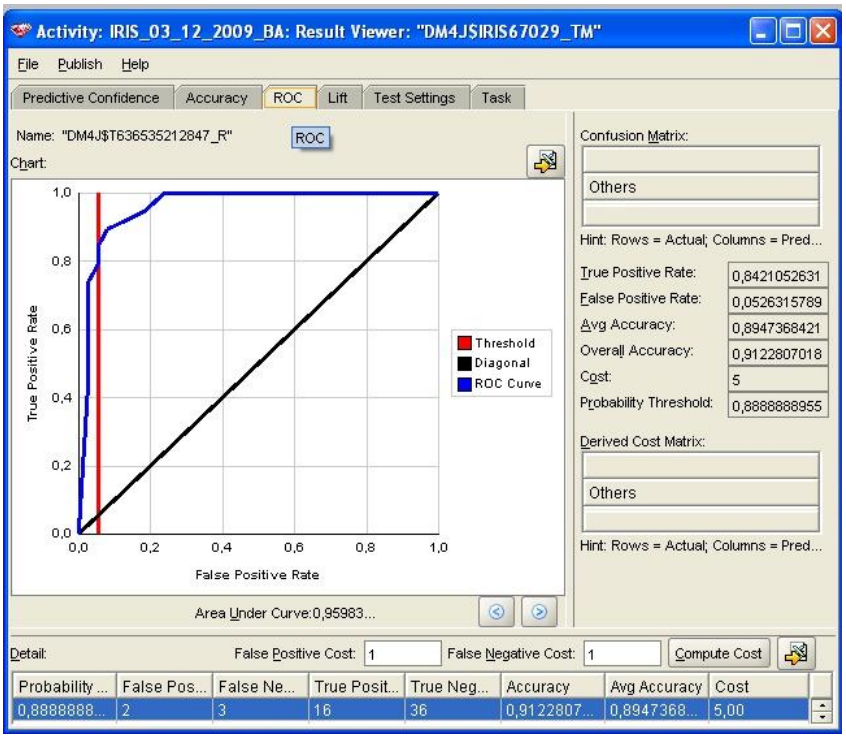
NB predvideva, da je vsak atribut pogojno neodvisen od drugih. V praksi ta predpostavka, tudi kadar je prekršena, bistveno ne zmanjša natančnosti napovedovanja. Uporablja se lahko tako za binarne kot večrazredne klasifikacijske probleme.



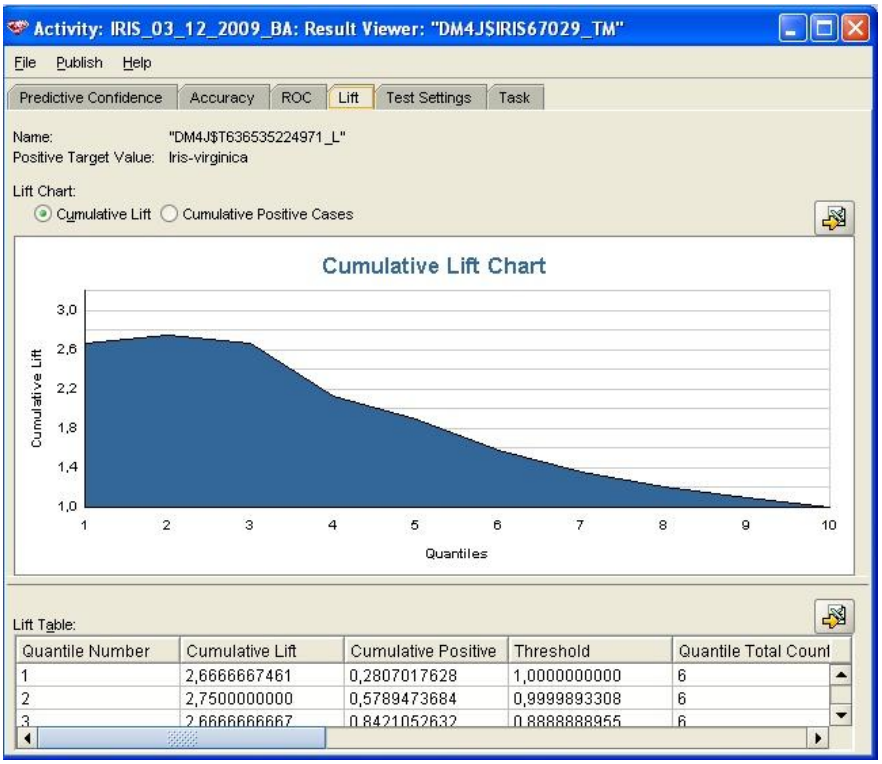
Slika 5-2 Rezultat klasifikacije Naivnega Bayesa (1. zavihek)



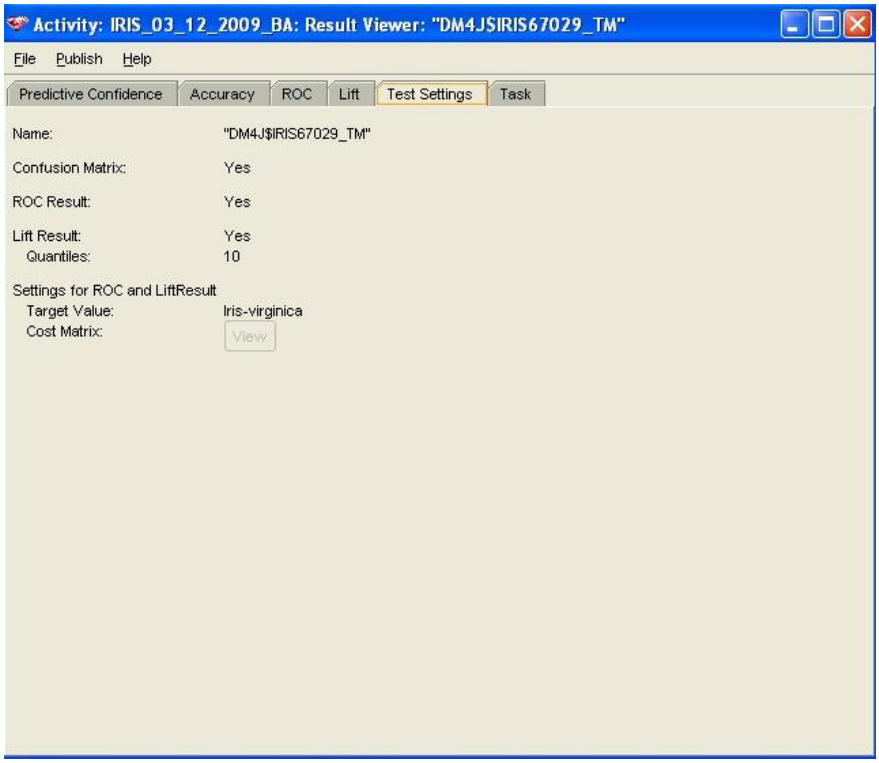
Slika 5-3 Rezultat klasifikacije Naivnega Bayesa (2. zavihek)



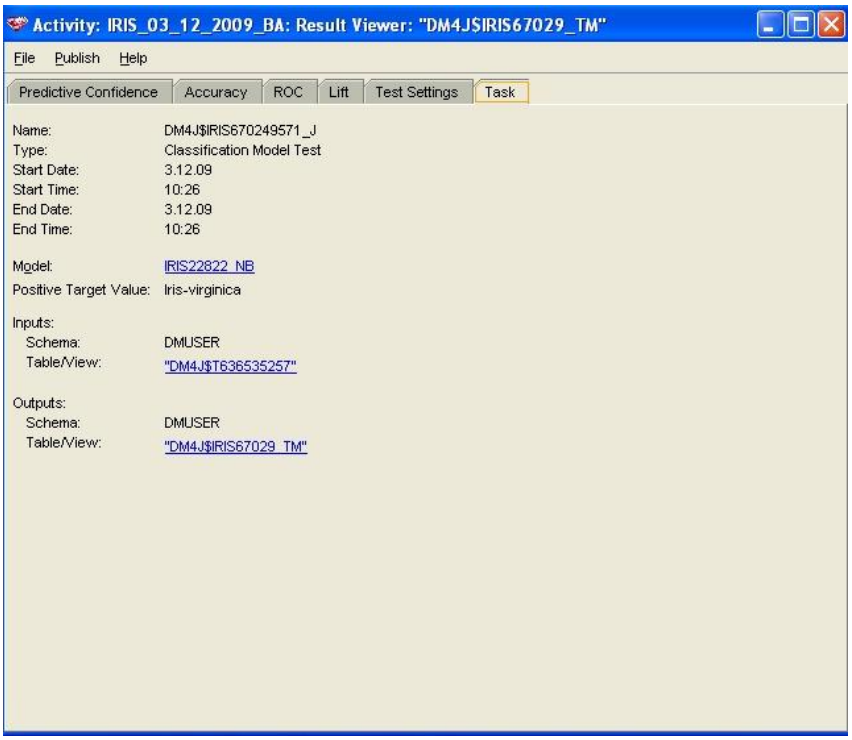
Slika 5-4 Rezultat klasifikacije Naivnega Bayesa (3. zavihek)



Slika 5-5 Rezultat klasifikacije Naivnega Bayesa (4. zavihek)



Slika 5-6 Rezultat klasifikacije Naivnega Bayesa (5. zavihek)



Slika 5-7 Rezultat klasifikacije Naivnega Bayesa (6. zavihek)

5.3 Klasifikacija: Metoda podpornih vektorjev

Metoda podpornih vektorjev (Support Vectior Machines SVM) je zmogljiv, vrhunski algoritem z močnim teoretičnim zaledjem osnovanim na Vapnik - Chervonenkis teoriji. SVM ima močne regulacijske lastnosti. Regulacija se nanaša na posploševanje modela napram novim podatkom.

