

Departman za poslediplomske studije

Savremene informacione tehnologije

Master studijski program

-Master rad-

Istraživanje podataka uz pomoć softverskog alata Rapidminer

Mentor:

Prof. dr Milan Milosavljević

Student:

Palkovač Atila

Broj indeksa : 410113/2014

Beograd, 2015

Sadržaj

Uvod.....	5
Abstract.....	5
1. DataMining	7
1.1. Uvod	7
1.2. Šta je istraživanje podataka (Data Mining)	7
1.3. Proces istraživanja podataka.....	8
1.4. Modeli istraživanja podataka.....	10
1.4.1. Regresija	10
1.4.2. Klasifikacija.....	11
1.4.3. Klasterovanje	12
1.4.4. Predviđanje	13
1.4.5. Analiza serija tokom vremena	13
1.4.6. Sumiranje.....	13
1.4.7. Asocijativna pravila	14
1.5. Čišćenje podataka	14
1.6. Rad sa podacima koji nedostaju	15
2. Primena ID3 algoritma (Stablo odlučivanja) u istraživanju podataka i grafički prikaz formiranog stabla odlučivanja.....	16
2.1. Uvod	16
2.2. Metoda učenja.....	17
2.3. Entropija kao klasifikator	18
2.4. Informacijski dobitak (Gain)	19
2.5. Podrezivanje stabla odlučivanja	19
2.6. Modifikacija ID3 algoritma	20
2.7. Grafički prikaz izračunatog stabla odlučivanja	33
2.8. Prednosti i nedostaci	35
2.9. Rad sa aplikacijom za mašinsko učenje uz pomoć ID3 algoritma	36
2.10. Mogućnosti primene stabla odlučivanja u raznim oblastima	39
3. Osnove rada sa softverskim alatom Rapidminer	41
3.1. Uvod	41
3.2. Osnovne funkcije Rapidminer-a	41
3.3. Povezivanje Rapidminer-a sa podacima	42
3.4. Kreiranje procesa u Rapidminer-u.....	44
3.5. Tipovi podataka u Rapidminer-u	47
3.6. Najznačajniji operatori	47
3.6.1. Process Control.....	48
3.6.1.1. Handle Exception	48

3.6.1.2.	Throw Exception	48
3.6.1.3.	Set Parameters	48
3.6.1.4.	Branch	49
3.6.2.	Utility.....	49
3.6.2.1.	Subprocess.....	49
3.6.2.2.	Set Macros	49
3.6.2.3.	Execute SQL.....	50
3.6.2.4.	Generate Sales Data.....	50
3.6.3.	Repository Access	51
3.6.3.1.	Retrieve	51
3.6.3.2.	Store.....	51
3.6.4.	Import	51
3.6.4.1.	Read Excel.....	51
3.6.4.2.	Read CSV	51
3.6.4.3.	Read Database	52
3.6.4.4.	Read Model	52
3.6.5.	Export	52
3.6.5.1.	Write Access.....	52
3.6.5.2.	Write CSV	52
3.6.5.3.	Write Database	53
3.6.5.4.	Write Model	53
3.6.6.	Data Transformation.....	53
3.6.6.1.	Rename	53
3.6.6.2.	Set Role	53
3.6.6.3.	Nominal to Text.....	54
3.6.6.4.	Discretize by Entropy.....	54
3.6.6.5.	Generate Atributes.....	55
3.6.6.6.	Select Atributes	55
3.6.6.7.	Remove Useless Atributes.....	55
3.6.6.8.	Set Data	55
3.6.6.9.	Normalize	56
3.6.6.10.	Replace Missing Values	56
3.6.6.11.	Filter Examples.....	56
3.6.6.12.	Pivot.....	57
3.6.6.13.	Append	57
3.6.7.	Modeling.....	57
3.6.8.	Evaluation.....	61
3.6.8.1.	Split Validation.....	61
3.6.9.	Connectors.....	61
3.6.9.1.	Read Dropbox.....	61
3.6.9.2.	Read Salesforce	62
3.6.9.3.	Search Twitter	62
3.7.	Portovi	62

4. Rapidminer u akciji	67
4.1. Uvod	67
4.2. Model za testiranje SPAM poruka.....	67
4.2.1. Ukratko o kreiranje modela za testiranje SPAM poruka.....	67
4.2.2. Kreiranje modela za testiranje SPAM poruka	69
4.2.3. Rezultat istraživanja modela za testiranje SPAM poruka	81
5. Zaključak	92

Uvod

U današnjem svetu smo okruženi raznim informacijama, koje nam mogu pomoći u donošenju raznih svakodnevnih i poslovnih odluka. Problem je što su ti podaci često sakriveni u gomili nepotrebnih informacija, pa ih nije lako uočiti. Za rešavanje ovog problema se brine nova naučna grana sa nazivom Istraživanje podataka (Data mining).

Trenutno postoji mnogo alata sa kojim možemo istraživati podatke, ali moramo naglasiti da ne postoji ni jedan univerzalan, koji će nam rešiti sve probleme. Svaki problem istraživanja podataka je novi izazov za koji ne znamo tačno kojim ćemo metodom ili alatom dobiti najbolji rezultat. Zbog toga se obično koriste više alata čije rezultate upoređujemo i biramo najbolje rešenje.

Ovaj rad je napisan sa ciljem da se što bolje objasni šta je ustvari istraživanje podataka, zatim kako se pravi aplikacija za istraživanje podataka i na kraju kako radi jedna aplikacija za istraživanje podataka.

U prvom poglavlju je teoretski deo vezan za istraživanje podataka. Tu su navedeni razni modeli za klasifikacije, kao i problemi koji se javljaju kod istraživanja podataka.

Drugo poglavlje se odnosi na objašnjenje kako se pravi program za istraživanje podataka. U ovom slučaju je to formiranje stabla odlučivanja (uz pomoć ID3 algoritma), kao i modifikacija samog algoritma koje se odnosi na lakše crtanje stabla odlučivanja.

Treće poglavlje se odnosi na upoznavanje sa softverskim alatom Rapidminer i prikazivanjem njegovih mogućnosti.

U četvrtom poglavlju je urađen jedan projekat istraživanja podataka, u kojem je napravljen model za klasifikaciju SPAM poruka. Na kraju su rezultati tog modela upoređeni sa rezultatima sličnih projekata koji su objavljeni na internetu.

Abstract

In today's world we are surrounded by all kinds of information that can help us in making a variety of everyday and business decisions. The problem is that these data are often hidden in a pile of unnecessary information, so it is not easy to spot. For solving this problem is a new branch of science concerned with the name of data mining.

Currently there are many tools with which we can explore information, but we must underline that there is no universal, that will solve all problems. Each issue of data mining is a new challenge for which we do not know exactly which method or tool will get the best result. Therefore, it is more commonly used tools which compare the results and choose the best solution.

This paper was written with the aim to better explain what is actually data mining, and then how to make applications for research data and finally how works one application for data mining.

In the first chapter is the theoretical part of the data mining. There are listed the different models for classification, as well as problems arising from data mining.

The second chapter refers to the explanation of how to make a program for data mining. In this case it is the creating of a decision tree (with the help of ID3 algorithm), as well as modifications of the algorithm that relates to more easily draw the decision tree.

The third chapter refers to the introduction of software tools Rapidminer and displaying its capabilities.

In the fourth chapter is done one project of data mining, in which we made a model for the classification of SPAM messages. At the end the results of the model were compared with the results of similar projects that have been published on the Internet.

1. Data mining

1.1. Uvod

Količina podataka u svetu se konstantno povećava i to je glavni razlog zašto istraživanje podataka (Data mining) počinje da izaziva toliko interesovanje. Glavni interes je da se iz te velike količine podataka izvuku korisne informacije i znanje, koje se mogu iskoristiti u raznim oblastima, kao što su menadžment, kontrola proizvodnje, analiza tržišta, itd.

Istraživanje podataka je prirodan rezultat evolucije informacione tehnologije. Od 1960 – ih godina baza podataka informaciona tehnologija se razvijaju iz primitivnih sistema za obradu fajlova do sofisticiranih i veoma moćnih sistema baza podataka. U 1970-im godinama je dalji razvoj baza podataka doveo do nastanka relacionih baza podataka, gde se podaci čuvaju u relacionim tabelama. Metodi za OLTP (on-line transaction processing), gde se upiti (query) vide kao read-only transakcije, postali su glavni alati za efikasno čuvanje, pretraživanje i upravljanje velikom količinom podataka. Sredinu 1980-ih godina karakteriše dalji razvoj relacionih tehnologija kao i istraživanje i razvoj novih sistema baza podataka, kao što su objektno orijentisani modeli. Baze podataka postaju orijentisane prema aplikacijama. Informacioni sistemi koji su zasnovani na internetu, kao što je WWW (World wide web) počinju da zauzimaju važnu ulogu u informacionoj industriji. Razvoj hardvera označava poslednjih desetak godina. Ovaj razvoj je doveo do toga da su postali dostupni veoma moćni računari, sa mogućnošću skladištenja veoma mnogo podataka. Dalji razvoj baza podataka doveo je do nastanka skladišta podataka (data warehouse). Skladišta podataka je skup podataka iz raznih izvora, koji su tako organizovani da je pristup tim podacima brz i jednostavan. Ovi podaci se obično koriste u menadžmentu za donošenje odluka. Tehnologija skladišta podataka uključuje čišćenje podataka, integraciju podataka i OLAP (On-line analytical processing). OLAP omogućuje funkcionalnosti kao što su sumiranje podataka, konsolidacija i agregacija podataka, sve to u funkciji sagledavanja podataka iz raznih uglova.

1.2. Šta je istraživanje podataka (Data Mining)

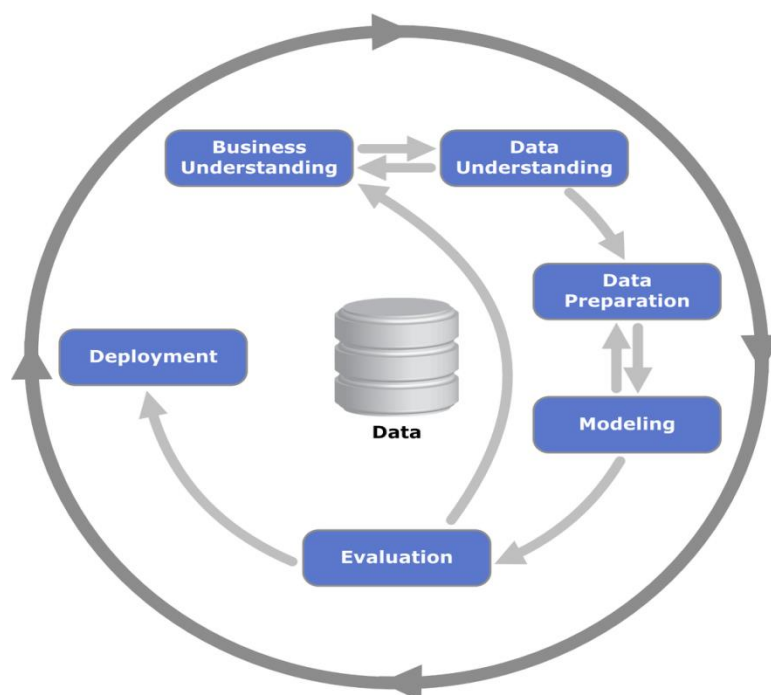
Istraživanje podataka je ukratko pronalaženje znanja iz velike količine podataka.

Kad god nešto kupimo preko interneta, platimo nešto kreditnom karticom ili čak surfujemo internetom mi formiramo neke podatke. Ovi podaci se skladište u velikim skladištima podataka raznih kompanija. Naši podaci ostavljaju trag o našim navikama, interesovanjima ili ponašanju. Ovi podaci omogućuju analitičarima da pronalaze razne obrasce i tako im olakšavaju donošenje raznih poslovnih odluka za bolje usluživanje kupaca i sl.

1.3. Proces istraživanja podataka

Iako se koreni istraživanja podataka mogu naći u 80-im godinama, sve do skoro nije bilo nikakvih standarda u ovoj oblasti. Tek 1999. godine se skupilo nekoliko velikih kompanija među kojima su bili Daimler-Benz, NCR Corp, SPSS Inc, i odlučili da standardizuju ovu oblast.

Kao rezultat ove saradnje je nastao tzv. CRISP-DM (CRoss-Industry Standard Process for Data Mining) standard u procesu istraživanju podataka.



Slika 1 CRISP-DM Process Diagram by Kenneth Jensen

Ovaj standard se sastoji od sledećih koraka:

a) Shvatanje organizacije

Ovaj korak je presudan korak za uspešno istraživanje podataka, ali ga mnogi zanemaruju u procesu istraživanja podataka. Cilj istraživanja je da se pronađu zanimljivi rezultati koje možemo iskoristiti za neki cilj naše kompanije. Ali ne možete početi da nešto gradite a da niste definisali svrhu za šta će nam trebati ono što gradimo.

Možemo istraživati podatke danima, ali ako ne znamo šta hoćemo da radimo, ili na koje pitanje tražimo odgovor, onda naš trud neće doneti nikakve rezultate.

b) Shvatanje podataka

Ovo je aktivnost koja ima za cilj otkrivanje suštine podataka kojima raspolazemo. Poželjno je znati odakle smo sakupili podatke, ko ih je sakupio, šta znače pojedine kolone, itd. Ponekad je potrebno da se sastanemo sa pojedincima koji su sakupili podatke da nam objasne poreklo i značenje podataka. Veoma je važno da se potvrdi verodostojnost i pouzdanost prikupljenih podataka.

Nepotpuni ili netačni podaci su mnogo gori nego da uopšte nemamo podatke, zato što su odluke koje su donose na osnovu takvih podataka pogrešne i mogu naneti štetu kompaniji.

c) Priprema podataka

Glavni zadatak ovog koraka je da se prečiste podaci da bi oni postali kvalitetniji za istraživanje.

Podaci koji se istražuju dolaze u mnogim oblicima i formatima. Neki su numerički, neki tekstualni ili možda slike. Ponekad podaci koji nisu u standardnom formatu kolona i redova mogu biti veoma pogodni za istraživanje.

Priprema podataka uključuje mnogo aktivnosti. Možemo spajati dva ili više seta podataka, ili smanjiti set podataka samo na one koje nam mogu biti korisne u istraživanju. Podatke treba očistiti od raznih anomalija, kao što su na primer nedostajući podaci koji mogu da utiču na donošenje odluka.

Postoje mnogo statističkih metoda koje se mogu upotrebiti za pripremu podataka, kao što su maksimalne i minimalne vrednosti, srednja vrednost, itd.

Pošto je ovaj proces veoma zahtevan on traje veoma dugo vremena, u nekim slučajevima i do 50% trajanja celog istraživanja podataka. Što bolje pripremimo podatke time skraćujemo vreme izračunavanja u istraživanju podataka.

Generalno gledano priprema podataka nam osigurava poboljšanje donetih odluka.

d) Modeliranje

Modeli su primenjeni algoritmi za pretragu, identifikovanje i prikazivanje obrazaca u našim podacima. Postoje dve vrste modela

- Modeli koji klasifikuju
- Modeli koji predviđaju

Modeli su tačka gde istraživanje podataka prelazi iz pripreme i razumevanja u sferu razvoja i tumačenja.

e) Procena

Sve analize podataka imaju potencijal da daje netačne rezultate. Bez obzira na to da li je model upozorio na netačne rezultate, naše istraživanje ne mora dati neki obrazac iz naših podataka. Ovo se može desiti ukoliko naš model nije dobro podešen da nađe obrasce, ili možda koristimo pogrešan model. Moguće je da u našim podacima stvarno nema nikavih zanimljivih obrazaca. Zbog toga nam ovaj korak istraživanja podataka pomaže u odluci koliko je pouzdan primenjen model.

Procenu možemo izvršiti koristeći brojne matematičke i logičke tehnike. Pored ovih tehnika procena mora uključiti i ljudski faktor. Pošto stručnjaci mogu na osnovu svog iskustva proceniti uspešnost modela.

f) Razvoj

Ovo je korak koji obuhvata automatizovanje modela, predstavljanje modela korisnicima, integraciju sa postojećim informacionim sistemom. Zatim se mere performanse i proveravaju rezultati modela.

1.4. Modeli istraživanja podataka

Modeli istraživanja podataka nam služe da pronalazimo obrasce u podacima koje istražujemo. Modele možemo podeliti u dve grupe: opisne i predskazujuće. Opisni modeli generalno opisuju karakteristike podataka u bazi podataka, dok predskazujući modeli daju predviđanja na osnovu podataka iz baze koja se istražuje.

U nekim slučajevima korisnici nemaju predstavu kakvi obrasci mogu da se jave u njihovim podacima i zbog toga bi oni hteli da istražuju podatke na više načina. Zbog toga je važno da sistem za istraživanje podataka ima mogućnost da vrši istraživanje na osnovu više modela. Sistem za istraživanje treba da omogući korisniku da se „igra“ sa podacima, da dodaje ili izbacuje dimenzije ili attribute, da radi pivote, itd. Sve ove operacije omogućuju da se podaci sagledaju sa svih strana i tako se podrobno istraže svi mogući obrasci koji se javljaju u njima.

Neki od najviše korišćenih modela za istraživanja podataka su [1]:

- Regresija (regression)
- Klasifikacija (Classification and prediction)
- Klasterovanje (Clustering)
- Predviđanje (Prediction)
- Analiza serija (Time series analysis)
- Sumiranje (Summarization)
- Asocijativna pravila (Association rules)

1.4.1. Regresija

Ovaj model koristimo za predviđanje vrednosti ciljnih atributa. Metod je zasnovan na ideji da se podaci uklapaju u neku vrstu funkcija (linearne, eksponencijalne, itd). Tako da sa ovom metodom pronalazimo funkciju koja najbolje opisuje podatke koji se istražuju.

Neki od modela koji se koriste su na primer linearna regresija i nelinearna regresija.

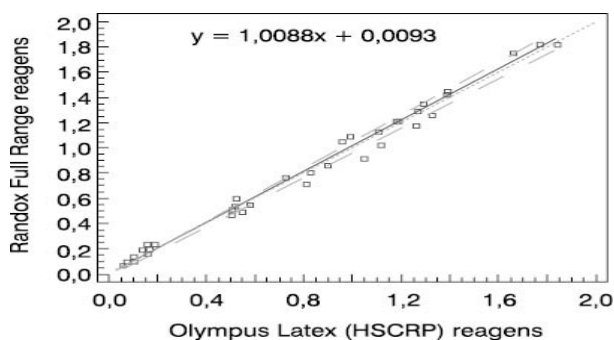
- Jednostavna linearna regresija predstavlja odnos između dve pojave i to takav da promenu jedne pojave prati približno linearna promena druge [2]: $Y = f(X) + e$

Y = deterministička komponenta + slučajna greška

Deterministički deo modela [2] : $f(X) = a + bX$

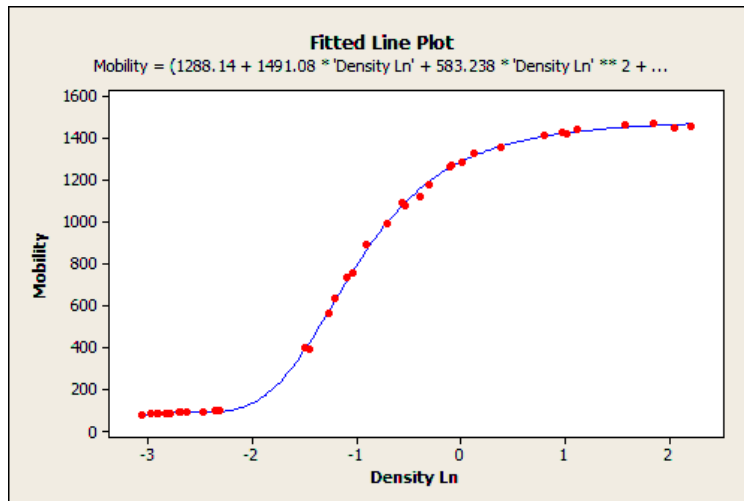
Y – zavisna varijabla, vrednost pojave čije se varijacije objašnjavaju modelom

X – nezavisna varijabla, stvarne vrdenosti pojave kojom se objašnjavaju varijacije zavisne varijable [2].



Slika 2 Primer lineane regresija [3]

- Ako se podaci ne mogu prikazati običnom linearnom funkcijom, onda možemo reći da se radi o nelinearnoj regresiji, i ona se prikazuje grafikom koji nije linearan.

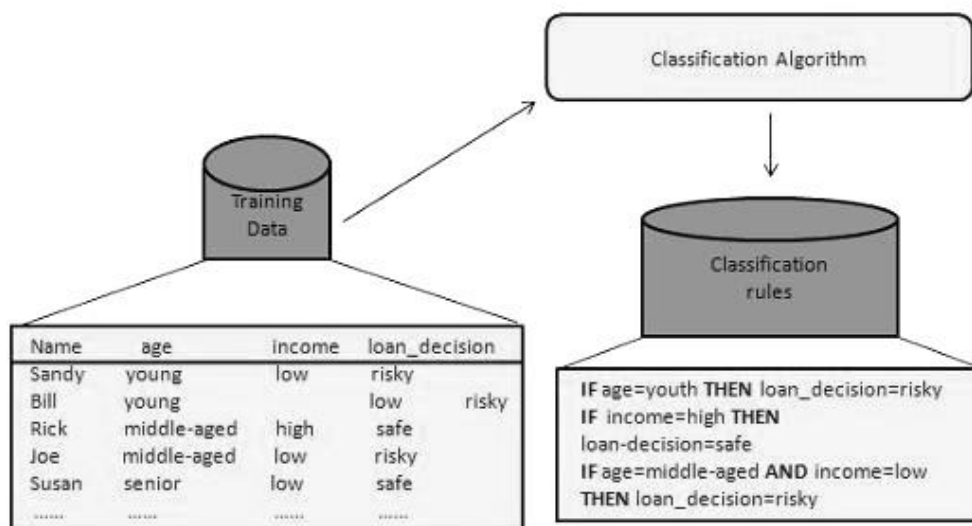


Slika 3 Primer nelinearne regresije [4]

1.4.2. Klasifikacija

Klasifikacija mapira podatke u unapred definisane grupe(classes). Ova tehnika se često naziva kao nadgledano učenje, zato što su klase unapred određene.

Model je obično predstavljen u formi pravila, stabla odlučivanja ili matematičkih formula.

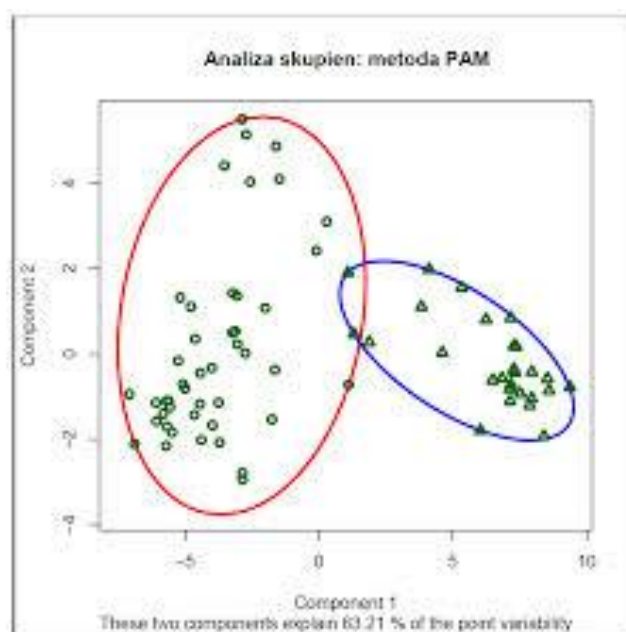


Slika 4 Primer klasifikacije [5]

1.4.3. Klasterovanje

Klasterovanje je veoma slično klasifikaciji jedino što grupe nisu unapred definisane. Grupe u koje će se mapirati podaci se određuju na osnovu podataka. Ovaj vid modeliranja se obično naziva nenadgledano učenje.

Grupisanje se postiže utvrđivanjem sličnosti između podataka koje istražujemo. Najslbližiji podaci se grupišu u klaster. Pošto klasteri nisu unapred određeni, obično eksperti za istraživanje podataka moraju dodatno da istraže šta je značenje kreiranih klastera.



Slika 5 Primer klastera [6]

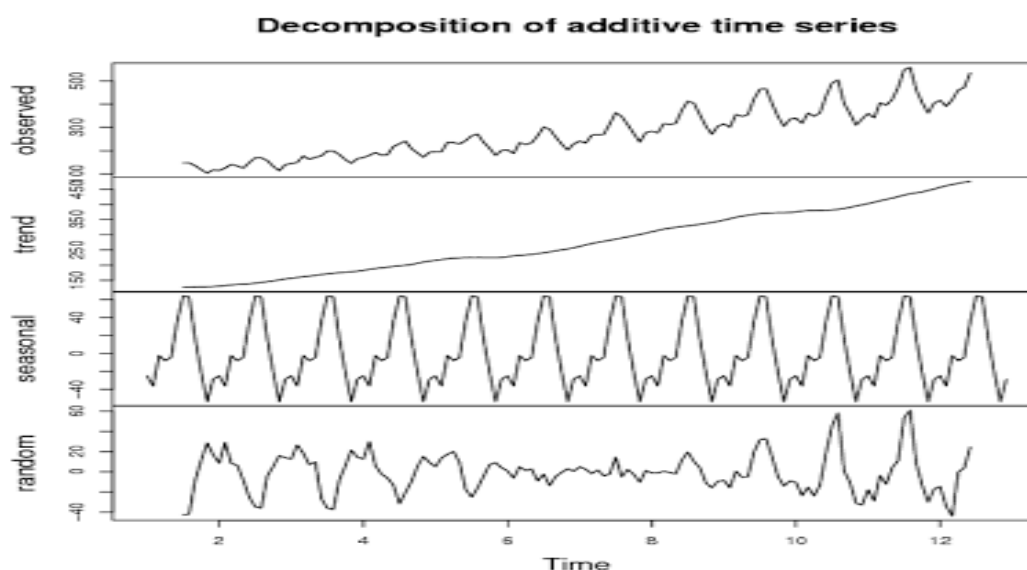
1.4.4. Predviđanje

Za razliku od ostalih modela istraživanja podataka, ovaj model nam daje predviđanje budućih stanja. Obično se koristi za prepoznavanje govora, prepoznavanje oblika, mašinsko učenje.

1.4.5. Analiza serija tokom vremena

Sa ovim modelom se ispituje promena vrednosti atributa tokom vremena. Obično se prikazuje grafički, gde se lako mogu uočiti promene. Postoje tri osnovne funkcije koje se izvršavaju tokom analize.

- ⇒ Mere rastojanja se koriste za utvrđivanje sličnosti između različitih serija
- ⇒ Struktura linije se ispituje da bi se utvrdilo ponašanje različitih serija
- ⇒ Koristi se istorijsko vreme da se predvide buduće vrednosti



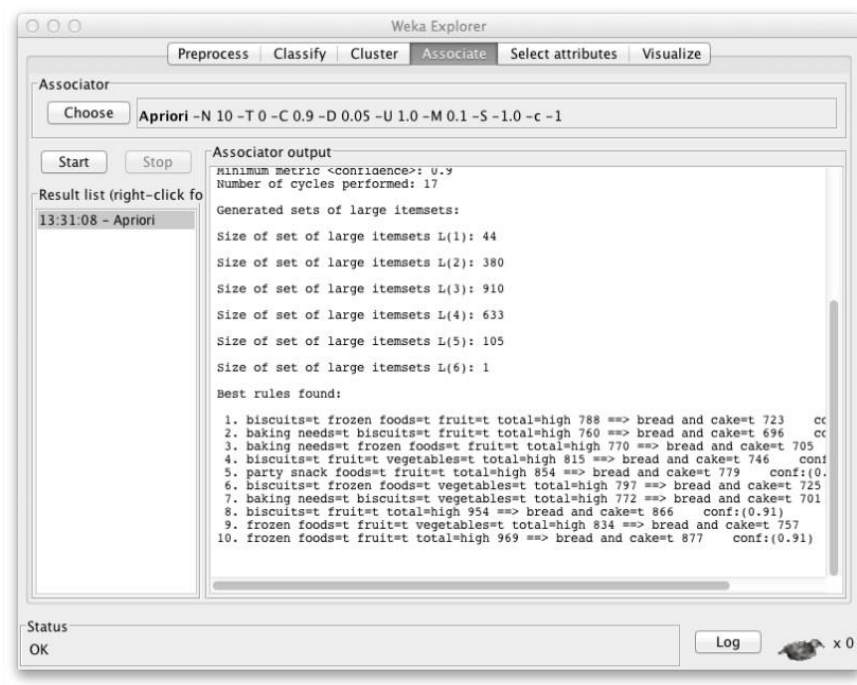
Slika 6 Primer time series [7]

1.4.6. Sumiranje

Ovaj metod mapira podatke u podskupove kojima pridružuje jednostavne opise. Obično se naziva i metod karakterizacije ili generalizacije. Možemo reći da izvlači reprezentativne informacije iz baze podataka. Ovo se može postići preuzimanje samo delova podataka, koji se posle mogu statistički obraditi.

