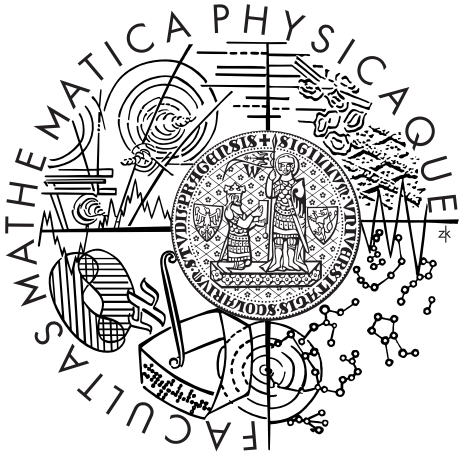
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Bc. Jana Hricová

Metody konstrukce klasifikátorů vhodných pro segmentaci zákazníků

Katedra pravděpodobnosti a matematické statistiky

Vedoucí diplomové práce:	prof. RNDr. Jaromír Antoch, CSc.
Studijní program:	Matematika
Studijní obor:	Finanční a pojistná matematika

Praha 2013

Chcela by som sa poďakovať všetkým, ktorí mi akýmkoľvek spôsobom pomohli pri spracovaní tejto diplomovej práce. Moje poďakovanie patrí predovšetkým vedúcemu práce, prof. RNDr. Jaromírovi Antochovi, CSc., za vedenie a cenné pripomienky pri spracovaní práce.

Osobitné poďakovanie patrí mojim rodičom a mojim najbližším, bez ktorých podpory a pomoci by som to určite nezvládla.

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Metody konstrukce klasifikátorů vhodných pro segmentaci zákazníků

Autor: Bc. Jana Hricová

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí diplomové práce: prof. RNDr. Jaromír Antoch, CSc., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: Diplomová práce pojednává o metodách, které jsou součástí široké oblasti analýzy dat, zvané klasifikace. V rámci klasifikačních metod jsou v práci představeny metody vhodné pro segmentaci zákazníků, které konstruují klasifikátory stromového typu. Podrobně je představena metodologie CART (Klasifikační a regresní stromy) a skupinové modely, vhodné pro konstrukci klasifikačních a regresních lesů, jmenovitě Bagging, Boosting, Arcing a Random Forest.

Popsané metody byly použity na reálná data z oblasti segmentace zákazníků a na simulované data v prostředí programu RStudio.

Klíčová slova: klasifikace, klasifikátory stromového typu, náhodné lesy

Title: Construction of classifiers suitable for segmentation of clients

Author: Bc. Jana Hricová

Department: Department of Probability and Mathematical Statistics

Supervisor: prof. RNDr. Jaromír Antoch, CSc., Department of Probability and Mathematical Statistics

Abstract: The master thesis discusses methods that are a part of the data analysis, called classification. In the thesis are presented classification methods used to construct tree like classifiers suitable for customer segmentation. Core methodology that is discussed in our thesis is CART (Classification and Regression Trees) and then methodologies around ensemble models that use historical data to construct classification and regression forests, namely Bagging, Boosting, Arcing and Random Forest.

Here described methods were applied to real data from the field of customer segmentation and also to simulated data, both processed with RStudio software.

Keywords: classification, tree like classifiers, random forests

Názov práce: Metódy konštrukcie klasifikátorov vhodných pre segmentáciu zákazníkov

Autor: Bc. Jana Hricová

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedúci diplomovej práce: prof. RNDr. Jaromír Antoch, CSc., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: Táto diplomová práca pojednáva o metódach, ktoré sú súčasťou širokej oblasti analýzy dát, zvanej klasifikácia. V rámci klasifikačných metód sú v práci predstavené metódy vhodné pre segmentáciu zákazníkov, ktoré konštruujú klasifikátory stromového typu. Podrobne je predstavená metodológia CART (Klasifikačné a regresné stromy) a skupinové modely, vhodné pre konštrukciu klasifikačných a regresných lesov, menovite Bagging, Boosting, Arcing a Random Forest.

Popísané metódy boli použité na reálne dáta z oblasti segmentácie zákazníkov a na simulované dáta v prostredí programu RStudio.

Kľúčové slová: klasifikácia, klasifikátory stromového typu, náhodné lesy

Obsah

Zoznam použitých symbolov

argmin	argument minimalizujúci funkciu (podobne argmax)
a, c	konštanty
C	množina kategórií,$C = (1,...,J)$
CART	metóda klasifikačných a regresných stromov (z angl. Classification and regression trees)
α	parameter zložitosti
$C_\alpha(T)$	kritérium zložitosti modelu T
$CV(.)$	odhad chyby pri krížovom overovaní
$d(\mathbf{x})$	klasifikačná funkcia (klasifikátor)
$e(t)$	chyba na trénovacom súbore
$e'(t)$	chyba na testovacom súbore
ESS	vysvetlený súčet štvorcov
GI	Giniho index
H	Entropia
J	počet kategórií
j_i	kategória i-tého znaku, $j_i \in C$, $i = 1,...,N$
$\mathcal{L}$	trénovací súbor
M	počet prediktorov (vysvetľujúcich premenných)
MAE	stredná absolútna chyba
ME	klasifikačná chyba (misclassification error)
MSE	stredná kvadratická chyba
N	celkový počet pozorovaní
N_j	počet pozorovaní v kategórii j
N_p	počet správne klasifikovaných pozorovaní
p_L	podiel pozorovaní z t, ktoré padnú do podmnožiny t_L (analogicky p_R)
p_{tj}	pravdepodobnosť kategórie j v uzle t
Q	množina otázok
$q \in Q$	otázka z množiny Q
R_t	región obsahujúci pozorovania uzlu t
R^2	koeficient determinácie
$r(.) = \mathsf{E}(Y\|\mathbf{X} = .)$	neznáma regresná krivka
$r_N^1(.)$	odhad regresnej krivky pomocou k_N najbližších susedov
$r_N^2(.)$	jadrový odhad regresnej krivky
$r_N^3(.)$	odhad regresnej krivky pomocou regresných stromov
RSS	reziduálny súčet štvorcov
$r(q,t)$	odhad MSE zodpovedajúcej odhadu generovanému rozkladom množiny t a otázke q
$\mathcal{T}$	testovací súbor

T	rozhodovací strom
T_0	maximálny strom
$\|T\|$	počet koncových uzlov
t	uzol, podmnožina priestoru $\mathcal{X}$
t_L,t_R	dcérske uzly, podmnožiny množiny t
TSS	úplný súčet štvorcov
$\mathcal{X}$	priestor hodnôt prediktorov (vysvetľujúcich premenných), $\mathcal{X} \subseteq \mathbb{R}_M$
$\mathbf{X}$	náhodný vektor vysvetľujúcich premenných, $\mathbf{X} = (X_1,...,X_M)$
X_i	náhodná veličina popisujúca chovanie znaku i, $i = 1,...,M$
Y	náhodná veličina, závislá premenná (vysvetľovaná premenná)
$(Y,\mathbf{X})$	náhodný vektor, kde Y je reálna náhodná veličina (regresný model)
$(y_i, \mathbf{x}_i)$	realizácie vektoru $(Y,\mathbf{X})$, kde $\mathbf{x}_i = (X_{i1},...,X_{iM})$, $i = 1,2,...,N$
$d_A(\mathbf{x})$	agregovaná klasifikačná funkcia (klasifikátor), resp. regresná
Bagging	akronym pre bootstrap aggregating, skupinový model pre konštrukciu lesov
B	počet bootstrapových výberov
Adaboost	akronym pre adaptive boosting, algoritmus pre konštrukciu lesov
w	vektor váh, $w = (w_1,...,w_N)$
WeakLearn	algoritmus vytvárajúci slabý model
Random Forest	algoritmus pre konštrukciu náhodných lesov
$\Theta_1,...,\Theta_S$	nezávislé rovnako rozdelené náhodné vektory
oob	pozorovania mimo bootstrapový výber
ntree	parameter udávajúci počet stromov v lese
m_0	počet náhodne vybraných prediktorov
MR	miera zlej klasifikácie
cpv	podiel klasifikácie pozorovaní do jednotlivých kategórií, (z angl. class proportion vote)
zpcpv	veľkosť zmeny pravdepodobnosti zaradenia vzorku do kategórie
Mprox	matica tesnosti

Úvod

Finančné inštitúcie a rôzne iné spoločnosti sa v posledných desiatkach rokov tradične potýkajú s narastajúcou konkurencieschopnosťou ostatných inštitúcií, obchodujúcich na finančnom trhu. Tento tlak posúva jednotlivé spoločnosti do situácie, kedy musia veľmi zreteľne a presne pracovať a komunikovať s klientelou, predovšetkým čo najlepšie zacieliť svoje služby a produkty koncovým užívateľom. Aby bola spoločnosť schopná takto efektívne zacieliť svoj marketing a portfólio produktov na klienta, je v prvom rade potrebná podrobná analýza tohto klientského kmeňa. Jedným z dôležitých inštrumentov, ktoré spoločnosti v dnešnej dobe využívajú, je analýza v zmysle segmentácie zákazníkov.

Segmentácii zákazníkov sa venuje oblasť nazývaná dobývanie znalostí. Dobývanie znalostí je široký pojem pre rôzne prístupy a metódy analýzy dát. Tento obor sa zameriava na nájdenie zaujímavých informácií z nazhromaždených dát, ideálne bez zásahu človeka. Samotné rozlišovanie a delenie do skupín je pritom jednou z najzákladnejších činností ľudstva.

V následujúcej práci sa zameriame na oblasť klasifikačných metód vhodných pre segmentáciu zákazníkov. V prvej kapitole si predstavíme základné pojmy z oblasti klasifikácie. Jednotlivé metódy analýzy dát sú podľa povahy skúmaného problému rozdelené na klasifikačné a regresné metódy. Tie sú následne kategorizované podľa rôznych hľadísk klasifikácie.

V druhej kapitole sa venujeme rozhodovacím stromom, ktoré sa využívajú v rámci segmentácie zákazníkov. Po predstavení základnej terminológie sa venujeme metodológii CART, ktorá položila základy pre konštrukciu binárnych stromov. Najskôr sa venujeme použitiu tohoto prístupu v klasifikácii a takisto je popísaný aj regresný prípad použitia stromov. Podrobne sa zameriavame na chybovosť modelu a jeho rozklad na systematickú chybu a rozptyl.

Tretia kapitola poskytuje teoretický úvod do skupinových modelov s náväznosťou na skupinové modely pozostávajúce zo stromov, nazývané lesy. Skupinové modely sú konštruované z dôvodu skvalitnenia celkovej klasifikácie. Kapitola sa venuje rôznym prístupom a metódam, pomocou ktorých dokážeme lesy konštruovať. Konkrétne sú predstavené metódy Bagging, Adaboost, Arcing a náhodné lesy (z angl. Random Forest).

Štvrtá kapitola je zameraná na praktickú aplikáciu predstavených metód. V prvej časti kapitoly sa podrobne venujeme analýze segmentácie zákazníkov na reálnych dátach klientov nemeckej banky pomocou metodológie CART a náhodných lesov. V rámci analýzy diskutujeme optimálne riešenie z pohľadu minimalizácie straty veriteľa v oboch prístupoch. Stratu veriteľa minimalizujeme penalizáciou zle klasifikovaných pozorovaní. Výsledky ukazujú, že celková chyba klasifikácie, získaná pomocou CART je porovnateľná s chybou klasifikácie pomocou náhodných lesov a skupinový model nepriniesol zlepšenie klasifikácie, ako bolo

predpokladané. Druhá časť štvrtej kapitoly sa venuje klasifikácii pomocou skupinových modelov a CART na reálnych a simulovaných dátach, tentokrát z rôznych oblastí a rôznej veľkosti. Na príkladoch klasifikačných a regresných dát testujeme redukciu chyby dosiahnutej na testovacích súboroch. Výsledky ukazujú, že v mnohých prípadoch vedú skupinové modely ku skvalitneniu celkovej klasifikácie a predikcie, narozdiel od predchádzajúceho príkladu segmentácie zákazníkov.

Kapitola 1

Úvod do klasifikácie

V každodennej realite nás život stavia do situácie, kedy sa musíme rozhodovať. Sú to rozhodnutia na rôznej úrovni a rôznej kvality. Kúpime dnes jednu alebo dve fľaše minerálky? Je človek, s ktorým trávime v posledných rokoch či mesiacoch ten, s ktorým prežijeme celý život? Samozrejme sa v týchto prípadoch rozhodujeme vo veľkej miere intuitívne a bez uvedomenia si, že aj za tým stojí určitá analýza a vyhodnotenie informácií. Kúpime dve fľaše, pretože máme veľký smäd alebo nemáme dostatok finančných prostriedkov, takže bude to len jedna fľaša a pod.

Zložitejšie úlohy však vyžadujú zložitejší prístup, než jednoduchú intuíciu. Je potrebných viac informácií a zložitejšie uvažovanie. V súčasnej dobe sa stretneme stále častejšie s dátovými súbormi vykazujúcimi charakteristické rysy, akými sú napríklad *zmes dátových typov* (dátový súbor obsahuje kategoriálne aj kvantitatívne premenné), *vysoký počet pozorovaných premenných* (dimenzionalita), *nehomogénnosť* (v rôznych častiach priestoru platia rôzne vzťahy), či *neštandardnosť datovej štruktúry* (dimenzia sa mení objekt od objektu) a pod.

Dátové súbory naberajú na objeme a rastie aj množstvo dát, ktoré sa snažíme interpretovať, či získať z nich nejakú informáciu. Všeobecne vyššia dimenzionalita spôsobuje, že dáta sú redšie a rozptýlenejšie. Tento problém je možno v niektorých prípadoch vyriešiť vyšším počtom pozorovaní, no nie vždy však veľký objem dát zaručuje bohatosť vnútornej štruktúry. Takisto sa často stáva, že niektoré zo súborov sú neúplné, tzn. pozorované veličiny úplne chýbajú, neboli merané alebo sú nezmyselné. S týmito vlastnosťami sa samozrejme zvyšujú aj nároky na spracovanie daných dátových súborov.

Spracovávanie a analýza dát je rozsiahlou a dôležitou témou v literatúre venovanej klasifikácii. Metódy klasifikácie sú predmetom záujmu nielen štatistikov, ale aj odborníkov z oblasti data miningu a z oblasti dobývania znalostí. V prílohe A sú predstavené základné princípy spracovávania dát a klasifikácie z hľadiska data miningu. Podrobne je popísaný priebeh analýzy využívaný v praxi.

Klasifikáciou rozumieme prístup, pri ktorom sa snažíme neznámy objekt zaradiť (klasifikovať) do konečného počtu vopred daných kategórií. Inými slovami, rozdeľujeme danú (teoretickú alebo konkrétnu) množinu objektov, javov alebo procesov na konečný počet podmnožín, ktoré obsahujú prvky s dostatočne podobnými spoločnými vlastnosťami.[1]

Vlastnosti, podľa ktorých klasifikáciu robíme, sú určené **klasifikačnými kritériami**. Objekty s podobnými vlastnosťami tvoria **klasifikačnú triedu**. Vlast-

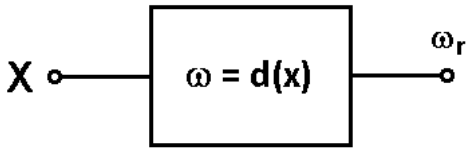
nosťou klasifikácie je, že musí byť v každom prípade úplná, tzn. že každý objekt je zaradený do práve jednej klasifikačnej triedy.

Klasifikátor (obr. 1.1) je algoritmus klasifikácie, ktorého vstup zodpovedá charakteru analyzovaných dát a hodnota jeho diskrétneho výstupu je **identifikátorom** klasifikačnej triedy, do ktorej klasifikátor pomocou **rozhodovacieho pravidla** zaradí vstupnú reprezentáciu dát. Rozhodovacie pravidlo je určované vo fáze učenia.

Platí, že

$$\omega_r = d(\mathbf{X}), \tag{1.1}$$

kde $\mathbf{X}$ predstavuje vstupnú reprezentáciu dát, $d(\mathbf{X})$ ako funkcia argumentu $\mathbf{X}$ je rozhodovacie pravidlo a ω_r, $r = 1,...,R$ je identifikátor r-tej klasifikačnej triedy.



Obr. 1.1: Klasifikátor

1.1 Typy premenných

Základnou formou reprezentácie dát je množina (prípadne usporiadaných) vektorov, obsahujúcich hodnoty veličín, ktoré o spracovávaných objektoch získavame. Hodnoty veličín, budeme označovať ako **premenné**.

Zo štatistického hľadiska delíme premenné v dátovom súbore na **závislú premennú** Y a **vysvetľujúcu premennú** X, nazývanú aj **prediktor**. Závislú premennú sa snažíme vysvetliť na základe vysvetľujúcich premenných. Príkladom závislej premennej môže byť počet jedincov určitého druhu a vysvetľujúcimi premennými údaje o prostredí, na základe ktorých sa snažíme túto početnosť odhadnúť. Uvažujme prípad z oblasti segmentácie zákazníkov. V prípade klientov banky, závislá premenná by mohol byť podiel rizikových klientov a vysvetľujúcimi premennými ich údaje o typoch účtov, úveroch, kreditných kartách a bonite.

Hlavným kritériom pre rozdelenie premenných je typ vzťahu medzi premennými. Podľa toho delíme premenné na:

kvantitatívne (**numerické - spojité, diskrétne**) premenné majú najširšie použitie. Na popísanie vzťahu medzi nimi môžeme prakticky použiť všetky matematické operácie.

kvalitatívne (alebo aj **kategoriálne**) premenné sú charakterizované vzťahom, pri ktorom je možné určiť, či sú dané hodnoty rovnaké alebo sa líšia. Vyjadrujú vlastnosti, ktoré sa musia vyjadriť vo forme kategórie, napr. chorý/zdravý; bez príznaku/s príznakom.

- **ordinálna** premenná – typ kategoriálnej premennej; jej hodnoty ale môžeme vzájomne zoradiť, je však obtiažne kvantifikovať jej hodnoty (napr. bolesť zanedbateľná, malá, stredná, veľká, neznesiteľná)

- **nominálna** premenná – typ kategoriálnej premennej; tentokrát ale nejde jej hodnoty zoradiť podľa veľkosti (napr. príslušnosť študenta UK k určitej fakulte),

  - špeciálnym typom nominálnej premennej je tzv. **dichotomická** premenná, ktorá podobne ako premenná binárna, či logická nadobúda dve hodnoty, ktoré sa navzájom vylučujú. V tomto prípade nejde určiť ich veľkosť, napr. pohlavie žena/muž

binárne (**logické**) premenné nadobúdajú dve diskrétne hodnoty. Špecifikom binárnej premennej je, že sa na ňu dá pozerať ako na najjednoduchší prípad intervalovej premennej s dvoma dielmi na škále. Z toho dôvodu je niekedy považovaná za kvantitatívnu premennú.

1.2 Delenie metód

V literatúre bolo navrhnuté množstvo prístupov, ktoré analyzujú dátové súbory. Veľmi dôležitú rolu medzi nimi hrajú metódy **klasifikačné**, resp. **regresné**.

Pri **klasifikácii** sa snažíme klasifikovať neznámy objekt do konečného počtu predom daných kategórií. Týmto spôsobom môžeme napr. zistiť, či je daný druh zvieraťa prítomný alebo neprítomný na určitej lokalite, prípadne príkladom z finančného prostredia, či je daný klient potencionálnym klientom pre nový hypotečný produkt atď. Pri tvorbe modelu sa vychádza z už nameraných dát. Vyberú sa parametre, ktoré sú najvýznamnejšie pre dané kategórie závislej premennej a na základe týchto poznatkov sa vytvorí model, ktorý sa snaží popísať daný problém tak, aby bol schopný zaraďovať neznáme vzorky.[2]

Druhou skupinou sú metódy, ktorých závislá premenná má **kvantitatívny** charakter. V týchto prípadoch ide o **regresný problém**.[1] Použitím parametrických regresných metód získame rovnicu s odhadnutými regresnými koeficientami, ktoré môžeme použiť pre predpoveď hodnôt závislej premennej.[2]

Hlavným rozdielom v týchto technikách je teda typ závislej premennej.

- V klasifikačných metódach je závislá premenná **kategoriálneho** typu a pomocou tejto techniky modelujeme jej závislosť na jednej alebo viacerých nezávislých premenných.

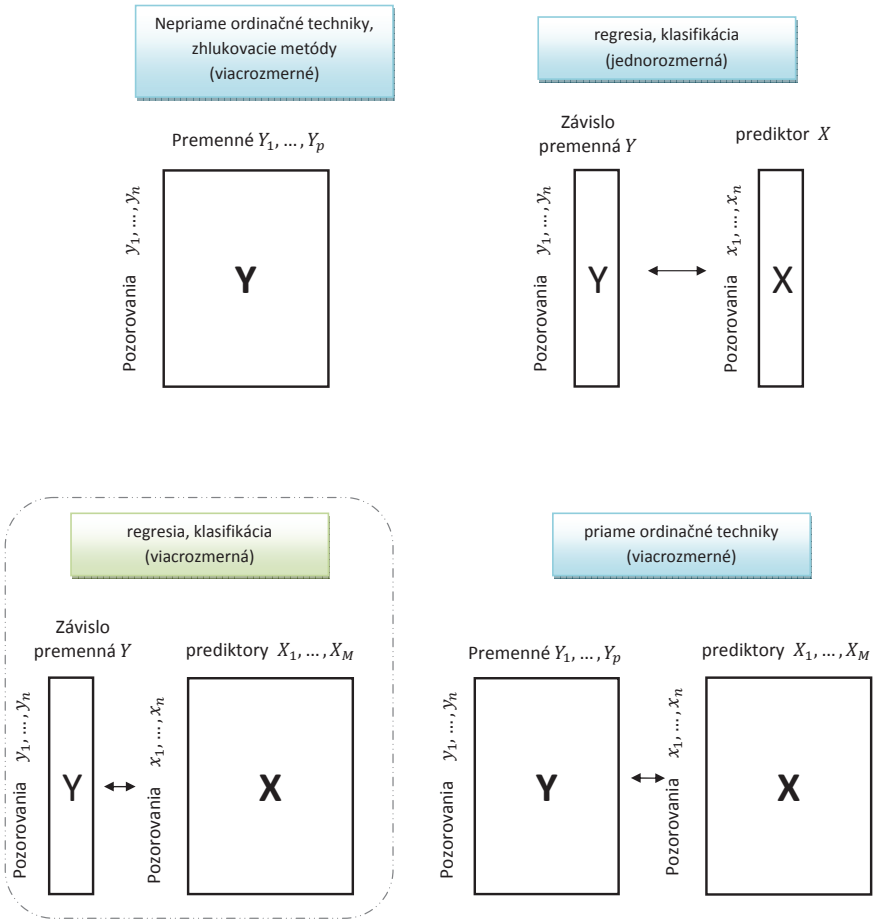
- Pri regresných metódach modelujeme závislosť **spojitej** závislej premennej na jednej, či viacerých nezávislých premenných.

V prípade, že v dátovom súbore chýba závislá premenná, využívajú sa metódy založené na vzdialenosti (podobnosti) jednotlivých pozorovaní, ako sú napríklad zhlukovacie algoritmy a ordinačné metódy. Delenie metód podľa charakteru dátových súborov na vstupe je zhrnutý názorne na obrázku 1.2.

V nasledujúcich kapitolách sa budeme venovať metódam, ktoré sú podľa spomenutého delenia viacrozmernými technikami pre klasifikáciu a regresiu.

[1] Spomeňme však, že v prípade logistickej regresie (a GLM modeloch) je odpoveď dichotomická a hľadáme model, ktorý by pacientov, žiadateľov o úver apod. čo najlepšie „rozdelil“ na tých, ktorí prežijú, či nie, ktorí zaplatia či nie apod. Takisto ide o regresné úlohy, je to však iný prístup k riešeniu.

[2] V rámci stromov CART, ktoré budú popísané v následujúcich kapitolách, je konštrukcia regresného stromu neparametrickou regresiou v zmysle tzv. regresogramu, kde delenie do tried je závislé na dátach.



Obr. 1.2: Rozdelenie metód podľa počtu závislých premenných a prediktorov

1.3 Kategorizácia klasifikačných metód

Klasifikácia je spojená s množstvom metód, na ktoré sa zameriavajú špecialisti z oblasti data miningu, z oblasti získavania znalostí z databáz, či špecialisti na neuronové siete. V jednej z prác v zborníku Robust z roku 2000 [3] nám autori ponúkajú podrobnú kategorizáciu týchto metód. Ide o delenie podľa rôznych hľadísk.[3]

1. Podľa ***predmetu klasifikácie***:
   (a) Objekty
   (b) Premenné
   (c) Kategórie premenných
      i. Jednej premennej (napr. zhluková analýza)
      ii. Dvoch premenných (dvojrozmerná zhluková analýza, korešpondenčná analýza)

[3]Často sa v literatúre klasifikačnými metódami považujú aj metódy zhlukovej analýzy. Je však veľmi dôležité rozlišovať medzi zhlukovou analýzou a klasifikáciou. **Zhlukovaním** budeme rozumieť postup, pri ktorom nie je predom známa príslušnosť žiadneho objektu a pri jeho použití je nutné stanoviť počet zhlukov, do ktorých bude analyzovaný súbor rozdelený.

iii. Viac premenných (optimálne škálovanie)

2. Podľa hľadiska, ***kedy a akým spôsobom je stanovený počet skupín***:

   (a) Počet stanovený pred analýzou

      i. Dané názorom užívateľa, ktorý analyzuje dáta (nehierarchická zhluková analýza)

      ii. Dané počtom hodnôt vysvetľovanej premennej (diskriminačná analýza)

         - vysvetľovaná premenná je dichotomická (logistická regresná analýza)
         - nominálna (multinomická logistická regresia, niekedy nazývaná aj polychotomická regresia)
         - ordinálna (ordinálna regresia)

   (b) Počet je zisťovaný analýzou, ktorej cieľom je klasifikácia

      i. Metódou môže byť navrhnutý konkrétny počet (faktorová analýza)

      ii. Počet stanovuje užívateľ na základe výsledkov analýzy (hierarchická zhluková analýza)

3. Podľa hľadiska, či ***sa môžeme pri klasifikácii riadiť nejakým vzorom*** (existuje učiteľ) ***alebo nie***:

   (a) Učenie s učiteľom (*supervised learning*), ktoré sa vzťahuje len na klasifikáciu objektov a známy počet skupín; slúži k odhadu hodnoty vysvetľovanej premennej, ktorá je kategoriálna. V literatúre data miningu sú ako klasifikačné označované len tieto metódy, ide teda o klasifikáciu v užšom význame:

      i. Diskriminačná analýza

      ii. Zobecnený lineárny model - GLM (*Generalized Linear Model*)

         A. Logistická regresná analýza

         B. Multinomická logistická regresná analýza

         C. Ordinálna regresná analýza

      iii. Kategoriálna regresná analýza

      iv. Klasifikačné stromy

         A. Metóda CHAID (*Chi-squared Automatic Interaction Detection*) - vysvetľovaná premenná môže byť nominálna, ale aj ordinálna (metóda je používaná i v prípade spojitej premennej)

         B. Metóda Exhaustive CHAID

         C. Metóda CART (*Classification and Regression Trees*) - vysvetľovaná premenná môže byť nominálna, ale aj ordinálna (metóda je používaná i v prípade spojitej premennej)

         D. Metóda QUEST (*Quick, Unbiased, Efficient Statistical Tree*) - len pre nominálnu vysvetľovanú premennú

         E. Ďalšie metódy (CLS, ID3, C4.5, AID,TREEDISC)

      v. Neuronové sítě

A. MLP (*MultiLayer Perceptrons*) - viacvrstvový perceptron
B. RBF (*Radial Basis Functions*) - radiálna bazická funkcia
C. PNN (*Probabilistic Neural Networks*) - pravdepodobnostné neuronové siete
D. LNN (*Linear Neural Networks*) - lineárne neuronové siete
E. LVQ (*Learning Vector Quantization*) - vektorová kvantizácia

vi. GUHA - len pre kategorizované premenné.

(b) Učenie bez učiteľa (*unsupervised learning*), ktoré zahŕňa okrem zhlukovania či segmentácie (objektov, premenných i kategórií) aj redukciu dát (premenných či kategórií). V literatúre data miningu se tieto metódy neoznačujú ako klasifikačné, ale spadajú do skupiny postupov, ktorých cieľom je zhlukovanie, prípadne segmentácia:

i. Zhluková analýza
A. Hierarchická zhluková analýza
B. Nehierarchická zhluková analýza
C. Dvojrozmerná zhluková analýza (*two-way joining*)

ii. Faktorová analýza

iii. Viacrozmerné škálovanie

iv. Korešpondenčná analýza

v. Optimálne škálovanie (analýza homogenity, kategoriálna analýza hlavných komponent)

vi. Neuronové siete
A. AR (*Adaptive Resonance Theory*)
B. KFM (*Kohonen Feature Maps*) - Kohonenove mapy
C. HNN (*Hopfield like Neural Network*) - sieť Hopfieldovho typu

vii. Genetické algoritmy

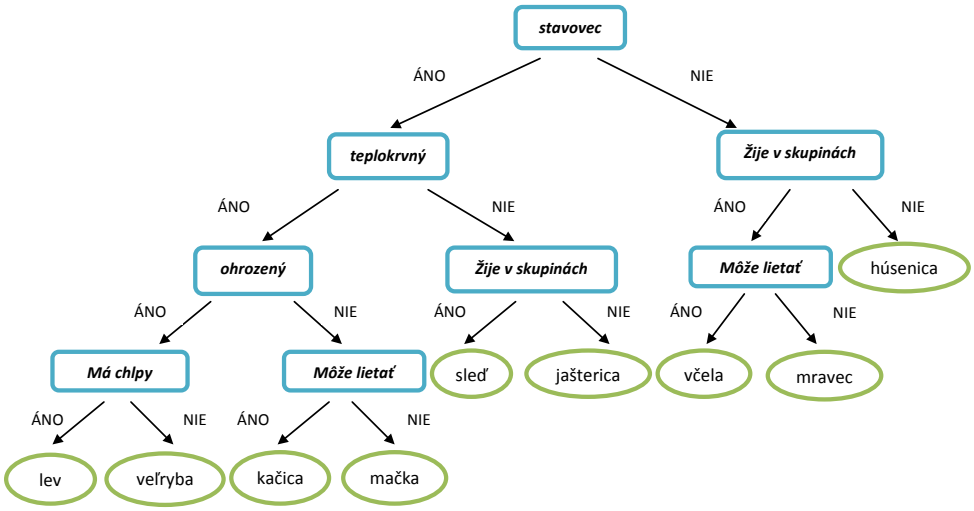
Kapitola 2

Rozhodovacie stromy

Jedným zo známych postupov pri riešení problémov regresnej analýzy alebo klasifikácie sú metódy založené na binárnych stromoch a rekurzívnom delení priestoru vysvetľujúcich premenných.