SVM modeli imajo funkcionalno obliko podobno živčni mreži in radialne osnovne funkcije, obe popularni tehniki podatkovnega rudarjenja. Nobeni od teh dveh algoritmov nimata tako dobro teoretično zasnovanega pristopa k regularizaciji, ki tvori osnovo SVM-ja. Kvaliteta posploševanja in lahkota učenja SVM-ja je daleč nad zmožnostjo katere od bolj tradicionalnih metod. SVM lahko modelira kompleksne, realne probleme kot so tekstovne in slikovne klasifikacije, prepoznavanje človeške ročne pisave ter bioinformatične in biosekvenčne analize.

SVM se dobro odreže na podatkovnih množicah, ki imajo veliko atributov, tudi če je učna množica manjša. Algoritem nima zgornje meje števila atributov, edina omejitev je omejitev strojne opreme.

Oracle Data Mining ima svojo lastno izvedbo SVM-ja, ki izkorišča številne koristi algoritma in kompenzira nekatere omejitve pripadajoče zgradbi algoritma. Oracle Data Mining SVM nas opremlja z skalabilnostjo in uporabnostjo, ki sta potrebni za kvaliteten sistem podatkovnega rudarjenja.

Klasifikacija SVM

Klasifikacija SVM-ja je osnovana na konceptu odločitvenih ravnin, ki določajo odločitvene meje. Odločitvena ravnina je tista, ki loči med množico atributov, ki pripadajo različnim razredom. SVM najde vektorje (»podporne vektorje«), ki definirajo separatorje tako, da dajo najširše ločitve razredov. Klasifikacija SVM podpira binarne in večrazredne ciljne spremenljivke.

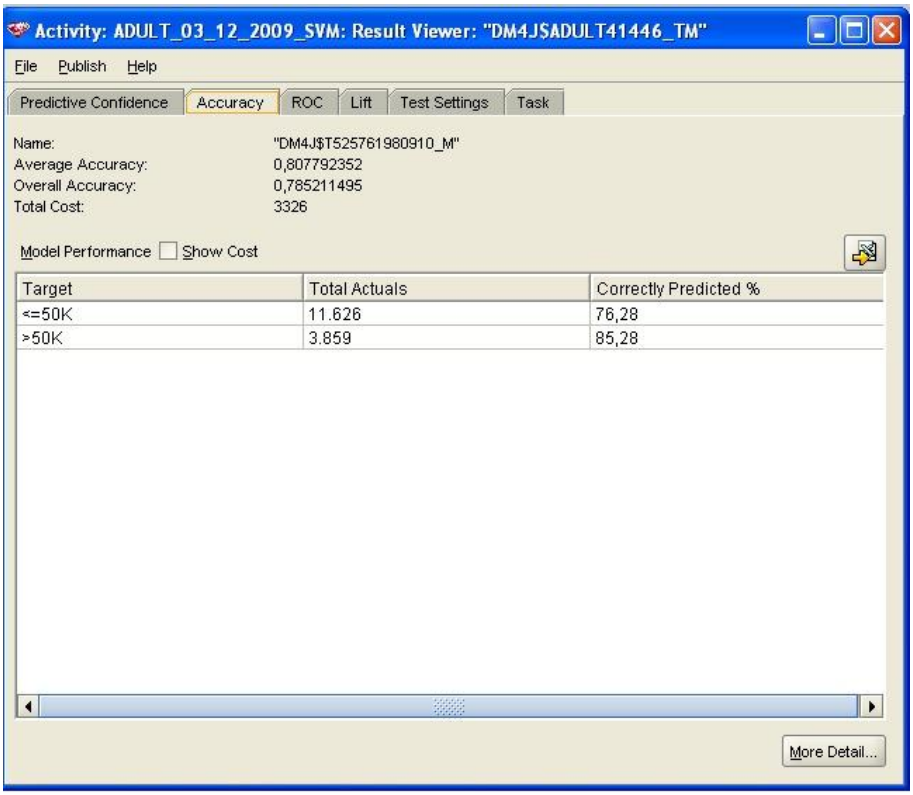
Uteži razredov

Pri klasifikaciji SVM so uteži poševno urejene za določanje relativne pomembnosti vrednosti ciljnih razredov.

SVM modeli so narejeni tako, da dosežejo najboljšo povprečno napoved v vseh razredih. Vendar, če učna množica podatkov ne predstavlja realistične distribucije, lahko vplivamo na model, da nadomestimo tiste vrednosti razredov, ki niso predstavljene v zadostni meri. Če povečamo utež za razred, se mora odstotek pravilnih napovedi za ta razred povečati.

Oracle Data Mining uporablja apriorne verjetnosti za določanje uteži za razred. Za uporabo apriornih verjetnosti na učnem modelu ustvari tabelo le-teh in njeno ime določi kot funkcijo grajenja modela.

Apriorne verjetnosti so povezane z verjetnostnimi modeli za popravljanje prilagajanja vzročnih procesov. SVM uporablja apriorne verjetnosti kot utežene vektorje, s katerimi vplivamo na optimizacijo in dajemo enemu razredu prednost pred drugim.



Slika 5-8 Rezultat klasifikacijske metode SVM

5.4 Klasifikacija: Odločitveno drevo

Uvod

Odločitveno drevo je sestavljeno iz notranjih vozlišč, vej in listov. Vozlišča ustrezajo atributom, veje podmnožicam in listi razredom. Eno odločitveno pravilo ustreza eni poti v drevesu od korena do lista. Pri tem so pogoji, ki jih srečamo na poti, konjuktivno povezani. Pri grajenju je ustavitveni pogoj lahko:

- Bodisi dovolj »čista« učna množica (npr. vsi ali večina primerov iz istega razreda)
- Bodisi premalo učnih primerov za zanesljivo nadaljevanje gradnje drevesa
- Ali pa je zmanjkalo (dobrih) atributov

Ključnega pomena je izbira »najboljšega atributa«. Za izbiro atributa se najpogosteje uporabljajo mere: informacijski prispevek, razmerje informacijskega prispevka, Gini-indeks in ReliefF.

Odločitveno drevo predstavlja klasifikacijsko funkcijo, ki je hkrati simbolični opis in povzetek zakonitosti v dani problemski domeni. Zato je drevo ponavadi zanimivo za strokovnjaka iz dane problemske domene, saj lahko iz drevesa razbere določene zakonitosti in strukturo.

List drevesa vsebuje informacijo o številu učnih primerov iz posameznih razredov, ki kaže na zanesljivost ocene verjetnostne porazdelitve razredov. Ker je zanesljivost ocene kvalitete atributa odvisna od števila učnih primerov, ni dobro, da se učna množica prehitro razdeli na majhne podmnožice. Zato algoritmi pogosto gradijo binarno odločitveno drevo. Gradnja binarnega odločitvenega drevesa omogoča, da lahko uporabimo za ocenjevanje atributov tudi meri informacijski prispevek in Gini-indeks, ki precenjujeta večvrednostne atribute. Z binarizacijo dobimo manjša drevesa, ki imajo ponavadi tudi boljšo klasifikacijsko točnost[5].

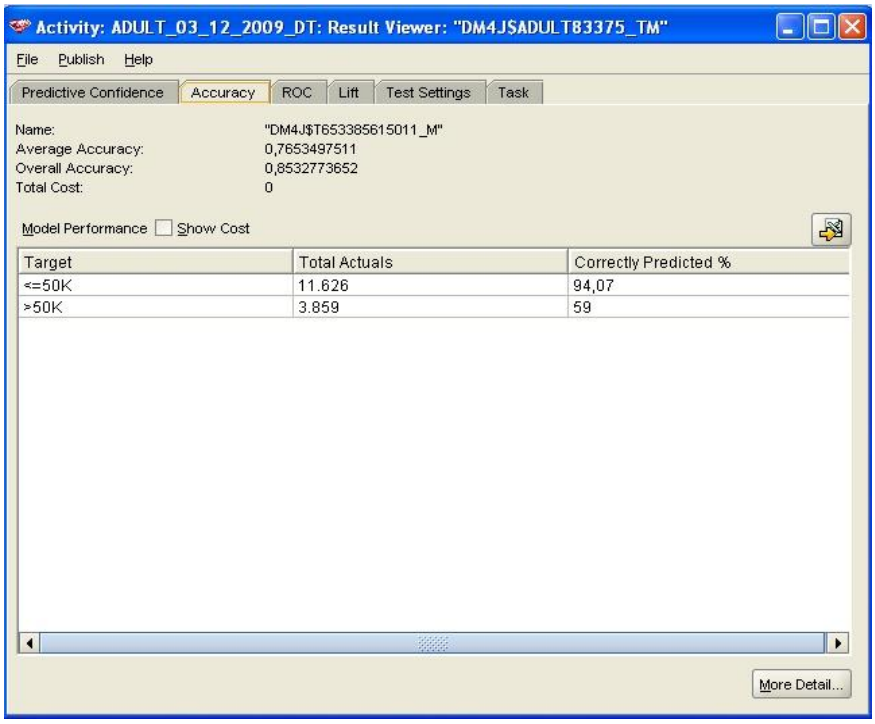
Prednosti odločitvenih dreves

Algoritem Odločitvenega drevesa zgradi točne in razlagalne modele z relativno majhno pomočjo uporabnika. Algoritem se lahko uporablja tako za binarno kot za večrazredno klasifikacijo problema. Je hiter tako ob izgradnji kot ob uporabi[7].

