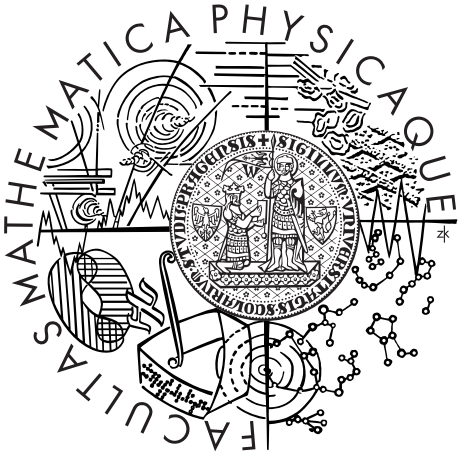
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Aleh Masaila

Regresní stromy

Katedra pravděpodobnosti a matematické statistiky

Vedoucí diplomové práce: Mgr. Tomáš Hanzák
Studijní program: Matematika
Studijní obor: Finanční a pojistná matematika

Praha 2013

Chtěl bych poděkovat svému vedoucímu Mgr.Tomášovi Hanzákovi za odborné vedení a celkovou pomoc při psání diplomové práce, a také doc. RNDr. Karlu Zvárovi, CSc. za cenné komentáře k této práci. Dále bych chtěl poděkovat své rodině a partnerce za podporu během mého studia.

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

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

V Praze dne ..... Podpis autora

Název práce: Regresní stromy

Autor: Aleh Masaila

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí diplomové práce: Mgr. Tomáš Hanzák

Abstrakt:

Ačkoliv regresní a klasifikační stromy se používají k analýze dat již několik desítek let, stále jsou ve stínu tradičnějších metod, jako jsou například lineární nebo logistická regrese. Tato práce si klade za cíl popsat několik nejznámější regresních stromů a zároveň přiblížit relativně nový směr v této oblasti - kombinací regresních stromů a komisních metod, tzv. regresní lesy. Součástí práce je i praktická část, kde vyzkoušíme vlastnosti, silné a slabé stránky zkoumaných metod na reálných datech.

Klíčová slova: regresní strom, CART, MARS, regresní les, bagging, boosting, náhodný les

Title: Regression trees

Author: Aleh Masaila

Department: Department of Probability and Mathematical Statistics

Supervisor: Mgr.Tomáš Hanzák

Abstract: Although regression and classification trees are used for data analysis for several decades, they are still in the shadow of more traditional methods such as linear or logistic regression. This paper aims to describe a couple of the most famous regression trees and introduce a new direction in this area - a combination of regression trees and committee methods, so called the regression forests. There is a practical part of work where we try properties, strengths and weaknesses of the examined methods on real data sets.

Keywords: regression tree, CART, MARS, regression forest, bagging, boosting, random forest

Obsah

1. Úvod

Pro mnoho vědních disciplín je běžný následující problém: vhodným způsobem aproximovat funkci několika proměnných, danou pouze jejími vstupy a výstupy. Neboli modelovat závislost náhodné veličiny Y na jedné či více proměnných, a to na základě dostupných dat $(y_i, x_{i1}, \ldots, x_{iJ}),\ i = 1, 2, \ldots, N$.

Označme

$$Y = f(X_1, \ldots, X_J) + \epsilon, \quad (X_1, \ldots, X_J) \in \mathcal{D},\ \mathcal{D} \subset \mathbb{R}^J \tag{1.1}$$

jako skutečný vztah mezi závislou, nebo také vysvětlovanou, a nezávislými, nebo také vysvětlujícími, proměnnými. Náhodná veličina ϵ odráží závislost Y na takových veličinách, které nebyly nebo ani nemohly být pozorovány. Naším cílem je sestavit $\hat{f}(X_1, \ldots, X_J)$, která bude co nejlépe aproximovat skutečnou funkci $f(X_1, \ldots, X_J)$.

Výsledná funkce $\hat{f}$ by měla mít následující vlastnosti:

- srozumitelnost - mezi závislou a nezávislou proměnnou by měl být jasný vztah
- přesnost - předpovědi funkce $\hat{f}$, označíme je jako $\hat{y}$, by měly být dostatečně blízko skutečným hodnotám y
- snadná počitatelnost - míněno počet operací, které je potřeba provést, abychom získali výsledný model

To, jakou z výše uvedených vlastností budeme upřednostňovat, záleží na konkrétní úloze. Především, zda chceme:

- předpovědět hodnotu závislé proměnné na základě známých hodnot
- pochopit vztah mezi závislými a nezávislými proměnnými

Existuje řada statistických nástrojů, pomocí kterých řešíme tento problém, například lineární a logistická regrese, diskriminační analýza atp. Mezi tyto nástroje patří i *regresní stromy*, na které je zaměřená tato práce.

Struktura práce je následující. Ve druhé kapitole popíšeme, co to jsou regresní stromy, a zavedeme potřebné definice, které s tím souvisí. Dále zde stručně popíšeme pravděpodobně nejrozšířenější metodu pro analýzu dat, lineární regresi. Děláme to z toho důvodu, že v páté kapitole budeme na reálných datech porovnávat predikční schopnosti regresních stromů a lineární regrese. Třetí kapitola je věnována popisu několika vybraných regresních stromů. Čtvrtá kapitola se zaměřuje na relativně nové techniky, které se nazývají *regresní lesy*. Tyto techniky vycházejí z regresních stromů a snaží se odstranit některé z jejích nedostatků. Pátá kapitola je věnována aplikaci metod na různých reálných datech. Šestá kapitola obsahuje závěr, kde shrneme výsledky našeho zkoumání.

2. Metody analýzy dat

Myšlenka analýzy dat pomocí stromových metod je v zásadě jednoduchá. Tyto metody rozdělí definiční množinu nezávislých proměnných $\boldsymbol{X} \in \mathcal{D},\ \mathcal{D} \subset \mathbb{R}^J$ na jednotlivé podmnožiny $R_m,\ m = 1, 2, \ldots, M$ tak, že $\mathcal{D} = \bigcup_{m=1}^{M} R_m$. Takto vzniklé podmnožiny mohou, ale nemusí být disjunktní. Následně je ke každé podmnožinám přiřazená nějaká jednoduchá funkce, například konstanta.

Stromové metody lze použít jak na řešení situací, kdy je vysvětlovaná proměnná Y spojitá, tak i situaci, kdy vysvětlovaná proměnná nabývá hodnoty z množiny $C = \{c_1, c_2, \ldots, c_p\}$. První typ stromů se nazývá *regresní*, druhý *klasifikační*. Většina technik je však použitelná pro oba dva typy úloh.

V současné době existuje velké množství algoritmů, které se navzájem liší především v tom, jakým způsobem rozdělí prostor $\mathcal{D}$ na jednotlivé podmnožiny R_m. Několik vybraných technik bude popsáno v následujících dvou kapitolách.

Pátá kapitola je věnována prezentaci výsledků praktického použití těchto metod na reálných datech, a zároveň ke srovnání s rozšířenějšími metodami, konkrétně s lineární regresí. Proto tato kapitola obsahuje i stručný popis této metody. Informačními zdroji zde jsou [1], [6] a [18].

2.1 Stromové metody

Nejdříve zavedeme pojem stromu:

Definice 2.1 *Regresní strom je funkce, která každému reálnému vektoru* $\boldsymbol{X} = (X_1, \ldots, X_J),\ \boldsymbol{X} \in \mathcal{D},\ \mathcal{D} \subset \mathbb{R}^J$ *přiřadí hodnotu* $Y \in \mathbb{R}$*, a to pomocí následujícího předpisu*

$$T(\boldsymbol{X}) = \sum_{m=1}^{M} \beta_m h_m(\boldsymbol{X}), \tag{2.1}$$

kde $\beta_m,\ m = 1, \ldots, M$ *jsou koeficienty, a* h_m *je předpis, který přiřadí vektoru vysvětlujících proměnných* $\boldsymbol{X}$ *nějakou jednoduchou funkci, například konstantu, pokud* $\boldsymbol{X} \in R_m$*, a 0, pokud* $\boldsymbol{X} \notin R_m,\ R_m \subset \mathcal{D}$*.*

Pokud náhodná veličina Y nabývá hodnot jenom z množiny $C = \{c_1, c_2, \ldots, c_p\}$, pak takový strom nazýváme *klasifikační*.

Jednotlivé stromové techniky se od sebe liší především dvěma věcmi. Způsobem, jakým se rozdělí $\mathcal{D}$ na jednotlivé podmnožiny R_m, a tvarem funkce h_m, kterou těmto podmnožinám přiřadí.

Na strom můžeme pohlížet ze dvou hledisek. Pomocí $T(\boldsymbol{X})$ označujeme strom jako funkci vektoru vysvětlujících proměnných $\boldsymbol{X}$ ve smyslu definice (2.1).

Dále lze strom T chápat jako soubor množin $\{t_m\}_1^{M'}$, kde t_m obsahuje část ze souboru pozorování $\mathcal{L} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N\}$. Specifické postavení má množina, která obsahuje všechna pozorování z $\mathcal{L}$. Takové množině říkáme *kořen stromu* (*root*) a značme ji t_1.

Pokud existují dvě takové množiny $t_l,\ t_r$, pro které platí

$$t_m = t_l \cup t_r \text{ a } t_l \cap t_r = \emptyset,$$

pak tyto dvě množiny nazýváme *levý* a *pravý následník* uzlu t_m.

Obecně ale jeden uzel nemusí mít jenom dva následníky. V takovém případě bychom definovali více následníků analogický. Uzly $t_a,\ t_b$ a t_c jsou následníky uzlu t_m pokud platí $t_m = t_a \cup t_b \cup t_c$, a $t_a,\ t_b$ a t_c jsou navzájem disjunktní množiny. Počet následníků se pro různé algoritmy může lišit, nicméně nejobvyklejší počet následníků jsou dva.

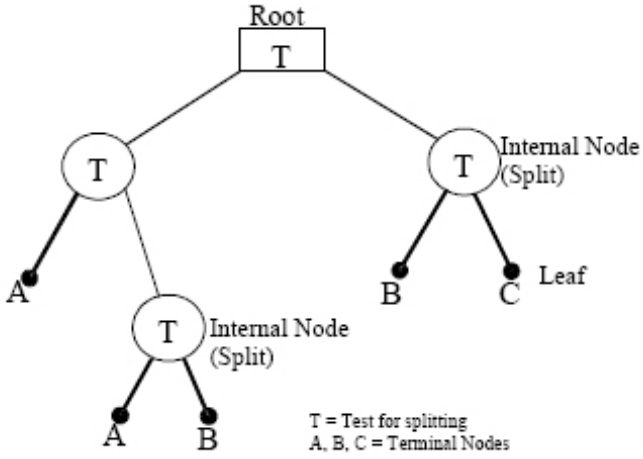
Pokud uzel t nemá žádného následníka, pak takovému uzlu říkáme *konečný uzel* (*terminal node*) nebo také *list*. Symbolem $\tilde{T}$ budeme značit množinu všech konečných uzlů stromu T a písmenem M počet prvků této množiny, neboli počet listů. Počet listů je stejný, jako počet množin R_m, které dělí definiční prostor $\mathcal{D}$. Pozorování z nějakého listu t_m spadají do podmnožiny $R_m \subset \mathcal{D}$.

Pokud budeme mluvit o T, pak tím rozumíme konkrétní strom, který byl vznikl za použití souboru pozorování $\mathcal{L}$. Obdobně t označuje konkrétní uzel, ve kterém se nachází část pozorování z $\mathcal{L}$.

Pro další použití ještě zavedeme následující pojem.

Definice 2.2 *Označme T_t takovou podmnožinu stromu T, která obsahuje uzel t a všechny jeho následníky až do úrovně listů. Takové podmnožině říkáme větev stromu T.*

Důvod, proč se těmto metodám říká stromové, je ten, že výsledný model můžeme snadno zobrazit pomocí přehledného grafu následujícího typu:



Obrázek 2.1: Rozhodovací strom (zdroj [9]).

Jak je vidět, výsledný model svým vzhledem připomíná obracený strom, se všemi příslušnými atributy, které byly výše popsané. První uzel nahoře je kořen, který obsahuje všechny pozorování. Ten se pak dále dělí na jednotlivé uzly, které jsou na konci ukončeny listy, v nichž jsou pozorováním, kterým je přiřazená nějaká hodnota pomocí funkce h_m.

2.2 Lineární regrese

Model lineární regrese je dobře znám, proto ho zde popíšeme jenom velice stručně. Pro další podrobnosti je zde [18] nebo [1].

Nechť máme náhodné veličiny $Y_1, \ldots, Y_N$ a matici daných čísel $\mathbb{X} = \{x_{ij}\}$ typu $N \times J$, kde $J < N$. Předpokládejme, že pro náhodný vektor $\boldsymbol{Y} = (Y_1, \ldots, Y_N)'$ platí

$$\boldsymbol{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{e},$$

kde $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)'$ je vektor neznámých parametrů a $\boldsymbol{e} = (e_1, \ldots, e_N)'$ je náhodný vektor. O náhodném vektoru $\boldsymbol{e}$ předpokládáme, že jeho střední hodnota a rozptyl splňují následující předpoklady: $\mathsf{E}\boldsymbol{e} = \boldsymbol{0}$, $\mathsf{var}\boldsymbol{e} = \sigma^2\boldsymbol{I}$. Takovému modelu říkáme *regresní*, a protože závislost $\boldsymbol{Y}$ na $\boldsymbol{\beta}$ je lineární, mluvíme o *lineární regresi*.

Typickým předpokladem je, že matice $\mathbb{X}$ má lineární nezávislé sloupce, a protože předpokládáme $J < N$, je hodnost matice $h(\mathbb{X}) = J$.

Parametry $\boldsymbol{\beta}$ se odhadují metodou nejmenších čtverců, tj. chceme minimalizovat následující výraz vzhledem ke koeficientům $\boldsymbol{\beta}$:

$$(\boldsymbol{Y} - \mathbb{X}\boldsymbol{\beta})'(\boldsymbol{Y} - \mathbb{X}\boldsymbol{\beta}) \tag{2.2}$$

Odhady koeficientů $\boldsymbol{\beta}$ značíme $\boldsymbol{b} = (b_1, \ldots\ b_J)'$ a z (2.2) snadno získáme tento odhad, který se rovná $\boldsymbol{b} = (\mathbb{X}'\mathbb{X})^{-1}\mathbb{X}'\boldsymbol{Y}$.

Soustavě lineárních rovnic $\mathbb{X}'\mathbb{X}\boldsymbol{b} = \mathbb{X}'\boldsymbol{Y}$, ze které se počítá ; $\boldsymbol{b}$, se říká *soustava normálních rovnic*. Vektor $\hat{\boldsymbol{Y}} = \mathbb{X}\boldsymbol{b} = \mathbb{X}(\mathbb{X}'\mathbb{X})^{-1}\mathbb{X}'\boldsymbol{Y}$ je nejlepší aproximace vektoru $\boldsymbol{Y}$, jaká může být vytvořená pomocí lineární kombinací sloupců dané matice $\mathbb{X}$.

Definujeme dále *reziduální součet čtverců* a *celkový součet čtverců*:

$$RSS = (\boldsymbol{Y} - \hat{\boldsymbol{Y}})'(\boldsymbol{Y} - \hat{\boldsymbol{Y}}) \tag{2.3}$$

$$TSS = (\boldsymbol{Y} - \bar{\boldsymbol{Y}})'(\boldsymbol{Y} - \bar{\boldsymbol{Y}}). \tag{2.4}$$

kde $\bar{\boldsymbol{Y}} = \bar{Y}\boldsymbol{1}$, a $\bar{Y}$ je průměr. Pomocí těchto dvou součtů definujeme *koeficient determinace* R^2:

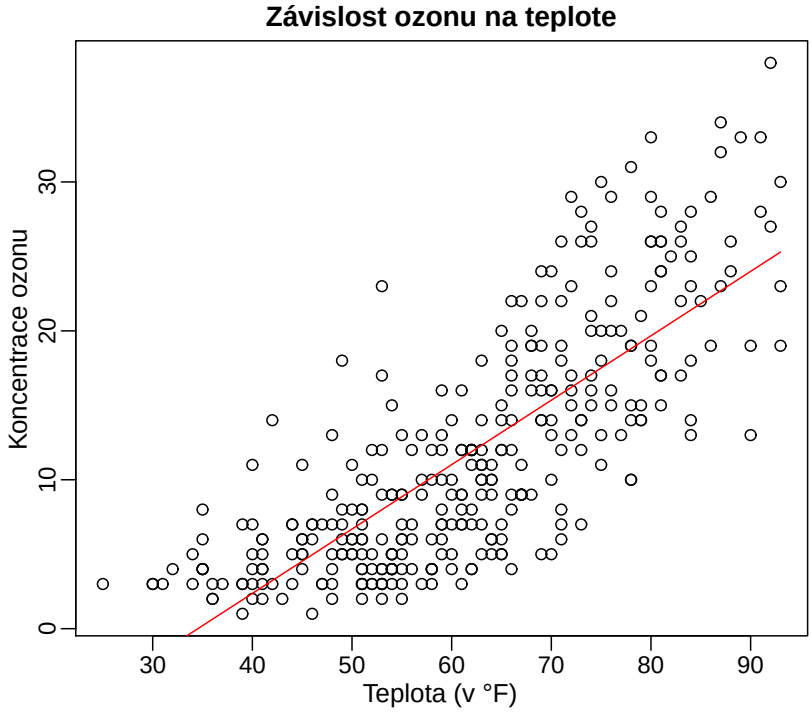
$$R^2 = 1 - \frac{RSS}{TSS} \tag{2.5}$$

Koeficient determinace ukazuje, jak velkou část výchozí variability hodnot se nám podařilo vysvětlit pomocí uvažované závislosti. Dá se také chápat jako míra zlepšení kvality predikce závislé proměnné oproti použití prostého průměru. Koeficient determinace nám může také sloužit k porovnání predikčních schopností různých modelů.

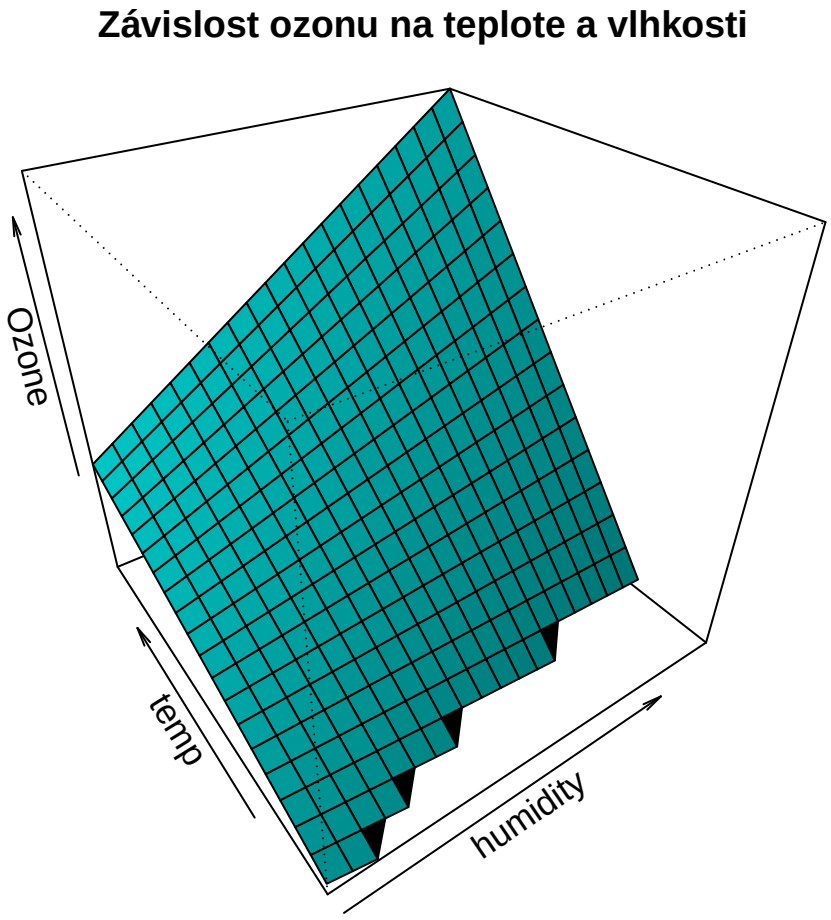
Pokud ale chceme používat koeficient determinace vypočtený dle vzorce (2.5), je nutné, aby byla splněná následující podmínka, a to, že první sloupec matice $\mathbb{X}$ je tvořen jedničkami, nebo aby vektor $\boldsymbol{1}$ patřil do lineárního obalu $\mathcal{M}(\mathbb{X})$.

Model, sestrojený za použitím všech sloupců matice $\mathbb{X}$, často nebývá tím nejjednodušším, a abychom tento model zjednodušili, lze použit *t-test*. Pomocí tohoto testu ověřujeme hypotézu $H_0 : \beta_j = \beta_j^0$, kde $j \in \{1, \ldots, J\}$. Pokud zvolíme $\beta_j^0 = 0$, testujeme, zda $\boldsymbol{Y}$ závisí na j-tým sloupci, nebo ho můžeme vypustit a přejít k jednoduššímu modelu.

Příklad 2.3 *Abychom lépe ukázali rozdíly mezi jednotlivými technikami, budeme ilustrovat použití jednotlivých metod na datech Ozone1 z balíčku Earth programu R project. Jedná se o data vztahující se k měření koncentrace ozónu v Los Angeles a pocházejí z [5]. Konkrétně budeme ilustrovat to, jak odhadují jednotlivé techniky závislost koncentrace ozónu na teplotě, respektive teplotě a vlhkosti vzduchu. Tento příklad neslouží ke srovnávání kvality modelů, ale pouze jako ukázka odlišných tvarů výsledných funkcí. S pokračováním příkladu se setkáme v dalších kapitolách.*



Obrázek 2.2: Na tomto grafu můžeme vidět regresní přímku proloženou naměřenými hodnotami koncentrace ozónu.



Obrázek 2.3: Odhadovaná závislost koncentrace ozónu při použití dvou nezávislých proměnných, konkrétně teploty a vlhkosti.

3. Regresní stromy

V této kapitole popíšeme dva stromy. Začneme s pravděpodobně nejznámějším, se stromem CART (*Classification and Regression Tree*), který je podrobně popsaný v [6]. Dále bude následovat MARS (*Multivariate Addaptive Regression Splines*), kde budeme vycházet především z [8]. Stručný přehled všech těchto technik lze najít také v [11].

3.1 CART

Metoda CART byla vyvinuta Breimanem a spol., a je podrobně popsána v již zmíněné publikaci [6]. Je použitelná jak při řešení regresních, tak i klasifikačních úloh. Postup je v obou případech obdobný. Tato práce je zaměřená na regresní stromy, proto podkapitola věnující se metodě CART bude popsána z regresního hlediska. Začneme s popisem algoritmu, pomocí kterého obdržíme výsledný model.

3.1.1 Algoritmus CART

Nechť máme k dispozici množinu pozorování $\mathcal{L} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N\}$, kde $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iJ})$ je vektor vysvětlujících proměnných a y_i je vysvětlovaná proměnná. Dále budeme značit $\mathbb{X}$ matici pozorování $\{x_{ij},\ i = 1, \ldots, N,\ j = 1, \ldots, J\}$, $\boldsymbol{y}$ vektor výstupů $(y_1, \ldots, y_N)$ a $\boldsymbol{x}$ bude označovat obecný vektor pozorování $(x_{\bullet 1}, \ldots, x_{\bullet J})$. Symbolem X_j budeme značit j-tou vysvětlující proměnnou.

Algoritmus CART se skládá ze čtyř kroků:

1.) Hledání nejlepšího dělení pro každý prediktor - v každém uzlu t se pro každou vysvětlující proměnnou $X_j,\ j = 1, \ldots, J$ najde nejlepší dělení. Dělení uzlu t je proces, při kterém pozorování z uzlu t se rozdělí do dvou skupin, a dají tak vzniknout následníkům uzlu t, uzlům t_l a t_r.

2.) Výběr nejlepšího dělení daného uzlu t - mezi nejlepšími děleními pro jednotlivé vysvětlující proměnné, které jsme našli v bodě jedna, vybereme to, které nejvíce snižuje nepřesnost stromu.

3.) Dělení uzlu - pozorování z uzlu rozdělíme do dvou následníků dle dělení vybraného v Kroku 2.

