UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA

ZAVRŠNI RAD

Tema: Klasterovanje potencijalnih korisnika osiguranja korišćenjem samoorganizujućih mapa

Mentor:
dr Ivana Dragović

Student:
Jovan Petrović 30/16

Beograd, 2020. godine

Klasterovanje potencijalnih korisnika osiguranja korišćenjem samoorganizujućih mapa

Apstrakt

Ovaj rad se bavi analizom karakteristika korisnika osiguravajućih usluga kroz problem klasifikacije i klasterovanja kako bi se utvrdili potencijalni korisnici osiguranja karavan vozila. Analiza osiguranika je neophodni segment rada bilo kog osiguravajućeg društva, kako zbog identifikacije i smanjenja šansi za prevare kod različitih tipova osiguranja, tako i zbog redukcije troškova marketinga pomoću efektivnije promocije i pravljenja različitih biznis strategija koje odgovaraju raznovrsnim grupama korisnika. Projekat na osnovu koga je zasnovan ovaj rad nastao je u vidu saradnje kompanije *msg global solutions* i Katedre za upravljanje sistemima na Fakultetu organizacionih nauka. Korišćen je skup podataka *Insurance Company Benchmark (COIL 2000)* preuzet sa „UCI machine learnig" repozitorijuma (http://kdd.ics.uci.edu/databases/tic/tic.html). Podaci iz ovog skupa podataka su pretprocesirani čime se došlo do odgovarajućih ulaznih podataka za klasifikator. Podaci se pre svega odnose na socio-demografske karakteristike, kao i na korišćenja drugih već postojećih osiguranja. Glavni cilj ovog rada je da se segmentira deo već postojećih korisnika, koji bi bio zainteresovan za kupovinu polise osiguranja karavan vozila, putem različitih metoda klasifikacije. Za potrebe ovog rada korišćena je vrsta veštačke neuronske mreže pod nazivom samoorganizujuće mape. Primenom ove mreže na dati problem došlo se do vrednosti metrike senzitivnost od 51%. Da bi se stekao bolji uvid u kvalitet ovog modela, klasifikacija je odrađena i pomoću modela potpornih vektora i *k* najbližih suseda, čime su vrednosti senzitivnosti dodatno poboljšane i iznosile su 61%, odnosno 74%. Drugi cilj ovog rada je segmentacija korisnika sličnih karakteristika u određen broj grupa kada je u pitanju njihov prihod. Dobijena su tri klastera koja su napravila jasnu razliku između različitih grupa korisnika.

Ključne reči: *klasifikacija, klasterovanje, neuronska mreža, osiguranje, karavan, samoorganizujuće mape, model potpornih vektora, k najbližih suseda*

Clustering of potential insurance customers using self-organising maps

Abstract

This paper attempts to analyze the characteristics of insurance customers by means of clustering and classifying them with regards to whether or not they purchased an insurance policy for their caravan. Analyzing the customer is an essential part of business for all insurance companies in order to identify and lower the chances of fraud for different types of insurance, but also to reduce marketing costs through a more effective promotion and various business strategies that each apply to a specific group of customers. This paper arose from a project that the company *msg global solutions* and the Department of Systems Theory and Control at the Faculty of Organizational Sciences have cooperated on. The data set which the paper uses belongs to Insurance Company Benchmark (COIL 2000), which was retrieved from their UCI machine learning repository (http://kdd.ics.uci.edu/databases/tic/tic.html). The data was preprocessed in such a way that revealed new input data for classification. It is mostly associated with customers' socio-demographic characteristics as well as their usage of other insurance policies. The main goal of this paper is to segment the existing base of customers who are interested in purchasing insurance for their caravan through various classification methods. The paper makes use of an artificial neural network called self-organizing map. Applying this network to the problem at hand achieved a sensitivity of 51%. In order to gain a better insight into the value of this model, classification was also performed using the support vector machine and the *k*-nearest neighbors algorithm. This way the sensitivity was further improved, reaching 61% and 74% respectively. Another goal of this paper is to segment customers who share similar characteristics into groups based on their income. This provided three clusters which revealed a clear difference between various groups of customers.

Keywords: classification, clustering, neural network, insurance, caravan, self-organizing maps, support vector machine, k-nearest neighbors

Biografija

Jovan Petrović rođen je 9.1.1998. godine u Čačku. Osnovnu školu „Emilija Ostojić“ u Požegi završio je sa Vukovom diplomom, a srednju školu Gimnazija „Sveti Sava“ u Požegi završio je sa odličnim uspehom. Fakultet organizacionih nauka u Beogradu, smer Informacioni sistemi i tehnologije (ISiT), upisuje 2016. godine. Položio je sve ispite sa prosečnom ocenom 9,49. Govori engleski jezik. Voli da istražuje različite programske jezike, a najviše da se bavi objekto-orijentisanim programiranjem u Javi. Od druge godine član je studentske organizacije FONIS gde je slušao različite kurseve vezane za programiranje i mašinsko učenje i bavio se organizacijom projekata. U četvrtoj godini postaje student demonstrator na Katedri za upravljanje sistemima, da bi krajem četvrte godine radio na projektu koji je nastao u saradnji kompanije *msg global solutions* i Katedre za upravljanje sistemima. U drugoj godini studiranja bio je dobitnik Studentske stipendije, u trećoj Stipendije za izuzetno nadarene studente, a u četvrtoj Dositejeve stipendije. U slobodno vreme voli da se bavi sportom, gleda filmove i izučava astrofiziku.

Sadržaj

1. Uvod ... 1
2. Primena računarske inteligencije u industriji osiguranja ... 3
3. Osnove veštačkih neuronskih mreža ... 5
   3.1. Istorijski pregled veštačkih neuronskih mreža ... 5
   3.2. Inspiracija iz biologije ... 6
   3.3. Model neurona ... 7
      3.3.1. Model neurona sa jednim ulazom ... 7
      3.3.2. Transfer funkcije ... 8
      3.3.3. Model neurona sa više ulaza ... 10
   3.4. Samoorganizujuće mape ... 11
      3.4.1. Nenadgledane samoorganizujuće mape ... 11
      3.4.2. Nadgledane samoorganizujuće mape ... 14
4. Opis konkretnog problema ... 15
5. Analiza i pretprocesiranje podataka ... 16
   5.1. Struktura podataka ... 16
   5.2. Eksploratorna analiza podataka ... 18
      5.2.1. Univarijaciona eksploratorna analiza ... 18
      5.2.2. Multivarijaciona eksploratorna analiza ... 21
   5.3. Redukcija atributa ... 23
      5.3.1. Filter metode ... 24
      5.3.2. Obmotavajuće metode – Boruta algoritam ... 26
6. Eksperiment ... 30
   6.1. Klasterovanje korišćenjem samoorganizujućih mapa ... 30
   6.2. Klasifikacija korišćenjem samoorganizujućih mapa ... 37
   6.3. Poređenje rezultata samoorganizujućih mapa sa rezultatima metode potpornih vektora i k najbližih suseda ... 42
7. Zaključak ... 45
8. Literatura ... 47

Prilog: Kod ... 50

Spisak slika

Slika 1: Prikaz biološkog neurona 7
Slika 2: Model neurona sa jednim ulazom 8
Slika 3: Odskočna funkcija..... 9
Slika 4: Linearna funkcija 9
Slika 5: Logsigmoidna funkcija..... 9
Slika 6: Model neurona sa više ulaza.....10
Slika 7: Višeslojna neuronska mreža 11
Slika 8: Okruženje neurona 12
Slika 9: Samoorganizujuća mapa sa vektorom ulaza i težinama.....12
Slika 10: Odnos korisnika koji su kupili osiguranje karavan vozila i onih koji nisu.....17
Slika 11: Ulaganje u osiguranje automobila u odnosu na izlaznu promenljivu 22
Slika 12: Ulaganje u osiguranje od požara u odnosu na izlaznu promenljivu.....23
Slika 13: Tabela korelacije.....25
Slika 14: Grafička reprezentacija boruta algoritma.....27
Slika 15: Grafička reprezentacija kretanja značajnosti atributa kroz iteracije 28
Slika 16: Udeo atributa u svakom klasteru.....32
Slika 17: Trening proces modela samoorganizujuće mape.....33
Slika 18: Broj opservacija pridodat svakom neuronu.....34
Slika 19: Udaljenost neurona od ulaznog vektora.....34
Slika 20: Hitmape.....35
Slika 21: Metoda lakta 36
Slika 22: Klasterovana mapa 36
Slika 23: Promene srednje distance do najbližeg neurona.....40
Slika 24: Nenadgledana i nadgledana mapa 41
Slika 25: Preciznost i senzitivnost samoorganizujućih mapa 42
Slika 26: Preciznost i senzitivnost svih modela.....43

Spisak tabela

Tabela 1: Značenje vrednosti socio-demografskih atributa.....19
Tabela 2: Deskriptivne statistike socio-demografskih atributa.....19
Tabela 3: Značenje vrednosti atributa za novčano ulaganje u osiguranja.....20
Tabela 4: Deskriptivne statistike atributa vezanih za ulaganje u osiguranja.....20
Tabela 5: Deskriptivne statistike atributa vezanih za broj različitih osiguranja.....21
Tabela 6: Značaj atributa tokom iteracija.....28
Tabela 7: Matrica konfuzije 37
Tabela 8: Evaluacione metrike za svih 9 modela.....42

1. Uvod

Sektor osiguranja se sastoji od osiguravajućih društava čija je osnovna uloga da svojim korisnicima pomognu da upravljaju rizikom poštujući formu polise osiguranja. Osnovni koncept osiguranja je da jedna strana, u ovom slučaju osiguravajuće društvo, garantuje isplatu ukoliko se ostvari određeni nepovoljni događaj, opisan u polisi osiguranja. U međuvremenu, druga strana, osiguranik, plaća male iznose tokom propisanog perioda vremena u zamenu za zaštitu od potencijalnih nepovoljnih događaja u budućnosti (Investopedia, 2020).

Društva za osiguranje imaju i investicionu ulogu. Zakonom je takođe propisano na koji način društva za osiguranje deo prikupljenih premija mogu ulagati u hartije od vrednosti, nekretnine, depozite, itd. kako bi ostvarila veće prihode i obezbedila stabilno, pozitivno i dugoročno poslovanje.

Svakako da je osnovni cilj svakog osiguravajućeg društva maksimiziranje profita, a samim tim i što veći broj korisnika osiguranja. Jedan od najefikasnijih načina za povećanje broja korisnika u bilo kojoj uslužnoj ili proizvodnoj kompaniji je marketing strategija koja se naziva unakrsna prodaja (eng. cross-selling). Unakrsna prodaja podrazumeva podsticanje postojećih kupaca da uz jedan proizvod ili uslugu kupe drugi srodan ili komplementaran proizvod ili uslugu s ciljem da se poveća prodaja i produbi odnos sa kupcem (Marketing Fancier, 2019).

U 2000. godini evropska osiguravajuća kompanija koja nudi mnoštvo različitih vrsta osiguranja suočila se sa izazovom unakrsne prodaje kada se kompanijin najnoviji tip osiguranja karavan vozila pokazao razočaravajućim u pogledu prodaje. Marketing sektor kompanije je smatrao da bi podsticanje već postojećih osiguranika da kupe osiguranje karavan vozila značajno unapredilo prodaju. Međutim, glavno pitanje je bilo kako segmentirati korisnike koji bi bili zainteresovani za ovaj tip osiguranja među nekoliko hiljada osiguranika i na koji način ostvariti što veći prihod uz što manje troškove marketinga.

Jedan od načina za targetiranje osiguranika jeste korišćenje metoda računarske inteligencije. Računarska inteligencija predstavlja deo nauke koji omogućava računarima da uče, bez potrebe da za to budu eksplicitno programirani (Samuel, 1959, str. 210-229). Takođe, računarska inteligencija može da se tumači kao sposobnost softverskog sistema da generalizuje na osnovu prethodnog iskustva i da potom koristi kreirane generalizacije kako bi pružio odgovore na pitanja koja se tiču entiteta/pojava koje pre nije sretao (Jovanović, 2016, str. 3).

Kao i u mnogim drugim proizvodnim ili uslužnim kompanijama, tako i u osiguravajućim društvima metode računarske inteligencije koje se tiču klasifikacije korisnika u one koji su zainteresovani za kupovinu određenog proizvoda, odnosno korišćenje određene usluge, i one koji nisu se jako često koriste. Pravilno klasifikovanje potencijalnih osiguranika može biti izrazito korisno u fazi promocije i na taj način u mnogome umanjiti troškove marketinga. Pored problema klasifikacije, čest slučaj u ovakvim kompanijama je želja menadžmenta da različitim tipovima kupaca pristupi na različit način, tako što će razviti posebnu biznis strategiju za svaku grupu kupaca sličnih karakteristika. Ove grupe moguće je odrediti različitim tehnikama klasterovanja.

Cilj ovog rada je da se izvrši binarna klasifikacija postojećih osiguranika, odnosno podela u dve grupe, na osnovu karakteristika primećenih u podacima, a koja će dovesti do zaključka da li je određeni korisnik zainteresovan za kupovinu osiguranja karavan vozila. Pored toga, tema rada biće i pronalaženje grupa kupaca sličnih karakteristika, kako bi se svakoj grupi moglo pristupiti na jedinstven način.

U narednom poglavlju biće detaljnije objašnjeno zašto računarska inteligencija ima veliku primenu u oblasti osiguranja. Biće navedeni najčešći problemi koji se mogu rešiti i predstavljena neka od rešenja koja su prisutna u industriji.

U trećem poglavlju biće objašnjeno šta su to neuronske mreže i na koji način su se razvijale kroz istoriju. Pored toga, biće predstavljeni neki od modela neuronskih mreža, dok će akcenat biti stavljen na samoorganizujuće mape koje će biti korišćene u samom eksperimentu.

Četvrto poglavlje služi za upoznavanje sa samim projektom. Biće opisan konkretan problem koji je rešavan.

U petom poglavlju biće detaljnije ispitana struktura korišćenog skupa podataka. Pokušaće da se utvrde određeni paterni i da se bolje razume njihov značaj. Biće opisane različite tehnike redukcije atributa i njihovom primenom originalni skup podataka biće sveden na finalni koji će se koristiti u eksperimentu.

Šesto poglavlje obuhvatiće sam eksperiment i u njemu će biti prikazani i detaljnije analizirani rezultati.

U sedmom poglavlju biće ukratko sumirano sve ono što je odrađeno u ovom radu.

Osmo poglavlje sadržaće listu referenci dela korišćenih prilikom izrade ovog rada.

2. Primena računarske inteligencije u industriji osiguranja

Računarksa inteligencija u oblasti industrije osiguranja je značajno unapredila menadžment upravljanja potraživanjima čineći ga bržim, kvalitetnijim i preciznijim. Industrija osiguranja se oduvek oslanjala na podatke prilikom računanja rizika, a u današnje vreme prolazi kroz period digitalne transformacije zahvaljujući tehnologijama koje koriste računarsku inteligenciju. U narednom delu biće navedeno nekoliko glavnih načina na koje računarska inteligencija transformiše industriju osiguranja (Negturu, n.d.).

Osiguravajuće kuće obrađuju hiljade potraživanja i odgovaraju na još više upita korisnika. Računarska inteligencija može da utiče na automatizaciju i poboljšanje procesa tako što će automatski slati potraživanja kroz sistem. U nekim situacijama, moguće je u potpunosti zaobići potrebu za ljudskom random snagom. Jedna od najbrže rastućih američkih osiguravajućih kompanija "Lemonade" koristi računarsku inteligenciju da obrađuje potraživanja dosta brže i obezbeđuje korisnicima brze isplate koristeći različite aplikacije kao što je četbot (Negturu, n.d.).

Rangiranje, odnosno određivanje cene predstavlja osnovu u oblasti osiguranja. Postoji poznata izreka u svetu osiguranja koja kaže da ne postoje loši rizici, samo loše procene cene. Ipak, i dalje postoje mnoge osiguravajuće kuće koje se oslanjaju na tradicionalne metode evaluiranja rizika. Računarska inteligencija može da ponudi menadžmentu nove alate i metode za klasifikovanje rizika i pravljenje tačnijih prediktivnih modela cene koji će uticati na smanjenje gubitaka. Dobar primer predstavlja "Zendrive", mobilna aplikacija koja prati ponašanje korisnika tokom vožnje da bi im kasnije bio ponuđen značajni popust tokom kupovine osiguranja automobila (Negturu, n.d.).

Korisnici od kompanija očekuju personalizovane servise koje odgovaraju njihovim potrebama, preferencijama i životnom stilu. Zbog toga ne treba puno govoriti o značaju izvlačenja zaključaka o korisnicima i njihovim karakteristikama iz podataka. Sve ovo je moguće zahvaljujući algoritmima računarske inteligencije, gde posebno do izražaja dolaze tehnike nenadgledanog učenja (Negturu, n.d.).

Prevare su jedna od većih briga u industriji osiguranja. Samo u Sjedinjenim Američkim Državama prevare koštaju ovaj sektor preko 40 miliona dolara godišnje (Federal Bireau of Investigation, 2010). Kada bi osiguravajuće kuće pronašle način da ublaže stepen prevara, to bi se direktno odrazilo na njihov profit. Upravo ovde računarska inteligencija stupa na scenu, tako što se određenim metodama klasifikacije mogu identifikovati potraživanja koja su posledica

ilegalnih akcija. Pariska kompanija “Shift Technology” pruža rešenje za bolje identifikovanje potencijalnih prevara u potraživanjima koje takođe omogućava jasna obrazloženja zašto je određeno potraživanje klasifikovano za prevaru i pruža najbolje sledeće korake za istraživanje slučaja (Negturu, n.d.).

3. Osnove veštačkih neuronskih mreža

Veštačka neuronska mreža (eng. artificial neural network) je računarski model koji se sastoji od nekoliko elemenata koji primaju ulaze i daju izlaze na osnovu predefinisanih aktivacionih funkcija (ScienceDirect, n.d.). Ovakvi modeli su napravljeni da mogu samostalno da „uče“ da obavljaju zadatke na osnovu primera, odnosno prethodnog iskustva i to bez prethodnog znanja o specifičnim karakteristikama konkretnog problema. Danas se neuronske mreže koriste za rešavanje raznih kategorija problema, uključujući klasifikaciju, regresiju, filterovanje, optimizaciju, prepoznavanje paterna i aproksimaciju funkcija (MathWorks, n.d.).

U nastavku teksta biće dat kratak istorijski pregled razvoja neuronskih mreža, njihova veza sa biološkim neuronima i prikaz veštačke neuronske mreže. Nakon toga, biće objašnjena konkretna vrsta veštačke neuronske mreže pod nazivom samoorganizujuća mapa koje će i biti korišćena za rešavanje problema.

3.1. Istorijski pregled veštačkih neuronskih mreža

Istorija veštačkih neuronskih mreža je popunjena mnogobrojnim istraživačima iz različitih oblasti nauke, od kojih su mnogi decenijama pokušavali da razviju koncepte koje mi danas shvatamo olako (Hagan, 2014, str. 1-2).

Moderno gledište na neuronske mreže počelo je 1940-ih godina radom Vorena Mekuloča (eng. Warren McCulloch) i Voltera Pitsa (eng. Walter Pitts) koji su pokazali da mreže veštačkih neurona mogu da predstave bilo koju aritmetičku ili logičku funkciju. Njihov rad se često posmatra kao početak rada iz oblasti neuronskih mreža (Hagan, 2014, str. 1-3).

Nakon njih značajan doprinos dao je Donald Heb (eng. Donald Hebb) koji je pokazao da klasično uslovljavanje postoji zbog svojstava pojedinačnih neurona. On je predložio mehanizam za učenje u biološkim neuronima (Hagan, 2014, str. 1-3).

Prva praktična primena veštačkih neuronskih mreža dogodila se u kasnim 1950-im godinama od strane Franka Rozenblata (eng. Frank Roseblatt). Rozenblat i njegove kolege su izgradili perceptron mrežu koja je bila sposobna za prepoznavanje paterna (eng. Pattern recognition). Njihov model je u mnogome povećao interesovanje za istraživanje neurosnkih mreža, ali se kasnije pokazalo da je bio limitiran na samo sužen skup problema koje je mogao da reši (Hagan, 2014, str. 1-3).

