Institut Ruđer Bošković

Otkrivanje znanja dubinskom analizom podataka

Priručnik za istraživače i studente

Verzija 1.46

Verzija prikladna za printanje dostupna na
http://lis.irb.hr/Prirucnik/prirucnik-otkrivanje-znanja.pdf

Sadržaj

1. Uvod

Dubinska analiza podataka, a posebno njena primjena u otkrivanju znanja su nove tehnike koje predstavljaju neizostavan dio suvremene analize podataka u znanstveno-istraživačkom radu. Zbog toga postoji potreba da svaki aktivni istraživač bude informiran o njihovu postojanju, da razumije njihove mogućnosti i značenje, te da je po potrebi sposoban koristiti dostupne alate. Ovaj priručnik je pregled osnovnih pojmova i postupaka otkrivanja znanja namijenjen sadašnjim i budućim znanstvenicima kojima dubinska analiza podataka nije cilj istraživanja već sredstvo sa kojim će unaprijediti svoj rad.

Priručnik je napisan za potrebe doktorskog studija Medicinskog fakulteta Sveučilišta u Zagrebu za studente koji pohađaju nastavu na predmetu „Otkrivanje znanja u medicinskim domenama“ te za studente doktorskog studija „Molekularna biologija“ koji pohađaju nastavu na predmetu „Postupci otkrivanja znanja“. Želja autora je da studentima olakša savladavanje gradiva u području sa kojim se u principu po prvi puta susreću te da im ostane kao podsjetnik nakon završetka studija.

Iako je većina primjera i ilustracija iz područja medicine, svi prikazani postupci i alati primjenjivi su na isti način u drugim područjima znanosti: od prirodnih i tehničkih znanosti do ekonomije i sociologije. Priručnik može poslužiti svima koji žele naučiti o dubinskoj analizi podataka a posebno onima koji ju trebaju i praktično primijeniti u svojim znanstvenim istraživanjima za potrebe otkrivanja novog znanja iz sakupljenih podataka.

2. Definicija pojmova

Primjena računala u svim područjima ljudske djelatnosti, stvaranje baza podataka, automatizirano zapisivanje informacija i povezivanje računala internetom omogućuje sakupljanje velikih količina podataka. Prirodno se nameće potreba da nam računala pomognu i u njihovoj analizi. U nekim područjima, kao na primjer u trgovačkim transakcijskim bazama ili pak u mjerenjima ekspresivnosti gena modernim uređajima, količine podataka mogu biti tako velike da ih ljudi praktično ne mogu niti pročitati pa je automatizirana analiza podataka i jedino rješenje. U drugima osjećamo da sakupljeni podaci potencijalno kriju važne informacije koje nam mogu pomoći u razumijevanju prošlosti ili optimizaciji ljudske aktivnosti u budućnosti. Čak i kada je skup raspoloživih podataka relativno skroman, broj mogućih hipoteza je ogroman te njihovo sustavno pretraživanje nije moguće bez primjene računala.

2.1. Dubinska analiza podataka

Dubinskom analizom podataka (engl. *data mining*) nazivamo primjenu računarskih postupaka i alata koji nam pomažu u analizi podataka. To je relativno novo područje računarskih znanosti koje se intenzivno razvija te je praktično nemoguće obuhvatiti sve tehnike koje ono uključuje. Što god danas nabrojali već sutra može biti prošireno nekim novim pristupom. Isto tako, ne postoji usuglašen pristup da se neki zadaci analize podataka trebaju obavljati određenim postupcima. Postoje samo pozitivna i manje pozitivna iskustva sa određenim postupcima i njihovom primjenom na konkretnim domenama. Sve što nam može pomoći u boljem razumijevanju podataka je dobro došlo a praktično jedino ograničenje je da postupak treba uključivati primjenu računala te da po potrebi može biti primijenjen na velikim skupovima podataka. To znači da danas, za razliku od recimo primjene statističkih postupaka, nismo u stanju definirati standardnu praksu dubinske analize podataka a svaka njena primjena je mala istraživačka pustolovina traženja dobrog rješenja.

2.2. Dubinska analiza podataka i statistika

Dubinska analiza podataka je u svojim osnovama usko vezana na statističke pojmove i postupke. U većini postupaka dubinske analize u određenom obliku postoji prebrojavanje skupa podataka i uspoređivanje tako dobivenih veličina. Ali za razliku od statistike koja je matematička disciplina usmjerena na pronalaženje i vjerojatnosno vrednovanje odnosa koji postoje u skupovima podataka, dubinska analiza je tehnička disciplina koja teži otkrivanju bilo kojih potencijalno korisnih informacija sadržanih u podacima.

Dubinska analiza podataka ne postavlja uvjete na karakteristike i raspodjelu podataka koji se analiziraju. Ona se najčešće koristi za analizu svih podataka koji su nam dostupni a koji obično i nisu sakupljani prvenstveno za potrebe analize. Cilj je izdvajanje potencijalno najzanimljivijih hipoteza. Zato što dubinska analiza ne postavlja uvjete na karakteristike podataka ona ne može jamčiti za relevantnost svojih rezultata. Kvaliteta rezultata dobivenih

dubinskom analizom nužno se mora naknadno utvrditi ili ljudskom analizom, ili primjenom na neovisnim podacima ili statističkom provjerom. Ako želimo statistički provjeriti neku hipotezu dobivenu dubinskom analizom onda treba sakupiti nezavisni skup podataka primjeren za statističko testiranje te hipoteze. To znači da su postupci dubinske analize podataka i statistički postupci komplementarni u smislu da su prvi pogodni za učinkovito otkrivanje potencijalne povezanosti i postavljanje hipoteza a drugi za testiranje postavljenih hipoteza.

2.3. Dubinska analiza podataka i ljudsko znanje

Proces dubinske analize podataka u širem smislu uključuje pripremu podataka, izvođenje računarskih postupaka analize te interpretaciju rezultata. U pripremi podataka i analizi rezultata nužno je sudjelovanje čovjeka, po mogućnosti eksperta za određeno područje, koji dobro razumije značenje ulaznih podataka te koji je u mogućnosti dobivenim rezultatima dati primjereno tumačenje u okvirima postojećeg ljudskog znanja. Sami po sebi rezultati dobiveni automatskim postupcima analize podataka nemaju nikakvo značenje!

Primjer
Jednostavan rezultat analize podataka može biti: klasa A ako X>3. Interpretacija će biti potpuno različita za marketinšku domenu gdje klasa A označava dobre kupce a podatak X njihova primanja i medicinsku domenu gdje klasa A označava neku dijagnozu a podatak X primjenu određene terapije. Unutar medicinske domene pak značenje otkrivene relacije se bitno može razlikovati ovisno da li je utjecaj terapije X na dijagnozu A dobro poznata činjenica ili sasvim iznenađujuća hipoteza.

Kvaliteta rezultata dubinske analize prvenstveno ovisi o kvaliteti i količini ulaznih podataka. Samo čovjek, posebno ekspert u području, u stanju je definirati i pripremiti kvalitetne izvore podataka te ih po mogućnosti primjereno povezati i transformirati tako da omogući izvođenje raspoloživih postupaka analize podataka. Proces pripreme podataka je iznimno važan te mu je posvećeno cijelo poglavlje 4. Ali taj proces je i dugotrajan pa je općenito priprema podataka vremenski najzahtijevniji dio dubinske analize podataka a cijela dubinska analiza, usprkos nužnoj primjeni računala, zahtijeva mnogo ljudskog rada.

Danas je jasno da je uključenje postojećeg ljudskog znanja u dubinsku analizu, i to ne samo u pripremu podataka i analizu rezultata, od odlučujuće važnosti za kvalitetu konačnog rezultata. Jedan od pristupa je aktivno uključivanje eksperta u usmjeravanje procesa analize (engl. *active mining*), drugi je povezivanje raspoloživih podataka sa bazama znanja (ontologijama) koje sadrže formalizirano ljudsko znanje o odnosima među konceptima. Razvijaju se i postupci za vizualizaciju podataka koji ih transformiraju u oblik prikladan za ljudsku interpretaciju (engl. *visual data mining*). Zanimljivi su i pokušaji da se do važnih informacija dođe na osnovi analize slobodnog teksta (engl. *text mining*) odnosno analize dostupnih internetskih dokumenata (engl. *web mining*).

2.4. Strojno učenje

Osnovno svojstvo po kojem se dubinska analiza podataka razlikuje od tradicionalne ili „obične“ analize je primjena postupaka strojnog učenja. Strojno učenje (engl. *machine learning*) je dio područja računarstva poznatog kao umjetna inteligencija (engl. *artificial intelligence*). Umjetna inteligencija se bavi razvojem računarskih postupaka koji su u stanju računalima simulirati čovjekovo (inteligentno) ponašanje a strojno učenje važnim podskupom tih postupaka koje karakterizira mogućnost učenja na osnovi prethodnog iskustva. Prethodno iskustvo materijalizirano je u obliku povijesnih podataka a rezultat učenja je najčešće model koji je moguće primijeniti u budućnosti sa ciljem poboljšanja ponašanja (inteligentnog računalnog) sustava.

Razvoj i postojanje učinkovitih postupaka za izgradnju modela na osnovi raspoloživih podataka otvorilo je mogućnost da se preko tih modela donesu vrlo značajni zaključci o samim podacima. Primjeri koji se ne uklapaju u modele potencijalno predstavljaju važne izuzetke a podaci koji se koriste u modelima vjerojatno su snažnije povezani sa ciljem modeliranja nego oni koji su izostavljeni. Konstruirani modeli mogu se primijeniti za razvrstavanje neklasificiranih primjera a istovremeno daju informaciju o logičkim vezama između podataka. Detektirani česti uzorci mogu pomoći u razumijevanju odnosa među konceptima i njihovoj pravilnoj hijerarhizaciji. Na tim osnovama postoje razne mogućnosti da se postupci strojnog učenja iskoriste u analizi podataka te su danas oni glavni metodološki alat dubinske analize podataka.

Značaj strojnog učenje je u tome da je to jedino područje umjetne inteligencije koje omogućuje da se kvantitet potencijalno velike količine podataka pretvori u sasvim novu kvalitetu koja se može interpretirati kao konceptualizacija odnosa između podataka i novo znanje. Zbog toga su postupci strojnog učenja predmet stalnog i intenzivnog znanstvenog istraživanja.

Uspješnost postupaka strojnog učenja ocjenjuje sa na osnovu prediktivne točnosti konstruiranih modela na podacima koji nisu korišteni u procesu učenja. Iako je realizirano na desetke raznih postupaka koji koriste razne načine prikaza odnosa među podacima, koji na različite načine definiraju ciljeve strojnog učenja i drugačije pretražuju skup mogućih modela, činjenica je da nije pronađen postupak koji je superioran ostalima za sve ili većinu testiranih domena. Problem proizlazi iz činjenice da koliko god bio velik skup za učenje on redovito predstavlja vrlo mali dio prostora koje treba generalizirati konstruiranim modelom. Zbog toga redovito postoji vrlo mnogo, ponekad i beskonačno mnogo modela koji idealno zadovoljavaju sve poznate podatke a bitno različito se ponašaju na podacima koji će biti sakupljeni u budućnosti. Iako su poznati neki principi, kao što je jednostavnost odabranog modela (poznato i kao princip Occam-ove oštrice) i činjenica da najbolji model ne treba nužno biti točan za sve dostupne podatke, ipak smo još vrlo daleko od jedinstvene teorije strojnog učenja. Istu sudbinu dijeli i dubinska analiza podataka koja koristi postupke strojnog učenja: stalno se otkrivaju novi i korisni postupci a za konkretan problem nemoguće je unaprijed znati koji od njih predstavlja idealan postupak koji treba primijeniti.

2.5. Inteligentna analiza podataka

Inteligentna analiza podataka (engl. *intelligent data analysis*) je drugi naziv za dubinsku analizu podataka. Pridjev „inteligentna“ naglašava da je to analiza podataka zasnovana na postupcima umjetne inteligencije, u prvom redu strojnog učenja.

2.6. Otkrivanje znanja

Primjena postupaka strojnog učenja u analizi podataka može rezultirati u formiranju novog znanja kojeg je moguće prenijeti drugim ljudima, publicirati u znanstvenim radovima ili uključiti u računalne ekspertne sustave. Otkrivanje znanja (engl. *knowledge discovery*) je proces koji nastaje integracijom postupaka strojnog učenja i ekspertne čovjekove analize njegovih rezultata s ciljem otkrivanja i formuliranja novog znanja. Iako je drugi dio ovog procesa kompleksna intelektualna aktivnost, cilj istraživanja u području otkrivanja znanja je a) osigurati da se strojnim učenjem učinkovito pretraži cijeli prostor moguće važnih modela i b) da se sistematiziraju procesi ljudskog tumačenja dobivenih rezultata [1].

Nužan uvjet da bi se rezultati strojnog učenja mogli primijeniti u otkrivanju znanja je da čovjek može razumjeti i interpretirati rezultate dobivene postupcima strojnog učenja. Suvremeni postupci strojnog učenja koji omogućuju izgradnju modela vrlo visoke prediktivne kvalitete, postupci zasnovani na potpornim vektorima (engl. *SVM, Support Vector Machines*), slučajnim šumama (engl. *Random Forest*), ili neuronskim mrežama (engl. *Neural Networks*), ne zadovoljavaju taj kriterij. Za razliku od strojnog učenja kod kojeg je postizanje visoke prediktivne kvalitete primarni cilj, otkrivanje znanja se bavi prvenstveno razvojem i primjenom postupaka strojnog učenja koji sistematski pretražuju skup mogućih modela a svoje rezultate prikazuju u relativno jednostavnom obliku pogodnom za ljudsko razumijevanje.

2.7. Proces otkrivanja znanja

Proces otkrivanja znanja se sastoji od sljedećih koraka: sakupljanja podataka, definiranja cilja otkrivanja znanja, prikaza podataka u formi prikladnoj da se primjene postupci strojnog učenja, izvođenja postupaka strojnog učenja te konačno, interpretaciji njihovih rezultata tako da se iz njih izvuku potencijalno novi i korisni zaključci. Po potrebi se rezultati u završnoj fazi verificiraju statističkim postupcima, vizualno ilustriraju te eventualno integriraju sa znanjem iz drugih izvora.

Često je sam proces iterativan jer izvedeni zaključci mogu ukazati na potrebu uključivanja dodatnih informacija u skup podataka ili sugeriraju drugačije definiranje cilja strojnog učenja. Konačni rezultat je da se postupci primjene strojnog učenja i interpretacije rezultata ponavljaju više puta dok god ekspert nije zadovoljan konačnim rezultatom.

2.8. Tipovi postupaka strojnog učenja

Za otkrivanje znanja primjereni su postupci učenja iz primjera. Osnovni tip je kad svaki primjer ima pridruženu klasu a cilj učenja je izgraditi model koji predviđa klasu na osnovi vrijednosti ostalih značajki primjera. Taj tip strojnog učenja naziva se učenje sa učiteljem (engl. *supervised learning*) jer nam je za generiranje modela potreban skup primjera za koje znamo točnu klasifikaciju. U osnovnom obliku koriste se postupci koji znaju generirati modele za primjere koji imaju samo dvije klase. Takvo učenje se naziva i učenjem koncepata (engl. *concept learning*). Njegova važnost je u tome što za učenje koncepata postoje odlični postupci za izgradnju modela a svaki višeklasni problem je moguće pretvoriti u seriju učenja koncepata.

Drugi tip učenja iz primjera je kada primjeri nisu klasificirani (engl. *unsupervised learning*). Tada je problem učenja definiran kao traženje grupa sličnih primjera (engl. *clustering*). Važna podgrupa ovog tipa učenja je slučaj kada primjeri nisu opisi jedinki već predstavljaju nabrajanje stavki koje se pojavljuju zajedno (problem potrošačke košarice, engl. *marketing basket analysis*) pa je problem učenja pronalaženje čestih skupova stavki. Treba svakako reći da ovaj tip problema nije rezerviran samo za problem potrošačke košarice već je na primjer i medicinske podatke moguće prikazati kao skupove stavki nekog bolesnika gdje su stavke simptomi, uzimanje lijekova i dijagnoze. U takvim slučajevima otkriveni česti skupovi mogu direktno ukazati na povezanost simptoma sa dijagnozom ili uzimanje lijekova sa simptomima.

2.9. Sakupljanje podataka

Proces sakupljanja podataka predstavlja integraciju raznih izvora podataka u jednu cjelinu pogodnu za analizu računalima. Proces pretpostavlja ekspertno znanje o cilju istraživanja, identifikaciju primjera, identifikaciju značajki koje su relevantne a istovremeno i dostupne za predmet istraživanja te konačno njihovo fizičko integriranje.

U jednostavnijim istraživanjima danas je uobičajeno korištenje Microsoft Excel alata koji omogućuje jednostavno integriranje i transformiranje podataka. Ponekad alati strojnog učenja ne prihvaćaju Excel formu podataka no to nije problem jer je uvijek moguće eksportirati podatke u tekst format (funkcijom Save as / Save as type /Text Tab delimited format) u formu koja je općenito prihvaćena.

2.10. Definiranje cilja otkrivanja znanja

Iako učenje iz neklasificiranih primjera izgleda privlačno jer ne traži definiranje cilja modeliranja niti poznavanje klase primjera, postupci učenja iz neklasificiranih primjera su znatno nepouzdaniji. Zbog toga treba problem otkrivanja znanja uvijek nastojati definirati kao klasifikacijski problem ili još bolje kao jedan ili više problema učenja razlikovanja primjera samo dvije klase. Na primjer, u medicinskoj domeni u kojoj nas zanimaju veze

simptoma sa dijagnozama treba definirati pozitivnu klasu bolesnika sa dijagnozom koja nas zanima i negativnu klasu sa zdravim osobama i bolesnicima sa drugim dijagnozama. Rezultat učenja će biti model koji će direktno izdvojiti sve važne značajke koje opisuju simptome vezane uz tu dijagnozu te što više, potencijalno ukazati na logičke veze između simptoma koje trebaju biti zadovoljene da bi dijagnoza bila opravdana.

2.11. Formalna priprema podataka

Da bi podaci bili pogodni za analizu danas postojećim postupcima strojnog učenja oni nužno moraju biti prikazani u obliku tablice. Uobičajeno je da redovi tablice prikazuju primjere (engl. *example or instance*) a svaki pojedini stupac neku od značajki (engl. *feature*) koju imaju primjeri. Drugi uobičajeni nazivi za značajke su atributi (engl. *attribute*) ili varijable (engl. *variable*). U ovom priručniku se pojam značajke veže uz fizičku karakteristiku primjera a pojam atribut uz način prikaza za potrebe strojnog učenja.

Rezultat formalne pripreme podataka je tablica koja ima primjere kao svoje retke od kojih je svaki opisan atributima. Skup atributa je zajednički za sve primjere što znači da svi primjeri trebaju biti opisani na isti način i u istom redoslijedu atributa. Za primjer vidi poglavlje 3.

2.12. Postupci strojnog učenja posebno prikladni za otkrivanje znanja

Za otkrivanje znanja pogodni su postupci strojnog učenja koji svoje rezultate prikazuju u formi koju može razumjeti čovjek. Od postupaka klasifikacijskog modeliranja to su postupci koji modele prikazuju u formi stabla odluke (engl. *decison tree*) ili pravila (engl. *rule*). Od postupaka namijenjenih učenju iz neklasificiranih podataka najprimjereniji za otkrivanje znanja je postupak za otkrivanje čestih uzoraka.

Za otkrivanje skupa atributa koje su bitno povezani sa klasom primjera mogu se iskoristiti i postupci strojnog učenja koji generiraju modele koje čovjek ne može jednostavno interpretirati. U tom pogledu posebno je koristan postupak slučajnih šuma. Taj postupak može biti koristan i za otkrivanje izuzetaka u skupu primjera. Iako tako otkrivene informacije ne predstavljaju pravo operabilno znanje, takav oblik rezultata također može znatno pomoći u razumijevanju podataka, odabiru važnih atributa te čišćenju podataka od grešaka. Ekspertna interpretacija izuzetaka može dovesti u određenim slučajevima i do formiranja pravog operabilnog znanja.

2.13. Alati za izvođenje postupaka strojnog učenja

Postoji mnogo alata koji omogućuju izvođenje postupaka strojnog učenja. Za znanstvena istraživanja od posebnog interesa su alati čije korištenje je besplatno.

Poslužitelj za analizu podataka (DMS od engl. *Data Mining Server*) javno je dostupni mrežni servis. Implementira samo jedan postupak strojnog učenja, ima vrlo ograničenu veličinu podataka ali mu je korištenje izuzetno jednostavno.

Weka je najpopularniji skup postupaka strojnog učenja. Uključuje i cijeli niz postupaka za pripremu podataka te neke mogućnosti vizualizacije rezultata. Koristi vlastiti formalizam prikaza podataka poznat kao arff format. Praktično je samo veličina radne memorije vlastitog računala ograničavajući faktor za veličinu podataka koji se mogu analizirati.

RapidMiner je novi alat koji se odlikuje jednostavnim grafičkim sučeljem za povezivanje operatora koji realiziraju različite funkcije transformacije podataka i indukcije modela. Ovaj alat bi uskoro mogao postati prvi pravi standard za analizu podataka prvenstveno zbog mogućnosti da se objavljuju i izmjenjuju cijeli procesi analize podataka (engl. *workflow*) koji omogućuju postizanje određenih rezultata.

3. Ilustrativni primjer

Pretpostavimo da nas zanimaju karakteristike (značajke) ljudi koji puše. Jedan od načina da to saznamo je da sakupimo informacije o pušačima, da iz sakupljenih primjera izgradimo model koji opisuje pušače te da na koncu interpretacijom modela zaključimo o karakteristikama pušača.

Prvo je potrebno uočiti da su nam za izgradnju modela koji opisuje pušače potrebni i primjeri koji opisuju pušače i primjeri koji opisuju nepušače. Znači da ćemo za naše istraživanje sakupljati podatke o svim osobama, da ćemo u jednom atributu imati informaciju da li su oni pušači ili nisu te da će nam taj atribut predstavljati klasifikator (ciljni atribut) za koji ćemo tražiti model.

