UNIVERZITA KOMENSKÉHO V BRATISLAVE
Fakulta matematiky, fyziky a informatiky

KREDITNÝ SKÓRING POMOCOU NÁHODNÝCH LESOV

Bc. Jozef Gábik

BRATISLAVA 2010

UNIVERZITA KOMENSKÉHO V BRATISLAVE
Fakulta matematiky, fyziky a informatiky

KREDITNÝ SKÓRING POMOCOU NÁHODNÝCH LESOV

Diplomová práca

Študijný program:	Ekonomická a finančná matematika
Študijný odbor:	9.1.9 Aplikovaná matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Školiteľ:	Doc. Mgr. Marián Grendár, PhD.

Bc. Jozef Gábik

BRATISLAVA 2010

Týmto by som chcel poďakovať vedúcemu mojej diplomovej práce, Mariánovi Grendárovi, za jeho odborné vedenie a ľudský prístup. Ďakujem taktiež mojej Aďejke za jej blízkosť, povzbudzovanie, modlitby a pomoc pri kontrole textu. Ďakujem svojej rodine za veľkú trpezlivosť a podporu. Ďakujem Peťovi za ochotu počúvať a riešiť problémy v TEXu. Ďakujem Zuzke za pomoc pri získavaní dát. A najviac ďakujem Bohu.

Abstrakt

GÁBIK, Jozef* : *Kreditný skóring pomocou náhodných lesov.* [Diplomová práca.] Univerzita Komenského v Bratislave. Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej matematiky a štatistiky.
Vedúci diplomovej práce: Doc. Mgr. Marián Grendár, PhD. Bratislava: FMFI UK, 2010. 84 s.

*email: jozef.gabik@gmail.com

V tejto diplomovej práci je predstavený proces vývoja skóringového modelu. Kreditný skóring v súčasnosti využíva rôzne štatistické techniky na odhad pravdepodobnosti zlyhania. V práci predstavíme dve z nich - náhodné lesy a podmienené náhodné lesy. Na reálnych bankových dátach vyvinieme modely pomocou týchto metód a pomocou logistickej regresie, ktorá je v súčasnosti najpoužívanejšia. Výsledky porovnáme na základe rôznych štatistických mier.

Kľúčové slová: náhodné lesy, podmienené náhodné lesy, kreditný skóring, Bazilej II

GÁBIK, Jozef: *Credit scoring by random forests.* [Master thesis.] Comenius University, Bratislava. Faculty of Mathematics, Physics and Informatics; Department of Applied Mathematics and Statistics.
Supervisor: Doc. Mgr. Marián Grendár, PhD. Bratislava: FMFI UK, 2010. 84 p.

In this master thesis is presented the process of scoring model development. Credit scoring is currently using various statistical techniques to estimate the probability of default. The thesis will introduce two of them - random forests and conditional random forests. We will develop on the real bank data scoring models using these methods and logistic regression, which is currently the most common. The results will be compared by different statistical measures.

Keywords: Random forests, Conditional random forests, Credit scoring, Basel II

Obsah

Úvod

Rastúca konkurencia a stále sa zvyšujúce tlaky na produkovanie zisku a plnenie obchodných plánov viedli úverové inštitúcie[1] k hľadaniu efektívnejších riešení, ako minimalizovať neočakávané straty a získať klientov, ktorí by boli schopní bez problémov splácať poskytnutý úver. Agresívny marketing viedol k potrebe hlbšieho rozpoznávania rizikovej skupiny potenciálnych klientov, a takisto k potrebe rýchleho a efektívneho zapracovania klientov do procesu schvaľovania a následného poskytovania úveru. Tento proces sa, aj vďaka rýchlo sa vyvíjajúcim informačným technológiám, mohol z veľkej časti zautomatizovať. Banky dnes od svojho manažmentu rizika očakávajú rýchle a kvalitné posudzovanie kreditnej kvality svojich klientov s čo najmenšími nákladmi. Samozrejme, od procesu schvaľovania sa očakáva, že bude minimalizovať zamietanie kvalitných klientov, ktorí sú schopní splácať úver, a že minimalizuje poskytovanie úveru nespoľahlivým, rizikovým klientom.

Manažment rizika je tiež vyzývaný k oceňovaniu kvality už existujúcich klientov. Banky sa snažia ponúkať svojim klientom ďalšie produkty - úlohou manažmentu rizika je vybrať „správnych" - teda nerizikových klientov, ktorí budú oslovení. A naopak, určenie rizikových klientov môže banke pomôcť urobiť vhodné kroky, aby minimalizovala potenciálne straty.

Z týchto a ďalších dôvodov vznikol *kreditný skóring*. V súčasnosti existuje veľké množstvo rôznych štatistických techník, ktoré na základe historických dát určujú kreditnú kvalitu klientov alebo žiadateľov o úver. V praxi banky používajú najmä skóringové modely založené na logistickej regresii. V prvej kapitole tejto práce sa však bližšie oboznámime s ďalšími dvoma metódami - náhodnými lesmi a podmienenými náhodnými lesmi. Uvedieme aj rôzne vedľajšie produkty týchto metód, ktoré môžu byť užitočné napr. pri príprave dát.

[1] V ďalšom texte budeme úverovú inštitúciu pre jednoduchosť nazývať *banka*, hoci úverovou inštitúciou môže byť aj nebankový subjekt.

V druhej kapitole popíšeme proces vývoja skóringového modelu. Keďže v súčasnosti hrá kreditný skóring dôležitú úlohu pri výpočte minimálnych kapitálových požiadaviek banky, povieme si aj o dokumentoch Bazilejského výboru pre bankový dohľad, v ktorých sú stanovené podmienky pre skóringové modely.

V poslednej časti na reálnych bankových dátach vyvinieme skóringové modely pomocou všetkých spomenutých metód. Na záver porovnáme ich predikčnú silu a interpretujeme vplyv niektorých charakteristík na kreditnú kvalitu klienta.

Pri vývoji modelov budeme využívať najmä štatistický softvér R, pomôžeme si aj softvérom SAS® Enterprise Miner.

Kapitola 1

Náhodné lesy

V tejto kapitole zadefinujeme všeobecný klasifikačný problém a popíšeme *náhodné lesy* ako klasifikátor, t.j nástroj na riešenie klasifikačného problému. Skôr ako sa však oboznámime s náhodnými lesmi, povieme si o ich základnej stavebnej jednotke - o *klasifikačnom a regresnom strome*. Keďže, ako neskôr spomenieme, klasifikačné a regresné stromy sú vychýlené, predstavíme si aj nevychýlené *podmienené stromy* - z týchto stromov sa vytvárajú analogicky *podmienené náhodné lesy*.

Okrem popisu samotných algoritmov uvedených klasifikátorov sa budeme snažiť vymenovať aj ich výhody a nevýhody. V časti o náhodných lesoch tiež ukážeme spôsob, ako určiť predikčnú chybu a významnosť jednotlivých prediktorov. Povieme si aj o použití niektorých vedľajších produktov týchto klasifikátorov, akým je napríklad matica blízkosti.

1.1 Klasifikačná úloha

Definícia 1. *Nech y je vysvetľovaná premenná, ktorá je náhodnou premennou z priestoru $\mathcal{Y} = \{C_1, ..., C_J\}$ - t.j. y má J tried ($J \geq 2$). Ďalej majme m-rozmerný vektor prediktorov $\mathbf{X} = (X_1, ..., X_m)$ vzatý z priestoru $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_m$. Majme reprezentáciu náhodného výberu $\{(y_i, X_{1i}, ..., X_{mi}),\ i = 1, ..., n\}$ veľkosti n z priestoru $\mathcal{Y} \times \mathcal{X}$ a nazvime ho trénovacou množinou. Potom* ***klasifikátorom*** *nazývame funkciu $h : \mathcal{X} \to \mathcal{Y}$, ktorá zobrazuje pozorovanie $(X_{1i}, ..., X_{mi})$ na nejakú (odhadnutú) triedu $\hat{y}_i$. Klasifikátor priraďuje túto triedu na základe informácie z trénovacej množiny. Hovoríme, že klasifikátor sa „učí" na trénovacej množine.*

Odhadnuté triedy by sa, prirodzene, mali čo najlepšie zhodovať so skutočnými triedami. Kvalitu klasifikátora v tomto zmysle môžeme posúdiť pomocou tzv.

	$\hat{y} = 1$	$\hat{y} = 0$	Chyba modelu
$y = 1$	True Positives	False Negatives	FN/(TP+FN)
$y = 0$	False Positives	True Negatives	FP/(FP+TN)

Tabuľka 1.1: Matica zatriedenia

matice zatriedenia (*confusion matrix*). Je to matica typu $J \times J$, kde jej ij-ty prvok zodpovedá počtu pozorovaní, ktoré mali skutočnú triedu C_i a klasifikátorom odhadnutú triedu C_j. Pre binárnu premennú y_i (teda $J = 2$) je matica zatriedenia znázornená v tabuľke 1.1. Často sa pre binárnu premennú triedy nazývajú *Positives* ($y = 1$) a *Negatives* ($y = 0$), preto sú použité aj pre túto tabuľku. Posledný stĺpec udáva spôsob výpočtu chýb modelu. Celková chyba modelu sa počíta ako (FN+FP)/N, kde N je počet všetkých pozorovaní v množine.

Niekedy je potrebné dať rozdielnu váhu chybe False Positives ako False Negatives. Hovoríme o rozdielnych relatívnych nákladoch. Typickým príkladom je test výskytu nejakej choroby u konkrétneho človeka. Náklady na liečbu zdravého človeka v prípade, že ho test klasifikuje ako chorého (False Positive) sú spravidla nižšie ako v opačnej situácii - ak chorého človeka považujeme za zdravého (False Negative). V tomto prípade sme ochotní mať vyššiu chybu pri False Positives, ak sa chyba pri False Negatives zníži.

1.2 Klasifikačné a regresné stromy (CART)

Jedným zo známych klasifikátorov sú klasifikačné a regresné stromy (*Classification And Regression Trees* - CART). S ich myšlienkou prišli v roku 1984 Breiman, Friedman, Olshen a Stone. Náhodné lesy sa skladajú práve z CART, preto je potrebné pochopiť, ako fungujú. Hoci z názvu tohto klasifikátora sa dá tušiť, že je aplikovateľný aj pre regresné problémy, kedy je vysvetľovaná premenná y spojitá, my sa budeme zaoberať iba klasifikačným problémom. Myšlienku CART algoritmu si ukážeme na krátkom príklade.

Predstavme si, že máme o klientoch banky informáciu, či zlyhali (neboli schopní splácať úver) alebo nie. Ďalej máme k dispozícii niekoľko charakteristík klienta - napr. výšku platu, vek, rodinný stav atď. Našou úlohou je na základe charakteristík - prediktorov ($\mathbf{X}$) predpovedať zlyhanie ($y \in \{0, 1\}$, „1" znamená zlyhanie). Metóda CART postupuje nasledovne - pre každý prediktor vyhodnotíme všetky možné delenia. Napríklad ak je prediktorom počet detí (ordinálna premenná) a

všetci klienti majú žiadne, 1, 2 alebo 3 deti, máme možnosť 3 delení - (0)-(1,2,3), (0,1)-(2,3), (0,1,2)-(3). Vo všeobecnosti, pre ordinálne alebo intervalové (spojité) premenné, ktoré nadobúdajú v trénovacej množine h rôznych hodnôt, máme $h-1$ rôznych delení. Ak je prediktorom napríklad národnosť (kategorická premenná) s 3 rôznymi hodnotami - Slovák, Čech a Maďar, môžeme deliť 3 spôsobmi (Slovák)-(Čech, Maďar), (Čech)-(Slovák, Maďar) a (Maďar)-(Slovák, Čech). Všeobecne, pre kategorické premenné s g rôznymi triedami je $2^{g-1}-1$ delení.

Zo všetkých možných delení vyberieme „najlepšie", podľa ktorého rozdelíme množinu všetkých klientov na dve podmnožiny (dcéry). Postup výberu „najlepšieho" delenia aplikujeme rekurzívne aj na tieto podmnožiny a takisto aj na ich dcéry a tak ďalej, dokiaľ je možné uskutočniť delenie, vzhľadom na zastavovacie kritériá, o ktorých si povieme nižšie. Ak sa niektorá množina už ďalej nedelí, nazývame ju terminálna.

Ak chceme pomocou stromu klasifikovať klienta, necháme ho „prejsť" stromom. Klientovi pridelíme triedu, ktorá je najpočetnejšia v terminálnej množine, do ktorej spadol. Napr. ak padne do množiny, kde je 90% zlyhaných a 10% nezlyhaných klientov, pridelíme mu triedu „zlyhal".

Tento postup sa nazýva vo všeobecnosti *rekurzívne binárne delenie* a využívajú ho aj iné klasifikátory stromovej štruktúry. Jeho algoritmus formulujeme s použitím váh pozorovaní $\mathbf{w} = (w_1, ..., w_n)$. Pre jednoduchosť uvažujme iba 0-1 váhy - každý *uzol* (množina pozorovaní) stromu je reprezentovaný vektorom váh pozorovaní, pozorovania s jednotkovými váhami patria do tohto uzla a pozorovania s nulovými váhami nie. Na začiatku budú mať všetky pozorovania jednotkové váhy. Algoritmus rekurzívneho binárneho delenia je potom nasledovný:

1. Pre pozorovania dané váhami $\mathbf{w}$ testujeme, či nie je splnené zastavovacie kritérium. Ak áno, algoritmus sa zastaví.

2. Pre $\mathbf{w}$ a pre každý prediktor X_j, $j = 1, ..., m$ urobíme všetky možné delenia hodnôt prediktora: $A \subset \mathcal{X}_j$ a $\mathcal{X}_j \setminus A$. Spomedzi nich vyberieme „najlepšie" delenie $A^* \subset \mathcal{X}_{j^*}$, čím rozdelíme $\mathcal{X}_{j^*}$ na dve disjunktné množiny (dcéry) A^* a $\mathcal{X}_{j^*} \setminus A^*$. Váhy pozorovaní $\mathbf{w}_{left}$ a $\mathbf{w}_{right}$ určia dve podmnožiny delenia, $w_{left,i} = w_i I(X_{j^*i} \in A^*)$ a $w_{right,i} = w_i I(X_{j^*i} \notin A^*)$ pre všetky $i = 1, ..., n$.
   $I(.)$ je indikačná funkcia, ktorá vráti jednotku, ak je podmienka pravdivá, inak vráti nulu.

3. Rekurzívne zopakujeme kroky 1. a 2. pre váhy $\mathbf{w}_{left}$ a $\mathbf{w}_{right}$.

Pre CART je zastavovacie kritérium dané tromi podmienkami: prvá - ak je počet pozorovaní v uzle menší ako vopred zvolené číslo, druhá - ak je v uzle zastúpená

len jedna trieda, a tretia - ak všetky prediktory nadobúdajú v uzle rovnakú hodnotu. Pre tieto prípady sa už uzol nedelí.

Zostáva nám povedať, aké je kritérium na určenie „najlepšieho" delenia. Cieľom delenia je rozdeliť množinu na dve podmnožiny tak, aby tieto boli čo najviac homogénne a súčasne čo najviac od seba odlišné (v zmysle vysvetľovanej premennej). Pomôžeme si teda pojmom nečistoty (*impurity*), ktorý vysvetlíme pre binárnu premennú.

1.2.1 Kritérium pre voľbu delenia

Nečistotou množiny M nazveme nezápornú funkciu pravdepodobnosti p, že pozorovania patriace do tejto množiny majú rovnakú triedu, napr. $y = 1$. Teda $p = P(y = 1|M)$ a nečistota M $I(M) = \phi(p)$. Od funkcie nečistoty ϕ očakávame, že ak bude množina M úplne homogénna (všetky pozorovania budú mať rovnakú triedu), $I(M)$ bude minimálna a ak budú v M obe triedy rovnako zastúpené, teda ak $p = 0.5$, $I(M)$ bude maximálna. Ďalej požadujeme, aby ϕ bola symetrická funkcia.

Týmto podmienkam vyhovujú okrem iných nasledujúce tri funkcie:

- Bayesova chyba: $\phi(p) = \min(p, 1 - p)$
- Entropia: $\phi(p) = -p\log(p) - (1 - p)\log(1 - p)$
- Gini index: $\phi(p) = p(1 - p)$

Všetky tri funkcie sú konkávne, majú minimum v $p = 0$ a $p = 1$ a maximum v $p = 0.5$. V praxi sa používa najmä Gini index, prípadne entropia.

Keď máme zadefinovanú nečistotu, môžeme sa pozrieť bližšie na kritérium pre voľbu „najlepšieho delenia". „Najlepším" delením na základe prediktora X_j bude delenie, ktoré najviac zníži nečistotu delenej množiny. Formálne si toto zníženie nečistoty definujeme ako:

$$\Delta I(A, M) = p(M_{left})I(M_{left}) - p(M_{right})I(M_{right})$$

Kde M_{left}, resp. M_{right} je ľavý, resp. pravý dcérsky uzol množiny M, vytvorené delením A, a $p(M_{left})$, resp. $p(M_{right})$ je pravdepodobnosť, že pozorovanie padne do ľavého, resp. pravého dcérskeho uzla. Tieto pravdepodobnosti sú odhadované početnosťami množín: $p(M_{left}) = |M_{left}|/|M|$ a $p(M_{right}) = |M_{right}|/|M|$.

Najlepšie delenie množiny M v zmysle zníženia nečistoty je potom:

$$A^* = \underset{A}{\operatorname{argmax}}\, \Delta I(A, M) \tag{1.1}$$

a X_{j^*} bude prediktor, ku ktorému toto delenie prislúcha. CART teda vyberie pri delení množiny M spomedzi všetkých možných delení A to, ktorého $\Delta I(A, M)$

je maximálne. Množiny M_{left} a M_{right} sú dané váhami $\mathbf{w}_{left}$ a $\mathbf{w}_{right}$ z 2. kroku algoritmu.

Ukázali sme, ako funguje algoritmus klasifikačného stromu pre binárnu vysvetľovanú premennú. Ak by mala vysvetľovaná premenná $J > 2$ tried, upraví sa iba funkcia nečistoty, napr. pre Gini index:

$$\phi(p_1, ..., p_J) = \sum_{i=1}^{J} p_i(1 - p_i) = 1 - \sum_{i=1}^{J} p_i^2$$

kde $p_i = P(y = C_i|M)$ sú pravdepodobnosti, že pozorovaniu v množine M prislúcha trieda C_i.

1.2.2 Zhrnutie CART

Výhodou CART je, že sú jednoduché na interpretáciu a pochopenie a dajú sa prehľadne graficky znázorniť. Dáta určené na vyvíjanie stromov nepotrebujú veľké úpravy a môžu obsahovať aj chýbajúce hodnoty. Stromy sa vedia jednoducho vysporiadať aj s kategorickými premennými. V porovnaní s parametrickými metódami dosahujú porovnateľnú presnosť. Medzi hlavné nevýhody stromov patrí nestabilita - ak urobíme malé zmeny v dátach, výsledky môžu byť veľmi odlišné. Preto je v teórii stromov CART ďalšou významnou kapitolou tzv. prerezávanie stromov (*pruning*). Keďže veľké, komplexné stromy preukazujú vysokú nestabilnosť, delenia, ktorých zníženie heterogenity nebolo výrazné, sa „odrežú", teda delená množina sa stane terminálnou. Zvýši sa stabilita, no za cenu zníženia presnosti. V algoritme náhodných lesov sa však stromy neprerezávajú, preto sa nebudeme tejto procedúre venovať.

Práve z dôvodu nestability je vhodné vytvoriť viacero stromov a nechať ich, aby sa na rozhodnutí o výslednej odhadovanej triede podieľali všetky. Na myšlienke agregovania viacerých stromov vznikli klasifikátory *bagging* (**b**ootstrap **agg**rega**ing**), *Adaboost* (**ada**ptive **boost**ing), *arcing* (**a**daptive **r**esampling and **c**combi**ning**) a *Random Forests* - náhodné lesy. My sa budeme venovať poslednému menovanému.

Ďalším a azda najväčším nedostatkom klasifikačných a regresných stromov je fakt, že pri výbere deliaceho prediktora vzniká odchýlka (*bias*). Upozorňujú na to napr. Hothorn et al. (2006a), ktorí uvádzajú, že CART uprednostňuje výber prediktorov, ktoré majú viac možných delení (napr. intervalové premenné) alebo tie, ktoré majú mnoho chýbajúcich hodnôt (*missing values*). Odchýlka je spôsobená tým, že pri výbere optimálneho delenia maximalizujeme cez všetky delenia naraz. Hothorn et al. (2006a) navrhujú zmeniť algoritmus tak, že najprv sa podľa dôle-

žitosti vyberie premenná X_{j^*} a následne vyberieme optimálne delenie spomedzi delení daných len touto premennou. Podrobnejšie sa tejto problematike budeme venovať neskôr.

1.3 Náhodné lesy

1.3.1 Algoritmus

Majme m prediktorov $X_1, ..., X_m$ a množinu trénovacích dát $\{(y_i, X_{1i}, ..., X_{mi}),\ i = 1, ..., n\}$ veľkosti n. Nech vysvetľovaná premenná y má J tried.

Algoritmus náhodných lesov funguje nasledovne:

1. Zvolíme parametre:
   - K - počet stromov, ktoré necháme vyrásť
   - m_{try} - počet prediktorov, ktoré sa podieľajú pri každom delení
   Pre $i = 1, ..., K$ opakujeme nasledovné kroky 2. až 5.

2. Z množiny trénovacích dát vyberieme náhodnú vzorku Θ_i veľkosti n s návratom - t.j. *bootstrap sampling*.

3. Z množiny prediktorov náhodne vyberieme bez návratu m_{try} prediktorov.

4. Skonštruujeme CART delenie náhodnej vzorky Θ_i pomocou m_{try} vybraných prediktorov.

5. Rekurzívne opakujeme kroky 3. a 4. pre každé ďalšie delenia v strome, až kým nevyrastie celý strom $h(\mathcal{X}, \Theta_i)$.

Týmto sme získali náhodný les zložený z K stromov. K je vo všeobecnosti dosť veľké, rádovo v stovkách až tisícoch. Ak chceme klasifikovať nové dáta, necháme ich prejsť všetkými stromami. Odhadovaná trieda bude najpočetnejšia trieda spomedzi všetkých výsledných tried stromov. Tento proces budeme v ďalšom texte nazývať *hlasovanie*.

Formálne si môžeme definovať náhodný les nasledovne:

Definícia 2. *Náhodný les je klasifikátor pozostávajúci z K klasifikátorov typu CART $\{h(x, \Theta_1), ..., h(x, \Theta_K)\}$, kde $\{\Theta_1, ..., \Theta_K\}$ sú nezávislé identicky rozdelené náhodné vektory. Výstupom náhodného lesa pre $x \in \mathcal{X}$ je trieda určená hlasovaním všetkých tých stromov $h(., \Theta_i)$, pre ktoré x nepatrilo do Θ_i.*

Z definície vyplýva, že ak pozorovanie x nepatrí do trénovacej množiny (napr. v prípade, ak máme nové dáta alebo množinu testovacích dát), na hlasovaní sa podieľajú všetky stromy.

1.3.2 Hraničná funkcia a chyba zo zovšeobecnenia

Predpokladajme, že máme trénovaciu množinu obsahujúcu prediktory a prislúchajúcu triedu vysvetľovanej premennej. Trénovacia množina je realizáciou náhodného výberu z populácie, konkrétnejšie realizáciou dvoch druhov náhodných premenných - množiny prediktorov a vysvetľovanej premennej.

Vezmime si teraz jeden konkrétny riadok z trénovacej množiny. Tento riadok budeme v ďalšom texte nazývať bod. Je samozrejmé, že hodnoty bodu budú rôzne pre rôzne realizácie trénovacej množiny. Teda bod je dvojica náhodných premenných, označíme ho $(\mathbf{X}, y)$. Množina prediktorov bodu je reprezentovaná náhodnou premennou $\mathbf{X}$ a trieda bodu je náhodná premenná y.

Predpokladajme ďalej, že máme súbor K klasifikátorov, $f_1(x), ..., f_K(x)$. Nateraz nie je dôležité, ako boli tieto klasifikátory skonštruované.

Hraničnú funkciu (*margin function*) tohto bodu definujeme ako

$$mg(\mathbf{X}, y) = \frac{1}{K}\sum_{k=1}^{K} I(f_k(\mathbf{X}) = y) - \max_{j \neq y} \frac{1}{K}\sum_{k=1}^{K} I(f_k(\mathbf{X}) = j), \tag{1.2}$$

kde $I(.)$ je identifikátor a j je nesprávna trieda.Teda hraničná funkcia je rozdiel priemerného počtu hlasov za správnu triedu y a priemerného počtu hlasov za nesprávnu triedu, ktorá mala najviac hlasov. Je mierou toho, ako spoľahlivá je klasifikácia. Ak je vysvetľovaná funkcia binárna, hraničná funkcia sa zjednoduší na tvar

$$mg(\mathbf{X}, y) = 2av_k I(f_k(\mathbf{X}) = y) - 1$$

Z definície hraničnej funkcie 1.2 môžeme vytvoriť vzťah pre chybu zo zovšeobecnenia (*generalization error*)

$$g = P_{\mathbf{X},y}(mg(\mathbf{X}, y) < 0) \tag{1.3}$$

kde $P_{\mathbf{X},y}(.)$ značí pravdepodobnosť nad priestorom reprezentovaným náhodnými premennými $\mathbf{X}, y$. Ak je hraničná funkcia nejakého bodu záporná, znamená to, že klasifikátor nesprávne určil triedu. Chyba zo zovšeobecnenia sa dá potom interpretovať ako pravdepodobnosť nad realizáciami bodov, že klasifikátor nesprávne určí triedu.

Konkrétne pre náhodné lesy, kde $f_k(x) = h_k(x, \Theta_k)$, dostávame hraničnú funkciu v tvare

$$mr(\mathbf{X}, y) = P_\Theta(h(\mathbf{X}, \Theta) = y) - \max_{j \neq y} P_\Theta(h(\mathbf{X}, \Theta) = j) \tag{1.4}$$

a pre prípad binárnej vysvetľovanej premennej

$$mr(\mathbf{X}, y) = 2P_\Theta(h(\mathbf{X}, \Theta) = y) - 1 \tag{1.5}$$

Breiman (2001) dokázal s využitím silného zákona veľkých čísel nasledujúcu vetu:

Veta 1. *S rastúcim počtom stromov v náhodnom lese pre postupnosť $\Theta_1, ...$ konverguje chyba zo zovšeobecnenia g k populačnej chybe zo zovšeobecnenia (population generalization error)*

$$P_{\mathbf{X},y}(mr(\mathbf{X}, y)) = P_{\mathbf{X},y}(P_\Theta(h(\mathbf{X}, \Theta) = y) - \max_{j \neq y} P_\Theta(h(\mathbf{X}, \Theta) = j) < 0). \tag{1.6}$$

Dôkaz vety je uvedený v článku (Breiman, 2001). Tu si treba uvedomiť, aké pravdepodobnosti sa vo vete používajú. Pre každý strom sa používajú bootstrapom náhodne vybrané dáta z trénovacej množiny, čomu zodpovedá pravdepodobnosť P_Θ. Trénovacia množina je realizácia náhodného výberu z náhodných premenných. Tomu zodpovedá pravdepodobnosť $P_{\mathbf{X},y}$.

Dôležitým dôsledkom vety je, že náhodné lesy nie sú náchylné na pretrénovanie (*overfitting*), ak pridávame ďalšie stromy.

1.3.3 Horná hranica chyby zo zovšeobecnenia

Hraničná funkcia 1.4 pre náhodné lesy berie trénovaciu množinu ako pevnú, teda pri jej výpočte používame iba jednu realizáciu bodu $(\mathbf{X}, y)$. Zrejme pre rôzne realizácie bodov budú aj hraničné funkcie týchto bodov rôzne. Preto je vhodné vziať strednú hodnotu hraničnej funkcie cez realizácie bodov. Podľa (Berk, 2008) je teda definovaná sila náhodného lesa $\{h(x, \Theta)\}$ ako:

$$s = E_{\mathbf{X},y} mr(\mathbf{X}, y). \tag{1.7}$$

Odhad sily teda môžeme vypočítať ako priemer hraničných funkcií cez náhodne vybrané trénovacie dáta. Čím vyššia je hodnota sily, tým lepšie.