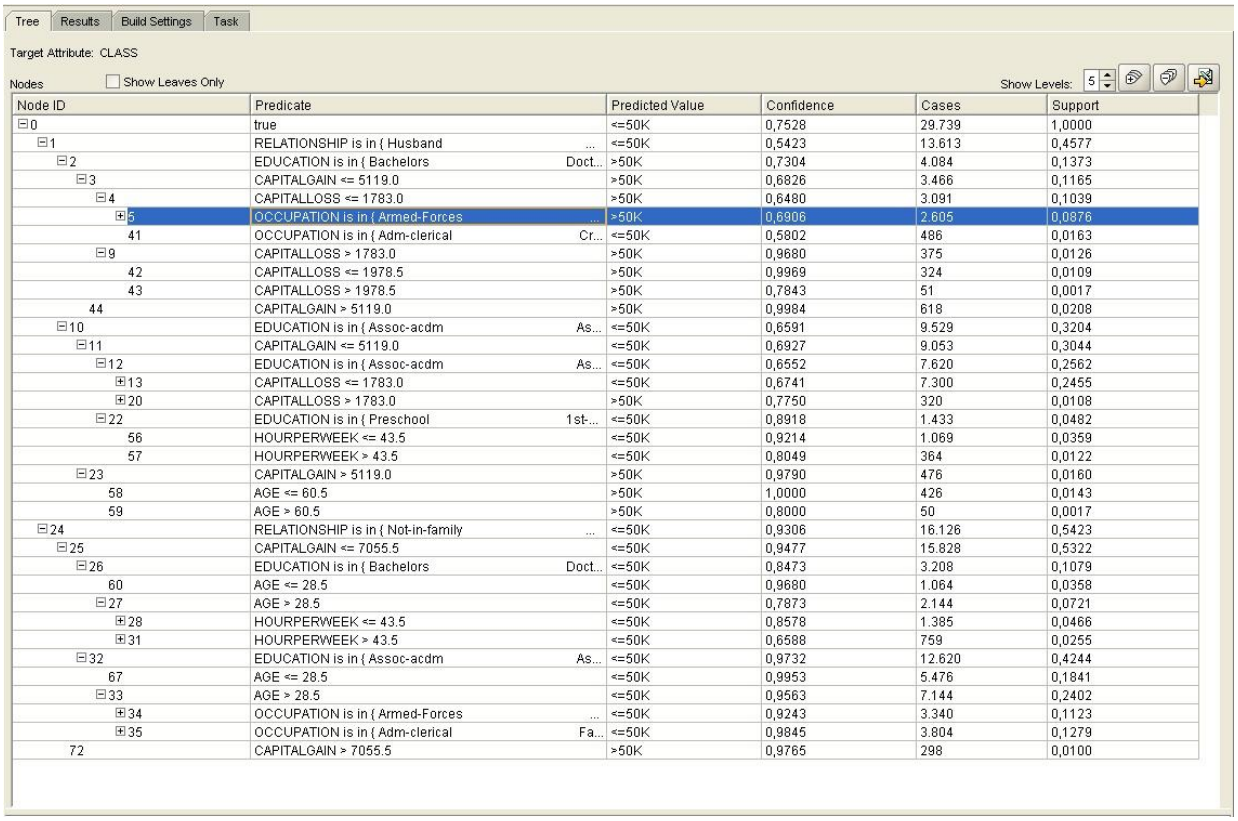
Preprečitev prevelikega ujemanja

V osnovi lahko algoritem zgradi vsako vejo drevesa tako globoko, da popolnoma pravilno klasificira vse učne primere. Čeprav je to včasih dobra strategija, lahko vodi v težave, če je v podatkih šum, ali kadar je učna množica podatkov premajhna za izdelavo reprezentativnega vzorca podatkovne množice. V teh primerih preprost algoritem naredi drevo, ki se preveč ujema z učno množico. To je stanje, kjer lahko model natančno napove z podatki uporabljenimi za izgradnjo modela, a se slabo odreže, če ga uporabimo na novih podatkih.

Da bi to preprečili, ODM podpira avtomatično rezanje in nastavljivo mejo pogojev, ki nadzorujejo rast drevesa. Ko je pogoju zadoščeno, meje pogojev preprečujejo nadaljna deljenja. Rezanje odstranjuje veje, ki imajo nepomemben napovedovalen vpliv.



Slika 5-9 Rezultat klasifikacije Odločitvenega drevesa



Node ID	Predicate	Predicted Value	Confidence	Cases	Support
⊟0	true	<=50K	0,7528	29.739	1,0000
⊟1	RELATIONSHIP is in { Husband ...	<=50K	0,5423	13.613	0,4577
⊟2	EDUCATION is in { Bachelors Doct...	>50K	0,7304	4.084	0,1373
⊟3	CAPITALGAIN <= 5119.0	>50K	0,6826	3.466	0,1165
⊟4	CAPITALLOSS <= 1783.0	>50K	0,6480	3.091	0,1039
⊞5	OCCUPATION is in { Armed-Forces ...	>50K	0,6906	2.605	0,0876
41	OCCUPATION is in { Adm-clerical Cr...	<=50K	0,5802	486	0,0163
⊟9	CAPITALLOSS > 1783.0	>50K	0,9680	375	0,0126
42	CAPITALLOSS <= 1978.5	>50K	0,9969	324	0,0109
43	CAPITALLOSS > 1978.5	>50K	0,7843	51	0,0017
44	CAPITALGAIN > 5119.0	>50K	0,9984	618	0,0208
⊟10	EDUCATION is in { Assoc-acdm As...	<=50K	0,6591	9.529	0,3204
⊟11	CAPITALGAIN <= 5119.0	<=50K	0,6927	9.053	0,3044
⊟12	EDUCATION is in { Assoc-acdm As...	<=50K	0,6552	7.620	0,2562
⊞13	CAPITALLOSS <= 1783.0	<=50K	0,6741	7.300	0,2455
⊞20	CAPITALLOSS > 1783.0	>50K	0,7750	320	0,0108
⊟22	EDUCATION is in { Preschool 1st-...	<=50K	0,8918	1.433	0,0482
56	HOURPERWEEK <= 43.5	<=50K	0,9214	1.069	0,0359
57	HOURPERWEEK > 43.5	<=50K	0,8049	364	0,0122
⊟23	CAPITALGAIN > 5119.0	>50K	0,9790	476	0,0160
58	AGE <= 60.5	>50K	1,0000	426	0,0143
59	AGE > 60.5	>50K	0,8000	50	0,0017
⊟24	RELATIONSHIP is in { Not-in-family ...	<=50K	0,9306	16.126	0,5423
⊟25	CAPITALGAIN <= 7055.5	<=50K	0,9477	15.828	0,5322
⊟26	EDUCATION is in { Bachelors Doct...	<=50K	0,8473	3.208	0,1079
60	AGE <= 28.5	<=50K	0,9680	1.064	0,0358
⊟27	AGE > 28.5	<=50K	0,7873	2.144	0,0721
⊞28	HOURPERWEEK <= 43.5	<=50K	0,8578	1.385	0,0466
⊞31	HOURPERWEEK > 43.5	<=50K	0,6588	759	0,0255
⊟32	EDUCATION is in { Assoc-acdm As...	<=50K	0,9732	12.620	0,4244
67	AGE <= 28.5	<=50K	0,9953	5.476	0,1841
⊟33	AGE > 28.5	<=50K	0,9563	7.144	0,2402
⊞34	OCCUPATION is in { Armed-Forces ...	<=50K	0,9243	3.340	0,1123
⊞35	OCCUPATION is in { Adm-clerical Fa...	<=50K	0,9845	3.804	0,1279
72	CAPITALGAIN > 7055.5	>50K	0,9765	298	0,0100

Slika 5-10 Odločitveno drevo

5.5 Regresija

Regresija je nadzorovana funkcija rudarjenja za predpostavljanje vezanega, številčnga razreda.

