Reinženjering poslovnih procesa	Knjiga	**4**
Business Process Reengineering	Volume	

Urednik serije/ Editor of serias: **Alempije V.Veljović**

REŠAVANJE KLASIFIKACIONIH PROBLEMA MAŠINSKOG UČENJA

Solving Machine Learning Classification Problems

Jasmina Đ. Novaković

Fakultet tehničkih nauka u Čačku Univerzitet u Kragujevcu

Faculty of Technical Sciences Cacak University of Kragujevac

REŠAVANJE KLASIFIKACIONIH PROBLEMA MAŠINSKOG UČENJA

Dr Jasmina Novaković, prof. str. studija

Recenzent:
Prof. dr Živadin Micić
Prof. dr Dragan Milovanović

Izdavač:
Fakultet tehničkih nauka u Čačku

Za izdavača:
Prof. dr Jeroslav Živanić, dekan

Štampanje odobreno odlukom Naučno-nastavnog veća
Fakulteta tehničkih nauka u Čačku broj 14-1366/9 od 11. 09. 2013. god.

Tiraž: 100 primeraka

Štampa:
Štamparija SaTCIP, Vrnjačka Banja

ISBN 978-86-7776-157-8

S A D R Ž A J

C O N T E N T S

Predgovor

Živimo u informacionom društvu u kome je prikupljanje podataka jednostavno, a njihovo skladištenje nije skupo. Autori Piatetsky-Shapiro i Frawley navode da se iznos uskladištenih informacija udvostručuje svakih dvadeset meseci [Piatetsky-Shapiro, Frawley, 1991]. Nažalost, iako se povećava količina uskladištenih informacija, sposobnost da se iste razumeju i koristite nije u skladu sa njihovim povećanjem. Mašinsko učenje obezbeđuje alate kojima se velike količine podataka mogu automatski analizirati. Jedna od osnova mašinskog učenja je selekcija atributa. Selekcijom atributa i identifikovanjem najznačajnijih atributa za učenje, učeći algoritmi se usresređuju na one aspekte podataka koji su najkorisniji za analizu i buduća predviđanja. Različite metode za selekciju atributa primenjene su u velikom broju algoritama za klasifikaciju. U većini slučajeva, proces selekcije atributa je jednostavan i brzo se izvršava. On omogućava eliminaciju irelevantnih i redundantnih podataka, i u mnogim slučajevima, poboljšava performanse učećih algoritama.

Mašinsko učenje je jedna od oblasti računarstva koja se poslednjih decenija najbrže razvija, a problem klasifikacije nepoznatih instanci u napred predefinisane klase, je jedan od najčešćih problema mašinskog učenja. Razvoj ove oblasti je oduvek bio zasnovan na komplementarnom povezivanju teorije i eksperimenata. Budući razvoj ove oblasti računarstva zahteva proširivanje i učvršćivanje teorijskih znanja, pre svega matematičkih, ali i znanja o specifičnostima oblasti primene, kao i njihovu adekvatnu formalizaciju.

Od 1970-tih godina smanjenje dimenzionalnosti podataka je plodno tlo za istraživanje i razvoj, i to u statističkom prepoznavanju oblika [Wyse, 1980; Ben-Bassat, 1982], mašinskom učenju i *data mining*-u. Ono danas predstavlja aktivno polje istraživanja u računarstvu.

Smanjenje dimenzionalnosti podataka je fundamentalni problem u mnogim oblastima, naročito u predviđanju, klasifikaciji dokumenata, bioinformatici, prepoznavanju objekata ili u modeliranju složenih tehnoloških procesa. U takvim aplikacijama, skupovi podataka sa hiljadama atributa nisu neuobičajeni. Za neke probleme svi atributi mogu biti važni, ali za neke druge probleme samo mali podskup atributa je obično relevantan.

Da bi se prevazišli problemi koje sa sobom nosi visoka dimenzionalnost podataka, dimenzionalnost podataka bi trebalo da bude smanjena. Ovo se može uraditi tako što se izabere samo podskup relevantnih atributa, ili kreiranjem novih atributa koji sadrže maksimum informacija o datoj klasi. Prva metodologija se zove selekcija atributa, dok se druga zove ekstrakcija atributa, i obuhvata linearne (PCA, Independent Component Analysis (ICA) i sl.) i nelinearne metode ekstrakcije atributa. Pronalaženje novih podksupova atributa je obično slabo rešiv problem, kao i mnogi problemi u vezi sa ekstrakcijom atributa koji su se pokazali kao *NP-hard* [Blum i Rivest, 1992].

Neki algoritami klasifikacije su nasledili sposobnost da se fokusiraju na relevantne karakteristike i ignorišu one irelevantne. Stabla odlučivanja su primer takve klase algoritama [Breiman *et al*., 1984; Quinlan, 1993], ali i višeslojni perceptron (eng. *Multilayer Perceptron* - MLP) sa jakim regulisanjem ulaznog sloja koji može isključiti nebitne atribute na automatski način [Duch *et al*., 2001]. Takođe, i takve metode mogu imati koristi od nezavisne selekcije ili ekstrakcije atributa.

S druge strane, neki algoritmi nemaju mogućnost odabira ili ekstrakcije atributa. Algoritam *k*-najbližeg suseda (eng. K-*nearest neighbour* - k-NN) je jedna porodica takvih metoda koje se u procesu treniranja podataka, snažno oslanja na metode odabira ili ekstrakcije relevantnih i nereduntantnih atributa.

Istraživanja u prvom delu su usmerena na izazove nadgledanog i nenadgledanog učenja i sagledavanja elemenata dizajna sistema koji uči. Posebna pažnja posvećena je ciljnoj funkciji, izboru prostora hipoteza, izboru algoritma i meri kvaliteta učenja.

U drugom delu, predmet istraživanja su metode redukcije dimenzionalnosti podataka. Razmatra se korelacija međusobno nezavisnih i zavisnih atributa s konceptom, i izvršena je klasifikacija atributa u četiri disjunktne klase: irelevantni atributi, slabo relevantni redundantni atributi, slabo relevantni neredundantni atributi i jako relevantni atributi. Prisustvo irelevantnih i redundantnih atributa negativno utiče na performanse induktivnog učenja, zbog čega optimalan skup atributa za učenje čine slabo relevantni neredundantni atributi i jako relevantni atributi. Usled potrebe analize metoda selekcije atributa, predmet posebnog istraživanja su metode filtriranja, metode prethodnog učenja, ugrađene metode i metode ekstrakcije atributa. Poseban predmet razmatranja su i karakteristike algoritama za selekciju atributa, kao što su: *Information Gain* (IG), *Gain Ratio* (GR), *Symmetrical Uncertainty* (SU), *Relief-F* (RF), *One-R* (OR) i *Chi-Squared* (CS).

U trećem delu razmatraju se mere za evaluaciju klasifikcijskih modela kao i metode za ocenu stvarne frekvencije grešaka klasifikacijskog modela, koje se razlikuju po pristupu problemu i svojstvima koje pokazuju. Takođe, prilikom treninga postoji mogućnost da se model previše prilagodi specifičnostima podataka za trening i da zbog toga daje loše rezultate kada se primeni na drugim podacima, zbog čega se ovaj problem posebno razmatra.

U sledećem četvrtom delu, razmatra se problem klasifikacije, koji predstavlja razvrstavanja nepoznate instance u jednu od unapred ponuđenih kategorija. U ovom delu analiziraju se klasifikacioni algoritmi, koji su korišćeni u eksperimentalnim istraživanjima za dokaz postavljenih hipoteza. To su sledeći algoritmi nadziranog učenja za izgradnju modela: IBk, *Naïve Bayes*, SVM, J48 stablo odlučivanja i RBF mreža.

U petom delu dat je prikaz izabranih problema učenja, koje u eksperimentalnom istraživanju koristimo za dokaz postavljenih hipoteza.

Šesti deo daje prikaz korišćene metodologije izvođenja eksperimenta i podešavanja parametara modela. Razmatra se tačnost i preciznost kojima merimo uspešnost dobijenog modela, kao i statistički testovi koje koristimo u istraživanjima, sa posebnim osvrtom na standardnu devijaciju i *t*-test.

U sledećem sedmom delu, nakon razmatranja postavki eksperimentalnog istaživanja, prikazani su rezultati istraživanja za različite metode filtriranja, i to za svaki klasifikacioni algoritam posebno.

Osmi deo daje razmatranje postavki eksperimentalnog istaživanja, prikaz rezultata istraživanja za različite metode prethodnog učenja za svaki klasifikacioni algoritam posebno.

U devetom delu, razmatrane su postavke eksperimentalnog istaživanja i prikazani su rezultati istraživanja za ekstrakciju atributa uz pomoć PCA metode za svaki klasifikacioni algoritam posebno.

U poslednjem delu, dat je rezime rada, potom zaključci razmatranja o uticaju prethodne selekcije atributa na klasifikacijske performanse algoritama nadziranog učenja. Na kraju, prikazani su pravci mogućih daljih istraživanja u ovoj oblasti.

Preface

We live in information society where collection of data is easy, and their storage is not expensive. Authors Piatetsky-Shapiro and Frawley stated that the amount of stored information doubles every twenty months [Piatetsky-Shapiro, Frawley, 1991]. Unfortunately, although it increases the amount of information stored, the same ability to understand and use not in accordance with their increase. Machine learning provides the tools that large amounts of data can be automatically analyzed. One of the foundations of machine learning is the attribute selection. The selection of attributes and to identify the most important attributes for learning, learning algorithms focus on those aspects of data that are useful for analysis and future predictions. Different methods for the selection of attributes have been applied in a number of algorithms for classification. In most cases, the process of attribute selection is simple and quick to execute. It allows the elimination of irrelevant and redundant data, and in many cases, improves performance learning algorithms.

Machine learning is a field of computer science that is rapidly developing in recent decades, and the problem of classification of unknown instances in the above predefined classes, is one of the most common problems of machine learning. The development of this area has always been based on complementary relation of theory and experiments. Future development of this field of computer science requires expansion and consolidation of theoretical knowledge, especially mathematics, and knowledge of specific areas of application, as well as their proper formalization.

Since the 1970s, reducing the dimensionality of data is a fertile ground for research and development, in statistical pattern recognition [Wyse, 1980, Ben-Bassat, 1982], machine learning and data mining. It is today an active field of research in computer science.

Reducing the dimensionality of data is a fundamental problem in many areas, especially in forecasting, classification of documents, bioinformatics, identification of objects and in modeling the complex industrial processes. In such applications, datasets with thousands of attributes are not uncommon. For some problems all attributes may be important, but for some other problems only a small subset of attributes is usually relevant.

In order to overcome the problems brought by the high dimensionality of the data, the dimensionality of the data should be reduced. This can be done by selecting a subset of relevant attributes, or creating a new attribute containing a maximum of information about that class. The first methodology is called the selection of attributes, while the other is called the extraction of attributes, including linear (PCA, Independent Component Analysis (ICA) etc..) and nonlinear methods to extract the attributes. Finding new subsets of attributes is usually poorly solvable problem, like many problems related to the extraction of attributes that have proven to be NP-hard [Blum and Rivest, 1992].

Some classification algorithms have inherited the ability to focus on relevant features and ignore irrelevant ones. Decision trees are an example of such a class of algorithms [Breiman et al., 1984, Quinlan, 1993], and multilayer perceptron (Eng. Multilayer Perceptron - MLP) with strong regulation of the input layer, which may exclude irrelevant attributes automaticly [Duch et al. , 2001]. Also, such methods can benefit from the independent selection or attribute extraction.

On the other hand, some algorithms do not have a selection or attribute extraction incorporate. Algorithm k-nearest neighbor (k-NN) is a family of such methods in the process of training data, relies heavily on the selection method and extraction of relevant and non-redudant attributes.

Research in the first part focuses on the challenges of supervised and unsupervised learning and observation elements of design systems that learn. Special attention is given to the target function, the choice of hypotheses space, selection algorithm and measure the quality of learning.

In the the second part object of study is the data dimensionality reduction method. Considers the correlation between each independent and dependent attributes of the concept, attributes were classified into four disjoint classes: irrelevant attributes, weakly relevant attributes are redundant, poorly relevant and non-redundant attributes and very relevant attributes. The presence of irrelevant and redundant attributes negatively affect on the performance of inductive learning, since an optimal set of attributes for learning are poorly relevant and non-redundant attributes and very relevant attributes. Due to the need analysis methods of selection attributes, the subject of separate investigations are filtering methods, the wrapper methods, and embedded methods to extract attributes. A special case considering the features and algorithms for the selection of attributes, such as Information Gain (IG), Gain Ratio (GR), Symmetrical Uncertainty (SU), Relief-F (RF), One-R (OR) and Chi-Squared (CS).

In the third section the performance measures of classification models and methods for the assessment of the real frequency of errors of classification models, which differ in their approach to the problem and the characteristics that show. Also, during the training it is possible to adapt the model to specific training data and therefore gives poor results when applied to other data, which is why this particular problem under consideration.

The following fourth part, discusses the problem of classification, which is the classification of unknown instances in one of the following categories in advance. This section analyzes the classification algorithms have been used in experimental studies to prove the hypotheses. These are the following supervised learning algorithms for building models IBK, Naïve Bayes, SVM, J48 decision tree and RBF networks.

The fifth chapter provides an overview of selected issues of learning, which we use in experimental research to prove the hypotheses.

The sixth section provides an overview of the methodology and the experiment setup of the model parameters. Discusses the accuracy and precision by which we measure the success of the obtained model, and the statistical tests used in research, with special emphasis on standard deviation and *t*-test.

In the seventh section, after review the settings of experimental research we present the results of various filter methods, for each particular classification algorithm.

The eighth section provides the settings of experimental research, and overview experimental results of the different wrapper methods for each particular classification algorithm.

In the ninth section, the settings are considered, and also experimental results of research for extract attributes using PCA method for each particular classification algorithm.

In the final section, are summarized the work and the conclusions of the previous discussion on the impact of attribute selection on the classification performance of supervised learning algorithms. Finally, show the directions of the possible future research in this area.

1. VEŠTAČKA INTELIGENCIJA I MAŠINSKO UČENJE: KONCEPTI I DEFINICIJE

U prvom delu, biće reči o veštačkoj inteligenciji i mašinskom učenju kao oblasti veštačke inteligencije koja se bavi izgradnjom prilagodljivih računarskih sisitema koji su sposobni da poboljšavaju svoje performanse koristeći informacije iz iskustva.

1.1. Izazovi mašinskog učenja

Jedna od oblasti računarstva koja se poslednjih decenija najbrže razvija je **veštačka inteligencija**. Za neke oblasti računarstva se smatra da su zaokružene i u njima se ne očekuju novi značajni prodori, ali od veštačke inteligencije se rezultati tek očekuju, uprkos tome što su već razvijeni mnogi „inteligentni“ sistemi koji funkcionišu izuzetno dobro. Na ovaj način, veštačka inteligencija dobija na atraktivnosti, a nova teorijska istraživanja i eksperimenti predstavljaju put ka novim primenama u najrazličitijim oblastima. Razvoj ove oblasti je oduvek bio zasnovan na komplementarnom povezivanju teorije i eksperimenata, tako da i budući razvoj zahteva proširivanje i učvršćivanje teorijskih znanja, pre svega matematičkih, ali i znanja o specifičnim oblastima primene, kao i njihovu adekvatnu formalizaciju.

Oblast veštačke inteligencije obuhvata dva pristupa veštačkom učenju [Hutchinson, 1993]. Prvi je motivisan proučavanjem mentalnih procesa i kaže da je veštačko učenje proučavanje algoritama sadržanih u ljudskom umu. Cilj je otkriti kako se to algoritmi mogu prevesti u formalne jezike i računarske programe. Drugi pristup je motivisan sa praktičnog stanovišta računara i ima manje grandiozne ciljeve. On podrazumeva razvoj programa koji uče iz prethodnih podataka, i kao takav je grana obrade podataka. Mašinsko učenje, kao napred navedeni drugi pristup veštačkom učenju, naglo se razvijao od svog nastanka sredinom sedamdesetih godina.

Mašinsko učenje je oblast veštačke inteligencije koja se bavi izgradnjom prilagodljivih računarskih sisitema koji su sposobni da poboljšavaju svoje performanse koristeći informacije iz iskustva. Mašinsko učenje je disciplina koja se bavi proučavanjem generalizacije i konstrukcijom i analizom algoritama koji generalizuju. Prva teorijska razmatranja mašinskog učenja pojavila su se kasnih 60-ih u radovima Golda, ali univerzalne teorijske osnove su se počele učvršćivati tek tokom 80-ih godina prošlog veka. U ovoj oblasti, najvažniji teorijski pristupi su Goldov model graničnog učenja (eng. *learning in the limit*), *Valiant*-ov PAC (eng. *Probably Approximately Correct*) model i verovatno najkompletnija — statistička teorija učenja.

Mašinsko učenje je zanimljivo i zbog svoje težnje da se približi ljudskom učenju po efikasnosti, kao i da ga objasni, odnosno pruži teorijski model za njega. Neka od najvažnijih pitanja mašinskog učenja su [Janičić i Nikolić, 2010]:

- Šta se može naučiti i pod kojim uslovima?
- Kako se povećava efikasnost učenja u zavisnosti od obima iskustva?
- Koji su algoritmi pogodni za koje vrste problema?

Odgovore na napred navedena najvažnija pitanja mašinskog učenja treba tražiti kroz teorijske modele učenja u okviru kojih se u ovom pogledu već došlo do značajnih rezultata. Praktični rezultati su često prethodili teorijskim, a razlog bi lako mogao biti taj što je ova oblast duboko motivisana praktičnim primenama. U mašinskom učenju postignuti su dobri rezultati u mnogim oblastima, kao što je prepoznavanje govora, prepoznavanje rukom pisanog teksta, vožnja automobila i slično. Ali ma koliko primene mašinskog učenja bile raznovrsne, postoje zadaci koji se često ponavljaju. Zato je moguće govoriti o vrstama zadataka učenja koje se često pojavljuju. Jedan od najčešćih zadataka učenja koji se javlja u praksi je klasifikacija. Klasifikacija predstavlja prepoznavanje vrste objekata, npr. da li određeno tkivo predstavlja maligno tkivo ili ne. Regresija je zadatak mašinskog učenja u kome objektima odgovaraju vrednosti iz skupa realnih brojeva, kao što je npr. predviđanje potražnje robe u zavisnosti od raznih faktora koji na nju utiču.

Osnovnom karakteristikom inteligentnog ponašavanja može se smatrati **deduktivno zaključivanje** vođeno zakonima logike. Deduktivno zaključivanje jedan je od osnovnih načina zaključivanja kod ljudi. Druga bitna karakteristika inteligentnog ponašanja je prilagođavanje ponašanja jedinke okolini u kojoj se ona nalazi, koja se može uočiti i kod živih organizama. Putem evolutivnih procesa, prilagodljivost se postiže i kod nižih organizama, ali je ova sposobnost sa tačke gledišta veštačke inteligencije posebno zanimljiva kod životinja i ljudi kod kojih se manifestuje u toku života jedinke. U toku života jedinke, prilagođavanje se postiže učenjem na osnovu primera iz iskustva i primenom naučnog znanja u sličnim situacijama u budućnosti.

Takođe, moguće je govoriti o donošenju zaključaka o nepoznatim slučajevima, na osnovu znanja o nekim drugim poznatim slučajevima. **Generalizacija** ili induktivno zaključivanje je proces u kome se znanje koje važi za neki skup slučajeva prenosi na neki njegov nadskup. Na ovaj način, generalizacija predstavlja jedan od osnovnih koncepata mašinskog učenja. Sa konceptom generalizacije je direktno povezan koncept apstrakcije. Da bi generalizacija bila uspešna, određeni aspekti entiteta o kojima se rezonuje moraju biti zanemareni ukoliko nisu od suštinskog značaja za generalizaciju. Zato je jedna od ključnih tema u teorijskom razmatranju mašinskog učenja kontrola generalizacije i apstrakcije.

Generalizacija je jedan od osnovnih načina za formiranje predstava o okruženju, situacijama ili uzročno posledičnim odnosima, odnosno za pravljenje modela podataka iz iskustva. Ako su u nekom domenu greške u zaključivanju prihvatljive, onda algoritmi generalizacije omogućavaju zaključivanje i bez temeljnog poznavanja i kompletnog formalnog opisivanja domena na koji se primenjuju. Nekada algoritmi induktivnog zaključivanja mogu biti efikasniji i od algoritama deduktivnog zaključivanja.

Postoji nekoliko razloga zašto sisteme mašinskog učenja treba koristiti. U izučavanju mnogih pojava, ovi sistemi su korisni u slučajevima: gde algoritamska rešenja nisu na raspolaganju, gde postoji nedostatak formalnih modela, ili je ograničena stručnost u razumevanju složenih funkcija. Oni imaju potencijal za otkrivanje novih odnosa među pojmovima i hipotezama ispitujući zapise uspešno rešenih predmeta i mogu gomilati znanje koje tek treba da bude formalizovano.

1.1.1. Nadgledano i nenadgledano učenje

U mašinskom učenju postoje dve glavne formulacije problema učenja, i to:

- Nadgledano učenje predstavlja pristup problemu učenja koji se odnosi na situacije u kojima se algoritmu zajedno sa podacima iz kojih uči daju i željeni izlazi.
- Nenadgledano učenje predstavlja pristup problemu učenja koji se odnosi na situacije u kojima se algoritmu koji uči pružaju samo podaci bez izlaza, a od algoritma koji uči očekuje se da sam uoči neke zakonitosti u podacima koji su mu dati.

Kao primer nadgledanog učenja, već je pomenuta klasifikacija tkiva na maligna i ona koja to nisu. Primer nenadgledanog učenja je tzv. klasterovanje, odnosno uočavanje grupe sličnih objekata kada ne znamo unapred koliko grupa postoji i koje su njihove karakteristike. U ovom slučaju, tkiva se mogu klasterovati po njihovoj sličnosti.

1.1.2. Ciljna funkcija i hipoteze

U mašinskom učenju, ono što je potrebno naučiti se definiše ciljnom funkcijom. Ciljna funkcija definiše željeno ponašanje sistema koji uči. Ako lekar želi da prepozna maligna tkiva kod pacijenta, ciljna funkcija takvim tkivima pridružuje 1, a ostalim -1.

Pri učenju su greške moguće i čak sasvim izvesne, pa tako učenje predstavlja približno određivanje ove ciljne funkcije, odnosno može biti viđeno kao aproksimiranje funkcija. Modelima podataka ili hipotezama nazivamo funkciju kojom aproksimiramo ciljnu funkciju. U našem primeru prepoznavanja tkiva model može biti npr. funkcija $sgn(ax + by + c)$ koja je pridruživala 1 svim tačkama sa jedne strane prave, a -1 tačkama sa druge.

Prostorom hipoteza nazivamo skup svih dopustivih hipoteza. Potencijalne reprezentacije hipoteza su raznovrsne, i one mogu predstavljati linearne funkcije, pravila oblika IF...THEN i sl. U našem primeru prepoznavanja tkiva hipoteze su reprezentovane pravama definisanim preko vrednosti koeficijenata a, b i c.

1.1.3. Nalaženje hipoteze

Nalaženje hipoteze koja najbolje aproksimira ciljnu funkciju možemo videti kao pretragu prostora hipoteza koja je vođenja podacima, a koju realizuje algoritam učenja. Za kvalitet učenja je od fundamentalnog značaja izbor prostora hipoteza. Iako izgleda paradoksalno, preterano bogatstvo prostora hipoteza po pravilu dovodi do lošijih rezultata, o čemu će biti diskutovano u nastavku teksta.

1.1.4. Podaci za trening i testiranje

Instance ili primerci se u računaru predstavljaju u obliku koji je pogodan za primenu algoritama učenja. Kod algoritama mašinskog učenja, najpogodniji način za predstavljanje instanci je pomoću nekih njihovih svojstava, odnosno atributa (eng. *feature*, *attribute*). Ta svojstva ili atributi predstavljaju karakteristike instanci, tako da svaki od izabranih atributa može imati vrednost koja pripada nekom unapred zadatom skupu. Vrednosti atributa su često numeričke, ali mogu biti i kategoričke, odnosno mogu predstavljati imena nekih kategorija kojima se ne mogu jednoznačno dodeliti smislene numeričke vrednosti ili uređenje. U mašinskom učenju, kada su izabrani atributi pomoću kojih se instance opisuju, onda se svaka instanca može predstaviti vektorom vrednosti atributa koje joj odgovaraju.

U mašinskom učenju, podaci na osnovu kojih se vrši generalizacija, nazivaju se **podacima za trening**, a njihov skup trening skup. S obzirom da testiranje naučenog znanja na podacima na osnovu kojih je učeno, obično dovodi do značajno boljih rezultata od onih koji se mogu kasnije dobiti u primenama, potrebno je pre upotrebe proceniti kvalitet naučenog znanja. Ovo se obično postiže tako što se razmatra koliko je naučeno znanje u skladu sa nekim unapred datim podacima za testiranje. Test skup čine podaci za testiranje. Test skup treba da bude takav da je disjunktan sa trening skupom.

1.2. Elementi dizajn sistema koji uči

Elementi dizajna sistema koji uči su [Janičić i Nikolić, 2010]:

- formulacija problema učenja: nadgledano ili nenadgledano učenje,
- zapis primera,
- izbor ciljne funkcije,
- izbor prostora hipoteza,
- izbor algoritma,
- izbor mera kvaliteta učenja.

U već pomenutom klasifikovanju tkiva na maligna i ona koja to nisu, razmatraćemo moguće elemente dizajna sistema koji uči. Kao primer, možemo uzeti 1000 tkiva koja su razvrstana u dve unapred fiksirane kategorije (benigni i maligni), tako da je zadatak učenja u ovom slučaju formulisan kao zadatak nadgledanog učenja.

Za zapis primera, možemo uzeti obeležje tkiva koje se sastoji od 12 impedansi merenim na različitim frekvencijama.

Izbor ciljne funkcije može biti izvršen npr. tako da ciljna funkcija *f* pridružuje vrednost 1 malignim tkivima, a -1 ostalim.

Izbor prostora hipoteza može biti izvršen tako da npr. prostor hipoteza odgovara skupu svih pravih u odgovarajućem prostoru. Hipoteze su funkcije koje pridružuju vrednost 1 tačkama sa jedne strane prave, a -1 tačkama sa druge strane prave. Hipoteze se biraju izborom vrednosti koeficijenata *a*, *b* i *c*.

Izbor algoritma može biti takav da algoritam učenja predstavlja npr. gradijentni spust za minimizaciju odstupanja između vrednosti ciljne funkcije i hipoteze na datim primerima.

Za meru kvaliteta učenja može biti uzet npr. udeo tačno klasifikovanih tkiva.

2. REDUKCIJA DIMENZIONALNOSTI PODATAKA

U drugom delu, razmatra se problem redukcije dimenzinalnosti podataka. Redukcija dimenzionalnosti podataka je aktivno polje u kompjuterskim naukama. U pojedinim aplikacijama skupovi podataka sa hiljadama atributa nisu retkost. Svi atributi mogu biti značajni za neke probleme, ali za neke ciljane namere samo mali podskup atributa je obično relevantan.

2.1. Pojam redukcije dimenzionalnosti podataka

Selekcija atributa se može definisati kao proces koji bira minimalni podskup M atributa iz izvornog skupa N atributa, tako da je prostor atributa optimalno smanjen prema određenom kriteriju ocenjivanja. Kako se dimenzionalnost domena širi, broj atributa N se povećava. Pronalaženje najboljeg podskupa atributa je obično nerešiv problem [Kohavi i John, 1997] i mnogi problemi vezani odabir atributa su se pokazali da su *NP*-teški [Blum i Rivest, 1992].

Selekcija atributa je aktivno polje istraživanja u računarskoj nauci. To je plodno polje za istraživanje i razvoj od 1970 godina u statističkom raspoznavanje uzoraka [Wyse *et al.*, 1980; Ben-Bassat, 1982; Siedlecki i Sklansky, 1988], mašinskom učenju i *data mining*-u [Blum i Langley, 1997; Dash i Liu, 1997; Dy i Brodley, 2000; Kim *et al.*, 2000; Das, 2001; Mitra *et al.*, 2002].

Selekcija atributa je osnovni problem u mnogim različitim područjima, posebno u predikciji, klasifikaciji, bioinformatici, prepoznavanju objekata ili u modeliranju složenih tehnoloških procesa [Quinlan, 1993; Doak, 1992; Talavera, 1999; Liu i Motoda, 1998]. Skupovi podataka sa hiljadama atributa nisu retkost u takvim aplikacijama. Svi atributi mogu biti važni za neke probleme, ali za neka ciljana istraživanja, samo mali podskup atributa je obično relevantan.

Selekcija atributa smanjuje dimenzionalnost podataka, uklanja suvišne, nevažne podatke, ili šum u podacima. To donosi neposredne efekte: ubrzanje algoritama *data mining*-a, poboljšanje kvaliteta podataka, performanse *data mining*-a, kao i povećanje razumljivosti dobijenih rezultata.

Algoritmi za selekciju atributa mogu se podeliti na filtere (eng. *filter*) [Almuallim i Dietterich, 1991; Kira i Rendell, 1992], metode prethodnog učenja (eng. *wrappers*) [Kohavi i John, 1997] i ugrađene (eng. *embedded*) pristupe [Blum i Langley, 1997]. Filter metode ocenjuju kvalitet odabranih atributa, nezavisno od algoritma za klasifikaciju, dok su metode prethodnog učenja metode koje zahtevaju primenu klasifikatora (koji bi trebao biti treniran na određenom podskupu atributa) za procenu kvaliteta. Ugrađene metode obavljaju odabir atributa tokom učenja optimalnih parametara (za na primer, neuronske mreže težine između ulaznog i skrivenog sloja).

Neki algoritmi klasifikacije su nasledili sposobnost da se usresrede na relevantne atribute i zanemaruju one nevažne. Stabla odlučivanja su

reprezentativan primer klase takvih algoritama [Breiman, 1984; Quinlan, 1993], ali takođe i MLP neuronske mreže, sa jakim regulisanjem ulaznog sloja, mogu isključiti nevažne atribute na automatski način [Duch *et al.*, 2001]. Takve metode, takođe mogu imati koristi od nezavisnog izbora atributa. S druge strane, neki algoritmi nemaju načine da izvrše izbor relevantnih atributa. K-NN algoritam pripada familiji takvih metoda koje klasifikuju nove primere pronalaženjem najbližih primeraka za trening, snažno se oslanjajući na metode za selekciju atributa.

Istraživači su proučavali različite aspekte selekcije atributa. Pretraga je ključna tema u proučavanju selekcije atributa [Doak, 1992], kao što su početna tačka pretrage, pravac pretrage, i strategija pretrage. Drugi važan aspekt selekcije atributa je kako meriti da li je odgovarajući podskup dobar [Doak, 1992]. Postoje filter metode [Siedlecki i Sklansky, 1988; Fayyad i Irani, 1992; Liu i Setiono, 1996], metode prethodnog učenja [John *et al.*, 1994; Caruana i Freitag, 1994; Dy i Brodley, 2000], a od nedavno i hibridne metode [Das, 2001; Xing *et al.*, 2001]. Prema informacijama o klasama koje su dostupne u podacima, postoje nadzirani [Xing *et al.*, 2001; Dash i Liu, 1997] i nenadzirani pristupi [Dash *et al.*, 1997; Dash i Liu, 1999; Talavera, 1999; Dy i Brodley, 2000].

Glavni cilj ovog rada je proveriti uticaj različitih filter metoda, metoda prethodnog učenja, ugrađenih metoda i ekstrakcije atributa na tačnost klasifikacije. U radu pokazujemo da nema jedne najbolje metode za redukciju dimenzionalnosti podataka, i da izbor zavisi od osobina posmatranog skupa podataka i primenjenih klasifikatora. U praktičnim problemima, jedini način kako bi bili sigurni da je najviša preciznost dobijena je testiranje datog klasifikatora sa više različitih podskupova atributa, dobijenih različitim metodama za selekciju atributa.

2.2. Efekti prethodne selekcije atributa

Neki od algoritama za učenje u toku procesa učenja vrše selekciju atributa nekom ugrađenom metodom. Zašto je onda prethodna selekcija atributa ipak neophodna, i u ovim slučajevima? Pozitivni efekti prethodne selekcije atributa su [Mišković, 2008]:

- smanjenje efekta visoke dimenzionalnosti, čime se popravljaju performanse kada se raspolaže ograničenim brojem primera (za istu preciznost, kod povećanja broja dimenzija za k , neophodno je n^{d+k} instanci-tačaka, što je povećanje za faktor n^k);
- povećanje kvaliteta generalizacije, odnosno tačnosti predviđanja na novim primerima, jer je manja verovatnoća preteranog podešavanja prema trenirajućim podacima, posebno u prisustvu šuma;
- povećanje razumljivosti naučenog znanja;
- značajno smanjenje vremena računanja.

Primena prethodne selekcije atributa je neophodna npr. u bionformatici, analizi slike ili zvuka i sl. Selekcija atributa je posebno značajna tehnika redukcije dimenzionalnosti, jer čuva originalno značenje atributa, koje je razumljivo čoveku. Tehnike transformacije prostora atributa (npr. Analiza glavnih komponenti) i tehnike kompresije na osnovu teorije informacija menjaju originalni model problema uvodeći nove atribute koji nemaju razumljivu interpretaciju u kontekstu problema koji se razmatra.

Veliki značaj u mašinskom učenju ima interakcija atributa, jer atributi u realnim konceptima i bazama podataka uglavnom nisu nezavisni. Određeni broj atributa u modelu često nije u korelaciji sa konceptom i nema isti značaj prilikom klasifikacije novih instanci. Obično, usled preteranog broja irelevantnih atributa u modelu dolazi do preteranog prilagođavanja trening skupu i loših performansi učenja.

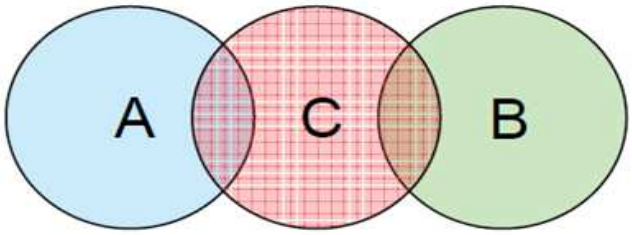
Zbog toga što algoritmi učenja ne mogu dovoljno uspešno da razreše ove situacije, posebno u slučaju velikog broja atributa, pristupa se smanjenju dimenzionalnosti podataka prethodnom selekcijom potencijalno relevantnih atributa.

2.3. Korelacija međusobno nezavisnih i zavisnih atributa s konceptom

U mašinskom učenju, prethodna selekcija podskupa relevantnih atributa zavisi od:

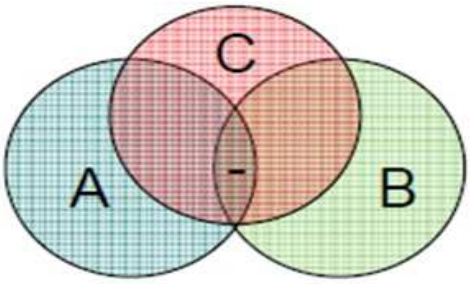
- njihove direktne korelacije sa konceptom, odnosno klasifikacionim atributom,
- od njihovih međusobnih interakcija, preko kojih atributa može biti u jakoj korelaciji sa konceptom, iako svaki pojedinačno nije u značajnijoj direktnoj korelaciji s konceptom.

Na slici 2.1, atributi A i B su međusobno nezavisni, iako oba atributa daju informaciju o klasifikaciji C, oni nemaju ništa zajedničko. Ovakvu situaciju pretpostavljaju mere značajnosti atributa koje istovremeno mere samo značaj pojedinačnih atributa, npr. *Information Gain*. Na slici 2.1, atributi A i B su međusobno nezavisni u odnosu na klasifikaciju C i sve što atributi A i B imaju zajedničko je deo informacije o klasifikaciji C. Navedenu pretpostavku o uslovnoj nezavisnosti atributa imaju metod *Naïve Bayes* i *Bayes*-ove mreže. Ako atributi nisu uslovno nezavisni u odnosu na oznaku klase, stabla odlučivanja su neefikasna. U slučaju da je $I(A;C|B) = 0$, atribut B je irelevantan za predviđanje C, što je osnova selekcije atributa filtriranjem. Pojedinačno uklanjanje atributa koji su u grupnoj korelaciji s konceptom može značajno smanjiti performanse naučenog koncepta, zbog čega je neophodno izvršiti analizu korelacija i identifikovati značajne grupne korelacije. Na slici 2.2, prikazana je korelacija međusobno zavisnih atributa s konceptom.



$$I(AB;C) = I(A;C) + I(B;C)$$

Slika 2.1: Korelacija međusobno nezavisnih atributa s konceptom [Mišković, 2008]



$$I(A;B|C) = 0$$

Slika 2.2: Korelacija međusobno zavisnih atributa s konceptom [Mišković, 2008]

2.4. Klasifikacija atributa

Prisustvo irelevantnih i redundatnih atributa negativno utiče na performanse induktivnog učenja. Moguća je klasifikacija atributa u četiri disjunktne klase:

- irelevantni atributi,
- slabo relevantni redundantni atributi,
- slabo relevantni neredundantni atributi,
- jako relevatni atributi.

Optimalan skup atributa za učenje čine atributi klase slabo relevantni neredundantni atributi i jako relevatni atributi.

2.5. Interakcija u selekciji atributa

Zbog velikog broja kombinacija atributa čije interakcije treba razmotriti $(O(2^N))$, gde je *N* broj atributa u modelu [Almuallim i Dietterich, 1991], dolazi do složenosti analize grupnih korelacija, što je razlog zbog čega se obično pribegava aproksimaciji. Tako npr. izvrši se samo delimična analiza korelacije pojedinih

atributa s klasom $O(N)$ ili se analiziraju samo neke od mogućih kombinacija (interakcije dužine 2 ili 3 atributa).

Autori Jakulin i Bratko [Jakulin i Bratko, 2004] predlažu otkrivanje interakcija pomoću svojstva ireducibilnosti, jer atribut gubi relevantnost kada se uklone atributi koji su s njim u interakciji. Isti autori koriste statističku meru značajnosti za ocenu i prikaz značajnih interakcija u formi grafa interakcija.

Autori [Jakulin i Bratko, 2003; Lavrac *et al.*, 2003] predlažu da se za otkrivanje interakcija koristi mera dobitka interakcija (eng. *interaction gain*), pomoću koga se mogu otkrivati interakcije atributa sa klasom (eng. *2-way*) i dva atributa s klasom (eng. *3-way*).

Prepoznavanje prisustva interakcija atributa radi prethodne selekcije pomoću metode *Relief-F*, koristićemo dalje u eksperimentalnom istraživanju.

Algoritam *Relief-F* vrši ocenu atributa na osnovu toga kako njegove vrednosti iz trenirajućeg skupa razlikuju primere koji su međusobno slični, odnosno bliski i vrši aproksimaciju razlike verovatnoća, što je dato sledećim izrazom:

F(A)=P(različita vrednost A|najbliži primer iz različite klase) –

P(različita vrednost A|najbliži primer iz iste klase) (2.1)

Relief-F, za svaki primer *x*, traži u trenirajućem skupu dva najbliža suseda, jedan iz iste, a drugi iz ostalih klasa (najbliži „pogodak“ x_{hit} i najbliži „promašaj“ x_{miss}) i računa sumu međusobnih rastojanja ovih vrednosti posmatranog atributa A. Na osnovu *n* primera, računa se suma rastojanja vrednosti za taj atribut, njihova srednja vrednost predstavlja ocenu atributa A:

$$F(A) = \frac{1}{n}\sum_{1}^{n}[-difference(x, x_{hit}) + difference(x, x_{miss}) \quad (2.2)$$

Distanca je 1 za različite vrednosti diskretnog atributa, za iste vrednosti je 0, dok je za kontinualne atribute rastojanje razlika samih vrednosti, normalizovana na interval $[0 \ldots 1]$.

Relief-F ima dva bitna poboljšanja:

- u prisustvu šuma u trenirajućem skupu radi pouzdanije ocene, koristi se prosečna udaljenost do *k* primera, umesto udaljenosti do najbližeg i najdaljeg suseda;
- za slučaj izostavljenih vrednosti u primerima, proširena je definicija funkcija rastojanja i rešen je problem učenja više klasa.

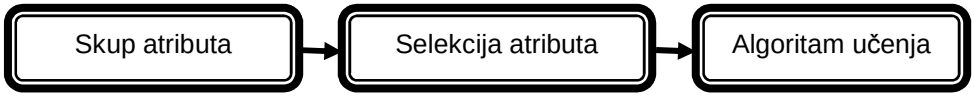
Relief-F ocenjuje i rangira svaki atribut globalnom funkcijom ocene $[-1 \ldots 1]$.

2.6. Metode prethodne selekcije

Raznovrsne tehnike rangiranja i selekcije atributa su predložene u literaturi koja obrađuje problematiku mašinskog učenja. Svrha ovih tehnika je da odbace irelevantne (nevažne) ili redundantne (suvišne) atribute iz datog skupa atributa.

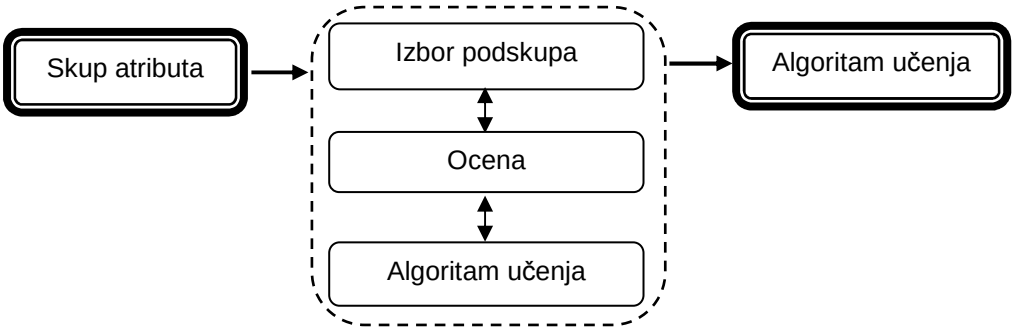
Metode prethodne selekcije pogodnog podskupa atributa dele se na:

- metode filtriranja (eng. *filter methods*),
- metode prethodnog učenja (eng. *wrapper methods*),
- ugrađene metode (eng. *embeeded methods*).



Slika 2.3: Model selekcije atributa filtriranjem, na osnovu [Kohavi i John, 1997]

Kod metode filtriranjem, podskup atributa se bira nezavisno od algoritma učenja, na osnovu neke ocene koja rangira sve atribute, npr. to može biti koeficijent korelacije vrednosti atributa sa vrednostima klasifikacionog atributa (klase).



Slika 2.4: Metod selekcije prethodnim učenjem, na osnovu [Kohavi i John, 1997]

Na slici 2.3. prikazan je model selekcije atributa filtriranjem, a na slici 2.4. metod selekcije prethodnim učenjem.

Kod metode prethodnog učenja podskup atributa se bira prema estimaciji tačnosti predviđanja koju daji izabrani klasifikator nakon učenja pravila za svaki razmatrani podskup. Učenje pravila se vrši nakon selekcije najbolje ocenjenog podskupa. Iscrpno ispitivanje svih mogućih podskupova, prihvatljivo je za mali broj atributa, jer je složenost takvog postupka iz klase složenosti *NP*-težak [Kohavi i John, 1997].

U odnosu na selekciju prethodnim učenjem, koje selekciju atributa posmatra kao spoljašnji sloj procesa indukcije, ugrađene metode selekcije predstavljaju deo osnovnog algoritma indukcije.

Tipični predstavnici ovih metoda su algoritmi za induktivno učenje stabala odlučivanja i produkciona pravila. Kao što su npr. ID3, C4.5, CART i RIPPER. Algoritmi za učenje stabala, koji kreiraju stablo od korena prema listovima i algoritmi učenja pravila, koji obično kreiraju konjuktivna pravila dodavanjem jednostavnih logičkih izraza sa samo jednim atributom, prilikom kreiranja novog čvora ili jednostavnog izraza, koriste funkcije za ocenu i izbor najpogodnijeg atributa za dodavanje u strukturu.

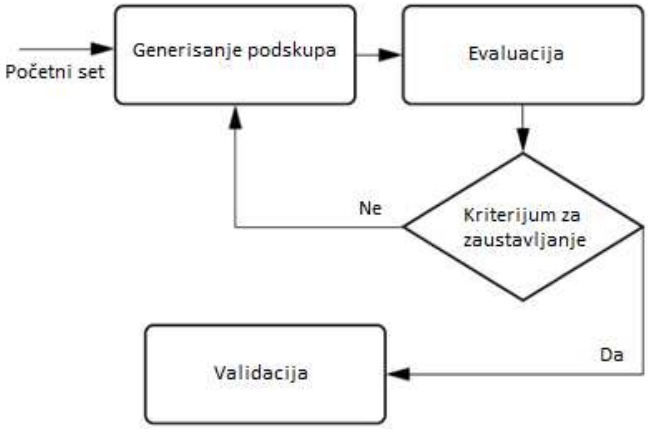
Postupak se prekida kada stablo ili skup pravila obuhvataju sve slučajeve iz obučavajućeg skupa. Atributi koji su upotrebljeni smatraju se relevantnim, dok se ostali izostavljaju iz daljeg razmatranja. Pored modela sekvencijalne selekcije atributa, postoje težinski modeli, gde se koriste težinske ocene.

2.7. Generalna struktura selekcije atributa

Opšta arhitektura većine algoritama za selekciju atributa sastoji se od četiri osnovna koraka (vidi sliku 2.5): generisanje podskupa, evaluacija podskupa, kriterijum za zaustavljanje i rezultat provere [Dash i Liu, 1997]. Algoritmi za selekciju atributa generišu podskup, procenjuju podskup, i rade to u petlji dok se uslov za zaustavljanje ne ispuni. Konačno, pronađeni podskup se proverava uz pomoć algoritma za klasifikaciju na određenim podacima.

Generisanje podskupa je proces traženja, što dovodi do stvaranja podskupova atributa koji će se proveravati. Ukupan broj kandidata za podskupove je 2^N, gde je *N* broj atributa u izvornom skupu podataka, što čini da je pretraživanje kroz prostor svih mogućih rešenja iscrpno, čak i sa umerenim brojem *N*. Ne-deterministički pretraga poput evolucijske pretrage se često koristi za izgradnju podskupova [Yang i Honavar, 1998]. Takođe, moguće je koristiti metode heurističkog pretraživanja. Postoje dve glavne familije tih metoda: dodavanje unapred [Koller i Sahami, 1996] (počevši sa praznim podskupom, možemo dodati atribute nakon lokalne pretrage) ili eliminacija unatrag (suprotno).

Procena podskupa se radi jer svaki podskup generisan od strane postupka za generisanje treba biti ocenjen korišćenjem određenog kriterijuma za ocenjivanje i da se uporedi sa prethodnim najboljim podskupom koji je ispoštovao ovaj kriterijum. Ako se utvrdi da je bolji, onda on zamenjuje prethodni podskup.



Slika 2.5: Generalna struktura selekcije atributa, na osnovu [Dash i Liu, 1997]

Bez odgovarajućeg **kriterijuma za zaustavljanje**, postupak za izbor atributa se može iscrpno izvršavati pre nego što prestane. Proces odabira atributa može prestati pod jednim od sledećih razumnih kriterijuma: (1) unapred definisati broj mogućih atributa koji će biti selektovani, (2) unapred definisati broj iteracija koje će se izvršavati, (3) u slučaju kada prilikom ukidanja ili dodavanja atributa ne postignemo dobijanje boljeg podskupa, (4) optimalni podskup prema kriterijumu procene je postignut.

Za odabrani najbolji podskup atributa treba izvršiti **validaciju** uz pomoć različitih testova koji se rade i na odabranom podskupu i na originalnom skupu i uporediti rezultate koristeći veštački generisane skupove podataka i/ili stvarne skupove podataka.

2.8. Metode filtriranja

U nastavku teksta objasnićemo selekciju atributa metodom filtriranja i daćemo prikaz sledećih metoda: Information Gain (IG), Gain Ratio (GR), Symmetrical Uncertainty (SU), Relief-F (RF), One-R (OR) i Chi-Squared (CS).

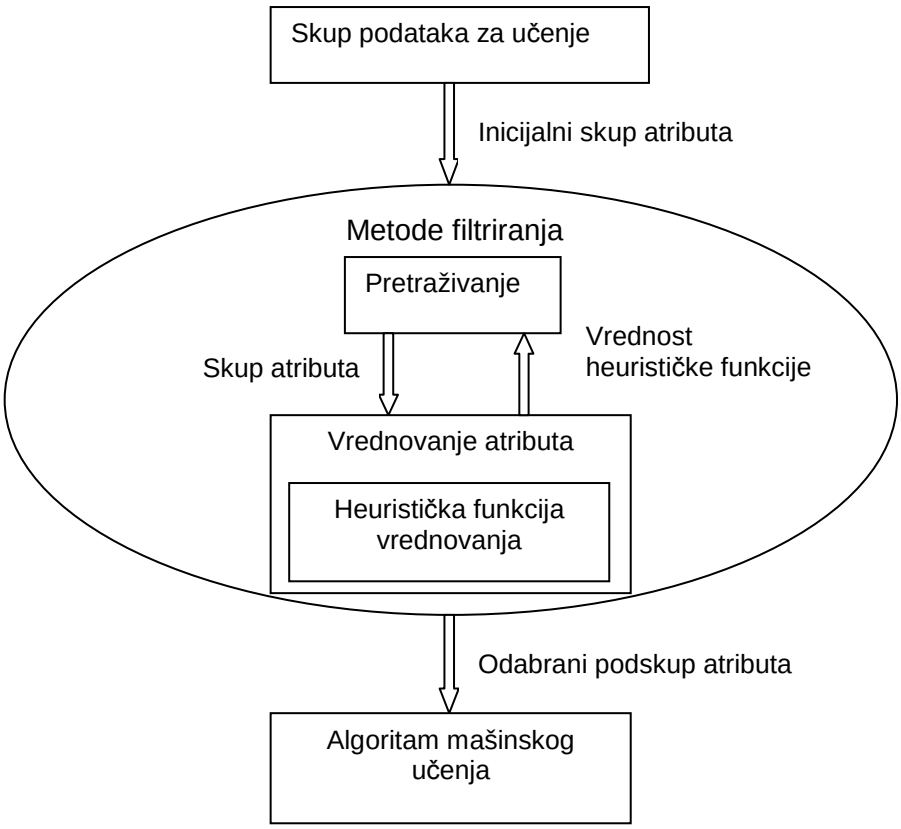
2.8.1. Selekcija atributa metodom filtriranja

Metode filtriranja funkcionišu nezavisno o odabranom algoritmu veštačkog učenja, za razliku od metode selekcije prethodnim učenjem. Kod ovih metoda vrednost atributa se heuristički procenjuje analizom opštih karakteristika podataka iz skupa za učenje. Metode filtriranja koriste više različitih tehnika odabira atributa, jer postoji više načina heurističkog vrednovanja atributa. Ove metode se dele u dve osnovne grupe, zavisno o tome vrednuje li korišćena heuristika podskupove atributa ili pojedinačne atribute. Na slici 2.6. grafički je prikazana selekcija atributa metodama filtriranja.

Prva grupa metoda, čija heuristika vrednuje pojedinačne atribute, odabir atributa vrši rangiranjem atributa prema oceni vrednosti koju proizvodi heuristika, tako što u podskup izabranih atributa ulaze oni atributi čija ocena vrednosti prelazi neki unapred odabrani prag. Takođe, postoji i mogućnost da se formira podskup izabranih atributa tako da se unapred odredi broj (ili relativni odnos) atributa koje podskup treba da sadrži, pa se odgovarajući atributi preuzimaju sa vrha rangirane liste.

Generalno, nedostatak metoda filtriranja koji vrednuju pojedinačne atribute je nemogućnost detekcije redundantnih atributa, zbog čega korelisanost nekog atributa s drugim rezultira sličnom ocenom vrednosti za oba atributa, pa će po pravilu oba atributa biti prihvaćena ili odbačena. Sledeći nedostatak ovih metoda je da je uvrštavanje atributa u konačni podskup prepušteno spoljnim kriterijima praga vrednosti ili broja atributa.

Druga grupa metoda, čija heuristika vrednuje podskupove atributa nemaju problem uvrštavanja atributa u konačni podskup, jer rezultat kojeg vraćaju nije rangirana lista pojedinačnih atributa već najbolje rangirani podskup atributa. S obzirom da se razmatraju podskupovi atributa, moguće je proveravati i redundantnost atributa u podskupu. Kod ove grupe metoda, potrebno je konstruisati heurističku funkciju vrednovanja na način da penalizira postojanje redundantnih atributa u posmatranom podskupu, kako bi se redundantni atributi eliminisali. Kod ove metode, s obzirom da heuristika vrednuje podskupove atributa, potrebno je pronaći podskup koji maksimizira heurističku funkciju vrednovanja. Iscrpno pretraživanje svih podskupova skupa atributa je nepraktično, što je slučaj i kod metoda selekcije prethodnim učenjem, zbog čega se koriste i slični načini pretraživanja.



Slika 2.6: Selekcija atributa metodama filtriranja [Ujević, 2004]

Prihvatljive rezultate daju pozitivna selekcija i negativna eliminacija, a često se koristi i metoda najboljeg prvog. Metode filtriranja koje vrednuju podskupove atributa su vremenski zahtevnije od filtera koji vrednuju pojedinačne atribute, jer postoji potreba pretraživanja podskupova atributa. Međutim, ovi zahtevi su neuporedivo manji u poređenju sa metodama prethodnog učenja jer se u svakom koraku pretraživanja izračunava samo vrednost heuristike vrednovanja, a nije potrebno više puta pozivati algoritam mašinskog učenja.

Heuristike koje vrednuju podskupove atributa često imaju uporište u statističkim postupcima analize podataka. Tako na primer, jedna od heuristika zasniva se na proceni korelacije među različitim atributima, gde se za svaki atribut posmatranog podskupa ocenjuje korelacija s atributom klase, kao i međusobna korelacija atributa u podskupu. Sa povećanjem korelacije atributa i klase vrednost heurističke funkcije raste, i opada ako se povećava međusobna korelisanost atributa. Zbog toga će nevažni atributi biti odbačeni jer nisu korelisani s klasom, a redundantni zbog visoke korelacije sa preostalim atributima podskupa.

Pored oslanjanja na statističke postupke, heuristika se može zasnivati i na postupcima mašinskog učenja. Neki algoritmi mašinskog učenja imaju ugrađene

mehanizme izbora atributa, pa se mehanizmi koji su inherentni jednom postupku mogu iskoristiti i u drugim postupcima kod kojih takvi mehanizmi ne postoje. Takav primer je upotreba stabala odlučivanja za odabir atributa, kada se na potpunom skupu podataka za učenje konstruiše stablo odlučivanja, pa se biraju samo oni atributi koji se zaista koriste u konstruisanom stablu, a onda se u fazi modeliranja koristi drugi algoritam mašinskog učenja. Opisani postupak je opravdan ako algoritam na redukovanom skupu podataka za učenje pokaže bolje klasifikacijske performanse od algoritma korišćenog za izbor podataka.

Generalno, izbor atributa metodama filtriranja traje znatno kraće u poređenju sa metodama prethodnog učenja, posebno kad su u pitanju skupovi podataka sa većim brojem atributa [Hall, 1999], zbog čega su metode filtriranja često praktičnije rešenje za analizu podataka od drugih metoda. Metode filtriranja se zbog nezavisnosti o algoritmu mašinskog učenja mogu koristiti u kombinaciji sa bilo kojom tehnikom modeliranja podataka, za razliku od metoda prethodnog učenja koje se moraju ponovo izvoditi pri svakoj promeni ciljne tehnike modeliranja.

2.8.2. Prikaz korišćenih metoda filtriranja u radu

U ovom radu, koristimo sledeće metode filtriranja za rangiranje atributa koje su statistički i entropijski zasnovane, a pokazuju dobre performanse u različitim domenima:

- *Information Gain* (IG),
- *Gain Ratio* (GR),
- *Symmetrical Uncertainty* (SU),
- *Relief-F* (RF),
- *One-R* (OR),
- *Chi-Squared* (CS).

Mera entropije se obično koristi u teoriji informacija [Abe i Kudo, 2005], koja karakteriše čistoću proizvoljnog uzorka, odnosno meru homogenosti skupa primera. Ona je u osnovi sledećih metoda: IG, GR i SU. Mera entropije se smatra merom nepredvidljivosti sistema. Entropija *Y* može se predstaviti kao:

$$H(Y) = -\sum_{y \in Y} p(y) log_2 (p(y)) \tag{2.3}$$

gde je $p(y)$ granična funkcija gustine verovatnoće za slučajnu promenljivu *Y*. Ako posmatrane vrednosti *Y* u trenirajućem skupu podataka *S* su podeljene u skladu sa vrednostima drugog atributa *X*, i entropija od *Y* sa obzirom na particiju uzrokovanu *X*-om je manja od entropije *Y* pre podele, onda postoji veza između atributa *Y* i *X*. Entropija od *Y* nakon posmatranja *X* je tada:

$$H(Y/X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y/x) \log_2 (p(y/x)) \tag{2.4}$$

gde je $p(y|x)$ uslovna verovatnoća od *Y* data *X*-om.

2.8.2.1. Information Gain

Statistička vrednost nazvana *informacisjki dobitak* je dobra kvantitativna mera vrednosti atributa za klasifikaciju primera kojom se meri kako dobro dati atribut razdvaja primere prema njihovoj klasifikaciji. Pored entropije kao mere „nečistoće“ u skupu primera, možemo definisati i meru efektivnosti atributa u klasifikaciji primera. Informacijski dobitak predstavlja očekivanu redukciju entropije uzrokovanu razdvajanjem primera na osnovu tog atributa.

S obzirom da je entropija merilo nečistoće u *S* trening skupu, možemo definisati meru koja odražava dodatne informacije o *Y* koje smo dobili od *X* koja predstavlja iznos za koji se smanjuje entropija *Y* [Dash i Liu, 1999]. Ova mera je poznata kao IG. Data je kao:

$$IG = H(Y) - H\left(Y/X\right) = H(X) - H\left(X/Y\right) \tag{2.5}$$

IG je simetrična mera (vidi jednačinu (2.5)). Dobijene informacije o *Y* nakon posmatranja *X* su jednake informacijama dobijenim o *X* nakon posmatranja *Y*. Slabost IG kriterijuma je da je pristrasan u korist atributa sa više vrednosti čak i kada nisu više informativne.

2.8.2.2. Gain Ratio

Gain Ratio je ne-simetrična mera koja je uvedena da nadoknadi pristranost IG [Hall i Smith, 1998]. GR je dato kao:

$$GR = \frac{IG}{H(X)} \tag{2.6}$$

Kao što je jednačina (2.6) predstavlja, kada varijabla *Y* mora da se predvidi, možemo normalizirati IG deljenjem entropijom od *X*, i obratno. Zbog ove normalizacije, GR vrednosti uvek su u rasponu od [0, 1]. Vrednost GR = 1 označava da je poznavanje *X* u potpunosti predviđa *Y*, i GR = 0 znači da ne postoji odnos između *Y* i *X*. Suprotno od IG, GR favorizuje varijable sa manjim vrednostima.

2.8.2.3. Symmetrical Uncertainty

Simetrična neizvesnost (eng. *Symmetrical Uncertainity* – SU) predstavlja metod selekcije atributa koji iz kompletnog skupa atributa eliminiše irelevantne atribute, na osnovu mere relevantnosti. SU se definiše na osnovu entropije atributa *H* kao:

$$SU(X,Y) = 2 \cdot \left[\frac{IG(X|Y)}{H(X) + H(Y)}\right] \tag{2.7}$$

gde je $IG(X|Y) = H(X) - H(X|Y)$, a $H(X|Y) = \sum_j P(y_j) \sum_i P(x_i|y_j) \log_2 P(x_i|y_j)$.

Kriterijum simetrične neizvesnosti kompenzuje inherentnu (urođenu) pristranosti IG tako što deli zbir entropija od *X* i *Y* [Dash i Liu, 1999]. Može se prikazati kao:

$$SU = 2\frac{IG}{H(Y) + H(X)} \tag{2.8}$$

SU uzima vrednosti, koje su normalizovane u rasponu [0, 1], zbog korektivnog faktora dva. Vrednost SU = 1 znači da je poznavanje jednog atributa potpuno predviđa, a SU = 0 označava da su *X* i *Y* nekorelisani. Slično GR, SU je pristrasan prema atributima sa manje vrednosti.

2.8.2.4. Chi-Squared

Izbor atributa putem *chi square* (X^2) testa je još jedan, vrlo često korišćen metod [Liu i Setiono, 1995]. *Chi square* procena atributa procenjuje vrednost atributa računanjem vrednosti *chi square* s obzirom na klasu. Početna H0 hipoteza je pretpostavka da dva atributa nisu povezana, i to je testirano od strane *chi square* formule:

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{ij} - E_{ij})}{E_{ij}^2} \tag{2.9}$$

gde je O_{ij} posmatrana frekvencija i E_{ij} je očekivana (teoretska) frekvencija, potvrđena nultom hipotezom. Što je veća vrednost χ^2, to je veći dokaz protiv hipoteze H0.

2.8.2.5. One-R

One-R je jednostavan algoritam koji je predlažio Holte [Holte, 1993]. On gradi jedno pravilo za svaki atribut u skupu podataka za učenje, a zatim odabira pravilo sa najmanjom greškom. On tretira sve numeričke vrednosti kao kontinuirane i koristi jednostavan način za deljenje raspona vrednosti u nekoliko disjunktnih intervala. On obrađuje nedostajuće vrednosti kao „nedostaje“ i to kao legitimnu vrednost. To je jedna od najprimitivnijih šema, jer stvara jednostavna pravila na osnovu samo jednog atributa. Iako predstavlja minimalni oblik razvrstavanja, on može biti koristan za određivanje osnovnih performansi i kao merilo uspešnosti ostalih algoritama učenja.

2.8.2.6. Relief-F

Relief–F za procenu atributa [Marko i Igor, 2003], procenjuje vrednost atributa ponavljajući uzorkovanja instanci i razmatrajući vrednost dobijenih atributa od najbližih instanci iste ili različite klase. Ova metoda dodeljuje ocenu težine za

svaki atribut na osnovu sposobnosti razlikovanja među klasama, a zatim bira one atribute čija težina prelazi korisnički definisani prag kao odgovarajućih atributa.

Izračunavanje težina se zasniva na verovatnoći najbližih suseda iz dve različite klase koje imaju različite vrednosti za atribute i verovatnoći da dva od suseda iz iste klase ima istu vrednost atributa. Ako je veća razlika između ove dve verovatnoće, atribut je više značajan. Inherentno, mera definisana za dve klase problema, može se proširiti i na više klasa, deleći problem u nizove od dve klase problema.

Relief–F [Marko i Igor, 2003] je metoda filtriranja kod koje se vrednuju pojedinačni atributi, a originalno je zamišljena za klasifikacijske probleme sa samo dve klase. Ovo je iterativna metoda koja u svakoj iteraciji koriguje podatak o važnosti pojedinog atributa, gde inicijalno, svi atributi imaju jednake težinske vrednosti. Kod ove metode u svakoj iteraciji postupka se na slučajan način bira jedan primer iz skupa podataka za učenje, a zatim se u skupu podataka za učenje pronalaze njegovi najbliži susedi iz iste i suprotne klase. Upoređivanjem vrednosti svakog atributa u izabranom primeru i pronađenim susedima ažuriraju se težinske vrednosti atributa. Generalno, važni atributi bi trebali imati bliske vrednosti za primere iste klase, a različite za primere suprotnih klasa, zbog čega se različite vrednosti atributa za primere suprotnih klasa boduju pozitivno, a za primere iste klase negativno. Kod ove metode ovaj postupak se ponavlja unapred definisani broj puta i na kraju, težinske vrednosti atributa predstavljaju ocenu njihove vrednosti. Generalno, pouzdanost ocene vrednosti raste s brojem iteracija, ali se produžava i vreme izvođenja.

Kod ove metode naknadno je opisani postupak proširen na probleme sa više klasa i dodat je mehanizam tretiranja šuma u podacima [Kononenko, 1994]. Problemi sa više klasa tretiraju se posmatranjem najbližih suseda iz svih preostalih klasa, te se njihov uticaj uzima u zavisnosti o apriornoj verovatnoći svake od klasa. Kod ove metode umanjuje se uticaj šuma u podacima usrednjavanjem doprinosa *k* najbližih suseda iz iste i različitih klasa za svaki slučajno odabrani primer.

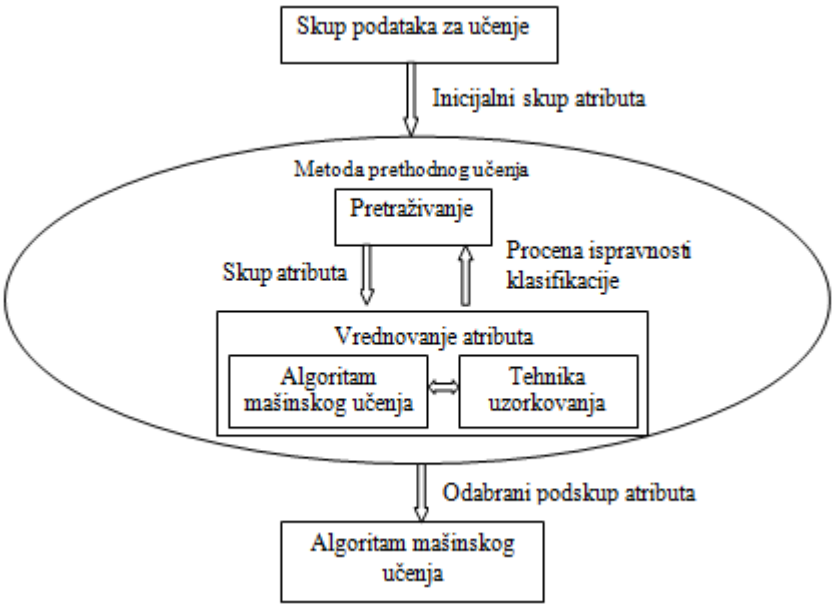
Proizvoljnost u izboru primera je negativna strana ove metode, te će svako pokretanje postupka vrednovanja atributa proizvesti različite težinske vrednosti zbog drugačijeg izbora primera.

2.9. Metode prethodnog učenja

Kod metoda prethodnog učenja koriste se određeni algoritmi za modeliranje kako bi se ocenili podskupovi atributa u odnosu na njihovu klasifikacijsku ili prediktivnu moć. Kod korišćenja ovih metoda u praksi se pojavljuju tri pitanja:

- kako pretražiti prostor svih mogućih podskupova atributa,
- kako proceniti uspešnost algoritma za modeliranje s obzirom na pretraživanje skupa atributa,
- koji postupak modeliranja koristiti kao crnu kutiju za metode prethodnog učenja.

