Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Dejan Ognjenović

Ugotavljanje konsistentnosti anketnih odgovorov s strojnim učenjem

diplomsko delo

visokošolski strokovništudijski program prve stopnje računalništvo in informatika

Mentor: prof. dr. Marko Robnik Šikonja

Ljubljana 2011

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00137/2011
Datum: 01.09.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DEJAN OGNJENOVIČ**

Naslov: **UGOTAVLJANJE KONSISTENTNOSTI ANKETNIH ODGOVOROV S STROJNIM UČENJEM**

ANALYSIS OF SURVEY CONSISTENCY WITH MACHINE LEARNING

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Anketiranje je pogost način zajema podatkov pri problemih in analizah v marketingu, sociooloških in psiholoških raziskavah, vendar je, še posebej pri spletnem anketiranju, vprašljiva kvaliteta zajetih podatkov. Eden od načinov preverjanja konsistentnosti odgovorov je kontrola ujemanja več povezanih vprašanj.
Predvidevamo, da lahko s pomočjo algoritmov strojnega učenja ta postopek izboljšamo. Implementirajte postopek, ki se za vsako posamezno anketno vprašanje nauči napovedni model iz vseh ostalih vprašanj. Preverite delovanje nekaj mer, ki konsistentnost in kvaliteto odgovorov anketirancev ocenijo iz napovedljivosti njihovih odgovorov. Predvidevamo, da slabo napovedljivi anketiranci bolj verjetno dajejo nekonsistentne odgovore. Na nekaj umetnih in realnih zbirkah anketnih podatkov preizkusite opisani postopek in rezultate primerjajte s statističnimi tehnikami za določanje izstopajočih primerov.

Mentor:

prof. dr. Marko Robnik Šikonja

Dekan:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Dejan Ognjenović, z vpisno številko **63080327**, sem avtor diplomskega dela z naslovom:

Ugotavljanje konsistentnosti anketnih odgovorov s strojnim učenjem

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Robnik Šikonje,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 14. septembra 2011 Podpis avtorja:

Zahvala.

Zahvalil bi se svojemu mentorju prof. dr. Marku Robnik Šikonji za strokovno usmerjanje in nasvete, potrpežljivost in spodbudo pri nastajanju diplomskega dela. Zahvalil bi se tudi svojim staršem za moralno in materialno podporo med študijem. Zahvala pa gre tudi vsem ostalim, ki so me vedno spodbujali in motivirali.

Hvala!

Kazalo

Povzetek

V nalogi smo ugotavljali kvaliteto anketnih odgovorov. Slaba lastnost anketiranja je, da ne vemo ali odgovori resnično odražajo mnenje anketiranca. Predvidevamo, da so taki anketiranci nekonsistentni in da jih lahko detektiramo z uporabo strojnega učenja.

Naš pristop k reševanju tega problema je s pomočjo strojnega učenja. Ideja je, da za vsako vprašanje v anketi sestavimo napovedni model. Z modeli dobimo distribucijo verjetnosti za vsak odgovor anketiranca. Pri tem uporabimo prečno preverjanje, da dobimo distribucije za vse primere. Distribucije ocenimo z različnimi metodami ocenjevanja kot so npr. Brierjeva ocena, informacijska vsebina, rangiranja, itd. Te ocene združimo in dobimo končno oceno nekonsistentnosti primerov. Nekonsistentne primere lahko za lažjo predstavo izrišemo.

Metodo smo razvili s programskim orodjem R. Uporabili smo pakete CORElearn [14], MASS [20] in rpart [16]. Za vizualizacijo smo uporabili paketa CORElearn in program Orange [5].

Pri testiranju smo uporabili podatkovne množice Monk, B2B, B2C, DSP in Hearing aid. Pri gradnji modelov smo zaradi točnosti najpogosteje uporabili naključna drevesa. Manjkajoče vrednosti smo zamenjali z uporabo metode najbližjih sosedov, modusom, povprečjem in izpustom vrstic. Za testiranje metode smo generirali nekonsistentne podatke in jih poskusili identificirati z metodami ocenjevanja. V ocenah je bila prisotna varianca, ki smo jo zmanjšali s povprečjem. Za lažjo predstavo in identifikacijo smo identificirane primere izrisovali.

Rezultati so odvisni od podatkov in metod ocenjevanja. Brierjeva ocena, verjetnosti, Brierjevo rangiranje ter rangiranje verjetnosti vrnejo dobre rezultate, saj so v večini primerov identificirale vse nekonsistentne primere. Ostale metode delujejo slabše. Predlagani pristop je pri velikem številu podatkov časovno precej zahteven.

Abstract

We researched the quality of survey responses. We don't know if answers really reflect the opinion of interviewees. We believe that inconsistent, respondents can be detected with the use of machine learning techniques.

Our idea is to build a prediction model for every question of a survey. With the models, we get a probability distribution for every answer in the survey. We use cross-validation to get distributions for all instances. We evaluate them with Brier score, information score, probabilities, classification accuracy, Birer ranking, information score ranking, probability ranking and classification accuracy ranking. We merge these scores, and get an inconsistency score for every instance (interviewee) of the survey. We visualize these inconsistent cases for a better comprehension.

We developed the method with the statistical system R and packages CORElearn [14], MASS [20] and rpart [16]. For the visualization we used package CORElearn and data mining software Orange [5].

For testing purposes we used data sets Monk, B2B, B2C, DPS and hearnig aid. As prediction models we mostly used random forests, because of their superb accuraccy. Missing values were imputed with the use of k-nearest neighbor (kNN), modus, mean, or the instance was simply removed from the data. We generated inconsistent data and tried to identify these cases. There were some variance in our incosistency scores, so we reduced it by averaging the scores. For a better comprehension and indetification, we have plotted the cases that were identified as inconsistent.

The results depend on the data and evaluation method. Brier score,

probabilities, Brier ranking and prabability ranking in most cases identified all inconsistent instances (interviewees). Other methods sometimes failed to identify inconsistent cases. The approach is computationaly demanding for larger datasets.

Poglavje 1

Uvod

Strojno učenje je veja računalništva, ki postaja iz dneva v dan bolj popularna in pomembna, saj živimo v svetu polnem podatkov, ki pa jih je potrebno predelati, da iz njih dobimo informacije in znanje. Rezultati strojnega učenja so ponavadi napovedni modeli, lahko pa so tudi funkcije, relacije, pravila in podobno. Metode strojnega učenja lahko v grobem razdelimo glede na uporabo naučenega znanja:

- klasifikacija oziroma uvrščanje,
- regresija,
- razvrščanje,
- učenje asociacij.

Klasifikacija in regresija predstavljata funkcijo, ki preslika prostor atributov v razred, oziroma v številsko vrednost. Z asociacijami je predstavljena logična relacija. Pri razvrščanju razredov ne poznamo in poskušamo na podlagi podobnosti primerov ugotoviti, koliko jih je, in jih določiti.

Napovednh modelov se naučimo iz preteklih primerov in jih uporabljamo za napovedovanje novih še neznanih primerov. Primeri so opisani z atributi oz. značilkami, ki imajo diskretne vrednosti (npr. spol), ali številske (npr. starost). Diskretni atribut, ki ga poskušamo napovedati, se imenuje

razred. Za gradnjo napovednega modela ne uporabimo vseh primerov iz učne množice, ampak jih nekaj prihranimo za testiranje (testni podatki), saj tako ocenimo, koliko je model zanesljiv, oziroma kako dobro napoveduje.

V diplomski nalogi poskušamo s pomočjo strojnega učenja ugotoviti kvaliteto anketnih odgovorov. Pri anketi želimo vedeti ali odgovori resnično odražajo mnenje anketirancev. Obstajajo metode za detekcijo nekvalitetnih odgovorov, ki pa niso najboljše [1, 12]. Eden od pristopov je, da ima sestavljalec anket ljudi, ki jim plačuje za izpolnjevanje anket [12].

Poskusili smo razviti metodo, ki bi bolje ocenila zanesljivost odgovorov kot že obstojoče metode. Naš pristop je, da s pomočjo napovednih modelov ocenimo napovedljivost anketiranca. Za vsako vprašanje v anketi zgradimo svoj napovedni model. Napovedni modeli vrnejo distribucijo verjetnosti za vsak odgovor anketiranca. Pri tem uporabimo prečno preverjanje, da dobimo distribucije za vse anketirance. Distribucije ocenimo z metodami, ki so opisane v 2. poglavju. Vse ocene določenega primera združimo in tako dobimo končno oceno napovedljivosti. Slabša napovedljivost pomeni nekonsistentnost anketiranca. Zaradi variance poženemo metodo večkrat in ocene povprečimo.

Ankete pogosto vsebujejo veliko manjkajočih vrednosti. V našem pristopu jih nadomestimo s povprečjem ali z metodo najbližjih sosedov. Model zgradimo za vsako vprašanje posebej in z modeli napovedujemo odgovore anketirancev. Te napovedi ocenimo in dobimo ocene kvalitete odgovora za vsakega anketiranca. Slabše ocenjene anketirance bi lahko odstranili.

Kvaliteta anket je pogosta tema raziskav različnih področij. Biemer in Lyberg sta to področje podrobno raziskala v svoji knjigi [12]. V njej opisujeta kako sestavljati ankete in kako jih evaluirati. Prav tako sta opisala katere napake lahko ankete vsebujejo (sampling, nonsampling, nonresponse error) in kako jih zmanjšati. Opisujeta tudi kako ravnati z manjkajočimi vrednostmi. Banda opisuje določen tip napak (nonsampling error) med katere spada tudi zanesljivost anketiranca [1]. Chandola, Banerjee in Kumar opisujejo metode za detekcijo izjem in izstopajocih primerov v podatkih [19]. O tem sta pisala

tudi Hodge in Austin [8]. Breiman opisuje kako lahko z uporabo naključnih dreves detektiramo izstopajoče primere [3]. V pregledani literaturi nismo opazili, da bi kdo za ugotavljanje kvalitete anket uporabil strojno učenje.

V 2. poglavju je podrobnejše opisan problem, obstoječi pristopi in naš pristop. V 3. poglavju so na kratko pojasnjeni algoritmi za gradnjo napovednih modelov in metode za ocenjevanje napovedljivosti anketirancev. Poglavje 4 opisuje uporabljene podatkovne množice, njihove spremembe in rezultate našega pristopa. Poglavje 5 podaje zaključke in ideje za izboljšave.

Poglavje 2

Opis problema

Z anketami zbiramo različne podatke. Slaba lastnost anketiranja je, da ne vemo, ali je anketiranec odgovarjal po resnici, oziroma, ali odgovori odražajo njegovo mnenje (*nonsampling error*). Temu pravimo zanesljivost anketiranja. Odgovorom nezanesljivih anketirancev pravimo tudi šum, saj so podatki lahko zavajajoči in nam prikažejo drugačno sliko od dejanskega stanja.

Eden od pristopov, kako napovedovati zanesljivost anketiranca, so vprašanja, ki so med seboj povezana (npr., vprašanja, katerih odgovori morajo biti vsi `DA` ali `NE`). V kolikor se povezani odgovori razlikujejo, lahko anketirancu predpišemo nekonsistentnost. Nekonsistentne anketirance odstranimo. Na ta način detektiramo nekonsistentne anketirance, ne pa tistih, katerih odgovori ne odražajo njihovega resničnega mnenja in mišlenja.

Drug način zagotavljanja zanesljivosti anket je, da ima izvajalec anket množico ljudi, ki jim plačuje za izpolnjevanje anket [12]. Dobimo le malo šuma, saj anketirani, zaradi plačila, po večini odgovarjajo boljše in je tako kvaliteta anket boljša. Metoda je dražja..

Metoda, ki se uporablja predvsem v spletnih anketah, je merjenje odzivnega časa [12]. Metoda izračuna ali izmeri povprečni čas za izpolnitev ankete. Anketirance z večjim odstopanjem od povprečnega časa se odstranijo.