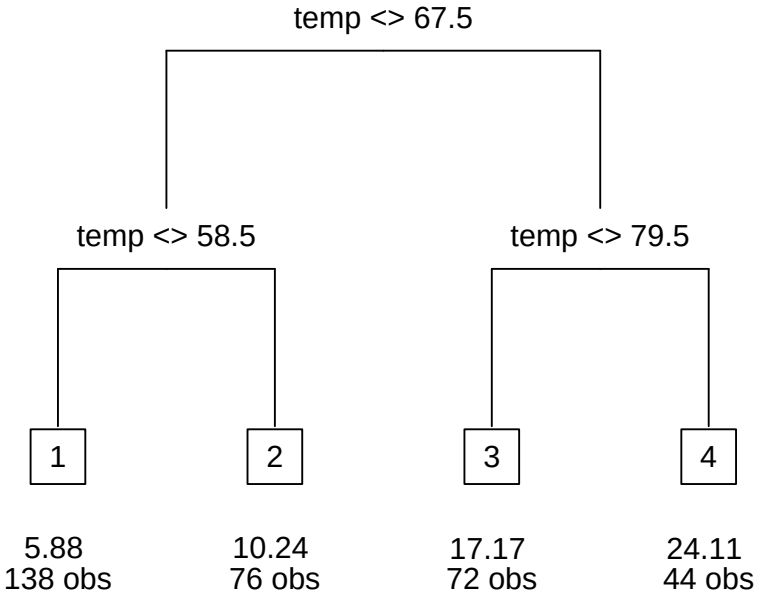
4.) Přiřazení hodnoty - Kroky 1-3 se opakují, dokud uzel není označen jako konečný, tj. z uzlu se stává list. Až se to stane, algoritmus přiřadí danému listu nějakou hodnotu. Algoritmus končí, když jsou všechny uzly označené jako listy.

Výsledkem je model následujícího tvaru:

$$T(\boldsymbol{x}) = \sum_{m=1}^{M_{max}} \beta_m I(\boldsymbol{x} \in R_m), \tag{3.1}$$

kde M_{max} je počet listů ve stromu.

Příklad 3.1 *Velkou výhodou stromů, a metody CART především, je jejich přehlednost a relativně snadná interpretace. Následující obrázek vznikl, když jsme pomocí metody CART modelovaly vztah koncentrace ozónu v atmosféře na teplotě.*



Obrázek 3.1: Klasický obrázek stromu, který modeluje závislost koncentrace ozónu v atmosféře na teplotě ve stupních Fahrenheita. Na začátku algoritmu kořen stromu obsahuje všech 330 pozorování. Opakováním kroků 2.) a 3.) algoritmus postupně najde nejlepší dělení jednotlivých uzlů. Například nejlepší dělení kořene je: pokud $x \leq 67,5$, pak pozorování jde do levého následníka, jinak do pravého. Jakmile se uzel dále nedělí, tak je mu přiřazená nějaká hodnota. Například listu číslo 1, tj. pozorováním, jejichž teplota je menší než 58,5 °F, je přiřazená hodnota 5,88.

Z výše uvedeného algoritmu plyne několik otázek:

1. Jaká hodnota je přiřazená listům?
2. Jak je definováno nejlepší dělení?
3. Jak se určí, zda je uzel konečný?

Než na tyto otázky odpovíme, musíme zavést několik důležitých pojmů. Začneme s tím, že si definujeme *nepřesnost stromu*. Nepřesnost stromu je míra toho, jak dobře odhadnuté hodnoty $\hat{\boldsymbol{y}} = T(\mathbb{X})$ se shodují se skutečnými pozorovanými hodnotami $\boldsymbol{y}$. Budeme ji značit symbolem $L(T)$. Strom se snažíme sestrojit tak, aby jeho nepřesnost byla co nejmenší.

V případě regresních stromů se nejčastěji nepřesnost počítá pomocí střední čtvercové chyby, respektive pomocí průměrné čtvercové chyby v případě konečného souboru pozorování. Lze ale použít i jiné míry nepřesnosti, například absolutní odchylku. Střední čtvercová chyba pro strom $T(\boldsymbol{X})$ je definována jako

$$MSE(T) = \mathsf{E}(Y - T(\boldsymbol{X}))^2. \tag{3.2}$$

Platí následující tvrzení (viz [1], s 60)

Tvrzení 3.2 *Budiž Y náhodná veličina a $\boldsymbol{X}$ náhodný vektor. Nechť $\mathsf{E}Y^2 < \infty$ a nechť rozdělení vektoru $(Y, \boldsymbol{X}')$ má hustotu vzhledem k nějaké σ-konečné míře $\lambda \times \nu$. Potom pro každou měřitelnou funkci $f(\boldsymbol{X})$ platí*

$$\mathsf{E}\left[Y - f(\boldsymbol{X})\right]^2 \geq \mathsf{E}\left[Y - \mathsf{E}(Y|\boldsymbol{X})\right]^2 \tag{3.3}$$

Rovnost ve vzorci (3.3) nastává pravě tehdy, když platí $f(\boldsymbol{X}) = \mathsf{E}(Y|\boldsymbol{X})$ s pravděpodobnosti jedna.

Toto tvrzení dává odpověď na první otázku. Aby nepřesnost stromu T byla nejmenší, pak strom T musí přiřadit náhodnému vektoru $\boldsymbol{X}$ podmíněnou střední hodnotu $\mathsf{E}(Y|\boldsymbol{X} = \boldsymbol{x})$.

Tyto i většina dalších tvrzení budou zde uváděná bez důkazů, všechny tyto rozšiřující informace lze nalézt ve zmiňovaných informačních zdrojích.

Pokud zvolíme za nepřesnost stromu střední čtvercovou chybu, tak $L(T)$ počítáme jako

$$L(T) = \frac{1}{N}\sum_{i=1}^{N}(y_i - T(\boldsymbol{x}_i))^2 \tag{3.4}$$

Protože pro $T(\boldsymbol{x})$ předpokládáme platnost (3.1), dostáváme:

$$L(T) = \frac{1}{N}\sum_{m=1}^{M_{max}}\sum_{(y_i:\ \boldsymbol{x}_i \in R_m)}(y_i - \beta_m)^2 \tag{3.5}$$

Nyní potřebujeme odhadnout, čemu se budou rovnat parametry β_m. Víme, že pro každou konstantu a platí:

Tvrzení 3.3 *Pokud nějaké* a *minimalizuje výraz $E(Y-a)^2$, pak $a = E(Y)$*

S pomocí tohoto tvrzení již snadno odpovíme na první otázku, a to jakou hodnotu přiřadí metoda CART listům stromu.

Tvrzení 3.4 *Odhad koeficientu β_m, který minimalizuje (3.5), je roven průměru hodnot závislých proměnných y_i patřícím pozorováním $\boldsymbol{x}_i$, které spadají do R_m. Označme odhad β_m symbolem b_m, pak platí*

$$b_m = \frac{1}{N_m}\sum_{(y_i:\ \boldsymbol{x}_i \in R_m)} y_i, \tag{3.6}$$

kde se sčítá přes všechny y_i takové, že $\boldsymbol{x}_i \in R_m$ a N_m je celkový počet pozorování v R_m, $m = 1, 2, \ldots, M_{max}$.

Toto tvrzení snadno dokážeme, pokud výraz (3.5) zderivujeme a položíme rovný nule.

Nyní přejdeme ke druhé otázce, a to jak najdeme nejlepší dělení (*split*) uzlu. Přesný proces dělení uzlu záleží na typu vysvětlující proměnné. Pokud je vysvětlující proměnná X_j spojitá nebo ordinální, pak je postup následující. Všechny unikátní hodnoty $(x_{1j}, \ldots, x_{Nj})$ se seřadí dle velikosti a postupně se vyzkouší všechny možnosti, jak rozdělit pozorování do dvou uzlů. Obecně, uzel, který není konečný, může být rozdělen na víc, než dva následníky, avšak větší počet následníků nepřináší žádné významnější zlepšení. Postupně vytvoříme dělení ve tvaru: pokud $x_{ij} \leq r$, pak dané i-té pozorování jde do levého následníka, jinak do pravého. Za hodnotu r se postupně dosazují všechny unikátní pozorované hodnoty $(x_{1j}, \ldots, x_{Nj})$.

Takto nám vznikne množina možných bodů dělení $\mathcal{S}$ uzlu t pro proměnnou X_j, ze které vybíráme nejlepší dělení.

Definice 3.5 *Nejlepší dělení s* uzlu t je takové dělení $s \in \mathcal{S}$, které nejvíce sníží nepřesnost stromu $L(T)$.*

Přesněji řečeno, pro každé dělení s uzlu t na uzly t_l, t_r nechť

$$\Delta L(s,t) = L(t) - L(t_l) - L(t_r).$$

Jako nejlepší dělení s^* je označeno takové dělení, pro které platí

$$\Delta L(s^*,t) = \max_{s \in S} \Delta L(s,t).$$

V případě nominálních vysvětlujících proměnných je situace o něco složitější. Pokud nominální proměnná X_j nabývá hodnot z množiny $C = \{c_1, \ldots, c_q\}$, jedno z řešení je vytvořit 2^{q-1} možných množin $\mathcal{A}$, a zkoušet dělení typu: pokud $x_{ij} \in \mathcal{A}$, pak pozorování jde do levého následníka, jinak do pravého.

Tento postup je ale pro větší počet kategorií nevhodný, neboť znamená velké nároky na výpočetní čas. Proto se používá následující heuristika. Nechť vysvětlující proměnná X_j je nominální, s hodnotami v množině C. Pak označíme $\bar{y}(c_l)$ jako průměr všech hodnot y_i, pro které platí $x_{ij} = c_l,\ l = 1, \ldots, q$. Tyto průměry seřadíme dle velikosti do posloupnosti

$$\bar{y}(c_{l_1}) \leq \bar{y}(c_{l_2}) \leq \ldots \leq \bar{y}(c_{l_q}).$$

Pak platí následující tvrzení:

Tvrzení 3.6 *Nejlepší dělení vysvětlující proměnné X_j v uzlu t je jedno z $q-1$ dělení*

$$x_j \in \{c_{l_1}, \ldots, c_{l_h}\},\ h = 1, \ldots, q-1$$

Tento postup zkrátí počet možných podmnožin z 2^{q-1} na $q-1$.

Postupně v daném uzlu vytvoříme množinu nejlepších dělení pro jednotlivé prediktory. Za nejlepší dělení uzlu t pak označíme to dělení, které nejvíce snižuje nepřesnost stromu.

Zbývá nám odpovědět na třetí otázku, jakým způsobem zastavit dělení stromu. Rozhodnutí, zda je daný uzel konečný nebo se bude i nadále dělit, závisí

na splnění zastavovacích kriteriích. Zastavovací kritérium je podmínka, po jejíž splnění je uzel prohlášen za konečný a dále se již nedělí. Bez těchto pravidel by dělení pokračovalo, dokud by v každém listu zbylo pouze jedno pozorování. Zastavovací kriterium tak významným způsobem ovlivňují velikost, a tedy i nepřesnost stromu. Uzel se prohlásí za konečný, pokud platí například:

- Všechna pozorování v uzlu mají shodné hodnoty všech vysvětlujících proměnných
- Strom dosáhne předem zvolené velikosti, tj. počet listů se rovná námi vybrané M
- Počet pozorování v uzlu je menší, než předem zvolený počet
- Následníci uzlu by měly menší počet pozorování, než požadované minimum
- Pokud pro nejlepší možné dělení s^* uzlu t nedosáhne zlepšení $\Delta L(s^*, t)$ požadovaného minima
- Uzel se stává čistým, neboli pozorování v něm mají stejnou hodnotu závislé proměnné (toto pravidlo je použitelné pouze pro klasifikační stromy)

3.1.2 Prořezávání stromu CART

V předchozí části jsme vyjmenovali několik zastavovacích kriterií, které určují velikost stromu. Nabízí se otázka, jak přesně definovat zastavovací kritéria, aby výsledný strom měl optimální velikostí a dobře aproximoval vztah mezi vysvětlujícími a vysvětlovanou proměnnými. Pokud by byly zastavovací kritéria nastavená přísně, výsledný strom bude malý, a tak nedokáže věrně zobrazit vztahy mezi vysvětlujícími a vysvětlovanou proměnnou. Na druhou stranu, pokud zastavovací kritéria budou mírná, vznikne příliš velký strom. Velký strom bývá přeučený (*overfitted*), tj. bude dobře použitelný jenom na data, pomocí kterých byl vytvořen.

Metoda CART řeší tento problém pomocí *prořezávání* (*pruning*). Prořezávání stromu je proces, kdy se postupně ze stromu odřezávají jednotlivé větve stromu.

Definice 3.7 *Odřezání větve T_t ze stromu T rozumíme odstranění z T veškerých následníků uzlu t. Samotný uzel t zůstává součástí stromu. Nově vzniklý strom značíme $T - T_t$.*

Budeme hledat následující posloupnost

$$T_{max} \supseteq T_1 \supset T_2 \supset T_3 \supset \ldots \supset t_1, \tag{3.7}$$

ze které vybereme optimální strom. Jedná se o posloupnost do sebe vnořených stromů, která začíná stromem T_{max} a končí kořenem t_1.

Abychom vybrali strom o správné velikosti, musíme upravit dříve zavedenou definici nepřesnosti stromů. Pokud pořád budeme brát jako kritérium kvality výraz (3.5), pak jako nejlepší strom bude označen T_{max}, protože jeho nepřesnost $L(T)$ je nejmenší. Strom má totiž následující vlastnost:

Tvrzení 3.8 *Pro libovolný uzel t jeho následníky t_l, t_r platí*

$$L(t) \geq L(t_l) + L(t_r).$$

Předchozí tvrzení říká, že libovolné dělení uzlu t nezhorší nepřesnost stromu. Tedy čím víc se strom bude dělit, tím bude jeho nepřesnost menší. Zavedeme proto nové měření nepřesnosti stromu, které zohledňuje i jeho velikost.

Definice 3.9 *Označme počet listů stromu T symbolem $|\tilde{T}|$. Reálné číslo $\alpha \geq 0$ nazveme parametr složitosti. Pak definujeme nepřesnost stromu s ohledem na jeho složitost $L_\alpha(T)$ jako*

$$L_\alpha(T) = L(T) + \alpha|\tilde{T}|. \tag{3.8}$$

Parametr α se dá chápat jako penalizace za jeden list stromu.

Parametr α nám pomůže najít požadovanou posloupnost (3.7). Hlavní myšlenkou při hledání posloupnosti je to, že pro každé α najdeme strom $T(\alpha) \subseteq T_{max}$, kdy $T(\alpha)$ je takový strom, jehož nepřesnost $L_\alpha(T)$ je nejmenší:

$$L_\alpha(T(\alpha)) = \min_{T \subseteq T_{max}} L_\alpha(T).$$

Ačkoliv hodnota α je spojitá, existuje jenom konečný počet podstromů stromu T_{max}. To znamená, že pro dané α je $T(\alpha)$ nejlepším stromem až do chvíle, kdy α dosáhne bodu α'. Pak se nejlepším stromem stane $T(\alpha')$.

Potřebujeme vyřešit dvě otázky, zda pro každé α existuje pravě jeden strom $T(\alpha) \subseteq T_{max}$, který minimalizuje $L_\alpha(T)$, a zda posloupnost minimalizujících stromů do sebe zapadá, tj. zda podstrom T_{k+1} dostanu ze stromu T_k pomocí odříznutí nějaké větve.

Problém jedinečnosti takového stromu je řešen pomocí vhodné definice.

Definice 3.10 *Strom $T(\alpha)$ je nejmenší strom, pokud pro jeho nepřesnost definovanou předpisem (3.8) platí*

(i) $L_\alpha(T(\alpha)) = \min_{T \subseteq T_{max}} L_\alpha(T)$

(ii) Pokud $L_\alpha(T) = L_\alpha(T(\alpha))$, *pak* $T(\alpha) \subseteq T$.

Je zřejmé, že pokud strom $T(\alpha)$ existuje, musí být jedinečný. To, že takový strom $T(\alpha)$ skutečně existuje, je dokázáno v [6] (s 284-287).

Nyní začneme s popisem algoritmu, který nám vytvoří posloupnost (3.7). Prořezávat nezačínáme strom s maximálním počtem uzlů T_{max}, ale strom T_1, pro který platí $L(T_1) = L(T_{max})$. Z T_{max} dostaneme T_1 tak, že odřízneme veškeré následníky uzlů $t \in \tilde{T}$, pro které platí $L(t) = L(t_l) + L(t_r)$.

Pro každou větev T_t definujme nepřesnost větve jako

$$L(T_t) = \sum_{t' \in \tilde{T}_t} L(t'),$$

kde $\tilde{T}_t$ je množina listů větve uzlu t. Platí $L(t) > L(T_t)$, kde $L(t)$ je nepřesnost uzlu t.

Komplexní nepřesnost uzlu t definujeme jako

$$L_\alpha(t) = L(t) + \alpha,$$

a komplexní nepřesnost větve s ohledem na její složitost se rovná

$$L_\alpha(T_t) = L(T_t) + \alpha|\tilde{T}_t|.$$

Dokud platí

$$L_\alpha(T_t) < L_\alpha(t),$$

pak má větev nižší nepřesnost než samotný uzel. Ale s rostoucí hodnotou α se z nerovnosti stane rovnost a to znamená, že nepřesnost větve je stejná jako nepřesnost uzlu, a tedy tuto větev lze ze stromu odstranit, aniž by se zvýšila jeho nepřesnost. Abychom našli vhodné α potřebujeme vyřešit následující nerovnost:

$$\alpha < \frac{L(t) - L(T_t)}{|\tilde{T}_t| - 1} \tag{3.9}$$

Toto využijeme při popisu algoritmu *odříznutí nejslabšího článku* (*weakest-link cutting*). Definujeme funkci $g_1(t), t \in T_1$ pomocí předpisu

$$g_1(t) = \begin{cases} \frac{L(t)-L(T_t)}{|\tilde{T}_t|-1} & t \notin \tilde{T}_1 \\ +\infty & t \in \tilde{T}_1 \end{cases}$$

Uzel $\bar{t}_1 \in T_1$ nazveme nejslabším, pokud pro něho platí $g_1(\bar{t}_1) = \min_{t \in T_1} g_1(t)$. Položíme $\alpha_2 = g_1(\bar{t}_1)$. Nejslabší uzel je myšlen ve smyslu, že s rostoucím α se nepřesnost větve tohoto uzlu bude jako první rovnat nepřesnosti samotného uzlu.

Definujeme nový strom T_2 jako $T_2 = T_1 - T_{\bar{t}_1}$, přičemž samotný uzel $\bar{t}_1$ zůstává součástí stromu, a zopakujeme celý postup s tím, že místo T_1 použijeme T_2.

Postupně dostaneme hledanou posloupnost (3.7).

Spojení mezi výše uvedeným algoritmem a prořezáváním s využitím nepřesnosti stromu definovanou předpisem (3.8) je dána následující větou:

Věta 3.11 *Nechť* $\{\alpha_k\}$ *je rostoucí posloupnost,* $k \geq 1$, *kde* $\alpha_1 = 0$. *Pak pro libovolné* k *a libovolné* α *takové, že* $\alpha_k \leq \alpha < \alpha_{k+1}$, *platí* $T(\alpha) = T(\alpha_k) = T_k$

Tato věta říká, jak prořezávání na základě celkové nepřesnosti pracuje. Začíná to se stromem T_1, najde větev nejslabšího spojení $T_{\bar{t}_1}$, a odřízne větev, když α dosáhne hodnoty α_2. Takto prořezáváme až do okamžiku, kdy obdržíme t_1.

Tímto způsob dostáváme konečnou posloupnost do sebe vnořených stromů

$$T_1 \supset T_2 \supset T_3 \supset \ldots \supset \ t_1,$$

kde $T_k = T(\alpha_k)$. Nyní z této posloupnosti potřebujeme vybrat strom o optimální velikosti.

Používají se dva způsoby, jak vybrat optimální strom. V obou dvou případech odhadujeme nepřesnost jednotlivých stromů v posloupnosti (3.7). Jedná

se o odhad pomocí *testovacího vzorku* (*test sample*) a pomocí *křížové validace* (*cross-validation*).

Odhad pomocí testovacího vzorku je používán v případě, kdy máme k dispozici dostatek pozorování. Množina dat $\mathcal{L}$ se rozdělí na trénovací data $\mathcal{L}_1$ o počtu N_1, a testovací $\mathcal{L}_2$ o počtu N_2. Posloupnost stromů $\{T_k\}$ se vytvoří na základě dat z $\mathcal{L}_1$, a následně se otestuje na datech z $\mathcal{L}_2$.

Odhad nepřesnosti stromu je pak dán vzorcem

$$L^{ts}(T_k) = \frac{1}{N_2} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{L}_2} (y_i - T_k(\boldsymbol{x}_i))^2$$

Pokud nemáme dostatek pozorování, preferujeme křížovou validaci. Rozdělíme pozorování $\mathcal{L}$ na V části, čímž dostaneme množiny $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_V$, v nichž každá množina $\mathcal{L}_v$ obsahuje stejný, nebo aspoň přibližně stejný, počet pozorování.

Označíme množinu $\mathcal{L}^{(v)} = \mathcal{L} - \mathcal{L}_v$ jako trénovací vzorek. Následně vytvoříme a prořízneme strom s využitím dat z $\mathcal{L}^{(v)}$. Pro každé v a pro každé α tak dostaneme strom $T^{(v)}(\alpha)$, což je strom s nejmenší celkovou nepřesností pro parametr α, jež byl vytvořen na základě pozorování z $\mathcal{L}^{(v)}$.

Nyní s použitím všech pozorování z $\mathcal{L}$, vybudujeme strom T, a následně ho prořežeme, čím dostaneme posloupnost (3.7), ze které budeme vybírat optimální strom, a u ní i odpovídající posloupnost parametrů $\{\alpha_k\}$. Definujeme α'_k jako geometrický průměr dvou sousedních hodnot, tj. $\alpha'_k = \sqrt{\alpha_k \alpha_{k+1}}$. Označíme $T_k^{(v)}(\boldsymbol{x})$ jako prediktor odpovídající stromu $T^{(v)}(\alpha'_k)$. Odhad nepřesnosti stromů pomocí křížové validace $L^{cv}(T_k)$ se pak rovná

$$L^{cv}(T_k) = \frac{1}{N} \sum_{v=1}^{V} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{L}_v} (y_i - T_k^{(v)}(\boldsymbol{x}_i))^2.$$

Jako strom o optimální velikosti označíme ten, který má nejnižší $L^{ts}(T_k)$, respektive $L^{cv}(T_k)$.

3.1.3 Poznámky k metodě CART

Klasifikační stromy

Klasifikační strom se liší od regresního tím, že vysvětlovaná proměnná nabývá hodnoty z konečné množiny $C = \{c_1, \ldots, c_p\}$. V takovém případě musíme použit jiný přístup pro přiřazení výsledné hodnoty k listu a pro výpočet nepřesnosti stromu.

Pro zjednodušení označme $C = \{1, \ldots, p\}$. Výsledná hodnota listu t je taková třída $c \in C$, do níž v daném uzlu spadá nejvíce pozorování. Formálně je to tedy

$$c(t) = \operatorname*{argmax}_c \hat{p}_{tc},$$

kde $c(t)$ je výsledná třída přiřazená uzlu t a $\hat{p}_{tc}$ je definováno předpisem

$$\hat{p}_{tc} = \frac{1}{N_t} \sum_{(y_i:\ \boldsymbol{x}_i \in t)} I(y_i = c) \tag{3.10}$$

Výraz (3.10) lze interpretovat jako odhad pravděpodobnosti, s jakou pozorování y_i spadá do třídy c za podmínky, že se nachází v uzlu t.

Co se tyče míry nepřesnosti stromů, respektive nepřesnosti daného uzlu t, tak v případě klasifikačního stromu nemůžeme použít střední čtvercovou chybu. Místo toho se pro stanovení nepřesnosti uzlu t nejčastěji používají tyto tři míry nepřesnosti:

1. Chyba špatné klasifikace (*Misclassification error*):

$$L(t) = \frac{1}{N_t} \sum_{(y_i:\ \boldsymbol{x}_i \in t)} I(y_i \neq c(t))$$

2. Giniho index:

$$L(t) = \sum_{c \neq c'} \hat{p}_{tc} \hat{p}_{tc'} = \sum_{c=1}^{p} \hat{p}_{tc}(1 - \hat{p}_{tc})$$

3. Entropie:

$$L(t) = -\sum_{c=1}^{p} \hat{p}_{tc} \log \hat{p}_{tc}$$

Více o mírách měření chyby stromu lze najít v již zmíněné publikaci [6].