Rozhodovací strom je tvorený súborom hierarchicky usporiadaných rozhodovacích pravidiel. Príklad jednoduchého stromu môžeme vidieť na obrázku 2.1, ktorý zobrazuje rozdelenie zvierat podľa rôznych kritérií z tabuľky 2.1. S touto stromovou štruktúrou sa stretávame pomerne často aj v bežnom živote, pretože je čitateľná a ľahko interpretovateľná. Príkladom je napríklad usporiadanie zložiek systému v počítači. U rozhodovacích stromov je zrejmá analógia s reálnymi stromami v prírode, preto bola prevzatá aj terminológia. Teda podobne ako pri reálnom strome hovoríme o tom, že rozhodovací strom rastie, vetví sa a prerezávame ho.

Celý vstupný súbor dát predstavuje **koreň** rozhodovacieho stromu. Ten sa postupne delí na **uzly**, teda hovoríme, že strom rastie. Podľa toho, či sa stromy vetvia na dve alebo viacero vetví, ich rozdeľujeme na **binárne** a **nebinárne** stromy. Koncové uzly, ktoré sa už nedelia sú označované ako **listy**.



Obr. 2.1: Rozhodovací strom

Rozhodovacie stromy rozdeľujeme podľa typu závislej premennej na regresné a klasifikačné stromy. Označme T strom s koncovými uzlami $t = (t_1,...,t_N)$. Pri

	teplokrvný	**môže lietať**	**stavovec**	**ohrozený**	**žije v skupinách**	**má chlpy**
mačka	ÁNO	NIE	ÁNO	NIE	NIE	ÁNO
kačica	ÁNO	ÁNO	ÁNO	NIE	ÁNO	NIE
sleď	NIE	NIE	ÁNO	NIE	ÁNO	NIE
lev	ÁNO	NIE	ÁNO	ÁNO	ÁNO	ÁNO
jašterica	NIE	NIE	ÁNO	NIE	NIE	NIE
veľryba	ÁNO	NIE	ÁNO	ÁNO	ÁNO	NIE
mravec	NIE	NIE	NIE	NIE	ÁNO	NIE
včela	NIE	ÁNO	NIE	NIE	ÁNO	ÁNO
húsenica	NIE	NIE	NIE	NIE	NIE	ÁNO

Tabuľka 2.1: Kritéria pre rozdelenie živočíchov

klasifikačnom strome sú pozorovania kategoriálnej závislej premennej Y s J kategóriami zaradené do niektorej z kategórií $c = (c_1,..,c_J)$, kde $J \geq 2$. Ak je závislá premenná spojitá $Y = (y_1,...,y_n)$, pozorovaniam je priradená hodnota predikovaná modelom $\hat{y}_i$ a výsledný strom bude regresný.

Do jednotlivých uzlov sú pozorovania premennej Y rozdelené na základe hodnôt vysvetľujúcich premenných (prediktorov) $X_1,...,X_M$. Samotné rozdelenie je znázornené graficky pomocou vetiev stromu. Ak máme prediktory kategoriálneho typu, ako v strome na obrázku 2.1, hodnoty y_i sú rozdelené podľa kategórií prediktoru X. Vidíme napríklad v prvom delení s prediktorom *stavovec* a dvoma kategóriami ÁNO a NIE sa premenná Y rozdelí do dvoch dcérskych uzlov. Pre kategóriu prediktoru ÁNO obsahuje prvý uzol kačicu, mačku, veľrybu, jaštericu a sleďa. V prípade nášho stromu odpovedáme pri delení na následujúcu otázku: Ktoré pozorovanie y_i patrí do množiny, kde $x_i \in A$, pričom A je neprázdna vlastná podmnožina množiny všetkých hodnôt veličiny X?

V prípade spojitého prediktoru rozdeľujeme Y pomocou hodnoty a daného prediktoru X. V tomto prípade patria pozorovania y_i do prvého uzlu, ak platí, že $x_i \geq a$ a do druhého uzlu, ak $x_i < a$. Príkladom takéhoto delenia môže byť napríklad určenie bonity klientov (závislá premenná) na základe ich výšky príjmu (prediktor). Rozdelenie by mohlo dopadnúť následovne: pri príjme $x \geq 30000$ Kč by sa jednalo o bonitných klientov a $x < 30000$ Kč o priemerných klientov. Samozrejme k presnejšiemu rozdeleniu týchto kategórií by bolo potrebné uvažovať viac prediktorov, ktoré majú dopad na posúdenie bonity klienta, ako napríklad kontokorentné úvery, limity kreditných kariet atď.

K danému vetveniu stromu je použitý vždy jeden prediktor. Rovnaký prediktor ale môže byť použitý aj v ďalšom vetvení. Toto pravidlo vedie k tomu, že každé pozorovanie y_i tak patrí len do jedného terminálneho uzla a je mu buď priradená kategória (klasifikačný strom) alebo priemer hodnôt (regresný strom) závislej premennej Y tohoto uzlu.

Stromy nekladú nároky na rozdelenie dát, ako napríklad normálne rozdelenie, konštantný rozptyl alebo nezávislosť prediktorov. Parametre algoritmu sú často určené experimentálne testovaním rôznych nastavení ich hodnôt. Tento postup však skrýva nebezpečenstvo najmä pri kalibrácii modelu, ktorá môže byť do istej miery subjektívna a závisí na skúsenosti bádateľa.[2]

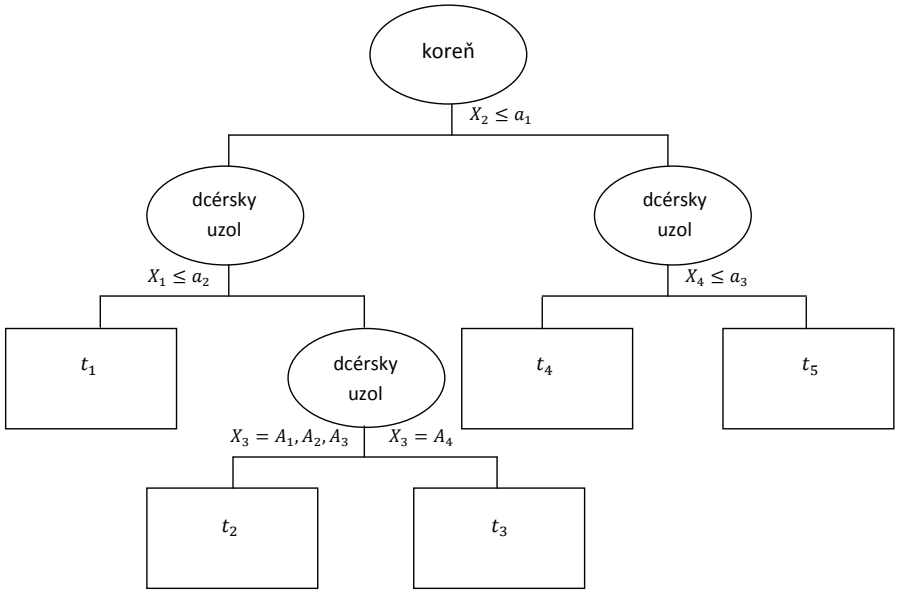
2.1 Metodológia CART (Classification and regression trees)

Metodológia vytvárania binárnych stromov je v literatúre označovaná pod pojmom CART, čo v preklade znamená klasifikačné a regresné stromy. CART patrí medzi najznámejšie a najpoužívanejšie algoritmy a je základným predstaviteľom binárnych stromov. Na tejto metóde sú vysvetlené základné princípy tvorby stromov, pretože ostatné stromy dostaneme modifikáciou pravidiel stromu CART.

Existuje mnoho ďalších metód pre tvorbu stromov, ktoré sa od klasických binárnych rozhodovacích stromov líšia. Napr. algoritmus pre nebinárne stromy CHAID pre kategoriálne a ordinálne premenné alebo metódy PRIM a MARS pre stromy určené pre regresné problémy. Posledné dve metódy nemajú grafický výstup pomocou znázornenia pravidiel. Výstupom metódy PRIM je sada rozhodovacích pravidiel bez stromovej štruktúry a u metódy MARS je to regresná rovnica.[2]

Principiálne je tvorba stromu vo všetkých algoritmoch veľmi podobná a líši sa predovšetkým v nájdení vhodného prediktoru X pre každú hierarchickú úroveň stromu a hodnoty tohto prediktoru a pre rozdelenie závislej premennej Y. Strom na obrázku 2.1 je teda príkladom binárneho klasifikačného rozhodovacieho stromu s kategoriálnou závislou premennou Y označujúcou kategórie živočíchov a kategoriálnymi prediktormi X_i ich vlastností.

Stromy typu CART sú vhodné pre kategoriálne i regresné úlohy a ako bolo spomenuté, rastú na základe rekurzívneho delenia. Stručný popis delenia uvedený v odstavcoch vyššie podľa rôznych typov prediktorov sme graficky zhrnuli na obrázku 2.2, ktorý zobrazuje všeobecnú štruktúru stromu CART. Začíname je-

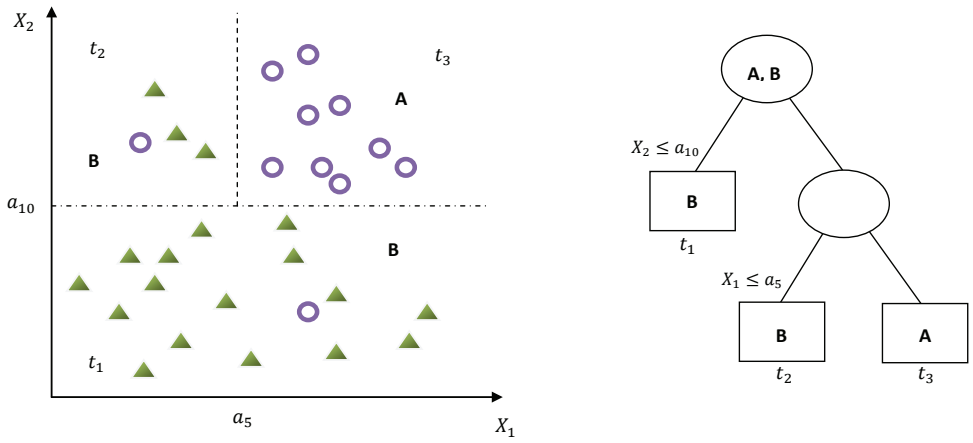


Obr. 2.2: Štruktúra všeobecného rozhodovacieho stromu CART

diným uzlom, koreňom, kam patria všetky pozorovania súboru. Následne sa tieto pozorovania delia do dcérskych uzlov, na základe hodnoty a_i prediktoru X_j (resp. kategórie A_i prediktoru X_j) a sú dalej opäť binárne delené na dalšie uzly. Indexy pri koncových uzloch, listoch, udávajú, v akom poradí došlo k oddeleniu. Predik-

tory X_1, X_2 a X_4 sú spojité a prediktor X_3 je kategoriálneho typu s kategóriami A_1,A_2,A_3,A_4.

Hodnoty a_i vysvetľujúcich premenných rozdeľujú daný priestor na sadu pravouholníkov, ktoré označujeme ako regióny R_t. Na obrázku 2.3 je príklad rozdelenia pozorovaní do kategórií A a B závislej premennej Y s použitím dvoch spojitých prediktorov X_1, X_2.



Obr. 2.3: Rozdelenie na regióny

Snažíme sa o také rozdelenie závislej premennej Y prediktorom X, aby hodnoty pozorovania y_i boli vnútri uzlu čo najhomogénnejšie a medzi uzlami čo najrozdielnejšie. Ktorý prediktor a jeho hodnota nám zaistí najlepšie rozdelenie nám určuje **kriteriálna štatistika** (*splitting criterium*), ktorá určuje homogenitu uzlu (*node impurity*). Kriteriálne štatistiky sa líšia podľa toho, či ide o klasifikačný alebo regresný strom. Podrobnejšie sa jednotlivým kritetiálnym štatistikám venujeme v sekciách 2.1.1 a 2.2.

Vlastná metodológia CART rozhodovacích stromov bola vytvorená prevažne pre potrebu klasifikácie. Zatiaľ čo v klasifikácii je použitie stromov bežne využívané, v regresii tomu tak nie je. Preto v literatúre často nájdeme rozobranú len túto situáciu a použitie stromov v regresii je uvádzané ako špeciálny prípad. Oba prístupy predstavil profesor Antoch v práci [4], kde sa venuje predovšetkým použitiu stromov v regresii. V následujúcich odstavcoch sa budeme zaoberať metodológiou CART pre konštrukciu klasifikačných stromov a následne pojednáme o použití regresných stromov a o rozdieloch v týchto prístupoch.

Uvažujme situáciu, kedy na jednotlivých objektoch meráme M znakov, vždy v tom istom poradí. Nech náhodná veličina X_i, $i = 1,...,M$, popisujúca chovanie znaku i, môže byť buď merateľného alebo kategoriálneho typu a predpokladajme, že môže nadobúdať hodnoty z $\mathcal{X}_i \subseteq \mathbb{R}_i$. Položme $\mathbf{X} = (X_1,...,X_M)$, potom všetky možné hodnoty vektoru $\mathbf{X}$ padnú do priestoru $\mathcal{X} = \mathcal{X}_1 * ... * \mathcal{X}_M \subseteq \mathbb{R}_M$. Ďalej predpokladajme, že každý objekt môže byť zaradený do práve jednej z J tried. Jednotlivé triedy označíme identifikátormi z množiny $C = (1,...,J)$.

Podobne ako v iných klasifikačných problémoch, naším cieľom je nájsť klasifikátor (klasifikačné pravidlo), tzn. systematickú cestu umožňujúcu predpovedať, ktorý objekt patrí do ktorej triedy z množiny C.

Pri konštrukcii klasifikátoru musíme mať vždy na pamäti dva hlavné ciele:

- nájsť čo najpresnejší klasifikátor

- odhaliť predpovednú štruktúru problému

Na ľubovoľný klasifikátor sa môžeme pozerať dvoma ekvivalentnými spôsobmi:

1. Ako na niektorú funkciu $d : \mathcal{X} \mapsto C$, tak, že $\forall \mathbf{x} \in \mathcal{X} : \ d(\mathbf{x})$ je rovná jednej a práve jednej hodnote z C, resp.;

2. Ako na niektorý rozklad priestoru $\mathcal{X}$ na J navzájom disjunktných podmnožín A_j takých, že $\forall \mathbf{x} \in A_j : \ d(\mathbf{x}) = j$. Tzn. že $A_j = \{\mathbf{x} \in \mathcal{X} |\ d(\mathbf{x}) = j\}$ a $\bigcup_{j=1} A_j = \mathcal{X}$, $A_i \cap A_j = \emptyset$ pokiaľ $i \neq j$. Jednotlivé podmnožiny A_j pritom nemusia byť nutne súvislé.

Cieľom je skonštruovať klasifikátor typu binárneho stromu, tzn. klasifikátor, pri ktorého použití závisí klasifikácia na zodpovedaní konečného počtu dichotomických otázok. Pri jeho konštrukcii bude nutné zodpovedať na následujúce otázky:

1. Voľba miery kvality klasifikátoru a spôsob jeho odhadu.

2. Stanovenie množiny otázok, podľa ktorých sú delené podmnožiny priestoru $\mathcal{X}$.

3. Výber pravidla priraďujúceho index triedy pre podmnožiny, ktoré už ďalej nebudú delené.

4. Nájdenie pravidla pre výber optimálneho delenia danej podmnožiny.

5. Určenie pravidla pre ukončenie delenia.

2.1.1 Voľba miery kvality klasifikátoru

Kriteriálna štatistika pre klasifikačné stromy je založená na pomere kategórií závislej premennej v potenciálnych uzloch. Najpoužívanejšími mierami kvality klasifikátorov sú *Giniho Index (GI)*, *Entropia(H)* a *klasifikačná chyba (ME)*.

$$GI = \sum_{j=1}^{J} p_{tj}(1 - p_{tj}) = 1 - \sum_{j=1}^{J} p_{tj}^2 \tag{2.1}$$

$$H = -\sum_{j=1}^{J} p_{tj} \log_2 p_{tj} \tag{2.2}$$

$$ME = 1 - \max p_{tj} \tag{2.3}$$

kde p_{tj} je podiel pozorovaní y_i s kategóriou j v uzle t z celkového počtu všetkých pozorovaní y_i v tomto uzle alebo, inými slovami, pravdepodobnosť kategórie j v uzle t.

2.1.2 Stanovenie množiny otázok

V praxi bola stanovená množina tzv. štandardných otázok Q definovaná následovne:

- Každé delenie závisí len na hodnote práve jednej premennej.
- Ak je X_i merateľná náhodná veličina, zvoľme otázky tvaru

$$\{Je\ X_i \leqslant c\ ?\}, \quad c \in \mathbb{R}_1.$$

  Vzhľadom k tomu, že máme konečný počet pozorovaní, možno určiť konečný počet konštánt c a teda aj konečný počet otázok. Konštanty c môžeme napríklad voliť ako stredy intervalov určených po sebe idúcimi pozorovaniami.

- Ak je X_i kategoriálna náhodná veličina nadobúdajúca hodnôt z množiny B, potom Q zahŕňa otázky tvaru

$$\{Je\ X_i \in S\ \ ?\}, \quad S\ \subseteq B.$$

Základnou nevýhodou množiny Q a následného delenia je, že nepokrýva lineárne závislosti. Preto bolo potrebné rozšíriť túto množinu aj o otázky, pokrývajúce lineárne kombinácie merateľných premenných v tvare

$$\left\{ Je\ \sum_i a_i X_i \leqslant c\ ? \right\}, \quad a_i, c \in \mathbb{R}_1,$$

a o otázky pokrývajúce boolovské kombinácie kategoriálnych premenných. S voľbou týchto zložitejších otázok vzrastá aj zložitosť výpočtu, pretože počet takýchto kombinácií je vysoký.

Vidíme, že otázky sú dichotomického[1] typu. Označme t ako podmnožinu priestoru $\mathcal{X}$, tj. $t \subseteq \mathcal{X}$. Potom každej otázke $q \in Q$ zodpovedá práve jedno delenie množiny t na podmnožiny t_L a t_R také, že

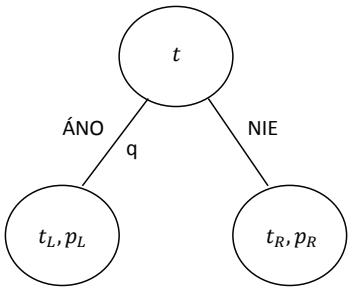
$$t_L \cup t_R = t \qquad \text{a} \qquad t_L \cap t_R = \emptyset. \tag{2.4}$$

Pre zjednodušenie predpokladajme, že do t_L budeme zaraďovať pozorovania z t také, pre ktoré je odpoveď na otázku $q \in Q$ ÁNO, analogicky do t_R ostatné pozorovania z množiny t. Schematicky toto delenie zobrazuje obrázok 2.4, kde p_L (resp. p_R) označuje podiel pozorovaní z t, ktoré padnú do podmnožiny t_L (resp. t_R).

2.1.3 Výber pravidla priraďujúceho index triedy

Výber pravidla priraďujúceho index triedy pre podmnožiny, ktoré už ďalej nebudú delené je skôr technickej povahy. Prax ukazuje, že výber pravidla nehrá v celej konštrukcii zďaleka tak kľúčovú rolu ako odpoveď na poslednú otázku, tj. určenie pravidla pre ukončenie delenia.[4]

[1] umožňujú len jednu z dvoch možných odpovedí napr. áno - nie, viem - neviem



Obr. 2.4: Schéma delenia na podmnožiny

2.1.4 Nájdenie pravidla pre výber optimálneho delenia danej podmnožiny

Základná idea pri delení niektorej množiny $t \subseteq \mathcal{X}$ na podmnožiny t_L a t_R spĺňujúce 2.4 spočíva v rozdelení množiny t na podmnožiny tak, aby dáta v oboch podmnožinách boli „čistejšie“, teda *homogennejšie* z hľadiska klasifikácie než v množine t. Znamená to, že množiny t_L a t_R by mali umožniť klasifikáciu objektov, ktoré do nich padnú minimálne tak presne ako do t, pokiaľ možno presnejšie. Optimálny rozklad pre daný uzol sa hľadá principiálne jednoduchým spôsobom, ale výpočetne náročným. Mechanicky sa preberú všetky možnosti, pre každú z nich sa z dát vypočíta kriteriálna štatistika a vyberie sa ten rozklad, ktorý túto štatistiku maximalizuje.

Najčastejšou kriteriálnou štatistikou pre stromy typu CART je Giniho index. Hodnota indexu sa rovná nule, ak je v koncovom uzle jediná kategória závislej premennej Y. Naopak, ak je v koncovom uzle v každej kategórii premennej Y rovnaký počet pozorovaní, potom Giniho index dosahuje maximum.

Pri rozdelení na dva dcérske uzly sa pre každý uzol spočíta GI. Celková hodnota Giniho indexu GI_{celk} pre rozdelenie je potom rovná váženému súčtu GI_i dcérskych uzlov. Pre binárny strom s dvoma dcérskymi uzlami platí, že

$$GI_{celk} = \sum_{i=1}^{2} \frac{N_i}{N_t} GI_i, \tag{2.5}$$

kde N_t je počet pozorovaní v materskom uzle a N_i počet v dcérskom.

Pre celkovú entropiu H_{celk} a celkovú klasifikačnú chybu ME_{celk} by sme použili analogický vzťah. Entropia H poskytuje veľmi podobné výsledky ako Giniho index GI. Maximum je dosiahnuté pri rovnomernom zastúpení kategórií premennej Y a minimum, ak pozorovanie v uzle náleží len do jednej kategórie.

Klasifikačná chyba ME vyjadruje podiel chybne klasifikovaných pozorovaní, teda $1 - ME$ predstavuje celkovú presnosť stromu. Táto miera sa využíva najmä k finálnemu meraniu presnosti stromu.

2.1.5 Určenie pravidla pre ukončenie delenia

V prípade klasifikácie sa ukazuje najdôležitejšou odpoveď na 5. otázku, tj. určenie pravidla pre ukončenie delenia. Strom nemôže rásť do nekonečna, jeho veľkosť je obmedzená veľkosťou vstupného súboru. Existujú však určité pravidlá, kedy sa rast zastaví skôr.

Rast sa sám zastaví v týchto prípadoch:

- V každom liste je len jedno pozorovanie.
- Všetky pozorovania v uzle majú rovnakú hodnotu všetkých prediktorov.
- Všetky pozorovania v uzle majú rovnakú hodnotu závislej premennej.

Strom môžeme obmedziť v raste nastavením niektorých parametrov a k ďalšiemu rozdeleniu už nedôjde, ak sme dosiahli jednu z týchto zadaných hodnôt:

- maximálny počet vetvení daného stromu
- maximálnu počet pozorovaní v koncovom uzle
- frakcia pozorovaní v uzle, ktorá už nemôže byť oddelená
- veľkosť chyby v potencionálnych dcérskych uzloch je menšia než určitá zadaná prahová hranica

2.1.6 Výber optimálneho stromu

Uvedené pravidlá v predchádzajúcom odstavci, pri ktorých sa rast stromu zastaví, môžu byť dosť subjektívne a ovplyvňujú veľkosť získaného stromu. Existuje niekoľko prístupov, ktoré nám pomáhajú pri určení správnej veľkosti stromu.

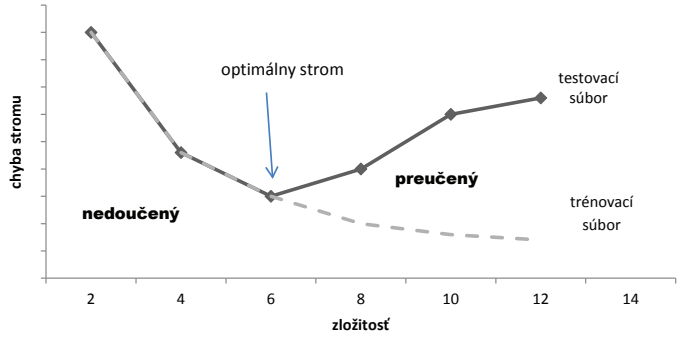
2.1.6.1 Metóda testovacieho súboru

Jednou z metód výberu optimálneho riešenia je rozdelenie súboru na trénovaciu a testovaciu časť, ak máme k dispozícii dostatočný počet pozorovaní. Zvykom je, že naša minulá skúsenosť potrebná pre konštrukciu klasifikátoru je sústredená v tzv. **trénovacom (učebnom) súbore**, tj. množina meraní znakov X_1,...,X_M na n objektoch (indivíduách) spolu s informáciou o ich konkrétnom zaradení (skutočnej klasifikácii). Inými slovami, pre nás bude učebným súborom množina dvojíc $\mathcal{L} = \{(\mathbf{x}_1, j_1),...,(\mathbf{x}_N, j_N)\}, \quad \mathbf{x}_i \in \mathcal{X}, \quad j_i \in C, \quad i = 1,...,N$, kde $\mathbf{x}_i$ a j_i poznáme. Ak to zhrnieme, na trénovacom súbore sa klasifikačný strom učí a rastie a testovací súbor slúži len k jeho otestovaniu.

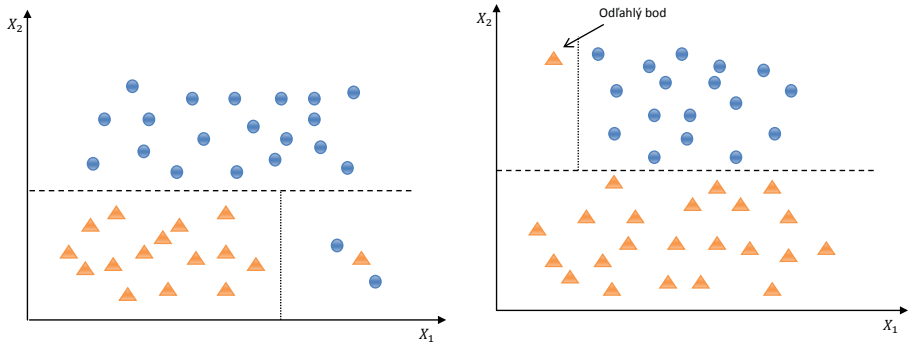
Pokiaľ je strom nedoučený *(underfitting)*, je príliš jednoduchý a chyba na testovacom aj trénovacom súbore bude vysoká. V opačnom prípade, ak je strom pretrénovaný *(overfitting)*, strom je zbytočne zložitý a chyba na trénovacom súbore je väčšinou malá, no na testovacom obvykle veľká. V týchto prípadoch je potrebné nájsť vhodný kompromis. Na obrázku 2.5 vidíme voľbu počtu terminálnych uzlov optimálneho stromu pomocou chyby na testovacom a trénovacom súbore.

Väčšie riziko, že dôjde k pretrénovaniu hrozí pri zložitejších modelov. Preto je dôležité pri optimálnej tvorbe stromu zahrnúť taktiež jeho zložitosť. Všeobecne platí pravidlo nazývané Occamova britva. Ak máme dva modely s podobnou chybou, vhodnejšie je vybrať ten menej zložitý.

Na nasledujúcom obrázku (2.6) si ukážeme, za akých okolností môže dôjsť k pretrénovaniu stromu. V oboch prípadoch uvažujeme spojité prediktory X_1



Obr. 2.5: Výber optimálneho stromu



(a) Nedostatok dát (b) Odľahlá hodnota

Obr. 2.6: Pretrénovanie stromu

a X_2, ktorých hodnoty rozdeľujú pozorovania y_i závislej premennej Y na tri regióny, predstavujúce koncové uzly vytvoreného stromu. Ide skutočne o veľmi zjednodušené grafické príklady pretrénovania.

V prvom prípade (viď 2.6a) vidíme, že pretrénovanie spôsobuje nedostatočný počet trénovacích dát v ľavej dolnej časti grafu. Chýbajúce hodnoty spôsobia obtiažnu predikciu správnej kategórie v tomto regióne pre testovacie pozorovania.

V druhom prípade (viď 2.6b) pozorujeme odľahlú hodnotu v prípade trénovacích dát, ktorá spôsobila vytvorenie ďalšieho rozhodovacieho pravidla a nový koncový uzol.

2.1.6.2 Prerezávanie stromu a krížové overovanie

Nie všetky pravidlá, ktoré sme uviedli v sekcii 2.1.5 nám ponúknu prijateľný výsledok, ako je napríklad prípad jedného pozorovania v každom liste. Veľkosť stromu je parameter, ktorý určuje zložitosť modelu. Ako bolo spomenuté, optimálna veľkosť stromu by mala byť adaptívne vybraná z dát. Po mnohých pokusoch sa v tejto metodológii pristúpilo k nasledujúcemu preferovanému postupu:

1. Priestor $\mathcal{X}$ je najskôr rozložený postupnosťou rekurzívnych delení (popísaných vyššie) na mnoho čo najmenších podmnožín, teda necháme vypestovať veľký strom T_0 a zastavenie delenia budeme spracovávať len vtedy, ak je dosiahnutá určená minimálna veľkosť uzla (napríklad 5). Ak máme k dispozícii dostatočnú pamäť počítača, každá výsledná podmnožina T_{max} bude obsahovať práve jedno pozorovanie.

2. Následne je aplikovaný algoritmus pomocou kritéria zložitosti stromu (*cost-complexity criterium*) kolapsujúci (rekombinujúci) počiatočný rozklad spätne do množiny $\mathcal{X}$.

Definujme podstrom $T \subset T_0$ každý strom, ktorý je možné získať prerezávaním T_0. To znamená strom, ktorý dostaneme kolapsovaním ľubovoľného počtu jeho vnútorných uzlov. Kritérium zložitosti stromu vyjadríme vzťahom:

$$C_\alpha(T) = DT + \alpha \mid T \mid, \tag{2.6}$$

kde $\mid T \mid$ je počet koncových uzlov v podstrome T a DT je chyba stromu. Pri kolapsovaní podstromov sa používajú opäť tie isté miery pre kvalitu klasifikátorov, ako pri konštrukcii, modifikované však o člen $\alpha|T|$, penalizujúci nás za príliš rozsiahle rozklady. Výsledkom je postupnosť do seba vnorených rozkladov priestoru $\mathcal{X}$, začínajúca podmnožinou T_{max} a končiaca samotným priestorom $\mathcal{X}$. Z tejto množiny je potrebné vybrať optimálne riešenie.

