UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Vybrané metódy klasifikácie dát - ich základné princípy a praktická implementácia

BAKALÁRSKA PRÁCA

2015 Jozef AGÁRDY

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Vybrané metódy klasifikácie dát - ich základné princípy a praktická implementácia

BAKALÁRSKA PRÁCA

Študijný program:	Ekonomická a finančná matematika
Študijný odbor:	9.1.9. Aplikovaná matematika (1114)
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	RNDr. Beáta Stehlíková, PhD.

Bratislava 2015 **Jozef AGÁRDY**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Jozef Agárdy
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Vybrané metódy klasifikácie dát - ich základné princípy a praktická implementácia
Selected data classification methods - their basic principles and practical implementation

Cieľ: 1. Naštudovať metódy klasifikácie dát (logistická regresia, rozhodovacie stromy, ďalšie metódy podľa vlastného výberu), vysvetliť ich základné myšlienky, implementáciu v softvéri R a ukázať názorné príklady ich použitia.

2. Použiť tieto metódy na zvolenom komplexnejšom príklade dát.

Vedúci: RNDr. Beáta Stehlíková, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.

Dátum zadania: 23.10.2014

Dátum schválenia: 23.11.2014 doc. RNDr. Margaréta Halická, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie Ďakujem svojej školiteľke RNDr. Beáte Stehlíkovej, PhD. za akceptovanie navrhnutej témy bakalárskej práce, ochotu, študijné materiály a príjemné konzultácie, pri ktorých mi vždy poskytla odbornú pomoc a podnetné pripomienky.

Abstrakt v štátnom jazyku

AGÁRDY, Jozef: Vybrané metódy klasifikácie dát - ich základné princípy a praktická implementácia [Bakalárska práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: RNDr. Beáta Stehlíková, PhD., Bratislava, 2015, 68 s.

V našej bakalárskej práci sa zaoberáme vybranými metódami na klasifikáciu dát, ich analýzou, porovnaním a implementáciou v prostredí R. Bližšie sa venujeme metódam logistickej regresie, rozhodovacím stromom a náhodným lesom, ktorých fungovanie a výhody popisujeme ako vo formálnom jazyku, tak na jednoduchom príklade a pomocou aplikácie v R. Získané poznatky v poslednej kapitole aplikujeme na zákaznícke dáta z vernostného programu, na ktorých porovnávame prediktívnu výkonnosť vysvetlených modelov.

Kľúčové slová: Logistická regresia, Rozhodovacie stromy, Náhodné lesy

Abstract

AGÁRDY, Jozef: Selected data classification methods - their basic principles and practical implementation [Bachelor Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: RNDr. Beáta Stehlíková, PhD., Bratislava, 2015, 68p.

In our final thesis related to the topic of data science and machine learning we are describing, analyzing and comparing selected classification methods as well as their implementation in statistical computing environment R. We are deeply considering methods like logistic regression, decision trees and random forests, their behavior and advantages we are defining using formal language, simple examples and application in R. In the last chapter we are applying models mentioned and knowledge gained to real-life customers data of a certain loyalty programme. Here we are comparing predictive efficiency as well as their possible usage to data where models have never been used.

Keywords: Logistic Regression, Decision Trees, Random Forests

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Zvyšujúca sa dostupnosť údajov v súčasnej informačnej spoločnosti viedla k potrebe vyvinutia uplatniteľného nástroja pre ich modelovanie a analýzu. Data mining a aplikované štatistické metódy sú vhodným prostriedkom pre získavanie potrebných vedomostí zo zozbieraných dát. Data mining, alebo dolovanie dát, možno definovať ako proces výberu, prieskumu a modelovania z veľkej databázy za účelom získania vzorov, segmentácie dát alebo predpovedania budúceho správania.

V posledných rokoch sa získavanie a ukladanie veľkých objemov dát stalo jednoduchším a lacnejším. To prinieslo revolúciu v spôsobe, akým ľudia pracujú, či už na poli vedy, alebo pri konkrétnej aplikácii metód v praxi, v denno-dennom živote. Typickým príkladom ich účelového zbierania, s ktorým sa stretávame každý deň, sú dáta z pokladničných blokov v CRM a vernostných programoch, nákupné transakcie z kreditných a debetných kariet, telekomunikačné dáta o hovoroch a využívaní telekomunikačných služieb. Tieto enormné množstvá dát vyžadujú sofistikovaný spôsob analýzy, výsledky analýz sú podkladom pre rozhodovanie manažmentu a ich zlá interpretácia môže spôsobiť značné finančné škody. Data mining a aplikované štatistické metódy preto hrajú čoraz väčšiu rolu v mnohých sférach vedy a výskumu, financií, v risk manažmente, v marketingu, v priemysle a plánovaní výroby a logistiky. Cieľom analýzy dát je zodpovedať položenú otázku, tá je zväčša smerovaná na predpovedanie budúcich skutočností, na predikciu, resp. klasifikáciu z existujúcich dát.

Termín klasifikácie dát je často asociovaný s pojmami ako machine learning (strojové učenie), rozpoznávaním vzorov a už s vyššie spomenutým data miningom (dolovaním v dátach). Klasifikačné a predikčné modely sa používajú na nájdenie potencionálne cenných vzorov v dátach, k predvídaniu výsledkov skúmaných udalostí. V súčasnosti sú k dispozícii početné techniky, od jednoduchých ako lineárna alebo logistická regresia, po komplexnejšie ako rozhodovacie stromy, náhodné lesy a neurónové siete. Komplexné modely zvyčajne dosahujú lepšiu prediktívnu výkonnosť, sú však neprehľadnejšie a preto nemôžu byť použité pre vysvetlenie predpovedí. Cieľom bakalárskej práce je priniesť prehľad týchto techník, porovnať ich chybovosť a ukázať uplatnenie na konkrétnych dátach z praxe.

V úvodnej časti každej z kapitol sa zaoberáme vysvetlením metód, ktoré budú po-

užité v ďalších častiach práce. Tieto časti sú spracované prevažne podľa publikácii [3] a [4], ich cieľom je priniesť matematické vysvetlenie fungovania modelu, načrtnúť jeho výhody a nevýhody. V duhej ďalšej časti kapitôl sú modely aplikované na vzorové dáta a ich výsledky sú zinterpretované. Cieľom poslednej kapitoly je nájsť, na základe poznatkov získaných v predchádzajúcich kapitolách, vhodný model podľa charakteru dát získaných z praxe.

1 Základné pojmy

1.1 Definícia klasifikačných úloh

V strojovom učení (machine learning) a v štatistike je za klasifikačnú metódu označená každá úloha, ktorá predikuje kategorické označenie tried. Klasifikácia dát prebieha na základe tréningovej množiny a daných zaradení do tried v jej klasifikačnom atribúte. Takto skonštruovaný model sa potom používa na predikciu nových pozorovaní v testovacej sade. Typickým a najilustratívnejším príkladom je klasifikácia emailových správ na príjemcom vyžiadané a nevyžiadané správy (spamy).

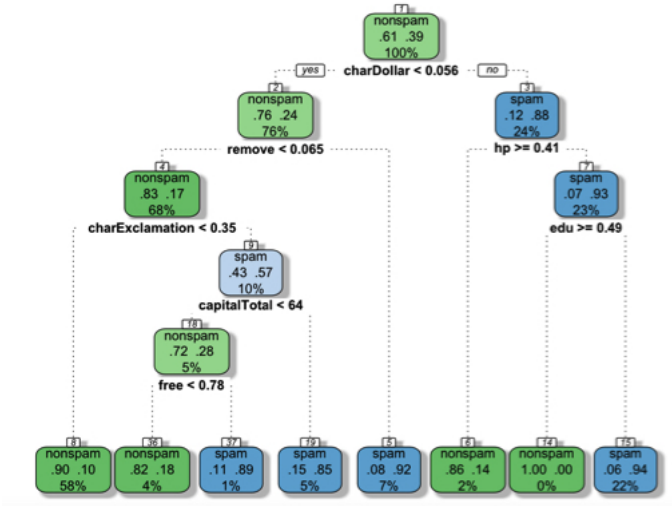
Pre pochopenie príkladu si potrebujeme vysvetliť význam dát o nevyžiadaných správach dostupných v knižnici kernlab v programovacom prostredí R [1]. V knižnici máme záznam o 4 601 emailových správach kategorizovaných na spam a nie-spam, každá zo správ ma 58 premenných, z toho 57 je prediktorov a faktorová informácia o tom, či ide o spam alebo nie. Bližšie si zreprodukujeme význam použitých premenných:

- prvých 48 premenných v datasete obsahuje frekvenciu slov (napr. Business) v každej zo správ,
- premenné začínajúce s „num“ reprezentujú frekvenciu číslovky v správe (napr. num555 reprezentuje frekvenciu čísla 555 v každej zo správ),
- v premenných 49 – 54 je obsiahnutá informácia o frekvencií interpunkčných znakov („;“ a pod.),
- premenné 55 až 57 reprezentujú priemernú, najdlhšiu a celkovú dĺžku slova napísaného kapitálkami.

Spomínanú faktorovú premennú pri klasifikačných úlohách predikujeme pomocou predikčných premenných. Jednoducho povedané, pomocou dát o frekvencií slov, čísloviek, interpunkčných znamienok a dĺžky slov budeme správy pomocou modelu rozdeľovať na vyžiadané a nevyžiadané. Príklad klasifikácie na základe vyššie spomenutých dát je vyobrazený na Obr. 1.

V terminolológii strojového učenia rozoznávame metódy s učiteľom (supervised learning) a metódy bez učiteľa (unsupervised learning).

Pri metódach s učiteľom, tak ako je už podľa názvu zrejmé, ide o učenie z tréningovej sady so správne zaklasifikovanými pozorovaniami, pre každé tréningové pozorovanie



Obr. 1: Príklad klasifikácie metódou rozhodovacích stromov na dátach o vyžiadených a nevyžiadaných správach

x_i máme prisluchájúcu odpoveď y_i. Na tomto vzťahu v tréningovej sade sa snažíme vybudovať model s cieľom predikovať budúce pozorovania. Klasifikácia, spolu s lineárnou regresiou a modernejšími metódami ako GAM a pod., je označená práve za metódu s učiteľom.

Naopak, komplikovanejším príkladom sú metódy bez učiteľa, kde vektoru tréningových pozorovaní x_i neprislúcha vektor závislej premennej y_i. Typickým príkladom metódy bez učiteľa je klasterová analýza alebo klasterizácia. Jej princípom je zaradenie jednotlivých pozorovaní x_i do skupín, klasterov, na základe vzorov v dátach, vzdialenosti alebo miery podobnosti. Najbežnejšou metódou klasterizácie je metóda k-priemerov. Predmetom našej bakalárskej práce je klasifikácia dát, budeme sa preto zaoberať iba metódami s učiteľom.

1.2 Základné pojmy

Vo všeobecnosti označíme termínom pozitívna takú udalosť, ktorú sme akceptovali, a naopak, udalosť, ktorú sme zamietli označíme za negatívnu [4].

Definujeme si ďalšie príbuzné pojmy:

- pravdivo pozitívna udalosť - udalosť, ktorú sme akceptovali správne,
- nepravdivo pozitívna udalosť - udalosť, ktorú sme akceptovali, avšak nesprávne,

- pravdivo negatívna udalosť - udalosť, ktorú sme správne zamietli,
- nepravdivo negatívna udalosť - udalosť, ktorú sme identifikovali nesprávne a zamietli ju.

Pojmy si pre jednoduchšie pochopenie intepretujeme na príklade:

- pravdivo pozitívna udalosť - nevyžiadaná správa správne označená za spam,
- nepravdivo pozitívna udalosť - správa, ktorá bola vyžiadaná, ale nesprávne označená za spam,
- pravdivo negatívna udalosť - správu sme správne označili za vyžiadanú,
- nepravdivo negatívna udalosť - správa nebola spamom, my sme ju však nesprávne označili za spam.

	Spam		
Test		+	-
	+	TP	FP
	-	FN	TN

Tabuľka 1: Základné pojmy klasifikačných metód, príklad

Senzitivita (Sensitivity)

$$P(pozitivny\ test \mid spam) = \frac{TP}{FN + TP}.$$

Špecifickosť (Specificity)

$$P(negativny\ test \mid nie\ spam) = \frac{TN}{FP + TN}.$$

Pozitívna klasifikovaná hodnota (Positive classified value)

$$P(spam \mid pozitivny\ test) = \frac{TP}{FP + TP}.$$

Negatívna klasifikovana hodnota (Negative classified value)

$$P(nie\ spam \mid negativny\ test) = \frac{TN}{FN + TN}.$$

Presnosť (Accuracy)

$$P(spravny\ pristup) = \frac{TN + TP}{FP + TP + FN + TN}.$$

1.3 Rozdelenie dát na tréningovú a testovaciu sadu

Na to, aby sme výsledky štatistickej analýzy vedeli aplikovať a posúdiť na nezávislých dátach, musíme našu databázu rozdeliť na dve separátne skupiny: na tréningové dáta (trainSpam) a na testovacie dáta (testSpam). V literatúre sa stretneme s rôznymi odporúčaniami na pomer tréningovej a testovacej sady [5].

Pomer tréningovej a testovacej sady dát určíme podľa počtu dát:

- v prípade, že máme veľký počet dát, rozdelíme ich na tri skupiny v pomeroch:
  - o 60 % pre tréningovú sadu,
  - o 20 % pre testovaciu sadu,
  - o 20 % určených na validáciu,
- v prípade, že máme stredne veľký súbor dát:
  - o 60 % pre tréningovú sadu,
  - o 40 % pre testovaciu sadu,
- v prípade, že máme malý počet dát:
  - o krížovú validáciu musíme spraviť,
  - o reportovať varovanie o veľkosti súboru dát.

Predpokladajme, že máme stredne veľký súbor dát, rozdelenie dát teda spravíme v pomere 60:40 v prospech tréningových dát (trainSpam), na ktorých model vystaviame.

Na rozdelenie využijeme príkaz *createDataPartition* dostupný v knižnici *caret*. Náhodné rozdeľovanie vzoriek sa vykonáva v rámci tried y (kde y je vzorka dát), pokiaľ y je faktorom, pri rozdelení sa príkaz snaží vyvážiť ditribúciu tried v rozhodovacom atribúte. V prípade y v tvare čísla je vzorka rozdelená do skupín podľa percentilu, rozdelovanie do jednotlivých vzoriek sa vykonáva v rámci týchto podskupín. [8].

```
library(caret)
data(spam)
trainIndex<-createDataPartition(spam$type,p=0.6,list=FALSE,
    times=1)
trainSpam<-spam[trainIndex,]
testSpam<-spam[-trainIndex,]
```

Načítame knižnicu caret a v nej dostupnú sadu dát o spamoch. Dáta rozdelíme podľa premennej type, ktorá obsahuje informáciu o tom, či správa je alebo nie je nevyžiadanou poštou (spamom alebo nie). Táto premenná je vstupom funkcie *createDataPartition*, rovnako je vstupom aj p, podiel vzorky prislúchajúcej k tréningovej sade, v našom prípade 60 %. Výstup nechceme mať v tvare list, preto ho nastavíme ako FALSE, chceme vytvoriť jednu partíciu, preto premennú times (počet partícií, ktoré chceme vytvoriť) nastavíme na hodnotu 1.