Ak si vezmeme nejaký konkrétny bod, výsledné triedy jednotlivých stromov sa môžu líšiť. Je to spôsobené náhodnosťou pri bootstrapových výberoch z trénovacej množiny a takisto náhodnosťou pri výbere do množiny kandidátov pre

každé delenie. V ideálnom prípade by mali tieto zdroje náhodnosti zabezpečovať nezávislosť výstupov z každého stromu. Je však potrebné pozrieť sa na závislosť stromov bližšie.

Pre binárnu vysvetľovanú premennú je postup relatívne priamočiary. Náhodne vyberieme jeden bod z populácie. Klasifikujeme tento bod každým stromom a zaznamenáme 1, ak je klasifikácia správna alebo 0, ak nesprávna. Tento proces opakujeme pre ďalšie náhodne vybrané body. Na konci vypočítame korelácie medzi výsledkami jednotlivých stromov. Vhodnou mierou pre závislosť stromov je priemer z vypočítaných korelácií. Je vhodné, ak je korelácia čo najmenšia.

Breiman (2001) ukázal, že chyba zo zovšeobecnenia, ktorú sme definovali pomocou hraničnej funkcie, závisí od sily náhodného lesa a od závislosti medzi jednotlivými stromami, konkrétne dokázal, že horná hranica pre túto chybu je

$$g^* = \frac{\overline{\rho}(1 - s^2)}{s^2}, \tag{1.8}$$

kde $\overline{\rho}$ je priemerná korelácia medzi stromami a s je sila náhodného lesa definovaná vzťahom 1.7.

Chyba zo zovšeobecnenia nebude rásť s rastúcim počtom stromov. Je preto vhodné použiť veľa stromov, čím bude skutočná hodnota chyby presnejšie aproximovaná.

1.3.4 Out-of-bag odhady

Už sme spomenuli, že v algoritme náhodných lesov sa pre každý strom vytvára bootstrapom nová trénovacia množina. Z teórie pravdepodobnosti vieme, že pri bootstrape - výbere n pozorovaní z množiny veľkosti n s návratom, sa pravdepodobnosť, že nejaké konkrétne pozorovanie nebude vybrané, rovná e^{-1}. To znamená, že pri každom bootstrapovom výbere nám popri vybraných dátach zostane približne 37% nevybraných dát. Tieto pozorovania môžeme využiť ako testovaciu množinu dát a budeme ich nazývať „out-of-bag" pozorovania (skrátene OOB). Na základe tejto testovacej množiny môžeme vytvoriť odhady chyby zo zovšeobecnenia g, ako aj sily s a priemernej korelácie $\overline{\rho}$.

Ako budeme postupovať? Každý strom necháme klasifikovať jemu prislúchajúce OOB pozorovania. Na základe týchto klasifikácií určíme pre každé pozorovanie hlasovaním najpočetnejšiu triedu a tá bude tomuto pozorovaniu priradená. Inými slovami - na klasifikácii pozorovania sa budú podieľať len tie stromy, ktoré boli skonštruované bez použitia tohto pozorovania. Každé pozorovanie zahrnuté v pôvodnej trénovacej množine je klasifikované zhruba 0.37-násobkom počtu stromov v lese. Z takto klasifikovaných pozorovaní môžeme pripraviť maticu za-

triedenia (confusion matrix). Pre odhad chyby zo zovšeobecnenia si najskôr definujme $Q(\mathbf{X}, j)$ - podiel z počtu stromov, ktoré pozorovaniu $(\mathbf{X}, y)$ priradili triedu j:

$$Q(\mathbf{X}, j) = \frac{1}{|T_{\mathbf{X}}|} \sum_{k=1}^{K} I(h(\mathbf{X}, \Theta_k) = j; (\mathbf{X}, y) \in T_{\mathbf{X}}), \tag{1.9}$$

kde $T_{\mathbf{X}}$ je množina tých stromov, kde pozorovanie $(\mathbf{X}, y)$ nebolo vybrané do trénovacej vzorky stromu a $|T_{\mathbf{X}}|$ je počet týchto stromov.

$Q(\mathbf{X}, y)$ bude teda podiel počtu stromov, ktoré správne klasifikovali pozorovanie $(\mathbf{X}, y)$ a naopak $1 - Q(\mathbf{X}, y)$ podiel nesprávne klasifikujúcich stromov. Priemer $1 - Q(\mathbf{X}, y)$ cez všetkých n pozorovaní v trénovacej množine nám dá požadovaný odhad chyby zo zovšeobecnenia:

$$\hat{g}_{oob} = \frac{1}{n} \sum_{i=1}^{n} 1 - Q(\mathbf{X}_i, y_i) = 1 - \frac{1}{n} \sum_{i=1}^{n} Q(\mathbf{X}_i, y_i), \tag{1.10}$$

Odhad chyby je s dostatočne veľkým počtom stromov relatívne presný, hoci Bylander (2002) ukázal, že je vychýlený smerom nahor.

Silu náhodného lesa sme definovali ako strednú hodnotu hraničnej funkcie. Uvedomme si, že výrazy $P_{\Theta}(h(\mathbf{X}, \Theta) = y)$ a $P_{\Theta}(h(\mathbf{X}, \Theta) = j)$ v hraničnej funkcii môžeme odhadnúť pomocou $Q(\mathbf{X}, y)$ a $Q(\mathbf{X}, j)$. Strednú hodnotu v definícii sily nahradíme priemerom, čím dostaneme výraz pre odhad sily:

$$\hat{s}_{oob} = \frac{1}{n} \sum_{i=1}^{n} (Q(\mathbf{X}_i, y_i) - \max_{j \neq y_i} Q(\mathbf{X}_i, j)). \tag{1.11}$$

Odhad priemernej korelácie je nad rámec tejto práce.

1.3.5 Zahrnutie nákladov

V časti 1.1 sme spomenuli relatívne náklady. Jedná sa o situáciu, kedy sa snažíme znižovať jeden druh chyby (napr. False Negatives) aj za cenu zvýšenia druhej chyby (False Positives), pretože náklady na chybu False Negatives sú vyššie ako pri False Positives.

Pre prípad binárnej vysvetľovanej premennej sa ponúka na zahrnutie nákladov do modelu viacero možností:

1. **Apriórne rozdelenie** - používa sa najmä vtedy, ak predpokladáme, že rozdelenie tried vysvetľovanej premennej je iné ako v trénovacej množine. V binárnom prípade je tento prístup ekvivalentný zmene relatívnych nákladov.

2. **Rozdielne váhy stromov** - les je vytvorený bez zmeny, rozdielne sú váhy pri hlasovaní stromov - klasifikovanie pozorovania do menej bežnej kategórie budeme brať napr. ako dva hlasy pre klasifikovanie do bežnej kategórie. Inými slovami, výsledná kategória je váženým priemerom kategórií stromov.

3. **Cut-off hranica** - podobný prístup ako 2. Bežný postup pri hlasovaní stromov je vziať najpočetnejšiu triedu - teda v binárnom prípade nad 50%. Zmenou tejto hranice môžeme zohľadniť náklady. Napríklad pozorovanie budeme klasifikovať ako „1", ak podiel hlasov za túto kategóriu bude vyšší ako 30%.

4. **Stratifikovaný bootstrap** - trénovacie množiny pre každý strom vyberieme tak, že pozorovania s kategóriou, pre ktorú majú chyby vyššie relatívne náklady, budú mať vyššiu šancu dostať sa do výberu. Tento prístup je podobný zmene apriórneho rozdelenia.

Prvý a štvrtý prístup ovplyvní aj stavbu samotných stromov v lese, druhý a tretí iba upravuje hlasovanie stromov. Štvrtý prístup môže byť veľmi užitočný vtedy, ak je binárna vysvetľovaná premenná značne nevyvážená v kategóriách (napr. 95% jednej kategórie k 5% druhej). V takýchto prípadoch môže byť problém vytvoriť použiteľné stromy.

1.3.6 Významnosť prediktorov

Ak berieme ako klasifikátor jeden strom CART, je veľmi jednoduché určiť, ktoré prediktory sú pri klasifikácii významné, ktoré menej, a ktoré sa nepodieľajú vôbec. Pre náhodný les zložený z mnohých stromov to už však také ľahké nie je. No v praxi je často popri správnej klasifikácii nemenej dôležitá aj informácia o tom, ako významné sú jednotlivé prediktory. Doposiaľ sa nepodarilo vyriešiť, ako najlepšie určiť významnosť prediktorov v náhodných lesoch. V (Berk, 2008) sa však ponúkajú dva použiteľné prístupy:

1. **Princíp zníženia nečistoty.** Vždy, keď je daný prediktor použitý pri delení množiny v strome, zaznamená sa hodnota, o koľko sa znížila nečistota pri delení (napr. Gini index). Mierou významnosti prediktora pre daný strom je potom suma všetkých týchto hodnôt. Spriemerovaním týchto mier cez množinu všetkých stromov v náhodnom lese získavame mieru významnosti prediktora. Avšak, táto miera nezohľadňuje predpovedaciu schopnosť. Miera je počítaná na základe informácií z trénovacích množín stromov, teda nie je získaná z testovacej množiny (OOB).

2. **Princíp permutácie.** Jeho myšlienka stojí na tom, že ak náhodne permutujeme hodnoty zúčastneného prediktora, zrušíme tým vzťah medzi prediktorom a vysvetľovanou premennou. Ak zmeriame predikčnú chybu v strome pred a po permutácii, rozdiel môže byť použitý ako miera významnosti. Čím väčší rozdiel, tým viac je prediktor zúčastnený pri predikcii, a teda tým je významnejší. Takto odvodenú významnosť prediktorov budeme nazývať permutačná významnosť. Presný postup si opíšeme v samostatnej časti.

Permutačná významnosť

Ak klasifikujeme kategorickú premennú, algoritmus na získanie permutačnej významnosti prediktorov vyzerá nasledovne:

1. Pre každý strom vypočítame jeho predikčnú chybu ν_k, $k = 1, ..., K$ s použitím OOB dát daného stromu. Chyba sa ráta z matice zatriedenia ako bolo popísané vyššie - teda ako podiel nesprávne klasifikovaných pozorovaní.

2. Ak máme p prediktorov, opakujeme krok 1. pre každý strom p-krát, ale každý raz s náhodne premiešanými - permutovanými hodnotami iného prediktora. Označme predikčnú chybu k-teho stromu s permutovanými hodnotami j-teho prediktora ako ν_{kj}. Ak j-ty prediktor v k-tom strome nebol použitý pri žiadnom delení, ν_{kj} bude rovné pôvodnej chybe bez permutácie ν_k. Ak použitý bol, náhodnosť permutácie spôsobí odstránenie väzby medzi daným prediktorom a vysvetľovanou premennou, teda chyba ν_{kj} by mala byť väčšia ako ν_k.

3. Pre každý z p prediktorov vypočítame priemer cez všetkých K stromov. Permutačná významnosť j-teho prediktora je teda definovaná ako

$$I_j = \frac{1}{K} \sum_{k=1}^{K} (\nu_{kj} - \nu_k), \quad j = 1, ..., p. \tag{1.12}$$

Jedným z hlavných nedostatkov tejto miery významnosti je, že vždy je náhodne permutovaný iba jeden prediktor - toto môže byť problémom, ak prediktory nie sú nezávislé. Riešeniu tohto problému sa budeme venovať neskôr.

V mnohých aplikáciách je prvoradé, aby vytvorený model mal čo najlepšiu predpovedaciu schopnosť. V tomto prípade je vhodné hodnotiť významnosť prediktorov pomocou permutačnej významnosti.

1.3.7 Parciálne závislosti

V predchádzajúcej časti sme opísali nástroje na meranie významnosti prediktorov. Často je však vhodné poznať, ako vysvetľovaná premenná závisí od konkrétneho prediktora. Dôležité je to napríklad v modeloch kreditného skóringu, kde odhadujeme pravdepodobnosť, že klient stratí platobnú schopnosť. Je možné, že nejaký model preukáže závislosť vysvetľovanej premennej na prediktore, ktorá sa prieči ekonomickej interpretácii (napr. rizikovosť by stúpala s rastúcim vekom, s rastúcim počtom rokov v zamestnaní). Ak by nastala takáto situácia a prediktor by bol navyše významný, mali by sa hľadať príčiny takéhoto správania a prediktor by mal byť z modelu odstránený.

Friedman (2001) pre náhodné lesy navrhol konštrukciu tzv. grafov parciálnej závislosti (*partial dependence plots*). Vychádzajú z už vytvoreného náhodného lesa. Algoritmus uvedieme pre prípad binárnej vysvetľovanej premennej, ($y \in \{0, 1\}$).

Predpokladajme, že chceme skúmať závislosť od prediktora X_j, ktorý nadobúda v rôznych hodnôt $c_{j1}, ..., c_{jv}$.

1. Pre $i = 1, ..., v$ vykonáme kroky 2. a 3.

2. Zostrojíme špeciálnu množinu dát tak, že X_j bude nadobúdať iba hodnotu c_{ji}. Hodnoty ostatných prediktorov ponecháme nezmenené.

3. Pozorovania tejto množiny dát necháme klasifikovať už vytvoreným náhodným lesom. Označme $P(y_k = 1|X_{jk} = c_{ji})$ podiel stromov, ktoré klasifikovali k-te pozorovanie ako $y_k = 1$. Ďalej nech

$$P(y = 1|X_j = c_{ji}) = \frac{1}{n} \sum_{k=1}^{n} P(y_k = 1|X_{jk} = c_{ji}).$$

   Tento výraz nám teda udáva akýsi odhad pravdepodobnosti pridelenia triedy $y = 1$, ak je $X_j = c_{ji}$.

4. Nech g je zobrazenie: $\mathcal{X}_j \rightarrow \langle 0, 1\rangle$, $g(c_{ji}) = P(y = 1|X_j = c_{ji})$. Zostrojíme graf tejto funkcie.

V praxi sa navyše často používa na zobrazenie logit transformácia zobrazenia g. Funkcia $\mathrm{logit}(x) = \ln\left(\frac{x}{1-x}\right)$. Výsledný graf parciálnej závislosti pre prediktor X_j by bolo zobrazenie:

$$\tilde{g}(c_{ji}) = \ln\left(\frac{P(y = 1|X_j = c_{ji})}{1 - P(y = 1|X_j = c_{ji})}\right).$$

Podobne by sme mohli vytvoriť graf parciálnej závislosti vzhľadom na triedu $y = 0$. Bol by však iba zrkadlovým obratením grafu pre $y = 1$. Treba zdôrazniť, že odhadnuté pravdepodobnosti nie sú úplne presné, celkový graf však dáva informáciu, ako na zmenu prediktora reaguje vysvetľovaná premenná.

1.3.8 Matica blízkosti

Ďalšou z vedľajších produktov náhodných lesov je matica blízkosti (*proximity matrix*). Jedná sa o maticu, ktorá dáva informáciu o tom, ako veľmi sú si jednotlivé pozorovania blízke, podobné. Zostrojí sa podľa nasledujúceho algoritmu, pričom opäť vychádzame z už vytvoreného náhodného lesa zloženého z K stromov:

1. Nech matica M je nulová matica typu $n \times n$, kde n je počet všetkých pozorovaní v trénovacej množine.

   Pre $k = 1, ..., K$ opakujeme nasledovné kroky 2. a 3.

2. k-ty strom necháme klasifikovať všetky dáta (celú trénovaciu množinu, teda aj OOB dáta).

3. Ak sa i-te a j-te pozorovanie vyskytlo v tej istej terminálnej množine tohto stromu, zvýšime ij-ty a ji-ty prvok matice M o jednotku.

4. Nakoniec normalizujeme maticu M delením každého jej prvku počtom pozorovaní n.

Výsledná matica M je maticou blízkosti. Jej ij-ty prvok M_{ij} ukazuje podiel stromov, ktoré zaradili pozorovania i a j do tej istej terminálnej množiny. Čím vyšší je tento podiel, tým sú si viac „blízke".

Keďže v praxi býva počet pozorovaní n často pomerne veľké číslo (rádovo tisíce až desaťtisíce), môže byť z technického hľadiska náročné pracovať s touto maticou. Treba poznamenať, že matica je symetrická, teda stačí ukladať len jej spodnú alebo hornú trojuholníkovú maticu. Ak je n príliš veľké, môžu sa pamätať iba dvojice pozorovaní s blízkosťou vyššou ako nejaká vhodne zvolená hranica.

Hoci sa môže samotná matica zdať kvôli veľkosti nepoužiteľná pre praktické účely, nesie v sebe dôležitú informáciu, ktorú využívajú niektoré aplikácie. Spomenieme si tieto tri:

1. **Zhlukovanie pozorovaní (*clustering*)**. Maticu blízkosti môžeme brať ako mieru podobnosti jednotlivých pozorovaní. Technika mnohorozmerného škálovania (*multidimensional scaling*) nám môže pomôcť ukázať, či majú pozo-

rovania tendenciu zhlukovať sa v priestore definovanom prediktormi a nakoľko sa tieto zhluky (*clusters*) líšia v triedach, do ktorých patria pozorovania jednotlivých zhlukov.

2. **Zisťovanie outlierov**. Outliery sú pozorovania, ktoré v istom zmysle „vyčnievajú" nad ostatnými, sú iné ako väčšina. Algoritmus na zisťovanie outlierov je založený na myšlienke, že riadok, resp. stĺpec v matici blízkosti, ktorý prislúcha outlierovi, by mal obsahovať veľa nízkych hodnôt.

3. **Priradenie hodnôt nevyplneným poliam (*missing values*)**. Budeme sa mu venovať v samostatnej časti.

Priradenie hodnôt nevyplneným poliam

V náhodných lesoch sa používajú dva spôsoby, ako nahradiť chýbajúce hodnoty. Prvou a rýchlou metódou je nahradenie týchto hodnôt mediánom dostupných hodnôt pre kvantitatívne prediktory a modusom pre kategorické prediktory. Ak chýbajúcich hodnôt nie je veľa, táto metóda môže byť postačujúca.

Druhá metóda je zložitejšia a výpočtovo náročnejšia. Je založená na matici blízkosti. Chýbajúce polia získame nasledovne:

1. Odhadneme chýbajúce polia „nahrubo" prvou metódou.

2. Ak chýbajúce pole je z kvantitatívnej premennej, nahradíme ju váženým priemerom vyplnených hodnôt tejto premennej. Ako váhy sa berú hodnoty z matice blízkosti medzi týmto chýbajúcim pozorovaním a vyplnenými pozorovaniami s tým, že sa znormalizujú, aby ich súčet bol rovný 1.

3. Ak chýbajúce pole je z kategorickej premennej, nahradíme ju najčastejšie sa vyskytujúcou vyplnenou hodnotou s váženými frekvenciami, kde váhy sú opäť z matice blízkosti tak ako v kroku 2.

Často sa tieto kroky opakujú v niekoľkých iteráciách - vždy sa vypočíta nová matica blízkosti s najaktuálnejšími nahradenými hodnotami, ktoré boli pôvodne chýbajúce. (Berk, 2008) uvádza, že v praxi je postačujúcich štyri až šesť iterácií.

Breiman (2001) však upozorňuje, že použitím týchto nahradených hodnôt má odhad OOB chyby (1.10) tendenciu byť príliš optimistický.

1.3.9 Nastavenie parametrov v náhodnom lese

V algoritme náhodných lesov nie je príliš veľa parametrov. Medzi tie najdôležitejšie patria minimálna veľkosť množiny v strome, počet stromov a počet náhodne

vyberaných prediktorov.

Minimálna veľkosť množiny určuje spodnú hranicu pri delení množín v CART. Ak nejaká množina má menej ako je táto hranica, táto množina už bude terminálna a nebude sa deliť. V algoritme CART hrá tento parameter dôležitú úlohu - čím je menší, tým menšia odchýlka, ale tým väčšia variancia. V náhodných lesoch je situácia odlišná - našou snahou je vytvoriť stromy, ktoré majú čo najmenšiu odchýlku. Vysoká variancia stromov je tolerovaná, pretože výsledky sú spriemerované cez veľký počet stromov. Breiman ako defaultnú hodnotu tohto parametra navrhol 1 v prípade klasifikačného problému a 5 v prípade regresie. Ak však máme veľký počet slabých prediktorov, ktoré sú navyše navzájom korelované, je vhodnejšie konštruovať menšie stromy a teda nastaviť tento parameter vyššie.

Počet stromov by mal byť aspoň niekoľko sto a nie je potrebné, aby bol vyšší ako niekoľko tisíc. V praxi sa používa väčšinou 500 stromov. Výhodou náhodných lesov je, že stromy sa môžu pridávať k už existujúcemu lesu, teda algoritmus sa nemusí spúšťať od začiatku.

Počet prediktorov m_{try} určuje, koľko prediktorov je náhodne vybraných pri každom delení stromov. Tento parameter najviac ovplyvňuje celkové výsledky náhodných lesov. Prekvapivým môže byť fakt, že aj pri nízkom počte vybraných prediktorov sa OOB odhad chyby veľmi nelíši od prípadu vyššieho počtu prediktorov. S veľkým počtom stromov má totiž každý prediktor dostatočnú príležitosť zúčastniť sa delenia. Samozrejme, tento parameter by sa mal voliť v závislosti od celkového počtu prediktorov a takisto od toho, ako silné sú tieto prediktory. V (Berk, 2008) je odporúčané vziať ako defaultnú hodnotu druhú odmocninu z celkového počtu prediktorov (zaokrúhlenú).

Žiaden z týchto troch parametrov však nemá na chybu náhodného lesa taký vplyv ako určenie relatívnych nákladov k jednotlivým chybám klasifikácie. Treba však mať na pamäti, že náklady nie sú ladiaci parameter lesov - sú určené so zreteľom na povahu dát a problému.

1.3.10 Zhrnutie, vlastnosti náhodných lesov

V tejto podkapitole sme sa snažili objasniť, ako funguje algoritmus náhodných lesov, ako možno merať niektoré charakteristiky modelu i jednotlivých vstupov. Predstavili sme rôzne vedľajšie produkty algoritmu, ktoré sú pri príprave dát a vyvíjaní modelu veľmi užitočné - napr. nahrádzanie chýbajúcich hodnôt.

Keďže v tejto práci budeme používať vždy binárny klasifikačný problém, neuvádzali sme niektoré špecifiká náhodných lesov v prípade viacerých tried v kla-

sifikácii alebo v prípade regresie. Čitateľovi môžeme odporučiť knihu (Berk, 2008) alebo článok od autora náhodných lesov (Breiman, 2001).

Náhodné lesy sú považované za veľmi silný nástroj ako pre klasifikáciu, tak i pre regresiu. Ak je hlavným kritériom pri tvorbe modelu presnosť predpovedí, potom sú náhodné lesy tou správnou voľbou. Vo viacerých prácach sa ukázalo, že náhodné lesy boli spomedzi rôznych štatistických nástrojov vždy medzi najsilnejšími. Porovnateľne silným nástrojom je *Adaboost*.

Ďalšou výhodou je fakt, že náhodné lesy sa vedia vysporiadať aj s dátami, ktoré majú viac premenných ako pozorovaní, kedy klasické parametrické metódy (napr. logistická regresia) zlyhajú. Takisto výsledný model môže obsahovať obrovský počet premenných.

Azda najväčšou nevýhodou náhodných lesov je fakt, že model je typu „black box", teda je akousi čiernou skrinkou, ktorá nám po vložení vstupov dodá výstupy. Hoci výstup môže byť presný, často je potrebné nahliadnuť dovnútra a pochopiť, ktoré premenné hrajú významnú rolu, a ako vplývajú na výstup. Pomôcť nám v tom môžu miery významnosti premenných a grafy parciálnych závislostí, no predsa nás nemusia úplne uspokojiť, ako je to napr. pri logistickej regresii.

Ďalším nedostatkom je vychýlenie pri výbere deliaceho prediktora, ktoré sme spomenuli pri zhrnutí CART. Keďže náhodné lesy obsahujú stromy, toto vychýlenie sa zo stromov prenesie aj do lesov.

1.4 Podmienené náhodné lesy

Náhodné lesy teda vykazujú vychýlenie a výsledky získané touto metódou nie sú optimálne. Tento nedostatok sa však dá riešiť - pôvod vychýlenia je už v samotných CART stromoch a vo voľbe deliaceho prediktora. Preto je potrebné definovať nové kritérium pre voľbu prediktora pri delení stromov. Ak by sme odstránili vychýlenie pri voľbe prediktora, dovolilo by nám to vytvoriť nevychýlené stromy a potom by bol nevychýlený aj náhodný les. V nasledujúcich riadkoch si priblížime koncept podmienenej inferencie, ktorá bude základom pre kritérium voľby prediktora. Strom vytvorený pomocou tohto konceptu budeme nazývať *podmienený strom*. Ak v algoritme náhodných lesov namiesto CART stromov použijeme podmienené stromy, výsledkom budú tzv. *podmienené náhodné lesy*. Budeme postupovať podľa práce (Hothorn et al., 2006a) .

1.4.1 Rekurzívne binárne delenie

Aby sme mohli pokračovať, je potrebné definovať si nanovo problém delenia, tentoraz ale s pomocou hypotéz o závislosti medzi vysvetľovanou premennou a prediktormi.

Majme vysvetľovanú premennú y, ktorá je náhodnou premennou z priestoru $\mathcal{Y}$. Ďalej majme m-rozmerný vektor prediktorov $\mathbf{X} = (X_1, ..., X_m)$ vzatý z priestoru $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_m$. Predpokladáme, že podmienené rozdelenie $D(y|\mathbf{X})$ premennej y dané premennými $\mathbf{X}$ závisí od funkcie f týchto premenných:

$$D(y|\mathbf{X}) = D(y|X_1, ..., X_m) = D(y|f(X_1, ..., X_m)),$$

pričom sa obmedzíme na regresné vzťahy založené na delení, t.j. chceme dostať r disjunktných podmnožín $B_1, ..., B_r$, ktoré pokryjú celý priestor $\mathcal{X} = \bigcup_{k=1}^{r} B_k$. Toto rozdelenie je získané pomocou trénovacej množiny $\mathcal{L}_n$, náhodného výberu n nezávislých a rovnako rozdelených pozorovaní s prípadnými chýbajúcimi hodnotami X_{ji},

$$\mathcal{L}_n = \{(y_i, X_{1i}, ..., X_{mi}); i = 1, ..., n\}.$$

Rekurzívne binárne delenie pre túto množinu a jeho algoritmus formulujeme použitím váh pozorovaní $\mathbf{w} = (w_1, ..., w_n)$. Pre jednoduchosť uvažujme iba 0-1 váhy - každý uzol (množina pozorovaní) stromu je reprezentovaný vektorom váh pozorovaní, pozorovania s jednotkovými váhami patria do tohto uzla a pozorovania s nulovými váhami nie. Algoritmus rekurzívneho binárneho delenia je potom nasledovný:

1. Pre pozorovania dané váhami $\mathbf{w}$ testujeme globálnu nulovú hypotézu o nezávislosti medzi ktorýmkoľvek z m prediktorov a vysvetľovanou premennou. Ak táto hypotéza nie je zamietnutá, algoritmus sa zastaví. Inak vyberieme j^*-ty prediktor X_{j^*} s najsilnejším vzťahom k y.

2. Zvolíme množinu $A^* \subset \mathcal{X}_{j^*}$, čím rozdelíme $\mathcal{X}_{j^*}$ na dve disjunktné množiny A^* a $\mathcal{X}_{j^*} \setminus A^*$. Váhy pozorovaní $\mathbf{w}_{left}$ a $\mathbf{w}_{right}$ určia dve podmnožiny delenia, $w_{left,i} = w_i I(X_{j^*i} \in A^*)$ a $w_{right,i} = w_i I(X_{j^*i} \notin A^*)$ pre všetky $i = 1, ..., n$. $I(.)$ je indikačná funkcia, ktorá vráti jednotku, ak je podmienka pravdivá, inak vráti nulu.