Pristop, ki smo ga uporabili pri zagotavljanju zanesljivosti anket, sodi na področje strojnega učenja. Iz odgovorov skušamo s pomočjo napovednih

/	a_1	a_2	a_3	...	a_m
1	$o_{1,1}$	$o_{1,2}$	$o_{1,3}$	...	$o_{1,m}$
2	$o_{2,1}$	$o_{2,2}$	$o_{2,3}$	...	$o_{2,m}$
3	$o_{3,1}$	$o_{3,2}$	$o_{3,3}$	...	$o_{3,m}$
⋮	⋮	⋮	⋮	⋱	⋮
n	$o_{n,1}$	$o_{n,2}$	$o_{n,3}$	...	$o_{n,m}$

Tabela 2.1: Oblika ankete, ki jo obdelujemo.

modelov dobiti ocene kvalitete odgovorov, s katerimi bi ocenili naučljivost odgovorov anketiranca. Naučljivost odgovorov nam ocenjuje konsistentnost odgovorov anketiranca. Primeri, ki so konsistentni, so lažje naučljivi kot tisti, ki so nekonsistentni. Npr., če anketiranec naključno izpolni anketo, je odgovore z naključnimi vrednosti veliko težje napovedati kot primere, ki ustrezajo vzorcem v podatkih. Nenapovedljivi primeri so torej nekvalitetni.

Tabela 2.1 prikazuje obliko zajetih podatkov. Prva vrstica predstavlja atribute, oziroma vprašanja ankete, prvi stolpec pa zaporedno številko anketiranca. Ostale vrednosti predstavljajo odgovore, ki so jih izbrali anketiranci. Oznaka n predstavlja število anketirancev, m pa število vprašanj.

Za vsak atribut, oziroma vprašanje, sestavimo model, ki bo napovedoval odgovore nanj. Npr., modela za prvo vprašanja se učimo iz vseh ostalih vprašanj. Tako sestavimo toliko modelov, kolikor je vprašanj v anketi. Model za vsak odgovor vrne distribucijo verjetnosti možnih odgovorov, kot prikazuje izraz (2.1). Črka i predstavlja i-ti primer, oziroma vrstico, j pa predstavlja j-ti odgovor, oziroma stolpec.

$$d_{i,j} = \begin{pmatrix} \overset{1}{0.1} & \overset{2}{0.2} & \overset{3}{0.2} & \overset{4}{0.3} & \overset{5}{0.2} \end{pmatrix} \tag{2.1}$$

Pri gradnji napovednih modelov uporabimo prečno preverjanje, da vsak primer enkrat nastopi v testni množici in tako dobimo distribucije verjetnosti za vse primere. V primeru, da prečnega preverjanja nebi uporabili, bi dobili distribucijo verjetnosti samo za testne primere, ne pa zudi za učne.

/	a_1	a_2	a_3	...	a_m
1	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	...	$d_{1,m}$
2	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	...	$d_{2,m}$
3	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	...	$d_{3,m}$
4	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	...	$d_{4,m}$
5	$d_{5,1}$	$d_{5,2}$	$d_{5,3}$	...	$d_{5,m}$
6	$d_{6,1}$	$d_{6,2}$	$d_{6,3}$	...	$d_{6,m}$
⋮	⋮	⋮	⋮	⋱	⋮
n	$d_{n,1}$	$d_{n,2}$	$d_{n,3}$	...	$d_{n,m}$

Tabela 2.2: Oblika ankete, ki nam jo vrne napovedni model.

Ko so modeli sestavljeni in z njimi napovemo odgovore, dobimo matriko distribucij, kot je to prikazano v tabeli 2.2. Odebeljene horizontalne črte predstavljajo množice k-kratnega prečnega preverjanja.

Vsaka vrstica vsebuje verjetnostne distribucije za vsa različna vprašanja. Te distribucije uporabimo z različnim ocenami (Brier, rank, ...), kot je to podrobneje opisano v poglavju 3.2. Za vsakega anketiranca združimo ocene vseh napovedanih odgovorov. Tako dobimo končne ocene, ki jih sortiramo in iz njih razberemo zanesljivost odgovorov. Primere, ki imajo izstopajoče ocene odstranimo, saj so zelo verjetno nezanesljivi.

Zaradi variance lahko z vsakim zagonom metode dobimo drugačne ocene slabih primerov, zato odstranimo le izrazito drugačne primere. Varianco lahko zmanjšamo s povprečenjem večkratne ponovitve gradnje vseh modelov na različnih delitvah primerov pri prečnem preverjanu. Izstopajoče primere lahko grafično predstavimo za lažjo predstavo in kontrolo pravilnosti postopka.

Poglavje 3

Algoritmi in metode

Obstajajo različni algoritmi za gradnjo napovednih modelov. Vsak od njih ima dobre in slabe lastnosti. Ne obstaja algoritem, ki bi bil dober za vse probleme. Primernost modela je odvisna od oblike in narave podatkov in jo je le težko vnaprej predvideti. V tem poglavju so na kratko opisani učni algoritmi, s katerimi smo gradili napovedne modele, metode ocenjevanja modelov ter načini nadomeščanja manjkajočih vrednosti v podatkih.

3.1 Učni algoritmi

Podatki, ki se pojavijo v anketah so različnih oblik. Vsebujejo lahko mankajoče vrednosti, lahko imajo številske ali diskretne vrednosti, vsebujejo lahko veliko šuma in podobno. Pri razvoju rešitve smo zato testirali več različnih algoritmov:

- naključna drevesa (Random Forest),
- naivni Bayes (Naive Bayes),
- metoda podpornih vektorjev (SVM).

Metoda naključnih dreves [2, 3, 17, 18] je klasifikator, ki je sestavljen iz več odločitvenih dreves. Naj bo $\boldsymbol{N}$ množica vseh učnih primerov (anketirancev) in $\boldsymbol{A}$ množica vseh atributov (vprašanj). Algoritem deluje tako, da za vsako izbere s ponavljanjem $n = |\mathbf{N}|$ učnih primerov. V povprečju se tako za učenje vsakega drevesa uporabi 63% vseh učnih primerov, 37% pa jih algoritem ne uporabi. Ti so imenovani OOB (Out-Of-Bag) primeri in se po gradnji uporabljajo za oceno napake. Med gradnjo drevesa algoritem v vsakem vozlišču izbere a atributov, ker je a veliko manjši od $|\mathbf{A}|$ ($log(\mathbf{A})$ ali $\sqrt{\mathbf{A}}$). Ko so drevesa zgrajena, vsako drevo klasificira nov primer in povprečje vseh klasifikacij vrne klasifikator kot rezultat metode.

Prednosti tega klasifikatorja so točnost, detekcija izjem in nemoteno delovanje pri veliki količini vhodnih podatkov. Slabost je nekaj pristranskosti do atributov z večjim številom vrednosti in časovna zahtevnost.

Naivni Bayes [9–11,13] je klasifikator, ki za klasifikacijo uporablja Bayesov teorem. Algoritem deluje tako, da izračuna pogojne verjetnosti za vsak razred pri danih vrednostih atributov, kot to prikazuje formula (3.1). $P(r_k)$ je apriorna verjetnost k-tega razreda, m je število atributov (vprašanj), $P(a_i|r_k)$ je verjetnost i-tega atributa A_i pri dani vrednosti a_i za k-ti razred. $P(apost_k)$ je verjetnost, ki jo vrne klasifikator za k-ti razred. Vrednosti lahko normaliziramo tako, da vsako verjetnost $P(a_i|r_k)$ delimo s $P(r_k)$.

$$P(apost_k) = P(r_k) \prod_{i=1}^{m} P(a_i|r_k) \tag{3.1}$$

Nov primer klasificiramo v razred z maksimalno vrednostjo $P(apost_k)$.

Dobre lastnosti so hitrost, robustnost in skalabilnost, saj klasifikator predpostavlja, da so si atributi med seboj neodvisni. Slabi lastnosti sta naivnost te predpostavke ter nemožnost direktne uporabe zveznih atributov, saj jih je potrebno diskretizirati.

SVM [10, 11, 13, 18] deluje tako, da vsak primer iz učne množice podatkov postavi v visokodimenzionalni vektorski prostor, nato pa s pomočjo podpornih vektorjev poišče hiperploskve, ki najbolje ločujejo vrednosti razredov. Podporni vektorji so dejansko primeri, ki so najbližji hiperploskvi, saj ostali ne vplivajo na lego hiperploskve. Če algoritem razredov ne more ločiti, povečamo število dimenzij hiperprostora.

Prednost SVM je njegova točnost in neprilagajanje učni množici, slabost pa je računska zahtevnost.

3.2 Metode ocenjevanja modelov

Napovedane odgovore smo ocenjevali s:

- klasifikacijsko točnost (classification accuracy) [10],
- Brierjevo mera (Brier score) [6, 10],
- informacijsko vsebino (Information score) [4, 10],
- verjetnostmi (probability),
- rangiranje (Rank).

Ko ocenimo matriko verjetnostnih distribucij, dobimo matriko ocen odgovorov, kjer je vsaka vrstica anketiranec, vsak stolpec pa vprašanje, tako kot to prikazuje tabela 3.1. Skrajno desni stolpec prikazuje končne ocene primerov, skrajno spodnja vrstica pa povprečja ocen atributov.

Končno oceno dobimo tako, da za vsak primer seštejemo vrednosti njegove vrstice. Naj bo m število vseh vprašanj v anketi, ter v_{ij} ocena odgovora anketiranca i pri vprašanju j, potem velja, da je končna ocena i-tega anketiranca enaka:

$$ocena_i = \sum_{j=1}^{m} v_{ij} \tag{3.2}$$

	a_1	a_2	a_3	...	a_m	vsote
1	$v_{1,1}$	$v_{1,2}$	$v_{1,3}$	...	$v_{1,m}$	$ocena_1$
2	$v_{2,1}$	$v_{2,2}$	$v_{2,3}$	...	$v_{2,m}$	$ocena_2$
3	$v_{3,1}$	$v_{3,2}$	$v_{3,3}$	...	$v_{3,m}$	$ocena_3$
⋮	⋮	⋮	⋮	⋱	⋮	⋮
n	$v_{n,1}$	$v_{n,2}$	$v_{n,3}$	...	$v_{n,m}$	$ocena_n$

povprečja	$mean_1$	$mean_2$	$mean_3$	...	$mean_m$

Tabela 3.1: Oblika matrike z ocenami.

Klasifikacijska točnost [10] je najpreprostejša ocena uspešnosti napovednih modelov. Vsaki napovedani odgovor anketiranca se primerja z njegovim dejanskim odgovorom. Če sta odgovora enaka, je ocena pri tem odgovoru 1, drugače pa je ocena 0. Večjo vsoto ocen ima anketiranec, bolj konsistenten je. Tabela 3.2 prikazuje primer matrike s klasifikacijsko točnostjo. Končne ocene lahko normaliziramo tako, da delimo z m, kjer je m število vseh atributov (vprašanj).

	a_1	a_2	a_3	a_4	vsote
1	1	1	0	0	2
2	1	0	1	0	2
3	1	1	1	1	4
4	0	0	0	1	1
5	1	1	1	0	3

povprečja	0.8	0.6	0.6	0.4

Tabela 3.2: Matrika z uporabo klasifikacijske točnosti.

Brierjeva ocena [6, 10] meri napako modela, kar pomeni, da je manjša ocena znamenje boljše kvaliteta odgovorov. Tabela 3.3 prikazuje, matriko z Brierjevo oceno. Naj bo g število vseh možnih odgovorov za eno vprašanje, d naj bo verjetnostna distribucija odgovorov, ki jo je vrnil napovedni model, u pa distribucija, ki predstavlja dejanski odgovor, tako da je $P(u_i) = 1$ za dejansko izbrano verjetnost in $P(u_{j\neq i}) = 0$ za vse ostale vrednosti. Brierjeva ocena za en odgovor je tako:

$$BS = \frac{1}{g}\sum_{j=1}^{g}(u_j - v_j)^2 \tag{3.3}$$

	a_1	a_2	a_3	a_4	vsote
1	0.055	0.238	0.157	0.085	0.535
2	0.080	0.153	0.144	0.085	0.462
3	0.090	0.210	0.150	0.121	0.571
4	0.161	0.159	0.207	0.165	0.692
5	0.055	0.192	0.145	0.085	0.477
povprečja	0.088	0.190	0.160	0.108	

Tabela 3.3: Matrika z Brierjevo oceno.