1.4 Pretrénovanie modelu (overfitting)

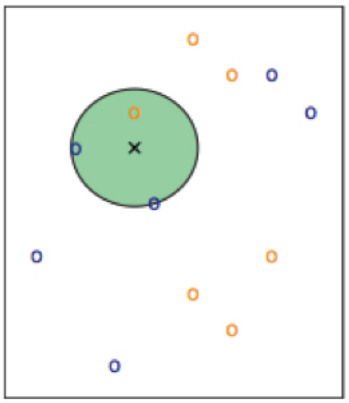
Model je pretrénovaným vtedy, keď namiesto príslušného vzťahu medzi prediktorom a závislou premennou popisuje náhodnú chybu. K pretrénovaniu dochádza, ak je model príliš zložitý, obsahuje príliš veľa prediktorov vzhľadom k počtu pozorovaní a potom má vo všeobecnosti slabú prediktívnu výkonnosť. Pretrénovanie je spôsobené tým, že rozhodovacie kritérium pre testovaciu sadu dát nie je rovnaké ako kritérium používané na posúdenie účinnosti modelu - model je vystavaný na tréningových dátach, jeho presnosť je však meraná na testovacej sade dát. Pri pretrénovaní si model namiesto učenia z tréningovej sady len zapamätá tréningové dáta, nie je tak použiteľný na testovacích dátach.

1.5 Príklad pretrénovania modelu

Pretrénovanie modelu si vysvetlíme na príklade najbližšieho suseda (k-nearest neighbor) [4]. Metóda sa snaží odhadnút podmienenú distribúciu Y, pre dané X a zaradiť pozorovanie do triedy podľa najvyššej odhadnutej pravdepodobnosti. Dané je prirodzené číslo K a testovacie pozorovanie x_0, potom metóda najbližšieho suseda najprv identifikuje K bodov v tréningovej sade, ktoré sú najbližšie k x_0, reprezentované N_0. Podmienenú pravdepodobnosť pre triedu j potom odhadneme ako zlomok bodov v N_0, ku ktorým zodpovedajúce hodnoty pre j sú:

$$P(Y = j \,|X = x_0\,) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j) \tag{1}$$

Metóda najbližšieho suseda potom zaklasifikuje x_0 do prislúchajúcej triedy s najvyššou pravdepodobnosťou. Metódu najbližšieho suseda ilustrujeme na jednoduchom príklade.



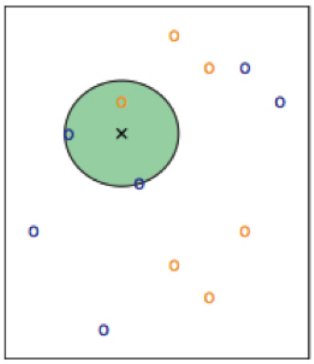
Obr. 2: Ilustratívny príklad pre metódu najbližšieho suseda [4]

Jednoduchý tréningový dataset pozostávajúci zo šiestich pozorovaní, ktoré označíme modrou farbou a šiestich pozorovaní klasifikovaných oranžovou farbou znazorníme graficky v Obr. 2. Naším cieľom je zaklasifikovať bod znázornený čiernym krížom do jednej z dvoch spomenutých tried. Pre ilustratívnosť si vyberieme hodnotu $K = 3$. Metóda najbližšieho suseda potom pre $K = 3$ najprv identifikuje 3 pozorovania, ktoré sú najbližšie k cieľu našej predikcie a toto susedstvo, hranicu, znázorní zeleným kruhom. V kruhu sa nachádzajú dve pozorovania znázornené modrou farbou a jedno pozorovanie s farbou oranžovou. Odhadnutá pravdepodobnosť, že kríž je z triedy modrých pozorovaní je teda $\frac{2}{3}$, respektíve, pravdepodobnosť, že kríž je z triedy oranžových pozorovaní $\frac{1}{3}$. Vyberieme triedu s vyššou pravdepodobnosťou, pomocou metódy najbližšieho suseda je náš kríž z triedy modrých pozorovaní.

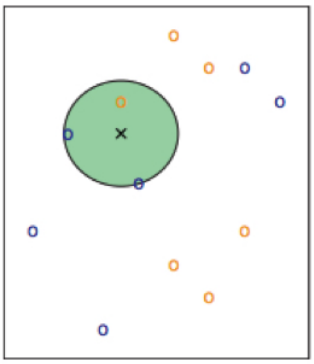
Voľba K je kľúčová pre charakter výstupu, ktorý dostaneme z metódy nabližšieho suseda. Motiváciou na vysvetlenie metódy najbližšieho suseda bolo znázorniť konkrétny príklad pretrénovania dát. Obr. 3 a Obr. 4 reprezentujú dva prípady použitia metódy najbližšieho suseda pre $K = 1$ (vľavo) a $K = 100$ (vpravo).

Pre $K = 1$ je hranica príliš flexibilná a prispôsobená naším dátam, model si v tomto prípade len zapamätal tréningové dáta a aj napriek predikčným schopnostiam, nie je použiteľný na dátach testovacích. Ak K rastie, metóda sa stáva menej flexibilnou a hranica sa stáva viac lineárnou. Pre úplnosť, hodnota chyby pre $K = 1$ predstavuje 0,1695 a pre $K = 100$, naopak, 0,1925. Ako sme však spomenuli vyššie, model pre príklad $K = 1$ javí známky pretrénovania.

Ako vhodným kompromisom medzi znázornenými príkladmi sa javí použitie $K = 10$.

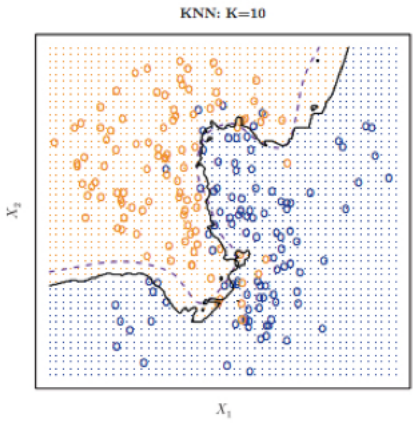


Obr. 3: Metóda najbližšieho suseda pre K = 1 [4]



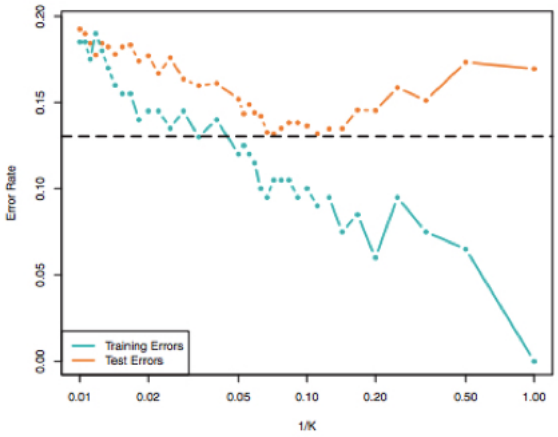
Obr. 4: Metóda najbližšieho suseda pre K = 100 [4]

Hranica, medzi zaklasifikovanými premennými je zvýraznená čiernou farbou.



Obr. 5: Metóda najbližšieho suseda pre K = 10 [4]

Vzťah medzi chybou modelu na tréningových dátach a chybou na testovacích dátach je nepatrný. Pre $K = 1$ sme dosiahli tréningovú chybu 0 %, ale naopak, chyba pri testovacích dátach môže byť výrazne vyššia. Vo všeobecnosti, ak používame flexibilnejšie klasifikačné metódy, tréningová chyba klesá, testovacia však nemusí. Do Obr. 6 je zakreslené porovnanie chýb metódy najbližšieho suseda pre tréningové a testovacie dáta ako funkciu $\frac{1}{K}$. Ak hodnota $\frac{1}{K}$ rastie, metóda sa stáva viac flexibilnou. Tréningová chyba konštatne klesá s narastájucou hodnotou $\frac{1}{K}$ s narastajúcou flexibilitou. Naopak, testovacia chyba nadobúda charakteristický tvar U, najprv klesá a potom znova narastá so zvyšujúcou sa flexibilitou (platí pre K aspoň 10). Úroveň flexibility je teda kľúčová pre nastavenie akejkoľvek klasifikačnej úlohy.



Obr. 6: Porovnanie chyby pre tréningové a testovacie dáta [4]

1.6 Krížová validácia

Vyššie sme spomenuli rozdiel medzi chybou na tréningových dátach a na testovacích. Krížová validácia validácia (cross-validation) je technika na ohodnotenie výkonnosti algoritmu použitého na nový dataset, na ktorom nebola trénovaná [3]. Metóda testuje, či sa výsledky štatistickej analýzy dajú zovšeobecniť na nezávislý súbor dát. Toto sa deje rozdeleným datasetu na dve podmnožiny, tréningovú a testovaciu. Pretože krížová validácia nepoužíva všetky dáta na výstavbu modelu, je to najčastejšie používanou metódou na zabránenie pretrénovania modelu počas tréningu.

V každom kole krížovej validácie sa pôvodný dataset náhodne rozdelí v predefinovanom pomere na tréningovú a testovaciu sadu. Na fitovanie modelu sa potom používa tréningový set dát, na vyhodnotenie výkonnosti naopak testovací dataset. Tento postup sa potom niekoľkokrát opakuje, priemerná hodnota chyby krížovej validácie sa používa ako výkonnostný indikátor.

Proces krížovej validácie vieme definovať do štyroch kľúčových bodov:

- rozdelenie dát na tréningovú a testovaciu sadu,
- výstavba modelu na tréningových dátach,
- vyhodnotenie modelu na testovacej sade dát,
- kalkulácia premernej chyby modelu a opakovanie procesu.

Metódy krížovej validácie:

- **K-fold:** rozdelenie dát do náhodne vybraných podmnožín o približne rovnakej

veľkosti. Jedna z podmnožín slúži na overenie modelu nafitovaného pomocou zvyšných podmnožín. Tento postup sa opakuje k-krát tak, že každá z podmnožín sa používa práve raz pre validáciu.

- **Rozdeľovanie (Holdout):** najjednoduchší prípad krížovej validácia, dáta sa rozdelia do dvoch podmnožín - do tréningovej a testovacej sady. Model sa fituje len pomocou tréningovej sady, pričom závislá premenná sa predikuje na základe sady testovacej. Na ohodnotenie modelu sa používa priemerná absolútna chyba na testovacích dátach.
- **Vynechanie (Leave-one-out):** rozdelí dáta do k-podmnožín, kde k je rovné počtu pozorovaní v datasete. Model je fitovaný na všetkých podmnožinách s výnimkou jednej, pričom predikuje práve na vynechanej množine.
- **Opakované náhodné delenie do podmnožín (repeated random subsampling):** využíva simuláciu Monte Carlo opakovane na náhodne delenie dát do skupín a agregovanie výsledkov po tom, čo prebehnú ostatné simulácie.

Krížová validácia nájde svoje použitie pri výbere vhodného prediktoru do modelu, výberu vhodnej funkcie na klasifikáciu, výberu vhodných parametrov do funkcie určenej na klasifikáciu alebo na porovnanie rôznych prediktorov.

2 Logistická regresia

Jednoduchým príkladom pre problém logistického modelovania je kategorizovanie spamov v emailovej schránke. Predpokladajme jednoduchý model, v ktorom naším ukazovateľom pre to, či sa jedná o spam alebo nie, bude počet slov „zadarmo“ v správe – túto premennú nazveme „počet“. Pravdepodobnosť toho, že správa je spam pri zadanom počte slov môžeme definovať ako [2]:

$$P(spam \ = \ ano \mid pocet)$$

Táto pravdepodobnosť je definovaná na intervale $[0, 1]$. Môžeme tak určiť hranicu, nad ktorou budeme správy považovať za spamy. Pravdepodobnosť definujeme [3] vo všeobecnom tvare:

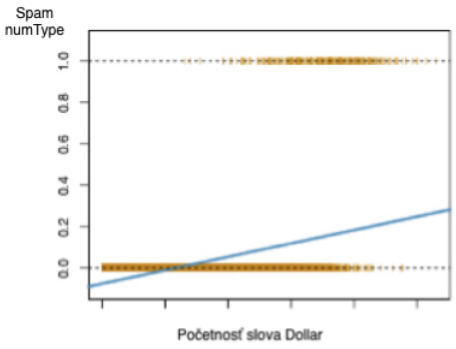
$$p(X) = P(Y \ = \ 1 \mid X)$$

Pomocou prediktora X teda chceme odhadovať premennú Y. Skôr, ako začneme s modelovaním, potrebujeme definovať funkciu tak, aby nám dávala výstup z intervalu $[0, 1]$. Prečo si však nevystačíme s lineárnou funkciou? Kategorické premenné z nášho jednoduchého modelu pojednávajúcom o spamoch v emailovej schránke môžeme jednoducho prepísať na binárne premenné pre lineárnu regresiu ako:

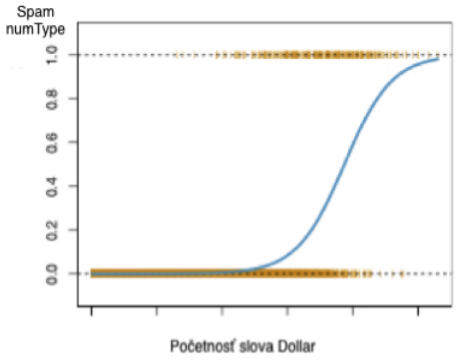
$$Y = \begin{cases} 0, & \text{ak nie je spam} \\ 1, & \text{ak je spam} \end{cases}$$

Ako môžeme vidieť na Obr. 7, pri jej použití budú niektoré z predikovaných hodnôt mimo intervalu $[0, 1]$ a nebudú teda interpretovateľné ako pravdepodobnosti. Niektoré z odhadnutých parametrov teda môžu vyjsť záporné, ako je to na nižšie vyobrazenom ilustratívnom Obr. 8, v prípade logistickej funkcie sú všetky odhadnuté parametre z intervalu $[0, 1]$.

Funkcií s výstupom v tomto intervale je samozrejme viac, pre účely nášho modelu budeme používať logistickú funkciu pre pravdepodobnosť $p(X)$. Logistická regresia je obľúbenou a často používanou metódou. Funkciu navrhol matematik pôsobiaci na Univerzite v Ghente, Pierre François Verhulst. Výhodou funkcie bolo, že jej počiatočný rast bol približne exponenciálny, približujúc sa k hodnote 1 (k platnosti) sa jej rast spomaľoval, až v limite zastavil na horizontálnej čiare. Výstup pri logistickej regresii je



Obr. 7: Lineárna regresia s binárnymi premennými [3]



Obr. 8: Jednoduchá logistická regresia [3]

dichotomický, nadobúda teda hodnoty „áno“ alebo „nie“. V kapitole najskôr rozoberáme jednoduchý model s jednou vysvetľujúcou premennou a jedným výstupom, v ďalších častiach kapitoly model s viacerými premennými a tiež len jedným výstupom.