1.4.7. Asocijativna pravila

Ovim metodom se otkrivaju veze između podataka. Najčešće se koristi za analizu podataka u marketingu. Na primer za otkrivanje proizvoda koji se zajedno prodaju.



Slika 7 Primer Association rule [8]

1.5. Čišćenje podataka

Kvalitet velikog skupa podataka iz realnog sveta zavisi od mnogo faktora, a jedan od najznačajniji je izvor odakle dolaze ti podaci. Unos podataka i sakupljanje podataka mogu dovesti do jednostavnih ali i kompleksnih grešaka. Nastanak ovih grešaka može da se smanji kontrolom pri unosu podataka, ali se one ne mogu u potpunosti isključiti.

Zato je potrebno da se postojeći podaci prečiste na neki način, da bi se smanjio obim tih grešaka. Pošto je radi o velikim skupovima podataka, ovaj proces može trajati veoma dugo. Ručno ispravljanje grešaka je veoma dugotrajno i sklono greškama. Zbog toga je potrebno da se koriste razni alati koji automatizuju proces čišćenja podataka.

Čišćenje podataka možemo definisati kao proces eliminacije grešaka i nekonzistentnosti u podacima [9].

Proces čišćenja podataka možemo u tri faze [10]:

- Definisane i određivanje tipa greške
- Pretraga i identifikacija grešaka
- Ispravljanje grešaka

Postoji mnogo metoda za određivanje grešaka, na primer [10]:

- Statističke metode određuju okvirne vrednosti podataka, kao što su standardna devijacija, aritmetička sredina. Ako neki podatak iskače iz granica, može se smatrati greškom. Ova metoda je brza i jednostavna, ali i sama može napraviti greške. Zbog toga se obično koristi sa nekom drugom metodom.
- Klasterovanjem se podaci grupišu u grupe. Podaci koji se ne mogu svrstati u neku grupu se smatraju greškama. Glavna karakteristika ove metoda je da je veoma komplikovana za izračunavanje.
- Model zasnovan na obrascima svrstava podatke u unapred određene grupe. Ukoliko neki podatak ne možemo svrstati u grupu, onda se ona smatra greškom.

1.6. Rad sa podacima koji nedostaju

Većina velikih skupova podataka iz realnog sveta imaju podatke koji nedostaju. Postoji mnogo razloga zašto se javljaju podaci koji nedostaju. Neke vrednosti jednostavno nisu zabeležene, neki podaci su zabeleženi, ali su ubačeni nakon našeg istraživanja u bazu podataka, neki podaci su možda greškom izbrisani.

Problem podataka koji nedostaju je veoma bitan za istraživanje podataka, i za rešavanje tog problema postoji mnogo metoda, ovo su neke od njih [10]:

- Brisanje ili ignorisanje slučajeva kod kojih se javlja podatak koji nedostaje
- Dodeljivanje najčešće vrednosti umesto podataka koji nedostaju
- Dodeljivanje svih vrednosti atributa atributu kojem nedostaje podatak
- Dodeljivanje srednje vrednosti podatku koji nedostaje

2. Primena ID3 algoritma (Stablo odlučivanja) u istraživanju podataka i grafički prikaz formiranog stabla odlučivanja

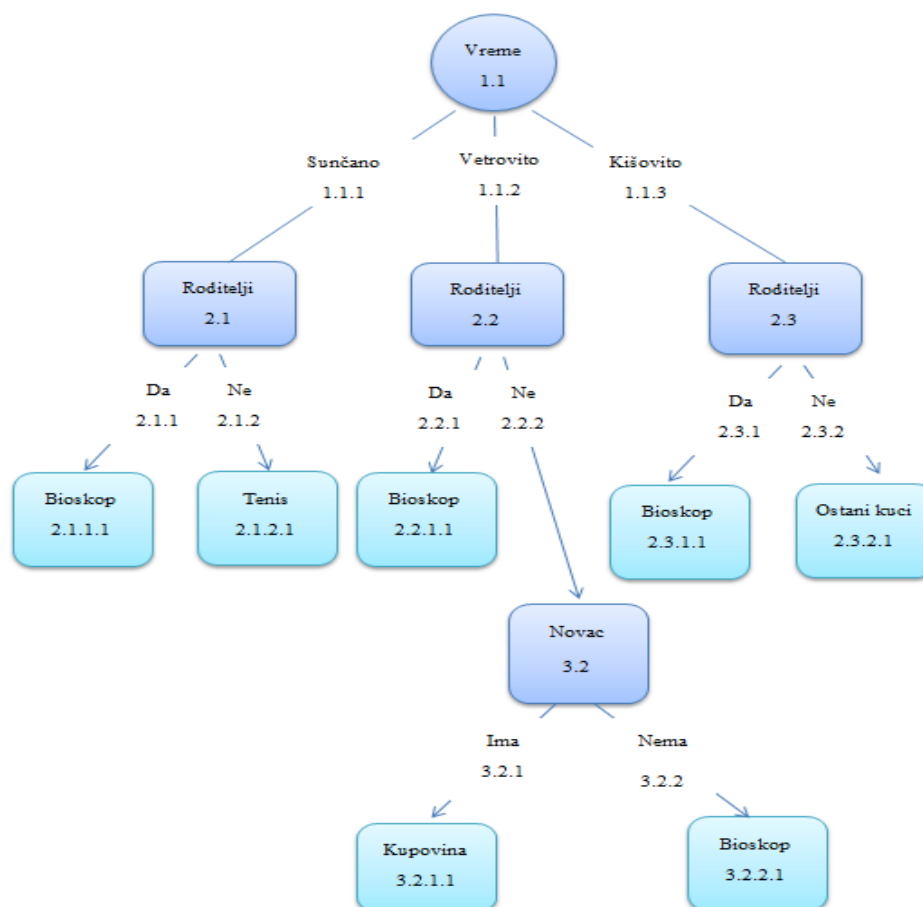
2.1. Uvod

Stablo odlučivanja je jedan od najčešće korišćenih metoda u istraživanju podataka. Kao rezultat ove metode se dobija jedna struktura u obliku stabla koja je lako razumljiva ljudima. Rezultat ove metode se takođe može predstaviti serijom if –then pravila kojom se povećava čitljivost ove metode. Stablo odlučivanja klasifikuje slučajeve sortirajući ih od korena (root) do krajnjih čvorova (leaf). Svaki čvor u stablu predstavlja neki atribut , a svaka grana koja izlazi iz čvora je određena koliko ima mogućih vrednosti za dati atribut.

Stablo odlučivanja ima sledeće elemente:

- Krajnji čvor (leaf node) – ovaj čvor predstavlja završetak grane. Ovim čvorom je definisan konačan odgovor za podatke sa te grane
- Čvor odluke (decision node) – definiše uslov za vrednost atributa
- Grana (branch) – povezuje čvorove

Izgled stabla odlučivanja:



Slika 8. Primer stabla odlučivanja

2.2. Metoda učenja

Većina algoritama za kreiranje stabla odlučivanja su varijacije osnovnog algoritma ID3 koji je J.Ross Quinlan predstavio 1986 godine.

Prvi korak u formiranju stabla odlučivanja je postavljanje pitanja “koji atribut treba da bude u korenu stabla?”

Kao odgovor na ovo pitanje, svaki atribut se statistički testira da bi se utvrdilo kako će on klasifikovati set podataka za učenje. Najbolji atribut se bira i postavlja za koren stabla odlučivanja. U zavisnosti od mogućih vrednosti ovog atributa kreiraju se grane koji predstavljaju skup podataka koji je ustvari filtriran početni skup podataka sa vrednošću te grane. Iz početnog seta podataka se više ne uzimaju u obzir podaci koji su bili iskorišćeni za čvor iz kojeg izlaze grane. Ovo znači da se oni samo jednom mogu pojaviti u određenoj grani stabla. Ovaj postupak se ponavlja rekurzivno za svaki čvor odlučivanja. Ovo se ponavlja dok svaki čvor ne zadovolji jedan od kriterijuma:

- svi atributi su već bili korišćeni u toj grani stabla, ili
- svi primeri koji pripadaju tom čvoru imaju istu klasu - prema tome radi se o krajnjem čvoru grane (entropija primera jednaka je nuli).

Podaci za kreiranje stabla odlučivanja treba da imaju formu:

- Ulazni podaci – na osnovu ovih podataka se vrši učenje
- Ciljni atribut – vrednost koje treba da se predvide stablom odlučivanja
- Ostali atributi - oni će biti testirani tokom učenja

Algoritam za kreiranje stabla odlučivanja izgleda ovako [11]:

Počni

Ako je S prazan, napravi jedan čvor s vrednošću Pogrešno;

Ako se S sastoji od primera s istom vrednošću ciljnog atributa, napravi jedan krajnji čvor s tom vrijednosti ciljnog atributa;

Ako je R prazan, tada napravi jedan (krajnji) čvor s vrednosti koja je najčešća za ciljni atribut za skup S; (u tom slučaju stablo će verovatno raditi i pogrešne klasifikacije na skupu S, u meri u kojoj su zastupljeni primeri ostalih klasa);

Neka je A atribut s najvećom vrednosti $\text{Gain}(A,S)$ između svih atributa u R;

Neka su $\{a_j | j=1,2, \dots, m\}$ vrednosti atributa A;

Neka su $\{S_j | j=1,2, \dots, m\}$ podskupovi S koji se sastoje od primera koji imaju a_j za atribut A ;

Napravi stablo s korenom označenim A , te granama $a_1, a_2, \dots, a_m$ koja vode na stabla

$(ID3(R-\{A\}, C, S_1), ID3(R-\{A\}, C, S_2), \dots, ID3(R-\{A\}, C, S_m))$;

Rekurzivno primeni ID3 to na podskupove $\{S_j | j=1,2, \dots, m\}$ sve dok oni nisu prazni;

Kraj

2.3. Entropija kao klasifikator

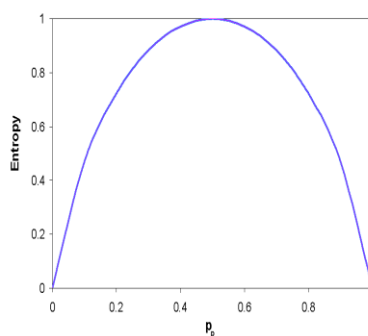
Glavno pitanje ID3 algoritma je izbor atributa koji će nam biti najbolji čvorovi odluke. Ovaj algoritam koristi statistički podatak nazvan informacioni dobitak (information gain) koji meri kako dobro određeni atribut razdvaja set podataka u zavisnosti od ciljnog atributa. ID3 koristi ovaj podatak za izbor kandidata za svaki korak u procesu formiranja stabla odlučivanja.

Da bi precizno definisali informacijski dobitak, potrebno je definisati meru, koja se često koristi u teoriji informacija, a naziva se entropija. Entropija predstavlja “čistoću” nekog skupa primera.

Definicija entropije je sledeća [11]:

$$\text{Entropija}(S) = - \sum_{i=1}^c p_i \log_2 p_i$$

U ovoj definiciji p_i označava verovatnoću primera u S , dok c označava broj različitih vrednosti ciljnog atributa. Za izračunavanje entropije uzima se da je $0 * \log 0 = 0$.



Slika 9. Entropija u slučaju binarnog klasifikacijskog problema, u ovisnosti o proporciji pozitivnih primera u skupu S (p_+ varira između 0 i 1) [11] .

2.4. Informacijski dobitak (Gain)

Uz datu entropiju kao meru 'nečistoće' u setu podataka, sada možemo definisati meru efektivnosti atributa u klasificiranju primera. Ta mera, informacijski dobitak, predstavlja očekivanu redukciju entropije uzrokovanu razdvajanjem primera na osnovu tog atributa. Informacijski dobitak, Gain (S,A), atributa A, u odnosu na skup primera S, definisan je kao [11]:

$$\text{Gain}(S,A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} * \text{Entropy}(S_v)$$

gde je Values(A) skup svih mogućih vrednosti atributa A, a S_v podskup od S, za koji atribut A ima vrednost v (t.j., $S_v = \{s \in S \mid A(s) = v\}$). Prvi član u jednačini za Gain je entropija originalnog skupa S dok je drugi član očekivana vrednost entropije nakon što je S razdvojen korišćenjem atributa A. Očekivana entropija opisana drugim članom jednostavno je zbroj entropija podskupova S_v , s težinom proporcionalnom delu primera $|S_v| / |S|$ koji pripadaju S_v . Gain (S,A) je dakle očekivana redukcija entropije uzrokovana poznavanjem vrednosti atributa A. Ili, na drugi način, Gain(S,A) je informacija o vrijednosti ciljnog atributa, uz poznate vrednosti.

2.5. Podrezivanje stabla odlučivanja

Generisano stablo odlučivanja perfektно klasifikuje sve primere iz seta podataka za učenje, osim ako se desi slučaj da nema atributa sa pozitivnim informacionim dobitkom. Takođe za neki nedosledan set podataka formiraće se stablo odlučivanja sa najmanje mogućih grešaka, ukoliko se opet ne desi slučaj da nema nijedan atribut sa pozitivnim informacionim dobitkom.

U praksi dobijeno stablo odlučivanja koje perfektно klasifikuje sve primere iz seta podataka za učenje baš i nije najbolje rešenje. Savršeno stablo odlučivanja formirano na setu podataka za učenje može dovesti do pogoršanja rezultata u setu podataka koje koristimo za testiranje. Ovaj slučaj se naziva „overfitting“.

Teoretski gledano jednostavnija stabla odlučivanja koja imaju neke greške a koje su formirane na osnovu seta podataka za učenje se više preporučuju od kompleksnih stabla odlučivanja koje nemaju nikakve greške.

Postoji nekoliko rešenja za izbegavanje „overfitting“-a kod stabla odlučivanja:

- Rast stabla odlučivanja se može zaustaviti pre nego se primeri iz seta podataka savršeno klasifikuju
- Stablo odlučivanja se može formirati do kraja a posle toga da se izvrši „skraćivanje“ grana po unapred određenom kriterijumu

Prvo rešenje se može učiniti boljim, ali se u praksi bolje pokazalo drugo rešenje. Ovo je zato što se ne može unapred predvideti kompleksnost stabla odlučivanja.

Kod rešavanja problema „overfitting“-a glavno pitanje je kako definisati idealnu kompleksnost. Najčešće se koriste sledeći pristupi:

- Koristi se poseban set podataka (drugačiji od onog na osnovu kojeg je formirano stablo odlučivanja) sa kojim se testira koliko je uspešno izvršeno “skraćivanje” stabla odlučivanja
- Na čvorovima koji su kandidati za “skraćivanje” vrši se poseban statistički test, koji nam može pokazati da li se postiže poboljšanje uklanjanjem tog čvora
- Definiše se mera kompleksnosti i kad se dostigne taj kriterijum rast stabla odlučivanja se zaustavlja. Ovaj pristup se zove “Minimum Description Length”.

Prvi od ovih pristupa je i najčešći. Set podataka se deli u dva seta podataka:

- Set podataka za učenje - na osnovu ovih podataka se formira stablo odlučivanja
- Set podataka za validaciju – ovaj set podataka služi za proveru efikasnosti “skraćivanja stabla”

2.6. Modifikacija ID3 algoritma

Kako je jedan od ciljeva ovog istraživanja grafički prikaz kreiranog stabla odlučivanja, došlo se do zaključka da bi se stablo odlučivanja moglo lakše grafički predstaviti, ako bi izračunati podaci bili predstavljeni u tri objekta, koji su nazvani Čvor, Grana i List.

Set podataka koji je korišćen u toku izrade a zatim i testiranju tačnosti stabla odlučivanja [12]:

Vreme	Roditelji	Novac	Odluka
Suncano	Da	Ima	Bioskop
Suncano	Ne	Ima	Tenis
Vetrovito	Da	Ima	Bioskop
Kisovito	Da	Nema	Bioskop
Kisovito	Ne	Ima	Ostani kuci
Kisovito	Da	Nema	Bioskop
Vetrovito	Ne	Nema	Bioskop
Vetrovito	Ne	Ima	Soping
Vetrovito	Da	Ima	Cinema
Suncano	Ne	Ima	Tenis

Opis modifikovanog algoritam za kreiranje stabla odlučivanja je sledeći:

1. Na osnovu kompletnog seta podataka izračunati Informatički dobitak (Gain) i Entropiju

Entropija se računa na sledeći način:

Prvo je potrebno da se izračuna koliko svaki ciljni atribut ima pojavljivanja u setu podataka koji ćemo obrađivati.

Odluka	Broj pojavljivanja
Bioskop	6
Tenis	2
Soping	1
Ostani kuci	1
Ukupan broj atributa	10

$$=-(6/10)*\text{LOG}((6/10);2)-(2/10)*\text{LOG}((2/10);2)-(1/10)*\text{LOG}((1/10);2)-(1/10)*\text{LOG}((1/10);2)$$

Entropija =1,570951

Zatim računamo entropiju za svaki atribut na sledeći način:

Atribut – Vreme

Opet računamo koliki je ukupan broj atributa, a zatim se računa koliko ih ima zbirno

Entropija (Suncano)

Vreme	Roditelji	Novac	Odluka
Suncano	Da	Ima	Bioskop
Suncano	Ne	Ima	Tenis
Suncano	Ne	Ima	Tenis
Ukupno			3
Entropija(Suncano)			
$=-(1/3)*\text{LOG}((1/3);2)-(2/3)*\text{LOG}((2/3);2)$			
0,918296			

Entropija(Kisovito)

Vreme	Roditelji	Novac	Odluka
Kisovito	Da	Nema	Bioskop
Kisovito	Ne	Ima	Ostani kuci
Kisovito	Da	Nema	Bioskop

Ukupno 3

Entropy(Kisovito)

$$=-(2/3)*\text{LOG}((2/3);2)-(1/3)*\text{LOG}((1/3);2)$$

0,918296

Entropija(Vetrovito)

Vreme	Roditelji	Novac	Odluka
Vetrovito	Da	Ima	Bioskop
Vetrovito	Ne	Nema	Bioskop
Vetrovito	Ne	Ima	Soping
Vetrovito	Da	Ima	Bioskop

Entropy(Vetrovito)

$$=-(3/4)*\text{LOG}((3/4);2)-(1/4)*\text{LOG}((1/4);2)$$

0,811278

Informatički dobitak(Gain) za atribut Vreme

Gain(S,vreme)

$$=1,571 - (3/10)*\text{Entropy}(\text{Sun})-(4/10)*\text{Entropy}(\text{Wind})-(3/10)*\text{Entropy}(\text{Rain})$$

0,695511

Postupak ponavljamo za sve attribute.

Atribut – Roditelji

Entropija(Da)

Vreme	Roditelji	Novac	Odluka
Suncano	Da	Ima	Bioskop
Vetrovito	Da	Ima	Bioskop
Kisovito	Da	Nema	Bioskop
Kisovito	Da	Nema	Bioskop
Vetrovito	Da	Ima	Bioskop

Entropija(Da)

$$=-(5/5)*\text{LOG}((5/5);2)-(5/10)*\text{LOG}((5/10);2)$$

0

Entropija(Ne)

Vreme	Roditelji	Novac	Odluka
Suncano	Ne	Ima	Tenis
Kisovito	Ne	Ima	Ostani kuci
Vetrovito	Ne	Nema	Bioskop
Vetrovito	Ne	Ima	Soping
Suncano	Ne	Ima	Tenis

Entropy(Ne)

$$=-(2/5)*\text{LOG}((2/5);2)-(1/5)*\text{LOG}((1/5);2)-(1/5)*\text{LOG}((1/5);2)-(1/5)*\text{LOG}((1/5);2)$$

1,921928

Informatički dobitak(Gain) za atribut Roditelji

Gain(S,Roditelji)

$$=1,571-(5/10)*\text{Entropy}(\text{Da})-(5/10)*\text{Entropy}(\text{Ne})$$

0,610036

Atribut – Novac

Entropija(Ima)

Weather	Parents	Money	Decision
Suncano	Da	Ima	Bioskop
Suncano	Ne	Ima	Tenis
Vetrovito	Da	Ima	Bioskop
Kisovito	Ne	Ima	Ostani kuci
Vetrovito	Ne	Ima	Soping
Vetrovito	Da	Ima	Bioskop
Suncano	Ne	Ima	Tenis

Entropy(Ima)

$$=-(3/7)*\text{Log}((3/7);2)-(2/7)*\text{Log}((2/7);2)-(1/7)*\text{Log}((1/7);2)-(1/7)*\text{Log}((1/7);2)$$

1,842371

Entropija(Nema)

Vreme	Roditelji	Novac	Odluka
Kisovito	Da	Nema	Bioskop
Kisovito	Da	Nema	Bioskop
Vetrovito	Ne	Nema	Bioskop

Entropy(Nema)

$$=(3/3)*\text{Log}(3/3;2)$$

0

Informatički dobitak za atribut Novac

```
Gain(S,Novac)
=1,571-(3/10)*Entropy(Nema)-(7/10)*Entropy(Ima)
0,28134
```

Na osnovu ovih izračunavanja dolazimo do zaključka da je najbolji kandidat Vreme pošto ima najveći Informatički dobitak. On će nam predstavljati prvi čvor i on će se nalaziti na prvom nivou stabla odlučivanja.



Slika 10. Prvi nivo stabla odlučivanja

U objekat Cvor upisujemo sledeće vrednosti:

Nivo = 1

Cvor = 1

Naziv = Vreme

Gain = 0,695511

2. Nakon izbora čvora u prvom nivou grupisaćemo vrednosti koje postoje za taj atribut

Vreme

Suncano

Vetrovito

Kisovito

Ove vrednosti nam predstavljaju grane koje izlaze iz čvora na prvom nivou.

U objekat Grana upisujemo sledeće vrednosti.

Nivo	Cvor	Grana	Naziv	Gain
1	1	1	Suncano	0,918
1	1	2	Vetrovito	0,811
1	1	3	Kisovito	0,918

Ovako izgleda prvi nivo stabla odlučivanja sa granama.



Slika 11. Prvi nivo stabla odlučivanja sa granama

3. Sledeći korak je kreiranje novog seta podataka za svaku granu i uklanjanje atributa koji je već iskorišćen- a to je Vreme

Novi set podataka za granu - Suncano

Vreme	Roditelji	Novac	Odluka
Suncano	Da	Ima	Bioskop
Suncano	Ne	Ima	Tenis
Suncano	Ne	Ima	Tenis

Nakon uklanjanja atributa **Vreme**, ostaje nam set podataka koji ćemo koristiti za testiranje ove grane

Roditelji	Novac	Odluka
Da	Ima	Bioskop
Ne	Ima	Tenis
Ne	Ima	Tenis

Ovo ponavljamo za sve tri grane pa dobijamo setove podataka sa kojima ćemo nastaviti izračunavanje stabla odlučivanja.

Grana - Suncano

Roditelji	Novac	Odluka
Da	Ima	Bioskop
Ne	Ima	Tenis
Ne	Ima	Tenis

Grana - Vetrovito

Roditelji	Novac	Odluka
Da	Ima	Bioskop
Ne	Nema	Bioskop
Ne	Ima	Soping
Da	Ima	Bioskop

Grana – Kisovito

Roditelji	Novac	Odluka
Da	Nema	Bioskop
Ne	Ima	Ostani kuci
Da	Nema	Bioskop

4. Za svaku granu na osnovu novog seta podataka računamo informatički dobitak. Takođe testiramo da li neki od ovih setova podataka zadovoljava uslov da jednoznačno određuje ciljni atribut. Atributi koji ima najveći informatički dobitak postaju novi čvorovi, a ukoliko je zadovoljen uslov da je ciljni atribut jednoznačno određen, onda postavljamo list sa vrednošću ciljnog atributa.

Primer obračuna za granu Suncano

Roditelji	Novac	Odluka
Da	Ima	Bioskop
Ne	Ima	Tenis
Ne	Ima	Tenis

Entropija za granu Suncano

```
Entropy(Suncano)
=-(1/3)*Log((1/3);2)-(2/3)*Log((2/3);2)
0,918296
```

$$\text{Entropy}(\text{Roditelji-Da})$$
$$=(1/1)*\text{Log}((1/1);2)$$

0

$$\text{Entropy}(\text{Roditelji-Ne})$$
$$=(2/2)*\text{Log}((2/2);2)$$

0

$$\text{Entropy}(\text{Novac-Ima})$$
$$=-(1/3)*\text{Log}((1/3);2)-(2/3)*\text{Log}((2/3);2)$$

0,918296

$$\text{Entropy}(\text{Novac-Nema})$$

0

Informatički dobitak za granu Suncano, atribut Roditelji

$$\text{Gain}(\text{Suncano,Roditelji})$$
$$=0,918-(1/3)*\text{Entropy}(\text{Da})-(2/3)*\text{Entropy}(\text{Ne})$$

0,918

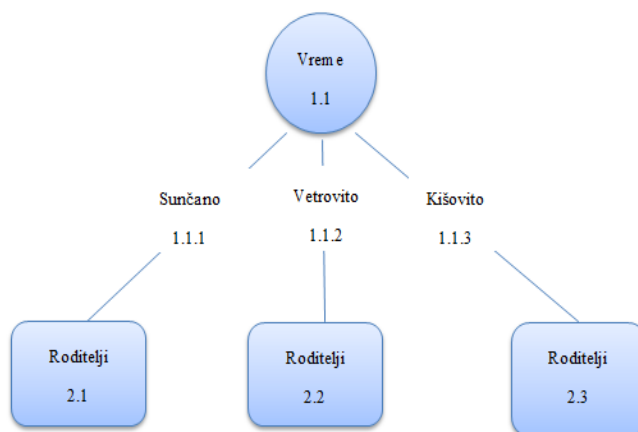
Postupak ponavljamo za granu Suncano, atribut Novac i dobijamo Informatički dobitak za granu Suncano, atribut Novac

$$\text{Gain}(\text{Suncano,Novac})$$
$$=0,918296-(3/3)*\text{Entropy}(\text{Ima})-(0/3)*\text{Entropy}(\text{Nema})$$

0,00

Iz obračuna nam sledi da je najbolji kandidat atribut Roditelji pošto ima najveći informatički dobitak.

Ovo ponavljamo za sve tri grane i nakon obračuna dobijamo čvorove drugog nivoa stabla odlučivanja.

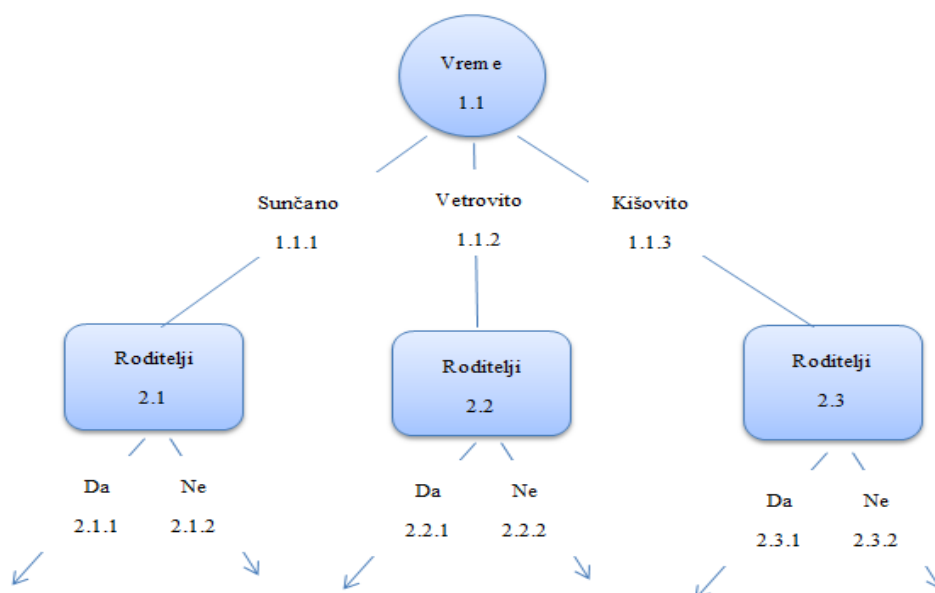


Slika 12. Prvi i drugi nivo stabla odlučivanja

U objekat Cvor dodajemo

Nivo	Cvor	Naziv
2	1	Roditelji
2	2	Roditelji
2	3	Roditelji

5. Za svaki čvor se određuju moguće vrednosti i dodaju se grane na svaki postojeći čvor i dobijamo drugi nivo stabla odlučivanja sa granama



Slika 13. Prvi i drugi nivo stabla odlučivanja sa granama

U objekat Grana upisujemo sledeće vrednosti.

