UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

NAPOVEDOVANJE PRENEHANJA UPORABE IZBRANE MOBILNE APLIKACIJE

Ljubljana, december 2016 KATJA MIKLAVČIČ

IZJAVA O AVTORSTVU

Podpisana Katja Miklavčič, študentka Ekonomske fakultete Univerze v Ljubljani, avtorica predloženega dela z naslovom Napovedovanje prenehanja uporabe izbrane mobilne aplikacije, pripravljenega v sodelovanju s svetovalko doc. dr. Damjano Kokol Bukovšek.

IZJAVLJAM

1. da sem predloženo delo pripravila samostojno;

2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;

3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbela, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;

4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;

5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;

6. da sem pridobila vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označila;

7. da sem pri pripravi predloženega dela ravnala v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobila soglasje etične komisije;

8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;

9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;

10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne _____ Podpis študentke:_____

KAZALO

KAZALO TABEL

KAZALO SLIK

UVOD

Mobilna aplikacija je programska oprema, namenjena uporabi na pametnih mobilnih napravah (Statista, b.l.). Uporaba mobilnih aplikacij je tako razširjena, da Statista (b.l.) napoveduje, da bo leta 2017 seštevek vseh nameščenih mobilnih aplikacij znašal že 268,7 milijard. Junija 2016 je imela trgovina z mobilnimi aplikacijami Google Play na voljo 2,2 milijona različnih mobilnih aplikacij, trgovina Apple's App Store pa dva milijona.

V magistrskem delu se osredotočimo na mobilno aplikacijo Hal mBills, namenjeno brezplačnemu nakazovanju denarnih sredstev med uporabniki aplikacije ter enostavnemu plačevanju položnic. Po uspešnem prenosu mobilne aplikacije na mobilno napravo, registraciji in avtentikaciji na tako imenovano mobilno denarnico prenesemo denarna sredstva s svojega transakcijskega računa, odprtega pri katerikoli banki v Sloveniji. Z denarnimi sredstvi v mobilni denarnici prosto razpolagamo in jih lahko porabimo za nakazila prijateljem ali pa z njimi enostavno plačujemo položnice. Položnice plačujemo s slikanjem teh ali pa se naročimo na avtomatsko prejemanje položnic izbranih izdajateljev v aplikacijo, ki jih nato z le nekaj kliki tudi plačamo (Apps, 2016; Ropret, 2015).

Zaradi velike konkurence ne le na trgu mobilnih aplikacij, temveč tudi spletnih bank in fizičnih mest za plačevanje položnic ter nakazovanje denarnih sredstev je za dobičkonosnost aplikacije Hal mBills pomembno ne samo privabljati nove uporabnike, temveč tudi obdržati obstoječe.

Namen magistrskega dela je usmerjanje marketinške kampanije, namenjene ohranitvi obstoječih uporabnikov. Za stroškovno učinkovito marketinško kampanijo je potrebno vedeti, na katere uporabnike se osredotočiti, saj z izborom ciljnih uporabnikov prihranimo pri stroških izvedbe kampanije. Ciljni uporabniki so tisti, ki so tik pred prenehanjem uporabe izbrane mobilne aplikacije (Ferle, 2010).

Cilj magistrskega dela je tako napovedati verjetnost prenehanja uporabe izbrane mobilne aplikacije za posameznega uporabnika. Raziskovalno vprašanje, na katerega želimo odgovoriti, se glasi: »S kakšno verjetnostjo bo uporabnik prenehal uporabljati izbrano mobilno aplikacijo?«

Za napovedovanje verjetnosti prenehanja uporabe izbrane mobilne aplikacije lahko uporabimo različne metode podatkovnega rudarjenja. Podatkovno rudarjenje je sistematično iskanje znanja v poljubnih vhodnih podatkih. Delimo ga na nadzorovano in na nenadzorovano učenje (Zhou, 2012).

Pri nadzorovanem učenju imamo v učni množici podatkov, iz katerih se algoritem uči, poleg vhodnih znane tudi izhodne podatke, ki jih želimo napovedati. Algoritem se na učnih podatkih nauči napovedovanja, kasneje pa ga uporabimo za napovedovanje na novih

vhodnih podatkih, za katere izhodni podatki niso znani. Nadzorovano učenje se dalje deli na klasifikacijo in regresijo. Pri klasifikaciji algoritem napoveduje diskretne vrednosti, pri regresiji pa zvezne (Berry & Linoff, 2004).

Pri nenadzorovanem učenju imamo na voljo le vhodne podatke, na podlagi katerih algoritem poskuša najti vzorec. Najpogostejši primer nenadzorovanega učenja je grupiranje (angl. *clustering*), kjer algoritem brez predhodnega znanja o pomembnosti posameznih atributov razdeli podatke v različne skupine (Berry & Linoff, 2004).

Napovedovanje prenehanja uporabe izbrane mobilne aplikacije sodi v kategorijo nadzorovanega učenja s postopkom klasifikacije. Najpogosteje uporabljene metode za takšno napovedovanje so logistična regresija, nevronske mreže in odločitvena drevesa (Clemente, Ginger-Bosch, & San Matias, b.l.). Odločimo se za uporabo logistične regresije in nevronskih mrež, ki jih s spreminjanjem različnih parametrov kalibriramo tako, da sta karseda uspešni pri napovedovanju prenehanja uporabe izbrane mobilne aplikacije.

Magistrsko delo je sestavljeno iz teoretičnega in empiričnega dela. Teoretični del zavzema prvi dve poglavji, empirični del pa tretje poglavje.

V prvem poglavju predstavimo trg mobilnih aplikacij, natančneje si ogledamo trgovini Google Play in App Store ter lestvico najboljših mobilnih aplikacij. Podrobneje opišemo izbrano mobilno aplikacijo ter njej konkurenčne storitve. Izbrano mobilno aplikacijo s konkurenčnimi storitvami primerjamo glede na Google Play statistike, njihove funkcionalnosti ter ceno.

V drugem poglavju se osredotočimo na metode podatkovnega rudarjenja. Podrobneje opišemo logistično regresijo in nevronske mreže, ki jih kasneje uporabimo za modeliranje odgovora na zastavljeno raziskovalno vprašanje. Pri logistični regresiji se dodatno osredotočimo na metodo največjega verjetja in oceno algoritma, na kratko predstavimo tudi test razmerja verjetij ter Waldov test. Pri nevronskih mrežah opišemo delovanje večslojne usmerjene nevronske mreže (angl. *multilayer feedforward network*) s premikanjem naprej in nazaj po mreži. Poglavje zaključimo z opisom različnih mer za ocenjevanje uspešnosti algoritmov.

Tretje poglavje temelji na anonimiziranih internih podatkih, pridobljenih s strani lastnika izbrane mobilne aplikacije, in opisuje izvedbo procesa podatkovnega rudarjenja. Poglavje začnemo z definicijo poslovnega problema ter nadaljujemo z analizo in pripravo podatkov. Temu sledijo opis postopka izdelave in kalibracije modela logistične regresije ter nevronskih mrež in pregled uspešnosti modelov ter analiza rezultatov.

Magistrsko delo zaključimo s sklepom, v katerem na kratko povzamemo rezultate iskanja odgovora na zastavljeno raziskovalno vprašanje ter predstavimo možnost za izboljšavo dela.

1 MOBILNE APLIKACIJE

Pametni mobilni telefon je mobilni telefon, ki je sposoben opravljati številne funkcije računalnika. Po navadi ima zaslon občutljiv na dotik, internetni dostop in operacijski sistem, na katerem se lahko izvajajo mobilne aplikacije (Smartphone, 2016).

Mobilna aplikacija je programska oprema, namenjena uporabi na pametnih mobilnih telefonih, tabličnih računalnikih ter ostalih mobilnih napravah (Statista, b.l.).

1.1 Trg mobilnih aplikacij

Mobilne aplikacije so na voljo v trgovinah z mobilnimi aplikacijami, od koder si jih uporabnik namesti na mobilno napravo. Najbolj razširjeni trgovini sta Google Play in Applova trgovina App Store. Aplikacije so v trgovinah za namestitev na voljo brezplačno ali pa so plačljive (Statista, b.l.).

1.1.1 Trgovina Google Play

Trgovina Google Play je naslednica trgovine Android Market, ki je začela delovati leta 2008 (History, 2016). V trgovini Google Play so na voljo mobilne aplikacije za mobilne naprave z operacijskim sistemom Android (Google Play Store, 2016).

Z junijem 2016 je v trgovini Google Play na voljo 2,2 milijona mobilnih aplikacij, seštevek vseh nameščenih mobilnih aplikacij do maja 2016 pa znaša 65 milijard (Statista, b.l.).

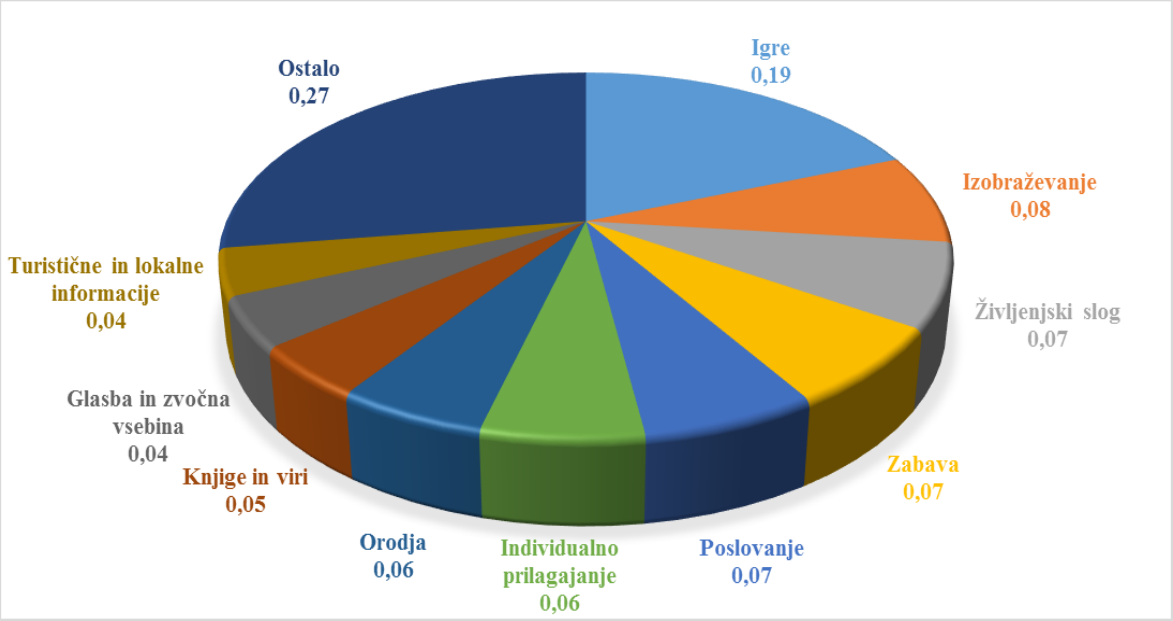
Trgovina Google Play (Apps, 2016) ima aplikacije razvrščene v več kategorij:

- android wear,
- animirano ozadje,
- časopisi in revije,
- družabno,
- družina,
- finance,
- fotografiranje,
- glasba in zvočna vsebina,
- google cast,
- igre,
- individualno prilagajanje,
- izobraževanje,
- knjige in viri,
- knjižnice in predstavitve,

- komunikacija,
- medicina,
- nakupovanje,
- orodja,
- poslovanje,
- prevoz,
- pripomočki,
- storilnost,
- stripi,
- šport,
- turistične in lokalne informacije,
- večpredstavnost in video vsebina,
- vreme,
- zabava,
- zdravo življenje,
- življenjski slog.

Iz Slike 1 so razvidne kategorije z največ mobilnimi aplikacijami v trgovini Google Play. Sedem kategorij z največ mobilnimi aplikacijami skupaj vsebuje več kot 70 % vseh mobilnih aplikacij, ki so na voljo v izbrani trgovini. V kategoriji finance je na voljo 2,14 % vseh mobilnih aplikacij trgovine Google Play (Top categories, 2016).

Slika 1: Delež mobilnih aplikacij v trgovini Google Play po posamezni kategoriji, 19. junij 2016



Povzeto in prirejeno po Top categories, 2016.

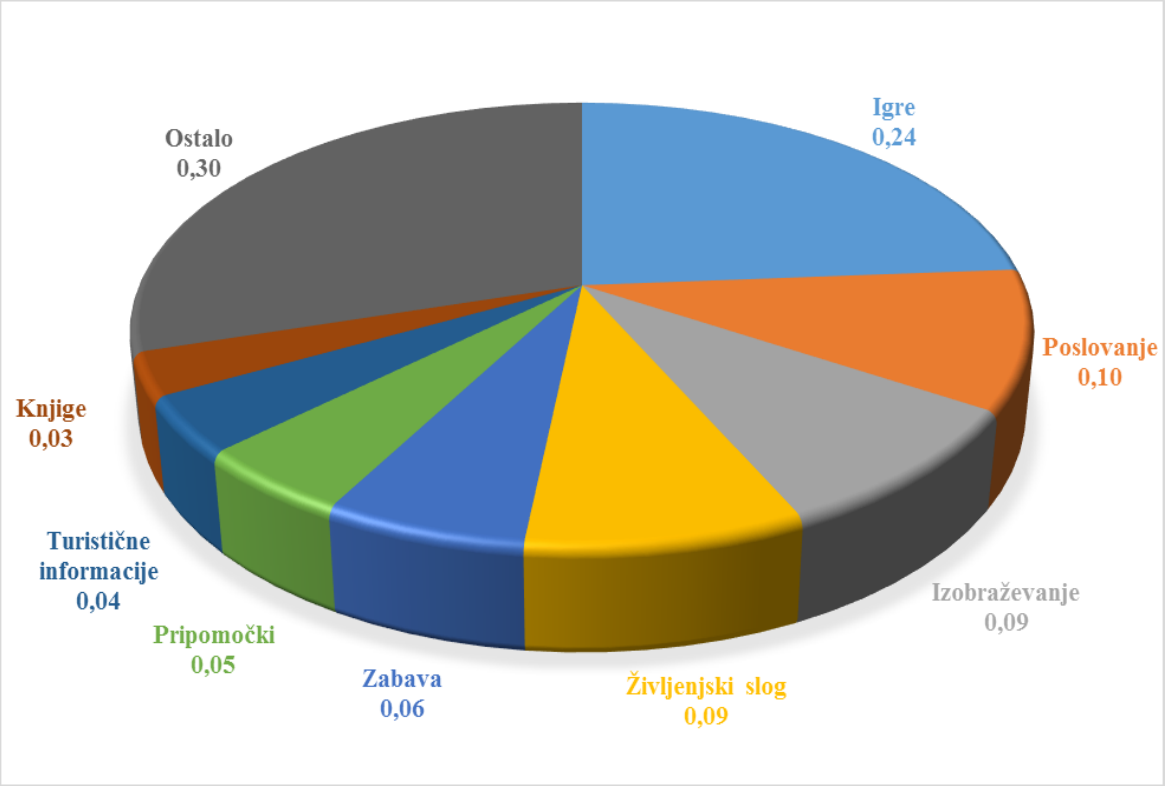
1.1.2 Trgovina App Store

Trgovina App Store je začela delovati leta 2008. V njej so na voljo mobilne aplikacije za mobilne naprave z operacijskim sistemom iPhone OS (Apple, 2008).

Z junijem 2016 je v trgovini App Store na voljo dva milijona aplikacij, seštevek vseh nameščenih mobilnih aplikacij do junija 2016 iz trgovine App Store pa znaša 130 milijard (Statista, b.l.).

Trgovina App Store ima aplikacije razvrščene v podobne kategorije kot trgovina Google Play. Iz Slike 2 so razvidne kategorije z največ mobilnimi aplikacijami v izbrani trgovini. V kategoriji finance je na voljo 2,25 % vseh mobilnih aplikacij v trgovini App Store ZDA (Pocket Gamer, b.l.).

Slika 2: Delež mobilnih aplikacij v trgovini App Store ZDA po posamezni kategoriji, 19. junij 2016



Povzeto in prirejeno po Pocket Gamer, App store metrics, b.l.

Iz Slike 1 in Slike 2 opazimo, da so v obeh trgovinah najpogosteje zastopane kategorije igre, poslovanje, izobraževanje, življenjski slog in zabava.

1.1.3 Lestvica najboljših mobilnih aplikacij

Trgovini App Store in Google Play imata med drugim na voljo tudi lestvico najboljših brezplačnih mobilnih aplikacij po lastnem izboru, ki se vsakodnevno spreminja. Tabela 1 prikazuje lestvico najboljših brezplačnih mobilnih aplikacij v Sloveniji na dan 19. junij 2016. Prikazanih je deset najboljših brezplačnih mobilnih aplikacij v obeh trgovinah. Opazimo, da sta si lestvici podobni, saj se v obeh trgovinah pojavi pet enakih aplikacij. Te so Messenger, Viber, slither.io, Snapchat in UEFA EURO 2016 Official App. Glede na Google Play so uvrščene v kategorije družabno, komunikacija in igre (Apps, 2016).

Tabela 1: Lestvica desetih najboljših brezplačnih mobilnih aplikacij v Sloveniji na dan 19. junij 2016

Uvrstitev	App Store	Google Play
1	Bud Farm: Grass Roots	Messenger
2	Slip Away	Viber
3	UEFA EURO 2016 Official App	slither.io
4	Snapchat	Facebook
5	slither.io	Snapchat
6	Insta.Vogue Skin Makeup - Retouch Wrinkle.s & Pimple.s Face.tune Edition	UEFA EURO 2016 Official App
7	iMusic BG	Instagram
8	YouTube	WhatsApp Messenger
9	Viber	Chatous
10	Messenger	Brain It On!- Physics Puzzles

Povzeto in prirejeno po App Annie, 2016.

Tabela 2: Lestvica desetih najboljših brezplačnih mobilnih aplikacij v kategoriji finance v Sloveniji na dan 19. junij 2016

Uvrstitev	App Store kategorija finance	Google Play kategorija finance
1	Mobilna banka Abamobi	Mobilna banka Abamobi
2	Hal mBills	NLB Klikin
3	Preveri račun	Hal mBills
4	Pay Pal	Plus 500
5	Xtrade – Online CDF Trading	Preveri račun
6	NLB Klikin	mBank@Net
7	Toshl Finance - save money, budget, track expenses and bills the fun way	Dh-Mobilni
8	Moj budget	Pay Pal
9	Dh-Mobilni	Mobilna banka GO!
10	Toshl Finance - expense tracker & budget manager	TebMobile

Povzeto in prirejeno po App Annie, 2016.

Tabela 2 prikazuje lestvico najboljših brezplačnih mobilnih aplikacij v kategoriji finance, v Sloveniji na dan 19. junij 2016. Prikazanih je deset najboljših brezplačnih mobilnih aplikacij v trgovinah App Store in Google Play. Opazimo, da sta si lestvici podobni, saj se v obeh trgovinah pojavi šest enakih aplikacij. Te so Mobilna banka Abamobi, NLB Klikin, Hal mBills, Preveri račun, Dh-Mobilni in Pay Pal.

1.2 Opis izbrane aplikacije

Mobilna aplikacija Hal mBills je namenjena enostavnemu plačevanju položnic in brezplačnemu nakazovanju denarnih sredstev med uporabniki (Ropret, 2015). Za namestitev na pametni mobilni telefon je v trgovinah Google Play in App Store na voljo od 9. novembra 2015 dalje. Dne 19. junija 2016 je bila v Sloveniji v trgovini Google Play v kategoriji finance uvrščena na tretje mesto in v trgovini App Store na drugo mesto. Iz trgovine Google Play je bila nameščena od 5.000- do 10.000-krat in ocenjena s povprečno oceno 4,3 (App Annie, 2016; Apps, 2016).

Po uspešni namestitvi aplikacije iz trgovine Google Play ali App Store, registraciji in avtentikaciji na tako imenovano mobilno denarnico prenesemo denarna sredstva s svojega transakcijskega računa, ki je lahko odprt pri katerikoli banki v Sloveniji. Denarna sredstva lahko prenesemo na tri načine: s trajnikom, nakazilom z univerzalnim plačilnim nalogom (v nadaljevanju UPN) in znotraj aplikacije z direktno bremenitvijo transakcijskega računa (Hal mBills, 2016; Ropret, 2015).

S sredstvi v mobilni denarnici lahko prosto razpolagamo in jih porabimo za nakazila prijateljem ali pa z njimi plačujemo položnice na dva enostavna načina. Pri prvem položnico slikamo, pri drugem pa se naročimo na avtomatsko prejemanje položnic izbranih izdajateljev v aplikacijo, ki jih lahko nato z le nekaj kliki tudi plačamo. Če imamo naloženih preveč denarnih sredstev, jih lahko prenesemo nazaj na transakcijski račun (Apps, 2016; Ropret, 2015).

Avtomatsko prejemanje položnic v aplikacijo je omogočeno za več kot 220 izdajateljev, med katerimi so tudi Zavarovalnica Triglav d.d., Petrol d.d., T-2 d.o.o. – v stečaju, Adriatic Slovenica d.d., SPL d.d. in RTV Slovenija. Prednosti za izdajatelje so predvsem v nižjih stroških poslovanja, saj prihranijo pri tisku, pakiranju in distribuciji položnic (Hal mBills, 2016).

Cenik omenjenih aktivnosti v izbrani aplikaciji na dan 19. junij 2016 prikazuje Tabela 3. Strošek plačila položnice znaša 0,25 evra (v nadaljevanju EUR), nakazilo denarnih sredstev drugemu uporabniku aplikacije je brezplačno (Hal mBills, 2016).

