UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA

# DIPLOMOVÁ PRÁCE

## Využití metod klasifikace pro podporu online prodeje obuvi

**Katedra matematické analýzy a aplikací matematiky**
Vedoucí bakalářské práce: **Mgr. Kamila Fačevicová, Ph.D.**
Vypracoval(a): **Bc. Tomáš Pohanka**
Studijní program: B1103 Aplikovaná matematika
Studijní obor Aplikace matematiky v ekonomii
Forma studia: prezenční
Rok odevzdání: 2020

# BIBLIOGRAFICKÁ IDENTIFIKACE

**Autor:** Bc. Tomáš Pohanka

**Název práce:** Využití metod klasifikace pro podporu online prodeje obuvi

**Typ práce:** Diplomová práce

**Pracoviště:** Katedra matematické analýzy a aplikací matematiky

**Vedoucí práce:** Mgr. Kamila Fačevicová, Ph.D.

**Rok obhajoby práce:** 2020

**Abstrakt:** Diplomová práce byla psána ve spolupráci s Prabos a. s., společností vyrábějící obuv. V rámci práce bylo úkolem v softwaru R vytvořit a porovnat různé predikční modely, které by mohly být vhodné pro doporučení nejvhodnějšího typu a velikosti obuvi na základě několika charakteristik chodidel. Toto vše s výhledem pomoci potenciálním zákazníkům nakupujícím boty online. V souvislosti s tímto úkolem bylo třeba nejdříve nasbírat data, na kterých byly metody dále trénovány. Součástí práce je teorie k některým použitým metodám v rozsahu, aby byl čtenář srozuměn s principem těchto metod.

**Klíčová slova:** klasifikace dat, regrese, KNN, lokální LDA, náhodné lesy, SVM, PLS, knihovna caret

**Počet stran:** 65

**Počet příloh:** 0

**Jazyk:** český

# BIBLIOGRAPHICAL IDENTIFICATION

**Author:** Bc. Tomáš Pohanka

**Title:** The Application of Classification Methods in Online Sales with Footwear

**Type of thesis:** Master's

**Department:** Department of Mathematical Analysis and Application of Mathematics

**Supervisor:** Mgr. Kamila Fačevicová, Ph.D.

**The year of presentation:** 2020

**Abstract:** The thesis was being written in cooperation with the footwear producer, Prabos, a. s. The main aim of this thesis was to create and to compare a variety of prediction models in order to facilitate choosing the right size and type of boots based on several foot characteristics. This would hopefully help potential company customers shopping online to buy the most suitable footwear for them. Before training the models we had had to collect the data, first. Some of the methods we have used are described in the theoretic part of the Thesis as just in detail as it is needed to make a reader understand using them.

**Key words:** classification, regression, KNN, localized LDA, random forests, SVM, PLS, caret package

**Number of pages:** 65

**Number of appendices:** 0

**Language:** Czech

**Prohlášení**

Prohlašuji, že jsem diplomovou práci zpracoval samostatně pod vedením paní Mgr. Kamily Fačevicové, Ph.D. a všechny použité zdroje jsem uvedl v seznamu literatury.

V Moravském Berouně dne ..... .....

podpis

# Obsah

**Poděkování**

Rád bych poděkoval paní Mgr. Kamile Fačevicové, Ph.D. za vedení této práce, rady a cennou pomoc v organizačních záležitostech. Dále bych chtěl poděkovat paní Mgr. Olze Gelové, panu Bc. Lukáši Najmonovi, paní Renátě Pohankové a slečně Vladimíře Jarošové za pomoc při sběru dat. V neposlední řadě pak děkuji společnosti Prabos, a. s. za poskytnutí pomůcek a technické podpory, bez které by tato práce nemohla vzniknout.

# Úvod

V dnešní době, kdy lidé tráví stále více času ve virtuálním světě, se již spousta běžných činností přesunula on-line. Málo koho už tak překvapí možnost si na internetu pořídit např. elektroniku, jídlo nebo třeba oblečení. Ruku v ruce s výhodami však s sebou tento způsob nakupování přináší i komplikace. Nejspíše každý známe někoho, kdo se někdy rozhodoval, který model telefonu si koupí, aniž by měl možnost si jej osobně vyzkoušet. Nebo třeba kterou velikost oblečení si má vybrat u konkrétního výrobce, aby nebylo příliš velké či naopak příliš malé. Problém volby velikosti se samozřejmě vztahuje i na obuv a společnost Prabos, a. s. se s tímto rozhodla svým zákazníkům pomoci.

Pro pomoc se svým nápadem se zástupci společnosti obrátili na Katedru matematické analýzy a aplikací matematiky naší Univerzity, což stálo u počátku této práce. K problému lze přistoupit tak, že potenciální zákazník poskytne několik charakteristik svých chodidel a vhodná velikost obuvi mu pak bude navržena na základě jejich podobnosti s jinými chodily, u kterých známe i optimální velikost obuvi. Tuto úlohu nazýváme statistická klasifikace a dnes již není problém ji rychle a elegantně vyřešit s využitím softwaru. Práce nicméně nebude omezena pouze na metody klasifikace, ale budeme hledat i další možné přístupy vhodné pro stanovený problém.

Přirozeně nás může napadnout otázka, které charakteristiky chodidel zvolit, a jak je objektivně kvantifikovat. Řešení nabídla sama firma Prabos, a. s., která nám poskytla software pro mobilní telefony, s jehož pomocí lze přibližně určit hodnoty několika proměnných pro každé chodidlo dospělého člověka. Před samotnou výstavbou modelů tedy bude třeba s pomocí této aplikace získat data pro další zpracování. Při sběru těchto dat pak budeme chtít získat hodnoty zkoumaných proměnných pro obě chodidla každé osoby s využitím zmíněné aplikace. Jelikož, jak bylo zmíněno, aplikace pracuje pouze přibližně, provedeme následně i ruční měření délky chodidla, abychom mohli ze získané množiny dodatečně odstranit ta pozorování, která se příliš liší od skutečných hodnot. Každý měřený člověk si následně vyzkouší velikost boty značky Prabos, která mu sedí nejvíce, přičemž volit si může sám ze dvou předem daných typů bot ten, ve kterém se cítí lépe.

Data poté budou dále zpracována v softwaru R a použita pro tvorbu prediktivního modelu. Praktická část bude provázena kódem použitým v tomto soft-

waru. V práci bude použito několik různých metod, proto je nedílnou součástí práce i srovnání míry jejich vhodnosti pro představený problém. Nakonec budeme chtít zjistit, zda se mezi těmito metodami najde taková, kterou by bylo možné dále použít pro tvorbu modelu později využitelného v praxi. V úvodu práce použité metody nejprve rozdělíme dle přístupu k stanovenému problému. Z každé skupiny metod pak zvolíme některou, na které princip objasníme.

Cíle práce tedy jsou nejdříve nasbírat vlastní data pro tvorbu modelů, příprava tohoto souboru ve statistickém softwaru R, především pak vytvoření predikčních modelů, jejich porovnání a vyhodnocení jejich vhodnosti pro stanovený problém.

# Kapitola 1

# Klasifikace

V situaci, kdy si člověk vybírá velikost bot v obchodě, ji většinou nejprve zvolí na základě svých zkušeností, boty si vyzkouší a případně svou volbu dle potřeby změní. Nakonec si však pravděpodobně zvolí tu velikost, která rozměrově nejvíce odpovídá jeho nohám. Ať už máme na mysli velikost, tvar nebo třeba šířku chodidel. Představme si, že všechny důležité údaje o noze umíme získat a proces tohoto výběru se tak můžeme pokusit automatizovat. Pak mluvíme o úloze statistické klasifikace. Samotná úloha se může zdát podobná statistickému shlukování, u klasifikace však předem známe počet klasifikačních tříd, které si můžeme představit jako shluky, ve kterých se pozorování shromažďují dle hodnot proměnných. Z matematického hlediska pak máme datovou matici $\mathbf{X}_{n\times p}$, tedy $n$ $p$-rozměrných pozorování. Každé z těchto pozorování pak náleží do některé z $G$ tříd, přičemž to, že $i$-té pozorování $(x_{i1}, \ldots, x_{ip})^T$ náleží do $g$-té třídy, budeme značit $\mathbf{x}_i \in A_g$.

Na vstupu máme množinu pozorování, u kterých známe i třídu, do které náleží. Na základě této množiny poté chceme vytvořit pravidlo, popřípadě pravidla, dle kterých se budeme rozhodovat, do které třídy zařadíme nová pozorování, tj. ta, u kterých třídu neznáme. Podoba těchto pravidel se pak liší v závislosti na použité metodě. U klasifikačních modelů rozlišujeme přesnost modelu, pokud jej použijeme pro zařazení pozorování, na kterých byl model sestavován (tzv. reklasifikace), a přesnost modelu, pokud jej použijeme pro klasifikaci nových pozorování. Může nastat situace, kdy sestavíme model, který naprosto přesně klasifikuje použitá data, ale při klasifikaci dalších pozorování selhává. Pak se potýkáme s přetrénováním modelu (overfitting). Naopak může nastat situace, kdy model selže při klasifikaci našich dat, nicméně bude úspěšný při klasifikaci nových dat. Ani tato situace není žádoucí, hledáme proto kompromis.

Za tímto účelem je důležité znát predikční schopnost našeho modelu. Většinou totiž nechceme zjistit, že náš model chybně klasifikuje nová pozorování, až v momentě, kdy chceme model dále využívat. Proto je doporučováno data, která máme k dispozici, rozdělit na trénovací množinu (větší část dat) a testovací množinu (zbývající část dat). Při dělení množiny klademe důraz na to, abychom dělení provedli náhodně, dále aby byla relativní četnost pozorování v klasifikačních

třídách v obou datových množinách stejná. V neposlední řadě, pokud data chceme centrovat či škálovat, provádíme tyto úkony na trénovací množině a testovací množinu centrujeme, resp. škálujeme stejnými hodnotami. V opačném případě by obě množiny s sebou nesly část informace té druhé a nebyly by tedy na sobě nezávislé [1, str. 104]. V případech, kdy se potýkáme s nedostatkem pozorování v jednotlivých třídách, můžeme zlepšit odhad predikční schopnosti modelu tak, že dělení dat a tvorbu modelu provedeme opakovaně. Tomuto říkáme křížová validace a podrobněji ji popíšeme v kapitole 1.6.

Měřit, jak moc je konkrétní metoda pro naše data vhodná, můžeme více způsoby. První, základní metodou, je prostá přesnost klasifikace dat. Tuto spočítáme jako podíl přesně klasifikovaných pozorování [2].

$$\text{Přesnost klasifikace} = \frac{\text{Počet správně klasifikovaných pozorování}}{\text{Celkový počet klasifikovaných pozorování}}.$$

Tato metrika nás bude v práci zajímat nejvíce, jelikož chceme zjistit, jaká část zákazníků by při nákupu dostala správnou obuv. Alternativou může být charakteristika Cohenovo kappa [3]. Ta může lépe odhadnout predikční schopnost modelu v situacích, kdy jsou jednotlivé třídy početně zastoupené nerovnoměrně. Cohenovo kappa definujeme následovně

$$\kappa = \frac{P_O - P_C}{1 - P_C}.$$

Charakteristika je obvykle definována jako míra shody klasifikace dvou na sobě nezávislých hodnotitelů. $P_O$ značí podíl pozorování, u kterých při klasifikaci došli ke shodě. $P_C$ pak vyjadřuje pravděpodobnost, se kterou by mohli dosáhnout shody, pokud by se rozhodovali náhodně. Kappa tedy porovnává skutečnou přesnost zvolené metody s přesností, které by dosáhla v případě, že by pozorování zařazovala náhodně. Může tedy nabývat i záporných hodnot v případě, že je přesnost klasifikace nižší, než by byla dosažena náhodnou klasifikací [3]. Výpočet charakteristiky si můžeme ilustrovat na příkladu.

**Příklad 1.** Data v následujícím příkladu jsou smyšlená. Uvažujme dva porotce v pěvecké soutěži, kteří rozhodují o postupu soutěžících do dalšího kola. Rozhodnutí porotců o postupu 50 zpěváků si můžeme prohlédnout v tabulce 1.1.

| | | **Porotce 1** | |
|---|---|---|---|
| | | Ano | Ne |
| **Porotce 2** | Ano | 18 | 7 |
| | Ne | 9 | 16 |

Tabulka 1.1: Tabulka k příkladu 1.

Pro výpočet charakteristiky nejprve spočteme pravděpodobnost shody obou porotců.

$$P_O = \frac{18+16}{18+7+9+16} = 0,68.$$

Dále zjistíme pravděpodobnost, se kterou pošlou, resp. nepošlou, soutěžícího do dalšího kola oba porotci.

$$P_{Ano} = \frac{18+7}{18+7+9+16} \times \frac{18+9}{18+7+9+16} = 0,27,$$

$$P_{Ne} = \frac{9+16}{18+7+9+16} \times \frac{7+16}{18+7+9+16} = 0,23.$$

Z těchto pravděpodobností pak vypočteme šanci, že se oba porotci shodnou, pokud by o postupu rozhodovali náhodně.

$$P_C = 0,27 + 0,23 = 0,5.$$

Nyní už máme vše potřebné pro výpočet Cohenova $\kappa$.

$$\kappa = \frac{0,68 - 0,5}{1 - 0,5} = 0,36.$$

□

Při výpočtu hodnoty charakteristiky pro konkrétní klasifikační metodu bychom namísto tabulky se dvěma hodnotiteli vycházeli z matice záměn (matice, do jejichž řádků jsou zaneseny četnosti predikcí pro jednotlivé klasifikační třídy a do sloupců skutečné četnosti pozorování v těchto třídách).

V následujícím textu si představíme některé základní skupiny predikčních metod, jejichž princip si ukážeme na vybrané metodě.

## 1.1. Metody založené na vzdálenosti

Metody založené na vzdálenosti patří mezi základní klasifikační metody. Mezi nejčastěji užívané metody patří $k$ - nejbližších sousedů (knn), kde $k \in \mathbf{Z}$ je parametr. Nepracuje s žádnými dodatečnými předpoklady ani požadavky na datový soubor [1] a lze ji využít i pro imputaci chybějících hodnot v datovém souboru [2]. Principem metody je vyhledat $k$ nejbližších sousedů pozorování $\mathbf{x}_{n+1}$, které chceme klasifikovat. Jako třídu, do které pozorování zařadíme, pak zvolíme tu, která se mezi těmito $k$ sousedy vyskytovala nejčastěji. Speciálně pro $k = 1$ je pozorování klasifikováno do stejné třídy jako jeho nejbližší soused, což bývá označováno jako metoda nejbližšího souseda. Často se pak $k$ volí jako liché číslo, což může pomoci předejít situaci, kdy jsou mezi sousedy nejčastěji zastoupeny dvě, nebo více tříd. V tomto případě pak metoda volí mezi těmito třídami

náhodně [4]. Vyšší hodnoty parametru mohou vést k tendenci metody zařazovat nová pozorování do nejpočetnějších tříd. Volba nižší hodnoty parametru pak může vést k tomu, že metoda bude ovlivněna šumem v datech. Optimální hodnotu parametru můžeme stanovit s pomocí křížové validace (viz kapitola 1.6). Pro měření vzdálenosti od sousedů lze volit mezi různými metrikami, často využívanou je metrika Euklidova [5]:

$$d_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + \ldots + (x_{ip} - x_{jp})^2}. \tag{1.1}$$

Vzdálenost je velmi citlivá na jednotky, ve kterých jsou dílčí proměnné udávány, proto je velmi důležité proměnné před výpočty škálovat. Pokud máme datový soubor se značně nevyváženým počtem pozorování v jednotlivých třídách, můžeme alternativně využít např. centroidovou metodu, resp. metodu průměrné vazby.

## 1.2. Metody diskriminační analýzy

Diskriminační analýza je souhrnné označení pro metody, které klasifikují pozorování na základě hodnoty diskriminační funkce. Tyto metody vychází z předpokladu, že pozorování v $g$-té třídě pochází všechna ze stejného pravděpodobnostního rozdělení určeného funkcí hustoty $f_g(\mathbf{x}),\ g = 1, \ldots, G$ [5]. Abychom byli schopni třídy odlišit, je navíc žádoucí, aby nebyly žádné dvě z těchto funkcí stejné

$$f_{g_1}(\mathbf{x}) \neq f_{g_2}(\mathbf{x}),\ \forall g_1, g_2 = 1, \ldots, G,\ g_1 \neq g_2.$$

Často pak předpokládáme, že objekt v $g$-té třídě pochází z $p$-rozměrného normálního rozdělení $N_p(\mu_g, \mathbf{\Sigma}_g)$ nicméně některé z metod od tohoto požadavku upouští (např. MDA). Dle [6, str. 522] navíc metody nejsou na porušení předpokladu normality citlivé. Pravidlo pro klasifikaci nových pozorování můžeme vyjádřit v různých tvarech, např. jako pravidlo maximální věrohodnosti. Dle tohoto pravidla vyčíslíme hodnotu funkce hustoty každé třídy v bodě $\mathbf{x}_{n+1}$ a pozorování zařadíme do třídy, hodnota jejíž funkce bude maximální

$$\mathbf{x}_{n+1} \in A_g,\ g = \arg \max_{g \in 1,\ldots,G} f_g(\mathbf{x}_{n+1}).$$

Jiné pravidlo je založeno na minimalizaci vzdálenosti nového pozorování od středů jednotlivých tříd [1, str. 105]. V tomto případě uvažujeme Mahalanobisovu vzdálenost

$$d_M(\mathbf{x}, \mu_g) = (\mathbf{x} - \mu_g)^T \mathbf{\Sigma}_g^{-1} (\mathbf{x} - \mu_g), \tag{1.2}$$

kde $\mu_g$ je vektor středních hodnot proměnných v rámci $g$-té třídy a $\mathbf{\Sigma}_g$ je jejich varianční matice.

### 1.2.1. Lokální lineární diskriminační analýza

Nadále pro připomenutí uvažujeme tréninkovou množinu o $n$ $p$-rozměrných pozorování. Každé z těchto pozorování náleží do některé z $G$ tříd. Úkolem bude sestavit s pomocí tréninkové množiny rozhodovací pravidlo, pomocí kterého budeme zařazovat do tříd nové objekty. U lineární diskriminační analýzy (LDA) předpokládáme, že objekty v $g$-té třídě pochází z $p$-rozměrného normální rozdělení se střední hodnotou $\mu_g$ a kovarianční maticí $\mathbf{\Sigma}$, která je pro všechny třídy stejná. Označme dále $\pi_g$ apriorní pravděpodobnost příslušnosti objektu do $g$-té třídy. Nové pozorování $\mathbf{x}_{n+1}$ zařadíme do třídy s nejvyšší aposteriorní pravděpodobností, což můžeme ekvivalentně vyjádřit s pomocí maximalizace diskriminačních funkcí [7]

$$h_g(\mathbf{x}) = (\mathbf{\Sigma}^{-1}\mu_g)^T\mathbf{x} - \frac{1}{2}\mu_g^T\mathbf{\Sigma}^{-1}\mu_g + \ln(\pi_g). \tag{1.3}$$

Skutečné hodnoty parametrů rozdělení obvykle neznáme, proto je v rovnici nahradíme výběrovými charakteristikami

$$\hat{h}_g(\mathbf{x}) = (\mathbf{S}^{-1}\bar{\mathbf{x}}_g)^T\mathbf{x} - \frac{1}{2}\bar{\mathbf{x}}_g^T\mathbf{S}^{-1}\bar{\mathbf{x}}_g + \ln(p_g), \tag{1.4}$$

kde $\bar{\mathbf{x}}_g$ značí vektor výběrových průměrů proměnných v rámci $g$-té třídy, $\mathbf{S}$ je výběrová kovarianční matice celého datového souboru a $p_g$ značí relativní četnost pozorování v $g$-té třídě.

Lokální obdoba LDA spočívá v tom, že nyní budeme při výpočtu charakteristik v diskriminačních funkcích uvažovat pouze $k$ nejbližších pozorování klasifikovanému objektu $\mathbf{x}_{n+1}$. Tato myšlenka je založena na domněnce, že některá dostatečně vzdálená pozorování mají na klasifikaci nového pozorování pouze malý vliv, proto je při klasifikaci můžeme zanedbat [7]. Dále zavedeme s pomocí vah $\omega_i$ vážený průměr $\bar{\mathbf{x}}_{g_L}$ a váženou relativní četnost $p_{g_L}$.

$$\bar{\mathbf{x}}_{g_L} = \frac{\sum_i \omega_i \mathbf{x}_i I_{\{\mathbf{x}_i \in A_g\}}}{\sum_i \omega_i I_{\{\mathbf{x}_i \in A_g\}}}, \quad p_{g_L} = \frac{\sum_i \omega_i I_{\{\mathbf{x}_i \in A_g\}}}{\sum_i \omega_i}.$$

Případně můžeme uvažovat apriorní pravděpodobnost stejnou pro všechny třídy. Pro odhad výběrové vážené varianční matice nejprve spočteme výběrovou varianční matici každé třídy. S pomocí těchto matic následně spočteme celkovou varianční matici [7]

$$\mathbf{S}_{g_L} = \frac{1}{1 - \sum_i \omega_i^2 I_{\{\mathbf{x}_i \in A_g\}}} \sum_i \omega_i \left[(\mathbf{x}_i - \bar{\mathbf{x}}_{g_L}) I_{\{\mathbf{x}_i \in A_g\}}\right] \left[(\mathbf{x}_i - \bar{\mathbf{x}}_{g_L}) I_{\{\mathbf{x}_i \in A_g\}}\right]^T,$$

$$\mathbf{S}_L = \frac{n}{n - G} \sum_g p_g \mathbf{S}_{g_L}.$$

Váhy pak volíme následovně

$$\omega_i = K\left(\frac{d_E(\mathbf{x}_i, \mathbf{x}_{n+1})}{d_E(\mathbf{x}_k, \mathbf{x}_{n+1})}\right),\ i = 1, \ldots, k.$$

K je některá váhová funkce, po které vyžadujeme, aby byla omezená na intervalu $\langle 0, \infty \rangle$, $d_E(\mathbf{x}_i, \mathbf{x}_{n+1})$ značí euklidovskou vzdálenost nového pozorování od $i$-tého pozorování a $d_E(\mathbf{x}_k, \mathbf{x}_{n+1})$ je opět euklidovská vzdálenost, tentokrát však mezi $\mathbf{x}_{n+1}$ a $k$-tým nejbližším pozorováním [8].