Grana			
Nivo	Cvor	Grana	Naziv
2	1	1	Yes
2	1	2	No
2	2	1	Yes
2	2	2	No
2	3	1	Yes
2	3	2	No

6. Postojeće setove podataka po granama dalje filtriramo i uklanjamo attribute koji su iskorišćeni.

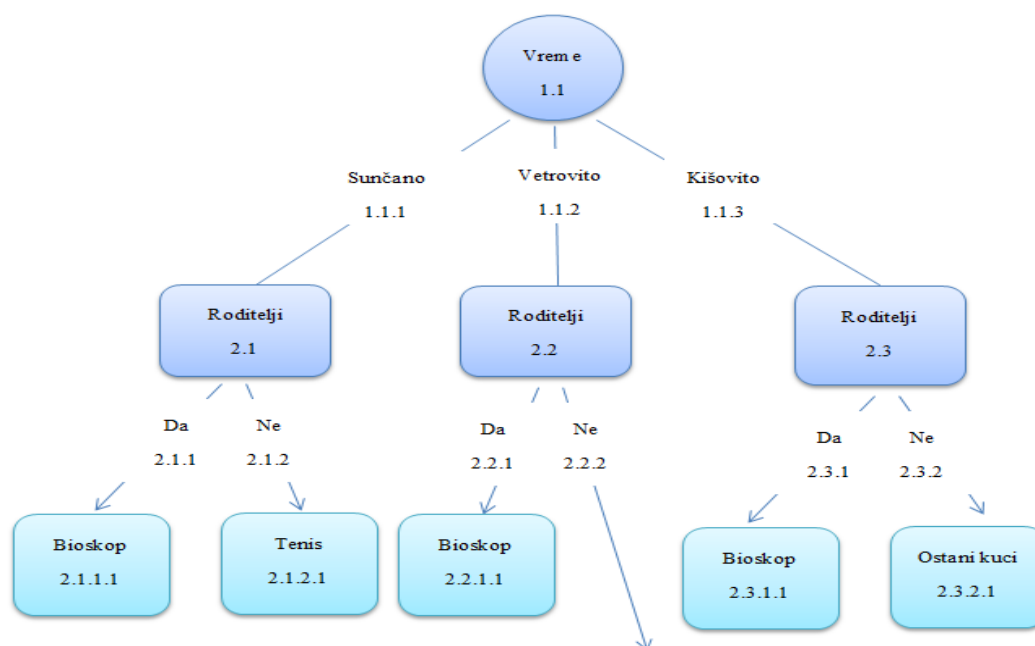
Grana – Suncano-Roditelji

Roditelji	Novac	Odluka
Da	Ima	Bioskop

Roditelji	Novac	Odluka
Ne	Ima	Tenis
Ne	Ima	Tenis

Za ovako dobijene setove podataka računamo informatički dobitak i testiramo da li su jednoznačno određeni ciljni atributi.

Iz ovog primera vidimo da je za atribut Roditelji sa vrednošću Da, grane Suncano-Roditelji ispunjen uslov da je ciljni atribut jednoznačno ispunjen. Tako da na kraj ove grane postavljamo list Bioskop. Ovaj postupak ponavljamo za sve grane i dobijamo sledeći izgled stabla odlučivanja



Slika 14. Prva tri nivoa stabla odlučivanja

U objekat List dodajemo:

Nivo	Cvor	Grana	List	Naziv
2	1	1	1	Bioskop
2	1	2	1	Tenis
2	2	1	1	Bioskop
2	3	1	1	Bioskop
2	3	2	1	Ostani kuci

Vidimo da je za granu Roditelji sa vrednošću Ne nije jednoznačno određen ciljni atribut, pa je potrebno za njega nastaviti izračunavanje. Svi ovakvi slučajevi se pamte u niz DT_Array, koji je ustvari niz definisan kao niz DataSetova.

Novac	Odluka
Nema	Bioskop
Ima	Soping

- Ispituje se da li je niz DT_Array prazan. Ako jeste izračunavanje stabla odlučivanja je završeno, inače se za svaki Dataset niza DT_Array izračunava informatički dobitak i određuje novi čvor.

U ovom slučaju se čvor postaje atribut Novac, koji ima dve grane.

U objekat Grana dodajemo:

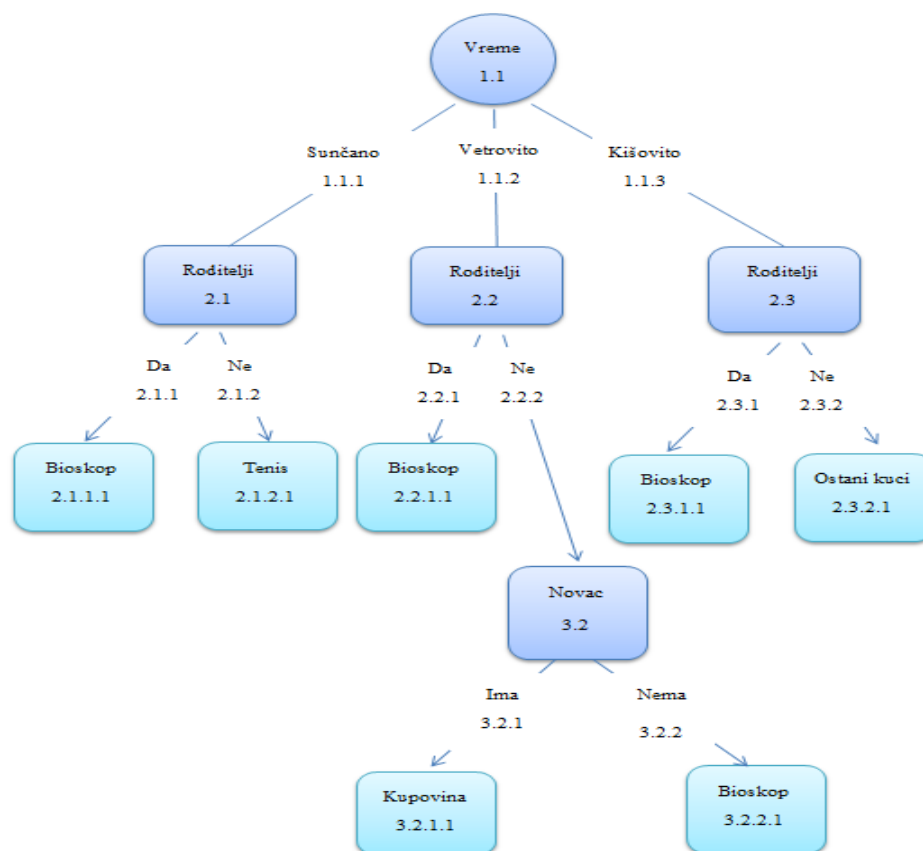
Nivo	Cvor	Grana	Naziv
3	1	1	Ima
3	1	2	Nema

8. Za svaku dobijenu granu se opet vrši filtriranje seta podatka i uklanjanje iskorišćenog atributa. Vrši se obračun informatičkog dobitka i testiranje da li je ciljni atribut jednoznačno određen.

9. Ispitivanje niza DT_Array se vrši dok ne dobijemo prazan niz i završava se formiranje stabla odlučivanja.

10.

Konačan izgled stabla odlučivanja izgleda ovako



Slika 15. Konačan izgled stabla odlučivanja

Podaci koji su upamćeni u objektima su sledeći:

Cvor		
Nivo	Cvor	Naziv
1	1	Vreme
2	1	Roditelji
2	2	Roditelji
2	3	Roditelji
3	1	Novac

Grana			
Nivo	Cvor	Grana	Naziv
1	1	1	Suncano
1	1	2	Vetrovito
1	1	3	Kisovito
2	1	1	Da
2	1	2	Ne
2	2	1	Da
2	2	2	Ne
2	3	1	Da
2	3	2	Ne
3	1	1	Ima
3	1	2	Nema

List				
Nivo	Cvor	Grana	List	Naziv
2	1	1	1	Bioskop
2	1	2	1	Tenis
2	2	1	1	Bioskop
2	3	1	1	Bioskop
2	3	2	1	Ostani kuci
3	1	1	1	Soping
3	1	2	1	Bioskop

2.7. Grafički prikaz izračunatog stabla odlučivanja

Jedan od ciljeva ovog istraživanja je bio da se formirano stablo odlučivanja grafički prikaže na što jednostavniji način, izbegavajući postojeće Treeview i slične kontrole koje postoje u Visual Studio 2012.

Modifikovani ID3 algoritam formira takvu strukturu podataka da je veoma lako nacrtati stablo odlučivanja koristeći standardne grafičke funkcije C# jezika, kao što su FillEllipse i DrawLine.

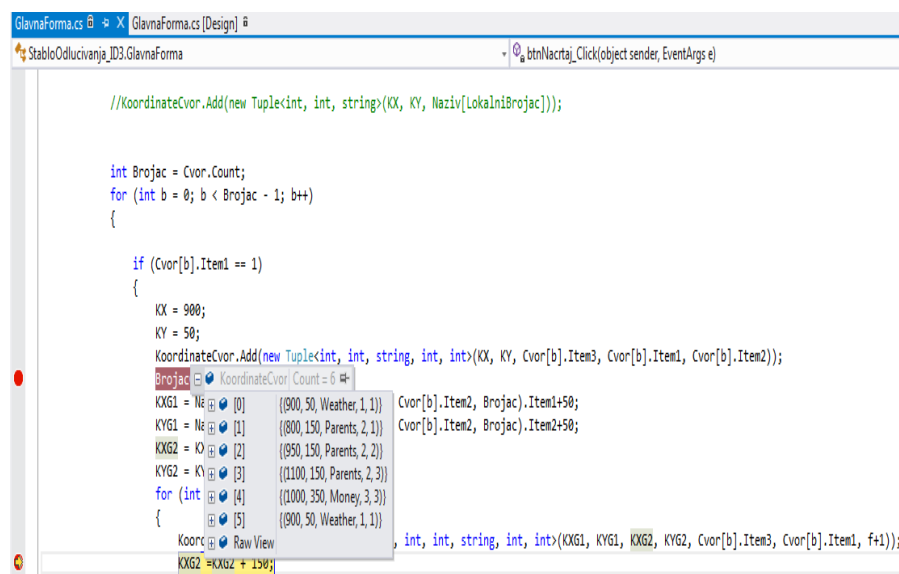
Čvorove ćemo predstaviti elipsama koje su obojene sa crvenom bojom, grane sa linijama crvene boje, dok će listovi biti predstavljeni sa kvadratima. U čvorove i listove se upisuju nazivi atributa i ciljnih atributa.

Algoritam za crtanje stabla odlučivanja je sledeći:

1. Glavna petlja je ispitivanje formiranih čvorova koji su zapamćeni u objektu Cvor.

Pretražuje se prvi nivo čvorova. Pošto je na prvom nivou čvorova jedan čvor onda se postavlja početna koordinata prvog čvora.

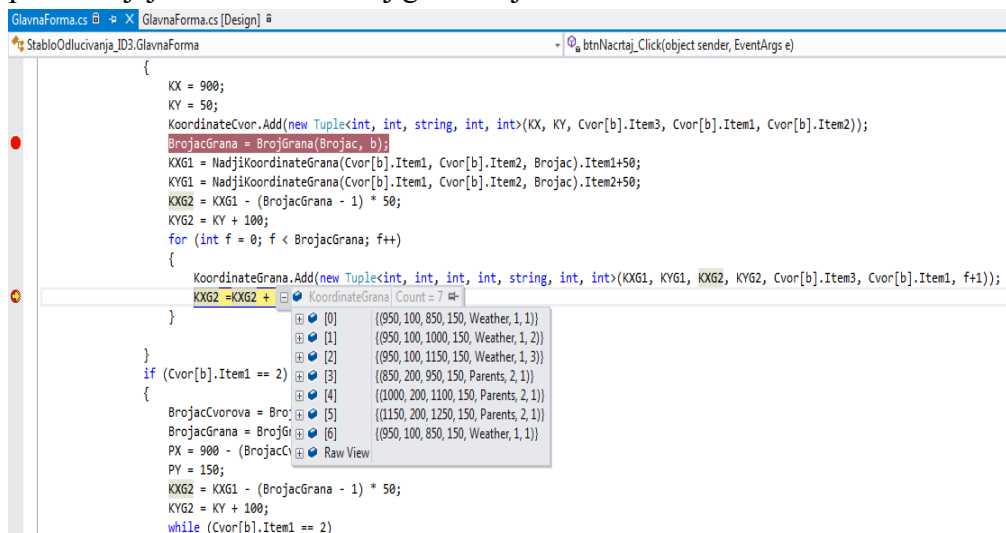
Za potrebe crtanja formiran je objekat KoordinateCvor(X, Y, Naziv čvora, Nivo, Čvor). Gde su X i Y koordinate elipse koja se crta i koja predstavlja čvor. Naziv čvora se upisuje na sredinu nacrtane elipse, a Nivo i Čvor predstavljaju kojem nivou pripada čvor i broja čvora u tom nivou.



Slika 16. Podaci iz objekta KoordinateCvor

- Zatim se izračuna koliko ima grana taj prvi čvor i računaju se koordinate grana prema formuli : $KXG2 = KXG1 - (BrojacGrana - 1) * 50$.

Za crtanje grana je formiran objekat `KoordinateGrana(KXG1, KYG1, KXG2, KYG2, Naziv čvora, Nivo, Čvor)`. `KXG1, KYG1` su koordinate početka linije koja kreće iz čvora a `KXG2, KYG2` su koordinate kraja linije koja se završava u čvoru na sledećem nivou. Naziv čvora je naziv iz kojeg čvora kreću grane a Nivo i čvor predstavljaju nivo čvora i broj grane koja izlazi iz čvora.



Slika 17. Podaci iz objekta `KoordinateGrana`

- Kad se završi formiranje objekata za prvi nivo, prelazi se na drugi nivo. Pretražuje se koliko ima čvorova u drugom nivou. Izračunavaju se koordinate za svaki čvor, vodeći računa o tome koliko ima čvorova. Svaka grana se pamti u objektu `KoordinateCvor`. U ovom koraku se takođe ispituje da li postoji neki krajnji List za trenutni nivo. Ukoliko ima onda se pamti u objektu `KoordinateList(X,Y,Naziv lista)`
- Postavlja se petlja koja za svaki čvor računa koordinate grana i pamti ih u objektu `KoordinateGrana`.
- Ovaj postupak se ponavlja povećavajući nivoe za jedan, sve do poslednjeg nivoa
- Kad se pređu svi nivoi imamo formirane objekte `KoordinateCvor`, `KoordinateGrana` i `KoordinateList` u kojem su sačuvani svi podaci potrebni za crtanje stabla odlučivanja.
- Poslednji korak je petlja koja prolazi kroz objekte i jednostavnim grafičkim naredbama crta elipse i linije koje povlači iz objekata.

Konačan izgled stablo odlučivanja je sledeći:



Slika 18. Ekran sa nacrtanim stablom odlučivanja

2.8. Prednosti i nedostaci

Prednosti metode stabla odlučivanja su:

- Generisanje lako razumljivih modela, koje se mogu predstaviti ili grafički ili uz pomoć if – then pravila;
- Sposobnost korišćenja nominalnih i numeričkih atributa;
- Stabla odlučivanja su sposobna za obradu setova podataka koja imaju greške;
- Stabla odlučivanja su sposobna za obradu setova podataka kojima nedostaju podaci;
- Stabla odlučivanja se smatraju za metod koji ne zahteva parameter.

Slabe strane metode stabla odlučivanja su:

- Stabla odlučivanja koja imaju relativno mali broj primera za učenje modela sklona su greškama;
- Moguć je “overfitting” podataka i zbog toga zahteva skraćivanje grana (prunning)

- Teže obrađuje numeričke podatke
- Većina algoritama zahteva da ciljni atribut ima diskretne vrednosti

2.9. Rad sa aplikacijom za učenje uz pomoć modifikovanog ID 3 algoritma

Aplikacija koja je napravljena za testiranje modifikovanog ID3 algoritma je veoma jednostavna za korišćenje.

Početni ekran aplikacije ima sledeći izgled.

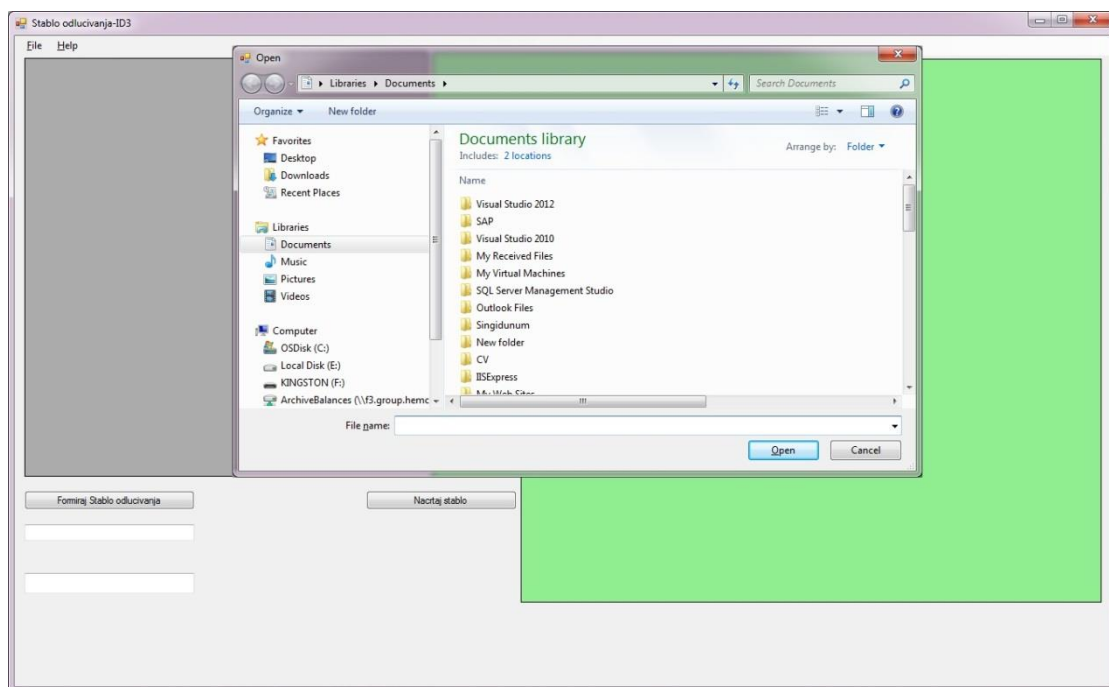


Slika.19. Početni ekran aplikacije

Podaci na osnovu kojih ćemo formirati stablo odlučivanja se učitavaju iz Excel fajlova, koje smo ranije pripremili [12].

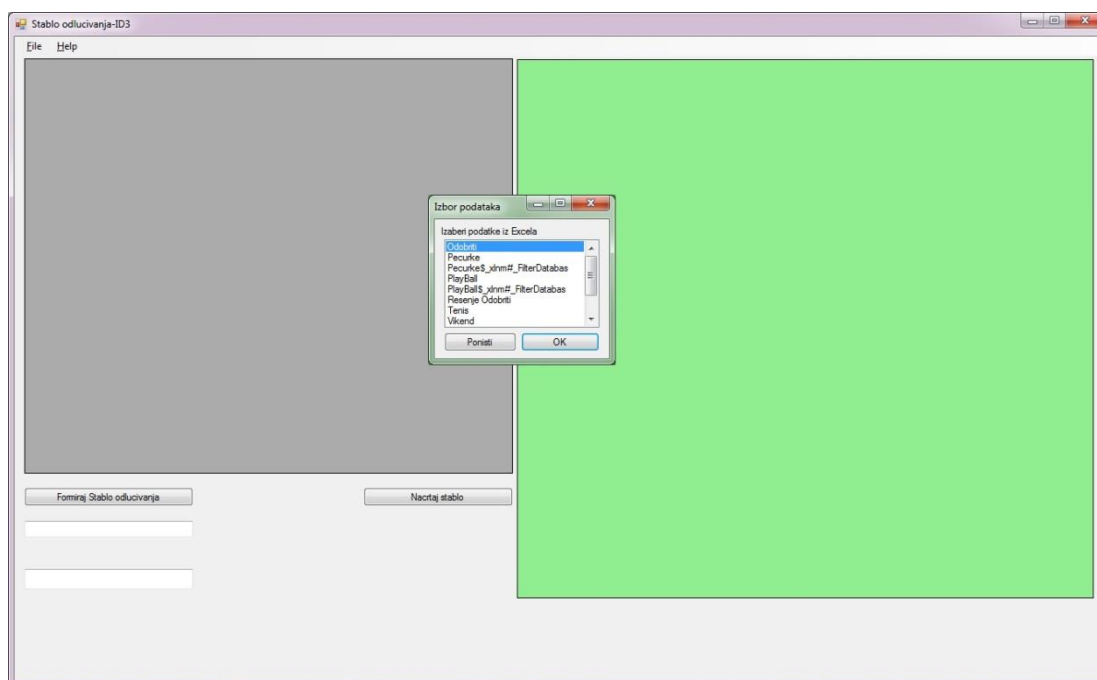
Vreme	Roditelji	Novac	Odluka
Suncano	Da	Ima	Bioskop
Suncano	Ne	Ima	Tenis
Vetrovito	Da	Ima	Bioskop
Kisovito	Da	Nema	Bioskop
Kisovito	Ne	Ima	Ostani kuci
Kisovito	Da	Nema	Bioskop
Vetrovito	Ne	Nema	Bioskop
Vetrovito	Ne	Ima	Soping
Vetrovito	Da	Ima	Bioskop
Suncano	Ne	Ima	Tenis

Izbor fajla se vrši opcijom File -> Open.



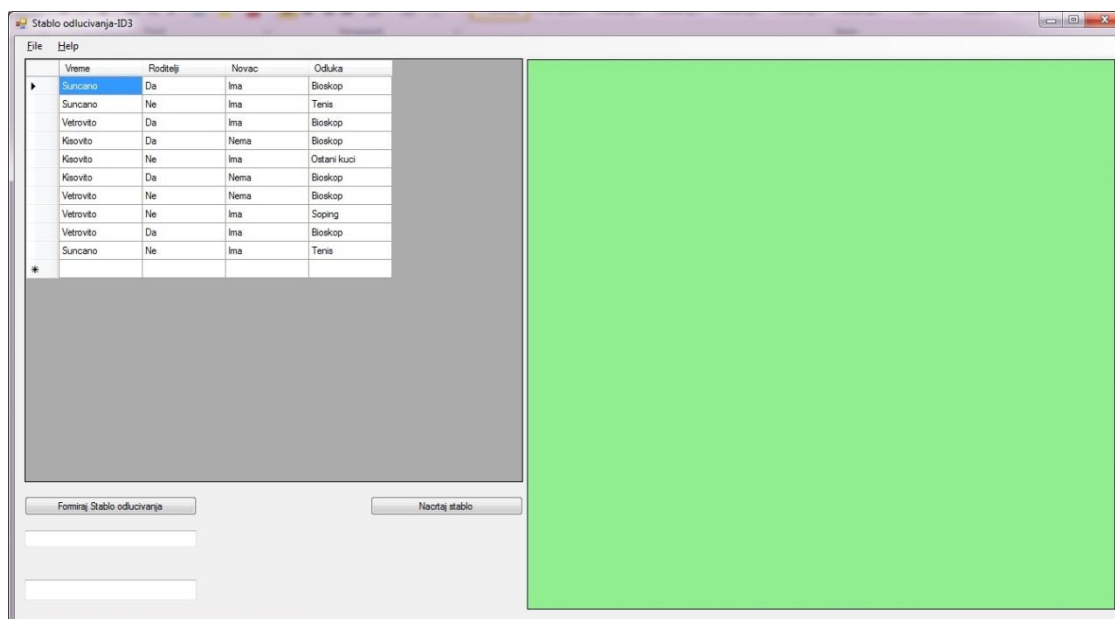
Slika 20. Učitavanje podataka u aplikaciju

Nakon izbora fajla u kojem je nalaze podaci koje želimo obraditi, potrebno je izabrati naziv sheet-a u Excel fajlu sa kojeg želimo učitati podatke.



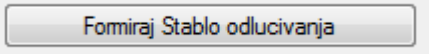
Slika 21. Izbor sheeta sa podacima iz Excel fajla

Podaci se zatim učitavaju u aplikaciju i pamte se u Datagrid objektu.

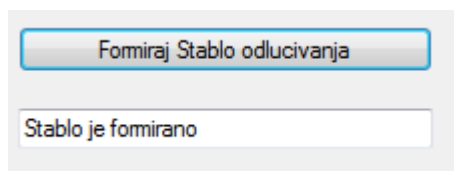


Slika 22. Učitani podaci iz Excel fajla

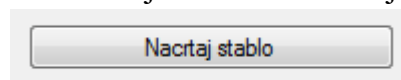
Kad smo učitali podatke spremni smo za formiranje stabla odlučivanja. Proces se započinje pritiskom na dugme.



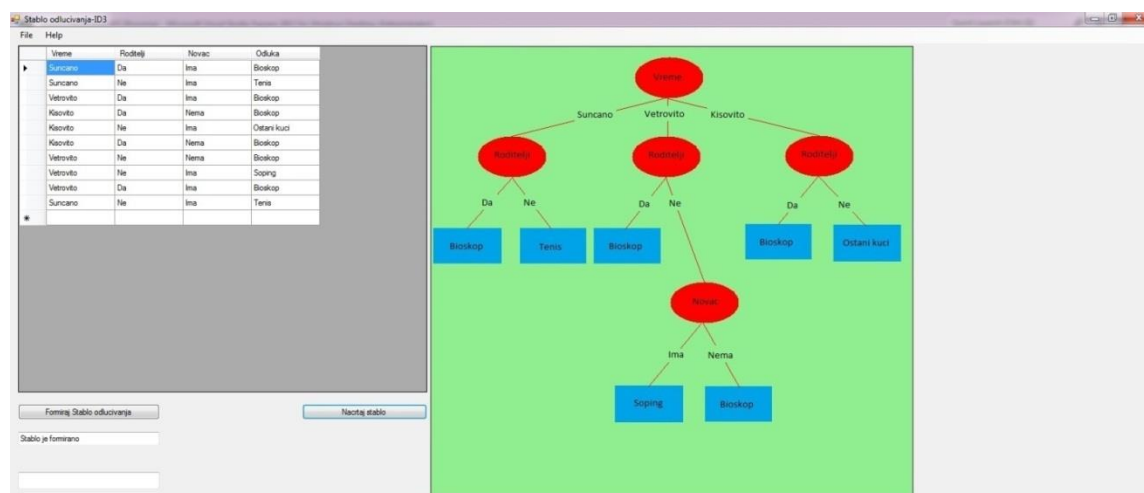
Formiranje stabla traje veoma kratko i nakon završetka formiranja dobijemo obaveštenje o završetku formiranja.



Pošto je stablo odlučivanja formirano možemo ga grafički prikazati pritiskom na dugme.



Aplikacija crta formirano stablo odlučivanja na formu, i konačan izgled ekrana izgleda ovako:



Slika 23. Stablo odlučivanja

2.10. Mogućnosti primene stabla odlučivanja u donošenju odluka u raznim oblastima

Stabla odlučivanja su našla široku upotrebu u raznim oblastima u kojem se barata sa velikim brojem podataka koje čovek ne može sam da sagleda i ne može da izvuče nikakav zaključak iz njih bez upotrebe računara.

