Zacílení marketingové kampaně

Daniel Knespl

2025-04-16

# Knihovny

Knihovny využívané při analýze, modelování, evaluaci a nasazení.

## Instalace

| install.packages("DT") install.packages("dplyr") install.packages("ggplot2") install.packages("plotly") install.packages("C50") install.packages("stringi") install.packages("nnet") install.packages("NeuralNetTools") install.packages("DBI") install.packages("odbc") install.packages("ROCR") install.packages("png") install.packages("grid") |
| --- |



## Volání

| library(DT) *# Interaktivní HTML datové matice* library(dplyr) *# Manipulace s datovou maticí* library(ggplot2) *# Statické grafy* library(plotly) *# Interaktivní grafy, generované ze statických* library(C50) *# C5 modelování* library(stringi) *# Zpracování textových řetězců (problémy s diakritikou)* library(caret) *# Analýza výsledků* library(nnet) *# Neuronové sítě* library(NeuralNetTools) *# Vizualizace neuronových sítí* library(DBI) *# Rozhraní pro komunikaci s databázemi* library(odbc) *# Připojení k databázi skrze ODBC* library(ROCR) *# Gains charty* library(png) *# Načítání png* library(grid) *# Vykreslování png*  invisible(Sys.setlocale("LC\_ALL", "C.UTF-8")) |
| --- |

# Definice funkcí

| *# Vykreslování interaktivních bar plotů* d\_plot <- function(matrix\_data, x\_param, p\_position, p\_title, x\_desc, y\_desc) {  *# Z matice matrix\_data vyber hodnoty atributu x\_param*  *# Způsob zobrazení zavisí na zvoleném p\_position*  *# p\_title je řetězec s titulkem grafu*  *# x\_desc a y\_desc tak jsou popisky x a y osy*  p<-ggplot(matrix\_data, aes(x = .data[[x\_param]], fill = pep)) +  geom\_bar(binwidth = 1, position = p\_position, alpha = 0.7, color="black",size=0.5) +  scale\_fill\_manual(values = c("ano" = "blue", "ne" = "red")) +   labs(title = p\_title,  x = x\_desc,  y = y\_desc) +  theme\_replace() ggplotly(p) } *# Záměna hodnot ano/ne pro hodnotu pep v dané matici* swap\_pep <- function(matrix){  matrix <- matrix %>% mutate(pep = ifelse(pep == "ano", "ne", "ano"))  return(matrix) } *# Normalizace pro neuronovou síť* normalize <- function(x) {  return ((x - min(x)) / (max(x) - min(x))) } |
| --- |

# Business understanding

* Kampaně mají obecně nízkou návratnost
* Cíl: najít množinu zákazníků, u které bude dosažena vyšší míra zájmu o produkt než pro celou populaci =>snížení nákladů na kampaň, zvýšení zisku firmy
* Zadání
  + Banka zavádí nový produkt PEP(Personal equity plan-osobní finanční plán) a hodlá provést marketingovou kampaň
  + Nejprve zaslala nabídku 600 náhodných klientů a získala údaje o jejich zájmu
  + Chceme vytvořit model pro předpověď, komu zasílat nabídku produktu
* Modelování
  + Klasifikační strom C5.0
  + Neuronová síť
  + Logistická regrese
* Evaluace
  + Gains graf
  + Tabulka záměn
* Nasazení
  + import a export dat do databáze

# Data understanding

| **Proměnná** | **Popis** |
| --- | --- |
| id | Identifikační číslo zákazníka |
| vek | Věk zákazníka |
| pohlavi | Pohlaví zákazníka |
| region | Typ regionu zákazníka |
| prijem | Příjem zákazníka za poslední rok |
| stav | Rodinný stav zákazníka |
| deti | Počet dětí zákazníka |
| uver\_auto | Indikátor, zda má zákazník úvěr na automobil |
| ucet\_sporici | Indikátor, zda má zákazník sporici ucet |
| ucet\_bezny | Indikátor, zda má zákazník bezny ucet |
| hypoteka | Indikátor, zda má zákazník hypotéku |
| pep | Indikátor, zda má zákazník zájem o nabízený produkt PEP |

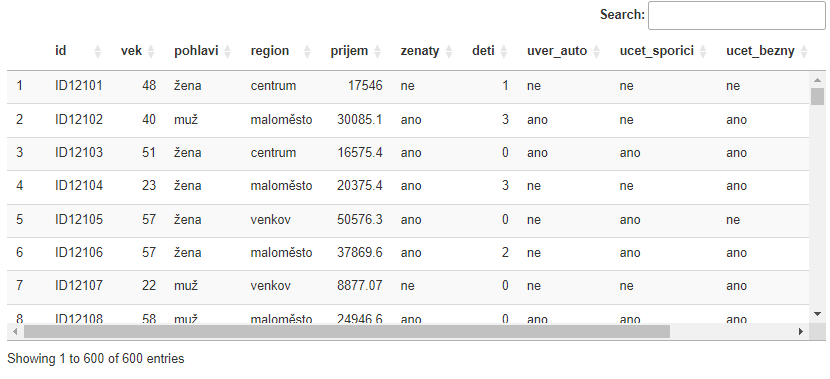
## Načítání dat

| *# načtení dat z lokálního úložiště* data\_matrix <- read.csv(  "D:/TUL/DM/semestralka/data\_model.txt",  header = TRUE, *# soubor obsahuje hlavičku*  sep = ",", *# oddělovač hodnot*  dec = ".", *# desetinný oddělovač*  fileEncoding = "windows-1250", *# kódování textu*  blank.lines.skip = TRUE *# přeskakování prázdných řádků* ) |
| --- |

Data jsou v souboru data\_model.txt uložena s hlavičkou, hodnoty jsou odděleny &#44; a pro desetinný oddělovač je &#46;. Data obsahují diakritiku a je nutné se jí zbavit. Při testování se mnohokrát stalo, že se soubor přestal načítat. Vždy to šlo opravit restartem RStudia.