Parameter $\alpha \geq 0$ vyjadruje kompromis medzi veľkosťou stromu a jeho presnosťou. Pre každé α hľadáme taký strom $T_\alpha \subseteq T_0$, ktorý minimalizuje $C_\alpha(T_\alpha)$. K určeniu odhadu α sa používa 5-násobné alebo 10-násobné krížové overovanie. Pomocou krížového overovania vyberieme také $\hat{\alpha}$, aby mal strom čo najväčšiu presnosť, ale zároveň, aby rozdiel chyby medzi testovacím a trénovacím súborom pri krížovom overovaní bola čo najmenšia.

Krížové overovanie v klasifikácii

V prípade, že nemáme k dispozícii dostatočný počet pozorovaní, vhodnejším prístupom určenia optimálneho riešenia je tzv. krížové overovanie (*cross validation*). Tento postup patrí medzi validačné techniky. Daný súbor slúži k vytvoreniu modelu a zároveň k jeho testovaniu. Tento postup vyžaduje viac pamäte počítača a takisto spotrebuje viac času, pretože v každom kroku je potrebné prejsť množstvo pomocných riešení.

Celý dátový súbor $\mathcal{L}$, pozostávajúci z N pozorovaní, je rozdelený na K nezávislých častí, ako môžeme vidieť na obrázku (2.7). Nech $\mathcal{L} = \mathcal{L}_1 \cup ... \cup \mathcal{L}_K, \quad \mathcal{L}_i \cap \mathcal{L}_j = \emptyset, \quad i \neq j$, potom $\mathcal{L} \setminus \mathcal{L}_k$ použijeme ku konštrukcii klasifikátoru $d(\mathbf{x})$ a $\mathcal{L}_k$ ku kontrole jeho kvality.

Nech $\kappa : \{1,...,N\} \to \{1,...,K\}$ je indexová funkcia vyjadrujúca časť dátového súboru, do ktorej je alokované i-té pozorovanie. Označme $d(\mathbf{x},\mathcal{L}_k), \quad k = 1,...,K$, klasifikátor skonštruovaný na množine $\mathcal{L} \setminus \mathcal{L}_k$ a aplikovaný na množinu $\mathcal{L}_k$.

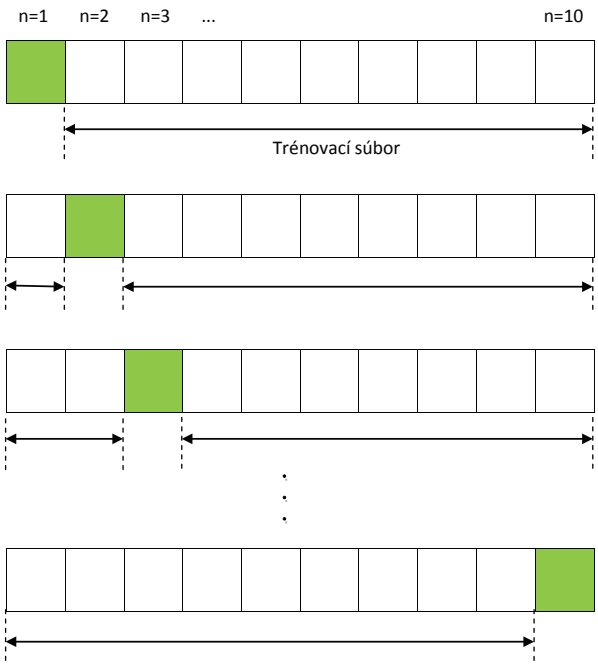
Potom odhad klasifikačnej chyby pri krížovom overovaní je definovaný vzťahom

$$CV(d) = \frac{1}{N} \sum_{i=1}^{N} I(d(\mathbf{x}_i,\mathcal{L}_{\kappa(i)}) = j_i),$$

kde I je funkcia identifikátoru.

Uvažujme množinu modelov $\{T_\alpha = d_\alpha(\mathbf{x})\}$, kde rovnako ako v predchádzajúcom vzťahu predstavuje $d_\alpha(\mathbf{x}, \mathcal{L}_k)$ klasifikátor zostrojený na množine $\mathcal{L} \setminus \mathcal{L}_k$ aplikovaný na množinu $\mathcal{L}_k$. Pre túto množinu modelov definujeme

$$CV(d,\alpha) = \frac{1}{N} \sum_{i=1}^{N} I(d_\alpha(\mathbf{x}_i,\mathcal{L}_{\kappa(i)}) = j_i).$$



Obr. 2.7: 10-násobné krížové overovanie

Funkcia $CV(d,\alpha)$ vyjadruje odhad krivky klasifikačnej chyby a my hľadáme parameter $\hat{\alpha}$, ktorý ho minimalizuje. Náš finálny vybraný model bude $T_{\hat{\alpha}} = d_{\hat{\alpha}}(\mathbf{x})$.

Typickou voľbou K je 5 alebo 10. V prípade, ak $K = N$, potom pre i-té pozorovanie je klasifikačný strom zostrojený pomocou všetkých pozorovaní okrem i-tého. Tento prípad je nazývaný anglicky *leave-one-out* krížové overovanie.

2.1.6.3 Presnosť stromu

Označme $e(t)$ chybu na trénovacom súbore (*resubstitution errors*) a $e'(t)$ chybu na testovacom súbore (*generalization errors*).

Ak by sme použili len trénovací súbor, potom dostaneme dva odhady celkovej chyby stromu [2]. Pesimistický odhad dostaneme vtedy, ak pre každý koncový uzol stromu platí, že $e'(t) = e(t) + 0{,}5$ [2]. Potom odhad celkovej chyby je

$$e'(T) = e(T) + K * 0{,}5,$$

kde K je počet koncových uzlov. Optimistický odhad dostaneme, ak predpokladáme, že chyba trénovacieho súboru je zhodná s chybou na súbore testovacom, tj. $e'(t) = e(t)$.

Príklad 2.1.1. Uvažujme strom s 20-timi terminálnymi uzlami a 10-timi chybne zaradenými pozorovaniami z trénovacieho súboru. Nech súbor obsahuje 100 meraní. Potom pesimistický odhad chyby je rovný $\frac{(10+20*0.5)}{100} = 20\%$ a optimistický $\frac{10}{100} = 10\%$.

[2]Správnosť daného vzťahu je podľa nášho názoru diskutabilná, keďže ako sa ukáže v praktickej aplikácii, chyba na testovacom súbore oproti chybe na trénovacom súbore nikdy nebola tak odlišná. Tvrdením sa odvolávame na literatúru [2]

K odhadu všeobecnej celkovej chyby stromu sa však oproti odhadu chyby na trénovacom súbore častejšie využíva testovací súbor, ktorý je vhodnejším ukazovateľom, ako dobre bude schopný vytvorený strom klasifikovať nové dáta.

Odhad pravdepodobnosti, že pozorovanie je správne klasifikované, udáva tzv. **celková správnosť** OA (*overall accuracy*)

$$OA = \frac{N_p}{N}, \tag{2.7}$$

kde N_p je počet správne klasifikovaných pozorovaní z celkového počtu N pozorovaní. Nezohľadňuje však rôznu veľkosť skupín ani rozdielnosť oproti náhodnému výsledku, a preto môže často dôjsť k podhodnoteniu alebo nadhodnoteniu kvality modelu.

Príklad 2.1.2. Uvažujme príklad jednoduchého klasifikačného stromu s dvomi kategóriami A_1,A_2. Nech počet pozorovaní v jednotlivých kategóriách je $A_1 = 100$ a $A_2 = 10$. Počet správne klasifikovaných pozorovaní je $A_1 = 100$ a $A_2 = 0$. Potom

$$OA = \frac{100}{110} \cong 0{,}91,$$

tj. percento správne klasifikovaných pozorovaní je zhruba 0.91. Vidíme však, že táto informácia pre nás nemá žiaden úžitok, pretože strom neodlíšil jednotlivé kategórie a klasifikuje pozorovania z kategórie A_2 ako pozorovania kategórie A_1.

Jednoduchou úpravou môžeme urobiť korekciu na veľkosť kategórií:

$$OA_{kateg} = \frac{1}{J} \sum_{j=1}^{J} \frac{N_{pj}}{N_j}, \tag{2.8}$$

kde J je celkový počet kategórií, N_{pj} je počet správne klasifikovaných pozorovaní v kategórii j a N_j je počet pozorovaní v kategórii j.

Príklad 2.1.3 (pokračovanie)**.** Percento správne klasifikovaných pozorovaní pre kategórie A_1,A_2 dostávame

$$OA_{kateg} = \frac{1}{2} \sum_{j=1}^{2} \frac{N_{pj}}{N_j} = \frac{1}{2} \left(\frac{100}{100} + \frac{0}{10} \right) = 0.5$$

Na základe celkovej správnosti OA môžeme vyjadriť optimistický odhad chyby stromu ako $e'(t) = 1 - OA$.

2.1.7 Primárne, zástupné a kompetitívne premenné

Rozhodovacie stromu môžu byť veľmi nestabilné a výsledný strom závisí na použitej kriteriálnej štatistike, na krížovom overovaní a nastavení parametrov rastu stromu. Pre rôzne trénovacie súbory pri krížovom overovaní môžeme dostať rôzne stromy. Okrem tohto prípadu, môžeme získať strom s iným vetvením použitím rôznych kombinácií prediktorov.

Predstavme si situáciu, kedy je pre prvé delenie vybraný najvýznamnejší prediktor a strom ďalej pokračuje v rozdeľovaní na ďalšie dcérske uzly. Čo keby

premenná X	kategória	uzol 1	uzol 2
primárna	A	90	10
	B	90	10
	C	20	80
zástupná	A	80	20
	B	85	15
	C	25	75
kompetitívna	A	80	20
	B	20	80
	C	10	90

Tabuľka 2.2: Určovanie primárnej, kompetitívnej a zástupnej premennej

sme však vybrali druhý najlepší prediktor, ktorý by pozorovania rozdelil úplne inak? Následne by asi boli použité iné prediktory pre ďalšie delenie. Takto vytvorený strom však môže mať rovnakú presnosť ako pôvodný strom. Výberom vždy najlepšieho prediktoru pre rozdelenie tak nemusíme dostať strom s najväčšou presnosťou[3].

Primárna premenná dosahuje z hľadiska kriteriálnej štatistiky najlepšie delenie daného uzlu a je použitá ako rozhodovacie pravidlo v strome. Môže sa stať, že premenná, ktorá je temer rovnako vhodná, tj. kriteriálna štatistika má podobnú hodnotu ako primárna premenná, zostane skrytá, aj keď môže mať väčšiu interpretačnú hodnotu. Takéto premenné sa nazývajú **zástupné** premenné (*surrogates*), či **kompetitivné** premenné.

Zástupné premenné nesú podobnú informáciu ako primárne a väčšinou sú s ňou korelované. Pre každý uzol je možné určiť, nakoľko rovnako rozdeľujú pozorovania v dcérskych uzloch v porovnaní s primárnymi premennými. Zástupné premenné majú veľký význam hlavne pri interpretácii.

Naproti tomu, kompetitívna premenná rozdeľuje daný uzol odlišne než primárna. Na základe hodnôt kriteriálnej štatistiky sa tak v prípade absencie primárnej premennej rozdelí uzol buď podľa kompetitívnej alebo zástupnej premennej. Vybraný je prediktor s ďalšou najlepšou hodnotou kriteriálnej štatistiky.

V tabuľke 2.2 podľa učebného textu [2] uvádza autorka ilustratívny príklad určovania typov premenných pri rozdelení pozorovaní troch kategórií A, B a C do dvoch dcérskych uzlov, pričom každá z kategórií obsahuje 100 pozorovaní. Vidíme, že rozdelenie počtu pozorovaní na základe primárnej a zástupnej premennej je takmer rovnaké. Čo sa týka kompetitívnej premennej, v prípade kategórie A rozdeľuje pozorovania rovnakým pomerom ako zástupná. V ďalších kategóriách sa rozdelenie výrazne líši.

2.2 Použitie rozhodovacích stromov pri regresii

V posledných dvadsiatich rokoch bola veľká pozornosť pri rozvoji neparametrických metód sústredená na problematiku regresnej analýzy. Medzi množstvom navrhnutých a vyšetrovaných odhadov hrajú najdôležitejšiu rolu modifikácie odhadov pomocou k_N najbližších susedov, resp. jadrové odhady. Obe z týchto nepa-

[3]poznamenajme ale, že to tak často býva

rametrických odhadov majú spoločný rys. Pri ich konštrukcii vychádzame z určitého predom zvoleného delenia priestoru vysvetľujúcich premenných, zatiaľ čo vplyv nameraných hodnôt vysvetľovanej premennej sa prejavuje až druhotne.[4]

Uvažujme náhodný vektor (Y,$\mathbf{X}$), kde Y je reálna náhodná veličina (vysvetľovaná premenná) a $\mathbf{X} = (X_1,...,X_M)$, $M \geqslant 1$, je náhodný vektor vysvetľujúcich premenných, ktoré môžu byť merateľné, ale aj kategoriálne náhodné veličiny. Ďalej predpokladáme, že $\forall i$, $i = 1,...,M$, existuje priestor $\mathcal{X}_i \subseteq \mathbb{R}_i$, ktorý pokrýva všetky možné hodnoty veličiny X_i. Tzn. že všetky možné hodnoty vektoru $\mathbf{X}$ padnú do niektorého priestoru $\mathcal{X} = \mathcal{X}_1 * ... * \mathcal{X}_M \subseteq \mathbb{R}_M$. Realizácie vektoru ($Y$,$\mathbf{X}$) budeme značiť ($y_i$, $\mathbf{x}_i$), kde $\mathbf{x}_i = (X_{i1},...,X_{iM})$, $i = 1,2,...,N$. Predpokladajme ďalej, že jednotlivé vysvetľujúce premenné meráme, resp. ich hodnoty zapisujeme, vždy v tom istom poradí. Cieľom je odhadnúť neznámu regresnú krivku $r(.) = \mathsf{E}\,(Y|\mathbf{X} = .)$ na základe pozorovaní (y_i, $\mathbf{x}_i$) , $i = 1,2,...,N$.

2.2.1 Konštrukcia odhadov pomocou k_N najbližších susedov

Najprv popíšeme konštrukciu odhadu $r_N^1(.)$ pomocou k_N najbližších susedov v bode $\mathbf{x} \in \mathcal{X}$.

Nech $\mathcal{K}_N(x) = \{i|\mathbf{x}_i$ je niektorý z k_N najbližších susedov medzi $\mathbf{x}_1,...,\mathbf{x}_N$ k bodu $\mathbf{x}\}$, kde k_N, $N = 1,2,...$ je postupnosť prirodzených čísel taká, že $k_N \to \infty$ a $\frac{k_N}{N} \to 0$, $N \to \infty$. Potom

$$r_N^1(\mathbf{x}) = \sum_{i=1}^{N} y_i w_i I(i \in \mathcal{K}_N(\mathbf{x})), \quad \mathbf{x} \in \mathcal{X}, \tag{2.9}$$

kde w_i, $i = 1,2,...,N$ sú váhy. Ide teda o vážený priemer z tých pozorovaní y_i, pre ktoré prislúchajúce $\mathbf{x}_i$ ležia „blízko“ bodu, v ktorom odhadujeme. Váhy spravidla volíme tak, aby sme preferovali tie pozorovania y_i, $i \in \mathcal{K}_N$, pre ktoré prislúchajúce $\mathbf{x}_i$ leží bližšie k bodu $\mathbf{x}$, v ktorom odhadujeme. Prípadne voľbou váh potlačíme vplyv odľahlých pozorovaní (outliers). Zo vzťahu (2.9) je zrejmé, že odhad počítame $\forall \mathbf{x} \in \mathcal{X}$ z pevného počtu k_N pozorovaní.

2.2.2 Jadrové odhady

Podobne aj u jadrového odhadu preferujeme tie pozorovania y_i (tj. dávame im väčšiu váhu), pre ktoré prislúchajúce $\mathbf{x}_i$ ležia bližšie k bodu, v ktorom odhadujeme. V tomto prípade sa však neobmedzujeme na pevný počet pozorovaní, z ktorých odhad počítame.

Typický jadrový odhad funkcie $r(\mathbf{x})$ môžeme zapísať v tvare

$$r_N^2(\mathbf{x}) = \frac{\sum_{i=1}^{N} y_i K(\frac{\mathbf{x}-\mathbf{x}_i}{h_N})}{\sum_{i=1}^{N} K(\frac{\mathbf{x}-\mathbf{x}_i}{h_N})}, \quad \mathbf{x} \in \mathcal{X}, \tag{2.10}$$

kde h_N, $N = 1,2,...$ je postupnosť nezáporných konštánt taká, že $h_N \searrow 0$ pre $N \to \infty$ a $K(.)$ je niektorá pravdepodobnostná hustota.

Ako je okamžite vidieť z oboch uvedených definícií, odhady typu (2.9)-(2.10) a ich modifikácie závisia podstatne na nameraných hodnotách vektoru vysvetľujúcich premenných $\mathbf{X}$. Ťažkosti s tým spojené sú zvlášť patrné pri viacdimenzionál-

nych problémoch, kde sú získané výsledky ťažko interpretovateľné a použiteľné najmä na „oblaku dát“.

2.2.3 Konštrukcia odhadov pomocou regresných stromov

V tomto odstavci sa sústredíme na konštrukciu odhadov regresných kriviek pomocou rekurzívneho delenia podmnožín priestoru $\mathcal{X}$. Tieto odhady možno graficky znázorniť pomocou binárnej stromovej štruktúry, ktorú nazývame **regresné stromy**. Systematicky ich môžeme zaradiť medzi neparametrické odhady, presnejšie povedané odhady po častiach konštantné. Jedná sa totiž o odhady rovné konštante na podmnožinách priestoru $\mathcal{X}$, pričom pripomínajú najjednoduchší neparametrický odhad, tzv. *regresogram*. Ich konštrukcia sa ale výrazne líši.[4]

Pri konštrukcii regresných stromov sa v prvej fáze rozdelí priestor $\mathcal{X}$ na L disjunktných podmnožín, pričom v druhej fáze je neznáma regresná krivka na podmnožine odhadnutá konštantou. Základný rozdiel od konštrukcie regresogramu spočíva v tom, že delenie jednotlivých podmnožín priestoru $\mathcal{X}$, začínajúce jeho delením samotným, je realizované rekurzívne tak, aby sa v každom kroku delenia od seba oddelili pozorovania s vysokými hodnotami y_i vysvetľovanej premennej od tých y_i s nízkymi hodnotami.

Výsledkom je rozklad množiny $\mathcal{X}$ na neprázdne podmnožiny t_1,...,t_L také, že

$$\bigcup_{i=1}^{L} t_i = \mathcal{X}, \quad t_i \cap t_j = \emptyset, \quad 1 \leqslant i \neq j \leqslant L. \tag{2.11}$$

Ako bolo spomenuté, na každej z podmnožín t_i je neznáma regresná krivka odhadnutá konštantou. Výsledný odhad môžeme zapísať v tvare

$$r_N^3(\mathbf{x}) = \sum_{i=1}^{L} c_i I(\mathbf{x} \in t_i), \quad \mathbf{x} \in \mathcal{X}, \tag{2.12}$$

kde c_i, $i = 1,...,L$ sú konštanty.

Konštrukcia hľadaných odhadov sa opiera podobne ako pri klasifikačných stromoch o zodpovedanie následujúcich základných otázok:

1. Voľba miery kvality odhadu.
2. Stanovenie množiny otázok, podľa ktorých sú delené podmnožiny priestoru $\mathcal{X}$.
3. Odhadnutie tvaru regresnej krivky na jednotlivých podmnožinách.
4. Nájdenie pravidla pre výber optimálneho delenia danej podmnožiny.
5. Určenie pravidla pre ukončenie delenia.

[4]Pripomeňme, že pri konštrukcii regresogramu postupujeme tak, že najprv rozložíme priestor $\mathcal{X}$ na L disjunktných obdĺžnikov (zvyčajne rovnakej veľkosti), a na každom z nich odhadneme neznámu regresnú krivku konštantou. Spravidla ako (vážený) priemer pozorovaní y_i, pre ktoré prislúchajúce $\mathbf{x}_i$ padli do daného obdĺžnika. Rozklad priestoru $\mathcal{X}$ vôbec nezávisí na nameraných hodnotách vysvetľovanej premennej Y.

2.2.3.1 Miery kvality odhadu

Existuje niekoľko kriteriálnych štatistík, ktoré sa používajú pri stanovení kvality odhadu. Ako bolo spomenuté, kriteriálna štatistika meria homogenitu v uzloch (node impurity). Za mieru kvality odhadu v regresných stromoch sa štandardne volí stredná kvadratická chyba MSE*(mean squared error)*:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \equiv \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{c}_i)^2,$$

kde $\hat{c}_i$ je odhad hodnôt regresnej krivky.

Ak chceme dosiahnuť väčšiu robustnosť procedúry, použijeme strednú absolútnu chybu MAE*(mean absolute error)*:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \equiv \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{c}_i|,$$

ktorá nepenalizuje veľké chyby tak výrazným neproporcionálnym spôsobom ako stredná kvadratická chyba MSE.

2.2.3.2 Stanovenie množiny otázok

Množina otázok, podľa ktorej sú delené podmnožiny priestoru $\mathcal{X}$, sa používa presne tak ako v prípade klasifikácie (viď 2.1.2).

2.2.3.3 Tvar odhadu

Odhadnutie tvaru neznámej regresnej krivky, tj. hodnoty konštánt c_i v (2.12), záleží podstatne na výbere miery kvality odhadu. Majme k dispozícii nejaký rozklad množiny v tvare (2.11). Potom pri voľbe strednej kvadratickej chyby MSE ako miery kvality odhadu ľahko ukážeme, že optimálne c_i budú v tvare

$$\hat{c}_i = \frac{1}{\text{card}\,\{\delta_{t_i}\}}\sum_{\delta_{t_i}} y_j, \quad i = 1,...,L, \tag{2.13}$$

kde $\delta_{t_i} = \{(y_j,\mathbf{x}_j)|\mathbf{x}_j \in t_i\}$.

Analogicky, ak zvolíme ako mieru rizika absolútnu štvorcovú chybu MAE, potom c_i budú v tvare

$$\hat{c}_i = \text{med}_{\delta_{t_i}}\, y_j, \quad i = 1,...,L. \tag{2.14}$$

2.2.3.4 Nájdenie pravidla pre výber optimálneho delenia

Nech $t \subseteq \mathcal{X}$. Pozrieme sa na to, ako pri zvolenej miere kvality odhadu a pri danej množine otázok Q nájdeme optimálny rozklad t splňujúci (2.4). Ako mieru zvoľme strednú štvorcovú chybu.

Ako sme už ukázali, ku každej otázke $q \in Q$ existuje nejaký rozklad splňujúci (2.4) a podľa (2.13) k nemu môžeme spočítať zodpovedajúce hodnoty konštánt c_L a c_R generujúce odhad. Pomocou nich ďalej môžeme odhadnúť strednú štvorcovú chybu zodpovedajúcu tomuto odhadu generovanému rozkladom množiny

t a otázke q. Jej hodnotu označíme $r(t,q)$. *Optimálnym delením* množiny t potom rozumieme delenie zodpovedajúce otázke $q^* \in Q$ poskytujúce najmenšiu strednú štvorcovú chybu medzi všetkými otázkami $q \in Q$, tj. pre ktoré

$$q^* = \arg\min_{q \in Q} r(t,q).$$

Ak je minimum dosiahnuté pre viacero delení, vyberieme za optimálne jedno z nich náhodne.

2.2.3.5 Určenie pravidla pre ukončenie delenia

Ak začneme deliť priestor $\mathcal{X}$, rekurzívnym opakovaním postupu vyššie dôjdeme až do štádia, kedy bude každá podmnožina pozostávať len z jedného pozorovania. Potom

$$\hat{c}_i = y_i, \quad i = 1,...,N,$$

čo je prakticky neprijateľný odhad, napriek tomu, že odpovedajúca chyba koncových uzlov $MSE = 0$. Problémom tu je, že pre takýto odhad nie je žiadna generalizácia. O pravidlách zastavenia rastu stromu sme sa zmienili v sekcii 2.1.5. Prirodzené riešenie v prípade regresných stromov by bolo zastaviť delenie v okamihu, kedy sa výrazne spomalil pokles zvolenej celkovej miery kvality odhadu. Toto pravidlo sa však neosvedčilo z dvoch následujúcich dôvodov:

- Neexistuje rozumné pravidlo, ktoré by umožnilo určiť dolnú hranicu poklesu miery kvality odhadu. Ak zvolíme prísnu hranicu, výsledky majú nízku vypovedateľnú hodnotu. Naopak aj je hranica stanovená moc voľne, výsledné odhady sú príliš rozsiahle a neprehľadné.

- Malý pokles miery kvality odhadu v jednom kroku nám ale nezaručuje to, že veľkosť poklesu v krokoch následujúcich bude takisto malá.

Analogicky, ako pri klasifikácii, sa pristúpilo k postupu skonštruovania rozsiahlejšieho klasifikátoru a jeho spätného kolapsovania. Pri kolapsovaní používame opäť tie isté miery pre kvalitu odhadov, ako pri konštrukcii, tj. stredné štvorcové (resp. absolútne) chyby, modifikované o člen penalizujúci nás za príliš rozsiahle rozklady. Z výslednej množiny vyberieme optimálne riešenie.

V prípade výberu optimálneho riešenia existujú dva prístupy, ktoré boli bližšie vysvetlené v predchádzajúcich odstavcoch. Prvým z nich je výber riešenia metódou **testovacieho súboru** v prípade, že máme k dispozícii dostatok pozorovaní. Pri tejto metóde rozdelíme pozorovaný súbor na dve skupiny. Pomocou jednej z nich budujeme odhad a druhú použijeme pre testovanie jeho kvality. V prípade, kedy nemáme dostatok pozorovaní je výhodnejšou metódou tzv. **krížové overovanie** (*cross-validation*), kde všetky dáta slúžia jednak ku konštrukcii odhadu a aj pre následné overenie jeho kvality. Tento postup vyžaduje viac pamäte počítača a takisto spotrebuje viac času, pretože v každom kroku je potrebné prejsť množstvo pomocných riešení.

Krížové overovanie v regresii

Nech $\mathcal{L} = \mathcal{L}_1 \cup ... \cup \mathcal{L}_K, \quad \mathcal{L}_i \cap \mathcal{L}_j = \emptyset, \quad i \neq j$, potom $\mathcal{L} \setminus \mathcal{L}_k$ použijeme ku konštrukcii regresnej krivky $r(\mathbf{x})$ a $\mathcal{L}_k$ ku kontrole kvality predikcie. Pre regresné

metódy definujeme odhad chyby pri krížovom overovaní následovne: Nech κ : $\{1,...,N\} \rightarrow \{1,...,K\}$ je indexová funkcia vyjadrujúca časť dátového súboru, do ktorej je i-té pozorovanie alokované pri randomizácii. Označme $r(\mathbf{x}, \mathcal{L}_k)$ odhad predikcie regresnej krivky skonštruovanej na množine $\mathcal{L} \setminus \mathcal{L}_k$ a aplikovanej na množinu $\mathcal{L}_k$.

Potom odhad predikčnej chyby krížového overovania vyjadríme vzťahom

$$CV(r) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, r(\mathbf{x}_i, \mathcal{L}_{\kappa(i)}),$$

kde $L(Y, r(\mathbf{X}))$ je stratová funkcia pre určenie chyby medzi Y a $r(\mathbf{X})$. Typické voľby stratovej funkcie sú:

$$L(Y, r(\mathbf{X})) = \begin{cases} (Y - r(\mathbf{X}))^2 \\ |Y - r(\mathbf{X})| \end{cases} \tag{2.15}$$

Majme množinu modelov $\{r_\alpha(\mathbf{x})\}$, kde $r_\alpha(\mathbf{x}, \mathcal{L}_k)$ označuje odhad predikcie regresnej krivky skonštruovanej na množine $\mathcal{L} \setminus \mathcal{L}_k$ a aplikovanej na množinu $\mathcal{L}_k$. Potom pre túto množinu modelov definujeme

$$CV(r,\alpha) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, r_\alpha(\mathbf{x}, \mathcal{L}_{\kappa(i)})).$$

Funkcia $CV(r,\alpha)$ vyjadruje odhad krivky predikčnej chyby a my hľadáme parameter $\hat{\alpha}$, ktorý ho minimalizuje. Náš finálny vybraný model bude $r_{\hat{\alpha}}(\mathbf{x})$, ktorým odhadneme všetky dáta.[5]

2.2.3.6 Určenie presnosti regresného stromu

Určovanie presnosti stromu sme si už bližšie popísali pre klasifikačné stromy. V prípade regresných stromov je ich kvalita určovaná pomocou koeficientu determinácie, rovnako ako v prípade lineárnej regresie.

Všeobecne v lineárnej regresii odhadujeme koeficienty regresnej krivky pomocou metódy najmenších štvorcov (*Ordinary Least Squares OLS*). Táto metodika je založená na minimalizácii **reziduálneho súčtu štvorcov** (*residual sum of squares RSS*)

$$RSS = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2. \tag{2.16}$$

Čím menšia je jeho hodnota, tým prijateľnejší by mal byť skonštruovaný model.

Definujme aj ďalšie používané typy súčtu štvorcov, ktorými sú **úplný súčet štvorcov** (*total sum of squares TTS*)

$$TSS = \sum_{i=1}^{N} (y_i - \bar{y})^2 \tag{2.17}$$

a **vysvetlený súčet štvorcov** (*explained sum of squares ESS*)

$$ESS = \sum_{i=1}^{N} (\hat{y}_i - \bar{y})^2. \tag{2.18}$$

Potom **koeficient determinácie** (*coefficient of determination*) je definovaný ako

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}. \tag{2.19}$$

Teda všeobecne je koeficient definovaný ako podiel variability závislej premennej Y, vysvetlenej modelom, k celkovej variabilite premennej Y.

V našom prípade ide o variabilitu vysvetlenú stromom, čo je

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

kde $\hat{y}_i = \bar{y}_t$ je priemer v príslušných koncových uzloch a odchýlka od priemeru v uzle t je spočítaná vždy od pozorovaní y_i zaradených do tohto koncového uzlu. V menovateli je počítaný súčet štvorcových odchýliek všetkých hodnôt pozorovaní y_i od priemeru $\bar{y}$ v koreni, ktorý obsahuje všetky pozorovania premennej Y.

