UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

UPORABA PODATKOVNEGA RUDARJENJA S STROJNIM UČENJEM PRI OBLIKOVANJU CEN AVTOMOBILSKIH PREMOŽENJSKIH ZAVAROVANJ

Ljubljana, avgust 2016 LUKA RUTAR

IZJAVA O AVTORSTVU

Podpisani Luka Rutar, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Uporaba podatkovnega rudarjenja s strojnim učenjem pri oblikovanju cen avtomobilskih premoženjskih zavarovanj, pripravljenega v sodelovanju s svetovalcem prof. dr. Igorjem Lončarskim in sosvetovalcem prof. dr. Markom Pahorjem,

IZJAVLJAM

1. da sem predloženo delo pripravil 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 poskrbel, 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 pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;

7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil 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 30.08.2016 Podpis študenta:_____

KAZALO

KAZALO TABEL

KAZALO SLIK

UVOD

Z razvojem informacijske tehnologije je prišlo do selitve podatkov v računalniško podprte podatkovne baze in posledično imajo danes zavarovalnice v teh podatkovnih bazah na voljo ogromno količino podatkov, ki so vezani na njihove zavarovalne pogodbe (podatki o zavarovancih, zavarovanih predmetih, škodnem dogajanju, plačanih premijah, popustih, …). Dostopnost teh podatkov predstavlja priložnost in hkrati izziv za podatkovno rudarjenje (angl. *data mining*) (Apte et al., 1998, str. 2), ki se ukvarja z iskanjem skritih povezav, ki so prisotne v poslovnih podatkih, z namenom ustvarjanja napovedi za uporabo v prihodnosti ter ima različne možnosti uporabe tudi v zavarovalništvu (Sumathi & Sivanandam, 2006).

Izdelki, ki jih zavarovalnica prodaja, so zavarovalne pogodbe. S sklenitvijo zavarovalne pogodbe zavarovalnica sprejme določene obveznosti, za kar ji zavarovanec plača določen znesek, ki ga imenujemo zavarovalna premija (Močivnik, 2010, str. 70). Ta znesek mora biti ustrezen, kar pomeni, da mora v povprečju zadoščati za pokritje vseh stroškov, ki jih ima zavarovalnica zaradi prevzetih obveznosti, nadalje pa tudi stroškov njenega poslovanja ter ob tem še prinašati določen dobiček (Chapados, 2010, str. 6). Vendar pa ni pomemben le skupen učinek vseh zavarovalnih pogodb, ampak je koristno, če višina premije čim bolje ustreza višini stroškov, ki jih ima zavarovalnica z določeno zavarovalno pogodbo, ta znesek si lahko predstavljamo kot tveganost te zavarovalne pogodbe; če zavarovalnica za manj tvegane pogodbe zaračuna nižjo premijo, lahko to pritegne nove stranke, ki bi pri konkurenčnih zavarovalnicah plačale več (Dugas et al., 2003, str. 200). Povišanje premije za bolj tvegane pogodbe pa poveča dobiček zaradi višjih prihodkov (v kolikor so stranke pripravljene plačati višjo premijo) ali zaradi manjših stroškov (v kolikor stranke pogodbe zaradi višje premije ne sklenejo) (Dugas et al., 2003, str. 200).

Ugotavljanje tveganosti posamezne zavarovalne pogodbe oziroma določanje ustrezne premije je za zavarovalnice torej zelo pomembno. Za ta namen je potrebno razviti model, s pomočjo katerega na podlagi znanih značilnosti zavarovalne pogodbe določimo ustrezno zavarovalno premijo. Omenjen model, ki je preprosto gledano nekakšen cenik zavarovanj, si lahko predstavljamo kot skupek pravil ali matematičnih enačb.

Moje magistrsko delo v najširšem pogledu predstavlja opis postopka razvoja takšnega modela za izbrano skupino zavarovanj, ki sem ga izvedel v okviru priprave magistrskega dela, ter ovrednotenje dobljenega modela v primerjavi z obstoječimi modeli, ki jih izbrana zavarovalnica uporablja v praksi. Pri razvoju modela sem se poslužil podatkovnega rudarjenja s strojnim učenjem, ki se je v ta namen že izkazalo za uspešno (Apte et al., 1998, str. 9; Chapados, 2010, str. 28) ter se mi je zdelo primerno tudi glede na naravo problema (velika količina podatkov in odvisnost rezultata od številnih parametrov, ki so med seboj lahko odvisni). Glavna značilnost uporabljenega pristopa je, da pravila oziroma enačbe, ki predstavljajo model, samodejno ustvarijo računalniški algoritmi na podlagi podatkov oziroma primerov iz preteklosti, ki jim jih predstavimo. Zaradi tega je potreben manjši človeški vložek, hkrati pa so dobljena pravila lahko bolj kompleksna in rezultati bolj natančni; po drugi strani je pravila in rezultate takšnega modela zaradi tega težje pojasniti, kar predstavlja oviro za uveljavitev tega pristopa. V svojem delu želim

prikazati, da se da z uporabo podatkovnega rudarjenja s strojnim učenjem pri določanju premij doseči dobre rezultate, ter nakazati možnost vzporedne uporabe tega pristopa in klasičnih metod ter tako prispevati k njegovi uveljavitvi.

Premijo, ki jo je potrebno plačati za sklenitev zavarovalne pogodbe, imenujemo bruto oziroma kosmata premija (angl. *gross premium*) in temelji na nevarnostni premiji (angl. *pure premium*), ki je neposredno povezana s tveganostjo zavarovanca (Chapados, 2010, str. 6) ter predstavlja pričakovano vrednost izplačil zaradi škodnih dogodkov (Dugas et al., 2003, str. 180). Kosmato premijo dobimo iz nevarnostne premije tako, da ji prištejemo dodatke za stroške in dobiček (Chapados, 2010, str. 6). Vendar pa je potrebno pri njenem oblikovanju upoštevati tudi razmere na trgu v zelo tekmovalnem zavarovalniškem poslu ter vpliv premij na stranke – na zadržanje obstoječih in pridobivanje novih (Dugas et al., 2003, str. 181; Smith, Willis, & Brooks, 2000, str. 532). Na problem določanja premij je torej treba gledati celoviteje in se ne omejiti samo na oceno tveganosti zavarovanca. Marketinški pristop k določanju bruto premije se po mojem mnenju kaže že v številnih popustih, ki jih izračun običajno vsebuje, nadalje pa še v možnosti dodatnih komercialnih popustov, ki se lahko ob sklepanju zavarovalne pogodbe dodelijo stranki, da jo prepričajo v sklenitev. Slednje lahko po mojem mnenju za zavarovalnice pomeni nevarnost pri prodaji zavarovanj preko zavarovalnih agentov, če so le-ti plačani s provizijo od prodaje in neodvisno od škodnega rezultata ter želijo zato zavarovanje prodati za vsako ceno.

Opisani problem je splošne narave, jaz pa ga v tem magistrskem delu obravnavam na primeru avtomobilskih premoženjskih zavarovanj, ki jih imenujemo tudi avtomobilski kasko in zato pogosto označujemo s kratico AK. To skupino zavarovanj sem izbral, ker je po podatkih izbrane zavarovalnice (glede na obračunano kosmato premijo ter glede na obračunano kosmato odškodnino) ena najbolj množičnih. To po eni strani pomeni veliko količino podatkov, nad katerimi sem lahko izvedel podatkovno rudarjenje, po drugi strani pa tudi, da vsaka izboljšava obstoječih postopkov, ki bi jo dosegli z uporabo izsledkov moje raziskave, lahko prinese večje finančne koristi. Pomembnost zavarovanj AK v portfelju izbrane zavarovalnice nadalje pomeni tudi, da so zelo natančno razdelana in da se v podatkovni bazi vodijo številni različni parametri, ki so vezani na te zavarovalne pogodbe, kar jih dela še toliko bolj primerne za obravnavo s podatkovnim rudarjenjem. Vendar pa pričakovane velike finančne koristi vsake izboljšave po drugi strani pomenijo tudi, da so ta zavarovanja stalna tarča poskusov izboljšav znotraj zavarovalnice, kar je za mojo raziskavo pomenilo izziv in težavo, kajti iluzorno je bilo pričakovati, da bom zlahka dosegel bistvene izboljšave.

Pojem podatkovnega rudarjenja se po mojih opažanjih pogosto povezuje in tudi zamenjuje s pojmom strojno učenje (angl. *machine learning*). Vendar pa je podatkovno rudarjenje širši proces, ki strojnega učenja nujno ne zajema, lahko pa za izgradnjo modela, ki je ena od faz podatkovnega rudarjenja (Chapman et al., 2000, str. 10), uporabimo tudi algoritme strojnega učenja; za uporabo tega, po mojem mnenju najbolj običajnega, pristopa sem se odločil tudi sam.

Namen mojega dela je opis izvedbe procesa podatkovnega rudarjenja z uporabo algoritmov strojnega učenja na podatkih avtomobilskega premoženjskega zavarovanja izbrane slovenske

zavarovalnice, s čemer želim prispevati k uveljavitvi tega pristopa. V svojem delu podajam celovit pregled uporabe tega postopka za določanje cen omenjenih zavarovalnih produktov, tako s stališča nevarnostne premije oziroma tveganosti zavarovanca kot s stališča tržne cene. Hkrati želim izpostaviti morebitne točke, kjer bi lahko z uporabo metod, ki sem jih izbral, dosegli izboljšanje obstoječega stanja in jih torej praktično uporabili. Kot končni rezultat predlagam tudi postopek, ki povezuje rezultate izvedenega procesa podatkovnega rudarjenja v celoto ter jih prenaša v praktično uporabo. Namen omenjenega postopka ni v zamenjavi obstoječih aktuarskih metod, ampak v njihovi dopolnitvi in vzporedni uporabi s ciljem povečanja dobička zavarovalnice. Vodilo mojega dela je torej praktična usmeritev ter čim bolj neposredna pot do vključitve rezultatov v poslovno okolje, zato v njem manj poudarka namenjam teoretičnim osnovam ter primerjavam različnih postopkov, ampak se pri tem opiram tudi na izsledke drugih avtorjev.

Opisan namen sem želel uresničiti skozi različne cilje. Prvi cilj je bil, da z uporabo podatkovnega rudarjenja na podatkih izbrane slovenske zavarovalnice zgradim model za ocenjevanje oziroma napovedovanje pričakovane vrednosti izplačil zaradi škodnih dogodkov, ki predstavlja nevarnostno premijo. Le-tega sem želel doseči s kombinacijo modelov za napovedovanje verjetnosti nastopa škodnega dogodka ter za napovedovanje višine odškodnine ob škodnem dogodku ter pri izgradnji teh modelov uporabiti algoritme strojnega učenja, ki so se v sorodnih raziskavah izkazali za uspešne. Kot stranski produkt procesa sem želel tudi odkriti atribute zavarovanca in zavarovalne pogodbe, ki imajo pomemben vpliv na tveganost, ter jih ločiti od tistih, ki ga nimajo. S primerjavo izgrajenega modela z obstoječim aktuarskim modelom sem želel ovrednotiti njegovo kvaliteto. Naslednji cilj je bil, da z uporabo podatkovnega rudarjenja razvijem model, ki bi posnemal delovanje zavarovalnega agenta in s katerim bi lahko torej ocenil primerno višino končne tržne cene zavarovalne pogodbe v nekem konkretnem primeru. Nadalje sem želel zasnovati postopek, s katerim bi lahko kombinacijo obeh modelov uporabili pri prodaji zavarovanj. Na podlagi rezultatov, ki sem jih dobil z uporabo izgrajenih modelov, sem želel odkriti ter opisati njihove prednosti in slabosti v primerjavi z obstoječim stanjem ter predlagati možnosti uporabe v praksi. Eden izmed mojih ciljev je bil tudi, da dokumentiram in opišem težave, na katere sem tekom raziskave naletel, ter podam praktične predloge za rešitve in s tem pomagam tako raziskovalcem kot tudi poslovnim uporabnikom, ki se bodo v prihodnosti srečali s podobnimi problemi.

Cilje magistrskega dela sem poskušal doseči s praktično uporabo postopka podatkovnega rudarjenja na podatkih izbrane slovenske zavarovalnice. Pri tem sem se opiral na izsledke drugih avtorjev, ki sem jih črpal iz literature v obliki knjig in strokovnih člankov, uporabiti pa sem moral tudi znanje, ki sem ga pridobil tekom dodiplomskega in podiplomskega študija ter pri delu, ki ga opravljam. Znotraj procesa podatkovnega rudarjenja sem uporabil algoritme strojnega učenja. Magistrsko delo je v grobem razdeljeno na dva dela. V prvem delu podajam opis področja raziskovanja, definicije vseh pomembnih pojmov, nekaj teoretičnih osnov raziskave in opise obstoječih algoritmov ter postopkov, ki sem jih v svoji raziskavi uporabil in so plod dela različnih drugih avtorjev. Drugi del vsebuje opis poteka postopka podatkovnega rudarjenja v konkretnem primeru ter predstavitev in razlago rezultatov z uporabo različnih metrik in grafičnih

predstavitev; kjer se mi je zaradi razumevanja to zdelo primerneje, navajam določene teoretične osnove, opise in definicije tudi znotraj tega drugega dela, vzporedno z opisom praktične izvedbe postopka.

Magistrsko delo je sestavljeno iz petih glavnih poglavij, ki se dalje delijo na podpoglavja. Prvo poglavje vsebuje opis področja, na katerem je potekala moja raziskava. To je zavarovalništvo, natančneje področje avtomobilskih premoženjskih zavarovanj. Za vsebino moje raziskave je tukaj, ob razlagi pojmov, ki se pojavljajo v nadaljevanju dela, pomemben oris tradicionalnih metod za izračun nevarnostne premije. Drugo poglavje je posvečeno definiciji podatkovnega rudarjenja in vseh pomembnih, z njim povezanih pojmov, opisu postopka ter posameznih faz podatkovnega rudarjenja ter pregledu postopkov, metod, ovir in njihovih rešitev, ki sem jih pri podatkovnem rudarjenju uporabil oziroma sem na njih naletel, nadalje pa še opisu uporabljenih algoritmov strojnega učenja. V tretjem poglavju je opisan obširen proces pridobivanja in priprave podatkov za strojno učenje ter težave, na katere sem pri tem naletel, ter, kako sem jih rešil. Četrto poglavje zajema opis izgradnje modelov za napovedovanje izbranih vrednosti z uporabo algoritmov strojnega učenja na pripravljenih podatkih ter ovrednotenje in razlago rezultatov, ki so bili ob tem doseženi, vzporedno s tem pa tudi opis možnosti za uporabo v praksi ter prednosti in slabosti takšne uporabe. V petem poglavju podajam predlog ter ovrednotenje postopka za kombinirano uporabo izgrajenih modelov s ciljem izboljšanja poslovanja zavarovalnice. Zadnje, sklepno poglavje predstavlja kratek povzetek vsebine predhodnih poglavij ter sklepno misel.

1 ZAVAROVALNE OSNOVE

1.1 Opredelitev pojma zavarovanja in ostalih pomembnih pojmov

Boncelj (1983, str. 13) **zavarovanje** definira kot ustvarjanje gospodarske varnosti z izravnavanjem gospodarskih nevarnosti. Izravnavanje gospodarskih nevarnosti izkorišča zakon velikih števil in poteka s porazdeljevanjem škode v času in prostoru (Močivnik, 2010, str. 28). V ožjem smislu lahko zavarovanje definiramo tudi kot nudenje zavarovalne zaščite (Močivnik, 2010, str. 73). Njegova osnova je **zavarovalna pogodba**, s katero se **zavarovalec** zaveže, da bo **zavarovalnici** (pravna oseba, ki se ukvarja z zavarovalno dejavnostjo) plačal **premijo**, zavarovalnica pa, da bo ob nastopu **zavarovalnega primera** izplačala **zavarovalnino** ali **odškodnino** (ali storila kaj drugega) (Fric & Prijanovič, 2010, str. 6; Močivnik, 2010, str. 70–71). Listina o zavarovalni pogodbi se imenuje **zavarovalna polica** (Močivnik, 2010, str. 70); omenjena pojma bom v nadaljevanju tega magistrskega dela uporabljal kot sinonima. Enako velja tudi za pojma **zavarovalec** (oseba, ki z zavarovalnico sklene zavarovalno pogodbo, lahko bi rekli tudi **sklenitelj**) in **zavarovanec** (oseba, katera ima interes za povrnitev škode ob nastopu zavarovalnega primera), ki sta po eni strani pogosto ista oseba (Močivnik, 2010, str. 69–72), po drugi strani pa razlikovanje med omenjenima vlogama v okviru tega magistrskega dela ni bistvenega pomena (če bo razlikovanje kje potrebno, bom to posebej navedel). Bolj pomembno je zavedanje, da pri avtomobilskih zavarovanjih niti zavarovanec niti zavarovalec ne predstavljata nujno osebe, ki je uporabnik (voznik) vozila, na kar se bom v nadaljevanju še vrnil.

Škodni dogodek Močivnik (2010, str. 61) definira kot negotov in nepričakovan dogodek, ki je neodvisen od volje zainteresiranih oseb in v trenutku deluje na zavarovano stvar ali osebe, definicija **zavarovalnega primera** (Močivnik, 2010, str. 71) pa vsebuje še pogoja, da do dogodka pride v času zavarovalnega kritja in da dogodek uresničuje zavarovane nevarnosti. Škodni dogodek je po tej definiciji torej širši pojem, ki ga lahko uporabljamo neodvisno od obstoja zavarovalne pogodbe. Sam se bom v nadaljevanju takšni delitvi izognil in bom pri uporabi besednih zvez **škodni dogodek** oziroma **škodni primer** oziroma tudi **škoda** vedno imel v mislih tiste škodne dogodke, ki so hkrati tudi zavarovalni primeri. Znesek, ki ga zavarovalnica izplača ob nastanku zavarovalnega primera, se imenuje **zavarovalnina** (Močivnik, 2010, str. 71), med tem ko se izraz **odškodnina** po strogi definiciji nanaša le na znesek, ki je izplačan oškodovancu iz naslova zavarovanja odgovornosti povzročitelja škode (Močivnik, 2010, str. 44). Vendar pa se v praksi po mojih izkušnjah ta delitev redko spoštuje in se izraz **odškodnina** uporablja za vsa izplačila zavarovalnice zaradi zavarovalnih primerov, zato ga bom s takšnim razširjenim pomenom v nadaljevanju tega dela uporabljal tudi sam.

Pomemben sestavni del zavarovalne pogodbe je **zavarovalna vsota**, ki pomeni najvišji možni znesek izplačila po posameznem zavarovalnem primeru oziroma znesek, za katerega jamči zavarovalnica in je praviloma tudi osnova za izračun premije (Fric & Prijanovič, 2010, str. 7; Močivnik, 2010, str. 70). Lahko pa zavarovalna pogodba vključuje tudi pogodbeno določeno **odbitno franšizo**, ki predstavlja soudeležbo zavarovanca pri škodi ter pomeni znesek, ki ga pri poravnavi škode zavarovalnica v vsakem primeru odbije od izplačila in ki je lahko določen absolutno ali v deležu od zavarovalne vsote ali višine škode ali pa celo kot kombinacija različnih oblik (Fric & Prijanovič, 2010, str. 8; Močivnik, 2010, str. 44). Namen odbitne franšize je, da zavarovanca spodbuja k preprečitvi škode, vendar mu vseeno zagotavlja zaščito ob velikih škodah; nadalje se z uvedbo odbitne franšize zavarovalnica izogne stroškom obravnave številnih manjših škod, ki jih zavarovanci lahko izravnajo sami, na podlagi vsega tega pa lahko zavarovancu ponudi nižjo premijo (Fric & Prijanovič, 2010, str. 8).

V zvezi z izplačili ob škodnih primerih je potrebno omeniti še pojem **škodne rezervacije**, ki pomeni rezervacijo za ocenjene bodoče obveznosti zavarovalnice za že nastale škode, ki jih zavarovalnica še ni izplačala; tukaj lahko gre za škode, ki so že bile prijavljene, vendar še ne v celoti izplačane, ali pa za že nastale škode, ki zavarovalnici sploh še niso bile prijavljene (Močivnik, 2010, str. 61). Rezervacijo za prvo obliko bom imenoval tudi **rezervacija po popisu**, saj je škoda že prijavljena in zavarovalnica velikost rezervacije oceni s popisom škode (na primer s popisom poškodovanih avtomobilskih delov, objektov, stvari); to rezervacijo, ki se torej oceni individualno za nek škodni primer, bom v nadaljevanju še omenil. Druga oblika škodne rezervacije bo v okviru tega magistrskega dela manj pomembna, saj se ne ocenjuje na ravni police, ampak agregirano na višji ravni. V praksi prvo obliko označujemo tudi s kratico **RBNS** (angl. *Reported But Not Settled*), drugo obliko pa s kratico **IBNR** (angl. *Incurred But Not Reported*) (Močivnik, 2010, str. 61).

Kot že omenjeno, je dolžnost zavarovalca po zavarovalni pogodbi, da zavarovalnici za riziko, ki ga je prevzela, plača **premijo**. Ta znesek seveda ni namenjen samo nadomeščanju škod, ampak

mora pokriti na primer tudi stroške poslovanja zavarovalnice. Znesek, ki ga plača zavarovalec, imenujemo **kosmata premija** in je sestavljena iz **obratovalnega dodatka** (tudi režijski, stroškovni dodatek), ki pokriva stroške zavarovalnega poslovanja, ter **funkcionalne premije**, ki je namenjena uresničevanju funkcij zavarovanja (Boncelj, 1983, str. 19). Slednja se dalje deli na **dodatek za prevencijo in represijo** ter **tehnično premijo**, ki daje sredstva za izravnavanje nevarnosti in je pri premoženjskih zavarovanjih, ki so predmet tega dela, dalje sestavljena iz **nevarnostne premije**, ki v povprečju zadošča za kritje vseh škod iz sprejetih rizikov, in **varnostnega dodatka**, ki je namenjen kritju negativnih odklonov v povprečnem škodnem dogajanju (Boncelj, 1983, str. 19–20; Močivnik, 2010, str. 41, 67).

Chapados (2010, str. 6) podaja nekoliko poenostavljeno delitev premije; kosmato oziroma **bruto premijo** (angl. *gross premium*) (Močivnik, 2010, str. 9) razdeli na že omenjeno **nevarnostno premijo** (angl. *pure premium*) in preostali del, ki je namenjen pokrivanju fiksnih in variabilnih **stroškov** ter **dobička** zavarovalnice. V okviru tega magistrskega dela nas bo zanimala predvsem **nevarnostna premija**, ki je neposredno povezana s **tveganostjo zavarovanca** (angl. *risk level*) in matematično predstavlja **pričakovano vrednost izplačil** (Chapados, 2010, str. 6); ker se z ostalimi deli kosmate premije v tem magistrskem delu ne bom ukvarjal, se mi zdi opisana poenostavljena delitev zadostna.

Omeniti velja še, da je premija, ki se plača na podlagi sklenjene zavarovalne pogodbe, po 6. členu Zakona o davku od prometa zavarovalnih poslov (Ur.l. RS, št. 96/2005-UPB1, v nadaljevanju ZDPZP) predmet davka od prometa zavarovalnih poslov (v nadaljevanju DPZP). Glede na podano definicijo kosmate premije (znesek, ki ga mora zavarovalec plačati) bi po mojem mnenju lahko DPZP vzeli tudi kot del kosmate premije oziroma obratovalnega dodatka, vendar pa gre tukaj za prihodek državnega proračuna (2. člen ZDPZP), zato menim, da tak pogled ni smiseln; v nadaljevanju bom tako dejanski plačani znesek (z vključenim DPZP) imenoval fakturirana premija, v kosmato premijo pa DPZP-ja ne bom štel.

Kosmata premija, ki jo zavarovalnica zaračuna zavarovalcu, se običajno določi na podlagi **premijskega cenika**, ki vsebuje neposredne cene zavarovanj ali premijske stopnje (odvisno od vrste zavarovanja) (Močivnik, 2010, str. 33, 49). **Premijska stopnja** je razmerje med premijo in prevzeto obveznostjo ter je običajno izražena v deležu od **zavarovalne vsote** (premijo v tem primeru na podlagi premijske stopnje torej dobimo tako, da le-to pomnožimo z zavarovalno vsoto) (Močivnik, 2010, str. 49).

Pri določanju oziroma izračunu kosmate premije se lahko na premijo (bodisi iz premijskega cenika bodisi na podlagi premijske stopnje), ki jo bom imenoval tudi »osnovna« premija, dodatno upoštevajo še **popusti**, ki pomenijo zmanjšanje premije oziroma cene ter jih lahko v grobem razdelimo na **tehnične popuste** (sem bom štel tudi popuste zaradi bonus/malus sistema, ki ga predstavljam v podpoglavju 1.2) ter **komercialne popuste** (Močivnik, 2010, str. 48).

Tehnični popusti se dodelijo na podlagi tehničnih značilnosti zavarovanega predmeta ali izvajanja zaščitnih ukrepov zavarovanca, ki imajo za posledico manjše tveganje za zavarovalnico

(Močivnik, 2010, str. 63). Ti popusti so običajno opredeljeni v premijskem ceniku, zato jih bom imenoval tudi **popusti po ceniku** (natančnejšo razčlenitev in popuste, ki sem jih zajel v ta pojem, bom navedel v nadaljevanju).

Komercialni popusti so popusti, ki niso strogo tehnični in niso vključeni v zavarovalne pogoje, ampak jih zavarovalnica odobri na osnovi komercialnega interesa, na primer interesa, da je sklenjeno čim večje število zavarovanj, kar prinaša boljšo razpršitev tveganja (Močivnik, 2010, str. 32). Omenjeni komercialni interes lahko opišemo kot željo, da zavarovanec zavarovanja ne sklene pri konkurenčni zavarovalnici, torej so komercialni popusti podvrženi konkurenci na trgu in kosmata premija, ki jo plača zavarovanec, predstavlja **tržno ceno** tega zavarovalnega kritja. Ker komercialni popusti v splošnem niso fiksni oziroma odvisni od nekega pravila, ampak jih običajno oseba, ki v imenu zavarovalnice sklene zavarovalno pogodbo, dodeli po lastni presoji z namenom, da se stranka odloči za sklenitev, jih bom imenoval tudi **variabilni** komercialni popusti.

Škodni rezultat Močivnik (2010, str. 19) definira kot razliko med premijo in škodami, kar pa se ne sklada s 5. točko 2. člena Navodila za izračun kazalnikov (Ur.l. RS, št. 36/2003), ki ga je izdala Agencija za zavarovalni nadzor (v nadaljevanju AZN), v katerem je definiran kot **razmerje med škodami in premijo** (to količino Močivnik (2010, str. 61) imenuje škodni količnik). V nadaljevanju bom uporabljal definicijo AZN, saj se mi zdi bolj prisotna v splošni rabi. Škodni rezultat (količnik) imenujemo **enostavni**, kadar upoštevamo samo obračunane odškodnine in premije, **merodajni**, če upoštevamo še spremembe škodne rezervacije in prenosne premije, ter **kombinirani** merodajni, če dodatno upoštevamo še stroške zavarovalnice (Močivnik, 2010, str. 19, 32, 37). Zadnjih dveh oblik v tem magistrskem delu ne uporabljam, zato se ne bom spuščal v podrobnosti izračuna.

Za nadaljevanje je pomemben še pojem **pro rata temporis**, ki pomeni način obračuna zavarovalne premije po načelu premosorazmernosti, in sicer se letna premija deli s številom dni v letu, rezultat pa se nato pomnoži z dejanskim številom dni kritja (Močivnik, 2010, str. 51). Ta način se na primer uporablja pri povračilu premije ob predčasni prekinitvi zavarovanja (na primer zaradi spremembe lastništva vozila) (Fric & Prijanovič, 2010, str. 19; Močivnik, 2010, str. 51). V nadaljevanju ga bom skrajšano imenoval kar pro-rata način, dobljeno premijo pa pro-rata premija.

Kot opombo naj omenim, da ima zavarovalnica možnost, da presežke iznad stopnje lastnega izravnavanja zavaruje pri drugi zavarovalnici, ki ima dovoljenje in je registrirana za opravljanje takih poslov ter jo imenujemo pozavarovalnica, opisano zavarovanje pa **pozavarovanje** (Močivnik, 2010, str. 48–49); ker tega dela poslovanja zavarovalnice v tem magistrskem delu ne obravnavam, ga ne bom podrobneje opisoval.

1.2 Avtomobilska zavarovanja

Zakon o zavarovalništvu (Ur.l. RS, št. 93/2015, v nadaljevanju ZZavar-1), ki v svojem 7. členu podaja delitev zavarovanj v zavarovalne vrste, pojma avtomobilskega zavarovanja neposredno ne pozna. Gre torej za pogovorni izraz, pod katerim pa (kot bo kmalu razvidno) lahko razumemo marsikaj. Zato je nujno, da pred nadaljevanjem podrobneje razčlenimo avtomobilska zavarovanja in opredelimo pojem avtomobilskega premoženjskega zavarovanja in s tem predmet raziskave, kot ga bom uporabljal v okviru tega magistrskega dela.

Fric in Prijanovič (2010, str. 14) avtomobilska zavarovanja razdelita na dve večji skupini, in sicer:

- zavarovanje odgovornosti za škodo, ki jo kot lastnik cestnega motornega vozila povzročimo tretjim osebam in
- premoženjsko zavarovanje, ki je namenjeno kritju škode na lastnem vozilu.

Prvo skupino imenujemo **zavarovanje avtomobilske odgovornosti** (v nadaljevanju zavarovanje **AO**), drugo pa **zavarovanja avtomobilskega kaska** (v nadaljevanju zavarovanje **AK**) (Fric & Prijanovič, 2010, str. 14). K tem zavarovanjem lahko priključimo še zavarovanje voznika za škodo zaradi telesnih poškodb (imenovano tudi AO Plus), nezgodno zavarovanje voznika in sopotnikov, zavarovanje avtomobilske asistence ter zavarovanje pravne zaščite (Fric & Prijanovič, 2010, str. 14).

Čeprav gre pri zavarovanju AO za zavarovanje odgovornosti, je treba poudariti, da je tudi to premoženjsko zavarovanje, pri katerem pa ne gre za zavarovanje stvari ali osebnih dobrin zavarovanca, temveč za zavarovanje njegovega premoženja, ki bi bilo ogroženo, če bi moral zaradi svoje odgovornosti za škodo tretji osebi plačati odškodnino (Ristin et al. v Fric & Prijanovič, 2010, str. 15; Močivnik, 2010, str. 73–74). Posredno pa so med drugim tudi tukaj zavarovane stvari, vendar ne zavarovančeve, ampak v lasti tretjih oseb.

Sam bom za namene tega magistrskega dela uporabil ožji pogled na premoženjska zavarovanja, in sicer bom obravnaval samo zavarovanje zavarovančevega premoženja v obliki stvari. S pojmom **avtomobilska premoženjska zavarovanja** bom v okviru tega magistrskega dela torej mislil zavarovanja AK, le-ta pa predstavljajo tudi predmet moje raziskave. Po 7. členu ZZavar-1 spadajo ta zavarovanja v 3. zavarovalno vrsto, ki pokriva zavarovanja kopenskih vozil (razen tirnih). Zavarovanje odgovornosti pri uporabi vozil (10. zavarovalna vrsta po 7. členu ZZavar-1) sem iz neposredne obravnave (s katero mislim določanje premij) torej izključil, prav tako pa tudi že omenjena zavarovanja, ki jih lahko priključimo zavarovanjem AO in AK (AO Plus, zavarovanje asistence, …); sem pa ta zavarovanja upošteval posredno, in sicer kot dodatno informacijo o profilu zavarovanca na zavarovanju AK.

Zavarovanje AK se pri večini slovenskih zavarovalnic še dalje deli, in sicer na t.i. **polni kasko** (polni AK) ter **delni kasko** (delni AK) (Fric in Prijanovič, 2010, str. 31). Pri delnem AK se lahko

vozilo zavaruje pred posameznimi nevarnostmi, pri čemer so le-te pri večini zavarovalnic združene v skupine, ki jih imenujemo **kombinacije delnega AK**, zavarovanec pa lahko sklene eno ali več izmed ponujenih kombinacij (Fric & Prijanovič, 2010, str. 33). Tako pri polnem kot pri delnem AK se zavarovane nevarnosti, ki jih zavarovanje krije in so lahko oblikovane v različne ponudbe oziroma **pakete**, od zavarovalnice do zavarovalnice razlikujejo, enako pa velja za sestavo kombinacij delnega AK (Fric & Prijanovič, 2010, str. 32–33).

Med kritimi nevarnostmi polnega AK bi kot najpomembnejše omenil prometno nesrečo, požarne nevarnosti (vključujejo delovanje naravnih sil) ter zlonamerna ali objestna dejanja tretjih oseb (Fric & Prijanovič, 2010, str. 32). Tipične nevarnosti, ki se lahko zavarujejo v okviru delnih AK kombinacij, pa so dotik divjadi ali domačih živali, razbijte stekla, dotik neznanega vozila s parkiranim zavarovanim vozilom, tatvina vozila (nekatere zavarovalnice imajo to nevarnost zajeto v polnem AK) ter zavarovanje stroškov najema nadomestnega vozila (Fric & Prijanovič, 2010, str. 32–34).

Zavarovanje AK se lahko sklene z odbitno franšizo ali brez (Fric & Prijanovič, 2010, str. 32). Je pa to odvisno tudi od pogojev posamezne zavarovalnice in izbrane kombinacije kritij, saj obe opciji namreč nista vedno na voljo.

Za sklenitev delnega AK ni nujno, da ima zavarovanec sklenjen polni AK, lahko pa delne AK kombinacije priključi polnemu AK; če zavarovanec sklene polni AK in dodatno še delni AK za nevarnost, ki je krita že v okviru polnega AK, se s tem v primeru nastopa škodnega primera zaradi te nevarnosti izogne odbitni franšizi (razen seveda, če ima sklenjen delni AK z odbitno franšizo) ter izgubi bonusa na polnem AK (opredelitev bonusa podajam v nadaljevanju) (Fric & Prijanovič, 2010, str. 33).

Pomemben element avtomobilskih zavarovanj, ki se uporablja tako pri zavarovanju AO kot pri zavarovanju polnega AK, je **sistem bonus/malus**, ki pomeni določanje zavarovalne premije ob upoštevanju števila prijavljenih škod; pri tem se uporabljajo t.i. **premijski razredi**, ki določajo odstotek izhodiščne premije, ki jo mora zavarovanec plačati (Fric & Prijanovič, 2010, str. 23, 33). Ob prvi sklenitvi je le-ta uvrščen v premijski razred, ki pomeni 100 % izhodiščne premije; če v tem zavarovalnem letu ne prijavi škode, se v naslednjem zavarovalnem letu razvrsti v nižji premijski razred, ki pomeni na primer 90 % izhodiščne premije (tukaj govorimo o **bonusu**), v nasprotnem primeru pa se razvrsti v višji premijski razred, ki pomeni na primer 120 % izhodiščne premije (**malus**) (Fric & Prijanovič, 2010, str. 23). Na bonus lahko torej gledamo kot na dodatni popust, na malus pa kot na doplačilo. Podajanju natančne lestvice premijskih razredov in pravil za prehode (za koliko premijskih razredov napreduje oziroma nazaduje zavarovanec, ki ni prijavil škode oziroma je prijavil škodo) (oboje podajata Fric in Prijanovič, 2010, str. 23) se bom na tem mestu izognil, saj se lahko med zavarovalnicami razlikujejo in s časom spreminjajo; so pa sistemi po mojih izkušnjah zelo podobni, verjetno tudi zato, ker zavarovalnice omogočajo prenos bonusa iz ene na drugo (če se zavarovanec odloči zamenjati zavarovalnico). Pomembno pa se mi zdi poudariti, da je lestvica navzgor in navzdol načeloma omejena (tipično med 50 % in 200 %) (Fric & Prijanovič, 2010, str. 23, 33).

V sklopu sistema bonus/malus je zelo pomembno omeniti še možnost sklenitve zavarovanja za **odkup prve škode**, ki mu pravimo tudi **zavarovanje izgube bonusa** in katerega bom v nadaljevanju še večkrat omenil; pri njem gre preprosto za to, da zavarovanec obdrži isti premijski razred kljub prijavi ene škode v preteklem zavarovalnem letu (Fric & Prijanovič, 2010, str. 24).

1.3 Tradicionalne metode za določanje premij pri AK zavarovanjih in povezani problemi ter primerjava z algoritmi strojnega učenja

Določanje premij je eden glavnih matematičnih problemov, s katerimi se srečujejo aktuarji; le-ti morajo najprej določiti pričakovano vrednost izplačil za škodne primere, ki predstavlja nevarnostno premijo, ki je, kot sledi tudi iz podpoglavja 1.1, osnova za izračun končne cene zavarovalne pogodbe (Dugas et al., 2003, str. 180). Omenjena pričakovana vrednost odškodnin je odvisna od informacij o zavarovancu in zavarovalni pogodbi (Dugas et al., 2003, str. 180). Modeli, ki jih pri tem uporabljajo aktuarji, temeljijo tako na statističnih analizah kot na izkušnjah, ekspertnem znanju in človeškem vpogledu (Apte et al., 1998, str. 1).

Osnova tradicionalnih metod je delitev prostora vhodnih parametrov o zavarovancu in zavarovalni pogodbi (na primer starost, spol, letno število prevoženih kilometrov) v skupine s podobnimi značilnostmi tveganja (Apte et al., 1998). Najpreprostejše so metode, ki temeljijo na tabelah (angl. *table-based methods*) (Chapados, 2010, str. 13). Pri teh metodah najprej identificiramo ne preveliko število spremenljivk, za katere menimo, da so primerne za oceno tveganja, ter tiste, ki so zvezne, pretvorimo v diskretne, tako da jih razdelimo v končno število intervalov (Chapados, 2010, str. 13). Različne kombinacije vrednosti diskretnih spremenljivk predstavljajo celice »tabele«, ki je seveda lahko več kot samo dvodimenzionalna (pri treh spremenljivkah recimo iz klasične tabele dobimo kocko) (Chapados, 2010, str. 13). Nato za vsako celico takšne večdimenzionalne tabele izračunamo vsoto izplačanih odškodnin (preko mnogo let) za vse pretekle zavarovalne pogodbe, ki glede na svoje parametre sodijo v določeno celico, ter iz dobljene vsote izračunamo povprečje, ki predstavlja nevarnostno premijo za tisto celico (Chapados, 2010, str. 13). Morebitne ostre prehode v vrednostih nevarnostne premije med sosednjimi celicami lahko zgladimo z uporabo linearne interpolacije (Chapados, 2010, str. 13). Problem takega pristopa je, da postane žrtev **prekletstva dimenzionalnosti** (angl. *curse of dimensionality*): prostor različnih kombinacij vhodnih parametrov raste tako hitro, da nam že ob relativno majhnem številu parametrov hitro zmanjka dovolj velikega števila primerov za posamezno kombinacijo, da bi lahko dobili zanesljive statistične ocene, celice z malo opazovanimi primeri pa so tudi bolj občutljive na posamezne redke, a zelo visoke škode, ki jih imenujemo **osamelci** (angl. *outliers*) (Chapados, 2010, str. 13). Če se poskusimo temu izogniti in se odločimo za manjše število spremenljivk ter tako zmanjšamo število celic, pa postane funkcija, ki preslika parametre zavarovalne pogodbe v nevarnostno premijo, preveč groba in premalo natančna predstavitev realnosti (Chapados, 2010, str. 13). Število celic lahko zmanjšamo tudi z uporabo več neodvisnih tabel, katerih rezultate nato združimo v končno oceno nevarnostne premije; težava te metode pa je, da ne zajame medsebojnih vplivov spremenljivk, ki se ne pojavljajo v isti tabeli, ampak jih upošteva kot neodvisne (Chapados, 2010, str. 14).

Pravkar zapisano dobro povzemajo tudi Dugas et al. (2003, str. 180–181) – za ustrezno oceno je relevantnih mnogo dejavnikov, pri tem je nevarno predpostavljati njihovo neodvisnost, upoštevati vse odvisnosti pa neobvladljivo (že omejeno prekletstvo dimenzionalnosti). Pri tradicionalnih metodah je aktuar tisti, ki mora odkriti najpomembnejše odvisnosti ter jih neposredno vnesti v model (Dugas et al., 2003, str. 181).

Delitev prostora je značilna tudi za modele, ki so zgrajeni na osnovi odločitvenih dreves; le-ti tako tudi trpijo za prekletstvom dimenzionalnosti (Chapados, 2010, str. 15). Manj težav s tem imajo posplošeni linearni modeli (angl. *generalized linear models*, v nadaljevanju GLM), ki pa so precej omejeni glede oblike funkcij, ki jih lahko predstavljajo (Chapados, 2010, str. 15).

Še en problem je prisotnost zelo visokih škod, ki se redko pojavljajo (osamelci) in ki jih vseeno nikakor ne smemo izločiti (kot bi to storili pri tradicionalnih robustnih statističnih metodah), saj vsebujejo pomembno informacijo o pričakovani višini odškodnin (Chapados, 2010, str. 16). Z drugimi besedami, porazdelitev škod je asimetrična z debelimi repi (angl. *fat tails*): vsebuje veliko število ničel (zavarovalne pogodbe brez škod) in nekaj nezanesljivih in zelo visokih vrednosti (Dugas et al., 2003, str. 181). Težava tukaj je v tem, da so vse te visoke vrednosti na isti strani povprečja (asimetrična porazdelitev), tako da jih ne moremo izločiti ali jim zmanjšati težo vpliva, kot bi to lahko storili pri simetrični porazdelitvi, kajti s tem bi v oceno vnesli močno pristranskost (angl. *bias*); z izločitvijo osamelcev bi bila ocena pričakovane vrednosti sistematično podcenjena (Dugas et al., 2003, str. 181). Ti primeri tudi povzročajo velike težave metodam, ki temeljijo na delitvi prostora, saj lahko zelo spremenijo rezultat v določeni celici, hkrati pa je ta vpliv omejen samo na tisto celico, četudi so ji sosednje po vhodnih parametrih podobne (Chapados, 2010, str. 16–17). Na težavo niso odporni niti modeli GLM, saj lahko že prisotnost nekaj osamelcev bistveno spremeni rezultat (Chapados, 2010, str. 17).

Možna rešitev za opisane težave so algoritmi strojnega učenja (Chapados, 2010, str. 19), še posebej dobri rezultati na področju ocenjevanja zavarovalnih premij so bili doseženi z uporabo nevronskih mrež (Chapados, 2010, str. 20), ki so znane po svoji zmožnosti predstavitve nelinearnih odvisnosti z razmeroma majhnim številom parametrov (teoretično podlago za to podajajo Dugas et al., 2003, str. 192); lahko bi rekli, da same zaznajo najbolj pomembne odvisnosti (Dugas et al., 2003, str. 181), tako da le-teh ni potrebno odkriti z oceno aktuarja in neposredno vnesti v model. Hkrati se nevronske mreže tudi uspešno spopadajo s problemom debelih repov (Dugas et al., 2003, str. 201). Eno izmed možnih rešitev za problem asimetričnih porazdelitev z debelimi repi (angl. *asymmetric heavy-tail distribution*) pa predstavlja tudi algoritem RRAT (angl. *Robust Regression for Asymmetric Tails*) (Takeuchi, Bengio, & Kanamori, 2002).