2.1 Logistická regresia s jednou premennou

Definujeme funkciu logistickej regresie pre pravdepodobnosť $p(X)$ v tvare [4]:

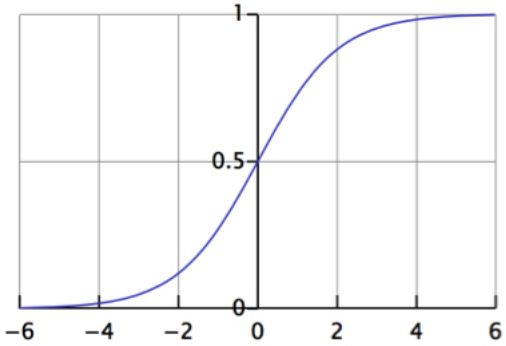
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 - e^{\beta_0+\beta_1 X}}, \tag{2}$$

kde X je nezávislá premenná a koeficienty β_0 a β_1 sú parametrami. Koeficient určuje mieru rastu krivky, v stručnosti teda:

- ak je $\beta_1 > 0$, potom $p(X)$ narastá s narastajúcou hodnotou X,
- ak je $\beta_1 < 0$, potom $p(X)$ klesá s narastajúcou hodnotou X,
- limitným prípadom je, ak $\beta_1 \to 0$, potom krivkou logistickej funkcie sa blíži k horizontálnej čiare.

2.2 Logistická regresia s viacerými premennými

V krátkosti si rozoberieme princíp fungovania logistickej regresie s viacerými premennými. Počet prediktorov je n a môžeme ich zapísať ako $X = (X_1, X_2, \ldots, X_n)$. Pre náš prípad spamov v emailovej schránke, môže takýmto prediktorom byť frekvencia slov v emailovej schránke – vysoká frekvencia slov ako napr. „order“, „credit“, „free“



Obr. 9: Priebeh logistickej funkcie

indikuje vysokú pravdepodobnosť, že správa je spamom.

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X+\ldots+\beta_n}.$$

Princíp fungovania si neskôr vysvetlíme na konkrétnom príklade v softvéri R, znova použijeme dáta z knižnice kernlab o emailových správach.

2.3 Metóda maximálnej vierohodnosti

V modeli nám vystupujú neznáme hodnoty $\beta = (\beta_1,\ \beta_2,\ \ldots, \beta_n)$, ktoré potrebujeme odhadnúť na základe testovacej sady dát. Pre každý tréningový bod z dát, máme vektor funkcií $x = (x_1,\ x_2,\ \ldots, x_n)$ a pozorovaných tried $y = (y_1,\ y_2,\ \ldots, y_n)$. Pravdepodobnosť triedy bude je buď p, ak $y_i = 1$, alebo $1-p$, ak $y_i = 0$. Na odhadovanie týchto parametrov je najčastejšie používanou metódou metóda maximálnej vierohodnosti [12].

Funkciu vierohodnosti môžeme vo všeobecnosti zapísať ako:

$$L(\beta_0,\ \beta) = \prod_{i=1}^{n} p(x_i)^{y_i}(1-p(x_i))^{1-y_i}.$$

Pre náš prípad logistickej funkcie uvedieme tvar log-vierohnodnosti:

$$l(\beta_0,\ \beta) = \sum_{i=1}^{n} y_i \log p(x_i) + (1-y_i)\log(1-p(x_i)) = \sum_{i=1}^{n} \log(1-p(x_i)) + \sum_{i=1}^{n} y_i \log(\frac{p(X)}{1-p(X)}).$$

S úpravami sme však ešte neskončili. Formulu 2 môžeme drobnými úpravami upraviť na tvar:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1-e^{\beta_0+\beta_1 X}},$$

získaný tvar zlogaritmujeme a dosadíme do funkcie log-vierohodnosti:

$$\sum_{i=1}^{n} -\log(1+e^{\beta_0+\beta_1 x_i}) + \sum_{i=1}^{n} y_i \log(^{\beta_0+\beta_1 x_i}).$$

Typicky, ak chceme funkciu maximalizovať, zderivujeme ju podľa každého z jej parametrov a tieto derivácie dáme do rovnosti s nulou. Po zderivovaní podľa β_j nadobúda vierohodnostná funkcia tvar:

$$\frac{\partial f}{\partial \beta_j}(\beta) = -\sum_{i=1}^{n} \frac{1}{1+e^{\beta_0+\beta_1 x_i}} e^{\beta_0+\beta_1 x_i} x_{ij} + \sum_{i=1}^{n} y_i x_{ij} = \sum_{i=1} x_{ij}(y_i - \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}).$$

Následné výraz upravíme podľa vzťahu 2 do tvaru:

$$\sum_{i=1}^{n}(y_i - p(x_i, \beta_0, \beta_1)x_{ij}).$$

Túto nelineárnu rovnicu nie sme schopný porovnať s nulou a vyriešiť presne. Vieme však nájsť len jej približné numerické riešenie, napríklad Newtonovou metódou. Existuje mnoho metód na riešenie úlohy numerickej optimalizácie, riešenie takýchto optimalizačných úloh ale nie je predmetom našej bakalárskej práce.

2.4 Zostavenie modelu v softvéri R

Ako sme už spomenuli vyššie, jednou z aplikácii logistického modelu je kategorizovanie emailových sprav podľa ich vyžiadanosti, na nevyžiadané (spamy) a vyžiadané správy. Pomocou knižnice kernlab v štatistickom softvéri R vytvoríme model, ktorým nám určí, či daná správa je spamom alebo nie.

```
library(kernlab)
```

Dáta budeme rozdeľovať na tréningovú a testovaciu sadu dát. Využijeme na to príkaz *createDataPartition* dostupný v knižnici *caret*, knižnicu načítame:

```
library(caret)
data(spam)
set.seed(3435)
```

Načítame si knižnicu *kernlab* a spomínané dáta. Máme tak záznam o 4 601 emailových správach kategorizovaných na spam a nie-spam, každá zo správ ma 58 premenných, z toho 57 je prediktorov a faktorová informácia o tom, či je o spam alebo nie, bude naša predikovaná premenná Y.

Vysvetlíme si význam použitých premenných:

- prvých 48 premenných v datasete obsahuje frekvenciu slov (napr. Business) v každej zo správ,
- premenné začínajúce s „num“ reprezentujú frekvenciu číslovky v správe (napr. num555 reprezentuje frekvenciu čísla 555 v každej zo správ),
- v premenných 49 – 54 je obsiahnutá informácia o frekvencií interpunkčných znakov („;“ a pod.),
- premenné 55 až 57 reprezentujú priemernú, najdlhšiu a celkovú dĺžku slova napísaného kapitálkami.

Na to, aby sme výsledky štatistickej analýzy vedeli aplikovať a posúdiť na nezávislých dátach, musíme našu databázu rozdeliť na dve separátne skupiny: na tréningové dáta (trainSpam) a na testovacie dáta (testSpam). Rozdelenie dát spravíme v pomere 60:40 v prospech tréningových dát, budú slúžiť na odhad parametrov β pomocou metódy maximálnej vierohodnosti.

```
trainIndex<-createDataPartition(spam$type,p=0.6,list=FALSE,
    times=1)
trainSpam<-spam[trainIndex,]
testSpam<-spam[-trainIndex,]
```

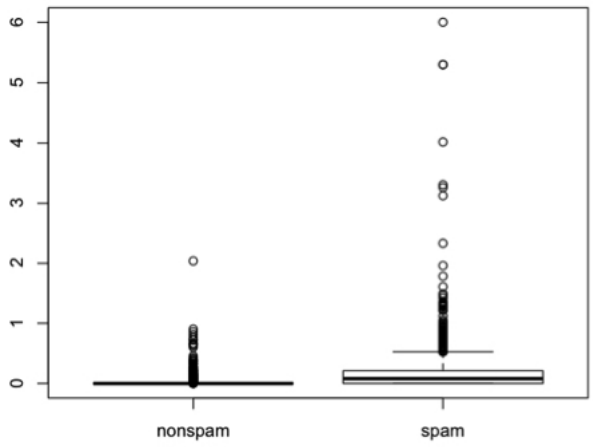
Vidíme, že v našom tréningovom súbore sa nachádza 1 088 správ označených ako spam.

```
table(trainSpam$type)


##
## nonspam    spam
##    1673    1088
```

Pozrieme sa, ako závisí typ správy na frekvencií slova Dollar v správe. Na Obr. 10 si môžeme všimnúť, že výskyt slova dolár je pri spame častejší, ako v správe, ktorá spamom nie je. Pomocou prediktora CharDollar teda vybudujeme náš model na klasifikáciu správ.

Faktorovú premennú spam/nie-spam najskôr prevedieme na numerickú s hodnotami 0 resp. 1 (0 pre vyžiadanú, 1 pre nevyžiadanú správu - spam),



Obr. 10: Boxplot pre Type a CharDollar

```
trainSpam$numType=(as.numeric(trainSpam$type))-1
```

a zostavíme model glm (Generalized linear model), kde v premennej funkcie family určíme, že ide o logistický model. V jednoduchom modeli odhadujeme našu numerickú premennú numType pomocou prediktora CharDollar.

```
fit<-glm(numType~charDollar,data=trainSpam,family=binomial)
```

Použili sme funkciu glm, ktorej argumenty si vysvetlíme:

- prvým argumentom je symbolický zápis modelu, formula, ktorú sa snažíme zostrojiť (v našom prípade ide o závislosť numType od premennej charDollar),
- argumentom data načítavame zdroj dát pre náš model,
- family popisuje rozdelenie z ktorého pochádza chyba (error) a obsahuje link na funkciu použitú v modeli (v našom prípade family=binomial s odkazom na funkciu link="logit", možnosť použiť aj „gaussian", „Gamma", „inverse.gaussian", „poisson" a pod.).

Funkcia je v softvéri R definovaná ako:

```
glm(formula,family=binomial,data,weights,subset,na.action,
    start=NULL,etastart,mustart,offset,control=list(...),model=
    TRUE,method="glm.fit",x=FALSE,y=TRUE,contrasts=NULL,...)
```

Žiadny z jej zvyšných argumentov v modeli nevyužijeme, preto s nimi nebudeme na tomto mieste zaoberať. Na klasifikáciu použijeme všeobecnú funkciu predict na

predpovedanie výsledkov rôznych fitovaných funkcií:

- prvým argumentom funkcie je object, ktorý obsahuje model fit,
- argument data so zdrojom dát,
- argument response zabezpečí, že funkcia vráti predikovanú pravdepodobnosť.

```
proba_train<-predict(fit,data=trainSpam,type="response")
```

Za spam označíme iba tie správy, pri ktorých je pravdepodobnosť (predpovedaný výsledok) ostro vyššia ako 95%.

```
predicted_spam_train<-as.numeric(proba_train>0.95)
table(predicted_spam_train,trainSpam$numType)
##
## predicted_spam_train    0    1
##                    0 1660  910
##                    1   13  178
```

Celkom 2 570 správ sme zakategorizovali ako vyžiadané:

- 1 660 správ zo všetkých vyžiadaných sme správne kategorizovali (pravdivo negatívne) ako žiaduce (65 %),
- 910 správ sme nesprávne (nepravdivo negatívne) označili za spamy, aj keď nimi neboli (35 %).

Z výpisu tiež môžeme vidieť, že z 191 emailových správ označených modelom za spam, spamom bolo 178 (93 %). Naopak, nesprávne sme identifikovali 13 správ (7 %), ktoré sme označili za žiaduce, aj keď nimi neboli. Rovnaký postup zopakujeme aj na našich testovacích dátach.

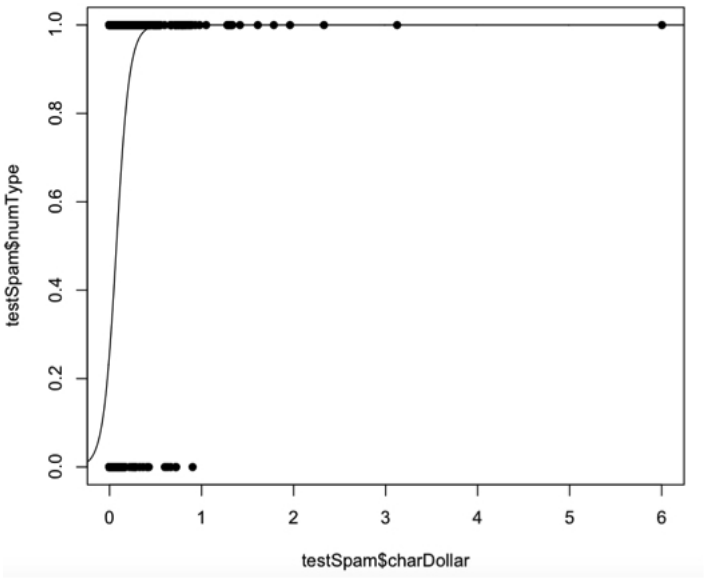
```
testSpam$numType<-(as.numeric(testSpam$type))-1
proba_test<-predict(fit,newdata=testSpam,type="response")
predicted_spam_test<-as.numeric(proba_test>0.95)
table(predicted_spam_test,testSpam$numType)
##
## predicted_spam_test    0    1
```

```
##                    0  1104   595
##                    1    11   130
```

Na našej testovacej sade dát sme celkom 1 699 správ zakategorizovali ako vyžiadané:

- 1 104 správ zo všetkých vyžiadaných sme správne (pravdivo negatívne) zakategorizovali ako žiaduce (65 %),
- 595 správ sme nesprávne (nepravdivo negatívne) označili za spamy, aj keď nimi neboli (35 %).

Z výpisu môžeme vidieť, že 141 emailových správ označených modelom za spam, spamom aj bolo 130. Nesprávne identifikovaných ako spam bolo 11 správ. Tvar logistickej funkcie je vyobrazený na Obr. 11.



Obr. 11: Tvar logistickej funkcie pre náš príklad

Pomocou modelu sme odhadli hodnoty koeficientov β.

