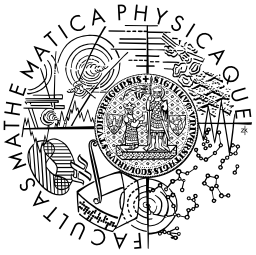


MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Nikita Kan

Porovnání metod pro diverzifikaci 0-1 proměnných

Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: RNDr. Pavel Charamza, CSc.

Studijní program: Finanční matematika

Praha 2023

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

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

V ..... dne ....

Podpis autora

Chtěl bych poděkovat svému vedoucímu RNDr. Pavlu Charamzovi, CSc. za cenné rady a čas, který mi věnoval. Dále bych chtěl poděkovat svým rodičům, kteří mi dali možnost studovat v zahraničí a podporovali mě během celého studia. Chtěl bych také poděkovat své přítelkyni, která vždy stála při mě a věřila mi. V neposlední řadě bych chtěl poděkovat svým kamarádům, bez kterých by můj studentský život byl nudný.

Název práce: Porovnání metod pro diverzifikaci 0-1 proměnných

Autor: Nikita Kan

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: RNDr. Pavel Charamza, CSc., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: Tato práce popisuje a porovnává možné přístupy a jejich matematické pozadí k problematice hledání funkčního modelu pro predikci závisle proměnné s alternativním rozdělením. Jako první metoda se uvádí logistická regrese. Popisují se různé stupně logistické regrese, odhadování parametrů v logistické regresi a metody určování významnosti regresorů. Jako druhá metoda se uvádějí rozhodovací stromy. Popisují se různé typy rozhodovacích stromů a metody jejich konstrukce. Popisuje se také aplikace rozhodovacích stromů v metodě typu „Boost“. Vysvětlují se způsoby porovnávání všech popsaných metod mezi sebou. Porovnání metod se provádí na reálných datech pro vyhodnocení účinnosti reklam v internetovém prostředí. Praktická část práce je zpracována v programu R.

Klíčová slova: diverzifikace, logistická regrese, boost, regresní stromy, klasifikační stromy

Title: Comparing 0-1 diversification methods

Author: Nikita Kan

Department: Department of Probability and Mathematical Statistics

Supervisor: RNDr. Pavel Charamza, CSc., Department of Probability and Mathematical Statistics

Abstract: The thesis describes and compares possible approaches and their mathematical base for finding a functional prediction model of the dependent variable with the alternative distribution. The first method is logistic regression. Different levels of logistic regression, estimation of parameters in logistic regression and methods for determining the significance of regressors are described. The second method is decision trees. Different types of decision trees and methods of their construction are described. The application of decision trees in the "Boost" method is also described. The comparison between the described methods is explained. Comparison of methods is carried out on real data to evaluate the effectiveness of advertising in the internet environment. The practical part of the thesis is performed in the program R.

Keywords: diversification, logistic regression, boost, regression tree, classification tree

Obsah

Úvod

Alternativní rozdělení, neboli Bernoulliho rozdělení, je jedním z druhů pravděpodobnostních rozdělení, které se používá pro popis situací, kdy může nastat jedna ze dvou vzájemně se vylučujících možností. Alternativní rozdělení lze vyjádřit jako náhodnou veličinu, která nabývá hodnoty 1 v případě úspěchu a hodnoty 0 v případě neúspěchu.

Bernoulliho rozdělení se objevuje v různých oblastech. Například banky se zabývají problematikou poskytování úvěrů. Před tím, než banka rozhodne, jestli poskytnout úvěr nebo ne, každá žádost o úvěr se pečlivě posuzuje. V praxi se k posuzování používají matematické modely, které na základě dat žadatele o úvěr rozhodují, jestli žadateli úvěr poskytnout, či nikoliv.

Z matematického hlediska problematiku rozhodování o poskytnutí úvěru můžeme vnímat jako na odhadování závislé proměnné s alternativním rozdělením, kde úspěch (resp. 1) by znamenal to, že žadatel úvěr dostane, zatímco neúspěch (resp. 0) by znamenal to, že se žadateli úvěr neposkytne.

Cílem této práce je popsat a porovnat možné přístupy a jejich matematické pozadí k problematice hledání funkčního modelu pro predikci závisle proměnné s alternativním rozdělením.

Kapitola 1 bude věnována úvodu do predikčních modelů. Popíšeme tam také typy proměnných, se kterými budeme v ostatních kapitolách pracovat. V druhé a třetí kapitolách popíšeme hlavní metody, které budeme používat v našich predikčních modelech. V Kapitole 2 se seznámíme s různými stupni logistické regrese a budeme se zabývat odhadováním parametrů v logistické regrese a určováním jejích významnosti. V Kapitole 3 popíšeme různé druhy rozhodovacích stromů a metody jejich konstrukce. Na konci třetí kapitoly také popíšeme metodu typu „Boost“. V Kapitole 4 vysvětlíme, jak budeme mezi sebou naše modely porovnávat. V Kapitole 5 budeme aplikovat naše metody na reálných datech od jedné vietnamské společnosti a provedeme porovnání našich modelů.

1. Motivace a základní pojmy

Abychom měli lepší představu o tom, co jsou predikční modely a jak tyto modely fungují, zavedeme ilustrativní příklad.

Představme si, že máme soubor minulých pozorování (viz Tabulka 1.1), kde je uvedena základní informace o žadatelích o úvěr a také výsledek rozhodnutí toho, jestli žadatel úvěr dostal, nebo ne. Tabulka 1.1 ilustruje, jak by takový soubor mohl vypadat. První sloupec tabulky nám říká, jestli daný žadatel dostal úvěr. To je naše cílová proměnná, kterou chceme predikovat a které budeme říkat *odezva*, neboli *závislá proměnná*. Ostatní sloupce tabulky představují nějakou informaci o žadateli, na základě které se rozhoduje, jestli se úvěr bude poskytnut, či ne. Těmto proměnným budeme říkat *regresory*, neboli *nezávislé proměnné*.

Zavedeme vlastní definice a důležité pojmy.

Definice 1.1. *Predikčním modelem nazveme libovolnou funkci f, která přiřazuje vektoru regresorů $\boldsymbol{x}$ odhad závislé proměnné $\hat{y}$ jako*

$$f(\boldsymbol{x},w) = \hat{y},$$

kde w je nějaký parametr predikčního modelu.

Poznámka 1.1. Odezva a regresory mohou být jak náhodnými veličinami, tak nenáhodnými. V této práci vektorem $\boldsymbol{x}$ rozumíme konkrétní realizaci náhodného vektoru regresorů $\boldsymbol{X}$ a y je konkrétní realizace náhodné závislé proměnné Y.

Definice 1.2. *Nechť máme n veličin $y_1, \ldots, y_n$, které závisí na vektorech regresorů $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Potom množina dvojic $L = \{(y_1, \boldsymbol{x}_1), \ldots, (y_n, \boldsymbol{x}_n)\}$ se nazývá učební vzorek (angl. learning sample).*

Ve statistice máme 2 typy proměnných: *nominální* a *kategoriální*. Nominální veličiny označují množství něčeho a dělí se na *spojité* a *diskrétní* veličiny. Příkladem spojité veličiny je cena akcie, zatímco příkladem diskrétní veličiny může být počet přednášek. Kategoriální proměnné vyjadřují členství v nějaké skupině nebo kategorii a dělí se na *ordinální* a *neordinální*. Neordinální veličiny se nedají seřadit, např. pohlaví. U ordinální veličiny existuje logické uspořádání, např. stupeň dosaženého vzdělání.

Závislou proměnnou s alternativním rozdělením můžeme vnímat jako kategoriální neordinální veličinu nebo jako nominální diskrétní veličinu. To, jak budeme

Poskytnutí úvěru	**Věk**	**Povolání**	**Měsíční mzda**
Ano	35	Advokát	70 000
Ne	23	Student	16 000
Ano	46	Hasič	60 000
Ano	55	Chirurg	97 000
...	...	...	...
Ano	24	Aktuár	48 000

Tabulka 1.1: Soubor minulých pozorování.

vnímat odezvu, bude ovlivňovat nejen způsob konstrukce predikčních modelů, ale i metody porovnání modelů mezi sebou.

Definice 1.3. *Řekneme, že predikční model $f^{(p)}$ je prediktor (angl. predictor), pokud přiřazuje vektoru regresorů $\boldsymbol{x}$ odhad závislé proměnné $\hat{y}$ jako*

$$f^{(p)}(\boldsymbol{x},w) = \hat{y},$$

kde w je nějaký parametr prediktoru a $\hat{y}$ je nominální veličina.

Definice 1.4. *Řekneme, že predikční model $f^{(c)}$ je klasifikátor (angl. classifier), pokud přiřazuje vektoru regresorů $\boldsymbol{x}$ odhad závislé proměnné $\hat{y}$ jako*

$$f^{(c)}(\boldsymbol{x},w) = \hat{y},$$

kde w je nějaký parametr klasifikátoru a $\hat{y}$ je kategoriální veličina.

Zavedli jsme pojmy, které budeme používat v dalších kapitolách, a můžeme začít popisovat matematické pozadí predikčních modelů.

2. Logistická regrese

První metoda, jak můžeme predikovat závislou proměnnou s alternativním rozdělením na základě nezávislých proměnných, je logistická regrese. V první a třetí podkapitolách, kde popisujeme model logistické regrese a odhadujeme parametry modelu, vycházíme ze Zváry (Zvára, 2008) a Anděla (Anděl, 2007). Ve druhé podkapitole, kde popisujeme různé stupně logistické regrese, vycházíme z nepublikovaných poznámek k přednáškám vedoucího této práce. Ve čtvrté podkapitole, kde určujeme významnost regresorů modelu, vycházíme z práce Hosmera, Lemeshowa a Sturdivanta (Hosmer, Lemeshow a Sturdivant, 2013).

2.1 Model logistické regrese

Nechť máme n nezávislých náhodných veličin $Y_1, \ldots, Y_n$ s alternativními rozděleními s parametry $\theta(\boldsymbol{x}_i)$, $i = 1, \ldots, n$. Předpokládáme, že odezva Y_i závisí na vektoru regresorů $\boldsymbol{x}_i^\top = (1, \boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n})$, $k < n$. Dále budeme předpokládat, že matice $\mathbb{X} = (x_{i,j})$ typu $n \times (k+1)$, kde $j = 0, \ldots, k$, má lineárně nezávislé sloupce. Matici $\mathbb{X}$ říkáme *regresní matice*.

Definice 2.1. *Řekneme, že závislá proměnná Y_i a vektor regresorů $\boldsymbol{x}_i$ splňují logistický regresní model, pokud platí*

$$\mathsf{E}\left[Y_i | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}},$$

kde $\boldsymbol{\beta}^\top = (\beta_0, \beta_1, \ldots, \beta_k)$ je vektor parametrů modelu, $i = 1, \ldots, n$.

Z vlastností alternativního rozdělení víme, že střední hodnota náhodné veličiny Y_i je hodnota parametru $\theta(\boldsymbol{x}_i)$, což je pravděpodobnost výskytu jevu, tj. platí

$$\mathsf{P}[Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i] = \theta(\boldsymbol{x}_i) = \frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}. \tag{2.1}$$

Poznámka 2.1. Existují další způsoby, jak se dá vyjádřit střední hodnotu náhodné veličiny Y_i jako funkci regresorů. Mohli bychom uvažovat lineární závislost

$$\theta(\boldsymbol{x}_i) = \boldsymbol{\beta}^\top \boldsymbol{x}_i,$$

kde bychom se narazili na problém, který spočívá v tom, že nemůžeme v tomto případě zaručit, že pro libovolné hodnoty $x_{i,1}, \ldots, x_{i,k}$ hodnota $\theta(\boldsymbol{x}_i)$ bude ležet v intervalu $(0, 1)$. Aby tato podmínka byla splněna, transformujeme vhodným způsobem $\boldsymbol{\beta}^\top \boldsymbol{x}_i$. Často se používají funkce *probit* f_P, funkce *log-log* f_{LL} a funkce *komplementární log-log* f_{CL} (angl. *complementary log-log*)(viz Hosmer a kol., 2013, odst. 10.7):

$$\begin{aligned}
f_P(\boldsymbol{x}_i) = \Phi^{-1}\left(\theta_P(\boldsymbol{x}_i)\right) = \boldsymbol{\beta}^\top \boldsymbol{x}_i \quad &\Rightarrow \quad \theta_P(\boldsymbol{x}_i) = \Phi\left(\boldsymbol{\beta}^\top \boldsymbol{x}_i\right), \\
f_{LL}(\boldsymbol{x}_i) = -\ln\left(-\ln\left(\theta_{LL}(\boldsymbol{x}_i)\right)\right) = \boldsymbol{\beta}^\top \boldsymbol{x}_i \quad &\Rightarrow \quad \theta_{LL}(\boldsymbol{x}_i) = \frac{1}{\mathrm{e}^{\mathrm{e}^{-\boldsymbol{\beta}^\top \mathrm{x_i}}}}, \\
f_{CL}(\boldsymbol{x}_i) = \ln\left(-\ln\left(1 - \theta_{CL}(\boldsymbol{x}_i)\right)\right) = \boldsymbol{\beta}^\top \boldsymbol{x}_i \quad &\Rightarrow \quad \theta_{CL}(\boldsymbol{x}_i) = 1 - \frac{1}{\mathrm{e}^{\mathrm{e}^{\boldsymbol{\beta}^\top \mathrm{x_i}}}},
\end{aligned}$$

kde Φ je distribuční funkce normovaného normálního rozdělení a $\theta_P(\boldsymbol{x}_i)$, $\theta_{LL}(\boldsymbol{x}_i)$ a $\theta_{CL}(\boldsymbol{x}_i)$ označují pravděpodobnost výskytu jevu. V modelu logistické regrese se používá funkce logit:

$$\mathsf{logit}(\theta(\boldsymbol{x}_i)) = \ln\left(\frac{\theta(\boldsymbol{x}_i)}{1-\theta(\boldsymbol{x}_i)}\right) = \boldsymbol{\beta}^\top \boldsymbol{x}_i \qquad \Rightarrow \qquad \theta(\boldsymbol{x}_i) = \frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}.$$

2.2 Stupně logistické regrese

V této podkapitole popíšeme 3 stupně logistické regrese. K tomu potřebujeme zavést následující definice a lemma.

Definice 2.2. *Označím p jako pravděpodobnost toho, že nějaký jev nastane. Pak podílu $\frac{p}{1-p}$ říkáme šance (angl. odds).*

Definice 2.3. *Řekneme, že diskrétní náhodné veličiny $X_1, \ldots, X_k$ jsou podmíněně nezávislé vzhledem k diskrétní náhodné veličině Y, pokud pro $x_1, \ldots, x_k$ z oboru hodnot náhodných veličin $X_1, \ldots, X_k$ a pro y z oboru hodnot náhodné veličiny Y platí*

$$\mathsf{P}\left[X_1 = x_1, \ldots, X_k = x_k | Y = y\right] = \prod_{j=1}^{k} \mathsf{P}\left[X_j = x_j | Y = y\right].$$

Lemma 2.1. *Mějme závislou proměnnou Y s alternativním rozdělením, která závisí na vektoru regresorů $\boldsymbol{X}^\top = (X_1, \ldots, X_k)$, kde $X_1, \ldots, X_k$ jsou diskrétní veličiny. Za předpokladu podmíněné nezávislosti regresorů $X_1, \ldots, X_k$ vzhledem k Y platí*

$$\frac{\mathsf{P}\left[Y = 1 | \boldsymbol{X} = \boldsymbol{x}\right]}{\mathsf{P}\left[Y = 0 | \boldsymbol{X} = \boldsymbol{x}\right]} = \mathsf{odds}_{total} \cdot \prod_{j=1}^{k} \mathsf{odds}_{ratio}\left(x_j\right), \tag{2.2}$$

kde $\boldsymbol{x}^\top = (x_1, \ldots, x_k)$, $\mathsf{odds}_{total} = \frac{\mathsf{P}[Y=1]}{\mathsf{P}[Y=0]}$ a $\mathsf{odds}_{ratio}\left(x_j\right) = \frac{\mathsf{P}[Y=1|X_j=x_j]}{\mathsf{P}[Y=0|X_j=x_j]} \cdot \frac{1}{\mathsf{odds}_{total}}$, $j = 1, \ldots, k$.

Důkaz.

$$\begin{aligned}
\frac{\mathsf{P}\left[Y = 1 | \boldsymbol{X} = \boldsymbol{x}\right]}{\mathsf{P}\left[Y = 0 | \boldsymbol{X} = \boldsymbol{x}\right]} &= \frac{\frac{\mathsf{P}[\boldsymbol{X}=\boldsymbol{x} \wedge Y=1]}{\mathsf{P}[\boldsymbol{X}=\boldsymbol{x}]}}{\frac{\mathsf{P}[\boldsymbol{X}=\boldsymbol{x} \wedge Y=0]}{\mathsf{P}[\boldsymbol{X}=\boldsymbol{x}]}} = \frac{\mathsf{P}\left[Y = 1\right] \cdot \mathsf{P}\left[\boldsymbol{X} = \boldsymbol{x} | Y = 1\right]}{\mathsf{P}\left[Y = 0\right] \cdot \mathsf{P}\left[\boldsymbol{X} = \boldsymbol{x} | Y = 0\right]} \\
&= \frac{\mathsf{P}\left[Y = 1\right]}{\mathsf{P}\left[Y = 0\right]} \cdot \frac{\prod_{j=1}^{k} \mathsf{P}\left[X_j = x_j | Y = 1\right]}{\prod_{j=1}^{k} \mathsf{P}\left[X_j = x_j | Y = 0\right]} \\
&= \frac{\mathsf{P}\left[Y = 1\right]}{\mathsf{P}\left[Y = 0\right]} \cdot \prod_{j=1}^{k} \frac{\frac{\mathsf{P}[Y=1|X_j=x_j] \cdot \mathsf{P}[X_j=x_j]}{\mathsf{P}[Y=1]}}{\frac{\mathsf{P}[Y=0|X_j=x_j] \cdot \mathsf{P}[X_j=x_j]}{\mathsf{P}[Y=0]}} \\
&= \frac{\mathsf{P}\left[Y = 1\right]}{\mathsf{P}\left[Y = 0\right]} \cdot \prod_{j=1}^{k} \frac{\frac{\mathsf{P}[Y=1|X_j=x_j]}{\mathsf{P}[Y=0|X_j=x_j]}}{\frac{\mathsf{P}[Y=1]}{\mathsf{P}[Y=0]}} = \mathsf{odds}_{total} \cdot \prod_{j=1}^{k} \mathsf{odds}_{ratio}\left(x_j\right).
\end{aligned}$$

□

Můžeme si všimnout, že podíl $\frac{\mathsf{P}[Y=1|\boldsymbol{X}=\boldsymbol{x}]}{\mathsf{P}[Y=0|\boldsymbol{X}=\boldsymbol{x}]}$ se dá přepsat jako $\frac{\mathsf{P}[Y=1|\boldsymbol{X}=\boldsymbol{x}]}{1-\mathsf{P}[Y=1|\boldsymbol{X}=\boldsymbol{x}]}$, což není nic jiného než šance. Pokud navíc zlogaritmujeme rovnost (2.2), pak dostaneme

$$\begin{aligned}\ln\left(\frac{\mathsf{P}\left[Y=1|\boldsymbol{X}=\boldsymbol{x}\right]}{\mathsf{P}\left[Y=0|\boldsymbol{X}=\boldsymbol{x}\right]}\right) &= \ln\left(\frac{\mathsf{P}\left[Y=1|\boldsymbol{X}=\boldsymbol{x}\right]}{1-\mathsf{P}\left[Y=1|\boldsymbol{X}=\boldsymbol{x}\right]}\right)\\ &= \ln\left(\mathsf{odds}_{total}\cdot\prod_{j=1}^{k}\mathsf{odds}_{ratio}\left(x_j\right)\right),\end{aligned} \tag{2.3}$$

odkud jednoduchými úpravy dostaneme

$$\mathsf{P}\left[Y=1|\boldsymbol{X}=\boldsymbol{x}\right]=\frac{\mathrm{e}^{\ln\left(\mathsf{odds}_{total}\right)+\sum_{j=1}^{k}\ln\left(\mathsf{odds}_{ratio}(x_j)\right)}}{1+\mathrm{e}^{\ln\left(\mathsf{odds}_{total}\right)+\sum_{j=1}^{k}\ln\left(\mathsf{odds}_{ratio}(x_j)\right)}}. \tag{2.4}$$

Vidíme, že v rovnosti (2.3) se objevil výraz $\ln\left(\frac{\mathsf{P}[Y=1|\boldsymbol{X}=\boldsymbol{x}]}{1-\mathsf{P}[Y=1|\boldsymbol{X}=\boldsymbol{x}]}\right)$, což je logaritmus šance, neboli funkce logit. Všimněme si také, že výraz (2.4) připomíná tvar logistické regrese, která je vyjádřena v rovnosti (2.1). Výraz (2.4) vede nás k prvnímu stupni logistické regrese.

Definice 2.4. *Řekneme, že závislá proměnná Y_i a vektor regresorů $\boldsymbol{x}_i$ splňují Nezávislý model (angl. Independence model), pokud platí*

$$\mathsf{P}\left[Y_i=1|\boldsymbol{X}_i=\boldsymbol{x}_i\right]=\frac{\mathrm{e}^{\ln\left(\mathsf{odds}_{total}\right)+\sum_{j=1}^{k}\ln\left(\mathsf{odds}_{ratio}(x_{i,j})\right)}}{1+\mathrm{e}^{\ln\left(\mathsf{odds}_{total}\right)+\sum_{j=1}^{k}\ln\left(\mathsf{odds}_{ratio}(x_{i,j})\right)}},$$

kde $\mathsf{odds}_{total}=\frac{\mathsf{P}[Y_i=1]}{\mathsf{P}[Y_i=0]}$ *a* $\mathsf{odds}_{ratio}\left(x_{i,j}\right)=\frac{\mathsf{P}[Y_i=1|X_{i,j}=x_{i,j}]}{\mathsf{P}[Y_i=0|X_{i,j}=x_{i,j}]}\cdot\frac{1}{\mathsf{odds}_{total}}$ *pro* $j=1,\ldots,k$ *a* $i=1,\ldots,n$.

Jedná se o specifický případ logistické regrese, kde vektor parametrů $\boldsymbol{\beta}$ je jednotkový vektor. Říkáme tomuto modelu Nezávislý, protože se předpokládá podmíněná nezávislost regresorů.

Poznámka 2.2. Výrazu $\ln\left(\mathsf{odds}_{ratio}\left(x_{i,j}\right)\right)$ říkáme *WOE* (*Weight of Evidence*) a značíme ho $\mathsf{WOE}_{i,j}$. WOE nám ukazuje, jak dobře kategorie regresoru „predikuje“ závislou proměnnou. Kladná WOE znamená, že šance na výskyt jevu s danou kategorií regresoru je větší než celková šance na výskyt jevu. Naopak, záporná WOE znamená, že šance na výskyt jevu s danou kategorií regresoru je menší než celková šance na výskyt jevu. Nulová WOE by znamenala, že daná kategorie regresoru nemá žádný vliv na pravděpodobnost výskytu jevu.

Odhad pravděpodobnosti výskytu jevu v Nezávislém modelu vyjádříme jako

$$\widehat{\mathsf{P}}\left[Y_i=1|\boldsymbol{X}_i=\boldsymbol{x}_i\right]=\frac{\mathrm{e}^{\ln\left(\widehat{\mathsf{odds}}_{total}\right)+\sum_{j=1}^{k}\ln\left(\widehat{\mathsf{odds}}_{ratio}(x_{i,j})\right)}}{1+\mathrm{e}^{\ln\left(\widehat{\mathsf{odds}}_{total}\right)+\sum_{j=1}^{k}\ln\left(\widehat{\mathsf{odds}}_{ratio}(x_{i,j})\right)}}, \tag{2.5}$$

kde $\widehat{\mathsf{odds}}_{total}=\frac{\sum_{i=1}^{n}\mathbb{I}[Y_i=1]}{\sum_{i=1}^{n}\mathbb{I}[Y_i=0]}$ a $\widehat{\mathsf{odds}}_{ratio}(x_{i,j})=\frac{\sum_{i=1}^{n}\mathbb{I}[Y_i=1|X_{i,j}=x_{i,j}]}{\sum_{i=1}^{n}\mathbb{I}[Y_i=0|X_{i,j}=x_{i,j}]}\cdot\frac{1}{\widehat{\mathsf{odds}}_{total}}$.

Poznámka 2.3. Symbolem $\mathbb{I}$ značíme funkci, která přiřazuje hodnotu 1 v případě splnění vnitřní podmínky a hodnotu 0 v případě nesplnění této podmínky. Například výrazem $\sum_{i=1}^{n} \mathbb{I}\left[Y_i = 1\right]$ rozumíme počet toho, kolikrát nastal jev.

Regresní matice v Nezávislém modelu má následující tvar:

$$\mathbb{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{1,2} & \dots & x_{n,k} \end{pmatrix}.$$

Definice 2.5. *Řekneme, že závislá proměnná Y_i a vektor regresorů $\boldsymbol{x}_i$ splňují WOE model, pokud platí*

$$\mathsf{P}\left[Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \frac{\mathrm{e}^{\beta_0 \cdot \ln\left(\mathsf{odds}_{total}\right) + \sum_{j=1}^{k} \beta_j \cdot \mathsf{WOE}_{i,j}}}{1 + \mathrm{e}^{\beta_0 \cdot \ln\left(\mathsf{odds}_{total}\right) + \sum_{j=1}^{k} \beta_j \cdot \mathsf{WOE}_{i,j}}},$$

kde β_j je parametr j-tého regresoru, $j = 0, \dots, k$ a $i = 1, \dots, n$.