2 PODATKOVNO RUDARJENJE

Podatkovno rudarjenje (angl. *data mining*) se ukvarja z iskanjem skritih povezav, ki so prisotne v poslovnih podatkih, z namenom ustvarjanja napovedi za uporabo v prihodnosti (Sumathi & Sivanandam, 2006, str. 5). Je proces, ki skuša na podlagi podatkov iz velikih

podatkovnih baz izluščiti uporabne informacije, ki na prvi pogled niso očitne (Sumathi & Sivanandam, 2006, str. 5). Pridobljene informacije lahko služijo za pridobitev konkurenčne prednosti in kot pomoč pri ključnih poslovnih odločitvah (Sumathi & Sivanandam, 2006, str. 9). Pri tem se uporabljajo statistične metode, strojno učenje, umetna inteligenca (angl. *artificial intelligence – AI*) ter metode vizualizacije podatkov (Sumathi & Sivanandam, 2006, str. 8).

2.1 Faze postopka podatkovnega rudarjenja

Proces podatkovnega rudarjenja oziroma njegov življenjski cikel (angl. *life cycle*) lahko razdelimo v več faz; le-te sistematično obravnava **CRISP-DM** pristop oziroma model (angl. *CRoss-Industry Standard Process for Data Mining*), ki predvideva naslednjih 6 faz (Chapman et al., 2000, str. 10):

1. razumevanje problema (angl. *business understanding*),
2. razumevanje podatkov (angl. *data understanding*),
3. priprava podatkov (angl. *data preparation*),
4. modeliranje (angl. *modeling*),
5. ovrednotenje modela (angl. *evaluation*),
6. vključitev v uporabo (angl. *deployment*).

Čeprav so faze navedene v vrstnem redu, to zaporedje ni trdno določeno; še več, vračanje na predhodne faze in ponovno napredovanje sta skoraj vedno zahtevana (Chapman et al., 2010, str. 10). Podatkovno rudarjenje je po svoji naravi torej cikličen proces (Chapman et al., 2010, str. 10).

V svoji raziskavi se nisem že v naprej odločil, da se bom strogo držal omenjenih faz, vendar pa se je v praksi moje delo z njimi dokaj dobro prekrivalo. Večkrat sem se moral tudi vračati na pretekle faze. Na podlagi tega menim, da CRISP-DM model dobro predstavlja najbolj naravni pristop k procesu podatkovnega rudarjenja.

Posamezne faze se zrcalijo tudi v strukturi tega magistrskega dela; izvedba celotnega procesa podatkovnega rudarjenja na praktičnem primeru je namreč bila glavno vodilo moje raziskave. Razumevanje problema je obdelano v uvodnem poglavju, prav tako pa bi sem uvrstil prvo in drugo poglavje. Tretje poglavje pokriva razumevanje ter pripravo podatkov, četrto poglavje pa modeliranje ter ovrednotenje modela. Vključitev v uporabo obravnavam predvsem v petem poglavju, se je pa na primernih mestih dotaknem tudi že v predhodnih poglavjih.

2.2 Strojno učenje

Proces odkrivanja vzorcev v podatkih, ki predstavlja podatkovno rudarjenje, mora biti avtomatski ali vsaj polavtomatski, odkriti vzorci pa morajo imeti pomen v smislu, da nam prinesejo neko prednost, običajno ekonomsko (Witten, Frank, & Hall, 2011, str. 5). Tehnično

osnovo tega procesa predstavlja **strojno učenje** (angl. *machine learning*), ki se uporablja, da izlušči informacije iz surovih podatkov v podatkovnih bazah (Witten et al., 2011, str. xxi). V okviru svoje raziskave sem ga uporabil v fazi modeliranja, ki jo predstavljam v četrtem poglavju.

2.2.1 Definicije

Vhod v sistem strojnega učenja je množica **primerov** (angl. *instances*) (Witten et al., 2011, str. 42). Omenjeno množico bom imenoval tudi **nabor podatkov**, pri čemer pa s tem ne bom vedno mislil striktno samo množice, ki je bila poslana v sistem strojnega učenja, ampak lahko tudi širše oblike te množice, ki pa sem jih zaradi različnih (v nadaljevanju sproti pojasnjenih) vzrokov kasneje skrčil, da sem dobil tisto končno množico, ki sem jo poslal v sistem strojnega učenja. Ker je v okviru moje raziskave po en primer iz nabora podatkov predstavljala ena zavarovalna pogodba oziroma polica, bom nabor podatkov včasih označeval tudi kot nabor polic. Posamezne primere bom v nadaljevanju imenoval tudi **učni** ali **testni** primeri, če bom želel poudariti, da so namenjeni učenju sistema ali pa preizkusu uspešnosti naučenega sistema.

Vsak primer iz nabora podatkov je opredeljen s svojimi vrednostmi za določene lastnosti oziroma značilnosti (na primer spol, starost); te značilnosti imenujemo **atributi** (angl. *attributes*), vrednosti atributov za posamezen primer pa so meritve količine oziroma lastnosti, ki jo atribut predstavlja (na primer moški, ženska, 15 let, 32 let) (Witten et al., 2011, str. 49). Pri tem je množica atributov fiksna in v naprej definirana (Witten et al., 2011, str. 49). Nabor podatkov si tako lahko predstavljamo kot tabelo (Tabela 1), v kateri posamezni primeri predstavljajo vrstice tabele, atributi njene stolpce, vrednosti atributov za posamezen primer pa so v celicah tabele (Palfy, 2009, str. 11–12; Witten et al., 2011, str. 49).

Količine, ki jih vrednosti atributov merijo, in s tem tudi same atribute lahko v grobem razdelimo na **numerične** (angl. *numeric*) in **nominalne** (angl. *nominal*) (Witten et al., 2011, str. 49). Pri numeričnih (na primer starost, višina) gre za številske vrednosti (cela ali realna števila), nominalni atributi (na primer spol, proizvajalec vozila) pa zavzemajo vrednosti iz v naprej definirane, končne množice vrednosti, pri čemer te vrednosti služijo kot imena za neke lastnosti (Witten et al., 2011, str. 49). Numerične atribute včasih imenujejo tudi zvezni (angl. *continuous*) (čeprav cela števila niso zvezna v matematičnem pomenu besede), nominalne pa **kategorični** (angl. *categorical*), lahko tudi diskretni (angl. *discrete*) (Witten et al., 2011, str. 49, 51). Primere različnih vrst atributov prikazuje Tabela 1 (vrsta atributa je navedena v drugi vrstici).

Obstajajo še nadaljnje delitve (Witten et al., 2011, str. 49–50), ki pa jih v okviru tega magistrskega dela nisem uporabljal, saj za mojo raziskavo niso bile pomembne, zato jih zaradi preglednosti tukaj tudi ne bom posebej navajal in opisoval. Omenil bi samo dve podvrsti kategoričnih atributov, in sicer **ordinalne** (angl. *ordinal*) ter **binarne**.

Ordinalni atributi se od ostalih nominalnih ločijo po tem, da lahko različne izmed možnih vrednosti uredimo v nek vrstni red, na primer hladno < mlačno < toplo, pri čemer je bistvo v tem, da vrednost »mlačno« leži med »hladno« in »toplo« (Witten et al., 2011, str. 50). Omenil

bi, da gledajo nekatere delitve (Palfy, 2009, str. 14) na nominalne atribute kot na podvrsto kategoričnih ter jih strogo ločijo od ordinalnih; sam v tem magistrskem delu izraz nominalni atributi uporabljam kot sinonim za kategorične atribute, ordinalni atributi pa so po takšni razdelitvi podvrsta nominalnih.

Binarni atributi, ki jih imenujemo tudi Boolovi (angl. *Boolean*), so poseben primer nominalnih atributov, pri katerem imamo samo dve možni vrednosti, najpogosteje drži (angl. ***true***) in ne drži (angl. ***false***) (lahko tudi da in ne ali 1 in 0) (Witten et al., 2011, str. 51). Če binarni atribut zakodiramo s številskima vrednostima 1 in 0, lahko nanj gledamo tudi kot na numerični atribut, čeprav je po vsebini bližje nominalnim, kajti zavzame lahko samo dve v naprej določeni vrednosti. To podrobnost omenjam zato, ker določeni algoritmi strojnega učenja ne podpirajo uporabe nominalnih atributov, zaradi česar je treba le-te pretvoriti v numerične, pri čemer se pogosto uporablja pretvorba v več binarnih atributov, kot bom opisal v nadaljevanju.

Tabela 1: Ilustrativni primer (izmišljene vrednosti) skrčenega nabora podatkov z nekaj izbranimi atributi ter dodanimi opisi atributov

Starost voznika (v letih)	**Zavarovanje AO Plus**	**Moč motorja vozila (v kW)**	**Barva vozila**	**Vrednost vozila (v EUR)**	**Znesek izplačanih odškodnin (v EUR)**
numerični	**binarni**	**numerični**	**nominalni**	**numerični**	**numerični**
vhod	**vhod**	**vhod**	**vhod**	**vhod**	**izhod**
35	da	65	črna	17.040	0,00
23	ne	45	zelena	12.500	2.356,12
46	ne	89	bela	23.000	0,00
25	da	80	rdeča	22.355	0,00
33	ne	90	siva	16.515	12.715,00
55	da	120	črna	35.000	0,00

Strojno učenje lahko v grobem razdelimo na **nadzorovano** (angl. *supervised*) in **nenadzorovano** (angl. *unsupervised*); cilj nadzorovanega strojnega učenja je napovedati vrednost izhoda na podlagi vhodnih vrednosti, pri nenadzorovanem strojnem učenju pa ni izhodne vrednosti, ki bi jo napovedovali, ampak je cilj opisati povezave in vzorce v množici vhodnih vrednosti (Hastie, Tibshirani, & Friedman, 2009, str. xi).

Nadzorovano strojno učenje je ime dobilo zaradi tega, ker je sistemu predstavljena izhodna vrednost učnega primera in tako na nek način deluje pod nadzorom; uspeh takega učenja lahko ocenimo tako, da naučeni sistem preizkusimo na neodvisni množici primerov (testni primeri), za katere sicer poznamo resnično vrednost izhoda, ki pa je ne razkrijemo stroju (Witten et al., 2011, str. 40). Vhodne vrednosti v sistemu nadzorovanega strojnega učenja imenujemo tudi **neodvisne spremenljivke** (angl. *independent variables*), izhodno vrednost pa **odvisna spremenljivka** (angl. *dependent variable*) (Hastie et al., 2009, str. 9). Te spremenljivke torej predstavljajo atribute; posledično bom v nadaljevanju pojma spremenljivka in atribut uporabljal izmenjaje. Tabela 1 predstavlja primer nabora podatkov, če bi želeli na podlagi značilnosti zavarovanca in zavarovalne pogodbe (neodvisne spremenljivke) z nadzorovanim strojnim učenjem zgraditi

sistem za napovedovanje zneska odškodnin, ki bodo na taki zavarovalni pogodbi izplačane, ta znesek (zadnji stolpec tabele Tabela 1) je v tem primeru odvisna spremenljivka.

Kot enega od primerov nenadzorovanega strojnega učenja naj omenim grupiranje (angl. *clustering*), kjer iščemo skupine primerov, ki so si podobni in sodijo skupaj (Sumathi & Sivanandam, 2006, str. 44; Witten et al., 2011, str. 40). Obstajajo sicer tudi še druge manjše podskupine strojnega učenja, kot na primer učenje z ojačitvijo, transdukcija in delno nadzorovano učenje (angl. *semisupervised*) (Palfy, 2009, str. 8–9); le-teh, prav tako kot nenadzorovanega strojnega učenja, tukaj ne bom nadalje opisoval in navajal primerov teh oblik, saj sem v tem magistrskem delu uporabljal izključno **nadzorovano strojno učenje**.

Pri nadzorovanem strojnem učenju lahko vpeljemo dodatno delitev glede na tip izhodne vrednosti: če je izhodna vrednost kategoričnega (nominalnega, diskretnega) tipa, govorimo o **klasifikaciji** (angl. *classification*), kadar pa je izhodna vrednost numerična (zvezna), govorimo o **regresiji** (angl. *regression*) (Hastie et al., 2009, str. 10; Palfy, 2009, str. 10).

Pri klasifikaciji imenujemo vrednost odvisne spremenljivke **razred** (angl. *class*) določenega primera iz nabora podatkov (Witten et al., 2011, str. 40). Vse različne možne vrednosti odvisne spremenljivke v primeru klasifikacije torej predstavljajo množico razredov, mi pa želimo napovedovati razred, ki mu pripada posamezen primer. Večkrat se zgodi, da imamo samo dva razreda, takrat govorimo o **dvorazrednem problemu** ali tudi binarni klasifikaciji (angl. *two-class, binary classification*) (nasprotje bi bilo angl. *multiclass*). Primeri »vprašanj«, ki predstavljajo dvorazredne probleme, so na primer, ali ima pacient bolezen A, ali je transakcija prevara, ali bo na zavarovalni pogodbi prišlo do škode in podobno.

2.2.2 Algoritmi nadzorovanega strojnega učenja

Označimo z X vektor neodvisnih spremenljivk, z X_j posamezno neodvisno spremenljivko iz vektorja X, z Y pa odvisno spremenljivko. Posamezne opazovane vrednosti teh spremenljivk, ki so vrednosti atributov posameznega primera iz nabora podatkov, označimo z malimi črkami, in sicer x_i in y_i, če govorimo o i-tem primeru. Pri nadzorovanem strojnem učenju pošljemo opazovane vrednosti x_i v umetni sistem, običajno računalniški program, ki ga imenujemo **učni algoritem** (angl. *learning algorithm*) (Hastie et al., 2009, str. 29). Ta algoritem na podlagi vhodnih vrednosti x_i izračuna izhodne vrednosti $f(x_i)$, pri čemer je lastnost učnega algoritma, da lahko spreminja preslikavo f na podlagi razlike med izračunanimi izhodnimi vrednostmi $f(x_i)$ ter dejanskimi opazovanimi vrednostmi odvisne spremenljivke y_i, ta proces je znan pod imenom učenje s primerom (angl. *learning by example*) (Hastie et al., 2009, str. 29). Če bi preslikavo f definirali preprosto kot $f(X)=\beta X$, potem si omenjeno spreminjanje preslikave f, ki ga izvaja učni algoritem, lahko predstavljamo kot spreminjanje koeficientov β (če je X vektor m vhodnih spremenljivk, potem je β vektor m koeficientov).

Rezultat nadzorovanega strojnega učenja je preslikava f, pri čemer upamo, da so izhodne vrednosti, ki jih z uporabo preslikave f izračunamo iz neodvisnih spremenljivk, dovolj blizu

dejanskim vrednostim odvisne spremenljivke, da je preslikava *f* uporabna v praksi (Hastie et al., 2009, str. 29), torej za izračun oziroma napoved vrednosti odvisne spremenljivke v primerih, ko nam njena dejanska vrednost ni znana. Cilj učnega algoritma je torej, da ustvari preslikavo *f* tako, da **generalizira** oziroma posploši znanje, ki ga vsebujejo učni primeri, na nove, pred tem še neznane primere (Palfy, 2009, str. 10).

Dobljena preslikava *f* predstavlja **model**, ki ga lahko uporabimo za napovedovanje vrednosti, ki nas zanimajo, a nam za določene primere niso znane ter bi jih zato želeli napovedati. Proces iskanja preslikave *f*, ki je za naš problem najprimernejša (si zanjo obetamo, da nam bo dala najboljše rezultate, najbolj natančne napovedi na novih primerih) lahko torej imenujemo tudi **modeliranje**; njegov potek v primeru moje raziskave je predstavljen v četrtem poglavju.

Poznamo veliko število različnih metod oziroma postopkov nadzorovanega strojnega učenja, ki nam na podlagi množice učnih primerov ustvarijo preslikavo *f*; te postopke imenujemo **algoritmi strojnega učenja** (isto ime lahko uporabimo tudi pri nenadzorovanem strojnem učenju, le da rezultat algoritma strojnega učenja tam ni preslikava iz neodvisnih v odvisne spremenljivke, ampak je v drugačni obliki). Ti algoritmi se lahko med seboj razlikujejo že po sami obliki preslikave *f* (lahko je to množica pravil, lahko gre za preprosto množenje in seštevanje vhodnih vrednosti ter koeficientov ali pa za bolj zapletene kombinacije različnih preslikav, ki jih izvedemo nad vhodnimi vrednostmi) ali pa po postopku (učnem algoritmu), ki ga uporabljajo za določitev najprimernejših parametrov preslikave *f* (v primeru $f(X)=\beta X$ bi določanje parametrov preslikave *f* pomenilo izračun koeficientov β). Oblika preslikave *f* ter postopek za določitev parametrov preslikave *f* sta lahko fiksna, lahko pa algoritem strojnega učenja omogoča uporabniku, da preko **parametrov** vpliva na obliko preslikave *f* ter na delovanje učnega algoritma.

Pravkar zapisano bo po mojem mnenju še bolj jasno razumljivo na podlagi opisov nekaj različnih algoritmov strojnega učenja, ki jih podajam v nadaljevanju; pri tem sem se omejil na algoritme, ki sem jih tudi praktično uporabil v okviru tega magistrskega dela.

2.2.2.1 Logistična regresija

Ideja logistične regresije temelji na uporabi linearne regresije za reševanje klasifikacijskega problema. Klasifikacijski problem lahko vedno rešujemo z uporabo regresije (kakršnekoli), in sicer ga pretvorimo v toliko regresijskih problemov, kot imamo razredov, pri tem vrednost odvisne spremenljivke postavimo na 1 za tiste učne primere, ki pripadajo temu razredu, za ostale pa jo postavimo na 0 (Witten et al., 2011, str. 125). Pri napovedovanju razreda za nek testni primer izračunamo izhodno vrednost iz naučenih regresijskih modelov za posamezne razrede ter izberemo tisti razred, katerega model nam je dal največjo vrednost (Witten et al., 2011, str. 125). Če pri tem uporabimo linearno regresijo, pri kateri izhodno vrednost izračunamo po linearnem modelu kot $Y = \beta_0 + \sum X_j\beta_j$, govorimo o večodzivni linearni regresiji (angl. *multiresponse linear regression*) (Witten et al., 2011, str. 125). Ena od težav takšnega pristopa, ki v praksi sicer večkrat da dobre rezultate, je, da izhodnih vrednosti regresijskih modelov za posamezne razrede

ne moremo obravnavati kot verjetnosti, da posamezen primer pripada tistemu razredu, saj lahko ležijo izven intervala [0,1] (Witten et al., 2011, str. 125). To rešuje sorodna statistična tehnika, imenovana **logistična regresija**, pri kateri zamenjamo odvisno spremenljivko, ki jo napovedujemo z linearnim modelom, tako da napovedujemo $\ln\frac{p(x)}{1-p(x)}$, pri čemer je *p(x)=P(Y=1|X=x)*, omenjeno transformacijo pa imenujemo **logit** preslikava (funkcija, transformacija) (prirejeno po Witten et al., 2011, str. 126). Takšna transformirana odvisna spremenljivka lahko zavzame vse vrednosti med -∞ in ∞, kar pomeni, da izhodne vrednosti linearnega modela izven intervala [0,1] ne predstavljajo več težave (Witten et al., 2011, str. 126). Končni model, ki nam da verjetnost, da primer pripada nekemu razredu, dobimo z inverzno preslikavo (prirejeno po Witten et al., 2011, str. 126), ki predstavlja logistično funkcijo sigmoidne oblike, in sicer

$$p(x) = P(Y = 1 \mid X = x) = \frac{1}{1 + e^{-\beta_0 - \sum X_j \beta_j}} \tag{1}$$

Koeficienti β_j, ki se pri linearni regresiji izračunajo tako, da se minimizira kvadratna napaka, se pri logistični regresiji dobijo z maksimiranjem log-verjetja (angl. *log-likelihood*), ki je za množico učnih primerov (x_i , y_i) podano kot

$$\sum_{i=1}^{n} (1 - y_i) \ln(1 - p(x_i)) + y_i \ln(p(x_i)) \tag{2}$$

pri čemer so y_i enaki 0 ali 1 (če je dejanska vrednost y_i enaka 0, želimo, da je verjetnost za razred 0, ki je podana kot $1-p(x_i)$, čim večja, ter obratno za primere, kjer je $y_i=1$) (prirejeno po Witten et al., 2011, str. 126).

Omeniti velja, da pri logistični regresiji primere klasificiramo v razreda tako, da jih ločimo s hiperravnino (angl. *hyperplane*) v prostoru primerov (če bi *X* sestavljali samo dve neodvisni spremenljivki, bi to bila premica); zapisano pomeni, da z logistično regresijo ne moremo popolnoma pravilno razločiti množice vhodnih primerov, ki jih ni možno ločiti po razredih z uporabo ene same hiperravnine v prostoru primerov (dokaz je v Witten et al., 2011, str. 126–127).

Razširitev pravkar zapisane razlage, ki obravnava dvorazredni klasifikacijski problem, na več razredov je možna po enakem postopku kot v primeru večodzivne linearne regresije, tako da izvedemo logistično regresijo za vsak razred posebej; ker se verjetnosti, ki jih tako dobimo za posamezne razrede, ne seštejejo v 1, jih je potrebno ustrezno združiti, za kar obstajajo učinkovite rešitve (Witten et al., 2011, str. 126).

2.2.2.2 Nevronske mreže

Umetne nevronske mreže (angl. *artificial neural networks – ANN*), ki jih običajno imenujemo kar **nevronske mreže** (angl. *neural networks*, v nadaljevanju NN), so navdih dobile v načinu, na katerega naj bi delovali človeški možgani; njihov osnovni gradnik je umetni **nevron** (angl. *neuron*), podobno kot je nevron osnovna celica v možganih (Guid & Strnad, 2015, str. 211–213). Umetni nevron ima množico vhodnih povezav, preko katerih prejme vhodno vrednost (signal), vhodno vrednost obdela ter rezultat pošlje na svoj izhod, pri čemer ima vsaka izmed vhodnih povezav pridruženo **utež** (angl. *weight*), ki določa moč povezave ter s katero se pomnožijo vhodne vrednosti (Guid & Strnad, 2015, str. 213–214). Množenje z utežmi predstavlja prvi korak obdelave vhodnih vrednosti, nadaljnja pa sta seštevanje pomnoženih vhodnih vrednosti ter preslikava rezultata z **aktivacijsko funkcijo** (angl. *activation function*), ki omeji amplitudo rezultata, tipično v interval [0,1] ali [-1,1]; najosnovnejša oblika aktivacijske funkcije je pragovna funkcija, ki ima izhod 1 (temu pravimo tudi, da se nevron sproži), če je njena vhodna vrednost večja ali enaka neki določeni vrednosti, ki jo imenujemo **prag** (angl. *threshold*), sicer pa je rezultat enak 0 (Guid & Strnad, 2015, str. 213–216). Omenjeni prag je zunanji parameter umetnega nevrona in si ga lahko zaradi bolj kompaktnega zapisa predstavljamo kot utež dodatne vhodne povezave s fiksno vrednostjo -1, rezultat pragovne funkcije pa je pri taki predstavitvi enak 1, če je vhodna vrednost večja ali enaka 0 (Guid & Strnad, 2015, str. 214–215). V praksi se namesto pragovne funkcije običajno uporablja **logistična funkcija** (angl. *logistic function*), na katero smo naleteli že pri opisu logistične regresije in je, za razliko od pragovne funkcije, odvedljiva; definirana je kot

$$\sigma(v) = \frac{1}{1 + e^{-av}} \tag{3}$$

pri tem pa je *a* parameter naklona (če pošljemo *a* proti neskončno, se logistična funkcija spremeni v pragovno) (Guid & Strnad, 2015, str. 216–217). Zaloga vrednosti pragovne in logistične funkcije je interval [0,1]; če želimo rezultat aktivacijske funkcije v intervalu [-1,1], potem lahko uporabimo funkciji signum ali hiperbolični tangens (tanh) (definiciji obeh funkcij za nadaljnjo razlago nista pomembni, zato ju ne bom navajal, bralec ju lahko najde v citirani literaturi) (Guid & Strnad, 2015, str. 215–217).

Če označimo z w_j utež *j*-te vhodne povezave nevrona, z x_j pripadajočo vhodno vrednost, z w_0 prag in postavimo $x_0 = -1$, lahko izhodno vrednost nevrona y ob uporabi logistične aktivacijske funkcije izračunamo kot

$$y = \sigma\left(\sum_{j=0}^{s} w_j x_j\right) \tag{4}$$

kjer je *s* število vhodnih povezav (Guid & Strnad, 2015, str. 214–216).

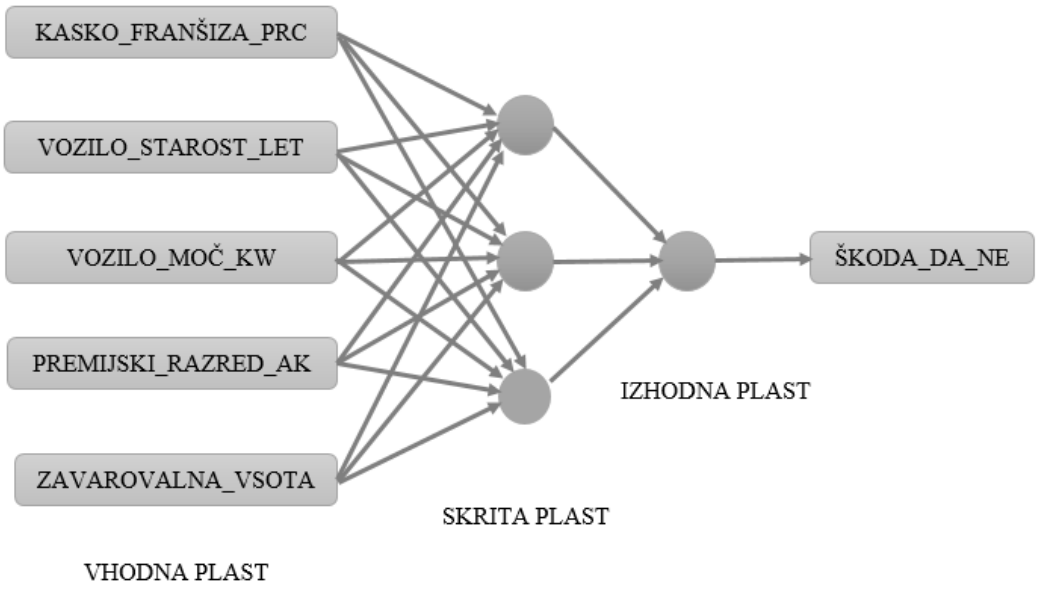
Nevroni preko vhodnih in izhodne povezave prejemajo in oddajajo signale (vrednosti), bodisi iz okolice oziroma v okolico bodisi od drugih oziroma drugim nevronom; pretok teh signalov oziroma arhitekturo (zgradbo) nevronske mreže si zato lahko predstavimo kot graf z usmerjenimi povezavami, v katerem so nevroni predstavljeni z vozlišči (Slika 1) (Guid & Strnad, 2015, str. 218–220). Poznamo različne mrežne arhitekture, tukaj pa bom predstavil samo **večplastno nevronsko mrežo s povezavami naprej** (angl. *mulitlayer feedforward neural network*), ki ji pravimo tudi **večplastni perceptron** (angl. *multilayer perceptron*, v nadaljevanju MLP) (Guid & Strnad, 2015, str. 221, 245), saj sem takšno nevronsko mrežo uporabil pri svojem delu, po mojem občutku pa je v praktični uporabi tudi najbolj razširjena. Vsebuje eno ali več **skritih plasti** (angl. *hidden layers*), katetrih nevrone imenujemo **skrite nevrone**, ker pri učenju nimamo neposrednega dostopa do njihovih izhodov (Guid & Strnad, 2015, str. 221; Gurney, 1997, str. 65); pri tem za rešitev problema običajno zadostuje ena sama skrita plast (Guid & Strnad, 2015, str. 264; Gurney, 1997, str. 75, 78). Ob tem ima takšna mreža še **vhodno plast** izvornih vozlišč (vrednosti neodvisnih spremenljivk oziroma atributov) ter **izhodno plast** nevronov, kjer dobimo rezultate preslikave f, ki jo predstavlja nevronska mreža. Nevronska mreža je polno povezana, če vsako vozlišče prejme izhodne signale vseh vozlišč prejšnje plasti (Gurney, 1997, str. 72), kar je veljalo tudi za nevronsko mrežo, ki sem jo uporabil v okviru tega magistrskega dela (obstajajo seveda tudi drugačne različice, t.i. delno povezane mreže, kjer nekatere povezave manjkajo (Guid & Strnad, 2015, str. 221)).

Učenje večplastnih perceptronov izvajamo z **algoritmom vzvratnega razširjanja** (angl. *back-propagation algorithm*, v nadaljevanju **back-propagation algoritem**), pri katerem na podlagi vhodnih vrednosti najprej izračunamo izhodno vrednost nevronske mreže, jo primerjamo z dejansko želeno izhodno vrednostjo ter izračunamo napako, nato pa ta signal napake pošljemo po mreži nazaj (od tod ime algoritma) ter prilagajamo vrednosti uteži tako, da postane odziv bližji želenemu (naučena informacija je pri nevronski mreži torej shranjena v utežeh) (Guid & Strnad, 2015, str. 250–251). Začetne vrednosti uteži postavimo naključno, postopek pa ponavljamo, dokler ni skupna napaka (MSE, glej podpoglavje 2.2.3.2) dovolj majhna (seveda lahko tudi omejimo število **epoh** (angl. *epoch*), pri čemer je epoha cikel, v katerem nevronski mreži predstavimo vse učne primer) (Guid & Strnad, 2015, str. 234, 254, 255).

Pomemben parameter učenja z back-propagation algoritmom je **hitrost učenja** (angl. *learning rate*), ki določa velikost prilagoditev, ki jih opravljamo na utežeh; z majhnimi vrednostmi je konvergenca k optimumu počasna, z velikimi pa se lahko zgodi, da optimum zgrešimo in sistem ne konvergira k stabilni rešitvi, ampak pride do oscilacije (Guid & Strnad, 2015, str. 225, 226, 255). To dilemo elegantno rešuje RPROP (angl. *resilient propagation*) algoritem (Riedmiller & Braun, 1993), ki je nadgradnja klasičnega back-propagation algoritma, njegova ideja pa je, da velikost sprememb uteži algoritem sproti prilagaja (podrobnosti delovanja presegajo okvir tega dela in se v njih na tem mestu ne bom spuščal). Prednosti RPROP algoritma sta hitrejša konvergenca k rešitvi ter neobčutljivost na začetne vrednosti njegovih parametrov, s katerimi se zato uporabnik ne rabi ukvarjati (Riedmiller & Braun, 1993, str. 588, 591). Učenje nevronske mreže, ki sem jo uporabljal v okviru moje raziskave, je bilo implementirano z RPROP

algoritmom, takšno nevronsko mrežo bom v nadaljevanju zaradi preprostosti imenoval kar **RPROP nevronska mreža**; pri tem je potrebno znova poudariti, da gre pri njej samo za izboljšavo algoritma, ki se uporablja za učenje, sama nevronska mreža pa je po svojih gradnikih in arhitekturi klasični MLP.

Slika 1: Primer nevronske mreže z eno skrito plastjo ter tremi skritimi nevroni v njej



Vir: Povzeto in prirejeno po N. Guid & D. Strnad, Umetna inteligenca (1. izd.), 2015, str. 222, slika 8.12; I. H. Witten et al., Data Mining: Practical Machine Learning Tools and Techniques (3rd ed.), 2011, str. 470, slika 11.28.

Pomemben problem pri uporabi nevronski mrež, kateremu je bilo v raziskavah posvečene že veliko pozornosti (Deepa & Sheela, 2013), je, koliko nevronov uporabiti v skriti plasti. Preveliko število skritih nevronov lahko vodi do prekomernega prileganja (angl. *overfitting*), ki pomeni, da nevronska mreža slabo posplošuje znanje na neznane primere, premajhno število pa, da ne moremo dobro predstaviti kompleksnih funkcij (govorimo tudi o angl. *underfitting*-u) (Guid & Strnad, 2015, str. 264; Xu & Chen, 2008, str. 683). Obstajajo razna pravila »preko palca« (angl. *rule of thumb*), na primer velikost skrite plasti naj bo nekje med velikostjo vhodne in izhodne plasti, vendar pa takšna pravila večinoma niso uporabna, ker je optimalno število odvisno od mnogih dejavnikov (velikost učne množice, kompleksnost funkcije, ki jo želimo predstaviti, …) (Xu & Chen, 2008, str. 683). Xu in Chen (2008, str. 684) na matematični in eksperimentalni podlagi predlagata pristop, ki upošteva velikost učne množice in število atributov; če označimo z N število učnih primerov in z d dimenzijo problema (število atributov) ter velja, da je N/d večje od 30, potem naj bi bilo optimalno število nevronov skrite plasti (označimo ga z n) blizu vrednosti

$$n = \sqrt{\frac{N}{d \ln N}} \tag{5}$$

Tega priporočila sem se kot izhodiščne točke v svoji raziskavi držal tudi sam, saj je omenjeni pogoj držal v vseh primerih, na katere sem naletel. Pri tem sem upošteval pripombo avtorjev, da pristop temelji samo na vrednostih *N* in *d* in da bi bilo smotrno vzeti v obzir tudi ostale dejavnike (kompleksnost funkcije, ki jo želimo predstaviti, prisotnost šuma v podatkih) (Xu & Chen, 2008, str. 685), zato sem preizkusil različna števila nevronov v skriti plasti (pri tem naj že tukaj navedem ugotovitev, da se je predlagana izhodiščna vrednost večinoma izkazala za dobro izbiro).

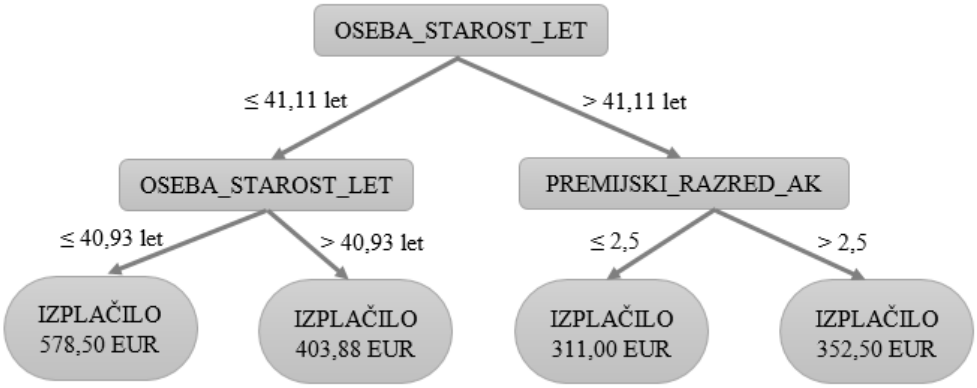
2.2.2.3 Odločitvena drevesa

Odločitvena drevesa (angl. *decision trees*) so sestavljena iz **vozlišč** (angl. *nodes*), v katerih se preveri vrednost določenega atributa (običajno se primerja z neko konstanto) in primer se na podlagi te primerjave preusmeri v ustrezno **vejo** (angl. *branch*), ki ga vodi do naslednjega vozlišča; to se ponavlja tako dolgo, dokler primer ne doseže **lista** (angl. *leaf*), ki predstavlja vozlišče, ki nima naslednikov, ampak je v njem podana izhodna vrednost, pri klasifikacijskih problemih je to razred (Witten et al., 2011, str. 64). Pri nominalnih atributih imamo iz vozlišča običajno posebno vejo za vsako izmed možnih vrednosti (v tem primeru ta atribut v kasnejših vozliščih ne bo več preverjan) ali pa vozlišče razdeli zalogo vrednosti nominalnega atributa v dve podmnožici (v tem primeru se lahko atribut v kasnejših vozliščih znova preveri) (Witten et al., 2011, str. 64). Pri numeričnih atributih preverjanje običajno pomeni primerjavo z določeno konstanto, v kateri se ugotovi, ali je vrednost atributa manjša od te konstante ali pa ne, kar nam da dve možni veji; imamo pa tudi druge možnosti, kot na primer posebna veja za manjkajoče vrednosti ali pa primerjava z določenimi intervali (Witten et al., 2011, str. 64). Numerični atribut je na svoji poti do lista pogosto preverjan večkrat, vsakič z uporabo drugačne konstante (Witten et al., 2011, str. 64). Poleg opisane preproste možnosti obstajajo še druge, kot na primer primerjava dveh atributov med seboj ali izračun vrednosti funkcije več atributov (Witten et al., 2011, str. 65). Kadar odločitveno drevo uporabljamo za reševanje regresijskega problema, je v listu numerična vrednost, ki predstavlja povprečno vrednost odvisne spremenljivke vseh učnih primerov, ki so prišli do tistega lista, takšno drevo pa imenujemo **regresijsko drevo** (angl. *regression tree*) (Witten et al., 2011, str. 67). To zasnovo je možno nadgraditi tako, da listi ne vsebujejo fiksne vrednosti napovedi, temveč se izhodna vrednost izračuna kot linearna kombinacija vrednosti določene podmnožice atributov, takšna drevesa se imenujejo **modelna drevesa** (angl. *model trees*) in lahko dajo boljše napovedi ob preprostejši obliki (manj vozlišč) v primerjavi z regresijskim drevesom (Witten et al., 2011, str. 67–68).

V nadaljevanju tega magistrskega dela bom tako za klasifikacijska odločitvena drevesa kot tudi za regresijska in modelna drevesa uporabljal skupno ime odločitvena drevesa (ali pa tudi samo drevesa, kadar bo iz konteksta jasno razvidno, da govorimo o odločitvenih drevesih), razen če

bo na določenem mestu obstajala potreba po razlikovanju, kar bom v tem primeru posebej navedel.

Slika 2: Ilustrativni primer (izmišljene vrednosti) odločitvenega (regresijskega) drevesa za napovedovanje višine odškodnine



Vir: Povzeto in prirejeno po I. H. Witten et al., Data Mining: Practical Machine Learning Tools and Techniques (3rd ed.), 2011, str. 68, slika 3.4(b).

Do sem sem nakazal nekaj različnih možnosti, ki jih imamo glede same oblike odločitvenega drevesa, ter opisal, kako ga uporabiti za izračun vrednosti odvisne spremenljivke na podlagi vrednosti atributov (neodvisnih spremenljivk). Vsa pozornost je torej veljala obliki in uporabi preslikave *f*, ki je rezultat strojnega učenja, torej opisu oblike, v kateri je predstavljeno znanje, ki je bilo pridobljeno na podlagi učnih primerov. Nič pa še ni bilo povedano o postopku (algoritmu), kako na podlagi učnih primerov zgraditi odločitveno drevo (preslikavo *f*). Tudi tukaj obstaja več možnosti, ki nam dajo različne rezultate. Podrobnejše opisovanje različnih algoritmov in možnosti, ki jih imamo pri gradnji odločitvenega drevesa, bi presegalo okvire tega magistrskega dela, tako da bom v nadaljevanju navedel samo nekaj glavnih idej in najpomembnejših možnosti.

Osnovni princip gradnje odločitvenega drevesa je preprost, in sicer poteka gradnja **rekurzivno** po principu **deli in vladaj** (angl. *divide and conquer*): izberemo atribut, ki ga bomo preverjali v prvem vozlišču, ki predstavlja **koren** (angl. *root*) drevesa, nato za vsakega od možnih rezultatov preverjanja (na primer vrednost atributa je enaka določeni nominalni vrednosti ali vrednost atributa je manjša od neke konstantne vrednosti) ustvarimo vejo in v njo »pošljemo« tiste učne primere, ki glede na izbrani pogoj sodijo vanjo, ter nad njimi rekurzivno ponovimo postopek, razen če je izpolnjen nek pogoj, ki ustavi nadaljnjo gradnjo (na primer vsi učni primeri so pripadniki istega razreda ali pa smo preverili že vse atribute) (Witten et al., 2011, str. 99–103). Pri tem pristopu je pomembno vprašanje, kako izbrati atribut, ki ga bomo uporabili za delitev v nekem vozlišču; cilj tukaj je, da čim prej ustvarimo vozlišča, ki bodo izpolnjevala ustavitveni pogoj in ne bo potrebna nadaljnja delitev (Witten et al., 2011, str. 100). Če govorimo o klasifikaciji, to pomeni, da želimo, da nam uporaba določenega atributa za delitev v nekem vozlišču kot naslednike tega vozlišča ustvari vozlišča, ki so čim bolj »čista« – najbolj čisto bi

bilo vozlišče, v katerem so vsi učni primeri predstavniki istega razreda in nadaljnja delitev ni potrebna (Witten et al., 2011, str. 100). Za merjenje uporabnosti atributa za delitev obstajajo različne metrike glede na to, ali opravljamo klasifikacijo (Witten et al., 2011, str. 100–107) ali regresijo (Palfy, 2009, str. 32–33). Opisan pristop, ki ga imenujemo tudi gradnja **od zgoraj navzdol** (angl. *top-down*), je skozi leta razvil in izpopolnil Quinlan, njegove raziskave pa so rezultirale v praktičnem in vplivnem sistemu za gradnjo odločitvenih dreves, ki se imenuje C4.5 (Witten et al., 2011, str. 107–108).

Zelo pomemben koncept pri gradnji odločitvenih dreves je tudi **obrezovanje** (angl. *pruning*), saj so odločitvena drevesa, zgrajena po opisanem postopku od zgoraj navzdol, običajno preveč prilagojena učnim primerom in imajo težave s posplošitvijo pridobljenega znanja na nove, neznane primere (Witten et al., 2011, str. 192). Ker bi natančno opisovanje različnih možnosti za obrezovanje presegalo okvir tega opisa, bom omenil samo glavni primer, in sicer postopek zamenjave poddrevesa (angl. *subtree replacement*), ki je predstavnik naknadnega obrezovanja (angl. *postpruning*) (nasprotje je sprotno obrezovanje (angl. *prepruning*), kjer se že med gradnjo odločimo, da prenehamo razvijati poddrevesa) (Witten et al., 2011, str. 195). Ideja zamenjave poddrevesa je, da se v nekem vozlišču odločimo za zamenjavo poddrevesa z listom; ta postopek izvajamo v smeri od listov proti korenu (Witten et al., 2011, str. 195–196). Takšna operacija bo seveda zmanjšala natančnost na naboru učnih primerov, kajti drevo smo praviloma gradili tako dolgo, dokler niso bila vsa vozlišča čista, vendar pa lahko z njo povečamo natančnost na testnih primerih (Witten et al., 2011, str. 195–196). Tukaj zato potrebujemo postopek, po katerem lahko ocenimo stopnjo napake, ki bi se pojavila na testni množici; ena možnost je, da si pri gradnji drevesa del učnih primerov zadržimo za to oceno, druga pa, da poskusimo stopnjo napake oceniti na podlagi učnih primerov, kar uporablja algoritem C4.5 (Witten et al., 2011, str. 197).

Med prednostmi odločitvenih dreves je potrebno omeniti, da je znanje izraženo v obliki zaporedja odločitev, ki so razumljive ljudem in opisujejo, kaj se je sistem naučil oziroma kaj je osnova za nove odločitve; izgrajeno odločitveno drevo lahko torej uporabimo za pridobitev novega znanja o problemu in ne samo za samo napovedovanje (Witten et al., 2011, str. 8–9). Naslednja prednost je tudi, da opravljajo izbiro atributov kot del postopka gradnje in so zato odporna na vključitev nepomembnih atributov, kar lahko zmede nekatere druge algoritme (Hastie et al., 2009, str. 352). Tako jih lahko izkoristimo tudi za izbiro pomembnih atributov (zmanjšanje dimenzije problema) za uporabo v drugih algoritmih, kot predlagajo Silipo, Adae, Hart in Berthold (2014, str. 12).