Neke od oblasti gde se uspešno upotrebljavaja stabla odlučivanja [13]:

- **Astronomija** : U ovoj oblasti su stabla odlučivanja korišćena u filtriranju šuma sa slika koje je napravio Hubble Space teleskop. Upotrebljena su kod klasifikacije galaksija i zvezda, kao i u otkrivanju kvazara.
- **Proizvodnja** :Stabla odlučivanja se koriste za povećanje produktivnosti, za optimizaciju procesa, otkrivanju grešaka u procesu proizvodnje Boeinga, kontrolu kvaliteta.
- **Medicina** : Upotrebljava se u dijagnostici, kardiologiji, psihijatriji, mamografiji.
- **Molekularna biologija** : Postoje projekti kao banka podataka ljudskih genoma koji nude velike mogućnosti za mašinsko učenje i istraživanje podataka.
- **Prepoznavanje oblika** : Stabla odlučivanja su korišćena u prepoznavanju trodimenzionalnih objekata.
- **Farmakologija** : U ovoj oblasti je stablo odlučivanja korišćeno za analizu lekova.
- **Fizika** :Stabla odlučivanja su korišćena u otkrivanju fizičkih čestica.

- Bolesti biljaka : Nedavno su stabla odlučivanja korišćena u nalaženju procenta opasnosti od sušenja borova.
- Sistemi napajanja : Kod ovih sistema je korišćena za procenu bezbednosti sistema i stabilnosti napajanja

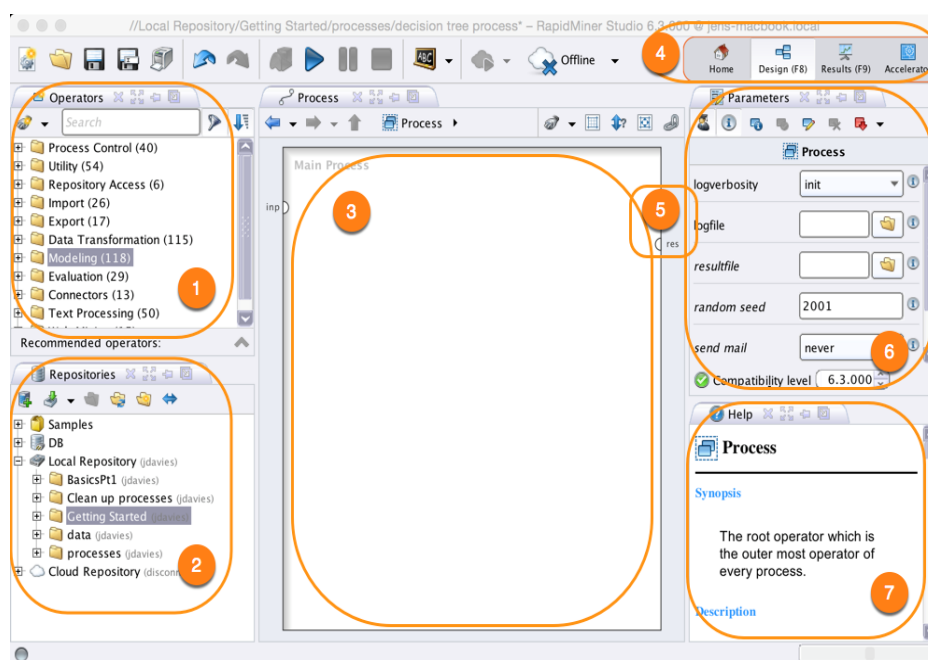
3. Prikaz alata za istraživanje podataka - Rapidminer

3.1. Uvod

Rapidminer je softverska platforma koja omogućuje integrisano okruženje za mašinsko učenje, istraživanje podataka, kao i za poslovnu analitiku. On se najčešće koristi za poslovne i industrijske aplikacije, kao i za potrebe istraživanja, edukacije, treninga. Podržava sve korake istraživanja podataka uključujući vizualno predstavljanje rezultata, validaciju i optimizaciju.

3.2. Osnovne funkcije Rapidminer-a

Glavni ekran izgleda ovako:



Slika 24. Glavni ekran Rapidminer-a [14]

Sledi kratak opis funkcionalnih celina koji su prikazani na slici 24 [14]:

1. **Operators** : ovde se nalaze sve funkcije koje podržava Rapidminer i one služe za kreiranje procesa. To su gradivni blokovi, grupisani po funkciji. Operator ima ulazne i izlazne portove. Svaki operator ima parametre kojima se kontroliše ponašanje operatera. U Rapidminer-u ima više od 1.500 operatora.
2. **Repositories**: mehanizam za skladištenje podataka. Preporučuje se da se koristi skladište podataka umesto čitanja direktno iz datoteke ili baze podataka
3. **Main Process**: deo ekrana gde se grade procesi za istraživanje podataka
4. **Perspectives**: radno područje za pristup specifičnim funkcijama
 - **Home** : pristup tutorialima, kreiranje novih ili otvaranje procesa
 - **Design** : panel za kreiranje procesa
 - **Results** : prikaz rezultata procesa.
 - **Accelerator** : radni prostor sa unapred definisanim procesima
5. **Ports**: Ulazno – izlazni mehanizam za operatore i procese

6. **Parameters:** podešavanje parametara za operatore. Postoje obični i ekspertni parametri koji su predstavljeni *italic* fontom. U slučaju dilema oko parametara, Rapidminer može predložiti vrednosti parametara na osnovu ranijih iskustava korisnika.
7. **Help:** opis funkcija za izabrani operator

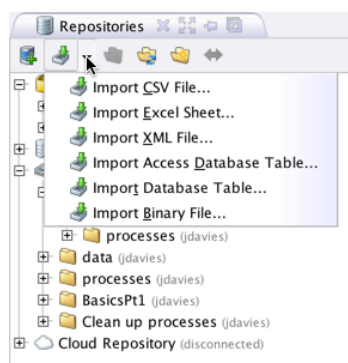
3.3. Povezivanje Rapidminer-a sa podacima

Podacima koje koristimo u toku istraživanja podataka možemo pristupiti na više načina. Oni mogu biti na lokalnom računaru, na SQL serveru ili čak na Cloud-u.

Ovde ćemo prikazati učitavanje jednog Excel fajla u lokalni folder [15].

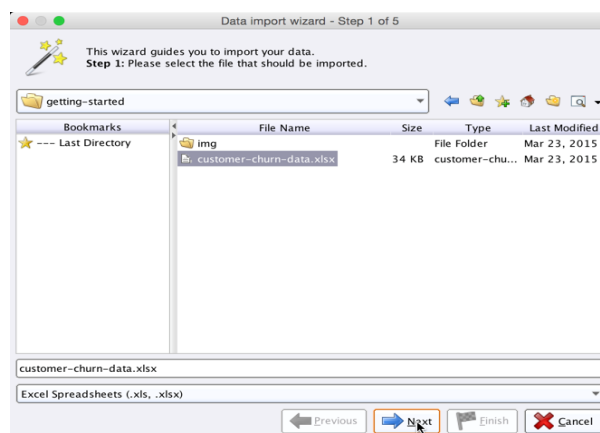
U panelu Repositories: kliknemo na ikonicu  koja nam služi za učitavanje podataka.

Iz padajućeg menija se bira opcija Import Excel sheet



Slika 25. Izgled Import Excel Sheet opcije

Sledeći korak je izbor excel fajla iz kojeg se učitavaju podaci.



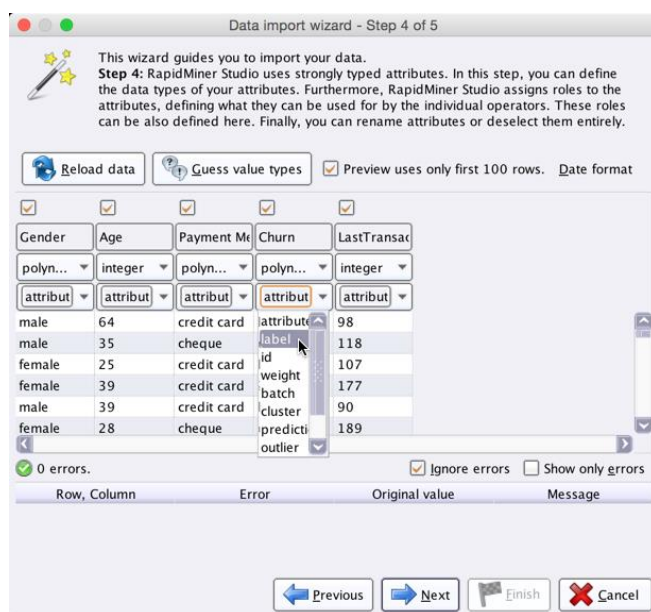
Slika 26. Izbor Excel fajla

Nakon izbora fajla i sheet-a, Rapidminer postavlja prvi red kao zaglavlje podataka a zatim prikazuje podatke koji su spremni za učitavanje.



Slika 27. Izgled podataka koji se učitavaju

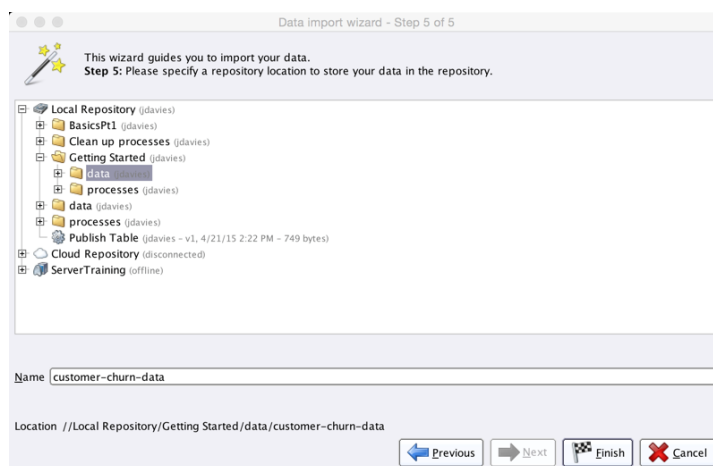
Posle ovog koraka potrebno je definisati tipove podataka koje učitavamo. Postoji mogućnost da se neka kolona ne učitava.



Slika 28. Definisanje tipova podataka koje učitavamo

Kod ovog koraka je bitno da se definiše **attribut**, njime određujemo koju ćemo kolonu koristiti za predviđanje rezultata i ovu kolonu označavamo sa **label**. Ovo je bitno kod nekih tipova istraživanja podataka kao što se stabla odlučivanja, itd.

Završni korak je određivanje lokacije gde ćemo snimiti naše podatke.

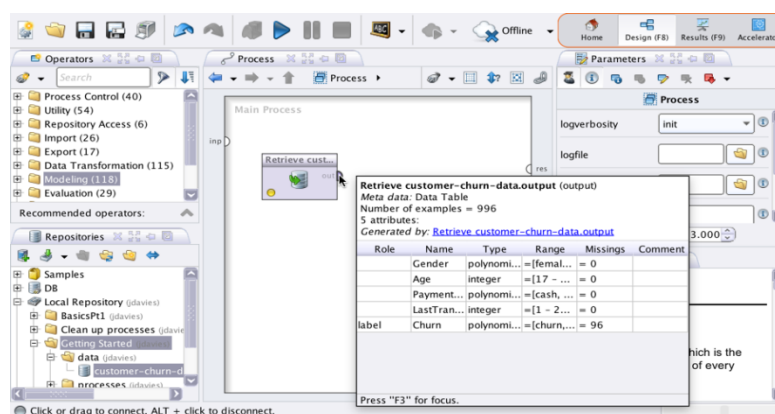


Slika 29. Izbor lokacije gde se snimaju podaci

3.4. Kreiranje procesa u Rapidminer-u

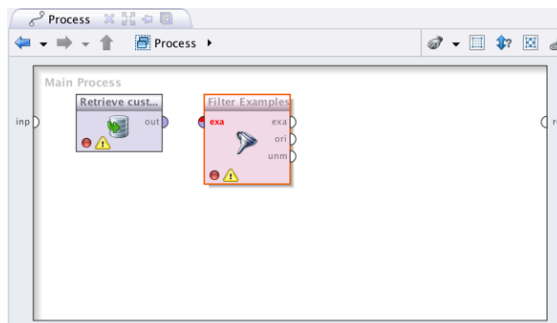
Procesi u Rapidminer-u nam služe za dizajniranje toka istraživanja podataka. Ceo tok istraživanja je grafički predstavljen, pa nam je tako rad veoma olakšan. Sve se svodi na prevlačenje raznih operatora na radnu površinu, podešavanje parametara i pravljenje veza između operatora.

Prvi korak u procesu je definisanje podataka koje ćemo obrađivati. Jednostavnim prevlačenjem ikonice iz Repositories-a, koja nam predstavlja podatke, na radnu površinu.

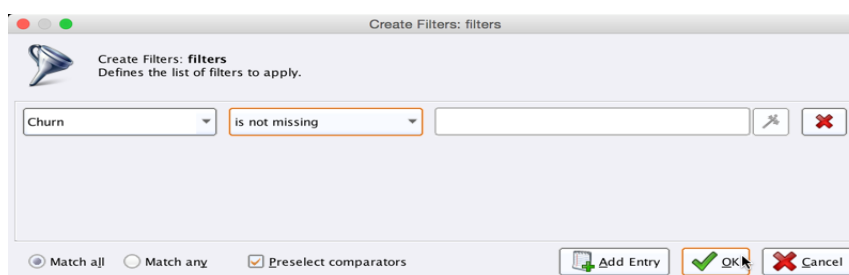


Slika30. Definisanje podataka za istraživanje [16]

Podacima sa kojima ćemo raditi često je potrebna priprema da bi se uspešno završio proces istraživanja. Ovde ćemo filtrirati podatke iz početnog skupa podataka. Prvo ćemo iz Repositorija prevući operator Filter examples , a zatim ćemo postaviti parametre za filter.

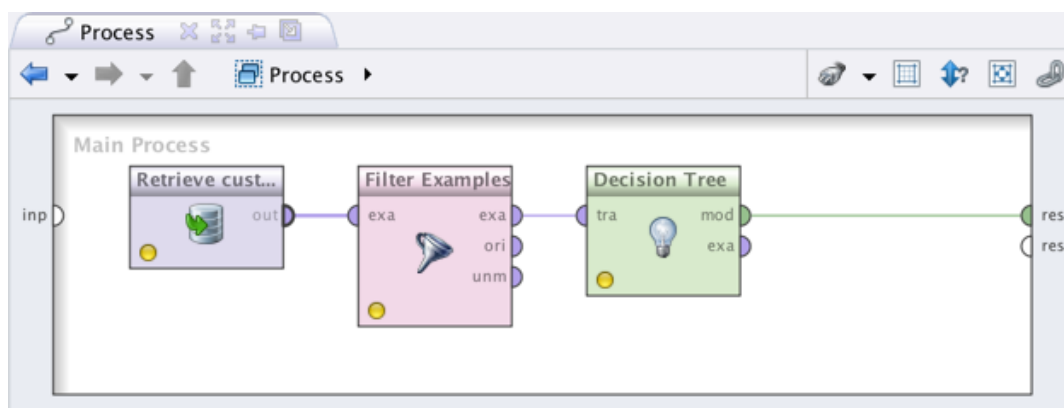


Slika 31. Filtriranje podataka [16]



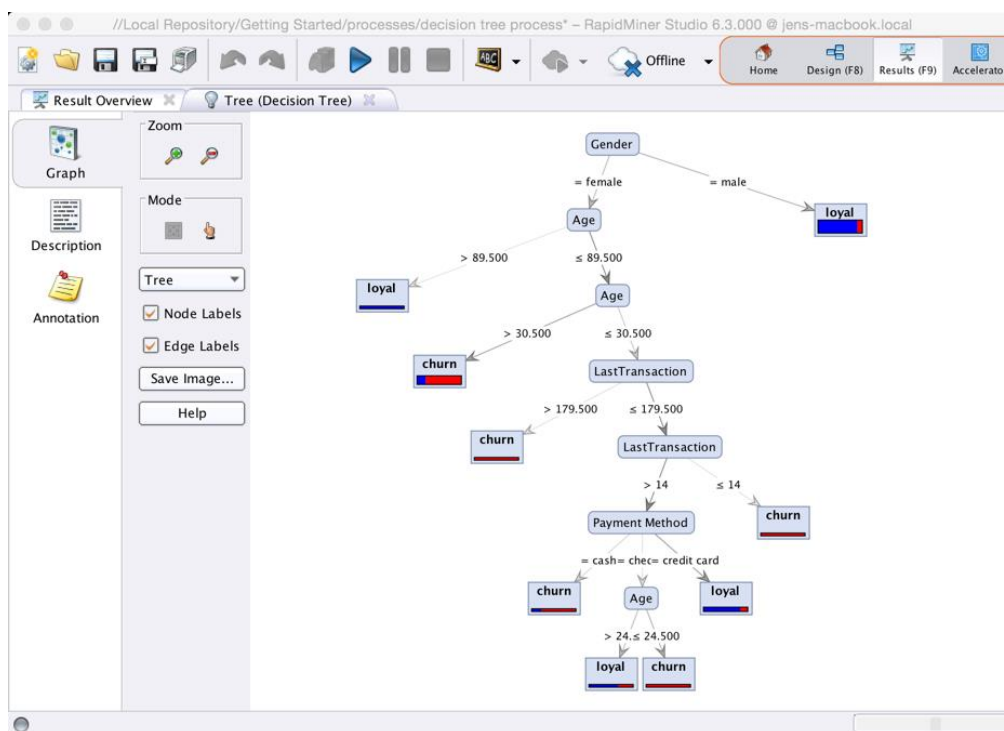
Slika 32. Definisanje filtera [16]

Kad smo završili sa pripremom podataka možemo da ubacimo operator za modelovanje podataka. U ovom primeru je ubačen operator koji formira stablo odlučivanja i povezaćemo portove.



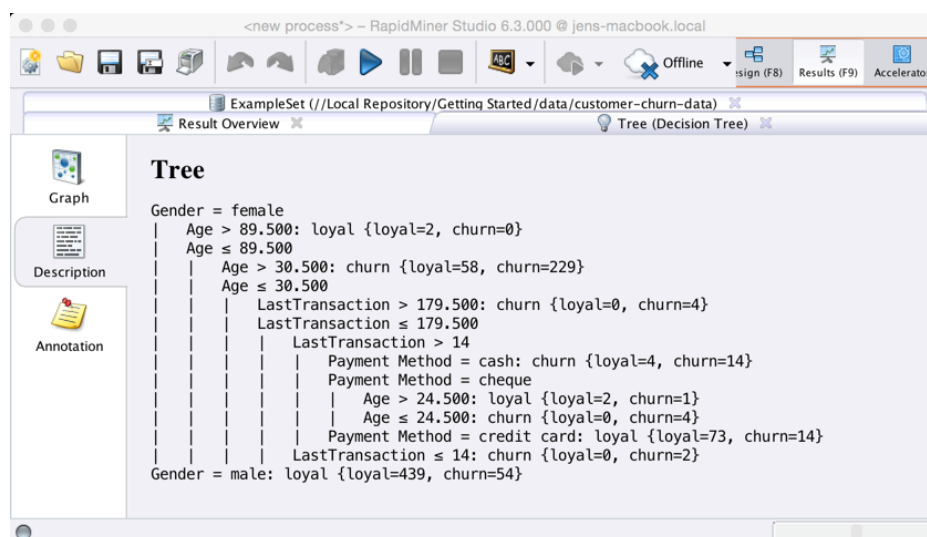
Slika 33. Proces formiranja stabla odlučivanja [16]

Posle povezivanja svih operatora možemo startovati proces koji smo definisali.



Slika34. Izgled stabla odlučivanja [16]

Rezultat stabla odlučivanja kao i ostalih modela možemo videti i u tekstualnoj formi.



Slika 35. Tekstualni prikaz rezultata [16]

3.5. Tipovi podataka u Rapidminer-u

Rapidminer podržava sledeće tipove podataka:

- **attribute** – ovde spadaju svi mogući podaci
- **binomial** – tačno dve vrednosti (yes/no ili true/false)
- **date** – datum u obliku 31.12.2014
- **date_time** – datum i vreme u obliku 31.12.2014 15:45
- **integer** - celi brojevi
- **nominal** – sve tekstualne vrednosti, uključujući i polynomial i binomial
- **numeric** – numeričke vrednosti, uključujući date, time, integer i realne brojeve
- **polynomial** – tekstualne vrednosti
- **real** - decimalni brojevi
- **text** – tekstualne vrednosti
- **time** – vremenske vrednosti u obliku 15:45

3.6. Najznačajniji operatori

Pošto Rapidminer ima oko 1.500 operatora, ovde ćemo opisati samo neke najznačajnije ili one najreprezentativnije za neke procese.

Operatori su podeljeni u sledeće grupe [17]:

- Process Control
- Utility
- Repository Access
- Import
- Export
- Data Transformation
- Modeling
- Evaluation
- Connectors

Slede opisi operatora prema vrsti.

3.6.1. Process Control

3.6.1.1. Handle Exception



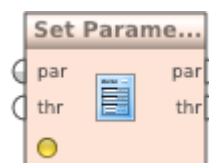
Funkcija koju obavlja ovaj operator je veoma slična obradi grešaka u programskim jezicima. Operator pokušava da izvrši proces definisan sa Try. Ukoliko nema grešaka proces vraća rezultat Try procesa. Ako se javila neka greška proces se ne zaustavlja nego se izvršava proces definisan sa Catch. Poruka o grešci se može snimiti pomoću *exception macro* parametra.

3.6.1.2. Throw Exception



Ovaj operator prikazuje grešku čim se pojavi neki podatak koji dovodi do netačnog rezultata. Proces se odmah zaustavlja.

3.6.1.3. Set Parameters



Ovaj parametar služi za definisanje parametara koje ćemo pridružiti nekom operatoru tokom procesa. Potrebno je definisati dva parametra:

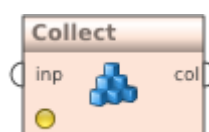
- set operator name – kojim definišemo naziv
- operator name – definiše se operator kojem ćemo preneti definisane parametre

3.6.1.4. Branch



Ovaj operator testira uslov koji smo definisali u parametrima. Ako je uslov zadovoljen onda se izvršava prvi proces, u drugom slučaju se izvršava drugi proces. Ovo je slično *If-Then-Else* naredbama u programskim jezicima.

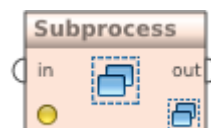
3.6.1.5. Collect



Ovaj operator sakuplja više ulaznih objekata u jedinstvenu kolekciju objekata.

3.6.2. Utility

3.6.2.1. Subprocess



Ovim operatorom definišemo neki podproces koji treba da se izvrši pre daljeg koraka. Ovo liči na potprograme u programiranju.

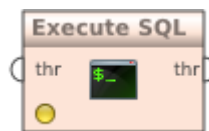
Kad se klikne na ovaj operator otvara se novi prozor u kome se definiše šta treba da se uradi u ovom koraku, a pre nastavka procesa.

3.6.2.2. Set macros



Ovaj operator se koristi za definisanje makroa koji se može referencirati sa `%{macro_name}` sa vrednostima parametara operatora tekućeg procesa. Ovaj makro mogu koristiti svi operatori u procesu koji dolaze posle definisanja makroa.

3.6.2.3. Execute SQL



Ovaj operator izvršava SQL naredbu na navedenoj bazi podataka iz tekućeg procesa. SQL query može se definisati kroz *query* parametar. SQL naredba može biti i u nekom fajlu. U tom slučaju je potrebno da se navede putanja do tog fajla u *query file* parametru. Ovaj operator se ne može koristiti za učitavanje podataka iz neke baze podataka.

3.6.2.4. Generate Sales Data



Ovaj operator generiše skup podataka za testiranje. Ovaj operater se koristi ako nemamo skup podataka koji prezentuje stvarne podatke za prodaju. On ima sedam običnih atributa i jedan specijalan.

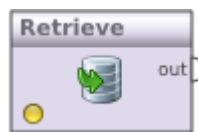
Atributi su:

- store_id
- customer_id
- product_id
- producy_category
- date
- amount
- single_price
- transaction_id – specijalan atribut

Broj primera koji se definišu određujemo sa number examples parametrom.

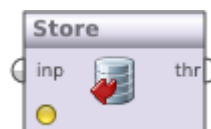
3.6.3. Repository Access

3.6.3.1. Retrieve



Ovaj operator se koristi za pristup podacima u skladištu podataka. Preporučuje se korišćenje ovog operatora za pristup podacima, zato što omogućuje pristup meta podacima. Najlakše se definiše tako što jednostavno prevučemo fajl iz Repositories u radni deo, i ankon toga Rapidminer sam definiše ovaj operator sa parametrima za pristup podacima.

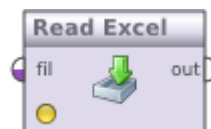
3.6.3.2. Store



Ovaj operator skladišti podatke u skladište podataka. Podaci mogu biti ulazni podaci iz nekog fajla ili rezultat nekog procesa. Potrebno je definisati parametar *repository entry* u kojem se definiše lokacije gde se smeštaju podaci.

3.6.4. Import

3.6.4.1. Read Excel



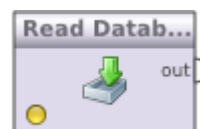
Ovaj operator se koristi za učitavanje podataka iz Excel tabela. Korisnik treba da definiše koja tabela i koji sheet se učitava. Format treba da je takav da svaki red predstavlja jedan primer a svaka kolona predstavlja atribut. Prvi red tabele se može koristiti kao naziv atributa.

3.6.4.2. Read CSV



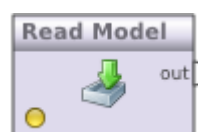
CSV fajlovi čuvaju podatke u tekstualnoj formi, gde su vrednosti obično razdvojene zarezom. Svaki red označava jedan primer. Ovi fajlovi se najlakše učitavaju uz pomoć *import configuration wizard-a*.

3.6.4.3. Read Database



Ovaj operator se koristi za čitanje podataka iz SQL baze podataka. Kad se izvrši ovaj operator, tabela koja se učitava se smešta u memoriju računara na kom se radi. Ovaj način rada povećava performanse izvršavanja procesa. Ako je tabela velika i ne može da se skladišti u memoriju, bolje je koristiti Stream Database operator, koji memoriše samo deo tabele i ovo utiče malo na performanse.

3.6.4.4. Read Model



Ovaj operator čita model iz fajla koji je naveden u *model file* parametru. Operator može da čita podatke zapisane u XML, XML Zipped i Binary formatu.

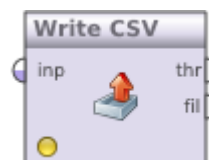
3.6.5. Export

3.6.5.1. Write Access



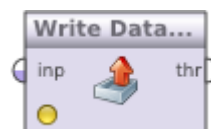
Ovaj operator služi za upisivanje podataka u navedenu Acces bazu podataka koristeći JDBC ili ODBC.

3.6.5.2. Write CSV



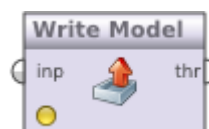
Ovaj operator služi za zapisivanje podataka u .CSV formatu, gde su vrednosti razdvojene nekim separatorom, obično zarezom. Ovaj separator se može navesti u *column separator* parametru. Praznim poljima predstavljena su vrednosti koji nedostaju.

3.6.5.3. Write Database



Operator služi za upisivanje podataka u navedenu SQL bazu podataka. Korisnik može definisati konekciju prema bazi kao i naziv tabele u koju se upisuju podaci. Ukoliko ne postoji definisana tabela onda će se kreirati nova tokom procesa upisivanja podataka. Moguće je pre upisivanja podataka primeniti neke operatore za filtriranje ili selektovanja podataka.

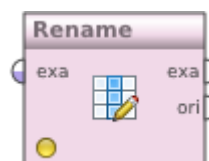
3.6.5.4. Write Model



Ovaj operator zapisuje model u fajl koji može biti u .XML, XML Zipped ili binarnom formatu. Ovaj operator je koristan zbog toga što možemo model sačuvati i posle upotrebiti u nekom drugom procesu.

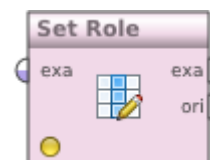
3.6.6. Data Transformation

3.6.6.1. Rename



Ovaj operator služi za preimenovanje jednog ili više atributa. Nazivi moraju biti jedinstveni. Ovaj operator ne utiče na tip ili rolu atributa. Za promenu tipa ili rolu treba koristiti operatore koji za to služe, kao na primer *Set Role* ili *Data transformation* operator

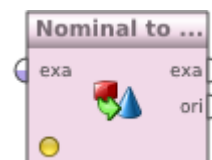
3.6.6.2. Set Role



Ovo je veoma bitan operator iz razloga što mnogi operatori za učenje zahtevaju određene role za attribute. Sa ovim operatorom menjamo rolu nekog atributa, i on važi samo u toku tekućeg procesa, ne menja se za stalno. Imamo

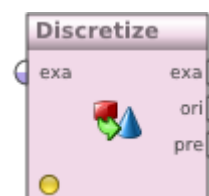
regularne i specijalne role. Regularne samo opisuju attribute, dok se specijalne role koriste za neki određeni zadatak. Specijalne role su: label, ID, prediction, cluster, weight i batch. Na primer za stablo odlučivanja je potrebno da je jedan atribut označen kao label, i on treba da bude ciljni atribut.