U isto vreme Bernard Vidrou (eng. Bernard Widrow) i Ted Hof (eng. Ted Hoff) pronalaze novi algoritam za učenje i koriste ga za učenje adaptivnih neuronskih mreža koje su bile slične strukture i mogućnosti kao Rozenblatov perceptron. Po njima se naziva Vidrou-Hofovo pravilo (eng. The Widrow-Hoff learning rule) koje se koristi i dan danas (Hagan, 2014, str. 1-3).

Nažalost, obe prethodno pomenute mreže su bile ograničene, tako da je u narednim decenijama došlo do pada interesovanja jer su mnogi istraživači smatrali da se neuronske mreže ne mogu unaprediti, pogotovo kada se pridoda i činjenica da tada nisu postojali dovoljno moćni računari za vršenje eksperimenata. Međutim i pored svih ograničenja i pada interesovanja, u 1970-im godinama su se desila neka značajna otkrića. 1972. godine su Tuvo Kohonen (eng. Teuvo Kohonen) i Džejms Anderson (eng. James Anderson) odvojeno razvili nove neuronske mreže koje mogu da se ponašaju kao memorija (Hagan, 2014, str. 1-3). Ovaj tip neuronskih mreža je danas poznat kao Kohonenove samoorganizujuće mape i upravo one će biti korišćenje tokom samog eksperimenta.

Od ostalih otkrića iz oblasti veštačkih neuronskih mreža definitivno treba izdvojiti algoritam propagacije unazad (eng. Backpropagation) korišćen za treniranje višeslojne perceptron mreže, koji i danas predstavlja jedan od najkorišćenijih načina treniranja višeslojnih neuronskih mreža. Ovaj algoritam je otkriven 1980-ih godina od strane više različitih istraživača. (Hagan, 2014, str. 1-4).

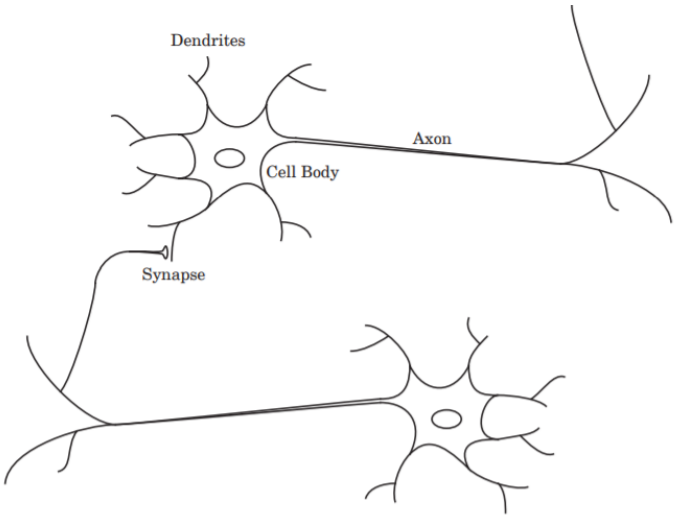
U poslednjim decenijama dolazi do značajnog povećanja računarske moći, pojavljuju se mnogobrojni istraživači sa različitim eksperimentima, a veštačke neuronske mreže se koriste u različitim oblastima (Hagan, 2014, str. 1-4).

3.2. Inspiracija iz biologije

Veštačke neuronske mreže su inspirisane biološkim procesima koje su naučnici proučavali tokom samog početka istraživanja ovog polja, tokom 1940-ih godina.

Mozak se sastoji iz velikog broja (približno 10^{11}) međusobno povezanih elemenata (približno 10^4 veza po elementu) koji se nazivaju neuroni. Za razumevanje veštačkih neuronskih mreža značajno je pomenuti tri glavna dela neurona, dendrite, telo neurona i akson. Dendriti su prijemne mreže nervnih vlakana koji omogućavaju telu neurona da primi električne signale. Telo neurona predstavlja samo jezgro neurona koje spaja pridošle signale. Akson predstavlja dugačko vlakno koje prenosi električni signal iz tela neurona u druge neurone. Tačka gde se akson jednog neurona spaja sa dendritima drugog naziva se sinapsa. Raspored

neurona i snaga pojedinačnih sinapsi uspostavljaju funkcionisanje neuronske mreže (Hagan, 2014, str. 1-8). Prikaz biološkog neurona dat je na slici 1.



Slika 1: Prikaz biološkog neurona[1]

Naravno, treba napomenuti da veštačke neuronske mreže nisu ni približno kompleksne kao mozak. Međutim, izdvajaju se dve ključne sličnosti između veštačkih i bioloških neurosnkih mreža:

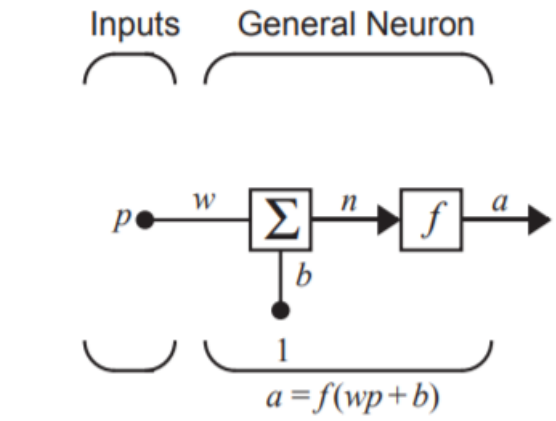
1) Obe mreže su izgrađene od jednostavnih računskih komponenti (iako su veštački neuroni mnogo jednostavniji od bioloških) koje su međusobno povezane.
2) Veze između neurona određuju način funkcionisanja same mreže.

3.3. Model neurona

3.3.1. Model neurona sa jednim ulazom

Ulaz u neuron predstavlja skalar *p* koji se množi skalarnom težinom *w* (eng. weigth). Na taj način se dobija proizvod *wp* koji se dalje šalje u sabirač. Drugi ulaz *1* množi se sa pristrasnošću *b* (eng. bias) i ovaj proizvod se takođe prosleđuje u sabirač. Izlaz iz sabirača *n*, koji se često naziva ulazom mreže (eng. net input), prosleđuje se u transfer funkciju *f* (eng. transfer function), koja daje skalarni neuronski izlaz (Hagan, 2014, str. 2-2). Prikaz neurona sa jednim ulazom dat je na slici 2.

[1] Preuzeto iz Hagan, 2014, str 1-8.



Slika 2: Model neurona sa jednim ulazom[2]

Izlaz iz neurona se računa prema sledećoj formuli:

$$a = f(w * p + b) \quad (1)$$

Kao što se može primetiti, izlaz zavisi od konkretne transfer funkcije koja je izabrana. O tranfer funkcijama biće više reči u sledećem poglavlju. Pristrasnost *b* je jako slična težini *w*, s tim što ima konstantan ulaz koj je jednak jedinici. Međutim, treba naglasiti da pristrasnost nije obavezan deo modela i može se isključiti ukoliko je to potrebno (Hagan, 2014, str. 2-3).

U najvećem broju slučajeva transfer funkcija se bira od strane dizajnera modela, a potom se parametri težina *w* i pristrasnost *b* prilagođavaju pomoću nekog pravila učenja kako bi veze između ulaza i izlaza dostigle odgovarajući cilj (Hagan, 2014, str. 2-3).

3.3.2. Transfer funkcije

Transfer funkcija može biti linearna ili nelinearna funkcija od *n*. Konkretna transfer funkcija se bira tako da zadovolji određene specifikacije problema koji neuron treba da reši (Hagan, 2014, str. 2-3).

Postoji veliki broj transfer funkcija koje se mogu upotrebljavati za rešavanje različitih problema. U nastavku teksta biće predstavljene tri koje se možda i najčešće koriste.

Hevisajdova ili odskočna funkcija prikazana je na slici 3. Izlaz neurona biće postavljen na nulu ukoliko je argument funkcije manji od *-b/w* ili jedinicu ukoliko je argument veći od istog količnika. Ova transformaciona funkcija se koristi da kreira neurone koji klasifikuju ulaze u dve različite kategorije (Hagan, 2014, str. 2-3).

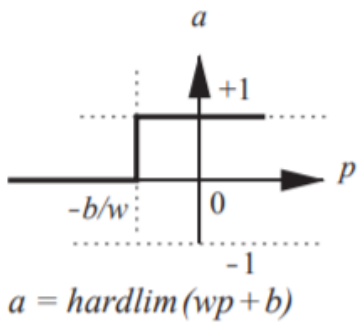
[2] Preuzeto iz Hagan, 2014, str. 2-2.

Linearna transfer funkcija je prikazana na slici 4, a koristi se u „adaline“ neuronskim mrežama (Hagan, 2014, str. 2-4).

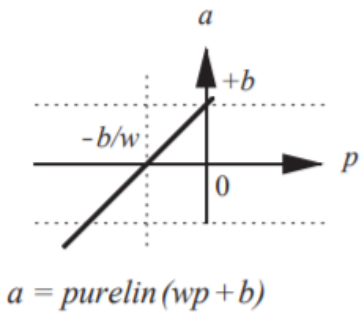
Logsigmoidna transfer funkcija od ulaza (kojim može biti bilo koja vrednost između plus i minus beskonačno) produkuje izlaz u rasponu od nula do 1 prema sledećoj jednačini

$$a = \frac{1}{1 + e^{-n}} \tag{2}$$

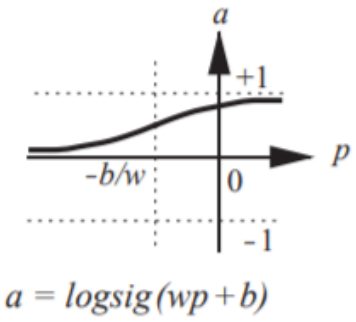
Logsigmoidna transfer funkcija je prikazana na slici 5 (Hagan, 2014, str. 2-4).



Slika 3: Odskočna funkcija[3]



Slika 4: Linearna funkcija[4]



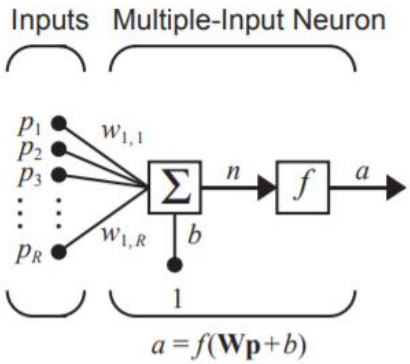
Slika 5: Logsigmoidna funkcija[5]

[3] Preuzeto iz Hagan, 2002, str. 2-4.
[4] Preuzeto iz Hagan, 2002, str. 2-4.
[5] Preuzeto iz Hagan, 2002, str. 2-5

3.3.3. Model neurona sa više ulaza

Mnogo češći slučaj u praksi je neuron sa više ulaza. Neuron sa *R* ulaza je prikazan na slici 6. Svaki ulaz $p_1, p_2, \ldots, p_R$ se množi odgovarajućom težinom $w_{1,1}$, $w_{1,2}$,...,$w_{1,R}$. Neuron sadrži pristrasnost b, koja se sabira sa otežanim ulazima, kako bi se formirao izlaz iz mreže *n*:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \ldots + w_{1,R}p_R + b \qquad (3)$$

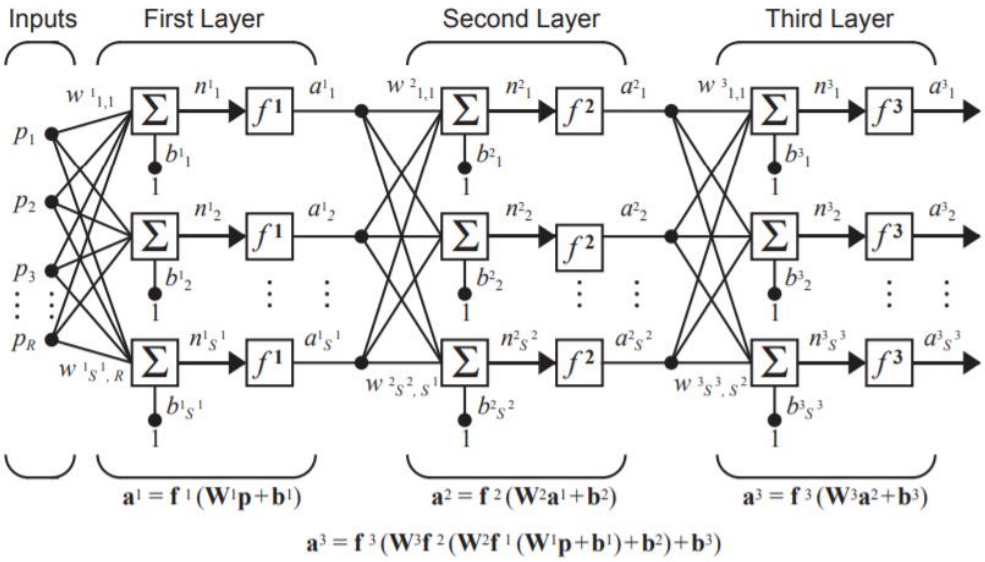


Slika 6: Model neurona sa više ulaza[6]

Snaga neuronskih mreža ogleda se u mogućnosti rešavanja kompleksnih problema. Jasno je da za takve probleme nije dovoljan samo jedan neuron niti samo jedan sloj mreže. Međutim spajanjem više neurona koji paralelno funkcionišu može se dobiti sloj, a kombinacijom više slojeva mogu se formirati vrlo kompleksne neuronske mreže.

Sloj neurona sadrži vektor ulaza, matricu težina, sabirače, vektor pristrasnosti, blokove transformacionih funkcija, kao i vektorski izlaz. Svaki element vektora ulaza **p** povezan je sa neuronom kroz matricu težina **W**. Svaki neuron prima ulaze pomnožene sa odgovarajućim težinama nakon čega se taj proizvod sabira sa ogovarajućom pristrasnošću b_i i na taj način se dobija argument transfer funkcije f koja će dati izlaz a_i. Zajedno svi izlazi čine vektor izlaza, a u isto vreme predstavljaju ulaze za naredni sloj mreže (Hagan, 2014, str. 2-9). Na slici 7 prikazana je mreža koja sadrži tri sloja.

[6] Preuzeto iz Hagan, 2002, str. 2-7



Slika 7: Višeslojna neuronska mreža[7]

3.4. Samoorganizujuće mape

U ovom delu rada biće detaljnije prikazana specijalna vrsta neuronskih mreža pod nazivom Kohonenove samoorganizujuće mape (eng. Kohonen self-organizing maps). Ime su dobile po finskom akademiku i istraživaču Tuvu Kohonenu (eng. Teuvo Kohonen) koji je još 1982. godine prvi put razvio ovaj tip neuronskih mreža. Ono po čemu se one razlikuju od većine drugih vrsta veštačkih neuronskih mreža je kompetitivno učenje (eng. competitive learning) nasuprot učenju zasnovanom na grešci (eng. error-correction learning) kao što je učenje propagacijom unazad (eng. backpropagation). Samoorganizujuće mape se pre svega koriste za klasterovanje opservacija koristeći nenadgledano učenje, međutim dodavanjem posebnog sloja nadgledanih samoorganizujućih mapa moguće je rešavati probleme klasifikacije.

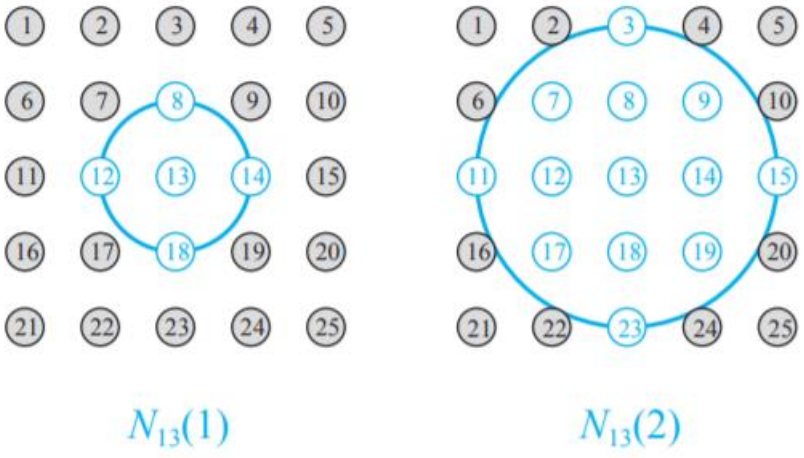
3.4.1. Nenadgledane samoorganizujuće mape

Zadatak ovog tipa neuronske mreže je da produkuje diskretizovanu reprezentaciju malog broja dimenzija (obično dve) ulaznog prostora skupa podataka, koja se naziva mapa. Glavna ideja je korišćenje neurona kako bi se centroidi postavili na samu mapu. Snaga ovog modela ogleda se u njegovoj sposobnosti da „organizuje sam sebe“ na osnovu određenih pravila učenja i interakcija (Tizhoosh, 2019).

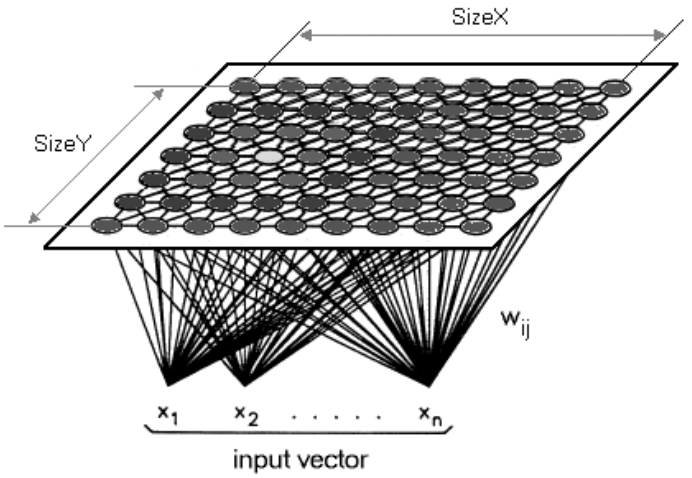
Zamislimo dvodimenzionalnu mapu pravougaonog oblika koja se sastoji od 25 neurona kao što je prikazano na slici 8. Vektor ulaza u neuronsku mrežu biće povezan sa svakim

[7] Preuzeto iz Hagan, 2014, str. 2-11.

neuronom koji će sadržati određene težine, kao što je prethodno objašnjeno, što se može videti na slici 9, dok će sami neuroni sadržati određen broj neurona u njihovom okruženju (eng. Neighborhood) u zavisnosti od toga koliko iznosi radijus (Tizhoosh, 2019). Na slici 8 može se videti okruženje neurona na dvodimenzionalnoj mapi kada radijus iznosi 1, odnosno 2.



Slika 8: Okruženje neurona[8]



Slika 9: Samoorganizujuća mapa sa vektorom ulaza i težinama[9]

Zadatak modela je pronaći vrednosti težina tako da susedni neuroni imaju najsličnije moguće vrednosti, tako što se ulazi dodeljuju neuronima koji su slični njima samima. To znači da će neuron koji će predstavljati ulaz imati težine čije su vrednosti jako slične vrednostima samog ulaza, dok će centri klastera biti upravo neuroni. U ovom slučaju neuroni su ti koji se prilagođavaju ulaznim podacima tako što menjaju svoje vrednosti težina i utiču na neurone u njihovom okruženju da takođe promene vrednosti težina u određenoj meri u zavisnosti od

[8] Preuzeto iz Hagan, 2014, str. 16-13.
[9] Preuzeto sa towardsdatascience

udaljenosti od centralnog neurona. Ukoliko se pogleda mapa sa radijusom 2 na slici 8 i pretpostavi da je neuron 13 centralni neuron koji predstavlja najveći broj ulaza, može se zamisliti trodimenzionalna Gausova površ koja sadrži vrh u nivou neurona 13 i polako se spušta do oboda njegovog okruženja (Tizhoosh, 2019).