Priamo zo vzťahu (2.19) plynie, že $0 \leq R^2 \leq 1$. Ak je hodnota R^2 vysoká, teda blízka jednej, potom skonštruovaný model padne dobre daným dátam. V prípade $R^2 = 1$ sme vysvetlili akúkoľvek variabilitu pomocou stromu a predikované hodnoty $\hat{y}_i$ sa zhodujú s pozorovanými hodnotami y_i. Naopak, ak je hodnota R^2 blízko nuly, model sa pre dané dáta príliš nehodí. Maximalizácia koeficientu R^2 teda zrejme zodpovedá kritériu najmenších štvorcov.

Opäť ako v prípade klasifikačného stromu môžeme určiť chybu trénovacieho súboru $e(t) = 1 - R^2_{tren}$ a chybu testovacieho súboru $e'(t) = 1 - R^2_{test}$.

2.3 Výhody a nevýhody CART

CART analýza má množstvo výhod oproti iným klasifikačných metódam, vrátane multinomickej logistickej regresie. Za prvé, je neodmysliteľne neparametrická. Inými slovami, nie sú žiadne predpoklady a požiadavky, pokiaľ ide o základné rozdelenie hodnôt vysvetľujúcich premenných. Kvôli tomu, je CART schopná spracovať číselné údaje, ktoré sú vysoko zošikmené alebo multimodálne, rovnako ako kategoriálne prediktory s buď ordinálnou alebo neordinálnou štruktúrou. To je dôležitá vlastnosť, ktorá skracuje analytikovi čas, ktorý by inak musel byť využitý pri určení, či sú premenné normálne rozdelené, a robiť transformáciu, ak nie sú. Nevýhodou stromov je však ich pomerne vysoká nestabilita, čo môžeme odstrániť napríklad použitím lesov, ktoré neskôr popíšeme.

CART identifikuje „deliace“ premenné na základe vyčerpávajúceho hľadania všetkých možností. Pretože sú využívané efektívne algoritmy, CART je schopná vyhľadávať všetky možné premenné ako deliace, dokonca aj pri problémoch so stovkami možných prediktorov.

Výhody

- Jednoduché grafické znázornenie v podobe stromovej štruktúry, z čoho plynie relatívna jednoduchosť pri interpretovaní.

- Nekladie žiadne podmienky na typ rozdelenia závislej premennej ani prediktorov.

- Závislá premenná aj prediktory môžu byť všetkých typov (kategoriálne, ordinálne aj spojité).
- Algoritmus tvorby stromu je odolný voči odľahlým hodnotám, ktoré možno včas odhaliť pri krížovej validácii.
- Je možné použiť korelované prediktory, pretože strom rastie hierarchicky a pre delenie sa vyberá vždy len jeden prediktor (zo všetkých možných korelovaných).
- Výsledky presnosti stromu môžeme ľahko porovnať s výsledkami u iných modelov. Koeficient determinácie R^2 u regresného stromu je zrovnateľný s R^2 u ostatných regresných techník a percento správne klasifikovaných pozorovaní s výstupmi iných klasifikačných metód.
- Ide o veľmi rýchlu metóda pri klasifikácii nových prípadov.
- Metóda vhodná pre klasifikáciu aj regresiu (pre regresiu s istými obmedzeniami).
- CART stromy sú vhodné aj pre veľký počet premenných a to všetkých typov.

Popisovaná metodológia je schopná okrem hlavného cieľa, tj. umožnenia klasifikácie (resp. predikcie v regresnom prípade), zvládnuť oveľa viac. Medzi inými ponúka možnosti:

- vyrovnať sa s chýbajúcimi pozorovaniami
- stanoviť dôležitosť sledovaných premenných;
- nájsť najlepšie náhradné otázky v každom rozhodovacom kroku;
- podstatne znížiť dimenzionalitu problému pre potreby klasifikácie (resp. predikcie);
- zvládnuť klasifikáciu aj v prípadoch, kedy sa dimenzionalita sledovaných objektov mení prípad od prípadu.

Nevýhody

Navzdory mnohým výhodám, existujú samozrejme aj nevýhody CART, ktoré by sme mali mať na pamäti.

- Nestabilita - tvar stromu veľmi závisí od dát, malá zmena spôsobí zmeny v rozhodovacích pravidlách vrámci uzlov, čo môže viesť k zmene výsledných klasifikácií/predikcií.
- Vzhľadom k nestabilite je nutná opatrnosť pri interpretácii stromu.
- Meranie presnosti stromu je výrazne závislé na krosvalidačnom mechanizme a ďalších parametroch pri validácii modelu vo fáze učenia (napr. pravidla pre zastavenie rastu stromu).

- Stromy sú nevhodné pre malý počet vzoriek a veľký počet kategórií závislej premennej.
- Vytváranie stromov vyžaduje skúsenosti s nastavením parametrov v procese validácie, ktoré je do značnej miery subjektívné.

2.4 Chyba predikcie a klasifikačná chyba modelu

Profesor Breiman v článku [6] uvádza, že rozdiel medzi klasifikačnou chybou na testovacom súbore a minimálnou dosiahnuteľnou chybou je súčet **vychýlenia** (*bias*) a **rozptylu** (*variance*). Nestabilné klasifikátory, ako napríklad stromy, sú charakteristické vysokou variabilitou a nízkym vychýlením. Pozrime sa na to, ako sú vychýlenie a rozptyl definované pre regresné a klasifikačné metódy.

2.4.1 Vychýlenie a rozptyl v regresii

Pojmy vychýlenie (*bias*) a rozptyl (*variance*) pochádzajú z dekompozície predikčnej chyby.

Uvažujme trénovací súbor $\mathcal{L} = \{(y_i,\mathbf{x}_i), i = 1,...,N\}$, kde y_i je numerická závislá premenná (výstup) a $\mathbf{x}_i$ sú mnohorozmerné vektory vysvetľujúcich premenných (vstup). Metódy ako neuronové siete, regresné stromy, lineárna regresia, atď. sú aplikované na trénovací súbor $\mathcal{L}$ a konštruujú prediktor $d(\mathbf{x},\mathcal{L})$ budúcich hodnôt y.

Uvažujme model $y = r(\mathbf{x}) + \varepsilon$, kde $\varepsilon \sim N(0,\sigma^2)$. Ďalej predpokladajme, že aplikujeme zostrojený prediktor d na testovací subor $\mathcal{T} = (y^*,\mathbf{x}^*)$, kde y^* je pozorovaná hodnota. Definujeme očakávanú chybu predikcie vzťahom

$$PE(d(\mathbf{x}^*)) = \mathsf{E}_{\mathbf{X},Y}(y^* - d(\mathbf{x}^*))^2,$$

kde

$$\begin{aligned}\mathsf{E}\left[(y^* - d(\mathbf{x}^*))^2\right] &= \mathrm{var}(d(\mathbf{x}^*)) + \mathrm{bias}^2(d(\mathbf{x}^*)) + \mathsf{E}\left[\varepsilon^2\right] \\ &= \mathrm{var}(d(\mathbf{x}^*)) + \mathrm{bias}^2(d(\mathbf{x}^*)) + \sigma^2\end{aligned}$$

Celkovú očakávanú chybu modelu môžeme teda rozložiť na tri zložky, podľa [7] a [8], vzťahom

$$PE(d) = E\varepsilon^2 + Bias^2(d) + Var(d), \tag{2.20}$$

kde $E\varepsilon^2$ je **šum** (*noise*), tj. reziduálna chyba alebo minimálna chyba, ktorú nie sme schopní modelom vysvetliť.

Vychýlenie určuje systematickú chybu modelu. Je to rozdiel optimálneho modelu od priemerného modelu. **Rozptyl** je variabilita výsledkov jednotlivých výberov. Vysoký rozptyl značí preučený model.

V skutočnosti máme ale k dispozícii len jeden trénovací súbor. Odhady zložiek môžeme dosiahnuť aplikáciou bootstrapových výberov, tj. zostrojíme B boostrapových súborov z $\mathcal{T}$. Na každý aplikujeme CART, vypočítame priemernú hodnotu $\underline{d}$. Pre každé $\mathbf{x}$ teda máme pozorovanú hodnotu y a jej predikcie $y_1,...,y_B$. Potom odhad vychýlenia je

$$\underline{d} - y, \tag{2.21}$$

a odhad rozptylu

$$\frac{1}{B-1}\sum_b (y_b - \underline{d})^2. \tag{2.22}$$

Šum predpokladáme nulový.

2.4.2 Dekompozícia klasifikačnej chyby v klasifikácii

V klasifikácii je závislá premenná y kategoriálneho typu, s kategóriami $\{1,...,J\}$. Trénovací súbor uvažujeme v tvare $\mathcal{L} = \{(y_i,\mathbf{x}_i), i = 1,...,N\}$, kde y_i je kategoriálna premenná. Klasifikačné metódy aplikované na súbor $\mathcal{L}$ konštruujú klasifikátor $C(\mathbf{x},\mathcal{L})$, umožňujúci predikciu budúcich hodnôt y. Opäť predpokladajme, že dáta v trénovacom súbore sú nezávislé a rovnako rozdelené vzorky z rozdelenia vektoru $(Y,\mathbf{X})$. Dekompozíciu klasifikačnej chyby definovalo mnoho autorov. V rámci tejto práce popíšeme definície vychýlenia a rozptylu pre klasifikačné metódy podľa Breimana [7], kde chyba zlej klasifikácie (*missclassification error*) je definovaná

$$PE(C(\mathbf{X},\mathcal{L})) = E_{\mathbf{X},Y}(C(\mathbf{x},\mathcal{L}) \neq Y)$$

Označme:

$$P(j|\mathbf{x}) = P(Y = j|\mathbf{X} = \mathbf{x})$$
$$P(\mathbf{dx}) = P(\mathbf{X} \in \mathbf{dx})$$

Minimálna miera zlej klasifikácie je určená „Bayesovským klasifikátorom $C^\star$“ definovaným vzťahom

$$C^\star(\mathbf{x}) = \arg\max_j P(j|\mathbf{x})$$

s mierou zlej klasifikácie

$$PE(C^\star) = 1 - \int \max_j(P(j|\mathbf{x}))P(\mathbf{dx})$$

Ďalej nech

$$Q(j|\mathbf{x}) = P_{\mathcal{L}}(C(\mathbf{x},\mathcal{L}) = j),$$

a definujeme agregovaný klasifikátor predpisom

$$C_A(\mathbf{x}) = \arg\max_j Q(j|\mathbf{x}).$$

Ide o agregáciu hlasovaním, kde uvažujeme, že máme nezávislé trénovacie súbory $\mathcal{L}_1,\mathcal{L}_2,\ldots$, pomocou ktorých zostrojíme klasifikátory $C(\mathbf{x},\mathcal{L}_1),C(\mathbf{x},\mathcal{L}_2),\ldots$ a pre každé $\mathbf{x}$ určíme $C_A(\mathbf{x})$ pomocou hlasov klasifikátorov pre najviac zastúpenú triedu.

Autor ďalej definuje množinu U (*unbiased set*) všetkých $\mathbf{x}$ takých, pre ktoré je klasifikátor C nevychýlený. Komplementom k tejto množine je množia B *biased set*. Ak je klasifikátor C nevychýlený pre $\mathbf{x}$, potom $C_A(\mathbf{x})$ je optimálny klasifikátor.

Vychýlenie klasifikátoru C je definované vzťahom

$$bias(C) = \mathsf{P}_{\mathbf{X},Y}(C^\star(\mathbf{X}) = Y, \mathbf{X} \in B) - \mathsf{E}_{\mathcal{L}}\,\mathsf{P}_{\mathbf{X},Y}(C(\mathbf{X},\mathcal{L}) = Y, \mathbf{X} \in B)$$

a jeho rozptyl

$$\mathrm{var}(C) = \mathsf{P}_{\mathbf{X},Y}(C^\star(\mathbf{X}) = Y, \mathbf{X} \in U\ ) - \mathsf{E}_{\mathcal{L}}\,\mathsf{P}_{\mathbf{X},Y}(C(\mathbf{X},\mathcal{L}) = Y, \mathbf{X} \in U).$$

Celková chyba klasifikátoru C je opäť vyjadrená ako dekompozícia do troch zložiek:

$$PE(C) = PE(C^\star) + Bias(C) + Var(C), \tag{2.23}$$

Poznamenajme, že agregovanie klasifikátora a nahradenie C spoločným klasifikátorom C_A redukuje rozptyl na nulu, ale nedáva žiadnu garanciu redukcie vychýlenia. Z definície, vychýlenie a rozptyl majú následujúce vlastnosti:

- Vychýlenie a rozptyl sú vždy nezáporné
- Rozptyl C_A je nulový
- Ak je C deterministický, tj. nezávisí na $\mathcal{L}$, potom je jeho rozptyl nulový
- Vychýlenie klasifikátoru $C^\star$ je nulové.

2.5 Slabé modely

Slabý model (*weak learner*) je definovaný ako model, ktorého vychýlenie je malé, ale vyznačuje sa vysokým rozptylom. Slabé modely majú teda vysokú presnosť, ale len na trénovacích súboroch. Sú využívané pri konštrukcii skupinových modelov, o ktorých si bližšie povieme v následujúcej kapitole. Dôležité však je, že v prípade skupinového modelu hľadáme taký model, aby mal nízku varianciu aj vychýlenie a ukázalo sa, že kombinovaním niekoľkých slabých modelov môžeme dosiahnuť zníženie oboch zložiek.

Stromy CART sú teda dobrým kandidátom pre použitie v skupinových modeloch. Neprerezané stromy majú totiž vysokú presnosť pre trénovací súbor (teda nízky bias), ale vysokú variabilitu (tj. výsledky medzi testovacím a trénovacím súborom sa líšia). Rozhodovacie stromy, na ktoré nie sú aplikované metódy pre hľadanie optimálnych stromov[5], sú podľa vyššie uvedenej definície slabými modelmi.

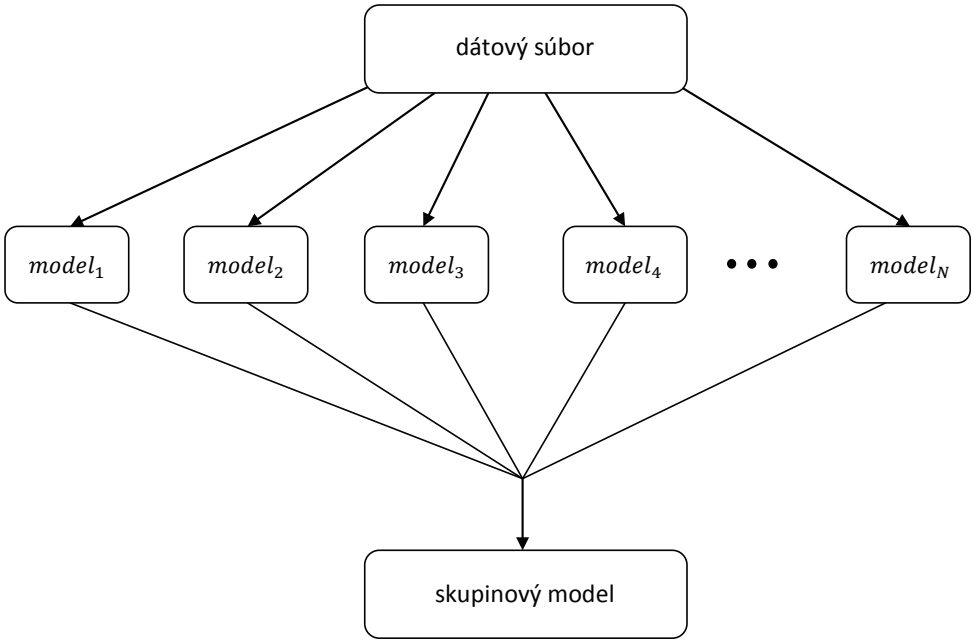
[5]Pripomeňme, že pri rozhodovacích stromoch sme pre určenie jeho optimálnej veľkosti museli takisto nájsť kompromis medzi rozptylom a skreslením.

Kapitola 3

Lesy

Doteraz sme sa venovali metodike, ktorá vytvorila pre daný dátový súbor jeden model - strom. S myšlienkou kombinácie viacerých stromov prišiel profesor Breiman v práci [9] a vytvoril tak systém nazvaný **náhodný les** (*random forest*). Skôr ako sa dostaneme k jednotlivým metódam konštrukcie lesov, pozrieme sa v krátkosti na problém všeobecne. Lesy môžeme zaradiť do skupiny tzv. **skupinových metód** (*ensemble methods*).

Princíp skupinových modelov je jednoduchý. Je očakávané, že v tomto prípade je skupine modelov (napr. rozhodovacích stromov) zadaný rovnaký problém, na ktorom sa učia. Následne sa výstupy naučených modelov kombinujú do jedného skupinového modelu, ako je to názorne ukázané na obrázku 3.1. Zaujíma nás, či kombináciou výsledkov jednotlivých modelov dosiahneme zlepšenie výsledkov klasifikácie alebo predikcie.



Obr. 3.1: Schéma tvorby skupinového modelu

V tomto momente sa vynára otázka, či môžeme vôbec kombináciou modelov získať presnejší model? Predpokladom pre použitie kombinácie modelov je podmienka, že jednotlivé modely budú rôzne. To dosiahneme napríklad tým, že

jednotlivé modely zostrojíme na vždy inom súbore, ktorý vznikne náhodným výberom z pôvodnej trénovacej skupiny dát. Ak by sme zostrojili modely na tom istom súbore, skupinový model by nepriniesol žiadne zlepšenie, pretože výsledok by bol zhodný s výsledkami jednotlivých modelov. Modely, ktoré takto zostrojíme budú vykazovať odlišné chyby (môžu a nemusia). Ich presnosť a stabilita sa overí pomocou testovacích súborov.

Niektoré klasifikačné a regresné metódy sú nestabilné v tom zmysle, že malé odchýlky v trénovacích súboroch alebo v konštrukcii môžu mať za následok veľké zmeny v konečnej podobe prediktoru. Napríklad rozhodovacie stromy v regresii a klasifikácii a neurónové siete sú nestabilné. Podľa Breimana v článku [6] sa môže pri nestabilných metódach ich presnosť zlepšiť kombinovaním. To je možné dosiahnuť vytváraním viacerých verzií prediktoru buď zmenami trénovacieho súboru alebo zmenami v konštrukcii metódy a následnou kombináciou týchto verzií do jedného výsledného prediktoru.

Klasifikačné a regresné stromy sa pestujú od 60. rokov. Silným metodologickým impulzom bola v 80. rokoch vtedy nová metóda `CART`, ktorú sme popísali v predchádzajúcich kapitolách. Medzitým, čo od druhej polovice 90. rokov pribúdali dalšie práce o stromoch, vo svete, najviac asi v pracovni Lea Breimana, jedného z „otcov“ `CART`-u, odštartovala nová etapa rozvoja metód analýzy dát založených na stromoch.

Klasifikačný les je klasifikačný model, ktorého klasifikačná funkcia vznikne kombináciou (podľa vhodne zvoleného pravidla) klasifikačných funkcií typicky niekoľkých desiatok klasifikačných stromov. Každý klasifikačný strom T určuje klasifikačnú fuknkciu d_T definovanú na $\mathcal{X}$, priestore hodnôt prediktorov, s hodnotami v množine tried $\{C_1,...,C_k\}$.

Podobne môžeme charakterizovať ***regresný les***, ktorý pozostáva z niekoľkých regresných stromov. Regresný strom T, narozdiel od klasifikačného stromu, priraďuje terminálnemu uzlu reálnu konštantu - odhad kvantitatívnej závislej premennej Y. Tj. regresný strom definuje regresnú funkciu d_T, ktorá je vnútri množín, prislúchajúcich terminálnemu uzlu, konštantná.

Vypestujme teda L stromov $T_1,...,T_L$. Výsledná „agregovaná“ klasifikačná (resp. regresná) funkcia $d_A(\mathbf{x})$, kde $\mathbf{x}$ je vektorom hodnôt prediktorov, vznikne vhodnou kombináciou klasifikačných (resp. regresných) funkcií $d_1(\mathbf{x}),...,d_L(\mathbf{x})$ jednotlivých stromov. V regresnom prípade je prirodzeným a jednoduchým spôsobom kombinácie aritmetický priemer

$$d_A(\mathbf{x}) = \frac{1}{L}\sum_{i=1}^{L} d_i(\mathbf{x}) \tag{3.1}$$

V prípade klasifikácie sa uplatňuje väčšinové hlasovanie, tj.

$$d_A(\mathbf{x}) = C_{i^\star}, \quad \text{ak} \quad \#\{j; d_j(\mathbf{x}) = C_{i^\star}\} = \max_{i=1,...,k} \#\{j; d_j(\mathbf{x}) = C_i\}, \tag{3.2}$$

kde symbol # označuje počet prvkov. Ekvivalentne môžeme zapísať

$$d_A(\mathbf{x}) = \arg\max_{i=1,...,k} \sum_{j=1}^{L} I(d_j(\mathbf{x}) = C_i), \tag{3.3}$$

kde $I(.)$ je funkcia identifikátoru. Ak dôjde k nejednoznačnosti zo zhody hlasov, rieši sa to napríklad znáhodnením.

Kombinovanie klasifikačných funkcií môže byť ešte o niečo zložitejšie. Pri hlasovaní môže váha hlasu každej z L klasifikačných funkcií závisieť na chybe stromu na trénovacích dátach, tj. presnejší strom má prirodzene väčšiu váhu. Prípadne váha hlasu nemusí byť rovnaká pre všetky hodnoty vektoru prediktorov $\mathbf{x}$. Napríklad môže závisieť od veľkosti listu, do ktorého $\mathbf{x}$ patrí alebo od toho, aký je list „čistý“, tj. ako výrazná je prevaha najfrekventovanejšej triedy. Zložitejšie váženie hlasov je už súčasťou niektorých metód konštrukcie lesov, ale zároveň je stále ešte predmetom výzkumu.[10]

V rámci myšlienky skupinových modelov sme naznačili postup, ako dosiahnuť, aby jednotlivé modely boli rôzne. Existuje viacero riešení tohoto problému. Niektoré z nich si teraz predstavíme.

3.1 Bagging

Bagging je skratkou, akronymom, pre „***b**ootsrap **agg**regat**ing**“ popísaný v článku [11].

Uvažujme trénovací súbor $\mathcal{L}$, pozostavujúci z dát $\{(y_i,\mathbf{x}_i), i = 1,...,N\}$, kde y je buď kategoriálna alebo numerická premenná. Predpokladajme, že máme procedúru, ktorá využitím tohoto trénovacieho súboru vytvorí prediktor $d(\mathbf{x},\mathcal{L})$. Teraz predpokladajme, že máme postupnosť trénovacích súborov $\{\mathcal{L}_k\}$, každý pozostáva z N nezávislých pozorovaní s rovnakým rozdelením ako súbor $\mathcal{L}$. Našou úlohou bude použiť $\{\mathcal{L}_k\}$ k tomu, aby sme dostali lepší prediktor ako $d(\mathbf{x},\mathcal{L})$. Budeme teda pracovať s postupnosťou prediktorov $d(\mathbf{x},\mathcal{L}_k)$.

Vo všeobecnosti máme ale k dispozícii jeden trénovací súbor $\mathcal{L}$. Ďalšie súbory vytvoríme pomocou metódy *bootstrap* tak, že náhodným výberom *s vracaním* vytvoríme z pôvodného dátového súboru ďalších B súborov $\mathcal{L}_1,...,\mathcal{L}_B$ o veľkosti N. Takto vytvorené súbory sú použité na vytvorenie jednotlivých klasifikačných (resp. regresných) stromov a výsledný klasifikačný (či regresný) les je daný väčšinovým hlasovaním s rovnakými váhami (resp. aritmetickým priemerom jednotlivých regresných funkcií).

Algoritmus 3.1 Bagging

Vstup: trénovací súbor $\mathcal{L} = \{(y_i,\mathbf{x}_i), i = 1,...,N\}$
algoritmus vytvárajúci slabý model (ozn. **WeakLearn**)
integer B {*udávajúci počet iterácií*}

1: **Inicializujeme**: $b = 1$
2: **repeat**
3: Vytvoríme bootstrapový súbor $\mathcal{L}_b$ {*tvoríme trénovací súbor*}
4: Pomocou **WeakLearn** vytvoríme strom T_b zo súboru $\mathcal{L}_b$ {*fáza učenia klasifikátoru* $d_b(\mathbf{x})$}
5: $b = b + 1$
6: **until** $b = B$

Výstup:

$$d_A(\mathbf{x}) = \arg\max_{y\in Y} \sum_{b=1}^{B} I(y = d_b(\mathbf{x}))$$

{$I(.)$ *je funkcia identifikátoru* }

Pri bootstrapovom výbere sú niektoré pozorovania z $\mathcal{L}$ vybrané opakovane, iné naopak vôbec. Počet opakovaní pre jednotlivé pozorovania z $\mathcal{L}$ má asymptoticky (pre $N \longrightarrow \infty$) Poissonovo rozdelenie so strednou hodnotou 1. Teda pravdepodobnosť, že pozorovanie nebude vybrané je približne $e^{-1} \approx 0.37$. Bootstrapový výber tvorí približne 63% pozorovaní z $\mathcal{L}$, 37% pozorovaní sa nedostane do výberu. Tieto pozorovania nazývame **oob** vzorky (angl. *out of bag samples*).

Autor v [11] uvádza, že pomocou takto vytvorených trénovacích súborov pre $B = 50$ boli následne vypestované stromy metódou CART. Skutočná (generalizačná) klasifikačná chyba sa v niekoľkých reálnych i umelých klasifikačných metódach znížila o $20\% - 47\%$. Podobné výsledky dosiahol v regresných úlohách ($22\% - 46\%$).

Zároveň autor na niekoľkých príkladoch v článku [7] demonštroval efekt baggingu na vychýlenie a rozptyl klasifikačnej (resp. predikčnej) chyby. Kombináciou stromov dospel k redukcii rozptylu výsledného modelu.

Ak sa vrátime k teoretických odhadom vychýlenia a rozptylu, ktoré sme dostali pomocou bootstrapových výberov, aplikáciou baggingu, ktorý pozorovania predikuje vzťahom

$$y = \frac{1}{B} \sum_b d_b(\mathbf{x}) = \underline{d}(\mathbf{x})$$

dostávame pre odhad vychýlenia rovnaký vzťah (2.21) a pre odhad rozptylu dosadením do vzťahu (2.22) dostávame

$$\frac{1}{B-1} \sum_b (\underline{d} - \underline{d})^2 = 0.$$

Teda použitím baggingu teoreticky odstránime rozptyl a vychýlenie ostáva nezmenené, v praxi sa však ukázalo, že pri použití baggingu dosiahneme redukciu rozptylu a dokonca mierny nárast vychýlenia.

3.2 Boosting

Boosting (to boost v preklade zosíliť) je pôvodne termín používaný v strojovom učení. Pri tejto metóde sa postupne vytvárajú modely so stále vyššou váhou hlasu. Každý nový model je ovplyvnený modelmi vytvorenými v predchádajúcich krokoch. V rámci analýzy dát sa takto obvykle označuje algoritmus `AdaBoost` (***ada**ptive **boost**ing*), ktorý navrhli Freund a Shapire v článku [12].

Majme klasifikačnú metódu (nemusí sa v tomto prípade jednať len o stromy), ktorá na základe trénovacích dát $\mathcal{L} = \{(\mathbf{x}_1,y_1),...,(\mathbf{x}_N,y_N)\}$ a vektoru váh $w = (w_1,...,w_N)$, priradených jednotlivým pozorovaniam, vytvára klasifikačný model T. Algoritmus `AdaBoost` konštruuje postupnosť rozdielnych modelov $T_1,...,T_L$ s klasifikačnými funkciami $d_1(\mathbf{x}),...,d_L(\mathbf{x})$ tak, že sa na základe predchádzajúcich výsledkov upravujú váhy prípadov. V prvom kroku je použitý vektor w_1 zadaný užívateľom (volia sa väčšinou rovnomerné váhy) a vytvorí sa model T_1. V ďalších krokoch sa k vytvoreniu modelu T_i, $i = 2,...,L$ použije vektor w_i, ktorý získame úpravou vektoru w_{i-1}. Úprava prebieha tak, že váhy pozorovaní chybne klasifikovaných modelom T_{i-1} sa zvýšia a naopak správne klasifikovaných pozorovaní sa znížia. To znamená, že klasifikačná metóda sa stále viac „sústredí“

Algoritmus 3.2 AdaBoost

Vstup: trénovací súbor $\mathcal{L} = (\mathbf{x}_1,y_1),...,(\mathbf{x}_N,y_N)$
integer T {*udávajúci počet iterácií*}
algoritmus vytvárajúci slabý model (ozn. **WeakLearn**)
rozdelenie D pre N prvkov

1: **Inicializujeme** váhový vektor: $w_i = D(i)$ pre $i = 1,...,N$ {*na začiatku sa volia rovnomerné váhy* $D(i) = \frac{1}{N}$}
2: **for** $i = 1$ **to** T **do**
3: normalizujeme váhy

$$\mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{i=1}^{N} w_i^t}$$

4: aplikujeme **WeakLearn** metódu s rozdelením $\mathbf{p}^t$, ktorá vráti klasifikačnú funkciu $d_t(\mathbf{x},\mathcal{L}) : \mathbf{X} \to [0,1]$
5: spočítame chybu d_t:

$$\epsilon_t = \sum_{i=1}^{N} p_i^t |d_t(x_i) - y_i|$$

6: nastavíme

$$\beta_t = \frac{\epsilon_t}{(1 - \epsilon_t)}$$

7: aktualizujeme nový vektor váh

$$w_i^{t+1} = w_i^t \beta_t^{1-|d_t(x_i)-y_i|}$$

8: **end for**

Výstup: výsledná klasifikačná funkcia

$$d_f(x) = \begin{cases} 1 & \text{ak} \quad \sum_{t=1}^{T}(\log \frac{1}{\beta_t}) d_t(x) \geq \frac{1}{2} \sum_{t=1}^{T}(\log \frac{1}{\beta_t}) \\ 0 & \text{inak} \end{cases}$$

na „obtiažnejšie“ pozorovania, ktoré sa ťažko zaraďujú do správnej triedy. Váha modelu závisí na chybe modelu na trénovacích dátach.