Klasifikační pravidlo pak má následující podobu

$$\hat{A}_g = \begin{cases} \arg\max\limits_g \hat{h}_{g_L}(\mathbf{x}),\ \exists g : \exp(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_g)^T \mathbf{S}_L^{-1}(\mathbf{x} - \bar{\mathbf{x}}_g)) > \frac{10^{-150}}{p_{g_L}}, \\ \arg\min\limits_g d_E(\mathbf{x}, \bar{\mathbf{x}}_g) \text{ jinak.} \end{cases}$$

Pravidlo v tomto tvaru nám říká, že nové pozorování přiřadíme do třídy, jejíž lokální diskriminační funkce je v daném bodě maximální. Dodatečná podmínka slouží pro vzácné případy, kdy se nové pozorování liší od vektoru průměrů všech G tříd natolik, že jsou všechny aposteriorní pravděpodobnosti příslušnosti do tříd téměř rovny nule. V tomto případě pozorování přiřadíme do třídy s nejmenší vzdáleností od vektoru výběrových průměrů [7].

## 1.3. Náhodné lesy

Náhodné lesy se, ostatně jako i další lesy, skládají ze stromů. V tomto případě se konkrétně jedná o stromy klasifikační. Abychom si mohli vysvětlit náhodné lesy, je nezbytné si nejdříve představit tyto stromy. Cílem klasifikačních stromů je zařadit pozorování do třídy na základě série otázek definující klasifikační pravidla. Na otázky lze v základním případě odpovědět pouze Ano / Ne (binární stromy). Velmi dobře tedy pracují s kategoriálními daty. Vhodnou formulací otázky lze však klasifikační stromy rozšířit i pro případ s numerickými proměnnými, např. Je délka nohy větší než 200 mm? Je šířka nohy v intervalu 100 mm – 120 mm? Obdobně můžeme postupovat i pro ordinální proměnné. Pokud uvažujeme navíc i spojitou závislou proměnnou, mluvíme pak o regresních stromech.

Každému pravidlu ve stromu říkáme uzel, speciálně tomu prvnímu pak říkáme kořen. Kořen se v klasifikačním stromu nachází nahoře a kladením dalších otázek postupujeme směrem dolů. Každá ze dvou možných odpovědí v uzlu tvoří novou větev stromu. Volba uzlů v každé z větví je nezávislá na volbě uzlů v jiné větvi. Ve dvou větvích vycházejících ze stejného uzlu tak můžeme dále postupovat s různými proměnnými nebo různými dělicími hodnotami té samé proměnné.

Strom větvíme až do chvíle, kdy všechna pozorování zařadíme do některé ze tříd, nebo vyčerpáme možnosti pro volbu dalších uzlů. Tento poslední uzel,

který už dále nevětvíme, nazýváme list. Otázkou však zůstává, jak zvolit vhodnou proměnnou, která data v daných uzlech rozdělí nejlépe. Případně jak zvolit hodnotu proměnné, s pomocí které budeme data dělit. K tomuto lze přistoupit různými způsoby, metoda zde popsaná se nazývá CART.

Dle této metody obvykle postupujeme tak, že rozdělíme data do tříd dle každé proměnné, kterou jsme pro strom ještě nevyužili. Hodnoty spojitých proměnných, kterých proměnná v našich datech nabývá, navíc seřadíme dle velikosti a jako možné dělení určíme každý průměr dvou sousedních hodnot. Pro daný uzel zvolíme proměnnou (resp. dělicí hodnotu), podle které data klasifikujeme do tříd co nejlépe. Aby tato volba zůstala objektivní, pokusíme se s pomocí tzv. kriteriální statistiky kvantifikovat to, jak dobře proměnná data dělí do tříd. Tuto charakteristiku můžeme vyjádřit ve více tvarech. Využít můžeme např. index Gini nebo entropii [9]. Hodnotu charakteristiky nejprve počítáme pro obě větve vycházející z uzlu. Index Gini v tomto tvaru

$$GI = \sum_{i \neq j} p_i p_j = 1 - \sum_i p_i^2, \tag{1.5}$$

kde $p_i$ je pravděpodobnost, že v uzlu bude pozorování klasifikováno do $i$-té třídy a $p_j$ pravděpodobnost, že pozorování klasifikujeme třídy jiné. Tyto pravděpodobnosti odhadneme jako relativní četnosti pozorování ve třídách v dané větvi.

Míra entropie vypadá následovně

$$H = -\sum_i p_i^2 \log p_i.$$

Obě statistiky dosahují minima v případě, kdy uzel zahrnuje pouze pozorování z jedné třídy, a toto minimum je rovno 0. Tzn. že nižší hodnoty indexů znamenají lepší dělení v dané větvi.

Indexy vypočtené pro dílčí větve následně agregujeme pro celý uzel [1]. Do uzlu pak zvolíme proměnnou, pro kterou bude vážený průměr (1.6) minimální.

$$GI_{celk} = P_l GI(p_l) + P_r GI(p_r), \text{ resp. } H_{celk} = P_l H(p_l) + P_r H(p_r), \tag{1.6}$$

kde $P_l$ a $P_r$ značí podíl všech pozorování v uzlu v levé, resp. pravé větvi. $GI(p_l)$ a $GI(p_r)$ jsou hodnoty indexu Gini v levé, resp. pravé větvi. Obdobně pro míru entropie.

Pokud bychom uvažovali pouze některé ze zmíněných pravidel, přidávali bychom větve až do chvíle, kdybychom se dostali až k uzlům, které by obsahovaly jen pozorování z jedné třídy, resp. nevyčerpali všechny proměnné v souboru. Takto bychom pravděpodobně získali poměrně přesnou klasifikaci pro trénovací množinu, nicméně snadno může dojít k přetrnénování modelu. Přistoupíme tedy k prořezávání stromu, tj. k odstranění nevýznamných větví. Matematicky minimalizujeme funkci

$$C(T) = R(T) + \alpha\,|T|\,,$$

kde $T$ je strom s $|T|$ listy, $R(T)$ je podíl chybně klasifikovaných pozorování a $\alpha \in (0, \infty)$ je penalizační parametr.

Uvedený postup výpočtu kriteriálních statistik může působit abstraktně, proto si jej představíme ještě jednou, tentokrát na příkladu.

| | donated | age | wealth_rating | interest_religion | frequency |
|---|---|---|---|---|---|
| **292** | 0 | 67 | 2 | 0 | INFREQUENT |
| **900** | 1 | 76 | 1 | 1 | FREQUENT |
| **1879** | 1 | 63 | 1 | 0 | INFREQUENT |
| **1162** | 0 | 66 | 3 | 0 | FREQUENT |
| **6467** | 0 | 63 | 0 | 0 | INFREQUENT |
| **3165** | 1 | 53 | 0 | 0 | FREQUENT |

Obrázek 1.1: Datový soubor donors.

**Příklad 2.** Výpočet kriteriálních statistik si vyzkoušíme na datovém souboru s údaji o peněžních dárcích organizaci pomáhající veteránům [10], jehož část využitou v příkladě si můžeme prohlédnout na obr. 1.1. Z datového souboru vybereme náhodně tři osoby, které dar poskytly, a tři, které ne. Pozorování budeme chtít rozdělit na dárce a ty, kteří dar neposkytli. Klasifikaci budeme provádět na základě věku, majetku osoby (na škále 0–3), náboženství a četnosti poskytování darů.

Nejprve budeme chtít stanovit kořen stromu. Podíváme se proto, jak jednotlivé proměnné rozdělí pozorování na dárce a ty ostatní. Nejvhodnější proměnnou stanovíme s pomocí indexu Gini.

Výpočet kriteriální statistiky si ukážeme nejprve na kategoriální proměnné `interest_religion`. V souboru je pět nevěřících a jedna osoba, která je věřící. Z nevěřících jsou pak tři lidé, kteří nepřispěli, a dva, kteří ano. S pomocí vztahu (1.5) pak

$$GI = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0,48.$$

Jediná věřící osoba v souboru dar poskytla

$$GI = 1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0.$$

Pro zvolenou proměnnou pak spočítáme vážený průměr s pomocí vztahu (1.6)

$$GI_{celk} = \frac{5}{6} \times 0,48 + \frac{1}{6} \times 0 = 0,4.$$

U proměnné věk postupujeme tak, že nejprve podle ní data seřadíme, spočítáme průměrný věk všech sousedních pozorování a spočítáme index obdobně, jako v předchozím případě. Např. průměrný věk mezi třetím a čtvrtým pozorováním v seřazeném souboru je 64,5. Data rozdělíme dle tohoto věku. Z osob mladších, než tento průměr, darovaly dvě. Z osob starších darovala jen jedna

$$< 64,5 : GI = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 \doteq 0,44,$$

$$> 64,5 : GI = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 \doteq 0,44,$$

$$GI_{celk} = \frac{3}{6} \times 0,44 + \frac{3}{6} \times 0,44 = 0,44.$$

Jako optimální dělení dle této proměnné pak zvolíme dělicí hodnotu s nejnižší hodnotou indexu Gini. U ordinální proměnné týkající se majetku osob index počítáme pro všechny možné případy, tzn. kdy je hodnota proměnné menší nebo rovna nule, poté menší nebo rovna jedné a nakonec menší nebo rovna dvěma.

Jakmile vypočteme hodnotu indexu pro všechny proměnné a všechna jejich možná dělení, zvolíme jako kořen proměnnou, pro kterou jsme získali nejnižší hodnotu indexu Gini. V našem případě dosáhneme nejnižší hodnoty s proměnnou `wealth_rating` $\leq 1$.

$$\leq 1 : GI = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0,375,$$

$$> 1 : GI = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0,$$

$$GI_{celk} = \frac{4}{6} \times 0,375 + \frac{2}{6} \times 0 = 0,25.$$

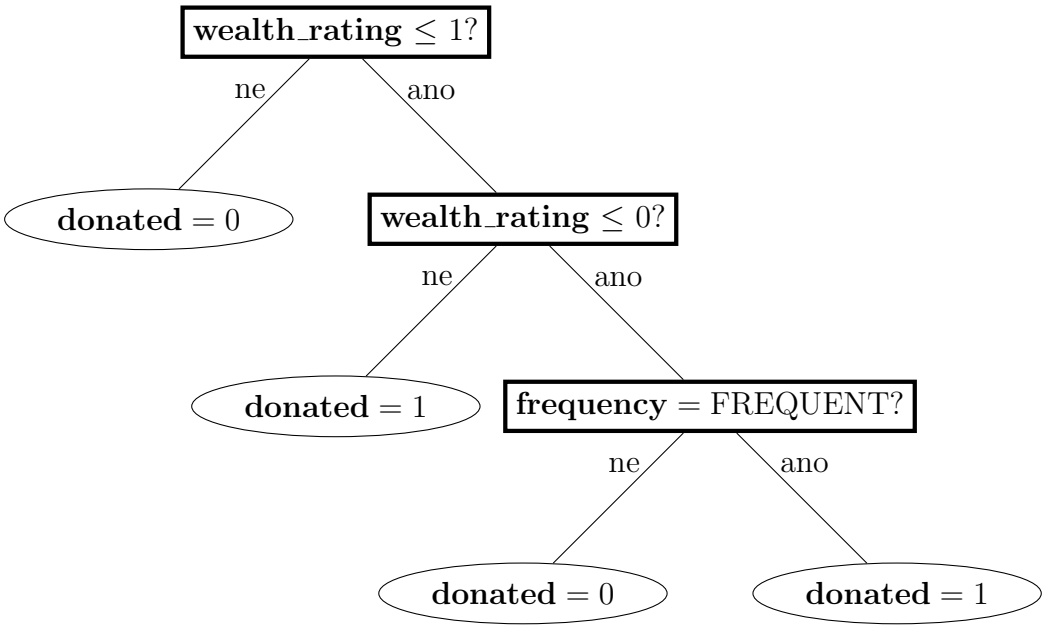
Jedna z větví vedoucí z kořene data rozdělí dokonale, proto ji označíme jako list. Druhou větev můžeme dále dělit. V dalším kroku zjistíme, že stejnou minimální hodnotu indexu získáme pro několik různých dělení. Zvolíme tedy např. opět proměnnou `wealth_rating` tentokrát s dělicí hodnotou 0. Stejnou proměnnou tedy můžeme do uzlu zvolit opakovaně s různými dělicími hodnotami.

$$\leq 0 : GI = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = 0,5,$$

$$> 0 : GI = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0,$$

$$GI_{celk} = \frac{2}{4} \times 0,5 + \frac{0}{4} \times 0 = 0,25.$$

Jednu z větví opět můžeme označit jako list. Druhou dále rozdělíme. Tentokrát můžeme volit mezi `age` $\leq 58$ a proměnnou `frequency` se stejnými hodnotami indexů. Obě dělení povedou k poslednímu listu. Rozhodovací strom z toho příkladu si můžeme prohlédnout na obrázku 1.2.




Obrázek 1.2: Rozhodovací strom.

□

Rozhodovací stromy lze poměrně jednoduše sestavit i interpretovat. Dobře klasifikují data, na nichž byl strom postaven, nicméně u nových pozorování mohou při predikci selhávat. Náhodné lesy, skládající se z těchto stromů, predikční schopnost zlepšují. Náhodné jim říkáme pro způsob, jakým volíme data pro sestavení dílčích stromů. V prvním kroku totiž z našich dat náhodně vybereme $n$ pozorování, tentokrát však s opakováním (technika zvaná bootsrapping). Dále v každém rozhodovacím uzlu uvažujeme pouze $m < n$ náhodně zvolených proměnných bez opakování. Dle [11, str. 96] je doporučeno volit $m = \sqrt{p}$. Takto budeme náhodný výběr několikrát opakovat a na každé takovéto datové množině sestavíme rozhodovací strom. Pozorování, která nebyla použita pro sestavení jednotlivých stromů, navíc klasifikujeme s pomocí celého lesa. Jako výslednou třídu volíme tu, která byla stromy zvolena nejčastěji. Pokud se mezi třídami najde více takových, je výsledná třída volena náhodně [12, str. 509]. Tento postup pak slouží pro odhad, jak dobré predikční schopnosti náš les má.

## 1.4. Metoda podpůrných vektorů

Stejně jako klasifikační stromy, lze i metodu podpůrných vektorů (support vector machines) využít jak pro klasifikaci, tak pro regresní analýzu [5]. Principem metody je najít lineární hranici (obecně nadrovinu, pro dvourozměrnou úlohu pak přímku, ve třech dimenzích rovinu), která odděluje pozorování z různých klasifikačních tříd. Tuto úlohu nemusí být vždy jednoduché vyřešit, jelikož se skupiny pozorování z různých tříd často vzájemně prolínají. Tento problém metoda řeší s pomocí transformace pozorování do vyšší dimenze, kde je již obvykle řešitelný. Pokud jsou dvě třídy lineárně separabilní, hranici hledáme tak, abychom maximalizovali oblast, která je určená nejmenší vzdáleností mezi dvěma pozorováními z různých tříd. Při zpětné transformaci jsou pak tyto hranice nelineární. V dalším textu budeme předpokládat, že máme pouze dvě klasifikační třídy, nicméně metodu lze samozřejmě rozšířit i pro více tříd (více v [13]).

Zmíněnou nadrovinu v prostoru vyšší dimenze, označme tuto dimenzi $r$, hledáme obecně ve tvaru

$$b_0 + \mathbf{b}^T\mathbf{x} = 0 \tag{1.7}$$

s koeficienty $b_0$, $\mathbf{b} = (b_1, b_2, \cdots, b_r)^T$ a vektorem $\mathbf{x} = (x_1, x_2, \cdots, x_r)^T$ o $r$ proměnných. Aby byla nadrovina definována jednoznačně, uvažujeme navíc podmínku $\mathbf{b}^T\mathbf{b} = 1$. Takto definovanou nadrovinu můžeme využít k separaci objektů na základě jejich polohy vůči ní, tzn. zda je výraz $b_0 + \mathbf{b}^T\mathbf{x}_i$ větší nebo menší jak nula [5]. Předpokládejme nyní, že jsou obě klasifikační třídy lineárně separabilní. Dále přiřaďme každému pozorování hodnotu $y_i \in \{-1, 1\}$ dle třídy, do které patří. Pak můžeme s využitím rovnice (1.7) definovat následující klasifikační pravidlo

$$y_i(b_0 + \mathbf{b}^T\mathbf{x}_i) > 0$$

pro $i = 1, \cdots, n$.

Pro správně klasifikovaný objekt tedy platí $\text{sgn}(y_i) = \text{sgn}(b_0 + \mathbf{b}^T\mathbf{x}_i)$. Takovýchto nadrovin bychom však nalezli nekonečně mnoho. Dělicí nadrovinu v optimální pozici označíme jako nadrovinu s maximálním okrajem a budeme ji hledat tak, že maximalizujeme její odstup, značíme $M$, od bodů z různých tříd [13, str. 341]. Odstup je určen nejmenší vzdáleností dělicí nadroviny od jedné ze dvou k ní rovnoběžných nadrovin. Ve dvourozměrném prostoru jsou tyto nadroviny jednoznačně určeny třemi body nacházejícími se nejblíže k dělicí nadrovině, dvěma z jedné třídy a jedním ze druhé třídy [5]. S rostoucím počtem dimenzí dat, pak roste počet bodů potřebných pro určení oblasti $M$. Tyto body nazýváme podpůrné vektory.

Takovouto úlohu (na obr. 1.3) můžeme formulovat ve tvaru

$$\mathrm{M} \to \max \text{ pro } b_0,\ \mathbf{b} = (b_1, b_2, \cdots, b_r)^T,\ \mathbf{b}^T\mathbf{b} = 1$$
$$\text{vzhledem k } y_i(b_0 + \mathbf{b}^T\mathbf{x_i}) \geq \mathrm{M} \quad \text{pro } i = 1, \cdots, n.$$

Dále úlohu rozšíříme pro případ, kdy jsou obě třídy vzájemně lineárně neoddělitelné i v novém transformovaném prostoru. I nadále budeme chtít maximalizovat $M$, nyní ale budeme muset tolerovat chyby v klasifikaci. Tzn. že některá pozorování ponecháme na špatné straně nadroviny. Úlohu doplníme o proměnné $\xi_i$, kterými kvantifikujeme chybu klasifikace. Tato proměnná nabývá 0 pro pozorování na správné straně nadroviny. Jinak nabývá kladného čísla a vyjadřuje vzdálenost pozorování od dělicí nadroviny.

Úlohu (na obr. 1.4) nyní formulujeme takto

$$\mathrm{M} \to \max \text{ pro } b_0,\ \mathbf{b} = (b_1, b_2, \cdots, b_r)^T,\ \mathbf{b}^T\mathbf{b} = 1$$
$$\text{vzhledem k } y_i(b_0 + \mathbf{b}^T\mathbf{x_i}) \geq \mathrm{M}(1 - \xi_i) \quad \text{pro } i = 1, \cdots, n,\ \xi_i \geq 0,\ \sum_i \xi_i \leq C.$$

Konstantou $C$ omezujeme chybu při klasifikaci pozorování. Obecně můžeme říct, že s větší hodnotou $C$ poroste velikost $M$. Optimální hodnotu této konstanty můžeme určit s pomocí křížové validace [13]. Klasifikační pravidlo, které získáme vyřešením úlohy nazýváme klasifikátor s podpůrnými vektory.

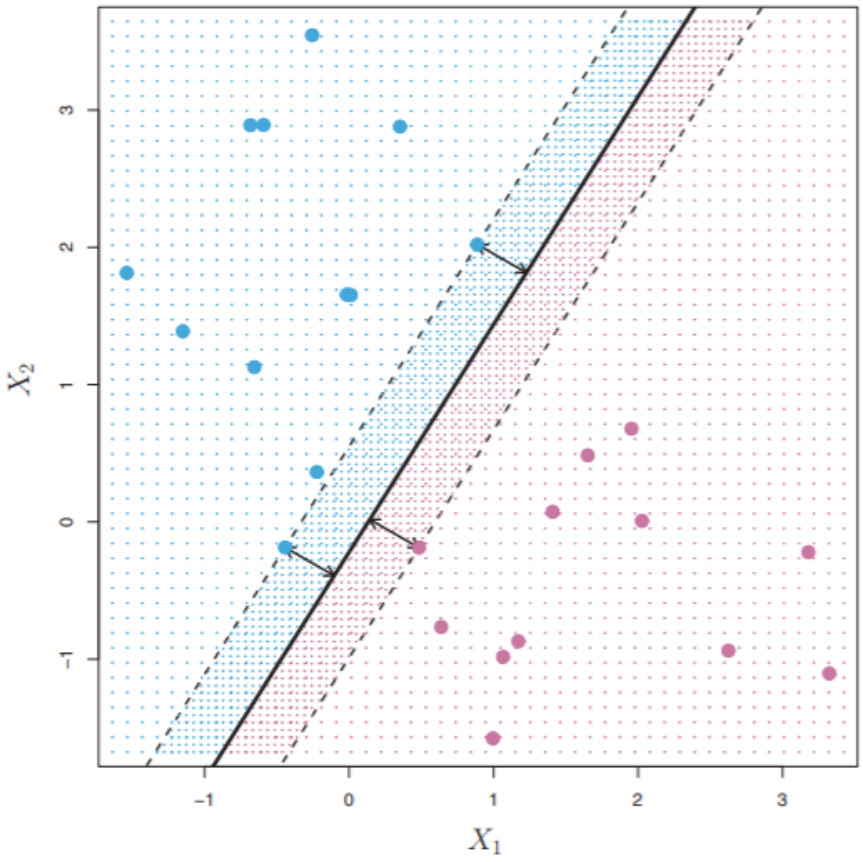
Výše zmíněné úlohy řešíme za předpokladu, že pozorování v $p$-rozměrném prostoru transformuje do prostoru o dimenzi $r$. Transformace prostoru společně s řešením maximalizačního problému úlohu poměrně komplikuje. Výpočty lze ale zjednodušit s pomocí tzv. jádrového triku. Ten spočívá v tom, že pozorování nepřevedeme do prostoru vyšší dimenze přímo, nicméně zavedením vhodné funkce můžeme získat představu o vztahu dvou pozorování v tomto prostoru. Lineární klasifikátor s podpůrnými vektory nejprve ekvivalentně vyjádříme s pomocí skalárního součinu

$$\langle \mathbf{x}_i, \mathbf{x}_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

ve tvaru

$$f(\mathbf{x}_{n+1}) = b_0 + \sum_{i=1}^{n} \alpha_i \langle \mathbf{x}_{n+1}, \mathbf{x}_i \rangle, i = 1, \ldots, n,$$

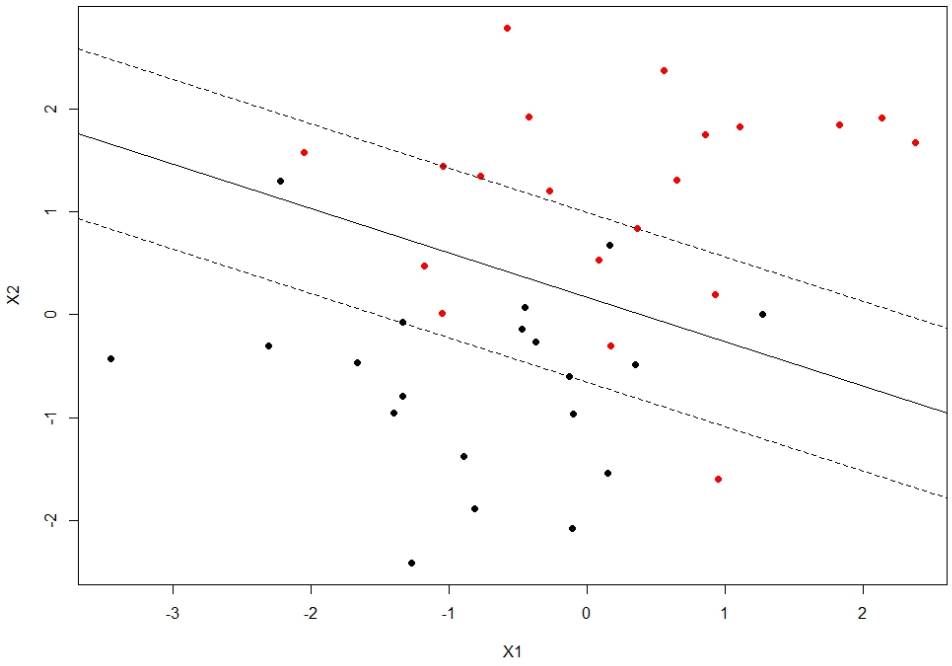
přičemž klasifikace pozorování závisí na znaménku funkce $f(\mathbf{x}_{n+1})$. Pro odhad parametrů $b_0$ a $\alpha_i$ je třeba vypočítat skalární součin všech dvojic pozorování v tréninkové množině. Tyto parametry jsou nicméně nenulové jen pro podpůrné vektory [13]. Skalární součin můžeme dále zobecnit s pomocí některých funkcí. Těmto funkcím říkáme jádrové a můžeme je zavést v několika tvarech




Obrázek 1.3: Příklad klasifikátoru s podpůrnými vektory pro dvě lineárně separabilní třídy [14].