3. Rekurzívne zopakujeme kroky 1. a 2. pre váhy $\mathbf{w}_{left}$ a $\mathbf{w}_{right}$.

Kľúčovým prvkom v algoritme je, ako sme už spomenuli, kritérium pre voľbu X_{j^*} v kroku 1. a takisto voľba delenia v kroku 2. Navyše je tu oproti všeobecnému algoritmu pri stromoch pridané kritérium zastavenia pomocou hypotézy

o nezávislosti. Algoritmus sa zastaví, ak hypotézu nemôžeme zamietnuť na hladine významnosti α, ktorá je vopred zvolená ako parameter. Algoritmus vytvára delenie $\{B_1, ..., B_r\}$ priestoru premenných $\mathcal{X}$, kde každá množina $B \in \{B_1, ..., B_r\}$ je pridružená k jednotlivým váham pozorovaní.

1.4.2 Podmienená inferencia

V 1. kroku algoritmu sme narazili na problém s nezávislosťou - je potrebné rozhodnúť, či je v niektorom z prediktorov obsiahnutá informácia o vysvetľovanej premennej. V každom uzle určeným váhami $\mathbf{w}$ je globálna nulová hypotéza formulovaná pomocou m čiastkových hypotéz $H_0^j : D(y|X_j) = D(Y)$, $H_0 = \bigcap_{j=1}^{m} H_0^j$. Ak táto hypotéza nie je zamietnutá na vopred zvolenej hladine α, rekurzia sa v príslušnom uzle zastaví. Ak globálna hypotéza je zamietnutá, meriame vzťah medzi y a každým z prediktorov $X_j, \quad j = 1, ..., m$ pomocou testovacej štatistiky alebo p hodnoty, ktorá určuje, ako veľmi je skutočnosť vzdialená od hypotézy H_0^j. Potrebujeme teda testovaciu štatistiku, pomocou ktorej budeme testovať tieto hypotézy.

Označme symbolom $S(\mathcal{L}_n, \mathbf{w})$ symetrickú grupu všetkých permutácií prvkov $(1, ..., n)$, ktoré majú váhu $w_i = 1$. Autori Hothorn et al. (2006a) navrhli merať vzťah medzi y a X_j pomocou lineárnej štatistiky v tvare

$$\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}) = \text{vec}\left(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(y_i, (y_1, ..., y_n))^\top\right) \in \mathbb{R}^{p_j q} \tag{1.13}$$

kde $g_j : \mathcal{X} \to \mathbb{R}^{p_j}$ je nenáhodná transformácia premennej X_j; $h : \mathcal{Y} \times \mathcal{Y}_n \to \mathbb{R}^q$ je funkcia vplyvu a závisí od vysvetľovanej premennej každého pozorovania v zmysle permutácií, ktoré spomenieme nižšie. Matica typu $p_j \times q$ je konvertovaná do vektoru dĺžky $p_j q$ pomocou operátora $\text{vec}(.)$, ktorý postupne berie stĺpce matice a vytvára z nich vektor.

Autori navrhli, ako vhodne voliť funkcie g_j a h v závislosti od typu premenných. Pre nominálnu vysvetľovanú premennú s J triedami $1, ..., J$ môže byť funkcia vplyvu v tvare

$$h(y_i, (y_1, ..., y_n)) = e_J(y_i), \tag{1.14}$$

kde $e_J(k)$ je jednotkový vektor dĺžky J s jednotkou na k-tom mieste.

Podobne pre nominálny prediktor X_j s triedami $1, ..., K$ bude funkcia g_j volená ako

$$g_j(X_{ji}) = e_K(X_{ji}) \tag{1.15}$$

a pre intervalové prediktory ako

$$g_j(X_{ji}) = X_{ji}. \tag{1.16}$$

Rozdelenie $\mathbf{T}_j(\mathcal{L}_n, \mathbf{w})$ za platnosti hypotézy H_0^j závisí od združeného rozdelenia $D(y, X_j)$, ktoré je v praxi skoro vždy neznáme. Za platnosti H_0^j však môžeme fixovať prediktor X_j a za tejto podmienky odvodiť podmienenú strednú hodnotu μ_j a kovariančnú maticu Σ_j štatistiky $\mathbf{T}_j(\mathcal{L}_n, \mathbf{w})$ pomocou všetkých možných permutácií vysvetľujúcej premennej. Tento princíp nás vedie k procedúre známej ako permutačný test. Odhad pre podmienenú strednú hodnotu $\mu_j \in \mathbb{R}^{p_j q}$ má tvar:

$$\mu_j = \mathbf{E}(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w})|S(\mathcal{L}_n, \mathbf{w})) = \text{vec}\left(\left(\sum_{i=1}^{n} w_i g_j(X_{ji})\right)\mathbf{E}(h|S(\mathcal{L}_n, \mathbf{w}))^\top\right) \tag{1.17}$$

a pre podmienenú kovariančnú hodnotu $\Sigma_j \in \mathbb{R}^{p_j q \times p_j q}$:

$$\begin{aligned}\Sigma_j &= \mathbf{Var}(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w})|S(\mathcal{L}_n, \mathbf{w})) \\ &= \frac{\mathbf{w}.}{\mathbf{w}. - 1}\mathbf{Var}(h|S(\mathcal{L}_n, \mathbf{w})) \otimes \left(\sum_i w_i g_j(X_{ji}) \otimes w_i g_j(X_{ji})^\top\right) \\ &- \frac{1}{\mathbf{w}. - 1}\mathbf{Var}(h|S(\mathcal{L}_n, \mathbf{w})) \otimes \left(\sum_i w_i g_j(X_{ji})\right) \otimes \left(\sum_i w_i g_j(X_{ji})\right)^\top\end{aligned} \tag{1.18}$$

kde $\mathbf{w}. = \sum_{i=1}^{n} w_i$ je suma váh pozorovaní, podmienená stredná hodnota funkcie vplyvu má tvar

$$\mathbf{E}(h|S(\mathcal{L}_n, \mathbf{w})) = \mathbf{w}.^{-1}\sum_i w_i h(y_i, (y_1, ..., y_n)) \in \mathbb{R}^q,$$

kovariančná matica je v tvare

$$\begin{aligned}\mathbf{Var}(h|S(\mathcal{L}_n, \mathbf{w})) = \mathbf{w}.^{-1}\sum_i w_i &(h(y_i, (y_1, ..., y_n)) - \mathbf{E}(h|S(\mathcal{L}_n, \mathbf{w}))) \\ &(h(y_i, (y_1, ..., y_n)) - \mathbf{E}(h|S(\mathcal{L}_n, \mathbf{w})))^\top\end{aligned}$$

a kde symbol $\otimes$ označuje Kroneckerov súčin.

Ak máme podmienenú strednú hodnotu a kovariančnú maticu štatistiky $\mathbf{T} \in \mathbb{R}^{pq}$, môžeme ju štandardizovať. Pre testovanie je vhodné mať jednorozmernú testovaciu štatistiku c. Obvyklou voľbou je jeden z nasledujúcich dvoch spôsobov:

$$c_{max}(\mathbf{T}, \mu, \Sigma) = \max_{k=1,...,pq} \frac{|(\mathbf{T} - \mu)_k|}{\sqrt{(\Sigma)_{kk}}} \tag{1.19}$$

$$c_{quad}(\mathbf{T}, \mu, \Sigma) = (\mathbf{T} - \mu)\Sigma^+(\mathbf{T} - \mu)^\top \tag{1.20}$$

kde Σ^+ je Mooreova-Penroseova pseudoinverzia matice Σ.
Štatistiky $c(\mathbf{T}_j, \mu_j, \Sigma_j)$, $j = 1, ..., m$ môžeme porovnať nevychýleným spôsobom len vtedy, ak sú merané v rovnakej škále, t.j. $p_j = p$, $j = 1, ..., m$, čo však vo všeobecnosti nie je splnené. Preto je potrebné porovnávať p hodnoty pre podmienené

rozdelenie týchto testovacích štatistík $c(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}), \mu_j, \Sigma_j)$. V 1. kroku algoritmu teda vyberieme prediktor, ktorého štatistika bude mať minimálnu p hodnotu, formálne X_{j^*}, $j^* = \text{argmin}_{j=1,...,m} P_j$, kde

$$P_j = P_{H_0^j}(c(\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}), \mu_j, \Sigma_j) \geq c(\mathbf{t}_j, \mu_j, \Sigma_j) | \mathcal{S}_j(\mathcal{L}_n, \mathbf{w}))$$

označuje p hodnotu podmieneného testu pre hypotézu H_0^j. Podmienené rozdelenie $P(c(\mathbf{T}_j, \mu_j, \Sigma_j) \leq z | \mathcal{S}_j(\mathcal{L}_n, \mathbf{w}))$ je podiel počtu permutácií $\sigma \in \mathcal{S}_j(\mathcal{L}_n, \mathbf{w})$, kedy štatistika neprekračuje z, k počtu všetkých permutácií. Pre niektoré špeciálne formy mnohorozmernej štatistiky $\mathbf{T}$ je vhodné použiť toto podmienené rozdelenie, najmä pre trénovacie množiny s menším počtom pozorovaní. Autori Hothorn et al. (2006b) však poukazujú na ďalšie možnosti určovania podmieneného rozdelenia a jeho p hodnoty. Využívajú pri tom fakt, že štatistika $\mathbf{T}$ sa dá aproximovať normálnym rozdelením s podmienenou strednou hodnotou μ a kovariančnou maticou Σ, ak $n \to \infty$. Potom asymptotické podmienené rozdelenie štatistiky c_{max} je normálne a asymptotické podmienené rozdelenie štatistiky c_{quad} má χ^2 rozdelenie s počtom stupňov voľnosti rovným hodnosti matice Σ.

Doposiaľ sme však nevyriešili prvú časť 1. kroku algoritmu - zastavovacie kritérium, ktoré je postavené na globálnej hypotéze H_0. Autori navrhujú použiť štatistiku, kde sú agregované všetky transformačné funkcie g_j:

$$\mathbf{T}_j(\mathcal{L}_n, \mathbf{w}) = \text{vec}\left(\sum_{i=1}^{n} w_i (g_1(X_{1i})^\top, ..., g_m(X_{mi})^\top)^\top h(y_i, (y_1, ..., y_n))^\top\right).$$

Iným, univerzálnejším prístupom je testovať hypotézu na základe p hodnôt P_j parciálnych hypotéz H_0^j, $j = 1, ..., m$. V tom prípade sa môže použiť Bonferroniho úprava p hodnôt. Hypotéza H_0 je zamietnutá, ak minimum z upravených p hodnôt je menšie ako α. Parameter α môže byť teda interpretovaný ako parameter určujúci veľkosť (počet uzlov) stromu.

Kritérium delenia

Ak sme zamietli hypotézu H_0 a vybrali prediktor X_{j^*}, môžeme pokračovať v algoritme krokom 2., kde je potrebné zvoliť delenie na množiny A^* a $\mathcal{X}_{j^*} \setminus A^*$. Hoci je možné použiť niektoré kritérium spomenuté pri CART, autori uvádzajú metódu vychádzajúcu z testovacej štatistiky $\mathbf{T}$ danej vzťahom (1.13). Kvalita delenia bude meraná pre všetky možné podmnožiny A množiny $\mathcal{X}_{j^*}$ prostredníctvom štatistiky

$$\mathbf{T}_j^A(\mathcal{L}_n, \mathbf{w}) = \text{vec}\left(\sum_{i=1}^{n} w_i I(X_{j^*i} \in A) h(y_i, (y_1, ..., y_n))^\top\right) \in \mathbb{R}^q. \tag{1.21}$$

Táto štatistika meria rozdielnosť medzi vzorkami $\{y_i|w_i > 0 \wedge X_{ji} \in A; i = 1, ..., n\}$ a $\{y_i|w_i > 0 \wedge X_{ji} \notin A; i = 1, ..., n\}$. Podmienená stredná hodnota μ_{j*}^A a kovariančná matica Σ_{j*}^A môžu byť vypočítané pomocou vzťahov (1.17) a (1.19). Najlepšie delenie A^* je potom dané maximalizáciou štatistiky c:

$$A^* = \underset{A}{\operatorname{argmax}}\, c(\mathbf{t}_{j*}^A, \mu_{j*}^A, \Sigma_{j*}^A) \tag{1.22}$$

V tomto prípade už nie je potrebné počítať podmienené rozdelenie tejto štatistiky. Aby sme predišli deleniam, kde jedna z podmnožín má veľmi málo pozorovaní alebo kde samotná delená množina je veľmi malá, je vhodné určiť minimálny počet pozorovaní pre podmnožiny i pre delenú množinu.

Chýbajúce hodnoty a pomocné delenia

Ak pozorovanie X_{ji} v premennej X_j chýba, položíme váhu $w_i = 0$ pri výpočte $\mathbf{T}_j(\mathcal{L}_n, \mathbf{w})$ vo vzťahu (1.13) a ak je X_j vybraná ako deliaca premenná, taktiež vo vzťahu (1.21) pre $\mathbf{T}_j^A(\mathcal{L}_n, \mathbf{w})$. Ak máme vytvorenú množinu A^*, použije sa „pomocné delenie" s cieľom nájsť delenie so zhruba rovnakým rozdelením pozorovaní ako originálne delenie cez množinu A^*. Pomocné delenie je urobené tak, že pôvodnú vysvetľovanú premennú nahradíme binárnou premennou $I(X_{ji} \in A^*)$ a delíme chýbajúce pozorovania algoritmom opísaným vyššie.

1.5 Podmienená významnosť prediktorov

V časti 1.3.6 sme si predstavili dve miery výnamnosti prediktorov. Prvú, založenú na princípe zníženia nečistoty (napr. Gini významnosť), a druhú, nazvanú aj permutačnú významnosť. Avšak Strobl et al. (2008) ukázali, že permutačná významnosť je v náhodných lesoch vychýlená a to tak, že prediktory s veľkým počtom kategórií a intervalové prediktory majú vyššiu významnosť. Ukázali sme, že ak namiesto náhodných lesov vytvoríme podmienené náhodné lesy, tento problém odstránime. No Strobl et al. (2008) navyše pomocou simulácií ukázali, že permutačná významnosť je nepresná, ak sa v trénovacej množine nachádzajú korelované prediktory. Odstránenie tohto nového druhu vychýlenia navrhli riešiť pomocou tzv. *podmienenej významnosti*.

Vychádzajme opäť z globálnej nulovej hypotézy H_0 o nezávislosti medzi vysvetľovanou premennou a ktorýmkoľvek z m prediktorov. Ak táto hypotéza platí,

permutácia hodnôt vysvetľovanej premennej y nemá vplyv ani na marginálne rozdelenie y, ani na združené rozdelenie y a prediktorov $X_1, ..., X_m$, pretože

$$D(y, X_1, ..., X_m) \overset{H_0}{=} D(y)D(X_1, ..., X_m).$$

Ak však nulová hypotéza neplatí, tá istá permutácia by viedla k odchýlke v združenom rozdelení alebo nejakej z neho odvodenej štatistike. Preto takáto odchýlka môže slúžiť ako indikátor, že dáta nemajú nezávislú štruktúru danú nulovou hypotézou.

Autori Strobl et al. (2008) uvádzajú, že permutačná významnosť, kde je jeden prediktor X_j permutovaný a y ako aj ostatné prediktory $Z = X_1, ..., X_{j-1}, X_{j+1}, ..., X_m$ zostavajú nezmenené, prislúcha nulovej hypotéze, že X_j je nezávislé od y aj od Z:

$$H_0 : X_j \perp y \wedge X_j \perp Z. \tag{1.23}$$

Za platnosti tejto hypotézy platí pre združené rozdelenie

$$D(y, X_j, Z) \overset{H_0}{=} D(y, Z)D(X_j). \tag{1.24}$$

Ak po permutácii združené rozdelenie alebo z neho odvodená štatistika bude mať odchýlku, môže byť spôsobená dvomi príčinami: buď porušením nezávislosti $X_j \perp y$ alebo porušením nezávislosti $X_j \perp Z$. Avšak cieľom určenia významnosti je sledovať iba porušenie $X_j \perp y$. Tu sa dostávame k odpovedi na otázku, prečo korelované prediktory majú umelo vyššiu významnosť. Permutovaním porušíme skôr nezávislosť medzi korelovanými prediktormi ako nezávislosť medzi prediktorom a vysvetľovanou premennou y.

Autori preto navrhli novú schému významnosti, tzv. *podmienenú permutačnú významnosť*, kde prediktor X_j je permutovaný iba vo vnútri skupín pozorovaní, kde $Z = z$. Táto permutačná schéma zodpovedá nulovej hypotéze

$$H_0 : (X_j \perp y)|Z \tag{1.25}$$

a za jej platnosti platia pre podmienené združené rozdelenie nasledujúce vzťahy:

$$\begin{aligned} D(y, X_j|Z) &\overset{H_0}{=} D(y|Z)D(X_j|Z), \\ D(y|X_j, Z) &\overset{H_0}{=} D(y|Z). \end{aligned}$$

Ak sú X_j a Z nezávislé, obe permutačné schémy dajú rovnaký výsledok. Avšak ak sú korelované, pôvodná permutačná schéma bude viesť k zvýšeniu významnosti týchto korelovaných prediktorov, čo je dôsledkom odchýlky z hypotézy o nezávislosti medzi X_j a Z.

Vynára sa však otázka, ako voliť skupiny, vo vnútri ktorých bude prediktor permutovaný. Inými slovami, ako určiť *mriežku*, ktorá nám rozdelí pozorovania na tieto skupiny. Rozdelením na $Z = z$ by sme pre spojité premenné alebo pre veľký počet premenných dostali veľké množstvo malých skupín. Autori navrhli vytvoriť mriežku na základe delení v príslušnom strome. Výhodou je, že toto delenie je už vytvorené pre každý strom.

Algoritmus pre podmienenú permutačnú významnosť prediktora X_j je potom nasledovný:

1. Pre každý strom vypočítame jeho predikčnú chybu ν_k, $k \in \{1, ..., K\}$, kde K je počet stromov, s použitím OOB dát daného stromu, ako podiel nesprávne klasifikovaných pozorovaní.

2. Pre všetky premenné $Z \subset \{X_1, ..., X_{j-1}, X_{j+1}, ..., X_m\}$, ktoré majú byť podmienené, a pre k-ty strom vyberieme deliace body tohto stromu a pomocou nich vytvoríme mriežku - delenie OOB dát na disjunktné podmnožiny.

3. Vo vnútri týchto podmnožín náhodne permutujeme premennú X_j a vypočítame predikčnú chybu po permutácii ν_{kj}. Rozdiel $\nu_{kj} - \nu_j$ je významnosť X_j pre k-ty strom.

4. Opakujeme kroky 2. a 3. pre všetkých K stromov. Podmienená permutačná významnosť prediktora X_j je priemer cez všetky stromy:

$$PI_j = \frac{1}{K} \sum_{k=1}^{K} (\nu_{kj} - \nu_k).$$

Premenné Z, ktoré majú byť podmienené, by mali zahŕňať všetky premenné korelované s X_j - vezmeme všetky tie premenné, ktorých korelácia s X_j spĺňa podmienku, že $1 - p$ hodnota je väčšia ako vopred zvolená hranica. Autori práce (Hothorn et al., 2006a) navrhujú zvoliť hranicu ako 0.2. Menšia hodnota hranice by mala efekt iba na silno korelované premenné. Na určenie p hodnoty sa používa ten istý permutačný test ako v 1. kroku algoritmu podmienených náhodných lesov.

Kapitola 2

Kreditný skóring

Kreditný skóring (credit scoring, risk scoring) môžeme definovať ako súbor rozhodovacích modelov a ich podkladových techník, ktoré banke pomáhajú *a)* pri rozhodovaní, či žiadateľovi poskytnúť úver - ak áno, za akých podmienok a prípadne aké ďalšie stratége voliť; *b)* alebo ako zvoliť úverové obmedzenia pre už existujúcich klientov a kam nasmerovať marketingové kampane. Vo všeobecnosti tieto rozhodovacie modely odhadujú úroveň rizika klienta alebo žiadateľa o úver. Teda neposkytujú 0-1 odhad, identifikáciu, či klient bude „dobrý" alebo „zlý", ale udávajú nám pravdepodobnosť, že sa klient stane „zlým", často nazývanú aj pravdepodobnosť zlyhania (probability of default). Skóre je potom ľubovoľná lineárna transformácia tejto pravdepodobnosti. Používa sa len kvôli lepšej interpretácii - človeku sa lepšie pozerá na prirodzené čísla (rádovo v stovkách) ako na pravdepodobnosť - teda desatinné číslo medzi 0 a 1.

Okrem odhadu úrovne rizika má kreditný skóring využitie aj v ďalších oblastiach:

- zefektívnenie procesu schvaľovania - vysoko rizikoví žiadatelia sú pridelení starším a skúsenejším pracovníkom na pobočke, zatiaľ čo menej rizikoví sú pridelení juniorom s menšími skúsenosťami. Takisto je možné zautomatizovať proces schvaľovania pre málo rizikových klientov, čím sa ušetria náklady i čas,

- vyhodnocovanie kvality portfólia pre potreby reportov,

- nastavenie minimálnych kapitálových požiadaviek - podľa dokumentu Basel II (2006), o ktorom si povieme v nasledujúcej časti, sa skóring využíva pri výpočte rizikovo vážených aktív.

2.1 Bazilejská dohoda

V 70. rokoch minulého storočia začali veľké banky pôsobiť nadnárodne – banky rozširovali sféru svojho pôsobenia za hranice svojich domovských krajín, no dohľad nad bankovým trhom bol iba na národnej úrovni. Preto bol v roku 1974 založený Bazilejský výbor pre bankový dohľad (Basel Comittee on Banking Supervision, ďalej BCBS). Je jedným z výborov Banky pre medzinárodné zúčtovanie (Bank for International Settlement). BCBS vznikol na podnet guvernérov centrálnych bánk 10 štátov, tzv. G10 a jeho cieľom bolo a je podporovať spoluprácu centrálnych bánk týchto krajín v oblasti bankového dohľadu. Na tento účel vyvíja prístupy, metódy a pravidlá obozretného podnikania komerčných bánk. V súčasnosti má BCBS 27 členských štátov (Slovensko medzi ne nepatrí), ktorých zástupcovia sa stretávajú štyrikrát ročne. Výbor udržiava vzťahy s centrálnymi bankami štátov, ktoré nie sú jeho členmi, za účelom šírenia štandardov bankového dohľadu. Robí tak aj prostredníctvom vydávania dokumentov, spomedzi ktorých najdôležitejšie sú *The Basel Capital Accord* (Basel I, 1988) a *The New Basel Capital Accord* (Basel II, 2006).

Basel I sa považuje za prvý medzinárodný dokument zaoberajúci sa meraním finančných rizík. Táto dohoda hovorí o minimálnych kapitálových požiadavkách banky. Aktíva sú rozdelené do 5 tried s rôznymi rizikovými váhami. Banky v štátoch, ktoré sa zaviazali plniť tieto požiadavky, sú povinné držať kapitál vo výške minimálne 8% rizikovo vážených aktív (*risk weighted assets* - RWA). Cieľom dokumentu bolo posilniť stabilitu medzinárodného bankového systému, zabrániť bankám vystavovať sa nadmerným úverovým rizikám a motivovať ich k držaniu likvidných a nízko rizikových aktív. Hoci dodržiavanie tejto dohody bolo povinné iba pre členské štáty BCBS, postupne sa pridávali aj desiatky ďalších krajín. Postupne sa však ukázali viaceré nedostatky spočívajúce najmä v hrubom a neobjektívnom priraďovaní rizikových váh.

Vzhľadom na nové trendy finančných trhov a prístupov v riadení rizík začal BCBS v roku 1999 rozsiahly medzinárodný konzultačný proces revízií dokumentu Basel I, ktorý vyústil do vydania novej dohody - Basel II. Popri kreditnom a trhovom riziku boli zavedené kapitálové požiadavky pre nový druh rizika – operačné riziko. Dokument tiež priniesol pravidlá pre sofistikovanejšie a presnejšie meranie kreditného rizika. Basel II je teda na rozdiel od jeho predchodcu rozsiahla, komplexná a metodologicky náročná koncepcia obozretného podnikania bánk, a preto bola a je výzvou predovšetkým pre banky a regulátorov.

Z obsahového hľadiska je nová dohoda o kapitáli postavená na troch pilieroch.

Pilier 1 obsahuje informácie o minimálnych kapitálových požiadavkach na kry-

tie kreditného, trhového a operačného rizika. Kreditné riziko môže byť merané nasledovnými prístupmi:

- štandardizovaný - banky požívajú na určenie rizikových tried ratingy z externej ratingovej agentúry,

- IRB (*Internaly Ratings-Based*) - prístup založený na internom ratingu. Banky, ktorým regulátor (centrálna banka) na základe splnenia určitých podmienok schválil IRB prístup, sa môžu spoľahnúť na vlastné interné odhady rizikových komponentov, vstupujúcich do výpočtu kapitálových požiadaviek pre daný obchod. Medzi rizikové komponenty patrí pravdepodobnosť zlyhania (*probability of default* - PD), strata v prípade zlyhania (*loss given default* - LGD), expozícia v momente zlyhania (exposure at default - EAD) a efektívna splatnosť (*effective maturity* - M). V niektorých prípadoch môže regulátor požadovať od banky, aby používala namiesto interného odhadu niektorého parametra regulátorom danú hodnotu. Preto sa IRB prístup delí na *Foundation* (základný) IRB prístup, kedy banka používa iba interný odhad PD, a *Advanced* (pokročilý) IRB prístup, kedy môže použiť svoje odhady všetkých komponentov.

V pilieri 2 sú uvedené pravidlá a postupy národných regulátorov pri monitorovaní a hodnotení kapitálovej primeranosti bánk, splnení podmienok používania jednotlivých metód stanovenia minimálnej kapitálovej požiadavky, ako aj hodnotení celkového systému riadenia rizík bánk. Pilier 3 obsahuje informácie o trhovej disciplíne a požiadavky pri zverejňovaní informácií bankami. Cieľom je prehĺbiť trhovú disciplínu tým, že banky budú o sebe zverejňovať viac informácií.

2.2 Druhy skóringu

Spomenuli sme, že nástroje kreditného skóringu sa používajú pri ohodnocovaní rizika jednak žiadateľov o úver (aplikačný skóring), ako aj klientov, ktorí už majú otvorený úverový produkt (behaviorálny skóring).

Aplikačný skóring nám pomáha rozlíšiť bezrizikových žiadateľov o úverový produkt od rizikových žiadateľov a na základe toho zvoliť niektoré z nasledujúcich stratégií:

- zamietnuť žiadosť o produkt vysoko rizikovým žiadateľom,

- zvoliť rizikovú maržu, ktorá je časťou úrokovej miery - menej rizikoví žiadatelia budú mať nižšiu úrokovú mieru,

- viac rizikovým žiadateľom prideliť nižší úverový limit (týka sa kreditných kariet alebo kontokorentov),
- žiadať od rizikových klientov vyššiu akontáciu (pri splátkových produktoch) alebo vyššiu mieru ručenia (napr. nehnuteľnosťou pri hypotekárnych produktoch),
- žiadať od viac rizikového klienta ďalšie informácie alebo potvrdenia - napr. overenie výšky mzdy.

Behaviorálny skóring je použitý pri ohodnocovaní rizika už existujúcich klientov. Môže sa použiť napríklad v nasledujúcich oblastiach:

- ponúknuť menej rizikovým klientom ďalšie výhodné produkty - banka osloví „dobrých" klientov, čo môže viesť k menej rizikovému portfóliu ponúkaného produktu,
- zvýšiť menej rizikovým klientom ich úverový limit,
- posúdiť, či predĺžiť platnosť expirovanej kreditnej karty,
- veľmi rizikovým klientom venovať zvýšenú pozornosť pre prípad potenciálneho zlyhania.

Okrem týchto dvoch najpoužívanejších existuje ešte viacero typov: skóring odhadujúci pravdepodobnosť bankrotu, podvodu alebo skóring odhadujúci pravdepodobnosť, že banka vymáhaním od klienta, ktorý zlyhal, získa späť peniaze. Pre účely marketingu to môže byť skóringový model identifikujúci klientov, u ktorých je pravdepodobné, že budú odpovedať na marketingovú kampaň.

2.3 Vývoj skórovacieho modelu

Proces vývoja skórovacieho modelu by mal vznikať za spolupráce medzi informačnými technológiami, dátovým analytikom a operačným tímom. Spolupráca by mala zaistiť konzistenciu s obchodnými zásadami a taktiež umožní výmenu skúseností a vedomostí počas vývoja. Skúsenosti hovoria, že vývoj s nedostatkom komunikácie môže viesť k problémom ako napríklad zahrnutie premennej, ktorá sa medzičasom prestala zbierať alebo je právne podozrivá, alebo navrhnutie stratégie, ktorá je v praxi nepoužiteľná.