Neúplná pozorování

Častým problémem analýzy dat jsou neúplná pozorování. To, jak si strom poradí s chybějícími hodnotami, záleží na tom, jaká hodnota chybí. Pokud chybí závislá proměnná, pak bude celé pozorování vyloučeno z množiny, která slouží pro stavbu stromu. Stejně tak pozorování bude vyloučeno, pokud chybí hodnoty všech nezávislých prediktorů.

Avšak pokud chybí hodnoty jenom některých z prediktorů, pak lze použít metodu *náhradního dělení* (*surrogate split method*). Předpokládejme $X_j^* < r^*$ pro nějaké $j = 1, \ldots, k$ je nejlepší dělení uzlu t. Pokud ale hodnota prediktoru X_j^* pro i-té pozorování chybí, pak rozhodnutí o tom, zda pozorování půjde do levého či pravého následníku uzlu t provedeme na základě náhradního dělení. Náhradní dělení uzlu t je druhé nejlepší dělení uzlu t. Předpokládáme, že hodnota prediktoru pro toto náhradní dělení je dostupná. Pokud ne, lze použít druhé náhradní dělení atd.

Významnost proměnných

Na začátku je důležité uvést, v jakém smyslu je zde používán termín významnost. Termín významnost, nebo spíše statistická významnost, je používán v souvislosti s testováním hypotéz. Statisticky významný je jev, jehož absence byla jakožto nulová hypotéza zamítnuta v rámci statistického testu.

Zde nebudeme chápat slovo významnost proměnných ve smyslu statistické významnosti, ale ve smyslu, zda proměnná je danou metodou považována za důležitou. Významnost proměnných v případě stromů se většinou měří dle velikosti

poklesu nepřesnosti stromu. Čím je proměnná důležitější, tím větší pokles nepřesnosti by měla způsobit.

Struktura stromů může vést k zavádějící myšlence, že významné proměnné jsou jenom takové, které slouží k dělení uzlů. Pozorování se v uzlu dělí podle nejlepšího dělení určité proměnné. Druhé a další nejlepší dělení se neobjeví. Avšak může dojít k situaci, kdy určitá proměnná by způsobila významný pokles nepřesností, avšak v daném uzlu není dělení podle této proměnné nejlepší. Proto lepší způsob měření významnosti proměnných je založen na využití náhradních dělení. Označme $\tilde{s}_{jt}$ jako nejlepší dělení pro j-tou proměnnou v uzlu t. Pak míra významnosti j-té proměnné je daná výrazem

$$M(j) = \sum_{t \in T} \Delta L(\tilde{s}_{jt}, t) \tag{3.11}$$

Odlehlá a vlivná pozorování

Kromě chybějících proměnných při analýze dat dělají problémy také vlivná a odlehlá pozorování. Metoda CART se umí celkem dobře vypořádat s vlivnými pozorováními, ale hůř s odlehlými. Odlehlá pozorování se metoda CART snaží umisťovat do samostatných uzlů. Takový postup přináší nízkou nepřesnost na trénovacích datech, ale ne na testovacích.

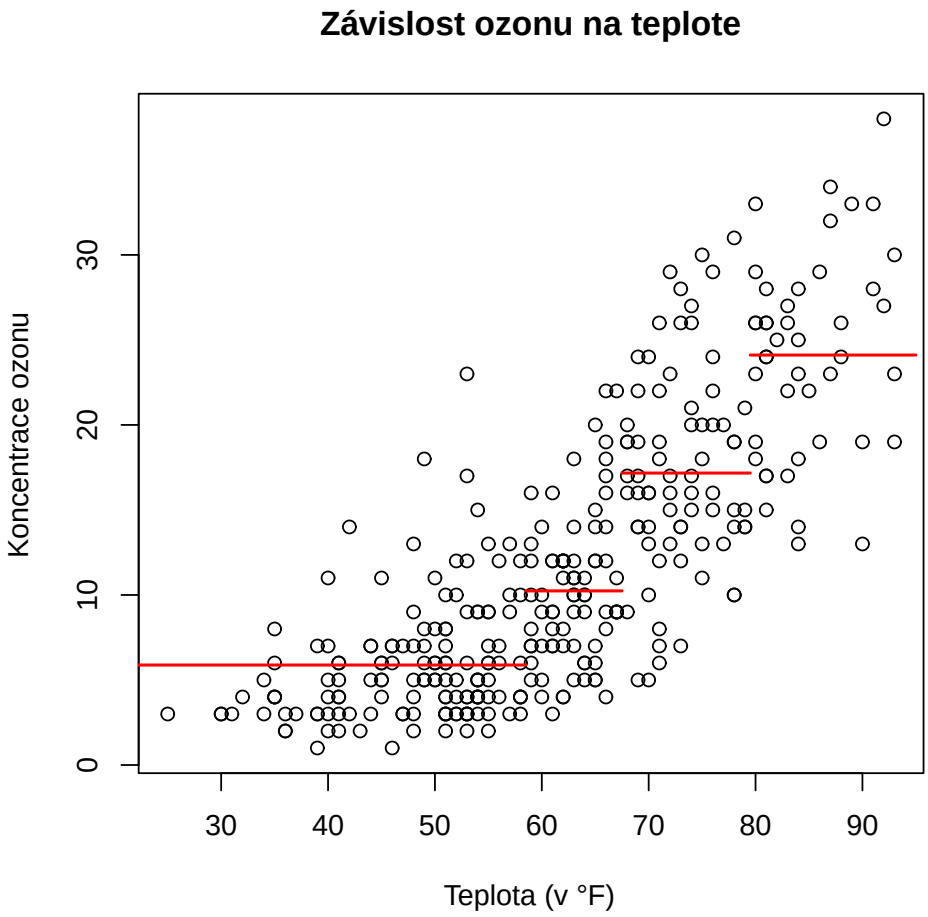
Počet následníku

Metoda CART používá binární štěpení jednotlivých uzlů. Lze upravit dělicí kritéria tak, aby místo dvou následníků vznikaly tři nebo i více. Někdy to může mít své opodstatnění, ale obecně to ale není dobrá strategie, protože by se data dělily moc rychle, a tím pádem by na spodní patra stromu zbývalo méně pozorování, což by snížilo kvalitu odhadu.

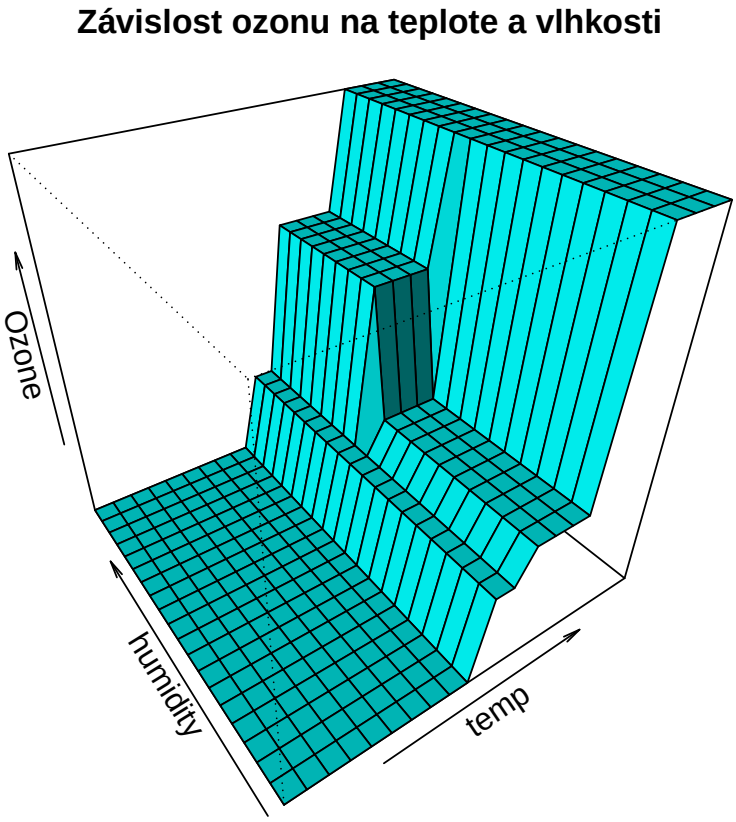
Nevýhody stromu CART

Velkým omezením stromových metod je jejich nestabilita. Strom se vytváří na základě konkrétních dat, a při použití jiných dat může vzniknout jiný strom. Nicméně nestabilita stromů může být i svým způsobem užitečná, indikuje totiž určitou nekvalitu dat - kolinearitu proměnných, nevhodně vybrané vysvětlující proměnné nebo velký podíl bílého šumu. Ve snaze omezit nestabilitu, byly vyvinuty další metody, které se snaží zachovat výhody regresních stromů, ale zároveň jsou stabilnější. Označují se jako regresní lesy, a jsou podrobněji rozebrané v další kapitole.

Výsledný model CART není spojitý, což nemusí být velký problém při klasifikačních úlohách, ale v případě regresních se jedná o vážný nedostatek, který snižuje výkonnost stromů. Metoda MARS, která řeší tento nedostatek a která se dá považovat za modifikovaný CART, je rozebrána v následující podkapitole.



Obrázek 3.2: Na tomto grafu můžeme vidět naměřené hodnoty koncentrace ozónu a to, jak dobře metoda CART aproximuje tyto data. První věcí, kterou si všimneme, je nespojitost výsledné funkce. Odpovídá to (3.1), který přiřadí pozorování $\boldsymbol{x}$ koeficient b_m pokud $\boldsymbol{x} \in R_m$, a 0 jinak. Zde například výsledný model přiřadí pozorování x hodnotu 5,88 pokud $x \in (-\infty,\ 58,5]$, a 0 jinak.



Obrázek 3.3: Třírozměrný model CART, který reprezentuje závislost koncentrace ozónu na teplotě a vlhkosti vzduchu.

3.2 MARS

Metoda MARS (*Multivariate Adaptive Regression Splines*) je dalším zástupcem stromových metod. Pracuje na obdobném principu jako metoda CART, ale zároveň se od ní liší v několika významných aspektech. A tak při jejím popisu vyjdeme ze znalostí metody CART, ze které změnou příslušných procesů vytvoříme metodu MARS.

3.2.1 Algoritmus MARS

Jak už víme, metoda CART předpokládá, že vztah mezi závislou a nezávislými proměnnými lze modelovat pomocí následující funkce

$$T(\boldsymbol{x}) = \sum_{m=1}^{M} \beta_m h_m(\boldsymbol{x}), \tag{3.12}$$

kde $h_m(\boldsymbol{x}) = I(\boldsymbol{x} \in R_m)$. Funkce h_m bývá označována jako *základní funkce* (*basis function*). V případě metody CART je základní funkce indikátor, zda pozorová-

ní $\boldsymbol{x}$ leží v podmnožině R_m.

Pomocí algoritmu, který byl popsán v předchozí kapitole, metoda CART odhadne koeficienty β_m a rozdělí definiční obor na vhodné podmnožiny R_m. Tomuto algoritmu se také říká *regresní rekurzivní dělení* (*recursive partitioning regression*).

Tento algoritmus nyní popíšeme. Nechť je $H(\eta)$ funkce definována pomocí následujícího předpisu:

$$H(\eta) = \begin{cases} 1 & \eta \geq 0 \\ 0 & \text{jinak} \end{cases} \tag{3.13}$$

Funkci $H(\eta)$ se říká *kroková funkce* (*step function*). Nepřesnost stromu T, která se znovu nejčastěji počítá pomocí střední čtvercové chyby, značíme $L(T)$.

Krok	**Algoritmus 1**
1	$h_1(\boldsymbol{x}) \leftarrow 1$
2	For $M = 2$ to $M = M_{max}$ do : $r^* \leftarrow \infty$
3	For $m = 1$ to $M - 1$ do :
4	For $j = 1$ to J do :
5	For $s \in \{x_{ij}, i = 1, \ldots, N\}$
6	$T(\boldsymbol{x}) \leftarrow \sum_{i=1, i\neq m}^{M-1} \beta_i h_i(\boldsymbol{x}) + \beta_m h_m(\boldsymbol{x}) H[+(X_j - s)] + \beta_M h_m(\boldsymbol{x}) H[-(X_j - s)]$
7	$r \leftarrow \min_{\beta_1,\ldots,\beta_M} L(T)$
8	if $r < r^*$ then $r^* \leftarrow r$; $m^* \leftarrow m$; $j^* \leftarrow j$; $s^* \leftarrow s$ end if
9	end for
10	end for
11	end for
12	$h_M(\boldsymbol{x}) \leftarrow h_{m^*}(\boldsymbol{x}) H[-(X_{j^*} - s^*)]$
13	$h_{m^*}(\boldsymbol{x}) \leftarrow h_{m^*}(\boldsymbol{x}) H[+(X_{j^*} - s^*)]$
14	end for
15	end algorithm

Na řádku (1) je definována první základní funkce. První základní funkce odpovídá kořenu stromu. Druhý řádek je stanovení počtu základních funkcí v modelu, což je analogie zastavovacího kritéria v případě stromu CART, kdy se růst stromu zastaví poté, co dosáhne zvoleného maximálního počtu listů M_{max}. Smyčka, která začíná na třetím řádku, vybírá z aktuálně konečných základních funkcí jednu, kterou se pokusí nahradit. Pro tento uzel se hledá nejlepší dělení. Smyčka, začínající na řádku (4), vybere jeden z J prediktorů, smyčka začínající na (5) pro vybraný prediktor postupně zkouší všechny možné body dělení s ve snaze najít nejlepší bod pro dělení.

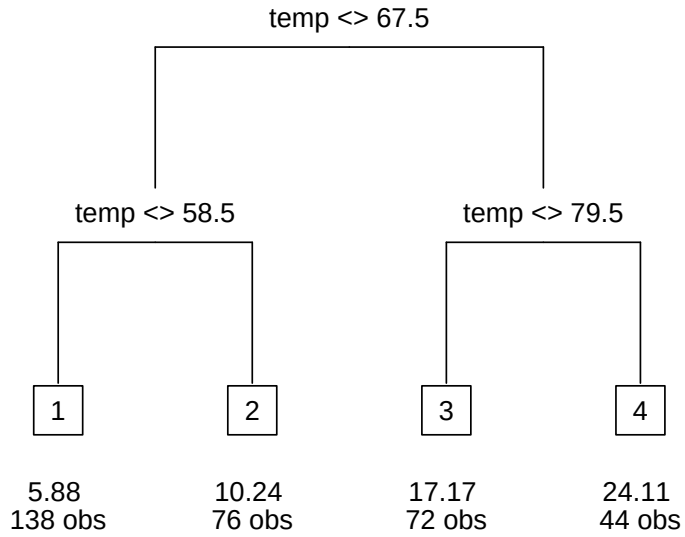
Vnitřní tělo algoritmu, řádky (6) až (8), se vztahují k hledání nejlepšího dělení. Zde m-tá základní funkce h_m a koeficient β_m nahradí se dvojici $\beta_m h_m(\boldsymbol{x}) H(+(X_j - s))$ a $\beta_M h_m(\boldsymbol{x}) H(-(X_j - s))$. Následně se odhadnou $b_1, \ldots, b_M$, a vypočítá se nepřesnost stromu $L(T)$. Pokud dojde k poklesu dosud nejmenší hodnoty nepřesnosti stromu, pak se uzel, prediktor a bod dělení označí jako nejlepší, a algoritmus

pokračuje dál v hledání. Pomocí tří vnitřních smyček se nakonec vybere základní funkce, která má být nahrazená, a také nejlepší dvojice základních funkcí, která ji nahradí. Samotné nahrazení probíhá na řádcích (12) a (13).

Základní funkce vyprodukované tímto algoritmem jsou ve tvaru

$$h_m = \prod_{k=1}^{K_m} H(q_k \cdot (X_{j_k} - s_k)). \tag{3.14}$$

Veličina K_m je počet dělení, kterých bylo zapotřebí ke vzniku h_m. Veličina q_k nabývá hodnot ± 1 a určuje, zda pozorování půjde do levého či pravého uzlu, x_{j_k} je označení, pro kterou proměnnou a na jaké úrovně bylo provedeno dělení a s_k je bod, určující kde a na jaké úrovni bylo provedeno dělení.



Příklad 3.12 *Například základní funkce stromu z obrázku 3.1 (strom CART modelující závislost koncentrace ozonu na teplotě) vypadají následovně:*

$h_1 = H(-(x - 67,5))H(-(x - 58,5))$
$h_2 = H(-(x - 67,5))H(+(x - 58,5))$
$h_3 = H(+(x - 67,5))H(-(x - 79,5))$
$h_4 = H(+(x - 67,5))H(+(x - 79,5))$

Metoda MARS se liší od metody CART především ve třech věcech, a to ve

- tvaru krokové funkce
- způsobu vytváření následníků
- výběru vysvětlujících proměnných, které mohou být použity pro dělení

Kroková funkce $H(\eta)$ nabývá v modelu CART pouze dvou hodnot, 0 a 1. Výsledkem je po částech konstantní nespojitý model, viz (3.2). Ačkoliv tento přístup má výhody v tom, že model je snadno interpretovatelný a není náročný na výpočetní čas, vzniklá nespojitost nepříznivě ovlivňuje přesnost modelu.

Metoda MARS se snaží tuto nespojitost odstranit, a proto místo krokové funkce definované předpisem (3.13) používá dvojici *lineárních splajnů* definovaných jako $(x-s)_+ = \max(x-s, 0)$ a $(x-s)_- = -\min(x-s, 0)$

Každá taková funkce je po částech lineární, s *centrem* (*knot*) v bodě s.

Druhou změnou oproti metodě CART je tvar výsledné základní funkce h_m. Základní funkce jsou součinem jednotlivých krokových funkcí H, které mají za argumenty výrazy $\pm(x-s)$, přičemž metoda CART neklade na tyto výrazy žádné omezující podmínky. Metoda MARS ale postupuje jinak. Základní funkce h_m jsou také součinem krokových funkcí, ale na rozdíl od metody CART je zde omezující podmínka na vstupující vysvětlující proměnné, a to, že součin nesmí obsahovat stejnou vysvětlující proměnnou více než jednou.

Posledním rozdílem oproti metodě CART je průběh dělicího procesu. V případě metody CART při hledání optimální množiny základních funkcí dochází k nahrazení původní funkce h_m, a k ní příslušné podmnožiny R_m, dvěma novými funkcemi h_l a h_r, s odpovídajícími podmnožinami R_l a R_r. Původní funkce h_m už nemůže byt přímo použitá pro vytvoření nových funkcí, jinými slovy podmnožina R_m není dále přístupná pro další dělení. Proces hledání nejlepších následníků pomocí metody MARS vypadá jinak.

Pro každou vysvětlující proměnnou X_j vytvoříme lineární splajny s centry v bodech x_{ij}. Vznikne nám množina *kandidátských funkcí* $\mathcal{C}$.

$$\mathcal{C} = \{(X_j - s)_+,\ (X_j - s)_-\},\ s \in \{x_{1j}, x_{2j}, \ldots, x_{Nj}\},\ j = 1, 2, \ldots, J \quad (3.15)$$

Funkce z této množiny využíváme pro tvorbu množiny základních funkcí $\mathcal{M}$. Do té nejdříve dáme základní konstantní funkci $h_1 = 1$. V každém dalším kroku do $\mathcal{M}$ přidáváme další funkce h_m. Nechť je v množině $\mathcal{M}$ je již M základních funkcí a M koeficientů b_m. Do množiny $\mathcal{M}$ přidáme další funkci ve tvaru

$$b_{M+1} h_m(\boldsymbol{x}) \cdot (x_j - s)_+ + b_{M+2} h_m(\boldsymbol{x}) \cdot (x_j - s)_-,\ m = 1, \ldots, M, \quad (3.16)$$

a to takovou, která způsobí největší pokles nepřesnosti stromu. Koeficienty b_{M+1} a b_{M+2} jsou odhadnuté pomocí metody nejmenších čtverců společně s ostatními koeficienty v modelu. Základní funkce se přidávají do $\mathcal{M}$, dokud počet funkcí M nedosáhne předem stanoveného počtu M_{max}.

Jak vidíme, proces hledání nejlepšího dělení metody MARS se liší od metody CART. Podstatnou změnou je to, že funkce, která se již nachází v množině základních funkcí, může být opakovaně použitá pro vytváření následníků. Pokud bychom to měli převést do stromové terminologie, znamená to, že jeden uzel lze dělit vícekrát a vytvářet různé dvojice následníků.

Tento přístup způsobuje dvě věci. Především to, že podmnožiny R_m, které se vztahují k výsledným základním funkcím h_m, nejsou disjunktní. To znamená, že při prořezávání stromu lze odstranit libovolné funkce, kromě základní h_1, a s nimi spojené podmnožiny, aniž by vznikala "díra" v definičním prostoru. Dalším, méně příjemným, důsledkem je ztráta klasické stromové struktury, což způsobuje horší interpretaci výsledného modelu.

Na základě výše uvedených poznámek upravíme Algoritmus 1, který označíme jako Algoritmus 2. Jedná se o algoritmus metody MARS vytvářející množinu základních funkcí.

Krok	**Algoritmus 2**
1	$h_1(\boldsymbol{x}) \leftarrow 1; M \leftarrow 2$
2	Loop until $M > M_{max} : r^* \leftarrow \infty$
3	For $m = 1$ to M do :
4	For $j \notin J_m$
5	For $s \in \{x_{ij},\ i = 1, \ldots, N\}$
6	$T(\boldsymbol{x}) \leftarrow \sum_{i=1}^{M} \beta_i h_i(\boldsymbol{x}) + \beta_{M+1} h_m(\boldsymbol{x})(X_j - s)_+ + \beta_{M+2} h_m(\boldsymbol{x})(X_j - s)_-$
7	$r \leftarrow \min_{\beta_1,\ldots,\beta_{M+2}} L(T)$
8	if $r < r^*$ then $r^* \leftarrow r;\ m^* \leftarrow m;\ j^* \leftarrow j;\ s^* \leftarrow s$ end if
9	end for
10	end for
11	end for
12	$h_{M+1}(\boldsymbol{x}) \leftarrow h_{m^*}(\boldsymbol{x})(X_{j^*} - s^*)_+$
13	$h_{M+2}(\boldsymbol{x}) \leftarrow h_{m^*}(\boldsymbol{x})(X_{j^*} - s^*)_-$
14	$M \leftarrow M + 2$
15	end loop
16	end algorithm

V takto zapsaném algoritmu můžeme vidět zmíněné změny oproti Algoritmu 1. Na řádcích (6), (12) a (13) místo původní krokové funkci $H(\eta)$ je použit lineární splajn. Na řádku (4) je kontrolní smyčka, která ověřuje, zda nově zkoumaná vysvětlující proměnná X_j nepatří do množiny J_m, což je množina již použitých vysvětlujících proměnných pro funkci h_m. Řádek číslo (6) má stejnou funkci jako řádek (6) v Algoritmu 1, avšak s jedním významným rozdílem. V Algoritmu 1 se m-tá základní funkce nahradila součtem dvou nových, v Algoritmu 2 m-tá funkce zůstává, a navíc se do modelu zkouší přidat nové funkce. Na řádcích (12)-(13) v obou dvou algoritmech dochází k přidávání nových nejlepších funkcí, s tím rozdílem, že v Algoritmu 1 jedna z nových funkcí nahrazuje původní, v Algoritmu 2 se přímo vytvoří dvě nové základní funkce.

3.2.2 Prořezávání stromu MARS

Stejně jako v případě stromu CART, je model vytvořený za použití Algoritmu 2 takzvaně přeučený. To znamená, že výsledná množina $\mathcal{M}$ je příliš velká, a i když na trénovacích datech má model nízkou nepřesnost, na testovacích se tato nepřesnost zvětší. Chceme tedy najít strom o optimální velikosti, čehož docílíme pomocí prořezávání.

Prořezávání pracuje na stejném principu, jako v případě metody CART. V každém kroku prořezávací algoritmus odstraní tu základní funkci, jejíž odstranění způsobí nejmenší nárůst nepřesností stromu. Výhodou oproti metodě CART je to, že z množiny základních funkcí může být odstraněna libovolná funkce h_m, kromě h_1, aniž by se tím narušila struktura stromu. Je to umožněno tím, že při hledání nového nejlepšího dělení se rodičovské funkce nenahrazují následníky,

a tedy definiční podmnožiny R_m, které patří k jednotlivým základním funkcím, nejsou disjunktní.