Kod metoda prethodnog učenja vrednost određenog skupa atributa izražava se pomoću stepena ispravnosti klasifikacije koju postiže model konstruisan uz korišćenje tih atributa. To znači da su ove metode tesno vezane za odabrani algoritam mašinskog učenja. Za zadati podskup atributa, ispravnost klasifikacije se ocenjuje korišćenjem tehnika uzorkovanja, na primer unakrsnom validacijom (eng. *cross-validation*). Kod ovih metoda za svaki posmatrani podskup atributa izgrađuje se model i ocenjuju se njegove performanse, tako da bolje performanse nekog modela ukazuju na bolji izbor atributa iz kojih je model nastao. Grafički prikaz metode prethodnog učenja i izbor atributa dat je na slici 2.7.



Slika 2.7: Metode prethodnog učenja i izbor atributa [Ujević, 2004]

Kod metoda prethodnog učenja postupak izbora atributa je računski vrlo zahtevan zbog učestalog izvođenja algoritma mašinskog učenja. Potrebno je dobiti ocenu performansi odgovarajućeg modela za svaki posmatrani podskup atributa, a metode ocene ispravnosti modela uglavnom zahtevaju usrednjavanje rezultata po većem broju izgrađenih modela. Kod ovih metoda za svaki posmatrani podskup atributa izgrađuje se više modela, a ukupan broj podskupova eksponencijalno raste s povećanjem broja atributa.

Iscrpno pretraživanje podskupova atributa se može sprovesti samo za mali broj atributa, budući da je taj problem *NP*-težak. Zato se koriste razne tehnike pretraživanja, kao što su: najbolji prvi (eng. *best-first*), granaj-pa-ograniči (eng. *branch-and-bound*), simulirano kaljenje (eng. *simulated annealing*), genetski algoritmi i sl. [Kohavi i John, 1997]. U praksi se pokazuje da pohlepne tehnike pretraživanja daju dobre rezultate, što znači da se nikad ne proveravaju već donesene odluke o tome da li da se atribut uključi (ili isključi) iz skupa.

1. $s :=$ **početno stanje**
2. **ponavljaj**
   2.1. $state := s$
   2.2. **pronađi sve lokalno dostupne alternative za** $state$
   2.3. izračunaj ocenu e(t) za svaku od pronađenih alternativa
   2.4. $s :=$ **alternativa sa najvećom ocenom** $e(s)$ **dok je** $e(s) > e(state)$
3. vrati state

Slika 2.8: Pohlepna tehnika [Ujević, 2004]

Pohlepne tehnike prostor rešenja prelaze tako da u svakom koraku pregledaju lokalno dostupne alternative, pa proces pretraživanja nastavljaju od najbolje od njih (tehnika uspona na vrh). Na slici 2.8. dat je prikaz pseudo koda za pohlepne tehnike.

Pohlepne tehnike se dele na izbor atributa unapred (eng. *forward selection*) i eliminacija atributa unatrag (eng. *backward elimination*) [Kohavi i John, 1997]. Moguće je i pretraživanje u oba smera (eng. *bidirectional search*).

Kod izbora atributa unapred postupak počinje sa praznim skupom atributa i u svakom koraku postupka dodaje se po jedan novi atribut. Eliminacija atributa unatrag je obrnuti postupak koji počinje sa punim skupom atributa i u svakom koraku se oduzima po jedan atribut. Opisani postupci su vrlo jednostavni, ali daju rezultate uporedive sa složenijim tehnikama pohlepnog pretraživanja kao što su zrakasto pretraživanje ili metoda najboljeg prvog.

Ako sa *n* označimo ukupan broj atributa, izbor atributa unapred i eliminacija atributa unatrag imaju složenost $O(n^2)$, i s obzirom da proizvode prihvatljive rezultate u razumnom vremenu, upravo ove dve tehnike pretraživanja se najčešće koriste u izboru atributa metodama prethodnog učenja.

Generalno, eliminacija atributa unatrag preferira veće podskupove atributa i može rezultirati nešto boljim klasifikacijskim performansama od odabira atributa unapred. S obzirom da se vrednost skupa atributa meri procenom ispravnosti klasifikacije, onda se zbog samo jedne optimistične procene oba postupka mogu preuranjeno završiti, i u tom slučaju eliminacija atributa unatrag će odabrati previše atributa, a odabir atributa unapred premalo. Usled nedostatka prognostičkih atributa može se ograničiti sposobnost zaključivanja, što će odraziti na nešto slabije klasifikacijske performanse. Manji broj izabranih atributa je poželjan u slučajevima kada je primarni cilj razumevanje međuzavisnosti i pravilnosti u podacima, jer su konstruisani modeli jednostavniji i naglašavaju najprediktivnije atribute.

Kod metoda prethodnog učenja najvažniji nedostatak je sporost pri izvođenju uslovljena pozivanjem ciljnog algoritma mašinskog učenja više puta, zbog čega ovim metodama ne odgovaraju obimni skupovi podataka za učenje sa većim brojem atributa.

Smatra se da metode prethodnog učenja omogućuju postizanje nešto boljih performansi klasifikacije, zbog tesne povezanosti s ciljnim algoritmom mašinskog učenja. Ovo ujedno može predstavljati i opasnost jer preterano

prilagođavanje skupa za učenje ciljnom algoritmu može naglasiti njegove nedostatke.

2.10. Ugrađene metode

Prethodno objašnjene metode selekcije atributa, metode filtriranja i prethodnog učenja, razmatraju selekciju atributa kao spoljašnji sloj procesa indukcije. Ugrađene metode vrše selekciju atributa u sklopu osnovnog algoritma induktivnog učenja, odnosno kao deo procesa generalizacije.

Neki od algoritama koji vrše selekciju atributa na ovakav način su [Mišković i Milosavljević, 2010]:

- algoritmi za induktivno učenje stabala odlučivanja, na primer C4.5,
- algoritmi za induktivno učenje pravila, npr. C45Rules, RIPPER i Empiric.Rules,
- neki algoritmi učenja neuronskih mreža, koji mogu da istovremeno vrše selekciju relevantnih atributa, npr. *Optimal Brain Damage*,
- neki algoritmi učenja metodom potpornih (eng. *support*) vektora, npr. *l1*-norm SVM i *lasso*.

Algoritmi za induktivno učenje stabala odlučivanja i algoritmi za induktivno učenje pravila minimizuju funkciju gubitka dodavanjem u konačni opis samo one atribute čije ispitivanje dovoljno smanjuje grešku na obučavajućem skupu. Učenje se prekida kada naučeno pravilo obuhvata najveći mogući broj slučajeva iz obučavajućeg skupa. Atributi koji su upotrebljeni smatraju se relevatnim, dok ostali atributi se izostavljaju. Na ovakav način, rešenje se dobija brzo, i ono je razumljivo za korisnika.

Algoritmi učenja ansambala u obliku slučajnih šuma (eng. *random forest*) se mogu iskoristiti i za ocenu važnosti atributa. Ovi algoritmi koriste tehniku zašumljavanja, koja se sastoji u permutovanju vrednosti atributa i učenju slučajnih stabala pre i posle ove promene.

Algoritam učenja produkcionih pravila Empiric.Rules vrši automatsku selekciju relevantnih atributa ugrađenim metodom, izborom nekog od više informacionih kriterijuma.

Kod standardne verzije algoritama učenja metodom potpornih vektora, svi težinski koeficijenti su različiti od nule, tako da algoritam koristi ravnopravno sve atribute. Ovi algoritmi vrše selekciju atributa indirektno, tako što koriste linearnu normu, tako da veliki broj težina poprima vrednost blisku nuli, čime se iz modela implicitno uklanjaju redundatni atributi.

U mnogim slučajevima obučavajući skupovi su oskudni i postoje međusobne interakcije atributa. Za selektovanje optimalnog podskupa atributa koriste se različite estimacije, koje se zasnivaju na različitim statističkim pretpostavkama, npr. nezavisnost atributa i dovoljan broj obučavajućih primera, koje nisu uvek zadovoljene. Zbog toga, ugrađene metode selekcije atributa, nisu

uvek dovoljne, pa se u mnogim praktičnim situacijama koriste metode prethodne selekcije atributa kako bi se performanse poboljšale.

2.11. Ekstrakcija atributa

U nekim aplikacijama skupovi podataka s hiljadama atributa nisu retkost, pri čemu nekada svi atributi mogu biti značajni za neke probleme, ali za neke ciljane namere samo mali podskup atributa je obično relevantan. Problem dimenzionalnosti se može prevladati:

- tako da se odabere samo podskup relevantnih atributa, ili
- stvaranjem novih atributa koji sadrže najviše informacija o klasi.

Prva metodologija se zove selekcija atributa, a druga se zove ekstrakcija atributa, a to uključuje linearnu (Analiza glavnih komponenata (eng. *Principal Component Analysis* - PCA), Nezavisnu analizu komponenti (ICA) i sl.) i ne-linearnu metodu ekstrakcije atributa. U nastavku teksta biće reči o PCA.

Karl Pearson je 1901. godine prvi opisao mogućnosti analize glavnih komponenata, ali Hotelling je dosta kasnije 1933. godine razradio praktične računske metode. Zbog kompleksnog računa, veća primena ove tehnike, usledila je sa dostupnošću računara [Manly, 1986].

PCA predstavlja tehniku formiranja novih, sintetskih varijabli koje su linearne složenice - kombinacije izvornih varijabli. Ovom tehnikom se redukuje dimenzionalnost, a koristi se u svrhu postizanja preglednosti i pojednostavljenja velikog broja podataka. Kod ove tehnike maksimalni broj novih varijabli koji se može formirati jednak je broju izvornih, a nove varijable nisu međusobno korelisane [Sharma, 1996]. Kod ove tehnike najvažniji aspekt je sažimanje i analiza linearne povezanosti većeg broja multivarijatno distribuiranih, kvantitativnih, međusobno korelisanih varijabli u smislu njihove kondenzacije u manji broj komponenti, novih varijabli, međusobno nekorelisanih, sa minimalnim gubitkom informacija.

	Varijable				
Vektori	X_1	X_2	X_3	...	X_p
1	x_{11}	x_{12}	x_{13}	...	x_{1p}
2	x_{21}	x_{22}	x_{23}	...	x_{2p}
⋮	⋮	⋮	⋮	...	⋮
n	x_{n1}	x_{n2}	x_{n3}	...	x_{np}

Slika 2.9: Ulazni podaci za analizu glavnih komponenata [Sharma, 1996]

Za analizu glavnih komponenata ulazni podaci čine *p* varijabli (obeležja, atributi, parametri ili slično) i *n* vektora (opažaja, individua, ispitanika, merenja, objekata ili slično) i podaci se mogu interpretirati kao *n* tačaka u *p*-dimenzionalnom vektorskom prostoru i imaju oblik matrice *p x n*. Na slici 2.9. prikazani su ulazni podaci za analizu glavnih komponenata.

Može se smatrati da su dve varijable koje su visoko korelisane istog ili sličnog sadržaja, pri čemu se ovom metodom veći broj takvih varijabli zamenjuje manjim brojem varijabli. Zato se vrši transformacija koordinatnog sistema, gde projekcije varijabli ulaznih podataka na koordinatne ose novog koordinatnog sistema predstavljaju nove, veštačke, varijable – glavne komponente (eng. *principal component*) koje se dobijaju kreiranjem *p* linearnih kombinacija izvornih varijabli. Cilj analize je kreiranje *p* linearnih kombinacija izvornih varijabli koje se nazivaju glavne komponente [Sharma,1996]:

$$\begin{aligned} \xi_1 &= \omega_{11}X_1 + \omega_{12}X_2 + \cdots + \omega_{1p}X_p \\ \xi_2 &= \omega_{21}X_1 + \omega_{22}X_2 + \cdots + \omega_{2p}X_p \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ \xi_p &= \omega_{p1}X_1 + \omega_{p2}X_2 + \cdots + \omega_{pp}X_p \end{aligned} \qquad (2.10)$$

gde su ξ_1, ξ_2, ….ξ_p glavne komponente, a ω_{ij} su koeficijenti tj. konstante koje čine koeficijente *j*-te varijable za *i*-tu glavnu komponentu. Konstante ω_{ij} se nazivaju svojstveni ili latentni vektori (eng. *eigenvectors*) i u geometrijskom smislu su u dvodimenzionalnoj strukturi ustvari, sinusi i kosinusi uglova novih osi, tj. glavnih komponenata. Transformisane vrednosti izvornih varijabli (2.10) predstavljaju zbirove glavnih komponenata (eng. *principal component scores*).

Konstante ω_{ij} procenjene su tako da je:

1. Ukupna varijansa je suma varijansi svih izvornih varijabli. Deo te ukupne varijanse objašnjen jednom glavnom komponentom naziva se svojstvena vrednost ili latentni koren. Svojstvena vrednost je najveća u prvoj glavnoj komponenti i u svakoj sledećoj njena je vrednost sve manja. Prva glavna komponenta, ξ_1, objašnjava maksimum varijanse iz podataka, druga glavna komponenta, ξ_2, objašnjava maksimum varijanse koja je ostala neobjašnjena prvom i tako dalje.

2. $\omega_{i1}^2 + \omega_{i2}^2 + \cdots \omega_{ip}^2 = 1 \quad i = 1 \ldots p$ (2.11)

3. $\omega_{i1}\omega_{j1} + \omega_{i2}\omega_{j2} + \cdots + \omega_{ip}\omega_{jp} = 0$ za sve $i \neq j$ (2.12)

Zbog neophodnosti fiksiranja skale novih varijabli zadat je uslov da zbir kvadrata konstanti iznosi 1, iz jednačine (2.11), kako ne bi bilo moguće povećati varijansu linearne kombinacije jednostavnom promenom skale. Uslov iz jednačine (2.12) osigurava međusobnu nekorelisanost novih varijabli, odnosno nove ose su međusobno ortogonalne.

S obzirom da je suma svih svojstvenih vrednosti jednaka ukupnoj varijansi cilj je iteracijskim postupkom, izdvojiti veći deo ukupne varijanse u nekoliko prvih glavnih komponenata i time redukovati broj izvornih varijabli. Svojstvena vrednost je zapravo varijansa izračunata iz seta zbirova glavne komponente, što se može prikazati sledećim jednačinama:

$$\begin{aligned}
\lambda x_1 &= \omega_{11}x_1 + \omega_{12}x_2 + \cdots + \omega_{1p}x_p \\
\lambda x_2 &= \omega_{21}x_1 + \omega_{22}x_2 + \cdots + \omega_{2p}x_p \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
\lambda x_p &= \omega_{p1}x_1 + \omega_{p2}x_2 + \cdots + \omega_{pp}x_p
\end{aligned}$$

ili u obliku matrice:

$$\lambda x = Wx \text{ ili } (W - \lambda \mathbf{I})x = \mathbf{0}, \tag{2.13}$$

gde je **I** jedinična matrica *p x p* sa vrednosti jedan u dijagonali, 0 je *p x 1* nul-vektor, a vrednosti skalara λ svojstvene su vrednosti matrice W . Ako se za *i*-tu svojstvenu vrednost λ_i, postavi x_1 = 1, tada se rezultirajući vektor sa *x* vrednosti:

$x_i = \begin{bmatrix} 1 \\ x_{2i} \\ x_{3i} \\ \vdots \\ x_{ni} \end{bmatrix}$ zove *i*-ti svojstveni vektor matrice A.

Proces dobijanja svojstvenih vektora i vrednosti je ključni matematički problem, a rešava se pomoću rastavljanja svojstvenih vrednosti, koji izražava bilo koju matricu tipa *n x p* (gde je n ≥ p) kao trostruki produkt tri matrice P, D i Q tako da

$$X = PDQ', \tag{2.14}$$

gde je X matrica tipa *n* x *p* ranga kolone *r*, P je *n* x *r* matrica, D je dijagonalna matrica *r* x *r*, a Q´ je matrica *r* x *p*. Matrice P i Q su ortogonalne pa je

$$P'P = \mathbf{I} \tag{2.15}$$

i

$$Q'Q = \mathbf{I}. \tag{2.16}$$

Kolona *p* matrice Q´ sadrži svojstvene vektore matrice X´X, a dijagonala matrice D sadrži korenske vrednosti korespondirajućih svojstvenih vrednosti matrice X´X. Takođe, svojstvene vrednosti matrica X´X i XX´ su iste. Ulazna matrica može biti ili matrica kovarijansi ili matrica korelacija, zavisno o problemu, tipu varijabli i skali njihovog merenja. Matrica kovarijansi *C* je simetrična, a kovarijanse cov_{ii} su varijanse S_i^2:

$$C = \begin{bmatrix} cov_{11} & cov_{12} \cdots & cov_{1p} \\ \vdots & \ddots & \vdots \\ cov_{p1} & cov_{p2} \cdots & cov_{pp} \end{bmatrix} \tag{2.17}$$

Matrica korelacija *R* (kao i *C*) mora biti simetrična:

$$R = \begin{bmatrix} r_{11} & r_{12} \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} \cdots & r_{pp} \end{bmatrix} = \begin{bmatrix} 1 & r_{12} \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} \cdots & 1 \end{bmatrix} \quad (2.18)$$

Očekuje se da će većina novih varijabli činiti šum, i imati tako malu varijansu da se ona može zanemariti. Većinu informacija sadrži prvih nekoliko ξ varijabli - glavnih komponenti, čije su varijanse značajne veličine. Na taj način, iz velikog broja izvornih varijabli kreirano je tek nekoliko glavnih komponenti koje nose većinu informacija i čine glavni oblik.

Međutim, ima situacija kada to nije tako i to u slučaju kada su izvorne varijable nekorelisane, tada analiza ne daje povoljne rezultate. Kada su izvorne varijable visoko pozitivno ili negativno korelisane mogu se postići najbolji rezultati. U tom slučaju se može očekivati da će na primer 20-30 varijabli biti obuhvaćeno sa 2 ili 3 glavne komponente.

U analizi glavnih komponenata osnovni koraci su:

- Potrebno je standardizirati varijable tako da im je prosek 0, a varijansa 1 kako bi sve bile na jednakom nivou u analizi, jer je većina setova podataka konstruisana iz varijabli različitih skala i jedinica merenja.
- Izračunati matrice korelacija između svih izvornih standardizovanih varijabli.
- Potom, pronaći svojstvene vrednosti glavnih komponenata.
- I na kraju, odbaciti one komponente koje nose proporcionalno malog udela varijanse (obično prvih nekoliko nose 80% - 90% ukupne varijanse).

Za interpretaciju glavnih komponenata osnovu čine svojstveni vektori. Njihove vrednosti su u prvoj glavnoj komponenti, najčešće, relativno ravnomerno raspoređene po svim izvornim varijablama, dok u drugoj glavnoj komponenti dolazi do njihove veće disproporcije, što omogućava izdvajanje izvorne varijable (ili tek nekoliko njih) sa jačim učešćem i pomaže u objašnjavanju i sažimanju ukupne varijabilnosti.

Dobijeni na ovaj način, zbirovi glavnih komponenata mogu poslužiti još i:

- za daljnju interpretaciju rezultata grafičkim predstavljanjem, čime se njihov relativni međusobni položaj može i vizuelno ispitati,
- kao ulazne varijable u drugim multivarijatnim metodama kao npr. klaster, regresijska i diskriminantna analiza. Prednost korišćenja zbirova je u tome što nove varijable nisu međusobno korelisane čime je rešen problem multikolinearnosti. Ali, probleme druge vrste tada može izazvati nemogućnost smislene interpretacije glavnih komponenata.

3. EVALUACIJA KLASIFIKACIJSKIH MODELA

U trećem delu razmatraju se mere za evaluaciju klasifikacijskih modela kao i metode za ocenu stvarne frekvencije grešaka klasifikacijskog modela, koje se razlikuju po pristupu problemu i svojstvima koje pokazuju. Takođe, biće reči o tome da prilikom treninga postoji mogućnost da se model previše prilagodi specifičnostima podataka za trening i da zbog toga daje lošije rezultate kada se primeni na drugim podacima.

3.1. Mere za evaluaciju klasifikacijskih modela

Za modeliranje pravilnosti u podacima postoji veći broj metoda. Takođe, variranjem parametara metode, pojedine metode na istom skupu primera za učenje rezultiraju različitim modelima. S obzirom da isti problem i isti skup podataka za učenje mogu proizvesti veći broj različitih modela, time se naglašava potreba za vrednovanjem kvaliteta modela u odnosu na posmatrani problem. To je razlog zbog čega je evaluacija otkrivenog znanja jedna od bitnih komponenti procesa inteligentne analize podataka. S obzirom da u ovom radu razmatramo klasifikacijske probleme, u daljem tekstu će biti reči o evaluaciji klasifikacijskih modela.

Zadatak evaluacije klasifikacijskih modela je izmeriti u kojem stepenu klasifikacija sugerisana izgrađenim modelom odgovara stvarnoj klasifikaciji primera, i u zavisnosti od načina posmatranja performansi modela, postoji više različitih mera za njihovu evaluaciju. U zavisnosti od karakteristika posmatranog problema i načina njegove primene, vrši se izbor najpogodnije mere.

Pri evaluaciji klasifikacijskih modela osnovni pojam je pojam greške. Ukoliko primena klasifikacijskog modela na izabranom primeru dovodi do rezultata prognoze klase koja je različita od stvarne klase primera onda je nastala greška prilikom klasifikacije. Ako je svaka greška podjednako značajna, tada je ukupan broj grešaka na posmatranom skupu primera dobar indikator rada klasifikacijskog modela.

Na ovom pristupu se zasniva tačnost kao mera za evaluaciju kvaliteta klasifikacijskih modela. Ovu meru možemo definisati kao odnos broja ispravno klasifikovanih primera prema ukupnom broju klasifikovanih primera.

$$Tačnost = \frac{broj\ ispravno\ klasifikovanih\ primera}{ukupan\ broj\ primera} \tag{3.1}$$

Osnovni nedostaci tačnosti kao mere za evaluaciju su sledeći: (1) zanemaruju se razlike između tipova grešaka; (2) zavisna je o distribuciji klasa u skupu podataka, a ne o karakteristikama primera.

U većem broju slučajeva u praktičnom rešavanju problema vrlo je važno razlikovati određene tipove grešaka. To je čest slučaj u medicini i npr. otkrivanju postojanja oboljenja kod pacijenta. Ako sistem treba da klasifikuje tkiva dojke na maligna i benigna na osnovu mamografskog snimka, onda ako sistem pogrešno označi obolelo tkivo kao zdravo tkivo, greška ima veći značaj, jer se neće uočiti postojanje bolesti i neće se primeniti odgovarajuća terapija. U slučaju da sistem prepozna zdravo tkivo kao bolesno, greška ima manji značaj, jer će se operacijom i daljom dijagnostikom utvrditi da pacijent nije oboleo.

U slučajevima kada je potrebno razlikovati više tipova grešaka rezultat klasifikacije se prikazuje u obliku dvodimenzionalne matrice grešaka, gde svaki red matrice odgovara jednoj klasi i beleži broj primera kojima je to prognozirana klasa, a svaka kolona matrice takođe je obeležena po jednom klasom i beleži broj primera kojima je to stvarna klasa.

Ako posmatramo npr. klasifikacijski problem sa 5 klasa, u kome treba da klasifikujemo emotivna stanja osoba koja se pojavljuju na video snimku u pet različitih emotivnih kategorija: srećan, tužan, besan, nežan i uplašen, onda možemo matricu greške prikazati kao na slici 3.1.

		Stvarna klasa				
		srećan	tužan	besan	nežan	uplašen
Prognozirana klasa	srećan	51	2	1	1	1
	tužan	3	23	1	1	0
	besan	2	2	17	0	0
	nežan	0	1	2	9	1
	uplašen	1	0	1	1	18

Slika 3.1: Ilustracija matrice grešaka za klasifikacijski problem prepoznavanja emotivnih stanja

Po dijagonali matrice nalazi se broj tačno klasifikovanih primera, dok ostali elementi matrice označavaju broj primera koji su neispravno klasifikovani kao neka od preostalih klasa. Iz matrice na slici 3.1. se vidi da su od ukupnog broja primera klase *srećan* pogrešno klasifikovana 6 primera, i to na sledeći način: tri su svrstana u klasu *tužan*, dva u klasu *besan*, nula u klasu *nežan*, i jedan u klasu *uplašen*. Možemo zaključiti da se korišćenjem matrice grešaka omogućava kvalitetnija analiza različitih tipova grešaka.

Iako najveći broj mera za evaluaciju klasifikacijskih modela se odnosi na klasifikacijske probleme sa dve klase, to ne predstavlja posebno ograničenje za upotrebu tih mera, s obzirom da se problemi sa većim brojem klasa mogu prikazati u obliku niza problema sa dve klase. Pri tome svaka od tih mera posebno izdvaja jednu od klasa kao ciljnu klasu, pri čemu se skup podataka deli na pozitivne i negativne primere ciljne klase, sa tim da u negativne spadaju primeri svih ostalih klasa. To je razlog zbog čega u nastavku teksta razmatramo klasifikacijski problem sa dve klase.

Matrice grešaka za klasifikacijski problem sa dve klase prikazane su na slici 3.2. Na osnovu slike može se zaključiti da su moguća četiri različita rezultata prognoze. Stvarno pozitivni i stvarno negativni ishodi predstavljaju ispravnu klasifikaciju, dok lažno pozitivni i lažno negativni ishodi predstavljaju dva moguća tipa greške. Lažno pozitivan primer je negativan primer klase koji je pogrešno klasifikovan kao pozitivan, a lažno negativan je u stvari pozitivan primer klase koji je pogrešno klasifikovan kao negativan. U kontekstu našeg istraživanja, ulazi u matrici grešaka imaju sledeće značenje [Kohavi i Provost, 1998]:

- *a* je broj tačnih predviđanja da je instanca negativna,
- *b* je broj netačnih predviđanja da je instanca pozitivna,
- *c* je broj pogrešnih predviđanja da je instanca negativna,
- *d* je broj tačnih predviđanja da je instanca pozitivna.

		Prognozirano	
		Negativni	Pozitivni
Stvarno	Negativni	*a*	*b*
	Pozitivni	*c*	*d*

Slika 3.2: Matrice grešaka za klasifikacijski problem sa dve klase

Nekoliko standardnih pojmova su definisani za matricu sa dve klase: tačnost, odziv, lažna pozitivna stopa, stvarna negativna stopa, lažno negativna stopa i preciznost. Tačnost je deo predviđanja u ukupnom broju predviđanja koji je tačan. Može se napisati koristeći sledeću jednačinu:

$$Tačnost = \frac{a+d}{a+b+c+d} \tag{3.2}$$

Odziv ili stvarno pozitivna stopa je udeo pozitivnih slučajeva koji su pravilno identifikovani i može se izračunati pomoću jednačine:

$$Odziv = \frac{d}{c+d} \tag{3.3}$$

Lažna pozitivna stopa je udeo negativnih slučajeva koji su pogrešno klasifikovani kao pozitivni, i izračunava se uz pomoć jednačine:

$$\textbf{Lažna pozitivna stopa} = \frac{b}{a+b} \tag{3.4}$$

Stvarna negativna stopa je definisana kao udeo negativih slučajeva koji su klasifikovani ispravno, i izračunava se pomoću jednačine:

$$\textbf{Stvarna negativna stopa} = \frac{a}{a+b} \tag{3.5}$$

Lažna negativna stopa je udeo pozitivnih slučajeva koji su pogrešno klasifikovani kao negativni, i izračunavaju se pomoću jednačine:

$$\textbf{Lažna negativna stopa} = \frac{c}{c+d} \tag{3.6}$$

Konačno, precizniznost je udeo prediktivnih pozitivnih slučajeva koji su tačni, i izračunava se pomoću jednačine:

$$\textbf{Preciznost} = \frac{d}{b+d} \tag{3.7}$$

Postoje slučajevi kada preciznost nije adekvatna mera. Tačnost određena pomoću jednačine (3.2) ne može biti adekvatna mera performanse kada broj negativnih slučajeva je mnogo veći od broja pozitivnih slučajeva [Kubat *et al.*, 1998]. Ako postoje dve klase i jedna je značajno manja od druge, moguće je dobiti visoku preciznost tako što će se sve instance klasifikovati u veću grupu. Pretpostavimo da postoji 1000 slučajeva, od toga 995 negativnih slučajeva i pet koji su pozitivni slučajevi. Ako ih sistem klasifikuje sve negativno, tačnost će biti 99,5%, iako klasifikator propušta sve pozitivne slučajeve. Ili, npr. u testovima koji ustanovljavaju da li je pacijent oboleo od neke bolesti, a tu bolest ima samo 1% ljudi u populaciji, test koji bi uvek prijavljivao da pacijent nema bolest bi imao preciznost od 99%, ali je neupotrebljiv. U ovakvim slučajevima, preciznost kao mera kvaliteta modela nije odgovarajuća, već je bitna mera osetljivost klasifikatora, odnosno njegova mogućnost da primeti instance koje se traže, u ovom slučaju bolesne pacijente.

U mašinskom učenju, većina klasifikatora pretpostavlja jednak značaj klase u smislu broja instanci i nivoa važnosti, odnosno sve klase imaju isti značaj. Standardne tehnike u mašinskom učenju nisu uspešne, kada se predviđaju manjinske klase u neuravnoteženom skupu podataka ili kada se lažno negativni smatraju važnijim od lažno pozitivnih. U stvarnim situacijama, nejednaki troškovi pogrešnih klasifikacija su česti, posebno u medicinskoj dijagnostici, tako da asimetrični troškovi pogrešne klasifikacije moraju biti uzeti u obzir kao važan faktor.

Klasifikatori osetljivi na troškove (eng. *cost-sensitive*) prilagođavaju modele prema troškovima pogrešne klasifikacije u fazi učenja, sa ciljevima kako bi se smanjili troškovi pogrešne klasifikacije umesto maksimiziranja tačnosti klasifikacije. Budući da mnogi praktični problemi klasifikacije imaju različite troškove povezane sa različitim vrstama grešaka, razni algoritmi za ocenu osetljivosti klasifikacija se koriste.

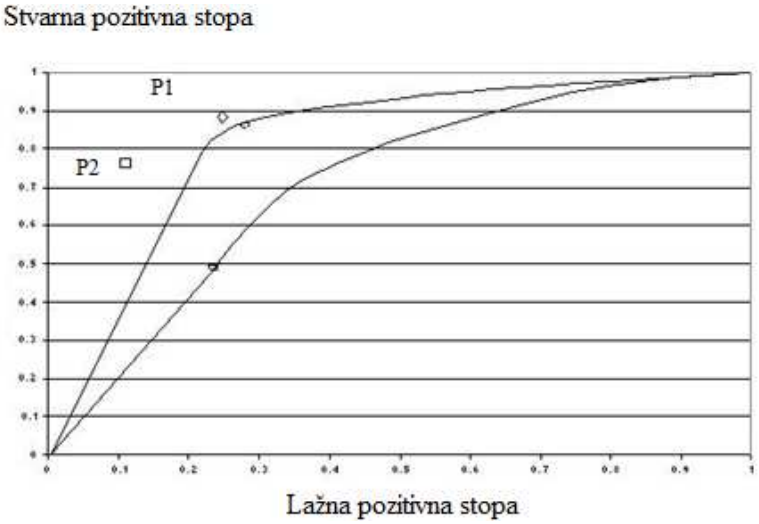
Komplementarnost je jedno od bitnih obeležja evaluacije klasifikacijskih modela. Upotrebom parova mera može se prikazati specifična tačnost klasifikacijskog modela sa donekle suprotstavljenih pozicija. Tako se na primer variranjem parametara odabrane tehnike modeliranja može nauštrb jedne od mera povećati specifična tačnost modela prikazana drugom merom. Ovo je optimizacijski problem u kojem se uz izbor odgovarajućih parametara na osnovu vrednosti jedne mere želi maksimizirati druga mera. U nekim slučajevima je kvalitet klasifikacijskog

modela potrebno izraziti jednim brojem, a ne parom zavisnih mera, što se postiže upotrebom parova mera. Upotrebom parova mera vrednost jedne mere se fiksira i uz taj uslov se posmatra samo druga mera. Tako na primer, može se posmatrati mera preciznosti uz fiksiranu vrednost odziva od 20% i na ovaj način izvedena mera naziva se preciznost na 20%. Ali, češće se koristi usrednjavanje jedne od mera po više fiksiranih vrednosti druge mere, na primer srednja preciznost u tri tačke (po pravilu se radi o vrednostima odziva od 20%, 50% i 80%).

Pored izvedenih mera, postoje i mere koje se ne zasnivaju na fiksiranju jedne komponente para mera, kao npr. F *-mera*, koja se definiše na sledeći način:

$$F - mera = \frac{2 \times odziv \times preciznost}{odziv + preciznost} \tag{3.8}$$

Drugi način za ispitivanje performansi klasifikatora je ROC graf. ROC grafovi su još jedan način osim matrice grešaka za ispitivanje performansi klasifikatora [Swets, 1988]. ROC graf je dvodimenzionalni prikaz koji na X osi predstavlja lažno pozitivnu stopu i na Y osi predstavlja stvarnu pozitivnu stopu. Tačka (0,1) je savršen klasifikator: klasifikuje sve pozitivne i sve negativne slučajeve ispravno. To je (0,1), jer je lažno pozitivna stopa 0 (nula), a stvarna pozitivna stopa je 1 (sve). Tačka (0,0) predstavlja klasifikator koji predviđa sve slučajeve da budu negativni, dok tačka (1,1) odgovara klasifikatoru koji predviđa da svaki slučaj bude pozitivan. Tačka (1,0) je klasifikator koji je netačan za sve klasifikacije. U mnogim slučajevima, klasifikator ima parametar kojim se može podešavati povećanje stvarne pozitivne stope po cenu povećanja lažno pozitivne stope ili smanjenja lažno pozitivne stope po ceni pada vrednosti stvarno pozitivne stope. Svaka postavka parametara daje par vrednosti za lažno pozitivnu stopu i stvarno pozitivnu stopu i niz takvih parova može se koristiti za predstavljanje ROC krive. Neparametarski klasifikator je predstavljen jednom ROC tačkom, kojoj odgovara par vrednosti za lažno pozitivnu stopu i stvarno pozitivnu stopu.



Slika 3.3: Primer ROC grafa, http://www2.cs.uregina.ca/

Slika 3.3. prikazuje primer jednog ROC grafa sa dve ROC krive, kao i dve ROC tačke označene sa P1 i P2. Neparametarski algoritmi proizvode jednu ROC tačku za određeni skup podataka. Osobine ROC grafa su:

- ROC kriva ili tačka je nezavisna od distribucije klase ili troškova grešaka [Kohavi i Provost, 1998].
- ROC graf sadrži sve informacije sadržane i u matrici grešaka [Swets, 1988].
- ROC kriva pruža vizuelni alat za ispitivanje sposobnosti klasifikatora za ispravno prepoznavanje pozitivnih slučajeva i broj negativnih slučajeva koji su pogrešno razvrstani.

Prostor ispod jedne ROC krive može se koristiti kao mera tačnosti u mnogim aplikacijama, i ona se naziva tačnost merenja zasnovana na površini [Swets, 1988]. Provost i Fawcett su 1997. godine [Provost i Fawcett, 1997] tvrdili da korišćenje tačnosti klasifikacije za poređenje klasifikatora nije adekvatna mera, osim ako su troškovi klasifikovanja i raspodele klase potpuno nepoznati, a jedan klasifikator mora biti izabran za svaku situaciju. Oni predlažu metodu procene klasifikatora pomoću ROC grafa i nepreciznih troškova i raspodele klase.

Drugi način upoređivanja ROC tačaka je pomoću jednačine koja izjednačava tačnost sa Euklidskom udaljenosti od savršenog klasifikatora, odnosno od tačke (0,1) na grafu. Na taj način uključujemo težinske faktore koji nam omogućuju da definišemo relativne nepravilne troškove klasifikacije, ako su takvi podaci dostupni.

3.2. Metode za evaluaciju klasifikacijskih modela

Pojam greške, odnosno frekvencije grešaka na nekom skupu primera je u osnovi svih mera za evaluaciju klasifikacijskih modela. Stvarna frekvencija grešaka klasifikacijskog modela je statistički definisana kao frekvencija grešaka na asimptotski velikom broju primera koji konvergiraju stvarnoj populaciji primera.

Takođe, bez obzira na tip posmatranih grešaka, empirijska frekvencija grešaka može se definisati kao odnos broja pogrešno klasifikovanih primera naspram ukupnog broja posmatranih primera. Na osnovu definicije proističe da empirijska frekvencija grešaka nekog klasifikatora bitno zavisi o skupu posmatranih primera. To znači da merenja na različitim skupovima primera rezultiraju različitim vrednostima empirijske frekvencija grešaka.

Zato, kada imamo neograničen broja primera, empirijska frekvencija grešaka teži ka stvarnoj kako se broj posmatranih primera približava beskonačnosti. Ali, u realnim situacijama broj primera je uvek konačan i relativno mali. Zato je osnovni zadatak metoda za evaluaciju klasifikacijskih modela ekstrapolacija empirijske frekvencije grešaka izmerene na konačnom broju primera na stvarnu, asimptotsku vrednost. Za ocenu stvarne frekvencije grešaka klasifikacijskog modela postoji više metoda, koje se razlikuju po pristupu problemu i svojstvima koje pokazuju. U nastavku teksta biće prikazane neke od metoda.

3.2.1. Metoda evaluacije na osnovu testnog skupa primera

Na osnovu skupa primera za učenje izgrađuje se klasifikacijski model. Reklasifikacijska frekvencija grešaka je frekvencija grešaka merena na skupu primera za učenje. Ako postoji neograničen broj primera za učenje koji konvergiraju stvarnoj populaciji primera, onda bi reklasifikacijska frekvencija grešaka izmerena na dovoljno velikom skupu primera za učenje bila vrlo blizu stvarne frekvencije grešaka. U realnim situacijama to nikad nije slučaj, zbog čega dolazi do preteranog prilagođavanja podacima za učenje. Usled preteranog prilagođavanja podacima za učenje rezultirajući model će dobro klasifikovati primere iz skupa podataka za učenje, ali će tačnost klasifikacije novih primera biti značajno manja. To znači da je reklasifikacijska frekvencija grešaka po pravilu znatno manja od stvarne.

Razlika stvarne i reklasifikacijske frekvencije grešaka je dobra mera stepena preteranog prilagođavanja modela podacima za učenje. Sposobnost ispravne klasifikacije primera koji nisu bili uključeni u proces stvaranja modela određuje stvarni kvalitet klasifikatora. Uobičajeno je da se u postupku generisanja modela ne koriste svi dostupni primeri poznate klasifikacije, već se inicijalni skup primera deli na dva dela: (1) skup primera za učenje koji se koristi za generisanje modela, (2) testni skup primera koji služi za evaluaciju rezultirajućeg modela.

Važni zahtevi pri oceni stvarne frekvencije grešaka klasifikacijskog modela je da ova dva skupa budu slučajno odabrana i nezavisna. Pri tome, slučajan izbor podrazumeva da rezultirajući skupovi primera moraju biti slučajni uzorci posmatrane populacije primera. Nezavisnost zabranjuje postojanje bilo kakve

korelacije između ova dva skupa, osim činjenice da potiču iz iste populacije primera.

Ako ovi zahtevi nisu ostvareni, postoji verovatnoća da će procena greške biti netačna, jer uzorak nije reprezentativan, odnosno loše predstavlja stvarne karakteristike populacije. Da bi se osiguralo da se evaluacija modela odvija nad podacima koji nisu korišćeni pri njegovoj izgradnji, vrši se podela inicijalnog skupa primera na skup za učenje i testni skup. S obzirom da se pri korišćenju modela radi sa dotad neviđenim primerima, frekvencija grešaka merena na testnom skupu primera predstavlja procenu stvarne greške klasifikacijskog modela.

Ovaj pristup ne garantuje dobru procenu na svim distribucijama primera, zbog čega je potrebno razmotriti i pitanje pouzdanosti procene frekvencije grešaka korišćenjem testnog skupa. Više metoda se koristi za ocenu stvarne frekvencije grešaka klasifikacijskog modela, a one se razlikuju po pristupu problemu i osobinama koje pokazuju.

Pouzdanost procene značajno zavisi o broju primera u testnom skupu, pri čemu je procena pouzdanija što je testni skup brojniji. Ključan preduslov za pouzdanost procene frekvencije grešaka je dovoljan broj primera u testnom skupu, ali je isto tako bitno da u fazi konstrukcije klasifikatora u skupu primera za učenje bude dovoljan broj primera. Sa premalim brojem primera u skupu za učenje dizajn klasifikacijskog modela ne može biti kvalitetan, zbog čega je uobičajeno da se iz inicijalnog skupa primera veći deo odvoji u skup za učenje. Uobičajeno se 2/3 ukupnog broja primera izdvoji u skup primera za učenje, a 1/3 u testni skup primera.

Faza konstrukcije i faza evaluacije klasifikacijskog modela raspolagat će sa dovoljnim brojem primera, samo ako je inicijalni skup primera dovoljno brojan. Ako inicijalni skup primera nije dovoljno brojan bolji rezultati će se postići korišćenjem metoda evaluacije koje su primerenije takvoj situaciji. Te druge metode se uglavnom zasnivaju na višestrukom ponavljanju postupka procene greške na testnom skupu, uz adekvatnu podelu inicijalnog skupa primera.

3.2.2. Metoda unakrsne validacije

U problemima veštačke inteligencije, nije redak slučaj da je dostupan samo srazmerno mali broj unapred klasifikovanih primera. Često se dešava, pogotovo u oblasti medicinskih istraživanja da se preliminarno ispitivanje sprovodi na vrlo malom broju pacijenata. Proces konstrukcije klasifikacijskog modela, kao i njegova evaluacija su tada posebno otežani, zbog čega je važno da se inicijalni skup klasifikovanih primera što bolje iskoristi i pri konstrukciji i pri evaluaciji modela. Metoda evaluacije korišćenjem testnog skupa u ovom slučaju može biti neprecizna, pogotovo ako se oslanja na jednu, moguće nekarakterističnu particiju skupa primera za učenje, odnosno testiranje modela.

Slučajan izbor je jedan od osnovnih zahteva pri formiranju skupova podataka za učenje i testiranje. Međutim, mogućnost da izabrani skupovi podataka ne predstavljaju reprezentativan uzorak populacije raste kada se ukupan broj raspoloživih primera smanjuje. Zbog toga evaluacija korišćenjem testnog skupa može rezultirati nepreciznom procenom frekvencije grešaka, i to zbog specifičnosti

u skupovima podataka za učenje odnosno testiranje koji nisu svojstvo populacije nego defekt uzrokovan izborom skupova.

Višestrukim ponavljanjem procesa evaluacije na testnom skupu koristeći različite slučajno izabrane skupove za učenje i testiranje, kao i usrednjavanjem dobijenih procena frekvencije grešaka, mogu se izbeći ove anomalije. Na ovom principu se zasniva metoda unakrsne validacije, uz odgovarajuću zamenu skupa podataka za učenje i testnog skupa u svakoj iteraciji [Kohavi, 1995].

Kod metode *k*-struke unakrsne validacije najpre se inicijalni skup primera po načelu slučajnog izbora podeli u *k* međusobno različitih particija približno iste veličine. Postupak je iterativan sa tim da se u jednoj iteraciji *k*–1 particija koristi kao skup za učenje, a konstruisani model se testira na preostaloj particiji koja predstavlja testni skup. Postupak se ponavlja *k* puta, tako da je svaka od particija po jednom u ulozi testnog skupa primera. Ocenu stvarne frekvencije grešaka klasifikacijskog modela predstavlja prosečna frekvencija grešaka svih *k* iteracija postupka.

Kod unakrsne validacije, često se postupak slučajnog izbora modifikuje na način da osigura približno jednaku zastupljenost klasa u svakoj od particija i ovaj postupak se naziva stratifikacija. Postupkom stratifikacije se poboljšava reprezentativnost svake od particija. Stratifikacija ne obezbeđuje reprezentativnost skupa za učenje ili testiranje, iako se ovim načinom postiže da je u svakoj iteraciji zastupljenost klasa u skupu za učenje i testiranje približno jednaka zastupljenosti u inicijalnom skupu primera. Ipak, eksperimentalni rezultati pokazuju da stratifikacija blago poboljšava rezultate evaluacije, posebno na manjim skupovima primera.

Računarska složenost evaluacije klasifikacijskog modela ovom metodom zavisi o broju particija, odnosno iteracija unakrsne validacije, pri čemu svaka iteracija uključuje zasebno konstruisanje i testiranje modela. U praksi se najčešće koristi stratifikovana 5-struka ili 10-struka unakrsna validacija, jer se pokazala kao dovoljno tačna, a nije računarski prezahtevna.

Ova metoda za evaluaciju klasifikacijskih modela ima prednost da su svi dostupni primeri iskorišćeni za testiranje, a i konstrukcija modela se u svakoj iteraciji koristi velikom većinom dostupnih primera. Metoda unakrsne validacije ima umerene računske zahteve u odnosu na neke druge iterativne metode evaluacije, ali ipak primetno veće od klasične evaluacije korišćenjem testnog skupa.

Ipak postoji kod ove metode određena varijabilnost, iako usrednjavanje greške po više testnih skupova u velikoj meri ublažava zavisnost o izboru skupa primera za učenje i testiranje. Uzrok ovome je slučajnost pri formiranju particija na početku procesa evaluacije, zbog čega treba očekivati da će metoda unakrsne validacije uz istu tehniku modeliranja i skup dostupnih podataka, ali uz različito particioniranje tog skupa može dati nešto drugačiju procenu frekvencije grešaka. Ponavljanjem celog postupka unakrsne validacije više puta, kao i usrednjavanjem rezultata može se postići ublažavanje ovog efekta.

3.2.3. Metoda izostavljanja jednog primera

Specijalan slučaj metode unakrsne validacije je metoda evaluacije klasifikacijskih modela izostavljanjem jednog primera. Označimo sa *n* ukupan broj

inicijalno dostupnih primera. Ako u svakoj od *n* iteracija izostavimo jedan primer klasifikacijski model se konstruiše na osnovu *n*–1 primera, a testira na jednom preostalom primeru. Konačna procena frekvencije grešaka u ovom slučaju je usrednjavanje grešaka po svim iteracijama postupka.

Dobre strane metode izostavljanja jednog primera su: (1) maksimalna iskoristljivost inicijalnog skupa primera i (2) postupak je determinističkog karaktera. Prva prednost ove metode je maksimalna iskoristljivost inicijalnog skupa primera što označava da se svaki pojedini primer koristi kao testni primer, a u svakoj iteraciji klasifikacijski model se gradi na maksimalnom mogućem skupu podataka za učenje. Druga prednost ove metode je deterministički karakter postupka, što znači da se svaka particija sastoji od samo jednog primera, čime je izbegnut proces slučajnog uzorkovanja.

Ovo za posledicu ima da evaluacija ovom metodom, uz istu tehniku modeliranja i isti skup dostupnih primera, uvek rezultira istom procenom frekvencije grešaka. Metoda evaluacije izostavljanjem jednog primera postiže posebno dobre rezultate u proceni stvarne frekvencije grešaka i retko kada sistematski odstupa od nje.

Pored ovih prednosti, najveći nedostatak metode izostavljanja jednog primera je velika računarska složenost postupka, budući da se klasifikacijski model konstruiše i testira *n* puta. Ova metoda je dobra za manje skupove primera, dok za veće skupove ova metoda može biti računarski složena i postupak obrade podataka može biti zahtevan.

3.2.4. Bootstrap metoda

Kao i metoda izostavljanja jednog primera, tako i *bootstrap* metoda evaluacije klasifikacijskih modela je namenjena uglavnom problemima sa malim brojem dostupnih primera [Efron i Tibshirani, 1993]. Zbog nedostataka koji postoje kod metode izostavljanja jednog primera, a to se pre svega odnosi na veliku varijansu procene greške na malom broju primera, koja dominira u ukupnoj nepreciznosti ove metode, primenjuje se *bootstrap* metoda evaluacije kako bi se smanjio efekat velike varijanse. Pri formiranju skupa podataka za učenje *bootstrap* metoda se bazira na uzorkovanju sa ponavljanjem. Ova metoda omogućava multipliciranje primera iz inicijalnog skupa, što znači da se u skupu podataka za učenje isti primer iz inicijalnog skupa primera može pojaviti više puta.

Ako sa *n* označimo ukupan broj inicijalno dostupnih primera, kod *bootstrap* metode skup primera za učenje takođe sadrži *n* elemenata, a nastaje slučajnim izborom sa ponavljanjem. Gotovo sigurno u skupu za učenje doći će do ponavljanja nekih primera iz inicijalnog skupa, ali takođe postojaće primeri iz inicijalnog skupa koji uopšte nisu zastupljeni u skupu za učenje, zbog čega će oni formirati testni skup primera. Kod *bootstrap* metode svaka iteracija uključuje ovakvo formiranje skupa primera za učenje i testiranje, kao i konstrukciju i testiranje klasifikacijskog modela. U postupku, broj iteracija nije čvrsto određen, a u praksi se obično radi o nekoliko stotina ponavljanja.

Kod ove metode srednja vrednost greške po svim iteracijama naziva se *e0* procenom greške. Kod umereno velikih skupova podataka *e0* procena daje

pesimističnu procenu stvarne greške, a razlog za ovo je da se klasifikacijski model izgrađuje na osnovu relativno malog broja različitih primera, jer prosečan broj međusobno različitih primera u skupu za učenje iznosi 0.632 ukupnog broja primera. Sa stanovišta verovatnoće možemo objasniti ovako izračunat broj međusobno različitih primera, jer prilikom izbora svakog pojedinog primera u skup za učenje, verovatnoća da određeni primer iz inicijalnog skupa neće biti odabran je $1-\frac{1}{n}$. Kod ove metode u skup za učenje bira se *n* primera, a verovatnoća da određeni primer neće biti izabran u *n* pokušaja dat je sledećim izrazom:

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} \approx 0.368 \qquad (3.9)$$

Aproksimacija data prethodnim izrazom (3.9) vredi za dovoljno veliko *n*. To znači da će testni skup sadržavati 0.368 ukupnog broja primera za dovoljno velike inicijalne skupove primera, dok ostatak od 0.632 ukupnog broja primera daje broj različitih primera u skupu za učenje.

Tabela 3.1. Uporedne karakteristke metoda za ocenu greške klasifikacijskog modela [Ujević, 2004]

	procena na osnovu testnog skupa	*k* unakrsna validacija	metoda izostavljanja jednog primera	*bootstrap* metoda
broj primera u skupu za učenje	*j* (najčešće 2/3 *n*)	*(k-1)n/k*	*n-1*	*n, j* različitih ($j \approx 0.632\ n$)
broj primera u testnom skupu	*n-j* (najčešće 1/3 *n*)	*n/k*	*1*	*n-j* ($\approx 0.368\ n$)
broj iteracija	1	*k*	*n*	nekoliko stotina

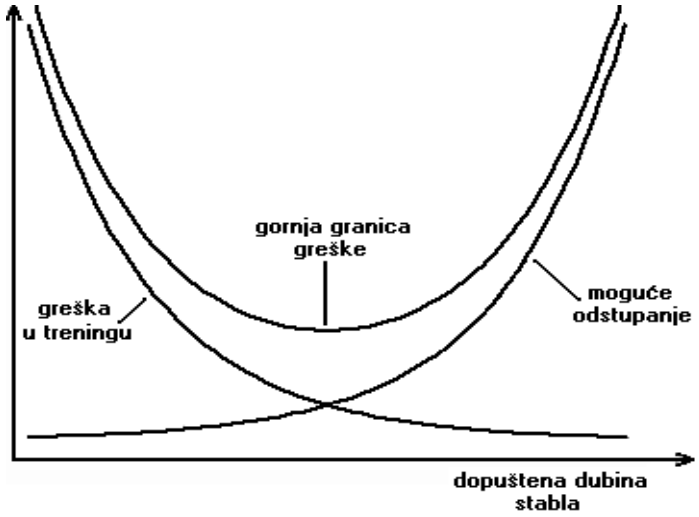
Bootstrap metoda je računski prilično zahtevna zbog većeg broja iteracija pa se uglavnom primenjuje na probleme sa manjim brojem dostupnih primera. I pored toga što *bootstrap* metoda nema problem velike varijanse, ova metoda nije uvek superiorna metodi izostavljanja jednog primera na manjim skupovima primera. Mala vrednost *e0* procene greške je bolji indikator dobrog modela od procene metodom izostavljanja jednog primera. Tabela 3.1. prikazuje uporedne karakteristike spomenutih metoda za ocenu greške klasifikacijskog modela.

3.3. Preterano prilagođavanje modela podacima za trening

Prilikom treninga postoji mogućnost da se model previše prilagodi specifičnostima podataka za trening i da zbog toga daje loše rezultate kada se primeni na drugim podacima. Tako, na primer podaci za trening mogu imati određene karakteristike kao plod slučajnosti, šuma ili predstavljati pristrasan uzorak celog skupa podataka. Dešava se u praksi da je podatke teško sakupiti, zbog čega se obično mora raditi sa podacima koji su raspoloživi bez obzira na njihove manjkavosti.

I pored toga što je prilikom treniranja modela potrebno da se postigne visok nivo preciznosti, potrebno je takođe i paziti da ne dođe do preteranog prilagođavanja podacima. Do preteranog prilagođavanja podacima dolazi usled bogatstva prostora hipoteza, odnosno skupa dopustivih modela. Ako je skup bogatiji onda je lakše naći model koji dobro odgovara podacima. Tako npr., ukoliko se pri učenju dopuštaju samo stabla dubine 1, koja testiraju samo jedan atribut svake instance, jasno je da takva stabla ne mogu lako postići visoku preciznost klasifikacije.

Pored ovih stabala i stabla koja su vrlo duboka i precizno opisuju svaku i najnebitniju specifičnost podataka za trening, u praksi se pokazuju nepouzdanim, pošto širi skupovi podataka ne moraju uvek imati sve specifičnosti skupa podataka za trening. Adekvatna hipoteza bi trebalo da apstrahuje, odnosno zanemari, takve specifičnosti. Međutim, sa povećanjem dozvoljene dubine stabla, povećava se moć učenja, odnosno verovatnoća da će u skupu dopustivih modela biti nađen onaj koji dobro opisuje podatke, zbog čega se smanjuje greška klasifikacije.



Slika 3.4: Greška klasifikacije u zavisnosti od bogatstva skupa dopustivih modela [Janičić i Nikolić, 2010]

Ako stablo dubine, na primer 1 ima visoku preciznost, to znači da je u podacima nađena jaka i vrlo jednostavna zakonitost. Međutim, ako je stablo visoke preciznosti vrlo duboko, to znači da je uočena zakonitost u podacima vrlo kompleksne prirode i stoga može biti specifična samo za podatke u trening skupu, zbog čega očekivano odstupanje greške na širem skupu podataka od greške na trening skupu može biti veliko.

Na slici 3.4. prikazane su tri krive. Prva kriva koja predstavlja ponašanje greške klasifikacije na trening skupu u zavisnosti od dozvoljene dubine stabla je opadajuća. Druga kriva koja predstavlja ponašanje odstupanja greške na širem skupu podataka od greške na trening podacima u zavisnosti od dozvoljene dubine stabla je rastuća. I poslednja treća kriva predstavlja gornju granicu greške klasifikacije u zavisnosti od dozvoljene dubine stabla i ona je zbir prethodne dve. Možemo zaključiti da i premale i prevelike vrednosti za dozvoljenu dubinu stabla vode lošim rezultatima, prve usled nefleksibilnosti dozvoljenih modela, a druge zbog preteranog prilagođavanja modela trening podacima.

Kod npr. stabala odlučivanja, problem preteranog prilagođavanja trening podacima, moguće je rešiti korišćenjem dva pristupa: (1) zaustavljanjem rasta stabla u toku njegove izgradnje i (2) naknadnim odsecanjem. Koristi se najčešće druga mogućnost, pri čemu se odsecanje vrši tako što se iterativno ponavlja u čvorovima u kojima se najviše povećava preciznost klasifikacije na skupu za testiranje sve dok dalje odsecanje ne počne da smanjuje preciznost klasifikacije. Da ne bi došlo do preteranog prilagođavanja trening podacima kod stabala odlučivanja, odsecanje stabla u određenom čvoru predstavlja zamenu celog podstabla čiji je to koren tim čvorom, s tim što mu se dodeljuje oznaka klase u koju se podaci u tom podstablu najčešće klasifikuju.

Isto rezonovanje se može sprovesti i za druge metode mašinskog učenja. Algoritam za učenje dobro generalizuje iz prikazanih primera kada model koji najbolje aproksimira ciljnu funkciju na raspoloživim instancama, takođe najbolje aproksimira ciljnu funkciju na svim mogućim instancama. Uspeh dobre generalizacije leži u adekvatnom upravljanju bogatstvom prostora hipoteza. Tako npr., neki algoritmi učenja poput metode potpornih vektora su konstruisani tako da prilikom izbora modela automatski rešavaju i ovaj problem.

4. PROBLEM KLASIFIKACIJE

U četvrtom delu, razmatra se problem klasifikacije, koji predstavlja razvrstavanja nepoznate instance u jednu od unapred ponuđenih kategorija. U ovom delu biće reči o klasifikacionim algoritmima, koji su kasnije korišćeni u eksperimentalnim istraživanjima za dokaz postavljenih hipoteza. To su sledeći algoritmi nadziranog učenja za izgradnju modela: IBk, *Naïve Bayes*, SVM, J48 stablo odlučivanja i RBF mreža.

4.1. Pojam klasifikacije

Klasifikacija je jedan od najčešćih zadatka mašinskog učenja, i predstavlja problem razvrstavanja nepoznate instance u jednu od unapred ponuđenih kategorija — klasa. U našoj prirodi da stvari oko sebe, kako bih ih bolje shvatili ili organizovali, klasifikujemo i kategorizujemo. Tako npr. klasifikacija se koristi u: dijagnostifikovanju bolesti, prognozi bolesti kod pacijenta, odabiru najbolje terapije za pacijenta od nekoliko mogućih, klasifikaciji kreditinih zahteva klijenata, proceni da li će i koji korisnici kupiti određeni proizvod, izboru ciljne grupe klijenata za marketinške kampanje, analizi slike, analizi glasa za biometrijska potrebe ili za potrebe analize zdravstvenog stanja osobe, prepoznavanju emotivnog stanja osoba na osnovu slike i glasa, dijagnostifikovanju zdravstvenog stanja biljaka ili životinja i slično. Primena klasifikacije je velika i u rešavanju problema u drugim oblastima. Važno zapažanje kod klasifikacije je da je ciljna funkcija u ovom problemu diskretna. U opštem slučaju, oznakama klasa se ne mogu smisleno dodeliti numeričke vrednosti niti uređenje. To znači da je atribut klase, čiju je vrednost potrebno odrediti, kategorički atribut.

Na primeru otkrivanja da li će biti povratka bolesti raka dojke kod žena objasnićemo problem klasifikacije. Predviđanje može da se radi na osnovu sledećih podataka: godina pacijenta, nastupanja menopauze, veličine tumora, veličine čvorova, stepena maligniteta, koja dojka je zahvaćena tumorom, položaja tumora, da li je vršeno zračenje ili ne kod pacijenta i slično. Na osnovu prikupljenih podataka o većem broju pacijenta, pri čemu skup podataka sadrži i podatke kada nema povratka bolesti raka dojke i kada ima povratka bolesti raka dojke, vrši se klasifikacija. Svaka instanca u skupu podataka se odnosi na stanje jednog pacijenta koje je opisano sa odgovarajućim brojem atributa.

Klasifikacija nekog objekta se zasniva na pronalaženju sličnosti sa unapred određenim objektima koji su pripadnici različitih klasa, pri čemu se sličnost dva objekta određuje analizom njihovih karakteristika. Pri klasifikaciji se svaki objekat svrstava u neku od klasa sa određenom tačnošću. Zadatak je da se na osnovu karakteristika objekata čija klasifikacija je unapred poznata, napravi model na

osnovu koga će se vršiti klasifikacija novih objekata. U problemu klasifikacija, broj klasa je unapred poznat i ograničen.

Proces klasifikacije se sastoji iz dve faze, pri čemu se u prvoj fazi gradi model na osnovu karakteristika objekata čija klasifikacija je poznata. Za izgradnju modela se koriste podaci koji se najčešće nalaze u tabelama. Svaka instanca uzima samo jednu vrednost atributa klase, a atribut klase može da ima konačan broj diskretnih vrednosti koje nisu uređene.

Klasifikacioni algoritam uči na osnovu poznatih klasifikacija tj. na osnovu instanci objekata čija klasifikacije je poznata. Pri tome, na osnovu vrednosti njihovih atributa i atributa klase, gradi se skup pravila na osnovu kojih će se kasnije vršiti klasifikacija. Metode klasifikacije su najčešće zasnovane na stablima odlučivanja, Bajesovim klasifikatorima, neuronskim mrežama, itd.

Nakon učenja, model se testira tj. procenjuje se njegova tačnost, pri čemu pod tačnošću podrazumevamo procenat instanci koje su tačno klasifikovane. Vrednost atributa klase svake testne instance poredi se sa vrednošću atributa klase koja je određena na osnovu modela. Važno je napomenuti da se za testiranje modela koriste instance koje nisu korišćene u fazi učenja.

Postoji više načina za izdvajanje testnih instanci, ali se najčešće izdvajaju slučajnim izborom, pre faze učenja, od instanci čija je klasifikacija poznata. Pri tome, ako je tačnost modela zadovoljavajuća onda se dalje koristi u klasifikaciji objekata čija vrednost atributa klase nije poznata.

Postoje sledeći kriterijumi kojima se porede i ocenjuju metode klasifikacije i to su:

- Tačnost, što predstavlja sposobnost klasifikatora da tačno klasifikuje instancu nepoznate vrednosti atributa klase.
- Brzina, što predstavlja broj operacija koje se izvrše pri konstrukciji i primeni klasifikatora.
- Robustnost, što predstavlja preciznost klasifikatora kada se primeni na podacima sa šumom ili podacima kojima nedostaju vrednosti nekih atributa.
- Skalabilnost, što predstavlja efikasnost metode ako se primenjuje na velike količine podataka.
- Interpretabilnost, što predstavlja jasan prikaz i razumevanje rezultata.

Metode rangiranja rangiraju svaki atribut u skupu podataka. Rezultati se potvrđuju korišćenjem različitih algoritama za klasifikaciju. Širok raspon algoritama za klasifikaciju je na raspolaganju, svaki sa svojim prednostima i slabostima. Ne postoji takav algoritam učenja koji najbolje radi sa svim problemima nadziranog učenja. Mašinsko učenje uključuje veliki broj algoritama kao što su:

- veštačke neuronske mreže,
- genetski algoritmi,
- probabilistički modeli,
- indukcijska pravila,
- stabla odlučivanja,

- statističke ili metode raspoznavanje uzoraka,
- *k*-najbliži susedi,
- *Naïve Bayes* klasifikatori i
- diskriminatorna analiza.

U ovom radu korišćeni su sledeći algoritmi nadziranog učenja za izgradnju modela, a to je IBk, *Naïve Bayes*, SVM, J48 stablo odlučivanja i RBF mreža. Prednost IBk je da su oni u mogućnosti da uče brzo sa vrlo malim skupom podataka. Prednost *Naïve Bayes* klasifikatora je da zahteva malu količinu trening podataka za procenu parametara potrebnih za klasifikovanje. Prednost SVM nad drugim metodama je pružanje boljih predviđanja neviđenih test podataka, pružanje jedinstvenih optimalnih rešenja za problem u treniranju i postojanje manje parametara za optimizaciju u poređenju sa drugim metodama. J48 stablo odlučivanja ima razne prednosti: jednostavan za razumevanje i interpretaciju, zahteva malu pripremu podataka, robustan je, dobro radi i sa velikim brojem podataka u kratkom vremenu. RBF mreže nude niz prednosti, uključujući i zahtevanje manje formalnih statističkih treninga, sposobnost da se implicitno detektuju složeni nelinearni odnosi između zavisnih i nezavisnih varijabli, sposobnost detektovanja svih mogućih interakcija između prediktorskih varijabli i dostupnost više algoritama za trening. Ovo poglavlje daje kratak pregled ovih algoritama.

4.2. Metode klasifikacije zasnovane na instancama

U nastavku teksta biće reči o metodama klasifikacije koje su zasnovane na istancama. Biće dat prikaz modela, razmatraće se stabilnost klasifikacije pomoću algoritma *k* najbližih suseda (eng. *n-nearest neighbours*), prednosti i nedostaci ovog algoritma, kao i prikaz pseudo koda.

4.2.1. Prikaz modela klasifikacije zasnovan na instancama

Šema *k* najbližih suseda koristila se još u pedesetim godinama dvadesetog veka [Fix i Hodges, 1951], a kao postupak klasifikacije pojavljuje se desetak godina kasnije [Johns, 1961], a najintenzivnije se koristila na području raspoznavanja uzoraka.

Klasifikacija zasnovana na instancama spada u najjednostavnije tehnike inteligentne analize podataka, jer ne vrši eksplicitnu generalizaciju ciljnog pojma na osnovu svojstava koja su izvodiva iz skupa za učenje, već se svodi na memorisanje skupa za učenje, odnosno pojedinačnih instanci koje sadrži. Osnovni oblik algoritma ne uključuje procesiranje instanci iz skupa za učenje u fazi konstrukcije modela, već samo njihovo memorisanje.

Klasifikacija novih instanci se obavlja prema principu najbližeg suseda, gde se nova instanca upoređuje s memorisanim instancama iz skupa za učenje korišćenjem definisane metrike. Metrika definiše rastojanje instanci na osnovu

vrednosti njihovih atributa, a odgovara intuitivnom shvatanju sličnosti instanci, tako da ako su instance sličnije, rastojanje je manje. Nova instanca se klasifikuje na osnovu pretraživanja skupa za učenje sa ciljem pronalaženja instance koja mu je u smislu rastojanja najbliža. Nova instanca koja se klasifikuje dobija klasu te instance.

Znači, element tehnike klasifikacije zasnovane na instancama koji utiče na oblik modela je metrika, pri čemu je u upotrebi više različitih metrika, a najčešće se koristi *Euklidska*. Ako sa $x = (x_1, x_2, \ldots, x_n)$ označimo vektor vrednosti atributa proizvoljne instance, Euklidska metrika je tada definisana izrazom na sledeći način:

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{4.1}$$

Euklidsko rastojanje instanci x i y se može pretstaviti na sledeći način:

$$d(x,y) = \|x - y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4.2}$$

Na sličan način mogu se definisati i druge metrike variranjem potencije koordinata vektora u definiciji Euklidske metrike. Na primer, izostavljanje kvadriranja uz upotrebu apsolutne vrednosti daje tzv. pravougaonu metriku. Generalno, velike razlike u vrednostima pojedinih koordinata dodatno se naglašavaju višim potencijama, nauštrb koordinata kod kojih je razlika u vrednosti mala.