Tabela 3: Cenik storitev Hal mBills

Storitev	**Cena (v EUR)**
Mesečno nadomestilo uporabe	
• do 31. decembra 2016	0
• od 1. januarja 2017 dalje (ob opravljenem vsaj enem plačilu)	0,25
Polnjenje mobilne denarnice	
• prvo polnjenje s transakcijskega računa v mesecu *	0
• vsa nadaljnja polnjenja s transakcijskega računa *	0,25
• nakazilo s strani drugega uporabnika	0
Nakazila iz mobilne denarnice	
• plačilo položnice z denarnimi sredstvi v mobilni denarnici	0,25
• plačilo položnice s transakcijskega računa z direktno bremenitvijo*	0,50
• prenos denarnih sredstev nazaj na transakcijski račun	0,25
• nakazilo denarnih sredstev drugemu uporabniku	0

Legenda: * cena ne vključuje morebitnih stroškov nakazila, ki vam jih lahko zaračuna vaša banka

Povzeto in prirejeno po Hal mBills, 2016.

V aplikaciji sta na voljo dva načina registracije, Hal mBills in Hal mBills Lite. Pri načinu Hal mBills mobilno denarnico povežemo s transakcijskim računom, zato jo lahko polnimo znotraj aplikacije z le enim klikom. Možna je tudi polnitev z UPN ali trajnikom. Pred uporabo se je potrebno osebno identificirati, nato pa imamo na voljo neomejeno plačevanje. Pri načinu Hal mBills Lite mobilne denarnice ne povežemo s transakcijskim računom, zato jo je potrebno polniti z UPN ali trajnikom. Osebna identifikacija ni potrebna. Največje posamezno plačilo iz mobilne denarnice je omejeno na 999 EUR, letni priliv pa na 2.500 EUR (Hal mBills, 2016).

1.3 Pregled konkurenčnih storitev

Širše gledano so konkurenčne storitve izbrani aplikaciji vse, ki nam omogočajo plačevanje položnic in/ali nakazilo denarnih sredstev v Sloveniji. Konkurenco tako najdemo v vseh mobilnih, spletnih ter fizičnih bankah in hranilnicah ter ostalih fizičnih mestih za plačevanje položnic (občinske blagajne, trafike, trgovine,...). Pri pregledu konkurenčnih storitev se osredotočimo na konkurenčne mobilne aplikacije v Sloveniji.

Glede na lestvico najboljših brezplačnih mobilnih aplikacij v kategoriji finance v Sloveniji na dan 19. junij 2016 izluščimo pet najboljših konkurenčnih storitev aplikaciji Hal mBills. Seznam petih najboljših konkurenčnih storitev in njihova uvrstitev v trgovinah App Store in Google Play sta prikazana v Tabeli 4.

Tabela 4: Lestvica petih najboljših konkurenčnih storitev v kategoriji finance v Sloveniji na dan 19. junij 2016

Uvrstitev	App Store, kategorija finance	Uvrstitev	Google Play, kategorija finance
1	Mobilna banka Abamobi	**1**	Mobilna banka Abamobi
6	NLB Klikin	**2**	NLB Klikin
9	Dh-Mobilni	**6**	mBank@Net
12	mBank@Net	**7**	Dh-Mobilni
18	Mobilna banka mLON	**9**	Mobilna banka GO!

Povzeto in prirejeno po App Annie, 2016.

Pod drobnogled vzamemo presek konkurenčnih storitev iz trgovine App Store in Google Play iz Tabele 4. Mobilno banko Abamobi, NLB Klikin, mBank@Net, Dh-mobilni in izbrano aplikacijo Hal mBills med seboj primerjamo po Google Play statistikah, ponujenih funkcionalnostih in ceni.

1.3.1 Pregled Google Play statistik

Trgovina Google Play ima za mobilne aplikacije, ki so v njej na voljo za namestitev, med drugim objavljene tudi njihove povprečne ocene in število namestitev (Apps, 2016). Število namestitev in povprečne ocene mobilnih aplikacij, izbranih v naši primerjavi, so razvidne iz Tabele 5.

Tabela 5: Statistike trgovine Google Play za mobilno aplikacijo Hal mBills in izbrane konkurenčne mobilne aplikacije v Sloveniji na dan 19. junij 2016

Google Play kategorija finance	Povprečna ocena	Število namestitev	Začetek spremljanja statistike
Mobilna banka Abamobi	4,6	10.000–50.000	7. maj 2014
NLB Klikin	4,2	10.000–50.000	26. oktober 2014
Hal mBills	4,3	5.000–10.000	9. november 2015
mBank@Net	4,4	10.000–50.000	20. avgust 2012
Dh-Mobilni	4,7	5.000–10.000	7. september 2014

Povzeto in prirejeno po App Annie, 2016.

Opazimo, da imajo konkurenčne aplikacije v trgovini Google Play 5.000–50.000 namestitev in povprečno oceno 4,2–4,7. Med aplikacijami prihaja le do manjših razlik v številu namestitev in povprečni oceni, čeprav so na tržišču na voljo različno dolgo. Aplikacija mBank@Net je na voljo od avgusta 2012 in ima enako število namestitev (10.000–50.000) kot aplikaciji NLB Klikin, ki je na voljo od oktobra 2014, ter Mobilna banka Abamobi, ki je na voljo od maja 2014. V malo več kot sedmih mesecih, odkar je na tržišču aplikacija Hal mBills, je po številu namestitev (5.000–10.000) ujela aplikacijo Dh-mobilni, ki je na tržišču od septembra 2014.

1.3.2 Pregled funkcionalnosti

Mobilna banka Abamobi komitentom banke Abanka d.d. (v nadaljevanju Abanka) ponuja plačevanje položnic s slikanjem položnic ali ročnim vnosom, pregled in plačevanje e-računov, prenos denarnih sredstev med transakcijskimi računi, pregled nad stanjem na vseh transakcijskih računih, debetnih ter plačilnih karticah, varčevanjih in kreditih, sklenjenih pri Abanki. Dodatno ponuja tudi iskanje bankomatov in poslovalnic Abanke, pregled menjalnih tečajev ter kalkulator varčevanj (Apps, 2016; Mobilna banka Abamobi, 2016).

Mobilna banka NLB Klikin komitentom banke NLB d.d. (v nadaljevanju NLB) ponuja plačevanje položnic s slikanjem položnic ali ročnim vnosom, prenos denarnih sredstev med uporabniki NLB Klikin in transakcijskimi računi, pregled nad stanjem in prometom vseh transakcijskih računov ter plačilnih karticah, odprtih pri NLB, iskanje bankomatov NLB, pregled aktualnih menjalnih tečajev, menjavo denarnih sredstev na transakcijskem računu v želeno valuto, informativno tečajnico ter preklic izgubljene ali ukradene kartice (Klikin, 2016).

Mobilna aplikacija Hal mBills komitentom katerekoli banke v Sloveniji ponuja pregled in plačevanje položnic s slikanjem, ročnim vnosom ali naročitvijo na avtomatsko prejemanje položnic izbranih izdajateljev. Dodatno ponuja tudi prenos denarnih sredstev med uporabniki aplikacije (Hal mBills, 2016).

Mobilna banka mBank@Net komitentom banke Nova KBM d.d. (v nadaljevanju NKBM) ponuja plačevanje položnic s skeniranjem ali z ročnim vnosom, prenos denarnih sredstev med transakcijskimi računi, pregled nad prometom na transakcijskih računih, karticah, varčevanjih in kreditih, sklenjenih pri NKBM, iskanje bankomatov NKBM, iskanje poslovalnic NKBM ter Pošte Slovenije d.o.o. Dodatno ponuja tudi aktualno tečajno listo, menjalnico, varčevalnik ter pregled nad stanjem in prometom Monete (Nova KBM, 2016).

Mobilna banka Dh-Mobilni komitentom hranilnice Delavska hranilnica d.d. (v nadaljevanju Delavska hranilnica) ponuja plačevanje položnic s slikanjem položnic, z ročnim vnosom, pregled poslovanja na transakcijskih računih, plačilnih karticah, hranilnih knjižicah, rentnih vlogah in kreditih, sklenjenih pri Delavski hranilnici, iskanje bankomatov ter poslovalnic Delavske hranilnice, aktualno tečajno listo in menjavo denarnih sredstev na transakcijskem računu v želeno valuto (Delavska hranilnica, 2016).

Iz zgornjih opisov funkcionalnosti izbranih mobilnih aplikacij opazimo, da imajo mobilne banke med seboj podobne funkcionalnosti, vendar se v samem konceptu razlikujejo od aplikacije Hal mBills. Opisane mobilne banke so namenjene le svojim komitentom in nadomeščajo ali dopolnjujejo že obstoječe spletne banke, medtem ko je aplikacija Hal mBills namenjena enostavnemu plačevanju položnic ter nakazilom denarnih sredstev drugim uporabnikom aplikacije, ki so lahko komitenti katerekoli banke v Sloveniji.

V Tabeli 6 je prikazan pregled funkcionalnosti posameznih mobilnih aplikacij glede na funkcionalnosti, ki jih ponuja Hal mBills.

Tabela 6: Funkcionalnosti posameznih mobilnih aplikacij na dan 27. junij 2016

	Mobilna banka Abamobi	**NLB Klikin**	**Hal mBills**	**mBank@Net**	**Dh-Mobilni**
Plačevanje položnic:					
• s slikanjem položnic	✓	✓	✓	✓	✓
• z ročnim vnosom	✓	✓	✓	✓	✓
• z avtomatskim prejemanjem	✓	×	✓	×	×
Prenos denarnih sredstev prijatelju	✓	✓	✓	✓	✓

Povzeto in prirejeno po Delavska hranilnica, 2016; Hal mBills, 2016; Klikin, 2016; Mobilna banka Abamobi, 2016; Nova KBM, 2016.

Opazimo, da le Mobilna banka Abamobi ponuja enake funkcionalnosti kot aplikacija Hal mBills, vendar se je pri Mobilni banki Abamobi za avtomatsko prejemanje položnic oz. e-računov potrebno lastnoročno naročiti pri vsakem izdajatelju posebej, kar pomeni, da je v to potrebno vložiti precej časa. Pri aplikaciji Hal mBills za naročitev na avtomatsko prejemanje položnic potrebujemo le nekaj klikov po aplikaciji, kar nam vzame samo minuto.

1.3.3 Cenovni pregled

Pri cenovnem pregledu mobilnih aplikacij predpostavimo, da imamo pri posamezni banki odprt klasičen račun brez akcij, posebnih ponudb in popustov za selektivne skupine (študentje, dijaki, upokojenci, člani gasilskega društva,...) ter da nismo obstoječi uporabnik pripadajoče spletne banke. Pri Abanki v pregled vzamemo komitenta s transakcijskim računom Aračun, pri NLB s Klasičnim računom, pri NKBM z Osebnim računom z debetno kartico in pri Delavski hranilnici z Osebnim računom za zaposlene. Pri aplikaciji Hal mBills predpostavimo, da mobilno denarnico polnimo enkrat mesečno.

Dodatno predpostavimo, da se položnice plačujejo z UPN, da so vsa plačila namenjena prejemnikom znotraj državnih meja, v domači valuti, v zneskih do 15.000 EUR na transakcijski račun odprt pri drugi banki. Pri prenosu denarnih sredstev prijatelju upoštevamo možnosti, da ima prijatelj odprt transakcijski račun pri isti banki ter da ima odprt transakcijski račun pri drugi banki. Cenik storitev mobilnih aplikacij z omenjenimi predpostavkami prikazuje Tabela 7.

Tabela 7: Cenik storitev mobilnih aplikacij na dan 27. junij 2016 v EUR

	Mobilna banka Abamobi	**NLB Klikin**	**Hal mBills**	**mBank@Net**	**Dh-Mobilni**
Enkratna pristopnina	5,00	10,00	0	33,50**	0
Mesečna uporabnina	0,65	0,30	0*	0,60**	0,40
Plačilo položnice z obrazcem UPN	0,39	0,38	0,25	0,38	0,20
Prenos denarnih sredstev prijatelju na transakcijski račun:					
• odprt pri isti banki	0	0	0	0	0
• odprt pri drugi banki	0,39	0,38	0	0,38	0,20

Legenda: * cena velja do 31. decembra 2016. Od 1. januarja 2017 dalje (ob opravljenem vsaj enem plačilu) bo mesečna uporabnina znašala 0,25 EUR
** mBank@Net je na voljo samo uporabnikom spletne banke Bank@Net. Cene tako veljajo za hkratno uporabo spletne banke Bank@Net in mobilne banke mBank@Net

Povzeto in prirejeno po Delavska hranilnica, 2016; Hal mBills, 2016; Klikin, 2016; Mobilna banka Abamobi, 2016; Nova KBM, 2016.

V cenovno primerjavo mobilnih aplikacij vzamemo dva uporabnika:

- prvega, ki mesečno plača šest položnic z UPN in prenese denarna sredstva prijatelju, ki ima transakcijski račun odprt pri isti banki, in prijatelju, ki ima transakcijski račun odprt pri drugi banki,
- drugega, ki mesečno plača deset položnic z UPN in prenese denarna sredstva dvema prijateljema, ki imata transakcijski račun odprt pri isti banki, in dvema prijateljema, ki imata transakcijski račun odprt pri drugi banki.

Za oba uporabnika vzamemo obdobje uporabe od 1. julija 2016 do 30. junija 2017 ob predpostavki, da se cenik in ponudba storitev pri nobeni mobilni aplikaciji v danem obdobju ne bosta spremenila ter da veljajo vse zgoraj opisane predpostavke.

Cenovno primerjavo mobilnih aplikacij za prvega uporabnika prikazuje Tabela 8. Opazimo, da je za takega uporabnika cenovno najugodnejša uporaba mobilne aplikacije Hal mBills. Malenkost dražja je uporaba mobilne banke Dh-mobilni ob predpostavki, da je uporabnik komitent Delavske hranilnice. Vse ostale mobilne banke, obravnavane v cenovni primerjavi, so neprimerno dražje.

Tabela 8: Cenovna primerjava mobilnih aplikacij za prvega uporabnika v obdobju od 1. julija 2016 do 30. junija 2017 v EUR

	Mobilna banka Abamobi	**NLB Klikin**	**Hal mBills**	**mBank@Net**	**Dh-Mobilni**
Enkratna pristopnina	5,00	10,00	0	33,50*	0
12 x mesečna uporabnina	7,80	3,60	1,50	7,20*	4,80
12 x plačilo 6 položnic z obrazcem UPN	28,08	27,36	18,00	27,36	14,40
12 x prenos denarnih sredstev prijatelju na transakcijski račun:					
• odprt pri isti banki	0	0	0**	0	0
• odprt pri drugi banki	4,68	4,56	0**	4,56	2,40
SKUPAJ	45,56	45,52	19,50	72,62	21,60

Legenda: * mBank@Net je na voljo samo uporabnikom spletne banke Bank@Net. Cene tako veljajo za hkratno uporabo spletne banke Bank@Net in mobilne banke mBank@Net
** storitev je izvedljiva ob predpostavki, da je prijatelj uporabnik mobilne aplikacije Hal mBills

Povzeto in prirejeno po Delavska hranilnica, 2016; Hal mBills, 2016; Klikin, 2016; Mobilna banka Abamobi, 2016; Nova KBM, 2016.

Tabela 9: Cenovna primerjava mobilnih aplikacij za drugega uporabnika v obdobju od 1. julija 2016 do 30. junija 2017

	Mobilna banka Abamobi	**NLB Klikin**	**Hal mBills**	**mBank@Net**	**Dh-Mobilni**
Enkratna pristopnina	5,00	10,00	0	33,50*	0
12 x mesečna uporabnina	7,80	3,60	1,50	7,20*	4,80
12 x plačilo 10 položnic z obrazcem UPN	46,80	45,60	30,00	45,60	24,00
24 x prenos denarnih sredstev prijatelju na transakcijski račun:					
• odprt pri isti banki	0	0	0**	0	0
• odprt pri drugi banki	9,36	9,12	0**	9,12	4,80
SKUPAJ	68,96	68,32	31,50	95,42	33,60

Legenda: * mBank@Net je na voljo samo uporabnikom spletne banke Bank@Net. Cene tako veljajo za hkratno uporabo spletne banke Bank@Net in mobilne banke mBank@Net
** storitev je izvedljiva ob predpostavki, da je prijatelj uporabnik mobilne aplikacije Hal mBills

Povzeto in prirejeno po Delavska hranilnica, 2016; Hal mBills, 2016; Klikin, 2016; Mobilna banka Abamobi, 2016; Nova KBM, 2016.

Cenovno primerjavo mobilnih aplikacij za drugega uporabnika prikazuje Tabela 9. Opazimo, da je tudi za takega uporabnika cenovno najugodnejša uporaba mobilne aplikacije Hal mBills. Malenkost dražja je uporaba mobilne banke Dh-mobilni ob predpostavki, da je

uporabnik komitent Delavske hranilnice. Vse ostale mobilne banke, obravnavane v cenovni primerjavi, so tudi za drugega uporabnika neprimerno dražje.

2 ALGORITMI PODATKOVNEGA RUDARJENJA

2.1 Podatkovno rudarjenje

Podatkovno rudarjenje je sistematično iskanje znanja v poljubnih vhodnih podatkih (Berry & Linoff, 2004). Vhodni podatki so za podatkovno rudarjenje podani v obliki tabele. Sestavljeni so iz množice statističnih enot, ki so med seboj neodvisne. Statistične enote predstavljajo vrstice tabele z vhodnimi podatki. V našem primeru so vhodni podatki sestavljeni iz množice uporabnikov izbrane mobilne aplikacije. Vsaka statistična enota je sestavljena iz vnaprej definiranih atributov, to so lastnosti posameznih statističnih enot, ki so v tabeli z vhodnimi podatki predstavljeni v stolpcih. V našem primeru nam izbrani atributi povedo nekaj o posameznih uporabnikih. Primeri atributa so starost, spol, dohodek, ipd. Atribute pogosto imenujemo tudi neodvisne oz. pojasnjevalne spremenljivke (Witten, Eibe, & Hall, 2011).

Tabelo z vhodnimi podatki pred obdelavo razdelimo na učno (angl. *training set*), validacijsko (angl. *validation set*) in testno množico (*test set*). Iz učne množice se algoritem uči, validacijsko množico uporabimo za izbiro najboljšega algoritma, testno pa za ocenjevanje uspešnosti izbranega algoritma (Berry & Linoff, 2004).

Podatkovno rudarjenje delimo na dve skupini: nadzorovano in nenadzorovano učenje (Berry & Linoff, 2004).

Pri nadzorovanem učenju imamo poleg vhodnih podatkov znane tudi izhodne podatke, ki jih želimo napovedati. Algoritem, ki se na učnih podatkih nauči napovedovanja, klasificiranja ali ocenjevanja, pozneje uporabimo na novih vhodnih podatkih, za katere izhodni podatki niso znani. Nadzorovano učenje se dalje deli na postopek klasifikacije in regresije (Zhou, 2012).

Pri klasifikaciji algoritem napoveduje diskretne vrednosti. Za vsako statistično enoto v vhodnih podatkih določi enega izmed vnaprej definiranih razredov (angl. *classes*). Razred je želeno izhodno diskretno stanje, ki ga želimo s klasifikacijo določiti. Vsako možno izhodno diskretno stanje, ki mu pravimo tudi odvisna spremenljivka, mora pripadati natanko enemu razredu (Marsland, 2009).

Primer klasifikacijskega problema je avtomatsko določanje vrednosti vstavljenega kovanca v avtomat. Ko ga vstavimo, lahko avtomat izmeri premer, težo, barvo kovanca, ipd.

Izmerjene mere predstavljajo atribute, na podlagi katerih algoritem določi, kateri vrednosti kovanca pripada pravkar vstavljeni kovanec (Marsland, 2009).

Pri regresiji algoritem napoveduje zvezne vrednosti. Za vsako statistično enoto v vhodnih podatkih napove izhodno stanje. V statistiki pri regresijskem problemu z matematično funkcijo opišemo krivuljo, ki se poskuša čim bolj približati vsem vhodnim podatkom, vključno z znanimi izhodnimi podatki, ki jih želimo napovedati. Regresijski problem je tako običajno problem aproksimacije ali interpolacije (Marsland, 2009).

Primer regresijskega problema je določanje tržne vrednosti hiše. Za atribute vzamemo površino in starost hiše, površino parcele, lokacijo, ipd. Na izbranih atributih se algoritem nauči napovedovati tržno vrednost katerekoli hiše. Uporaba opisanega algoritma lahko služi kot pripomoček nepremičninskih agencij (Hu, Wang, & Feng, 2013).

Pri nenadzorovanem učenju imamo na voljo le vhodne podatke, na podlagi katerih algoritem za tovrstno učenje poskuša najti vzorec. Najpogostejši primer je grupiranje, kjer algoritem razdeli podatke po skupinah brez predhodnega znanja o pomembnosti posameznih atributov v vhodnih podatkih (Zhou, 2012).

Napovedovanje prenehanja uporabe izbrane mobilne aplikacije sodi v klasifikacijski problem. Najpogosteje se za tak problem uporabi logistično regresijo, nevronske mreže in odločitvena drevesa (Clemente et al., b.l.). Odločimo se za uporabo logistične regresije in nevronskih mrež, ki so podrobneje opisane v nadaljevanju.

Logistično regresijo izberemo zaradi njene vsesplošne in pogoste uporabe v literaturi, predvsem pri napovedovanju odhajanja strank (angl. *churn prediction*), zaradi njenega preprostega modeliranja ter enostavne razlage končnih rezultatov. Nevronske mreže izberemo za primerjavo, da vidimo, ali je uporaba sofisticirane metode bolj smiselna od uporabe preprostejše logistične regresije.

2.2 Logistična regresija

Logistična regresija meri razmerje med izhodnim stanjem in enim ali več neodvisnimi atributi tako, da ocenjuje verjetnost, s katero posamezna statistična enota pripada določenemu razredu. Omenjeno verjetnost ocenjuje s pomočjo logistične funkcije (Hosmer, Hosmer, Le Cessie, & Lemeshow, 1997).

V našem primeru napovedovanja prenehanja uporabe izbrane mobilne aplikacije imamo definirana dva razreda, ki ju zaradi preprostosti označimo z 0 in 1. Posamezen izhodni podatek y_i tako pripada ali razredu 0 ali 1 in ima Bernoullijevo porazdelitev z verjetnostjo p_i:

$$y_i \sim \begin{pmatrix} 0 & 1 \\ 1 - p_i & p_i \end{pmatrix} \quad (1)$$

za vsak $i = 1, 2, \ldots, m$, kjer je m število statističnih enot v vhodnih podatkih (Forbes, Evans, Hastings, & Peacock, 2011).