O regresiji

Regresija je funkcija podatkovnega rudarjenja, ki predvideva število. Dobiček, prodaja, obrestne mere, vrednosti hiš, kvadrature, temperature ali razdalja se lahko predvidijo z uporabo regresijskih metod. Na primer, regresijski model se lahko uporabi za predvidevanje vrednosti hiše na osnovi lokacije, števila sob, velikosti zemljišča in ostalih dejavnikov.

Regresijska naloga se prične s podatkovno celoto, kjer so ciljne vrednosti znane. Tak primer je regresijski model, ki predvideva, da bi bile lahko vrednosti hiš predvidene glede na opazovane podatke velikega števila hiš skozi daljše časovno obdobje. Poleg vrednosti lahko podatki spremljajo starost hiše, kvadraturo, število sob, davke, šolska območja, oddaljenost od nakupovalnih središč in tako naprej. Vrednost hiše bi bil ciljna spremenljivka, ostale lastnosti pa bi bili predikatorji, podatki za vsako hišo pa bi tvorili primer.

V procesu grajenja modela regresijski algoritem ocenjuje vrednost ciljne spremenljivke kot funkcijo predikatorjev za vsak primer podatkov, na katerih gradimo model. Odnosi med predikatorji in ciljno spremenljivko so povzeti v modelu, ki se lahko uporabi za drugo podatkovno celoto, kjer vrednosti ciljne spremenljivke niso znane.

Regresijski modeli so testirani s seštevanjem različnih statistik, ki merijo razliko med predvidevanimi in pričakovanimi. Historični podatki za regresijski projekt so navadno razdeljeni v dve skupini podatkov: prvi za grajenje modela, drugi za testiranje modela.

Grajenje modela ima številne aplikacije v analizi trendov, načrtovanju podjetništva, marketingu, finančnih napovedih, napovedovanju časovnih vrst.

Kako regresija deluje?

Razumevanje matematike, ki je uporabljena pri analizi regresije, za razvoj in uporabo kvalitetnih regresijskih modelov za podatkovno rudarjenje ni potrebno. Kljub vsemu, pa je koristno razumeti nekaj osnovnih konceptov.

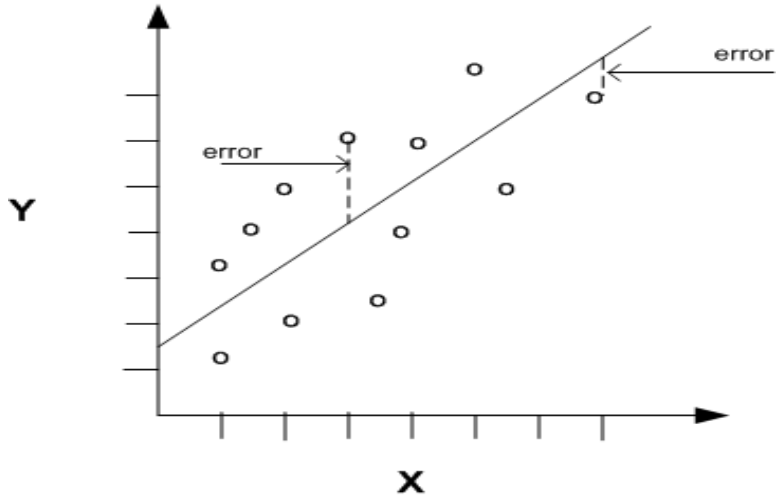
Z regresijsko analizo želimo določiti vrednosti parametrov za funkcijo, ki povzročijo, da funkcija kar najbolje ustreza dani podatkovni množici. Naslednja enačba izraža te odnose in simbole. Prikazuje, da je regresija proces ocenjevanja vrednosti neprestane ciljne spremenljivke (y) funkcije (F), enega ali več predikatorjev (x_1 , x_2 , ..., x_n), množici parametrov (θ_1 , θ_2 , ..., θ_n) in stopnjo napake (e).

$y = F(x,\theta) + e$

Prediktorji (atributi) so neodvisne spremenljivke, od katerih je odvisna vrednost odvisne (ciljne) spremenljivke. Regresijski parametri so prav tako znani kot regresijski koeficienti. Proces učenja regresijskega modela vključuje iskanje vrednosti parametrov, ki minimalizirajo stopnjo napake, na primer, vsoto kvadratnih napak. Obstajajo različne družine regresijskih funkcij in različni načini merjenja napak.

Linearna regresija

Linearna regresija je tehnika, ki se jo lahko uporabi, če se odnos med predikatorji in ciljno spremenljivko lahko približa ravni črti. Regresija z enim prediktorjem je za prikaz najpreprostejša.



Slika 5-11 Preprosta linearna regresija z enim prediktorjem

Linearna regresija z enim samim prediktorjem je lahko izražena z naslednjo enačbo.

$y = \theta_2 X + \theta_1 + e$

Regresijska parametera v preprosti linearni regresiji sta:

- Nagib črte (θ_2) — kot med podatkovno točko in regresijsko premico
- Sečišče y (θ_1) — točka, kjer x prečka os y (x = 0)

Večrazsežnostna linearna regresija (z več spremenljivkami)

Izraz večrazsežnostna regresija se nanaša na linearno regresijo z dvema ali več prediktorji (x_1, x_2, …, x_n). Ko je uporabljenih več prediktorjev, regresijske premice ni moč predstaviti v dvodimenzionalnem prostoru. Vendar pa je premica lahko sešteta z razširitvijo enačbe za linearno regresijo z enim prediktorjem za vse prediktorje

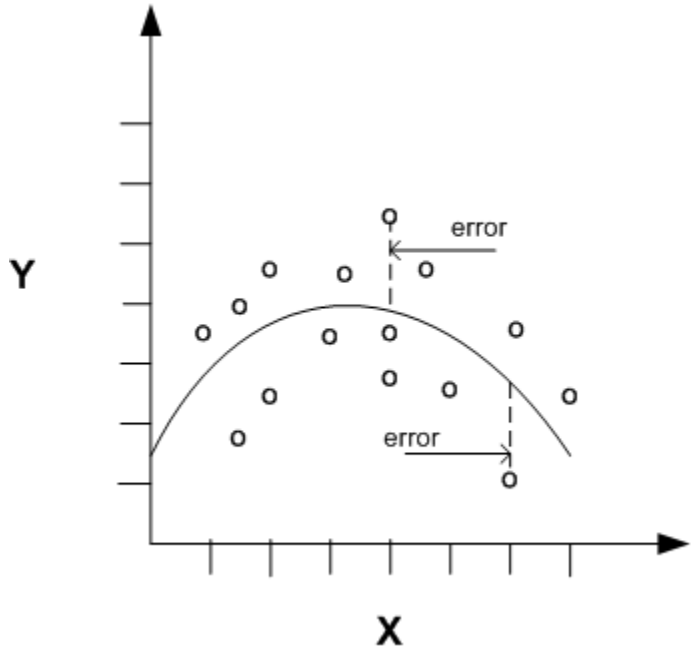
$y = \theta_1 + \theta_2 X_1 + \theta_3 X_2 + ... \theta_n X_{n-1} + e$

Regresijski koeficienti

V večrazsežnostni regresiji so regresijski parametri pogosto imenovani koeficienti. Ko zgradiš model večrazsežnostne regresije, algoritem izračuna koeficient za vsak prediktor, ki ga model uporablja. Koeficient je merilo učinka prediktorja x na ciljno spremenljivko y. Za analizo regresijskih koeficientov in ocenitev, kako dobro regresijska črta ustreza podatkom, so na voljo številne statistike.

Nelinearna regresija

Pogosto odnos med x in y ne more biti prikazan z ravno črto. V tem primeru se lahko uporabi metodo nelinearne regresije. Nelinearni regresijski modeli definirajo y kot funkcijo x z uporabo enačbe, ki je bolj zapletena kot enačba linearne regresije..



Slika 5-12 Nelinearna regresija z enim indikatorjem

Večrazsežnostna nelinearna regresija

Izraz večrazsežnostna nelinearna regresija se nanaša na nelinearno regresijo z dvemi ali več prediktorji (x_1, x_2, …, x_n). Ko se uporabi več prediktorjev, se nelinearnega odnosa ne da prikazati v dvodimenzionalnem prostoru.

Meje intervala zaupanja

Regresijski model napove vrednost ciljne spremenljivke za vsak primer vhodnih podatkov. Poleg napovedi lahko nekateri regresijski algoritmi identificirajo meje intervalov zaupanja, ki so zgornje in spodnje meje intervala, znotraj katerega predvidena vrednost najverjetneje leži.

Ko je zgrajen model za napovedovanje z danim zaupanjem, bo interval zaupanja izračunan skupaj z napovedjo. Na primer, model lahko predvideva, da bo vrednost hiše 500,000 EUR s 95% verjetnostjo, da bo vrednost med 475,000 EUR in 525,000 EUR

Regresijske statistike

Koren povprečne kvadratne napake in povprečna absolutna napaka sta pogosto uporabljeni statistiki za ocenjevanje splošne kvalitete regresijskega modela. Na voljo so različne statistike, odvisno od regresijskih metod, ki jih uporablja algoritem.