Autori článku [12] popísali okrem základného algoritmu `AdaBoost`, určeného len pre klasifikačnú úlohu s dvoma triedami (tj. $Y = \{0,1\}$), aj modifikácie algoritmu. Algoritmy `AdaBoost.M1` a `AdaBoost.M2` sú určené pre úlohy s viacero triedami a `AdaBoost.R` pre regresné úlohy s hodnotami závislej premennej v intervale [0,1].

3.3 Arcing

Rovnako ako v predchádzajúcich metódach, *Arcing* je akronymom (***a**daptive **r**esampling and **c**ombin**ing***) pre ďalší postup konštrukcie lesov podľa autora Lea Breimana v článku [7].

Arcing predstavuje spojenie myšlienok oboch predchádzajúcich metód. Váhy prípadov sa postupne upravujú, rovnako ako v spomínanom algoritme `AdaBoost`, ale ich použitie je iné. Namiesto toho, aby všetky pozorovania s týmito získanými váhami vstupovali do analýzy, sú ako v baggingu vytvárané výbery s vracaním, pričom váhy (náležito normované) slúžia ako pravdepodobnosti „vytiahnutia" $\{p(n)\}$ pri bootstrape.

Algoritmus 3.3 Arc-fs

Vstup: trénovací súbor $\mathcal{L} = \{(y_i,\mathbf{x}_i), i = 1,...,N\}$
integer K $\{$*udávajúci počet iterácií*$\}$

1: **Inicializujeme** pravdepodobnosti: $p(i) = \frac{1}{N}$ $\{p(i)$ *predstavuje pravdepodobnosť i-tého prípadu v* $\mathcal{L}\}$
2: **for** $k = 1$ **to** K **do**
3: pravdepodobnosti $\{p^{(k)}(i)\}$ použijeme pri bootstrapovom výbere z $\mathcal{L}$ na zkonštruovanie trénovaceho súboru $\mathcal{L}^k$
4: zkonštuujeme klasifikátor C_k použitím $\mathcal{L}^k$
5: aplikuj C_k na pôvodnú množinu $\mathcal{L}$ a spočítaj $\delta(i)$, kde

$$\delta(i) = \begin{cases} 1 & \text{ak } i \text{ -tý prípad je klasifikovaný nesprávne;} \\ 0 & \text{inak} \end{cases}$$

6: definujeme $\varepsilon_k = \sum_i p^{(k)}(i)\delta(i), \qquad \beta_k = \frac{(1-\varepsilon_k)}{\varepsilon_k}$
7: aktualizujeme nové pravdepodobnosti pre $(k+1)$-vý krok

$$p^{(k+1)}(i) = \frac{p^{(k)}(i)\beta_k^{\delta(i)}}{\sum p^{(k)}(i)\beta_k^{\delta(i)}}$$

8: **end for**

Výstup: po K krokoch sú $C_1,...,C_K$ kombinované pomocou váženého hlasovania tak, že C_K majú váhy $\log(\beta_k)$

3.4 Náhodné lesy (Random forest)

Doteraz sme popisovali metódy, ktoré sú použiteľné nielen na stromy, ale aj na ľubovoľnú klasifikačnú, či regresnú metódu. Metóda `Random Forest` (v preklade *náhodné lesy*) sa ale týka špecificky stromov a lesov. Jej autorom je už spomínaný profesor Leo Breiman, ktorý v článku [9] vytvoril kombináciou stromov les v snahe o zlepšenie klasifikácie, prípadne predikcie. Náhodné lesy môžu byť použité pre klasifikáciu aj regresiu a čo je dôležité, odstraňujú niektoré problémy, ktoré nastávajú pri použití stromov. Predovšetkým ide o nestabilitu stromov, ktorú sme spomínali v predchádzajúcej kapitole. Autor ponúka následujúcu definíciu náhodného lesa.

Definícia 1. Náhodný les je tvorený súborom stromov $T_1,...,T_S$, ktorých klasifikačné (resp. regresné) funkcie môžeme vyjadriť v tvare

$$\{d(\mathbf{x},\Theta_k), k = 1,...,S\},$$

kde $\mathbf{x}$ je vektor hodnôt prediktorov a $\Theta_1,...,\Theta_S$ sú nezávislé rovnako rozdelené náhodné vektory. Pre metódu `Random Forest` sa využívajú binárne stromy typu CART.

Náhodné lesy sú podstatnou zmenou oproti baggingu, ktorý necháva narásť veľké množstvo stromov a potom ich priemeruje. V mnohých prípadoch fungujú náhodne lesy veľmi podobne ako spomínané metódy a ich výhodou je jednoduché učenie a ladenie. Z toho dôvodu sú lesy populárne a implementované v rôznych balíčkoch.

Uvažujme opäť súbor trénovacích dát $\mathcal{L} = \{(y_i,\mathbf{x}_i), i = 1,...,N\}$. Trénovacími súbormi pre jednotlivé stromy $T_k, k = 1,...,S$ sú (rovnako ako v baggingu) bootstrapové výbery z datového súboru $\mathcal{L}$.

Náhodný les zvyšuje presnosť (znižuje vychýlenie) tým, že necháva narásť stromy do veľkej hĺbky, neprerezáva ich a zároveň udržiava znesiteľný rozptyl kombinovaním výsledkov jednotlivých stromov. Oproti ostatným lesom sa tu snaží autor zaistiť aj nízku koreláciu medzi jednotlivými stromami. Bootstrapové výbery totiž nie sú nezávislé, keďže sú tvorené výberom s vracaním. Teda výsledné stromy budú korelované a môže dôsť k nadhodnoteniu výsledkov klasifikácie alebo predikcie. Zníženie korelácie medzi stromami dosiahneme tým, že pri voľbe vetvenia pre daný uzol sa z M prediktorov $X_1,...,X_M$, ktoré máme k dispozícii, najskôr náhodne vyberie niekoľkých m_0. Následne sa najlepšie vetvenie hľadá klasicky, ale len medzi tými vetveniami, ktoré sú založené na vybraných m_0 veličinách.

Pozorovania, ktoré sú obsiahnuté v k -tom bootstrapovom výbere $\mathcal{L}_k$, sú použité ako trénovací súbor. Naopak pozorovania, ktoré sa do výberu nedostali, slúžia k odhadu chyby. Odhady chyby na testovacom súbore sa nazývajú ***oob*** (z anglického *out-of-bag*, *out-of-bootstrap sample*) **odhady**. Celkový počet *oob* pozorovaní tvorí tretinu dátového súboru.

Metóda `Random Forest` bola vyvinutá pre súbory obsahujúce veľké množstvo prediktorov, ale zároveň veľmi dobre funguje aj na malých dátových súboroch. Náhodné lesy môžeme použiť pre množstvo problémov, ako sú:

- klasifikácia alebo predikcia
- meranie významnosti premenných

- efekt premenných na predikciu
- zhlukovanie
- detekcia odľahlých hodnôt

Algoritmus 3.4 Random Forest

Vstup: trénovací súbor $\mathcal{L} = \{(y_i,\mathbf{x}_i), i = 1,...,N\}$
integer $ntree$ {*udávajúci počet stromov v lese*}
integer m_0 {*udávajúci počet prediktorov*}
integer min {*udávajúci minimálnu hodnotu veľkosti uzlu*}

1: **Inicializujeme**: $k = 1$
2: **repeat**
3: Vytvoríme bootstrapový súbor $\mathcal{L}_k$ o veľkosti N {*tvoríme trénovací súbor*}
4: Náhodne vyberieme m_0 prediktorov
5: Pomocou `CART` vytvoríme strom T_k zo súboru $\mathcal{L}_k$ {*Rast stromu sa zastaví, ak koncové uzly dosiahnu veľkosť min*}
6: Zaraď *oob* pozorovania {*tj. testovací súbor*} a urči výslednú klasifikačnú triedu (resp. predikciu) všetkých *obb* pozorovaní
7: $k = k + 1$
8: **until** $k = ntree$

Výstup: Spočítame celkový výsledok klasifikácie (resp. predikcie) pomocou väčšinového hlasovania (3.3) (priemerovania (3.1))

3.4.1 Voľba parametov m_0 a $ntree$

Určenie najvhodnejšej hodnoty závisí na úlohe a užívateľ pri tom môže experimentovať. Jednoduchou cestou, ako určiť vhodný počet stromov $ntree$, je opakované prevedenie experimentov s rôznymi nastaveniami, aby sme získali les, ktorý má najmenšiu chybovosť. To však vyžaduje vysokú časovú náročnosť, preto je vhodné vybrať taký počet stromov, ktorý bude dostačujúci pre optimálnu klasifikáciu. Na začiatku teda nastavíme počet stromov v lese na vyššiu hodnotu (napr. 20 násobok počtu prediktorov). Po určitom čase začne klasifikačná chyba s pribúdajúcim počtom stromov konvergovať ku určitej hodnote *oob* odhadu a stabilizuje sa. Potom minimálnu veľkosť lesa určíme ako počet stromov, kedy sa chyba *oob* odhadu s pribúdajúcimi stromami už nemení.

Pre počet náhodne vybraných prediktorov m_0 je doporučené následovné nastavenie:

- pre klasifikáciu sa volí $m_0 = \sqrt{M}$ a minimálna veľkosť terminálneho uzlu $min = 1$
- pre regresiu sa volí $m_0 = \frac{M}{3}$ a minimálna veľkosť terminálneho uzlu $min = 5$

Vyššie uvedené hodnoty slúžia ako defaultné nastavenia vo väčšine softvérov. V niektorých prípadoch sa ukazuje ako dobrá voľba m_0 blízke $\log_2 M$, alebo dokonca najlepšie $m_0 = 1$. To znamená, že sa prediktor, ktorý sa má použiť pre vetvenie vyberá úplne náhodne. To inými slovami znamená, že nezáleží na tom,

aké sa v konečnom dôsledku vetvenie vyberie, hlavné je, že sa priestor prediktorov vôbec nejako rozparceluje. Následné hlasovanie (resp. priemerovanie v prípade regresie) to dá potom všetko do poriadku.[1]

3.4.2 Významnosť premenných

Spomenuli sme, že náhodné lesy sú vhodné pre úlohy s mnoho prediktormi. Nie všetky prediktory však nesú významnú informáciu. Pre interpretáciu výsledkov poskytnutých metódou náhodných lesov je dôležité zistiť, ktoré premenné sú najdôležitejšie. K tomu slúži meranie významnosti premenných. Náhodné lesy poskytujú dve merania významnosti. Významnosť je počítaná buď pomocou Giniho indexu alebo je založená na poklese miery zlej klasifikácie MR (*misclassification rate*), kedy sú hodnoty prediktoru náhodne permutované [5].

Významnosť založená na randomizácii

Postup merania významnosti premenných môžeme zhrnúť do niekoľkých jednoduchých krokov. Po vytvorení i-tého stromu pre i-tý bootstrapový výber sú oob pozorovania stromom klasifikované do jedného z koncových uzlov a je určená presnosť ich klasifikácie OA (tj. percento správne klasifikovaných pozorovaní, viď vzorec (2.7)), príp. predikcie (presnosť stromu je založená na koeficiente determinácie R^2, viď vzorec (2.19)). Následne sú hodnoty j-tého prediktoru z oob výberu náhodne permutované a opäť sa pomocou príslušného stromu zistí výsledok klasifikácie, resp. predikcie pre tieto pozorovania. Na konci porovnáme výsledky pozorovaní u j-tého prediktoru zaťaženého šumom (randomizovaného) so správnou klasifikáciou týchto pozorovaní. Pokles presnosti predikcie stromu, ktorý nastane pri tejto randomizácii, je zpriemerovaný cez všetky stromy a je použitý ako meranie významnosti prediktorov. Týmto spôsobom získame hodnoty MR pre j-tý prediktor, ktoré určujú jeho významnosť. Meranie je často vyjadrené v percentách a je štandardizované na maximálnu hodnotu MR najvýznamnejšieho prediktoru, tj. najvýznamnejší prediktor má hodnotu $MR = 100\%$. Toto opakujeme pre každý prediktor.

Myšlienka celého postupu je jednoduchá: teda pokiaľ „zámena“ hodnôt nemá žiaden vplyv na výsledok, potom premenná nemá význam. Čím väčší rozdiel medzi náhodou a skutočnosťou dostaneme, tým väčší je aj vplyv premennej. Meranie významnosti nám ukazuje predikčnú silu danej premennej.

Významnosť založená na Giniho idexe

Pri rozdelení uzlu na dva dcérske uzly, pri ktorom je použitý Gini index (2.1), dochádza k poklesu tohoto indexu. Súčet poklesu v GI v jednotlivých stromoch pre každý prediktor udáva jeho významnosť.

Pozrime sa ešte na prípad, kedy hľadáme optimálnu sadu prediktorov. Pokiaľ je premenná významná, nemusí byť nutne použitá pri tvorbe lesa. Rovnako ako u stromov, môže byť použitá zástupná premenná s podobnou informáciou.

[1] To je mimochodom dosť protichodné s optimalizáciou stromov. Pripomeňme, že vetvenie sa pri stromoch optimalizuje za cenu drastických nákladov na výpočet a s mnohokrát nie práve uspokojujúcimi výsledkami generalizačnej chyby.

Sada významných premenných nemusí byť počet premenných použitých (či nutných) pre klasifikáciu, resp. predikciu, je však maximálnou sadou. Väčšinou sú ale všetky významné premenné vo výslednom lese obsiahnuté, pretože výberom rôznych trénovacích súborov sa nám líšia aj výsledné stromy.

V prípade veľkého počtu premenných je les na začiatku spustený raz so všetkými premennými a potom znovu len s použitím významných premenných, čím sa výrazne šetrí čas pri hľadaní optimálneho lesa.

3.4.3 Efekt premenných na predikciu

Použitie náhodného lesa pre klasifikáciu pozorovaní nám poskytne aj ďalšie užitočné informácie, hlavne pre interpretáciu získaných výsledkov. Okrem významnosti premenných nás môže zaujímať aj pre ktorú kategóriu, resp. kategórie, je významná. Túto informáciu získame z grafu efektu premennej na predikciu [2].

Pre prípady oob pozorovaní poznáme kategórie, do ktorej bolo pozorovanie zaradené. Môžeme teda zistiť podiel klasifikácie pozorovaní do jednotlivých kategórií *cpv* (*class proportion vote*).

Príklad 3.4.1. Uveďme jednoduchý príklad pre štyri kategórie A, B, C, D. Zo 100 stromov bolo pri klasifikácii zaradených 10 pozorovaní do A kategórie, 50 do B a 40 do C. Hodnoty *cpv* pre jednotlivé kategórie budú:

$$cpv_A = 0{,}1; \quad cpv_B = 0{,}5; \quad cpv_C = 0{,}4; \quad cpv_D = 0.$$

Pre každú kategóriu $j, j = 1,...,J$ a každú premennú prediktoru $\mathbf{X}_m$, $m = 1,...,M$ spočítame pravdepodobnosť zaradenia každého vzorku do kategórie j, kedy je m-tá premenná $\mathbf{X}_m$ randomizovaná a získame cpv_{Ran}. Rozdiel medzi cpv_{Ran} pre randomizovanú premennú a *cpv* pre premennú bez šumu udáva veľkosť zmeny pravdepodobnosti, označme *zpcpv* pre každú kategóriu u všetkých vzorkov:

$$zpcpv = cpv - cpv_{Ran}$$

Myšlienka je teda rovnaká ako pri určovaní významnosti premenných. Ak zobrazíme hodnoty *zpcpv* v závislosti od hodnôt premennej, získame graf efektu premennej na predikciu. Záporné hodnoty *zpcpv*[2] indikujú, že by klasifikácia tejto premennej bola pre ostatné triedy presnejšia.

3.4.4 Tesnosť (*proximity*)

Zmienili sme, že náhodné lesy je možné využiť rovnako ako techniku pre ordinačnú alebo zhlukovú analýzu, k čomu slúži meranie tesnosti. Stromy sa v náhodných lesoch neprerezávajú a nechávajú sa narásť veľké, z čoho je zrejmé, že pozorovania v koncových uzloch si budú veľmi podobné.

Po vytvorení všetkých stromov sú pozorovania (trénovacie aj testovacie) zaradené každým stromom. Vždy, keď sa ocitnú v rovnakom uzle, vzrastie ich tesnosť

[2]Zo vzťahu vidíme, že záporné hodnoty nastanú, ak je pravdepodobnosť vzorku pre danú kategóriu pri randomizovanej premennej vyššia, ako pri správnom poradí hodnôt tejto premennej.

o 1. Pre každý pár pozorovaní môžeme teda určiť, koľkokrát sa vyskytol v rovnakom koncovom uzle cez všetky stromy. Na konci sú tieto hodnoty normalizované podelením celkovým počtom stromov. Hodnoty tesnosti sú v intervale $\langle 0,1 \rangle$, pričom 1 znamená maximálnu tesnosť, tj. vo všetkých stromoch boli pozorovania zaradené rovnako, naopak 0 znamená, že pozorovania sa nikdy nevyskytli v rovnakom uzle.[13]

Uvažujme pozorovania $\mathbf{x}_i$ a $\mathbf{x}_j$, kde $i,j = 1,...,N$. Pri meraní tesnosti medzi týmito pozorovaniami je vytvorená matica tesnosti

$$Mprox = prox\{\mathbf{x}_i, \mathbf{x}_j\}$$

o veľkosti $N \times N$. Takto vytvorená matica je symetrická, pozitívne definitná a nadobúda hodnôt z intervalu $\langle 0,1 \rangle$ s prvkami na diagonále rovnými 1. Túto maticu podobnosti, prípadne maticu vzdialenosti $1 - Mprox$ je možné použiť v zhlukovacích a ordinačných metódach.

Najčastejšie využitie merania tesnosti je pre výpočet faktorových os vo viacrozmernom škálovaní, ktoré umožňuje projekciu vzoriek do priestoru s menej dimenziami, pričom zachováva vzdialenosť medzi objektmi.Tento postup môže byť veľmi využiteľný pre vizualizáciu výsledkov klasifikácie alebo k ohodnoteniu prekryvu jednotlivých skupín.

3.4.5 Prototypy kategórií

Meranie tesnosti je využité aj pri určení prototypov jednotlivých kategórií. Pre každú kategóriu j nájdeme pozorovanie, pri ktorom je počet pozorovaní z tej istej kategórie medzi jeho k najbližšími susedmi, ktorí sú definovaní na základe merania tesnosti. Medzi k pozorovaniami je určený medián a kvartily pre každý prediktor. Mediány sú prototypmi danej kategórie a kvartily nám dávajú odhad ich stability. Hodnoty sú štandardizované (odpočítaním 0,5 percentilu a podelením rozsahom medzi 0,5 a 0,95 percentilom). Pre kategoriálne premenné je prototypom kategória s najväčšou frekvenciou.

3.4.6 Detekcia odľahlých hodnôt

Pomocou matice tesnosti Breiman v špecifikácii náhodných lesov [13] a [15] definoval aj detekciu odľahlých pozorovaní. Za odľahlé pozorovanie je považované také, ktoré má najmenšiu tesnosť k ostatným pozorovaniam vo svojej klasifikačnej kategórii. Je teda definované len pre kategóriu, do ktorej patrí.

Autor definuje priemerný štvorec tesnosti pozorovania i od všetkých pozorovaní k v rovnakej kategórii j vzťahom

$$\bar{P}(i) = \sum_{c(k)=j} prox^2(i,k).$$

Miera odľahlosti pozorovania i je potom definovaná ako:

$$out(i) = \frac{N_j}{\bar{P}(i)},$$

kde N_j je počet pozorovaní v kategórii j. Ak je priemerná tesnosť i k ďalším pozorovaniam k v rovnakej kategórii malá, potom hodnota $out(i)$ bude veľká.

Pre všetky pozorovania v danej kategórii spočítame medián zo všetkých $out(i_j)$ hodnôt a ich absolútnu odchýlku od mediánu. Odpočítaním mediánu od hodnoty $out(i)$ a podelením jej absolútnou odchýlkou získame finálne meranie odľahlosti pozorovaní i.

Hodnoty $out(i) < 0$ sú prevedené na nulu. Pokiaľ je $out(i) > 10$, pozorovanie je považované za odľahlé.

3.4.7 Chýbajúce hodnoty

Použitím náhodných lesov pre klasifikáciu môžeme doplniť chýbajúce hodnoty. Existujú dve možnosti, ako les túto náhradu prevádza. Prvá, rýchlejšia a jednoduchšia cesta (ale menej presná) je nahradenie chýbajúcej hodnoty x_n mediánom hodnôt m-tej premennej v kategórii j závislej premennej. Pokiaľ je premenná kategoriálna, k doplneniu hodnoty je použitá kategória s najvyššou frekvenciou opäť len v príslušnej kategórii závislej premennej.

Druhá možnosť využíva maticu tesnosti. Táto varianta je pomerne presná a vhodná pre dátové súbory s veľkým množstvom chýbajúcich hodnôt. Ide o iteratívny proces, preto časová náročnosť vzrástla oproti predchádzajúcemu riešeniu. Chýbajúca hodnota x_n m-tej premennej X_m je nahradená váženým priemerom pozorovaní x_k, ktorých hodnoty poznáme. Ako váha je použitá hodnota z matice tesnosti $prox(x_n, x_k)$. U kategoriálnej premennej je chýbajúca hodnota opäť nahradená najviac frekventovanou hodnotou, ktorá je vážená tesnosťou.

Nahradené hodnoty sú použité pri ďalšej iterácii lesa a sú spočítané nové hodnoty tesnosti. Proces sa zastaví, pokiaľ už nedochádza k žiadnemu zlepšeniu, príp. je možné zvoliť pevne počet iterácií.

Kapitola 4

Praktická aplikácia

V tejto časti práce budeme aplikovať algoritmy, predstavené v predchádzajúcich kapitolách. Všetky použité skripty a dátové súbory sú obsiahnuté na priloženom CD, ktorého podrobný popis je v prílohe D.

Najskôr budeme riešiť úlohu segmentácie zákazníkov. Prvým krokom v riešení a analýze segmentácie zákazníkov je špecifikácia problému, ktorý chceme následnou klasifikáciou vysvetliť a interpretovať. Segmentácia zákazníkov totiž môže byť zameraná na riešenie rôznych problémov. Jedným z príkladov je, že chce banka (prípadne iná inštitúcia) pod rastúcim vplyvom konkurencie zlepšiť svoje služby. Klasifikáciou môžeme dospieť k lepšiemu pochopeniu klientov a cielenejšie zacieliť marketingové akcie ku koncovým užívateľom. Príkladom takéhoto typu môže byť snaha banky uložiť časť peňazí klientov na zvláštny účet s napr. dlhšou výpovednou lehotou, lepšie úročený a reklamou správne zacieliť na takýchto potencionálnych klientov. Ďalším z riešených problémov môže byť otázka včasného rozpoznania klientov, ktorí predstavujú rizikovú skupinu z hľadiska splácania úveru, resp. rozlíšiť medzi rôznymi skupinami klientov (bonitní, problémoví).

V našom prípade sa zameráme na segmentáciu zákazníkov žiadajúcich o poskytnutie úveru. Pri poskytovaní osobných pôžičiek sa prihliada hlavne na bonitu klienta, teda úverovú schopnosť klienta, ktorá sa odvíja od výšky príjmov, výdajov, od počtu vyživovaných detí v domácnosti a niekedy sú zahrnuté aj údaje o vzdelaní, ktoré kladne, prípadne záporne vplývajú na bonitu. Ďalšími dôležitými faktormi je ochota a schopnosť zákazníkov splácať správne plynúce splátky úrokov a istiny. Potenciálny dlžník je pri posudzovaní zaradený buď do skupiny bezproblémových dlžníkov alebo do triedy problematických klientov, pri ktorých prebieha bližšie skúmanie, v ktorom prípade nebude úver poskytnutý. Každý úverový zákazník sa vyznačuje radom vlastností, ktoré charakterizujú jeho osobnú, ekonomickú a právnu situáciu. Na základe týchto vlastností a informácií o klientovi sa pokúsime dosiahnuť štatisticky správne rozhodovanie o udelení alebo zamietnutí úveru.

Budeme analyzovať dátový súbor bývalých klientov nemenovanej veľkej banky v Nemecku s počtom 1000 bývalým dlžníkom, ktorých životné podmienky charakterizujú hlavne ordinálne a nominálne charakteristiky. Realita ukázala, že 30 percent z týchto bývalých zákazníkov nebolo schopných splácať úver podľa vopred dohodnutých podmienok.

4.1 Popis dátového súboru

Dátový súbor pozostáva z 1000 prípadov klientov charakterizovaných 20 prediktormi. Klienti sú zaradení do dvoch skupín z hľadiska poskytnutia úveru pomocou kategoriálnej premennej `kredit` s hodnotami **0** (klient nie je schopný splácať úver za stanovených podmienok) a **1** (klient je schopný splácať úver). Súbor bol už predspracovaný a kategoriálne premenné boli prevedené na nominálne a ordinálne premenné podľa tabuľky B.1 v prílohe B, kde je podrobný popis uvažovaných charakteristík klientov.

4.2 Aplikácia CART

V prípade dátového súboru **kredit** sme konštruovali klasifikačný strom. Konštrukcia prebiehala pomocou softvéru RStudio a knižnice `rpart`. Najskôr bol konštruovaný klasifikátor na celom dátovom súbore a postupovalo sa následovne:

Model Klasifikačný strom necháme vyrásť do maximálnej hĺbky.

Kriteriálna štatistika Pre delenie bol použitý Giniho index

Validácia modelu Aplikujeme 10-násobné krížové overovanie a na základe parametru zložitosti α (*complexity parameter*) vyberieme optimálnu hodnotu tak, aby mal strom dostatočnú presnosť a zároveň chyba medzi testovacím a trénovacím súborom pri krížovom overovaní bola čo najmenšia.

Orezávanie Pomocou optimálnej hodnoty α z predchádzajúceho kroku skonštruujeme optimálny strom pomocou orezávania.

Chyba klasifikácie Na základe vytvoreného stromu určíme chybu klasifikácie $e(t)$(*resubstitution error).*

Klasifikačný strom konštruujeme pomocou jednoduchého skriptu v RStudiu zobrazeného v ukážke kódu 4.1.

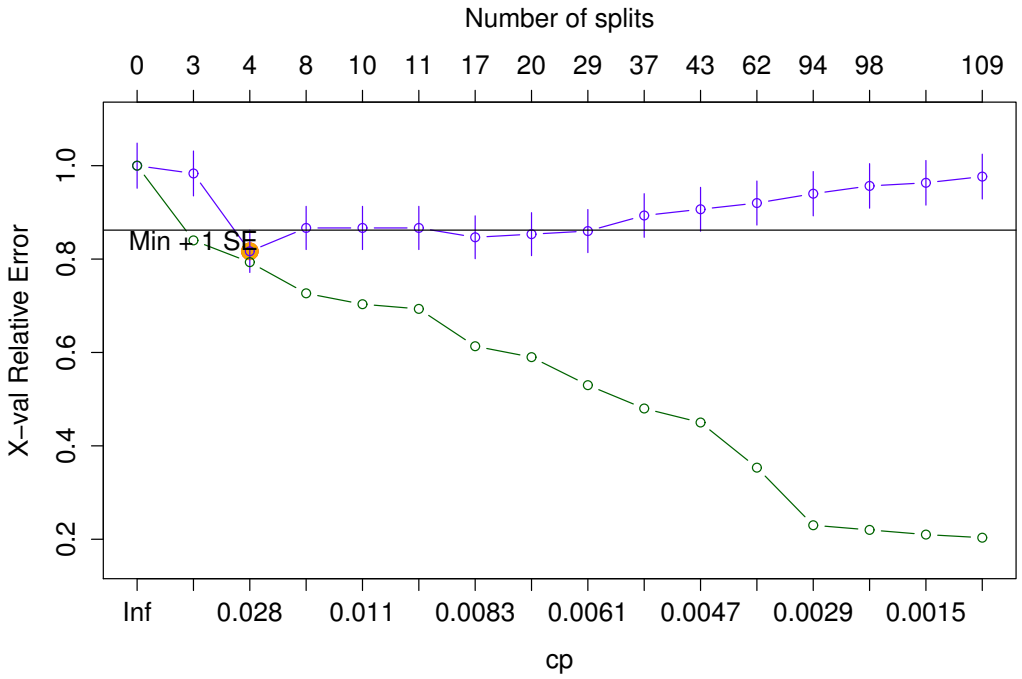
```
rpart(formula = kredit ~ ., data = kredit, cp = 0, xval = 10)
```

Ukážka kódu 4.1: Konštrukcia rozhodovacieho stromu

V teoretickej časti práce sme sa zmienili, že pri hľadaní optimálneho stromu vyberáme optimálny strom pomocou kritéria zložitosti, viď vzťah (2.6), kde hľadáme α minimalizujúce chybu pri krížovom overovaní. V softvérovom prevedení, parameter `cp` predstavuje hodnotu α podelenú chybou klasifikácie (resp.predikcie) v koreni stromu. Ak nastavíme parameter zložitosti na hodnotu `cp` $= 0$, docielime konštrukciu maximálneho stromu. Parameter `xval` $= 10$ predstavuje aplikáciu 10-násobného krížového overovania, tj. každý podsúbor bude obsahovať 100 pozorovaní. Okrem spomenutých a použitých parametrov má funkcia `rpart` k dispozícii parametre `minsplit` (minimálny počet pozorovaní, ktoré majú byť ešte oddelené do ďalšieho uzlu) a `minbucket` (udávajúci minimálny počet pozorovaní v koncovom uzle). Pokiaľ je nastavený len jeden z nich, potom pre druhý z nich platí, že `minbucket` $= \frac{\texttt{minsplit}}{3}$ alebo `minsplit` $= 3 *$ `minbucket`. Ostatné parametre

sa týkajú hľadania počtu zástupných a kompetitívnych premenných. Všeobecne sú nastavené `maxcompete` (počet kompetitívnych premenných) a `maxsurrogate` (počet zástupných premenných) na nulu.