Druhý stupeň logistické regrese je zobecnění Nezávislého modelu, kde každý regresor bude mít svou příslušnou váhu.

Odhad pravděpodobnosti výskytu jevu ve WOE modelu vyjádříme jako

$$\widehat{\mathsf{P}}\left[Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \frac{\mathrm{e}^{\widehat{\beta}_0 \cdot \ln\left(\widehat{\mathsf{odds}}_{total}\right) + \sum_{j=1}^{k} \widehat{\beta}_j \cdot \widehat{\mathsf{WOE}}_{i,j}}}{1 + \mathrm{e}^{\widehat{\beta}_0 \cdot \ln\left(\widehat{\mathsf{odds}}_{total}\right) + \sum_{j=1}^{k} \widehat{\beta}_j \cdot \widehat{\mathsf{WOE}}_{i,j}}},$$

kde $\widehat{\mathsf{WOE}}_{i,j} = \ln\left(\widehat{\mathsf{odds}}_{ratio}(x_{i,j})\right)$ pro $j = 1, \dots, k$ a $\widehat{\beta}_j$ je odhad parametru β_j pro $j = 0, \dots, k$. Problematikou odhadů parametrů $\beta_0, \dots, \beta_k$ se budeme zabývat v následující podkapitole „Odhad parametrů“.

Regresní matice ve WOE modelu má následující tvar:

$$\mathbb{X} = \begin{pmatrix} \ln\left(\widehat{\mathsf{odds}}_{total}\right) & \widehat{\mathsf{WOE}}_{1,1} & \dots & \widehat{\mathsf{WOE}}_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ \ln\left(\widehat{\mathsf{odds}}_{total}\right) & \widehat{\mathsf{WOE}}_{n,1} & \dots & \widehat{\mathsf{WOE}}_{n,k} \end{pmatrix},$$

kde v každém sloupci hodnota WOE bude nabývat tolik hodnot, kolik kategorií obsahuje daný regresor.

Definice 2.6. *Řekneme, že závislá proměnná Y_i a vektor regresorů $\boldsymbol{x}_i$ splňují model Plné logistické regrese (angl. Full logistic regression), pokud platí*

$$\mathsf{P}\left[Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \frac{\mathrm{e}^{\beta_0 + \sum_{j=1}^{k} \sum_{s_j=1}^{z_j - 1} \beta_{s_j} \cdot x_{i,s_j}}}{1 + \mathrm{e}^{\beta_0 + \sum_{j=1}^{k} \sum_{s_j=1}^{z_j - 1} \beta_{s_j} \cdot x_{i,s_j}}},$$

kde z_j je roven počtu kategorií, pokud j-tý regresor je kategoriální veličina, a $z_j = 2$, pokud j-tý regresor je nominální veličina, $j = 1, \dots, k$ a $i = 1, \dots, n$.

Třetí stupeň logistické regrese je obecný model logistické regrese. V tomto modelu každý j-tý kategoriální regresor bude mít pro každou svou s_j-tou kategorii

váhu β_{s_j}. Je důležité si uvědomit, že nominální regresor bude mít jenom jednu váhu, resp. parametr.

Odhad pravděpodobnosti výskytu jevu v modelu Plné logistické regrese vyjádříme jako

$$\widehat{\mathsf{P}}\left[Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \frac{\mathrm{e}^{\widehat{\beta}_0 + \sum_{j=1}^{k} \sum_{s_j=1}^{z_j - 1} \widehat{\beta}_{s_j} \cdot x_{i,s_j}}}{1 + \mathrm{e}^{\widehat{\beta}_0 + \sum_{j=1}^{k} \sum_{s_j=1}^{z_j - 1} \widehat{\beta}_{s_j} \cdot x_{i,s_j}}}.$$

Regresní matice v tomto modelu má následující tvar:

$$\mathbb{X} = \begin{pmatrix} 1 & 1 & 0 & \dots & x_{1,j} & \dots & 1 \\ 1 & 0 & 1 & \dots & x_{2,j} & \dots & 0 \\ 1 & 0 & 0 & \dots & x_{3,j} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & 1 & 0 & \dots & x_{n,j} & \dots & 1 \end{pmatrix}, \tag{2.6}$$

kde počet sloupců odpovídající každému kategoriálnímu regresoru je o 1 menší než počet kategorií daného regresoru. Vynecháváme vždycky jeden sloupec u každého kategoriálního regresoru, aby regresní matice měla lineárně nezávislé sloupce. Z tohoto důvodu první sloupec u regresní matice (2.6) je vektor jedniček, tj. $(x_{1,0}, \dots, x_{n,0})^\top = (1, \dots, 1)^\top$. Je jasné, že pokud regresor je nominální veličina, tak tomuto regresoru bude odpovídat jenom jeden sloupec v regresní matice, jako například j-tý regresor v matici (2.6). Prvky sloupců, které odpovídají kategoriálním regresorům regresní matice v tomto modelu, jsou tzv. *binární proměnné* (angl. *dummy variables*), které nabývají pouze hodnoty 0 a 1. Podle sloupců příslušných regresorů můžeme snadno určit do jaké kategorie patří daný subjekt. Například, nechť druhý a třetí sloupec matice (2.6) označují první proměnnou „rodinný stav“, která má 3 kategorie: svobodný, ženatý a rozvedený. Pak hodnota 1. pozorování dané proměnné $(x_{1,1}, x_{1,2}) = (1, 0)$ bude značit kód kategorie „svobodný“, hodnota 2. pozorování $(x_{2,1}, x_{2,2}) = (0, 1)$ bude značit kategorii „ženatý“ a hodnota 3. pozorování $(x_{3,1}, x_{3,2}) = (0, 0)$ bude značit kategorii „rozvedený“.

V praxi se používají WOE model a model Plné logistické regrese, zatímco Nezávislý model se používá jenom v případě malého počtu dat. V Nezávislém a WOE modelech potřebujeme, aby všechny regresory byly kategoriálními nebo nominálními diskrétními veličinami.

2.3 Odhad parametrů

V této podkapitole budeme odhadovat vektor parametrů $\boldsymbol{\beta}$ pomocí *metody maximální věrohodnosti*. Hlavní myšlenka této metody spočívá v tom, že chceme najít takový odhad vektoru parametrů $\boldsymbol{\beta}$, který by maximalizoval pravděpodobnost toho, že pozorované hodnoty $Y_1, \dots, Y_n$ procházejí z předpokládaného rozdělení pravděpodobnosti. Abychom mohli použít tuto metodu, zavedeme takzvanou *věrohodnostní funkci*, která je definována jako

$$L_n(\boldsymbol{\beta}) = \prod_{i=1}^{n} f\left(Y_i | \theta(\boldsymbol{x}_i)\right),$$

kde f^* je hustota náhodné veličiny Y_i a $\theta(\boldsymbol{x}_i) = \frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}$ je parametr rozdělení, ze kterého prochází Y_i. Vzhledem k tomu, že Y_i má alternativní rozdělení, tak věrohodnostní funkce bude mít tvar

$$L_n(\boldsymbol{\beta}) = \prod_{i=1}^{n} \theta(\boldsymbol{x}_i)^{Y_i} \cdot (1 - \theta(\boldsymbol{x}_i))^{1-Y_i}.$$

Zavedeme také *logaritmickou věrohodností funkci*, která je definována jako

$$\ell_n(\boldsymbol{\beta}) = \ln\left(L_n(\boldsymbol{\beta})\right).$$

Z výpočetního hlediska se s ní lépe pracuje a také platí, že $L_n(\boldsymbol{\beta})$ a $\ell_n(\boldsymbol{\beta})$ nabývají vzhledem k $\boldsymbol{\beta}$ svého maxima v tomtéž bodě, protože logaritmus je ryze rostoucí funkce. Takže dostáváme následující vztah:

$$\ell_n(\boldsymbol{\beta}) = \ln\left(L_n(\boldsymbol{\beta})\right) = \sum_{i=1}^{n} \Big[Y_i \cdot \ln\left(\theta(\boldsymbol{x}_i)\right) + (1 - Y_i) \cdot \ln\left(1 - \theta(\boldsymbol{x}_i)\right) \Big].$$

Vzhledem k tomu, že $\theta(\boldsymbol{x}_i) = \frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}$, výše uvedený vztah můžeme přepsat jako

$$\begin{aligned}
\ell_n(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \left[Y_i \cdot \ln\left(\frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}\right) + (1 - Y_i) \cdot \ln\left(1 - \frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}\right) \right] \\
&= \sum_{i=1}^{n} \left[Y_i \cdot \left(\ln\left(\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}\right) - \ln\left(1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}\right) \right) + (1 - Y_i) \cdot \left(\ln 1 - \ln\left(1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}\right) \right) \right] \\
&= \sum_{i=1}^{n} \left[Y_i \cdot \boldsymbol{\beta}^\top \boldsymbol{x}_i - \ln\left(1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}\right) \right].
\end{aligned}$$

Abychom našli *maximálně věrohodný odhad* $\widehat{\boldsymbol{\beta}}_n$ vektoru parametrů $\boldsymbol{\beta}$, který je definován jako $\widehat{\boldsymbol{\beta}}_n = \underset{\boldsymbol{\beta} \in \mathbb{R}^{k+1}}{\arg\max}\, L_n(\boldsymbol{\beta})$, tak musíme spočítat parciální derivace logaritmické věrohodností funkce podle složek vektoru $\boldsymbol{\beta}$ a položit je rovné nule:

$$\frac{\partial \ell_n(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n} \left[Y_i \cdot x_{i,j} - x_{i,j} \cdot \frac{\mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}}{1 + \mathrm{e}^{\boldsymbol{\beta}^\top \boldsymbol{x}_i}} \right], \qquad j = 0, \ldots, k.$$

Tím dostaneme soustavu $k+1$ tzv. *věrohodnostních rovnic*, řešení kterých nám dá odhad $\widehat{\boldsymbol{\beta}}_n$. K výpočtu této soustavy rovnic se používají numerické metody, například Newton-Raphsonova metoda.

Spočítáme také asymptotické odhady rozptylů $\widehat{\mathrm{var}}(\widehat{\beta}_j)$ odhadnutých parametrů $\widehat{\beta}_j$, $j = 0, \ldots, k$, které se nám budou hodit v následující podkapitole „Významnost regresorů“. Dle Anděla (2007, Věta 7.100) platí asymptotická normalita maximálně věrohodného odhadu

$$\sqrt{n} \cdot \left(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta} \right) \xrightarrow{\mathrm{D}} \mathsf{N}\left(\mathbf{0}, \boldsymbol{J}^{-1}(\boldsymbol{\beta}) \right), \tag{2.7}$$

kde $\boldsymbol{J}(\boldsymbol{\beta})$ je *Fisherova informační matice*, prvky které jsou definovány jako

$$\Big(\boldsymbol{J}(\boldsymbol{\beta})_{j,t} \Big) = -\,\mathsf{E}\, \frac{\partial^2 \ln f\left(Y_i | \theta(\boldsymbol{x}_i)\right)}{\partial \beta_j \partial \beta_t}, \qquad \text{pro } j, t = 0, \ldots, k.$$

*V této podkapitole symbolem f značíme hustotu náhodné veličiny. V dalších podkapitolách tím značením budeme zase rozumět predikční model.

Pokud spočítáme nějaký konzistentní odhad Fisherovy informační matice, tak dostaneme asymptotický odhad varianční matice. Jako konzistentní odhad Fisherovy informační matice můžeme použít *pozorovanou informační matici* $\boldsymbol{J}_n(\boldsymbol{\beta}|\boldsymbol{Y})$, prvky které jsou definovány jako

$$\left(\boldsymbol{J}_n(\boldsymbol{\beta}|\boldsymbol{Y})_{j,t}\right) = -\frac{1}{n}\cdot\sum_{i=1}^{n}\frac{\partial^2 \ln f\left(Y_i|\theta(\boldsymbol{x}_i)\right)}{\partial\beta_j\partial\beta_t},$$

kde $\boldsymbol{Y}^\top = (Y_1,\ldots,Y_n)$ a $j,t = 0,\ldots,k$. Pro prvky matice $\boldsymbol{J}_n(\boldsymbol{\beta}|\boldsymbol{Y})$ platí

$$\begin{aligned}\left(\boldsymbol{J}_n(\boldsymbol{\beta}|\boldsymbol{Y})_{j,t}\right) &= -\frac{1}{n}\cdot\sum_{i=1}^{n}\frac{\partial^2 \ln f\left(Y_i|\theta(\boldsymbol{x}_i)\right)}{\partial\beta_j\partial\beta_t}\\ &= -\frac{1}{n}\cdot\sum_{i=1}^{n}\frac{\partial}{\partial\beta_j}\left(Y_i\cdot x_{i,t} - x_{i,t}\cdot\frac{\mathrm{e}^{\boldsymbol{\beta}^\top\boldsymbol{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}^\top\boldsymbol{x}_i}}\right)\\ &= \frac{1}{n}\cdot\sum_{i=1}^{n}\left[x_{i,t}\cdot x_{i,j}\cdot\frac{\mathrm{e}^{\boldsymbol{\beta}^\top\boldsymbol{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}^\top\boldsymbol{x}_i}}\cdot\left(1-\frac{\mathrm{e}^{\boldsymbol{\beta}^\top\boldsymbol{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}^\top\boldsymbol{x}_i}}\right)\right].\end{aligned}$$

Po dosazení vektoru $\widehat{\boldsymbol{\beta}}_n$ do matice $\boldsymbol{J}_n(\boldsymbol{\beta}|\boldsymbol{Y})$ dostaneme matice $\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})$, která bude konzistentním odhadem $\boldsymbol{J}(\boldsymbol{\beta})$ (viz Zvára, 2008, odst. 12.2). Jelikož matice $\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})$ je čtvercová, symetrická a pozitivně definitní matice, tak existuje čtvercová, symetrická a pozitivně definitní matice $\left(\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})\right)^{1/2}$ taková, že

$$\left(\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})\right)^{1/2}\cdot\left(\left(\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})\right)^{1/2}\right)^\top = \boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y}).$$

Takže pomocí tohoto rozkladu dostaneme z výrazu (2.7)

$$\sqrt{n}\cdot\left(\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})\right)^{1/2}\cdot\left(\widehat{\boldsymbol{\beta}}_n-\boldsymbol{\beta}\right)\xrightarrow{\mathrm{D}}\mathsf{N}\left(\mathbf{0},\mathbf{I}\right),$$

kde $\mathbf{I}$ je jednotková matice.

Takže asymptotickým odhadem varianční matice je matice $n\cdot\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})$. Prvky na diagonále matice $\left(n\cdot\boldsymbol{J}_n(\widehat{\boldsymbol{\beta}}_n|\boldsymbol{Y})\right)^{-1}$ jsou naše asymptotické odhady rozptylů $\widehat{\mathrm{var}}(\widehat{\beta}_j)$, $j=0,\ldots,k$.

2.4 Významnost regresorů

Když jsme spočítali odhady parametrů modelu, tak musíme určit, které regresory jsou v našem modelu významné. Děláme to z několika důvodů. Zaprvé, pokud model obsahuje příliš hodně regresorů, tak se objevuje riziko „přefitování“. To znamená, že model natolik přesně predikuje závislou proměnnou na základě původních dat, že pro nová data model už může být zavádějící. Zadruhé, čím je náš model větší, tím je jeho interpretace složitější.

Hlavní myšlenka toho, jak se provádí určování významnosti regresorů, spočívá v následující otázce: Je model, který zahrnuje daný regresor lepší než model, který daný regresor nezahrnuje?

Zaprvé si ukážeme, jak se dá určit významnost jednotlivých regresorů. Vzhledem k tomu, že platí asymptotická normalita maximálně věrohodného odhadu, tak pro každý odhadnutý parametr $\widehat{\beta}_j$, kde $j = 0, \ldots, k$, platí

$$\frac{\widehat{\beta}_j - \beta_j}{\sqrt{\widehat{\mathrm{var}}(\widehat{\beta}_j)}} \xrightarrow{\mathrm{D}} \mathsf{N}(0,1).$$

Budeme testovat nulovou hypotézu, že $\beta_j = 0$, oproti alternativě, že $\beta_j \neq 0$. Výraz $\beta_j = 0$ znamená, že regresor je nevýznamný. Tedy testujeme nulovou hypotézu, že regresor je nevýznamný, oproti alternativě, že regresor je významný. Budeme používat tzv. *Waldovu statistiku*

$$W = \frac{\widehat{\beta}_j}{\sqrt{\widehat{\mathrm{var}}(\widehat{\beta}_j)}},$$

která má za platnosti nulové hypotézy asymptoticky normované normální rozdělení $\mathsf{N}(0,1)$. Testujeme hypotézu na hladině α tak, že porovnáme hodnotu $|W|$ s $z_{1-\frac{\alpha}{2}}$, což je $\left(1-\frac{\alpha}{2}\right)$-tý kvantil normovaného normálního rozdělení. Hypotézu zamítneme ve prospěch alternativy, pokud $|W| > z_{1-\frac{\alpha}{2}}$. Můžeme také sestrojit asymptotický interval spolehlivosti pro parametr β_j

$$\mathsf{P}\left[\widehat{\beta}_j - z_{1-\frac{\alpha}{2}} \cdot \sqrt{\widehat{\mathrm{var}}(\widehat{\beta}_j)} < \beta_j < \widehat{\beta}_j + z_{1-\frac{\alpha}{2}} \cdot \sqrt{\widehat{\mathrm{var}}(\widehat{\beta}_j)}\right] \longrightarrow 1-\alpha \qquad \text{při } n \to \infty.$$

Teď popíšeme způsob, jak testovat významnost více regresorů současně. Obvyklý postup je založen na *testu poměrem věrohodností* (*Wilksův test*) (viz Zvára, 2008, str. 232), který porovnává hodnoty logaritmické věrohodností funkce pro $\widehat{\boldsymbol{\beta}}_n$ a $\tilde{\boldsymbol{\beta}}_n$ pomocí statistiky

$$LR = 2 \cdot \left(\ell_n(\widehat{\boldsymbol{\beta}}_n) - \ell_n(\tilde{\boldsymbol{\beta}}_n)\right) = 2 \cdot \ln\left(\frac{L_n\left(\widehat{\boldsymbol{\beta}}_n\right)}{L_n\left(\tilde{\boldsymbol{\beta}}_n\right)}\right),$$

kde $\tilde{\boldsymbol{\beta}}_n$ je maximálně věrohodný odhad podmodelu. Podmodelem rozumíme model, který v sobě zahrnuje jen část regresorů z původního modelů, tj. $\tilde{\boldsymbol{\beta}}_n \subseteq \widehat{\boldsymbol{\beta}}_n$. Výrazu $\frac{L_n(\widehat{\boldsymbol{\beta}}_n)}{L_n(\tilde{\boldsymbol{\beta}}_n)}$ říkáme *věrohodnostní poměr* (angl. *likelihood ratio*). Za platnosti testovaného podmodelu má LR statistika asymptoticky chí kvadrát rozdělení o v stupních volností χ^2_v, kde v je rozdíl počtů parametrů v porovnávaných modelech (viz Zvára, 2008, str. 179).

Teď můžeme popsat náš algoritmus, který určí, které regresory jsou pro náš model významné. Tomuto algoritmu říkáme *dopředná postupná selekce* (angl. *forward stepwise selection*) (viz Hosmer a kol., 2013, odst. 4.3.1). Tento algoritmus funguje tak, že pomocí výše popsané statistiky postupně přidává do modelu jenom nejvýznamnější regresory a kontroluje významnost dříve přidaných a vynechává z nich nejvíce nevýznamné. Algoritmus se skládá ze 3 částí: *Přidání regresorů*, *Zpětná eliminace* a *Finální kontrola*.

Zaveďme značení, které budeme v tomto algoritmu používat. Nechť $m_{(0)}$ je začáteční podmnožina indexů parametrů nejmenšího uvažovaného modelu. Označím s jako číslo iterace v algoritmu a $m_{(s)} \subseteq \{0, 1, \ldots, k\}$ je množina indexů parametrů modelu v s-té iteraci. Stále platí, že j je index parametru, $j = 0, \ldots, k$. Nechť

$\ell_{m_{(s)}}$ je logaritmická věrohodností funkce v s-té iteraci v modelu, který obsahuje parametry s indexy z množiny $m_{(s)}$ a $\ell_{m_{(s)}\cup\{j\}}$ je logaritmická věrohodností funkce v s-té iteraci v modelu, který obsahuje j-tý parametr a parametry s indexy z množiny $m_{(s)}$. Krok Přidání parametrů má hladinu testu α_F, krok Zpětná eliminace má hladinu testu α_B a krok Finální kontrola má hladinu testu α_{st}. Defaultně $s = 0$ a $m_{(0)} = \{0\}$.

Poznámka 2.4. Obvykle α_F, α_B jsou větší než standardní $\alpha_{st} = 0{,}05$. Jejích hodnoty se pohybují kolem 0,15 pro α_F a kolem 0,20 pro α_B.

Algoritmus 2.1. Začínáme s $m_{(0)} = \{0\}$ a $s = 0$.

Přidání parametrů:

1. Pro každé j, které není v $m_{(s)}$, spočítáme $LR_j^{(s)} = 2\left(\ell_{m_{(s)}\cup\{j\}} - \ell_{m_{(s)}}\right)$. Pak pro dané j spočítáme p-hodnotu jako $p_j^{(s)} = \mathsf{P}\left[G > LR_j^{(s)}\right]$, kde G je náhodná veličina s chí kvadrát rozdělením o v stupních volností χ^2_v. V případě WOE modelu $v = 1$ a v případě modelu Plné logistické regrese $v = z_j - 1$, kde z_j je počet kategorií u j-tého regresoru.

2. Nechť $\widehat{j}^{(s)} = \underset{j\notin m_{(s)}}{\arg\min}\, p_j^{(s)}$ je index parametru s nejmenší p-hodnotou $p_j^{(s)}$ v s-té iteraci a $\widehat{p}^{(s)}$ je nejmenší p-hodnota, která odpovídá indexu $\widehat{j}^{(s)}$. Pokud $\widehat{p}^{(s)} \geq \alpha_F$, tak jdeme do kroku Finální kontrola. Pokud $\widehat{p}^{(s)} < \alpha_F$, tak:

   2.1. $s \overset{\text{def}}{=} s + 1$.

   2.2. $m_{(s)} \overset{\text{def}}{=} m_{(s-1)} \cup \{\widehat{j}^{(s-1)}\}$.

   2.3. jdeme do kroku Zpětná eliminace.

Zpětná eliminace:

1. Pro každé j, které je v $m_{(s-1)}$, spočítáme $LR_j^{(s)} = 2\left(\ell_{m_{(s)}} - \ell_{m_{(s)}\setminus\{j\}}\right)$. Pak pro dané j spočítáme p-hodnotu jako $p_j^{(s)} = \mathsf{P}\left[G > LR_j^{(s)}\right]$.

2. Nechť $\widehat{j}^{(s)} = \underset{j\in m_{(s-1)}}{\arg\max}\, p_j^{(s)}$ a $\widehat{p}^{(s)}$ je největší p-hodnota odpovídající indexu $\widehat{j}^{(s)}$. Pokud $p_j^{(s)} < \alpha_B$, tak se vrátíme do kroku Přidání parametrů. Pokud $p_j^{(s)} \geq \alpha_B$, tak:

   2.1. $m_{(s)} \overset{\text{def}}{=} m_{(s)} \setminus \{\widehat{j}^{(s)}\}$.

   2.2. jdeme zpátky do kroku Přidání parametrů.

Finální kontrola:

1. Pro každé j, které je v $m_{(s)}$, spočítáme $LR_j^{(s)} = 2\left(\ell_{m_{(s)}} - \ell_{m_{(s)}\setminus\{j\}}\right)$. Pak pro dané j spočítáme p-hodnotu jako $p_j^{(s)} = \mathsf{P}\left[G > LR_j^{(s)}\right]$.

2. Nechť $\widehat{j}^{(s)} = \underset{j\in m_{(s-1)}}{\arg\max}\, p_j^{(s)}$ a $\widehat{p}^{(s)}$ je největší p-hodnota odpovídající indexu $\widehat{j}^{(s)}$. Pokud $p_j^{(s)} < \alpha_{st}$, tak se algoritmus zastaví. Pokud $p_j^{(s)} \geq \alpha_{st}$, tak:

2.1. $s \stackrel{\text{def}}{=} s + 1$.

2.2. $m_{(s)} \stackrel{\text{def}}{=} m_{(s-1)} \setminus \{\widehat{j}^{(s-1)}\}$.

2.3. zopakujeme krok Finální kontrola.