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0587     0.0496  -21.35   <2e-16 ***
## charDollar   14.1235     0.7817   18.07   <2e-16 ***
```

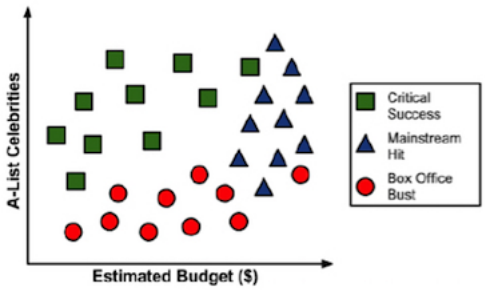
Ukážeme si, ako pomocou týchto koeficientov odhadnúť pravdepodobnosť pre hodnotu char$dollar rovnú 0,109. Odhadujeme teda pravdepodobnosť pre správu, v ktorej početnosť slova dolár je 10,9 %.

$$\frac{e^{\beta_0+\beta_1 X}}{1-e^{\beta_0+\beta_1 X}} = \frac{e^{-1,0587+14.1235*0,109}}{1-e^{-1,0587+14.1235*0,109}}$$

Pravdepodobnosť, že správa s početnosťou slova dolár rovnou 10,9 % je spamom, odhadneme na 61,79 %.

3 Rozhodovacie stromy

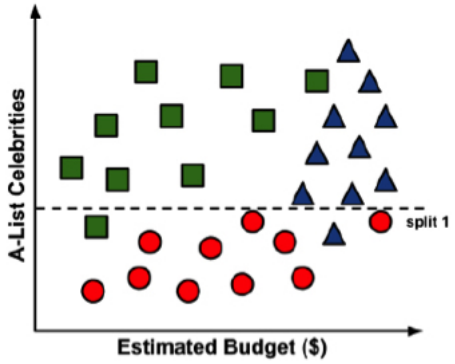
Ako je podľa názvu evidentné, rozhodovacie stromy budujú model v tvare stromu. Model je budovaný na slede logických rozhodnutí porovnateľných s vývojovým diagramom – jednotlivé body, ktoré obsahujú rozhodnutie budeme nazývať rozhodovacie uzly. Rozhodovacie stromy sú budované na heuristickom princípe nazývanom rekurzívne delenie a jeho začiatok je v koreňovom uzle predstavujúcom kompletný súbor dát. Model si vysvetlíme na jednoduchom príklade filmov Hollywoodu [9], príklad je len ilustračný a neobsahuje konkrétne dáta. Filmy sú rozdelené na tri kategórie: mainstreamové hity (mainstream hits), voľba kritika (critic's choice) a prepadáky (box office bust). Na základe našich premenných, ktorými sú počet celebrít a odhadnutý rozpočet na natočenie filmu, budeme vysvetľovať, prečo spadli do práve jednej zo spomenutých kategórií. Najskôr si zobrazíme závislosť týchto dvoch premenných v grafe na Obr. 12 - farebne a rozličnými symbolmi zvýrazníme charakter filmu.



Obr. 12: Zobrazenie filmov podľa ich charakteristík: odhadnutého rozpočtu a počtu celebrít [9]

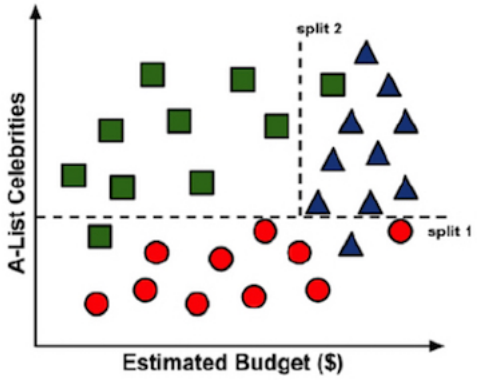
Pomocou metódy rekurzívneho delenia sme rozdelili funkciu označujúcu počet celebrít na skupiny. Čiara na Obr. 13 rozdelila dáta na dve skupiny − s vysokým a nízkym počtom celebrít hrajúcich vo filme.

Rovnako môžeme spraviť rozdelenie podľa funkcie označujúcej odhadovaný rozpočet filmov. Vertikálna čiara na Obr. 14 rozdelila dáta na − na filmy s vysokým rozpočtom a na filmy bez vysokého rozpočtu. Dáta sú však na Obr. 14 rozdelené do troch skupín. Skupina vpravo hore sa vyznačuje vysokým rozpočtom a vysokým počtom účinkujúcich celebrít. Môžeme si všimnúť, že táto skupina je zložená prevažne z filmov označených za mainstreamové hity, iba 1 film z 10 je z kategórie voľby kritika. Rovnako skupina



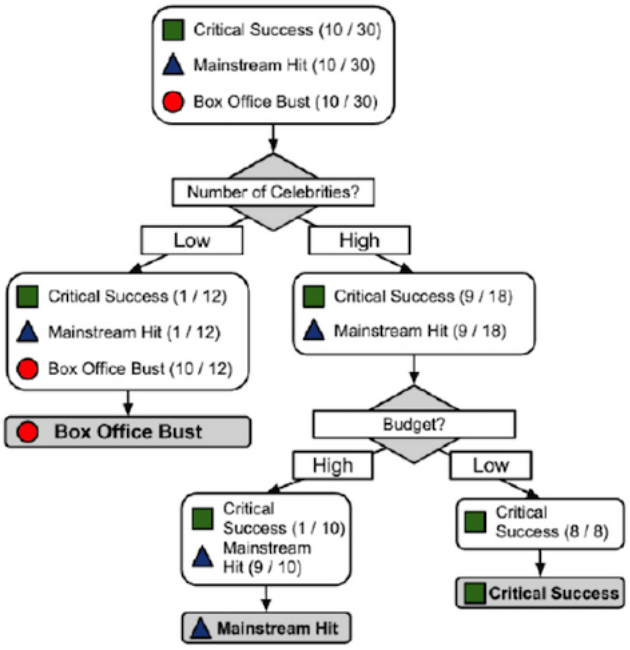
Obr. 13: Rozdelenie dát na skupiny podľa počtu celebrít [9]

filmov s nízkym zastúpením celebrít obsahuje prevažne filmy označené za prepadáky (Box Office Bust).



Obr. 14: Rozdelenie dát na skupiny podľa odhadnutého rozpočtu [9]

Najrozhodujúcejšou premennou pri výstavbe modelu je počet celebrít. V našom prípade ide o faktorovú premennú a rozdelí model do prvej sady dát – buď na filmy s nízkym počtom alebo filmy s vysokým počtom celebrít. Na rozdiel od logistickej regresie, modelovaná premenná nemusí nadobúdať len dve hodnoty 0 a 1. Ako sme už spomenuli vyššie, ak vo filme účinkuje nízky počet celebrít, ide s najväčšou pravdepodobnosťou o prepadák (Box Office Bust). Dáta o rozpočte a počte celebrít, z ktorých sú zostrojené grafy vyššie, aj keď ich nepoznáme, považujme pre nás za tréningové. Na základe tých dát sme si kvantifikovali pravdepodobnosť, že film je prepadákom (Box Office Bust) na 83 %. Pre strom zostrojený na tréningových dátach považujme túto pravdepodobnosť za postačujúcu. V prípade, že vo filme účinkoval vysoký počet celebrít, pokračujeme v rekurzívnom delení a pozrieme sa na premennú rozpočet. 90 %



Obr. 15: Výsledný rozhodovací strom [9]

filmov s vysokým rozpočtom sa stalo mainstreamovými hitmi, naopak všetky filmy s vysokým počtom celebrít a s nízkym rozpočtom sa stali voľbou kritika.

3.1 Výstavba modelu, kritérium entropie

Metóda rozhodovacích stromov rozkladá celistvý dataset rekurzívnym delením na menšie a menšie podmnožiny, konečným výsledkom je tak strom s rozhodovacími a koncovými uzlami. Uzol s najväčšou výpovednou hodnotou zodpovedá najlepšiemu prediktoru a je nazvaný rozhodovacím uzol. Rozhodovacie stromy zvládnu ako kategorizované, tak aj číselné údaje. Naším modelom budeme historicky predpovedať pravdepodobnosť prežitia, resp. neprežitia ľudí na palube RMS Titanic, lode plaviacej sa zo Southamptnu do New Yorku, ktorá sa tragicky potopila po zrážke s ľadovcom v roku 1912. Predpoveď budeme zakladať na dátach dostupných o pasažieroch. Model budeme budovať a vysvetľovať na dátach Titanic implementovaných v R v balíku datasets [1]. Takýchto dátových sád je viac, pre nami opisovanú klasifikáciu je však najvhodnejšia sada z balíka datasets s kategorickými premennými. Ide o štvordimenzionálne pole zložené z 2 201 záznamov so štyrmi kategorickými premennými:

- pohlavím (Sex), rozdeleným na ženy (Female) a mužov (Male),

- vekom (Age), s kategóriami dospelý (Adult) a dieťa (Child),
- ekonomickým statusom (Class) s troma triedami a posádkou (1st, 2nd, 3rd a Crew),
- a informáciou o prežití tragédie (Survived).

V Tabuľke 2 sú pre prehľadnosť vyobrazené dáta z datasetu:

Class	Sex	Age	Survived	Freq	Survived	Freq
1st	Male	Child	No	0	Yes	5
2nd	Male	Child	No	0	Yes	11
3rd	Male	Child	No	35	Yes	13
Crew	Male	Child	No	0	Yes	0
1st	Female	Child	No	0	Yes	1
2nd	Female	Child	No	0	Yes	13
3rd	Female	Child	No	17	Yes	14
Crew	Female	Child	No	0	Yes	0
1st	Male	Adult	No	118	Yes	57
2nd	Male	Adult	No	154	Yes	14
3rd	Male	Adult	No	387	Yes	75
Crew	Male	Adult	No	670	Yes	192
1st	Female	Adult	No	4	Yes	140
2nd	Female	Adult	No	13	Yes	80
3rd	Female	Adult	No	89	Yes	76
Crew	Female	Adult	No	3	Yes	20

Tabuľka 2: Skladba dát pre dataset Titanic [1]

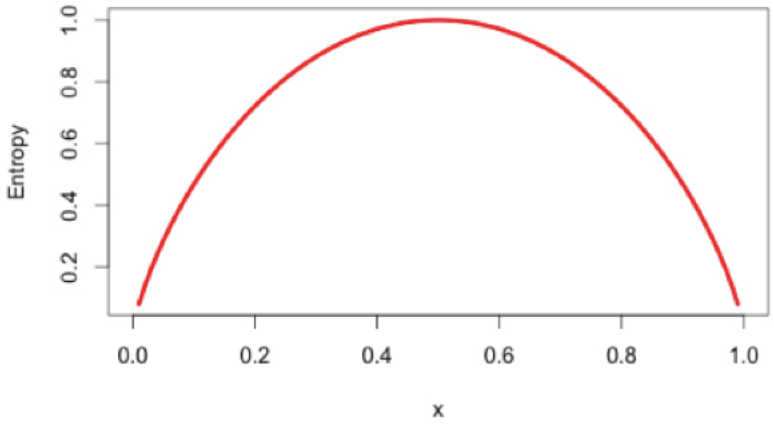
3.2 Algoritmus a použité kritérium

Na výstavbu modelu budeme používať metódu ID3 od J. R. Quinlana. Rozhodovací strom je budovaný z koreňového uzla smerom dole a zahŕňa rekurzívne rozdeľovanie skupín na podskupiny s homogénnymi hodnotami. Algoritmus ID3 používa kritérium

entropie na výpočet homogenity vzorky, v prípade, že vzorka je úplne homogénna, entropia je nulová, v opačnom prípade, ak je vzorka rovnomerne rozdelená, tak entropia je jednotková. V modeli budeme využívať dva typy entropie: entropiu pre tabuľky s jedným atribútom a entropiu využívajúcu tabuľku s frekvenciami dvoch atribútov. Nech náhodná premenná X nadobúda jednu z dvoch hodnôt a alebo b, pravdepodobnosť nadobudnutia hodnoty a označme ako $p(a)$, resp. hodnoty b ako $p(b)$, pričom platí, že $p(b) = 1 - p(a)$. Entropiu zadefinujeme ako:

$$Entropia(a, b) = -p(a)\log_2(p(a)) - p(b)\log_2(p(b))$$

a jej priebeh je znázornený na Obr. 16 nižšie.



Obr. 16: Priebeh funkcie entropie

Funkcia entropie dosahuje svoje maximum, ak je pravdepodobnosť $p(a) = \frac{1}{2}$, čo znamená, že pravdepodobnosť, že náhodná premenná nadobudne hodnotu a je rovná $p(a) = \frac{1}{2}$, resp. pre druhú z hodnôt $p(b) = \frac{1}{2}$. Entropia je rovná nule (čo je aj jej minimum), ak pravdepodobnosť nadobudnutia jednej z hodnôt je rovná $p = 1$.

Ak náhodná premenná nadobúda n rôznych hodnôt s pravdepodobnosťami $p_1, p_2, \ldots, p_n$, tak entropiu definujeme ako:

$$\sum_{i=1}^{n} p_i \log_i(p_i) \tag{3}$$

Ak počítame entropiu pre dáta, v situácii, keď sú tieto dáta rozdelené do n kategórii, tak p_i vo vzťahu 3 dostávajú relatívne početnosti dát v jednotlivých triedach.

Ďalšie používané kritéria [10]

- **Koeficient Gini (Gini coefficient)**

$$\sum_{n=1}^{N} p_n(1-p_n)$$

Koeficient Gini je definovaný ako miera celkovej variance naprieč triedami N. Hodnota p_n predstavuje podiel pozorovaní z k-tej kategórie k celkovému počtu pozorovaní – súčtom všetkých hodnôt p_n je teda $\sum_{n=1}^{N} p_n = 1$. Hodnota indexu Gini je malá, ak ak všetky p_n blízko k 1 alebo k 0. Z tohto dôvodu je index označovaný za mieru čistoty uzlov – malá hodnota znamená, že uzol obsahuje vyšší podiel pozorovaní z jednej kategórie.

- **Pomer vierohodnosti pre χ^2 štatistiku (Likelihood Ratio Chi-Squared Statistics, Attneave, 1956)**

$$G^2(a_i, S) = 2\log(2)\ |S|\ Gain(a_i, S),$$

Pomer vierohodnosti pre χ^2 štatistiku môžeme vypočítať podľa vzťahu uvedeného vyššie, kde $A = (a_1,\ a_2,\ \ldots, a_n)$ predstavuje sadu vstupných parametrov a S tréningovú sadu dát. Tento pomer je dôležitý pre meranie štatistickej významnosti kritéria obsiahnutej informácie. Hypotéza H_0 hovorí, že vstupný atribút a cieľový atribút sú nezávislé. Ak hypotéza H_0 platí, tak testovacia štatistika pochádza z χ^2 rozdelenia s $(dom(a_i)-1)(dom(y)-1)$ stupňami voľnosti, kde $dom(a_i)$ predstavuje dimenziu vstupných parametrov a $dom(y)$ dimenziu výstupu.