Matematično upanje je izračunano kot

$$E(y_i) = 0 \cdot (1 - p_i) + 1 \cdot p_i = p_i \quad (2)$$

za vsak $i = 1, 2, \ldots, m$. Varianca je izračunana kot

$$D(y_i) = E(y_i^2) - E(y_i)^2 = 0^2 \cdot (1 - p_i) + 1^2 \cdot p_i - p_i^2 = p_i (1 - p_i) \quad (3)$$

za vsak $i = 1, 2, \ldots, m$. Tako matematično upanje kot varianca sta odvisna od verjetnosti p_i (Forbes et al., 2011).

Velja tudi, da je

$$P(y_i = 1) = E(y_i) = p_i \quad (4)$$

za vsak $i = 1, 2, \ldots, m$ (Shalizi, 2016).

Za klasificiranje statističnih enot v ustrezen razred potrebujemo zvezo med vhodnimi podatki $X_i^{(j)}$ in izhodnim podatkom y_i. Vhodni podatek $X_i^{(j)}$ predstavlja vrednost j-tega atributa pri i-ti statistični enoti. Par (X_i, y_i) predstavlja i-to statistično enoto. Medsebojno zvezo vzpostavimo z definiranjem verjetnosti p_i v odvisnosti od vhodnih podatkov.

Najprej definiramo razmerje obetov (angl. *odds ratio*):

$$\text{razmerje obetov}_i = \frac{p_i}{1 - p_i} \quad (5)$$

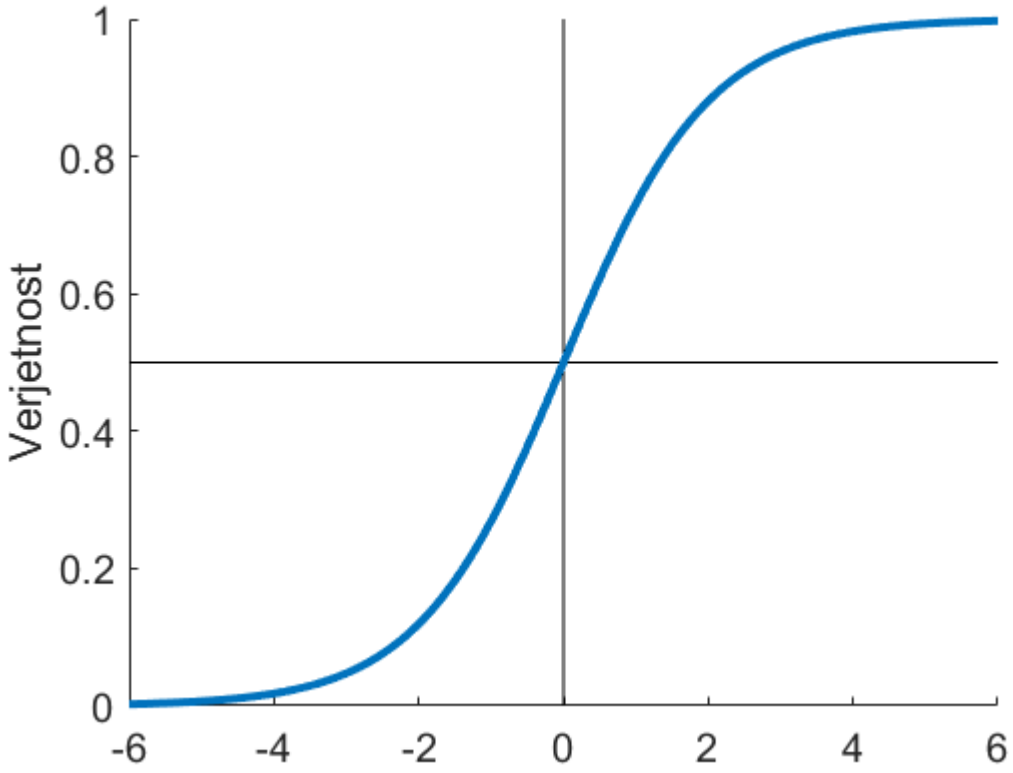
za vsak $i = 1, 2, \ldots, m$. Razmerje obetov predstavlja razmerje med tem, da izhodni podatek, ki ga algoritem določi, pripada razredu 1, in tem, da pripada razredu 0. Če je verjetnost, da izhodni podatek, ki ga algoritem določi, pripada razredu 1, enaka $1/3$, potem je razmerje obetov enako 1 proti 2. Razmerje obetov ima zalogo vrednosti med 0 in ∞ (James, Witten, Hastie, & Tibshirani, 2013).

Razmerje obetov logaritmiramo

$$\log(\textit{razmerje obetov}_i) = \log\left(\frac{p_i}{1 - p_i}\right) \quad (6)$$

za vsak $i = 1, 2, \ldots, m$. Dobljeno funkcijo imenujemo logit transformacija verjetnosti p_i in jo prikažemo na Sliki 3. Ima zalogo vrednosti na intervalu $(-\infty, \infty)$. Verjetnosti manjše od 0,5 so predstavljene z negativnimi vrednostmi, verjetnosti večje od 0,5 pa s pozitivnimi. Verjetnost 0,5 je na grafu predstavljena z vrednostjo 0 (Rodriguez, 2007).

Slika 3: Logit transformacija verjetnosti



Vir: G. Rodriguez, Lecture Notes on Generalized Linear Models, 2007, str. 7, slika 3.1.

Predpostavimo, da je logit transformacija verjetnosti p_i enaka

$$\text{logit}(p_i) = X_i\theta \quad (7)$$

za vsak $i = 1, 2, \ldots, m$, kjer je X_i^T vektor atributov za posamezno statistično enoto i, θ pa vektor regresijskih koeficientov. Logit transformacija je sedaj linearna kombinacija vhodnih atributov (Rodriguez, 2007).

Razmerje obetov dobimo z eksponiranjem enačbe (7)

$$\frac{p_i}{1 - p_i} = e^{X_i\theta} \quad (8)$$

za vsak $i = 1, 2, \ldots, m$ (Rodriguez, 2007).

V kolikor enačbo (8) izrazimo drugače, dobimo

$$p_i = \frac{e^{X_i\theta}}{1 + e^{X_i\theta}} = \frac{1}{1 + e^{-X_i\theta}} \quad (9)$$

za vsak $i = 1, 2, \ldots, m$. Dobljena enačba je logistična funkcija, ki nam pove, s kakšno verjetnostjo posamezna statistična enota i pripada razredu 1 (Hand, Heikki, & Smyth, 2001).

2.2.1 Metoda največjega verjetja za logistično regresijo

Logistična regresija se uči klasificirati statistične enote v vhodnih podatkih tako, da določi vektor regresijskih koeficientov s pomočjo metode največjega verjetja. Vektor regresijskih koeficientov dobimo tako, da najprej izračunamo pripadajočo funkcijo verjetja $L(\theta)$

$$L(\theta) = \prod_{i=1}^{m} p_i^{y_i}(1 - p_i)^{1-y_i} \quad (10)$$

(Hosmer, Lemeshow, & Sturdivant, 2013).

Nato funkcijo verjetja logaritmiramo. Njen logaritem označimo z $l(\theta)$

$$\begin{aligned} l(\theta) &= \log(L(\theta)) \\ &= \sum_{i=1}^{m} \left(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right) \\ &= \sum_{i=1}^{m} y_i \log\left(\frac{p_i}{1 - p_i}\right) + \sum_{i=1}^{m} \log(1 - p_i) \\ &= \sum_{i=1}^{m} y_i X_i\theta - \sum_{i=1}^{m} \log(1 + e^{X_i\theta}) \end{aligned} \quad (11)$$

(Shalizi, 2016).

Maksimum funkcije $l(\theta)$ poiščemo tako, da funkcijo odvajamo po regresijskih koeficientih

$$\begin{aligned} \frac{\partial l}{\partial \theta_j} &= \sum_{i=1}^{m} y_i X_i^{(j)} - \sum_{i=1}^{m} \frac{1}{1 + e^{X_i\theta}} e^{X_i\theta} X_i^{(j)} \\ &= \sum_{i=1}^{m} (y_i - p_i) X_i^{(j)} \end{aligned} \quad (12)$$

in rešimo enačbo

$$\frac{\partial l}{\partial \theta_j} = 0 \quad (13)$$

za vsak $j = 1, 2, \ldots, n$, kjer je n število atributov v vhodnih podatkih. Slednjo rešimo numerično, kar lahko naredimo z različnimi programi, vključno z Matlabom. Enačba (13) predstavlja vsoto razlike med izhodnimi podatki in verjetnostmi, ki jih algoritem določi, pomnoženo z vhodnimi podatki. Rešitev so regresijski koeficienti, ki jih zapišemo v vektor regresijskih koeficientov θ in minimizirajo razliko med izhodnimi podatki in verjetnostmi, ki jih algoritem določi. Razliki med izhodnimi podatki in verjetnostmi, ki jih algoritem določi, pravimo tudi napaka (angl. *error*). Napake imajo pri logistični porazdelitvi z dvema razredoma Bernoullijevo porazdelitev (Bishop, 2006; Hosmer et al., 2013).

2.2.2 Ocena algoritma logistične regresije

Po zasnovi algoritma logistične regresije je potrebno le tega testirati, ali je statistično boljši od algoritma, ki ne vsebuje nobenega atributa, ampak le konstanto. Torej, ali je res, da s pomočjo atributov algoritem logistične regresije natančneje določi izhodne vrednosti kot brez njih (Menard, 2001).

Pri klasifikaciji nas poleg samih verjetnosti lahko zanima tudi, kateri atributi so v algoritmu pomembni, torej ali je klasifikacijski algoritem, ki vsebuje določen atribut, v nekem smislu natančnejši kot algoritem, ki tega atributa ne vsebuje. V ta namen ocenimo statistično značilnost posameznih atributov (Hosmer et al., 2013).

Za testiranje statistične značilnosti algoritmov in za oceno statistične značilnosti posameznih atributov lahko uporabimo različne teste, med njimi sta tudi test razmerja verjetij (angl. *likelihood ratio test*) in Waldov test (angl. *Wald test*). Test razmerja verjetij med seboj primerja odklon algoritma, ki ne vsebuje izbranih atributov, z odklonom algoritma, ki izbrane atribute vsebuje (Peng & So, 2002).

V našem primeru za testiranje statistične značilnosti algoritma uporabimo test razmerja verjetij, za testiranje statistične značilnosti posameznih atributov pa Waldov test.

2.2.2.1 Test razmerja verjetij

Test razmerja verjetij med seboj primerja algoritem, ki ne vsebuje izbranih atributov, z algoritmom, ki izbrane atribute vsebuje. Drugače povedano, test razmerja verjetij med seboj primerja algoritem, ki ima pri izbranih atributih pripadajoče regresijske koeficiente enake 0, z algoritmom, ki te omejitve nima (Engle, 1984).

V primeru uporabe testa razmerja verjetij za testiranje statistične značilnosti algoritma ta med seboj primerja algoritem, ki ima vse regresijske koeficiente enake 0 ($\theta \in \Theta_0$) in tako vsebuje le konstanto, z algoritmom, ki te omejitve nima ($\theta \in \Theta$). Pripadajoča enačba je

$$\lambda = -2 \log \left[\frac{\max_{\theta \in \Theta_0} L(\theta)}{\max_{\theta \in \Theta} L(\theta)} \right] \quad (14)$$
$$= 2 \left[\max_{\theta \in \Theta} l(\theta) - \max_{\theta \in \Theta_0} l(\theta) \right]$$

Statistika λ ima hi-kvadrat porazdelitev s številom prostostnih stopenj enakim razliki med številom atributov v obeh algoritmih. V primeru testiranja algoritma z vsemi atributi z algoritmom, ki vsebuje le konstanto, je ta razlika enaka *n*, kjer *n* predstavlja število vseh atributov (Tsay, 2008).

V kolikor je pripadajoča p-vrednost (angl. *p-value*) manjša od izbrane (v našem primeru 0,05), sledi, da je vsaj en regresijski koeficient različen od 0, kar pomeni, da je algoritem, ki vsebuje izbrane atribute, statistično značilen in primeren za uporabo (Hosmer et al., 2013).

Pripadajočo p-vrednost označimo z α in jo za naš primer izračunamo kot

$$P(\chi^2(n) > \lambda) \leq \alpha \quad (15)$$

kjer je *n* število prostostnih stopenj hi-kvadrat porazdelitve statistike λ (Hosmer et al., 2013).

2.2.2.2 Waldov test

Waldov test uporabimo za testiranje statistične značilnosti posameznih atributov. Test med seboj primerja regresijske koeficiente z oceno njihove standardne napake (Peng & So, 2002).

Enovariatna Waldova testna statistika je definirana kot

$$W_j = \frac{\theta_j}{SE(\theta_j)} \quad (16)$$

za vsak $j = 0, 1, \ldots, n$, kjer je $SE(\theta_j)$ standardna napaka regresijskega koeficienta θ_j. Waldova statistika ima v primeru, ko ničelna hipoteza drži (da je regresijski koeficient posameznega atributa enak 0), ko so podatki normalno porazdeljeni in ko je standardna napaka regresijskega koeficienta θ_j znana, standardno normalno porazdelitev. V kolikor je standardna napaka regresijskega koeficienta θ_j ocenjena, ima Waldova statistika Studentovo t porazdelitev z $m - 1$ prostostnimi stopnjami, kjer je *m* število statističnih enot v naših vhodnih podatkih (Rodriguez, 2007).

Pri velikih vzorcih Studentova t porazdelitev konvergira k standardno normalni porazdelitvi. Posledično ima Waldova statistika pri velikih vzorcih približno standardno normalno porazdelitev (Rodriguez, 2007).

V kolikor je pripadajoča p-vrednost manjša od izbrane (v našem primeru 0,05), sledi, da je regresijski koeficient različen od 0, kar pomeni, da je posamezen atribut statistično značilen in ima pojasnjevalno moč (Hosmer et al., 2013).

Pripadajočo p-vrednost izračunamo podobno kot v enačbi (15), le da hi-kvadrat porazdelitev zamenjamo s standardno normalno ali s Studentovo t porazdelitvijo z $m - 1$ prostostnimi stopnjami

$$P\left(F^{-1}_{t_{m-1}} > W_j\right) \leq \alpha_j \qquad (17)$$

za vsak $j = 0, 1, \ldots, n$ (Rodriguez, 2007).

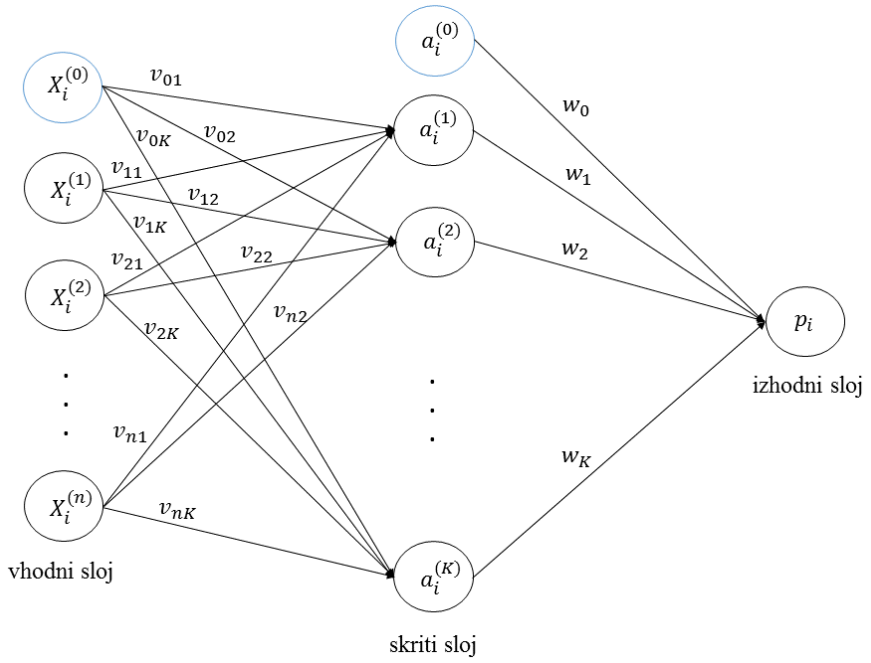
2.3 Nevronske mreže

Nevronske mreže skušajo posnemati delovanje človeških možganov. Ti so sestavljeni iz med seboj povezanih nevronov, ki sprejemajo električne impulze iz sosednjih nevronov. Impulze akumulirajo, vse dokler ni presežen določen prag. Ko je ta presežen, nevron pošlje impulz sosednjemu nevronu. Zmožnost shranjevanja električnih impulzov in določanje praga nadzorujejo biokemični procesi, ki se čez čas spreminjajo (Nisbet, Elder, & Miner, 2009).

Nevronske mreže so sestavljene iz umetnih nevronov in imajo sposobnost akumuliranja in oddajanja signalov. Akumuliranje signalov zajemata sprejem signalov in seštevek teh, oddajanje pa je predstavljeno z matematično funkcijo, ki ji pravimo prenosna funkcija (Nisbet et al., 2009).

Pri napovedovanju prenehanja uporabe izbrane mobilne aplikacije se odločimo za uporabo večslojne usmerjene nevronske mreže, ki je sestavljena iz vhodnega, enega ali več skritih ter izhodnega sloja oz. nivoja (angl. *layer*). Vsak sloj je sestavljen iz enega ali več nevronov. Pri binarnem klasifikacijskem problemu se najpogosteje uporabi večslojno usmerjeno nevronsko mrežo, sestavljeno iz enega skritega sloja in izhodnega sloja, ki vsebuje le en nevron. Tudi mi se odločimo za uporabo take nevronske mreže. Za lažjo predstavo je struktura te razvidna iz Slike 4 (Viaene, Baesens, Van den Poel, Vanthienen, & Dedene, 2002).

Slika 4: Večslojna usmerjena nevronska mreža



Povzeto in prirejeno po A. Ng, Machine Learning, 2016.

Kot lahko vidimo iz Slike 4, je naša večslojna usmerjena nevronska mreža sestavljena iz treh slojev. Vhodni sloj vsebuje toliko nevronov, kot je atributov v naših vhodnih podatkih z dodatnim nevronom, ki predstavlja konstanto. Njegovo vrednost označimo z $X_i^{(0)}$. Velja, da je $X_i^{(0)} = 1$ za vsak $i = 1, 2, \ldots, m$ (Zhou, 2012).

Vsak nevron j iz vhodnega sloja je povezan z vsakim nevronom k iz skritega sloja z utežmi v_{jk}. Število skritih nevronov določimo sami, pri čemer moramo vedeti, da premalo nevronov lahko privede do izgube informacij med vhodnimi in izhodnimi podatki, preveč nevronov pa do pretiranega prilagajanja (angl. *overfitting*). Izbranemu številu skritih nevronov dodamo nevron, ki predstavlja konstanto. Njegovo vrednost označimo z $a_i^{(0)}$. Velja, da je $a_i^{(0)} = 1$ za vsak $i = 1, 2, \ldots, m$. Vsak nevron k iz skritega sloja je dalje povezan z izhodnim nevronom z utežmi w_k (Berry & Linoff, 2004; Bishop, 2006).

Oddajanje signalov vhodnih nevronov skritim nevronom in skritih nevronov izhodnemu nevronu poteka s pomočjo prenosne funkcije. Izbiramo lahko med različnimi prenosnimi funkcijami, izmed katerih so najpogosteje uporabljene hiperbolični tangens ter logistična in linearna funkcija. Izbira prenosne funkcije je odvisna od zastavljenega raziskovalnega vprašanja. Pri napovedovanju prenehanja uporabe izbrane mobilne aplikacije za prenosno funkcijo uporabimo logistično funkcijo (Zidar & Biloslavo, 2010).

Treniranje nevronskih mrež lahko poteka na različne načine. Najpogosteje je uporabljeno premikanje naprej in nazaj po mreži. Pri premikanju naprej izračunamo izhodne verjetnosti in pripadajoče uteži za dane vhodne podatke, pri premikanju nazaj pa prilagodimo uteži glede na napako, ki jo algoritem stori pri napovedovanju (Marsland, 2009; Sharma & Panigrahi, 2011).

Premikanje naprej po mreži opišemo s pomočjo prenosne funkcije g. Za posamezno statistično enoto i in posamezen nevron k v skritem sloju izračunamo njegovo vrednost kot

$$a_i^{(k)} = g\left(\sum_{j=0}^{n} X_i^{(j)} v_{jk}\right) \quad (18)$$

za vsak $i = 1, 2, \ldots, m$ in za vsak $k = 1, 2, \ldots, K$ (Marsland, 2009).

Enačbo (18) v vektorski obliki zapišemo kot

$$a_i = g(X_i v) \quad (19)$$

za vsak $i = 1, 2, \ldots, m$, kjer je $a_i \in R^{1xK}$ vektor vrednosti pripadajočih nevronov v skritem sloju za posamezno statistično enoto i (Bishop, 1995).

Z upoštevanjem izbire logistične funkcije za prenosno funkcijo enačbo (19) zapišemo kot

$$a_i = g(X_i v) = \frac{1}{1 + e^{-X_i v}} \quad (20)$$

za vsak $i = 1, 2, \ldots, m$ (Marsland, 2009).

Po izračunu vrednosti nevronov v skritem sloju na podoben način izračunamo še vrednost nevrona v izhodnem sloju. Izhodna verjetnost p_i, ki jo algoritem določi, je tako enaka

$$\begin{aligned} p_i &= g\left(\sum_{k=0}^{K} a_i^{(k)} w_k\right) \\ &= g(a_i w) \\ &= \frac{1}{1 + e^{-a_i w}} \end{aligned} \quad (21)$$

za vsak $i = 1, 2, \ldots, m$ (Bishop, 2006). Z izračunom vrednosti vseh nevronov v nevronski mreži je premikanje naprej po mreži končano. Sedaj si oglejmo še premikanje nazaj po mreži.

