Postupak optimiziranog izbora oglasa na temelju stabala odlučivanja

Babić, Dominik

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:633437

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Diplomski studij računarstva

POSTUPAK OPTIMIZIRANJA IZBORA OGLASA NA TEMELJU STABALA ODLUČIVANJA

Diplomski rad

Dominik Babić

Osijek, 2016.

Sadržaj

1. UVOD 1
1.1. Zadatak diplomskog rada 1
2. INTERNET OGLAŠAVANJE 2
2.1. Osnovni pojmovi i koncepti Internet oglašavanja 3
2.1.1. Koncepti oglašavanja 3
2.2. Prikazno Internet oglašavanje (*eng. Display Advertising*) 4
2.3. Monetizacijski modeli 6
2.4. Važnost podataka u Internet oglašavanju 7
3. SUSTAVI ZA ISPORUKU OGLASA 8
3.1. Oglašivač, kampanja i oglas 8
3.2. Izdavač, stranica i oglasno mjesto 9
3.3. Uobičajene funkcionalnosti 10
3.4. Tehnička izvedba sustava za isporuku oglasa 11
4. METODOLOGIJA 13
4.1. Skup podataka za analizu 13
4.2. Stabla odlučivanja 16
4.2.1. GINI Indeks 18
4.2.2. Entropija 18
4.2.3. Greška klasifikacije 18
4.3. Priprema podataka za treniranje stabala 19
4.4. Postupak treniranja stabla 22
5. REZULTATI 24
5.1. *Decision Tree* učeći operator 24
5.2. *Decision Stump* učeći operator 30
5.3. *Random Forest* učeći operator 33
5.4. *Stacking, Vote, Gradient Boosted Trees* i *Forward Selection* 37
6. ZAKLJUČAK 41
LITERATURA 42

SAŽETAK 43
SUMMARY 44
ŽIVOTOPIS 45

1. UVOD

Krajem 90-ih godina prošloga stoljeća Internet oglašavanje postalo je osnovno sredstvo mrežnog marketinga. U tu svrhu kreirani su specijalizirani sustavi za oglašavanje čija je osnovna zadaća odabir i prikaz oglasa na Internet stranicama. Kako se monetizacija oglašavanja temelji na broju klikova oglasa, potrebno je prikaz oglasa optimizirati tako da se maksimizira učinkovitost pojedine marketinške kampanje.

Kako bi se prikaz oglasa optimizirao potrebno je provesti trening algoritma strojnog učenja koji će za svakog potencijalnog potrošača provesti odabir prikladnog oglasa. Podatkovni skup sadrži informacije o potrošačima po specifičnim atributima na temelju kojih se može generirati stablo odlučivanja. Neki od algoritama koji se mogu primijeniti za generiranje stabla odlučivanja su *C4.5, CART* (*Classification and Regression Trees*), *MARS* (*Multivariate Adaptive Regression Splines*) i drugi. Potrebno je osigurati da se treniranjem algoritma postigne visoka preciznost klasifikacije ne samo na podatkovnom skupu kojim se vršilo testiranje, već i na podacima koji se prikupljaju u stvarnom vremenu.

No, prije samog treniranja potrebno je obraditi podatke tako da sadrži isključivo korisne informacije. Filtriranjem podataka moguće je ukloniti one atribute koji ne nose korisne informacije.

U okviru diplomskog rada potrebno je korištenjem stabala odlučivanja optimizirati odabir oglasa na temelju prikupljenih podataka o posjetitelju. Prikupljeni podaci sadrže atribute na temelju kojih se generira stablo odlučivanja. No, prije generiranja stabla potrebno je podatke pripremiti, a potom provesti treniranje algoritma strojnog učenja.

1.1. Zadatak diplomskog rada

Na temelju prikupljenih podataka o posjetiteljima, potrebno je napraviti prediktor izbora kampanje oglašavanja takve da se maksimizira omjer klikova i impresija (*eng. Click-through Rate, CTR*). Podatke je potrebno pripremiti te odabrati oglas kao objekt predikcije.

2. INTERNET OGLAŠAVANJE

Internet oglašavanje je oblik marketinga i oglašavanja koji koristi Internet za dostavu promotivnih poruka potrošačima. Slično drugim oblicima oglašavanja, Internet oglašavanje uključuje izdavača (*eng. publisher*) koji integrira oglase u svoj sadržaj te oglašavača (*eng. advertiser*) koji pruža oglase. Neki od tipova Internet oglašavanja uključuju oglašavanje elektroničkom poštom, oglašavanje putem tražilice, oglašavanje putem društvenih medija te mnogi drugi oblici.

U svrhu Internet oglašavanja primjenjuju se poslužitelji oglasa (*eng. Ad Server*) koji, osim što dostavljaju oglase, prikupljaju statističke podatke o oglašavanju. Poslužitelji oglasa su specijalizirani poslužitelji čija je svrha pohranjivanje promotivnog sadržaja te dostavljanje istog putem raznih digitalnih platformi. Detaljnije u poglavlju *3*.

Samo prikazivanje oglasa se iskazuje u impresijama (*eng. Impression*). Svaki put kada se oglas prikaže korisniku, to se računa kao impresija. Broj prikaza oglasa iliti impresije su često mjera na temelju koje se naplaćuje oglašavanje. Alternativno, oglašavanje se može naplaćivati po kliku oglasa. U tom slučaju definira se CPC (*eng. Cost Per Click*), novčani iznos koji oglašavač mora platiti za svaki klik oglasa. Detaljnije u poglavlju o monetizaciji. Uspješnost pojedine kampanje se izražava omjerom klikova oglasa i prikaza oglasa, *CTR* (*eng. Click Through Rate*), i definira koliko puta se nakon prikaza oglasa kliknulo na isti. Ujedno, *CTR* predstavlja svojevrsnu stopu zainteresiranosti posjetitelja za proizvod. Svaki put kada korisnik nakon klika oglasa i posjeta stranici oglašivača izvrši kupovinu, registraciju ili preuzimanje sadržaja, računa se konverzija (*eng. Conversion*). Konverzija podrazumijeva učinkovitost kojom marketinška kampanja prikazivanjem oglasa generira klijente tj. vrši konverziju potencijalnih klijenata u stvarne klijente.

Oglašavanje putem Interneta uključuje nekoliko različitih metoda opisanih u potpoglavljima 2.1. i 2.2.

2.1. Osnovni pojmovi i koncepti Internet oglašavanja

Od pojave prvog oblika Internet oglašavanja pa do danas razvile su se brojne metode na kojima se potencijalnim potrošačima mogu prikazivati oglasi. Time se stvorila potreba za standardiziranjem pojedinih oblika oglašavanja. Ujedno se nastoji potrošačima prezentirati oglase tako da se ne izazove osjećaj prisilne interakcije. U nastavku su opisane osnovne metode Internet oglašavanja.

2.1.1. Koncepti oglašavanja

Osnovni oblici Internet oglašavanja uključuju oglašavanje elektroničkom poštom, prikazno oglašavanje (*eng. Display Advertising)*, međuprostorne (*eng. Interstitial*) oglase, oglašavanje putem tražilice (*eng. Search Engine Marketing*), oglašavanje putem društvenih medija, oglašavanje putem mobilnih uređaja, *adware*, *affiliate marketing* i drugi[1].

Oglašavanje elektroničkom poštom je opće poznati primjer Internet oglašavanja i zasniva se na slanju promotivnih poruka putem elektroničke pošte grupi trenutnih i potencijalnih potrošača. Mnoge organizacije koriste *email* oglašavanje za obavještavanje svojih potrošača o novim proizvodima i ponudama. *Email* oglašavanje se često označava kao neželjena pošta što velikim dijelom ovisi o sadržaju same elektroničke pošte. Nešto što izgleda kao uobičajen oglas može zaraziti korisnikovo računalo s ciljem povećanja broja računala koja šalju neželjenu poštu. Unatoč lošim konotacijama, oglašavanje elektroničkom poštom i dalje ostaje popularan izbor.

Prikazno oglašavanje je oblik oglašavanja koji uključuje različite multimedijske formate za prenošenje poruke potrošačima. Postoji više tipova oglasa koji spadaju pod prikazno oglašavanje, detaljnije u poglavlju *2.2*.

Međuprostorni oglasi se pojavljuju prije nego korisnik dobije pristup sadržaju Internet stranice. Korisnik je potom prisiljen čekati određeni vremenski period prije pristupanja sadržaju. Međuprostorni oglasi obično sadrže tekstualne poveznice na stranicu oglašivača.

Oglašavanje putem tražilice je osmišljeno kako bi se povećala vidljivost određene stranice u rezultatima pretraživanja. Tražilice nude sponzorirane i nesponzorirane rezultate temeljene na upitu pretraživanja. Sponzorirani rezultati obično sadrže indikator kojim se identificiraju.

Oglašavanje putem društvenih mreža je oblik oglašavanja kojim organizacije promoviraju svoje proizvode na društvenim mrežama tako što učestalo objavljuju posebne ponude korisnicima društvenih mreža putem svojih profila.

Oglašavanje putem mobilnih uređaja se zasniva na prikazivanju multimedijskih oglasa kroz mobilne aplikacije ili Internet preglednike. *Mobile Marketing Association* nastoji standardizirati oglašavanje putem mobilnih uređaja definirajući dimenzije i podatkovnu veličinu oglasa[2].

Adware je programski paket čija je osnovna zadaća prikazivanje oglasa na korisnikovom računalu. Oglasi se mogu pojavljivati u samom programskom paketu ili se mogu integrirati u tražilice. *Adware* mora tražiti dopuštenje korisnika prilikom instalacije, u suprotnom se označava kao *malware*.

Affiliate marketing je oblik Internet oglašavanja u kojem oglašivač nagrađuje treću stranu za svakog korisnika, potrošača ili posjetitelja generiranog njihovim promoviranjem proizvoda oglašivača.

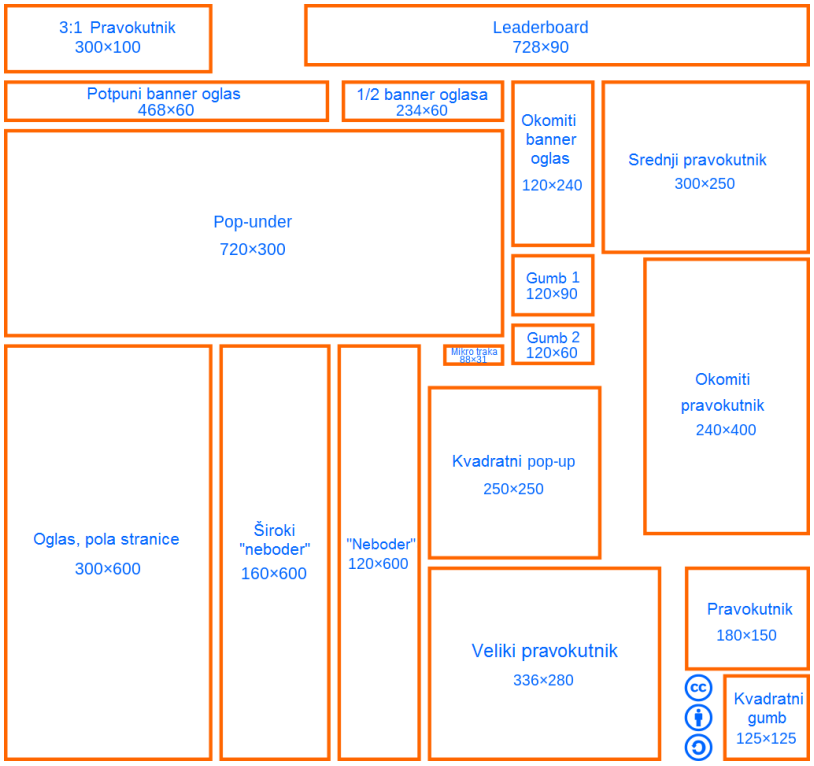
2.2. Prikazno Internet oglašavanje (*eng. Display Advertising*)

Prikazno oglašavanje je oblik oglašavanja koji uključuje različite multimedijske formate (npr. tekst, slika, video, animacija, …), ugrađene u Internet stranicu, za prenošenje poruke potrošačima. Oglašivači primjenom sustava za oglašavanje često prikupljaju podatke o pretraživanju pojedinog korisnika kako bi razlučili koji oglas ponuditi kojem korisniku. Prostor za oglašavanje na stranici se plaća po nekom od modela opisanih u poglavlju 2.3. o monetizaciji.

Postoji nekoliko tipova oglasa koji spadaju u kategoriju prikaznog oglašavanja, neki od njih su[3]:

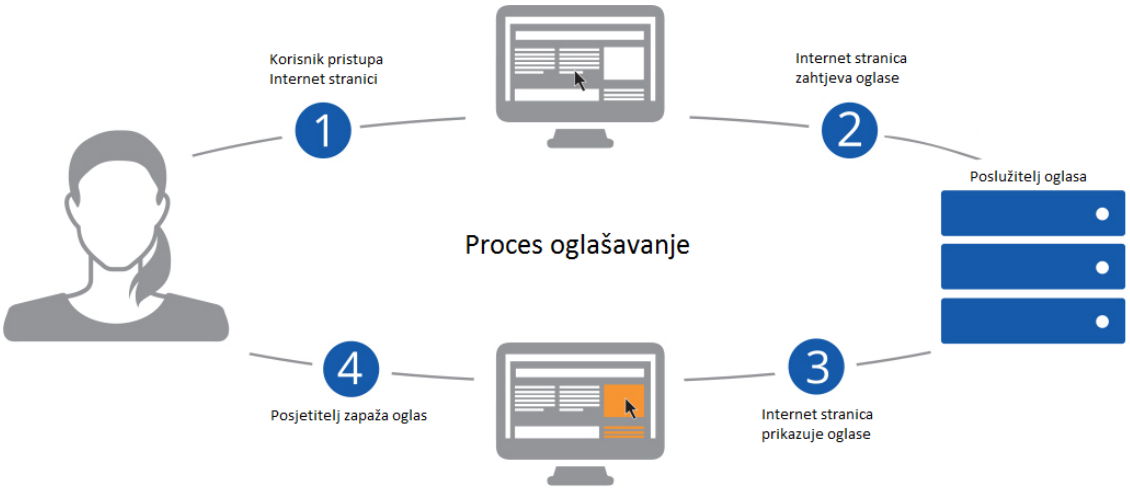
- *Banner* oglasi – grafički oglasi prikazani na Internet stranici, koriste multimediju za prikaz oglasa,
- *Frame ad* – prvi oblik *banner* oglasa, izdavači Internet stranica odvajaju prostor na stranici za oglase, točne dimenzije oglasa su standardizirane, detaljnije u nastavku,
- *Pop-up / Pop-under* – oglas se otvara u novom prozoru preglednika, iznad ili ispod postojećeg prozora preglednika, preporučuje se izbjegavanje ovog oblika oglašavanja,
- *Floating ad* – nameću se na sadržaj stranice time ga zaklanjajući i onemogućujući pregledavanje, korisnik ih uklanja sam ili oglas nestane nakon određenog vremena.