Nechali sme teda vytvoriť maximálny strom. Keďže pozorovaní a uzlov v takto vytvorenom strome je veľmi veľa, nebudeme ho uvádzať ani v textovej ani v grafickej podobe. Zistili sme však, že niektoré koncové uzly obsahujú len jedno pozorovanie. Takto vytvorený strom môžeme považovať za pretrénovaný. Výsledný strom budeme chcieť validovať a zistiť jeho optimálnu veľkosť, na čo slúži práve krížové overovanie. Graficky si zobrazíme vývoj chyby na trénovacom a testovacom súbore pri krížovom overovaní, ktorý je zobrazený na obrázku 4.1. V hornej časti grafu, na hornej x-ovej osi, je uvedený počet delení daného stromu, dolná os zobrazuje hodnoty parametru `cp` a y-ová os zobrazuje chybu dosiahnutú pri krížovom overovaní. Zobrazená je závislosť geometrických priemerov z intervalu hodnôt `cp` na chybe testovacích a trénovacích súborov pri krížovom overovaní. Zelenou farbou je zobrazený vývoj chyby na trénovacích dátach a modrou na testovacích dátach.



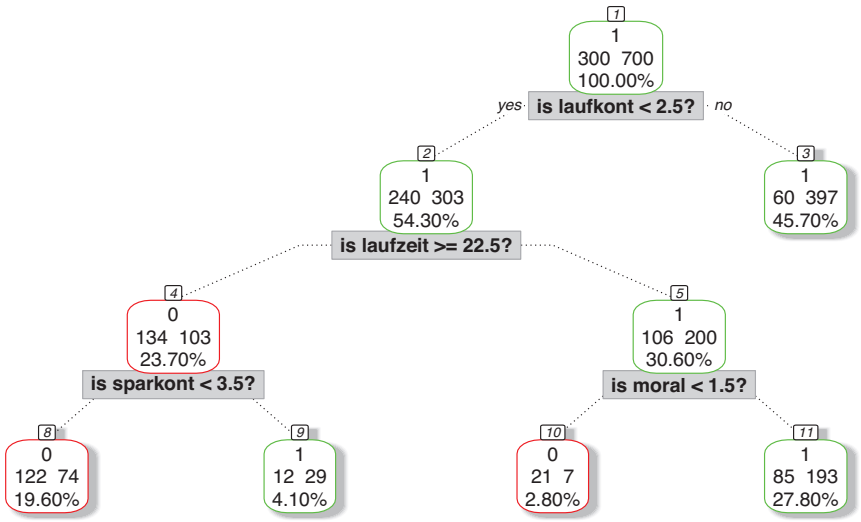
Obr. 4.1: Vývoj relatívnej chyby pri krížovom overovaní

Vidíme, že sa chyba na testovacom súbore v strome obsahujúcom viac ako 4 delenia zvyšuje. Podľa kritéria zložitosti vyberieme taký strom, ktorého chyba medzi testovacím a trénovacím súborom pri krížovom overovaní je čo najmenšia a zároveň chceme dosiahnuť aj čo najväčšiu presnosť stromu. Krivka závislosti testovacej chyby na `cp` postupne klesá pri počte 4 delení dosahuje hodnotu nižšiu ako referenčná čiara, ktorá uvádza hodnotu minimálnej chyby pri krížovom overovaní plus 1 SE, tj. štandardná chyba odhadu. Doporučuje sa vybrať strom, ktorého priemerná hodnota `cp` leží ako prvá pod čiarou. V našom prípade tretí strom s počtom delení 4.

Rovnaké výsledky a hodnoty `cp` a chýb môžeme v RStudiu zobraziť pomocou tabuľky (uvádzame skrátený výstup), ktorá obsahuje hodnoty chýb pre trénovací a testovací súbor a smerodatnú odchýlku testovacej chyby:

```
          CP nsplit rel error    xerror       xstd
1  0.05167      0     1.000     1.000     0.0483
2  0.04667      3     0.840     0.983     0.0481
3  0.01667      4     0.793     0.817     0.0453
4  0.01167      8     0.727     0.867     0.0462
5  0.01000     10     0.703     0.867     0.0462
6  0.00889     11     0.693     0.867     0.0462
7  0.00778     17     0.613     0.847     0.0459
```

Náš optimálny strom má hodnotu `cp` v intervale $\langle 0.0167, 0.0467 \rangle$. Zvolíme teda hodnotu `cp` = 0,0167. Dostali sme veľmi jednoduchý model stromového klasifikátoru s 5 koncovými uzlami, ktorý je graficky zobrazený na obrázku 4.2.



Obr. 4.2: Optimálny strom

Z piatich koncových uzlov klasifikujú 3 uzly klientov ako vyhovujúcich a schopných splatiť pôžičku v stanovených podmienkach a 2 koncové uzly klasifikujú klientov ako nevyhovujúcich. V každom uzle (nie len koncovom) je zobrazená trieda klientov (0 alebo 1), ďalej počet nevyhovujúcich klientov a počet vyhovujúcich klientov a percentuálne vyjadrenie počtu pozorovaní v uzle z celkového počtu pozorovaní. Pozorovania sú rozdelené do dcérskych uzlov na základe hodnoty prediktoru. Deliace hodnoty premenných a operátory sa vzťahujú vždy k ľavej časti stromu.

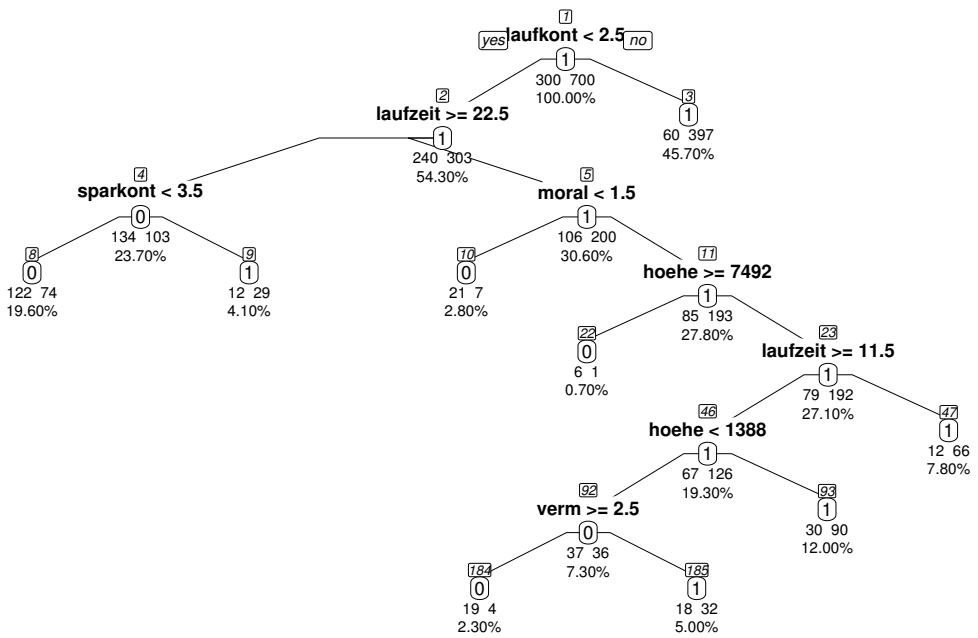
K prvému deleniu došlo podľa prediktoru `laufkont`, ktorý vyjadruje stav bežného účtu v banke. Hodnotám menším ako 2,5, tj. že klient nemá žiaden účet v banke alebo má nulový stav, prípadne debetný stav účtu zodpovedá okolo 54,3 % klientov dátového súboru. V rámci uzlu vidíme, že pravdepodobnosť zlej klasifikácie je veľmi vysoká, až 240 klientov nevyhovuje podmienkam splácania úveru a sú zaradení ako vyhovujúci. Klienti, ktorých hodnoty prediktoru `laufkont` sú 3 a 4, teda stav ich účtu sa pohybuje v kladných hodnotách sú pomerne presne klasifikovaní ako vyhovujúci. Klasifikačná chyba v tomto uzle je okolo 16 %.

Druhé delenie prebieha podľa hodnôt prediktoru `laufzeit`, tj. trvanie úveru v mesiacoch, kde vidíme, že klienti žiadajúci o dlhodobé pôžičky trvajúce viac

ako 22,5 mesiacov sú bankou považovaní za rizikových klientov. Tento dcérsky uzol (na obr. označený číslom 4) je ďalej rozdelený podľa prediktoru `sparkont` vyjadrujúceho vlastníctvo sporiaceho účtu prípadne cenných papierov v určitej hodnote nemeckých mariek. Pochopiteľne pre hodnoty účtu a cenných papierov nad 500 DM sú klienti klasifikovaní ako vyhovujúci, inak ostávajú zaradení v triede nevyhovujúcich klientov.

Dcérsky uzol, označený číslom 5, delí vyhovujúcich klientov na dve skupiny podľa prediktoru `moral`, tj. platobnej morálky. Opäť je oddelená už len malá skupinka klientov, ktorých platobná morálka mala v minulosti váhavý priebeh, prípadne majú kritický stav účtu, resp. existujúce úvery v inej banke, ktorí sú špecifikovaní ako nevyhovujúci.

Posúdenie toho, aký strom je optimálny záleží na subjektívnom názore analytika. Ak sa vrátime k obrázku 4.1, môžeme za prijateľný model považovať aj strom na obrázku 4.3 s počtom delení 8 a počtom koncových uzlov 9, ktorý je zložitejší, teda presnejší z hľadiska chyby na trénovacom súbore. Jeho dosiahnutá chyba na testovacom súbore však mierne prekračuje referenčnú čiaru. Ak sa pozrieme na koncové uzly, vidíme že v porovnaní s optimálnym stromom sa ďalším delením vytvárajú uzly s malým počtom pozorovaní, napríklad uzol č.22 a uzol č.184, ktoré vymedzujú ďalšie kritériá pre posúdenie nevyhovujúcich klientov. Sú nimi `hoehe`, tj. výška úveru a `werm`, tj.najväčšie existujúce aktívum. Klienti s vysokou výškou úveru nad 7 492 DM sú nevyhovujúci rovnako ako klienti s výškou úveru nižšou ako 1 388 DM a s aktívami: rodinný dom, pozemok, stavebné sporenie, príp. životné poistenie.



Obr. 4.3: Zložitejší strom s vyšším rozdielom medzi chybami

4.2.1 Klasifikačná chyba modelu

Na základe vytvoreného optimálneho stromu môžeme určiť chybu klasifikácie daného modelu, ktorá je $e(t) = 0{,}238$. Porovnaním pozorovaných hodnôt a hodnôt predikovaných modelom v tabuľke vidíme (stĺpce predstavujú hodnoty určené klasifikátorom a riadky pozorované hodnoty), že 81 klientov označených ako dobrých a schopných splatiť úver bolo klasifikovaných do skupiny klientov nevyhovujúcich podmienkam pridelenia úveru. V opačnom prípade až 157 nevyhovujúcich klientov bolo klasifikovaných ako klienti schopní splácať úver.

```
           predicted
               0   1
observed   0 143 157
           1  81 619
> error.rpart
[1] 0.238
```

Z hľadiska banky je tento výsledok pomerne neprijateľný. Správna klasifikácia nevhodných klientov je veľmi dôležitá, keďže práve títo klienti predstavujú riziko a stratu pre banku. Z toho dôvodu použijeme pri tvorbe modelu maticu strát (*loss matrix*).

4.2.2 Tvorba klasifikátorov s penalizáciou zlej klasifikácie

Matica strát je využívaná k penalizácii chybnej klasifikácie v prípadoch, ako sme popisovali vyššie, teda, že je podstatný rozdiel, či je nevyhovujúci klient klasifikovaný ako vyhovujúci (označme to ako chybu typu I, *false negative*) a naopak (chyba typu II, *false positive*). V následujúcich konštrukciách boli použité rovnaké parametre ako v predchádzajúcom prípade a naviac bola zavedená matica

$$L = \begin{pmatrix} 0 & L_{fn} \\ L_{fp} & 0 \end{pmatrix},$$

ktorá vyjadruje váhy, ktorými majú byť penalizované jednotlivé zlé zaklasifikované pozorovania pri danom delení na dcérske uzly.

Najskôr sme zvolili penalizáciu v pomere 2:1, teda je 2-krát horšia chyba typu I, ako typu II s maticou strát

$$L = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}$$

Výsledok klasifikácie po orezaní na optimálnu dĺžku stromu sa mierne zmenil, jednak sa znížila celková chyba klasifikácie a zároveň klesol počet chýb typu I:

```
           predicted
               0   1
observed   0 196 104
           1 118 582
> error.rpart
[1] 0.222
```

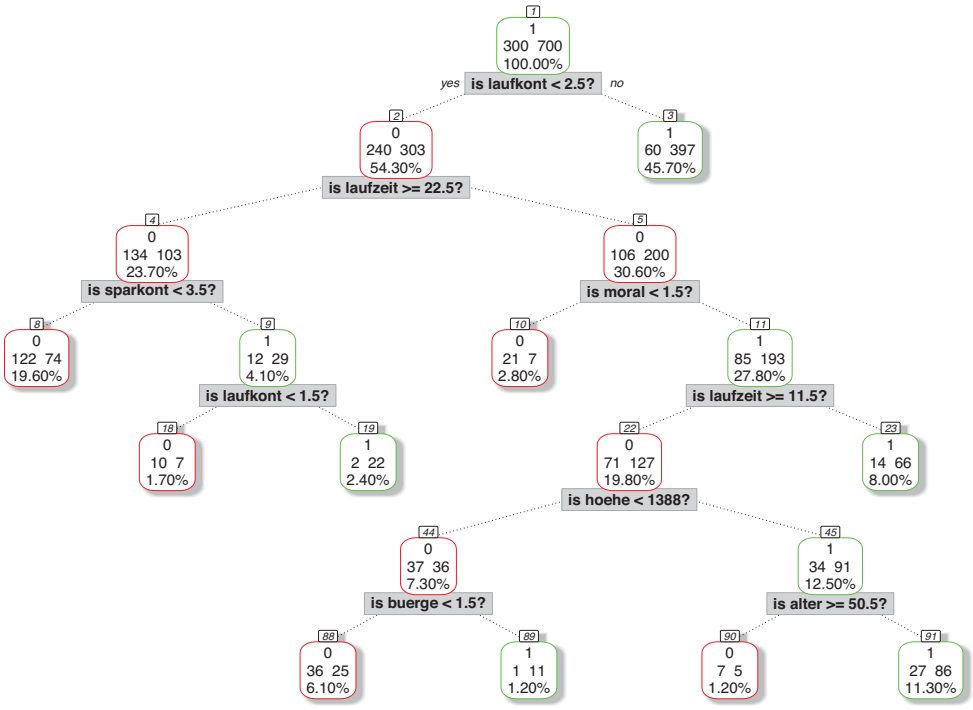
Ešte stále však tento výsledok predstavuje vysoké riziko pre banku. Penalizácia v pomere 4:1 priniesla zaujímavejšie výsledky z pohľadu redukcie nevyhovujúcich klientov ako vyhovujúcich:

```
           predicted
               0   1
observed   0 257 43
           1 291 409
> error.rpart
[1] 0.334
```

Vidíme, že celková chyba klasifikácie vzrástla o 10 percent, ale redukcia chyby typu I je dosť podstatná a z celkového počtu 300 klientov označených ako nevyhovujúcich je len 43 zle klasifikovaných. Narozdiel od toho sa podstatne zvýšil počet chyby typu 2.

Takže dokázali sme podstatne zredukovať chybu zle klasifikovaných nevyhovujúcich klientov, celková chyba klientov v tejto triede je okolo 14,3 %, zatiaľ čo pôvodný model klasifikoval týchto klientov s chybou 52,3 %. V prípade dobrých klientov, ktorí boli pôvodným modelom klasifikovaní s chybou okolo 11,6 %, sú teraz za nových podmienok klasifikovaní s chybou okolo 41,6 %.

Uvažujme situáciu z pohľadu veriteľa, tj. banky, že osoba, ktorá nesplní stanovené podmienky zmluvy a úver nesplatí, bude stáť veriteľa okolo 1000 DM, zatiaľ čo odmietnutím klienta, ktorý by pôžičku splatil, ho bude stáť 600 DM na poplatkoch, ktoré by zaplatil. Potom klasifikáciou pôvodným modelom stratila banka 205 600 DM. V druhom prípade prišla o 174 800 DM a použitím tretieho modelu bola jej strata 217 600 DM. Pri tomto scenári sa javí zvolená penalizácia



Obr. 4.4: Optimálny strom minimalizujúci stratu veriteľa

2:1 ako najvhodnejšia. Žiadna ďalšia uvažovaná penalizácia nám neposkytla lepší výsledok, teda penalizáciou 2:1 minimalizujeme celkovú stratu veriteľa. Vyjadríme ešte celkový zisk veriteľa pomocou zvoleného modelu, ktorý činí 174 400 DM. Pri pôvodnom nepenalizovanom modeli by činil zisk 165 800 DM a samozrejme v prípade tretieho modelu by bol zisk len 27 800 DM.

Vhodný model s penalizáciou je zobrazený na obrázku 4.4. Vidíme, že oproti optimálnemu stromu bez penalizácie dostávame zložitejšiu štruktúru stromu s 8-mimi koncovými uzlami. Oproti stromu na obrázku 4.3, je vyváženejší a využíva v ďalších deleniach iné prediktory. Okrem už spomenutých sa objavujú prediktory `buerge`, tj. ostatné pohľadávky alebo ručiteľstvo a `alter`, teda vek žiadateľa.

4.3 Aplikácia Random Forest

Ako bolo popisované v teoretickej časti tejto diplomovej práce, konštrukciou náhodných lesov by sme mali dospieť ku skvalitneniu klasifikácie. Tentokrát sme na výpočet a konštrukciu náhodného lesa použili knižnicu `randomForest` dostupnú pre RStudio. Knižnica presne implementuje náhodné lesy podľa programovej predlohy Lea Breimana napísanej pôvodne v jazyku Fortran. Iné matematické a štatistické softvéry (ako napríklad Matlab) túto metódu nemajú ani implementovanú, prípadne len z časti. Čo sa týka napríklad klasifikačných a regresných stromov, tak ich implementácia je dostupná vo viacerých softvéroch.

Náhodný les sme konštruovali pomocou skriptu uvedeného v ukážke kódu 4.2.

```
randomForest(formula = kredit ~ ., data = kredit,
type = "classification", importance = TRUE, proximity = TRUE)
```

Ukážka kódu 4.2: Konštrukcia náhodného klasifikačného lesa

Všetky ostatné hodnoty boli ponechané v základnom nastavení:
`randomForest(x, y=NULL, xtest=NULL, ytest=NULL,` / *testovací súbor nieje predom zadaný;*
`ntree=500,` / *počet stromov v lese;*

```
mtry=if (!is.null(y) && !is.factor(y))max(floor(ncol(x)/3), 1)
else floor(sqrt(ncol(x))),
```

/*počet náhodne vybraných prediktorov p;* $\sqrt{p}$ *pre klasifikáciu a p/3 pre regresiu*
`replace=TRUE` /*pozorovanie môže byť vybrané viackrát, pri procese rozdelenia na oob vzorky*;
`classwt=NULL` /*váha kategórií závislej premennej;*
`strata, sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),`
/*parametre pre stratifikovaný výber;*

```
nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
```

/*minimálny počet pozorovaní v koncovom uzle, 1 pri klasifikácii a 5 pri regresii;*
`maxnodes = NULL,` /*maximálny počet koncových uzlov stromu*;
`importance=FALSE,` /*výpočet významnosti premenných*;
`localImp=FALSE,` /*významnosť každého pozorovania*;

`nPerm=1,` */počet iterácií, kedy sú oob pozorovania permutované pre výpočet významnosti (importance) premenných, zatiaľ len pre regresiu;*
`proximity,` */výpočet matice tesnosti*;
`oob.prox=proximity,` */matica tesnosti len pre oob pozorovania*;
`norm.votes=TRUE,` */výsledné hlasovanie je vyjadrené ako podiel, inak priamo počet*;
`do.trace=FALSE,` */zobrazí výstupy procesu hľadania*;

```
keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE,
keep.inbag=FALSE, ...)
```

/ FALSE – *výsledok lesa nebude uložený vo finálnom výstupe.*

Na základe tohto základného nastavenia zostrojíme náhodný les a dostaneme následujúci výstup:

```
  Type of random forest: classification
                      Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 23.2%
Confusion matrix:
     bad good class.error
bad  132  168  0.56000000
good  64  636  0.09142857
```

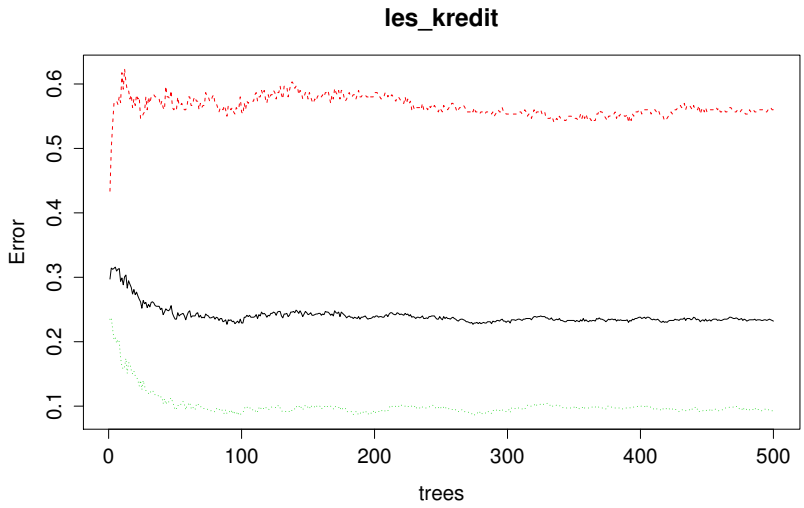
Les pozostáva z 500 stromov a náhodne vyberá zo 4 prediktorov. Chyba klasifikácie je 0,232, čo je v porovnaní s výsledkom klasifikácie pomocou jedného stromu, bez penalizácie maticou strát, o 0,6 percent lepšia. V klasifikačnej tabuľke sú uvádzané opäť ako v predchádzajúcich odstavcoch pozorované hodnoty v riadkoch a predikované hodnoty v stĺpcoch. Čo sa týka klasifikačných chýb v rámci jednotlivých tried, chyba typu I vzrástla v porovnaní s CART modelom a chyba typu II sa naopak zlepšila.

Stromy v zostrojenom náhodnom lese sa neprerezávajú. Počet uzlov v stromoch sa pohybuje okolo 201:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  171.0   194.0   201.0   201.3   209.0   229.0
```

Pri konštrukcii náhodného lesa je potrebné zadať niekoľko parametrov. Prvým z nich je parameter vyjadrujúci, koľko stromov má byť v lese skonštruovaných. Na obrázku 4.5, ktorý zobrazuje závislosť chyby na počte stromov v lese vidíme, že chyba klasifikácie sa stabilizuje pri hodnote okolo 250 stromov. Preto parameter `ntree` nastavíme na túto hodnotu.

Dôležité je určiť počet náhodné vybraných prediktorov, v RStudiu je na to parameter `mtry`, na základe ktorých sa budú jednotlivé uzly deliť. Pri náhodnom lese, ktorý sme skonštruovali boli vždy náhodne použité 4 prediktory. Pomocou funkcie `tuneRF` necháme vyhľadávať optimálny počet prediktorov. Pomocou parametru `improve` nastavíme, o koľko musí byť chyba na oob vzorku lepšia než je stanovená hodnota. Začíname od pôvodnej hodnoty `mtry=4` a dostávame chybu



Obr. 4.5: Závislosť chyby na počte stromov

na oob vzorku, OOB error=23,9 %, pričom ďalšie hodnoty `mtry` dosahujú vyššie chyby. Pri opakovanom spustení dostaneme odlišné výsledky, teda rozhodnutie je opäť subjektívne. Všeobecne platí, že pri nižšom náhodne zvolenom počte prediktorov sú výsledné stromy menej korelované.

```
mtry = 4  OOB error = 23.9%
Searching left ...
mtry = 3  OOB error = 24.2%
-0.0125523 0.05
Searching right ...
mtry = 6  OOB error = 24.5%
-0.0251046 0.05
```

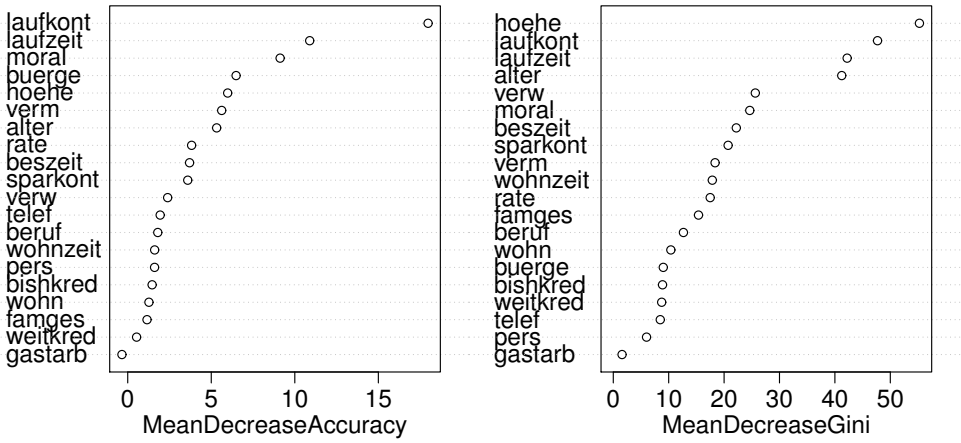
Rozhodli sme sa teda ponechať hodnotu `mtry=4` a môžeme skonštruovať náhodný les s optimálnymi parametrami. Rozdiel vo výsledkoch je minimálny, chyba klasifikácie sa zlepšila o 0,2 %.

```
Call:
 randomForest(formula = kredit ~ ., data = kredit,
 importance = TRUE, proximity = TRUE,
 ntree = 250, mtry = 4)

               Type of random forest: classification
                     Number of trees: 250
No. of variables tried at each split: 4

        OOB estimate of  error rate: 23%
Confusion matrix:
     bad good class.error
bad  135  165  0.55000000
good  65  635  0.09285714
```

Pre interpretáciu výsledkov nám poslúžia ďalšie funkcie, ktoré ponúka táto knižnica. Zaujímavé je pozrieť sa na to, ktoré premenné sú najdôležitejšie pre determináciu. Na obrázku 4.6 vidíme významnosť založenú na poklese klasifikačnej chyby pri randomizácii premennej a významnosť založenú na Giniho indexe, obe sú teoreticky popísané v sekcii 3.4.2. Poradie dôležitosti je v jednotlivých prípadoch odlišné.



Obr. 4.6: Významnosť založená na poklese chyby pri randomizácii a na Giniho indexe

V prípade významnosti založenej na poklese klasifikačnej chyby pri randomizácii ja najdôležitejšou premennou `laufkont` (stav bežného účtu a jeho trvanie), ďalej `laufzeit` (trvanie pôžičky v mesiacoch) a `moral` (platobná morálka klienta). Ďalšou významnou premennou je `buerge` (ručiteľstvo klienta alebo ostatné pohľadávky) a nasledujú premenné popisujúce ekonomické a sociálne postavenie klienta.

Najdôležitejšou premennou, pri významnosti založenej na Giniho indexe, je `hoehe` (výška úveru v DM), ďalej `laufkont` (trvajúci bežný účet v banke a jeho stav), `laufzeit` (trvanie pôžičky v mesiacoch) a `alter` (vek žiadateľa). Nasledujú premenné, ktoré sa ďalej dotýkajú ekonomickej situácie klienta a až ako posledné sú informácie o jeho sociálnom postavení.

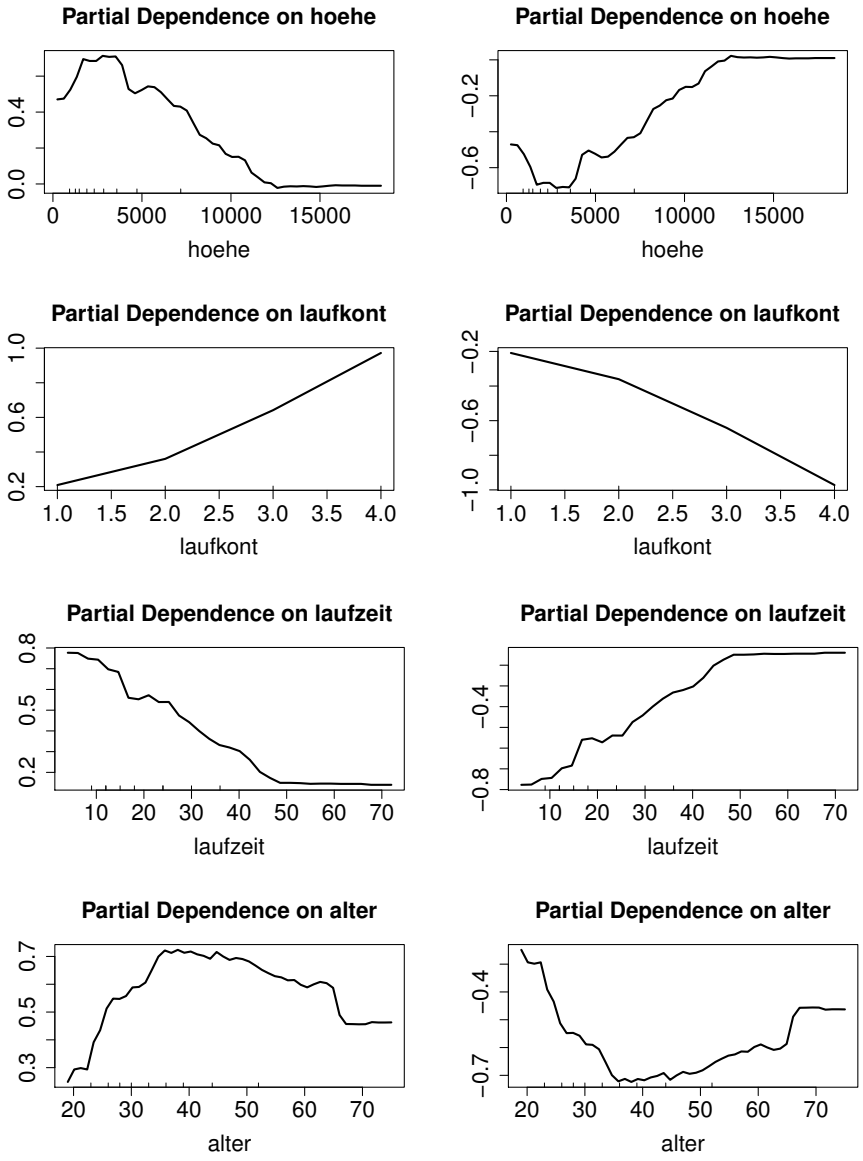
Najvýznamnejšie premenné odzrkadľujú tradičné kritériá bánk pri posudzovaní bonity klienta, takže dosiahnuté výsledky podporujú správnosť vytvoreného modelu.

V následujúcej tabuľke 4.1 je zobrazený počet, koľkokrát bola každá premenná použitá pri konštrukcii stromov a vidíme, že z vyššie popísaných štyroch významných premenných sú 3 najčastejšie používané pri delení.

laufkont	**laufzeit**	**moral**	**verw**	**hoehe**	**sparkont**	**beszeit**
2084	3508	1948	2864	4761	1722	2482
rate	**famges**	**buerge**	**wohnzeit**	**verm**	**alter**	**weitkred**
2054	1885	849	2161	2126	4276	1041
wohn	**bishkred**	**beruf**	**pers**	**telef**	**gastarb**	
1236	1223	1789	829	1153	220	

Tabuľka 4.1: Počet použitia danej premennej pri konštrukcii lesa

Získali sme teda predstavu o významnosti premenných, teraz sa pozrieme na ich chovanie v rámci jednotlivých kategórií. V následujúcich grafoch (obr. 4.7) vidíme efekt premennej na pravdepodobnosť kategórie, o ktorom sme teoreticky pojednávali v odstavci 3.4.3. Na ľavej strane sú zobrazené grafy pre kategóriu vyhovujúcich klientov, na pravej strane naopak pre klientov nevyhovujúcich. Vidíme, že pomocou týchto premenných dokážeme dobre charakterizovať jednotlivé kategórie. Klienti schopní splatiť úver sa vyznačujú nízkou výškou úveru, majú na bežnom účte kladný stav svojich prostriedkov, trvanie úveru je maximálne okolo 30-40 mesiacov a sú v produktívnom veku okolo 40 rokov. Rizikom pre banku je teda poskytovanie vysokých pôžičiek na dlhú dobu. Samozrejmosťou je, že klienti s nulovým stavom, prípadne debetným stavom účtu predstavujú takisto vysoké riziko. Tento stav úzko súvisí s vekom žiadateľov, kde predovšetkým mladí ľudia okolo 20 rokov a dôchodcovia (u nich je to však oveľa menšie riziko) takisto nie sú vhodným kandidátom pre poskytnutie úveru.



Obr. 4.7: Efekt premennej na pravdepodobnosť kategórie

Podobnú informáciu získame z prototypov kategórií založených na matici tesnosti. Prototypom je medoid z jeho najbližších susedov z príslušnej kategórie.