Najprej se spomnimo logistične regresije. Za izračun regresijskih koeficientov uporabimo metodo največjega verjetja, kjer maksimum funkcije največjega verjetja poiščemo s pomočjo enačbe (12). Podobno storimo pri nevronskih mrežah, kjer je za prenosno funkcijo

uporabljena logistična funkcija. Omenjeno enačbo prilagodimo za uporabo v nevronskih mrežah tako, da izhodno verjetnost, ki jo algoritem določi, definiramo z enačbo (21).

Maksimum naše funkcije verjetja poiščemo s pomočjo parcialnih odvodov enačbe

$$\begin{aligned} l(v,w) &= \sum_{i=1}^{m} \left(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right) \\ &= \sum_{i=1}^{m} y_i \log\left(\frac{p_i}{1 - p_i}\right) + \sum_{i=1}^{m} \log(1 - p_i) \\ &= \sum_{i=1}^{m} y_i a_i w - \sum_{i=1}^{m} \log(1 + e^{a_i w}) \end{aligned} \tag{22}$$

(Bishop, 1995).

Parcialni odvodi po utežeh, ki povezujejo skriti in izhodni sloj, so enaki

$$\begin{aligned} \frac{\partial l}{\partial w_k} &= \sum_{i=1}^{m} y_i a_i^{(k)} - \sum_{i=1}^{m} \frac{1}{1 + e^{a_i w}} e^{a_i w} a_i^{(k)} \\ &= \sum_{i=1}^{m} (y_i - p_i) a_i^{(k)} \end{aligned} \tag{23}$$

za vsak $k = 0, 1, \ldots, K$ (Ng, 2016; Rojas, 1996).

Parcialni odvodi po utežeh, ki povezujejo vhodni in skriti sloj pa so enaki

$$\begin{aligned} \frac{\partial l}{\partial v_{jk}} &= \frac{\partial l}{\partial a_k} \cdot \frac{\partial a_k}{\partial v_{jk}} \\ &= \sum_{i=1}^{m} \left(y_i w_k - \frac{1}{1 + e^{a_i w}} e^{a_i w} w_k\right)\left(\frac{-X_i^{(j)} e^{-X_i v_k}}{(1 + e^{-X_i v_k})^2}\right) \\ &= -\sum_{i=1}^{m} (y_i - p_i) w_k X_i^{(j)} a_i^{(k)} (1 - a_i^{(k)}) \end{aligned} \tag{24}$$

za vsak $k = 1, 2, \ldots, K$ in za vsak $j = 0, 1, \ldots, n$ (Ng, 2016; Rojas, 1996).

Za izračun optimalnih uteži rešimo enačbi

$$\frac{\partial l}{\partial w_k} = 0 \tag{25}$$

za vsak $k = 0, 1, \ldots, K$ in

$$\frac{\partial l}{\partial v_{jk}} = 0 \tag{26}$$

za vsak $k = 1, 2, \ldots, K$ ter za vsak $j = 0, 1, \ldots, n$ (Bishop, 1995).

Za izpeljavo zgoraj definiranih izračunov parcialnih odvodov funkcije največjega verjetja $l(v,w)$ uporabimo premikanje nazaj po mreži. V ta namen najprej definiramo napako izhodnega sloja nevronske mreže kot

$$\delta_i = (y_i - p_i) \qquad (27)$$

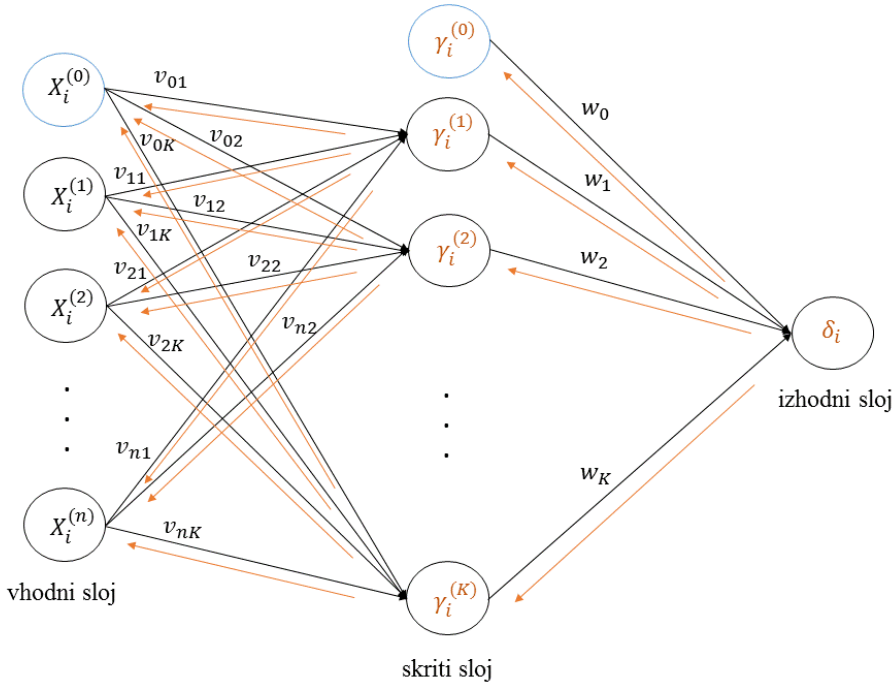
za vsak $i = 1, 2, \ldots, m$ in napake skritega sloja kot

$$\gamma_i^{(k)} = \delta_i w_k a_i^{(k)} (1 - a_i^{(k)}) \qquad (28)$$

za vsak $i = 1, 2, \ldots, m$ in za vsak $k = 0, 1, \ldots, K$ (Ng, 2016).

Za lažjo predstavo je premikanje nazaj po mreži prikazano na Sliki 5 z oranžno barvo.

Slika 5: Premikanje nazaj po večslojni usmerjeni nevronski mreži



Povzeto in prirejeno po A. Ng, Machine Learning, 2016.

Iz zgoraj definiranih napak izračunamo parcialne odvode uteži, ki povezujejo skriti in izhodni sloj kot

$$Dw_k = \sum_{i=1}^{m} \delta_i a_i^{(k)} \qquad (29)$$

za vsak $k = 0, 1, \ldots, K$ in parcialne odvode uteži, ki povezujejo vhodni in skriti sloj, kot

$$Dv_{jk} = \sum_{i=1}^{m} \gamma_i^{(k)} X_i^{(j)} \quad (30)$$

za vsak $k = 1, 2, \ldots, K$ in za vsak $j = 0, 1, \ldots, n$ (Ng, 2016).

Za izračun optimalnih uteži rešimo enačbi

$$Dw_k = 0 \quad (31)$$

za vsak $k = 0, 1, \ldots, K$ in

$$Dv_{jk} = 0 \quad (32)$$

za vsak $k = 1, 2, \ldots, K$ in za vsak $j = 0, 1, \ldots, n$. Enačbi rešimo numerično, kar lahko naredimo z različnimi programskimi jeziki in metodami. Najpogostejša metoda v vsaki iteraciji prilagodi uteži s pomočjo zgoraj izračunanih parcialnih odvodov z namenom zniževanja celotne napake algoritma. Po prilagoditvi uteži ponovno izračuna izhodno verjetnost, pripadajočo napako in parcialne odvode, ki jih ponovno uporabi za prilagoditev uteži. Postopek prilagajanja uteži ponavlja, dokler celotna napaka učne množice ni manjša od vnaprej definiranega kriterija (Bishop, 2006; Smith & Gupta, 2002).

Pri izračunu optimalnih uteži se moramo zavedati, da funkcija napake ni konveksna, zato se lahko zgodi, da namesto globalnega maksimuma najdemo lokalnega. Vendar Bishop (2006) pravi, da je za uspešno uporabo nevronskih mrež dovolj, da najdemo zadovoljivo dober lokalni maksimum.

2.4 Ocenjevanje uspešnosti algoritmov podatkovnega rudarjenja

Skoraj nemogoče je konstruirati algoritem, ki bo za vsako statistično enoto v naših vhodnih podatkih pravilno napovedal izhodni podatek. Za izbiro najprimernejšega algoritma je tako potrebno uporabiti različne mere za ocenjevanje njihove uspešnosti. Izbira različnih mer je odvisna predvsem od zastavljenega raziskovalnega vprašanja in podatkov, ki jih imamo na voljo (Vuk & Curk, 2006).

Pri klasifikacijskem problemu lahko uporabimo dve vrsti klasifikatorjev: binarnega (angl. *binary classifier*) ali verjetnostnega (angl. *probabilistic classifier*). Binarni klasifikator določi enega izmed vnaprej definiranih razredov, torej da bo stranka ostala ali da bo odšla, medtem ko verjetnostni klasifikator določi, s kakšno verjetnostjo stranka pripada enemu izmed vnaprej definiranih razredov, torej s kakšno verjetnostjo bo stranka ostala ali odšla (Burez & Van den Poel, 2009).

Pri uporabi binarnega klasifikatorja lahko vedno označimo en izhodni razred kot pozitiven (oziroma 1) in enega kot negativnega (oziroma 0). Pozitiven razred vsebuje *P* resnično pozitivnih statističnih enot naše testne množice, negativen pa *N* resnično negativnih. Klasifikator vsaki statistični enoti določi enega izmed razredov, vendar pri nekaterih statističnih enotah stori napako in ne določi pravilnega razreda. Za oceno uspešnosti klasifikatorja preštejemo število pravilno pozitivnih (angl. *true positive*), pravilno negativnih (angl. *true negative*), nepravilno pozitivnih (angl. *false positive*) in nepravilno negativnih (angl. *false negative*) klasificiranih statističnih enot (Vuk & Curk 2006).

Pravilno pozitivne so tiste statistične enote, ki so s klasifikatorjem uvrščene v pozitivni razred in so tudi resnično pozitivne. Število pravilno pozitivnih statističnih enot v nadaljevanju označimo s *TP*. Pravilno negativne statistične enote so tiste, ki so s klasifikatorjem uvrščene v negativni razred in so tudi resnično negativne. Število pravilno negativnih statističnih enot v nadaljevanju označimo s *TN*. Nepravilno pozitivne so tiste statistične enote, ki so s klasifikatorjem uvrščene v pozitivni razred, a so resnično negativne. Število nepravilno pozitivnih statističnih enot v nadaljevanju označimo s *FP*. Nepravilno negativne statistične enote so tiste, ki so s klasifikatorjem uvrščene v negativni razred, a so resnično pozitivne. Število nepravilno negativnih statističnih enot v nadaljevanju označimo s *FN* (Chawla, 2005).

Seštevek pravilno pozitivnih in nepravilno negativnih statističnih enot je enak vsem resnično pozitivnim statističnim enotam, $TP + FN = P$. Seštevek pravilno negativnih in nepravilno pozitivnih statističnih enot je enak vsem resnično negativnim statističnim enotam, $TN + FP = N$. Seštevek pravilno pozitivnih in nepravilno pozitivnih statističnih enot ($TP + FP$) predstavlja vse statistične enote, ki jih klasifikator klasificira za pozitivne, seštevek pravilno negativnih in nepravilno negativnih statističnih enot ($TN + FN$) pa vse statistične enote, ki jih klasifikator klasificira za negativne (Burez & Van den Poel, 2009).

Verjetnostni klasifikator je funkcija $f:X \rightarrow [0,1]$, ki vsaki statistični enoti X_i določi realno število $f(X_i)$ za vsak $i = 1, 2, ..., m$. Verjetnostni klasifikator se z uporabo mejne vrednosti t (angl. *threshold*) lahko enostavno spremeni v binarni klasifikator tako, da vse statistične enote, ki imajo napovedano verjetnost višjo od izbrane mejne vrednosti, tj. $f(X_i) \geq t$ za vsak $i = 1, 2, ..., m$, razvrstimo v pozitivni razred, ostale pa v negativnega. Izpeljani binarni klasifikator je funkcija mejne vrednosti. Na njem lahko uporabimo enake mere uspešnosti kot na običajnem binarnem klasifikatorju. S spreminjanjem mejne vrednosti verjetnostnega klasifikatorja dobimo družino binarnih klasifikatorjev (Vuk & Curk, 2006).

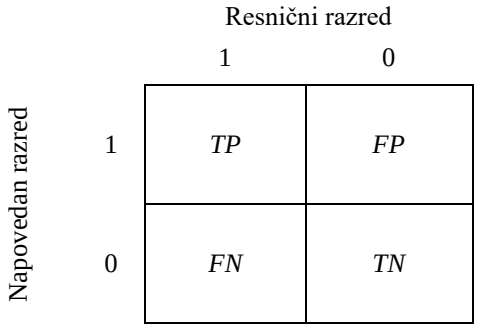
Za oceno binarnih klasifikatorjev se med drugim uporablja naslednje mere: klasifikacijsko matriko (angl. *confusion matrix*), natančnost (angl. *accuracy*), specifičnost (angl. *specificity*), mero nepravilne pozitivnosti (angl. *FPrate)*, občutljivost (angl. *sensitivity* ali *recall* ali *TPrate*), pozitivno napovedno vrednost (angl. *precision,* v nadaljevanju PNV) in F mero (angl. *F score*). Pri oceni družine binarnih klasifikatorjev pa sta dodatno uporabljeni

tudi ROC krivulja (angl. *Receiver Operating Characteristic*) in površina pod ROC krivuljo oz. AUC (angl. *Area Under the Curve*).

2.4.1 Klasifikacijska matrika

Klasifikacijska matrika je pri uporabi binarnega klasifikatorja velikosti 2 x 2, kjer njeni stolpci predstavljajo resnične razrede, vrstice pa napovedane razrede s strani klasifikatorja.

Slika 6: Klasifikacijska matrika



Povzeto in prirejeno po A. Ng, Machine Learning, 2016.

Iz klasifikacijske matrike so izpeljane vse ostale mere za ocenjevanje uspešnosti binarnih klasifikatorjev, ki sledijo v nadaljevanju (Chen, Liaw, & Breiman, 2004).

2.4.2 Natančnost

Natančnost je definirana kot

$$natančnost = \frac{TP + TN}{P + N} \quad (33)$$

in meri odstotek vseh pravilno klasificiranih statističnih enot. Zavzema lahko vrednosti med 0 in 1, kjer višja vrednost pomeni večjo natančnost. Natančnost je ena izmed pogosto uporabljenih mer kvalitete binarnih klasifikatorjev. Uporabljena je tudi za izračun klasifikacijske napake (angl. *misclassification error*), saj je

$$klasifikacijska\ napaka = 1 - natančnost \quad (34)$$

(Amin, Shehzad, Khan, Ali, & Anwar, 2015).

2.4.3 Specifičnost in mera nepravilne pozitivnosti

Specifičnost je definirana kot

$$specifičnost = \frac{TN}{N} \quad (35)$$

iz česar sledi

$$1 - specifičnost = \frac{FP}{N} = mera\ nepravilne\ pozitivnosti \quad (36)$$

Specifičnost meri odstotek pravilno klasificiranih resnično negativnih statističnih enot in zavzema vrednosti med 0 in 1, kjer višja vrednost pomeni boljšo sposobnost klasifikatorja za določanje negativnega razreda. Mera nepravilne pozitivnosti pa meri odstotek resnično negativnih statističnih enot, ki so klasificirane kot pozitivne statistične enote. Zavzema vrednosti med 0 in 1, kjer je boljši klasifikator z nižjo vrednostjo mere nepravilne pozitivnosti (Galar, Fernandez, Barrenechea, Bustince, & Herrera, 2012).

2.4.4 Občutljivost

Občutljivost je definirana kot

$$občutljivost = \frac{TP}{P} \quad (37)$$

in meri odstotek pravilno klasificiranih resnično pozitivnih statističnih enot. Pogosto je uporabljena za oceno kvalitete binarnih klasifikatorjev. Zavzema lahko vrednosti med 0 in 1, kjer višja vrednost pomeni boljšo sposobnost klasifikatorja za določanje pozitivnega razreda (Clemente et al., b.l.).

2.4.5 PNV

PNV je definirana kot

$$PNV = \frac{TP}{TP + FP} \quad (38)$$

in meri odstotek pravilno klasificiranih pozitivno klasificiranih statističnih enot. Je ena izmed pogosto uporabljenih mer za oceno kvalitete binarnih klasifikatorjev. Zavzema lahko vrednosti med 0 in 1, kjer je boljši klasifikator z višjo vrednostjo PNV (Lopez, Fernandez, Garcia, Palade, & Herrera, 2013).

2.4.6 F mera

F mera nam omogoča združiti PNV in občutljivost v eno samo mero. Z njo lahko primerjamo algoritme z različnimi PNV in občutljivostmi. Zamislimo si, da imamo tri algoritme z izračunano PNV in občutljivostjo kot v Tabeli 10 (Ng, 2016).

Tabela 10: PNV in občutljivost za izbrane algoritme

	PNV	Občutljivost
Algoritem 1	0,5	0,4
Algoritem 2	0,7	0,1
Algoritem 3	0,02	1,0

Vir: A. Ng, Machine Learning, 2016.

Odločitev, kateri algoritem je v tem primeru najboljši, ni trivialna, zato si pomagamo s F mero.

F mera je definirana kot

$$F\ mera = 2\frac{PNV \cdot občutljivost}{PNV + občutljivost} \quad (39)$$

in lahko zavzame katerokoli vrednost na intervalu $[0, 1]$, pri čemer je boljši klasifikator tisti, ki zavzame višjo vrednost (Shaza & Abraham, 2013).

Z uporabo F mere izračunamo, da je v Tabeli 10 najboljši algoritem 1. Rezultat je razviden iz Tabele 11.

Tabela 11: F mera za izbrane algoritme

	F mera
Algoritem 1	0,44
Algoritem 2	0,18
Algoritem 3	0,04

2.4.7 ABC mera

Poleg že znanih mer za ocenjevanje uspešnosti izberemo tudi podobno mero, kot so jo Au, Li in Ma (2003) uporabili pri izbiri optimalne mejne vrednosti. Poimenujemo jo ABC mera. Več podrobnosti o ABC meri sledi v poglavju 3.4, kjer jo uporabimo tudi za izračun optimalne mejne vrednosti.

ABC mera je definirana kot

$$ABC\ mera = \frac{TN}{N} - \frac{FN}{P} \quad (40)$$

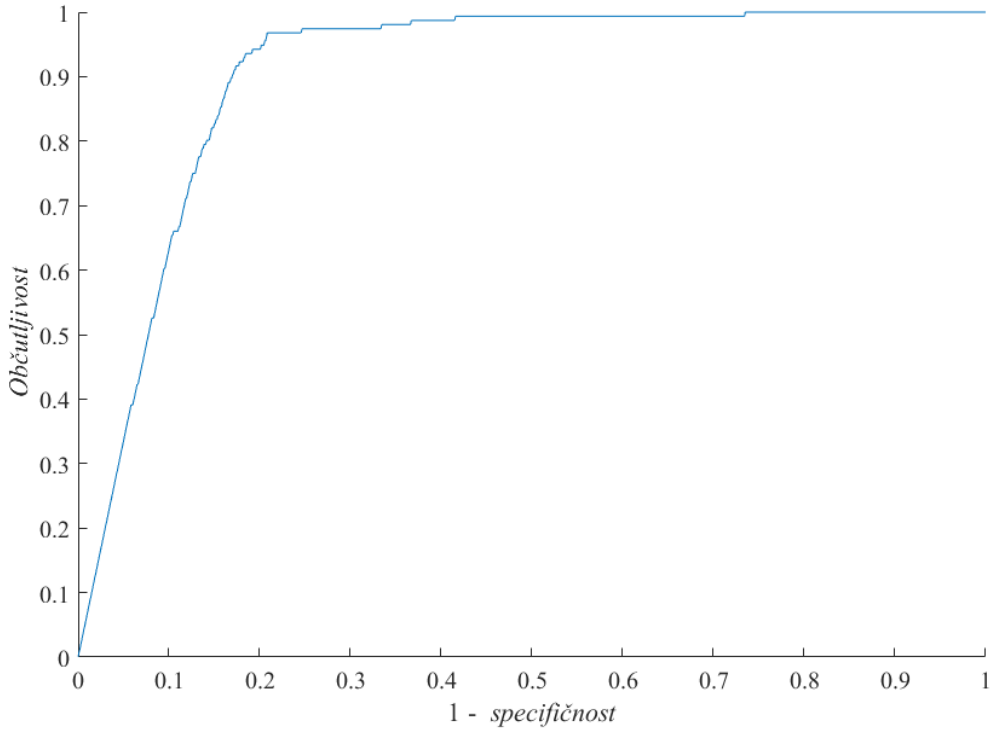
in meri število pravilno klasificiranih statističnih enot med resnično negativnimi statističnimi enotami, kateremu odšteje število nepravilno klasificiranih statističnih enot med resnično

pozitivnimi statističnimi enotami. Zavzame lahko katerokoli vrednost na intervalu [-1, 1], pri čemer je boljši klasifikator tisti, ki zavzame višjo vrednost.

2.4.8 ROC krivulja

Ko želimo oceniti natančnost verjetnostnega klasifikatorja neodvisno od izbrane mejne vrednosti, uporabimo ROC krivuljo. Graf, na katerem prikažemo ROC krivuljo, je parametrično definiran z *x = 1 - specifičnost* in *y = občutljivost*. S spreminjanjem mejne vrednosti verjetnostnega klasifikatorja dobimo družino binarnih klasifikatorjev, ki so na omenjenem grafu predstavljeni s točkami *x = 1 - specifičnost(t)* in *y = občutljivost(t)* in skupaj tvorijo ROC krivuljo. Primer ROC krivulje je prikazan na Sliki 7 (Chawla, 2005).

Slika 7: ROC krivulja



Povzeto in prirejeno po A. Ng, Machine Learning, 2016.

Idealna točka na ROC krivulji je (0, 1), ki predstavlja primer, kjer so vse resnično pozitivne statistične enote klasificirane pravilno in niti ena resnično negativna statistična enota ni klasificirana kot pozitivna (Chawla, 2005).

ROC krivulja je neodvisna od razmerja resnično pozitivnih in resnično negativnih statističnih enot v testni množici in je tako primerna za primerjavo klasifikatorjev, kjer se to razmerje spreminja (Burez & Van den Poel, 2009).