Sljedeća tablica prikazuje sakupljene podatke o 8 osoba od kojih su 4 pušači a 4 nepušači. Pored ciljnog atributa PUSAC tu su i drugi koji sadrže informacije o imenu, spolu, zanimanju, nivou obrazovanja, prihodu i težini.

IME	STAROST	SPOL	OBRAZ	ZANIM	TEZINA	PRIHOD	PUSAC
jan	30	muski	nize	radnik	27.3	14000	da
janko	55.5	muski	srednje	radnik	90	20000	ne
zora	?	zenski	visoko	ucitelj	65.2	1000	ne
tanja	18	zenski	srednje	student	55.1	0	ne
tom	70	muski	visoko	?	60	9000	da
tomi	35	muski	srednje	prof	33	16000	ne
stev	42.2	muski	nize	vozac	27	7500	da
marc	29	muski	?	konobar	31	8300	da

Primjer 1. Skup primjera: pušači - nepušači

Primjenom DMS poslužitelja kao rezultat modeliranja dobiti će se sljedeće pravilo:

PUSAC ako spol=muski i ako prihod < 15000.

Isti ili sličan model moguće je dobiti i drugim alatima.

Treba svakako uočiti da dobiveno pravilo točno karakterizira razlike između primjera koji opisuju pušače i nepušače. Svi pušači su muškarci i imaju prihod ispod 15000. Ili drugim riječima svi nepušači su ili žene ili muškarci koji zarađuju više od 15000. Sa stanovišta kvalitete na skupu za učenje kao i interpretabilnosti rezultata možemo biti sasvim zadovoljni dobivenim rezultatom. Međutim sa rezultatom ne možemo biti zadovoljni kao ljudi koji poznaju problem pušenja i koji znaju da postoji veliki broj žena koje puše. Također, možemo očekivati da će prediktivna kvaliteta generiranog modela biti loša barem iz upravo spomenutog razloga.

Slaba kvaliteta modela je posljedica problema u sakupljanju podataka. Mi u 8 sakupljenih primjera nemamo niti jednu ženu pušača. Postupci strojnog učenja kao izvor informacija o svijetu imaju samo podatke koje im dajemo u obliku primjera. U konkretnom slučaju nije postojala niti teoretska prilika da se konstruira pravilan model o ženama pušačima. Jedino rješenje je proširiti skup primjera i ponoviti postupak indukcije modela.

Zaključak je da kvaliteta induciranih modela u cijelosti ovisi o kvaliteti ulaznih podataka. To se podjednako odnosi i na izbor i količinu raspoloživih primjera kao i na izbor i količinu atributa sa kojima opisujemo primjere.

4. Priprema podataka

Priprema podataka je prvi i vrlo važan korak u procesu otkrivanja znanja. O kvaliteti pripreme podataka u znatnoj mjeri ovisi kakvo znanje se može otkriti. To je radom vrlo intenzivan proces a mogu ga uspješno obaviti samo osobe koje dobro razumiju ciljeve otkrivanja znanja, koje imaju dostup do relevantnih podataka te razumiju njihovo značenje i uvjete pod kojima su sakupljani [2].

4.1. Koliko primjera treba sakupiti ?

Odgovor je: čim više! Sa porastom broja primjera raste pouzdanost da otkriveno znanje točno opisuje odnose u domeni istraživanja. Što više, veći broj primjera omogućuje da se pouzdano otkriju složeniji odnosi koji mogu predstavljati novost u svom području. A primjeri koji su izuzeci u malim skupovima primjera imaju priliku u velikim skupovima primjera postati predstavnici rijetkih ali važnih podskupina sa potencijalno važnim posljedicama.

U medicinskim domenama uobičajeno je da je broj primjera od 100 do 300. U biološkim domenama taj je broj ponekad i manji, što ograničava složenost detektiranih odnosa na samo osnovne relacije. U domenama ispod 50 primjera se ne može očekivati pouzdano otkrivanje niti elementarnih odnosa.

Ovisno o alatu, gornja granica broja primjera je određena raspoloživim memorijskim prostorom, i osim za DMS, obično je veća od 10.000. Treba imati na umu da vrijeme izvođenja svih postupaka strojnog učenja znatno raste sa brojem primjera. Preporuka je da se početne analize ne rade sa više od 500 primjera jer će i one jasno ukazati na rezultate koji se mogu očekivati i sa znatno većim brojem primjera. 5000 primjera u skupu za učenje je iskustveni maksimum do kojeg se može očekivati poboljšanje kvalitete rezultata.

Za velike skupove raspoloživih primjera preporučljivo je samo dio koristi za otkrivanje znanja a preostali dio iskoristi za nezavisnu verifikaciju rezultata.

Pored kvantitete važna je i kvaliteta primjera. U prvom redu to znači da treba skupljati prvenstveno primjere koji imaju sve poznate vrijednosti svojih atributa. Posebno je važno da sakupljeni primjeri dobro reprezentiraju cijelu populaciju koja je predmet istraživanja. Ponavljanje istih i vrlo sličnih primjera malo će doprinijeti kvaliteti konačnog rezultata.

Od procesa sakupljanja i pripreme podataka se ne očekuje statistička stratifikacija po pojedinim klasama ili atributima (npr. ne očekuje se da imamo jednak broj muških i ženskih bolesnika, niti jednak broj onih koji primaju i ne primaju neku terapiju). Ali se očekuje da svaka od relevantnih podskupina bude primjereno zastupljena.

4.2. Koliko atributa treba sakupiti ?

Odgovor je: čim više! Pri tome je u prvom redu naglasak na kvaliteti atributa koja se odražava u činjenici da su atributi čim bolje povezani sa predmetom istraživanja. Ako smo u dilemi da li je neki atribut povezan sa predmetom istraživanja onda ga svakako treba uključiti. Prisutnost atributa koji nemaju nikakve veze sa predmetom istraživanja u principu neće narušiti kvalitetu konačnog rezultata već eventualno samo povećati naš trud oko sakupljanja i pripreme podataka.

U medicinskim domenama broj atributa je obično od 10-100 dok je u biološkim ponekad i veći od 1000. Dostupni alati u principu bez problema prihvaćaju primjere opisane sa takvim brojem atributa.

4.3. Vrste atributa

Postoje dvije osnovne vrste atributa: nominalni i numerički. Primjeri numeričkih veličina su '3' i '8.12', a nominalnih 'crven' i 'zdrav'. DMS poslužitelj razlikuje dvije podvrste numeričkih atributa: kontinuirani i cjelobrojni.

Svaki primjer mora sadržavati kompletan skup atributa i to uvijek u istom poretku. Pri tome je dozvoljeno da neke od vrijednosti nisu poznate. Nepoznate vrijednosti se tipično označavaju znakom '?'.

Svaki atribut mora biti uvijek istog tipa za sve primjere. Preporuča se na početku skupljanja podataka unaprijed odrediti kojeg tipa će biti koji atribut i toga se treba dosljedno držati.

Vrlo preporučljivo je za nominalne atribute unaprijed odrediti koje vrijednosti mogu pospremiti. Na primjer samo 'da' ili 'ne'. Ili osnovni skup boja kao što je 'žut', 'zelen', 'plav' i 'crven'. Preporučljivo je da je skup vrijednosti za svaki nominalni atribut bude relativno malen. Nikako se ne preporučuje da bude veći od 10. U slučajevima kada bi mogao biti veći, preporuča se grupiranje pojedinih vrijednosti u općenitije koncepte.

Preporučljivo je za numeričke atribute unaprijed odrediti raspon vrijednosti koji mogu poprimiti.

Atributi ne mogu biti slobodni tekst niti mogu uključivati nabrajanje više pojmova.

4.4. Posebni atributi

Preporučljivo je uvesti jedan nominalni atribut koji jednoznačno identificira primjere. To može znatno pomoći u analizi sakupljenih podataka. Takav atribut neće smetati u procesu induciranja modela, a ako se on pojavi u modelu, onda ukazuje na to da postoje primjeri koji predstavljaju izuzetke koje nije bilo moguće generalizirati drugim atributima.

Pored toga ima smisla uvesti numerički atribut koji označava redoslijed sakupljanja ili predstavlja transformaciju datuma sakupljanja u numerički atribut. Takav atribut može pomoći u otkrivanju promjena koje se dešavaju u domeni (na primjer smanjuje se ukupni broj pušača u populaciji) a u nekim slučajevima njegovo uključenje u model može pomoći da se uoče promjene u načinu sakupljanja podataka.

4.5. Da li jedna značajka može generirati više atributa ?

Da. Nema nikakve zapreke da se neka složena karakteristika primjera prikaže sa više atributa. Primjer je podatak o krvnom tlaku koji redovito prikazujemo sa dva numerička atributa koji odgovaraju dijastoličkim i sistoličkim vrijednostima. Pri tome njihova vrlo značajna korelacija nije nikakav problem za njihovu analizu. Dodatno se može uvesti i atribut sa vrijednostima 1-4 koje odgovaraju liječničkoj dijagnozi sniženog, normalnog, povišenog i visokog tlaka.

Posebnu pažnju treba posvetiti kompleksnim značajkama kao što je pojam pušač. Potpuno je neprimjereno karakteristiku pušenja prikazati brojem cigareta na dan tako da nepušačima odgovara vrijednost 0, a onda bivše pušače označiti posebnom vrijednosti, na primjer -1. Moguće rješenje je imati jedan nominalni atribut sa vrijednostima 'pušač', 'nepušač', i 'bivši_pušač' a znatno informativnije rješenje je imati dva atributa od kojih je prvi numerički i predstavlja broj cigareta a drugi nominalni sa značenjem bivši pušač sa vrijednostima 'da' i 'ne'.

4.6. Da li više značajki može generirati jedan atribut ?

Da. I to je dobra praksa ako postoji ekspertno znanje koje omogućuje takvu integraciju u cilju generiranja atributa koji mogu biti korisni za cilj istraživanja. Primjer je atribut koji označava da li u porodici postoje slične bolesti sa vrijednostima 'da' ili 'ne' i koji nastaje integriranjem više različitih informacija o bolesniku. Sličan je i atribut o liječenju od ovisnosti koji nastaje integracijom sakupljenih informacija o raznim vrstama ovisnosti.

Zaključak je da je dobro imati puno podataka i na njihovim osnovama vrlo različito prikazanih atributa.

4.7. Pretvorba nominalnih u numeričke atribute

Ako se uoči da se vrijednosti nekog nominalnog atributa mogu poslužiti po veličini ili po nekom redoslijedu tada se svakako preporuča da se iskoristi mogućnost da se atribut pretvori u numerički sa cjelobrojnim vrijednostima koje odgovaraju nominalnim vrijednostima. Primjer za veličinu je da se vrijednosti 'malo', 'srednje', 'veliko' i 'vrlo_veliko' mogu pretvoriti u vrijednosti '1', '2', '3' i '4'. Primjer za redoslijed je da se dani u tjednu umjesto imenima 'ponedjeljak', 'utorak' itd. označe vrijednostima '1'-'7'.

Iako numeričke vrijednosti nisu toliko intuitivne kao nominalne, prednost ovakvog načina prikaza je da se u modeliranju mogu pokazati kao vrlo korisne jer omogućuju otkrivanje relacija kao na primjer, da je značajna vrijednost atributa veća 2 što bi u prvom primjeru značilo da je značajno kada je vrijednost atributa velika ili vrlo velika a u drugom primjeru da se nešto desilo u srijedu do nedjelje. Korištenjem nominalnih atributa takve relacije nisu moguće.

Ponekad mogućnost ovakve pretvorbe i nije sasvim očita. Primjer je sa imenima boja crveno, zeleno i plavo koje u nekim domenama ima smisla pretvoriti u njihovu valnu dužinu. Ako smo u dilemi da li pretvorba ima smisla, nema nikakve zapreke da se zadrži i prikaz nominalnim vrijednostima i doda atribut sa numeričkim vrijednostima.

Problem sa cirkularnim atributima je da nemaju jednoznačno definiran početak. Primjer su dani u tjednu. Moguće je označiti ponedjeljak sa '1' pa je u tom slučaju nedjelja broj '7' ali je isto tako korektno označiti nedjelju sa '1' pa je tada ponedjeljak broj '2'. Poteškoća je da u nekim primjenama može biti prikladan jedan način a ponekad pak onaj drugi. Ako smo u dilemi, nema zapreke da u istoj domeni koristimo istovremeno oba atributa sa različito kodiranima danima u tjednu.

4.8. Nepoznate vrijednosti

Ako podatak ne znamo jer nije izmjeren ili upisan ili naprosto nije primjeren u danom primjeru onda on ima nepoznatu vrijednost. Svaki atribut bez obzira na vrstu može imati nepoznate vrijednosti koje se tipično označavaju sa ‘?’. Treba nastojati da nepoznatih vrijednosti bude malo.

U nekim slučajevima nepoznata vrijednost može imati posebno značenje. Primjer je da ne znamo vrijednost nalaza jer je bolesnik bio i suviše slab da bi se izvršilo mjerenje. U takvim slučajevima ima smisla kod nominalnih atributa umjesto oznake '?' uvesti za takve primjere vrijednost 'neprimjereno' a kod numeričkih atributa svakom od njih dodati posebni nominalni atribut za značenjem 'neprimjereno_mjerenje' sa vrijednostima 'da' i 'ne'.

4.9. Nabrajanje vrijednosti

Informacija o uzimanju lijekova je tipična značajka koja traži nabrajanje. Prikaz se ne može riješiti jednim atributom. Moguće rješenje je da se za svaki lijek uvede poseban atribut a da vrijednosti tih atributa mogu biti samo 'da' ili 'ne'. Takvo rješenje nije ni praktično ni korisno. Nije praktično jer lista lijekova i pripadajućih atributa može biti vrlo duga. Nije korisno jer je malo vjerojatno da će informacija o uzimanju nekog specifičnog lijeka imati dobra svojstva generalizacije. Rješenje koje se preporuča je da se uvede nekoliko atributa koji odgovaraju grupama lijekova koji su potencijalno povezani sa predmetom istraživanja (antihipertenzivi, antikoagulanti i slično). Vrijednosti tih atributa mogu biti 'da' ili 'ne' i označavaju da li osoba uzima jedan ili više lijekova iz te grupe. U posebnim slučajevima moguće je da ti atributi

imaju više nivoa. Na primjer '0' za ne koristi inzulin, '1' za inzulin u tabletama a '2' za inzulin u injekcijama. Ili da ti atributi označavaju broj lijekova iz iste grupe koje osoba koristi.

Drugi primjer nabrajanja je opis ljudskog obroka. Umjesto nabrajanja koje bi trebalo služiti dobivanju informacije o prehrambenim navikama pojedinca, preporuča se koristiti nominalni atribut sa malim skupom dobro definiranih i informativnih vrijednosti kao što su 'nisko_kaloričan', 'mediteranski', 'dijetalni' i slično. Pri tome je potrebno ekspertno znanje i iskustvo da bi se odabrali odgovori koji će biti najinformativniji za predmet istraživanja. Potrebne su i primjerene definicije pojmova sa ciljem jednoznačnog kodiranja primjera.

4.10. Vremenski ili prostorni niz istih vrijednosti

Vremenski niz nastaje kada je ista numerička vrijednost mjerena u više navrata. Primjer je količina šećera u krvi mjerena svakih mjesec dana u posljednjih jednu do dvije godine. Uključivanje takvih podataka može biti izuzetno korisno za otkrivanje smjera i intenziteta promjene ključnih značajki koje utječu na predmet istraživanja. Iako je moguće da se za svako takvo mjerenje uvede zaseban atribut, mogućnost da sustavi strojnog učenja dobro iskoriste takve podatke je relativno skroman. Preporuča se da se umjesto stvarnih vrijednosti uključe posebni atributi koji opisuju potencijalno važne značajke niza. Primjeri takvih atributa su srednja vrijednost niza, srednja promjena vrijednosti (engl. *slope*), standardna devijacija, maksimalna vrijednost i slično. Takvi atributi mogu opisivati niz u cjelini ili njegove dijelove, na primjer srednja vrijednost odnosno srednja promjena vrijednosti prije šest mjeseci ili na početku promatranja, te razlike srednje vrijednosti i srednje promjene na početku i kraju promatranja.

Potrebna izračunavanja za pripremu ovakvih atributa preporuča se obaviti unutar Excel tablice.

Uključivanje u modele pojedinih atributa koji opisuju značajke niza može sugerirati potrebu da se neki efekti detaljnije istraže, odnosno da se dodaju nove detaljnije ili specifičnije značajke. Zbog toga je proces pronalaženja dobrih značajki koji opisuju nizove često iterativan.

4.11. Dugi vremenski nizovi i slike

Dugi vremenski niz nastaje automatskim mjerenjem neke veličine. Takav niz može uključivati repetitivne pojave koje se mogu mijenjati po intenzitetu i trajanju. Primjer je izmjereni EKG signal. Direktno korištenje takvih velikih skupova podataka kao i informacija u bilo kojem slikovnom obliku nije prikladno za sustave strojnog učenja. Potrebno je automatsko izdvajanje skupa značajki koje se mogu koristiti kao atributi. Sustavi za takve primjene su rijetki i predmet su znanstvenih istraživanja.

4.12. Relacije generirane atributima

O vrsti atributa ovisi kakve relacije će se moći koristiti u modelima. Za nominalne atribute moguće su jedino relacije jednako ili različito. Primjer je atribut koji opisuje boje i moguće relacije su da je nešto crveno ili nešto nije plavo. Za kontinuirane numeričke atribute moguće su samo relacije veće ili manje. Na primjer, nešto je već od 3.5 ili manje od 14.1.

Za usporedbu pretpostavimo da smo boje prikazali numeričkom veličinom njihove valne dužine. Tada možemo izraziti da je nešto manje od 480 što bi značilo da je plavo ili ljubičasto odnosno veće od 550 što bi značilo da je žuto, narančasto ili crveno. Taj primjer pokazuje kako se jezik modela bitno razlikuje sa načinom prikaza atributa. Sustavi strojnog učenja su izrazito loši u automatskom pretvaranju oblika atributa. Problem se rješava povećanjem složenosti modela. Na primjer, da je nešto plavo u slučaju numeričkog prikaza boja označava se složenim izrazom „valna dužina veća od 450 i valna dužina manja od 480“. A da je nešto crveno ili narančasto u slučaju nominalnog atributa rješava se primjenom logičke operacije ILI bilo na nivou uvjeta ili cijelih modela.

Kod nekih sustava strojnog učenja, kao što je DMS opisan u sljedećem poglavlju, posebnu a vrlo zanimljivu podvrstu numeričkih atributa predstavljaju cjelobrojni numerički atributi. Oni mogu poprimati samo cjelobrojne vrijednosti 0-1000 a bitno svojstvo im je da se za njih mogu koristiti relacije kao da su i numerički i nominalni atributi istovremeno. Znači da neki atribut može biti manji od 5, veći od 10 ali i jednak 2 i različit od 3. To svojstvo može biti od velike koristi ali ga treba koristiti samo za diskretne veličine koje po svojoj definiciji ne mogu poprimiti necjelobrojne vrijednosti. Primjeri su da je neki student na prvoj, drugoj ili petoj godini studija odnosno da neka osoba stanuje na prvom, drugom ili desetom katu zgrade. U takvim slučajevima opravdano je u modelima koristiti relacije student je na drugoj godini ili student nije na prvoj godini ali isto tako da neko stanuje ispod trećeg kata ili živi iznad osmog kata zgrade.

5. Data mining server: poslužitelj za analizu podataka

Korištenje poslužitelja za analizu podataka (Data Mining Serever, DMS) koji je javno dostupan na internet adresi **http:\\dms.irb.hr** predstavlja najjednostavniji i najbrži način za početak procesa otkrivanja znanja. Ovaj poslužitelj realizira u obliku mrežnog servisa samo jedan postupak strojnog učenja i to postupak indukcije pravila za otkrivanje podgrupa (engl. *subgroup discovery*).

Postupak otkrivanja podgrupa detaljno je opisan u sljedećem poglavlju dok ovo poglavlje samo opisuje način korištenja poslužitelja. Drugi alati koji omogućavaju korištenje većeg broja raznih postupaka strojnog učenja i koji se mogu koristiti za analizu bitno većih skupova podataka opisani su u poglavlju 8.

Prednost DMS poslužitelja je izuzetna jednostavnost. Nije potrebna instalacija programske podrške, oblik u kojem treba pripremiti podatke je jednostavan a korisnik može procesom generiranja modela upravljati preko samo jednog parametra. Nedostaci su vrlo ograničena veličina skupa podataka (do najviše 250 primjera sa 50 atributa) te mogućnost da se rješavaju samo klasifikacijski problemi sa dvije klase.

5.1. Definicija klasifikacijskog problema

Kao i većina drugih postupaka strojnog učenja, otkrivanje podgrupa rješava problem razlikovanja primjera različitih klasa. Zbog toga je prvi korak dubinske analize definiranje klasa primjera koje želimo razlikovati odnosno čije značajke želimo otkriti.

Za primjer podataka o pušačima-nepušačima koji je već opisan u poglavlju 3, cilj sakupljanja podataka bio je razumijevanje koja svojstva karakteriziraju pušače. Zbog toga je logično da su nam klase primjera pušači i nepušači. To znači da nam klasu primjeru određuje atribut PUSAC sa svojim vrijednostima 'da' i 'ne'.

5.2. Formalni oblik podataka

Za korištenje DMS poslužitelja korisnik treba na svom računalu pripremiti podatke u obliku tablice sa N+1 redaka (N je broj primjera) i M stupaca (M je broj atributa koji opisuju primjere). U prvom redu tablice trebaju biti imena atributa.

Tablica treba biti pohranjena u datoteci u običnom tekstualnom obliku koji je moguće čitati, pisati i mijenjati alatom kao što je Microsoft Wordpad. Podaci pripremljeni Microsoft Excel alatom trebaju se eksportirati u tekstualni oblik.

Izbor atributa klase označava se stavljanjem uskličnika ('!') ispred naziva atributa. Primjer 2 prikazuje isti skup podataka iz poglavlja 3 ali sada primjeren za korištenje DMS poslužitelja.