Informacijska vsebina [4, 10] za razliko od Brierjeve ocene upošteva apriorne verjetnosti razredov in verjetnosti, ki jih vrne napovedni model. Naj bo p apriorna verjetnost dejanskega odgovora na določeno vprašanje, p' pa verjetnost odgovora, kot ga vrne napovedal model. Torej je informacijska vsebina za en odgovor anketiranca enaka:

$$INF_j = \begin{cases} -log_2(p) + log_2(p'), & p' \geq p \\ -(-log_2(1-p) + log_2(1-p')), & p' < p \end{cases} \tag{3.4}$$

	a_1	a_2	a_3	a_4	vsote
1	0.326	2.189	0.028	0.419	2.962
2	0.602	0.866	0.425	0.456	2.349
3	1.306	1.513	1.109	0.008	3.936
4	0.169	0.651	0.037	0.104	0.961
5	0.580	0.940	0.402	0.261	2.183
povprečja	0.596	1.231	0.400	0.249	

Tabela 3.4: Matrika z informacijsko vsebino.

Z **verjetnostmi** ocenjujemo tako, da za vsak dejanski odgovor anketiranca vzamemo njegovo verjetnost iz napovedane distribucije odgovorov. Ta verjetnost služi kot ocena pravilnosti odgovora anketiranca. Večja je napovedana verjetnost odgovora, boljša je ocena. Končne ocene lahko, kot pri klasifikacijski točnosti, normaliziramo. Tabela 3.5 prikazuje matriko verjetnosti.

	a_1	a_2	a_3	a_4	vsote
1	0.238	0.742	0.260	0.066	1.306
2	0.011	0.046	0.015	0.001	0.073
3	0.346	0.178	0.361	0.317	1.202
4	0.883	0.815	0.802	0.301	2.801
5	0.130	0.959	0.381	0.182	1.652
povprečja	0.321	0.548	0.363	0.173	

Tabela 3.5: Matrika z verjetnostmi.

Odgovore anketirancev ocenimo z Brierjevo oceno, informacijsko vsebino ali s katero drugo metodo. **Rangovna metoda** oceni odgovore tako, da za vsako vprašanje posebej razvrsti ocene odgovorov po naraščajočem vrstnem redu, pri tem pa je rang oziroma pozicija, ocena odgovora. Nižja uvrstitev pomeni slabšo oceno, kar kaže na nenapovedljivost in nekonsistentost primera. Rangiranje je način ocenjevanja, ki lahko združi tudi klasifikacijske in regresijske modele. Tabela 3.6 prikazuje rangirane verjetnosti iz tabele 3.5.

	a_1	a_2	a_3	a_4	vsote
1	3	3	2	2	10
2	1	1	1	1	4
3	4	2	3	5	14
4	5	4	5	4	18
5	2	5	4	3	14

Tabela 3.6: Matrika z rangiranjem verjetnosti.

3.3 Nadomeščanje manjkajočih vrednosti

Ankete lahko vsebujejo mnogo manjkajočih vrednosti. Uporabili smo:

- kNN (metoda najbližjih sosedov) napoved manjkajoče vrednosti,
- izpust vrstice,
- povprečje oz. modus.

Algoritem **kNN** [11,13,17] s pomočjo razdalj izračuna podobnosti primerov (anketirancev) in namesto manjkajoče vrednosti vstavi najbolj pogost odgovor k najbližjih primerov. Razdalje za izračun podobnosti primerov so evklidska, manhatanska, hammingova, itd. Uporabili smo evklidsko razdaljo.

Izpust vrstice [7] je najpreprostejši način za delo z manjkajočimi vrednosti, saj enostavno izpustimo primer, ki vsebuje manjkajočo vrednost. V primeru, ko je manjkajočih vrednosti malo, je ta metoda sprejemljiva, če pa imamo veliko manjkajočih vrednosti, je praktično neuporabna.

Povprečje oziroma **modus** [7] vstavi povprečno vrednost številskega atributa v manjkajočo vrednost. V primeru diskretnega atributa se vstavi najbolj pogosta vrednost atributa (modus).

Poglavje 4

Testiranje in rezultati

V tem poglavju so najprej predstavljene podatkovne množice, nato pa testiranje in rezultati. Uporabili smo podatkovne množice monk, ankete B2B, B2C, DPS ter hearing aid. Za izrisovanje grafov smo uporabili paket CORElearn v0.9.35 v jeziku R, ter program Orange. S CORElearn smo izrisovali tipične primere, z Orange pa celotno zbirko podatkov.

4.1 Podatkovne množice in njihova obdelava

V tem razdelku opisujemo podatkovne množice, na katerih smo testirali našo metodo, opisujemo dodane primere ter kako generiramo nekonsistentne primere.

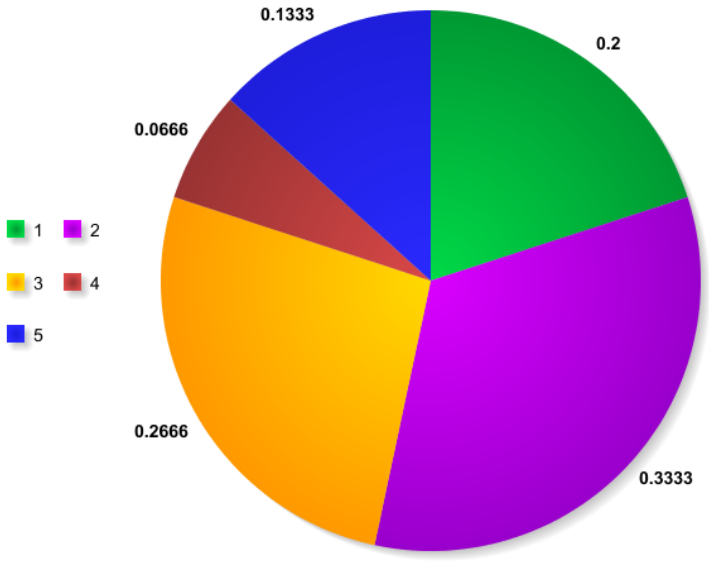
4.1.1 Generiranje primerov z obratnim rangiranjem

Za generiranje testnih primerov smo vzeli nekaj primerov z najvišjo oceno in v naraščajočem vrstnem redu uredili verjetnosti odgovorov v vsaki od njihovih napovedi. To pomeni, da smo odgovor z najnižjo verjetnostjo postavili na najvišje mesto (glej tabelo 4.1). Sestavili smo novo verjetnostno distribucijo tako, da smo dobljeni rang (mesto) delili z komulativno vsoto vseh mest (tabela 4.1). Vsak odgovor tako dobi distribucijo verjetnosti. Pri sestavljanju testnih primerov za vsako vprašanje naključno izberemo odgovor, s

tem da upoštevamo novo distribucijo verjetnosti (slika 4.1). Uporabili smo metodo ruletnega kolesa [10].Tako z večjo verjetnostjo generiramo nekonsistenten primer, saj imamo večjo verjetnost, da izberemo odgovor, ki je bil v originalni distribuciji najmanj verjeten.

#	1	2	3	4	5
verjetnost	0.2	0.1	0.15	0.3	0.25
Rang	3	1	2	5	4
Obratni rang	3	5	4	1	2
Nove verjetnosti	$\frac{3}{15}$	$\frac{5}{15}$	$\frac{4}{15}$	$\frac{1}{15}$	$\frac{2}{15}$

Tabela 4.1: Tabela prikazuje primer obratnega rangiranja.



Slika 4.1: Primer nove distribucije verjetnosti za primer iz tabele 4.1.

4.1.2 Monk

Podatkovna zbirka monk ne vsebuje anketnih odgovorov, vendar je primerna za testiranje našega pristopa. Prednost te podatkovne zbirke je lažje testiranje metod, saj poznamo podatke in pravila, kako so bili generirani.

Atributi	Razred	a1	a2	a3	a4	a5	a6
Vrednosti	0,1	1,2,3	1,2,3	1,2	1,2,3	1,2,3,4	1,2

Tabela 4.2: Tabela prikazuje atribute in vrednosti zbirke monk.

Podatkovno zbirko monk sestavlja 6 diskretnih atributov in eden razred. Zbirka vsebuje 432 primerov. Tabela 4.2 prikazuje vrednosti, ki jih lahko vsebujejo atributi te podatkovne zbirke. Zbirka je sestavljena iz 3 različnih problemov. Uporabili smo 1. in 3 zbirko.

Primer v prvi podatkovni zbirki ima razred 1, če

$$(a1 = a2) \vee (a5 = 1)$$

drugače pa je razred 0.

V tretji podatkovni zbirki ima primer razred 1, če

$$((a5 = 3) \wedge (a4 = 1)) \vee ((a5 \neq 4) \wedge (a2 \neq 3))$$

drugače pa je razred 0.

V podatkovno množico smo vstavili primere, ki so nekonsistentni in neskladni s pravili. Vstavili smo primere, ki so prikazani v tabeli 4.3.

id	Razred	a1	a2	a3	a4	a5	a6
433	1	3	2	1	2	2	1
434	0	1	3	2	3	1	2
435	1	1	2	1	1	3	1
436	1	1	3	1	1	4	1

Tabela 4.3: Tabela prikazuje vstavljene primere.

4.1.3 B2B (Business-to-Business)

Anketna podatkovna množica B2B vsebuje 11 atributov in razred. Vrednosti atributov in razreda so cela števila od 1 do 5. Podatkovna zbirka je majhna, saj vsebuje samo 44 primerov, in ima nekaj manjkajočih vrednosti. Ker podatkov ne poznamo, je težje določiti konsistentnost primerov. Tabela 4.4 prikazuje dodatno vstavljene primere za testiranje. Iz stolpcev tabele 4.4 je razviden vzorec, kako smo sestavili te primere.

id	45	46	47	48
PERF_PRO	1	3	1	5
PERF_MEE	1	3	2	4
PERF_DEL	1	3	3	3
PERF_ON	1	3	4	2
PERF_LEA	1	3	5	1
PERF_EAS	1	3	1	1
PERF_TEC	1	3	2	2
PERF_RES	1	3	3	3
PERF_KNO	1	3	4	4
PERF_BRA	1	3	5	5
PERF_ADM	1	3	1	5
GS	1	3	2	4

Tabela 4.4: Dodatno vstavljeni primeri v podatke B2B.

4.1.4 B2C (Business-to-Customer)

Podatkovna zbirka B2C vsebuje 64 atributov in razred. Vrednosti vseh atributov in razreda so cela števila od 1 do 10. Podatkovna zbirka vsebuje 4032 primerov (anketirancev). Vsebuje tudi manjkajoče vrednosti. Stolpci v tabeli 4.5 prikazujejo vzorec dodanih primerov. Vzorec je podoben kot pri množici B2C.

id	4033	4034	4035	4036
Q25A2	10	3	1	10
Q25A3	10	3	2	9
Q25A4	10	3	3	8
Q25A5	10	3	4	7
Q25A6	10	3	5	6
Q25A6	10	3	6	5
Q25A6	10	3	7	4
Q25A6	10	3	8	3
Q25A6	10	3	9	2
Q25A6	10	3	10	1
Q25A6	10	3	1	1
⋮	⋮	⋮	⋮	⋮
Q19	10	3	5	6

Tabela 4.5: Dodatno vstavljeni primeri v podatke BCB.

4.1.5 DPS

Podatkovna zbirka DPS vsebuje 7 atributov in razred. Vrednost razreda (zadovoljstvo uporabnikov) so cela števila od 1 do 10, vrednosti vseh ostalih atributov pa so cela števila od 1 do 5. Zbirka vsebuje 265 anketirancev in ne vsebuje manjkajočih vrednosti. To množico smo testirali brez dodanih primerov.

4.1.6 Hearing aid(slušni aparati)

Podatkovna zbirka **hearing aid** vsebuje odgovore anket o zadovoljstvu s slušnimi aparati. Zbirka vsebuje 6 atributov, oziroma vprašanj, in 230 anketirancev. Vrednosti atributov so cela števila od 1 do 5. Množica vsebuje manjkajoče vrednosti. To množico smo testirali brez dodanih primerov.

4.2 Testiranje in rezultati

4.2.1 Monk

Podatkovna zbirka **monk**, opisana v poglavju 4.1.1, je primerna za testiranje metod, saj podatki ne vsebujejo manjkajočih vrednosti in šuma. Modeli so bili zgrajeni z naključnimi drevesi. Ko v podatke vstavimo primere, ki niso konsistentni (glej tabelo 4.3), dobimo zanje nizke končne ocene.

Najboljše rezultate dobimo z Brierjevo oceno, informacijsko vsebino in z verjetnostmi, saj so dodani primeri ocenjeni najslabše in rezultati imajo majhno varianco. Najslabše rezultate smo dobili z klasifikacijsko točnostjo in metodami rangiranja, ki dodane primere ne razpoznajo in imajo večjo varianco kot ostale metode. Končne ocene metod so prikazane v tabeli 4.6, končne ocene metod rangiranja pa v tabeli 4.7.