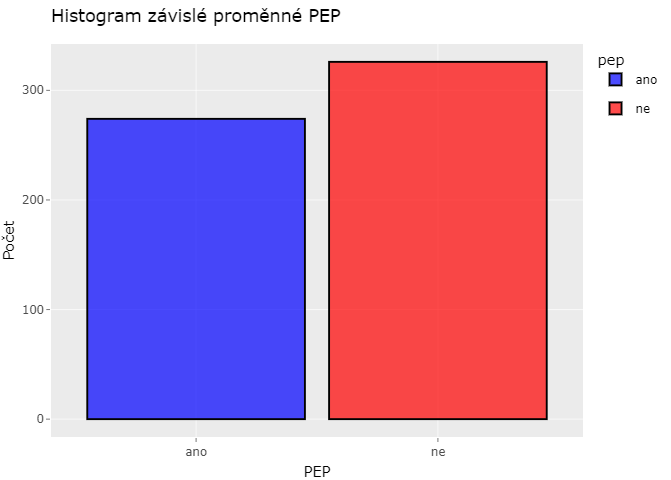
| *#Vykreslení interaktivní datové matice - tabulka šmírovačka* datatable(data\_matrix, options = list(scrollY = "300px", paging = FALSE,scrollX=TRUE,fixedColumns = list(leftColumns = 1))) |
| --- |

Pokud by jste četli interaktivní HTML verzi této dokumentace, měli by jste možnost scrollovat a hledat v datové matici. Zde je alespoň pohled na ní.



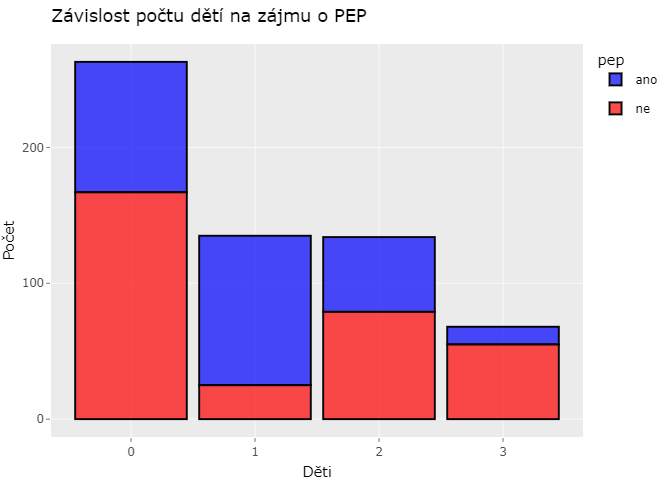
## Analýza dat

| d\_plot(data\_matrix,"pep","stack", "Histogram závislé proměnné PEP", "PEP", "Počet") |
| --- |

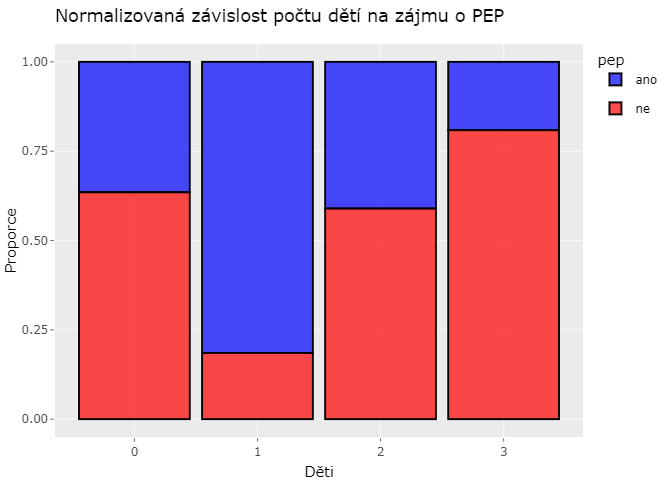


Poměr hodnot “ano” a hodnot “ne” je 45% : 55%, proměnná je tedy vyvážená.

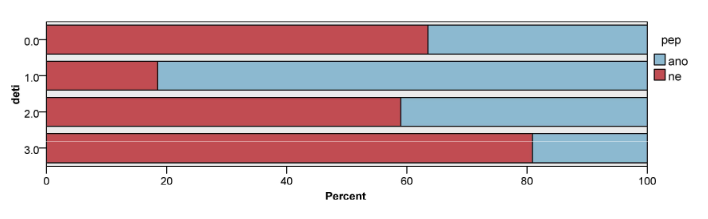
| d\_plot(data\_matrix,"deti","stack", "Závislost počtu dětí na zájmu o PEP", "Děti", "Počet") |
| --- |



| d\_plot(data\_matrix,"deti", "fill", "Normalizovaná závislost počtu dětí na zájmu o PEP", "Děti", "Proporce") |
| --- |



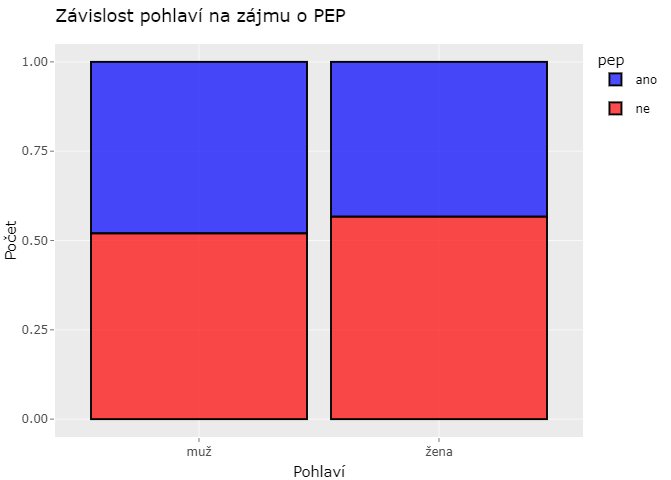
Předchozí graf zobrazuje proporcionální zastoupení pep hodnot pro záznamy s různým počtem dětí. Fascinujícím ale je, že modeler ukazuje naprosto totožné hodnoty. To by neměl být tak velký problém, ale v modeleru jsme zaměnili hodnoty “ano” a hodnoty “ne”, proto by dávalo smysl, že grafy nebudou shodné. Ale jak lze vidět na následujícím grafu, jsou totožné.