3.6.6.3. Nominal to Text



Ovaj operator služi za konvertovanje svih nominalnih vrednosti u stringove. Ako neka vrednost nedostaje, nova vrednost će takođe biti prazna.

3.6.6.4. Discretize by Entropy

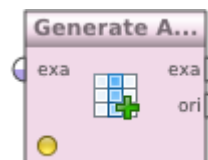


Ovaj operator diskretizuje izabrane numeričke vrednosti u nominalne vrednosti. Diskretizacija se vrši tako što se prave podskupovi koji minimizuju entropiju u napravljenim podskupovima. Ovaj metod se rekursivno ponavlja za podskupove dok se ne dostigne konačni kriterijum.

Diskretizacija se može vršiti na četiri načina:

- **Discretize by Binning** , prave se podskupovi gde je obim svih podskupova isti
- **Discretize by Frequency** , prave se podskupovi gde je broj jedinstvenih vrednosti skoro isti
- **Discretize by Size** , podskupovi imaju veličinu koju zadaje korisnik
- **Discretize by User Specification** , korisnik zadaje kriterijum po kojem se vrši diskretizacija

3.6.6.5. Generate Attributes



Ovaj operator generiše novi atribut iz postojećih atributa seta podataka koji istražujemo. Novi atribut se može definisati uz pomoć raznih matematičkih funkcija, funkcija transformacija i sl. Ukoliko je neka vrednost nedefinisana, onda novi atribut dobija vrednost „?“.

3.6.6.6. Select Attributes



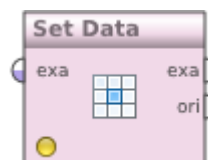
Sa ovim operatorom možemo definisati koji atribut ćemo koristiti a koji nam je nebitan za istraživanje.

3.6.6.7. Remove Useless Attributes



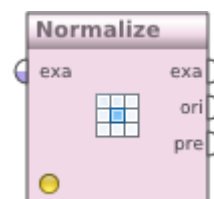
Ovaj operator uklanja beskorisne attribute iz seta podataka. Prag za beskorisne attribute definiše korisnik.

3.6.6.8. Set Data



Ovaj operator služi za postavljanje vrednosti za neke attribute izabranog seta podataka. Parametar *attribute name* određuje koji atribut će biti setovan. Dok parametar *value* određuje novu vrednost.

3.6.6.9. Normalize



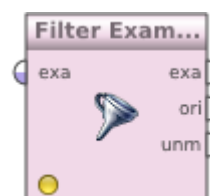
Normalizacija je funkcija koja prepravlja vrednosti atributa tako da se uklape u određeni obim.

3.6.6.10. Replace Missing Values



Ovaj operator zamenjuje vrednosti koji nedostaju sa unapred definisanim vrednostima. Ove vrednosti mogu biti minimalne, maksimalne ili srednje vrednosti tog atributa. Takođe se nedostajuće vrednosti mogu zameniti i nulom ili nekom drugom vrednošću.

3.6.6.11. Filter Examples



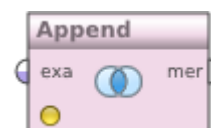
Ovaj operator filtrira skup podataka i kao izlaz ima samo one primere koji zadovoljavaju navedene uslove. Sa ovim operatorom se često filtriraju primeri koji nemaju vrednosti. Takođe se može koristiti za filtriranje tačnih i netačnih predviđanja posle testiranja nekog modela.

3.6.6.12. Pivot



Ovaj operator grupiše skup podataka slično kao pivot funkcija u Excelu. Sa *group attribute* parametrom se određuju atributi po kojem se vrši grupisanje, dok se sa *index attribute* definišu se atributi na čije se vrednosti primenjuje neka funkcija (sumiranje, srednja vrednost, maximum, minimum i sl.)

3.6.6.13. Append



Ovaj operator spaja dva ili više skupova podataka u jedan kombinovan skup. Uslov je da svi skupovi podataka imaju isti raspored atributa. Pošto se ovaj spojen skup podataka nalazi u meoriji on nije pogodan za rad sa jako velikim skupovima podataka.

3.6.7. Modeling

Rapidminer raspolaže sa velikim brojem modelovanje podataka, zato ćemo navesti samo spisak postojećih modela. Na kraju rada će biti urađeno nekoliko primera istraživanja podataka gde će se koristiti neki od ovih modela [17].

Classification and Regression

- Lazy Modeling

 - Default Model

 - k-NN

- Bayesian Modeling

 - Naive Bayes

 - Naive Bayes (Kernel)

- Tree Induction

 - CHAID

 - Decision Stump

- Decision Tree
 - Decision Tree (Multiway)
 - Decision Tree (Weight-Based)
 - ID3
 - Random Forest
 - Random Tree
- Rule Induction
 - Rule Induction
 - Single Rule Induction
 - Single Rule Induction (Single Attribute)
 - Subgroup Discovery
 - Tree to Rules
- Neural Net Training
 - AutoMLP
 - Neural Net
 - Perceptron
- Function Fitting
 - Gaussian Process
 - Linear Regression
 - Local Polynomial Regression
 - Polynomial Regression
 - Relevance Vector Machine
 - Seemingly Unrelated Regression
 - Vector Linear Regression
- Logistic Regression
 - Logistic Regression
 - Logistic Regression (Evolutionary)
- Support Vector Modeling
 - Fast Large Margin
 - Hyper Hyper
 - Support Vector Machine
 - Support Vector Machine (Evolutionary)
 - Support Vector Machine (LibSVM)
 - Support Vector Machine (Linear)
 - Support Vector Machine (PSO)
- Discriminant Analysis
 - Linear Discriminant Analysis

Quadratic Discriminant Analysis

Regularized Discriminant Analysis

Meta Modeling

AdaBoost

Additive Regression

Bagging

Bayesian Boosting

Classification by Regression

Find Threshold (Meta)

Hierarchical Classification

MetaCost

Polynomial by Binominal Classification

Relative Regression

Stacking

Subgroup Discovery (Meta)

Transformed Regression

Vote

Attribute Weighting

Data to Weights

Weight by Chi Squared Statistic

Weight by Component Model

Weight by Correlation

Weight by Deviation

Weight by Gini Index

Weight by Information Gain

Weight by Information Gain Ratio

Weight by PCA

Weight by Relief

Weight by Rule

Weight by SVM

Weight by Tree Importance

Weight by Uncertainty

Weight by User Specification

Weight by Value Average

Weights to Data

Optimization

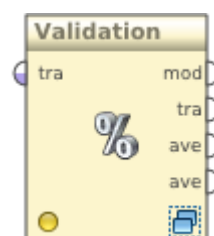
Optimize Weights (Backward)

- Optimize Weights (Evolutionary)
- Optimize Weights (Forward)
- Optimize Weights (PSO)
- Clustering and Segmentation
 - Agglomerative Clustering
 - DBSCAN
 - Expectation Maximization Clustering
 - Extract Cluster Prototypes
 - Flatten Clustering
 - Random Clustering
 - Support Vector Clustering
 - Top Down Clustering
 - X-Means
 - k-Means
 - k-Means (Kernel)
 - k-Means (fast)
 - k-Medoids
- Association and Item Set Mining
 - Apply Association Rules
 - Create Association Rules
 - FP-Growth
 - Generalized Sequential Patterns
 - Item Sets to Data
 - Unify Item Sets
- Correlation and Dependency Computation
 - ANOVA Matrix
 - Correlation Matrix
 - Covariance Matrix
 - Grouped ANOVA
 - Mutual Information Matrix
 - Rainflow Matrix
 - Transition Graph
 - Transition Matrix
- Similarity Computation
 - Cross Distances
 - Data to Similarity
 - Data to Similarity Data

- Similarity to Data
- Model Application
 - Apply Model
 - Create Formula
 - Group Models
 - Ungroup Models
 - Update Model
- Thresholds
 - Apply Threshold
 - Create Threshold
 - Find Threshold
 - Select Recall
- Confidences
 - Drop Uncertain Predictions
 - Generate Prediction
 - Generate Prediction Ranking
 - Rescale Confidences

3.6.8. Evaluation

3.6.8.1. Split Validation



Ovaj operator nam služi za testiranje modela u tom smislu da deli podatke u dva skupa podataka. Jedan skup je skup podataka za testiranje a drugi za formiranje modela. Sa ovim operatorom se procenjuje koliko tačno radi model.

3.6.9. Connectors

3.6.9.1. Read Dropbox



Ovaj operator pravi konekciju sa Dropbox skladištem podataka. Da bi se podaci učitali potrebno je kombinovati ovaj operator sa operator koji učitava

određeni tip fajla. Na primer, da bismo učitali Excel fajl, potrebno je kombinovati ovaj operator sa Read Excel operatorom.

3.6.9.2. Read Salesforce



Sa ovim operatorom učitavamo podatke sa Salesforce platforme. Ovaj operator omogućuje kreiranje jednostavnih upita, ili korišćenje naprednog SOQL editora koji ima Salesforce.

3.6.9.3. Search Twitter



Ovaj operator pretražuje Twitter statuse. Moguće je kreiranje upita kojim možemo definisati uslove za učitavanje podataka.

3.7. Portovi

Pošto svaki operator obrađuje neki skup podataka, potrebno je definisati ulazne podatke i izlazne podatke za svaki operator. Za definisanje ovih podataka služe portovi, i oni su predstavljeni sa malim polukrugovima na ivicama operatora. Definisanje podataka je veoma jednostavno i vrši se na takav način što se ulazni i izlazni portovi povezuju linijama koje povlačimo sa mišem.

Svaki operator ima definisane ulazne i izlazne portove, tako da je naš zadatak da spajajući portove operatora definišemo koje ćemo podatke da prosledimo operatoru i zatim koje izlazne podatke ćemo koristiti u daljem procesu.

Ovo je spisak portova koje postoje u Rapidminer-u [18]:

Skraćenica	Značenje	Opis
ano	<i>Anova</i>	ANOVA matrix for ANOVA significance test
ann	<i>Annotation</i>	Annotations extracted from the input object
arc	<i>Archive</i>	Archive file generated during execution of the operator
ass	<i>Association</i>	Association rules that have been discovered in a frequent item set
att	<i>Attribute</i>	Attribute weights (in and out)
ave	<i>Average</i>	Performance measures; estimate of performance using the model built on the complete delivered data set
clu	<i>Cluster model</i>	Cluster model created when clustering an example set
clu	<i>Clustered set</i>	Example set given to the clustering operator; may contain an attribute with a cluster role (describes the cluster of each example)
col	<i>Collection</i>	Collection of objects
con	<i>Condition</i>	Any object can be supplied; the condition specified in parameters is tested on this object
cov	<i>Covariance</i>	Covariance matrix
dic	<i>Dictionary</i>	Example set used for replacing 'from' values with 'to' values in a given example set
dis	<i>Distance measure</i>	SimilarityMeasure object
doc	<i>Document</i>	Document or document set
err	<i>Error</i>	Standard error output
est	<i>Estimated performance</i>	Performance vector of the SVM model which gives an estimation of statistical performance of this model
exa	<i>Example set</i>	Example set
fil	<i>File</i>	File object
fla	<i>Flat</i>	Flat collection or flat clustering model
for	<i>Formula</i>	Formula result

fre	<i>Frequent</i>	Frequent item or item sets for association rule learning
gro	<i>Grouped</i>	Grouped models, attributes, items
hie	<i>Hierarchical</i>	Hierarchical clustering model
inp	<i>Input</i>	Input source, can take various objects
ite	<i>Item sets</i>	Frequent item sets (groups of items that often appear together in the data)
joi	<i>Join</i>	Join of the left and right example sets
lab	<i>Labeled data</i>	Model that was given in input is applied on the example set and the updated example set is delivered from this port
lef	<i>Left</i>	Left input port expecting an example set, which is used as the left example set for a join
lif	<i>Lift chart</i>	Lift Pareto chart for the given model and example set
mat	<i>Matrix</i>	Correlations matrix of all attributes of the input example set
mer	<i>Merged</i>	Merged example set
mod	<i>Model</i>	Default model from this output port
obj	<i>Object</i>	IO object
ori	<i>Original</i>	Input example set is passed without changing to this port
out	<i>Output</i>	Output port
par	<i>Parameter set</i>	Set of parameters that can be applied on an operator
pat	<i>Patterns</i>	GSP algorithm is applied on the given example set; resultant sequential patterns set is delivered through this port
per	<i>Performance</i>	Performance Vector for selected attributes
pre	<i>Preprocessing</i>	Preprocessing model with information regarding the operator's parameters in the current process
ran	<i>Random forest</i>	Model of a random forest
ref	<i>Reference</i>	Provided reference data or reference set
req	<i>Request set</i>	Provided example set

res	<i>Result set</i>	Distance or similarity between examples of the request set and reference set
rig	<i>Right</i>	Right input port expecting an example set, which is used as the right example set for a join
roc	<i>ROC curve</i>	Calculated ROC curves for included models
rul	<i>Rules</i>	Association rules that have been discovered in a frequent item set
sec	<i>Second</i>	Input take an example set derived from the output of the Generate ID operator in an attached example process
seg	<i>Segment</i>	Segment of an image
sel	<i>Selected</i>	Object specified by the index parameter is returned through this port
ses	<i>Session</i>	Session example set
sig	<i>Significance</i>	Significance test results of performance vector comparison is delivered through this port
sim	<i>Similarity</i>	Calculated similarity between each example of the given example set with every other example of the same set
sin	<i>Single</i>	Single object of the given collection, which is processed in the inner part of the operator
sta	<i>Stacking</i>	Stacking examples or model
sto	<i>Stored</i>	Through this port, the input object is passed without changing to the output
sub	<i>Subtrahend</i>	Expects an example set; example set must have ID attribute
sup	<i>Superset</i>	Superset of input example sets
thr	<i>Through</i>	Objects are passed through without changing
thr	<i>Threshold</i>	Threshold output of the Select Recall operator
tra	<i>Training</i>	Training data to train a model (example set)
uni	<i>Union</i>	Union of the input example sets
unl	<i>Unlabeled</i>	Examples that are not labelled and therefore not used when training a model
unm	<i>Unmatched</i>	Examples that did not match a specified pattern in the original example

set		
unr	<i>Unrelated</i>	Examples that were unrelated to a specified pattern in the original example set
vis	<i>Visualization</i>	Self-organizing map (SOM) visualization
wei	<i>Weights</i>	Attribute weights
wor	<i>Word</i>	Expects or outputs a word list
xsl	<i>XSLT</i>	EXtensible Stylesheet Language (XSLT) document

4. Rapidminer u akciji

4.1. Uvod

Da bi se što bolje prikazale mogućnosti Rapidmire-a u istraživanju podataka, ovde će biti prikazano kreiranje modela za testiranje da li je neka mail poruka SPAM ili ne.

SPAM poruke možemo definisati kao neželjene poruke koje se šalju velikom broju neselektovanih primaoca. Ove poruke mogu da izazovu bezbednosne pretnje kao i zagušivanje mreže ukoliko se ne filtriraju na vreme. Skeniranjem mejlova i filtriranjem ovih neželjenih i potencijalno opasnih poruka, možemo da ih sklonimo u karantin, gde ne predstavljaju rizik za naše mreže.

U ovom projektu za trening modela je korišćena baza SPAM podataka koja se nalazi na sajtu *Kalifornijskog univerziteta Irvine* pod nazivom SMS SPAM Collection [19].

Ta baza podataka ima tri atributa:

- ID - jedinstveni broj mejla
- SPAM – oznaka da li je mejl svrstana kao SPAM ili HAM
 - SPAM – neželjeni mejlovi
 - HAM – bezopasni (regularni) mejlovi
- Text – sadržaj poruke

Baza za testiranje našeg modela se nalazi na stranici *Dublinskog tehnološkog instituta* [19]. Struktura baze podataka je identična bazi za trening modela.

4.2. Model za testiranje SPAM poruka

4.2.1. Ukratko o kreiranju modela za testiranje SPAM mejlova

Klasifikacija mejlova kao SPAM poruka je u stvari proces razdvajanja mejlova kao SPAM (neženjena poruka) ili HAM (regularna poruka) koristeći klasterovanje mejlova ili klasifikaciju u grupe u skladu sa njegovim sadržajem.

Ovaj proces se može uraditi uz pomoć mašinskog učenja, koristeći razne algoritme. Ovo su neki od algoritama koje se mogu koristiti [20].

- Naive Bayesian

Klasifikator Naive Bayes generiše model predviđanja sa snažnom nezavisnošću od pretpostavke. Predstavlja semantički jasan i jednostavan pristup prikazu, upotrebi i indukciji probabilističkog znanja. Nazvan je “naivnim”, jer pojednostavljuje problematiku oslanjajući se na dve važne pretpostavke: pretpostavlja da su atributi koji se koriste u predviđanju uslovno nezavisni uz poznatu klasifikaciju i da ne postoje sakriveni atributi koji bi mogli uticati na proces predikcije. Ove pretpostavke omogućavaju vrlo

efikasne algoritme, kako za klasifikaciju tako i za mašinsko učenje. Za uslovno nezavisne attribute $A_1, \dots, A_k$ verovatnoća za klasni atribut C se izračunava po pravilu:

$$P(A_1, \dots, A_k | C) = \prod_{i=1}^k P(A_i | C)$$

Jednostavanost, efikasanost, laka interpretacija i pogodnost za male skupove podataka osnovne su prednosti Naive Bayes modela. Međutim, u praksi, snažna nezavisnost od pretpostavke može da naruši međusobnu zavisnost atributa [21].

- Support Vector Machine

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. Kao nadzirana metoda koja analizira podatke i prepoznaje obrasce, ona je strogo utemeljena na statističkim teorijama učenja i istovremeno smanjuje trening i test greške [22].

- k-Nearest Neighbour

Klasifikator k-NN direktno koristi trening skup za klasifikaciju, bez korištenja ikakve matematičke forme za funkcije gustine verovatnoće osnovnih klasa. Kod NN (Nearest Neighbor) klasifikatora svaki uzorak se klasifikuje prema klasi svog najbližeg suseda iz trening skupa. Kod k-NN klasifikatora, pronalazi se k najbližih suseda uzorka koji se klasifikuje i glasanjem se utvrđuje kojoj klasi uzorak pripada. Detaljnije, za svaki ulazni vektor obeležja koji se klasifikuje (pripada test skupu), pronalaze se klase k najbližih vektora suseda iz trening skupa, a zatim se ulazni vektor svrstava u onu klasu koja ima najviše članova. Kao mjera klase najbližih suseda koristi se metrika. Najčešće je to Euklidova ili Mahalanobisova distanca. Za $k=1$ dobija se najjednostavniji slučaj (NN klasifikator), gdje se klasa ulaznog vektora obeležja određuje prema klasi najbližeg (po izabranoj metrici) vektora iz trening skupa [23].

- k-Means

K-Means je algoritam čija glavna uloga je klasifikacija n objekata u k grupa. Cilj je da se u okviru iste grupe (engl. cluster) nađu objekti koji su slični. Na prvi pogled čini se da je taj zadatak jednostavan i da se ne razlikuje

od proste klasifikacije. Ali glavna razlika K-Means od ostalih algoritama koji vrše prostu klasifikaciju jeste njegova sposobnost da pre izvršavanja klasifikacije nema već postojeće grupe u koje treba da rasporedi objekte već u toku svog rada „uči“ kako da izvrši klasifikaciju.

Kao meru sličnosti objekata ovaj algoritam najčešće koristi njihovu euklidsku udaljenost u p-dimenzionalnom prostoru, gde p predstavlja broj atributa objekata koji učestvuju u klasifikaciji. Formula za izračunavanje udaljenosti u p-dimenzionalnom prostoru je:

$$d_{ij} = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

Grupisanje se vrši minimizacijom rastojanja između objekata tako da na kraju u jednoj grupi ostaju samo oni objekti koji su prostorno najbliži [24].

4.2.2. Kreiranje modela za testiranje SPAM poruka

U našem projektu ćemo koristiti SVM (Support Vector Machine) algoritam.

Naš model ćemo prvo „naučiti“ koji sve izrazi se javljaju u porukama koji su označeni kao SPAM. Zatim ćemo taj model testirati na podacima za testiranje, da bismo videli koliko je naš model tačan u razvrstavanju poruka na SPAM i HAM poruke.

Ovaj model je testiran skupu podataka od oko 5.500 sadržaja mejl poruka. Poruke se nalaze na sajtu *UCI-Machine Learning repository* [25]. Poruke su označene kao SPAM ili HAM u zavisnosti od njegovog sadržaja.

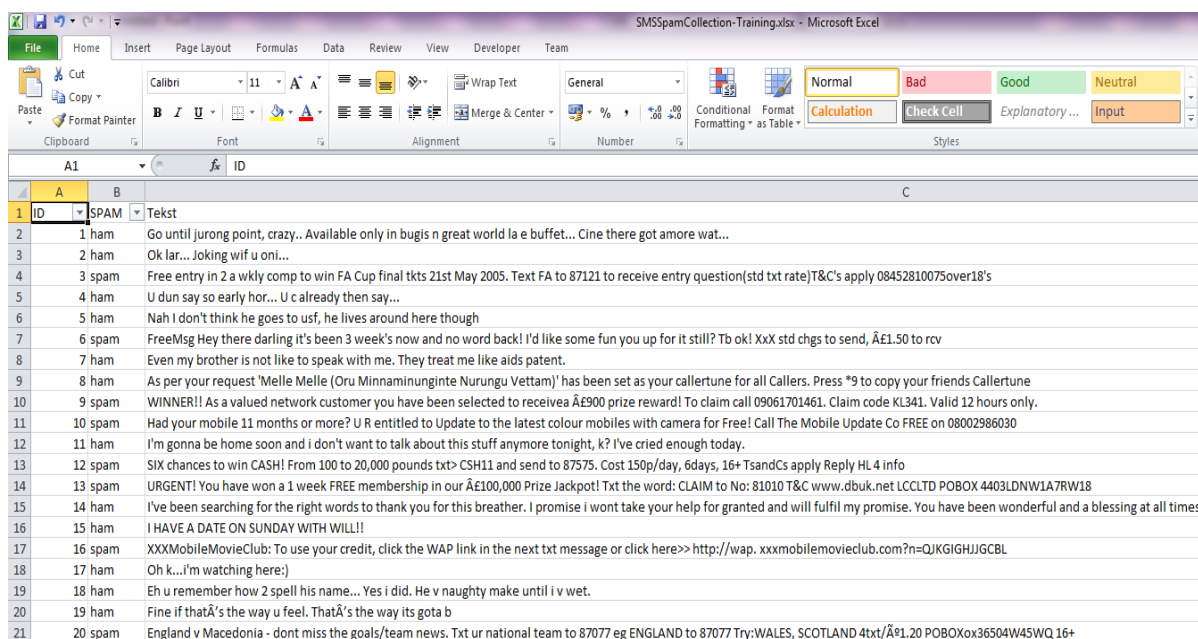
Oznaka	Broj poruka
SPAM	747
HAM	4.827
Ukupno:	5.574

Pošto model koji smo koristili ne omogućava deljenje skupa podataka na skup za trening i skup za testiranje, onda smo pripremili jedan skup za testiranje u odnosu otprilike 60:40 . Taj skup ćemo učitati u naš model i testiraćemo naš model na njemu.

Za testiranje je upotrebljen skup od oko 4.000 mejlova. Ovaj skup za testiranje modela je sakupljan iz raznih izvora, gde su obično bili mali skupovi podataka, pa su na kraju svi oni dodati u jedan skup podataka.

Oznaka	Broj poruka
SPAM	2.396
HAM	1.732
Ukupno:	4.128

Podaci koji će nam poslužiti za trening modela su sačuvani u .csv formatu i imaju sledeći izgled:

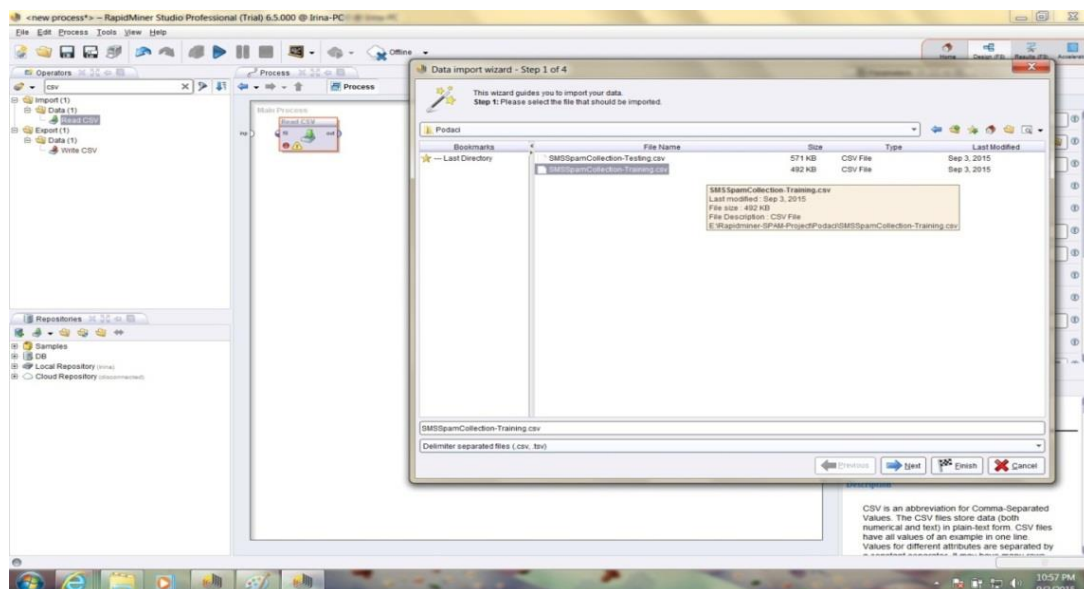


ID	SPAM	Tekst
1	ham	Go until jurong point, crazy... Available only in bugis n great world la e buffet... Cine there got amore wat...
2	ham	Ok lar... Joking wif u oni...
3	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
4	ham	U dun say so early hor... U c already then say...
5	ham	Nah I don't think he goes to usf, he lives around here though
6	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, Â£1.50 to rcv
7	ham	Even my brother is not like to speak with me. They treat me like aids patent.
8	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
9	spam	WINNER!! As a valued network customer you have been selected to receive a Â£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
10	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
11	ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.
12	spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL4 info
13	spam	URGENT! You have won a 1 week FREE membership in our Â£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
14	ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.
15	ham	I HAVE A DATE ON SUNDAY WITH WILL!!
16	spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJGIGIHJGCB
17	ham	Oh k...i'm watching here;
18	ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.
19	ham	Fine if that's the way u feel. That's the way its gota b
20	spam	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/Â£1.20 POBOXox36504W45WQ 16+

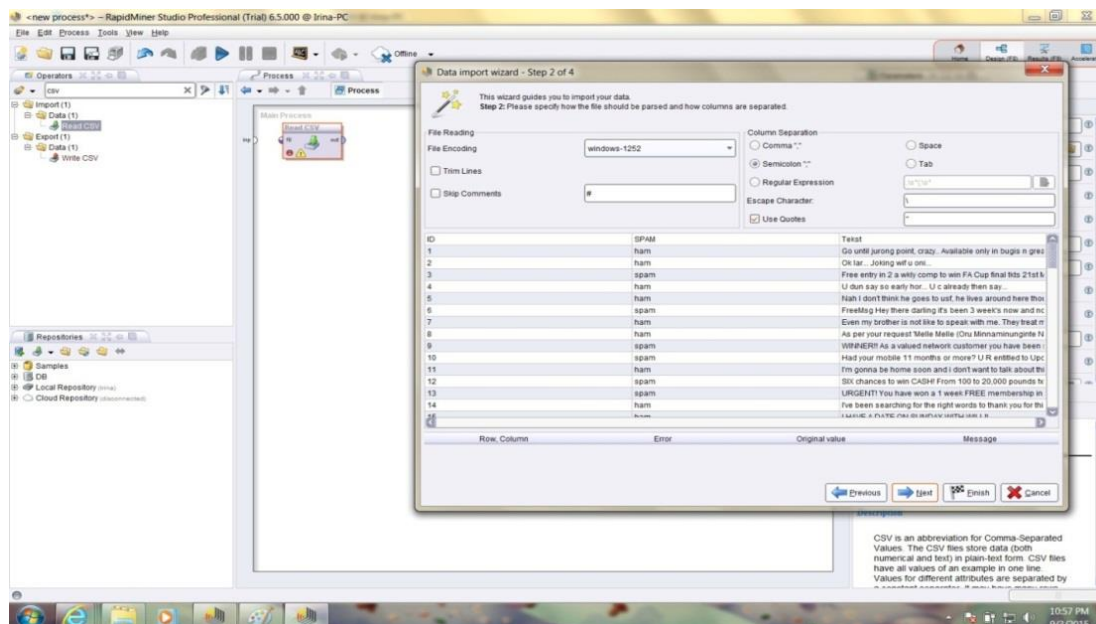
Slika 36. Podaci za trening modela

Prvi korak u kreiranju našeg modela sa kojim ćemo testirati da li je neka mejl poruka SPAM jeste povezivanje našeg modela sa podacima.