Na vývoji by sa mali podľa (Siddiqi, 2006) podieľať nasledujúci ľudia:

- **špecialista pre skóring** - človek (alebo viacerí ľudia), ktorý vyvíja skóringový model. Mal by mať znalosti v oblasti data miningu a štatistických analýz, mal by dobre poznať dátový sklad spoločnosti a rozumieť štatistickým princípom použitým v modeli. Mal by mať skúsenosti s implementáciou a používaním modelov v oblasti rizika. Špecialista ručí za to, že dáta sú zbierané podľa špecifikácie a že model je štatisticky správny.
- **manažér portfólia** - zodpovedný za riadenie kvality portfólia a použitie skórovacích modelov. Mal by byť odborníkom vo vývoji a implementácii stratégií v oblasti rizík s použitím skóringových modelov a dobre poznať politiku riadenia rizík spoločnosti. Dozerá na to, aby bol skórovací model vyvinutý správne z obchodného hľadiska.
- **produktový manažér** - môže ponúknuť hlbší pohľad na rizikový profil klientov v jednotlivých produktoch. Podieľajú sa aj na tvorbe formulára žiadosti, kde sa zbierajú nové dáta.
- Okrem týchto ľudí je potrebný **IT manažér** zodpovedný za softvérové produkty v spoločnosti a často aj za dátový sklad, **projektový manažér**, ktorý dohliada na chod celého procesu, **prevádzkový manažér** a **právnik**.

Proces vývoja skóringového modelu má nasledujúce štádiá:

1. **plánovanie**,
2. **príprava dát a určenie parametrov projektu**,
3. **vytvorenie vývojovej vzorky**,
4. **vyvíjanie modelu**,
5. **tvorba manažérskych reportov**,
6. **implementácia modelu**,
7. **monitoring**.

O každej fáze vývoja si povieme v jednotlivých sekciách.

2.4 Plánovanie

Táto fáza zahŕňa vytvorenie obchodného plánu - teda mali by sa stanoviť ciele projektu (napr. nárast ziskovosti, vytvorenie menej rizikového portfólia atď.) V

tomto bode sa taktiež určí, či bude skóringový model vytvorený interne - zamestnancami spoločnosti, alebo ho bude vyvíjať externá spoločnosť. Nasleduje vytvorenie projektového plánu, ktorý zahŕňa určenie možných rizík projektu (spojené najmä s nedostupnosťou alebo nízkou kvalitou dát) a vytvorenie projektového tímu, kde sa rozdelia úlohy i zodpovednosti za jednotlivé časti projektu.

2.5 Príprava dát a určenie parametrov projektu

Fáza prípravy dát je väčšinou najdlhšia a najnáročnejšia časť celého procesu. Na jej základe môžeme určiť, či je vôbec vývoj modelu realizovateľný. Ak totiž vytvoríme model, ktorý bude dosahovať skvelý výkon, no vývojová vzorka dát je nespoľahlivá, môžeme tento model „hodiť do koša". Inými slovami, zo zlých vstupov sa nedá vytvoriť dobrý, dôveryhodný výstup - pre túto situáciu sa používa skratka GIGO (*Garbage In, Garbage Out*, doslova preložené ako *smeti dnu, smeti von*).

Táto fáza taktiež obsahuje stanovenie parametrov modelu zahŕňajúce napr. vylúčenia niektorých pozorovaní, definíciu vysvetľovanej premennej, časového intervalu, počas ktorého zbierame dáta - vzorkového intervalu (*sample window*), a intervalu, počas ktorého pozorujeme vysvetľovanú premennú - kontrolného intervalu (*performance window*).

Nemenej dôležitou súčasťou tejto fázy je určenie metodológie modelu.

2.5.1 Definícia parametrov projektu

Vzorkový a kontrolný interval

Všetky skóringové modely využívajú predpoklad, že budúce správanie klientov bude odrážať ich správanie v minulosti. Vychádzajúc z tohto predpokladu môžeme pre účely aplikačného skóringu odhadnúť správanie budúcich klientov tým, že analyzujeme správanie existujúcich klientov. K tejto analýze potrebujeme zhromaždiť dáta - informácie o účtoch otvorených počas určitého obdobia (vzorkový interval) a následne sledovať ich správanie počas ďalšieho obdobia (kontrolný interval).

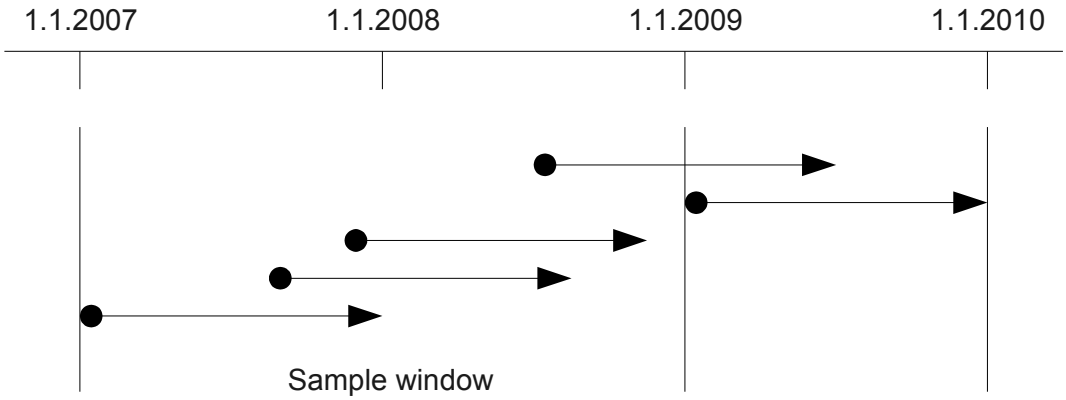
Kontrolný interval je časový interval dĺžky väčšinou 1 rok. Z tohto obdobia získavame iba jednu premennú - vysvetľovanú premennú, ktorá nesie informáciu o tom, či obchod alebo klient zlyhal. Deň pred začiatkom tohto intervalu sa často nazýva *deň analýzy*. Všetky ostatné premenné pochádzajú najneskôr z dňa analýzy.

V prípade aplikačného skóringu sú to predovšetkým dáta získané zo žiadosti klienta.

Medzi typické premenné, ktoré sa zbierajú pre potreby aplikačného skóringu, patria:

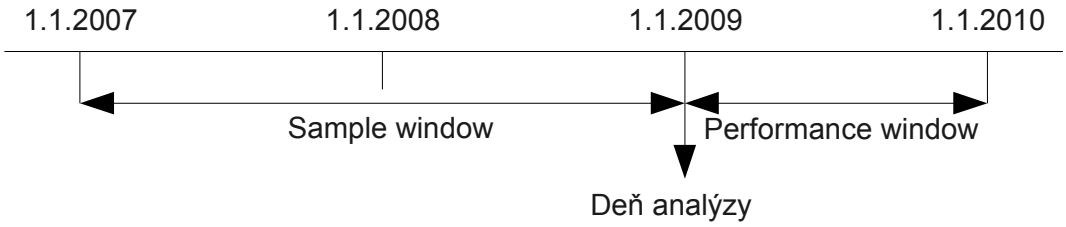
- osobné údaje o klientovi, napr. vek, príjem, zamestnanie, rodinný stav, vzdelanie... - týka sa fyzických osôb,
- pomerové ukazovatele vytvorené z finančných výkazov (pre právnické osoby a živnostníkov),
- informácie z úverového registra (databáza informácií o kreditnej kvalite klientov, do ktorého prispievajú všetky alebo väčšina bánk),
- informácie pochádzajúce z banky v prípade, ak žiadateľ už je klientom banky.

Na obrázku 2.1 znázorňuje každá šípka jedno pozorovanie - začiatok šípky je deň analýzy tohto pozorovania. Dĺžka šípky je v tomto prípade jeden rok a znázorňuje dĺžku kontrolného intervalu. Na konci šípky sa pýtame, či sa dané pozorovanie dostalo počas tohto obdobia do stavu zlyhania. Pozorovania berieme z obdobia od 1.1.2007 do 31.12.2008, čo zodpovedá vzorkovému intervalu. Všimnime si, že ak je dĺžka kontrolného intervalu jeden rok a predpokladáme, že dnes je 1.1.2010, potom najaktuálnejšie pozorovanie môže mať deň analýzy 31.12.2008 (Výnimku tvoria tie pozorovania, ktoré „stihli" zlyhať do aktuálneho dátumu). Bez splnenia predpokladu rovnakého správania klientov v minulosti a v budúcnosti by teda vyvíjanie modelu nemalo veľký zmysel.



Obrázok 2.1: Schéma výberu pozorovaní pre aplikačný skóring

V prípade behaviorálneho skóringu je situácia mierne odlišná. Deň analýzy sa volí spoločný pre všetky obchody. Kontrolný interval je jednoročné obdobie odo dňa analýzy, vzorkový interval sú dva alebo viac rokov spätne odo dňa analýzy. Príklad dvojročného vzorkového intervalu a jednoročného kontrolného intervalu je na obrázku 2.2.



Obrázok 2.2: Schéma výberu pozorovaní pre behaviorálny skóring

Pre účely behaviorálneho skóringu sa zbierajú dáta dávajúce informáciu o správaní klienta - napr. koľko čerpal zo svojho úverového limitu, či splácal načas splátky, výšku nesplatenej časti úveru, transakcie na účte atď.

Vylúčenia pozorovaní

Často sa stáva, že niektoré typy účtov alebo klientov je potrebné vylúčiť. Vo všeobecnosti, do vývojovej vzorky použijeme len tie typy pozorovaní, ktoré budeme modelom skórovať. Uvedieme zopár príkladov:

- klienti, ktorí majú neprirodzené správanie - napr. ak bol u nich odhalený podvod,
- klienti, pri ktorých proces schvaľovania nevyužíva skóre - napr. VIP klienti, zamestnanci, klienti s predschváleným úverom, zahraniční klienti,
- klienti, u ktorých nie je možné posúdiť, či zlyhali alebo nie - nedostatočná dĺžka kontrolného intervalu, úmrtie počas tohto intervalu,
- pozorovania pochádzajúce z regiónu (týka sa najmä väčších krajín) alebo z trhu, kde už spoločnosť nepôsobí - napr. lízingová spoločnosť, ktorá sa rozhodne predávať iba osobné autá, no v minulosti predávala okrem nich aj špeciálne zariadenia alebo priemyselné stroje, by mala pozorovania týkajúce sa týchto zariadení vylúčiť,

- ak mala banka marketingovú kampaň, ktorá zvýšila podiel určitej skupiny ľudí v portfóliu klientov (napr. hypotéka pre mladé rodiny), no predpokladá, že v budúcnosti už táto skupina nebude mať také zastúpenie.

Prečo je potrebné vylúčiť tieto pozorovania? Pretože vývojová vzorka by mala reprezentovať celú populáciu, portfólio daného produktu.

Definícia stavu zlyhania - defaultu

Pri definovaní stavu zlyhania - defaultu alebo „zlého" klienta by mala banka uvažovať o nasledujúcich veciach:

- definícia musí byť v zhode s cieľmi položenými pri plánovaní,
- čím „prísnejšia" je definícia, tým lepšie je rozlíšenie medzi „dobrými" a „zlými", no môže vzniknúť problém s nedostatkom „zlých" prípadov,
- a naopak, čím voľnejšia je definícia, tým viac máme „zlých" prípadov, no tým horšie je rozlíšenie medzi „dobrými" a „zlými",
- definícia by mala byť interpretovateľná a jasná, mala by sa dať získať z dát, ktoré ma banka k dispozícii.

Pre banky, ktoré prechádzajú na IRB prístup podľa (Basel II, 2006), problém definovania defaultu odpadá. Dokument totiž definuje zlyhanie jednotne pre všetky banky nasledovne:

Za vznik zlyhania v súvislosti s konkrétnym dlžníkom sa považuje, keď nastane jedna alebo obidve z týchto udalostí:

a) banka usúdi, že dlžník pravdepodobne nesplatí svoje kreditné záväzky voči banke v plnej výške bez toho, aby banka urobila úkony, akými je napríklad realizácia zabezpečenia (ak bolo poskytnuté);

b) dlžník je v omeškaní viac ako 90 dní pri splácaní akéhokoľvek podstatného kreditného záväzku voči banke.

Banka môže usúdiť, že dlžník pravdepodobne nesplatí svoje kreditné záväzky, ak je tento záväzok uvedený do nenarastajúceho stavu, ak je núdzovo reštrukturalizovaný, ak bol na klienta vyhlásený konkurz, a pri ďalších situáciách.

2.5.2 Segmentácia

V niektorých prípadoch použitie viacerých skóringových modelov pre dané portfólio poskytuje lepší odhad rizika ako jeden model. Zvyčajne to je situácia, kedy je populácia zložená z viacerých subpopulácií, ktoré sa od seba líšia. Jeden model nemusí byť dostatočný, pretože každú subpopuláciu charakterizujú iné premenné. Proces identifikovania týchto subpopulácií sa nazýva *segmentácia*. Poznáme dva hlavné typy segmentácie:

1. heuristická segmentácia - založená na skúsenostiach a expertných znalostiach portfólia,

2. segmentácia s použitím štatistických techník.

V oboch prípadoch by však segmenty mali mať dostatočný počet „dobrých" aj „zlých" pozorovaní, aby sa pre tieto segmenty dali vyvíjať zmysluplné skóringové modely.

Typické charakteristiky používané v heuristickej segmentácii sú demografické dáta (vek, región, počet rokov v banke...), typ produktu, druh sprostredkovania obchodu (na pobočke, cez internet, cez finančného sprostredkovateľa...), typ žiadateľa (nový/už existujúci klient) atď.

Samozrejme, každá potenciálna segmentácia musí byť analyzovaná - empiricky potvrdená. Jednou z možných metód je pozorovať správanie nejakej charakteristiky. Uvedieme si ilustračný príklad uvedený v (Siddiqi, 2006). Segmentáciou podľa veku sa rozdelila celá vzorka na dve skupiny - žiadatelia do 30 rokov a nad 30 rokov. Kvalitu segmentácie môžeme pozorovať na dvoch premenných - spôsobe bývania a počte otvorených úverových produktov. V tabuľke 2.1 sledujeme podiel „zlých" klientov pre jednotlivé skupiny. Vidíme, že charakteristiky majú rozdielny vývoj pre oba segmenty. Pre žiadateľov mladších ako 30 rokov je bežné, že žijú ešte v rodičovskom dome. Ak však u rodičov bývajú aj po tridsiatke, môže to byť prejav toho, že nie sú úplne samostatní, čo sa ukázalo aj vo vyššej miere rizika. Podobná situácia je pri počte úverových produktov - ak má mladý človek veľa produktov, môže to byť prejavom jeho životného štýlu - je ochotný žiť v zadĺžení. Takíto ľudia majú často problém so splácaním úverov.

Typickými štatistickými nástrojmi pri druhom type segmentácie je zhlukovanie pomocou metódy k priemerov alebo SOM (Self-Organizing Maps). Taktiež sa používajú rozhodovacie stromy - napr. CART spomínané v prvej kapitole. Aplikovateľná je aj metóda náhodných lesov a konkrétne jeden z jej vedľajších produktov - matica blízkosti.

	Vek > 30	Vek < 30	Bez segmentácie
Stav bývania			
Podnájom	2.1%	4.8%	2.9%
Vlastné bývanie	1.3%	1.8%	1.4%
U rodičov	3.8%	2.0%	3.2%
Počet produktov			
0	5.0%	2.0%	4.0%
1-3	2.0%	3.4%	2.5%
4 a viac	1.4%	5.8%	2.3%

Tabuľka 2.1: Príklad segmentácie podľa veku

2.5.3 Metodológia modelu

Špecialista pre kreditný skóring má na výber pomerne veľké množstvo dostupných matematických techník, ktoré sú vhodné na vývoj skóringového modelu. V súčasnosti najpoužívanejšou metódou je logistická regresia. V prípade, že sa v logistickej regresii použijú ako vstupy tzv. *WOE premenné*, ktoré sú odvodené z originálnych premenných pomocou zoskupovania ich hodnôt na podintervaly alebo podmnožiny a o ktorých si povieme viac neskôr, výstupom je tzv. skórkarta (*scorecard*), ktorej ilustrácia je v tabuľke 2.2. Napr. 30-ročný klient s vysokoškolským vzdelaním a príjmom 850 bude mať za prvé tri premenné v skórkarte $75 + 67 + 61 = 203$ bodov skóre.

Medzi ďalšie použiteľné techniky patria: neurónové siete, lineárna regresia, support vector machines (SVM), rozhodovacie stromy (napr. CART, ale aj C4.5), model k najbližších susedov (k-NN), matematické programovanie a ďalšie. V posledných rokoch idú do popredia metódy založené na agregovaní viacerých modelov do jedného celku (tzv. *ensemble learning*), a najmä na agregovaní rozhodovacích stromov. Takto vznikli metódy bagging, arcing, Adaboost a náhodné lesy. Náhodné lesy, na rozdiel od zvyšných troch metód, sa nesnažia vytvoriť čo najpresnejšie stromy, cieľom totiž nie je minimalizovať chybu stromov, ale celého lesa. V prvej kapitole sme si taktiež predstavili metódu podmienených náhodných lesov, ktorá je nevychýlenou alternatívou náhodných lesov.

Voľba metódy pre vývoj modelu závisí od viacerých faktorov:

- kvality dostupných dát. Ak je v dátach prítomný veľký počet chýbajúcich polí (missingov), je vhodnejšie použiť techniku rozhodovacích stromov (a teda aj náhodných lesov). To isté platí pre dáta, kde sú silné korelácie medzi

Premenná	Atribút	Skóre
Vek	18 - 23	63
Vek	23 - 31	75
Vek	31 - 45	87
Vek	> 45	99
Vzdelanie	základoškolské	-10
Vzdelanie	stredoškolské	42
Vzdelanie	vysokoškolské	67
Príjem	< 400	12
Príjem	400 - 900	61
Príjem	900 - 1 750	84
Príjem	> 1 750	101
...	...	...

Tabuľka 2.2: Príklad aplikačnej skórkarty (jej časť)

prediktormi,

- veľkosti vzorky. Niektoré metódy potrebujú ku kvalitným výsledkom veľké množstvo pozorovaní.

- IT možností pri implementácii. Napríklad model pomocou neurónových sietí môže byť ideálny z hľadiska predikcie, no je nepoužiteľný, ak ho IT riešenia neumožnia implementovať do praxe,

- potreby ľahkej interpretácie modelu. Toto je zvlášť kameň úrazu napr. pri neurónových sieťach. Ťažko interpretovateľný model sa často nazýva „čierna skrinka" (*black box*). Naproti tomu skórkarta vytvorená pomocou logistickej regresie má jasnú interpretáciu.

- schopnosti merať kvalitu modelu - pre IRB prístup podľa dokumentu Basel II (2006) je potrebná pravidelná validácia modelu na základe viacerých štatistík merajúcich predikčnú silu a stabilitu modelu.

2.6 Vytvorenie vývojovej vzorky

Ak máme definované parametre projektu a segmentáciu, môžeme pokračovať vytvorením dátovej vzorky (alebo viacerých vzoriek), ktoré budú obsahovať množinu charakteristík - prediktorov a vysvetľovanú premennú pre každé pozorova-

nie, ktoré vstúpi do vývoja skóringového modelu. Dôležitým krokom je výber vhodných prediktorov. Tieto premenné môžu byť volené s ohľadom na nasledujúce faktory:

- očakávaná predikčná sila - môže byť získaná z predchádzajúcich analýz a projektov alebo zo skúseností schvaľovateľov úverov,
- spoľahlivosť a robustnosť - zber niektorých informácií môže byť zmanipulovaný; dáta, ktorých vyplnenie v dotazníku nie je povinné, môžu byť z veľkej časti nevyplnené,
- interpretovateľnosť - napr. prístup, kedy vytvárame kombinácie (pomery, súčiny...) z jednotlivých ukazovateľov, nám môže pomôcť nájsť premennú, ktorá má možno predikčnú silu, ale nie je interpretovateľná,
- právny pohľad - dáta podozrivé z právneho hľadiska by sa nemali používať (napr. pohlavie klienta),
- dostupnosť dát v budúcnosti - je potrebné uistiť sa, že zber dát, ktoré sa chystáme použiť, je aktuálny a plánovaný aj v budúcnosti.

(Siddiqi, 2006) uvádza, že skóringový model býva vyvinutý pomocou dát dva až tri roky starých a očakáva sa jej používanie v období približne dvoch ďalších rokov. Preto je vhodné skúmať aj trend charakteristík v čase - hoci to neovplyvní samotné dáta, táto analýza môže pomôcť pri tvorbe vhodných stratégií.

Vývojová a validačná vzorka

V praxi sa často množina všetkých pozorovaní delí do vývojovej a validačnej vzorky. Väčšina, 70% - 80% pozorovaní je použitých na vývoj, zvyšných 20% - 30% slúži na validáciu modelu. Ak máme k dispozícii len malý počet pozorovaní, používajú sa všetky na vývoj. Model môže byť v tomto prípade validovaný na množine náhodne vybranej vzorky, každá veľkosti 50% - 80%.

Pre vývoj modelu je podľa (Siddiqi, 2006) vhodné mať okolo 2 000 „dobrých" a 2 000 „zlých" pozorovaní a v prípade aplikačného skóringu navyše 2 000 zamietnutých žiadostí. V praxi býva často veľkým problémom najmä dostatočný počet „zlých" pozorovaní.

Ak nepoužívame všetky dáta, pozorovania vstupujúce do vzoriek by mali byť vybrané náhodne a tak, aby reprezentovali populáciu, ktorá bude v budúcnosti skórovaná. V prípade viacerých segmentov sa musia vytvoriť okrem vzoriek pre každý segment aj nesegmentované vzorky - kvôli analýze, ktorá meria „výhodu" použitia viacerých modelov oproti jednému pre všetky segmenty.

Oversampling

V prípade, že v populácii je veľmi nevyvážený podiel „zlých" a „dobrých" pozorovaní, môžu sa vytvoriť vzorky, ktoré majú iný, vyšší podiel „zlých" pozorovaní ako v celej populácii. Táto technika sa nazýva *oversampling*. Pri jej použití je však potrebné upraviť odhadovanú pravdepodobnosť defaultu.

2.7 Vyvíjanie modelu

Skôr, ako sa začne so samotným vývojom modelu, je veľmi odporúčané preskúmať dodané dáta vo vývojovej vzorke. Jednoduché štatistiky ako distribúcia hodnôt, priemer, medián, minimum, maximum a podiel chýbajúcich hodnôt pre každú premennú nám môže pomôcť lepšie spoznať portfólio a odhaliť prípadné chyby. Takisto by mala byť kontrolovaná interpretácia konkrétnych hodnôt (napr. uistiť sa, že hodnota „0" reprezentuje naozaj nulu a nie chýbajúcu hodnotu).

2.7.1 Chýbajúce hodnoty a outliere

Väčšina dát v bankovom sektore obsahuje chýbajúce hodnoty - **missingy**. Zatiaľ čo niektoré metódy sú schopné pracovať aj s missingami (napr. rozhodovacie stromy), iné vyžadujú kompletné dáta bez missingov (napr. logistická regresia). Existuje viacero prístupov k chýbajúcim hodnotám:

1. vylúčiť všetky pozorovania, ktoré majú chýbajúcu hodnotu. Toto môže vyústiť do veľmi malej vzorky alebo vzorky, ktorá nebude reprezentatívna,
2. vylúčiť premenné, ktoré majú značnú časť (napr. 50%) missingov - vhodné najmä vtedy, keď neočakávame, že sa v budúcnosti situácia pri zbere dát zlepší,
3. zahrnúť premenné s týmito hodnotami do modelu - missing bude braný ako špecifická hodnota vstupujúca do modelu. V d'alšom texte si bližšie vysvetlíme, ako prideliť túto špecifickú hodnotu,
4. pripísať missingu reálnu hodnotu použitím nejakej štatistickej techniky (napr. pomocou matice blízkosti z metódy náhodných lesov).

1., 2. a 4. prístup predpokladá o chýbajúcich hodnotách, že nenesú žiadnu informáciu. Toto však v bankových dátach nemusí byť nutne pravda - chýbajúca hodnota premennej sa napr. môže vzťahovať k inej premennej alebo indikovať

zlé správanie. Dáta by mali byť dostatočne spoľahlivé, takže chýbajúce hodnoty majú zvyčajne svoj význam a nie sú náhodné. Napríklad žiadatelia, ktorí majú nové zamestnanie, pravdepodobne nevyplnia pole *Počet rokov v zamestnaní*. Preto sa skôr odporúča 3. prístup, ktorý predpokladá, že chýbajúce hodnoty nesú v sebe informáciu. Navyše, zahrnutie missingov do modelu odstráni problém s budúcimi dátami, ktoré budú opäť obsahovať chýbajúce hodnoty.

Ak však z charakteru premennej je zrejmé, že sa jedná o chýbajúcu hodnotu, ktorá by mala byť nahradená (to, že chýba, nemá žiadnu interpretáciu), je vhodné použiť 4. prístup - pripísať missingom nejakú hodnotu. Tu sa môže opäť použiť ďalšia z aplikácií matice blízkosti získanej pomocou metódy náhodných lesov, ktorú sme si predstavili v prvej kapitole.

Pri skúmaní dát by sa mala takisto upriamiť pozornosť na **outliere** - hodnoty, ktoré sú za prirodzenými hranicami premennej, napr. vek klienta 95 rokov. Môžu to byť pravdivé hodnoty, ale pravdepodobnejšie je, že boli chybne zadané do databázy. Outliere môžu mať zlý vplyv najmä pri parametrických modeloch (napr. logistická regresia). Tieto hodnoty môžu byť vylúčené alebo im môže byť pridelený priemer danej premennej. Metóda náhodných lesov, konkrétne matica blízkosti, nám môže pomôcť hľadať outliere.

2.7.2 Počiatočná analýza prediktorov

V tejto analýze meriame predikčnú silu každej premennej. Slabé premenné alebo tie, ktoré sa správajú proti ekonomickej interpretácii, vylúčime z modelovania.

Ak je navrhnuté v metodológii, že model sa bude vyvíjať pomocou logistickej regresie, je vhodné transformovať zostávajúce premenné na zoskupené WOE (*weight of evidence*) premenné, čo nám umožní vytvoriť skórkartu vo forme podobnej tabuľke 2.2.

WOE premenné sa z intervalových premenných vytvoria tak, že celý interval hodnôt pôvodnej premennej sa rozdelí na podintervaly - tzv. atribúty. Pri kategorických premenných analogicky rozdelíme všetky triedy na podskupiny. Každému atribútu sa pridelí číselná hodnota, WOE váha. Príklad rozdelenia konkrétnej premennej - veku na atribúty je v tabuľke 2.3.