Lineární: $K(\mathbf{x}_i, \mathbf{x}'_i) = \mathbf{x}_i^T \mathbf{x}'_i$.
Polynom stupně $d$: $K(\mathbf{x}_i, \mathbf{x}'_i) = (1 + \mathbf{x}_i^T \mathbf{x}'_i)^d$.
Radiální bázová funkce: $K(\mathbf{x}_i, \mathbf{x}'_i) = \exp(-c\, d_E^2(\mathbf{x}_i, \mathbf{x}'_i))$ pro $c > 0$.

Uvedené jádrové funkce jsou ty, které se používají nejčastěji. Funkci, kterou chceme při klasifikaci použít, musíme pro algoritmy v softwaru určit předem. Pro některé známé klasifikační problémy lze nalézt doporučení, kterou z nich použít. Každopádně parametry funkce, pro které bude metoda dosahovat nejlepší predikce, můžeme určit s pomocí křížové validace (kapitola 1.6).




Obrázek 1.4: Příklad klasifikátoru s podpůrnými vektory pro dvě lineárně neseparabilní třídy.

## 1.5. Regresní analýza

U klasifikace požadujeme, aby třídy nabývaly pouze diskrétních hodnot (závisle proměnná je diskrétní). Toto platí i pro naši úlohu, nicméně vhodnou úpravou závisle proměnné ji lze pojmout i jinak. Pokud naše třídy dokážeme uspořádat dle velikosti, můžeme je pak vyjádřit jako ordinální numerickou diskrétní proměnnou. Pro tento typ problému již existují algoritmy, které představují přechod mezi regresní analýzou a klasifikací (např. PLSDA). Nicméně v této práci náš přístup zjednodušíme a využijeme klasickou regresní analýzu. K důvodům této volby se vrátíme v kapitole 3.

Úlohou regresní analýzy je tedy kvantifikovat a vyjádřit vztah mezi prediktory a závisle proměnnými. Pro tuto úlohu obvykle volíme přístup s využitím metody nejmenších čtverců. Tato metoda však selhává v momentech, kdy máme v datech více regresorů než-li pozorování, tzn. $n < p$, nebo dále pokud můžeme v matici nezávislé proměnných pozorovat multikolinearitu (její sloupce jsou lineárně závislé). V obou situacích nám může pomoci metoda částečných nejmenších čtverců, kterou si představíme blíže [5].

### 1.5.1. Metoda částečných nejmenších čtverců

Mějme nejprve pouze jednu závisle proměnnou $\mathbf{y}_{n\times 1}$ a nezávisle proměnné uspořádané do matice $\mathbf{X}_{n\times p}$, jak je v regresi zvykem. Stejně jako v případě metody nejmenších čtverců budeme hledat vztah

$$\mathbf{y} = \mathbf{X}\beta + \epsilon,$$

kde $\beta_{p\times 1}$ je vektor regresních koeficientů a $\epsilon_{n\times 1}$ vektor chyb.

V PLS, obdobně jako třeba v analýze hlavních komponent, však nejprve transformujeme matici $\mathbf{X}$ na matici skórů a zátěží dle vztahu

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}_\mathbf{X},$$

kde matice $\mathbf{T}_{n\times k}$ je matice skórů, $\mathbf{P}_{p\times k}$ je matice zátěží, což jsou koeficienty lineární kombinace skórů, a matice $\mathbf{E}_\mathbf{X}$ jsou chyby, kterých se dopouštíme aproximací původních proměnných. Tento člen odpadá, pokud zvolíme počet komponent $k$ roven počtu původních proměnných. Na rozdíl od PCA však tentokrát nové proměnné nehledáme tak, aby byl maximální jejich rozptyl, nýbrž kovariance mezi skóry a proměnnou $\mathbf{y}$.

Existuje několik algoritmů pro hledání nových proměnných, nejčastěji však pracují na následujícím principu [5]:

- První komponenta je vypočtena tak, aby byla maximilizována kovariance mezi ní a proměnnou $\mathbf{y}$.
- Informace (rozptyl) komponenty je odstraněn z dat, což nazýváme deflace. Z geometrického hlediska je deflace projekce datové matice do podprostoru kolmého na komponentu z předešlého kroku. Po deflaci získáme matici $\mathbf{X}_{RES}$, která má stejný počet proměnných jako původní matice, ale její dimensionalita je nižší o 1.
- Z matice $\mathbf{X}_{RES}$ je opět vypočtena další komponenta tak, aby byla kovariance mezi ní a proměnnou $\mathbf{y}$ maximální.
- Postup opakujeme, dokud nepřestane docházet ke změně modelu.

V závislosti na použitém algoritmu pak metoda generuje buďto ortogonální skóry, nebo ortogonální zátěže. S každou přidanou komponentou modelujeme proměnnou $\mathbf{y}$ lépe. Optimální počet komponent pro predikci metodou odhadujeme s pomocí křížové validace. K původním regresním parametrům se můžeme vrátit s pomocí následujících vztahů

$$\begin{aligned}
&\mathbf{y} = \mathbf{X}\beta + \epsilon = \mathbf{T}\mathbf{P}^T\beta + \epsilon = \mathbf{T}\mathbf{a} + \epsilon,\\
&\hat{\mathbf{a}} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{y},\\
&\hat{\beta} = \mathbf{P}\hat{\mathbf{a}} = \mathbf{P}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{y}.
\end{aligned}$$

Pokud do našeho modelu chceme zahrnout více, než jednu závisle proměnnou, tj. místo vektoru nyní máme matici $\mathbf{Y}_{n\times q}$, označujeme metodu PLS2. Redukovat pak budeme kromě dimenze matice $\mathbf{X}$ i dimenzi matice závisle proměnných následovně

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}_\mathbf{X}, \quad \mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{E}_\mathbf{Y}.$$

Matice skórů $\mathbf{U}$ a $\mathbf{T}$ i matice zátěží $\mathbf{P}$ a $\mathbf{Q}$ mají obecně $k$ sloupců, kde $k \leq \min(p, q, n)$. Označme dále vektory $\mathbf{u}_j, \mathbf{t}_j, \mathbf{p}_j$ a $\mathbf{q}_j$ $j$-tý sloupec matice $\mathbf{U}, \mathbf{T}, \mathbf{P}$, resp. $\mathbf{Q}$, $j = 1, \ldots, k$. Mezi skóry existuje vnitřní lineární vztah

$$\mathbf{u}_j = d_j\mathbf{t}_j + \mathbf{h}_j,$$

přičemž $\mathbf{h}_j$ je reziduum a $d_j$ regresní parametr. Pokud jsou prvky $\mathbf{h}_j$ malé, můžeme říct, že $x$-skóry $j$-té komponenty dobře predikují $y$-skóry, tím pádem i závisle proměnné.

Cílem PLS2 je maximalizovat kovarianci mezi skóry $\mathbf{X}$ a $\mathbf{Y}$, přičemž v optimalizační úloze obvykle kovarianci mezi dvěma skóry $\mathbf{t}$ a $\mathbf{u}$ odhadujeme pomocí výběrové kovariance ve tvaru $\frac{\mathbf{t}^T\mathbf{u}}{(n-1)}$ s podmínkou na normu vektorů $\|\mathbf{t}\| = \|\mathbf{u}\| = 1$. Je však možné použít i robustní odhad [5].

Problém ještě upravíme s pomocí vztahů $\mathbf{t} = \mathbf{X}\mathbf{w}$ a $\mathbf{u} = \mathbf{Y}\mathbf{c}$. Dostáváme tedy úlohu

$$\operatorname{cov}(\mathbf{X}\mathbf{w}, \mathbf{Y}\mathbf{c}) \to \max \quad \|\mathbf{X}\mathbf{w}\| = \|\mathbf{Y}\mathbf{c}\| = 1. \tag{1.8}$$

Řešením úlohy (1.8) získáme první skóry $\mathbf{t}_1$ a $\mathbf{u}_1$. Abychom vypočítali ty další, musíme do úlohy přidat dodatečné podmínky. Za ty obvykle volíme ortogonalitu skórů $\mathbf{t}_j^T\mathbf{t}_l = 0$, resp. $\mathbf{u}_j^T\mathbf{u}_l = 0$ pro $1 \leq j \leq l \leq k$. Alternativou pak mohou být podmínky na ortogonalitu zátěží.

Jelikož úloha maximalizace nezáleží na konstantě, můžeme problém (1.8) ještě upravit dosazením vztahu pro výběrovou kovarianci uvedeným dříve

$$\mathbf{t}^T\mathbf{u} = (\mathbf{X}\mathbf{w})^T\mathbf{Y}\mathbf{c} = \mathbf{w}^T\mathbf{X}^T\mathbf{Y}\mathbf{c} \to \max \tag{1.9}$$

za stejných podmínek jako v úloze (1.8). Vektory $\mathbf{w}$ a $\mathbf{c}$ můžeme získat s pomocí singulárního rozkladu matice $\mathbf{X}^T\mathbf{Y}$.

SPLS (Sparse Partial Least Squares) rozšiřuje metodu PLS o volbu pouze důležitých proměnných. Motivací je především lepší interpretace vypočtených parametrů, než-li u metody PLS v případě velkého množství nevýznamných proměnných v datech. Při výpočtu skórů algoritmy metody PLS totiž využívají informaci všech proměnných (více v [16]).

Úlohu lze řešit, stejně jako v úloze (1.9), s pomocí singulárního rozkladu (SVD). Dle vlastností SVD můžeme úlohu ekvivalentně formulovat takto

$$\mathbf{w}^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{c} \to \max \quad \|\mathbf{X}\mathbf{w}\| = \|\mathbf{Y}\mathbf{c}\| = 1. \tag{1.10}$$

V dalším textu budeme značit $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X} = \mathbf{M}$ a uvažujeme pro zjednodušení rozklad matic

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}_X, \quad \mathbf{Y} = \mathbf{T}\mathbf{Q}^T + \mathbf{E}_Y, \quad \mathbf{w} = \mathbf{c}.$$

Úlohu SPLS pak s úpravou úlohy na minimalizaci formulujeme dle [16] následovně

$$\min_{\mathbf{w},\mathbf{d}} \left\{ -\kappa \mathbf{w}^T\mathbf{M}\mathbf{w} + (1-\kappa)(\mathbf{d}-\mathbf{w})^T\mathbf{M}(\mathbf{d}-\mathbf{w}) + \lambda_1 \|\mathbf{d}\|_1 + \lambda_2 \|\mathbf{d}\|_2^2 \right\} \tag{1.11}$$

za podmínek $\mathbf{w}^T\mathbf{w} = 1, \|\mathbf{d}\| = 1$, kde $\mathbf{d}$ je pomocný vektor zlepšující vlastnosti úlohy. Dále $\lambda_1 \|\mathbf{d}\|_1$ a $\lambda_2 \|\mathbf{d}\|_2^2$ jsou $L_1$, resp. $L_2$ penalizační členy.

Obdobně jako v úloze (1.8) takto získáme pouze první skóry. Další skóry získáme opět tak, že zavedeme dodatečné podmínky, které se liší dle užitého algoritmu. Dále označíme $K$ maximální počet komponent, které požadujeme, $A$ množinu aktivních indexů (indexy významných proměnných) a $\mathbf{X}_A$ matici $\mathbf{X}$ pouze s proměnnými z množiny $A$. Postupujeme podle tohoto obecného algoritmu

1. Zadáme $\hat{\mathbf{B}}^{PLS} = 0, A = \emptyset, k = 1$.
2. Pokud $k < K$
   - vyřešíme úlohu (1.11), najdeme $\hat{w}$,
   - aktualizujeme množinu $A$ indexy $\{i : \hat{w}_i \neq 0\} \cup \left\{ i : \hat{\mathbf{B}}_i^{PLS} \neq 0 \right\}$,
   - vyřešíme úlohu PLS pro matici $\mathbf{X}_A$ a $k$ komponent,
   - aktualizujeme $\hat{\mathbf{B}}^{PLS}$ odhadem z předchozího kroku,
     $k = k + 1$,
     provedeme deflaci.

Označení $\hat{\mathbf{B}}^{PLS}$ pouze zdůrazňuje, že jsou regresní parametry řešeny s pomocí metody částečných nejmenších čtverců. Úloha (1.11) má čtyři parametry $(\kappa, \lambda_1, \lambda_2, k)$. Parametr $\kappa$ ovlivňuje konvexitu účelové funkce, abychom se vyhnuli nalezení pouze lokálního minima úlohy, a je doporučeno jej volit menší než $\frac{1}{2}$. Parametry $\lambda_1, \lambda_2$ ovlivňují počet proměnných, které budeme v úloze uvažovat. Pokud nastavíme $\lambda_2 = \infty$, závisí úloha pouze na parametru $\lambda_1$ [16].

Vyřešením úlohy algoritmem SPLS tedy budeme schopni zformulovat vztah $\hat{\mathbf{Y}} = \mathbf{X}\hat{\mathbf{B}}$. Pro vyrovnané hodnoty však bude obecně platit $\hat{y}_{i\cdot} \in R^q,\ i = 1, \ldots, n$, zatímco klasifikační třídy jsou celá čísla. Vyrovnané hodnoty proto matematicky zaokrouhlíme na celá čísla.

## 1.6. Křížová validace

Kvalitu modelu můžeme, jak bylo uvedeno dříve, provést na testovacích datech. Tento přístup však má svá úskalí v tom, že rozdělení datové množiny provádíme náhodně. Pokud máme k dispozici jen málo pozorování v jednotlivých třídách, získáme odhad kvality modelu, který se může oproti skutečné hodnotě dosti lišit. Jiná náhodná volba trénovací množiny by pak mohla vést ke značně odlišnému odhadu kvality [1]. Klasifikace pozorování z testovací množiny může být negativně ovlivněna i např. přítomností odlehlých pozorování. Abychom odhad predikčních schopností sestaveného modelu zpřesnili, můžeme použít křížovou validaci. Principem je opět volba části dat, na které budeme model trénovat, a části, na které jej budeme ověřovat, nicméně tentokrát budeme volbu provádět opakovaně.

Jedním z možných přístupů je vybrat pro testování pouze jedno náhodně zvolené pozorování (Leave-One-Out Cross Validation). Za vhodnou volbu je však dle [1, str. 110] považováno vynechání 10% datové množiny (Ten-Fold Cross Validation). V tomto případě rozdělíme datovou množinu náhodně na deset částí. Postupně je pak každá z částí použita jako testovací množina a zbývajících devět pro trénování modelu. Jako odhad přesnosti modelu můžeme použít např. průměr nebo medián dílčích odhadů.

Kromě odhadu predikčních schopností modelu můžeme křížovou validaci použít i v případech, kdy do algoritmu některé metody vstupuje jeden či více parametrů, a my hledáme jeho optimální hodnotu. Tomuto procesu říkáme ladění [2]. Na všech trénovacích množinách pak zkoušíme každou uvažovanou hodnotu parametru a pro finální model vybereme tu, se kterou dosáhneme nejlepší odhadnuté přesnosti.

# Kapitola 2

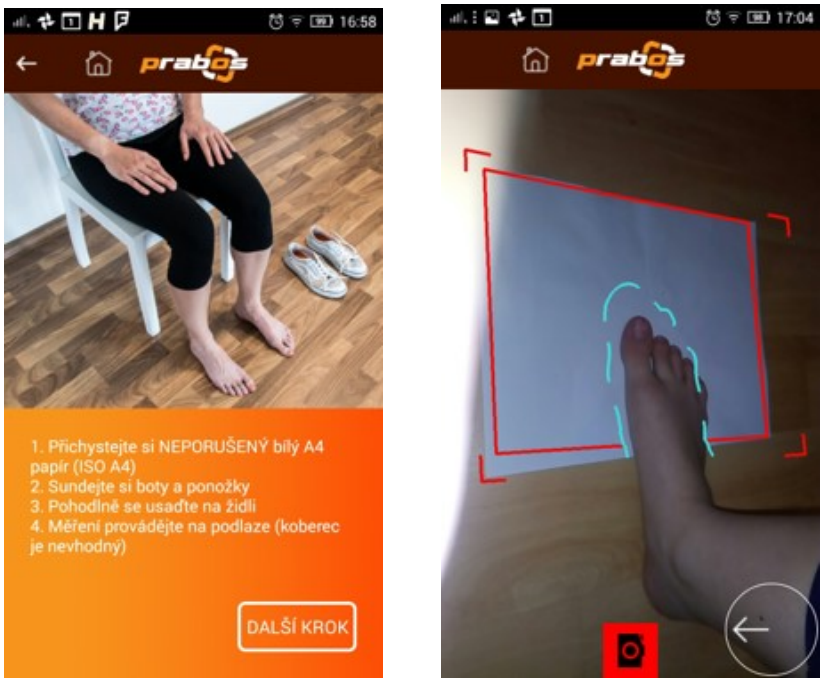
# Data

Ještě než můžeme začít se samotnou výstavbou modelu, je nejprve třeba nasbírat data a připravit je pro další zpracování. Každému člověku budeme chtít během měření změřit obě chodidla s využitím k tomu určené aplikace. Dále budeme zaznamenávat i další potencionálně důležité informace pro co nejpřesnější klasifikaci. Získaná data budeme dále zpracovávat v softwaru R. Toto obnáší mj. vytvoření přehledného datového souboru vhodného pro další práci a odstranění pozorování, která aplikace naměřila nepřesně. S takto zpracovanými daty budeme moci přikročit k trénování a evaluaci klasifikačních modelů.

## 2.1. Aplikace pro měření chodidla a sběr dat

Pokud si vybíráme obuv, kterou si nemůžeme vyzkoušet, většinou její velikost přibližně určujeme z délky chodidla od konce paty po špičku palce, případně dle našich zkušeností. Velikost obuvi ale může být ovlivněna přirozeně i dalšími proměnnými, jako např. výškou nártu nebo šířkou chodidla. Toto jsme při tvorbě modelu chtěli od počátku respektovat. Proto byla při sběru dat nezbytnou pomůckou mobilní aplikace, s jejíž pomocí jsme pro každé chodidlo lehce získali několik hodnot, které mohou být při volbě velikosti boty potenciálně významné.

Aniž bychom zacházeli do technických detailů, pojďme se nejdříve seznámit s tím, jak tato aplikace vlastně funguje. Aplikace existuje ve verzi pro mobilní telefon s operačním systémem Android nebo iOS. Dále je pro její správné fungování potřeba mít na mobilním telefonu fotoaparát a internetové připojení. Po zahájení měření nás aplikace provede celým procesem, přičemž je třeba pořídit tři fotografie každého chodila z různých úhlů dle pokynů na obrazovce. Chodidlo je třeba před fotografováním v souladu s instrukcemi položit na prázdný bílý papír formátu A4. Pokud fotoaparát chodidlo vhodně nezabírá, aplikace nedovolí fotografii udělat. Po skončení měření aplikace vygeneruje 3D model obou chodidel, odhadne několik jejich charakteristik porovnáním s listem papíru pod nohou, vygeneruje kód měření a data odešle na cloudové úložiště. Rozhraní aplikace




Obrázek 2.1: Náhled na proces měření chodidel v aplikaci.

si můžeme prohlédnout na obrázku 2.1.

Aplikace odhadne dvacet tři různých proměnných pro každé chodidlo. Abychom si udělali představu, s čím budeme dále pracovat, popíšeme každou z nich v tabulce 2.1. V té uvedeme anglický název proměnné tak, jak jej udává aplikace, a český popis, co tato proměnná vyjadřuje. V tabulce budeme pracovat s pojmem osa chodidla, což je úsečka procházející nejzazším bodem chodidla na patě a středem spojnice první a páté metatarzální kosti. Těchto kostí se v chodidle nachází pět. První a pátá metatarzální kost se nachází v místech, kde jsou polštářky chodidla. Některé proměnné budeme vysvětlovat s pomocí projekcí chodila na jednotlivé osy v trojrozměrné soustavě souřadnic.