2.2.2.4 Ansambelsko učenje in naključni gozdovi

Pri **ansambelskem učenju** (angl. *ensemble learning*) gre za to, da kombiniramo različne modele, ki smo jih zgradili na podlagi učnih primerov; pri tem lahko za učenje posameznih modelov uporabimo različne podmnožice učnih primerov, ki smo jih dobili iz osnovnega nabora učnih primerov (Witten et al., 2011, str. 351). Rezultati so lahko zelo dobri, ena od slabosti pa je, da jih je težje razložiti in razumeti pot do njih, saj lahko ansambli vsebujejo veliko število modelov (Witten et al., 2011, str. 351–352). Za kombiniranje rezultatov posameznih modelov v

končni rezultat imamo več metod, glavne so bagging, boosting in stacking (Witten et al., 2011, str. 352). **Bagging** in **boosting** kombinirata modele istega tipa (na primer odločitvena drevesa), rezultat pa se dobi z glasovanjem ali povprečjem v primeru regresije; razlika je, da pri baggingu posamezne modele gradimo neodvisno, boosting pa je iterativen proces, pri katerem poskušamo v nekem koraku zgraditi model, ki bo pravilno napovedal tiste primere, pri katerih so bili modeli iz prejšnjih korakov neuspešni, končni rezultat pa se tukaj dobi z uteženim glasovanjem ali uteženim povprečjem (Witten et al., 2011, str. 358). Pomembna lastnost bagginga (okrajšava za angl. *bootstrap aggregating*) je, da učnih primerov, ki jih uporabi za gradnjo posameznih modelov, ne izbere v obliki neodvisnih podmnožic iz osnovnega nabora učnih primerov (v tem primeru bi nam zmanjkalo učnih primerov za gradnjo velikega števila modelov), ampak z naključnim vzorčenjem z vračanjem (angl. *bootstrap sampling*) ustvari novo množico učnih primerov enake velikosti (izbira takega pristopa pomeni, da se nekateri učni primeri iz osnovnega nabora morda sploh ne uporabijo, drugi pa se lahko uporabijo tudi po večkrat) (Witten et al., 2011, str. 354). **Stacking** se običajno uporablja za združevanje rezultatov različnih algoritmov strojnega učenja, pri čemer za to poskrbi poseben algoritem strojnega učenja, ki se poskuša naučiti, katerim izmed osnovnih modelov gre najbolj zaupati in kako jih čim bolje združiti (Witten et al., 2011, str. 369).

Naključni gozdovi (angl. *random forests*) so oblika ansambla odločitvenih dreves, ki jo je predlagal Breiman (2001). Pri tem se uporablja kombinacija bagginga ter naključne izbire atributov v posameznem vozlišču, zgrajena drevesa pa se ne obrezujejo (Breiman, 2001, str. 11). Končni rezultat se dobi z glasovanjem (Breiman, 2001, str. 6). Uporaba naključno izbrane podmnožice atributov (angl. *using random features*), imenovana tudi metoda naključnega podprostora (angl. *random subspaces method*), je namenjena vnosu raznolikosti v ansambel (zmanjšanju korelacije med drevesi); le-ta je potrebna, da ne dobimo samih enakih ali zelo podobnih dreves, vendar pa bagging sam za sebe vnese manjšo stopnjo raznolikosti, kajti porazdelitev naključno izbranih množic učnih primerov je podobna tisti izvorne množice (Breiman, 2001, str. 10; Witten et al., 2011, str. 356–357).

2.2.3 Načini za oceno uspešnosti modelov

Pri ocenjevanju uspešnosti izgrajenega modela nas zanima, kako dobro se bo obnesel na novih, še neznanih primerih in ne, kakšna je njegova uspešnost na primerih, katerih izhodno vrednost poznamo in smo jih uporabili za učenje (Witten et al., 2011, str. 148). Lahko bi preverili uspešnost izgrajenega modela na teh učnih primerih, vendar pa se izkaže, da to nikakor ni dobra ocena uspešnosti na neznanih primerih (Witten et al., 2011, str. 148). Vzemimo za primer odločitveno drevo, ki se nauči pravilno klasificirati vsak učni primer; takšno odločitveno drevo na učni množici ne bo naredilo nobene napake in bo pravilno klasificiralo vse primere, vendar pa bo dosegalo porazne rezultate na neznanih primerih, saj je napovedovanje premočno odvisno od podrobnosti učnih primerov, na podlagi katerih je drevo bilo zgrajeno, govorimo o **prekomernem prileganju** (angl. *overfitting*) učni množici (Kononenko, 2005, str. 76; Witten et al., 2011, str. 29, 88).

Zaradi zapisanega je pomembno, da uspešnost modela ocenimo na množici primerov, ki ni sodelovala v učenju (seveda moramo poznati dejanske izhodne vrednosti primerov iz te množice), imenujemo jo **testna množica** (angl. *test set*); množica primerov, ki se uporabi za učenje, pa je **učna množica** (angl. *training set*) (Witten et al., 2011, str. 149). Včasih uporabljamo tudi **validacijsko množico** (angl. *validation set*), ki je namenjena oceni uspešnosti v postopku optimizacije parametrov ali izbire določenega modela, testna množica pa se v tem primeru uporabi za končno oceno uspešnosti izbranega optimiziranega modela (Witten et al., 2011, str. 149). Ko smo ocenili uspešnost, lahko primere iz testne (in morebitne validacijske) množice priključimo primerom iz učne množice in na celotnem naboru podatkov zgradimo model za uporabo v praksi ter tako v največji možni meri izkoristimo podatke, ki so nam na voljo (Witten et al., 2011, str. 149).

Metoda, pri kateri del podatkov izločimo iz učne množice in uporabimo za testiranje (pogosto se uporabi dve tretjini podatkov za učenje ter eno tretjino za testiranje), se imenuje **zadržanje** (angl. *holdout*) in predstavlja težavo, če je količina podatkov, ki so nam na voljo, majhna – za izgradnjo dobrega modela potrebujemo čim več učnih podatkov, za dobro oceno uspešnosti pa čim več testnih podatkov (Witten et al., 2011, str. 150, 152). Dobro oceno uspešnosti bomo lahko dobili le, če sta obe množici tudi reprezentativni za naš problem, vendar pa se lahko zgodi, da imamo »smolo« in dobljena vzorca nista reprezentativna (Witten et al., 2011, str. 150, 152). Zamislimo si klasifikacijski problem, kjer se je po naključju zgodilo, da so vsi primeri nekega razreda pristali v testni množici – v tem primeru ne moremo pričakovati, da jih bo model pravilno razvrščal; tej težavi se lahko izognemo s takšno izbiro vzorcev, da so predstavniki vseh razredov v obeh množicah primerno zastopani (Witten et al., 2011, str. 152). Opisan pristop imenujemo **razslojevanje** (angl. *stratification*) ali **razslojeno vzorčenje** (angl. *stratified sampling*) (Witten et al., 2011, str. 152). Dodatno omilitev pristranskosti ocene uspešnosti, ki bi jo povzročila izbira testne množice, lahko dosežemo tako, da postopek večkrat ponovimo z različnimi naključnimi vzorci (angl. *repeated holdout*) (Witten et al., 2011, str. 152–153).

Pomembna statistična tehnika za spopadanje z omenjenimi težavami je uporaba **k-kratnega prečnega vrednotenja** (angl. *k-fold cross-validation*), pri kateri nabor primerov razdelimo na k približno enako velikih particij (angl. *fold*), najpogosteje z uporabo razslojevanja, od katerih nato eno uporabimo kot testno množico, preostalih k–1 pa za učenje; postopek ponovimo k krat, tako da je vsaka izmed particij po enkrat uporabljena kot testna množica (Witten et al., 2011, str. 153).

Standardni pristop je uporaba 10-kratnega prečnega vrednotenja, pri čemer število 10 temelji na številnih testih ter ima tudi nekaj teoretične podpore (Witten et al., 2011, str. 153). Vendar pa Witten et al. (2011, str. 153) ugotavljajo tudi, da ne gre za magično število in da bo 5-kratno ali 20-kratno prečno vrednotenje verjetno skoraj enako dobro.

V konkretnem primeru, s katerim sem se ukvarjal v svoji raziskavi, sem se posluževal tako delitve na učno in testno množico (zadržanja torej) kot tudi prečnega vrednotenja (v obeh primerih sem, kadar sem se ukvarjal s klasifikacijo, uporabil razslojeno vzorčenje), pri čemer sem se odločil za k=4. Uporabljeno metodo bom navedel sproti, v splošnem pa bi svojo odločitev

utemeljil s tem, da sem imel na voljo veliko količino podatkov. To je po eni strani pomenilo, da sta bili učna in testna množica veliki že pri enostavni delitvi (na primer 75 % proti 25 %), po drugi strani pa tudi, da bi bila uporaba 10-kratnega prečnega vrednotenja časovno preveč zahtevna. Nadalje sem za dokončno oceno uspešnosti uporabil posebno testno množico (ki predstavlja scenarij praktične uporabe), tako da sem zadržanje dela učnih podatkov oziroma prečno vrednotenje v bistvu uporabljal samo v vmesnih korakih kot validacijsko množico za optimizacijo parametrov in izbiro algoritmov. Seveda pri izbranem pristopu obstaja možnost, da sem imel v kakšnem primeru veliko »srečo« ali »smolo« pri naključni izbiri validacijske množice oziroma particij prečnega vrednotenja in je bila zato moja izbira modelov, ki sem jih uporabljal v nadaljevanju, suboptimalna.

2.2.3.1 Klasifikacijski problemi

Obstaja veliko število različnih mer, ki merijo uspešnost klasifikatorja (modela, ki rešuje klasifikacijski problem) (Kononenko, 2005, str. 75–86), ki imajo vsaka svoje posebnosti, zaradi katerih se mi zdi pomembno, da se uporabnik zaveda, kaj neka mera predstavlja. Prav tako niso vse mere primerne za vse probleme, pozoren je recimo treba biti pri reševanju problemov, kjer je napačna klasifikacija primerov iz nekaterih razredov hujša napaka (ima višjo »ceno«) kot napačna klasifikacija primerov iz drugih razredov (Kononenko, 2005, str. 77; Witten et al., 2011, str. 163–177). Manj kritično je na primer, če transakcijo, ki ni prevara, napačno klasificiramo kot prevaro in jo natančneje preverimo, kot pa če transakcijo, ki je prevara, klasificiramo kot veljavno in jo odobrimo.

V nadaljevanju se bom omejil predvsem na mere, ki se uporabljajo pri dvorazrednih klasifikacijskih problemih, saj je bil takšen tudi klasifikacijski problem, s katerim sem se srečal v raziskavi v okviru tega magistrskega dela; prav tako ne bom predstavil vseh možnih mer, ampak samo tiste, ki sem jih uporabil v tem magistrskem delu oziroma se mi zdijo pomembne za nadaljnje razumevanje.

Pri dvorazrednih problemih pogosto naletimo na primere, v katerih lahko enega od razredov smatramo kot »**pozitivnega**«, drugega pa kot »**negativnega**« (Kononenko, 2005, str. 82). Primer takšnega problema je recimo klasifikacija zavarovalnih pogodb glede na to, ali je na pogodbi škoda ali ne (oziroma ali bo na njej škoda v primeru napovedovanja na novih pogodbah); ta primer predstavlja tudi problem, s katerim sem se med drugim ukvarjal v okviru tega dela.

Za nadaljnjo razlago uvedimo najprej ustrezno notacijo in označimo (Kononenko, 2005, str. 82):

- s TP (angl. *true positives*) število pozitivnih primerov, ki so bili (pravilno) klasificirani kot pozitivni,
- s FP (angl. *false positives*) število negativnih primerov, ki so bili (napačno) klasificirani kot pozitivni,
- s TN (angl. *true negatives*) število negativnih primerov, ki so bili (pravilno) klasificirani kot negativni in

- s FN (angl. *false negatives*) število pozitivnih primerov, ki so bili (napačno) klasificirani kot negativni.

Definirajmo sedaj nekaj mer. **Klasifikacijska točnost** (angl. *classification accuracy, success rate*) je podana kot (Kononenko, 2005, str. 83 – tiskarski škrat v formuli!; Witten et al., 2011, str. 177)

$$T = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

in predstavlja delež pravilno klasificiranih primerov (Witten et al., 2011, str. 164) ter si jo lahko predstavljamo tudi kot verjetnost, da bo naključno izbran primer pravilno klasificiran (Kononenko, 2005, str. 75). Definiramo lahko tudi **stopnjo napake** (angl. *error rate*), ki je preprosto enaka 1 minus klasifikacijska točnost (Witten et al., 2011, str. 164).

Senzitivnost (angl. *sensitivity*), ki jo imenujemo tudi **priklic** (angl. *recall*), pomeni delež pravilno klasificiranih pozitivnih primerov (izmed vseh pozitivnih primerov) (angl. *true positive rate*, v nadaljevanju **TPR**) in je definirana kot (Kononenko, 2005, str. 82–85)

$$Senz = Priklic = \frac{TP}{TP + FN} = TPR \tag{7}$$

Specifičnost (angl. *specificity*) pomeni delež pravilno klasificiranih negativnih primerov (izmed vseh negativnih primerov) in je enaka 1 minus delež negativnih primerov, ki so bili napačno klasificirani kot pozitivni (angl. *false positive rate*, v nadaljevanju **FPR**), torej (Kononenko, 2005, str. 82–83)

$$Spec = \frac{TN}{TN + FP} = 1 - \frac{FP}{TN + FP} = 1 - FPR \tag{8}$$

Kot navaja Chawla (2005, str. 855), klasifikacijska točnost ni primerna mera, kadar so podatki neuravnoteženi (zelo majhno število pozitivnih primerov v primerjavi s številom negativnih) oziroma je napačna klasifikacija v primeru enega razreda hujša napaka kot obratno; zamislimo si problem, pri katerem imamo le 5 % pozitivnih primerov – na takšnem primeru bi klasifikator, ki bi preprosto vse primere klasificiral kot negativne, dosegel kar 95 % klasifikacijsko točnost (podobno ugotavljata tudi He in Garcia, 2009, str. 1264).

Težava je torej, da klasifikacijska točnost »zlije« in z enako težo upošteva pravilne klasifikacije tako pozitivnega kot negativnega razreda, kar pa, kot smo videli, v praksi večkrat ni zaželeno. Senzitivnost in specifičnost tega problema nimata, saj merita uspešnost na vsakem izmed razredov posebej. Senzitivnost 90 % pomeni, da bomo uspešno našli 90 % pozitivnih primerov, 10 % pa jih bomo napačno opredelili kot negativne; specifičnost 75 % bi pomenila, da bomo 25 % negativnih primerov napačno opredelili kot pozitivne (Kononenko, 2005, str. 83). Težava

tukaj je, da visoka vrednost ene izmed mer še ničesar ne pove o uspešnosti modela, kajti model, ki preprosto vse primere klasificira kot pozitivne, bo dosegel 100 % senzitivnost, vendar pa bo njegova specifičnost enaka 0 % (in obratno, če vse primere klasificiramo kot negativne) (Kononenko, 2005, str. 83). Potrebujemo torej nek kompromis, ki ga Chawla (2005, str. 855) opiše z besedami, da zahtevamo dokaj visoko stopnjo pravilnih klasifikacij v pozitivnem razredu ter za doseganje tega cilja dovolimo neko manjšo napako v negativnem razredu, Kononenko (2005, str. 83) pa isto opisuje kot trud, da maksimiziramo senzitivnost ob upoštevanju spodnje sprejemljive meje specifičnosti. Slednjo seveda tudi potrebujemo, kajti če bi veliko večino negativnih primerov klasificirali kot pozitivne in jih dalje obdelovali (pregledali potencialnega bolnika, preverili sumljivo transakcijo), bi to prav tako povzročilo stroške, uporaba klasifikatorja pa ne bi prinesla kakšne večje koristi.

Zanima nas torej razmerje med senzitivnostjo in specifičnostjo oziroma razmerje med pravilno klasificiranimi pozitivnimi primeri (TPR) in napačno klasificiranimi pozitivnimi primeri (FPR); analizo tega razmerja nam omogoča **ROC krivulja** (angl. *Receiver Operating Characteristic curve*) (primer ROC krivulje prikazuje Slika 5) (Kononenko, 2005, str. 83–84). V idealnem primeru imamo TPR 100 % (vsi pozitivni primeri pravilno klasificirani) in FPR 0 % (noben dejansko negativni primer se ni prikradel med pozitivno klasificirane) (Chawla, 2005, str. 855).

Na x osi grafa ROC krivulje (Slika 5) je prikazana vrednost FPR (ali 1 minus specifičnost), na y osi pa TPR; nek klasifikator tako vrne eno točko na grafu, razen v primeru, če nam ne vrne samo klasifikacije, ampak tudi verjetnost za klasifikacijo v nek razred (kot smo videli na primeru logistične regresije, podpoglavje 2.2.2.1); v tem primeru lahko s spreminjanjem praga odločitve (običajni primer bi bil, da primer klasificiramo v tisti razred, ki ima verjetnost večjo od 50 %, če govorimo o dvorazrednem problemu) spreminjamo razmerje med senzitivnostjo in specifičnostjo ter izrišemo ROC krivuljo (Kononenko, 2005, str. 84). Točka v levem spodnjem kotu grafa pomeni, da vse primere klasificiramo kot negativne (torej TPR in FPR oba enaka 0 %), točka v desnem zgornjem kotu pa, da vse primere klasificiramo kot pozitivne (TPR in FPR oba enaka 100 %); diagonala predstavlja klasifikator, ki deluje naključno, idealni klasifikator (TPR 100 % in FPR 0 % – kot smo že povedali) pa predstavlja točko v levem zgornjem kotu oziroma ROC krivulja bi potekala iz točke (0,0) v točko (0,1) in potem v točko (1,1) (Kononenko, 2005, str. 84). Krivulje uporabnih klasifikatorjev se torej nahajajo nad diagonalo ter po možnosti čim bližje levemu zgornjemu kotu grafa (Kononenko, 2005, str. 84).

Druga možnost za predstavo oziroma konstrukcijo ROC krivulje je, da vse primere uredimo padajoče glede na napovedano verjetnost za pozitivni razred (ali pa naraščajoče glede na verjetnost za negativni razred), se nato pomikamo po tako urejeni tabeli navzdol (od večje proti manjši verjetnosti za pozitivni razred) in v primeru, da je primer dejansko pozitiven, naredimo na grafu korak navzgor, sicer pa naredimo korak v desno (Witten et al., 2011, str. 169–172). Takšen pristop ustreza spreminjanju praga – če bi na neki točki v urejeni tabeli potegnili črto (tam postavili prag), bi dobili vrednost TPR na osnovi števila (dejansko) pozitivnih primerov nad to črto, vrednost FPR pa na osnovi števila (dejansko) negativnih primerov nad to črto (Witten et al., 2011, str. 172). »Nižje« (pri manjši verjetnosti v urejenem seznamu) kot bomo potegnili

to črto, več pozitivnih primerov bo nad njo in večjo vrednost senzitivnosti (TPR) bomo imeli ter višje bomo na grafu, vendar pa bomo s tem običajno zajeli tudi nekaj dodatnih negativnih primerov (ki bi jih pri takem pragu napačno klasificirali kot pozitivne – FPR) in bomo s tem na grafu bolj desno. Ta razlaga torej direktno predstavlja že omenjen kompromis med senzitivnostjo in specifičnostjo. Iz zapisane razlage pa je razvidno še nekaj – za ROC krivuljo torej niso pomembne dejanske vrednosti verjetnosti, ampak samo vrstni red primerov glede na verjetnost (Witten et al., 2011, str. 169). Upoštevaje to ugotovitev lahko ROC krivuljo torej narišemo za nek določen vrstni red primerov (ki bi naj ponazarjal njihovo pomembnost), same numerične vrednosti pa ne rabijo biti nujno prave verjetnosti iz intervala [0,1].

Če želimo ROC krivulje kvantitativno primerjati med seboj, si lahko pomagamo s **ploščino pod ROC krivuljo** (angl. *area under (the ROC) curve*, v nadaljevanju **AUC**); večja vrednost AUC v grobem pomeni boljši klasifikator (idealni klasifikator, ki poteka skozi točko (0,1), bi imel vrednost AUC enako 1 oziroma 100 %) (Kononenko, 2005, str. 84–85; Witten et al., 2011, str. 177). Vrednost AUC ima tudi lepo razlago, in sicer si jo lahko predstavljamo kot verjetnost, da bo klasifikator naključno izbran pozitivni primer uvrstil višje (mu dodelil višjo verjetnost za pozitivni razred) kot pa naključno izbran negativni primer (Kononenko, 2005, str. 85; Witten et al., 2011, str. 177). Ta razlaga se nadalje lepo navezuje na prej zapisano ugotovitev, da je za ROC krivuljo pomemben vrstni red primerov in ne numerična vrednost verjetnosti.

V konkretnem klasifikacijskem problem, s katerim sem se srečal v okviru tega magistrskega dela, sem za oceno uspešnosti klasifikatorjev uporabljal predstavitev z ROC krivuljo ter vrednost AUC. Imel sem namreč problem, pri katerem so bili pozitivni primeri (zavarovalne pogodbe s škodami) bistveno manj pogosti od negativnih. Nadalje me tudi ni toliko zanimala sama klasifikacija, ampak sem za nadaljnje delo potreboval verjetnost – kot ugotavljajo Apte et al. (1998, str. 3), moramo pri napovedovanju nastopa škodnih primerov na zavarovalnih pogodbah namreč upoštevati potencial (verjetnost, frekvenco) za nastop in ne fiksne klasifikacije v pozitivni ali negativni razred. Zaradi tega se mi je zdela ROC krivulja, ki upošteva verjetnosti, primerna za uporabo v mojem primeru. V zvezi z uporabo vrednosti AUC za numerično predstavitev uspešnosti klasifikatorjev naj omenim še pomembno podrobnost, da jo bom v nadaljevanju tega dela podajal kot pravo verjetnost iz intervala [0,1] in ne v alternativni obliki, ki bi jo predstavljalo navajanje v odstotkih.

2.2.3.2 Regresijski problemi

To podpoglavje je posvečeno predstavitvi mer, ki jih lahko uporabimo za ovrednotenje rezultatov oziroma primerjavo modelov, kadar imamo opravka z regresijskimi problemi, s katerimi sem se tudi srečal v okviru tega magistrskega dela (napovedovanje višine odškodnine ob nastopu škodnega primera, napovedovanje pričakovane vrednosti izplačil, napovedovanje višine komercialnega popusta na zavarovalni pogodbi). Omejil se bom na meri, ki sem ju v okviru magistrskega dela tudi dejansko uporabil.

Glavna mera, ki sem jo uporabljal in je tudi sicer osnovna ter najbolj pogosto uporabljana metrika (Dal Pozzolo, 2010, str. 30; Palfy, 2009, str. 19), je bila **povprečna kvadratna napaka** (angl. *mean squared error*, v nadaljevanju **MSE**), ki je definirana kot

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2 \tag{9}$$

pri čemer so p_i napovedane (angl. *predicted*) in r_i dejanske vrednosti, n pa je število primerov. Mero MSE so za problem ocenjevanja nevarnostne premije kot primerno ocenili tudi Dugas et al. (2003) ter jo uporabili za oceno uspešnosti v svoji raziskavi.

Rezultate sem primerjal tudi na podlagi **povprečne absolutne napake** (angl. *mean absolute error*, v nadaljevanju **MAE**), ki je definirana kot (Palfy, 2009, str. 19)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i| \tag{10}$$

Njeni lastnosti sta, da je manj občutljiva na posamezne ekstremne vrednosti (osamelce) kot MSE, saj napak ne kvadrira (Palfy, 2009, str. 19), in da so (iz istega razloga) rezultati v enakem velikostnem razredu kot vrednosti, med katerimi računamo napako.

Ker gre pri regresiji za napovedovanje numeričnih vrednosti (na primer višine izplačila) in ne za razrede in verjetnosti, je prav, da upoštevamo tudi **enoto**, v kateri so te vrednosti podane. Ta enota se namreč odraža tudi na merah MSE in MAE. Če so vrednosti, ki jih primerjamo (napovedane in dejanske), izražene v evrih (v nadaljevanju EUR), potem je MSE podana v EUR^2, MAE pa v EUR. Tak primer je pri meni nastopil pri napovedovanju (pričakovane) višine odškodnine (oziroma nevarnostne premije), kar obravnavam v podpoglavjih 4.2.2 in 4.2.3. Drug primer v zvezi z enotami pri merah MSE in MAE pa sem imel pri napovedovanju višine komercialnega popusta v odstotkih (podpoglavje 4.3); ker ta količina predstavlja delež in nima enote, so tudi vse vrednosti MSE in MAE v tem primeru brez enote.

2.2.4 Obravnava neuravnoteženih problemov

Načeloma lahko vsak nabor podatkov, ki ima različne porazdelitve posameznih razredov, smatramo za **neuravnoteženega** (angl. *imbalanced*), vendar pa to oznako običajno uporabljamo za nabore podatkov, pri katerih gre za občutna neravnotežja med posameznimi razredi, na primer 1 : 100 ali 1 : 1.000 in več (He & Garcia, 2009, str. 1264). Oznaka ni omejena samo na dvorazredne probleme (He & Garcia, 2009, str. 1264), vendar pa se bom v nadaljevanju razlage posvetil samo tem, saj je bil tak tudi primer, s katerim sem se srečal v okviru svoje raziskave. Pri neuravnoteženem dvorazrednem problemu ima eden izmed razredov, imenujemo ga **večinski** (angl. *majority*), torej bistveno več predstavnikov v naboru podatkov kot drugi, ki ga imenujemo **manjšinski** (angl. *minority*). V mojem konkretnem primeru so večinski razred predstavljale

zavarovalne pogodbe brez škodnih primerov (negativni primeri), manjšinskega pa zavarovalne pogodbe, na katerih je prišlo do škodnega primera (pozitivni primeri). Na podlagi znanja o primeru bi ocenil tudi, da je vzrok neuravnoteženosti v mojem primeru ležal v sami naravi problema (angl. *intrinsic*) in ne v zunanjih vzrokih (angl. *extrinsic*) (He & Garcia, 2009, str. 1264).

Problem neuravnoteženosti, ki se v praktičnih primerih večkrat pojavlja, je, da lahko zelo oteži učenje s standardnimi algoritmi strojnega učenja, saj so povezave, ki sledijo iz manjšinskega razreda, pogosto šibkejše in bolj redke, ker je primerov tega razreda manj; standardni algoritmi večinoma težijo k čim večji klasifikacijski točnosti, pri čemer predpostavljajo, da imajo vse klasifikacijske napake enako težo, kar v praktičnih primerih pogosto ni res (glej tudi podpoglavje 2.2.3) (Ganganwar, 2012, str. 42; He & Garcia, 2009, str. 1266). Za rešitev tega problema so bile predlagane številne rešitve, tako na ravni podatkov kot na ravni algoritmov (Ganganwar, 2012, str. 42). Primer rešitve na ravni algoritma je cenovno občutljivo učenje (angl. *cost-sensitive learning*), ki upošteva različno težo oziroma »ceno« napačne klasifikacije za posamezne razrede (Ganganwar, 2012, str. 44). Rešitve na ravni podatkov uporabljajo različne oblike **vzorčenja** za zmanjšanje neuravnoteženosti med razredi (Ganganwar, 2012, str. 43). V nadaljevanju se bom posvetil opisu metod, ki sem jih vzel v obzir oziroma uporabil v okviru tega magistrskega dela. Vse sodijo v skupino rešitev z uporabo vzorčenja in jih lahko dalje razdelimo v dve skupini (Ganganwar, 2012, str. 43):

- **nadvzorčenje** (angl. *oversampling*), pri katerem povečamo velikost manjšinskega razreda z replikacijo obstoječih primerov v manjšinskem razredu, in
- **podvzorčenje** (angl. *undersampling*), pri katerem za učenje uporabimo podmnožico večinskega razreda (nekatere primere iz večinskega razreda odstranimo).

Prednost teh metod je v tem, da delujejo kot predobdelava podatkov pred samim strojnim učenjem in jih zato lahko uporabimo v kombinaciji s katerimkoli algoritmom strojnega učenja (Ganganwar, 2012, str. 43). Slabosti nadvzorčenja sta, da je zaradi replikacije obstoječih primerov bolj verjetno, da bo prišlo do prekomernega prileganja (lahko se zgodi, da algoritem strojnega učenja ustvari pravilo, ki pokriva en sam repliciran primer), ter da umetno povečanje nabora učnih primerov poveča časovno zahtevnost strojnega učenja (Chawla, Bowyer, Hall, & Kegelmeyer, 2002, str. 328; Ganganwar, 2012, str. 43–44). Ravno obraten vpliv na časovno zahtevnost ima podvzorčenje, saj z odstranjevanjem primerov večinskega razreda zmanjša število učnih primerov; slabost podvzorčenja pa je, da zavržemo potencialno uporabne podatke (Ganganwar, 2012, str. 43). Osnovni izvedbi teh metod delujeta naključno, nadvzorčenje torej naključno z vračanjem izbira primere iz manjšinskega razreda in jih dodaja, podvzorčenje pa odstrani naključno izbrane primere večinskega razreda (Ganganwar, 2012, str. 43). Za omilitev omenjenih slabosti obeh metod obstajajo nadgradnje, pri podvzorčenju recimo metoda, ki odstranjuje takšne primere večinskega razreda, ki se zdijo redundantni (Ganganwar, 2012, str. 43). Nadgradnja nadvzorčenja pa je algoritem **SMOTE** (angl. *Synthetic Minority Over-sampling Technique*) (Chawla et al., 2002). Algoritem SMOTE ne replicira obstoječih primerov manjšinskega razreda, ampak iz njih ustvari nove »sintetične« primere v prostoru med

obstoječimi primeri (Chawla et al., 2002, str. 328–330; Ganganwar, 2012, str. 43). Ideja je, da področja odločitve v prostoru primerov manjšinskega razreda ne postanejo preveč specifična za posamezne primere (se skrčijo), kot se zgodi v primeru naključnega nadvzorčenja z vračanjem, ampak da se razširijo (Chawla et al., 2002, str. 352). Chawla et al. (2002, str. 331) pri tem predlagajo kombinacijo podvzorčenja in nadvzorčenja z uporabo algoritma SMOTE; hkrati povzemajo tudi rezultate drugih raziskav, ki so pokazali, da uporaba naključnega nadvzorčenja z vračanjem nima posebnega pozitivnega učinka (samostojna uporaba te metode ni bistveno izboljšala rezultatov strojnega učenja, uporaba v kombinaciji s podvzorčenjem pa ni dala boljših rezultatov kot uporaba samo podvzorčenja) (Chawla et al., 2002, str. 326).

2.2.5 Izbira atributov

Pri uporabi podatkovnega rudarjenja v praksi imamo pogosto na razpolago bogat nabor atributov, vendar pa uporaba velikega števila atributov pri strojnem učenju ni nujno dobra izbira, saj ima lahko za posledico slabše posploševanje znanja na nove primere kot pa uporaba primerne manjše podmnožice atributov (Caruana & Freitag, 1994, str. 1). **Izbira atributov** (angl. *attribute selection*) pomeni iskanje omenjene primerne podmnožice atributov. Tudi Silipo et al. (2014, str. 3) navajajo, da lahko **zmanjšanje dimenzije problema** (angl. *dimensionality redcution*), ki ga predstavlja odstranitev nekaterih atributov, poveča uspešnost algoritmov strojnega učenja ter hkrati tudi pospeši izvajanje le-teh, ter v ta namen predlagajo uporabo različnih tehnik. Dimenzijo seveda lahko zmanjšujemo samo na tak način in do te mere, da s tem ne poslabšamo natančnosti napovedovanja (Silipo et al., 2014, str. 6). Prva predlagana tehnika je izločitev atributov s preveč **manjkajočimi vrednostmi** (angl. *missing values*), saj atributi, katerih vrednost poznamo samo za majhni delež primerov, najbrž ne bodo uporabni za napovedovanje v večini primerov (Silipo et al., 2014, str. 6). Nadalje predlagajo tudi izločitev atributov z **nizko razpršenostjo** (varianco) (angl. *low variance*), pri tem razpršenost jemljejo kot merilo, koliko informacije vsebuje nek atribut (Silipo et al., 2014, str. 8). Tretja predlagana metoda zajema iskanje parov atributov z **veliko medsebojno odvisnostjo** (visoko korelacijo) (angl. *high correlation*) ter izločitev enega izmed atributov v takšnem paru; visoka korelacija namreč pomeni, da atributa nosita podobno informacijo in odstranitev enega izmed njiju ne bo pomenila občutnega zmanjšanja informacije v naboru podatkov (Silipo et al., 2014, str. 9). Naslednji predlagani metodi sta **metoda glavnih komponent** (angl. *Principal Component Analysis*, v nadaljevanju **PCA**) in izbira atributov z uporabo ansambla odločitvenih dreves oziroma **naključnega gozda** (Silipo et al., 2014, str. 11–12). Metode PCA pri svojem delu nisem uporabil, zato je ne bom opisoval, uporabo naključnega gozda za izbiro atributov pa skozi opis uporabe v praksi predstavljam v podpoglavju 4.2.2.2. Zadnji izmed predlaganih tehnik sta metodi **odstranjevanja atributov** (angl. *backward feature elimination*) in **dodajanja atributov** (angl. *forward feature construction*), ki korak po korak iščeta najprimernejši atribut za odstranitev ali dodajanje, in sicer uporabita tistega, katerega odstranitev ali dodajanje nam da najboljšo natančnost napovedi na novem naboru atributov (pri odstranjevanju začnemo z vsemi atributi, pri dodajanju pa s prazno množico atributov) (Silipo et al., 2014, str. 14–17). Ti metodi, za razliko od ostalih metod, delujeta v povezavi s konkretnim algoritmom strojnega učenja, ki se uporabi za učenje na trenutnem naboru atributov in oceno natančnosti modela na testni

množici; seveda lahko izbran nabor atributov kasneje po želji uporabimo tudi v povezavi z drugimi algoritmi strojnega učenja (Silipo et al., 2014, str. 14–16). Slabost zadnjih dveh metod je velika časovna zahtevnost, zato Silipo et al. (2014, str. 19) predlagajo, da jih v primeru problemov z veliko atributi uporabimo zatem, ko smo dimenzijo problema že zmanjšali z uporabo drugih metod. Še eno slabost teh dveh metod, ki sicer lahko dasta dobre rezultate v smislu natančnosti napovedovanja (Silipo et al., 2014, str. 19), omenjata Caruana in Freitag (1994, str. 3), in sicer metodi delujeta samo v eni smeri in ne omogočata, da odstranjen atribut kasneje znova dodamo nazaj oziroma dodanega odvzamemo, čeprav bi se na novem naboru atributov to lahko izkazalo za koristno. Omeniti velja tudi, da vse opisane metode niso uporabne za vse vrste atributov; PCA in metoda nizke razpršenosti delujeta samo na numeričnih atributih, metoda visoke korelacije pa ni uporabna za pare atributov različnih vrst (en nominalni in en numerični atribut) (Silipo et al., 2014, str. 19).

V okviru tega magistrskega dela sem izmed navedenih tehnik uporabil koncepte vseh razen PCA in metode dodajanja atributov, vendar pa sem si izvedbo metod in njihove podrobnosti deloma prilagodil; opisal jih bom sproti na mestih, kjer sem jih uporabil.

3 PRIPRAVA PODATKOV ZA PODATKOVNO RUDARJENJE

Pred uporabo algoritmov strojnega učenja v procesu podatkovnega rudarjenja sem moral pridobiti podatke o relevantnih zavarovanjih ter jih shraniti v primerni obliki. Ta del raziskave se je izkazal za časovno zelo zahtevnega, zanj sem v raziskavi porabil največ časa, kar je skladno z navedbami Fernandeza (2003, str. 15), ki ugotavlja, da se približno 70 % časa pri podatkovnem rudarjenju porabi za pripravo podatkov. Tukaj ne mislim samo tehnične izvedbe pridobivanja podatkov iz podatkovnih virov in shranjevanja v ustrezno obliko, ampak tudi na izbiro relevantne množice podatkov in ustrezne oblike za predstavitev le-teh; menim, da je omenjeno tesno povezavo z analizo in razumevanjem problema ter njegovih lastnosti, saj s pravilno pripravo podatkov pridobimo vpogled v vsebino in možnosti ter omejitve uporabe podatkov (Pyle, 1999, str. 3). Gledano s stališča CRISP-DM pristopa sem tukaj združil koraka razumevanja podatkov ter priprave podatkov, ki se pogosto prepletata in ju je torej smiselno obravnavati združeno (Nisbet, Elder, & Miner, 2009, str. 50).

Zaradi velikih prednosti, ki jih pridobimo s pravilno pripravljenimi podatki (Pyle, 1999, str. 3), sem tej fazi praktičnega dela raziskave namenil velik del časa ter jo skušal izvesti čim bolj kvalitetno. V tem poglavju bom predstavil korake, ki sem jih izvedel, težave, na katere sem pri tem naletel, in uporabljene rešitve zanje.

Podatki so v podatkovni bazi izbrane zavarovalnice shranjeni v relacijski podatkovni bazi. Podatki o posameznem zavarovanju, ki predstavlja en učni oziroma testni primer, se nahajajo v različnih tabelah. Zahteva za postopke strojnega učenja je bila, da so podatki podani v eni tabeli, v kateri je vsak učni oziroma testni primer predstavljen v eni vrstici, pri čemer stolpci predstavljajo posamezne atribute. Na poti do končnega nabora atributov in podatkov v zahtevani obliki sem izvedel naslednje korake:

1. izbor nabora podatkov,
2. identifikacija relevantnih tabel v podatkovni bazi,
3. identifikacija ter izpeljava atributov,
4. definicija tabele za vmesno shranjevanje podatkov,
5. priprava PL/SQL kode za pridobivanje vrednosti izbranih atributov,
6. kopiranje podatkov v tabelo za vmesno shranjevanje,
7. pregled in čiščenje pridobljenih podatkov ter atributov,
8. končni izbor podatkov ter shranjevanje v primerni obliki za strojno učenje.

V nadaljevanju podajam podrobnosti posameznih korakov; čeprav so le-te v nekaterih detajlih vezane na specifičen problem in nabor podatkov, menim, da je pregled potencialnih težav, možnih rešitev ter sprejetih odločitev lahko uporaben v sorodnih raziskavah in tudi širše; po drugi strani pa je v opisanih podrobnostih zajeta tudi večina znanja o problemu.

3.1 Izbor nabora podatkov

Predmet moje raziskave so bila zavarovanja AK, zato sem se seveda omejil samo na podatke za ta tip zavarovanj. Vendar pa je AK dosti preveč razvejan in kompleksen produkt, da bi ga bilo možno obravnavati kot zaključeno celoto, zato sem si moral postaviti dodatne omejitve ter se osredotočiti samo na določen del zavarovanj znotraj AK. V tem koraku sem tako sprejel odločitve, na podlagi katerih sem dobil osnovni nabor podatkov, na katerem sem nato izvajal nadaljnje korake. Omejitve, ki sem jih v tem koraku sprejel, so bile naslednje.

- Omejil sem se samo na **zasebni sektor**, izločil pa sem poslovni sektor. Pri vozilih, ki jih zavarujejo podjetja, praviloma ne poznamo voznika oziroma je voznikov več, zato tveganosti zavarovalne pogodbe ne bi mogli povezovati z lastnostmi voznika, kot sta na primer starost in spol. Seveda se zavedam, da tudi pri zasebnem sektorju zavarovanec ali uporabnik ni vedno voznik ter da lahko isto vozilo uporablja več oseb, na primer članov družine, vendar pa tega na podlagi podatkov v podatkovni bazi ni moč ugotoviti in posebej obravnavati. Po drugi strani pa me v raziskavi ni zanimala samo tveganost pogodbe, ampak tudi marketinški del. Glede na posebnosti pogodb poslovnega sektorja, kot so na primer generalne pogodbe, znotraj katerih se zavaruje več vozil, ter popusti na sklenitev pogodb za več let, sem sklepal, da pri trženju zavarovanj za poslovni sektor (kot navajata tudi Fric in Prijanovič (2010, str. 73)) veljajo drugačne značilnosti ter je bilo te pogodbe smiselno izločiti tudi s tega stališča.
- Omejil sem se samo na **polni kasko**. Polni kasko pri izbrani zavarovalnici obsega naslednje skupine zavarovanih nevarnosti: prometna nesreča, delovanje naravnih sil, požar in eksplozija ter poškodovanje vozila. Pod zavarovanja AK sicer sodi tudi zavarovanje dodatnih nevarnosti, ki jih lahko zavarovanec zavaruje v obliki kombinacij delnega AK (na primer škoda na parkirišču, tatvina vozila) (glej tudi podpoglavje 1.2). Pri tem veljajo za vsako izmed delnih kasko kombinacij posebne značilnosti, zaradi česar jih je po mojem mnenju potrebno obravnavati ločeno od polnega kaska ter tudi ločeno vsako posebej, zato sem jih izključil iz množice obravnavanih zavarovanj. Sem pa informacijo o delnih kasko

kombinacijah, ki jih je zavarovanec dodatno sklenil, upošteval v obliki atributov obravnavanih zavarovanj polnega kaska.

- Zavarovanje polnega kaska je v podatkovni bazi zapisano ločeno po posameznih skupinah zavarovanih nevarnosti, zato sem se odločil, da v tej fazi tudi sam podatke pridobim ločeno, vsako zavarovanje polnega kaska mi je tako praviloma dalo 4 ločene vrstice v pridobljenih podatkih. Obstajajo sicer posebni produkti, za katere omenjena ločitev ne drži, ker pa je bil njihov delež majhen, sem jih izločil iz obravnave.
- Omejil sem se samo na **osebna vozila**, ki so v premijskem ceniku izbrane zavarovalnice definirana kot motorna vozila za prevoz oseb, ki imajo poleg vozniškega sedeža še največ 8 sedežev.
- Omejil sem se samo na primere, ki so bili sklenjeni po sistemu cenikov, ki je bil v rabi od junija 2005, **najzgodnejši datum sklenitve**, ki se je med temi primeri pojavil, je 12.06.2005. Ceniki, ki so bili v uporabi pred tem, so drugače strukturirani in tako manj primerljivi, starejši podatki pa so tudi manj kvalitetni ter zaradi časovne oddaljenosti morda manj relevantni za napovedovanje.
- Zajetih primerov **časovno navzgor** v tej fazi eksplicitno nisem omejil, ampak sem to prepustil za kasnejše faze. Implicitno pa sem jih omejil s prehodom na najnovejši sistem cenikov AK, ki je v rabi od aprila 2015 ter je drugače strukturiran. Ocenil sem, da ta zavarovanje ne bodo prišla v poštev, saj se na datum poročanja (glej naslednjo točko) še niso iztekla.
- Podatke sem iz podatkovne baze pridobil na datum 29.02.2016. Ta datum bom v nadaljevanju imenoval **datum poročanja**. Zapisano ne pomeni, da sem vse podatke pridobil točno na omenjeni dan, ampak da sem pri kasnejših pridobivanjih upošteval stanje podatkov na dan 29.02.2016, ne pa tudi kasnejših sprememb.
- Izvzel sem **dodatne obračune** k policam. Eden izmed virov dodatnih obračunov so delna vračila premij ob predčasni prekinitvi zavarovanja, na primer zaradi prodaje vozila. Te razlike v premiji seveda nisem zanemaril, ampak sem jo upošteval tako, da sem letno premijo v primerih predčasne prekinitve pro-rata preračunal na dejansko trajanje zavarovanja. Drug vir dodatnih obračunov pa so dodatno zaračunani ali vrnjeni zneski zaradi napak, ki so nastale pri prvotni sklenitvi police, recimo zaradi napačno določenega premijskega razreda ali neupravičeno priznanega popusta. Žal sem pri pregledu primerov ugotovil, da te vrste dodatnih obračunov v podatkovni bazi niso shranjene na nek sistemiziran način, ki bi omogočal avtomatsko obdelavo (poleg spremembe premije bi moral ugotoviti in zajeti tudi spremembe parametrov zavarovalne pogodbe). Ker ročno popravljanje zaradi velike količine podatkov ni bilo izvedljivo, sem se odločil, da takšne popravke zanemarim.