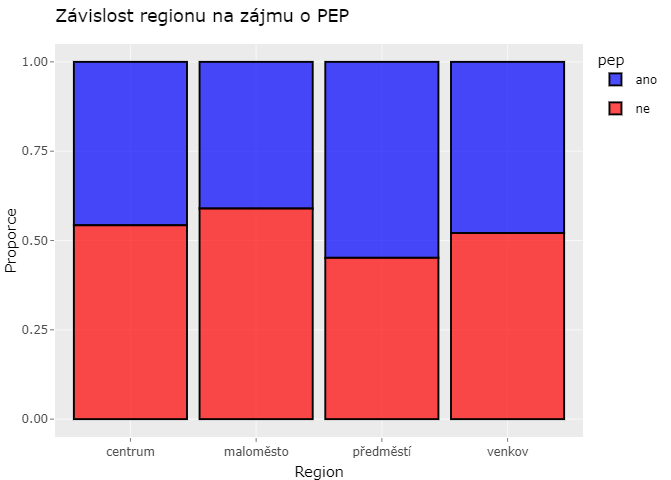
Toto zjištění mě zaskočilo a po chvíli experimentování s modelerem jsem zjistil, že výměna, kterou jsme udělali v modeleru nedělá vůbec nic. Modely i grafy jsou stejné. Proto abych korektně napodobil modeler udělal jsem funkci **swap\_pep** (definována dříve),která prohodí hodnoty “ano” a “ne”, ale v kódu jsem návratovou hodnotu nikde neuložil.

| *# Záměna hodnot ano/ne atributu pep* invisible(swap\_pep(data\_matrix)) |
| --- |

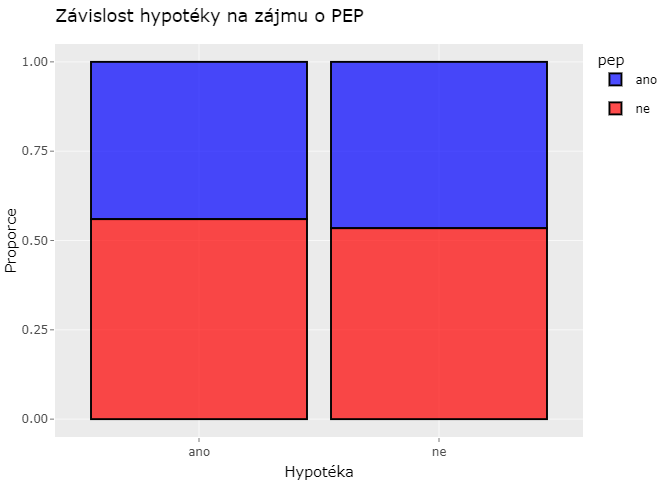
| d\_plot(data\_matrix,"pohlavi", "fill", "Závislost pohlaví na zájmu o PEP", "Pohlaví", "Počet") |
| --- |



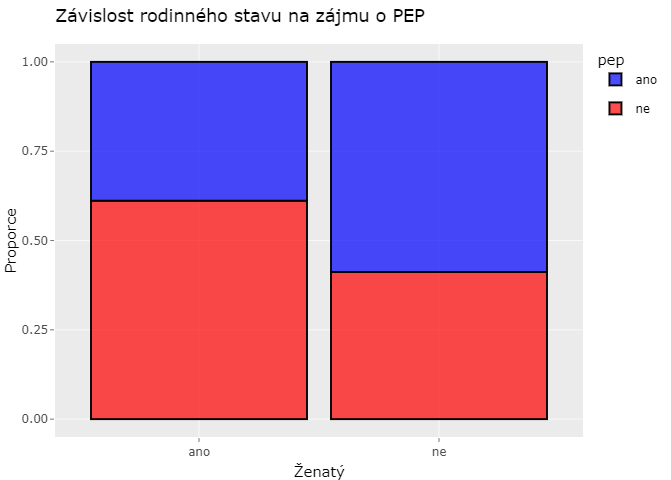
| d\_plot(data\_matrix,"region", "fill", "Závislost regionu na zájmu o PEP", "Region", "Proporce") |
| --- |



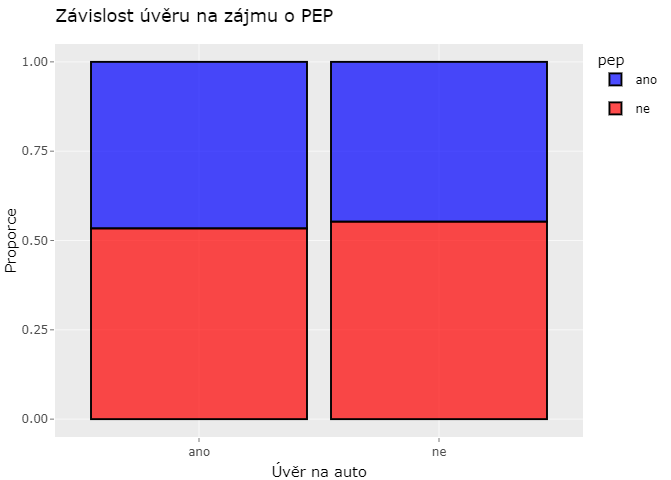
| d\_plot(data\_matrix,"hypoteka", "fill", "Závislost hypotéky na zájmu o  PEP", "Hypotéka", "Proporce") |
| --- |



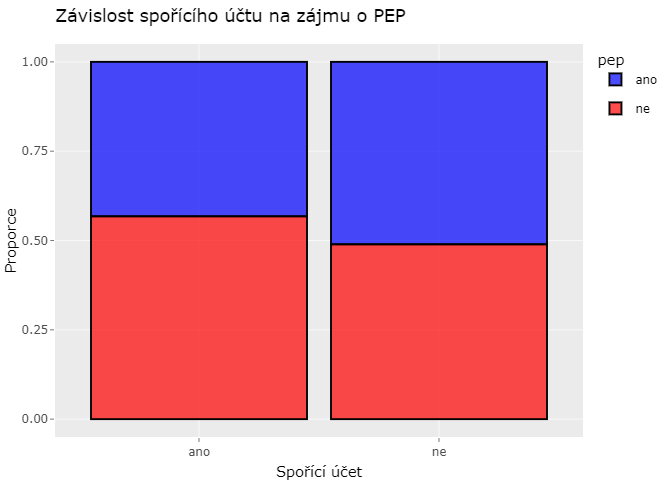
| d\_plot(data\_matrix,"zenaty", "fill", "Závislost rodinného stavu na zájmu o PEP", "Ženatý", "Proporce") |
| --- |