Tabulka 2.1: Proměnné z aplikace na měření chodidel

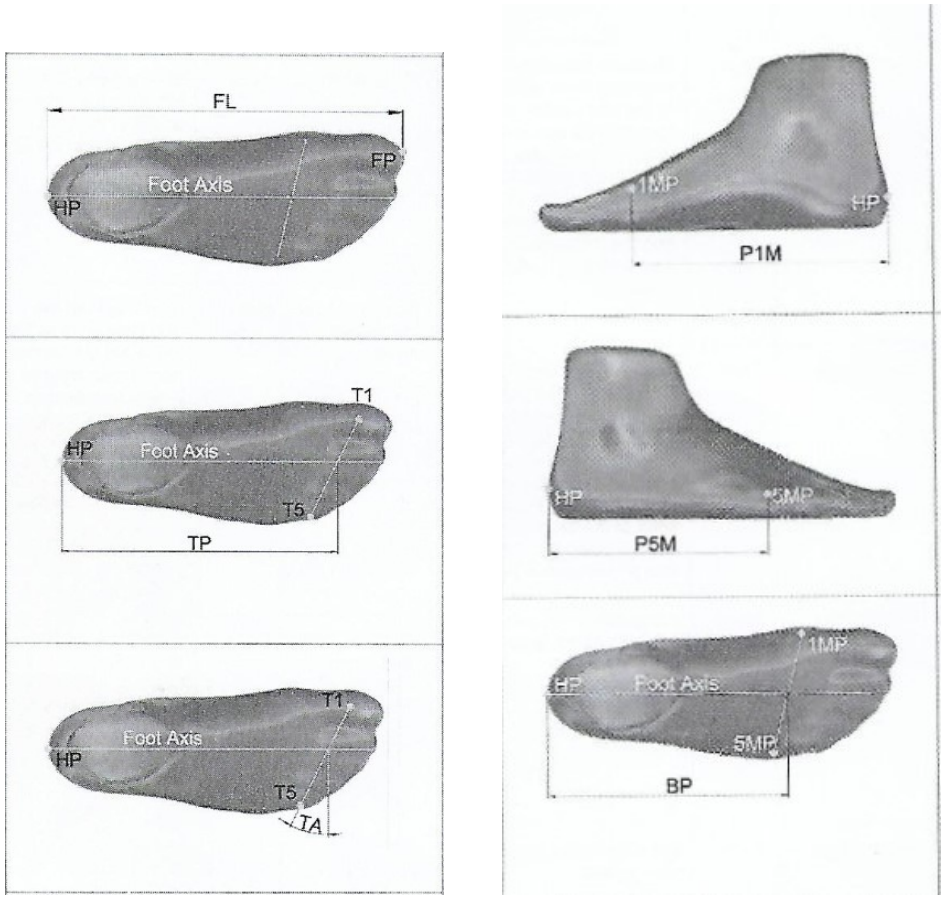
| **Název proměnné** | **Popis proměnné** |
|---|---|
| Foot.Length..mm. | Délka nohy v milimetrech mezi nejvzdálenějšími body chodidla. (V trojrozměrné soustavě souřadnic, kde osa $x$ znázorňuje délku, $y$ šířku a $z$ výšku, tato proměnná vyjadřuje velikost projekce chodidla na osu $x$.) Viz obr. 2.2 vlevo nahoře. |
| Toes.Position..mm. | Velikost projekce na osu $x$ (viz výše) od paty po průsečík osy nohy se spojnicí středů palce a malíčku u nohy. Viz obr. 2.2 vlevo uprostřed. |

Tabulka 2.1: Proměnné z aplikace na měření chodidel

| **Název proměnné** | **Popis proměnné** |
|---|---|
| Toes.Angle..degrees. | Úhel ve stupních, který svírá spojnice středů malíčku a palce s kolmicí na osu chodidla vedenou v místě jejich průsečíku. Viz obr. 2.2 vlevo dole. |
| Toes.Girth..mm. | Obvod konvexního obalu prstů procházejícího přes středy palce a malíčku v milimetrech. |
| Toes.Height..mm. | Výška stejného konvexního obalu jako u předchozí proměnné v milimetrech. |
| Toes.Width..mm. | Šířka konvexního obalu (viz výše) v milimetrech. |
| Position.of.1st.Metatarsal. head..mm. | Velikost projekce chodidla na osu $x$ v milimetrech od paty po pozici první metatarzální kost. Viz obr. 2.2 vpravo nahoře. |
| Position.of.5th.Metatarsal. head..mm. | Velikost projekce chodidla na osu $x$ v milimetrech od paty po pozici pátou metatarzální kost. Viz obr. 2.2 vpravo uprostřed. |
| Ball.Position..mm. | Velikost projekce chodidla na osu $x$ v milimetrech od paty po průsečík osy chodidla se spojnicí první a páté metatarzální kosti. Viz obr. 2.2 vpravo dole. |
| Ball.Angle..degrees. | Úhel ve stupních, který svírá spojnice první a páté metatarzální kosti s kolmicí vedenou na osu procházející středem chodidla v místě průsečíku těchto úseček. |
| Ball.Girth..mm. | Obvod chodidla v milimetrech měřený v místě metatarzálních kostí. |
| Ball.Height..mm. | Velikost projekce chodidla na osu $z$ v milimetrech v oblasti metatarzálních kostí. |
| Medial.Ball.Height..mm. | Výška chodidla v milimetrech v místě první metatarzální kosti. |
| Lateral.Ball.Height..mm. | Výška chodidla v milimetrech v místě páté metatarzální kosti. |
| Ball.Width..mm. | Velikost projekce chodidla v oblasti metatarzálních kostí na osu $y$ . |
| Instep.Girth..mm. | Obvod chodidla v milimetrech měřený přes nárt v místě poloviny délky nohy. |
| Instep.Height..mm. | Výška chodidla v milimetrech měřená v oblasti poloviny délky nohy. |
| Instep.Width..mm. | Největší šířka chodidla v místě poloviny délky nohy v milimetrech. |
| Navicular.Position..mm. | Velikost projekce chodidla na osu $x$ od paty po člunkovitou kost. |

Tabulka 2.1: Proměnné z aplikace na měření chodidel

| **Název proměnné** | **Popis proměnné** |
|---|---|
| Navicular.Height..mm. | Výška chodidla v milimetrech v místě člunkovité kosti. |
| Heel.Width..mm. | Největší šířka chodidla v milimetrech v místě určeném 16 % délky nohy od nohy. |
| Instep.to.heel.Girth..mm. | Obvod chodidla v milimetrech měřený přes patu a místo určené polovinou délky nohy. |




Obrázek 2.2: Obrázek k tabulce 2.1 (od dodavatele aplikace).

Kód měření byl při měření stěžejní údaj. Kromě něj jsme u měřených lidí zaznamenali i zvolenou nejvhodnější velikost a typ obuvi. Každý člověk měl možnost volby mezi dvěma typy bot značky Prabos. Každý z těchto typů je vyráběn na jiném kopytu, což je obuvnická pomůcka, na které je bota šita a tvarována. Boty různých typů o stejné velikosti se tedy mohou mírně lišit v rozměrech. Oba použité modely bot můžeme vidět na obrázku 2.3. Při zkoušení obuvi bylo důležité

pohlížet na obě chodidla nezávisle. Chodidla jsou u každého člověka více či méně odlišná, přirozeně pak může nastat situace, kdy téže osobě vyhovuje na každém chodidlu jiná bota. Dále jsme zaznamenávali i pohlaví, věk měřené osoby a nakonec jsme i dodatečně změřili délku obou chodidel na obuvnickém měřidlu. Jak bylo zmíněno, aplikace pracuje s odhadem rozměrů chodidel. Porovnáním délky chodidla z aplikace s délkou naměřenou ručně můžeme zpětně vyřadit ta pozorování, která byla aplikací odhadnuta nepřesně. U některých osob byl kromě délky měřen ručně i obvod nohy odpovídající proměnné `Ball.Girth..mm.`

Obrázek 2.3: Oba zmíněné typy bot značky Prabos, a.s. Vlevo: Acotango, vpravo: Vagabund [18].

Závěrem je třeba zmínit, že je známo, že aplikace může pracovat nepřesně v případech, kdy se na měřené noze nachází náplast, jsou na ní nalakované nehty, popřípadě ponožka či silonka. Jakoukoliv takovouto skutečnost jsme také zaznamenali pro účely případné další analýzy. Uváděny byly v datovém souboru i některé další poznámky pro marketingové účely Prabos, a. s.

Sběr dat byl limitován nároky, zmíněnými výše, které si aplikace při měření klade pro její správné fungování. Dále jsme se pochopitelně potýkali i s nevolí oslovených lidí k poskytnutí údajů či pořízení fotografií. Kromě dat, která jsme získali my, poskytli několik desítek měření i zástupci samotné společnosti Prabos, a. s., nicméně většinu z nich nebylo možné využít, jelikož nesplňovala požadavky a nároky, které jsme na data kladli. I přes toto všechno se pro další práci nakonec podařilo získat data 168 osob, což znamená 336 různých chodidel.

## 2.2. Automatizace stahování dat z aplikace

Data zaznamená během měření jsme z praktických důvodů přepsali do souboru formátu `.xlsx`, aby s nimi bylo možné dále pracovat jak v tabulkovém procesoru, tak v R. Ukázku tohoto souboru si můžeme prohlédnout na obrázku 2.4.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Kód | Pohlaví | Věk | Délka P | Obvod P | Délka L | Obvod L | Velikost P | Typ P | Velikost L | Typ L | Poznámka | | | |
| 2 | 7ECNNM | Ž | 25 | 250 | 22 | 250 | 22,5 | 39 | A | 39 | A | Silonky + nalakované nehty | | | |
| 3 | 7GRO59 | M | 21 | 252 | 23 | 255 | 24,5 | 40 | V | 40 | V | | | | |
| 4 | DBDU32 | M | 19 | 285 | 26 | 280 | 24,5 | 46 | V | 46 | V | Boty se líbí, seděly nejlépe z nošených | | | |
| 5 | XMTZPS | M | 57 | 265 | 25,5 | 265 | 26,5 | 42 | A | 42 | A | Trochu těsnější | | | |
| 6 | 15HFC6 | Ž | 53 | 245 | 24 | 248 | 24,5 | 39 | A | 39 | A | | | | |
| 7 | BMQJDZ | Ž | 26 | 254 | 23 | 255 | 23 | 40 | V | 40 | V | Silonky | | | |
| 8 | ES9WQN | Ž | 26 | 229 | 18,5 | 230 | 18 | 36 | A | 36 | A | | | | |
| 9 | NXABXF | M | 32 | 248 | 19,5 | 254 | 19,5 | 40 | V | 40 | V | | | | |
| 10 | MWNBDB | Ž | 32 | 253 | 22 | 248 | 22 | 40 | A | 39 | A | | | | |
| 11 | 0Q1TQF | Ž | 20 | 231 | 21 | 232 | 21 | 37 | A | 37 | A | nalakované nehty | | | |
| 12 | MQG3WF | M | 15 | 260 | 23,5 | 258 | 25,3 | 41 | A | 41 | A | | | | |
| 13 | ZVDH5M | Ž | 29 | 244 | 20 | 245 | 20 | 38 | A | 38 | A | | | | |
| 14 | 2BKL0E | Ž | 31 | 233 | 21,5 | 234 | 22 | 38 | A | 38 | A | Nalakované nehty; lepší by bylo 38.5 | | | |
| 15 | 62F885 | M | 30 | 275 | 23 | 277,5 | 24 | 44 | V | 44 | V | | | | |
| 16 | D5Z311 | Ž | 30 | 237 | 21,5 | 236 | 22 | 37 | A | 37 | A | | | | |
| 17 | 5VND3E | Ž | 31 | 250 | 25 | 250 | 24,5 | 40 | V | 40 | V | | | | |
| 18 | A6IUHE | M | 31 | 262 | 24,5 | 262 | 24,5 | 41 | A | 41 | A | | | | |
| 19 | ES9WQN | Ž | 26 | 229 | 18,3 | 230 | 18 | 36 | A | 36 | A | | | | |
| 20 | 4RIDA0 | Ž | 30 | 240 | 19,5 | 244 | 20,5 | 38 | A | 39 | A | | | | |
| 21 | P7E0OM | Ž | 23 | 240 | 20,5 | 242 | 20,5 | 38 | A | 38 | A | | | | |
| 22 | SGA665 | Ž | | 247,5 | 22 | 247,5 | 23 | 39 | A | 39 | A | | | | |
| 23 | ESWQ68 | Ž | 27 | 259 | 23,5 | 261 | 23 | 40 | A | 40 | A | | | | |
| 24 | X9DM2S | Ž | 43 | 249 | 22,5 | 250 | 22,5 | 40 | V | 40 | V | | | | |
| 25 | 3NI1J0 | Ž | 20 | 230 | 22 | 230 | 21 | 38 | V | 38 | V | | | | |

Obrázek 2.4: Ukázka souboru se záznamy z měření.

V souboru však chybí další důležité údaje, rozměry chodidel z aplikace. Ty prozatím zůstaly uložené v cloudovém úložišti chráněném heslem. Než abychom pracně procházeli úložiště a stahovali soubory s rozměry jeden po druhém, využijeme k získání dat R, konkrétně pak balíček `RCurl`. Ten v sobě ukrývá funkce pro práci s webovými adresami. Kromě `RCurl` využijeme i knihovnu `readxl` pro import souborů ve formátech programu Excel v R.

Nejprve do R importujeme soubor zmíněný v úvodu této kapitoly. Poté jej převedeme na objekt třídy `data.frame` a sloupec s kódy uložíme do samostatného vektoru.

```
>mereni <- read_excel('C:/Users/Pohanka/Desktop/
                       Prabos/mereni.xlsx')
>mereni <- as.data.frame(mereni)
>kody <- mereni[,1]
```

Soubory s daty z aplikace jsou uloženy na webovém úložišti a pro přístup k němu je třeba zadat uživatelské jméno a heslo. Každé měření je pak uloženo ve zvláštním adresáři. Část adresy společnou pro všechna měření uložíme do proměnné. Skutečné jméno a heslo v kódu z bezpečnostních důvodů nahradíme slovy `jmeno` a `heslo`.

```
>adresa <- 'https://jmeno:heslo
            @avatar3d.ibv.org:8443/webdav/PRABOS/Tests/'
```

Název každého souboru se pak liší podle toho, jestli se v něm nachází data o levém či pravém chodidle. S využitím předchozích poznatků, tak dvěma `for` cykly vygenerujeme všechny adresy souborů s rozměry levých, resp. pravých chodidel. Ke spojení jednotlivých částí adres v celek využijeme funkci `paste()`.

```
>soubor_leva <- "_left_mes.csv"
>soubor_prava <- "_right_mes.csv"

>adresy_leva <- matrix(rep(0,nrow(kody)),ncol=1)
>adresy_prava <- matrix(rep(0,nrow(kody)),ncol=1)

>for(i in 1:nrow(mereni)){adresy_leva[i,]=
   paste(c(adresa,kody[i,],"/", kody[i,],soubor_leva),collapse="")
   }

>for(i in 1:nrow(mereni)){adresy_prava[i,]=
   paste(c(adresa,kody[i,],"/", kody[i,],soubor_prava),collapse="")
   }
```

Nyní máme připravené dva vektory s webovými adresami, na kterých by se měla nacházet data z provedených měření. Ta nyní s pomocí R všechna importujeme tak, že vytvoříme dvě nulové matice, které budeme v dalších `for` cyklech postupně vyplňovat daty z webu. Jména proměnných si uložíme do vektoru z libovolného staženého souboru.

```
>names=names(read.csv2(adresy_leva[1,],skip=1,header=TRUE,sep=';'))
>names=list(NULL,names)
>mereni_leva=matrix(rep(0,dim(mereni)[1]*23),
                   nrow=dim(mereni)[1],dimnames=names)
>mereni_prava=matrix(rep(0,dim(mereni)[1]*23),
                   nrow=dim(mereni)[1],dimnames=names)
```

Nastat může i situace, že některá adresa z vektoru neexistuje, tj. že se daný soubor na cloudu nenachází. K tomuto může nejčastěji dojít v případě, že byl soubor z úložiště již smazán, případně pokud nebyl vůbec vytvořen. Tzn. že aplikace při měření selhala, obvykle z důvodu nedostatečného internetovému připojení. Tento fakt musíme při stahování dat v cyklu zohlednit s pomocí podmínky `if`, `else` tak, že necháme R nejdříve otestovat, zda daná adresa existuje, a v případě, že ne, vyplní řádek matice hodnotou `NA`.

```
>for(i in 1:length(kody)){if(url.exists(adresy_leva[i,1])==TRUE){
    y=read.csv2(adresy_leva[i,1],skip=2,header=FALSE,sep=";",
      dec=".",stringsAsFactors=FALSE,colClasses=rep("numeric",23))
```

```
    y=as.matrix(y)
    mereni_leva[i,]=y}
  else{mereni_leva[i,]=rep(NA,23)} }
```

Obdobně budeme postupovat i s daty pro pravá chodidla.

Opět si můžeme např. na souboru `mereni_leva`, již uloženém do formátu `.xlsx`, prohlédnout, jak takto získaná data vypadají (obr. 2.5).

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Kód | Foot.Length..mm. | Toes.Position..mm. | Toes.Angle..degrees. | Toes.Girth..mm. | Toes.Height..mm. | Toes.Width..mm. | Position.of.1st.Metatarsal.head..mm. | Position.of.5th.Metatarsal.head..mm. | Ball.Position..mm. |
| 2 | AP1NMG | 226,88 | 190,069 | 23,361 | 209,63 | 22,7934 | 93,426 | 167,737 | 147,825 | 157,295 |
| 3 | AQ6JX6 | 261,162 | 218,628 | 24,774 | 236,194 | 25,4584 | 105,978 | 192,701 | 170,094 | 180,699 |
| 4 | XNVFCE | 248,596 | 206,432 | 19,6931 | 231,343 | 28,7366 | 102,371 | 182,571 | 164,912 | 173,473 |
| 5 | ETF0MC | 233,831 | 199,394 | 19,247 | 215,916 | 32,1005 | 92,9085 | 173,082 | 155,939 | 164,885 |
| 6 | ZUMYZA | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI |
| 7 | X5U9MK | 281,54 | 239,635 | 26,3536 | 244,91 | 29,1819 | 109,13 | 210,926 | 187,488 | 198,749 |
| 8 | 4EF5XY | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI |
| 9 | A8KUND | 261,461 | 219,696 | 29,2276 | 222,242 | 31,1237 | 97,8337 | 195,126 | 168,565 | 181,492 |
| 10 | JRL9UT | 255,605 | 213,807 | 26,559 | 213,75 | 23,5458 | 96,4244 | 187,142 | 163,313 | 174,629 |
| 11 | JDDLEP | 276,579 | 236,885 | 20,7252 | 245,423 | 33,0102 | 107,713 | 204,861 | 187,181 | 195,919 |
| 12 | XV9JU8 | 221,098 | 183,793 | 25,3049 | 197,207 | 27,9828 | 86,3008 | 169,037 | 149,085 | 158,521 |
| 13 | UTIX2R | 257,346 | 221,724 | 17,8912 | 238,504 | 34,0767 | 103,267 | 194,972 | 179,165 | 187,365 |
| 14 | KHHW0L | 253,015 | 211,207 | 24,1296 | 221 | 23,8675 | 99,3009 | 186,948 | 165,993 | 176,047 |
| 15 | T87BXL | 243,382 | 203,411 | 22,5849 | 214,635 | 22,5056 | 96,4059 | 175,913 | 156,838 | 166,059 |
| 16 | WNHVXN | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI |
| 17 | A9CMP7 | 235,63 | 195,303 | 25,0435 | 216,291 | 24,5418 | 97,0583 | 171,284 | 151,614 | 160,979 |
| 18 | LBJCBB | 240,787 | 200,492 | 26,8365 | 220,867 | 26,8238 | 97,3592 | 177,981 | 154,003 | 165,585 |
| 19 | V7VZIN | 235,619 | 197,242 | 25,5148 | 207,795 | 21,9883 | 93,465 | 173,369 | 153,144 | 162,728 |
| 20 | MFPMCR | 273,305 | 227,376 | 25,4055 | 233,009 | 28,9633 | 103,535 | 205,209 | 178,15 | 191,463 |
| 21 | JSLAPA | 283,785 | 240,15 | 26,4513 | 229,983 | 24,7475 | 103,113 | 208,457 | 183,17 | 195,315 |
| 22 | PE1NNU | 237,329 | 199,288 | 24,9898 | 210,817 | 22,1572 | 94,205 | 176,226 | 156,845 | 166,002 |
| 23 | OAFU39 | 225,396 | 190,636 | 22,7824 | 190,64 | 22,0808 | 85,3276 | 166,435 | 146,987 | 156,343 |
| 24 | YXBUXE | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI | #NENÍ_K_DISPOZICI |
| 25 | TVMRBP | 237,935 | 198,402 | 24,2013 | 213,996 | 24,6207 | 95,9139 | 174,208 | 157,019 | 165,105 |
| 26 | 3HX59U | 235,078 | 198,678 | 23,8698 | 190,489 | 23,3045 | 85,3774 | 167,165 | 152,489 | 159,714 |
| 27 | 0B87OI | 238,778 | 199,823 | 23,4971 | 216,038 | 27,0796 | 95,9898 | 176,635 | 156,61 | 166,285 |
| 28 | MV56B6 | 261,119 | 220,239 | 33,2218 | 211,038 | 25,273 | 95,3466 | 190,724 | 165,382 | 176,61 |
| 29 | JCH6RE | 232,365 | 194,468 | 24,187 | 215,014 | 28,5758 | 94,4783 | 174,875 | 152,757 | 163,588 |

Obrázek 2.5: Ukázka ze souboru s daty z aplikace (`mereni_leva`).

## 2.3. Další zpracování dat před tvorbou modelů

Po předchozích krocích máme nyní tři soubory s daty – záznamy sepsané během měření, hodnoty z aplikace pro levá chodidla a hodnoty z aplikace pro pravá chodidla. Z těchto tří souborů budeme chtít vytvořit jediný, se kterým budeme pracovat dále.