```
     laufkont laufzeit moral verw  hoehe sparkont beszeit rate
good        4       15     4    3 1898.0        2       4    4
bad         1       36     2    3 3365.5        1       3    4
     famges buerge wohnzeit verm alter weitkred wohn bishkred
good      3      1        3    2    37        3    2        2
bad       3      1        4    3    30        3    2        1
     beruf pers telef gastarb
good     3    1     1       1
bad      3    1     1       1
```

Ak porovnáme hodnoty prototypov, vidíme, že rozdiely sú najmä u 4 najvýznamnejších premenných. Ak sa pozrieme na vek jednotlivých reprezentantov kategórií, tj. premenná `alter`, predchádzajúce grafy nám poskytli lepšiu interpretáciu. Naše závery z predchádzajúcich grafov sa však potvrdili aj pri prototypoch.

Z hodnôt prototypov dostávame ešte ďalšie informácie o iných premenných. Napríklad premenná `moral` s nízkymi hodnotami okolo 2, čo predstavuje buď kritický priebeh predchádzajúcich úverov alebo neznámy priebeh, je charakteristická pre klientov nevyhovujúcich pre poskytnutie úveru. Zatiaľ čo hodnota 4, predstavujúca dobrý priebeh a platobnú morálku, charakterizuje klientov vhodných pre poskytnutie úveru.

Dĺžka zamestnania, premenná `beszeit`, takisto hrá rolu pri charakterizácii oboch skupín klientov. Nevhodní klienti majú dĺžku zamestnania kratšiu ako 4 roky. Pre nevhodných klientov je takisto charakteristické, že nemajú žiadne rezervy, tj. premenná `sparkont` má hodnotu 1.

Rozdiel v hodnotách premenných u prototypov zaznamenáme aj pri premennej `werm`, tj. najväčšie existujúce aktívum, kde klienti označení ako nevhodní majú stavebné sporenie, prípadne životné poistenie (hodnota 3). V prípade vhodných klientov ide o vlastníctvo auta (hodnota 2).

Ďalšie hodnoty premenných u prototypov sú zhodné, čo vyjadruje aj ich nízka významnosť.

4.3.1 Random Forest s penalizáciou zlej klasifikácie

Ako bolo spomenuté, výsledná klasifikácia však nie je podľa predstáv banky. Pomocou parametru `cutoff` môžeme opäť penalizovať zle klasifikované pozorovania.

Parameter predstavuje vektor rovnakej dĺžky, ako je počet jednotlivých tried. Predvolená hodnota je $1/k$, kde k je počet tried (tj. vyhráva trieda podľa väčšinového hlasovania). Pri zmene hodnôt je „víťaznou“ triedou pri hlasovaní tá, ktorá má maximálny pomer podielu hlasov ku `cutoff` [14].

Pri nastavení napríklad `cutoff` $= (0{,}25; 0{,}75)$ sme dosiahli podstatné zníženie chyby typu I, chyba celkovej klasifikácie stúpla na 34 % a model je porovnateľný s CART modelom pri penalizácii 4:1.

```
Call:
 randomForest(formula = kredit ~ ., data = kredit,
 importance = TRUE,proximity = TRUE, cutoff = c(0.25, 0.75))

               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 34%
Confusion matrix:
     bad good class.error
bad  251   49   0.1633333
good 291  409   0.4157143
```

Pozrieme sa na prototypy takto vytvoreného náhodného lesa:

```
     laufkont laufzeit moral verw hoehe sparkont beszeit rate
good        4       18     4    2  2255        3       4    4
bad         2       24     2    2  2255        1       3    4
     famges buerge wohnzeit verm alter weitkred wohn bishkred beruf
good      3      1        4    2  37         3    2        2     3
bad       2      1        3    3  31         3    2        1     3
     pers telef gastarb
good    1     1       1
bad     1     1       1
```

Rozdiely jednotlivých reprezentantov sa znížili, najmä vo výške úveru, kde sú hodnoty tentokrát rovnaké a takisto v dĺžke trvania úveru. Zvýraznil sa rozdiel v stave sporiaceho účtu a takisto vekový rozdiel. Aj rodinný stav a pohlavie sa špecifikovali rôzne, predstaviteľom vhodných klientov je muž ženatý alebo vdovec, nevhodnými typmi sú slobodní muži a ženy buď rozvedené, oddelene žijúce a vydaté. Hodnoty platobnej morálky boli očakávané, s dobrým priebehom už ukončených úverov je charakterizovaný vhodný klient, v prípade žiadnej histórie úverov je charakterizovaný nevyhovujúci klient.

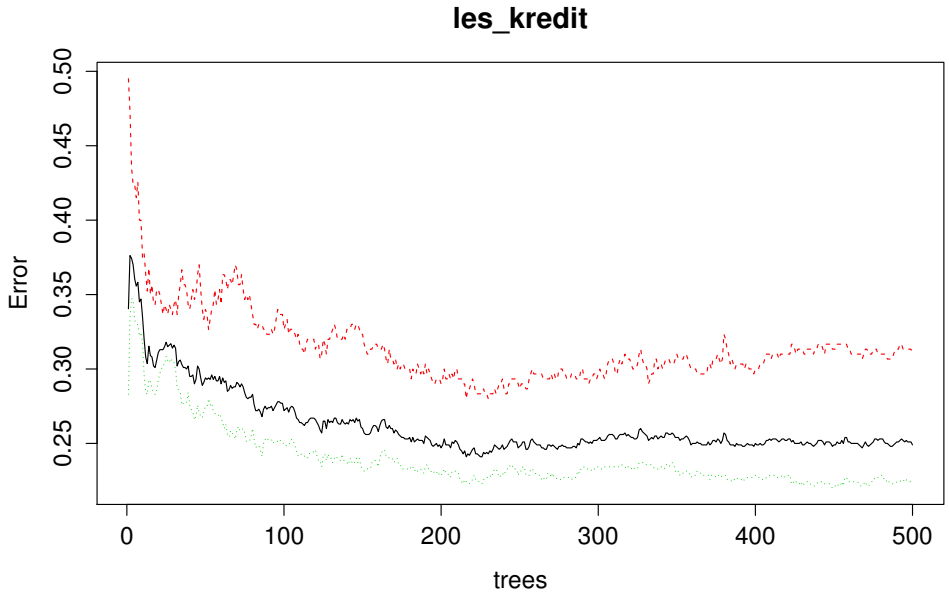
Pozrieme sa ešte na scenár veriteľa, popisovaný vyššie. Za takejto situácie by klasifikáciou pôvodným náhodným lesom prišla banka o 204 000 DM, pričom celkový zisk by činil 177 000 DM. S penalizovaným modelom by prišla dokonca o 223 600 DM a zisk je pochopiteľne len 21 800 DM. Stratu veriteľa, tj. banky, minimalizujeme zostrojením náhodného lesa s penalizujúcim vektorom `cutoff` $= (0{,}36; 0{,}64)$, pričom celková strata banky bude 186 600 DM a zároveň celkový zisk banky je vyčíslený na 139 800 DM. Vidíme, že v tomto prípade model, ktorý minimalizuje stratu veriteľa, neposkytuje zároveň najvyšší zisk. Chyba klasifikácie optimálneho lesa pre minimálnu stratu veriteľa je 24,9 %, pričom chyba klasifikácie prvej kategórie je 31 % a vhodných klientov model klasifikuje s chybou okolo 22 %.

```
Call:
 randomForest(formula = kredit ~ ., data = kredit,
 importance = TRUE,
 proximity = TRUE, cutoff = c(0.36, 0.64))
```

```
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 24.9%
Confusion matrix:
     bad good class.error
bad  207   93   0.3100000
good 156  544   0.2228571
```

Parameter `mtry` bol ponechaný na hodnote 4 a `ntree` bol nastavený na 300 stromov, ako vidíme na grafe závislosti celkovej chyby klasifikácie od počtu stromov 4.8.

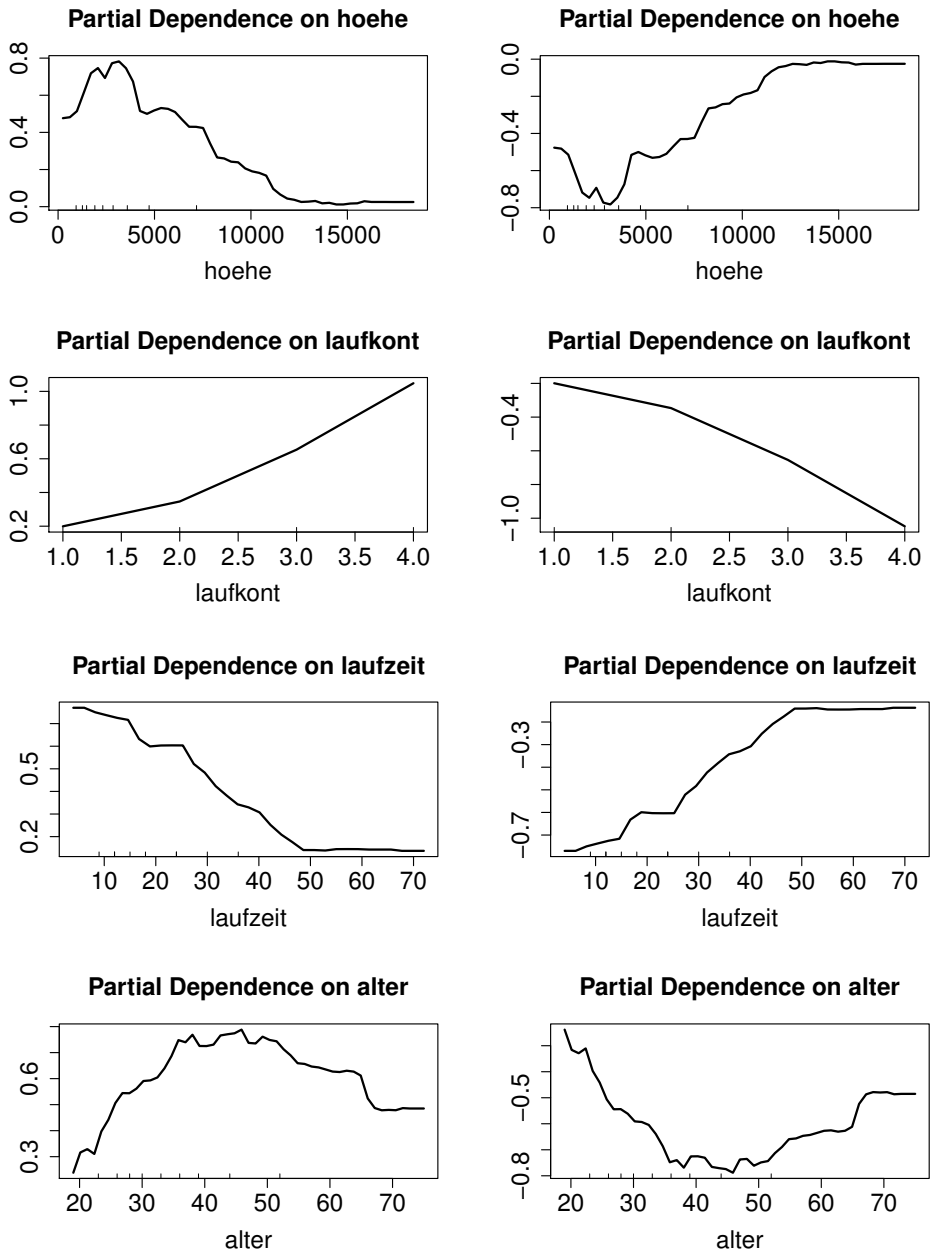


Obr. 4.8: Závislosť chyby na počte stromov

Na obrázku 4.9 je zobrazený efekt jednotlivých premenných na pravdepodobnosť kategórie a v porovnaní s pôvodným modelom nepozorujeme výrazné zmeny.

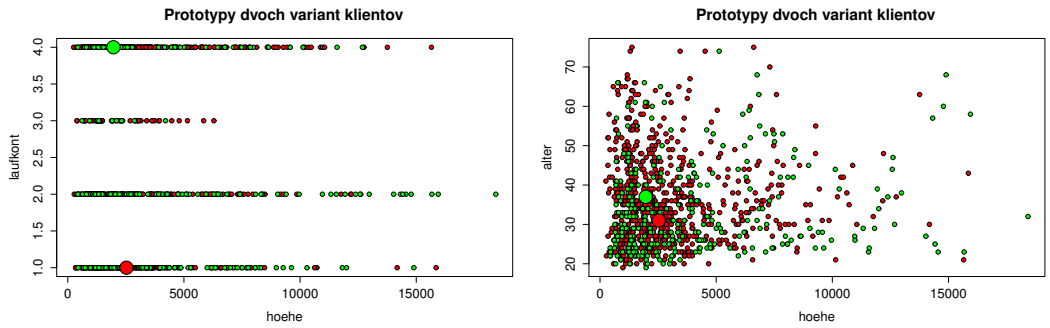
Rozdielov v reprezentantoch jednotlivých kategórií oproti predošlým modelom nie je veľa. Opäť vidíme podstatnejší rozdiel medzi výškou úveru. Celkovo sa hodnoty štyroch najvýznamnejších premenných znížili oproti pôvodnému náhodnému lesu pre kategóriu nevyhovujúcich klientov.

```
     laufkont laufzeit moral verw hoehe sparkont beszeit rate
good        4       15     4    3  1965        3       4    4
bad         1       24     2    2  2526        1       3    4
     famges buerge wohnzeit verm alter weitkred wohn bishkred beruf
good      3      1        3    2    37        3    2        2     3
bad       3      1        3    3    31        3    2        1     3
     pers telef gastarb
good    1     1       1
bad     1     1       1
```



Obr. 4.9: Efekt premennej na pravdepodobnosť kategórie

Na obrázku 4.10 sú zobrazené prototypy kategórií vrátane všetkých pozorovaní. Zelenou farbou je označený prototyp vhodných klientov a červenou nevhodných. Na grafe (4.10a) vidíme zobrazenie hodnôt výšky úveru v závislosti od stavu a trvania bežného účtu. Druhý graf (4.10b) zobrazuje závislosť výšky úveru od veku klienta. Obidva grafy potvrdzujú výsledky, ktoré sme interpretovali doposiaľ, tj. nevyhovujúci klient je charakterizovaný nižším vekom (sú to najmä mladí ľudia a z časti potom dôchodcovia). Nevyhovujúce kritérium je takisto žiaden založený bežný účet v banke, prípadne debetný a nulový stav na jeho účte a vysoká hodnota žiadaných úverov s dlhou dobou splatnosti. Zároveň vidíme, že hodnoty vyhovujúcich a nevyhovujúcich klientov sa prekrývajú vo všetkých kategóriách. Každý klient tak musí byť posúdený individuálne a rozdiely by sme určite našli aj v ďalších menej dôležitých premenných vytvoreného modelu.



(a) Prototyp `hoehe` vs. `laufkont` (b) Prototyp `hoehe` vs. `alter`

Obr. 4.10: Prototypy kategórií

4.3.2 Zhrnutie

Z vytvorených modelov klasifikačných stromov CART a náhodných lesov by bol v prípade banky preferovaný model CART s penalizáciou zlej klasifikácie 2:1, ktorý poskytol nielen najnižšiu stratu banke, ale aj vysoký zisk, porovnateľný so ziskom nepenalizovaného modelu náhodných lesov. Penalizácia náhodných lesov sa ukázala ako vhodný nástroj pre zlepšenie zlej klasifikácie a minimalizovanie straty, avšak zisk bol oproti pôvodnému modelu náhodných lesov menší. Segmentácia zákazníkov na daných modeloch je porovnateľná s kritériami z praxe pri posudzovaní bonity klienta.

4.4 Zmeny chyby klasifikácie pri použití CART a skupinových modelov

V tejto časti sa zameráme na zhodnotenie výsledkov klasifikácie z hľadiska chyby klasifikácie na testovacích súboroch. V rámci analýzy sme použili okrem nášho pôvodného súboru ďalšie súbory, ktoré sú reálne aj simulované a konštruujú klasifikačné a regresné stromy. Súbory pre klasifikačné stromy a lesy sú popísané v tabuľke 4.3 a regresné stromy v tabuľke 4.2. Podrobnejší popis týchto súborov je uvedený v prílohe C. Či už pri regresných alebo klasifikačných typoch súborov boli vybrané súbory s malým aj veľkým počtom prediktorov a pozorovaní.

Tabuľka 4.2: Dátové súbory pre regresné metódy

Dáta	Trénovací s.	Testovací s.	vstup
BostonHousing	456	50	13
Abalone	3759	418	8
Friedman1	200	1000	10
Friedman2	200	1000	4
Friedman3	200	1000	4

Tabuľka 4.3: Dátové súbory pre klasifikačné metódy

Dáta	Trénovací s.	Testovací s.	vstup	počet tried
kredit	900	100	20	2
BreastCancer	629	70	9	2
Ionosphere	316	35	33	6
ringnorm	200	1000	20	2

4.4.1 Priebeh výpočtov

Dátový súbor bol náhodne rozdelený na testovací súbor $\mathcal{T}$ a trénovací súbor $\mathcal{L}$. Pri reálnych súboroch tvorí testovací súbor $\mathcal{T}$ 10 % pôvodných dát. Pri simulovaných súboroch obsahoval testovací súbor väčšie množstvo pozorovaní ako trénovací. Pre jednotlivé metódy prebiehali výpočty nasledujúcim spôsobom:

CART

i Klasifikačný strom je konštruovaný z $\mathcal{L}$ použitím 10-násobného krížového overovania. Kriteriálnou štatistikou je Gini index. Klasifikačná chyba $e_s(\mathcal{L},\mathcal{T})$ bola určená aplikáciou skonštruovaného klasifikátoru na testovací súbor $\mathcal{T}$.

ii Regresný strom je konštruovaný analogicky s využitím 10-násobného krížového overovania. Celková stredná štvorcová chyba $e_s(\mathcal{L},\mathcal{T})$ bola opäť určená aplikáciou skonštruovaného stromu na testovací súbor $\mathcal{T}$.

Náhodné delenie dát na trénovací $\mathcal{L}$ a testovací $\mathcal{T}$ súbor je opakované 10 krát a následne je vypočítaná chyba klasifikácie $\bar{e}_s(\mathcal{L},\mathcal{T})$ ako priemer chýb cez všetky iterácie.

Bagging

i Bootstrapový výber $\mathcal{L}_B$ je náhodný výber z $\mathcal{L}$ a na jeho základe strom rastie. Pôvodný súbor $\mathcal{L}$ je využitý ako testovací súbor pre určenie správneho orezania stromu. Tento postup je opakovaný pri klasifikácii 50 krát (pri regresii 25 krát), čím konštruujeme klasifikátory $\phi_1(\mathbf{x}),...,\phi_{50}(\mathbf{x})$ (prediktory $\phi_1(\mathbf{x}),...,\phi_{25}(\mathbf{x})$).

ii Klasifikačnú chybu $e_B(\mathcal{L},\mathcal{T})$ určíme aplikáciou skonštruovaného modelu klasifikátorov na testovací súbor $\mathcal{T}$.

iii V regresnom prípade je pre $(y_n, \mathbf{x}_n)$ agregovaný prediktor určený priemerovaním a je určená stredná štvorcová chyba $e_B(\mathcal{L},\mathcal{T})$.

Adaboost.M1

i Pri konštrukcii lesa pomocou tejto metódy bolo použitých 50 stromov a následne bola určená klasifikačná chyba.

Random Forest

i Pri konštrukcii náhodného lesa (regresného aj klasifikačného) sa použilo 100 stromov a opäť boli určené jednotlivé chyby.

Voľba počtu stromov pri jednotlivých metódach je rovnaká, ako voľba Breimana v jeho článkoch [11] a [9]. Takisto výber dátových súborov zodpovedá z časti tým, ktoré použil autor.

4.4.2 Zhrnutie výsledkov: Regresné úlohy

Na regresné dáta sme použili metódy CART, Bagging a Random Forest. Tabuľka 4.4 zobrazuje priemerné štvorcové chyby na jednotlivých testovacích súboroch, ktoré sme získali v 10-tich iteráciách, kde v každej iterácii boli dátové súbory rozdelené náhodne na testovací a trénovací súbor. Testovacie súbory pri reálnych dátatch tvorili v každom prípade 10 % pôvodného súboru. Dátové súbory `Friedman1`, `Friedman2` a `Friedman3` pozostávajú z generovaných dát a testovací súbor obsahuje 5-krát viac dát.

Tabuľka 4.4: Priemerné stredné štvorcové chyby na testovacích súboroch

Dáta	CART	Bagging	Random Forest
BostonHousing	20.306	10.617	9.678
Abalone	5.719	4.930	4.602
Friedman1	12.164	6.510	6.952
Friedman2	31142.930	21581.620	20704.620
Friedman3	0.045	0.024	0.023

Výsledky potvrdzujú zlepšenie klasifikácie pri použití skupinových modelov. Náhodné lesy vykazujú takmer vo všetkých prípadoch najlepšie zlepšenie v strednej štvorcovej chybe. Výnimkou je dátový súbor `Friedman1`, kde použitie baggingu ponúklo o niečo lepšie výsledky. Pokles v MSE pre tento súbor je takmer 50 % medzi chybou CART stromu a chybou pri baggingu.

Takto výrazný pokles je dosiahnutý aj pri dátovom súbore `Boston Housing`. Pre tento dátový súbor sú najvhodnejšou metódou náhodné lesy.

V prípade dátového súboru `Abalone` je zlepšenie najmiernejšie, čo môže byť spôsobené vysokým počtom pozorovaní a teda presnejšou tvorbou regresného stromu CART.

Simulované dáta vykazujú podobné výsledky. Chyba klasifikácie je v prípade súboru `Friedman2` oproti iným súborom veľmi vysoká, ale aj tu skupinové modely podstatne znižujú túto chybu. Analogicky ako reálny súbor `Abalone`, aj `Friedman3` nevykazuje tak podstatné rozdiely v jednotlivých chybách.

Na príkladoch však vidíme, že aj keď boli vybrané rôznorodé súbory, s veľkým aj malým počtom pozorovaní, skutočne sme dosiahli zlepšenie v chybách predikcie a skupinové modely zlepšujú predikciu modelu.

4.4.3 Zhrnutie výsledkov: Klasifikačné úlohy

Na klasifikačné dáta sme použili metódy CART, Bagging, Adaboost.M1 a Random Forest. Tabuľka 4.5 zobrazuje priemerné chyby klasifikácie na testovacích

súboroch v 10-tich iteráciách, analogicky ako v regresnom prípade. Dátový súbor `ringnorm` je tvorený z generovaných dát a jeho testovací súbor obsahuje opäť 5-krát viac dát. Ostatné dátové súbory obsahujú reálne dáta a ich testovacie súbory pozostávajú z 10 % pôvodných dát.

Dátový súbor `Breast Cancer` obsahoval chýbajúce hodnoty prediktorov v testovacích a trénovacích dátach. Funkciu `randomForest` však môžeme aplikovať len na kompletné dátové súbory. Preto sme pred samotnou konštrukciou modelov doplnili chýbajúce hodnoty spôsobom, popísaným v príručke ku programu Random Forest [15]. Na každý trénovací a testovací súbor bola najskôr aplikovaná funkcia `rfImpute` (viď. ukážka kódu 4.3). Na začiatku sú volené štartovacie hodnoty chýbajúcich hodnôt následovne:

- pre numerické prediktory sú chýbajúce hodnoty nahradené mediánom hodnôt v hodnotách daného prediktoru
- pre kategoriálne prediktory sú chýbajúce hodnoty nahradené najviac frekventovanou triedou v hodnotách daného prediktoru