Postavlja se pitanje kako pronaći centralni neuron koji će imati najsličnije vrednosti ulazu. To se postiže takmičenjem neurona. Za ulazni vektor **x** cilj je pronaći *i-ti* neuron koji sadrži težine čije su vrednosti najsličnije ulazima. Sličnost u ovom smislu se posmatra kao geometrijska udaljenost, a najčešće se koristi Euklidsko rastojanje. Dakle, zadatak se može definisati kao minimiziranje distance ili maksimiziranje skalarnog proizvoda vektora težina *i-tog* neurona i ulaznog vektora **x**. Za svaki neuron *j* u okruženju *N(i)* pobedničkog neurona *i* ažuriraju se težine neurona *j* (*Wj*). Težine izvan ovog okruženja ostaju nepromenjene (Tizhoosh, 2019).

Mogu se uočiti tri ključne faze modela samoorganizujućih mapa:

1) Takmičenje (eng. competition)
2) Saradnja (eng. collaboration)
3) Ažuriranje težina (eng. weight update)

U fazi takmičenja cilj je pronaći neuron najsličniji ulazu. Matematički to se može predstaviti na sledeći način:

$$i(x) = \arg\max \left\| x - W_j \right\|_2 \; ; j = 1,2, \dots, m \; ; m = broj\ neurona \tag{4}$$

U samom početku težine su nasumične, međutim ideja je da se u nekom trenutku pronađe dobar način ažuriranja težina.

Druga faza, saradnja, obuhvata proces nalaženja neurona koji se nalaze u okruženju pobedničkog neurona i dodeljivanje značaja svakom od neurona okruženja. Kao što je već rečeno, rastojanje se najčešće računa kao Euklidska distanca, dok se značaj svakom od neurona dodeljuje najčešće koristeći Gausovu funkciju.

Treća faza obuhvata ažuriranje težina tako što se u svakoj iteraciji na iznos prethodne težine dodaje određena vrednost, kao što se može videti u sledećoj formuli:

$$W_j(n+1) = W_j(n) + \Delta W_j \tag{5}$$

$$\Delta W_j = \eta Y_j X - g(Y_j) W_j \tag{6}$$

$$g(Y_j) = \eta Y_j \tag{7}$$

Umanjenik jednačine za dobijanje vrednosti ΔW_j odnosi se na Hebovo pravilo (eng. Hebb's rule), dok se umanjilac odnosi na takozvano „pravilo zaboravljanja" (eng. Forgetting rule). Vrednost η uzima vrednosti iz intervala od 0 do 1 u zavisnosti u kojoj meri koliko promena želimo da napravimo. Idealno, η bi trebalo da bude funkcija vremena, jer je praksa da se dosta promena pravi u početku, a malo na kraju. Razlog tome je što promene posle velikog broja iteracija mogu poremetiti ono što je algoritam prethodno naučio (Tizhoosh, 2019).

3.4.2. Nadgledane samoorganizujuće mape

Da bi se samoorganizujuće mape iskoristile za rešavanje problema klasifikacije, potrebno je pridodati drugu samoorganizujuću mapu prvoj nenadgledanoj. Razlika između dve mape se ogleda u dimenziji težina, kao i njihovom treningu. Težine nenadgledane samoorganizujuće mape su iste dimenzije kao ulazni podaci. Nasuprot tome, težine druge, nadgledane samoorganizujuće mape su iste dimenzije kao izlazna promenljiva konkretnog zadatka. Dakle, nenadgledana mapa se koristi za pronalaženje pobedničkog neurona, dok nadgledana povezuje taj neuron sa konkretnom klasom (Riese et al. 2019, str. 8).

Može se reći da nadgledane samoorganizujuće mape koje se mogu koristiti za klasifikaciju ili regresiju predstavljaju svojevrsnu dopunu nenadgledanih mapa, s tim što se sada računa matrica verovatnoće promene klase, a ažuriraju diskretne težine nadgledane mape. Verovatnoća promene klase čvora *i* je jednaka proizvodu težine trenutne klase, stope učenja (eng. learning rate) i težine koja se odnosi na distancu okruženja pobedničkog neurona. Jednostavan prag (eng. threshold) bi doveo do statičkog ponašanja mape posle određenog broja iteracija. Zbog toga se uključuje određena nasumičnost u proces odlučivanja da li će doći do promene. Ta nasumičnost predstavlja vrednost uniformno raspodeljenu između 0 i 1 u svakoj iteraciji. Ažuriranje težina zavisi od toga da li je ova vrednost manja od prethodno izračunate verovatnoće. Ukoliko jeste doći će do promene u klasu koja se odnosi na trenutnu iteraciju, a u suprotnom neće doći do promene. Posle maksimalnog broja iteracija, nadgledana samoorganizujuća mapa je u potpunosti istrenirana, a svakom neuronu je dodeljena jedna klasa (Riese et al. 2019, str. 8).

4. Opis konkretnog problema

Kao što je već pomenuto, projekat na osnovu koga je nastao ovaj rad bazira se na problemu binarne klasifikacije osiguranika jedne evropske osiguravajuće kuće, na osnovu toga da li žele da kupe osiguranje za karavan vozilo.

Skup podataka korišćen u samom eksperimentu obezbedila je holandska kompanija koja se bavi data mining-om, *Sentient Machine Research.* Podaci su podeljeni na trening skup (5822 opservacije) koji sadrži podatke o izlaznoj promenljivoj i test skup (4000 opservacija) koji ne sadrži podatke o izlaznoj promenljivoj.

Svaka opservacija, odnosno red u skupu podataka sačinjena je od 86 atributa (kolona), od koji se prvih 43 odnose na socio-demografske karakteristike pojedinačnih osiguranika, a narednih 43 na korišćenje već postojećih vrsta osiguranja kompanije. Socio-demografske karakteristike su izvedene iz poštanskog broja (eng. zip code) kome pripada dati korisnik, tako da su vrednosti ovih promenljivih identične za one korisnike koji dolaze iz područja sa istim poštanskim brojem. Poslednja kolona predstavlja izlaznu promenljivu čije su vrednosti 0 ili 1, gde je 0 indikator da korisnik ne želi da kupi uslugu karavan osiguranja, a 1 indikator da želi. Kako je cilj kompanije da se što tačnije predvide svi potencijalni kupci karavan osiguranja kako bi se promocija usmerila isključivo na njih i na taj način smanjili troškovi, za pozitivnu klasu izlazne promenljive koristiće se kupci karavan osiguranja, dok će se za negativnu koristiti oni korisnici koji nisu zainteresovani za ovaj tip osiguranja.

Ceo projekat je odrađen u programskom jeziku R (verzija 3.6.2), kao jednom od najadekvatnijih programskih jezika za potrebe računarske inteligencije. Programsko okruženje korišćeno za rad je RStudio (verzija 1.2.5033).

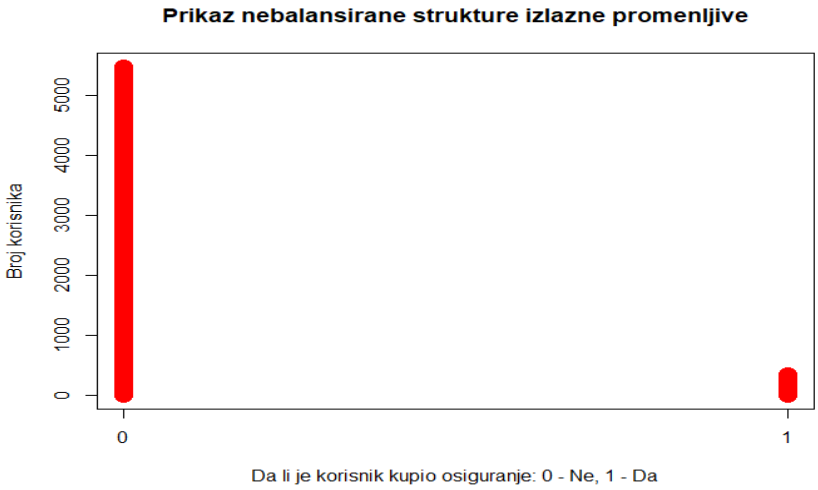
5. Analiza i pretprocesiranje podataka

U narednom delu akcenat će biti na analizi i pretprocesiranju podataka koji su neophodni da bi se originalni skup podataka bolje razumeo, a potom i obradio i da bi se na kraju od njega dobio redukovani na kom će se vršiti sam eksperiment.

5.1. Struktura podataka

S obzirom da je za potrebe ovog projekta korišćen gotov skup podataka, prvi zadatak je bio učitati podatke u samo programsko okruženje. S obzirom da skup podataka sadrži veliki broj atributa, čak 86, prikazivanje rečnika podataka u ovom radu ne bi bilo pregledno, međutim on je dostupan na sledećem linku http://kdd.ics.uci.edu/databases/tic/dictionary.txt. Prva karakteristika podataka koja se može zapaziti je da su svih 86 promenljivih celobrojni tipovi (integer), dok je u rečniku podataka objašnjeno koja skala je korišćena za svaku pojedinačnu promenljivu.

Neizbalansiranost skupa podataka predstavlja čestu situaciju prilikom rešavanja problema mašinskog učenja. Neizbalansiranost skupa podataka predstavlja nejednaku distribuciju klasa pojedinih promenljivih (Medium, 2019). Ovo je čest slučaj u rešavanju problema mašinskog učenja, pogotovu kada se radi o problemima klasifikacije (npr. u medicini kada se ispituje da li osoba ima neku retku bolest gde će po pravilu samo mali procenat osoba stvarno biti zaražen). U ovom projektu ispituje se da li će korisnik nekog od osiguranja biti zainteresovan i za kupovinu karavan osiguranja. Logičkim razmišljanjem može se zaključiti da će mali broj korisnika želeti da kupi ovo osiguranje, pre svega zbog činjenice da mali broj korisnika zapravo i poseduje karavan vozilo. Da je ovo razmišljanje ispravno može se videti na jednostavnom grafiku koji pokazuje da samo 6.4% korisnika u trening setu poseduje osiguranje za karavan vozilo.



Slika 10: Odnos korisnika koji su kupili osiguranje karavan vozila i onih koji nisu

Ukoliko se rešava problem klasifikacije, a izlazna promenljiva je nebalansirana, vrlo lako je moguće napisati "lažne" algoritme koji će svakom novom slučaju dodeliti najčešće pojavljivanje vrednosti izlazne klase i na taj način postići veliki procenat tačnosti (eng. accuracy). Međutim, to ne treba da zavara, jer će se uspešnost klasifikatora ogledati pre svega u drugim metrikama koje računaju udeo pozitivne klase što su senzitivnost (eng. Sensitivity - odnos tačno predviđenih pozitivnih promenljivih i svih varijabli koje su stvarno pozitivne) i preciznost (eng. Percision - odnos tačno predviđenih pozitivnih promenljivih i svih predviđenih pozitivnih promenljivih). Ove metrike daće odličan uvid u to koliko dobro se predviđaju pozitivne opservacije i biće jako korisne u evaluaciji modela, međutim o njima će više reči biti tokom samog eksperimenta.

Međutim, postoje i određene tehnike koje rešavaju probem nebalansiranog skupa podataka. Najpoznatije među njima su "sampling" tehnike i to "undersampling" i "oversampling". Prva predstavlja proces nasumičnog brisanja opservacija iz dominantne klase (u ovom slučaju negativne klase koja se odnosi na korisnike koji nisu zainteresovani za kupovinu karavan osiguranja) radi izjednačavanja broja opservacija obe klase. "Oversampling" predstavlja proces nasumičnog generisanja uzorka iz opservacija nedominantne klase takođe radi izjednačavanja broja opservacija obe klase. Postoji više različitih "oversampling" tehnika. Najpoznatije su "random oversampling" i "SMOTE" (synthetic minority oversampling technique). Prva se odnosi na nasumično biranje opservacija nedominantne klase i njihovo ponavljanje do izjednačavanja broja opservacija obe klase. "SMOTE" bira *k* najbližih suseda za svaku opservaciju nedominantne klase, spaja ih sa datom opservacijom i na tim putanjama generiše nove slučajeve (Medium, 2019).

U ovom radu, svaki model ćemo testirati na tri skupa podataka i to nebalansiranom skupu, "undersampling" skupu i "oversampling" skupu i pokušati da utvrdimo da li i na koji način "sampling" tehnike poboljšavaju korišćene modele. Međutim, pre kreiranja novih skupova podataka uvek je poželjno "osloboditi" se viška atributa, odnosno onih atributa za koje se smatra da ne doprinose samoj klasifikaciji nekom od metoda redukcije atributa. Pre svega toga, nije loše malo bolje se upoznati sa skupom podataka.

5.2. Eksploratorna analiza podataka

Nakon prikupljanja podataka, uobičajeni korak predstavlja upoznavanje sa podacima i sagledavanje šta predstavljaju same promenljive i na koji način mogu uticati na rešavanje postavljenog problema. Poželjno je upoznati se sa njihovim glavnim karakteristikama, što je najčešće slučaj uz pomoć vizuelnih metoda. U ovom delu projekta nije neophodno koristiti statističke modele, već je prioritet razumeti podatke izvan formalnih modela i testova nad postavljenim hipotezama. Stoga, sledeći korak u radu predstavlja eksploratorna analiza podataka. Treba napomenuti da skup podataka korišćen u ovom projektu ne sadrži nedostajuće vrednosti, tako da podatke možemo u startu smatrati za adekvatno pripremljene za analizu.

Eksploratorna analiza podataka predstavlja proces inicijalnog istraživanja podataka radi otkrivanja paterna, uočavanja anomalija, testiranja hipoteza i provere pretpostavki uz pomoć deskriptivne statistike i grafičke reprezentacije (Medium, 2018).

5.2.1. Univarijaciona eksploratorna analiza

Univarijaciona analiza je najjednostavniji oblik eksploratorne analize podataka. "Uni" znači "jedan" što zapravo govori da se ovaj tip analize bavi isključivo pojedinačnim promenljivama, a ne uzrocima i vezama (za razliku od regresije) i njena glavna svrha je da opiše podatke, odnosno da pronađe paterne u njima (Statistics How To, 2019).

Kako 81 od 86 promenljivih iz skupa podataka spada u jednu od tri kategorije po kojima su predstavljene promenljive, u ovom delu ćemo se pozabaviti deskriptivnim statistikama svake od ove tri kategorije.

Prva kategorija obuhvata socio-demografske podatke čiji opseg varira od 0 do 9 gde svaki broj predstavlja određeni procentualni opseg zastupljenosti pojedinačne karakteristike u oblasti sa datim poštanskim brojem, kao što je i prikazano u tabeli 1.

Tabela 1: Značenje vrednosti socio-demografskih atributa

Broj	Procentualni opseg
0	0%
1	1 – 10%
2	11 – 23%
3	24 – 36%
4	37 – 49%
5	50 – 62%
6	63 – 75%
7	76 – 88%
8	89 – 99%
9	100%

U tabeli 2 prikazano je prvih 15 socio-demografskih atributa.

Tabela 2: Deskriptivne statistike socio-demografskih atributa

	mean	std	min	25%	50%	75%	max
Roman.catholic	0.701	1.015	0	0	0	1	7
Protestant	4.638	1.721	0	4	5	6	9
Other.religion	1.050	1.011	0	0	1	2	5
No.religion	3.263	1.606	0	2	3	4	9
Married	6.189	1.896	0	5	6	7	9
Living.together	0.873	0.962	0	0	1	1	7
Other.relation	2.287	1.711	0	1	2	3	9
Singles	1.887	1.779	0	0	2	3	9
Household.without.children	3.237	1.609	0	2	3	4	9
Household.with.children	4.303	1.984	0	3	4	6	9
High.level.education	1.485	1.646	0	0	1	2	9
Medium.level.education	3.307	1.723	0	2	3	4	9
Lower.level.education	4.592	2.280	0	3	5	6	9
High.status	1.899	1.814	0	0	2	3	9
Entrepreneur	0.403	0.787	0	0	0	1	5

Može se primeti da većina socio-demografskih atributa sadrži maksimalne opsege, odnosno sadrži opservacije i sa 0 i 100 procenata, što dosta govori o raznolikosti korisnika iz različitih sredina. Samo tri promenljive odstupaju od ove osobine.

Druga kategorija obuhvata novčano ulaganje svakog pojedinačnog korisnika u svaku pojedinačnu već postojeću vrstu osiguranja. Opseg vrednosti ove grupe atributa takođe varira od 0 do 9 gde svaka vrednost predstavlja određeni opseg novčanog ulaganja u pojedinačno osiguranje.

Tabela 3: Značenje vrednosti atributa za novčano ulaganje u osiguranja

Broj	Novčani opseg
0	0 f
1	1 – 49 f
2	50 – 99 f
3	100 – 199 f
4	200 – 499 f
5	500 – 999 f
6	1000 – 4999 f
7	5000 – 9999 f
8	10000 – 19999 f
9	Preko 20000 f

Tabela 4 prikazuje prvih 10 atributa koji govore o ulaganju novca u različita osiguranja.

Tabela 4: Deskriptivne statistike atributa vezanih za ulaganje u osiguranja

	mean	std	min	25%	50%	75%	max
Cont.private.third.party.insr	0.765	0.957	0	0	0	2	3
Cont.third.party.insr.firms	0.039	0.357	0	0	0	0	6
Cont.third.party.insr.agriculture	0.074	0.508	0	0	0	0	4
Cont.car.pol	2.956	2.921	0	0	5	6	9
Cont.delivery.van.pol	0.055	0.566	0	0	0	0	7
Cont.motorcycle.scooter.pol	0.170	0.889	0	0	0	0	7
Cont.lorry.pol	0.009	0.238	0	0	0	0	9
Cont.trailer.pol	0.019	0.201	0	0	0	0	5
Cont.tractor.pol	0.094	0.604	0	0	0	0	7
Cont.agricultural.machines.pol	0.012	0.215	0	0	0	0	6

Može se zaključiti da za razliku od prve kategorije, mali broj atributa obuhvata maksimalan opseg 0-9. Razlog tome je što samo tri osiguranja imaju bar jednog korisnika koji je uložio preko 20000 novčanih jedinica u to osiguranje i to su, kao što se moglo i pretpostaviti, životno osiguranje, osiguranje automobila i kamiona. Takođe, možemo primetiti da su standardne devijacije ovih atributa dosta manje od socio-demografskih atributa, što i ne treba da čudi jer veliki broj korisnika nije kupovao određena pojedinačna osiguranja a samim tim nije ni ulagao novčana sredstva u njih. Treba napomenuti da od ovog zaključka odstupa osiguranje automobila sa standardnom devijacijom blizu 3, jer ipak veliki broj korisnika kupuje ovaj tip osiguranja, a s obzirom na različite finansijske mogućnosti opseg ulaganja je takođe veliki. Međutim, ono što je možda i najbitnije za zaključiti iz deskriptivne statistike ovih promenljivih je da samo osiguranje automobila i od požara imaju medijanu različitu od nule odakle proističe da ogroman broj korisnika uopšte nije ni kupovao druga osiguranja.

Poslednja kategorija promenljivih odnosi se na broj polisa svakog pojedinačnog osiguranja. Opseg vrednosti ove kategorije atributa varira od 0 do 12 gde svaki broj predstavlja ujedno i broj polisa konkretnog osiguranja.