Krok Přidání parametrů se dá interpretovat tak, že testujeme hypotézu o platnosti menšího modelu oproti alternativě, že platí větší model. Počítáme p-hodnotu pro každý nový potenciální parametr a vybíráme ten, který „nejvíce“ porušuje hypotézu. Pokud tato hodnota bude menší než α_F, tak přidám příslušný parametr do modelu. V tomto algoritmu nepracujeme s „klasickými“ p-hodnotami, které se používají při testování hypotéz. Jedná se spíš o indikátory „relativní významnosti“ parametrů. V modelu Plné logistické regrese vektor parametrů má tvar $\boldsymbol{\beta}^\top = \left(\beta_0, \beta_{1_1}, \beta_{2_1}, \ldots, \beta_{z_1-1}, \ldots, \beta_{1_j}, \ldots, \beta_{z_j-1}, \ldots, \beta_{1_k}, \ldots, \beta_{z_k-1}\right)$, kde z_j je počet kategorií j-tého kategoriálního regresoru ($z_j = 2$, pokud j-tý regresor je nominální veličina). Když počítáme $\ell_{m_{(s)} \cup \{j\}}$ nebo $\ell_{m_{(s)} \setminus \{j\}}$ v tomto modelu, tak jako j-tý parametr bereme všechny parametry s příslušnými indexy $1_j, \ldots, z_j - 1$. V tomto modelu v se rovná $z_j - 1$, protože u matice regresorů počet sloupců u každého kategoriálního regresoru je o 1 menší než počet kategorií daného regresoru.

Zpětná eliminace se provádí, protože přidáním nového regresoru do modelu můžeme snížit významnost starých regresorů. V této části znovu testujeme hypotézu o platnosti menšího modelu oproti alternativě, že platí větší model. Vzhledem k tomu, že obvykle nechceme vynechávat parametry, tak spočítáme p-hodnotu pro každý starý parametr a vybereme ten, který má největší p-hodnotu. Pokud tato hodnota bude menší než α_B, tak to znamená, že všechny staré regresory jsou stále významné v našem modelu. Pokud tato hodnota bude větší než α_B, tak odstraníme příslušný regresor, neboť on „nejméně“ porušuje hypotézu.

V kroku Finální kontrola ještě jednou otestujeme všechny významné regresory ale už na hladině α_{st}. Na výstupu dostaneme finální skupinu regresorů, které jsou pro náš model významné.

Poznámka 2.5. Existují další metody hledání nejvíce odpovídajícího modelu. Například: *Výběr nejlepší podmnožiny* (angl. *Best subset selection*)(viz Hosmer a kol., 2013, odst. 4.3), metoda *Skoků a Hranic* (angl. *Leaps and Bounds*)(viz Furnival a Wilson, 1974, str. 499–511) a také *Zpětná postupná selekce* (angl. *Backward stepwise selection*) (viz Hosmer a kol., 2013, odst. 4.3).

3. Rozhodovací stromy

Rozhodovací stromy jsou druhou metodou predikce závislé proměnné na základě regresorů. V této kapitole vycházíme hlavně z práce Hastie, Tibshirani a Friedmana (Hastie, Tibshirani a Friedman, 2009, odst. 9–10).

3.1 Úvod do rozhodovacích stromů

Metoda analýzy dat pomocí rozhodovacích stromů je dobře pochopitelná a ilustrativní. Obecně rozhodovací stromy fungují tak, že dělí množinu hodnot regresorů na disjunktní podmnožiny a pak přiřazují každé z těchto podmnožin nějakou konstantu, která je odhadem závislé proměnné.

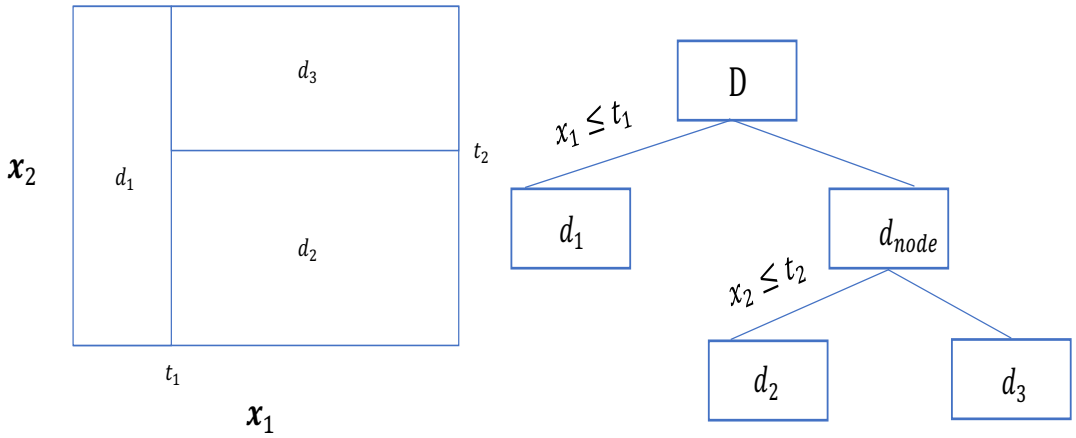
Definice 3.1. *Nechť* $\mathrm{D} \subseteq \mathbb{R}^k$ *je množina hodnot regresorů* $\boldsymbol{x}_i$*, kde* $i = 1, \ldots, n$*, a* $\Psi = \{(d_1, c_1), \ldots, (d_m, c_m)\}$ *je množina dvojic* (d_s, c_s)*, kde* $d_1, \ldots, d_m$ *jsou disjunktní podmnožiny množiny* D*, kterým jsou přiřazeny příslušné hodnoty* c_s*, kde* $s = 1, \ldots, m$*. Pak definujeme strom (angl. tree) jako funkci* T*, která přiřazuje vektoru* k *regresorů* $\boldsymbol{x}_i^\top = (x_{i,1}, \ldots, x_{i,k})$ *hodnotu* c_s *pomocí následujícího předpisu:*

$$T(\boldsymbol{x}_i) = \sum_{s=1}^{m} c_s \cdot \mathbb{I}[x_i \in d_s].$$

Množinu dvojic $\{(d_1, c_1), \ldots, (d_m, c_m)\}$ budeme zkrátka značit $\{(d_s, c_s)\}_{s=1}^m$ a budeme říkat, že Ψ je *parametr stromu* T.

Poznámka 3.1. Pokud pracujeme s odezvou, která je nominální proměnná, tj. hodnoty c_s, $s = 1, \ldots, m$, jsou nominální veličiny, tak funkce T je prediktor ve smyslu definice 1.3. Pokud odezva je kategoriální proměnná, tj. hodnoty c_s jsou kategoriální veličiny, tak funkce T je klasifikátor ve smyslu definice 1.4. V následujících třech podkapitolách místo pojmů prediktor a klasifikátor budeme používat pojem strom, který nebude mít úplnou souvislost se stromem z teorie grafů.

Ukážeme si, jak obecně fungují rozhodovací stromy na následujícím ilustrativním příkladu.



Obrázek 3.1: Proces dělení množiny D.

Příklad 3.1. Nechť odezva závisí na dvou regresorech x_1 a x_2. Rozdělím množinu hodnot regresorů D na 3 disjunktní podmnožiny následujícím způsobem. Zaprvé, rozdělíme množinu D na 2 podmnožiny: $x_1 \leq t_1$ a $x_1 > t_1$. Pak podmnožinu $x_1 > t_1$ rozdělíme na další dvě podmnožiny: $x_2 \leq t_2$ a $x_2 > t_2$. Tím dostaneme 3 disjunktní podmnožiny d_1, d_2 a d_3. V každé podmnožině bychom predikovali naši odezvu jako konstantu c_s, $s = 1, 2, 3$. Znázorníme tento proces (viz Obrázek 3.1). Obrázek 3.1 zobrazuje stejné operace dělení množiny D ale různými způsoby. Pravý obrázek je přehlednější, protože na něm lépe vidíme, jak probíhá proces dělení. Navíc, pokud bychom měli více nezávislých proměnných, tak by se nám levý obrázek těžko kreslil, zatímco pravý obrázek se dá vždycky jednoduše nakreslit bez ohledu na počet regresorů. Z pravého obrázku můžeme také vidět, proč takovým metodám říkáme stromy.

Definice 3.2. *Mějme strom T s parametrem $\Psi = \{(d_s, c_s)\}_{s=1}^{m}$, kde $d_1, \ldots, d_m$ jsou disjunktní podmnožiny množiny hodnot regresorů* D*, kterým jsou přiřazeny příslušné hodnoty c_s, kde $s = 1, \ldots, m$. Pak*

a) množině D *se říká kořen stromu (angl. root);*

b) množině regresorů, které splňují příslušné podmínky dělení, říkáme vnitřní uzel stromu (angl. internal node). Podmnožinám vnitřního uzlu říkáme následníky uzlu;

c) disjunktním podmnožinám d_s neboli vnitřním uzlům, které se dál nedělí, říkáme listy stromu (angl. terminal nodes).

Poznámka 3.2. Na pravém Obrázku 3.1 můžeme vidět, že „větvení“ stromu se začíná kořenem D, který rozdělíme na 2 vnitřní uzly d_1 a d_{node}. Uzel d_1 se stane listem, zatímco uzel d_{node} bude mít 2 následníky d_2 a d_3, které budou zároveň listy stromu.

Abychom mohli „vysázet“ strom, musíme odpovědět na 3 hlavní otázky:

- Jak spočítat hodnoty listů?
- Jak určit podmínky dělení stromu?
- Jak velký by měl být strom?

Existují různé metody konstrukce rozhodovacích stromů, my ale popíšeme jenom 3 metody: algoritmus *CART* (*Classification and Regression Trees*), algoritmus *CHAID* (*Chi-square Automatic Interaction Detector*) a aplikace stromů v metodě *Boost*.

3.2 Regresní stromy

Konstrukce stromu pomocí algoritmu CART závisí na tom, jestli odezva je nominální nebo kategoriální proměnná. V této podkapitole předpokládáme, že odezva je nominální proměnná. Stromu, který byl zkonstruován pro nominální odezvu, budeme říkat *regresní strom*.

Začneme s popisem toho, jak určit podmínky dělení stromu. V tomto kroku chceme najít takovou kombinaci regresoru a jeho hodnoty, která „nejlépe“ dělí

množinu D na 2 podmnožiny. Proto zavedeme algoritmus, který hledá nejlepší dělení množiny D pomocí *reziduálního součtu čtverců* RSS_s, který je definován jako

$$RSS_s = \sum_{i:\, \boldsymbol{x}_i \in d_s} (y_i - c_s)^2 \qquad \text{pro } s = 1, \ldots, m.$$

Vzhledem k tomu, že jako minimalizační kritérium používáme reziduální součet čtverců, tak nejlepší odhad závislé proměnné je aritmetický průměr hodnot závislých proměnných y_i, kde hodnoty regresorů $\boldsymbol{x}_i$ patří do podmnožiny d_s, $s = 1, \ldots, m$ a $i = 1, \ldots, n$:

$$c_s = \text{ave}\,(y_i \mid \boldsymbol{x}_i \in d_s) = \frac{1}{N_s} \sum_{i:\, \boldsymbol{x}_i \in d_s} y_i,$$

kde N_s je počet pozorování, které patří do podmnožiny d_s.

Algoritmus 3.1. Algoritmus pro nejlepší dělení množiny D začínáme s kořenem stromu D.

1. Pokud regresor je nominální nebo ordinální kategoriální proměnná, tak jdeme do kroku 1.1. Pokud regresor je neordinální kategoriální proměnná, tak jdeme do kroku 1.2.

   1.1. Rozdělíme množinu D na dvě podmnožiny $d_1(j,t) = \{\boldsymbol{x}_i : x_{i,j} \leq t\}$ a $d_2(j,t) = \{\boldsymbol{x}_i : x_{i,j} > t\}$ tak, aby $j \in \{1, \ldots, k\}$ a $t \in$ D byly řešením:

   $$\min_{j,t} \left[\sum_{i:\, \boldsymbol{x}_i \in d_1(j,t)} (y_i - c_1)^2 + \sum_{i:\, \boldsymbol{x}_i \in d_2(j,t)} (y_i - c_2)^2 \right],$$

   kde $c_1 = \text{ave}\,(y_i \mid \boldsymbol{x}_i \in d_1(j,t))$ a $c_2 = \text{ave}\,(y_i \mid \boldsymbol{x}_i \in d_2(j,t))$.

   1.2. Rozdělíme množinu D na podmnožiny $d_1(j,t) = \{\boldsymbol{x}_i : x_{i,j} = t\}$ a $d_2(j,t) = \{\boldsymbol{x}_i : x_{i,j} \neq t\}$ tak, aby j a t byly řešením

   $$\min_{j,t} \left[\sum_{i:\, \boldsymbol{x}_i \in d_1(j,t)} (y_i - c_1)^2 + \sum_{i:\, \boldsymbol{x}_i \in d_2(j,t)} (y_i - c_2)^2 \right]$$

2. Když najdeme příslušné j a t, tak dostaneme 2 podmnožiny: d_1 a d_2. Zopakujeme krok 1. na podmnožině d_1 a na podmnožině d_2. Po dělení těchto podmnožin dostaneme další 4 podmnožiny. Tento proces dělení podmnožin bude probíhat, dokud nenarazíme na nějaké *zastavovací pravidlo*. V našem případě takové pravidlo bude minimální počet pozorování v listu. To znamená, že algoritmus bude dělit podmnožiny, dokud v každém listu nebude méně než nějaký určitý počet pozorování.

Jako výsledek algoritmu 3.1 dostaneme strom s nějakým množstvím listů. Jinými slovy, najdeme vhodné dělení množiny D na podmnožiny d_s a tedy i parametr stromu $\Psi = \{(d_s, c_s)\}_{s=1}^m$.

Zbývá zodpovědět poslední otázku o velikosti stromu. Velikost stromu určuje komplexitu modelu. Příliš velký strom může vést k přefitování, zatímco malý strom může nezachytit důležitou strukturu. Než začneme popisovat jeden ze způsobů určování optimální velikosti stromu, zavedeme následující definice.

Definice 3.3. *Nechť T_0 je strom. Pak definujeme podstrom (angl. subtree) T jako libovolný strom, který můžeme dostat prořezáváním stromu T_0, tj. libovolným počtem zkolabování jeho vnitřních uzlů.*

Zkolabováním uzlu rozumíme odstranění jeho následníků, takže zkolabovaný uzel se stává listem. Fakt, že strom T je podstromem stromu T_0 budeme značit $T \subset T_0$.

Definice 3.4. *Nechť T je strom s parametrem $\Psi = \{(d_s, c_s)\}_{s=1}^{m}$, $T \subset T_0$. Pak definujeme kritérium cenové náročnosti $C_\alpha(T)$ (angl. cost-complexity criterion) jako funkci*

$$C_\alpha(T) = \alpha \cdot m + \sum_{s=1}^{m} \sum_{i:\, \boldsymbol{x}_i \in d_s} (y_i - c_s)^2,$$

kde $\alpha \geq 0$ je tzv. ladící parametr (angl. tuning parameter).

Pomocí metody *minimálního prořezávání cenové náročnosti* (angl. *minimal cost-complexity pruning*) (viz Breiman a kol., 1984, odst. 3.3) dostaneme pro každé $\alpha > 0$ unikátní nejmenší podstrom $T \subset T_0$, který minimalizuje naše kritérium $C_\alpha(T)$.

Zjednodušeně řečeno metoda minimálního prořezávání cenové náročnosti funguje tak, že na začátku sestrojíme velký strom T_0, který pak začneme „prořezávat“, dokud nenarazíme na kořen. Prořezávání probíhá tak, že postupně kolabujeme takové vnitřní uzly, zkolabování kterých vede k nejmenšímu přírůstku v hodnotě $\sum_{s=1}^{m} \sum_{i:\, \boldsymbol{x}_i \in d_s} (y_i - c_s)^2$. Tím dostaneme konečnou posloupnost podstromů. Dá se ukázat, že pro každé $\alpha \geq 0$ najdeme v této posloupnosti podstromů nejmenší podstrom, který minimalizuje kritérium cenové náročnosti. Velké hodnoty α vedou k menším velikostem stromu, a naopak. Všimneme si, že pokud $\alpha = 0$, tak dostáváme původní strom T_0.

Optimální velikost stromu budeme určovat pomocí následujícího algoritmu (viz Breiman a kol., 1984, odst. 8.4–8.5)

Algoritmus 3.2.

1. Sestrojíme velký strom T_0 pomocí algoritmu 3.1, ve kterém jako zastavovací pravidlo použijeme počet pozorování v listu. To znamená, že „větvení“ stromu zastavíme, pokud v každém listu bude méně než N_{min} pozorování, kde N_{min} je nějaký předem stanovený minimální počet pozorování v každém listu.

2. Pomocí metody minimálního prořezávání cenové náročnosti dostaneme unikátní posloupnost podstromů

   $$T_0 \supset T_{\alpha_1} \supset T_{\alpha_2} \supset \cdots \supset T_{\alpha_p} \supset T_{root},$$

   kde T_{root} je strom, který se sestává jenom z kořenu, T_0 je strom z kroku 1. a $\alpha_1 < \cdots < \alpha_p$ jsou ladící parametry. Potřebujeme vybrat strom z této posloupnosti, který má nejoptimálnější velikost.

3. Rozdělíme náhodně náš učební vzorek $L = \{(y_1, \boldsymbol{x}_1), \ldots, (y_n, \boldsymbol{x}_n)\}$ na 2 učební vzorky L_1 a L_2. Pak zopakujeme krok 1. a krok 2. na učebním vzorku L_1. Tím dostaneme posloupnost podstromů

   $$T_0^{(L_1)} \supset T_{\alpha_1}^{(L_1)} \supset \cdots \supset T_{\alpha_p}^{(L_1)} \supset T_{\alpha_{root}}^{(L_1)}.$$

Pak pro každé α_v spočítáme hodnotu $R\left(T_{\alpha_v}^{(L_1)}\right)$ jako

$$R\left(T_{\alpha_v}^{(L_1)}\right) = \frac{1}{N_{L_2}} \sum_{i:\,(y_i,\boldsymbol{x}_i)\in L_2} \left[y_i - T_{\alpha_v}^{(L_1)}(\boldsymbol{x}_i)\right]^2,$$

kde N_{L_2} je počet dvojic množiny L_2 a $v = 1, \ldots, p$. Označím $\widehat{\alpha}$ jako ladící parametr, kterému odpovídá nejmenší hodnota $R\left(T_{\alpha_v}^{(L_1)}\right)$.

4. Z původní posloupnosti podstromů $T_0 \supset T_{\alpha_1} \supset T_{\alpha_2} \supset \cdots \supset T_{\alpha_p} \supset T_{root}$ vybereme takový strom T_{α_v}, ladící parametr kterého koresponduje s $\widehat{\alpha}$.

Na výstupu algoritmu 3.2 dostaneme strom $T_{\widehat{\alpha}}$ s optimální velikostí a s příslušným parametrem Ψ.

Poznámka 3.3. Existují také další způsoby, jak můžeme určovat velikost stromu, např. *křížová validace* (angl. *cross validation*) (viz Hastie a kol., 2009, odst. 7.10).

3.3 Klasifikační stromy

V této podkapitole popíšeme metodu konstrukce stromu pomocí algoritmu CART pro kategoriální odezvu. Kategoriální závislá proměnná nabývá konečně mnoha hodnot $1, \ldots, H$, které vyjadřují příslušnost k nějaké kategorii. Stromu, který byl zkonstruován pro kategoriální odezvu, budeme říkat *klasifikační strom*.

Metoda konstrukce klasifikačního stromu je obdobná metodě konstrukce regresního stromu. Zaprvé, označme $p_{s,h}$ jako poměr h-té kategorie v s-tém listu, tj. platí

$$p_{s,h} = \frac{1}{N_s} \sum_{i:\,\boldsymbol{x}_i\in d_s} \mathbb{I}\left[y_i = h\right] \qquad \text{pro } s = 1, \ldots, m \text{ a } h = 1, \ldots, H.$$

Hodnota listu c_s v algoritmu CART (v případě kategoriální odezvy) je hodnota kategorie, která má největší poměr v listu d_s, tj. platí

$$c_s = \arg\max_h p_{s,h} \qquad \text{pro } s = 1, \ldots, m.$$

Podmínky dělení klasifikačního stromu určíme pomocí algoritmu 3.1. U regresních stromů v algoritmu 3.1 jsme jako minimalizační kritérium používali reziduální součet čtverců, které se v případě klasifikačního stromu nehodí. V klasifikačních stromech jako minimalizační kritérium používáme jedno ze tří kritérií: *klasifikační chybu* ME_s (angl. *misclassification error*), *Giniho nečistotu* GI_s (angl. *Gini impurity*) a *entropii* EN_s (angl. *entropy*), které jsou definovány jako

$$\begin{aligned} ME_s &= 1 - p_{s,c_s}, \\ GI_s &= \sum_{h=1}^{H} p_{s,h} \cdot (1 - p_{s,h}), \\ EN_s &= -\sum_{h=1}^{H} p_{s,h} \cdot \ln p_{s,h}, \end{aligned}$$

kde platí, že $p_{s,h} \cdot \ln p_{s,h} = 0$ pro $p_{s,h} \in \{0, 1\}$, $s = 1, \ldots, m$. Giniho nečistota a entropie se používají častěji, protože jsou citlivější ke změnám hodnot poměrů kategorií v listech. Takže pokud jako minimalizační kritérium budeme používat entropii, tak první krok algoritmu 3.1 vypadá tak:

1. Pokud regresor je nominální nebo ordinální kategoriální proměnná, tak jdeme do kroku 1.1. Pokud regresor je neordinální kategoriální proměnná, tak jdeme do kroku 1.2.

   1.1. Rozdělíme množinu D na dvě podmnožiny $d_1(j,t) = \{\boldsymbol{x}_i : x_{i,j} \leq t\}$ a $d_2(j,t) = \{\boldsymbol{x}_i : x_{i,j} > t\}$ tak, aby $j \in \{1,\ldots,k\}$ a $t \in$ D byly řešením:

$$\min_{j,t} \left[-\sum_{h=1}^{H} p_{1,h} \cdot \ln p_{1,h} - \sum_{h=1}^{H} p_{2,h} \cdot \ln p_{2,h} \right],$$

   kde

$$p_{1,h} = \frac{1}{N_{d_1(j,t)}} \sum_{i:\, \boldsymbol{x}_i \in d_1(j,t)} \mathbb{I}\,[y_i = h],$$
$$p_{2,h} = \frac{1}{N_{d_2(j,t)}} \sum_{i:\, \boldsymbol{x}_i \in d_2(j,t)} \mathbb{I}\,[y_i = h].$$

   1.2. Rozdělíme množinu D na podmnožiny $d_1(j,t) = \{\boldsymbol{x}_i : x_{i,j} = t\}$ a $d_2(j,t) = \{\boldsymbol{x}_i : x_{i,j} \neq t\}$ tak, aby j a t byly řešením

$$\min_{j,t} \left[-\sum_{h=1}^{H} p_{1,h} \cdot \ln p_{1,h} - \sum_{h=1}^{H} p_{2,h} \cdot \ln p_{2,h} \right].$$

Druhý krok algoritmu 3.1 zůstává beze změn. Velikost klasifikačního stromu určíme pomocí metody minimálního prořezávání cenové náročnosti s adaptací pro klasifikační stromy (viz Breiman a kol., 1984, odst. 3.1–3.4).

3.4 CHAID

V této podkapitole popíšeme metodu konstrukce stromu pomocí algoritmu CHAID (viz Kass, 1980). Při popisu algoritmu CHAID vycházíme z informací v příručce k softwaru SPSS (viz SPSS 16.0 Algorithms, 2007, str. 744–752). Tento algoritmus se dá aplikovat jak na nominální tak i na kategoriální odezvu, ale v této práci popíšeme algoritmus CHAID jenom pro dvoukategoriální neordinální odezvu. V tomto modelu všechny regresory musí být kategoriálními proměnnými.

Algoritmus CHAID umožňuje dělit vnitřní uzly na víc než 2 následníky a skládá se ze 2 částí: *Sloučení* (angl. *Merging*) a *Rozdělení* (angl. *Splitting*).

Vysvětlíme značení, které budeme v tomto algoritmu používat. Mějme učební vzorek $L = \{(y_1, \boldsymbol{x}_1), \ldots, (y_n, \boldsymbol{x}_n)\}$, kde $\boldsymbol{x}_i^\top = (x_{i,1}, \ldots, x_{i,k})$, $i = 1, \ldots, n$, je vektor k regresorů a platí, že každý j-tý regresor $\boldsymbol{x}_j^\top = (x_{1,j}, \ldots, x_{n,j})$ má m_j kategorií, $j = 1, \ldots, k$. Nechť $p_{merge,j}$ je p-hodnota odpovídající části sloučení kategorií regresoru $\boldsymbol{x}_j$ a $p_{split,j}$ je p-hodnota odpovídající části rozdělení kategorií regresoru $\boldsymbol{x}_j$. Označím α_{merge} jako hladinu testu pro část sloučení a α_{split} jako hladinu testu pro část rozdělení. Definice a způsob počítání p-hodnot uvedeme po algoritmu CHAID.

Algoritmus 3.3. Algoritmus CHAID začínáme s kořenem stromu D.

Sloučení: pro všechny $j = 1, \ldots, k$ postupujeme následovně

1. Pokud $\boldsymbol{x}_j$ má jenom 2 kategorie, tj. $m_j = 2$, tak označíme $\boldsymbol{x}_j \overset{\text{def}}{=} \widehat{\boldsymbol{x}}_j$ a $m_j \overset{\text{def}}{=} \widehat{m}_j$ a pokračujeme krokem 4. Pokud $\boldsymbol{x}_j$ má víc než 2 kategorie, tak jdeme do kroku 2.

2. Pro všechny možné dvojice kategorií regresoru $\boldsymbol{x}_j$ spočítáme $p^{(w)}_{merge,j}$, kde $w = 1, \ldots, n_j$ a n_j je počet možných dvojic kategorií regresoru $\boldsymbol{x}_j$. Vybereme největší $p^{(w)}_{merge,j}$ a označíme ji jako $p_{merge,j}$. Pokud regresor $\boldsymbol{x}_j$ je ordinální proměnná, tak možné dvojice kategorií jsou takové dvojice, kategorie kterých jsou k sobě přilehlé.