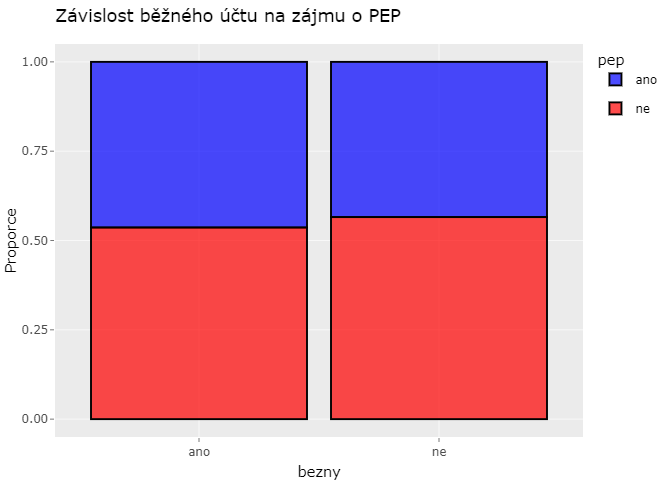
| d\_plot(data\_matrix,"uver\_auto", "fill", "Závislost úvěru na zájmu o PEP", "Úvěr na auto", "Proporce") |
| --- |



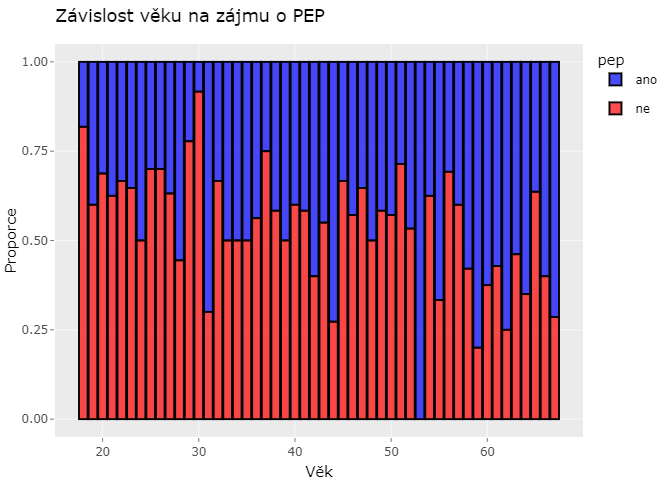
| d\_plot(data\_matrix,"ucet\_sporici", "fill", "Závislost spořícího účtu na zájmu o PEP", "Spořící účet", "Proporce") |
| --- |



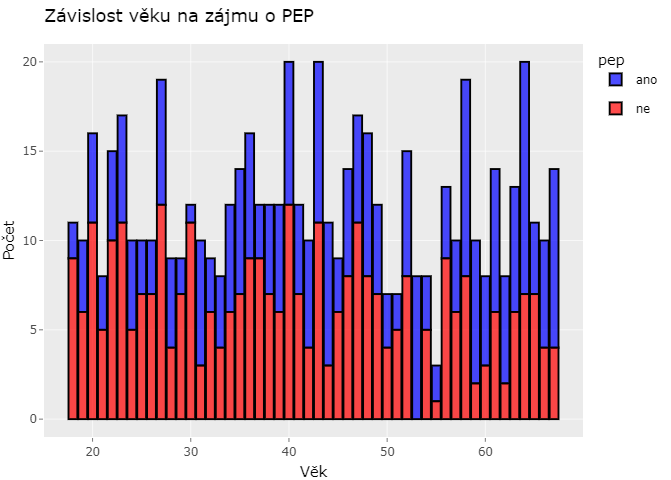
| d\_plot(data\_matrix,"ucet\_bezny", "fill", "Závislost běžného účtu na zájmu o PEP", "bezny", "Proporce") |
| --- |



| d\_plot(data\_matrix,"vek", "fill", "Závislost věku na zájmu o PEP", "Věk", "Proporce") |
| --- |



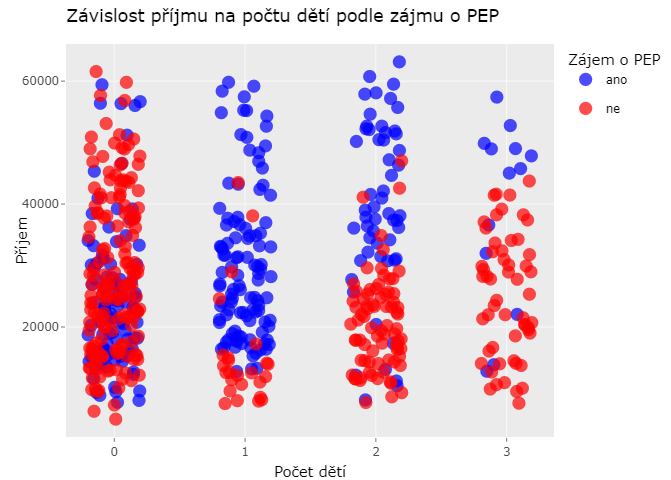
| d\_plot(data\_matrix,"vek", "stack", "Závislost věku na zájmu o PEP", "Věk", "Počet") |
| --- |



Z grafů se zdá, že PEP nemá na většině prediktorů silnou závislost. Výjimkami jsou zde prediktory **deti**, **zenaty**, **sporici\_ucet** a **region**. Prediktor **vek** se zdá mít jemně rostoucí trend pro **PEP** = “ano”.

| *# Vykreslení Jitter scatter plotu, popisující závislost mezi počtem dětí a příjmy* p<-ggplot(data\_matrix, aes(x = deti, y = prijem, color = pep)) +  geom\_jitter(alpha = 0.7, size = 3, width = 0.2, height = 0) +   scale\_color\_manual(values = c("ano" = "blue", "ne" = "red")) +   labs(title = "Závislost příjmu na počtu dětí podle zájmu o PEP",  x = "Počet dětí", y = "Příjem", color = "Zájem o PEP") +  theme\_replace() ggplotly(p) |
| --- |

Na grafu vykresleném tímto kódem lze vidět, jak ovlivňuje příjem a počet dětí cílovou proměnou **pep**. Pokud by se tento graf překryl s grafem popisující proporcionální závislost mezi dětmi a pepem, došlo by k velmi velkému překryvu. Zejména pak při počtu dětí > 0.