Vysvetlime si význam jednotlivých stĺpcov tabuľky: Prvý stĺpec udáva rozdelenie veku na atribúty. Všimnime si, že chýbajúce pozorovania sú vo zvláštnom atribúte. Ďalšie dva stĺpce dávajú informáciu o absolútnom a relatívnom počte pozorovaní v jednotlivých atribútoch. Podobne je to pre počty „dobrých" G (good) a „zlých" B (bad) pozorovaní. B rate je podiel „zlých" v danom atribúte. Posledný

Vek	Počet	Distr.	G	Distr. G	B	Distr. B	B rate	WOE
missing	100	2.50%	86	2.38%	14	3.65%	14.00%	-42.72
18 - 22	400	10.00%	304	8.41%	96	25.00%	24.00%	-108.98
23 - 26	600	15.00%	492	13.61%	108	28.13%	18.00%	-72.61
27 - 29	900	22.50%	810	22.40%	90	23.44%	10.00%	-4.53
30 - 35	1000	25.00%	950	26.27%	50	13.02%	5.00%	70.20
36 - 44	700	17.50%	680	18.81%	20	5.21%	2.86%	128.39
44+	300	7.50%	294	8.13%	6	1.56%	2.00%	164.94
Spolu	**4000**	**100.00%**	**3616**	**100.00%**	**384**	**100.00%**	**9.60%**	

Tabuľka 2.3: Analýza WOE premennej - veku

stĺpec, WOE, nám udáva váhu atribútu, vypočítanú nasledujúcim vzťahom:

$$WOE_i = 100.\ln\left(\frac{\text{Distr.G}_i}{\text{Distr.B}_i}\right). \tag{2.1}$$

Násobenie stovkou sa používa len kvôli lepšiemu prehľadu a porovnávaniu. Atribúty sa musia voliť podľa viacerých pravidiel:

- každý by mal obsahovať dostatočný počet pozorovaní (napr. 5%). Chýbajúce hodnoty v príklade toto pravidlo nespĺňajú, preto by bolo možno vhodnejšie priradiť ich k atribútu, ktorý má podobný podiel „zlých" (23 - 26 alebo 27 - 29),
- každý musí mať aspoň 1 „dobré" a aspoň 1 „zlé" pozorovanie, kvôli výpočtu WOE,
- WOE susedných atribútov by sa malo od seba čo najviac odlišovať,
- v atribútoch pre kategorické premenné by sa mali nachádzať kategórie, ktoré má zmysel spolu zoskupovať,
- WOE by mala spĺňať logický trend, teda napr. pre premennú vek očakávame, že čím starší je klient, tým je menej rizikový. WOE by teda s rastúcim vekom malo monotónne stúpať. Niekedy je preto potrebné znížiť počet atribútov a tým zaistiť logický trend premennej. Logický trend pre niektoré premenné môže samozrejme nadobúdať aj tvar písmena U alebo obráteného U, ak okrajové hodnoty znamenajú rizikovejšie správanie ako hodnoty okolo stredu.

Predikčná sila WOE premennej sa môže merať pomocou tzv. informačnej hodnoty (information value, IV):

$$IV = \sum_{i=1}^{n} (\text{Distr.G}_i - \text{Distr.B}_i).\ln\left(\frac{\text{Distr.G}_i}{\text{Distr.B}_i}\right), \tag{2.2}$$

kde n je počet atribútov danej premennej. Malé hodnoty IV znamenajú nízku silu premennej, hodnoty medzi 0.1 až 0.3 sú považované za stredne silné a vyššie hodnoty ako silné.

2.7.3 Tvorba modelu

V závislosti od zvolenej metodológie modelu môžeme začať s jeho vyvíjaním. V prvej kapitole sme si podrobne predstavili metódu náhodných lesov. Keďže cieľom práce je porovnať ho s typickým modelom kreditného skóringu - logistickou regresiou, v krátkosti si ju predstavíme.

Logistická regresia pre binárnu vysvetľovanú premennú y používa súbor k vysvetľovaných premenných na odhadnutie pravdepodobnosti, že vysvetľovaná premenná bude 1. Táto pravdepodobnosť je transformovaná pomocou funkcie logit. Rovnica logistickej regresie je potom nasledovná:

$$\text{logit}(p) = \beta_0 + \beta_1 X_1 + ... + \beta_k X_k = \beta^\top X, \tag{2.3}$$

kde $\beta = (\beta_0, \beta_1, ..., \beta_k)^\top$, $X = (1, X_1, ..., X_k)^\top$, β_0 je posunutie regresnej priamky (konštanta) a β_1 až β_k sú odhadované koeficienty príslušných premenných X_1 až X_k. Skalárny súčin $\beta^\top X$ sa nazýva skóre. Toto skóre nadobúda kladné aj záporné hodnoty rádovo v jednotkách, preto je vhodné transformovať ho lineárnou funkciou na nové skóre, ktoré je lepšie čitateľné (napr. skóre od 0 po 1000). Typicky je táto funkcia volená tak, že čím je vyššie skóre, tým lepší (menej rizikový) je klient.

Odhadovaná pravdepodobnosť p má tvar:

$$p = P(y = 1|X) = \frac{e^{\beta^\top X}}{1 + e^{\beta^\top X}}. \tag{2.4}$$

Cieľom je vložiť do modelu dostatočný počet zmysluplných premenných - malým počtom získame len úzky rizikový profil klienta. Samozrejme, premenné by nemali byť navzájom korelované a takisto nesmie byť prítomná multikolinearita. Model vytvorený pomocou logistickej regresie máva v praxi 8 až 15 premenných. Tu vidno výhodu náhodných lesov - do predikcie môžeme zahrnúť veľký počet premenných, čím získame komplexnejší profil klienta.

2.7.4 Zahrnutie zamietnutých žiadostí

Pri vývoji aplikačného skóringu využívame iba pozorovania, o ktorých vieme, že sa v kontrolnom intervale dostali do stavu defaultu alebo nie - teda zahŕňame len schválené žiadosti. Tieto však nie sú reprezentatívnou vzorkou, pretože neobsahujú žiadosti, ktoré boli zamietnuté analytikom alebo starším skóringovým modelom. Môžeme teda predpokladať, že tieto žiadosti získajú v priemere nižšie skóre ako žiadosti schválené. V knihe (Siddiqi, 2006) je uvedených viacero spôsobov, ako zahrnúť zamietnuté žiadosti do modelu (známe ako *Reject Inference*), my si uvedieme algoritmus jedného z nich, tzv. *tvrdý cutoff*:

1. Vytvoríme model, tak ako doteraz, použitím vývojovej vzorky, kde sú len schválení klienti.

2. Pomocou tohto modelu oskórujeme vzorku zamietnutých klientov, resp. prisúdime ich pravdepodobnosť zlyhania (PD).

3. Určíme cutoff - hranicu pre PD, pričom zamietnuté žiadosti s PD väčším ako cutoff budeme brať ako „zlé" a zvyšné ako „dobré". Odporúča sa voliť ju ako najhoršie PD, ktoré je v súčasnosti pri schvaľovaní ešte akceptovateľné.

4. Pridáme takto odvodené „dobré" a „zlé" žiadosti k vývojovej vzorke a vyvinieme model ešte raz.

Táto metóda je jednoduchá a má hlavný nedostatok vo voľnosti voľby cutoff hranice.

Aj v tomto probléme by mohla byť osožná metóda náhodných lesov a konkrétne matica blízkosti, do ktorej by sa pridali aj zamietnuté žiadosti. Následne by sa skúmali najbližšie známe (schválené) pozorovania každého neznámeho (zamietnutého) pozorovania a na základe nich by sa priradila trieda tomuto pozorovaniu.

2.7.5 Výber finálneho modelu

Aj keď sme doposiaľ spomínali vždy vývoj jedného modelu, v praxi je bežné, že sa vyvinie viacero modelov. Na mieste je potom otázka: ktorý model je ten najlepší? A v akom zmysle je najlepší? Odpoveď môžeme hľadať porovnávaním rôznych štatistických ukazovateľov a mier. Ak to ukazovateľ umožňuje, aplikujeme ho aj na validačnú vzorku dát. Situácia, keď sa štatistika pre vývojovú vzorku značne líši od štatistiky pre validačnú vzorku, môže naznačovať pretrénovanie na vývojovej vzorke a teda nestabilitu modelu.

Chybné klasifikovanie

V úvode prvej kapitoly sme si predstavili maticu zatriedenia pozorovaní a s tým spojené chyby nesprávneho klasifikovania. Pozorovanie je zatriedené ako „zlé", ak má hodnotu skóre nižšiu ako vopred zvolená hranica (*cutoff*).

Táto miera však nezohľadňuje presné hodnoty skóre, iba to, či je skóre vyššie alebo nižšie ako cutoff. Preto v praxi nie je príliš používaná.

Informačné kritériá

Akaikeho a Schwarz-Bayesovo informačné kritérium penalizujú pridávanie parametrov do modelu. Samostatne nemajú význam, slúžia len pre porovnanie dvoch modelov, pričom nižšia hodnota znamená lepší model - s vyššou predikčnou silou. Nevýhodou je, že sú použiteľné len pre parametrické metódy, teda nedajú sa vypočítať napr. pre náhodné lesy.

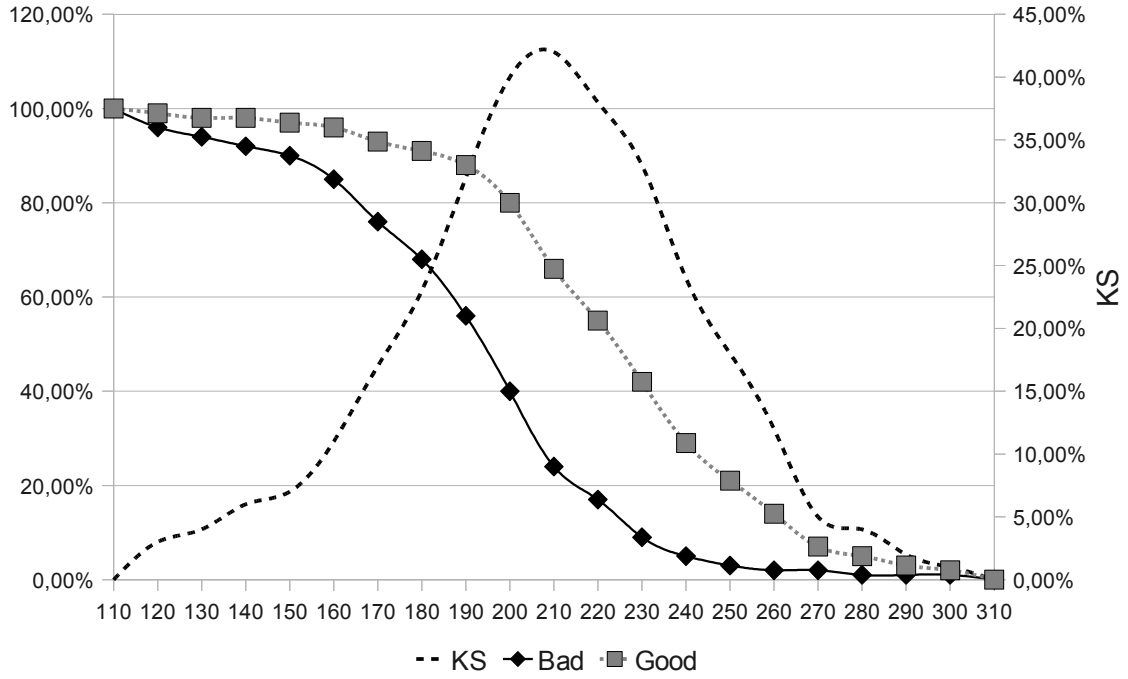
Kolmogorovova-Smirnovova štatistika

Takisto meria predikčnú silu modelu - udáva maximálny vertikálny rozdiel medzi kumulatívnymi distribúciami „dobrých" a „zlých" pozorovaní podľa skóre, ako je naznačené na obrázku 2.3. Čím vyššia hodnota, tým lepšie model rozlišuje „dobré" pozorovania od „zlých".

ROC krivka

ROC (*Receiver Operating Characteristic*) krivku získame, ak nanášame do grafu kumulatívne percento „dobrých" pozorovaní ku kumulatívnemu percentu „zlých" pozorovaní vzhľadom na meniace sa skóre (Obr. 2.4).

Čím bližšie je krivka k bodu $(0, 1)$, tým je model silnejší. Krivka ideálneho modelu by bola spojnica bodov $(0, 0) \rightarrow (0, 1) \rightarrow (1, 1)$. Takýto model by presne rozlíšil „dobrého" klienta od „zlého". Model, ktorý skóre prideľuje náhodne, by mal ROC krivku podobnú spojnici bodov $(0, 0) \rightarrow (1, 1)$. Preto kvalitu modelu vyjadruje veľkosť plochy pod krivkou - táto miera sa nazýva **AUC** (Area Under Curve) alebo aj **c-štatistika** - hodnota 0.5 znamená model s náhodným skóre a hodnota 1 optimálny model. Ekvivalentná miera je **Gini koeficient** - je to iba lineárna transformácia c-štatistiky tak, aby jej obor hodnôt bol medzi 0 a 1, $\mathrm{Gini} = 2\mathrm{c} - 1$. Ďalšou štatistikou odvodenou z ROC krivky je **Pietra index**, ktorý je rovný polovici z maximálnej vertikálnej vzdialenosti medzi ROC krivkou a diagonálou. Pietra index je ekvivalentný ku Kolmogorovovej-Smirnovovej štatistike.



Obrázok 2.3: Kolmogorovova-Smirnovova štatistika

Lorenzova krivka

Táto krivka je podobná ROC krivke. Rozdiel je v tom, že nanášame do grafu kumulatívne percento „zlých" pozorovaní ku kumulatívnemu percentu všetkých pozorovaní.

Brierovo skóre

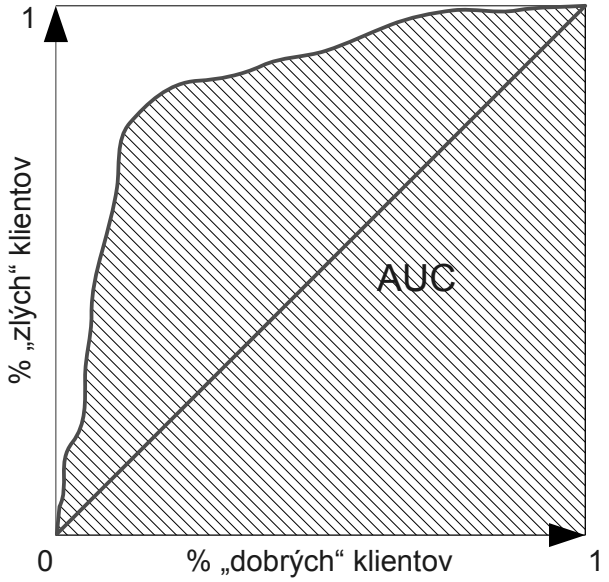
Brierovo skóre je definované ako priemer štvorcov rozdielov medzi skutočnou triedou (1 znamená zlyhanie, 0 inak) a odhadnutou pravdepodobnosťou zlyhania. Je to vlastne suma štvorcov rezíduí:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$$

Čím nižšia je jeho hodnota, tým je model presnejší.

Miery na základe entropie

V dokumente Working Paper No. 14 (2005) sa okrem doteraz spomenutých štatistík uvádzajú aj miery založené na informačnej entropii. Podľa tohto konceptu sa pozeráme na klienta (alebo na obchod) ako na experiment - či zlyhá alebo nie. Ak



Obrázok 2.4: ROC krivka

nemáme informáciu o skóre alebo odhadnutom PD, potom informačná entropia je definovaná ako

$$IE(p) = -(p(d)\log(p(d)) + (1-p(d))\log(1-p(d))),$$

kde $p(d)$ je pravdepodobnosť zlyhania získaná ako podiel zlyhaných pozorovaní v celej vzorke. Ak uvažujeme model, ktorý každému pozorovaniu priradí skóre s, potom podmienená informačná intropia vo vzorke je

$$CIE = -E[(p(d|s)\log(p(d|s)) + (1-p(d|s))\log(1-p(d|s)))].$$

Strednú hodnotu odhadujeme pomocou priemeru cez všetky pozorovania vo vzorke. Dá sa ukázať, že $CIE \leq IE$. Ich rozdiel teda môže byť ďalšou vhodnou mierou predikčnej sily modelu. Nazýva sa Kullbackova - Leiblerova divergencia,

$$KLD = IE - CIE.$$

Zrejme však KLD je citlivé na celkovú nepodmienenú pravdepodobnosť zlyhania, preto je tento rozdiel normalizovaný nepodmienenou informačnou entropiou. Táto miera sa nazýva koeficient podmienenej informačnej entropie (*Conditional Information Entropy Ratio*),

$$CIER = \frac{IE - CIE}{IE}.$$

2.8 Tvorba manažérskych reportov

Manažérske reporty majú pomôcť pri rozhodovaní, ako nastaviť cutoff, teda minimálnu hranicu skóre, kedy je banka ochotná schváliť takúto žiadosť. Typickým reportom je tzv. *prírastková tabuľka* (Gains table), ktorá je vytvorená na celej vzorke pozorovaní a používa sa najmä pri aplikačnom skóringu. Ukážka jej časti je v tabuľke 2.4. Celý interval skóre je rozdelený na malé podintervaly. Každý riadok tabuľky nám dáva informáciu o tom, koľko pozorovaní získalo skóre patriace do daného intervalu a aký je podiel „zlých" v tomto intervale (stĺpec Inteval B rate). Kumulatívne počty nám hovoria, koľko je všetkých, „dobrých" a „zlých" pozorovaní, ktoré získali aspoň dolnú hranicu skóre daného intervalu. Najdôležitejšie sú posledné dva stĺpce. Ak by sa manažment banky rozhodol, že cutoff bude rovný spodnej hranici intevalu skóre v nejakom riadku (napr. 215), očakávaný podiel zlyhaných klientov bude 5.94% a očakávaný podiel schválených klientov bude 66.20%. Čím nižšie bude zvolený cutoff, tým viac klientov bude schválených, no o to horšie bude výsledné portfólio.

Skóre	**Počet**	**Kumul. počet**	**Kumul. Good**	**Kumul. Bad**	**Interval B rate**	**Kumul. B rate**	**Approval rate**
210-214	345	6965	6538	427	9.86%	6.13%	69.50%
215-219	500	6620	6227	393	7.60%	5.94%	66.20%
220-224	450	6120	5765	355	7.11%	5.80%	61.20%
225-229	345	5670	5347	323	6.38%	5.70%	56.70%
...							

Tabuľka 2.4: Časť prírastkovej tabuľky

Banka môže použiť aj viac ako jeden cutoff, resp. rozdeliť interval skóre na tzv. ratingové pásma alebo rizikové triedy. Tieto pásma sa typicky označujú písmenami A, B, C,... Pri schvaľovaní potom môžu byť napr. automaticky schválené žiadosti s ratingom A a B, žiadosti s ratingom C a D posúdi úverový analytik, a tie s ratingom E budú automaticky zamietnuté.

Okrem prírastkových tabuliek sú užitočné aj reporty na základe jednej konkrétnej premennej, ktorá vystupuje v skóringovom modeli. Sledujú sa distribúcie a podiely „zlých" prípadov v jednotlivých atribútoch premennej.

2.9 Implementácia modelu

Dôležitým prvkom tejto fázy je predimplementačná validácia modelu. Zatiaľ čo pri validácii modelu sa skúma robustnosť modelu porovnávaním distribúcií skóre vývojovej a validačnej vzorky, tu je cieľ odlišný - chceme sa uistiť, že vyvinutý model je platný aj pre aktuálnu populáciu klientov. Keďže vývojová a validačná vzorka je v niektorých prípadoch dva až tri roky stará, v populácii mohli nastať významné posuny v profile klienta.

2.9.1 Stabilita systému

Praktickým nástrojom na skúmanie zmien v populácii je tabuľka stability systému. Jej ukážka je v tab. 2.5.

Skóre	Actual %	Expected %	(A-E)	A/E	ln(A/E)	Index
0-169	7%	8%	-1%	0.8750	-0.1335	0.0013
170-179	8%	10%	-2%	0.8000	-0.2231	0.0045
180-189	7%	9%	-2%	0.7778	-0.2513	0.0050
190-199	11%	10%	1%	1.1000	0.0953	0.0010
...						

Tabuľka 2.5: Časť tabuľky stability systému

A (Actual) označuje aktuálnu populáciu, E (Expected) populáciu použitú pri vývoji modelu. Stĺpec Index je vypočítaný ako

$$(\mathbf{Actual}\% - \mathbf{Expected}\%)\ln(\mathbf{Actual}\%/\mathbf{Expected}\%).$$

Suma stĺpca Index sa nazýva index stability. Je používaný pre vyhodnotenie stability systému - hodnoty menšie ako 0.10 znamenajú, že v populácii sa neudiala žiadna významná zmena. Hodnoty medzi 0.10 a 0.25 znamenajú malú zmenu, ktorá by mala byť ďalej prešetrená. Index stability nad 0.25 naznačuje výraznú zmenu v populácii. Index nám dáva iba informáciu, či sa zmenila populácia, no nehovorí, v akom zmysle sa prípadná zmena udiala. Pre tieto účely je vhodný napr. graf distribúcie skóre aktuálnej populácie vzhľadom na distribúciu skóre očakávanej populácie. Dôležité je taktiež hľadať príčiny rozdielnosti populácií. Na to slúži analýza stability podľa nejakej konkrétnej charakteristiky. Tabuľka 2.6 ukazuje príklad distribúcie podľa veku. Je zrejmé, že v populácii nastal posun smerom k mladším klientom, čo mohlo spôsobiť zhoršenie kvality portfólia.

Vek	Actual %	Expected %	A-E
18 - 24	21%	12%	9%
25 - 29	25%	19%	6%
30 - 37	28%	32%	-4%
38 - 45	6%	12%	-6%
46 +	20%	25%	-5%

Tabuľka 2.6: Tabuľka stability systému

V prípade výraznej zmeny populácie sa ponúka len málo riešení. Vývijanie nového modelu pravdepodobne neodstráni problém, pretože by sa použila tá istá vývojová vzorka. V takýchto prípadoch je vhodné vytvoriť prírastkovú tabuľku (predstavenú z predchádzajúcej časti) na základe aktuálnych dát. Keďže v aktuálnej populácii nemáme informáciu o zlyhaní, odporúča sa v prírastkovej tabuľke ponechať podiel „zlých" nezmenený.

2.9.2 Stanovenie ďalších stratégií

Ďalšou časťou implementácie je súbor rozhodnutí, ktoré určia stratégiu. Patria medzi ne:

- kombinovanie skóre v prípade, že je v banke vyvinutých viac modelov, pričom každý je zameraný na iné charakteristiky klienta. Napríklad pre fyzické osoby - podnikateľov môžeme vytvoriť dva aplikačné modely - prvý, založený na finančných ukazovateľoch (získaných z finančných výkazov) a druhý, kde vstupujú demografické a všetky ostatné údaje,
- nastavenie cutoff hraníc,
- stanovenie pravidiel - KO kritérií (napr. žiadatelia mladší ako 18 rokov, nezamestnaní, podniky v konkurze atď),
- ďalšie pravidlá, ktoré sú posudzované úverovým analytikom pre každú žiadosť individuálne (tzv. *overrides*).

2.10 Monitoring

Ak chceme zabezpečiť, aby vyvinutý a do praxe zavedený model dával presné výsledky, je potrebné pravidelne monitorovať jeho výkon a správanie portfólia.

Na základe monitorovania modelu môžeme rozhodnúť, či je sila modelu dostatočná alebo či treba model upraviť, prípadne úplne nahradiť iným.

Banky, ktoré sa rozhodli používať podľa dokumentu Basel II (2006) IRB prístup, majú povinnosť pravidelne monitorovať všetky modely, ktoré produkujú odhady rizikových parametrov (PD, LGD, EAD a rating) použitých pri výpočte minimálnych kapitálových požiadaviek. V dokumente Working Paper No. 14 (2005) sú definované konkrétne nástroje, ktoré sa používajú pri validácii a monitoringu týchto modelov. Model kreditného skóringu, ako sme si ho predstavili, nám produkuje odhad parametra PD, teda pravdepodobnosti zlyhania, a taktiež je základom pre pridelenie ratingovej triedy. Štatistické miery a iné nástroje, ktoré sa používajú pri monitoringu skóringových modelov, sme si už predstavili v časti 2.7.5.

Okrem predikčnej sily modelu sa monitoruje kreditná kvalita populácie a jej stabilita, pričom sa skúmajú príčiny prípadných zmien.

Kapitola 3

Kreditný skóring pomocou náhodných lesov

V tejto kapitole je našou snahou vyvinúť skóringové modely pomocou metódy náhodných lesov a metódy podmienených náhodných lesov. Pomocou rôznych mier predstavených v časti 2.7.5 ich porovnáme s modelom založeným na logistickej regresii. Najprv je však potrebné, postupujúc podľa procesu vývoja opísaného v predchádzajúcej kapitole, pripraviť si vhodné dáta a zostrojiť vývojovú vzorku.

3.1 Príprava dát a vytvorenie vývojovej vzorky

Pre účely vývoja modelov nám boli poskytnuté interné dáta pochádzajúce z jednej z komerčných bánk na Slovensku. Tabuľka obsahuje dáta o približne 15 000 úverových obchodov. V tabuľke je 231 premenných, ktoré nesú informácie o obchode, behaviorálne ukazovatele (platby, delikvenčné polia - dni po splatnosti a príslušné dlžné sumy za rôzne časové intervaly atď.), údaje o klientovi a položky z finančných výkazov klienta. Z charakteru dát sa dá usúdiť, že klienti sú živnostníci a menší podnikatelia, a že tabuľka je určená pre modely behaviorálneho skóringu.

Definícia vysvetľovanej premennej `default_flg` je v zhode s jednou z podmienok daných dokumentom (Basel II, 2006) - obchod je v stave defaultu, ak je aspoň jedna splátka v delikvencii (omeškaní) viac ako 90 dní.

3.1.1 Parametre projektu a segmentácia

V tabuľke je daný deň analýzy (ADT) rovnaký pre všetky pozorovania. Kontrolný interval má dĺžku jeden rok a vzorkový interval dva roky.

Keďže model má odhadovať pravdepodobnosť, že z „dobrého" klienta (v období vzorkového intervalu) sa v priebehu kontrolného intervalu stane „zlý" klient, je potrené vylúčiť tie pozorovania, ktoré sú v stave defaultu k ADT alebo kedykoľvek predtým. Túto informáciu nesie premenná `smp_default_flg`, ktorá odfiltrovala 420 pozorovaní. Podobne sme odstránili obchody zatvorené pred ADT, ďalej tie, ktoré boli zatvorené pred koncom kontrolného intervalu a súčasne neprešli do stavu defaultu počas tohto intervalu. Aby sme mohli pozorovať aspoň niektoré behaviorálne premenné, je potrebné odstrániť pozorovania, ktoré nemajú dostatočnú históriu - ako minimálne trvanie obchodu od jeho otvorenia k ADT sme zvolili obdobie troch mesiacov.

Počty vylúčených pozorovaní sú v tabuľke 3.1.

Podmienka	**Vylúčené**	**Zostatok**
Všetky pozorovania	0	15 007
default vo vzorkovom intervale	420	14 587
zatvorenie pred ADT	79	14 508
zatvorenie pred (ADT + 1 rok) a súčasne „dobrý" v kontrolnom intervale	4 333	10 175
otvorený neskôr ako (ADT - 3 mesiace)	3 650	6 525

Tabuľka 3.1: Počty vylúčených pozorovaní

Kategorická premenná `system_code` obsahuje 5 tried, ktoré určujú typ obchodu (kreditné karty, splátkové obchody, hypotekárne produkty, kontokorent). Keďže sú tieto typy navzájom veľmi odlišné, bolo by vhodné podľa tejto premennej vykonať segmentáciu populácie. Prispieva tomu aj fakt, že viaceré premenné v tabuľke sú definované iba pre niektoré typy obchodu (napr. čerpanie na kreditných kartách), a že podiely „zlých" pozorovaní v jednotlivých typoch sa veľmi líšia. Avšak problémom je, že pre niektoré typy obchodov máme iba veľmi malý počet pozorovaní, resp. malý podiel „zlých" pozorovaní. Riešením z pohľadu banky by mohlo byť zvýšenie počtu pozorovaní *a)* vziať viac dátumov analýzy a tým zahrnúť aj obchody, ktoré sme kvôli uzavretiu pred alebo v rámci kontrolného intervalu vylúčili, alebo *b)*, ak je v pôvodnej tabuľke len vzorka z celkovej populácie, vziať celú túto populáciu. Oba spôsoby sú však pre nás nerealizovateľné.

Aj napriek tomu však budeme vyvíjať modely iba na jednom type obchodu - splátkové obchody, ktorý má najvyššie zastúpenie „zlých" pozorovaní. Po všetkých úpravách sme teda dospeli k vzorke, ktorá obsahuje 2 496 obchodov, z toho

231 zlyhaných, čo je 9.3%-ný podiel. Tento podiel samozrejme nemusí zodpovedať skutočnému podielu zlyhaných obchodov v portfóliu.