U napred datim izrazima (4.1) i (4.2) se implicitno pretpostavlja da su svi atributi numeričkog tipa, tj. da su vrednosti koordinata brojevi, a kako bi se definicija Euklidskog rastojanja mogla primeniti na nominalne atribute, potrebno je definisati operaciju razlike nad nominalnim vrednostima.

Ako su sa a_i, $a_j \in Dom(A_i)$ označene dve proizvoljne vrednosti nominalnog

atributa A_i, onda je razlika vrednosti a_i i a_j definisana *0–1 funkcijom razlikovanja* (4.3), na sledeći način:

$$a_i - a_j = \begin{cases} 0, za\ a_i = a_j \\ 1, \quad inače \end{cases} \tag{4.3}$$

Zbog različitih skala merenja za različite numeričke atribute postoji problem vezan za korišćenje različitih metrika. Tako na primer, atribut čiji se raspon vrednosti kreće unutar desetih delova merne jedinice imat će zanemariv uticaj na konačni rezultat u odnosu na atribut sa rasponom vrednosti od nekoliko desetina mernih jedinica. Zato je kod metoda klasifikacije zasnovane na instancama

uobičajen postupak normalizacije svih numeričkih atributa na interval [0,1], korišćenjem funkcije (4.4):

$$f(x) = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.4)$$

gde x_{min} i x_{max} označavaju najmanju, odnosno najveću vrednost posmatranog atributa. Kod metoda klasifikacije zasnovanim na instancama neodređene vrednosti atributa se tretiraju slično nominalnim atributima, tj. proširuje se definicija operacije razlike, tako da se razlika definiše na način da je neodređena vrednost maksimalno udaljena od bilo koje posmatrane vrednosti atributa.

Standardna definicija Euklidskog rastojanja iz izraza (4.2) se može koristiti i za instance s neodređenim vrednostima atributa, ako se koristi ovako definisana funkcija razlike.

4.2.2. Pretraživanje prostora rešenja

Kod klasifikacije zasnovane na instancama ne sprovodi se eksplicitna generalizacija svojstava ciljnog pojma, tj. ne pretražuje se prostor rešenja u potrazi za što boljim modelom. Kod klasifikacije se pojavljuje samo jedan implicitni klasifikacijski model koji je u potpunosti određen skupom za učenje i funkcijom rastojanja. Osnovni algoritam se može nadograđivati tako da se u određenom opsegu modifikuje skup instanci za učenje ili funkcija rastojanja.

Nizom operacija nad modelom se sprovodi modifikacija osnovnog klasifikacijskog modela, pa se može se govoriti o postupku pretraživanja prostora rešenja. Jedna od varijanti klasifikacije zasnovane na instancama nastoji redukovati broj instanci u skupu za učenje, prvenstveno radi smanjenja opsega pretraživanja pri klasifikaciji novih instanci, jer skup instanci za učenje po pravilu sadrži veliki broj redundantnih instanci. Kod problema klasifikacije najvažnije su instance koje se nalaze u blizini granica među klasama, a instance iz unutrašnjosti omeđenog područja klasa mogu se izostaviti bez posledica na tačnost klasifikacije.

Da bi se dobio generalizovan model, postupak formiranja podskupa instanci je iterativan, a sastoji se od uvrštavanja ili eliminisanja instanci prema unapred definisanom kriterijumu. Potrebno je u takvom modelu nastojati da se zadrže reprezentativne instance, koje posredstvom funkcije rastojanja dobro generaliziraju područje u kojem se nalaze, a iz skupa izbaciti instance koji bitno ne pridonose oblikovanju područja odgovarajuće klase. Formiranje reprezentativnog skupa instanci, pretraživanju pristupa po načelu odozdo na gore, tj. od pojedinačnih instanci ka reprezentativnim instancama sa proširenom sferom uticaja. Postoji više različitih kriterijuma prihvatanja odnosno eliminisanja instanci, ali se uglavnom radi o nepovratnim strategijama pretraživanja pohlepnog karaktera. Prema najjednostavnijem kriterijumu prosuđuje se instanca prema rezultatu klasifikacije korišćenjem skupa do tada izdvojenih reprezentativnih instanci. Ako se radi o netačnoj klasifikaciji instance, onda se ona pridodaje u skup

reprezentativnih instanci, jer je evidentno da menja granice klasa. Ako se radi o tačnoj klasifikaciji instance, onda se ona proglašava suvišnom, pod pretpostavkom da je njena informativnost već sadržana u skupu pomoću kojeg je klasifikovana.

Napred dati kriterijum za izbor instanci ima više nedostataka:

- u početnoj fazi procesa pretraživanja postoji nezanemariva verovatnoća odbacivanja instanci koje se mogu pokazati važnim za tačnost klasifikacije rezultirajućeg modela;
- pored ovoga, izabrani podskup reprezentativnih instanci ne zavisi samo o polaznom skupu, već i o redosledu evaluacije instanci;
- i možda najbitniji nedostatak se odnosi na loše ponašanje u uslovima šuma u podacima, jer s obzirom da se u skup uvrštavaju i netačno klasifikovane instance, ovaj kriterijum ima tendenciju akumuliranja instanci sa šumom u rezultirajućem skupu instanci, što dovodi do smanjenja njegove reprezentativnosti.

Zbog svega napred rečenog, u praksi se češće upotrebljavaju drugi, nešto složeniji kriterijumi za izbor reprezentativnih instanci.

4.2.3. Udaljenost instanci

Osim izborom instanci za pamćenje, na oblik klasifikacijskog modela se može uticati i modifikacijom funkcije rastojanja. Jednak uticaj svih atributa u instanci na konačan rezultat je jedno od svojstava Euklidskog rastojanja, ali su u praksi retki problemi kod kojih svi atributi imaju jednaku klasifikacijsku vrednost, čime se stvara mogućnost za poboljšanje tehnike klasifikacije zasnovane na instancama. Modifikacija funkcije rastojanja na način da valorizuje klasifikacijski potencijal različitih atributa je jedno od mogućih rešenja. Način da se ovo postigne je standardno proširenje Euklidskog rastojanja koje podrazumeva uvođenje težinskih vrednosti atributa. Ako sa w_i označimo težinsku vrednost pridruženu atributu A_i, onda modifikovano Euklidsko rastojanje instanci x i y možemo predstaviti na sledeći način:

$$d_w(x, y) = \sqrt{\sum_{i=1}^{n} w_i^2 (x_i - y_i)^2} \tag{4.5}$$

Veći uticaj na proračun rastojanja instanci pruža mu veća težinska vrednost pridružena atributu. Variranje težinskih vrednosti atributa je jedan od načina korekcije klasifikacijskog modela u tehnici klasifikacije zasnovane na instancama. I pored toga što postoje nezavisni postupci ocene klasifikacijske vrednosti atributa, u kontekstu ove tehnike modeliranja težine atributa se najčešće određuju interno, prilikom formiranja relevantnog podskupa instanci za učenje.

Svim atributima je inicijalno pridružena težinska vrednost 1, koja se iterativno modifikuje pri razmatranju svake od instanci iz skupa za učenje. U podskupu relevantnih instanci se pronalazi instanca *y* najbliža posmatranoj instanci *x*, kao i pri klasifikaciji instanci. Ako instance *x* i *y* pripadaju istoj klasi, smanjuje se težinska vrednost atributa čije se vrednosti u instancama *x* i *y* najviše razlikuju, jer se razlika u vrednostima tih atributa pripisuje slabijoj korelaciji sa klasom, kao i što se u slučaju da instance *x* i *y* pripadaju različitim klasama, težinska vrednost atributa sa najvećom razlikom vrednosti povećava. Povećanje, odnosno smanjenje težinske vrednosti proporcionalno je razlici vrednosti atributa u instancama *x* i *y*.

Generalno, postoje i radikalno drugačiji pristupi definisanju funkcije rastojanja, pri čemu je jedan od njih pristup verovatnoće, kod kojeg se definišu operacije transformacije instanci za učenje. Posmatranjem niza operacija pomoću kojih se jedna od instanci može transformisati u drugu, kao i izračunavanje verovatnoće da se takva transformacija dogodi uz slučajan izbor operacija i njihov redosled, može se utvrditi rastojanje dve instance [Cleary i Trigg, 1995]. Posmatranjem svih nizova operacija koje dovode do tražene transformacije instanci, zajedno sa verovatnoćom svake od njih se poboljšava robusnost.

Ako se ovako definiše rastojanje, prednost koju dobijamo je mogućnost uniformnog tretiranja numeričkih i nominalnih atributa, definisanjem odgovarajućih operacija transformacije za svaki od njih. Značajna prednost u nekim primenama, je i da verovatnoća interpretacija rastojanja osim kategoričke klasifikacije može kao rezultat ponuditi i distribuciju verovatnoće pripadanja svakoj od klasa.

4.2.4. Šum u podacima za učenje

Bazični oblik tehnike klasifikacije zasnovan na instancama je prilično podložan problemu šuma u podacima za učenje, a razlog tome je da se klasifikacija nove instance oslanja na samo jednu (najbližu) instancu iz skupa za učenje.

Bitno smanjenje uticaja šuma može se sprovesti proširenjem postupka klasifikacije prema principu *k* najbližih suseda, gde se umesto izdvajanja samo jedne najbliže instance iz skupa za učenje, izdvaja *k* najbližih instanci, za neki unapred određeni mali broj *k*. U klasifikaciji nove instance učestvuje *k* pronađenih instanci, po principu većinskog glasanja, pri čemu se instanci pridružuje najfrekventnija klasa unutar izdvojenih *k* instanci. Specijalni slučaj ovog uopštenja za *k*=1 predstavlja osnovni algoritam klasifikacije.

Generalno, vrednost konstante *k* zavisi o količini šuma u podacima za učenje, pri čemu ako je više šuma, povoljnije je izabrati veće vrednosti konstante *k*. Na ovaj način se postiže poboljšanje tačnosti klasifikacije u uslovima šuma, zbog čega je varijanta *k* najbližih suseda gotovo u potpunosti istisnula osnovni oblik algoritma. Za posmatrani skup instanci, može se dokazati da za $|S|\rightarrow\infty$ i $k\rightarrow\infty$ na način da $k/|S| \rightarrow 0$, verovatnoća pogrešne klasifikacije teži teoretskom minimumu.

Postoji i drugi pristup tretiranju šuma u podacima koji se sastoji od detekcije instanci sa šumom i njihovog izdvajanja iz skupa za učenje, pri čemu je

praćenje klasifikacijskih performansi svake instance iz skupa za učenje uobičajen način detekcije nepoželjnih instanci. Kod ovog pristupa se unapred odrede dva praga tačnosti klasifikacije, u svrhu odbacivanja i prihvatanja instanci u skup instanci za pamćenje. Ako tačnost klasifikacije istance padne ispod praga odbacivanja onda se one eliminišu iz skupa za učenje, a ako tačnost klasifikacije pređe prag prihvatanja te istance se koriste za klasifikaciju. Pri tome se instance čija se tačnost klasifikacije nalazi između dva praga ne koriste pri klasifikaciji, ali se njihove klasifikacijske performanse prate i korigiraju svaki put kada su izdvojene kao najbliže instance prilikom klasifikacije drugih instanci.

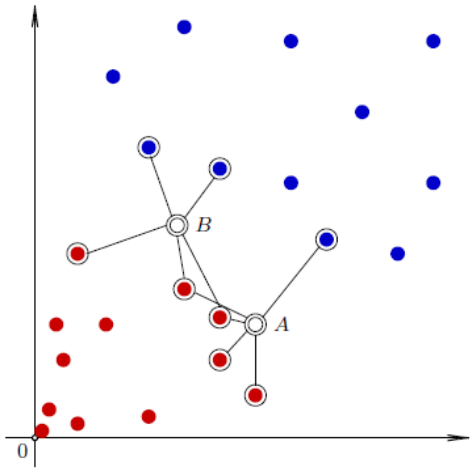
Za određivanje pragova odbacivanja i prihvatanja instanci koristi se pretpostavka *Bernoulli*-jevog procesa, pri čemu tačna klasifikacija odgovara uspešnom ishodu eksperimenta. Ovi pragovi se određuju upoređenjem očekivane frekvencije uspeha pojedine instance i očekivane frekvencije uspeha slepe klasifikacije za odgovarajuću klasu, tj. one klasifikacije koja uvek predviđa spomenutu klasu. Za kriterijum prihvatanja potrebno je da donja granica intervala pouzdanosti posmatrane instance bude viša od gornje granice intervala pouzdanosti slepe klasifikacije, a za kriterijum odbacivanja potrebno je da za posmatranu istancu gornja granica intervala pouzdanosti ocene frekvencije uspeha bude niža od donje granice intervala pouzdanosti slepe klasifikacije. Nije neophodno koristiti isti nivo pouzdanosti za određivanje praga odbacivanja i prihvatanja instanci, tako npr., varijanta algoritma klasifikacije zasnovana na instancama poznata pod nazivom *IB3* (eng. *Instance-Based learner ver. 3*), koristi kriterijum odbacivanja koji je nešto blaži od praga prihvatanja, jer se ne gubi puno odbacivanjem instanci umereno slabih klasifikacijskih performansi – veliki su izgledi da će u kasnijoj fazi procesa biti nadomešteni instancama sa boljim rezultatima klasifikacije.

4.2.5. Stabilnost klasifikacije pomoću algoritma *k* najbližih suseda

Slika 4.1. prikazuje nepoznate instance A i B. Metodom *k* najbližih suseda uz korišćenje Euklidskog rastojanja, instanca A, biva klasifikovana u crvenu klasu za sve vrednosti *k* od 1 do 5. Klasifikacija instance A je postojana zato što se ona nalazi blizu crvenih instanci, a udaljeno od plavih instanci. Za razliku od instance A, klasa instance B može da varira u zavisnosti od broja *k*, i to tako da za *k* = 1 instanca B se klasifikuje u crvenu klasu, za *k* = 2 ne može se odlučiti, za *k* = 3 instanca B se klasifikuje u plavu klasu, za *k* = 4 ponovo nije moguće odlučiti, a za *k* = 5, ona se ponovo klasifikuje u crvenu klasu [Janičić i Nikolić, 2010]. Klasifikacija instance B nije postojana jer se ona nalazi blizu instanci iz obe klase. Iz napred navedenog, možemo zaključiti da je metoda *k* najbližih suseda postojana u unutrašnjosti oblasti koju zauzimaju instance jedne klase, ali je nepostojana na obodu te oblasti.

Nepostojanost klasifikacije osim što može da se demonstrira menjanjem parametra *k*, ona se takođe može analizirati i za fiksiranu vrednost parametra *k*, i to tako da je za manje vrednosti parametra *k* nepostojanost pri variranju vrednosti atributa instance veća nego za veće vrednosti parametra *k*. Empirijski se određuje

vrednost parametra *k*, evaluacijom uspešnosti klasifikacije za različite vrednosti parametra *k*, tako što se bira vrednost *k* za koju je klasifikacija bila najuspešnija. Lokalnost je važno svojstvo metoda zasnovanih na instancama, jer se nepoznata instanca klasifikuje isključivo ili uglavnom na osnovu poznatih instanci koje se nalaze u njenoj blizini. Zbog ovog svojstva, metode klasifikacije pomoću algoritma *k* najbližih suseda doprinose fleksibilnosti modela koje grade.



Slika 4.1: Stabilnost klasifikacije pomoću algoritma *k* najbližih suseda [Janičić i Nikolić, 2010]

4.2.6. Prednosti i nedostaci klasifikacije zasnovane na instancama

Metode klasifikacije zasnovane na instancama ne grade eksplicitan model podataka u vidu neke funkcije kao što to radi većina metoda mašinskog učenja. Zato se klasifikacija ne vrši na osnovu već formulisanog modela, nego na osnovu skupa instanci za trening, tako što instance predviđene za treniranje se čuvaju i bivaju upotrebljene tek kad je potrebno klasifikovati nepoznatu instancu. Na ovaj način se većina izračunavanja premešta iz faze učenja u fazu primene.

Metoda *k* najbližih suseda se zasniva na jednostavnom principu da nepoznatu instancu treba klasifikovati u klasu čije su instance najsličnije nepoznatoj. Koncept sličnosti se najjednostavnije formalizuje preko funkcija rastojanja, pri čemu što je rastojanje između dva objekta veće, to je sličnost između njih manja i obrnuto. Moguće je birati različite funkcije rastojanja, pri čemu je pretpostavka da izabrana funkcija rastojanja relevanta za posmatrani domen i da stvarno oslikava različitost između dva objekta. Pošto se izabere funkcija rastojanja, metoda *k* najbližih suseda se sastoji u nalaženju *k* instanci iz trening skupa koje su najbliže nepoznatoj instanci i njenom klasifikovanju u klasu čiji se elementi najčešće javljaju među pronađenih *k* najbližih suseda.

Osnovni oblik klasifikacije zasnovan na instancama ima više nedostataka:

- Postupak klasifikacije novih instanci može biti spor u slučaju velikih skupova za učenje, budući da klasifikacija svake instance zahteva pretraživanje celog skupa za učenje.
- Bez korišćenja više najbližih instanci pokazuje priličnu osetljivost na šum u podacima za učenje.
- Takođe, nije prilagođen problemima kod kojih atributi imaju različit klasifikacijski potencijal, a klasifikacijske performanse posebno narušavaju irelevantni atributi.

Ovaj tip klasifikacije postaje ponovo popularan početkom 1990-ih kroz radove D. Aha [Aha, 1992], u kojima se nadogradnjama osnovnog postupka umanjuju spomenuti nedostaci, tako što se uvode težinske vrednosti atributa i postupak filtriranja instanci sa šumom, čime se značajno poboljšavaju klasifikacijske sposobnosti ove tehnike, podižući ih na nivo uporediv sa ostalim popularnim tehnikama. Pored ovoga, odbacivanje nepotrebnih instanci značajno redukuje broj instanci koji se pamte, čime se značajno smanjuju potrebni resursi i vreme klasifikacije.

Možemo zaključiti da je najvažnija prednost tehnike klasifikacije zasnovane na instancama u odnosu na stabla odlučivanja i klasifikacijska pravila mogućnost izražavanja proizvoljnih po delovima linearnih granica među klasama, a osnovni nedostatak je činjenica da klasifikacijski model nije izražen eksplicitno, u obliku koji bi bio deskriptivan u terminima domena klasifikacijskog problema.

4.3. Metode *Bayes*-ove klasifikacije zasnovane na verovatnoći

Probabilistički pristup indukciji znanja prikazan u ovom radu dodeljuje verovatnoću klasifikacije instanci u pojedine klase. *Bayes*-ova teorema je osnov ovakvog probabilističkog pristupa.

4.3.1. Osnove *Bayes*-ove teoreme

Bayes-ova teorema omogućuje izbor najverovatnije hipoteze iz skupa hipoteza *H* na osnovu skupa za učenje *D*, a uz uticaj predodređenih verovatnoći svake od ponuđenih hipoteza u skupu *H*. Na slici 4.2. prikazan je Thomas Bayes, koji je živeo i radio u osamnaestom veku.

Slika 4.2: Thomas Bayes (1701 –1761)

Najpre, potrebno je definisati verovatnoće:

- $P(h)$ – početna verovatnoća hipoteze *h*, koja nam omogućava da prikažemo početno znanje o verovatnoćama različitih hipoteza. Ako to znanje ne posedujemo možemo svim hipotezama pridodati jednaku početnu verovatnoću.
- $P(D)$ – početna verovatnoća pojavljivanja instance *D*, koja označava verovatnoću pojavljivanja *D* bez obzira na to koja je hipoteza ispravna.
- $P(D|h)$ – uslova verovatnoća pojavljivanja *D* uz uslov ispravnosti hipoteze.
- $P(h|D)$ – uslovna verovatnoća ispravnosti hipoteze *h* nakon pojavljivanja instance *D*, i ona je zanimljiva sa stanovišta indukcije znanja jer omogućava procenu ispravnosti hipoteza nakon posmatranja pojave novih instanci *D*.

Ova teorema nam omogućava da izračunamo $P(h|D)$ preko izraza:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{4.6}$$

U mnogim problemima potrebno je naći *maximum a posteriori* (*MAP*) hipotezu, odnosno najverovatniju hipotezu *h* iz *H* uz uslov pojavljivanja *D*. Primjenjujući *Bayes*-ov teorem na svaku hipotezu *h* iz skupa *H* i zatim birajući najverovatniju, lako izračunavamo MAP hipotezu:

$$h_{MAP} = \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} = \arg\max_{h \in H} P(D|h)P(h) \tag{4.7}$$

U izrazu 4.7. isključili smo $P(D)$ jer ta verovatnoća predstavlja konstantu nezavisnu od *h*. Ukoliko pretpostavimo da su sve hipoteze iz skupa *H* jednako verovatne, tada možemo zanemariti uticaj parametra $P(h)$ i tada procenjujemo h_{MAP} samo na osnovu $P(D|h)$. Hipoteza koja maksimizira $P(D|h)$ nazivamo ML (eng. *maximum likelihood*) hipoteza:

$$h_{ML} = \arg\max_{h \in H} P(D|h) \tag{4.8}$$

U klasifikacionim problemima se dosta koristi *Bayes*-ova teorema. U ovom radu korišćen je metod klasifikacije pod nazivom *Naïve Bayes* klasifikator.

4.3.2. Naïve Bayes klasifikator

Ako klasifikaciju predstavimo kao pronalaženje najverovatnije klasifikacije v_{MAP} tada se ona može izračunati na sledeći način:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n) \tag{4.9}$$

U izrazu 4.9. je dat najverovatniji element konačnog skupa V svih mogućih klasifikacija ulazne instance. Ako svaku instancu prikažemo kao skup vrednosti atributa, i ako je poznat skup trening instanci koji je definisan takođe istim skupom atributa, onda prethodni izraz možemo pisati kao:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n) \tag{4.10}$$

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2, \dots, a_n | v_j) P(v_j)}{P(a_1, a_2, \dots, a_n)} \tag{4.11}$$

$$v_{MAP} = \arg\max_{v_j \in V} P(a_1, a_2, \dots, a_n | v_j) P(v_j) \tag{4.12}$$

Na osnovu podataka za treniranje računamo vrednost izraza. Problem sa izračunavanjem izraza $P(a_1, a_2, \dots, a_n | v_j)$ proizilazi iz međusobne zavisnosti vrednosti atributa tako da je broj mogućih izraza jednak broju svih mogućih različitih *n*-torki $\{a_1, a_2, \dots, a_n\}$ pomnoženih sa brojem svih mogućih klasifikacija. Svaki primer je potrebno videti u ulaznom skupu mnogo puta kako bi mogli pouzdano oceniti tražene verovatnoće.

Ovaj klasifikator uvodi pojednostavljenje u vidu pretpostavljene međusobne nezavisnosti vrednosti atributa u *n*-torkama $\{a_1, a_2, \dots, a_n\}$ tako da se izraz može napisati kao:

$$P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j) \tag{4.13}$$

Možemo napisati izraz za klasifikaciju *Naïve Bayes* klasifikatorom na sledeći način:

$$v_{NBj} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \tag{4.14}$$

U ovom slučaju, broj različitih verovatnoća koje treba izračunati iz podataka za trening iznosi broj različitih vrednosti atributa pomnoženo sa brojem različitih mogućih klasifikacija što predstavlja mnogo manji broj nego broj potreban kako bi se dobila verovatnoća $P(a_1, a_2, \dots, a_n | v_j)$.

U realnim situacijama, uslov nezavisnosti koji je pretpostavljen relativno je strog i može predstavljati problem, ali u praktičnoj upotrebi *Naïve Bayes* klasifikator je pokazao korisnost zbog jednostavnosti implementacije i zadovoljavajućih rezultata. Ako su zaista svi posmatrani atributi nezavisni, *Naïve Bayes* klasifikacija v_{NBj} je jednaka klasifikaciji v_{MAP}.

Kao primer korišćenja *Naïve Bayes* klasifikatora možemo posmatrati slučaj odlaska u šetnju:

Ako Vreme.Sunčano i Temperatura.Hladno i Vetrovito.Da i Vlažnost.Visoka, tada Otići u šetnju.DA (0.1) *i Otići u šetnju.NE* (0.9).

Probabilistički pristup u otkrivanju znanja dodeljuje verovatnoću klasifikacije instanci u pojedine klase, gde je 0.1 verovatnoća za *Otići u šetnju.DA* 0.1, a 0.9 za *Otići u šetnju.NE*. U ovom slučaju se radi o binarnoj klasifikaciji, gde je suma verovatnoća za *Otići u šetnju.DA* i za *Otići u šetnju.NE* jednaka 1. Verovatnoća se određuje frenkvencijskom interpretacijom i posmatranjem svakog atributa nezavisno, što predstavlja pretpostavku „naivnosti“.

4.3.3. Prednosti i nedostaci klasifikacije zasnovane na *Naïve Bayes* klasifikatoru

Naïve Bayes klasifikator predstavlja veoma brz klasifikator, pogodan za klasifikaciju jer ima male zahteve što se tiče upotrebe memorije. On je jednostavna statistička šema učenja i vrlo se često koristi u klasifikacionim problemima, a nekada je uspešniji i od mnogih složenijih pristupa. Robustan je za nerelevantne podatke, jer će se oni međusobno poništavati, a takođe, dobro se pokazao i u domenima, gde postoji veliki broj podjednako relevantnih podataka. Ovaj klasifikator je optimalan ukoliko je tačna predpostavka nezavisnosti podataka. Možemo zaključiti da *Naïve Bayes* klasifikator ima sledeće osobine:

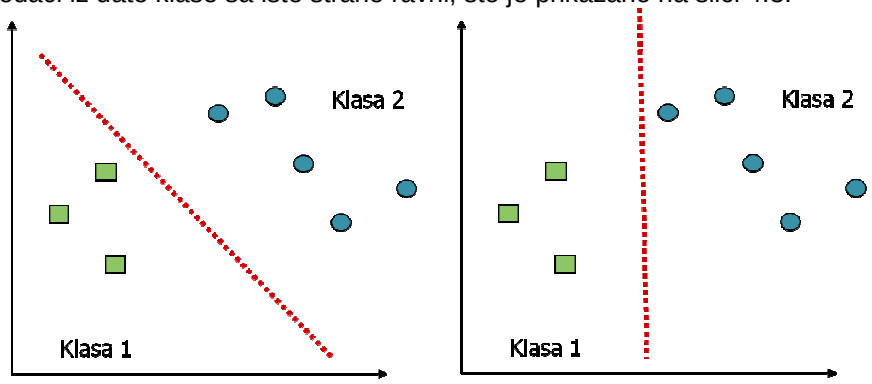
- mali zahtevi za memorijom,
- brzi trening i brzo učenje,
- jednostavnost,
- često radi iznenađujuće dobro.

4.4. Metoda potpornih vektora

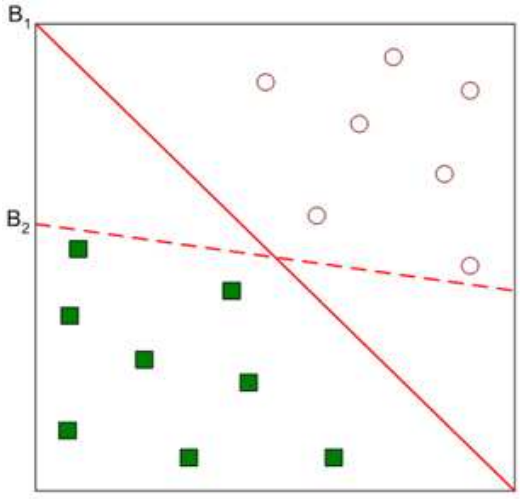
Metoda potpornih vektora (eng. *Support Vector Machine* - SVM) je binarni klasifikator koji konstrukcijom hiper-ravni u visoko-dimenzionalnom prostoru stvara model koji predviđa kojoj od dve klase pripada nova instanca. Ova metoda je razvijena od strane *Vapnik*-a i saradnika 1995. godine i uživa veliku popularnost zbog veoma dobrih rezultata koji se dobijaju.

4.4.1. Osnovne postavke

U mašinskom učenju, metoda potpornih vektora je popularna zbog svojih dobrih performansi. Kao nadzirana metoda koja analizira podatke i prepoznaje obrasce, ona je strogo utemeljena na statističkim teorijama učenja i istovremeno smanjuje trening i test greške. Osnovna ideja ove metode je da se u vektorskom prostoru u kome su podaci predstavljeni, nađe razdvajajuća hiper-ravan tako da su svi podaci iz date klase sa iste strane ravni, što je prikazano na slici 4.3.



Slika 4.3: Zadatak faze treninga: naći optimalnu ravan koja razdvaja podatke za trening, *poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt*

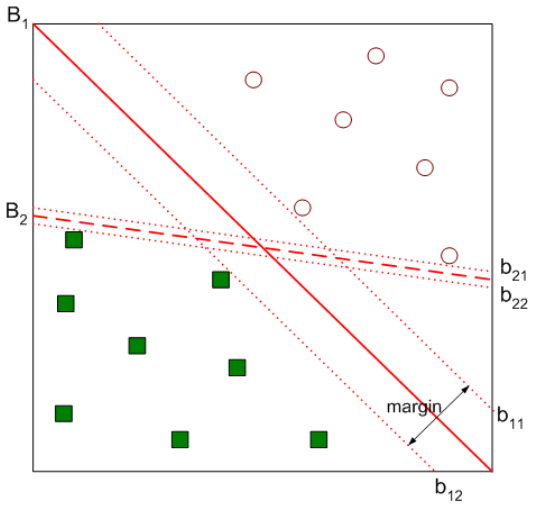


Slika 4.4: Koje rešenje je bolje B_1 ili B_2 i kako definisati „bolje“ rešenje?, *poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt*

Kod korišćenja ove metode, postavlja se pitanje koje je rešenje bolje i na koji način definisati „bolje“ rešenje, što je prikazano na slici 4.4.

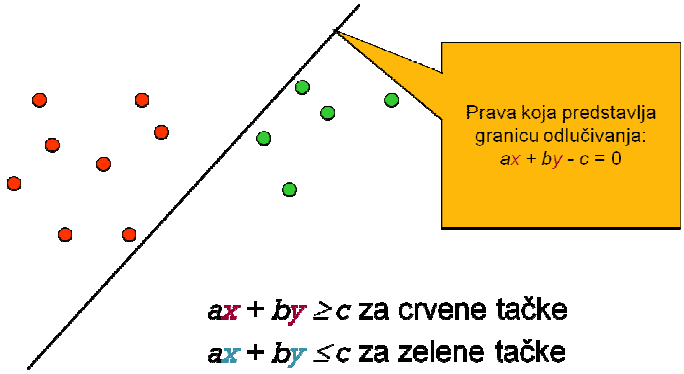
Ako pretpostavimo da su podaci linerano razdvojivi, u fazi treniranja treba naći optimalnu razdvajajuću hiper-ravan, odnosno ravan sa maksimalnom

„marginom“ (što pretstavlja rastojanje od trenirajućih podataka). U tom slučaju nađena hiper-ravan (tj. njena jednačina) je model (slika 4.5). Potom, na osnovu modela izračunavamo rastojanje od hiper-ravni i na osnovu toga određujemo klasu (iznad/ispod ravni).



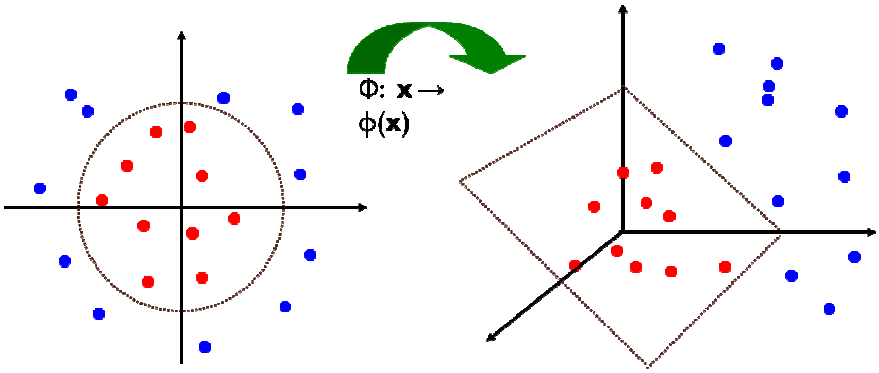
Slika 4.5: Naći hiper-ravan koja maksimizuje veličinu margine→ B_1 je bolje od B_2 , *poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt*

Na slici 4.6. prikazan je linearni klasifikator SVM, kod koga prava $ax + by - c = 0$ predstavlja granicu odlučivanja.



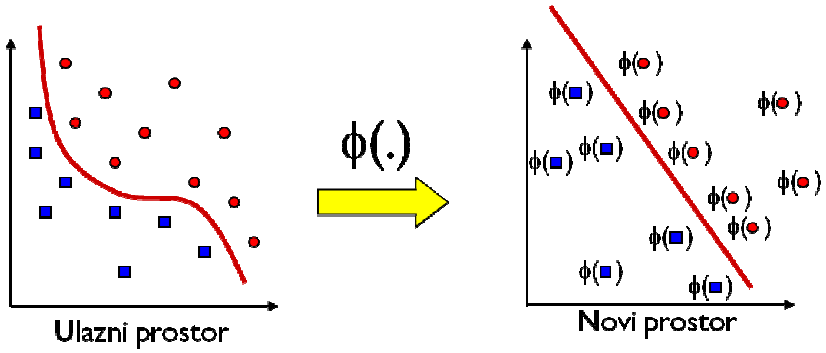
Slika 4.6: Linearni klasifikator SVM,
poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt

SVM određuje optimalno rešenje koje maksimizuje razdaljinu između hiper-ravni i tačaka koje su blizu potencijalne linije razdvajanja i predstavlja intuitivno rešenje: ako nema tačaka blizu linije razdvajanja, onda će klasifikacija biti relativno laka.



Slika 4.7: Preslikavanje u više-dimenzioni prostor u kome je skup podataka za trening linearno razdvojiv,
poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt

U slučaju linearno ne-razdvajajućih problema, koristimo nelinearni SVM, pri čemu je osnovna ideja da se osnovni (ulazni) vektorski prostor preslika u neki više-dimenzioni prostor u kome je skup podataka za trening linearno razdvojiv. Na slici 4.7. prikazano je preslikavanje u više-dimenzioni prostor u kome je skup podataka za trening linearno razdvojiv.



Slika 4.8: Nelinearni SVM,
poincare.matf.bg.ac.rs/~nenad/ip.2013/6.SVM_klasifikacija.ppt

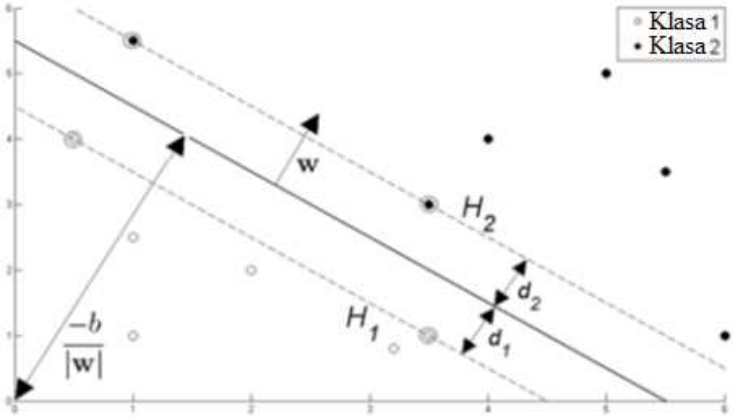
SVM konstruiše hiper-ravan ili skup hiper-ravni u visokom dimenzionalnom prostoru, koji se može koristiti za klasifikaciju, regresiju, ili druge probleme. Mnoge hiper-ravni mogu služiti za klasifikovanje podataka, najbolja hiper-ravan je ona koja predstavlja najveće razdvajanje, ili marginu između dve klase. Generalno govoreći, kada je veća margina onda je manja greška generalizacije klasifikatora. Izabrana je hiper-ravan sa maksimalnom marginom, za koji važi da je rastojanje od nje do najbliže tačke podataka na svakoj strani maksimalna. Na slici 4.8. prikazan je nelinearni SVM.

4.4.2. Linearno odvojive klase

Ako u skupu za učenje imamo *L* vektora, odnosno tačaka u *D*-dimenzionalnom prostoru, gde svaki uzorak x_i ima *D* atributa, odnosno komponenti vektora i pripada jednoj od dve klase $y_i = -1 \; ili \; 1$, tada oblik jednog ulaznog podatka možemo prikazati izrazom:

$\{x_i, y_i\}$ gde je $i = 1 \dots L$, $y_i \in \{-1,1\}\; x \in R^D$, (4.15)

gde se pretpostavlja da su podaci linearno odvojivi, što znači da možemo nacrtati pravac u koordinatnom sistemu sa osama x_1 i x_2 za slučaj *D* = 2, odnosno hiper-ravan za slučaj $D \geq 2$. Izrazom $w \cdot x + b = 0$ možemo opisati hiper-ravan pri čemu je *w* normala hiper-ravni i $\frac{b}{\|w\|}$ vertikalna udaljenost hiper-ravni od ishodišta koordinatnog sistema. Uzorci najbliži razdvajajućoj hiper-ravni su potporni vektori i zato se najteže klasifikuju. Cilj metoda potpornih vektora jeste da izabere hiper-ravan maksimalno udaljenu od najbližih uzoraka obe klase. Na slici 4.9. je dat grafički prikaz dve linearno odvojive klase.



Slika 4.9: Prikaz dve linearno odvojive klase [Fletcher, 2009]

Na ovakav način se implementacija metode potpornih vektora svodi na izbor parametara *w* i *b*, takvih da ulazne podatke možemo opisati sledećim izrazima:

$$x_i \cdot w + b \geq +1 \; za \; y_i = +1 \quad (4.16)$$

$$x_i \cdot w + b \leq -1 \; za \; y_i = -1 \quad (4.17)$$

Kombinovanjem dva prethodna izraza dobijamo:

$$y_i(x_i \cdot w + b) - 1 \geq 0, \forall i \quad (4.18)$$

Ravni H_1 i H_2 na kojima leže potporni vektori možemo prikazati sledećim izrazima:

$$x_i \cdot w + b = +1 \; za \; H_1 \tag{4.19}$$
$$x_i \cdot w + b = -1 \; za \; H_2 \tag{4.20}$$

Ako definišemo vrednosti d_1 i d_2 kao rastojanje od H_1 i H_2 do hiper-ravni, ekvidistantnost hiper-ravni od H_1 i H_2 podrazumeva $d_1 = d_2 = \frac{1}{\|w\|}$, pri čemu vrednost d_1, **odnosno** d_2 nazivamo marginom. Da bi izabrali hiper-ravan maksimalno udaljenu od potpornih vektora, potrebno je maksimizirati marginu, što je ekvivalentno pronalaženju:

$$min\|w\| \; takav \; da \; [y_i(x]_i \cdot w + b) - 1 \; za \geq 0, \forall i \tag{4.21}$$

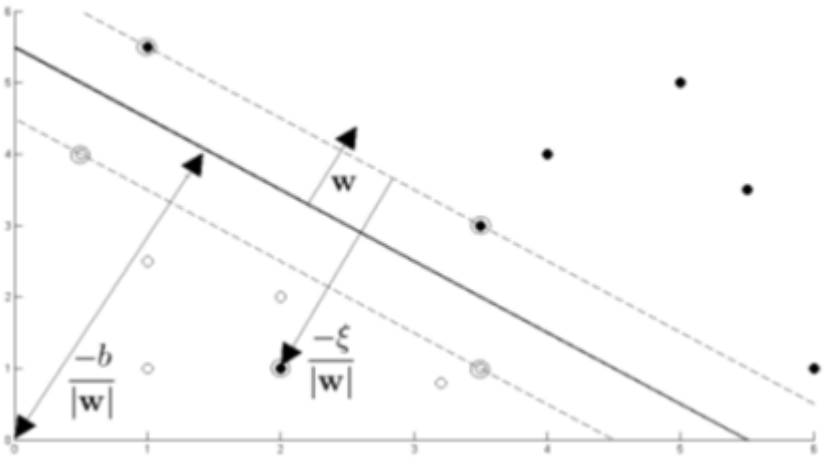
4.4.3. Linearno neodvojive klase

Da bi metode sa potpornim vektorima koristili i za na linearno neodvojive klase, potrebno je ublažiti uslove (4.16) i (4.17) uvođenjem nenegativne vrednosti ξ_i:

$$x_i \cdot w + b \geq +1 - \xi_i \; za \; y_i = +1 \tag{4.22}$$
$$x_i \cdot w + b \leq -1 + \xi_i \; za \; y_i = -1 \tag{4.23}$$

Kombinovanjem prethodna dva izraza dobijamo sledeći izraz:

$$[y_i(x]_i \cdot w + b) - 1 + \xi_i \geq 0, \xi_i \geq 0, \forall i \tag{4.24}$$



Slika 4.10: Prikaz dve linearno neodvojive klase [Fletcher, 2009]

Primenjena metoda se naziva metoda meke margine (eng. *soft margin method* [Fletcher, 2009]), a izvorno je nastala sa idejom dozvoljavanja pogrešnog označavanja klasa pre samog postupka učenja. Slika 4.10. prikazuje hiper-ravan

kroz dve linearno neodvojive klase, gde je vidljiv i uzorak sa pogrešne strane hiper-ravni zbog kojeg prostor nije linearno odvojiv. Mera rastojanja tog uzorka od pripadajućeg potpornog vektora je ξ .

Izbor razdvajajuće hiper-ravni svodi se na pronalaženje:

$$\frac{\min 1}{2}\|w\|^2 + C\sum_{i=1}^{L}\xi_i \text{ takav da } [y_i(x]_i \cdot w + b) - 1 + \xi_i \geq 0, \forall i \tag{4.25}$$

gde vrednost *C* predstavlja faktor greške, kojim dozvoljavamo određene greške pri treniranju, bez čega pronalazak hiper-ravni ne bi bio moguć.

Problem se može rešiti upotrebom metode Lagranžovih koeficijenata, što se može pokazati korisnim kod nelinearnih jezgara, i tada se dobija efikasan iterativan algoritam LSVM. Primer veoma jednostavne i efikasne implementacije je SMO (eng. *Sequential Minimal Optimization*), koji koristi razbijanje na najmanji podproblem gde se onda lako određuje vrednost jednog po jednog preostalog koeficijenta [Platt, 1999] umesto skupog numeričkog rešavanja problema kvadratnog programiranja. U nastavku rada prikazaćemo pseudo kod za SMO koji smo u radu koristili.

Osim SMO, poznata je i SVMLight implementacija koju koristi program za etiketiranje SVMTool, kao i SVR (eng. Support Vector Regression) koji koristi model takve funkcije koji koristi samo deo skupa primera a ostale ignoriše.

U eksploataciji uz date pretpostavke SMV je izuzetno efikasna metoda, i nije metoda učenja instancama u osnovnom obliku - međutim, postoje implementacije (SVMHeavy, [http://www.ee.unimelb.edu.au/staff/apsh/svm/]) koje podržavaju i ,,lenjo“ učenje. Inkrementalno, odnosno lenjo učenje, nasuprot radoznalim metodama (eng. *eager*, ili *batch learning*) je poželjna osobina učenja ako postoji zahtev za stalnim menjanjem baze znanja, gde svaka takva promena ne povlači ponavljanje celog postupka učenja već samo efikasno inkrementalno dodavanje znanja.

Iako obuka kod SVM može biti zahtevnija za veliki broj primera i klasa, ona je u suštini linearno kompleksna *O*(*nm*) (gde je *m* dimenzija prostora) za razliku od ostalih sličnih poznatih metoda mašinskog učenja koje mahom eksponencijalno zavise od *m*.

4.4.4. Kernel funkcija

Opisani linearni klasifikator se naziva klasifikator optimalne granice (eng. *maximum margin classifier*). Metoda potpornih vektora je uopšteni klasifikator optimalne granice za nelinearnu klasifikaciju, što se postiže postupkom poznatim pod nazivom Kernel trik (eng. *Kernel trick*) [Fletcher, 2009]. Osnovna ideja je da se u izrazu (4.25) zameni ulazni vektor x_i sa funkcijom $\phi(x_i)$, koja ulazni vektor preslikava iz *n*-dimenzionalnog u *m*-dimenzionalni prostor, uz $m \gg n$, kako bi dobili uzorke koji su linearno odvojivi. Računanje unutarnjeg produkta vektora $\phi(x_i)$ i *w* predstavlja problem jer je nova dimenzionalnost puno veća, ponekad i

beskonačna. Zato se koristi Kernel funkcija (eng. *Kernel function*) $K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)$, pomoću koje je moguće izvršiti izračunavanje na mnogo jednostavniji način.

Da bi metode potpornih vektora vršile dobro klasifikaciju potrebno je dobro odabrati parametare Kernel funkcije i ranije pomenuti parametar *C* – faktor greške.

Postoji teorija o tome kako kontruisati ispravan kernel za dati problem. Obično se koriste uobičajeni kerneli ili njihova kombinacija. Matematička teorija (teorema *Mercer*-a) definiše uslove koje data funkcija treba da zadovolji da bi predstavljala sklarni proizvod u nekom vektorskom prostoru: simetričnost, pozitivna definitnost, itd. Svaka funkcija koja zadovolji te uslove može da bude korišćena kao kernel. Navedeni primeri kernel funkcija su dovoljni u većini primena (naročito ako se uzme u obzir zatvorenost).

Izbor odgovarajućeg kernela za određenu primenu je često težak zadatak. Nužan i dovoljan uslov za kernel da bi bio valjan je da mora zadovoljiti *Mercer* teoremu, ali osim toga, ne postoji matematički strukturirani pristup koji kaže koji kernel treba koristiti. Naravno, očekuje se da nelinearni kernel koji se koristi u C-SVC bude bolji nego linearni kernel, ako je poznato da su podaci ne linearno odvojivi. Izbor kernela rezultira u različitim vrstama C-SVC sa različitim nivoima uspešnosti.

Koristi se nekoliko standardnih oblika kernel funkcija:

- linearna - $K(x_i, x_j) = x_i^T \cdot x_j$,
- polinomna - $K(x_i, x_j) = \left((x)_i^T \cdot x_j + a\right)^b$,
- Gausova (eng. *Radial Basis Function* - RBF) - $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0$,
- racionalna kvadratna - $K(x_i, x_j) = 1 - \frac{\|x_i - x_j\|^2}{\|x_i - x_j\|^2 + \tau}$,
- sigmoidalna - $K(x_i, x_j) = \tanh(kx_i^T \cdot x_j + a)$.

Linearni kernel je najjednostavniji kernel. Pokazuje dobre performanse za linearno odvojive podatake, ali začudo, radi jako dobro čak i u slučajevima nelinearnih podataka. Budući da je isti princip i za linearni i polinomni kernel, a transformacija dovodi do različitog prostora rešenja, perfomansa polinomnog kernela se očekuje da bude otprilike ista kao linearnog kernela. Očekuje se da RBF kernel bude puno bolji od linearnog ili polinomnog kernela, ali ovaj kernel je teško dizajnirati, jer je teško doći do optimalnog γ i odabira odgovarajućeg *C* koji radi najbolje za zadati problem. RBF kernel je često prvi izbor u poređenju sa polinomnim kernelom, jer broj hiperparametara utiče na složenost odabira modela, a polinomni kernel ima više hiperparametara od RBF kernela. RBF kernel ima najširu primenu, koja odgovara preslikavanju u beskonačno-dimenzionalni prostor. Sve što je linearno razdvojivo u početnom prostoru karakteristika, razdvojivo je i u prostoru određenim ovom funkcijom. Širinu „zvona“ *Gaussove* krive određuje

parametar γ . Za klasifikaciju, sigmoidni kernel nije tako efikasan kao što su ostala tri kernela. Sigmoidni kernel ne mora nužno biti pozitivno definisan, a parametri *γ* i *r* moraju biti ispravno odabrani.

Za potrebe ovog istraživanja se koristio RBF kernel.

4.4.5. Klasifikacija u slučaju postojanja više klasa

Metoda potpornih vektora je binarni klasifikator, što znači da razvrstava neki nepoznati uzorak u jednu od dve klase. Ako je potrebno izvršiti klasifikaciju uzoraka u više od dve klase, ovaj problem ne možemo rešiti samo jednim klasifikatorom. Problem se rešava na sledeća dva načina:

- Prvi način je „jedan-protiv-svih“ (eng. *one-versuss-all*). Konstruisanjem *n* binarnih klasifikatora od kojih svaki razvrstava uzorke ili u jednu od klasa ili u preostalih *n*-1 klasa. Novi uzorci se klasifikuju korišćenjem strategije „pobednik-odnosi-sve“ (eng. *winner-takes-all*), što znači da svaki klasifikator, osim izlaza, daje i meru sigurnosti u svoj izbor. Od svih klasifikatora čiji izbor nije „all“ uzima se izbor onoga koji je najsigurniji u svoj izbor, a ako svi klasifikatori odaberu „all“, verovatno se radi o nepostojećoj klasi ili uzorku kojeg nije moguće klasifikovati. U slučaju da svi klasifikatori odaberu „all“, kao izbor se najčešće uzima izbor suprotan onom klasifikatoru koji je odabrao „all“ sa najmanjom sigurnošću.
- Drugi način je „jedan-protiv-jednog“ (eng. *one-versus-one*), pri čemu se konstruiše $\frac{n(n-1)}{2}$ binarnih klasifikatora od kojih svaki svrstava uzorke u jednu od dve klase. Postupkom glasanja se vrši klasifikacija novih uzoraka, pri čemu se svaka binarna klasifikacija smatra jednim glasanjem za jednu od dve klase, čime se broj glasova za klasu koja je odabrana pri toj binarnoj klasifikaciji uvećava za jedan. Kada se izvede svih $\frac{n(n-1)}{2}$ postupaka glasanja, uzorku se dodeljuje klasa sa najviše postignutih glasova, a u slučaju da dve klase imaju jednak broj glasova, najčešće se vrši izbor klase sa manjim indeksom.

4.4.6. Klasifikacija pomoću biblioteke libSVM

Biblioteka libSVM [Chang i Lin, 2001] (eng. *A Library for Support Vector Machines*) sadrži podršku za klasifikaciju uzoraka metodom potpornih vektora, uz niz dodatnih alata koji olakšavaju pripremu ulaznih podataka i izbor ispravnih parametara. Biblioteka libSVM je implementirana u programskim jezicima *C++*, *Java*, *Python* i *Matlab*. U ovom radu je korišćena implementacija u *Java* programskom jeziku.

Najpre je potrebno sprovesti postupak skaliranja ulaznih podataka na raspon [-1, 1], a nakon toga potrebno je izabrati optimalne parametre *C* i γ za RBF funkciju. Ova biblioteka nudi alat za izbor optimalnih parametara postupkom unakrsne validacije u skripti *grid*. Postupak unakrsne validacije obavlja se tako da se skup ulaznih podataka za učenje podeli u *n* podskupova, i tada se svaki od *n* podskupova testira korištenjem SVM-a naučenog na preostalih (*n-1*) podskupova. Vrednost parametara *C* i γ se eksponencijalno povećava i svaki put se izvodi unakrsna validacija, kako bi se pronašli najbolji parametri. Nakon pronalaska optimalnih parametara, prelazi se na detaljniju unakrsnu validaciju oko dobijenih parametara kako bi se dodatno povećala tačnost. Prethodno skalirani skup ulaznih vrednosti prima skripta *grid* i crta graf uspešnosti unakrsne validacije sa različitim parametrima.

4.5. Stabla odlučivanja

Primene metode stabla odlučivanja uključuje rešavanje problema poput nivoa kompleksnosti stabla, tretman kontinuiranih atributa, tretman atributa s neodređenim vrednostima, poboljšanja efikasnosti algoritma i sl. U daljem tekstu detaljnije ćemo se pozabaviti ovim problemima.

4.5.1. Predstavljanje modela

Učenje stabala odlučivanja je metoda aproksimacije diskretnih ciljnih funkcija u kome se naučena funkcija predstavlja u vidu stabla, gde svakom čvoru stabla odgovara test nekog atributa instance, grane koje izlaze iz čvora različitim vrednostima tog atributa, a listovima odgovaraju vrednosti ciljne funkcije. Instance posmatrane pojave su opisane vrednostima svojih atributa. Postupak klasifikacije se vrši polazeći od korena, potom spuštajući se niz granu koja odgovara vrednosti testiranog atributa instance koju klasifikujemo i kada se dođe do lista, klasa se dodeljuje instanci.

Ako stablo odlučivanja instanci dodeljuje neku klasu, to znači da instanca ispunjava sve uslove koji su definisani putanjom od korena do odgovarajućeg lista kroz stablo i oblika su *atribut=vrednost*. Putanje kroz stablo predstavljaju konjunkcije ovakvih uslova i za svaku klasu moguće je uočiti putanje koje se završavaju listovima koji odgovaraju toj klasi. Disjunkcija svih takvih konjunkcija definiše instance koje pripadaju datoj klasi prema datom stablu.

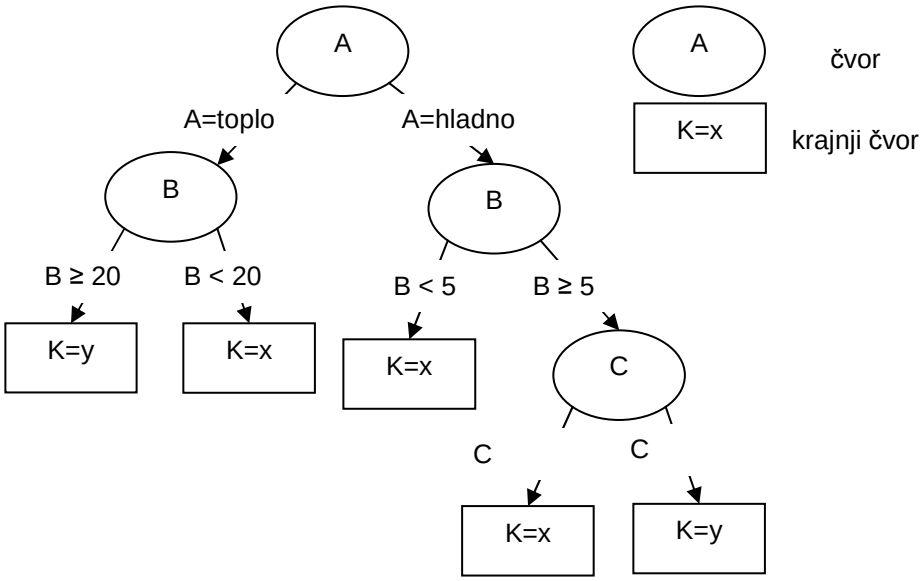
Kod stabla odlučivanja razlikujemo dva tipa čvorova povezanih granama (Slika 4.11):

- krajnji čvor (eng. *leaf node*) - kojim završava određena grana stabla i on definiše klasu kojoj pripadaju primeri koji zadovoljavaju uslove na toj grani stabla;

- čvor odluke (eng. *decision node*) - ovaj čvor definiše određeni uslov u obliku vrednosti određenog atributa, iz kojeg izlaze grane koje zadovoljavaju određene vrednosti tog atributa.

Za klasifikaciju primera, stablo odlučivanja se može koristiti tako da se krene od prvog čvora odlučivanja u korenu stabla i kreće se po onim granama stabla koji primer sa svojim vrednostima zadovoljava sve do krajnjeg čvora koji klasifikuje primer u jednu od postojećih klasa problema.

Stablo odlučivanja može se posmatrati u formi povezanog grafa sa strukturom stabla, gde su unutrašnji čvorovi u stablu odlučivanja označeni sa nazivima atributa, a grane koje izlaze iz unutrašnjih čvorova označene su mogućim vrednostima odgovarajućeg atributa. Primeri se razvrstavaju u klase koje predstavljaju listove stabla odlučivanja. U ovom slučaju se klasifikacija sprovodi tako što se sledi određeni put od korena stabla do nekog od listova. Unutrašnji čvorovi stabla predstavljaju test na vrednost određenog atributa, pa se put gradi dodavanjem one grane koja odgovara vrednosti atributa u posmatranom primeru, a put se završava u nekom od listova. Na ovaj način se primer klasifikuje u klasu kojom je list označen.



Slika 4.11: Primer jednostavnog stabla odlučivanja

Kod ovog modela, prostor pretraživanja se sastoji od svih stabala koje je moguće konstruisati koristeći atribute i vrednosti iz skupa podataka za učenje. Modifikacija stabla se vrši na način da se jedan od listova zameni podstablom visine 1 i to operacijom transformacije kojom se prelazi prostor pretraživanja.

Postupak pretraživanja je usmeravan funkcijom vrednovanja, koja zavisi o tačnosti klasifikacije, ali i veličini rezultirajućeg stabla. Delovanje funkcije

vrednovanja zasnovano je na konceptima iz teorije informacija, a ogleda se u izboru grananja pri konstrukciji stabla odlučivanja.

Prema principu *Ockham*-ove oštrice, za objašnjenje nekog fenomena treba pretpostaviti što je moguće manje pretpostavki, eliminišući tj. odsecajući kao oštricom one pretpostavke, koje ne doprinose predviđanjima hipoteze ili teorije. Kada više različitih teorija ima jednaku mogućnost predviđanja, princip preporučuje da se uvede što je moguće manje pretpostavki i da se postulira sa što je moguće manje hipotetičkih entiteta. Ako primenimo ovaj princip na stabla odlučivanja, uz sličnu tačnost klasifikovanja, izglednije je da će jednostavnija (tj. manja) stabla odlučivanja bolje klasifikovati dotad neviđene primere.

4.5.2. Postupak pretraživanja

Osnovni algoritam konstrukcije stabala odlučivanja star je nekoliko decenija, a razvio ga je J. Ross Quinlan [Quinlan, 1986]. Osnovna verzija algoritma poznata je pod nazivom ID3 (eng. *Induction of Decision Trees*), dok su kasnije verzije algoritma uklanjale neka od ograničenja izvornog algoritma, i poboljšavale klasifikacijske performanse. Algoritam konstrukcije stabala odlučivanja C4.5 je danas najpoznatiji i verovatno najviše korišćen algoritam [Quinlan, 1993].

Kod ovih algoritama pretraživanju se pristupa po načelu *odozgo na dole*, tj. od opšteg ka specifičnom i koristi se strategija nepovratnog pretraživanja i to pohlepna metoda uspona na vrh. Pretraživanje je relativno brzo jer se pregleda samo manji deo prostora pretraživanja, ali je postupak podložan zamci lokalnih maksimuma.

Osnovni algoritam je u svojoj osnovi rekurzivan, što znači da postupak u svojoj definiciji koristi sam sebe, odnosno zahteva da delovi problema koje je razdvojio od drugih bivaju nezavisno podvrgnuti istom postupku.

Ako je sa S označen skup podataka za učenje, sa C={Ci, 1 ≤ i ≤ |C|} skup odgovarajućih klasa, a sa A={Ai, 1 ≤ i ≤ |A|} skup odgovarajućih atributa, sa S' □ S i A' □ A □ A parametri koji se prosleđuju algoritmu, s tim da se inicijalno u rekurziju ulazi sa S' = S i A' = A, onda su osnovni koraci algoritma [Ujević, 2004]:

1. Ako je *S'* prazan, stablo odlučivanja je list označen globalno najfrekventnijom klasom C_i unutar *S*.
2. Ako se *S'* sastoji od primera samo jedne klase C_j, stablo odlučivanja je list označen klasom C_j.
3. Ako je A' prazan, stablo odlučivanja je list označen najfrekventnijom klasom Ck unutar S'.
4. Inače, izaberi atribut Ai□A'. Ako je {aj, 1 ≤ j ≤ n} skup svih vrednosti atributa Ai. Podeli S' na n podskupova S'j tako da S'j sadrži sve primere iz S' kod kojih atribut Ai ima vrednost aj, odnosno S'j = {s□S', Ai(s) = aj}. Stvori unutrašnji čvor označen atributom A_i, kao i grane označene njegovim vrednostima a_j. Za granu označenu vrednošću a_j konstruiši podstablo rekurzivnim pozivom postupka sa parametrima S'_j i $A' - \{A_j\}$.

Algoritam konstruiše stablo odlučivanja od korena prema listovima, pri čemu se u svakom koraku rekurzije generiše podstablo visine 1, uz to se koriste samo oni primeri koji pripadaju tom podstablu. Podstabla se konstruišu izborom jednog od atributa, pri čemu su iz razmatranja isključeni svi oni atributi koji su pre iskorišćeni u istoj grani stabla. Svaki atribut se može pojaviti najviše jednom na bilo kojem putu od korena do lista. U ovom osnovnom obliku algoritma, implicitno se pretpostavlja da su svi atributi nominalnog tipa. Postupak rekurzije se odvija sve dok za posmatrani čvor stabla nije zadovoljen jedan od dva kriterijuma. Prvi kriterijum je da je skup podataka za učenje koji pripada čvoru prazan ili se sastoji od primera samo jedne klase, zbog čega je dalje grananje nepotrebno. Drugi kriterijum podrazumeva da su svi atributi već iskorišćeni na putu od korena do posmatranog čvora, zbog čega dalje grananje nije moguće.

Jednom odabrani atribut postaje osnova za grananje stabla, bez mogućnosti da se taj izbor naknadno preispita. Način izbora atributa je presudan za kvalitet konačnog rezultata, jer struktura stabla odlučivanja zavisi isključivo o izboru atributa za grananje u svakom koraku rekurzije. To je razlog zbog koga kriterijum izbora atributa za grananje predstavlja centralni deo algoritma, koji usmerava pretraživanje u skupu potencijalnih rešenja. Da li će rezultirajuće stablo biti glomazna struktura preterano prilagođena skupu za učenje, ili kompaktni prikaz opštih pravilnosti koje postoje u podacima, zavisi od načina izbora atributa.

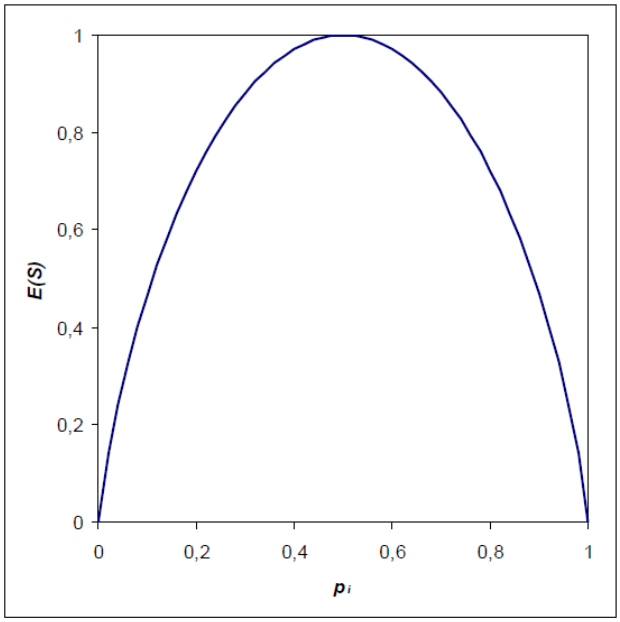
4.5.3. Način izbora atributa

U algoritmu konstrukcije stabala odlučivanja, funkcija vrednovanja aproksimacije vezana je uz izbor atributa koji će poslužiti kao kriterijum grananja u unutrašnjim čvorovima stabla. Pri tome, treba nastojati što ranije zaustaviti rekurzivni proces grananja, jer je cilj konstrukcija što manjeg stabla odlučivanja. Bazični način zaustavljanja rekurzije su čvorovi kojima pripadaju primeri samo jedne od klasa, zbog čega je poželjno kao kriterijum grananja izabrati one atribute koji proizvode što homogenije podskupove primera za učenje kao rezultat grananja.

Entropija ili informacijska vrednost predstavlja dobru meru (ne)homogenosti nekog skupa. Za klasifikacijski problem sa dve klase označimo sa p_1 relativnu frekvenciju klase C_1, a sa p_2 relativnu frekvenciju klase C_2 u skupu primera za učenje S. Sledećim izrazom možemo pretstaviti entropiju skupa za učenje:

$$E(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \qquad (4.26)$$

uz pretpostavku da važi $0\log_2 0 = 0$. Logaritmi u izrazu (4.26) su negativni jer je $p_1, p_2 \leq 1$ ($p_1 + p_2 = 1$), pa je entropija uvek veća ili jednaka 0. Minimalnu vrednost 0 entropija dobija kada svi primeri iz skupa za učenje S pripadaju istoj klasi, tj. kada je $p_1 = 1$ ili $p_2 = 1$. Maksimalnu vrednost 1 entropija dostiže kada skup za učenje S sadrži jednak broj primera obe klase, tj. kada je $p_1 = p_2$.



Slika 4.12: Entropija u zavisnosti od relativne frekvencije klasa kod binarne klasifikacije [Ujević, 2004]

Na slici 4.12. prikazana je funkcija entropije u zavisnosti od relativne frekvencije jedne od klasa u skupu podataka za učenje za klasifikacijski problem sa dve klase.

Entropija se meri u bitovima, jer jedna od interpretacija entropije iz teorije informacija kaže da ona specificira minimalnu količinu informacije (izraženu u bitovima) potrebne da se kodira klasifikacija slučajno izabranog primera iz skupa *S*, uz činjenicu da primer pripada posmatranom čvoru. Kod klasifikacijskih problema sa više klasa potrebno je uopštiti definiciju entropije, ali zadržati poželjna svojstva vezana za dosezanje minimuma i maksimuma. Sledećim izrazom možemo definisati entropiju skupa primera *S*:

$$E(S) = \sum_{i=1}^{|C|} - p_i \log_2 p_i \tag{4.27}$$

pri čemu $|C|$ označava broj klasa prisutnih u skupu podataka za učenje *S*, a p_i relativnu frekvencija klase C_i unutar *S*, $1 \leq i \leq |C|$. U ovom slučaju maksimalna vrednost entropije iznosi $log_2 |C|$, a postiže se pri jednakoj zastupljenosti svih klasa unutar *S*, dok minimum entropije i dalje iznosi 0, za slučaj kada svi primeri iz *S* pripadaju istoj klasi.

Informacijski dobitak se definiše na osnovu entropije i on služi kao mera efektivnosti atributa u klasifikaciji primera. Informacijski dobitak atributa A_i u odnosu na *S* definiše se sledećim izrazom:

$$IGain(S, A_i) = E(S) - \sum_{a_j \in Dom(A_i)} \frac{|S_j|}{|S|} E(S_j) \tag{4.28}$$

gde je sa A_i označen proizvoljni atribut koji se pojavljuje u skupu podataka za učenje S, a $S_j \subseteq S$ označava skup $S_j = \{s \in S, A_i(s) = a_j\}$.