Koren povprečne kvadratne napake

Koren povprečne kvadratne napake (Root Mean Squared Error (RMSE)) je kvadratni koren povprečne kvadratne razdalje od podatkovne točke do premice.

Formula prikazuje RMSE v matematičnih simbolih. Večji simbol sigme prikazuje seštevek; j predstavlja trenutni predikator in n predstavlja število predikatorjev.

Formula 5-1: RMSE = $\sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \bar{y_j})^2}$

Povprečna absolutna napaka

Povprečna absolutna napaka (Mean Absolute Error (MAE)) je povprečje absolutne vrednosti napake. MAE je zelo podobna RMSE, a je manj občutljiva za velike napake. Formula prikazuje MAE z matematičnimi simboli. Velik simbole sigme predstavlja vsoto; j prestavlja trenuten predikator in n predstavlja število predikatorjev.

Formula 5-2: $MAE = \frac{1}{n}\sum_{j=1}^{n}|Y_j - \bar{Y_j}|$

Regresijski algoritmi

Oracle Data Mining podpira dva regresijska algoritma. Oba algoritma sta ustrezna za podatkovne množice, ki imajo veliko dimenzionalnost (veliko lastnosti).

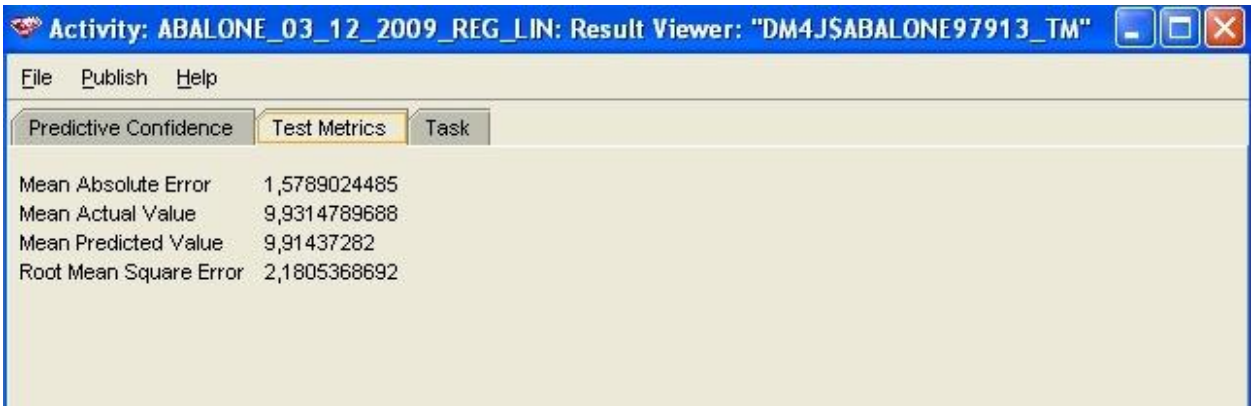
- Posplošeni linearni modeli (GLM)

  GLM je popularna statistična metoda za linearno modeliranje. ODM implementira GLM za regresijo in binarno klasifikacijo.

- Metoda Podpornih Vektorjev (SVM)

  SVM je vpliven, zapleten algoritem za linearno nelinearno regresijo. ODM implementira SVM za regresijo in klasifikacijo ter iskanje izjem.

  SVM regresija podpira dve jedri: Gaussovo jedro za nelinearno regresijo in linearno jedro za linearno regresijo. SVM podpira tudi aktivno učenje.



Slika 5-13 Rezultat linearne regresije

5.6 Primerjava metod (Weka proti Oracle Data Mining)

Metoda	Oracle	Weka
Odločitveno drevo	Decision tree	J48
Naivni Bayes	Naive Baayes	NaiveBayes
Metoda podpornih vektorjev	Support Vector Machines	SMO
Logistična regresija*	Logistic Regression	Logistic
Linearna regresija	Linear Regression (GLM)	LinearRegression
SVM regresija	Support Vector Machines	SMOreg

Tabela 5-1 primerjava metod

*Logistična regresija v Oraclu lahko klasificira le binarne probleme.

6 Testiranje

6.1 Testna metodologija

Za vsako podatkovno množico sem dobil eno ali več podatkovnih datotek. Nekatere so bile razdeljene na učno in testno množico. Te sem najprej združil v eno. Vse datoteke so bile tipa csv (Comma Separated Values). Vse sem uvozil v Accessovo podatkovno bazo, kjer sem , če je bilo to potrebno, dodal primarni ključ. Pregledano tabelo sem skupaj s ključem izvozil nazaj v datoteko tipa csv.

Weka

Za delo z aplikacijo Weka sem vsako podatkovno datoteko najprej spremenil v tip arff. To je Wekin podatkovni tip, za katerega tudi omogoča pretvarjanje iz csv. Nekaj datotek sem, zavoljo testiranja, spremenil ročno in ugotovil, da ni težavno. V Weki sem najprej uporabljal način Explorer, kjer sem se spoznaval z okoljem, metodami, rezultati itd. Za samo testiranje sem uporabljal okolje Experimenter, kjer sem si izbral 4 klasifilacijske oz 2 regresijske metode, odvisno od problema, ter podatkovno množico, nad katero sem želel izvesti vse operacije. Na koncu sem si izbral še datoteko, kamor želim da Weka zapiše vse rezultate. Med nastavitvami samega modeliranja sem dodal le delitev podatkovne množice na učno in testno v razmerju 66% proti 33%, vse ostalo pa sem pustil privzeto. Rezultati so bili že takoj zelo dobri, tako da ostalih stvari ni bilo potrebno spreminjati.

Oracle Data Miner

Pri Oraclu sem moral najprej ročno ustvariti tabelo ter nato uvoziti podatke. Po pregledu podatkov je bilo uvažanje zaključeno. V Oracle Data Minerju sem si najprej izbral podatkovno tabelo, ki sem jo želel uporabiti za modeliranje.V meniju sem si izbral metodo, ID tabele, ciljno spremenljivko (razred) ter nastavitve metode oz. algoritma. Na začetku sem pustil privzete nastavitve, a se v nekaterih primerih niso najbolj izkazale in so bili zato potrebni manjši popravki. Omejitev Oracle Data Minerja je to, da ne pozna ostalih načinov modeliranja razen razbitja množice na učni in testni del. Tudi tu sem jo razdelil v razmerju 66% proti 33%. Rezultate je shranil v izhodno tabelo, lahko pa jih tudi izvoziš kot excelov document. V Oracle Data Minerju imaš shranjene vse aktivnosti tako, da se lahko vračaš nazaj in pregledueš tako rezultate kot nastavitve.

6.2 Testni podatki

6.2.1 Abalone

Osnovni podatki

št. primerov: 4177
razpon vrednosti: 1-29
standardna deviacija: 3.224

Vir

Podatki prihajajo iz izvirne študije:

Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn and Wes B Ford (1994) "The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait", Sea Fisheries Division, Technical Report No. 48 (ISSN 1034-3288). Podatke je prispeval Sam Waugh, Oddelek za računalništvo, Univerza v Tasmaniji.

Opis

V podatkovnem setu Abalone (morsko uho) napovedujemo starost morskih ušes na podlagi fizičnih mer. Starost morskega ušesa je določena z razrezom lupine skozi storž in štetja krogov skozi mikroskop. Dolgočasen in dolgotrajen proces.

Druge meritve, ki jih lažje dobimo, se uporabljajo za napoved starosti (št. krogov). Dodatne informacije, kot so vremenski vzorci in lokacija, so lahko potrebne za rešitev problema.

Atributi

Sex – spol (diskreten) {M, F, in I (infant) }
Length – Dolžina najdaljše lupine (zvezen) mm
Diameter - Premer (zvezen) mm
Height – Višina (zvezen) mm
Whole weight – Celotna teža (zvezen) v gramih
Shucked weight – Teža mesa (zvezen) v gramih
Viscera weight – Teža drobovja (zvezen) v gramih
Shell weight – Teža lupine (zvezen) v gramih
Rings – Število obročev (celo število)

6.2.2 Adult

Osnovni Podatki

Št. primerov: 48842
Št. razredov: 2
Št. atributov: 14
Porazdelitev po razredih: <=50k (69%) , >50k (31%)

Vir

Izločevanje je opravil Barry Becker iz popisa prebivalsta l.1994. Set sprejemljivo čistih podatkov je bilo pripravljenih z uporabo naslednjih pogojev: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&&(HRSWK>0)).

Opis

V podatkovnem setu Adult (Odrasli) napovedujemo, ali oseba zasluži več kot 50 000 $ na leto.

Atributi

>50K, <=50K.
age: continuous.
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: continuous.
education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.
marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Female, Male.
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.
native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic …

6.2.3 Breast Cancer Wisconsin

Osnovni podatki

št. primerov: 645
št razredov 2
št. Atributov: 10
porazdelitev po razredih: 2 (64%), 4 (36%)

Vir

Podatkovni set so ustvarili Dr. William H. Wolber, Olvi L. Mangasarian W. Nick Street iz Univerze v Wsiconsinu. Nick Street pa ga je tudi doniral.