3.2 Identifikacija tabel in atributov ter izpeljava atributov

Drugega in tretjega od navedenih korakov, ki sem jih navedel na začetku poglavja, sem izvedel povezano, in sicer sem se na podlagi znanja o problemu lahko že takoj odločil za določene atribute, ki so se mi zdeli pomembni, nato pa sem na podlagi poznavanja strukture podatkovne baze določil tabele, iz katerih sem lahko pridobil vrednosti teh atributov. Tej množici tabel sem

dodal še ostale, ki so bile kakorkoli povezane z zavarovanji AK, ter dobljeno množico tabel pregledal in med njihovimi stolpci iskal takšne, ki bi lahko hranili zanimive podatke za moj problem. V tej fazi sem namreč želel kar se da celovito zajeti vse atribute avtomobilskih premoženjskih zavarovanj, ki so na voljo v podatkovni bazi izbrane zavarovalnice, izbor relevantnih atributov pa opraviti šele kasneje na osnovi pridobljenih podatkov.

Določene atribute sem vseeno že takoj izločil, če sem ugotovil, da zanje sicer obstaja stolpec v tabeli v podatkovni bazi, vendar pa v veliki večini primerov v njem ni podatka, kar imenujemo **redkost** (angl. *sparsity*) podatkov (Pyle, 1999, str. 70). Konceptualno to ustreza metodi izbire atributov z izločanjem atributov z veliko manjkajočimi vrednostmi. Primer je bil recimo datum izdaje vozniškega dovoljenja zavarovanca. Ker ni bilo drugih primernih atributov z redkimi podatki, ki bi jih lahko z omenjenim združil v nov atribut, kot to za primer redkih podatkov predlaga Pyle (1999, str. 70), sem se odločil, da ga izločim. V tej fazi sem izločil tudi atribute, za katere sem ocenil, da za raziskavo niso relevantni, na primer priimek osebe.

Pri iskanju in definiranju atributov sem v grobem naletel na naslednje skupine primerov:

1. podatek v celici izvorne tabele predstavlja atribut in ga je mogoče uporabiti neposredno ali z zelo preprosto transformacijo, te atribute bom imenoval **enostavni atributi**,
2. podatek v celici izvorne tabele predstavlja atribut, vendar ga ni mogoče neposredno uporabiti, ampak je potrebna transformacija ali kombinacija več podatkov, te atribute bom imenoval **atributi s transformacijo**,
3. sam sem moral definirati atribut, ki smiselno predstavlja neko značilnost posameznega primera, pri čemer je vrednost takega atributa definirana kot transformacija kombinacije enega ali več podatkov, ki so za posamezen primer shranjeni v izvornih tabelah, te atribute bom imenoval **kompleksni atributi**,
4. na podlagi že izbranih atributov sem definiral novega, če sem ocenil, da bi izpeljana informacija lahko pomembno vplivala na rezultat in da bi predstavitev v izpeljanem atributu lahko pomagala strojnemu učenju, te atribute bom imenoval **izpeljani atributi**.

V nadaljevanju podajam podrobnejši opis posameznih skupin in primerov v njih.

3.2.1 Enostavni atributi

Predstavniki prve skupine so na primer dan sklenitve, datum rojstva in spol zavarovanca, poštna številka stalnega prebivališča zavarovanca, letnik vozila, območje registracije, zavarovalna vsota, moč ter prostornina motorja in podatki o vozilu, ki sem jih dobil iz kataloga vozil, kot na primer proizvajalec, navor motorja, število konjskih moči, tip motorja, izvedba karoserije, število vrat in podobno.

Atributov v tej skupini ni bilo težavno identificirati, so pa bile v določenih primerih potrebne preproste transformacije, na primer ugotavljanje dneva v tednu iz datuma sklenitve, ali pa je bilo podatek potrebno iskati v različnih tabelah. Na primer zavarovalno vsoto, za katero sem ocenil,

da bi lahko imela pomemben vpliv, sem iskal v več virih, od bolj zanesljivih proti manj zanesljivim; pri podatkih o vozilu iz kataloga vozil pa so bile določene lastnosti shranjene v obliki identifikatorja, ki ga je bilo potrebno preko druge tabele v relacijski podatkovni bazi (takšno tabelo imenujemo tudi šifrant) pretvoriti v človeku razumljivo vrednost.

Čeprav so tukaj omenjeni atributi, ki so vezani na zavarovanca, na pogled preprosti za pridobivanje (preberemo ustrezno celico v tabeli oseb), je na njih vseeno vezana podrobnost, ki jo velja omeniti. Na polici namreč ni zmeraj navedena samo ena **oseba** (zavarovanec), ampak je lahko nanjo vezanih več oseb z različnimi vlogami, na primer zavarovanec, sklenitelj oziroma zavarovalec, plačnik računa, uporabnik, lizingodajalec. Nabor atributov za podatkovno rudarjenje sem si zamislil tako, da vsakemu primeru ustreza natanko ena oseba, in sicer tista, za katero sklepam, da je najpogostejši voznik vozila. V večini primerov zasebnega sektorja je to zavarovanec, kar pa ne drži nujno v primeru lizingov; tukaj je v vlogi zavarovanca običajno lizingodajalec, ki je pravna oseba in ne more biti voznik, lizingojemalec, za katerega sklepam, da je dejanski voznik vozila, pa nastopa v vlogi uporabnika ali pa sklenitelja in plačnika računa. Podobni so primeri, ko sicer ne gre za lizing, vendar je lastnik vozila pravna oseba in jih prav tako najdemo v okviru zasebnega sektorja. Ker mi ni bila na voljo dokumentacija, v kateri bi bila zapisana natančna pravila glede vlog oseb na polici, sem vpogled v različne primere, ki se pojavljajo, pridobil z iskanjem po različnih kriterijih (na primer zavarovanec je pravna oseba in na polici obstaja fizična oseba). Na osnovi pridobljenega znanja sem nato definiral pravila, na podlagi katerih sem v postopku pridobivanja podatkov poskušal izmed oseb na polici določiti najpogostejšega voznika vozila. Glavni kriterij za to je bil, da mora to biti fizična oseba; podatek o tem, ali je neka oseba fizična ali pravna, je sicer na voljo v podatkovni bazi, vendar pa napačne vrednosti niso redke, zato sem uporabil še dodatne kriterije na podlagi vpisanih podatkov o rojstnem datumu in spolu (za katera sem pričakoval, da sta izpolnjena le v primeru fizičnih oseb, vendar se je izkazalo, da tudi to ne drži zmeraj). V primeru, da ustrezne fizične osebe na polici moj postopek ne najde, upošteva tudi pravno osebo, vendar to označi v posebnem atributu. Odločitev, kaj s takšnimi primeri narediti, sem na tem mestu preložil na kasneje.

Zaradi preprostosti bom v nadaljevanju za izbrano osebo s police, za katero sklepam, da je najpogostejši voznik vozila, ter sem jo določil po zgoraj opisanem postopku, uporabljal izraz **zavarovanec**. Če se bom na katerem mestu s to oznako želel omejiti na vlogo, kot je zapisana na zavarovalni pogodbi, bom to posebej zapisal. Ker sem bil mnenja, da je informacija o tem, ali gre pri izbrani osebi dejansko za zavarovanca z zavarovalne pogodbe ali pa nastopa v kateri drugi vlogi (na primer v vlogi uporabnika), morda lahko relevantna za napovedi, sem jo shranil v poseben atribut.

3.2.2 Atributi s transformacijo

V drugi skupini imamo primere, kjer je bila, zaradi načina shranjevanja ali zaradi slabe kvalitete podatkov, potrebna transformacija. Sem sodi recimo **trajanje zavarovanja**, s tem mislim število dni kritja, za katerega sem pričakoval, da bo imelo pomemben vpliv na tveganost zavarovalne pogodbe (daljši čas izpostavljenosti pomeni večjo možnost za nastanek škodnega primera).

Zavarovanje AK se po pogojih izbrane zavarovalnice sicer praviloma sklepa za eno leto, vendar pa obstajajo izjeme, kot na primer krajši čas zavarovanja zaradi uskladitve z datumom registracije. V podatkovni bazi sta na voljo podatka o prvem in zadnjem dnevu kritja in trajanje zavarovanja je v običajnem primeru preprosta razlika teh dveh datumov. Vendar pa se le-ta v posebnih primerih, kot je na primer predčasna prekinitev zavarovanja, ne posodobita, ampak se datum prekinitve v podatkovno bazo zapiše posebej. Upošteval pa sem tudi primer, ko zavarovanec novo polico sklene že po poteku police za preteklo leto, pri čemer se datum začetka nove police izenači z datumom poteka predhodne police, vendar pa so nevarnosti, ki jih krije zavarovanje AK, krite šele od dneva sklenitve naprej (posebna klavzula na zavarovalni polici). Ker to dejansko pomeni krajši čas izpostavljenosti, se mi je zdelo smiselno, da to upoštevam. Za določanje trajanja zavarovanja sem tako moral uporabiti kombinacijo štirih različnih podatkov ter pri tem preveriti njihovo skladnost oziroma smiselnost; za datum prekinitve zavarovanja se je na primer izkazalo, da je precej nezanesljiv ter da lahko pomeni tudi kaj drugega, na primer datum izstavitve police za naslednje leto, zato sem ga kot datum poteka kritja upošteval le, če je ležal med prvotnimi datumi kritja.

Naslednji atribut, ki bi ga uvrstil v skupino atributov s transformacijo, je podatek o soudeležbi zavarovanca v primeru škode, t.i. odbitni franšizi, ki je lahko podan v odstotkih od zavarovalne vsote ali v fiksnem znesku. Da bi bili ti primeri primerljivi, jih je bilo potrebno uskladiti; najprej sem moral ugotoviti, v kateri enoti je odbitna franšiza shranjena ter jo v kombinaciji z zavarovalno vsoto pretvoriti še v drugo obliko. Ker se mi je podatek o odbitni franšizi zdel pomemben, sem se odločil, da ga shranim v obeh oblikah, torej v dveh ločenih atributih, in odločitev o uporabi enega ali drugega sprejmem kasneje.

Večjo podskupino atributov s transformacijo predstavljajo primeri, pri katerih je bilo zaradi normalizacije relacijske podatkovne baze podatek potrebno poiskati po določenem ključu v isti ali drugi izvorni tabeli. Pomemben predstavnik so vsa dodatna zavarovanja, ki jih ima zraven polnega kaska zavarovanec še sklenjena za isto vozilo na isti ali celo drugi zavarovalni polici. To so na primer delne kasko kombinacije, zavarovanje asistence, zavarovanji AO in AO Plus, zavarovanje izgube bonusa ter nezgodno zavarovanje voznika in potnikov. Pri delnih kasko kombinacijah sem kot atribut zajel tudi podatek o morebitni odbitni franšizi na tisti delni kasko kombinaciji. V to podskupino bi uvrstil tudi atribute, ki povedo, ali gre v nekem primeru za leasing, za last pravne osebe ali za voznika z manj kot tremi leti vozniških izkušenj (t.i. mladi voznik). Pri tem je potrebno povedati, da se zadnja dva podatka v obliki doplačil pojavljata samo na zavarovanju AO, tako da ju je bilo potrebno iskati pri zavarovanju AO, kar je bilo možno, če je bila ustrezna AO polica najdena; v nasprotnem primeru sem s posebno vrednostjo atributa označil, da podatek ni znan.

3.2.3 Kompleksni atributi

3.2.3.1 Popusti in doplačila

Prva velika podskupina kompleksnih atributov so **popusti** in **doplačila**. Le-ti so v podatkovni bazi izbrane zavarovalnice shranjeni v dveh različnih tabelah, pomen posameznega zapisa v tabeli je določen s kodo popusta. Pri tem posamezen popust ali doplačilo praviloma predstavlja neko značilnost zavarovanca ali zavarovalne pogodbe, na podlagi katere je bil priznan popust ali obračunano doplačilo, in te značilnosti sem seveda želel zajeti v obliki atributov. Imamo pa tudi komercialne popuste, ki sem jih zajel v poseben atribut.

Definiranja atributov, ki so vezani na popuste in doplačila, sem se lotil tako, da sem najprej s poizvedbo iz podatkovne baze pridobil vse različne popuste in doplačila, ki se pojavljajo na naboru podatkov, ki sem ga določil v prvem koraku priprave podatkov. Poizvedba mi je vrnila 134 različnih popustov in doplačil. Pri tem je bilo v nekaterih primerih iz opisa takoj razvidno, kaj predstavlja, na primer popust za določen premijski razred ali pa doplačilo za taksi vozila. Pri nekaterih drugih primerih pa ni bilo tako (na primer »šalterski popust«), tukaj sem poskušal značilnosti in pravila takšnega popusta ali doplačila ugotoviti z zamudnim pregledom primerov zavarovanj, na katerih se je pojavil ali pojavilo, saj v večini teh primerov nisem uspel najti ustrezne dokumentacije, ki bi opisovala določen popust ali doplačilo.

V zvezi s popusti in doplačili sem ugotovil, da se mnogi primeri pojavljajo zelo redko. Na primer doplačilo za taksi vozila se pojavi v 0,23 % primerov, popust za opravljeno šolo varne vožnje pa v 0,04 % primerov osnovnega nabora podatkov. Za takšne primere se mi je zdelo nesmiselno, da jih zajamem v obliki posebnih atributov, torej da bi na primer definiral atribut, ki pove, ali gre v nekem primeru za taksi vozilo ali ne; po drugi strani pa takšne informacije nisem želel kar zavreči, saj predstavlja neko značilnost, ki bi lahko imela vpliv na opazovane spremenljivke. Težava tukaj ni v tem, da ne bi imeli podatka (manjkajoča vrednost v podatkovni bazi), ampak v tem, da je določenih primerov, na primer taksi vozil, v primerjavi s celotnim obsegom podatkov zelo malo. Kljub temu se mi je problem zdel soroden že omenjeni redkosti podatkov. V zvezi s tem Pyle (1999, str. 70) navaja, da se orodja za podatkovno rudarjenje v kombinaciji z visoko redkostjo podatkov slabo obnašajo; če po zelo redkih podatkih vseeno želimo rudariti, Pyle (1999, str. 70–71) predlaga, da jih združimo v manjše število atributov, tako da vsak izmed njih nosi informacijo iz več originalnih podatkov.

Sledil sem predlaganemu pristopu, tako da sem po moji oceni sorodne popuste in doplačila združil v skupine ter posamezno skupino predstavil v obliki atributa. Seveda takšni atributi več niso mogli biti binarni (popust ali doplačilo in s tem neka značilnost obstaja ali ne), ampak sem vrednost atributa definiral kot vsoto deležev popustov in doplačil, ki so prisotni pri posameznem primeru. Na tak način sem želel doseči, da bo vrednost atributa zvezno odražala zmanjšano (popust) ali povečano (doplačilo) tveganje neke zavarovalne pogodbe. Čisto vseh popustov in doplačil nisem združil v skupine, ampak sem za nekatere, ki se pojavljajo pogosteje in za katere sem ocenil, da bi jih bilo zanimivo obravnavati ločeno, dodal ločene atribute. V tej fazi sem za

popuste in doplačila tako definiral 6 atributov za skupine popustov in doplačil (način plačila, komercialni, paketni, po oceni strokovne službe in ločeno popusti ter doplačila glede na značilnosti, ki zmanjšujejo ali povečujejo tveganje) ter 6 atributov za posamezne popuste in doplačila (za premijski razred, za starost ter za visoko vrednost vozila, za brezškodno dogajanje v preteklem letu, za posebni popust ter za majhno število opravljenih kilometrov).

Na koncu sem dodal še poseben ločen atribut za odstotek popustov **po ceniku**. Sem sem štel vse popuste in doplačila, razen komercialnih ter popustov in doplačil za način plačila. Pomembno je poudariti, da odstotek v omenjenem atributu ni preprosta vsota ostalih kategorij, kajti različni popusti (in doplačila) na zavarovalno pogodbo se ne seštevajo, ampak se vsak naslednji obračuna na premijo, že znižano ali povišano za predhodne. Razlog, da sem se pri prej opisanem grupiranju popustov in doplačil v skupine odločil za seštevanje, je bil v tem, da sem želel, da je neka značilnost, ki je bila vzrok za popust ali doplačilo, v združenem atributu vedno predstavljena z enako težo, ne glede na prisotnost ostalih popustov in doplačil na zavarovalni pogodbi. Kot merilo skupnega učinka vseh teh popustov in doplačil (razen komercialnih in tistih za način plačila) pa sem se odločil dodati še omenjeni posebni atribut, recimo za primer, da zaradi zmanjšanja dimenzije problema ne bi uporabil vseh ostalih atributov za popuste in doplačila, kajti pri določenih skupinah sem tudi po združitvi opazil nizko razpršenost vrednosti, zaradi česar bi jih v sklopu izbire atributov utegnil odstraniti.

Pri izbrani zavarovalnici so zavarovanci pri sklepanju polnega kaska razvrščeni v 20 premijskih razredov; za vsak premijski razred je določena vrednost popusta (bonus) ali doplačila (malus) na izhodiščno premijo, zavarovanci pa so v te razrede razvrščeni oziroma med njimi prehajajo glede na preteklo škodno dogajanje (glej tudi podpoglavje 1.2). Zraven odstotka popusta ali doplačila glede na premijski razred sem dodatno definiral še atribut s številčno vrednostjo premijskega razreda, ki je na nek način natančnejši, saj lahko ima več premijskih razredov enako vrednost popusta ali doplačila. Podoben sistem velja tudi pri sklepanju zavarovanj AO, vendar pa se premijski razred AO določa ločeno od AK, zato sem za AO definiral ločena atributa. Seveda sem slednji podatek lahko pridobil le, če ima zavarovanec sklenjeno zavarovanje AO pri izbrani zavarovalnici ter sem omenjeno polico v podatkovni bazi uspel najti, saj neposredna povezava med njima ne obstaja (iskal sem police, ki se časovno prekrivajo z zavarovanjem AK ter sem zanje preko številke šasije ugotovil, da so vezane na isto vozilo).

3.2.3.2 Zgodovina premij in izplačil za zavarovanca

Druga večja podskupina kompleksnih atributov so bile informacije o zavarovančevi zgodovini v obliki **premij**, ki jih je plačal, ter **izplačil** za škodne primere, ki jih je prejel. Odločitev, da vključim te informacije, sem zasnoval na predpostavki, da bi na podlagi preteklih izplačil morda lahko odkrili bolj tvegane zavarovance. Vendar pa se mi sam podatek o izplačilih ni zdel zadosten, saj je pomembno tudi, pri kakšni izpostavljenosti (število in trajanje sklenjenih zavarovanj) je do njega prišlo (pri zavarovancih z več zavarovanji sem pričakoval več škod). Kot merilo za izpostavljenost sem dodal atribut za znesek plačanih premij, pri čemer sem upošteval fakturirano premijo, zmanjšano za DPZP, torej kosmato obračunano premijo. Pri tem se mi je

zdelo pomembno še, da ne upoštevam samo skupnih zneskov, ampak podam to informacijo ločeno po določenih skupinah zavarovanj, kajti povezava med odvisnimi spremenljivkami ter zavarovančevo zgodovino premij in izplačil je lahko različna za različne vrste zavarovanj. Skupine zavarovanj, po katerih sem pridobil podatke, sem določil empirično. Za več zavarovancev sem pridobil podatke ter poskušal identificirati skupine podobnih zavarovanj, pri katerih se pri fizičnih osebah pojavlja največ premije ter izplačil. Pri združevanju zavarovanj v skupine sem se oprijel delitve na zavarovalne vrste, ki jo navaja 7. člen ZZavar-1, ter se odločil za naslednje skupine:

- nezgodna zavarovanja (1. zavarovalna vrsta po 7. členu ZZavar-1),
- škodna zavarovanja (8. in 9. zavarovalna vrsta po 7. členu ZZavar-1),
- AK (3. zavarovalna vrsta po 7. členu ZZavar-1),
- AO (10. zavarovalna vrsta po 7. členu ZZavar-1),
- ostala zavarovanja (vsa zavarovanja, ki ne sodijo v 1., 3., 8., 9. ali 10. zavarovalno vrsto po 7. členu ZZavar-1, vendar največ do vključno 18. zavarovalne vrste – življenjska zavarovanja sem torej izključil),
- skupaj vse prejšnje skupine (1. do 18. zavarovalna vrsta po 7. členu ZZavar-1).

Omenjeni podatki seveda niso bili neposredno na voljo v kakšni tabeli v podatkovni bazi, ampak jih je bilo potrebno pridobiti z združitvijo več tabel. Pomembno se mi zdi poudariti, da sem želel podatke, kot bi jih pridobili na dan, ko je zavarovanec sklenil polico za AK zavarovanje, ki predstavlja učni oziroma testni primer, za katerega pridobivamo vrednosti atributov, torej sem smel upoštevati samo obračunano premijo ter izplačila do tega dneva. Paziti sem moral tudi na police, katerih kritje se na ta dan še ni izteklo. Pri tem sem se odločil upoštevati le tiste police, katerih kritje se je izteklo najkasneje en dan pred dnevom sklenitve AK police, za katero pridobivamo vrednosti atributov (druga možnost bi bila uporaba merodajne premije).

Seveda pa to, da se je kritje izteklo, še ne pomeni nujno, da je tudi škodno dogajanje na takšni polici zaključeno, saj še zmeraj lahko pride do izplačil in tudi prijav škodnih primerov. To sem lahko delno kompenziral s tem, da sem vsoti izplačil prištel še znesek rezervacije po popisu na ustrezni dan, kar pa je spet le približek, saj ne reši primerov, ko škodni primer, ki se je že zgodil, še sploh ni prijavljen. Ta približek bi lahko izboljšal, če bi dopustil nek časovni interval med iztekom kritja police, katere prispevek računamo, ter dnevom sklenitve AK police, vendar pa bi na tak način več polic izpadlo.

3.2.3.3 Premije za primerjavo in napovedovanje

Med kompleksne atribute bi uvrstil tudi različne premije na polici, razen osnovne premije brez popustov in doplačil, ki bi jo uvrstil med enostavne atribute. V atribute sem dodal dve različici kosmate premije, in sicer sem pri izračunu enkrat upošteval samo popuste in doplačila po ceniku, drugič pa tudi komercialne popuste. Atributa sta bila namenjena primerjavam rezultatov in ne za napovedovanje ali učenje. V zvezi z napovedovanjem tržne cene zavarovanja sem kot atribut shranil tudi odstotek komercialnih popustov na polici.

3.2.3.4 Podatki o škodah, ki se napovedujejo

Med atribute za napovedovanje sodijo tudi podatki o **škodnih primerih** na polici, ki sem jih vzel za merilo tveganosti zavarovanca. V osnovi sta to dva atributa, in sicer **število škodnih primerov** ter **znesek izplačil** za odškodnine do datuma poročanja. Pri tem sem upošteval samo škodne primere, ki sodijo na polni kasko, ki je bil predmet moje raziskave, pri čemer sem jih moral še dodatno razvrstiti v ustrezno skupino zavarovanih nevarnosti polnega kaska, ki predstavlja primer v naboru podatkov.

Posebno pozornost sem namenil **štetju škodnih primerov**, ki se je izkazalo za precej bolj zapleteno, kot če bi samo sešteli število škodnih spisov, ki so vezani na obravnavano polico. Za posamezen škodni primer sem lahko pridobil podatke o njegovem statusu, vsoti izplačil ter o rezervaciji po popisu na datum poročanja. Pri tem so lahko nastopili naslednji primeri:

1. škodni primer je zaključen, vsota izplačil je večja od 0, rezervacija po popisu je enaka 0,
2. škodni primer je zaključen, vsota izplačil ter rezervacija po popisu sta enaki 0,
3. škodni primer je še odprt, vsota izplačil ter rezervacija po popisu sta enaki 0,
4. škodni primer je še odprt, vsota izplačil je enaka 0, rezervacija po popisu je večja od 0,
5. škodni primer je še odprt, vsota izplačil je večja od 0.

Za primere pod prvo točko ni bilo dvomov in sem jih seveda štel. Za primere pod drugo točko je bilo potrebnega več premisleka; nek škodni dogodek se je namreč verjetno zgodil in na to bi lahko gledal tudi s stališča, da je ta zavarovanec bolj tvegan, neodvisno od tega, ali je zavarovalnica kasneje zavrnila izplačilo (na primer zaradi kršitve zavarovalnih pogojev) ali pa je zavarovanec morda umaknil škodni zahtevek (na primer zaradi visoke odbitne franšize ali ker ni želel nazadovanja pri uvrstitvi v premijske razrede). Vendar pa so okoliščine, ki lahko privedejo do takšnega zaključka, del zavarovalne pogodbe in, ker me je zanimala tveganost zavarovanca v smislu zavarovalne pogodbe, sem se odločil, da takšnih primerov ne štejem. S tem sem se izognil tudi škodnim spisom, ki so bili odprti pomotoma (na primer dva škodna spisa za isti škodni primer) ter jih tudi najdemo pod drugo točko. Za primere pod tretjo točko sem ugotovil, da so ob predpostavki, da opazujemo samo police, katerih kritje se je izteklo že dalj časa pred datumom poročanja, redki ter po vsebini sorodni primerom pod točko 2, zato jih tudi nisem štel.

Največ dvomov sem imel glede točke 4. Tukaj sem ugotovil, da gre ob predpostavki, ki sem jo navedel v zvezi s tretjo točko, večinoma za primere, kjer je prišlo do tožb, ker zavarovalnica, na primer na podlagi pogojev zavarovanja ali zaradi suma zavarovalniške goljufije, ne želi izplačati odškodnine in te tožbe še niso dokončno rešene. Takšnih primerov sicer ni bilo dosti, vendar pa so rezervirani zneski za potencialno odškodnino, če bi do izplačila prišlo, visoki (tudi 80.000 EUR in več). Zaradi potencialno velikega vpliva na rezultate (če bi štel škodni primer, bi bilo smiselno upoštevati tudi rezervacijo po popisu) in negotovih iztekov teh primerov, ki morda celo bolj težijo v smer, da do izplačila ne bo prišlo, sem se odločil, da tudi teh primerov ne štejem.

Primere pod točko 5 sem obravnaval neodvisno od zneska rezervacije ter sem jih štel, saj bi bilo težko upravičiti izpuščanje škodnih primerov, na katerih so se zgodila izplačila. Sem si pa zastavil vprašanje, ali naj vsoti izplačil prištejem tudi znesek rezervacije po popisu na datum poročanja, če je le-ta večji od 0. Pri tem sem v zvezi z rezervacijo po popisu ugotovil sorodnost s primeri iz točke 4. Iz tega razloga sem se odločil, da zneska rezervacije po popisu ne upoštevam, sem si ga pa za vsak slučaj shranil kot dodaten atribut, ki bi ga lahko uporabil, če bi želel spremeniti to odločitev.

Kot rezultat zgornje analize sem prišel do zaključka, da štejem tiste škodne primere, pri katerih je vsota izplačil za odškodnine večja od 0 ter da v znesek odškodnin vključim samo dejanska izplačila, ne pa tudi zneska rezervacije po popisu.

3.2.3.5 Pomožni podatki o škodah

Kot sem zapisal že v podpoglavju 3.1, sem podatke v tej fazi pridobil brez časovne omejitve navzgor (razen omenjene implicitne omejitve zaradi prehoda na nov sistem cenikov), vendar z zavedanjem, da bo takšna omejitev nujno potrebna, saj sem želel v nabor podatkov zajeti čim manj primerov, na katerih bo še prišlo do sprememb, na primer izplačil odškodnin. Te omejitve nisem želel določiti subjektivno, ampak s časovno analizo škodnega dogajanja (podpoglavje 3.6). Za ta namen sem v tej fazi dodal nekaj pomožnih atributov za opazovanje škodnega dogajanja, kateri niso bili namenjeni učenju ali napovedovanju. Ločeno sem shranil število prijavljenih škodnih primerov ter znesek izplačil za odškodnine v prvem, drugem in tretjem letu po začetku kritja police ter posebej še v preostalih letih. Pri tem prvo leto po začetku kritja police pomeni leto, v katerem je obstajalo kritje po polici, če predpostavimo, da je polica sklenjena za eno leto. Kot dodatno informacijo sem v poseben atribut shranil še število odprtih škodnih spisov na datum poročanja.

Sicer bi podatke iz tega podpoglavja lahko pridobil tudi ločeno s posebno poizvedbo, vendar pa se mi je zdelo preprosteje, da jih dodam kar na tem mestu, ter s stališča analize tudi bolj praktično, da so shranjeni kar zraven osnovnih podatkov; v primeru, da bi se odločil za spremembo nabora podatkov, ki sem jih zajel, bi analizo zlahka ponovil na novem naboru, brez spreminjanja pomožnih poizvedb.

3.2.4 Izpeljani atributi

Izpeljani atributi so atributi, ki jih dobimo z uporabo dveh ali več osnovnih atributov (Sharma & Osei-Bryson, 2009, str. 58). V nekaterih primerih lahko uporaba določenih atributov neodvisno drug od drugega rezultira v slabšem modelu, uporaba izpeljanega atributa (na primer količnika dveh atributov) pa se s človeškega stališča zdi tudi bolj smiselna (Sharma & Osei-Bryson, 2009, str. 58). Sharma in Osei-Bryson (2009, str. 58) ugotavljata, da lahko z dodajanjem izpeljanih atributov pogosto pridemo do bolj natančnih modelov, pri čemer pa je izpeljevanje odvisno od človeške inteligence ter poglobljenega znanja o problemu.

Drug primer, v katerem je potrebna izpeljava novih atributov, so **monotone spremenljivke**; to so spremenljivke, ki se povečujejo brez meje (Pyle, 1999, str. 71). Katerakoli spremenljivka, ki je vezana na tek časa, na primer datum, je monotona spremenljivka (Pyle, 1999, str. 71). Če želimo monotone spremenljivke uporabiti v podatkovnem rudarjenju, jih je potrebno pretvoriti v nemonotono obliko (Pyle, 1999, str. 71).

Tekom identifikacije atributov sem tudi v svojem primeru zaznal določene pare atributov, pri katerih se mi je zdelo smiselno, da jih uporabim za izpeljavo novih atributov. V tej fazi sem dodal naslednje izpeljane atribute.

- Razlika v številu dni med datumom sklenitve in datumom začetka kritja zavarovanja. V tem primeru gre za razliko dveh monotonih spremenljivk, ki v podatkovnem rudarjenju nista neposredno uporabni (nek določen datum sklenitve iz učne množice, ki je že mimo, se ne more več ponoviti, tako da je neuporaben za napovedovanje primerov, ki se bodo pojavili v prihodnosti). Z vsebinskega stališča sem se za uporabo tega atributa odločil zaradi možnosti, da bi lahko obstajale razlike med osebami, ki pridejo novo zavarovanje skleniti zadnji dan pred potekom starega ali pa celo že po poteku, ter tistimi, ki takšne stvari uredijo že prej.
- Starost zavarovanca ob dnevu začetka kritja.
- Starost vozila v letih ob dnevu začetka kritja.
- Razmerje med zgodovinskim zneskom odškodnin in premij zavarovanca. Podrobnosti o teh atributih sem podal v podpoglavju 3.2.3.2. Tukaj sem za vsak par takih atributov dodal po en izpeljan atribut, skupno torej 6 atributov. Z uvedbo izpeljanih atributov sem želel algoritmu strojnega učenja pomagati do tega, da zneska upošteva povezano. Po drugi strani pa z vsebinskega stališča vidim tudi razloge za ločeno uporabo, saj nižja kot sta oba zneska, manj realno sliko kažeta (primer zavarovanca, ki ima samo eno polico ter je na njej prišlo do škode).

3.2.5 Ostali atributi

Poleg vseh opisanih atributov sem dodal še nekatere druge, ki pa niso bili namenjeni za strojno učenje ali bili predmet napovedovanja, ampak sem jih uporabil za druge namene. Za lažje razvrščanje polic po letih ter za uveljavljanje omejitev nabora podatkov v kasnejših fazah sem dodal naslednje atribute:

- datum začetka kritja police v številu dni od določenega fiksnega datuma v preteklosti,
- leto začetka kritja police,
- datum sklenitve police v številu dni od določenega fiksnega datuma v preteklosti.

Omenjeni fiksni datum iz preteklosti mi je bil seveda v vseh fazah raziskave znan, tako da je bila vedno možna pretvorba nazaj v datum v berljivi obliki.

Nazadnje naj omenim še atribute, ki sem jih dodal za navezovanje na različne tabele v relacijski podatkovni bazi. Tukaj gre za identifikatorje, ki igrajo vlogo tujega ključa (angl. *foreign key*). To so bili številka police, številka predpolice, številka police, na kateri ima zavarovanec sklenjeno zavarovanje AO, identifikacijska številka (v nadaljevanju ID) osebe, ID za povezovanje oseb (če je za isto osebo v tabeli oseb več vnosov), ID postavke na polici, ID cenika, ID vozila ter ID-ji popustov in doplačil (združeni v en atribut, ločeni z vejico).

3.2.6 Poimenovanje atributov

Korak identifikacije atributov mi je prinesel tudi prvi vpogled v njihove vrednosti. Atribute, za katere sem menil, da jih bo potrebno v nadaljnjih korakih še podrobneje pregledati in obravnavati, sem si označil z uporabo predpon v imenih, in sicer z uporabo:

- predpone G_ za nominalne atribute, pri katerih se določene vrednosti pojavijo zelo redko in jih bo potrebno združiti,
- predpone X_ za atribute, ki v izgradnji modela verjetno ne bodo prišli v poštev (zaradi velikega števila manjkajočih vrednosti ali zaradi nizke razpršenosti vrednosti – podpoglavje 2.2.5),
- predpone N_ za atribute iz podpoglavij 3.2.3.5 in 3.2.5.

3.3 Definicija tabele za vmesno shranjevanje podatkov

Skupno sem v fazi identifikacije tabel in atributov izbral 28 tabel ter okoli 170 atributov. Na podlagi izbranih atributov sem ustvaril novo tabelo, v katero sem lahko shranil pridobljene podatke, saj je bilo zaradi velike količine le-teh neizvedljivo, da bi ob vsaki zahtevi po podatkih te pridobil neposredno iz izvornih tabel. V praksi tega koraka nisem izvedel striktno za prejšnjimi, ampak sem stolpce nove tabele definiral sproti ob identifikaciji ter izbiri posameznega atributa.

3.4 Priprava PL/SQL kode ter pridobivanje podatkov

V tem podpoglavju opisujem peti in šesti korak priprave podatkov. Po končani identifikaciji atributov ter definiciji tabele za vmesno shranjevanje podatkov sem v programskem jeziku PL/SQL napisal procedure za polnjenje podatkov v ustvarjeno tabelo. Takšno je bilo približno zaporedje dela, v praksi pa sem nekatere dele programske kode napisal že sproti v prejšnjih korakih in tudi iz tega koraka sem se občasno vrnil na prejšnje, če sem zaznal kakšno pomanjkljivost, na primer atribut, ki sem ga pozabil ter bi ga bilo dobro imeti, ali možnost definicije atributa v primernejši obliki. Programska koda za pridobivanje podatkov je zajemala eno glavno proceduro, ki je napolnila večino podatkov, za kar je uporabljala tudi nekaj manjših pomožnih funkcij. Določene dele podatkov pa sem napolnil v ločenih procedurah, to so bili podatki o škodah ter podatki o zgodovini premij in izplačil zavarovanca. Pri škodah sem se tako odločil zaradi časovne zahtevnosti pridobivanja tega dela podatkov ter zato, ker sem pričakoval,

da bo zaradi pomembnosti teh atributov tu potrebnega več testiranja in ponovnega polnjenja. Pri podatkih o zgodovini izplačil in premij je bila glavni vzrok velika časovna zahtevnost, zato sem moral pridobivanje tega dela podatkov še posebej premišljeno optimizirati (po prvi, najenostavnejši različici bi polnjenje lahko trajalo tudi več kot en teden). Optimizacijo sem izvedel s tabelami za vmesno shranjevanje, iz katerih sem lahko hitreje pridobil za nek primer relevantne police ter časovno zahtevne operacije tako izvedel na čim manjšem naboru zavarovalnih polic.

Ostale pomembne podrobnosti, ki sem jih pri pisanju programske kode moral upoštevati, sem že opisal v podpoglavju 3.2 pri posameznih atributih; pogosto so bile vezane na slabo kvaliteto podatkov v podatkovni bazi, zaradi česar je bilo v programski kodi potrebno opraviti več preverjanj, pogosto tudi navzkrižnih med različnimi podatki, ter napisati kodo za čim boljše reševanje nepravilnosti. Končno število vrstic programske kode za polnjenje tabele s podatki je bilo nekaj manj kot 3.000.

3.5 Pregled in čiščenje podatkov ter atributov

S tem, ko sem imel osnovni nabor podatkov uspešno shranjen v tabeli, ki sem jo definiral za ta namen, priprava podatkov še ni bila zaključena, ampak jih je bilo potrebno še dodatno pregledati in prečistiti. Ta korak bi razdelil na tri večje sklope, in sicer:

- iskanje in odprava napak ter čiščenje atributov,
- transformacije podatkov,
- obravnava manjkajočih vrednosti.

Podrobnosti vsakega sklopa podajam v nadaljevanju.

3.5.1 Iskanje in odprava napak ter čiščenje atributov

Tudi ta del raziskave mi je vzel kar precej časa, saj se pri tako obsežni nalogi, kot so jo predstavljali prejšnji koraki, seveda pojavijo tudi napake. Pred nadaljevanjem dela sem moral preveriti podatke, ki sem jih pridobil, najti morebitne napake ter jih popraviti. Kot **napako** sem v tej fazi smatral podatke, ki niso bili skladni s podatki v izvornih tabelah ali pa sem ocenil, da ne ustrezajo dejanskemu stanju nekega primera (na primer avtomobil z vrednostjo nekaj 10 EUR). Temu sta ustrezala dva glavna vira napak, in sicer:

- napake v programski kodi za pridobivanje podatkov,
- napake zaradi slabe kvalitete podatkov v izvornih tabelah, kjer bi spet ločil dva primera:
  - slabe kvalitete podatkov prej sploh nisem opazil in obravnaval,
  - obravnava nekvalitetnih podatkov ni bila dovolj izpopolnjena.

Napake sem skušal odkriti s pregledom vrednosti v tabeli za vmesno shranjevanje, pri čemer sem bil pozoren predvsem na naslednje:

- najmanjše ter največje vrednosti numeričnih atributov,
- nabor vrednosti pri nominalnih atributih (kot predlagano v Witten et al., 2011, str. 59) ter število primerov posamezne vrednosti.

Gre za preverjanje podatkov glede na sprejemljive vrednosti (Nisbet et al., 2009, str. 56–57), pri čemer si lahko pomagamo z različnimi orodji, ki omogočajo hiter vpogled v podatke, na primer s prikazom porazdelitve vrednosti. Sam se v tem koraku nisem poslužil kakšnega orodja, temveč sem to nalogo izvedel kar s poizvedbami v podatkovni bazi. To mi je omogočilo več fleksibilnosti pri poizvedbah ter takojšnje nadaljevanje dela brez predhodnega spoznavanja z novim orodjem, hkrati pa ni bilo potrebe po izvažanju podatkov iz podatkovne baze v drugo orodje ob vsaki spremembi. Prednost je bila tudi v tem, da sem lahko podatke v svoji tabeli preko pomožnih atributov (podpoglavje 3.2.5) vedno hitro povezal z ostalimi tabelami v podatkovni bazi. Glavni slabosti sta bili počasno napredovanje in včasih mukotrpno pisanje poizvedb; ocenjujem, da odločitev ni bila najboljša ter da bi lahko s pomočjo namenskega orodja to nalogo opravil hitreje.

Kadar sem našel sumljive vrednosti, sem moral raziskati vzrok, kar je često pomenilo, da sem moral ročno pregledati posamezne primere. Če sem odkril, da napaka dejansko je prisotna, sem jo moral odpraviti, kar je pomenilo vrnitev na prejšnji korak, implementacijo ustrezne obravnave v PL/SQL kodi ter ponovno polnjenje podatkov. Določene primere sem lahko rešil tudi neposredno s posodobitvijo podatkov v svoji tabeli za vmesno shranjevanje s SQL stavkom, pri čemer pa sem ustrezne dopolnitve zmeraj vnesel tudi v osnovno PL/SQL kodo. Obravnavo napak zaradi slabe kvalitete podatkov sem se trudil rešiti sistemsko, torej napisati programsko kodo za splošno obravnavo, je pa bilo tudi nekaj primerov napak, za katere sem napisal namensko obravnavo (večinoma je šlo za zamenjavo neustreznih vrednosti z ročno ugotovljenimi pravilnimi vrednostmi). V zelo redkih primerih sem se poslužil tudi brisanja primerov, ki so vsebovali veliko nesmiselnih podatkov (Nisbet et al., 2009, str. 57) in pri katerih včasih tudi ročno sploh ni bilo mogoče ugotoviti, katere bi bile pravilne vrednosti.

Nisem pa se omejil samo na vrednosti podatkov, ampak je ta korak zajemal tudi čiščenje atributov. Tukaj sem bil pozoren predvsem na sledeče:

- atributi z nizko razpršenostjo, na primer več kot 99,5 % primerov je imelo vrednost 1, preostali 0 ali obratno,
- atributi z veliko manjkajočimi ali napačnimi vrednostmi, ki jih sistemsko nisem mogel popraviti;

v teh primerih sem uporabil eno izmed naslednjih rešitev:

- odstranitev atributa (konceptualno je to ustrezalo izbiri atributov na podlagi razpršenosti ali števila manjkajočih vrednosti – podpoglavje 2.2.5),
- sprememba oblike atributa.

Med atributi z nizko razpršenostjo naj kot primera omenim zavarovanje izgube vrednosti vozila ter zavarovanje ponovnega opravljanja vozniškega izpita, ki ju lahko zavarovanec dodatno sklene, podobno kot delne kasko kombinacije. Teh primerov je bilo med več kot 2.000.000 vrstic v osnovnem naboru podatkov le nekaj 10, zato sem se odločil, da atributa odstranim. Podoben primer predstavlja dodatno zavarovanje asistence, ki ga ima v eni izmed oblik sklenjenega velika večina zavarovancev; tudi ta atribut sem izločil. Spremembo oblike atributa pa sem na primer uporabil pri dveh izmed delnih kasko kombinacij s podobno vsebino, ki se med seboj izključujeta, zato sem ločena atributa združil v skupnega.

3.5.2 Transformacije podatkov

V izbrani množici atributov so bili tako numerični kot nominalni. Ker uporaba nominalnih atributov pri nekaterih algoritmih strojnega učenja, na primer pri nevronskih mrežah ali logistični regresiji, ni mogoča, jih je za te algoritme potrebno pretvoriti v numerične. Eden od načinov, kako to dosežemo, je uvedba novih atributov, pri čemer za vsako od vrednosti nominalnega atributa uvedemo nov binarni atribut, imenovan tudi psevdo-spremenljivka (Pyle, 1999, str. 193). Pyle (1999, str. 193) to tehniko imenuje 1-proti-n premapiranje (angl. *one-of-n remapping*) ter izpostavlja očitno slabost, da s tem izjemno povečamo dimenzijo problema, če ima nominalni atribut veliko število različnih možnih vrednosti oziroma če imamo veliko število takšnih atributov. Druga težava, na katero sem računal, je bila nizka razpršenost vrednosti psevdo-spremenljivk za nominalne vrednosti, ki se redko pojavljajo.