3. Pokud $p_{merge,j} \leq \alpha_{merge}$, tak označíme $\boldsymbol{x}_j \overset{\text{def}}{=} \widehat{\boldsymbol{x}}_j$, $m_j \overset{\text{def}}{=} \widehat{m}_j$ a pokračujeme krokem 4. Pokud $p_{merge,j} > \alpha_{merge}$, tak sloučíme dvě kategorie odpovídající p-hodnotě $p_{merge,j}$. Sloučením dvou kategorií jsme překategorizovali regresor $\boldsymbol{x}_j$, takže dostaneme nově překategorizovaný regresor $\widehat{\boldsymbol{x}}_j$ s novým počtem kategorií $\widehat{m}_j$. Přeznačíme $\widehat{\boldsymbol{x}}_j \overset{\text{def}}{=} \boldsymbol{x}_j$, $\widehat{m}_j \overset{\text{def}}{=} m_j$ a jdeme zpátky do kroku 1.

4. Pro regresor $\widehat{\boldsymbol{x}}_j$ spočítáme p-hodnotu $p_{split,j}$. Když pro každé $j = 1, \ldots, k$ bude spočítána hodnota $p_{split,j}$, tak jdeme do kroku Rozdělení.

Rozdělení:

1. Vybereme regresor $\widehat{\boldsymbol{x}}_j$ s nejmenší p-hodnotou $p_{split,j}$, kterou označíme jako $\widehat{p}_{split,j}$.

2. Pokud $\widehat{p}_{split,j} > \alpha_{split}$, tak se vnitřní uzel stává listem. Pokud $\widehat{p}_{split,j} \leq \alpha_{split}$, tak:

   2.1. Rozdělíme vnitřní uzel (na začátku je to kořen stromu D) na $\widehat{m}_j$ podmnožin $d_l(j, t_l) = \{\boldsymbol{x}_i : x_{i,j} = t_l, i = 1, \ldots, n\}$, kde t_l je l-tá kategorie regresoru $\widehat{\boldsymbol{x}}_j$, $l = 1, \ldots, \widehat{m}_j$.

   2.2. Označím $\widehat{\boldsymbol{x}}_j \overset{\text{def}}{=} \boldsymbol{x}_j$ a $\widehat{m}_j \overset{\text{def}}{=} m_j$. Jdeme zpátky do kroku Sloučení, který aplikujeme na všechny podmnožiny $d_l(j, t_l)$.

Zastavovací pravidla:

1. Vnitřní uzel je „čistý“, tj. všechny regresory v tomto uzlu odpovídají jedné hodnotě odezvy. Vzhledem k tomu, že máme závislou proměnnou s alternativním rozdělením, tak čistý vnitřní uzel je taková množina d, pro kterou platí, že $d = \{\boldsymbol{x}_i : y_i = 1, i = 1, \ldots, n\}$ nebo $d = \{\boldsymbol{x}_i : y_i = 0, i = 1, \ldots, n\}$.

2. Všechny regresory vnitřního uzlu mají stejné hodnoty, tj. v tomto vnitřním uzlu se vyskytuje jenom jedna kategorie pro každý regresor.

3. Dosáhli jsme maximální velikost stromu nebo minimálního počtu pozorování ve vnitřním listu.

V části Sloučení chceme u každého regresoru sloučit takové kategorie, které jsou nejvíc „podobné“ vzhledem k závislé proměnné. Podobnost budeme určovat pomocí *Pearsonova chí-kvadrát testu*, neboli *testováním nezávislosti* χ^2 *testem*, který umožňuje testovat nulovou hypotézu, že dvě kategoriální veličiny jsou nezávislé. V této části algoritmu pro každou možnou dvojici kategorií regresoru $\boldsymbol{x}_j$

sestrojíme kontingenční tabulku 2×2 (viz Tabulka 3.1), která bude mít 2 určité kategorie regresoru po řádcích a 2 kategorie odezvy po sloupcích. Z kontingenční tabulky spočítáme *statistiku Pearsonova chí-kvadrát testu* Q_w^2, $w = 1, \ldots, n_j$, jako

$$Q_w^2 = \sum_{s=1}^{2} \sum_{l=1}^{2} \frac{\left(n_{l,s} - \frac{n_{l+} n_{+s}}{N}\right)^2}{\frac{n_{l+} n_{+s}}{N}},$$

kde N, $n_{l,s}$, n_{l+} a n_{+s} jsou hodnoty z příslušné kontingenční tabulky 2×2 (viz Tabulka 3.1). Pomocí hodnoty Q_w^2 spočítáme p-hodnotu $p_{merge,j}^{(w)}$ jako

$$p_{merge,j}^{(w)} = \mathsf{P}\left[\chi^2_{(2-1)(2-1)} > Q_w^2\right],$$

kde $\chi^2_{(2-1)(2-1)}$ je náhodná veličina z chí kvadrát rozdělení s $(2-1)\cdot(2-1)$ stupni volnosti. Pokud daná dvojice kategorií regresoru $\boldsymbol{x}_j$ „stejně predikuje“ odezvu, tak dostaneme velkou p-hodnotu $p_{merge,j}^{(w)}$. Proto v části Sloučení algoritmu CHAID pro každý regresor vybíráme největší p-hodnotu $p_{merge,j}^{(w)}$. Tímto způsobem sloučíme „podobné“ kategorie u každého regresoru a dostaneme nově překategorizované regresory $\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_k$ s odpovídajícími počty kategorií $\widehat{m}_1, \ldots, \widehat{m}_k$

Pro část Rozdělení algoritmu CHAID potřebujeme spočítat p-hodnoty $p_{split,j}$. Uděláme to tak, že zase sestrojíme kontingenční tabulku pro každý regresor, ale tentokrát po řádcích budou všechny kategorie regresoru $\widehat{\boldsymbol{x}}_j$. Pak z kontingenčních tabulek spočítáme statistiky Pearsonova chí-kvadrát testu Q_j^2 jako

$$Q_j^2 = \sum_{s=1}^{2} \sum_{l=1}^{\widehat{m}_j} \frac{\left(n_{l,s} - \frac{n_{l+} n_{+s}}{N}\right)^2}{\frac{n_{l+} n_{+s}}{N}},$$

kde $\widehat{m}_j$ je počet kategorií regresoru $\widehat{\boldsymbol{x}}_j$ a N, $n_{l,s}$, n_{l+}, n_{+s} jsou hodnoty z příslušné kontingenční tabulky (viz Tabulka 3.1). Hodnotu $p_{split,j}$ pak spočítáme jako

$$p_{split,j} = B_j \cdot \mathsf{P}\left[\chi^2_{(2-1)(\widehat{m}_j-1)} > Q_j^2\right],$$

kde $\chi^2_{(2-1)(\widehat{m}_j-1)}$ je náhodná veličina z chí kvadrát rozdělení s $(2-1)\cdot(\widehat{m}_j - 1)$ stupni volnosti a B_j je tzv. *multiplikátor Bonferroni*. V této části algoritmu hledáme takový regresor, který „nejvíc“ porušuje nulovou hypotézu, že daný regresor a odezva jsou nezávislé. Proto vybíráme regresor, kterému odpovídá nejmenší hodnota $p_{split,j}$. Podle tohoto regresoru budeme dělit příslušný vnitřní uzel.

	$y = 0$ $(s = 1)$	$y = 1$ $(s = 2)$	$\sum$
$x_{1_j} = t_1$	$n_{1,1}$	$n_{1,2}$	n_{1+}
$\ldots$	$\ldots$	$\ldots$	$\ldots$
$x_{l_j} = t_l$	$n_{l,1}$	$n_{l,2}$	n_{l+}
$\ldots$	$\ldots$	$\ldots$	$\ldots$
$x_{\widehat{m}_j} = t_{\widehat{m}_j}$	$n_{\widehat{m}_j,1}$	$n_{\widehat{m}_j,2}$	$n_{\widehat{m}_j+}$
$\sum$	n_{+1}	n_{+2}	N

Tabulka 3.1: Kontingenční tabulka.

Poznámka 3.4. Používáme multiplikátor Benferonni, protože v algoritmu CHAID narazíme na problém *násobného testování hypotéz* (angl. *multiple comparisons problem*). Jedná se o případ, kdy testujeme zároveň více hypotéz, což vede ke zvýšení *chyby prvního druhu* (zamítnutí pravdivé nulové hypotézy). Tento problém můžeme vyřešit tak, že podělíme hladinu testu na počet testovaných hypotéz. Ekvivalentní úprava je taková, že můžeme vynásobit p-hodnotu počtem testovaných hypotéz. B_j je počet toho, kolika způsoby můžeme rozdělit m_j kategorií regresoru $\boldsymbol{x}_j$ do $\widehat{m}_j$ kategorií. V případě ordinálního kategoriální regresoru platí

$$B_j = \binom{m_j - 1}{\widehat{m}_j - 1},$$

kde m_j je původní počet kategorií regresoru $\boldsymbol{x}_j$ a $\widehat{m}_j$ je počet kategorií tohoto regresoru po části sloučení. Pokud regresor je neordinální kategoriální proměnná, tak multiplikátor Bonferroni definován jako

$$B_j = \sum_{v=0}^{\widehat{m}_j - 1} (-1)^v \frac{(\widehat{m}_j - v)^{m_j}}{v!\,(\widehat{m}_j - v)!}.$$

Hodnoty listů v algoritmu CHAID určíme stejným způsobem jako v předchozí podkapitole a velikost stromu určíme pomocí křížové validace (viz Hastie a kol., 2009, odst. 7.10).

3.5 Boosting

Poslední metoda pro predikce závislé proměnné s alternativním rozdělením, kterou popíšeme v této práci, je metoda Boosting. Přestože Boosting není stromová metoda, popisujeme ji v kapitole „Rozhodovací stromy“, protože rozhodovací stromy budeme používat jako část Boosting algoritmu.

Metoda Boosting spočívá v tom, že vytváří „silný“ klasifikátor pomocí kombinování „slabých“ klasifikátorů. Slabým klasifikátorem rozumíme klasifikátor, jehož přesnost je jenom o trochu lepší než házení mincí. Boosting začíná tím, že odhaduje slabý klasifikátor na celém učebním vzorku a následně upravuje váhy pozorování na tomto učebním vzorku. Poté se odhaduje další slabý klasifikátor na upraveném učebním vzorku a zase se upravují váhy pozorování. Tento postup se opakuje, dokud není dosaženo požadované úrovně přesnosti anebo maximálního počtu iterací.

V této podkapitole popíšeme algoritmus *Diskrétní AdaBoost* (angl. *Discrete AdaBoost*), což je metoda Boosting pro predikce odezvy, která nabývá hodnot 1 nebo -1. Naším cílem je odhadnout slabé klasifikátory $t_1(\boldsymbol{x}_i), \ldots, t_M(\boldsymbol{x}_i)$ tak, aby jejich kombinace nám dala silný klasifikátor $T(\boldsymbol{x}_i)$:

$$T(\boldsymbol{x}_i) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m \cdot t_m(\boldsymbol{x}_i)\right),$$

kde M je počet iterací algoritmu a α_m jsou váhy slabých klasifikátorů. V Boostingu tyto váhy slouží k tomu, aby slabé klasifikátory s lepší přesností měly větší vliv na konečný silný klasifikátor, a naopak. V algoritmu Diskrétní AdaBoost se obecně dá použít různé slabé klasifikátory, my ale jako slabý klasifikátor budeme

používat tzv. *pařez* (angl. *stump*), což je klasifikační strom s dvěma listy. Jedná se o strom, který se větví jenom jednou podle jednoho regresoru. Slabé klasifikátory budou nabývat hodnot 1 nebo -1.

Algoritmus 3.4. První část algoritmu Diskrétní AdaBoost s adaptací pro pařezy bude určování slabých klasifikátorů, které se budou používat v každé iteraci. Pro $j = 1, \ldots, k$ pomocí prvního kroku algoritmu 3.1 s adaptací pro klasifikační stromy sestrojíme pařez $t_j(\boldsymbol{x}_i)$, což je klasifikační strom, který se dělí podle j-tého regresoru. Vzhledem k tomu, že slabé klasifikátory nabývají hodnot 1 a -1, tak překódujeme v učebním vzorku všechny odezvy rovné 0 na -1. Na začátku dáme každému pozorování v učebním vzorku $L = \{(y_1, \boldsymbol{x}_1), \ldots, (y_n, \boldsymbol{x}_n)\}$ váhu $w_i = \frac{1}{n}$, $i = 1, \ldots, n$.

Pak pro $m = 1, \ldots, M$:

1. Pro každý pařez $t_j(\boldsymbol{x}_i)$ spočítáme *chybovost* er_j (angl. *error rate*) jako

$$er_j = \sum_{i=1}^{n} w_i \cdot \mathbb{I}\left[y_i \neq t_j(\boldsymbol{x}_i)\right].$$

   Vybereme pařez s nejmenší chybovostí a označím ho $t_m(\boldsymbol{x}_i)$, tj. slabý klasifikátor v m-té iteraci.

2. Spočítáme váhu α_m klasifikátoru $t_m(\boldsymbol{x}_i)$ jako

$$\alpha_m = \ln\left(\frac{1 - er_j}{er_j}\right).$$

3. Přepočítáme váhy w_i jako

$$w_i \stackrel{\text{def}}{=} w_i \cdot \mathrm{e}^{\alpha_m \cdot \mathbb{I}[y_i \neq t_j(\boldsymbol{x}_i)]} \qquad \text{pro } i = 1, \ldots, n.$$

4. Nechť $N = \sum_{i=1}^{n} w_i$. Pak

$$w_i \stackrel{\text{def}}{=} \frac{w_i}{N} \qquad \text{pro } i = 1, \ldots, n.$$

   Upravíme tím váhy v učebním vzorku tak, aby jejích součet dával 1.

Po skončení tohoto algoritmu dostaneme finální klasifikátor $T(\boldsymbol{x}_i)$. Počet iterací M určíme pomocí křížové validace (viz Hastie a kol., 2009, odst. 7.10).

4. Srovnání kvality modelů

V této kapitole popíšeme, jak budeme porovnávat mezi sebou naše modely. Musíme si uvědomit, že vzhledem k tomu, že výstup prediktorů je odlišný od výstupu klasifikátorů, tak se prediktory porovnávají jiným způsobem než klasifikátory. Takže potřebujeme určit skupinu prediktorů a klasifikátorů.

4.1 Shrnutí predikčních modelů

V Kapitole 2 jsme popsali 3 stupně logistické regrese, kde obecný odhad pravděpodobnosti výskytu jevu vypadal jako

$$\widehat{\mathsf{P}}\left[Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \frac{\mathrm{e}^{\widehat{\boldsymbol{\beta}}^\top \boldsymbol{x}_i}}{1 + \mathrm{e}^{\widehat{\boldsymbol{\beta}}^\top \boldsymbol{x}_i}},$$

kde $\widehat{\boldsymbol{\beta}}$ je vektor odhadů parametrů a $\boldsymbol{x}_i$ je vektor regresorů z příslušné regresní matice $\mathbb{X}$, $i = 1, \ldots, n$. Vzhledem k tomu, že odhad pravděpodobnosti výskytu jevu je odhad střední hodnoty závislé proměnné s alternativním rozdělením, tak se můžeme dívat na odhad pravděpodobnosti výskytu jevu jako na odhad závislé proměnné, tj. platí

$$\widehat{\mathsf{P}}\left[Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \widehat{\mathrm{E}}\left[Y_i | \boldsymbol{X}_i = \boldsymbol{x}_i\right] = \widehat{y}_i = f^{(p)}\left(\boldsymbol{x}_i, w\right),$$

kde w by byl příslušný vektor odhadnutých parametrů $\widehat{\boldsymbol{\beta}}$. Takže všechny 3 modely logistické regrese jsou prediktory ve smyslu definice 1.3.

Připomeneme si také definici stromu:

$$T(\boldsymbol{x}_i) = \sum_{s=1}^{m} c_s \cdot \mathbb{I}[x_i \in d_s],$$

kde d_s, $s = 1, \ldots, m$, jsou disjunktní podmnožiny množiny hodnot regresorů D. V Kapitole 2.1 jsme si uvědomili, že pokud hodnoty c_s jsou nominální veličiny, tak funkce T je prediktor, zatímco pokud hodnoty c_s jsou kategoriální veličiny, tak funkce T je klasifikátor. Z toho plyne, že strom, který je zkonstruován pomocí metody CART (regresní strom), je prediktor, zatímco stromy, které jsou zkonstruovány pomocí CART (klasifikační strom) a CHAID, jsou klasifikátory. Takže platí:

$$T(\boldsymbol{x}_i) = \widehat{y}_i = \begin{cases} f^{(p)}\left(\boldsymbol{x}_i, w\right), & \text{pro CART (regresní strom),} \\ f^{(c)}\left(\boldsymbol{x}_i, w\right), & \text{pro CART (klasifikační strom) a CHAID,} \end{cases}$$

kde w by byl parametr příslušného stromu $\Psi = \{(d_s, c_s)\}_{s=1}^{m}$.

Z Kapitoly 3.5 už víme, že metoda Boosting spočívá ve vytváření silného klasifikátoru. Tedy platí

$$T(\boldsymbol{x}_i) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m \cdot t_m(\boldsymbol{x}_i)\right) = \widehat{y}_i = f^{(c)}\left(\boldsymbol{x}_i, w\right),$$

kde w by byl vektor vah $\alpha_1, \ldots, \alpha_M$.

Pro jednoduchost přidáme prediktorům a klasifikátorům krátké názvy, které budou vyplývat z názvů metod, pomocí kterých predikční modely byly zkonstruovány. Takže budeme mít 3 klasifikátory: CART-C (tj. klasifikační strom), CHAID a BOOST. Budeme mít také 4 prediktory: IND (tj. Nezávislý model), WOE (tj. WOE model), FLR (tj. Plný logistický model) a CART-R (tj. regresní strom).

Musíme si také uvědomit, že všechny prediktory a klasifikátory byly zkonstruovány na základě dat z učebního vzorku. V praxi se často používají tzv. *trénovací* a *testovací data*. Trénovací data slouží k vybudování modelu, zatímco testovací data slouží k testování tohoto modelu. V našem případě učební vzorek obsahoval jenom trénovací data. Takže srovnávat naše predikční modely budeme pomocí dat z jiného učebního vzorku (obsahujícího testovací data).

4.2 Srovnání klasifikátorů

Odhad závislé proměnné u CART-C a CHAID je hodnota kategorie, která má největší poměr v příslušném listu. Takže jako odhad odezvy budeme dostávat buď 1 anebo 0. U BOOST na výstupu dostáváme buď 1 anebo -1. Pro srovnatelnost s jinými klasifikátory překódujeme všechny odhady odezvy u BOOST rovné -1 na 0. Takže pokud otestujeme klasifikátor na testovacích datech, tak na výstupu dostaneme nějaké množství $\hat{1}$ (odezva, která byla odhadnuta jako 1) a nějaké množství $\hat{0}$ (odezva, která byla odhadnuta jako 0).

Klasifikátory budeme porovnávat pomocí tzv. *chybové matice* $\mathbb{M}$ (angl. *confusion matrix*):

$$\mathbb{M} = \begin{pmatrix} & 0 & 1 \\ \hat{0} & a & b \\ \hat{1} & c & d \end{pmatrix},$$

kde $\hat{0}$, $\hat{1}$ značí odhady závislé proměnné a 0, 1 značí skutečnou hodnotu závislé proměnné. Písmeno a je počet *pravdivě negativních* (angl. *true negative*) rozhodovaní, b je počet *falešně negativních* (angl. *false negative*) rozhodovaní, c je počet *falešně pozitivních* (angl. *false positive*) rozhodovaní a d je počet *pravdivě pozitivních* (angl. *true positive*) rozhodovaní. Počet falešně negativních rozhodovaní b znamená počet toho, kolikrát jsme odhadli odezvu jako kategorii 0, když ve skutečnosti patří do kategorie 1. Analogicky pro a, c, d.

Jako kritérium kvality klasifikátorů můžeme používat tzv. *chybovou míru* MR (angl. *misclassification rate*), která je definována jako

$$MR = \frac{b + c}{a + b + c + d},$$

kde a, b, c, d jsou prvky chybové matice $\mathbb{M}$. Chybová míra nám říká, jakou část odhadů jsme zařadili do nesprávných kategorií.

4.3 Srovnání prediktorů

Výstup našich prediktorů je číslo z intervalu $(0, 1)$. Abychom mohli přiřazovat našim odhadům kategorie, potřebujeme znát tzv. *rozhodovací hranici* ξ (angl.

cutoff). Pokud budeme vědět rozhodovací hranici, tak všechny odhady, které budou menší nebo rovny této hranici, budou zařazeny do kategorie 0, zatímco ostatní odhady budou zařazeny do kategorie 1. Takže pro každé ξ dostaneme nějaké množství $\hat{1}$ a nějaké množství $\hat{0}$. To znamená, že pro každé ξ budeme schopní sestrojit chybovou matici $\mathbb{M}$.

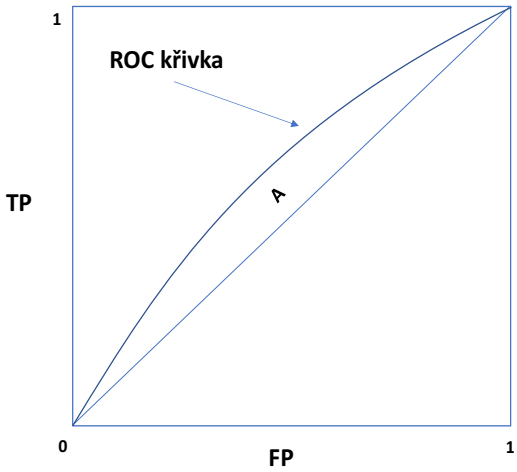
Prediktory budeme porovnávat pomocí tzv. *diverzifikační schopností*, což je v našem případě míra toho, jak dobře model odděluje závislé proměnné nabývajících hodnot 1 od závislých proměnných nabývajících hodnot 0. Míru diverzifikace můžeme graficky znázornit pomocí tzv. *ROC křivky* (angl. *Receiver Operating Characteristic curve*). Abychom mohli nakreslit ROC křivku, potřebujeme pro každé ξ spočítat *míru falešné pozitivity* FP (angl. *false positive rate*) a *míru pravdivé pozitivity* TP (angl. *true positive rate*) jako

$$FP = \frac{c}{a+c},$$
$$TP = \frac{d}{b+d},$$

kde a, b, c, d jsou prvky příslušné chybové matice. Míra falešné pozitivity nám říká, jakou část odhadů, které ve skutečnosti patří do kategorie 0, jsme zařadili do kategorie 1. Míra pravdivé pozitivity nám říká, jakou část odhadů, které ve skutečnosti patří do kategorie 1, jsme správně zařadili do kategorie 1. Křivku ROC pak definujeme jako množinu dvojic

$$\left\{\left(FP(\xi), TP(\xi)\right) \in \mathbb{R}^2, \xi \in (0,1)\right\}.$$

Ideálně by pro nás bylo najít takové ξ, že $FP(\xi) = 0$ a $TP(\xi) = 1$, což by znamenalo, že prediktor je schopen rozlišovat závislé proměnné nabývajících hodnot 1 od závislých proměnných nabývajících hodnot 0 s 100% přesností. V ideálním případě by křivka měla být co nejblíže levému hornímu rohu grafu (viz Obrázek 4.1), což by znamenalo, že model má vysokou diverzifikační schopnost. Naopak, křivka, která se blíží diagonále, znamená, že model má nízkou diverzifikační schopnost a výkon je náhodný.



Obrázek 4.1: ROC křivka.

Jako číselnou interpretaci ROC křivky budeme používat *Giniho koeficient G* (angl. *Gini coefficient*), což je dvojnásobek obsahu plochy mezi ROC křivkou a diagonálou jednotkového čtverce, tj. platí

$$G = 2 \cdot A,$$

kde A je obsah plochy mezi ROC křivkou a diagonálou jednotkového čtverce.

5. Aplikace na data

5.1 Vysvětlení dat

Jedna vietnamská společnost udělala reklamní kampaň na prodej hotovostních půjček. V rámci kampaně společnost rozesílala SMS zprávy, volala a používala různé reklamy na internetu. SMS zprávy a reklamy obsahovaly odkaz na jejich webovou stránku, přes kterou by nějaká osoba mohla požádat o úvěr.

Od této společnosti jsme dostali data. Jedná se o 54 486 pozorování (bez NA hodnot, tj. bez chybějících hodnot), ze kterých 279 jsou pozitivní výsledky, což je kolem 0,51 %. Pozitivním výsledkem rozumíme to, že osoba se nějakým způsobem dostala na webovou stránku (buď přes odkaz, nebo kvůli volání), podala žádost (tj. vyplnila formulář) a získala úvěr. Máme k dispozici 19 charakteristik (tj. regresorů), na základě kterých můžeme odhadovat pravděpodobnost výskytu pozitivního jevu. Jedná se o proměnné: request_amount, max_emi, F_income_type, income, F_credit_history, lead_age, F_device, F_viet_name, F_region, F_gender, F_operator, F_renumbered, F_ad, products, F_new_id_card, prev_tele, prev_sms, days_since_eligible, days_since_this_assign. Jejich popis najdeme v Příloze A. Přívlastek „F_“ znamená, že daná proměnná je kategoriální neordinální proměnná. Regresory, které daný přívlastek neobsahují, jsou kategoriální ordinální proměnné. V celé této kapitole jako názvy regresorů budeme používat zkratky, které byly použity při zpracování dat v programu R. V Příloze B najdeme grafy závislosti odezvy na jednotlivých regresorech.