Opis

Lastnosti so izračunane na podlagi digitalizirane slike aspiracije z tanko iglo prsne mase. Opisujejo karakteristike jeder celice na sliki.

Ločilna ravnina, opisana zgoraj, je bila pridobljena z uporabo "Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992] ", klasifikacijske metode, ki uporablja linearno programiranje za izdelavo odločitvenega drevesa. Pomembne lastnosti so bile izbrane z uporabo napornega iskanja v prostoru 1-4 lastnosti in 1-3 ločilne ravnine.

Linearni program, uporabljen za pridobitev ločilne ravnine v 3-dimenzionalem prostoru, je opisan v [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

6.2.4 Car

Osnovni podatki

št. primerov: 1728
št. razredov: 4
št. atributov: 6
porazdelitev po razredih: unacc (70%), acc (22%), vgood(4%), good (4%)

Vir

Podatkovni set je ustvaril Marko Bohanec, doniral pa ga je skupaj z Blažem Zupanom, oba iz Inštitua Jožef Štefan, Ljubljana.

Opis

Podatkovni set Ocenjevanje avtomobila je bil izpeljan iz preprostega hiearhičnega odločitvenega modela, izvirno razvitega za demonstracijo DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990. Model ocenjuje avtomobile na podlagi naslednje strukture:

CAR sprejemljivost avtomobila
. PRICE celotna cena
. . buying nakupna cena
. . maintance cena vzdrževanja
. TECH tehnične lastnosti
. . COMFORT udobje
. . . doors število vrat
. . . persons število oseb
. . . lug_boot velikost prtljažnika
. . safety ocena varnosti avtomobila

Zaradi znane osnovne strukture je lahko ta podatkovni set še posebej uporaben za testiranje konstruktivne indukcije in metod odkrivanja struktur.

6.2.5 Forest Fires

Osnovni podatki

št. primerov 517
št. atributov: 13
razpon vrednosti: 0 - 1090.84
standardna deviacija: 63.656

Vir

Podatke sta sestavila in donirala Paulo Cortez ter Aníbal Morais, oddelek za informacijske sisteme, Univerza Minho, Portugalska.

Opis

To je težak regresijski problem, katerega cilj je napovedati pogorel prostor v gozdnem požaru. Podatki so iz severovzhodne regije na Portugalskem. V preteklih testiranjih se je najbolj izkazal algoritem SVM.

Atributi

1. X - x-os prostorska kooridnata znotraj zemljevida Montesinho parka: 1 do 9
2. Y - y-os prostorska kooridnata znotraj zemljevida Montesinho parka: 2 do 9
3. month – mesec v letu: 'jan' do 'dec'
4. day – dan v tednu: 'mon' do 'sun'
5. FFMC - FFMC indeks iz sistema FWI: 18.7 do 96.20
6. DMC - DMC indeks iz sistema FWI: 1.1 do 291.3
7. DC - DC indeks iz sistema FWI: 7.9 do 860.6
8. ISI - ISI indeks iz sistema FWI : 0.0 do 56.10
9. temp – temperature v stopinjah celzija: 2.2 do 33.30
10. RH – relativna vlažnost v %: 15.0 do 100
11. wind – hitrost vetra v km/h: 0.40 do 9.40
12. rain - dež v mm/m2 : 0.0 do 6.4
13. area – pogorel prostor v gozdu(v ha): 0.00 do 1090.84

6.2.6 Iris

Osnovni podatki

št. primerov: 150
št. razredov: 3
št. atributov: 4
porazdelitev po razredih: Iris-setosa(33,3%), Iris-versiciolor(33,3%), Iris-virginica(33,3%)

Vir

Avtor tega slavnega podatkovnega seta je R.A. Fisher, ki je podatke zbral l. 1936. Podatkovni set je doniral Michael Marshall l.1988.

Opis

To je najbrž najbolj znan podatkovni set v literaturi prepoznavanja vzorcev. Set vsebuje 3 razrede po 50 primerov, kjer se vsak razred sklicuje na tip rastline perunike. Gre za zelo enostavno problemsko domeno.

Atributi

1. Sepal_length - dolžina venčnega lista v cm
2. Sepal_width - širina venčnega lista v cm
3. Petal_length - dolžina cvetnega lista v cm
4. Petal_width - širina cvetnega lista v cm

5. class:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

6.2.7 Spect

Osnovni podatki

št. primerov: 267
št. razredov: 2
št. atributov:23
porazdelitev po razredih: 0 (21%), 1(79%)

Vir

Podatkovni set so ustvarili Krzysztof J. Cios, Lukasz A. Kurgan iz univerze v Coloradu, ter Lucy S. Goodenday z medicinske fakultete iz Ohia. Donirala sta ga Lukasz A.Kurgan ter Krzysztof J. Cios.

Opis:

Podatki o »cardiac Single Proton Emission Computed Tomography" (SPECT) slikah. Vsak pacient je lahko klasificiran bodisi kot normalen bodisi kot abnormalen. Podatkovni set vključuje podatke o 267 SPECT slikah. Algoritem CLIP3, ki zgradi pravila, doseže 84% točnost pri napovedi razreda.

Atributi

1. OVERALL_DIAGNOSIS: 0,1 (razredni atribut, binaren)
2. F1: 0,1 (delna diagnoza 1, binaren)
3. F2: 0,1 (delna diagnoza 2, binaren)
4. F3: 0,1 (delna diagnoza 3, binaren)
5. F4: 0,1 (delna diagnoza 4, binaren)
6. F5: 0,1 (delna diagnoza 5, binaren)
7. F6: 0,1 (delna diagnoza 6, binaren)
8. F7: 0,1 (delna diagnoza 7, binaren)
9. F8: 0,1 (delna diagnoza 8, binaren)
10. F9: 0,1 (delna diagnoza 9, binaren)
11. F10: 0,1 (delna diagnoza 10, binaren)
12. F11: 0,1 (delna diagnoza 11, binaren)
13. F12: 0,1 (delna diagnoza 12, binaren)
14. F13: 0,1 (delna diagnoza 13, binaren)
15. F14: 0,1 (delna diagnoza 14, binaren)
16. F15: 0,1 (delna diagnoza 15, binaren)
17. F16: 0,1 (delna diagnoza 16, binaren)
18. F17: 0,1 (delna diagnoza 17, binaren)
19. F18: 0,1 (delna diagnoza 18, binaren)
20. F19: 0,1 (delna diagnoza 19, binaren)
21. F20: 0,1 (delna diagnoza 20, binaren)
22. F21: 0,1 (delna diagnoza 21, binaren)
23. F22: 0,1 (delna diagnoza 22, binaren)

6.2.8 Tic tac toe

Osnovni podatki

št. primerov: 958
št. razredov: 2
št. atributov: 9
porazdelitev po razredih: positive (65%), negative (35%)

Vir

Podatkovni set je ustvaril in doniral David W. Aha.

Opis

Podatkovni set vključuje vse možne kombinacije na koncu igre tic-tac-toe, kjer je »X« prvi igralec na potezi. Ciljni atribut je »Zmaga za X«.

Atributi

1. top-left-square - polje zgoraj levo {x,o,b}
2. top-middle-square - polje zgoraj sredina {x,o,b}
3. top-right-square - polje zgoraj desno {x,o,b}
4. middle-left-square - polje sredina levo {x,o,b}
5. middle-middle-square - polje sredina sredina {x,o,b}
6. middle-right-square - polje sredina desno {x,o,b}
7. bottom-left-square - polje spodaj levo {x,o,b}
8. bottom-middle-square - polje spodaj sredina {x,o,b}
9. bottom-right-square - polje spodaj desno {x,o,b}
10. Class: {positive,negative}

6.2.9 Wine

Osnovni podatki

št. primerov:178
št. razredov: 3
št. atributov:13
porazdelitev po razredih: 1(33%), 2(40%), 3(27%)

Vir

Podatkovni set je ustvarila Forina, M. et al, PARVUS iz inštituta za analizo zdravil in hrane ter tehnologij v Italiji. Set je doniral Stefan Aeberhard.

Opis

Podatkovni set je rezultat kemičnih analiz vin, ki so bila izdelana v isti regiji v Italiji, a izpeljana iz treh različnih kultur. Analiza je ugotovila količino trinajstih sestavin najdenih v vsaki od treh vrst vina.

Atributi

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

14) Class {1,2,3}

Vsi atributi so zvezni.

6.2.10 Yeast

Osnovni podatki

št.primerov: 1484
št. razredov: 10
št. atributov: 8
porazdelitev po razredih: MIT(16%), NUC(29%), CYT(31%),ME1(3%), EXC(2,5%), ME2(3%), ME3(11%), VAC(2,5%), POX(1,5%), ERL(0.5%)

Vir

Podatke je uredil Kenta Nakai, Inštitut za molekularno in celično biologijo, univerza v Osaki, Japonska. Doniral jih je Paul Horton.

Opis

V tem podatkovnem setu je potrebno napovedati kraj proteina.