Takšnih primerov je bilo v mojih podatkih kar veliko. Glavni primer so bili proizvajalci ter modeli vozil. Zdelo se mi je vsekakor zanimivo dati strojnemu učenju možnost, da ugotovi morebitno povezavo med proizvajalcem vozila ter tveganostjo zavarovanca, vendar pa se je v naboru mojih podatkov pojavilo kar 61 različnih proizvajalcev ter preko 3.000 različnih modelov vozil. Seveda bi bilo popolnoma nesmiselno dodati takšno število psevdo-spremenljivk, dodaten problem pa vidim tudi v tem, da bi za mnoge imel zelo malo število učnih primerov in smiselna povezava z rezultati ne bi bila mogoča. Po drugi strani pa bi znali biti proizvajalci, ki se pojavijo v velikem številu primerov (nad 7 %, nekateri celo malo pod 15 %) vsekakor zanimivi. Zato sem se odločil, da takšne nominalne atribute že na tej stopnji obdelam ter pustim samo nekaj najbolj pogostih ter zanimivih vrednosti, vse ostale pa združim v novo vrednost, ki sem jo v večini primerov označil kot »Ostalo«. Naj omenim, da je tak pristop podoben, kot če bi v tej fazi pustil vse različne vrednosti ter kasneje, po kreiranju psevdo-spremenljivk, pri izbiri atributov zmanjšal dimenzijo problema z odstranjevanjem določenih psevdo-spremenljivk (metoda izbire atributov na podlagi razpršenosti).

Podoben primer so bile tudi pošte stalnega prebivališča zavarovanca in prodajne enote, kjer je bilo zavarovanje prodano. Tukaj sem se odločil, da jih, glede na prvo številko poštne številke,

združim v 8 poštnih okrajev. Enak hierarhičen pristop sem uporabil pri registrskih območjih avtomobilov, le da sem te ročno, na podlagi geografske bližine, združil v večje regije, na primer Primorska (registrske oznake GO, KP, PO) ali Dolenjska (KK, NM), pri čemer pa nekaterih bolj zastopanih (MB, LJ) nisem združil.

Drugačen tip transformacije sem opravil pri barvi vozila. Le-ta je v podatkovni bazi shranjena v obliki kod s tremi znaki, pri katerih prvi znak pove, ali gre za kovinsko barvo, drugi pove barvo kot takšno (10 stopenj od bele do črne), tretji pa intenziteto barve (svetla, srednja, temna). Tukaj sem se odločil, da naredim poseben atribut za kovinsko barvo ter posebnega za naziv barve, tretji znak pa sem zanemaril, da ne bi dobil prevelikega števila psevdo-spremenljivk.

3.5.3 Obravnava manjkajočih vrednosti

Kot je razvidno že iz dosedanjega opisa, so pri določenih primerih vrednosti za nekatere atribute manjkale. Za uporabo teh primerov v algoritmih strojnega učenja sem moral te **manjkajoče vrednosti** (angl. *missing values*) nadomestiti z neko veljavno vrednostjo ali pa primer izbrisati. Pri tem sem se odločil za različne metode. Pri nominalnih atributih sem se odločil za posebno kategorijo (»Neznano« ali »X«). Pri numeričnih pa sem preveril, ali lahko vrednost vsaj ocenim na podlagi ostalih, kar pristop uvršča v skupino **metod enkratnega vstavljanja** (angl. *single-impute methods*) (Palfy, 2009, str. 45). Veliko manjkajočih vrednosti je bilo v primeru podatkov iz kataloga vozil – če ustreznega ID-ja vozila nisem imel ali ga v katalogu nisem našel, so vse vrednosti manjkale. Vendar pa sem vedno imel vsaj en podatek o vozilu, in sicer moč motorja v kW (dejansko je v osnovnem naboru podatkov obstajalo 0,004 % primerov, ki tega podatka niso imeli, ki pa sem jih izločil, kar predstavlja **metodo brisanja nepopolnih vzorcev** (angl. *case deletion*) (Palfy, 2009, str. 44)). Sklepal sem, da so si vozila s podobno močjo motorja podobna tudi v drugih lastnostih, na primer prostornini motorja. Manjkajoče vrednosti sem zato nadomestil tako, da sem podatke najprej uredil po moči motorja, nato pa vzel vrednost iz zadnje predhodne vrstice, ki je imela podatek izpolnjen; manjkajoče podatke na začetku urejenega nabora podatkov (vrednosti v prvih vrsticah so manjkale) pa sem zapolnil s podatki iz prve neprazne vrstice. Pri atributih za število vrat, število prestav ter število valjev motorja sem uporabil celoštevilske vrednosti, ki se najbolj pogosto pojavljajo (na primer največji delež vozil v mojem naboru podatkov ima 5 vrat) (na to bi lahko gledali kot na različico metode nadomeščanja s povprečno vrednostjo). Poseben pristop sem uporabil še pri oceni premijskega razreda za AO – tukaj sem sklepal, da bi voznik lahko bil v podobnem premijskem razredu kot za AK (kar seveda ni nujno res) ter v primeru manjkajočega podatka za AO uporabil AK vrednost (to je spet metoda enkratnega vstavljanja). Pri starosti vozila ob dnevu začetka kritja sem se odločil za **metodo nadomeščanje s povprečno vrednostjo** (Palfy, 2009, str. 44); vstavljeno vrednost sem nato uporabil, da sem v takšnih primerih izračunal vrednost atributa za letnik vozila (starost vozila in letnik vozila sta vedno manjkala v paru).

Posebne transformacije sem opravil še pri dveh atributih za popusta, ki se pojavljata šele od določenega datuma naprej. V teh dveh primerih sem vrednost 0, ki pomeni, da popusta na polici ni, za primere, ki so bili sklenjeni pred začetkom dodeljevanja tega popusta, zamenjal z

vrednostjo »X«, saj menim, da ne bi bilo dobro enačiti primerov, kjer popust ni bil dodeljen, s tistimi iz preteklosti, ko ni niti mogel biti dodeljen. To transformacijo omenjam v sklopu obravnave manjkajočih vrednosti, ker menim, da vsebinsko dejansko gre za manjkajočo vrednost (ne vemo, ali bi bil popust dodeljen ali ne, če bi takrat že obstajal), ki pa je bila zaradi narave pridobivanja podatkov zapolnjena z 0.

3.6 Izbor podatkov

V tem koraku sem na podlagi analize podatkov, ki sem jih zajel v prejšnjih korakih, dopolnil omejitve iz prvega koraka. Dodal sem naslednje omejitve.

- Zajel sem samo police, na katerih je zavarovanec **fizična oseba** ter sta izpolnjena podatka **datum rojstva** ter **spol**. Razlogi za omejitev na fizične osebe se navezujejo na razlago v podpoglavju 3.2.1 ter na omejitev na zasebni sektor, ki sem jo podrobneje predstavil v prvem koraku priprave podatkov. Fizičnih oseb je v osnovnem naboru podatkov sicer bilo 92,46 %. Omejitev glede na datum rojstva ter spol sem dodal zato, ker sem ugotovil, da je v veliko primerih, kjer ta podatka nista bila izpolnjena, zavarovanec v resnici pravna oseba, ki pa je bila v podatkovni bazi napačno označena kot fizična. Za občutek, kakšno težo je imela ta omejitev, naj navedem, da od oseb, ki so bile označene kot fizične, 0,06 % ni imelo vpisanega veljavnega datuma rojstva, od preostalih pa še nadaljnjih 0,64 % ni imelo podatka o spolu. Konceptualno je tukaj opisana odločitev po mojem mnenju blizu brisanju nepopolnih vzorcev pri obravnavi manjkajočih vrednosti.
- Izločil sem nekatere posebne AK produkte, ki so predstavljali manjši delež primerov in bi zaradi specifičnih lastnosti zaslužili posebno obravnavo.
- Na podlagi analize škodnega dogajanja sem določil zgornjo časovno mejo za podatke, ki so še primerni za mojo raziskavo. Podrobnosti predstavljam v nadaljevanju.

3.6.1 Časovna omejitev podatkov za raziskavo navzgor

Eden izmed ciljev moje raziskave je bila napoved tveganosti zavarovanca na podlagi podatkov o škodnem dogajanju, zato je bilo pomembno, da so ti podatki čim bolj popolni. Škodni primeri se seveda lahko pojavijo samo v obdobju kritja zavarovalne police, kar pa ne pomeni, da je ob izteku le-tega škodno dogajanje zaključeno. Zavarovanci lahko škode, ki so nastale v obdobju kritja police, namreč prijavijo tudi po poteku le-tega, upoštevati pa je potrebno tudi čas, ki je potreben za obravnavo škodnega primera; tako lahko po poteku kritja police mine še kar nekaj časa do trenutka, ko so vsi škodni primeri dokončno zaključeni. V svojo raziskavo sem zato želel zajeti takšen nabor polic, pri katerem je večina škodnih primerov že zaključena in je znano končno izplačilo. Poiskati sem torej moral najkasnejši še sprejemljivi datum za zajem primera v raziskavo. Za določitev ustreznega kriterija sem se odločil analizirati škodno dogajanje na policah. Pri tem sem opazoval naslednje kategorije:

- prijave škodnih primerov,

- izplačila,
- število nezaključenih škodnih primerov.

Prvi dve kategoriji sem opazoval v odvisnosti od časovne oddaljenosti od začetka kritja police z natančnostjo enega leta. Zadnjo kategorijo pa sem opazoval v odvisnosti od datuma sklenitve police ter datuma poročanja.

V zvezi s prijavami škodnih primerov sem opazil, da je daleč največji del škodnih primerov prijavljen že tekom kritja police, bistveno manjše, a nezanemarljivo število pa še v prvem letu po poteku kritja police. Rezultat ni presenetljiv, saj naj bi po pogojih za AK zavarovanja izbrane zavarovalnice zavarovanci škodne primere prijavili najkasneje v roku 8 dni od trenutka, ko so izvedeli za nastanek primera; ker se lahko le-ti dogodijo tudi manj kot 8 dni pred iztekom kritja, je razumljivo, da so nekateri škodni primeri prijavljeni tudi po poteku kritja police, četudi bi se vsi zavarovanci držali tega pogoja.

Glede deleža izplačil sem ugotovil, da je v prvem letu po izteku kritja police precej večji, kot delež prijavljenih škodnih primerov v tem obdobju. Tudi ta rezultat ne preseneča, saj je za obdelavo škodnega primera potreben čas in lahko od prijave škodnega primera do zadnjega izplačila in dokončne rešitve preteče nekaj časa. Moje izkušnje so tudi takšne, da gre pri škodnih primerih, ki se razrešujejo dalj časa, običajno za višje zneske izplačil. Ne glede na to pa je moja analiza pokazala, da je prispevek izplačil po preteku dveh let od začetka kritja police majhen.

Na podlagi teh rezultatov sem sklenil, da moram od začetke kritja police do datuma, ko polico zajamem v raziskavo, pustiti najmanj dve leti časa. Temu kriteriju sem se iz previdnosti odločil dodati še eno dodatno leto. Glede na datum poročanja (29.02.2016) je to pomenilo, da lahko po tem kriteriju zajamem police, katerih kritje se je začelo pred 01.03.2013.

Nazadnje sem se odločil preveriti še število odprtih škodnih primerov na datum poročanja v odvisnosti od datuma sklenitve police. Pri tem sem ugotovil, da je število še odprtih škodnih primerov na datum poročanja za police, ki so bile sklenjene pred 01.01.2013, skoraj zanemarljivo, nato pa počasi raste ter močno naraste za police, ki so bile sklenjene po začetku leta 2014; slednje se mi zdi skladno z ugotovitvami analize izplačil.

Ker sem želel mejo postaviti na leto natančno in ne na kak datum med letom, sem se v kombinaciji obeh ugotovitev odločil, da v raziskavo zajamem police, ki so bile sklenjene najkasneje 31.12.2012.

4 MODELIRANJE IN PREDSTAVITEV REZULTATOV

V tem poglavju združeno obravnavam predvsem četrto in peto fazo procesa podatkovnega rudarjenja glede na CRISP-DM pristop, torej modeliranje in ovrednotenje modela, na primernih mestih pa se dotaknem tudi vključitve v uporabo (šesta faza).

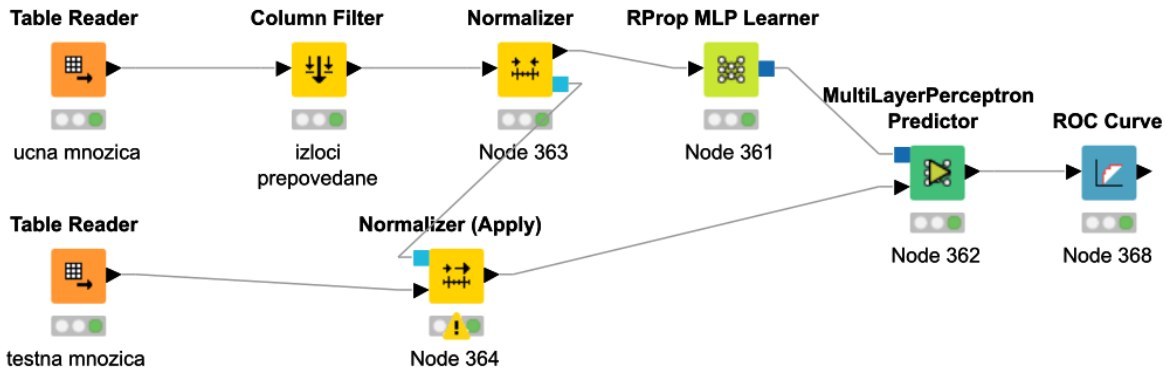
4.1 Predpriprava

4.1.1 Orodja za podatkovno rudarjenje

Za uporabo pri izvedbi tega dela raziskave sem vzel v obzir programske pakete **R**, **Weka** (angl. *Waikato Environment for Knowledge Analysis*) (Hall et al., 2009) in **KNIME** (angl. *The Konstanz Information Miner*) (Berthold et al., 2009). Vsi so brezplačno dostopni pod licenco GNU GPL (angl. *GNU General Public License*). Pri Weki moram omeniti njeno dvojno naravo – po eni strani je celovito orodje z uporabniškim vmesnikom (kateri se mi je osebno zdel nekoliko neroden za uporabo), po drugi strani pa lahko Wekine implementacije algoritmov strojnega učenja, ki so napisane v programskem jeziku Java, uporabljamo tudi samostojno.

Orodje KNIME mi je bilo od omenjenih v mojih začetnih testiranjih najbolj všeč, zato sem se odločil, da za izvedbo modeliranja in ovrednotenja rezultatov uporabim to orodje. Delo v njem poteka s povezovanjem **vozlišč** (angl. *nodes*) v **delotoke** (angl. *workflows*). Primer delotoka v orodju KNIME prikazuje Slika 3. Vozlišča so namenjena izvedbi raznih operacij, od preprostih (na primer izločanje ali preimenovanje atributov, filtriranje primerov glede na nek kriterij, prikaz rezultatov) do zelo kompleksnih (gradnja modela z algoritmom strojnega učenja, napovedovanje na osnovi izgrajenega modela, nadvzorčenje z algoritmom SMOTE). Vozlišča imajo različno število vhodov in izhodov, ki jih uporabnik v uporabniškem vmesniku povezuje (izhod enega vozlišča na vhod naslednjega) ter s tem »pošlje« rezultat obdelave podatkov v enem vozlišču v nadaljnjo obdelavo v drugem. Število različnih vozlišč, ki so na voljo in predstavljajo razne operacije, je res veliko, uporabniki pa jih lahko razvijajo in prispevajo tudi sami. Podpora na spletnih forumih se mi je zdela zelo dobra, na spletu pa so dostopni tudi številni primeri delotokov, ki nam lahko pomagajo pri učenju uporabe orodja KNIME oziroma nam dajo ideje, kako rešiti nek naš problem. Dobra je tudi dokumentacija vozlišč (in njihovih nastavitev), ki je praktično prikazana znotraj uporabniškega vmesnika (ko kliknemo na neko vozlišče). Orodje KNIME preko posebnih vozlišč omogoča uporabo Wekinih algoritmov strojnega učenja, nadalje pa lahko v temu namenjenih vozliščih izvajamo tudi R-ovo programsko kodo (obojega sem se tekom raziskave tudi poslužil); tako lahko združimo prednosti – po mojem mnenju odličnega – uporabniškega vmesnika orodja KNIME z določenimi funkcionalnostmi ostalih dveh omenjenih orodij. Uporaba drugih orodij znotraj orodja KNIME je tudi ena izmed možnih rešitev, če želimo izvesti operacijo, ki je KNIME preko svojih vozlišč ne podpira; še ena možnost je uporaba vozlišč, v katerih lahko izvajamo poljubno Java programsko kodo. V orodju KNIME tako skoraj ni omejitev glede operacij nad podatki, ki jih lahko izvajamo, ob tem pa nam nudi preprost in prijazen uporabniški vmesnik ter izdatno podporo (dokumentacija, primeri, spletni forumi), kar omogoča, da se dela z orodjem KNIME zelo hitro naučimo in ga začnemo učinkovito uporabljati za reševanje dejanskega problema. Še ena lastnost, ki se pri delu z velikimi količinami podatkov hitro izkaže za zelo koristno, je, da lahko sestavljene delotoke izvajamo po delih, pri čemer vmesni rezultati ostanejo shranjeni v okviru vozlišča, brez da bi jih morali eksplicitno zapisati v datoteko na trdem disku (kar seveda tudi lahko storimo). Če torej na primer samo spremenimo prikaz rezultatov ali algoritem strojnega učenja, nam ni potrebno delotoka izvesti od samega začetka, ki običajno vsebuje časovno zahtevno branje in predobdelavo velike količine podatkov.

Slika 3: Preprost primer delotoka v orodju KNIME – filtriranje atributov, učenje nevronske mreže in ovrednotenje na testni množici s pomočjo ROC krivulje



4.1.2 Izbran scenarij

V raziskavi sem želel preveriti uporabo pristopa na kar se da praktičnem primeru. Zato sem si zamislil naslednji scenarij: recimo, da smo na koncu leta 2011 ter želimo zgraditi model za uporabo v prihodnje. Ker želimo za izgradnjo modela uporabiti le čim bolj popolne podatke, bomo, skladno z ugotovitvami iz podpoglavja 3.6.1, uporabili police, ki so bile sklenjene najkasneje 31.12.2008. Dobljeni model bomo začeli uporabljati na primerih iz leta 2012, vendar pa takoj še ne bomo mogli zanesljivo oceniti njegove uspešnosti; če striktno sledimo omejitvi iz podpoglavja 3.6.1, bomo oceno uspešnosti opravili šele v letu 2016, seveda pa bomo že prej lahko spremljali rezultate.

Imel sem torej vse potrebne podatke za preizkus omenjenega scenarija, saj, skladno s podpoglavjem 3.6.1, police, ki so bile sklenjene v letu 2012, smatram za zaključene. Te police sem uporabil za končno oceno uspešnosti. Model sem se, enako kot bi postopali v realnosti, namenil zgraditi na policah, ki so bile sklenjene najkasneje 31.12.2008, ter na njih opraviti tudi optimizacijo parametrov modela, saj tega seveda nisem smel narediti na policah iz leta 2012, kajti v realnosti v trenutku izgradnje modela teh podatkov še ne bi imeli.

Pomembno je poudariti še, da eden od parametrov tega preizkusa ne ustreza povsem scenariju, ki bi nastopil v realnosti. Če bi model gradili v začetku leta 2012, bi kot datum poročanja, torej datum, na katerega smo pridobili podatke, vzeli 31.12.2011, moji podatki pa so bili pridobljeni z datumom poročanja 29.02.2016. To pomeni, da so bile v mojih podatkih zajete tudi obdelave škod (prijave, izplačila, odkloni, …), ki so se zgodile med 01.01.2012 in 29.02.2016. Vendar pa sem menil, da lahko glede na ugotovitve iz podpoglavja 3.6.1 to podrobnost na tem mestu zanemarim.

4.1.3 Uporabljen pristop

Pri spoznavanju s procesom podatkovnega rudarjenja sem ugotovil, da v koraku modeliranja na podlagi podatkov iščem model, ki bo vhodne podatke preslikal v napovedi. To iskanje poteka v

večdimenzionalnem prostoru, ki je določen vsaj z izbranim algoritmom strojnega učenja in njegovimi parametri, z izbiro atributov, ki mu jih predstavimo (podpoglavje 2.2.5), ter z izbiro, izvedbo in parametri povezanih postopkov (na primer za odpravo neuravnoteženosti problema). Obstaja torej nešteto različnih kombinacij, pri čemer nam lahko vsaka da različen model; večji del prostora rešitev preiščemo, večje so možnosti, da bomo našli dober model. Kolikšen del bomo preiskali, pa je seveda odvisno od naših zahtev in časa, ki ga imamo na razpolago.

Glede na zapisano mi je bilo jasno, da si bom pri iskanju modela moral postaviti določene **omejitve** in se zadovoljiti s tem, da preiščem le del omenjenega prostora; izbrane omejitve in razloge zanje bom sproti navedel pri vsakem od posameznih korakov modeliranja.

Kot sem že omenil, sem imel zavarovanje polnega AK v podatkih še dalje razdeljeno glede na zavarovane nevarnosti. V nadaljevanju raziskave sem se osredotočil samo na zavarovano nevarnost **prometne nesreče**, na katero sodi največji del zneskov premij in odškodnin (okoli 75 %), hkrati pa ta kategorija ni toliko odvisna od naravnih pojavov, na primer toče in poplav, ki po moji subjektivni oceni niso vezani na lastnosti zavarovanca oziroma njegove vozniške sposobnosti (lahko pa so seveda na primer na kraj prebivališča). Seveda bi bilo raziskavo mogoče analogno ponoviti še na ostalih treh skupinah zavarovanih nevarnosti (naravne sile, požar, poškodovanje vozila).

Modeliranja sem se, kot rečeno, lotil v orodju **KNIME**. Tehničnim podrobnostim modeliranja se v nadaljevanju ne bom posvečal oziroma bom navedel le najbolj pomembne ali zanimive podrobnosti.

4.1.4 Predhodna izločitev nekaterih atributov

V tem koraku sem še pred izvedbo strojnega učenja pregledal nabor atributov ter odstranil tiste, ki so predstavljali **anahronistične spremenljivke** ali pa so izkazovali **veliko medsebojno odvisnost**.

Anahronistična spremenljivka (angl. *anachronistic variable*) je spremenljivka oziroma atribut, ki vsebuje informacije, ki v resnici niso na razpolago v trenutku, ko potrebujemo napoved (Pyle, 1999, str. 77). Pri napovedovanju nastopa škodnega primera bi bil tak primer atribut z zneskom povprečne odškodnine na zavarovalni pogodbi. V naboru učnih in testnih primerov mi je bil ta podatek seveda na voljo (škodno dogajanje na teh policah je bilo v večini primerov že zaključeno), vendar pa ga ob sklenitvi police, ki predstavlja trenutek, ko bi potrebovali napoved mojega modela, še ne poznamo. Če bi se ta atribut nekako prikradel v nabor atributov za učenje, bi lahko na njegovi podlagi algoritem skoraj povsem zanesljivo napovedal prisotnost škodnega primera na učnih in testnih podatkih in dosegel neverjetno dobre rezultate (povsod, kjer je znesek povprečne odškodnine večji od 0, je očitno prišlo do škodnega dogodka), vendar pa bi bil neuporaben za napovedovanje na primerih v praksi. Čeprav se zdi, da je navedeno očitno in morda ni vredno omembe, pa je, kot opozarja Pyle (1999, str. 77), pri velikem številu atributov

hitro možno spregledati kakšnega, ki je anahronističen, še zlasti, ker vsi primeri niso tako očitni kot ta, ki sem ga navedel.

Seznam atributov sem zato pregledal in iz nabora atributov za učenje odstranil vse, ki so glede na trenutek, ko bi moje modele uporabili za napovedovanje, predstavljali anahronistične spremenljivke, na primer atribute, ki so vsebovali informacije o škodnem dogajanju na polici ali o višini komercialnih popustov. To ne velja atribute, ki so predstavljali izhodne vrednosti, ki sem jih želel napovedovati, saj je le-te pri nadzorovanem strojnem učenju potrebno predstaviti algoritmu (seveda kot odvisne in ne kot neodvisne spremenljivke); ker so se napovedovane spremenljivke od koraka do koraka modeliranja razlikovale, sem moral na tej stopnji ohraniti vse, nato pa v posameznem koraku izločiti tiste, ki jih v njem nisem napovedoval in jih nisem smel poslati v algoritem strojnega učenja, saj bi predstavljale anahronistične spremenljivke. Omenim naj še, da sem iz nabora podatkov za napovedovanje pričakovane višine odškodnine odstranil tudi vse podatke o premiji na polici – tukaj sicer ne gre nujno za anahronistično spremenljivko, če bi se odločili moj model uporabiti za napovedovanje v povezavi z obstoječim premijskim modelom, vendar pa sem želel, da je moj model od njega neodvisen.

Odstranjevanje atributov z veliko medsebojno odvisnostjo pa predstavlja uporabo ene izmed tehnik za zmanjšanje dimenzije problema oziroma izbiro atributov, ki sem jo že omenil v podpoglavju 2.2.5. Postopka sem se lotil ročno, in sicer sem na podlagi poznavanja problema in pomena posameznih atributov ocenil, kateri pari atributov predstavljajo isto ali zelo podobno količino, ter v teh primerih enega izmed njih odstranil. Kot primere (seznam ni popoln) odstranjenih atributov lahko omenim ceno vozila po katalogu vozil (ohranil sem zavarovalno vsoto), vrsto goriva (ohranil sem vrsto motorja), bonus/malus v odstotkih (ohranil sem premijski razred), prostornino in moč motorja iz kataloga vozil (ista podatka sem imel v drugih dveh atributih, za katera je bil vir neposredni vnos in sta vsebovala manj manjkajočih vrednosti), število konjskih moči motorja (isto informacijo v drugi enoti predstavlja moč motorja v kW) in odbitno franšizo v EUR (ohranil sem vrednost odbitne franšize v odstotkih).

4.2 Napovedovanje pričakovanega zneska izplačil (tveganosti)

Odločil sem se, da bom ta problem razdelil v dva dela; v prvem delu sem poskušal napovedati verjetnost nastanka škodnega primera na polici. V drugem delu sem nato poskušal napovedati višino odškodnine, če pride do škodnega primera (vhodni podatki za napovedovanje so tukaj bili samo primeri s škodami). Po zaključku obeh delov sem rezultate združil. Podoben pristop predlaga tudi Dal Pozzolo (2010, str. 40).

4.2.1 Napovedovanje nastopa škodnega primera

V tem delu je šlo za **klasifikacijski problem**, pri katerem sem imel samo dva možna rezultata, 0 (škodnega primera ni) ter 1 (škodni primer je), problem je bil torej **dvorazredni**. Atribut za število škodnih primerov sem zato pretvoril v binarnega. Za merilo uspešnosti pri primerjavi rezultatov sem uporabil AUC vrednost, rezultate pa grafično prikazal z ROC krivuljo.

Chapados (2010) opisuje pozitivne lastnosti nevronskih mrež pri napovedovanju nevarnostne premije (ki je seveda neposredno povezana s pričakovano višino odškodnin) pri avtomobilskih zavarovanjih, zato sem si najprej ogledal ta algoritem strojnega učenja. Vendar pa najdemo tudi raziskave, v katerih se nevronske mreže niso posebej izkazale, težave pa so bile tudi s časom izvajanja (Dal Pozzolo, 2010, str. 58; Šajn Juršič, 2014, str. 53–55). Na težave (daljši časi izvajanja ob ne navdušujočih rezultatih) pri uporabi nevronske mreže sem v začetnih testiranjih naletel tudi sam; v teh testih se je kot uspešen izkazal Wekin algoritem **Logistic** (le Cessie & van Houwelingen, 1992), ki implementira **logistično regresijo** (podpoglavje 2.2.2.1), kar je skladno z rezultati sorodne raziskave (Šajn Juršič, 2014, str. 53–54). Na podlagi teh ugotovitev sem si za ta del svoje raziskave (skladno z načrtom, da upoštevam pozitivne rezultate drugih avtorjev) zastavil naslednji plan:

1. uporabi algoritem Logistic,
2. optimiziraj parametre,
3. izberi atribute,
4. z ugotovljenimi nastavitvami znova preizkusi nevronsko mrežo,
5. najboljši najdeni model uporabi v nadaljnji raziskavi.

Nabor podatkov, na katerem sem začel graditi model, so bile police, ki so bile sklenjene v letih 2006, 2007 in 2008. Za leto 2005 zaradi razlogov, ki sem jih navedel v podpoglavju 3.1, nisem imel podatkov za celo leto, zato sem se ga odločil izpustiti, saj se mi je zdelo primerneje, da je nabor polic določen na leto natančno. Zraven tega je v prehodnih obdobjih vedno večja možnost, da so v podatkih napake. Ker sem imel torej na razpolago samo podatke za 3 leta, sem se odločil, da nabora dodatno ne omejujem.

V zvezi s primeri, ki so imeli škode, sem ugotovil, da predstavljajo približno 10,3 % vseh učnih primerov. Problem torej ni bil tako zelo neuravnotežen kot v primeru podatkov o rakavih pacientih, ki ga navajata He in Garcia (2009, str. 1264), kjer je bilo pozitivnih primerov le nekaj čez 2 %. Kljub temu sem se odločil, da preizkusim, ali lahko s tehnikami za odpravo neuravnoteženosti dosežem izboljšanje rezultatov. Pri tem sem se odločil za metodo naključnega podvzorčenja. Uporaba naključnega nadvzorčenja z vračanjem se mi ni zdela primerna, ker lahko povzroči prekomerno prileganje, rezultati raziskav pa tudi niso pokazali kakšnega pomembnega pozitivnega vpliva uporabe te tehnike (Chawla et al., 2002, str. 326–328). Možna rešitev za to je algoritem SMOTE (Chawla et al., 2002), ki sem ga pred nadaljevanjem zato tudi preizkusil, vendar se, kot bom opisal v nadaljevanju, na mojem primeru ni izkazal za uspešnega.

4.2.1.1 Preizkus algoritma SMOTE

Za prvi preizkus algoritma SMOTE sem uporabil celoten nabor polic, ki so bile sklenjene v letih 2006, 2007 in 2008 ter delež polic brez škod najprej zmanjšal na 15 % osnovne vrednosti, kar je pomenilo, da je bilo polic s škodami približno 43 %. To pomeni podvzorčenje, ki ga v kombinaciji z algoritmom SMOTE predlagajo Chawla et al. (2002, str. 331). Nato sem uporabil

algoritem SMOTE, ki je dodal umetne (sintetične) primere polic s škodami, tako da jih je bilo enako število kot tistih brez škod. Končni rezultat učenja na takšnem naboru mi je dal AUC vrednost 0,6485 (kar je slabše od rezultatov uporabe samo podvzorčenja, ki jih prikazuje Tabela 2). Z opisanimi nastavitvami je bil algoritem SMOTE tudi zelo počasen. Zatem sem preizkusil enak nabor polic, pri čemer sem z razslojenim vzorčenjem velikost vzorca zmanjšal na 20 %, kar je precej pohitrilo delovanje. V tem primeru sem lahko delež polic brez škod zmanjšal na 35 % osnovne vrednosti, delež polic s škodami je tako pred uporabo algoritma SMOTE znašal približno 25 %, kar je pomenilo, da je SMOTE za vsak primer s škodo dodal še 2 umetna primera. Dobljena vrednost AUC je znašala 0,6218. Poizkusil sem še s še manjšim podvzorčenjem, delež polic brez škod sem zmanjšal na 45 % njegove osnovne vrednosti, kar je pomenilo, da je moral algoritem SMOTE za izenačenje zastopanosti obeh razredov dodati še več umetnih primerov. Rezultat AUC je bil 0,6226. Na podlagi teh rezultatov sem se odločil, da uporabo algoritma SMOTE v svojem primeru opustim. Naj omenim, da sem preizkus algoritma SMOTE izvedel z že pripravljenim vozliščem v orodju KNIME. Predstavljene vrednosti so bile dobljene na validacijski množici v deležu 25 % nabora podatkov, ki sem jo dobil z razslojenim vzorčenjem glede na napovedovano spremenljivko, za strojno učenje pa sem uporabil algoritem Logistic.

4.2.1.2 Preizkus podvzorčenja

Podvzorčenje sem izvedel tako, da sem naključno izločil nekatere primere izmed tistih, ki niso imeli škod. To sem znotraj orodja KNIME storil z uporabo vozlišča Row Sampling, ki omogoča nastavljanje deleža izbranih primerov. Odločiti sem se torej moral, koliko odstotkov primerov brez škod, ki predstavljajo večinski razred, bom ohranil, kar za moj namen ni bilo posebej intuitivno, saj je bolj zanimiv podatek delež primerov s škodami po podvzorčenju, zato sem moral narediti preračun. Podvzorčenje sem nato preizkusil pri različnih deležih števila škod po podvzorčenju ter opazoval vrednost AUC na validacijski množici velikosti 25 %. Tabela 2 prikazuje dobljene vrednosti AUC ter ohranjen delež primerov brez škod za kombinacije, ki sem jih preizkusil.

Tabela 2: Vrednosti AUC za različne stopnje podvzorčenja v kombinaciji z algoritmom Logistic

Delež primerov škod po podvzorčenju (v %)	Delež ohranjenih primerov brez škod (v %)	Vrednost AUC
10	100	0,6522
20	45	0,6524
25	35	0,6522
35	21	0,6522
50	12	0,6512

Rezultati kažejo, da podvzorčenje ni imelo pomembnega vpliva. Podobno se je zgodilo tudi že v drugih raziskavah, ki jih povzemata He in Garcia (2009, str. 1265) in navajata, da iz njih sledi, da neuravnoteženost ni edini razlog za težave pri strojnem učenju, ampak dodaten dejavnik, ki lahko poveča težave zaradi drugih vzrokov (na primer kompleksnosti podatkov).

V nadaljevanju sem kljub temu uporabil podvzorčenje večinskega razreda, in sicer sem število primerov brez škod zmanjšal na 35 % osnovne vrednosti. Za tak pristop sem se odločil zaradi pozitivnega vpliva na čas obdelave podatkov (ki ga omenja tudi Ganganwar (2012, str. 43)); iz rezultatov sem namreč sklepal tudi, da primeri brez škod ne prinesejo pomembne dodatne informacije ter da ne bo negativnega vpliva na rezultate, če jih nekaj odstranim.

4.2.1.3 Izbira atributov

Izbire atributov sem se lotil po metodi odstranjevanja atributov, ki pa sem si jo nekoliko prilagodil. V posameznem koraku odstranjevanja atributov nisem preizkusil enega po enega, ampak se poskusil odstraniti večjo skupino atributov (kot predlaga tudi Šajn Juršič (2014, str. 25)), za katero sem sumil, da nima posebnega vpliva, ter opazoval vrednost AUC, dobljeno na 25 % validacijski množici. Če sem ugotovil, da odstranitev ni bistveno zmanjšala vrednosti AUC (v nekaterih primerih jo je celo povečala), sem skupino odstranil in nadaljeval z zmanjšanim številom atributov. Zaključil sem, ko nisem več našel atributa, ki bi ga lahko odstranil brez posebnega vpliva na vrednost AUC. V začetnih fazah odstranjevanja so se spremembe AUC ob odstranitvi tudi po 5 in več atributov tipično gibale nekje med 0,0001 in 0,0003, na koncu pa več nisem našel atributa, ki bi povzročil spremembo, ki bi bila manjša od 0,001; to je bil zame znak, da z odstranjevanjem preneham. Gibanje vrednosti AUC sem si grafično predstavil in ga prikazuje Slika 4; začel sem s 141 atributi, na koncu mi jih je ostalo 17.

Opazil sem, da se je pri 36 preostalih atributih trend gibanja AUC obrnil občutneje navzdol, vendar pa sem vmes naletel na nekaj atributov, ki so vrednost povečali. Zato sem se vrnil na točko 36 preostalih atributov (atribute, ki sem jih odstranil zatem, sem dodal nazaj) ter odstranil samo tiste, katerih odstranitev je izkazala pozitiven vpliv na vrednost AUC. Po drugi strani pa sem v območju pred točko 36 atributov opazil nekaj močnejših padcev AUC, zato sem poizkusil z dodajanjem različnih kombinacij atributov, ki sem jih odstranil v tistem koraku. Tako mi je uspelo AUC povečati, na koncu sem imel 39 atributov ter vrednost AUC 0,6537, kar je izboljšava v primerjavi z izhodiščnim stanjem (141 atributov, AUC 0,6522). Seveda je potrebno dodati, da s takšnim pristopom izbire atributov nisem preveril vseh možnih kombinacij, tako da rezultat ni nujno optimalen.

Pri razlagi moram omeniti, da so spremembe AUC dejansko bile zelo majhne, iz česar sklepam, da je algoritem glavne informacije izluščil iz samo nekaj atributov. Lahko bi se tudi odločil za nadaljevanje s samo 17 ali še manj atributi, vendar pa razlika 0,001 v vrednosti AUC na velikosti mojih vzorcev še zmeraj pomeni uspešnejšo razvrstitev pozitivnega primera v razredu velikosti okoli 100 primerov, kar se mi pri napovedovanju škod polnega AK, kjer so izplačila lahko velika, ni zdelo zanemarljivo.

Sklepam, da je imel izhodiščni model preveč atributov, ki niso prinesli nove informacije, ampak so samo povečali dimenzijo problema ter so zato dali nekoliko slabši rezultat. Atribute, ki sem jih na koncu izbral, prikazuje Tabela 3, podrobnejših opisov tu ne bom navajal, saj menim, da je osnovni pomen atributa razviden iz imena.

Slika 4: Gibanje vrednosti AUC pri odstranjevanju atributov v kombinaciji z algoritmom Logistic

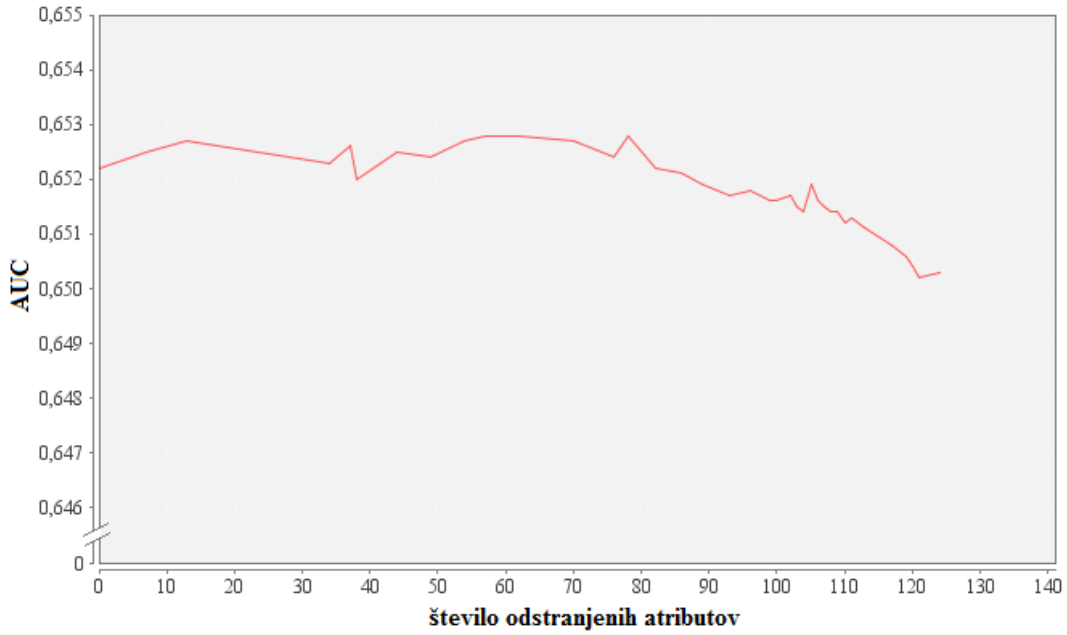


Tabela 3: Rezultati postopka izbire atributov za napovedovanje nastopa škodnega primera

Ime atributa	**Ime atributa**	**Ime atributa**
POL_TRAJANJE_DNI	P_ZAV_KM	NOVAGORICA_OSB_POSTA_GRP_N1
POL_DIFF_SKL_ZACKRI_DNI	ZAVAROVALNA_VSOTA	0_OSB_MLADI_VOZNIK
POL_IS_LEASING	AK_SKODE	1_OSB_MLADI_VOZNIK
OSB_STAROST_LET	AK_RAZMERJE	1_IZGUBA_BONUSA_AO
KASKO_FRANSIZA_PRC	AO_SKODE	0_IZGUBA_BONUSA_AOP
VOZILO_STAROST_LET	AO_RAZMERJE	1_VOZILO_KOVINSKA_BARVA
VOZILO_KW_ZRE	MARIBOR_OSB_POSTA_GRP_N1	0_VOZILO_KOVINSKA_BARVA
VOZILO_CCM_ZRE	CELJE_OSB_POSTA_GRP_N1	Citroen_VOZILO_PROIZVAJALEC
IZGUBA_BONUSA_AK	MURSKASOBOTA_OSB_POSTA_GRP_N1	Ford_VOZILO_PROIZVAJALEC
P_NACIN_PLACILA	NOVOMESTO_OSB_POSTA_GRP_N1	Opel_VOZILO_PROIZVAJALEC
PREMIJSKI_RAZRED_AK	LJUBLJANA_OSB_POSTA_GRP_N1	VW_VOZILO_PROIZVAJALEC
PREMIJSKI_RAZRED_AO	KOPER_OSB_POSTA_GRP_N1	Prednjipogon_VOZILO_POGON
P_PAKETNI	KRANJ_OSB_POSTA_GRP_N1	Zadnjipogon_VOZILO_POGON

Morda naj edino omenim podrobnost, da imamo pri nekaterih psevdo-atributih v naboru atributov tako psevdo-atribut za vrednost 0 kot tudi psevdo-atribut za vrednost 1 (primer 0_OSB_MLADI_VOZNIK in 1_OSB_MLADI_VOZNIK). Tukaj gre za primere, kje obstaja še tretja možnost, in sicer »neznano« (X_OSB_MLADI_VOZNIK); le-te v naboru atributov ni, ker je vrednost 1 v takem psevdo-atributu ekvivalentna vrednosti 0 v obeh preostalih psevdo-atributih, tako da ne prinese dodatne informacije. Je pa iz tega, da sta bila v nabor atributov izbrana oba preostala psevdo-atributa razvidno, da je razlikovanje med 0 in »neznano«

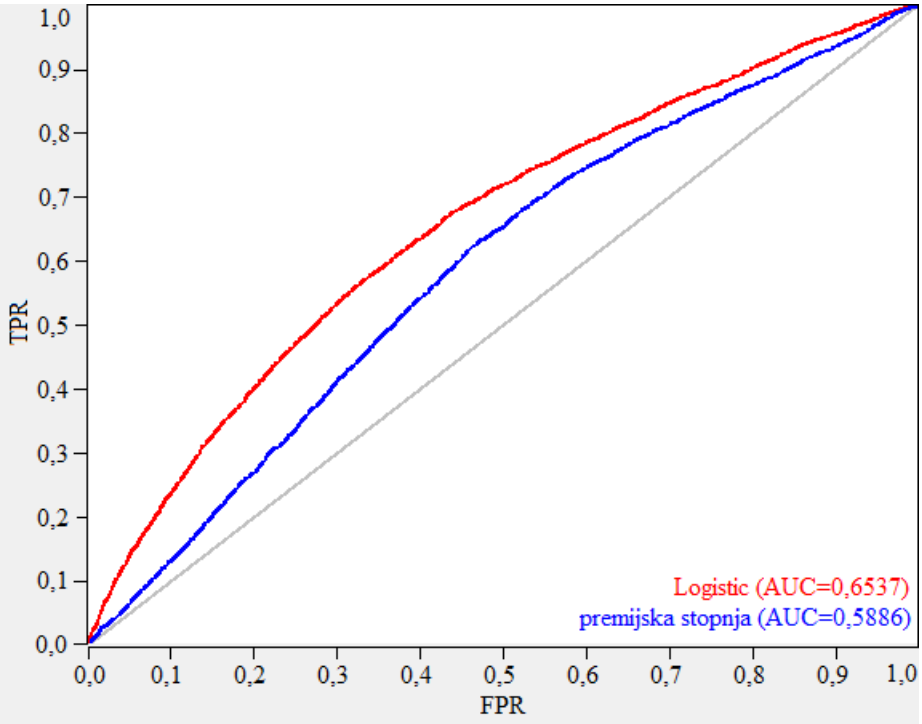
pri tem atributu za rezultate pomembno; nasprotno od tega pri atributu IZGUBA_BONUSA_AO ta razlika najbrž ni toliko pomembna, saj imamo v naboru izbranih atributov samo psevdo-atribut za vrednost 1, čeprav tudi tukaj obstajata še dve drugi možnosti (0 in X).