Za ovo ćemo koristiti operator *Read CSV* i Import Wizard koji postoji za ovaj operator, i koji nam jako olakšava povezivanje sa podacima.

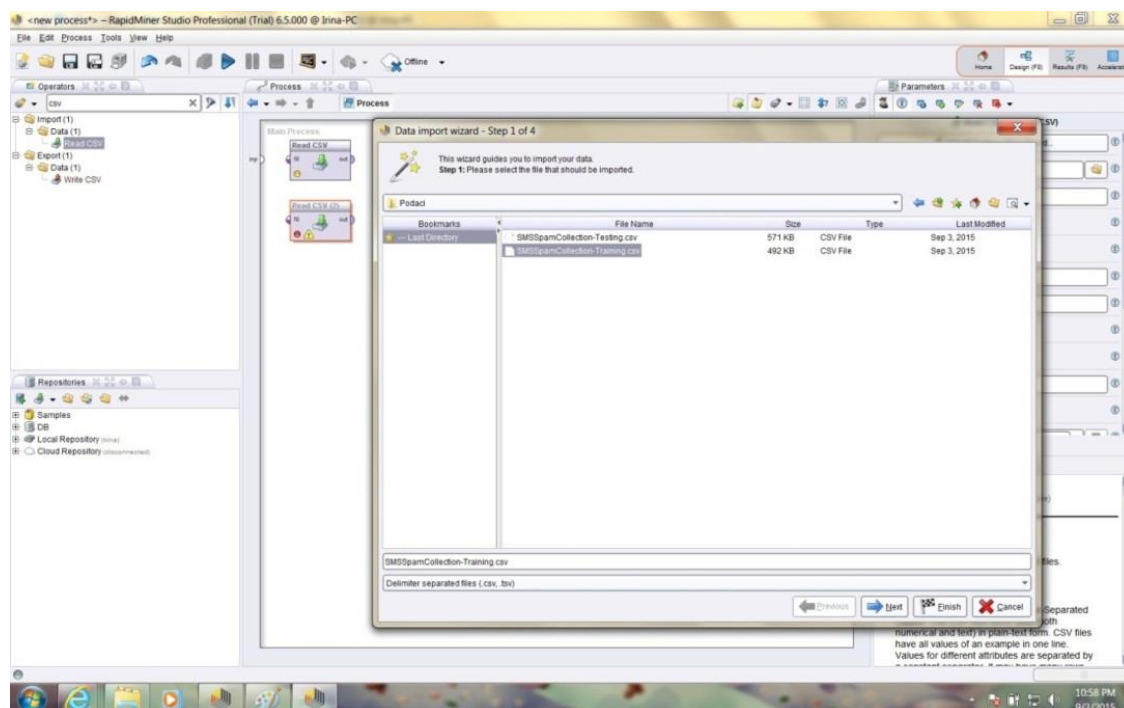


Slika 37. Definisanje fajla iz kojem se nalaze podaci za trening modela



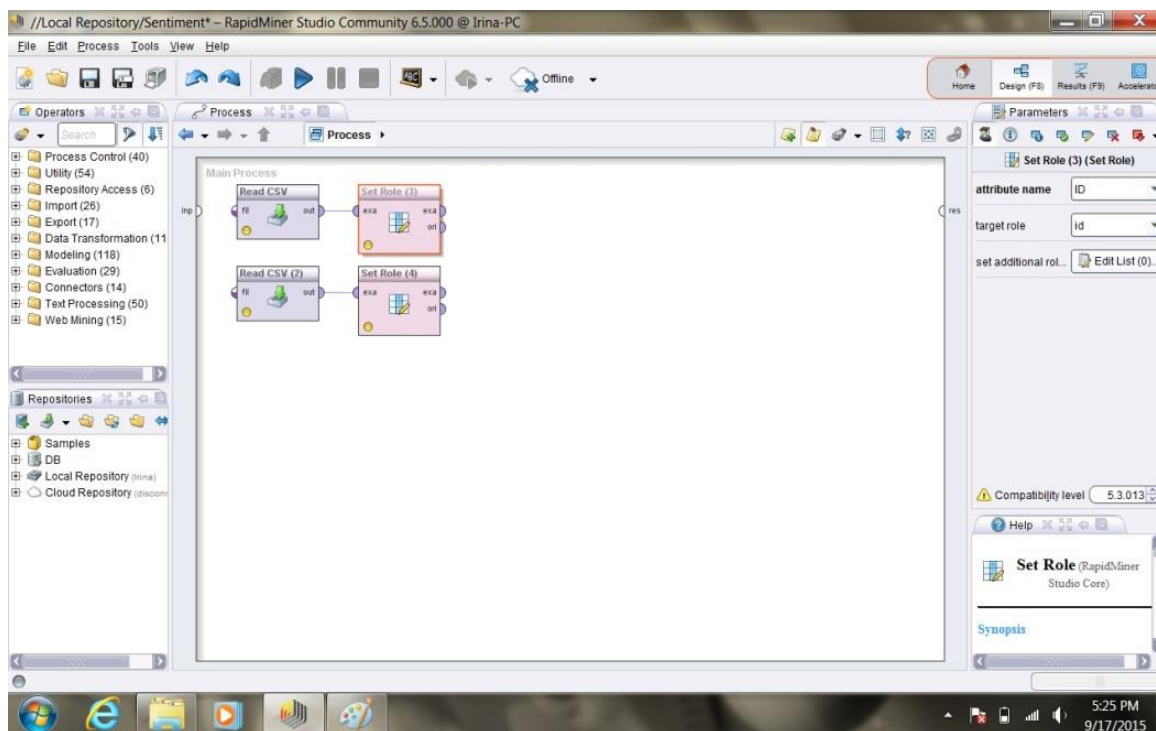
Slika 38. Definisanje separatora .csv fajla

Pošto ćemo u našem modelu koristiti podatke za trening i za testiranje, odmah ćemo definisati i podatke za testiranje modela.



Slika 39. Definisanje fajla iz kojeg se nalaze podaci za testiranje modela

Kad smo definisali ulazne podatke, potrebno je da definišemo ID attribute naših skupova podataka kao *target role id*.



Slika 40. Definisanje target rola ID

Sledeći korak je definisanje “word vector”-a uz pomoć operatora *Process document from data*.

Kako se u stvari formira „word vector“?

Ovaj vektor meri učestalost ponavljanja svakog izraza u dokumentu.

Formira se na sledeći način:

Na primer imamo dve rečenice [26]:

- 1) John likes to watch movies. Mary likes movies too.
- 2) John also likes to watch football games.

Na osnovu ove dve rečenice se formira lista jedinstvenih rečenica u dokumentu.

["John", "likes", "to", "watch", "movies", "also", "football", "games", "Mary", "too"]

U ovoj listi ima 10 jedinstvenih reči, pa će vektor imati 10 dimenzija.

Sledeći korak je da se izbroji broj ponavljanja ovih reči u naše dve rečenice.

Na primer reč „John“ se u prvoj rečenici javlja jednom, dok se reč „likes“ javlja dva puta. Na ovaj način smo došli do naša dva vektora koji predstavljaju naše rečenice

1) [1, 2, 1, 1, 2, 0, 0, 0, 1, 1]

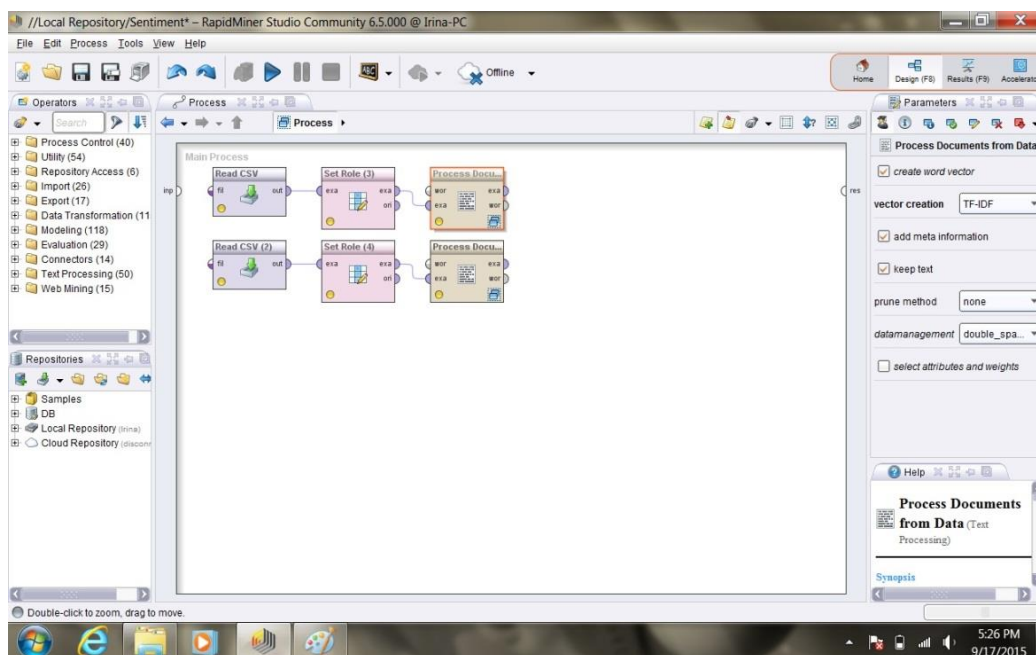
2) [1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

Vektor ćemo definisati uz pomoć TF-IDF metoda. Ovaj metod se sastoji od dva pojma:

- **TF (Term Frequency)** – broji se frekvencija ponavljanja pojmova u dokumentu. Pošto se neki izraz može ponavljati više puta u dokumentu onda se obično frekvencija deli sa ukupnim brojem pojmova u dokumentu
- **IDF (Inverse Document Frequency)** – meri važnost pojmova. Kad se računa TF svi pojmovi se smatraju da imaju istu važnost. Međutim neki izrazi (the, that, is, itd.) se mogu ponavljati mnogo puta u dokumentu, ali one imaju malu važnost. Zbog toga se oni moraju zanemariti.

Formula za računanje je sledeća [27]:

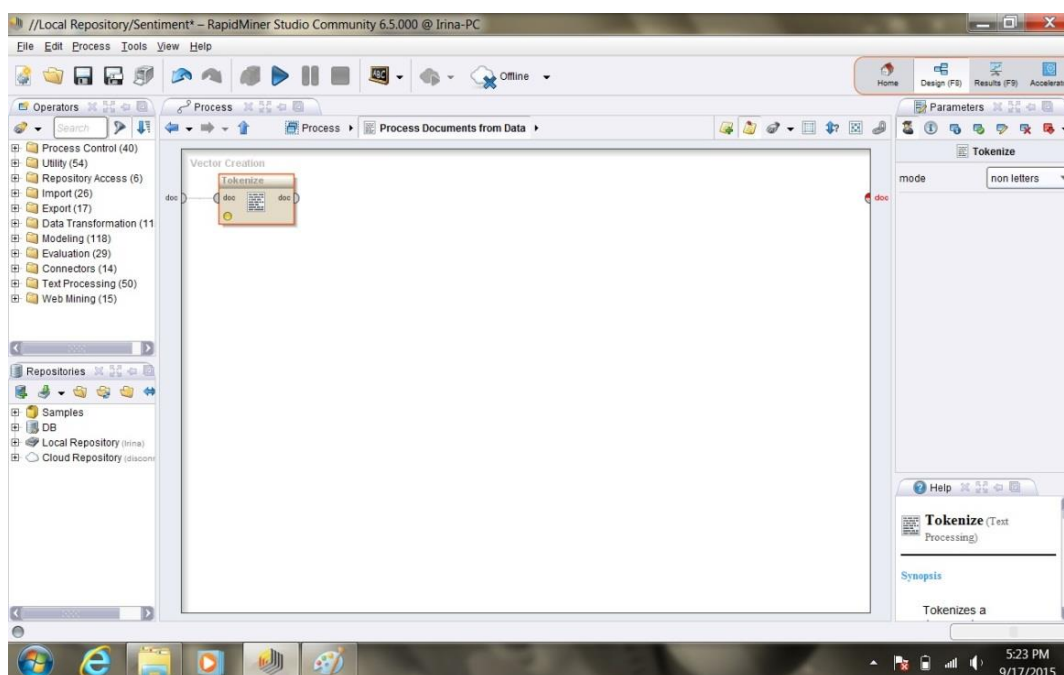
$$\text{IDF}(t) = \frac{\log e \text{ (Ukupan broj dokumenata)}}{\text{Ukupan broj dokumenta koji sadrže izraz } t}$$



Slika 41. Definisanje operatora Process document from data

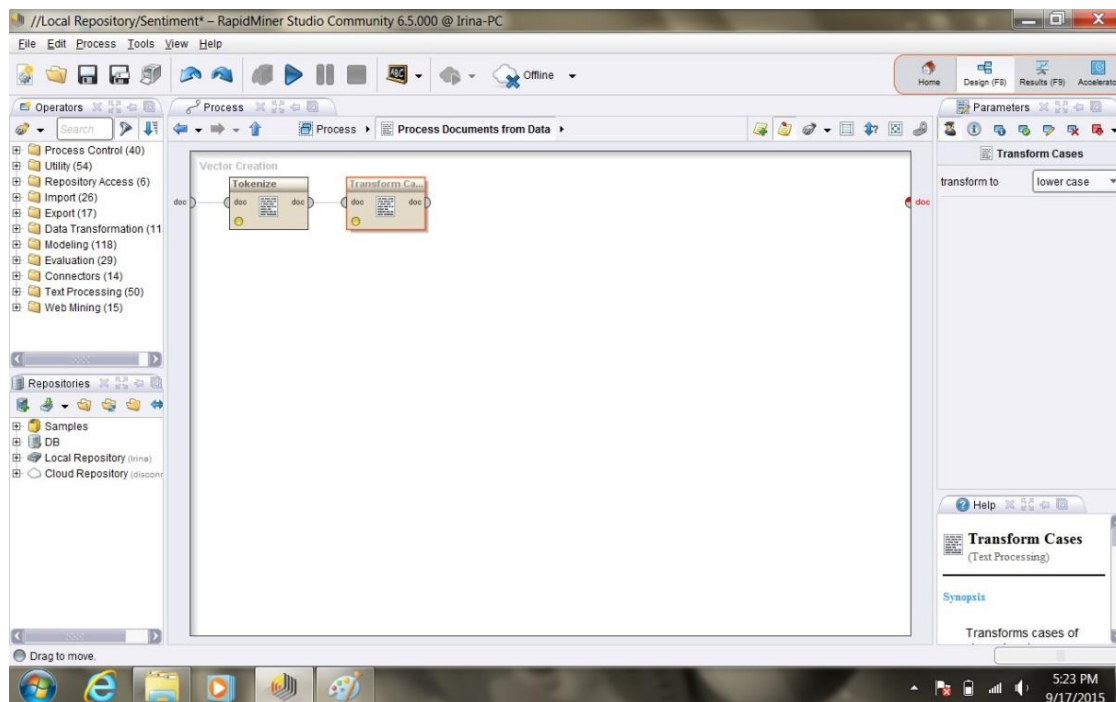
Da bismo kreirali vektor moramo iskoristiti neke operatore u podprocesu operatora *Process document from data*.

Prvi operator koji ćemo iskoristiti je *Tokenize*. Ovaj operator razbija tekst u fraze, reči, simbole ili neke druge značajne elemente u tzv. Tokene. Svaki zarez, razmak, tačka ili specijalni znak se uklanja i svi alfabetski stringovi se čuvaju kao tokeni.



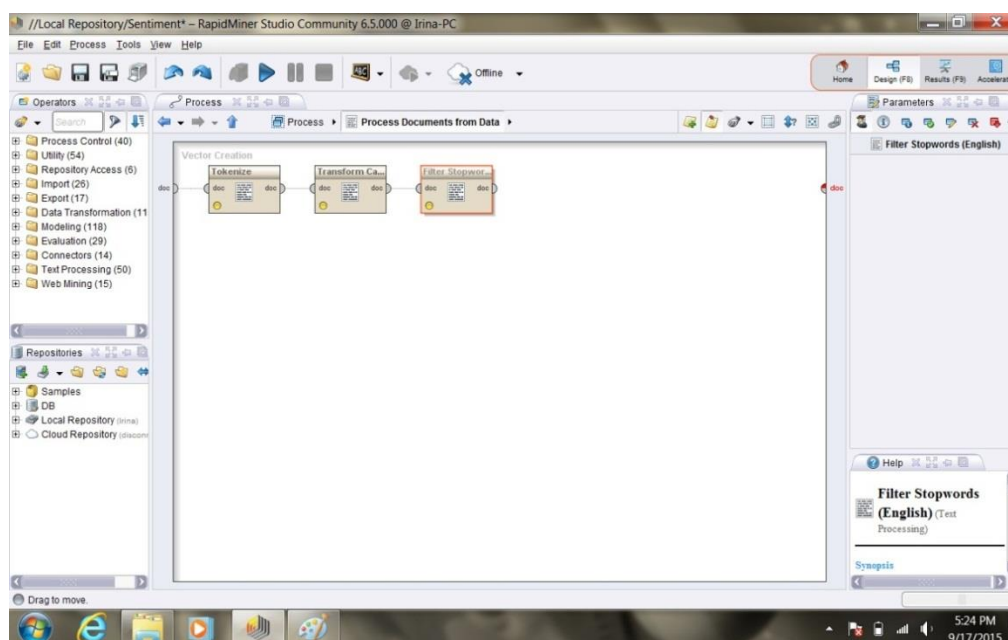
Slika 42. Operator Tokenize

Posle ovoga ćemo sav tekst pretvoriti u mala slova. Ovo se radi da bi se eventualne greške u kucanju tretirale isto.



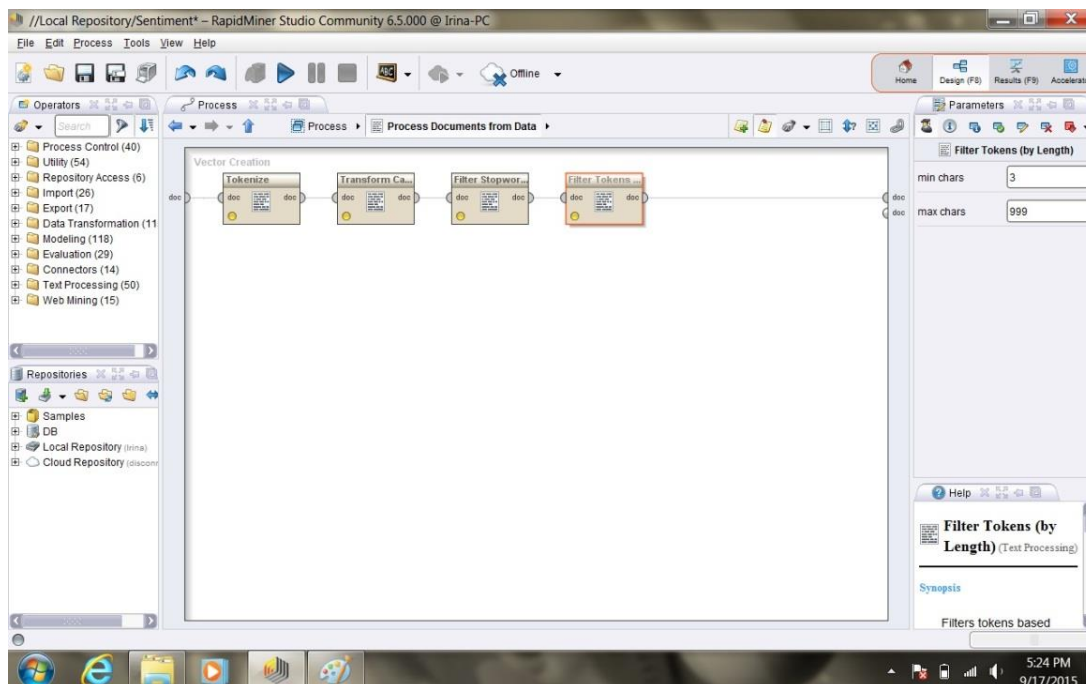
Slika 42. Operator za pretvaranje teksta u mala slova

Nakon ovoga ćemo izbaciti sve tzv. “stop words” za engleski. To su neke od najčešćih i najkraćih reči kao što su The, That, Who, Take i slične. One se izbacuju radi poboljšanja performansi modela.



Slika 43. Operator za izbacivanje “stop words”-a iz obrade

Poslednji korak će biti ograničavanje dužine teksta koji ćemo obrađivati u našem modelu. Ovde ćemo staviti da je dužina reči između 3 i 999 slova.

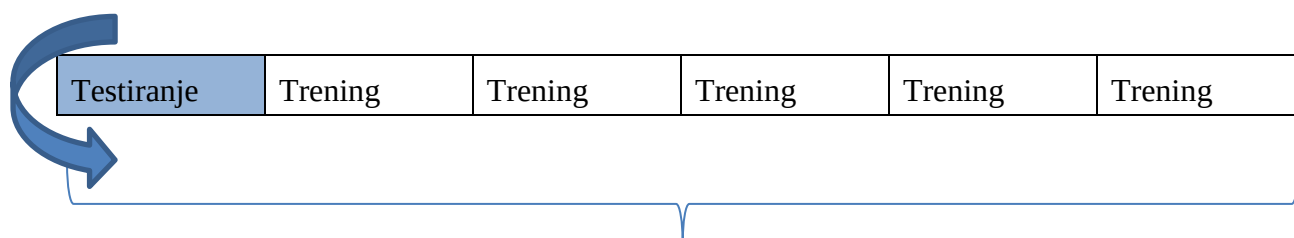


Slika 44. Ograničavanje dužine reči

Posle ovog koraka podaci su spremni za obradu. Iskoristićemo operator *X-Validation* za treniranje i testiranje modela. Ovaj operator ima dva podprocesa, jedan je za obuku modela a drugi za testiranje modela. Tokom faze testiranja se mere i performanse testiranja.

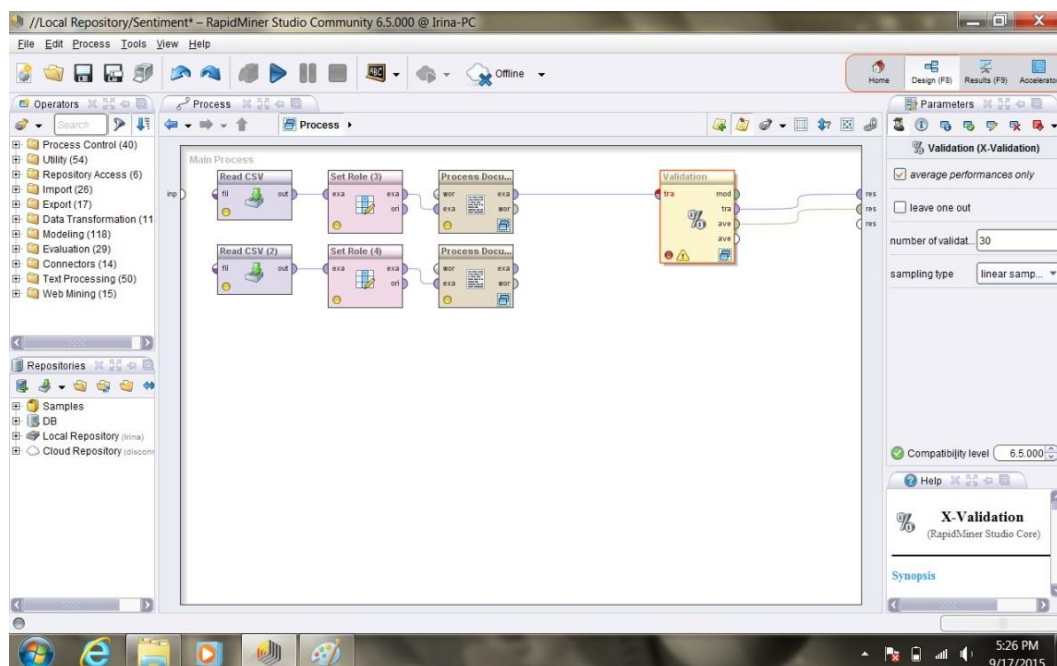
Ulazni skup podataka se deli na podskupove iste veličine, od kojih se jedan podskup zadržava kao skup za testiranje a ostali se koriste za obuku modela. Ovaj proces se ponavlja onoliko puta koliko ima podskupova, tako da se svaki podskup iskoristi za skup za testiranje.

Skup podataka za trening:



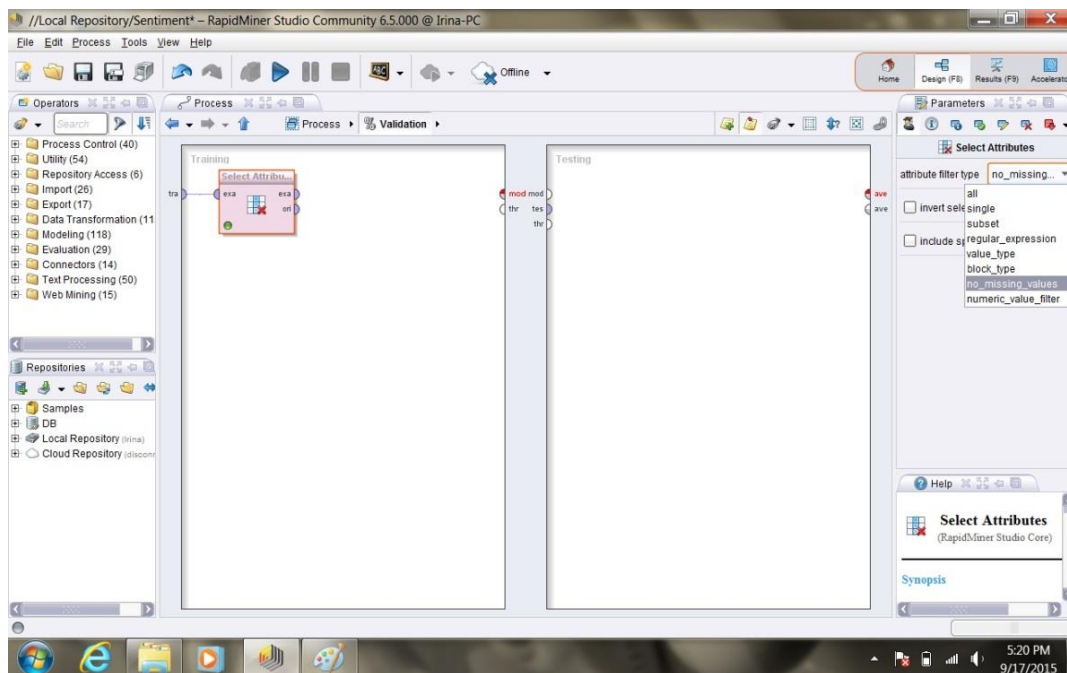
Svaki podskup se rotira i svaki će se iskoristiti za testiranje.

Broj podskupova se definiše sa parametrima ovog operatora.