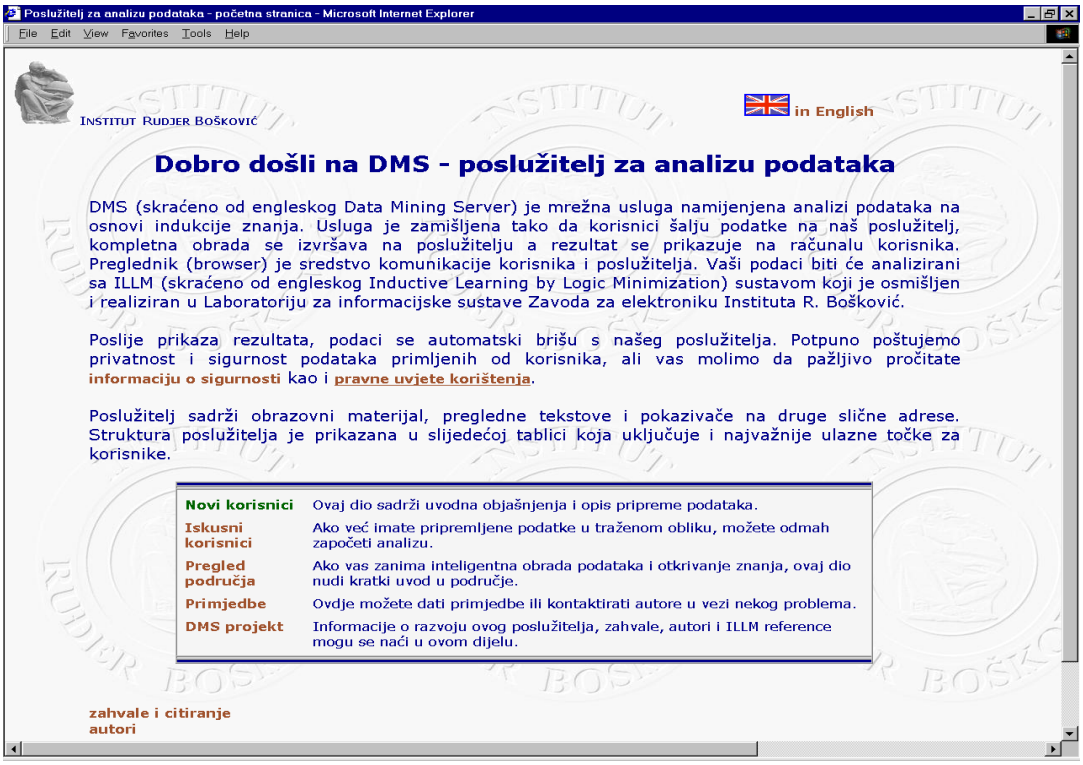
Osim što je nužno odabrati ciljni atribut koji određuje klase primjera, potrebno je označiti i koji primjeri predstavljaju pozitivnu klasu koju želimo modelirati. U našem slučaju mi želimo generirati model za pušače te ćemo zbog toga ispred svakog 'da' u stupcu atributa klase '!PUSAC' staviti uskličnik (vidi Primjer 2). U slučaju da želimo generirati model(e) za klasu nepušača tada bismo uskličnik stavili ispred svakog 'ne' u tom stupcu.

IME	STAROST	SPOL	OBRAZ	ZANIM	TEZINA	PRIHOD	!PUSAC
jan	30	muski	nize	radnik	27.3	14000	!da
janko	55.5	muski	srednje	radnik	90	20000	ne
zora	?	zenski	visoko	ucitelj	65.2	1000	ne
tanja	18	zenski	srednje	student	55.1	0	ne
tom	70	muski	visoko	?	60	9000	!da
tomi	35	muski	srednje	prof	33	16000	ne
stev	42.2	muski	nize	vozac	27	7500	!da
marc	29	muski	?	konobar	31	8300	!da

Primjer 2. Podaci iz Primjera 1 pripremljeni za analizu sa DMS poslužiteljem

5.3. Početna stranica

Slika 1. prikazuje početnu stranicu internet poslužitelja za analizu podataka (DMS) na hrvatskom jeziku. Njoj se pristupa preko osnovne stranice, koja je na engleskom jeziku pritiskom na hrvatsku zastavicu. Stranica daje osnovne informacije o poslužitelju, obavještava da se on bez ograničenja može koristiti za razne namjene ali i upozorava da ne može osigurati potpuna privatnost podataka koji mu se daju na analizu.



Slika 1. Početna stranica poslužitelja za analizu podataka (DMS)

Preko te stranice može se pristupiti do liste sadržaja prikazane u tablici na dnu stranice. Poveznica „Novi korisnici“ upućuje na opis načina korištenja poslužitelja. Svakom novom korisniku preporuča se pažljivo čitanje ovih uputa.

Poveznica „Iskusni korisnici“ omogućuje da se direktno pređe na analizu podataka. Preostale poveznice omogućuju čitanje literature vezane uz ovaj poslužitelj i primjenu strojnog učenja u dubinskoj analizi podataka odnosno pristup nekim informacijama o samom projektu poslužitelja kojeg je financiralo Ministarstvo znanosti i športa Republike Hrvatske 2000. godine.

5.4. Glavna stranica za analizu podataka

Proces analize podataka sastoji se od sljedećih osnovnih koraka:

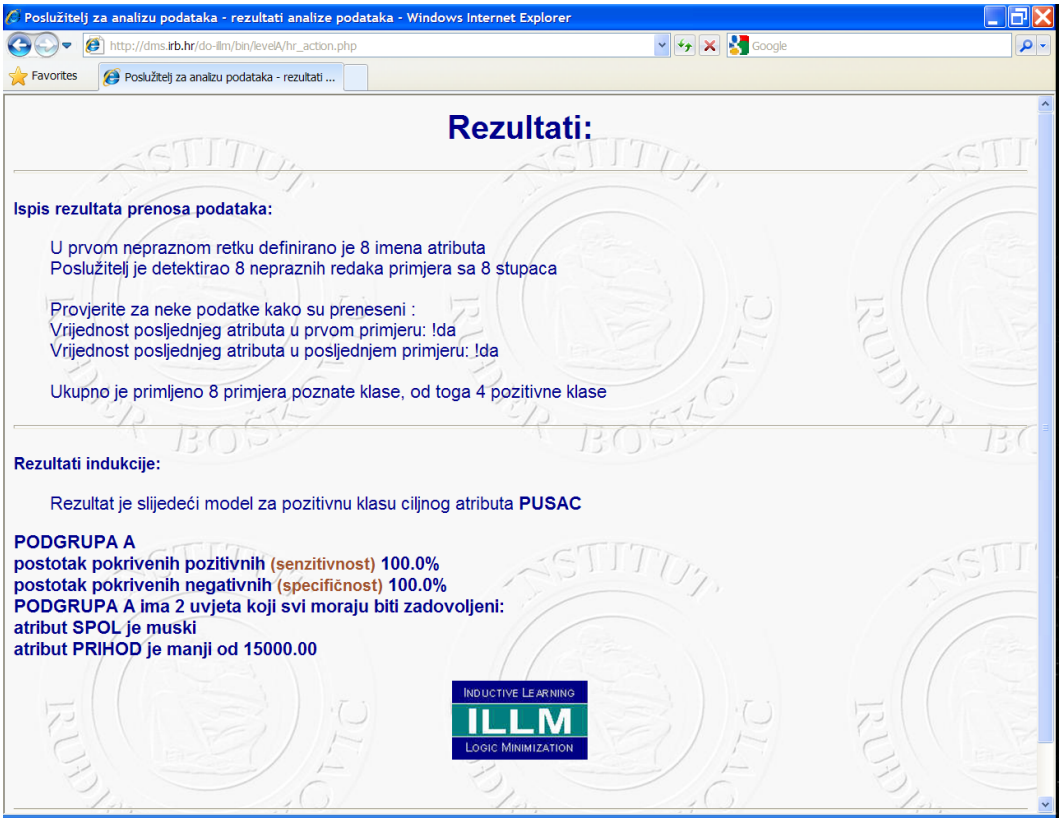
1) pripreme podataka u traženom formatu na vlastitom računalu (vidi poglavlje 5.2)
2) prijenos podataka na poslužitelj korištenjem web preglednika (Internet Explorer, Mozilla Firefox...)
3) prijem rezultata (pravila koji opisuju modele) čitanjem internet stranice koju je za nas pripremio poslužitelj.

Slika 2. prikazuje glavnu stranicu poslužitelja na hrvatskom jeziku koja je namijenjena prijenosu podataka.

Slika 2. Glavna stranica poslužitelja za prijenos podataka

Tipkom „Browse...“ pretražuje se vlastito računalo i odabire se ime datoteke sa pripremljenim podacima. Pretpostavlja se da su podaci formalno korektno pripremljeni. Nakon toga možemo pritisnuti tipku „Počni indukciju“ te očekivati odgovor poslužitelja.

5.5. Rezultati indukcije



Slika 3. Pravilo dobiveno indukcijom: Pušači su muškarci sa prihodom manjim od 15000

Slika 3. prikazuje odgovor poslužitelja. U prvom, gornjem dijelu je ispis rezultata prijenosa podataka. Sadrži informaciju o broju primjera i atributa koje je primio poslužitelj te vrijednost posljednjeg od atributa u prvom i posljednjem primjeru. Na osnovi ovih informacija korisnik može provjeriti kao prvo, da li su podaci korektno pripremljeni a zatim i da li je njihov prijenos uspješno izvršen.

U drugom dijelu, ispod crte je rezultat indukcije. Sastoji se od 1-3 opisa podgrupa koje su označene sa PODGRUPA A do PODGRUPA C.

U prvom dijelu opisa svake podgrupe navedena je njena kvaliteta pokrivanja primjera koji su korišteni u procesu indukcije. Kvaliteta je izražena sa dvije vrijednosti: senzitivnosti, koja pokazuje koji postotak pozitivnih primjera pravilo (korektno) pokriva, i specifičnosti, koja pokazuje koji postotak negativnih primjera pravilo (korektno) ne pokriva. U primjeru pušača-nepušača inducirano pravilo je točno za sve primjere jer ono pokriva sve pozitivne primjere a niti jedan negativni. Zbog toga su njegove vrijednosti senzitivnosti i specifičnosti jednake 100% (Slika 3). U realnim primjerima rijetko možemo očekivati tako visoku kvalitetu induciranih modela.

U drugom dijelu opisa podgrupe je specifikacija pravila koje opisuje podgrupu. U našem primjeru se ono sastoji od dva literala: (SPOL = muski) i (PRIHOD manji od 15000). Pravilna interpretacija je da su pušači muškarci sa prihodom manjim od 15000.

5.6. Greške u pripremi podataka

Iako je format pripreme podataka vrlo jednostavan, nužno je da se on dosljedno poštuje. Česte greške su izostavljanje vrijednosti koja je nepoznata (umjesto da se ona označi sa '?'), korištenje hrvatskih znakova 'č', 'ž' i ''š' što nije dopušteno, te korištenje decimalnog zareza umjesto decimalne točke. U slučaju postojanja greške, indukcija se neće izvesti a korisnik će dobiti obavijest o greški sa upozorenjem na mogući izvor problema. Slika 4. ilustrira način javljanje greške u slučaju kada je u jednoj od numeričkih vrijednosti korišten znak decimalnog zareza.

Način pripreme podataka i formalni uvjeti koje trebaju zadovoljavati podaci su detaljno prikazani na samom poslužitelju u dijelu koji je namijenjen novim korisnicima pod nazivom „Kako pripremiti datoteku“.

Slika 4. Obavijest o greški u slučaju kada je decimalni zarez korišten umjesto decimalne točke

5.7. Primjeri pripremljenih datoteka

Od velike koristi mogu biti datoteke koje se nalaze na poslužitelju u dijelu namijenjenom novim korisnicima pod nazivom „Primjeri pripremljenih datoteka“. One pokazuju kako izgleda formalno korektno pripremljena datoteka i to u raznim formatima u kojima je korišten razmak, zarez, točka-zarez, odnosno TAB kao znak razdvajanja podataka. Rad poslužitelja uvijek se može testirati tako da se prvo prenese neka od pripremljenih datoteka na vlastito računalo, a zatim se ona pošalje poslužitelju na analizu preko glavne stranice.

Od posebne je važnosti datoteka za dijagnosticiranje meningoencephalitisa. Ta je datoteka kopija podataka pripremljenih za međunarodno takmičenje u području otkrivanja znanja (JSAI KDD Challenge 2001) te predstavlja odličan primjer realnog skupa podataka u medicinskim primjenama. Vrlo ilustrativan je skup pripremljenih atributa od strane organizatora takmičenja te postupak transformacije forme podataka tako da datoteka zadovoljava formalne uvjete DMS poslužitelja. Ova datoteka omogućuje otkrivanje potencijalno važnih medicinskih zaključaka vezanih na dijagnosticiranje i liječenje meningoencephalitisa, posebno vezano na razlikovanje virusnih i bakterijskih uzročnika.

5.8. Parametri procesa indukcije

Korisnik može utjecati na rad poslužitelja sa nekoliko parametara koji se odabiru na glavnoj stranici pri prijenosu podataka.

Prvo se odabire tip znaka razdvajanja koji se koristi u pripremljenoj datoteci. Ovaj izbor ne utječe na rezultat indukcije.

Nakon toga se odabire broj pravila koji će se generirati. Broj može biti u granicama 1-3.

Slijedi parametar generalizacije koji može biti 0.1-100. Sa vrijednostima do 10 inducirana pravila će u principu biti vrlo precizna (sa malim brojem pogrešnih pokrivanja negativnih primjera) a sa vrijednostima iznad 10 ona će biti sve općenita te će pokrivati veći broj i pozitivnih i negativnih primjera. U procesu analize treba uvijek eksperimentirati sa nekoliko vrijednosti ovog parametra te na osnovi ekspertnog razumijevanja rezultata odlučiti koji dobiveni modeli su optimalni za dani slučaj. Način utjecaja parametra generalizacije na karakteristike modela opisan je u poglavlju 7.1.6.

Posljednji parametar omogućuje uključenje detekcije šuma. Odabir ovog parametra neće utjecati na rezultat indukcije pravila ali će zbog dodatnog vremena potrebnog za otkrivanje šumnih primjera cijeli proces biti dugotrajniji.

5.9. Otkrivanje izuzetaka u skupu primjera

Opcija otkrivanja šuma omogućuje da se izdvoji do pet primjera koji potencijalno predstavljaju izuzetke u skupu za učenje. Ako postoje, ovi izuzeci će biti nabrojani u rezultatima analize podataka i to iza ispisa informacija o prijenosu podataka a prije prikaza induciranih primjera (Slika 5).

Važno je uočiti da detektiranje šuma ne utječe direktno na indukciju pravila. Ako korisnik analizom utvrdi da se zaista radi o šumu, bilo kao posljedica stvarnih izuzetaka ili zbog grešaka u sakupljanju podataka, tada se preporuča te primjere privremeno ili trajno izbrisati iz skupa za učenje i ponoviti proces indukcije. Na taj način će se omogućiti da detektirani šum više ne utječe na kvalitetu novih pravila.

Postupak otkrivanja šuma opisan je u poglavlju 7.1.8.

Slika 5. Rezultat indukcije pravila za meningoencephalitis domenu za razlikovanje bakterijskih i virusnih oboljenja. U srednjem dijelu slike su četiri primjera detektiranog šuma

6. Interpretacija rezultata

Za pravilnu interpretaciju rezultata dobivenih primjenom postupaka strojnog učenja bitno je ekspertno poznavanje domene koja se analiza. To uključuje razumijevanje problema istraživanja i razumijevanje značenja analiziranih podataka i uvjeta pod kojima su oni sakupljeni. Isto tako, važno je razumjeti korištene postupke strojnog učenja, njihova ograničenja i značenje formalizma prikaza rezultata.

Rezultati primjene strojnog učenja su u nekoliko oblika. Osnovni oblik su modeli koji se prikazuju ili pravilima ili stablima odlučivanja.

Drugi oblik rezultata su izdvojeni šumni primjeri. Oni predstavljaju potencijalno važan rezultat koji zavređuje ljudsku pažnju jer mogu biti ili posljedica grešaka u procesu sakupljanja podataka ili predstavnici vrlo rijetkih, atipičnih pojava. U prvom slučaju to omogućuje korekciju grešaka na nivou jednog primjera ili korekciju postupka sakupljanja podataka uopće. U drugom slučaju, pravilnom interpretacijom izuzetaka može se doći do vrijednog ljudskog znanja koje dobro nadopunjava interpretaciju općenitih koncepata koji se dobivaju kao rezultat modela.

Treći oblik su liste najznačajnijih atributa izdvojenih po nekom kriteriju.

6.1. Usporedba modela sa ekspertnim očekivanjem

Slaganje modela sa postojećim ljudskim znanjem samo potvrđuje postojeće znanje i nije osobito korisno za otkrivanje novog znanja. Ali može biti vrlo korisno kao potvrda da je naša praksa koja je služila kao osnova za sakupljanje podataka potpuno u skladu sa općenito prihvaćenim ili preporučenim ponašanjem u sličnim situacijama.

Neslaganje modela sa postojećim znanjem može ukazivati na nereprezentativan skup sakupljenih primjera, na neodgovarajući odabir atributa odnosno na greške u mjerenjima ili sakupljanju podataka. Ukoliko sa priličnom sigurnošću možemo isključiti ovakve probleme onda dobiveni model može predstavljati dobru osnovu za otkrivanje novog znanja.

6.2. Logička i statistička povezanost ne znači uzročno-posljedičnu povezanost

Povezanost vrijednosti atributa i klasa ciljnog atributa dobivena kao rezultat modeliranja znači samo njihovu logičku ili statističku povezanost. Značenje uzročno-posljedične povezanosti može im dati isključivo ljudska interpretacija na osnovi ekspertnog znanja o domeni. Modeli u nikom slučaju ne sugeriraju niti dokazuju uzročno-posljedičnu povezanost.

Prethodna tvrdnja je točna čak i kada postoji vremenski razmak u sakupljanju podataka odnosno značenju atributa. To što se neka pojava javila ranije u nikom slučaju ne znači da je uzrokovala povezanu pojavu koja se pojavila kasnije. Moguće objašnjenje je da postoji treća, neprimijećena pojava a da su dvije koje smo mi primijetili i zabilježili obje samo njene posljedice. Primjer je da neka bolest izaziva prvo povišenu temperaturu a kasnije i osip na koži. Nepravilna interpretacija je da povišena temperatura uzrokuje osip iako su obje statistički povezane a povišena temperatura se pojavljuje ranije od osipa.

6.3. Formuliranje znanja

Dobivenu povezanost između vrijednosti atributa i ciljnih klasa treba izraziti u ljudima prihvatljivom obliku i formi standardnoj za ljudsku komunikaciju u domeni istraživanja. Na primjer, model AKO 'hipertrofija lijeve klijetke' jednako 'da' ONDA je klasa 'ishemija srca' može se izraziti kao „Rizična skupina za ishemiju srca su osobe sa hipertrofijom lijeve klijetke“.

Uvijek treba nastojati da se rezultat izražava jasno i precizno kako bi znanje bilo korisno i provjerljivo.

6.4. Statistička provjera rezultata

Korisno je dobivene rezultate potkrijepiti statističkim rezultatima. Na primjer, prethodno otkrivenu relaciju možemo opisati činjenicom da na skupu za učenje sa kojim raspolažemo 85% bolesnika koji imaju hipertrofiju lijeve klijetke imaju također i ishemiju srca a od ukupnog broja bolesnika sa dijagnozom ishemije srca njih 35% ima i hipertrofiju lijeve klijetke.

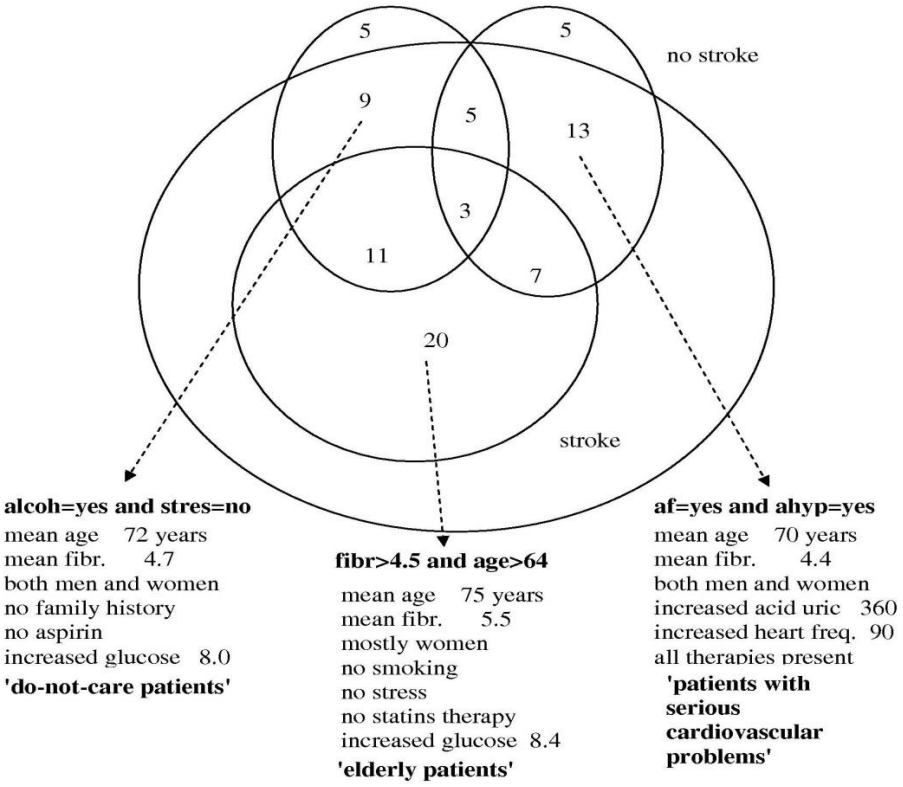
Ovakav tip statističke analize objašnjava zašto je korišteni sustav za strojno učenje povezao svojstvo hipertrofije i ishemije. Međutim on nije potvrda ispravnosti cijelog postupka otkrivanja znanja jer su i indukcija modela i statistička karakterizacija rezultata zasnovani na istim podacima. Jedina ispravna provjera je ona napravljena na *nezavisnim podacima* sakupljenim neovisno o podacima koji su korišteni za modeliranje i interpretaciju modela. Jedino takva provjera omogućuje stvarnu potvrdu kvalitete rezultata otkrivanja znanja.