Tabela 5: Deskriptivne statistike atributa vezanih za broj različitih osiguranja

	mean	std	min	25%	50%	75%	max
Num.of.private.third.party.insr	0.400	0.492	0	0	0	1	2
Num.of.third.party.insr.firms	0.014	0.126	0	0	0	0	5
Num.of.third.party.insr.agricult	0.021	0.144	0	0	0	0	1
Num.of.car.pol	0.557	0.609	0	0	1	1	1
Num.of.delivery.van.pol	0.011	0.130	0	0	0	0	5
Num.of.motorcycle.scooter.pol	0.040	0.224	0	0	0	0	8
Num.of.lorry.pol	0.002	0.068	0	0	0	0	4
Num.of.trailer.pol	0.011	0.116	0	0	0	0	3
Num.of.tractor.pol	0.034	0.250	0	0	0	0	6
Num.of.agricultural.machines.pol	0.005	0.110	0	0	0	0	6

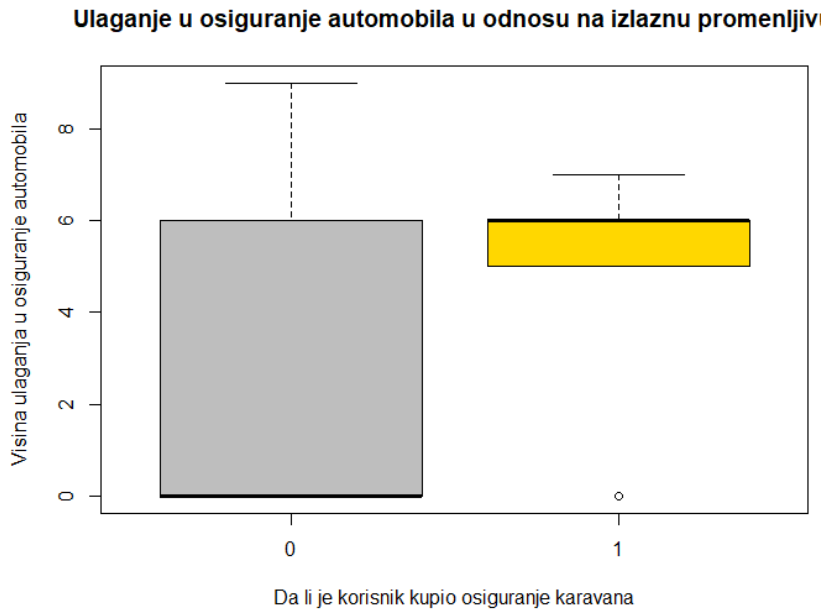
Tabela 4 prikazuje prvih 10 atributa koji govore o broju različitih osiguranja. Što se ove grupe atributa tiče, primetno je da su standardne devijacije ponovo jako niske, ovoga puta sve ispod 1, dok se na prva dva mesta takođe nalaze osiguranje automobila i od požara. Samo osiguranje automobila ima korisnika sa 12 polisa, dok čak 4 tipa osiguranja nemaju ni jednog korisnika sa brojem osiguranja većim od 1. Takođe, kao i kod promenljivih koje se odnose na ulaganja u osiguranja, samo 2 tipa imaju medijanu različitu od nule i to ponovo osiguranje automobila i od požara.

Dakle, lako se uočava da se socio-demografski atributi u mnogome razlikuju od onih koji se odnose na osiguranja. Paterni koji se uočavaju u drugoj i trećoj kategoriji promenljivih dovode do pitanja da li ti atributi uopšte imaju značaja u određivanju vrednosti izlazne promenljive, pre svega činjenica da u obe pomenute kategorije samo dva tipa osiguranja imaju medijanu različitu od nule, a samo tri tipa treći kvartil različit od nule. Međutim, da bi se sa većom dozom sigurnosti moglo reći da većina ovih promenljivih nije značajna za problem klasifikaicije korisno je poslužiti se metodama redukcije atributa što će i biti tema u kasnijem delu rada.

5.2.2. Multivarijaciona eksploratorna analiza

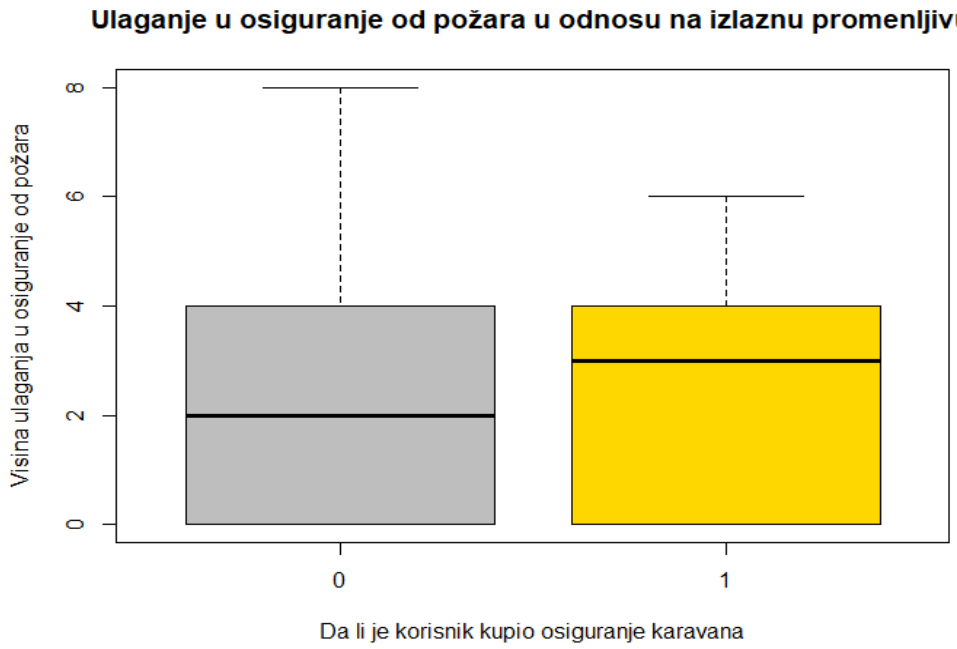
Za razliku od univarijacione analize koja se bavi pojedinačnim promenljivama, multivarijaciona eksploratorna analiza se bavi vezama između promenljivih. Analizirajući pojedinačne promenljive pokazalo se da najveći broj atributa koji se odnose na pojedinačna osiguranja zapravo sadrži ogromne procente polja sa vrednošću nula. Odstupanje od ovog

paterna moglo se primetiti kod osiguranja automobila i od požara, tako da će ovaj deo rada biti posvećen vizuelnom prikazu odnosa ova dva atributa sa izlaznom promenljivom.



Slika 11: Ulaganje u osiguranje automobila u odnosu na izlaznu promenljivu

Sa slike 11 može se primetiti velika razlika ponašanja korisnika u pogledu ulaganja u osiguranje automobila i kupovine osiguranja karavan vozila. Postoji samo jedan izuzetak u vidu korisnika koji nije osigurao automobile, a jeste karavan vozilo. Zapaža se, takođe, velika razlika u medijani u ove dve grupe. Više od pola korisnika koji nisu kupili karavan osiguranje, nisu kupili ni osiguranje automobila, odnosno nisu ulagali u njega, dok polovina korisnika koji su osigurali karavan, uložili su preko hiljadu novčanih jedinica u osiguranje automobila. Ova raznolikost govori u prilog tome da će promenljiva koja se odnosi na ulaganje u osiguranje automobila definitivno imati značaja prilikom klasifikacije.



Slika 12: Ulaganje u osiguranje od požara u odnosu na izlaznu promenljivu

Pogledajmo sada kakav je odnos ulaganja u osiguranje od požara i izlazne promenljive. Može se primetiti da razlike između dve grupe nisu toliko velike kao u prethodnom slučaju. Postoje korisnici koji nisu kupovali ni jednu od ove dve vrste osiguranja, međutim interesantno je videti koliko novca ulažu u osiguranje od požara obe grupe osiguranika. Medijana grupe koja je kupila karavan osiguranje je nešto veća, što nas dovodi do zaključka da oni koji su kupili osiguranje karavan vozila pretenduju da daju nešto veće sume novca za osiguranje od požara, ali interesantno je da su to sume do 5000 novčanih jedinica, dok oni koji se odluče da daju mnogo veće sume novca u osiguranje od požara, u iznosu od 5 do 20 hiljada, ne odlučuju se za kupovinu osiguranja karavan vozila. Sve ovo govori u prilog tome da za razliku od prethodno analiziranog atributa, sada nije moguće sa sigurnošću reći koliko značaja ima ova promenljiva u određivanju izlazne.

5.3. Redukcija atributa

Nakon stečenog uvida u određene karakteristike, odnosno paterne kod promenljivih u skupu podataka, sledeći korak predstavlja redukcija, odnosno smanjivanje broja promenljivih koje će klasifikator koristiti. Pitanje koje se nameće samo po sebi je zašto uopšte redukovati broj promenljivih i na taj način izgubiti određeni deo informacija koje pruža originalni skup podataka. Zapravo postoji više razloga koji su zaslužni za ovaj korak pripreme podataka.

Prvi je mogućnost nastajanja “overfitting-a”, problema vrlo čestog za visokodimenzionalne skupove podataka. Naime, ukoliko je broj atributa veliki, povećava se

mogućnost da model previše dobro opiše trening podatke, ali da izgubi sposobnost generalizacije nad novim uzorcima. Kako je cilj mašinskog učenja što tačnije predviđanje, odnosno klasifikovanje, lako se uočava da prevelika preciznost na trening setu zapravo nije ni blizu objektivan prikaz kvaliteta nekog modela.

Drugi razlog je taj što je želja svakog projektanta modela da on bude jednostavan i razumljiv, što se lako izgubi ukoliko je broj promenljivih veliki.

Sledeći razlog je takozvani "Garbage In Garbage Out" problem. Loš kvalitet ulaza proizvešće loš kvalitet izlaza. Ovo je čest slučaj ukoliko skup podataka sadrži veliki broj takozvanih neinformativnih atributa (Medium, 2019).

Kako skup podataka namenjen za ovaj projekat sadrži čak 86 atributa, od kojih veliki broj ne nosi značajnu informaciju, jasno je da je smanjenje broja promenljivih poželjno da ne bi došlo do problema nastalih zbog gorepomenutih slučajeva.

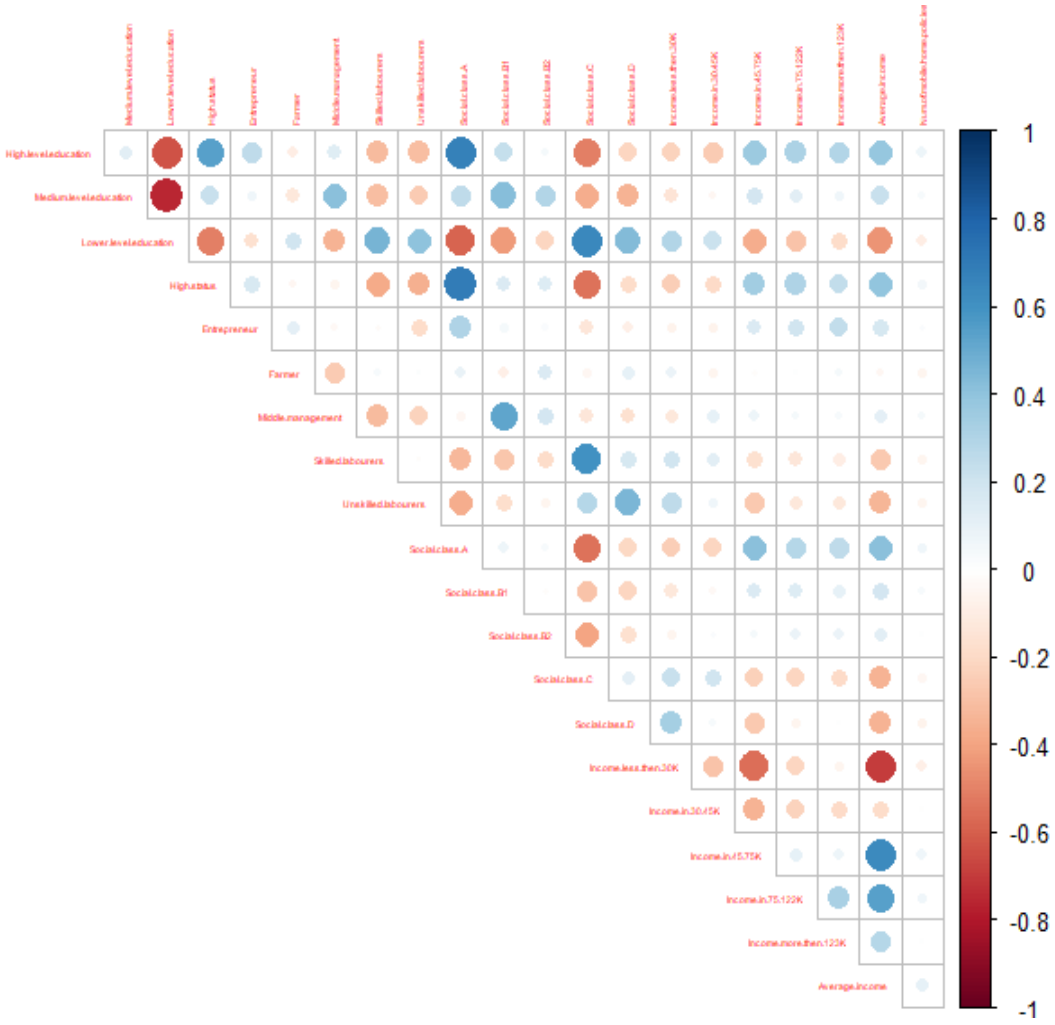
U praksi postoji mnoštvo metoda za redukciju atributa, međutim većina se može podeliti u tri grupe: filter metode, obmotavajuće metode (eng. wrapper-based) i ugrađene metode (eng. embedded). U ovom radu, prvo je korišćena jedna filter metoda, na koju je nadovezana jedna obmotavajuća metoda da bi se dobio konačan skup podataka spreman za upotrebu od strane modela (Medium, 2019).

5.3.1. Filter metode

Ovaj tip metoda zasniva se na rangiranju svakog atributa jednom od univarijacionih metrika i potom biranju atributa sa najvišim rangom. Neke od univarijacionih metrika su varijansa, hi-kvadrat (eng. chi-square), koeficijenti korelacije, kao i informaciona dobit (Medium, 2019).

U ovom radu korišćena je metrika koeficijenta korelacije. Kako se skup podataka sastoji od ordinalnih atributa, a uslovi za korišćenje Pirsonovog koeficijenta korelacije nisu zadovoljeni, odlučeno je da se koristi Spirmanov koeficijent korelacije (Laerd Statistics, n.d.). Ispitana je korelacija između određenih socio-demografskih karakteristika za koje se smatralo da bi mogle biti visoko korelisane, što se ispostavilo kao slučaj sa promenljivima koje se tiču obrazovanja, visine prihoda i pripadnosti određenoj socijalnoj klasi. Nakon ovog ispitivanja odlučeno je da se izbace promenljive koje se tiču pripadnosti socijalnoj klasi jer su one bile u najslabijoj korelaciji sa izlaznom promenljivom u odnosu na ostale pomenute. Iz istog razloga neće se koristiti promenljive koje se odnose na edukaciju. Takođe, iz grupe socio-demografskih

atributa izbacuju se oni koji se odnose na religiju koja nema praktičnog uticaja na to da li će osoba kupiti karavan osiguranje. Takođe, informacije o prihodu su sadržane u promenljivim koje obuhvataju date opsege prihoda, tako da atribut koji govori o prosečnom prihodu može da se izbaci iz skupa podataka.



Slika 13: Tabela korelacije

Kao što je već pokazano u prethodnom delu rada, kategorije promenljivih koje se odnose na ulaganje novca u određene tipove osiguranja i broj kupljenih osiguranja nose malu informaciju pre svega jer su to vrste osiguranja koja se ne susreću toliko često u praksi, pa samim tim većina korisnika ih i ne kupuje, odakle proizilazi da je veliki broj polja u skupu podataka jednak nuli. Malo detaljnijom analizom dolazi se do zaključka da samo tri vrste osiguranja imaju broj polja jednak nuli koji iznosi manje od 90% i kada je u pitanju ulaganje novca i kada je u pitanju broj kupljenih osiguranja. Treba napomenuti da sva tri tipa, a to su osiguranje automobila, osiguranje od požara i osiguranje treće strane imaju značajno više korisnika od svih ostalih osiguranja tako da se za sada ostavljaju u skupu podataka koji će se koristiti za klasifikaciju. Pored njih u ovom skupu ostaće i osiguranje broda s obzirom na visoku korelaciju sa izlaznom promenljivom. Sve ostale vrste osiguranja nose vrlo malu informaciju

tako da će se agregirati jednostavnim sabiranjem i na taj način dobijaju se još dve promenljive, jedna koja se odnosi na ulaganja i druga koja se odnosi na broj kupljenih osiguranja, koje ulaze u skup podataka.

5.3.2. Obmotavajuće metode – Boruta algoritam

U prethodnom delu, pomoću filter metoda redukcije podataka i iskustva, odnosno logičkog zaključivanja, originalan skup podataka smanjen je sa 86 atributa na 41. Taj broj je pogodniji za upotrebu u modelima, jer u određenoj meri umanjuje šanse za nastanak problema u slučaju velikog broja promenljivih koji su opisani u prethodnom delu rada. Međutim, postoji li način na koji se može proveriti da li svi preostali atributi stvarno nose značajnu informaciju, odnosno da li će imati uticaja prilikom klasifikacije? Naravno, moguće je sprovesti još neku metodu redukcije podataka. U ovom delu rada biće opisana jedna veoma korisna tehnika, koja spada u obmotavajuću grupu metoda i to oko algoritma slučajne šume, a naziva se Boruta algoritam.

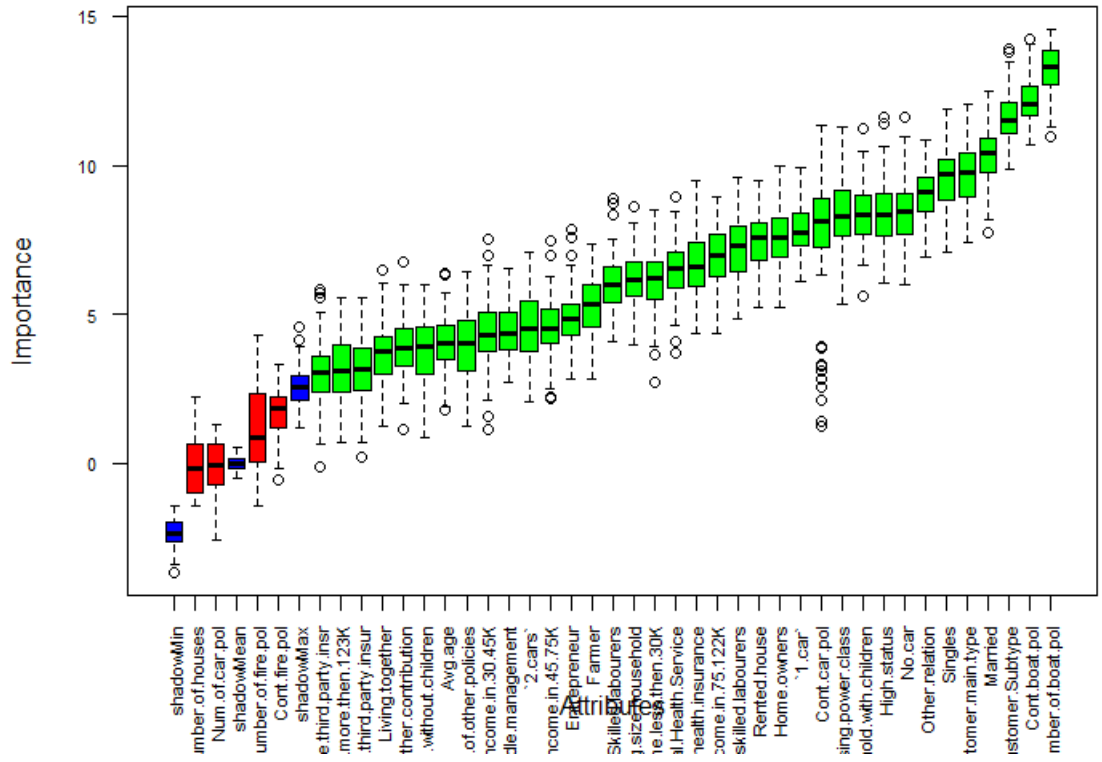
Boruta algoritam sastoji se iz nekoliko koraka:

1. Prvo, dodaje određenu dozu nasumičnosti postojećem skupu podataka tako što kreira kopije svih promenljivih koje se sastoje od nasumično izmešanih vrednosti originalnih promenljivih. Ovi atributi se nazivaju „shadow“ atributi.
2. Nakon toga, boruta trenira klasifikator slučajne šume na proširenom skupu podataka i primenjuje meru značajnosti (podrazumevana je “mean decrease accuracy”) da bi se evaluirao značaj svakog atributa.
3. U svakoj iteraciji, ovaj algoritam proverava da li originalni atribut ima veći značaj od najboljeg od svih „shadow“ atributa (da li originalna promenljiva ima veći Z skor od maksimalnog Z skora među odgovarajućim „shadow“ atributima) i izbacuje atribute koji se pokazuju neznačajnim.
4. Na kraju, algoritam prestaje sa radom ili kada se sve promenljive pokažu značajnim, odnosno neznačajnim, ili kada se dostigne određeni limit krugova slučajne šume.