```
train=rfImpute(formula = Class ~ ., data = train)
test=rfImpute(formula = Class ~ ., data = test)
```

Ukážka kódu 4.3: Výpočet chýbajúcich hodnôt prediktorov

Na dáta s doplnenými štartovacími hodnotami je následne aplikovaná funkcia `randomForest`, ktorej matica tesnosti je použitá na spresnenie chýbajúcich hodnôt. Pre spojité prediktory je doplnená hodnota váženým priemerom z nechýbajúcich pozorovaní, kde váhy sú hodnoty ich tesnosti. Pre kategoriálne prediktory je doplnenou hodnotou kategória s najväčšou priemernou tesnosťou.

Následne sme postupovali rovnako, ako v iných súboroch dát, kde sa na trénovacích dátach vytvoril les, ktorý bol následne aplikovaný na testovacie dáta a bola určená chyba klasifikácie.

Tabuľka 4.5: Priemerné chyby klasifikácie na testovacích súboroch

Dáta	CART	Bagging	Adaboost.M1	Random Forest
kredit	0.2710	0.2410	0.2420	0.2300
BreastCancer	0.0601	0.0386	0.0358	0.0272
Ionosphere	0.0969	0.0852	0.0656	0.0594
ringnorm	0.2556	0.114	0.0942	0.0743

Aj v prípade klasifikačných dát dostávame rovnaké závery, a to celkové zlepšenie chyby klasifikácie na testovacích súboroch pri použití skupinových modelov.

Pri dátovom súbore `kredit` je dosiahnutá chyba pomocou bagging-u o 0,1 % lepšia ako pri metóde Adaboost.M1. V ostatných prípadoch je rozdiel medzi dosiahnutými chybami pre tieto dve metódy podstatnejší a stále je lepšou variantou Adaboost. Rozdiel medzi CART stromom a náhodným lesom je okolo 5 % a v prípade analýzy kreditného rizika je tento rozdiel už dosť dôležitý a podstatný.

V ďalších dvoch dátových súboroch reálnych dát pozorujeme analogicky pokles chyby klasifikácie použitím skupinových modelov. Zatiaľ čo veľkosť chyby

pre bagging a Adaboost sú pre dátový súbor `Breast Cancer` takmer totožné, v prípade súbory `Ionosphere` je tento rozdiel okolo 2 %.

Simulované dáta `ringnorm` poskytujú zaujímavé výsledky. Rozdiel medzi najhoršou a najlepšou dosiahnutou hodnotou klasifikačnej chyby je okolo 17 %. Najväčší pokles chyby je medzi hodnotou chyby CART stromu a pri použití baggingu (okolo 14 %). Ďalšie poklesy už nie sú tak razantné.

Vidíme, že chyba klasifikácie dosiahnutá pri jednotlivých metódach a súboroch závisí od skúmaného problému. Neplatí vždy striktne, že náhodné lesy poskytujú najlepšie výsledky z hľadiska chyby. Príkladom bola úloha segmentácie zákazníkov, kedy náhodné lesy nepriniesli obrovské zlepšenie v skonštruovanom modeli. Avšak pri aplikácii modelov na testovacie dáta sme demonštrovali, že práve aplikácia skupinových modelov poskytuje výrazne lepšie výsledky v klasifikácii a predikcii nových pozorovaní. Z toho dôvodu by sme v rámci segmentácie zákazníkov pre nových klientov volili model skonštruovaný pomocou náhodných lesov.

Záver

Klasifikácia, ako štatistická disciplína, zahŕňa široké spektrum metód a algoritmov používaných pri analýze dátových súborov. Jednou z oblastí analýzy dát je v práci popisovaná segmentácia zákazníkov, ktorá hrá dôležitú rolu vo vzťahu klienta so spoločnosťou. Vhodne prevedená segmentácia môže pomôcť správne zacieliť marketingové kampane na koncového zákazníka, zároveň predstavuje vhodný nástroj na ochranu proti rizikám, vyplývajúcim z poskytovania produktového portfólia spoločnosti.

Cieľom práce bolo uviesť čitateľa do problematiky klasifikácie a segmentácie zákazníkov, ktorá je však omnoho obšírnejšia než zahŕňa táto práca. V rámci analýzy segmentácie zákazníkov sa v praxi stretneme s radou rôznych metód pre riešenie danej úlohy. Doporučuje sa použiť viacero metód a vybrať z nich tú najvhodnejšiu, prípadne ich výsledky kombinovať. Mojou úlohou bolo rozpoznanie rizikových klientov z hľadiska poskytovania pôžičiek, k čomu je vhodné zvoliť algoritmy pre tvorbu rozhodovacích stromov a rozhodovacích pravidiel.

Úvodné kapitoly diplomovej práce poskytujú teoretický základ metód konštruujúcich klasifikátory stromového typu. Podrobne sú predstavené aj skupinové modely, nazývané lesy, ktoré vznikajú kombináciou stromových klasifikátorov. Popísané postupy a metódy sú doplnené praktickou aplikáciou, ktorá pomáha lepšie porozumieť danú problematiku. Vhodným prínosom sú aj grafické výstupy a ukážky kódov implementované v prostredí RStudio, ktoré dopĺňajú zrozumiteľnosť popisovaných výsledkov segmentácie klientov.

V reálnom svete môže klasifikácia a segmentácia zákazníkov odhaliť zaujímavé skutočnosti, ktoré nie sú na prvý pohľad viditeľné a zrejmé. Na finančných dátach klientov banky som demonštrovala postup segmentácie a tvorbu klasifikačných stromov a lesov. Analýza priniesla zaujímavé výsledky, ktoré sú interpretované a diskutované v predchádzajúcej kapitole.

Zároveň práca zahŕňa aj konštrukciu regresných stromov a lesov, s podrobným teoretickým výkladom a poukázaním na rozdiely oproti klasifikačným stromom. V praktickej časti sú následne jednak na klasifikačných ako aj na regresných dátových súboroch prezentované klasifikačné a predikčné schopnosti modelov aplikované na nové testovacie vzorky dát.

Téma diplomovej práce bola zaujímavá a podnetná, pretože teoretické znalosti dopĺňa praktickou aplikáciou metód, ktorá celej práci dodáva zrozumiteľnosť. Myslím si, že výsledná práca predstavuje ucelený pohľad na konštrukciu klasifikačných a regresných stromov a lesov, a na ich aplikáciu v praxi.

Literatúra

[1] Holčík, J.: *Analýza a klasifikace dat.* První vydání, Brno: AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o., 2012, ISBN 978-80-7204-793-2.

[2] Komprdová, K.: *Rozhodovací stromy a lesy.* První vydání, Brno: AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o., 2012, ISBN 978-80-7204-785-7.

[3] Řezánková, H.; Húsek, D.: Klasifikace v programových systémech pro analýzu dat. *Sborník Robust'2000*, 2001: str. 257 – 266.

[4] Antoch, J.: Klasifikace a regresní stromy. *Sborník Robust'88*, 1988: s. 1–7.

[5] Hastie, T.; Tibshirani, R.; Friedman, J.: *The Elements of Statistical Learning.* Second Edition, New York: Springer-Verlag, 2008, 763 s.

[6] Breiman, L.: Bias, variance, and arcing classifiers. *Technical report*, ročník 460, 1996: s. 1–25.

[7] Breiman, L.: Arcing classifiers. *The Annals of Statistics*, ročník 26, č. 3, 1998: s. 801–849.

[8] Dietrich, G. T.: Bias-Variance Theory. `http://web.engr.oregonstate.edu/~tgd/classes/534/slides/part9.pdf`, slidy z prednášok.

[9] Breiman, L.: Random Forest. *Machine Learning*, ročník 45, 2001: s. 5–32.

[10] Klaschka, J.; Kotrč, E.: Klasifikační a regresní lesy. *Sborník Robust'2004*, 2004: s. 177–184.

[11] Breiman, L.: Bagging predictors. *Machine Learning*, ročník 24, 1996: s. 123–140.

[12] Freund, Y.; Schapire, E. R.: A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, ročník 55: s. 119–139.

[13] Breiman, L.; Cutler, A.: Random Forests. `http://www.stat.berkeley.edu/~breiman/RandomForests/`, stránka s podrobnou špecifikáciou metódy Random Forest.

[14] Liaw, A.: Package 'randomForest'. `http://cran.r-project.org/web/packages/randomForest/randomForest.pdf`, príručka ku knižnici Random Forest.

[15] Breiman, L.: Manual–Setting Up, Using, And Understanding Random Forests V4.0. `http://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf`, manuál k programu Random Forest vo Fortrane.

[16] Friedman, H. J.: Multivariate Adaptive Regression Splines. *The Annals of Statistics*, ročník 19, č. 1, 1991: s. 1–67.

Zoznam obrázkov

Zoznam tabuliek

Dodatok A

Základné princípy spracovávania dát

Téme spracovávania a analýze dát sa podrobne venuje autor Jiří Holčík v knihe Analýza a klasifikace dat [1]. Podľa autora je potrebné tieto rozhodovacie postupy najskôr sformalizovať, následne algoritmizovať a implementovať ich spravidla v počítačovom prostredí. Dôležitou súčasťou je samotná príprava vstupných dát vo formalizovanej podobe, vhodná k tomuto spracovávaniu.

Spracovávaním dát sa všeobecne snažíme skúmať vzťahy medzi stavmi, javmi a procesmi, ktoré charakterizujú určitý objekt a sú charakterizované nameranými dátami.

Vychádzame z takzvaného **reálneho problému** týkajúceho sa **reálneho objektu**, ktorý je prvotným impulzom pre dobývanie znalostí a informácií. Reálnym problémom je napríklad otázka nájdenia skupín zákazníkov obchodného domu alebo klientov banky, ktorým chceme ponúknuť určitý produkt. Teda reálny objekt je v tomto prípade štruktúra zákazníkov a klientov. V prípade klientov banky môžeme hľadať potencionálnych záujemcov o kreditné karty, prípadne hypotečné či spotrebné úvery. Reálny objekt o sebe poskytuje informáciu o stave (napr. vek, región, výška príjmu, počet využívaných produktov atď.). Táto informácia je ukrytá v dátach, ktoré objekt generuje a my sme schopní ich primerane presne zmerať. Dáta sú všeobecne mnohorozmerné (stav objektu je popísaný rôznymi premennými) a sú dynamické v čase.

A.1 Cieľ spracovania dát

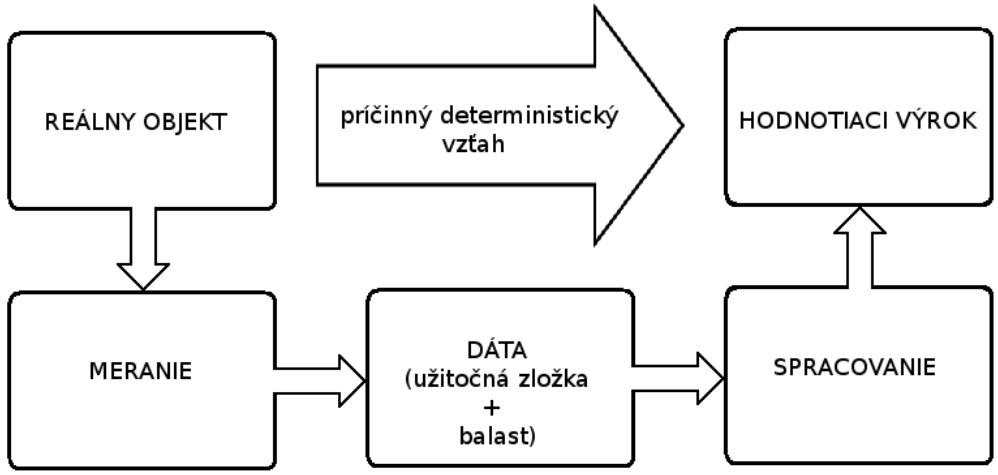
Cieľom spracovania (analýzy) dát je spravidla posúdenie skúmaného reálneho objektu, ktorý je zdrojom analyzovaných dát. Toto posúdenie môže najčastejšie vyústiť do:

- rozhodnutia o type či charaktere objektu – napríklad, že dané zviera je medveď hnedý alebo že daná budova je postavená v barokovom slohu – **klasifikačná**, resp. **rozpoznávacia úloha**;

- posúdenia kvality stavu analyzovaného objektu - napríklad, že pacient je v poriadku alebo má zdravotné problémy v podobe špecifikácie konkrétnej choroby - opäť **klasifikačná**, resp. **rozpoznávacia úloha**;

- rozhodnutie o budúcnosti objektu - aké bude sociálne zloženie obyvateľstva na danom území v danom čase - **klasifikačná** alebo taktiež **predikčná úloha** [1] ;

Teda hľadáme cestu od reálneho objektu k formálnemu výroku o jeho kvalite, stave, prípadne budúcnosti, ako je graficky znázornené na obrázku A.1.



Obr. A.1: Cieľ a kroky spracovania dát

A.1.1 Meranie

Ak hovoríme o spracovaní a analýze dát, potom v zobrazenom reťazci väčšinou ignorujeme blok MERANIE. Je však dôležité myslieť na to, že práve tento blok stojí za vznikom rôznych rušivých zložiek, ktoré namerané údaje obsahujú. V štatistike tieto rušivé zložky dát označujeme pojmom **variabilita dát**, ktorú je potrebné odstrániť, potlačiť, prípadne dostatočne vysvetliť.

A.1.2 Dáta

S meraním úzko súvisí následný blok DÁTA. Pre špecifikáciu problému a analyzovanie reálneho objektu je potrebné získať všetky dostupné dáta, ktoré môžu byť použité pri jeho riešení. Znamená to posúdenie všetkých dostupných údajov

[1] **Klasifikácia** a **predikcia** sú dva pojmy, ktorých použitie v odborných textoch a literatúre často splýva.

Pojem **predikcia** zjavne nesie časové (prípadne priestorové) hľadisko, ak ho používame vo význame predpovede či prognózy, čo sa stane alebo nestane v budúcnosti. V tomto význame je používaný napríklad v analýze a spracovaní časových radov.

Niekedy sa snaží odborná literatúra zámenu pojmov rozmotať konštatovaním, že pojem klasifikácia je používaný, ak sa použije klasifikačný algoritmus pre známe dáta. Pokiaľ sú dáta nové, pre ktoré nepoznáme klasifikačnú triedu, tak hovoríme o predikcii klasifikačnej triedy (to by znamenalo, že za klasifikáciu považujeme len procesy spojené s návrhom klasifikátoru, vlastná činnosť klasifikátoru by potom mala byť nazývaná predikciou).

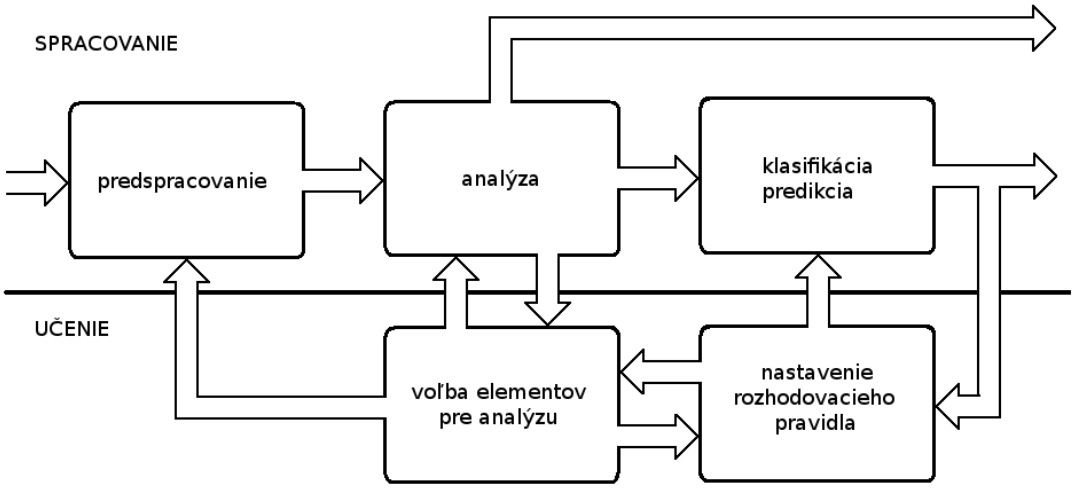
Za príjemnejšie rozlíšenie oboch pojmov považujeme výklad, že pojem klasifikácia používame, pokiaľ vyberáme identifikátor klasifikačnej tiedy z určitého diskrétneho konečného počtu možných identifikátorov. Pokiaľ určujeme (predikujeme) spojitú hodnotu, napríklad pomocou regresie, potom hovoríme o predikcii, aj keď tento pojem nezbytne časovú dimenziu nemá.[1]

a zváženie, nakoľko sú relevantné pri riešení. V niektorých prípadoch je potrebné pracovať aj s dátami, ktoré sú archivované po dlhšiu dobu a sú niekedy dokonca uložené v niekoľkých rôznych systémoch.

Často je vhodné uvažovať aj externé dáta, popisujúce prostredie, v ktorom sa analyzované deje odohrávajú. Ak sa vrátime k problému klientov banky, tak dôležitou informáciou je kalendárne obdobie (napr. Vianoce, Veľká noc, sviatky, dni, kedy dostávajú výplatu, obdobie dovoleniek a prázdnin atď.) Samozrejme má na zákazníkov vplyv aj počasie, prebiehajúce kampane a reklamy, v niektorých prípadoch ovplyvňujú rozhodovanie aj politické udalosti (napríklad názor klientov o vstupe do druhého piliera dôchodkovej reformy je ovplyvnený prognózou volieb v ČR).

A.1.3 Spracovanie

Po meraní a získaní všetkých dostupných informácií nasleduje ich SPRACOVANIE. Tento blok je rozsiahlejší a zahŕňa v sebe tri následné, podstatou odlišné operácie. Chronologicky sa jedná najskôr o **predspracovávanie** dát, **analýzu** a následnú **klasifikáciu**, resp. **predikciu**, ktoré môžu byť podporené fázou **učenia** ako je zobrazené na obrázku A.2.



Obr. A.2: Podrobná schéma bloku SPRACOVANIE

Teraz sa pozrieme na tieto tri základné piliere spracovania s tým, že zároveň pojednáme aj o dvoch základných podporných blokoch fázy učenia, ktorými sú **voľba elementov pre analýzu** a **nastavenie rozhodovacieho pravidla**.

A.1.3.1 Predspracovanie dát

Hlavným cieľom tejto fázy je takzvané **čistenie** a **filtrácia** dát. Zameriavame sa na postupy spojené s odstránením rušivých elementov z dát(odstránenie odľahlých hodnôt), rekonštrukciu alebo doplnenie chýbajúcich údajov, prípadne redukciu dát (odstránenie nezmyselných zložiek). V rámci tejto fázy sa dáta získané k riešeniu špecifikovaného problému pripravujú do formy vyžadovanej pre použitie a aplikáciu metód v analýze. Vo veľkom množstve prípadov sa môže jednať o náročné výpočtové operácie. Týmto procesom zaisťujeme čitateľnosť

a zrozumiteľnosť dát, čím sa zvyšuje aj ich kvalita. Predspracovanie dát sa principiálne môže líšiť od charakteru dát, ktoré môžu byť **statické** alebo **dynamické**. V oboch prípadoch však nájdeme pri spracovaní nejaký spoločný základ.

Súčasťou predspracovania môže byť aj prevod hodnôt kategoriálnych premenných do hodnôt, s ktorými možno následne robiť výpočty.

Z dôvodu porovnania hodnôt rôznych premenných sa často ich hodnoty upravujú napríklad **normalizáciou** (hodnota premennej vztiahnutá k definovanej norme), **centrovaním** (odčítanie strednej hodnoty) prípadne **štandardizáciou** (centrovaná hodnota je vztiahnutá k určitej špecifickej hodnote, často k smerodatnej odchýlke). Stále si ale musíme uvedomiť, že napríklad spomínanou štandardizáciou voči smerodatnej odchýlke sme sa v dátach zbavili informácie o strednej hodnote a rozptyle, čo môžu byť dôležité údaje pri následnom spracovávaní. Preto sa musí vždy zodpovedne zvážiť, či operácia, ktorá jednu fázu zjednoduší nespôsobí komplikácie z hľadiska ďalšieho riešenia.

A.1.3.2 Analýza dát a blok voľby elementov pre analýzu

Pojmom **analýza** rozumieme rozbor, metódu skúmania zložitejších skutočností rozkladom na jednoduchšie. Ide o dekompozíciu celku na elementárne časti, pričom cieľom je identifikovať ich podstatné a nutné vlastnosti, spoznať ich podstatu a zákonitosti. Analýza sa využíva v rôznych vedách, ale aj v bežnom živote, kde na základe detailného pozorovania chceme dospieť k výsledku.

V rámci spracovávania dát to v praxi znamená nájsť závislosti medzi hodnotami použitých premenných, nájsť zákonitosti v rozložení týchto hodnôt, prípadne stanoviť miery korelácie. V niektorých prípadoch ide priamo o nájdenie vhodného matematického vzťahu, ktorý vyjadruje funkčnú závislosť medzi použitými premennými. Výsledky analytických výpočtov môžu byť použité k spracovanie, prípadne pozmenení dát do vstupného formátu pre následný klasifikačný blok, ale takisto môžu byť aj finálnym výsledkom spracovania dát bez naväzujúcej klasifikácie.

Ďalším dôležitým krokom je redukcia dát a výber významných premenných, ktoré budú použité pri klasifikačných metódach, resp. predikcii. Nie je všeobecne predpísaný spôsob, akým sa tieto príznakové premenné, ktoré nesú najviac informácií pre klasifikáciu, určujú. Teória ponúka čiastočné riešenie, pri ktorom sa vyberá potrebný počet premenných z konečnej predom určenej množiny, prípadne sa pôvodné príznakové veličiny vyjadria pomocou menšieho počtu skrytých nezávislých premenných, ktoré sa nedajú priamo zmerať, ale môžu i nemusia mať vecnú interpretáciu.

A.1.3.3 Klasifikácia a nastavenie rozhodovacieho pravidla

V kapitole 1 bola predstavená klasifikácia objektov ako prístup, ktorý umožňuje zaraďovať na základe rozhodovacích pravidiel v klasifikátore neznáme objekty do klasifikačných tried. V literatúre rozlišujeme dva typy klasifikátorov - **deterministický** a **nedeterministický**. V prípade deterministického klasifikátoru budeme rozumieť postup, kedy sú vstupné dáta spracované s vždy rovnakým výstupom. Ak v postupoch vychádzame z pravdepodobnostných charakteristík spracovávaných dát, hovoríme o nedeterministickom klasifikátore a jeho opakovaným použitím dospejeme k rôznej klasifikácii. Nedeterministický klasifikátor

nemusí byť vždy len pravdepodobnostný. Príkladom sú aj iné matematické disciplíny, pracujúce s neurčitosťou, ako napríklad fuzzy logika, fuzzy algebra.

Často sa pri práci s algoritmami vyskytuje delenie na **parametrické** a **neparametrické** algoritmy, metódy, modely. Parametrický algoritmus pracuje na základe danej funkcie, ktorej konkrétne vlastnosti sú určené a môžu sa meniť s hodnotami konečného počtu stanovených parametrov. Príkladom parametrického klasifikačného algoritmu je *prahová klasifikácia* (vstupný obraz je zaradený do klasifikačnej triedy v prípade, že hodnota, charakterizujúca daný objekt, buď prekračuje alebo neprekračuje danú prahovú úroveň. Hodnotu prahovej úrovne určujeme vo fáze učenia. Typickým predstaviteľom parametrických klasifikačných algoritmov sú preto rozhodovacie stromy, ktorým bude venovaných niekoľko nasledujúcich kapitol. Ako príklad neparametrického klasifikačného algoritmu uvedieme *klasifikáciu podľa minimálnej vzdialenosti* od etalónu klasifikačnej triedy. V tomto prípade určujeme vzdialenosť vstupného obrazu od všetkých etalónov klasifikačných tried a obraz zaradíme do triedy, ktorej etalon má k obrazu najmenšiu vzdialenosť. Pre stanovenie vzdialenosti používame rôzne metriky, napr. Euklidovská metrika.

Rozdelenie podľa parametrickosti klasifikačného algoritmu ale nič nepredurčuje, pokiaľ ide o charakter algoritmov učenia klasifikátorov. Pre klasifikačné stromy napríklad existuje veľká trieda trénovacích algoritmov, ktoré si nekladú žiadne nároky na spôsob učenia. Učiaci postup nieje závislý na žiadnych partikulárnych parametroch, sú to teda algoritmy neparametrické. Klasifikácia je často krát pomerne jednoduchý postup a to zaujímavé, čo sa týka voľby klasifikátoru, je spôsob jeho návrhu, prípadne učenie. V odbornej literatúre je preto náhľad na typ klasifikátoru predurčený charakterom učiaceho algoritmu.

Rozhodovacie pravidlo hrá významnú rolu v rámci klasifikačného algoritmu. Na jeho základe je vstupná množina dát rozdelená do klasifikačných tried. Rozhodovacie pravidlá pri klasifikácii pracujú na základe vzdialenosti alebo podobnosti vstupných dát a vzormi klasifikačných tried, hraníc rozdeľujúcich obrazový priestor dát. Mieru príslušnosti ku klasifikačnej triede určujú **diskriminačné funkcie**, niekedy aj doplnkové logické pravidlá.

Výstupom tohoto bloku je návrh a určenie rozhodovacieho pravidla, ktoré bude uplatnené počas klasifikácie. V prípade, že pravidlo je parametrické, určujeme v tejto fáze aj jeho parametre.

Návrh všeobecného tvaru rozhodovacieho pravidla nie je formalizovaný a závisí vo veľkej miere na skúsenostiach konštruktéra buď s danou reálnou úlohou alebo s charakterom získaných dát. Čo sa týka návrhu parametrov rozhodovacieho pravidla, ten štandardne vedie k použitiu optimalizačnej úlohy. Deje sa to na základe takzvanej **učebnej** alebo **trénovacej** množiny, ktorá obsahuje vstupné obrazy spojené s informáciou o predpokladanej správnej klasifikácii. V tom prípade hovoríme o **učení s učiteľom** (podľa miery spoľahlivosti údaja o predpokladanej klasifikácii ide buď o učenie s *dokonalým* alebo *nedokonalým* učiteľom).

Ak nemáme trénovaciu množinu k dispozícii, potom tento blok zahŕňa len návrh všeobecného tvaru rozhodovacieho pravidla a prípadné nastavovanie parametrov prebieha zároveň so samotnou klasifikáciou. Tento postup nazývame **učením bez učiteľa**.

Dodatok B

Popis dátového súboru `kredit`

Tabuľka B.1: Dátový súbor **kredit**

Premenná	Popis		Hodnota
kredit	závislá premenná	úver bude splatený	1
		úver nebude splatený správne	0
laufkont	trvajúci bežný účet v banke	žiaden stav, príp. debet	2
		0 <= ... < 200 DM	3
		... >= 200 DM alebo min. 1 rok	4
		žiadny účet	1
laufzeit	trvanie úveru v mesiacoch		
moral	platobná morálka	žiadne úvery/všetky splatené	2
		predchádzajúce úvery v poriadku	4
		ešte trvajúce úvery, zatiaľ v poriadku	3
		váhavý priebeh úveru	0
		kritický stav účtu/existujúce úvery v inej banke	1
verw	účel úveru	nové auto	1
		ojazdené auto	2
		nábytok a zariadenie	3
		rádio/televízor	4
		veci do domácnosti	5
		rekonštrukcia	6
		vzdelanie	7
		dovolenka	8
		rekvalifikácia,školenia	9
		podnikanie	10
		iné	0
hoehe	výška úveru v DM		
sparkont	sporiaci účet cenné papiere	< 100,- DM	2
		100,- <= ... < 500,- DM	3
		500,- <= ... < 1000,- DM	4
		>= 1000,- DM	5
		neurčené / žiaden sporiaci účet	1
beszeit	dĺžka súčasného zamestnania	nezamestnaný	1
		<= 1 rok	2
		1 <= ... < 4 roky	3
		4 <= ... < 7 rokov	4
		>= 7 rokov	5
rate	splátky v % z príjmu	>= 35	1
		25 <= ... < 35	2
		20 <= ... < 25	3
		< 20	4

Pokračuje na ďalšej strane

Pokračovanie z predchádzajúcej strany

Premenná	Popis		Hodnota
famges	rodinný stav a pohlavie	mužské: rozvedený / žijúci oddelene	1
		ženské: rozvedená/ žijúca oddelene / vydatá	2
		mužské: slobodný	2
		mužské: ženatý / vdovec	3
		ženské: slobodná	4
buerge	ostatné pohľadávky, ručiteľstvo	žiadne	1
		spolužiadateľ	2
		ručiteľ	3
wohnzeit	momentálne bývanie od	<= 1 rok	1
		1 <= ... < 4 roky	2
		4 <= ... < 7 rokov	3
		>= 7 rokov	4
verm	najväčšie existujúce aktívum	dom a pozemok	4
		stavebné sporenie / životné poistenie	3
		auto/iné	2
		neurčené/ žiadne aktíva	1
alter	vek v rokoch		
weitkred	ostatné úvery	v inej banke	1
		v obchodnom dome, zásielkový predaj alebo iné	2
		žiadne	3
wohn	bývanie	podnájom	2
		osobné vlastníctvo	3
		beznákladové bývanie	1
bishkred	počet predchádzajúcich úverov s bankou (vrátane aktuálnych)	jeden	1
		dva až tri	2
		štyri až päť	3
		šesť a viac	4
beruf	povolanie	nezamestnaný/nekvalifikovaný striedajúci prácu	1
		nekvalifikovaný, stabilné povolanie	2
		robotníci, vyučení zamestnanci, úradníci	3
		výkonný, samostantný, povýšený zamestnanec	4
pers	počet nezaopatrených osôb	0 až 2	2
		3 a viac	1
telef	telefónny kontakt	nie	1
		áno pod menom klienta	2
gastarb	zahraničný pracovník	áno	1
		nie	2

Dodatok C

Popis dátových súborov

C.1 Dátové súbory pre regresné stromy a lesy

Boston Housing Dátový súbor pozostáva z 506 prípadov zodpovedajúcich sčítaniu traktov v aglomerácii Bostonu. Premenná y vyjadruje medián ceny domov v danej oblasti. K dispozícii máme 13 prediktorov, predovšetkým socio-ekonomického charakteru. Dátový súbor bol použitý už pri mnohých štúdiách.

Abalone Rozsiahly dátový súbor pozostávajúci z 4174 prípadov ústricových mušlí, ktoré sú charakterizované ôsmimi prediktormi, dosiahnuté pri fyzikálnych meraniach schránok ústric. Premenná y udáva vek mušle, ktorý sa meria pomocou kruhov viditeľných na schránke.

Friedman1 Simulovaný dátový súbor vytvorený Friedmanom pre MARS článok [16]. V prvom dátovom súbore máme 10 nezávislých prediktorov x_1,...,x_{10}, kde každý z nich má rovnomerné rozdelenie $R(0,1)$. Premenná y je daná vzťahom

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0{,}5)^2 + 10x_4 + 5x_5 + \varepsilon,$$

kde $\varepsilon \sim N(0,1)$. Friedman udáva výsledky pre dátové súbory s veľkosťou 50,100,150 a 200 príkladov. My sme použili veľkosť 200.

Friedman2, Friedman3 Tieto dátové súbory sú príkladmi simulovania impedancie a fázového posunu v obvodoch striedavého prúdu. Obsahujú dáta štyroch premenných definovaných vzťahmi

$$y = \left(x_1^2 + \left(x_2 x_3 - \left(\frac{1}{x_2 x_4}\right)\right)^2\right)^{\frac{1}{2}} + \varepsilon_2,$$

$$y = \tan^{-1}\left(\frac{x_2 x_3 - \frac{1}{x_2 x_4}}{x_1}\right) + \varepsilon_3,$$

kde x_1,x_2,x_3 a x_4 sú rovnomerne rozdelené v intervaloch

$$\begin{aligned} 0 &\leq x_1 \leq 100, \\ 20 &\leq \frac{x_2}{2\pi} \leq 280, \\ 0 &\leq x_3 \leq 1, \\ 1 &\leq x_4 \leq 11. \end{aligned}$$

Šumy $\varepsilon_2, \varepsilon_3$ majú rozdelenie $N(0, \sigma_2^2)$ a $N(0, \sigma_3^2)$, kde σ_2, σ_3 sú volené tak, aby pomer $\frac{signal}{noise}$ bol 3:1.

C.2 Dátové súbory pre klasifikačné stromy a lesy

kredit Dátový súbor klientov banky pozostávajúci z 1000 prípadov a 2 kategoriálnymi triedami. *Good*: klienti vyhovujúci podmienkam úveru, *bad*: klienti neschopní splatiť úver. Z pôvodných dát, 700 klientov je označených za vyhovujúcich a 300 ako nevyhovujúcich. Podrobný popis dátového súboru je uvedený v prílohe B.

Breast Cancer Dátový súbor získaných z Madisonskej nemocnice, ktorý sa vzťahuje ku karcinómu prsníka. Obsahuje 699 údajov o pacientoch, z ktorých 458 má nezhubný a 241 zhubný nádor. Každý pacient je charakterizovaný 9 prediktormi vyjadrujúcimi bunkové charakteristiky.

Ionosphere Radarové dáta od Johns Hopkins University, pozostávajúce z 351 prípadov charakterizovaných 33 prediktormi. Predmetom skúmania sú elektróny v ionosfére. Kategoriálna premenná nadobúda dve hodnoty: *good*- ak radarové merania ukazovali na nejaký typ štruktúry v ionosfére, *bad* sú označené tie, ktoré ju nemajú. Ich signály prechádzajú skrz ionosféru. Z celkového súboru 226 je označených *good* a 125 ako *bad*.

ringnorm Simulované dáta vytvorené Breimanom v článku [7], ktoré obsahujú 20 prediktorov a 2 kategórie tried. Prvá trieda je z mnohorozmerného normálneho rozdelenia s nulovou strednou hodnotou a kovariančnou maticou 4 krát identická matica. Druhá trieda má jednotkovú kovariančnú maticu a strednú hodnotu (a,...,a), kde $a = \frac{1}{\sqrt{20}}$.

Dodatok D

Obsah priloženého CD

- `prilohy` ..... zložka príloh k diplomovej práci
  - `classification` ..... klasifikačné dáta a skripty
    - *`názov dátového súboru`* ..... názov používaný v práci
      - `skripty`
      - `test` ..... testovacie súbory
      - `train` ..... trénovacie súbory
  - `Kreditnehmern` ..... úloha segmentácie zákazníkov
    - `kredit.asc` ..... pôvodný súbor `kredit`
    - `kredit.Rda` ..... súbor `kredit` vo formáte `.Rdata`
    - `skripty`
    - `test` ..... testovacie súbory
    - `train` ..... trénovacie súbory
  - `regression` ..... regresné dáta a skripty
    - *`názov dátového súboru`* ..... názov používaný v práci
      - `skripty`
      - `test` ..... testovacie súbory
      - `train` ..... trénovacie súbory
- `text`
  - `thesis.pdf` ..... text práce vo formáte PDF
  - `thesis.ps` ..... text práce vo formáte PS