Brier		Verjetnosti		Info.		Klas. točnost	
pozicija	primer	pozicija	primer	pozicija	primer	pozicija	primer
1	**433**	1	**433**	1	**433**	1	**433**
2	**435**	2	**435**	2	**435**	2	1
3	**436**	3	**436**	3	**436**	3	9
4	171	4	53	4	53	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	230	**434**
47	**434**	121	**434**	259	**434**	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	259	**436**
436	36	436	36	436	36	⋮	⋮
						294	**435**
						⋮	⋮
						436	213

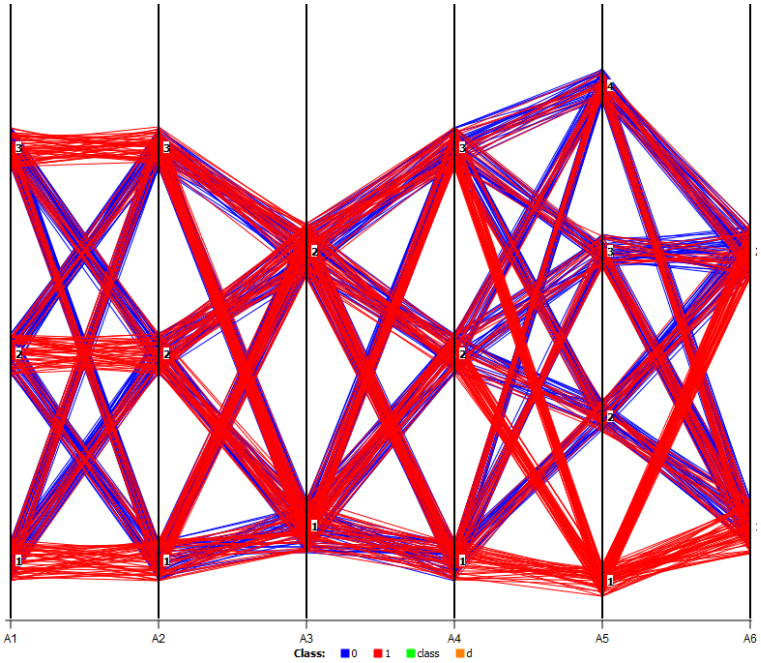
Tabela 4.6: Tabele prikazuje ocene napovedljivosti primerov množice Monk z uporabo Brierjeve ocene, verjetnostmi, informacijsko vsebino ter klasifikacijsko točnostjo.

Rang Brier		Rang verjetnosti		Rang info.		Rang klas. točnost	
pozicija	primer	pozicija	primer	pozicija	primer	pozicija	primer
1	112	1	112	1	159	1	366
2	159	2	159	2	193	2	135
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
9	**433**	24	**433**	22	**433**	212	**436**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
19	**436**	69	**436**	64	**436**	215	**433**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
53	**435**	103	**435**	104	**435**	222	**434**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
430	**434**	435	**434**	435	**434**	244	**435**
⋮	⋮	436	142	436	142	⋮	⋮
436	142					436	211

Tabela 4.7: Tabele prikazuje range primerov množice Monk.

Dodane primere lahko izrišemo z `parallel coordinates plot` v programu `Orange`. Slika 4.2 prikazuje vse primere iz podatkovne množice. Vsaka črta predstavlja en primer, horizontalne črte predstavljajo atribute, barve črt pa predstavljajo razred.

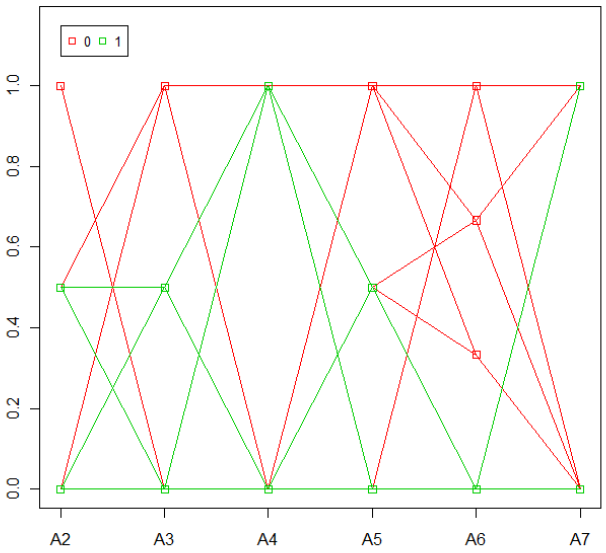
Iz slike 4.2 je možno razbrati vzorec, ki ga imajo primeri z določeno vrednostjo napovednega atributa. Opazno je npr., da imajo primeri rdeče barve (razred 1), atribut *a1* in *a2* večinoma enake vrednosti. Prav tako je opazno, da imajo vsi primeri, ki imajo atribut A5 enak 1, razred 1. Tako je možno razbrati, kakšen vzorec imajo konsistentni in najbolj pogosti primeri.



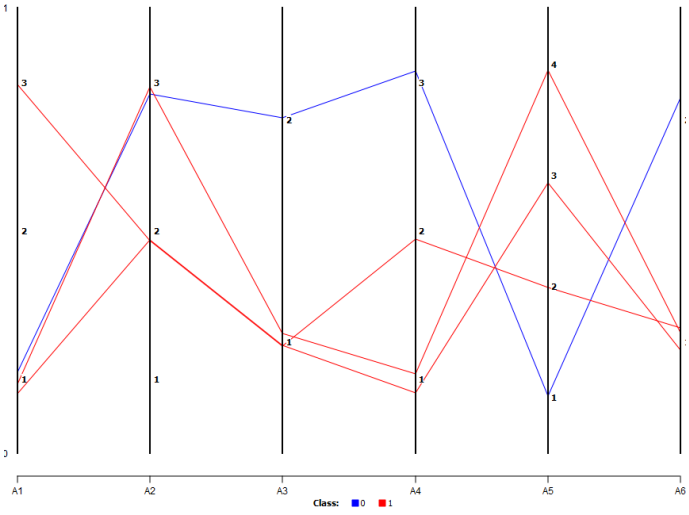
Slika 4.2: Vizualizacija primerov podatkovne množice monk.

To lahko razberemo tudi iz slike 4.3, ki prikazuje najbolj tipične primere v podatkih. Tipični primeri so tisti, ki se najbolj pogosto pojavljajo v podatkih. Zelena barva prikazuje primere, ki imajo razred 1, rdeča pa primere, ki imajo razred 0.

Slika 4.4 prikazuje dodane primere. Če primerjamo sliko 4.4 s sliko 4.2 ali s 4.3 vidimo, da imajo primeri nekaj nekonsistentnosti. Npr., primer iz slike 4.4, ki ima vrednost razred 0 (modra barva), ima atribut A5 vrednost 1. Iz slike 4.2 in 4.3 takega vzorca ne opazimo. To kaže na nekonsistentnost primera.



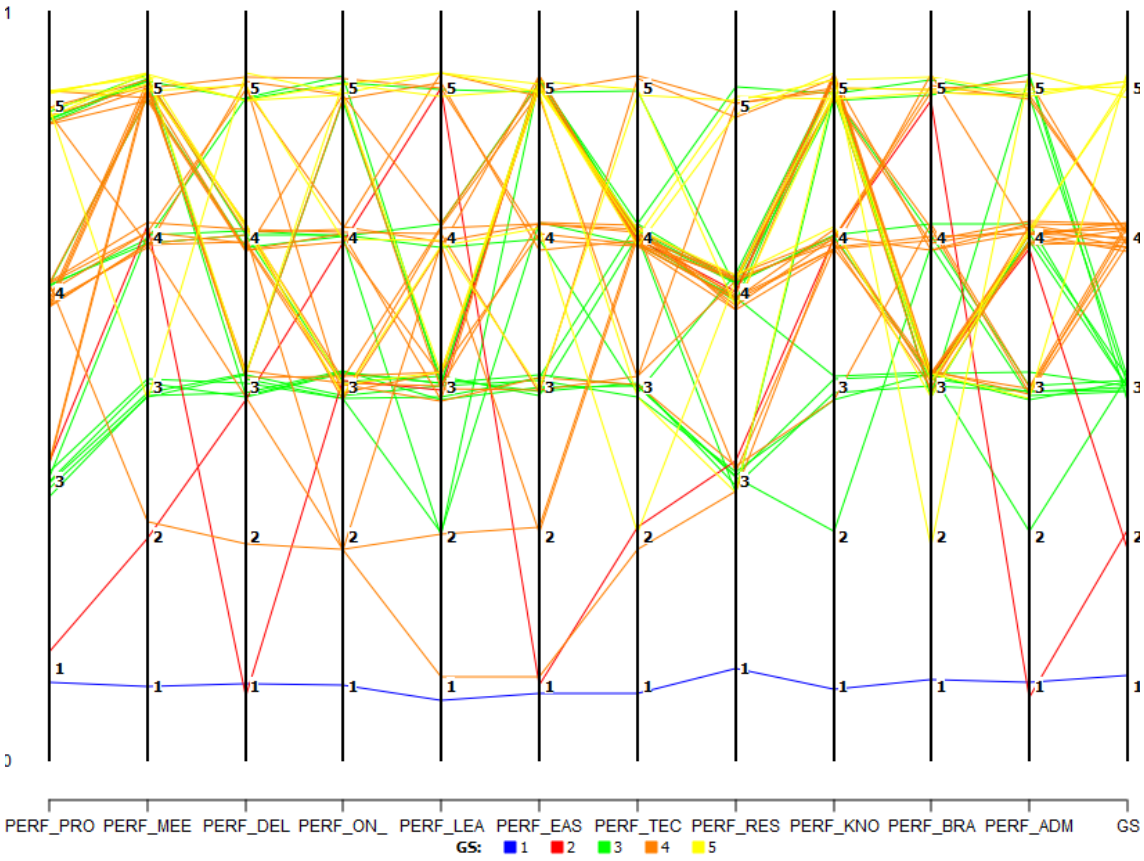
Slika 4.3: Vzorec najbolj tipičnih primerov.



Slika 4.4: Dodani nekonsistentni primeri.

4.2.2 B2B (Business-to-Business)

Zbirka **B2B** vsebuje malo primerov in ima manjkajoče vrednosti. Z metodo najbližjih sosedov smo zamenjali vse manjkajoče vrednost. Modeli so bili zgrajeni z naključnimi drevesi. Slika 4.5 prikazuje vse primere pobarvane z atributom GS (razred). Iz slike je razvidno, da imajo nekateri primeri, predvsem zeleni, zelo podobne odgovore. Opazimo tudi, da so odgovori (vrednosti) 1 in 2 zelo redki.



Slika 4.5: Primeri v zbirki B2B

Končne ocene naših metod na tej množici podatkov so nestabilne zaradi velike variance, ki je posledica majhne količine primerov. Za stabilnejše ocene smo modele sestavili večkrat in njihove končne ocene povprečili. Tabela 4.8 prikazuje ocene končnih povprečenih ocen metod, tabela 4.9 pa ocene metod rangiranj.

Na teh podatkih so ocene rangiranj podobne ocenam ostalih metod. Najboljše rezultate vrnejo Brierjeva ocena in verjetnosti, najslabše pa informacijska vsebina. Ker je primerov zelo malo je težje določiti konsistentnost, saj z dodajanjem primerov povečamo šum. Tako lahko vsak dodani primer spremeni končne ocene.

Brier	
pozicija	primer
1	**45**
2	**47**
3	33
4	**48**
⋮	⋮
48	**46**

Verjetnosti	
pozicija	primer
1	**45**
2	**47**
3	**48**
4	33
⋮	⋮
48	**46**

Info. vsebina	
pozicija	primer
1	7
2	33
⋮	⋮
15	**47**
16	**48**
⋮	⋮
35	**45**
⋮	⋮
48	**46**

Klas. točnost	
pozicija	primer
1	**45**
2	**47**
3	13
⋮	⋮
7	**48**
⋮	⋮
48	**46**

Tabela 4.8: Tabela prikazuje ocene metod množice B2B.