5.2 Aplikace Nezávislého modelu

V Nezávislém modelu nepotřebujeme počítat odhady vektoru parametrů $\boldsymbol{\beta}$, protože předpokládáme, že $\boldsymbol{\beta}$ je jednotkový vektor. Takže nebudeme v tomto modelu určovat ani významnost regresorů. Nicméně, potřebujeme ověřit stabilitu odds_{ratio} jednotlivých regresorů. Začneme tím, že rozdělíme učební vzorek na trénovací a testovací data. Pak pro každou kategorii každého regresoru spočítáme odds_{ratio} na trénovacích datech a zvlášť na testovacích datech. Potom porovnáme mezi sebou tyto odds_{ratio}. Pokud rozdíly mezi těmito odds_{ratio} budou velké, tak to znamená, že odds_{ratio} jsou nestabilní.

V Tabulce 5.1 můžeme vidět, že většina regresorů jsou nestabilní. To může souviset s tím, že v našem učebním vzorku je jenom 0,51 % pozitivních jevů. V Nezávislém modelu budeme uvažovat jenom té regresory, kategorie kterých mají rozdíl v odds_{ratio} méně než 20 % (viz Tabulka 5.1). Takže budeme uvažovat jenom následující regresory: F_gender, F_renumbered a F_new_id_card.

Když už máme vybrané proměnné, tak můžeme počítat odhady pravděpodobnosti výskytu pozitivního jevu. Výpočet budeme provádět pomocí výrazu (2.5). Počítat budeme na testovacích hodnotách s použitím odds_{ratio} z trénovacího vzorku. Když sestrojíme ROC křivku (viz Obrázek 5.1), tak nám vyjde Giniho koeficient rovný 0,54. Vidíme, že přesnost Nezávislého modelu je jenom o trochu lepší než házení mincí.

Název regresoru	odds_{ratio} **na trénovacím vzorku**	odds_{ratio} **na testovacím vzorku**	**rozdíl v %**
request_amount:1	0,87	0,4	54,17
request_amount:2	0,99	1,04	-5,28
request_amount:3	0,95	1,25	-32,16
request_amount:4	1,03	1,06	-3,14
max_emi:1	0,93	0,4	56,63
max_emi:2	1,41	1,45	-2,59
max_emi:3	0,97	1,04	-6,94
F_income_type:BL	1,59	0,54	66,07
F_income_type:BNL	1,13	1,25	-10,59
F_income_type:LC	0,93	0,96	-3,38
F_income_type:LS	0,98	1,35	-37,44
F_income_type:OTH	0,74	0,51	31,47
income:1	0,84	0,86	-2,3
income:2	1,02	0,71	30,21
income:3	1,12	1,37	-21,65
F_credit_history:HB	1,47	1,44	2,05
F_credit_history:LDT	0,98	1,09	-11,23
F_credit_history:LPT	0,99	0,79	20,16
F_credit_history:NB	0,63	0,81	-29,46
lead_age:1	2,2	1,77	19,8
lead_age:2	1,2	1,25	-4,66
lead_age:3	0,4	0,54	-36,88
F_device:Android	0,8	0,86	-6,97
F_device:iPhone	1,39	1,24	11,17
F_device:Other	0,66	1,51	-129,12
F_device:Windows	0,88	1,17	-33,49
F_viet_name:NoSymbols	0,83	0,51	37,66
F_viet_name:Symbols	1,03	1,09	-5,57
F_region:Central	0,91	0,67	26,51
F_region:East	1,49	1,13	24,64
F_region:North	1,23	0,62	49,47
F_region:South	0,87	0,95	-8,52
F_region:West	0,89	1,4	-56,49
F_gender:F	1.25	1.06	15.32
F_gender:M	0.86	0.97	-12.47
F_operator:MobiFone	1,3	1,2	7,59
F_operator:Small operator	3,45	3,52	-1,96
F_operator:Vietnamobile	0,01	0,89	-88,11
F_operator:Viettel	0,79	0,77	1,82
F_operator:VinaPhone	1,19	1,23	-3,34
F_renumbered:reassigned	0,92	0,87	5,38
F_renumbered:unchanged	1,05	1,08	-2,82

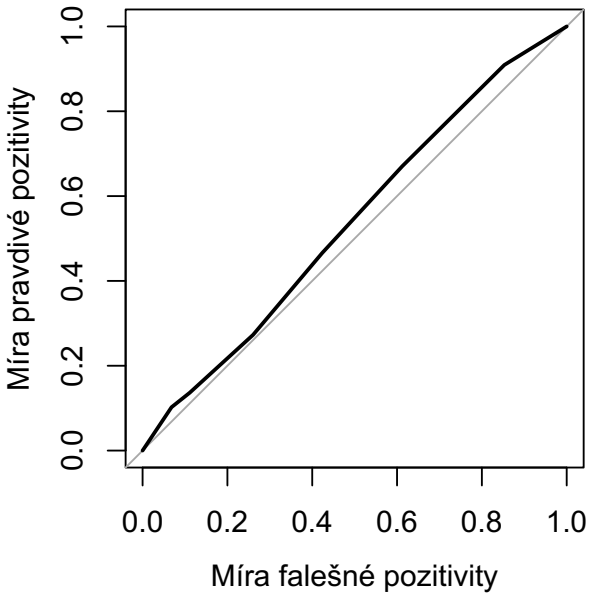
Název regresoru	odds_{ratio} **na trénovacím vzorku**	odds_{ratio} **na testovacím vzorku**	**rozdíl v %**
F_ad:FB	0,95	0,91	4,24
F_ad:GG	1,05	0,83	20,37
F_ad:Multi	0,29	1,92	-555,86
F_ad:NO VISIT	0,35	0,75	-113,14
F_ad:other	1,01	2,24	-120,5
products:1	1,06	0,96	9,91
products:2	0,91	1,14	-25,58
products:3	0,38	1,31	-246,84
F_new_id_card:New	1,22	1,17	4,18
F_new_id_card:Old	0,88	0,9	-3,05
prev_tele:1	2,15	1,3	39,54
prev_tele:2	1,04	1,1	-5,61
prev_tele:3	0,52	0,62	-19,49
prev_tele:4	0,72	1,12	-56,61
prev_sms:1	2,38	0,6	74,93
prev_sms:2	1,52	1,45	4,46
prev_sms:3	1,23	0,79	36,09
prev_sms:4	0,43	0,62	-43,52
prev_sms:5	0,94	1,53	-62,76
days_since_eligible:1	2,8	1,99	29,01
days_since_eligible:2	1,73	1,26	27,05
days_since_eligible:3	0,44	1,1	-149,08
days_since_eligible:4	0,17	0,34	-97,53
days_since_this_assign:1	2,49	1,48	40,6
days_since_this_assign:2	1,2	0,9	24,68
days_since_this_assign:3	0,32	0,85	-164,97
days_since_this_assign:4	0,46	0,89	-90,72

Tabulka 5.1: Srovnání odds_{ratio} na trénovacím a testovacím vzorku.

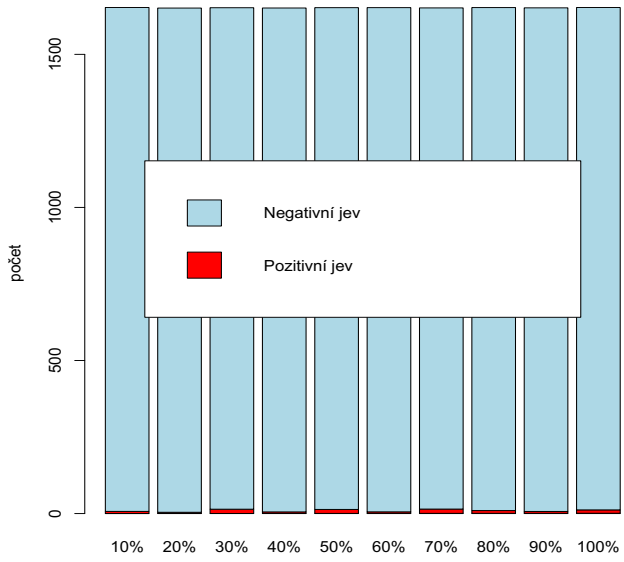
Můžeme také sestrojit histogram pozitivních a negativních výsledků (viz Obrázek 5.2). Tento histogram sestrojíme tak, že spočítané odhady pravděpodobnosti výskytu pozitivního jevu seřadíme od nejmenšího k největšímu a pro každé 10 % odhadů spočítáme počet pozitivních a negativních jevů. Pokud bychom měli učební vzorek s větším počtem pozitivních výsledků, tak bychom dostali histogram, kde počet pozitivních výsledků roste a počet negativních výsledků klesá.

Takže můžeme usoudit, že Nezávislý model velice špatně odhaduje pravděpodobnost výskytu pozitivního jevu.

Kód postupu výpočtů v programu R najdeme v Příloze C.1.



Obrázek 5.1: ROC křivka pro Nezávislý model.



Obrázek 5.2: Histogram pozitivních a negativních jevů v Nezávislém modelu.

5.3 Aplikace WOE modelu

V Nezávislém modelu jsme už počítali odds_{ratio} a ověřovali jejich stabilitu. Takže to už nemusíme dělat znovu. U WOE modelu také potřebujeme regresory, kategorie kterých mají stabilní odds_{ratio}. Na rozdíl od Nezávislého modelu tady budeme uvažovat regresory, kde aspoň polovina kategorií mají rozdíl v odds_{ratio} méně než 30 % (viz Tabulka 5.1). Aplikujeme tento nekonzervativní

postup, protože ve WOE modelu ještě budeme provádět dopřednou postupnou selekci (viz algoritmus 2.1), což také zmenší počet regresorů, které budou uvažovány ve finálním WOE modelu. Takže budeme uvažovat jenom 15 následujících regresorů: request_amount, max_emi, income, F_credit_history, lead_age, F_viet_name, F_region, F_gender, prev_tele, F_operator, products, F_renumbered, days_since_eligible, F_new_id_card, F_device.

Dále vezmeme trénovací data se spočítanými odds_{ratio} z Nezávislého modelu a transformujeme trénovací hodnoty na WOE hodnoty. Pomocí těchto odds_{ratio} přepočítáme také hodnoty v testovacím vzorku na WOE hodnoty.

Teď máme všechno připravené na výpočet parametrů $\boldsymbol{\beta}$ na trénovacím vzorku. Pomocí programu R dostaneme odhady parametrů $\boldsymbol{\beta}$ a příslušné p-hodnoty (viz Tabulka 5.2).

Poznámka 5.1. Tabulka 5.2 obsahuje 3 sloupce: první sloupec je sloupec názvů regresorů (kde Intercept je parametr β_0), druhý sloupec obsahuje odhady parametrů příslušných regresorů, třetí sloupec obsahuje p-hodnoty. Tyto p-hodnoty dostaneme tak, že budeme testovat nulovou hypotézu, že daný regresor je nevýznamný v modelu, tj. parametr regresoru je roven 0, oproti alternativě, že regresor je významný v modelu, tj. parametr regresoru není roven 0. V podkapitole „Významnost regresorů“ jsme zmínili, že se pro výpočet těchto p-hodnot používá Waldova statistika. Kódy signifikantnosti označují, mezi kterými hodnotami leží daná p-hodnota. Například, regresor income má p-hodnotu 0,049432**, kde dvě hvězdičky nám říkají, že tato p-hodnota leží v intervalu od 0,001 do 0,01.

Druhým krokem bude to, že aplikujeme dopřednou postupnou selekci (viz algoritmus 2.1). Po aplikace dopředné postupné selekce dostaneme 11 signifikantních regresorů (viz Tabulka 5.3).

Název regresoru	**Odhad parametru**	**p-hodnota**
(Intercept)	-5,28558	< 2e-16 ***
request_amount	1,45373	0,303804
max_emi	1,71469	0,006038 *
income	1,28246	0,049432 **
F_credit_history	1,31048	2,64e-05 ***
lead_age	-0,83519	3,88e-06 ***
F_device	0,98138	0,000325 ***
F_viet_name	0,70072	0,483480
F_region	0,55969	0,172258
F_gender	1,46892	0,000272 ***
F_operator	1,45177	3,63e-09 ***
F_renumbered	-0,07996	0,946855
products	0,83080	0,051563 .
F_new_id_card	0,66326	0,161734
prev_tele	0,75084	1,90e-05 ***
days_since_eligible	1,55663	< 2e-16 ***

Kódy signifikantnosti: 0 ’***’ 0,001 ’**’ 0,01 ’*’ 0,05 ’.’ 0,1 ’ ’ 1

Tabulka 5.2: Odhady parametrů ve WOE modelu.

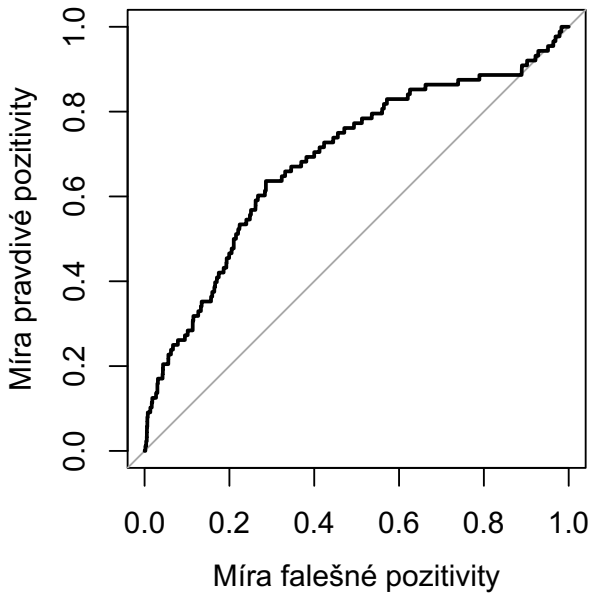
Název regresoru	**Odhad parametru**	**p-hodnota**
(Intercept)	-5,28442	< 2e-16 ***
days_since_eligible	1,55436	< 2e-16 ***
F_credit_history	1,32742	1,95e-05 ***
F_operator	1,44876	1,36e-09 ***
F_device	0,98441	0,000294 ***
lead_age	-0,82596	4,82e-06 ***
prev_tele	0,71985	3,13e-05 ***
F_gender	1,44325	0,000325 ***
income	1,46081	0,021628 *
max_emi	1,61080	0,008407 **
products	0,84044	0,049053 *
F_region	0,77966	0,043325 *

Kódy signifikantnosti: 0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

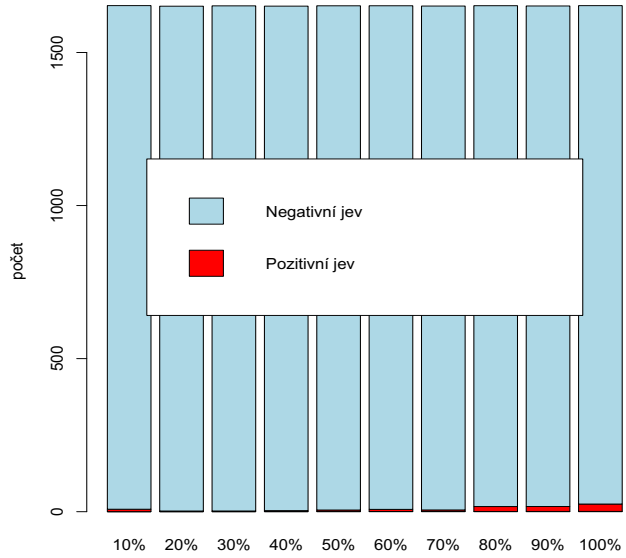
Tabulka 5.3: Odhady parametrů ve WOE modelu po dopředné postupné selekce.

Teď vyzkoušíme náš model na testovacích datech. Sestrojíme ROC křivku a dostaneme, že Giniho koeficient je roven 0,69 (viz Obrázek 5.3).

Poznámka 5.2. Na první pohled se může zdát, že máme docela přesný model. Bohužel, náš učební vzorek je krásným příkladem toho, jaké výsledky můžeme dostávat, když pracujeme s tzv. *nevyváženým* učebním vzorkem. Nevyvážený učební vzorek je datový soubor, ve kterém existuje proměnná, kategorie které má příliš vysoký podíl ve srovnání s ostatními kategoriemi dané proměnné. V našem případě je to odezva, která má jen 0,51 % pozitivních výsledků. Nevyvážený



Obrázek 5.3: ROC křivka pro WOE model.



Obrázek 5.4: Histogram pozitivních a negativních jevů ve WOE modelu.

učební vzorek může způsobit různé problémy. Například, model, který byl natrénován na takovém datovém souboru, bude predikovat negativní výsledky s vysokou přesností, ale přesnost predikce pozitivních výsledků bude velice nízká (viz Obrázek 5.4). Nicméně, takový model bude mít vysoký Giniho koeficient.

Kód postupu výpočtů v programu R najdeme v Příloze C.2.

5.4 Aplikace modelu Plné logistické regrese

Začneme tím, že rozdělíme učební vzorek na trénovací a testovací data. Pak odhadneme vektor parametrů $\boldsymbol{\beta}$ na trénovacím vzorku a pomocí programu R dostaneme odhady parametrů a také příslušné p-hodnoty (viz Tabulka 5.4). Když aplikujeme dopřednou postupnou selekci, tak do finální kontroly dojde 12 regresorů a po finální kontrole dostaneme 8 regresorů (viz Tabulka 5.5). V Tabulce 5.6 pro zbylé 8 regresorů jsou spočítány hodnoty LR statistik, které se používají v testu poměrem věrohodností, a příslušné p-hodnoty, které vyjadřují významnost regresorů v modelu.

Po vyzkoušení modelu na testovacích datech nakreslíme ROC křivku (viz Obrázek 5.5) a dostaneme Giniho koeficient rovný 0,68. Ze stejného důvodu jako u WOE modelu dostáváme vysoký Giniho koeficient. Nakreslíme také histogram pozitivních a negativních jevů, abychom mohli vidět, že i tento model špatně predikuje pravděpodobnost výskytu pozitivního jevu (viz Obrázek 5.6). Musíme si také uvědomit, že model Plné logistické regrese bude mít 23 parametry $\boldsymbol{\beta}$ (viz Tabulka 5.5), což může být vzhledem k počtu pozitivních jevů v učebním vzorku zavádějící.

Kód postupu výpočtů v programu R najdeme v Příloze C.3.

Název regresoru	**Odhad parametru**	**p-hodnota**
(Intercept)	-4,780	< 2e-16 ***
request_amount:2	0,3886	0,115290
request_amount:3	-0,2315	0,292461
request_amount:4	0,0653	0,776124
max_emi:2	-0,2570	0,262816
max_emi:3	-0,4471	0,037885 *
F_income_type:BNL	-0,3830	0,154130
F_income_type:LC	-0,6027	0,055779 .
F_income_type:LS	-0,5446	0,066306 .
F_income_type:OTH	-0,7071	0,019700 *
income:2	0,1629	0,257232
income:3	-0,0746	0,572288
F_credit_history:LDT	-0,1229	0,623062
F_credit_history:LPT	-0,2759	0,279329
F_credit_history:NB	-1,1960	0,000104 ***
lead_age:2	1,5760	3,99e-09 ***
lead_age:3	-0,2331	0,321683
F_device:iPhone	0,5006	0,001296 **
F_device:Other	0,0309	0,975784
F_device:Windows	0,2913	0,534957
F_viet_name:Symbols	0,2024	0,368077
F_region:East	0,4748	0,099939 .
F_region:North	0,1424	0,608484
F_region:South	0,0667	0,777516
F_region:West	0,2071	0,428398
F_gender:M	-0,4768	0,001920 **
F_operator:Small operator	0,8334	0,124753
F_operator:Vietnamobile	-14,440	0,963567
F_operator:Viettel	-0,9098	2.50e-06 ***
F_operator:VinaPhone	-0,2456	0,235389
F_renumbered:unchanged	-0,0309	0,846480
F_ad:GG	0,0547	0,782193
F_ad:Multi	-1,2400	0,223980
F_ad:NO VISIT	-1,1050	0,281718
F_ad:other	-4,393e-05	0,999891
products:2	-0,6687	0,0405*
products:3	-0,3685	0,1795
F_new_id_card:Old	-0,2008	0,2249
prev_tele:2	-0,9125	0.000264 ***
prev_tele:3	0,3505	0,0723 .
prev_tele:4	0,2674	0,1059
prev_sms:2	0,0799	0,8262
prev_sms:3	0,1384	0,6101
prev_sms:4	0,0282	0,8934
prev_sms:5	0,1592	0,3615

Kódy signifikantnosti: 0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Název regresoru	**Odhad parametru**	**p-hodnota**
days_since_eligible:2	-3,3370	< 2e-16 ***
days_since_eligible:3	-0,2999	0,2497
days_since_eligible:4	0,3837	0,0926 .
days_since_this_assign:2	-0,4769	0,0587 .
days_since_this_assign:3	0,8509	0,0006***
days_since_this_assign:4	0,0185	0,9447

Kódy signifikantnosti: 0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Tabulka 5.4: Odhady parametrů v modelu Plné logistické regrese.

Název regresoru	**Odhad parametru**	**p-hodnota**
(Intercept)	-5,160748	< 2e-16 ***
days_since_eligible:2	-3,332138	< 2e-16 ***
days_since_eligible:3	-0,296729	0,252231
days_since_eligible:4	0,387694	0,089024 .
F_credit_history:LDT	-0,039187	0,873656
F_credit_history:LPT	-0,156151	0,532530
F_credit_history:NB	-1,117794	0,000216 ***
F_operator:Small operator	0,853719	0,109004
F_operator:Vietnamobile	-14,444902	0,963977
F_operator:Viettel	-0,856673	9.89e-7 ***
F_operator:VinaPhone	-0,219018	0,265349
prev_tele:2	-0,908656	5.72e-5 ***
prev_tele:3	0,407633	0,030971 *
prev_tele:4	0,265343	0,104226
lead_age:2	1,614082	1.02e-10 ***
lead_age:3	-0,248315	0,283289
days_since_this_assign:2	-0,455477	0,064175 .
days_since_this_assign:3	0,913028	0,000123 ***
days_since_this_assign:4	-0,006466	0,980039
F_device:iPhone	0,600621	6.26e-5 ***
F_device:Other	0,069265	0,945530
F_device:Windows	0,374422	0,420444
F_gender:M	-0,446493	0,002827 **

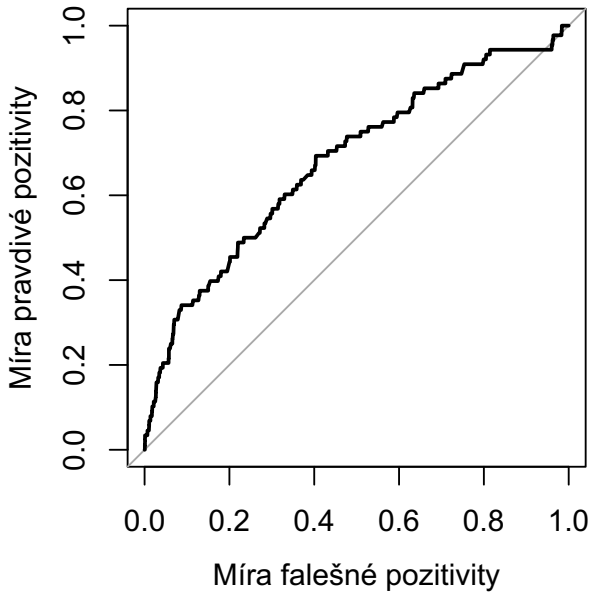
Kódy signifikantnosti: 0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Tabulka 5.5: Odhady parametrů v modelu Plné logistické regrese po dopředné postupné selekce.

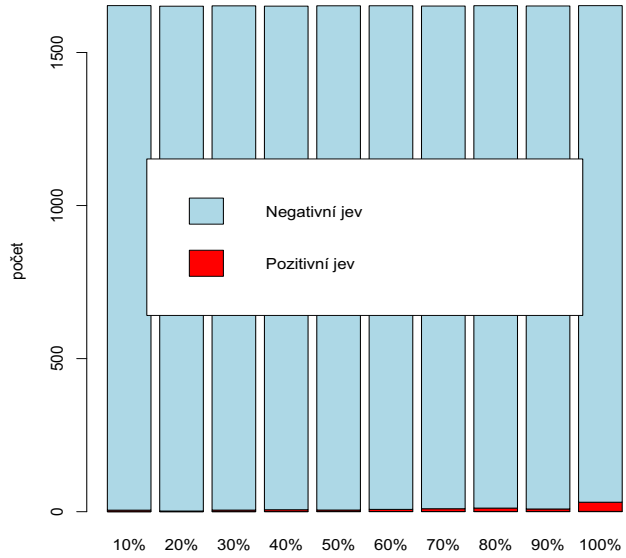
Název regresoru	LR **statistika**	**p-hodnota**
days_since_eligible	128,982	< 2,2e-16 ***
F_credit_history	24,737	1,752e-05 ***
F_operator	39,305	6,025e-08 ***
prev_tele	26,394	7,885e-06 ***
lead_age	36,939	9,521e-09 ***
days_since_this_assign	18,352	0,0003721 ***
F_device	15,818	0,0012359 **
F_gender	8,744	0,0031063 **

Kódy signifikantnosti: 0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Tabulka 5.6: LR statistiky v modelu Plné logistické regrese po dopředné postupné selekce.