Informacijski dobitak $IGain(S, A_i)$ predstavljen izrazom (4.28) predstavlja očekivanu redukciju entropije (tj. dobitak na homogenosti) uzrokovanu poznavanjem vrednosti atributa A_i, gde prvi član izraza predstavlja entropiju originalnog skupa S, a težinska suma u drugom članu iskazuje očekivanu vrednost entropije podskupova nastalih grananjem na osnovu atributa A_i.

Kao kriterijum za izbor atributa u algoritmu stabala odlučivanja koristi se upravo informacijski dobitak, budući da se grananjem nastoji što ranije postići homogenost rezultirajućih podskupova. U svakom čvoru od dostupnih atributa za grananje izabere se onaj koji proizvodi najveći informacijski dobitak. Heuristika zasnovana na teoriji informacija ne garantuje konstrukciju najmanjeg stabla odlučivanja, ali dobro ispunjava svoju ulogu redukcije opsega pretraživanja u skupu potencijalnih rešenja.

Svojstvo informacijskog dobitka da favorizuje atribute sa većim brojem vrednosti, pri konstrukciji stabala odlučivanja može predstavljati problem. Ekstremni slučaj je da atribut ima različitu vrednost za svaki primer iz skupa za učenje, pri čemu grananje na osnovu ovog atributa particionira skup za učenje na jednočlane podskupove. Entropija takvog grananja je 0, pa je informacijski dobitak maksimalan, a kao rezultat dobijamo stablo koje se sastoji samo od korena i listova za svaki od primera iz skupa za učenje. U smislu prediktivnih sposobnosti, dobijeno stablo je beskorisno.

Da bi se smanjio uticaj ovog problema, koristi se korekcija kriterijuma za izbor atributa. Ova korekcija u obzir uzima broj i kardinalnost podskupova koji nastaju kao rezultat grananja. Kao kriterijum izbora atributa, pri konstrukciji stabala odlučivanja, po pravilu se koristi korigovana mera dobitka prikazana sledećim izrazom:

$$Gain(S, A_i) = \frac{IGain(S, A_i)}{-\sum_{a_j \in Dom(A_i)} \frac{|S_j|}{|S|} \log_2 \frac{|S_j|}{|S|}} \tag{4.29}$$

U izrazu (4.29) korektivni faktor je imenilac, koji raste s povećanjem broja rezultirajućih podskupova odnosno smanjenjem njihove kardinalnosti, čime se penaliziraju atributi s većim brojem vrednosti. Takođe, postoji mogućnost da korekcija informacijskog dobitka u nekim slučajevima preterano preferira atribute s manjim brojem vrednosti, na štetu informacijski vrednijih atributa. Za eliminisanje

ove mogućnosti koristi se standardna provera koja traži da atribut koji maksimizira korigovani dobitak mora imati informacijski dobitak jednak ili veći od proseka svih posmatranih atributa.

4.5.4. Izbegavanje nepotrebnog grananja stabla

Algoritam u principu može generisati stablo, dovoljno kompleksno da tačno klasifikuje sve primere iz skupa podataka za učenje, iako je to u određenim slučajevima razumna strategija, u većini situacija to rađa dodatne probleme, bilo zbog šuma u podacima, ili pak nedovoljno velikog uzorka podataka koji bi trebao reprezentovati populaciju primera za određeni klasifikacijski problem. Jednostavni algoritam bi generisao stablo koje se preterano dobro prilagođava podacima za učenje (eng. *over-fitting*).

Značajnu poteškoću u primeni metoda stabla odlučivanja, ali i drugih tehnika modeliranja podataka predstavlja *over-fitting*. Moguća rešenja za izbegavanje *over-fitting*-a su:

- rešenja koja zaustavljaju proces rasta stabla pre nego što se postigne savršena klasifikacija primera iz skupa podataka za učenje,
- rešenja u kojima se najpre generiše stablo koje savršeno klasifikuje primere, a potom se određene grane stabla „skraćuju“ prema prethodno definisanom kriterijumu.

Drugi se pristup u praksi pokazao pouzdanijim, iako se na prvi pogled prvi pristup čini direktnijim, što je posledica toga što je teško unapred definisati željenu kompleksnost stabla odlučivanja.

Određivanje optimalne kompleksnosti, odnosno veličine stabla za konkretni probleme moguće je uz pomoć sledećih pristupa:

- korišćenje posebnog skupa primera, odnosno validacijskog skupa, koji je različit od onog korišćenog za generisanje stabla, da bi se ocenila uspešnost „skraćivanja“ stabla,
- korišćenje posebnog statističkog testa na čvorovima koji su kandidati za „skraćivanje“, kojima se pokazuje da li će se izbacivanjem tog čvora postići poboljšanje,
- korišćenje eksplicitne mere kompleksnosti kodiranja primera stablom odlučivanja, koja zaustavlja rast stabla kada je taj kriterijum zadovoljen.

Prvi pristup se i najčešće koristi, i on podrazumeva da se primeri dele u dva skupa: skup za učenje koji se koristi za generisanje stabla, i skup za proveru, koji se koristi za proveru efikasnosti metode skraćivanja stabla.

Napred prikazani algoritam konstrukcije stabala odlučivanja nastoji konstruisati stablo koje savršeno klasifikuje primere iz skupa podataka za učenje, ali u tome ne uspeva samo u granama za koje se rekurzija zaustavlja zbog nedostatka atributa za grananje. Tada, skup primera *S'* koji odgovara

posmatranom listu sadrži više od jedne klase, zbog čega se list označava najfrekventnijom klasom među njima. Probabilistička interpretacija je alternativa prethodnom pristupu, i po ovom pristupu list se označava svim klasama iz *S'*, uz pridruživanje pripadajuće verovatnoće svakoj od njih, što odgovara relativnoj frekvenciji klase unutar *S'*. Algoritam vrši razgranavanje stabla nastojeći akomodirati svaki primer iz skupa podataka za učenje dok god postoje mogući atributi za grananje.

Ipak, savršena klasifikacija primera iz skupa podataka za učenje ne garantuje dobre klasifikacijske performanse na dotada neviđenim primerima, a uzrok tome može biti činjenica da skup podataka za učenje nije reprezentativan uzorak cele populacije primera ili zbog postojanja šuma u podacima za učenje, kako u prognostičkim atributima tako i u klasi. Kao posledicu imamo preterano razgranato stablo odlučivanja zbog grananja na atributima koji samo prividno proizvode informacijski dobitak, dok je stvarni uzrok grananja šum u primerima za učenje, i ova pojava se naziva preterana prilagođenost podacima za učenje.

Ako posmatramo primere iz skupa podataka za učenje *S* slučajno raspoređene u klase C_1 i C_2, odnosno tako da ne postoji korelacija između prognostičkih atributa i klase primera, i neka je relativna frekvencija klase C_1 unutar *S* označena sa *p*, a klase C_2 sa 1–*p*, tada se bez smanjenja opštosti može pretpostaviti $p \geq 0.5$.

U slučaju najjednostavnijeg stabla odlučivanja koje se sastoji samo od korena označenog klasom C_1, očekivana frekvencija grešaka iznosi 1–*p*, budući da takvo stablo svaki primer klasifikuje u klasu C_1.

U slučaju stabla koje se sastoji od korena i dva lista označena klasama C_1 i C_2 i ako test u korenu stabla funkcioniše tako da primeru dodeljuje klasu C_1 s verovatnoćom *q*, a klasu C_2 s verovatnoćom 1–*q*, očekivana frekvencija grešaka takvog stabla može se pretstaviti sledećim izrazom:

$$q(1-p)+(1-q)p = p+q-2pq \tag{4.30}$$

I s obzirom da vredi:

$$1-p \leq p+q-2pq, \text{ za svaki } q, \text{ uz } p \geq 0.5 \tag{4.31}$$

Zbog ovoga, klasifikacijske performanse jednostavnog nerazgranatog stabla sa jednim čvorom nadmašiće bilo koje binarno stablo sa 3 čvora. Postoje dva načina za izbegavanja nepotrebnog grananja stabla. Prvi način je *zaustavljanje grananja* prilikom konstrukcije stabla odlučivanja pre postizanja savršene klasifikacije primera iz skupa za učenje, dok je drugi način *naknadno podrezivanje* razgranatog stabla koje se sprovodi nakon procesa konstrukcije stabla odlučivanja. Svaki od pristupa ima svoje prednosti. Efikasniji pristup je zaustavljanje grananja stabla, jer se izbegava konstrukcija nepotrebnih podstabala koja opet treba posebnim postupkom podrezivati, ali postoji rizik od preranog zaustavljanja rasta stabla. U slučaju dva atributa koje odvojeno posmatramo, oni se mogu se činiti gotovo nevažnima, ali njihova kombinacija može imati izrazite prediktivne sposobnosti, zbog čega je u ovom slučaju bolje koristiti pristup podrezivanja, koje kao podlogu uzima potpuno razgranato stablo.

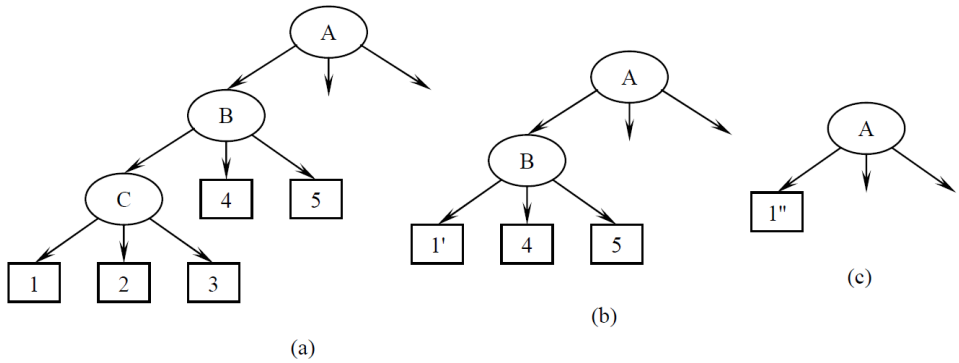
Prilikom konstrukcije stabala odlučivanja, kriteriji za zaustavljanje grananja uglavnom se oslanjaju na statističke tehnike ocene relevantnosti atributa izabranog za grananje, i pri tome se često se koristi χ^2 test. Test χ^2 nastoji da utvrdi statističku nezavisnost vrednosti atributa A_i i klase primera u skupu za učenje S.

Ako sa pk(S) označimo relativnu frekvenciju klase Ck unutar S, a sa p'k(Sj) očekivanu relativnu frekvenciju klase Ck unutar Sj, i pri tome Sj = {s□S, Ai(s) = aj},

i ako su vrednosti atributa A_i i klase nezavisne, tada za sve skupove S_j koji nastaju kao rezultat grananja na atributu A_i vredi $p'_k(Sj) = p_k(S)$. U tom slučaju izraz (4.32) ima χ^2 distribuciju sa $|Dom(A_i)|-1$ stepena slobode.

$$\sum_{a_j \in Dom(A_i)} \sum_{k=1}^{|C|} \frac{(p_k(S_j) - p'_k(S_j))^2}{p'_k(S_j)} \qquad (4.32)$$

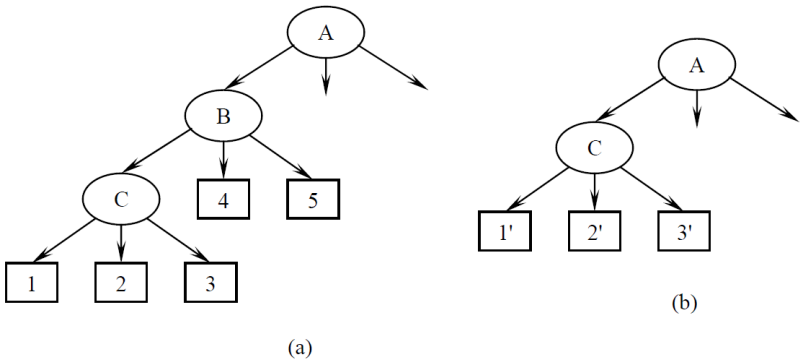
U algoritmu konstrukcije stabala odlučivanja, zaustavljanje grananja se može sprovesti na način da se u obzir uzimaju samo oni atributi čija nezavisnost u odnosu na klasu može biti odbačena sa vrlo visokom pouzdanošću. Najpoznatije tehnike podrezivanja stabala odlučivanja su zamena podstabala i izdizanje podstabala, a primena jedne od njih ne isključuje drugu.



Slika 4.13: Primer zamene podstabala [Witten *et al.*, 2005]

Jednostavnija tehnika je zamena podstabala, koja može celo podstablo redukovati na jedan list, pri čemu se menjaju verovatnoće klasa u listu. Kod ove tehnike postupak se sprovodi nad unutrašnjim čvorovima stabla koji kao decu imaju samo listove (odnosno posmatraju se samo podstabla visine 1) i za svako takvo podstablo razmatra se opravdanost zamene samo jednim listom, uz pripadajuću korekciju raspodele verovatnoće među klasama. Ovaj postupak se sprovodi od listova prema korenu stabla, tako da se uzastopnom primenom listom mogu zameniti podstabla proizvoljne visine. Na slici 4.13. dat je primer uzastopne primene zamene podstabala na čvorovima *C* i *B*, tako da je celo podstablo sa korenom *B*, na slici (a) u završnom rešenju zamenjeno jednim listom *1"* na slici (c).

Složenija operacija podrezivanja je izdizanje stabala, a sprovodi se nad unutrašnjim čvorovima koji kao decu imaju barem jedan čvor koji nije list, i izabere se jedno dete takvoga čvora, i njime se zamenjuje polazni čvor. Celo podstablo pod odabranim detetom je izdignuto na jedan nivo više u odnosu na polazno stablo. Opisanim postupkom efektivno nestaju ostala deca polaznog čvora, zbog čega je potrebno primere koji su pripadali njima reklasifikovati u novonastalo podstablo.



Slika 4.14: Primer izdizanja podstabala, gde je čvor C izdignut [Witten *et al.*, 2005]

Slika 4.14. prikazuje stablo odlučivanja pre i posle primene izdizanja podstabla na čvoru *B*, pri čemu se čvor *B* zamenjuje detetom *C*, a primeri koji su pripadali ostaloj deci čvora *B* reklasifikuju se u novo podstablo sa korenom *C*. Reklasifikacija utiče na verovatnoće pridružene klasama u listovima novog podstabla. Izdizanje stabala je vremenski potencijalno zahtevna operacija, zbog potrebe za reklasifikacijom. U praktičnim implementacijama, kao kandidat za izdizanje razmatra se samo najpopularnije dete polaznog čvora, odnosno ono kojem pripada najviše primera za učenje.

Kod tehnike podrezivanja stabla odlučivanja ključan element je kriterijum na osnovu kojeg se odlučuje treba li potencijalnu operaciju podrezivanja zaista i sprovesti. Cilj podrezivanja je smanjenje frekvencije grešaka na dotada neviđenim primerima, zbog čega je neophodno proceniti frekvenciju grešaka u svakom od čvorova stabla. Poređenjem procenjene frekvencije grešaka za originalno podstablo i njegove predložene alternative dobijene podrezivanjem, donosi se odluka da li treba sprovesti operaciju podrezivanja.

Za procenu stvarne frekvencije grešaka u čvorovima stabla odlučivanja, koristi se više tehnika, od kojih najčešće korišćena koristi standardnu verifikacijsku tehniku na odvojenom skupu primera. Pri tome se izvorni skup primera za učenje deli na dva dela, skup koji će služiti za generisanje stabla odlučivanja i validacijski skup koji će služiti za proveru opravdanosti operacije podrezivanja stabla. Nedostatak ovog pristupa zasniva se na činjenici da se stablo odlučivanja konstruiše iz manjeg broja primera, zbog izdvajanja validacijskog skupa. Pored ove, postoje i druge tehnike koje se oslanjaju na skup podataka za učenje, a imaju heuristički pristup oceni stvarne frekvencije grešaka. Statistička utemeljenost postupka je pod znakom pitanja, i pored toga što se koriste nekim statističkim

izračunavanjima, jer koristi isti skup podataka za učenje na osnovu kojeg je izgrađeno stablo. U praksi ipak pokazuju dobre rezultate pri određivanju opsežnosti operacija podrezivanja, čime se opravdava njihova primena.

4.5.5. Tipovi atributa kod algoritma za konstrukciju stabla odlučivanja

Osnovni oblik algoritma za konstrukciju stabla odlučivanja, a to je ID3, ograničen je na nominalne atribute. Prvi zahtev je da ciljni atribut mora imati ograničen broj klasa, a drugi zahtev je da atributi koji se testiraju u čvorovima odlučivanja takođe moraju imati diskretne vrednosti. Drugi zahtev se može relativno lako zadovoljiti i u slučaju da je atribut numeričkog tipa, odnosno u slučaju realnih numeričkih varijabli, i to prethodnom diskretizacijom.

Međutim, kasnije verzije algoritma na jednostavan način uvode efikasnu podršku radu sa numeričkim atributima. Korišćenjem nominalnog atributa, grananje stabla rezultira granom za svaku od mogućih vrednosti posmatranog atributa. Opisani oblik testa ne može se primeniti na numeričke atribute, zbog čega se kod njih po pravilu testiranje vrednosti ograničava na binarni test oblika $A_i(s) < x$, gde je x konstanta izabrana za taj test. Kod ovakvog testa unutrašnji čvor stabla ima dve izlazne grane, i to jednu za pozitivni, a drugu za negativni ishod testa. Za opisano grananje, pripadajući informacijski dobitak se računa na standardni način, uz napomenu da notacija sume po različitim vrednostima atributa nije prikladna, već se sumira po (binarnoj) particiji skupa za učenje. Opisani testovi na numeričkim atributima sudeluju u procesu izbora atributa za grananje, zajedno sa ostalim atributima koji su na raspolaganju.

Jedini problem je izbor granice interesantnih intervala za formiranje testa na numeričkom atributu, odnosno vrednosti konstante x. Zbog toga što se u skupu podataka za učenje pojavljuje samo konačan skup njegovih vrednosti, uobičajeno je da se kao kandidati za granicu x posmatraju aritmetičke sredine susednih vrednosti, zbog čega je potrebno sortirati primere za učenje prema vrednosti posmatranog numeričkog atributa.

Ako numerički atribut Ai u skupu za učenje S' može da poprimi m različitih vrednosti, i neka je $(v_j)_{j=1}^{m}$ niz uzlazno sortiranih vrednosti atributa Ai, odnosno vj≤vj+1 za svaki j□{1,..., m–1}, onda izraz (4.33) daje m –1 kandidata za vrednost granice testa.

$$x_j = \frac{v_j + v_{j+1}}{2}, j \in \{1, \ldots, m-1\} \tag{4.33}$$

Vrednost x_j za koju je informacijski dobitak maksimalan bira se za granicu testa. Vrednosti x_j koje maksimiziraju informacijski dobitak uvek se nalaze između primera koji pripadaju različitim klasama, zbog čega je dovoljno posmatrati samo one vrednosti xj za koje skup {s□S', Ai(s)=vj □ Ai(s)=vj+1} sadrži primere barem dve klase.

Potrebno je izvršiti sledeće izmene algoritma konstrukcije stabala odlučivanja, ako postoje numerički atributi. Prva izmena se odnosi na sortiranje primera prema vrednosti svakog od numeričkih atributa. Čini se da je sortiranje potrebno sprovoditi u svakom koraku rekurzije, odnosno za svaki unutrašnji čvor stabla u kojem se razmatraju numerički atributi, ali s obzirom da redosled primera u roditelju inducira redosled u deci, možemo zaključiti da je sortiranje potrebno izvršiti samo jednom, u korenu stabla. Druga izmena se odnosi na prosleđivanje skupa atributa u sledeći korak rekurzije.

Pri grananju stabla na nominalnom atributu A_i, svaka grana odgovara jednoj vrednosti tog atributa, pa je svako dalje ispitivanje vrednosti atributa A_i u bilo kojoj od nastalih grana nepotrebno, zbog čega se u sledećem koraku rekurzije prosleđuje skup atributa $A' - \{A_i\}$. Međutim, ovo ne važi kod numeričkih atributa, jer binarni test ne iskorišćava u potpunosti informacije koje atribut nosi, zbog čega ponovno grananje korišćenjem istog numeričkog atributa, ali uz drugu granicu testa, može rezultirati povećanjem informacijskog dobitka. Ovo znači da višestruko testiranje istog atributa na putu od korena do lista ima smisla u slučaju numeričkih atributa, zbog čega se kod grananja na numeričkom atributu u sledeći korak rekurzije prosleđuje celi skup atributa A'.

4.5.6. Nedostajuće vrednosti atributa

Ako u primerima nedostaju vrednosti atributa, onda nastaju poteškoće pri konstrukciji stabla odlučivanja i pri klasifikaciji primera izgrađenim stablom odlučivanja. U nekim praktičnim primenama postoje atributi kod kojih određeni procenat primera ima nedostajuće vrednosti, kao na primer, u medicinskoj oblasti gde je čest slučaj da su određeni rezultati laboratorijskih testova dostupni samo za deo pacijenata. Tada je uobičajeno da se vrednosti tih atributa odrede na osnovu ostalih pacijenata koji poseduju rezultate tih testova.

Ako razmotrimo situaciju u kojoj treba izračunati $Gain(S, A)$ za čvor n u stablu, za atribut A da bi odredili da li taj atribut predstavlja kandidata za test na čvoru n. Ako uvedemo pretpostavke da je vrednost $A(x)$ nepoznata, gde $c(x)$ predstavlja vrednost klase primera x, onda su mogući načini rešavanja ovog problema:

- prvi, da se umesto neodređene vrednosti za atribut A koristi najčešća vrednost za taj atribut u primerima koji se nalaze na čvoru n.
- drugi, da se nadomešćuje sa najčešćom vrednosti tog atributa kod primera iste klase $c(x)$, na čvoru n.

U slučaju da vrednost atributa u primerima nije zabeležena, i da to nosi neko značenje, onda je opravdano uvođenje nove vrednosti tipa „nepoznato“, kojom se menjaju sve nedostajuće vrednosti atributa u primerima za učenje i u primerima za klasifikaciju. Tako uvedena vrednost se i pri konstrukciji i pri klasifikaciji tretira na standardan način, pa nikakve izmene algoritama nisu potrebne.

4.5.7. Prednosti i nedostaci stabala odlučivanja

Ova tehnika modeliranja pravilnosti u podacima je intenzivno korišćena i često izučavana, pri čemu su istraživane i različite varijacije postupka konstrukcije stabala, od različitih kriterijuma za izbor atributa grananja, do drugih metoda podrezivanja stabla, ili modifikovanog oblika testova u čvorovima (npr. korišćenjem više od jednog atributa ili vrednosti [Breiman *et al.*, 1984]).

Stabla odlučivanja vrlo su moćna i popularna tehnika modeliranja za klasifikacijske i predikcijske probleme, a njena privlačnost leži pre svega u činjenici da nudi modele podataka u „čitljivom“, razumljivom obliku - odnosno u obliku pravila. Za neke je probleme od ključne važnosti samo tačnost klasifikacije ili predikcije modela i u takvim slučajevima čitljivost modela nije od presudne važnosti. No, u drugim situacijama upravo sposobnost interpretiranja modela „ljudskim“ jezikom je od ključne važnosti.

Naročito važno kod primene na velikim skupovima podataka je primena korekcije postupka koji smanjuje potrebne računarske i vremenske zahteve. Primer jedne takve korekcije je da se pri konstrukciji stabla koristi samo manji, slučajno izabrani deo skupa primera za učenje, tzv. *prozor*. Ovako konstruisano stablo klasifikuje preostali deo skupa za učenje, i izdvaja pogrešno klasifikovane primere, koji se pridodaju prozoru i postupak se iterativno ponavlja na ovako izmenjenom skupu primera, sve dok se ne postigne zadovoljavajuća tačnost klasifikacije na preostalim primerima. Ovaj postupak se po pravilu zaustavlja nakon samo nekoliko iteracija i on je primetno brži od konstrukcije stabla na celom skupu za učenje.

Za korišćenje tehnike stabla odlučivanja osnovni preduslovi su:

- opis u obliku parova vrednosti-atributa - instance moraju biti opisane konačnim brojem atributa;
- prethodno definisan konačan broj klasa - kojima instance pripadaju moraju biti definisane unapred i treba ih biti konačan broj;
- klase moraju biti diskretne - svaka instanca mora pripadati samo jednoj od postojećih klasa, kojih mora biti znatno manje nego broj instanci;
- značajan broj instanci - obično je poželjno da u skupu instanci za generisanje stabla odlučivanja postoji barem nekoliko stotina instanci.

Prednosti korišćenja tehnike stabala odlučivanja kod klasifikacijskih problema su:

- sposobnost za generisanje razumljivih modela,
- eksplicitno izdvajanje atributa bitnih za određeni klasifikacijski problem,
- relativno mali zahtevi za računarske resurse (vreme i memorija),
- sposobnost korišćenja svih tipova atributa (kategoričkih i numeričkih),
- stabla odlučivanja jasno odražavaju važnost pojedinih atributa za konkretni klasifikacijski problem.

Tehnika stabla odlučivanja je posebno primenljiva u slučaju kada je neophodno predstavljanje disjunkcija uslova, kada podaci za trening sadrže greške i kada u trening skupu postoje instance kojima nedostaju vrednosti nekih atributa.

Nedostaci korišćenja tehnike stabla odlučivanja kod klasifikacijskih problema su:

- da su manje prikladne za probleme kod kojih se traži predikcija kontinuiranih vrednosti ciljnog atributa,
- da su sklona greškama u više-klasnim problemima sa relativno malim brojem instanci za učenje modela,
- da u nekim situacijama generisanje stabla odlučivanja može biti računarski zahtevan problem. Tako na primer, sortiranje kandidata za testiranje na čvorovima stabla može biti zahtevno, kao i metode „skraćivanja“ stabla, kod kojih je često potrebno generisati velik broj stabala da bi odabrali ono koje je najbolje za klasifikaciju primera određenog problema,
- da nisu dobro rešenje za klasifikacijske probleme kod kojih su regije određenih klasa „omeđane“ nelinearnim krivama u više-dimenzionalnom atributnom prostoru. Mnoge metode stabla odlučivanja testiraju u svojim čvorovima vrednosti jednog atributa, i time formiraju pravougaone regije u više-dimenzionalnom prostoru.

Ova tehnika nije podjednako pogodna za sve probleme učenja, na primer u slučaju kada je potrebno instance predstaviti pomoću vrednosti fiksnog broja atributa i onda kada skup vrednosti nije diskretan i mali. Ako postoje kontinualne vrednosti atributa može se primeniti diskretizacija tako što bi se skup podelio u podintervale, a svakom podintervalu se pridružuje oznaka koja zamenjuje vrednosti atributa iz tog intervala u zapisima instanci. Osnovni nedostatak stabala odlučivanja je sklonost preteranom prilagođavanju podacima za učenje. Pri izboru tehnike modeliranja za konkretni klasifikacijski problem potrebno je imati u vidu nabrojane prednosti i nedostatke.

4.6. RBF neuronske mreže

U nastavku teksta biće reči o metodama klasifikacije koje su zasnovane na neuronskim mrežama. Biće date osnove razvoja neuronskih mreža, prikaz modela neuronskih mreža i statičke neuronske RBF mreže, prednosti i nedostaci ovog algoritma, kao i prikaz pseudo koda za RBF mrežu.

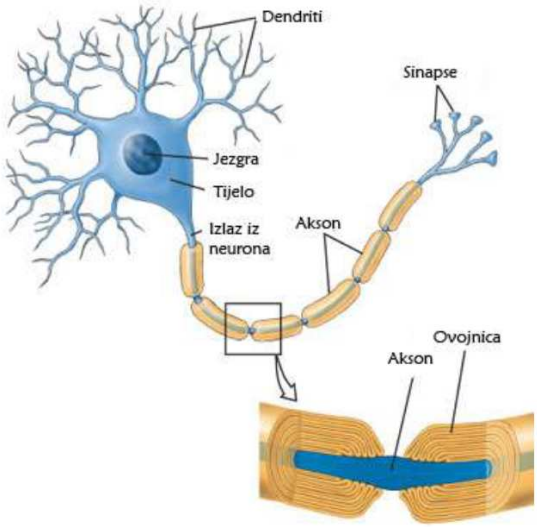
4.6.1. Osnove razvoja neuronskih mreža

Ove mreže nastale su iz težnje za razvijanjem matematičkih struktura koje bi bile u mogućnosti da oponašaju rad ljudskog mozga, kao i da koriste te strukture

u rešavanju praktičnih problema. Postoji više različitih vrsta neuronskih mreža, ali ih sve možemo svrstati u statičke ili dinamičke neuronske mreže. U ovom radu koristi se model statičke neuronske RBF mreže.

Kako bi razumeli osnovnu strukturu veštačkih neuronskih mreža potrebno je razmotriti osnovnu strukturu ljudskog mozga. Ljudski mozak, čija je struktura složena, a mreža neurona gusta, sastoji se od oko 10^{11} neurona koji su međusobno povezani u slojeve, koji čine složenu mrežu. Biološka ćelija koja obrađuje informacije je neuron. Zbog složene strukture neurona još uvek nije došlo do detaljnijih saznanja o funkcionisanju ljudskog mozga.

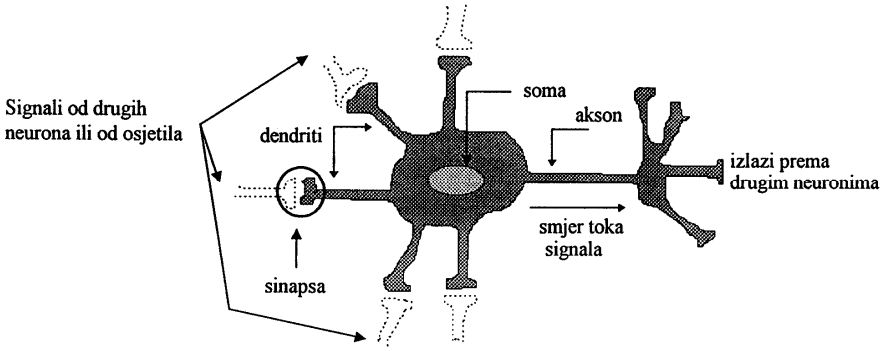
Biološki neuron, koji je prikazan na slici 4.15, poslužio je kao model za veštački neuron. Biološki neuron se može pojednostavljeno prikazati kao stanica sastavljena od tela (soma), velikog broja dendrita i aksona. Šematski prikaz biološkog neurona dat je na slici 4.16.



Slika 4.15: Biološki neuron, na osnovu: *The Biological Basis of Behavior*, http://cwx.prenhall.com/bookbind/pubbooks/morris5/chapter2/custom1/deluxecontent

Telo biološkog neurona ima nukleon koji sadrži informaciju o nasleđenim obeležijima i plazmu koja omogućava produkovanje signala potrebnih neuronu, dok se akson može prikazati kao tanka cevčica čiji je jedan kraj povezan na telo neurona, a drugi se deli na niz grana. Krajevi ovih grana završavaju se malim zadebljanjima koja najčešće dodiruje dendrite, a ređe telo neurona. Sinapsa se naziva mali razmak između završetka aksona prethodnog neurona i dendrita ili tela sledećeg neurona. Kroz dendrite neuron prima impulse od ostalih neurona, a signale koje proizvodi telo predaje preko aksona. Kod biološkog neurona funkcija aksona je da formira sinaptičke veze s drugim neuronima. Telo neurona generiše impulse koji putuju kroz akson do sinapsi i ti signali dolaze do dendrita zavisno o sinaptičkom prenosu koji je uslovljen većim brojem faktora, između ostalih i ranijim sinaptičkim prenosima. Na određen način, sinapse predstavljaju memorijske članove biološke neuronske mreže. Signali koji dođu do tela drugog neurona,

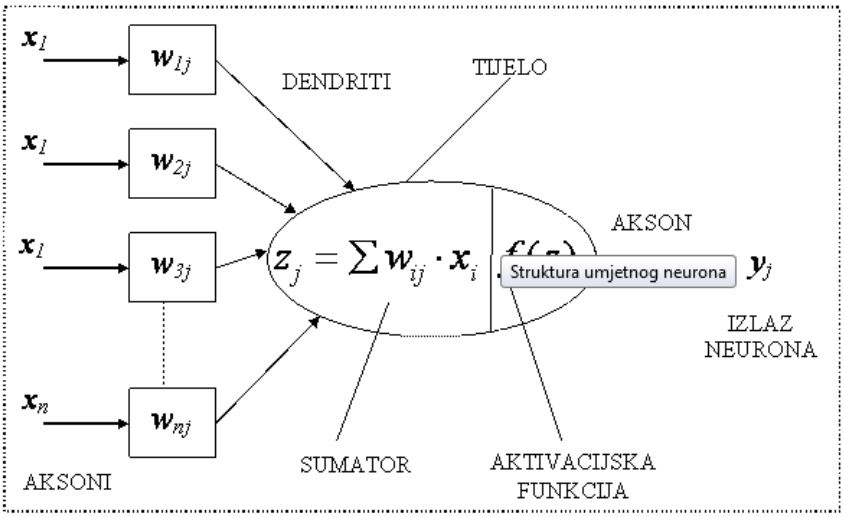
mogu biti pobuđujući ili smirujući. Ako je zbir signala pobuđujući, telo neurona će generisati impulse prema drugima neuronima. Možemo zaključiti da se rad biološkog neurona odvija kroz dve operacije: sinaptička operacija (pridodavanje važnosti ulaznim signalima u neuron) i somatska operacija („sabiranje“ ulaznih signala i generisanje impulsa zavisno o rezultatu). Dati prikaz biološkog neurona uopšteno prikazuje njegov rad i postavlja okvir u kojem su se razvili prvi modeli veštačkih neurona.



Slika 4.16: Šematski prikaz biološkog neurona [Vranješ, 2003]

Neuronska mreža (eng. *neural network*) je skup veštačkih neurona koji su međusobno povezani i interaktivni kroz operacije obrade signala. Ova mreža je uređena po uzoru na rad ljudskog mozga. Za veštački neuron se može reći da idejom oponaša osnovne funkcije biološkog neurona, gde se telo biološkog neurona zamenjuje sumatorom, ulogu dendrita preuzimaju ulazi u sumator, izlaz sumatora je akson veštačkog neurona, a uloga praga osetljivosti bioloških neurona preslikava se na tzv. aktivacijske funkcije.

Na slici 4.17. predstavljena je struktura veštačkog neurona. Funkcijske sinaptičke veze biološkog neurona sa njegovom okolinom preslikavaju se na težinske faktore, preko kojih se i ostvaruje veza veštačkog neurona sa njegovom okolinom.



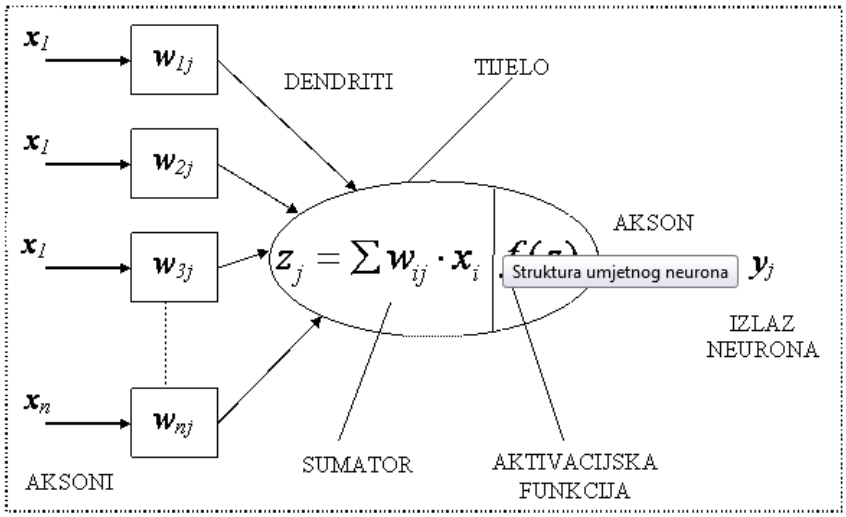
Slika 4.17: Veštački neuron, na osnovu *Umjetne neuronske mreže*, http://www.tsrb.hr/meha/index.php?option=com_content&task=view&id=14&Itemid

4.6.2. Veštački modeli neurona

Veštačke modele neurona moguće je razvrstati u dve osnovne grupe: statičke i dinamičke modele neurona.

4.6.2.1 Statički modeli neurona

Prvi model neurona razvili su McCulloch i Pitts i on je obrađivao signale pomoću prethodno navedenih operacija, somatske i sinaptičke. Ovaj jednostavni model neurona zove se perceptron. Sinaptičku operaciju predstavlja množenje svakog ulaznog signala x_i sa težinskim koeficijentom w_i. Sumiranjem tako otežanih signala i upoređivanjem zbira sa pragom osetljivosti neurona w_{n+1} predstavlja somatsku operaciju. U slučaju kada je zbir veći od praga osetljivosti aktivacijska funkcija ψ generiše izlazni signal y iznosa 1, a ako je zbir manji, generiše se izlazni signal iznosa 0. Značajnu teoremu o učenju perceptrona koja glasi: *perceptron može naučiti sve što može predstaviti,* dokazao je Rosenblat 1962. godine. Predstavljanje je sposobnost aproksimiranja određene funkcije, a učenje postupak koji podešavanjem parametara mreže postiže to da ona postane zadovoljavajuća aproksimacija određene funkcije. Šematski prikaz perceptrona dat je na slici 4.18.



Slika 4.18: Šematski prikaz perceptrona [Vranješ, 2003]

Matematički opis perceptrona dat je sledećim izrazima:

$$v(t) = \sum_{i=1}^{n} w_i(t) \cdot x_i(t) - w_{n+1}, \tag{4.34}$$

$$y(t) = \psi(v), \tag{4.35}$$

gde je:

- $\mathbf{x}_u(t) = [x_1(t), \ldots, x_n(t)]$ - vektor ulaznih signala neurona, pobudni vektor;
- $\mathbf{w}_s(t) = [w_1(t), \ldots, w_n(t)]$ - vektor sinaptičkih težinskih koeficijenata;
- w_{n+1} - prag osetljivosti neurona;
- $v(t)$ - izlaz operacije konfluencije – mera sličnosti ulaznih signala sa sinaptičkim koeficijentima;
- $\psi(v)$ - aktivacijska funkcija;
- $y(t)$ - izlaz neurona.

Kada se vektor ulaza proširi članom x_{n+1}=1, tada izraz (4.36) možemo napisati na sledeći način:

$$v(t) = \sum_{i=1}^{n+1} w_i(t) \cdot x_i(t) = \mathbf{w}^T(t)\mathbf{x}(t), \tag{4.36}$$

gde su:

- $\mathbf{x}(t) = [x_1(t), \ldots, x_n(t), x_{n+1}(t)]$-proširenи vektor ulaznih signala neurona;

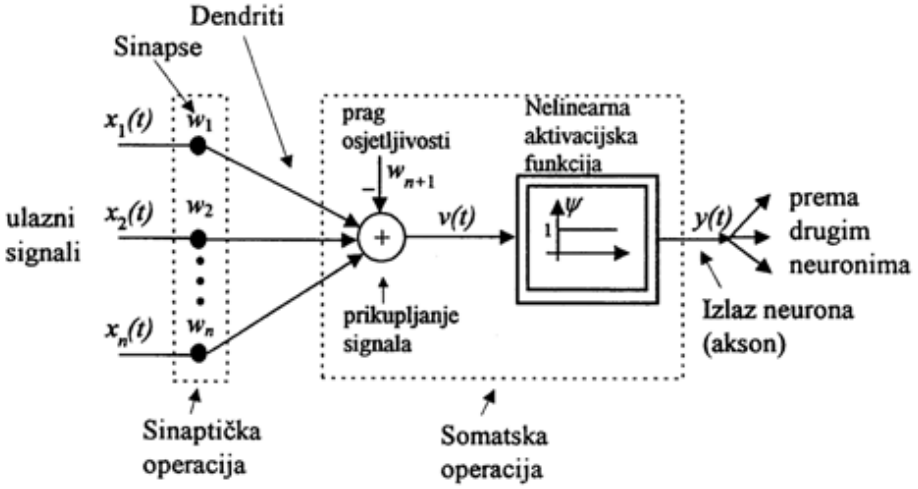
- $\mathbf{w}(t) = [w_1(t), \ldots, w_n(t), w_{n+1}(t)]$ - vektor sinaptičkih težinskih koeficijenata proširen pragom osetljivosti neurona.

Sa matematičkog gledišta, veštački neuron ima dve operacije: operaciju konfluencije (4.36) i aktivacijsku funkciju (4.35). Sa biološkog gledišta, operacija konfluencije predstavlja dodeljivanje težine ulaznim signalima $x(t)$ zavisno o akumuliranom znanju u sinapsama $w(t)$, dok sa matematičkog gledišta, operacija konfluencije predstavlja skalarni proizvod vektora $x(t)$ i $w(t)$. Meru sličnosti između proširenog ulaznog vektora $x(t)$ i vektora težinskih koeficijenata $w(t)$ predstavlja izlaz operacije konfluencije. Kao operaciju konfluencije, većina neuronskih mreža ima u sebi skalarni proizvod, ali ne i RBF neuronske mreže kod kojih se umesto skalarnog proizvoda koristi Euklidsko rastojanje između vektora $x(t)$ i $w(t)$. Preslikavanje izlazne vrednosti operacije konfluencije $v(t)$ u izlazni signal neurona $y(t)$ ograničen unutar [0,1] za unipolarne i [-1,1] za bipolarne signale, vrši aktivacijska funkcija $\psi(v)$.

Zbog toga što statički neuron ne sadrži dinamičke članove, njegov izlaz zavisi samo o trenutnim vrednostima ulaznih signala i težinskim koeficijentima, što ga čini strukturno stabilnim.

4.6.2.2. Dinamički model neurona

Kod dinamičkog modela neurona izlaz zavisi osim o trenutnim vrednostima ulaza i težinskim koeficijentima sinaptičkih veza, i o prošlim stanjima sinaptičkih veza. Dinamički model neurona omogućava prostiranje signala i unapred, ali i unazad preko unutrašnjih povratnih veza. Na slici 4.19. prikazan je uopšteni dinamički model neurona, koji se sastoji od operacije konfluencije, diskretnog dinamičkog člana drugog reda, nelinearne aktivacijske funkcije promenjivog ugla i povratnog signala sa izlaza neurona.



Slika 4.19: Uopšteni dinamički model neurona [Vranješ, 2003]

Uopšteni model dinamičkog neurona možemo prikazati sledećim izrazima:

$$v_1(k) = \sum_{i=1}^{n+1} w_i(k) \cdot x_i(k) \tag{4.37}$$

$$v(k) = a_0 v_1(k) + a_1 v_1(k-1) + a_2 v_1(k-2) + c_1 a_0 y(k-1) + c_1 a_1 y(k-2) + c_1 a_2 y(k-3) + c_1 a_2 y(k-3) - \frac{b_1}{a_2} v(k-1) - \frac{b_2}{a_2} v(k-2) \tag{4.38}$$

$$y(k) = \psi\left[g_a \cdot v(k)\right] \tag{4.39}$$

gde je *k* oznaka diskretnog vremena. Ako u uopštenom modelu dinamičkog neurona nekim parametrima pridružimo nepromenjive vrednosti, onda dobijamo modele neurona često korišćenih neuronskih mreža:

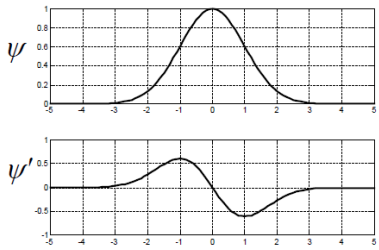
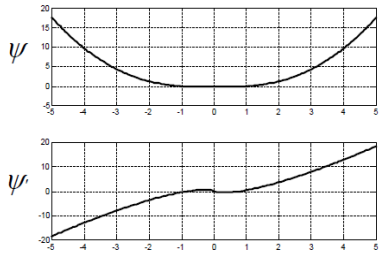
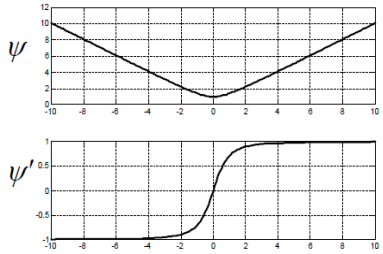
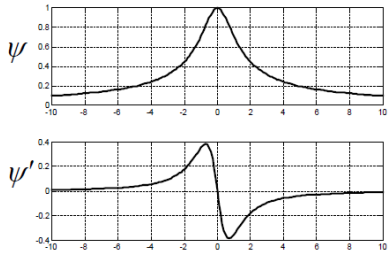
- MLP mreže: $a_0=1, a_1=a_2=b_1=b_2=c_1=0, g_a=1$;
- povratne neuronske mreže: $a_0=1, a_1=a_2=b_1=b_2=0, g_a=1$;
- neuronske mreže sa vremenskim kašnjenjem: $b_1=b_2=c_1=0, g_a=1$;
- dinamičke neuronske mreže: $c_1=0$.

4.6.3. RBF mreže

Ove neuronske mreže su dvoslojne statičke neuronske mreže, gde nulti (ulazni) sloj prosleđuje ulaze u mrežu na ulaz prvog sloja sačinjenog od neurona sa aktivacijskim funkcijama sa kružnom osnovicom i predstavlja njeno tzv. receptivno polje.

Drugi sloj je sloj mreže, koji je ujedno i njen izlazni sloj, i sastoji se od perceptrona sa linearnom aktivacijskom funkcijom jediničnog aktivacijskog pojačanja. Aktivacijske funkcije u prvom sloju RBF mreže koje su najčešće korišćene prikazane su u tabeli 4.1.

Tabela 4.1. Najčešće korišćene aktivacijske funkcije kod RBF modela [Petrović, 2009]

Naziv funkcije	Izraz za funkciju i njenu derivaciju	Grafički prikaz funkcije i derivacije
***Gauss*-ova funkcija**	$\psi(v) = e^{\frac{v^2}{2\sigma^2}}$ $\psi'(v) = -\frac{v}{\sigma^2} e^{\frac{v^2}{2\sigma^2}}$	
***Thin-plate-splin* funkcija**	$\psi(v) = v^2 \ln(v)$ $\psi'(v) = 2v \ln(v) +$	
Višekvadratna funkcija	$\psi(v) = \sqrt{v^2 + \sigma^2}$ $\psi'(v) = \frac{v}{\sqrt{v^2 + \sigma^2}}$	
Inverzna višekvadratna funkcija	$\psi(v) = \frac{1}{\sqrt{v^2 + \sigma^2}}$ $\psi'(v) = \frac{v}{(v^2 + \sigma^2)}$	
Napomena: $v \geq 0, \sigma > 0;$ *U primerima* $\sigma = 1.$		

RBF mreža ima sposobnost aproksimacije proizvoljne kontinuirane nelinearne funkcije, i njena aproksimaciona sposobnost određena je položajem središta RBF neurona, varijacijom aktivacijskih funkcija, kao i iznosima težinskih koeficijenata izlaznog sloja mreže. Algoritmima učenja se izračunavaju odgovarajuće vrednosti ovih parametara RBF mreže. RBF neuronske mreže se posebno koriste u slučaju aproksimacija jednostavnih i vremenski malo promenljivih nelinearnosti kada je moguće unapred na odgovarajući način rasporediti središta i odrediti iznose varijanse RBF neurona, a učenje mreže se može svesti samo na podešavanja težinskih koeficijenata izlaznog sloja. Ponašanje RBF neuronskih meža, u ovom slučaju, postaje linearno zavisno o parametrima.

Svojstva RBF mreže značajno određuje raspored središta RBF neurona. RBF funkcije se tradicionalno koriste za interpolaciju nelinearnih više varijabilnih funkcija, pri čemu je broj središta jednak broju podataka, tako da se u svaki ulazni podatak postavlja po jedno središte. Aproksimaciju proizvoljne nelinearne kontinuirane funkcije moguće je postići i sa manjim brojem dobro raspoređenih središta.

U svojim radovima Broomhead i Lowe [Broomhead i Lowe, 1988] su predložili da se središta postave u slučajno odabrane ulazne podatke. Postoji i mogućnost jednolikog rasporeda središta u prostoru ulaznih podataka. Varijanse aktivacijskih funkcija manje utiču na ponašanje mreže i obično se izaberu kao drugi koren proizvoda rastojanja neurona od dva najbliža susedna neurona prema Moody i Darken, 1989. godine [Moody i Darken, 1989]. Ove mreže i sa slučajnim ravnomernim rasporedom središta RBF neurona mogu aproksimirati proizvoljnu kontinuiranu nelinearnu funkciju, ali potrebni broj RBF neurona može biti jako veliki. Proširenjem postupka učenja mreže i na podešavanje položaja središta možemo postići smanjenje broja RBF neurona. Ponašanje RBF mreže, u ovom slučaju, postaje nelinearno zavisno o parametrima, ali i sa uporedivim aproksimacijskim svojstvima [Petrović, 2009].

Tabela 4.1. prikazuje najčešće korišćene aktivacijske funkcije kod RBF modela. U ovom radu koriste se aktivacijske funkcije sa *Gauss*-ovom funkcijom i ovakve RBF neuronske mreže još se nazivaju *Gauss*-ove RBF neuronske mreže.

4.6.4. Trening RBF mreža

Optimalna arhitektura RBF mreže se obično određuje eksperimentalno, ali neke praktične smernice postoje. Procedura rešavanja problema pomoću neuronskih mreža se sastoji od: prikupljanja i pripreme podataka, treninga mreže, testiranja mreže, i određivanja optimalnih parametara mreže i treninga eksperimentalnim putem (broj neurona, broj slojeva neurona, parametri algoritma za učenje i podaci za trening).

Priprema podataka za RBF mreže obuhvata: filtriranje, normalizaciju i redukciju dimenzionalnosti. Uspeh rešavanja u potpunosti zavisi od podataka koji se koriste za trening mreže. Potrebno je voditi računa o teorijskoj opravdanosti – reprezentativnosti korišćenih podataka za određeni problem. Ovo je vrlo specifično u zavisnosti od problema koji se rešava. Trening RBF mreže obuhvata:

određivanje optimalnih parametara mreže i algoritma za trening, određivanje broja skrivenih slojeva i broja neurona u svakom sloju (više ne znači bolje, cilj je imati što manje), dinamičko podešavanje parametara, validaciju parametara (sa probnim skupom), određivanje trening i test skupa podataka i rešavanje problema pretreniravanja i generalizacije.

Trening izlaznih težina (eng. *outputs weights*) je jednostavan kada izlazni neuroni koriste linearnu aktivaciju. U RBF mreži postoje tri vrste parametara koje je potrebno da budu određeni za prilagođavanje mreže određenom zadatku: središnji vektori C_i, izlazne težine ω_i i RBF parametri širine β_i. U sekvencijalnom treningu težine se ažuriraju u svakom vremenskom koraku. Za neke zadatke ima smisla definisati funkciju cilja i odabrati vrednosti parametara koje minimiziraju njenu vrednost. Najčešća ciljna funkcija je najmanja kvadrata funkcija, koja eksplicitno uključuje zavisnosti od težina.

$$K(\omega) \cong \sum_{t=1}^{\omega} K_t(\omega) \tag{4.40}$$

gde

$$K_t(\omega) \cong [y(t) - \varphi(x(t), \omega)]^2. \tag{4.41}$$

Minimizacija najmanje kvadratne ciljne funkcije uz pomoć optimalnog izbora težina optimizuje tačnost.

Postoje situacije u kojima više ciljeva, poput glatkoće i tačnosti, mora biti optimiziran. U tom slučaju je korisno optimizovati regulisanu ciljnu funkciju kao

$$H(\omega) \cong K(\omega) + \lambda S(\omega) \cong \sum_{t=1}^{\omega} H_t(\omega) \tag{4.42}$$

gde $S(\omega) \cong \sum_{t=1}^{\omega} S_t(\omega)$ i $H_t(\omega) \cong K_t(\omega) + \lambda S_t(\omega)$ pri čemu optimizacija *S* maksimizira glatkoću i λ .

4.6.5. Svojstva klasifikacije neuronskim mrežama

Na težim klasifikacijskim problemima, klasifikacija neuronskim mrežama se pokazala vrlo dobrom upravo kod onih problema kod kojih je teško ili nemoguće koristiti klasične tehnike simboličkog učenja. Pored ovoga, neuronske mreže su dobro prilagođene klasifikaciji u uslovima šuma u podacima.

Ova mreža ima sposobnost aproksimacije proizvoljne nelinearne kontinuirane funkcije, pri čemu tri parametra određuju njenu aproksimacijsku sposobnost, a to su: položaj središta neurona, varijanse aktivacijskih funkcija neurona i težinski koeficijenti izlaznog sloja mreže.

Korišćenjem različitih algoritama učenja ovi parametri se podešavaju da bi se dobilo odgovarajuće ponašanje mreže. Ove mreže su posebno efikasne u slučajevima kada je moguće unapred rasporediti središta neurona i odrediti iznose varijansi RBF neurona, čime se učenje mreže svodi na podešavanje težinskih

koeficijenata izlaznog sloja. U ovom slučaju, ponašanje RBF neuronske mreže postaje linearno zavisno o parametrima, što je velika prednost, ali da bi se dobili kvalitetni rezultati na ovaj način, potreban je jako veliki broj neurona. Kako bi ovo izbegli, postupak učenja mreže proširuje se i na podešavanje središta i varijansi neurona RBF mreže, kako bi se znatno smanjio broj RBF neurona. Na ovaj način, ponašanje RBF mreže postaje nelinearno zavisno o parametrima.

Nedostatak neuronskih mreža je relativno spor i zahtevan proces indukcije modela u poređenju sa klasičnijim tehnikama, čak do nekoliko redova veličine [Quinlan, 1994].

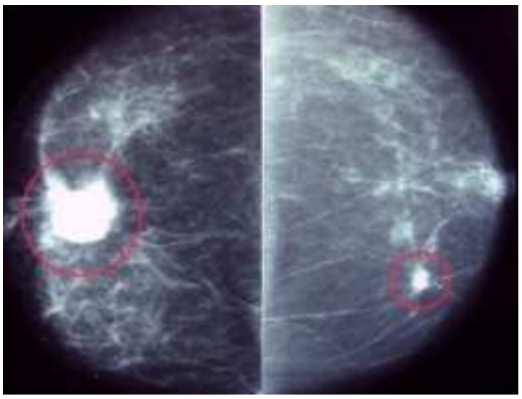
Još jedan važan nedostatak je i činjenica da klasifikacijski model reprezentovan neuronskom mrežom nije eksplicitno izražen, u obliku strukturnog opisa važnih odnosa među varijablama. Model je implicitan i skriva odnose varijabli u mrežnoj strukturi i velikom broju težinskih vrednosti, nije razumljiv ni podložan verifikaciji ili interpretaciji u okviru domena izvornog klasifikacijskog problema.

5. OPIS IZABRANIH PROBLEMA UČENJA

U petom delu biće dat prikaz izabranih problema učenja, koje ćemo u eksperimentalnom istraživanju koristiti za dokaz postavljenih hipoteza.

Za potrebe eksperimentalnog istraživanja koristili smo 15 realnih skupova podataka i 3 veštačka, preuzeta iz UCI repozitorijuma [Frank i Asuncion, 2010], koji je namenjen istraživačima koji pročavaju probleme veštačke inteligencije.

Rak dojke (breast cancer – bc): zadatak ovog seta podataka je da predvidi da li ima ili nema povratka bolesti raka dojke kod pacijenata. Predviđanje se radi na osnovu godina (pri čemu su pacijenti razvrstani po sledećim kategorijama godišta: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99), nastupanja menopauze (pre 40 godina, posle 40 godina, ili nije došlo do menopauze), veličine tumora (pri čemu je veličina tumora razvrstana u sledeće kategorije: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59), veličine čvorova (razvrstavanje je urađeno po sledećim kategorijama: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39), stepena maligniteta (stepen maligniteta je razvrstan u tri kategorije: 1, 2, 3), zahvaćene dojke tumorom (leva dojka, desna dojka), položaja tumora (levo dole, levo gore, desno dole, desno gore, centralno) da li je vršeno zračenje ili ne kod pacijenta.



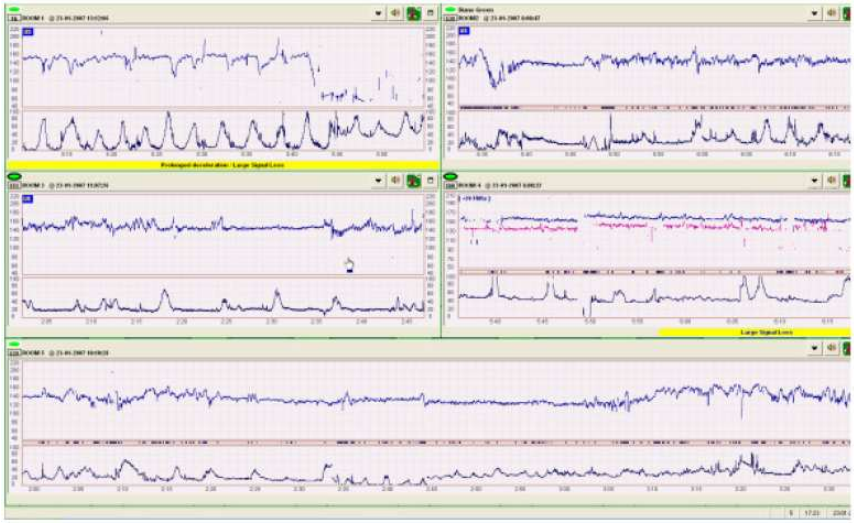
Slika 5.1: Mamografski snimci dojke [http://weinsteinimaging.com]

Na slici 5.1. prikazani su mamografski snimci dojke. U ovom setu podataka postoji 201 instanca jedne klase (nema povratka bolesti raka dojke) i 85 instanci druge klase (ima povratka bolesti raka dojke). Svaka instanca koja se odnosi na stanje jednog pacijenta je opisana sa 9 atributa. U ovom skupu podataka postoje vrednosti koje nedostaju, odnosno nepostoje vrednosti za sve atribute svih instanci.

Odobravanje kredita (credit approval - ca): ovaj set sadrži podatke koji se odnose na korišćenje kreditne kartice [Quinlan, 1987; Quinlan, 1993]. Kod ovog

seta podataka, svi atributi imena i vrednosti su promenjene u besmislene simbole kako bi se zaštitila tajnost podataka. Ovaj set podataka je interesantan za istraživanje jer postoji dobra mešavina atributa – kategoričkih i numeričkih vrednosti. Set podataka za odobravanje kredita sadrži 690 instanci, 15 atributa i dve klase čije su vrednosti odobriti kredit ili ga ne odobriti (jedna klasa je zastupljena sa 44.5%, a druga klasa je zastupljena sa 55.5%). U ovom setu podataka u 37 slučajeva (5% svih slučajeva) nedostaje jedna ili više vrednosti.

Kreditni podaci (Statlog german credit data - cg): ovaj skup podataka omogućava klasifikovanje potencijalnih korisnika kredita na one koji imaju mali ili visok rizik za odobravanje kredita. Ovo klasifikovanje se vrši na osnovu statusa postojećeg tekućeg računa i vremena kada je on otvoren, kreditne istorije (da li je korisnik do sada uzimao kredit i da li je bio redovan prilikom njegovog vraćanja), svrhe kredita (novi auto, postojeći automobil, nameštaj/oprema, radio/TV, kućni aparati, popravke, obrazovanje, odmor, prekvalifikacija, poslovni razlozi i drugi), iznosa kredita, štednog računa/obveznica, sadašnjeg zaposlenja (i ako nije zaposlen, koliko dugo je bez posla), vrednosti rate u odnosu na raspoloživ dohodak, osobni status (razveden, u braku, samac), pola, da li je korisnik dužnik/jemac po nekom drugom kreditu, sadašnje prebivalište (koliko dugo je državljanin) i svojstvo nekretnine, postojanje polise osiguranja života, starost u godinama, broja postojećih kredita u toj banci, radnog odnosa i vrste radnog odnosa i vrste posla koje obavlja (nekvalifikovan, kvalifikovan i visokokvalifikovan radnik), broja ljudi koji mogu garantovati za kredit i da li je korisnik kredita strani radnik ili ne. Ovaj skup podataka ima 1000 instanci i 20 atributa (7 numeričkih i 13 kategoričkih).



Slika 5.2: Fetalni kardiogram [Ayres-de-Campos *et al.,* 2008]

Ultrazvuk (cardiography – ct): ovaj skup podataka sastoji se od atributa merenja fetalnog otkucaja srca i atributa kontrakcije materice na ultrazvuku koje su klasifikovali doktori [Ayres de Campos *et al.,* 2000]. Slika 5.2. prikazuje fetalni

kardiogram. Fetalni kardiogrami su automatski obrađeni i odgovarajući dijagnostički pokazatelji su mereni. Fetalne kardiograme su razvrstali tri ekspert akušera i svakom od kardiograma dodeljena je određena klasa. Klasifikacija je urađena u odnosu na morfološke obrasce i na stanje fetusa. Ovaj set podataka sadrži 2126 instanci i 23 atributa. Atributi koji su posmatrani u ovom setu podataka se odnose na: brzinu otkucaja u minuti, broj ubrzanja u sekundi, broj fetalnih pokreta u sekundi, broj kontrakcija materice u sekundi, broj lakih usporenja u sekundi, broj teških usporenja u sekundi, broj produženih usporenja u sekundi, procenat vremena sa abnormalnim kratkoročnim varijabilnostima, srednju vrednost kratkotrajne varijabilnosti, procenat vremena sa abnormalnim dugoročnim varijabilnostima, srednju vrednost dugotrajne varijabilnosti, širinu histograma, minimum na histogramu, maksimum na histogramu, broj pikova na histogramu, broj nultih vrednosti na histogramu, mod histograma, srednju vrednost histograma, varijansu histograma i trend histograma. Set podataka se može koristiti u eksperimentima koji koriste 10 klasa (klase su razvrstane brojčano od 1 do 10) ili 3 klase (klase su razvrstane kao normalno, sumnjivo i patološko stanje).

Slika 5.3: Tkivo jetre i patološke promene na njemu usled prisustva hroničnog hepatitisa C [http://www.cpmc.org/advanced/liver/patients/topics/HepatitisC-profile.html]

Hepatitis (hepatitis – he): glavni cilj ovog skupa podataka je predvideti hoće li sa hepatitisom pacijenti umreti ili ne. Ovo predviđanje se vrši na osnovu stanja pacijenta i to: njegovog uzrasta (razvstano po klasama godišta: 10, 20, 30, 40, 50, 60, 70, 80), pola, korišćenja steroida (vrednosti mogu biti da ili ne), korišćenja antivirusnih lekova (vrednosti mogu biti da ili ne), postojanja umora (vrednosti mogu biti da ili ne), malaksalosti (vrednosti mogu biti da ili ne), anoreksije (vrednosti mogu biti da ili ne), veličine jetre (uvećana jetra ili ne) i oblika jetre, bolesti slezine, bilirubina (vrednosti su razvrstane u sledeće kategorije: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00), albumina (vrednosti su razvrstane u sledeće kategorije: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0), ALK fosfata (vrednosti su razvrstane u sledeće kategorije: 33, 80, 120, 160, 200, 250), histopatologije i sl. [Diaconis i Efron, 1983; Cestnik *et al*., 1987]. Slika 5.3. prikazuje tkivo jetre i patološke

promene na njemu usled prisustva hroničnog hepatitisa C. U ovom skupu podataka, postoje dve klase za predviđanje: prva klasa koja predviđa da će pacijent preživeti (123 instance) i druga klasa koja predviđa da će pacijent umreti (32 instance). Ovaj skup podataka sadrži 155 instanci i 19 atributa, sa vrednostima koje nedostaju za pojedine atribute.

Slika 5.4: Alkoholom oštećena jetra
[http://www.treatment4addiction.com/addiction/alcohol/liver-damage/]

Slika 5.5: Rentgenski snimak raka pluća
[http://medigalenic.blogspot.com/2009/12/lung-cancer-treatments.html]

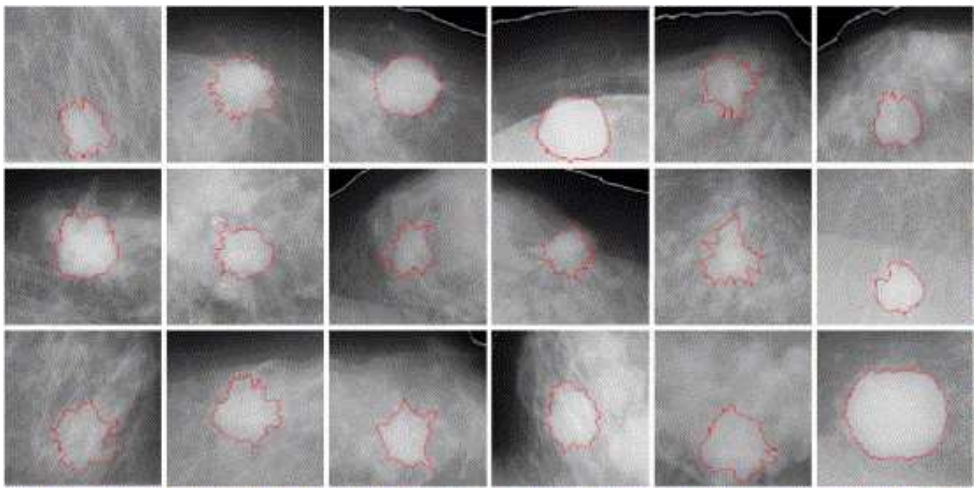
Jetra (liver disorders - li): u skupu podataka pod nazivom jetra, prvih pet atributa su testovi krvi pacijenata i to: zapremina eritrocita, alkalna fosfataza, alanine aminotransferaze, aspartat aminotransferaza, gama-glutamil transpeptidase; dok se druga dva atributa odnose na broj popijenih alkoholnih pića, i da li je pacijent svakodnevno pijan. Smatra se da ovi atributi ukazuju na bolesti jetre, koja bi mogla proizaći između ostalog i iz preteranog konzumiranja alkohola.

Slika 5.4. prikazuje oštećenu jetru usled preteranog konzumiranja alkohola. Svaka instanca u skupu podataka odnosi se na podatke jednog muškog pacijenta koji se podvrgao testu. U ovom skupu podataka postoje 345 instance i 6 atributa, bez vrednosti koje nedostaju za atribute.

Rak pluća (lung cancer –lc): set podataka za rak pluća sadrži podatke koji opisuju tri vrste patološkog oblika raka pluća. Rentgenski snimak raka pluća prikazan je na slici 5.5. Ove podatke su prvo koristili istraživači Hong i Young za ilustraciju dobrih performansi optimalno diskriminativnih ravni, čak i u loše datim postavkama [Hong i Yang, 1991].