# Data preparation

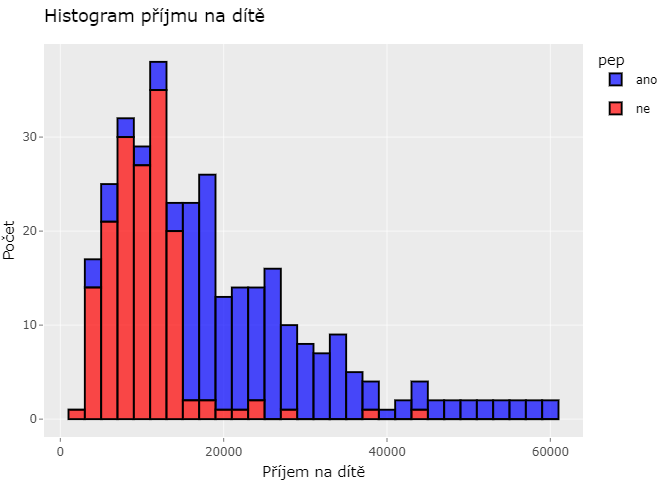
Pro testování jsem data rozdělil na trénovací a testovací, podle nové partition proměnné bude později rozdělena hlavní matice na trénovací a testovací matici.

| set.seed(123456) *# Seed pro rozdělení na trénovací a testovací data v poměru 50/50*  *# Rozdělení na trénovací a testovací data* data\_matrix$partition <- sample(c("train", "test"), nrow(data\_matrix), replace = TRUE)  *# Výpočet prediktoru prijem\_na\_dite* data\_matrix$prijem\_na\_dite <- ifelse(data\_matrix$deti == 0,  data\_matrix$prijem,  data\_matrix$prijem /  data\_matrix$deti) |
| --- |

Z analýzy vyšlo najevo, že příjem na dítě může být dobrým prediktorem, proto jsem zde takový prediktor vytvořil. Pro kontrolu je zde zobrazení histogramu hodnot pouze pro případy, kdy má záznam alespoň jedno dítě.

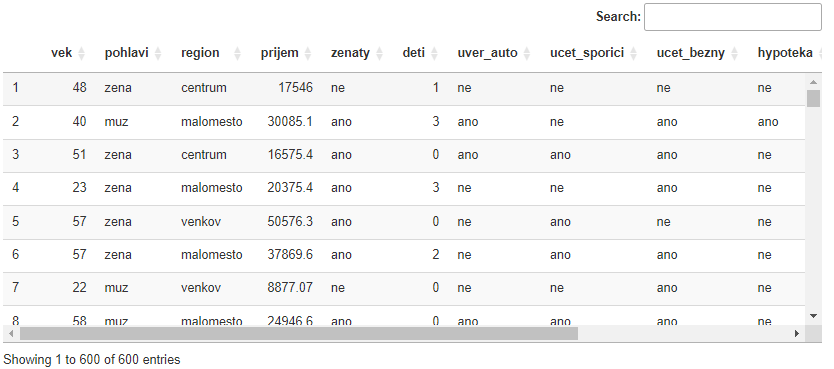
| *# Odfiltrování záznamů bez dětí* filtered\_data <- subset(data\_matrix, deti>0) *# Vykreslení histogramu příjmu na dítě* p <- ggplot(filtered\_data, aes(x = prijem\_na\_dite, fill = pep)) +  geom\_histogram(binwidth = 2000, color = "black", alpha = 0.7) +  scale\_fill\_manual(values = c("ano" = "blue", "ne" = "red")) +  labs(title = "Histogram příjmu na dítě",  x = "Příjem na dítě",  y = "Počet") +  theme\_replace() ggplotly(p) |
| --- |

Následující graf pěkně ukazuje že od příjmu na dítě +- 16000 je velice pravděpodobné, že záznam má zájem o PEP.



Trénované modely nefungují dobře s diakritikou, došlo tedy k očištění.

| *# Očištění dat od diakritiky a od parametru id (není prediktorem)* data\_matrix <- data\_matrix %>%  mutate(across(where(is.character), ~stri\_trans\_general(., "latin-ascii"))) %>%  select(-id) %>% *# Remove 'id' & derived columns*  mutate(across(where(is.character), as.factor))  *# Vykreslení datové matice, na které budou trénovány a testovány modely* datatable(data\_matrix, options = list(scrollY = "300px", paging = FALSE,scrollX=TRUE,fixedColumns = list(leftColumns = 1))) |
| --- |



Posledním krokem v přípravě dat je rozdělení této matice na testovací a trénovací, tentokrát již bez proměnné partition.

| *# Rozdělení hlavní matice na trénovací a testovací submatice* training\_matrix <- subset(data\_matrix, partition == "train") %>% select(-partition) test\_matrix <- subset(data\_matrix, partition == "test") %>% select(-partition) |
| --- |

# Modeling

V rámci modelování jsem vytvořil čtyři modely

* C5 Děti + Příjem
* C5 Děti + Příjem + Příjem na děti
* Logistická regrese
* Neuronová síť

## C5 Děti + Příjem

První model využívá rozhodovacího C5 stromu, bez využití prediktoru **prijem\_na\_dite**.

| *#Příprava matic pro C5* C5\_training\_matrix <- test\_matrix C5\_test\_matrix <- test\_matrix *# Definice prediktorů* predictors <- C5\_training\_matrix %>% select(-pep, -prijem\_na\_dite) test\_predictors <- C5\_test\_matrix %>% select(-pep,-prijem\_na\_dite) *# Trénování modelu* c5\_children\_model <- C5.0(x = predictors, y = C5\_training\_matrix$pep)  *# TESTOVÁNÍ* test\_predictions <- predict(c5\_children\_model, test\_predictors) *# Příprava na porovnání výsledků (nastavení shodných úrovní predikce a hledané proměnné)* C5\_test\_matrix$pep <- factor(C5\_test\_matrix$pep, levels = c("ano", "ne")) test\_predictions <- factor(test\_predictions, levels = c("ano", "ne"))  *# PŘÍPRAVA PRO EVALUACI* *# Výpočet matice záměn* c5c\_conf\_matrix <- confusionMatrix(test\_predictions, C5\_test\_matrix$pep) *# Hodnoty pro tvorbu gains* prob\_c5c <- predict(c5\_children\_model, C5\_test\_matrix, type = "prob")[,"ano"] |
| --- |