Obrázek 5.5: ROC křivka pro model Plné logistické regrese.

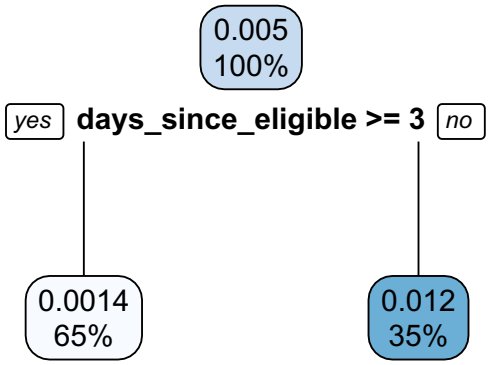


Obrázek 5.6: Histogram pozitivních a negativních jevů v modelu Plné logistické regrese.

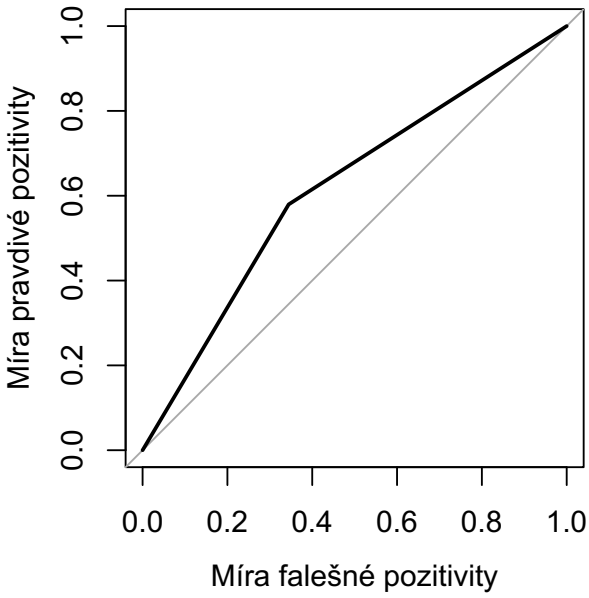
5.5 Aplikace CART (regresní strom)

Rozdělíme učební vzorek na trénovací a testovací data. Pak na základě trénovacích dat sestrojíme velice velký strom (viz algoritmus 3.1). Potom pomocí metody minimálního prořezávání cenové náročnosti (viz algoritmus 3.2) spočítáme ladící parametr $\widehat{\alpha}$ s odpovídající hodnotou chyby křížové validace (tj. $R\left(T_{\alpha_v}^{(L_1)}\right)$ z algoritmu 3.2). Vyjde nám, že $\widehat{\alpha} = 0{,}004\,845\,282$, což odpovídá stromu, který sestává jen z kořenu. Proto vybereme ladící parametr, který odpovídá druhé nejmenší hodnotě chyby křížové validace. Hodnota druhého ladícího parametru se rovná 0,002 624 270. Pomocí tohoto parametru dostaneme strom s optimální velikostí (viz Obrázek 5.7).

Poznámka 5.3. Z Obrázku 5.7 můžeme vidět větvení našeho stromu. Vzhledem k tomu, že máme nevyvážený datový soubor (viz poznámka 5.2), tak jsme dostali malý strom (pařez). Vidíme, že každý vnitřní uzel obsahuje 2 čísla. Horní číslo označuje odhad odezvy (tj. hodnotu listu), zatímco dolní číslo označuje, kolik procent pozorování z celého vzorku obsahuje daný vnitřní uzel. Pod vnitřním uzlem se nachází podmínka dělení: název regresoru, podle kterého se tento uzel dělil, s příslušnou hodnotou regresoru. Takže jsme dostali strom, který se dá interpretovat tak, že pokud hodnota regresoru days_since_eligible je větší nebo rovná 3, tak odhadneme odezvu jako hodnotu rovné 0,0014. Pokud hodnota regresoru bude menší než 3, tak odhadneme odezvu jako hodnotu rovnou 0,012.



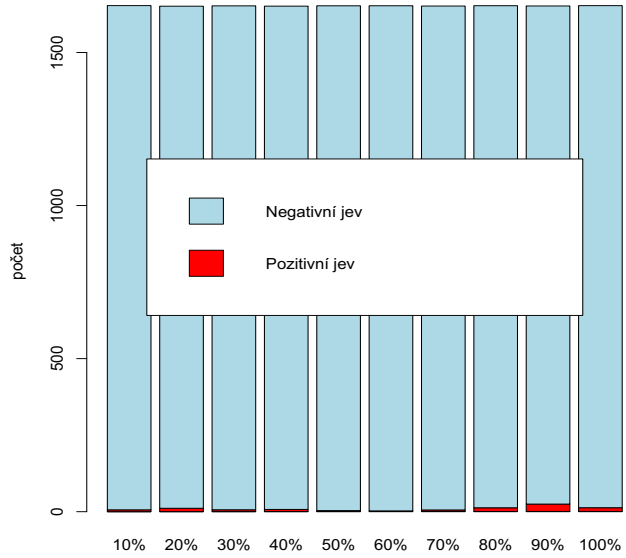
Obrázek 5.7: Strom sestrojený metodou CART (regresní strom).



Obrázek 5.8: ROC křivka pro CART (regresní strom).

Když otestujeme náš model na testovacích datech, tak dostaneme Giniho koeficient rovný 0,62 a příslušnou ROC křivku (viz Obrázek 5.8). Sestrojíme také Histogram pozitivních a negativních jevů (Obrázek 5.9).

Kód postupu výpočtů v programu R najdeme v Příloze C.4.



Obrázek 5.9: Histogram pozitivních a negativních jevů pro CART (regresní strom).

5.6 Aplikace CART (klasifikační strom)

Stejně jako u regresního stromu rozdělíme učební vzorek na trénovací a testovací data. Potom na základě trénovacích dat sestrojíme velice velký strom a pomocí metody minimálního prořezávání cenové náročnosti s adaptací pro klasifikační stromy (viz Breiman a kol., 1984, odst. 3.1–3.4) spočítáme ladící parametr $\widehat{\alpha}$ s odpovídající hodnotou chyby křížové validace. Jako výsledek dostaneme, že $\widehat{\alpha} = 0{,}001\,745\,200$, což zase odpovídá stromu, který se sestává jen z kořenu. Proto vybereme ladící parametr, který odpovídá druhé nejmenší hodnotě chyby křížové validace. Dostaneme parametr $\widehat{\alpha} = 0{,}000\,951\,930$, pomocí kterého sestrojíme strom optimální velikosti (viz Obrázek 5.10).

Poznámka 5.4. Z Obrázku 5.10 můžeme vidět, jak probíhá větvení klasifikačního stromu. Na rozdíl od regresního stromu, vnitřní uzly klasifikačního stromu obsahují 3 čísla. Horní číslo označuje odhad kategorie odezvy, číslo uprostřed označuje podíl úspěchů v daném uzlu a dolní číslo označuje, kolik procent pozorování z celého vzorku obsahuje daný vnitřní uzel.

Schematicky, proces větvení probíhá následujícím způsobem:

1) pokud days_since_eligible ≥ 3, tak $\widehat{y} = 0$. Jinak:

2) pokud F_credit_history$\neq$LDT, LPT, NB, tak $\widehat{y} = 0$. Jinak:

3) pokud F_operator$\neq$Vietnamobile,Viettel, tak $\widehat{y} = 0$. Jinak:

4) pokud F_device=Android nebo Other, tak $\widehat{y} = 0$. Jinak:

5) pokud F_region=Central nebo South, tak $\widehat{y} = 0$. Jinak:

6) pokud lead_age < 2, tak $\widehat{y} = 0$. Jinak:

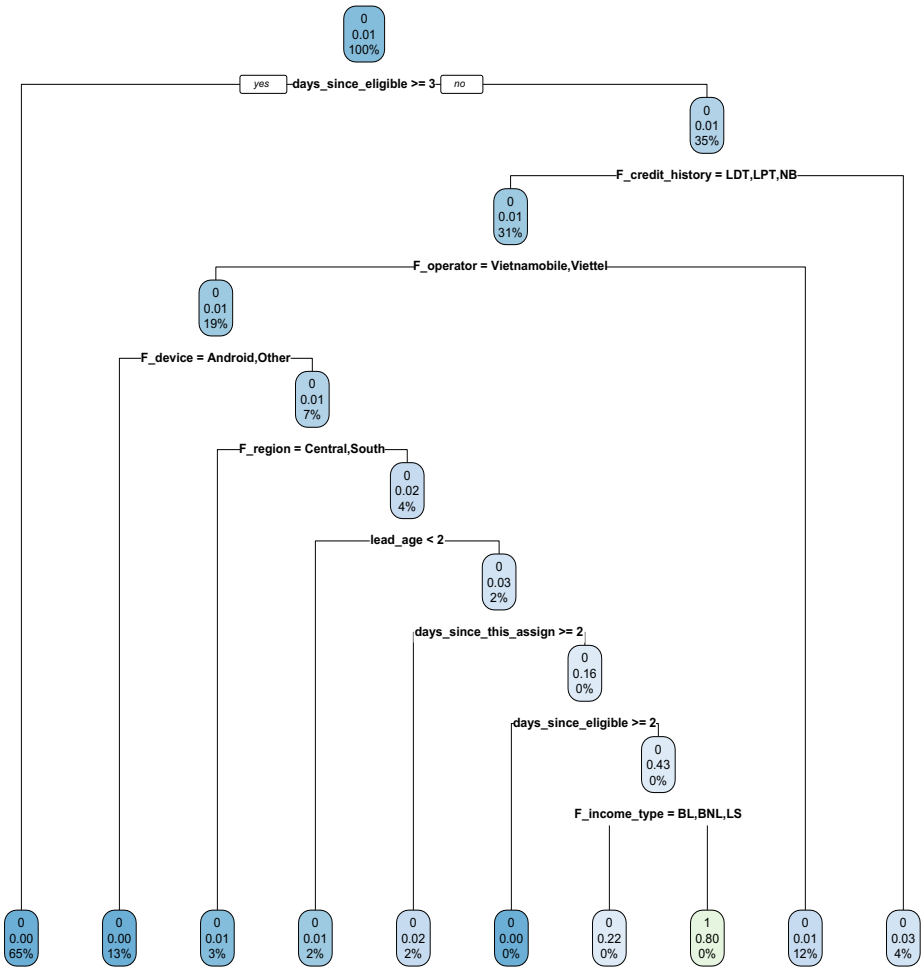
7) pokud days_since_this_assign$\geq$2, tak $\widehat{y} = 0$. Jinak:

8) pokud days_since_eligible$\geq$2, tak $\widehat{y} = 0$. Jinak:

9) pokud F_income_type$\neq$BL,BNL,LS, tak $\widehat{y} = 1$. Jinak $\widehat{y} = 0$.

Vzhledem k tomu, že tento model je klasifikátor, tak ho budeme testovat pomocí chybové matice a chybové míry. Na základě testovacích dat dostaneme následující chybovou matici $\mathbb{M}$ jako

$$\mathbb{M} = \begin{pmatrix} & 0 & 1 \\ \hat{0} & 16\,428 & 88 \\ \hat{1} & 1 & 0 \end{pmatrix},$$



Obrázek 5.10: Strom sestrojený metodou CART (klasifikační strom).

odkud dostaneme, že chybová míra MR je rovna 0,01. Přestože máme nízkou chybovou míru, můžeme si všimnout, že jsme neodhadli správně žádný skutečný pozitivní jev. To zase souvisí s tím, že máme nevyvážený učební vzorek.

Kód postupu výpočtů v programu R najdeme v Příloze C.5.

5.7 Aplikace CHAID

Začneme tím, že rozdělíme učební vzorek na trénovací a testovací data. Pomocí křížové validace (viz Hastie a kol., 2009, odst. 7.10) na základě trénovacích hodnot určíme optimální velikost stromu a na základě testovacích hodnot sestrojíme strom (viz Obrázek 5.11).

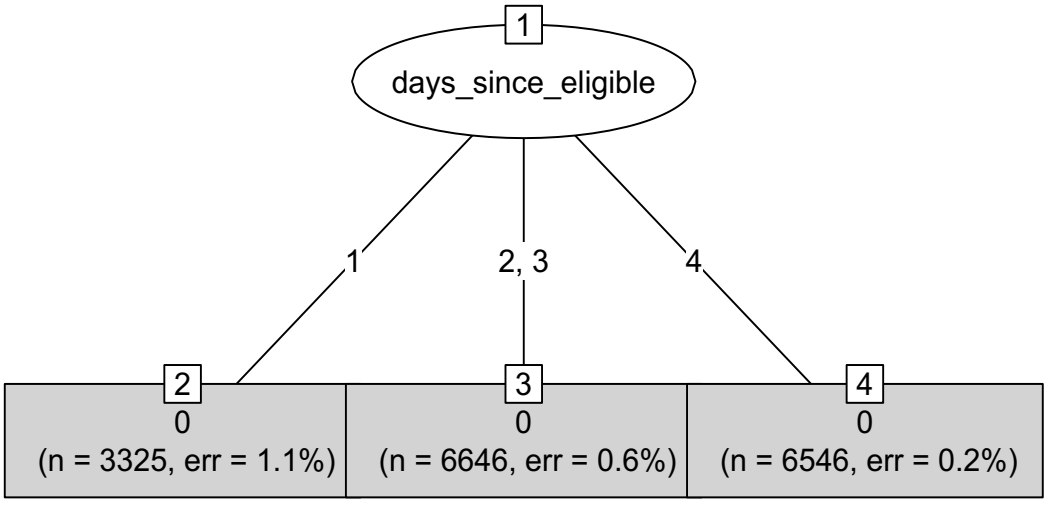
Na základě testovacích dat spočítáme chybovou matici $\mathbb{M}$

$$\mathbb{M} = \begin{pmatrix} & 0 & 1 \\ \hat{0} & 16\,429 & 88 \\ \hat{1} & 0 & 0 \end{pmatrix},$$

ze které spočítáme chybovou míru $MR = 0{,}01$. I tady můžeme vidět, že jsme neodhadli správně žádný pozitivní výsledek.

Kód postupu výpočtů v programu R najdeme v Příloze C.6.

Poznámka 5.5. Z Obrázku 5.11 můžeme vidět, jak probíhá větvení stromu pomocí CHAID. Listy stromu vypadají jinak než obyčejné vnitřní uzly. Listy jsou ve čtvercích a obsahují 4 čísla: horní číslo označuje pořádkové číslo uzlu, číslo uprostřed označuje odhad listu a poslední 2 čísla nám říkají, kolik pozorování obsahuje daný list (označeno jako n) a jaká je jeho chybová míra (označeno jako err).



Obrázek 5.11: Strom sestrojený metodou CHAID.

Vnitřní uzly obsahují jen pořádkové číslo uzlu a název regresoru, podle kterého se daný uzel rozděluje. Čísla na šipkách označují kategorie regresoru. Vzhledem k tomu, že máme nevyvážený učební vzorek, tak jsme dostali malý strom, který vždycky odhaduje odezvu jako hodnotu 0 podle regresoru days_since_eligible.

5.8 Aplikace Boosting

V tomto modelu také začneme tím, že rozdělíme učební vzorek na trénovací a testovací data. Kvůli časové složitosti algoritmu křížové validace (viz Hastie a kol., 2009, odst. 7.10) v programu R se nám nepodařilo určit počet iterací. Proto jako počet iterací vezmeme hodnotu 100. Pak aplikujeme algoritmus 3.4 a sestrojíme silný klasifikátor na základě trénovacích dat. Potom na základě testovacích dat spočítáme chybovou matici $\mathbb{M}$

$$\mathbb{M} = \begin{pmatrix} & 0 & 1 \\ \hat{0} & 16\,429 & 88 \\ \hat{1} & 0 & 0 \end{pmatrix}.$$

Z chybové matice spočítáme chybovou míru $MR = 0{,}01$. Je z možných řešení by mohlo být použití metody Boosting spolu s regresními stromy místo klasifikačních. Kód postupu výpočtů v programu R najdeme v Příloze C.7.

5.9 Porovnání výsledků predikčních modelů

V této podkapitole shrneme výsledky naších modelů. Jak jsme už zmínili v Kapitole 4, klasifikátory a prediktory budeme porovnávat zvlášť.

Z Tabulky 5.7 mohli bychom usoudit, že WOE model nejlépe ze všech modelů odděloval pozitivní výsledky od negativních. Na druhou stranu u WOE modelu nebyl splněn předpoklad stability odds_{ratio}. Takže model Plné logistické regrese bude mít výhodu proti WOE modelu. Nezávislý model byl nejhorším ze všech modelů. Z Tabulky 5.8 je těžké usoudit, který model byl přesnější, jelikož všechny tři mají stejně nízkou chybovou míru a všechny tři neodhadly správně ani jeden pozitivní výsledek. Z toho můžeme usoudit, že klasifikátory jsou pro daná data nevhodné.

Přestože jsme nedostali srozumitelné výsledky, to bylo dobrým ukázkovým příkladem toho, jaké výsledky můžeme dostávat při práci s nevyváženým datovým souborem. Analýza takových dat vyžaduje použití komplexnějšího matematického aparátu.

Název prediktoru	**Giniho koeficient**
IND	0,54
WOE	0,69
FLR	0,68
CART-R	0,62

Tabulka 5.7: Srovnání prediktorů

Název klasifikátoru	**Chybová míra**
CART-C	0,01
CHAID	0,01
BOOST	0,01

Tabulka 5.8: Srovnání klasifikátorů

Závěr

Cílem této práce bylo vysvětlení matematického aparátu funkčních modelů pro predikci závisle proměnné s alternativním rozdělením. Seznámili jsme se třemi stupni logistické regrese, kde jsme vysvětlili, jak se dají počítat parametry modelu. Popsali jsme také metody určování významnosti regresorů. Dále jsme se seznámily s dvěma druhy rozhodovacích stromů, kde jsme popsali různé algoritmy na sestrojení stromu a hledání jeho optimální velikosti. Ukázali jsme také, jak se dají aplikovat stromy v metodě Boosting. Na závěr teoretické části této práce jsme vysvětlili, jak porovnávat mezi sebou predikční modely.

V praktické části této práce jsme se pokusili aplikovat popsané modely na reálných datech od jedné vietnamské společnosti. Vysvětlili jsme problém nevyvážených datových souborů, na který jsme narazili při zpracování dat a který nám způsobil problémy ve výsledcích. Při aplikaci modelů jsme se snažili pomocí programu R ilustrovat výsledky.

Podařilo se nám popsat odhad závislé proměnné s alternativním rozdělením z různých pohledů, vysvětlit problematiku jednotlivých modelů a ukázat principy aplikace těchto modelů na reálných datech.

Seznam použité literatury

Anděl, J. (2007). *Základy matematické statistiky*. Matfyzpress, Praha. ISBN 80-7378-001-1.

Breiman, L., Friedman, J. H., Olshen, R. A. a Stone, C. J. (1984). *Classification and regression trees*. Wadsworth Statistics/Probability Series. Chapman & Hall/CRC, first edition. ISBN 978-0-412-04841-8.

Furnival, G. M. a Wilson, R. W. (1974). Regressions by Leaps and Bounds. *Technometrics*, **16**(4), 499–511.

Hastie, T., Tibshirani, R. a Friedman, J. (2009). *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition. ISBN 978-0-387-84857-0.

Hosmer, D. W., Lemeshow, S. a Sturdivant, R. X. (2013). *Applied Logistic Regression*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, New Jersey, third edition. ISBN 978-0-470-58247-3.

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, **29**(2), 119–127.

Zvára, K. (2008). *Regrese*. Matfyzpress, Praha. ISBN 978-80-7378-041-8.

Seznam obrázků

Seznam tabulek

A. Popis regresorů

Elektronická příloha obsahuje kompletní datový soubor, který byl použít pro účely aplikace modelů.

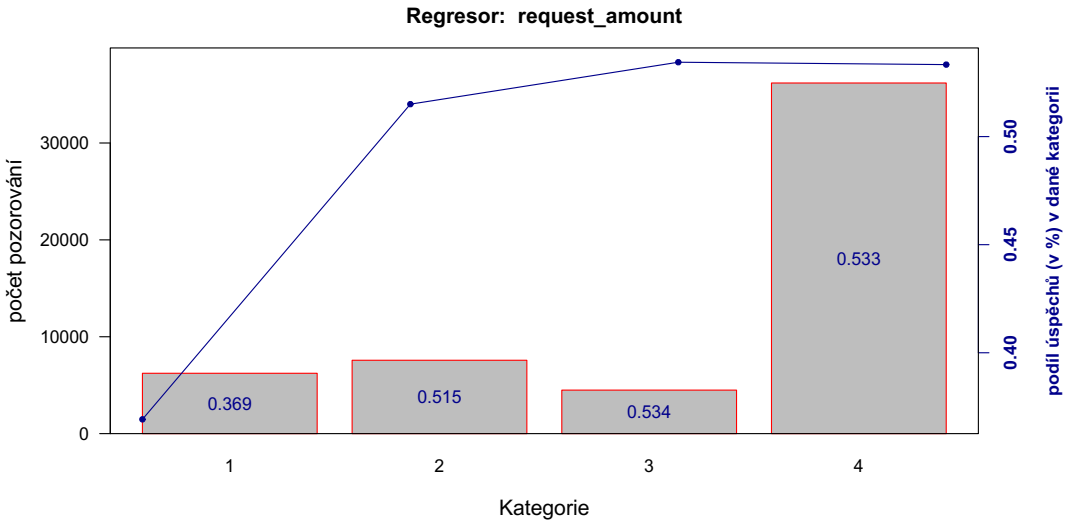
Název regresoru	**Význam**	**Kategorie**	**Kód kategorie**
request_amount	výše půjčky (v tisících VND)	1 000–10 000	1
		10 000–20 000	2
		20 000–30 000	3
		30 000–500 000	4
max_emi	maximální částka, kterou klient je schopen měsíčně splácet (v tisících VND)	170–1 000	1
		1 000–1 962	2
		1 962–50 000	3
F_income_type	druh příjmů	pracovní smlouva s platem v hotovosti;	LC
		pracovní smlouva s platem na účet;	LS
		podnikatelství s licencí;	BL
		podnikatelství bez licence;	BNL
		ostatní	OTH
income	výše příjmů (v tisících VND)	0–9 000	1
		9 000–12 000	2
		12 000–1 000 000	3
F_credit_history	kreditní historie klienta	byl v defaultu;	HB
		měl úvěr během posledních 3 měsíců;	LDT
		měl úvěr před víc než 3 měsíce;	LPT
		nikdy půjčoval	NB
lead_age	rozdíl mezi datem, kdy klient vyplnil formulář a datem poslání SMS-zprávy	0–4	1
		5–12	2
		13–318	3

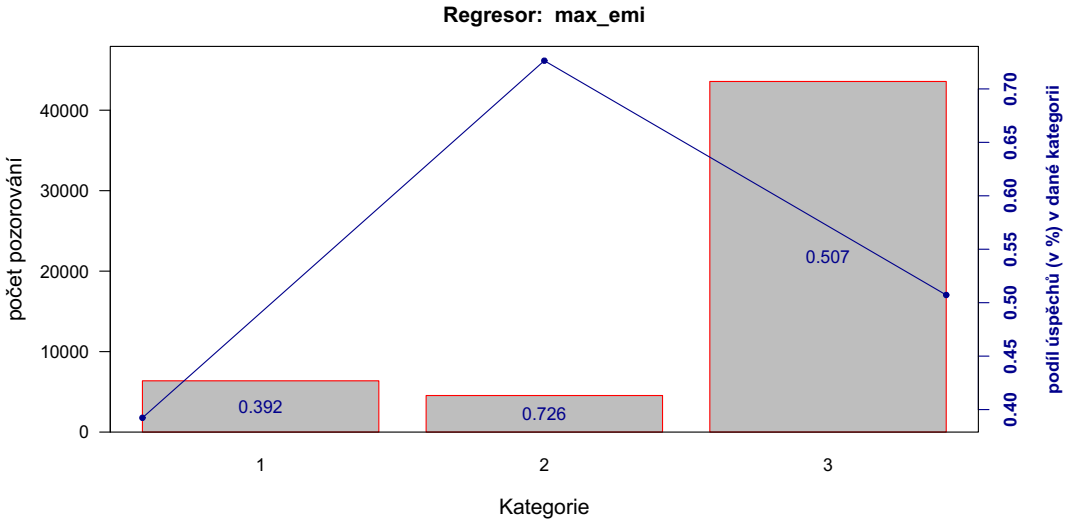
Název regresoru	Význam	Kategorie	Kód kategorie
F_device	mobilní operační systém klienta	iOS android windows ostatní	iPhone Android Windows Others
F_viet_name	indikátor použití vietnamských znaků při vyplňování jména	Ano, použil Ne, nepoužil	Symbols NoSymbols
F_region	v jaké části země bydlí klient	východ západ centr sever jih	East West Central North South
F_gender	pohlaví klienta	muž žena	M F
F_operator	mobilní operator klienta	Viettel VinaPhone Small operator MobiFone Vietnamobile	Viettel VinaPhone Small operator MobiFone Vietnamobile
F_renumbered	indikátor toho, jestli klient má nové nebo staré telefonní číslo	staré nové	unchanged reassigned
F_ad	kategorie reklamy, ze které klient přišel na web	facebook; google; vietnamský prohlížeč; ostatní; klient se dostal na WEB bez reklamy	FB GG Multi other NO VISIT
products	počet toho, kolik různých nabídek zaregistroval	1–2 3 4-16	1 2 3
F_new_id_card	indikátor toho, jestli klient má nový nebo starý typ ID karty	starý nový	Old New