Rang Brier	
pozicija	primer
1	**45**
2	**47**
3	33
4	13
5	**48**
⋮	⋮
48	**46**

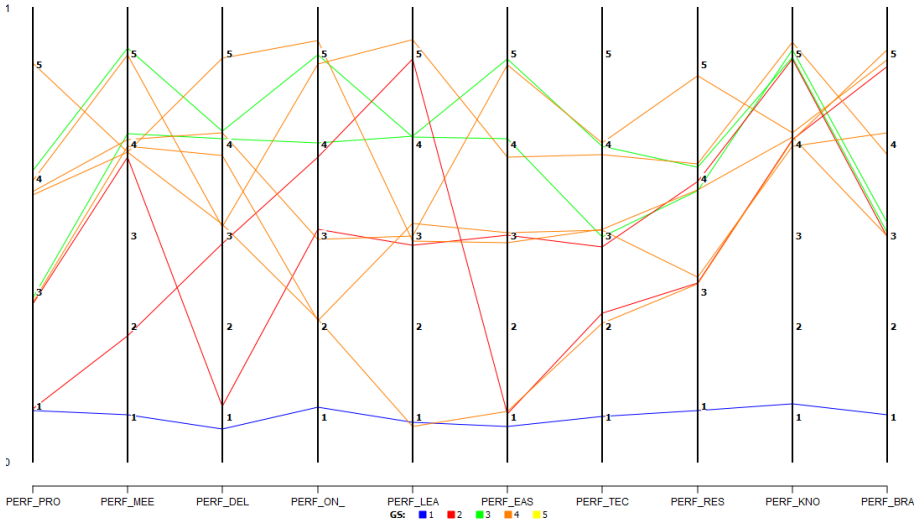
Rang verjetnosti	
pozicija	primer
1	**45**
2	**47**
3	**48**
4	33
⋮	⋮
48	**46**

Rang info.	
pozicija	primer
1	7
2	13
⋮	⋮
16	**47**
17	**48**
⋮	⋮
29	**45**
⋮	⋮
48	**46**

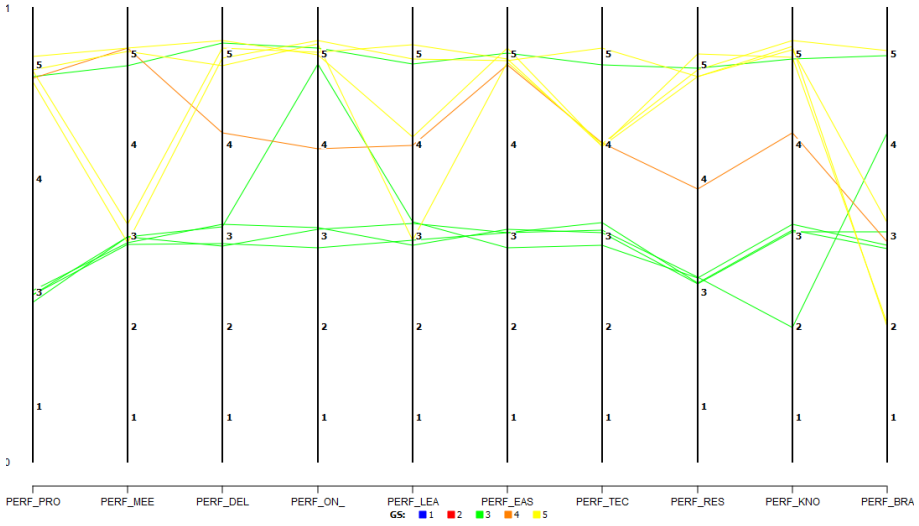
Rang klas. točnost	
pozicija	primer
1	**47**
2	**45**
3	11
⋮	⋮
11	**48**
⋮	⋮
44	**46**
⋮	⋮
48	3

Tabela 4.9: Tabela prikazuje ocene metod rangiranja za množico B2B.

Sliki 4.6 in 4.7 prikazujeta 10 najslabše, oziroma najboljše ocenjenih primerov. Primere smo izbrali iz rezultatov, ki nam jih vrne Brierjeva ocena in metoda verjetnosti. Opazno je, da odgovori nekonsistentnih primerov močno varirajo, medtem ko odgovori konsistentnih dosti manje. Večina konsistentnih primerov ima vrednost razreda GS 5 in 3 (rumena in modra), nekonsistentni pa imajo vrednost razreda 4 (oranžna).



Slika 4.6: 10 nekonsistentnih primerov podatkovne množice B2B.



Slika 4.7: 10 konsistentnih primerov podatkovne množice B2B.

4.2.3 B2C (Business-to-Customer)

Zbirka **B2C** vsebuje veliko primerov in mnogo manjkajočih vrednosti. Manjkajoče vrednosti smo zamenjali modusom, saj je ta metoda na velikih podatkovnih množicah hitrejša kot metoda najbližjih sosedov (knn). Na teh podatkih je opazno, da je naš pristop računsko zahteven. Sestavili smo modele z uporabo naključnih dreves in naivnega Bayesa. Tabela 4.10 prikazuje čas procesiranja podatkov z različnimi vhodnimi podatki.

# atributov	# primerov	k-preč. prev.	metoda	čas delovanja
65	4032	2	nakljucna drevesa	5340s
65	4032	3	naivni bayes	771s
32	4032	3	nakljucna drevesa	3540s
32	4032	3	naivni bayes	432s
10	4032	3	nakljucna drevesa	1473s
10	4032	10	naivni bayes	53s
10	2016	3	nakljucna drevesa	744s
10	2016	10	naivni bayes	30s
10	1008	3	nakljucna drevesa	386s

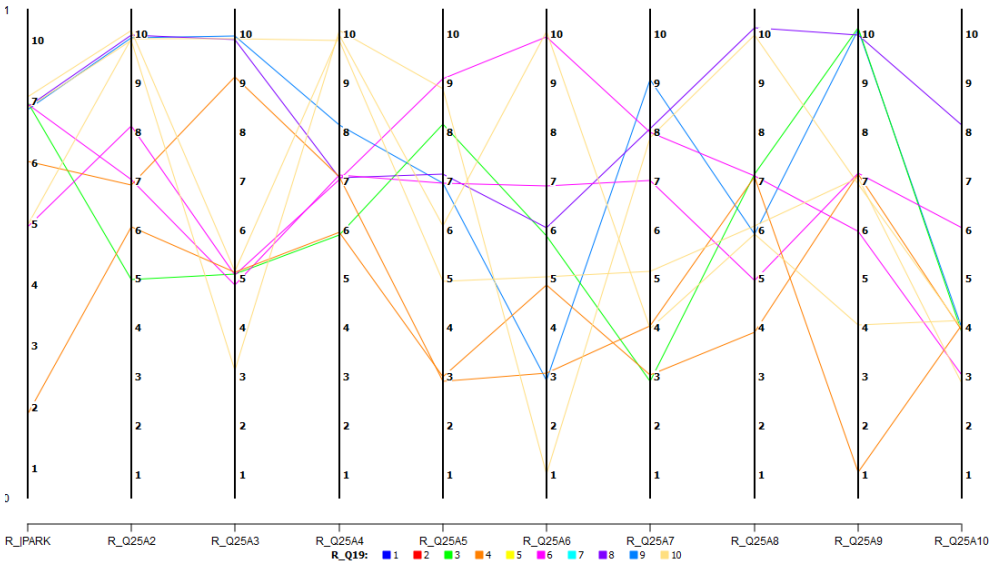
Tabela 4.10: Časi delovanja metode z različnimi parametri.

Tabela 4.11 prikazuje končne ocene, ki so bile zgrajene z naivnim Bayesovim klasifikatorjem. Zanesljivost teh ocen je v tem primeru vprašljiva, saj nimamo informacij o atributih (vprašanj) te ankete. Zaradi velike količine podatkov je izrisovanje vseh primerov nesmiselno, zato slika 4.8 prikazuje samo primere z najslabšo oceno (prvih 10), slika 4.9 pa primere z najboljšo oceno. Iz slik lahko razberemo, da so odgovori nekonsistentnih zelo razpršeni, medtem ko so si odgovori konsistentnih več ali manj podobni.

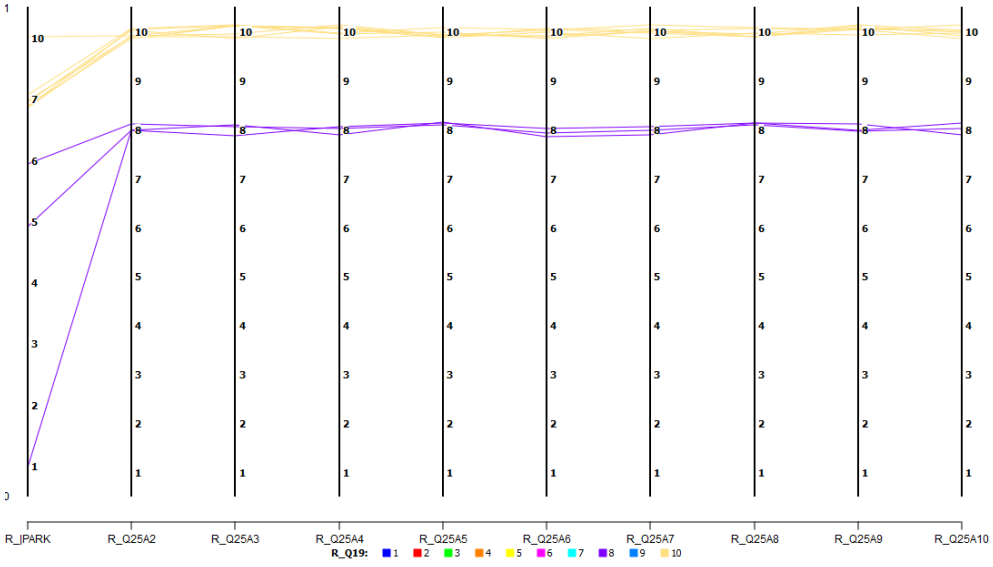
pozicija	primer	ocena
1	212	0.004
2	1253	0.004
3	3349	0.006
⋮	⋮	⋮
4031	3831	6.152
4032	3976	6.152

pozicija	primer	ocena
1	4006	1.834
2	1603	1.821
3	2050	1.808
4	36	2.686
⋮	⋮	⋮
4031	3831	0.573
4032	3976	0.573

Tabela 4.11: Leva tabela vsebuje ocene konsistentnosti z verjetnostmi, desna tabela pa ocene z Brierjevo ocena na anketi B2C.



Slika 4.8: Prikaz nekonsistentnih primerov v anketi B2C



Slika 4.9: Prikaz konsistentnih primerov v anketi B2C

Z uporabo obratnega rangiranja smo generirali primere, ki smo jih uporabili na zgrajenih primerih. Generirani primeri so prikazani v tabeli 4.12. Zaradi velikega števila atributov je prikazanih samo nekaj atributov.

ID	4036	4037	4038	4039	4040
PARK	1	1	7	1	2
Q25A2	4	8	9	9	8
Q25A3	6	8	4	7	6
Q25A4	7	9	8	7	6
Q25A5	10	9	7	6	8
Q25A6	7	8	9	7	8
Q25A7	6	10	5	8	6
Q25A8	7	9	4	5	7
⋮	⋮	⋮	⋮	⋮	⋮
Q25J64	7	1	8	7	9
C	4	8	6	7	1

Tabela 4.12: 5 generiranih primerov z uporabo obratnega ranga za podatke B2C.

Tabeli 4.13 in 4.14 prikazujeta rezultate vseh ocenjevalnih metod. Razen metode ranga verjetnosti, rezultati ostalih metod niso najboljši. Zaradi velikosti podatkov in časovne zahtevnosti ne moremo povprečiti rezultatov, posledica tega pa je večja varianca (razprašenost) ocen. Najslabše rezultate vrneta informacijska vsebina in rang klasifikacijske točnosti, najboljše pa rang verjetnosti, ki identificira vse generirane primere.

Brier		Verjetnosti		Info. vsebina		Klas. točnost	
pozicija	primer	pozicija	primer	pozicija	primer	pozicija	primer
1	3195	1	2595	1	1855	1	2595
2	1002	2	1002	2	505	2	1002
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
155	**4037**	24	**4037**	184	**4037**	16	**4037**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
195	**4036**	49	**4036**	1331	**4040**	35	**4036**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
253	**4040**	56	**4040**	1625	**4036**	59	**4040**
⋮	⋮	⋮	⋮	⋮	⋮	60	**4038**
256	**4039**	79	**4038**	2234	**4039**	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	118	**4039**
284	**4038**	102	**4038**	2326	**4038**	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	4039	850
4039	850	4039	850	4039	4032	4040	3831
4040	3831	4040	3831	4040	3831		

Tabela 4.13: Tabela prikazuje ocene metod množice B2C.