3.1.2 Vytváranie nových prediktorov

Z premenných, ktoré sú vo vzorke, je možné vytvoriť ďalšie odvodené premenné, ktoré môžu mať vyššiu predikčnú silu ako pôvodné. Ukážeme si konkrétne príklady, ako sme v softvéri R vytvárali nové premenné:

- rozdiely medzi dátumami, napr. `adt_duration = (adt - open_dt)/(365/12)` je počet mesiacov, ktorý uplynul medzi otvorením obchodu a dátumom analýzy,
- vytvorenie binárnych premenných (nazývané aj *flag*, *dummy premenná*), napr. `execution_flg = (adt - exe_dt < 365)` je flag, ktorý nesie informáciu o tom, či v poslednom roku pred dňom analýzy mal klient exekúciu,
- „zjednotenie" dvoch premenných, napr. tržby dané vo finančných výkazoch `fs_sales = ifelse(is.na(de_sales), se_sales, de_sales)`, pričom v prípade, že klient má vyplnené výkazy jednoduchého aj podvojného účtovníctva, prednosť majú údaje z podvojného účtovníctva,
- ukazovatele z finančných výkazov, napr. `fsr_roa = (fs_net_income/fs_total_assets)` je ROA, teda rentabilita aktív,
- relatívne behaviorálne ukazovatele, napr. `rel_outst_3m = outst_3m/approved_amt` je relatívna dlžná suma po splatnosti v období 3 mesiace pred ADT, teda podiel absolútnej dlžnej sumy a schválenej sumy,
- priemerné výšky splátok, napr. za 3 mesiace pred ADT: `avg_inst_3m = installment_3m/ installment_3m_cnt`.

Prediktory môžu byť odvodené aj pomocou funkcií `sum()` `max()`, `min()`, priemer alebo medián cez viacero príbuzných premenných. Vždy je však dôležité, aby vytvorená premenná bola ekonomicky interpretovateľná.

Podľa uvedených príkladov sme vytvorili približne 40 nových premenných, teda spolu s pôvodnými je po tejto fáze vo vzorke asi 270 premenných. Samozrejme, sú tu ešte premenné, ktoré je potrebné odstrániť. Patria medzi ne:

- premenné, ktoré majú iba jednu hodnotu, prípadne zanedbateľný počet iných hodnôt, napr. `system_code` alebo `currency_code` - iba v dvoch prípadoch je obchod vedený v mene EUR, ostatné majú vyplnený kód SKK,

- premenné, ktoré sú nevyplnené, napr. čerpanie na kreditných kartách alebo kontokorentoch - tieto sa netýkajú splátkových obchodov, preto sú prázdne,
- údaje z podvojného a jednoduchého účtovníctva - pretože sme ich spájali do spoločných premenných,
- všetky premenné pochádzajúce z kontrolného intervalu okrem vysvetľovanej premennej `default_flg`,
- všetky premenné týkajúce sa obdobia 24 mesiacov pred ADT s výnimkou `installment_24m_cnt` a `installment_24m` (počet splátok a ich suma v tomto období), nakoľko sme zistili, že sú totožné s príslušnými premennými z obdobia 12 mesiacov pred ADT,
- identifikátory - obchodu, klienta, finančného výkazu.

To, že zo vzorky odstraňujeme tieto premenné, neznamená, že nie je potrebné zbierať ich v budúcnosti - niektoré z nich sa napríklad podieľajú na výpočte odvodených premenných. Vo vzorke by však boli zbytočné a zhoršovali by celkovú prehľadnosť pri vyvíjaní modelov.

Celú vzorku použijeme pre vývoj modelov, teda nebudeme ju deliť na vývojovú a validačnú vzorku. V prvej kapitole sme spomenuli, že metóda náhodných lesov ako aj podmienených náhodných lesov nepotrebuje na validáciu zvláštnu množinu dát, pretože využíva OOB dáta. V prípade logistickej regresie budeme merať kvalitu modelu prostredníctvom cross-validácie, ktorej techniku popíšeme neskôr.

3.2 Vývoj modelov

3.2.1 Chýbajúce hodnoty

Prvým krokom vo fáze vyvíjania modelov je analýza všetkých premenných vo vzorke. Keďže žiadna z metód, ktoré používame, nepripúšťa[1] chýbajúce hodnoty vo vzorke, je potrebné sa s nimi vysporiadať. V časti 2.7.1 sme spomenuli 4 prístupy k chýbajúcim hodnotám. My budeme postupovať podľa 3. a 4. prístupu.

Pri analýze dát sme zistili, že približne 10% pozorovaní má vždy chýbajúce hodnoty v premenných `product_code` (kód produktu), `employee_cnt` (počet zamestnancov), `marital` (stav), `education` (dosiahnuté vzdelanie) a ďalších.

[1]Hoci sme uviedli, že CART a teda aj náhodné lesy môžu pracovať aj s dátami obsahujúcimi chýbajúce hodnoty, funkcia `randomForest()` ich z technických dôvodov neprijíma.

Všetky tieto premenné pochádzajú pravdepodobne zo žiadosti o úver a z finančných výkazov. Mohli nastať dve možnosti - buď klienti získali produkt bez vyplnenia žiadosti (predschválené obchody) alebo tieto dáta existujú, no v dôsledku nejakej chyby neboli napárované. V prvom prípade by missingy niesli informáciu, preto by bolo vhodnejšie brať ich ako špecifickú hodnotu (3. prístup), v druhom prípade by bolo lepšie nahradiť ich reálnou hodnotou, pretože tieto informácie existujú, no nie sú vo vzorke dostupné (4. prístup). Ak by sme vyvíjali model aplikačného skóringu, bolo by nutné tieto pozorovania zo vzorky vylúčiť. V prípade behaviorálneho skóringu však tieto premenné nehrajú veľmi významnú rolu a keďže majú vyplnené behaviorálne premenné, ponechávame ich vo vzorke.

Vo zvyšných premenných sú chýbajúce hodnoty len vo veľmi málo pozorovaniach (< 0.5%).

Podľa 3. prístupu zahrnieme missingy do modelu tak, že vytvoríme WOE premenné. Túto procedúru sme robili v softvéri SAS® Enterprise Miner a konkrétne v uzle *Interactive Grouping*, ktorý po nastavení parametrov (minimálne 5% pozorovaní v každom atribúte, entropia ako kritérium pre rozdeľovanie premennej na atribúty) sám navrhne delenie každej premennej na atribúty. Tu je však potrebný aj ľudský úsudok - pri kategorických premenných by v jednom atribúte nemali byť kategórie, ktoré nemá zmysel zlučovať - napr. pri premennej `education` by nemalo byť v jednom atribúte základné vzdelanie spolu s vysokoškolským, na druhej strane, stredoškolské a stredoškolské s maturitou je v prípade potreby možné spojiť. Tu treba poznamenať, že nám chýbali niektoré dodatočné informácie, napr. premenná `product_code` obsahuje 15 rôznych kódov, ktorých zmysel nám nebol jasný. Je preto možné, že sme dali do spoločného atribútu napríklad účelový a bezúčelový splátkový obchod. V reálnej praxi by to bol typický príklad nedostatočnej komunikácie medzi špecialistom pre skóring a produktovým manažérom.

Všetky intervalové premenné sme skontrolovali, či spĺňajú logický trend. Pri premenných, kde sme očakávali monotónne klesajúci alebo rastúci vývoj WOE, sme sa snažili upraviť hranice atribútov alebo zmenšiť ich počet. Týmito úpravami sa nám väčšinou znížila informačná hodnota 2.2. Ak sa nám nepodarilo pri niektorej premennej dosiahnuť logický trend, vylúčili sme ju zo vzorky (napr. premenná `avg_inst_24m` alebo `years_in_bank` - koľko rokov je klient zákazníkom banky). Výsledkom bola tabuľka so 104 vysvetľujúcimi WOE premennými (ktorých názov začína reťazcom `w_`) a jednou vysvetľovanou premennou `default_flg`. Budeme ju nazývať *WOE vzorka*.

V ďalšom prístupe sme missingom pripisovali hodnoty na základe matice blízkosti z metódy náhodných lesov, opísanej v časti 1.3.8. V softvéri R slúži na to funkcia `rfImpute()`, pričom sme ako parametre zvolili počet iterácií `iter=5` a počet stromov v náhodnom lese `ntree=300`. Takto sme získali tabuľku, pričom názvy premenných sme nemenili, nakoľko s pôvodnými dátami obsahujúcimi chýbajúce hodnoty ďalej nebudeme pracovať. Túto tabuľku budeme nazývať v ďalšom texte *Imput vzorka*.

3.2.2 Modely pomocou náhodných lesov

V tejto časti sme aplikovali metódu náhodných lesov na obe vzorky dát, pričom sme sa snažili nájsť optimálny parameter `mtry` určujúci počet náhodne vybraných prediktorov pri každom delení. Softvér R umožňuje hľadanie tohto parametra pomocou funkcie `tuneRF()`, avšak táto optimalizuje iba vzhľadom na celkovú klasifikačnú chybu OOB dát. Keďže v skóringových modeloch nepoužívame 0-1 odhady, či klient nezlyhá alebo zlyhá, ale pravdepodobnosť zlyhania, je vhodnejšie hľadať optimálny parameter vzhľadom na štatistiky predstavené v časti 2.7.5.

Pri ladení modelov sme vďaka sledovaniu grafov parciálnych závislostí zistili, že vzorky pre jednotlivé stromy je potrebné vytvárať stratifikovaným náhodným výberom a nie bootstrapom. Prečo je tomu tak, si vysvetlíme na nasledujúcej jednoduchej simulácii.

Simulácia - bootstrap a stratifikovaný výber

Pomocou softvéru R sme vygenerovali vysvetľujúcu binárnu premennú `y` a 4 vysvetľované premenné:

`x1` - intervalová premenná so slabou pozitívnou koreláciou s `y`,

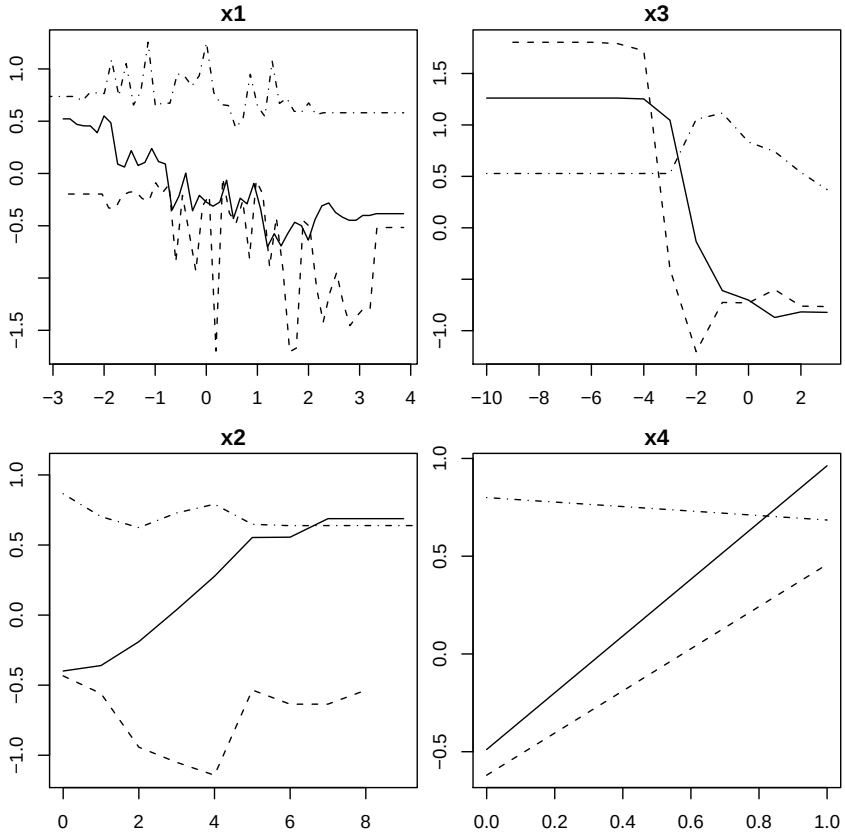
`x2` - ordinálna premenná silne pozitívne korelovaná s `y`,

`x3` - ordinálna premenná so silnou negatívnou koreláciou s `y`,

`x4` - binárna premenná, pozitívne korelovaná s `y` a tiež silne korelovaná s `x1`.

Ďalej sme vytvorili 3 vzorky, ktoré sa líšili zastúpením triedy `y=1`: 10%, 50% a 90%. Označme ich D10, D50 a D90. Na každej z nich sme vyvinuli model pomocou metódy náhodných lesov. Grafy parciálnych závislostí v týchto lesoch a na základe príslušných vzoriek sú na obrázku 3.1.

Z obrázkov možno vidieť, že závislosti od jednotlivých premenných sa v rôznych lesoch veľmi líšia. Napr. v prípade premennej `x2`: na intervale $[0, 4]$ pre

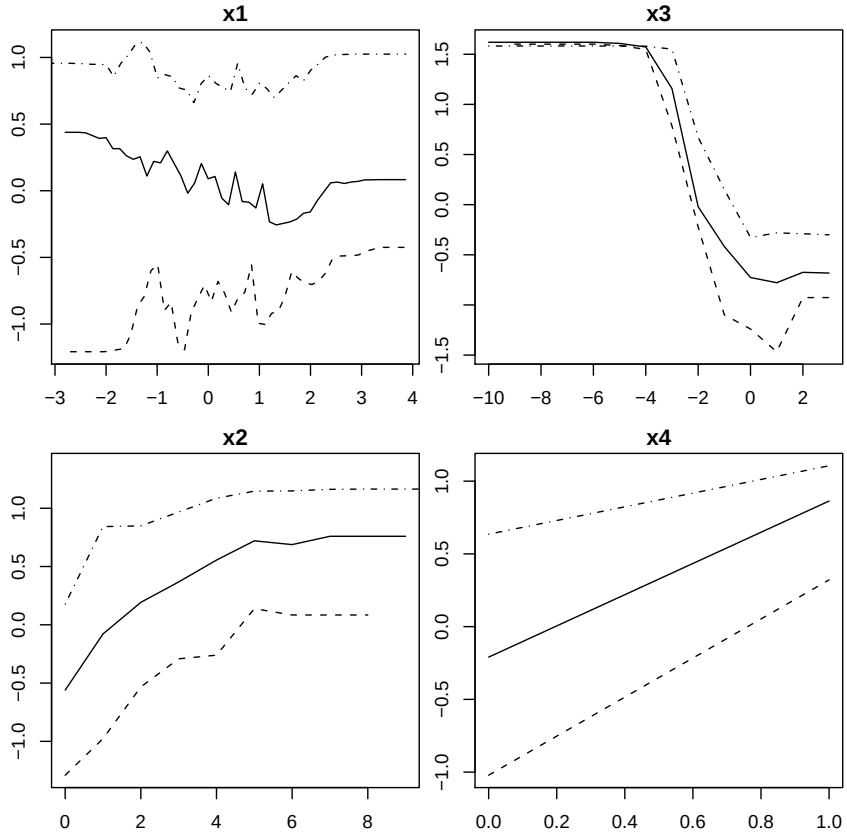


Obrázok 3.1: Parciálne závislosti simulovaných premenných: vzorke D10 zodpovedá čiarkovaná krivka, D50 plná krivka a vzorke D90 bodkočiarkovaná krivka.

vzorku D10 klesá pradepodobnosť, že $y = 1$, pre D50 výrazne rastie a pre vzorku D90 je približne na jednej úrovni. Na základe korelácie by sme očakávali, že táto pravdepodobnosť bude rásť, ako je to v prípade vzorky D50. Podobné rozdiely sú i pre premenné `x3` a `x4`. Zvláštnosťou je závislosť od premennej `x1`. Podľa pozitívnej, hoci slabej korelácie s `y` sme očakávali rast kriviek. Krivky však skôr kolíšu okolo jednej hodnoty (D90) alebo mierne klesajú.

Tento postup sme zopakovali ešte raz, ale s jedným rozdielom. Pri vytváraní vzorky pre každý jeden strom v lesoch sme nepoužili metódu bootstrapu, ale stratifikovaný výber tak, aby sa podiel oboch tried v týchto vzorkách rovnal. To samozrejme znížilo počet pozorovaní v týchto vzorkách, nakoľko môžeme vyberať maximálne toľko pozorovaní, koľko ich je v najmenej početnej triede. Parciálne závislosti (na obrázku 3.2) sa v tomto prípade líšia oveľa menej. Podstatný rozdiel zostáva stále iba v premennej `x1`.

Z tejto simulácie nám vyplýva jedna dôležitá vec: ak použijeme pri vývoji náhodných lesov vzorku, v ktorej je nevyvážené zastúpenie tried binárnej vysvetľu-



Obrázok 3.2: Parciálne závislosti simulovaných premenných: vzorke D10 zodpovedá čiarkovaná krivka, D50 plná krivka a vzorke D90 bodkočiarkovaná krivka.

júcej premennej, a nepoužijeme pritom stratifikovaný náhodný výber do vzoriek pre jednotlivé lesy, vysvetľovaná premenná môže závisieť od vysvetľovanej premennej neočakávaným spôsobom.

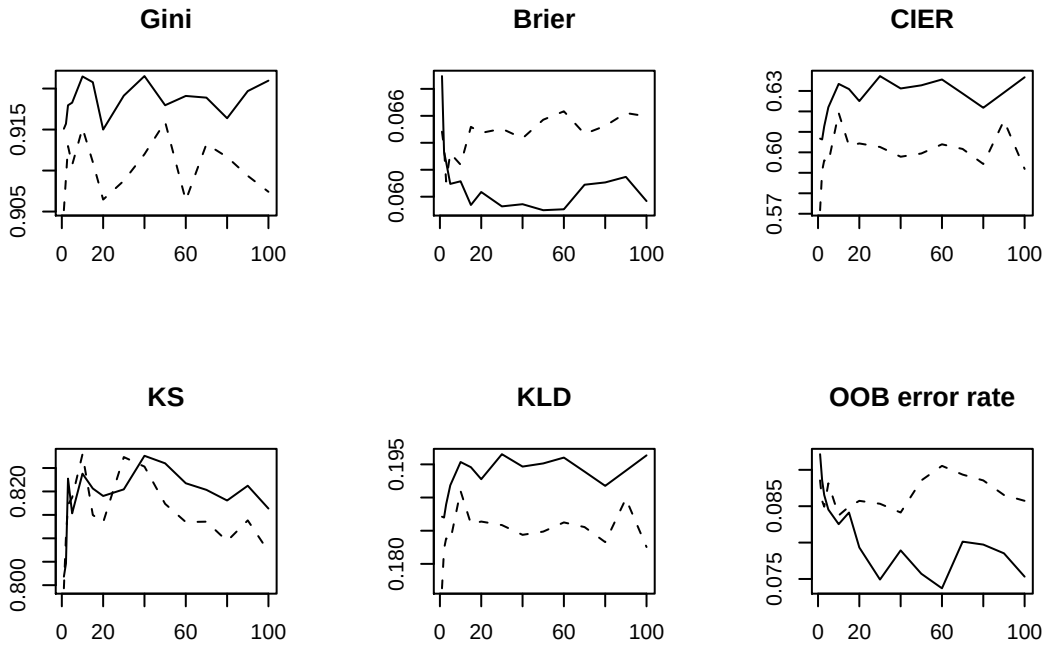
Pre tieto dve trojice náhodných lesov sme porovnali štatistiky. Pre vzorku D50 sa štatistiky oboch lesov takmer vôbec nelíšia, aj napriek zníženiu pozorovaní pri stratifikovanom výbere. Pre vzorky D10 a D90 sa štatistiky použitím stratifikácie mierne zlepšili, až na hodnotu Brierovho skóre.

Ladenie modelov

Na obidvoch vzorkách sme spustili algoritmus náhodných lesov pomocou funkcie `randomForest()` pre rôzne parametre `mtry` a počet stromov `ntree=150`, čo je pomerne málo, no na prvotný pohľad nám postačí. Namiesto bootstrapu sme použili stratifikovaný výber s 200 pozorovaniami pre každú triedu[2]. Vývoj

[2]Vo vzorkách je 231 zlyhaných klientov.

štatistík môžeme vidieť na obrázku 3.3.



Obrázok 3.3: Vývoj štatistík s meniacim sa parametrom `mtry` pre náhodné lesy. WOE vzorke zodpovedajú plné krivky, Imput vzorke čiarkované.

Už na prvý pohľad môžeme vidieť, že modely spúšťané na WOE vzorke sú silnejšie ako modely s rovnakými parametrami vyvinuté na Imput vzorke. Rozdiely však nie sú veľmi výrazné a Gini koeficient nad 90% ako aj Kolmogorovova-Smirnovova štatistika (KS) vyššia ako 80% svedčia o veľmi vysokej predikčnej sile modelov bez ohľadu na parameter `mtry`. Môžeme si všimnúť, že grafy Kullbackovej-Leiblerovej divergencie a pomeru podmienenej informačnej entropie (CIER) sa až na posun a škálovanie zhodujú. Je to v dôsledku toho, že informačná entropia sa vo vzorke nemení. Do štatistík sme zahrnuli aj mieru chybného klasifikovania OOB dát. Táto sa drží v rozmedzí 7 - 9%. V modeloch však nie sú zahrnuté relatívne náklady a cutoff hranica je nastavená na 50%. Ich zmenou by sa táto chyba mohla zmeniť. Celkovo však môžeme povedať, že výsledky nie sú veľmi citlivé na parameter `mtry` (treba si všimnúť škálovanie grafov na obrázku).

Pozrime sa však bližšie na jednotlivé vzorky a konkrétnu voľbu parametra `mtry`. Pre Imput vzorku môžeme vidieť, že vhodný počet prediktorov bude menší ako 40 - hodnoty štatistík (KS, Brierovo skóre) sa od tejto hodnoty vyššie zhoršujú. Vy-

vinuli sme preto ešte raz modely s `mtry` rovným 2, 5, 10, 20 a 40. Tentokrát sme však nechali „vyrásť" v týchto lesoch 1 000 stromov, aby sme získali presnejšie odhady. Hodnoty štatistík týchto model sa líšili len veľmi málo. Gini koeficient bol rovný približne 91.5%, KS štatistika bola na úrovni 81 až 83%. Ukázali sme teda dobrú vlastnosť náhodných lesov - presnosť nie je príliš závislá od voľby parametra `mtry`. Pre ďalšie porovnávanie vezmeme však iba jeden z modelov, a to model s počtom vybraných prediktorov 10, ktorý metóda vyberá defaultne[3]. Model označíme ako RF.I10. Tento budeme ďalej porovnávať s modelmi vyvinutými inými metódami.

Pri WOE vzorke je situácia podobná. Štatistiky sú relatívne stabilné, výnimkou sú len príliš malé hodnoty `mtry`. Analogicky ako pri Imput vzorke, zväčšíme počet stromov na 1 000 pre `mtry` rovné 10, 20 a 40. Štatistiky sa opäť líšia len nepatrne, mierne lepším je model s `mtry` rovným 10. Tento vezmeme pre ďalšie porovnávanie a označíme ho RF.W10.

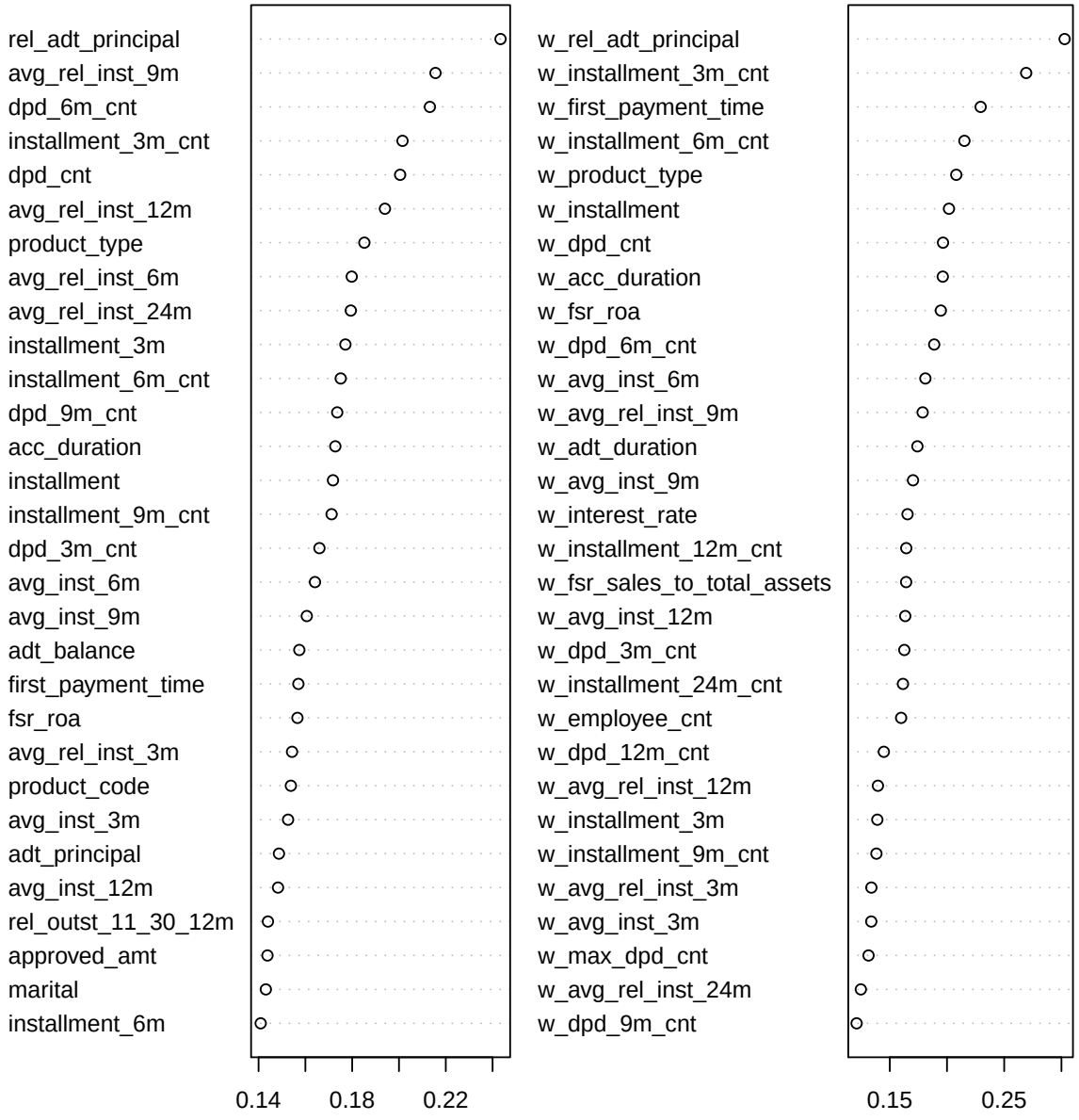
Vedľajším produktom funkcie `randomForest()` je aj významnosť premenných spomenutá v časti 1.3.6. Na obrázku 3.4 je zobrazená permutačná významnosť prvých 30 najvýznamnejších premenných pre modely RF.I10 a RF.W10. Najvýznamnejšou premennou v oboch vzorkách je `(w_)rel_adt_principal` - relatívny zostatok k ADT. Významné sú aj premenné nesúce informáciu o splátkach, informácie o produkte a o trvaní obchodu. Prekvapivým faktom je, že delikvenčné premenné nezaujali prvé miesta, hoci sú veľmi korelované s vysvetľovanou premennou. Vysvetlivky k jednotlivým názvom premenných sú v prílohe 1.

3.2.3 Modely pomocou podmienených náhodných lesov

Pri vývoji týchto modelov sme postupovali podobne ako v predchádzajúcej časti. Vo funkcii `cforest()` sa však nedá jednoducho nastaviť stratifikovaný náhodný výber, preto sme použili obyčajný bootstrap. Taktiež chýba funkcia generujúca grafy parciálnych závislostí, preto sme neoverili, či nevyvážená vzorka nemá podobný vplyv na parciálne závislosti ako je tomu pri náhodných lesoch.

Hodnoty štatistík môžeme vidieť na obrázku 3.5. Ani pri podmienených náhodných lesoch nemá voľba počtu prediktorov pri delení významnú rolu, štatistiky sú horšie len pre veľmi malé hodnoty `mtry`.