Výsledkem prořezávání je posloupnost stromů $T_{M_{max}}, T_1, \ldots, t_1$. Jednotlivé stromy T_m, kde m je počet základních funkcí, jsou nejlepší možné stromy, které dokáže vyprodukovat daný algoritmus. Mezi těmito stromy musíme vybrat ten o optimální velikosti.

Ačkoliv pro výběr optimálního stromu se i zde může použít křížová validace, z početních důvodů se používá zobecněná křížová validace, kterou značíme GCV (*Generalized cross-validation*). Toto kritérium je definováno jako

$$GCV(m) = \frac{\sum_{i=1}^{N}(y_i - T_m(\boldsymbol{x}_i))^2}{\left(1 - \frac{M(m)}{N}\right)^2}, \tag{3.17}$$

kde $M(m)$ je veličina, která se označuje jako skutečný počet parametrů modelu.

Skutečný počet parametrů definujeme jako $M(m) = r + cK$. Veličina r je počet takových základních funkcí v modelu, které nelze vytvořit pomocí lineární kombinace ostatních základních funkcí, K je počet centrů, které byly vybrané Algoritmem 2, a c je penalizační konstanta, která většinou nabývá hodnoty 2 nebo 3.

Kritérium $GCV(m)$ se spočítá pro každý model, a následně jako nejlepší model se vybere ten, jehož $GCV(m)$ je nejnižší.

3.2.3 Poznámky k metodě MARS

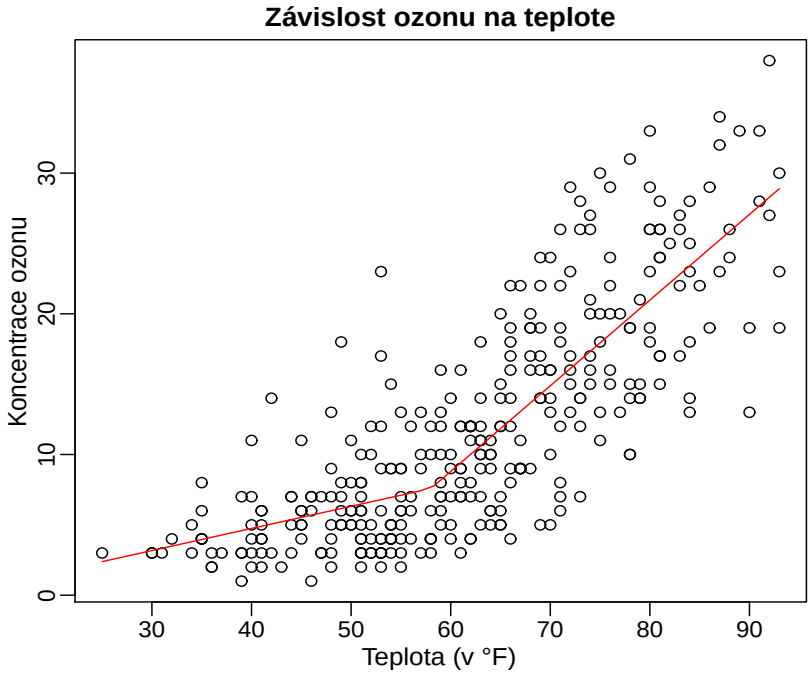
Ačkoliv se metoda MARS počítá do stromových metod, nemá na rozdíl od metody CART klasickou přehlednou stromovou strukturu, proto interpretace výsledného modelu je obtížnější.

Stejně jako u metody CART, tak i u metody MARS je problém s kolinearitou. Proto, pokud není možnost provést úpravu dat před samotným použitím metody MARS, je navrhnuta následující úprava nepřesnosti L (Algoritmus 2, řádek (7)):

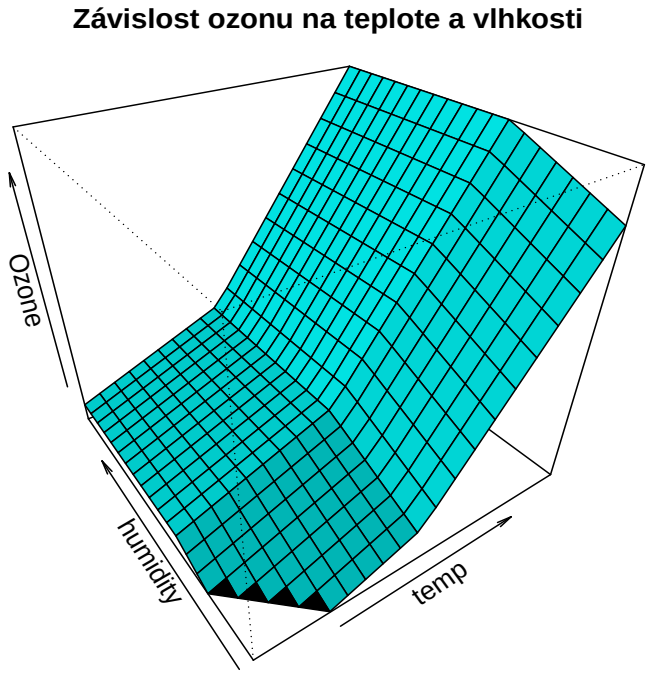
$$L(T) \leftarrow L(T)(1 + \gamma I(X_j \in P)) \tag{3.18}$$

V okamžiku, kdy do množiny $\mathcal{M}$ přidáváme další základní funkci ve tvaru (3.16), kontrolujeme, zda proměnná, která se vyskytuje v kandidátské funkci, je už obsažená v množině P, což je množina všech již použitých vysvětlujících proměnných. Pokud se tato proměnná nepoužije poprvé, pomocí parametru γ nepřesnost zvýšíme. Parametr γ reguluje míru penalizace za příliš mnoho použitých vysvětlujících proměnných. Velikost tohoto parametru záleží na míře kolinearity mezi proměnnými a na tom, zda preferujeme vyšší aproximaci nebo lepší interpretovatelnost modelu.

Metoda MARS ošetřuje vlivná a odlehlá pozorování obdobným způsobem jako metoda CART. Odlehlá pozorování se snaží izolovat do samostatných uzlů, takže ty sice mají velký vliv na koeficienty v rámci uzlů, ale celková přesnost modelu tím není ovlivněná.



Obrázek 3.4: Aproximace dat stromem MARS při použití jedné vysvětlující proměnné (teploty). V tomto případě je výsledný model (3.12) součtem následujících tří základních funkcí: $T(x) = b_1 h_1 + b_2 h_2 + b_3 h_3$, kde koeficienty $\boldsymbol{b}$ se rovnají $b_1 = 7,58, b_2 = 0,61$ a $b_3 = -0,16$, a základní funkce $\boldsymbol{h}$ jsou ve tvaru $h_1 = 1, h_2 = (x - 58)_+$ a $h_3 = (x - 58)_-$.



Obrázek 3.5: Výsledný model při použití dvou vysvětlujících proměnných (teplota a vlhkost vzduchu).

4. Regresní lesy

Stromy mají jednu velkou nevýhodu, a tou je nestabilita. Při změně dat, na základě kterých byl vytvořen strom, může dojít k poměrně výrazné proměně výsledného modelu, což může nepříznivě ovlivnit jeho predikční schopnosti i interpretaci. Řešení tohoto problému nabízí *regresní* a *klasifikační lesy*.

Regresní lesy patří do tzv. komisních nebo souborových metod (*committee, ensemble methods*), jejichž hlavní myšlenkou je kombinace více samostatných modelů do jednoho celku.

Na základě těchto informací není těžké si domyslet, co se skrývá pod pojmem regresní les. Regresní les je model, který přiřazuje pozorování $\boldsymbol{x}$ hodnotu na základě kombinace výsledků několika regresních stromů. Analogicky, klasifikační les je kombinací několika klasifikačních stromů.

Jako jednotlivé modely pro tyto komisní metody mohou sloužit nejenom stromy, ale například i neuronové sítě či lineární regrese. Použití komisních metod se ale nehodí pro každou techniku.

V této kapitole popíšeme tři techniky, *bagging*, *boosting* a *náhodný les* (*random forest*). První dvě jsou obecně navržené metody, které lze použít i na jiné techniky než stromy, náhodný les je pak přímo navržen pro stromy.

4.1 Bagging

Autorem metody bagging (***b**ootstrap **agg**regat**ing***) je Leo Breiman, jeden z tvůrců metody CART. Jako zdroj informací nám poslouží [2] a [3].

Nechť máme soubor pozorování $\mathcal{L} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, N\}$, kde y_i nabývá hodnot z $\mathbb{R}$. Dále uvažujme posloupnost souborů pozorování $\{\mathcal{L}_k\}_1^K$, takových že $\mathcal{L}_k = \{(\boldsymbol{x}_{ki}, y_{ki}),\ i = 1, \ldots, N\}$.

Označme jako $T(\boldsymbol{x}, \mathcal{L})$ model, který byl sestrojen za použití dat ze souboru $\mathcal{L}$. Obdobně označíme $T(\boldsymbol{x}, \mathcal{L}_k)$ jako model, sestrojených na základě souboru $\mathcal{L}_k$.

Hlavní myšlenkou baggingu je, že jako odhad vysvětlované proměnné y nebudeme brát $T(\boldsymbol{x}, \mathcal{L})$, ale průměr přes všechny výsledky, které obdržíme z jednotlivých modelů $T(\boldsymbol{x}, \mathcal{L}_k)$, tj.

$$F_A(\boldsymbol{x}) = \frac{1}{K} \sum_{k=1}^{K} T(\boldsymbol{x}, \mathcal{L}_k). \tag{4.1}$$

V případě, že y nabývá hodnot z množiny $C = \{c_1, c_2, \ldots, c_p\}$, tak samozřejmě hledat průměr pro $T(\boldsymbol{x}, \mathcal{L}_k)$ nemá význam. V takovém případě se pro odhad y používá *většinové hlasování*. Označme $N_{c_j} = |k; T(\boldsymbol{x}, \mathcal{L}_k) = c_j|$, pak

$$F_A(\boldsymbol{x}) = \underset{c_j}{\operatorname{argmax}}\ N_{c_j}, \tag{4.2}$$

čili výsledná třída je ta, které se objevila při klasifikaci jednotlivými stromy nejčastěji. Místo většinového hlasování můžeme také použít *vážené hlasování*, kdy

každému stromu je přiřazená váha, která zohledňuje nepřesnost stromu. Čím je hodnocení stromu přesnější, tím má jeho hlas větší váhu.

Avšak zřídka máme k dispozici tolik pozorování, abychom mohli mít K různých souborů pro vytvoření vhodných modelů. Proto se soubor $\mathcal{L}_k$ vytvoří jako *bootstrapový výběr* z $\mathcal{L}$, tj. soubor $\mathcal{L}_k$ vznikne jako výběr s vracením z $\mathcal{L}$. Velikost bootstrapového výběru je nejčastěji stejná, jako velikost základního souboru pozorování. Označme takovou posloupnost výběrů jako $\mathcal{L}_k^{(B)},\ k = 1, \ldots, K$.

Pro odhad vysvětlované proměnné y místo $F_A(\boldsymbol{x})$ použijeme $F_B(\boldsymbol{x})$, který je definován jako

$$F_B(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K} T(\boldsymbol{x}, \mathcal{L}_k^{(B)}). \tag{4.3}$$

Jednotlivé modely $T(\boldsymbol{x}, \mathcal{L}_k^{(B)})$ vybudujeme stejným způsobem, který jsme popsali v kapitole (3.1). Nejdříve na základě pozorování ze souboru $\mathcal{L}_k^{(B)}$ obdržíme maximální strom. Ten následně prořežeme a dostaneme posloupnost stromů

$$T_1^{(B)} \supset \ldots \supset t_1^{(B)},$$

ze které vybereme ten o optimální velikosti. Optimální strom vybereme tak, že spočítáme nepřesnost stromu na tzv. *out-of-bag* pozorováních. Pozorování out-of-bag jsou ta pozorování, která nebyla vybrána do souboru $\mathcal{L}_k$. Protože bootstrapový výběr je výběrem s opakováním, při výběru o velikost N se některá pozorování vyskytnou vícekrát, zatímco některá pozorování se nevyberou vůbec.

Místo použití out-of-bag pozorování lze pro výběr optimálního stromu použít i celý výběr $\mathcal{L}$. Pak ale dochází k podhodnocení odhadu nepřesnosti modelu.

Odpověď, proč průměrování přes několik modelů nedává horší výsledky než použití jednoho modelu, vychází z následující rovnice, která platí pro všechna $y_i,\ i = 1, \ldots, N$:

$$\frac{1}{K}\sum_{k=1}^{K}(y_i - T(\boldsymbol{x}_i, \mathcal{L}_k))^2 = y_i^2 - 2y_i\frac{1}{K}\sum_{k=1}^{K} T(\boldsymbol{x}_i, \mathcal{L}_k) + \frac{1}{K}\sum_{k=1}^{K}(T(\boldsymbol{x}_i, \mathcal{L}_k))^2. \tag{4.4}$$

Na poslední člen této rovnosti aplikujeme Jensenovou nerovnost: pokud je funkce g konvexní a je definovaná na intervalu I, $\lambda_1, \ldots, \lambda_n$ jsou nezáporná, pak pro libovolná $x_1, \ldots, x_n \in I$ platí

$$\lambda_1 g(x_1) + \ldots + \lambda_n g(x_n) \geq g(\lambda_1 x_1 + \ldots + \lambda_n x_n)$$

Druhá mocnina je konvexní funkce, proto lze aplikovat Jensenovou nerovnost na poslední člen (4.4):

$$\frac{1}{K}\sum_{k=1}^{K}(T(\boldsymbol{x}_i, \mathcal{L}_k))^2 \geq \left(\frac{1}{K}\sum_{k=1}^{K} T(\boldsymbol{x}_i, \mathcal{L}_k)\right)^2. \tag{4.5}$$

S použitím označení (4.1) a nerovnosti (4.5) na rovnici (4.4) dostaneme následující nerovnost:

$$\frac{1}{K}\sum_{k=1}^{K}(y_i - T(\boldsymbol{x}_i, \mathcal{L}_k))^2 \geq (y_i - F_A(\boldsymbol{x}_i, \mathcal{L}))^2 \tag{4.6}$$

Pokud předchozí rovnici sečteme přes $\forall i$ a vydělíme počtem pozorování N, dostaneme výraz

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N}(y_{ki}-T(x_{ki},\mathcal{L}_k))^2 \geq \frac{1}{N}\sum_{i=1}^{N}(y_i-F_A(\boldsymbol{x}_i,\mathcal{L}))^2, \tag{4.7}$$

což je hledaný výsledek a znamená, že průměrná čtvercová chyba F_A je menší nebo rovná průměrné čtvercové chybě jednotlivých stromů. To, o kolik chyba bude nižší, závisí na tom, jak výrazná bude nerovnost (4.5). U metod, kdy změna $\mathcal{L}$ příliš neovlivní výsledný model, se obě strany nerovnosti (4.6) téměř rovnají, a tak použití baggingu nevede k významnému zlepšení.

Stromy však patří mezi metody, kdy změna souboru dat může významně ovlivnit výsledný model, proto použití baggingu může významně vylepšit kvalitu předpovědi. Breiman v [2] uvádí, že vypočtená střední čtvercová chyba baggingu je o 21 % až 46 % nižší, než když se na stejná data použije obyčejná metoda CART. Tato čísla se pokusíme ověřit v páté kapitole.

Na závěr přidáme ještě dvě krátké poznámky ohledně velikosti lesu, neboli počtu stromů v lesu, a způsobu určení důležitosti proměnných.

Přesný počet stromů není explicitně dán. Breiman v [2] uvádí, že v případě regresní úlohy les o velikosti 25 stromů produkuje uspokojivé výsledky, v případě kvalifikačních úloh je potřeba počet stromů zdvojnásobit.

Důležitost proměnných určujeme obdobným způsobem, jako v případě stromu CART. Čili spočítáme pokles nepřesnosti pro každou proměnnou v každém jednotlivém stromě, a výsledek pak vydělíme počtem stromů. Proměnná s největším takto spočítaným poklesem nepřesnosti je označená za nejvýznamnější.

4.2 Boosting

Boosting je další metoda, která využívá skupinového rozhodování. Ačkoliv na první pohled pracuje stejně jako bagging, ve skutečnosti se od něho značně liší. Stejně jako bagging kombinuje výsledky z více sestavených stromů. Ale zatímco v případě baggingu se jedná o kombinaci K samostatných stromů, v případě boostingu jednotlivé stromy samostatné nejsou. Boosting staví stromy postupně, přičemž nový strom závisí na svém předchůdci.

Ačkoliv boosting vznikl původně pro řešení klasifikačních úloh, lze ho úspěšně aplikovat i na úlohy regresní. Jako zdroj informace nám poslouží [7] a [11].

4.2.1 Algoritmus boosting

Z předchozích kapitol víme, že strom vytvořený metodou CART může být vyjádřen následující rovnicí

$$T(\boldsymbol{x};\Theta)=\sum_{m=1}^{M}\beta_m I(\boldsymbol{x}\in R_m),$$

kde $\Theta=\{R_m,\beta_m\}_1^M$. Přesné odhady hodnot těchto parametrů dostaneme, když hledáme nejmenší hodnotu nepřesnosti stromu L, čili hledáme řešení následující

rovnice:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\ L(T(\boldsymbol{x}; \Theta))$$

Stejně jako bagging i boosting sčítá jednotlivé stromy:

$$F_K(x) = \sum_{k=1}^{K} T(\boldsymbol{x}; \hat{\Theta}_k), \tag{4.8}$$

kde $\hat{\Theta}_k$ je soubor parametrů příslušející ke k-tému stromu. Tím ale podoba s baggingem končí. Bagging jako odhad pozorování y použije průměr přes všechny stromy v lese, kdežto boosting hledá sadu parametrů $\hat{\Theta}_k$ jako

$$\hat{\Theta}_k = \underset{\Theta_k}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{k-1}(\boldsymbol{x}_i) + T(\boldsymbol{x}_i; \Theta_k)), \tag{4.9}$$

kde F_{k-1} je model ve tvaru (4.8) s počtem stromů $k-1$. Výraz $L(y_i, F_{k-1}(\boldsymbol{x}_i) + T(\boldsymbol{x}_i; \Theta_k))$ vyjadřuje rozdíl mezi pozorovanou hodnotou y_i a odhadem vypočteným pomocí výrazu $F_{k-1}(\boldsymbol{x}_i) + T(\boldsymbol{x}_i; \Theta_k)$. Tento rozdíl je vypočtený na základě zvolené nepřesnosti, což může být střední čtvercová nebo absolutní chyba.

Každá sada parametrů pro k-tý strom $\hat{\Theta}_k = \{R_{km}, \beta_{km}\}_1^{M_k}$ tak závisí na předchozím modelu F_{k-1}.

Za předpokladu, že již máme určené podmnožiny R_{km}, stanovit koeficienty b_{km} je celkem snadné

$$b_{km} = \underset{\beta_{km}}{\operatorname{argmin}} \sum_{(y_i:\ \boldsymbol{x}_i \in R_{km})} L(y_i, F_{k-1}(\boldsymbol{x}_i) + \beta_{km}). \tag{4.10}$$

Takže hlavní problém je, najít vhodné rozdělení definičního oboru $\mathcal{D}$ do jednotlivých regionů $\{R_{km}\}_1^{M_k}$. Obtížnost řešení závisí na zvolené nepřesnosti stromů a zda se jedná o klasifikační nebo regresní strom (najít řešení pro klasifikační úlohy je obecně o něco těžší než pro regresní).

Výraz (4.9) s libovolně zvolenou nepřesnosti L lze řešit pomocí numerické optimalizace. Celková nepřesnost při aproximaci skutečných hodnot y funkci $F_K(\boldsymbol{x})$ je daná rovnicí

$$L(F_K) = \sum_{i=1}^{N} L(y_i, F_K(\boldsymbol{x}_i)). \tag{4.11}$$

Pokud budeme na chvíli ignorovat skutečnost, že jsme omezeni konkrétním tvarem funkce F_K, tj. F_K nebude představovat součet K stromů, pak úlohu minimalizovat výraz (4.11) lze vidět jako numerickou optimalizaci

$$\hat{\mathbf{F}}_K = \underset{\mathbf{F}_K}{\operatorname{argmin}}\ L(\boldsymbol{y}, \mathbf{F}_K), \tag{4.12}$$

kdy se snažíme minimalizovat rozdíl mezi skutečnými hodnotami y a $\mathbf{F}_K$. Symbolem $\mathbf{F}_K \in \mathbb{R}^N$ značíme předpovědi aproximující funkce F_K:

$$\mathbf{F}_K = \{F_K(\boldsymbol{x}_1), \ldots, F_K(\boldsymbol{x}_N)\}.$$

Numerická optimalizace předpokládá existence řešení (4.12) ve tvaru sumy vektorů

$$\mathbf{F}_K = \sum_{k=0}^{K} \mathbf{h}_k, \quad \mathbf{h}_k \in \mathbb{R}^N, \tag{4.13}$$

kde $\mathbf{F}_0 = \mathbf{h}_0$ je počáteční odhad a $\mathbf{F}_k$, což je součet k vektorů, závisí na $\mathbf{F}_{k-1}$.

K nalezení vhodného $\mathbf{h}_k$ použijeme *iterační metodu největšího spadu s optimálním krokem*. Hledáme krok $\mathbf{h}_k$ ve tvaru $\mathbf{h}_k = -\rho_k \mathbf{g}_k$, kde ρ_k je velikost kroku a $\mathbf{g}_k \in \mathbb{R}^N$ je směr kroku. Jednotlivé složky vektoru $\mathbf{g}_k$ jsou ve tvaru

$$g_{ki} = \left[\frac{\partial L(y_i, F_{k-1}(\boldsymbol{x}_i))}{\partial F_{k-1}(\boldsymbol{x}_i)} \right] \tag{4.14}$$

Pokud máme spočítaný směr kroku, pak délku kroku stanovíme jako

$$\rho_k = \underset{\rho}{\operatorname{argmin}}\ L(\mathbf{F}_{k-1} - \rho \mathbf{g}_k). \tag{4.15}$$

Takže aktuální řešení rovnice (4.12) je upraveno na:

$$\mathbf{F}_k = \mathbf{F}_{k-1} - \rho_k \mathbf{g}_k, \tag{4.16}$$

a celý proces hledání se znovu opakuje.

V případě algoritmu boostingu se snažíme v každém kroku minimalizovat (4.9) s využitím již dříve spočteného modelu F_{k-1} Proto předpovědi stromu $T(\boldsymbol{x}_i; \Theta_k)$ jsou analogií k (4.14). Rozdíl je pak v tom, že funkce F_{k-1} ve výrazu (4.14) nejsou omezené, kdežto $(T(\boldsymbol{x}_1; \Theta_k), \ldots, T(\boldsymbol{x}_N; \Theta_k))$ je omezen podmínkou, že se jedná o předpovědi konkrétního typu funkce, zde stromů. Hledání vhodného ρ_k je analogií hledání řešení (4.10).

Gradient (4.14) se vypočítá snadno pro libovolně zvolenou nepřesnost L. Jelikož je gradient $\mathbf{g}_k$ v k-tém kroku nejlepším krokem směrem k optimálnímu řešení (4.12), snažíme se náš strom vytvořit tak, aby platilo

$$\tilde{\Theta}_k = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{N} (-g_{ik} - T(\boldsymbol{x}_i; \Theta))^2. \tag{4.17}$$

Výsledkem bude rozdělení definičního prostoru $\mathcal{D}$ do regionů $\tilde{R}_{km}$, které se sice budou lišit od těch, které bychom dostali přímo vyřešením (4.9), nicméně budou takřka stejně přesné a navíc mnohem snadněji počítatelné.