- **Koeficient obsiahnutej informácie (Gain Ratio, Quinlan, 1993)**

$$Gainratio(a_i, S) = \frac{Gain(a_i, S)}{Entropia(a_i, S)}$$

Gain ratio je modifikáciou veľkosti obsiahnutej informácie Gain. Gain ratio je veľké, ak sú dáta rozdelené rovnomerne, naopak malé, ak je podiel pozorovaní z jednej kategórie vysoký.

Ďalšie používané algoritmy [10]:

- C4.5 – vychádza z ID3 algoritmu a používa rozhodovacie kritérium Gain Ratio. Rozdeľovanie do podmnožín končí, ak počet podmnožín presiahne ich vopred definovaný počet. Algoritmus zvládne aj číselné atribúty.

- CART – skratka pre klasifikačné a regresné stromy. Strom je konštruovaný tak, že každý vnútorný uzol má dve vychádzajúce vetvy. Dôležitá je schopnosť algoritmu vytvárať regresné stromy, pričom pri regresných stromoch sa nepredpovedá klasifikačná premenná ale hodnota (reálne číslo).

3.3 Výpočet entropie pre jeden atribút

Entropia pre atribút Survived

Ako sme uviedli v Kapitole 2, v prípade, že sada dát S je rozdelená na triedy $S_1,\ S_2,\ \ldots, S_n$, entropiu počítame ako:

$$Entropia(a,b) = \sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} = \sum_{i=1}^{c} p_i\ \log_2 p_i.$$

Survived	
No	1 490
Yes	711

Tabuľka 3: Skladba dát podľa prežitia pasažierov

V Tabuľke 3 sú počty cestujúcich rozlíšených podľa toho, či prežili tragédiu, môžeme si všimnúť, že tragédiu prežilo 711 cestujúcich. Teraz vypočítame entropiu pre atribút Survived:

$$Entropia_{Survived}(No, Yes) = -p(No)\log_2 p(No) - p(Yes)\log_2 p(Yes) = \frac{1490}{2201}\ \log_2 \frac{1490}{2201} - \frac{711}{2201}\ \log_2 \frac{711}{2201}.$$

3.4 Výpočet entropie pre dva atribúty

K ďalším výpočtom budeme potrebovať entropiu pre tabuľku s dvoma atribútmi. Ak označíme X množinu hodnôt jedného a atribútu a $P(c)$ podiel dát, ktoré majú atribút $c \in X$, entropia je daná vzťahom:

$$E(T,X) = \sum_{c \in X} P(c)\ Entropy(c).$$

Survived				
		No	Yes	
Sex	Female	126	344	470
	Male	1 364	367	1 731
				2 201

Tabuľka 4: Skladba dát podľa prežitia a pohlavia pasažierov

Dáta ďalej rozdelíme podľa dvoch atribútov, podľa toho, či cestujúci prežil tragédiu a podľa jeho pohlavia – interpretujeme ich pomocou Tabuľky 4. Vysvetlíme si výpočet entropie pre dva atribúty Survived a Sex:

$$Entropia_{Survived,Sex}(No, Yes) = P(Male)Entropia_{Survived|Male}(No, Yes)+$$
$$P(Female)Entropia_{Survived|Female}(No, Yes) = \frac{470}{2201} * 0,84 + \frac{1731}{2201} * 0,74 = 0,76.$$

Rovnakým spôsobom sme vypočítali veľkosť entropie pre každý zo sledovaných atribútov.

Survived				
		No	Yes	
Age	Child	52	57	109
	Adult	1 438	654	2 092
				2 201

Tabuľka 5: Skladba dát podľa prežitia a veku pasažierov

Podľa dát z Tabuľky 5 vypočítame entropiu pre dva atribúty Survived a Sex:

$$Entropia_{Survived,Age}(No, Yes) = P(Adult)Entropia_{Survived|Adult}(No, Yes)+$$
$$P(Child)Entropia_{Survived|Child}(No, Yes) = \frac{109}{2201} * 0,999 + \frac{2092}{2201} * 0,893 = 0,898.$$

Posledným parametrom je ekonomická trieda pasažierov. Dáta si rozdelíme v Tabuľke 6 podľa toho, či pasažier prežil tragédiu a podľa ekonomickej triedy, ktorou

Survived				
		No	Yes	
Class	1st	122	203	325
	2nd	167	118	285
	3nd	528	178	706
	Crew	673	212	885
				2 201

Tabuľka 6: Skladba dát podľa prežitia a ekonomickej triedy pasažierov

cestoval. Výpočet entropie pre dva atribúty, Survived a Class, zapíšeme ako:

$$Entropia_{Survived,Class}(No, Yes) = P(1st)Entropia_{Survived|1st}(No, Yes)+$$
$$P(2nd)Entropia_{Survived|2nd}(No, Yes) + P(3rd)Entropia_{Survived|3rd}(No, Yes)+$$
$$P(Crew)Entropia_{Survived|Crew}(No, Yes) = \frac{325}{2201} * 0,958 + \frac{285}{2201} * 0,977+$$
$$\frac{706}{2201} * 0,811 + \frac{885}{2201} * 0,795 = 0,848.$$

3.5 Veľkosť obsiahnutej informácie

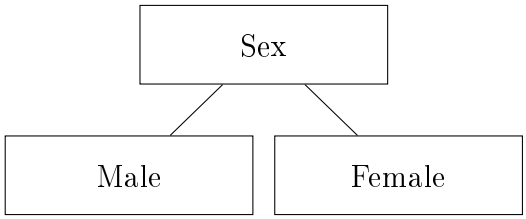
Výstavba modelu je založená na hľadaní atribútu s najväčšou obsiahnutou informáciou (t. j. uzla pre najviac homogénnu vzorku):

$$Gain(T, X) = Entropia(T) - Entropia(T, X),$$

Výpočet potom pre náš príklad vyzerá nasledovne:

$$Gain(Survived, Age) = Entropia(Survived) - Entropia(Survived, Age) =$$
$$0,904 - 0,898 = 0,006,$$
$$Gain(Survived, Sex) = Entropia(Survived) - Entropia(Survived, Sex) =$$
$$0,904 - 0,761 = 0,143,$$
$$Gain(Survived, Class) = Entropia(Survived) - Entropia(Survived, Class) =$$
$$0,904 - 0,848 = 056.$$

Najviac informácie je obsiahnuté v atribúte Sex. Táto skutočnosť sa dala očakávať, keďže prednostné právo na opustenie lode mali ženy a deti. Prediktor nazveme rozhodovacím uzlom. Rozhodovací strom tak vieme zjednodušenie zakresliť v Obr. 17 ako:



Obr. 17: Rozhodovací strom pre pohlavie (Sex)

Hľadáme ďalšie uzly pre obe rozhodnutia. A pokračujeme takto rekurzívne, až kým sa nedostaneme ku koncovým uzlom.

Survived	
No	1 364
Yes	367

Tabuľka 7: Rozhodovací strom pre uzol Male

Začneme s uzlom pre mužov (Male) a pozrieme sa na to, koľko z pasažierov tragédiu prežilo. Vypočítame si entropiu pre atribút Survived pre dáta o mužoch:

$$Entropia_{Survived,Male}(No, Yes) = 0,745.$$

Survived				
		No	Yes	
Age	Child	35	29	64
	Adult	1329	338	1667
				1731

Tabuľka 8: Skladba dát uzla pre mužov podľa prežitia a veku pasažierov

V Tabuľke 8 rovnaké dáta ďalej rozdelíme aj podľa atribútu vek (Age). Vypočítame entropiu pre atribúty Age a Male, rovnako ako veľkosť obsiahnutej informácie:

$$Entropia_{Survived,Age|Male}(No, Yes) = 0,737$$

$$Gain = 0,008$$

Survived				
		No	Yes	
Class	1st	118	62	180
	2nd	154	25	179
	3nd	422	88	510
	Crew	670	192	862
				1731

Tabuľka 9: Skladba dát uzla pre mužov podľa prežitia a ekonomickej triedy pasažierov

Posledným nepoužitým atribútom je atribút ekonomickej triedy pasažierov (Class) interpretujeme v Tabuľke 9 a dopočítame entropiu:

$$Entropia_{Survived,Class|Male}(No, Yes) = 0,733.$$

Veľkosť obsiahnutej informácie je v tomto prípade 0,012, čo je viac ako v prípade atribútu Age. Rozhodovací strom s doplneným uzlom pre mužov potom vieme vyobraziť v tvare ako na Obr. 18.

$$Gain = 0,012$$

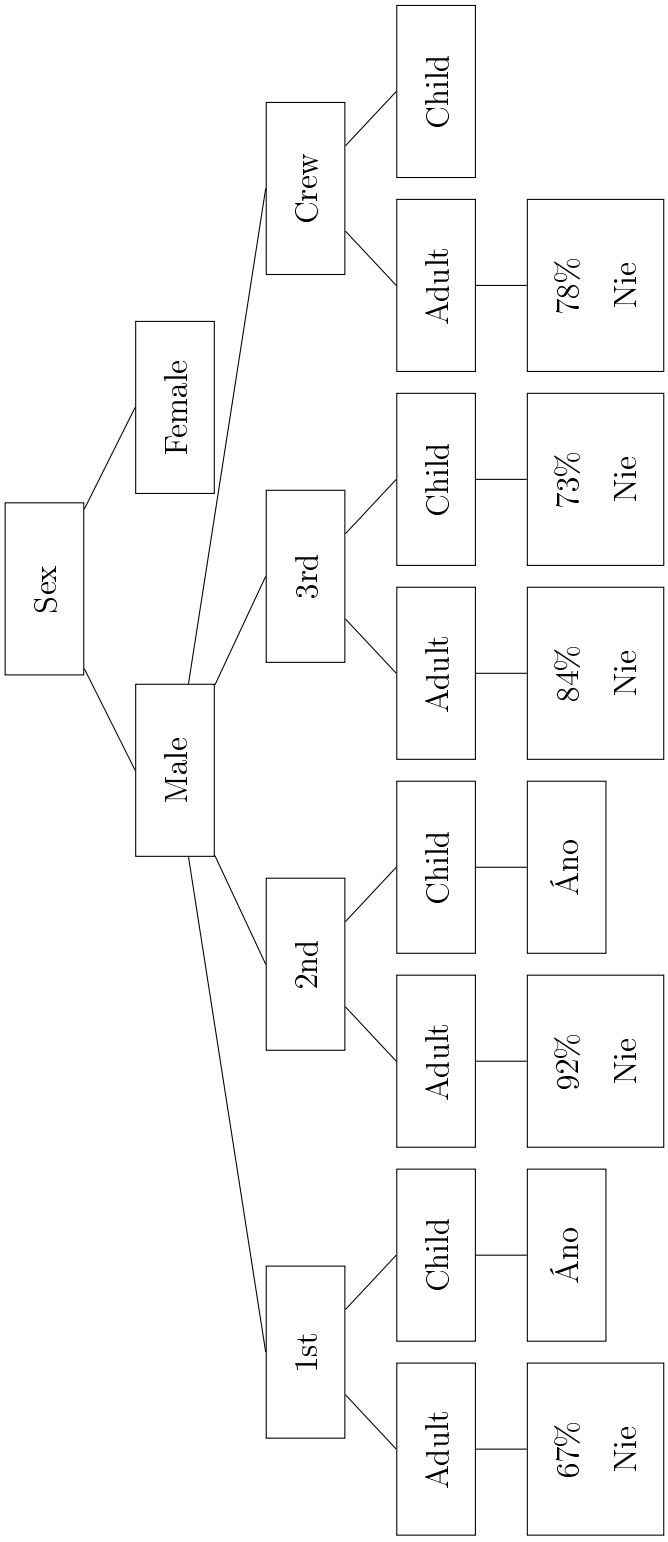
3.6 Rozhodovací strom pre uzol Female

Rovnako ako pri dátach o mužoch, vypočítame entropiu pre uzol žien (Female). Potrebné dáta si zapíšeme v Tabuľke 10.

Tabuľka 10: Skladba dát uzla pre ženy podľa prežitia pasažierov

Survived	
No	126
Yes	344

$$Entropia_{Survived,Male}(No, Yes) = 0,839$$



Obr. 18: Rozhodovací strom s doplneným uzlom pre mužov

Tieto dáta rozdelíme podľa ďalšieho atribútu vek (Age) a zobrazíme ich v Tabuľke 11.

Survived				
		No	Yes	
Age	Child	17	28	45
	Adult	109	316	425
				470

Tabuľka 11: Skladba dát uzla pre ženy podľa prežitia a veku pasažierov

Dopočítame entropiu a veľkosť obsiahnutej informácie, ktorú porovnáme s veľkosťou pre atribút ekonomická trieda.

$$Entropia_{Survived,Age|Female}(No, Yes) = 0,834$$

$$Gain = 0,004$$

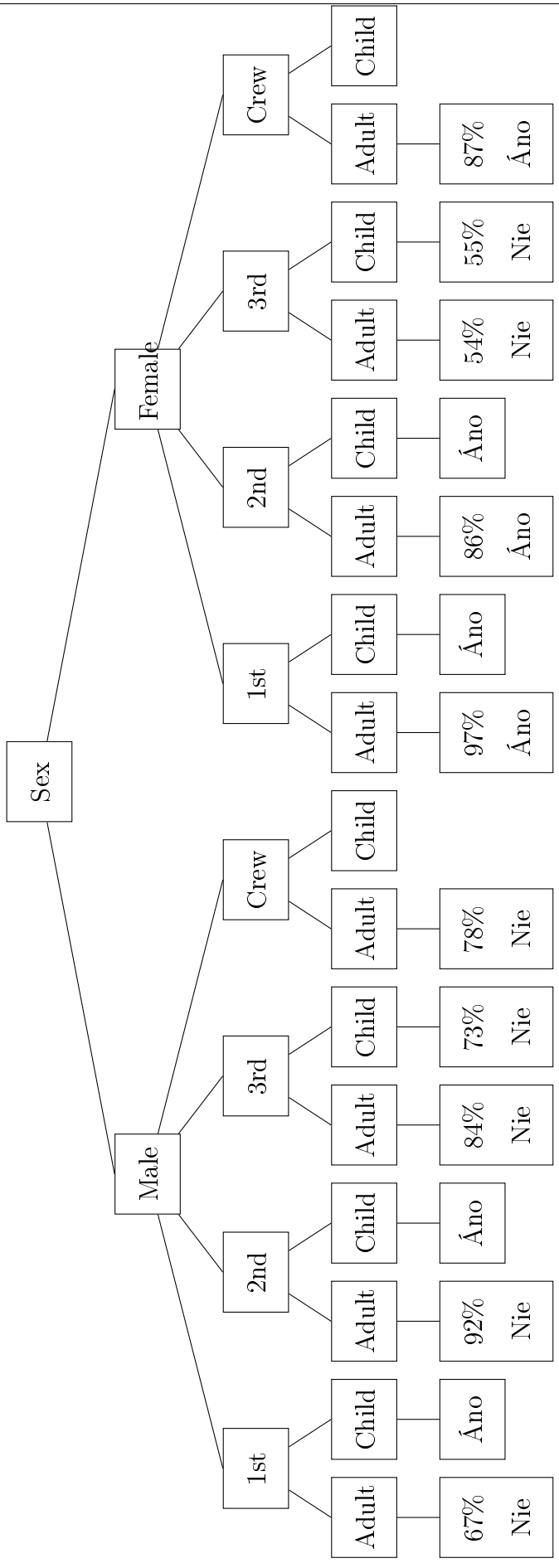
Survived				
		No	Yes	
Class	1st	4	141	145
	2nd	13	93	106
	3nd	106	90	196
	Crew	3	29	32
				479

Tabuľka 12: Skladba dát uzla pre ženy podľa prežitia a ekonomickej triedy pasažierov

$$Entropia_{Survived,Class|Female}(No, Yes) = 0,620$$

$$Gain = 0,219$$

Dáta z Tabuľky 12 sme použili na výpočet vyššie spomenutej entropie a veľkosti obsiahnutej informácie, tá je vyššia ako pre atribút Age. Výsledný strom potom vieme zapísať v tvare vyobrazenom na Obr. 19.

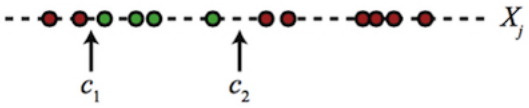


Obr. 19: Výsledný rozhodovací strom

3.7 Príklad spojitých atribútov

Prediktor Vek (Age) vystupoval v našom modeli ako kategorická premenná s dvoma hodnotami: dieťa a dospelý (child a adult). Vek však môže byť definovaný ako reálne číslo a obsahovať tak nekonečne veľa zlomových bodov c. Potrebujeme teda nájsť správny zlomový bod c pre test z každého z uzlov X_j:

$$X > c$$



Obr. 20: Príklad hraničných hodnôt medzi hodnotami s rôznou klasifikáciou

Pri ID3 algoritme hraničná hodnota c_i (threshold) musí ležať medzi dvoma hodnotami v_i a v_{i+1}, ktoré majú rôznu klasifikáciu.

Takýchto hodnôt c_i, v ktorých sa mení typ klasifikačnej premmenej môže byť viac, pre každú z týchto hodnôt je potrebné vypočítať veľkosť obsiahnutej informácie $Gain(c_i)$ – vyberáme tak hraničnú hodnotu c_i (threshold) s najväčšou obsiahnutou informáciou (information gain) ako $\arg\max_i Gain(c_i)$.

3.8 Zostavenie modelu v softvéri R

Na výstavbu modelu požívame funkciu *rpart* dostupnú v knižnici *rpart*. Knižnica slúži na rekurzívne delenie pre klasifikáciu, rozhodovacie a regresné stromy. Realizácia väčšiny použitých funkcí pochádza z knihy Breiman, Friedman, Olshen a Stone (1984).