Data z aplikace spojíme do jediného objektu tak, aby bylo v každém řádku jedno chodidlo. K tomuto připojíme vybrané proměnné, které jsme během měření zaznamenali, jako třeba pohlaví, věk nebo ideální velikost boty. Velikost v tomto kroku zároveň spojíme s typem boty do jediné proměnné, která bude použita jako klasifikační. Jakmile budeme s tímto hotovi, ihned vytvoříme dodatečnou proměnnou, a to rozdíl mezi délkou nohy naměřenou měřidlem a délkou nohy odhadnutou aplikací. Tato proměnná bude později důležitá pro odhalení nepřesných měření z aplikace. V následujícím kódu používáme funkce z knihovny `dplyr` [19].

```
>data_raw=c(mereni$Age,mereni$Age)
>data_raw = data_raw %>%
    as_tibble() %>%
    mutate(Sex=c(mereni$Sex,mereni$Sex),
      Size=factor(paste(c(mereni$Size_L,mereni$Size_P),
```

```
    c(mereni$Type_L,mereni$Type_P))))
>data_raw=cbind(data_raw,rbind(mereni_leva,mereni_prava)[,-1])
>data_raw=data_raw %>%
   as_tibble() %>%
   mutate(rozdil=data_raw$Foot.Length..mm.-c(mereni$Length_L,
 mereni$Length_P))
```

S tímto soubor se teď můžeme blíže seznámit s pomocí funkce `summary()` (obr. 2.6).

```
> summary(data_raw)
      Age        Sex          Size     Foot.Length..mm. Toes.Position..mm. Toes.Angle..degrees.
 Min.   :14.00   M:138   38 A   : 58   Min.   :195.5    Min.   :157.0      Min.   : 14.39
 1st Qu.:23.00   Ž:198   39 A   : 43   1st Qu.:240.6    1st Qu.:201.7      1st Qu.: 21.31
 Median :32.00           40 A   : 26   Median :251.7    Median :211.3      Median : 23.02
 Mean   :35.14           37 A   : 24   Mean   :253.6    Mean   :213.3      Mean   : 23.57
 3rd Qu.:46.75           41 A   : 21   3rd Qu.:267.5    3rd Qu.:225.6      3rd Qu.: 24.89
 Max.   :78.00           42 A   : 21   Max.   :313.7    Max.   :264.8      Max.   :148.09
 NA's   :42              (Other):143   NA's   :50       NA's   :50         NA's   :50
 Toes.Girth..mm. Toes.Height..mm. Toes.Width..mm.  Position.of.1st.Metatarsal.head..mm.
 Min.   :162.0   Min.   :12.45    Min.   : 74.96   Min.   :140.2
 1st Qu.:214.8   1st Qu.:24.30    1st Qu.: 94.66   1st Qu.:176.6
 Median :226.2   Median :26.89    Median :100.80   Median :184.4
 Mean   :227.9   Mean   :27.61    Mean   :101.35   Mean   :185.6
 3rd Qu.:240.2   3rd Qu.:30.07    3rd Qu.:107.22   3rd Qu.:196.1
 Max.   :279.3   Max.   :75.30    Max.   :125.13   Max.   :233.7
 NA's   :50      NA's   :50       NA's   :50       NA's   :50
 Position.of.5th.Metatarsal.head..mm. Ball.Position..mm. Ball.Angle..degrees. Ball.Girth..mm.
 Min.   :117.8                        Min.   :135.2      Min.   :  0.00694    Min.   :160.1
 1st Qu.:157.6                        1st Qu.:166.7      1st Qu.: 10.66962    1st Qu.:230.2
 Median :164.7                        Median :173.8      Median : 13.07980    Median :244.0
 Mean   :166.4                        Mean   :175.7      Mean   : 12.83130    Mean   :245.6
 3rd Qu.:176.0                        3rd Qu.:185.6      3rd Qu.: 14.54637    3rd Qu.:259.5
 Max.   :207.4                        Max.   :220.4      Max.   :149.59800    Max.   :348.4
 NA's   :50                           NA's   :50         NA's   :50           NA's   :50
 Ball.Height..mm. Medial.Ball.Height..mm. Lateral.Ball.Height..mm. Ball.Width..mm.
 Min.   :26.27    Min.   :14.88           Min.   : 9.60            Min.   : 66.80
 1st Qu.:39.24    1st Qu.:27.01           1st Qu.:18.63            1st Qu.: 95.07
 Median :41.78    Median :29.93           Median :20.15            Median :101.08
 Mean   :42.71    Mean   :30.13           Mean   :20.25            Mean   :101.55
 3rd Qu.:45.97    3rd Qu.:32.67           3rd Qu.:21.69            3rd Qu.:106.85
 Max.   :78.25    Max.   :68.42           Max.   :35.81            Max.   :144.59
 NA's   :50       NA's   :50              NA's   :50               NA's   :50
Instep.Girth..mm. Instep.Height..mm. Instep.Width..mm. Navicular.Position..mm.
Min.   :165.0     Min.   :43.09      Min.   : 60.91    Min.   : 73.59
1st Qu.:234.0     1st Qu.:64.78      1st Qu.: 81.42    1st Qu.: 90.55
Median :248.3     Median :69.04      Median : 87.21    Median : 95.34
Mean   :249.5     Mean   :70.14      Mean   : 87.43    Mean   : 96.30
3rd Qu.:266.3     3rd Qu.:75.20      3rd Qu.: 92.75    3rd Qu.:102.23
Max.   :316.1     Max.   :96.35      Max.   :113.97    Max.   :122.01
NA's   :50        NA's   :50         NA's   :50        NA's   :50
Navicular.Height..mm. Heel.Width..mm. Instep.to.heel.Girth..mm.     rozdil
Min.   :23.88         Min.   :35.82   Min.   :163.2              Min.   :-69.4440
1st Qu.:38.69         1st Qu.:58.55   1st Qu.:326.0              1st Qu.: -5.7270
Median :42.69         Median :63.26   Median :343.1              Median : -2.2300
Mean   :43.28         Mean   :63.32   Mean   :345.8              Mean   : -5.0153
3rd Qu.:46.58         3rd Qu.:67.53   3rd Qu.:366.2              3rd Qu.:  0.1495
Max.   :70.08         Max.   :86.41   Max.   :500.5              Max.   : 18.5330
NA's   :50            NA's   :50      NA's   :50                 NA's   :61
```

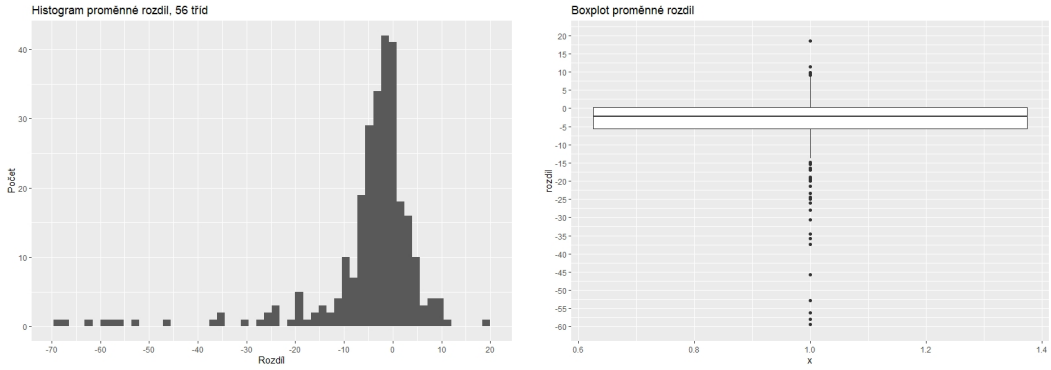
Obrázek 2.6: Náhled na souhrn neupraveného datového souboru.

Jakmile si tento souhrn prohlédneme, mělo by nás zaujmout především to, že se v datech nachází 50 pozorování, pro která data z aplikace chybí. U dalších

11 nemáme k dispozici hodnotu rozdílu délek nohy, jelikož pro ně nemáme uvedenou kontrolní délku chodidla. V případech, kdy se v datech nachází relativně vysoký počet chybějících pozorování, je doporučováno chybějící hodnoty doplnit s pomocí některé z imputačních metod [1, str. 6]. Nicméně musíme vzít v úvahu fakt, že pro zmíněná pozorování nemáme k dispozici hodnotu ani jedné proměnné z aplikace. Ať už bychom využili jakoukoliv z metod, prakticky bychom jen generovali nová pozorování bez jakéhokoliv podkladu. Tento přístup by nám zřejmě nepřinesl jakoukoliv relevantní informaci, proto řádky s `NA` hodnotami u proměnných z aplikace odstraníme. K tomu použijeme funkci `drop_na()` z knihovny `tidyr` [20]. Jako argument funkce můžeme použít libovolnou proměnnou z aplikace, jelikož u těchto pozorování nemáme hodnotu žádné z nich.

```
>data_raw=data_raw %>%
  drop_na(Foot.Length..mm.)
```

Po odstranění pozorování bez hodnot z aplikace budeme chtít odhalit chodidla, která aplikace změřila nepřesně. Jedním z možných přístupů by mohla být některá z metod mnohorozměrné detekce odlehlých pozorování. Nicméně je třeba si uvědomit, že při měření byl kladen důraz na to, aby si lidé zvolili velikost, ve které se cítí nejlépe. V této fázi navíc nedokážeme z proměnných určit ty, které mají na volbu velikosti skutečně vliv. Vzhledem k tomuto nemáme záruku, že tyto metody neoznačí za odlehlá i ta pozorování, která byla změřena přesně, pouze se však liší od ostatních v rámci třídy, navíc třeba jen dle irelevantních proměnných. Nejjednodušší se tak zdá použít některou z metod jednorozměrné detekce odlehlých hodnot na námi vytvořenou proměnnou `rozdil`. Ta vznikla jako rozdíl mezi délkou nohy odhadnutou aplikací a délkou nohy, kterou jsme během měření zjistili z ševcovského měřidla. Nejprve si vykresleme histogram a krabicový graf hodnot zmíněné proměnné, abychom zjistili, jestli se v datech nějaké odlehlé hodnoty vyskytují (obr. 2.7).




Obrázek 2.7: Histogram a krabicový graf proměnné `rozdil`.

Ještě než budeme vyvozovat z grafů závěry, je třeba si uvědomit, že odchylka mezi hodnotou z aplikace a hodnotou z měřidla je přirozená. Od aplikace nemůžeme očekávat zcela přesné hodnoty a ani s měřidlem nebylo možné určit délku naprosto precizně. V histogramu na obrázku 2.7 si můžeme všimnout, že se střední hodnota odchylky v měřeních nachází blízko nule, přičemž většina odchylek je koncentrována právě v její blízkosti. Z obou grafů je nicméně patrné, že se v našem souboru nachází i několik pozorování s velkou hodnotou odchylky, a těm bude třeba dále věnovat pozornost.

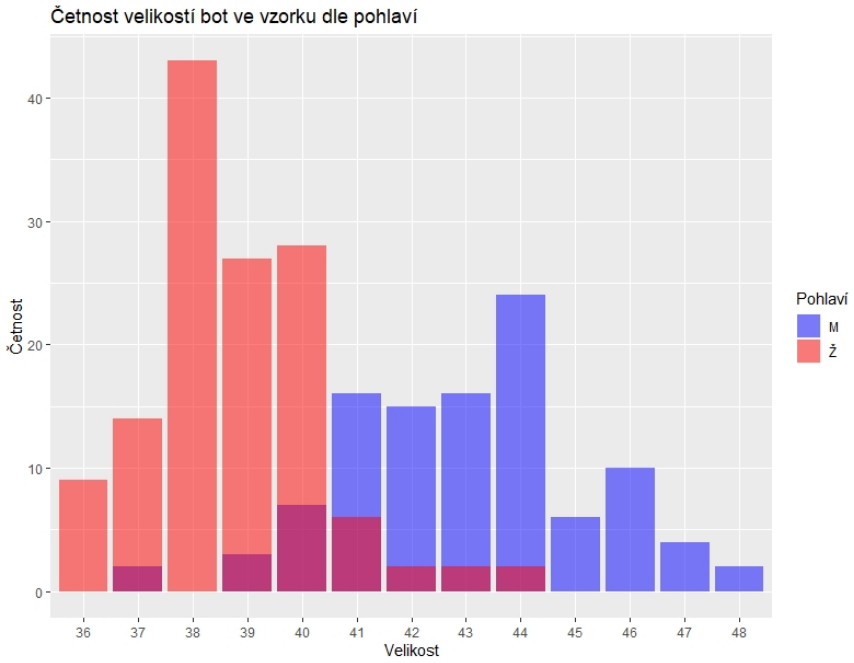
Metod detekce odlehlých pozorování existuje několik. My v práci uvedeme jednu z často užívaných, založenou na mezikvartilovém rozpětí (IQR). U této metody je jako odlehlá brána hodnota, která je od mediánu proměnné vzdálená více než 1,5 násobek IQR. Připomeňme nyní, že se ve zkoumané proměnné stále nachází 11 chybějících pozorování. Tato si označíme a budeme k nim přistupovat individuálně. Dále se musíme rozhodnout, jaký přístup k chybně naměřeným pozorováním zvolíme. Je zřejmé, že jejich ponechání v souboru by negativně ovlivnilo klasifikační modely. Imputaci hodnot jsme z důvodů uvedených dříve zamítli. Nezbývá, než tato pozorování rovněž ze souboru vyřadit.

```
>rozdil_na=data_raw %>%
        filter(is.na(rozdil)==TRUE) %>%
        mutate(is_outlier=NA)

>a=median(data_raw$rozdil,na.rm=TRUE)

>data_raw=data_raw %>%
        mutate(is_outlier=((rozdil>a+1.5*IQR(rozdil,na.rm=TRUE))|
          (rozdil<a-1.5*IQR(rozdil,na.rm=TRUE))))

>data_raw=data_raw %>%
            filter(is_outlier==FALSE)

>data_raw=rbind(data_raw,rozdil_na)
```

Po odstranění odlehlých pozorování se může stát, že námi zvolená metoda v datech odhalí další pozorování, která je třeba vyřadit. Postup tedy opakujeme, dokud žádná další takováto pozorování neobjevíme. K 11 zbývajícím pozorováním, pro která nemáme hodnotu rozdílu, budeme přistupovat individuálně. V datech ponecháme ta, jejichž délka nohy spadá do intervalu délek nohou ostatních pozorování o stejné velikosti a typu boty.

## 2.4. Pohled na data po zpracování

Po zmíněných provedených úpravách souboru budeme nadále pracovat již pouze s 238 pozorováními. Ještě než však přejdeme k samotné klasifikaci, je vhodné se s daty lépe seznámit. Nejdříve nás bude zajímat, zastoupení velikostí v našem vzorku. Četnost jednotlivých velikostí s sebou totiž nese důležitou informaci, a sice apriorní pravděpodobnost, s jakou bude nově příchozí pozorování zařazeno do jednotlivých tříd. Při sběru dat nebyl kladen důraz na reprezentativnost vzorku. Účastníci měření byli oslovováni náhodně s důrazem na kvantitu bez jakékoliv další selekce. Nemůžeme tudíž automaticky předpokládat, že lze rozdělení četností velikostí v našich datech zobecnit na celou populaci. Pokud by se tedy rozdělení u našeho vzorku významně lišilo od toho pro celou populaci, museli bychom přistoupit ke korekci. Z dostupných zdrojů, např. [21], můžeme usoudit, že velikosti obuvi bude veličina, kterou lze přibližně aproximovat normálním rozdělením za předpokladu, že populaci rozdělíme dle pohlaví. Tato rozdělení se pak budou lišit střední hodnotou.



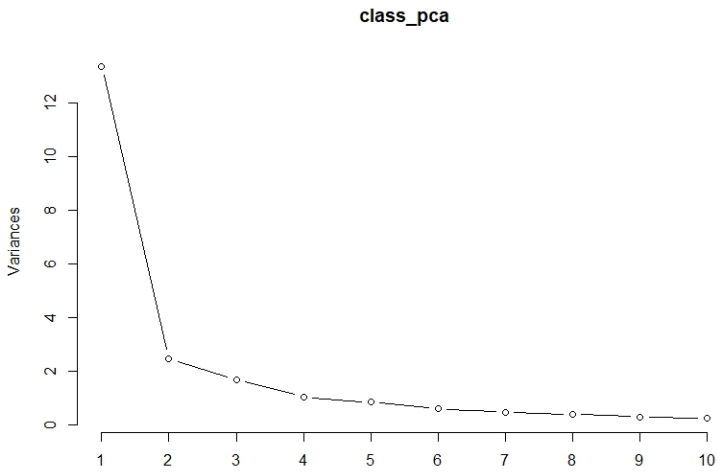

Obrázek 2.8: Četnost velikostí bot v datech.

Dle grafu na obr. 2.8 nemůžeme rozdělení velikostí v našem souboru považovat za normální, nicméně tvarem se neliší významně. Změnou apriorního rozdělení pravděpodobnosti v modelech, bychom zřejmě nedosáhli výrazného zlepšení úspěšnosti klasifikace. Nicméně budeme se mít na pozoru, zda nebudou mít některé metody tendenci přiřazovat menším nohám velikost č. 38.

Dále bychom data z aplikace chtěli vizualizovat. Konkrétně by nás mohlo

zajímat, zda pozorování tvoří shluky dle jejich velikostí bot. Více jak tři proměnné samozřejmě nelze jednoduše zobrazit tak, aby graf zůstal snadno interpretovatelný. Použijeme tedy Analýzu hlavních komponent (PCA). Soubor očištěný od chybějících a odlehlých pozorování máme uložený v objektu `data_clean`. Jelikož je PCA při hledání nových proměnných citlivá na měřítko jednotlivých proměnných, soubor necháme před analýzou škálovat. Zároveň si vykreslíme i screeplot.

```
>class_pca=prcomp(data_clean[,5:26],scale=TRUE)
>screeplot(class_pca, type='lines' )
```

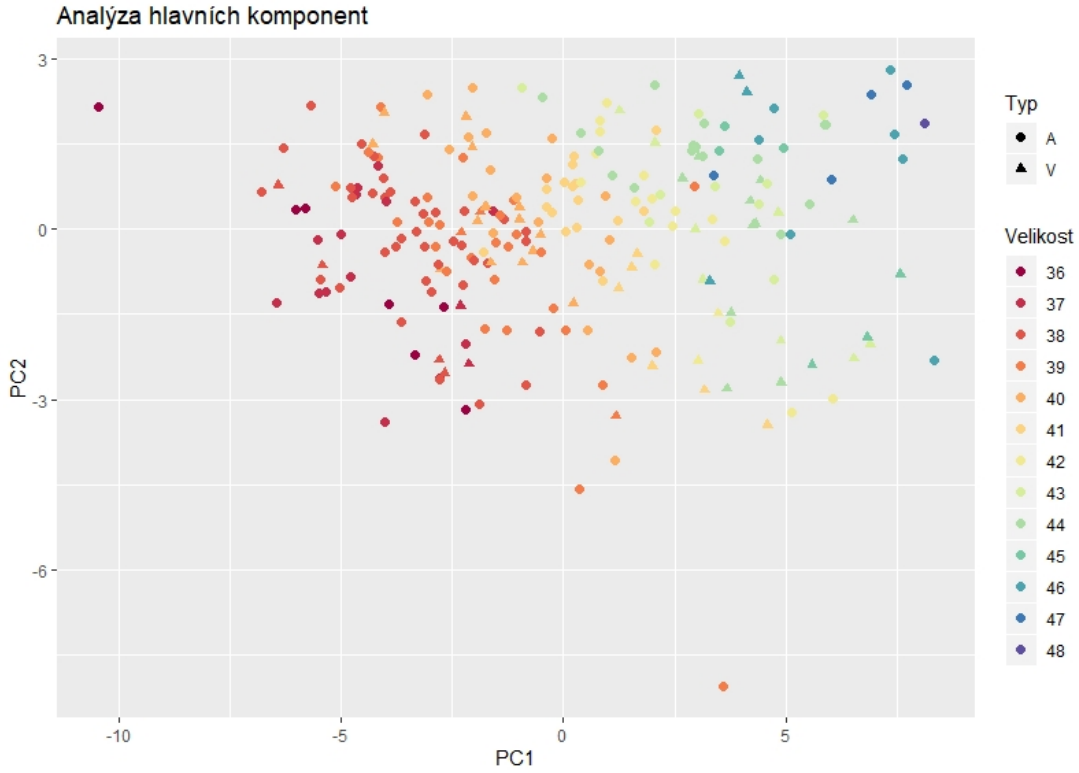



Obrázek 2.9: Screeplot k provedené PCA.

Dle pravidla bodu zlomu by mělo pro interpretaci stačit použít první dvě komponenty (viz obr. 2.9), které vysvětlují téměř 72 % variability dat. Proměnnou `Size`, ve které je uložený i typ boty, rozdělíme na proměnnou `Size` a `Type`. Běžně užívané barevné palety v R obsahují pro diskrétní proměnné nejvýše 12 barev. V našich datech se však vyskytuje 13 různých velikostí, takže budeme muset některou z palet nejprve rozšířit na 13 barev. Poté už jen uložíme všechny proměnné, které chceme vykreslit, do jediného objektu třídy `data.frame`.

```
>data_clean_separated=separate(data_clean,
          'Size',c('Size','Type'),sep=' ')
>data_clean_separated$Size=as.factor(data_clean_separated$Size)
>data_clean_separated$Type=as.factor(data_clean_separated$Type))

>my_palette=colorRampPalette(brewer.pal(10,'Spectral'))(13)
```

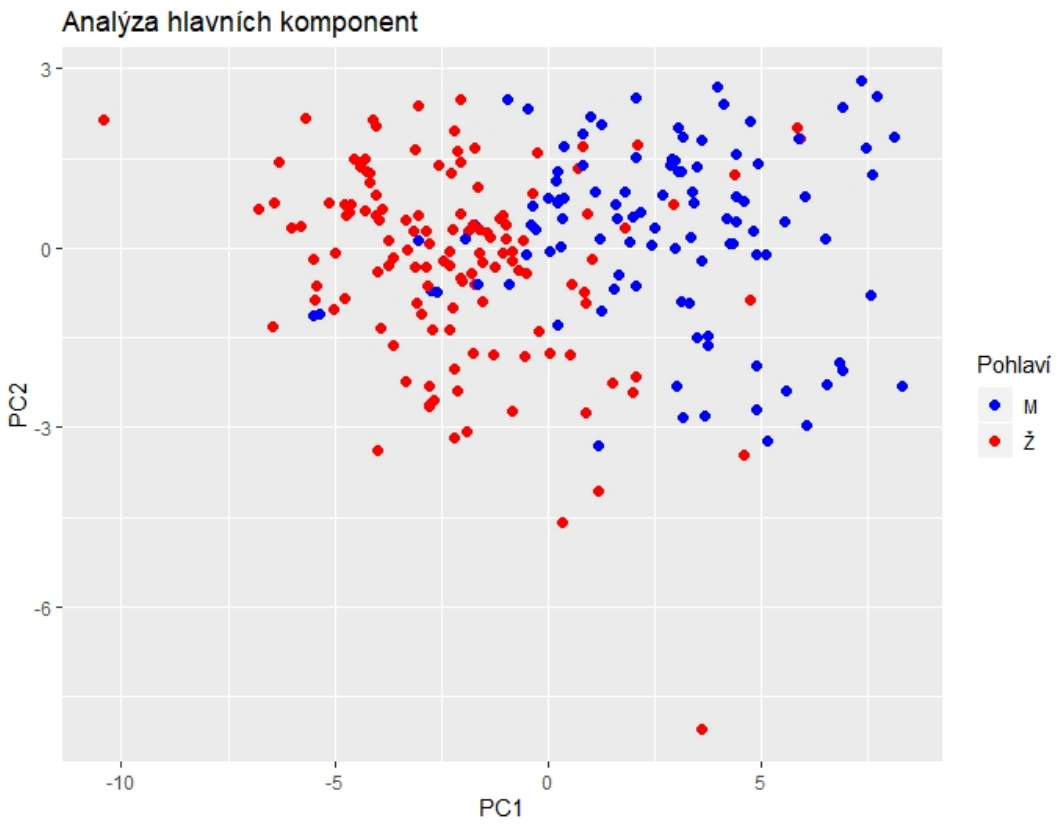
```
>class_pca_2=data.frame(pc1=class_pca$x[,1],pc2=class_pca$x[,2],
          Size=as.factor(data_clean_separated$Size),
          Type=as.factor(data_clean_separated$Type))

>ggplot(class_pca_2,aes(x=PC1,y=PC2,col=Size,shape=Type)) +
      geom_point() +
      scale_colour_manual(values = my_palette) +
      labs(title="Analýza hlavních komponent",
           col="Velikost",shape="Typ")
```




Obrázek 2.10: PCA, graf prvních dvou proměnných s barevně odlišenými velikostmi.

Na základě dvou komponent (viz obr. 2.10) nemůžeme říct, že by se pozorování jednoznačně shlukovala dle velikosti boty. Shluky se vzájemně překrývají, což už nyní může naznačovat problémy při klasifikaci. Graf vykreslíme ještě jednou, ale tentokrát pozorování barveně rozdělíme dle pohlaví, abychom ilustrovali důležitost této proměnné při klasifikaci.