## C5 Děti + Příjem + Příjem na děti

Druhý model také využívá rozhodovacího C5 stromu, ale tentokrát využívá **prijem\_na\_dite**.

| *#Příprava matic pro C5* C5\_training\_matrix <- test\_matrix C5\_test\_matrix <- test\_matrix *# Definice prediktorů* predictors <- C5\_training\_matrix %>% select(-pep) test\_predictors <- C5\_test\_matrix %>% select(-pep) *# Trénování modelu* c5\_model <- C5.0(x = predictors, y = C5\_training\_matrix$pep)  *# TESTOVÁNÍ* test\_predictions <- predict(c5\_model, test\_predictors) *# Příprava na porovnání výsledků (nastavení shodných úrovní predikce a hledané proměnné)* C5\_test\_matrix$pep <- factor(C5\_test\_matrix$pep, levels = c("ano", "ne")) test\_predictions <- factor(test\_predictions, levels = c("ano", "ne"))  *# PŘÍPRAVA PRO EVALUACI* *# Výpočet matice záměn* c5m\_conf\_matrix <- confusionMatrix(test\_predictions, C5\_test\_matrix$pep) *# Hodnoty pro tvorbu gains*  prob\_c5m <- predict(c5\_model, C5\_test\_matrix, type = "prob")[, "ano"] |
| --- |

## Logistická regrese

Třetí model využívá logistické regrese pro předpověď, kdy je hodnota **pep** rovná “ne”.

| lr\_min\_confidence <- 0.49 *#Příprava matic pro logistickou regresi* lr\_training\_matrix <- training\_matrix lr\_test\_matrix <- test\_matrix *# Definice prediktorů* test\_predictors <- lr\_test\_matrix %>% select(-pep) *# Trénování modelu* logistic <-glm(pep ~ . ,family="binomial", data = lr\_training\_matrix) *#pep ~ . - predikuj pep na základě všech proměnných* *#family="binomial" - typ glm(zobecněný/obecný lineární model), kdy výsledek má být kategorizován do dvou úrovní*  *# TESTOVÁNÍ* *# "response" zde vrací procentuální konfidenci, že výsledek je "ne"* test\_probabilities <- predict(logistic, test\_predictors, type = "response") *# Určení hodnoty predikce* lr\_predictions <- ifelse(test\_probabilities > lr\_min\_confidence, "ne", "ano") *# Pokud je pravděpodobnost větší než 58%, je výsledkem predikce ne, jinak ano.* *# Příprava na porovnání výsledků (nastavení shodných úrovní predikce a hledané proměnné)* lr\_predictions <- factor(lr\_predictions, levels = c("ano", "ne")) lr\_test\_matrix$pep <- factor(lr\_test\_matrix$pep, levels = c("ano", "ne"))  *# PŘÍPRAVA PRO EVALUACI* *# Tvorba matice záměn* lr\_conf\_matrix <- confusionMatrix(lr\_test\_matrix$pep,lr\_predictions) *# Hodnoty pro tvorbu gains* prob\_lr <- predict(logistic, lr\_test\_matrix, type = "response") *# Hodnoty pro tvorbu histogramu* lr\_results <- lr\_test\_matrix lr\_results$predicted\_prob <- test\_probabilities lr\_results$predicted\_class <- lr\_predictions |
| --- |

## Neuronová síť

Čtvrtý a poslední model využívá neuronové sítě.

| *# Příprava matic pro neuronovou síť* n\_training\_matrix <- training\_matrix %>% select(-prijem) n\_test\_matrix <- test\_matrix %>% select(-prijem) *# Příprava proměnných pro neuronovou síť* factor\_vars <- c("pep","zenaty", "uver\_auto", "ucet\_sporici", "ucet\_bezny", "hypoteka", "pohlavi","region","deti") for (var in factor\_vars) {  n\_training\_matrix[[var]] <- as.numeric(n\_training\_matrix[[var]]) - 1  n\_test\_matrix[[var]] <- as.numeric(n\_test\_matrix[[var]]) -1 } *# Normalizace hodnot* n\_training\_matrix$vek <- normalize(n\_training\_matrix$vek) n\_training\_matrix$prijem\_na\_dite <- normalize(n\_training\_matrix$prijem\_na\_dite) n\_test\_matrix$vek <- normalize(n\_test\_matrix$vek) n\_test\_matrix$prijem\_na\_dite <- normalize(n\_test\_matrix$prijem\_na\_dite) *# Trénování modelu* neurn <- nnet(pep ~ ., n\_training\_matrix, size = 7, decay = 0.1, maxit = 500)  *# TESTOVÁNÍ* *# "raw" zde vrací procentuální konfidenci, že výsledek je "ano"* n\_predictions <- predict(neurn, n\_test\_matrix, type = "raw") *# Příprava na porovnání výsledků (nastavení shodných úrovní predikce a hledané proměnné)* n\_test\_matrix$pep <- ifelse(n\_test\_matrix$pep==1, "ano", "ne") n\_test\_matrix$pep <- factor(n\_test\_matrix$pep, levels = c("ano", "ne")) n\_predictions <- ifelse(n\_predictions > 0.5, "ano", "ne") n\_predictions <- factor(n\_predictions, levels = c("ano", "ne")) *# PŘÍPRAVA PRO EVALUACI* *# Tvorba matice záměn* n\_conf\_matrix <- confusionMatrix(n\_test\_matrix$pep,n\_predictions) *# Hodnoty pro tvorbu gains* prob\_n <- predict(neurn, n\_test\_matrix, type = "raw") |
| --- |

# Evaluation

## C5 Děti + Příjem

| print(c5c\_conf\_matrix) *# Matice záměn* |
| --- |

Confusion Matrix and Statistics

Reference

Prediction ano ne

ano 122 9

ne 16 154

Accuracy : 0.9169

95% CI : (0.8798, 0.9455)