```
library(rpart)
```

Dalšou použitou knižnicou je *RcolorBrewer*, slúži na zobrazenie grafov, stromov pre modely *rpart*.

```
library(rpart.plot)
data(ptitanic)
```

Dáta budeme rozdelovať na tréningovú a testovaciu sadu dát. Využijeme na to príkaz *createDataPartition* dostupný v knižnici *caret*.

```
library(caret)
library(rattle)
set.seed(3435)
```

Rozdelenie na tréningovú a testovaciu sadu dát je v pomere 60:40. Tréningová sada dát (training) obsahuje 786 dát, testovacia sada (testing) 523. Dáta boli rozdelené pomocou príkazu *createDataPartition*, jeden z jeho parametrov p predstavuje percentuálny podiel tréningovej sady dát na ich celkovom počte.

```
trainIndex<-createDataPartition(ptitanic$survived,p=0.6,list=
    FALSE,times=1)
training<-ptitanic[trainIndex,]
testing<-ptitanic[-trainIndex,]
```

Na výstavbu rozhodovacieho stromu použijeme funkciu *rpart* z knižnice *rpart* s nasledovnými parametrami:

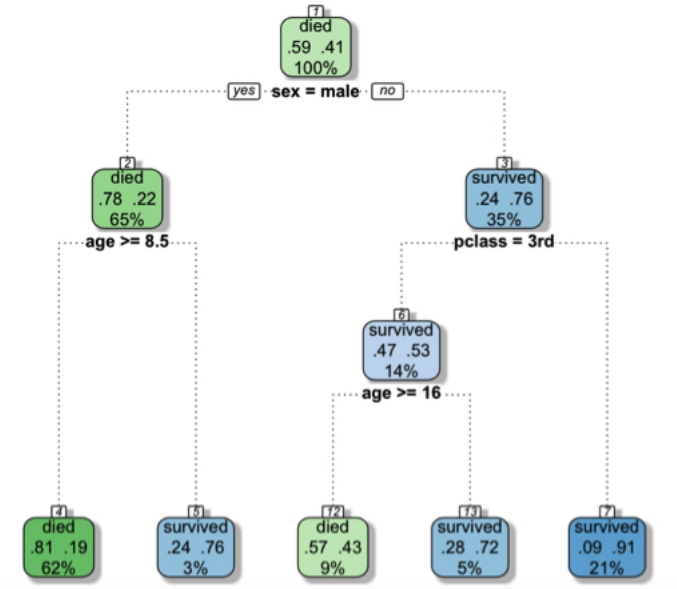
- **formula** - závislosť prežitia od pohlavia Survived Sex,
- **data** - zdroj dát,
- **method** - metóda, použili sme metódu „class“ pretože našou predikovaná hodnota survived je kategorickou premennou, ďalšie metódy „anova“ (pre regresné stromy), „poisson“ (pre poissonovu regresiu) a „exp“ (pre exponencálnu regresiu),
- **nepovinné argumenty funkcie** ako napríklad weights - váhy, subset - určuje časť z dát, ktoré majú byť v modeli použité, na.action - odstráni riadky dát, v ktorých chýba predikovaná hodnota, zároveň však zachová tie s chýbajúcim prediktorom, x a y - zachovávajú kópiu matice x, resp. závislú premennú vo výsledku.

```
fit<-rpart(survived~sex,data=training,method="class")
```

Vykreslíme rozhodovací strom zostrojeného *rpart* funkciou pomocou príkazu *fancyRpartPlot* z knižnice *rpart.plot*.

```
fancyRpartPlot(fit)
```

Z obrázku vidíme, že 62 % pasažierov lode Titanic tragédiu neprežilo. Pokiaľ bol pasažier muž (63 % pasažierov bolo mužského pohlavia), tak mal len 18 % šancu na



Obr. 21: Rozhodovací strom pre model s jednou vysvetľujúcou premennou

prežitie. Naopak, tragédiu prežilo 27 % žien, celkovo tvorili 37 % všetkých cestujúcich lode. Klasifikácia na základe modelu s jedným prediktorom a na tréningových dátach, type je znova class pre klasifikačný strom.

```
prediction_training<-predict(fit ,data=training ,type="class")
```

Výsledky si zobrazíme ako prehľadný výstup funkcie. Hodnoty v riadkoch predstavujú naše predikované hodnoty, hodnoty v stĺpcoch skutočné dáta (sada dát training v stĺpci survived), pomocou tabuľky tak vieme odhadnúť presnosť nášho modelu.

```
table(prediction_training ,training$survived)
##
## prediction_training  died  survived
##             died      406       100
##             survived   80       200
```

Celkom o 506 pasažieroch z tréningovej sady sme povedali, že tragédiu neprežili:

- 406 pasažierov sme zakategorizovali správne, náš výsledok je teda pravdivo negatívny (true negative), 80 % tragédiu neprežilo,
- o 100 pasažieroch sme nesprávne povedali, že tragédiu prežili (20 %), tento výsledok označíme podľa Kapitoly 1 za nepravdivo negatívny.

Tragédiu prežilo 280 ľudí z tréningovej sady. Z nich 200 ľudí označil náš model správne, pravdivo pozitívne, môžeme teda povedať, že na tréningových dátach korektne identifikoval 71 % pasažierov, ktorí tragédiu prežili. Klasifikácia na rovnakom modeli ale na testovacích dátach, argumenty funkcie ostali nezmenené.

```
prediction_testing<-predict(fit,newdata=testing,type="class")
table(prediction_testing,testing$survived)
##
## prediction_testing  died  survived
##            died      275       60
##            survived   48      140
```

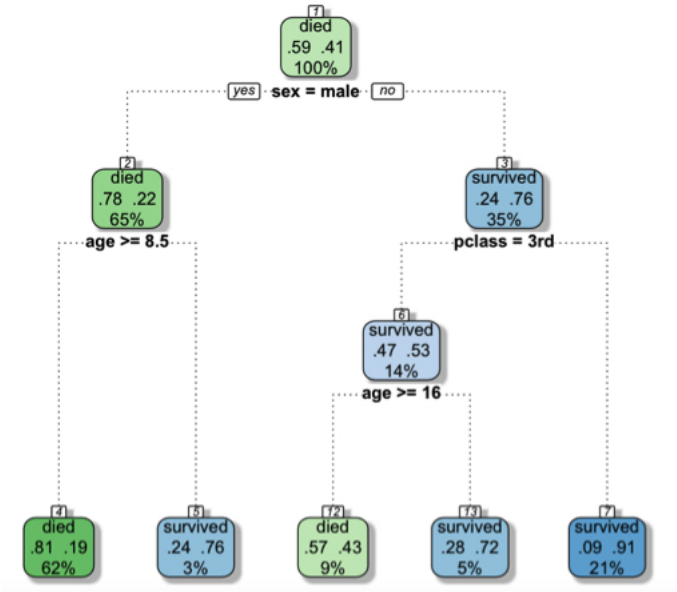
Podľa výpisu vidíme, že sme korektne zamietli 82 % cestujúcich. Naopak, pri 75 % cestujúcich bol náš výsledok pravdivo pozitívny, zaklasifikovali sme ich teda tiež správne.

Pri **výstavbe modelu s viacerými premennými** rozdelíme dáta na tréningové a testovacie v pomere 60:40.

```
trainIndex<-createDataPartition(ptitanic$survived,p=0.6,list=
    FALSE, times = 1)
training_viac<-ptitanic[trainIndex,]
testing_viac<-ptitanic[-trainIndex,]
fit1<-rpart(survived~pclass+sex+age+sibsp+parch,data=training_
    viac,method="class")
```

```
prediction_training_viac<-predict(fit1,data=training_viac,type
    ="class")
table(prediction_training_viac,training_viac$survived)
##
## prediction_training_viac  died  survived
##                  died      437       90
##                  survived   49      210
```

Celkom 527 pasažierov z tréningovej sady sme kategorizovali tak, že tragédiu neprežili:



Obr. 22: Rozhodovací strom pre model s viacerými vysvetľujúcimi premennými

- 437 pasažierov (83 %), sme kategorizovali správne, pravdivo negatívne, neprežili teda tragédiu,
- o 90 pasažieroch sme nesprávne povedali, že tragédiu prežili (9 %),

Tragédiu prežilo 249 ľudí z tréningovej sady. Z nich 210 ľudí označil náš model pravdivo pozitívne, korektne teda identifikoval 84 % cestujúcich.

```
prediction_testing_viac<-predict(fit1,newdata=testing_viac,type
    = "class")
table(prediction_testing_viac,testing_viac$survived)
##
## prediction_testing_viac died survived
##                    died      267       63
##                    survived   56      137
```

Nás model na testovacích dátach korektne zamietol 81 % pasažierov. Naopak, z tých pasažierov, ktorí tragédiu prežili, model korektne, teda pravdivo pozitívne, identifikoval 71 % cestujúcich.

4 Náhodné lesy

Jedným zo spôsobov, ako navýšiť prediktívnu výkonnosť metódy rozhodovacích stromov spomenutej v predchádzajúcej kapitole, je použiť náhodné lesy - skombinovať viacero rozhodovacích stromov do jedného modelu. Princíp fungovania metódy náhodných lesov si vysvetlíme na už v Kapitole 3 spomenutých dátach o pasažieroch lode Titanic, dostupných v knižnici *rpart.plot* ako *ptitanic*. Na základe prediktorov ako ekonomická trieda pasažiera, vek, pohlavie, počet súrodencov na palube budeme opätovne predpovedať, či sa cestujúci tragédu prežil alebo nie. Na testovacej množine vybudujeme náhodný les.

4.1 Vrecovanie (Bagging)

Pre všetky tri rozhodovacie stromy si zostrojíme podmnožinu z celkovej množiny dát v pomere $\frac{2}{3}$. Množinu budujeme s návratom, údaje o niektorých pasažieroch sa teda v podmnožine môžu vyskytovať viackrát. Pre ilustráciu (tieto podmnožiny pri výstavbe lesu nepoužijeme), na množine o šiestich pasažieroch s návratom zostrojíme tri podmnožiny o rozmere $\frac{2}{3}$ zo vzorky.

Z celkovej ilustratívnej vzorky pasažierov v Tabuľke 13.

pclass	survived	sex	age	sibsp
1st	survived	female	29	0
1st	died	female	2	1
1st	died	male	30	1
1st	died	female	25	1
1st	survived	male	48	0
1st	died	male	39	0

Tabuľka 13: Ilustratívna vzorka pasažierov lode Titanic

zostrojíme náhodne a s návratom tri separátne podmnožiny o rozmere 66,7 % zo vzorovej množiny (o počte štyroch pasažierov).

V prvej podmnožine (Tabuľka 14) sa opakujú údaje o tridsaťročnom pasažierovi, ktorý tragédiu neprežil.

pclass	survived	sex	age	sibsp
1st	survived	female	29	0
1st	died	male	30	1
1st	died	female	2	1
1st	died	male	30	1

Tabuľka 14: Vytvorenie 1. podmnožiny pomocou vrecovania

V druhej podmnožine (Tabuľka 15) sme aj napriek náhodnému výberu s opakovaním dosiahli, že sa žiadne z dát neopakujú.

pclass	survived	sex	age	sibsp
1st	died	female	2	1
1st	survived	female	29	0
1st	survived	male	48	0
1st	died	female	25	1

Tabuľka 15: Vytvorenie 2. podmnožiny pomocou vrecovania

V poslednej sade dát (Tabuľka 16) sa nám opakujú údaje o prvom pasažierovi (mužovi, ktorý tragédiu neprežil).

pclass	survived	sex	age	sibsp
1st	survived	female	29	0
1st	died	male	30	1
1st	died	female	2	1
1st	died	male	30	1

Tabuľka 16: Vytvorenie 3. podmnožiny pomocou vrecovania

Na každej z podobne skonštruovaných podmnožín budeme budovať rozhodovací strom.

4.2 Voľba prediktorov

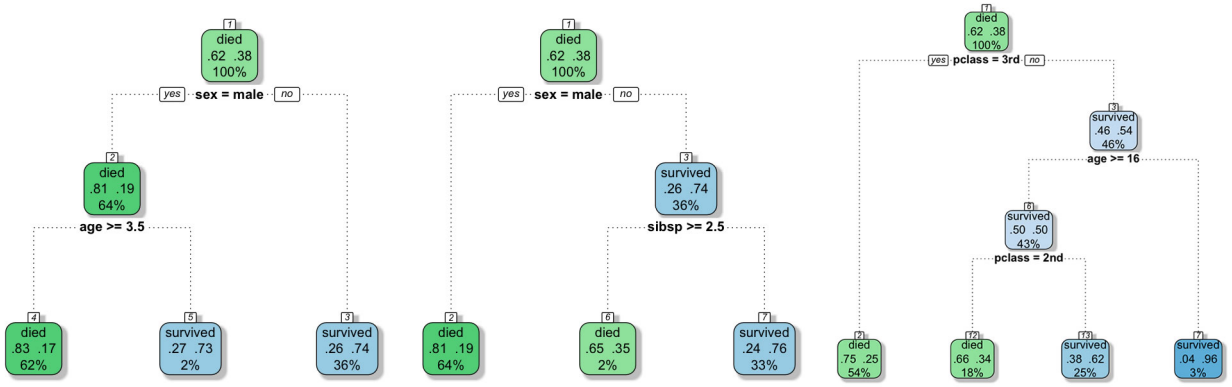
Ďalším zdrojom náhodnosti je výber prediktora [6]. Z množiny p premenných, v našom prípade štyroch prediktorov, vyberieme pre každý zo stromov $\sqrt{p}$ prediktorov (a teda 2 premenné). Takýmto rozdelením, v prípade, že je v datasete jeden silný prediktor a viacero priemerných, predchádzame použitiu rovnakého rozhodovacieho uzlu (podmienke) na vrchole stromu. Aj napriek tomu výsledné rozhodovacie stromy sú navzájom veľmi podobné, korelované, čo nevedie k výraznému zníženiu variance v porovnaní s jednoduchým rozhodovacím stromom.

Z množiny štyroch prediktorov teda vyberieme tri náhodné podskupiny o dvoch premenných:

1. skupina premenných	2. skupina premenných	3. skupina premenných
sex	sex	age
age	sibsp	pclass

Tabuľka 17: Ilustratívna vzorka pasažierov lode Titanic

Pre každú náhodnú podmnožinu a náhodne zvolenú dvojicu prediktorov v Obr. 23 zostrojíme rozhodovací strom.



Obr. 23: Presnosť v závislosti od počtu rozhodovacích stromov

Predstavme si, že máme pasažiera, trojročného chlapca, ktorý cestoval so svojimi dvoma súrodencami druhou ekonomickou triedou. Prvý rozhodovací strom ho klasi-

fikuje ako zachráneného, podľa druhého stromu je označený za mŕtveho a nakoniec podľa posledného je opätovne označený za živého. Výslednú klasifikáciu určíme podľa kritéria majoritnej voľby, každej z možných udalostí (prežitie, smrť) priradíme počet hlasov, vypočítame jej početnosť. Udalosť s najväčšou početnosťou je našou výslednou klasifikáciou, o našom cestujúcom teda povieme, že tragédiu prežil. Podobný postup opakujeme pre každého cestujúceho, resp. pre každý riadok dát.

4.3 Výstava modelu, vstupné parametre

Fungovanie algoritmu náhodného lesu si na tomto mieste formalizujeme [4]. Základnou myšlienkou je z jednej vzorky dát vytvoriť viacero rozhodovacích stromov na náhodne zvolených podmnožinách, pre každú z podmnožín dát náhodne vybrať skupinu prediktorov a vystavať strom.

Chystáme sa zostrojiť B rozhodovacích stromov. Pre každý z $b = 1$ až B rozhodovacích stromov:

- Náhodne a s návratom vytvoríme podmnožinu dát Z_b o rozmere N. Veľkosť podmnožiny je najčastejšie $\frac{2}{3}$ z celkového počtu dát (identická pre každý zo stromov).
- Pomocou rekurzívneho delenia vystaviame rozhodovací strom T_b na podmnožine dát Z_b. Nasledujúce kroky opakujeme pre každý z rozhodovacích uzlov (až kým nedosiahneme v poradí posledný uzol podľa zvoleného počtu n_{min}).
  - Spomedzi všetkých m prediktorov náhodne vyberieme p premenných. Počet premenných p najčastejšie vyberieme ako $\sqrt{m}$. Použitie menšieho výberu parametrov p je nápomocné práve vtedy, keď máme vyšší počet navzájom korelovaných prediktorov. Počtu premenných sa budeme držať počas celej výstavby rozhodovacieho stromu.
  - Pomocou niektorého z rozhodovacích kritérií (napríklad pomocou entropie alebo koeficientu Gini) vyberieme najlepšiu premennú z m.
  - Rozdelíme uzol na dve dcérske vetvy a pokračujeme v algoritme.
- Naším výstupom je skupina stromov $T_{B_{b=1}^{B}}$.

Pri klasifikácii sa výsledné rozhodovacie uzly vyberú podľa kritéria majoritnej voľby [7]. To znamená, že pre každý uzol vyberieme voľbu s najväčšou početnosťou, najväč-

ším počtom hlasov. V prípade binárnej premennej sa pozrieme na počet hlasov pre odpovede *Áno* a počet hlasov pre *Nie* a vyberieme voľbu s väčšou početnosťou, skóre s väčšou percentuálnou pravdepodobnosťou. Formálne zapísané, nech $C_b(x)$ je klasifikačná predikcia b-teho rozhodovacie stromu. Potom klasifikačná predikcia pre náhodný les $C_{nl}^B(x)$ je rovná majoritnej voľbe z $C_b(x)_{b=1}^B$. V regresnom prípade túto voľbu spravíme ako priemer závislých premenných.

4.4 Významnosť premenných

Najrozšírenejšou metódou pre meranie dôležitosti niektorej z premenných v náhodnom lese je zvýšenie priemernej chyby, resp. pokles priemernej presnosti (pri klasifikačnej úlohe je takouto chybou zlá klasifikácia) [3]. Predikčná schopnosť modelu je vypočítaná pri výstavbe b-teho stromu na OOB vzorke. Vzorka OOB (out-of-bag sample) je množina pozorovaní, ktoré nie sú použité pre vybudovanie súčasného stromu, používajú sa na odhad chyby predikcie a tiež na ohodnotenie významnosti premenných. Po prvom výpočte sú hodnoty pre j-tú premennú náhodne permutované v OOB sade a presnosť je vypočítaná opätovne. Priemerný pokles presnosti je potom vypočítaný ako výsledok priemeru všetkých permutácií stromov, použitý ako hodnota významnosti premennej j v náhodnom lese.

4.5 Zostavenie modelu v softvéri R

Model vystaviame na kompletnej sade dát o pasažieroch lode Titanic dostupných v knižnici *rpart.plot*.