Interactive Advertising Bureau, organizacija zadužena za istraživanje i standardiziranje Internet oglašavanja, u svojim smjernicama[4] definira dimenzije za prikazno oglašavanje (Slika *2.1.*), maksimalnu veličinu oglasa u bajtovima, način implementacije i mnoge druge specifikacije.



Sl. 2.1. *IAB dimenzije oglasa, širina x visina*

Sam proces prikazivanja oglasa se odvija kroz nekoliko faza prikazanih na slici *2.2*.



Sl. 2.2. *Proces oglašavanja*

2.3. Monetizacijski modeli

Izdavači Internet stranica se često odlučuju na iznajmljivanje prostora za oglašavanje. Neovisno o odabranom monetizacijskom modelu oglašivač isporuku oglasa putem izdavačevih stranica plaća. Sama novčana kompenzacija može biti zasnovana na broju prikaza oglasa, broju klikova oglasa ili prodaji proizvoda tj. usluga[5].

CPM (*eng. Cost Per Mille*) modelom se usluge oglašavanja naplaćuju za svakih 1000 impresija. *CPM* je moguće računati prema izrazu (*2-1*):

$$CPM = \frac{Trošak\ oglašavanja}{Generirane\ impresije} \times 1000 \tag{2-1}$$

CPM se izražava u novčanim jedinicama i služi kao mjera troška kampanje oglašavanja u odnosu na broj generiranih impresija. *CPM* je standardni monetizacijski model u oglašavanju.

Alternativno, naknada oglašavanja se može naplaćivati po broju klikova oglasa. Jedina razlika u odnosu na *CPM* je ta što *CPC* (*eng. Cost Per Click*) model naplaćuje svaki klik oglasa umjesto prikaza istih. Samim time, prikazivanje oglasa ne generira prihod izdavaču tj. trošak oglašivaču. Ekvivalent *CPC* modela se može računati u odnosu na *CPM* model sa specifičnim *CTR-om* prema izrazu (*2-2*):

$$CPC = \frac{CPM}{1000 \times CTR} \tag{2-2}$$

CPC se izražava u novčanim jedinicama po kliku oglasa. Postoje dva tipa *CPC* modela – model s fiksnom stopom (*eng. Flat-rate CPC*) i model zasnovan na ponudama (*eng. Bid-based CPC*). U slučaju modela s fiksnom stopom oglašivač i izdavač dogovaraju fiksnu svotu novca koja se mora platiti za svaki klik oglasa. Model zasnovan na ponudama sukobljava različite oglašivače koji izdavača obavještavaju o maksimalnom iznosu koji su voljni platiti za klik oglasa.

Uz spomenute modele pojavljuje se i *CPA* (*eng. Cost Per Acquisition*) model koji mjeri trošak konverzije posjetitelja u potrošača tj. trošak oglašavanja u odnosu na broj narudžbi. Izdavači ostvaruju novčanu dobit samo ako je prikazivanje oglasa rezultiralo kupovinom proizvoda ili usluge.

$$CPA = \frac{Trošak\ oglašavanja}{Broj\ narudžbi} \tag{2-3}$$

2.4. Važnost podataka u Internet oglašavanju

Prilikom pretraživanja Interneta korisnik posjećuje razne stranice koje kao mehaniku praćenja koriste kolačiće (*eng. Cookies*). Kolačići prvenstveno služe za ubrzanje pristupa stranici, javljajući poslužitelju da je korisnik već prije posjetio stranicu. Zahvaljujući ovoj mehanici, moguće je pohranjivati korisničke preferencije.

Slično Internet stranicama sustavi za oglašavanje koriste kolačiće kako bi mogli pratiti kojim oglasima su korisnici bili izloženi prilikom pregledavanja Internet stranica. Time poslužitelji oglasa prikupljaju podatke o oglašavanju i navikama posjetitelja tj. potrošača, sve to s ciljem povećanja efikasnosti oglašavanja.

Stvaranje podatkovnog skupa o oglašavanju omogućuje oglašivačima razumijevanje navika posjetitelja te je korištenje tih podataka ključ uspješnog oglašavanja. Prikupljanjem što većeg broja informacija o pojedinom korisniku moguće je stvoriti to vjerniju sliku korisnika. Analizom navika korisnika osigurava se veća preciznost oglašavanja, korisnicima se prikazuju samo oni oglasi koji su njima relevantni dok oglašivači dobivaju mogućnost stvaranja nove klijentske baze.

Kako su svi korisnici različiti analizom podataka osigurava se prikazivanje različitih oglasa različitim korisnicima što zauzvrat nudi povećanje broja klijenata.

3. SUSTAVI ZA ISPORUKU OGLASA

Sustavi za oglašavanje su specijalizirani poslužitelji koji pohranjuju oglase i dostavljaju ih posjetiteljima Internet stranica. Uz pohranjivanje i dostavljanje oglasa posjetiteljima poslužitelji oglasa prate statistiku pojedine kampanje prikupljanjem podataka o impresijama i klikovima oglasa. Time je moguće pratiti uspješnost pojedine marketinške kampanje[6].

Poslužitelji oglasa podržavaju različite pristupe ciljanom oglašavanju što omogućuje oglašivačima preciznije oglašavanje različitim tipovima korisnika. Neki od pristupa su:

- *Ad Targeting* – definira se kao sužavanje grupe korisnika kojima se pojedini oglasi prikazuju,
- *Behavioral Targeting* – vrši se nakon određivanja ciljane skupine korisnika, oglašavanje se zasniva na interesima koji su slični korisnikovim,
- *Ad Metric* – obavještava oglašivača o oglasima s niskim *CTR*-om, potiče na promjenu pristupa oglašavanju,
- *Geo Targeting* – metoda određivanja lokacije korisnika i dostavljanje oglasa na temelju iste.

Sami poslužitelji se dijele na udaljene i lokalne. Lokalni poslužitelji su obično u vlasništvu samog izdavača i služe za dostavu oglasa stranici izdavača. Time izdavač ima potpunu kontrolu nad sadržajem oglašavanja. Udaljeni poslužitelji su vlasništvo treće strane zadužene za oglašavanje na stranicama većeg broja izdavača.

3.1. Oglašivač, kampanja i oglas

U najužem smislu oglašivač je osoba ili organizacija koja prodaje proizvod ili uslugu. Oglašivači plaćaju izdavačima usluge oglašavanja po nekom od monetizacijskih modela. Oni profitiraju prodajom proizvoda ili usluge koju oglašava.

Oglašivač može samostalno vršiti oglašavanje pomoću sustava za posluživanje ili može angažirati organizaciju specijaliziranu za Internet oglašavanje – treća strana ili *affiliates*.

Angažiranjem treće strane oglašivač se obvezuje na isplatu dijela zarade od prodaje proizvoda ili usluge za svakog klijenta generiranog radom angažirane organizacije.

Promoviranje proizvoda ili usluga se smatra marketinškom kampanjom. Kampanja se može planirati s različitim ciljevima – promocija novog proizvoda, povećanje prodaje postojećeg,

smanjenje utjecaja negativnih recenzija. Uspješnost pojedine marketinške kampanje mjeri se pomoću *CTR*-a.

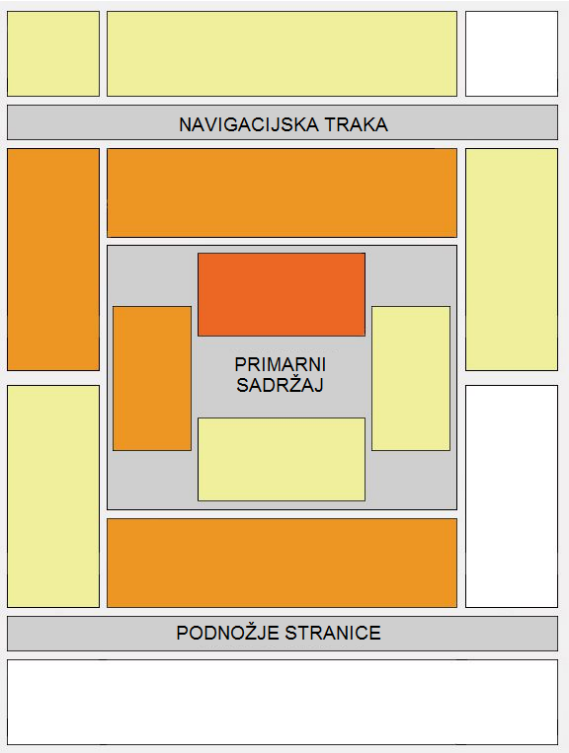
Search Engine Marketing Professional Organization, neprofitna organizacija u službi marketinške industrije, definira oglas kao „reklamu koju osoba vidi nakon slanja upita tražilici ili Internet stranici“[7]. Preciznije, oglasi su javne obavijesti dizajnirane kako bi informirale i motivirale javnost na poduzimanje akcija koje oglašivač želi.

3.2. Izdavač, stranica i oglasno mjesto

U Internet oglašavanju, izdavač je svatko tko profitira prikazivanjem oglasa potencijalnim klijentima oglašivača. Izdavači prikazuju oglase posjetiteljima svoje Internet stranice te ih preusmjeravaju prema stranicama oglašivača gdje mogu provesti kupovinu proizvoda ili usluga.

Izdavači na svojim stranicama odvajaju prostor za oglase po standardima koje je postavio *IAB*. Prostor koji se odvaja je definiran prema tipu oglasa, standardne dimenzije oglasa u pikselima dane su slikom *2.1*.

Pravilnim pozicioniranjem oglasa povećava se efikasnost oglašavanja. *Google Adsense* na temelju prikupljenih podataka o oglašavanju predlaže pozicioniranje oglasa kao na slici 3.1.



Slika 3.1. *Google Adsense Heatmap*

Intenzivnije boje prikazuju preporučenu poziciju oglasa na stranici. Uobičajena pozicija *banner* oglasa je ispod navigacijske trake.

3.3. Uobičajene funkcionalnosti

Uz prethodno navedene funkcije sustava za oglašavanje, praćenje broja prikaza i klikova oglasa, postoje mnoge kompleksnije funkcije koje nastoje povećati efikasnost marketinške kampanje. Neke od tih funkcija su[8]:

- *Behavioral* i *Geo targeting*,
- praćenje i upravljanje oglasnim prostorom na stranici,
- optimiziranje kampanje,
- dostavljanje višestrukih oglasa,
- ograničavanje frekvencije pojavljivanja oglasa.

Behavioral i *Geo targeting* se često kombiniraju, time je moguće ograničiti oglase specifično po lokaciji klijenta te njegovim interesima, što može rezultirati povećanjem interesa klijenta za prikazane oglase. Obično, sustavi koji poslužuju oglase temeljene na navikama svojih klijenata imaju veću vjerojatnost postizanja željenog cilja oglašavanja nego oni koji ne prate navike klijenata.

Analizom prostora za oglašavanje poslužitelji mogu predložiti poziciju oglasa kojom bi se postigla veća efikasnost kampanje. Ispravan odabir prostora oglasa ima veliki značaj na ishod kampanje. Pravilnim odabirom prostora oglasa lakše je pridobiti pažnju potencijalnih klijenata.

Kako bi se izbjegao pad interesa za određeni oglas sustavi za oglašavanje imaju svojstvo ograničavanja broja uzastopnog prikazivanja istog oglasa. Ako se istom korisniku jedan oglas prikaže nekoliko puta, sustav blokira prikazivanje tog oglasa određeni vremenski period. Ova funkcionalnost se obično primjenjuje na kampanje čiji se učinak mjeri *CTR*-om, dok za kampanje čiji su ciljevi podizanje svijesti o proizvodu to često nije opcija.

3.4. Tehnička izvedba sustava za isporuku oglasa

Sustavi za isporuku oglasa dijele se prema mogućnostima koje nude. Neka od svojstava sustava za oglašavanje su: podrška mobilne platforme, podrška video oglasa, podjela po formatima oglasa, generiranje izvještaja te prikupljanje podataka za ciljano oglašavanje.

Zbog velikom broja mobilnih uređaja razvila se potreba za podrškom oglašavanja putem mobilnih platformi. Zbog toga većina sustava za oglašavanje ima podršku za mobilne platforme.

Isto vrijedi i za podršku video oglasa, većinom sustavi za oglašavanje podržavaju video oglase različitih formata.

Sustavi za oglašavanje mogu podržavati velik broj formata oglasa, neki od kojih su:

- Slika,
- *Flash*,
- HTML/ XML,
- *Rich Media Video*,
- HTML 5,
- *Display ads*,
- *Full Page Overlay*,
- *Expandable banner*,
- *Pop-up / Pop-under*,
- *3D Flip banner*,
- *Carousel banner*,
- i mnogi drugi.

Generiranje izvještaja o radu sustava za oglašavanje može biti u stvarnom vremenu, putem obavijesti, izvještavanje u određenom vremenskom periodu i mnogi drugi oblici izvještavanja. Generirani izvještaji mogu sadržavati neke od sljedećih informacija:

- broj impresija,
- broj klikova,
- prosječan efektivni *CPM*,
- *CTR*,
- prihod od oglašavanja,
- i mnoge druge informacije.

Za potrebe ciljnog oglašavanja sustav može prikupljati različite informacije o korisniku u svrhu efikasnijeg oglašavanja. Prikupljeni podaci mogu biti:

- datum i vrijeme,
- dan u tjednu,
- jezik,
- zemlja, regija, grad,
- vremenska zona,
- ISP,
- OS i verzija,
- preglednik i verzija,
- podaci o mobilnom uređaju (*brand*, model, svojstva),
- IP adresa,
- ključne riječi pretraživanja,
- i mnogi drugi.

U nastavku rada su opisani prikupljeni podaci i metode rukovanja podacima.

4. METODOLOGIJA

Strojno učenje predstavlja moćan alat za predviđanje rezultata temeljeno na podacima iz prošlosti. Ono se temelji na algoritmima koji generiraju model na temelju podatkovnog skupa. Generirani model se potom može primjenjivati na podatke kojima se želi predvidjeti konačni rezultat. Sami algoritmi se dijele na dva tipa – nadgledano učenje (*eng. supervised learning*) i nenadgledano učenje (*eng. unsupervised learning*). *Supervised learning* se odnosi na rad s podacima koji imaju točno definirane ulazne i izlazne podatke (npr. Za svaki ulaz *X* definiran je izlaz *Y*, na temelju prošlih saznanja o parovima *X* i *Y* algoritam na temelju poznate varijable *X* predviđa rezultat varijable *Y*). *Unsupervised learning* drukčije pristupa podacima, nisu definirane ulazne i izlazne varijable, algoritam na temelju podataka pronalazi skrivene uzorke. Stabla odlučivanja na kojima se temelji diplomski rad spadaju u skupinu *supervised learning* algoritama. Strojno učenje ima široku primjenu u svijetu - *spam* filteri, prepoznavanje glasa, robotika, medicina, oglašavanje i analiza podataka su samo neki od primjera.