2.4.9 AUC

Območje pod ROC krivuljo imenujemo tudi AUC in je pogosta mera kvalitete verjetnostnega klasifikatorja. Izračuna se z naslednjo formulo:

$$AUC = \int_0^1 \frac{TP}{P} d\frac{FP}{N} = \frac{1}{P \cdot N} \int_0^N TP dFP \qquad (41)$$

AUC ima zalogo vrednosti realnih števil med 0 in 1, kjer ima naključni klasifikator (klasifikator, ki klasificira na podlagi meta kovanca) AUC = 0,5 in popolni klasifikator AUC = 1. V praksi naj bi bili klasifikatorji nekje med tema dvema vrednostima, boljše je, če je ta bližje vrednosti 1 (Vuk & Curk, 2006).

AUC nam pove, kakšna je verjetnost, da ima naključno izbrana pozitivna statistična enota napovedano višjo verjetnost kot naključno izbrana negativna statistična enota. Pokaže nam, kako dobro klasifikator loči pozitivne statistične enote od vseh statističnih enot (Vuk & Curk, 2006).

2.4.10 Izbira mer za ocenjevanje uspešnosti

Pri modeliranju algoritma za napoved odhajanja strank se pogosto srečamo s problemom razredne neuravnoteženosti (angl. *class imbalance*), kjer je število statističnih enot v pozitivnem razredu v primerjavi s številom statističnih enot v negativnem razredu majhno. Ena izmed metod za ravnanje z razredno neuravnoteženostjo je uporaba primernih mer za ocenjevanje uspešnosti algoritmov (Weiss, 2004).

Ena izmed primernih mer za ocenjevanje uspešnosti algoritmov je uporaba ROC krivulje namesto bolj razširjene AUC mere. AUC mera vrednoti oba razreda enakovredno in je v primeru razredne neuravnoteženosti pristranska (Burez & Van den Poel, 2009).

Primernejša je tudi uporaba občutljivosti in PNV namesto bolj razširjene natančnosti. Uporaba natančnosti ni primerna, saj ima manj zastopan pozitivni razred manj vpliva na natančnost kot bolj zastopan negativni razred. Zamislimo si, da imamo v naših vhodnih podatkih razmerje med negativnim in pozitivnim razredom enako 90 : 10 v prid negativnega razreda. V tem primeru natančnost klasifikatorja za negativne statistične enote šteje devetkrat toliko kot natančnost za pozitivne statistične enote, kar klasifikatorju oteži dobro klasificiranje pozitivnih statističnih enot. Namesto občutljivosti in PNV v našem primeru uporabimo F mero, saj je z njo primerjava algoritmov lažja (Weiss, 2004).

Namesto specifičnosti uporabimo ABC mero, katere sestavni del je tudi izračun specifičnosti.

Poleg uporabe primernih mer za ocenjevanje uspešnosti algoritmov so pri problemu razredne neuravnoteženosti možne tudi druge rešitve. Zelo pogosta rešitev je vzorčenje (angl. *sampling*), s katerim želimo minimizirati ali celo izločiti razredno neuravnoteženost. Vzorčenje delimo na podvzorčenje (angl. *under sampling*), kjer izločimo statistične enote bolj zastopanega razreda, in na nadvzorčenje (angl. *over sampling*), kjer statistične enote manj zastopanega razreda podvojimo, potrojimo,... (Guo, Yin, Dong, Yang, & Zhou, 2008).

Pri podvzorčenju lahko z odstranitvijo določenih statističnih enot iz naših vhodnih podatkov odstranimo tudi pomembne informacije, ki jih te statistične enote vsebujejo. Posledično to lahko pripelje do konstrukcije slabšega klasifikatorja (Guo et al., 2008).

Pri nadvzorčenju v vhodne podatke pogosto dodamo kopije že obstoječih statističnih enot, kar pomeni, da algoritmu dejansko ne predstavimo nobenih novih informacij. Poleg tega lahko nadvzorčenje pripelje do problema pretiranega prilagajanja (Chen et al., 2004).

V literaturi pripisujejo boljše rezultate uporabi podvzorčenja (Chen et al., 2004), zato bi se teoretično tudi mi odločili za podvzorčenje, vendar se zaradi pomanjkanja statističnih enot v naših vhodnih podatkih pri naslovitvi problema razredne neuravnoteženosti odločimo le za uporabo primernejših mer za ocenjevanje uspešnosti.

3 IZVEDBA PROCESA PODATKOVNEGA RUDARJENJA

3.1 Definicija poslovnega problema

V Sloveniji je na voljo veliko mobilnih aplikacij, spletnih bank in fizičnih mest za plačevanje položnic ter nakazovanje denarnih sredstev, zato je konkurenca naši izbrani aplikaciji velika.

Zaradi velike konkurence, novih in inovativnih poslovnih modelov ter vedno boljših storitev je pridobivanje novih uporabnikov vedno dražje (Lazarov & Capota, 2007). Za dobičkonosnost naše izbrane aplikacije Hal mBills je zato pomembno ne le privabljati nove uporabnike, temveč tudi obdržati obstoječe.

Van den Poel in Lavriviere (2004) povzemata naslednje koristi obdržanja obstoječih uporabnikov:

- zmanjšanje potrebe po iskanju novih in potencialno tveganih uporabnikov,
- poglobljeno poznavanje potreb že obstoječih uporabnikov,
- dolgoročni uporabniki zapravljajo več,
- dolgoročni uporabniki, ki so zadovoljni s storitvijo, lahko z dobro besedo privabijo nove uporabnike,
- dolgoročni uporabniki so manj občutljivi na marketinške aktivnosti konkurence,

- izguba obstoječih uporabnikov pripelje do manjše prodaje in povečane potrebe po privabljanju novih uporabnikov, kar je pet do šestkrat dražje od aktivnosti, namenjenih obdržanju obstoječih uporabnikov.

Shaaban, Helmy, Khedr in Nasr (2012) odhajanje uporabnikov delijo na:

- **neprostovoljni odhod:** ko ponudnik storitve prekine pogodbo z uporabnikom,
- **prostovoljni odhod**, ki se dalje deli na:
  - **načrtni odhod:** uporabnik se odloči za menjavo ponudnika storitve. Razlogi za menjavo so lahko želja po uporabi nove tehnologije, cenejše ali kvalitetnejše storitve, različni sociološki ali psihološki faktorji in priročnost,
  - **naključni odhod:** uporabnik prekine pogodbo brez odhoda h konkurenci. To stori zaradi določene spremembe v življenju, kot so na primer selitve, ali finančnih problemov.

V idealnem primeru bi se pri analizi odhajanja uporabnikov osredotočili na tiste, ki so našo izbrano mobilno aplikacijo načrtno prenehali uporabljati, saj lahko na take uporabnike vplivamo z različnimi marketinškimi pristopi. Vendar pa ločitev načrtnega in naključnega prenehanja uporabe v našem primeru ni izvedljiva, saj nimamo podatkov o razlogu prenehanja uporabe posameznega uporabnika. Po besedah Hadden, Tiwari, Roy in Ruta (2008) naključni odhod predstavlja zelo majhen del prostovoljnega odhoda, zato vključitev tega v našo analizo ne predstavlja velikega problema. V našo analizo smo tako zajeli vse uporabnike, ki so prostovoljno prenehali uporabljati izbrano mobilno aplikacijo, ne glede na to, ali sodijo v načrtno ali naključno skupino odhodov.

Lazarov in Capota (2007) odhajanje uporabnikov delita tudi na:

- **popolni odhod:** kjer uporabnik uradno prekine pogodbo,
- **skriti odhod:** kjer pogodba ni uradno prekinjena, vendar uporabnik storitve že dlje časa ne uporablja več aktivno,
- **delni odhod:** kjer uporabnik delno uporablja našo storitev in delno konkurenčno storitev.

Pri analizi prenehanja uporabe naše izbrane mobilne aplikacije napovedujemo odhajanje uporabnikov, ki spadajo ali v kategorijo skriti odhod ali v kategorijo delni odhod ob pogoju, da je delna uporaba izbrane mobilne aplikacije nižja od postavljenega kriterija. Uporabnike, ki spadajo v kategorijo popolni odhod, zaradi pomanjkljivih podatkov v našo analizo ne vključimo.

Za stroškovno učinkovito marketinško kampanijo, namenjeno ohranitvi obstoječih uporabnikov, je pomembno vedeti, na katere uporabnike se osredotočiti, saj z izborom ciljnih

uporabnikov prihranimo pri stroških izvedbe kampanije. Pomembno je vedeti, kateri so tisti uporabniki, ki so tik pred tem, da prenehajo uporabljati aplikacijo (Ferle, 2010).

Za napovedovanje ciljnih uporabnikov uporabimo logistično regresijo in nevronske mreže. Vprašanje, na katerega želimo odgovoriti, se glasi: »S kakšno verjetnostjo bo uporabnik prenehal uporabljati izbrano mobilno aplikacijo?«

Odgovor na vprašanje je tabela uporabnikov, razvrščenih po verjetnosti prenehanja uporabe izbrane mobilne aplikacije. Glede na predvidene stroške marketinške kampanije, življenjske vrednosti posameznega uporabnika in verjetnosti prenehanja uporabe, bodo izbrani uporabniki deležni različnih marketinških pristopov z namenom ponovnega prepričanja v uporabo izbrane mobilne aplikacije.

3.2 Opis podatkov

Podatke zaradi poslovne skrivnosti v nadaljevanju razkrivamo omejeno. Podatki, uporabljeni pri napovedovanju, so interni anonimizirani podatki, pridobljeni s strani lastnika izbrane mobilne aplikacije, in se nanašajo na obdobje od 9. novembra 2015 do 31. julija 2016. Imamo dve vrsti pridobljenih podatkov, in sicer podatke, ki zajemajo osnovne lastnosti uporabnikov, ter vsakodnevne zapise njihove uporabe izbrane mobilne aplikacije.

Podatki, ki zajemajo osnovne lastnosti uporabnikov, so med drugim sestavljeni iz osebnih podatkov uporabnika, njegovega statusa (ali je zahteval odjavo in izbris iz sistema), operacijskega sistema mobilne naprave, na kateri uporablja izbrano mobilno aplikacijo, ter datuma in načina registracije (Hal mBills ali Hal mBills Lite).

V kolikor ima uporabnik izbran način registracije Hal mBills, so zanj na voljo tudi podatki o datumu, načinu in statusu podpisa pogodbe ter oznaka banke, katere transakcijski račun ima povezan z mobilno denarnico.

O uporabnikih, ki so zahtevali odjavo in izbris iz sistema, imamo na voljo zelo malo podatkov.

Seznam vseh pridobljenih podatkov, ki zajemajo osnovne lastnosti uporabnikov, je predstavljen v Tabeli 12.

Tabela 12: Oznake in opisi podatkov, ki zajemajo osnovne lastnosti uporabnika v izbrani mobilni aplikaciji

Oznaka podatka	**Opis podatka**
CLIENT_ID	Anonimna identifikacijska oznaka posameznega uporabnika
CLIENT_STATUS	Zavzema dve možni vrednosti: • ACTIVE, v kolikor je uporabnik aktiven, • REMOVED, v kolikor je uporabnik zahteval odjavo in izbris iz sistema
DATE_OF_CLIENT_STATUS_CHANGE	Datum zadnje spremembe
EMAIL	Anonimna domena naslova spletne pošte
BIRTH_DATE	Datum rojstva
CITY	Poštna številka in pošta stalnega prebivališča
REGISTRATION_TIME	Datum registracije v izbrano mobilno aplikacijo
AUTHENTICATION_TIME	Datum podpisa pogodbe
REGISTRATION_STATUS	Zavzema dve možni vrednosti: • AUTHENTICATED, v kolikor je uporabnik podpisal pogodbo, • NOT AUTHENTICATED, v kolikor uporabnik še ni podpisal pogodbe
MANDATE_STATUS	Zavzema tri možne vrednosti: • FULL - Contract signed, v kolikor uporabnik uporablja različico Hal mBills in je že podpisal pogodbo, • FULL - Contract not signed (yet), v kolikor uporabnik uporablja različico Hal mBills in še ni podpisal pogodbe, • LITE, v kolikor uporabnik uporablja različico Hal mBills Lite
CONTRACT_SIGNATURE	Način podpisa pogodbe zavzema tri možne vrednosti: • DPD, če je bila pogodba v podpis dostavljena s kurirjem, • In person, če se je uporabnik za podpis pogodbe osebno zglasil na sedežu podjetja, • Digital certificate, v kolikor je bila pogodba podpisana z digitalnim potrdilom
CLIENTS_BANK	Anonimna oznaka banke
OS_TYPE	Operacijski sistem mobilne naprave

Podatki o vsakodnevnih zapisih uporabe izbrane mobilne aplikacije so med drugim sestavljeni iz podatkov o prenosih denarnih sredstev nazaj na transakcijski račun, plačilih avtomatsko prejetih položnic v aplikacijo ter položnic, plačanih s slikanjem, nakazilih denarnih sredstev drugim uporabnikom in polnjenju mobilne denarnice.

Vsakodnevni zapisi uporabe izbrane mobilne aplikacije se nahajajo v različnih tabelah. Imena tabel in podatkov, ki jih te vsebujejo, so razvidna iz Tabele 13.

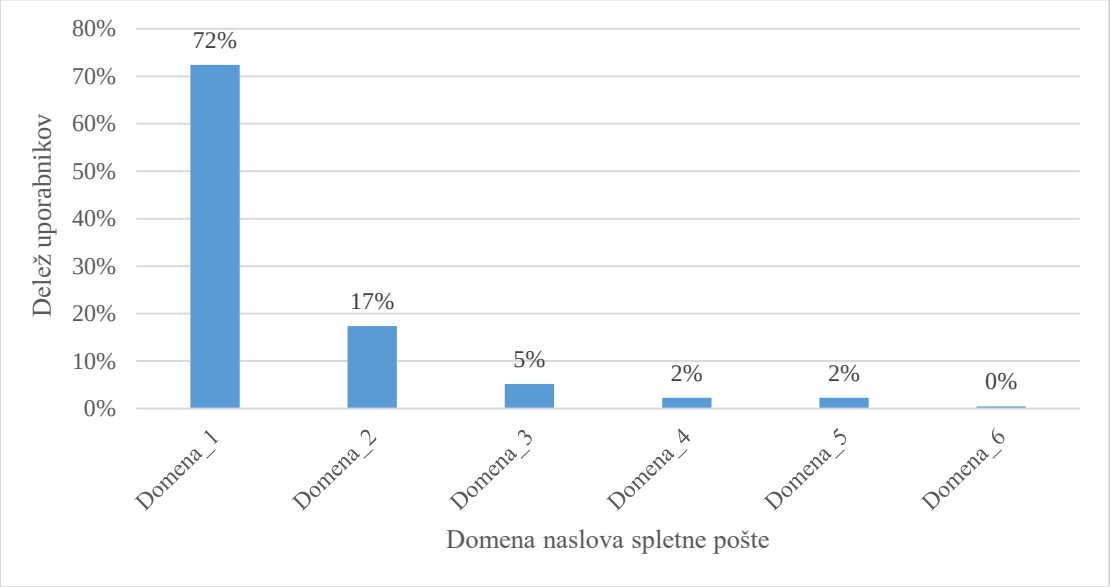
Tabela 13: Oznake tabel in opisi podatkov, ki vsebujejo vsakodnevne zapise uporabe izbrane mobilne aplikacije

Oznaka tabele	**Opis podatkov**
CASHOUT	Datum, ura in znesek prenosa denarnih sredstev nazaj na transakcijski račun
EINVOICE_payments_from_wallet	Anonimni naziv izdajatelja položnice, datum, ura in znesek plačila avtomatsko prejete položnice v aplikacijo z denarnimi sredstvi v mobilni denarnici
EINVOICE_payments_with_topup	Anonimni naziv izdajatelja položnice, datum, ura in znesek plačila avtomatsko prejete položnice v aplikacijo iz transakcijskega računa z direktno bremenitvijo
P2P_payments	Datum, ura, znesek in prejemnik nakazila denarnih sredstev drugemu uporabniku
PHOTOPAY_payments_from_wallet	Anonimni naziv izdajatelja položnice, datum, ura in znesek plačila položnice s slikanjem z denarnimi sredstvi v mobilni denarnici
PHOTOPAY_payments_with_topup	Anonimni naziv izdajatelja položnice, datum, ura in znesek plačila položnice s slikanjem iz transakcijskega računa z direktno bremenitvijo
SUBSCRIPTIONS_v_imenu_drugega	Anonimni naziv izdajatelja položnice, na katerega avtomatsko prejemanje položnic v aplikacijo je uporabnik naročen v tujem imenu
SUBSCRIPTIONS_v_svojem_imenu	Anonimni naziv izdajatelja položnice, na katerega avtomatsko prejemanje položnic v aplikacijo je uporabnik naročen
TOPUP_for_PAYMENT	Datum, ura in znesek polnjenja mobilne denarnice zaradi plačila položnice s transakcijskega računa z direktno bremenitvijo
TOPUP_with_CT	Datum, ura in znesek polnjenja mobilne denarnice z UPN
TOPUP_with_DD	Datum, ura in znesek polnjenja mobilne denarnice preko direktne bremenitve

Za vhodne podatke napovedovanja prenehanja uporabe izbrane mobilne aplikacije vzamemo uporabnike, ki so se registrirali do vključno 31. marca 2016. Izmed njih izločimo uporabnike, ki so zahtevali odjavo in izbris iz sistema (CLIENT_STATUS = REMOVED), ter uporabnike, ki uporabljajo različico Hal mBills Lite (MANDATE_STATUS=LITE), saj o njih nimamo na voljo dovolj podatkov. Tako dobimo izbrane uporabnike za izvedbo naše analize. V nadaljevanju nekatere podatke, ki zajemajo osnovne lastnosti izbranih uporabnikov, prikažemo grafično.

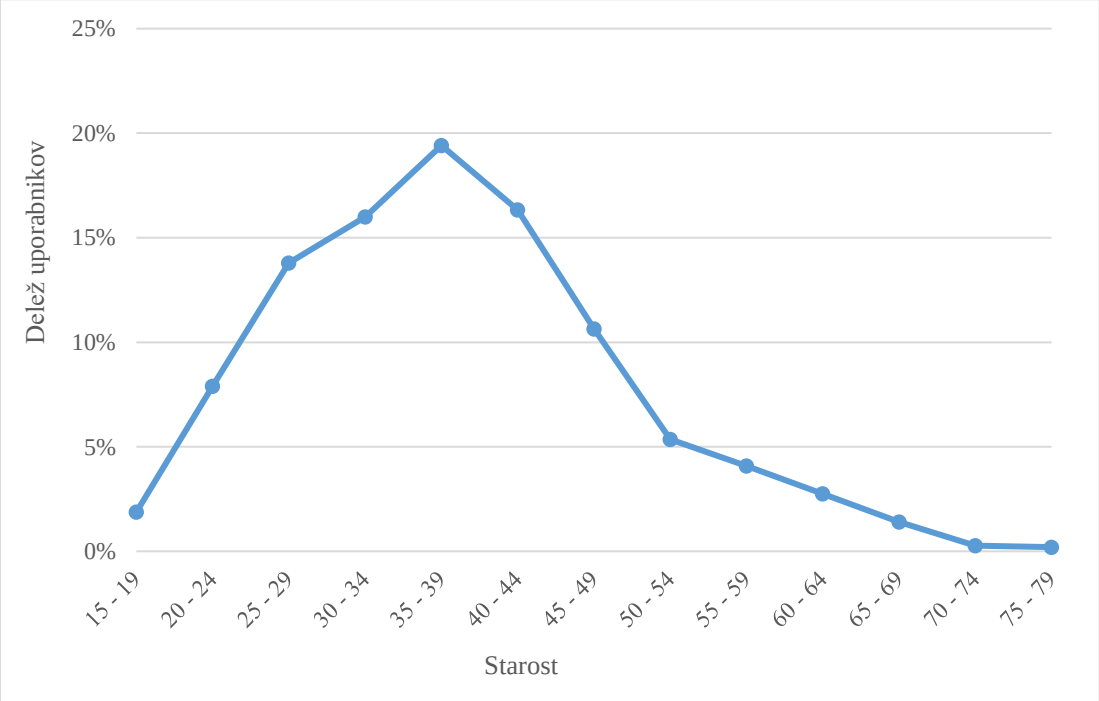
Slika 8 prikazuje, kakšen delež uporabnikov izbrane mobilne aplikacije se je v slednjo registriralo z naslovom spletne pošte iz posamezne domene. Kar 72 % uporabnikov se je v izbrano mobilno aplikacijo registriralo iz Domena_1, s 17 % ji sledi Domena_2, ostale domene so zastopane v manjšini.

Slika 8: Domene naslova spletne pošte izbranih uporabnikov



Podatke o datumih rojstva posameznih uporabnikov pretvorimo v starosti na dan 31. marec 2016 in jih prikažemo na Sliki 9. Izbrano mobilno aplikacijo uporabljajo prebivalci Slovenije od 15. do 79. leta starosti, med katerimi jih je največ starih 37 let.