Autori ne daju nikakvu informaciju o pojedinačnim varijablama, niti o tome gde su podaci izvorno korišćeni. Postoje 32 instance i 56 atributa, sa vrednostima koje nedostaju za pojedine atribute. Svi atributi u ovom skupu podataka su numerički i imaju celobrojne vrednosti od 0 do 3.

Mamografska masa (mammographic mass - ma): zadatak ovog skupa podataka je da predvidi ozbiljnost (benigni ili maligni) mamografskih lezija na osnovu BI-RADS atributa i starosti pacijenta [Elter *et al*., 2007]. Smatra se da je danas mamografija najefikasnija metoda za skrining raka dojke koja je dostupna. Međutim, niska pozitivna prediktivna vrednost biopsije dojke na osnovu interpretacije mamograma dovodi do približno 70 posto nepotrebnih biopsija sa benignim ishodom. Da bi se smanjio visok broj nepotrebnih biopsija dojke, predloženi su računarski programi koji treba da pomognu doktorima u odluci da li je neophodno obavljanje biopsije dojke kada postoje sumnjive lezije koje se vide na mamografskom snimku ili je potrebno samo pratiti pacijenta.



Slika 5.6: Izdvajanje konture na osnovu senki mamografske mase [Nakagawa *et al*., 2004]

Slika 5.6. prikazuje način izdvajanja kontura na osnovu senki mamografske mase. Ovaj skup podataka može se koristiti za procenu težine (benigne ili maligne) lezije na osnovu BI-RADS atributa i godišta pacijenta. Na Institutu za radiologiju Univerziteta Erlangen-Nürnberg između 2003 i 2006. godine prikupljeno je 516

benignih i 445 malignih masa koje su identifikovane putem digitalnih mamograma. Ovaj skup podataka sadrži sledeće atribute: starost pacijenta koja je izražena u godinama (celobrojna vrednost); posmatrani oblik mase koji može biti okarakterisan kao okrugao, ovalni, lobularni, nepravilni; potom margina mase koja može biti okarakterisana kao omeđana, sa mikro promenama, zamagljena, loše definisana, sumljiva; potom gustoća mase koja može biti visoka, srednja, niska i sa sadržajem masti; i ozbiljnost stanja pacijenta koje može biti benigno ili maligno. U skupu podataka, svaka instanca je povezana sa BI-RADS procenom koja se kreće u u rasponu od 1 (definitivno benigni) do 5 (vrlo sugestivna malignost) koja je dodeljena na osnovu procene dva radiologa. U ovom skupu podataka nedostaju vrednosti za pojedine atribute.

***MONK* problemi**: ovi problemi pripadaju klasi veštačkih (sintetičkih) domena, pri čemu svaki od tri problema koristi istu reprezentaciju podataka za upoređenje algoritama mašinskog učenja. *Monk* problemi su bili osnovni problemi koji su izučavani na prvoj međunarodnoj konferenciji posvećenoj uporednoj analizi različitih algoritama za učenje [Thrun *et al.*, 1991]. Jedna značajna karakteristika ovog poređenja je da je izvedena od strane više istraživača, od kojih je svaki bio zagovornik tehnike koju je testirao (pri čemu su istraživači često bili i kreatori tih metoda). U tom smislu, rezultati su manje pristrasni u poređenju sa rezultatima koje dobija jedna osoba koja uobičajeno zagovara određeni način učenja, i rezultati tačnije odražavaju problem generalizacije različitim tehnikama učenja. Ovaj skup podataka ima 432 instance i ima 7 atributa (6 atributa opisuje pojavu, dok je sedmi atribut jedinstveni simbol za svaku instancu, i njega ne uzimamo u razmatranje) i nema nedostajuće vrednosti za atribute. Za svaki problem, skup podataka je podeljen na trening i test set podataka. Skup sadrži podatke za primerke robota koji je opisan sa šest nominalnih obeležja:

$$\text{Oblik glave} \in \{\text{okrugla}, \text{kvadratna}, \text{osmougaona}\}$$
$$\text{Oblik tela} \in \{\text{okrugao}, \text{kvadratni}, \text{osmougaoni}\}$$
$$\text{Da li je nasmejan} \in \{\text{da}, \text{ne}\}$$
$$\text{Šta drži} \in \{\text{mač}, \text{balon}, \text{zastavu}\}$$
$$\text{Boja jakne} \in \{\text{crvena}, \text{žuta}, \text{zelena}, \text{plava}\}$$
$$\text{Da li ima kravatu} \in \{\text{da}, \text{ne}\}$$

Postoje tri *Monk* problema i svaki problem ima 432 instance.

Monk1 (m1): problem kod ovog seta podataka se može izraziti na sledeći način:

$$(\textit{oblik glave} = \textit{obliku tela}) \textit{ ili } (\textit{boja jakne} = \textit{crvena})$$

Ovaj problem je težak zbog interakcija između prva dva atributa. Možemo primetiti da je jedino vrednost boje jakne korisna.

Monk2 (m2): problem kod ovog seta podataka se može izraziti na sledeći način:

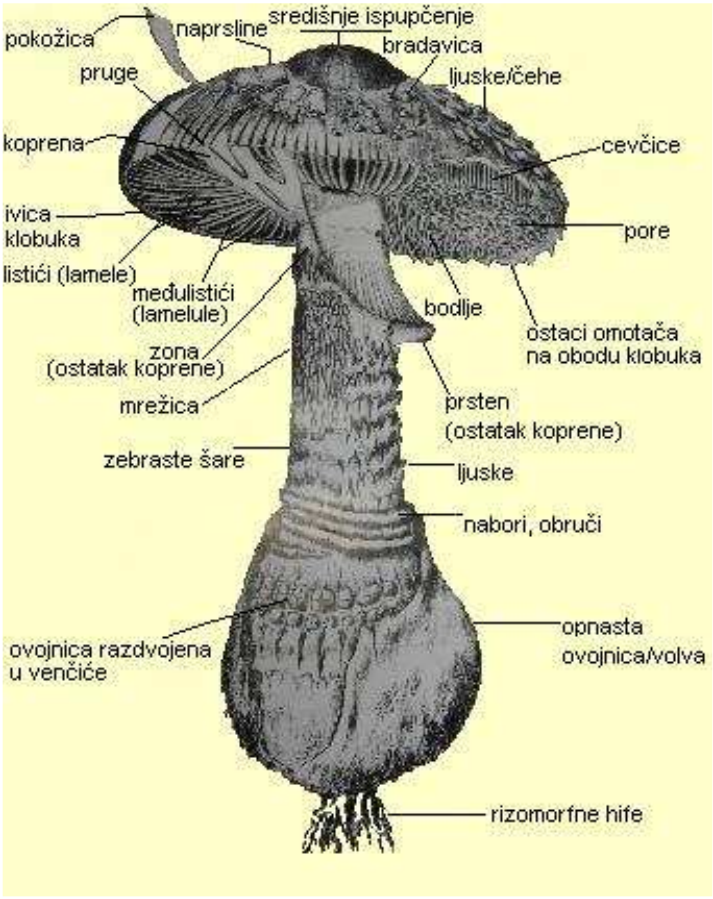
Tačno dva atributa imaju prvu vrednost koja je dodeljena svakom od atributa.

To znači da tačno dva iskaza za robota treba da budu tačna u skupu svih datih iskaza: {oblik glave je okrugao, oblik tela je okrugao, robot je nasmejan, robot drži mač, boja jakne je crvena i robot ima kravatu}. Problem je težak zbog pojava udvojenih interakcija atributa i činjenice da je samo jedna vrednost svakog atributa značajna. Možemo primetiti da je svih šest atributa relevantno za ovaj problem.

Monk3 (m3): problem kod ovog seta podataka se može izraziti na sledeći način:

$$(boja\ jakne = zelena\ i\ šta\ drži = mač\ ) \text{ ili}$$
$$(boja\ jakne \neq plave\ i\ oblik\ tela\ \neq osmougaoni)$$

Standardni trening set za ovaj problem ima 5% dodatog šuma. To je jedini *Monk* problem kome je dodat šum. Moguće je postići oko 97% tačnosti klasifikacije koristeći samo izraz: $boja\ jakne \neq plave\ i\ oblik\ tela\ \neq osmougaoni$.



Slika 5.7: Građa gljiva [http://www.pcelica.co.rs/gljive/gradja/gradja-gljive.php]

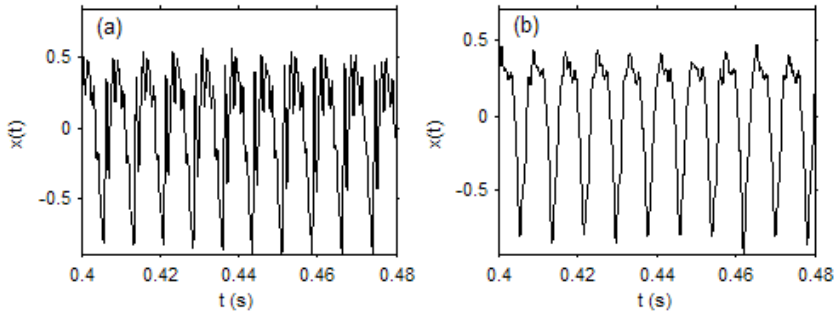

Slika 5.8: Gljive

Gljive (mushroom – mu): ovaj skup podataka uključuje opise hipotetičkih uzoraka koji odgovaraju 23 vrsti gljiva *Agaricus* i *Lepiota* familiji [Schlimmer, 1987]. Svaka vrsta je identifikovana kao definitivno jestiva, definitivno otrovna ili nepoznatog jestivog sastava i ne preporučuje se za jelo. Ne postoji jednostavno pravilo za određivanje jestivosti gljiva na osnovu njihovih karakteristika. Na slici 5.7. prikazana je građa gljiva, dok slika 5.8. prikazuje uzorke gljiva. Ovaj skup podataka ima 8124 instanci i 23 atributa.

Da li je gljiva otrovna ili ne, utvrđuje se na osnovu sledećih karakteristika: oblik klobuka (zvono, konus, konveksan, ravan, utonuo); površine klobuka (vlaknasta, sa kanalima, pokrivena ljuspama, glatka); boje klobuka (smeđa, boja kože, boja cimeta, siva, zelena, ružičasta, ljubičasta, crvena, bela, žuta); trusišta, tj. dela koji se nalazi na donjoj strani klobuka (na listićima, u unutrašnjosti cevčica, na unutrašnjoj površini, u unutrašnjosti plodnog tela); mirisa (bademа, anisa, smrada, ustajali miris, bez mirisa, opor miris); stručka (mesto gde je stručak pričvrćen za klobuk, da li je srastao sa drugim delovima, visine i debljine, spoljnog oblika, oblika donjeg dela stručka, površine stručka); boje spora (crna, smeđa, boja kože, čokolada, zelena, narančasta, ljubičasta, bela, žuta); kako su naseljene gljive (u izobilju, grupisano, brojno, razasute, svega nekoliko, osamljeno); i staništa (trava, lišće, livade, staze, urbano, na otpadu, šume). Sve vrednosti u ovom skupu podataka su kategoričke vrednosti. Kod nekih atributa postoje nedostajuće vrednosti.

Parkinson (Parkinson – pa): ovaj skup podataka je kreirao Max Little sa Univerziteta u Oksfordu, u saradnji sa Nacionalnim centrom za glas i govor, koji je smešten u Denveru, Kolorado. Originalna studija objavljena je u svrhu ekstrakcije atributa iz govornih signala kod osoba koje imaju govorni poremećaj. Slika 5.9. prikazuje dva primera govornog signala: (a) zdrave osobe, (b) osobe obolele od Parkinsa [Little *et al*., 2009]. Ovaj set podataka se sastoji od niza biomedicinskih merenja glasa kod 31 osobe, od toga 23 obolele od Parkinsonove bolesti [Little *et al*., 2007]. U ovom setu podataka postoji 195 instanci i 23 atributa. Atributi ovog seta podataka su: prosečna osnovna govorna frekvencija, najveća osnovna govorna frekvencija, minimalna osnovna govorna frekvencija, podrhtavanje, apsolutno podrhtavanje, nekoliko mera varijacije u fundamentalnoj frekvenciji, nekoliko mera varijacije u amplitudi, dve mere odnosa buke na tonskim komponentama u glasu, dve nelinearne dinamičke mere, signal fraktala i tri nelinearne mere osnovne varijacije frekvencije.

Svaka kolona u tabeli je osobena karakteristika glasa osobe, a svaki red odgovara jednom od 195 snimaka glasa određene osobe. Glavni cilj ovog skupa podataka je odvajanje zdravih ljudi od onih osoba koje su obolele od Parkinsa, na osnovu kolone „Status“ koja ima moguće vrednosti 0 za zdrave osobe i 1 za osobe sa Parkinsonovom bolešću.



Slika 5.9: Dva primera govornog signala: (a) zdrave osobe, (b) osobe

obolele od Parkinsa [Little *et al*., 2009]. Horizontalna osa predstavlja vreme u sekundama, na vertikalnoj osi je prikazana amplituda signala (bez jedinične mere)

Dijabetes (Pima Indijans dijabetes – pi): radi dijagnostifikovanja dijabetesa iz većeg skupa podataka izdvojeni su podaci za žene koje su starije od 21 godinu i pripadaju Pima Indijancima [Smith *et al*., 1988]. U ovom setu podataka dijagnostifikovano je da li pacijent pokazuje znakove dijabetesa prema kriterijima Svetske zdravstvene organizacije (tj. ako se otkrije tokom rutinske medicinske kontrole ili ako 2 sata nakon opterećenja glukoza je barem 200 mg/dl u svakom ispitivanju). Stanovništvo koje je učestvovalo u ovom istraživanju živi u neposrednoj blizini Phoenix-a, Arizona u SAD-u (slika 5.10).

Slika 5.10: Arizona Pima Indijanci [http://indiancountrytodaymedianetwork.com/article/mexico-vs.-arizona-pima-indians-3258]

U ovom skupu podataka postoji 768 instanci i 8 atributa koji imaju numeričke vrednosti. Atributi u ovom skupu podataka su: broj trudnoća, koncentracija glukoze na tašte i posle 2 sata u oralnom testu opterećenja glukozom, dijastolni krvni pritisak, debljina kožnog nabora nad tricepsom, da li se nakon 2 sata vrednost insulina vratila na početnu vrednost, indeks telesne mase, postojanje dijabetesa u porodici i starost. Skup podataka sadrži podatke o 500 osoba koje nisu dijabetičari i 268 koje su dijabetičari. Kod nekih instanci, za neke atribute postoje vrednosti koje nedostaju.

Segmentacija slike (image segmentation – se): slučajevi su izvučeni slučajnim izborom iz baze podataka 7 slika spoljnog okruženja [Piater *et al*., 1999]. Slike su ručno segmentirane kako bi se izvršila klasifikacija za svaki piksel. Svaka instanca u skupu podataka je 3x3 regija. U ovom skupu podataka ima 210 podataka za trening i 2100 test podataka. Skup podataka sadrži 19 numeričkih atributa, bez nedostajućih vrednosti za atribute. Klasa ovog skupa podataka ima moguće vrednosti: površina cigle, nebo, lišće, cement, prozor, put i trava. Skup podataka ima 30 slučajeva po klasi za obuku podataka i 300 slučajeva po klasi za testiranje podataka.

Osnovni sastavni elementi svih digitalnih slika su pikseli. Pikseli su mali susedni kvadrati u matrici preko dužine i širine digitalne slike. Svi pikseli u bilo kojoj digitalnoj slici su iste veličine. Pikseli su jednobojni, pri čemu je svaki piksel jedna boja koja je uklopljena iz neke kombinacije tri primarne boje crvena, zelena i plava. Segmentacija slike je proces klasifikovanja piksela slike u različite klase prema nekim unapred definisanim kriterijima. Stvaranje klasifikatora visokih performansi obično uključuje značajnu količinu ljudskog napora budući da se trening obavlja preko *off-line* rukom označenih piksela kao trening primera. Korisnost trenirajućeg seta je teško odrediti *a priori*, pa velike količine podataka obično moraju biti na raspolaganju.

Slika 5.11: Neobrađena slika [http://vis-www.cs.umass.edu/old/projects/itl/example.html]

Slika 5.12: Obrađena slika nakon piksel klasifikacije [http://vis-www.cs.umass.edu/old/projects/itl/example.html]

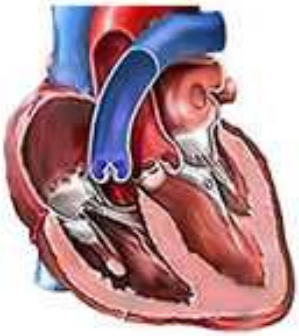
Predložena metoda piksel klasifikacije [Piater, 1999] omogućuje korisniku selekciju piksela na slici tako što će se reći programu koja je vrsta površine taj piksel i to sa klikom na dugme. Sistem tada reklasifikuje sliku i prikazuje njenu klasifikaciju korisniku. Ako je korisnik zadovoljan sa urađenim, posao je obavljen, a ako nije zadovoljan klasifikacijom, korisnik može kliknuti na područje slike koje je pogrešno klasifikovano i sistem će reklasifikovati sliku prema kliku. Kada je korisnik zadovoljan sa rezultatom, onda se klasifikacija može koristiti i na drugim slikama. U nastavku teksta dat je primer klasifikacije, tako što se obrada prethodne slike radi tako da se pikseli razvrstaju u 4 kategorije i to: crveni u krov, plavi u nebo, zeleni u

travu i žuti u ciglu. Neobrađena slika prikazana je na slici 5.11, dok je obrađena slika prikazana je na slici 5.12.

Slika 5.13: Različite bolesti soje
[http://www.agweb.com/article/Prevent_soybean_diseases_with_Headline_207165/]

Soja (soybean – so): zadatak je dijagnostifikovati bolesti u biljkama soje [Michalski i Chilausky, 1980]. U ovom skupu podataka postoji 307 primeraka opisanih sa 35 kategoričkih atributa. Vrednost atributa je merena posmatranjem svojstava lišća i različitih biljnih abnormalnosti. U setu podataka postoji 19 klasa za bolesti soje. Neke od bolesti soje su: plamenjača i bakteriozna pegavost koje su najčešće oboljenje lista, na stablu su najštetniji rak stabla i bela trulež, na korenu ugljenasta trulež, dok seme najčešće oboleva od truleži. Različite bolesti soje prikazane su na slici 5.13. Ovaj skup podataka za neke atribute ima nepoznate vrednosti.

Srce (Statlog heart - sh): zadatak je predvideti odsutnosti ili prisutnosti bolesti srca na osnovu starosti, pola, odgovarajućeg tipa bola u grudima, krvnog pritiska u mirovanju, nivoa holesterola i šećera u krvi, elektrokardiografskih rezultata, najvećeg broja otkucaja srca, promene pokazatelja prilikom napora i slično. Srčane bolesti su jedan od najčešćih razloga smrtnosti u svetu, a posledica su nezdravog načina života, pre svega neumerenosti u jelu i piću, slaboj telesnoj aktivnosti i visokom nivou stresa (slika 5.14). Ovaj set podataka sadrži 13 atributa (koji su izvađeni iz većeg skupa od 75 atributa). Klasa za ovaj set podataka ima dve vrednosti: odsustvo i prisustvo bolesti srca. Postoji 270 posmatranja, bez vrednosti koje nedostaju za pojedine atribute.



Slika 5.14: Srce [http://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)]

Slika 5.15: Glasanje kongresmena
[http://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records]

Glasanje kongresmena (congressional voting records – vo): u ovom setu podataka stranačku pripadnost američkog Predstavničkog doma karakteriše kako su kongresmeni glasali na 16 ključnih pitanja kao što su trošenje na obrazovanje i imigracija [Schlimmer, 1987]. U Americi Predstavnički dom je jedan od dva doma Kongresa SAD-a; drugi je Senat. U Domu svaka država je proporcionalno predstavljena prema udelu u ukupnom stanovništvu i ima pravo na najmanje jednog predstavnika; najmnogoljudnija država trenutno ima 53 predstavnika. U Domu ukupan broj zastupnika je 435 prema posebnom zakonu iz 1911. godine, iako Kongres može zakonom izmeniti taj broj. Svaki predstavnik ovog Doma služi mandat od dve godine. Slika 5.15. prikazuje glasanje kongresmena SAD-a.

Atributi koje ovaj skup podataka sadrži odnosi se na trošenje sredstava za: hendikepiranu decu, projekat podele troškova za vodu, zamrzavanje lekarskih taksi, pomoć verskim grupama u školi, razvoj satelita i raketa, pomoć imigraciji, smanjenje poreza korporacijama, obrazovanje, sudstvo, kazneno-popravne domove, kao i bescarinski izvoz. Klase u ovom skupu podataka imaju dve vrednosti: demokrate i republikanci. U ovom skupu podataka postoji 435 instanci (267 demokrati, 168 republikanci), svi atributi su binarni i postoje nedostajuće vrednosti.

Tabela 5.1. Prikaz setova podataka. Podrazumevana tačnost klasifikacija je tačnost predviđanja većinske klase na celom skupu podataka. Svi skupovi podataka su realni, osim *m1*, *m2* i *m3* koji su veštački. „CV“ označava 10-struku unakrsnu validaciju

Skup	Atributi			Broj klasa	Veličina za treniranje	Veličina za testiranje	Referentna tačnost
	ukupno	kategorički	numerički				
bc	9	9	0	2	286	CV	70.30
ca	15	9	6	2	690	CV	55.50
cg	20	13	7	2	1000	CV	50.10
ct	23	0	23	3	2126	CV	95.00
he	19	13	6	2	155	CV	78.10
li	6	0	6	2	345	CV	58.10
lc	56	0	56	3	32	CV	26.80
ma	5	0	5	2	961	CV	84.00
m1	6	6	0	2	124	308	50.00
m2	6	6	0	2	169	263	67.13
m3	6	6	0	2	122	310	52.78
mu	22	22	0	2	8124	CV	51.80
pa	23	0	23	2	195	CV	76.00
pi	8	0	8	2	768	CV	65.10
se	19	0	19	7	2310	CV	14.30
so	35	35	0	19	683	CV	13.47
sh	13	3	10	2	270	CV	55.00
vo	16	16	0	2	435	CV	61.40

U tabeli 5.1. prikazane su uporedne karakteristike posmatranih setova podataka. Postoji 18 skupova podataka, od toga 15 skupova podataka su realni skupovi, što znači da su dobijeni prikupljanjem podataka iz realnih sistema koji postoje. Ostala tri skupa podataka *m1*, *m2* i *m3* su veštački skupovi podataka, što znači da podaci nisu dobijeni iz realnog sistema, već su podatke kreirali istraživači za potrebe istraživanja. Da bi dobili referentne podatke tokom istraživanja u radu smo koristili i realne i veštačke skupove podataka za dokazivanje postavljenih hipoteza.

Pri tome, posmatran je ukupan broj atributa u svakom setu podataka, kao i broj onih atributa koji pripadaju kategoriji kategoričkih ili numeričkih atributa. Kod onih algoritama i metoda, čiji ulaz podataka ne podržava kategoričke ili numeričke atribute, vršena je odgovarajuća priprema podataka, pre same obrade podataka. Pet setova podataka ima više atributa od 20, i to *lc* sa 56, *so* sa 35, *pa* i *ct* sa 23 i *mu* sa 22. Najmanje atributa imaju setovi podataka *ma* sa 5, *li*, *m1*, *m2* i *m3* sa 6 atributa. Možemo zaključiti da se u posmatranim skupovima podataka nalaze i skupovi sa izuzetno velikim brojem atributa, kao i oni skupovi koji imaju mali broj atributa, što je dobro sa stanovišta istraživanja. Posmatrani skupovi podataka su balansirani jer postoje skupovi koji sadrže samo ili kategoričke ili numeričke atribute, kao i skupovi podataka koji sadrže i kategoričke i numeričke podatke.

Što se tiče broja klasa u posmatranim skupovima podataka, samo dva skupa podataka imaju veći broj klasa od 3, i to *se* koji ima 7 klasa i *so* koji ima 19 klasa. Razlog za ovo je činjenica, što se u najvećem broju slučaja u problemima klasifikacije razvrstavanje postojećih instanci vrši u dve, eventualno tri klase, a ređe u veći broj klasa.

U tabeli 5.1. vidimo da broj instanci predviđen za treniranje varira od malog broja prikupljenih istanci što je slučaj sa *lc* koji ima samo 32 istance do skupova koji imaju mnogo veći broj istanci kao što je npr. slučaj sa *mu* koji ima 8124 istanci za trening. Što se tiče veličine skupa za testiranje, inicijalno kod svih realnih skupova podataka, imali smo pripremljen jedan skup podataka, iz koga smo metodom 10-struke unakrsne validacije izdvajali podatke koji će služiti za testiranje. Istraživači koji su kreirali veštačke skupove podataka *m1*, *m2* i *m3* su odvojili podatke u dve grupe i to one koji će služiti za treniranje i one koji će služiti za testiranje, pri čemu je manji broj podataka korišćen za trening (u proseku oko 25%), a veći deo služi za testiranje tačnosti klasifikacije. U poslednjoj koloni tabele prikazana je referentna tačnost za realne i veštačke skupove podataka.

6. REZULTATI UČENJA I ESTIMACIJA PERFORMANSI NAUČENOG ZNANJA

U šestom delu rada, biće reči o metodologiji izvođenja eksperimenta i podešavanju parametara modela. Biće razmatrana tačnost i preciznost kojima merimo uspešnost dobijenog modela, kao i statistički testovi koje koristimo u istraživanjima, sa posebnim osvrtom na standardnu devijaciju i *t*-test.

6.1. Opis metodologije izvođenja eksperimenta

Eksperiment je rađen uz pomoć WEKA (Waikato Environment for Knowledge Analysis), alata za pripremu i istraživanje podataka razvijen na Waikato Univerzitetu na Novom Zelandu. Ovaj alat poseduje podršku za ceo proces istraživanja počevši od pripreme podataka preko procene i korišćenja različitih algoritama.

WEKA je napisana u Javi i distribuira se pod GNU General Public Licence. Ovaj alat radi na skoro svim platformama i tesitran je na Linux, Windows i Macintosh operativnim sistemima. Verzija koja je korišćena je poslednja stabilna verzija WEKA-e 3.6 i može se preuzeti sa *web* adrese http://www.cs.waikato.ac.nz/ml/weka/index.html. U ovom alatu implementirane su najčešće metode koje se javljaju u istraživanju podataka, a to su: klasifikacija, regresija, klasterovanje, asocijacija i izbor atributa. Većina implementiranih metoda omogućava podešavanje parametara prema konkretnom problemu i podacima. WEKA poseduje čak 49 metoda za pripremu podataka. Ovi podaci mogu biti učitani u nekoliko različitih formata datoteka. WEKA podržava razne formate datoteka, ali je prepručljivo koristiti ARFF format datoteke koji je osnovni podržani format. U našem istraživanju, korišćen je ARFF format, a podaci koji nisu izvorno u tom formatu, konvertovani su naknadno u ARFF format. Ovaj alat podržava i ostale formate, a to su: CVS, C4.5 ili binarni. Alat omogućava da se podaci takođe preuzmu sa URL adrese ili iz SQL baze podatka.

WEKA ima tri različita grafička korisnička okruženja: *Explorer*, *Knowledge Flow* i *Experimenter*. Od ovih okruženja, najlakši način za korišćenje WEKA-e je *Explorer*. *Explorer* omogućava laku i efikasnu primenu svih funkcionalnosti alata putem izbornih menija. Korisničko okruženje *Knowledge Flow*, omogućava korisniku da samostalno definiše sekvencijalnu obradu podatka. Glavni nedostatak *Explorer*-a je činjenica da sve podatke čuva u glavnoj memoriji – po otvaranju datoteke ili baze podataka ceo sadržaj se učitava odmah, što dovodi do toga da je primena moguća samo na srednjim i malim bazama podatka. Ovo okruženje omogućava precizno definisanje obrade podatka povezivanjem komponenti koje predstavljaju izvore podataka, metode za pripremu, algoritme, metode evaulacije i grafički prikaz. Ako algoritmi imaju mogućnosti inkrementalnog učenja, podaci će biti učitani i obrađivani sekvencijalno. Poslednje okruženje, *Experimenter*,

osmišljeno je da pomogne korisniku da odgovori na osnovno pitanje pri upotrebi klasifikacije i regresije: koje metode i parametre koristiti za dati problem. *Experimenter* omogućava poređenje različitih klasifikatora i filtera sa različitim parametrima. Poređenje može da se realizuje i kroz *Explorer* ali *Experimenter* nudi automatizaciju celog procesa, testove ispravnosti i performansi sistema. Takođe, iza svih ovih grafičkih okruženja nalazi se osnovna funkcionalnost WEKA-e, kojima se može pristupiti i iz komandne linije. Za probleme koje smo mi izučavali, problem redukcije dimenzionalnosti podataka, korišćena su korisnička okruženja *Explorer* i *Experimenter*.

Glavne prednosti ovog alata su širok spektar metoda za pripremu podataka, izbor atributa i algoritama integrisanih u jednom alatu. Takođe, bilo koja od metoda koja je implementirana u alatu WEKA može biti pozivana iz korisničkog koda, što za posledicu ima olakšan razvoj novih aplikacija za istaživanje podataka uz minimum dodatnog kodiranja. Ovaj alat je potpuno besplatan, jednostavno i lako se instalira na svakoj platformi, a GUI ga čini jednostavnim za korišćenje.

Nedostatak WEKA-e je dokumentacija, jer WEKA konstantno raste i dokumentacija daje samo listu raspoloživih algoritama. Posebno je dokumentacija za grafičko korisničko okruženje ograničena. Skalabilnost je drugi mogući problem pri radu sa WEKA-om, jer se tokom rada sa velikim količinama podataka vreme za obradu drastično povećava. Takođe, nedostatak je i činjenica da u GUI okruženju nisu implementirane sve funkcionalnosti WEKA-e pa je u radu neke opcije potrebno pozivati iz komandne linije.

Prilikom traženja modela koji najbolje aproksimira ciljnu funkciju, potrebno je dati i mere kvaliteta modela, odnosno učenja. Različite mere se mogu koristiti u zavisnosti od vrste problema, ali u slučaju problema klasifikacije se obično koristi preciznost, odnosno broj tačno klasifikovanih instanci podeljen ukupnim brojem instanci. U našim eksperimentalnim istraživanjima koristili smo tačnost klasifikacije kao meru kvaliteta modela.

Da bi dobili pouzdaniji način evaluacije naučenog znanja koristili smo tzv. unakrsnu validaciju, gde smo ceo skup podataka kojim smo raspolagali delili na *n* približno jednakih podskupova. Pri tome smo jedan podskup izdvajali i trening vršili na ostalih *n-1* podskupova, a nakon treninga, kvalitet naučenog znanja ocenjivali na izdvojenom podskupu. Opisani postupak smo ponavljali za sve ostale izdvojene podskupove i kao finalnu ocenu kvaliteta uzimali prosek dobijenih ocena za svaki od podskupova. U našem eksperimentalnom istraživanju smo za vrednost *n* uzimali broj 10. Unakrsnu validaciju smo koristili u našem eksperimentalnom istraživanju, jer opisani postupak daje stabilniju ocenu kvaliteta, a prednost ovog metoda je i da se u svakom od *n* koraka unakrsne validacije koristi velika količina podataka pri treniranju, a sve raspoložive instance u jednom trenutku su iskorišćene za testiranje.

Za klasifikaciju, za sve realne skupove podataka, korišćena je 10-struka unakrsna validacija, koja je pri tome bila uvek ponovljena 10 puta. Za veštačke skupove podataka *m1*, *m2* i *m3*, s obzirom da smo inicijalno imali odvojene test i trening podatke, uradili smo spajanje podataka test i trening skupa, vodeći računa da pri treniranju koristimo prvih 22.3% podataka za *m1* problem, za *m2* problem prvih 28.1% podataka, za *m3* problem prvih 22.0% podataka, kako bi organizovali eksperiment kako je on originalno zamišljen. Na ovaj način dobijamo uporedivost

rezultata sa ostalim istraživačima koji su koristili ove setove podataka. Znači u novom skupu podataka koji smo pripremili za eksperiment, na početku skupa smeštamo trening podatke, pa onda u nastavku skupa test podatke. Podatke koji su inicijalno u *arff* formatu smo prebacili u *csv* format, izvršili spajanje i ponovo vratili u *arff* format. Tokom eksperimentalnih istraživanja najpre smo koristili podatke iz novo dobijenog seta za trening (prvih 22.3% podataka za *m1* problem, za *m2* problem prvih 28.1% podataka, za *m3* problem prvih 22.0% podataka), pa onda za testiranje, vodeći računa da ne radimo slučajni izbor podataka za trening i testiranje. Ceo eksperiment smo ponovili 10 puta.

Za neke algoritme učenja je neophodno da sve vrednosti postoje za sve atribute u svim instancama. U našem slučaju, za SVM algoritam je bilo neophodno da postoje sve vrednosti svih atributa. S obzirom da su postojali setovi podataka sa nedostajućim vrednostima, da bi mogli da koristimo SVM algoritam, bilo je neophodno zameniti nedostajuće vrednosti sa procenjenim vrednostima za dati skup. Ovu zamenu smo radili kod svih setova podataka koji imaju nedostajuće vrednosti, pre korišćenja SVM algoritma. Ostali algoritmi učenja su mogli da se sami izbore sa nedostajućim vrednostima za pojedine atribute u nekim od istanci.

U eksperimenalnom istraživanju koristili smo filter metode, metode prethodnog učenja i ekstrakciju atributa radi smanjenja dimenzionalnosti podataka. Eksperimentalnim istraživanjem nisu obuhvaćene ugrađene metode, jer ove metode vrše selekciju atributa u sklopu osnovnog algoritma induktivnog učenja, odnosno kao deo procesa generalizacije. Zbog toga nije moguće vršiti uporedne analize efekata redukcije dimenzionalnosti podataka kod ovih metoda. Za razliku od ovih metoda, metode filtriranja, prethodnog učenja i ekstrakcije atributa razmatraju selekciju atributa kao spoljašnji sloj procesa indukcije.

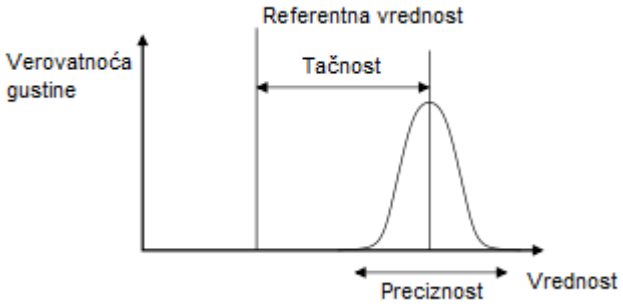
U mnogim slučajevima trening skupovi su oskudni i postoje međusobne interakcije atributa. Izbor optimalnog podskupa atributa vrši se različitim estimacijama, koje se zasnivaju na različitim statističkim pretpostavkama, npr. nezavisnost atributa i dovoljan broj trening primera, koje nisu uvek zadovoljene. To je razlog zbog čega ugrađene metode selekcije atributa, nisu uvek dovoljne, pa se u mnogim praktičnim situacijama koriste metode prethodne selekcije atributa kako bi se performanse poboljšale.

Za svaku metodu koja je korišćena u svrhu smanjenja dimenzionalnosti podataka korišćen je skup mogućih rešenja, koji je potom bio propušten kroz klasifikatore IBk, *Naïve Bayes*, SVM, J48 i RBF mreže. U svim eksperimentima, izabrano je ono rešenje za broj atributa koji će se dalje koristiti u istraživanju, koji daje najveću tačnost klasifikacije.

U rezulatatima istraživanja, kada upoređujemo par algoritama, predstavićemo rezultate tačnosti klasifikacije za svaki algoritam na svakom skupu podataka. Važno je napomenuti da kad smo koristili 10-struku unakrsnu validaciju za ocenu tačnosti i preciznosti, da je unakrsna validacija nezavisna spoljna petlja, ne ista kao i unutrašnja 5-struka unakrsna validacija koja je deo algoritma za izbor atributa kod metode prethodnog učenja.

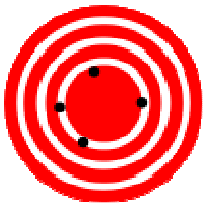
Koristili smo 5-struku unakrsnu validaciju koja je deo algoritma za izbor atributa kod metode prethodnog učenja, jer se na taj način izbegava višestruka unakrsna validacija za velike skupove podataka, kako bi izbegli veliko zahtevano vreme za obradu podataka. Sa druge strane nismo koristili manje vrednosti za

unakrsnu validaciju od 5, jer se u praksi pokazalo da je za male skupove podataka potrebno uraditi više puta unakrsnu validaciju kako bi se prevazišao problem visoke varijanse koji je rezultat male količine podataka za obradu.



Slika 6.1: Tačnost i preciznost, na osnovu [Taylor, 1999]

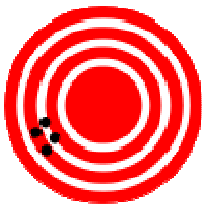
Naši rezultati daju tačnost i preciznost koja je dobijena kao srednja vrednost deset ponavljanja i svaki put uz 10-struku unakrsnu validaciju. Takođe, prikazujemo i standardnu devijaciju. Da bi smo utvrdili da li je razlika između dva algoritma značajna ili ne, mi prikazujemo vrednosti *t*-testa, koje ukazuju na verovatnoću da je jedan algoritam bolji od drugih. U nastavku teksta, ukratko objasnićemo zašto smo koristili u eksperimentalnim rezultatima i tačnost i standardnu devijaciju.



Slika 6.2: Visoka tačnost, ali niska preciznost, na osnovu BIMP i ISO 5725

Na slici 6.1. prikazani su pojmovi tačnosti i preciznosti. Tačnost pokazuje bliskost rezultata merenja sa stvarnom vrednošću, a preciznost ukazuje na ponovljivost, odnosno reproduktivnost merenja. Preciznost merenog sistema, koja se naziva i ponovljivost, pokazuje stepen u kojem ponovno merenje pod nepromenjenim uslovima daje iste rezultate. Sistem za merenje može biti tačan, ali nije precizan, precizan, ali nije tačan, niti tačan niti precizan, ili oboje i tačan i precizan. Merni sistem je dobro dizajniran ako je i tačan i precizan. Uporedićemo pojmove tačnosti i preciznosti na primeru mete.

Na slici 6.2. je prikazana visoka tačnost, ali niska preciznost, dok je na slici 6.3. prikazana visoka preciznost, ali niska tačnost. Analogija koja se ovde koristi je u cilju da se objasni razlika između tačnosti i preciznosti. U toj analogiji, ponovljena merenja su upoređena sa strelicama koje pogađaju metu. Tačnost opisuje bliskost strelice ka ciljnom centru. Strelice koje pogađaju bliže centru smatra se da pogađaju tačnije.



Slika 6.3: Visoka preciznost, ali niska tačnost, na osnovu BIMP i ISO 5725

Za nastavak analogije, ako je veliki broj strelica ispucan, preciznost bi bila veličina klastera strelice. Kada su sve strelice grupisane zajedno, klaster se smatra preciznim jer su sve strelice pogodile u blizini istog mesta, čak i ako nužno ne pogađaju u blizini samog centra. Merenja su tada precizna, iako nisu nužno tačna. Idealni merni uređaj je i tačan i precizan, sa merenjima koja su sva blizu oko poznate vrednosti.

To vredi i kada se merenja ponavljaju i dobiju prosečne vrednosti. U tom slučaju, termin standardna greška se ispravno primenjuje: preciznost proseka je jednaka poznatoj standardnoj devijaciji procesa podeljena korenom broja merenja proseka. To znači, da naša merena standardna devijacija podeljena sa korenom iz 10, što predstavlja broj merenja proseka, odgovara preciznosti. Takođe, centralna granična teorema pokazuje da je raspodela verovatnoća prosečnih merenja bliže normalnoj raspodeli nego kod pojedinačnih merenja.

U našem eksperimentalnom istraživanju, kad god smo uporedili dva ili više algoritama, u radu dajemo tabelu tačnosti klasifikacije, i prikazujemo dve vrste grafova sa stubićima. Jedan graf sa stubićima prikazuje apsolutnu razliku u tačnosti klasifikacije i drugi graf sa stubićima prikazuje apsolutnu razliku u standardnoj devijaciji za tačnost klasifikacije. Upoređivanje će uglavnom biti takvo da drugi algoritam je algoritam kod koga je urađena predselekcija atributa, a prvi algoritam je standardni algoritam bez predselekcije atributa. Kad je vrednost stubića veća od nule, drugi algoritam sa predselekcijom atributa nadmašuje svojom vrednošću prvi algoritam koji je standardni algoritam.

Kada smo prikazivali rezultate za potrebno vreme za trening podataka, oni su izražavani u jedinicama CPU sekundi. Eksperiment je rađen na AMD Phenom (tm) 9650 Quard-Core Pocessor 2.31 GHz sa 4GB RAM-a. Takođe, kod upoređivanja algoritama, dajemo tabelu potrebnog vremena za trening i prikazujemo dve vrste grafova sa stubićima. Jedan graf sa stubićima pokazuje apsolutnu razliku u potrebnom vremenu za trening i drugi graf sa stubićima pokazuje apsolutnu razliku u standardnoj devijaciji za potrebno vreme treninga.

6.2. Statistički testovi (testovi značajnosti)

U eksperimentalnom istraživanju kod izvođenja statističkih testova postoje određeni koraci kojih se treba pridržavati da bi zaključak bio pouzdan, a to su: postavljanje nulte hipoteze, biranje nivoa pouzdanosti, određivanje veličine uzorka, biranje statističkog testa za testiranje hipoteze, utvrđivanje kritične vrednosti za odabrani statistički test, prikupljanje podataka, izračunavanje statističke veličine za

odabrani statistički test, donošenje statističkog zaključka i izražavanje statističkog zaključka.

U nastavku teksta biće reči o merama centralne tendencije, merama varijabilnosti i testovima hipoteza koje smo u radu koristili.

Centralna tendencija je težnja ka okupljanju podataka skupa oko jedne centralne vrednosti, koja je opšta i reprezentativna za celu distribuciju. Njihova uloga je da, zanemarujući individualne razlike između podataka skupa, istaknu onu veličinu koja je za sve njih karakteristična i koja može da služi kao sredstvo za upoređivanje raznih serija. U mere centralne tendencije spada: aritmetička sredina, medijana, moda, geometrijska sredina i harmonijska sredina. Mi ćemo u našim istraživanjima koristiti aritmetičku sredinu.

Aritmetička sredina je srednja vrednost (procena parametra μ), čija se vrednost dobija deljenjem sume eksperimentalno dobijenih vrednosti sa brojem merenja, što je dato u sledećem izrazu:

$$\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{6.1}$$

Kao mere varijabilnosti, koje daju informaciju o različitim odstupanjima u statističkom skupu, može da se koristi interval varijacije (raspon), standardna devijacija, varijansa i koeficijent varijanse.

Interval varijacije je razmak od najmanje do najveće vrednosti obeležja posmatranja. Predstavlja najnetačniju meru grupisanja rezultata oko neke srednje vrednosti.

$$R = x_n - x_1,\ x_1 < x_2 < \cdots x_n \tag{6.2}$$

gde je *x* merena veličina.

Standardna devijacija je mera odstupanja vrednosti obeležja od aritmetičke sredine, i data je sledećim izrazom:

$$s = \sqrt{\frac{\sum(\overline{x} - x_i)^2}{n-1}} \tag{6.3}$$

Varijansa je prosečno kvadratno odstupanje od aritmetičke sredine, dato izrazom:

$$s^2 = \frac{\sum(\overline{x} - x_i)^2}{n-1} \tag{6.4}$$

U našim istraživanjima smo koristili standardnu devijaciju kao meru varijabilnosti, odnosno informaciju o različitim odstupanjima u statističkom skupu.

Za testiranje hipoteze koriste se parametrijski i neparametrijski testovi. Parametrijske metode koriste se za upoređivanje dve ili više grupa podataka i zasnivaju se na pretpostavci da su podaci normalno raspodeljeni. Ove metode se uvek zasnivaju na teoriji verovatnoće i uvek se u njima pojavljuje potreba za ocenjivanjem pojedinih parametara (srednje vrednosti, standardne devijacije ili varijanse). Međutim, kada ne može sa sigurnošću da se utvrdi da li je raspodela jedne grupe podataka normalna, izračunavanje pojedinih parametara i primena

parametrijskih metoda daju vrlo nepouzdane zaključke. U tim slučajevima se primenjuju neparametrijske metode, koje se zasnivaju na pretpostavci da postoji bilo koja verovatnoća raspodele.

Za eliminisanje „spoljnih“ rezultata, vrednosti koje se izdvajaju u odnosu na ostale, može se koristiti *Dixon*-ov test (Q-test) – za male uzorke, ili *Grubbs*-ov test (G-test). *F*-test služi da utvrdi da li je razlika između varijansi dva uzorka značajna.

Mi ćemo u našim istraživanjima koristiti *t*-test koji se koristi za utvrđivanje postojanja sistematskih grešaka. Koristi se u sledećim slučajevima: (1) kada se upoređuje srednja vrednost grupe podataka sa pravom vrednošću (određivanje tačnosti), (2) kada se upoređuju srednje vrednosti dve grupe podataka, (3) kod paralelnih određivanja.

Kod upoređivanja eksperimentalno određene srednje vrednosti sa pravom vrednošću, parametar *t* se izračunava prema sledećoj jednačini:

$$t = \frac{(\bar{x} - \mu) \times \sqrt{N}}{s} \tag{6.5}$$

$\bar{x}$ – aritmetička sredina merenih vrednosti, μ - prava vrednost, *N* – broj merenja, *s* - standardna devijacija.

Dobijena vrednost se upoređuje sa kritičnom *t*-vrednošću, koja se za dati nivo pouzdanosti i broj stepeni slobode, očitava u tabeli. Ako vrednost *t* prelazi određenu kritičnu vrednost nulta hipoteza se odbacuje. U suprotnom ne postoje dokazi za postojanje sistematske greške (ovo ne znači da sistematska greška ne postoji već samo da ona nije izražena).

U našem eksperimentalnom istraživanju koristili smo uporedni *t*-test (eng. *Paired T-Test*), gde je nivo značajnosti postavljen na vrednost 0.05. Ako imamo simultano određivanje tačnosti klasifikacije u različitim setovima podataka pomoću dve metode, za utvrđivanje da li se dobijena vrednost različitim metodama značajno razlikuje koristimo uporedni *t*-test. Uporednim *t*-testom se testira značajnost srednje vrednosti razlike parova *d* prema sledećoj jednačini:

$$t = \frac{\bar{d}\sqrt{N}}{s_d} \tag{6.6}$$

gde je s_d – standardna devijacija dobijenih razlika. Ukoliko je izračunata vrednost parametra *t* veća od tablične (kritične vrednosti), nulta hipoteza se odbacuje i kaže se da se *d* značajno razlikuje od nule, odnosno da je razlika u parovima statistički značajna.

U tabelama koje slede za tačnost klasifikacije različitih klasifikatora i u tabelama za vreme potrebno za trening podataka su prikazane oznake „+“ i „-“, koje označavaju da je određeni rezultat statistički bolji (+) ili lošiji (-) od osnovnog klasifikatora na nivou značajnosti koji je specificiran na vrednost od 0,05.

U tabelama za tačnost klasifikacije različitih klasifikatora oznaka „+“ označava značajno veću vrednost za tačnost klasifikacije, dok „-“ označava značajno manju vrednost za tačnost klasifikacije.

U tabelama koje sadrže podatke o vremenu potrebnom za trening podataka oznaka „+“ označava značajno manju vrednost za potrebno vreme, što znači da se radi o statistički boljem rezultatu dok „-“ označava značajno veću vrednost za potrebno vreme što znači da se radi o statistički lošijem rezultatu. S obzirom da vreme potrebno za trening podataka može da se menja, ako primenimo različite metode za redukciju dimenzionalnosti podataka, dobro je da tokom eksperimenta možemo da dobijemo manje vrednosti za potrebno vreme treniranja, jer onda naš algoritam radi brže, što je posebno značajno ako imamo problem u realnom vremenu. Znači da su manje vrednosti u tabelama za vreme bolje, zbog čega se i smisao statističkih pokazatelja „+“ i „-“ menja u odnosu na tačnost klasifikacije.

7. ESTIMACIJA TAČNOSTI KLASIFIKACIJE ZA METODE FILTRIRANJA

U sedmom delu, nakon razmatranja postavki eksperimentalnog istaživanja, biće prikazani rezultati istraživanja za različite metode filtriranja i to za svaki klasifikacioni algoritam posebno.

7.1. Postavke eksperimentalnog istraživanja

Metode filtriranja funkcionišu nezavisno o izabranom algoritmu veštačkog učenja, za razliku od metode selekcije prethodnim učenjem. Vrednost atributa se heuristički procenjuje analizom opštih karakteristika podataka iz skupa za učenje. Ove metode koriste više različitih tehnika izbora atributa, jer postoji više načina heurističkog vrednovanja atributa. Metode filtriranja se dele u dve osnovne grupe, zavisno o tome vrednuje li korišćena heuristika podskupove atributa ili pojedinačne atribute.

U ovom radu, koristimo sledeće metode filtriranja za rangiranje atributa koje su statistički i entropijski zasnovane, a pokazuju dobre performanse u različitim domenima: IG, GR, SU, RF, OR i CS.

Za redukciju dimenzionalnosti podataka kod klasifikacionih problema veštačke inteligencije, u ovom radu smo kod metoda filtriranja za potrebe rangiranja atributa za sve skupove podataka, koristili potpuni skup podataka za treniranje, umesto 10-struke unakrsne validacije. Međutim, nakon redukcije veličine posmatranog skupa podataka, za potrebe klasifikacije, koristili smo 10-struku unakrsnu validaciju.

Sve metode filtriranja, IG, GR, SU, RF, OR i CS su odradile rangiranje atributa za svaki pojedinačni skup podataka. S obzirom da metode rangiranja prikazuju sve atribute po onom redosledu kakav je njihov značaj za klasifikacioni problem, ove metode ne vrše automatski redukciju broja atributa.

Da bi se uz pomoć ovih metoda izvršila redukcija broja atributa, postoje dve mogućnosti: (1) korišćenje praga, ili (2) korišćenje odgovarajućeg broja atributa za svaki set podataka i svaku od metoda filtriranja. U slučaju prve mogućnosti, korišćenja praga, za svaki set podataka i za svaku od metoda filtriranja, pretražuje se skup svih mogućih rešenja tako što se uzima početna vrednost praga i ona se u svakoj daljoj iteraciji inkrementira za određenu vrednost inkrementa, i propušta se kroz svaki od klasifikatora, dok se nedostigne unapred zadata vrednost praga. Početna vrednost praga npr. može biti 0.00, a potom se sa što manjom vrednošću za inkrement, dostiže unapred zadata gornja granica praga, npr. 0.05. Tokom ovog inkrementiranja za vrednost praga, meri se tačnost klasifikacije za svaki klasifikator, a najbolje rešenje za izbor broja atibuta je ono rešenje koje daje najveću tačnost klasifikacije za izabrani prag.

Tabela 7.1. Broj atributa u originalnom skupu podataka i broj atributa selektovan uz pomoć metoda filtriranja. Pretraživanjem skupa svih mogućih rešenja za svaku metodu je pronađen optimalni broj atributa.

Skup	Orig. skup	IG	GR	SU	RF	OR	CS
bc	9	8	3	3	2	8	3
ca	15	1	2	1	1	2	1
cg	20	5	17	16	19	19	17
ct	23	18	12	8	21	19	6
he	19	1	1	5	6	2	1
li	6	4	4	4	4	5	4
lc	56	5	17	2	4	9	4
ma	5	3	2	2	4	2	2
m1	6	4	5	5	3	5	4
m2	6	5	5	5	5	5	5
m3	6	2	2	2	2	2	2
mu	22	9	5	5	8	7	7
pa	23	21	21	21	13	22	21
pi	8	1	4	1	1	4	1
se	19	16	17	16	14	16	16
so	35	33	33	31	32	34	23
sh	13	3	9	3	6	3	3
vo	16	5	5	5	5	2	2

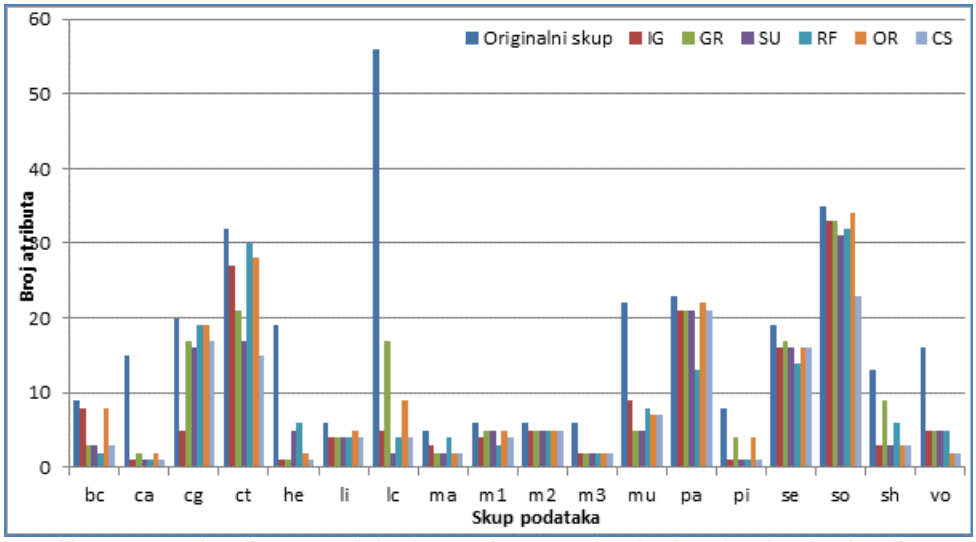
Druga mogućnost za izbor broja atributa određenog skupa podataka je pretraživanje skupa svih mogućih rešenja za broj atributa koji će se koristiti u klasifikatoru. Postupak je sličan predhodnom. U ovom slučaju, pretražuje se skup svih mogućih rešenja tako što se uzimaju sve moguće vrednosti za broj atributa odgovarajućeg skupa podataka za svaku od metoda filtriranja i propušta se svako pojedinačno rešenje kroz svaki od klasifikatora. Tokom ovog postupka, meri se tačnost klasifikacije za svaki klasifikator, a najbolje rešenje za izabrani broj atibuta je ono koje daje najveću tačnost klasifikacije.

U ovom eksperimentalnom istraživanju korišćena je druga mogućnost, odnosno selektovanje broja atributa koji će se koristiti u klasifikatoru, kako bi dobili što veću tačnost klasifikacije za dati skup podataka i posmatranu metodu filtriranja.

U tabeli 7.1. prikazan je optimalan broj atributa za potrebe klasifikacije, nakon pretraživanja skupa svih mogućih rešenja za svaku od metoda. U tabeli je prikazana i originalna veličina skupa, kako bi se uporedili efekti redukcije dimenzionalnosti podataka. U deset setova podataka, od 18 posmatranih, tačno pola ili više od pola metoda je smanjilo originalni broj atributa na pola. Ti setovi podataka su *bc, ca, he, lc, ma, m3, mu, pi, sh* i *vo*.

Na slici 7.1. prikazan je broj atributa u originalnom skupu podataka i optimalan broj atributa dobijen metodama filtriranja. Najveću dobrobit od redukcije dimenzionalnosti podataka ima skup podataka *lc,* gde od 56 atributa, metodom filtriranja smo izdvojili mali broj atributa, čak manje od jedne šestine, za svaku od

metoda, izuzev metode GR, koji su relevantni za posmatrani problem klasifikacije. Za skup podataka *ca* uočavamo da su sve metode filtriranja, pokazale da su najviše dva atributa značajna za posmatrani problem klasifikacije, a da ostali atributi ne utiču na postizanje veće pouzdanosti klasifikacije. Za skup podataka *he*, koji originalno ima 19 atributa, sve metode filtriranja pokazuju da je najviše 6 atributa značajno za izučavanu pojavu. Kod veštačkog skupa podataka *m3*, sve metode filtriranja pokazuju da su samo dva atributa značajna za posmatrani problem klasifikacije. Metode filtriranja za skup podataka *pi*, pokazuju da najviše 4 atributa su značajna za problem klasifikacije, a u slučaju seta podataka *vo* 5 atributa.



Slika 7.1: Broj atributa u originalnom skupu podataka i optimalan broj atributa dobijen metodama filtriranja

Ako imamo simultano određivanje tačnosti klasifikacije u različitim setovima podataka pomoću dve metode, za utvrđivanje da li se dobijena vrednost različitim metodama značajno razlikuje koristimo uporedni *t*-test. U eksperimentalnom istraživanju koristili smo uporedni *t*-test, gde je nivo značajnosti postavljen na vrednost 0.05. Tokom eksperimentalnog istraživanja testirali smo značajnost srednje vrednosti razlike parova *d* prema izrazu: $t = \frac{\bar{d}\sqrt{N}}{s_d}$ gde je s_d – standardna devijacija dobijenih razlika. Ako je izračunata vrednost parametra *t* veća od kritične vrednosti, nulta hipoteza se odbacuje i kaže se da se *d* značajno razlikuje od nule, odnosno da je razlika u parovima statistički značajna.

U tabelama koje slede za tačnost klasifikacije različitih klasifikatora i u tabelama za vreme potrebno za trening podataka su prikazane oznake „+“ i „-“, koje označavaju da je određeni rezultat statistički bolji (+) ili lošiji (-) od osnovnog klasifikatora na nivou značajnosti koji je specificiran na vrednost od 0,05.

U nastavku eksperimentalnog istraživanja, za izabrani optimalan broj atributa, za svaki skup podataka i metodu filtriranja, proveravana je tačnost klasifikacije korišćenjem različitih algoritama, i to IBk, *Naïve Bayes*, SVM, J48 i RBF mreže. U nastavku teksta prikazani su dobijeni rezultati. Treba uočiti da su prikazane različite skale na slikama za apsolutnu tačnost klasifikacije, standardnu devijaciju za tačnost klasifikacije, vreme treninga i standardnu devijaciju za vreme treninga, kako bi se bolje uočile razlike koje postoje među rezultatima.

7.2. IBk

Za svaki skup podataka i metodu filtriranja i za izabrani optimalan broj atributa, proveravana je tačnost klasifikacije korišćenjem algoritma IBk. Tabela 7.2. prikazuje tačnost klasifikacije za IBk za originalni skup podataka i redukovani skup podataka, dobijen nakon primene metoda filtriranja.

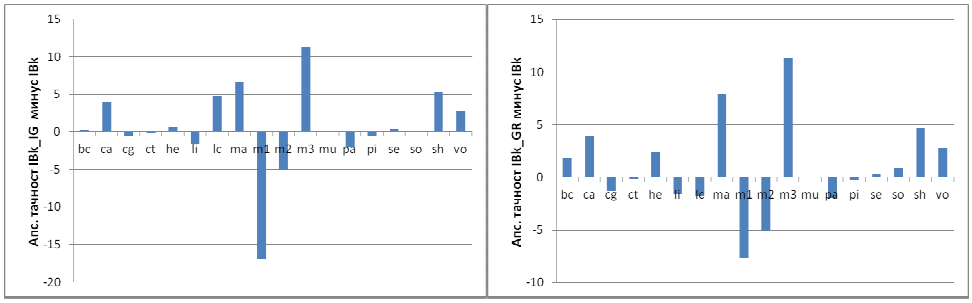
Tabela 7.2. Tačnost klasifikacije IBk algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	72.85	73.06	74.67	74.64	71.72	72.85	73.51
ca	81.57	85.51 +	85.46 +	85.51 +	85.51 +	85.45 +	85.51 +
cg	71.88	71.33	70.59	70.75	71.13	71.72	70.29
ct	98.85	98.79	98.74	98.81	98.79	98.77	98.76
he	81.40	81.97	83.78	81.02	83.33	83.65	81.91
li	62.22	60.62	60.62	60.62	64.02	60.29	60.62
lc	68.75	73.50	67.00	75.25	70.67	63.67	68.92
ma	75.60	82.27 +	83.49 +	83.38 +	75.18	82.75 +	83.36 +
m1	99.87	82.87 -	92.21	92.63	100.00	80.30 -	82.87 -
m2	72.22	67.13 -	67.13 -	67.13 -	68.98 -	75.00 +	67.13 -
m3	85.88	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +
mu	100.00	100.00	100.00	100.00	100.00	100.00	100.00
pa	95.91	93.92	93.97	93.97	97.08	95.29	94.27
pi	70.62	69.99	70.39	69.99	69.99	71.21	69.99
se	97.15	97.57 +	97.49 +	97.57 +	97.48	97.57 +	97.57 +
so	91.20	91.26	92.06	91.79	91.89	91.11	91.68
sh	76.15	81.52	80.81	81.22	78.26	81.56	81.78
vo	92.58	95.38 +	95.36 +	95.36 +	96.04 +	95.63 +	95.63 +

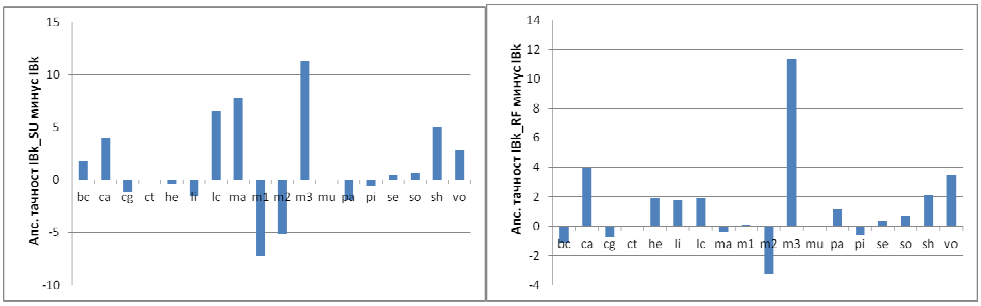
Na slikama 7.2, 7.3. i 7.4. prikazana je apsolutna razlika u tačnosti klasifikacije IBk algoritma na osnovnom skupu podataka i IBk algoritma sa različitim metodama filtriranja. Primenjeni metod filtriranja IG je u skoro dve trećine skupova podataka (11 skupova) pokazao iste ili bolje rezultate od IBk algoritma na osnovnom skupu podataka. U 5 skupova podataka rezultati su bili i statistički bolji.

Metod filtriranja GR je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od IBk algoritma na osnovnom skupu podataka. Takođe, kao i kod metode IG, kod 5 skupova podataka rezultati su bili i statistički bolji.

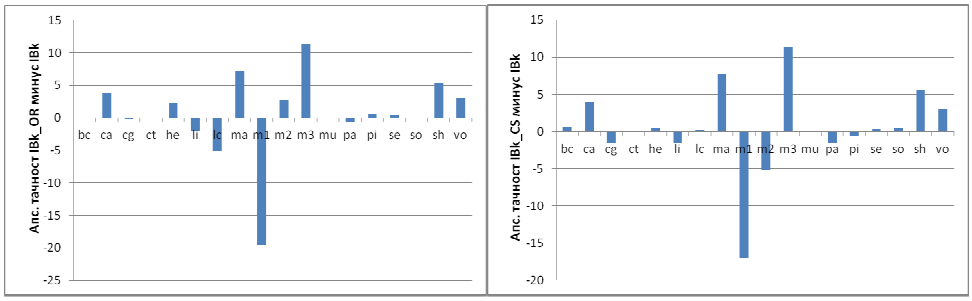
Možemo uočiti da u šest setova podataka (*ca*, *ma, m2, m3, se* i *vo*) imamo dobijene rezultate za bar jednu od metoda filtriranja koji su statistički bolji od osnovnog klasifikatora. Ni u jednom setu podataka, nemamo značajno lošije podatke za sve metode filtriranja, što znači da uvek možemo izabrati metodu za dati skup podataka koja ima statistički bolje rezultate ili rezultate koji su približni originalnom skupu podataka. Kod tri skupa podataka: *ca, m3* i *vo* sve primenjene metode filtriranja daju statistički bolje rezultate od osnovnog klasifikatora.



Slika 7.2: Apsolutna tačnost klasifikacije IBk_IG minus IBk i IBk_GR minus IBk



Slika 7.3: Apsolutna tačnost klasifikacije IBk_SU minus IBk i IBk_RF minus IBk



Slika 7.4: Apsolutna tačnost klasifikacije IBk_OR minus IBk i IBk_CS minus IBk

Primenjeni metod filtriranja SU je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od IBk algoritma na osnovnom skupu podataka. U 5 skupova podataka rezultati su bili i statistički bolji. Metod filtriranja

RF je u dve trećine skupova podataka (12 skupova) pokazao iste ili bolje rezultate od IBk algoritma na osnovnom skupu podataka. U 3 skupa podataka, rezultati su bili i statistički bolji.

Metod filtriranja OR je u skoro dve trećine skupova podataka (11 skupova) pokazao iste ili bolje rezultate od IBk algoritma na osnovnom skupu podataka, a u 6 skupova podataka, rezultati su bili i statistički bolji. Primenjeni metod filtriranja CS je u skoro dve trećine skupova podataka (11 skupova) pokazao iste ili bolje rezultate od IBk algoritma na osnovnom skupu podataka, a u 5 skupova podataka rezultati su bili i statistički bolji.

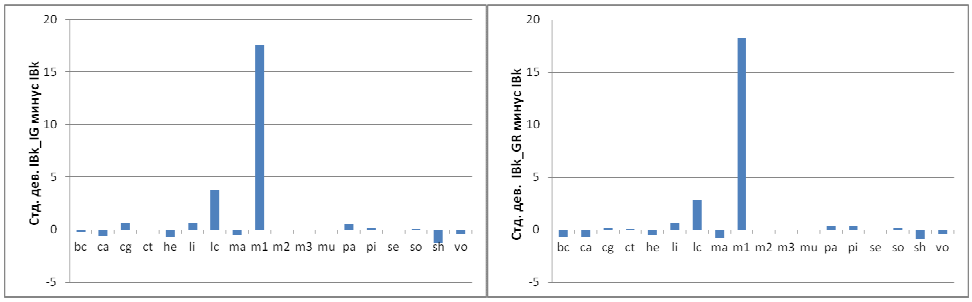
Korišćenjem IBk klasifikatora, možemo da zaključimo da je RF metoda filtriranja u najvećem broju slučaja dovela do statistički boljih rezultata na posmatranim skupovima podataka.