Prilikom izrade rada korišten je programski paket *RapidMiner v7.2* s podacima koje su prikupili kolege iz tvrtke *AdCumulus*.

4.1. Skup podataka za analizu

Operatori stabala odlučivanja korišteni u radu generiraju model treniranjem algoritma na temelju podatkovnog skupa. No, prije samog treniranja stabla potrebno je analizirati podatke i poduzeti određene korake kako bi se isti pripremili za rad. *RapidMiner* programski paket podržava veliki broj formata podataka s kojima se može raditi.

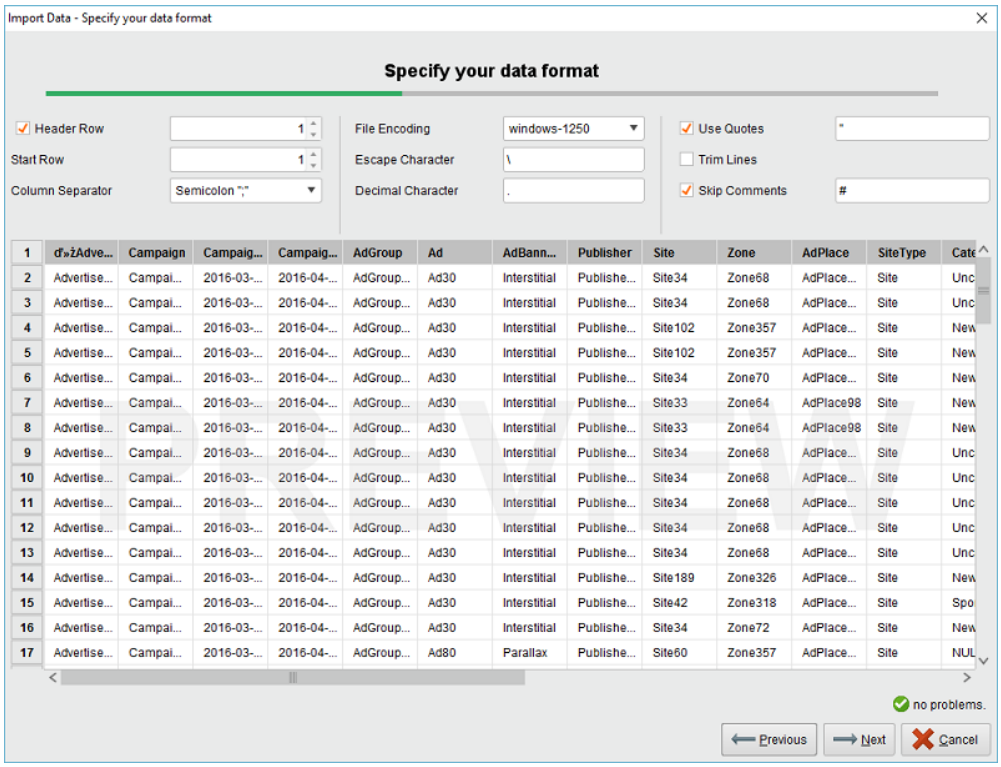
Korišteni podaci su dani u formatu vrijednost razdvojenih zarezom (*eng. Comma-separated Values*). Prvi redak podataka je zaglavlje (*eng. Header*) kojim se definiraju nazivi atributa, svi ostali redci predstavljaju objekte kojima su definirane vrijednosti atributa zaglavlja.

RapidMiner podržava nekoliko tipova atributa[9]:

- *Binominal* – tip podatka koji ima isključivo dvije vrijednosti (npr. 1/0, *true/false*, …),
- *Date* – datum bez vremenske oznake (npr. 01.01.2016),
- *Date*_time – datum s vremenskom oznakom (npr. 01.01.2016 00:00:01),
- *Integer* – cjelobrojni tip podatka (npr. -1234, 0, 5678),
- *Nominal* – svi tipovi tekstualnih vrijednosti, uključuje *polynominal* i *binominal* tipove,
- *Numeric* – svi tipovi brojčanih vrijednosti, uključuje *date, date_time, integer, real,*
- *Polynominal* – višestruke tekstualne vrijednosti (npr. *Ad30, Ad33, Ad80, …*),

- *Real* – racionalni brojevi,
- *Text*,
- *Time* – vrijeme bez datuma (npr. 14:45:00).

Prilikom dodavanja novih podataka u *RapidMiner* program nudi opcije podešavanja retka zaglavlja, sistemskog kodiranja i mnoge druge prikazane slikom 4.1.

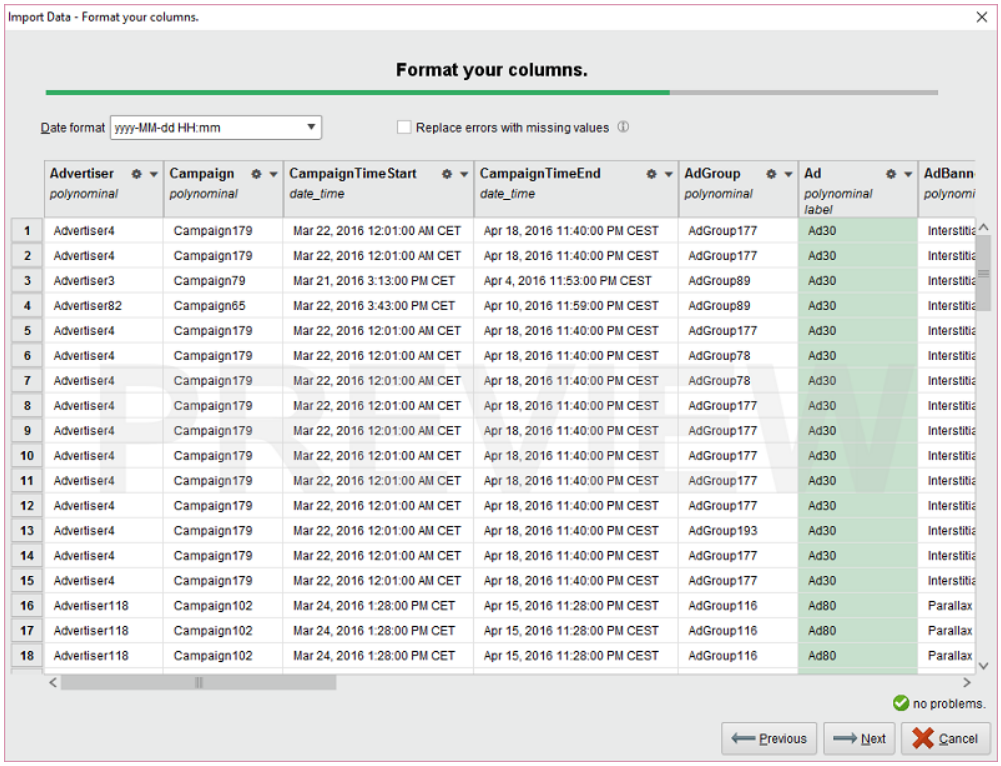


Slika 4.1. *Unos podataka u RapidMiner*

Slika 4.1 prikazuje samo dio atributa. Atributi dani u podacima su:

- *Advertiser, Campaign, CampaignTimeStart, CampaignTimeEnd, AdGroup, Ad, AdBannerType, Publisher, Site, Zone, AdPlace, SiteType, Categories, Country, Region, DeviceOs, DeviceBrand, DeviceModel, DeviceType, ISP, DateHour, Impressions, Clicks* te *Fingerprint.*

Za svaki podatak se može definirati tip i uloga. Tekstualni podaci inherentno poprimaju podatkovni tip *polynominal* dok brojčani poprimaju *numeric*. Atributi *CampaignTimeStart*, *CampaignTimeEnd* te *DateHour* predstavljaju vremenski format koji nije automatski prepoznat. S tim u vidu potrebno je podesiti tip za spomenute atribute što se može učiniti sljedećim korakom dodavanja podataka u *RapidMiner* kao što je prikazano slikom 4.2.



Slika 4.2. *Definiranje tip i uloga atributa*

Slikom 4.2. je prikazan ispravan odabir tipa za atribute *CampaignTimeStart*, *CampaignTimeEnd* te *DateHour* koji je u danom slučaju *date_time*. Kako bi *RapidMiner* ispravno prepoznao atribute tipa *date_time* korisnik mora definirati vremenski format kojim se generaliziraju vrijednosti atributa. Prethodno spomenuti atributi koriste vremenski format 'yyyy-MM-dd HH:mm' gdje *yyyy* predstavlja godinu, *MM* mjesec u godini, *dd* dan u mjesecu, *HH* sate u danu zadane 24-satnim formatom te *mm* za minute u satu. U slučaju kada format datuma ne odgovara formatu korištenom za atribute tipa *date_time* program obavještava korisnika kako nedostaju vrijednosti za pojedini atribut – sve vrijednosti *date_time* atributa se postave na '**?**' što program prepoznaje kao nepostojanu vrijednost (*eng. Missing value*).

Osim ispravnog postavljanja tipa za spomenute atribute potrebno je postaviti ulogu za ciljni atribut. Uvodni paragraf 4. poglavlja diplomskog rada opisuje dva tipa algoritama strojnog učenja te je opisano kako *supervised learning* algoritmi na temelju ulaznih podataka predviđaju izlazni podatak. Ciljni atribut (*eng. Target attribute*) predstavlja taj izlazni podatak tj. atribut koji je potrebno predviđati. U tu svrhu se postavlja uloga atributa *Ad* na *label* čime se označava ciljni atribut.

Zadnji atribut kojemu se postavlja uloga je *Fingerprint* – korisnički identifikator čija uloga se postavlja na *id*. Alternativno, moguće je u potpunosti ukloniti atribut *Fingerprint* iz

podatkovnog skupa bez da se promijeni točnost klasifikacije. Unatoč tomu, pri treniranju i testiranju stabala odlučivanja atribut *Fingerprint* je ipak zadržan.

Uz spomenute uloge postoje i druge koje su primjenjive za *clustering* te *weight-based* stabla odlučivanja.

Potrebno je napomenuti kako podatkovni skup sadrži 253703 linije 24-dimenzionalnih podataka, ciljni atribut tih podataka sadrži 82 unikatne vrijednosti. Tekstualni atributi imaju od minimalno 2 vrijednosti do maksimalno 1339 različitih vrijednosti. Na temelju statističkih podataka koje pruža *RapidMiner* moguće je očekivati kompleksno stablo s velikim brojem grana.

Nakon dodavanja podataka za obradu moguće je započeti treniranje algoritma.

4.2. Stabla odlučivanja

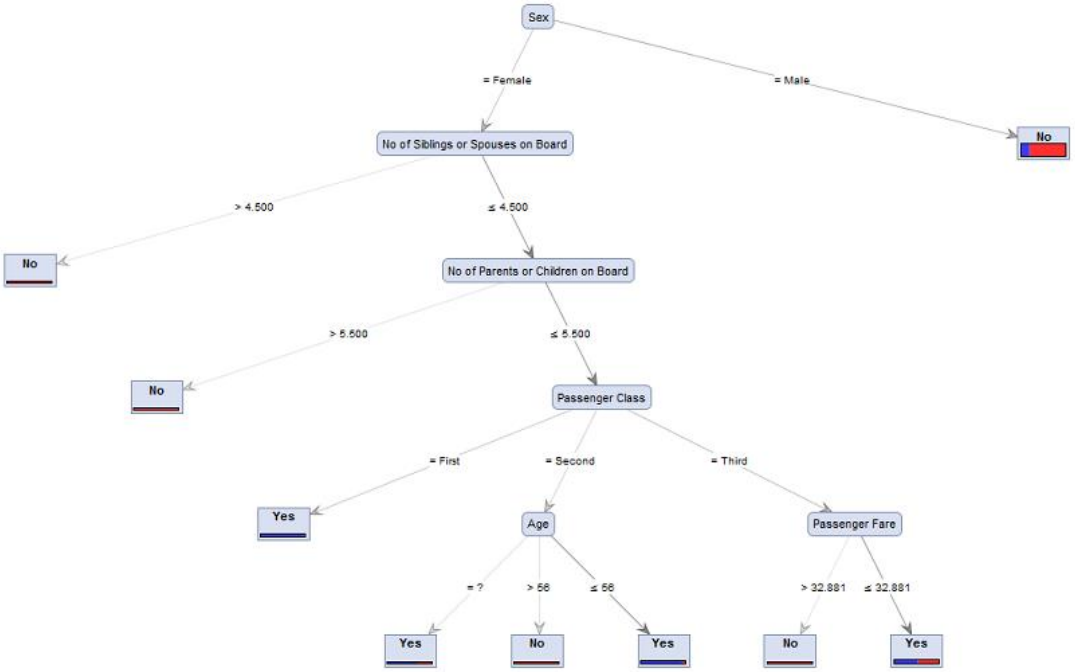
Stablo odlučivanja je hijerarhijsko stablo kojim se provodi klasifikacija podataka na temelju vrijednosti atributa podatkovnog skupa. Stabla odlučivanja se dijele na dva tipa:

- Klasifikacijska stabla (*eng. Classification tree*),
- Regresijska stabla (*eng. Regression tree*).

U slučaju regresije prikupljeni podaci sadrže isključivo numeričke vrijednosti na temelju kojih se predviđa numerički izlaz. Klasifikacija se odnosi na rad s tekstualnim podacima, gdje ciljni atribut može poprimiti određeni raspon vrijednosti tekstualnog tipa. Problematika ovog rada se odnosi na klasifikaciju podataka.

Neovisno o tipu stabla struktura ostaje ista. Svako stablo se sastoji od čvorova (*eng. node*) i grana (*eng. branch*). Svaki čvor je vezan za određeni atribut dok svaka grana koja izlazi iz tog čvora poprima neku od mogućih vrijednosti atributa. Svako stablo započinje korijenskim čvorom i završava listom (*eng. Leaf node*). List predstavlja ishod klasifikacije, vrijednost ciljnog atributa.

Za primjer strukture stabla odlučivanja generiran je model o preživjelima s Titanica, stablo je prikazano slikom 4.3.