Algoritmus boostingu s využitím numerické optimalizace bude následující:

Krok	**Algoritmus 3**
1	$F_0 = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \beta)$
2	for $k = 1$ to K do:
3	$g_{ik} = \left[\frac{\partial L(y_i, F_{k-1}(\boldsymbol{x}_i))}{\partial F_{k-1}(\boldsymbol{x}_i)} \right], \ i = 1, \ldots, N$
4	vytvoříme strom $T(\boldsymbol{x}; \tilde{\Theta}_k), \ \tilde{\Theta}_k = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{N} (-g_{ik} - T(\boldsymbol{x}_i; \Theta))^2$
5	$b_{km} = \underset{\beta_{km}}{\operatorname{argmin}} \sum_{(y_i: \ \boldsymbol{x}_i \in R_{km})} L(y_i, F_{k-1}(\boldsymbol{x}_i) + \beta_{km}), \ m = 1, \ldots, M_k$
6	$F_k(\boldsymbol{x}) = F_{k-1}(\boldsymbol{x}) + \sum_{m=1}^{M_k} b_{km} I(\boldsymbol{x} \in R_{km})$
7	end for
8	end algorithm

Na prvním řádku je první odhad, který odpovídá výsledku stromu s jedním uzlem. Pak pro námi zvolený počet stromu K opakujeme výpočet gradientu (řádek (3)), výstavbu stromů s využitím gradientu (řádek (4)), a přiřazení konstanty k listům (řádek (5)). Na řádku (6) vzniká nový model, a celý cyklus se opakuje.

4.2.2 Volitelné parametry boostingu

Stejně jako v případě samostatného stromu se musí vyřešit otázka, jak velké by měly být jednotlivé stromy ve výsledném modelu (4.8). Jak již víme z kapitoly o metodě CART, možné řešení je vypěstovat co největší strom, který se pak prořeže do optimální velikosti. V případě boostingu to ale není ideální postup, protože takový postup zvyšuje početní náročnost.

Tento problém se vyřeší tak, že velikost stromů se stanoví předem, a je stejná pro každý strom. V každé iteraci tak vznikne strom s počtem listů M.

To, jaký počet listů bychom měli vybrat, nám může napovědět vlastnosti funkce, pomocí které aproximujeme vztah mezi závislou a nezávislými proměnnými, tj. funkce $\hat{f}$:

$$\hat{f} = \underset{f}{\operatorname{argmin}} \ \mathsf{E}_{XY} L(Y, f(\boldsymbol{X})). \tag{4.18}$$

Důležitou vlastnosti této funkce je stupeň interakce jednotlivých nezávislých proměnných mezi sebou. Stupeň interakce proměnných je zachycen v ANOVA rozkladu funkce $\hat{f}$:

$$\hat{f}(x) = \sum_j \hat{f}_j(x_j) + \sum_{jk} \hat{f}_{jk}(x_j, x_k) + \sum_{jkl} \hat{f}_{jkl}(x_j, x_k, x_l) + \ldots. \tag{4.19}$$

První součet v (4.19) je součet přes všechny funkce jedné proměnné. Takovým

funkcím se říká hlavní efekty. Jednotlivé funkce z množiny $\left\{\hat{f}_j(x_j)\right\}_1^J$ se hledají tak, aby jejich součet nejlépe aproximoval $\hat{f}$.

Druhá suma představuje součet funkcí, které mají za argument dvou nezávislé proměnné a jejích součet spolu se součtem přes hlavní efekty nejlépe aproximuje funkci $\hat{f}$.

V případě boostingu úroveň interakce přímo souvisí s velikostí stromů. Strom s M listy znamená, že funkci $\hat{f}$ odhadujeme s úrovní interakce maximálně $M-1$. Počet listů by tak měl odrážet úroveň interakce, která má při rozkladu (4.19) ještě smysl. Tahle úroveň nebývá známá, ve většině případů je ale malá. Často nám pro dostatečně přesnou aproximaci funkce $\hat{f}$ stačí pouze hlavní efekty. To znamená, že náš model se skládá ze stromů, které mají pouze dva listy. Takovému stromu říkáme *pařez* (*stump*). V praktické části diplomové práce vyzkoušíme, jak souvisí přesnost modelu a velikost jednotlivých stromů.

Kromě velikosti jednotlivých stromů potřebujeme také určit celkový počet stromů v regresním lesu. Každé provedení smyčky začínající na řádku (2) v algoritmu 3 snižuje chybu modelu. Nicméně pořád se jedná o snižování chyby na trénovacích datech. Proto snaha najít ideální K pro vývojový vzorek není ideální, protože takový model pak nemusí odpovídajícím způsobem pracovat na celé populaci.

Z tohoto důvodu se používá strategie, která se jmenuje *zmenšování* (*shrinkage*). Jedná se o techniku, kdy pomocí parametru ν regulujeme příspěvek jednotlivých stromů ke snížení nepřesnosti modelu. Nechť $0 < \nu < 1$, pak nahradíme řádek (6) algoritmu 3 výrazem

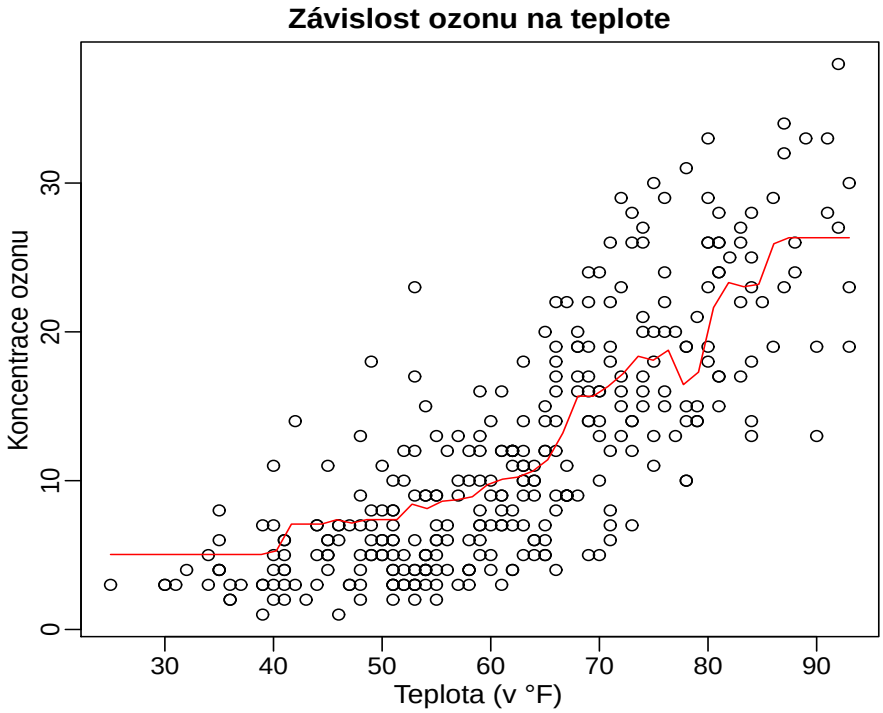
$$F_k(\boldsymbol{x}) = F_{k-1}(\boldsymbol{x}) + \nu \sum_{m=1}^{M} b_{km} I(\boldsymbol{x} \in R_{km}) \tag{4.20}$$

Je zřejmý, že s rostoucím ν klesá počet stromů K, a obráceně. Zároveň podle [7] by model s menším ν a k tomu odpovídajícímu K by měl mít nižší chybu na testovacích datech, než model s $\nu = 1$, tj. model, který optimalizuje chybu na trénovacích datech.

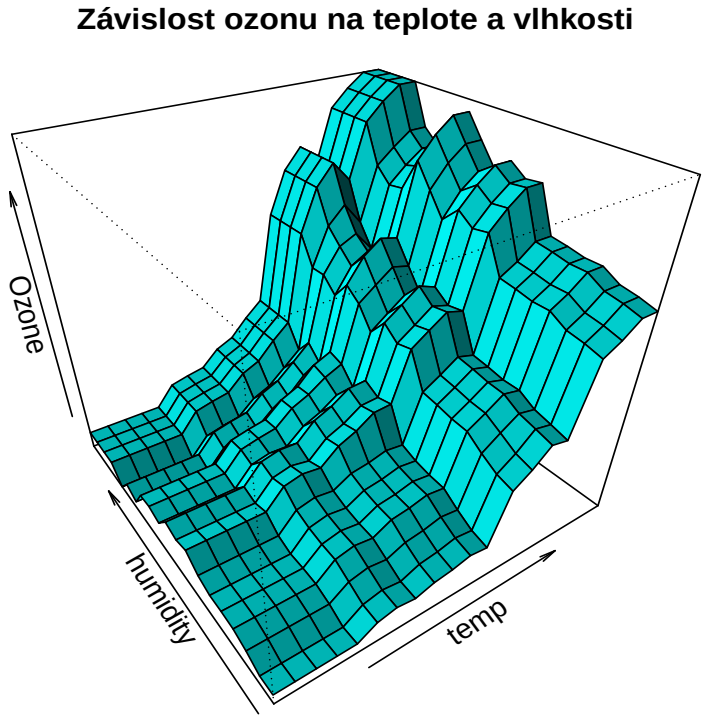
Poslední věcí, o které se zmíníme, je způsob, jakým se určí významnost proměnných. Stejně jako metoda CART i boosting využívá k určení důležitosti proměnné velikost snížení nepřesnosti stromů. Na rozdíl od stromu CART, který sčítá poklesy nepřesnosti i pro proměnné, které nakonec nebyly použité k dělení, boosting sčítá poklesy nepřesnosti jenom v těch uzlech, kde došlo k dělení dle dané proměnné. Takový součet je spočítán pro všechny stromy v modelu a následně zprůměrován.

Tuto změnu si můžeme dovolit proto, že máme k dispozici K stromů, a ne jenom jeden, jako v případě metody CART. Navíc použití zmenšování pomáhá odkrývat skryté závislosti mezi závislou a nezávislými proměnnými, proto není potřeba zahrnovat i nerealizované poklesy nepřesnosti.

Jelikož pokles nepřesnosti stromů, respektive lesů není relativní mírou významnosti, je zvykem přiřadit proměnné s největším průměrným součtem poklesů nepřesnosti hodnotu 100, a významnost ostatních proměnných se vztahuje k této proměnně.



Obrázek 4.1: Odhad závislosti koncentrace ozónu na teplotě pomocí boostingu. Ačkoliv na grafu je výsledná funkce spojitá, ve skutečnosti tomu tak není.



Obrázek 4.2: Aproximace dat pomoci boostingu

4.3 Náhodný les

Náhodný les (*random forest*) je další metoda, která byla vyvinutá Leo Breimanem. I ona využívá kolektivní rozhodování, kdy výsledek obdržíme po kombinací výsledků vícero modelů.

Jak už víme, kolektivní metody bagging i boosting nejsou určené jenom pro stromy a dají se s větším či menším úspěchem použit i pro jiné typy modelů. Naproti tomu náhodný les je technika, která se používá výhradně pro stromy.

Princip, na kterém pracuje náhodný les, je podobný baggingu. Nejdříve pomocí bootstrapových výběrů $\mathcal{L}_1^{(B)}, \ldots, \mathcal{L}_K^{(B)}$ vybuduje K stromů. Poté se pro dané pozorování $\boldsymbol{x}$ a pro všechna k vypočtou hodnoty $\hat{y}_k = T(\boldsymbol{x}, \mathcal{L}_k^{(B)})$, a spočítá se průměr přes všechny předpovědi $\hat{y}_1, \ldots, \hat{y}_K$. Tento průměr je pak označen jako odhad skutečné hodnoty y na základě pozorování $\boldsymbol{x}$.

To, čím se od sebe bagging a náhodný les liší, je způsob, jakým se budují jednotlivé stromy v lesu. Stromy v baggingu vznikají stejně jako samostatný strom CART, tj. nejdříve se z dostupného vzorku dat $\mathcal{L}_k$ vytvoří maximální strom. Tento strom je následně prořezán a pomocí pozorování, které se nedostaly do $\mathcal{L}_k$, je vybrán strom o optimální velikosti.

Oproti tomu náhodný les využívá metody *náhodného výběru vysvětlujících proměnných* (*random features selection*). Nejdříve zvolíme parametr m_0, $m_0 \in \{1, \ldots, J\}$. Tento parametr určuje, mezi kolika náhodně vybranými nezávislými proměnnými se bude hledat nejlepší dělení. V každém uzlu se náhodně zvolí m_0 proměnných, z nich se vybere ta, na základě které se provede nejlepší dělení, a tento postup se opakuje, dokud nejsou uplatněná zastavovací kritéria. Vzniklý strom se ale na rozdíl od stromu CART už zpětně neprořezává.

Použitelnost náhodného lesu vychází z následujících dvou vět.

Věta 4.1 *Pokud počet stromů K roste do nekonečna, skoro jistě platí*

$$\mathsf{E}_{\boldsymbol{X},Y}(Y - \frac{1}{K}\sum_{k=1}^{K} T(\boldsymbol{X}, \mathcal{L}_k))^2 \to \mathsf{E}_{\boldsymbol{X},Y}(Y - \mathsf{E}_{\mathcal{L}} T(\boldsymbol{X}, \mathcal{L}))^2, \tag{4.21}$$

kde $\mathcal{L}$ je náhodný výběr, $\mathcal{L} = \{(\boldsymbol{X}_i, Y_i,\ i = 1, \ldots, N\}$ a $\mathsf{E}_{\mathcal{L}}$ je střední hodnota, kdy za náhodnou veličinu považujeme výběr $\mathcal{L}$. Výraz $\mathcal{L}_k$ pak označuje k-tou realizaci náhodného výběru $\mathcal{L}$.

Důkaz této věty vychází ze zákona velkých čísel.

Druhá věta obhajuje způsob výběru proměnných pro dělení jednotlivých uzlů. Označme pravou stranu rovnice (4.21) symbolem $L(RF)$ a pojmenujme to *zobecněná nepřesnost lesu*. Dále označme

$$L(Tr) = \mathsf{E}_{\mathcal{L}}\mathsf{E}_{\boldsymbol{X},Y}(Y - T(\boldsymbol{X}, \mathcal{L}))^2$$

jako *zobecněná nepřesnost stromu*. Vztah mezi takto definovanými nepřesnostmi lesu a stromu definuje následující věta.

Věta 4.2 *Předpokládejme, že pro všechny $\mathcal{L}$ platí $\mathsf{E}Y = \mathsf{E}_{\boldsymbol{X}} T(\boldsymbol{X}, \mathcal{L})$. Pak*

$$L(RF) \leq \ \bar{\rho} L(Tr), \tag{4.22}$$

kde $\bar{\rho}$ *je vážená korelace mezi rezidui* $(Y - T(\boldsymbol{X}, \mathcal{L}))$ *a* $(Y - T(\boldsymbol{X}, \mathcal{L}'))$. *Výběry* $\mathcal{L}$ *a* $\mathcal{L}'$ *jsou nezávislé. Váženou korelaci definujeme jako*

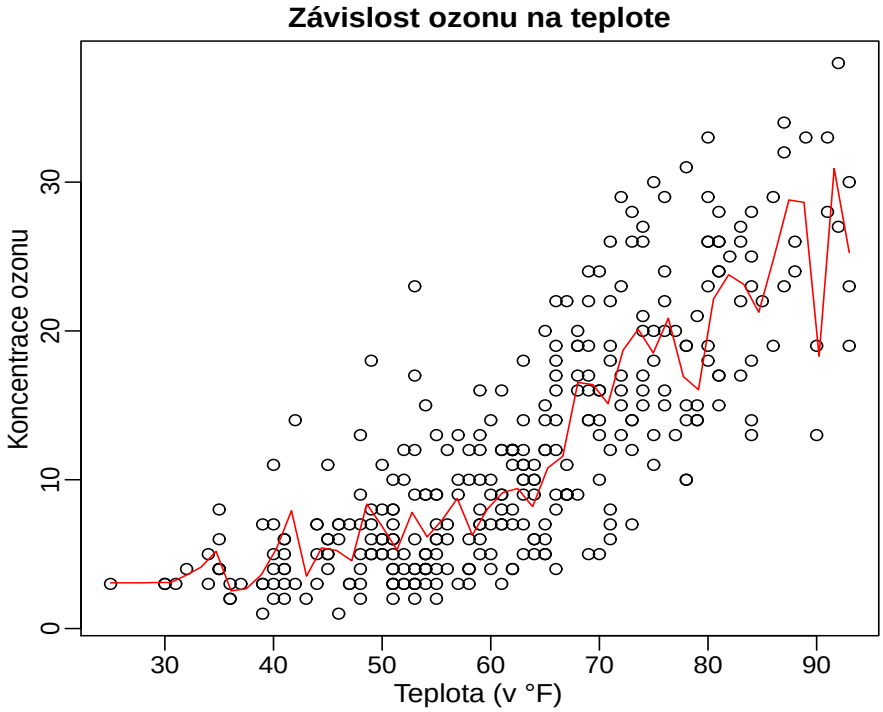
$$\bar{\rho} = \frac{\mathsf{E}_{\mathcal{L}}\mathsf{E}_{\mathcal{L}'}(\rho(\mathcal{L}, \mathcal{L}')sd(\mathcal{L})sd(\mathcal{L}'))}{\mathsf{E}_{\mathcal{L}}(sd(\mathcal{L}))^2},$$

kde $sd(\mathcal{L}) = \sqrt{\mathsf{E}_{\boldsymbol{X},Y}(Y - T(\boldsymbol{X}, \mathcal{L}))^2}$ *a* $\rho(\mathcal{L}, \mathcal{L}')$ *je korelační koeficient reziduí* $(Y - T(\boldsymbol{X}, \mathcal{L}))$ *a* $(Y - T(\boldsymbol{X}, \mathcal{L}'))$.

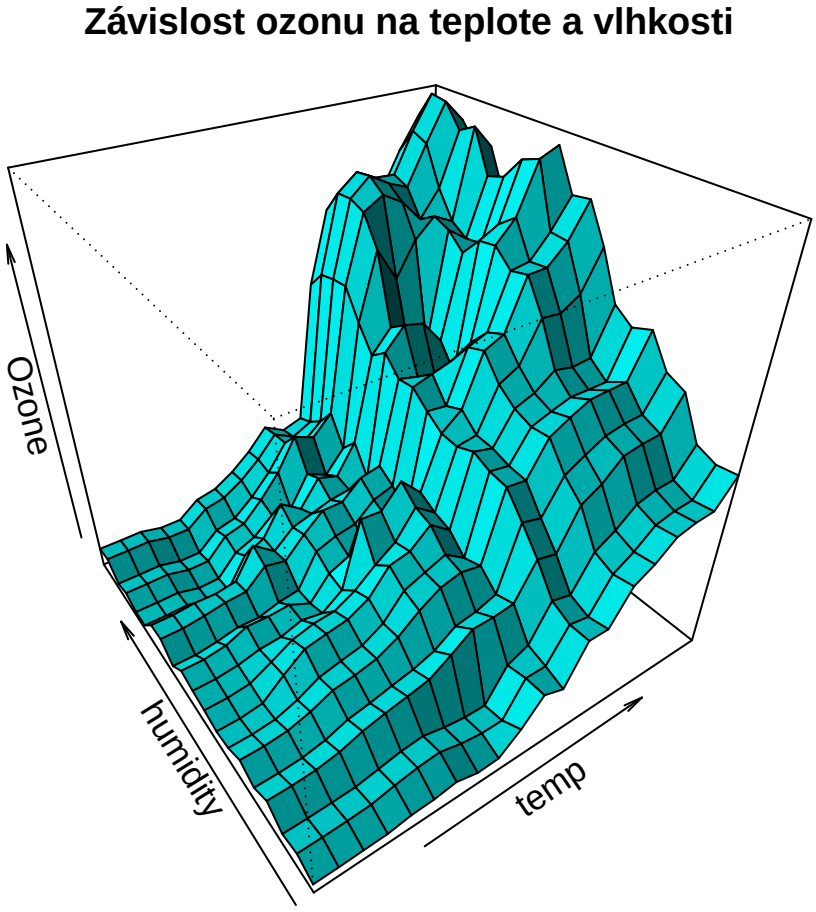
Důkaz věty (4.22) nalezneme v [4].

Tato věta určuje to, co ovlivňuje celkovou nepřesnost regresního lesu: nízká nepřesnost jednotlivých stromů v lese a nízký korelační koeficient mezi rezidui stromů. Z toho vyplývá také to, proč se stromy v náhodném lesu neprořezávají a proč je výběr proměnných pro dělení náhodný. Neprořezaný strom má nejnižší možnou nepřesnost na vývojových datech. U samostatného stromu CART nám tato skutečnost vadila, protože snižovala použitelnost stromu na jiných datech. Zde se ale snažíme minimalizovat chybu celého lesu, proto můžeme vynechat prořezávání stromů. Pomocí náhodného výběru proměnných pro dělení se snažíme snížit korelační koeficient mezi rezidui, protože ten závisí na realizaci náhodného výběru $\mathcal{L}$.

Významnost proměnných v regresním lesu je založená na tom, jak výrazný pokles nepřesnosti dokáže konkrétní proměnná způsobit. Nejdříve se vypočítá pokles pro všechny stromy v lese a následně se zprůměruje.



Obrázek 4.3: Odhad závislosti koncentrace ozónu na teplotě pomocí náhodného lesu. Obdobně jako v bagginu i zde by měla být výsledná funkce po částech konstantní. Červená křivka zobrazuje předpovědi náhodného lesu. Tento průběh funkce jsme dostali, když $|\mathcal{L}_k^{(B)}| = 330$.



Obrázek 4.4: Aproximace dat pomocí náhodného lesu v případě dvou proměnných. Velikost $\mathcal{L}_k^{(B)}$ je 330.

5. Praktická část

Tato část je věnována aplikaci výše uvedených metod na reálná data. Cílem je srovnat schopnost jednotlivých technik poradit si s různými typy dat. Budeme chtít srovnat, jak přesně dokážou modely aproximovat data, a také, které z vysvětlujících proměnných budou jednotlivými technikami označené jako nejvýznamnější. Celkem vyzkoušíme všech šest metod, které jsme popsali v předchozích kapitolách: lineární regresi, CART, MARS, bagging, boosting a náhodný les.

5.1 Porovnávaní vlastností metod

Tato sada údajů pochází z [15]. Jedná se o hodnoty z měření koncentrace ozónu v Los Angeles v roce 1976 a byly použité Leo Breimanem v [5]. Celkem obsahují 330 pozorování, a kromě koncentrace ozónu máme hodnoty dalších 9 proměnných.

Název proměnné	Popis proměnné
O3	maximální koncentrace ozónu během dne (v ppm - počet částic na jeden milion (*parts per million*))
vh	výška rtuti v barometru
wind	rychlost větru (míle za hodinu)
humidity	vlhkost v %
temp	teplota (ve stupních Fahrenheita)
ibh	výška inverzní vrstvy (ve stopách)
dpg	barický (tlakový) gradient
ibt	teplota inverzní vrstvy (ve stupních Fahrenheita)
vis	viditelnost (v mílích)
doy	den roku

Tabulka 5.1: Seznam proměnných dat Ozone

Postup hledání nejlepšího modelu byl zvolen následující. Data byla náhodně rozdělená na trénovací vzorek o velikosti 275 pozorování, a testovací vzorek, který obsahoval zbývajících 55 pozorování. Model byl vyvinut na trénovacím vzorku, a následně byly vytvořeny předpovědi pro testovací vzorek. Tyto předpovědi byly porovnány se skutečnými naměřenými hodnotami.

Tento postup byl zopakován celkem 1500×, a výsledné hodnoty byly zprůměrovány.

Je řada kritérií, pomocí kterých lze měřit kvalitu aproximace. Zde jako míra nepřesnosti bude sloužit střední čtvercová chyba. Střední čtvercová chyba byla vybrána pro svůj snadný výpočet, jasnou interpretaci a také proto, že v případě stromů je to nejčastěji používaná míra nepřesnosti.

V kapitole o lineární regresi jsme uvedli, že pomocí koeficientu determinace vypočteného dle vzorce (2.5) lze také měřit kvalitu predikce. Důležité ovšem je, aby model obsahoval absolutní člen nebo aby $\mathbf{1} \in \mathcal{M}(\boldsymbol{X})$. Bez těchto předpokladů výše uvedený vzorec nemá smysl. Jejich platnost ale u všech metod nemůžeme zaručit, proto toto kritérium nepoužijeme.