(Analytics Vidhya, 2019)

Nakon 90 iteracija algoritam je pronašao čak 36 značajnih promenljivih i samo 4 neznačajne što ide u prilog tome da je prvobitna redukcija podataka odrađena na dobar način. Među neznačajnim atributima nalazi se samo jedan koji pripada grupi socio-demografskih karakteristika i to je u pitanju broj kuća. Ostale tri, ulaganje u osiguranje od požara i broj osiguranja od požara, kao i broj osiguranja automobila, iako imaju značajno više pojavljivanja od drugih atributa koji se odnose na vrste osiguranja, takođe su proglašene neznačajnim. Sve

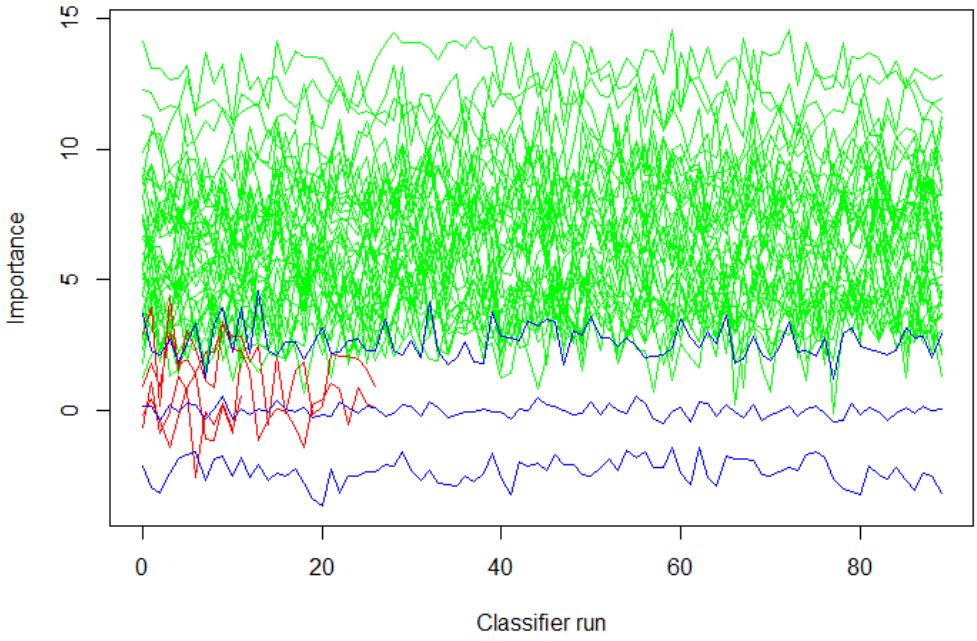
ovo ide u prilog činjenici da će glavnu ulogu u određivanju vrednosti izlazne promenljive imati socio-demografske karakteristike. Na slici 14 su prikazani boks-plotovi za svaki atribut kada je u pitanju njegov značaj za klasifikaciju izlazne promenljive, kao i boks-plotovi minimalnih, srednjih i maksimalnih vrednosti „shadow“ atributa.



Slika 14: Grafička reprezentacija boruta algoritma

Idealno, „shadow“ atributi ne bi trebalo da imaju veliki značaj. Zapravo, značaj ovih atributa treba da bude blizak nuli, ali zbog nasumične fluktuacije može imati vrednost različitu od nule. Dakle, značaj „shadow“ atributa može se koristiti kao referentna vrednost za određivanje koliko je značajan svaki atribut. Na slici su s leva na desno poređani atributi po značaju u rastućem redosledu. Vidimo na početku prvo 4 crvena boks-plota koji označavaju 4 pomenute promenljive koje nisu prošle test značajnosti, dok posle njih slede sve one promenljive koje su prošle test.

Moguće je iscrtati i istoriju značaja atributa. Ako vrednost pada u okviru plavih linija, te promenljive najverovatnije neće proći test značajnosti, dok u zelenu oblast upadaju promenljive koje imaju mnogo veći značaj nego „shadow“ atributi. Istorija značaja atributa prikazana je na slici broj 15.



Slika 15: Grafička reprezentacija kretanja značajnosti atributa kroz iteracije

Na osnovu tabele 6 može se analizirati dodatna statistika koja govori kako se kretao značaj atributa tokom iteracija.

Tabela 6: Značaj atributa tokom iteracija

	mean	median	min	max	normHits	decision
Cont.private.third.party.insr	3.015	3.054	-0.113	5.821	0.711	Confirmed
Cont.car.pol	7.672	8.136	1.248	11.362	0.944	Confirmed
Cont.fire.pol	1.667	1.867	-0.522	3.308	0.044	Rejected
Cont.boat.pol	12.129	12.058	10.660	14.203	1.000	Confirmed
Num.of.private.third.party.insr	3.171	3.165	0.195	5.541	0.711	Confirmed
Num.of.car.pol	-0.132	-0.052	-2.580	1.296	0.000	Rejected
Num.of.fire.pol	1.089	0.845	-1.433	4.298	0.044	Rejected
Num.of.boat.pol	13.164	13.294	10.947	14.575	1.000	Confirmed
Other.contribution	3.929	3.888	1.166	6.733	0.878	Confirmed
Num.of.other.pol	4.047	4.035	1.230	6.425	0.889	Confirmed

Data statistika pokazuje koliko su iznosile minimalna, maksimalna, srednja vrednost kao i medijana vrednosti značaja svakog atributa. Pored toga, “normHits” govori koji procenat vremena se svaka promenljiva pokazala značajnijom nego „shadow“ atribut. Vidimo da je ovaj procenat blizak ili jednak nuli za atribute koji nisu prošli test, dok je za ostale daleko iznad 50%, a za mnoge je i jednak 100%, što govori da bi ti atributi trebalo da imaju veliku ulogu u klasifikatoru.

Nakon redukcije atributa, originalni skup podataka sveden je na završni skup koji se sastoji od 36 promenljivih za koje se smatra da će imati najveći značaj prilikom klasifikovanja izlazne promenljive, tako da su se stekli svi preduslovi za izvođenje eksperimenta.

6. Eksperiment

Kako je redukcija atributa završena i dobijen završni skup podataka, stekli su se svi preduslovi za obavljanje samog eksperimenta. Odlučeno je da model za rešavanje problema na koji se stavlja akcenat budu Kohonenove samoorganizujuće mape. Način na koji one funkcionišu objašnjen je u drugom poglavlju, a sada će biti prikazana njihova implementacija u R programskom jeziku. Kako se rešavanje problema klasifikacije korišćenjem ovog tipa neuronskih mreža nadovezuje na nenadgledane samoorganizujuće mape koje se koriste za klasterovanje opservacija, u prvom delu eksperimenta će biti upravo izvršeno klasterovanje. Radi bolje vizualizacije i lakšeg interpretiranja rezultata, klasterovanje opservacija će se obaviti korišćenjem samo atributa koji se odnose na prihode korisnika po opštinama, dok će klasifikator, naravno, sadržati sve promenljive iz završnog skupa podataka.

6.1. Klasterovanje korišćenjem samoorganizujućih mapa

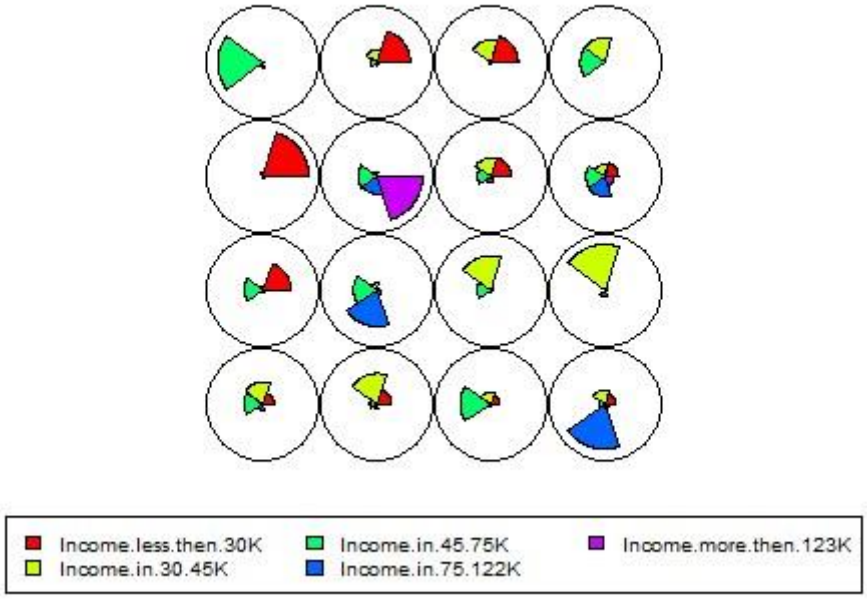
Samoorganizujuće mape predstavljaju vrstu veštačkih neuronskih mreža koja koristi tehniku nenadgledanog učenja da bi bilo moguće vizuelizovati visokodimenzionalne skupove podataka u niskodimenzionalne, najčešće dvodimenzionalne reprezentacije. Takođe, ovaj tip neuronskih mreža se koristi za klasterovanje, tako što grupiše slične opservacije (Rpubs, n.d.).

Klasterovanje predstavlja podelu svih opservacija u određen broj grupa tako da su opservacije u istoj grupi mnogo sličnije nego onima u drugim grupama (Analytics Vidhya, 2019). Čemu služi ovakva podela, biće objašnjeno upravo na primeru koji je obrađen u eksperimentu. Pretpostavimo da vlasnici osiguravajuće kuće žele da bolje razumeju preferencije svojih korisnika, da bi mogli da unaprede poslovanje. Da li je moguće pogledati detalje svakog pojedinačnog korisnika i razviti jedinstvenu biznis strategiju za svakog od njih? Teoretski, verovatno da, ali u praksi se ovo nikad ne radi. Mnogo bolje rešenje bi bilo klasterovati sve korisnike u određen broj grupa u zavisnosti od nekih njihovih karakteristika i razviti strategije za svaku grupu pojedinačno. Karakteristike koje su uzete u obzir u ovom radu su zapravo podaci o prihodima korisnika po opštinama u kojima žive, jer su upravo prihodi jedan od najvažnijih faktora koji može uticati na korisnike da li će kupiti određenu vrstu osiguranja, dok kompanija na osnovu njih, cene različitih polisa osiguranja i uslova njihovog korišćenja može razviti različite strategije gde bi na svakoj grupi bila upotrebljena najefektivnija strategija za nju.

Funkcije pomoću kojih je moguće kreirati model samoorganizujućih mapa u R programskom jeziku je moguće koristiti uz paket **kohonen**. Kako su opsezi svih pet atributa

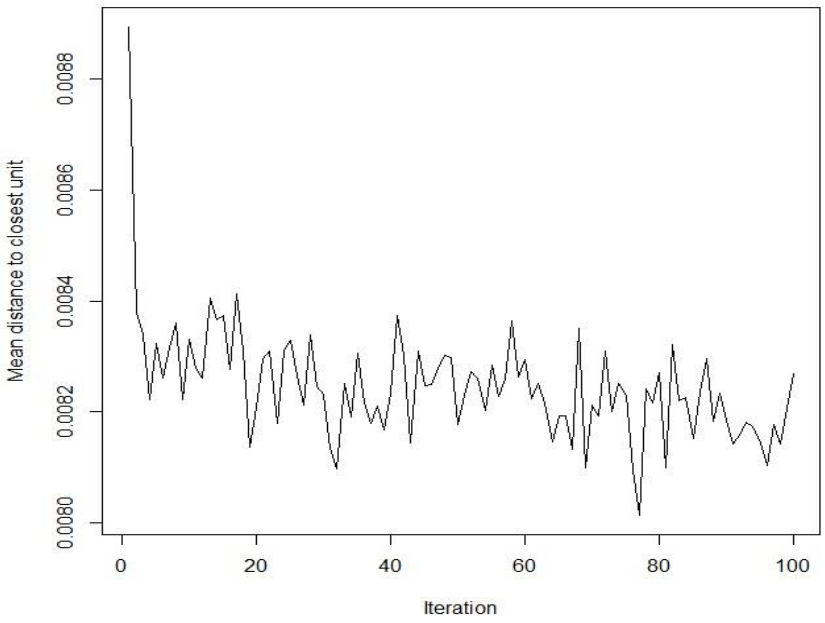
koje se odnose na prihod jednaki, nije potrebno normalizovati ili standardizovati podatke. Prvi korak predstavlja pravljenje matrice sa vrednostima svih opservacija za ovih pet promenljivih, koja ujedno predstavlja i prvi argument funkcije ***som*** koja se koristi za kreiranje modela. Drugi argument se odnosi na samu mapu (eng. grid). Mapa se dobija korišćenjem funkcije ***somgrid.*** Ona zapravo čuva koordinate mape koja će se koristiti. Potrebno je izabrati veličinu obe dimenzije i oblik mape između pravougaonog (eng. rectangular) i šestougaonog (eng. hexagonal). Upravo je ova dva oblika Kohonen predložio zbog efikasnosti implementacije (Hagan, 2014, str. 16-13). U ovom radu izabran je pravougaoni oblik mape sa veličinom obe dimenzije od četiri. Upravo iz tog razlika na kraju će biti dobijeno 16 čvorova, odnosno neurona, nad kojima će se izvršiti klasterovanje. Treći parametar je stopa učenja (eng. learning rate) odnosno vektor koji sadrži dva broja koji se odnose na veličinu promene težina (upotrebljena je podrazumevana stopa učenja koja predtsavlja linearni pad od 0,05 do 0,01 u okviru podrazumevanih 100 iteracija). Četrvrti i poslednji parametar koji je korišćen za treniranje modela je radijus okruženja pobedničkog neurona koji je postavljen na jedinicu (R documentation, n.d.).

Vizuelizacija svih klastera prikazana je na slici 16 na kojoj se može videti svih 16 neurona i koliki udeo ima svaka od 5 promenljivih na bilo kom neuronu. Brojanje čvorova kreće s leva na desno odozdo na gore. Može se primetiti da se osmi, deveti i trinaesti čvor izdvajaju po tome što obuhvataju opštine koje imaju gotovo stoprocentan udeo korisnika u prihodu iz samo jednog opsega. U devetom neuronu se nalaze oni korisnici iz opština sa najnižim prihodom (ispod 30 hiljada), u osmom oni sa malo višim prihodom (od 30 do 45 hiljada), dok se u trinaestom nalaze korisnici iz opština sa prihodom od 45 do 75 hiljada. Na osnovu svega navedenog ne iznenađuje što se peti korisnik iz trening skupa podataka nalazi u trinaestom čvoru (ima vrednost 9 za atribut *Income.in.45.75K*, dok su ostalo nule), dok se na primer dvadeset deveti korisnik nalazi u devetom čvoru (ima vrednost 9 za atribut *Income.less.than.30K*, dok su ostalo nule). Ostali klasteri imaju nešto raznovrsniju strukturu korisnika, mada se za svaki od njih može izvući odgovarajući patern i shodno tome napraviti adekvatna biznis strategija.



Slika 16: Udeo atributa u svakom klasteru

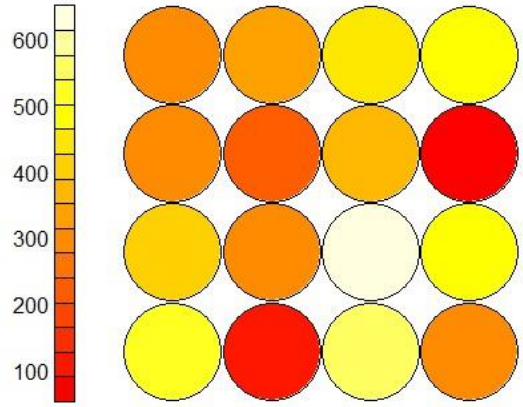
Prilikom odvijanja trening procesa samoorganizujuće mape distanca između težina svakog neurona predstavljenih tim neuronom se smanjuje i u idealnom slučaju distanca bi trebalo da dostigne minimum (Rpubs, n.d.). Na slici 17 može se videti ovaj progres tokom vremena. Ako se kriva konstantno smanjuje potrebno je povećati broj iteracija. U ovom modelu broj iteracija nije morao da bude ovoliko veliki jer vidimo da distanca naglo opada samo prvih nekoliko iteracija, tako da bi se slični rezultati verovatno dobili i sa znatno manjim brojem iteracija.



Slika 17: Trening proces modela samoorganizujuće mape

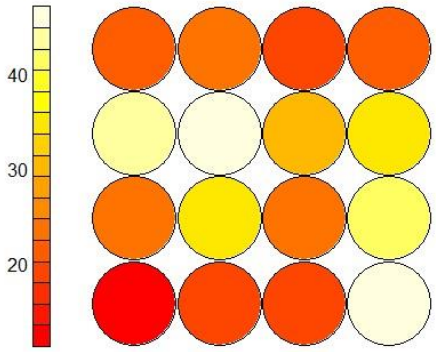
Kohonen paket omogućava vizuelizaciju broja opservacija koje su pridodate svakom čvoru, odnosno neuronu. Ova metrika se može koristiti kao mera kvaliteta mape. Idealno, distribucija opservacija je relativno uniformna. Velike vrednosti u određenim delovima mape sugerišu da bi veća mapa možda bila korisna, dok prazni čvorovi sugerišu da je veličina mape prevelika za dati broj opservacija (Rpubs, n.d.).

Na slici 18 vizuelizovan je broj opservacija po pojedinačnim neuronima. Može se primetiti da broj opservacija po neuronu varira od 100 do 700, tako da raspodela nije uniformna kao što je bila želja, mada s obzirom na konkretan domen problema smatrano je da bi preko 16 neurona dovelo do većeg broja klastera što možda i ne bi bilo korisno jer bi ponovo zahtevalo suviše specifične biznis strategije.



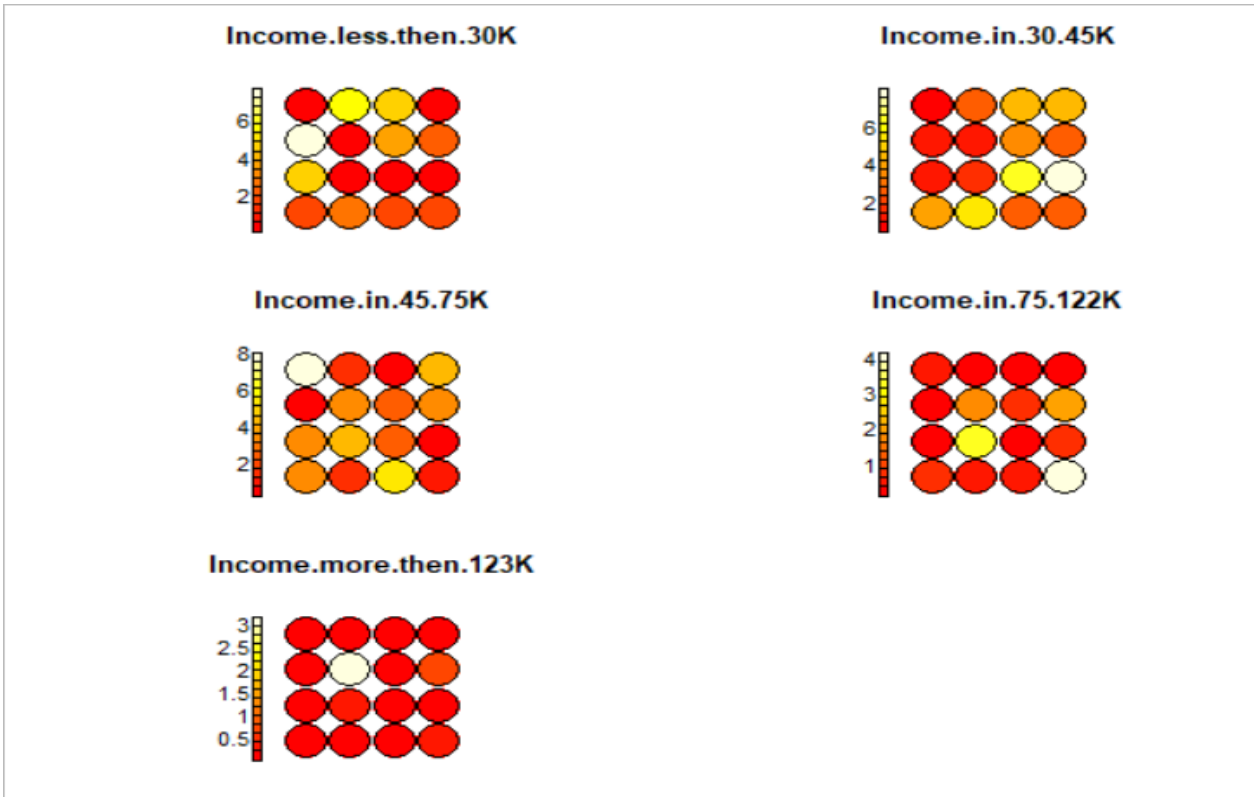
Slika 18: Broj opservacija pridodat svakom neuronu

Još jedna korisna vizuelizacija omogućena ovim paketom, prikazana na slici 19 pokazuje koji neuroni su bliži ili dalji od vektora ulaza. Tamnije boje govore da su ti neuroni dosta bliži ulaznom vekotru, dok što boje postaju svetlije udaljenost između neurona i ulaznih vektora postaje veća.