6.5. Statistička karakterizacija modela

Cilj modeliranja je izgradnja jednostavnih i korisnih modela koji dobro razlikuju pojedine klase primjera. Ali za ljudsko razumijevanje modela važna je i informacija koja detaljnije opisuje modele i sa atributima koji nisu korišteni u direktnoj specifikaciji modela. Ona se može dobiti relativno jednostavnom statističkom analizom modela. Za ovaj tip analize može se koristiti isti skup podataka na kojem je model induciran.

Za primjer je uzeta domena moždanog udara u kojoj treba razlikovati bolesnike sa CT potvrđenim moždanim udarom od onih koji su imali ozbiljne ali samo prolazne ili subjektivne simptome moždanog udara. Model dobiven DMS sustavom se sastoji od sljedeća tri pravila:
A: 'moždani udar' AKO bolesnik nije pod stresom i sklon je konzumiranju alkohola.
B: 'moždani udar' AKO 'fibrinogen' veći od 4.5 mmol/l i 'starost' veća od 64 godine
C: 'moždani udar' AKO 'atrijska fibrilacija' 'da' i bolesnik koristi antihipertenzive

Svako od spomenutih pravila je točno za otprilike 30-40% osoba koje su doživjele moždani udar. Iako se radi o relativno kvalitetnim pravilima, njihova ekspertna interpretacija je teška bez dodatne statističke analize. Analiza je napravljena za svako pravilo posebno i to tako da su komparativno uspoređene podskupine bolesnika koji zadovoljavaju pravilo i stvarno su imale moždani udar (dio stvarno pozitivnih primjera) sa osobama za koje znamo da nisu imali moždani udar (svi negativni primjeri).



Slika 6. Prikaz pokrivanja primjera modelima, preklapanja modela te prikaz njihova značenja dobiven indukcijom pravila. Prikazan je rezultat za domenu moždanog udara

Statistička analiza uključuje računanje srednje vrijednosti za sve raspoložive atribute a ekspertnom evaluacijom izdvajaju se oni po kojima se ove tri podgrupe bolesnika

međusobno bitno razlikuju ili se one bitno razlikuju od grupe negativnih primjera. Na taj način su izdvojene sljedeće karakteristike:
A: srednja starost 72 godine, srednja vrijednost fibrinogena 4.7, i muškarci i žene, nema pozitivne porodične anamneze, ne koriste aspirin, povišena glukoza na 8.0.
B: srednja starost 75 godina, srednja vrijednost fibrinogena 5.5, uglavnom žene, nepušači, bez stresa, bez terapije statinima, povišena glukoza na 8.4.
C: srednja starost 70 godina, srednja vrijednost fibrinogena 4.4, i muškarci i žene, povišena mokraćna kiselina na 360 mmol/l, povišena frekvencija srca na 90, prisutno mnogo raznih terapija.

Povezivanjem svojstava koji su uključeni u opis modela i onih dobivenih statističkom analizom modela došlo se do interpretacije da: pravilo B predstavlja prvenstveno starije osobe (žene), pravilo A osobe koje pretjerano ne vode brigu o svom zdravlju (ne koriste lijekove ali konzumiraju alkohol) a pravilo C osobe sa ozbiljnim kardiovaskularnim problemima (koriste puno različitih lijekova a imaju pozitivne laboratorijske nalaze).

Primjer pokazuje važnost statističke analize u interpretaciji modela dobivenih korištenjem postupaka strojnog učenja. Slika 6. prikazuje način da se tekstualni opis modela i faktora koji potvrđuju te modele može kombinirati sa grafičkim prikazom broja primjera koje modeli pokrivaju. Slika je iskorištena i za pokazivanje postotka primjera (osoba) koje pokriva više od jednog modela te se može zaključiti o njihovoj relativnoj neovisnosti.

6.6. Interpretacija značenja atributa

U interpretaciji modela treba neprestano imati na umu uvjete pod kojima su sakupljani primjeri. Tako na primjer, ako za klasu bolesnika dobijemo da oni imaju često normalne vrijednosti kolesterola, vrlo lako je moguće da to ne znači da bolest nije povezana sa vrijednošću kolesterola već upravo obratno. Liječnici, svjesni opasnosti od povišenog kolesterola tim bolesnicima, redovito prepisuju lijekove za snižavanje kolesterola pa je to razlog što oni nemaju povišene vrijednosti.

Drugi primjer je da među bolesnima može biti malo pušača ne zato jer bolest nije povezana sa pušenjem već zato jer su bolesnici na nagovor liječnika ili zbog subjektivnih poteškoća prestali pušiti.

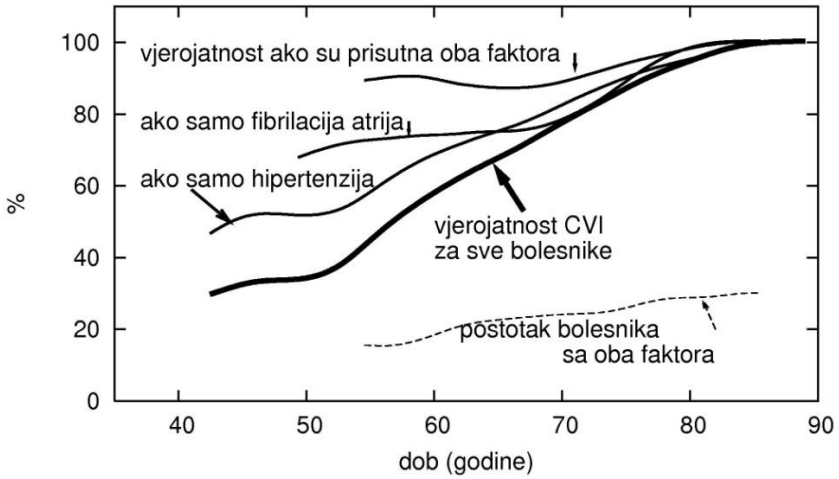
6.7. Skrivene veze među atributima

U otkrivanju skrivenih veza od presudne je važnosti ekspertno znanje čovjeka. Na primjer, ako se ustanovi da postoji povezanog povišenog krvnog tlaka sa većom tjelesnom težinom tada svakako treba provjeriti da li je to možda indirektna posljedica toga što muškarci imaju češće problema sa povećanim krvnim tlakom a muškarci su prosječno teži od žena. Ili je eventualno dominantno povećanje tlaka sa godinama starosti kada prirodno raste i tjelesna težina ljudi.

6.8. Kombinacije atributa

Povezanost ciljne klase sa vrijednostima pojedinih atributa se može otkriti i relativno jednostavnom statističkom analizom. Prava snaga primjene postupaka strojnog učenja je u mogućnosti učinkovitog otkrivanja značajnih kombinacija vrijednosti atributa. Njihovoj interpretaciji treba posvetiti posebnu pažnju.

Kada se inducirano pravilo sastoji od više uvjeta koji su u skladu sa ekspertnim očekivanjem, tada se njihova povezanost u pravilu interpretira kao posebna važnost njihove kombinacije. Tako je u domeni moždanog udara otkriveno pravilo koje povezuje istovremenu prisutnost atrijske fibrilacije i hipertenzije kod osoba koji su doživjeli moždani udar. Oba ova čimbenika poznata su kao faktori rizika a novost je u činjenici da je vjerojatnost moždanog udara posebno velika kada su oba prisutna.



Slika 7. Vjerojatnost pojave moždanog udara. Naglašena je znatno povećana vjerojatnost ako su prisutna i fibrilacija atrija i hipertenzija

Slika 7. prikazuje vjerojatnost moždanog udara za osobe sa hipertenzijom, za osobe sa fibrilacijom atrija te za one koji imaju i hipertenziju i fibrilaciju atrija. Sve vjerojatnosti su pokazane ovisno o dobi osoba. Iz slike je jasno vidljivo kako prisutnost oba rizična faktora znatno povećava rizik od moždanog udara. Crtkanom linijom u donjem desnom uglu pokazan je postotak osoba sa oba rizična faktora te je lako zaključiti na važnost otkrivene kombinacije.

6.9. Interpretacija prividnih kontradikcija

O kontradikcijama govorimo kada se vrijednosti atributa koji je povezan sa ciljnom klasom ne slaže sa ekspertnim znanjem. Prividna kontradikcija je kada se ona pojavljuje u kombinaciji sa

jednim ili više atributa koji nisu kontradiktorni ekspertnom znanju. Pažljiva interpretacija u takvim slučajeva je izuzetno važna.

Primjer iz tehničke prakse je da automobil ima ozbiljan kvar ako mu motor troši ulje a ulje ne kapa na površinu ispod motora. U tom pravilu kontradikcija je u činjenici da nekapanje ulja iz motora, koja je sama za sebe dobra vijest, u kombinaciji sa trošenjem ulja postaje izrazito loša vijest. Najme, ako motor troši ulje tako da ono kapa iz motora onda će to zahtijevati manji popravak a ako ono ne kapa, tada to upućuje na trošenje ulja unutar motora i na znatno ozbiljniji kvar.

Primjer prividne medicinske kontradikcije je pravilo: "U rizičnoj skupini za ishemiju srca su osobe iznad 53 godine sa povišenim kolesterolom (iznad 6.1) ali one koje nisu suviše debele (indeks tjelesne mase ispod 30)." U tom pravilu povišena starost i povišeni kolesterol su poznati rizici ishemije srca. Neočekivana je pojava snižene tjelesne mase jer je poznato da je upravo povišena težina faktor rizika za ovu bolest. Pravilna interpretacija ovakvog rezultata je sljedeća: Poznato je da je povišeni kolesterol i povišena tjelesna težina faktori rizika za ishemiju srca kao i to da debljina utječe na povišenje kolesterola. Pravilo pokazuje da mogu postojati i drugi uzročnici povišenog kolesterola te da su upravo u starijoj životnoj dobi oni opasniji za rizik srčane ishemije od slučaja kada je povišeni kolesterol posljedica debljine.

6.10. Interpretacija niza pravila

Kada se isti ili slični atributi pojavljuju u više pravila tada je moguće povezati njihovu interpretaciju. Primjer iz domene moždanog udara su dva inducirana pravila:
A: moždani udar AKO fibrinogen veći od 5.05.
B: moždani udar AKO fibrinogen veći od 3.85 i osoba nije pod stresom.

Pravilna interpretacija je da je povišeni fibrinogen sam po sebi vrlo opasan (interpretacija prvog pravila). Stres utječe na povišenje nivoa fibrinogena (ekspertno znanje) ali takav fibrinogen nije opasan kao onaj čiji nam uzrok nije poznat. Ovaj drugi dio interpretacije slijedi iz drugog pravila u kojem je kritična vrijednost fibrinogena bitno niža.

6.11. Interpretacija graničnih vrijednosti

Značajna karakteristika modela dobivenih strojnim učenjem je da uvjeti zasnovani na numeričkim atributima uključuju granične vrijednosti za pozitivnu odluku. U prethodnom primjeru imamo granice fibrinogena od 5.05 odnosno 3.85. Isto tako u istoj domeni imamo graničnu vrijednost za mokraćnu kiselinu veću od 384 mmol/l i sistolički tlak veći od 155mmHg.

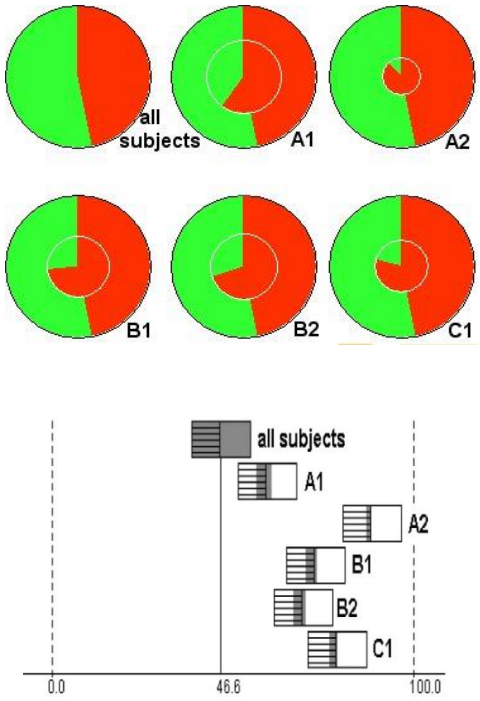
Važnost ovih graničnih vrijednosti je što proizlaze direktno iz sakupljenih podataka. U slučaju kada su granične vrijednosti u skladu sa postojećim ljudskim znanjem ili iskustvom, dobiveni rezultat dokazuje reprezentativnost sakupljenih primjera. U slučaju kada su odabrane

vrijednosti različite od očekivanja tada njihova ekspertna interpretacija, po potrebi uz dodatnu evaluaciju, može predstavljati važno novo znanje.

Svakako treba uočiti da se odabrane granične vrijednosti mogu razlikovati od pravila do pravila, kao u slučaju fibrinogena u prethodnom primjeru. To je posljedica činjenice da su odabrane granice prilagođene pravilu u cjelini i predstavljaju specifičnost i vrijednost primjene postupaka strojnog učenja. U našem primjeru to znači da granična vrijednost fibrinogena ovisi o starosti osobe odnosno o prisutnosti stresa što je različito od uobičajenog medicinskog pristupa koji nastoji definirati jednu graničnu vrijednost primjenjivu u svim situacijama.

6.12. Vizualizacija rezultata

Vrlo važno je dobivene rezultate nastojati ilustrirati grafički jer je to dobar način prenosa otkrivenog znanja drugim ljudima. Na taj način moguće je prikazati i kvalitetu otkrivenih modela i novo otkrivene odnose u podacima. Ovo posljednje već je ilustrirano slikama 6 i 7 dok slike 8 i 9 pokazuju neke načine prikaza kvalitete modela.



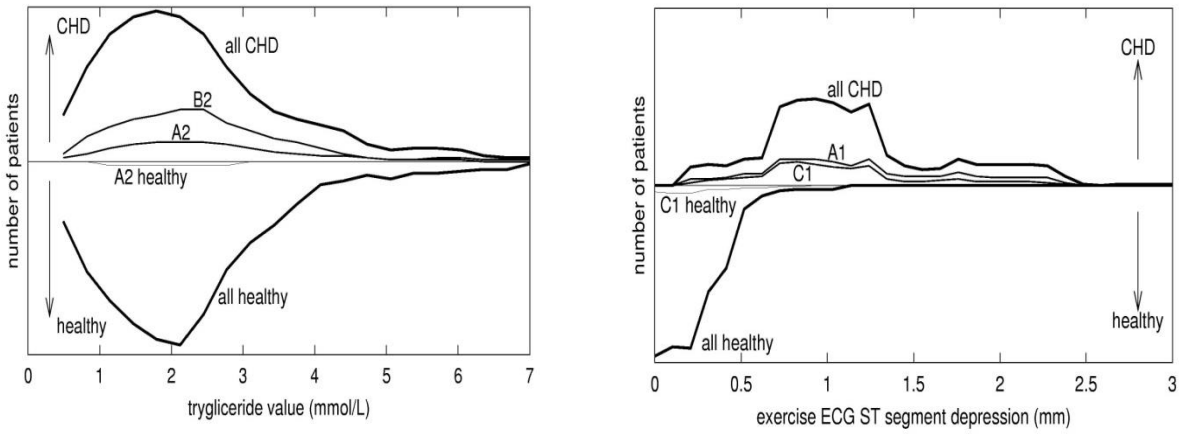
Slika 8. Načini prikaza kvalitete induciranih modela

Slika 8. prikazuje kvalitetu ukupno 5 modela označenih sa A1-C1. Na lijevoj slici veliki krug prikazuje sve primjere korištene u analizi a manji krug dio primjera koji zadovoljavaju uvjete

modela. Sa svjetlijom (zelenom) bojom prikazani su negativni a sa tamnijom (crvenom) bojom pozitivni primjeri. Iz slike je vidljivo da modeli koji pokrivaju manji broj primjera (na primjer A2) mogu pokrivati skoro samo primjere jedne klase (tamne-crvene) dok općeniti modeli, kao na primjer A1 i B2, imaju manju homogenost jedne klase ali da je ta homogenost još uvijek bitno veća od homogenosti cijelog skupa primjera (prikazano krugom „all subjects“).

Slika 8. donji dio prikazuje istu informaciju za modele A1-C1 tako da položaj kvadratića pokazuje pozitivnu prediktivnu vrijednost modela (model je više desno što je ta vrijednost veća) a njegov središnji zatamnjeni dio pokazuje ukupan dio primjera koje model zadovoljava.

Slika 9. prikazuje da se kvaliteta modela u obliku pokrivanja primjera može prikazati i ovisno o nekoj varijabli od posebnog interesa. Na osi x je odabrana varijabla a na osi y broj primjera (osoba) koji zadovoljavaju određene modele odnosno, svi primjeri koji su bili raspoloživi za analizu. Broj primjera jedne (pozitivne) klase prikazuje se iznad osi x a broj primjera druge (negativne) klase ispod nje. Usporedbom krivulja iznad i ispod osi x može se zaključiti na razlike koje postoje između klasa a zatim na položaj primjera koje pokriva promatrani model s obzirom na varijablu koja je na osi x.



Slika 9. Prikaz kvalitete pokrivanja modela ovisno o vrijednosti triglicerida (lijevo) i spuštanju ST spojnice (desno) za bolesnike sa srčanom ishemijom (iznad osi x) i zdrave osobe (ispod osi x)

7. Postupci strojnog učenja

Strojnim učenjem se naziva svaki računarski postupak koji omogućuje da se buduće djelovanje unaprijedi na osnovi informacija iz prošlosti. U svom najosnovnijem obliku strojno učenje treba omogućiti pravilno klasificirati primjere. Za to je potreban skup primjera za koje je poznata klasifikacija i koji služe za izgradnju modela. Nakon toga se model može koristiti za klasifikaciju primjera za koje ona nije poznata. Kvaliteta učenja ocjenjuje se na osnovi točnosti koju izgrađeni model postiže na testnom skupu primjera koji nije bio dio skupa za učenje [3,4].

Za otkrivanje znanja važni su postupci strojnog učenja koji generiraju modele u obliku razumljivom za čovjeka. Generirani modeli se u principu niti ne koriste za predviđanje klase budućih primjera već se interpretacijom modela dolazi do znanja koje predstavlja novu vrijednost. To ne znači da prediktivna kvaliteta nije važna: modeli sa boljim prediktivnim svojstvima ukazuju na čvršću povezanost klasifikacije sa atributima koji ih opisuju te će i znanje formirano na toj osnovi biti pouzdanije i značajnije.

U ovom poglavlju prikazana su dva postupka klasifikacijskog modeliranja te jedan neklasifikacijskog modeliranja na osnovi transakcijskih primjera.

7.1. Učenje pravila koji opisuju važne podgrupe primjera

Ovaj postupak strojnog učenja poznat kao otkrivanje podgrupa (engl. *subgroup discovery*) značajan je jer je posebno prilagođen potrebama otkrivanja znanja. On predstavlja metodološku osnovu DMS poslužitelja [5].

Otkrivanje podgrupa kao i drugi postupci učenja pravila [6] prikazuje inducirane modele u obliku skupa pravila. Svako pravilo ima oblik AKO su zadovoljeni uvjeti ONDA je primjer u određenoj klasi. Uvjeti (engl. *condition*) se u stručnoj terminologiji strojnog učenja nazivaju i literalima (engl. *literal*). Svaki literal ispituje vrijednost jednog ili više atributa primjera te na toj osnovi ima vrijednosti istino i neistino. Primjer literala je 'visina veća od 5' i 'boja je crvena'.

S ciljem jednostavnije ljudske interpretacije pravila koriste samo logičku operaciju I (engl. *AND*). Takav oblik naziva se konjunkcija uvjeta. Primjer pravila je

„AKO (osoba je muškog spola) I (prihod manji od 15.000) I (osoba nije visokog obrazovanja) ONDA klasa pušač“.

U prethodnom primjeru literali su prikazani u okruglim zagradama. Pravilo je zadovoljeno ili istinito ako su istiniti svi literali koji čine uvjetni AKO dio pravila.

Model se sastoji od jednog ili više pravila. Model je ispunjen ili istinit ako je zadovoljeno barem jedno od pravila. To znači da su pravila međusobno povezana logičkom ILI operacijom koja se naziva i disjunkcija, Primjer modela od tri pravila je:

A: AKO (osoba je muškog spola) I (prihod manji od 15.000) ONDA pušač
B: AKO (osoba je student) I (barem jedan od roditelja je pušač) ONDA pušač
C: AKO (osoba je često pod stresom) I (sistolički tlak je manji 110) I (osoba je mršava sa indeksom tjelesne mase manjim od 20) ONDA pušač.

Za razliku od drugih postupaka učenja pravila, otkrivanje podgrupa ne izgrađuje prvenstveno pravila visoke prediktivne točnosti već relativno kratka pravila koja opisuju podskupine primjera koje se bitno razlikuju od primjera suprotne klase. Postupak se može koristiti samo za indukciju modela u skupovima primjera sa dvije klase a višeklasni problemi se moraju transformirati u niz problema sa dvije klase.

7.1.1. Postupak formiranja literala

Literali se formiraju različito ovisno da li je atribut nominalan ili numerički.

Za nominalne atribute formiraju se svi literali oblika atribut je jednak ili različit od neke vrijednosti. Za svaki pojedini nominalni atribut, na primjer atribut A1, svi potencijalno zanimljivi literali se dobiju tako da se promatraju vrijednosti koje taj konkretni atribut poprima u raspoloživom skupu primjera. Literali oblika (A1=x) se dobiju tako da se za vrijednost 'x' redom stave sve vrijednosti koji se pojavljuju u pozitivnim primjerima a literali oblika (A1#x) se dobiju tako da se za vrijednosti 'x' stave sve vrijednosti koje se pojavljuju u negativnim primjerima.