Slika 4.3. *Stablo odlučivanja, Titanic - vjerojatnost preživljavanja*

Pri generiranju stabla odlučivanja koristi se neki od specijaliziranih algoritama (npr. *ID3, C4.5, CART, MARS*) koji su rekurzivni i temeljeni na Huntovom algoritmu, princip kojeg se zasniva na tri koraka[10]:

1. *Ispitivanje podatkovnog skupa i pronalaženje najboljeg atributa za roditeljski čvor,*
2. *Razdvajanje podataka na temelju vrijednosti odabranog atributa,*
3. *Rekurzivno ponavljanje za svaki generirani dječji čvor temeljeno na preostalim atributima.*

Osnovni problem pri generiranju stabla odlučivanja je optimiziranje razdvajanja elemenata. Kako bi se problem kvantizirao, uveden je pojam „nečistoće čvorova“ (*eng. Node Impurity*). „Nečistoća čvorova“ je mjera koja govori koliko često bi nasumično odabran element iz skupa podataka bio pogrešno klasificiran kada bi se nasumično klasificirao u skladu s distribucijom mogućih vrijednosti tog skupa. Postoji nekoliko metoda kojima se može računati „nečistoća“[11] tj. optimizirati odabir atributa za razdvajanje elemenata:

1. *GINI Indeks,*
2. *Entropija,*
3. *Greška klasifikacije (eng. Classification Error).*

4.2.1. GINI Indeks

GINI Indeks se računa prema izrazu (4-1):

$$I_G(t) = 1 - \sum_j p(j|t)^2 \tag{4-1}$$

gdje je *p* vjerojatnost pojavljivanje klase *j* za čvor *t*.

GINI indeks je mjera kojom se određuje vjerojatnost pogrešne klasifikacije nasumično odabranog elementa ako je istomu dodijeljena nasumično odabrana klasa iz danog skupa. GINI indeks poprima maksimalnu vrijednost u slučaju uniformne distribucije klasa. U slučaju kada svi elementi pripadaju istoj klasi, GINI indeks poprima minimalnu vrijednost.

4.2.2. Entropija

Entropija se računa izrazom (4-2):

$$I_H(t) = -\sum_j p(j|t) \log_2 p(j|t) \tag{4-2}$$

gdje je $p(j|t)$ vjerojatnost pojavljivanja klase *j* u čvoru *t* uz uvjet da je $p(j|t) \neq 0$.

Entropija iznosi 0 ako svi elementi pripadaju istoj klasi dok maksimum postiže kada postoji uniformna distribucija klasa.

4.2.3. Greška klasifikacije

Greška klasifikacije se računa po izrazu (4-3):

$$I_E(t) = 1 - max\{p(j|t)\} \tag{4-3}$$

gdje $max\{p(j|t)\}$ predstavlja vjerojatnost najzastupljenije klase čvora t. Greška klasifikacije poprima maksimum u slučaju uniformne distribucije klasa, dok minimum poprima kada svi elementi pripadaju istoj klasi.

Za inicijalni odabir atributa korijenskog čvora vrši se proračun nečistoće po nekom od opisanih kriterija. Za svaki čvor se vrši proračun informacijske dobiti (*eng. Information Gain*)

kako bi se doznala dobit razdvajanjem podatkovne tablice u podskupove na temelju vrijednosti atributa. Informacijska dobit se računa po izrazu (4-4):

$$IG(T, a) = H(T) - H(T|a) \tag{4-4}$$

gdje je *H(T)* entropija roditeljske podatkovne tablice, a *H(T|a)* suma entropija dječjih podatkovnih tablica. Na temelju izvršenog proračuna nečistoća i informacijske dobiti vrši se odabir optimalnog atributa kojim se maksimizira informacijska dobit. Atribut s najvećom informacijskom dobiti se postavlja kao čvor te se vrši grananje po klasama atributa. Potom se atribut čvora uklanja iz podatkovne tablice i započinje proračun iduće iteracije. Za svaki generirani čvor se vrše proračuni sve dok se generiranjem ne formiraju listovi.

Operatori za treniranje stabla odlučivanja u programskom paketu *RapidMiner* nude četiri osnovna kriterija po kojima se vrši razdvajanje elemenata. Uz GINI indeks i informacijsku dobit, postoji mogućnost odabira kriterija omjera dobiti (*eng. Gain ratio*) te točnosti (*eng. accuracy*).

Omjer dobiti je izvedeni oblik informacijske dobiti koji vrši normalizaciju dobiti uzimajući u obzir broj potencijalnih ishoda pri razdvajanju elemenata. Za razliku od kriterija informacijske dobiti koji daje prednost atributima s većim brojem mogućih ishoda, omjer dobiti nastoji osigurati jednaku mogućnost odabira time što „kažnjava“ atribute s velikim brojem mogućih ishoda.

Kriterij točnosti, za razliku od prethodno spomenutih kriterija, pri razdvajanju elemenata odabire samo onaj atribut kojim se postiže najveća točnost cjelokupnog stabla.

4.3. Priprema podataka za treniranje stabala

Prije početka treniranja stabla potrebno je prikupljene podatke podijeliti na dva podatkovna skupa – skup za treniranje (*eng. Training set*) i skup za testiranje (*eng. Test set*). Postoji nekoliko preporuka za podjelu podataka ovisno o veličini podatkovnog skupa. Omjer po kojem se podaci dijele na skup za treniranje i testiranje ima značajan utjecaj na male podatkovne skupove gdje se povećanjem skupa za treniranje može ujedno i poboljšati i narušiti ishod predikcije. U slučaju velikih podatkovnih skupova, poput skupa korištenog u izradi ovog diplomskog rada, manjim promjenama omjera ne opažaju se značajnije promjene ishoda predikcije. Odabran je preporučeni omjer od 80% ukupnih podataka u svrhu treniranja te 20% u svrhe testiranja.

Podjelu podataka je moguće provesti primjenom *Split Data* operatora u *RapidMiner*-u. U tu svrhu je kreiran novi proces s dva operatora, *Generate Attributes* kojim se generiraju novi

atributi te *Split Data* operator kojim se podaci dijele po korisnički definiranom omjeru. Operator *Split Data* nudi nekoliko načina kojima se podaci mogu podijeliti:

- *Linear sampling* – podjela podataka na skupove bez promjene njihovog redoslijeda,
- *Shuffled sampling* – kreiraju se skupovi nasumičnim odabirom podataka,
- *Stratified sampling* – kreiraju se skupovi tako da svaki ima uniformnu distribuciju klasa.

Za podjelu podataka je odabrana opcija *Shuffled sampling*. Proces je prikazan slikom 4.4.

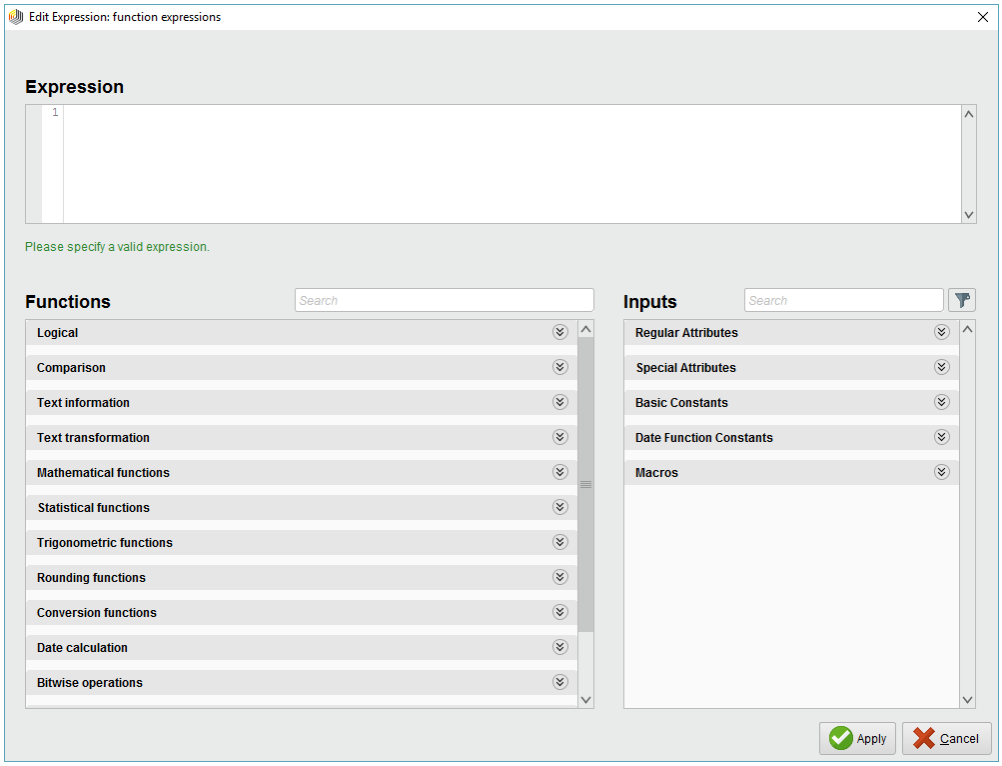


Slika 4.4. *Proces podjele podataka primjenom Split Data operatora*

Ulaz procesa čini originalni skup podataka koji se prosljeđuje operatoru *Generate Attributes* kako bi se generirao novi atribut. Izlaz iz spomenutog operatora je originalni skup podataka s dodanim novim atributom koji se potom dijeli na dva podskupa pomoću *Split Data* operatora. *Split Data* operator ima onoliko izlaza koliko skupova korisnik želi kreirati, slika 4.4. prikazuje dva izlaza čiji skupovi se pohranjuju u lokalni repozitorij s nazivom *trainingSet* i *testingSet*.

Skup za treniranje, kao što sam naziv ukazuje, koristi se za treniranje stabla tj. za generiranje modela. Kako bi se potvrdila ispravnost modela potrebno je isti primijeniti na nepoznati skup podataka. Primjenom modela na nepoznate podatke dobije se povratna informacija o performansama modela.

Prilikom podjele podataka osim *Split Data* operatora korišten je i operator *Generate Attributes* kojim je moguće kreirati nove atribute na temelju postojećih. Generiranje novih atributa uključuje dodjeljivanje naziva atributu i definiranje izraza kojim se atribut generira. Sam operator nudi veliki broj predefiniranih izraza za različite tipove podataka. Tako je moguće razdvajanje tekstualnih atributa, primjena matematičkih funkcija na numeričke atribute, rukovanje vremenskim atributima i mnoge druge operacije. Prozor za definiranje izraza generiranja novog atributa je prikazan slikom 4.5.



Slika 4.5. *Definiranje izraza za generiranje novog atributa*

Izraz je moguće formirati klikom željene funkcije s popisa dostupnih uz dodavanje postojećih atributa kao argumenata korištenih funkcija. Prilikom treniranja stabla s dostupnim podacima pokazalo se kako ishod klasifikacije postaje točniji ako se generira novi atribut kao razlika vremena početka i kraja kampanje. Atribut je nazvan *CampaignTime* i definira trajanje kampanje izraženo u sekundama.

Potencijalni problem pri treniranju stabala odlučivanja je mogućnost da generirani model bude savršeno prilagođen podacima za treniranje (*eng. Overfitting*) i time ne bude primjenjiv na druge podatke. Kako bi se izbjegao *overfitting* stvorene su dvije preventivne metode kojima se stablo savršeno prilagođeno skupu za treniranje nastoji učiniti manje specifičnim te se time prilagođava za rad s nepoznatim podacima. Spomenute metode su obrezivanje stabla tokom treniranja (*eng. pre-pruning*) i obrezivanje stabla nakon treniranja (*eng. post-pruning, pruning*).

Pre-pruning funkcionira na način da se pri treniranju ranije zaustavlja izgradnja stabla prije nego se isto savršeno prilagodi podacima skupa za treniranje. Kako nije moguće efikasno odlučiti kada zaustaviti izgradnju stabla često se odlučuje za *post-pruning* metodu.

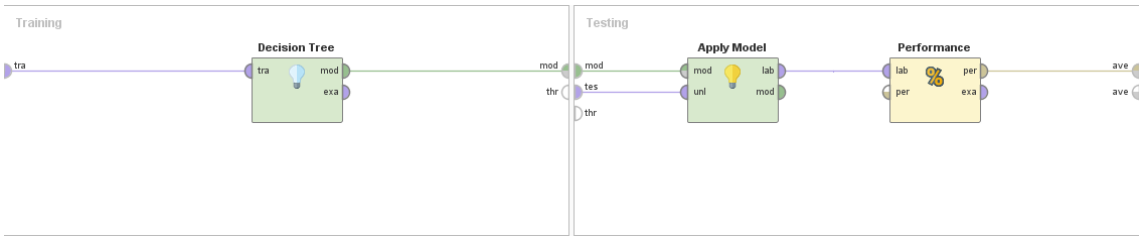
Post-pruning se provodi na potpuno formiranom modelu stabla. Na takvom stablu se potom provodi obrezivanje. Opcija za *post-pruning* u programskom alatu *RapidMiner* sadrži numerički parametar pouzdanosti (*eng. Confidence*) koji se koristi pri estimaciji greške obrezivanja.

Obrezivanje se vrši od listova prema korijenskom čvoru. Za svaku granu se vrši proračun greške klasifikacije za slučaj stabla u kojem je ta grana obrezana i stabla u kojem nije obrezana. Ako je greška obrezanog stabla manja od greške neobrezanog stabla, tada se grana obrezuje.

4.4. Postupak treniranja stabla

Treniranje stabla odlučivanja u programskom alatu *RapidMiner* je vrlo intuitivno. Kako bi se započelo treniranje potrebno je odabrati željeni operator za treniranje stabla te operator za mjerenje učinkovitosti. Najjednostavniji proces kojim se trenira stablo uključuje operator *X-Validation* iliti *Cross-Validation* kojim se mjeri učinkovitost treniranog stabla, *Decision Tree* operator kojim se generira model, *Apply Model* operator kojim se spomenuti model primjenjuje na podatke te *Performance* operator kojim se generira vektor performansi potreban za rad *X-Validation* operatora.

X-Validation operator je ugniježđeni operator tj. sadrži potproces podijeljen u dva dijela – *training* dijelom se trenira stablo te se generira model koji se prosljeđuje *testing* dijelu unutar kojeg se dobiveni model primjenjuje na unakrsno-vrednujući (*eng. Cross-validation*) podatkovni skup kako bi se moglo generirati vektor performansi. *X-Validation* operator za ulaz prima podatkovni skup od kojega se jedan podskup odvaja za potrebe treniranja – *cross-validation* skup. Ovisno o broju vrednovanja koje korisnik želi provesti mijenja se veličina *cross-validation* skupa. Radom s *X-Validation* operatorom je odabrano izvođenje 10 vrednovanja čime se ulazni podatkovni skup dijeli na 10 podskupova. Vrednovanjem se svaki podskup mora jednom koristiti za potrebe testiranja modela. Prikaz potprocesa *X-Validation* operatora je dan slikom 4.6.