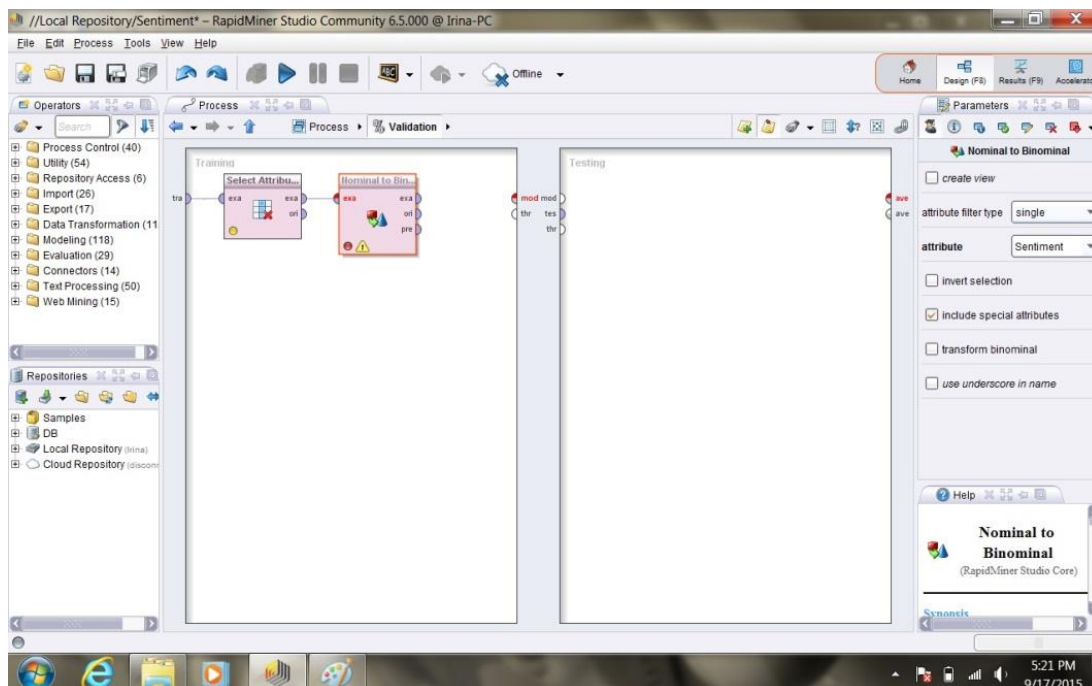
Slika 45. Operator X-Validation sa brojem validacije i načinom određivanja podskupova

U podprocesu za obuku ćemo prvo izabrati za obradu sve atribute, i definisati da se uzimaju u obzir samo podaci koji su kompletni, bez nedostajućih vrednosti za neke atribute.



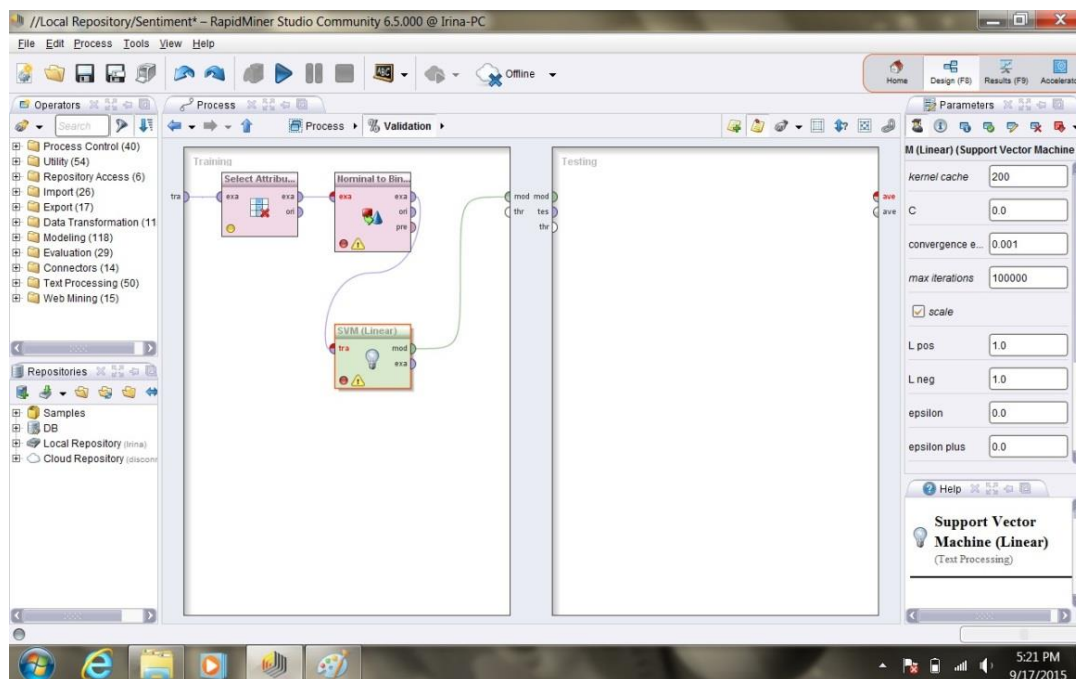
Slika 46. Izbor atributa i definisanje filtera

Ukoliko imamo nominalne vrednosti njih ćemo pretvoriti u binominalne sa operatorom *Nominal to Binominal*.



Slika47. Pretvaranje nominalnih vrednosti u binominalne

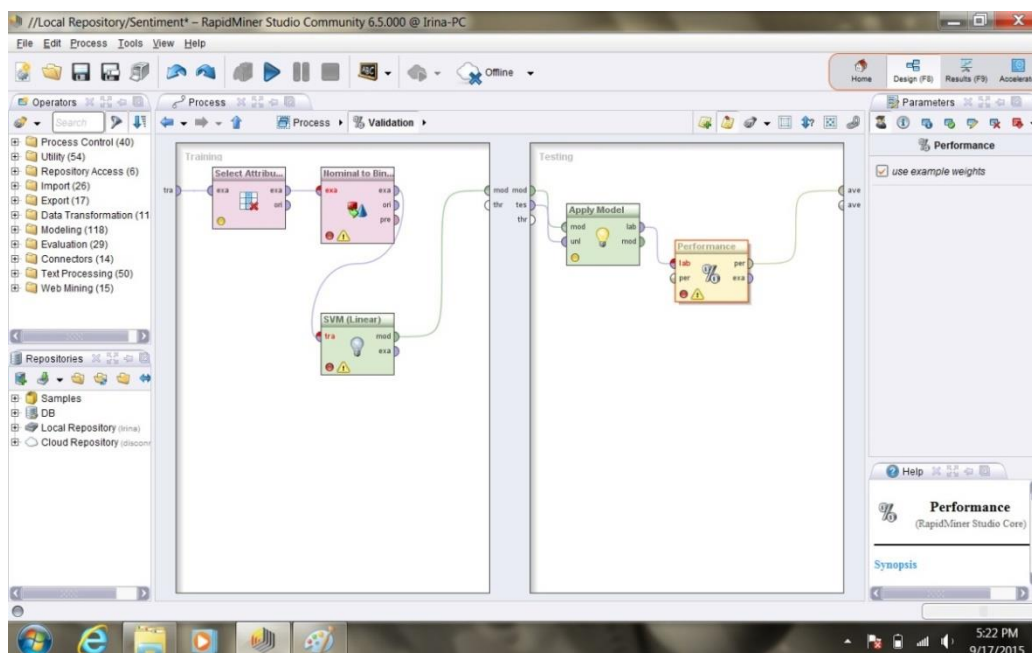
Sad je potrebno da postavimo operator koji će obraditi naše podatke i napraviti model za testiranje mejlova. U ovom slučaju ćemo koristiti operator SVM (Support Vector Machine – Linear).



Slika 48. Operator za generisanje modela SVM-Linear

Kad smo definisali ove parametre, prelazimo u podproces za testiranje modela.

U njemu definišemo dva operatora *Apply model* i *Performance*. Prvi primenjuje naučeni modela na skup podataka za testiranje a sa drugim testiramo performanse obučenog modela.

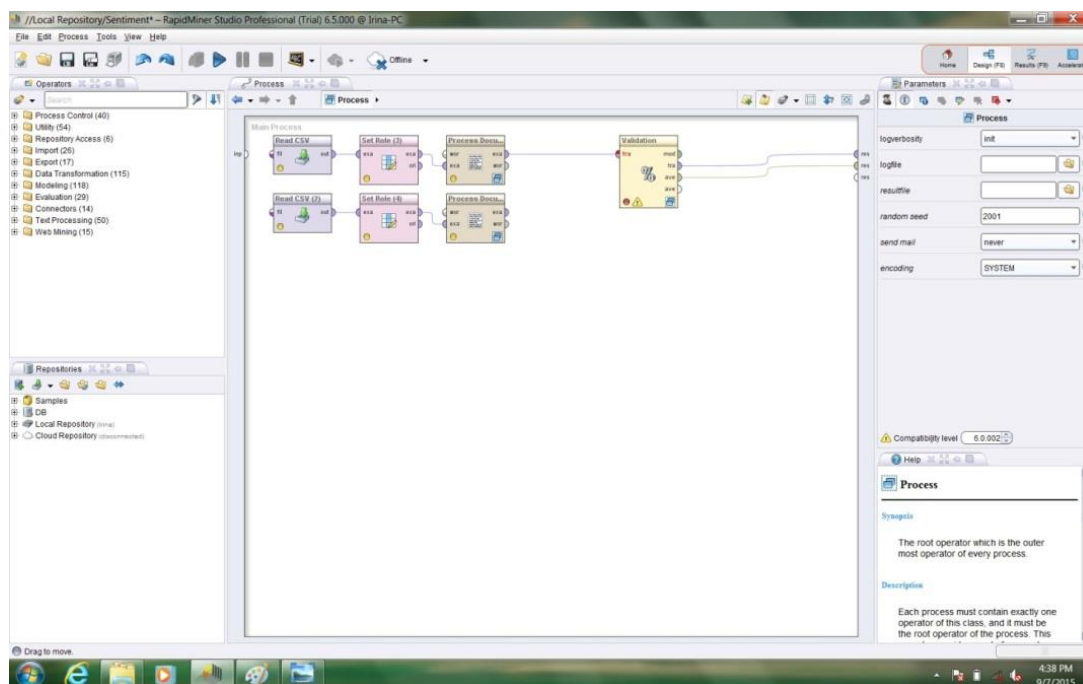


Slika 49. Operatori Apply Model i Performance

Ovim smo završili sa definisanjem modela.

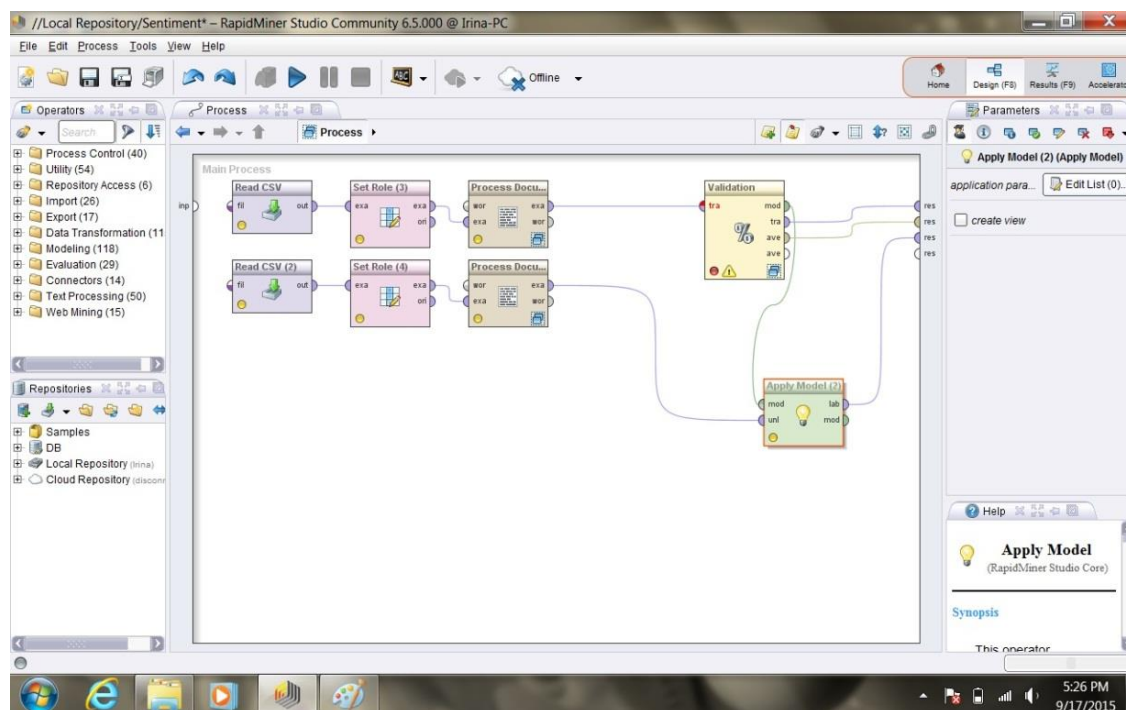
Operator *X-Validation* nema mogućnost za deljenje skupa podataka na deo za trening i testiranje u nekom odnosu (obično se deli skup podataka u odnosu 70:30%) pa je zbog toga pripremljen skup podataka na koji ćemo primeniti ovaj model i sa njim ćemo ga testirati. Odnos skupa podataka za trening i testiranje je otprilike 60:40. Ovaj odnos zavisi od količine podataka koji koristi model. Ukoliko imamo malo podataka treba biti oprezan sa deljenjem. Ako ostavimo mnogo podataka u skupu za trening, ostaće nam malo za testiranje. Sa manje podataka za testiranje statistika učinka će imati veliku varijansu. Kad je skup podataka veliki onda nije ni toliko bitno da li ćete deliti podatke u donosu 90:10 ili 70:30.

Ponovićemo postupak definisanja operatora *Process document from data* za skup podataka sa kojim ćemo testirati obučeni model. Njega definišemo na identičan način kao i za skup podataka za obuku.



Slika 50. Definisan operator za skup podataka na koji ćemo primeniti model

Kad smo definisali parametre i procese za ovaj operator primenjujemo model na skup podataka za testiranje i pokrećemo proces obuke za naš model.



Slika 51. Konačan izgled modela za testiranje SPAM poruka

4.2.3. Rezultat istraživanja modela za testiranje SPAM poruka

Glavni cilj kreiranja našeg modela je bio da se napravi model kojeg ćemo “naučiti” da klasifikuje tekst nekog mejla kao SPAM ili ne. Pošto se ovo ne može izvesti sa 100 % preciznošću, poželjno je da tačnost našeg modela bude barem blizu ovog procenta.

Nakon pokretanja procesa, sama obrada je trajala oko 5 minuta.

Prvi izveštaj koji nam daje Rapidminer se odnosi na preciznost sa kojom će naš model razvrstavati tekst kao SPAM ili HAM.

U koloni *prediction(SPAM)* nam je prikazano šta je predvideo naš model na osnovu podataka koje smo uneli.

U sledeće dve kolone se nalaze podaci o poverenju (confidence) sa kojim je model predvideo da li je tekst SPAM ili ne.

U nastavku se nalazi tekst poruke koji se analizira. U sledećim kolonama su prikazane značajne rečenice koje je naš model prepoznao u tekstovima poruka. Na osnovu frekvencije ponavljanja ovih rečenica u SPAM i HAM porukama SVM model klasifikuje tekstove kao SPAM ili HAM.

Row No.	ID	SPAM	prediction(SPAM)	confidence(ham)	confidence(spam)	text	aah	aaoooright	aathi	abbey
1	1	ham	ham	0.771	0.229	jurong point crazy	0	0	0	0
2	2	ham	ham	0.800	0.200	lar joking wif oni	0	0	0	0
3	3	spam	spam	0.298	0.702	free entry wky cor	0	0	0	0
4	4	ham	ham	0.786	0.214	dun say hor say	0	0	0	0
5	5	ham	ham	0.786	0.214	nah think goes us	0	0	0	0
6	6	spam	spam	0.312	0.688	freemsg hey darli	0	0	0	0
7	7	ham	ham	0.774	0.226	brother speak tre:	0	0	0	0
8	8	ham	ham	0.771	0.229	request melle me	0	0	0	0
9	9	spam	spam	0.295	0.705	winner valued net	0	0	0	0
10	10	spam	spam	0.351	0.649	mobile months er	0	0	0	0
11	11	ham	ham	0.784	0.216	want talk stuff any	0	0	0	0
12	12	spam	spam	0.309	0.691	chances win cast	0	0	0	0
13	13	spam	spam	0.306	0.694	urgent won week	0	0	0	0
14	14	ham	ham	0.801	0.199	searching words	0	0	0	0
15	15	ham	ham	0.771	0.229	date sunday	0	0	0	0
16	16	spam	spam	0.310	0.690	xxxmobilemoviecl	0	0	0	0
17	17	ham	ham	0.793	0.207	watching	0	0	0	0
18	18	ham	ham	0.778	0.222	remember spell r	0	0	0	0
19	19	ham	ham	0.777	0.223	fine thatâ feel that	0	0	0	0
20	20	spam	spam	0.311	0.689	england macedor	0	0	0	0
21	21	ham	ham	0.771	0.229	seriously spell na	0	0	0	0

Slika 52. Tačnost razvrstavanja teksta kao SPAM ili HAM

Word vector koji je formiran ima 6.930 jedinstvenih reči pa stoga možemo reći da ovaj vektor ima toliko dimenzija.

Row No.	ID	SPAM	prediction(S...	confidence(ham)	confidence(spam)	text	aah	aaoooooight	aathi	abbey
1	1	ham	ham	0.771	0.229	jurong point crazy	0	0	0	0

Slika 53 Dimenzije word vector-a

Confusion Matrix se često koristi u opisivanju performansi modela klasifikacije. Ova matrica prikazuje mnogo detaljnije raspored tačnih i netačnih klasifikacija za svaku klasu. Redovi matrice predstavljaju atribut dok kolone predstavljaju predviđanja. Izgled matrice i značenje podataka je veoma jednostavan:

- A – broj tačnih predviđanja negativnih rezultata
- B – broj netačnih predviđanja pozitivnih rezultata
- C – broj netačnih predviđanja negativnih rezultata
- D – broj tačnih predviđanja pozitivnih rezultata

		Predviđanje	
		Tačan	Netačan
Stvarno stanje	Pozitivno	D	B
	Negativno	A	C

Ovu matricu Rapidminer prikazuje i tabelarno i grafički.

	true ham	true spam	class precision
pred. ham	4773	170	96.56%
pred. spam	2	575	99.65%
class recall	99.96%	77.18%	

accuracy: 96.88% +/- 0.95% (mikro: 96.88%)

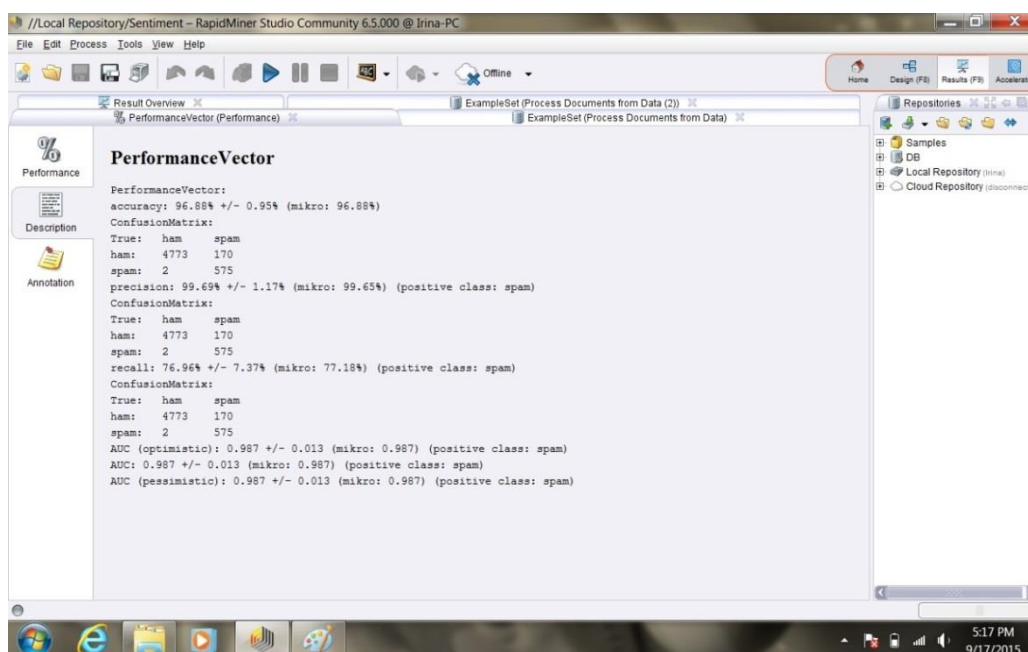
Slika 54. Tabelarni prikaz Confusion Matrix-a za naš model

Sa ove slike možemo videti da je naš model u slučaju da je poruka HAM klasifikovao tačno 4.773 poruka u HAM.

U dva slučaja je klasifikovao poruku kao SPAM a u stvari je ona normalna HAM poruka.

Od poruka koje su stvarno SPAM poruke model je klasifikovao 170 kao HAM, dok je 575 poruka koje su stvarno SPAM klasifikovao kao SPAM.

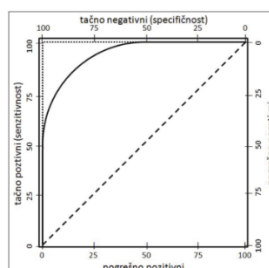
Tačnost (Accuracy) modela je 96,88 +/- 0,95.



Slika 55. Tekstualni prikaz Confusion Matrix-a za naš model

Sledeća pregled koji nam daje Rapidminer je tzv. ROC (Receiver Operating Characteristic) kriva.

ROC kriva je grafički prikaz senzitivnosti i specifičnosti za svaki mogući granični skor (rezultat na testu) u koordinatnom sistemu gde su na ordinati (y) prikazane vrednosti senzitivnosti (proporcija tačnih pozitivnih), a na apscisi (x) vrednosti specifičnosti oduzete od 1 (1 - specifičnost)², čime se dobija proporcija falš pozitivnih rezultata [28].

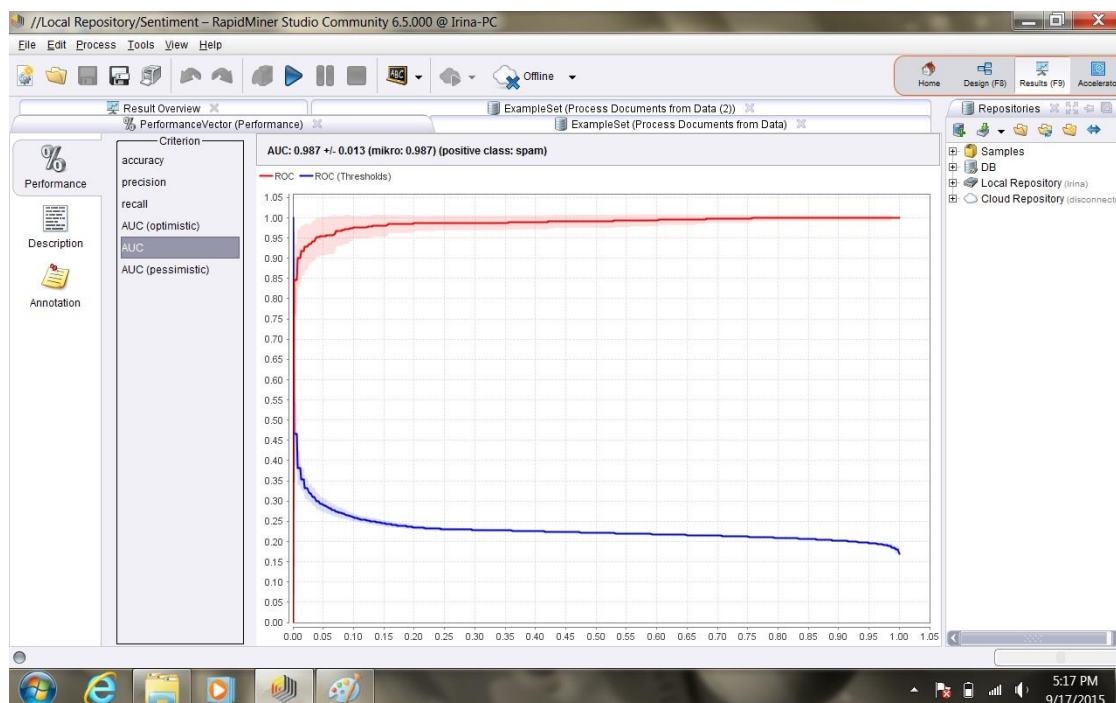


Slika 56. Primer ROC krive [28]

Ako je pojava koju merimo takva da se oni koji poseduju kvalitet i oni koji ga ne poseduju uopšte ne, onda naš test ima šanse 50:50% da slučajno “pogodi” koja je poruka SPAM, i koji god granični skor na testu da uzmemo, proporcija ispravno klasifikovanih ostaje ista. Ova situacija bi na ROC grafiku bila predstavljena dijagonalnom linijom koja spaja donji levi i gornji desni ugao, odnosno dve nulte tačke sa Slike 55. Ova dijagonala se obično zove dijagonala slučajnog ishoda (engl. chance diagonal) i prikazana je isprekidanom linijom na Slici 55 [28].

Sastavni deo ROC analize je pojam površina ispod ROC krive (Area under the Curve: AUC). Interpretacija ove površine, koja se najčešće koristi, je da je AUC pokazatelj verovatnoće da će na osnovu testa viši skor za slučajno odabranog ispitanika imati osoba sa datim kvalitetom, nego osoba bez kvaliteta [29].

Rapidminer je za naš model napravio sledeću ROC-krivu:



Slika 57. ROC kriva za naš model

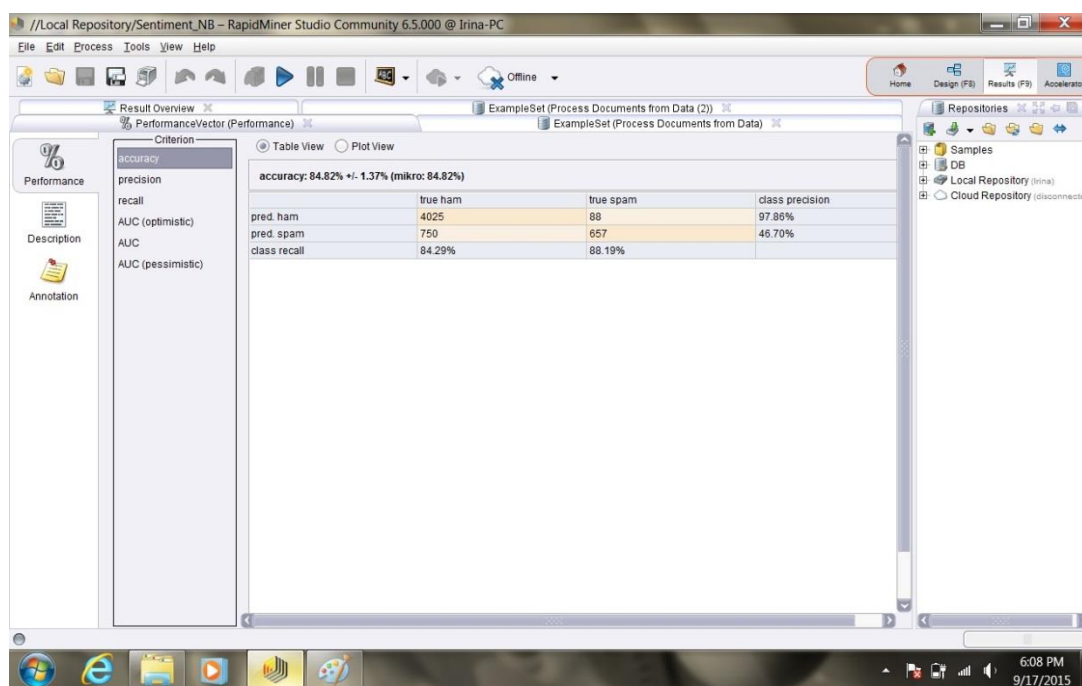
Na grafiku se vidi da kriva kreće iz donjeg levog ugla gde su granični skorovi viši, senzitivnost niska, a specifičnost visoka. Kako idemo ka gornjem levom uglu, granični skorovi su sve niži, senzitivnost sve viša, a specifičnost sve niža. Pošto je površina ispod krive velika, možemo reći da naš model dobro klasifikuje poruke kao HAM ili SPAM.

Radi testiranja različitih modela klasifikacije primenjeno je još dva različita modela sa klasifikaciju.