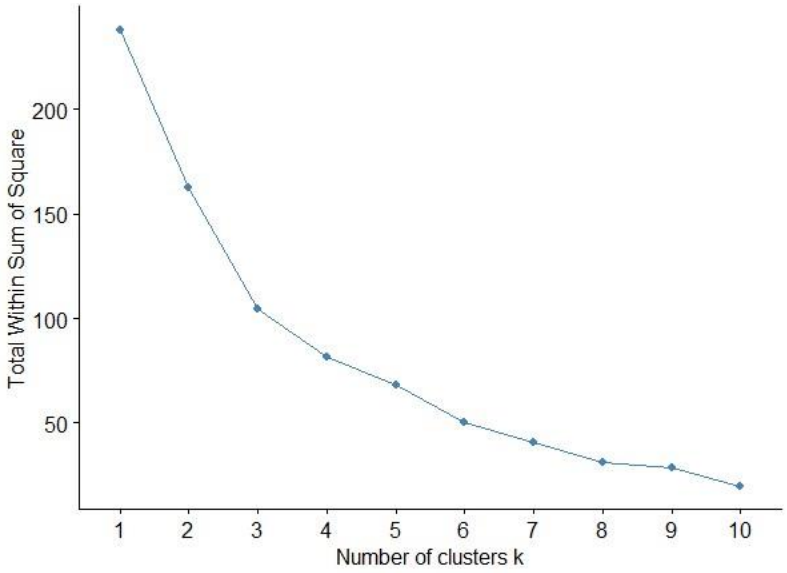
Slika 19: Udaljenost neurona od ulaznog vektora

Hitmape (eng. heatmaps) su verovatno najvažnija moguća vizuelizacija kada se govori o samoorganizujućim mapama. One omogućavaju vizuelizaciju svake pojedinačne promenljive kroz mapu. Na slici 20 je moguće videti višestruke hitmape i potom ih uporediti radi identifikacije značajnih oblasti na mapi (Rpubs, n.d.). Interesantno je zapaziti da je najsvetliji čvor na prvoj hitmapi deveti, na drugoj osmi, a na trećoj trinaesti što se podudara sa pređašnjim zapažanjem sa slike 16 gde je bio analiziran udeo atributa u svakom klasteru.



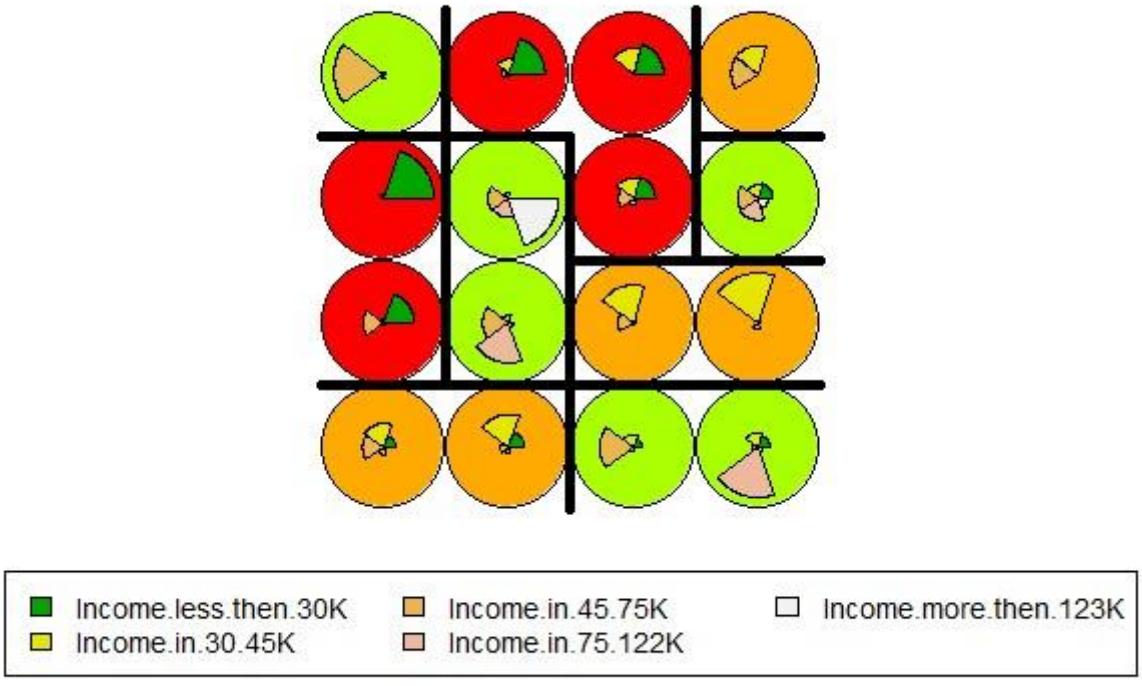
Slika 20: Hitmape

Nakon detaljne vizualizacije korišćenjem **kohonen** paketa, sada je moguće izvršiti klasterovanje čvorova na mapi koja je dobijena, da bi se izolovale grupe sa sličnim metrikama. Pronalaženje broja klastera koji bi bio pogodan može se izvršiti upotrebom **K-means** algoritma i metode lakta (eng. Elbow method). Metoda lakta je dosta sistematičniji pristup određivanju broja klastera umesto nasumičnog nagađanja. Bazirana je na sumi kvadratnih razlika između svake ulazne tačke, odnosno opservacije i centra klastera, odnosno ukupnom kvadratnom rastojanju (eng. total within sum of squares). Broj klastera se određuje tako što se bira onaj broj gde je najveći pad u ukupnom kvadratnom rastojanju između svake dve uzastopne vrednosti *k*. Taj broj će zapravo odgovarati najmanjem uglu na grafiku koji je prikazan na slici 21 (zbog toga je i dobila naziv metoda lakta).



Slika 21: Metoda lakta

Za iscrtavanje grafika je korišćen paket **factoextra** i njegova funkcija **fviz_nbclust**. Deluje da je najmanji ugao za k koje iznosi tri, tako da će u nastavku biti korišćena tri klastera. Korišćenjem **K-means** algoritma izvršeno je kalsterovanje mape, kao što je i vizuelizovano na slici 22.



Slika 22: Klasterovana mapa

Dobijena klaster mapa omogućava profilisanje dobijenih klastera, tako da nije potrebno posmatrati deskriptivne rezultate svakog klastera ili tražiti dublji smisao u podacima (Rpubs, n.d.). Lako se mogu uočiti sledeće karakteristike svakog klastera:

1) Klaster 1 (crveni): obuhvata pretežno opštine u kojima žive korisnici donjeg sloja koji imaju najniža primanja (ispod 30 hiljada), kao i određeni broj korisnika koji imaju veća primanja (uglavnom od 30 do 45 hiljada).
2) Klaster 2 (narandžasti): obuhvata pretežno opštine u kojima žive korisnici sa visinom prihoda između 30 i 45 hiljada, mada se mogu naći i korisnici u okolnim nivoima. Ove korisnike smatrati srednjim slojem.
3) Klaster 3 (svetlo zeleni): obuhvata opštine u kojima žive korisnici sa visinom prihoda u tri najviše kategorije. Ove korisnike možemo smatrati višim slojem.

6.2. Klasifikacija korišćenjem samoorganizujućih mapa

Pre prelaska na prikaz rezultata samog modela za rešavanja problema klasifikacije ukratko će biti predstavljene evaluacione metrike koje su korišćene prilikom provere performansi modela. Svaka od korišćenih evaluacionih metrika se računa na osnovu matrice konfuzije. Kod problema binarne klasifikacije kao što je ovaj ona predstavlja tabelu sa dva reda i dve kolone, gde se u redovima nalaze predviđene a u kolonama stvarne pozitivne i negativne klase.

Tabela 7: Matrica konfuzije

Predviđeno/Stvarno	Pozitivna klasa	Negativna klasa
Pozitivna klasa	Tačno pozitivno (*TP*)	Netačno pozitivno (*FP*)
Negativna klasa	Netačno negativno (*FN*)	Tačno negativno (*TN*)

Radi evaluacije performansi modela korišćene su sledeće:

1) Preciznost (eng. percision, positive predicted values – PPV) – procenat ispravno klasifikovanih instanci (Davis & Goadrich, 2006, str. 235). Računa se prema sledećoj formuli:

$$Preciznost = \frac{TP}{TP + FP} \tag{8}$$

2) Odziv ili senzitivnost (eng. recall or sensitivity) - procenat ispravno klasifikovanih pozitivnih instanci od ukupnog broja pozitivno klasifikovanih primera (Davis & Goadrich, 2006, str. 236). Računa se prema sledećoj formuli:

$$Odziv = \frac{TP}{TP + FN} \tag{9}$$

3) F1 mera (eng. F1 measure) – predstavlja meru koja „balansira“ između preciznosti i odziva. Potreba za ovim balansom uslovljena je činjenicom da su ove dve metrike antagonističkog karaktera. Poboljšavanje jedne nužno utiče na pogoršavanje druge. Računa se prema sledećoj formuli:

$$F1 = 2 * \frac{Preciznost * Odziv}{Preciznost + Odziv} \tag{10}$$

Jako je bitno razumeti zašto su izabrane baš ove metrike za evaluaciju performansi modela. Kao što je rečeno u uvodu, cilj osiguravajuće kuće je sprovođenje strategije unakrsne prodaje, odnosno podsticanje postojećih kupaca da kupe neku novu polisu osiguranja. Kako ova marketing strategija nosi sa sobom potrošnju novčanih i materijalnih resursa, želja je maksimiziranje odnosa profit minus troškovi. Zbog ove činjenice, ne može se svaka greška u klasifikaciji posmatrati podjednako lošom. Greška tipa 2 (eng. type 2 error – false negative) se odnosi na sve one korisnike za koje je pogrešno predviđeno da ne žele da kupe osiguranje karavan vozila. Ovo je najopasnija greška u našem problemu, jer kompanija nikako ne bi sebi smela da dopusti luksuz da „ispusti“ potencijalne kupce, tako što im neće predstaviti novi tip osiguranja. Greška tipa 1 (eng. type 1 error- false positive) se odnosi na sve one korisnike za koje je pogrešno predviđeno da žele da kupe osiguranje. Svakako da i ova loša procena može osiguravajuću kompaniju koštati dosta uzaludno potrošenih resursa i da je želja da se izbegne, ali u ovom slučaju razmišljanje je da je ona manje loša od prethodne, odnosno da su nepotrebni troškovi manji od potencijalnog profita koji nije dobijen.

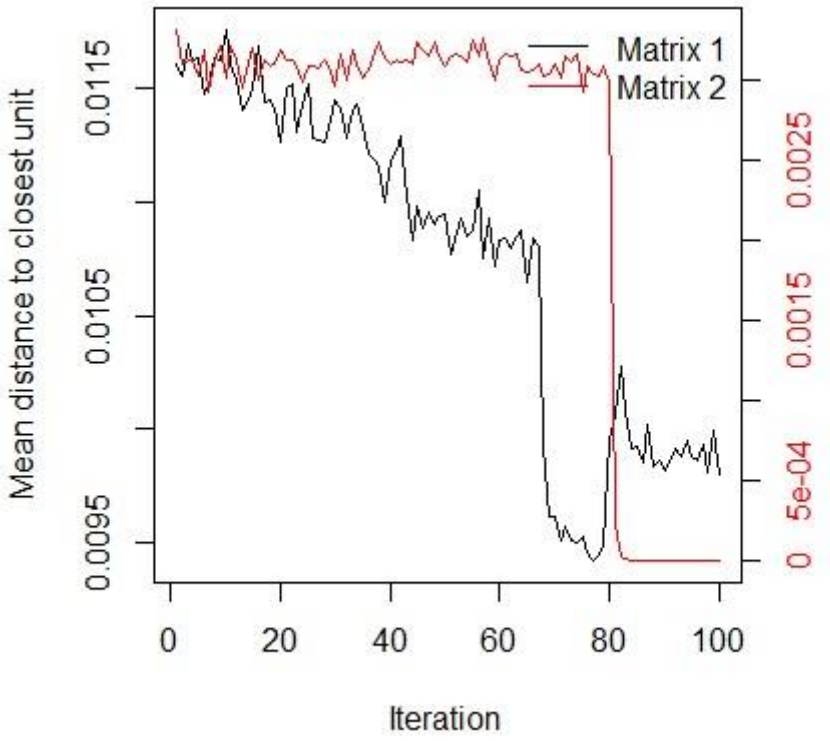
Zbog svega navedenog odlučeno je da se za najbolji model smatra onaj sa najvećim odzivom, mada će se obratiti pažnja i na preciznost i F1 meru koja obuhvate ove obe mere. Takođe, vredi napomenuti da se pri evaluaciji performansi posmatrala još jedne metrika – tačnost (eng. accuracy) koja predstavlja procenat ispravno klasifikovanih opservacija, međutim ona nije bila odlučujuća jer ne pravi razliku između dva tipa greške. Metrika koja se odnosi na tačno predviđene negativne vrednosti nije se posebno posmatrala jer se sa pravom očekivalo da će u svakom modelu ona iznositi veoma visok procenat, jer se očekuje da će se retko dešavati da algoritam proceni da korisnik ne želi da kupi polisu a da to nije slučaj. Takođe, treba imati u vidu da će se često dešavati da algoritam proceni da korisnik želi da kupi polisu a da to nije slučaj, tako da se očekuje izrazito male vrednosti metrike koja se odnosi na preciznost.

U poglavlju 3 objašnjeno je na koji način se samoorganizujuće mape mogu koristiti za klasifikaciju podataka. U pitanju je nadgledana verzija samoorganizujućih mapa za mapiranje visokodimenzionalnih paterna u dvodimenzionalnu mapu. U R programskom jeziku moguće ih je koristiti pomoću **xyf** algoritma (X-Y fused SOMs). Jedan vektor je kreiran za svaki ulazni objekat tako što se povežu ulazne i izlazna promenljiva. Samoorganizujuća mapa je trenirana na uobičajen način, ali sa jednim izuzetkom: distanca između ulaza i čvora je jednaka sumi odvojenih distanci ulaznog i izlaznog prostora (R documentation, n.d.).

Kao što je već rečeno, neizbalansiranost originalnog skupa podataka može ponekad uticati na model da sporije uči. Zbog toga će biti odrađena klasifikacija korišćenjem i "undersampled" i "oversampled" skupa podataka, pored neizbalansiranog i upoređeni rezultati da bi se pronašao najbolji model samoorganizujuće mape. Prilikom kreiranja modela biće vršena optimizacija dimenzija, odnosno oblika mape, kao i broja iteracija.

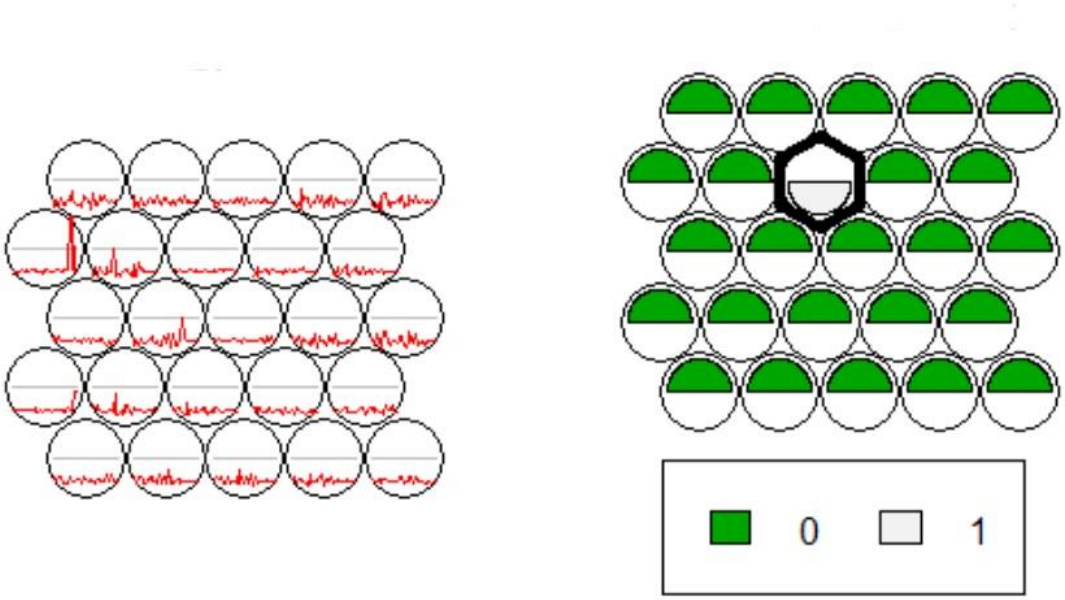
Što se tiče neizbalansiranog (nepromenjenog) skupa podataka, treniranjem modela sa različitim vrednostima parametara, utvrđeno je da se najbolji klasifikator dobija ukoliko se koriste šestougaona mapa dimenzija 5 puta 5 i 100 iteracija. Razlog tome je što je najvažnija metrika – odziv iznosio 0,361, dok je takođe i F1 mera bila veća od svih ostalih modela (0,102). Možda na prvi pogled ove vrednosti deluju nisko, ali treba imati u vidu da su ovo realni podaci i da je pitanje da li je moguće metodama računarske inteligencije istrenirati neki model koji će postizati značajno bolje rezultate. Svakako, korisno bi bilo rešiti problem korišćenjem i nekog drugog klasifikatora kako bi se dobio bolji uvid u to koliko je model samoorganizujućih mapa adekvatan za rešavanje ovog konkretnog problema, što će i biti slučaj u kasnijem delu rada.

Na slici 23 može se videti kako se kretala srednja distanca nezavisnih i zavisne opservacije do najbližeg neurona. Može se primetiti da se ova distanca na početku malo smanjivala, da bi veliki pad došao tek posle velikog broja iteracija.



Slika 23: Promene srednje distance do najbližeg neurona

Poslednja vizuelizacija koja će biti prikazana odnosi se na uporedni prikaz nenadgledane i nadgledane mape. Što se tiče nenadgledane mape ovaj put nije moguće samo na osnovu nje zaključiti koji čvor sadrži koliki procenat atributa iz razloga što su sada korišćeni svi atributi a ne samo njih pet, tako da u R-u nije bilo moguće vizuelizovati kao što je bio slučaj sa manjim brojem atributa. Na nadgledanoj mapi se vidi da se samo jedan čvor odnosi na korisnike koji žele da kupe polisu osiguranja karavan vozila, što i nije iznenađujuće s obzirom na neizbalansiranu strukturu skupa podataka.