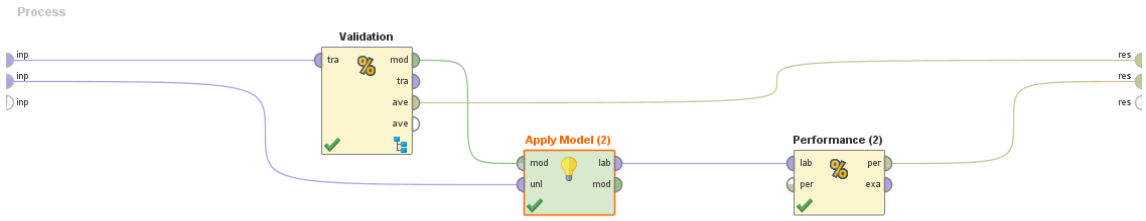


Slika 4.6. *X-Validation potproces*

Training dio sadrži neki od učećih operatora, primjer sa slike 4.6. sadrži *Decision Tree* operator kojem se može definirati kriterij dijeljenja atributa, maksimalna dubina stabla te treba li primijeniti *pre-pruning* ili *post-pruning*. Izlaz *Decision Tree* operatora je model koji se u *testing* dijelu primjenjuje na *cross-validation* skup pomoću *Apply Model* operatora. Nakon primjene

modela dobije se rezultat predikcije ciljnog atributa na temelju kojeg se provodi mjerenje performansi modela za što se primjenjuje *Performance* operator. Svakim vrednovanjem se generira jedan vektor performansi. Taj vektor sadrži rezultate točnosti klasifikacije za model predviđanja vrednovan trenutnim *cross-validation* skupom. Svakom iteracijom se generira jedan vektor performansi dok se ukupni rezultat dobije kao prosjek svih vektora. Potrebno je napomenuti kako *X-Validation* procjenjuju performanse modela na nepoznatim podacima.

Kao dodatnu provjeru ispravnosti potrebno je generirani model primijeniti na skup za testiranje primjer čega je dan slikom 4.7.



Slika 4.7. *Primjena modela na nepoznate podatke*

Alternativno, moguće je koristiti neki od ansambl operatora. Ansambl operator umjesto jednog stabla odlučivanja generira više stabala s ciljem poboljšanja performansi. Neki od ansambl operatora su:

- *Random Forest* – generira korisnički definiran broj stabala na temelju kojih se glasanjem odabire najefikasnije stablo,
- *Gradient Booster Trees* – generira model predviđanja kao ansambl slabijih modela, primjenom *boosting* metoda povećava se točnost klasifikacije,
- *AdaBoost* – iterativno generira model predviđanja, adaptivni algoritam koji svakom iteracijom generira model u korist pogrešno klasificiranih podataka,
- *Vote* – na temelju skupa za treniranje generira nekoliko modela primjenom različitih učećih algoritama te na temelju izglasavanja odlučuje koji model se primjenjuje na nepoznate podatke,
- *Hierarchical Classification* – generira model predviđanja na temelju korisnički definirane hijerarhijske strukture,
- i drugi.

Neki od navedenih operatora su ugniježđeni te im je potrebno dodati učeće operatore kako bi se generirao model.

5. REZULTATI

Proučavanjem podataka i testiranjem različitih pristupa treniranju stabala odlučivanja prikupljeni su rezultati za svaki od generiranih modela. Neki od operatora nisu korišteni zbog ograničenja računala – nedovoljno radne memorije. Rezultati prve tablice potpoglavlja služe kao referentna vrijednost po kojoj se vrednuje ishod rada ostalih procesa za dani učeći operator.

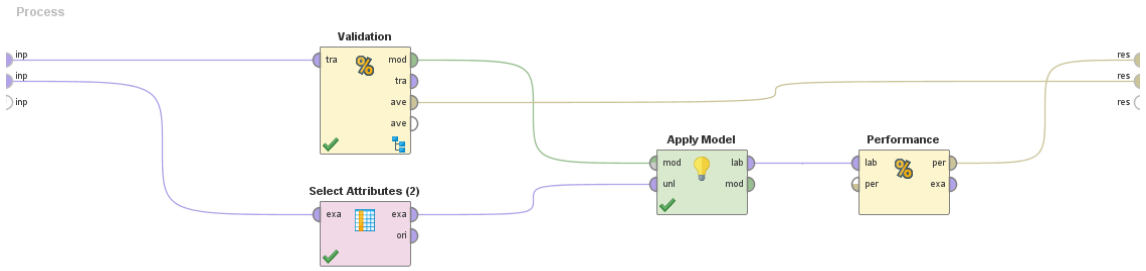
5.1. *Decision Tree* učeći operator

Prvi u nizu testiranih učećih operatora je *Decision Tree*. Spomenuti operator je testiran u različitim situacijama – s originalnim atributima, s ansambl operatorima, s novim atributom te s filterom podataka. Različite situacije daju različite rezultate koji su opisani u nastavku.

Primjenom *Decision Tree* operatora na prikupljene podatke za proces prikazan slikom 5.1., bez generiranja novih atributa, uz parametar maksimalne dubine stabla 23 i uključen *pruning* sa stopom pouzdanosti 0.25 dobiveni su rezultati točnosti klasifikacije po različitim kriterijima dani tablicom 5-1.:

Tablica 5-1. *Rezultati klasifikacije primjenom Decision Tree operatora*

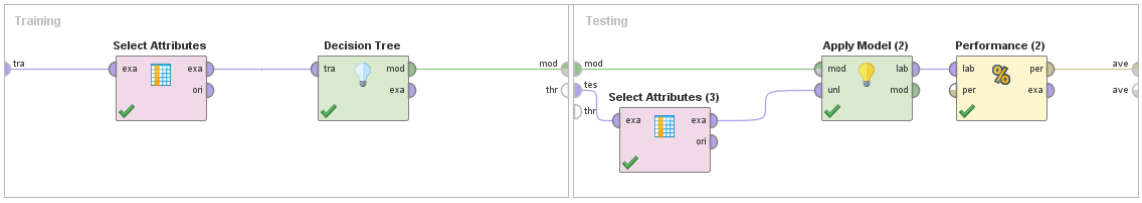
Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	73.46% +/- 0.35%	73.48%
informacijska dobit	72.87% +/- 0.18%	72.62%
GINI indeks	72.88% +/- 0.20%	72.56%
točnost	72.76% +/- 0.26%	72.66%



Slika 5.1. *Prikaz procesa prediktora primjenom Decision Tree operatora*

Proces sa slike 5.1. sadrži *X-Validation* operator čiji potproces služi za treniranje stabla odlučivanja. Generirani model se nakon treniranja stabla primjenjuje na nepoznate podatke kako bi se dobila točnost klasifikacije tj. kako bi se testirala mogućnost modela da na temelju poznatih ulaznih podataka izvrši ispravan odabir vrijednosti ciljnog atributa na temelju poznatih podataka

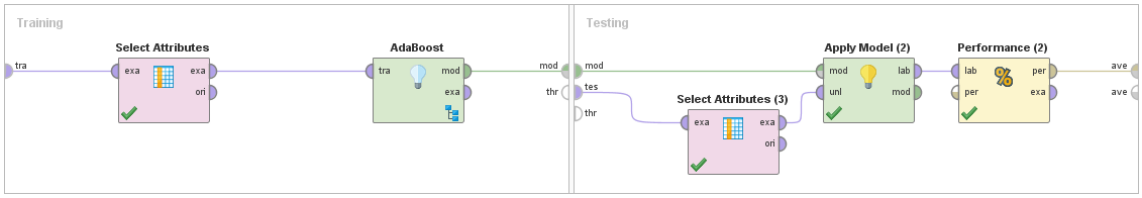
iz prošlosti. U kontekstu rada to znači da se testira točnost ispravnosti odabranog oglasa na temelju saznanja o korisniku. Potproces *X-Validation* operatora je prikazan slikom 5.2.



Slika 5.2. *X-Validation potproces, primjena Decision Tree operatora*

Prethodno korišteni operator moguće je kombinirati s ansambl operatorima kako bi se poboljšali rezultati. Neki od testiranih ansambl operatora su *AdaBoost, Bagging, Stacking* te *Vote*. Navedeni ansambl operatori unutar svog potprocesa moraju sadržavati učeći operator. Zbog kompleksnosti izvedbe procesa, 10 instanci kros-validacije kojima se provodi treniranje više stabala pomoću ansambl operatora, vremenski i memorijski zahtjevi značajno rastu te nije moguće provesti testiranje određenih učećih operatora u tako definiranoj strukturi.

Rezultati dobiveni primjenom *AdaBoost* ansambl operatora i *Decision Tree* operatora odgovaraju onim podacima danim tablicom 5-1. dok je prikaz potprocesa dan slikom 5.3.



Slika 5.3. *Primjena AdaBoost ansambl operatora*

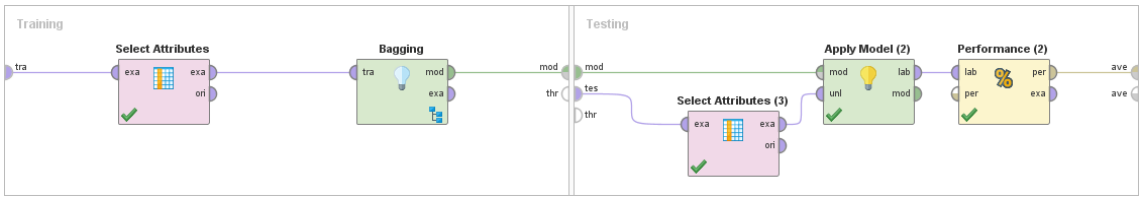
Sljedeći u nizu testiranih ansambl operatora je *Bagging* koji od ulaznog podatkovnog skupa svakom iteracijom kreira novi podskup te daje model na temelju tog podskupa. Svakom iteracijom učeći algoritam daje drugi model treniran podskupom podataka. Krajnji ishod klasifikacije je rezultat najčešće predviđene klase na temelju svih generiranih modela.

Primjenom *Bagging* ansambl operatora postižu se rezultati prikazani tablicom 5-2.

Tablica 5-2. *Rezultat Bagging ansambl operatora s Decision Tree učećim operatorom*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	73.47% +/- 0.38%	73.30% ↓
informacijska dobit	72.92% +/- 0.24%	72.79% ↑
GINI indeks	72.89% +/- 0.23%	72.70% ↑
točnost	72.83% +/- 0.22%	72.61% ↓

Ishod predikcije varira ovisno o postavljenom kriteriju, no poboljšanje tj. pogoršanje točnosti klasifikacije je zanemarivo malo. Potproces *X-Validation* operatora koji sadrži *Bagging* ansambl operator je prikazan slikom 5.4.



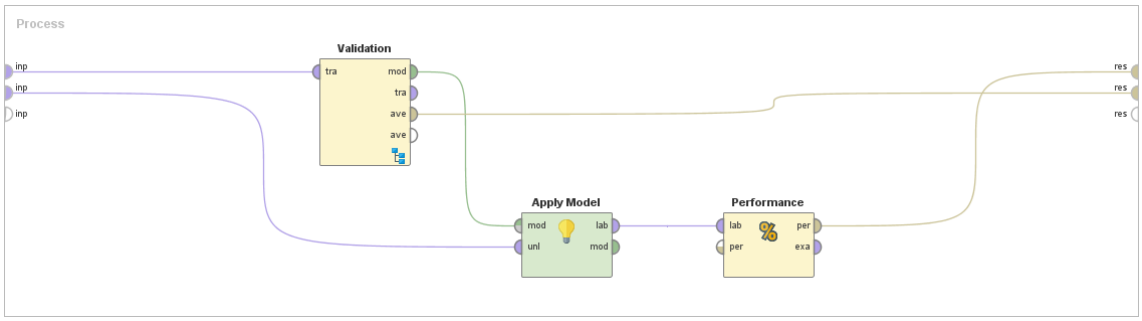
Slika 5.4. *Primjena Bagging ansambl operatora*

Za sve prethodne primjere korišteni su originalni atributi. S nadom da će točnost klasifikacije biti veća generiran je novi atribut *CampaignTime* kao vremenska razlika atributa *CampaignTimeStart* i *CampaignTimeEnd*. Uključivanjem novog atributa u treniranje modela *Decision Tree* operator daje sljedeće rezultate prikazane tablicom 5-3:

Tablica 5-3. *Točnost klasifikacije Decision Tree operatora s uključenim novim atributom*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	73.40% +/- 0.38%	73.45% ↓
informacijska dobit	72.86% +/- 0.17%	72.61% ↓
GINI indeks	72.88% +/- 0.20%	72.53% ↓
točnost	73.26% +/- 0.24%	73.13% ↑

Ovisno o odabranom kriteriju postoji malo poboljšanje tj. pogoršanje rezultata. Prikaz procesa s novim atributom je dan slikom 5.5.



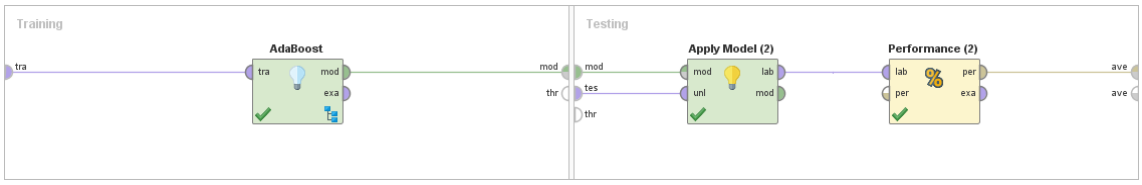
Slika 5.5. *Prikaz procesa s uključenim novim atributom*

Uključivanjem novog atributa u proces koji koristi *AdaBoost* ansambl operator s *Decision Tree* učećim operatorom postižu se rezultati dani tablicom 5-4.

Tablica 5-4. *Rezultat AdaBoost ansambl operatora s Decision Tree učećim operatorom, novi atribut*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	73.40% +/- 0.38%	73.45% ↓
informacijska dobit	72.86% +/- 0.17%	72.61% ↓
GINI indeks	72.88% +/- 0.20%	72.53% ↓
točnost	73.26% +/- 0.24%	73.13% ↑

Potproces *X-Validation* operatora s primijenjenim *AdaBoost* ansambl operatorom je prikazan na slici 5.6.