Primjer
Imamo sljedeći skup za učenje koji se sastoji od 4 primjera sa 4 atributa. Dva primjera su u pozitivnoj klasi a dva primjera u negativnoj klasi:

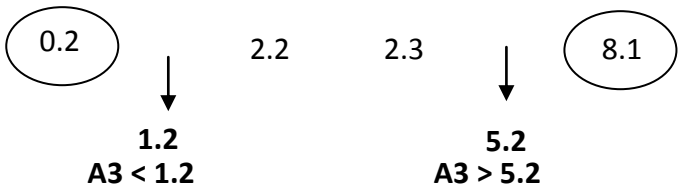
	A1	A2	A3	A4	klasa
prim1	a	b	0.2	5	poz.
prim2	a	c	8.1	3	poz.
prim3	c	d	2.2	9	neg.
prim4	d	d	2.3	2	neg.

U ovom primjeru postoje dva nominalna atributa. Za prvi se formiraju sljedeći literali: (A1=a), (A1#c) i (A1#d) a za drugi atribut literali: (A2=b), (A2=c) i (A2#d).

Za kontinuirane numeričke atribute formiraju su literali oblika veće ili manje od neke vrijednosti. Za svaki pojedini atribut, na primjer A3 u gornjem primjeru, uzimaju se sve vrijednosti koje se pojavljuju u skupu za učenje te se one slažu u rastući niz. U tom rastućem nizu traže se susjedna mjesta gdje je sa lijeve strane vrijednost u negativnoj klasi, koju

označimo sa x_n a sa desne vrijednost u pozitivnoj klasi koju označimo sa x_p. Za sva takva mjesta formiramo literale ($A3 > (x_p-x_n)/2$). Isto tako, u tom istom rastućem nizu traže se susjedna mjesta gdje je vrijednost iz pozitivne klase x_p lijevo a vrijednost iz negativne klase x_n desno. Za takve parove formiramo literale ($A3 < (x_n-x_p)/2$). Znači da su numeričke vrijednosti koja se pojavljuje u literalima odabrane tako da su na pola udaljenosti između susjednih vrijednosti u različitim klasama sa ciljem da što bolje omoguće odvajanje primjera u tim klasama.

Primjer
Slika 10. prikazuje vrijednosti za atribut A3 iz prethodnog primjera. Vrijednosti su posložene u rastući niz i to tako da su vrijednosti u pozitivnoj klasi zaokružene. Vidljivo je da vrijednosti '1.2' i '5.1' koje su označene strelicama, leže na pola puta između vrijednosti u različitim klasama. Zbog toga će one u konkretnom primjeru generirati literale A3 < 1.2 i A3 > 5.1.



Slika 10. Postupak generiranja literala za kontinuirane numeričke atribute

Za diskretne atribute literali se formiraju kao da su istovremeno i nominalni i kontinuirani. Za primjer atributa A4 iz gornjeg primjera formirati će se sljedeći literali: (A4=3), (A4=5), (A4#2), (A4#9), (A4>2.5) i (A4<7).

7.1.2. Literali imaju logičke vrijednosti

Osnovno svojstvo literala je da su oni logičke varijable čija vrijednost je ili istinita ili neistinita. Po tome se oni bitno razlikuju od atributa čije vrijednosti su ili numeričke ili nominalne. Čak i kada atribut ima samo dvije nominalne vrijednosti, pa čak i kada se one zovu 'istinito' i 'neistinito', to je bitno drugačije od literala. Treba imati na umu da se i za atribut sa samo dvije nominalne vrijednosti mogu formirati četiri različita literala.

7.1.3. Literali su gradivni materijal za pravila

Literali, zato što su logičke varijable, mogu se povezivati logičkim operacijama I (engl. *AND*) i ILI (engl. *OR*). Povezivanjem nastaju složeniji izrazi koje nazivamo pravilima. I cijela pravila, bez obzira kako složena bila, su logičke varijable sa vrijednostima istinito ili neistinito. U logičkom smislu pravila i literali su ekvivalentni pojmovi. Pravilo se može sastojati i od samo jednog literala.

U učenju podgrupa pravila se formiraju samo povezivanjem literala logičkom operacijom I. Složeniji modeli se dobivaju tako da se formira više pravila a oni se povežu logičkom operacijom ILI. Znači da je rezultat modela istinit ako je istinito bilo koje od pravila koje ga sačinjavaju. Takvim pristupom može se realizirati po volji složena logička formula. Cijeli model može biti ili istinit ili neistinit te i za njega vrijede ista logička pravila kao i za pojedina pravila odnosno literale.

7.1.4. Tablica istinitosti

Ako je neki logički izraz, bez obzira da li se radi o literalu, pravilu ili modelu, istinit za neki primjer onda kažemo da ga pokriva. Cilj dobrog klasifikatora je pronaći izraz koji dobro razlikuje primjere pozitivne klase od primjera negativne klase. Drugim riječima, cilj nam je naći izraz koji pokriva čim više pozitivnih primjera a istovremeno pokriva čim manji broj negativnih primjera. U idealnom slučaju naš logički izraz će biti istinit za sve pozitivne a neistinit za sve negativne primjere.

Osnovne mjere kvalitete logičkih izraza su:
TP (engl. *true positives*) - broj primjera pozitivne klase za koje je izraz istinit (dobro)
TN (engl. *true negatives*) - broj primjera negativne klase za koje izraz nije istinit (dobro)
FP (engl. *false positives*) - broj primjera negativne klase koji za koje je izraz istinit (loše)
FN (engl. *false negatives*) - broj primjera pozitivne klase za koje izraz nije istinit (loše).

Osnovne mjere kvalitete uobičajeno je prikazati u tablici istinitosti (engl. *confusion matrix*).

istinitost izraza	stvarna klasa	
	pozitivna	negativna
istinit	**TP**	**FP**
neistinit	**FN**	**TN**

Ako sa P označimo broj pozitivnih primjera, sa N broj negativnih primjera te sa E ukupan broj primjera (E=P+N), tada vrijedi da je TP+FN=P a TN+FP=N.

Izvedene mjere kvalitete logičkih izraza su:

- senzitivnost (osjetljivost) TP/P
- specifičnost TN/N
- točnost (TP+TN)/E
- pozitivna prediktivna vrijednost (engl. *positive predictive value, precision*) TP/(TP+FP)
- poboljšanje (engl. *lift*) TP*E/((TP+FP)*P)

Koja od mjera će se koristiti u određenom slučaju ovisi o tome koje od svojstava klasifikatora se želi naglasiti. Točnost je dobra kao integralna mjera ako je podjednak broj pozitivnih i negativnih primjera te ako je opasnost od nepravilne klasifikacije i pozitivnih i negativnih

primjera podjednaka. U medicini je praksa kvalitetu izražavati kombinacijom senzitivnosti i specifičnosti a o određenom slučaju onda ovisi da li će se cijeniti klasifikatori koji su bolji u pogledu senzitivnosti ili oni koji su bolji po specifičnosti. U marketinškim domenama uobičajeno je korištenje poboljšanja jer ono direktno izražava relativni dobitak korištenja klasifikatora prema slučaju kada nemamo nikakav klasifikator.

7.1.5. Odabir relevantnih literala

Iz postupka formiranja literala je vidljivo da se za svaki atribut formira više literala te da skup svih potencijalno formiranih literala može biti vrlo velik. Između njih treba odabrati samo one koji predstavljaju relevantne odnose između primjera u različitim klasama. Na primjer, ako skup za učenje ima 50 pozitivnih i 50 negativnih primjera kao relevantne možemo uzeti samo one literale koji korektno razdvajaju barem 10 pozitivnih od barem 10 negativnih primjera. Sve one koji ne zadovoljavaju ovaj uvjet, na primjer literal koji ne pokriva samo 9 negativnih primjera bez obzira za koliko pozitivnih primjera on bio točan, ćemo izostaviti iz skupa relevantnih literala. Ovakav tip relevantnosti se naziva apsolutna relevantnost literala.

Drugi tip je relativna relevantnost. Kažemo da je literal L1 relevantniji od literala L2 ako razdvaja barem sve parove pozitivnih i negativnih primjera kao i L2 te eventualno i još neke druge. Sve literale za koji postoji neki drugi relevantniji literal također izostavljamo is skupa literala. Rezultat tog procesa je da u skupu literala ostaje relativno mali broj literala koji najbolje, za dani skup za učenje, razdvajaju pozitivne od negativnih primjer. Time što svaki od njih omogućuje razlikovanje puno pozitivnih od puno negativnih primjera ostvarena je jedna od pretpostavki da pravila koja ćemo formirati sa takvim literalima imaju priliku dobro razlikovati i primjere koji nisu uključeni u skup za učenje.

7.1.6. Formiranje pravila

Ako neki literal pokriva sve pozitivne primjere a istovremeno ne pokriva niti jedan negativni primjer tada je on idealno rješenje koje predstavlja cijeli model i optimalan klasifikator. U praksi je takav slučaj rijedak. Zato se kombinacijom više literala pokušavaju izgraditi pravila koja imaju svojstva bolja od pojedinih literala. Postupak kreće sa literalom koji je najbolje kvalitete a onda mu se pokušava dodati neki drugi literal u logičkoj I kombinaciji koji čim više podiže njegovu kvalitetu. Postupak se ponavlja dok je god moguće pronaći literal koji može poboljšati kvalitetu cijele kombinacije.

Kvaliteta rezultata se osigurava time što se koriste samo relevantni literali te što i svaka kombinacija literala mora zadovoljavati uvjet relevantnosti. Osim toga, ne kreće se sa jednim literalom već ih se uzima više te se u svakoj iteraciji čuva više najboljih kombinacija literala. Proces završava kada se niti jedno od međurješenja ne može poboljšati a konačno rješenje je ona kombinacija među svim međurješenjima koja je najveće kvalitete.

Kvaliteta literala, njihovih kombinacija i cijelih pravila u postupku otkrivanja znanja mjeri se izrazom TP/(FP+g). Ovako definirana kvaliteta je to veća što izraz pokriva veći broj pozitivnih

primjera (velik TP) odnosno što pokriva manji broj negativnih primjera (mali FP). Parametar g omogućuje dobivanje cijelog spektra rješenja, od vrlo specifičnih do vrlo senzitivnih pravila. Sa stanovišta strojnog učenja ona su sva jednako vrijedna ali im važnost i primjenjivost u konkretnoj domeni može biti bitno različita. Za velike vrijednosti parametra g (10-100) moguće je očekivati općenita pravila koja pokrivaju velik broj primjera a za njegove male vrijednosti (1-10) precizna pravila koja pokrivaju mali broj primjera.

7.1.7. Formiranje skupa pravila

Ako formirano pravilo predstavlja idealan klasifikator onda je ono i konačni rezultat modeliranja. Ako to nije slučaj, tada se može pokušati poboljšati kvaliteta modela tako da se formira dva ili više pravila koji zajedno čine model.

Postupak se izvodi tako da na početku svi primjeri imaju jednaku važnost. Nakon formiranja prvog pravila potraži se skup pozitivnih primjera koje pokriva to pravilo i smanji im se važnost. U drugoj iteraciji ponavlja se isti postupak sa razlikom da se u mjeri za kvalitetu literala, međurješenja i konačnog rješenja TP/(FP+g) umjesto vrijednosti TP uzima suma važnosti pozitivnih primjera koje pokriva izraz. Time se optimizira prvenstveno pokrivanje važnih, do tada prethodnim pravilom nepokrivenih pozitivnih primjera. Nakon formiranja drugog pravila smanjuje se važnost pozitivnim primjerima koje on pokriva. Postupak se može ponoviti više puta dok svi pozitivni primjeri nisu pokriveni barem sa jednim pravilom.

Pravila su unutar modela povezana ILI logičkom operacijom što znači da model pokriva primjer ako ga pokriva barem jedno od pravila. Pozitivan primjer će biti dobro klasificiran ako ga dobro klasificira bilo koje pravilo a negativni primjer samo ako ga dobro klasificiraju sva pravila. Zato će kombinacija pravila u modele biti uspješnija ako pravila pokrivaju vrlo malo negativnih primjera a to se postiže korištenjem niske vrijednosti parametra generalizacije.

7.1.8. Otkrivanje šuma

Šum (engl. *noise*) u podacima je definiran kao primjeri koji iz bilo kojeg razloga odstupaju od karakteristika većine ostalih primjera. Uzroci mogu biti ili u procesu sakupljanja, zbog grešaka bilo koje vrste, ili zato što su ti primjeri stvarni izuzeci kao predstavnici vrlo rijetkih ali realnih pojava. U realnim primjenama, uključujući i medicinske, česte su pojave oba tipa šuma u podacima [7].

Otkrivanje i eliminacija šuma je važna jer je poznato da postojanje šuma može bitno smanjiti kvalitetu induciranih modela. U otkrivanju znanja eksplicitna detekcija šuma je još i bitno važnija jer omogućuje otkrivanje rijetkih koncepata čija argumentirana ekspertna interpretacija može biti izuzetno važna kao novo znanje.

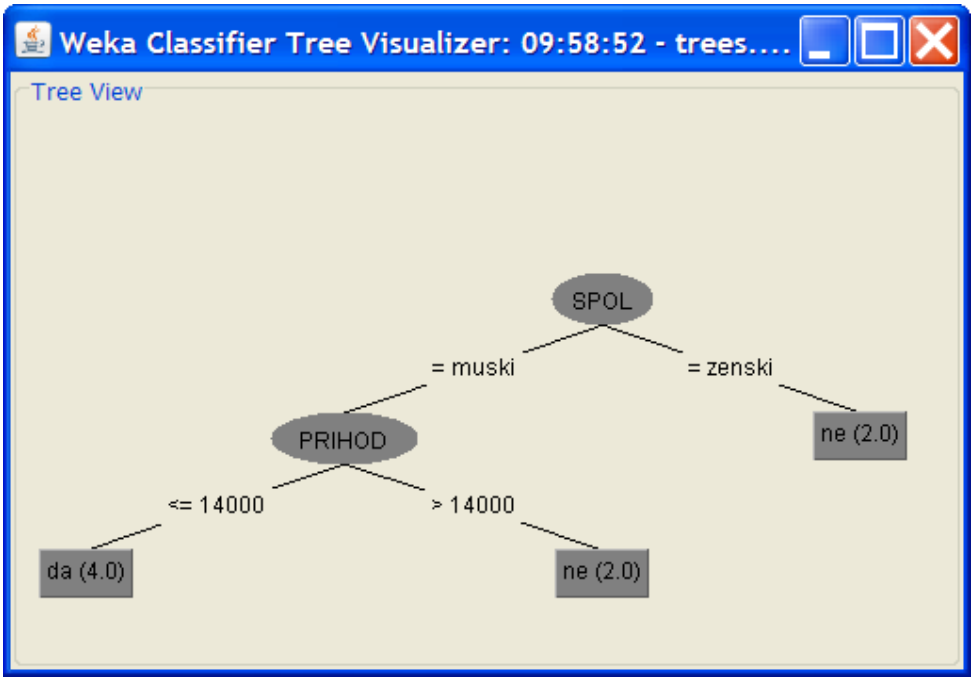
Unutar postupka otkrivanja podgrupa, detekcija šuma je zasebni postupak koji se po potrebi odvija nakon generiranja literala i eliminacije nerelevantnih literala a prije formiranja pravila kao kombinacije relevantnih literala. Postupak se zasniva na mogućnosti da se za svaki skup

primjera može odrediti minimalni skup literala dovoljan za konstrukciju potpuno korektnog modela. Koncept saturacije skupa za učenje pokazuje da će uključivanje šumnih primjera u skup za učenje nužno imati za posljedicu povećanje složenosti potpuno točnog modela za sve primjere a sa tim povezano i povećanje broja literala u minimalnom skupu. Na toj osnovi se otkrivanje šuma definira kao postupak otkrivanja onih primjera koji omogućuju smanjenje veličine minimalnog skupa literala nužnog za konstrukciju potpuno točnog rješenja [7].

7.2. Učenje stabla odlučivanja

Stabla odlučivanja su jedno od prvih i osnovnih postupaka strojnog učenja [8,9]. I danas je to popularan postupak zbog svoje jednostavnosti izvođenja a posebno zbog mogućnosti da se rezultat indukcije može grafički prikazati. Dodatna prednost je što se postupak može direktno koristiti za rješavanje problema više klasa. Slika 11. prikazuje stablo odlučivanja dobiveno korištenjem Weka sustava za 8 primjera pušača-nepušača opisanih u poglavlju 3.

U čvorovima stabla odlučivanja su imena atributa koji se koriste za odluku a na granama koje izlaze iz svakog čvora su uvjeti koje treba zadovoljiti vrijednost tog atributa da bi se primjer usmjerio tom granom. Grananje stabla završava sa čvorovima (listovima) koji određuju klasu kojoj pripadaju primjeri koji su s obzirom na vrijednosti svojih atributa razvrstani u taj određeni čvor. Pored imena klase takvi čvorovi sadrže i broj primjera skupa za učenje koji su razvrstani u taj čvor. Iz tih brojeva se može zaključiti koliko primjera je bilo razlog za generiranje uprave te grane.



Slika 11. Stablo odlučivanja za razlikovanje pušača od nepušača za skup podataka opisan u poglavlju 3

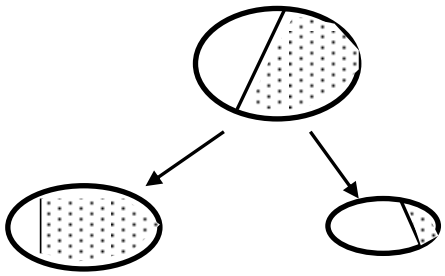
Na slici 11 u korijenu imamo čvor sa atributom „SPOL“ iz kojeg izlaze grane „=zenski“ i „=muski“. U prvom slučaju grana odmah završava krajnjim čvorom jer su svi primjeri koji imaju ženski spol u istoj klasi. Klasa tog krajnjeg čvora je 'ne' a podatak koji slijedi „(2,0)“ kazuje da imamo dva primjera čiji spol je ženski i koji su nepušači i niti jedan primjer čiji spol je ženski a koji je pušač.

Granom „=muski“ dolazimo u čvor odluke u kojem je atribut „PRIHOD“. Iz njega izlaze dvije grane i to „<=14000“ koja završava u krajnjem čvoru 'da' sa karakteristikom „(4,0)“ i grana „>14000“ koja završava u krajnjem čvoru 'ne' sa karakteristikom „(2,0)“. Ova potonja znači da u njoj završavaju muškarci koji imaju visok prihod, da su oni nepušači i da su takva dva primjera. Prva grana predstavlja odredište muškaraca sa niskim prihodima koji su pušači i kojih ima ukupno 4 primjera.

Interpretacijom ovakvog stabla vidimo da je dobiveni rezultat vrlo sličan onom dobivenom sa indukcijom pravila. I u jednom i u drugom slučaju su pušači muškarci malih prihoda. Razlika je jedino u vrijednosti granice prihoda koji se smatraju niskima. To je posljedica razlika u načinu rada pojedinih postupaka. U slučajevima sa velikim brojem primjera razlike između rezultata mogu biti bitno veće.

7.2.1. Postupak izgradnje stabla

Postupak započinje od skupa primjera koji su u različitim klasama i koji su opisani nekim atributima. U korijen stabla ćemo odabrati onaj atribut čije grane će omogućiti da su primjeri razvrstani tako da su dobiveni podskupovi primjera svi u istoj klasi. Ako to nije moguće, onda nam je cilj da podskupovi budu čim homogeniji s obzirom na klase kojima pripadaju.



Slika 12. Postupak dijeljenja skupa primjera pri izgradnji stabla odlučivanja. Izabire se atribut koji omogućuje da su manji podskupovi primjera čim homogeniji

Slika 12. prikazuje kako neki atribut dijeli početni skup primjera dvije klase (točkastih i bijelih primjera) u dva manja podskupa od kojih prvi sadrži većinu primjera prve klase a drugi većinu primjera druge klase. Rezultat dijeljenja se zatim nastoji poboljšati tako da se odabere sljedeći atribut koji će za svaki podskup dodatno moći povećati homogenost primjera. Postupak završava kada su u podskupovima svi primjeri iste klase ili nije moguće pronaći atribut koji može poboljšati konačno rješenje.

Kvaliteta homogenosti skupa primjera mjeri se entropijom. Za slučaj dvije klase ona je definirana izrazom $E = -(p_1 * \log_2(p_1) + p_2 * \log_2(p_2))$. U tom izrazu vrijednost p_1 je dio primjera koji je u klasi 1 a p_2 je dio primjera u klasi 2. Kada imamo jednak broj primjera u obije klase tada je $p_1 = p_2 = 0.5$, $\log_2(p_1) = \log_2(p_2) = -1$, a odgovarajuća entropija je E=1. Ako su svi primjeri u prvoj klasi onda je $p_1 = 1$ a $p_2 = 0.0$, $\log_2(p_1) = 0.0$ te E=0. Za sve ostale slučajeve je vrijednost entropije između 0 i 1 i ona je to manja što je homogenost veća.

Između svih mogućih atributa postupak izgradnje stabla odlučivanja izabire onoga koji omogućuje najmanju prosječnu entropiju za sve podskupove koji se generiraju. Ako niti jedan atribut ne može smanjiti entropiju početnog skupa primjera, postupak završava.

Za numeričke atribute odabire se granična veličina koja dijeli skup primjera u dvije podskupine. Između svih mogućih vrijednosti ona se izabire tako se maksimizira smanjivanje entropije. Potencijalno najvažnije vrijednosti su one koje leže između dvije vrijednosti koje se pojavljuju u primjerima različitih klasa. Njihov odabiran je sličan postupku za formiranje literala pri učenju pravila koji je opisan u poglavlju 7.1.1.

Za nominalne atribute sa mnogo različitih vrijednosti moguće je lakše postići homogenost rezultirajućih podskupova primjera jer se originalni skup podataka dijeli u velik broj malih skupova. U takvim slučajevima se postignuta homogenost dodatno umanjuje sa logaritmom broja različitih vrijednosti. Alternativno, diskretne vrijednosti se grupiraju i to tako da osiguraju najveću homogenost rezultirajućih podskupova primjera.