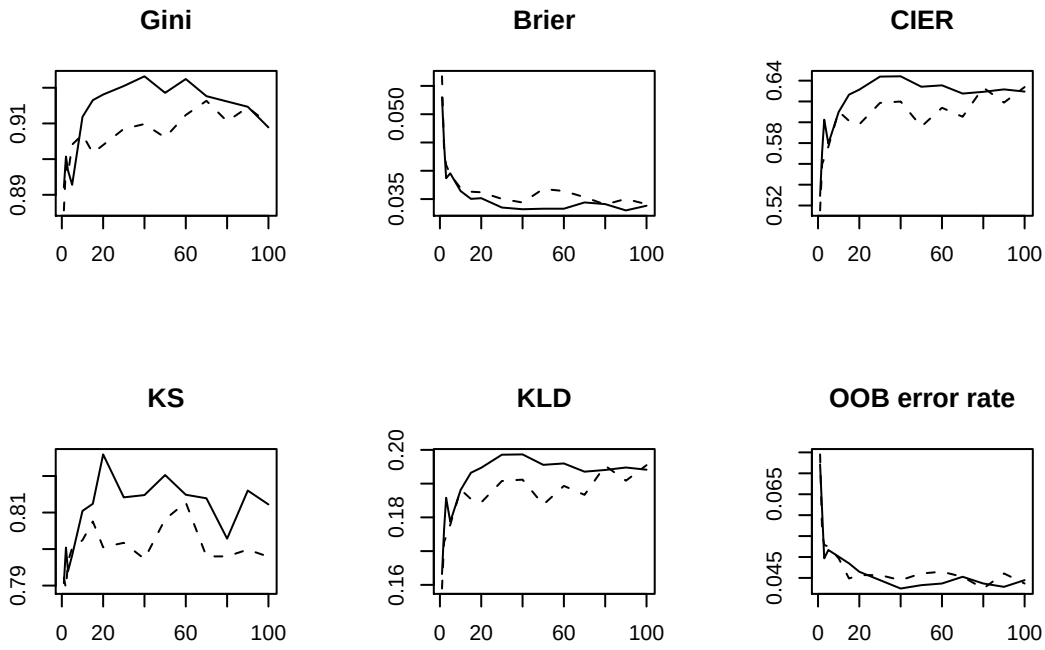
[3]Defaultná hodnota `mtry` pri klasifikácii náhodnými lesmi je druhá odmocnina z počtu prediktorov (zaokrúhlená).



Obrázok 3.4: Permutačná významnosť prediktorov pre modely RF.10 (vľavo) a RF.W10 (vpravo). Uvedených 30 najvýznamnejších prediktorov.

Ďalej sme vyvíjali modely na Imput vzorke pre `mtry` rovné 15, 60 a 70, tentokrát s použitím 500 stromov, pretože algoritmus podmieneného náhodného lesa je výpočtovo náročnejší. V štatistikách opäť nebol žiaden výrazný rozdiel, Gini koeficient bol 90 až 92%, KS štatistika mala 80%. Ako najlepší sme vybrali model s `mtry` = 60, tento model označíme CF.I60.

Pre WOE vzorku sme zvolili hodnoty parametra `mtry` 20, 40 a 60, pričom sme do ďalšieho porovnávania zvolili posledný, CF.W60.



Obrázok 3.5: Vývoj štatistík s meniacim sa parametrom `mtry` pre podmienené náhodné lesy. WOE vzorke zodpovedajú plné krivky, Imput vzorke čiarkované.

V časti 1.5 sme predstavili podmienenú permutačnú významnosť prediktorov, ktorá je na rozdiel od nepodmienenej permutačnej významnosti nevychýlená. V softvéri R slúži na to funkcia `varimp()`. Pôvodne sme chceli túto významnosť počítať priamo pre modely CF.I60 a CF.W60, z technických dôvodov[4] sme však museli ísť inou cestou, postupujúc podľa nasledujúceho algoritmu:

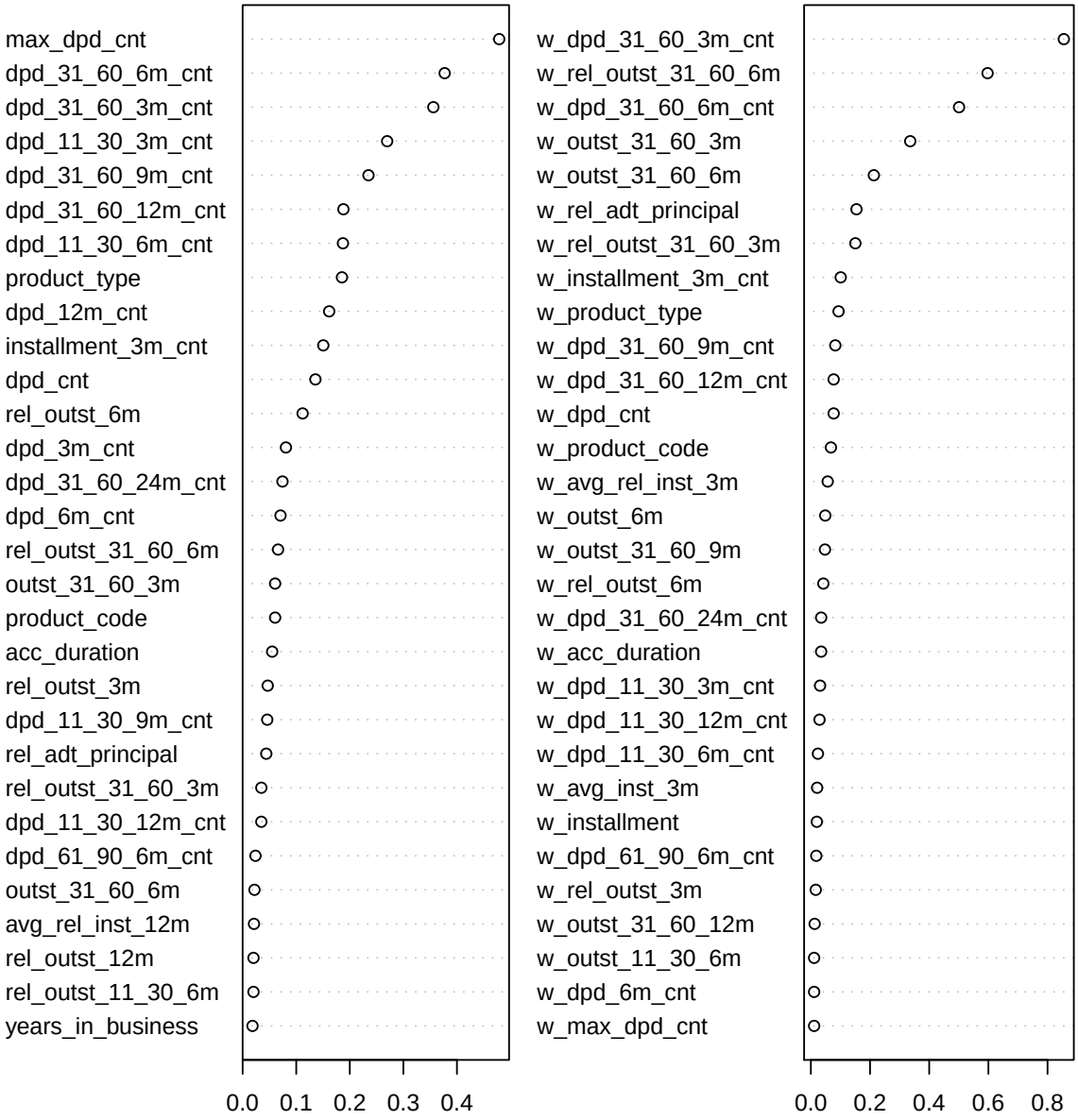
1. Zo vzorky (Imput, resp. WOE) sme náhodne vybrali približne 12% pozorovaní tak, že podiel „zlých" aj „dobrých" pozorovaní bol približne taký istý ako v pôvodnej vzorke (stratifikovaný výber).

2. Na tomto výbere dát sme vytvorili podmienený les s počtom 150 stromov a parametrom `mtry` rovným 60.

3. Vypočítali sme na základe OOB dát tejto malej vzorky podmienenú významnosť prediktorov.

4. Kroky 1. až 3. sme opakovali viackrát (10) a výsledky spriemerovali.

[4]Funkcia sa nedokázala vysporiadať s veľkosťou dát.



Obrázok 3.6: Podmienená významnosť prediktorov pre modely CF.I60 (vľavo) a CF.W60 (vpravo). Uvedených 30 najvýznamnejších prediktorov.

Výsledky pre obe vzorky sú znázornené na obrázku 3.6. Môžeme vidieť, že medzi najvýznamnejšími prediktormi sú, podobne ako pri modeloch RF.I10 a RF.W10 na obrázku 3.4, delikvenčné ukazovatele. Okrem nich sa vyskytujú ešte premenné nesúce informáciu o produkte (`product_code` a `product_type`), informácie o splátkach, dĺžka trvania obchodu `acc_duration` alebo nesplatená časť istiny (`rel_adt_principal`). Ak sa nejaká premenná nachádza v prvej tridsiatke jednej vzorky, väčšinou je aj jej príslušná premenná v tridsiatke pri druhej vzorke,

často sa však ich poradie líši.

3.2.4 Modely pomocou logistickej regresie

Keďže v oboch vzorkách je relatívne veľký počet premenných, je vhodné mať nejaké metódy, ktoré nám pomôžu pri ich výbere do modelu. V softvéri R je funkcia `stepAIC()`, ktorá hľadá submodel zadaného modelu s čo najmenšou hodnotou Akaikeho informačného kritéria. Pre veľké množstvo premenných a pozorovaní sme ju z technických príčin nepoužili. Jednou z možností by mohlo byť použitie premenných, ktoré mali vysokú významnosť (podľa obrázkov 3.4 a 3.6).

Zhlukovanie premenných

Pri analýze premenných nám môže poslúžiť zhlukovanie premenných. Premenné, ktoré sú navzájom korelované, sa spájajú do zhlukov. Algoritmus zhlukovania je navrhnutý nasledovne: na začiatku sú všetky premenné vo svojom zhluku. V každom kroku spojíme do jedného zhluku tie dva zhluky, ktoré sú najbližšie. Blízkosť zhlukov A a B je definovaná ako $\min(\mathrm{cor}(X,Z)^2; X \in A, Z \in B)$, kde $\mathrm{cor}(X,Z)$ je korelačný koeficient medzi premennými X a Z. Na konci sú všetky premenné v jednom zhluku.

Ďalej sme postupovali tak, že v stave, keď boli premenné rozdelené v 25 zhlukoch, z každého sme vybrali premennú najviac korelovanú s vysvetľovanou premennou `default_flg`. Ak táto korelácia bola aspoň 10%, vložili sme premennú do modelu. Aplikovaním tohto postupu na Imput vzorku nám vstúpilo do modelu 11 premenných. Z dôvodu nesignifikantnosti[5] alebo neinterpretovateľnosti sme ďalej niektoré z modelu odstránili. Výsledný model mal 6 premených. Ich zoznam je uvedený v prílohe 2. Tu sú uvedené aj všetky ostatné modely vytvorené pomocou logistickej regresie ako príkazy v softvéri R. Model označíme ako LR.Iz.

Pri meraní štatistík tohto modelu a aj všetkých ostatných založených na logistickej regresii sme využívali techniku cross-validácie. Jej postup sme zvolili takto: vytvorili sme stratifikovaný[6] náhodný výber o veľkosti približne 63% z celkového počtu pozorovaní v pôvodnej vzorke. Táto podvzorka predstavuje trénovaciu množinu dát. Na nej vyvinieme model s použitím hore uvedených premenných. Zvyšné pozorovania sú použité na validáciu. Pomocou ich odhadov

[5]Nesignifikantnosť na hladine významnosti 5%. Túto hladinu sme použili aj v ďalších modeloch, prípadne sme ju podľa potreby mierne zvýšili.

[6]Podiel „zlých" a „dobrých" pozorovaní bol približne rovnaký ako v pôvodnej vzorke.

určíme všetky nami používané štatistiky. Tento postup opakujeme viackrát (50) a výsledky spriemerujeme.

Podiel pozorovaní v trénovacej vzorke sme zvolili 63%, aby sme sa priblížili OOB odhadom. Pri bootstrape je totiž vybrané približne také percento pozorovaní, zvyšok sú OOB dáta.

Model LR.Iz má Gini koeficient 88.7%, KS štatistika je na úrovni 80.5%. Hoci má tento model mierne nižšiu predikčnú silu, treba podotknúť, že vďaka svojej konštrukcii zachytáva rôznorodejšiu informáciu o obchode a klientovi, pretože berie premenné z rôznych zhlukov. Táto vlastnosť sa odrazila aj vo veľkosti zovšeobecnených variančných inflačných faktorov (GVIF) premenných. GVIF používame ako alternatívu k variančnému inflačnému faktoru (VIF) v prípade, že v modeli vystupujú kategorické premenné. Vtedy sa podľa (Fox, 1992) pre porovnávanie berie hodnota $\mathrm{GVIF}^{1/2\mathrm{df}}$, kde df je počet kategórií kategorickej premennej znížený o jednotku. Pre intervalové premenné sa berie $\mathrm{df} = 1$. V modeli LR.Iz boli všetky hodnoty $\mathrm{GVIF}^{1/2\mathrm{df}}$ menšie ako 1.3.

Technikou zhlukovania premenných a následnými úpravami sme vyvinuli na WOE vzorke model LR.Wz, ktorý obsahuje 6 premenných. V súlade s ekonomickou interpretáciou musí byť odhadnutý koeficient každej zúčastnenej premennej záporný, čo je v tomto modeli splnené. Cross-validáciou sme určili Gini koeficient 89% a KS štatistiku 81%. VIF pre všetky premenné bol menší ako 1.3.

Krokové metódy

V softvéri SAS® Enterprise Miner sú vyvinuté tri metódy, ktoré určujú výber parametrov podľa vopred zvoleného kritéria. Ak zvolíme ako kritérium AIC, potom sú tieto metódy podobné funkcii `stepAIC()`. Rozdiel je však v tom, že neprehľadávajú všetky submodely modelu, ale v postupných krokoch pridávajú alebo odoberajú premenné na základe ich p hodnoty. Táto hodnota je pravdepodobnosť platnosti hypotézy, že koeficient pri danej premennej je nulový. Jednou z možností je nepoužiť žiadne kritérium výberu, vtedy metódy vrátia model z posledného kroku ich algoritmu. Metóda *backward* (spätná) zahrnie na začiatku všetky premenné do modelu a postupne eliminuje tie premenné, ktorých p hodnota je vyššia ako vopred zvolená hranica, tzv. *Stay value*, napr. 5%. Opačne postupuje *forward* (dopredná) metóda, ktorá postupne do modelu pridáva premenné, ktorých p hodnota je nižšia ako ďalšia zvolená hranica, tzv. *Entry value*. Tretia z metód, *stepwise*, začína tak ako *forward*, no kombinuje obe predošlé metódy - odstraňuje premenné s p hodnotou vyššou ako *Stay value* a súčasne pridáva premenné s p hodnotou

nižšou ako *Entry value*. Pre oba typy hraníc sme použili hodnotu 5%. Pri výbere modelu sme nepoužili v týchto metódach žiadne kritérium.

Aplikovaním metódy *stepwise* na Imput vzorku sme získali model s 13 premennými, avšak 5 z nich boli delikvenčné ukazovatele, čo sa prejavilo na vysokých hodnotách $\mathrm{GVIF}^{1/2\mathrm{df}}$ (rovné približne 3, čo zodpovedá VIF ≈ 9 pre intervalové premenné). Preto sme postupne odoberali premenné s najvyšším GVIF, čím niektoré premenné prestali byť signifikantné. Finálny model s označením LR.Is obsahoval 8 premenných.

Metódou *forward* sme získali ten istý model ako použitím metódy *stepwise*. Posledná z metód, *backward*, nám vrátila model s 27 premennými, z ktorých až 19 bolo delikvenčných. Podobnými úpravami ako pri predchádzajúcej metóde sme získali model so 7 premennými. Maximálny VIF bol približne 1.5. Model sme nazvali LR.Ib.

Pri WOE vzorke sme aplikovaním *forward* metódy získali model so 16 premennými. Dve z nich, konkrétne `w_avg_rel_inst_3m` a `w_installment_freq`, sme museli z modelu odstrániť, pretože ich koeficienty mali kladné znamienko. Kvôli pomerne vysokej hodnote VIF (> 3) sme odstránili aj premennú `w_dpd_cnt`. Nakoniec sme z modelu odstránili premennú `w_rel_outst_31_60_6m`, ktorá bola nesignifikantná. Výsledný model LR.Ws mal 12 premenných.

Podobnými úpravami modelov získaných pomocou metód *forward* a *backward* sme vyvinuli modely LR.Wf, resp. LR.Wb.

3.3 Porovnanie modelov

Štatistiky všetkých modelov, ktoré sme vybrali, sú uvedené v tabuľke 3.2. Poďme sa bližšie pozrieť najprv na modely vyvinuté na Imput vzorke. Môžeme tvrdiť, že vzhľadom na hodnotu Gini koeficientu majú modely vyvinuté pomocou lesov mierne vyššiu predikčnú silu ako modely založené na logistickej regresii. Ostatné štatistiky však tvrdia opak - hodnoty KS štatistiky, Brierovho skóre i mier založených na entropii sú lepšie pre modely logistických regresií. Spomedzi týchto sa najslabším zdá byť model LR.Iz, ktorý sme vyvinuli pomocou metódy zhlukovania premenných. Každopádne, rozdiely v štatistikách sú veľmi malé a hociktorý z modelov má vysokú predikčnú silu.

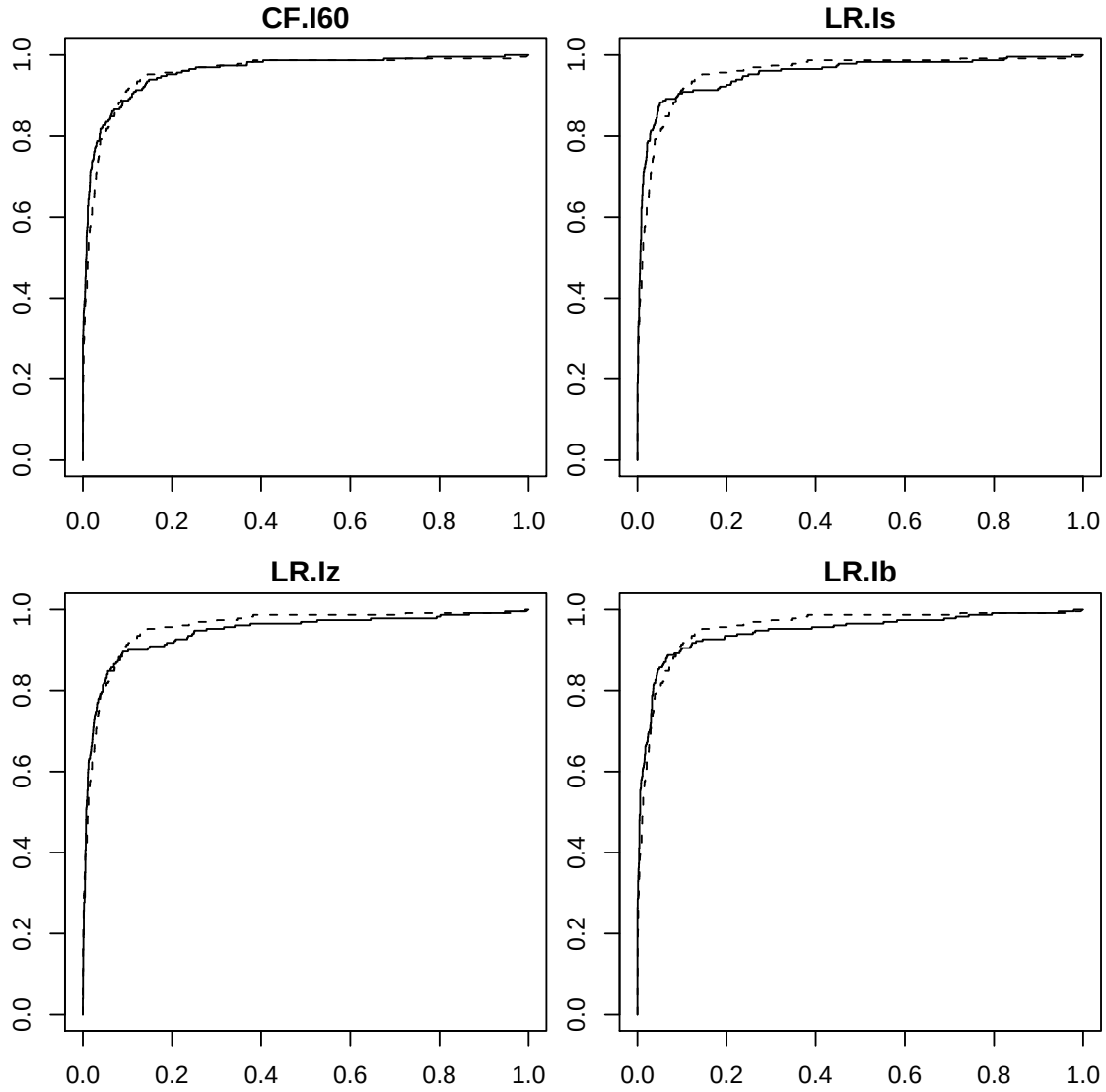
Pri výbere modelu nám ešte môžu pomôcť ROC krivky znázornené na obrázku 3.7. Pripomeňme, že ROC krivka vzniká nanášaním kumalatívneho percenta ne-

Model	Gini	KS	Brier	KLD	CIER
RF.I10	0.9132	0.8158	0.0637	0.1854	0.6010
CF.I60	0.9193	0.7978	0.0352	0.1893	0.6137
LR.Iz	0.8868	0.8052	0.0384	0.2004	0.6477
LR.Is	0.9095	0.8361	0.0338	0.2156	0.6968
LR.Ib	0.8929	0.8220	0.0363	0.2097	0.6780
RF.W10	0.9236	0.8268	0.0597	0.1983	0.6429
CF.W60	0.9211	0.8170	0.0331	0.1973	0.6398
LR.Wz	0.8894	0.8049	0.0397	0.2004	0.6476
LR.Ws	0.9265	0.8391	0.0288	0.2275	0.7353
LR.Wf	0.9278	0.8381	0.0291	0.2266	0.7326
LR.Wb	0.9273	0.8431	0.0295	0.2283	0.7379

Tabuľka 3.2: Štatistiky finálnych modelov

zlyhaných obchodov (vodorovná os) ku kumulatívnemu percentu zlyhaných obchodov (zvislá os) vzhľadom na meniace sa skóre (v našom prípade odhadnutá pravdepodobnosť defaultu PD). Pre porovnanie je na každom grafe čiarkovane ROC krivka modelu RF.I10. Vidíme, že ROC krivky modelov vyvinutých pomocou lesov sú veľmi podobné (CF.I60). Pre zvyšné modely však môžeme vidieť určité odlišnosti. Model LR.Iz po určitú hladinu PD tiež presne kopíruje krivku modelu RF.I10. Približne 90% zlyhaných a 10% nezlyhaných obchodov má PD vyššie ako táto hladina. Od tohto bodu sa však rast krivky LR.Iz znižuje, čo znamená, že ďalším znižovaním hladiny PD pribúdajú zvyšné zlyhané obchody len pozvoľne. Týmto možno vysvetliť 2.5%-ný rozdiel v Gini koeficientoch týchto modelov. Model LR.Is má zo začiatku ROC krivku dokonca strmšiu ako RF.I10, teda lepšie rozlišuje medzi „dobrými" a „zlými", podobne však v približne tej istej hladine PD zmierni tempo rastu.

Pre WOE vzorku je situácia mierne odlišná. Modely vyvinuté pomocou logistickej regresie majú až na LR.Wz približne rovnakú hodnotu Gini koeficientu a lepšie hodnoty ostatných štatistík. Môžeme to zdôvodniť tým, že táto vzorka bola vyvinutá špeciálne pre potreby logistickej regresie. Z intervalových premenných, ktoré nadobúdajú spravidla mnoho hodnôt, sme vytvorili WOE premenné majúce len toľko hodnôt, koľko obsahujú atribútov, teda približne 3 až 6. Zaujímavosťou však je, že hoci pri delení podľa WOE premenných mali stromy omnoho menej voľnosti ako pri delení podľa pôvodných premenných, predikčná sila oboch lesov na Imput vzorke je mierne nižšia ako na WOE vzorke. Navyše možno predpokla-

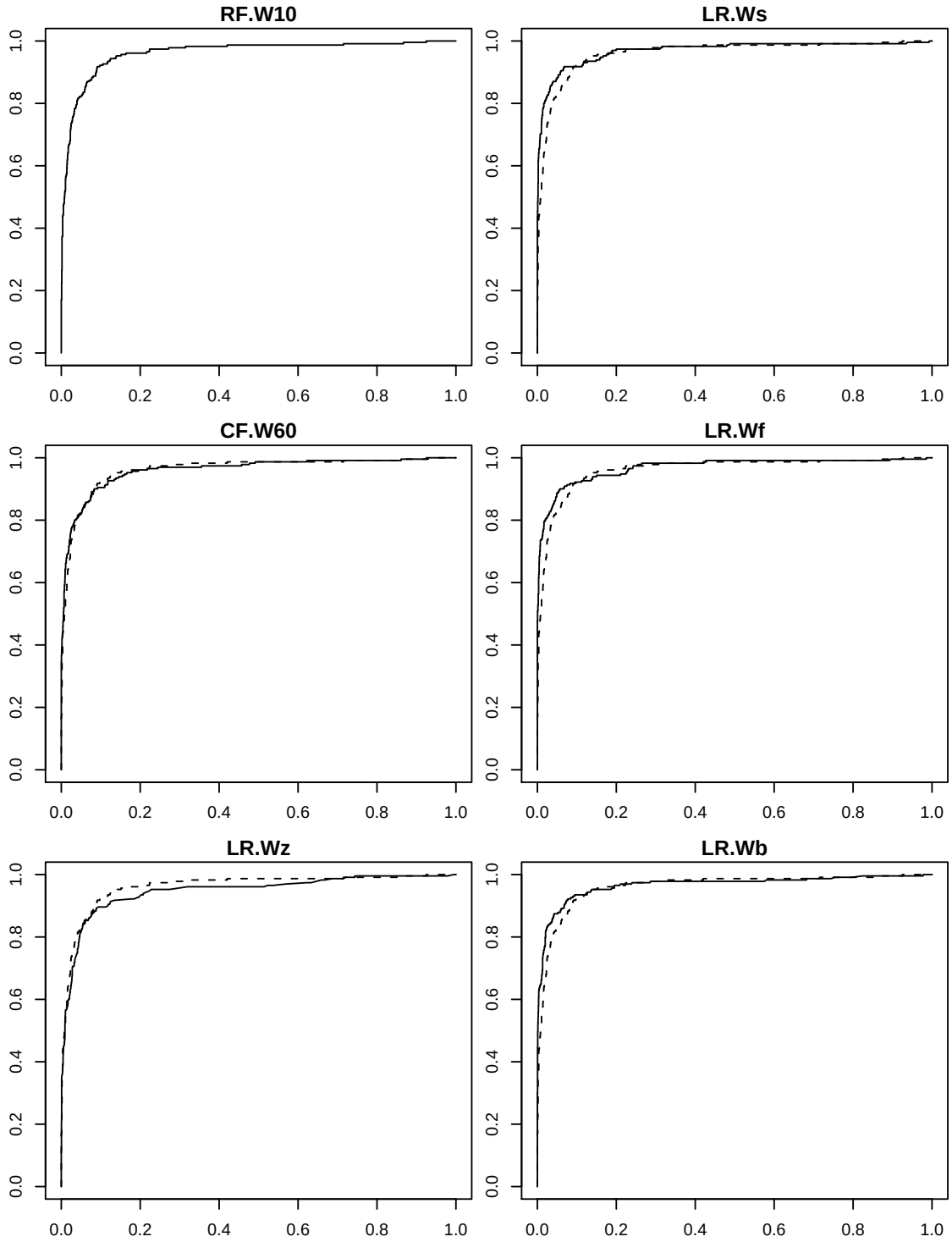


Obrázok 3.7: Porovnanie ROC krivky modelu RF.I10 (čiarkovaná krivka) s krivkami ostatných modelov vyvinutých na Imput vzorke.

dať, že delením podľa WOE premenných získame stabilnejšie výsledky, keďže každá z nich spĺňa logický trend.