```
library(rpart.plot)
data(ptitanic)
```

Na zostrojenie náhodného lesu použijeme knižnicu *randomForest* obsahujúcu algoritmus pre náhodné lesy Breimena a Cutlera na klasifikáciu a regresiu.

```
library(randomForest)
set.seed(3435)
```

Niektoré z pozorovaní pre premennú vek (age) obsahujú hodnoty *NA*. Nevýhodou náhodných lesov oproti rozhodovacím stromom je, že na dátach s chýbajúcimi hod-

notami model nevieme zostrojiť. Odstránime teda pozorovania s nevyplneným vekom pomocou príkazu *na.omit*.

```
ptitanic<-na.omit(ptitanic)
```

Podľa Kapitoly 1 rozdelíme dataset na tréningovú a testovaciu sadu v pomere 60:40.

```
trainIndex<-createDataPartition(ptitanic$survived, p=0.6, list
    = FALSE, times = 1)
training <- ptitanic[ trainIndex,]
testing  <- ptitanic[-trainIndex,]
```

Náhodný les zostrojíme pomocou funkcie *randomForest* dostupnej v knižnici s identickým názvom s nasledujúcimi parametrami:

- **ntree** - počet zostrojených rozhodovacích stromov. Nie je vhodné použiť príliš malé číslo, aby sme zaistili, že každý riadok vzorky bude zastúpený v náhodne zvolenej podmnožine niekoľkokrát.
- **mtry** - počet náhodne zvolených prediktorov z celkového počtu p pre každý z uzlov. Počet prediktorov sa líši pre klasifikačné metódy $\sqrt{p}$ oproti regresným úlohám $\frac{p}{3}$.
- **nodesize** - minimálna veľkosť koncových uzlov. Nastavenie tohto čísla na väčšie spôsobí, že vytvorené stromy budú menšie. Štandardne je táto hodnota nastavená na 1 pre klasifikáciu a 5 pre regresné úlohy.
- **importance** - nastavíme hodnotu na TRUE pre možnosť hodnotenia významu prediktorov.

Našou závislou premennou je prežitie/smrť pasažiera Titanicu, ktorú budeme klasifikovať na základe prediktorov ako vek (age), ekonomickú triedu pasažierov (pclass), pohlavie (sex), počet súrodencov (sibsp) na palube a počet rodičov alebo detí na palube (parch).

```
fit<-randomForest(factor(survived)~.,data=training,importance=
    TRUE, ntree=20, nodesize=1, mtry=round(sqrt(5)))
```

```
prediction_training<-predict(fit,type="class")
confusionMatrix(prediction_training,training$survived)
```

```
##
## prediction   0   1
##          0 323  83
##          1  48 174
```

Úspešnosť nášho modelu na tréningových dátach skontrolujeme pomocou príkazu *confusionMatrix*, ktorého vstupom sú predikované hodnoty a skutočné hodnoty podľa vzorky tréningovýh dát. Náš model zaklasifikoval 406 pasažierov tak, že tragédiu neprežijú (z toho 323 správne, pravdivo negatívne, čo predstavuje 79,6 %). Tragédiu podľa modelu prežilo 222 pasažierov (z toho model správne, pravdivo pozitívne, odhadol 174, čo predstavuje 78,4 %). Celková presnosť modelu na tréningových dátach predstavuje 79,1 %.

Úspešnosť nášho modelu overíme aj na separátnej skupine testovacích dát.

```
prediction_testing<-predict(fit,testing,type="class")
confusionMatrix(prediction_testing,testing$survived)
##
## prediction   0   1
##          0 219  54
##          1  28 116
```

Na nich náš model pravdivo negatívne odhadol 219 pasažierov, 80,2 % zo všetkých pasažerov označených za mŕtvych. Naopak, pravdivo pozitívne, označil model 116 pasažierov (čo je 80,6 % zo všetkých označených za zachránených).

Celková presnosť na testovacích dátach predstavuje 80,3 %.

5 Použitie modelov na dátach z praxe

Doteraz sme sa pri interpretácii modelov venovali vzorovým dátam dostupných v knižniciach prostredia R. To, ako funguje výkonnosť modelov v praxi, teraz otestujeme na dátach o zákazníkoch z vernostného programu, na ktorých žiadny z nich nebol predtým testovaný. Máme dataset z direct marketingovej komunikácie, emailu odoslaného na adresu 42 119 zákazníkov. O každom zo zákazníkov vieme:

- pohlavie,
- vek,
- dobu od jeho registrácie do programu po odoslanie kampane,
- ochotu otvárať emailové správy - podiel otvorených k celkovému počtu prijatých správ,
- počet pokladničných účtov - počet jeho nákupov,
- celkový obrat na účtoch - koľko člen programu minul.

V zmysle modelu teda budeme klasifikovať odoslané emailové správy na otvorené alebo neotvorené na základe spomenutých prediktorov. V datasete máme informáciu o 17 155 otvorených a 24 964 neotvorených správach.

Dáta najskôr podľa Kapitoly 1 rozdelíme na tréningovú a testovaciu sadu v pomere 60:40 (v tréningovej sade tak dostaneme 25 272 správ, v testovacej 16 847).

```
trainIndex<-createDataPartition(data$otvaranie, p=0.6, list =
    FALSE, times = 1)
training<-data[ trainIndex,]
testing <-data[-trainIndex,]
```

5.1 Logistická regresia

Naším prvým použitým prístupom bude logistická regresia. Podľa Kapitoly 2 teda zostrojíme model na tréningových dátach, najskôr s jedným regresorom, napríklad premennou o podiele otvorených správ ku všetkým odoslaným:

```
fit<-glm(otvoril~otvaranie,data=training,family=binomial)
```

Celková presnosť modelu na tréningových dátach predstavuje takmer 78,0 % (pravdivo negatívne sme modelom zaklasifikovali 12 770 správ, čo predstavuje 79,1 % zo správ označených za neprečítané, pravdivo pozitívne 75,9 % zo všetkých správ označených za otvorené).

```
proba_training<-predict(fit,type="response")
predicted_training<-as.numeric(proba_training>0.5)
table(predicted_training,training$otvoril)
##
## prediction     0     1
##          0 12770  3366
##          1  2206  6931
```

Pozrieme sa na výstup funkcie s jedným prediktorom:

```
summary(fit)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4112  -0.6819  -0.4132   0.6669   2.4964

## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.54258    0.04250  -83.36   <2e-16 ***
## otvaranie    6.60655    0.08143   81.13   <2e-16 ***
```

Z Tabuľky 18 vidíme, že kritériom pre zaklasifikovanie správy za otvorenú bola početnosť otvorených správ na úrovni viac ako 53,6 %.

Pre nás dôležitou je však výkonnosť modelu na testovacej sade dát:

```
proba_test<-predict(fit,newdata=testing,type="response")
predicted_ test<-as.numeric(proba_test>0.5)
table(predicted_ test,testing$otvoril)
##
## prediction     0     1
##          0  8476  2244
```

no	predicted_training	otvaranie
1.	1	0,5362
...	...	...
5.	1	0,5364
...	...	...
8.	1	0,5366
...	...	...
8863.	1	1

Tabuľka 18: Zákazníci, ktorí podľa modelu otvoria správu a ich podiel otvorených správ

```
##          1        1512        4614
```

Celková presnosť modelu dosiahla 77,7 % a je teda porovnateľná s presnosťou na tréningovej sade. Pozitívne sme klasifikovali 75,3 % všetkých správ označených za otvorené, naopak negatívne zaklasifikovaných bolo 79,1 %.

Pokúsime sa prediktívnu výkonnosť zvýšiť zahrnutím viacerých prediktorov. Zostrojíme zložený model obsahujúci všetky premenné dostupné v našom dátovom zdroji v tvare:

```
fit<-glm(otvoril~otvaranie+vek+dlzka_v_programe+pohlavie+pocet
    _uctov+obrat_uctov,data=training,family=binomial)
```

V tomto prípade je celková presnosť porovnateľná s modelom s jedným regresorom - dosiahla hodnotu 77,9 %. Premenná o podiele otvorených emailov sa teda aj numericky, nielen empiricky, zdá ako najviac závislá na predikovanie otvorenia ďalšej prijatej emailovej správy. Pozrieme sa ešte na výstup funkcie pre viac prediktorov:

```
summary(fit)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4167  -0.6826  -0.4117   0.6660   2.5112

## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
```

```
##  (Intercept)       -3.702e+00  7.074e-02 -52.343  < 2e-16 ***
##  otvaranie          6.573e+00  8.196e-02  80.196  < 2e-16 ***
##  obrat_uctov        3.126e-05  4.884e-05   0.640 0.522164
##  pocet_uctov        1.141e-03  1.732e-03   0.659 0.510013
##  dlzka_v_programe  -1.060e-06  5.568e-06  -0.190 0.849039
##  vek                4.943e-03  1.463e-03   3.379 0.000727 ***
##  pohlavie          -5.580e-02  3.434e-02  -1.625 0.104174
```

Len pre úplnosť informácie, celková presnosť na testovacích dátach sa znížila len o dve desatiny percentuálneho bodu oproti presnosti na treningovej množine - na úroveň 77,7 %.

5.2 Rozhodovacie stromy

Rozhodovací strom vybudujeme opätovne na celej sade prediktorov a pozrieme sa, ktoré rozhodovacie kritérium sa dostalo do najvyššieho uzla. Pomocou funkcie *rpart* dostupnej v knižnici s rovnakým názvom a popísanej v Kapitole 3 vybuduje model v tvare:

```
fit<-rpart(otvoril~otvaranie+vek+dlzka_v_programe+pohlavie+
    pocet_uctov+obrat_uctov,data=training,method="class")
```

```
prediction_training<-predict(fit,type="class")
table(prediction_training,training$otvoril)
##
##  prediction     0     1
##           0  12560  3069
##           1   2514  7129
```

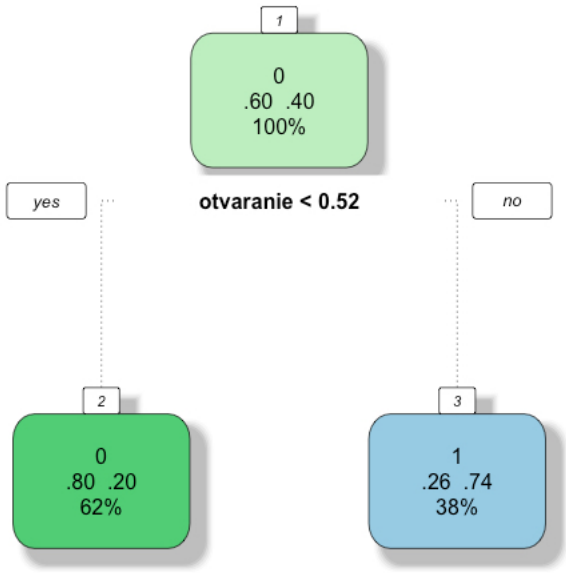
Presnosť na tréningovej sade dát v takomto prípade predstavuje 77,9 % (čo je nepatrné zhoršenie oproti logistickej regresií o 0,1 percentuálneho bodu). Naopak, pravdivo pozitívne sme zaklasifikovalo 73,9 % všetkých správ označených za prečítané.

Pozrieme sa, či sa prediktívna výkonnosť zmenila na testovacej množine dát.