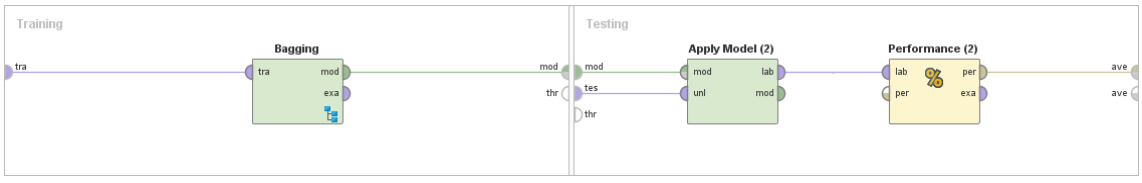
Slika 5.6. *Potproces X-Validation operatora s AdaBoost ansambl operatorom, novi atribut*

Primjenom *Bagging* ansambl operatora na podatke s novim atributom postižu se rezultati prikazani tablicom 5-5.

Tablica 5-5. *Rezultat Bagging ansambl operatora s Decision Tree učećim operatorom, novi atribut*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	73.44% +/- 0.37%	73.20% ↓
informacijska dobit	72.93% +/- 0.24%	72.79% ↑
GINI indeks	72.89% +/- 0.23%	72.70% ↑
točnost	73.34% +/- 0.23%	73.08% ↑

Potproces *X-Validation* operatora s primijenjenim *Bagging* ansambl operatorom je prikazan slikom 5.7.



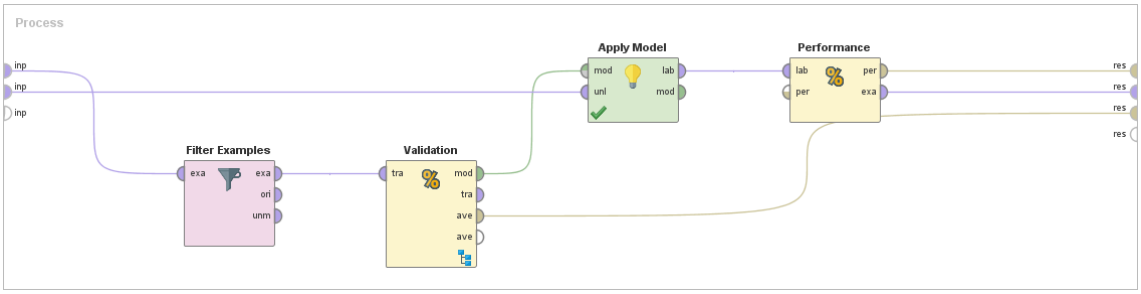
Slika 5.7. *Potproces X-Validation operatora s Bagging ansambl operatorom, novi atribut*

Budući da neki podaci za pojedine atribute imaju nepoznate vrijednosti (npr. *Unknown, NULL*) provedeno je treniranje prediktora filtriranim podacima. Uklonjeni su oni podaci čiji atributi sadrže nepoznate vrijednosti. Treniranjem stabla na takvim podacima primjenom *Decision Tree* učećeg operatora postignuti su sljedeći podaci prikazani tablicom 5-6:

Tablica 5-6. *Decision Tree, filtriranje podataka*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	71.64% +/- 0.36%	70.55% ↓
informacijska dobit	70.53% +/- 0.35%	69.70% ↓
GINI indeks	70.48% +/- 0.43%	69.67% ↓
točnost	71.41% +/- 0.30%	69.92% ↓

Prikaz procesa s uključenim filterom podataka je dan slikom 5.8. u nastavku.



Slika 5.8. *Decision Tree, filtriranje podataka*

Isti filter primijenjen je s ansambl operatorima, točnost klasifikacije dobivena tim pristupom za *AdaBoost* ansambl operator je dana tablicom 5-7 dok su rezultati *Bagging* ansambl operatora dani tablicom 5-8.

Tablica 5-7. *AdaBoost ansambl operator, filtriranje podataka*

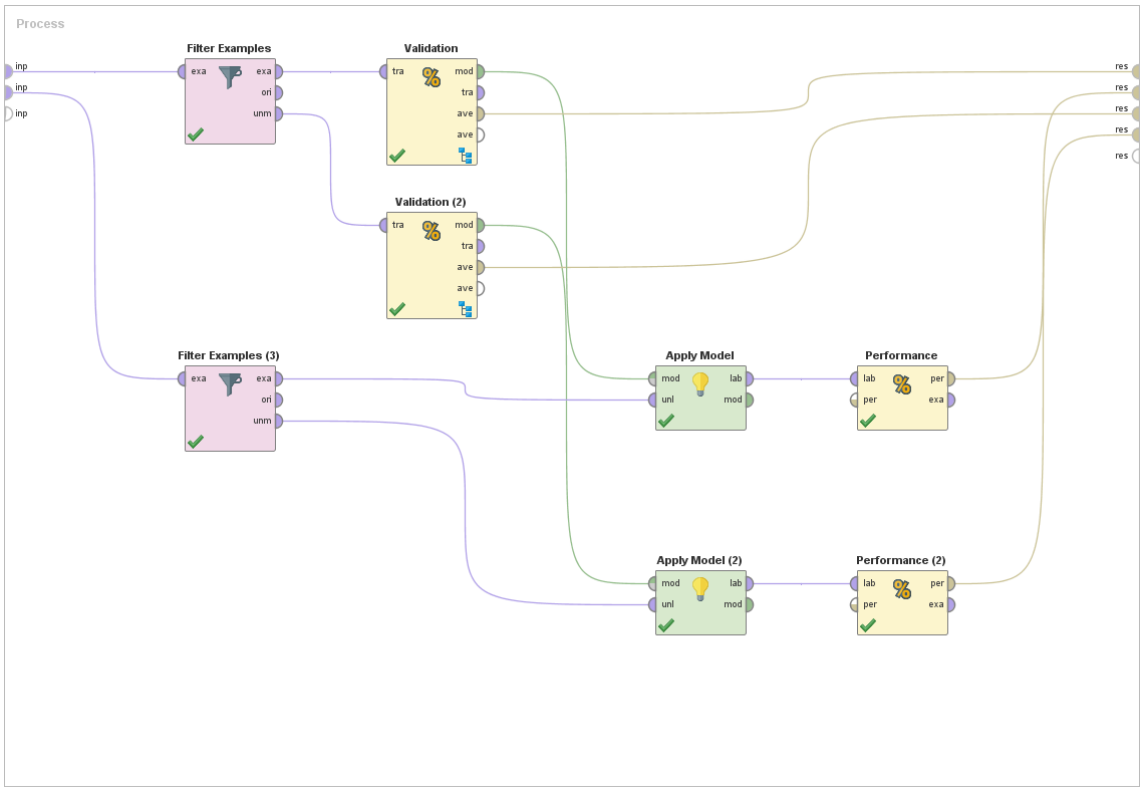
Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	71.64% +/- 0.36%	70.55% ↓
informacijska dobit	70.53% +/- 0.35%	69.70% ↓
GINI indeks	70.48% +/- 0.43%	69.67% ↓
točnost	71.41% +/- 0.30%	69.92% ↓

Tablica 5-8. *Bagging ansambl operator, filtriranje podataka*

Kriterij:	***Rezultat X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	71.61% +/- 0.31%	70.49% ↓
informacijska dobit	70.70% +/- 0.31%	69.80% ↓
GINI indeks	70.64% +/- 0.31%	69.72% ↓
točnost	71.30% +/- 0.29%	70.01% ↓

Iz dobivenih rezultata radom s *Decision Tree* operatorom, neovisno o tome primjenjuje li se ansambl operator ili ne, filtriraju li se podaci ili ne, točnost klasifikacije se ne mijenja značajnije u odnosu na prvi primjer.

Budući da se filtriranjem uklanjaju nepotpuni podaci moguće je pristupiti problemu predikcije na način da se generiraju dva modela – jedan za potpune podatke, jedan za nepotpune. Filter je primijenjen tako da iz skupa za treniranje ukloni svaki podatak čiji atribut sadrži nepoznatu vrijednost. Većina filtriranih podataka ima jedan atribut s nepoznatom vrijednošću, stoga je odabrano treniranje dva stabla odlučivanja čiji modeli se primjenjuju na potpune i nepotpune podatke. Prikaz procesa kojim je treniranje stabala provedeno dan je slikom 5.9.



Slika 5.9. *Treniranje stabala za predikciju na temelju poznatih i nepoznatih vrijednosti*

Ishod rada procesa prikazanog slikom 5.9. je dan tablicom 5-9. u nastavku.

Tablica 5-9. *Rezultati treniranja stabala odlučivanja procesom sa slike 5.9 primjenom dva modela, Decision Tree*

Podaci:	**Kriterij:**	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
Poznati	*omjer dobiti*	71.60% +/- 0.34%	71.43%
	informacijska dobit	70.55% +/- 0.37%	70.95%
	GINI indeks	70.55% +/- 0.32%	70.77%
	točnost	71.32% +/- 0.29%	70.73%
Nepoznati	*omjer dobiti*	75.19% +/- 0.25%	74.83%
	informacijska dobit	74.76% +/- 0.20%	74.33%
	GINI indeks	74.68% +/- 0.24%	74.31%
	točnost	75.30% +/- 0.26%	74.95%

Postignuti rezultati u prosjeku daju ishod klasifikacije vrlo sličan onom po tablici 5-1. No zbog treniranja dva specifična modela predviđanja ovisno o tome sadrže li ulazni podaci potpune ili nepotpune informacije, vrši se odabir modela predviđanja koji je treniran s potpunim ili nepotpunim podacima.

5.2. *Decision Stump* učeći operator

Decision Stump operator generira stablo odlučivanja koje ima samo jedno dijeljenje atributa te se može efikasno primjenjivati na nepoznate podatke. Parametri korištenih operatora pri testiranju su postavljeni na *default* vrijednosti.

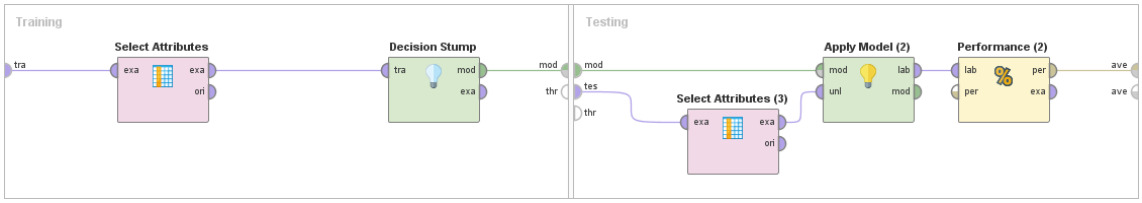
Primjenom *Decision Stump* operatora na proces sa slike 5.1 postižu se rezultati dani tablicom 5-10.:

Tablica 5-20. *Rezultati klasifikacije primjenom Decision Stump operatora*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	46.46% +/- 2.69%	47.46%
informacijska dobit	68.19% +/- 0.22%	67.90%
GINI indeks	56.58% +/- 5.81%	67.90%
točnost	56.58% +/- 5.81%	67.90%

Iz tablice 5-10. se može zaključiti kako se najbolji rezultat postiže uz postavljen kriterij informacijske dobiti operatora *Decision Stump*. Nadalje, spomenuti operator je moguće koristiti u sinergiji s *AdaBoost* operatorom koji poboljšava ishod klasifikacije generiranjem više stabala

odlučivanja. Proces za treniranje stabla formiran je po onom sa slike 5.2. uz razliku u potprocesu *X-Validation* operatora kao što je prikazano slikom 5.10.



Slika 5.10. *X-Validation potproces, primjena Decision Stump operatora*

Decision Stump operator se često primjenjuje u kombinaciji s *AdaBoost* ansambl operatorom čime se postižu bolji rezultati točnosti klasifikacije. Primjenom *AdaBoost* ansambl operatora s *Decision Stump* operatorom postižu se sljedeći rezultati dani tablicom 5-11:

Tablica 5-31. *Rezultat AdaBoost ansambl operatora s Decision Stump učećim operatorom*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	41.09% +/- 0.00%	40.79% ↓
informacijska dobit	68.19% +/- 0.22%	67.90% ~
GINI indeks	56.58% +/- 5.81%	67.90% ~
točnost	56.58% +/- 5.81%	67.90% ~

Potproces *X-Validation* operatora odgovara onom sa slike 5.3. s ključnom razlikom u učećem operatoru korištenom u potprocesu *AdaBoost* ansambl operatora.

Ako se isti učeći operator kombinira s *Bagging* ansambl operatorom dobiju se rezultati dani tablicom 5-12.

Tablica 5-42. *Rezultat Bagging ansambl operatora s Decision Stump operatorom*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	47.60% +/- 0.63%	47.46% ~
informacijska dobit	44.79% +/- 0.12%	44.57% ↓
GINI indeks	44.79% +/- 0.12%	44.57% ↓
točnost	41.0% +/- 0.00%	40.80% ↓

Prikaz potprocesa s *Bagging* ansambl operatorom odgovara onom sa slike 5.4., jedina razlika je učeći operator korišten u potprocesu spomenutog ansambl operatora.

Uvođenjem novog atributa u proces treniranja stabla potrebno je provesti testiranje sa i bez ansambl operatora za *Decision Stump* učeći operator. Za slučaj bez primjene ansambl operatora postižu se isti rezultati kao oni dani tablicom 5-10.

Uključivanjem *AdaBoost* ansambl operatora u proces za podatke s novim atributom postižu se rezultati iz tablice 5-13:

Tablica 5-53. *Rezultat AdaBoost ansambl operatora s Decision Stump učećim operatorom, novi atribut*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	41.09% +/- 0.00%	40.79% ↓
informacijska dobit	68.19% +/- 0.22%	67.90% ~
GINI indeks	56.58% +/- 5.81%	67.90% ~
točnost	56.58% +/- 5.81%	67.90% ~

Primjenom *Bagging* ansambl operatora postiže se točnost klasifikacije dana tablicom 5-14.

Tablica 5-64. *Rezultat Bagging ansambl operatora s Decision Stump učećim operatorom, novi atribut*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	47.60% +/- 0.63%	47.46% ~
informacijska dobit	-	-
GINI indeks	-	-
točnost	-	-

Slično *Decision Tree* učećem operatoru, provedeno je filtriranje podataka s nepoznatim vrijednostima i za *Decision Stump* operator. Za proces bez primjene ansambl operatora postignuti su sljedeći rezultati dani tablicom 5-15:

Tablica 5-75. *Decision Stump, filtriranje podataka*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	37.04% +/- 1.93%	40.79% ↓
informacijska dobit	68.86% +/- 0.23%	66.72% ↓
GINI indeks	49.67% +/- 0.35%	53.10% ↓
točnost	49.67% +/- 0.35%	53.10% ↓

Primjenom *Adaboost* ansambl operatora postižu se rezultati dani tablicom 5-16 dok *Bagging* ansambl operator postiže iste rezultate kao one prikazane tablicom 5-14.