V grafu na obr. 2.11 vidíme i přes jistý překryv zřetelně oddělená pozorování



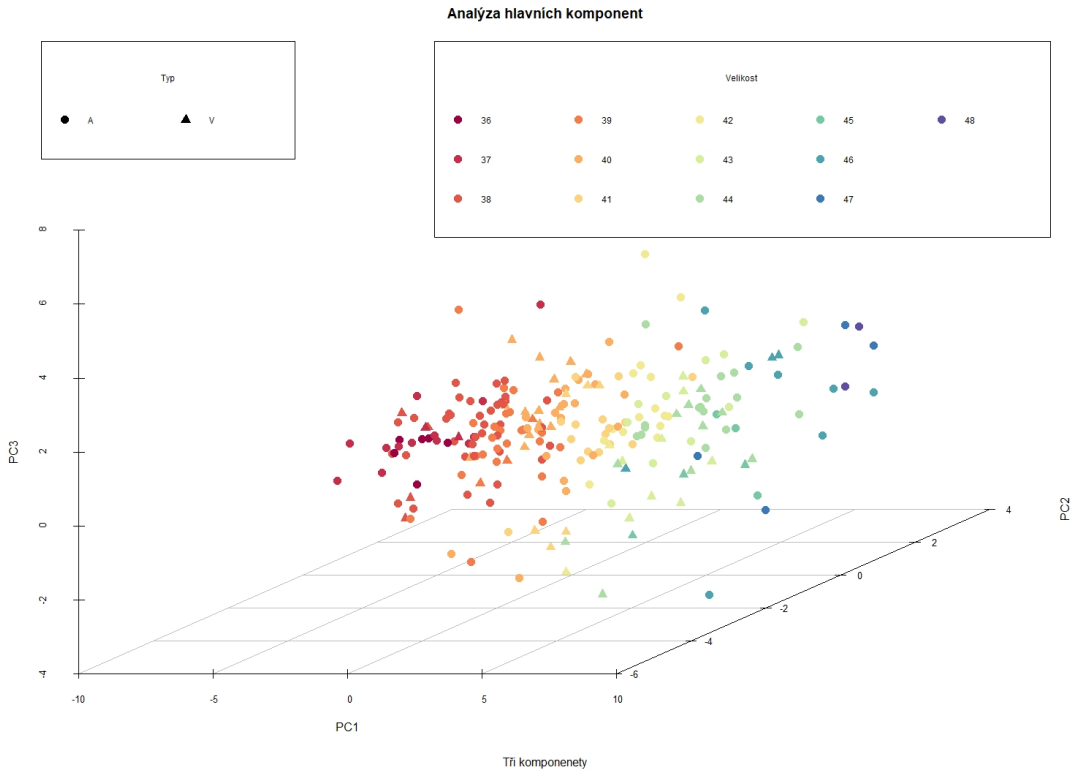

Obrázek 2.11: PCA, graf prvních dvou proměnných s barevně odlišeným pohlavím.

dle pohlaví, což nás utvrzuje v domněnce, že je tato proměnná pro nadcházející klasifikační úlohu důležitá.

Na závěr zkusíme, zda přidáním třetí komponenty, čímž vysvětlíme již skoro 80 % variablity, docílíme lepšího shlukování pozorování dle velikostí. Pro trojrozměrný graf využijeme knihovnu `scatterplot3d` [22]. Pro alespoň trochu lepší přehlednost odstraníme jedno pozorování, které se v grafu nachází daleko od hlavní skupiny všech ostatních.

```
>class_pca_3=data.frame(PC1=class_pca$x[,1],
        PC2=class_pca$x[,2],PC3=class_pca$x[,3],
        Size=as.factor(data_clean_separated$Size),
        Type=data_clean_separated$Type)

>pca3d = scatterplot3d(class_pca_3$PC1, class_pca_3$PC2,
     class_pca_3$PC3,xlab='PC1', ylab='PC2', zlab='PC3',
     color=my_palette[as.numeric(class_pca_3$Size)],
     box=FALSE,pch=(as.numeric(class_pca_3$Type)+15),
```

```
      main='Analýza hlavních komponent',sub='Tři komponenety',
      cex.symbols=1.5,xlim=c(-8,9),ylim=c(-6,3),zlim=c(-4,8))

>legend('topleft', legend = levels(class_pca_3$Type),
      pch = as.numeric(class_pca_3$Type)+15, text.width=0.55,
      horiz = TRUE, xpd = TRUE, title='Typ',cex = 0.75,pt.cex=1.5)

>legend('topright', legend = levels(class_pca_3$Size),
      col= my_palette, ncol=5,pch = 16,xpd = TRUE, text.width=0.55,
      title='Velikost',cex = 0.75,pt.cex=1.5)
```




Obrázek 2.12: PCA, graf prvních třech proměnných.

Ani po přidání třetí komponenty nelze v grafu pozorovat zřetelné shlukování. Zejména u těch skupin pozorování, které se vzájemně liší o jednu velikost, dochází k prolínání a v grafu nelze shluky identifikovat jednoznačně. Stále jsme však pracovali jen se zjednodušenou strukturou dat. V momentě, kdy budeme pracovat se všemi proměnnými, stále můžeme dosáhnout uspokojivých výsledků.

# Kapitola 3

# Tvorba a srovnání modelů

Nyní můžeme přejít k samotnému tréninku predikčních modelů. V R existují desítky balíčků, které umožňují pracovat s některými regresními či klasifikačními metodami. Každý z těchto balíčků přidává do R funkce, které se liší nejen názvem, ale i jejich argumenty a syntaxí, ačkoliv mají ve výsledku velmi podobné výstupy. Ve chvíli, kdy se chystáme pracovat s několika různými metodami, se náš skript může snadno stát nepřehledným. Nemluvě o čase stráveném vyhledáváním správných funkcí a dokumentace. Proto budeme v této kapitole pracovat téměř výhradně s balíčkem `caret` (**C**lassification **A**nd **RE**gression **T**raining), který umožňuje snadné použití klasifikační a regresních metod z různých knihoven. Navíc obsahuje i další funkce pro přípravu dat a jejich vizualizaci před samotnou tvorbou modelu. Teoreticky by tak tedy bylo možné tento balíček použít pro výstavbu celého predikčního modelu v R. Zájemce o důkladnější seznámení s knihovnou si dovolíme odkázat na manuál [2], v práci pak uvedeme jen informace důležité pro naši úlohu.

V datech nejprve provedeme ještě několik málo posledních úprav tentokrát již s využitím knihovny `caret`. V souladu s kapitolou 1 je třeba buď to data náhodně rozdělit na trénovací a testovací množinu, nebo využít křížovou validaci. V námi zvolené knihovně je křížová validace součástí tréninku modelu automaticky. Kromě měření predikčních schopností metody algoritmus zároveň během validace hledá nejvhodnější hodnotu ladicích parametrů, pokud nezvolíme jinak. To znamená, že není nutné rozdělení dat provádět manuálně. Nicméně my data přeci jen před trénováním modelu rozdělíme. Můžeme tak získat při predikci další potencionálně důležité informace.

Před dělením dat se ještě budeme věnovat proměnné pohlaví. V kapitole 2.4 jsme graficky ověřili, že pohlaví je pro daný problém důležitá proměnná. Nyní je potřeba se rozhodnout, jakým způsobem vliv této proměnné při tvorbě modelu zohledníme. V ideálním případě bychom vytvořili dva oddělené modely dle pohlaví. Pro tento přístup však nemáme u jednotlivých pohlaví dostatek pozorování pro všechny velikosti. Data proto ponecháme pohromadě a pohlaví do modelu zahrneme jako faktorovou proměnnou.

```
>levels(data_clean$Sex)=c(0,1)

>set.seed(1)
>train_data=createDataPartition(y = data_clean$Size,p=.8,list=FALSE)

>training=data_clean[train_data,3:26]
>testing=data_clean[-train_data,3:26]
```

Výše uvedenými příkazy jsme náš datový soubor náhodně rozdělili v poměru 80 % : 20 %. V případě, že bychom chtěli provést křížovou validaci ručně, můžeme využít funkci `createFolds`. Tento postup by však byl časově velmi náročný, křížovou validaci tedy necháme provést algoritmy automaticky a testovací soubor využijeme jen, abychom ilustrovali chování jednotlivých metod. Porovnáním relativních frekvencí jednotlivých klasifikačních tříd v tréninkové a testovací množině neodhalíme významné odchylky, což je rovněž žádoucí.

Pro trénink modelů slouží funkce `train()`. Pokud nezvolíme jinak, tato funkce testuje efekt různých hodnot ladicích parametrů na přesnost klasifikace, resp. regrese. Sama pak zvolí optimální hodnotu těchto parametrů na základě zvoleného kritéria. Pro klasifikaci můžeme volit mezi přesností a Cohenovou Kappou, pro regresi pak průměrnou absolutní chybu (MAE), koeficient determinace ($R^2$) nebo odmocninu ze střední kvadratické chyby (RMSE). V neposlední řadě funkce `train()` na dostupných datech odhadne prediktivní schopnost modelu. Algoritmus, podle kterého funkce pracuje, uvedeme níže [2].

1. Zadej množiny hodnot parametrů pro ladění.
2. **for** *Každou množinu parametru*
   - **for** *Každou iteraci dělení dat*
     - Urči vzorek dat pro testování.
     - [Volitelné] Předzpracuj data.
     - Natrénuj model na zbývajících datech.
     - Predikuj závisle proměnnou testovacích dat.
   - **end**
   - Spočítej průměrnou výkonnost modelu na testovacích vzorcích.
3. **end**
4. Urči optimální hodnoty parametrů.
5. Vytvoř model na celé vstupní množině dat s optimálními hodnotami parametrů.

Jako jeden z argumentů funkce `train()` můžeme zadat objekt třídy `trainControl`, kterým přizpůsobíme proces trénování modelu našim potřebám. V kódu uvedeném níže zvolíme způsob provedení křížové validace jako pětkrát opakované náhodné dělení datové množiny na deset složek. Celkem tedy bude model testován na padesáti různých částech dat. Čím vyšší hodnotu opakování zvolíme, tím přesněji budeme moci předpovědět chování modelu s novými pozorováními, nicméně zároveň bude proces tréninku modelu výpočetně náročnější.

```
>ctrl <- trainControl(method = 'repeatedcv',number=10,repeats=5)
```

Množina testovaných hodnot parametrů, pokud metoda nějaký parametr vyžaduje, je určena automaticky. Nicméně je možné ji upravit s pomocí funkce `expand.grid()`. My této možnosti využijeme v případě, že při tréninku modelu byla jako optimální zvolena hodnota na hranici testované množiny.

V objektu `preProcess` můžeme zvolit způsob předzpracování dat. Toto zahrnuje imputaci chybějících hodnot, kterou v našem případě nebudeme potřebovat, nebo centrování a škálování proměnných. Toto předzpracování je naopak u některých vícerozměrných metod žádoucí. Při trénování modelu využijeme centrování průměrem dané proměnné a škálování její směrodatnou odchylkou. Testovací množina pak bude předzpracována stejným způsobem s hodnotami z tréninkové množiny.

Klasifikaci s pomocí balíčku si ukážeme na metodě $k$ nejbližších sousedů. Funkci poté nebude problém zobecnit pro libovolnou podporovanou metodu. Argumenty funkce, které se budou pro různé metody měnit, jsou `method`, případně `tuneGrid`. Využijeme několik různých metod tak, abychom vyzkoušeli i různé přístupy k zadanému problému. V kapitole 1 jsme se dozvěděli, že některé metody, především pak metody diskriminační analýzy, kladou na data speciální předpoklady, ačkoliv na ně dílčí metody nemusí být příliš citlivé. Proto dále uvedeme několik různých variant DA, které na rozdělení dat kladou různé podmínky. Pokud bychom se chtěli některé metodě věnovat blíže, předpoklady budou ověřeny zpětně. Porovnávané metody budou (v závorce uvádíme hodnotu argumentu `method` ve funkci `train()`) $k$ nejbližších sousedů (knn), lineární diskriminační analýza (lda), regularizovaná diskriminační analýza (rda), metoda podpůrných vektorů s lineární, resp. radiální jádrovou funkcí (svmLinear, svmRadial), diskriminační analýza – směs (mda), penalizační diskriminační analýza (pda), lokalizovaná lineární diskriminační analýza (loclda) a náhodný les (rf).

Kromě uvedených metod modelovaných s pomocí `caret`, doplníme později ještě dvě metody. Půjde o metodu nejmenších částečných čtverců (PLS) a její obdobu (SPLS), která v rámci svého algoritmu volí jen významné proměnné. Obě funkce jsou rovněž součástí balíčku `caret`, nicméně v současné době není u vstupních dat podporována vícerozměrná závisle proměnná. Kvůli tomuto omezení nebyly rovněž využity metody PLSDA, resp. SPLSDA. Klasifikační me-

tody jsme totiž chtěli doplnit i o takový přístup, který využije ordinální informaci proměnné velikost, čehož můžeme dosáhnout pouze modelováním velikosti a typu boty odděleně. Abychom si umožnili provést analýzu opakovaně se stejnými výsledky, je třeba před každým krokem, kdy do algoritmu vstupuje pseudonáhoda, využít funkci `set.seed()`.

```
>set.seed(123)

>knnGrid=expand.grid(k=c(1,3,5,7,9))

>fit.knn=train(
  training[,c(1,3:24)],
  training[,2],
  method = "knn",
  trControl = ctrl,
  tuneGrid = knnGrid,
  preProcess=c("center","scale"))

>fit.knn
k-Nearest Neighbors

200 samples
23 predictor
23 classes: '36 A', '37 A', '37 V', '38 A', '38 V', '39 A', '39 V',
'40 A', '40 V', '41 A', '41 V', '42 A', '42 V', '43 A', '43 V', '44
A', '44 V', '45 A', '
45 V', '46 A', '46 V', '47 A', '48 A'

Pre-processing: centered (22), scaled (22), ignore (1)
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 184, 181, 177, 181, 178, 179, ...
Resampling results across tuning parameters:

k Accuracy  Kappa
1 0.45727   0.41135
3 0.29733   0.23926
5 0.29451   0.23366
7 0.30794   0.24562
9 0.28159   0.21618

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 1.
```

Algoritmus spočítal kvalitu predikce metody pro každou hodnotu parametru na padesáti různých částech dat a v kódu výše pak můžeme vidět průměrnou hodnotu obou dostupných metrik. Obdobně budeme postupovat pro všechny uvedené metody. Průměrná hodnota metriky však nemusí pro srovnání metod postačovat za každé situace a mohly by nás zajímat další charakteristiky polohy. Ty získáme, použijeme-li následující kód.

```
>results <- resamples(list(knn=fit.knn,lda=fit.lda, rda=fit.rda,
   svm_radial=fit.svmr, svm_linear=fit.svml,
   mda=fit.mda,pda=fit.pda, loclda=fit.loclda, rf=fit.rf))

>summary(results)

Call:
summary.resamples(object = results)

Models: knn, lda, rda, svm_rad, svm_lin, mda, pda, loclda, rf
Number of resamples: 50

Accuracy
          Min.  1st Qu.  Median   Mean   3rd Qu.  Max.  NA's
knn     0.25000 0.37649 0.47339 0.45727 0.52632 0.65000 0
lda     0.15000 0.27597 0.35682 0.34280 0.40000 0.52941 0
rda     0.00000 0.01042 0.26136 0.21647 0.37782 0.58824 0
svm_rad 0.14286 0.27778 0.35000 0.35205 0.40000 0.52941 0
svm_lin 0.10526 0.30109 0.36068 0.35278 0.40000 0.61111 0
mda     0.16667 0.34837 0.39010 0.38527 0.44271 0.55000 20
pda     0.15000 0.28782 0.36603 0.36160 0.40882 0.55000 0
loclda  0.15000 0.31579 0.38889 0.38541 0.44861 0.60000 0
rf      0.26087 0.39348 0.43961 0.44467 0.50000 0.66667 0

Kappa
          Min.  1st Qu.  Median   Mean   3rd Qu.  Max.  NA's
knn     0.19786 0.32629 0.42212 0.41135 0.49022 0.62060 0
lda     0.09091 0.21649 0.30328 0.28679 0.34678 0.47893 0
rda     0.00000 0.00000 0.19598 0.17797 0.31720 0.54406 0
svm_rad 0.05025 0.20408 0.28049 0.27793 0.33264 0.47891 0
svm_lin 0.02121 0.22781 0.30379 0.29594 0.34929 0.57432 0
mda     0.12338 0.28800 0.34319 0.33524 0.37956 0.51613 20
pda     0.09091 0.22968 0.31325 0.30714 0.35484 0.50139 0
loclda  0.06849 0.26066 0.32881 0.33316 0.40041 0.56757 0
rf      0.19877 0.33779 0.38730 0.39223 0.45130 0.63390 0
```

Pozornému čtenáři jistě neunikne, že metoda MDA během klasifikace ve dvaceti případech selhala. Nejednalo se však o žádnou závažnou chybu. Metoda nebyla schopná natrénovat model v momentě, kdy byla v algoritmu pro některé třídy uvažována směs více dílčích rozdělení, než-li bylo k dispozici pozorování.

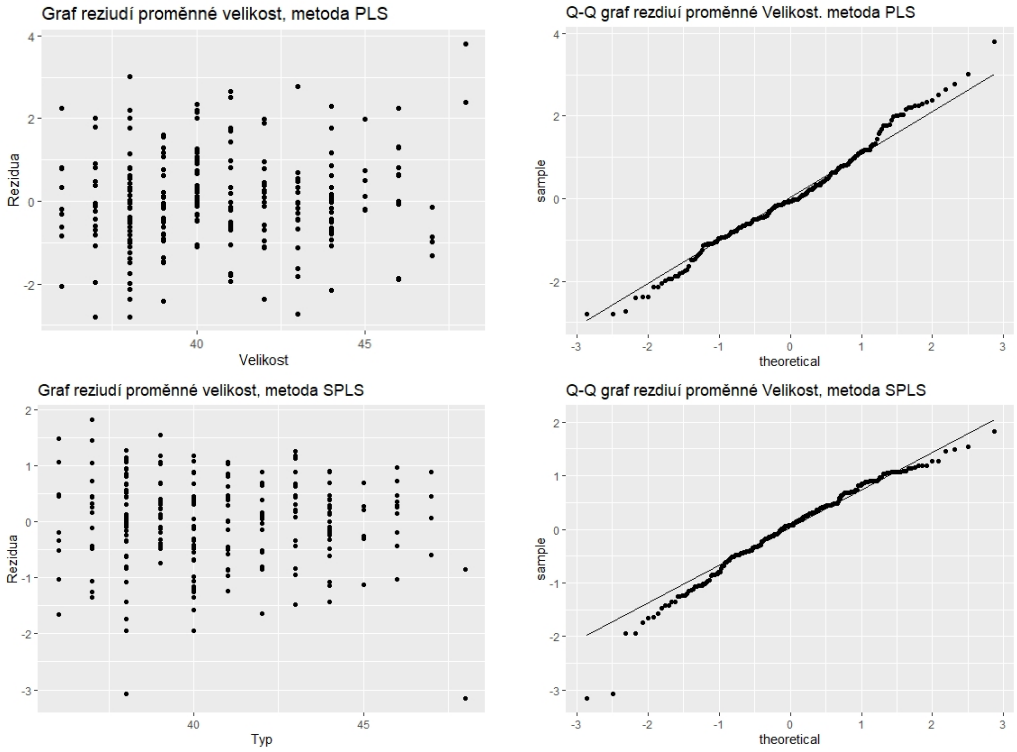
Pojďme nyní ke zmíněným metodám doplnit PLS a SPLS z knihoven `pls` [23], resp. `spls` [24]. Nejprve však ověříme, že v našich datech můžeme pozorovat multikolinearitu, a použití těchto metod tak má opodstatnění. Konkrétně využijeme funkci `omcdiag()` knihovny `mctest` [25].

```
>training_pls=training
>training_pls=separate(training_pls,"Size",c("Size","Type"),sep="")
>training_pls$Size=as.numeric(training_pls$Size)
>training_pls$Type=as.numeric(as.factor(training_pls$Type))-1
>training_pls$Sex=as.numeric(training_pls$Sex)-1
>training_pls=list(X=data.matrix(training_pls[,c(1,4:25)]),
   Y=data.matrix(training_pls[,c(2,3)]))

>omcdiag(training_pls$X,training_pls$Y[,1])

Call:
omcdiag(x = training_pls$X, y = training_pls$Y[, 1])

Overall Multicollinearity Diagnostics

                       MC Results detection
Determinant |X'X|:         0.0000         1
Farrar Chi-Square:     11302.3278         1
Red Indicator:             0.6211         1
Sum of Lambda Inverse: 18064.2547         1
Theil's Method:            0.5747         1
Condition Number:       7611.6687         1

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test
```

Všechny dostupné testy multikolinearitu v datech odhalily, můžeme tedy přejít k tvorbě modelů. Postup si ilustrujeme pouze na první z nich, postup u druhé bude obdobný. Trénovací množinu jsme nejprve upravili tak, abychom problém mohli modelovat s pomocí regrese se dvěma závislými proměnnými. Tj. rozdělili jsme velikost a typ do dvou proměnných. Obdobně budeme postupovat s množinou testovací. Křížová validace pro volbu parametrů metod se ten-

tokrát provádí odděleně s pomocí funkce `mvr_dcv()`.

```
>set.seed(123)

>fit.pls.cv <- mvr_dcv(Y X, data=training_pls, ncomp=8,
   method="simpls", segments=10, scale=TRUE)
>fit.pls.cv$afinal
[1] 1

>fit.pls<-mvr(Y X,data=training_pls,method="simpls",ncomp=1,
   scale=TRUE)
```




Obrázek 3.1: Bodový a Q-Q graf reziduí proměnné `Size` pro metody PLS (nahoře) a SPLS (dole).