No Information Rate : 0.5415

P-Value [Acc > NIR] : <2e-16

Kappa : 0.8321

Mcnemar's Test P-Value : 0.2301

Sensitivity : 0.8841

Specificity : 0.9448

Pos Pred Value : 0.9313

Neg Pred Value : 0.9059

Prevalence : 0.4585

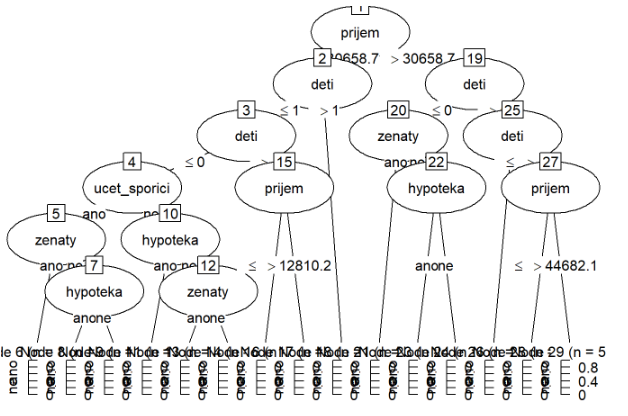
Detection Rate : 0.4053

Detection Prevalence : 0.4352

Balanced Accuracy : 0.9144

'Positive' Class : ano

| plot(c5\_children\_model) *# Dendrogram* |
| --- |



| varImp(c5\_children\_model) *# Významnost prediktorů* |
| --- |

Overall

prijem 100.00

deti 100.00

zenaty 34.88

ucet\_sporici 28.24

hypoteka 22.26

vek 0.00

pohlavi 0.00

region 0.00

uver\_auto 0.00

ucet\_bezny 0.00

První model dosahuje úspěšnosti 91.69%. Prediktory **prijem**, **deti**, **zenaty**, **ucet\_sporici** a **hypoteka** se ukázaly jako důležité. Na některé tyto prediktory jsem v části analýzy upozorňoval. Jiné se však ukázaly jako nedůležité - **vek**, **region**.

## 

## C5 Děti + Příjem + Příjem na dítě

| *print(c5m\_conf\_matrix) # Matice záměn* |
| --- |

Confusion Matrix and Statistics

Reference

Prediction ano ne

ano 128 9

ne 10 154

Accuracy : 0.9369

95% CI : (0.9032, 0.9616)

No Information Rate : 0.5415

P-Value [Acc > NIR] : <2e-16

Kappa : 0.8728

Mcnemar's Test P-Value : 1

Sensitivity : 0.9275

Specificity : 0.9448

Pos Pred Value : 0.9343

Neg Pred Value : 0.9390

Prevalence : 0.4585

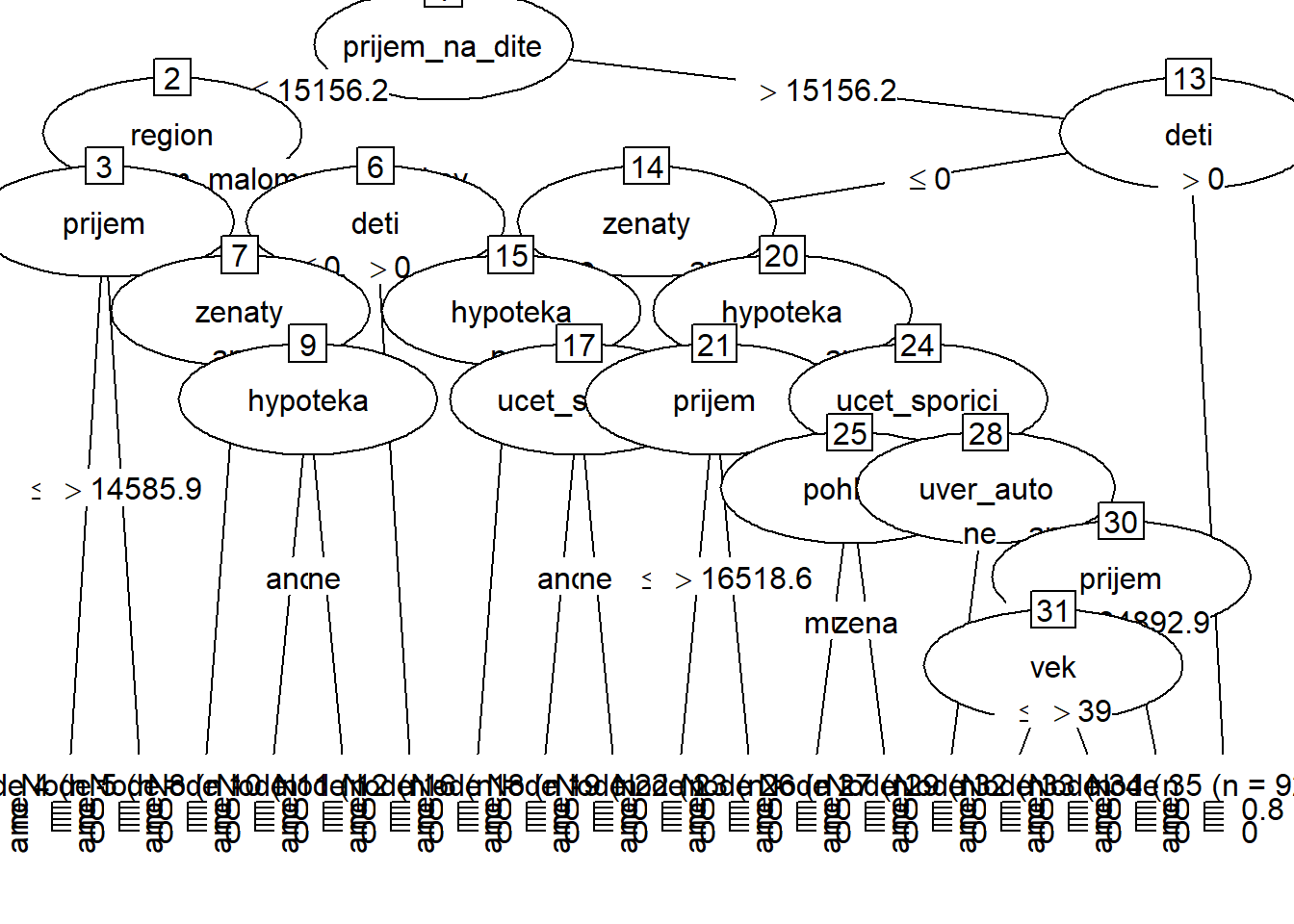
Detection Rate : 0.4252

Detection Prevalence : 0.4551

Balanced Accuracy : 0.9362

'Positive' Class : ano

| plot(c5\_model) *# Dendrogram* |
| --- |



| varImp(c5\_model) *# Významnost prediktorů* |
| --- |

Overall

prijem\_na\_dite 100.00

deti 96.68

zenaty 39.53

hypoteka 36.54

region 35.22

prijem 20.60

ucet\_sporici 14.62

uver\_auto 4.98

pohlavi 4.65

vek 1.99

ucet\_bezny 0.00

Druhý model dosahuje úspěšnosti 93.69%. Narozdíl od prvního modelu, je využito více prediktorů - například **region**, na který jsem upozorňoval v analýze, ale v prvním modelu nebyl vůbec využit.