Krajši komentar glede vsebine izbranih atributov podajam v podpoglavju 4.2.2.2.

4.2.1.4 ROC krivulje

ROC krivulja izgrajenega modela je predstavljena na grafu (Slika 5), na katerega sem za primerjavo dodal tudi, kakšno oceno tveganja lahko izpeljemo na podlagi premije, ki je bila dobljena na osnovi aktuarskih izračunov.

Slika 5: ROC krivulja izgrajenega modela z algoritmom Logistic ter primerjava s premijsko stopnjo za police, sklenjene v obdobju 2006–2008 (validacijska množica)

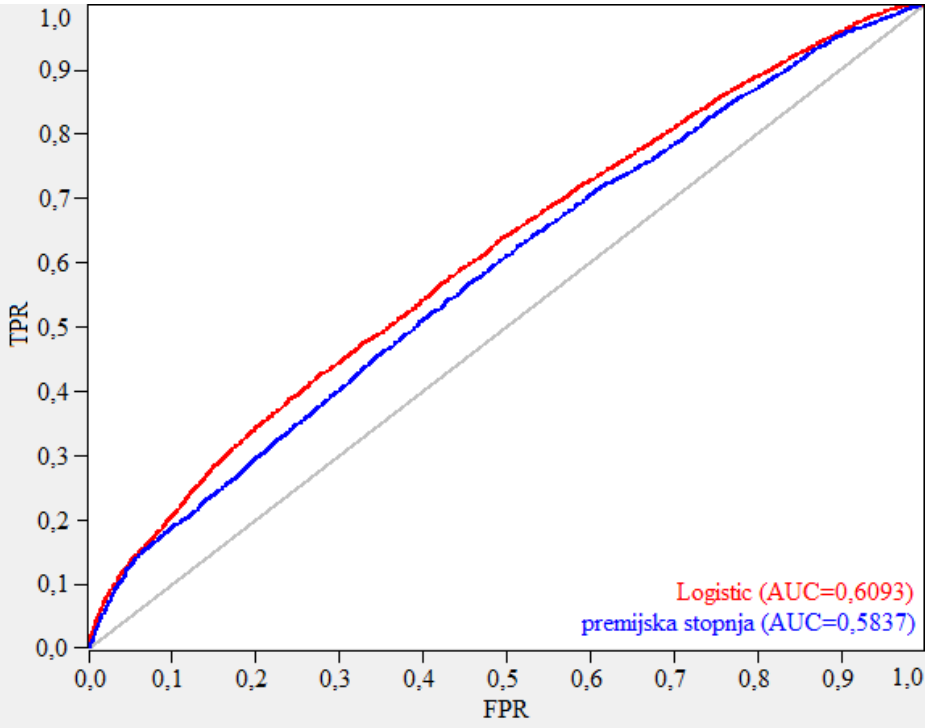


Ob tem sem moral upoštevati, da premija pri zavarovanju AK ne rabi odražati samo verjetnosti za nastanek škodnega primera, ampak tudi višino odškodnine, za katero sem na tem mestu predpostavil, da je v veliki meri povezana z vrednostjo vozila. Iz tega razloga sem kot mero tveganosti na podlagi premije oziroma aktuarskega izračuna vzel razmerje med premijo, ki jo je plačal zavarovanec, ter zavarovalno vsoto; to razmerje imenujemo tudi premijska stopnja (Močivnik, 2010, str. 49). Pri tem sem upošteval končno ceno, ki jo je plačal zavarovanec, saj je v mojem primeru dala krivuljo z višjo vrednostjo AUC, kot če bi upošteval premijo, ki ne vsebuje komercialnih popustov. V zvezi s tem zato sklepam, da je priznavanje komercialnih popustov v aktuarskem izračunu vsaj delno upoštevano. Naslednja ugotovitev na podlagi grafa (Slika 5) pa je tudi, da je bil izgrajeni model pri napovedovanju škodnih dogodkov za te primere uspešnejši.

Slednje še zlasti velja za levi spodnji kot grafa, kamor se razvrstijo najbolj tvegani primeri; tukaj je krivulja na podlagi premijske stopnje precej položna, kar pomeni, da je sem razvrstila večje število primerov, ki na koncu niso imeli škod.

Pri navedeni primerjavi s premijsko stopnjo je zelo pomembno upoštevati, da prihajajo primeri validacijske množice, na katerih sem preveril model, iz enakega obdobja kot primeri učne množice, na podlagi katerih je model nastal (kar v praktični uporabi nikoli ne bi bil slučaj), aktuarski model, ki je v tistem trenutku veljal, pa je bil narejen na podlagi podatkov iz časa pred obravnavanim obdobjem. Takšna primerjava torej ni realna, zato sem preizkus ponovil na scenariju, ki sem si ga zadal v podpoglavju 4.1.2. Izgrajeni model sem uporabil za napovedi na primerih iz leta 2012 ter ponovno opravil primerjavo s premijsko stopnjo (Slika 6).

Slika 6: ROC krivulja izgrajenega modela z algoritmom Logistic ter primerjava s premijsko stopnjo za police, sklenjene v letu 2012



Krivulji sta si na teh podatkih (Slika 6) bistveno bolj blizu in napovedi na podlagi modela so le malo boljše, kar pa še vedno velja čez celoten potek krivulj. Za opaziti je tudi, da se je AUC premijske stopnje zmanjšal, je pa njena krivulja na začetku bolj strma, tako da je možno, da je bil ta model do leta 2012 prilagojen tako, da je začel bolje razpoznavati najbolj tvegane primere in jim dodelil višje premije. Vsekakor pa je na preverjenih primerih prednost izgrajenega modela v primerjavi s prejšnjo primerjavo (Slika 5) v tem delu skopnela; glavno prednost algoritem Logistic sedaj pridobi v tem, da med le nekoliko manj tvegane (vendar še vedno tvegane – približno med 10 % in 35 % najbolj tveganih primerov) razvrsti večje število primerov s škodami. Sta pa vizualno empirično ugotovljeno obe krivulji v srednjem delu precej bolj

vzporedni z diagonalo, kar pomeni, da modela sicer znata izluščiti najbolj in najmanj tvegane primere, v srednjem delu pa je razvrstitev bolj naključna.

Zadnja primerjava po mojem mnenju nudi pomembno ugotovitev, saj prikazuje, kako velika je lahko razlika v uspešnosti napovedovanja med testiranjem izgrajenega modela na testni množici tekom modeliranja ter med uporabo modela v praksi. Razlogi so po mojem mnenju lahko različni, možna bi bila razlaga, da je do razlike prišlo zaradi spremenjenih zunanjih okoliščin in podatki niso reprezentativni za stanje v letu 2012 ali pa zaradi prekomernega prileganja, ne smemo namreč pozabiti, da je bila optimizacija (izbira algoritmov in parametrov ter atributov) narejena na validacijski množici. Prav tako je možno, da je šlo pri prvem preizkusu za nabor primerov, ki so bili še posebej primerni za napovedovanje, ali pa da se je pri drugem dogodilo ravno obratno in so primeri iz leta 2012 odstopali od »običajnih« vzorcev, ki se jih je naučil algoritem.

4.2.1.5 Vpliv nabora podatkov za učenje

Ker z rezultatom preizkusa na policah iz leta 2012 nisem bil najbolj zadovoljen, sem se odločil raziskati, ali je vzrok morda v tem, da moji učni primeri zaradi časovne oddaljenosti niso bili dovolj podobni tistim, ki sem jih želel napovedovati. Odločil sem se, da ponovno opravim učenje z obstoječimi nastavitvami modela, torej z istimi naborom izbranih atributov ter nespremenjenimi ostalimi parametri (ponovna izbira bi sicer lahko izboljšala rezultate, vendar pa bi bilo časovno zahtevno znova preveriti vse različne kombinacije), in pri tem dodam še police iz let 2009 in 2010. Ker s tem več ni bila izpolnjena omejitev, ki sem si jo zadal v podpoglavju 3.6.1, več nisem smel zanemariti, da so bili moji prvotni podatki pridobljeni z datumom poročanja 29.02.2016 in da bi bili podatki za police, sklenjene v letih 2009 in 2010, na dan 31.12.2011 manj popolni. Ker sem hotel zagotoviti, da scenarij moje raziskave še naprej temelji na realnih predpostavkah, sem znova pridobil podatke o škodah, vendar tokrat z datumom poročanja 31.12.2011. Strojno učenje sem nato ponovil na razširjenem naboru podatkov, tj. s policami, ki so bile sklenjene med leti 2006 in 2010. Dobljena vrednost AUC na policah iz leta 2012 je bila 0,6123, kar je izboljšava, četudi razlika ni zelo velika; vseeno sem na podlagi rezultata sklenil, da so police iz let 2009 in 2010 prinesle določeno dodatno informacijo za primere iz leta 2012 ter jih je smiselno upoštevati, zato sem v nadaljevanju uporabljal ta model.

Odgovoriti pa moram na vprašanje, ali sem z upoštevanjem polic iz let 2009 in 2010 naredil napako, zaradi katere moj testni scenarij ni več realen. Menim, da ne. Če bi se modeliranja lotili na začetku leta 2012, rezultatov res ne bi mogli preveriti na primerih iz leta 2012 in ugotoviti slabše napovedne zmožnosti modela ter prilagoditi nabora primerov, na katerih smo zgradili model. Vendar pa obstaja možnost, da bi na podlagi preteklih izkušenj lahko ocenili, da ima časovna bližina primerov, na katerih izvajamo učenje, večji vpliv od škodnih primerov, ki jih ne upoštevamo, ker še niso prijavljeni ali obdelani do stopnje izplačila. Prav tako bi zaradi našega znanja o problemu morda prepoznali morebitne zunanje vplive ter ustrezno prilagodili nabor primerov za izgradnjo modela.

4.2.1.6 Uporaba nevronske mreže

Na tem mestu sem se odločil, da znova preizkusim še uporabo nevronske mreže z eno skrito plastjo ter pri tem uporabim parametre (podvzorčenje, izbrani atributi, nabor polic), ki sem jih določil pri uporabi algoritma Logistic. Za določitev števila nevronov skrite plasti sem se za začetek držal priporočila, ki ga podajata Xu in Chen (2008, str. 684) (opisano v podpoglavju 2.2.2.2, enačba (5)), ter izbral 21 skritih nevronov. Za učenje sem uporabil KNIME vozlišče RProp MLP Learner, v katerem je implementiran RPROP algoritem za učenje nevronske mreže (Riedmiller & Braun, 1993) (podpoglavje 2.2.2.2). Vse atribute sem pred učenjem normaliziral v interval [0,1]. Učenje sem nato preizkusil pri različnih kombinacijah parametrov z razdelitvijo nabora podatkov na 75 % učno in 25 % validacijsko množico (police 2006–2010). Rezultati so prikazani v tabeli Tabela 4; dodan je tudi stolpec z rezultati dobljenega modela na testnem scenariju (police 2012, vsi primeri).

Tabela 4: Vrednost AUC pri različnih številih nevronov skrite plasti pri uporabi RPROP nevronske mreže (poudarjene so pomembne razlike ter najboljši rezultati)

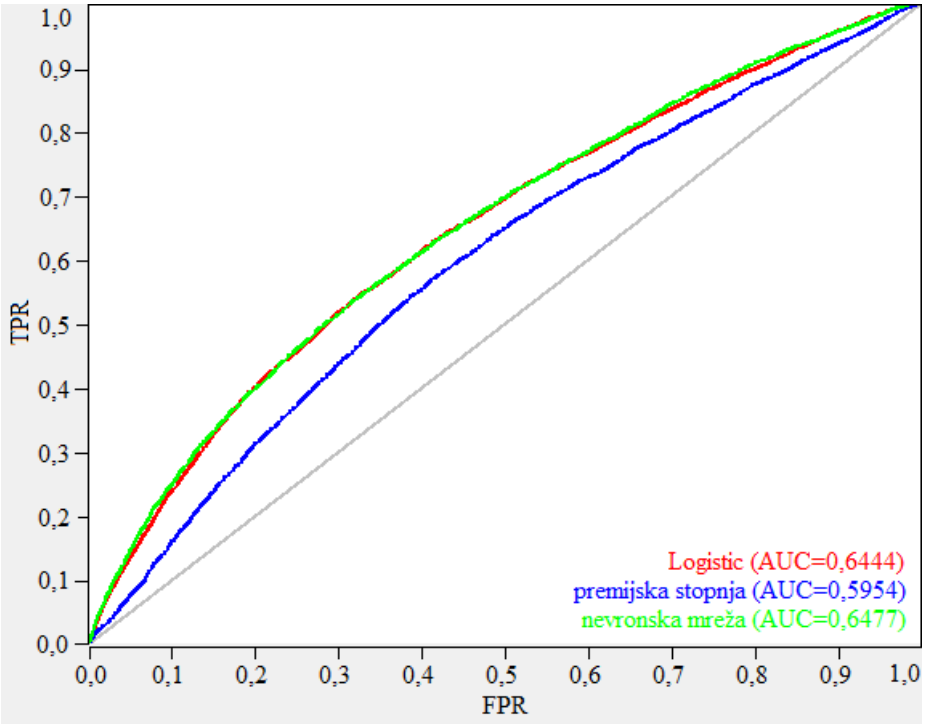
Število nevronov skrite plasti	**Število atributov**	**Število iteracij učenja**	**Vrednost AUC (police 2006–2010)**	**Vrednost AUC (police 2012)**
21	39	100	**0,6477**	0,6114
15	39	100	0,6442	0,6075
12	39	100	0,6468	0,6082
10	39	100	0,6463	0,6104
8	39	100	0,6450	0,6086
20	**141**	100	0,6383	0,5951
12	**141**	100	0,6371	0,5928
15	39	**250**	0,6472	**0,6125**
15	39	**500**	0,6469	0,6095
21	39	**500**	0,6450	0,6062

Na podlagi rezultatov sem ugotovil, da se pri mojem problemu RPROP nevronska mreža obnese podobno dobro kot algoritem Logistic (Slika 7), sem pa za to moral uporabiti zmanjšan nabor atributov (pri uporabi vseh 141 atributov so bili rezultati slabši (Tabela 4)). Pri številu nevronov skrite plasti se je priporočena vrednost 21 izkazala za dobro izbiro, podobno dobre rezultate pa mi je uspelo doseči tudi s 15 nevroni v skriti plasti z uporabo večjega števila iteracij (epoh) učenja. Vendar pa se je po drugi strani izkazalo tudi, da pretiravanje z eno ali drugo vrednostjo (število nevronov v skriti plasti, število iteracij učenja) nima pozitivnega vpliva na rezultate, predvidevam, da je prišlo do prekomernega prileganja. Glede na rezultate sem se odločil, da za primerjave v nadaljevanju uporabim model, ki sem ga zgradil na 21 nevronih v skriti plasti s 100 iteracijami učenja.

Omenim naj, da sem preizkusil tudi KNIME vozlišče MultilayerPerceptron, ki vsebuje Wekino implementacijo nevronske mreže ter za učenje uporablja klasični back-propagation algoritem, vendar pa nisem dobil zadovoljivih rezultatov (nižje vrednosti AUC ob bistveno daljšem času

izvajanja); kot stranski produkt lahko torej podam ugotovitev, da se je RPROP izboljšava back-propagation algoritma v mojem primeru izkazala za zelo uspešno.

Slika 7: ROC krivulji izgrajenega modela z RPROP nevronsko mrežo ter z algoritmom Logistic ter primerjava s premijsko stopnjo za police, sklenjene v obdobju 2006–2010 (validacijska množica)



4.2.2 Napovedovanje višine odškodnine ob škodnem primeru

V tem delu sem želel na podlagi atributov nekega primera, na katerem je prišlo do škodnega dogodka, napovedati višino odškodnine. Vhodni podatki so tukaj bili torej samo primeri s škodami, pri čemer pa nisem smel upoštevati atributa za število škod, kajti le-ta v času napovedovanja za nek nov primer še ne bo znan; če bi napovedovanje nadalje razdelil in v vmesnem koraku napovedoval še število škod, bi bilo drugače (število škod sicer ne bi bilo znano, ampak bi uporabil napoved). V slednjem primeru bi na tem mestu napovedoval povprečno odškodnino, pri mojem pristopu pa je šlo za napovedovanje celotne odškodnine, ki je torej v enem koraku zajemalo število škod ter povprečno odškodnino; za takšen pristop sem se odločil, ker delež primerov, ki so imeli več kot eno škodo, ni bil velik, okoli 7 %.

Pri napovedovanju višine odškodnine, za razliko od napovedovanja nastopa škodnega primera, ni šlo za klasifikacijski temveč za **regresijski problem**. Mera, s katero sem primerjal rezultate, je bila zato tukaj drugačna, in sicer MSE (glede na dejansko ter napovedano višino odškodnine, enačba (9)). Nabor podatkov sem v tem delu omejil na police, ki so bile sklenjene v letih 2006 do 2008 (nisem uporabil razširjenega nabora kot opisano v podpoglavju 4.2.1.5), kajti do

trenutka, ko je znano končno izplačilo, lahko mine precej več časa kot do trenutka, ko zaznamo škodo.

4.2.2.1 Izbira algoritmov

Za izvedbo regresije sem vzel v obzir dva algoritma, in sicer:

- naključni gozd,
- RPROP nevronsko mrežo.

Uporaba naključnega gozda odločitvenih dreves, ki je v orodju KNIME omogočena z vozliščem Tree Ensemble Learner (Regression), se mi je zdela privlačna zaradi tega, ker kot stranski produkt omogoča izbiro atributov (Silipo et al., 2014, str. 13). RPROP nevronska mreža se je po drugi strani na mojih podatkih pri napovedovanju nastopa škodnega primera že izkazala za uspešno, prav tako pa je bila uspešno uporabljena na sorodnem problemu (Dugas et al., 2003), zato se mi je zdelo smiselno, da jo uporabim tudi tukaj. Razmišljal sem tudi o regresiji z uporabo metode podpornih vektorjev (angl. *Support Vector Machines – SVM*), ki je v orodju KNIME dostopna z vključitvijo knjižnice LIBSVM (Chang & Lin, 2011), vendar pa sem to misel opustil zaradi ne najboljših izkušenj v prvih preizkusih (počasno delovanje in povprečni rezultati), ki so nekako potrjevali slabe rezultate pri podobnem problemu, o katerih poročajo Chapados et al. (2001, str. 1373).

V zvezi z uporabo nevronske mreže za regresijo naj omenim pomembno podrobnost, in sicer je bilo potrebno pred izvedbo strojnega učenja s tem algoritmom vse atribute ter tudi vrednost, ki sem jo napovedoval, normalizirati v interval [0,1], na koncu pa rezultate z istim modelom spet denormalizirati (uporabil sem KNIME vozlišče Denormalizer).

4.2.2.2 Izbira atributov z uporabo naključnega gozda

Lastnost odločitvenih dreves je, da znajo uporabljati nominalne atribute, tako da tukaj v tem koraku ni bilo potrebe po uvedbi psevdo-atributov. Pri uporabi naključnega gozda imamo, zraven nastavitev posameznega drevesa, še možnost konfiguracije gozda. Uporabil sem gozd 100 dreves, pri čemer se je vsako drevo učilo na 100% velikosti vzorca, ki je bil dobljen z uporabo vzorčenja z vračanjem (lastnost naključnega gozda). Velikost vzorca atributov, ki so se uporabili za posamezno drevo, sem pustil na privzeti vrednosti, ki je enaka kvadratnemu korenu števila atributov, vzorec pa se je na novo pridobil za vsako vozlišče. Dobljena vrednost MSE z uporabo 4-kratnega prečnega vrednotenja je bila 5.505.887 EUR^2. KNIME vozlišče za naključni gozd v zvezi z izbranimi atributi ponuja zanimivo statistiko, ki sem jo uporabil pri izbiri atributov. Kot že omenjeno, se je pri učenju mojega naključnega gozda za vsako vozlišče posameznega drevesa izbral naključni vzorec atributov, ki so se preverili. Atribut, ki se v nekem vozlišču preveri, imenujemo kandidat; algoritem nato izmed vseh kandidatov izbere najprimernejšega. Po učenju dobimo statistiko, kolikokrat je bil nek atribut kandidat in kolikokrat je bil dejansko izbran; statistika je ločena za prvo, drugo in tretjo raven drevesa (Slika 8).

Slika 8: Statistika atributov pri gradnji naključnega gozda, ki jo vrne KNIME vozlišče Tree Ensemble Learner (Regression)

Row ID	#splits	#can	#splits (level 1)	#candidates (level 1)	#splits (level 2)	#can
P_STAROST_VOZILA	0	3	0	31	5	83
TATVINA_FRAN_PRC	0	4	1	29	3	94
AO_RAZMERJE	0	5	1	30	5	91
VOZILO_DIFF_MASA_KG	0	5	3	31	15	101
VOZILO_EMISIJSKI_RAZRED	0	6	0	29	7	91
UNICENJE	0	6	0	31	2	93
D_POVECANA_NEVARNOST	0	6	0	33	0	113
VOZILO_MEDOSNA_RAZD_MM	0	6	10	38	16	114
POL_DIFF_SKL_ZACKRI_DNI	0	7	2	34	10	84
VOZILO_KATALIZATOR	0	7	7	32	12	94
OSB_IS_ZAVAROVANEC	0	7	1	27	5	98
NADOMESTNO_VOZILO	0	7	1	31	2	100
NEZGODE_KP	0	7	1	35	8	101
NEZGODNO_POTNIKI	0	7	1	45	3	102
ZAVAROVALNA_VSOTA	7	7	26	32	59	111
AO_KP	0	7	0	35	7	113
AK_KP	0	8	0	40	7	88
VOZILO_CCM_ZRE	7	8	15	33	24	90
D_VREDNOST_VOZILA	5	8	13	38	24	91
AK_SKODE	0	8	2	32	7	97
AK_RAZMERJE	0	8	4	38	13	99
OSB_MLADI_VOZNIK	0	8	0	25	1	106
VOZILO_STAROST_LET	0	8	1	26	13	109
VOZILO_TURNCIRC	3	9	5	34	18	110

Izbire atributov sem se lotil tako, da sem statistiko uredil naraščajoče glede na število primerov, ko je bil nek atribut kandidat, ter bil pozoren na atribute, ki so izstopali po številu dejanskih izbir. To sem storil ločeno za prvo, drugo in tretjo raven ter z unijo vseh treh izbir dobil nabor zanimivih atributov. Izbiro zanimivih atributov oziroma določitev mejne vrednosti razmerja med številom izbir in številom primerov, ko je bil atribut kandidat, sem opravil po občutku, in sicer sem sprejel kandidate, ki so bili na neki ravni izbrani vsaj v 25 % primerov. V prvem koraku sem izbral 32 atributov. Izbiro sem preveril tako, da sem ponovil učenje ter pridobil novo vrednost MSE, ki je tokrat znašala 5.501.724 EUR^2 in se je torej zmanjšala (prejšnja vrednost 5.505.887 EUR^2), čeprav sem odstranil 52 atributov; iz tega sklepam, da slednji niso vsebovali pomembne informacije za napoved. Celoten postopek sem ponovil še enkrat ter število atributov zmanjšal na 20. Nova vrednost MSE je znašala 5.549.777 EUR^2, kar je bilo slabše kot na začetku, zato sem sklepal, da sem tokrat odstranil preveč atributov, ter se odločil sprejeti nabor 32 atributov, ki sem ga izbral v prejšnjem koraku (Tabela 5).

Izbrane atribute sem primerjal z izbranimi atributi pri napovedovanju nastopa škodnega primera (podpoglavje 4.2.1.3, Tabela 3). Pri tem se kar nekaj atributov ponovi, večinoma gre tukaj za podatke, za katere bi po mojem mnenju na podlagi izkušenj pričakovali, da so povezani z napovedovanimi spremenljivkami, torej verjetnostjo nastopa škode ter pričakovano višino odškodnine; prav tako so nekateri izmed njih pogosto med parametri v premijskih cenikih za zavarovanja AK oziroma se upoštevajo ob sklenitvi zavarovanja (na primer upoštevanje škodnega rezultata zavarovanca ob dodeljevanju višjih popustov) in je bil torej njihov vpliv ugotovljen tudi s klasičnimi aktuarskimi analizami. Pri tem ne smemo pozabiti že omenjenega,

da napovedovanje višine odškodnine vsebuje tudi del tveganosti v obliki števila škodnih primerov, ki ga nisem posebej napovedoval.

Tabela 5: Izbrani atributi za napovedovanje višine odškodnine

Ime atributa	**Ime atributa**	**Ime atributa**
POL_TRAJANJE_DNI	ZAVAROVALNA_VSOTA	AK_RAZMERJE
POL_DIFF_SKL_ZACKRI_DNI	VOZILO_NAVOR_NM	OSB_POSTA_GRP_N1
OSB_STAROST_LET	VOZILO_ST_VALJEV	VOZILO_OBM_REG_ID_2
KASKO_FRANSIZA_PRC	VOZILO_ST_PRESTAV	VOZILO_PROIZVAJALEC
VOZILO_STAROST_LET	VOZILO_MEDOSNA_RAZD_MM	VOZILO_RAZRED_GRP
VOZILO_KW_ZRE	VOZILO_VOZNIK_MASA_KG	VOZILO_KAROSERIJA
VOZILO_CCM_ZRE	VOZILO_DIFF_MASA_KG	VOZILO_KATALIZATOR
PREMIJSKI_RAZRED_AK	VOZILO_SIRINA_MM	VOZILO_POGON
PREMIJSKI_RAZRED_AO	VOZILO_VISINA_MM	IZGUBA_BONUSA_AO
D_VREDNOST_VOZILA	VOZILO_TURNCIRC	IZGUBA_BONUSA_AOP
POPUSTI_CENIK_ODS	VOZILO_GRDCLEAR	

Med omenjene skupne atribute sodijo trajanje kritja police (koliko časa traja izpostavljenost), moč in prostornina motorja, starost vozila, izbrana odbitna franšiza, premijski razred zavarovanca, zavarovalna vsota, zgodovina škodnega dogajanja zavarovanca pri avtomobilskih zavarovanjih (enostavni škodni rezultat), lokacija (pošta zavarovančevega naslova oziroma območje registracije), prav tako pa so pomembni tudi popusti po ceniku, ki naj bi odražali povečano ali zmanjšano tveganje posameznega primera (pri napovedovanju nastopa škodnega primera so se med izbrane atribute uvrstili paketni popusti, tukaj pa skupen atribut za popuste po ceniku). V obeh napovedih je bila med izbranimi atributi tudi starost zavarovanca, za spol pa se po drugi strani ni izkazalo, da bi imel vpliv.

Med atributi, ki niso skupni, so se pri napovedovanju verjetnosti škodnega dogodka pojavili nekateri, ki bi jih tudi sam prej povezal s tveganostjo kot pa z višino odškodnine, na primer doplačilo za manj kot tri leta vozniških izkušenj ter dejstvo, da je nekdo zavaroval izgubo bonusa na AK (tisti, ki tega zavarovanja nimajo sklenjenega, manjših škod morda sploh ne bodo prijavili). Po drugi strani se pri napovedovanju višine odškodnine pojavi precej več atributov, ki so vezani na lastnosti vozila, kar me tudi ne preseneča, saj so ti podatki verjetno povezani s stroški popravila (ki bi znali biti večji pri močnejših in večjih avtomobilih).

V obeh primerih se je izkazal tudi vpliv proizvajalca, pri napovedovanju verjetnosti škodnega dogodka so bili to psevdo-atributi za proizvajalce Citroën, Ford, Opel in Volkswagen. Zanimivo je morda še, da se nikjer ni pojavila barva vozila (razen razlika med kovinsko in navadno barvo pri napovedovanju nastopa škodnega dogodka).

Vseh ostalih atributov, ki so se pojavili v enem ali drugem primeru, ter možnih vzrokov ne bom komentiral; če bi želel analizirati vse podrobnosti glede izbranih atributov, bi po mojem mnenju bila potrebna natančnejša raziskava, kar pa ni bil namen mojega dela; komentar izbranih

atributov v tem poglavju podajam bolj kot stranski produkt. V primeru razširitve raziskave na to področje bi po mojem mnenju bilo najprej potrebno natančneje določiti optimalni nabor atributov, morda tudi na različnih učnih množicah (iz različnih obdobij). Nadalje pa bi bilo pri nekaterih atributih smiselno raziskati tudi smer vpliva; samo dejstvo, da je bil v izbor vključen psevdo-atribut za proizvajalca Opel ali pa atribut za kovinsko barvo vozila, nam namreč nič ne pove o tem, ali so takšna vozila bolj ali manj tvegana.

4.2.2.3 Primerjava rezultatov

V tem delu podajam pregled rezultatov, ki sem jih dobil z izbranima algoritmoma strojnega učenja na različnih množicah podatkov ter z različnimi nastavitvami pri napovedovanju višine odškodnine (Tabela 6). Za mero sem vzel vrednost MSE, pri kateri nižje vrednosti pomenijo bolj natančen model. Na učnih podatkih sem uporabil 4-kratno prečno vrednotenje. V tej primerjavi so upoštevani samo primeri s škodami, razširitev na celoten nabor podatkov prikazujem v podpoglavju 4.2.3.

Tabela 6: Primerjava rezultatov modelov, dobljenih pri različnih algoritmih ter parametrih strojnega učenja, pri napovedovanju višine odškodnine (poudarjeni sta najboljši vrednosti)

Algoritem	**MSE za podatke 2006–2008, samo primeri s škodami (v EUR^2)**	**MSE za podatke testnega scenarija 2012, samo primeri s škodami (v EUR^2)**
Naključni gozd, 100 odločitvenih dreves, 32 atributov	5.501.724	6.375.734
Naključni gozd, 50 odločitvenih dreves, 32 atributov	5.592.459	6.608.240
RPROP NN, 25 skritih nevronov, 100 iteracij učenja, 32 atributov	4.629.953	5.974.455
RPROP NN, 20 skritih nevronov, 100 iteracij učenja, 32 atributov	**4.565.848**	**5.878.321**
RPROP NN, 20 skritih nevronov, 250 iteracij učenja, 32 atributov	5.190.885	6.290.884
RPROP NN, 12 skritih nevronov, 100 iteracij učenja, 32 atributov	5.081.323	6.877.181
RPROP NN, 12 skritih nevronov, 250 iteracij učenja, 32 atributov	6.143.568	7.436.917

Glavni namen pridobivanja tukaj predstavljenih rezultatov (Tabela 6) je bil ta, da so mi pomagali pri izbiri algoritmov in parametrov, ki sem jih uporabil v nadaljevanju. Na podlagi rezultatov (Tabela 6) sem se odločil, da v nadaljevanju uporabim RPROP nevronsko mrežo z 20 nevroni v skriti plasti ob 100 iteracijah učenja, saj mi je dala najnižjo vrednost MSE.

4.2.3 Združitev napovedi v pričakovano vrednost izplačil

V tem delu sem moral povezati prejšnji dve napovedi v končno napoved pričakovane vrednosti izplačil; le-to lahko vzamemo za mero tveganosti zavarovanca, predstavlja pa tudi nevarnostno premijo.

Najbolj naravno se zdi, da za posamezen primer enostavno pomnožimo napoved verjetnosti nastopa škodnega dogodka (podpoglavje 4.2.1) z napovedjo višine odškodnine (podpoglavje 4.2.2). Vendar pa je treba upoštevati, da je bilo napovedovanje obeh omenjenih vrednosti optimizirano ločeno in da nikjer ni bila upoštevana optimizacija napovedi kot celote, na kar opozarjajo tudi Apte et al. (1998, str. 3). Zaradi tega sem tudi na tej stopnji uporabil strojno učenje. Napovedi sem najprej pomnožil, s čemer sem želel vključiti omenjeno naravno razlago obeh vrednosti; rezultat sem zatem še optimiziral glede na dejanske vrednosti odškodnin, in sicer sem za to uporabil kar RPROP nevronsko mrežo z enim nevronom v skriti plasti. Učenje nevronske mreže poteka tako, da se poskuša minimizirati MSE (podpoglavje 2.2.2.2), kar je po ugotovitvah Dugasa et al. (2003) primerno za problem ocenjevanja nevarnostne premije. Prednost uporabe takšne metode pred preprosto linearno regresijo je bila tudi v tem, da ni vračala negativnih rezultatov, kateri, upoštevaje pomen napovedovane vrednosti, seveda niso zaželeni.
Vhodne primere v ta del raziskave je predstavljala validacijska množica velikosti 25 % nabora polic iz let 2006–2008, na kateri sem pred tem izvedel napovedi verjetnosti škodnega dogodka ter višine odškodnine. Pri tem sem zagotovil, da so se algoritmi, ki so opravili ti dve napovedi, učili le na preostalih 75 % podatkov, noben primer iz validacijske množice torej ni bil uporabljen pri učenju. Pri združevanju rezultatov sem te vhodne podatke še dalje razdelil v razmerju 50 % proti 50 % na učni del, na katerem sem učil nevronsko mrežo iz tega koraka, ter preostali validacijski del; v vsakem delu je bilo približno 17.000 primerov. Takšno razmerje sem izbral zato, da se validacijska množica ne bi preveč zmanjšala, hkrati pa sem imel samo en vhodni atribut (produkt prejšnjih dveh napovedi), zato se mi ni zdelo potrebno, da je v učni množici veliko število primerov. Izgrajeni model nevronske mreže sem nato uporabil tudi pri napovedovanju končne pričakovane vrednosti odškodnine na realnem testnem scenariju, ki so ga predstavljale police iz leta 2012 (približno 45.000 primerov).

Omeniti velja še eno podrobnost, in sicer sem se, tako pri učenju in validaciji kot pri realnem testnem scenariju, omejil na police, ki imajo trajanje 1 leto. S stališča strojnega učenja je razlog v tem, da nisem pričakoval, da bodo lahko modeli, ki morajo upoštevati številne atribute, enako dobro zajeli trajanje kritja, kot to lahko storimo z uporabo pro-rata premije. Praktičen razlog pa je, da se zavarovanje AK (razen redkih izjem) sklepa skoraj izključno za obdobje 1 leta; police, ki imajo krajše trajanje, so v največji meri posledica predčasne prekinitve pogodbe, na primer zaradi prodaje vozila. Iz tega sledi, da bi bil moj model v praktični uporabi torej v veliki večini primerov uporabljen za napovedovanje kritja za obdobje 1 leta, zato menim, da s to omejitvijo nisem naredil napake in da je to realna predpostavka za ovrednotenje rezultatov.

V prejšnjih dveh podpoglavjih sem ovrednotil rezultate več različnih algoritmov ter njihovih parametrov. Podajanje združenih rezultatov za vse možne kombinacije se mi na tem mestu zaradi preglednosti ne zdi smiselno, zato se bom omejil na kombinacijo napovedi, ki sem ju dobil z uporabo nevronske mreže. Pri prvi vrednosti, napovedi nastopa škodnega primera, sem uporabil RPROP nevronsko mrežo z 21 nevroni v skriti plasti, učenje modela je trajalo 100 iteracij. Pri napovedovanju višine odškodnine ob nastopu škodnega primera pa sem uporabil model RPROP nevronske mreže z 20 skritimi nevroni, ki je bil prav tako naučen v 100 iteracijah. Ta modela sem izbral zato, ker sta na validacijski množici izkazala najboljše rezultate (rezultatov na

primerih iz leta 2012 za izbiro nisem smel uporabiti, saj s tem moj testni scenarij ne bi bil več realen), v prid temu pa govorijo tudi sorodne raziskave (na primer Dugas et al., 2003). Tukaj navedeni parametri izbranih nevronskih mrež veljajo za vse v nadaljevanju prikazane rezultate, zato jih ne bom vsakič posebej navajal.

4.2.3.1 Nevarnostna premija za primerjavo

Za ovrednotenje mojih rezultatov se mi je zdelo skoraj nujno, da jih primerjam tudi z nevarnostno premijo, ki je bila dobljena z aktuarskim izračunom. Težava pri tem je bila, da mi ta nevarnostna premija ni bila znana. Za svoje primere sem imel na razpolago samo podatke iz podatkovne baze, kjer ta vrednost ni navedena, ampak je zapisana samo kosmata premija, ki jo je plačal zavarovanec. Sestavo te premije in povezavo z nevarnostno premijo sem opisal že v podpoglavju 1.1, vendar pa deležev oziroma vrednosti posameznih delov nisem poznal, tako da je nisem mogel preprosto izračunati, ampak sem moral najti način, da jo vsaj približno ocenim.

Dodatno težavo pri tej oceni so predstavljali tudi variabilni komercialni popusti. Če bi nevarnostno premijo ocenil iz končne premije, vključno s komercialnimi popusti, ki je bila plačana za nek primer, bi ti popusti zameglili osnovno nevarnostno premijo oziroma oceno tveganosti, ki jo izraža. Slednje bom pojasnil na primeru. Imejmo dva zavarovanca z enakimi lastnostmi, za katera je bila posledično tudi izračunana nevarnostna premija po veljavnem aktuarskem modelu enaka. Če bi eden od njiju pri sklenitvi zavarovanja dobil dodaten komercialni popust in plačal nižjo premijo ter bi nevarnostno premijo za oba zavarovanca po enakem postopku (in je ta postopek monotona funkcija, kar se mi v tem primeru zdi smiselno) ocenili iz te končne plačane premije, bi bil ta ocena v primeru zavarovanca, ki je dobil dodaten komercialni popust, nižja, kar pa ne ustreza izhodiščnemu stanju (zavarovanca z enakimi lastnostmi ter enako nevarnostno premijo).

Zaradi navedenega sem se odločil, da kot izhodišče za oceno nevarnostne premije po aktuarskem modelu uporabim kosmato premijo brez upoštevanja variabilnih komercialnih popustov, označimo jo s *k*. To premijo sem imel v pridobljenih podatkih že izračunano, saj sem na ta problem pomislil že tekom priprave podatkov.

Vendar pa sem bil mnenja, da mora tudi aktuarski model variabilne komercialne popuste nekako upoštevati, saj je znano, da obstajajo (seveda jih ne more upoštevati na ravni posameznega zavarovanca); le-ti so tako predstavljali še eno dodatno sestavino razlike med uporabljeno kosmato premijo *k* ter nevarnostno premijo. Sklenil sem, da nimam podatkov, na podlagi katerih bi lahko posamezne sestavine omenjene razlike ustrezno ocenil, zato sem se odločil za preprostejši pristop, in sicer sem te sestavine strnil v en sam faktor α, katerega sem definiral kot vrednost, ki mi bo dala takšno oceno nevarnostne premije αk, da bo MSE med to oceno in dejanskim zneskom odškodnin minimalna. To oceno sem opravil ločeno za validacijsko (police iz let 2006–2008) ter testno množico (police iz leta 2012).

Minimizacijo vrednosti MSE glede na faktor α sem izvedel z uporabo funkcije »optim« iz orodja R, ki sem ga poklical kar iz orodja KNIME z uporabo vozlišča R Snippet. Dobljeni vrednosti α sta bili 0,639 za obdobje 2006–2008 ter 0,552 za obdobje 2012. To pomeni, da po opravljeni oceni nevarnostna premija predstavlja 63,9 % oziroma 55,2 % kosmate premije brez komercialnih popustov *k*. Razlog za to kar občutno razliko med obravnavanima obdobjema pojasnjujem v podpoglavju 4.3.1.

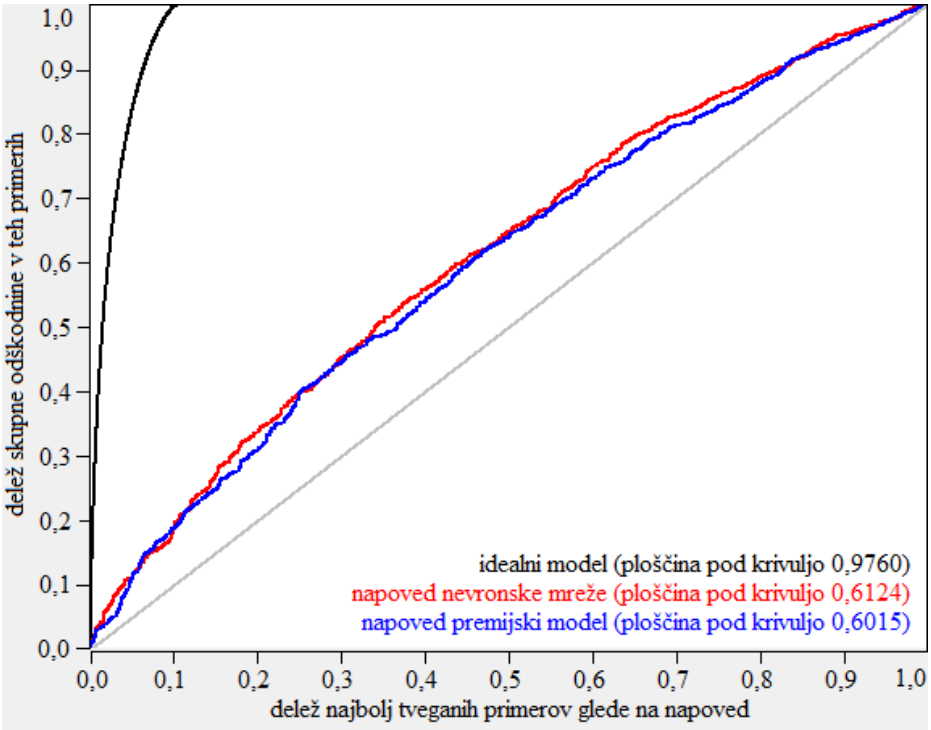
Rezultat sem tudi primerjal z vsebino letnih poročil izbrane zavarovalnice, in sicer z objavljenimi podatki o stroških poslovanja v odstotku od kosmate obračunane zavarovalne premije pri zavarovanju kopenskih motornih vozil (razen tirnih) (3. zavarovalna vrsta po 7. členu ZZavar-1); le-ti so v posameznih letih seveda različni in se gibljejo med 21 % in 27 %, povprečje pa je okoli 25 % (preveril sem letna poročila izbrane zavarovalnice za leta 2000–2015). S tem podatkom sicer ne morem v celoti ovrednotiti svoje ocene nevarnostne premije (razlika med njo in kosmato premijo seveda niso samo stroški poslovanja), lahko pa ugotovim, da si podatki in moja ocena vsaj ne nasprotujejo.