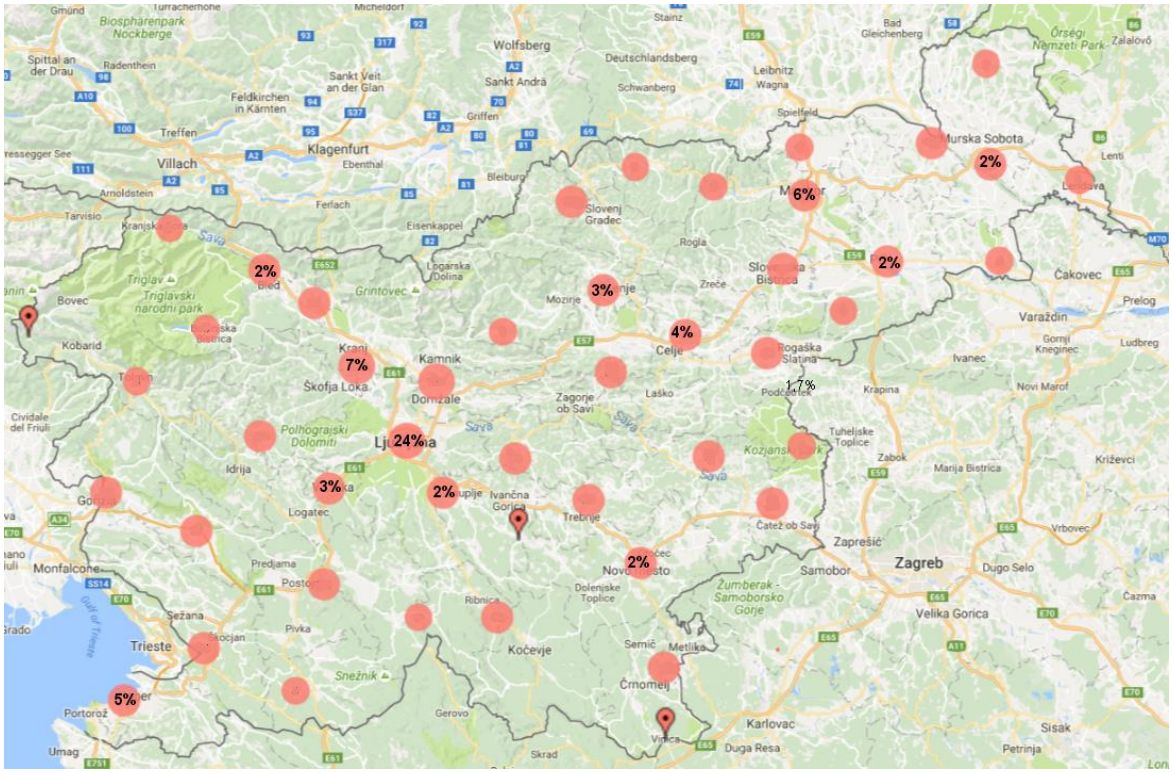
Slika 9: Starostna sestava uporabnikov izbrane mobilne aplikacije



Uporabnike, iz katerih sestavimo tabelo vhodnih podatkov, glede na poštno številko in pošto stalnega prebivališča razvrstimo na zemljevid Slovenije, ki je prikazan na Sliki 10. Rdeče

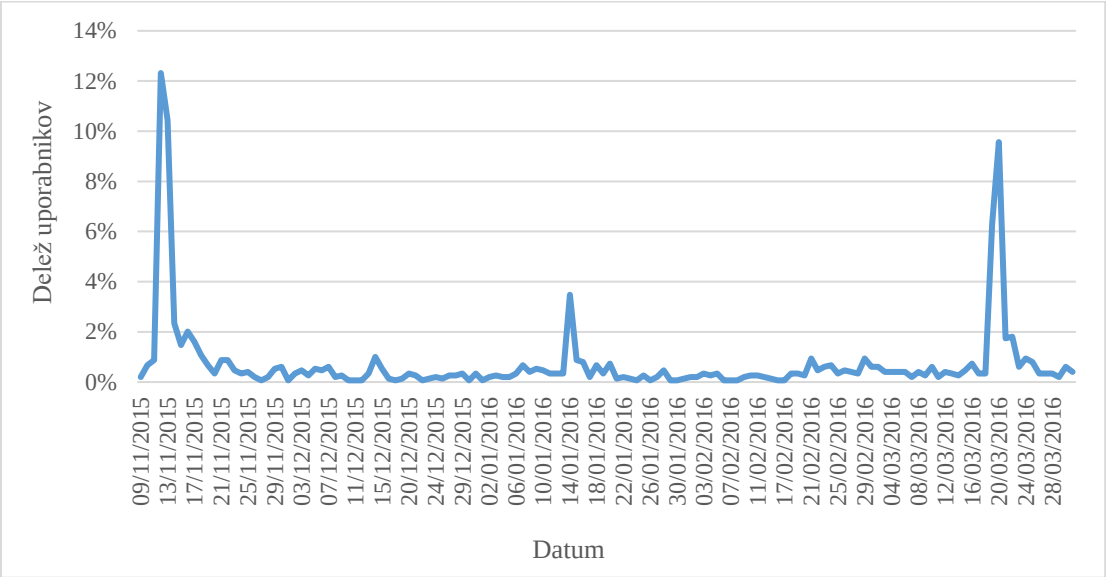
pike na zemljevidu označujejo lokacijo stalnega prebivališča uporabnikov. Lokacije, kjer število uporabnikov presega 2 % vseh izbranih uporabnikov, imajo na zemljevidu poleg lokacije zapisan tudi delež uporabnikov s stalnim prebivališčem na tisti lokaciji. Uporabniki izbrane mobilne aplikacije prihajajo iz celotne Slovenije. Kar 44 % uporabnikov prihaja iz Ljubljanske regije (poštna številka med 1000 in 1434), 36 % uporabnikov je koncentriranih v sedmih največjih slovenskih mestih Ljubljana, Maribor, Celje, Kranj, Koper, Velenje in Novo mesto.

Slika 10: Lokacija stalnega prebivališča uporabnikov izbrane mobilne aplikacije



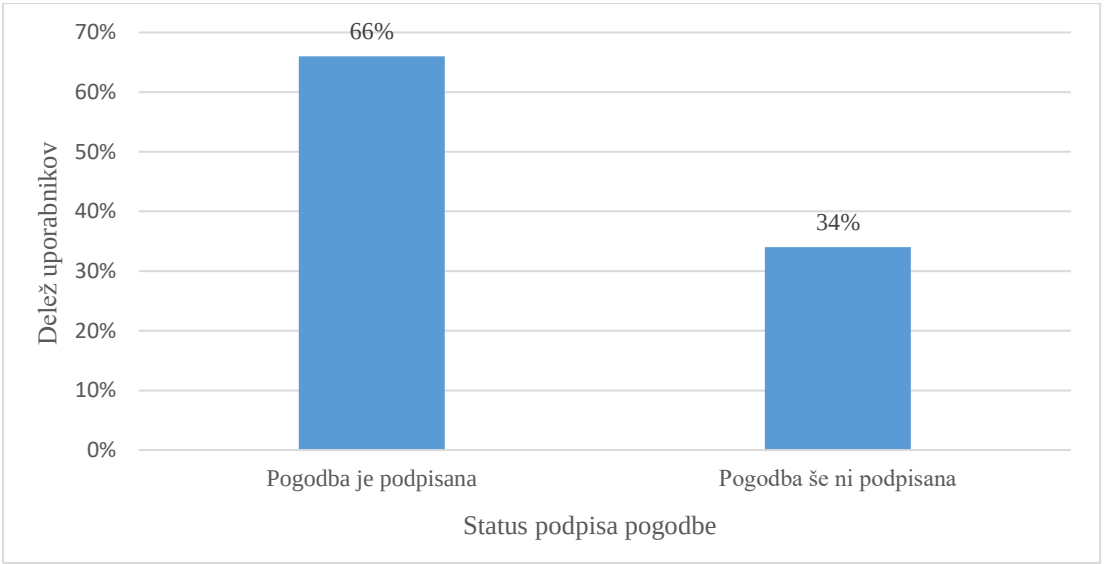
Slika 11 prikazuje, kdaj so se uporabniki registrirali v izbrano mobilno aplikacijo. V veliki večini se je na dan registriralo do 1 % novih uporabnikov, ki zadostujejo kriterijem za vključitev v našo analizo. Izjema so dnevi med 12. in 18. novembrom 2015, 14. januar 2016 in dnevi med 19. in 22. marcem 2016. Razlogi za registracijo več kot 31 % novih uporabnikov med 12. in 18. novembrom 2015 so prav gotovo medijske objave o pričetku dostopnosti izbrane mobilne aplikacije. Razlogi za registracijo več kot 3 % uporabnikov dne 14. januarja 2016 so prav tako medijske objave o izbrani mobilni aplikaciji, za registracijo 19 % novih uporabnikov med 19. in 22. marcem 2016 pa oglasi med prenosom skokov iz Planice.

Slika 11: Datum registracije uporabnikov v izbrano mobilno aplikacijo



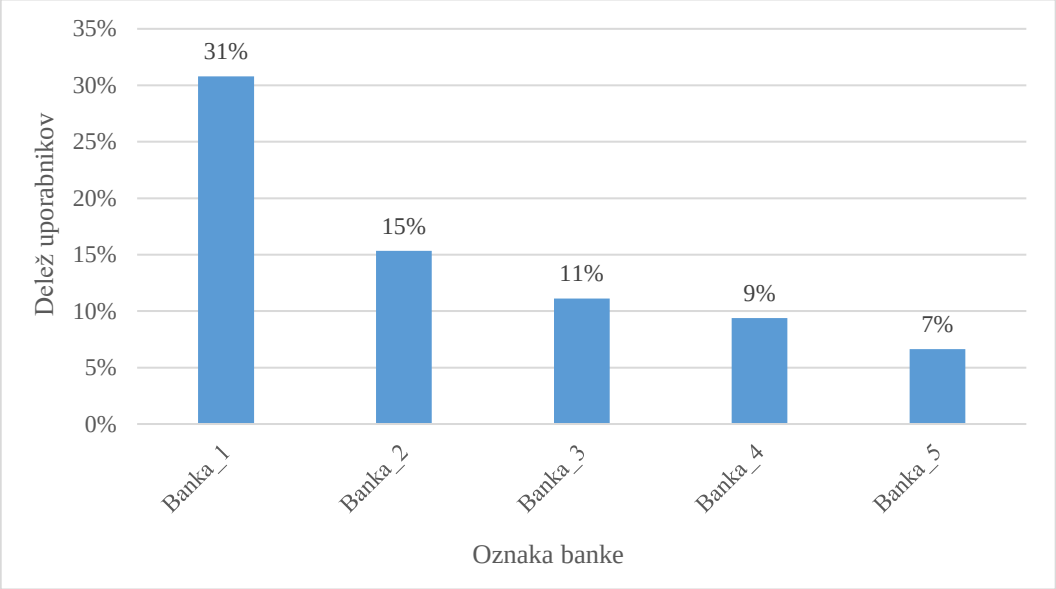
Pregled stanja podpisa pogodbe je razviden iz Slike 12. Kar 66 % uporabnikov je do 31. marca 2016 pogodbo že podpisalo.

Slika 12: Stanje podpisa pogodbe uporabnikov izbrane mobilne aplikacije



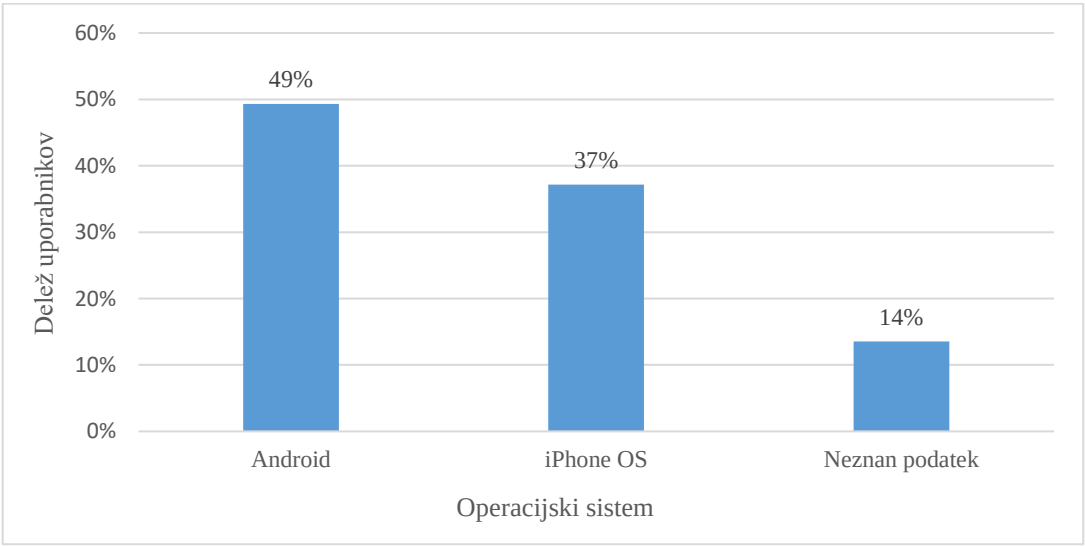
Uporabniki izbrane mobilne aplikacije imajo transakcijski račun, s katerim so se prijavili v izbrano mobilno aplikacijo, odprt pri šestnajstih različnih bankah in hranilnicah v Sloveniji. Kot je razvidno iz Slike 13, ima kar 31 % uporabnikov svoj transakcijski račun odprt pri banki Banka_1, kateri v razponu med 15 % in 6 % sledijo še štiri banke. Ostale banke, kjer število odprtih transakcijskih računov ne presega 6 % uporabnikov, grafično niso prikazane.

Slika 13: Oznaka banke uporabnikov izbrane mobilne aplikacije



Iz Slike 14 je razvidno, koliko uporabnikov si je izbrano mobilno aplikacijo namestilo na določen operacijski sistem. Za 14 % uporabnikov je ta podatek neznan, 49 % uporabnikov si je izbrano mobilno aplikacijo namestilo na mobilno napravo z operacijskim sistemom Android, 37 % pa na mobilno napravo z operacijskim sistemom iPhone OS.

Slika 14: Operacijski sistem mobilne naprave, na katero so si uporabniki namestili izbrano mobilno aplikacijo



3.3 Priprava podatkov

Pri podatkovnem rudarjenju je zelo pomembna priprava vhodnih podatkov. Vhodne podatke je potrebno pripraviti v obliki tabele (Witten et al., 2011).

Pridobljene podatke najprej pregledamo za obstoj nekonsistentnosti in manjkajočih vrednosti. Za lažjo predstavo jih izrišemo tudi grafično, kar prikažemo v poglavju 3.2. Po spoznanju s podatki se odločimo, katere uporabnike vzamemo v našo analizo, na katero časovno obdobje vsakodnevnih zapisov njihove uporabe se osredotočimo, kako ter katere atribute kreiramo in kako določimo izhodni podatek, ki ga želimo napovedati.

Kot omenjeno v poglavju 3.2, iz pridobljenih podatkov vzamemo uporabnike, ki so se registrirali do vključno 31. marca 2016 in izmed njih izločimo tiste, ki so zahtevali odjavo in izbris iz sistema, ter tiste, ki uporabljajo različico Hal mBills Lite. Za kreiranje atributov uporabimo podatke, ki zajemajo njihove osnovne lastnosti ter vsakodnevne zapise njihove uporabe v mesecu aprilu in maju.

Najprej kreiramo atribute iz podatkov, ki zajemajo osnovne lastnosti uporabnikov. Iz opisnih podatkov, kot so domena naslova spletne pošte, poštna številka in pošta stalnega prebivališča, status registracije ter oznaka banke, ustvarimo slamnate atribute, kjer za *z* različnih vrednosti posameznega podatka ustvarimo $z - 1$ atributov (Hosmer et al., 2013). Npr. za podatek o oznaki banke uporabnika imamo 16 možnih vrednosti omenjenega podatka. Večina vrednosti je zastopana v manjšini, zato te združimo v skupno vrednost, imenovano Ostalo. Tako nam ostane šest različnih vrednosti. Te so Banka_1, Banka_2, Banka_3, Banka_4, Banka_5 in Ostalo. Za podatek o oznaki banke uporabnika tako kreiramo pet slamnatih atributov, ki zavzamejo vrednosti 0 ali 1.

Tabela 14 prikazuje atribute, kreirane iz danih podatkov o osnovnih lastnostih uporabnikov. Iz vseh ostalih podatkov, ki v tabeli niso prikazani, se atributov iz različnih razlogov ne da kreirati.

Tabela 14: Atributi, kreirani iz danih podatkov o osnovnih lastnostih uporabnikov

Oznaka podatka	**Ime atributa**
EMAIL	Domena_1, Domena_2
BIRTH_DATE	Starost
CITY	Veliko_mesto, Ljubljanska_regija
REGISTRATION_TIME	Cas_od_registracije
REGISTRATION_STATUS	Podpis_pogodbe
CLIENTS_BANK	Banka_1, Banka_2, Banka_3, Banka_4, Banka_5
OS_TYPE	Operacijski_sistem

Iz vsakodnevnih zapisov uporabe izbrane mobilne aplikacije kreiramo atribute, ki predstavljajo število transakcij v mesecu aprilu, njihovo rast v mesecu maju ter izdajatelje plačanih položnic. Imena teh so razvidna iz Tabele 15.

Tabela 15: Atributi, kreirani iz vsakodnevnih zapisov uporabe izbrane mobilne aplikacije

Oznaka tabele	**Ime atributa**
CASHOUT	April_st_cashout, Rast_st_cashout
EINVOICE_payments_from_wallet in EINVOICE_payments_with_topup	April_st_wallet_einvoice, Rast_st_wallet_einvoice, April_st_topup_einvoice, Rast_st_topup_einvoice, Einvoice_izdajatelj_638, Einvoice_izdajatelj_749, Einvoice_izdajatelj_687, Einvoice_izdajatelj_7
P2P_payments	April_st_placanih_P2P, Rast_st_placanih_P2P, April_st_prejetih_P2P, Rast_st_prejetih_P2P
PHOTOPAY_payments_from_wallet in PHOTOPAY_payments_with_topup	April_st_wallet_photopay, Rast_st_wallet_photopay, April_st_topup_photopay, Rast_st_topup_photopay, Einvoice_izdajatelj_654, Einvoice_izdajatelj_757, Einvoice_izdajatelj_193, Einvoice_izdajatelj_857
TOPUP_with_CT, TOPUP_with_DD	April_st_topup_CT, Rast_st_topup_CT, April_st_topup_DD, Rast_st_topup_DD
vse tabele skupaj	April_aktiven, Maj_aktiven

Število transakcij v mesecu maju je pri vseh atributih iz Tabele 15 namesto s preprostim številom definirano kot rast glede na mesec april. Rast je izračunana kot

$$Rast\_st\_xyz = \begin{cases} \frac{Maj\_st\_xyz - April\_st\_xyz}{April\_st\_xyz}, \text{ če je } April\_st\_xyz \neq 0 \\ 0, \text{ če je } April\_st\_xyz = 0 \text{ in } Maj\_st\_xyz = 0 \\ 1, \text{ če je } April\_st\_xyz = 0 \text{ in } Maj\_st\_xyz \neq 0 \end{cases} \quad (42)$$

za vsak *xyz*, kjer *April_st_xyz* predstavlja število *xyz* transakcij v mesecu aprilu, *Maj_st_xyz* pa v mesecu maju.

Iz vseh tabel v Tabeli 15 kreiramo dva atributa, April_aktiven in Maj_aktiven, ki sta enaka 1, v kolikor je uporabnik storil vsaj eno transakcijo v izbranem mesecu, in enaka 0, v kolikor ta ni storil nobene transakcije. Za transakcijo se štejejo prenos denarnih sredstev nazaj na transakcijski račun, nakazilo denarnih sredstev drugemu uporabniku, prejem nakazila denarnih sredstev od drugega uporabnika, kakršnokoli plačilo položnic ter polnjenje mobilne denarnice.

Tako iz pridobljenih podatkov o osnovnih lastnostih uporabnikov in vsakodnevnih zapisov njihove uporabe izbrane mobilne aplikacije kreiramo 41 začetnih atributov. Zaradi velikega števila manjkajočih vrednosti pri atributu, kreiranemu iz podatka o operacijskem sistemu mobilne naprave, tega izločimo iz analize. Tako nam ostane 40 atributov, iz katerih sestavimo tabelo vhodnih podatkov.

Iz tabele vhodnih podatkov naredimo pet dodatnih različic. Eno brez osamelcev, dve s transformacijo osnovne tabele vhodnih podatkov in dve s transformacijo tabele, ki ne vsebuje osamelcev. Dodatne različice naredimo zato, da preverimo, ali imata odstranitev osamelcev in transformacija podatkov kakšen vpliv na uspešnost naših algoritmov.

Osamelce odstranimo tako, da pri atributih s približno normalno porazdelitvijo vsem statističnim enotam, katerih vrednost presega zgornjo ali spodnjo mejo, določimo vrednost zgornje oziroma spodnje meje. Zgornjo in spodnjo mejo za posamezen atribut izračunamo kot tri standardne odklone od povprečja (Brownlee, 2013).

Obstaja več metod transformacije atributov. Najpogostejši sta standardizacija in min-max transformacija. Pri standardizaciji posameznim vrednostim atributa odštejemo matematično upanje in delimo s standardno napako atributa, pri min-max transformaciji pa posameznim vrednostim atributa odštejemo minimalno vrednost in delimo z razliko med maksimalno in minimalno vrednostjo atributa (Clemente et al., b.l.).

Iz vseh tabel z vhodnimi podatki naredimo korelacijsko matriko in izločimo atribute, ki so med seboj preveč korelirani (s korelacijo večjo od 0,9). Mejo za korelacijo privzamemo iz Nisbet et al. (2009). V našem primeru ni potrebno odstraniti nobenega atributa, saj noben nima medsebojne korelacije večje od 0,9.

Po uspešno kreiranih atributih kreiramo tudi izhodni podatek, ki ga želimo napovedati. Kreiramo ga iz junijskih in julijskih vsakodnevnih zapisov uporabe izbrane mobilne aplikacije. V kolikor je uporabnik v mesecu juniju in juliju opravil vsaj eno zvesto transakcijo, je označen kot zvesti uporabnik izbrane mobilne aplikacije. V kolikor uporabnik v mesecu juniju ali juliju ni opravil nobene zveste transakcije, je označen kot uporabnik, ki je prenehal z uporabo izbrane mobilne aplikacije. Za zvesto transakcijo se štejeta kakršnokoli plačilo položnice z denarnimi sredstvi v mobilni denarnici ali iz transakcijskega računa z direktno bremenitvijo ter nakazilo denarnih sredstev drugemu uporabniku.

V naših podatkih imamo 15 % zvestih uporabnikov in 85 % uporabnikov, ki so prenehali z uporabo izbrane mobilne aplikacije, zato imajo zvesti uporabniki izhodni podatek, ki ga želimo napovedati, enak 1, uporabniki, ki so prenehali z uporabo izbrane mobilne aplikacije, pa enak 0.