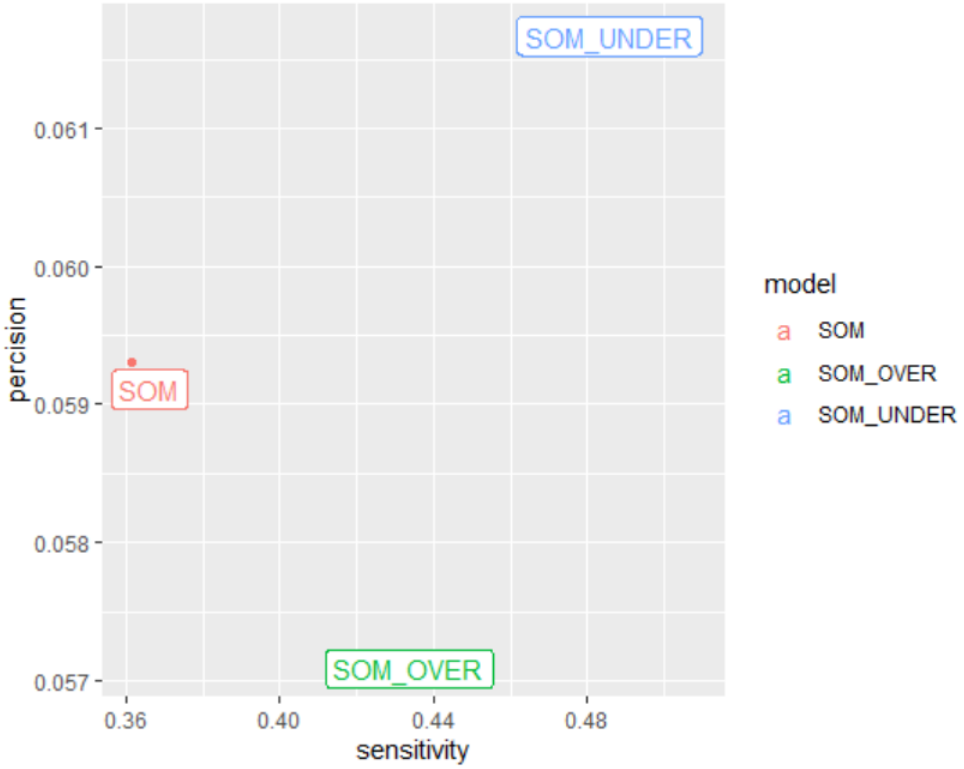


Slika 24: Nenadgledana i nadgledana mapa

Treniranjem modela na „undersampeled“ i „oversampeled“ skupovima podataka dobijeni su nešto drugačiji rezultati. Nakon što je model istreniran na „undersampeled“ skupu dobijeni rezultati pokazuju da je odziv bolji za gotovo petnaest procenata. Preciznost se nije značajno promenila međutim da je ovaj broj opservacija nedovoljan za učenje samoorganizujućih mapa pokazuju niske vrednosti tačnosti koja je tek za jedan procenat iznad 50% kod modela sa optimizovanim parametrima. Treba naglasiti da je najbolje rezultate postigla mapa pravougaonog oblika dimenzija 3 puta 3 što i ne treba da čudi jer je logično da manji broj čvorova bude potreban ovom skupu podataka jer sadrži čak 17 puta manje ulaznih podataka, dok je podrazumevani broj iteracija od 100 i ovaj put bio sasvim dovoljan.

Što se tiče „oversampeled“ skupa podataka odziv se nalazi tačno između prethodna dva skupa, preciznost je i dalje neznačajno promenjena, dok je tačnost ponovo niska, tek par procenata iznad 50%. Najbolja mapa je kao kod neizbalansiranog skupa dimenzija 5 puta 5 pravougaonog oblika.

Uporedi prikaz evaluacionih metrika odziv i preciznost predstavljen je na slici 25.



Slika 25: Preciznost i senzitivnost samoorganizujućih mapa

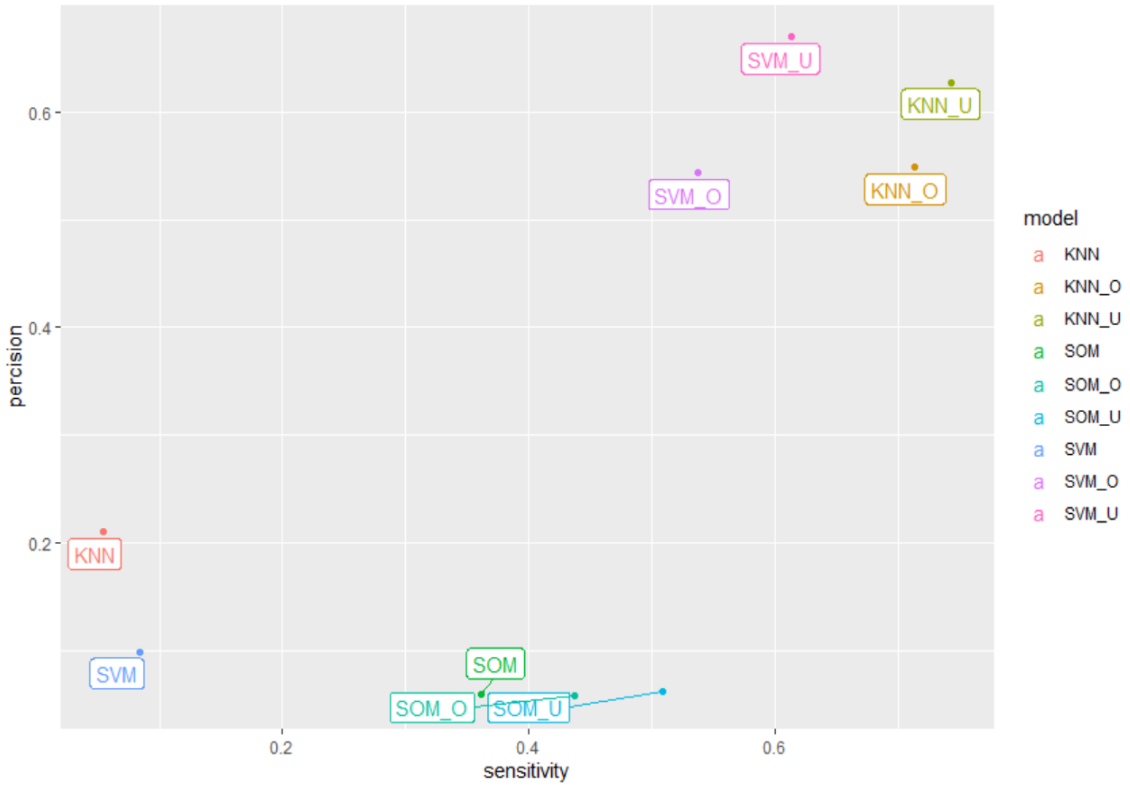
6.3. Poređenje rezultata samoorganizujućih mapa sa rezultatima metode potpornih vektora i k najbližih suseda

Kako bi se stekao bolji uvid u to koliko je model samoorganizujućih mapa odgovarajući za rešavanje ovog problema, ali i utvrdili na koji način se korigovanje nebalnisranosti skupa podataka odražava na treniranje različitih modela, odlučeno je da se problem klasifikacije reši i korišćenjem modela potpornih vektora i *k* najbližih suseda. Nakon klasifikovanja pomoću ova dva modela dobijeno je ukupno 9 rezultata[10] koji se mogu videti u tabeli 8, dok se je na slici 26 grafički prikazan odnos preciznosti i senzitivnosti svih 9 kombinacija.

Tabela 8: Evaluacione metrike za svih 9 modela

	Tačnost	Preciznost	Odziv	NPV	F1
„SOM“	0.621	0.059	0.361	0.940	0.101
„Undersampled SOM“	0.510	0.06	0.508	0.942	0.11
„Oversampled SOM“	0.537	0.057	0.436	0.938	0.101
„SVM“	0.9	0.098	0.084	0.942	0.090
„Undersampled SVM“	0.655	0.669	0.613	0.643	0.640
„Oversampled SVM“	0.555	0.543	0.537	0.565	0.540
„KNN“	0.931	0.209	0.054	0.942	0.086
„Undersampled KNN“	0.651	0.627	0.743	0.685	0.680
„Oversampled KNN“	0.574	0.548	0.713	0.619	0.620

[10] Tri modela puta tri različita skupa podataka



Slika 26: Preciznost i senzitivnost svih modela

Kao što se može primetiti na prvi pogled, raznolikost između rezultata je velika, a klasifikovanje pomoću nova dva modela je definitivno imala smisla, kao i „undersample-ovanje“ i „oversample-ovanje“ skupa podataka. Ove dve tehnike rešavanja problema neizbalansiranosti izlaznog atributa su najmanje uticala na samoorganizujuće mape, međutim i taj uticaj definitivno nije zanemarljiv, pogotovu što je doveo do određenog poboljšanja senzitivnosti kao najvažnije metrike u ovom radu. Međutim, taj uticaj je mnogo izraženiji kod modela poptpornih vektora i k najbližih suseda, gde primećujemo jako sličan patern.

Ova dva modela u praksi su odlično oslikali problem neizbalansirane strukture podataka koji je teoretski obrađen u petom poglavlju. Na neizbalansiranom skupu podataka, oba modela su ostvarila jako dobru tačnost, bolju od bilo kog drugog modela, međutim ona nije validan prikaz snage ova dva modela na datom skupu podataka. Pristrasnost u trening setu je uticala na ova dva algoritma da u mnogome ignorišu manjinsku klasu. To zapravo znači da su istrenirani na taj način da su veliku većinu test opservacija klasifikovali u većinsku klasu i na taj način ostvarili veliku tačnost jer je struktura test seta takođe neizbalansirana. U nekim slučajevima je tačnost bila najveća čak kada apsolutno svi test korisnici klasifikovani kao oni koji ne žele da kupe polisu osiguranja karavan vozila, međutim ovakvi slučajevi su zanemareni jer ne pomažu u cilju da predvidimo ko bi potencijalno bio zainteresovan za ovu polisu.

Zbog svega navedenog, ova neizbalnsirana struktura je promenjena i modeli su trenirani na novim skupovima podataka i dobijeni su mnogo bolji rezultati za dve ključne metrike: senzitivnost i preciznost (iako je u ovim slučajevima tačnost bila značajno manja). Može se primetiti da su „undersampled“ skupovi podataka dali nešto bolje rezultate. Razlog ovome može biti to što su manje skloni „overfitting-u“ zbog manjeg broja trening opservacija.

Zaključeno je da je najbolji model k najbližih suseda koji je treniran na „undersampled“ skupu podataka. Njegova senzitivnost od gotovo 75% poprilično smanjuje grešku prve vrste (false negative) koja je najopasnija za osiguravajuću kuću jer bi na taj način izbegla mnogo slučajeva u kojima ne bi promovisala ovaj tip osiguranja kod korisnika koji su za njega i zainteresovani. Ovaj model ima i najbolju vrednost metrike F1 što pokazuje da je najbolji i u slučaju kada su greška prve i druge vrste podjednako loše. Ukoliko bi se kompanija služila jako skupim resursima za promovisanje i smatrala kao najveći problem neadekvatno uložen novac u promovisanje ovog tipa osiguranja korisnicima koji definitivno nisu zainteresovani za njega, najbolje bi bilo koristiti model potpornih vektora koji ima najveću preciznost (67%).

7. Zaključak

Tema, odnosno cilj ovog rada bila je klasifikacija već postojećih korisnika osiguravajuće kuće u odnosno na to da li žele da kupe polisu osiguranja karavan vozila, kao i klasterovanje korisnika u grupe gde bi se u istim klasterima nalazili korisnici sa sličnim karakteristikama. Tokom rada, bilo je različitih izazova koje je trebalo savladati. Za početak, ključno je bilo razumeti sam domen industrije osiguranja, uticaj računarske inteligencije na industriju, kao i to na koji način se kvalitetni modeli klasifikacije i klasterovanja mogu iskoristiti u praksi. Nakon toga je bilo potrebno bolje se upoznati sa strukturom podataka, što je u ovom slučaju bilo jako bitno radi razumevanja na koji način neizbalansirana struktura podataka može da utiče na snagu modela. Takođe, originalni skup podataka je uključivao čak 86 atributa od kojih mnogi nisu imali prediktivnu moć, tako da je jako bitan deo rada predstavljala redukcija atributa i formiranje završnog skupa podataka, nakon čega je tek bilo moguće raditi na samim modelima.

Dobijeni završni skup podataka je sadržao 36 promenljivih, a pored originalnog skupa sa neizbalansiranom strukturom napravljena su još dva sa izrazito balansiranom strukturom tako što su uklonjene opservacije većinske klase ili pridodate opservacije manjinske klase. Pre rada na samom klasifikatoru, vrsta neuronskih mreža samoorganizujuće mape iskorišćena je za klasterovanje podataka na osnovu visine prihoda korisnika u svakoj od opština odakle dolaze. Dobijeno je 16 neurona, posle čega su primenom K-means algoritma dobijena tri klastera u kojima se mogla videti jasna distinkcija u primanjima korisnika što je i bila želja na samom početku izrade modela.

Nakon toga, samoorganizujuće mape su iskorišćene i za rešavanje problema klasifikacije, pri čemu su se najbolje pokazale samoorganizujuće mape na „undersampled“ skupu podataka gde je senzitivnost iznosila 51%. Da bi se stekao bolji uvid u to da li ovaj model nije idealan za dati skup podataka ili jednostavno podaci sami po sebi su takve prirode da iz njih nije moguće izvući puno korisnih informacija, napravljena su još dva klasifikatora koji su dali jako zanimljive rezultate. Pokazalo se da je ipak moguće izvući dosta više iz ovog skupa podataka jer su i model potpornih vektora i k najbližih suseda dali značajno bolje rezultate senzitivnosti i preciznosti, ali samo na „undersampled“ i „oversampled“ skupu podataka. Što se tiče senzitivnosti najbolje se pokazao model *k* najbližih suseda na „undersampled“ skupu (74%), dok je najbolju preciznost imao model potpornih vektora na istom tom skupu (67%). Gledajući ove dve metrike, modeli potpronih vektora i *k* najbliži suseda na neizbalansiranom skupu su se najlošije pokazali od svih 9 kombinacija, međutim isto tako imali su ubedljivo najveću tačnost.

Razlog tome je upravo struktura podataka, jer ova dva modela su poprilično zanemarivali manjinsku, pozitivnu klasu, koja je za ovaj rad bila ključna.

Prilikom analize ovih rezultata treba imati u vidu da su podaci koji su korišćeni stvarni podaci jedne osiguravajuće kuće, tako da nije bilo realno za očekivati postizanje izrazito dobrih rezultata metrika, jer se u primerima u praksi to retko i dešava. Takođe, treba imati u vidu i da su stvarni podaci u pitanju, nekad i nije moguće izvući veliko znanje iz njih, jer računarska inteligencija koliko god bila sofisticirana uvek zavisi od kvaliteta ulaznih podataka.

Ipak, postignuti rezultati se definitivno mogu unaprediti za poboljšanje poslovanja osiguravajuće kuće. Rezultati se potencijalno mogu unaprediti modifikovanjem strukture podataka i na druge načine jer je jasno da je ona imala najznačajniju ulogu u njihovom poboljšanju, ali i treniranjem novih vrsta modela koji bi možda dali i nešto drugačije rezultate. Takođe, jedan od načina na koji bi moglo da se dođe do unapređenja je detaljnije ispitivanje korisnika radi dobijanja informacija šta je to što je bilo presudno u njihovoj odluci i koji su sve faktori uticali. Na taj način bi ulazni podaci mogli da se mnogo kvalitetnije organizuju, a do sada je definitivno postalo jasno da povećavanjem kvaliteta ulaza povećavamo mogućnost za unapređenjem izlaza.

8. Literatura

6 Ways Machine Learning is Changing Insurance. (n.d.). Netguru Blog on Fintech. Pristupio 16.4.2020. https://www.netguru.com/blog/machine-learning-insurance-insurtech-fintech-underwriting

A Brief Overview of the Insurance Sector. (29.1.2020). Investopedia. Pristupio 12.4.2020. https://www.investopedia.com/ask/answers/051915/how-does-insurance-sector-work.asp

Artificial Neural Network. (n.d.). Artificial Neural Network - an overview | ScienceDirect Topics. Pristupio 10.5.2020. https://www.sciencedirect.com/topics/earth-and-planetary-sciences/artificial-neural-network

Boruta a wrapper algorithm in R package to perform feature selection. (25.6.2019). Analytics Vidhya. Pristupio 15.5.2020. https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/

Clustering Introduction & different methods of clustering. (14.9.2019). Analytics Vidhya. Pristupio 4.6.2020. https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/

Davis, J., Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240). ACM.

Feature selection in Python using Filter method. (17.11.2019). Medium. Pristupio 10.5.2020. https://towardsdatascience.com/feature-selection-in-python-using-filter-method-7ae5cbc4ee05

Federal Bureau of Investigation. (17.3.2010). Insurance Fraud. FBI. Pristupio 16.4.2020. https://www.fbi.gov/stats-services/publications/insurance-fraud

Hagan, M. T., Demuth, H. B., Beale, M. (2002). Neural Network Design. China Machine Press, Beijing.

Having an Imbalanced Dataset? Here Is How You Can Fix It.(20.4.2019). Medium. Pristupio 3.5.2020. https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb

Jovanović, J. (2016). Mašinsko učenje. Laboratorija za veštačku inteligenciju. Pristupio 13.4.2020. http://ai.fon.bg.ac.rs/wp-content/uploads/2016/10/ML_intro_2016.pdf.

Kohonen R documentation. (n.d.). Pristupio 5.6.2020. https://www.rdocumentation.org/packages/kohonen/versions/2.0.19/topics/som

MathWorks. (n.d.). Neural Networks Provide Solutions to Real-World Problems: Powerful new algorithms to explore, classify, and identify patterns in data. MATLAB & Simulink. Pristupio 10.5.2020. https://www.mathworks.com/company/newsletters/articles/neural-networks-provide-solutions-to-real-world-problems-powerful-new-algorithms-to-explore-classify-and-identify-patterns-in-data.html

Riese, F.M., Keller, S., Hinz, S. (2019). Supervised and Semi-Supervised Self-Organizing Maps for Regression and Classification Focusing on Hyperspectral Data. Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology.

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development.

Self-Organizing Maps. (n.d.). Rpubs. Pristupio 4.6.2020. https://rpubs.com/AlgoritmaAcademy/som

Spearman's Rank-Order Correlation. (n.d.). Spearman's Rank-Order Correlation - A guide to when to use it, what it does and what the assumptions are.Pristupio 10.5.2020. https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php

The 5 Feature Selection Algorithms every Data Scientist should know. (28.7.2019). Medium. Pristupio 9.5.2020. https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2

Tizhoosh, H.R. (2019). Machine Intelligence, Clustering, K-means, SOM. University of Waterloo. Pristupio 17.5.2020.