Pojem střední čtvercová chyba se již několikrát v této práci objevil. Střední čtvercová chyba stromu se rovná

$$MSE(T) = \mathsf{E}(Y - T(\boldsymbol{X}))^2.$$

V případě konečného souboru pozorování pak používáme *průměrnou čtvercovou chybu*:

$$\widehat{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - T(\boldsymbol{x}_i))^2.$$

Kromě přesnosti modelů počítáme také významnost proměnných. Jelikož se jedná o různé techniky s různými způsoby výpočtů důležitosti proměnných, nelze výsledky přímo porovnat. Proto byla zvolena následující úprava. Pro každou z technik se důležitost proměnných vypočte pro ně typickým způsobem. Pro lineární regresi zobrazíme hodnotu t-statistiky, zbylé techniky používají míru poklesu nepřesnosti modelu.Protože ale nemůžeme přímo porovnávat velikost poklesu nepřesnosti pro jednotlivé proměnné v různých technikách, provedeme úpravu, která je popsaná v [6]. Hodnota pro pokles nepřesnosti nejvýznamnější proměnné obdrží hodnotu 100, a velikost poklesu pro ostatní proměnné bude vyjádřená jako podíl z tohoto největšího poklesu.

Veškeré testy jsou prováděny pomocí programu *R project*, volně dostupného na *http://www.r-project.org/*. Jmenovitě pak použijeme balíčky [12], [14], [13], [15] a [16].

Předtím, než uvedeme výsledky testování, zmíníme se krátce o odhadu optimálního počtu stromů v lesích.

5.1.1 Velikost regresního lesu

Důležitou otázkou je, kolik stromů je potřeba v případě baggingu, boostingu a náhodného lesu. Počet stromů závisí na konkrétní úloze a na dalších použitých parametrech.

Postup je stejný, jako výše popsaný. Nejdříve jsme rozdělili pozorování na trénovací a testovací vzorek, na trénovacím jsme vyvinuli model, a na testovacím ho ověřili. Tento postup jsme zopakovali celkem 1500×. Obdržené průměrné čtvercové chyby jsme pak zprůměrovali.

V tabulce 5.2 je uveden průměr průměrných čtvercových chyb pro různé velikosti lesu vytvořeného metodou bagging. Zároveň bylo testováno, jak ovlivní přesnost modelu počet pozorování v trénovacím vzorku.

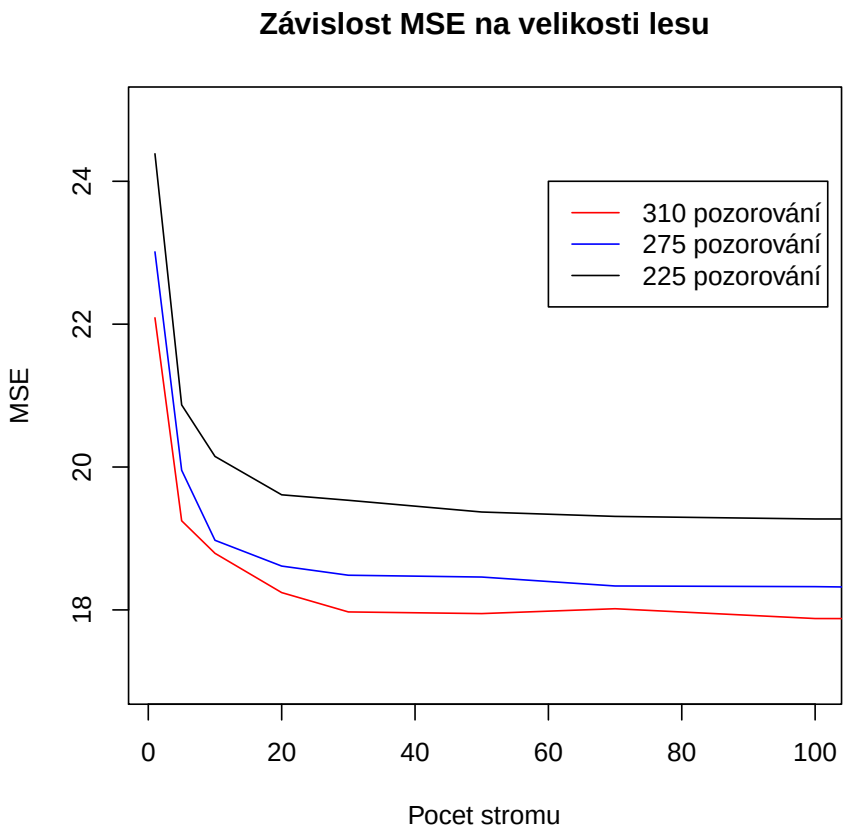
Jak vidíme, s rostoucím počtem stromů roste i přesnost modelu. Avšak toto zlepšení je po překročení 50 stromů nevýznamné - pro trénovací vzorek o velikosti 275 je rozdíl průměrné čtvercové chyby pro lesy o velikosti 50 a 500 pouhých 0,18. Lze tedy souhlasit se závěrem plynoucím z [2], že pro regresní úlohy nám postačí les o velikosti několika desítek stromů.

Hodnota $\widehat{MSE}$ je ovlivněná nejenom počtem stromů v lese, ale také počtem pozorování v trénovacím vzorku. Celkem očekávaně s rostoucím počtem pozorování hodnota $\widehat{MSE}$ klesá. Větší trénovací vzorek poskytuje více informaci pro

sestrojení modelu, který dokáže lépe odhadnout vztah mezi závislou a nezávislými proměnnými.

Počet stromů v lese	Velikost trénovacího vzorku		
	310	275	225
1	22,09	23,01	24,39
5	19,25	19,96	20,87
10	18,79	18,97	20,15
20	18,24	18,61	19,61
30	17,97	18,49	19,53
50	17,95	18,46	19,37
70	18,02	18,33	19,31
100	17,88	18,33	19,27
150	17,87	18,27	19,27
200	17,75	18,30	19,27
300	17,77	18,29	19,26
500	17,78	18,28	19,23

Tabulka 5.2: Hodnoty $\widehat{MSE}$ pro různě velké lesy při použití na data Ozone - metoda bagging.



Obrázek 5.1: Závislost $\widehat{MSE}$ na počtu stromů v lese - bagging.

V případě boostingu máme více parametrů, které musíme zvolit. Jde o parametr ν, který udává, o kolik snížíme příspěvek jednoho stromu, dále velikost stromu M', která je stejná pro všechny stromy v lese, a také celkový počet stromů K. Programové vybavení R neumožňuje nastavit velikost stromu jako počet listů, ale jako počet pater stromu. Proto zde parametr M' neoznačuje počet základních funkcí ve výsledné množině, ale počet pater stromu.

Abychom našli vhodné parametry, zvolili jsme následovný postup. Vybrali jsme tři hodnoty parametrů $\nu = 1,\ 0,1,\ 0,01$, a čtyři velikosti stromů $M' = 1, 3, 5, 9$. V kapitole věnované boostingu jsme se zmínili, že parametr ν a počet stromů spolu souvisí, čím menší je parametr ν, tím víc stromů by měl mít les, aby výsledný model měl nízkou nepřesnost. Optimální velikost lesu by se tak měla lišit pro každý ze tři zvolených ν. Z tohoto důvodu se K pohybuje v rozmezí od 10 do 2000 stromů.

Celkem jsme tedy vyzkoušeli 12 modelů s různou kombinací těchto parametrů. Cílem bylo vybrat takovou kombinaci, která by zaručovala maximální přesnost aproximace při co nejjednodušším modelu.

V následující tabulce jsou shrnuté výsledky našeho zkoumání.

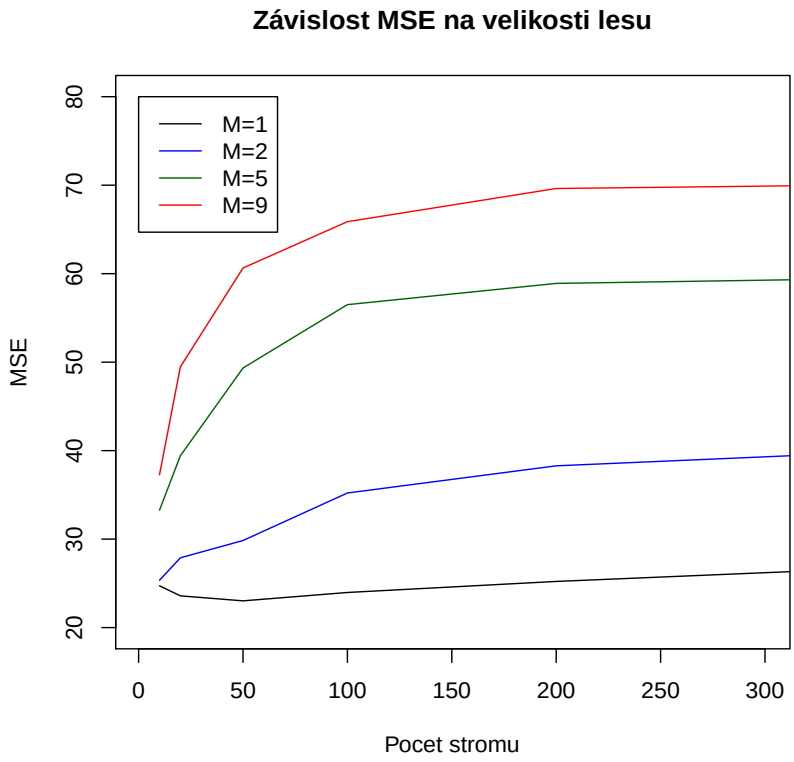
ν	M'	K							
		10	20	50	100	200	500	1000	2000
1	1	24,72	23,59	**23,02**	23,97	25,22	28,18	30,55	31,09
	2	25,35	27,89	29,84	35,21	38,28	41,34	42,74	40,31
	5	33,27	39,42	49,34	56,50	58,90	60,00	61,21	60,50
	9	37,27	49,47	60,64	65,88	69,63	70,44	72,00	69,74
0,1	1	32,00	23,69	18,81	17,11	**16,21**	16,35	17,13	18,47
	2	27,98	20,00	16,33	**15,67**	15,72	16,64	17,58	18,60
	5	24,85	17,87	15,68	15,87	16,65	17,53	17,87	17,99
	9	23,91	17,37	15,95	16,54	17,13	17,69	17,92	18,00
0,01	1	59,19	53,94	42,75	32,27	23,81	18,62	16,80	15,88
	2	58,26	52,23	39,47	28,24	19,96	16,01	**15,16**	15,27
	5	57,35	50,67	36,58	25,00	17,78	15,22	15,34	15,89
	9	57,11	50,27	35,96	24,18	17,21	15,36	15,75	16,41

Tabulka 5.3: Hodnoty $\widehat{MSE}$ pro různě velké lesy při použití na data Ozone - metoda boosting.

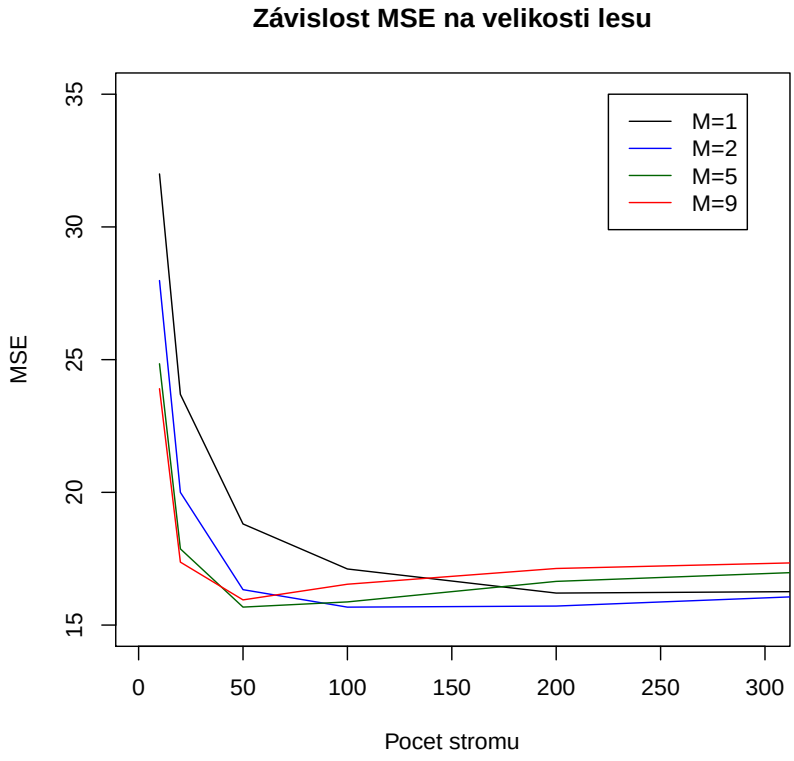
První čtyři řádky shrnují výsledky pro model s $\nu = 1$. To znamená, že tento model byl sestrojen tak, aby měl co nejmenší chybu na trénovacích datech. Nejnižší hodnota $\widehat{MSE}$ je pro kombinaci $M' = 1$ a $K = 50$ a je rovná 23,02. Tato hodnota je ale o poznání větší než nejlepší výsledky modelu s $\nu = 0,1$. Mezi modely s $\nu = 0,1$ je nejlepší kombinace $M' = 2$ a $K = 100$ stromů (15,67). Pro $\nu = 0,01$ je nejlepší kombinace $M' = 2$ a $K = 1000$ (15,16).

Nicméně, kromě průměrné čtvercové chyby musíme brát v úvahu také početní náročnost a celkovou složitost modelu. Není těžké odvodit, že model s více patry bude složitější na výpočet než model s jedním patrem, čili jenom dvěma listy. Z tohoto důvodu pro další testování zvolíme ten model, který je dostatečně přesný,

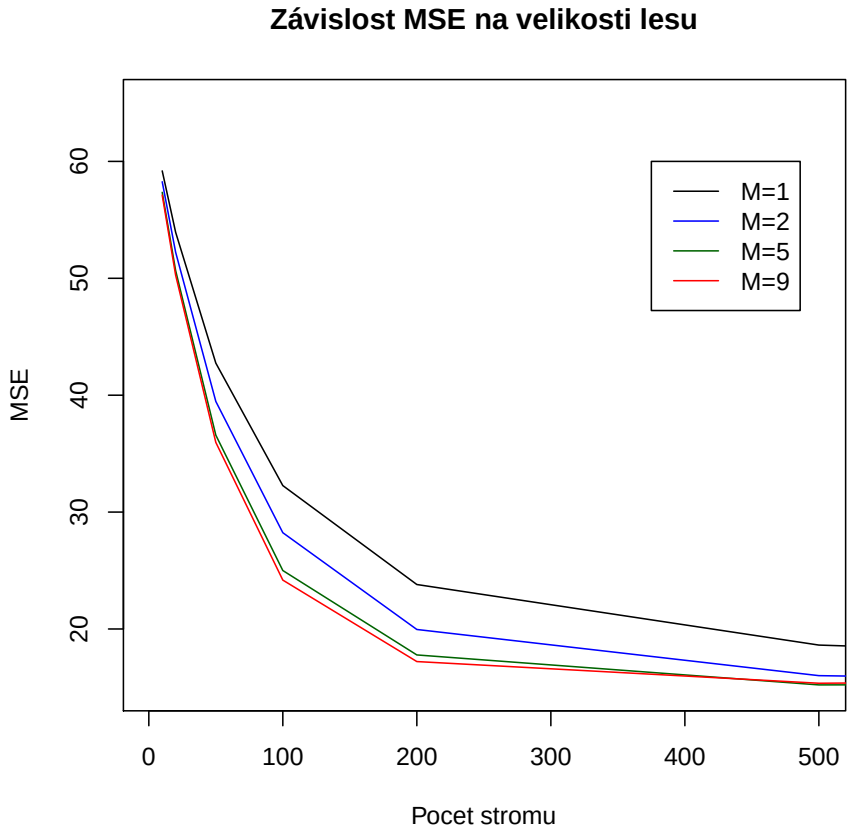
ale zároveň není výpočetně náročný. Proto byl zvolen model s $\nu = 0,1$, se dvěma listy a celkovým počtem stromů 200, jehož hodnota $\widehat{MSE}$ byla 16,21.



Obrázek 5.2: Závislost $\widehat{MSE}$ na počtu stromů a velikosti stromů, při $\nu = 1$



Obrázek 5.3: Závislost $\widehat{MSE}$ na počtu stromů a velikosti stromů, při $\nu = 0,1$



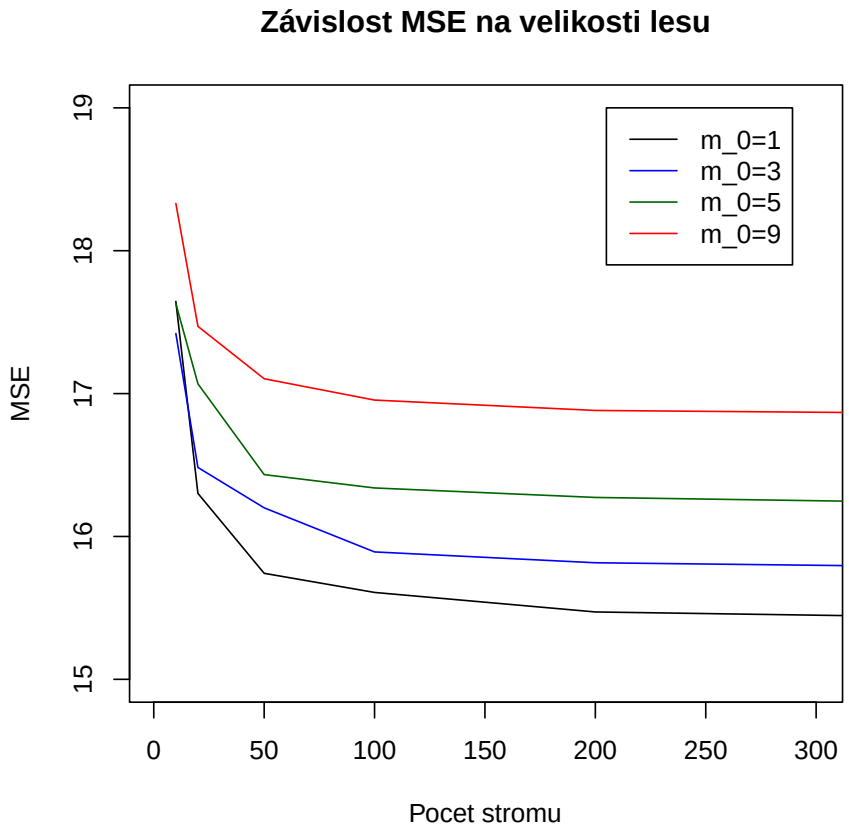
Obrázek 5.4: Závislost $\widehat{MSE}$ na počtu stromů a velikosti stromů, při $\nu = 0,01$

U posledního modelu, náhodného lesu, máme dva parametry, které můžeme nastavit. Jeden je počet stromů v lese, a druhý je parametr m_0, který řídí počet náhodně zvolených proměnných, mezi nimiž se vybere nejlepší dělení. Postupně vyzkoušíme $m_0 = 1, 3, 5, 9$. Výsledky jsou shrnuté v následující tabulce

Počet stromů v lese	m_0			
	1	3	5	9
10	17,50	17,51	17,63	18,33
20	16,30	16,48	17,07	17,47
50	15,74	16,20	16,43	17,10
100	15,61	15,89	16,34	16,95
200	15,47	15,82	16,27	16,88
500	15,40	15,76	16,20	16,84
1000	15,42	15,74	16,20	16,81
2000	15,40	15,73	16,19	16,79

Tabulka 5.4: Hodnoty $\widehat{MSE}$ pro různě velké lesy při použití na data Ozone - metoda náhodný les.

V souladu s teorií vidíme, že s rostoucím počtem stromů roste i přesnost modelu. Po překročení určitého počtu stromů ale tento nárůst není významný. Proto zvolíme $K = 50$. Co se týče parametrů m_0, tam je situace poměrně jasná. Pro $m_0 = 1$ je $\widehat{MSE}$ nejnižší pro libovolný počet stromů.



Obrázek 5.5: Závislost $\widehat{MSE}$ na počtu stromů v lese - náhodný les.

5.1.2 Výsledky a komentáře

V této podkapitole uvedeme hlavní výsledky testování. Na základě výsledků z předchozí kapitoly jsme stanovili tyto parametry regresních lesů: pro bagging se jedná o $K = 30$, pro boosting $\nu = 0,1$, $M' = 1$ a $K = 200$, a pro náhodný les $m_0 = 1$ a $K = 50$.

Predikční schopnosti

Nejdříve se podíváme na predikční schopnosti jednotlivých technik. Pro stanovení průměrné čtvercové chyby použijeme postup, který jsme popsali na začátku.

Pro nalezení výsledného modelu lineární regrese jsme použili krokovou (*stepwise*) regresi, konkrétně proceduru *step* implementovanou v programu *R*. Tato procedura hledá model s nejmenší hodnotou *AIC* - Akaikeho informační kritérium (viz [18], s 142). Toto kritérium bylo navrženo jako

$$AIC = -l(\hat{\boldsymbol{\theta}}) + 2q,$$

kde l je logaritmická věrohodnostní funkce a q je počet složek maximálně věrohodného odhadu $\hat{\boldsymbol{\theta}}$. V případě lineárního normálního modelu s neznámým rozptylem máme

$$AIC = N(1 + \log(2\pi\hat{\sigma^2})) + 2(r + 1),$$

kde $\hat{\sigma^2}$ je odhad σ^2 metodou maximální věrohodnosti a r je hodnost matice $\mathbb{X}$.

Při každém opakování se nejdříve sestrojil model se všemi dostupnými proměnnými, následně pomocí *stepwise* algoritmu se vybral ten optimální, který byl použit při předpovědi hodnot v testovacím vzorku.

V tabulce 5.5 jsou uvedené průměrné čtvercové chyby pro všech šest modelů.

Metoda	$\widehat{MSE}$
Lineární regrese	20,28
CART	22,72
MARS	16,17
Bagging	18,26
Boosting	16,19
Náhodný les	15,87

Tabulka 5.5: Hodnota $\widehat{MSE}$ pro různě velké lesy při použití na data Ozone.

Model s nejnižším průměrnou čtvercovou chybou je náhodný les, následovaný stromem MARS a boostingem. Strom CART se ukázal jako nejnevhodnější přístup. Projevila se nevýhoda metody CART, a to horší schopnost poradit si s daty, kde existuje nějaká silnější lineární závislost mezi závislou a nezávislými proměnnými a proto je tu také velký rozdíl mezi metodou CART a MARS. Strom MARS díky tomu, že se jedná o spojitý model, který aproximuje data pomocí přímek, dokázal mnohem lépe odhadnout vztah mezi vysvětlující a vysvětlovanou proměnnou.

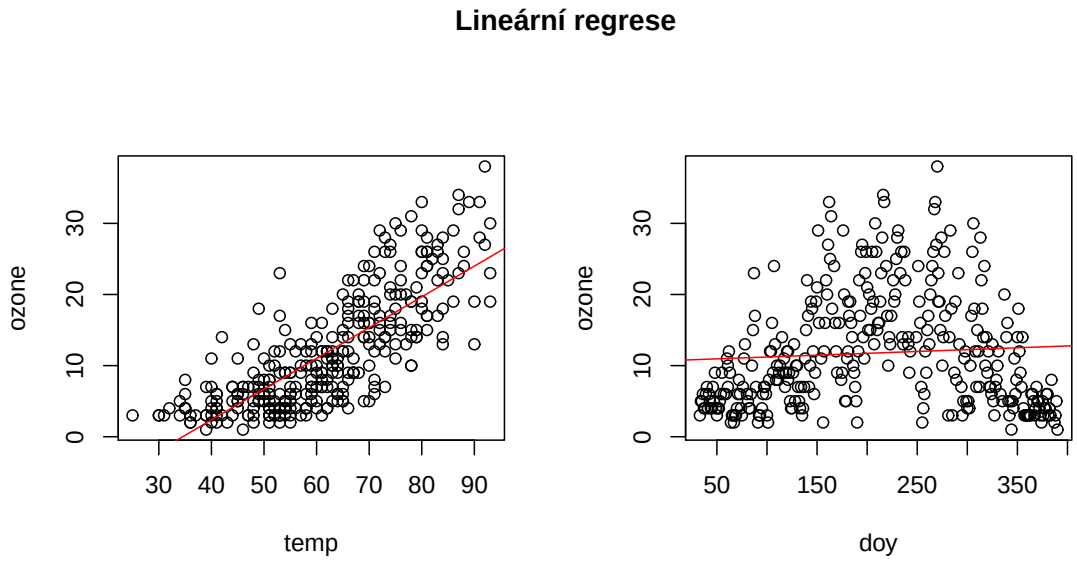
Použití baggingu přineslo zlepšení oproti samotnému stromu CART, ale i tento model má horší predikční schopnosti než metoda MARS, boosting nebo náhodný

les. Velikost boostrapového výběru pro jednotlivé stromy byla stejná jako velikosti trénovacích dat, tj. s opakováním bylo vybráno 275 pozorování. Byly vyzkoušeny i jiné velikosti, 50 a 250, ale tyto změny nepřinesly žádné výrazné zlepšení nebo zhoršení.