Problem nepoznatih vrijednosti u primjerima rješava se tako da se takvi primjeri ne uzimaju u obzir pri računanju homogenosti podskupova ili se takvi primjeri raspoređuju u više podskupova i to proporcionalno raspodjeli primjera sa poznatim vrijednostima atributa.

7.2.2. Podrezivanje stabla

Jedan od osnovnih problema strojnog učenja je da se izgrađeni model prilagodi raspoloživim podacima tako dobro da izgubi svojstvo generalizacije. Gubitak generalizacije primjećuje se tako da model ima slabu prediktivnu kvalitetu na primjerima koji nisu bili dostupni u procesu izgradnje modela. Ovaj efekt se zove preprilagođenje (engl. *overfitting*) i glavni je uzročnik smanjenoj prediktivnoj kvaliteti induciranih modela. Preprilagođenje se može i mjeriti i jednako je razlici kvalitete predikcije nekog modela na primjerima koji su se koristili u postupku indukcije i onih koji su sakupljeni kasnije.

Problem preprilagođenja i načini njegova rješavanja su prvo i najcjelovitije istraženi kod izgradnje stabla odlučivanja. Najčešći oblik rješavanja tog problema je ograničavanje složenosti modela sukladno principu Occam-ove oštrice [10]. Za stabla odlučivanja postupak se naziva podrezivanje (engl. *prunning*) i sastoji se u tome da se ograniči složenost izgrađenog stabla. Složenost se može ograničiti u procesu formiranja stabla ili u naknadnom podrezivanju već izgrađenog stabla.

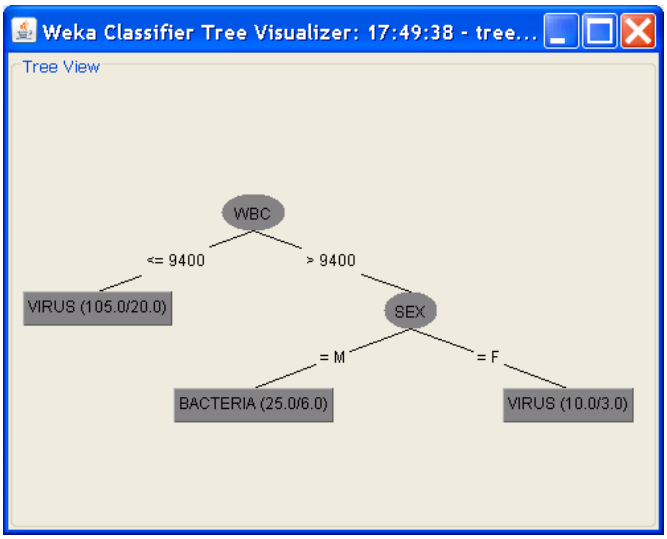
Najjednostavniji način ograničavanja složenosti stabla odlučivanja je da se ograniči najmanji broj primjera u nekom čvoru za čije dijeljenje je moguće uvesti novi atribut. To praktično znači da ako u nekom čvoru imamo 10 primjera pozitivne klase i jedan primjer negativne klase nećemo uvesti dodatni atribut koji bi omogućio idealno razdvajanje primjera ove dvije klase.

```
WBC <= 9400
|   ONSET = SUBACUTE
|   |   AGE <= 30: BACTERIA (2.0)
|   |   AGE > 30: VIRUS (3.0)
|   ONSET = ACUTE
|   |   FOCAL = minus: VIRUS (72.0/8.0)
|   |   FOCAL = plus
|   |   |   FEVER <= 4: BACTERIA (9.0/2.0)
|   |   |   FEVER > 4: VIRUS (16.0/2.0)
|   ONSET = CHRONIC: BACTERIA (1.0)
|   ONSET = RECURR: VIRUS (2.0)
WBC > 9400
|   SEX = M: BACTERIA (25.0/6.0)
|   SEX = F: VIRUS (10.0/3.0)
```

Slika 13. Tekstualni prikaz stabla odlučivanja za problem razlikovanja bakterijskog i virusnog meningitisa dobiven klasifikatorom J48 unutar Weka alata sa standardnom vrijednošću parametra podrezivanja (M=2)

Slika 13. prikazuje stablo odlučivanja dobiveno Weka sustavom za meningitis domenu. Cilj modela je razlikovanje bakterijskog i virusnog meningitisa za slučaj kada su isključeni neki od najznačajnijih atributa. Model je dobiven sa standardnim podrezivanjem ograničenim na čvorove sa samo dva primjera. U korijenu stabla je atribut WBC (broj bijelih krvnih zrnaca). Uočljiva je relativno velika složenost modela koja nije pogodan za grafički prikaz te koji može biti nespretan za ljusku interpretaciju.

Slika 14. prikazuje rezultat postignut za isti skup podataka uz omogućeno jače podrezivanje stabla. Rezultat je postignut tako da je zabranjeno dijeljenje čvorova sa 10 ili manje primjera. U alatu Weka klasifikator za učenje stabla odlučivanja se naziva J48 a spomenuti parametar koji određuje minimalnu vrijednost čvora koji se može dijeliti se naziva M.



Slika 14. Grafički prikaz stabla odlučivanja uz značajno podrezivanje. Ovaj rezultat je dobiven sa vrijednošću parametra minimalne veličine čvora koji se može dalje dijeliti jednakom 10 (parametar M za klasifikator J48)

Za ljudsku interpretaciju je značajno da je ovakvo stablo jednostavnije te da uključuje samo bitne atribute. Zato se u postupcima otkrivanja znanja redovito koristi veliko podrezivanje stabala odlučivanja.

U slučajevima kada se modeli trebaju koristiti za predikciju i važna je njihova prediktivna kvaliteta, tada se odabire ona vrijednost podrezivanja koja maksimizira prediktivnu točnost. To se postiže tako da se pokuša sa više različitih vrijednosti parametra M a prediktivna točnost se mjeri ili na nezavisnom testnom skupu podataka ili se procjenjuje postupkom unakrsne validacije (vidi poglavlje 8.1.2).

7.3. Učenje čestih uzoraka

Učenje čestih uzoraka (engl. *association rule learning*) je oblik učenja pravila koji se može koristiti i kada ciljni atribut nije poznat [11]. Ovaj pristup se može koristiti samo ako su primjeri prikazani kao skup stavki (engl. *item*). Slijedi skup podatka sa 5 primjera i stavkama prikazanim slovima a-h.

```
primjer1 a,b,c,e
primjer2 a,b,d,e,f
primjer3 b,e,g
primjer4 a,c,d,e,g,h
primjer5 a,d,e,f,h
```

Svakako treba uočiti da primjeri nisu prikazani atributima kao u klasifikacijskom učenju, da primjeri mogu imati različitu dužinu (broj stavki od kojih se sastoje), te da stavke ne mogu biti niti numeričke niti nominalne već samo ili uključene u primjer ili ne.

Rezultat indukcije su uzorci prikazani pravilima. Mjere kvalitete su podrška (engl. *support*) i pouzdanost (engl. *confidence*).

Podrška predstavlja učestalost uzorka i mjeri se omjerom broja primjera u kojima se pojavljuju sve stavke koji čine uzorak i broja svih primjera. U gornjem skupu primjera podrška za stavku 'a' je 4/5, za stavku 'b' 3/5 a za par stavki 'ab' podrška je 2/5. Pravilo „AKO a ONDA b“ i pravilo „AKO b ONDA a“ imaju istu podršku i ona je jednaka 2/5=40% kao i za par stavki 'ab' koje ga čine.

Pouzdanost se mjeri omjerom broja primjera u kojima se pojavljuju sve stavke koje čine uzorak i broja primjera u kojem se pojavljuju one stavke koje su u AKO dijelu pravila. Za pravilo „AKO a ONDA b“ par stavki 'ab' se pojavljuje 2 puta, samo stavka 'a' 4 puta pa je pouzdanost pravila jednaka 2/4 ili 50%. Za pravilo „AKO b ONDA a“ par stavki 'ab' se pojavljuje 2 puta, samo stavka 'b' 2 puta pa je pouzdanost pravila jednaka 2/2 ili 100%.

Pravilo „AKO b ONDA a“ znači da u primjerima u kojem se pojavi stavka 'b' možemo očekivati da se pojavi i stavka 'a'. To pravilo ima pouzdanost 100% i to znači da se na skupu za učenje u svakom primjeru u kojem se pojavila stavka 'b' obvezno pojavila i stavka 'a'. Takvo pravilo može predstavljati zanimljiv uzorak čija važnost u znatnoj mjeri ovisi o njegovoj učestalosti (podršci) koja u konkretnom slučaju iznosi 40%. Cilj indukcije je pronalaženje uzoraka (pravila) koja imaju istovremeno i visoku podršku i visoku pouzdanost.

7.3.1. Otkrivanje skupova stavki sa visokom podrškom

Prvi i osnovni korak otkrivanja čestih uzoraka je otkrivanje onih skupova stavki koji imaju visoku podršku. Nužan uvjet da bi neki skup stavki imao visoku podršku je da sve stavke koje su uključene imaju i pojedinačno visoku podršku. Ako neka stavka ima niku podršku, manju od granične vrijednosti koju odaberemo kao minimalno zanimljivu, tada možemo biti sigurni da ona ne može sudjelovati niti u jednom skupu koji će imati visoku podršku. Isto tako, i par stavki koji ima nisku podršku se ne može pojaviti u nekom većem skupu koji ima visoku podršku.

	stavke i njihovi skupovi				
	podrška = 1/5	podrška = 2/5	podrška = 3/5	podrška = 4/5	podrška = 5/5
osnovni skup stavki		c, f, g, h	b, d	a	e
parovi	.....	ab, ac, af, ah, be, ce, de, df, dh, ef, eg, eh	**ad, de**	**ae**	-
trojke	.....	def, deh	**ade**	-	-
četvorke	.....	adef, adeh	-	-	-

niska podrška ⟵ ⟶ ***visoka podrška***

Tablica 1. Postupak otkrivanja skupova čestih stavki. Masnim slovima su označeni skupovi koji zadovoljavaju uvjet da je minimalna podrška barem 50%

Tablica 1 prikazuje postupak otkrivanja stavki visoke podrške na ranije definiranom skupu primjera. Vidljivo je da samo stavke koje svaka zasebno imaju visoku podršku mogu sudjelovati u skupovima koji imaju visoku podršku. Uz pretpostavku da smo unaprijed odredili da nas zanimaju samo skupovi sa najmanje 50% podrške, traženje se odmah može ograničiti samo na stavke 'a', 'b', 'd', i 'e'. Potencijalno zanimljivi su samo skupovi 'ad', 'de', 'ae' te 'ade'. Posebno je zanimljiv ovaj posljednji skup jer uključuje čak 3 stavke.

7.3.2. Otkrivanje pravila koja opisuju značajne uzorke

Skupovi čestih stavki su polazna točka za otkrivanje pravila koja predstavljaju opis potencijalno značajnih uzoraka u skupu primjera. Pravila imaju oblik

AKO stavka_1 I stavka_2 I ..... ONDA stavka_X

Za skup stavki 'ade' moguće je formirati pravila:

AKO a I d ONDA e (pouzdanost 3/3 = 100%)
AKO a I e ONDA d (pouzdanost 3/4= 75%)
AKO d I e ONDA a (pouzdanost 3/3 = 100%)

Interpretacija za prvo od pravila je: ako se u primjeru pojave stavke 'a' i 'd' onda se može očekivati da će se u primjeru pojaviti i stavka 'e'.

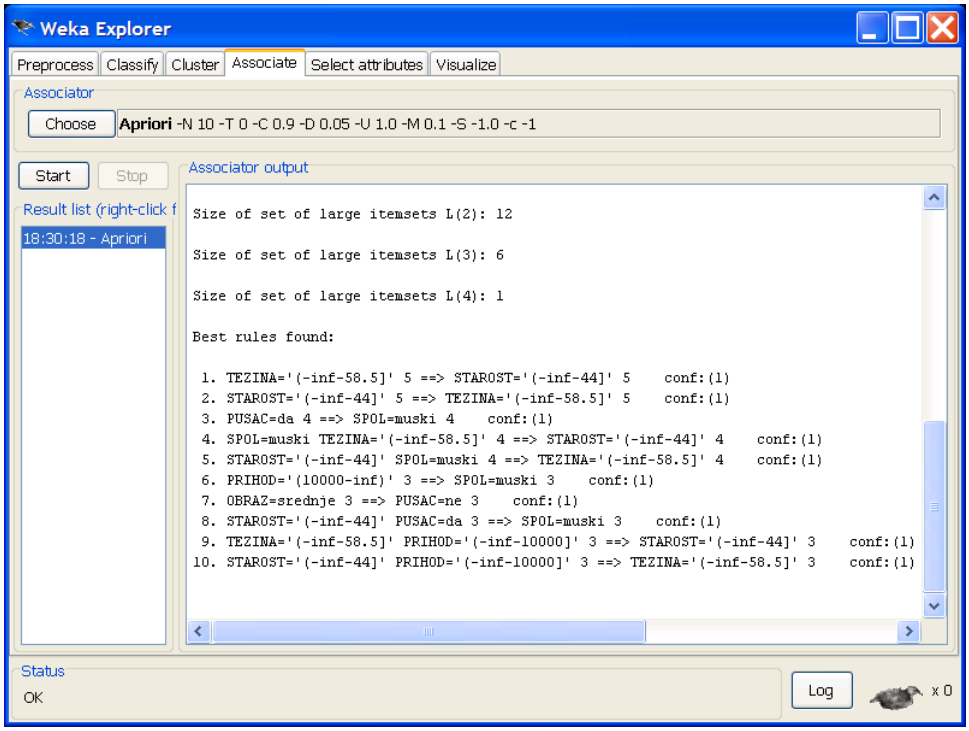
Za neki skup čestih stavki postupak formiranja pravila se sistematično ponavlja za sve moguće kombinacije stavki koje ga čine. Zato što krećemo od skupova stavki koje su česte, pravila će nužno opisivati uzorke koji su česti u skupu primjera. Da bi pravila bila i značajna, za svako od njih se računa i pouzdanost. Uz pretpostavku da smo unaprijed odredili da nas zanimaju samo uzorci sa najmanje 90% pouzdanosti, u gornjem primjeru će biti eliminirano drugo pravilo a preostala dva će biti uvrštena u skup čestih uzoraka.

Važnost postupka otkrivanja čestih uzoraka je što postoje problemi otkrivanja znanja koji prirodno podatke skupljaju u obliku primjera koji su opisani stavkama. Najpoznatije su transakcijske domene, poznate i kao problem 'potrošačke košarice' u kojem su primjeri kupci a stavke predmeti koje je pojedini kupac zajedno kupio. Uzorke predstavljaju pravila tipa: ako je kupac kupio mineralnu vodu i vino onda će vjerojatno kupiti i neke grickalice. Odnosno ako je posjetitelj gledao film A i B onda će ga zanimati i film C.

7.3.3. Pretvaranje atributa u stavke

Osnovna prednost postupka otkrivanja uzoraka je što ne zahtijeva definiciju ciljne klase. Pored toga, opisani postupak je i vrlo brz te se može uspješno koristiti i za velike baze primjera. Sve to se može iskoristiti i za primjere opisane atributima ako se njihove vrijednosti prethodno pretvore u stavke.

Za nominalne atribute se to postiže tako da svaka od vrijednosti koja se pojavljuje u primjerima postaje posebna stavka. Za numeričke atribute se vrijednosti grupiraju u dvije ili više podgrupa i to tako da dobijemo, na primjer grupe niskih, srednjih i visokih vrijednosti. Svaka od tih grupa postaje zasebna značajka koja je ili prisutna u primjeru ili nije prisutna. Postupak je sličan formiranju literala u učenju pravila ali zato što nisu poznate klase primjera, postupak opisan u poglavlju 7.1.1 se ne može direktno primijeniti. Umjesto toga, grupiranje se vrši tako da grupe ili predstavljaju jednake raspone vrijednosti ili da svaka podgrupa obuhvaća podjednak broj primjera.



Slika 15. Najznačajniji uzorci otkriveni u podacima o pušačima i nepušačima

Slika 15. prikazuje pravila koja definiraju najznačajnije uzorke za 8 primjera pušača i nepušača opisanih u poglavlju 3. Rezultat je dobiven Apriori postupkom unutar Weka sustava. Numerički atributi STAROST, TEZINA i PRIHOD pretvoreni su u po dvije značajke koje predstavljaju grupe niskih i visokih vrijednosti. Granične vrijednosti su određene na sredini između minimalne i maksimalne vrijednosti koja se pojavljuje u primjerima. Značajka „TEZINA='(-inf-58.5]'“ označava osobe niske tjelesne težine (od beskonačno male vrijednosti do granične vrijednosti) a značajka „PRIHOD='(10000-inf)'“ osobe visokih primanja (od granične vrijednosti do beskonačne vrijednosti).

Slika prikazuje ukupno 10 pravila od kojih prvo govori da ako je mala težina ispitanika da je onda on i mlad. Uz skupove značajki na slici su prikazani i brojevi koji označavaju koliko primjera sadrži određene značajke. Vidljivo je da prvo pravilo opisuje uzorak koji je vrlo čest jer je prisutan u čak 5 od 8 primjera. A on je i potpuno pouzdan (pouzdanost jednaka 1, odnosno 100 %) što znači da svi su svi ispitanici male težine (do 58.5) ujedno i mlađe osobe (do 44). Potpuno precizna su i sva ostala pravila, uključujući i posljednje koje govori da su mlađe osobe manjih prihoda ujedno i osobe manje težine. Pravila su poredana po podajućoj vrijednosti podrške.

Dobiveni rezultat pokazuje da postupak omogućuje otkrivanje mnogo potencijalno zanimljivih uzoraka ali da često pravilna interpretacija nije lagana zbog tipično vrlo velikog broja otkrivenih uzoraka.

Isto tako, treba uočiti da među najznačajnijim uzorcima nema pravila kojeg smo inducirali učenjem pravila i stabla odlučivanja kada smo imali definiranu ciljnu klasu. To je posljedica neoptimalne podjele prihoda u podgrupe u kojoj je kao granična vrijednost uzeta vrijednost 10000. Ona je odabrana jer je na pola udaljenosti između minimalne vrijednosti 0 i maksimalne vrijednosti od 20000 zabilježene u primjerima. Nasuprot tome, poznavanje ciljne klase nam je omogućilo izdvajanje vrijednosti 14000 odnosno 15000 kao optimalne vrijednosti koja razdvaja primjere različitih klasa i koje za klasifikacijske probleme predstavlja bolje rješenje.

8. Alati za dubinsku analizu podataka

Za znanstvene primjene od posebno su važnosti alati koji u sebi integriraju više raznih postupaka strojnog učenja i pripreme podataka te na taj način omogućavaju izvođenje različitih eksperimenata otkrivanja znanja. Neki znanstveni časopisi inzistiraju na ponovljivosti rezultata a za to je nužno, pored javno dostupnih podataka, i da su korišteni postupci javno dostupni. Weka i RapidMiner alati zadovoljavaju oba kriterija. Dodatno, oba alata su besplatna.

8.1. Korištenje Weka sustava

Korištenje Weka sustava započinje instalacijom na vlastito računalo programa dostupnog na http://www.cs.waikato.ac.nz/ml/weka/. Sustav se neprestano nadograđuje [3] a posljednja stabilna verzija 3.6.4 zahtijeva da se sa spomenute stranice prenese na svoje računalo datoteka „weka-3-6-4jre.exe“ te da se pokrene njeno izvođenje.

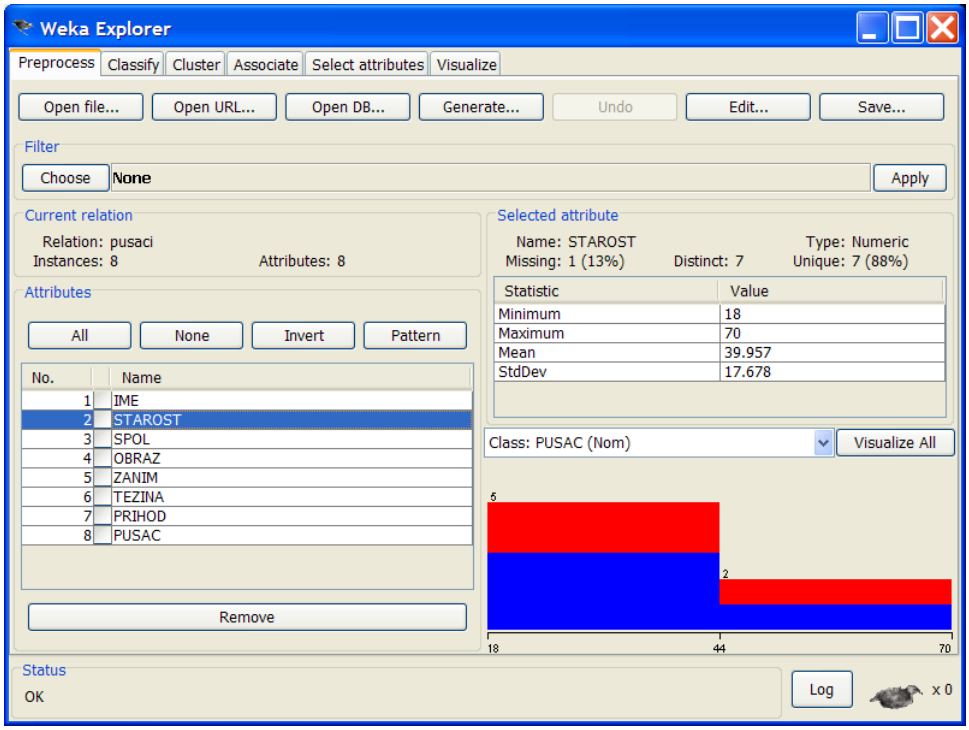
Slika 16. Početni prozor Weka sustava

U početnom prozoru sustava (Slika 16) tipkom „Explorer“ se prelazi u dio namijenjen analizi podataka. Postupak započinje otvaranjem datoteke sa podacima.