Tablica 5-86. *AdaBoost ansambl operator, filtriranje podataka*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	36.39% +/- 0.00%	40.79% ↓
informacijska dobit	68.86% +/- 0.23%	66.72% ↓
GINI indeks	36.39% +/- 0.00%	40.79% ↓
točnost	36.39% +/- 0.00%	40.79% ↓

Na temelju prikupljenih podataka za rad *Decision Stump* učećeg operatora može se zaključiti kako *Bagging* ansambl operator drastično narušava učinkovitost učećeg operatora. Primjenom *AdaBoost* ansambl operatora točnost klasifikacije je često ista kao ona procesa bez spomenutog ansambl operatora. Filtriranje nepoznatih podataka također narušava točnost klasifikacije, stoga se preporuča zadržati sve podatke i koristiti ih za treniranje stabla. Zbog memorijskih zahtjeva *Bagging* ansambl operatora, tablica 5-14 nije popunjena za sve kriterije.

Primjenom dvostrukog modela sa slike 5.9. uz *Decision Stump* učeći operator postignuti su rezultati dani tablicom 5-17.

Tablica 5-17. *Rezultati rada primjenom dva modela, Decision Stump*

Podaci:	**Kriterij:**	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
Poznati	*omjer dobiti*	37.04% +/- 1.93%	35.94%
	informacijska dobit	68.86% +/- 0.23%	68.40%
	GINI indeks	49.67% +/- 0.35%	49.86%
	točnost	49.67% +/- 0.35%	49.86%
Nepoznati	*omjer dobiti*	52.66% +/- 0.02%	52.46%
	informacijska dobit	67.54% +/- 0.16%	67.28%
	GINI indeks	57.60% +/- 0.34%	56.98%
	točnost	57.60% +/- 0.34%	56.98%

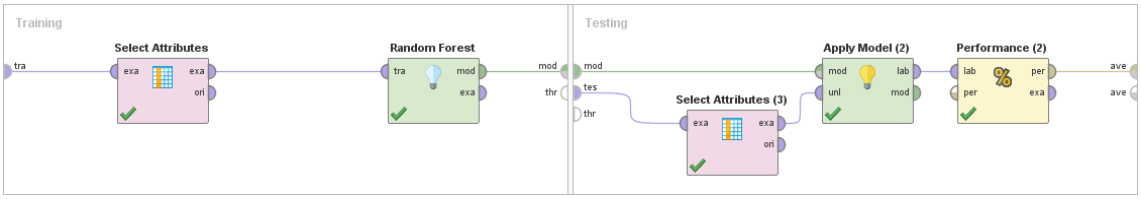
Rezultati iz tablice 5-17 ukazuju na nižu točnost klasifikacije primjenom *Decision Stump* operatora na proces sa slike 5.9. Iako su rezultati relativno dobri za kriterij informacijske dobiti u odnosu na druge kriterije, opet postoji mali pad točnosti klasifikacije te nije preporučljivo koristiti ovakav proces za predikciju rezultata.

5.3. *Random Forest* učeći operator

Osim *Decision Tree* i *Decision Stump* operatora testiran je i *Random Forest* operator. Parametri *Random Forest* operatora su podešeni za generiranje 20 stabala odlučivanja maksimalne dubine 23 uz uključen *pruning* sa stopom pouzdanosti 0.25. Rezultati rada s *Random Forest* operatorom bez filtriranja i korištenja novog atributa su dani tablicom 5-18. Potproces za treniranje stabla prikaza je slikom 5.11.

Tablica 5-18. *Rezultati klasifikacije primjenom Random Forest operatora*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	64.51% +/- 3.72%	66.36%
informacijska dobit	64.60% +/- 3.30%	65.89%
GINI indeks	64.89% +/- 2.70%	65.31%
točnost	56.48% +/- 3.08%	57.42%



Slika 5.11. *X-Validation potproces, primjena Random Forest operatora*

Nad istim podacima je provedeno testiranje s *AdaBoost* ansambl operatorom uz *Random Forest* učeći operator te su postignuti rezultati prikazani tablicom 5-19.

Tablica 5-19. *Rezultat AdaBoost ansambl operatora s Random Forest operatorom*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	60.42% +/- 7.41%	65.85% ↓
informacijska dobit	66.14% +/- 1.61%	66.82% ↑
GINI indeks	66.03% +/- 1.97%	66.10% ↑
točnost	60.09% +/- 3.33%	57.27% ↓

Kao što je bio slučaj s prethodnim učećim operatorima, *Bagging* ansambl operator je primijenjen s *Random Forest* učećim operatorom te su dobiveni rezultati u tablici 5-20.

Tablica 5-90. *Rezultat Bagging ansambl operatora s Random Forest operatorom*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	68.28% +/- 0.21%	67.94% ↑
informacijska dobit	68.25% +/- 0.14%	67.94% ↑
GINI indeks	68.28% +/- 0.21%	67.94% ↑
točnost	63.44% +/- 0.14%	63.01% ↑

Uključivanjem novog atributa u treniranje stabla, *Random Forest* učeći operator daje sljedeće rezultate prikazane tablicom 5-21:

Tablica 5-101. *Točnost klasifikacije Random Forest operatora s uključenim novim atributom*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	65.57% +/- 3.93%	64.12% ↓
informacijska dobit	68.96% +/- 1.12%	65.02% ↓
GINI indeks	66.94% +/- 3.51%	64.38% ↓
točnost	63.07% +/- 4.86%	63.03% ↑

Dodavanjem *AdaBoost* ansambl operatora u proces postižu se rezultati dani tablicom 5-22.

Tablica 5-112. *Rezultat AdaBoost ansambl operatora s Random Forest operatorom, novi atribut*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	66.21% +/- 0.95%	66.77% ↑
informacijska dobit	66.89% +/- 0.72%	67.04% ↑
GINI indeks	67.00% +/- 0.84%	67.65% ↑
točnost	62.37% +/- 3.21%	60.75% ↑

Primjenom *Bagging* ansambl operatora s *Random Forest* učećim operatorom postižu se sljedeći rezultati prikazani tablicom 5-23:

Tablica 5-123. *Rezultat Bagging ansambl operatora s Random Forest operatorom, novi atribut*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	66.49% +/- 0.44%	66.55% ↑
informacijska dobit	67.03% +/- 0.22%	66.72% ↑
GINI indeks	66.86% +/- 0.37%	66.46% ↑
točnost	58.14% +/- 1.43%	60.35% ↑

Filtriranjem nepoznatih podataka te treniranjem stabla s *Random Forest* učećim operatorom postižu se rezultati dani tablicom 5-24.

Tablica 5-134. *Random Forest, filtriranje podataka*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	65.57% +/- 3.93%	64.14% ↓
informacijska dobit	68.96% +/- 1.12%	65.02% ↓
GINI indeks	66.94% +/- 3.51%	64.38% ↓
točnost	63.07% +/- 4.86%	63.03% ↑

Dodavanjem *AdaBoost* ansambl operatora postižu se rezultati dani tablicom 5-25, dok se primjenom *Bagging* ansambl operatora dobiju rezultati dani tablicom 5-26.

Tablica 5-145. *AdaBoost ansambl operator, filtriranje podataka*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	68.28% +/- 0.79%	64.26% ↓
informacijska dobit	69.00% +/- 0.71%	65.81% ↓
GINI indeks	68.34% +/- 0.61%	65.70% ↑
točnost	67.56% +/- 0.91%	62.03% ↑

Tablica 5-156. *Bagging ansambl operator, filtriranje podataka*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	68.15% +/- 0.37%	65.11% ↓
informacijska dobit	68.35% +/- 0.40%	64.94% ↓
GINI indeks	68.44% +/- 0.46%	65.55% ↑
točnost	62.33% +/- 3.71%	59.90% ↑

Iz prikupljenih podataka se može primijetiti kako u odnosu na druge učeće operatore *Random Forest* postiže bolje rezultate točnosti klasifikacije u kombinaciji s *Bagging* ansambl operatorom. Primjenom *X-Validation* operatora u kombinaciji s nekim od ansambl operatora rezultira povećanjem broja iteracija treniranja učećeg operatora. Pri svakoj od 10 iteracija rada *X-Validation* operatora provodi se 10 iteracija rada *Bagging* ili *AdaBoost* ansambl operatora. Svakom iteracijom rada *Bagging* ili *AdaBoost* ansambl operatora se vrši treniranje učećeg operatora. U slučaju *Random Forest* učećeg operatora to rezultira generiranjem 2000 stabala odlučivanja.

Primjenom *Random Forest* operatora na proces sa slike 5.9. testirana je predikcija ciljnog atributa podjelom podataka na potpune i nepotpune. Treniranjem i primjenom dvaju modela postignuti su sljedeći rezultati točnosti klasifikacije dani tablicom 5-27:

Tablica 5-27. *Rezultati rada primjenom dva modela, Random Forest*

Podaci:	**Kriterij:**	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
Poznati	*omjer dobiti*	65.57% +/- 3.93%	67.17%
	informacijska dobit	68.96% +/- 1.12%	68.34%
	GINI indeks	66.94% +/- 3.51%	67.40%
	točnost	63.07% +/- 4.86%	66.11%
Nepoznati	*omjer dobiti*	62.49% +/- 4.87%	64.95%
	informacijska dobit	66.34% +/- 1.45%	68.37%
	GINI indeks	66.05% +/- 1.06%	69.69%
	točnost	58.55% +/- 1.50%	53.25%

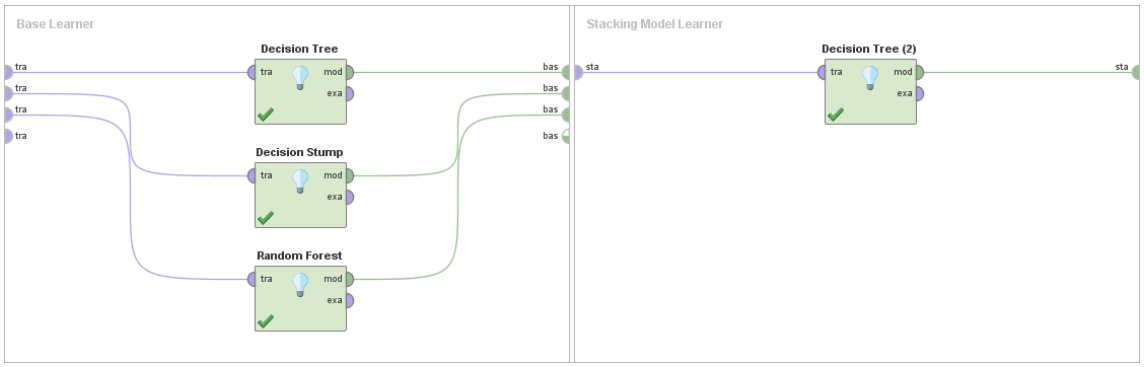
Iz tablice 5-27 se može zaključiti kako se primjenom dvaju modela postižu bolji rezultati točnosti klasifikacije za potpune podatke u odnosu na nepotpune što nije bio slučaj za *Decision Tree* i *Decision Stump* učeće operatore. Odabirom kriterija informacijske dobiti postiže se najbolji rezultat točnosti klasifikacije za *Random Forest* učeći operator.

5.4. *Stacking, Vote, Gradient Boosted Trees* i *Forward Selection*

Kombiniranjem više ansambl operatora povećavaju se vremenski zahtjevi izvođenja procesa. *Gradient Boosted Trees* ansambl operator zbog kompleksnosti nije moguće ugnijezditi u *AdaBoost* ansambl operator. Razlog čega je povećanje vremena treniranja s par sati na nekoliko desetaka sati. Iz tog razloga *Gradient Boosted Trees* ansambl operator nije korišten u kombinaciji s drugim ansambl operatorima.

Osim tri prethodno korištena ansambl operatora s potprocesom ostaju još samo dva specifična ansambl operatora koji ne generiraju model iteracijama rada učećih operatora već se oslanjaju na glasovanje i grupiranje modela.

Prvi u nizu specifičnih ansambl operatora je *Stacking* koji je, kao i mnogi drugi, ugniježđeni ansambl operator, sadrži potproces podijeljen u dva segmenta – *Base Learner* segment sadrži učeće operatore dok *Stacking Model Learner* segment sadrži jedan učeći operator kojim se generirani modeli *Base Learner* segmenta povezuju u jedan model. Primjer potprocesa *Stacking* ansambl operatora je prikazan slikom 5.12.



Slika 5.12. *Potproces Stacking ansambl operatora*

Base Learner segment sadrži tri od korištenih osnovnih učećih operatora – *Decision Tree, Decision Stump* te *Random Forest*. *Stacking Model Learner* koristi *Decision Tree* za spajanje modela u jedan zajednički.

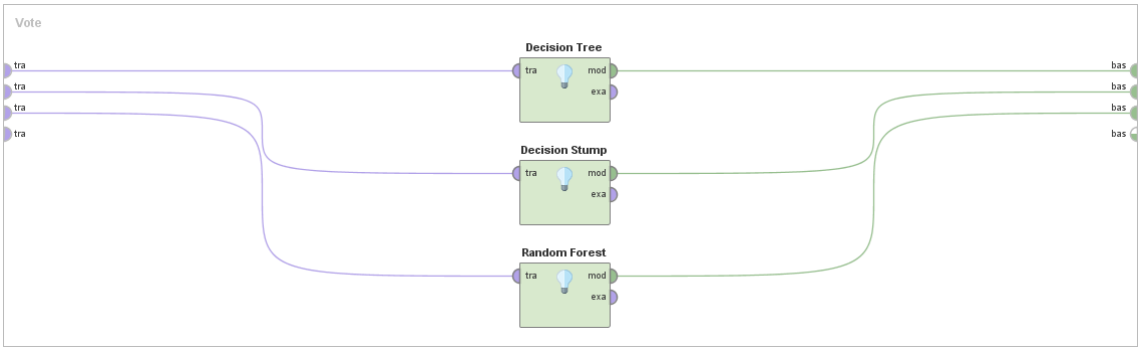
Ishod klasifikacije primjenom *Stacking* ansambl operatora dan je tablicom 5-28.

Tablica 5-28. *Rezultat Stacking ansambl operatora*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	73.30% +/- 0.33%	73.13%
informacijska dobit	72.79% +/- 0.20%	72.56%
GINI indeks	72.76% +/- 0.17%	72.57%
točnost	72.86% +/- 0.23%	73.14%

Rezultati *Stacking* ansambl operatora nisu usporedivi s prethodnim primjerima pošto je riječ o kombiniranju više modela treniranih različitim operatorima. *Stacking* nastoji generalizirati model tako da ishod klasifikacije bude optimalan.