Brierjev rang	
pozicija	primer
1	1017
2	1512
⋮	⋮
229	**4037**
⋮	⋮
291	**4036**
⋮	⋮
321	**4039**
⋮	⋮
337	**4040**
⋮	⋮
531	**4038**
⋮	⋮
4039	4032
4040	3831

Rang verjetnosti	
pozicija	primer
1	**4037**
2	**4040**
3	**4036**
4	**4039**
5	3195
6	**4038**
⋮	⋮
4039	4032
4040	3831

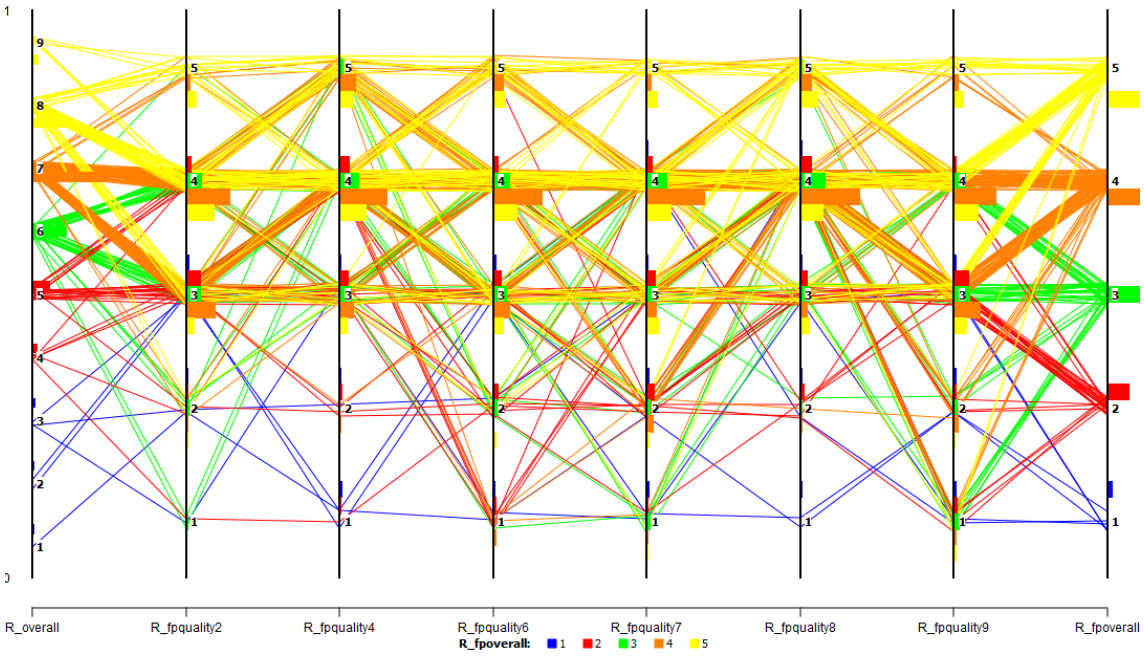
Rang info. vsebine	
pozicija	primer
1	1855
2	1463
⋮	⋮
93	**4037**
⋮	⋮
546	**4040**
⋮	⋮
582	**4036**
⋮	⋮
786	**4039**
⋮	⋮
896	**4038**
⋮	⋮
4039	3831
4040	2762

Rang klas. točnosti	
pozicija	primer
1	3890
2	1002
⋮	⋮
2517	**4037**
⋮	⋮
2584	**4036**
⋮	⋮
2614	**4038**
2647	**4040**
⋮	⋮
2779	**4039**
⋮	⋮
4039	1587
4040	3477

Tabela 4.14: Tabela prikazuje ocene rangiranih metod množice B2C.

4.2.4 DPS

V podatkovni množici **DPS** ni bilo dodanih nobenih novih primerov in je bilo testiranje opravljeno na neobdelanih podatkih. Ocene smo povprečili, da bi zmanjšali varianco in bi tako dobili boljšo predstavo podatkov. Slika 4.10 prikazuje vse primere množice DPS pobarvane z atributom fpoverall.



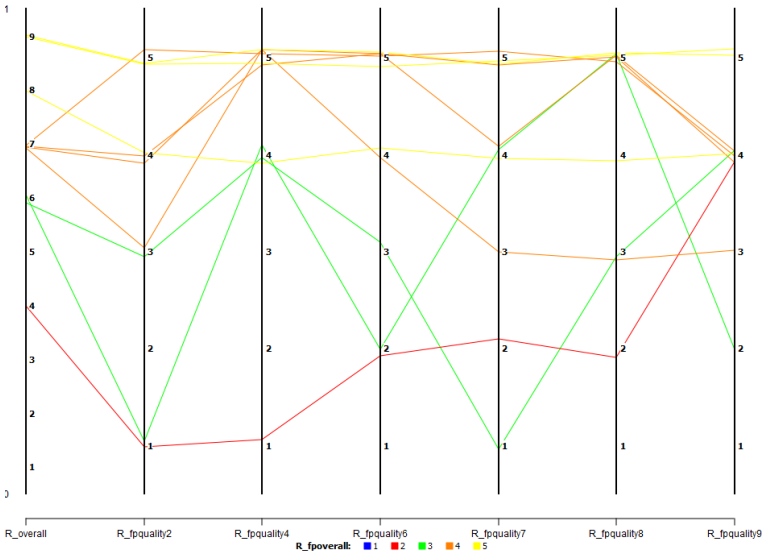
Slika 4.10: Vizualizacija primerov podatkovne množice DPS.

Tabela 4.15 prikazuje 10 najslabše ocenjenih primerov. Levi del tabele prikazuje ocene rangiranja verjetnosti, desni pa rangiranja Brierjeve ocene. Ocene niso normalizirane in imajo visoke vrednosti. Ti primeri so izrisani na sliki 4.11 in 4.12. Slika 4.11 prikazuje primere z najslabšo oceno pri rangiranju verjetnosti, slika 4.12 pa prikazuje primere z najslabšo oceno pri rangiranju Brierjeve ocene. Iz slike 4.11 opazimo, da ima primer s $fpoverall = 2$ (rdeče barve) vrednosti atributov, ki so netipične za te primere. Prav tako imata netipične vrednosti primera z fpoverall 3. To nakazuje na nekonsistentnost teh primerov, saj so vzorci iz slike 4.11 drugačni od tistih na sliki 4.10. Iz

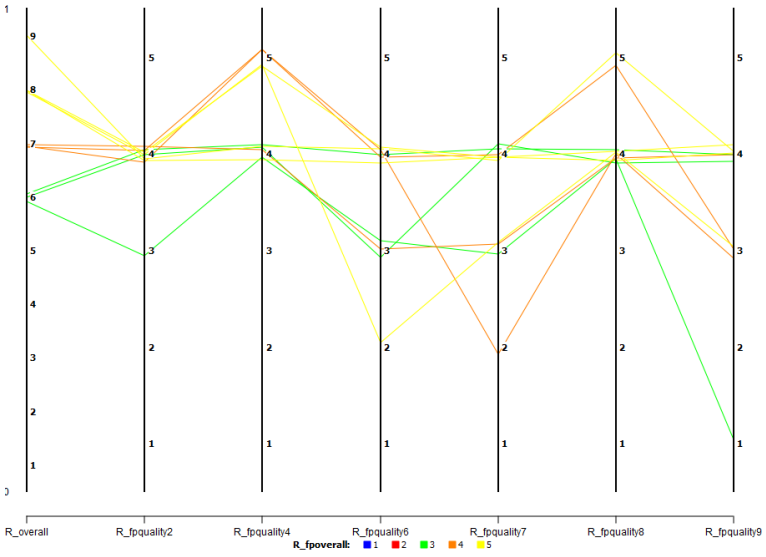
Rang verjetnosti		Rang Brierja	
primer	ocena	primer	ocena
50	675.625	257	1528.75
265	710.875	52	1374.625
259	717.125	128	1353.125
105	725.875	45	1344.875
8	727.625	39	1335.25
182	773.5	102	1332.5
147	782.5	260	1324.625
257	783.25	126	1323.5
62	797.5	250	1298.5
136	805.875	230	1291.75

Tabela 4.15: Ocene prvih 10 najslabše ocenjenih primerov v množici DPS.

slike 4.12 so nekonsistentni primeri manj razvidni, saj imajo primeri podobne vzorce kot tisti na sliki 4.10. Razlog za to je v metodi ocenjevanja, ki te primere oceni kot nekonsistentne. Ti primeri so lahko nekonsistenti, vendar jih je vizualno težje identificirati kot tiste, ki so prikazani na sliki 4.11.



Slika 4.11: Vizualizacija 10 najslabše ocenjenih primerov z rangiranjem verjetnosti za podatke DPS.

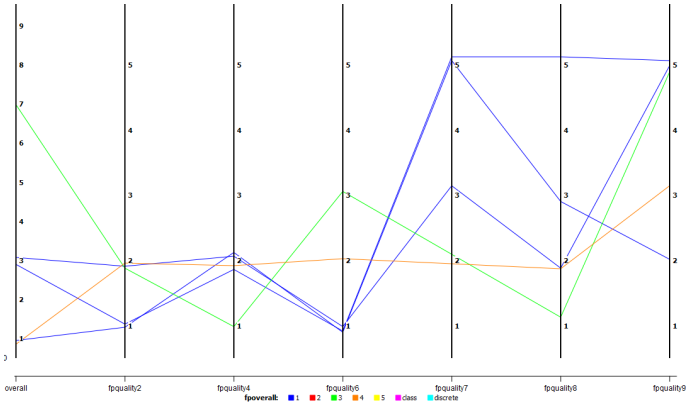


Slika 4.12: Vizualizacija 10 najslabše ocenjenih primerov z rangiranjem Brierjeve ocene za podatke DPS.

Kot pri množici B2C smo tudi tukaj generirali primere z obratnim rangiranjem. Tabela 4.16 prikazuje generirane primere, ki smo testirali na že sestavljenih napovednih modelih. Slika 4.13 prikazuje generirane primere pobarvane po atributu fpoverall. Tabeli 4.17 in 4.18 prikazujeta ocene konsistentnosti primerov. Najboljše rezultate v tej množici vrnejo verjetnosti, rang verjetnosti in Brierjeva ocena. Prav tako dobre rezultate vrne klasifikacijska točnost, saj vse generirane primere oceni z nizkimi ocenami. Slabše ocene vrne informacijska vsebina.

ID	266	267	268	269	270
overall	7	3	1	1	3
fpquality2	2	2	2	1	1
fpquality4	1	2	2	2	2
fpquality6	3	1	2	1	1
fpquality7	2	3	2	5	5
fpquality8	1	2	2	3	5
fpquality9	5	5	3	2	5
fpoverall	3	1	4	1	1

Tabela 4.16: 5 generiranih primerov z uporabo obratnega ranga za podatke DPS.



Slika 4.13: Vizualizacija generiranih primerov iz podatkovne množice DPS.

Brier	
pozicija	primer
1	**266**
2	**270**
3	**269**
4	1
5	8
6	224
7	**267**
⋮	⋮
14	**268**
⋮	⋮
269	239
270	205

Verjetnosti	
pozicija	primer
1	**270**
2	**266**
3	1
4	**269**
5	8
6	**267**
7	5
8	6
9	**268**
⋮	⋮
269	205
270	239

Info. vsebina	
pozicija	primer
1	146
2	130
⋮	⋮
186	**266**
⋮	⋮
243	**270**
⋮	⋮
249	**268**
⋮	⋮
257	**269**
⋮	⋮
264	**267**
⋮	⋮
269	265
270	259

Klas. točnost	
pozicija	primer
1	1
2	**266**
⋮	⋮
7	**267**
8	**270**
9	**268**
10	**269**
⋮	⋮
269	239
270	258

Tabela 4.17: Tabela prikazuje ocene metod množice DPS.