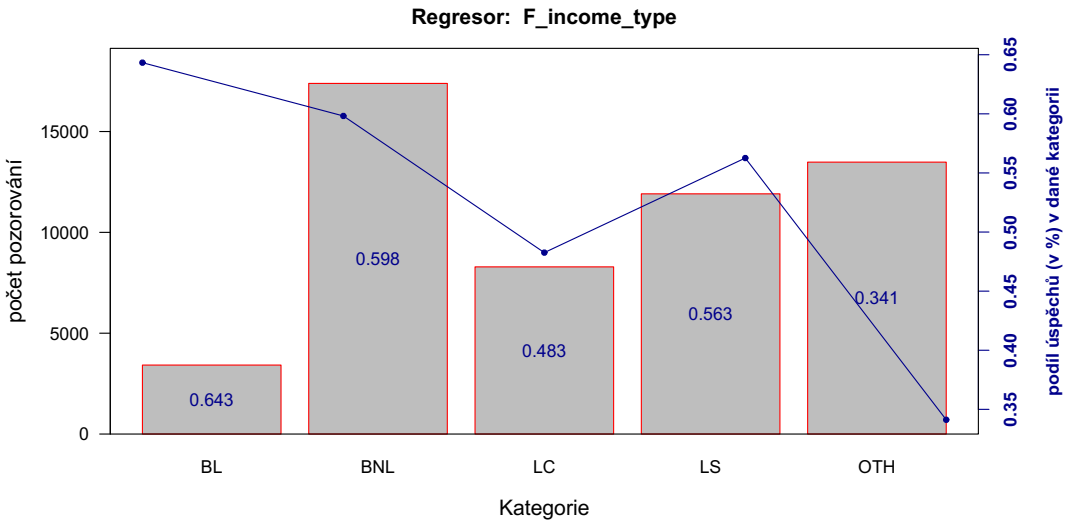
Název regresoru	**Význam**	**Kategorie**	**Kód kategorie**
prev_tele	počet toho, kolikrát volali klientu	0–1	1
		2	2
		3	3
		4–16	4
prev_sms	počet toho, kolikrát posílali SMS zprávu klientu	0	1
		1–2	2
		3–4	3
		5–12	4
		13–41	5
days_since_eligible	počet dnů mezi posláním nové SMS zprávy a předchozím podáním žádosti o úvěr	1–4	1
		5	2
		6–33	3
		34-892	4
days_since_this_assign	počet dnů mezi posláním nové SMS a tím, kdy klientu byl naposledy nabídnut úvěr	1–4	1
		5	2
		6–31	3
		32–688	4

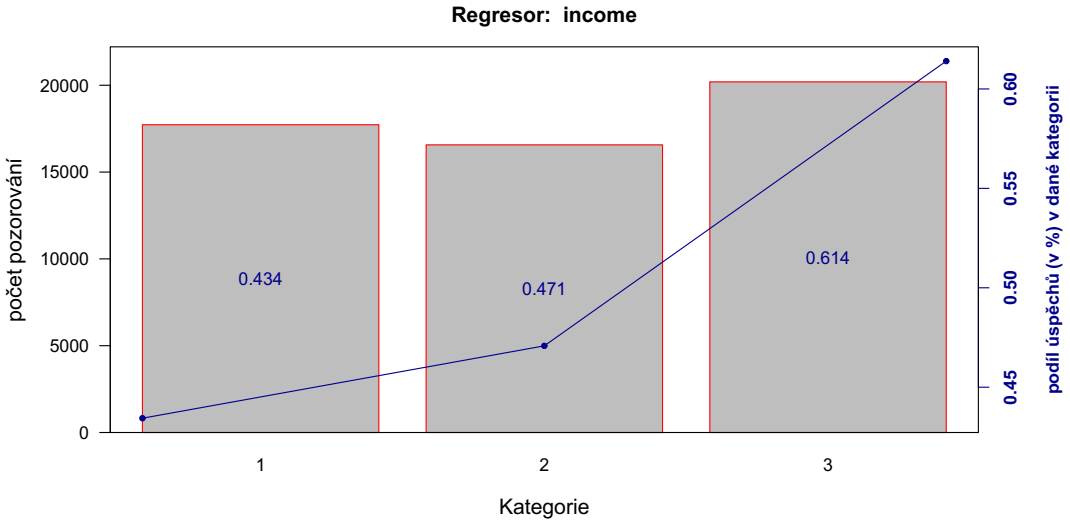
Tabulka A.1: Popis regresorů.

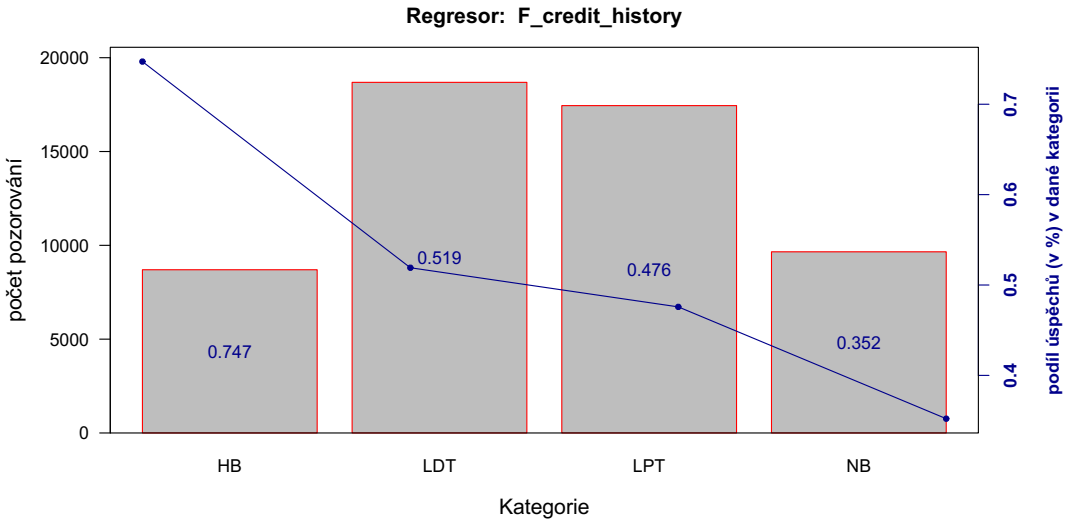
B. Závislost odezvy na regresorech



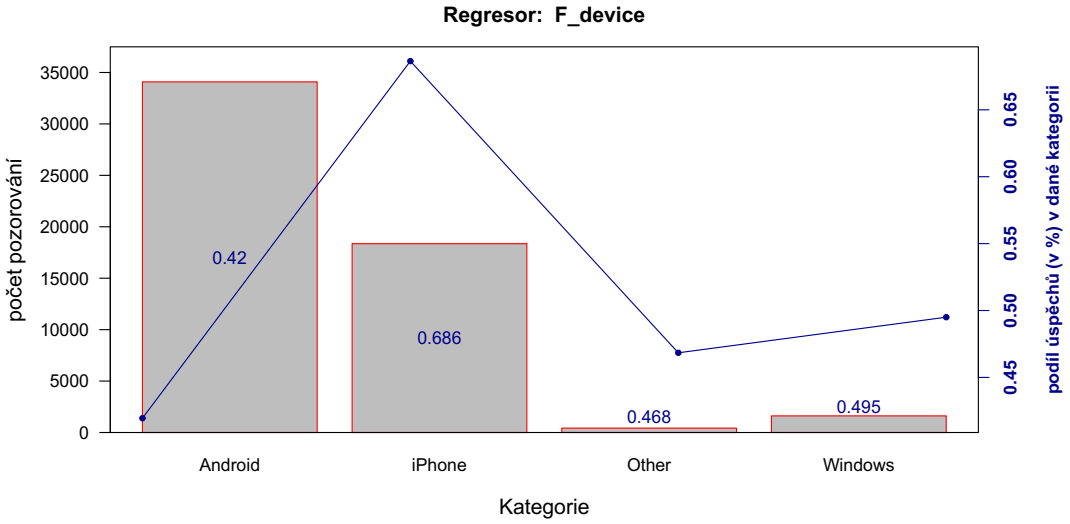


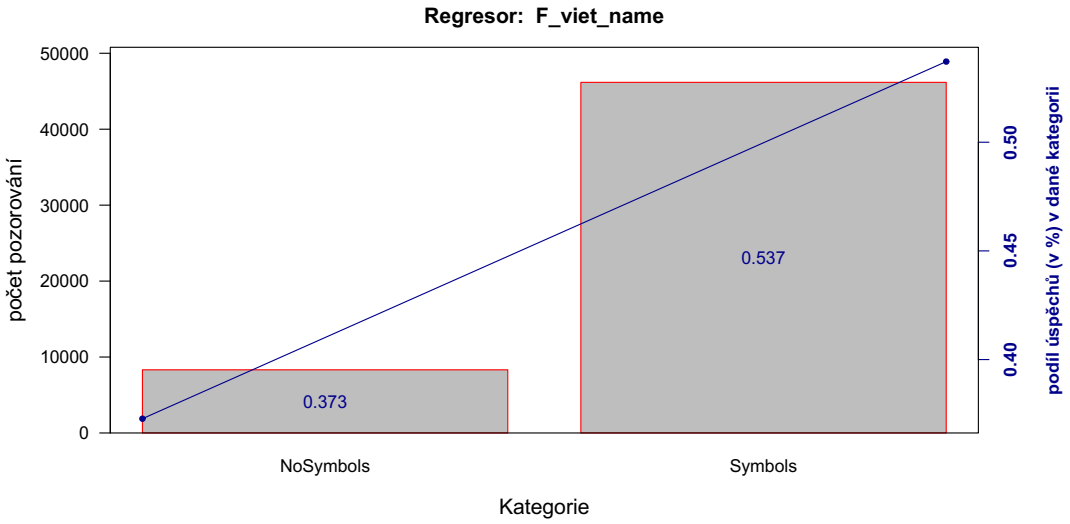

Regresor: F_income_type
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.643
0.598
0.483
0.563
0.341
BL
BNL
LC
LS
OTH
Kategorie

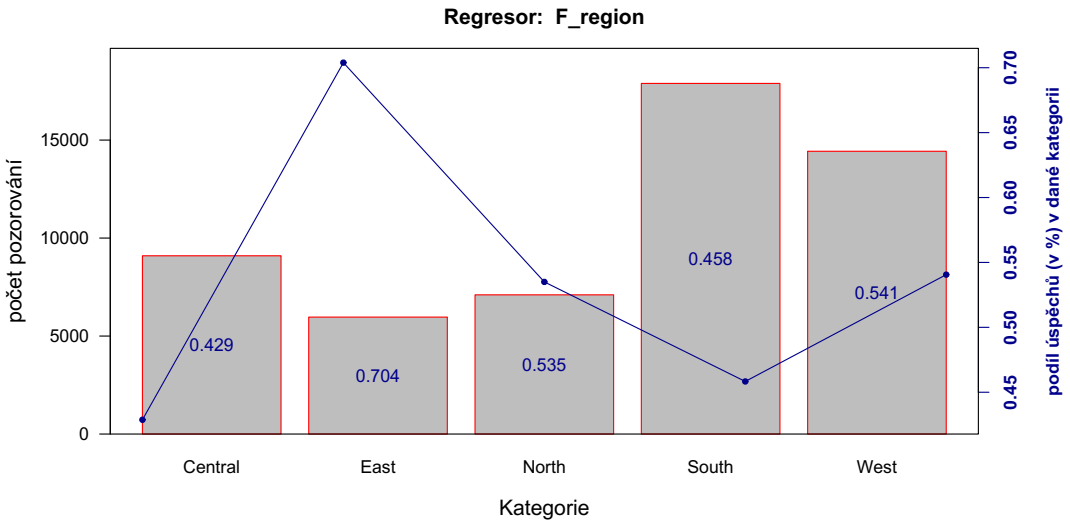

Regresor: income
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.434
0.471
0.614
1
2
3
Kategorie

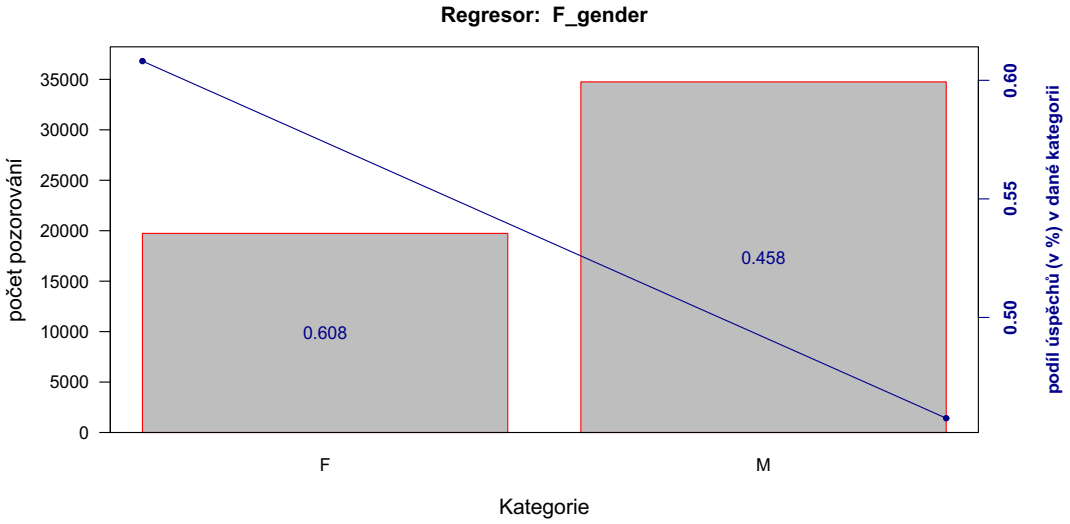

Regresor: F_credit_history
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.747
0.519
0.476
0.352
HB
LDT
LPT
NB
Kategorie

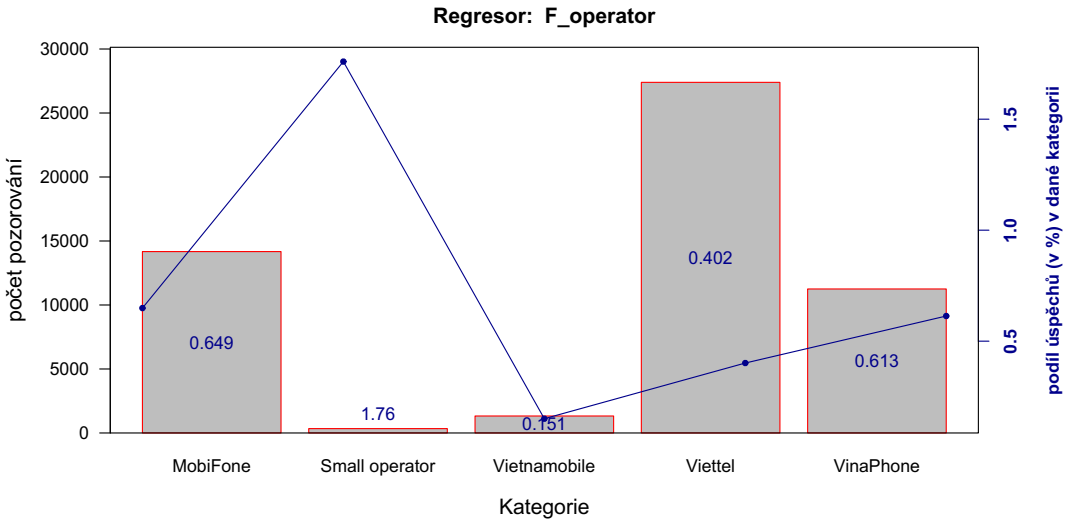

Regresor: lead_age
počet pozorování
podíl úspěchů (v %) v dané kategorii
1.052
0.622
0.228
1
2
3
Kategorie

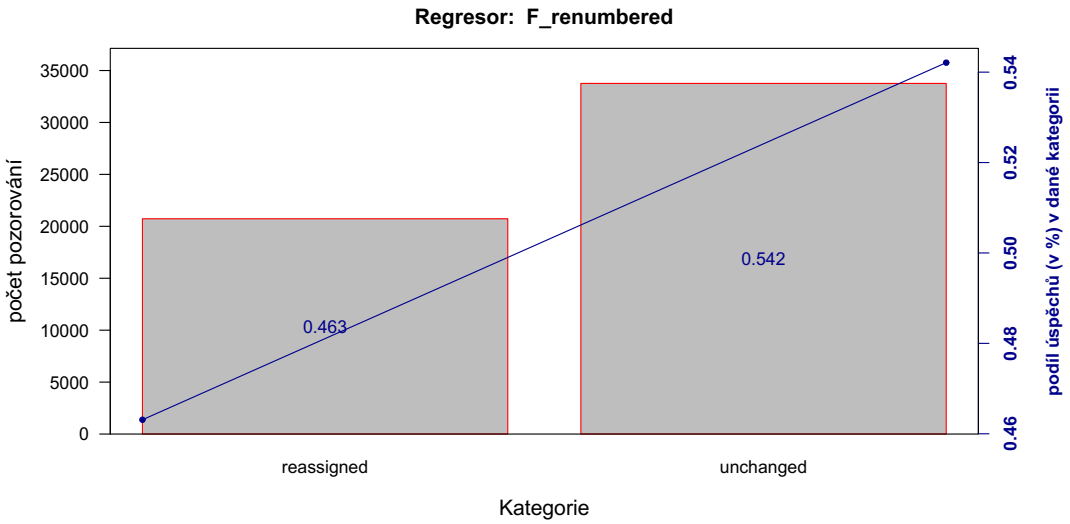

Regresor: F_device
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.42
0.686
0.468
0.495
Android
iPhone
Other
Windows
Kategorie

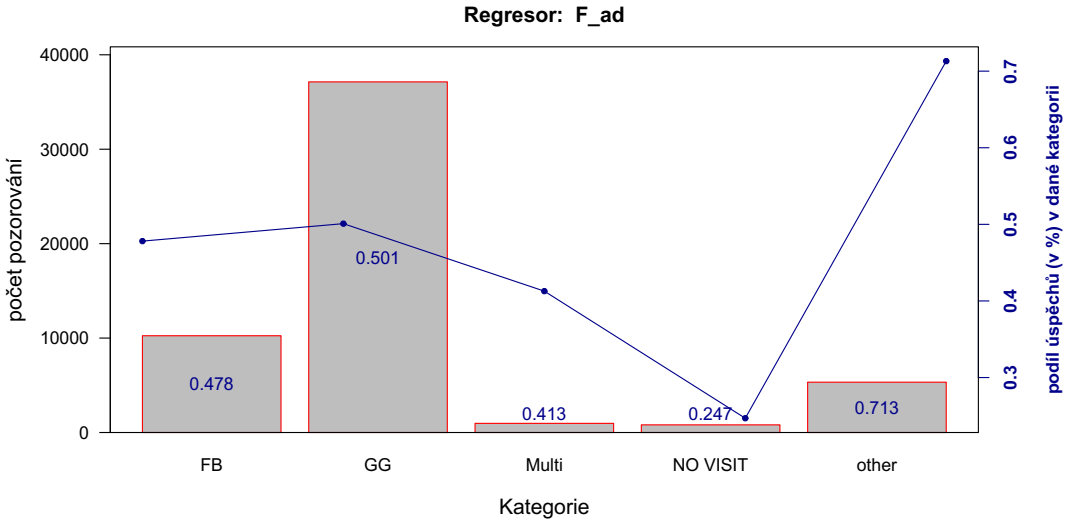

Regresor: F_viet_name
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.373
0.537
NoSymbols
Symbols
Kategorie

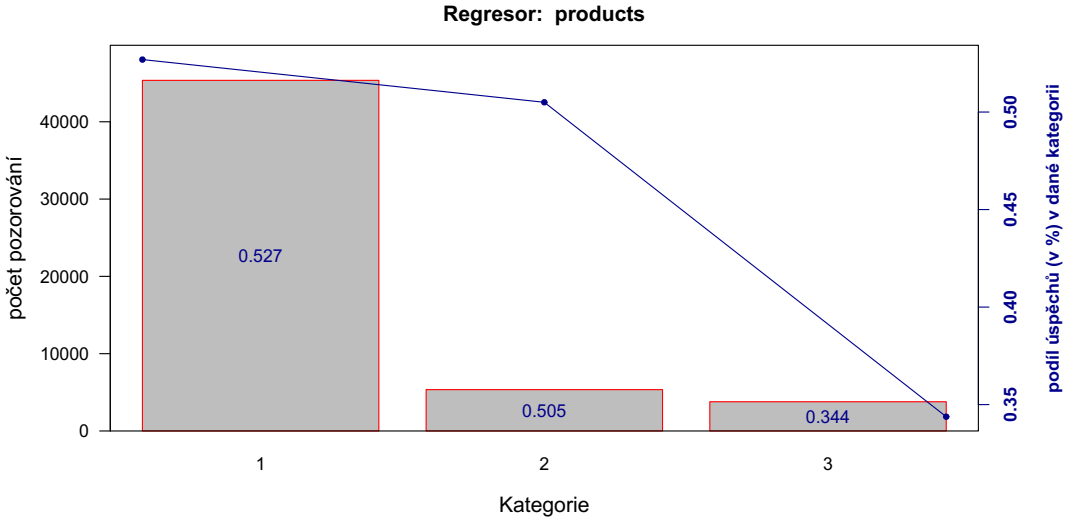

Regresor: F_region
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.429
0.704
0.535
0.458
0.541
Central
East
North
South
West
Kategorie

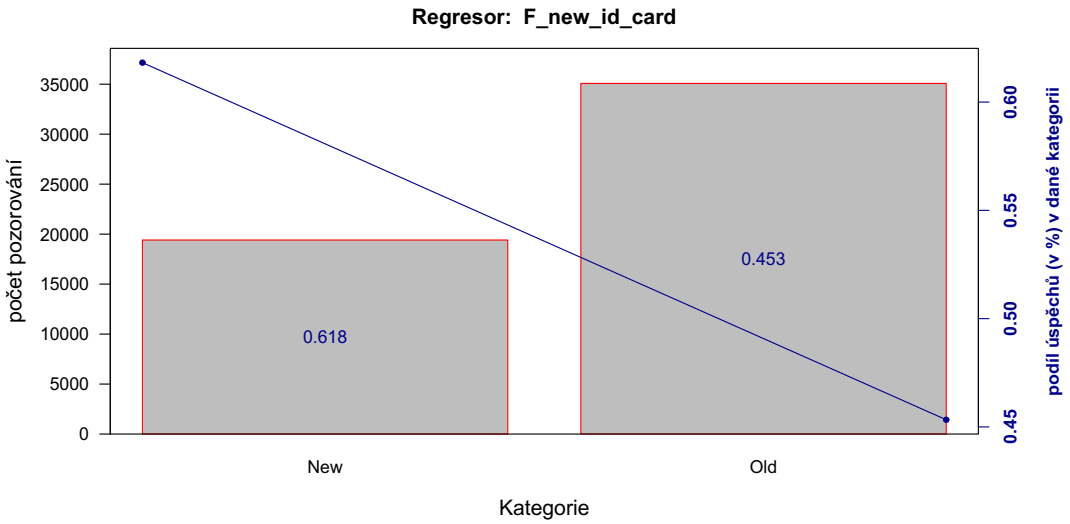

Regresor: F_gender
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.608
0.458
F
M
Kategorie


Regresor: F_operator
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.649
1.76
0.151
0.402
0.613
MobiFone
Small operator
Vietnamobile
Viettel
VinaPhone
Kategorie

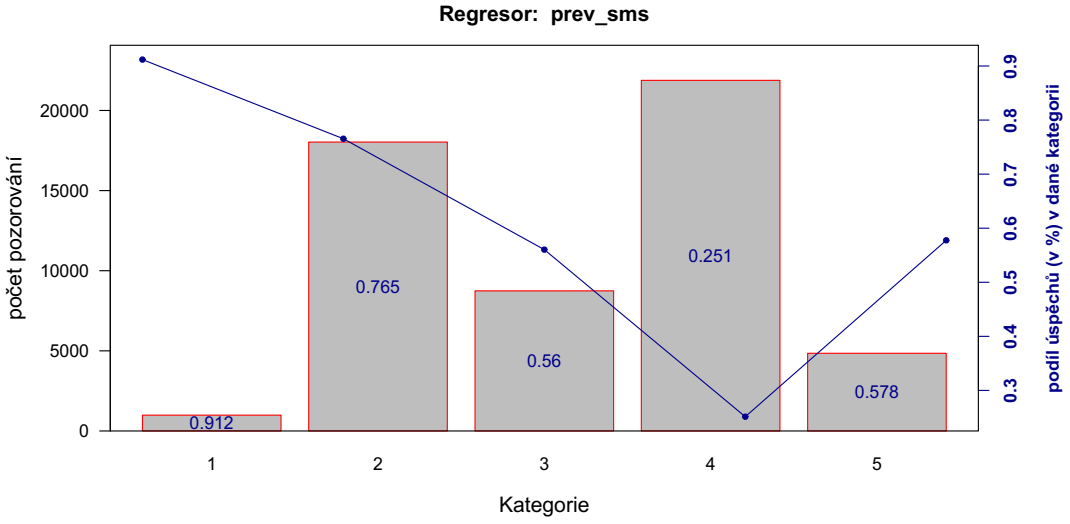

Regresor: F_renumbered
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.463
0.542
reassigned
unchanged
Kategorie