Weka sustav koristi arff format podataka. Osnovna karakteristika ovog formata podataka je da postoji definicija imena i tipa atributa. Sve definicije atributa navode se ispred samih podataka koji započinju nakon oznake „@data“. Podaci o pušačima i nepušača ilustriraju kako izgleda ovaj oblik podataka.

```
%podaci o pusacima i nepusacima u ARFF formatu
@relation pusaci
@attribute IME string
@attribute STAROST numeric
@attribute SPOL { muski, zenski }
@attribute OBRAZ { nize, srednje, visoko}
@attribute ZANIM string
@attribute TEZINA numeric
@attribute PRIHOD numeric
@attribute PUSAC {da, ne}
@data
jan,     30,     muski,    nize,      radnik,    27.3,    14000,    da
janko,   55.5,   muski,    srednje,   radnik,    90,      20000,    ne
zora,    ?,      zenski,   visoko,    ucitelj,   65.2,    1000,     ne
tanja,   18,     zenski,   srednje,   student,   55.1,    0,        ne
tom,     70,     muski,    visoko,    ?,         60,      9000,     da
tomi,    35,     muski,    srednje,   prof,      33,      16000,    ne
stev,    42.2,   muski,    nize,      vozac,     27,      7500,     da
marc     29,     muski,    ?,         konobar,   31,      8300,     da
```

Jednostavniji način pripreme podataka za Weka sustav je preko Excel tablice. U tom slučaju ne treba posebno definirati atribute već samo navesti njihova imena u prvom redu tablice. Kada je završena priprema podataka, tada tablicu treba zapisati u csv formatu i takvu datoteku otvoriti sa Weka sustavom.



Slika 17. Weka Explorer nakon prijenosa podataka o pušačima i nepušačima

Slika 17. pokazuje centralno mjesto za analizu podataka nakon uspješnog prijenosa podataka. Karakteristika Weka sustava je da pruža koristan i jednostavan uvid u podatke. Slika prikazuje kako su odabirom imena atributa STAROST (lijevi dio slike) dobivene u desnom dijelu njegove numeričke karakteristike (srednja vrijednost i standardna devijacija) te grafička ilustracija raspodjele klasa primjera ovisno o vrijednostima ovog atributa.

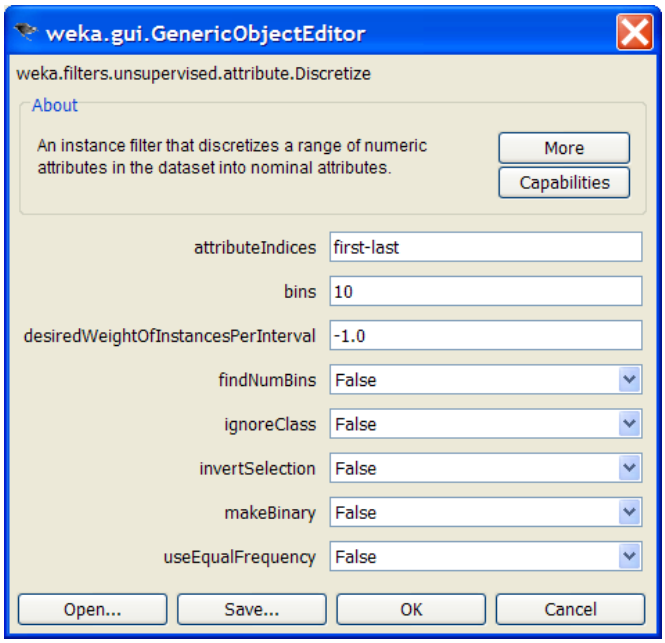
Odabirom jednog ili više atributa i pritiskom na tipku „Remove“ moguće je jednostavno isključiti neke atribute iz analize. Pritiskom na tipku „Edit...“ moguće je prekontrolirati učitane podatke te ih po potrebi i promijeniti za potrebe analize.

Nakon učitavanja podataka otvara se mogućnost da se počnu koristiti osnovne funkcije sustava. Izbor se vrši u gornjem dijelu Explorer prozora pritiskom na tipke „Preprocess“ (transformacija podataka), „Classify“ (postupci izgradnje modela za klasificirane primjere), „Associate“ (otkrivanje čestih uzoraka) i „Select attributes“ (odabir podgrupe najznačajnijih atributa).

8.1.1. Transformacija podataka

Nakon odabira tipkom „Preprocess“, pritiskom na tipku „Choose“ dobiva se izbornik sa velikim brojem vrlo raznih funkcija koje omogućuju transformaciju podataka. Podijeljene su u grupe ovisno o tome da li su primjeri klasificirani ili nisu te da li se želi transformirati atribute ili primjere.

Jedna od važnijih funkcija je ona koja omogućuje transformaciju numeričkog atributa u skup nominalnih vrijednosti. Funkcija se naziva „Discretize“ i postoji i za skupove klasificiranih primjera i za one kada klasifikacija nije poznata.



Slika 18. Prozor za izbor parametara funkcije "Discretize"

Nakon izbora funkcije potrebno je podesiti njene parametre. Slika 18. prikazuje prozor koji dobijemo nakon pritiska na ime već odabrane funkcije „Discretize“. Najvažniji među njima je „bins“ koji određuje na koliko podgrupa će se podijeliti vrijednosti. Osnovna vrijednost je 10 a za male skupove podataka je preporučljivo odabrati 2. Važan je i „useEqualFrequency“ koji ima osnovnu vrijednost False što znači da će generirane podgrupe pokrivati isti raspon veličina. Promjenom na vrijednost True moguće je odrediti da treba podjednak broj primjera rasporediti po svakoj podgrupi.

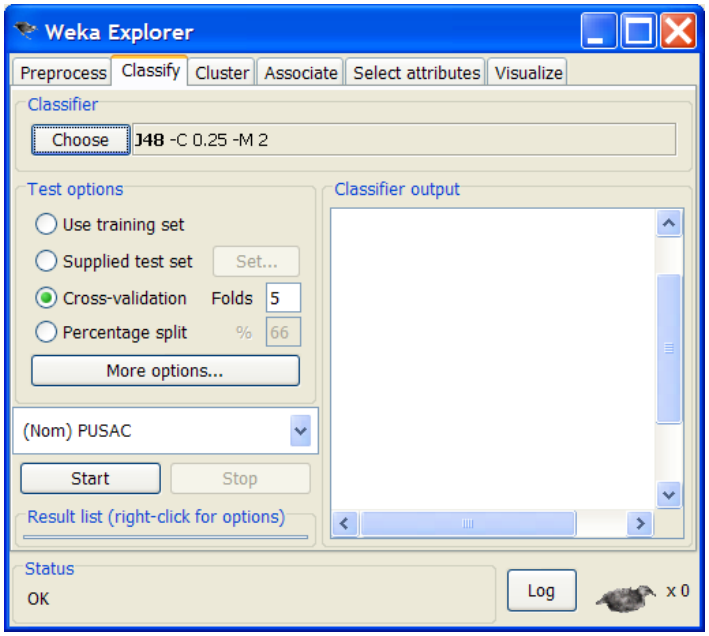
U prozoru izbora vrijednosti parametara pritiskom na tipku „More“ dobiva se više informacija o samoj funkciji te opis značenja pojedinih parametara.

Nakon što su podešeni parametri i prozor za njihov odabir zatvoren, odabrana funkcija se izvodi pritiskom na tipku „Apply“.

8.1.2. Postupci izgradnje modela za klasificirane primjere

Ako su naši podaci klasificirani, postupak izgradnje modela započinje tipkom „Classify“ nakon čega je pritiskom na tipku „Choose“ moguće izabrati neki od ponuđenih postupaka. Izbor je prilično velik a najznačajniji postupci su induciranje stabala odlučivanja poznat kao J48 (čiji rezultati su već pokazani na slikama 11 i 15) i JRip postupak za učenje pravila originalno poznat kao RIPPER [6].

Za izvorne podatke o pušačima i nepušačima neće biti dozvoljeno pokretanje velikog broja postupaka. Razlog je u činjenici što podaci uključuju atribute IME i ZANIM koji su tipa string koji dozvoljava slobodni tekst. Jednostavno rješenje je uklanjanje tih atributa. Nakon toga će većina ugrađenih postupaka biti dostupna za ovaj skup primjera.



Slika 19. Odabir postupka izgradnje stabla odlučivanja J48 uz unakrsnu validaciju sa 5 podgrupa primjera

Slika 19. pokazuje da smo odabrali učenje stabla odlučivanja koji je u Weka sustavu poznat kao J48. Za sve klasifikacijske postupke postoji mogućnost izbora načina provjere prediktivne točnosti induciranog modela. Ako ne postoji nezavisni testni skup a skup za učenje ima više od pedesetak primjera, preporučljivo je iskoristiti mogućnost unakrsne validacije. Za male skupove primjera, kao što je skup primjera o pušačima i nepušačima, treba odabrati isti skup za testiranje (bez unakrsne validacije) te započeti izgradnju modela pritiskom na tipku „Start“. Nakon uspješnog završetka, pojavljuje se natpis „treesJ48“ sa vremenom završetka modeliranja. Ako se želi dobiti grafički prikaz generiranog stabla, kao što je prikazan na Slici 11, treba pritisnuti desnu tipku miša na spomenuti natpis i odabrati funkciju „Visualize tree“.

8.1.3. Otkrivanje čestih uzoraka

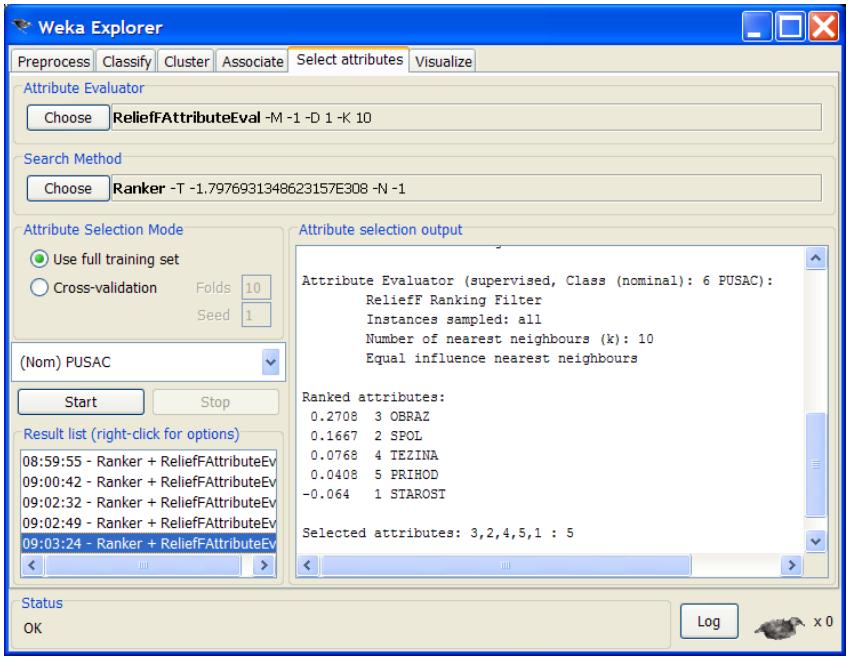
Nakon odabira tipkom „Associate“, pritiskom na tipku „Choose“ dobiva se izbornik sa malim brojem funkcija od kojih je najvažnija Apriori.

Ova funkcija se može koristiti samo ako ne postoje string atributi te ako su numerički prethodno pretvoreni u nominalne. Postupak sa korištenjem funkcije „Discretize“ je objašnjen u poglavlju 8.1.1.

Slika 15. prikazuje oblik rezultata koji se može dobiti korištenjem Apriori funkcije.

8.1.4. Odabir podgrupe najznačajnijih atributa

Nakon odabira tipkom „Select Attributes“ moguće je izabrati veći broj raznih postupaka za izbor značajnih atributa. Postupak se sastoji od kombinacije dvije funkcije pa se prvom tipkom „Choose“ odabire mjera sa kojom će se evaluirati atributi a sa drugom tipkom „Choose“ se odabire kako će se pretraživati skup atributa. Za odabrani postupak evaluacije nije moguće koristiti sve postupke pretraživanja ali Weka sustav će nas obavijestiti ako pokušamo napraviti nedozvoljenu kombinaciju i odmah predložiti neki od primjerenih načina pretraživanja.



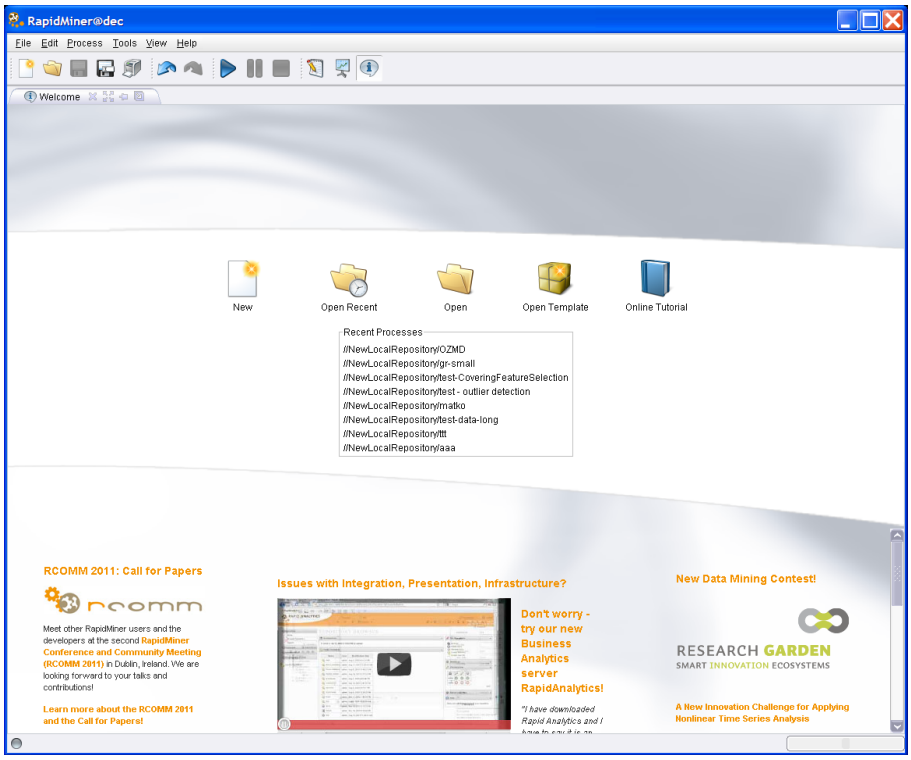
Slika 20. Rangiranje atributa uz pomoć funkcije ReliefF

Postupci odabira atributa važni su u strojnom učenju jer se eliminacijom nebitnih atributa iz skupa za učenje može poboljšati prediktivna kvaliteta induciranih modela. Za otkrivanje znanja postupci odabira atributa su korisni jer pružaju informaciju o važnosti pojedinih atributa.

Postoji velik broj načina evaluacije atributa koji ih rangiraju po različitim kriterijima. Niti jedan od načina nije apsolutno najbolji. To je posebno važno imati na umu za primjene u otkrivanju znanja jer različiti postupci mogu različito rangirati atribute a interpretacija ovisi o postupku koji je primijenjen. Jedan od poznatijih načina rangiranja koji uzima u obzir i međuzavisnosti među atributima je Relief algoritam. Slika 20. prikazuje način odabira funkcije ReliefF unutar Weka sustava te dobivenu ranglistu atributa za primjere o pušačima i nepušačima. Treba uočiti da je ovaj postupak odabrao atribut OBRAZ kao najznačajniji za objašnjavanje razlika između klasa pušača i nepušača a da je atribut PRIHOD, koji je i indukcija pravila i indukcija stabla odlučivanja koristila u izgradnji modela, tek na petom

mjestu. Važnost i značenje ovakvom rezultatu rangiranja može dati samo ljudska interpretacija.

8.2. Korištenje RapidMiner sustava



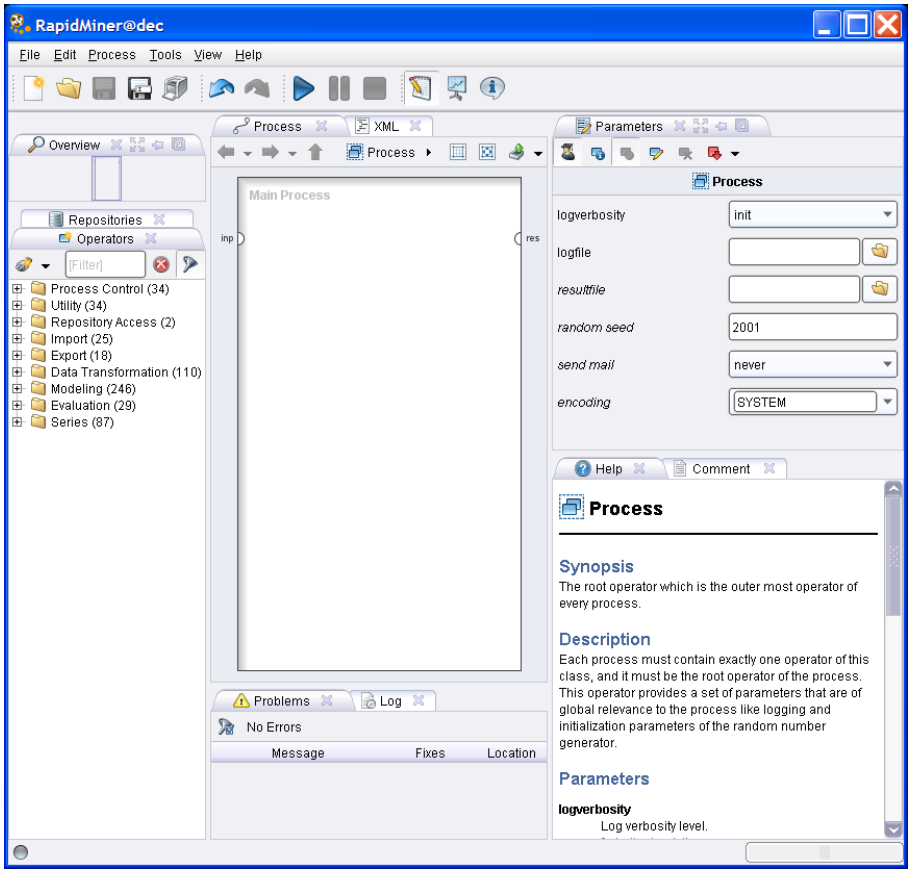
Slika 21. Početni prozor RapidMiner sustava

RapidMiner sustav je suvremeni sustav za dubinsku analizu podataka koji se odlikuje kvalitetnim korisničkim sučeljem. Nove verzije uključuju i funkcije preuzete iz Weka sutava.

Korištenje RapidMiner sustava započinje preuzimanjem instalacijskog programa (trenutna verzija rapidminer-5.1.001x32-install.exe) dostupnog na http://rapid-i.com/content/view/181/196/. Na ovoj adresi je i korisnička dokumentacija, upute za instalaciju te upute za korištenje snimljene kao kratki filmovi.

Slika 21. prikazuje početak rada sa RapidMiner sustavom. Nakon pritiska na tipku „New“ otvara se „Repository Browser“ koji omogućuje određivanje mjesta i imena datoteke za zapisivanje generiranih procesa. Za početak je dozvoljeno ovaj prozor zatvoriti bez unošenja nekog imena te krenuti sa praznim radnim prostorom prikazanim na Slici 22.

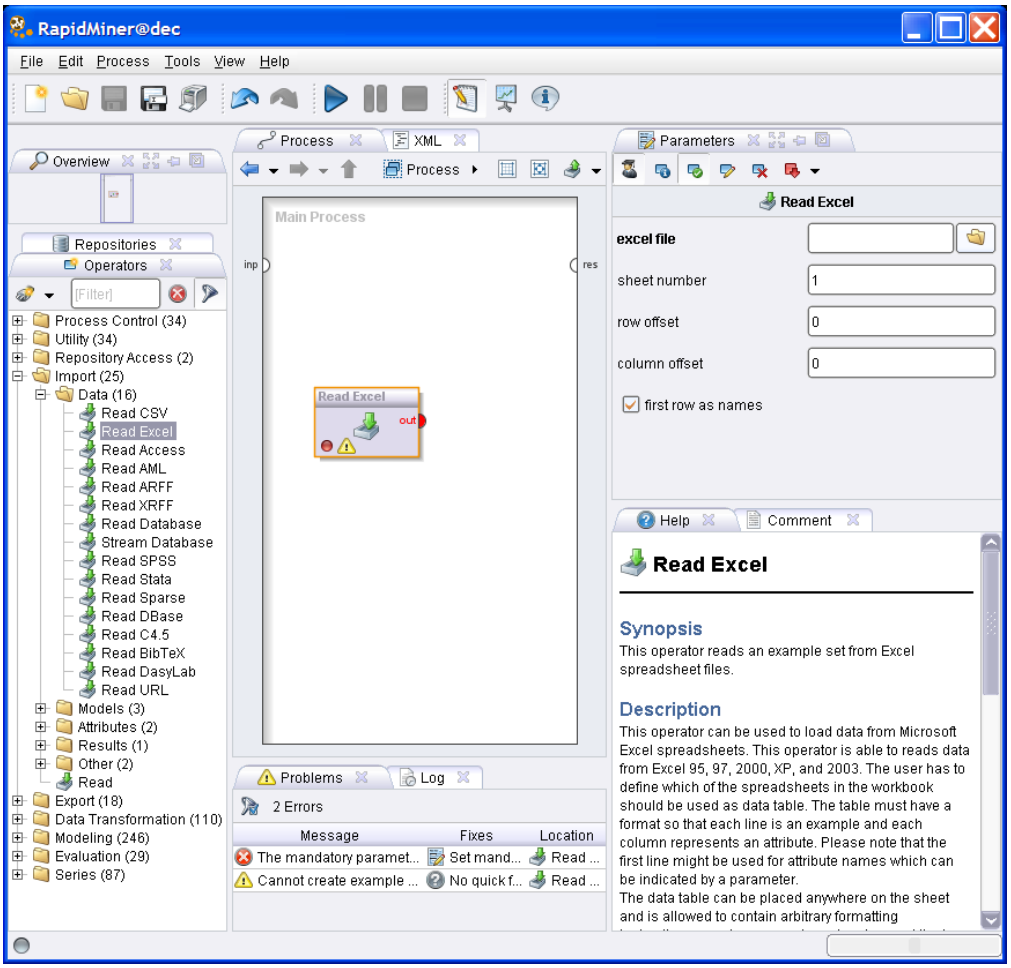
U lijevom dijelu radnog prostora je izbornik raspoloživih funkcija, u središnjem dijelu je prostor za prikazivanje procesa, u gornjem desnom dijelu je prostor za definiranje parametara pojedinih funkcija a u donjem desnom je prostor za upute koje objašnjavaju način korištenja pojedinih funkcija uključenih u proces.