Uz spomenute ansambl operatore, provedeno je testiranje *Vote* ansambl operatora koji izglasavanjem odabire najefikasniji model za primjenu na nepoznate podatke. *Vote* potproces sadrži tri učeća operatora prikazana slikom 5.13.



Slika 5.13. *Potproces Vote ansambl operatora*

Rezultati postignuti primjenom *Vote* ansambl operatora su dani tablicom 5-29.

Tablica 5-29. *Rezultat Vote ansambl operatora*

Kriterij:	**Rezultat *X-Validation* operatora:**	**Točnost klasifikacije:**
omjer dobiti	68.11% +/- 0.47%	67.93%
informacijska dobit	68.02% +/- 0.51%	67.88%
GINI indeks	68.02% +/- 0.51%	67.88%
točnost	63.32% +/- 0.50%	63.01%

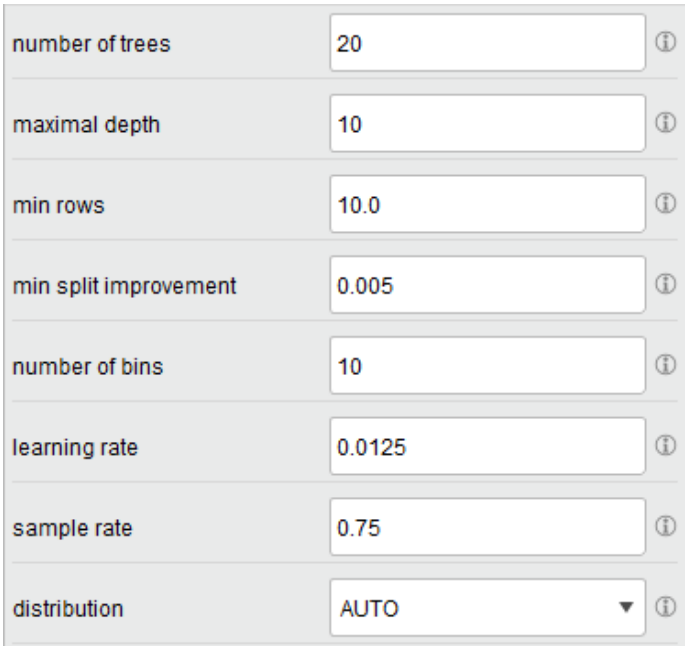
Vote ansambl operator na temelju predikcije pojedinog modela donosi odluku o konačnoj vrijednosti ciljnog atributa – ako više od polovice modela ima isti rezultat predikcije, *Vote* taj ishod predikcije označava kao konačnu odluku klasifikacije. Pojedine odluke klasifikacije se temelje na predikciji generiranih modela *Decision Tree*, *Decision Stump* te *Random Forest* učećih operatora.

Gradient Boosted Trees je još jedan u nizu testiranih ansambl operatora koji generira skup stabala odlučivanja povećane točnosti klasifikacije posljedica čega je usporenje procesa i smanjenje mogućnosti interpretiranja. Rezultat postignut primjenom spomenutog ansambl operatora je dan tablicom 5-30 u nastavku.

Tablica 5-160. *Rezultat rada Gradient Boosted Trees ansambl operatora*

Rezultat *X-Validation* operatora:	**Točnost klasifikacije:**
75.58% +/- 0.19%	75.50%

Podešavanjem parametara ansambl operatora, principom pokušaja i pogreške, otkriveni su optimalni parametri prikazani slikom 5.14 za koje se postiže najbolji rezultat točnosti klasifikacije.



Slika 5.14. *Parametri Gradient Boosted Trees ansambl operatora*

Gradient Boosted Trees podešen po parametrima sa slike 5.14. generira 20 stabala za svaku klasu ciljnog atributa. Na temelju korištenih parametara, ansambl operator je generirao 1600 stabala odlučivanja. Osim podataka o točnosti klasifikacije i grafova stabala *Gradient Boosted Trees* ansambl operator analizira važnost pojedinog atributa pri formiranju stabla. Lista važnosti atributa je prikazana slikom 5.15. u nastavku.

Variable Importances:

Variable	Relative Importance	Scaled Importance	Percentage
Campaign	1270159.625000	1.000000	0.734893
AdGroup	98795.406250	0.077782	0.057161
AdBannerType	95016.421875	0.074807	0.054975
Advertiser	83788.101563	0.065967	0.048478
Publisher	76654.851563	0.060351	0.044351
DateHour	40673.187500	0.032022	0.023533
Site	26620.173828	0.020958	0.015402
DeviceModel	13037.965820	0.010265	0.007544
AdPlace	8393.873047	0.006609	0.004857
ISP	4580.389648	0.003606	0.002650

Categories	1872.343628	0.001474	0.001083
Impressions	124.235626	0.000098	0.000072
CampaignTime	68.720268	0.000054	0.000040
Country	68.052330	0.000054	0.000039
CampaignTimeStart	45.962158	0.000036	0.000027
SiteType	42.689945	0.000034	0.000025
CampaignTimeEnd	13.187911	0.000010	0.000008
DeviceOs	1.886814	0.000001	0.000001
Clicks	0.000005	0.000000	0.000000
DeviceType	0.000000	0.000000	0.000000

Slika 5.15. *Važnost atributa*

Na temelju popisa važnosti atributa postavlja se pitanje: „Koliko je najviše atributa potrebno kako bi se postigla najveća točnost klasifikacije?“. Budući da je razlika u dobivenim rezultatima često zanemarivo mala (manje od 0.50%), postoji mogućnost da svi učeći operatori koriste isti osnovni skup atributa za koje postižu najbolji rezultat točnosti klasifikacije. Time bi se moglo objasniti zašto ne postoji značajnije poboljšanje tj. pogoršanje rezultata točnosti klasifikacije. Kako bi se pronašao odgovor na postavljeno pitanje testiran je *Forward Selection* operator koji izdvaja atribute na temelju kojih se može postići najveća točnost klasifikacije. Ovim postupkom se može značajno smanjiti broj atributa – izdvajaju se samo najbitniji atributi, ostali atributi samo unose šum pri treniranju stabla odlučivanja.

Forward Selection funkcionira na način da postepeno povećava broj korištenih atributa sve dok se dodavanjem novih atributa povećava točnost klasifikacije. Budući da je *Forward Selection* ugniježđeni operator potrebno je koristiti učeći operator u njegovom potprocesu. Ovisno o korištenom učećem operatoru vremenski zahtjevi se mijenjaju. Kako bi se otkrio dovoljan skup atributa za treniranje stabla korišten je *Decision Tree* učeći operator u potprocesu *Forward Selection* operatora. Radom operatora je otkriveno kako je za treniranje stabla dovoljno šest atributa – *CampaignTimeStart, AdBannerType, DateHour, Categories, Fingerprint* te *Ad*. Različiti učeći operatori daju različite rezultate, no zbog velikih vremenskih zahtjeva *Forward Selection* operatora korišten je isključivo *Decision Tree* učeći operator.

6. ZAKLJUČAK

Stalnim porastom korisnika Internet usluga javlja se potreba za novim metodama oglašavanja. U tu svrhu se razvijaju sustavi koji ciljano traže potencijalne potrošače na temelju prijašnjih saznanja o *online* navikama korisnika. Glavni zadatak spomenutih sustava je na efikasan i efektivan način privući korisnike na stranice oglašivača. Cilj ovog diplomskog rada je primjenom strojnog učenja, temeljenog na stablima odlučivanja, realizirati sustav za dostavu prikladnih oglasa korisniku. Za treniranje stabala odlučivanja korišteni su višedimenzionalni podaci koji sadrže 24 atributa s informacijama o korisnicima i prikazanim oglasima.

Za kreiranje takvog sustava tj. modela predviđanja oglasa korišten je programski paket *RapidMiner* koji nudi mnoštvo operatora kojima je moguće trenirati stabla odlučivanja. Neki od primijenjenih operatora su *Decision Tree, Decision Stump, Random Forest, AdaBoost, Bagging* te mnogi drugi. Testiranjem različitih operatora zaključeno je kako najbolje rezultate daje *Gradient Boosted Trees* operator. Poboljšanje u točnosti predviđanja oglasa u odnosu na *Decision Tree* učeći operator je približno 2%. Nažalost nije bilo moguće provesti daljnja testiranja spomenutog operatora u kombinaciji s drugim ansambl operatorima zato što vremenski zahtjevi drastično rastu. No, rezultat od 75.50% točno klasificiranih oglasa je vrlo dobar. Ujedno je potrebno naglasiti kako podatkovni skup sadrži velik broj atributa koji ne doprinose točnosti predikcije oglasa. Naime, primjenom *Forward Selection* operatora koji eliminira one atribute čijim uključivanjem se ne doprinosi točnosti predviđanja, može se zaključiti kako je za najbolje rezultate predviđanja dane u prethodnom poglavlju potrebno od četiri do osam atributa.

Uzevši u obzir formu podataka – opći oblik klasa atributa tj. klase atributa *Ad* imaju opći oblik *Ad* popraćen numeričkom oznakom (npr. *Ad30, Ad80*), ne može se ni na koji način odredi što se tim oglasom prikazuje. Osim spomenutog atributa istu formu prate i atributi *Advertiser, Campaign, AdGroup, Publisher, Site, Zone* i *AdPlace*. Primjenom *Forward Selection* operatora se pokazalo kako upravo ti atributi, izuzev atributa *Ad* koji je ciljni atribut, ni na koji način ne doprinose poboljšanju točnosti predikcije. *RapidMiner* ima mogućnost pretvorbe tekstualnih podataka u numeričke, no po završetku konverzije nije bilo moguće trenirati stablo odlučivanja zbog velikih memorijskih zahtjeva zbog čega taj pokušaj nije ni evidentiran. Daljnje testiranje nakon konverzije se preporuča.

Rezultati točnosti predviđanja su vrlo visoki budući da je riječ o komercijalnim podacima. Postoji mogućnost povećanja točnosti predviđanja ako bi se povećala specifičnost pojedinih atributa (npr. Promjena klasa atributa *AdGroup* iz oblika *AdGroup1, AdGroup2,* itd. u specifične nazive *Hardware, Software* i slično).

LITERATURA

[1] Kumar, A.. (2016). Types of Online Ad Formats [online]. Ads Nets Review. Dostupno na: http://adnetsreview.com/types-of-online-ad-formats/ [21.6.2016.]

[2] (2011) Mobile Advertising Guidelines [online]. Mobile Marketing Association. Dostupno na: http://www.mmaglobal.com/files/mobileadvertising.pdf [21.6.2016.]

[3] Seven Different Ad Types [online]. AdAgency. Dostupno na: https://adagency.ijoomla.com/seven-different-ad-types [21.6.2016.]

[4] IAB Display Advertising Guidelines. Interactive Advertising Bureau. Dostupno na: http://www.iab.com/guidelines/iab-display-advertising-guidelines/ [21.6.2016.]

[5] (2008) What is CPM, CPC, CPA and CTR ? – Basics of Mobile Publishing [online]. Mobile Advertising Really Works. Dostupno na: http://bit.ly/28SNmvg [22.6.2016.]

[6] Ad Server [online]. Know Online Advertising. Dostupno na: http://www.knowonlineadvertising.com/ad-server/ [23.6.2016.]

[7] SEM Glossary of Terms [online]. SEMPO. Dostupno na: http://www.sempo.org/?page=glossary [24.6.2016.]

[8] What is an Ad Server – FEATURES OF AN AD SERVER [online]. Know Online Advertising. Dostupno na: http://www.knowonlineadvertising.com/ad-server/what-is-an-ad-server/ [29.6.2016.]

[9] Getting Started Glossary – RapidMiner data types [online]. RapidMiner Documentation. Dostupno na: http://docs.rapidminer.com/studio/getting-started/important-terms.html [1.7.2016.]

[10] Verma, R.. (2009). The Data Mining Hypertextbook: Decision Tree Induction Algorithms [online]. Dostupno na: http://bit.ly/2bum5Uh [2.7.2016.]

[11] Raschka, S.. Machine Learning FAQ: Why are implementations of decision tree algorithms usually binary and what are the advantages of the different impurity metrics? [online]. Dostupno na: http://sebastianraschka.com/faq/docs/decision-tree-binary.html [2.7.2016.]

SAŽETAK

Na području Internet oglašavanja često se javlja problem pronalaska efektivne i efikasne metode dostave oglasa potencijalnim korisnicima. Uobičajeno rješenje tog problema je primjena specijaliziranih sustava za oglašavanje kojima se nastoji dostavljati oglase na temelju prikupljenih informacija o korisniku.

Cilj ovoga rada je primjenom stabala odlučivanja, iterativnih algoritama strojnog učenja, realizirati sustav koji na temelju prikupljenih informacija o prijašnjim korisnicima optimizira prikazivanje oglasa budućim korisnicima.

Različitim pristupima treniranju stabla odlučivanja prikupljeni su rezultati za operatore *Decision Tree, Decision Stump* i *Random Forest* samostalno te u kombinaciji s ansambl operatorima *AdaBoost* i *Bagging*. Također su dani rezultati za *Vote, Stacking* i *Gradient Boosted Trees* ansambl operatore te za *Forward Selection* operator koji daje bolji uvid u korisnost pojedinih atributa pri treniranju stabala odlučivanja.

Analizom rezultata i prikupljenih podataka predlažu se metode kojima bi se povećala točnost sustava temeljenog na stablima odlučivanja.

SUMMARY

In the field of Internet advertising often emerges a problem of finding effective and efficient methods of ad delivery to potential customers. Typically the solution to this problem is the use of specialized systems for advertising where ad delivery is based on the information collected about the user.

The objective of this paper is to create a system by using decision trees and iterative machine learning algorithms that, based on the information gathered about previous customers, optimizes ad delivery to the future customers.

Different approaches for training the decision trees are used and results are collected for Decision Tree, Decision Stump and Random Forest operators alone and in their combination with the ensemble operators such as AdaBoost and Bagging. The results are also provided for the Vote, Stacking and Gradient Boosted Trees ensemble operators and the Forward Selection operator which provides better insight into the usefulness of certain attributes during the training of decision trees.

By analyzing the results and the collected data, methods are proposed which would increase the accuracy of a system based on decision trees.

ŽIVOTOPIS

Dominik Babić rođen je 2. veljače 1993. godine. Osnovnu školu Augusta Cesarca u Ivankovu je pohađao od 1999. do 2007. godine. Od 2007. do 2011. godine je pohađao Tehničku školu Ruđera Boškovića u Vinkovcima, smjer elektrotehnike. Godine 2011. upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu sveučilišta Josipa Jurja Strossmayera u Osijeku koji i završava 2014. godine. Po završetku preddiplomskog studija upisuje diplomski studij računarstva na istom sveučilištu, smjer procesno računarstvo.