Na obrázku 3.8 sú ROC krivky. Opäť môžeme pozorovať podobnosť kriviek pre modely vyvinuté pomocou lesov (CF.W60). Krivka modelu LR.Wb takmer celá leží nad krivkou RF.W10 alebo sa jej dotýka.

Na základe štatistík sa nedá jednoznačne určiť najlepší model. Pre Imput vzorku dosahuje najvyšší Gini koeficient model CF.I60, no v ostatných štatistikách je horší. Tento model by sme však ako finálny nevolili, pretože o ňom nemáme dostatočné informácie (grafy parciálnych závislostí). Spomedzi modelov vyvinutých na WOE



Obrázok 3.8: ROC krivky modelov vyvinutých na WOE vzorke. Pre porovnanie je čiarkovane zobrazená ROC krivka modelu RF.W10.

vzorke sú najlepšie modely LR.Ws, LR.Wf a LR.Wb. Popri štatistikách je však samozrejme potrebné zvážiť i ďalšie faktory a to najmä aké konkrétne premenné

vstupujú do modelov. Čím vyšší je počet rôznorodých premenných v modeli, tým širší rizikový profil klienta tento model zachytí. V lesoch sa pri predikcii má šancu zúčastniť každá premenná, čo je jeho veľkou výhodou. Vtedy je však potrebné skontrolovať, aký vplyv má každá konkrétna premenná na vysvetľovanú premennú.

3.4 Parciálne závislosti

Viackrát sme spomenuli, že pre banku je okrem predikčnej sily modelu dôležité dostatočne dobre vysvetliť, ako jednotlivé premenné prispievajú pri predikovaní vysvetľovanej premennej v celkovom modeli. V dobre zostrojených modeloch založených na logistickej regresii je tento problém jednoducho riešiteľný. Ak sú navyše použité WOE premenné, môžeme model zapísať do prehľadnej skórkarty.

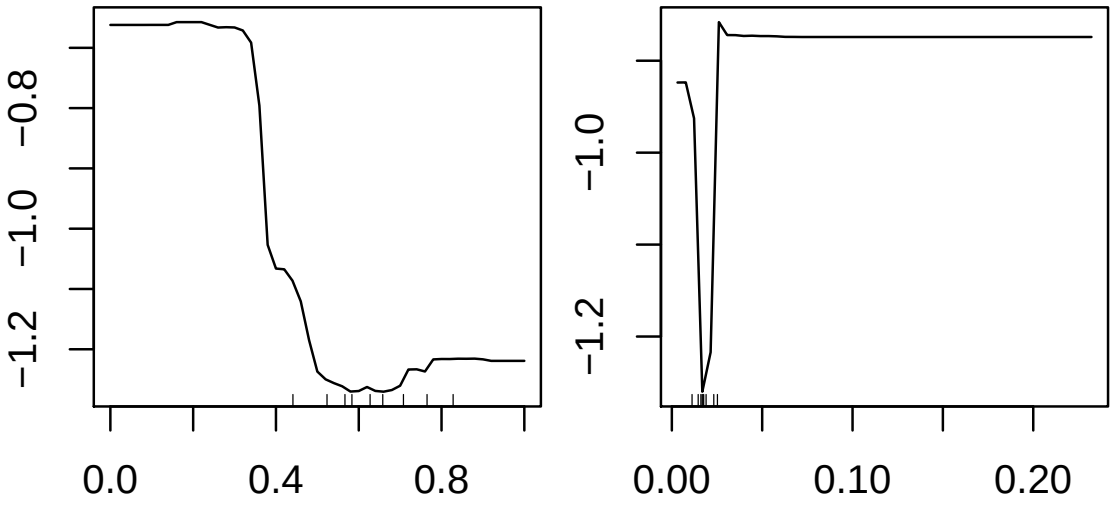
Pri náhodných a podmienených náhodných lesoch to také jednoduché nie je. V časti 1.3.7 sme popísali metódu, ktorá nám môže pomôcť odhaliť, ako vysvetľovaná premenná `default_flg` závisí od nejakej zvolenej premennej. Práve vďaka nej sme zistili, že je potrebné vyvíjať modely náhodných lesov pomocou stratifikovaného výberu. Ako sme spomenuli, v súčasnosti v softvéri R nie je funkcia, ktorá by generovala grafy parciálnych závislostí pre podmienené náhodné lesy. Vytvorili sme ich preto len pre náhodné lesy pomocou funkcie `partialPlot()`. Konkrétne sme použili model RF.I10.

V práci uvádzame len niektoré z premenných, pričom sme ich vyberali podľa významnosti (obrázok 3.4) a podľa toho, či už podobná premenná nie je zobrazená.

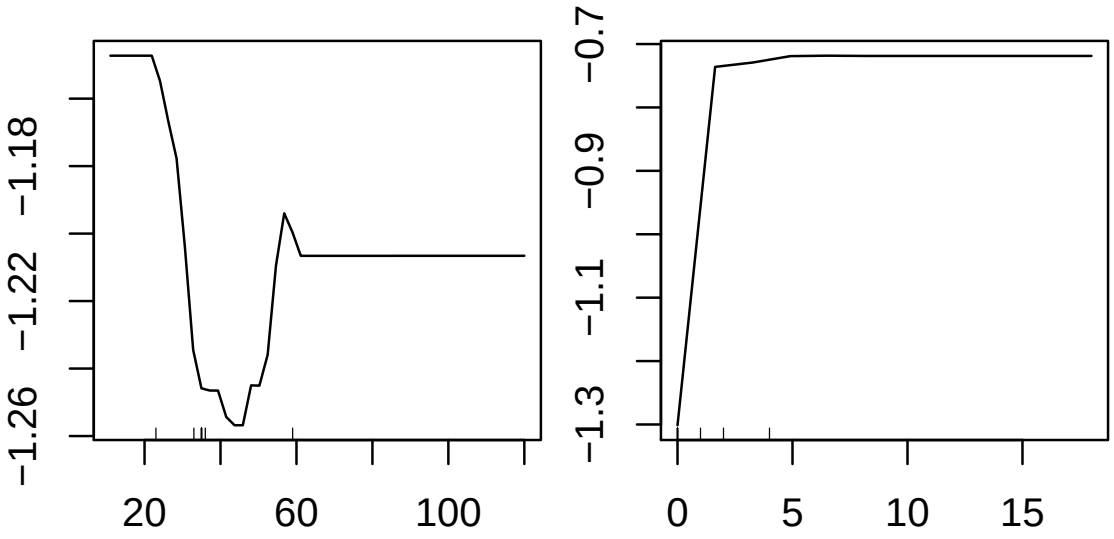
Na obrázku 3.9 vľavo je zobrazená závislosť vysvetľovanej premennej od relatívneho zostatku k ADT, t.j. pomeru nesplatenej časti k schválenej sume. Pri tomto obrázku sme si uvedomili, že táto závislosť je umelo vytvorená voľbou vzorky. Kvôli vyššiemu počtu zlyhaných obchodov vo vzorke sme totiž vložili tie obchody, ktoré zlyhali v kontrolnom intervale a súčasne boli zatvorené pred koncom kontrolného intervalu. Je zrejmé, že tieto obchody mali už väčšiu časť poskytnutej sumy splatenú. Vďaka tomu sú takmer splatené obchody (s viac ako dvomi tretinami celkovej poskytnutej sumy) v lese považované za veľmi rizikové. Treba poznamenať, že táto premenná by mala byť zo vzorky odstránená, pretože jej predikčnú silu spôsobuje konštrukcia vzorky. Každopádne, tento príklad je aspoň ukážkou toho, že každá premenná vstupujúca do modelu by mala byť dôsledne skontrolovaná a že užitočným nástrojom je práve graf parciálnej závislosti.

Druhý graf dáva informáciu o závislosti od priemernej splátky za posledných

9 mesiacov, takisto delenú schválenou sumou obchodu. Na obrázkoch sú na vodorovnej osi zaznačené decily premennej. Vidíme, že väčšina klientov mala túto priemernú splátku na úrovni 5-10% poskytnutej sumy. Títo sú menej rizikoví ako



Obrázok 3.9: Grafy parciálnej závislosti od premenných `rel_adt_principal` (vľavo) a `avg_rel_inst_9m` (vpravo).



Obrázok 3.10: Grafy parciálnej závislosti od premenných `acc_duration` (vľavo) a `dpd_6m_cnt` (vpravo).

klienti s extrémnymi hodnotami. Veľmi nízke hodnoty môžu znamenať obchody s dlhým trvaním, kedy sa úver spláca viac rokov menšími čiastkami. Keďže sa jedná o portfólio menších podnikateľov, je možné, že títo klienti si stanovili dlhoročný podnikateľský plán a hoci naň dostali úver, je pravdepodobné, že zlyhajú skôr ako klienti s podnikateľským plánom v menšom časovom horizonte. Opačný extrém predstavuje úvery s krátkym trvaním a s nižšou sumou. O tieto majú záujem skôr živnostníci s menším obratom (alebo začínajúci), ktorí sú pravdepodobne rizikovejší ako podnikatelia s vyšším obratom.

Naše predpoklady o vzťahu medzi dĺžkou úveru a pravdepodobnosťou zlyhania sa potvrdili v grafe parciálnej závislosti od premennej `acc_duration` na obrázku 3.10. Tu navyše vidíme, že krátkodobé úvery sú rizikovejšie ako dlhodobé. Úvery s dĺžkou 3 až 4 roky sú najmenej rizikové.

Druhým grafom je závislosť od delikvenčnej premennej `dpd_6m_cnt`, teda počte delikvencií za posledných 6 mesiacov pred ADT. Klient bez delikvencií je prirodzene menej rizikový. Avšak zaujímavý je fakt, že či mal klient 2 alebo viac delikvencií, je rizikový rovnako - model takmer vôbec nerozlišuje medzi takýmito prípadmi.

Grafy parciálnych závislostí pre WOE premenné v modeli si nebudeme zobrazovať, nakoľko nie sú veľmi zaujímavé. Tiež je však vhodné skontrolovať, či sú krivky týchto závislostí klesajúce, nakoľko čím nižšia je hodnota WOE atribútu, tým rizikovejšie je pozorovanie.

Záver

Cieľom tejto diplomovej práce bolo predstaviť metódu náhodných lesov a aplikovať ju v kreditnom skóringu na reálnych dátach z bankového prostredia.

V prvej kapitole sme definovali všeobecný klasifikačný problém. Aby sme mohli pokračovať predstavením metódy náhodných lesov, bolo potrebné oboznámiť sa s ich základnou stavebnou jednotkou - klasifikačnými a regresnými stromami - CART. Popri algoritme náhodných lesov sme sa venovali aj problematike odhadu sily náhodných lesov a chyby zo zovšeobecnenia pomocou out-of-bag pozorovaní.

Náhodné lesy nie sú iba klasifikátor - uviedli sme si algoritmy viacerých vedľajších produktov, ktoré umožňujú merať významnosť prediktorov v modeli, určiť, ako vysvetľujúca premenná závisí od konkrétneho prediktora alebo priradiť chýbajúcim poliam reálne hodnoty. Väčšinu z týchto techník sme aj v praxi použili.

Keďže CART je vychýlený pri výbere deliaceho prediktora a toto vychýlenie sa prenáša aj do náhodných lesov, predstavili sme aj ich nevychýlené alternatívy - podmienené stromy a podmienené náhodné lesy.

V ďalšej kapitole sme sa venovali procesu vývoja skórovacieho modelu. Popísali sme jeho jednotlivé fázy od stanovenia cieľov cez prípravu dát až po vývoj a monitorovanie modelu. Jedným z najdôležitejších použití kreditného skóringu je počítanie rizikovo vážených aktív, podľa ktorých banka nastavuje svoje minimálne kapitálové požiadavky. Koná tak v súlade s dokumentami vydanými Bazilejským výborom pre bankový dohľad, najmä Bazilejskou dohodou (Basel II), ktorú sme v krátkosti tiež predstavili.

Nadobudnuté vedomosti sme využili v poslednej časti práce. Z tabuľky reálnych dát pripravených pre potreby behaviorálneho kreditného skóringu sme vytvorili vývojovú vzorku pre jeden typ obchodu - splátkové úvery malých podnikateľov a živnostníkov. Z pôvodných premenných sme odvodili ďalšie premenné, aby sme získali čo najviac informácií o konkrétnom obchode a klientovi. Naopak,

premenné, ktoré z rôznych dôvodov nebolo možné zahrnúť do vzorky, sme odstránili. Keďže v niektorých premenných sa vyskytovali nevyplnené hodnoty, použili sme na ich nahradenie dve metódy. Prvá bola založená na matici blízkosti z metódy náhodných lesov. Druhá rozdelila obor hodnôt každej premennej na atribúty, ktorým sme priradili váhy (WOE) v závislosti od šance, že obchod s týmto atribútom zlyhá. Týmto sme získali dve vzorky, na ktoré sme postupne aplikovali obe metódy náhodných aj podmienených náhodných lesov. Taktiež sme kvôli porovnaniu vyvinuli modely pomocou logistickej regresie, ktoré sú v súčasnosti v praxi asi najpoužívanejšie.

Pri vývoji modelov pomocou náhodných lesov sme zistili prekvapujúcu vec. Ak použijeme pri tvorbe trénovacích vzoriek pre každý strom v lese náhodný výber pomocou bootstrapu, model síce má vysokú predikčnú silu, no závislosti vysvetľujúcej premennej od jednotlivých prediktorov sa správajú proti logickej interpretácii. Problém bol v nevyváženosti zlyhaných a nezlyhaných obchodov vo vzorke. Použitím stratifikovaného výberu z dát s rovnakým počtom pozorovaní v oboch triedach sme dosiahli model, ktorého prediktory vplývali na vysvetľujúcu premennú podľa očakávania. Takisto sa výrazne zmenila permutačná významnosť premenných. Toto zistenie sme sa snažili demonštrovať aj jednoduchou simuláciou.

Keď sme pri vyvíjaní modelov pomocou podmienených lesov hľadali optimálny parameter určujúci počet náhodne vybraných prediktorov pri delení, prišli sme k pomerne vysokej hodnote 60 z približne 100 prediktorov pre obe vzorky. Väčšinou je totiž postačujúci počet rovnajúci sa približne odmocnine z celkového počtu prediktorov, teda 10, ako tomu bolo aj v prípade náhodných lesov pre obe vzorky.

Treba poznamenať, že pri výpočte podmienenej permutačnej významnosti prediktorov v podmienených lesoch sme mali technické problémy, ktoré sme sa snažili obísť výpočtovo menej náročným postupom. Vo funkcii vytvárajúcej podmienené lesy navyše chýba možnosť stratifikovaného výberu, je teda možné, že sa správajú podobne ako náhodné lesy. Parciálnu závislosť prediktorov sme však neoverili, pretože táto funkcia pre podmienené lesy absentuje. Dokiaľ sa neaktualizuje balík s funkciami podmienených lesov, odporúčame použiť v softvéri R radšej náhodné lesy, ktoré majú vyvinuté aj rôzne iné užitočné funkcie.

Na základe logistickej regresie sme vyvinuli viacero modelov, čím sme načrtli niektoré z techník pre vysporiadanie sa s veľkým množstvom prediktorov vo vzorke. Tu sme popri softvéri R využili aj softvér SAS® Enterprise Miner.

Predikčnú silu modelov sme porovnávali využitím viacerých štatistík, ktoré boli navrhnuté Bazilejským výborom pre bankový dohľad. Celkovo môžeme na základe hodnôt týchto štatistických mier vyhlásiť, že všetky modely na oboch vzorkách dosiahli vysokú predikčnú silu a rozdiely neboli veľmi výrazné. Možno by bolo zaujímavejšie vyvinúť a porovnať modely pre aplikačný skóring, ktoré zvyčajne dosahujú výrazne nižšiu predikčnú silu. Aplikačné premenné sú totiž spravidla slabšie ako behaviorálne premenné. Sila jednotlivých modelov by sa mohla líšiť výraznejšie.

Nakoniec sme analyzovali parciálne závislosti vysvetľovanej premennej od niektorých vybraných prediktorov. Snažili sme sa taktiež o ich ekonomickú interpretáciu.

Literatúra

Basel Committee on Banking Supervision (1988). „Basel I: International Convergence of Capital Measurement and Capital Standards." *Bank for International Settlements.* URL http://www.bis.org/publ/bcbs04a.pdf

Basel Committee on Banking Supervision (2006). „Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework - Comprehensive Version." *Bank for International Settlements.* URL http://www.bis.org/publ/bcbs128.pdf

Basel Committee on Banking Supervision (2005). „Working Paper No. 14 - Studies on the Validation of Internal Rating Systems." *Bank for International Settlements.* URL http://www.bis.org/publ/bcbs_wp14.pdf

Berk, R. (2008). „Statistical learning from a regression perspective." *Springer.*

Breiman, L., Friedman, J., Olsen. R., Stone, C. (1984). „Classification and regression trees." *Wadsworth International (California).*

Breiman, L. (2001). „Random Forests." *Machine Learning* **45**, 5-32. URL http://www.springerlink.com/content/u0p06167n6173512/fulltext.pdf

Bylander, T. (2002). „Estimating Generalization Error on Two-Class Datasets Using Out-of-Bag Estimates." *Machine Learning* **48**, 287-297. URL http://www.cs.utsa.edu/~bylander/pubs/ml00-final.pdf

Fox, J., Monette, G. (1992). „Generalized Collinearity Diagnostics." *Journal of the American Statistical Association* **87**(417), 178-183. URL http://www.jstor.org/stable/2290467

Friedman, J.H. (2001). „Greedy Function Approximation: A Gradient Boosting Machine." *The Annals of Statistics* **29**, 1189-1232. URL http://www.salfordsystems.com/doc/GreedyFuncApproxSS.pdf

Hothorn T., Hornik K., Zeileis A. (2006a). „Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics* **15**(3), 651-674. URL http://statmath.wu-wien.ac.at/~zeileis/papers/Hothorn+Hornik+Zeileis-2006.pdf

Hothorn, T., Hornik, K., Wiel, MA., Zeileis, A. (2006b). „A Lego System for Conditional Inference." *The American Statistician* **60**, 257-263. URL http://statmath.wu.ac.at/~zeileis/papers/Hothorn+Hornik+VanDeWiel-2006.pdf

Siddiqi, N. (2006). „Credit Risk Scorecards. Developing and implementing inteligent credit scoring." *John Wiley & Sons, Inc.*

Strobl, C., Boulesteix, A.L., Kneib, T., Augustin, T., Zeileis, A- (2008). „Conditional variable importance for Random Forests." *BMC Bioinformatics* **9:307**. URL http://www.biomedcentral.com/1471-2105/9/307/

Strobl, C., Hothorn, T., Zeileis, A. (2009). „Party on! A New, Conditional Variable Importance Measure for Random Forests Available in the party Package." *The R Journal* **1/2**, 14-17. URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Strobl~et~al.pdf

Príloha 1

Zoznam použitých premenných s vysvetleným významom.

adt_balance	nesplatené časti všetkých splátkových úverov klienta
adt_principal	nesplatená časť k ADT
acc_duration	dĺžka obchodu v mesiacoch
adt_duration	trvanie obchodu k ADT v mesiacoch
approved_amt	schválená suma
avg_inst_Xm	priemer splátok za posledných X mesiacov pred ADT
avg_rel_inst_Xm	pomer avg_inst_Xm k schválenej sume
bank_services	iné služby bánk
default_flg	vysvetľujúca binárna premenná
dpd_A_B_Xm_cnt	počet delikvencií trvajúcich A až B dní za posledných X mesiacov pred ADT
dpd_Xm_cnt	počet delikvencií za posledných X mesiacov pred ADT
dpd_cnt	počet dní v delikvencii pred ADT
education	najvyššie dosiahnuté vzdelanie
employee_cnt	počet zamestnancov
execution_6m_flg	mal klient za posledných 6 mesiacov exekúciu?
execution_flg	mal klient za posledný rok exekúciu?
first_payment_time	počet mesiacov medzi dátumom prvej splátky a ADT
fs_current_assets	bežné aktíva
fs_sales	tržby
fsr_roa	rentabilita aktív
fsr_sales_tot_assets	pomer tržieb k celkovým aktívam
installment	výška splátky
installment_Xm	suma splátok za posledných X mesiacov pred ADT
installment_Xm_cnt	počet splátok za posledných X mesiacov pred ADT
installment_freq	kód frekvencie splátok
interest_rate	úroková sadzba
legal_form	kód právnej formy
marital	stav
max_dpd_cnt	počet dni najdlhšej delikvencie pred ADT
nace_sector	kód typu ekonomickej činnosti
outst_A_B_Xm	celková dlžná suma príslušná k dpd_A_B_Xm_cnt
outst_Xm	celková dlžná suma príslúchajúca k dpd_Xm_cnt
product_code	kód produktu
product_type	typ produktu
rel_adt_principal	pomer nesplatenej časti k ADT k schválenej sume
rel_outst_A_B_Xm	pomer outst_A_B_Xm k schválenej sume
rel_outst_Xm	pomer outst_Xm k schválenej sume

Príloha 2

Formuly modelov založených na logistickej regresii (kód zo softvéru R).

```
# modely logistickej regresie pre Imput vzorku
glm(formula, data = Imput, family = 'binomial')
# LR.Iz
formula="default_flg~product_type+max_dpd_cnt+installment_3m_cnt+
  (education=='1')+avg_rel_inst_12m+execution_6m_flg"
# LR.Is
formula="default_flg~dpd_12m_cnt+execution_flg+installment_3m_cnt+
  (product_code=='52')+max_dpd_cnt+outst_31_60_3m+product_type+
  rel_adt_principal"
# LR.Ib
formula="default_flg~acc_duration+adt_principal+dpd_31_60_3m_cnt+
  execution_6m_flg+installment_3m_cnt+rel_adt_principal+dpd_12m_cnt"

# modely logistickej regresie pre WOE vzorku
glm(formula, data = WOE, family = 'binomial')
# LR.Iz
formula="default_flg~w_rel_outst_31_60_6m+w_fsr_roa+w_nace_sector+
  w_bank_services+w_execution_6m_flg+w_rel_adt_principal"
# LR.Is
formula="default_flg~w_avg_inst_6m+w_dpd_61_90_3m_cnt+w_dpd_6m_cnt+
  w_execution_6m_flg+w_fs_current_assets+w_fsr_sales_tot_assets+
  w_installment_3m_cnt+w_installment_6m+w_legal_form+w_max_dpd_cnt+
  w_product_type+w_rel_adt_principal"
# LR.If
formula="default_flg~w_avg_inst_6m+w_dpd_61_90_3m_cnt+w_dpd_6m_cnt+
  w_execution_6m_flg+w_fs_current_assets+w_fsr_sales_tot_assets+
  w_installment+w_installment_3m_cnt+w_installment_6m+w_legal_form+
  w_product_type+w_rel_adt_principal"
# LR.Ib
formula="default_flg~w_acc_duration+w_adt_duration+
  w_installment_6m+w_execution_flg+w_fs_current_assets+w_dpd_cnt+
  w_installment_3m_cnt+w_rel_outst_31_60_6m+w_rel_adt_principal+
  w_avg_inst_6m+w_installment"
```

Príloha 3

Naprogramované funkcie (kód zo softvéru R).

```
# vypocet statistickych mier pouzitych v praci
# library(ROCR)
Miery=function(y,yhat,k=50)
{N=length(y)
Gini=unlist(performance(prediction(yhat,y),"auc")@y.values)*2-1
b.id=(1:N)*y
b.id=b.id[b.id>0]              # oznacenie indexov zlyhanych
g.id=(1:N)*(1-y)
g.id=g.id[g.id>0]              # oznacenie indexov nezlyhanych
b.pd=yhat[b.id]                # PD zlyhanych
g.pd=yhat[g.id]                # PD nezlyhanych
KS=ks.test(b.pd,g.pd)$statistic
Brier=sum((y-yhat)^2)/N
#entropy measures
ent=c()
All=c()
for (i in 1:k)                 # rozdelenie [0,1] na k intervalov
  {all=sum((i-1)/k<=yhat & yhat<i/k)
   b=sum(y*((i-1)/k<=yhat & yhat<i/k))
   pb=b/all
   All=c(All,all)
   pg=1-pb
   if(is.nan(pb) | pb==0 | pb==1) ent=c(ent,0)
   else ent=c(ent,pb*log(pb)+pg*log(pg))
  }
# toto vazime poctom pozorovani kazdeho intervalu
CIE=-sum(ent*All/N)
pb=sum(y)/N
IE=-pb*log(pb)-(1-pb)*log(1-pb)
output=c(Gini, KS, Brier, IE-CIE, 1-CIE/IE)
names(output)=c("Gini","KS","Brier","KLD","CIER")
return(output)
}

# ------------------------------------------------------------
# cross-validacia pre modely logistickej regresie
crossval=function(data, glmfit, K=50, rate=0.63)
{vysl=matrix(nrow=K,ncol=5)  # 5 je pocet statistik
b.id=1:length(y)*y
b.id=b.id[b.id>0]              # oznacenie indexov zlyhanych
g.id=1:length(y)*(1-y)
g.id=g.id[g.id>0]              # oznacenie indexov nezlyhanych
for (i in 1:K)
 {b.sample = sample(b.id, floor(rate*length(b.id)))
  g.sample = sample(g.id, floor(rate*length(g.id)))
  # nova regresia na nahodne vybranej podvzorke
  n.glm = glm(glmfit$call, data = data[c(b.sample,g.sample), ],
              family = 'binomial')
  # nove out-of-sample odhady (n.yhat) a prislusne triedy (n.y)
  n.yhat = predict(n.glm, newdata=data[-c(b.sample,g.sample), ],
                   type= 'r')
```

```
  n.y = glmfit$y[-c(b.sample,g.sample)]
  vysl[i,]=Miery(n.y, n.yhat)
 }
# priemerujeme vypocitane miery
output=c(rep(1/K,K)%*%vysl)
names(output)=c("Gini","KS","Brier","KLD","CIER")
return(output)
}

# ------------------------------------------------------------------
# funkcia pre vypocet zhlukov premennych v data.frame df
varclusters=function(df)
{
x=matrix(c(unlist(df)), ncol=length(df))
clus=seq(1:ncol(x))
Clus=seq(1:ncol(x))
corr=1-cor(x)^2+diag(rep(1,ncol(x)))
while(var(clus)>0)
    {ind=which(corr==min(corr), arr.ind=TRUE)[1,]
      corr[ind[1],ind[2]]=1
      corr[ind[2],ind[1]]=1
      if(clus[ind[1]]!=clus[ind[2]])
 {clus[clus==clus[ind[1]] | clus==clus[ind[2]] ] = min(clus[ind])
          Clus=rbind(Clus,clus)
         }
     }
# Clus - matica, v kazdom dalsom riadku ma o jeden zhluk menej
# premenne v 1 zhluku maju spolocne cislo
return(Clus)
}
```

Príloha 4

Kód simulácie z časti 3.2.2 (kód zo softvéru R).

```
library(randomForest)
set.seed(45)
N=2000
# generovanie premennych
y=c(rep(1,N/2),rep(0,N/2))
x1=rnorm(N)+y/2*abs(rnorm(N))
x2=floor(2*abs(rnorm(N))+2*abs(y*rnorm(N)))
x3= floor(rnorm(N)-3*abs(y*rnorm(N)))
x4=(x1+y*abs(rnorm(N))>1.2)+0
# data.frames
D=data.frame(y=as.factor(y),x1,x2,x3,x4)
D10=D[c(1:100,1001:1900),]
D50=D[c(1:500,1001:1500),]
D90=D[c(1:900,1001:1100),]
# korelacie premennych v jednotlivych vzorkach
cor(D)

# RF s bootstrapom
rf10=randomForest(y~. , data=D10, mtry=2)
rf50=randomForest(y~. , data=D50, mtry=2)
rf90=randomForest(y~. , data=D90, mtry=2)
# RF so stratifikaciou
strf10=randomForest(y~. , data=D10, mtry=2, sampsize=c(80,80))
strf50=randomForest(y~. , data=D50, mtry=2, sampsize=c(80,80))
strf90=randomForest(y~. , data=D90, mtry=2, sampsize=c(80,80))

# Parcial dep. plots pre x1 v rf10 a strf10
partialPlot(  rf10,D10, x1 , which.class="1")
partialPlot(strf10,D10, x1 , which.class="1")
```