Slika 22. Osnovni radni prostor RapidMiner sustava pripremljen za početak definiranja procesa analize podataka

RapidMiner prihvaća više raznih formata podataka, uključujući i arff format preuzet od Weka sustava. Ali tu je i funkcija „Import/Data/Read Excel“ koja može učitavati direktno i Excel dokumente.

Odabirom funkcije „Import/Data/Read Excel“ ona se potezom miša može prenijeti u prostor za prikazivanje procesa (Slika 23). Žuti trokutić i crven točka koji se pojavljuju na toj funkciji signaliziraju da nešto nije u redu sa tom funkcijom. U konkretnom slučaju problem je što ta funkcija nužno treba imati definirano ime datoteke koja sadrži podatke. To se čini u desnom gornjem dijelu radnog prostora.



Slika 23. Funkcija za čitanje Excel datoteke

Read Excel.output (output)
Meta data: Data Table
Number of examples = 8
8 attributes:
Generated by: Read Excel.output
Data: SimpleExampleSet: 8 examples, 8 regular attributes, no special attributes

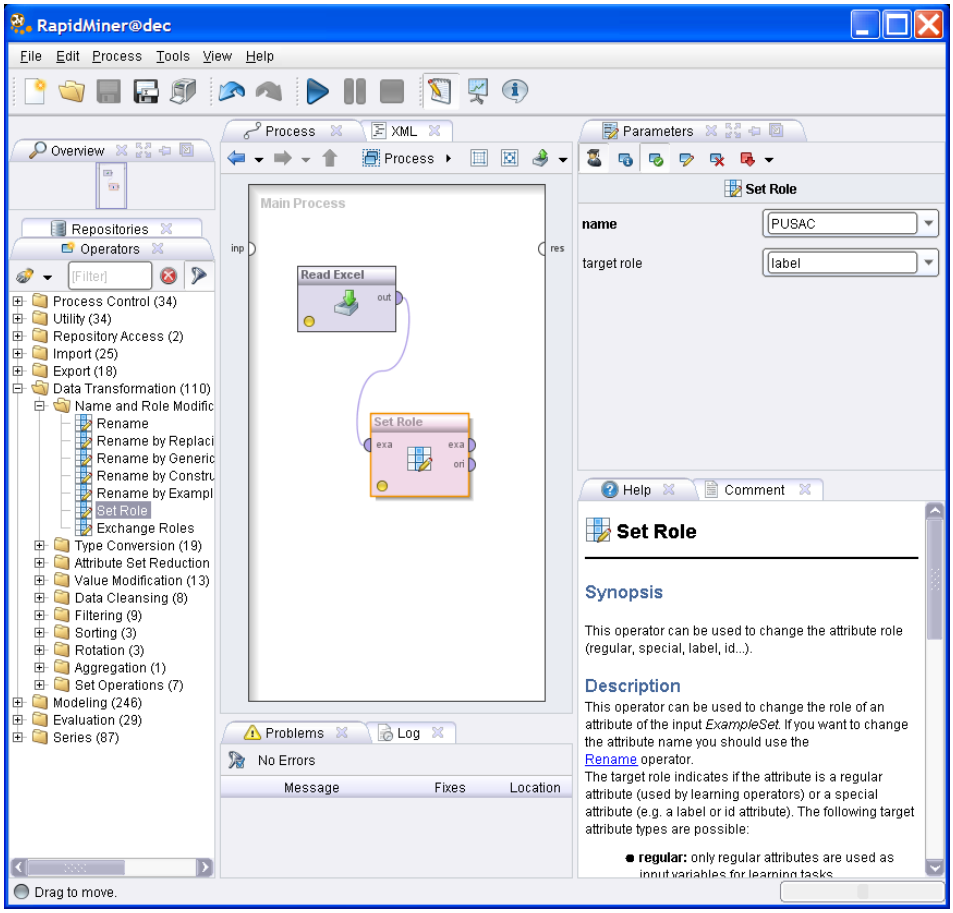
Role	Name	Type	Range	Missings
	IME	nominal	=[jan, janko,...	= 0
	STAROST	nominal	=[30.0, 55.5,...	= 0
	SPOL	nominal	=[muski, ze...	= 0
	OBRAZ	nominal	=[nize, sred...	= 0
	ZANIM	nominal	=[radnik, uci...	= 0
	TEZINA	integer	=[27 - 90]	= 0
	PRIHOD	integer	=[0 - 20000]	= 0
	PUSAC	nominal	=[da, ne]	= 0

Press "F3" for focus.

Slika 24. Prikaz informacija o datoteci sa podacima

Odmah nakon uspješnog definiranja imena datoteke, gasi se žuti trokutić a onaj crveni dobiva zelenu boju. Polukrug na desnoj strani funkcije je mjesto gdje je izlaz podataka iz te funkcije. Dolaskom miša na to mjesto pojavljuje se prozor koji prikazuje datoteku sa podacima u obliku liste atributa, njihovih tipova i raspona vrijednosti (Slika 24).

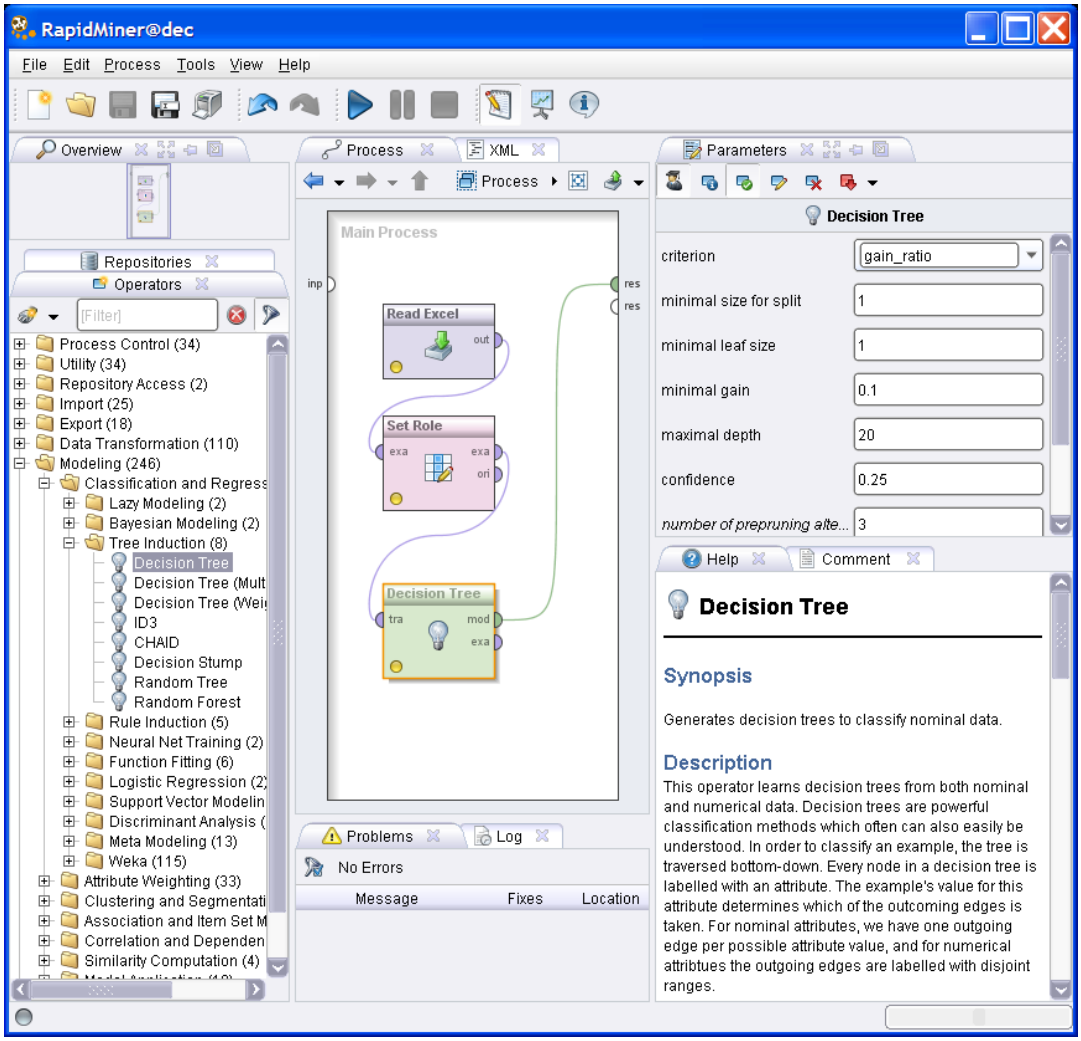
Slika 25. prikazuje kako se uvođenjem funkcije „Data Transformation/Name and Role Modification/Set Role“ definira atribut klase primjera. Nakon što je mišem prenesena funkcija u radni prostor, u gornjem desnom dijelu se odabere ime atributa (PUSAC) a zatim se on definira kao „label“, što znači da on određuje klase primjera.



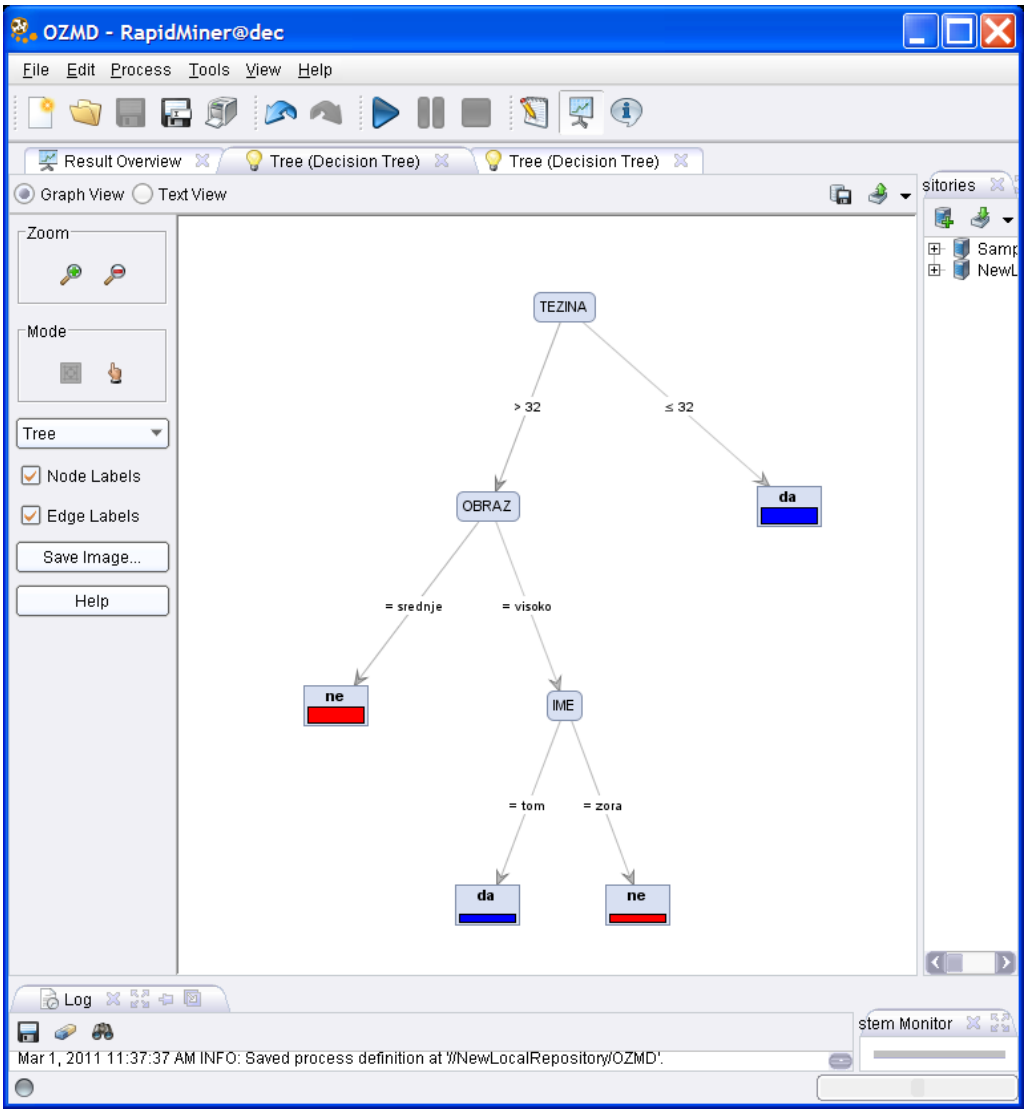
Slika 25. Odabir atributa PUSAC kao ciljnog atributa koji određuje klase primjera

Svakako treba uočiti da je potrebno crtom spojiti izlaz funkcije za čitanje podataka sa funkcijom za određivanje ciljnog atributa. To se postiže time da se lijevom tipkom miša pritisne na polukrug izlaza podataka prve funkcije i zatim se istom tipkom pritisne na polukrug ulaza podataka druge funkcije. Standardno se svi ulazi podataka nalaze sa lijeve strane funkcija. Odmah po uspješnom spajanju koje je označeno linijom koja povezuje izvor i odredište podataka, informacije o skupu podataka u vidu liste atributa se pojavljuju i na izlazu druge funkcije. Jedina razlika je da se sada uz atribut PUSAC pojavljuje natpis „label“.

Izlaz funkcije za određivanje ciljnog atributa („exa“) spreman je za indukciju modela. Slika 26. prikazuje proces u koji je sada uključena i funkcija „Decision Tree“ . Njen izlaz „mod“ je spojen kao izlaz cijelog procesa „res“. Pritiskom na plavi trokut u gornjem dijelu radnog prostora moguće je sada startati proces. Rezultat u obliku stabla odlučivanja (Slika 27) je rezultat indukcije.



Slika 26. Indukcija stabla odlučivanja



Slika 27. Stablo odlučivanja inducirano RapidMiner funkcijom „Decision Tree“

9. Što dalje ?

Kao prvo, preporuča se u potpunosti upoznati alate za dubinsku analizu podataka. Za Weka sustav na raspolaganju su dvije PowerPoint prezentacije koje opisuju korištenje tog sustava. Dostupne su na http://www.cs.waikato.ac.nz/ml/weka/index_documentation.html. Za RapidMiner sustav postoji cijeli niz dobrih video snimki koje se mogu naći na http://rapid-i.com/content/view/189/198/.

Nakon toga preporuča se detaljnije upoznati postupke strojnog učenja. Učenje pravila i stabla odlučivanja prikazani u ovom priručniku samo su osnovni tipovi. Postupci najbližih susjeda (engl. *k-NN, k-nearest Neighbor*), naivni Bayes (engl. *Naïve Bayes*), Bayesove mreže (engl. *Bayesian Networks*), neuronske mreže (engl. *Neural Networks*) i postupci otkrivanja skupina primjera (engl. *clustering methods*) standardni su postupci koji su svoju primjenjivost već dokazali u mnogim domenama. Pored toga bilo bi dobro razumjeti i moći koristiti postupke koji danas omogućuju postizanje vrhunske prediktivne točnosti: postupke potpornih vektora i slučajne šume. Ovaj posljednji posebno je važan za otkrivanje znanja jer omogućuje otkrivanje važnosti atributa, otkrivanje izuzetaka u podacima te otkrivanje relativne udaljenosti među primjerima.

Internet je dobar izvor informacija. Za svaki od spomenutih postupaka postoji vrlo kvalitetna Wiki stranica koja osim opisa postupaka nudi i najvažnije poveznice na relevantnu literaturu koja je također najčešće dostupna u elektronskom obliku.

Ako se odlučite za kupovinu knjige ponajprije se preporuča [3] jer pored opisa postupaka sadrži i način njihova korištenja unutar Weka sustava, te [4] jer je to već cijeli niz godina standardni udžbenik na mnogim sveučilištima u svijetu. Ponekad su na internetu dostupni besplatno i cijeli udžbenici. Primjer je „Introduction to Machine Learning“ koji se može naći na http://ai.stanford.edu/~nilsson/mlbook.html.

Rezultate najnovijih istraživanja objavljuju časopisi „Data Mining and Knowledge Discovery“, „Machine Learning“ i „Journal of Machine Learning Research“. Ovaj posljednji omogućuje slobodno elektronsko čitanje svih objavljenih radova. Za medicinske primjene dubinske analize podataka važni časopisi su „Artificial Intelligence in Medicine“ i „Applied Intelligence“.

Relevantni radovi objavljuju se na konferencijama: „International Conference on Machine Learning“ (ICML) i „Principles and Practice of Knowledge Discovery in Databases“ (PKDD). Za medicinske primjene zanimljive konferencije su: „Conference on Artificial Intelligence in Medicine“ (AIME) i „International Conference on Health Informatics“ (HEALTHINF).

Svakako za preporučiti je i KD Nuggets (http://www.kdnuggets.com/), specijalizirani internet portal koji sadrži bogatstvo informacija o postupcima dubinske analize podataka, njihovim primjenama, tečajevima i dostupnoj literaturi. Portal objavljuje i besplatne elektronske novine koje redovito izlaze dva puta mjesečno.

9.1. Znanstveni izazovi

Za razvoj strojnog učenja trajni izazovi su povećanje prediktivne točnosti induciranih modela te učenje iz složenih oblika podataka, u prvom redu relacijskih baza podataka, zvučnih zapisa i slika. Ovo posljednje zanimljivo je i za područje otkrivanja znanja, posebno otkrivanje važnih značajki iz dugih vremenskih signala kao što je EKG te slika koje se dobivaju rendgenskim i ultrazvučnih pregledima.

Povezanost sa statističkim postupcima i njihovo aktivno uključivanje u postupke strojnog učenja je specifičan izazov za područje otkrivanja znanja. Cilj te integracije bi trebalo biti da se ne samo učinkovito pretražuje prostor hipoteza već da se osigura i statistička značajnost dobivenih rezultata koja bi bila znanstveno i stručno prihvaćena. Proces se suočava sa nekoliko teoretskih problema od kojih je najznačajnija pretpostavka o identičnosti statističkih svojstava skupa za učenje i skupa na kojem će se primjenjivati inducirani modeli. Ova pretpostavka je nužna za sve probabilističke postupke strojnog učenja a istovremeno postoji stvarna potreba za modelima koji se mogu koristiti u okruženjima koja su bitno drugačija od onih u kojima su sakupljani podaci. Primjeri takvih domena su predviđanje dobrih poslovnih odluka ili sprečavanje provala u informatičke sustave. Karakteristično za medicinske primjene otkrivanja znanja su etička i financijska ograničenja pri sakupljanju podataka. Zato često imamo podatke o bolesnicima koji su liječeni samo u jednoj ustanovi a potrebno je izgraditi modele koji će pomoći u prevenciji bolesti u općoj populaciji ili pak u liječenju u medicinskim ustanovama koje imaju sasvim drugačiju raspodjelu karakteristika bolesnika. U svim tim slučajevima ostaje otvoren problem procjene kvalitete i primjenjivosti izgrađenih modela.

Postojeće ljudsko znanje znatno može pomoći u otkrivanju novog znanja. Ali da bi se ono moglo iskoristiti to znanje treba biti formalizirano tako da ga mogu koristiti i postupci strojnog učenja. Prikaz znanja ontologijama danas je općenito prihvaćeno kao najbolje rješenje formalizacije znanja. Izazovi su kako znanje sadržano ontologijama iskoristiti u strojnom učenju i evaluaciji dobivenih modela. Pored toga, aktualni proces razvoja postupaka prikaza znanja je koncentriran na formulaciju odnosa među konceptima a zapostavljeno je znanje o činjenicama pa tako danas ne postoje ontologije koje specificiraju da je normalna ljudska temperatura do 37 stupnjeva Celziusa a da je normalan dijastolički tlak oko 80 mmHg. Činjenica je da bi se mogle naći internet stranice sa ovakvim i sličnim sadržajem, odnosno da bi se one mogle ako već ne postoje, relativno jednostavno izgraditi, ali danas smo vrlo daleko od postupaka strojnog učenja koji bi bili u stanju u svom radu iskoristiti i ovakve informacije.

10. Literatura

[1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. *From Data Mining to Knowledge Discovery in Databases.* AI Magazine, 17(3):37-54, 1996.

[2] D. Pyle. *Data preparation for data mining*. Morgan Kaufmann, 1999.

[3] I.H. Witten, E. Frank, M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, treće izdanje, 2011.

[4] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[5] D. Gamberger, N. Lavrač. *Expert-guided subgroup discovery: Methodology and Application*. Journal of Artificial Intelligence Research, 17:501-527, 2002.

[6] W.W. Cohen. *Fast effective rule induction*. Proceedings of International Conference on Machine Learning (ICML-95), 115-123, 1995.

[7] D. Gamberger, N. Lavrač, S. Džeroski. *Noise detection and elimination in data preprocessing: experiments in medical domains*. Applied Artificial Intelligence, 14:205-223, 2000.

[8] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone. *Classification and Regression Trees*. Wadsworth 1984.

[9] J.R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, 1993.

[10] Blumer, A. Ehrenfeucht, D. Haussler, M.K. Warmuth. *Occam's razor*. Information Processing Letters. 24:377-380, 1987.

[11] R. Agrawal R. Srikant. *Fast algorithms for mining association rules in large databases*. Proceedings of the 20th International Conference on Very Large Data Bases, 487-499, 1994.