Tabela 7.3. Standardna devijacija za tačnost klasifikacije IBk algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

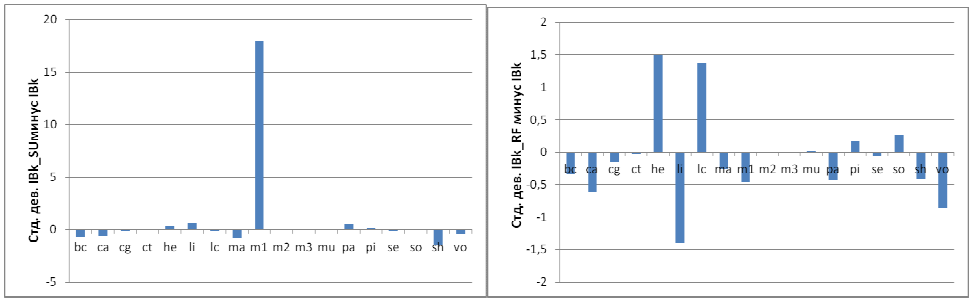
Skup	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	6.93	6.73	6.26	6.30	6.60	6.93	6.30
ca	4.57	3.96	3.93	3.96	3.96	3.94	3.96
cg	3.68	4.29	3.87	3.59	3.52	3.79	4.05
ct	0.77	0.74	0.78	0.73	0.74	0.82	0.72
he	8.55	7.86	8.10	8.93	10.05	8.88	7.99
li	8.18	8.80	8.80	8.80	6.78	8.55	8.80
lc	22.33	26.12	25.21	22.14	23.70	22.68	25.59
ma	3.90	3.37	3.13	3.10	3.64	3.08	3.09
m1	0.46	18.05	18.74	18.43	0.00	25.50	18.05
m2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mu	0.00	0.00	0.00	0.00	0.01	0.00	0.00
pa	4.52	5.07	4.84	5.00	4.10	4.78	4.69
pi	4.67	4.84	5.07	4.84	4.84	4.75	4.84
se	1.11	1.03	1.05	1.03	1.05	1.03	1.03
so	3.00	3.03	3.19	3.09	3.26	3.01	3.16
sh	8.46	7.21	7.65	7.01	8.05	6.74	6.86
vo	3.63	3.21	3.20	3.20	2.76	2.76	2.76

Standardna devijacija je u statistici apsolutna mera disperzije u osnovnom skupu, koja nam govori koliko u proseku elementi skupa odstupaju od aritmetičke sredine skupa. Najmanja moguća vrednost standardne devijacije je 0 i to se dešava kada su svi rezultati u distribuciji jednaki. Ova mera je osetljiva na ekstremne vrednosti, jer se bazira na distanci pojedinačnih rezultata od aritmetičke sredine. Vrednost standardne devijacije je $s \in [0, +\infty)$. U ovom eksperimentalnom istraživanju, kao meru varijabilnosti, koja daje informaciju o različitim odstupanjima u statističkom skupu, koristili smo standardnu devijaciju. Standardna devijacija je

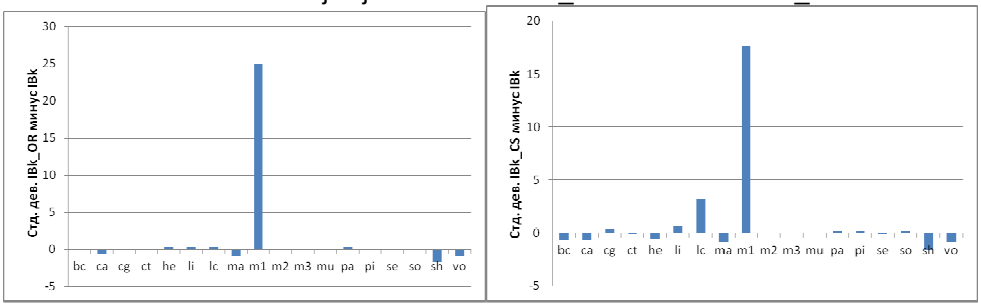
mera odstupanja vrednosti obeležja od aritmetičke sredine, i data je sledećim izrazom: $s = \sqrt{\frac{\sum(\bar{x} - x_i)^2}{n-1}}$.



Slika 7.5: Standardna devijacija za tačnost IBk_IG minus IBk i IBk_GR minus IBk



Slika 7.6: Standardna devijacija za tačnost IBk_SU minus IBk i IBk_RF minus IBk



Slika 7.7: Standardna devijacija za tačnost IBk_OR minus IBk i IBk_CS minus IBk

U poglavlju 6 smo izneli tvrdnju da je dobar onaj algoritam koji daje sličan rezultat u svim slučajevima, odnosno vrednost standardne devijacije je minimalna. Tabela 7.3. prikazuje standardnu devijaciju za tačnost klasifikacije IBk algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa, osim u slučaju *m1* skupa podataka. U slučaju *m1* skupa podataka za sve metode filtriranja dobijamo veliku vrednost za standardnu devijaciju, osim za metodu RF.

Na slikama 7.5, 7.6. i 7.7. prikazana je apsolutna razlika u vrednostima standardne devijacije za tačnost klasifikacije IBk algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, u pozitivnom i negativnom smeru, to je i veće odstupanje između standardnih devijacija. Najmanje odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka pokazuje metoda RF, koja je kod nekih skupova podataka uspela da smanji, a kod nekih da poveća standardnu devijaciju.

Tabela 7.4. Potrebno vreme za trening (u sekundama) IBk algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda

Skup	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	0.00	0.00	0.00	0.00	0.02 -	0.01 -	0.00
ca	0.00	0.00	0.00	0.00	0.19 -	0.03 -	0.00
cg	0.00	0.00	0.00	0.00	0.48 -	0.06 -	0.00
ct	0.00	0.03 -	0.03 -	0.03 -	4.17 -	0.24 -	0.03 -
he	0.00	0.00	0.00	0.00	0.01-	0.01 -	0.00
li	0.00	0.00	0.00	0.00	0.03 -	0.01-	0.00
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.00	0.00	0.00	0.00	0.17 -	0.02 -	0.00
m1	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m2	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m3	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
mu	0.00	0.02 -	0.03 -	0.02 -	30.20 -	0.76 -	0.02 -
pa	0.00	0.00	0.00	0.00	0.03 -	0.02 -	0.00
pi	0.00	0.00	0.00	0.00	0.16 -	0.02 -	0.00
se	0.00	0.06 -	0.06 -	0.06 -	3.13 -	0.18 -	0.05 -
so	0.00	0.00	0.00	0.00	0.42 -	0.07 -	0.00
sh	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
vo	0.00	0.00	0.00	0.00	0.06 -	0.02 -	0.00

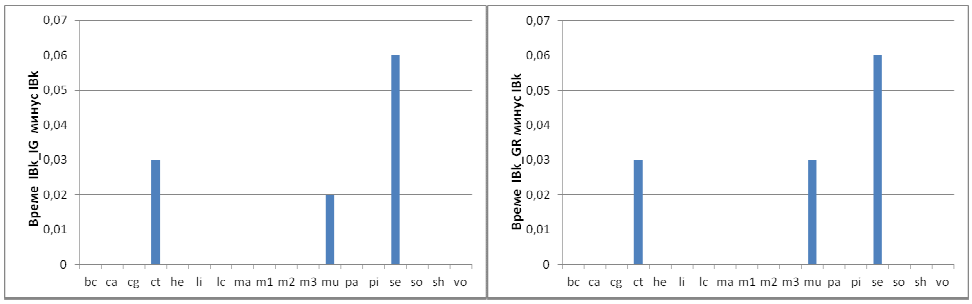
U tabeli 7.4. prikazano je potrebno vreme za trening u sekundama IBk algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda. Potrebno vreme za trening podataka IBk klasifikatora za sve originalne skupove podataka iznosi 0.00, dok za filter metode ono je nešto veće. Kod samo tri skupa podataka (*ct, mu* i *se*), ni jedna od metoda ne daje minimalno potrebno vreme za trening, dok kod svih ostalih skupova podataka u jednako ili više od pola slučajeva metode filtriranja daju minimalno potrebno vreme za trening.

Na slikama 7.8, 7.9. i 7.10. prikazana je apsolutna razlika u potrebnom vremenu za trening IBk algoritma na osnovnom skupu podataka i IBk algoritma sa različitim metodama filtriranja. Primenjeni metodi filtriranja IG, GR, SU i CS su samo u tri skupa podataka pokazali nešto lošije rezultate za potrebno vreme za trening i kod tih skupova podataka, rezultati su bili i statistički lošiji.

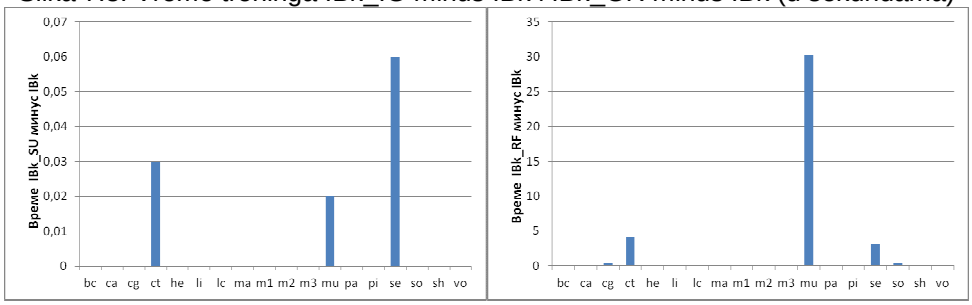
Metod filtriranja RF je u svim skupovima podataka, osim u jednom, pokazao lošije rezultate za potrebno vreme za trening od IBk algoritma na osnovnom skupu podataka, a ti rezultati su bili i statistički lošiji.

Metod filtriranja OR je u svim skupovima podataka pokazao lošije rezultate od IBk algoritma na osnovnom skupu podataka, a ovi rezultati su bili i statistički lošiji.

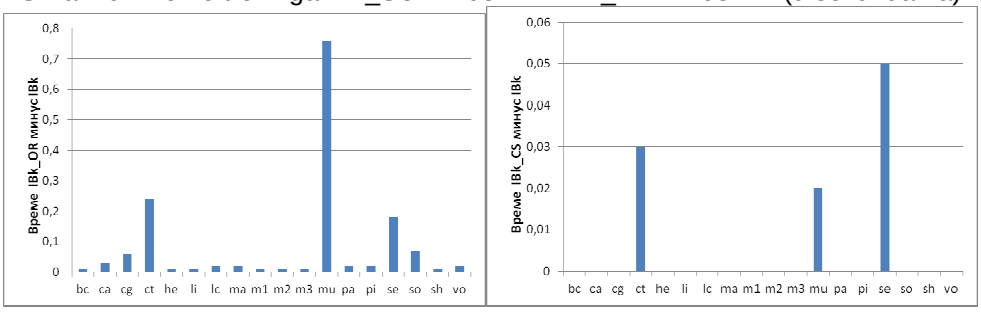
Korišćenjem IBk klasifikatora, možemo da zaključimo da su IG, GR, SU i CS metode filtriranja u najmanjem broju slučaja dovele do statistički lošijih rezultata za potrebno vreme za trening na posmatranim skupovima podataka.



Slika 7.8. Vreme treninga IBk_IG minus IBk i IBk_GR minus IBk (u sekundama)



Slika 7.9: Vreme treninga IBk_SU minus IBk i IBk_RF minus IBk (u sekundama)



Slika 7.10: Vreme treninga IBk_OR minus IBk i IBk_CS minus IBk (u sekundama)

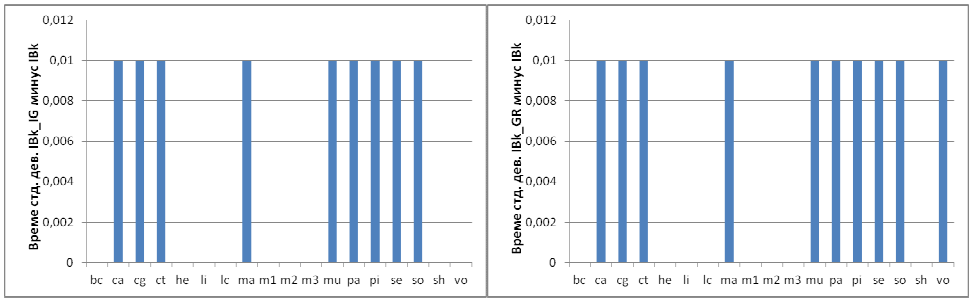
Tabela 7.5. prikazuje standardnu devijaciju za vreme treninga IBk algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno

između standardnog algoritma i algoritama koji koriste predselekciju atributa. Nešto veće vrednosti za standardnu devijaciju za vreme treninga ima RF metoda filtriranja.

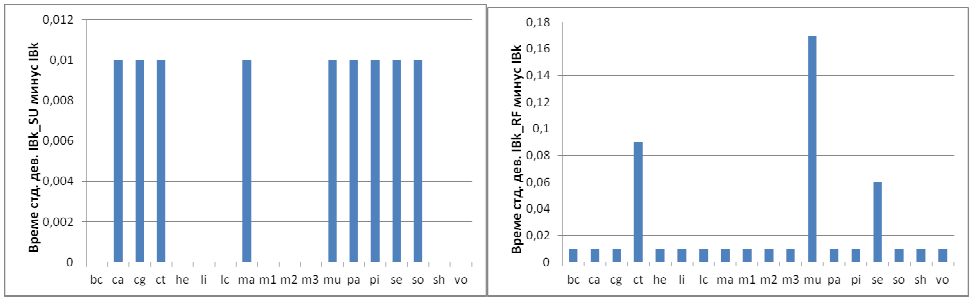
Tabela 7.5. Standardna devijacija za vreme treninga (u sekundama) IBk algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	IBk	IBk_IG	IBk_GR	IBk_SU	IBk_RF	IBk_OR	IBk_CS
bc	0.00	0.00	0.00	0.00	0.01	0.01	0.00
ca	0.00	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.00	0.01	0.01	0.01	0.01	0.01	0.01
ct	0.00	0.01	0.01	0.01	0.09	0.02	0.01
he	0.00	0.00	0.00	0.00	0.01	0.01	0.00
li	0.00	0.00	0.00	0.00	0.01	0.01	0.00
lc	0.00	0.00	0.00	0.00	0.01	0.01	0.00
ma	0.00	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m2	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m3	0.00	0.00	0.00	0.00	0.01	0.01	0.00
mu	0.00	0.01	0.01	0.01	0.17	0.01	0.01
pa	0.00	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.00	0.01	0.01	0.01	0.01	0.01	0.01
se	0.00	0.01	0.01	0.01	0.06	0.01	0.01
so	0.00	0.01	0.01	0.01	0.01	0.01	0.00
sh	0.00	0.00	0.00	0.00	0.01	0.01	0.00
vo	0.00	0.00	0.01	0.00	0.01	0.01	0.00

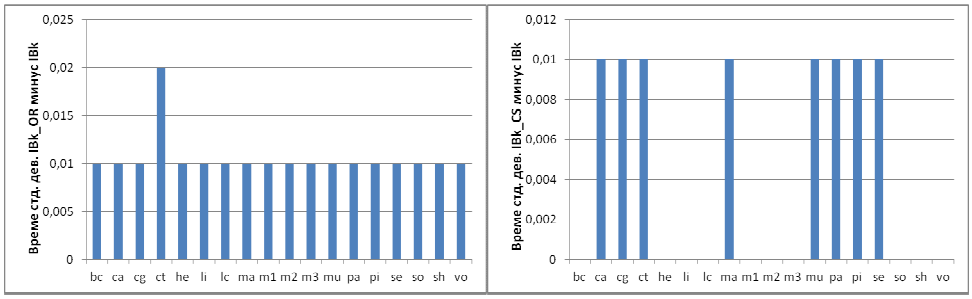
Na slikama 7.11, 7.12. i 7.13. prikazana je apsolutna razlika u vrednostima standardne devijacije za vreme treninga IBk algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, u pozitivnom i negativnom smeru, to je i veće odstupanje između standardnih devijacija. Najveće odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka pokazuje metoda RF, koja je kod svih skupova podataka uspela da poveća standardnu devijaciju.



Slika 7.11: Standardna devijacija za vreme IBk_IG minus IBk i IBk_GR minus IBk



Slika 7.12: Standardna devijacija za vreme IBk_SU minus IBk i IBk_RF minus IBk



Slika 7.13: Standardna devijacija za vreme IBk_OR minus IBk i IBk_CS minus IBk

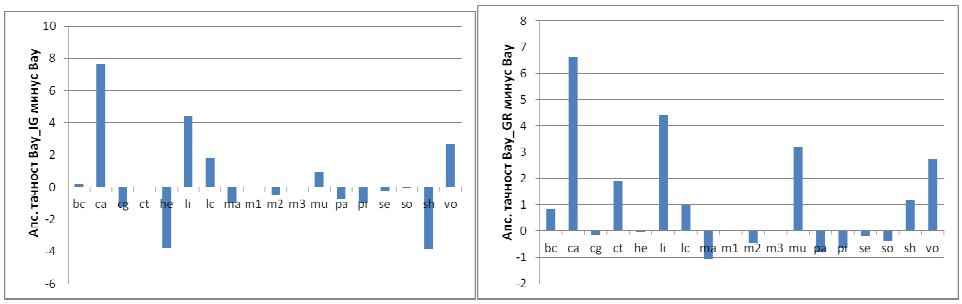
7.3. Naïve Bayes

Na osnovu prikazanih podataka u tabeli 7.6. možemo uočiti da u sedam setova podataka (*ca*, *ct*, *m2, mu, pa, se* i *vo*) imamo dobijene rezultate za bar jednu od metoda filtriranja koji su statistički bolji od osnovnog klasifikatora. I pored smanjenja dimenzionalnosti podataka, ni u jednom setu podataka, nemamo značajno lošije podatke za sve metode filtriranja, što znači da uvek možemo izabrati metodu za dati skup podataka koja ima statistički bolje rezultate ili rezultate koji su približni originalnom skupu podataka. Sve metode filtriranja imaju statistički bolje rezultate od osnovnog klasifikatora u slučaju tri skupa podataka: *ca, mu* i *vo*.

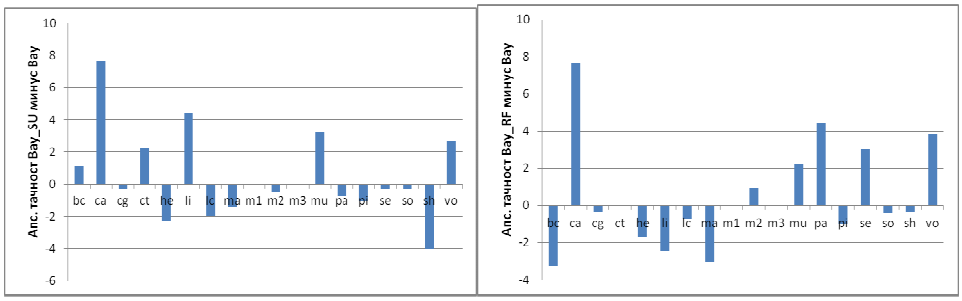
Apsolutna razlika u tačnosti klasifikacije *Naïve Bayes* algoritma na osnovnom skupu podataka i *Naïve Bayes* algoritma sa različitim metodama filtriranja prikazana je na slikama 7.14, 7.15. i 7.16. Primenjeni metod filtriranja IG je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, a u 3 skupa podataka, rezultati su bili i statistički bolji. Kod merenja tačnosti klasifikacije, metod filtriranja GR je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka. Kod metode GR, u 4 skupa podataka rezultati su bili i statistički bolji.

Tabela 7.6. Tačnost klasifikacije *Naïve Bayes* algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

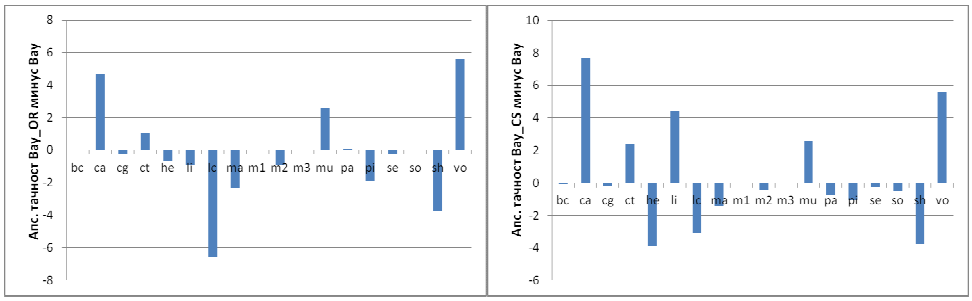
Skup	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay_CS
bc	72.70	72.87	73.54	73.78	69.45	72.70	72.59
ca	77.86	85.51 +	84.49 +	85.51 +	85.51 +	82.54 +	85.51 +
cg	75.16	73.95	75.01	74.87	74.79	74.88	74.94
ct	87.30	87.31	89.21 +	89.57 +	87.28	88.32 +	89.68 +
he	83.81	80.01	83.77	81.55	82.12	83.17	79.95
li	54.89	59.29	59.29	59.29	52.41	53.99	59.29
lc	78.42	80.25	79.42	76.42	77.67	71.83	75.33
ma	82.64	81.62	81.58	81.26	79.59 -	80.29 -	81.25
m1	74.64	74.64	74.64	74.64	74.64	74.64	74.64
m2	61.57	61.11 -	61.11 -	61.11 -	62.50 +	60.65 -	61.11 -
m3	97.22	97.22	97.22	97.22	97.22	97.22	97.22
mu	95.76	96.68 +	98.95 +	98.95 +	97.97 +	98.33 +	98.33 +
pa	69.98	69.21	69.21	69.26	74.44 +	69.99	69.26
pi	75.75	74.72	75.09	74.72	74.72	73.86	74.72
se	80.17	79.92	79.98	79.92	83.20 +	79.92	79.92
so	92.94	92.91	92.56	92.62	92.52	92.94	92.43
sh	83.59	79.74	84.78	79.59	83.22	79.81	79.85
vo	90.02	92.71 +	92.76 +	92.71 +	93.88 +	95.63 +	95.63 +



Slika 7.14: Apsolutna tačnost klasifikacije Bay_IG minus Bay i Bay_GR minus Bay



Slika 7.15: Apsolutna tačnost klasifikacije Bay_SU minus Bay i Bay_RF minus Bay



Slika 7.16: Apsolutna tačnost klasifikacije Bay_OR minus Bay i Bay_CS minus Bay

Primenjeni metod filtriranja SU je u nešto manje od pola skupova podataka (8 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka. U 4 skupa podataka, rezultati su bili i statistički bolji. Metod filtriranja RF je u nešto manje od pola skupova podataka (8 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, dok u čak 6 skupova podataka, rezultati su bili i statistički bolji.

Tabela 7.7. Standardna devijacija za tačnost klasifikacije *Naïve Bayes* algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

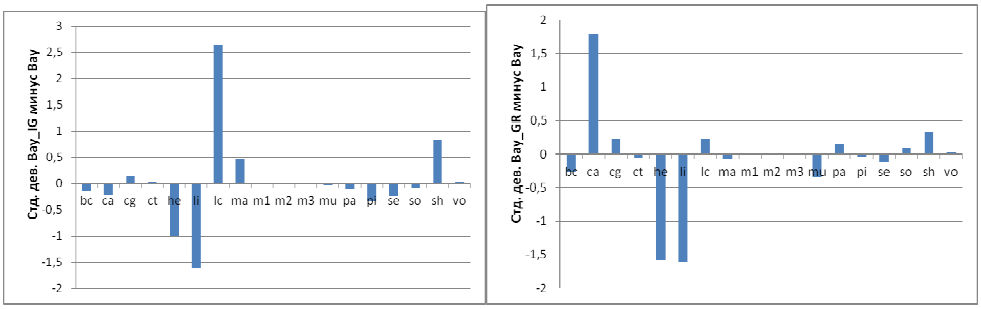
Skup	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay _CS
bc	7.74	7.61	7.47	7.22	6.54	7.74	6.79
ca	4.18	3.96	5.97	3.96	3.96	5.53	3.96
cg	3.48	3.63	3.71	3.73	3.48	4.01	3.65
ct	2.21	2.23	2.16	2.09	2.21	2.21	1.99
he	9.70	8.70	8.12	9.48	10.38	8.84	8.80
li	8.83	7.22	7.22	7.22	8.44	9.74	7.22
lc	21.12	23.77	21.34	22.35	22.03	17.16	23.39
ma	3.11	3.59	3.05	3.33	3.68	3.54	3.33
m1	4.26	4.26	4.26	4.26	4.26	4.26	4.26
m2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mu	0.73	0.69	0.38	0.38	0.55	0.44	0.44
pa	9.51	9.41	9.65	9.59	8.88	9.44	9.71
pi	5.32	4.99	5.29	4.99	4.99	5.90	4.99
se	2.12	1.89	2.01	1.89	2.29	1.89	1.89
so	2.92	2.84	3.00	2.89	2.90	2.89	3.01
sh	5.98	6.81	6.31	6.69	7.11	6.56	6.75
vo	3.91	3.94	3.94	3.94	3.53	2.76	2.76

Prilikom utvrđivanja tačnosti klasifikacije, metod filtriranja OR je u pola skupova podataka (9 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, a u 4 skupa podataka, rezultati su bili i statistički bolji. Primenjeni metod filtriranja CS je u nešto manje od pola skupova podataka (7 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, a u 4 skupa podataka rezultati su bili i statistički bolji.

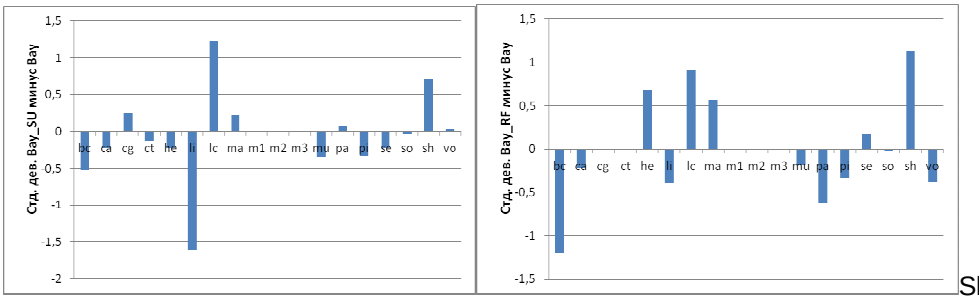
Korišćenjem *Naïve Bayes* klasifikatora, možemo da zaključimo da je RF metoda filtriranja u najvećem broju slučaja dovela do statistički boljih rezultata na posmatranim skupovima podataka.

S obzirom da smo izneli tvrdnju da je dobar onaj algoritam koji daje sličan rezultat u svim slučajevima, odnosno vrednost standardne devijacije je minimalna, razmatraćemo u nastavku teksta vrednosti za standardnu devijaciju za tačnost klasifikacije. Tabela 7.7. prikazuje standardnu devijaciju za tačnost klasifikacije *Naïve Bayes* algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Može se uočiti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa, u svim slučajevima kod svih skupova podataka. Za razliku od IBk algoritma, kod *Naïve Bayes* algoritma imamo manja odstupanja u vrednostima standardne devijacije za tačnost klasifikacije.

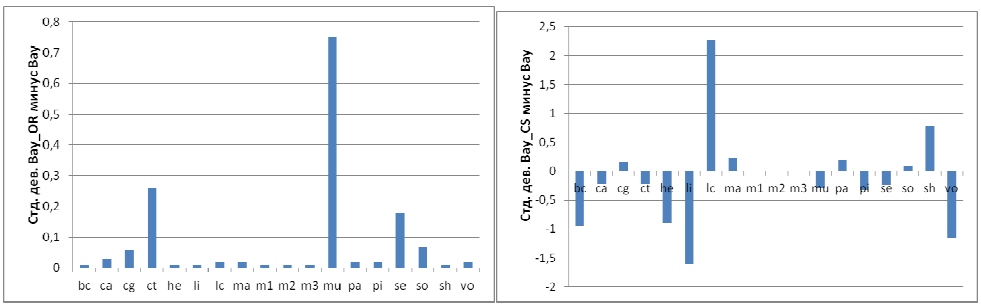
Na slikama 7.17, 7.18. i 7.19. prikazana je apsolutna razlika u vrednostima standardne devijacije za tačnost klasifikacije *Naïve Bayes* algoritma za originalni i redukovani skup podataka uz pomoć filter metoda.



Slika 7.17: Standardna devijacija za tačnost Bay_IG minus Bay i Bay_GR minus Bay



Slika 7.18: Standardna devijacija za tačnost Bay_SU minus Bay i Bay_RF minus Bay



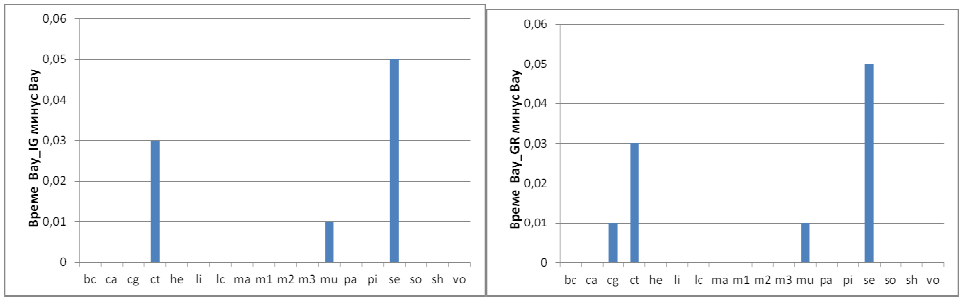
Slika 7.19: Standardna devijacija za tačnost Bay_OR minus Bay i Bay_CS minus Bay

Vrednosti na skali sa apsolutnom razlikom su manje na ovim slikama u odnosu na IBk algoritam, kako bi se uočile razlike u vrednostima između različitih metoda, s obzirom da su one zaista male. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuju, a ukoliko ona više odstupa od nule, u pozitivnom i negativnom smeru, to je i veće odstupanje između standardnih devijacija. Najmanje odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka, pokazuje metoda OR, dok najveće odstupanje ima metoda IG i CS, koje su kod nekih skupova podataka uspele da smanje, a kod nekih da povećaju standardnu devijaciju.

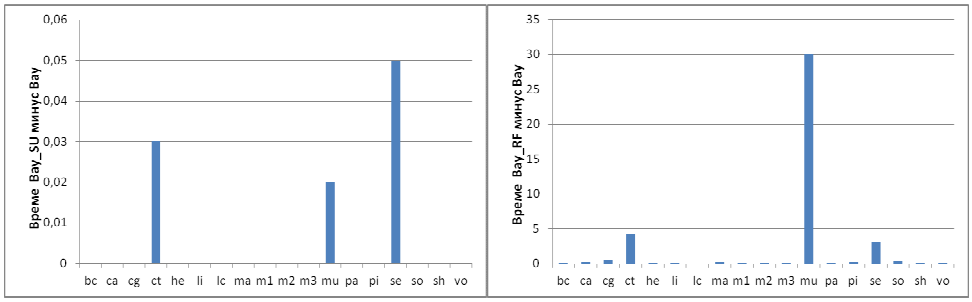
Tabela 7.8. Potrebno vreme za trening (u sekundama) *Naïve Bayes* algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda

Skup	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay _CS
bc	0.00	0.00	0.00	0.00	0.02 -	0.01 -	0.00
ca	0.00	0.00	0.00	0.00	0.19 -	0.03 -	0.00
cg	0.00	0.00	0.01	0.00	0.48 -	0.06 -	0.00
ct	0.01	0.04 -	0.04 -	0.04 -	4.26 -	0.27 -	0.04 -
he	0.00	0.00	0.00	0.00	0.01 -	0.01 -	0.00
li	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.00	0.00	0.00	0.00	0.17 -	0.02 -	0.00
m1	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m2	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m3	0.00	0.00	0.00	0.00	0.04 -	0.01 -	0.00
mu	0.01	0.02 -	0.02 -	0.03 -	30.15 -	0.76 -	0.02 -
pa	0.00	0.00	0.00	0.00	0.03 -	0.02 -	0.00
pi	0.00	0.00	0.00	0.00	0.16 -	0.02 -	0.00
se	0.01	0.06 -	0.06 -	0.06 -	3.13 -	0.19 -	0.06 -
so	0.00	0.00	0.00	0.00	0.42 -	0.07 -	0.00
sh	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
vo	0.00	0.00	0.00	0.00	0.06 -	0.02 -	0.00

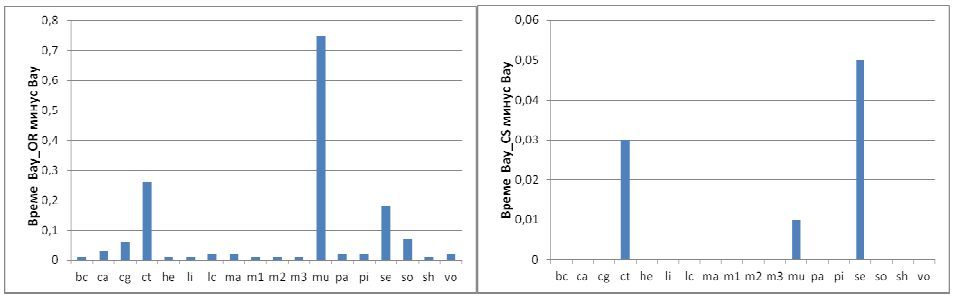
Potrebno vreme za trening *Naïve Bayes* algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda prikazano je u tabeli 7.8. Potrebno vreme za trening podataka *Naïve Bayes* klasifikatora za sve originalne skupove podataka je malo i iznosi najviše 0.01, dok za filter metode ono je nešto veće. Kod samo tri skupa podataka (*ct, mu* i *se*), ni jedna od metoda ne daje minimalno potrebno vreme za trening, dok kod svih ostalih skupova podataka u jednako ili više od pola slučajeva metode filtriranja daju minimalno potrebno vreme za trening.



Slika 7.20: Vreme treninga Bay_IG minus Bay i Bay_GR minus Bay (u sekundama)



Slika 7.21: Vreme treninga Bay_SU minus Bay i Bay_RF minus Bay (u sekundama)



Slika 7.22: Vreme treninga Bay_OR minus Bay i Bay_CS minus Bay (u sekundama)

Na slikama 7.20, 7.21. i 7.22. prikazana je apsolutna razlika u potrebnom vremenu za trening *Naïve Bayes* algoritma na osnovnom skupu podataka i *Naïve Bayes* algoritma sa različitim metodama filtriranja. Primenjeni metod filtriranja IG je u samo tri skupa podataka pokazao nešto lošije rezultate za potrebno vreme za trening i kod tih skupova podataka, rezultati su bili i statistički lošiji. Metod filtriranja GR je u četiri skupa podataka pokazao nešto lošije rezultate za potrebno vreme za trening, a kod 3 skupa podataka rezultati su bili i statistički lošiji.

Primenjeni metod filtriranja SU je u 3 skupa podataka pokazao lošije rezultate za potrebno vreme za trening od *Naïve Bayes* algoritma na osnovnom skupu podataka, a u tim skupovima podataka, rezultati su bili i statistički lošiji. Metod filtriranja RF je u skoro svim skupovima podataka (17 skupova) pokazao lošije rezultate za potrebno vreme za trening od *Naïve Bayes* algoritma na osnovnom skupu podataka, a u tim skupovima podataka, rezultati su bili i statistički lošiji.

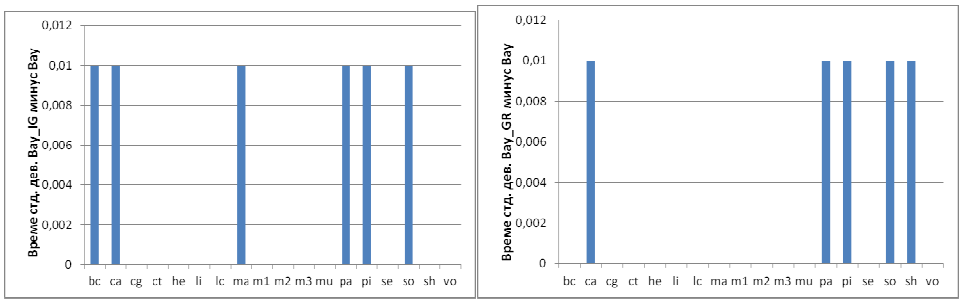
U svim skupovima podataka metod filtriranja OR je pokazao lošije rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, a ovi rezultati su bili i statistički lošiji. Primenjeni metod filtriranja CS je u samo 3 skupa podataka pokazao lošije rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, a u 3 skupa podataka, rezultati su bili i statistički lošiji.

Korišćenjem *Naïve Bayes* klasifikatora, možemo da zaključimo da su IG, GR, SU i CS metode filtriranja u najmanjem broju slučaja doveli do statistički lošijih rezultata za potrebno vreme za trening na posmatranim skupovima podataka.

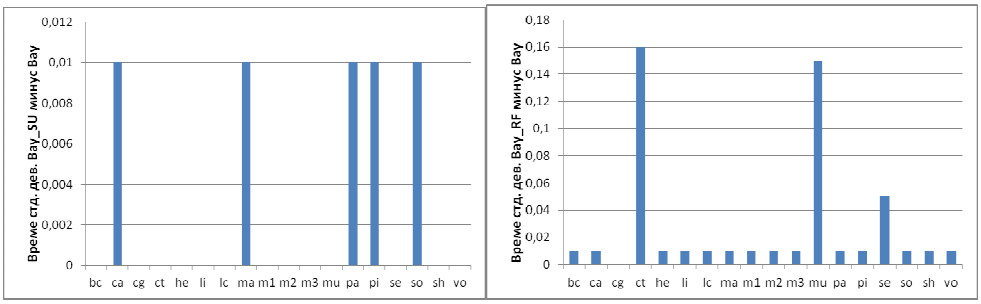
Tabela 7.9. Standardna devijacija za vreme treninga (u sekundama) *Naïve Bayes* algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	Bay	Bay_IG	Bay_GR	Bay _SU	Bay_RF	Bay_OR	Bay _CS
bc	0.00	0.01	0.00	0.00	0.01	0.01	0.00
ca	0.00	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ct	0.01	0.01	0.01	0.01	0.17	0.02	0.01
he	0.00	0.00	0.00	0.00	0.01	0.01	0.00
li	0.00	0.00	0.00	0.00	0.01	0.01	0.00
lc	0.00	0.00	0.00	0.00	0.01	0.01	0.00
ma	0.00	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m2	0.00	0.00	0.00	0.00	0.01	0.01	0.00
m3	0.00	0.00	0.00	0.00	0.01	0.01	0.00
mu	0.01	0.01	0.01	0.01	0.16	0.01	0.01
pa	0.00	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.00	0.01	0.01	0.01	0.01	0.01	0.01
se	0.01	0.01	0.01	0.01	0.06	0.01	0.01
so	0.00	0.01	0.01	0.01	0.01	0.01	0.01
sh	0.00	0.00	0.01	0.00	0.01	0.01	0.01
vo	0.00	0.00	0.00	0.00	0.01	0.01	0.00

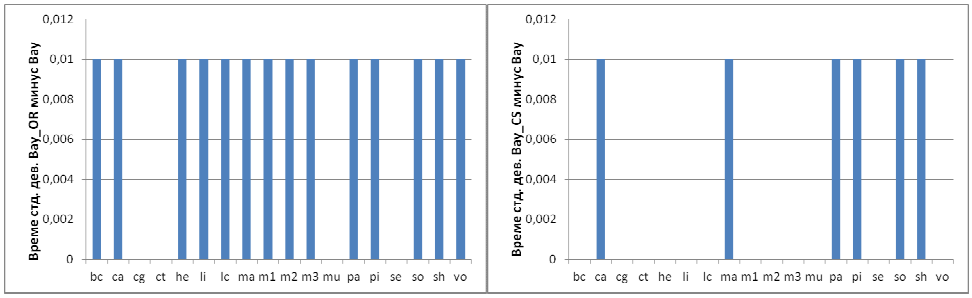
Tabela 7.9. prikazuje standardnu devijaciju za vreme treninga *Naïve Bayes* algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa. Nešto veće vrednosti za standardnu devijaciju za vreme treninga ima RF metoda filtriranja.



Slika 7.23: Standardna devijacija za vreme Bay_IG minus Bay i Bay_GR minus Bay



Slika 7.24: Standardna devijacija za vreme Bay_SU minus Bay i Bay_RF minus Bay



Slika 7.25: Standardna devijacija za vreme Bay_OR minus Bay i Bay_CS minus Bay

Na slikama 7.23, 7.24. i 7.25. prikazana je apsolutna razlika u vrednostima standardne devijacije za vreme treninga *Naïve Bayes* algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Najveće odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka pokazuje metoda RF, koja je kod svih skupova podataka osim jednog, uspela da poveća standardnu devijaciju.

7.4. SVM

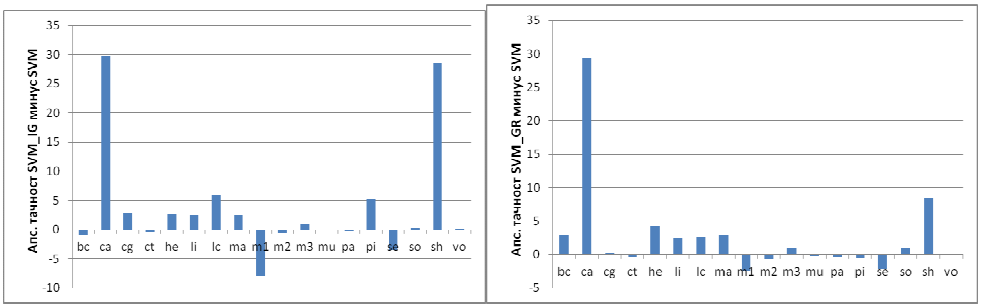
Uvidom u podatke prikazane u tabeli 7.10. možemo uočiti da korišćenjem SVM algoritma u deset skupova podataka (*ca*, *cg, li, ma, m1, m3, pa, pi, so* i *sh*) imamo dobijene rezultate za bar jednu od metoda filtriranja koji su statistički bolji od osnovnog klasifikatora. Samo u dva seta podataka, imamo značajno lošije podatke za sve metode filtriranja. Kod tri skupa podataka: *ca, m3* i *sh* sve metode filtriranja su statistički bolje od osnovnog klasifikatora.

Tabela 7.10. Tačnost klasifikacije SVM algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

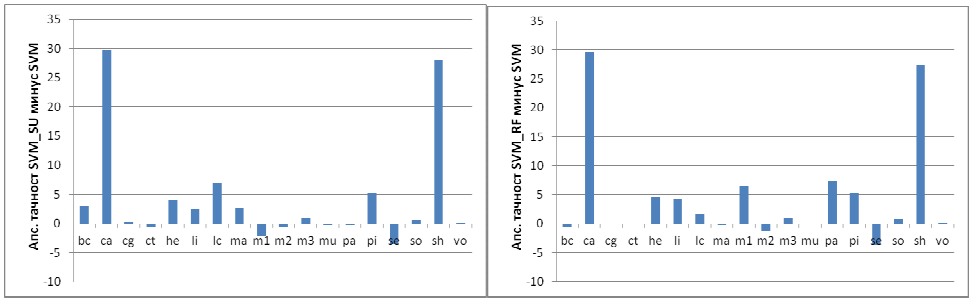
Skup	SVM	SVM_IG	SVM_GR	SVM_SU	SVM_RF	SVM_OR	SVM_CS
bc	72.18	71.24	75.09	75.30	71.48	72.18	74.32
ca	55.80	85.51 +	85.19 +	85.51 +	85.51 +	85.43 +	85.51 +
cg	70.00	72.85 +	70.25	70.24	70.00	72.12 +	70.23
ct	81.01	80.57 -	80.58 -	80.50 -	80.88	80.90	79.93 -
he	79.38	82.15	83.70	83.31	84.09	84.49	81.97
li	59.37	61.77	61.77	61.77	63.64 +	60.79	61.77
lc	72.67	78.58	75.25	79.58	74.33	74.08	75.00
ma	80.29	82.68 +	83.15 +	83.06 +	79.95	82.46	83.03 +
m1	91.37	83.32 -	88.96	89.14	97.83 +	85.26 -	83.32 -
m2	67.82	67.13 -	67.13 -	67.13 -	66.67 -	67.59 -	67.13 -
m3	96.30	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +	97.22 +
mu	100.00	99.98	99.72 -	99.72 -	99.99	99.98	99.98
pa	79.36	79.00	79.00	79.00	86.67 +	79.15	79.00
pi	65.11	70.36 +	64.59	70.36 +	70.36 +	64.49	70.36 +
se	63.98	60.52 -	61.85 -	60.52 -	60.36 -	60.52 -	60.52 -
so	93.63	93.88	94.55 +	94.20	94.44	93.59	93.62
sh	55.93	84.48 +	64.41 +	84.07 +	83.33 +	84.41 +	84.59 +
vo	95.61	95.63	95.63	95.63	95.63	95.63	95.63

Na slikama 7.26, 7.27. i 7.28. prikazana je apsolutna razlika u tačnosti klasifikacije SVM algoritma na osnovnom skupu podataka i SVM algoritma sa primenjenim različitim metodama filtriranja. Primenjeni metod filtriranja IG je u više od pola skupova podataka (11 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka, a u 6 skupova podataka rezultati su bili i statistički bolji. Metod filtriranja GR je u više od pola skupova podataka (11 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka. Kod metode GR, u 5 skupova podataka rezultati su bili i statistički bolji.

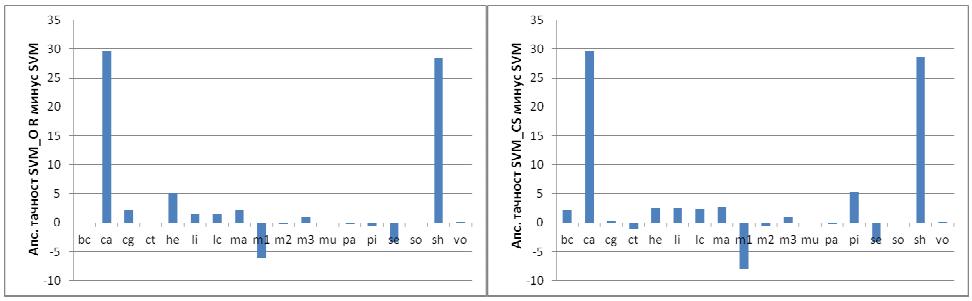
Primenjeni metod filtriranja SU je u dve trećine skupova podataka (12 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka, dok su u 5 skupova podataka rezultati bili i statistički bolji. Metod filtriranja RF je u dve trećine skupova podataka (12 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka. U 7 skupova podataka, rezultati su bili i statistički bolji.



Slika 7.26: Apsolutna tačnost klasifikacije SVM_IG minus SVM i SVM_GR minus SVM



Slika 7.27: Apsolutna tačnost klasifikacije SVM_SU minus SVM i SVM_RF minus SVM



Slika 7.28: Apsolutna tačnost klasifikacije SVM_OR minus SVM i SVM_CS minus SVM

Metod filtriranja OR je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka, a u 4 skupa podataka rezultati su bili i statistički bolji. Primenjeni metod filtriranja CS je u više od pola skupova podataka (11 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka, a u 5 skupova podataka rezultati su bili i statistički bolji.

Korišćenjem SVM klasifikatora, možemo da zaključimo da je RF metoda filtriranja u najvećem broju slučaja dovela do statistički boljih rezultata na posmatranim skupovima podataka.

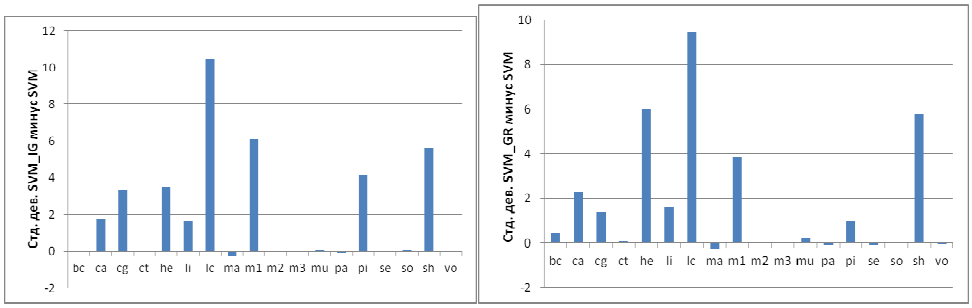
Tabela 7.11. prikazuje standardnu devijaciju za tačnost klasifikacije SVM algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz

tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa. Ove razlike su nešto veće u odnosu na *Naïve Bayes* algoritam, a nešto manje u odnosu na IBk algoritam. U slučaju *lc* skupa podataka za sve metode filtriranja dobijamo najveće vrednosti za standardnu devijaciju. Pored većih vrednosti u apsolutnim iznosima, kod SVM algoritma i ovog skupa podataka, primenom svih metoda filtriranja dobijamo značajno veće vrednosti za standardnu devijaciju.

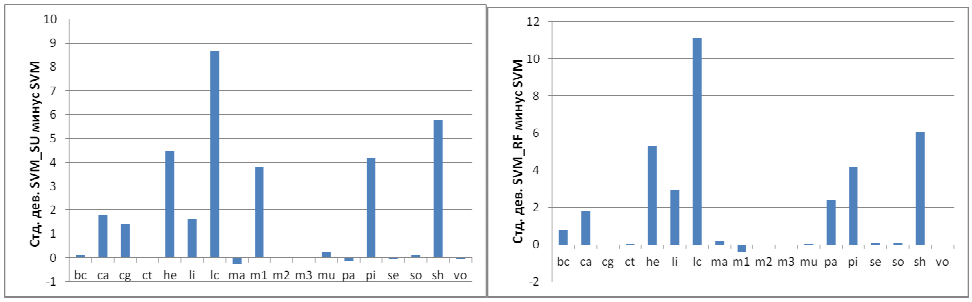
Tabela 7.11. Standardna devijacija za tačnost klasifikacije SVM algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	SVM	SVM_IG	SVM_GR	SVM_SU	SVM_RF	SVM_OR	SVM_CS
bc	5.86	5.86	6.29	5.97	6.64	5.86	5.97
ca	2.18	3.96	4.47	3.96	3.96	4.01	3.96
cg	0.00	3.35	1.38	1.44	0.00	2.58	1.34
ct	0.99	1.02	1.05	1.04	1.02	1.04	0.93
he	2.26	5.80	8.29	6.75	7.57	8.10	6.13
li	2.28	3.91	3.91	3.91	5.23	3.55	3.91
lc	11.12	21.59	20.56	19.76	22.24	21.12	23.27
ma	3.41	3.18	3.14	3.12	3.61	3.20	3.10
m1	3.10	9.20	6.98	6.93	2.71	8.68	9.20
m2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
m3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mu	0.00	0.09	0.23	0.23	0.04	0.09	0.09
pa	4.46	4.35	4.35	4.35	6.87	4.33	4.35
pi	0.34	4.53	1.32	4.53	4.53	1.63	4.53
se	3.47	3.43	3.36	3.43	3.53	3.43	3.43
so	2.22	2.31	2.25	2.35	2.28	2.17	2.63
sh	1.12	6.76	6.94	6.87	7.19	6.56	6.61
vo	2.77	2.76	2.76	2.76	2.76	2.76	2.76

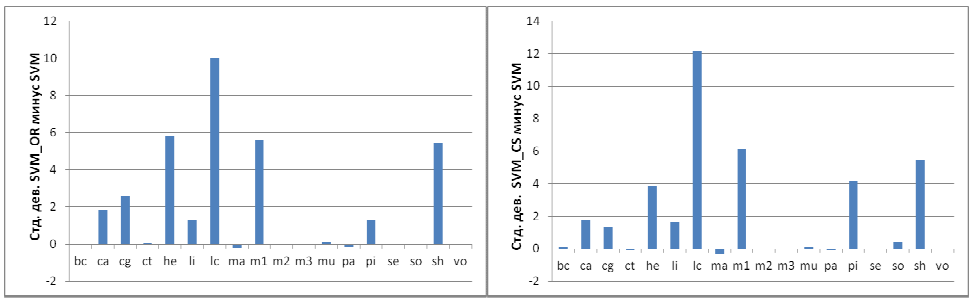
Slike 7.29, 7.30. i 7.31. prikazuju apsolutnu razliku u vrednostima standardne devijacije za tačnost klasifikacije SVM algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, u pozitivnom i negativnom smeru, to je i veće odstupanje između standardnih devijacija. Gotovo sve metode pokazuju odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka u istoj meri, s tim što možemo da zapazimo u odnosu na prethodne algoritme, da su ovo pozitivna odstupanja, odnosno standardna devijacija algoritma SVM sa prethodnom selekcijom atributa različitim metodama je veća u odnosu na standardni SVM algoritam.



Slika 7.29: Standardna devijacija za tačnost SVM_IG minus SVM i SVM_GR minus SVM



Slika 7.30: Standardna devijacija za tačnost SVM_SU minus SVM i SVM_RF minus SVM



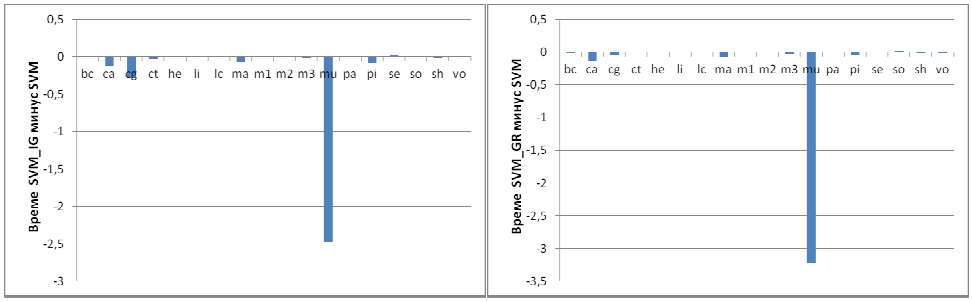
Slika 7.31: Standardna devijacija za tačnost SVM_OR minus SVM i SVM_CS minus SVM

U tabeli 7.12. prikazano je potrebno vreme za trening SVM algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda. Potrebno vreme za trening podataka SVM klasifikatora za sve originalne skupove podataka je nešto veće nego kod IBk i *Naïve Bayes* algoritma. Možemo uočiti da neke metode filtriranja kod SVM algoritma smanjuju, a neke povećavaju neophodno vreme za trening podataka. Kod svih skupova podataka, izuzev jednog (*se*), potrebno vreme za trening podataka bar jednom od metoda filtriranja je jednako ili manje od osnovnog klasifikatora.

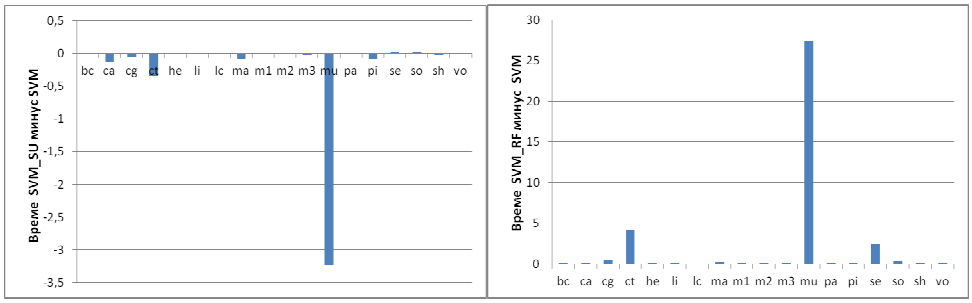
Tabela 7.12. Potrebno vreme za trening (u sekundama) SVM algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda

Skup	SVM	SVM_IG	SVM_GR	SVM_SU	SVM_RF	SVM_OR	SVM_CS
bc	0.02	0.02	0.01	0.01 +	0.03	0.03	0.01
ca	0.15	0.02 +	0.02 +	0.02 +	0.21 -	0.05 +	0.02 +
cg	0.42	0.13 +	0.38	0.37	0.83 -	0.50	0.38
ct	3.78	3.75	3.78	3.43 +	7.91 -	3.78	2.90 +
he	0.01	0.01	0.01	0.01	0.02	0.02	0.01
li	0.03	0.03	0.03	0.03	0.05 -	0.04 -	0.03
lc	0.01	0.01	0.01	0.01	0.01	0.03 -	0.01
ma	0.13	0.06 +	0.05 +	0.05 +	0.29 -	0.07 +	0.05 +
m1	0.04	0.04	0.04	0.04	0.07 -	0.05 -	0.04
m2	0.05	0.05	0.05	0.05	0.10 -	0.05	0.05
m3	0.03	0.01 +	0.01 +	0.01 +	0.06 -	0.02	0.01 +
mu	3.72	1.24 +	0.49 +	0.48 +	31.05 -	1.51 +	0.77 +
pa	0.01	0.01	0.01	0.01	0.03 -	0.03 -	0.01
pi	0.15	0.06 +	0.10 +	0.06 +	0.21 -	0.12 +	0.06 +
se	3.72	3.74	3.73	3.74	6.15 -	3.74	3.74
so	0.89	0.88	0.91	0.91	1.28 -	0.96	0.85
sh	0.03	0.01 +	0.02	0.01 +	0.04 -	0.03	0.01 +
vo	0.02	0.01	0.01	0.01	0.07 -	0.03 -	0.01 +

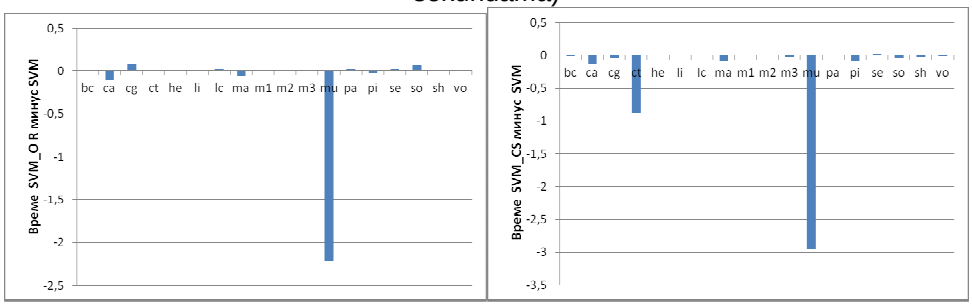
Na slikama 7.32, 7.33. i 7.34. prikazana je apsolutna razlika u potrebnom vremenu za trening SVM algoritma na osnovnom skupu podataka i SVM algoritma sa različitim metodama filtriranja. Primenjeni metod filtriranja IG je u 10 skupova podataka pokazao nešto bolje rezultate za potrebno vreme za trening, a u 7 skupova podataka rezultati su bili statistički bolji. Metod filtriranja GR je u 9 skupova podataka pokazao nešto bolje rezultate za potrebno vreme za trening, a u 5 skupova podataka rezultati su bili statistički bolji.



Slika 7.32: Vreme treninga SVM_IG minus SVM i SVM_GR minus SVM (u sekundama)



Slika 7.33: Vreme treninga SVM_SU minus SVM i SVM_RF minus SVM (u sekundama)



Slika 7.34: Vreme treninga SVM_OR minus SVM i SVM_CS minus SVM (u sekundama)

Primenjeni metod filtriranja SU je u više od pola skupova podataka (10 skupova) pokazao bolje rezultate za potrebno vreme za trening od SVM algoritma na osnovnom skupu podataka, a u 8 skupova podataka rezultati su bili i statistički bolji. Metod filtriranja RF je u svim skupovima podataka pokazao lošije ili iste rezultate za potrebno vreme za trening od SVM algoritma na osnovnom skupu podataka, a u skoro svim skupovima podataka rezultati su bili i statistički lošiji.

Metod filtriranja OR je u pet skupova podataka pokazao bolje rezultate od SVM algoritma na osnovnom skupu podataka, a u četiri slučaja rezultati su bili i statistički bolji. Primenjeni metod filtriranja CS je u 11 skupova podataka pokazao bolje rezultate od SVM algoritma na osnovnom skupu podataka, a u 8 skupa podataka rezultati su bili i statistički bolji.

Korišćenjem SVM klasifikatora, možemo da zaključimo da su SU i CS metode filtriranja u najvećem broju slučaja doveli do statistički boljih rezultata za potrebno vreme za trening na posmatranim skupovima podataka.

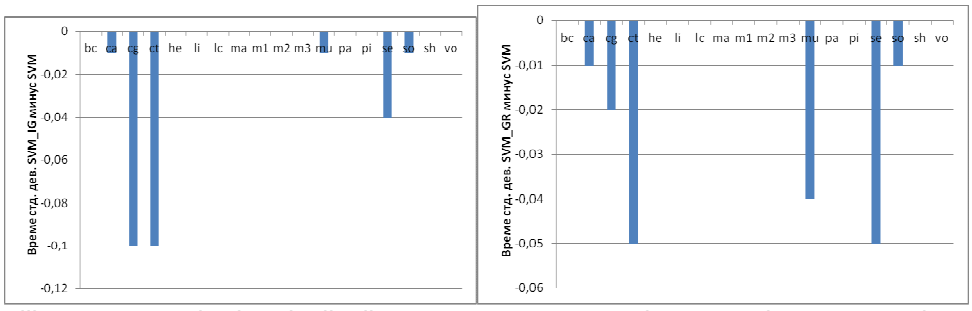
Tabela 7.13. prikazuje standardnu devijaciju za vreme treninga SVM algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Kod GR i IG metode filtriranja vrednosti za standardnu devijaciju za vreme treninga su iste ili manje u odnosu na osnovni algoritam učenja.

Na slikama 7.35, 7.36. i 7.37. prikazana je apsolutna razlika u vrednostima standardne devijacije za vreme treninga SVM algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od

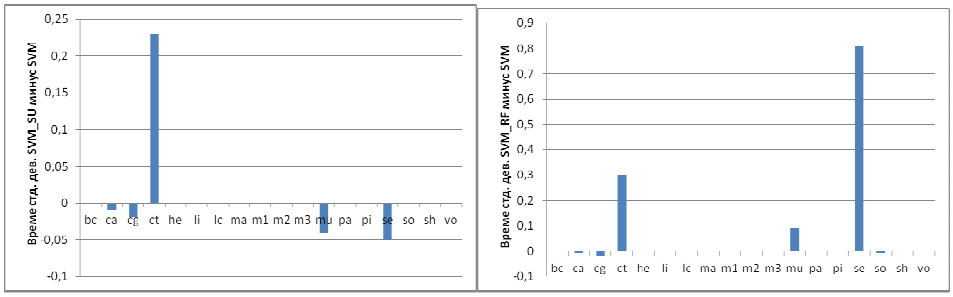
nule, to je i veće odstupanje između standardnih devijacija. Najveće odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka pokazuje metoda RF.

Tabela 7.13. Standardna devijacija za vreme treninga (u sekundama) SVM algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

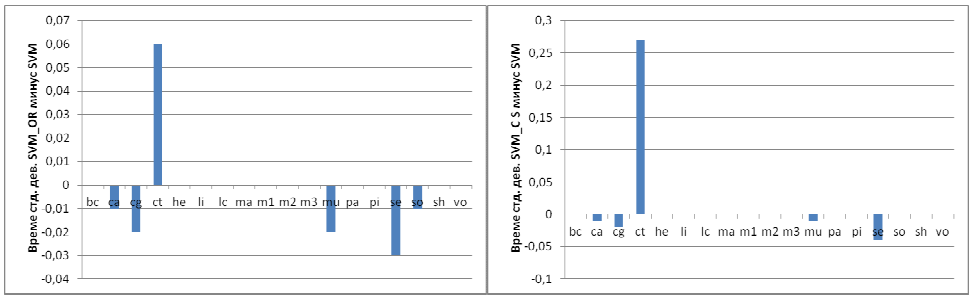
Skup	SVM	SVM_IG	SVM_GR	SVM_SU	SVM_RF	SVM_OR	SVM_CS
bc	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ca	0.02	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.11	0.01	0.09	0.09	0.09	0.09	0.09
ct	0.13	0.03	0.08	0.36	0.43	0.19	0.40
he	0.01	0.01	0.01	0.01	0.01	0.01	0.01
li	0.01	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ma	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m2	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m3	0.01	0.01	0.01	0.01	0.01	0.01	0.01
mu	0.07	0.06	0.03	0.03	0.16	0.05	0.06
pa	0.01	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.01	0.01	0.01	0.01	0.01	0.01	0.01
se	0.08	0.04	0.03	0.03	0.89	0.05	0.04
so	0.10	0.09	0.09	0.10	0.09	0.09	0.10
sh	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.01	0.01	0.01	0.01	0.01	0.01	0.01



Slika 7.35: Standardna devijacija za vreme SVM_IG minus SVM i SVM_GR minus SVM



Slika 7.36: Standardna devijacija za vreme SVM_SU minus SVM i SVM_RF minus SVM



Slika 7.37: Standardna devijacija za vreme SVM_OR minus SVM i SVM_CS minus SVM

7.5. J48

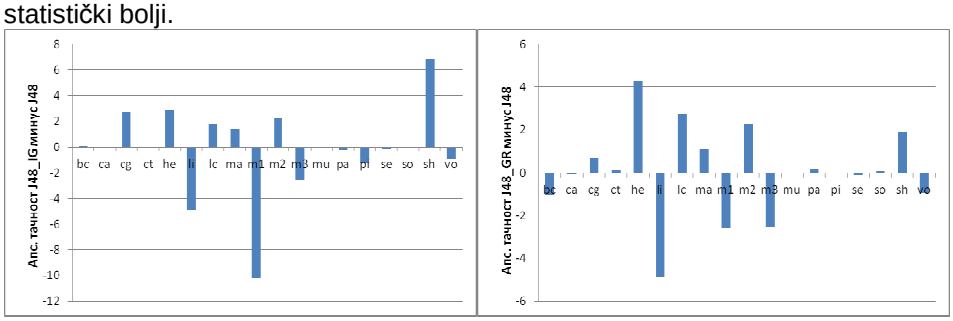
Na osnovu prikazanih podataka u tabeli 7.14. za tačnost klasifikacije J48 algoritma možemo uočiti da u četiri seta podataka (*cg*, *ma, m2* i *sh*) imamo dobijene rezultate za bar jednu od metoda filtriranja koji su statistički bolji od osnovnog klasifikatora. U svim setovima podataka osim jednog seta podataka *m3*, nemamo značajno lošije podatke za sve metode filtriranja, što znači da uvek možemo izabrati metodu za dati skup podataka koja ima statistički bolje rezultate ili rezultate koji su približni originalnom skupu podataka. Kod samo jednog skupa podataka *m3* sve metode filtriranja su statistički lošije od osnovnog klasifikatora.

Slike 7.38, 7.39. i 7.40. prikazuju apsolutnu razliku u tačnosti klasifikacije J48 algoritma na osnovnom skupu podataka i J48 algoritma sa različitim metodama filtriranja. Primenjeni metod filtriranja IG je u pola skupova podataka (9 skupova) pokazao iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka, dok su u 3 skupa podataka rezultati i statistički bolji. Metod filtriranja GR je u više od pola skupova podataka (11 skupova) pokazao iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka, a u ni u jednom skupu podataka rezultat nije bio statistički bolji.