6.2.11 KDD Cup 1999

Osnovni podatki

Št. primerov: 4 898 431
Št. atributov: 41 (plus ključ in razred)
Št. Razredov: 23

To podatkovno množico so uporabili na konferenci KDD-99 ter tekmovanju The Third International Knowledge Discovery and Data Mining Tools Competition. Naloga na tekmovanju je bila izdelati detektor vdorov, napovedni model, ki bi bil sposoben razlikovati med »slabimi« povezavami oz. vdori in »dobrimi«, normalnimi povezavami.

7 Testiranje uspešnosti metod v primerjavi s sistemom Weka

7.1 Manjše množice

Tabela 7-1 BreastCancerWisconsin (645 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	93.00	4	95.00	2	-2.00	2
Naivni Bayes	95.00	3	94.00	2	1.00	1
Metoda podpornih vektorjev	96.00	4	96.00	2	0.00	2
Logistična regresija	97.00	3	95.00	2	2.00	1

Tabela 7-2 Car Evaluation (1728 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	91.00	8	90.00	2	1.00	6
Naivni Bayes	83.00	3	88.00	2	-5.00	1
Metoda podpornih vektorjev	85.00	4	93.00	3	-8.00	1
Logistična regresija	//	//	94.00	2	//	//

Tabela 7-3 Iris (150 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	91.00	8	100.00	2	-9.00	6
Naivni Bayes	91.00	3	100.00	2	-9.00	1
Metoda podpornih vektorjev	87.00	4	100.00	2	-13.00	2
Logistična regresija	//	//	100.00	2	//	//

Tabela 7-4 Spect (267 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	79.00	4	72.00	2	7.00	2
Naivni Bayes	81.00	3	73.00	2	8.00	1
Metoda podpornih vektorjev	85.00	4	67.00	2	18.00	2
Logistična regresija	84.00	5	66.00	2	18.00	3

Tabela 7-5 Tic Tac Toe (958 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	66.00	3	100.00	2	-34.00	1
Naivni Bayes	69.00	3	99.00	2	-30.00	1
Metoda podpornih vektorjev	98.00	5	100.00	2	-2.00	3
Logistična regresija	98.00	3	100.00	2	-2.00	1

Tabela 7-6 Wine (178 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	87.00	3	96.00	2	-9.00	1
Naivni Bayes	91.00	3	98.00	2	-7.00	1
Metoda podpornih vektorjev	100.00	5	98.00	2	2.00	3
Logistična regresija	0.00	0	98.00	2	-98.00	-2

Tabela 7-7 Yeast (1484 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	56.00	4	53.00	2	3.00	2
Naivni Bayes	51.00	3	62.00	2	-11.00	1
Metoda podpornih vektorjev	59.00	6	61.00	53	-2.00	-47
Logistična regresija	//	//	62.00	3	//	//

Tabela 7-8 Adult (45222 vrstic)

Metoda	Oracle(%)	Oracle(t)	Weka(%)	Weka(t)	Razlika(%)	Razlika(t)
Odločitveno drevo	85.00	16	0.00	0	85.00	16
Naivni Bayes	80.00	8	0.00	0	80.00	8
Metoda podpornih vektorjev	78.00	15	0.00	0	78.00	15
Logistična regresija	83.00	18	0.00	0	83.00	18

Tabela 7-9 Abalone (4177 vrstic)

Metoda	Srednja absolutna napaka (Oracle)	Koren povprečne kvadratne napake (Oracle)	Čas (Oracle)
SVM regresija	1.48	2.1	16
Linearna regresija	1.57	2.1	3
Metoda	Srednja absolutna napaka (Weka)	Koren povprečne kvadratne napake (Weka)	Čas (Weka)
SVM regresija	1.49	2.137	47
Linearna regresija	1.56	2.183	2

Tabela 7-10 Forest fires (517 vrstic)

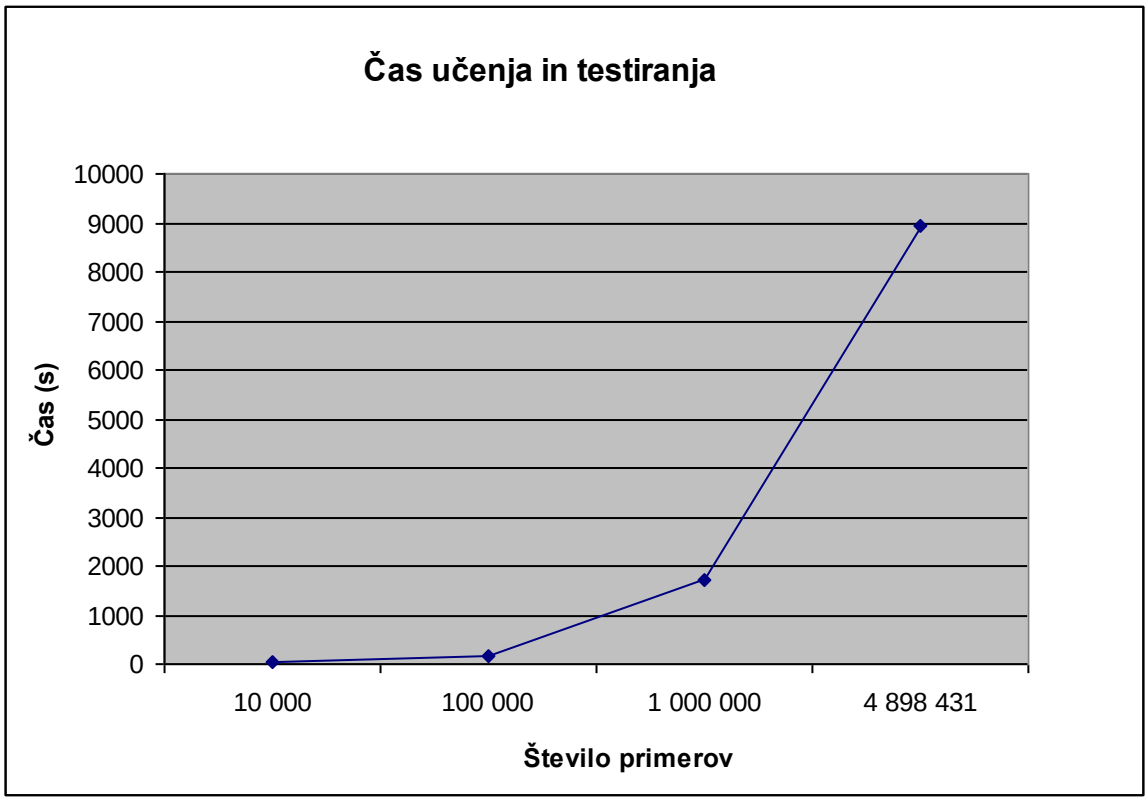
Metoda	Srednja absolutna napaka(Oracle)	Koren povprečne kvadratne napake (Oracle)	Čas (Oracle)
SVM regresija	24.96	58.16	5
Linearna regresija	21.72	58.93	3
Metoda	Srednja absolutna napaka (Weka)	Koren povprečne kvadratne napake (Weka)	Čas (Weka)
SVM regresija	9.82	26.74	6
Linearna regresija	18.86	28.55	2