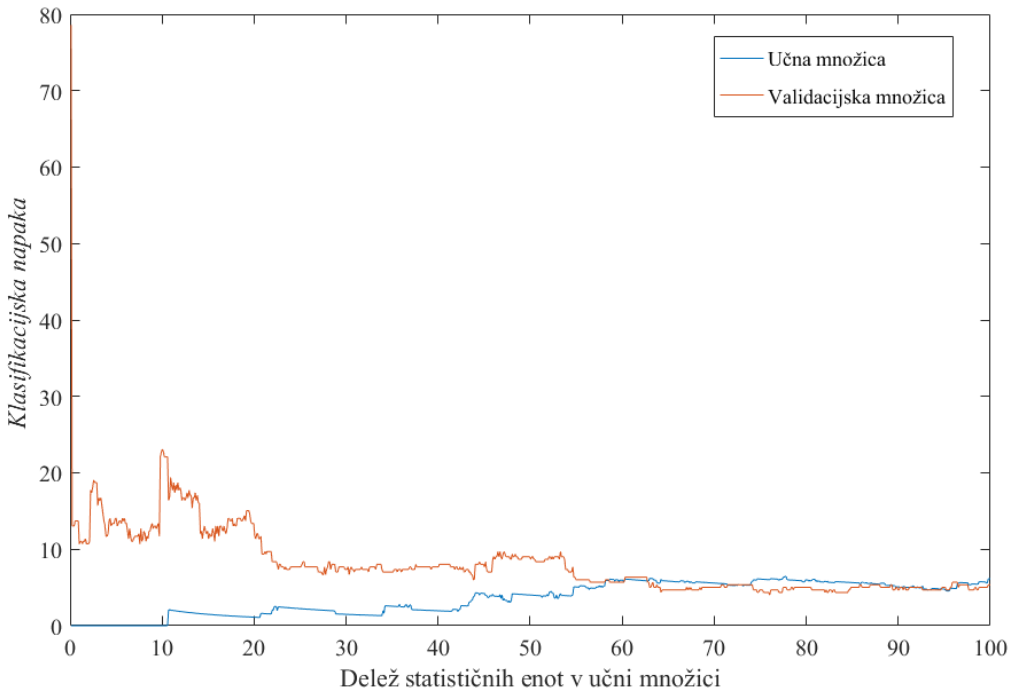
Vhodne podatke skupaj s pripadajočimi izhodnimi podatki, ki jih želimo napovedati, razdelimo na učno, validacijsko in testno množico. Učna množica vsebuje 60 % naključno izbranih statističnih enot, validacijska in testna pa vsaka po 20 %. Procentualno delitev smo povzeli iz Ng (2016).

3.4 Izdelava in kalibracija algoritma logistične regresije

Algoritem logistične regresije izdelamo v programskem jeziku Matlab. Izdelamo ga na tri načine. Najprej kodo za algoritem logistične regresije napišemo sami, od začetka do konca, nato uporabimo že vgrajeno funkcijo *fitglm*, nazadnje pa še vgrajeno funkcijo *stepwiseglm*. Ugotovimo, da z lastno kodo in vgrajeno funkcijo *fitglm* dobimo enake rezultate. Dodana vrednost izdelave lastne kode za logistično regresijo je v zavedanju vseh korakov, ki jih algoritem izvede, uporabljenih predpostavk, načina optimizacije in izrisu krivulje učenja (angl. *learning curve*) algoritma.

S krivuljo učenja preverimo, ali naš algoritem deluje pravilno in ali se pretirano ali premalo (angl. *underfitting*) prilagaja podatkom. Primer krivulje učenja je podan na Sliki 15. V kolikor sta si pri velikem številu statističnih enot v učni množici klasifikacijski napaki učne in validacijske množice zelo različni, se algoritem pretirano prilagaja podatkom, v kolikor sta si zelo blizu in sta obe visoki, pa se algoritem premalo prilagaja podatkom (Ng, 2016). Na Sliki 15 je primer algoritma na naših podatkih o napovedovanju prenehanja uporabe izbrane mobilne aplikacije. Ta je primeren, saj se danim podatkom ne prilagaja pretirano, niti premalo.

Slika 15: Krivulja učenja



Pri izdelavi lastne kode ali uporabi funkcije *fitglm* najprej na učni množici uporabimo logistično regresijo za učenje napovedovanja podatkov, nato izračunamo rezultate za učno, validacijsko in testno množico. Algoritem testiramo, ali je statistično značilen z uporabo

testa razmerja verjetij. Izkaže se, da je pripadajoča p-vrednost v vseh primerih manjša od izbrane vrednosti 0,05, iz česar sledi, da je vsaj en regresijski koeficient različen od 0. To pomeni, da je algoritem, ki vsebuje izbrane atribute, statistično značilen in primeren za uporabo.

Za ocenjevanje uspešnosti algoritma uporabimo mere, opisane v poglavju 2.4.10. Te so ABC mera, F mera in ROC krivulja. Za njihov izračun je potrebno določiti optimalno mejno vrednost. Slednjo izberemo podobno, kot so jo izbrali Au et al. (2003). Optimalna mejna vrednost je tista, ki maksimizira število pravilno klasificiranih statističnih enot med tistimi, ki bodo prenehali z uporabo, in minimizira število nepravilno klasificiranih statističnih enot med tistimi, ki bodo z uporabo izbrane mobilne aplikacije nadaljevali.

Marketinške kampanije, namenjene ohranitvi obstoječih uporabnikov, se lotimo le v primeru, ko pričakujemo, da bomo imeli z njo manjše stroške kot zaslužek z ohranitvijo ciljnih uporabnikov. To pomeni, da je marketinški strošek večji od nič in manjši od koristi ohranitve uporabnika. Najmanjše stroške celotne marketinške kampanije dobimo tako, da jo namenimo vsem, ki nameravajo prenehati z uporabo (maksimiziramo število pravilno klasificiranih statističnih enot med tistimi, ki bodo prenehali uporabljati aplikacijo), in hkrati le tistim, ki resnično nameravajo prenehati z uporabo (minimiziramo število nepravilno klasificiranih statističnih enot med tistimi, ki bodo še naprej uporabljali aplikacijo).

V našem primeru imajo tisti, ki bodo še naprej uporabljali aplikacijo, izhodni podatek enak vrednosti 1 in tisti, ki bodo z uporabo prenehali, izhodni podatek enak vrednosti 0. Optimalno mejno vrednost tako izračunamo s formulo

$$\max_t \left(\frac{TN(t)}{FP(t) + TN(t)} - \frac{FN(t)}{TP(t) + FN(t)} \right) \quad (43)$$

Naš cilj je najti najboljši algoritem s čim manj atributi, zato po izračunu vseh mer za ocenjevanje uspešnosti pogledamo, kateri so statistično značilni atributi v našem modelu. Za statistično značilne atribute izberemo tiste, ki imajo pri Waldovem testu p-vrednost nižjo od 0,05. Algoritem logistične regresije ponovno poženemo na istih podatkih, vendar iz njih predhodno izločimo vse atribute, ki niso statistično značilni. Pri izločanju atributov moramo po besedah Hosmer et al. (2013) paziti, da v kolikor imamo opravka s slamnatimi atributi in smo za njihov opis kreirali $z - 1$ atributov, je potrebno v algoritmu imeti prisotne ali vse ali pa nobenega slamnatega atributa.

Izračunamo rezultate za učno, validacijsko in testno množico ter algoritem testiramo za statistično značilnost v primerjavi z algoritmom, ki vsebuje le konstanto, in z zgoraj omenjenim algoritmom, ki vsebuje vse atribute. Ponovno določimo optimalno mejno vrednost in ocenimo uspešnost algoritma. Dobljene rezultate med seboj primerjamo.

Nato za izgradnjo modela uporabimo vgrajeno funkcijo *stepwiseglm*, ki v model logistične regresije postopno dodaja različne kombinacije atributov. Vgrajena funkcija nam vrne seznam izbranih atributov in njihove p-vrednosti ter p-vrednost celotnega algoritma v primerjavi z algoritmom, ki vsebuje le konstanto. Algoritem testiramo za statistično značilnost z obema algoritmoma, omenjenima zgoraj. Izračunamo rezultate za učno, validacijsko in testno množico, določimo optimalno mejno vrednost ter ocenimo uspešnost algoritma.

Opisan postopek ponovimo na vseh različicah podatkov, omenjenih v poglavju 3.3.

3.5 Izdelava in kalibracija nevronskih mrež

Model nevronskih mrež prav tako izdelamo v programskem jeziku Matlab. Izdelamo ga tako, da kodo napišemo sami.

Uporabimo večslojno usmerjeno nevronsko mrežo, sestavljeno iz vhodnega, skritega ter izhodnega sloja. Število nevronov v skritem sloju in število iteracij pri iskanju maksimuma funkcije verjetja določimo tako, da preizkusimo več različnih možnosti in nato izberemo najboljšo glede na ABC mero, opisano v poglavju 2.4.7.

Preizkusimo kombinacije, ki vsebujejo od 20 do 160 nevronov v skritem sloju in od 20 do 30.000 iteracij pri iskanju maksimuma funkcije verjetja. Po izbiri najboljše kombinacije nevronov v skritem sloju in števila iteracij izračunamo pripadajoče rezultate za učno, validacijsko in testno množico. Optimalno mejno vrednost določimo enako kot pri algoritmih logistične regresije, z istimi merami ocenimo tudi uspešnost nevronskih mrež.

Naš cilj je najti najboljši algoritem s čim manj atributi, zato po izračunu vseh mer za ocenjevanje uspešnosti ponovno zaženemo nevronske mreže na istih podatkih, vendar iz njih predhodno izločimo atribute, izločene že pri logistični regresiji. Prvič izločimo atribute, ki niso statistično značilni pri uporabi vgrajene funkcije *fitglm*, drugič pa atribute, ki niso izbrani pri uporabi vgrajene funkcije *stepwiseglm*.

Zopet optimiziramo število nevronov v skritem sloju ter število iteracij pri iskanju maksimuma funkcije verjetja glede na ABC mero. Po ustrezni optimizaciji izračunamo rezultate za učno, validacijsko in testno množico. Določimo optimalno mejno vrednost ter ocenimo uspešnost algoritma z do sedaj znanimi merami. Dobljene rezultate med seboj primerjamo.

Opisan postopek tako kot pri algoritmih logistične regresije ponovimo na vseh različicah podatkov, omenjenih v poglavju 3.3.

3.6 Uspešnost algoritmov

V nadaljevanju opisujemo rezultate uspešnosti naslednjih algoritmov:

- **LR algoritem:** algoritem logistične regresije na vseh atributih, ki za napovedovanje rezultatov uporablja vgrajeno funkcijo *fitglm*,
- **LR stat. algoritem:** algoritem logistične regresije na predhodno izločenih atributih, ki niso statistično značilni v LR algoritmu in za napovedovanje rezultatov uporablja vgrajeno funkcijo *fitglm*,
- **LR stepwise algoritem:** algoritem logistične regresije, ki postopno dodaja različne kombinacije atributov z vgrajeno funkcijo *stepwiseglm*,
- **NM algoritem:** nevronske mreže na vseh atributih,
- **NM stat. algoritem:** nevronske mreže na predhodno izločenih atributih, ki niso statistično značilni v LR algoritmu,
- **NM stepwise algoritem:** nevronske mreže na predhodno izločenih atributih, ki niso izbrani pri uporabi vgrajene funkcije *stepwiseglm*.

In naslednjih različic podatkov:

- **osnovni podatki:** uporabljena je osnovna tabela vhodnih podatkov brez nadaljnje obdelave,
- **standardizirani podatki:** uporabljena je standardizirana osnovna tabela vhodnih podatkov,
- **min-max podatki:** uporabljena je min-max transformacija osnovne tabele vhodnih podatkov,
- **osamelci:** uporabljena je osnovna tabela vhodnih podatkov, iz katere so odstranjeni osamelci,
- **standardizirani osamelci:** uporabljena je standardizirana osnovna tabela vhodnih podatkov, iz katere so odstranjeni osamelci,
- **min-max osamelci:** uporabljena je min-max transformacija osnovne tabele vhodnih podatkov, iz katere so odstranjeni osamelci.

Tabela 16: P-vrednosti pri testiranju statistične značilnosti algoritmov logistične regresije

Algoritem	**Osnovni podatki ***	**Osamelci ****
LR algoritem	8,03e-85	1,26e-85
LR stat. algoritem	5,27e-98	6,00e-99
LR stepwise algoritem	2,56e-101	1,36e-102

Legenda: * rezultati so enaki tudi za različici standardizirani in min-max podatki
** rezultati so enaki tudi za različici standardizirani in min-max osamelci

Pri testiranju statistične značilnosti LR algoritma, LR stat. algoritma in LR stepwise algoritma ugotovimo, da so pri vseh različicah podatkov vsi statistično značilni, saj so vse pripadajoče p-vrednosti manjše od $1{,}26e^{-85}$. Podrobnejši rezultati so razvidni iz Tabele 16.

Pri medsebojni primerjavi algoritmov logistične regresije ugotovimo, da se pri vseh različicah podatkov LR algoritem statistično razlikuje od LR stat. algoritma, medtem ko se statistično ne razlikuje od LR stepwise algoritma. Pri vseh različicah podatkov se med seboj statistično razlikujeta tudi LR stat. algoritem in LR stepwise algoritem. Pripadajoče p-vrednosti pri medsebojni primerjavi algoritmov logistične regresije so razvidne iz Tabele 17.

Tabela 17: P-vrednosti pri medsebojni primerjavi algoritmov logistične regresije

		LR algoritem	**LR stat. algoritem**
Osnovni podatki *	LR stat. algoritem	1,27e-02	/
	LR stepwise algoritem	0,70	2,30e-06
Osamelci **	LR stat. algoritem	2,30e-02	/
	LR stepwise algoritem	1,00	1,02e-08

Legenda: * rezultati so enaki tudi za različici standardizirani in min-max podatki
** rezultati so enaki tudi za različici standardizirani in min-max osamelci

Uspešnost algoritmov ocenimo s F in ABC mero ter ROC krivuljo. Izračunamo tudi natančnost, ki jo uporabimo za testiranje, ali se algoritem pretirano ali premalo prilagaja podatkom. Ugotovimo, da se algoritmi logistične regresije pri standardiziranih in min-max osnovnih podatkih ter osamelcih pretirano prilagajajo podatkom. Prav tako se pretirano prilagaja podatkom NM algoritem pri min-max osnovnih podatkih ter osamelcih. Obstajajo različni načini za odpravo pretiranega prilagajanja, vendar se zaradi dobrih rezultatov pri ostalih algoritmih ter različicah podatkov odločimo, da tega problema ne naslovimo in ga ne odpravljamo. Rezultate algoritmov, ki se pretirano prilagajajo, v tabelah rezultatov ne prikažemo in jih ne upoštevamo.

Tabela 18: Rezultati F mere pri napovedovanju prenehanja uporabe izbrane mobilne aplikacije

Algoritem	**Osnovni podatki**	**Standardizirani podatki**	**Min-max podatki**	**Osamelci**	**Standardizirani osamelci**	**Min-max osamelci**
LR algoritem	0,78	/	/	0,81	/	/
LR stat. algoritem	0,82	/	/	0,82	/	/
LR stepwise algoritem	0,80	/	/	0,79	/	/
NM algoritem	0,74	0,76	/	0,76	0,80	/
NM stat. algoritem	0,81	0,75	0,77	0,82	0,84	0,84
NM stepwise algoritem	0,80	0,80	0,79	0,82	0,74	0,76

Tabela 18 prikazuje rezultate F mere. Opazimo, da so pri vseh algoritmih in vseh različicah podatkov rezultati F mere med 0,74 in 0,84. To so v praksi zelo dobri rezultati. Rezultati se med algoritmi ne razlikujejo močno, kar pomeni, da so si odstotki pravilno klasificiranih pozitivno klasificiranih ter resnično pozitivno klasificiranih statističnih enot med seboj podobni.

Tabela 19 prikazuje rezultate ABC mere. Opazimo, da so pri vseh algoritmih in vseh različicah podatkov rezultati ABC mere med -0,04 in 0,13. Rezultati so boljši predvsem pri standardiziranih in min-max osnovnih podatkih ter osamelcih. Pri analizi podatkov se moramo zavedati, da lahko ti zaradi majhnega vzorca statističnih enot pri različni razdelitvi učne, validacijske ter testne množice variirajo. Zgodi se lahko, da je pri eni razdelitvi en algoritem boljši od drugega, pri drugi razdelitvi pa ravno obratno, zato zaključkov o tem, kateri algoritem je boljši, v našem primeru ni smiselno postavljati. Rečemo lahko le, da se vsi algoritmi med seboj podobno dobro odrežejo.

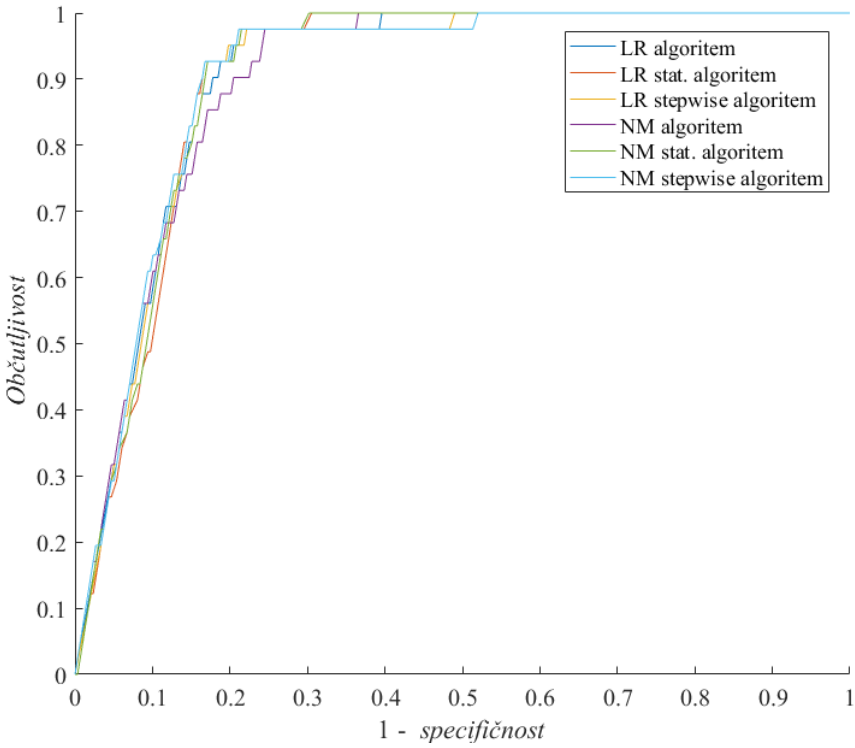
Tabela 19: Rezultati ABC mere pri napovedovanju prenehanja uporabe izbrane mobilne aplikacije

Algoritem	**Osnovni podatki**	**Standardizi-rani podatki**	**Min-max podatki**	**Osamelci**	**Standardizi-rani osamelci**	**Min-max osamelci**
LR algoritem	-0,04	/	/	-0,01	/	/
LR stat. algoritem	0,05	/	/	0,05	/	/
LR stepwise algoritem	-0,01	/	/	-0,01	/	/
NM algoritem	-0,09	-0,04	/	-0,07	0,10	/
NM stat. algoritem	0,05	0,13	0,13	0,05	0,07	0,07
NM stepwise algoritem	-0,01	-0,01	0,10	0,05	0,13	0,02

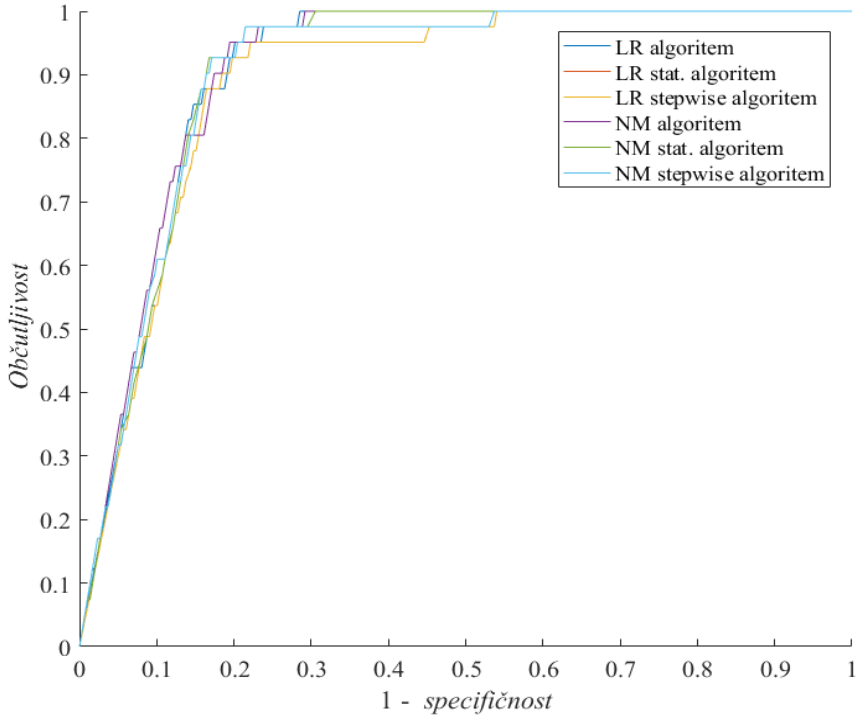
Pri primerjavi ROC krivulj se odločimo za ločene prikaze ROC krivulj za največ šest kombinacij algoritmov in/ali različic podatkov. Prikaz več kot šestih ROC krivulj hkrati je namreč nepregleden. Najprej prikažemo ROC krivulje za vse algoritme pri osnovnih podatkih, nato ROC krivulje za vse algoritme pri osamelcih. Sledi prikaz ROC krivulj NM stat. algoritma pri vseh različicah podatkov ter ROC krivulj NM stepwise algoritma pri vseh različicah podatkov.

Slika 16 prikazuje ROC krivulje za vse algoritme pri osnovnih podatkih. Opazimo, da so si ROC krivulje med seboj zelo podobne, odsekoma sta malo nižji le ROC krivulji NM algoritma ter NM stepwise algoritma.