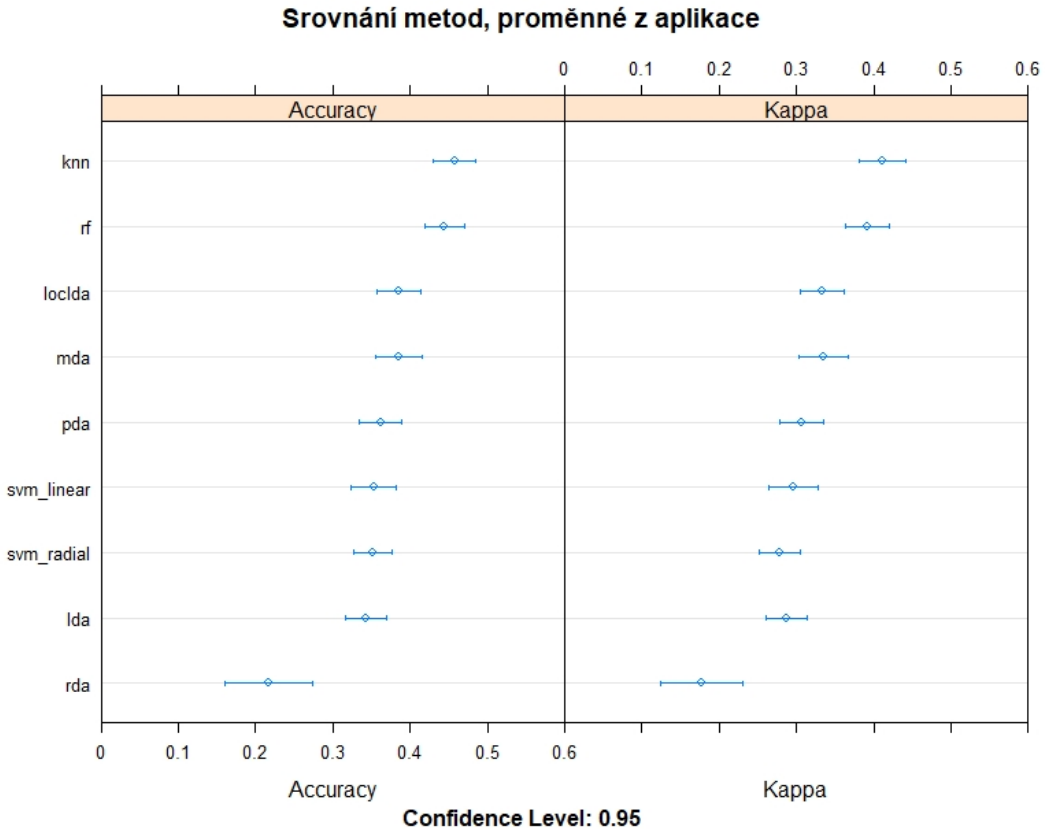
Regresní metody dosáhly na tréninkové množině přesnosti 30% a 41% pro PLS resp. pro SPLS, což jsou naše referenční hodnoty pro srovnání s ostatními metodami. Podívejme se ještě na rezidua obou metod, abychom ověřili, zda jsou pro námi zvolený problém vhodné. Konkrétně pak na bodový graf a Q-Q graf reziduí proměnné `Size`. Proměnná `Type` je dichotomická a bylo by vhodnější ji modelovat např. s pomocí logistické regrese, nicméně v této fázi by to predikční

model zkomplikovalo.

Na obrázku 3.1 můžeme vidět, že pro metodu PLS jsou předpoklady pro rezidua téměř splněny. Pro metodu SPLS se zřejmě nemůžeme opřít o předpoklad homoskedascity ani normality.

Vraťme se zpět ke srovnání metod. Textový zápis s charakteristikami polohy trénovaných metod uvedený výše není jediná pomůcka, se kterou můžeme metody porovnat. Balíček `caret` dále ve spolupráci s knihovnou `lattice` umí vykreslit přehledné srovnávací grafy. Pokud argumenty funkce `dotplot()` nespecifikujeme, je do grafu zanesena průměrná hodnota dostupných metrik. Metody jsou navíc dle těchto průměrů sestupně seřazeny. Kromě průměru je pak v grafu vyznačen i interval, ve kterém se s pravděpodobností 95% nachází skutečná hodnota metriky. Alternativně je pak možné do grafu vykreslit např. krabicový graf.

```
>dotplot(results)
```




Obrázek 3.2: Graf srovnání klasifikačních metod.

V grafu na obr. 3.2 můžeme vidět, že, alespoň tedy pro naše data, žádná metoda průměrnou přesností klasifikace nepřekročila 50%. Tato čísla však nene-

sou informaci o tom, jak moc velkých chyb se jednotlivé metody při klasifikaci dopouštěly. S využitím testovací množiny můžeme ilustrovat, jak se přibližně jednotlivé metody během klasifikace pletly. Pro přesnější interpretaci můžeme využít matici záměn nebo predikční graf.

Reprezentace formou matice záměn se pro velký počet klasifikačních tříd, jako v našem případě, může stát lehce nepřehlednou. Tuto formu si tak tedy budeme opět ilustrovat pouze na metodě knn. Nejprve využijeme testovací množinu pro predikci velikosti boty a následně pak zobrazíme matici. Informace pro předzpracování dat (centrování a škálování) jsou v objektu `fit.knn` již uloženy a nová pozorování jsou tak upravena automaticky.

```
>knn.predictions=predict(fit.knn, testing)

>confusionMatrix(knn.predictions,testing$Size)
```

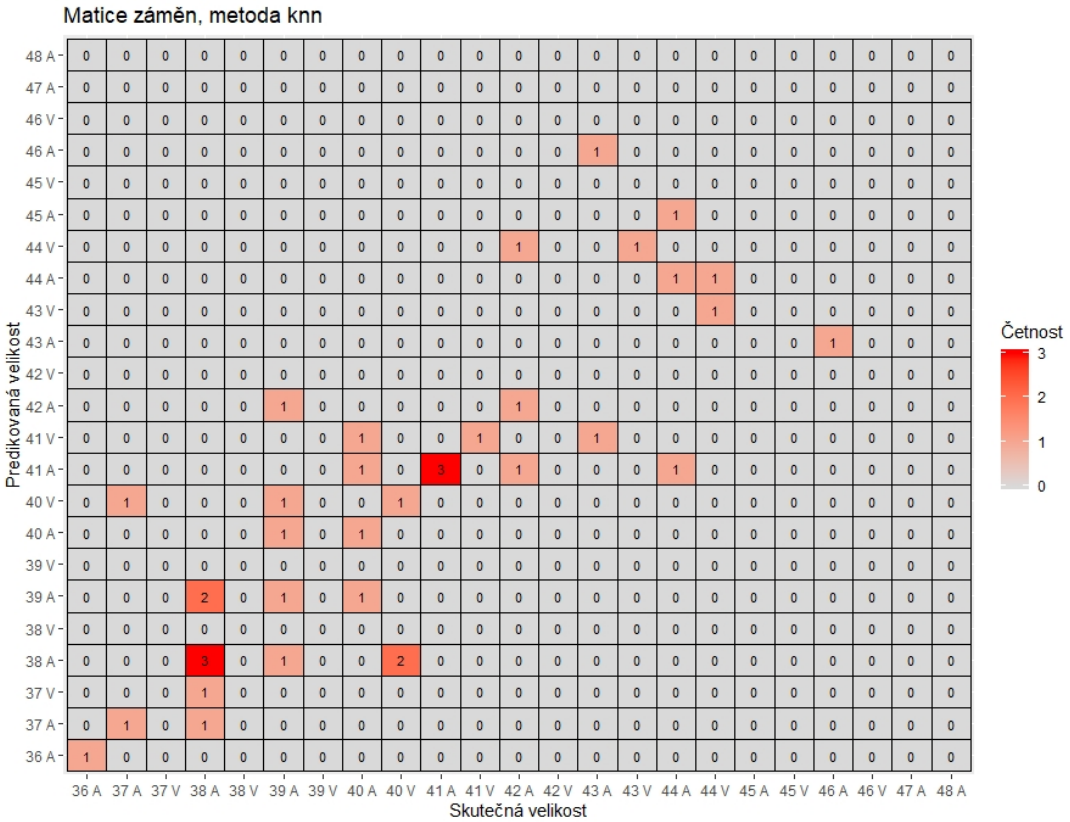
```
Confusion Matrix and Statistics

          Reference
Prediction 36 A 37 A 37 V 38 A 38 V 39 A 39 V 40 A 40 V 41 A 41 V 42 A 42 V 43 A 43 V 44 A 44 V 45 A 45 V 46 A 46 V 47 A 48 A
     36 A    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     37 A    0    1    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     37 V    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     38 A    0    0    0    3    0    1    0    0    2    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     38 V    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     39 A    0    0    0    2    0    1    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     39 V    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     40 A    0    0    0    0    0    1    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     40 V    0    1    0    0    0    1    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     41 A    0    0    0    0    0    0    0    1    0    3    0    1    0    0    0    1    0    0    0    0    0    0    0
     41 V    0    0    0    0    0    0    0    1    0    0    1    0    0    1    0    0    0    0    0    0    0    0    0
     42 A    0    0    0    0    0    1    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0
     42 V    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     43 A    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0
     43 V    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0
     44 A    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1    0    0    0    0    0    0
     44 V    0    0    0    0    0    0    0    0    0    0    0    1    0    0    1    0    0    0    0    0    0    0    0
     45 A    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0
     45 V    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     46 A    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0
     46 V    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     47 A    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     48 A    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

Overall Statistics

               Accuracy : 0.3684
                 95% CI : (0.2181, 0.5401)
    No Information Rate : 0.1842
    P-Value [Acc > NIR] : 0.005765

                  Kappa : 0.3086

 Mcnemar's Test P-Value : NA
```

Obrázek 3.3: Matice záměn pro metodu knn.

Pokud by zvolená metoda klasifikovala pozorování přesně, byla by matice (na obr. 3.3) diagonální. Jak bylo předpokládáno, matice je poměrně nepřehledná. K usnadnění interpretace matice si můžeme pomoci s grafickými prvky balíčku `ggplot2` [26]. Tentokrát však matici záměn vytvoříme pomocí funkce `confusion()` balíčku `mlearning`. Uvedený postup vychází z kódu dostupném on-line [27].

```
>confusion=confusion(knn.predictions,testing$Size)
>confusion_mat=data.frame(confusion)
>confusion_mat=ddply(confusion_mat, "Actual", transform)
```

```
>ggplot() +
   geom_tile(aes(x=Actual, y=Predicted, fill=Freq),
     data=confusion_mat,color="black",size=0.1) +
   labs(title="Matice záměn, metoda knn",x="Skutečná velikost",
     y="Predikovaná velikost",fill="Četnost") +
   geom_text(aes(x=Actual,y=Predicted, label=Freq),
     data=confusion_mat, size=3, colour="black") +
   scale_fill_gradient(low="grey85",high="red") +
   geom_tile(aes(x=Actual,y=Predicted),
     data=subset(confusion_mat, as.character(Actual)==as.character(
     Predicted)), color="black", size=0.3, fill="black", alpha=0)
```

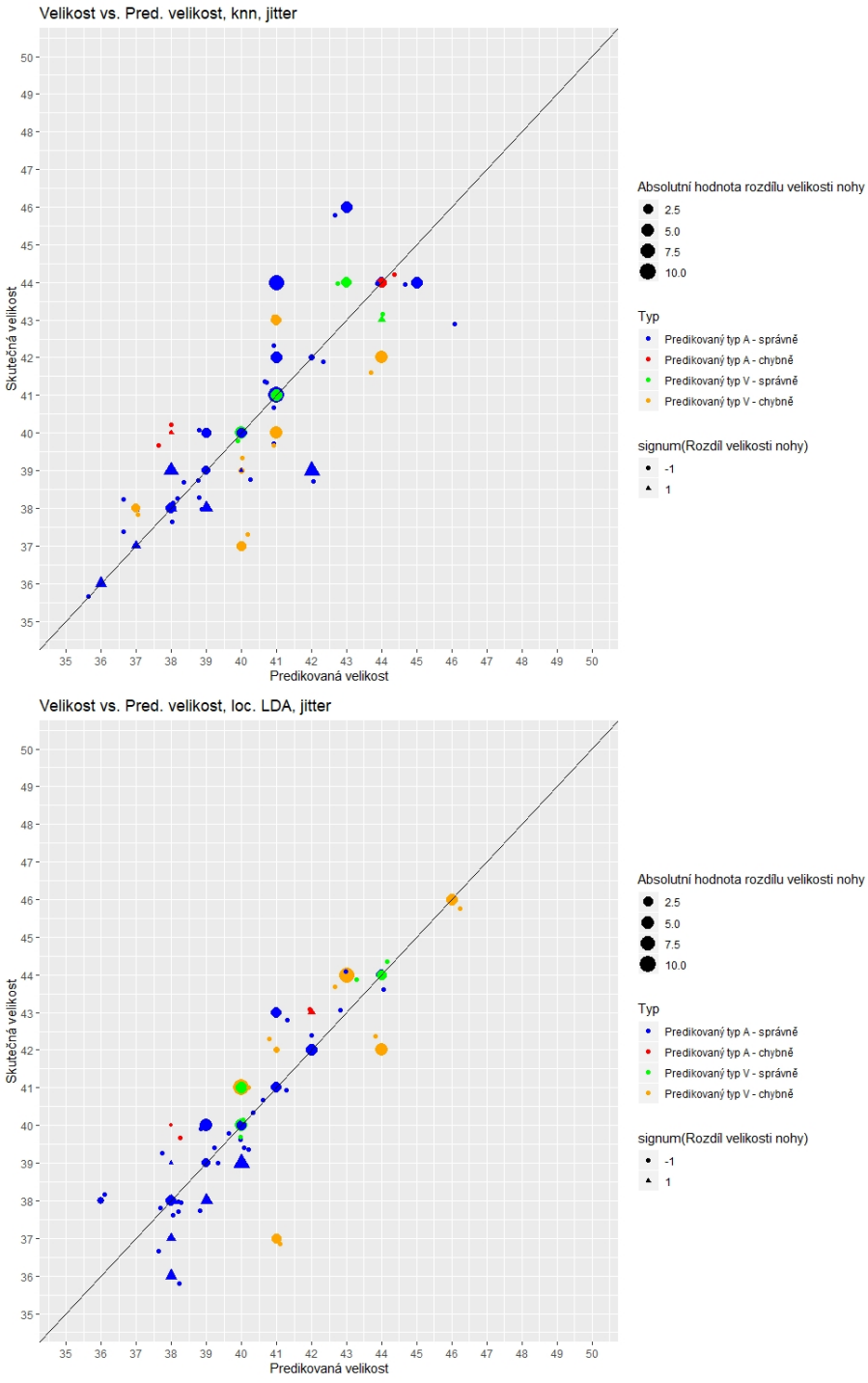



Obrázek 3.4: Graficky upravená matice záměn.

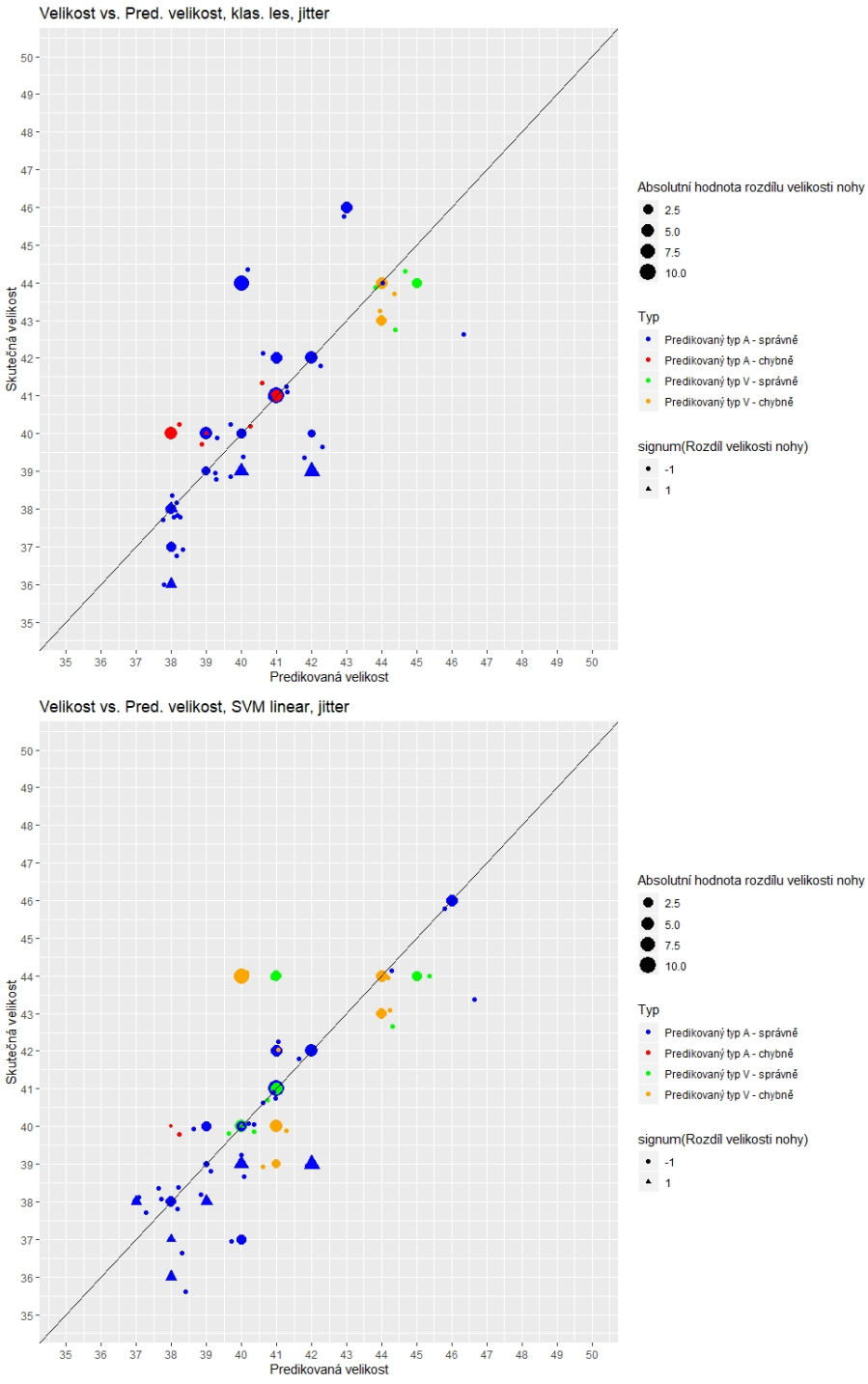
Ještě více informací, než z matice záměn, můžeme získat z predikčních grafů. Pro jejich vykreslení využijeme opět predikce získané funkcí `predict()`. Pro účely vykreslení grafů zároveň predikce rozdělíme do dvou proměnných zvlášť pro ve-

likost a typ. Všechny informace, zanesené do grafu, musí být v případě knihovny `ggplot2` uložené do jednoho objektu, proto zároveň vytvoříme objekt třídy `data.frame`. V práci uvedeme grafy tří nejlepších metod dle přesnosti, dále přidáme grafy pro SVM a SPLS, abychom mohli porovnat zástupce všech představených přístupů k našemu problému.

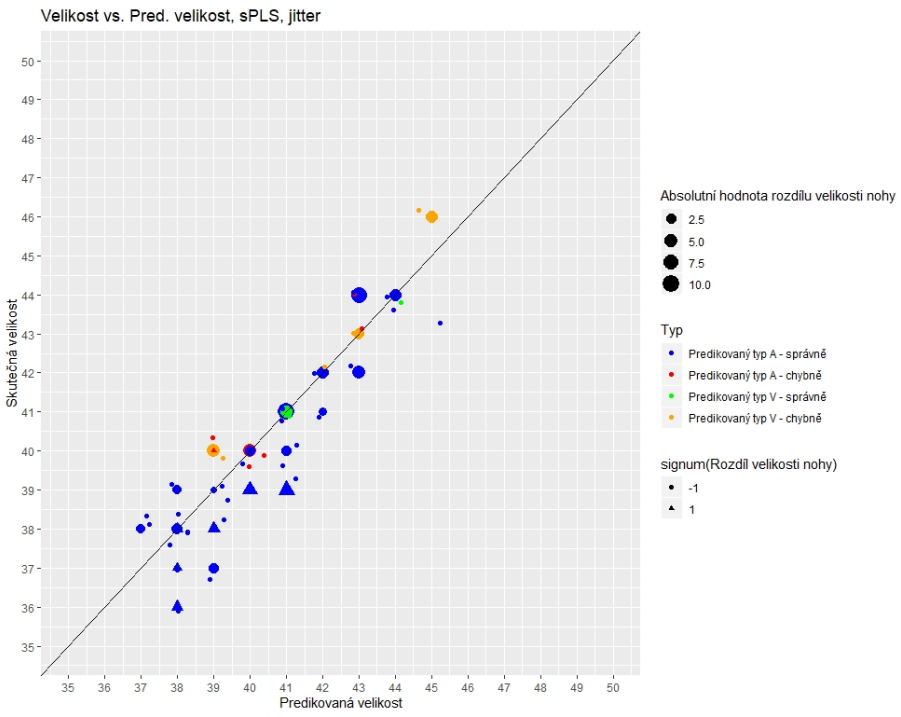
```
>knn.predictions=predict(fit.knn, testing)
>knn.predictions.sep=unlist(strsplit(x=as.character(knn.predictions),
   split=""))
>knn.predictions.sep=matrix(knn.predictions.sep,ncol=2,byrow=TRUE)

>knn.df=data.frame(x=as.numeric(knn.predictions.sep[,1]),
   y=testing_sep$Size,col=rep(0,38),diff=data_clean[-train_data,27])

>for (i in 1:38) {
   if (knn.predictions.sep[i,2]=="A"&
    knn.predictions.sep[i,2]==testing_sep[i,3]) {knn.df$col[i]=1}
   else if (knn.predictions.sep[i,2]=="A"&
    knn.predictions.sep[i,2]!=testing_sep[i,3]) {knn.df$col[i]=2}
   else if (knn.predictions.sep[i,2]=="V"&
    knn.predictions.sep[i,2]==testing_sep[i,3]) {knn.df$col[i]=3}
   else {knn.df$col[i]=4}

>ggplot(data=knn.df,aes(x=x,y=y,col=as.factor(col))) +
   geom_point(aes(size = abs(diff), shape=as.factor(sign(diff)))) +
   geom_jitter() +
   scale_x_continuous(breaks=seq(from=35,to=50,by=1),
     limits=c(35,50)) +
   scale_y_continuous(breaks=seq(from=35,to=50,by=1),
     limits=c(35,50)) +
   labs(title="Velikost vs. Pred. velikost, knn, jitter") +
   xlab("Predikovaná velikost") +
   ylab("Skutečná velikost") +
   geom_abline(intercept = 0, slope = 1) +
   scale_colour_manual(name="Typ", breaks=c(1,2,3,4),
     labels=c("Predikovaný typ A - správně",
      "Predikovaný typ A - chybně",
      "Predikovaný typ V - správně",
      "Predikovaný typ V - chybně"),
     values=c("blue","red","green","orange")) +
   scale_size(name="Absolutní hodnota rozdílu velikosti nohy") +
   scale_shape_discrete(name="signum(Rozdíl velikosti nohy)",
     breaks=c(-1,1), labels=c("-1","1"))
```




Obrázek 3.5: Predikční graf pro metody knn a lokální LDA.




Obrázek 3.5: Grafy predikcí pro metody náhodný les a SVM s lineární jádrovou funkcí.




Obrázek 3.5: Grafy predikcí pro metodu SPLS.