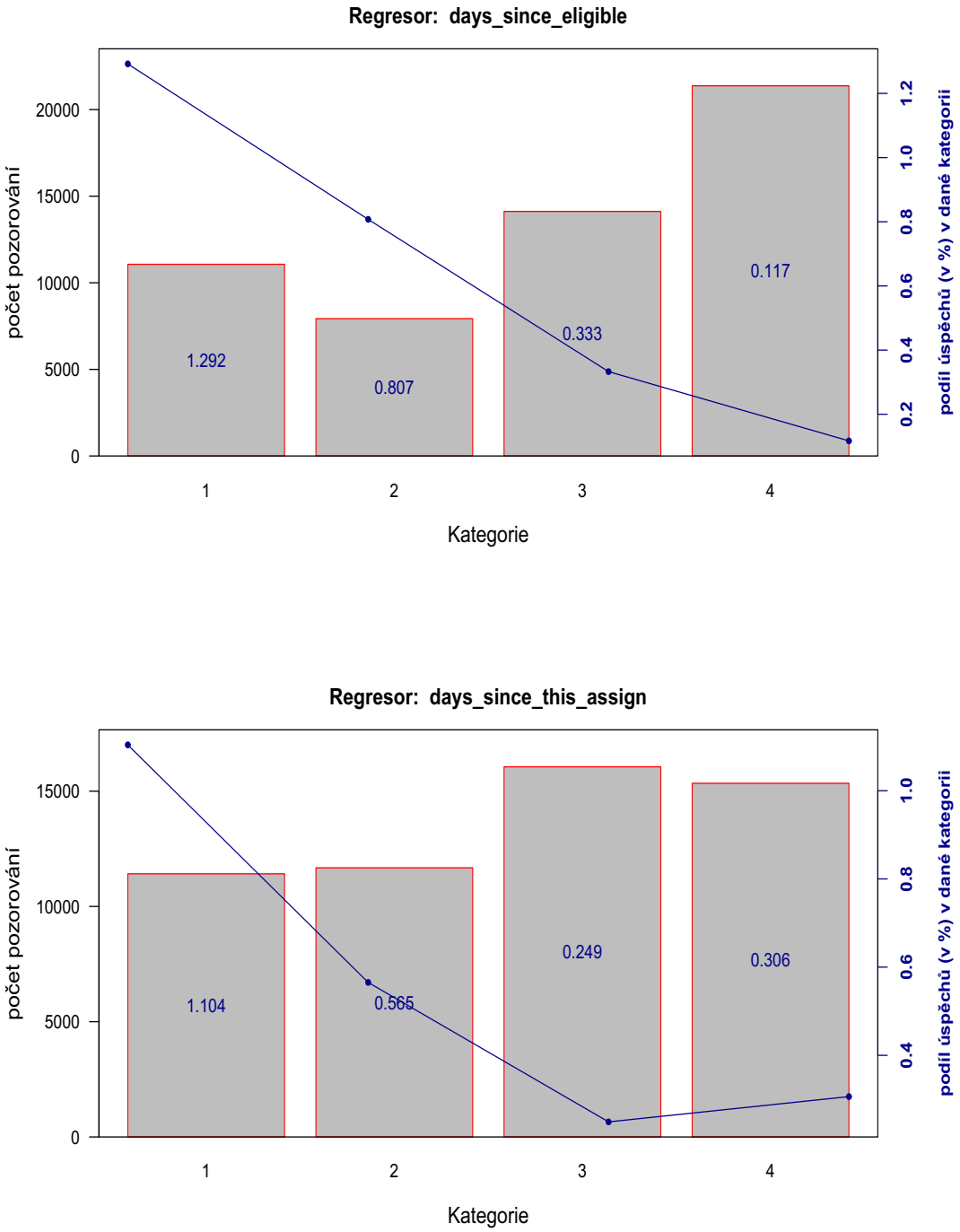

Regresor: F_ad
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.478
0.501
0.413
0.247
0.713
FB
GG
Multi
NO VISIT
other
Kategorie


Regresor: products
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.527
0.505
0.344
1
2
3
Kategorie


Regresor: F_new_id_card
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.618
0.453
New
Old
Kategorie


Regresor: prev_tele
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.961
0.543
0.285
0.433
Kategorie


Regresor: prev_sms
počet pozorování
podíl úspěchů (v %) v dané kategorii
0.912
0.765
0.56
0.251
0.578
Kategorie



Obrázek B.1: Grafy závislosti odezvy na regresorech.

C. Zdrojové kódy v programu R

C.1 Kód pro Nezávislý model

```
# Nahrajeme data a příslušné balíčky
>library(ROCR)
>library(pROC)
>library(readxl)
>datar <- read_excel("C:/Users/Desktop/Data.xlsx")
>datar=data.frame(datar)
>columnames=names(datar)
>y=datar$odezva

# Rozdělíme data na trénovací (70%) a testovací (30%)
>set.seed(42)
>sample_ind <- sample(c(TRUE, FALSE), nrow(datar), replace=TRUE,
prob=c(0.7,0.3))
>train_ind  <- datar[sample_ind, ]
>test_ind   <- datar[!sample_ind, ]

# Výpočet odds_ratio na trénovacím vzorku
>odds_ratio_train=c()
>odds_total_train=length(train_ind$odezva[train_ind$odezva==1])/
(length(train_ind$odezva[train_ind$odezva==0]))

>for (j in 1:19){
  kateg=length(levels(train_ind[,j]))
  for (k in 1:kateg){
    odds_ratio_train=append(odds_ratio_train,(1/odds_total_train)*
    (length(train_ind$odezva[train_ind$odezva==1&
    train_ind[,j]==levels(train_ind[,j])[k]]))/
    length(train_ind$odezva[train_ind$odezva==0&
    train_ind[,j]==levels(train_ind[,j])[k]]))
  }
}

# Výpočet odds_ratio na testovacím vzorku
>odds_ratio_test=c()
>odds_total_test=length(test_ind$odezva[test_ind$odezva==1])/
(length(test_ind$odezva[test_ind$odezva==0]))

>for (j in 1:19){
  kateg=length(levels(test_ind[,j]))
  for (k in 1:kateg){
    odds_ratio_test=append(odds_ratio_test,(1/odds_total_test)*
    (length(test_ind$odezva[test_ind$odezva==1&
    test_ind[,j]==levels(test_ind[,j])[k]]))/
    length(test_ind$odezva[test_ind$odezva==0&
    test_ind[,j]==levels(test_ind[,j])[k]]))
  }}
```

```
# Ověření stability odds_ratio
>nazvy=c()
>for (j in 1:19){
  kateg=levels(datar[,j])
  for (k in 1:length(kateg)){
    nazvy=append(nazvy,paste(columnames[j],kateg[k],sep=":"))
  }
}
>stab=data.frame(cbind(nazvy,round(odds_ratio_train,2),
round(odds_ratio_test,2),
round(100*(odds_ratio_train-odds_ratio_test)/odds_ratio_train,2)))
# Takže budeme uvažovat jenom 10-tý, 12-tý a 15-tý regresory

# Výpočet WOE pro IND model pro vybrané regresory
>ind=train_ind[,c(10,12,15,20)]
>ind2=test_ind[,c(10,12,15,20)]
>for (j in 1:3){
  odds_ratio_ind=c()
  kateg=levels(ind[,j])
  for (k in 1:length(kateg)){
    odds_ratio_ind=append(odds_ratio_ind,(1/odds_total_train)*
    (length(ind$odezva[ind$odezva==1&ind[,j]==levels(ind[,j])[k]]))/
    length(ind$odezva[ind$odezva==0&ind[,j]==levels(ind[,j])[k]]))
  }
  for (k in 1:length(kateg)){
    ind2[,j]=as.character(ind2[,j])
    ind2[,j][ind2[,j]==kateg[k]]=log(odds_ratio_ind[k])
  }
}

# Výpočet pravděpodobnosti výskytu pozitivního jevu
>p=c()
>for (i in 1:length(ind2$odezva)){
p=append(p,exp(log(odds_total_train)+as.numeric(ind2[i,1])+
as.numeric(ind2[i,2])+as.numeric(ind2[i,3]))/
(1+exp(log(odds_total_train)+as.numeric(ind2[i,1])+
as.numeric(ind2[i,2])+as.numeric(ind2[i,3]))))
}

# Sestrojení ROC křivky a výpočet Giniho koeficientu
>par(pty = "s")
>roc(ind2$odezva, p, plot=TRUE, legacy.axes=TRUE,
xlab="Míra falešné pozitivity",
    ylab="Míra pravdivé pozitivity")


# Sestrojíme histogram negativních a pozitivních výsledků
>poz_ind=data.frame(p,ind2$odezva)
>poz_ind=poz_ind[order(p),]
>bad_ind=c()
```

```
>good_ind=c()
>meze=as.numeric(round(quantile(c(1:length(p)),
probs=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)),0))
>for (j in 1:10){
  if (j==1){
    good_ind=append(good_ind,length(poz_ind[meze[j]:meze[j+1],2]
    [poz_ind[meze[j]:meze[j+1],2]==1]))
    bad_ind=append(bad_ind,length(poz_ind[meze[j]:meze[j+1],2]
    [poz_ind[meze[j]:meze[j+1],2]==0]))
  } else {
    good_ind=append(good_ind,length(poz_ind[(1+meze[j]):meze[j+1],2]
    [poz_ind[(1+meze[j]):meze[j+1],2]==1]))
    bad_ind=append(bad_ind,length(poz_ind[(1+meze[j]):meze[j+1],2]
    [poz_ind[(1+meze[j]):meze[j+1],2]==0]))
  }
}
>hist_ind=rbind(c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1),
good_ind,bad_ind)
>barplot(hist_ind,col = c("lightblue","red"),
        names.arg=c("10%","20%","30%","40%","50%","60%","70%","80%",
        "90%","100%"),ylab = "počet")
>legend("top",c("Negativní jev","Pozitivní jev"),
fill =c("lightblue","red"),inset=c(0,0.3))
```

C.2 Kód pro WOE model

```
# Vytvoření WOE souboru s vybranými regresory pro trénovací a testovací
# vzorky
>woe_train=train_ind[,-c(3,13,17,19)]
>woe_test=test_ind[,-c(3,13,17,19)]
>for (j in 1:15){
  odds_ratio_ind=c()
  kateg=levels(woe_train[,j])
  for (k in 1:length(kateg)){
    odds_ratio_ind=append(odds_ratio_ind,(1/odds_total_train)*
    (length(woe_train$odezva[woe_train$odezva==1&
    woe_train[,j]==levels(woe_train[,j])[k]]))/
    length(woe_train$odezva[woe_train$odezva==0&
    woe_train[,j]==levels(woe_train[,j])[k]]))
  }
  for (k in 1:length(kateg)){
    woe_train[,j]=as.character(woe_train[,j])
    woe_train[,j][woe_train[,j]==kateg[k]]=log(odds_ratio_ind[k])
    woe_test[,j]=as.character(woe_test[,j])
    woe_test[,j][woe_test[,j]==kateg[k]]=log(odds_ratio_ind[k])
  }
  woe_test[,j]=as.numeric(woe_test[,j])
  woe_train[,j]=as.numeric(woe_train[,j])
}
```

```
# Spočítáme parametry regresorů
>set.seed(42)
>woe=glm(odezva ~ ., data = woe_train,family=binomial)
>summary(woe)

# Pustíme dopřednou postupnou selekci
>library(StepReg)
>set.seed(42)
>stepwise_woe=stepwiseLogit(formula = odezva~.,data = woe_train,
sle=0.15, sls=0.2,selection = "forward",sigMethod = "LRT")
# Podíváme se na zbylé regresory
>stepwise_woe$‘Selected Varaibles‘

# Definujeme model, ve kterém zůstanou regresory po dopředné postupné
# selekce
>set.seed(42)
>pro_fk_woe=glm(odezva ~ days_since_eligible+F_credit_history+
F_operator+F_device+lead_age+prev_tele+F_gender+
income+max_emi+products+F_region, data =woe_train, family = binomial)

# Uděláme finální kontrolu
# 1.iterace
>set.seed(42)
>fk_woe=step(pro_fk_woe, direction = "backward", test = "Chisq")
>woe=pro_fk_woe
>summary(woe)


# Otestujeme model, nakreslíme ROC křivku a spočítáme Giniho koeficient
>results_woe <- predict(woe, woe_test,type = "response")
>par(pty = "s")
>roc(woe_test$odezva, results_woe, plot=TRUE, legacy.axes=TRUE,
xlab="Míra falešné pozitivity",  ylab="Míra pravdivé pozitivity")

# Sestrojíme histogram negativních a pozitivních výsledků
>p=as.numeric(results_woe)
>poz_woe=data.frame(p,woe_test$odezva)
>poz_woe=poz_woe[order(p),]
>bad_woe=c()
>good_woe=c()
>meze=as.numeric(round(quantile(c(1:length(p)),
probs=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)),0))
>for (j in 1:10){
  if (j==1){
    good_woe=append(good_woe,length(poz_woe[meze[j]:meze[j+1],2]
    [poz_woe[meze[j]:meze[j+1],2]==1]))
    bad_woe=append(bad_woe,length(poz_woe[meze[j]:meze[j+1],2]
    [poz_woe[meze[j]:meze[j+1],2]==0]))
  } else {
    good_woe=append(good_woe,length(poz_woe[(1+meze[j]):meze[j+1],2]
    [poz_woe[(1+meze[j]):meze[j+1],2]==1]))
```

```
    bad_woe=append(bad_woe,length(poz_woe[(1+meze[j]):meze[j+1],2]
    [poz_woe[(1+meze[j]):meze[j+1],2]==0]))
  }
}
>hist_woe=rbind(c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1),
good_woe,bad_woe)
>barplot(hist_woe,col = c("lightblue","red"),
        names.arg=c("10%","20%","30%","40%","50%","60%",
        "70%","80%","90%","100%"),ylab = "počet")
>legend("top",c("Negativní jev","Pozitivní jev"),
fill = c("lightblue","red"),inset=c(0,0.3))
```

C.3 Kód pro model Plné logistické regrese

```
# Rozdělíme data na trénovací (70%) a testovací (30%) vzorky
>set.seed(42)
>sample_flr <- sample(c(TRUE, FALSE), nrow(datar), replace=TRUE,
prob=c(0.7,0.3))
>train_flr  <- datar[sample_flr, ]
>test_flr   <- datar[!sample_flr, ]

# Spočítáme parametry regresorů
>set.seed(42)
>flr=glm(odezva ~ ., data = train_flr,family=binomial)
>summary(flr)

# Pustíme dopřednou postupnou selekci
>set.seed(42)
>stepwise_flr=stepwiseLogit(formula = odezva~.,data = train_flr,
sle=0.15, sls=0.2,selection = "forward",sigMethod = "LRT")

# Podíváme se na zbylé regresory
>stepwise_flr$‘Selected Varaibles‘

# Definujeme model, ve kterém zůstanou regresory po dopředné postupné
# selekce
>set.seed(42)
>pro_fk_flr=glm(odezva ~ >days_since_eligible+F_credit_history+
F_operator+prev_tele+lead_age+days_since_this_assign+ F_device+
F_gender+F_new_id_card+products+max_emi+F_income_type,
data =train_flr, family = binomial)
>summary(pro_fk_flr)

# Uděláme finální kontrolu
# 1.iterace
>set.seed(42)
>fk_flr=step(pro_fk_flr, direction = "backward", test = "Chisq")
# funkce udělala 2 iterace, kde v první iterace vynechala F_income_type
# (neboť měla největši p-hodnotu) a ve druhé iteraci vynecháme max_emi
# a pustíme to ještě jednou
```

```
# 3.iterace
>set.seed(42)
>fk_flr=step(update(fk_flr, ~ . -products), direction = "backward",
test = "Chisq")
# Vyšlo nám, že products má největší p-hodnotu, takže ji vynecháme
# a pustíme to ještě jednou

# 4.iterace
>set.seed(42)
>fk_flr=step(update(fk_flr, ~ . -F_new_id_card),
direction = "backward", test = "Chisq")
# Vyšlo nám, že F_new_id_card má největší p-hodnotu, takže ji vynecháme
# a pustíme to ještě jednou

# Vyšlo nám, že všechny zbylé regresory jsou významné
>set.seed(42)
>flr=glm(odezva ~ days_since_eligible + F_credit_history +
F_operator +prev_tele + lead_age + days_since_this_assign +
F_device +F_gender, data = train_flr,family=binomial)
>summary(flr)

# Otestujeme model, nakreslíme ROC křivku a spočítáme Giniho koeficient
>results_flr <- predict(flr, test_flr,type = "response")
>par(pty = "s")
>roc(test_flr$odezva, results_flr, plot=TRUE, legacy.axes=TRUE,
xlab="Míra falešné pozitivity", ylab="Míra pravdivé pozitivity")

# Sestrojíme histogram negativních a pozitivních výsledků
>p=as.numeric(results_flr)
>poz_flr=data.frame(p,test_flr$odezva)
>poz_flr=poz_flr[order(p),]
>bad_flr=c()
>good_flr=c()
>meze=as.numeric(round(quantile(c(1:length(p)),
probs=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)),0))
>for (j in 1:10){
  if (j==1){
    good_flr=append(good_flr,length(poz_flr[meze[j]:meze[j+1],2]
    [poz_flr[meze[j]:meze[j+1],2]==1]))
    bad_flr=append(bad_flr,length(poz_flr[meze[j]:meze[j+1],2]
    [poz_flr[meze[j]:meze[j+1],2]==0]))
  } else {
    good_flr=append(good_flr,length(poz_flr[(1+meze[j]):meze[j+1],2]
    [poz_flr[(1+meze[j]):meze[j+1],2]==1]))
    bad_flr=append(bad_flr,length(poz_flr[(1+meze[j]):meze[j+1],2]
    [poz_flr[(1+meze[j]):meze[j+1],2]==0]))
  }
}
>hist_flr=rbind(c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1),
good_flr,bad_flr)
```

```
>barplot(hist_flr,col = c("lightblue","red"), names.arg=c("10%","20%",
"30%","40%","50%","60%","70%","80%","90%","100%"),ylab = "počet")
>legend("top",c("Negativní jev","Pozitivní jev"),
fill = c("lightblue","red"),inset=c(0,0.3))
```

C.4 Kód pro CART (regresní strom)

```
>library(rpart)
>library(rpart.plot)
# Rozdělíme data na trénovací (70%) a testovací (30%) vzorky
>set.seed(42)
>sample_r <- sample(c(TRUE, FALSE), nrow(datar), replace=TRUE,
prob=c(0.7,0.3))
>train_r  <- datar[sample_r, ]
>test_r   <- datar[!sample_r, ]

# Sestrojíme velký strom
>rtree=rpart(odezva~., data=train_r, method="anova",
control=rpart.control(minsplit=10, cp=0))
>plot(rtree)

# Pomoci metody minimálního prořezávání cenové náročnosti dostaneme
# optimální alpha, pomocí které určíme velikost stromu
>cost_table=printcp(rtree)
>bestcp <- rtree$cptable[which.min(rtree$cptable[,"xerror"]),"CP"]
>bestcp=0.00262427

# Sestrojíme strom optimální velikosti
>pruned.tree <- prune(rtree, cp = bestcp)
>rpart.plot(pruned.tree)

# Otestujeme model, nakreslíme ROC křivku a spočítáme Giniho koeficien
>pred.tree = predict(pruned.tree, test_r)
>par(pty = "s")
>roc(test_r$odezva, pred.tree, plot=TRUE, legacy.axes=TRUE,
xlab="Míra falešné pozitivity", ylab="Míra pravdivé pozitivity")

# Sestrojíme histogram negativních a pozitivních výsledků
>p=as.numeric(pred.tree)
>poz_r=data.frame(p,test_r$odezva)
>poz_r=poz_r[order(p),]
>bad_r=c()
>good_r=c()
>meze=as.numeric(round(quantile(c(1:length(p)),
probs=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)),0))
>for (j in 1:10){
  if (j==1){
    good_r=append(good_r,length(poz_r[meze[j]:meze[j+1],2]
    [poz_r[meze[j]:meze[j+1],2]==1]))
    bad_r=append(bad_r,length(poz_r[meze[j]:meze[j+1],2]
```

```
    [poz_r[meze[j]:meze[j+1],2]==0]))
  } else {
    good_r=append(good_r,length(poz_r[(1+meze[j]):meze[j+1],2]
    [poz_r[(1+meze[j]):meze[j+1],2]==1]))
    bad_r=append(bad_r,length(poz_r[(1+meze[j]):meze[j+1],2]
    [poz_r[(1+meze[j]):meze[j+1],2]==0]))
  }
}
>hist_r=rbind(c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1),good_r,bad_r)
>barplot(hist_r,col = c("lightblue","red"), names.arg=c("10%","20%",
"30%","40%","50%","60%","70%","80%","90%","100%"),ylab = "počet")
>legend("top",c("Negativní jev","Pozitivní jev"),
fill = c("lightblue","red"),inset=c(0,0.3))
```

C.5 Kód pro CART (klasifikační strom)

```
# Rozdělíme data na trénovací (70%) a testovací (30%) vzorky
>data_kat=datar
>set.seed(42)
>sample_c <- sample(c(TRUE, FALSE), nrow(data_kat), replace=TRUE,
prob=c(0.7,0.3))
>train_c  <- data_kat[sample_c, ]
>test_c   <- data_kat[!sample_c, ]

# Sestrojíme velký strom
>ctree=rpart(odezva~., data=train_c,
method="class",control=rpart.control(minsplit=10, cp=0))
>plot(ctree)

# Pomoci metody minimálního prořezávání cenové náročnosti dostaneme
# optimální alpha, pomocí které určíme velikost stromu
>cost_table_c=printcp(ctree)
>bestcp_c=ctree$cptable[which.min(ctree$cptable[,"xerror"]),"CP"]
>bestcp_c=0.00095193

# Sestrojíme strom optimální velikosti
>pruned.tree_c <- prune(ctree, cp = bestcp_c)
>rpart.plot(pruned.tree_c,tweak=1.2)

# Spočítáme chybovou matici, odkud dostaneme chybovou míru
>pred.prune_c = predict(pruned.tree_c, test_c, type="class")
>table(pred.prune_c, test_c$odezva)
```

C.6 Kód pro CHAID

```
# Převedeme všechny proměnné na kategoriální
>data_kat=datar
>for (j in 1:19){
```

```
  if (substr(columnames[j],1,2)!="F_"){
    x=data_kat[,j]
    x=ordered(as.factor(x), levels = levels(as.factor(x)))
    data_kat[,j]=x
  }
}
# Rozdělíme data na trénovací (70%) a testovací (30%) vzorky
>set.seed(42)
>sample_ch <- sample(c(TRUE, FALSE), nrow(data_kat), replace=TRUE,
prob=c(0.7,0.3))
>train_ch  <- data_kat[sample_ch, ]
>test_ch   <- data_kat[!sample_ch, ]

# Sestrojíme velký strom
>library("CHAID")
>tree_chaid=chaid(odezva~., train_ch)
>plot(tree_chaid,type = "simple")


# Uděláme křížovou validaci
>set.seed(42)
>folds <- sample(rep(1:5, length.out = nrow(train_ch)))
# Uděláme 100 stromů a vyzkoušíme je na jiném souboru. Pak uděláme
# tabulku velikostí stromu a chyb a vybereme odtud příslušnou velikost
>for (k in 1:5){
  test_set <- subset(train_ch, folds == k)
  train_set <- subset(train_ch, folds != k)
  err=c()
  for (s in 1:100){
   strom=chaid(odezva~., train_set,control =chaid_control(maxheight=s))
    pred_cv <- predict(strom,test_set)
    a=table(pred_cv, test_set$odezva)
    err=append(err,1-(a[1,1]+a[2,2])/sum(a))
  }
  tabl_cv=cbind(tabl_cv,err)
}
# Spočítáme střední hodnotu chyby 5-souborové křížové validace
>tabl_cv=cbind(tabl_cv,
(tabl_cv[,2]+tabl_cv[,3]+tabl_cv[,4]+tabl_cv[,5]+tabl_cv[,6])/5)
# Vyjde nám, že nejoptimálnější velikost je 1
>opv=1

# Sestrojíme strom optimální velikosti
>tree_chaid=chaid(odezva~., test_ch,
control =chaid_control(maxheight=opv))
>plot(tree_chaid,type = "simple")

# Spočítáme chybovou matici, odkud dostaneme chybovou míru
>pred_ch = predict(tree_chaid, test_ch)
>table(pred_ch, test_ch$odezva)
```

C.7 Kód pro Boosting

```
# Rozdělíme data na trénovací (70%) a testovací (30%) vzorky
>set.seed(42)
>sample_b <- sample(c(TRUE, FALSE), nrow(datar), replace=TRUE,
prob=c(0.7,0.3))
>train_b  <- datar[sample_b, ]
>test_b   <- datar[!sample_b, ]

# Převedeme hodnoty závislých proměnných na jiné kódování
>train_b$odezva <- ifelse(train_b$odezva == "0", -1, 1)
>test_b$odezva <- ifelse(test_b$odezva == "0", -1, 1)

# Pustíme Boosting s počtem iterací rovný 100
>library(JOUSBoost)
>ada = adaboost(as.matrix(subset(train_b,select = -20)),
train_b$odezva, tree_depth = 1, n_rounds = 100, verbose = TRUE)
>pred <- predict(ada, as.matrix(subset(test_b,select = -20)))

# Spočítáme chybovou matici, odkud dostaneme chybovou míru
>table(pred,test_b$odezva)
```