Unakrsna prodaja ili cross-selling: prednosti i strategije. (24.5.2019). Marketing Fancier. Pristupio 12.4.2020. https://marketingfancier.com/unakrsna-prodaja-ili-cross-selling/

Univariate Analysis: Definition, Examples. (20.1.2019). Statistics How To. Pristupio 5.5.2020. https://www.statisticshowto.com/univariate/

What is Exploratory Data Analysis? (23.5.2018). Medium. Pristupio 4.5.2020. https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

XYF R documentation. (n.d.). Pristupio 6.6.2020. https://www.rdocumentation.org/packages/kohonen/versions/2.0.19/topics/xyf

Prilog: Kod

Struktura podataka

```
train.data <- read.csv("ticdata2000.csv.xls",header = F,  stringsAsFactors = FALSE)

test.data <- read.csv("ticeval2000.csv.xls",header = F,  stringsAsFactors = FALSE)

test.target <- read.csv("tictarget.csv",header = F,  stringsAsFactors = FALSE)

test.data <- cbind(test.data,test.target)

summary(train.data)

str(train.data)

table(train.data$Num.of.mobile.home.policies)

plot(table(train.data$Num.of.mobile.home.policies), type = "h", col = "red", lwd = 15, xlab = "Da li je korisnik kupio osiguranje: 0 - Ne, 1 - Da", ylab = "Broj korisnika", main = "Prikaz nebalansirane strukture izlazne promenljive")
```

Univarijaciona analiza podataka

```
str(train.data)

summary(train.data)

# prva kategorija

apply(data[6:43], 2, summary)

apply(data[6:43], 2, sd)

# druga kategorija

uni.eda.cont <- apply(data[44:64], 2, summary)

apply(data[44:64], 2, sd)

class(uni.eda.cont)

data.uni.eda.cont <- as.data.frame(uni.eda.cont)

sum(data.uni.eda.cont["Max.",]==9)

sum(data.uni.eda.cont["Median",]!=0)

sum(data.uni.eda.cont["3rd Qu.",]!=0)

# treća kategorija

uni.eda.num.of <- apply(data[65:85], 2, summary)

apply(data[65:85], 2, sd)

class(uni.eda.num.of)

data.uni.eda.num.of <- as.data.frame(uni.eda.num.of)

sum(data.uni.eda.num.of["Max.",]==12)

sum(data.uni.eda.num.of["Max.",]==1)

sum(data.uni.eda.num.of["Median",]!=0)

sum(data.uni.eda.num.of["3rd Qu.",]!=0)
```

Multivarijaciona analiza podataka

```
by(data$Cont.car.pol, data$Num.of.mobile.home.policies, summary)

by(data$Cont.fire.pol, data$Num.of.mobile.home.policies, summary)

boxplot(data$Cont.car.pol~data$Num.of.mobile.home.policies, col = c("grey","gold"),
    main = "Ulaganje u osiguranje automobila u odnosu na izlaznu promenljivu",
    xlab = "Da li je korisnik kupio osiguranje karavana",
    ylab = "Visina ulaganja u osiguranje automobila")

boxplot(data$Cont.fire.pol~data$Num.of.mobile.home.policies,
    col = c("grey","gold"), main = "Ulaganje u osiguranje od požara u odnosu na izlaznu promenljivu",
    xlab = "Da li je korisnik kupio osiguranje karavana",
    ylab = "Visina ulaganja u osiguranje od požara")
```

Filter metode

```
install.packages("corrplot")

library(corrplot)

train.data[,86] <- as.integer(train.data[,86])

corr_matrix<- cor(train.data[,c(16:29,37:42,86)])

corr_matrix<- cor(train.data[,c(16:29,37:42,86)], method = "spearman")

corrplot(corr_matrix, type = "upper", diag = F,tl.pos = NULL, tl.cex = 0.3, tl.col = "red", tl.offset = 0.4,
tl.srt = 90)

data <- rbind(train.data,test.data)

corr_matrix_2<- cor(data[,c(16:29,37:42)])

corrplot(corr_matrix_2, type = "upper", diag = F,tl.pos = NULL, tl.cex = 0.3, tl.col = "red", tl.offset = 0.4,
tl.srt = 90)

corr_matrix_2<- cor(train.data[,c(44:64,86)])

corrplot(corr_matrix_2, type = "upper", diag = F,tl.pos = NULL, tl.cex = 0.3, tl.col = "red", tl.offset = 0.4,
tl.srt = 90)

corr_matrix_3<- cor(data[,c(16:18,25:29,37:41,86)])

corrplot(corr_matrix_3, type = "upper", diag = F,tl.pos = NULL, tl.cex = 0.3, tl.col = "red", tl.offset = 0.4,
tl.srt = 90)

#Dobijanje agregirane varijable za uloženi novac u ostala osiguranja

train.data.other.contribution <- train.data[, 44:64]

train.data.other.contribution$Cont.private.third.party.insr <- NULL

train.data.other.contribution$Cont.car.pol <- NULL

train.data.other.contribution$Cont.fire.pol <- NULL

train.data.other.contribution$Cont.boat.pol <- NULL

train.data.other.contribution <- as.data.frame(apply(train.data.other.contribution, 1, sum))

#Dobijanje agregirane varijable za broj kupljenih ostalih osiguranja
```

```
train.data.other.number.of.pol <- train.data[, 65:85]
train.data.other.number.of.pol$Num.of.private.third.party.insur <- NULL
train.data.other.number.of.pol$Num.of.car.pol <- NULL
train.data.other.number.of.pol$Number.of.fire.pol <- NULL
train.data.other.number.of.pol$Number.of.boat.pol <- NULL
train.data.other.number.of.pol <- as.data.frame(apply(train.data.other.number.of.pol, 1, sum))
train.data.new <- train.data[,-c(6:9, 42, 16:18, 25:29, 45, 46, 48:58, 60, 62:64, 66, 67, 69:79, 81, 83:85)]
train.data.new$Other.contribution <- train.data.other.contribution
train.data.new$Number.of.other.policies <- train.data.other.number.of.pol
#Za data set za testiranje dobijanje agregirane varijable za uloženi novac u ostala osiguranja
test.data.other.contribution <- test.data[, 44:64]
test.data.other.contribution$Cont.private.third.party.insr <- NULL
test.data.other.contribution$Cont.car.pol <- NULL
test.data.other.contribution$Cont.fire.pol <- NULL
test.data.other.contribution$Cont.boat.pol <- NULL
test.data.other.contribution <- as.data.frame(apply(test.data.other.contribution, 1, sum))
#Za data set za testiranje dobijanje agregirane varijable za broj kupljenih ostalih osiguranja
test.data.other.number.of.pol <- test.data[, 65:85]
test.data.other.number.of.pol$Num.of.private.third.party.insur <- NULL
test.data.other.number.of.pol$Num.of.car.pol <- NULL
test.data.other.number.of.pol$Number.of.fire.pol <- NULL
test.data.other.number.of.pol$Number.of.boat.pol <- NULL
test.data.other.number.of.pol <- as.data.frame(apply(test.data.other.number.of.pol, 1, sum))
test.data.new <- test.data[,-c(6:9, 42, 16:18, 25:29, 45, 46, 48:58, 60, 62:64, 66, 67, 69:79, 81, 83:85)]
test.data.new$Other.contribution <- test.data.other.contribution
test.data.new$Number.of.other.policies <- test.data.other.number.of.pol
```

Obmotavajuće metode – Boruta algoritam

```
# Boruta algoritam
library(Boruta)
set.seed(111)
borutaAlgorithm <- Boruta(Num.of.mobile.home.policies ~ ., data = train.data.new, doTrace = 2)
print(borutaAlgorithm)
borutaAlgorithm$finalDecision
# Varijable koje nisu odbacene
nonRejected <- getNonRejectedFormula(boruta)
```

```
# Varijable koje su potvrdjene kao znacajne
confirmed <- getConfirmedFormula(boruta)
# las = 2 - nazivi atributa da bufduvertikalni, cex.axis - velicina fonta
plot(borutaAlgorithm, las = 2, cex.axis = 0.7)
plotImpHistory(borutaAlgorithm)
attStats(borutaAlgorithm)
```

Klasterovanje korišćenjem samoorganizujućih mapa

```
train.data.new.boruta.income.som <-
train.data.new.boruta[,c('Income.less.then.30K','Income.in.30.45K','Income.in.45.75K','Income.in.75.122K','Income.more.then.123K')]
str(train.data.new.boruta.income.som)
summary(train.data.new.boruta.income.som)
# Kako su opsezi identični nije potrebna normalizacija
library(kohonen)
train.data.new.boruta.income.som.matrix <- as.matrix(train.data.new.boruta.income.som)
#SOM
#specifing grid
set.seed(111)
g <- somgrid(xdim = 4, ydim = 4, topo = "rectangular")
map <- som(train.data.new.boruta.income.som.matrix, grid = g, alpha = c(0.05,0.01),
        radius = 1, keep.data = TRUE, dist.fcts = "euclidean")
plot(map, type = 'codes', palette.name = rainbow, main = "Mapping of codes")
plot(map, type = 'mapping')
map$unit.classif
train.data.new.boruta.income.som[,6] <- map$unit.classif
head(train.data.new.boruta.income.som)
# Open a bigger device window explicitly
dev.new(width=10, height=10)
plot(map, type = 'changes')
plot(map)
map$codes
plot(map, type = 'counts')
plot(map, type = 'dist.neighbours')
heatmap.som <- function(model){
  for (i in 1:5) {
    plot(model, type = "property", property = getCodes(model)[,i],
```

```
        main = colnames(getCodes(model))[i])
      }
    }
    par(mfrow=c(3,2))
    heatmap.som(map)
    par(mfrow=c(1,1))
    # clustering
    library(factoextra)
    set.seed(111)
    fviz_nbclust(map$codes[[1]], kmeans, method = "wss")
    set.seed(111)
    clust <- kmeans(map$codes[[1]], 3)
    plot(map, type = "codes", bgcol = rainbow(9)[clust$cluster], main = "Cluster Map")
    add.cluster.boundaries(map, clust$cluster)
    # know cluster each data
    train.data.new.boruta.income.som[,6] <- NULL
    ads.cluster <- data.frame(train.data.new.boruta.income.som, Cluster = clust$cluster[map$unit.classif])
    View(ads.cluster)
```

Klasifikacija korišćenjem samoorganizujućih mapa

```
    train.data.new.boruta.ssom.normalizedX <- scale(train.data.new.boruta[,-35])
    test.data.new.boruta.ssom.normalizedX <- scale(test.data.new.boruta[,-35],
                                    center = attr(train.data.new.boruta.ssom.normalizedX, "scaled:center"),
                                    scale = attr(train.data.new.boruta.ssom.normalizedX, "scaled:scale"))
    train.data.new.boruta.ssom.normalizedY <- train.data.new.boruta[,35]
    Y <- test.data.new.boruta[,35]
    test.data.new.boruta.targetzero <- test.data.new.boruta
    test.data.new.boruta.targetzero[,35] <- 0
    test.data.new.boruta.ssom.normalizedXY <- list(independent = test.data.new.boruta.ssom.normalizedX,
dependent = test.data.new.boruta.targetzero[,35])
    # Izgradnja modela
    library(kohonen)
    set.seed(111)
    map.ssom <- xyf(train.data.new.boruta.ssom.normalizedX,
            classvec2classmat(factor(train.data.new.boruta.ssom.normalizedY)),
            grid = somgrid(5,5,"hexagonal"),
            rlen = 100)
```

```
plot(map.ssom, type = 'changes')
plot(map.ssom)
plot(map.ssom, type = 'count')
# Predikcija
pred.test <- predict(map.ssom, newdata = test.data.new.boruta.ssom.normalizedXY)
pred.test.cm <- table(Predicted = pred.test$predictions[[2]], Actual = Y)
pred.test.eval <- compute.eval.metrics(pred.test.cm)
pred.test.eval
# Granice klastera
par(mfrow = c(1,2))
plot(map.ssom, type = 'codes', main = c("Unsupervised SOM", "Supervised SOM"))
map.ssom.hc <- cutree(hclust(dist(map.ssom$codes[[2]])), 2)
add.cluster.boundaries(map.ssom, map.ssom.hc)
par(mfrow = c(1,1))
```

Poređenje rezultata samoorganizujućih mapa sa rezultatima metode potpornih vektora i k najbližih suseda

```
# SVM with unbalanced dataset
library(e1071)
set.seed(111)
mymodel <- svm(Num.of.mobile.home.policies ~ ., data = train.data.new.boruta.standardized,
          kernel = "radial")
summary(mymodel)
svm.pred <- predict(mymodel, test.data.new.boruta.standardized)
tab <- table(Predicted = svm.pred, Actual = test.data.new.boruta.standardized$Num.of.mobile.home.policies)
tab
svm.eval <- compute.eval.metrics(tab)
svm.eval
# Tuning
set.seed(123)
tmodel <- tune(svm, Num.of.mobile.home.policies ~ ., data = train.data.new.boruta.standardized,
    ranges = list(epsilon = seq(0,0.3,0.1), cost = 2^(2:3)))
dev.new(width=10, height=10)
plot(tmodel)
summary(tmodel)
mymodel <- tmodel$best.model
```

```
summary(mymodel)

set.seed(111)

svm.pred <- predict(mymodel, test.data.new.boruta.standardized)

tab <- table(Predicted = svm.pred, Actual = test.data.new.boruta.standardized$Num.of.mobile.home.policies)

tab

svm.eval <- compute.eval.metrics(tab)

svm.eval

# SVM with undersampled dataset

library(e1071)

set.seed(111)

mymodel <- svm(Num.of.mobile.home.policies ~ ., data = train.data.new.boruta.under.standardized,

        kernel = "sigmoid")

summary(mymodel)

svm.pred <- predict(mymodel, test.data.new.boruta.under.standardized)

tab <- table(Predicted = svm.pred, Actual =
test.data.new.boruta.under.standardized$Num.of.mobile.home.policies)

tab

svm.eval <- compute.eval.metrics(tab)

svm.eval

# Tuning

set.seed(123)

tmodel <- tune(svm, Num.of.mobile.home.policies ~ ., data = train.data.new.boruta.under.standardized,

        ranges = list(epsilon = seq(0,0.7,0.1), cost = 2^(2:5)))

dev.new(width=10, height=10)

plot(tmodel)

summary(tmodel)

mymodel <- tmodel$best.model

summary(mymodel)

set.seed(111)

svm.pred <- predict(mymodel, test.data.new.boruta.under.standardized)

tab <- table(Predicted = svm.pred, Actual =
test.data.new.boruta.under.standardized$Num.of.mobile.home.policies)

tab

svm.eval <- compute.eval.metrics(tab)

svm.eval

# SVM with oversampled dataset

library(e1071)
```

```
set.seed(111)
mymodel <- svm(Num.of.mobile.home.policies ~ ., data = train.data.new.boruta.over.standardized,
        kernel = "sigmoid")
summary(mymodel)
svm.pred <- predict(mymodel, test.data.new.boruta.over.standardized)
tab <- table(Predicted = svm.pred, Actual =
test.data.new.boruta.over.standardized$Num.of.mobile.home.policies)
tab
svm.eval <- compute.eval.metrics(tab)
svm.eval
# Tuning
set.seed(123)
tmodel <- tune(svm, Num.of.mobile.home.policies ~ ., data = train.data.new.boruta.under.standardized,
        ranges = list(epsilon = seq(0,0.7,0.1), cost = 2^(2:5)))
dev.new(width=10, height=10)
plot(tmodel)
summary(tmodel)
mymodel <- tmodel$best.model
summary(mymodel)
set.seed(111)
svm.pred <- predict(mymodel, test.data.new.boruta.under.standardized)
tab <- table(Predicted = svm.pred, Actual =
test.data.new.boruta.under.standardized$Num.of.mobile.home.policies)
tab
svm.eval <- compute.eval.metrics(tab)
svm.eval
# KNN with unbalanced dataset
str(train.data.new.boruta.under)
apply(X = train.data.new.boruta.under[,-35],
    MARGIN = 2,
    FUN = function(x) length(boxplot.stats(x)$out))
train.data.new.boruta.under.st <- apply(X = train.data.new.boruta[,-35],
            MARGIN = 2,
            FUN = function(x) scale(x, center = TRUE, scale =
                        TRUE))
train.data.new.boruta.st <- as.data.frame(train.data.new.boruta.st)
train.data.new.boruta.st$Num.of.mobile.home.policies <- train.data.new.boruta$Num.of.mobile.home.policies
```

```
str(train.data.new.boruta.st)
summary(train.data.new.boruta.st)
library(class)
# create a knn model with k=5
set.seed(111)
knn.pred <- knn(train = train.data.new.boruta.standardized[,-35],
                test = test.data.new.boruta.standardized[,-35],
                cl = train.data.new.boruta.standardized$Num.of.mobile.home.policies,
                k = 11)
# print several predictions
head(knn.pred)
# create the confusion matrix
knn.cm <- table(predicted = knn.pred, true = test.data.new.boruta.standardized$Num.of.mobile.home.policies)
knn.cm
# compute the evaluation metrics
knn.eval <- compute.eval.metrics(knn.cm)
knn.eval
# cross-validation
library(e1071)
# define cross-validation (cv) parameters; we'll perform 10-fold crossvalidation
numFolds = trainControl( method = "cv", number = 10)
# define the range for the k values to examine in the cross-validation
cpGrid = expand.grid(.k = seq(from=3, to = 5, by = 2))
set.seed(10320)
# run the cross-validation
knn.cv <- train(x = train.data.new.boruta.st[,-35],
                y = train.data.new.boruta.st$Num.of.mobile.home.policies,
                method = "knn",
                trControl = numFolds,
                tuneGrid = cpGrid, )
knn.cv
# plot the cross-validation results
plot(knn.cv)
# build a new model with the best value for k
best_k <- knn.cv$bestTune$k
best_k
```

```
knn.pred2 <- knn(train = train.data.new.boruta.st[,-35],
                 test = test.data.new.boruta[,-35],
                 cl = train.data.new.boruta.st$Num.of.mobile.home.policies,
                 k = best_k)
# create the confusion matrix
knn.cm2 <- table(predicted = knn.pred2, true = test.data.new.boruta$Num.of.mobile.home.policies)
knn.cm2
# compute the evaluation metrics
knn.eval2 <- compute.eval.metrics(knn.cm2)
knn.eval2
# KNN with undersampled dataset
str(train.data.new.boruta)
apply(X = train.data.new.boruta[,-35],
    MARGIN = 2,
    FUN = function(x) length(boxplot.stats(x)$out))
train.data.new.boruta.st <- apply(X = train.data.new.boruta[,-35],
                                  MARGIN = 2,
                                  FUN = function(x) scale(x, center = TRUE, scale =
                                                            TRUE))
train.data.new.boruta.st <- as.data.frame(train.data.new.boruta.st)
train.data.new.boruta.st$Num.of.mobile.home.policies <- train.data.new.boruta$Num.of.mobile.home.policies
str(train.data.new.boruta.st)
summary(train.data.new.boruta.st
library(class)
# create a knn model with k=5
set.seed(111)
knn.pred <- knn(train = train.data.new.boruta.under.standardized[,-35],
                test = test.data.new.boruta.under.standardized[,-35],
                cl = train.data.new.boruta.under.standardized$Num.of.mobile.home.policies,
                k = 49)
# print several predictions
head(knn.pred)
# create the confusion matrix
knn.cm <- table(predicted = knn.pred, true =
test.data.new.boruta.under.standardized$Num.of.mobile.home.policies)
knn.cm
```

```
# compute the evaluation metrics
knn.eval <- compute.eval.metrics(knn.cm)
knn.eval
# cross-validation
library(e1071)
# define cross-validation (cv) parameters; we'll perform 10-fold crossvalidation
numFolds = trainControl( method = "cv", number = 10)
# define the range for the k values to examine in the cross-validation
cpGrid = expand.grid(.k = seq(from=3, to = 5, by = 2))
set.seed(10320)
# run the cross-validation
knn.cv <- train(x = train.data.new.boruta.st[,-35],
                y = train.data.new.boruta.st$Num.of.mobile.home.policies,
                method = "knn",

trControl = numFolds,
                tuneGrid = cpGrid, )
knn.cv
# plot the cross-validation results
plot(knn.cv)
# build a new model with the best value for k
best_k <- knn.cv$bestTune$k
best_k
knn.pred2 <- knn(train = train.data.new.boruta.st[,-35],
                 test = test.data.new.boruta[,-35],
                 cl = train.data.new.boruta.st$Num.of.mobile.home.policies,
                 k = best_k)
# create the confusion matrix
knn.cm2 <- table(predicted = knn.pred2, true = test.data.new.boruta$Num.of.mobile.home.policies)
knn.cm2
# compute the evaluation metrics
knn.eval2 <- compute.eval.metrics(knn.cm2)
knn.eval2
# KNN with oversampled dataset
library(class)
# create a knn model with k=5
```

```
set.seed(111)
knn.pred <- knn(train = train.data.new.boruta.over.standardized[,-35],
                test = test.data.new.boruta.over.standardized[,-35],
                cl = train.data.new.boruta.over.standardized$Num.of.mobile.home.policies,
                k = 29)
# print several predictions
head(knn.pred)
# create the confusion matrix
knn.cm <- table(predicted = knn.pred, true =
test.data.new.boruta.over.standardized$Num.of.mobile.home.policies)
knn.cm
# compute the evaluation metrics
knn.eval <- compute.eval.metrics(knn.cm)
knn.eval
```