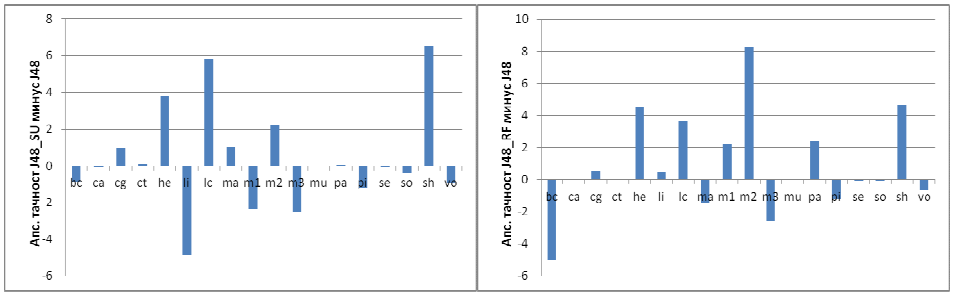
Tabela 7.14. Tačnost klasifikacije J48 algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48_CS
bc	74.28	74.35	73.23	73.40	69.27 -	73.10	72.84
ca	85.57	85.51	85.51	85.51	85.51	85.51	85.51
cg	71.25	73.95 +	71.91	72.22	71.78	71.49	71.94
ct	98.57	98.57	98.69	98.70	98.57	98.63	98.92
he	79.22	82.10	83.51	83.00	83.78	83.96	81.92
li	65.84	60.97	60.97	60.97	66.32	64.47	60.97
lc	79.25	81.00	82.00	85.08	82.92	80.58	81.33
ma	82.19	83.57 +	83.29	83.19	80.76	82.60	83.16
m1	97.80	87.63 -	95.21	95.43	100.00	87.63 -	89.99
m2	63.48	65.72	65.72	65.72	71.74 +	71.93 +	65.72
m3	98.92	96.39 -	96.39 -	96.39 -	96.39 -	96.39 -	96.39 -
mu	100.00	100.00	100.00	100.00	100.00	100.00	100.00
pa	84.74	84.49	84.90	84.79	87.14	84.48	84.74
pi	74.49	73.27	74.49	73.27	73.27	73.63	73.27
se	96.79	96.69	96.69	96.70	96.71	96.67	96.69
so	91.78	91.70	91.84	91.38	91.70	91.70	91.71
sh	78.15	84.96 +	80.04	84.63 +	82.81 +	85.07 +	85.22 +
vo	96.57	95.63	95.63	95.63	95.93	95.63	95.63

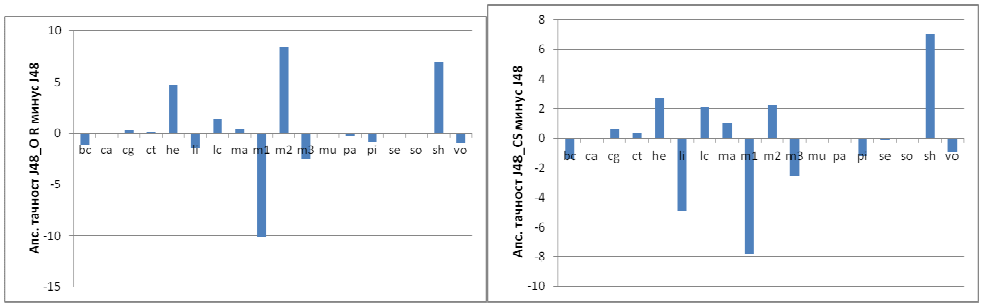
Primenjeni metod filtriranja SU je u pola skupova podataka (9 skupova) pokazao iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka, dok je u jednom skupu podataka rezultat bio i statistički bolji. Metod filtriranja RF je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka, dok je u 2 skupa podataka rezultat bio i statistički bolji.



Slika 7.38: Apsolutna tačnost klasifikacije J48_IG minus J48 i J48_GR minus J48



Slika 7.39: Apsolutna tačnost klasifikacije J48_SU minus J48 i J48_RF minus J48



Slika 7.40: Apsolutna tačnost klasifikacije J48_OR minus J48 i J48_CS minus J48

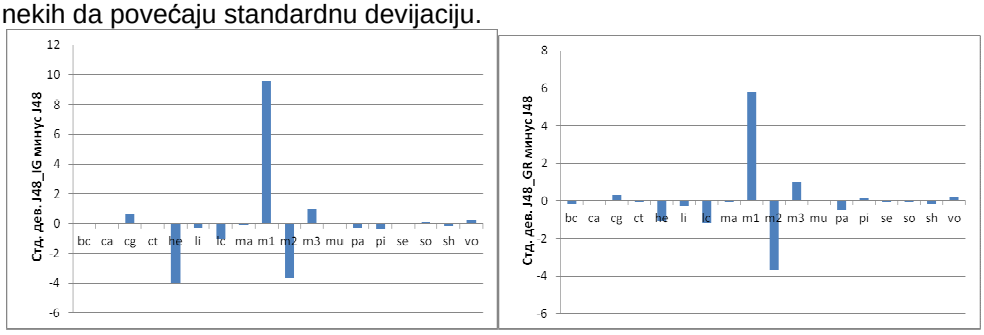
Tabela 7.15. Standardna devijacija za tačnost klasifikacije J48 algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48_CS
bc	6.05	6.07	5.88	5.43	5.22	5.55	5.55
ca	3.96	3.96	3.96	3.96	3.96	3.96	3.96
cg	3.17	3.83	3.48	3.42	3.40	3.50	3.48
ct	0.89	0.92	0.86	0.91	0.89	0.88	0.77
he	9.57	5.60	8.50	7.60	8.12	8.04	5.93
li	7.40	7.12	7.12	7.12	8.24	7.91	7.12
lc	21.50	20.48	20.33	16.76	17.10	21.02	19.96
ma	3.21	3.14	3.13	3.09	3.62	3.11	3.07
m1	3.45	13.03	9.25	9.10	0.00	13.03	12.07
m2	4.48	0.79	0.79	0.79	5.34	5.39	0.79
m3	1.23	2.20	2.20	2.20	2.20	2.20	2.20
mu	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pa	8.01	7.73	7.54	7.57	7.31	7.62	7.74
pi	5.27	4.93	5.41	4.93	4.93	5.57	4.93
se	1.29	1.28	1.28	1.27	1.33	1.28	1.27
so	3.19	3.30	3.17	3.19	3.36	3.17	3.04
sh	7.42	7.23	7.29	7.21	6.95	6.67	6.75
vo	2.56	2.76	2.76	2.76	2.71	2.76	2.76

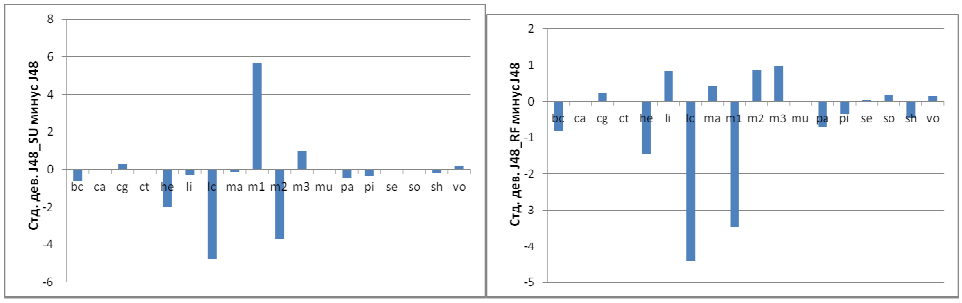
Metod filtriranja OR je u nešto manje od pola skupova podataka (8 skupova) pokazao iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka, a u 2 skupa podataka rezultati su bili i statistički bolji. Primenjeni metod filtriranja CS je u pola skupova podataka (9 skupova) pokazao iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka, a u 1 skupu podataka rezultat je bio i statistički bolji.

Korišćenjem J48 klasifikatora, možemo da zaključimo da je IG metoda filtriranja u najvećem broju slučaja dovela do statistički boljih rezultata na posmatranim skupovima podataka.

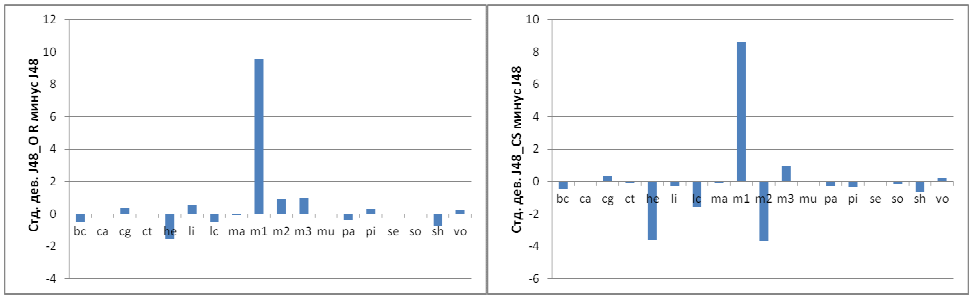
Na slikama 7.41, 7.42. i 7.43. prikazana je apsolutna razlika u vrednostima standardne devijacije za tačnost klasifikacije J48 algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuju, a ukoliko ona više odstupa od nule, u pozitivnom i negativnom smeru, to je i veće odstupanje između standardnih devijacija. Najmanje odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka, pokazuje metoda RF, dok najveće odstupanje imaju metode IG, OR i CS, koje kod nekih skupova podataka su uspele da smanje, a kod nekih da povećaju standardnu devijaciju.



Slika 7.41: Standardna devijacija za tačnost J48_IG minus J48 i J48_GR minus J48



Slika 7.42: Standardna devijacija za tačnost J48_SU minus J48 i J48_RF minus J48



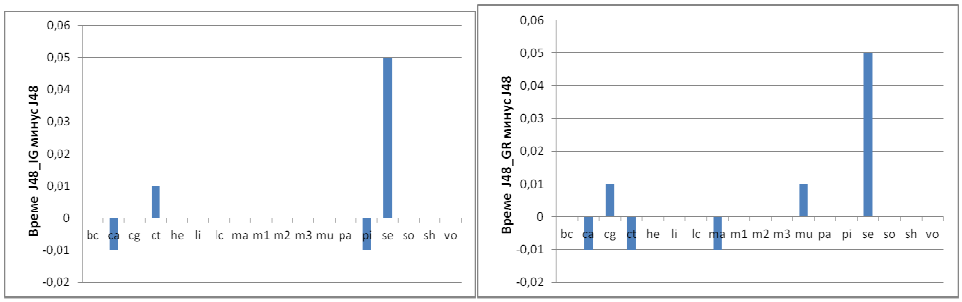
Slika 7.43: Standardna devijacija za tačnost J48_OR minus J48 i J48_CS minus J48

Tabela 7.16. Potrebno vreme za trening (u sekundama) J48 algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda

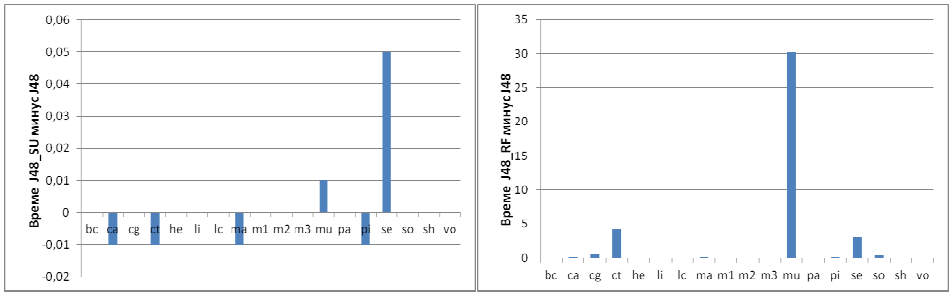
Skup	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48_CS
bc	0.00	0.00	0.00	0.00	0.02 -	0.01 -	0.00
ca	0.01	0.00	0.00	0.00	0.19 -	0.03 -	0.00
cg	0.01	0.01 +	0.02	0.01	0.49 -	0.07 -	0.01
ct	0.11	0.12 -	0.10 +	0.10 +	4.33 -	0.34 -	0.07 +
he	0.00	0.00	0.00	0.00	0.01 -	0.01 -	0.00
li	0.00	0.00	0.00	0.00	0.03 -	0.01	0.00
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.01	0.01	0.00	0.00	0.18 -	0.02 -	0.00
m1	0.00	0.00	0.00	0.00	0.00	0.05 -	0.01 -
m2	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
m3	0.00	0.00	0.00	0.00	0.05 -	0.01 -	0.00
mu	0.03	0.03	0.04	0.04	30.29 -	0.78 -	0.03
pa	0.01	0.01	0.01	0.01	0.03 -	0.03 -	0.01
pi	0.01	0.00	0.01	0.00	0.16 -	0.03 -	0.00
se	0.09	0.14 -	0.14 -	0.14 -	3.21 -	0.27 -	0.14 -
so	0.01	0.01	0.01	0.01	0.43 -	0.08 -	0.01
sh	0.00	0.00	0.00	0.00	0.03 -	0.01 -	0.00
vo	0.00	0.00	0.00	0.00	0.06 -	0.02 -	0.00

S obzirom na tvrdnju koju smo izneli da je dobar onaj algoritam koji daje sličan rezultat u svim slučajevima, odnosno vrednost za standardnu devijaciju je minimalna, razmatraćemo standardnu devijaciju za tačnost klasifikacije za J48 algoritam. Tabela 7.15. prikazuje standardnu devijaciju za tačnost klasifikacije J48 algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa, osim u slučaju *m1* skupa podataka. U slučaju *m1* skupa podataka za sve metode filtriranja dobijamo veliku vrednost za standardnu devijaciju, osim metode RF.

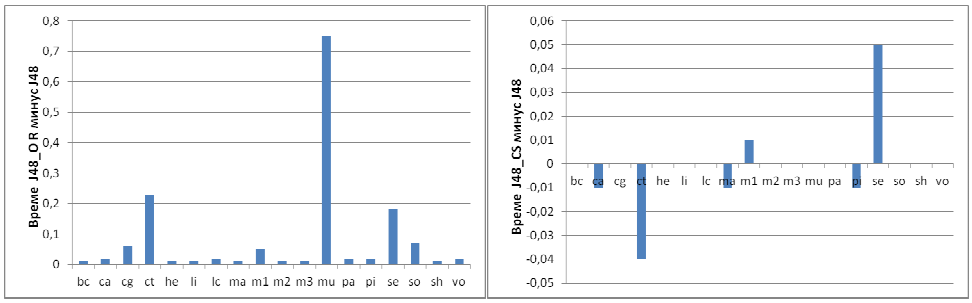
Najmanja odstupanja u standardnoj devijaciji pokazuje metoda RF, u odnosu na druge metode, gde je za pojedine skupove podataka uspela da dobije kako manje, tako i veće vrednosti za standardnu devijaciju.



Slika 7.44: Vreme treninga J48_IG minus J48 i J48_GR minus J48 (u sekundama)



Slika 7.45: Vreme treninga J48_SU minus J48 i J48_RF minus J48 (u sekundama)



Slika 7.46. Vreme treninga J48_OR minus J48 i J48_CS minus J48 (u sekundama)

Tabela 7.16. prikazuje potrebno vreme za trening J48 algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda. Potrebno vreme za trening podataka J48 klasifikatora za sve originalne skupove podataka iznosi najviše 0.11, dok za filter metode ono u nekim slučajevima veće, a u nekim manje. Za samo jedan skup podataka (*se*), ni jedna od metoda ne daje isto ili manje potrebno vreme za trening podataka, dok kod svih ostalih skupova podataka bar jedna od metoda filtriranja daje isto ili manje vreme za trening podataka kao kod originalnog skupa podataka.

Na slikama 7.44, 7.45. i 7.46. prikazana je apsolutna razlika u potrebnom vremenu za trening J48 algoritma na osnovnom skupu podataka i J48 algoritma sa različitim metodama filtriranja. Metod filtriranja IG je u samo dva skupa podataka pokazao nešto lošije rezultate za potrebno vreme za trening i kod tih skupova podataka, rezultati su bili i statistički lošiji. Metod filtriranja GR je u samo tri skupa podataka pokazao nešto lošije rezultate za potrebno vreme za trening i kod 1 skupa podataka rezultat je bio i statistički lošiji.

Primenjeni metod filtriranja SU je samo u 2 skupa podataka pokazao lošije rezultate za potrebno vreme za trening od J48 algoritma na osnovnom skupu podataka, a u 1 skupu podataka rezultat je bio i statistički lošiji. Metod filtriranja RF je u 16 skupova podataka pokazao lošije rezultate za potrebno vreme za trening od J48 algoritma na osnovnom skupu podataka, a u skoro svim skupovima podataka rezultati su bili i statistički lošiji.

Metod filtriranja OR je u svim skupovima podataka pokazao lošije rezultate od J48 algoritma na osnovnom skupu podataka, a ovi rezultati su bili u gotovo svim slučajevima i statistički lošiji. Primenjeni metod filtriranja CS je u samo 2 skupa podataka pokazao lošije rezultate od J48 algoritma na osnovnom skupu podataka, a u 2 skupa podataka rezultati su bili i statistički lošiji.

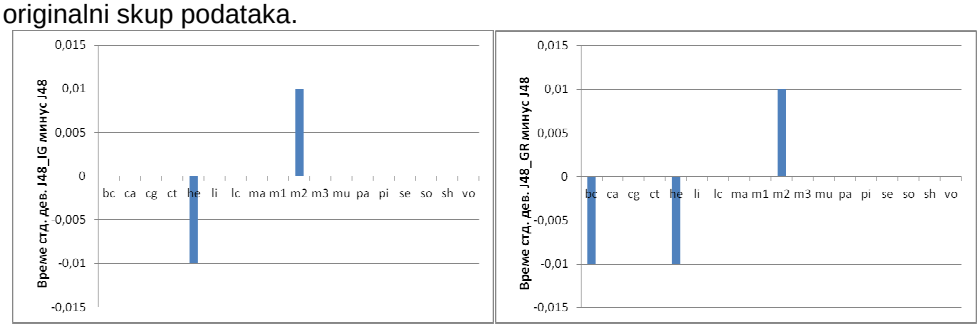
Korišćenjem J48 klasifikatora, možemo da zaključimo da su GR i SU metode filtriranja u najmanjem broju slučaja dovele do statistički lošijih rezultata za potrebno vreme za trening na posmatranim skupovima podataka.

Tabela 7.17. Standardna devijacija za vreme treninga (u sekundama) J48 algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

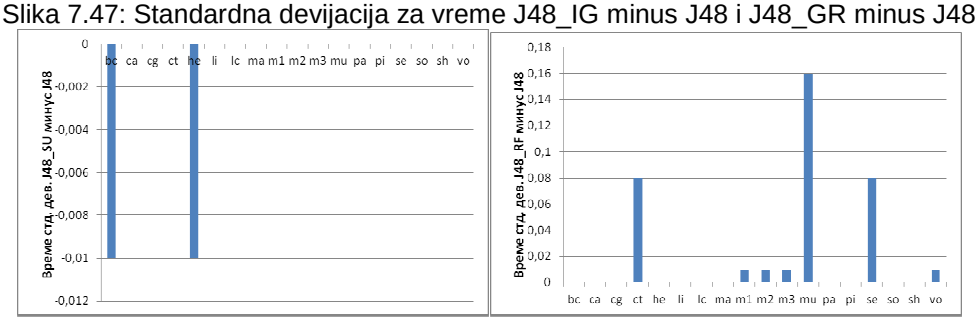
Skup	J48	J48_IG	J48_GR	J48_SU	J48_RF	J48_OR	J48_CS
bc	0.01	0.01	0.00	0.00	0.01	0.01	0.00
ca	0.01	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ct	0.01	0.01	0.01	0.01	0.09	0.01	0.01
he	0.01	0.00	0.00	0.00	0.01	0.01	0.00
li	0.01	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.00	0.00	0.00	0.00	0.00	0.01	0.00
ma	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.00	0.00	0.00	0.01	0.01	0.01
m2	0.00	0.01	0.01	0.00	0.01	0.01	0.00
m3	0.00	0.00	0.00	0.00	0.01	0.01	0.00
mu	0.01	0.01	0.01	0.01	0.17	0.02	0.00
pa	0.01	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.01	0.01	0.01	0.01	0.01	0.01	0.01
se	0.01	0.01	0.01	0.01	0.09	0.01	0.01
so	0.01	0.01	0.01	0.01	0.01	0.01	0.01
sh	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.00	0.00	0.00	0.00	0.01	0.01	0.00

Tabela 7.17. prikazuje standardnu devijaciju za vreme treninga J48 algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa. Nešto veće vrednosti za standardnu devijaciju za vreme treninga ima RF metoda filtriranja.

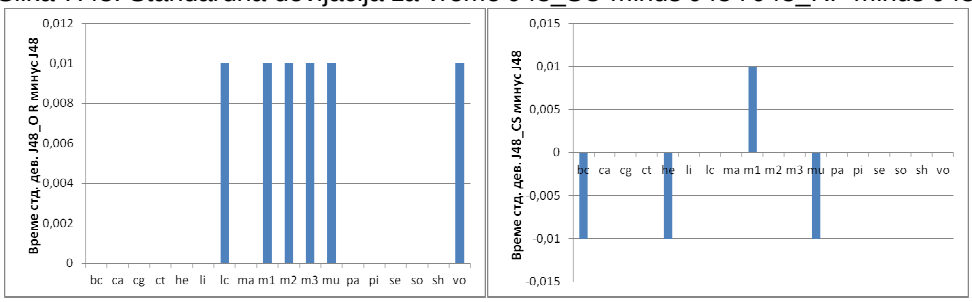
Na slikama 7.47, 7.48. i 7.49. prikazana je apsolutna razlika u vrednostima standardne devijacije za vreme treninga J48 algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Metoda SU za sve skupove podataka ima manje ili iste vrednosti za standardnu devijaciju za vreme treninga u odnosu na originalni skup podataka.



Slika 7.47: Standardna devijacija za vreme J48_IG minus J48 i J48_GR minus J48



Slika 7.48: Standardna devijacija za vreme J48_SU minus J48 i J48_RF minus J48



Slika 7.49: Standardna devijacija za vreme J48_OR minus J48 i J48_CS minus J48

7.6. RBF mreže

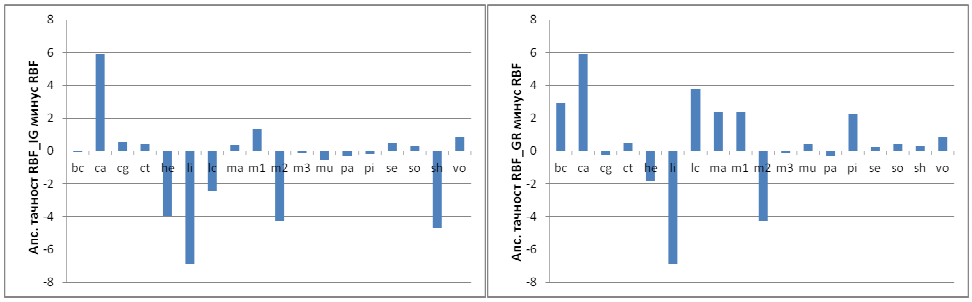
Za tačnost klasifikacije kod RBF algoritma možemo uočiti na osnovu tabele 7.18. da u tri seta podataka (*ca*, *m1* i *se*) imamo dobijene rezultate za bar jednu od metoda filtriranja koji su statistički bolji od osnovnog klasifikatora.

Tabela 7.18. Tačnost klasifikacije RBF algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

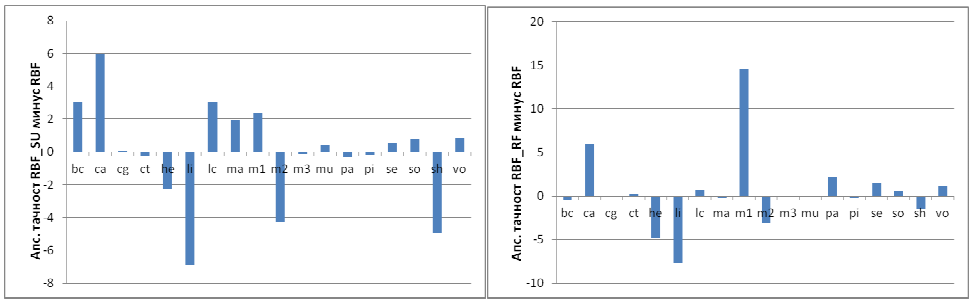
Skup	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	71.41	71.34	74.32	74.46	71.00	71.20	73.62
ca	79.55	85.51 +	85.43 +	85.51 +	85.51 +	85.10 +	85.51 +
cg	73.58	74.12	73.33	73.64	73.54	73.16	73.54
ct	97.93	98.35	98.41	97.65	98.13	96.90	96.27 -
he	85.29	81.31	83.45	83.05	80.49	82.69	81.25
li	65.06	58.16 -	58.16 -	58.16 -	57.33 -	60.96	58.16 -
lc	76.00	73.58	79.75	79.00	76.75	72.92	74.92
ma	77.31	77.66	79.67	79.24	77.07	77.51	79.16
m1	75.36	76.70	77.76	77.76	90.01 +	75.37	76.70
m2	67.82	63.53	63.54	63.53	64.77	64.77	63.53
m3	96.54	96.39	96.39	96.39	96.39	96.39	96.39
mu	98.61	98.06	98.99	98.99	98.43	98.55	98.55
pa	81.22	80.92	80.92	80.92	83.39	81.98	80.67
pi	74.04	73.84	76.28	73.84	73.84	75.32	73.84
se	87.31	87.84	87.56	87.84	88.88 +	87.84	87.84
so	90.79	91.11	91.20	91.59	91.29	90.57	91.42
sh	83.11	78.44	83.44	78.15	81.56	78.44	78.52
vo	93.73	94.60	94.60	94.60	94.92	95.63	95.63

Ni u jednom setu podataka, nemamo značajno lošije podatke za sve metode filtriranja, što znači da uvek možemo izabrati metodu za dati skup podataka koja ima statistički bolje rezultate ili rezultate koji su približni originalnom skupu podataka. Kod jednog skupa podataka (*ca*) sve metode filtriranja su statistički bolje od osnovnog klasifikatora.

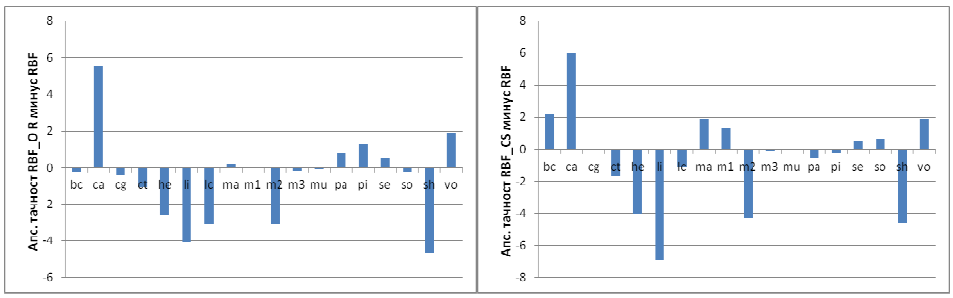
Na slikama 7.50, 7.51. i 7.52. prikazana je apsolutna razlika u tačnosti klasifikacije RBF algoritma na osnovnom skupu podataka i RBF algoritma sa različitim metodama filtriranja. Primenjeni metod filtriranja IG je u skoro pola skupova podataka (8 skupova) pokazao iste ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, dok u 1 skupu podataka rezultat je bio i statistički bolji. Metod filtriranja GR je u dve trećine skupova podataka (12 skupova) pokazao iste ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, dok u samo 1 skupu podataka rezultat je bio i statistički bolji.



Slika 7.50: Apsolutna tačnost klasifikacije RBF_IG minus RBF i RBF_GR minus RBF



Slika 7.51: Apsolutna tačnost klasifikacije RBF_SU minus RBF i RBF_RF minus RBF



Slika 7.52: Apsolutna tačnost klasifikacije RBF_OR minus RBF i RBF_CS minus RBF

Primenjeni metod filtriranja SU je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, dok u 1 skupu podataka rezultat je bio i statistički bolji. Metod filtriranja RF je u manje od pola skupova podataka (8 skupova) pokazao iste ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, dok u 3 skupa podataka rezultati su bili i statistički bolji.

Metod filtriranja OR je u nešto manje od pola skupova podataka (7 skupova) pokazao iste ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, a u 1 skupu podataka rezultat je bio i statistički bolji. Primenjeni metod filtriranja CS je u nešto manje od pola skupova podataka (7 skupova) pokazao iste

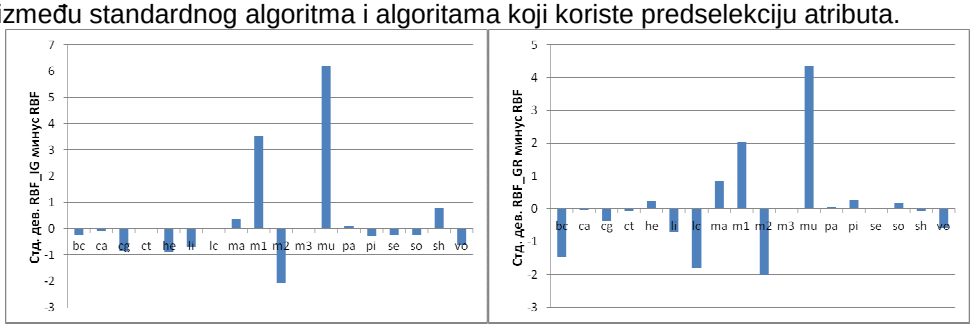
ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, a u 1 skupu podataka, rezultat je bio i statistički bolji.

Korišćenjem RBF klasifikatora, možemo da zaključimo da je RF metoda filtriranja u najvećem broju slučaja dovela do statistički boljih rezultata na posmatranim skupovima podataka.

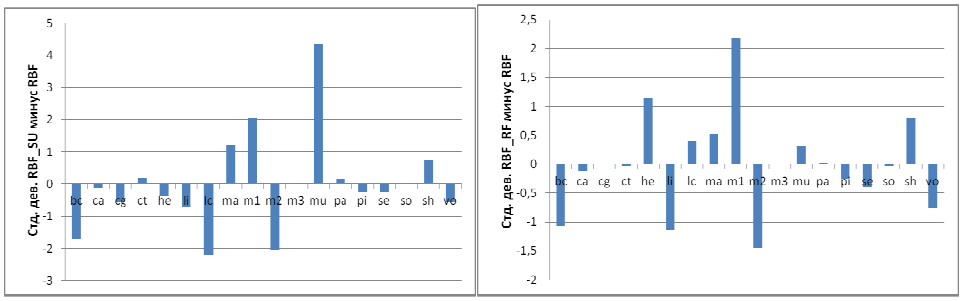
Tabela 7.19. Standardna devijacija za tačnost klasifikacije RBF algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	7.88	7.66	6.41	6.16	6.81	8.34	6.28
ca	4.07	3.96	4.03	3.96	3.96	4.14	3.96
cg	4.30	3.46	3.92	3.74	4.31	4.25	4.03
ct	1.02	1.02	0.94	1.21	1.00	2.19	1.29
he	8.29	7.38	8.54	7.90	9.44	8.25	7.51
li	8.80	8.10	8.10	8.10	7.66	9.62	8.10
lc	22.91	22.91	21.10	20.70	23.31	22.17	22.52
ma	3.31	3.67	4.14	4.51	3.83	4.35	4.50
m1	5.92	9.44	7.97	7.97	8.10	7.97	9.44
m2	6.24	4.19	4.21	4.19	4.79	4.79	4.19
m3	2.19	2.20	2.20	2.20	2.20	2.20	2.20
mu	0.58	6.77	4.93	4.93	0.90	4.86	4.86
pa	7.37	7.49	7.42	7.53	7.39	7.24	7.35
pi	4.91	4.65	5.18	4.65	4.65	5.31	4.65
se	2.15	1.91	2.15	1.89	1.76	1.91	1.90
so	2.92	2.69	3.09	2.93	2.90	2.92	3.24
sh	6.50	7.28	6.44	7.25	7.29	7.13	7.28
vo	3.87	3.25	3.28	3.30	3.10	2.76	2.76

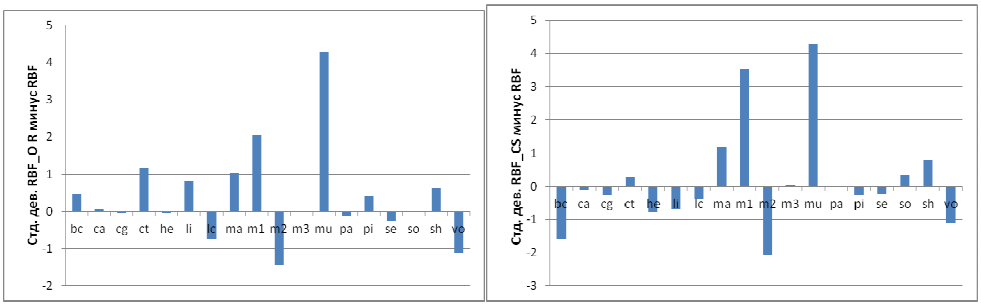
Tabela 7.19. prikazuje standardnu devijaciju za tačnost klasifikacije RBF algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa.



Slika 7.53: Standardna devijacija za tačnost RBF_IG minus RBF i RBF_GR minus RBF



Slika 7.54: Standardna devijacija za tačnost RBF_SU minus RBF i RBF_RF minus RBF



Slika 7.55: Standardna devijacija za tačnost RBF_OR minus RBF i RBF_CS minus RBF

Tabela 7.20. Potrebno vreme za trening (u sekundama) RBF algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda

Skup	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	0.01	0.01	0.01	0.00	0.02 -	0.02	0.01
ca	0.03	0.01 +	0.01 +	0.01 +	0.20 -	0.04 -	0.01 +
cg	0.05	0.02 +	0.04	0.04	0.53 -	0.10 -	0.04
ct	0.39	0.44	0.34	0.34	4.57 -	0.55 -	0.34
he	0.01	0.00	0.00	0.00	0.01	0.02 -	0.00
li	0.01	0.01	0.01	0.01	0.03 -	0.02	0.01
lc	0.00	0.00	0.00	0.00	0.00	0.02 -	0.00
ma	0.02	0.02	0.02	0.02	0.19 -	0.03 -	0.02
m1	0.01	0.01	0.01	0.01	0.06 -	0.02	0.01
m2	0.01	0.01	0.01	0.01	0.06 -	0.02	0.01
m3	0.01	0.01	0.01	0.01	0.06 -	0.02	0.01
mu	0.49	0.33 +	0.34 +	0.33 +	30.38 -	1.08 -	0.33 +
pa	0.02	0.02	0.02	0.02	0.04 -	0.04 -	0.02
pi	0.03	0.01 +	0.02 +	0.01 +	0.17 -	0.04 -	0.01 +
se	4.04	4.09	3.77	4.20	7.41 -	3.96	4.28
so	248.83	214.28	238.50	242.58	266.20	249.31	248.23
sh	0.01	0.01	0.01	0.01	0.04 -	0.02	0.01
vo	0.01	0.01	0.01	0.01	0.07 -	0.03 -	0.01

Na slikama 7.53, 7.54. i 7.55. prikazana je apsolutna razlika u vrednostima standardne devijacije za tačnost klasifikacije RBF algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, to je i veće odstupanje između standardnih devijacija. Najmanje odstupanje u standardnoj devijaciji pokazuje RF metoda, tako što za pojedine skupove podataka standardna devijacija ima manju, ali u nekim slučajevima i veću vrednost.

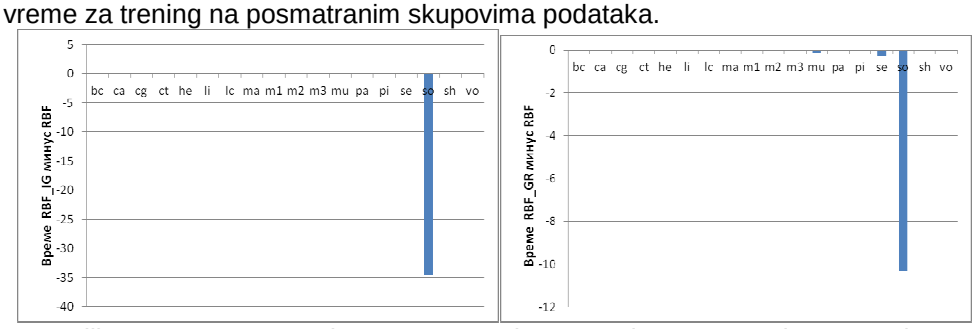
Potrebno vreme za trening RBF algoritma koji koristi originalni i redukovani skup podataka uz pomoć filter metoda prikazano je u tabeli 7.20. Potrebno vreme za trening podataka RBF klasifikatora za sve originalne skupove podataka iznosi ispod 1.00 sekunde, osim za dva seta podatka *se* i *so*, kod kojih je značajno veće. Potrebno vreme za trening je kod nekih metoda filtriranja veće, a kod nekih je manje u odnosu na originalni skup podataka. Kod svih skupova podataka, bar jedna od metoda filtriranja daje iste ili bolje rezultate za vreme potrebno za treniranje u odnosu na originalni skup.

Slike 7.56, 7.57. i 7.58. prikazuju apsolutnu razliku u potrebnom vremenu za trening RBF algoritma na osnovnom skupu podataka i RBF algoritma sa različitim metodama filtriranja. Primenjeni metod filtriranja IG i GR nije ni u jednom skupu podataka pokazao lošije rezultate za potrebno vreme za trening; kod 4, odnosno 3 skupa podataka respektivno, rezultati su bili i statistički bolji.

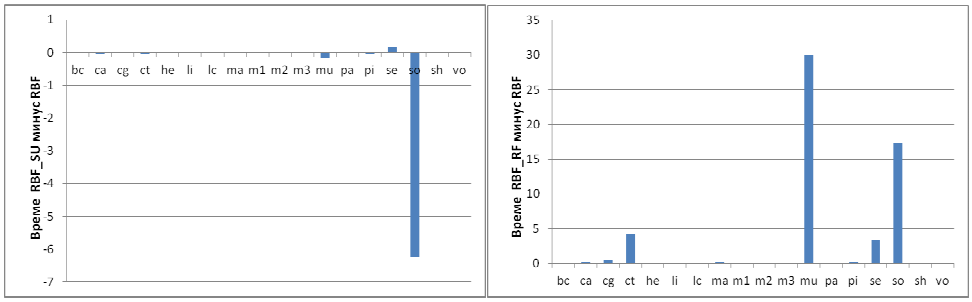
Primenjeni metod filtriranja SU je samo u jednom skupu podataka pokazao lošije rezultate za potrebno vreme za trening od RBF algoritma na osnovnom skupu podataka, a u 3 skupa podataka rezultati su bili i statistički bolji. Metod filtriranja RF je u svim skupovima podataka pokazao iste ili lošije rezultate za potrebno vreme za trening od RBF algoritma na osnovnom skupu podataka, a u skoro svim skupovima podataka rezultati su bili i statistički lošiji.

Metod filtriranja OR je u skoro svim skupovima podataka pokazao lošije rezultate od RBF algoritma na osnovnom skupu podataka, a ovi rezultati su u većini slučaja bili i statistički lošiji. Primenjeni metod filtriranja CS je u samo 1 skupu podataka pokazao lošije rezultate od RBF algoritma na osnovnom skupu podataka, a u 3 skupa podataka rezultati su bili i statistički bolji.

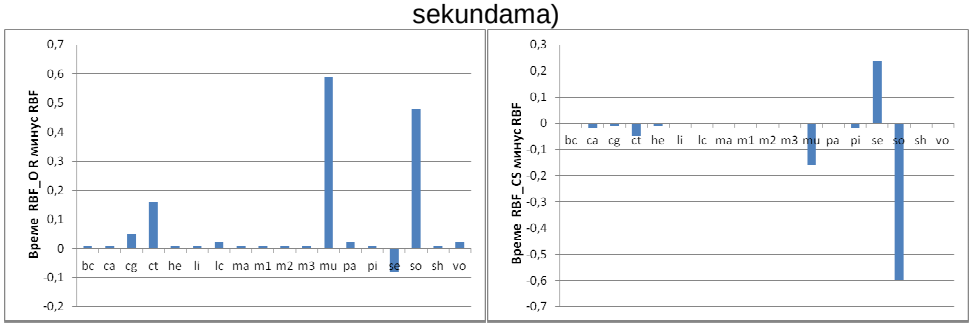
Korišćenjem RBF klasifikatora, možemo da zaključimo da je IG metoda filtriranja u najvećem broju slučaja dovela do statistički boljih rezultata za potrebno vreme za trening na posmatranim skupovima podataka.



Slika 7.56: Vreme treninga RBF_IG minus RBF i RBF_GR minus RBF (u sekundama)



Slika 7.57: Vreme treninga RBF_SU minus RBF i RBF_RF minus RBF (u sekundama)

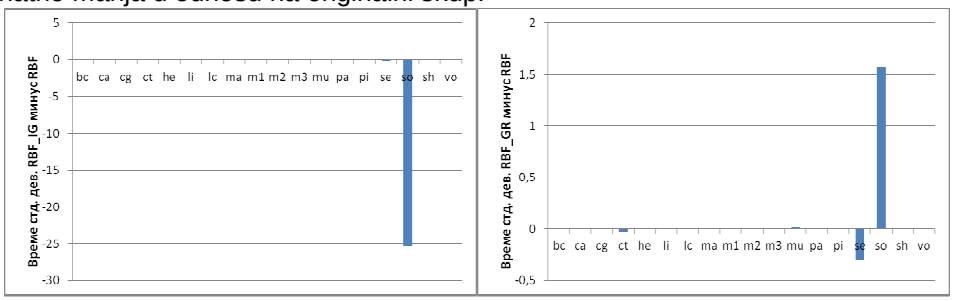


Slika 7.58: Vreme treninga RBF_OR minus RBF i RBF_CS minus RBF (u sekundama)

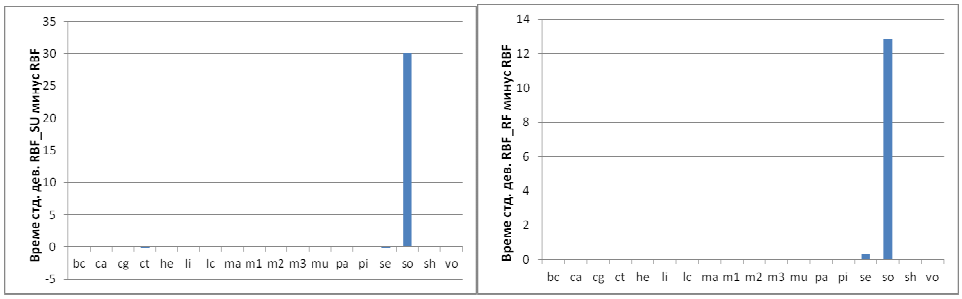
Tabela 7.21. Standardna devijacija za vreme treninga (u sekundama) RBF algoritma za originalni i redukovani skup podataka uz pomoć filter metoda

Skup	RBF	RBF_IG	RBF_GR	RBF_SU	RBF_RF	RBF_OR	RBF_CS
bc	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ca	0.01	0.01	0.01	0.01	0.01	0.01	0.01
cg	0.01	0.01	0.01	0.01	0.02	0.01	0.01
ct	0.09	0.11	0.06	0.06	0.10	0.07	0.05
he	0.01	0.01	0.01	0.01	0.01	0.01	0.01
li	0.01	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.01	0.01	0.01	0.01	0.01	0.01	0.00
ma	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m2	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m3	0.01	0.01	0.01	0.01	0.01	0.01	0.01
mu	0.06	0.06	0.08	0.08	0.13	0.04	0.04
pa	0.01	0.01	0.01	0.01	0.01	0.01	0.01
pi	0.01	0.01	0.01	0.01	0.01	0.01	0.01
se	1.30	1.19	1.00	1.27	1.61	1.21	1.30
so	129.82	104.56	131.39	159.87	142.67	122.82	120.93
sh	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.01	0.01	0.01	0.01	0.01	0.01	0.01

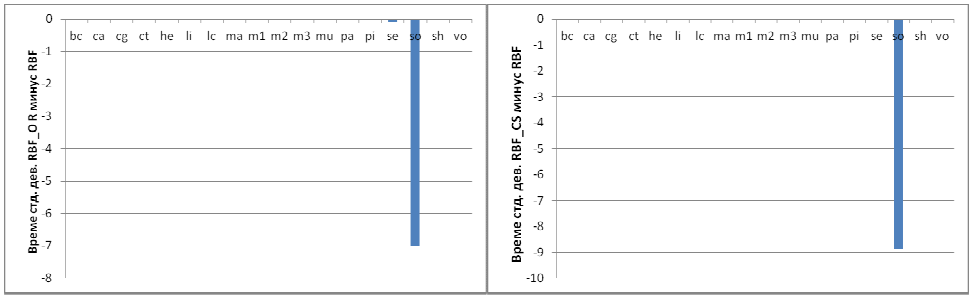
Tabela 7.21. prikazuje standardnu devijaciju za vreme treninga RBF algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa, osim za skup podataka *so* gde je uz pomoć nekih metoda ova vrednost znatno veća ili znatno manja u odnosu na originalni skup.



Slika 7.59: Standardna devijacija za vreme RBF_IG minus RBF i RBF_GR minus RBF



Slika 7.60: Standardna devijacija za vreme RBF_SU minus RBF i RBF_RF minus RBF



Slika 7.61: Standardna devijacija za vreme RBF_OR minus RBF i RBF_CS minus RBF

Na slikama 7.59, 7.60. i 7.61. prikazana je apsolutna razlika u vrednostima standardne devijacije za vreme treninga RBF algoritma za originalni i redukovani skup podataka uz pomoć filter metoda. Najveće odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka pokazuje metoda SU za *so* skup podataka.

8. ESTIMACIJA TAČNOSTI KLASIFIKACIJE ZA METODE PRETHODNOG UČENJA

U osmom delu, nakon razmatranja postavki eksperimentalnog istaživanja, biće prikazani rezultati istraživanja za različite metode prethodnog učenja, i to za svaki klasifikacioni algoritam posebno.

Kod metoda prethodnog učenja koriste se određeni algoritmi za modeliranje kako bi se ocenili podskupovi atributa u odnosu na njihovu klasifikacijsku ili prediktivnu moć. Kod ovih metoda vrednost određenog skupa atributa izražava se pomoću stepena ispravnosti klasifikacije koju postiže model konstruisan uz korišćenje tih atributa. Za klasifikaciju, za sve skupove podataka, korišćena je 10-struka unakrsna validacija, koja je pri tome bila uvek ponovljena 10 puta. Upoređivana je tačnost klasifikacije IBk, *Naïve Bayes*, SVM, J48 i RBF mreže na originalnom skupu podataka kao i na redukovanom skupu podataka dobijenom sa metodom prethodnog učenja.

Kod ovih metoda za svaki posmatrani podskup atributa izgrađuje se model i ocenjuju se njegove performanse, tako da bolje performanse nekog modela ukazuju na bolji izbor atributa iz kojih je model nastao. Postupak izbora atributa je računski vrlo zahtevan zbog učestalog izvođenja algoritma mašinskog učenja. Potrebno je dobiti ocenu performansi odgovarajućeg modela za svaki posmatrani podskup atributa, a metode ocene ispravnosti modela uglavnom zahtevaju usrednjavanje rezultata po većem broju izgrađenih modela. Kod ovih metoda za svaki posmatrani podskup atributa izgrađuje se više modela, a ukupan broj podskupova eksponencijalno raste s povećanjem broja atributa.

U ovom eksperimentalnom istraživanju metoda prethodnog učenja, kao metoda redukcije dimenzionalnosti podataka je koristila: različite klasifikatore za selekciju atributa, 5-struku unakrsnu validaciju i prag za ponavljanje unakrsne validacije ako standardna devijacija pređe ovu vrednost koji je podešen na 0.01.

Iscrpno pretraživanje podskupova atributa se može sprovesti samo za mali broj atributa, budući da je taj problem *NP*-težak. Zato se koriste razne tehnike pretraživanja, kao što su: najbolji prvi (eng. *best-first*), granaj-pa-ograniči (eng. *branch-and-bound*), simulirano kaljenje (eng. *simulated annealing*) i genetski algoritmi.

Kod metode prethodnog učenja, za pretraživanje prostora rešenja koristili smo heuristiku, kako bi ubrzali pretraživanje. Heuristika predstavlja iskustvena pravila o prirodi problema i osobinama cilja čija je svrha da se pretraživanje brže usmeri ka cilju. Heuristički ili usmereni postupak pretraživanja je onaj postupak pretraživanja koji koristi heuristiku kako bi suzio prostor pretraživanja. U ovom radu korišćen je heuristički postupak „pohlepnog najboljeg prvog“ (eng. *greedy best-first*), koji pretražuje podskup atributa koristeći algoritam uspona na vrh (eng. *hill climbing*). Postavljanje broja uzastopnih čvorova sa dozvoljenim ne-poboljšanjima kontroliše nivo praćenja unazad. Najbolji prvi može započeti sa praznim skupom

atributa i pretraživati prema unapred, odnosno početi sa punim skupom atributa i pretraživati unazad, ili početi u bilo kojoj tački i pretraživati u oba smera (razmatranja svih mogućih pojedinačnih atributa za dodavanje ili brisanje u određenoj tački). U radu smo koristili smer pretraživanja unapred, što znači da smo započeli sa praznim skupom, a kao kriterijum za kraj pretraživanja postavili smo 5 uzastopnih čvorova sa dozvoljenim ne-poboljšanjima. Glavni razlog za izbor smera pretraživanja unapred je računski, jer je izgradnja klasifikatora sa nekoliko atributa mnogo brža nego kada ima više atributa. Iako u teoriji, pretraživanje unazad od punog skupa atributa, može lakše uhvatiti interakciju atributa, metoda je izuzetno računski skupa.

Tabela 8.1. Broj atributa u originalnom skupu podataka i broj atributa selektovan uz pomoć metode prethodnog učenja za različite klasifikatore

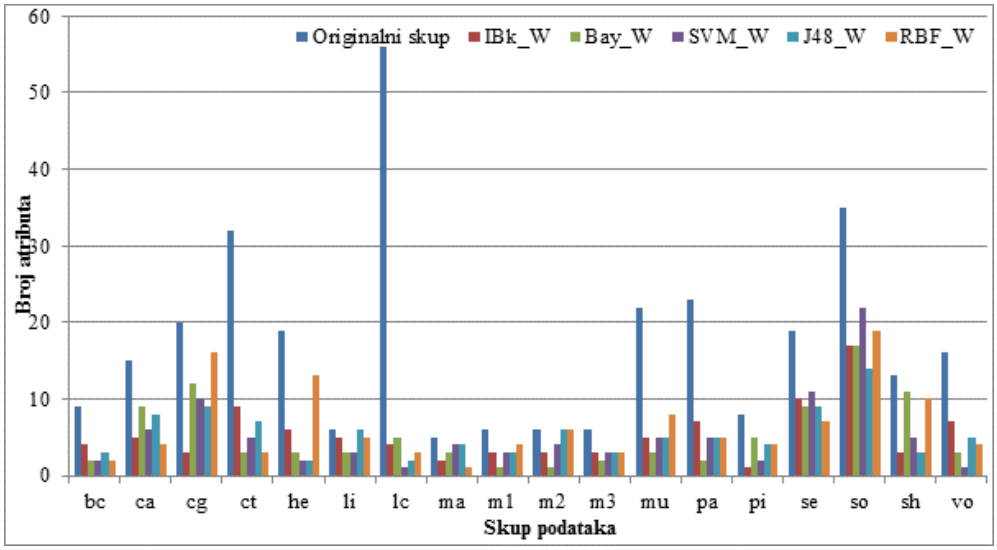
Skup	Orig. skup	IBk	Bay	SVM	J48	RBF
bc	9	4	2	2	3	2
ca	15	5	9	6	8	4
cg	20	3	12	10	9	16
ct	23	9	3	5	7	3
he	19	6	3	2	2	13
li	6	5	3	3	6	5
lc	56	4	5	1	2	3
ma	5	2	3	4	4	1
m1	6	3	1	3	3	4
m2	6	3	1	4	6	6
m3	6	3	2	3	3	3
mu	22	5	3	5	5	8
pa	23	7	2	5	5	5
pi	8	1	5	2	4	4
se	19	10	9	11	9	7
so	35	17	17	22	14	19
sh	13	3	11	5	3	10
vo	16	7	3	1	5	4

U eksperimentalnom istraživanju, kao i kod metoda filtriranja koristili smo uporedni *t*-test, gde je nivo značajnosti postavljen na vrednost 0.05.

S obzirom da su postojali setovi podataka sa nedostajućim vrednostima, da bi mogli da koristimo SVM algoritam, bilo je neophodno zameniti nedostajuće vrednosti sa procenjenim vrednostima za dati skup, jer sam algoritam SVM nije mogao da se izbori sa nedostajućim vrednostima za pojedine atribute u nekim od istanci.

U tabeli 8.1. prikazan je optimalan broj atributa za potrebe klasifikacije, nakon pretraživanja skupa mogućih rešenja za svaki od klasifikatora. Tabela prikazuje i originalnu veličinu skupa, kako bi se uporedili efekti redukcije dimenzionalnosti podataka. Od 18 posmatranih setova podataka, u 15 setova

podataka (svi osim *li*, *ma* i *m2*), tačno pola ili više od pola klasifikatora je smanjilo originalni broj atributa na pola.



Slika 8.1: Broj atributa u originalnom skupu i optimalan broj atributa dobijen metodama prethodnog učenja

Slika 8.1. prikazuje broj atributa u originalnom skupu podataka i optimalan broj atributa dobijen metodama prethodnog učenja. Najveću dobrobit od redukcije dimenzionalnosti podataka ima skup podataka *lc,* gde od 56 atributa, metodom prethodnog učenja smo izdvojili mali broj atributa relevantnih za posmatrani problem klasifikacije, čak isto ili manje od pet, za svaki od klasifikatora.

Koristeći metode prethodnog učenja za čak 7 skupova podataka, svi klasifikatori smanjuju broj atributa na isto ili više od pola. Ti skupovi podatak su: *bc, ct, lc, m3, mu, pa* i *vo.* Možemo uočiti da su ove metode dovele do značajne redukcije dimenzionalnosti podataka. Ako uporedimo podatke prikazane na slikama 7.1. i 8.1. možemo uočiti da je redukcija dimnzionalnosti podataka u značajno većoj meri urađena kod metoda prethodnog učenja. Za razliku od metoda filtriranja, ne postoje setovi podataka gde su svi klasifikatori izabrali isti broj značajnih atributa za dati skup podataka.

U nastavku eksperimentalnog istraživanja, za izabrani optimalan broj atributa, za svaki skup podataka i klasifikator, proveravana je tačnost klasifikacije korišćenjem različitih algoritama, i to: IBk, *Naïve Bayes*, SVM, J48 i RBF mreže. U nastavku teksta prikazani su dobijeni rezultati. Prikazane su različite skale na slikama za apsolutnu tačnost klasifikacije, standardnu devijaciju za tačnost, vreme treninga i standardnu devijaciju za vreme, kako bi se bolje uočile razlike koje postoje među rezultatima.

U tabelama koje slede za tačnost klasifikacije različitih klasifikatora i u tabelama za vreme potrebno za trening podataka su prikazane oznake „+“ i „-“,

koje označavaju da je određeni rezultat statistički bolji (+) ili lošiji (-) od osnovnog klasifikatora na nivou značajnosti koji je specificiran na vrednost od 0,05.

Tabela 8.2. Tačnost klasifikacije različitih klasifikatora za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja

Skup	IBk	IBk_W	Bay	Bay_W	SVM	SVM_W	J48	J48_W	RBF	RBF_W
bc	72.85	69.81	72.70	72.26	72.18	73.47	74.28	72.95	71.41	74.01
ca	81.57	85.22 +	77.86	85.67 +	55.88	85.86 +	85.57	84.43	79.55	85.91 +
cg	71.88	71.70	75.16	74.00	70.00	72.62 +	71.25	71.72	73.58	73.93
ct	98.85	98.42	87.30	98.49 +	81.01	98.38 +	98.57	98.88	97.93	98.67 +
he	81.40	81.85	83.81	82.21	79.38	83.90	79.22	81.90	85.29	82.12
li	62.22	59.66	54.89	59.46	59.37	60.62	65.84	66.36	65.06	62.86
lc	68.75	70.67	78.42	79.33	72.67	77.42	79.25	78.83	76.00	76.08
ma	75.60	83.02 +	82.64	82.01	80.27	82.03	82.19	82.47	77.31	81.11 +
m1	99.87	100.00	74.64	74.64	91.37	97.83 +	97.80	100.00	75.36	88.16 +
m2	79.08	65.72 -	62.79	65.72 +	65.44	65.72	63.48	65.72	67.82	65.67
m3	97.46	98.92 +	96.39	96.39	96.39	98.92 +	98.92	98.92	96.54	97.49
mu	100.00	100.00	95.76	99.63 +	100.00	100.00	100.00	100.00	98.61	97.12
pa	95.91	93.40	69.98	82.04 +	79.36	97.64 +	84.74	86.24	81.22	87.47 +
pi	70.62	67.76	75.75	76.11	65.11	71.76 +	74.49	73.44	74.04	75.79
se	97.15	97.08	80.17	89.83 +	64.76	90.91 +	96.79	96.73	87.88	91.82
so	91.20	94.77 +	92.94	92.67	90.04	89.17	91.78	91.74	84.48	84.41
sh	76.15	78.56	83.59	84.30	55.93	81.74 +	78.15	81.74	83.11	82.59
vo	92.58	94.92 +	90.02	95.75 +	95.63	95.54	96.57	95.24 -	93.73	94.94

Tabela 8.2. prikazuje tačnost klasifikacije različitih klasifikatora za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja. Možemo uočiti da u svim setovima podataka imamo dobijene rezultate za bar jednu od metoda prethodnog učenja koji su statistički bolji od osnovnog klasifikatora. Samo u dva seta podataka *m2* i *vo*, imamo značajno lošije podatke za neku od metoda prethodnog učenja.

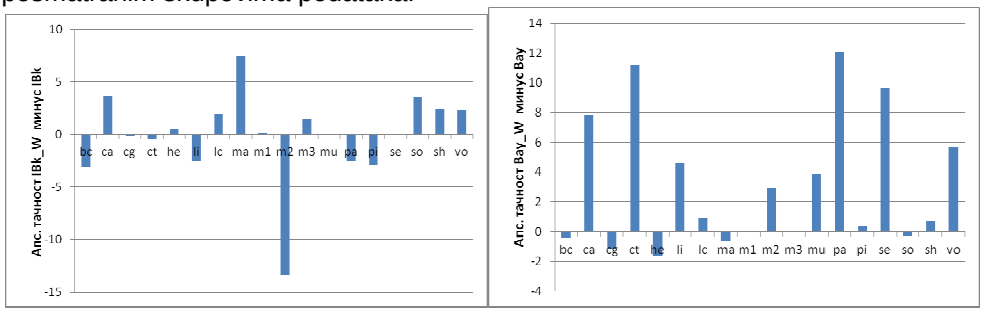
Na slikama 8.2, 8.3. i 8.4. prikazana je apsolutna razlika u tačnosti klasifikacije različitih algoritma na osnovnom skupu podataka i istih tih algoritma sa metodama prethodnog učenja. Metod prethodnog učenja sa IBk klasifikatorom je u više od pola skupova podataka (10 skupova) pokazao iste ili bolje rezultate od IBk algoritma na osnovnom skupu podataka, a u 5 skupova podataka rezultati su bili i statistički bolji. Metod prethodnog učenja sa *Naïve Bayes* klasifikatorom je u više od dve trećine skupova podataka (13 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, a u 7 skupova podataka rezultati su bili i statistički bolji.

Metod prethodnog učenja sa SVM klasifikatorom je u skoro svim skupovima podataka (16 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka. U 9 skupova podataka rezultati su bili i statistički bolji. Metod prethodnog učenja sa J48 klasifikatorom je u više od pola skupova

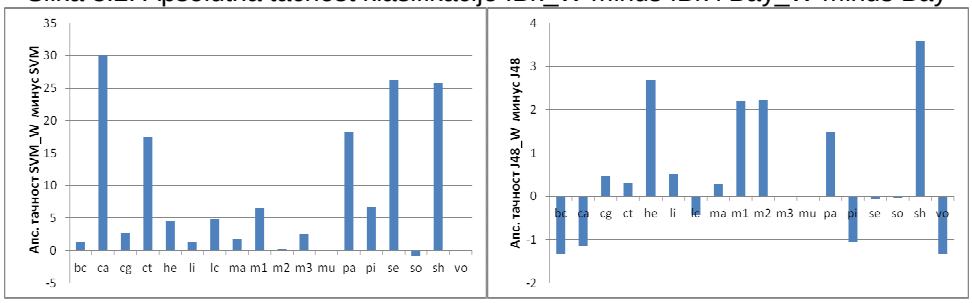
podataka (11 skupova) pokazao iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka, ali ne postoji rezultat koji bi bio i statistički bolji.

Metod prethodnog učenja sa RBF klasifikatorom je u dve trećine skupova podataka (12 skupova) pokazao iste ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, a u 5 skupova podataka rezultati su bili i statistički bolji.

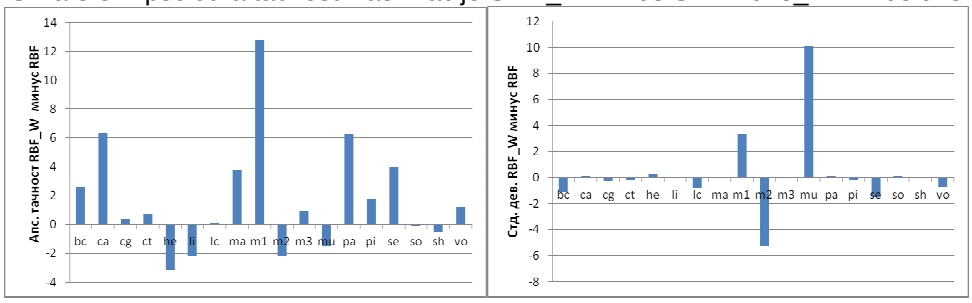
Korišćenjem metode prethodnog učenja možemo da zaključimo da je SVM klasifikator u najvećem broju slučaja doveo do statistički boljih rezultata na posmatranim skupovima podataka.



Slika 8.2: Apsolutna tačnost klasifikacije IBk_W minus IBk i Bay_W minus Bay



Slika 8.3: Apsolutna tačnost klasifikacije SVM_W minus SVM i J48_W minus J48



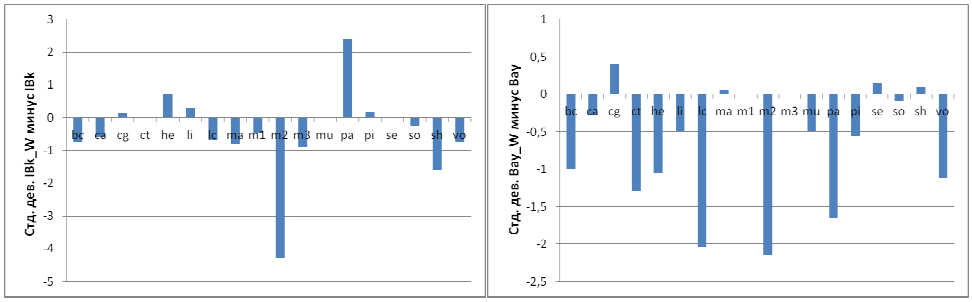
Slika 8.4: Apsolutna tačnost klasifikacije RBF_W minus RBF i standardna devijacija za tačnost RBF_W minus RBF

S obzirom na iznetu tvrdnju da je dobar onaj algoritam koji daje sličan rezultat u svim slučajevima, odnosno vrednost za standardnu devijaciju je minimalna, razmatraćemo vrednosti za standardnu devijaciju za tačnost klasifikacije. Tabela 8.3. prikazuje standardnu devijaciju za tačnost klasifikacije

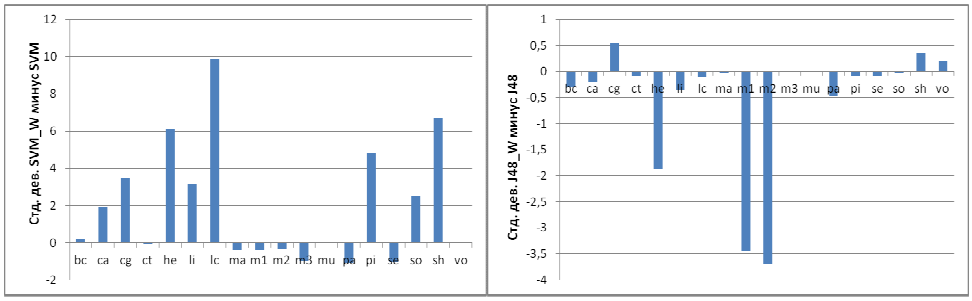
različitih algoritma za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa, osim u slučaju SVM algoritma, kod koga su vrednosti standardne devijacije za tačnost klasifikacije značajno veće sa metodom prethodnog učenja. Veća odstupanja u standardnoj devijaciji pokazuje metoda RBF mreže, u odnosu na druge algoritme, ali samo za pojedine skupove podataka. Generalno, vrednosti standardne devijacije za tačnost klasifikacije je kod ostalih algoritama manja sa metodom prethodnog učenja.

Tabela 8.3. Standardna devijacija za tačnost klasifikacije različitih klasifikatora za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja

Skup	IBk	IBk_W	Bay	Bay_W	SVM	SVM_W	J48	J48_W	RBF	RBF_W
bc	6.93	6.19	7.74	6.74	5.86	6.04	6.05	5.74	7.88	6.75
ca	4.57	3.98	4.18	3.90	2.14	4.05	3.96	3.75	4.07	4.20
cg	3.68	3.82	3.48	3.88	0.00	3.48	3.17	3.71	4.30	4.03
ct	0.77	0.77	2.21	0.92	0.99	0.89	0.89	0.80	1.02	0.82
he	8.55	9.29	9.70	8.65	2.26	8.38	9.57	7.69	8.29	8.53
li	8.18	8.47	8.83	8.34	2.28	5.42	7.40	7.05	8.80	8.81
lc	22.33	21.66	21.12	19.08	11.12	21.03	21.50	21.40	22.91	22.12
ma	3.90	3.10	3.11	3.17	3.41	3.01	3.21	3.19	3.31	3.35
m1	0.46	0.00	4.26	4.26	3.10	2.71	3.45	0.00	5.92	9.26
m2	5.06	0.79	2.94	0.79	1.14	0.79	4.48	0.79	6.24	0.97
m3	2.13	1.23	2.20	2.20	2.20	1.23	1.23	1.23	2.19	2.20
mu	0.00	0.00	0.73	0.23	0.00	0.00	0.00	0.00	0.58	10.65
pa	4.52	6.93	9.51	7.85	4.46	3.37	8.01	7.53	7.37	7.44
pi	4.67	4.85	5.32	4.76	0.34	5.18	5.27	5.18	4.91	4.72
se	1.11	1.11	2.12	2.27	2.66	1.62	1.29	1.20	2.57	1.04
so	3.00	2.74	2.92	2.83	0.39	2.88	3.19	3.16	0.86	0.94
sh	8.46	6.87	5.98	6.07	1.12	7.85	7.42	7.78	6.50	6.44
vo	3.63	2.87	3.91	2.79	2.76	2.77	2.56	2.76	3.87	3.19



Slika 8.5: Standardna devijacija za tačnost IBk_W minus IBk i Bay_W minus Bay



Slika 8.6: Standardna devijacija za tačnost SVM_W minus SVM i J48_W minus J48

Na slikama 8.4, 8.5. i 8.6. prikazana je apsolutna razlika u vrednostima standardne devijacije za tačnost klasifikacije različitih algoritma za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, to je i veće odstupanje između standardnih devijacija. Najmanje odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka, pokazuje algoritam *Naïve Bayes* i J48, dok najveće odstupanje ima algoritam SVM i RBF mreže.