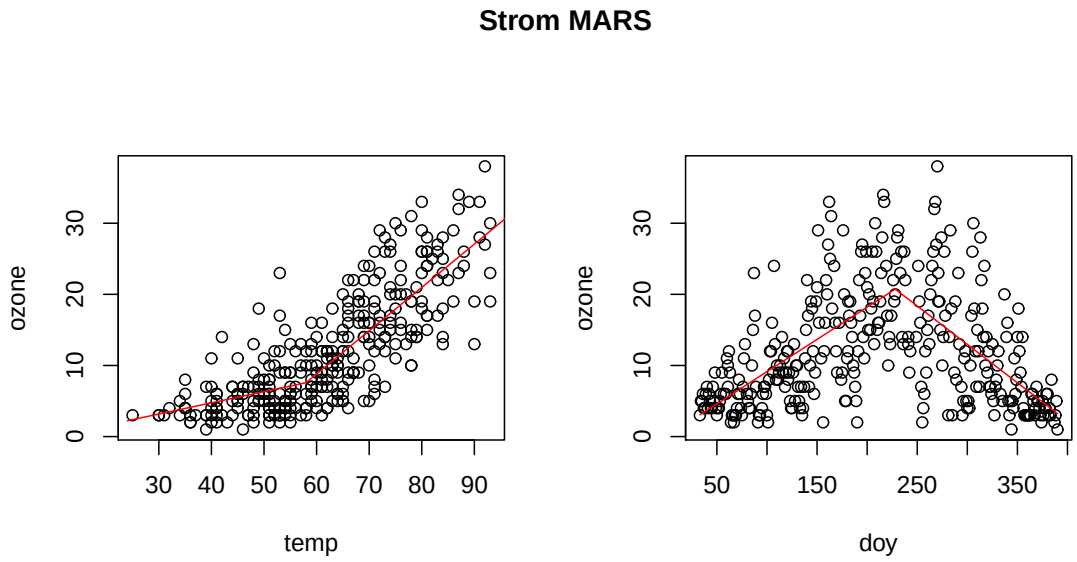
V podkapitole 4.1 jsme uvedli, že Breiman v [2] uvádí, že při použití baggingu klesla vypočtené střední čtvercové chyby o 21 % až 46 % u jím testovaných dat oproti metodě CART. V našem případě je tento pokles přibližně 20%. Můžeme proto potvrdit, že použití baggingu značně zvyšuje predikční schopnosti oproti samostatnému stromu CART.

Srovnání lineární regrese a stromu MARS

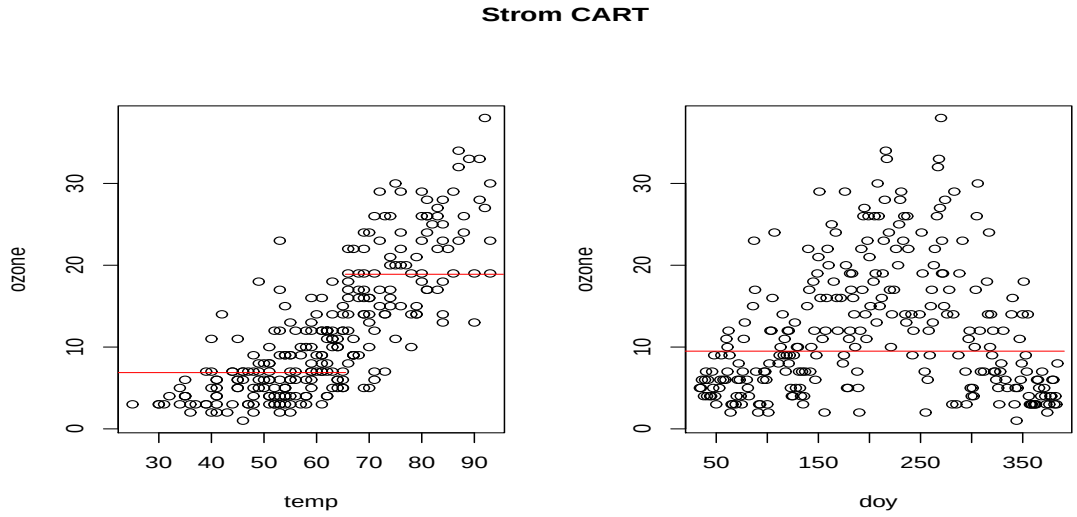
V této části provedeme srovnání lineární regrese a stromu MARS, neboť tyto metody jsou si podobné ve smyslu, že se snaží aproximovat vztah mezi závislou a nezávislými proměnnými pomocí přímky, respektive spojité křivky po částech lineární, která minimalizuje součet druhých mocnin rozdílů mezi skutečnými pozorováními y a předpovědmi $\hat{y}$. Strom MARS je viditelně přesnější. Následující grafy ukazují, jak modelují lineární regrese a strom MARS závislost koncentrace ozónu na teplotě (*temp*), respektive na dni v roce (*doy*). Pro ilustraci je přidán i stejný graf pro strom CART.



Obrázek 5.6: Závislost koncentrace ozonu na proměnných *temp* a *doy* - lineární regrese



Obrázek 5.7: Závislost koncentrace ozonu na proměnných *temp* a *doy* - strom MARS



Obrázek 5.8: Závislost koncentrace ozonu na proměnných *temp* a *doy* - strom CART

V případě proměnné *temp* je proložená přímka celkem věrně kopíruje data, a je stejně přesná jako strom MARS. Je to dáno tím, že závislost koncentrace ozónu na teplotě se zdá být lineární, což je přesně typ závislosti, které dokáže lineární regrese dobře modelovat. Jiné to ale je v případě proměnné *doy*. Koncentrace ozónu vykazuje nelineární závislost na této proměnné, proto proložená přímka lineární regrese je horší, než po částech lineární křivka stromu MARS.

Dá se tedy říct, že strom MARS dokáže díky použití lineárních splajnů modelovat nelineární závislosti mezi proměnnými, a v případě lineární závislosti je metoda MARS srovnatelná s lineární regresí. Nicméně při hledání nejlepšího

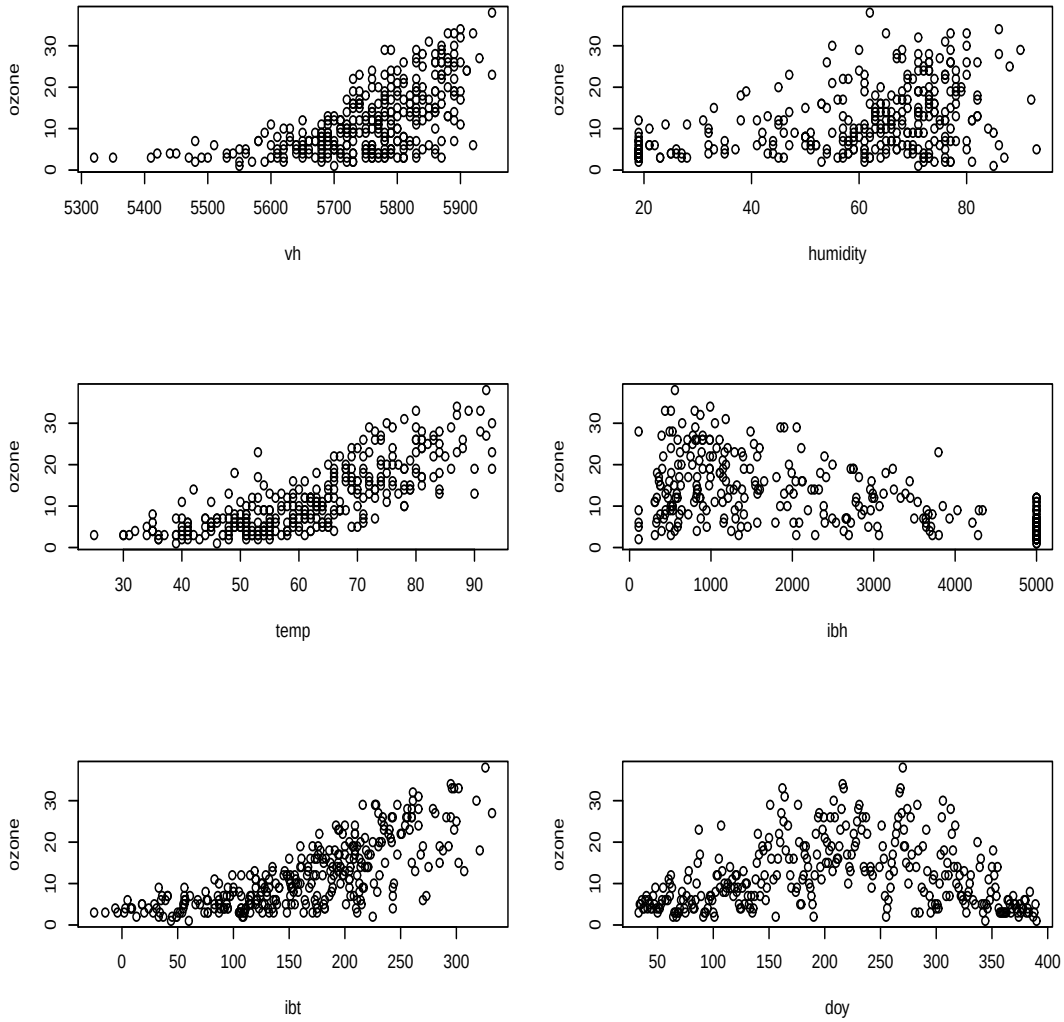
regresního modelu také často používáme nejrůznější transformace proměnných a interakce mezi nimi, které mohou vylepšit obyčejný lineární model.

Transformace vybraných proměnných v lineární regresi

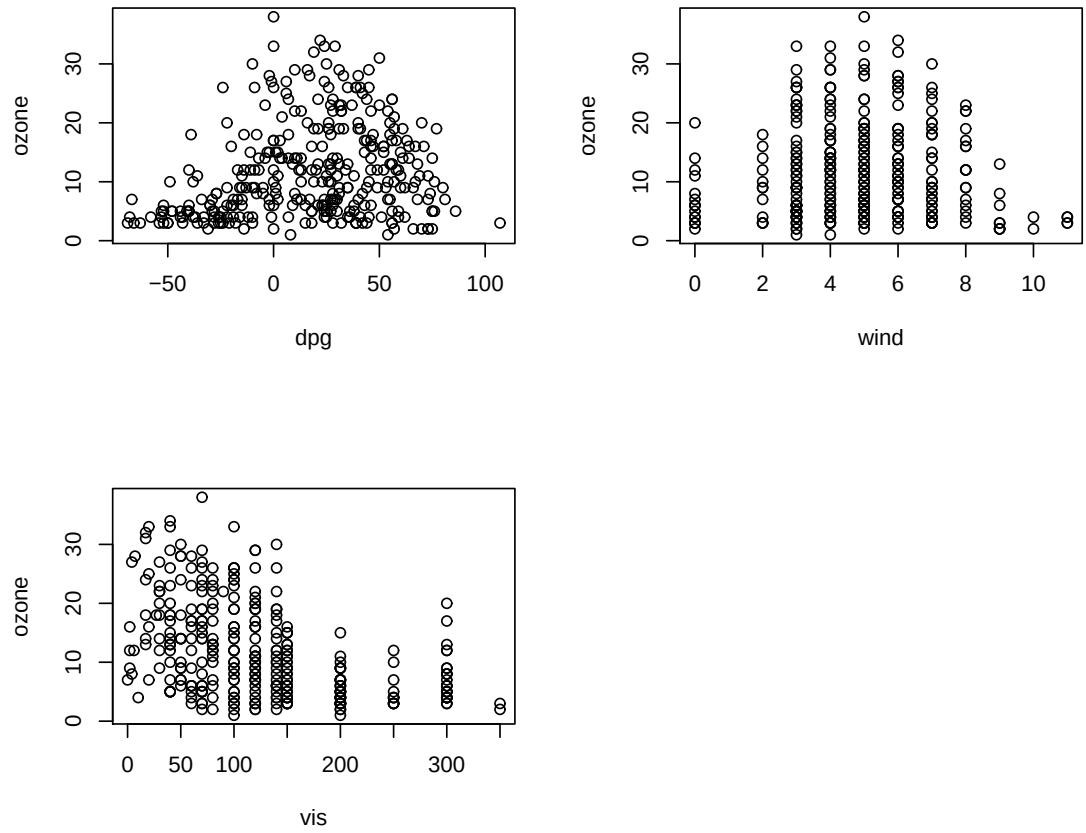
Zkusíme odpovědět na otázku, zda můžeme vylepšit model lineární regrese pomocí nějakých transformací. Případně, zda vynechání stepwise procedury a tedy ponechání všech dostupných proměnných v modelu dokáže zlepšit predikční schopnosti.

Prosté ponechání všech proměnných v modelu nezlepšilo jeho predikční schopnosti. Průměrná čtvercová chyba zůstala takřka stejný ($\widehat{MSE} = 20,40$), jako v případě modelu s použitím *stepwise* procedury ($\widehat{MSE} = 20,28$).

Transformace naopak přinesla viditelné zlepšení. Do modelu byly přidané následující transformace: proměnné *humidity*, *vh*, *ibh* a *temp* byly zlogaritmované, proměnné *wind*, *dpg* a *vis* byly umocněné na druhou. Tato úprava vyplývá z obrázků 5.9 a 5.10. Takto rozšířený model byl znovu upraven pomocí stepwise procedury. Díky těmto transformacím $\widehat{MSE}$ klesl na 18,27.



Obrázek 5.9: Závislost koncentrace ozónu na vysvětlujících proměnných



Obrázek 5.10: Závislost koncentrace ozónu na vysvětlujících proměnných

Interakce mezi proměnnými

Nyní zkusíme do modelu lineární regrese bez transformací přidat interakce (součin proměnných) mezi proměnnými. Bylo vyzkoušeno přidání následujících interakcí - *humidity*:*temp*, *humidity*:*wind*, *humidity*:*doy*, *doy*:*vh*, *ibt*:*temp*, *ibt*:*doy*, *temp*:*doy*. Tyto interakce byly zvolené na základě fyzikálních vlastností, vlhkost vzduchu závisí na teplotě a větru, tlak závisí na ročním období apod. Model s těmito interakcemi měl průměrnou čtvercovou chybu 16,87 ($\widehat{MSE}$ původního modelu je 20,28).

Z výše popsaného můžeme udělat závěr, že pomocí interakcí a transformací lze zvýšit predikční schopnosti lineárního modelu, avšak ani jedna úprava nepřekonala strom MARS nebo náhodný les.

Stabilita metod

Stromy jsou označované jako nestabilní metody, tj. že i při malé změně vývojových dat může výsledný model doznat značné změny. Abychom to ověřili, uložíme $\widehat{MSE}$ z každého z 1500 opakování, a následně spočítáme směrodatnou odchylku pro takto vypočítané průměrné čtvercové chyby. Čím větší bude odchylka, tím nestabilnější je metoda. V tabulce 5.6 jsou vypočtené směrodatné odchylky[1].

Metoda	LR	CART	MARS	Bagging	Boosting	RF
Směrodatná odchylka	3,68	4,85	3,55	4,21	3,68	4,08

Tabulka 5.6: Standardní odchylky jednotlivých modelů

Největší směrodatnou odchylku má strom CART, což potvrzuje předpoklad, že se jedná spíše o nestabilní metodu. Druhá nejvyšší hodnota patří baggingu, což vysvětlujeme tak, že les vytvořený pomocí baggingu, se skládá z nestabilních stromů CART, jejichž nestabilita sice v průměru zaručuje lepší predikční schopnosti modelu, ale zároveň se přenáší i na celý les. Třetí nejvyšší směrodatnou odchylku má náhodný les.

Zbývající tři metody mají směrodatné odchylky na úrovni lineární regrese, takže je můžeme označit za stabilní techniky. V příloze A jsou shrnuté další popisné statistiky a boxploty pro jednotlivé modely.

Významnost proměnných

Na konec tohoto příkladu se podíváme na to, které proměnné byly vybrané jako nejdůležitější. V následující tabulce 5.7 jsou uvedeny prediktory a jejich významnost pro jednotlivé techniky.

Proměnná	LR	CART	MARS	Bagging	Boosting	RF
doy	2,9797	45,45 %	31,50 %	48,19 %	8,03 %	47,57 %
dpg	0,4287	33,34 %	29,09 %	51,73 %	8,42 %	41,20 %
humidity	3,9290	43,53 %	23,34 %	53,57 %	5,96 %	43,08 %
ibh	0,7229	81,39 %	20,97 %	81,08 %	18,54 %	68,83 %
ibt	1,9843	100,00 %	14,07 %	100,00 %	38,60 %	96,38 %
temp	5,0183	91,11 %	100,00 %	85,11 %	100,00 %	100,00 %
vh	0,9205	48,61 %	19,38 %	53,71 %	3,55 %	69,69 %
vis	1,9314	57,77 %	20,85 %	56,58 %	7,90 %	42,47 %
wind	0,4414	1,01 %	4,61 %	9,31 %	0,75 %	18,83 %

Tabulka 5.7: Proměnné dle významnosti

Jak se dalo čekat, důležitost jednotlivých proměnných se pro jednotlivé metody liší. Nicméně je zde několik proměnných, na jejichž významu se shodne více metod. Především je to proměnná, která udává naměřenou teplotu (*temp*). Tuto proměnnou označily jako nejvýznamnější čtyři ze šesti technik, zbývající dvě ji označily jako druhou nejvýznamnější. Druhou v pořadí dle významu je teplota

[1]LR - lineární regrese, RF - náhodný les

inverzní vrstvy (*ibt*). Kromě nejvýznamnější proměnný se všechny techniky zároveň shodly na té nejméně významné. Rychlost větru (*wind*) byla v pěti případech označená jako nejméně významná proměnná.

Ačkoliv každá metoda stanovuje významnost proměnné svým vlastním způsobem, všechny techniky dokážou shodně identifikovat dva "extrémy", ty nejvýznamnější i ty nejméně významné proměnné.

Odpověď na otázku, proč významnost jednotlivých proměnných se tak liší pro různé metody, dostaneme, když se podíváme na obrázky 5.9 a 5.10. Lineární regrese označila jako významné proměnné *temp*, *humidity* a *ibt*. Všechny tyto proměnné vykazují známky lineární závislosti, čili jde jimi poměrně snadno proložit přímku (viz Srovnání lineární regrese a stromu MARS). Stejná věc platí i pro strom MARS. Jako nejdůležitější proměnné jsou pro strom MARS ty, které dokáže aproximovat vhodnou lomenou přímkou, což jsou proměnné *temp*, *doy* nebo *dpg*.

Strom CART, bagging a náhodný les mají takřka shodné pořadí významnosti proměnných. Tento fakt není překvapivý, neboť jak bagging, tak i náhodný les mají základ ve stromě CART. I tyto metody označují jako významné ty proměnné, které dokážou dobře rozdělit pozorování do skupin s co nejodlišnějším průměrem.

V případě boostingu proměnná, která nejvíce snížila nepřesnost modelu, je *temp*, druhá v pořadí *ibt* snížila nepřesnost takřka třikrát méně. Tato skutečnost vyplývá z vlastností algoritmu 4 (podkapitola 4.2.1), a to konkrétně, že s rostoucím k, které představuje pořadí stromu v modelu boostingu, se bude odhadovat stále menší hodnota rozdílu mezi předpovědi a vysvětlovanou proměnnou y, takže absolutní velikosti poklesů nepřesnosti stromů se budou postupně zmenšovat. Proto proměnné, které se vyskytují na začátku, sníží nepřesnost mnohem více než ty, které se vyskytnou později. Což se stalo v tomto případě. Proměnná *temp* je významná, a proto byla na začátku použita několikrát k dělení, čímž největší poklesy nepřesnosti patří této proměnné.

Korelace proměnných

Na závěr se ještě podíváme na korelaci proměnných.

	O3	vh	wind	hum	temp	ibh	dpg	ibt	vis	doy
O3	1,00	-	-	-	-	-	-	-	-	-
vh	0,61	1,00	-	-	-	-	-	-	-	-
wind	0,00	-0,23	1,00	-	-	-	-	-	-	-
hum	0,45	0,07	0,22	1,00	-	-	-	-	-	-
temp	0,78	0,81	-0,01	0,34	1,00	-	-	-	-	-
ibh	-0,59	-0,50	0,20	-0,24	-0,53	1,00	-	-	-	-
dpg	0,21	-0,15	0,34	0,65	0,19	0,04	1,00	-	-	-
ibt	0,75	0,85	-0,16	0,20	0,86	-0,78	-0,10	1,00	-	-
vis	-0,44	-0,36	0,13	-0,40	-0,39	0,39	-0,13	-0,42	1,00	-
doy	0,07	0,34	-0,25	0,04	0,24	0,04	-0,15	0,22	-0,22	1,00

Tabulka 5.8: Personův korelační koeficient spočtený z celého vzorku 330 pozorování.

Vidíme vysokou korelaci mezi proměnnou *temp*, *ibt* a *vh*, tj. mezi teplotou, teplotou inverzní vrstvy a tlakem. Na základě této skutečnosti zkusíme vytvořit model bez použití proměnných *ibt* a *vh*, a ponecháme z uvedené trojice jenom proměnnou *temp*, neboť ta se jeví jako nejvýznamnější na základě výsledků z tabulky 5.7. Výsledky jsou v následující tabulce

Metoda	$\widehat{MSE}$
Lineární regrese	20,53
CART	23,86
MARS	15,82
Bagging	18,98
Boosting	16,63
Náhodný les	17,63

Tabulka 5.9: Hodnota $\widehat{MSE}$ pro různě velké lesy při použití na data Ozone při vypuštění proměnných *ibt* a *vh*.

Jak vidíme, výsledky zůstaly podobné jako v případě použití všech proměnných. Výsledky lineární regrese a stromu MARS jsou takřka stejné, ostatní metody se mírně zhoršily. Nejvíce se zhoršil náhodný les.

Pro toto zhoršení lze najít dva důvody. Ten první vyplývá z věty (4.2), která říká, že nepřesnost náhodného lesu je ovlivněná korelací mezi rezidui náhodných výběrů, ze kterých se vytváří jednotlivé stromy. Vyloučení dvou proměnných způsobilo, že výběry proměnných pro dělení uzlu začnou si být podobnější, a tím i rezidua modelů korelovanější. Se snížením přesnosti také souvisí to, jaké prediktory byly odstraněny. Jedná se o druhou a třetí nejvýznamnější proměnnou pro náhodný les. S jejích odstraněním se snížila pravděpodobnost, že pro dělení uzlu bude vybraná taková proměnná, která rozdělí pozorování na dvě co možná nejodlišnější skupiny.

5.2 Časová náročnost

Druhý příklad praktické části bude zaměřený na časovou náročnost jednotlivých technik. Budeme chtít změřit čas, který potřebují metody k vytvoření modelu s využitím dat z trénovacího vzorku. K měření času budeme používat proceduru *system.time* v programu *R*.

Časovou náročnost budeme měřit na dvou různých datových sadách. První datovou sadu označíme jako data Boston (viz [10]). Tato datová sada obsahuje celkem 506 pozorování a 16 měřených proměnných. Tato data byly shromážděné v roce 1978 ve městě Boston a jeho okolí za účelem stanovit průměrnou cenu domu na základě různých charakteristik. Také my zkusíme modelovat závislost ceny domu na ostatních proměnných. Některé z těchto charakteristik můžeme vidět v tabulce 5.10, seznam všech 16 měřených proměnných je pak v příloze B.

Název proměnné	Popis proměnné
crime	počet zločinů na jednoho obyvatele
ptratio	počet žáků na učitele
lstat	procento obyvatelů z nižší sociální skupiny
highway	vzdálenost od dálnice
value	cena domů

Tabulka 5.10: Vybrané proměnné datové sady Boston

Druhá sada dat pochází z roku 1994, z kanadské provincie Ontario (viz [17]). Tato sada má pouze 4 proměnné a 3989 pozorování. Proměnné vidíme v následující tabulce:

Název proměnné	Popis proměnné
age	věk
sex	pohlaví
compositeHourlyWages	hodinová sazba (v dolarech)
yearsEducation	doba vzdělávání (v letech)

Tabulka 5.11: Seznám proměnných dat Ontario

V tomto případě zkusíme modelovat vztah mezi hodinovou sazbou a ostatními proměnnými.