Rang Brier	
pozicija	primer
1	**270**
2	**269**
3	8
4	**267**
5	1
6	224
7	**266**
⋮	⋮
21	**268**
⋮	⋮
269	209
270	244

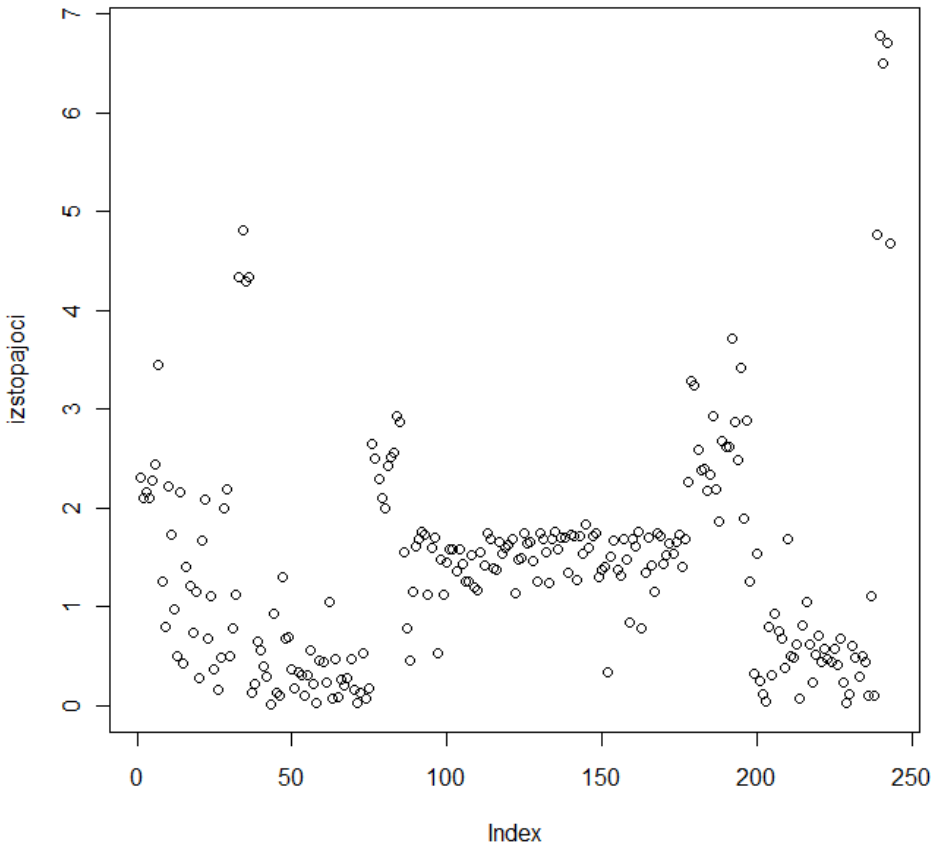
Rang verjetnosti	
pozicija	primer
1	**270**
2	**269**
3	8
4	**267**
5	**266**
⋮	⋮
18	**268**
⋮	⋮
269	209
270	251

Rang info.	
pozicija	primer
1	119
2	147
⋮	⋮
173	**266**
⋮	⋮
207	**270**
⋮	⋮
245	**268**
⋮	⋮
250	**267**
251	**269**
⋮	⋮
269	14
270	259

Rang klas. točnost	
pozicija	primer
1	147
2	264
⋮	⋮
87	**266**
⋮	⋮
122	**268**
⋮	⋮
146	**269**
⋮	⋮
157	**267**
⋮	⋮
160	**270**
⋮	⋮
269	88
270	108

Tabela 4.18: Tabela prikazuje ocene rangiranih metod množice DPS.

Slika 4.14 prikazuje primere iz podatovne množice DPS. Metoda rfOutliers vrne vrednosti za vse primere iz učne množice modela. Ker smo uporabili 10 kratno prečno preverjanje, nam rfOutliers vrne vrednosti za 90% primerov (za to množico, 243 od 270 primerov). Izbrali smo napovedni model, ki je bil zgrajen iz učne množice, ki vsebuje dodatno generirane primere. X-os so vsi primeri od 1 do 243, Y-os pa so vrednosti teh primerov, ki jih vrne metoda *rfOutliers*. Povprečje teh vrednosti je 1.4. Vsi primeri, ki močno odstopajo od povprečja so lahko nekonsistentni primeri.



Slika 4.14: Vizualizacija primerov za podatke DPS.

Iz slike lahko na grobo ocenimo delež nekonsistentnih primerov. Dodatno generirani primeri so označeni od 238 do 243, namesto od 265 do 270. Ti primeri so prikazani v desnem kotu slike. Opazno je, da je metoda rfOutliers ocenila dodane primere kot izjeme, oz. izstopajoče primere.

Tabela 4.19 prikazuje razvrstitev najbolj izstopajocih primerov metode rfOutliers. V oklepaju prikazujemo originalno ime primera. Opazimo, da je metoda najvišje ocenila novo generirane primere, kar pomeni, da so po vsej verjetnosti to izstopajoči primeri. Potrebno je omeniti, da npr. primera 71 in 43 sta prav tako lahko izjemi, saj sta dosti oddaljena od povprečja ocen (v tem primeru 1.4).

rfOutliers

pozicija	ocena	primer
1	6.784	**240** (267)
2	6.705	**242** (269)
3	6.498	**241** (268)
4	4.809	34
5	4.758	**239** (266)
6	4.668	**243** (270)
7	4.331	36
⋮	⋮	
242	0.022	71
243	0.007	43

Tabela 4.19: Tabela prikazuje izstopajoce primere za podatke DPS, ki nam jo vrne metoda rfOutliers.

4.2.5 Hearing aid

Hearing aid je podatkovna množica, ki vsebuje odgovore ankete o slušnih aparatih. Podatkovna množica vsebuje malo manjkajočih vrednosti, zato smo uporabili metodo najbližjih sosedov za njihovo zamenjavo. V podatkovno zbirko nismo vstavili novih primerov. Metoda nam je vrnila ocene vseh primerov. Ocene smo povprečili, da bi dobili manjšo varianco. Tabela 4.20 in 4.21 nam prikazuje 10 najslabše ocenjenih primerov z različnimi metodami ocenjevanja. Ocene niso normalizirane.

Opazimo, da je več primerov (npr. 74, 41, 69, itd.) v vseh metodah ocenjenih slabo, kar nakazuje na nekonsistenstost teh primerov. Če primerjamo razvrstitev primerov na tabeli 4.20 in 4.21 opazimo, da so si zelo podobne.

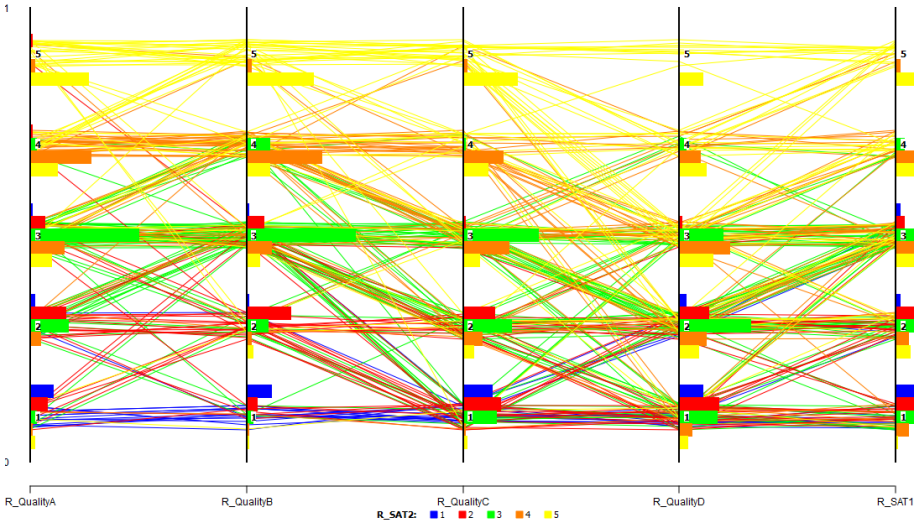
Brierjeva ocena		Verjetnosti		Info. vsebina		Klas. Točnost	
primer	ocena	primer	ocena	primer	ocena	primer	ocena
40	1.817	74	0.846	74	-1.224	65	0
69	1.735	41	0.945	41	-0.364	74	0
74	1.703	69	1.021	87	0.16	175	0
15	1.661	40	1.187	114	0.381	41	0.2
87	1.659	149	1.201	144	0.388	69	0.2
42	1.606	61	1.218	159	0.424	42	0.4
159	1.602	186	1.26	69	0.448	61	0.4
41	1.594	185	1.316	65	0.603	123	0.4
30	1.591	175	1.337	175	0.648	199	0.4
76	1.589	199	1.363	61	0.784	168	0.6

Tabela 4.20: Ocene prvih 10 najslabše ocenjenih primerov v množici Hearing aid.

Brierjev rang		Rang verjetnosti		Rang info. vsebine		Rang Klas. Točnost	
primer	ocena	primer	ocena	primer	ocena	primer	ocena
40	1206.2	74	203.6	74	169.4	65	185.6
69	1197	41	237.4	41	235	83	220.0
74	1196	69	237.8	87	256.8	61	223.0
87	1125	40	275.4	69	286.6	69	231.0
42	1125	149	291.2	114	286.6	74	236.4
41	1095.2	61	291.8	159	286.6	41	239.8
30	1076.8	185	305	144	290.8	175	243.4
31	1066	175	308.4	175	294.6	42	261.4
36	1065	186	310.2	61	309.6	40	307.4
15	1062.8	42	320.2	49	315.2	139	317.4

Tabela 4.21: Ocene prvih 10 najslabše ocenjenih primerov z rangiranjem v množici Hearing aid.

Slika 4.15 prikazuje vse primere pobarvane z atributom SAT2. Iz slike lahko razberemo vzorce in naravo podatkov. V te podatke nismo vstavili novih primerov, da ne bi vplivali na napovedne modele in končne rezultate. Generirali smo nekonsistentne primere, ki smo jih uporabili na že sestavljenih modelih. To dosežemo tako, da generiramo primere z vrednostmi, ki imajo obratno verjetnostno distribucijo vrednosti kot normalni primeri. Poskusili smo, kako bi model ocenil take primere.

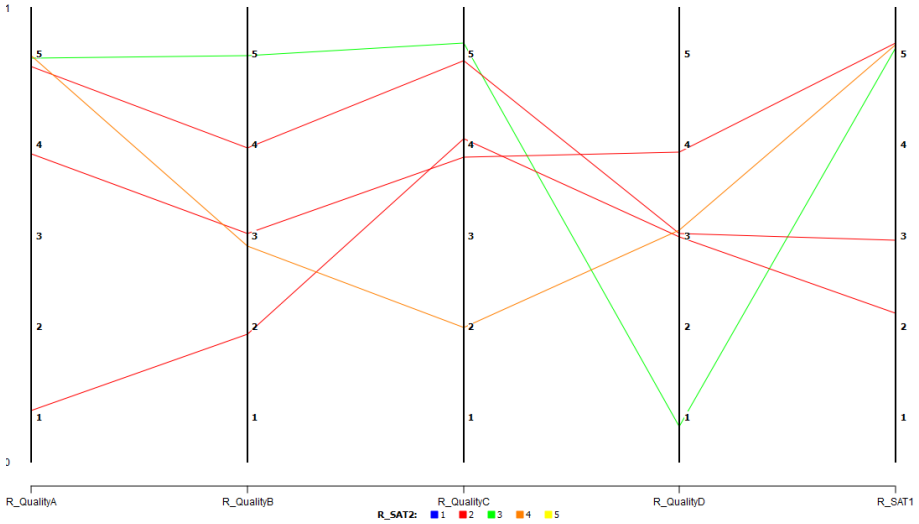


Slika 4.15: Vizualizacija vseh primerov iz podatkovne množice hearing aid.

ID	Q_A	Q_B	Q_C	Q_D	SAT1	SAT2
231	5	3	2	3	5	4
232	1	2	4	3	2	2
233	5	4	5	3	3	2
234	5	5	5	1	5	3
235	4	3	4	4	5	2

Tabela 4.22: 5 novo generiranih primerov z uporabo obratnega ranga za množico hearing aid.

Primeri, ki so bili generirani in njihove ocene so prikazani v tabeli 4.22. Slika 4.16 prikazuje generirane primere. Ti primeri se razlikujejo od vzorcev na sliki 4.15. Model je uspešno prepoznal novo generirane primere kot nekonsistentne (tabela 4.23 in 4.24). Primer 232 je od vseh generiranih primerov najbolj napovedljiv, oz. konsistenten, medtem ko je primer 234 najslabše uvrščen. Najboljše uvrstitve ima rangiranje Brierja in rangiranje verjetnosti, najslabše pa rang klasifikacijske točnosti.



Slika 4.16: Vizualizacija generiranih primerov za množico Hearing aid.