Vprašal sem se tudi, ali je primerjava s tako ocenjeno nevarnostno premijo smiselna in poštena. Glede na to, da je eden od kriterijev za primerjavo rezultatov vrednost MSE, je približek, ki minimizira MSE, prednost za aktuarski model in dejanska nevarnostna premija lahko ima tudi večjo vrednost MSE. Po drugi strani pa razlike med kosmato in nevarnostno premijo v realnosti najbrž ne moremo preprosto strniti v en sam faktor, ampak je le-ta različna za različne skupine primerov in je preslikava med eno in drugo premijo bolj zapletena funkcija; če bi torej to razliko tudi sam zmodeliral z bolj fleksibilno funkcijo z več parametri, bi lahko z minimizacijo dobil nevarnostno premijo z nižjo MSE. Na podlagi zapisanega menim, da primerjava mojih rezultatov s tako ocenjeno nevarnostno premijo ni nesmiselna, je pa treba imeti v mislih, kako je bila ta nevarnostna premija pridobljena ter da nujno ne ustreza dejanski vrednosti.

4.2.3.2 Grafična predstavitev rezultatov

Rezultate sem si najprej ogledal z uporabo **grafov pridobitve** (angl. *gain chart, enrichment plot, lift chart*) (Slika 9, Slika 10). Takšna predstavitev je bila v sorodnih raziskavah že večkrat uporabljena (Apte et al., 1999, str. 55; Dal Pozzolo, 2010, str. 39; Kolyshkina & Brookes, 2002, str. 10). Pri tem posamezne primere razvrstimo glede na neko vrednost, v mojem primeru je to bila napovedana pričakovana višina odškodnine, razvrstitev pa je bila padajoča. Ta razvrstitev predstavlja vrstni red primerov na x osi grafa; primeri z največjo napovedano vrednostjo odškodnine so torej na levi strani. Vrednosti na y osi pa predstavljajo dejansko kumulativno vsoto izplačanih odškodnin za vse primere do vključno opazovanega (torej odškodnina opazovanega primera plus vsota odškodnin za vse primere, ki so na x osi levo od njega); ta znesek ni predstavljen v absolutni vrednosti, ampak kot delež od vsote vseh dejansko izplačanih odškodnin. Vrednost za primer, ki je na x osi na meji 10 %, torej pomeni, kolikšen delež dejanske skupne odškodnine je zajet v 10% najbolj tveganih primerov glede na razvrstitev modela (Dal Pozzolo, 2010, str. 40).

Slika 9: Graf pridobitve za rezultate nevronske mreže in primerjava s premijskim modelom za police, sklenjene v obdobju 2006–2008 (validacijska množica)



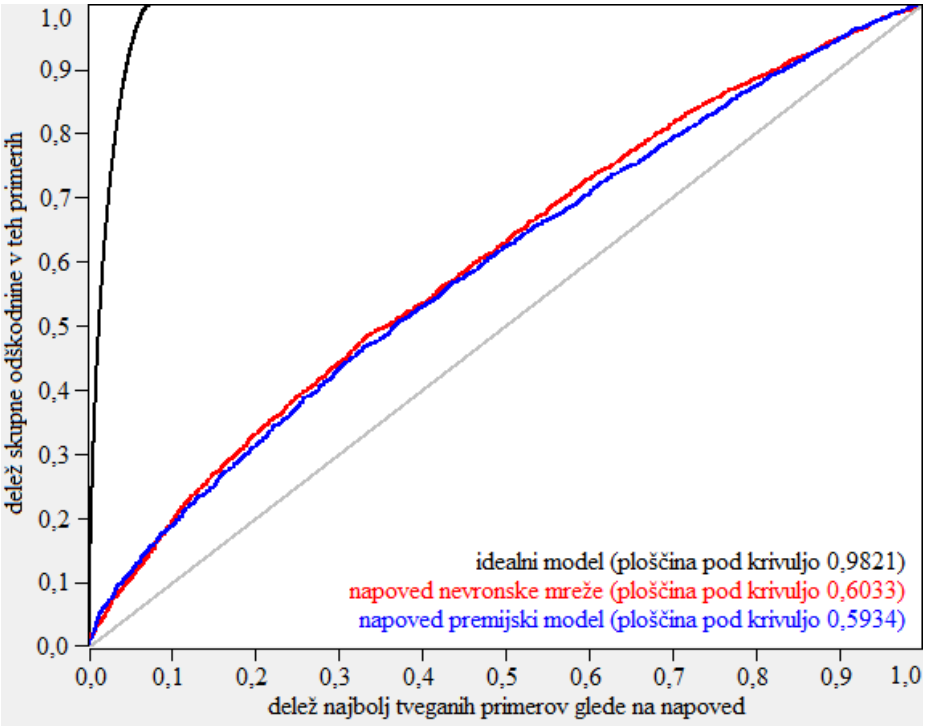
Takšna predstavitev je podobna ROC krivulji, vendar pa so tudi razlike. Pri ROC krivulji za idealni model dobimo kvadrat, vrednost AUC pa je enaka 1 (podpoglavje 2.2.3.1); pri tej predstavitvi takšnega rezultata ne moremo dobiti, saj se za vsak naslednji obravnavan primer na x osi pomaknemo desno. Za ilustracijo te razlike in za primerjavo sem zato na omenjene grafe pridobitve vedno dodal tudi krivuljo, ki bi jo dal idealni model (v tem primeru so primeri na x osi urejeni glede na dejansko vrednost odškodnine). Diagonala na grafu tudi v tem primeru, podobno kot pri ROC krivulji, predstavlja pričakovano krivuljo, ki bi jo dobili, če bi model primere razvrščal povsem naključno (Dal Pozzolo, 2010, str. 40).

Prestavitev lahko v numerično vrednost strnemo tako, da izračunamo ploščino pod krivuljo, pri čemer večja ploščina načeloma pomeni boljši model. Moramo pa biti previdni, če primerjamo to vrednost pri rezultatih, ki smo jih dobili na različnih množicah podatkov, saj največja možna vrednost, ki jo predstavlja krivulja idealnega modela, ni vedno enaka; za takšne primerjave moramo zato dobljene ploščine normalizirati glede na vrednost, ki bi jo dobili z idealnim modelom (Dal Pozzolo, 2010, str. 40).

Slika 9 prikazuje graf pridobitve za police validacijske množice, torej police, ki so bile sklenjene v letih 2006–2008 in niso bile uporabljene pri učenju. Na graf sem za primerjavo dodal tudi rezultate, ki jih predstavlja **premijski model**, primeri so za to krivuljo na x osi torej urejeni glede na premijo oziroma premija predstavlja pričakovano vrednost odškodnine; pri tem sem upošteval nevarnostno premijo, ki sem jo ocenil po postopku, ki sem ga opisal v podpoglavju 4.2.3.1. Ker je bila ocena opravljena samo z zmanjšanjem kosmate premije brez komercialnih popustov za

določen faktor, to na graf pridobitve nima vpliva in bi z uporabo kosmate premije brez komercialnih popustov dobili enako krivuljo (skaliranje namreč ne spremeni vrstnega reda). Dobljeni graf (Slika 9) nakazuje, da je model z nevronsko mrežo v splošnem nekoliko bolje razvrstil primere po tveganosti kot premijski model, vendar pa je razlika zelo majhna. Podobno opazimo tudi na grafu, ki prikazuje rezultate na testnem scenariju, torej policah, ki so bile sklenjene v letu 2012 (Slika 10); tukaj bi opozoril na levi spodnji kot grafa, ki kaže, da je premijski model nekoliko bolje razpoznal najbolj tvegane primere. Pri legendi krivulj na grafu so v oklepaju podane tudi ploščine pod krivuljo, ki pa niso normalizirane glede na idealni model (v nasprotnem primeru bi imel idealni model vrednost 1).

Slika 10: Graf pridobitve za rezultate nevronske mreže in primerjava s premijskim modelom za police, sklenjene v letu 2012 (realni testni scenarij)



Pri razlagi rezultatov moram opozoriti še na eno pomembno omejitev grafa pridobitve. Za ta graf je pomembno samo, kako neka ocena razvrsti primere, dejanske absolutne vrednosti zneskov pa so nepomembne (podobno lastnost smo srečali že pri ROC krivulji). Če dobljeno napoved pričakovane višine odškodnine pomnožimo ali delimo s poljubnim faktorjem (s čemer lahko dobimo čisto nesmiselne vrednosti, na primer pričakovano vrednost odškodnine v višini 10.000.000 EUR), bomo dobili enako krivuljo na grafu pridobitve. Zaradi tega je pomembno, da napovedi modela ovrednotimo še na druge načine ter si jih ogledamo tudi v absolutnih vrednostih, te rezultate podajam v nadaljevanju.

4.2.3.3 Numerično ovrednotenje učinka uporabe modelov ter diskusija

Primerjavo rezultatov napovedi nevronske mreže in premijskega modela, pri katerem sem kot napoved pričakovane vrednosti odškodnine upošteval nevarnostno premijo, ki sem jo ocenil kot opisano v podpoglavju 4.2.3.1, podajam v tabeli Tabela 7. Ta primerjava prikazuje skupen učinek uporabe posameznega modela preko vseh zavarovanj posameznega nabora podatkov, na katerem sem model preizkusil, primerjavo po podmnožicah pa podajam v podpoglavju 4.2.3.4.

Rezultati kažejo, da se je glede na kriterij MSE (prva vrstica s podatki v tabeli Tabela 7) nevronska mreža na validacijski množici (police iz let 2006–2008) bolje obnesla, pri policah iz leta 2012 pa je boljšo vrednost MSE dosegel premijski model. Opazna pa je tudi velika razlika v vrednosti MSE med validacijsko množico in realnim testnim scenarijem; to razliko sem raziskal in ugotovil, da je za njo odgovoren en škodni primer z zelo visokim zneskom izplačane odškodnine. Odškodnina je v tem primeru približno 5,5 krat večja od druge največje odškodnine v tem naboru podatkov (police iz leta 2012), hkrati pa gre tudi za največjo odškodnino celotnega nabora podatkov, ki sem ga pridobil v koraku priprave podatkov.

Tabela 7: Primerjava natančnosti nevronske mreže (NN) in premijskega modela pri določanju pričakovane vrednosti izplačil preko vseh zavarovanj posameznega nabora podatkov glede na različne kriterije z označenimi boljšimi vrednostmi

Nabor podatkov	**Police 2006–2008 (validacija)**		**Police 2012 (testni scenarij)**		**Police 2012 (brez osamelca)**	
Mera \ Model	**Premijski**	**NN**	**Premijski**	**NN**	**Premijski**	**NN**
Povprečna kvadratna napaka (MSE) (v EUR^2)	535.057	**531.474**	**750.275**	762.538	451.960	**450.042**
Povprečna absolutna napaka (MAE) (v EUR)	255,920	**255,093**	243,845	**238,675**	241,287	**236,056**
Povprečna predznačena razlika (MSD) (v EUR)	-4,127	**-3,087**	20,217	**15,190**	22,781	**17,815**
$\sum napoved / \sum škode$ (v %)	97,15	**97,87**	116,56	**112,44**	119,07	**114,91**

Rezultate sem zato primerjal še enkrat, vendar sem ta ekstremni primer tokrat izločil (zadnja dva stolpca v tabeli Tabela 7). Vrednosti MSE so bile tukaj bolj primerljive s tistimi, ki sem jih dobil na validacijski množici, obenem pa se je tukaj nevronska mreža spet izkazala za nekoliko uspešnejšo. Naj poudarim, da s svojim pristopom ne želim namigovati, da lahko takšne osamelce kar izločimo, če nam kvarijo rezultate, ne nazadnje so porazdelitve z debelimi repi značilne za odškodnine pri avtomobilskih zavarovanjih (Chapados, 2010, str. 8). Vendar se mi kljub temu zdi smiselno predstaviti tudi te rezultate, še zlasti zato, ker sem tukaj opazoval samo obdobje enega leta in imel to »smolo«, da se je ravno v tem obdobju pojavil omenjeni primer; če bi opazoval daljše časovno obdobje, bi se vpliv takšnega osamelca porazdelil in ne bi imel več tako velikega vpliva na ovrednotenje rezultatov modela (razen, če bi se v daljšem obdobju pojavilo še več takšnih primerov, ampak v tem slučaju potem več ne bi mogli govoriti o osamelcih). V

obzir pa je potrebno vzeti tudi to, da se AK zavarovanja za primere, kjer obstaja možnost tako visokih odškodnin (za zelo drage avtomobile torej), običajno sklepajo po posebnih pogojih in s posredovanjem ter odobritvijo strokovne službe, tako da ni niti pričakovano niti potrebno, da bi bile napovedi modela ustrezne za takšne primere. Nazadnje pa se mi ta primer zdi tudi dober opomnik, kako velik vpliv lahko ima tak osamelec ter da je treba rezultate zato vedno še dodatno analizirati.

Kot alternativo za izločitev vpliva osamelcev lahko uporabimo tudi povprečno absolutno napako MAE, ki jo predstavljam v drugi vrstici s podatki v tabeli Tabela 7. Ker ta mera ne kvadrira razlike med napovedanimi in dejanskimi vrednostmi, je manj občutljiva na vpliv osamelcev (Palfy, 2009, str. 19), kar je razvidno tudi iz zgornjih rezultatov (model nevronske mreže nam da nekoliko boljši rezultat tudi, če osamelca ne izločimo). Pozoren je potrebno biti tudi na to, da je, za razliko od MSE, vrednost MAE v enakem velikostnem razredu (in enoti) kot primerjane vrednosti.

Zraven vrednosti MSE me je zanimala tudi primerjava vsote nevarnostne premije (napovedane z mojim modelom ter ocenjene iz kosmate premije) in vsote vseh odškodnin, katera v eni številki (razmerje obeh vrednosti) predstavi, koliko so napovedi kumulativno gledano blizu realnosti (zadnja vrstica v tabeli Tabela 7). Seveda ta kumulativni pogled nič ne pove o tem, kako natančno je bila nevarnostna premija razporejena na posamezne primere (ta številka bi bila enaka tudi, če bi na vse primere razporedili enak znesek ustrezne višine), vendar pa lahko to drugo informacijo delno razberemo iz grafov pridobitve, ki sem jih že predstavil (Slika 9, Slika 10); rezultate je torej potrebno gledati povezano. Iz rezultatov v tabeli Tabela 7 je razvidno, da je bila vsota napovedi nevronske mreže v vseh primerih nekoliko bližje dejanski vsoti odškodnin (bližje vrednosti 100 % v tabeli torej) kot pa vsota nevarnostne premije iz premijskega modela; za oba modela pa je značilno, da sta na validacijski množici »pobrala« nekoliko premalo premije za pokritje vseh škod, ravno obratno pa velja za testni scenarij iz leta 2012.

Podoben pogled kot pravkar opisana vrednost nam da tudi **povprečna predznačena razlika** (angl. *mean signed difference*, v nadaljevanju **MSD**), ki je definirana kot $^1/_n \sum(p_i - r_i)$ (oznake v formuli so enake kot pri enačbi (9)) in jo podajam v predzadnji vrstici tabele Tabela 7. Vrednost -3,087 EUR pri rezultatih nevronske mreže na validacijski množici na primer pomeni, da smo v povprečju na posamezen primer pobrali dobre 3 EUR premalo nevarnostne premije, da bi lahko na koncu pokrili celotno vsoto vseh izplačanih odškodnin.

Rezultati modela, ki sem ga dobil z uporabo nevronskih mrež, so bili na pogled sicer za odtenek boljši (razen po kriteriju MSE za police 2012 brez odstranitve osamelca), vendar pa razlike niso bile velike, zaradi česar sem bil mnenja, da so razlike lahko tudi samo plod naključja. To sem se odločil preveriti, in sicer sem testiral ničelno hipotezo $H_0: MSE_{premija} - MSE_{NN} = 0$ proti alternativni hipotezi $H_a: MSE_{premija} - MSE_{NN} \neq 0$. Uporabil sem parni t-test, kot ga opisujeta Devore in Berk (2012, str. 511). Najprej sem definiral razliko kvadratnih napak

$$d_i = (p_{premija\,i} - r_i)^2 - (p_{NN\,i} - r_i)^2 \tag{11}$$

pri čemer sta $p_{premija\,i}$ in $p_{NN\,i}$ napovedi premijskega modela ter modela nevronske mreže na i-tem primeru, r_i pa dejanska vrednost izplačanih odškodnin na istem primeru. t-vrednost za testiranje hipoteze v tem primeru dobimo po enačbi

$$t = \sqrt{n}\frac{\bar{d}}{s_D} = \sqrt{n}\frac{MSE_{premija} - MSE_{NN}}{s_D} \tag{12}$$

pri čemer je n velikost vzorca, $\bar{d}$ vzorčno povprečje in s_D vzorčni standardni odklon vrednosti d_i (Devore & Berk, 2012, str. 511). Velja torej $\bar{d} = \frac{1}{n}\sum d_i$ in $s_D = \sqrt{(1/n-1)\ \sum(d_i - \bar{d})^2}$ (Devore & Berk, 2012, str. 25, 33).

Ker je bila moja alternativna hipoteza, da MSE vrednosti nista enaki, in ni zajemala, katera od njiju je boljša (sicer bi imeli na primer $H_a: MSE_{premija} - MSE_{NN} > 0$), sem uporabil dvostranski (angl. *two-tailed*) test (Devore & Berk, 2012, str. 462). Iz t-vrednosti sem določil P-vrednosti, kot opisujeta Devore in Berk (2012, str. 462), pri čemer sem ploščine pod krivuljo (t porazdelitev z $n-1$ prostostnimi stopnjami) dobil iz orodja KNIME (vozlišče Paired t-test), rezultate pa prikazujem v tabeli Tabela 8; ob tem naj omenim, da je bil n v vseh mojih primerih zelo velik (nad 10.000), kar pomeni, da je razlika med t porazdelitvijo z $n-1$ prostostnimi stopnjami in standardno normalno porazdelitvijo zanemarljiva in bi lahko za določanje P-vrednosti uporabil tudi slednjo (z-test) (Kononenko, 2005, str. 100).

Tabela 8: t-vrednosti in P-vrednosti za testiranje hipoteze H_o: $MSE_{premija}$-MSE_{NN}=0 proti H_a: $MSE_{premija}$-MSE_{NN}≠0

Vrednost \ Nabor podatkov	**Police 2006–2008 (validacija)**	**Police 2012 (testni scenarij)**	**Police 2012 (brez osamelca)**
t-vrednost	1,8834	-0,8619	1,6888
P-vrednost	0,0597	0,3887	0,0913

Če za stopnjo značilnosti α izberemo vrednost 0,05 (5 %), na podlagi rezultatov v tabeli Tabela 8 ugotovimo, da ničelne hipoteze pri tej stopnji značilnosti ne moremo zavrniti. Ničelno hipotezo namreč lahko zavrnemo, če je P-vrednost manjša ali enaka izbrani stopnji značilnosti α (Devore & Berk, 2012, str. 457). Na tej podlagi zaključujem, da – glede na kriterij MSE preko vseh zavarovanj iz posameznega testnega nabora podatkov – nobeden od primerjanih modelov ni statistično značilno boljši od drugega in da so razlike v vrednostih MSE verjetneje posledica naključja.

Na te mestu bi se zato dotaknil raziskave Dugasa et al. (2003), v kateri je avtorjem pri napovedovanju nevarnostne premije z uporabo nevronskih mrež uspelo doseči bistveno boljše rezultate v primerjavi s premijskim modelom (seveda tistim, ki je veljal za njihov nabor podatkov) (Dugas et al., 2003, str. 199). Pod omenjeno raziskavo se je podpisalo 6 avtorjev in je

trajala 1 leto, naročnik pa je bila velika severnoameriška zavarovalnica (Dugas et al., 2003, str. 180). V njej je bila uporabljena posebna arhitektura nevronske mreže, ki je najboljše rezultate dala združena v mešani model (angl. *mixture model*), kot so ga predlagali Jacobs, Jordan, Nowlan in Hinton (1991) (Dugas et al., 2003, str. 192–195). Takšna arhitektura mi ni bila na voljo za neposredno uporabo, njena implementacija pa bi žal presegala okvir moje raziskave. To pa ni edina razlika, saj iz podatkov o raziskavi sklepam, da so lahko avtorji vložili bistveno več časa kot jaz tudi v optimizacijo ostalih dejavnikov, na primer v pripravo podatkov in izbiro atributov. Na podlagi zapisanega se mi ne zdi presenetljivo, da sam nisem uspel doseči tako dobrih rezultatov v primerjavi s premijskim modelom. Pri tem je po mojem mnenju potrebno upoštevati tudi, da omenjena raziskava izvira iz leta 2003 ter da so bili nekateri dejavniki, na podlagi katerih je njihov model dosegel boljše rezultate od premijskega, morda že zajeti v premijskem modelu, s katerim sem primerjal svoj model, saj je smiselno sklepati, da zavarovalnice težijo k nenehnemu izboljševanju modelov. Nazadnje pa seveda ni zanemarljivo tudi, da je šlo za drug nabor podatkov in da nimamo podlage, na kateri bi lahko trdili, da bi se model Dugasa et al. (2003) tudi na mojem naboru podatkov odrezal statistično značilno bolje od premijskega.

Razlika med mojim pristopom in pristopom Dugasa et al. (2003) je bila tudi v tem, da sam nevarnostne premije nisem napovedoval neposredno, ampak v dveh korakih (preko napovedovanja nastopa škodnega dogodka in višine odškodnine v primeru škodnega dogodka), kar Apte et al. (1998, str. 3) odsvetujejo. Iz tega razloga sem se odločil, da tudi sam preizkusim neposredno napovedovanje; uporabil sem RPROP nevronsko mrežo in nekaj različnih podmnožic atributov, ki sem jih sestavil na osnovi unije izbranih atributov v obeh korakih posrednega napovedovanja (podpoglavji 4.2.1.3 in 4.2.2.1), preizkus pa sem opravil tudi z vsemi atributi. Rezultatov tukaj ne bom posebej predstavljal, naj pa omenim, da mi ni uspelo doseči izboljšanja vrednosti MSE v primerjavi z napovedovanjem nevarnostne premije v dveh korakih, ampak so bile vrednosti MSE približno 2 krat višje. Pri tem moram poudariti, da sem uporabil običajno RPROP nevronsko mrežo in opravil le nekaj preizkusov (različne množice atributov, različno število nevronov v skriti plasti) ter da bi morda lahko z večjim vložkom v optimizacijo, na primer z uporabo kakšne kombinacije modelov (zanimivo bi mogoče bilo preizkusiti v podpoglavju 2.2.2.4 omenjeni boosting), dosegel boljše rezultate.

Omenil bi še raziskavo, ki so jo opravili Smith et al. (2000). Njeni avtorji menijo, da direktni pristop k napovedovanju pričakovanih odškodnin (na ravni posameznega zavarovanca) ni primeren, zato so v ta namen uporabili nenadzorovano strojno učenje z grupiranjem, kar njihov pristop približuje tradicionalnim metodam (Smith et al., 2000, str. 538). Glede na rezultate svojega dela ter na rezultate sorodnih raziskav, s katerimi sem se srečal, menim, da tudi v direktnem napovedovanju obstaja potencial za dobre rezultate, zato ga ne bi odpisal že od začetka. Bi pa bilo zanimivo na istem naboru podatkov preveriti še podoben pristop, kot so ga uporabili Smith et al. (2000), in primerjati rezultate, kar predstavlja možnost za nadgradnjo moje raziskave.

4.2.3.4 Primerjava glede na podmnožice zavarovancev ter možnost uporabe v praksi

Kot sem zapisal že v uvodnem poglavju, ni pomemben zgolj učinek modela na celotnem naboru zavarovalnih pogodb, ampak je pomembno tudi, da je model **pošten** (angl. *fair*) do posameznih podmnožic zavarovancev, kar pomeni, da naj bi bila povprečna nevarnostna premija znotraj določene podmnožice približno enaka povprečni odškodnini (Dugas et al., 2003, str. 197). Za namene takšne primerjave lahko omenjene podmnožice izberemo glede na različne posamezne atribute (na primer spol, starost, kraj prebivališča, …) ali pa tudi glede na kombinacije več atributov, kar pa nas lahko v skrajnem primeru pripelje na raven posameznih zavarovalnih pogodb, na kateri primerjava ni več smiselna (Dugas et al., 2003, str. 197–198).

Sam sem se odločil, da primerjavo izvedem na 10 (približno) enako velikih podmnožicah zavarovalnih pogodb, ki sem jih določil glede na **starost** zavarovanca ob sklenitvi zavarovalne pogodbe (podmnožice sem dobil z uporabo KNIME vozlišča Auto-Binner). Starost sem izbral zato, ker se je pojavila med izbranimi atributi tako pri modelu za napovedovanje nastopa škodnega primera (podpoglavje 4.2.1.3) kot tudi pri modelu za napovedovanje višine odškodnine ob škodnem primeru (podpoglavje 4.2.2.2), hkrati pa se ne upošteva v izračunu po premijskem modelu zavarovalnice, s katerim sem primerjal svoj model.

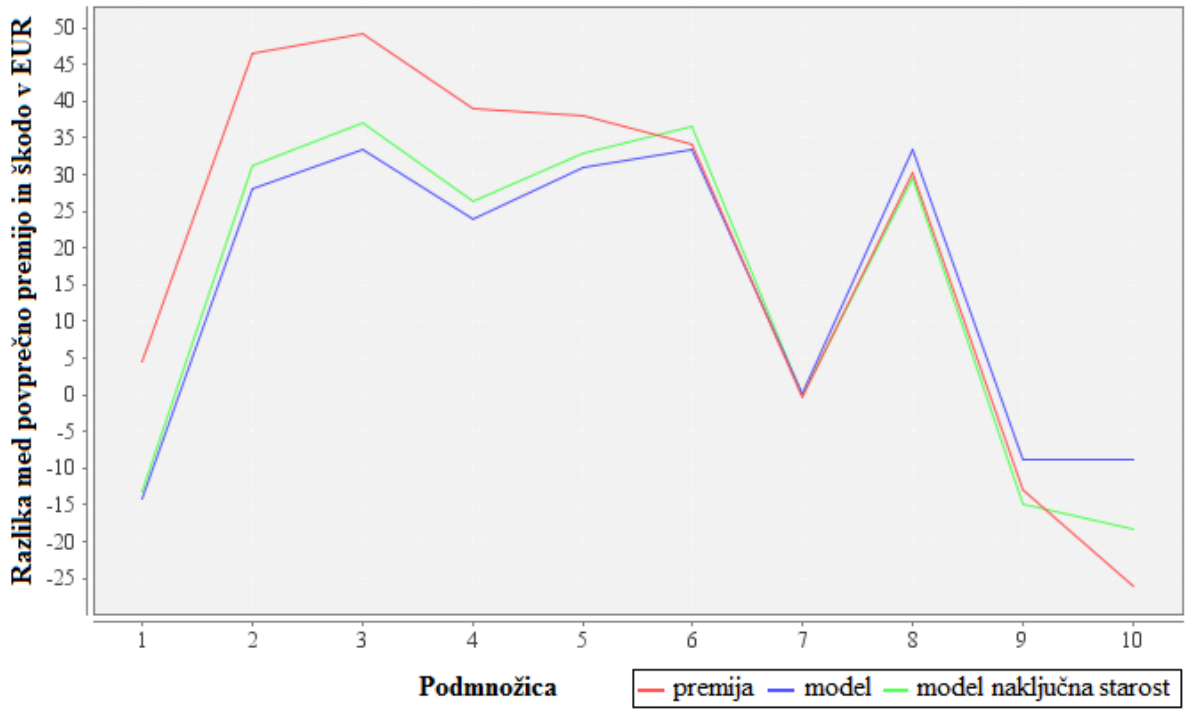
Primerjavo sem opravil na policah iz leta 2012 (realni testni scenarij), pri čemer osamelca nisem izločil. Rezultate primerjave sem grafično predstavil in jih prikazuje Slika 11. Izbrane podmnožice zavarovancev glede na starost (x os na sliki Slika 11) so označene s številkami od 1 do 10; pri tem recimo podmnožica 1 zajema 10 % najmlajših zavarovancev, podmnožica 2 zavarovance, ki so v skupini 20 % najmlajših zavarovancev, vendar niso med 10 % najmlajših (torej niso v podmnožici 1), podmnožica 10 pa 10 % najstarejših zavarovancev. Na y osi grafa (Slika 11) je prikazana razlika med povprečno premijo in odškodnino za določeno podmnožico, pri čemer vrednosti, ki so bližje 0, pomenijo bolj pošteno določeno nevarnostno premijo za neko podmnožico. Poudariti velja, da so funkcije na sliki Slika 11 diskretne ter definirane samo na ravni določene podmnožice ter sem jih v obliki krivulj, ki z linearno interpolacijo povezujejo točke na grafu, predstavil izključno zaradi večje preglednosti.

Rezultatom premijskega modela (rdeča črta na sliki Slika 11) ter svojega modela (modra črta na sliki Slika 11) sem za primerjavo dodal še rezultate, ki sem jih dobil s svojim modelom, če sem pred tem naključno premešal starosti zavarovancev (zelena črta na sliki Slika 11). Na ta način sem želel preveriti, ali je morebitna prednost mojega modela pred premijskim modelom v resnici odvisna od uporabe starosti zavarovanca pri izračunu nevarnostne premije; da bi bili ti rezultati čim bolj realni in neodvisni od »sreče« pri naključnem premetavanju starosti zavarovancev, sem napovedovanje ponovil 10 krat, vsakič z drugače naključno premešanimi starostmi zavarovancev, ter uporabil povprečje rezultatov.

Ker se pri ovrednotenju rezultatov nisem želel zanašati zgolj na vizualno oceno (večje razlike pri podmnožicah, kjer je premijski model slabši od mojega), sem izračunal še povprečno absolutno napako (MAE) preko vseh podmnožic ter dobil vrednosti 21,54 EUR (moj model),

23,98 EUR (moj model z naključno premešanimi starostmi) ter 28,06 EUR (premijski model), ki so potrdile mojo oceno uspešnosti modelov na podlagi slike Slika 11; mimogrede naj omenim, da mi je tudi izračun povprečne kvadratne napake (MSE) dal enak vrstni red.

Slika 11: Razlika med povprečno premijo in povprečno odškodnino za police, sklenjene v letu 2012, za različne modele po posameznih podmnožicah zavarovancev glede na starost



Opisana primerjava nakazuje možnost uporabe mojih modelov oziroma na splošno modelov, ki jih dobimo s podatkovnim rudarjenjem s strojnim učenjem, v praksi. V kombinaciji s tukaj predstavljenim oziroma kakšnim drugim podobnim postopkom jih lahko uporabimo za odkrivanje podmnožic zavarovancev, pri katerih bi se dalo izboljšati poštenost modela, ki ga zavarovalnica uporablja (odpraviti pristranskost), nadalje pa tudi za odkrivanje atributov, ki bi jih bilo v ta namen smiselno vključiti v uporabljan model.

Izboljšave v smeri bolj poštenega določanja premij posameznim podmnožicam zavarovancev prinašajo prednosti tako zavarovalnici kot zavarovancem. Pri slednjih bodo imeli korist pripadniki tistih podmnožic, ki jim model zavarovalnice zaračunava previsoke premije (v mojem konkretnem primeru bi to bile zlasti podmnožice 2, 3, 4 in 5) ter bi ob uporabi izboljšanega modela plačali manj. Po drugi strani bo zavarovalnica lahko z bolj poštenimi (nižjimi) premijami računala na prihod novih strank iz teh podmnožic ter posledično povečanje tržnega deleža in dobička (Dugas et al., 2003, str. 200). Povišanje premij za pripadnike podmnožic, ki jim model, ki ga uporablja zavarovalnica, zaračunava prenizke premije (podmnožici 9 in 10 v mojem konkretnem primeru), bo koristilo zavarovalnici, saj bo pomenilo višje prihodke pri zavarovancih, ki bodo sprejeli višje premije, oziroma nižje odhodke za škode pri zavarovancih, ki se bodo odločili za prestop k drugi zavarovalnici (Dugas et al., 2003, str. 200).

4.3 Napovedovanje višine komercialnega popusta

Pomemben sestavni del tržne cene zavarovalne pogodbe so po mojem mnenju variabilni komercialni popusti, ki predstavljajo razliko med premijo, ki se izračuna na podlagi premijskega cenika, ter končno ceno zavarovalne pogodbe. V tem delu sem želel napovedati višino omenjenih komercialnih popustov za nek primer (v odstotkih glede na kosmato premijo pred komercialnimi popusti). Moja predpostavka tukaj je bila, da višina dodeljenega komercialnega popusta ni naključna, ampak da je odvisna od lastnosti zavarovanca in zavarovalne pogodbe.

Napovedovanje višine komercialnega popusta se mi je zdelo zanimivo na osnovi predpostavke, da želi zavarovalni agent zaradi provizije, ki jo dobi in se običajno meri v odstotkih od pobrane premije, zavarovalno pogodbo prodati po čim višji ceni, hkrati pa je pripravljen na zavarovalno pogodbo dati dodatne variabilne komercialne popuste z namenom, da jo sploh proda, saj je zanj bolje, da dobi vsaj nekaj provizije; končna cena, ki jo z dodatkom komercialnega popusta dobimo, bi tako morala biti blizu optimalni (seveda odvisno tudi od tega, koliko optimalno deluje zavarovalni agent). Če bi uspel moj model zadovoljivo dobro napovedovati ta znesek, bi torej lahko nekako simuliral vlogo zavarovalnega agenta ter bi ga lahko uporabili kot njegovo zamenjavo (na primer pri direktnem sklepanju zavarovanj preko spleta).

Ker bom v nadaljevanju večkrat govoril o ceni oziroma tržni ceni zavarovanja, moram zapisati, da jo v nadaljevanju svojega dela obravnavam brez dodanega davka od prometa zavarovalnih poslov (DPZP) ter brez popustov in doplačil za način plačila. V praksi se na znesek, ki ga bom označeval kot tržna cena, aplicirata še ta dva dodatka, da se dobi končni znesek, ki ga mora zavarovalec plačati.

4.3.1 Izbor nabora podatkov

Pri vhodnih podatkih za to napoved mi ni bilo potrebno uporabiti omejitev iz podpoglavja 3.6.1, kajti višina dodeljenega komercialnega popusta na zavarovalni pogodbi je znana takoj, ko je le-ta sklenjena. Seveda pa sem nabor vhodnih podatkov vseeno moral omejiti do 31.12.2011, če sem želel, da moj testni scenarij (police iz leta 2012) ostane realen. Dodatno pa sem želel upoštevati tudi domnevo, da se okoliščine, ki vplivajo na višino komercialnih popustov, s časom spreminjajo in da bodo primeri iz leta 2006 za napovedovanje primerov v letu 2012 najbrž manj merodajni. Da bi ocenil ta vpliv, sem si najprej ogledal odstotek dodeljenih komercialnih popustov glede na kosmato premijo pred komercialnimi popusti po posameznih letih (od 2006 do 2012). Podatki so mi pokazali, da odstotek komercialnih popustov po letih narašča in da je moja domneva najbrž pravilna. Vendar pa me je konstantno naraščanje vseeno presenetilo, zato sem se odločil, da si po posameznih letih ogledam še spremembe v povprečni tržni ceni (premiji z upoštevanimi komercialnimi popusti) ter povprečni premiji pred dodelitvijo komercialnih popustov. V zvezi s tem sem ugotovil, da je tržna cena AK zavarovanja (tudi tukaj sem se omejil samo na riziko prometne nesreče, kot sem zapisal v podpoglavju 4.1.3) med leti 2006 in 2011 ostajala približno enaka, premija po ceniku ter odstotek komercialnih popustov pa sta se zviševala. V praksi to pomeni, da so bila po pravilih premijskega cenika AK zavarovanja vsako

leto dražja, vendar se je hkrati večal tudi odstotek dodeljenih komercialnih popustov, tako da je tržna cena (končna premija, ki jo plača zavarovanec) ostajala približno enaka. To naraščanje kosmate premije brez komercialnih popustov je torej tudi pojasnilo, zakaj se je pri moji oceni nevarnostne premije premijskega modela (podpoglavje 4.2.3.1) izkazalo, da za leta 2006–2008 le-ta po opravljeni oceni predstavlja 63,9 % kosmate premije brez komercialnih popustov, za leto 2012 pa le še 55,2 %; nevarnostna premija se torej ni znižala, ampak se je povečala osnova, od katere sem jo računal.

Je pa ravno za leto 2012, ki je bilo predmet mojega testnega scenarija, stanje nekoliko drugačno. Povprečna premija pred komercialnimi popusti je ostala približno enaka kot v letu 2011, se je pa občutno znižala povprečna tržna cena (delež komercialnih popustov se je tudi od leta 2011 do leta 2012 povečal). Razloge za to težko komentiram, saj niso bili predmet moje raziskave, možen vzrok bi po mojem mnenju na primer lahko bil vpliv konkurenčnih zavarovalnic.

Na osnovi ugotovitev sem za ta del svoje raziskave sprejel naslednji odločitvi:

- strojno učenje in validacijo modela za napovedovanje višine komercialnih popustov bom opravil na policah iz leta 2011,
- pri napovedovanju primerov iz testnega scenarija (police, ki so bile sklenjene v letu 2012) bom napovedi prilagodil za faktor, ki predstavlja razmerje med deležema komercialnih popustov v letih 2012 in 2011.

Zadnjo točko zgornjega seznama sem sprejel zato, da bi izločil vpliv spremembe tržnih razmer, kajti v svoji raziskavi sem se želel omejiti le na ocenjevanje primernega (oziroma za prodajo potrebnega) komercialnega popusta na osnovi lastnosti zavarovanca in zavarovalne pogodbe. Osnovana je tudi na predpostavki, da je bilo konec leta 2011 možno predvideti in vsaj približno oceniti padec povprečne tržne cene AK zavarovanja v letu 2012 in s tem povezano povečanje deleža komercialnih popustov. Da bi bila predstavitev mojih rezultatov vseeno korektna, sem se nadalje odločil, da bom rezultate na policah iz leta 2012 predstavil s prilagoditvijo in brez nje.

4.3.2 Izbira atributov in algoritma

Odločil sem se, da bom tudi tukaj izbiro atributov opravil z uporabo naključnega gozda, kot sem to storil že pri napovedovanju višine odškodnine ob škodnem primeru (podpoglavje 4.2.2.2). Pomembna prednost takšnega pristopa je po mojem mnenju ta, da nas dokaj hitro in usmerjeno pripelje do končnega izbora oziroma vsaj do bistveno zmanjšanega nabora atributov. Postopek, ki sem ga uporabil, je bil podoben kot v podpoglavju 4.2.2.2, zato ga ne bom znova podrobno opisoval, ampak bom navedel samo razlike.

Parametri uporabljenega naključnega gozda so bili enaki kot v podpoglavju 4.2.2.2, nisem pa tokrat uporabil prečnega vrednotenja, temveč sem rezultate preveril na 25 % validacijski množici (učenje je potekalo na preostalih 75 % nabora vhodnih podatkov, ki so ga predstavljale police iz leta 2011); vzrok za takšno odločitev je bil v tem, da sem imel približno 4 krat večjo količino

vhodnih podatkov, saj nisem bil omejen samo na primere s škodami, kar je pomenilo tudi daljše čase izvajanja. Filtriranje atributov glede na izbran kriterij (v nadaljnjo obravnavo sem vključil kandidate, ki so bili na eni od ravni sprejeti vsaj v 25 % primerov) sem tokrat avtomatiziral, kar se je sicer izkazalo za nepotrebno, saj se je nabor atributov že v prvem koraku zelo zmanjšal (s 84 na 18 atributov), na preostalih pa sem se nato odločil opravljati ročne preizkuse. Pri tem sem deloma »kršil pravila« in kak atribut, ki ni izpolnjeval zastavljenega kriterija, vendar sem bil mnenja, da bi lahko bil pomemben, ročno dodal nazaj; prav tako pa sem poskušal tudi z odvzemanjem atributov, ki so sicer prišli skozi 25 % sito, vendar so se mi zdeli manj pomembni ali redundantni glede na kak drug izbran atribut s podobnim pomenom (na primer poštni okraj prebivališča zavarovanca in registrsko območje vozila ali pa moč in prostornina motorja). Vendar pa takšna preizkušanja večinoma niso prinesla kakšnega izboljšanja glede na izbrani kriterij (za mero uspešnosti sem uporabil MSE) in vedno znova sem pristal na enaki ali zelo podobni množici atributov, do katere sem prišel že v prejšnjih korakih; to sem vzel tudi kot pokazatelj, da je uporabljen postopek primeren za izbiro atributov (tudi povsem avtomatiziran), kajti z ročnimi posegi mi ni uspelo izboljšati rezultatov. Ko v statistiki obravnavanih atributov tudi ni bilo več primerov, ki bi negativno posebej izstopali (v smislu, da bi bili zelo redko izbrani), sem se odločil, da z izbiro atributov z naključnim gozdom končam.

V naslednjem koraku sem preizkusil RPROP nevronsko mrežo; glede začetnega števila nevronov v skriti plasti sem se spet držal priporočila, ki ga podajata Xu in Chen (2008, str. 684) in sem ga natančneje opisal v podpoglavju 2.2.2.2 (enačba (5)). Ob tem sem poskušal tudi z malo manjšim in malo večjim številom (preizkušane vrednosti so bile v območju med 8 in 20 skritimi nevroni); prav tako nisem preizkusil izključno nabora atributov, ki mi ga je dal naključni gozd, ampak sem tudi tukaj še nekoliko eksperimentiral (omeniti je potrebno, da so bile množice atributov pri nevronski mreži malo drugačne, saj je bila pri nominalnih atributih potrebna uvedba psevdo-spremenljivk).

Preizkusil sem kar nekaj (okoli 10) različnih kombinacij atributov; ker so podatki specifični za moj primer, se mi na tem mestu ne zdi smiselno navajati vsebine kombinacij in pripadajočih rezultatov, ampak bom podal le povzetek. Moja glavna ugotovitev je, da se je pri tem problemu naključni gozd odrezal nekoliko bolje kot nevronska mreža. Vrednost MSE na sprejeti množici atributov je z uporabo naključnega gozda (100 dreves) znašala 59,265, z nevronsko mrežo pa sem najboljši rezultat dosegel pri spremenjeni različici te množice (odstranil sem registrsko območje vozila) ter z 11 skritimi nevroni, vrednost MSE je bila 62,554 (oba podatka sta za 25 % validacijsko množico iz nabora polic iz leta 2011).

Na podlagi teh rezultatov sem se odločil, da za napovedovanje višine komercialnega popusta uporabim algoritem naključnega gozda odločitvenih dreves ter nabor 15 atributov, ki sem ga izbral v kombinaciji s tem algoritmom in ga prikazujem v tabeli Tabela 9.

V zvezi z izbiro algoritma menim, da ni presenetljivo, da se je naključni gozd odločitvenih dreves tukaj nekoliko bolje odrezal, saj je pristop odločitvenega drevesa podoben poti do višine komercialnega popusta v praksi. Določi jo namreč človek, ki se pri tem **odloča** glede na določene

kriterije. Pri tem je sicer lahko omejen s pravili, kolikšen je največji komercialni popust, ki ga lahko v določenih okoliščinah dodeli (na primer glede na škodni rezultat zavarovanca ali glede na višino ostalih popustov). Se pa po mojem mnenju odločitve na podlagi objektivnih kriterijev mešajo s subjektivno oceno osebe, ki v imenu zavarovalnice sklepa zavarovalno pogodbo, in že zaradi te subjektivne komponente ocena modela ne more biti vedno enaka dejanski vrednosti.