Věnujme se na chvíli interpretaci grafů na obrázku 3.5. Osa $y = x$ spojuje body, na kterých se nachází správně klasifikované velikosti. Pokud se několik pozorování nachází ve stejném bodě, jsou rozptýlena pomocí funkce `geom_jitter()`. Díky tomu si můžeme z grafů udělat lepší představu o přesnosti klasifikace. Barevně jsou rozlišené jednotlivé případy, které mohly nastat u klasifikace typů. Správně klasifikované jsou ty body, které jsou modré a zelené. Červené a oranžové jsou naopak klasifikovány chybně. Velikostně se body liší dle absolutní hodnoty rozdílu mezi délkou nohy naměřenou ručně a délkou z aplikace. Čím větší bod je, tím větší byl rozdíl. Tvarem je pak odlišeno znaménko u zmíněného rozdílu. Vzpomeneme-li si na způsob výpočtu rozdílu, pak si uvědomíme, že trojúhelníkem jsou označena pozorování, u kterých aplikace délku nohy nadhodnotila oproti námi zjištěné hodnotě.

Už jen z letmého pohledu na grafy můžeme usoudit, že ačkoliv zvolené metody pracovaly s přibližně stejnou přesností, některé z nich přeci jen klasifikovaly s o něco menší chybou. Na první pohled se zdá, že nejvhodnější metody z tohoto hlediska, by mohly být locLDA a SPLS. Na základě grafů nemůžeme jednoznačně říct, že za chybovost klasifikace může pouze odchylka při měření. Nicméně u pozorování s nadhodnocenými rozměry můžeme přeci jen pozorovat tendenci přiřazovat větší velikosti obuvi. Další věc, která nás v předešlém textu zajímala, byla, zda budou mít metody tendenci přiřazovat menším nohám velikost 38. Z grafů se opravdu zdá, že uvedené metody oproti jiným velikostem

přiřazují 38 chybně častěji. Nicméně na celkové přesnosti klasifikace by se změna rozdělení projevila pravděpodobně jen v řádech několika procent.

Pojďme si domněnky vyslovené na základě grafů ověřit exaktně, a sice tabulkou s přesností klasifikace pro testovací množinu. Tentokrát však i v horizontu ± jedna velikost.

| | Přesně | ± jedna velikost, typ přesně | Přesně velikost | ± jedna velikost |
|---|---|---|---|---|
| KNN | 36.84 | 65.79 | 39.47 | 76.31 |
| LDA | 52.63 | 71.05 | 60.52 | 84.21 |
| RDA | 44.74 | 68.42 | 52.63 | 84.21 |
| SVMrad | 50.0 | 73.68 | 50.0 | 78.95 |
| SVMlin | 42.1 | 68.42 | 44.74 | 78.95 |
| MDA | 47.37 | 73.68 | 55.26 | 86.48 |
| PDA | 52.61 | 73.68 | 60.52 | 84.21 |
| locLDA | 47.37 | 71.05 | 50.0 | 84.21 |
| rf | 47.37 | 68.42 | 55.26 | 81.58 |
| PLS | 26.31 | 55.26 | 34.21 | 71.05 |
| sPLS | 36.84 | 65.79 | 50.0 | 89.47 |

Tabulka 3.1: Srovnání přesnosti klasifikace na testovacím souboru (v %).

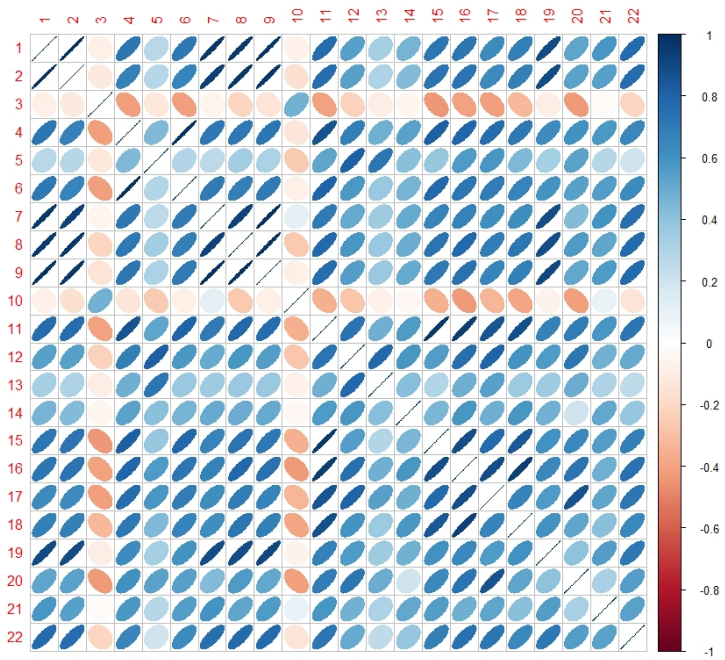
V tabulce 3.1 je v prvním sloupci uveden podíl pozorování testovací množiny, která byla zařazena naprosto přesně. Ve druhém sloupci pak podíl těch, která byla zařazena přesně dle typu, navíc ale tolerujeme chybu o jednu velikost. Třetí sloupec srovnává přesnost v případě, kdy byla pozorování zařazena správně dle velikosti, ale nebyl vybrán správný typ obuvi. V posledním sloupci je podíl pozorování, která byla zařazena s tolerancí chyby o jednu velikost. Z tabulky můžeme vidět, že metody dosahují uspokojivé přesnosti predikce až ve chvíli, pokud tolerujeme chybu a neuvažujeme typy obuvi. Zřejmě bychom tak dosáhli lepších hodnot, pokud by typ obuvi do tréninku modelů vůbec nevstupoval, případně pokud by typ bylo možné určit odděleně od velikosti. Během sběru dat se zdálo, že jeden z typů je oproti stejným velikostem typu druhého trochu širší. Zkusili jsme tedy vytvořením vhodných dodatečných proměnných, jako např. poměr šířky. resp. obvodu nohy v různých místech a délky nohy, otestovat, zda je mezi typy bot opravdu v tomto ohledu rozdíl. T-test středních hodnot však významný rozdíl neodhalil. Typ tedy není možné z dostupných dat odhadnout separovaným pravidlem.

Přesnost predikce pro metody uvedené v práci není dostatečná, aby mohla být některá z metod dále použitá v praxi. Nicméně pro přesnost predikce na základě proměnných z aplikace na měření chodidel nemáme žádnou referenční hodnotu, se kterou bychom predikci mohli porovnat. Dodatečně tedy s využitím stejného postupu natrénujeme modely znovu, tentokrát však využijeme jen minimum informací. Jako prediktory použijeme jen data, která může každý uvést i bez této

aplikace, a sice velikost nohy z měřidla a pohlaví.

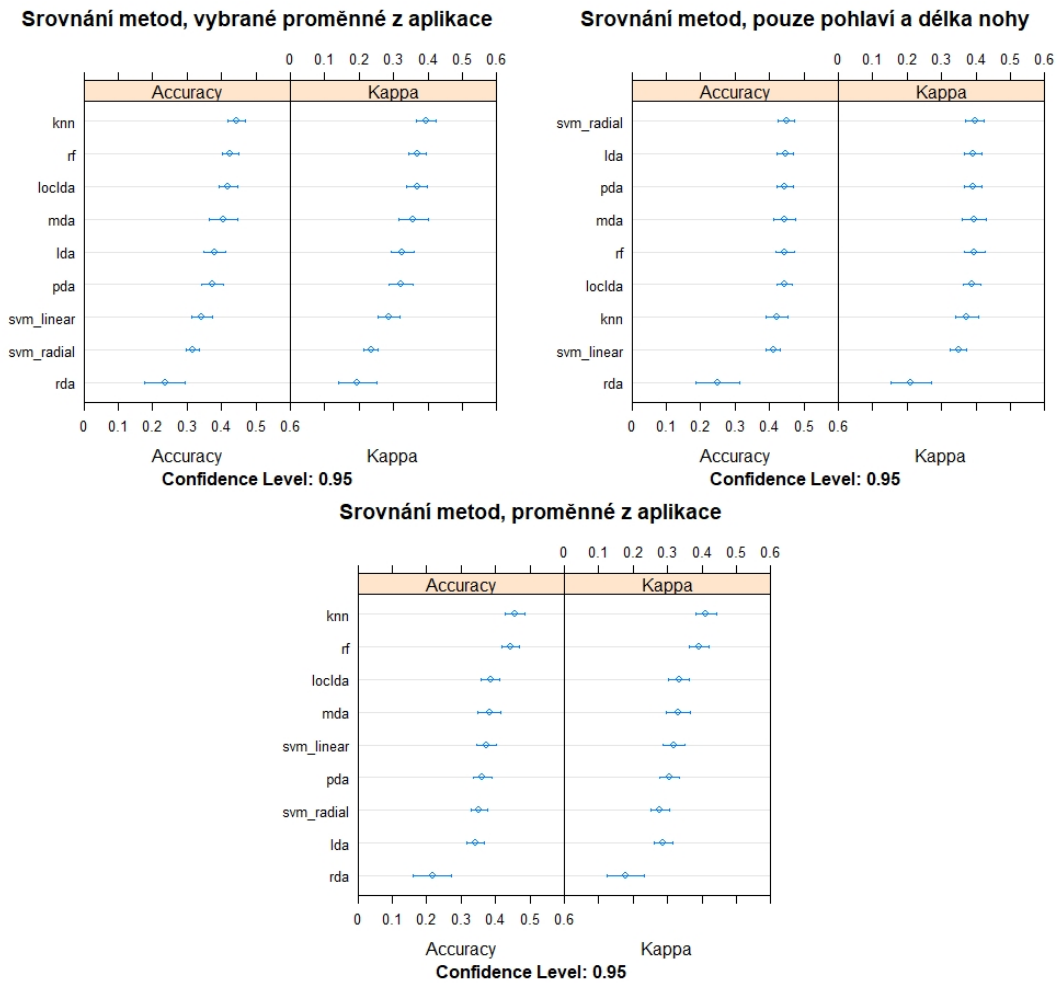
V [2] je navíc uvedeno, že některé prediktivní metody dosahují lepších výsledků, pokud se v datech nevyskytují prediktory s vysokými hodnotami párových korelačních koeficientů. Nejprve zjistíme, zda se tato situace týká i našich dat tak, že vykreslíme korelační matici s pomocí balíčku `corrplot`.

```
>cordata=data_clean[,5:26]
>names(cordata)=c(as.character(1:22))
>cor.matrix=cor(cordata)
>corrplot(cor.matrix,method="ellipse")
```




Obrázek 3.6: Graficky znázorněná korelační matice použitých prediktorů.

Z grafu na obr. 3.6 můžeme vidět, že se v datech opravdu nachází hned několik dvojic prediktorů s vysokou, často kladnou, hodnotou korelace. Zkusíme proto přidat i srovnání metod po odstranění některých z těchto proměnných. Přidáním hodnoty `'corr'` do argumentu `preProcess` ve funkci `train()` můžeme výběr proměnných ponechat na počítači. Před trénováním modelu pak algoritmus nejdříve vyčíslí hodnoty párových korelačních koeficientů mezi prediktory a označí dvojice, jejichž koeficient přesahuje mez 0,75. Prediktor, který se mezi takovýmito dvojicemi vyskytne nejčastěji, je odstraněn a proces se opakuje. Výsledky vykreslíme do grafů, jako v předchozím případě, pro srovnání přidáme ještě jednou graf s původními modely.




Obrázek 3.7: Grafy se srovnáním metod. Vlevo nahoře metody trénované pouze s pomocí vybraných proměnných dle párových korelačních koeficientů. Vpravo nahoře metody trénované pouze s pomocí proměnných Pohlaví a Délka nohy. Dole metody trénované se všemi proměnnými z aplikace.

Dle grafů na obrázku 3.7 můžeme u některých metod po odstranění prediktorů na základě korelačních koeficientů opravdu pozorovat mírné zlepšení predikce. Pokud jsme pozorování klasifikovali pouze na základě délky nohy a pohlaví, zlepšila se přesnost predikce některých metod až o přibližně 10%. Tyto domněnky můžeme potvrdit, jakmile si prohlédneme přesné hodnoty průměrů. Původní modely tedy nedosahují vyšší přesnosti předpovědí, než ty, které predikovaly velikost jen na základě dat, ke kterým aplikace není třeba. Skutečnost, že ani s využitím pouhé délky nohy jsme nedosáhli vysoké přesnosti klasifikace, nasvědčuje tomu, že do volby velikosti bot vstupují i další proměnné, jak jsme se na začátku domnívali.

Jelikož byly v práci uvedeny různé metody, které k predikci přistupují různými

způsoby, jeví se nepravděpodobně, že bychom dosáhli výrazně lepších výsledků volbou některé další metody. Jako problematická se nám při sestavování modelu zdá nepřesnost aplikace, která byla použita pro měření chodidel. Toto se pak projevuje vzájemným prolínáním sousedních tříd a skokovitým nárůstem přesnosti, pokud při klasifikaci tolerujeme chybu. V predikčních grafech testovací množiny jsme však pozorovali, že odchylka v naměřených hodnotách není patrně jediným faktorem, který způsobuje chybovost predikce. Podíl na ní může nést i to, že do volby obuvi zasáhly i pocity a dojmy při zkoušení bot. Eliminovat tento subjektivní faktor však při sběru dat nebylo cílem a ani možné.

# Závěr

Nakupování on-line je v dnešní době rychle se rozvíjející odvětví. Stále větší podíl nákupů probíhajících na internetu vystavuje prodejce novým výzvám. V obuvnictví může být takovouto výzvou umožnit zákazníkům volbu optimální velikosti bot z pohodlí domova. Tato myšlenka byla inspirací i k napsání této diplomové práce, která pak vznikla ve spolupráci se společností Prabos, a. s. vyrábějící outdoorovou obuv.

Hlavním úkolem v práci bylo zjistit, zda je pro automatizaci procesu volby vhodné velikosti obuvi možné využít prediktivní metody mnohorozměrné statistické analýzy. Velikost boty jsme určovali pro dva různé druhy bot mírně se lišící tvarem. Aby konkrétní boty zákazníkovi padly, bylo třeba znát číselné charakteristiky jeho chodidel. Za tímto účelem nám poskytla společnost aplikaci, která dokáže na základě série fotografií odhadnout rozměry chodidla umístěného na listu papíru. Tuto aplikaci jsme použili při tvorbě datového souboru. Každý člověk změřený aplikací měl následně za úkol si vybrat velikost a typ obuvi, který mu padne nejlépe. Během měření jsme zaznamenávali i další údaje, které mohly být při tvorbě modelů potenciálně důležité. Kromě námi nasbíraných dat poskytla několik desítek měření i společnost Prabos, a. s.

Data byla dále zpracovávána v softwaru R. Nejprve bylo třeba automatizovat proces stahování dat z cloudu. Dále byla všechna podstatná data spojena do jediného souboru. Ten byl pak očištěn od těch nejvíce nepřesných měření, k čemuž jsme využili délku nohy naměřenou na ševcovském měřidle. Očištěná data jsme dále použili k tvorbě modelů v R s pomoc balíčku `caret`. V práci byly použity především klasifikační metody, mezi nimi pak metody diskriminační analýzy, $k$ nejbližších sousedů, metody podpůrných vektorů nebo náhodné lesy. Ty jsme následně doplnili o regresní metody, konkrétně metodu nejmenších částečných čtverců a její obdobu s výběrem významných proměnných. Teoretický základ k některým těmto metodám jsme uvedli na začátku práce.

Metrikou, která nás při porovnávání metod zajímala, byla přesnost predikce, tj. podíl správně zařazených pozorování. Dle této metriky byly při predikci nejúspěšnější metoda nejbližšího souseda a náhodné lesy s přesností přesahující 40%. Proto jsme použité metody shledali pro naši úlohu nevyhovující, což jsme dále demonstrovali srovnáním s predikcí velikostí pouze na základě délky nohy a pohlaví. Mezi možné důvody vysoké chybovosti při předpovědích jsme uvedli nepřesnost

aplikace při měření chodidla. Aplikace byla citlivá na některé faktory, jako třeba internetové připojení. Především pak ale pracovala pouze s odhady proměnných, což mohlo mít za následek vzájemné prolínání sousedních tříd. Toto mělo zřejmě za následek i výrazné zvýšení přesnosti predikce až na téměř 90%, pokud jsme tolerovali chybu o jednu velikost. Dalším důvodem mohlo být to, že jsme během měření nedokázali zcela potlačit subjektivitu při volbě velikosti.

Hlavního cíle práce bylo dosaženo. Ačkoliv v současné době nelze tento přístup k zavedení do praxe doporučit, jsme schopni na základě našeho výzkumu zformulovat několik doporučení, která by mohla v budoucnu pomoci dosáhnout lepších výsledků. Samotná myšlenka, doporučit vhodnou velikost boty na základě několika rozměrů chodidla, je ve své podstatě správná a očekávali bychom, že bude fungovat lépe, než jak jsme mohli pozorovat v této práci. K původní myšlence by se bylo možné znovu vrátit v momentě, kdy bude měřící aplikace fungovat s větší přesností. Dále doporučujeme zajistit větší rozsah tréninkové množiny tak, aby bylo možné vytvořit separované modely dle pohlaví, což se v práci jevilo jako vhodnější přístup. V neposlední řadě by k větší přesnosti predikce pravděpodobně přispělo, pokud bychom od modelu neočekávali vedle určení velikosti i volbu typu boty. Nicméně je třeba respektovat, že stejné velikosti bot různých typů nemůžeme kvůli různým tvarům považovat za vzájemně nahraditelné.

Pro mě osobně byl největší výzvou v rámci práce sběr dat, který byl chvílemi velmi psychicky i fyzicky náročný. Jelikož se jednalo o můj první podobný projekt, nebylo během analýzy někdy snadné ani udržet kód přehledný tak, aby bylo možné na něj dále bez problému navázat nebo jej zpětně editovat a udržet tak v práci systematičnost. Během zpracování a analýzy dat jsem si značně prohloubil znalosti v softwaru R. Především pak v oblasti zpracování a vizualizace dat a tvorby predikčních modelů. Tvorba modelů a jejich srovnávání pak byla i část, která mě v práci bavila nejvíce.

# Literatura


[1] Wehrens, R.: *Chenometrics with R: Multivariate Data Analysis in the Natural Sciences nad Life Sciences* Springer, Heidelberg, 2011, ISBN 978-3-642-17840-5.

[2] The `caret` Package [online], dostupné z: https://topepo.github.io/caret/index.html [citováno 2. 3. 2020].

[3] Sim, J., Wright, Ch. C.: *The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements* Physical Therapy., Vol. 85, No. 3, 2005, s. 257–268.

[4] Machine Learning in R for beginners [online], dostupné z: https://www.datacamp.com/community/tutorials/machine-learning-in-r [citováno 26. 4. 2020].

[5] Varmuza, K., Filzmoser, P.: *Introduction to Multivariate Statistical Analysis in Chemometrics* CRC Press, New York, 2009, ISBN 978-1-4200-5947-2.

[6] Hendl, J.: *Přehled statistických metod: Analýza a metaanalýza dat* Portál, Praha, 2012, ISBN 978-80-262-0200-4.

[7] Czogiel, I., Luebke, K., Zentgraf, M., Weihs, C.: *Localized Linear Discriminant Analysis* Technical Report. No. 10, 2006.

[8] Localized Linear Discriminant Analysis (LocLDA) [online], dostupné z https://www.rdocumentation.org/packages/klaR/versions/0.6-14/topics/loclda [citováno 15. 3. 2020].

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

[10] Datový soubor 'donors.csv' [online], dostupné z: https://assets.datacamp.com/production/repositories/718/datasets/9055dac929e4515286728a2a5dae9f25f0e4eff6/donors.csv [citováno 15. 4. 2020].

[11] Tattar, P. N.: *Hands-On Ensemble Learning with R: A Beginner's Guide to Combining the Power of Machine Learning Algorithms Using Ensemble Techniques* Packt Publishing, Birmingham, 2018, ISBN 978-1-78862-414-5.

[12] Yong, Q., Baoming, L., Lijun, D., Limin, J., Zhigang, L., Min, A.: *Proceedings of the 4th International Conference on Electrical and Information Technologies for Rail Transportation (EITRT) 2019* Springer Nature, Singapore, 2020, ISBN 978-981-15-2865-1.

[13] James, G., Witten, D., Hastie, T., Tibshirani, R.: *An Introduction to Statistical Learning: With applications in R* Springer, New York, 2013, ISBN 978-1-4614-7137-0.

[14] Support Vector Machines (Detailed Explenation) [online], dostupné z: https://towardsdatascience.com/support-vector-machine-support-vector-classifier-maximal-margin-classifier-22648a38ad9c [citováno 28. 4. 2020].

[15] Chung, D., Keles, S.: *Sparse Partial Least Squares Classification for High Dimensional Data* Austrian Journal of Statistics. Vol. 9, No. 1, 2010.

[16] Chung, D., Keles, S.: *Sparse partial least squares regression for simultaneous dimension reduction and variable selection* Journal of the Royal Statistical Society: Series B (Statistical Methodology). Vol. 72, No. 1, 2010, s. 3–25.

[17] Olson Hunt M. J., Weissfeld L., Boudreau R. M., Aizenstein H., Newman A. B., Simonsick E. M., Van Domelen D. R., Thomas F., Yaffe K., Rosano C.: *A variant of sparse partial least squares for variable selection and data exploration.* Frontiers in Neuroinformatic. Vol. 8, 2014.

[18] E-shop Prabos, a. s. [online], dostupné z: https://eshop.prabos.cz/ [citováno 24. 3. 2020].

[19] A Grammar of Data Manipulation [online], dostupné z: https://www.rdocumentation.org/packages/dplyr/versions/0.7.8 [citováno 30. 4. 2020].

[20] Easily Tidy Data with '`spread()`' and '`gather()`' Functions [online], dostupné z: https://www.rdocumentation.org/packages/tidyr/versions/0.8.3 [citováno 30. 4. 2020].

[21] Mickle, K., Munro, B., Lord, S., Menz, H., Steele, J.: *Foot shape of older people: Implications for shoe design.* Footwear Science. Vol. 2, No. 3, 2010, s. 131–139.

[22] 3D Scatter Plot [online], dostupné z: https://www.rdocumentation.org/packages/scatterplot3d/versions/0.3.3 [citováno 1. 5. 2020].

[23] Partial Least Squares and Principal Component Regression [online], dostupné z: https://www.rdocumentation.org/packages/pls/versions/2.7-2 [citováno 29. 4. 2020].

[24] Sparse Partial Least Squares (SPLS) Regression and Classification [online], dostupné z: https://www.rdocumentation.org/packages/spls/versions/2.2-3 [citováno 29. 4. 2020].

[25] Multicollinearity Diagnostic Measures [online], dostupné z: https://www.rdocumentation.org/packages/mctest/versions/1.2.5 [citováno 29. 4. 2020].

[26] Dokumentace ke knihovně `ggplot2` [online], dostupné z: https://www.rdocumentation.org/packages/ggplot2/versions/3.3.0 [citováno 25. 4. 2020].

[27] ggplot2_tricks [online], dostupné z: https://gist.github.com/ctokheim/c3ab56a4db7311487761 [citováno 25. 4. 2020].