Slika 16: ROC krivulje vseh algoritmov pri osnovnih podatkih



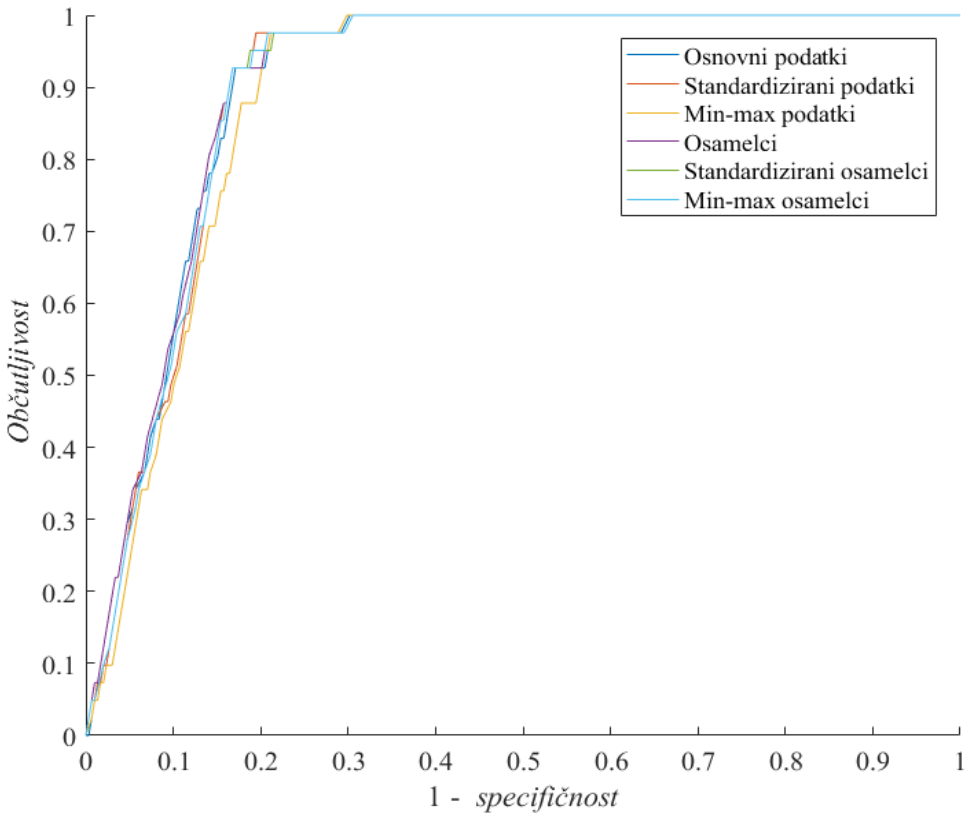
Slika 17: ROC krivulje vseh algoritmov pri podatkih brez osamelcev



Tudi ROC krivulje algoritmov pri podatkih, ki ne vsebujejo osamelcev, so si med seboj zelo podobne, kar je razvidno iz Slike 17. Odsekoma sta malo nižji le ROC krivulji LR stepwise algoritma ter NM stepwise algoritma.

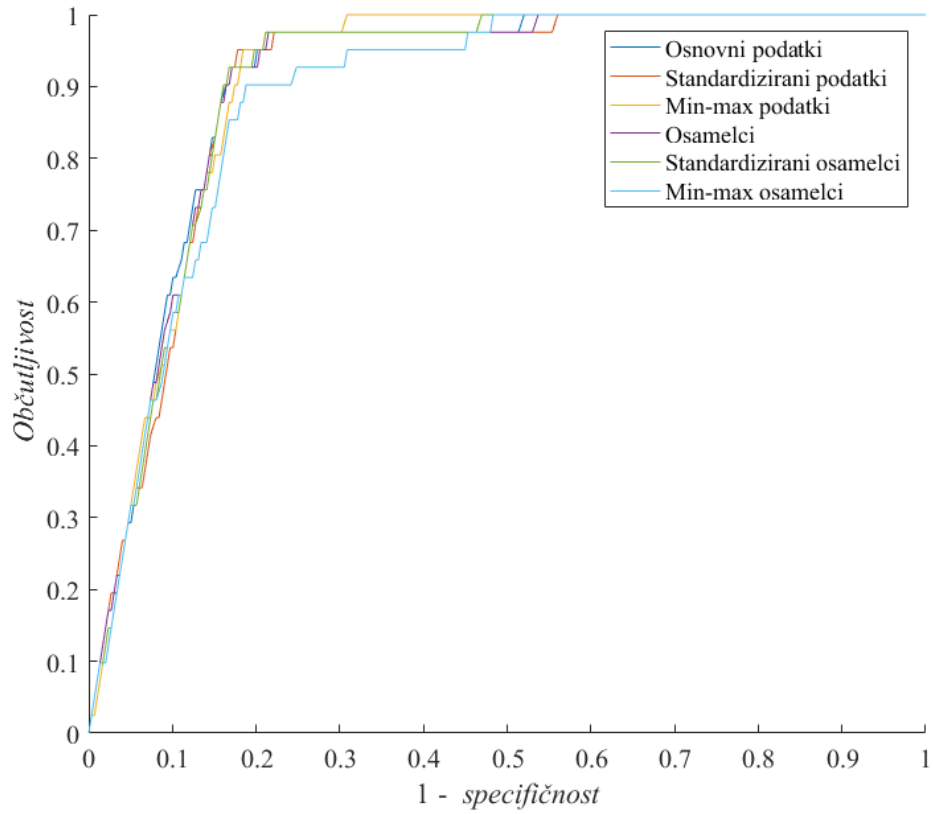
Slika 18 prikazuje ROC krivulje NM stat. algoritma pri vseh različicah podatkov. Tudi tu opazimo, da so si krivulje med seboj zelo podobne, odsekoma je nižja le krivulja pri min-max različici podatkov.

Slika 18: ROC krivulje NM stat. algoritma pri vseh različicah podatkov



Slika 19 prikazuje ROC krivulje NM stepwise algoritma pri vseh različicah podatkov. Tudi tu so si krivulje med seboj zelo podobne, odsekoma je nižja le krivulja pri različici podatkov min-max osamelci.

Slika 19: ROC krivulje NM stepwise algoritma pri vseh različicah podatkov



Tudi pregled ROC krivulj potrjuje, da se vsi algoritmi pri vseh različicah podatkov podobno odrežejo ter da se vsi odrežejo dobro.

3.7 Razlaga rezultatov

Rezultat napovedovanja prenehanja uporabe izbrane mobilne aplikacije je tabela uporabnikov z verjetnostjo prenehanja uporabe izbrane mobilne aplikacije. Pri vsakem algoritmu in različici podatkov dobimo različne verjetnosti prenehanja uporabe za uporabnike. Odločimo se, da uporabimo povprečje vseh primernih rezultatov. Primerni rezultati so tisti, pri katerih se algoritem podatkom ne prilagaja niti pretirano niti premalo.

Poleg verjetnosti prenehanja uporabe izbrane aplikacije nas zanima tudi, kateri so tisti atributi, ki so najbolj prispevali k napovedanim verjetnostim. Ker dobimo z LR stat. algoritmom in z LR stepwise algoritmom podobne rezultate kot pri LR algoritmu ter z NM stat. algoritmom in NM stepwise algoritmom podobne rezultate kot pri NM algoritmu, sklepamo, da najbolj prispevajo k napovedi ravno atributi, uporabljeni v teh algoritmih.

Tabela 20 prikazuje atribute, uporabljene pri posameznem algoritmu in posamezni različici podatkov. Opazimo, da se pri vseh kombinacijah pojavijo atributi Starost, April_aktiven ter Maj_aktiven.

Tabela 20: Uporabljeni atributi pri posameznem algoritmu in različici podatkov

Algoritem	**Osnovni podatki***	**Osamelci****
LR stat. algoritem oz. NM stat. algoritem	Starost, Rast_st_cashout, Rast_st_topup_DD, April_aktiven, Maj_aktiven	Starost, April_aktiven, Maj_aktiven
LR stepwise algoritem oz. NM stepwise algoritem	Starost, Banka_1, Banka_2, Banka_3, Banka_4, Banka_5, Einvoice_izdajatelj_638, Einvoice_izdajatelj_749, April_st_wallet_photopay, April_aktiven, Maj_aktiven	Starost, Banka_1, Banka_2, Banka_3, Banka_4, Banka_5, April_st_wallet_einvoice, Rast_st_wallet_einvoice, Rast_st_topup_einvoice, Einvoice_izdajatelj_749, April_st_wallet_photopay, April_aktiven, Maj_aktiven

Legenda: * rezultati so enaki tudi za različici standardizirani in min-max podatki
** rezultati so enaki tudi za različici standardizirani in min-max osamelci

Podrobneje si ogledamo vektor regresijskih koeficientov pri LR stat. algoritmu in LR stepwise algoritmu pri osnovnih podatkih ter osamelcih.

Rezultati prikazujejo, da so v povprečju starejši uporabniki bolj zvesti. Starejši uporabniki po odkritju aplikacije, ki jim ustreza, to uporabljajo, medtem ko mlajši uporabniki iščejo vedno nove aplikacije. Glede na življenjsko vrednost uporabnika je smiselno pri nadaljnjih marketinških pristopih, ceniku ter funkcionalnosti izbrane mobilne aplikacije upoštevati omenjeno ugotovitev.

Rezultati prikazujejo tudi, da bodo uporabniki, ki so izbrano mobilno aplikacijo uporabljali v zaporednih predhodnih mesecih, z uporabo nadaljevali, kar se sklada z našimi pričakovanji že pred samo analizo. Zanimivo bi bilo raziskati, zakaj nekateri uporabniki po namestitvi aplikacije z njeno uporabo sploh ne pričnejo. V kolikor bi prepričali uporabnike, da že na začetku pričnejo z uporabo izbrane mobilne aplikacije, bi se število njenih zvestih uporabnikov verjetno povišalo.

Pri rezultatih je zanimivo še to, da so vsi regresijski koeficienti atributov, prikazanih v Tabeli 20, pozitivni, z izjemo regresijskih koeficientov, pripadajočih nekaterim slamnatim

atributom za oznako bank ter atributu Rast_st_cashout. To pomeni, da v kolikor je uporabnik komitent določene banke in/ali v kolikor trend prenosa denarnih sredstev nazaj na transakcijski račun raste, višja je verjetnost prenehanja uporabe izbrane mobilne aplikacije.

V kolikor so vrednosti ostalih atributov, katerim pripadajo pozitivni regresijski koeficienti, višje, močnejša je zvestoba uporabnika in nižja je verjetnost prenehanja uporabe izbrane mobilne aplikacije. Omenjeni atributi so trend polnitve mobilne denarnice preko direktne bremenitve, število in trend plačil avtomatsko prejetih položnic v aplikacijo z denarnimi sredstvi v mobilni denarnici, trend plačila avtomatsko prejetih položnic v aplikacijo iz transakcijskega računa z direktno bremenitvijo in število plačil položnic s slikanjem z denarnimi sredstvi v mobilni denarnici v predhodnem obdobju. Močnejša zvestoba se predvideva tudi za komitente določenih bank ter določene izdajatelje položnic, na katere avtomatsko prejemanje so naročeni uporabniki.

SKLEP

V magistrskem delu smo predstavili uporabo logistične regresije in nevronskih mrež za napovedovanje prenehanja uporabe izbrane mobilne aplikacije. Ugotovili smo, da se različni algoritmi logistične regresije in nevronskih mrež na različnih različicah podatkov podobno odrežejo, zato smo se odločili, da za končni rezultat vzamemo povprečje vseh algoritmov pri primernih različicah podatkov. Različica podatkov je za posamezen algoritem primerna, ko se podatkom ne prilagaja niti pretirano niti premalo.

Na postavljeno raziskovalno vprašanje smo tako uspešno odgovorili, vendar vseeno obstajajo možnosti za izboljšave. Izboljšave vidimo predvsem v podaljšanju časovne vrste v vhodnih podatkih, kar v obdobju izdelave magistrskega dela žal ni bilo mogoče, saj je izbrana mobilna aplikacija precej nova na trgu.

Z daljšo časovno vrsto bi imeli na voljo ne le večje število uporabnikov, temveč tudi večje število začetnih atributov. Število začetnih atributov bi se povečalo, ker se mobilna aplikacija Hal mBills nenehno izboljšuje z dodajanjem vedno novih funkcionalnosti. V juliju 2016 je bila aplikaciji dodana podpora za spletno plačevanje, v septembru 2016 pa možnost samodejnega mesečnega polnjenja mobilne denarnice in plačevanja računov v izbranih lokalih ter restavracijah s slikanjem QR kode (App Annie, 2016; Hal mBills, 2016).

Kljub kratki časovni vrsti so naši rezultati analize dobri in smo z njimi zadovoljni.

LITERATURA IN VIRI

1. Amin, A., Shehzad, S., Khan, C., Ali, I., & Anwar, S. (2015). Churn Prediction in Telecommunication Industry using Rough Set Approach. V D. Camacho, S. W. Kim & B. Trawinski (ur.), *New Trends in Computational Collective Intelligence* (str. 83–95). Cham: Springer.
2. *App Annie*. Najdeno 20. junija 2016 na spletnem naslovu https://www.appannie.com/dashboard/home/?_ref=header
3. Apple. (2008, 10. julij). *iPhone 3G on Sale Tomorrow*. Najdeno 29. maja 2016 na spletnem naslovu http://www.apple.com/pr/library/2008/07/10iPhone-3G-on-Sale-Tomorrow.html
4. *Apps*. Najdeno 17. junija 2016 na spletnem naslovu https://play.google.com/store/apps/
5. Au, T., Li, S., & Ma, G. (2003). Applying and Evaluating Models to Predict Customer Attrition Using Data Mining Techniques. *Journal of Comparative International Management*, *6*(1), 10–22.
6. Berry, M. J. A., & Linoff, G. S. (2004). *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. Indianapolis: Wiley Publishing.
7. Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.
8. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Cambridge: Springer.
9. Brownlee, J. (2013, 31. december). How to Identify Outliers in your Data. *Machine Learning Process*. Najdeno 24. junija 2016 na spletnem naslovu http://machinelearningmastery.com/how-to-identify-outliers-in-your-data/
10. Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, *36*(3), 4626–4636.
11. Bush, G. (1989, 12. april). Principles of ethical conduct for government officers and employees. *Executive Order No. 12674. Pt. 1*. Najdeno 18. novembra 1997 na spletnem naslovu http://www.usoge.gov/exorders/eo12674.html
12. Chawla, N. V. (2005). Data Mining for Imbalanced Datasets: An Overview. V O. Maimon & L. Rokach (ur.), *The Data Mining and Knowledge Discovery Handbook*, (str. 853–867). New York: Springer.
13. Chen, C., Liaw, A., & Breiman, L. (2004). Using random forests to learn imbalanced data. Najdeno 11. junija 2016 na spletnem naslovu http://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf
14. Clemente, M., Ginger-Bosch, V., & San Matias, S. (b.l.). *Assesing Classification methods for churn prediction by composite indicators*. Valencia: Department of Applied Statistics, Operations Research and Quality, Universitat Politecnica de Valencia.
15. *Delavska hranilnica*. Najdeno 27. junija 2016 na spletnem naslovu http://www.delavska-hranilnica.si/DH,,.htm
16. Engle, R. F. (1984). Chapter 13 Wald, likelihood ratio, and Lagrange multiplier tests in econometrics. V Z. Griliches & M. D. Intriligator (ur.), *Handbook of Econometrics, Volume 2* (str. 775–826). California: Elsevier.

17. Ferle, M. (2010, 27. december). Napovedna analitika. *Monitor PRO*. Najdeno 10. aprila 2016 na spletnem naslovu http://www.monitorpro.si/41771/praksa/napovedna-analitika/
18. Forbes, C., Evans, M., Hastings, N., & Peacock, B. (2011). *Statistical Distributions* (4th ed.). Hoboken: John Wiley & Sons.
19. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A Review on Ensembles for the Class Imbalance Problem: Bagging, Boosting and Hybrid Based Approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4), 463–484.
20. *Google Play Store*. Najdeno 20. junija 2016 na spletnem naslovu http://www.androidcentral.com/google-play-store
21. Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the class imbalance problem. *Fourth International Conference on Natural Computation*, *4*, 192–201.
22. *Hal mBills.* Najdeno 20. junija 2016 na spletnem naslovu http://www.mbills.si/
23. Hadden, J., Tiwari, A., Roy, R., & Ruta, D. (2008). Churn Prediction: Does Technology Matter? *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, *2*(4), 524–530.
24. Hand, D., Heikki, M., & Smyth, P. (2001). *Principles of Data Mining*. Massachusetts: Massachusetts Institute of Technology.
25. *History*. Najdeno 2. junija 2016 na spletnem naslovu https://www.android.com/history/#/donut
26. Hosmer, D. W., Hosmer,T., Le Cessie, S., & Lemeshow, S. (1997). A comparison of godness-of-fit tests for the logistic regression model. *Statistics in Medicine*, *16*(9), 965–980.
27. Hosmer, D. W. JR., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). Hoboken: John Wiley & Sons.
28. Hu, G., Wang, J., & Feng, W. (2013). Multivariate Regression Modeling for Home Value Estimates with Evaluation Using Maximum Information Coefficient. V R. Lee (ur.), *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2012* (str. 69–81). Heidelberg: Springer.
29. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to Statistical Learning with Applications in R*. New York: Springer.
30. *Klikin.* Najdeno 22. junija 2016 na spletnem naslovu https://www.nlb.si/klikin
31. Lazarov, V., & Capota, M. (2007). Churn Prediction. Najdeno 5. junija 2016 na spletnem naslovu http://www.vladislav.lazarov.pro/files/research/papers/churn-prediction.pdf
32. Lopez, V., Fernandez, A., Garcia, S., Palade, V., & Herrera, F. (2013) An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250, 113–141.
33. Marsland, S. (2009). *Machine Learning: An Algorithmic Perspective*. Palmerston North: Taylor & Francis Group.
34. Menard, S. (2001). Applied Logistic Regression Analysis. *Sage University Paper series on Qantitative Applications in the Social Sciences*, *7*(106).

35. *Mobilna banka Abamobi.* Najdeno 22. junija 2016 na spletnem naslovu http://www.abanka.si/mobilna-banka-abamobi/?MapaId=96021
36. Ng, A. (2016). *Machine Learning* [Video File]. Stanford University. Najdeno 1. junija 2016 na spletnem naslovu https://www.coursera.org/learn/machine-learning
37. Nisbet, R., Elder, J., & Miner, G. (2009). *Handbook of Statistical Analysis and Data Mining Applications*. Amsterdam: Elsevier.
38. *Nova KBM.* Najdeno 27. junija 2016 na spletnem naslovu https://www.nkbm.si/vstopna-stran
39. Peng, C. Y. J., & So, T. S. H. (2002). Logistic Regression Analysis and Reporting: A Primer. *Understanding Statistics*, *1*(1), 31–70.
40. Pocket Gamer. (b.l.). *App store metrics.* Najdeno 19. junija 2016 na spletnem naslovu http://www.pocketgamer.biz/metrics/app-store/
41. Rodriguez, G. (2007). Lecture Notes on Generalized Linear Models. Najdeno 1. julija 2016 na spletnem naslovu http://data.princeton.edu/wws509/notes/
42. Rojas, R. (1996). *Neural networks: A Systematic Introduction*. Berlin: Springer.
43. Ropret, M. (2015, 12. november). Hal mBills obljublja preprostejše plačevanje položnic. *Delo*. Najdeno 31. maja 2016 na spletnem naslovu http://www.delo.si/znanje/infoteh/hal-mbills-obljublja-enostavnejse-placevanje-poloznic.html
44. Shaaban, E., Helmy, Y., Khedr, A., & Nasr, M. (2012). A Proposed Churn Prediction Model. *International Journal of Engineering Research and Applications*, *2*(4), 693–697.
45. Shalizi, C. R. (2016). *Advanced Data Analysis from an Elementary Point of View*. Cambridge: Cambridge University Press.
46. Sharma, A., & Panigrahi, P. K. (2011). A Neural Network based Approach for Predicting Customer Churn in Cellular Network Services. *International Journal of Computer Applications*, *27*(11), 26–31.
47. Shaza, M. A. E., & Abraham, A. (2013) A review of Class Imbalance Problem. *Journal of Network and Innovative Computing*, 1, 332–340.
48. Smartphone. (b.l.). V *Oxford Dictionaries*. Najdeno 2. junija 2016 na spletni strani http://www.oxforddictionaries.com/definition/english/smartphone
49. Smith, K. A., & Gupta, J. N. D. (2002). *Neural networks in business: Techniques and Applications*. Hershey: Idea Group Publishing.
50. Statista. (b.l.). *Statistics and facts about mobile app usage.* Najdeno 11. junija 2016 na spletnem naslovu http://www.statista.com/topics/1002/mobile-app-usage/
51. *Top categories.* Najdeno 19. junija 2016 na spletnem naslovu http://www.appbrain.com/stats/android-market-app-categories
52. Tsay, R. S. (2008). Lecture: Inference about sample mean. Najdeno 18. junija 2016 na spletnem naslovu http://faculty.chicagobooth.edu/ruey.tsay/teaching/ama/lec2-08.pdf
53. Van den Poel, D., & Lariviere, B. (2004). Customer Attrition Analysis For Financial Services Using Proportional Hazard Models. *European Journal of Operational Research*, *157*(1), 196–217.

54. Viaene, S., Baesens, B., Van den Poel, D., Vanthienen, J., & Dedene, G. (2002). Bayesian neural network learning for repeat purchase modelling in direct marketing. *European Journal of Operational Research*, *138*(1), 191–211.
55. Vuk, M., & Curk, T. (2006). ROC Curve, Lift Chart and Calibration Plot. *Metodološki zvezki*, *3*(1), 89–108.
56. Weiss, G. M. (2004). Mining with Rarity: A Unifying Framework. *SIGKDD Explorations Newsletter*, *6*(1), 7–19.
57. Witten, I. H., Eibe, F., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques*. Amsterdam: Elsevier.
58. Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. London: Taylor & Francis Group.
59. Zidar, R., & Biloslavo, R. (2010). Nevronske mreže kot nova metoda za reševanje poslovnih problemov in možnosti uporabe v managementu. *Management*, *3*(5), 279–291.