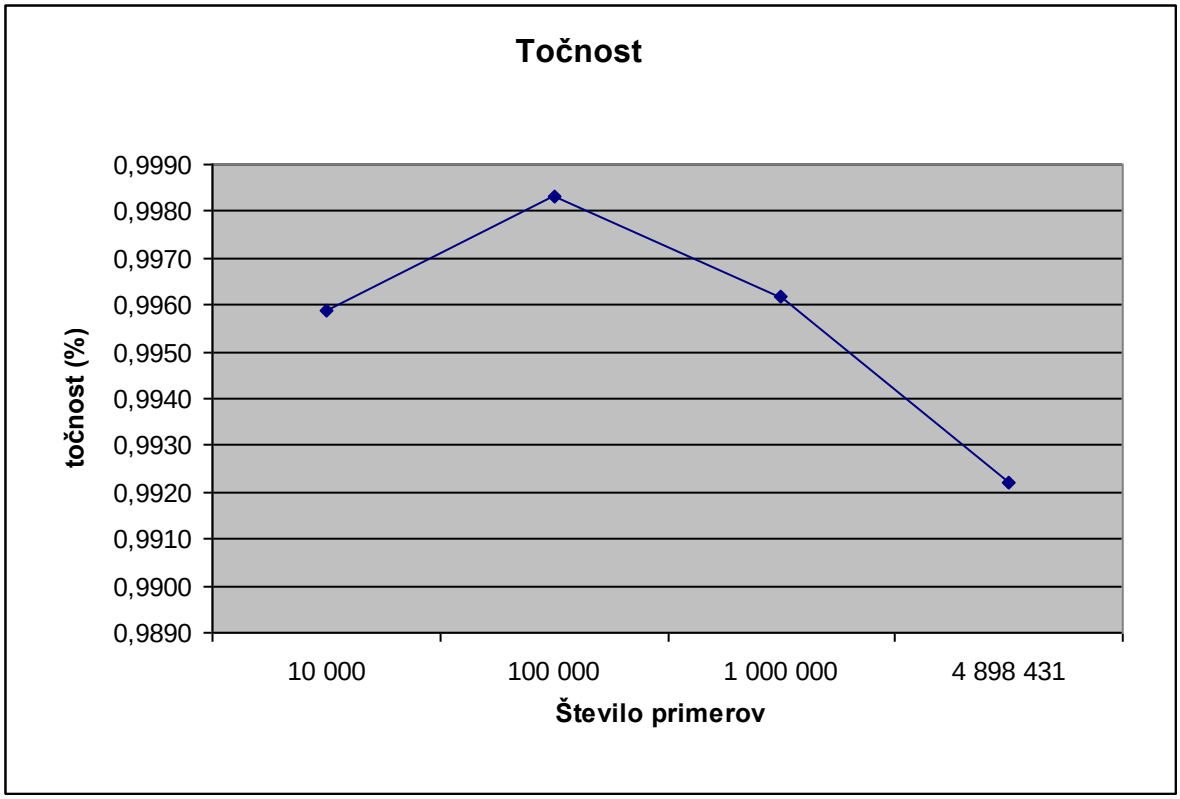
7.2 Večja množica

Želel sem primerjati rezultate tudi pri večjih množicah (večjih od 50 000 vrstic), vendar zaradi omejitev Weke to ni bilo mogoče. Na mojem osebnem računalniku, ki ima 2Gb delovnega pomnilnika to ni bilo moč izvesti, zato sem se osredotočil le na Oracle. Primerjal sem eno množico, večkrat razdeljeno na različne velikosti. Primerjal sem točnost in čas izvajanja pri različnih velikostih.

Tabela 7-11 KDDCUP_99 (4898431 vrstic)

Število podatkov	Celotna točnost(%)	Čas (s)
10 000	0.9959	26
100 000	0.9983	153
1 000 000	0.9962	1701
4 898 431	0.9922	8940




Točnost
0,9990
0,9980
0,9970
0,9960
0,9950
0,9940
0,9930
0,9920
0,9910
0,9900
0,9890
točnost (%)
10 000
100 000
1 000 000
4 898 431
Število primerov

8 Zaključek

Točnost

Zaključim lahko, da je Weka pri manjših problemih v splošnem enostavnejša za uporabo kot Oracle Data Miner, saj njene privzete nastavitve zadostujejo za dobre rezultate. Tudi čas, ki ga potrebuje za izvajanje, je večinoma krajši. Pri uporabi sem opazil, da so pri Weki zelo dobre že osnovne nastavitve, medtem ko Oracle potrebuje precej popravljanja za dosego vsaj približno enakih rezultatov.

Uporabniški vmesnik

Uporabniški vmesnik je tako pri Weki kot pri Oraclu služil svojemu namenu. Oba sta enostavna za uporabo, saj večjih težav pri navajanju na vsakega in prehajanju med obema nisem imel. Izbira podatkovnega vira, algoritma in mesta zapisa rezultatov ne zahtevajo veliko časa. Prednost Wekinega vmesnika je po mojem mnenju možnost dveh načinov delovanja. V prvem lahko hitro menjamo algoritme in podrobne nastavitve algoritma in tako lažje najdemo optimalen model. Te nastavitve lahko potem uporabimo v drugem načinu, kjer izberemo vse podatkovne vire in vse algoritme, Weka pa potem sama izvaja posamezne kombinacije. Pri Oraclu je sicer tudi možno spreminjanje nastavitev, vendar je postopek precej počasnejši kot pri Weki. Prednost ODM-ja pa je dobra preglednost vseh aktivnosti, tako tistih v teku kot že zaključenih. Za pregled aktivnost datotek ni potrebno iskati po celem računalniku, ker so vse shranjene na enem mestu in so vidne takoj, ko se prijavimo v vmesnik. Na hiter in enostaven način lahko pregledujemo pretekle modele, rezultate itd. Ta del je pri Weki precej bolj neroden in nepregleden. Dolgoročno gledano je Oraclov način boljši.

Sistemski viri

Sistemski viri so šibka točka javanske aplikacije Weka. Tu je prednost Oracla ogromna. Težava Weke je, da ne zmore izvajati algoritmov z velikimi podatkovnimi seti. Na mojem sistemu je imela na voljo 1500Mb pomnilnika in ni uspela izvesti podatkovne množice z 45 000 vrsticami. Oracle se na tem področju zelo dobro odreže.

Cena

Weka je brezplačna in za to ceno ponudi veliko. Tudi Oracle Data Miner je brezplačen, vendar sta podatkovna baza Oracle in dodatek Data Mining plačljiva. Cena Oraclove podatkovne baze z dodatkom »Data Mining« stane okoli 70 000$[9]. Cena Enterprise Edition Oraclove baze za en procesor stane 47,500$, dodatek Data Mining pa 23,000$. K temu je potrebno prišeti še morebitne popravke in podporo.

Komu sta namenjena?

Vsekakor sta si Weka in Oracle Data Miner tako različna, kot sta si različna uporabnika, katerima sta namenjena. Weka je primerna za vse tiste, ki se srečujejo s problemi »normalnih« razsežnosti, ki se učijo podatkovnega rudarjenja ali pa le potrebujejo orodje za testiranje. Oracle je že zaradi cene nedostopen večini morebitnih uporabnikov. Ima pa Oraclovo orodje boljši pregled nad aktivnostmi. Priporočal bi ga vsem tistim, ki delajo z velikimi množicami podatkov (nad 50000 vrstic) in potrebujete stalni pregled nad modeli, nad njihovimi rezultati in jih želite istočasno tudi spreminjati in izboljševati.

Nasvet

Vsakemu, ki se odloča med obema orodjema, bi svetoval da razmisli o svojih potrebah. Če naenkrat delate z malo podatkovnimi množicami, in, če so le te manjše velikosti, bi priporočal Weko, saj ne potrebujete stalnega pregleda nad vsemi modeli. Če pa bi radi oz. morate istočasno delati z večimi podatkovnimi množicami oz. so le-te večje, potem je vsekakor moje priporočilo Oracle.

Splošno

Splošno gledano sta mi bili všeč obe orodji. Nobeno ni izrazito boljše ali slabše od drugega, saj ima vsako svoje prednosti in slabosti. Na koncu, ko seštejem vse pluse in minuse, bi rekel, da ima Oracle Data Miner veliko boljše okolje. Res pa je tudi, da, predvsem pri nastavljanju parametrov metod, Oracle Data Miner zahteva večji angažma analitika, da bi lahko z gotovostjo rekli, da je boljši od Weke.

9 Viri in literatura

1. Pete Chapman (NCR), Julian Clinton (SPSS), Randy Kerber (NCR), Thomas Khabaza (SPSS), Thomas Reinartz (DaimlerChrysler), Colin Shearer (SPSS) and Rüdiger Wirth (DaimlerChrysler), »CRISP-DM 1.0 Step-by-step data mining guide", str. 3–33. 2000. Dostopno na: http://www.crisp-dm.org/CRISPWP-0800.pdf
2. CRoss Industry Standard Processfor Data Mining. Dostopno na: http://www.crisp-dm.org/index.htm
3. »C4.5 algorithm». Dostopno na: http://en.wikipedia.org/wiki/C4.5_algorithm
4. Data Mining Group. Dostopno na: http://www.dmg.org/
5. Kononenko I., *Strojno učenje*, založba FE in FRI,str. 249-254, 2005.
6. Mark Hall, »Weka«, str. 7-11, 2005. Dostopno na: weka.sourceforge.net/presentations/KDD05.ppt
7. Oracle, »Oracle Data Mining Concepts«, poglavje 4-18, 2008. Dostopno na: http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129.pdf
8. Oracle, »Oracle Data Mining tutorial«, str. 9-12. Dostopno na: http://download.oracle.com/odm/odminer/tutorial/11.1/Oracle-Data-Miner-11g-Tutorial.zip
9. Oracle, »Oracle Technology Global Price List«. Dostopno na: http://www.oracle.com/corporate/pricing/technology-price-list.pdf
10. »Oracle Acquires Data Mining Business Of Thinking Machines Corporation«, 1999. Dostopno na: http://www.hoise.com/primeur/99/articles/monthly/AE-PR-08-99-37.html
11. Pentaho comunity, »PMML support in Weka«. Dostopno na:http://wiki.pentaho.com/display/DATAMINING/PMML+Support+in+Weka
12. UCI machine learning repository, »Abalone data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Abalone
13. UCI machine learning repository, »Adult data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Adult
14. UCI machine learning repository, »Breast Cancer Wisconsin (Diagnostic) data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
15. UCI machine learning repository, »Car Evaluation data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Car+Evaluation
16. UCI machine learning repository, »Forest fires data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Forest+Fires
17. UCI machine learning repository, »Iris Data Set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Iris
18. UCI machine learning repository, »KDD Cup 1999 Data Data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data
19. UCI machine learning repository, »SPECT Heart data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/SPECT+Heart
20. UCI machine learning repository, »Tic Tac Toe Endgame data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame
21. UCI machine learning repository, »Wine data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Wine

22. UCI machine learning repository, »Yeast data set«. Dostopno na: http://archive.ics.uci.edu/ml/datasets/Yeast