Brier	
pozicija	primer
1	69
2	40
3	74
4	**234**
5	**233**
⋮	⋮
8	**235**
⋮	⋮
13	**231**
⋮	⋮
30	**232**
⋮	⋮
234	5
235	6

Verjetnosti	
pozicija	primer
1	**234**
2	**235**
3	74
⋮	⋮
6	**233**
⋮	⋮
13	**231**
⋮	⋮
32	**232**
⋮	⋮
234	6
235	5

Info. vsebina	
pozicija	primer
1	74
2	**234**
3	41
4	**233**
⋮	⋮
17	**235**
⋮	⋮
22	**231**
⋮	⋮
95	**232**
⋮	⋮
234	227
235	228

Klas. točnost	
pozicija	primer
1	65
2	69
3	74
⋮	⋮
15	**233**
16	**234**
17	**235**
⋮	⋮
42	**231**
43	**232**
⋮	⋮
234	230
235	8

Tabela 4.23: Tabela prikazuje ocene metod množice Hearing aid.

Rang Brier	
pozicija	primer
1	69
2	74
3	**234**
4	40
5	**233**
6	**235**
⋮	⋮
18	**231**
⋮	⋮
32	**232**
⋮	⋮
269	6
270	5

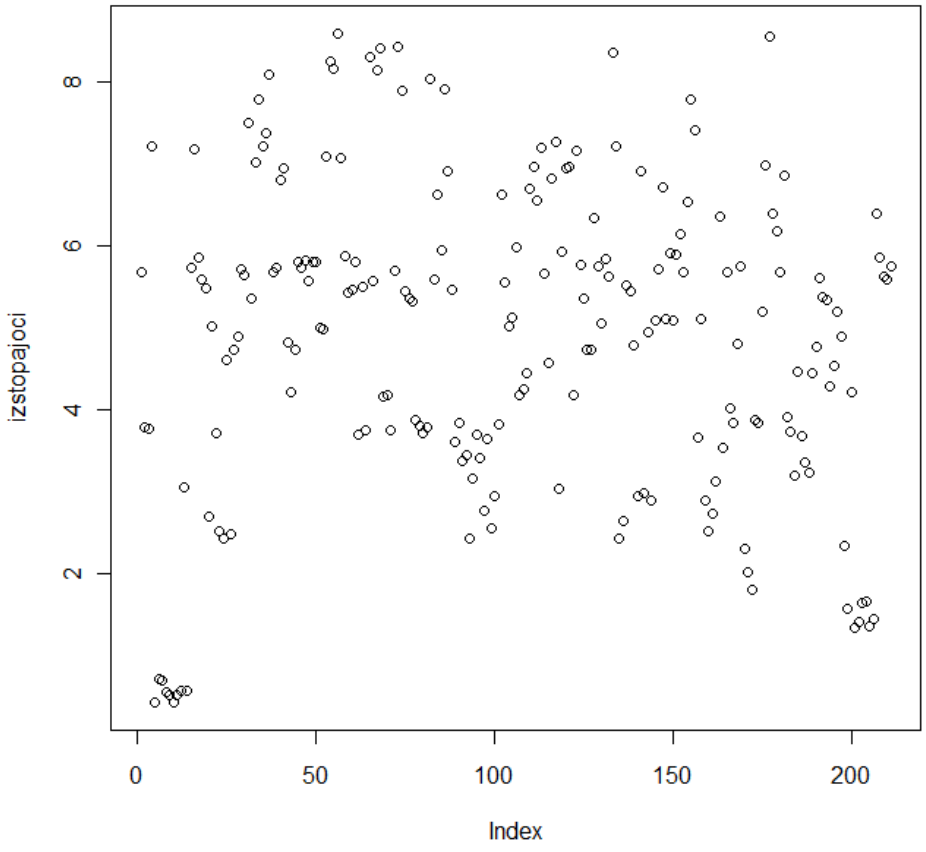
Rang verjetnosti	
pozicija	primer
1	**234**
2	**235**
3	74
4	69
5	**233**
⋮	⋮
11	**231**
⋮	⋮
32	**232**
⋮	⋮
269	5
270	6

Rang info.	
pozicija	primer
1	74
2	**234**
3	72
⋮	⋮
8	**233**
⋮	⋮
17	**231**
18	**235**
⋮	⋮
84	**232**
⋮	⋮
269	230
270	228

Rang klas. točnost	
pozicija	primer
1	168
2	65
3	175
⋮	⋮
110	**233**
⋮	⋮
118	**234**
⋮	⋮
121	**235**
⋮	⋮
159	**231**
⋮	⋮
166	**232**
⋮	⋮
269	13
270	221

Tabela 4.24: Tabela prikazuje ocene rangiranih metod množice Hearing aid.

Paket CORElearn vsebuje metodo *rfOutliers* s katero lahko detektiramo izstopajoce primere (outliers). Slika 4.17 prikazuje primere iz podatovne množice hearing aid. Zaradi 10 kratnega prečnega preverjanja, metoda rfOutliers vrne le vrednosti za 90% primerov (za to množico, 211 od 235 primerov). X-os so vsi primeri od 1 do 211, Y-os pa so vrednosti teh primerov, ki vrne metoda *rfOutliers*. Povprečje teh vrednosti je 4.8. Iz slike lahko opazimo gruče primerov, ki so oddaljene od povprečja. Ti primeri so po vsej verjetnosti izjeme.



Slika 4.17: Vizualizacija primerov za množico Hearing aid.

Iz slike lahko na grobo ocenimo delež nekonsistentnih primerov. Dodatno generirani primeri so označeni od 206 do 211, namesto od 230 do 235.

Tabela 4.25 prikazuje razvrstitev najbolj izstopajocih primerov metode rfOutliers. V oklepaju prikazujemo originalni ime primera. Metoda dodatnih primerov tokrat ni identificirala, saj so v bližini povprečja 4.8.

rfOutliers		
pozicija	ocena	primer
1	8.588	56
2	8.545	177
3	8.427	73
⋮	⋮	
125	5.574	**210** (234)
⋮	⋮	
129	5.616	**209** (233)
⋮	⋮	
146	5.749	**211** (235)
⋮	⋮	
154	5.855	**208** (232)
⋮	⋮	
167	6.391	**207** (231)
⋮	⋮	
209	0.507	9
210	0.433	5
211	0.420	10

Tabela 4.25: Izstopajoči primeri za podatke Hearing aid z metodo rfOutliers.

4.3 Časi izvajanja

Ugotovili smo, da je naša metoda časovno zahtevna. V tem razdelku bomo prikazali čase trajanja metode na podatkih.

podatki	# atributov	# primerov	čas izvajanja (sekunde)
monk	7	432	80s
B2B	12	44	30s
B2C	65	4032	6780s
DPS	8	265	88s
Hearing aid	6	230	58s

Tabela 4.26: Časi izvajanj metode na podatkovnih množicah z 10 kratnim prečnim preverjanjem.

Tabela 4.26 prikazuje parametre, ki vplivajo na hitrost naše metode. Tabela vsebuje podatkovne množice, ki smo jih uporabili ter število atributov in primerov. Na čas izvajanja vpliva tudi povprečenje, saj moramo metodo večkrat zagnati. Tabela prikazuje rezultate brez povprečenja. Za napovedne modele je uporabljen algoritem naključnih dreves.

Meritve so bile izvršene na računalniku z dvojedernim procesorjem Intel Core2 Duo p8600 s frekvenco 2.4GHz in 2GB pomnilnika.

Na čas izvajanja metode močno vpliva algoritem, ki ga uporabimo pri gradnji modelov. Zaradi dobrih lastnosti smo za gradnjo modelov uporabili naključna drevesa. Kot smo omenili v poglavju 3.1, je slabost naključnih dreves časovna zahtevnost. Ker za vsak atribut gradimo novi model, nam časovna zahtevnost metode naraste. Dodatno nam jo poveča prečno preverjanje, saj moramo za vsako množico ponovno zgraditi nove napovedne modele.

Poglavje 5

Zaključek

V okolju R smo izdelali postopek za oceno napovedljivosti, oziroma konsistentnosti anketirancev. Nekonsistentni primeri so tisti, ki imajo netipične in nenapovedljive odgovore. Manjkajoče vrednosti smo obravnavali z metodo najbližjih sosedov (knn), povprečjem, modusom in izpustom vrstice. Za vsak atribut smo sestavili napovedni model. Napovedni modeli nam vrnejo distribucijo verjetnosti za vsaki odgovor anketiranca. Vsak odgovor, oziroma njegovo distribucijo verjetnosti, smo ocenili z različnimi metodami, kot so Brier, verjetnost, informacijska vsebina, klasifikacijska točnost, Brierjevo rangiranje, rang verjetnosti, rang informacijske vsebine in rang klasifikacijske točnosti. Te ocene smo uporabili kot merilo za napovedljivost primerov. Metodo smo pognali večkrat, ocene pa smo povprečili, da bi zmanjšali varianco. Da smo dobili rezultate za vse primere, smo uporabljali prečno preverjanje.

Ugotovili smo, da uspešnost metode varira med podatkovnimi množicami. V večini primerov metoda dobro deluje. Pri podatkih, ki jih ne poznamo, smo primere izrisali in na podlagi njihovih ocen in slik ocenili njihovo nekonsistentnost. Metoda je časovno zahtevna in zato manj primerna za velike podatkovne množice.

V nadaljnem delu bi bilo potrebno naš pristop izpopolniti z novimi algoritmi za gradnjo napovednih modelov in novimi metodami ocenjevanja

konsistentnosti. Predvsem bi bilo potrebno metodo optimizirati in jo pohitriti. Prav tako bi bilo zanimivo videti drugačno vizualizacijo primerov in podatkov kot npr. radviz, kjer so primeri prikazani na krožnici na kateri so razporejeni atributi, ki otežujejo in privlačijo primere.

Literatura

[1] Jeremiah P. Banda. Nonsampling error in surveys. Technical report, United Nations Secretariat, 2003.

[2] Leo Breiman. Random forest. Technical report, Statistics Department University of California Berkeley, 2001.

[3] Leo Breiman and Adele Cutler. Random forests. `http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`.

[4] John Carroll and Ted Briscoe. Sparkle project. `http://www.ilc.cnr.it/sparkle/wp3.2/node25.htmlm`.

[5] Tomaž Curk, Janez Demšar, Qikai Xu, Gregor Leban, Uroš Petrovič, Ivan Bratko, Gad Shaulsky, and Blaž Zupan. Microarray data mining with visual programming. *Bioinformatics*, 21:396–398, February 2005.

[6] The European Virtual Organisation for Meteorological Training. The brier score – accuracy of a probability forecast. `http://www.eumetcal.org/resources/ukmeteocal/verification/www/english/msg/ver_prob_forec/uos2/uos2_ko1.htm`.

[7] Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2007.

[8] Victoria J. Hodge and Jim Austin. A survey of outlier detection methodologies. Technical report, White Rose University Consortium, 2004.

[9] Mark A. Hall Ian H. Witten, Eibr Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, third edition, 2011.

[10] Marko Robnik Šikonja Igor Kononenko. *Inteligentni sistemi*. Založba FE in FRI, 2010.

[11] Stephen Marsland. *Machine Learning: An Algorithmic Perspective*. CRC Press, 2009.

[12] Lars E. Lyberg Paul P. Biemer. *Introduction to Survey Quality*. Wiley and Sons, 2003.

[13] Toby Segaran. *Programming Collective Intelligence*. O'Reilly media, 2007.

[14] Marko R. Sikonja and Petr Savicky. Package CORElearn. `http://cran.r-project.org/web/packages/CORElearn/CORElearn.pdf`.

[15] Paul Teetor. *R Cookbook*. O'Reilly media, 2011.

[16] Terry M. Therneau and Beth Atkinson. Package rpart. `http://cran.r-project.org/web/packages/rpart/rpart.pdf`.

[17] Luis Torgo. *Data Mining with R: Learning with Case Studies*. CRC Press, 2011.

[18] Jerome Friedman Trevor Hastie, Robert Tibshirani. *The elements of statistical learning: Data mining, inference and prediction*. Springer, second edition, 2008.

[19] Arindam Banerjee Varun Chandola and Vipin Kumar. Outlier detection: A survey. Technical report, University of Minnesota, 2007.

[20] Kurt Hornik Venables, Brian Ripley and Albrecht Gebhardt. Package MASS. `http://cran.r-project.org/web/packages/MASS/MASS.pdf`.