```
prediction_testing<-predict(fit,newdata=testing,type = "class"
    )
```

```
table(prediction_testing, testing$otvoril)
##
## prediction    0    1
##          0 8266 2101
##          1 1624 4856
```

Dosiahli sme celkovú presnosť na úrovni 77,9 % (nepatrne vyššia ako pri použití logistickej regresie na testovacej množine). V Obr. 24 si vykreslíme si výsledný rozhodovací strom:



Obr. 24: Rozhodovací strom pre zákazníkov z vernostného programu

V Obr. 24 vidíme, že model odignoroval všetky prediktory s výnimkou prediktoru o podiele otvorených ku všetkým prijatým správam. Rozhodovacím kritériom v tomto prípade bola početnosť otvorených správ nižšia ako 52 %, keď model správu označil ako neprečítanú. Pri logistickej regresií bolo toto hraničné kritérium na úrovni 54 % (správne sme za prečítané pomocou rozhodovacieho stromu zaklasifikovali o 251 správ viac).

5.3 Náhodné lesy

Prediktívna výkonnosť, ktorú poskytlo klasifikovanie pomocou rozhodovacieho stromu, je pre náš prípad postačujúca. Na našich dátach ešte otestujeme model, ktorý vychádza

práve z metódy rozhodovacích stromov a prediktívnu výkonnosť by mal ešte navýšiť - náhodné lesy.

Model zostrojíme pomocou príkazu *randomForest* dostupnom v rovnomennej knižnici. Vstupom príkazu je počet rozhodovacích stromov (*ntree*), minimálna veľkosť koncového uzla (*nodesize*, pre klasifikáciu nastavená na 1) a počet prediktorov pre rozhodovacie stromy (*mtry*). Počet rozhodovacích stromov sme zatiaľ empiricky nastavili na hodnotu 20, kvôli výpočtovej zložitosti, túto hodnotu si neskôr zanalyzujeme. Prediktory pre každý rozhodovací strom sa vyberajú náhodne z celkovej množiny premenných, najčastejšou hodnotou pre *mtry* je odmocnina z celkového počtu prediktorov, použijeme ju teda ako vstup pre náš model.

```
fit<-randomForest(factor(otvoril)~vek+dlzka_v_programe+
    pohlavie+pocet_uctov+obrat_uctov+otvaranie,data=training,
    importance=TRUE, ntree=20, nodesize=1, mtry=round(sqrt(6)),na
    .omit=TRUE)
```

Pozrieme sa na výkonnosť modelu na tréningových a testovacích dátach.

```
prediction_testing<-predict(fit,newdata=testing,type = "class"
    )
table(prediction_testing,testing$otvoril)
##
## prediction     0     1
##          0 12176  3527
##          1  2743  6826
```

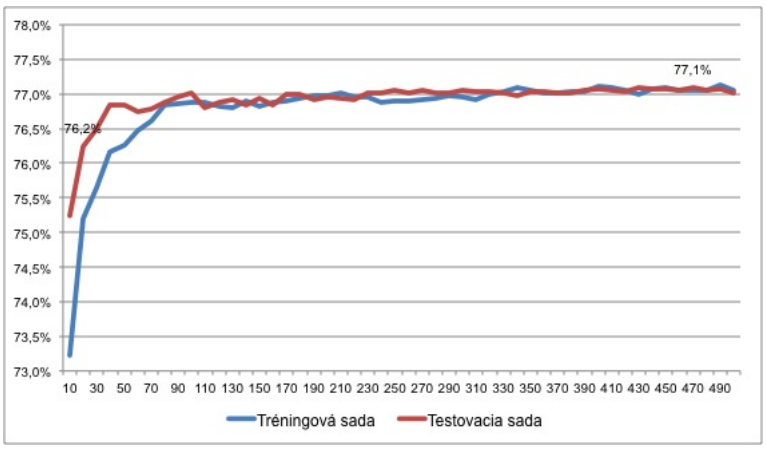
Pravdivo negatívne sme odhadli 12 176 z celkového počtu 15 703 správ označených za neotvorené (čo predstavuje 77,5 % negatívne zaklasifikovaných hodnôt). Za pravdivo pozitívne sme označili 71,3 % zo všetkých správ modelom označených za otvorené. Celková presnosť modelu tak na tréningovej množine predstavuje takmer 75,2 %. Skutočnú výkonnosť však musíme otestovať na testovacej sade dát.

```
prediction_testing<-predict(fit,newdata=testing,type = "class"
    )
table(prediction_testing,testing$otvoril)
```

```
##
## prediction    0    1
##          0 8362 2319
##          1 1683 4483
```

V tomto prípade sme pravdivo negatívne zaklasifikovali 78,3 % správ modelom označených za neotvorené, pravdivo pozitívne naopak takmer 72,7 % správ. Presnosť nášho modelu na testovacej sade dát predstavuje viac ako 76,2 %, čo je porovnateľná výkonnosť akú sme dosiahli s jednoduchým rozhodovacím stromom a jedným prediktorom.

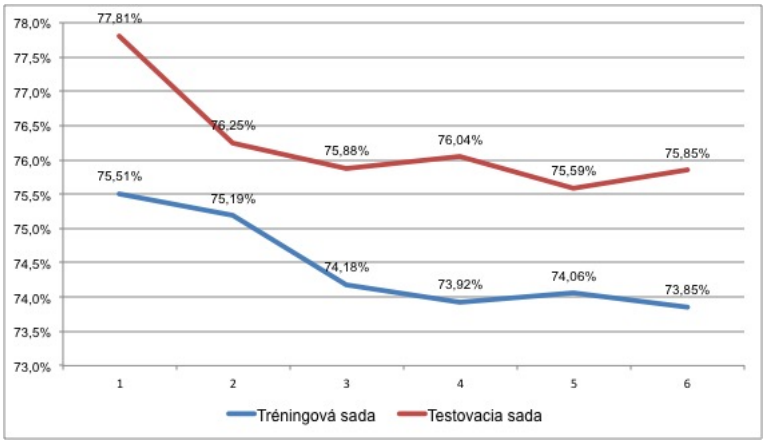
Pozrieme sa ešte na to, aký vplyv na prediktívnu výkonnosť na testovacej a tréningovej sade má použitie vyššieho počtu rozhodovacích stromov ako základ pre náhodný les. Porovnali sme presnosť pri použití 10 až 500 rozhodovacích stromov - najlepším výsledok na testovej sade sme dosiahli pri použití *ntree* rovnému 470 (77,1 %). Výsledok je len o deväť desatín lepší, ako sme dosiahli pri použití 20 stromov, avšak na úkor výpočtovej zložitosti. Pre naše potreby je postačujúci aj nižší počet rozhodovacích stromov.



Obr. 25: Presnosť v závislosti od počtu rozhodovacích stromov

Pre tento počet stromov (20) zanalyujeme aj hodnotu počtu prediktorov. Pre *mtree* = 20 sme sa pozreli na počet prediktorov v intervale od 1 po 6 (v datasete máme šesť premenných). Priebeh presnosti na tréningovej a testovacej sade je vyobrazený v Obr. 27.

Môžme si všimnúť, že počet prediktorov, ktorý sme navolili podla odporúčania pre nastavenie modelu na $\sqrt{6}$, má druhú najlepšiu presnosť ako na tréningovej, tak na

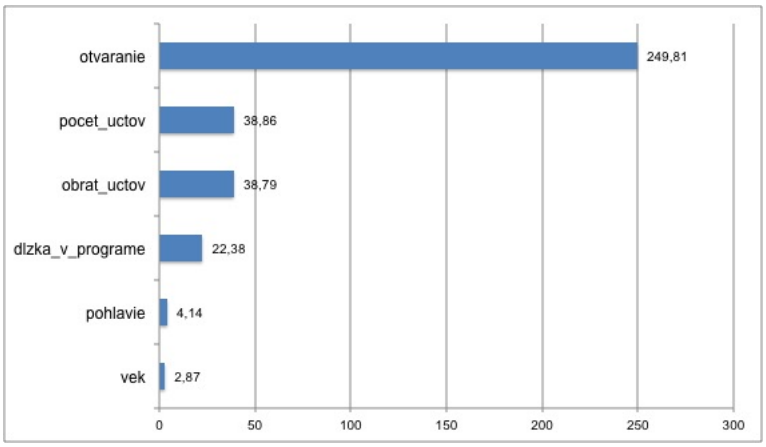


Obr. 26: Presnosť v závislosti od počtu prediktorov

testovacej sade dát.

Nevýhodou náhodných lesov oproti jednoduchým rozhodovacím stromov je strata interpretovateľnosti. Pomocou funkcie *importance* dostupnej v balíku *randomForest* sa pozrieme na dôležitosť jednotlivých prediktorov modelu v zmysle priemerného poklesu presnosti.

```
importance(fit ,type=1)
```



Obr. 27: Priemerný pokles presnosti

Tak, ako bolo zrejmé z už použitých modelov, najdôležitejšou premennou v modeli je podiel otvorených emailov k celkovému počtu prijatých emailových správ.

Po použití parametrov z našej analýzy, a teda počtu stromov (ntree) rovnému 470 a počtu prediktorov (mtry) rovnému 1 sme dosiahli celkovú presnosť na testovacej sade na úrovni 80,0 %, čo je najlepší výsledok na testovacej sade z porovnávaných metód.

```
fit<-randomForest(factor(otvoril)~vek+dlzka_v_programe+
    pohlavie+pocet_uctov+obrat_uctov+otvaranie,data=training,
    importance=TRUE, ntree=470,nodesize=1, mtry=1,na.omit=TRUE)
```

```
prediction_testing<-predict(fit,testing)
confusionMatrix(prediction_testing,testing$otvoril)
##
## prediction    0    1
##          0 8678 2344
##          1 1367 4458
```

Pravdivo negatívne sme pomocou tohto modelu zaklasifikovali 8 678 správ z 11 022 (čo je predstavuje 78,7 % zo všetkých správ modelom označených za neprečítané). Pravdivo pozitívne bolo zaklasifikovaných 4 458 z 5 825 (76,5 % zo správ modelom označených za prečítané).

5.4 Zhrnutie

V tejto kapitole sme si ukázali, že aj na reálnych dátach z vernostného programu je možné vybudovať fungujúci model s relatívne vysokou presnosťou. Jeho interpetovateľnosť pri zostrojení na datasete ako je ten náš je otázna - najvýznamnejším prediktorom bol pomer otvorených ku všetkým na zákazníka odoslaným správam, čo je očakávaným a smerodajným výsledkom aj bez akéhokoľvek modelovania. Naopak, vysvetľujúce premenné ako pohlavie, dĺžka v programe alebo jeho nákupné správanie sa neukázali ako signifikantné faktory pre otváranie direct marketingových správ. K zostrojeniu interpretovateľnejšieho a presnejšieho modelu by sme potrebovali obsiahlejšie dáta, pri ďalšom modelovaní (a pri väčších možnostiach zdrojovej databázy) by náš dataset obsahoval informáciu o:

- pomer počtu otvorených k odoslaným správam za posledné tri mesiace,
- pomer počtu otvorených k odoslaným správam v predposlednom kvartáli,
- pomer počtu otvorených k odoslaným správam v predposlednom polroku,
- zariadenie, z ktorého sú správy otvárané (mobilný telefón, tablet, desktop/notebook) - vývoj v čase podobne ako pri pomere počtu otvorených k odoslaným

správam,

- priemerný čas od odoslania do otvorenia správy.

Pri výstavbe by sme preferovali prediktory v tvare časových radov, ich vývoj v čase.

Záver

Klasifikačné metódy, téma našej bakalárskej práce, v súčasnosti ponúkajú široké spektrum aplikovateľných metód - čo otvára otázku na ich porovnanie, vysvetlenie vlastností a aplikáciu. V prvých štyroch kapitolách našej práce sme sa zamerali na teoretické vysvetlenie princípu fungovania a demonštráciu využiteľnosti metód na vzorových dátach. V rámci rozsahu záverečnej práce sme sa bližšie pozreli na tri metódy - logistickú regresiu, rozhodovacie stromy a náhodné lesy.

Dôležitou časťou bakalárskej práce je implementácia metód v prostredí R. Praktickú aplikáciu, rovnako ako prípravu dát, ich rozdelenie na tréningové a testovacie množiny, vstupné parametre sme podrobne vysvetlili v závere každej z kapitol.

Kľúčovou časťou bakalárskej práce je piata kapitola, v ktorej sme všetky teoretické znalosti z predchádzajúcich kapitol aplikovali na réalne dáta z vernostného programu - klasifikovali sme otváranie direct marketingových kampaní oslovenými členmi programu. V tejto časti sa ukázali všetky spomenuté modely na našich dátach ako porovnateľné - presnosť sa, ako pomer správne zaklasikovaných ku všetkým správam v datasete, v každom z prípadov pohybovala v intervale medzi 75 - 80 %. Využiteľnosť metód sme tak demonštrovali na reálnych dátach, na ktorých žiadna z nich nebola predtým použitá.

Možností rozšírenia závarečnej práce je viac. Tak, ako sme už navrhli na konci piatej kapitoly, pri budovaní modelu je možné použiť väčší počet vysvetľujúcich premenných - priebeh otvárania v čase, pomer otvorených ku všetkým odoslaným správam za posledné tri mesiace, vo štvrtom až šiestom mesiaci a v predposlednom polroku. Po teoretickej stránke sa vieme zamerať na ďalšie klasifikačné metódy - neurónové siete, metódy AdaBoost, LDA alebo QDA. Oboje si však vyžadujú väčší rozsah ako ponúka táto práca a možnosti dátovej sady.

V bakalárskej práci sme využili časť znalostí nadobudnutých v predmetoch ako pravdepodobnosť a štatistika, štatistické metódy a počítačová štatistika.

Zoznam použitej literatúry

[1] Dawson, R.: *Survival of passengers on the Titanic*, dostupné na internete (01.5.2015): http://stat.ethz.ch/R-manual/R-devel/library/datasets/html/Titanic.html

[2] Giudici, P.: *Applied data mining: Statistical Methods for Business and Industry*, John Wiley & Sons, Chichester, 2003

[3] Hastie T., Tibshirani R., Friedman J.: *Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, New York, 2009

[4] James G., Witten D., Hastie T., Tibshirani R.: *An Introduction to Statistical Learning: with Application in R*, New York, 2014

[5] John Hopkins University: *Data Science Specialization*, dostupné na internete (01.5.2015): http://github.com/DataScienceSpecialization

[6] Karatzoglou A., Smola A., Hornik K.: *Kernel-based Machine Learning Lab*, dostupné na internete (01.5.2015): http://cran.r-project.org/web/packages/kernlab/kernlab.pdf

[7] Kuhn M., Johnson K.: *Applied Predictive Modeling*, Springer, New York, 2013

[8] Kuhn M.: *Classification and Regression Training*, dostupné na internete (01.5.2015): http://cran.r-project.org/web/packages/caret/caret.pdf

[9] Lantz, B.: *Machine Learning with R*, Packt Publishing, Birmingham, 2013

[10] Rokach M., Maimom O.: *Top-Down Induction of Decision Trees: Classifiers—A Survey*, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 35 (2005), 476-486

[11] Sayad S.: *An Introduction to Data Mining*, dostupné na internete (01.5.2015): http://www.saedsayad.com

[12] Shalizi C.: *Data Mining*, učebné texty, dostupné na internete (01.5.2015): http://www.stat.cmu.edu/ cshalizi/350