Tabela 8.4. Potrebno vreme za trening (u sekundama) različitih klasifikatora za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja

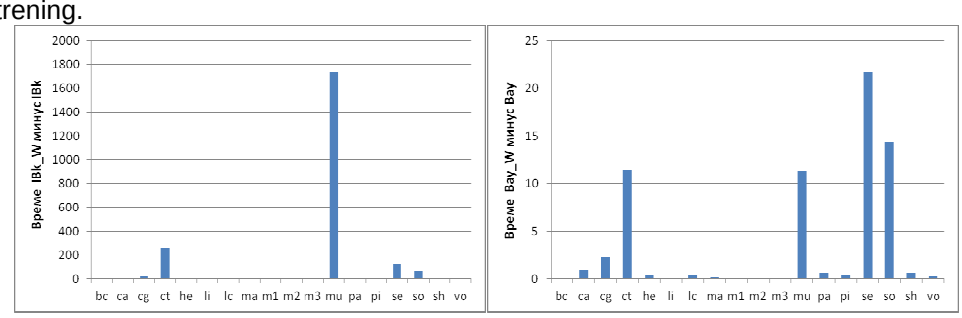
Skup	IBk	IBk_W	Bay	Bay_W	SVM	SVM_W	J48	J48_W	RBF	RBF_W
bc	0.00	0.96 -	0.00	0.12 -	0.01	6.28 -	0.00	0.31 -	0.01	3.69 -
ca	0.00	10.03 -	0.00	0.98 -	0.14	84.73 -	0.01	3.95 -	0.02	30.63 -
cg	0.00	23.08 -	0.00	2.30 -	0.50	312.81 -	0.01	12.20 -	0.05	67.23 -
ct	0.00	254.81 -	0.01	11.48 -	3.73	555.86 -	0.10	26.38 -	0.35	1189.54 -
he	0.00	1.30 -	0.00	0.41 -	0.01	18.07 -	0.00	0.57 -	0.01	11.10 -
li	0.00	0.59 -	0.00	0.15 -	0.02	6.95 -	0.00	0.47 -	0.01	2.59 -
lc	0.00	1.21 -	0.00	0.39 -	0.00	35.65 -	0.00	0.32 -	0.00	11.84 -
ma	0.00	3.52 -	0.00	0.18 -	0.13	16.72 -	0.00	0.68 -	0.02	4.33 -
m1	0.00	1.01 -	0.00	0.06 -	0.03	5.88 -	0.00	0.15 -	0.01	4.38 -
m2	0.00	0.92 -	0.00	0.06 -	0.04	6.30 -	0.00	0.07 -	0.01	1.81 -
m3	0.00	0.93 -	0.00	0.05 -	0.02	3.82 -	0.00	0.08 -	0.01	3.35 -
mu	0.00	1733.88-	0.00	11.33 -	3.73	1572.20 -	0.03	20.34 -	0.49	943.60 -
pa	0.00	3.09 -	0.00	0.67 -	0.02	62.12 -	0.01	3.87 -	0.02	19.36 -
pi	0.00	4.07 -	0.00	0.47 -	0.14	40.10 -	0.01	2.03 -	0.03	10.49 -
se	0.00	122.89 -	0.01	21.74 -	3.65	3012.16 -	0.08	90.07 -	2.61	11271.55
so	0.00	68.93 -	0.00	14.38 -	0.81	8806.81	0.01	29.48 -	179.02	1389781.94-
sh	0.00	1.82 -	0.00	0.72 -	0.02	19.20 -	0.00	1.66 -	0.01	11.02 -
vo	0.00	5.44 -	0.00	0.29 -	0.02	4.71 -	0.00	0.46 -	0.01	21.21 -

U tabeli 8.4. prikazano je potrebno vreme za trening u sekundama različitih klasifikatora koji koriste originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja. Potrebno vreme za trening podataka IBk klasifikatora za sve originalne skupove podataka iznosi 0.00 sekundi, dok za metode prethodnog učenja ono je značajno veće. Potrebno vreme za trening podataka *Naïve Bayes* klasifikatora za sve originalne skupove podataka iznosi manje od 0.01 sekundi, dok za metode prethodnog učenja ono je veće.

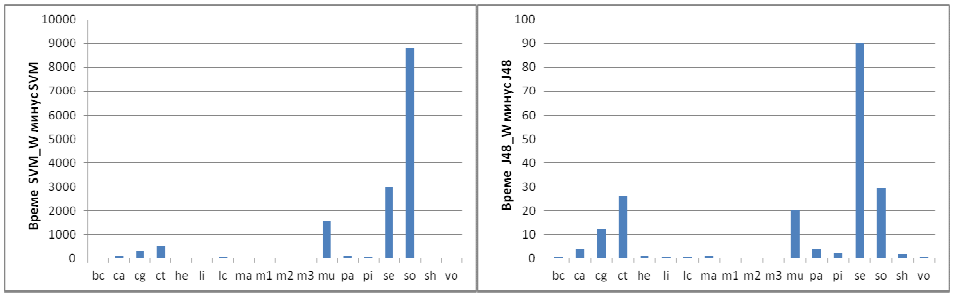
Za potrebno vreme treninga podataka SVM klasifikatora za sve originalne skupove podataka ono iznosi manje od 3.73 sekunde, dok za metode prethodnog učenja ono je značajno veće. Potrebno vreme za trening podataka J48 klasifikatora za sve originalne skupove podataka iznosi manje od 0.1 sekunde, dok za metode prethodnog učenja ono je veće. Potrebno vreme za trening podataka RBF klasifikatora za sve originalne skupove podataka iznosi manje od 179.02 sekunde, dok za metode prethodnog učenja ono je značajno veće.

Na slikama 8.7, 8.8. i 8.9. prikazana je apsolutna razlika u potrebnom vremenu za trening različitih klasifikatora na osnovnom skupu podataka i sa redukovanim brojem atributa uz pomoć metoda prethodnog učenja. Možemo da uočimo na osnovu slika i verikalne ose koja prikazuje vreme u sekundama, da klasifikator RBF sa metodom prethodnog učenja zahteva najviše vremena za učenje, u slučaju *so* skupa podataka i do 1389781.94 sekundi, a potom klasifikator SVM koji za isti skup podataka zahteva vreme od 8806.81 sekundi. Kod metode prethodnog učenja i kod velikih setova podataka, i to posebno kod *mu*, *se* i *so* pri korišćenju SVM i RBF algoritma, zadatak redukcije dimenzionalnosti podataka je bio zahtevan i u pogledu memorijskih resursa kojih je bilo potrebno obezbediti, kao i u pogledu zahtevanog vremena za izvršavanje algoritma. Ovo nije bio slučaj kod metoda filtriranja i ekstrakcije, ni za jednu od primenjenih metoda na bilo kom skupu podatka.

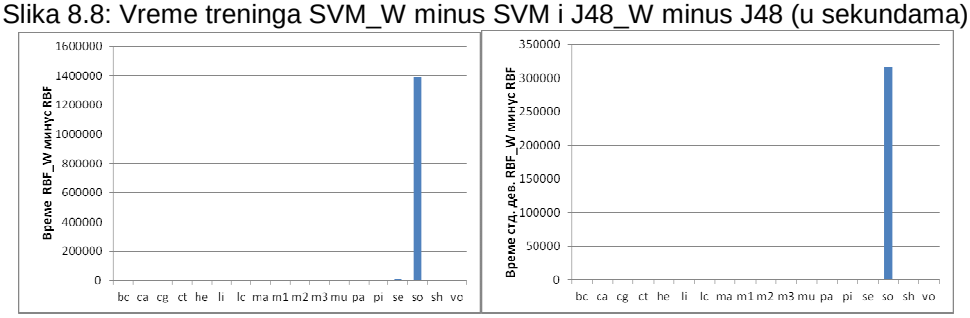
Možemo uočiti da su ove metode dovele do značajnog povećanja potrebnog vremena za trening podataka. Ako uporedimo podatke prikazane na slikama koji se odnose na potrebno vreme za trening korišćenjem filter metoda i metoda prethodnog učenja, možemo uočiti da je u značajno većoj meri potrebno vremena metodama prethodnog učenja. Takođe, dokaz iznetog tvrđenja nalazi se i u tabeli 8.4, gde za sve skupove podataka i za sve korišćene algoritme, rezultati za potrebno vreme treninga su statistički lošiji, odnosno zahtevano je više vremena za trening.



Slika 8.7: Vreme treninga IBk_W minus IBk i Bay_W minus Bay (u sekundama)



Slika 8.8: Vreme treninga SVM_W minus SVM i J48_W minus J48 (u sekundama)



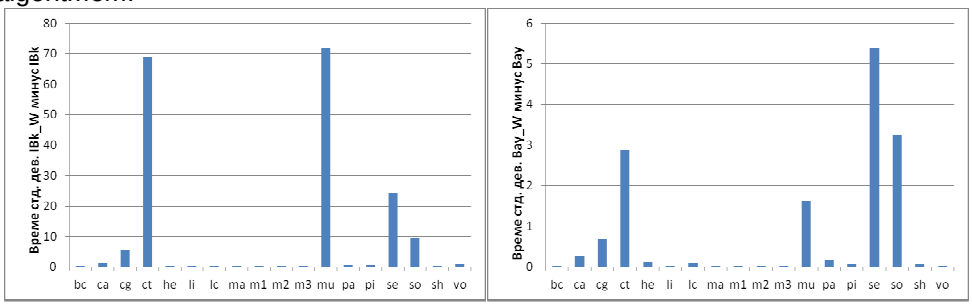
Slika 8.9: Vreme treninga RBF_W minus RBF (u sekundama) i standardna devijacija za vreme RBF_W minus RBF

Tabela 8.5. Standardna devijacija potrebnog vremena za trening (u sekundama) različitih klasifikatora za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja

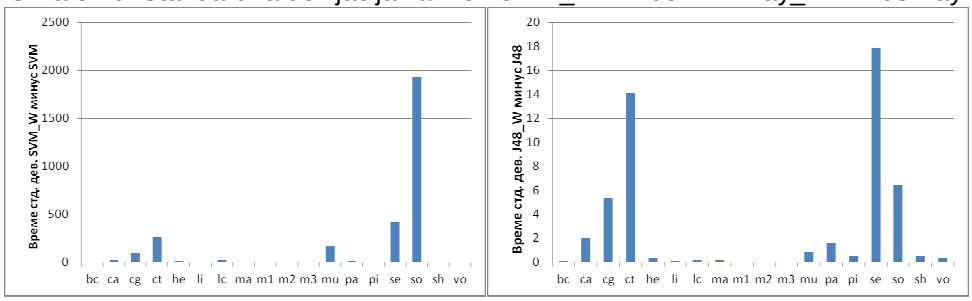
Skup	IBk	IBk_W	Bay	Bay_W	SVM	SVM _W	J48	J48_W	RBF	RBF_W
bc	0.00	0.17	0.00	0.02	0.00	1.21	0.01	0.09	0.01	0.64
ca	0.00	1.39	0.01	0.29	0.01	22.20	0.01	2.00	0.01	7.76
cg	0.00	5.76	0.00	0.68	0.06	101.38	0.00	5.38	0.01	12.99
ct	0.00	68.98	0.01	2.90	0.21	259.99	0.01	14.17	0.09	346.38
he	0.00	0.43	0.00	0.13	0.01	9.76	0.01	0.36	0.01	2.67
li	0.00	0.08	0.00	0.03	0.01	1.19	0.01	0.08	0.01	0.33
lc	0.00	0.32	0.00	0.10	0.01	20.60	0.00	0.14	0.01	3.05
ma	0.00	0.46	0.00	0.04	0.01	2.83	0.01	0.15	0.01	0.44
m1	0.00	0.08	0.00	0.01	0.01	0.61	0.00	0.01	0.01	0.60
m2	0.00	0.06	0.00	0.01	0.01	0.19	0.00	0.01	0.01	0.35
m3	0.00	0.07	0.00	0.01	0.01	0.35	0.00	0.01	0.01	0.29
mu	0.01	71.89	0.01	1.64	0.09	164.05	0.01	0.88	0.07	256.98
pa	0.00	0.77	0.00	0.18	0.01	15.40	0.01	1.58	0.02	4.52
pi	0.00	0.69	0.00	0.08	0.01	5.09	0.01	0.47	0.01	1.41
se	0.00	24.38	0.01	5.40	0.30	413.59	0.01	17.86	0.13	1370.49
so	0.00	9.60	0.00	3.25	0.09	1931.81	0.01	6.46	148.71	317542.22
sh	0.00	0.26	0.00	0.07	0.01	4.97	0.01	0.47	0.01	1.05
vo	0.00	1.09	0.00	0.04	0.01	0.68	0.00	0.34	0.01	4.79

Tabela 8.5. prikazuje standardnu devijaciju za vreme treninga različitih algoritama za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja. Iz tabele se može videti da se standardne devijacije razlikuju dosta između standardnog algoritma i algoritama koji koriste metode prethodne redukcije atributa. U slučaju svih algoritama, kod svih skupova podataka, vrednosti za standardnu devijaciju za vreme treninga su veće kod klasifikatora koji koriste metode prethodne redukcije atributa, u odnosu na standardni algoritam.

Na slikama 8.9, 8.10. i 8.11. prikazana je apsolutna razlika u vrednostima standardne devijacije za vreme treninga algoritama za originalni i redukovani skup podataka uz pomoć metoda prethodnog učenja. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, u pozitivnom i negativnom smeru, to je i veće odstupanje između standardnih devijacija. Najveće odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka pokazuju metode prethodnog učenja sa RBF i SVM algoritmom.



Slika 8.10: Standardna devijacija za vreme IBk_W minus IBk i Bay_W minus Bay



Slika 8.11: Standardna devijacija za vreme SVM_W minus SVM i J48_W minus J48

9. ESTIMACIJA TAČNOSTI KLASIFIKACIJE ZA EKSTRAKCIJU ATRIBUTA

U devetom delu, nakon razmatranja postavki eksperimentalnog istaživanja, biće prikazani rezultati istraživanja za ekstrakciju atributa uz pomoć PCA metode, i to za svaki klasifikacioni algoritam posebno.

Problem dimenzionalnosti se može prevladati tako da se odabere samo podskup relevantnih atributa ili stvaranjem novih atributa koje sadrže najviše informacija o klasi. Prva metodologija se zove selekcija atributa, a druga se zove ekstrakcija atributa, a to uključuje linearnu (PCA, ICA i sl.) i ne-linearnu metodu ekstrakcije atributa. U eksperimentalnom istraživanju koristili smo PCA metodu, kao metodu ekstrakcije atributa.

PCA predstavlja tehniku formiranja novih, sintetskih varijabli koje su linearne složenice - kombinacije izvornih varijabli. Ovom tehnikom se redukuje dimenzionalnost, a maksimalni broj novih varijabli koji se može formirati jednak je broju izvornih, pri čemu nove varijable nisu međusobno korelisane. Očekuje se da će većina novih varijabli činiti šum, i imati tako malu varijansu da se ona može zanemariti. Većinu informacija će poneti prvih nekoliko varijabli - glavnih komponenti, čije su varijanse značajne veličine. Na taj način, iz velikog broja izvornih varijabli kreirano je tek nekoliko glavnih komponenti koje nose većinu informacija i čine glavni oblik. Naravno, ima situacija kada to nije tako, i to u slučaju kada su izvorne varijable nekorelisane, tada analiza ne daje povoljne rezultate.

U analizi glavnih komponenata osnovni koraci su: standardizacija varijabli, izračunavanje matrice korelacije, pronalaženje svojstvenih vrednosti glavnih komponenti i odbacivanje komponenti. Najpre, potrebno je standardizovati varijable tako da im je prosek 0, a varijansa 1 kako bi sve bile na jednakom nivou u analizi, jer je većina setova podataka konstruisana iz varijabli različitih skala i jedinica merenja. Potom, potrebno je izračunati matrice korelacija između svih izvornih standardizovanih varijabli, a nakon toga, pronaći svojstvene vrednosti glavnih komponenata. Na kraju, potrebno je odbaciti one komponente koje imaju proporcionalno mali udeo varijanse (obično prvih nekoliko komponenti ima 80% - 90% ukupne varijanse). U eksperimentalnom istraživanju koristili smo za prag odbacivanja vrednost od 95%.

U tabelama koje slede za tačnost klasifikacije različitih klasifikatora i u tabelama za vreme potrebno za trening podataka su prikazane oznake „+“ i „-“, koje označavaju da je određeni rezultat statistički bolji (+) ili lošiji (-) od osnovnog klasifikatora na nivou značajnosti koji je specificiran na vrednost od 0,05. Takođe, prikazane su različite skale na slikama za apsolutnu tačnost klasifikacije, standardnu devijaciju za tačnost, vreme treninga i standardnu devijaciju za vreme, kako bi se bolje uočile razlike koje postoje među rezultatima.

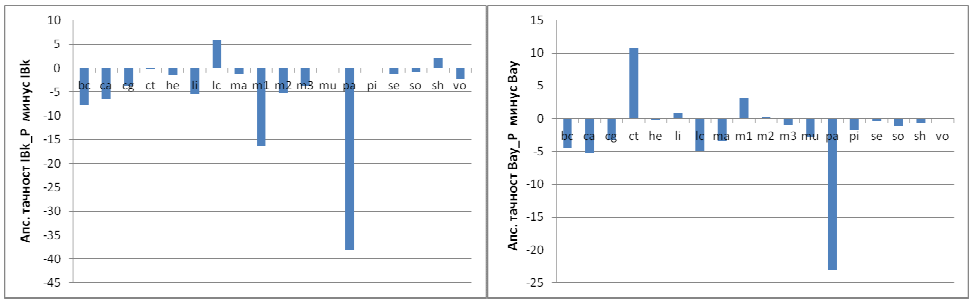
Tabela 9.1. Tačnost klasifikacije različitih klasifikatora za originalni i redukovani skup podataka uz pomoć PCA

Skup	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	72.85	65.08 -	72.70	68.17	72.18	71.50	74.28	67.94 -	71.41	68.58
ca	81.57	75.12 -	77.86	72.67 -	55.88	85.49 +	85.57	79.91 -	79.55	77.09
cg	71.88	68.08 -	75.16	71.85 -	70.00	75.53 +	71.25	68.25	73.58	71.56
ct	98.85	98.59	87.30	98.11 +	81.01	98.97 +	98.57	98.14	97.93	98.28
he	81.40	79.79	83.81	83.68	79.38	85.90 +	79.22	80.31	85.29	83.71
li	62.22	56.81 -	54.89	55.85	59.37	62.29	65.84	56.36 -	65.06	61.46
lc	68.75	74.58	78.42	73.50	72.67	71.67	79.25	63.08	76.00	72.75
ma	75.60	74.27	82.64	79.18 -	80.28	81.97	82.19	80.46	77.31	79.08
m1	99.87	83.38 -	74.64	77.81	91.37	100.00 +	97.80	96.71	75.36	83.52 +
m2	79.08	73.89	62.79	63.06	65.44	61.20 -	63.48	76.34 +	67.82	67.73
m3	97.46	93.81	96.39	95.33	96.39	98.92 +	98.92	96.54 -	96.54	95.96
mu	100.00	100.00	95.76	92.88 -	100.00	99.94 -	100.00	99.75 -	98.61	98.31
pa	95.91	57.86 -	69.98	46.90 -	79.36	82.89	84.74	82.41	81.22	58.38 -
pi	70.62	70.54	75.75	74.08	65.11	76.38 +	74.49	70.92	74.04	73.63
se	97.15	95.93 -	80.17	79.82	63.98	91.68 +	96.79	91.43 -	87.31	87.04
so	91.20	90.48	92.94	91.83	93.63	92.94	91.78	86.76 -	93.63	92.97
sh	76.15	78.19	83.59	82.85	55.93	83.37 +	78.15	76.41	83.11	81.93
vo	92.58	90.31	90.02	90.09	95.63	94.25	96.57	90.27 -	93.73	92.50

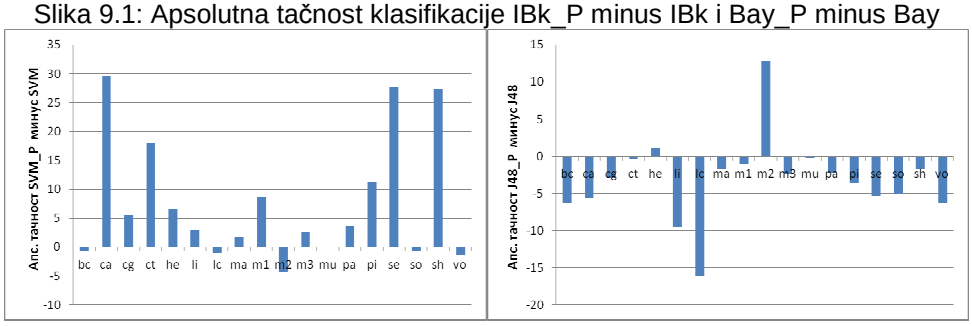
Možemo uočiti da u deset setova podataka (*ca, cg, ct, he, m1, m2, m3, pi, se* i *sh*) imamo dobijene rezultate za tačnost klasifikacije za redukovani skup podataka uz pomoć PCA metode za bar jedan od klasifikatora koji su statistički bolji od osnovnog klasifikatora (tabela 9.1). Ni u jednom setu podataka, nemamo značajno lošije podatke za sve klasifikatore, što znači da uvek možemo izabrati klasifikator za dati skup podataka koja ima statistički bolje rezultate ili rezultate koji su približni originalnom skupu podataka.

Na slikama 9.1, 9.2. i 9.3. prikazana je apsolutna razlika u tačnosti klasifikacije različitih algoritma na osnovnom skupu podataka i redukovanom skupu podataka korišćenjem PCA. Kod IBk algoritma uz korišćenje PCA je u samo 3 skupa podataka pokazao iste ili bolje rezultate za tačnost klasifikacije od IBk algoritma na osnovnom skupu podataka, ali ni u jednom skupu podataka rezultati nisu bili i statistički bolji. PCA je kod *Naïve Bayes* algoritma u manje od trećine skupova podataka (5 skupova) pokazao iste ili bolje rezultate od *Naïve Bayes* algoritma na osnovnom skupu podataka, a samo u 1 skupu podataka rezultati su bili i statistički bolji.

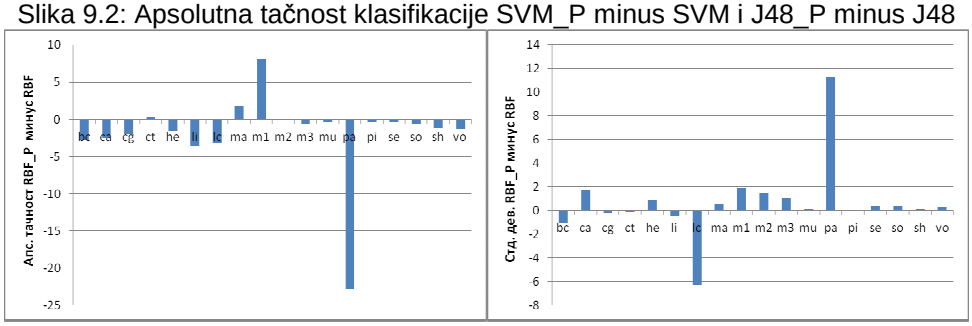
PCA je kod SVM algoritma u dve trećine skupova podataka (12 skupova) pokazao iste ili bolje rezultate od SVM algoritma na osnovnom skupu podataka. U 9 skupova podataka rezultati za tačnost klasifikacije su bili i statistički bolji. Kod primene PCA na J48 algoritam, u samo 2 skupa podataka pokazao je iste ili bolje rezultate od J48 algoritma na osnovnom skupu podataka. Samo u 1 skupu podataka rezultati su bili i statistički bolji.



Slika 9.1: Apsolutna tačnost klasifikacije IBk_P minus IBk i Bay_P minus Bay



Slika 9.2: Apsolutna tačnost klasifikacije SVM_P minus SVM i J48_P minus J48



Slika 9.3: Apsolutna tačnost klasifikacije RBF_P minus RBF i standardna devijacija za tačnost RBF_P minus RBF

PCA kod RBF algoritma je u samo 3 skupa podataka pokazao iste ili bolje rezultate od RBF algoritma na osnovnom skupu podataka, a u 1 skupu podataka, rezultati su bili i statistički bolji.

Korišćenjem PCA za redukciju dimenzionalnosti podataka, možemo da zaključimo da je SVM algoritam u najvećem broju slučaja doveo do statistički boljih rezultata na posmatranim skupovima podataka.

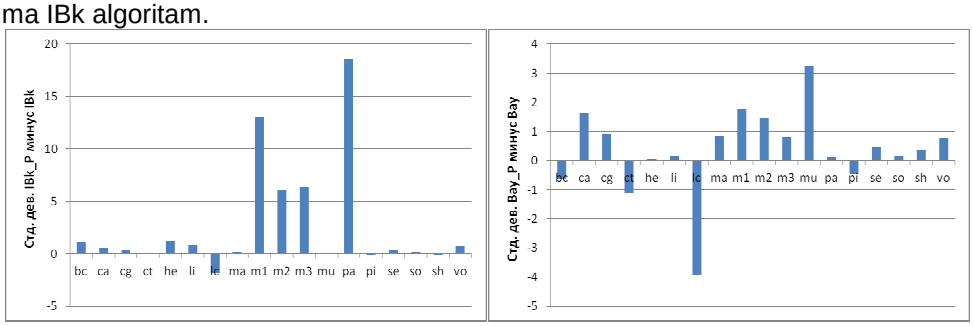
Tabela 9.2. prikazuje standardnu devijaciju za tačnost klasifikacije različitih algoritma za originalni i redukovani skup podataka uz pomoć PCA metode. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste predselekciju atributa, osim u slučaju IBk i RBF algoritama. IBk i RBF algoritmi sa metodama prethodnog

učenja imaju generalno veće vrednosti za standardnu devijaciju za tačnost klasifikacije od standardnog algoritma IBk.

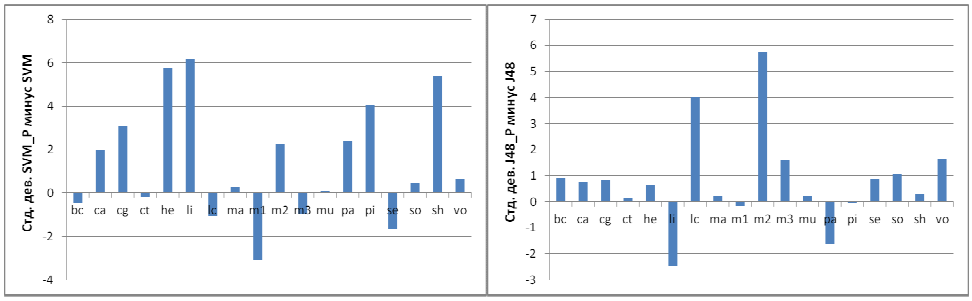
Tabela 9.2. Standardna devijacija za tačnost klasifikacije različitih klasifikatora za originalni i redukovani skup podataka uz pomoć PCA

Skup	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	6.93	8.07	7.74	7.11	5.86	5.38	6.05	6.95	7.88	6.81
ca	4.57	5.15	4.18	5.83	2.14	4.13	3.96	4.73	4.07	5.83
cg	3.68	4.09	3.48	4.40	0.00	3.08	3.17	3.98	4.30	4.06
ct	0.77	0.85	2.21	1.09	0.99	0.80	0.89	1.06	1.02	0.91
he	8.55	9.81	9.70	9.75	2.26	8.01	9.57	10.20	8.29	9.12
li	8.18	8.99	8.83	9.00	2.28	8.47	7.40	4.93	8.80	8.31
lc	22.33	20.46	21.12	17.18	11.12	10.05	21.50	25.52	22.91	16.62
ma	3.90	4.04	3.11	3.95	3.42	3.67	3.21	3.41	3.31	3.86
m1	0.46	13.50	4.26	6.01	3.10	0.00	3.45	3.30	5.92	7.86
m2	5.06	11.11	2.94	4.40	1.14	3.40	4.48	10.24	6.24	7.70
m3	2.13	8.48	2.20	3.02	2.20	1.23	1.23	2.85	2.19	3.17
mu	0.00	0.02	0.73	3.98	0.00	0.08	0.00	0.20	0.58	0.70
pa	4.52	23.11	9.51	9.61	4.46	6.87	8.01	6.37	7.37	18.59
pi	4.67	4.58	5.32	4.86	0.34	4.37	5.27	5.20	4.91	4.88
se	1.11	1.44	2.12	2.58	3.47	1.80	1.29	2.15	2.15	2.50
so	3.00	3.12	2.92	3.08	2.22	2.67	3.19	4.26	2.22	2.64
sh	8.46	8.37	5.98	6.35	1.12	6.50	7.42	7.73	6.50	6.58
vo	3.63	4.35	3.91	4.69	2.76	3.40	2.56	4.20	3.87	4.11

Na slikama 9.3, 9.4. i 9.5. prikazana je apsolutna razlika u vrednostima standardne devijacije za tačnost klasifikacije različitih algoritma za originalni i redukovani skup podataka uz pomoć PCA metode. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, u pozitivnom i negativnom smeru, to je i veće odstupanje između standardnih devijacija. Najmanje odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka, pokazuje *Naïve Bayes* algoritam, dok najveće odstupanje ima IBk algoritam.



Slika 9.4: Standardna devijacija za tačnost IBk_P minus IBk i Bay_P minus Bay



Slika 9.5: Standardna devijacija za tačnost SVM_P minus SVM i J48_P minus J48

U tabeli 9.3. prikazano je potrebno vreme za trening u sekundama različitih algoritma za klasifikaciju koji koriste originalni i redukovani skup podataka uz pomoć PCA metode. Potrebno vreme za trening podataka IBk klasifikatora za sve originalne skupove podataka iznosi 0.00 sekunde, dok je za IBk sa PCA ono veće. Potrebno vreme za trening podataka *Naïve Bayes* klasifikatora za sve originalne skupove podataka iznosi manje od 0.01 sekunde, dok je za *Naïve Bayes* sa PCA ono takođe veće.

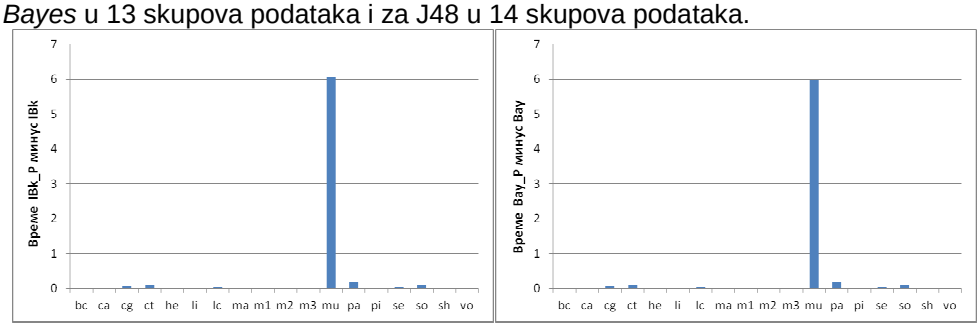
Tabela 9.3. Potrebno vreme za trening (u sekundama) različitih klasifikatora za originalni i redukovani skup podataka uz pomoć PCA

Skup	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	0.00	0.01 -	0.00	0.01 -	0.02	0.07 -	0.00	0.03 -	0.01	0.03 -
ca	0.00	0.03 -	0.00	0.03 -	0.14	0.26 -	0.01	0.07 -	0.03	0.08 -
cg	0.00	0.07 -	0.00	0.08 -	0.51	0.89 -	0.01	0.19 -	0.05	0.25 -
ct	0.00	0.12 -	0.01	0.13 -	3.74	0.66 +	0.10	0.27 -	0.36	0.41
he	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.01	0.01	0.01
li	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.00	0.01	0.01
lc	0.00	0.04 -	0.00	0.04 -	0.01	0.04 -	0.00	0.04 -	0.00	0.04 -
ma	0.00	0.00	0.00	0.00	0.13	0.10 +	0.01	0.01	0.02	0.03 -
m1	0.00	0.01	0.00	0.01 -	0.04	0.14 -	0.00	0.02 -	0.01	0.03 -
m2	0.00	0.00	0.00	0.01 -	0.05	0.10 -	0.00	0.02 -	0.01	0.03 -
m3	0.00	0.01 -	0.00	0.01 -	0.03	0.12 -	0.00	0.02 -	0.01	0.03 -
mu	0.00	6.06 -	0.00	5.96 -	3.73	19.52 -	0.03	7.60 -	0.51	8.19 -
pa	0.00	0.18 -	0.00	0.19 -	0.03	0.36 -	0.01	0.23 -	0.02	0.23 -
pi	0.00	0.00	0.00	0.01	0.15	0.10 +	0.01	0.02 -	0.03	0.03
se	0.00	0.04 -	0.01	0.05 -	3.75	0.55 +	0.08	0.16 -	4.04	3.58
so	0.00	0.09 -	0.00	0.09 -	0.75	1.14 -	0.01	0.26 -	0.79	1.26 -
sh	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.01	0.01	0.02 -
vo	0.00	0.00	0.00	0.01 -	0.02	0.03 -	0.00	0.01 -	0.02	0.03 -

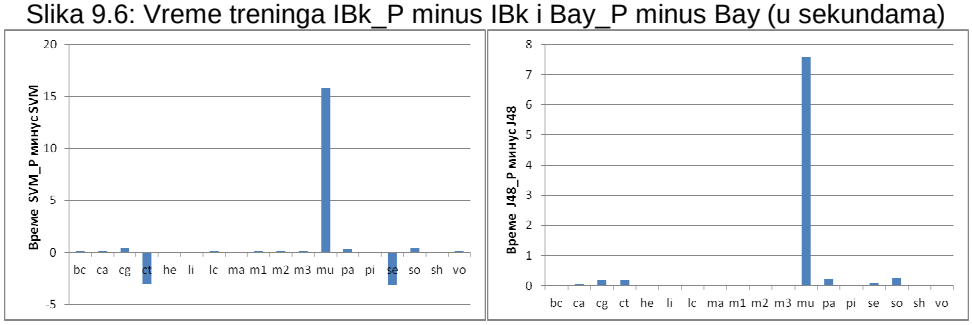
Za potrebno vreme treninga podataka SVM klasifikatora za sve originalne skupove podataka ono iznosi manje od 3.75 sekunde, dok je za SVM sa PCA ono

veće, osim u skupovima podataka *ct*, *li*, *ma*, *pi* i *se*. Potrebno vreme za trening podataka J48 klasifikatora za sve originalne skupove podataka iznosi manje od 0.10 sekundi, dok za J48 sa PCA je ono takođe nešto veće. Potrebno vreme za trening podataka RBF klasifikatora za sve originalne skupove podataka iznosi manje od 4.04 sekunde, dok je za RBF sa PCA ono veće, osim u skupu podataka *se*.

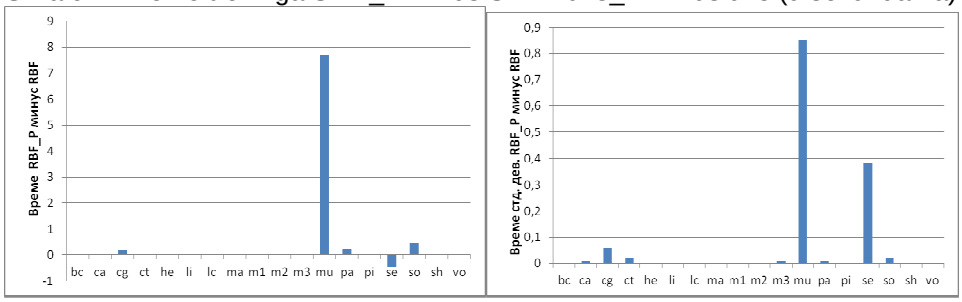
Na slikama 9.6, 9.7. i 9.8. prikazana je apsolutna razlika u potrebnom vremenu za trening različitih algoritma na osnovnom skupu podataka i istih algoritama sa PCA metodom za redukciju dimenzionalnosti podataka. Kod tri algoritma, IBk, *Naïve Bayes* i J48 u svim skupovima podataka PCA metoda je pokazala nešto lošije ili iste rezultate za potrebno vreme za trening. Rezultati su bili i statistički lošiji kod ovih algoritama i to: za IBk u 10 skupova podataka, za *Naïve Bayes* u 13 skupova podataka i za J48 u 14 skupova podataka.



Slika 9.6: Vreme treninga IBk_P minus IBk i Bay_P minus Bay (u sekundama)



Slika 9.7: Vreme treninga SVM_P minus SVM i J48_P minus J48 (u sekundama)



Slika 9.8: Vreme treninga RBF_P minus RBF (u sekundama) i standardna devijacija za vreme RBF_P minus RBF

Algoritam SVM je u 7 slučaja pokazao iste ili bolje rezultate za potrebno vreme za trening od SVM algoritma na osnovnom skupu podataka, a u 4 skupa podataka rezultati su bili i statistički bolji. Algoritam RBF je u 1 slučaju pokazao iste ili bolje rezultate za potrebno vreme za trening od RBF algoritma na osnovnom skupu podataka, ali u tom skupu podataka rezultat nije bio i statistički bolji.

Korišćenjem PCA metode, možemo da zaključimo da je SVM algoritam u najvećem broju slučaja doveo do statistički boljih rezultata za potrebno vreme za trening na posmatranim skupovima podataka. Takođe, možemo uočiti da PCA metoda nije dovela do značajnijeg povećanja potrebnog vremena za trening podataka, u odnosu na metodu prethodnog učenja.

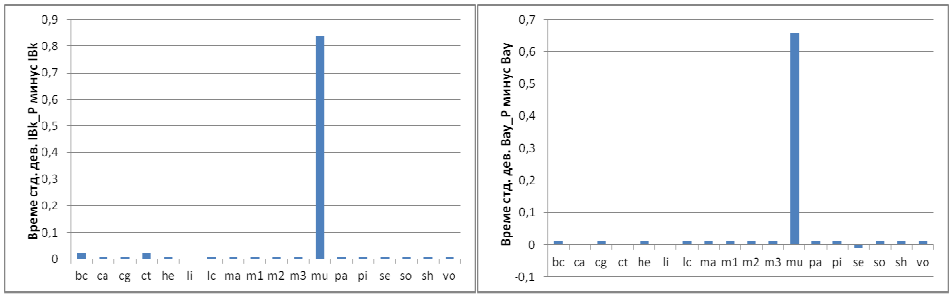
Tabela 9.4. prikazuje standardnu devijaciju za vreme treninga različitih algoritma za originalni i redukovani skup podataka uz pomoć PCA metode. Iz tabele se može videti da se standardne devijacije generalno ne razlikuju puno između standardnog algoritma i algoritama koji koriste ekstrakciju atributa. Nešto veće vrednosti za standardnu devijaciju za vreme treninga imaju svi algoritmi za jedan set podataka *mu*.

Tabela 9.4. Standardna devijacija za potrebno vreme za trening (u sekundama) različitih klasifikatora za originalni i redukovani skup podataka uz pomoć PCA

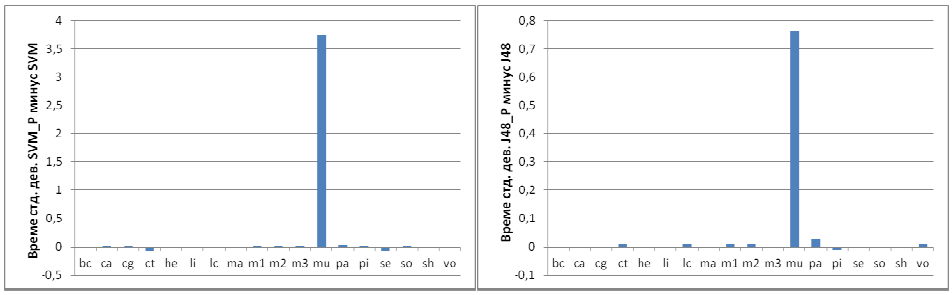
Skup	IBk	IBk_P	Bay	Bay_P	SVM	SVM_P	J48	J48_P	RBF	RBF_P
bc	0.00	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
ca	0.00	0.01	0.00	0.00	0.01	0.02	0.01	0.01	0.01	0.02
cg	0.00	0.01	0.00	0.01	0.02	0.03	0.01	0.01	0.01	0.07
ct	0.00	0.02	0.01	0.01	0.14	0.07	0.01	0.02	0.08	0.10
he	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
li	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01
lc	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.01
ma	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
m1	0.00	0.01	0.00	0.01	0.01	0.02	0.00	0.01	0.01	0.01
m2	0.00	0.01	0.00	0.01	0.01	0.02	0.00	0.01	0.01	0.01
m3	0.00	0.01	0.00	0.01	0.01	0.03	0.00	0.00	0.00	0.01
mu	0.00	0.84	0.01	0.67	0.07	3.81	0.00	0.76	0.07	0.92
pa	0.00	0.01	0.00	0.01	0.01	0.04	0.01	0.04	0.01	0.02
pi	0.00	0.01	0.00	0.01	0.01	0.02	0.01	0.00	0.01	0.01
se	0.00	0.01	0.01	0.00	0.16	0.08	0.01	0.01	1.30	1.68
so	0.00	0.01	0.00	0.01	0.07	0.08	0.01	0.01	0.09	0.11
sh	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
vo	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.01

Na slikama 9.8, 9.9. i 9.10. prikazana je apsolutna razlika u vrednostima standardne devijacije za vreme treninga različitih algoritma za originalni i redukovani skup podataka uz pomoć PCA metode. Ako je vrednost na slikama približna nuli, onda se standardne devijacije mnogo ne razlikuje, a ukoliko ona više odstupa od nule, to je i veće odstupanje između standardnih devijacija. Najveće

odstupanje u standardnoj devijaciji u odnosu na originalni skup podataka pokazuje SVM algoritam.



Slika 9.9: Standardna devijacija za vreme IBk_P minus IBk i Bay_P minus Bay



Slika 9.10: Standardna devijacija za vreme SVM_P minus SVM i J48_P minus J48

10. DISKUSIJA REZULTATA I DALJA ISTRAŽIVANJA

U poslednjem delu, biće dat rezime rada, a potom i zaključci razmatranja o uticaju prethodne selekcije atributa na klasifikacijske performanse algoritama nadziranog učenja. Na kraju, biće prikazani pravci mogućih daljih istraživanja u ovoj oblasti.

10.1. Rezime

Osnovna hipoteza je da je moguće znatno poboljšati performanse sistema za induktivno učenje pravila u problemima klasifikacije, primenom različitih metoda i tehnika redukcije dimenzionalnosti podataka. Da bi se dokazala postavljena hipoteza, implementirane su i empirijski testirane različite metode i tehnike redukcije dimenzionalnosti podataka.

U radu se razmatra problem redukcije dimenzinalnosti podataka, gde se selekcija atributa definiše kao proces koji bira minimalni podskup M atributa iz izvornog skupa N atributa, tako da je prostor atributa optimalno smanjen prema određenom kriteriju ocenjivanja. Pronalaženje najboljeg podskupa atributa je obično nerešiv problem i mnogi problemi vezani za izbor atributa su se pokazali da su *NP*-teški. Svi atributi mogu biti važni za neke probleme, ali za neka ciljana istraživanja, samo mali podskup atributa je obično relevantan. Algoritme za selekciju atributa smo podelili na filtere, metode prethodnog učenja i ugrađene pristupe. Filter metode ocenjuju kvalitet odabranih atributa nezavisno od algoritma za klasifikaciju, dok su metode prethodnog učenja metode koje zahtevaju primenu klasifikatora (koji bi trebao biti treniran na određenom podskupu atributa) za procenu kvaliteta. Ugrađene metode obavljaju izbor atributa tokom učenja optimalnih parametara (za na primer, neuronske mreže težine između ulaznog i skrivenog sloja).

Glavni cilj ovog rada je proveriti uticaj različitih filter metoda, metoda prethodnog učenja, ugrađenih metoda i ekstrakcije atributa na tačnost klasifikacije. Eksperimentalna istraživanja su sprovedena uz korišćenje veštačkih i prirodnih skupova podataka. Eksperimentalni rezultati pokazuju da primenjene metode efikasno doprinose otkrivanju i eliminisanju nebitnih, redundatnih podataka, kao i šuma u podacima. U mnogim slučajevima opisane metode prethodne selekcije atributa odabiraju relevantne atribute u skupovima podataka, i doprinose većoj tačnosti klasifikacije.

Generalno, od tri grupe metoda, najbolje rezultate vezano za tačnost klasifikacije pokazale su metode prethodnog učenja. Uopšteno, nedostatak metoda filtriranja koji vrednuju pojedinačne atribute je nemogućnost detekcije redundantnih atributa, zbog čega korelisanost nekog atributa s drugim rezultira sličnom ocenom

vrednosti za oba atributa, pa će po pravilu oba atributa biti prihvaćena ili odbačena. Sledeći nedostatak ovih metoda je da je uvrštavanje atributa u konačni podskup prepušteno spoljnim kriterijima praga vrednosti ili broja atributa.

Metode filtriranja koje vrednuju podskupove atributa su vremenski zahtevnije od filtera koji vrednuju pojedinačne atribute, jer postoji potreba pretraživanja podskupova atributa. Međutim, ovi zahtevi su neuporedivo manji u poređenju sa metodama prethodnog učenja jer se u svakom koraku pretraživanja izračunava samo vrednost heuristike vrednovanja, a nije potrebno više puta pozivati algoritam mašinskog učenja.

Generalno, odabir atributa metodama filtriranja traje znatno kraće u poređenju sa metodama prethodnog učenja, posebno kad su u pitanju skupovi podataka sa većim brojem atributa, zbog čega su metode filtriranja često praktičnije rešenje za analizu podataka od drugih metoda. Metode filtriranja se zbog nezavisnosti o algoritmu mašinskog učenja mogu koristiti u kombinaciji sa bilo kojom tehnikom modeliranja podataka, za razliku od metoda prethodnog učenja koji se moraju ponovo izvoditi pri svakoj promeni ciljne tehnike modeliranja.

Kod metoda prethodnog učenja najvažniji nedostatak je sporost pri izvođenju uslovljena pozivanjem ciljnog algoritma mašinskog učenja više puta, zbog čega ovim metodama ne odgovaraju obimni skupovi podataka za učenje sa većim brojem atributa.

Smatra se da metode prethodnog učenja omogućuju postizanje nešto boljih performansi klasifikacije, zbog tesne povezanosti s ciljnim algoritmom mašinskog učenja. Ovo ujedno može predstavljati i opasnost jer preterano prilagođavanje skupa za učenje ciljnom algoritmu može naglasiti njegove nedostatke.

Metoda ekstrakcije atributa uz pomoć PCA metode očekuje da će većina novih varijabli činiti šum i imati tako malu varijansu da se ona može zanemariti, na osnovu čega se iz velikog broja izvornih varijabli kreira tek nekoliko glavnih komponenti koje nose većinu informacija i čine glavni oblik. Međutim, ima situacija kada to nije tako, i to u slučaju kada su izvorne varijable nekorelisane, tada analiza ne daje povoljne rezultate. Kada su izvorne varijable visoko pozitivno ili negativno korelisane mogu se postići najbolji rezultati. Još jedan problem kod ove metode je nemogućnost smislene interpretacije glavnih komponenata.

10.2. Zaključci

U radu pokazujemo da nema jedne najbolje metode za redukciju dimenzionalnosti podataka, i da izbor zavisi od osobina posmatranog skupa podataka i primenjenih klasifikatora. U praktičnim problemima, jedini način kako bi bili sigurni da je najviša preciznost dobijena je testiranje datog klasifikatora sa više različitih podskupova atributa, dobijenih različitim metodama za prethodnu selekciju atributa. Sprovedena istraživanja u nadgledanom učenju, pokušavaju dati uvid u prednosti i ograničenja različitih metoda prethodne selekcije atributa. Sa ovakvim uvidom i predznanjem za određeni konkretni problem, stručnjaci mogu odabrati koje metode treba primeniti. Takav je slučaj sa nekim od metoda

prethodne selekcije atributa, koje mogu da poboljšaju (ili da ne degradiraju) izvršenje algoritama mašinskog učenja, dok u isto vreme postižu smanjenje broja atributa koji se koriste u učenju. Neke od prikazanih metoda, imale su problem kod izbora relevantnih atributa, kada u podacima postoji snažna interakcija između atributa, ili kada imamo skupove podataka sa oskudnim brojem instanci.

10.3. Dalja istraživanja

Uočeno je da neke od metoda predhodne selekcije atributa imaju problem da izaberu atribut koji ima lokalne prediktivne mogućnosti, jer se dešava da su zasenjeni od strane atributa koji imaju jake, na globalnom nivou prediktivne mogućnosti. Ako je broj takvih atributa veći, onda se može pojaviti i kumulativni efekat. U tim slučajevima, može se dopustiti redundansa u skupu podataka, ako ona nema negativne efekte na algoritme nadziranog učenja. Obično ova redundansa nema negativne efekte na C4.5 i IB1, ali može imati na *Naïve Bayes*.

Bilo bi interesantno primeniti neke od tehnika u rešavanju problema otkrivanja lokalnog nivoa prediktivnosti atributa. Te tehnike bi mogle poboljšati klasifikacijske performanse algoritama učenja. Takav pristup zahteva korišćenje određenog algoritma učenja i odgovarajuće tehnike koja bi u kombinaciji sa metodama predhodne selekcije atributa rezultirala hibridnim sistemom.

Atributi izabrani od strane metoda prethodne selekcije atributa uglavnom predstavljaju dobru osnovu za formiranje odgovarajućeg podskupa atributa. Bilo bi interesantno istražiti kako metode prethodnog učenja reaguju, ako koriste kao početni podskup atributa onaj koje je odabrala neka od metoda filtriranja. To znači, da bi se u ovom slučaju, umesto pretraživanja unapred i u nazad, kada se koristi u startnoj poziciji prazan ili potpun skup atributa, koristio početni podskup atributa koje je izabrala odgovarajuća metoda filtriranja. Na taj način bi se smanjilo potrebno vreme za traženje odgovarajućeg podskupa metodama prethodnog učenja. U našim eksperimentalnim rezultatima smo pokazali da se metodama prethodnog učenja dobija u najvećem broju slučaja bolja tačnost klasifikacije, ali da je potrebno vreme za izvršenje ove metode veliko. Takođe, ovaj pristup može poboljšati performanse klasifikacije kod metoda prethodnog učenja kada metode rade sa manjim skupovima gde se dobijaju manje tačnosti klasifikacije, jer se dešava da u tim slučajevima postanu zarobljene u lokalnom maksimumu.

LITERATURA

[1] N. Abe, M. Kudo, *Entropy criterion for classifier-independent feature selection*, Lecture Notes in Computer Science, Volume 3684/2005, 689-695, 2005.

[2] D. Aha, *Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms*, International Journal of Man-Machine Studies, Volume 36, Issue 2, pp. 267–287, Academic Press Ltd, London, UK, Feb. 1992.

[3] H. Almuallim, T. G. Dietterich, *Learning with many irrelevant features*, Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), pp. 547-552, Anaheim, CA: AAAI Press, 1991.

[4] D. Ayres de Campos et al., *SisPorto 2.0 A Program for automated analysis of cardiotocograms*, J Matern Fetal Med 5:311-318, 2000.

[5] D. Ayres de Campos, P. Sousa, A. Costa, J. Bernardes, *Omniview-SisPorto® 3.5—a central fetal monitoring station with online alerts based on computerized cardiotocogram ST event analysis*, J. Perinat. Med. 2008; 36:260-4.

[6] M. Ben-Bassat, *Pattern recognition and reduction of dimensionality*, In P. R. Krishnaiah and L. N. Kanal, editors, Handbook of statistics-II, pp 773-791, North Holland, 1982.

[7] A.L. Blum, R.L. Rivest, *Training a 3-node neural networks is NP-complete*, Neural Networks, 5:117-127, 1992.

[8] A.I. Blum, P. Langley, *Selection of relevant features and examples in machine learning*, Artificial Intelligence, vol 97, 1997, 245-271.

[9] L. Breiman, J.H. Friedman, R.H. Olshen, C.J. Stone, *Classification and regression trees*, Wadsworth and Brooks, Monterey, CA, 1984.

[10] D.S. Broomhead, D. Lowe, *Multivariate functional interpolation and adaptive networks*, Complex Systems, 2:321-355, 1988.

[11] R. Caruana, D. Freitag, *Greedy attribute selection*, In Proceedings of International Conference on Machine Learning (ICML-94), Menlo Park, California, 1994, AAAI Press/The MIT Press, 28–36.

[12] G. Cestnik, I. Konenenko, I. Bratko,I., *Assistant-86: a knowledge-elicitation tool for sophisticated users*, In I.Bratko & N.Lavrac (Eds.) Progress in Machine Learning, 31-45, Sigma Press, 1987.

[13] C. Chang, C. Lin, *LIBSVM: a Library for Support Vector Machines*, 2001. http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.

[14] J.G. Cleary, L.E. Trigg, *K*: An instance-based learner using an entropic distance measure*, In Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, 1995.

[15] S. Das, *Filters, wrappers and a boosting-based hybrid for feature selection*, In Proceedings of the Eighteenth International Conference on Machine Learning, 2001.

[16] M. Dash, H. Liu, *Feature selection methods for classifications*, Intelligent Data Analysis: An International Journal, 1(3), 1997.
[17] M. Dash, H. Liu, J. Yao, *Dimensionality reduction of unsupervised data*, In Proceedings of the Ninth IEEE International Conference on Tools with AI (ICTAI'97), November, 1997, Newport Beach, California, 1997, IEEE Computer Society, 532–539.
[18] M. Dash, H. Liu, *Handling large unsupervised data via dimensionality reduction*, In Proceedings of 1999 SIGMOD Research Issues in Data Mining and Knowledge Discovery (DMKD-99) Workshop, 1999.
[19] P. Diaconis, B. Efron, *Computer-intensive methods in statistics*, Scientific American, Volume 248, 1983.
[20] J. Doak, *An evaluation of feature selection methods and their application to computer security*, Technical report, Davis CA: University of California, Department of Computer Science, 1992.
[21] W. Duch, R. Adamczak, K. Grabczewski, *A new methodology of extraction, optimization and application of crisp and fuzzy logical rules*, IEEE Transactions on Neural Networks, vol. 12, pp. 277-306, 2001.
[22] J. G. Dy, C.E. Brodley, *Feature subset selection and order identification for unsupervised learning*, In Proceedings of the Seventeenth International Conference on Machine Learning, 2000, 247–254.
[23] B. Efron, R. Tibshirani, *An introduction to the bootstrap*, Chapman and Hall, London, 1993.
[24] M. Elter, R. Schulz-Wendtland, T. Wittenberg, *The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process*, Medical Physics 34(11), pp. 4164-4172, 2007.
[25] U.M. Fayyad, K.B. Irani, *The attribute selection problem in decision tree generation*, In AAAI-92, Proceedings of the Ninth National Conference on Artificial Intelligence, AAAI Press/The MIT Press, 1992, 104–110.
[26] E. Fix, J.L. Hodges, *Discriminatory analysis; non-parametric discrimination: consistency properties*, Technical Report 21-49-004(4), USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
[27] T. Fletcher, *Support Vector Machines Explained*, 2009, http://www.tristanfletcher.co.uk/SVM%20Explained.pdf.
[28] A. Frank, A. Asuncion, *UCI Machine learning repository* [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science, 2010.
[29] M.A. Hall, L.A. Smith, *Practical feature subset selection for machine learning*, Proceedings of the 21st Australian Computer Science Conference, 181–191, 1998.
[30] Mark A. Hall, *Correlation-based feature selection for machine learning*, The University of Waikato, Doctoral disseration, 1999.
[31] R.C. Holte, *Very simple classification rules perform well on most commonly used datasets*, Machine Learning, 11:63-91, 1993.
[32] Z.Q. Hong, J.Y. Yang, *Optimal discriminant plane for a small number of samples and design method of classifier on the plane*, Pattern Recognition, Vol. 24, No. 4, pp. 317-324, 1991.

[33] A. Hutchinson, *Algorithmic learning*, Clarendon Press, Oxford, 1993.
[34] Jakulin, I. Bratko, *Analyzing attribute dependencies*, Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003), Cavtat-Dubrovnik, Croatia, September 22-26, 2003.
[35] Jakulin, I. Bratko, *Testing the significance of attribute interactions*, Proceedings of the Twenty-first International Conference on Machine Learning (ICML-2004), Eds. R. Greiner and D. Schuurmans, pp. 409-416, Banff, Canada, 2004.
[36] P. Janičić, M. Nikolić, *Veštačka inteligencija*, Matematički fakultet u Beogradu, pp. 161, 2010.
[37] M.V. Johns, *An empirical Bayes approach to non-parametric two-way classification*, Studies in item analysis and prediction, Stanford University Press, Palo Alto, 1961.
[38] G.H. John, R. Kohavi, K. Pfleger, *Irrelevant feature and the subset selection problem*, In W.W. and Hirsh H. Cohen, editor, Machine Learning: Proceedings of the Eleventh International Conference, New Brunswick, N.J., 1994, Rutgers University, 121–129.
[39] Y. Kim, W. Street, F. Menczer, *Feature selection for unsupervised learning via evolutionary search*, In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, 365–369.
[40] K. Kira, L.A. Rendell, *The feature selection problem: tradional methods and a new algorithm*, In: Proc. AAAI-92, San Jose, CA, 1992, 122-126.
[41] R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Francisco, 1995.
[42] R. Kohavi, G.H. John, *Wrappers for feature subset selection*, Artificial Intelligence - Special issue on relevance archive, Volume 97 Issue 1-2, Dec. 1997, pp. 273 – 324.
[43] R. Kohavi, F. Provost, *Glossary of terms*, Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process (volume 30, Number 2/3, February/March 1998).
[44] D. Koller, M. Sahami, *Toward optimal feature selection*, In International Conference on Machine Learning, 1996, 284-292.
[45] Kononenko, *Esimating attributes: analysis and extensions of Relief*, In Proceeding of the European Conference on Machine Learning, 1994.
[46] S.B. Kotsiantis, *Supervised machine learning: a review of classification techniques*, Informatica 31(2007) 249-268, 2007.
[47] M. Kubat, R. Holte, S. Matwin, *Machine learning for the detection of oil spills in satellite radar images*, Machine Learning, 30, 195–215, 1998.
[48] N. Lavrac, D. Gamberger, H. Blockeel, L. Todorovski (Eds.), *Lecture notes in artificial intelligence*, Vol. 2838, Springer, pp. 229-240, 2003.
[49] M.A. Little, P.E. McSharry, S.J. Roberts, D.A.E. Costello, I.M. Moroz, *Exploiting nonlinear recurrence and fractal scaling properties for voice*

disorder detection, BioMedical Engineering OnLine 2007, 6:23, 26 June 2007.
[50] M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, L.O. Ramig, *Suitability of dysphonia measurements for telemonitoring of Parkinson's disease*, IEEE Transactions on Biomedical Engineering in 2009, 56(4):1015-1022.
[51] H. Liu, R. Setiono, *Chi2: Feature selection and discretization of numeric attributes*, Proc. IEEE 7th International Conference on Tools with Artificial Intelligence, 338-391, 1995.
[52] H. Liu, R. Setiono, *A probabilistic approach to feature selection - a filter solution*, In L. Saitta, editor, Proceedings of International Conference on Machine Learning (ICML-96), July 3-6, 1996, Bari, Italy, 1996, San Francisco: Morgan Kaufmann Publishers, CA, 319–327.
[53] H. Liu, H. Motoda, *Feature selection for knowledge discovery and data mining*, Kluwer Academic Publishers, 1998.
[54] B.J.F. Manly, *Multivariate methods*, Chapman & Hall, London, UK, 1986.
[55] R.S. Marko, K. Igor, *Theoretical and empirical analysis of relief and ReliefF*, Machine Learning Journal, 53:23–69. doi: 10.1023/A:1025667309714, 2003.
[56] R.S. Michalski, R.L. Chilausky, *Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis*, International Journal of Policy Analysis and Information Systems, Vol. 4, No. 2, 1980.
[57] P. Mitra, C. A. Murthy, S. K. Pal, *Unsupervised feature selection using feature similarity*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3):301–312, 2002.
[58] V. Mišković, *Induktivno učenje razumljivog znanja na osnovu oskudnih obučavajućih skupova*, doktorska disertacija, Univerzitet Singidunum, Beograd, 2008.
[59] V. Mišković, M. Milosavljević, *Ugrađeni metodi selekcije atributa u algoritmima induktivnog učenja*, LIII konferencija „ETRAN 2010", Donji Milanovac, Srbija, 2010.
[60] J. Moody, C. Darken, *Fast learning in networks of locally-tuned processing units*, Neural Computation, 1:281-294, 1989.
[61] T. Nakagawa, T. Harab, H. Fujitab, T. Iwasec, T. Endod, K. Horitae, *Automated contour extraction of mammographic mass shadow using an improved active contour model*, Elsevier, International Congress Series, Volume 1268, June 2004, pp 882–885.
[62] Petrović, *Osnove inteligentnog upravljanja (sustavi upravljanja zasnovani na umjetnim neuronskim mrežama)*, Zagreb, 2009.
[63] J.H. Piater, E.M. Riseman, P.E. Utgoff, *Interactively training pixel classifiers*, Published in the International Journal of Pattern Recognition and Artificial Intelligence 13(2), 1999.
[64] G. Piatetsky-Shapiro, W.J.E. Frawley, *Knowledge discovery in databases*, MIT Press, Cambridge, Mass., 1991.

[65] J.C. Platt, *Fast training of Support Vector Machines using sequential minimal optimization*, Advances in kernel methods, Pages 185-208, MIT Press Cambridge, MA, USA, 1999.
[66] F. Provost, T. Fawcett, *Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions*, In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97).
[67] S. Russell, P. Norvig, *Artificial intelligence: a modern approach*, Second edition, Prentice Hall, 2003.
[68] J.S. Schlimmer, *Concept acquisition through representational adjustment* (Technical Report 87-19), Doctoral disseration, Department of Information and Computer Science, University of California, Irvine, 1987.
[69] S. Sharma, *Applied multivariate techniques,* New York: John Wiley and Sons, Inc, 1996.
[70] W. Siedlecki, J. Sklansky, *On automatic feature selection*, International Journal of Pattern Recognition and Artificial Intelligence, 2:197–220, 1988.
[71] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, R.S. Johannes, *Using the ADAP learning algorithm to forecast the onset of diabetes mellitus*, In Proceedings of the Symposium on Computer Applications and Medical Care, pp. 261-265, IEEE Computer Society Press, 1988.
[72] J. Swets, *Measuring the accuracy of diagnostic systems*, Science 240, 1285-1293, 1988.
[73] L. Talavera, *Feature selection as a preprocessing step for hierarchical clustering*, In Proceedings of Internationl Conference on Machine Learning (ICML'99), 1999.
[74] J. R. Taylor, *An introduction to error analysis: the study of uncertainties in physical measurements*, University Science Books, Sausalito, CA, 1999, pp. 128–129.
[75] S.B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Dzeroski, S.E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R.S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, J. Zhang, *The MONK's problems - a performance comparison of different learning algorithms*, Technical Report CS-CMU-91-197, Carnegie Mellon University in Dec. 1991.
[76] F. Ujević, *Postupci analize podataka u izgradnji profila korisnika usluga*, magistarski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 2004.
[77] D. Vranješ, *Modeliranje trenja primjenom RBF neuronskih mreža*, diplomski rad br. 1364, Sveučilište u Zagrebu, Fakultet elektrotrhnike i računarstva, Zagreb, srpanj 2003.
[78] S.M. Weiss, C.A. Kulikowski, *Computer systems that learn*, Morgan Kaufmann Publishers, San Mateo, California, 1991.
[79] J.R. Quinlan, *Induction of decision trees*, Machine Learning 1, 1986.
[80] J.R. Quinlan, *Simplifying decision trees*, Int J Man-Machine Studies 27, Dec 1987, pp. 221-234.

[81] J.R. Quinlan, *C4.5: Programs for machine learning*, San Mateo, Morgan Kaufman, 1993.
[82] J.R. Quinlan, *Comparing connectionist and symbolic learning methods*, In Computational Learning Theory and Natural Learning Systems, Vol.1, MIT Press, Cambridge, 1994.
[83] I.H. Witten, E. Frank, *Data mining: practical machine learning tools and techniques*, Second Edition, Morgan Kaufmann, San Francisco, 2005, pp. 17.
[84] N. Wyse, R. Dubes, A.K. Jain, *A critical evaluation of intrinsic dimensionality algorithms*, In E.S. Gelsema and L.N. Kanal, editors, Pattern Recognition in Practice, pp 415–425, Morgan Kaufmann Publishers, Inc., 1980.
[85] E. Xing, M. Jordan, R. Karp, *Feature selection for high-dimensional genomic microarray data*, In Proceedings of the Eighteenth International Conference On Machine Learning, 2001.
[86] J. Yang, V. Honavar, *Feature subset selection using a genetic algorithm*, IEEE Intelligent Systems 13:44-49, 1998.

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

62
005.591.4:658.5
005.8

NOVAKOVIĆ, Novaković J., 1965-
Rešavanje klasifikacionih problema mašinskog učenja
= Solving Machine Learning Classification Problems
/ Jasmina Đ. Novaković: Fakultet tehničkih nauka,
2013 (Vrnjačka Banja : SaTCIP). - 196 str. :
graf. prikazi, tabele ; 25 cm. -
(Rešavanje klasifikacionih problema mašinskog učenja; #knj. #4 =
Business Process Reingeneering ; #vol. #1 /
urednik serije Alempije V. Veljović)

Na nasl. str.: Univerzitet u Kragujevcu. -
Tiraž 100.
Napomene i bibliografske reference uz
tekst. - Bibliografija: str. 191-196.

ISBN 978-86-7776-157-8
1. Уп. ств. насл.
a) Реинжењерство b) Пословни процеси -
Реинжењеринг c) Управљање пројектима
COBISS.SR-ID 200078348