Tabela 9: Izbrani atributi za napovedovanje višine komercialnega popusta

Ime atributa	**Ime atributa**	**Ime atributa**
OSB_STAROST_LET	POPUSTI_CENIK_ODS	VOZILO_OBM_REG_ID_2
KASKO_FRANSIZA_PRC	AK_KP	IZGUBA_BONUSA_AO
PREMIJSKI_RAZRED_AK	OSTALO_KP	IZGUBA_BONUSA_AOP
PREMIJSKI_RAZRED_AO	SKUPAJ_KP	VOZILO_PROIZVAJALEC
P_KM_ODS	OSB_POSTA_GRP_N1	VOZILO_RAZRED_GRP

Zapisano je delno razvidno tudi iz nabora izbranih atributov (Tabela 9); njihova imena so po mojem mnenju dovolj zgovorna, zato ne bom vsakega posebej opisoval. Pojavlja se kar nekaj atributov, ki predstavljajo popuste po ceniku in imajo že sicer vpliv na končno premijo, tukaj imam v mislih atributa za odstotek popustov po ceniku ter za premijski razred. Možna razlaga bi po mojem mnenju bila, da je višina komercialnega popusta omejena z višino ostalih popustov ter da se v primeru zavarovancev, ki imajo manj bonusa (so v višjem premijskem razredu), le-to delno izravna z večjimi komercialnimi popusti ter da na koncu plačajo podobno ceno kot zavarovanci iz nižjih premijskih razredov. Izbran je bil tudi atribut, ki pove odstotek dodatnega popusta za zavarovance, ki letno prevozijo malo kilometrov; tudi tukaj je možna razlaga, da se tistim, ki dobijo ta popust, v povprečju prizna manj komercialnega popusta. Pojavijo se tudi kar trije atributi (končnica _KP), ki povedo znesek premij tega zavarovanca v preteklosti, zanimivo pa je, da ni nobenega atributa, ki bi ta znesek povezal s škodami. Možen sklep bi bil, da je za dodelitev komercialnega popusta pomembno predvsem, da je nekdo zvesta stranka in je zavarovalnici prinesel že veliko premije, manj pomemben pa je njegov škodni rezultat. Seveda bi bilo vse tukaj zapisane domneve potrebno preveriti z dodatno raziskavo, ki pa je na tem mestu nisem opravil in predstavlja možnost za nadgradnjo. V njej bi bilo zanimivo raziskati tudi vpliv ostalih atributov iz nabora izbranih atributov (Tabela 9), predvsem vpliv posameznih vrednosti iz zaloge vrednosti nominalnih atributov (ali dobivajo zavarovanci iz mariborskega poštnega okraja večje ali manjše komercialne popuste, pri katerih proizvajalcih avtomobilov so dodeljeni največji oziroma najmanjši komercialni popusti in podobno).

4.3.3 Predstavitev rezultatov in možnosti uporabe modela

Rezultate napovedovanja višine komercialnega popusta glede na različne mere podajam v tabeli Tabela 10. Pri tem naj poudarim, da sem napovedoval **odstotek** komercialnega popusta glede na kosmato premijo pred komercialnimi popusti. V predzadnji vrstici tabele Tabela 10 je podano razmerje med končnim zneskom komercialnih popustov, če bi upoštevali moje napovedi, in dejanskim končnim zneskom komercialnih popustov za določen nabor polic, v zadnji vrstici pa

enako razmerje še za vsoto pobranih premij. V zadnjem stolpcu sem za realni testni scenarij (police iz leta 2012) dodal rezultate, ki sem jih dobil tako, da sem napovedi pomnožil s korekcijskim faktorjem, ki predstavlja razmerje med deležema komercialnih popustov glede na kosmato premijo brez upoštevanja le-teh za leti 2012 in 2011 in je v mojem primeru znašal 1,334 (podrobnejšo razlago glede te prilagoditve sem podal v podpoglavju 4.3.1).

Tabela 10: Rezultati napovedovanja odstotka komercialnega popusta z algoritmom naključnega gozda na validacijski množici (police iz leta 2011) in realnem testnem scenariju (police iz leta 2012)

Mera \ Nabor podatkov	**Police 2011 (25 % validacijska množica)**	**Police 2012 (testni scenarij)**	**Police 2012 (prilagojeno za 1,334)**
Povprečna kvadratna napaka (MSE)	59,265	112,073	73,615
Povprečna absolutna napaka (MAE)	5,734	8,355	6,406
Povprečna predznačena razlika (MSD)	-0,028	-6,448	0,161
$\sum kom.pop.napoved \Big/ \sum kom.pop.$ (v %)	98,24	76,52	102,08
$\sum premije\ napoved \Big/ \sum premije$ (v %)	100,50	109,64	99,15

Na podlagi rezultatov v tabeli Tabela 10 ugotavljam, da razlika med validacijsko množico in prilagojenimi rezultati za leto 2012 ni posebej velika, iz česar sklepam, da je bilo strojno učenje uspešno; po drugi strani pa je vidno tudi, da bi bili rezultati bistveno slabši, če ne bi upošteval korekcijskega faktorja. V primeru, da moja predpostavka iz podpoglavja 4.3.1 ne drži in konec leta 2011 ne bi uspeli ustrezno oceniti sprememb na trgu v letu 2012 (padec tržne cene AK zavarovanja) ter bi v praksi uporabili napovedi brez prilagoditve, bi dobili napačne napovedi, ki bi lahko vodile v napačne poslovne odločitve. Zato je po mojem mnenju pomembno, da modela, ko je uveden v praktično uporabo, ne sprejmemo kot dokončnega enkrat za vselej, ampak ves čas sproti vrednotimo njegove rezultate in po potrebi ustrezno ukrepamo; skratka, kot ugotavljajo tudi Chapman et al. (2000, str. 10), proces podatkovnega rudarjenja s tem, ko je rešitev prenesena v uporabo, ni končan.

Končna zneska komercialnih popustov oziroma pobranih premij, ki bi ju dobili z uporabo komercialnih popustov mojega modela, sta zelo blizu dejanskima zneskoma na osnovi komercialnih popustov, ki so bili v resnici dodeljeni na posamezni polici (zadnji dve vrstici tabele Tabela 10). To nakazuje, da bi moj model lahko bil ustrezna zamenjava za zavarovalnega agenta. Tipičen primer uporabe bi bila direktna prodaja zavarovanj preko spleta. Za zavarovalnico bi to pomenilo manjše stroške za provizije, ki jih izplača zavarovalnim agentom, in del tega prihranka bi lahko prenesla na zavarovance v obliki nižjih premij, kar bi pomenilo obojestransko korist. Pri tem se je sicer potrebno zavedati, da zavarovalni agenti zavarovancem načeloma nudijo tudi pomoč v obliki svetovanja pri izbiri najprimernejše oblike zavarovanja ter

ob škodnih dogodkih in da bi bili zavarovanci v tem primeru za to podporo prikrajšani. Druga možnost uporabe modela pa je tudi pomoč (manj izkušenim) zavarovalnim agentom, ki bi jim model predlagal ustrezno višino komercialnega popusta.

Ne smemo pa pozabiti, da navedena rezultata končnih zneskov komercialnih popustov oziroma pobranih premij, ki bi ju dobili z uporabo komercialnih popustov mojega modela, predstavljata samo del resnice, saj zajemata predpostavko, da bi bile vse te police s takšnimi komercialnimi popusti tudi dejansko sklenjene. V resnici se nekateri (ali v skrajnem primeru vsi) zavarovanci, ki bi jim model dodelil nižji komercialni popust od tistega, ki so ga prejeli v resnici, ter s tem višjo premijo, ne bi odločili za sklenitev. Po drugi strani bi zavarovanci, ki jim je model dodelil višji komercialni popust, to z veseljem sprejeli ter sklenili zavarovanje po nižji ceni, kar bi pomenilo nižje prihodke za zavarovalnico ob enakih stroških. Verjetnejše od omenjenega skrajnega primera se mi sicer zdi, da bi nekateri zavarovanci police sklenili tudi po višji ceni (zavarovalni agent je lahko napačno ocenil zavarovanca in mu ponudil višji komercialni popust, kot bi bilo potrebno), hkrati pa bi bile sklenjene tudi nekatere dodatne police (primeri, ki jih nimamo v naboru podatkov, saj je zavarovalni agent ponudil prenizek komercialni popust ter polica ni bila sklenjena).

Za pridobitev rezultatov na osnovi zadnjega opisanega scenarija bi morali poznati najvišjo premijo, ki jo je posamezen zavarovanec bil pripravljen plačati, hkrati pa tudi podatke vseh strank, ki so se zanimale za sklenitev zavarovanja pri izbrani zavarovalnici ter se zaradi previsoke premije na koncu niso odločile za sklenitev. Omenjeni podatki nam v praksi niso neposredno dostopni, lahko pa jih poskusimo oceniti; ker bi natančnejša ocena z izvedbo tržne raziskave presegala okvir mojega dela, sem v ta namen sprejel določene predpostavke ter na njihovi osnovi izvedel simulacijo, s katero sem želel vsaj približno prikazati učinek uporabe svojega modela v praksi. Podrobnosti omenjene simulacije ter rezultate predstavljam v podpoglavju 5.2.

5 PREDLOG PRAKTIČNE UPORABE

V tem delu podajam predlog, kako bi lahko kombinacijo vseh napovedi, ki sem jih opisal v četrtem poglavju, uporabili v praksi. Po CRISP-DM pristopu gre tukaj za fazo vključitve v uporabo (šesta faza). Moj predlog predvideva uporabo teh napovedi kot pomoč v trenutku sklepanja zavarovalne pogodbe. Algoritem, ki ga predlagam, je sledeč:

1. operater (na primer zavarovalni agent) vnese podatke o zavarovalni pogodbi in stranki (zavarovancu), ki želi pogodbo skleniti, v sistem,
2. po postopku, ki sem ga opisal v podpoglavju 4.2.3 in zajema napovedi iz podpoglavij 4.2.1 in 4.2.2, se oceni nevarnostna premija,
3. nevarnostna premija se po določenih pravilih pretvori v kosmato premijo; to premijo bom imenoval **minimalna premija** in predstavlja premijo, ki je enaka pričakovani višini vseh stroškov zavarovalnice s to zavarovalno pogodbo (torej ne pokriva le pričakovane višine odškodnin, kot nevarnostna premija, ampak tudi ostale pričakovane stroške),

4. izračuna se premija v skladu s premijskim cenikom (na tej stopnji je ta brez kakršnihkoli komercialnih popustov),
5. po postopku, ki sem ga opisal v podpoglavju 4.3, se napove višina komercialnega popusta za podane lastnosti zavarovanca in zavarovalne pogodbe,
6. ocenjen odstotek komercialnega popusta iz 5. koraka se aplicira na kosmato premijo v skladu s premijskim cenikom iz 4. koraka,
7. dobljena premija iz 6. koraka se primerja z minimalno premijo iz 3. koraka in določi se višji izmed zneskov,
8. če je znesek iz 7. koraka nižji od premije iz 4. koraka, se razlika operaterju prikaže kot priporočena višina komercialnega popusta v odstotkih, v nasprotnem primeru se kot priporočena višina komercialnega popusta prikaže vrednost 0,
9. če je ocenjena minimalna premija iz 3. koraka višja ali enaka izračunani premiji v skladu s premijskim cenikom iz 4. koraka, se operaterju onemogoči dodeljevanje dodatnih komercialnih popustov, ampak lahko zavarovanje sklene samo po izračunani premiji po premijskem ceniku brez komercialnih popustov (znesek iz 4. koraka),
10. operater lahko uporabi priporočeno višino komercialnega popusta ali pa po svoji presoji dodeli komercialni popust v drugačni višini, vendar pa končna cena ne sme biti nižja od izračunane minimalne premije iz 3. koraka.

Pravil iz 3. koraka predlaganega postopka v sklopu te raziskave ne obravnavam; tukaj gre za postopek preračuna nevarnostne premije v kosmato premijo pred komercialnimi popusti. Uporabi se lahko enak postopek, kot ga zavarovalnica že uporablja za pretvorbo nevarnostne premije, izračunane po aktuarskem modelu, v kosmato premijo pred komercialnimi popusti; dober pregled metod, ki se pri tem uporabljajo, podajata Werner in Modlin (2016).

S pravilom v 9. koraku želim preprečiti oziroma omejiti sklepanje zavarovalnih pogodb, ki so po oceni mojega modela bolj tvegane kot to ocenjuje premijski cenik in bi po oceni modela prinesle izgubo. To je prvi izmed ciljev predlaganega postopka. Ker pa je napoved minimalne premije po mojem modelu lahko tudi napačna, mora postopek seveda vseeno dovoljevati sklenitev zavarovalne pogodbe po ceni, ki je bila izračunana v skladu s premijskim cenikom (4. korak), vendar pa pri takih mejnih primerih ne dovoljuje komercialnih popustov; v mojem predlogu gre, kot sem zapisal že v uvodnem poglavju, namreč za dopolnitev in ne za nadomestitev obstoječih aktuarskih modelov in premijskih cenikov.

Drugi izmed ciljev postopka, ki ga predlagam, je, da pomaga operaterjem, ki v imenu zavarovalnice sklepajo zavarovalno pogodbo, tako, da jim poskuša predlagati optimalno ceno (v obliki ustreznega odstotka komercialnega popusta). Ker gre samo za predlog optimalne cene (ki ga določi model na podlagi množice preteklih primerov), ima operater možnost, da po svoji presoji dodeli drugačen odstotek komercialnega popusta; ta je lahko nižji (kar je za zavarovalnico še bolje) ali pa tudi višji. Zavedati se je namreč treba, da bo model iz podpoglavja 4.3 včasih podal tudi napačno oceno, hkrati pa lahko ima operater še dodatno specifično znanje o posameznem primeru, ki ga preko atributov, ki so na razpolago, ne more vnesti v model.

Obenem pa želim s predlaganim postopkom tudi preprečiti sklepanje zavarovalnih pogodb s previsokimi komercialnimi popusti, ki bi bile v škodo zavarovalnice, kar bi lahko zavarovalni agenti storili z namenom, da na vsak način dobijo vsaj nekaj provizije (ali pa tudi, da omogočijo znancu čim bolj ugoden nakup zavarovanja). To je tretji izmed ciljev predlaganega postopka in je zajet v omejitvi v njegovem 10. koraku.

Dodam naj, da je uporaba postopka možna tudi pri direktnem trženju zavarovanj brez vmesnega operaterja, na primer preko svetovnega spleta. V tem primeru prvi korak izvede stranka sama, izračunana priporočena cena (torej cena po premijskem ceniku z dodanim priporočenim komercialnim popustom) pa se takoj ponudi stranki. Seveda stranka, za razliko od operaterja, nima možnosti, da bi uporabila drugačno višino komercialnega popusta, ampak lahko ponujeno ceno samo sprejme ali pa ne. Zanimiva bi tukaj morda bila implementacija sistema, ki bi se lahko s stranko tudi pogajal (Bella, Ferri, Hernández-Orallo, & Ramírez-Quintana, 2009), vendar pa bi bilo potrebno najti način, kako preprečiti, da bi stranke začele izkoriščati poznavanje delovanja sistema, če bi se zanj razvedelo. Kadar je cena po premijskem ceniku nižja od izračunane minimalne premije, obstaja več možnosti delovanja sistema – lahko dovolimo sklenitev po ceni, ki je bila izračunana po premijskem ceniku, lahko ceno dvignemo na višino minimalne premije, lahko pa v takih primerih sploh onemogočimo neposredno sklepanje in zahtevamo, da stranka zavarovanje sklene pri strokovnem delavcu zavarovalnice. Za odločitev, katero različico uporabiti v praksi, bi bilo po mojem mnenju najbolje spremljati primere, ki bi se pojavili, ter sistem po potrebi naknadno prilagoditi.

5.1 Ovrednotenje učinka predlaganega postopka

Za ovrednotenje učinka predlaganega postopka sem si najprej ogledal, kakšen vpliv bi imele posamezne omejitve na police iz leta 2012 (realni testni scenarij), pri čemer že omenjenega osamelca nisem izločil iz nabora podatkov. Ker me je v tej točki zanimal le vpliv omejitev, ki jih uvaja moj postopek, sem v 5. koraku namesto z modelom napovedanega komercialnega popusta uporabil dejansko dodeljeni komercialni popust na polici.

Ugotovil sem, da bi sistem zaradi pogoja minimalne premije (7. in 8. korak predlaganega postopka) omejil višino komercialnega popusta v 48,40 % primerov. Police, na katerih je v veljavo stopila omenjena omejitev, so pri tem imele slabše (višje) razmerje med škodami in premijami, ki predstavlja enostavni škodni količnik (78,91 % proti 59,90 %). V primeru, da bi uporabili omejene komercialne popuste in bi vse police kljub temu bile sklenjene (kar se v resnici skoraj zagotovo ne bi zgodilo), bi pobrali 9,80 % več premije, skupno razmerje med škodami in pobranimi premijami pa bi se izboljšalo s 66,80 % na 60,84 %. Druga skrajnost je, da nobena od polic z omejenim komercialnim popustom ne bi bila sklenjena; v tem primeru bi zabeležili 36,30 % padec pobrane premije ter izboljšanje enostavnega škodnega količnika na 59,90 %. Ob tem je potrebno opozoriti, da nižji škodni količnik ob nižjem znesku premij (nižjem tržnem deležu) ne pomeni nujno večjega dobička, saj je potrebno upoštevati tudi višino razlike med premijami in škodami, iz katere mora zavarovalnica pokriti fiksne stroške poslovanja ter zahteve

po dobičku (v mojem obravnavanem primeru bi imela zavarovalnica za to na razpolago približno 635.000 EUR manj), hkrati pa nižji tržni delež pomeni tudi manjšo razpršitev tveganja.

Nadalje sem preveril še pogoj iz 9. koraka predlaganega postopka ter ugotovil, da je bila premija po ceniku (brez komercialnih popustov) nižja od minimalne premije na podlagi mojega modela v 23,10 % primerov (vsi ti primeri so sicer zajeti med 48,40 % primerov, pri katerih je prišla v poštev omejitev na podlagi 7. in 8. koraka), tudi tukaj pa so imeli primeri, pri katerih je bil pogoj izpolnjen, slabše razmerje med škodami in premijami (94,40 % proti 61,68 %). Če v teh primerih ne bi dopustili sklenitve police z izračunano premijo po ceniku, ampak bi zahtevali najmanj minimalno premijo po mojem modelu in bi bile vse police sklenjene, bi dosegli 14,95 % povečanje pobrane premije in izboljšanje enostavnega škodnega količnika na 58,11 % (v drugem skrajnem primeru, ko nobena od polic z omejitvijo ne bi bila sklenjena, bi dobili enake rezultate kot prej, saj je nepomembno, ali bi v tem koraku ponudili premijo po ceniku brez komercialnih popustov ali minimalno premijo, kajti polica v nobenem primeru ne bi bila sklenjena).

Na podlagi rezultatov sklepam, da v naboru zavarovalnih pogodb, ki sem ga preučeval, obstajajo pogodbe, na katerih so bile zaračunane prenizke premije, bodisi zaradi prenizke premije po ceniku bodisi zaradi previsokih komercialnih popustov, predlagani postopek oziroma moj model pa je bil vsaj delno uspešen pri zaznavanju takšnih primerov. Postopek, ki takšne primere poskuša omejiti, bi v prvi vrsti pomenil korist za zavarovalnico, vendar pa bi bilo pred vključitvijo v uporabo potrebno natančneje raziskati njegove učinke v smislu tržnega deleža ter tudi s stališča tržne strategije zavarovalnice (na primer morebitno zavestno nudenje zavarovanj AK po prenizki ceni strankam, ki prinašajo dobiček pri drugih vrstah zavarovanj). Za zavarovance predlagan sistem sam po sebi ne bi prinašal prednosti, kajti nekateri izmed njih bi morali plačati višje premije, pri ostalih pa ne bi bilo spremembe. To bi bilo drugače, če bi se zavarovalnica odločila, da del morebitnega povečanja dobička, ki bi ga dosegla z uporabo predlaganega postopka, prenese na zavarovance v obliki znižanja premij, kar bi bilo koristno tudi zanjo, saj bi lahko pridobila nove zavarovance ter uravnovesila prej opisano znižanje tržnega deleža zaradi omejitev z minimalno premijo. Tak proces bi v bistvu pomenil bolj pošteno določanje premij po podmnožicah zavarovancev, ki sem ga obravnaval že v podpoglavju 4.2.3.4.

Na podlagi zapisane diskusije med drugim zaključujem tudi, da je določanje ustrezne cene zavarovalne pogodbe zapleten postopek, pri katerem je potrebno zraven pričakovane vrednosti izplačil upoštevati še številne druge dejavnike (na primer vpliv na tržni delež ter pokrivanje fiksnih stroškov, vpliv na prodajo drugih zavarovalnih pogodb, vpliv na razpršitev tveganja); kljub temu pa je čim bolj pošteno določena premija bistvenega pomena, saj prinaša koristi tako za zavarovalnico kot za zavarovance.

5.2 Preizkus delovanja predlaganega sistema s simulacijo

V tem delu raziskave sem želel s simulacijo na svojem realnem testnem scenariju (police, ki so bile sklenjene v letu 2012, podpoglavje 4.1.2) preveriti posledice uporabe predlaganega postopka v praksi ter hkrati tudi ovrednotiti model za napovedovanje višine komercialnega popusta

(podpoglavje 4.3). Ker bi raziskava trga presegala okvir mojega magistrskega dela in tudi ni bila med mojimi cilji, sem moral za izvedbo simulacije sprejeti nekaj predpostavk. Predpostavil sem, da je premija, ki jo je neka stranka pripravljena plačati za zavarovanje, določena z višino komercialnega popusta ter tako izločil ostale vplive na to vrednost, na primer zavarovalno vsoto. Nadalje sem predpostavil, da je ta »zahtevan« komercialni popust normalno porazdeljen, pri čemer sem parametre porazdelitve določil iz vzorca, ki so ga predstavljale police, ki so bile sklenjene v letu 2012 (vzorčno povprečje, vzorčni standardni odklon); to predpostavko sem povzel po podobni predpostavki, ki so jo v svoji raziskavi uspešno uporabili Bella et al. (2009, str. 9, 15). Primeri v naboru podatkov seveda predstavljajo samo dejansko sklenjene police, ne poznamo pa primerov, pri katerih se stranke niso odločile za sklenitev (in so polico sklenile pri konkurenčni zavarovalnici ali pa sploh ne); v zvezi s tem sem predpostavil, da je potencialnih kupcev dvakrat toliko (in se jih pri obstoječi cenovni politiki 50 % ni odločilo za sklenitev). Vzorec, na katerem sem izvedel simulacijo, sem tako dobil iz obstoječega vzorca (police iz leta 2012, osamelca nisem izločil) z naključnim vzorčenjem z vračanjem (angl. *bootstrap*) v velikosti 200 %. Omeniti velja tudi način preračuna nevarnostne premije na podlagi napovedi mojega modela v minimalno premijo (3. korak predlaganega postopka); v ta namen sem uporabil faktor, ki sem ga izračunal v podpoglavju 4.2.3.1, pri čemer pa sem moral nadalje upoštevati, da ta faktor predvideva določene komercialne popuste in je zato dobljena premija nekoliko previsoka, kar sem izničil tako, da sem nanjo apliciral še delež komercialnih popustov na policah iz leta 2012.

V simulaciji sem za vsak primer iz predpostavljene porazdelitve naključno izbral zahtevan komercialni popust ter na njegovi podlagi določil najvišjo ceno, pri kateri še pride do sklenitve, ter jo primerjal z dejansko plačano premijo (ki predstavlja trenutno cenovno politiko in popust, ki ga je dodelil zavarovalni agent) (drugi stolpec v tabeli Tabela 11) ter s premijo po svojem modelu; v primeru, da je bila ta najvišja sprejemljiva cena nižja, sem primer izločil (ni prišlo do sklenitve). Delovanje sistema sem nadalje zastavil tako, da sem v primerih, ko je bila ocenjena minimalna premija višja od premije po ceniku, »ponudil« sklenitev po premiji po ceniku brez komercialnih popustov (9. korak algoritma). Da bi čim bolj izločil vpliv naključja v simulaciji (naključno izbiranje z vračanjem, izbor zahtevanega komercialnega popusta za posameznega zavarovanca), sem jo ponovil 10 krat (seveda vsakič z drugimi naključnimi vrednostmi) ter uporabil povprečje rezultatov.

Pri primerjavi s premijo po svojih modelih sem preizkusil tri različice, in sicer sem prvič uporabil samo napovedano višino komercialnega popusta (podpoglavje 4.3, uporabil sem prilagojene vrednosti za leto 2012) (tretji stolpec v tabeli Tabela 11), drugič omejitev višine komercialnega popusta na osnovi ocenjene minimalne premije v kombinaciji z dejansko dodeljenim komercialnim popustom (četrti stolpec v tabeli Tabela 11) ter tretjič celoten postopek, ki sem ga predlagal na začetku petega poglavja (kombinacija prejšnjih dveh različic) (peti stolpec v tabeli Tabela 11). Ker ponujena premija tudi v primeru uporabe mojega postopka temelji na premiji po ceniku (moj postopek samo določi višino komercialnega popusta ter postavi spodnjo mejo za premijo), sem nazadnje preizkusil še vpliv znižanja izhodiščne premije po ceniku, kar predstavlja že omenjen (podpoglavje 5.1) prenos pridobljenih koristi na zavarovance; pri tem sem preizkusil

različne odstotke znižanja (med 2,5 % in 15 % v korakih po 2,5 %) ter najboljše rezultate (glede na pobrano premijo in razliko med premijami in škodami) dobil pri vrednosti 7,5 %, te rezultate predstavljam v zadnjem stolpcu tabele Tabela 11.

Tabela 11 prikazuje s simulacijo dobljeno vsoto premij (p), izplačil (s) ter razliko med njima ($p - s$) v odstotku od dejansko realiziranih zneskov na policah iz leta 2012 (p_r, s_r, $p_r - s_r$), nadalje pa še enostavni škodni količnik s/p; predstavitev rezultatov v deležu od dejanske realizacije (in ne v absolutnih zneskih) se mi je zdela bolj informativna, hkrati pa pomeni tudi zaščito podatkov izbrane zavarovalnice.

Tabela 11: Rezultati preizkusa izgrajenih modelov in predlaganega postopka s simulacijo po različnih scenarijih

Vrednost \ Scenarij	**Dejansko plačana premija (v %)**	**Popust model (v %)**	**Omejitev minimalna premija (v %)**	**Predlagan postopek (v %)**	**Predlagan postopek in 7,5 % znižana premija po ceniku (v %)**
Premije (p/p_r)	101,50	102,35	78,85	80,54	93,15
Izplačila (s/s_r)	102,38	103,59	68,66	71,26	87,66
Razlika ($(p - s)/(p_r - s_r)$)	99,75	99,85	99,34	99,19	104,20
Količnik (s/p)	67,37	67,61	58,17	59,11	62,86

Iz drugega stolpca (Tabela 11) gre razbrati, da so rezultati uporabe dejansko plačane premije v simulaciji (premija po ceniku in od agenta dodeljen komercialni popust) blizu v resnici realiziranim vrednostim, iz česar sklepam, da uporabljene predpostavke niso nujno neresnične. Prav tako so rezultati podobni rezultatom uporabe komercialnega popusta po oceni mojega modela (tretji stolpec tabele Tabela 11), iz česar zaključujem, da je moj model dokaj dobro posnemal delovanje zavarovalnega agenta. Omejitev na podlagi minimalne premije (četrti stolpec v tabeli Tabela 11) je občutno znižala znesek pobranih premij, vendar pa se je še bolj znižal znesek izplačil in razlika med premijami in škodami je ostala približno enaka; slednje pomeni tudi, da je predlagana omejitev precej izboljšala enostavni škodni količnik. Celoten predlagan postopek (peti stolpec tabele Tabela 11) je dal dokaj podobne rezultate kot uporaba samo omejitve z minimalno premijo, kar je pričakovano, saj je, kot že zapisano, dodeljevanje komercialnih popustov z mojim modelom dalo primerljive rezultate, kot bi jih dosegel zavarovalni agent (to je bil tudi cilj izgradnje tega modela).

V zvezi z učinkom omejitve z minimalno premijo ocenjujem, da so se v realnosti nekatere zavarovalne pogodbe sklepale s prenizko premijo, ki ni ustvarjala dodatne razlike med pobrano premijo in izplačili, katera pa je potrebna za pokritje stroškov poslovanja ter ustvarjanje dobička. Predlagan postopek je takšne primere omejil; končni učinek je podobna razlika med premijami in izplačili ob občutno nižjem znesku pobranih premij. Slabosti tega sta nižji tržni delež ter manjša razpršitev tveganja, prednost pa je manjši znesek variabilnih stroškov, ki jih je potrebno pokriti iz omenjene razlike.

Tudi ideja o prenosu pridobljenih koristi na zavarovance v obliki nižjih premij (zadnji stolpec tabele Tabela 11) se je izkazala za uspešno, saj je izravnala del izgubljenega tržnega deleža, delež števila sklenjenih pogodb v simulaciji glede na dejansko realizirano vrednost v letu 2012 se je namreč povečal s 65 % pri uporabi predlaganega postopka brez znižanja premije na 81 % ob uporabi istega postopka ter znižanih premij; hkrati je ta ukrep povečal razliko med premijami in škodami, ki je osnova za dobiček zavarovalnice, korist v obliki nižjih premij pa so imeli tudi zavarovanci.

SKLEP

V pričujočem magistrskem delu sem podal primer izvedbe procesa podatkovnega rudarjenja s strojnim učenjem za določanje cen zavarovanj AK, kar je bil glavni namen mojega dela.

Pri tem sem najprej podal pregled vseh pojmov, s katerimi se pri tem srečamo, ter teoretični opis konkretnih postopkov, ki pridejo v poštev pri izvedbi. Iz tega dela je razvidno, da gre za interdisciplinarni proces, pri katerem potrebujemo za tehnično izvedbo znanja s področja računalništva in algoritmov, hkrati pa moramo problem tudi vsebinsko zelo dobro poznati, v konkretnem primeru je to pomenilo znanja s področja zavarovalništva. Obstaja namreč kopica vsebinskih podrobnosti, katere je potrebno upoštevati in ki onemogočajo, da bi nad podatki enostavno pognali algoritme strojnega učenja ter dobili zadovoljive rezultate.

Pravkar zapisano je razvidno iz dokaj obširnega poglavja, v katerem sem opisal postopek priprave podatkov. To poglavje se od literature, v kateri je splošen proces priprave podatkov za podatkovno rudarjenje dobro pokrit, razlikuje po tem, da je vezano na konkreten problem, ki sem ga obravnaval, in se mi kot takšno zdi uporabno kot pomoč pri izvajanju podobnih raziskav; seveda vse podrobnosti najbrž ne bodo enake, lahko pa moj opis druge raziskovalce morda opomni na kakšno podrobnost, ki so jo pozabili upoštevati, ali jim da idejo za rešitev kakšnega problema. S tem opisom sem izpolnil enega izmed ciljev, ki sem si jih zadal na začetku raziskave. Konkreten rezultat tega dela raziskave je tudi skupek procedur za pridobivanje nabora podatkov za podatkovno rudarjenje iz podatkovne baze izbrane zavarovalnice, ki pa je seveda uporaben samo interno.

V nadaljevanju sem s kombinacijo modelov za napovedovanje verjetnosti za nastop škodnega primera ter za napovedovanje višine odškodnine ob škodnem primeru razvil model za napoved pričakovane višine odškodnine, ki predstavlja nevarnostno premijo oziroma tveganost zavarovalne pogodbe. Pri tem sem, skladno s cilji svojega dela, uporabil strojno učenje, natančneje nevronsko mrežo, ki se je v sorodnih raziskavah izkazala za uspešno.

Stranski produkt tega procesa je bil nabor atributov, ki so se izkazali za relevantne za opravljene napovedi. Priprava seznama teh atributov je tudi predstavljala enega izmed mojih ciljev. Identificiran nabor atributov se v precejšnji meri prekriva s »klasičnimi« parametri za določanje premij pri zavarovanjih AK, kar po mojem mnenju govori v prid temu, da ti podatki dejansko vplivajo na tveganost, po drugi strani pa prikazuje, da se da tudi z metodo podatkovnega

rudarjenja s strojnim učenjem, kjer se pustimo voditi podatkom, priti do podobnih rezultatov kot s poglobljenimi statističnimi analizami. V delih, kjer se nabora parametrov ne prekrivata, pa so s strojnim učenjem izbrani atributi lahko odskočna deska za iskanje parametrov, ki bi jih bilo v prihodnosti morda smiselno upoštevati v aktuarskih izračunih.

Med zastavljenimi cilji je bila tudi primerjava izgrajenega modela z obstoječim aktuarskim modelom. V tej primerjavi se moj model za napovedovanje nevarnostne premije sicer ni izkazal za bistveno izboljšavo, vendar pa menim, da je treba kot uspeh vzeti že to, da sem s strojnim učenjem uspel z relativno majhnim časovnim vložkom (v primerjavi z vložkom zavarovalnice) doseči primerljive rezultate.

Sicer pa, kot sem omenil že na začetku, namen modela, ki sem ga razvil, ni bil v tem, da bi konkuriral obstoječim aktuarskim modelom, ampak da bi jih dopolnjeval. V okviru svojega magistrskega dela sem tako, vzporedno z ovrednotenjem modelov, nakazal tudi možnosti za njihovo praktično uporabo ter prednosti in slabosti, ki bi jih prinesla, kar je tudi sodilo med cilje mojega dela. Ena izmed omenjenih možnosti je uporaba znotraj postopka, ki sem ga predlagal v petem poglavju. Za uporabo v tem postopku sem razvil tudi model za ocenjevanje primerne višine komercialnega popusta. S tem ter s predlogom združenega postopka sem izpolnil še dva izmed svojih ciljev. Izgrajene modele ter predlagan postopek sem na koncu ovrednotil še z izvedbo simulacije (ki sicer ne predstavlja nujno rezultatov, ki bi jih dobili v praksi, saj je temeljila na kar nekaj predpostavkah) ter izpostavil pozitivne in negativne učinke uporabe predlaganega postopka.

Na podlagi spoznanj, do katerih sem prišel tekom raziskave, menim, da na obravnavanem področju obstajajo možnosti za uspešno uporabo predstavljenih postopkov ter da lahko moje delo služi kot pomoč pri njihovem uvajanju.

LITERATURA IN VIRI

1. Apte, C., Grossman, E., Pednault, E., Rosen, B., Tipu, F., & White, B. (1998). Insurance Risk Modeling Using Data Mining Technology. Najdeno 6. marca 2016 na spletnem naslovu https://www.researchgate.net/publication/242404528_Insurance_Risk_Modeling_Using_Data_Mining_Technology
2. Apte, C., Grossman, E., Pednault, E. P. D., Rosen, B. K., Tipu, F. A., & White, B. (1999). Probabilistic estimation-based data mining for discovering insurance risks. *IEEE Intelligent Systems and their Applications, 14*(6), 49–58. Najdeno 6. marca 2016 na spletnem naslovu http://ieeexplore.ieee.org.nukweb.nuk.uni-lj.si/ielx5/5254/17571/00809568.pdf
3. Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (2009). Feature Dependent Models. Najdeno 11. junija 2016 na spletnem naslovu http://users.dsic.upv.es/~abella/papers/FDM.pdf
4. Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., & Wiswedel, B. (2009). KNIME – The Konstanz Information Miner. *SIGKDD Explorations 11*(1), 26–31. Najdeno 19. junija 2016 na spletnem naslovu http://www.kdd.org/exploration_files/p4V11n1.pdf
5. Boncelj, J. (1983). *Zavarovalna ekonomika*. Maribor: Obzorja.
6. Breiman, L. (2001). Random Forests. *Machine Learning 45*(1), 5–32. Najdeno 17. junija 2016 na spletnem naslovu http://link.springer.com/content/pdf/10.1023%2FA%3A1010933404324.pdf
7. Caruana, R., & Freitag, D. (1994). Greedy Attribute Selection. Najdeno 10. marca 2016 na spletnem naslovu http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.3576&rep=rep1&type=pff
8. Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2*(3), 27:1–27:27. Najdeno 5. junija 2016 na spletnem naslovu http://kimiko.biome.tk/20140731/Reference/2011%20-%20Chang%20and%20Lin%20-%20LIBSVM%20A%20Library%20for%20Support%20Vector%20Machines.pdf
9. Chapados, N., Bengio, Y., Vincent, P., Ghosn, J., Dugas, C., Takeuchi, I., & Meng, L. (2001). Estimating Car Insurance Premia: a Case Study in High-Dimensional Data Inference. *Advances in Neural Information Processing Systems 14*(1), 1369–1376. Najdeno 8. junija 2016 na spletnem naslovu http://www-labs.iro.umontreal.ca/~vincentp/Publications/insurance_nips2001.pdf
10. Chapados, N. (2010, 8. februar). Data Mining Algorithms for Actuarial Ratemaking. Najdeno 6. marca 2016 na spletnem naslovu http://www.apstat.com/documents/apstat_datamining_insurance.pdf
11. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide. b.k.: CRISP-DM consortium. Najdeno 12. junija 2016 na spletnem naslovu https://www.the-modeling-agency.com/crisp-dm.pdf
12. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research 16*, 321–357.

Najdeno 28. maja 2016 na spletnem naslovu https://www.jair.org/media/953/live-953-2037-jair.pdf
13. Chawla, N. V. (2005). Data Mining for Imbalanced Datasets: An Overview. V O. Maimon & L. Rokach (ur.), *Data Mining and Knowledge Discovery Handbook* (str. 853–867). New York: Springer.
14. Dal Pozzolo, A. (2010). *Comparison of Data Mining Techniques for Insurance Claim Prediction*. Bologna: Alma Mater Studiorum - Università di Bologna, Facoltà di Scienze Statistiche.
15. Deepa, S. N., & Sheela, K. G. (2013). Review on Methods to Fix Number of Hidden Neurons in Neural Networks. *Mathematical Problems in Engineering 2013*(6). Najdeno 3. junija 2016 na spletnem naslovu http://downloads.hindawi.com/journals/mpe/2013/425740.pdf
16. Devore, J. L., & Berk, K. N. (2012). *Modern Mathematical Statistics with Applications* (2nd ed.). New York: Springer.
17. Dugas, C., Bengio, Y., Chapados, N., Vincent, P., Denoncourt, G., & Fournier, C. (2003). Statistical Learning Algorithms Applied to Automobile Insurance Ratemaking. Najdeno 6. marca 2016 na spletnem naslovu http://www.iro.umontreal.ca/~lisa/pointeurs/dugas03.pdf
18. Fernandez, G. (2003). *Data Mining Using SAS Applications*. Boca Raton: Chapman & Hall/CRC.
19. Fric, L., & Prijanovič, D. (2010). *Premoženjska zavarovanja* (1. izd.). Ljubljana: Slovensko zavarovalno združenje.
20. Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering 2*(4), 42–47. Najdeno 24. maja 2016 na spletnem naslovu http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.3344&rep=rep1&type=pdf
21. Guid, N., & Strnad, D. (2015). *Umetna inteligenca* (1. izd.). Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko.
22. Gurney, K. (1997). *An introduction to neural networks*. Boca Raton: CRC Press.
23. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations 11*(1), 10–18. Najdeno 19. junija 2016 na spletnem naslovu http://www.kdd.org/exploration_files/p2V11n1.pdf
24. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2nd ed.). New York: Springer.
25. He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering 21*(9), 1263–1284. Najdeno 12. junija 2016 na spletnem naslovu http://ieeexplore.ieee.org.nukweb.nuk.uni-lj.si/stamp/stamp.jsp?tp=&arnumber=5128907
26. Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive Mixture of Local Experts. *Neural Computation, 3*, 79–87. Najdeno 6. marca 2016 na spletnem naslovu http://www.cs.toronto.edu/~fritz/absps/jjnh91.pdf
27. Kolyshkina, I., & Brookes, R. (2002, 22. oktober). Data mining approaches to modelling insurance risk. Najdeno 8. marca 2016 na spletnem naslovu http://docs.salford-systems.com/insurance4211.pdf

28. Kononenko, I. (2005). *Strojno učenje* (2. izd.). Ljubljana: Fakulteta za računalništvo in informatiko.
29. le Cessie, S., & van Houwelingen, J. C. (1992). Ridge Estimators in Logistic Regression. *Applied Statistics 41*(1), 191–201. Najdeno 22. junija 2016 na spletnem naslovu http://www.inf.unibz.it/dis/teaching/DWDM/project2010/LogisticRegression.pdf
30. Močivnik, P. (2010). *Razlaga zavarovalniških izrazov* (1. izd.). Ljubljana: Slovensko zavarovalno združenje.
31. Navodilo za izračun kazalnikov. *Uradni list RS* št. 36/2003.
32. Nisbet, R., Elder, J. F., & Miner, G. (2009). *Handbook of statistical analysis and data mining applications*. Amsterdam: Elsevier.
33. Palfy, M. (2009). *Nadomeščanje manjkajočih vrednosti s pomočjo regresijskega rotacijskega gozda* (doktorska disertacija). Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko.
34. Pyle, D. (1999). *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann.
35. Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks (ICNN) 1*, 586–591. Najdeno 8. junija 2016 na spletnem naslovu http://ieeexplore.ieee.org.nukweb.nuk.uni-lj.si/stamp/stamp.jsp?tp=&arnumber=298623
36. Sharma, S., & Osei-Bryson, K.-M. (2009). Role of Human Intelligence in Domain Driven Data Mining. V L. Cao, P. S. Yu, C. Zhang & H. Zhang (ur.), *Data Mining for Business Applications* (str. 53–61). New York: Springer.
37. Silipo, R., Adae, I., Hart, A., & Berthold, M. (2014). Seven Techniques for Dimensionality Reduction. Najdeno 4. junija 2016 na spletnem naslovu https://www.knime.org/files/knime_seventechniquesdatadimreduction.pdf
38. Smith, K. A., Willis, R. J., & Brooks, M. (2000). An Analysis of Customer Retention and Insurance Claim Patterns Using Data Mining: A Case Study. *The Journal of the Operational Research Society, 51*(5), 532–541. Najdeno 6. marca 2016 na spletnem naslovu http://www.jstor.org.nukweb.nuk.uni-lj.si/stable/pdf/254184.pdf
39. Sumathi, S., & Sivanandam, S. N. (2006). *Introduction to Data Mining and its Applications*. Berlin: Springer.
40. Šajn Juršič, N. (2014). *Analiza primernosti uporabe metod podatkovnega rudarjenja za modeliranje tveganja pri zavarovanju motornih vozil* (magistrsko delo). Ljubljana: Ekonomska fakulteta.
41. Takeuchi, I., Bengio, Y., & Kanamori, T. (2002). Robust regression with asymmetric heavy-tail noise distributions. *Neural Computation 14*(10), 2469–2496. Najdeno 14. junija 2016 na spletnem naslovu http://www.iro.umontreal.ca/~lisa/pointeurs/TR1198.pdf
42. Werner, G., & Modlin, C. (2016). *Basic Ratemaking* (5th ed.). b.k.: Casualty Actuarial Society.
43. Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Amsterdam: Morgan Kaufmann Publishers.
44. Xu, S., & Chen, L. (2008). A novel approach for determining the optimal number of hidden layer neurons for FNN's and its application in data mining. *Proceedings of the 5th International Conference on Information Technology and Applications (ICITA '08)*, 683–

686. Najdeno 3. junija 2016 na spletnem naslovu http://eprints.utas.edu.au/6995/1/02-au-xu.pdf
45. Zakon o davku od prometa zavarovalnih poslov. *Uradni list RS* št. 96/2005-UPB1.
46. Zakon o zavarovalništvu. *Uradni list RS* št. 93/2015.