5.2.1 Výsledky a komentáře

Parametry regresních lesů byly vybrané následující. Pro data Boston se jedná o

- bagging - počet stromů $K = 30$
- boosting - počet stromů $K = 200$, velikost jednoho stromu $M' = 5$, parametr $\nu = 0,1$
- náhodný les - počet stromů $K = 50$, velikost jednoho stromu $M = 5$

a pro data Ontario

- bagging - počet stromů $K = 30$
- boosting - počet stromů $K = 50$, velikost jednoho stromu $M' = 1$, parametr $\nu = 0,1$
- náhodný les - počet stromů $K = 20$, velikost jednoho stromu $M = 1$

Parametry regresních lesů byly určeny stejným způsobem, jaký byl popsán v podkapitole 5.1.2.

Protože v tomhle případě není primárním cílem najít nejlepší model, ale zjistit časovou náročnost metod, provedeme rozdělení dat na trénovací a testovací vzorek pouze 500×. Velikost trénovacího vzorku v případě dat Boston je 400 pozorování, velikost testovácího vzorku je 106 pozorování (21 % z celkového počtu pozorování), v případě dat Ontario je 3000 pozorování v trenovácím a 989 v testovacím vzorku (25 % z celkového počtu pozorování).

Přesnost aproximace

Výsledky přesnosti aproximace jsou shrnuté v následující tabulce.

Metoda	Boston	Ontario
Lineární regrese	24,44	42,95
CART	23,44	41,63
MARS	15,40	40,42
Bagging CART	16,96	40,12
Boosting	12,42	38,77
Náhodný les	11,20	41,08

Tabulka 5.12: Hodnoty $\widehat{MSE}$ pro jednotlivé metody při použití na data Boston a Ontario

Pro data Boston se jako nejlepší modely jeví náhodný les, boosting a strom MARS. Obdobně je to v případě dat Ontario s tím rozdílem, že náhodný les byl v trojici nejlepších metod nahrazen baggingem.

Lineární regrese měla v obou dvou případech nejvyšší průměrnou čtvercovou chybu. Avšak je pravděpodobné, že úpravou lineárního modelu (transformací některých proměnných, přidání interakcí apod.) bychom obdrželi přesnější model.

Časová náročnost

Při každém opakování byly časy pro výstavbu jednotlivých modelů ukládané, a poté zprůměrované. V tabulce 5.13 vidíme výsledky tohoto testování. Všechny testy probíhaly na notebooku Futjisu Siemens Amilo 1510 (procesor AMD Mobile Sempron 3200+). Je pravděpodobné, že na jiném počítači by tyto časy byly odlišné. Nicméně tyto hodnoty nám dávají odhad, jak se od sebe jednotlivé metody liší.

Metoda	Boston	Ontario
Lineární regrese	0,251	0,022
CART	0,052	0,234
MARS	0,059	0,077
Bagging	0,273	0,917
Boosting	0,389	0,151
Náhodný les	0,224	0,225

Tabulka 5.13: Časy jednotlivých metod v sekundách

V případě datové sady Ontario, tj. hodně pozorování a málo vysvětlujících proměnných, je nejrychlejší lineární regrese, následovaná stromem MARS a boostingem. Díky malému počtu vysvětlujících proměnných je nejrychlejší lineární regrese, protože v tomto případě nebylo zapotřebí použít stepwise proceduru, která by značně prodloužila dobu potřebnou pro výstavbu modelu. Pokud nebudeme používat stepwise proceduru ani na datech Boston, výstavba lineárního modelu se zrychlí na 0,014 sekund, a zároveň $\widehat{MSE}$ o něco klesne, na 23,76.

Strom CART je viditelně pomalejší na datech Ontario než na datech Boston. Tuto skutečnost si vysvětlujeme jednoduše velikostí trénovacího vzorku. Čím větší je trénovací vzorek, tím větší prvotní strom CART sestrojí, a tím déle bude také trvat jeho následné prořezávání. Počet vysvětlujících proměnných zde nehraje až takovou roli. Důležitá jsou nastavení zastavovacích pravidel. Strom se dělí, dokud nezačne platit nějaké zastavovací pravidlo, například minimální počet pozorování v uzlu.

V případě stromu CART Pro obě sady dat platilo, že uzel se smí dále dělit, pokud obsahuje minimálně 20 pozorování, a počet pozorování v jeho následnících bude aspoň třetina z minimálního počtu dostupného pro dělení, tj. zaokrouhleně 6 pozorování. Změnou zastavovacích pravidel by se čas pro výstavbu modelu CART změnil. Pokud v případě dat Ontario nastavíme minimální počet umožňující dělení na 100, čas potřebný pro stavbu jednoho stromu se sníží na 0,133, přičemž průměrná čtvercová chyba zůstane stejná, $\widehat{MSE} = 41,55$.

Metoda MARS má pro obě dvě sady dat takřka stejné časy. Zde hrají důležitou roli vlastnosti algoritmu MARS, konkrétně že základní funkce algoritmu (3.16) smí obsahovat jenom různé vysvětlující proměnné. Takže v případě dat Ontario základní funkce se skládá nejvýše z tří členů, kdežto v případě datové sady Boston může těch členů mít až 16. Tato skutečnost kompenzuje to, že v případě dat Ontario algoritmus musí prohledat více pozorování, aby našel ideální bod pro dělení.

Pro bagging platí stejné věci jako pro jednotlivý strom CART, z většího počtu pozorování v trénovacím vzorku se při stejných zastavovacích pravidlech vytvoří větší strom. Vytvořit a prořezat větší strom trvá jednoduše déle, než vytvořit a prořezat menší strom. I zde rychlost metody lze výrazně ovlivnit nastavením zastavovacích pravidel.

V případě boostingu a náhodného lesu se časy porovnávají obtížně, protože pro každou datovou sadu byl použit jiný výsledný model. Lepší možnost srovnání nabízí příloha C, kde jsou shrnuté kompletní výsledky pro všechny testované

metody. Obecně se dá říct, že čím větší les a čím větší je samotný strom v lese, tím déle trvá sestrojit model.

Na základě výsledků z 5.13 můžeme říct, že pro data typu Boston, tj. středně velká data s větším počtem vysvětlujících proměnných, jsou nejlepší metody náhodný les a strom MARS. Obě dvě metody jsou rychlé a dosáhly nízké průměrné čtvercové chyby. Boosting také dosáhl nízké $\widehat{MSE}$, ale je takřka 7,5× pomalejší než strom MARS.

Při použití na data typu Ontario, tj. málo vysvětlujících proměnných a vysoký počet pozorování, je lineární regrese jednoznačně nejrychlejší. Ale na druhou stranu její průměrná čtvercová chyba je také nejvyšší. A tak za nejlepší model by se dal označit boosting, který dokázal nejlépe predikovat při nízkém čase.

6. Závěr

Na závěr stručně shrneme výhody a nevýhody jednotlivých metod, se kterými jsme se setkali v této práci. V této práci jsme představili celkem dva regresní stromy, CART a MARS, a tři regresní lesy, bagging, boosting a náhodný les. Dlužno ovšem dodat, že metody bagging a boosting lze považovat za stromové metody jenom zčásti, neboť jsou použitelné i na jiné typy modelů.

Strom CART je rychlý algoritmus, jeho schopnost aproximovat data je srovnatelná, dokonce i o něco lepší než v případě lineární regrese. Nevýhodou je jeho nestálost (viz příloha A), při změně dat se výsledný model může jak zlepšit, tak i zhoršit. Další výhodou stromu CART je jeho snadný výklad a možnost přehledného grafického zobrazení.

Strom MARS je stejně rychlý jako CART, a navíc ve všech testovacích případech dokázal lépe popsat vztah mezi vysvětlovanou a vysvětlující proměnnou. Tento strom ale na rozdíl od stromu CART nemá tak přehlednou strukturu, a proto je hůř interpretovatelný.

Bagging výrazně zlepšuje predikční schopnosti stromu CART a je jednoduchý na pochopení. Nicméně v porovnání s ostatními komisními metodami je pomalejší a navíc nedokáže předpovídat s takovou přesností.

Boosting je rychlý a přesný algoritmus, jeho nevýhodou ale může být složitější nastavení parametrů, které se liší pro každou datovou sadu. Za další nevýhodu tohoto algoritmu lze považovat zkreslování významnosti jednotlivých proměnných (viz podkapitola 5.1.2 - Významnost proměnných), kdy přidává na významnosti důležitějším proměnným, kdežto ostatní se pak jeví jako méně podstatné.

Stejné věci jako pro boosting, platí i pro náhodný les. Co se týče schopnosti predikovat, náhodný les je pravděpodobně nejlepší z metod, které zde byly vyzkoušeny.

Obecně se dá říct, že stromy, respektive lesy, se vyplatí používat v případě, kdy máme více vysvětlujících proměnných a chceme především predikovat. I pomocí lineární regrese můžeme dosáhnout výsledků jen o něco málo horších, než jsou výsledky regresních stromů a lesů, ale v takovém případě musíme často použít nejrůznější transformace a interakce mezi proměnnými. Navíc, lineární regrese patří mezi parametrické metody, takže naše data by navíc měly splňovat určité předpoklady (viz [18]).

V případě regresních stromů a lesů nemusíme řešit předpoklady, ani provádět úpravy proměnných, což dělá práci s těmito metodami snadnější a rychlejší, navíc s lepšími výsledky než při použití lineární regrese. Je ale nutné znát základní principy toho, jak jednotlivé stromy a lesy pracují, v opačném případě se z nich stane nepoužitelná černá skříňka.

Literatura

[1] Anděl, J.: *Základy matematické statistiky*, MATFYZPRESS, Praha, 2007

[2] Breiman, L.: *Bagging Predictors*, Machine Learing, Boston, 1996

[3] Breiman, L.: *Out-of-bag Estimation*, Technical report, Dept. Statistics, University California, Berkeley, CA. 1997

[4] Breiman, L.: *Random Forest*, Technical report, Dept. Statistics, University California, Berkeley, CA. 2001

[5] Breiman, L., Friedman, J.L.: *Estimating optimal transformations for multiple regression and correlation*, JASA, 80, pp. 580-598, 1985

[6] Breiman, L., Friedman, J. H., Olshen R. H., Stone C. J.: *Classification and Regression Trees*, Wadsworth, 1984

[7] Friedman, J. H.: *Greedy function approximation: the gradient boosting machine*. Technical report, Stanford Univ., 1999

[8] Friedman, J. H.: *Multivariate Adaptive Regression Splines*. Annals of Statistics 19/1, 1U141, 1991

[9] Geospatial Media and Communications Pvt. Ltd.: *Subpixel Estimation of Impervious Surface Using Regression Tree Model: Accuracy of The Estimation at Different Spatial Scales*, dostupné na *http://gisdevelopment.net/technology/rs/ma06_110a.htm*

[10] Harrison, D. Rubinfeld, D.L.: *Hedonic prices and the demand for clean air*. J. Environ. Economics & Management, 5: 81-102, 1978, dostupné na *http://biostat.mc.vanderbilt.edu*

[11] Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*, Springer, 2001

[12] Hothorn, T., Hornik, T., Strobl, C., Zeileis, A.: *Package party - A Laboratory for Recursive Partytioning*, verze 0.9-99996, 20.12.2011

[13] Kuhn, Max a spol.: *Package caret - Classification and Regression Training*, verze 5.15-023, 20.03.2012, dostupné na *http://caret.r-forge.r-project.org*

[14] Liaw, A., Wiener, M.: *Package randomForest - Breiman and Cutler's random forests for classification and regression*, verze 4.6-6, 06.01.2012, dostupné na *http://cran.r-project.org/*

[15] Milborrow, S.: *Package earth - Multivariate Adaptive Regression Spline Models*, verze 3.2-2, 14.03.2012, dostupné na *http://cran.r-project.org/*

[16] Ridgeway, G.: *Package gbm - Generalized Boosted Regression Models*, verze 1.6-3.1, 07.05.2010, dostupné na *http://cran.r-project.org/*

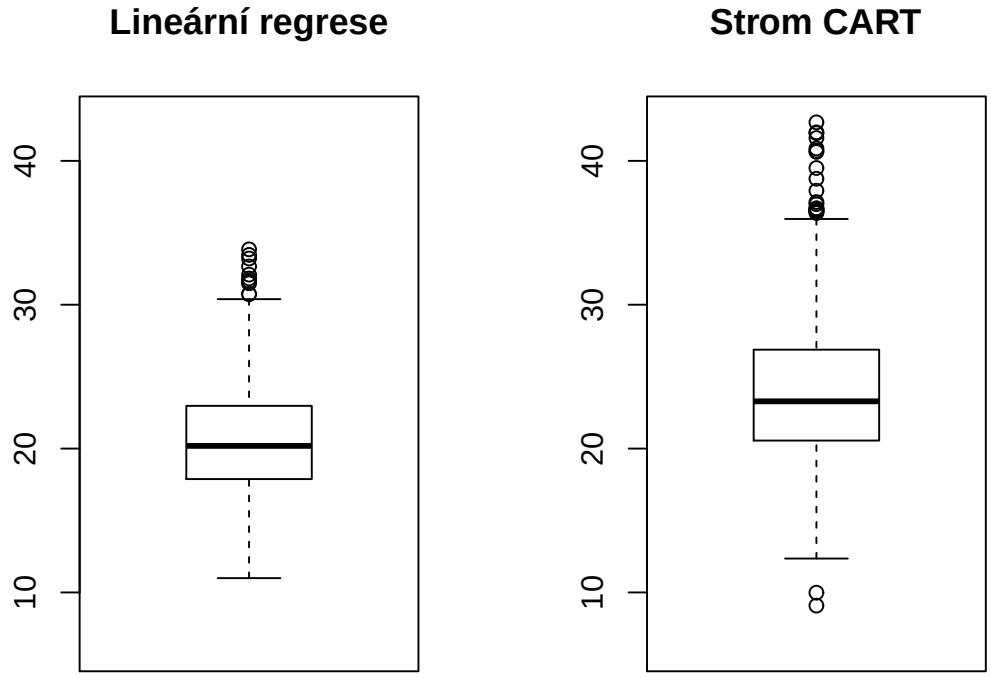
[17] Statistics Canada: *Wave of the Survey of Labour and Income Dynamics*, York University Institute for Social Research, Ontario, 1994. Dostupné na *http://www.socialsciences.mcmaster.ca*

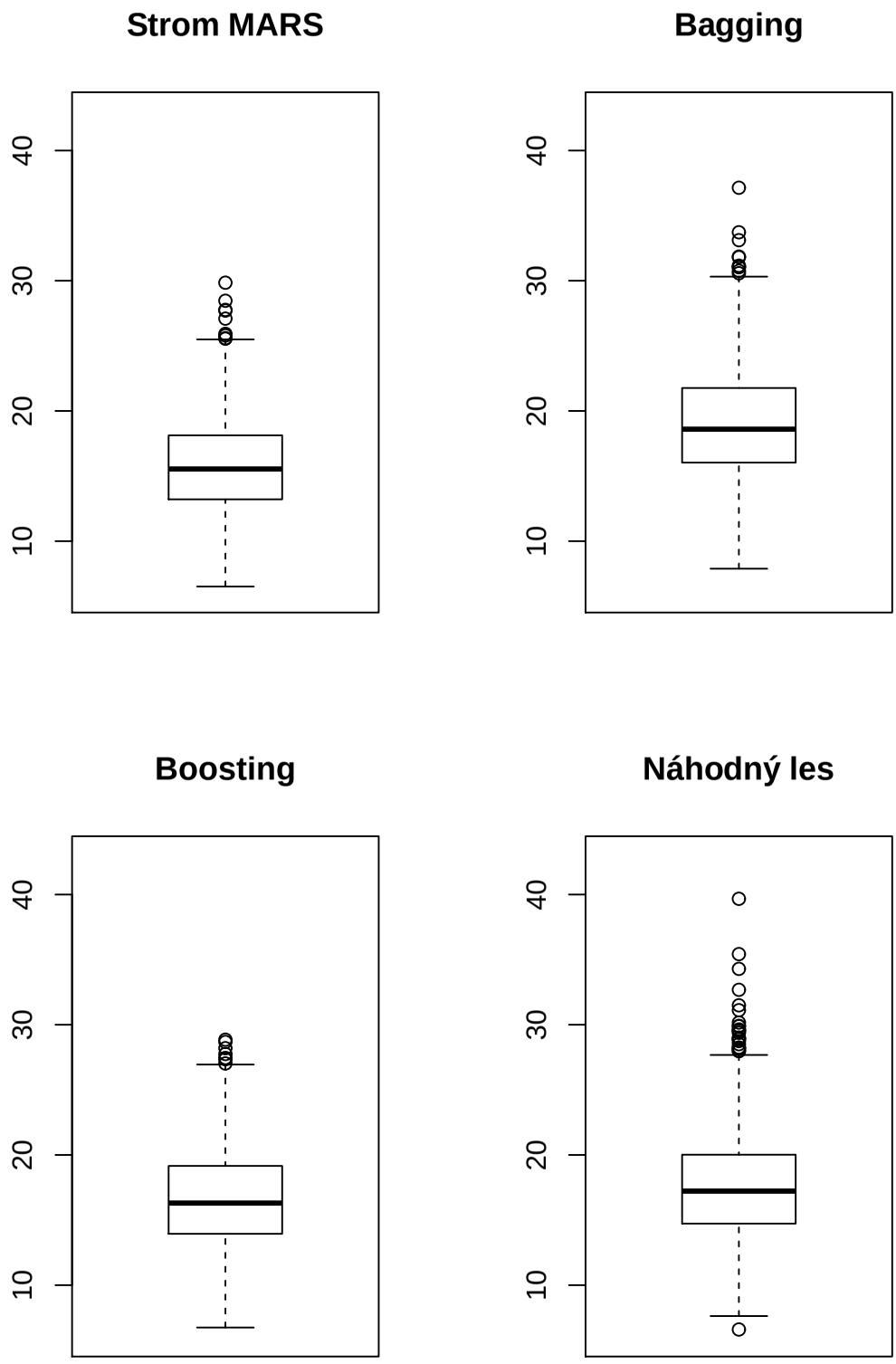
[18] Zvára, K.: *Regrese*, MATFYZPRESS, Praha, 2008

Příloha A: Popisné statistiky dosažených $\widehat{MSE}$ pro jednotlivé metody

Metoda	Min.	1.kv.	Medián	Průměr	3. kv.	Max.	St. odchylka
LR	10,99	17,88	20,19	20,53	22,97	33,85	3,68
CART	9,08	20,56	23,28	23,86	26,87	42,68	4,85
MARS	6,51	13,21	15,54	15,82	18,12	29,85	3,55
Bagging	7,89	16,03	18,60	18,98	21,76	37,14	4,21
Boosting	6,75	13,94	16,30	16,63	19,16	28,85	3,68
RF	6,59	14,73	17,22	17,63	20,01	39,67	4,08

Tabulka 6.1: Souhrnné informace o průměrných čtvercových chybách, které byly vypočtený jednotlivými technikami. Tabulka obsahuje vypočtené minimální a maximální dosaženou hodnotu $\widehat{MSE}$, dále 1. a 3. kvartil, medián, průměr a standardní odchylku.




Strom MARS
Bagging
Boosting
Náhodný les
10
20
30
40

Příloha B: Seznam proměnných datové sady Boston

Název proměnné	Popis proměnné
longitude	zeměpisná délka
latitude	zeměpisná šířka
value	hodnota domu (v 1000$)
crime	počet zločinů na jednoho obyvatele
residential	procento plochy města určené pro zástavbu
industrial	procento plochy města určené pro podnikání
river	přítomnost řeky v blizkosti domu
nox	koncentrace oxidu dusičného
rooms*	počet pokojů na obyvatele domu
older	procento domů postavených před rokem 1940 v dané lokalitě
distance	vzdálenost do byznys center v Bostonu
highway*	dostupnost dálnic
tax†	míra zdanění nemovitého majetku v dané lokalitě
ptratio†	počet žáků na učitele v dané lokalitě
lstat*	procento obyvatel z nižší sociální třídy v dané lokalitě
black†	procento obyvatel černé rasy v dané lokalitě

Tabulka 6.2: Seznám proměnných dat Boston

* proměnné, na jejichž významnosti shodlo více metod

† proměnné, na jejichž nevýznamnosti shodlo více metod

Příloha C: Výpočetní časy jednotlivých lesů

Počet stromů	Boston	Ontario
5	0,060	0,127
10	0,101	0,227
20	0,185	0,432
30	0,271	0,630
50	0,440	1,040
70	0,605	1,446
100	0,859	2,053
150	1,288	3,087
200	1,707	4,091
300	2,565	6,140
500	4,272	10,222

Tabulka 6.3: Čas potřebný pro výstavbu různě velkých lesů - bagging

ν	M'	K							
		10	20	50	100	200	500	1000	2000
1	1	0,083	0,091	0,120	0,166	0,262	0,545	1,030	1,985
	2	0,087	0,107	0,159	0,245	0,424	0,960	1,843	3,611
	5	0,108	0,142	0,254	0,432	0,791	1,857	3,658	7,223
	9	0,126	0,182	0,351	0,625	1,177	2,820	5,583	11,128
0,1	1	0,082	0,090	0,119	0,170	0,266	0,551	1,038	1,994
	2	0,089	0,105	0,162	0,253	0,426	0,964	1,856	3,638
	5	0,106	0,140	0,253	0,435	0,806	1,888	3,706	7,323
	9	0,125	0,180	0,355	0,644	1,238	2,991	5,970	12,039
0,01	1	0,082	0,091	0,118	0,169	0,263	0,552	1,028	1,997
	2	0,089	0,107	0,160	0,250	0,421	0,957	1,844	3,613
	5	0,107	0,139	0,245	0,421	0,773	1,887	3,745	7,398
	9	0,124	0,180	0,338	0,600	1,148	2,880	5,838	11,749

Tabulka 6.4: Čas potřebný pro výstavbu různě velkých lesů - boosting (data Boston)

ν	M'	K							
		10	20	50	100	200	500	1000	2000
1	1	0,033	0,043	0,072	0,125	0,226	0,531	1,043	2,065
	2	0,036	0,055	0,099	0,178	0,331	0,798	1,573	3,129
	3	0,042	0,062	0,122	0,218	0,414	1,006	1,998	3,975
0,1	1	0,032	0,044	0,076	0,126	0,229	0,536	1,045	2,062
	2	0,038	0,054	0,103	0,180	0,335	0,796	1,567	3,116
	3	0,044	0,064	0,127	0,222	0,414	0,994	1,970	3,930
0,01	1	0,033	0,042	0,073	0,126	0,226	0,546	1,065	2,092
	2	0,038	0,055	0,103	0,185	0,342	0,825	1,608	3,141
	3	0,043	0,066	0,128	0,232	0,441	1,046	2,014	3,952

Tabulka 6.5: Čas potřebný pro výstavbu různě velkých lesů - boosting (data Ontario)

Počet stromů	1	5	9	15
10	0,028	0,053	0,076	0,111
20	0,041	0,095	0,140	0,208
50	0,082	0,211	0,324	0,496
100	0,146	0,406	0,634	0,979
200	0,279	0,795	1,252	1,937
500	0,675	1,968	3,107	4,822
1000	1,378	3,960	6,251	9,663
2000	3,359	8,507	13,182	19,949

Tabulka 6.6: Čas potřebný pro výstavbu různě velkých lesů s různým počtem náhodně volených proměnných (1, 5, 9, 15) - náhodný les (data Booston)

Počet stromů	1	2	3
10	0,133	0,071	0,128
20	0,225	0,142	0,261
50	0,546	0,336	0,627
100	1,094	0,670	1,280
200	2,152	1,358	2,544
500	6,352	7,488	13,811
1000	12,856	14,972	27,606

Tabulka 6.7: Čas potřebný pro výstavbu různě velkých lesů s různým počtem náhodně volených proměnných (1, 2, 3) - náhodný les (data Ontario). Čas potřebný pro vytvoření lesu o velikosti 2000 stromů se bohužel nepodařilo určit - hardwarové nároky na paměť při vytvoření tak velkého lesu byly větší než možnosti testovacího notebooku.