-Naïve Bayes

-AdaBoost

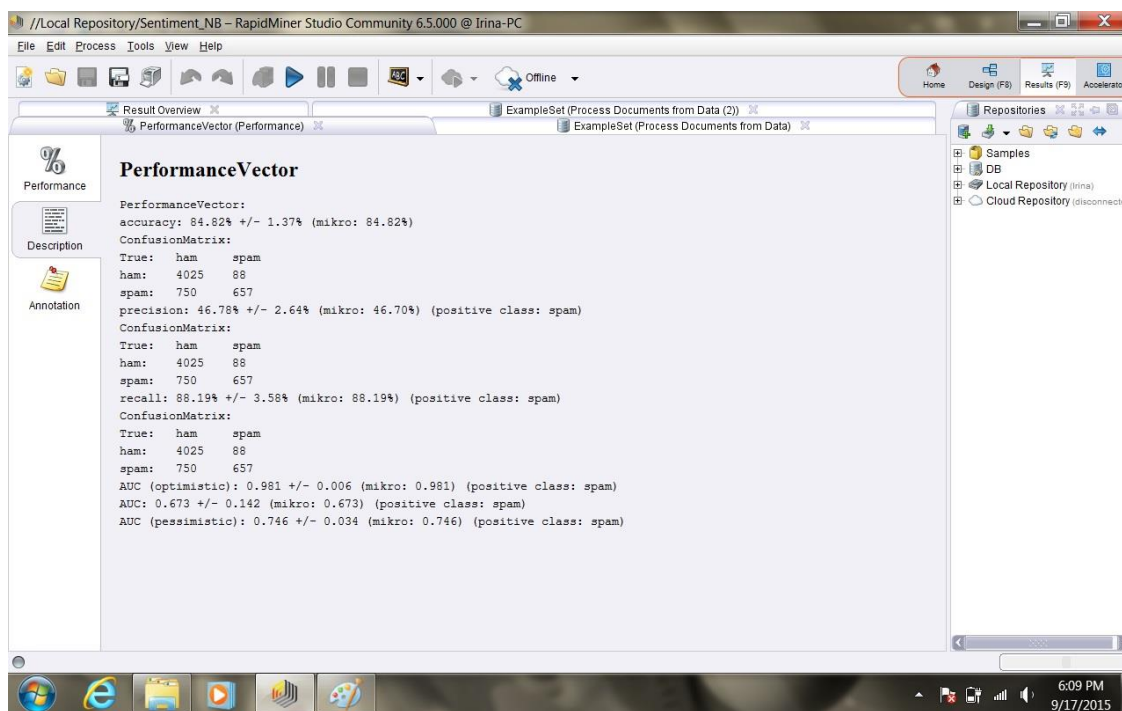
Kad smo koristili Naive Bayes model dobijeni su sledeći rezultati:



Slika 58 Performance vector za Naive Bayes model klasifikacije

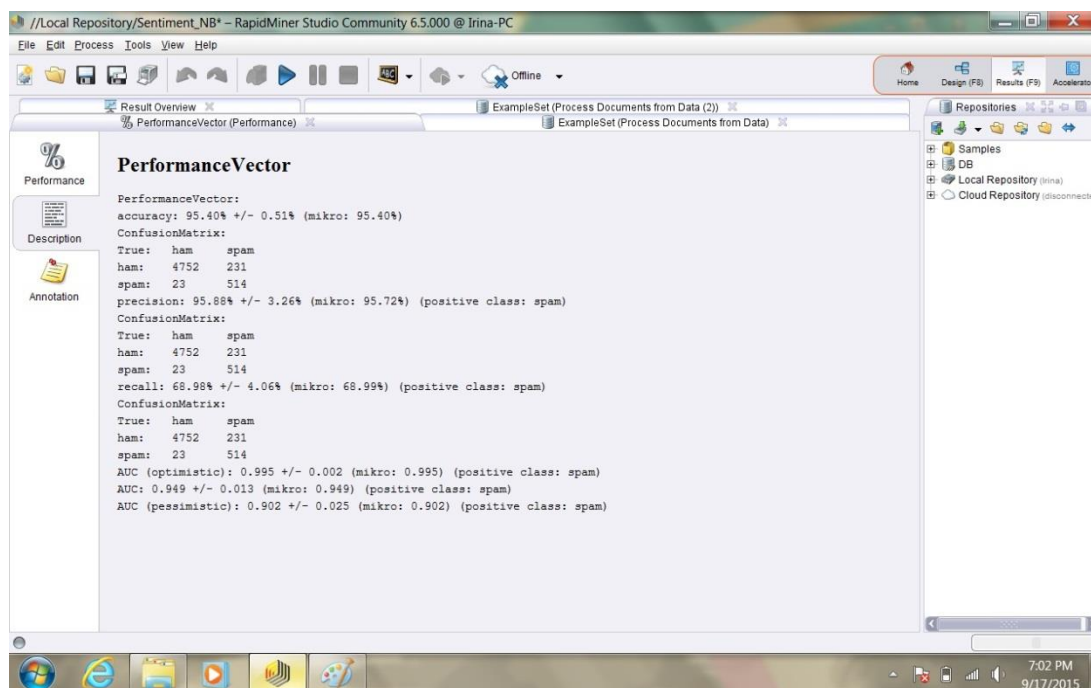


Slika 59 ROC kriva za Naive Bayes model klasifikacije

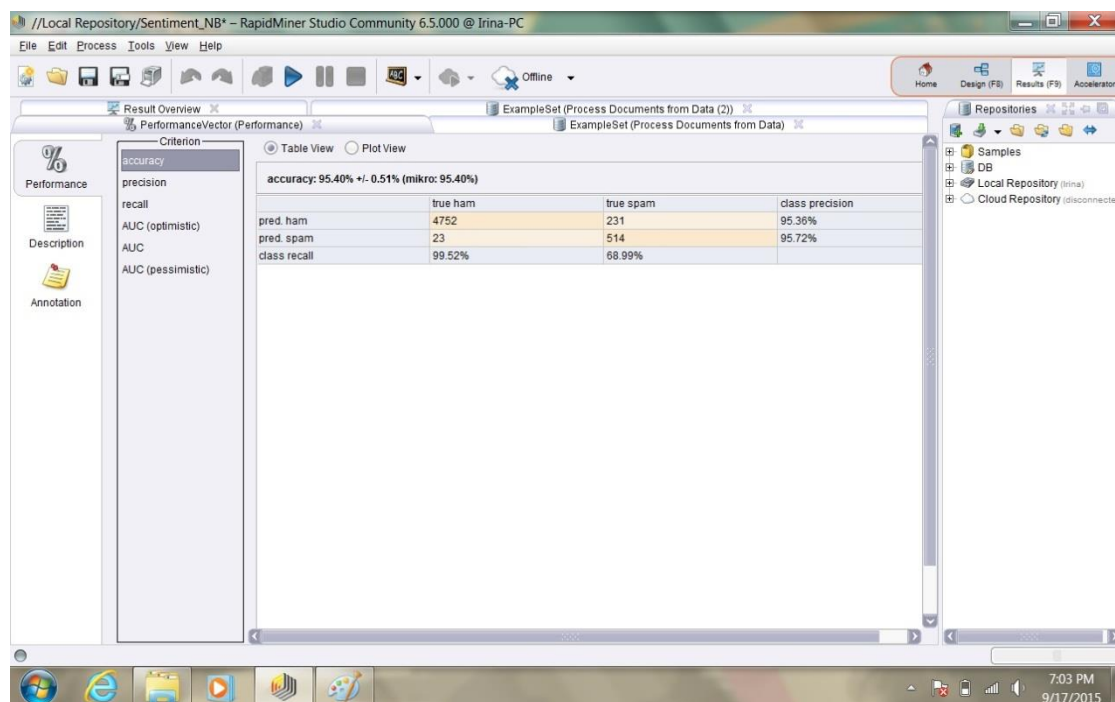


Slika 60 Confusion Matrix za Naive Bayes model klasifikacije

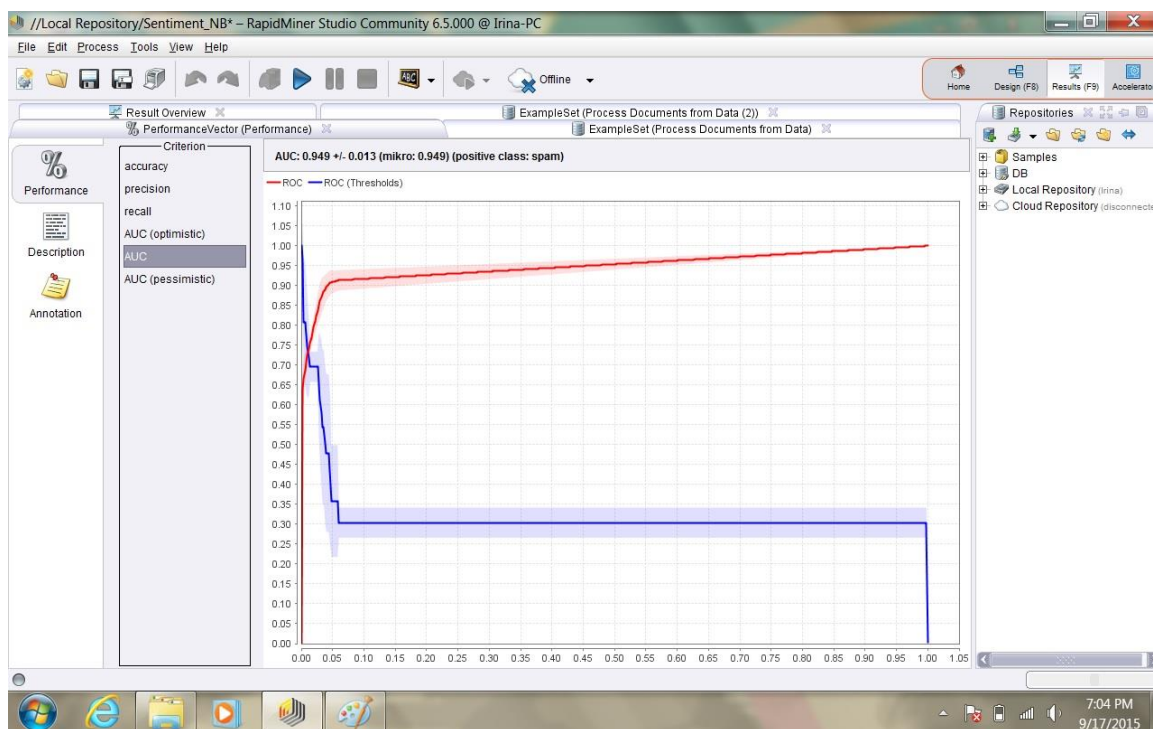
Model kalsifikacije AdaBoost je dao sledeće rezultate:



Slika 61 Confusion Matrix za AdaBoost model klasifikacije



Slika 62 Performance vector za AdaBoost model klasifikacije



Slika 63 ROC kriva za AdaBoost model klasifikacije

Rezultate našeg istraživanja ćemo uporediti sa rezultatima koje je dobio Houshmand Shirani-Meht u svom radu SMS Spam Detection using Machine learning Approach [30]. On je koristio isti skup podataka kao i onaj koji je korišćen u ovom radu. U tom radu je korišćeno više metoda koji su dali različite rezultate [30]:

Model	Spam caught (SC) %	Blocked Hams (BH)%	Accuracy %
	Prepoznati Spamovi	Blokirani Hamovi	Tačnost
Multinomial Naive Bayes	94,47	0,51	98,88
Support Vector Machine	92,99	0,31	98,86
k-nearest neighbor	82,60	0,40	97,47
Random Forest	90,62	0,29	98,57
Adaboost	92,17	0,51	98,59

Iz rezultata tog istraživanja dolazi se do zaključka da Support Vector Machine model daje najbolje rezultate.

Vrednosti iz tabele dobijamo iz Confusion Matrix – a na sledeći način:

$$SC = \frac{\text{Ukupno SPAM klasifikacija}}{\text{Ukupan broj SPAM poruka}} \quad [30]$$

Procenat prepoznatih spamova sa našim modelom:

$$SC = \frac{575}{575+170} = 77,18 \%$$

$$BH = \frac{\text{Ukupno HAM klasifikacija}}{\text{Ukupan broj HAM poruka}} \quad [30]$$

Procenat blokiranih HAM poruka kod našeg modela

$$BH = \frac{2}{4,773+2} = 0,04 \%$$

Kad ovo uporedimo sa našim rezultatima:

Model	Spam caught (SC) %	Blocked Hams (BH)%	Accuracy %
	Prepoznati Spamovi	Blokirani Hamovi	Tačnost
SVM–Houshmand Shirani	92,99	0,31	98,86
SVM – naš model	77,18	0,04	96,88
Multinomial Naive Bayes – Houshmand Shirani	94,47	0,51	98,88
Multinomial Naive Bayes – naš model	88,18	15,70	84,82
Adaboost-Houshmand Shirani	92,17	0,51	98,59
Adaboost – naš model	68,99	0,48	95,40

Ako uporedimo rezultate koje smo dobili sa rezultatima prikazanog u radu Houshmand Shirani, vidimo da naš model slabije klasifikuje SPAM poruke po svim modelima kalsifikacije, dok se HAM poruke veoma dobro klasifikuju, kod nekih modela i bolje nego kod uporednih rezulata.

Naš model Multinomial Naive Bayes daje najbolje rezultate u klasifikaciji SPAM poruka ali veoma loše klasifikuje HAM poruke.

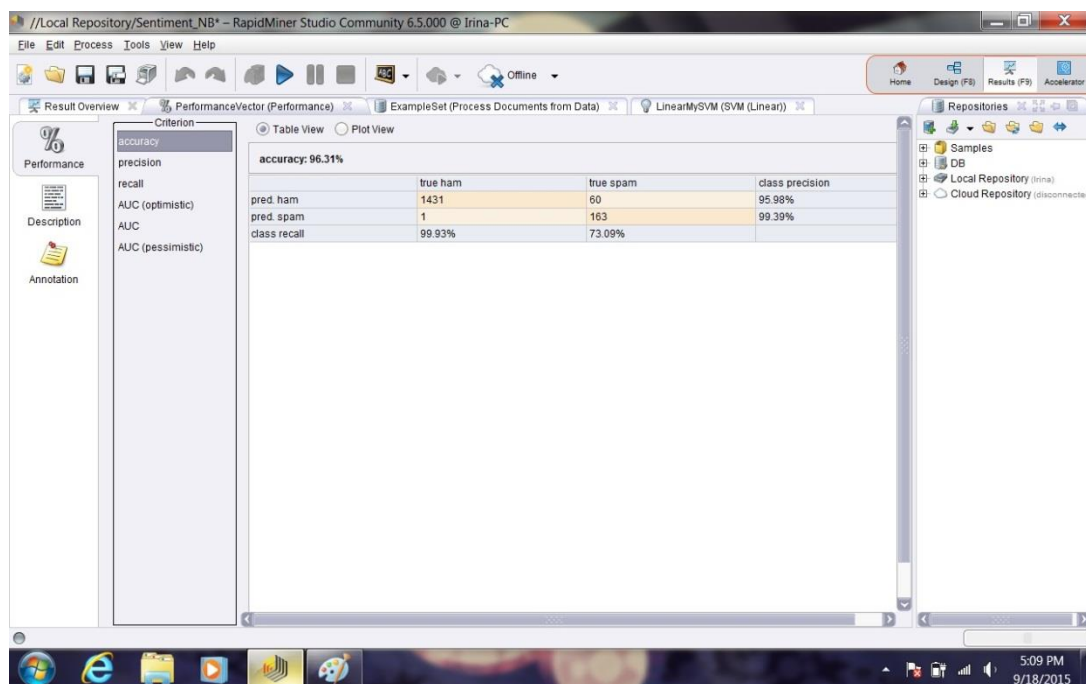
Support Vector Machine model je loše prepoznavao SPAM poruke, ali je skoro svaku HAM poruku tačno klasifikovao, tako da generalno gledano za ovaj model možemo reći da se pokazao kao najbolji klasifikator poruka u slučaju našeg modela za klasifikaciju SPAM poruka.

Na kraju je izvršen još jedan test. Umesto operatora *X-Validation* je iskorišćen operator *Split-Validation*. Karakteristika ovog operatora je da on deli skup podataka koji koristimo na skup za trening i skup za testiranje u odnosu koji definišemo u parametrima ovog operatora.

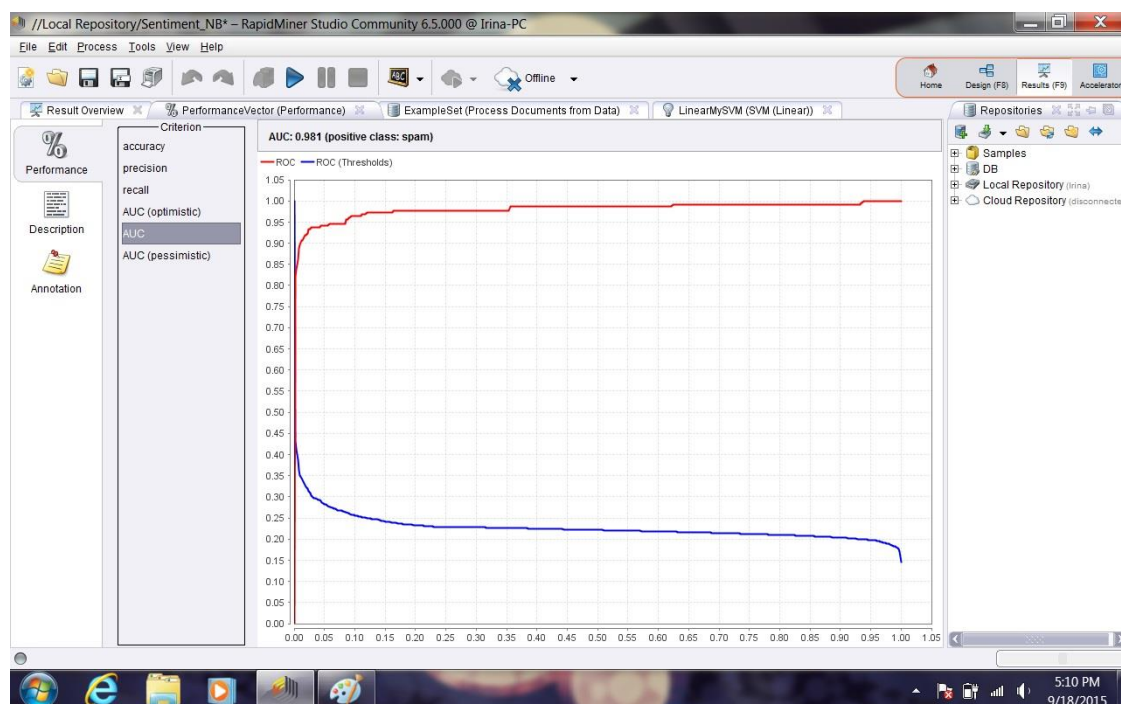
Testirani su odnosi:

- 50 : 50
- 60 : 40
- 70 : 30
- 80 : 20
- 90 : 10

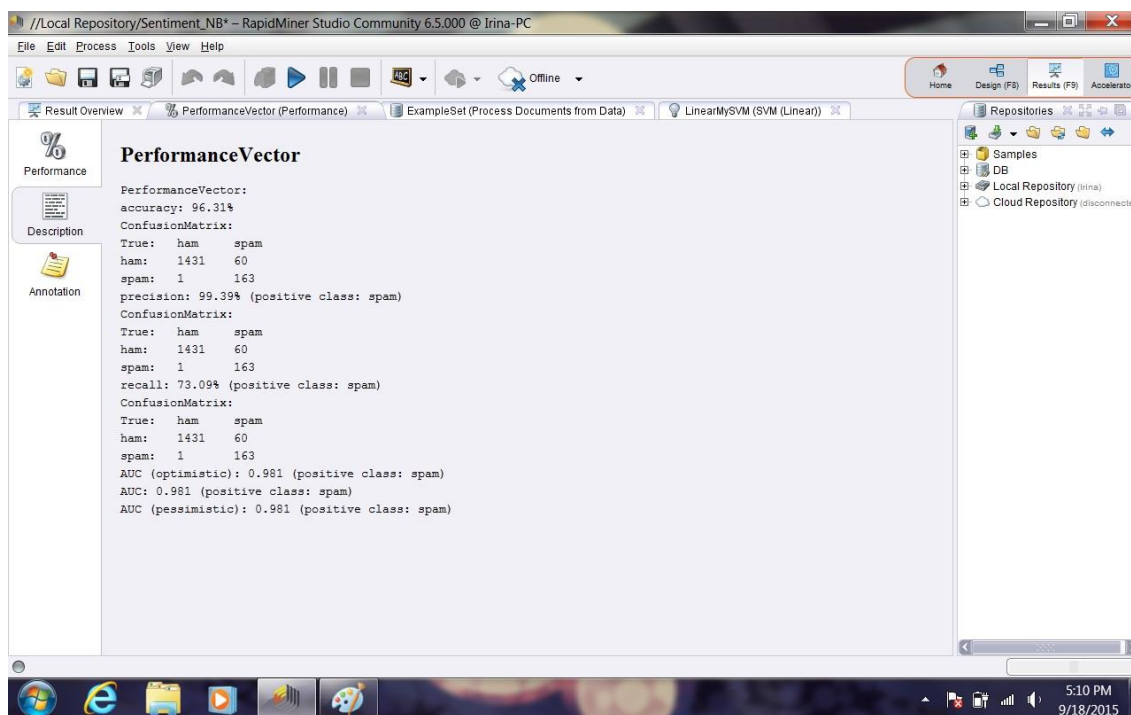
Najbolji rezultat je postignut kod deljenja skupa podataka u odnosu 70:30.



Slika 64 Performance Vector sa operatorom Split-Validation



Slika 65 ROC kriva sa operatorom Split-Validation



Slika 66 Confusion Matrix sa operatorom Split-Validation

Model	Spam caught (SC) %	Blocked Hams (BH)%	Accuracy %
Operator za Cross Validation	Prepoznati Spamovi	Blokirani Hamovi	Tačnost
SVM – X-Validation	77,18	0,04	96,88
SVM – Split Validation	73,09	0,07	96,31

Možemo zaključiti da je naš model dao bolji rezultat kad smo koristili eksterni skup podataka za testiranje na koji smo primenili naš model, nego kad je izvorni skup deljen na skup za trening i skup za tesiranje.

5. Zaključak

Priliv podataka nikad nije bio veći, tako da je potreba za istraživanjem podataka sve veći i veći. Ove tehnike nam pružaju dosad neslućene uvide u podatke omogućavajući nam bolje donošenje odluka i predviđanje.

Jedan od najmoćnijih i najfleksibilnijih alata za istraživanje podataka je Rapidminer.

Rapidminer pokriva sve relevantne korake u procesu istraživanja podataka, od učitavanja podataka, transformacije, integracije, agregacije i vizualizacije do automatizovanog izbora funkcija, parametara i optimizacije procesa. Integracija sa ostalim sličnim alatima kao što su WEKA, R i slični je veoma laka.

Samo kreiranje modela za istraživanje podataka je veoma pojednostavljeno, tako da se ceo proces sastoji od biranja operatora i njihovo prevlačenje na radnu površinu, gde se zatim spajaju ulazni i izlazni portovi u zavisnosti od podataka sa kojim radimo. Parametri za operatore se mogu automatski učitavati iz ranijih projekata, što dodatno omogućuje rad u ovom softverskom alatu.

Pored mnogo operatora kojim raspolaže Rapidminer je veoma moćan i na polju grafičkog predstavljanja rezultata. Postoje mnogo vrsta grafika, sa puno parametara koje možemo promeniti za bolji prikaz rezultata.

Algoritmi koje koristi rapidminer su optimizovani za što bržu obradu podataka. Ovo se naročito primećuje kod rada sa velikim skupovima podataka.

Jedna od moćnih karakteristika je i tzv. *preprocessing* podataka. Postoji veliki broj operatora za pripremu podataka, zatim za njihovu transformaciju i agregaciju. Sve to omogućuje brži rad sa velikim skupovima podataka.

Dokumentacija koja postoji za rad se ovim alatom je veoma detaljna i omogućuje brzo i lako savladavanje ovog alata. Takođe postoji veoma dobra podrška korisnicima i brzo im se izlazi u susret u slučaju nekog problema.

Veoma je bitno da ovaj softver radi i na sporijim računarima i ne zahteva neke specijalne instalacije. Naravno za velike skupove podataka potrebno je da imamo računar sa mnogo RAM memorije i što više procesora.

Kao i sve moderni softverski alati Rapiminer nudi i rad u "oblacima", gde možemo čuvati podatke i naše modele.

Na kraju se mora dodati da Rapidminer nije savršen alat i ne može se primeniti u svim situacijama. Kako je istraživanje podataka specifična grana nauke, stručnjaci koji se bave istraživanjem podataka bi trebalo da imaju više različitih softverskih alata, da bi mogli da uporede rezultate koje dobijaju sa njima i da izaberu najadekvatniji alat za svoje istraživanje.

Literatura

- [1] Data Mining-introductory and advanced topics, Margaret H. Dunham
- [2] <http://oliver.efri.hr/~statist/Regresija%20i%20korelacija.pdf>
- [3] <http://www.biochemia-medica.com/content/procjena-analitickih-znacajki-reagensa-randox-full-range-za-odredjivanje-koncentracije-c-reak>
- [4] <http://blog.minitab.com/blog/adventures-in-statistics/what-is-the-difference-between-linear-and-nonlinear-equations-in-regression-analysis>
- [5] http://www.tutorialspoint.com/data_mining/dm_classification_prediction.htm
- [6] <http://quantup.eu/Szkolenia/data-mining-in-practise-with-r/>
- [7] <http://www.rdatamining.com/examples/time-series-analysis>
- [8] <http://machinelearningmastery.com/market-basket-analysis-with-association-rule-learning/>
- [9] <http://ceur-ws.org/Vol-39/paper3.pdf>
- [10] Data mining and knowledge discovery handbook, Oded Maimon & Lior Rokach
- [11] Tom Mitchel , Machine learning, 1997
- [12] <http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture11.html> , Simon Colton , “My Weekend” Example
- [13] Sreerama Murthy, Decision Tree:A multi-disciplinary survey, 1998,
<http://cmappublic3.ihmc.us/rid=1MVPFT7ZQ-15Z1DTZ-14TG/Murthy%201998%20DMKD%20Automatic%20Construction%20of%20Decision%20Trees.pdf>
- [14] <http://docs.rapidminer.com/studio/getting-started/ui-overview.html>
- [15] <http://docs.rapidminer.com/studio/getting-started/1-importing-data.html>
- [16] <http://docs.rapidminer.com/studio/getting-started/3-creating-model.html>
- [17] <http://docs.rapidminer.com/studio/operators/>
- [18] <http://docs.rapidminer.com/studio/getting-started/important-terms.html#port-info>

- [19] <https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection>
- [20] <http://www.scribd.com/doc/260608553/An-Analysis-of-Various-Algorithms-For-Text-Spam-Classification-and-Clustering-Using-RapidMiner-and-Weka>
- [21] <http://infoteh.etf.unssa.rs.ba/zbornik/2012/radovi/RSS-7/RSS-7-8.pdf>
- [22] <http://www.ftn.kg.ac.rs/konferencije/rppo13/Monografije/MONOGRAFIJA%20BR%204%20RESAVANJE%20KLASIFIKACIONIH%20PROBLEMA%20MASINSKOG%20UCENJA.pdf>
- [23] <http://dsp.etfbl.net/students/maric.pdf>
- [24] [http://www.google.rs/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=2&cad=rja&uact=8&ved=0CCIQFjABahUKEwj37Njw8NzHAhVLWxoKHXYqDG4&url=http%3A%2F%2Fhome.etf.rs%2F~vm%2Fos%2Fdmsw%2F2014%2FJovan%2520Markovic\(3227-2014\).docx&usg=AFQjCNEIbZcbDUmp6PJlibxOVLW1p6DwqQ&bvm=bv.101800829,b.s.1,d.bGg](http://www.google.rs/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=2&cad=rja&uact=8&ved=0CCIQFjABahUKEwj37Njw8NzHAhVLWxoKHXYqDG4&url=http%3A%2F%2Fhome.etf.rs%2F~vm%2Fos%2Fdmsw%2F2014%2FJovan%2520Markovic(3227-2014).docx&usg=AFQjCNEIbZcbDUmp6PJlibxOVLW1p6DwqQ&bvm=bv.101800829,b.s.1,d.bGg)
- [25] <http://archive.ics.uci.edu/ml/machine-learning-databases/00228/>
- [26] https://en.wikipedia.org/wiki/Bag-of-words_model
- [27] <http://www.tfidf.com/>
- [28] Wojtek J. Krzanowski and David J. Hand: ROC curves for continuous data
- [29] <http://psihologija.ff.uns.ac.rs/primenjena/clanci/20114335.pdf>
- [30] <http://cs229.stanford.edu/proj2013/ShiraniMehr-SMSSpamDetectionUsingMachineLearningApproach.pdf>