Je důležité zmínit, že kvalita modelů je silně spojena s daty na kterých byly trénovány. Pro jiný seed náhodnosti rozdělení dat, vycházel tento model horší než původní. Dendrogram byl mnohem jednodušší, ale přesnost byla kolem 89%.

## Lineární regrese

| print(lr\_conf\_matrix) *# Matice záměn* |
| --- |

Confusion Matrix and Statistics

Reference

Prediction ano ne

ano 60 78

ne 39 124

Accuracy : 0.6113

95% CI : (0.5537, 0.6667)

No Information Rate : 0.6711

P-Value [Acc > NIR] : 0.9875915

Kappa : 0.1999

Mcnemar's Test P-Value : 0.0004429

Sensitivity : 0.6061

Specificity : 0.6139

Pos Pred Value : 0.4348

Neg Pred Value : 0.7607

Prevalence : 0.3289

Detection Rate : 0.1993

Detection Prevalence : 0.4585

Balanced Accuracy : 0.6100

'Positive' Class : ano

| summary(logistic) *# Nastavení modelu* |
| --- |

Call:

glm(formula = pep ~ ., family = "binomial", data = lr\_training\_matrix)

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.851e+00 5.863e-01 3.156 0.00160 \*\*

vek -1.467e-03 1.335e-02 -0.110 0.91247

pohlavizena 3.459e-01 2.560e-01 1.351 0.17655

regionmalomesto -5.921e-02 3.120e-01 -0.190 0.84950

regionpredmesti -1.681e-01 4.207e-01 -0.400 0.68941

regionvenkov -1.985e-01 3.706e-01 -0.536 0.59211

prijem 1.586e-05 2.918e-05 0.543 0.58686

zenatyne -1.002e+00 2.686e-01 -3.731 0.00019 \*\*\*

deti -8.055e-02 2.234e-01 -0.361 0.71844

uver\_autone 3.513e-01 2.605e-01 1.348 0.17755

ucet\_sporicine -7.598e-01 2.827e-01 -2.687 0.00720 \*\*

ucet\_beznyne -1.769e-01 2.931e-01 -0.604 0.54600

hypotekane -3.287e-01 2.696e-01 -1.220 0.22264

prijem\_na\_dite -6.097e-05 3.022e-05 -2.018 0.04363 \*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 412.06 on 298 degrees of freedom

Residual deviance: 368.08 on 285 degrees of freedom

AIC: 396.08

Number of Fisher Scoring iterations: 4

| varImp(logistic) *# Významnost prediktorů* |
| --- |

Overall

vek 0.1099220

pohlavizena 1.3514418

regionmalomesto 0.1897594

regionpredmesti 0.3996551

regionvenkov 0.5357833

prijem 0.5433862

zenatyne 3.7314588

deti 0.3605436

uver\_autone 1.3483494

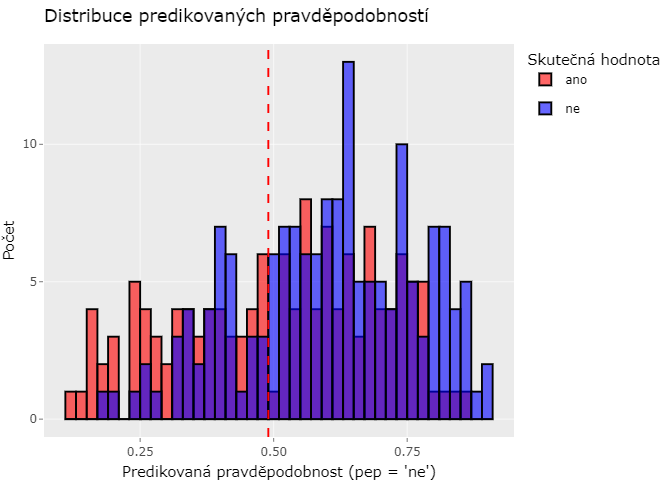
ucet\_sporicine 2.6874811

ucet\_beznyne 0.6037610

hypotekane 1.2195343

prijem\_na\_dite 2.0175889

| *#histogram pravděpodobností* ggplot(lr\_results, aes(x = predicted\_prob, fill = pep)) +  geom\_histogram(binwidth = 0.02, position = "identity", alpha = 0.6, color = "black") +  scale\_fill\_manual(values = c("ano" = "red", "ne" = "blue")) +  geom\_vline(xintercept = lr\_min\_confidence, linetype = "dashed", color = "red") +  labs(  title = "Distribuce predikovaných pravděpodobností",  x = "Predikovaná pravděpodobnost (pep = 'ne')",  y = "Počet",  fill = "Skutečná hodnota"  ) +  theme\_replace() |
| --- |



Logistická regrese dopadla ze všech modelů nejhůře - přesnost 61.13% (při minimální konfidenci 0.58) ne příliš lepší než náhodný výběr. Logistická regrese byla nejvíce volatilní s různými seedy. Na seedu “420301” s nastavením minimální konfidence na 0.53, dosahovala přesnosti 66.46%.

## Neuronová síť

| print(n\_conf\_matrix) *# Matice záměn* |
| --- |

Confusion Matrix and Statistics

Reference

Prediction ano ne

ano 144 19

ne 31 107

Accuracy : 0.8339

95% CI : (0.7869, 0.8741)

No Information Rate : 0.5814

P-Value [Acc > NIR] : <2e-16

Kappa : 0.6632

Mcnemar's Test P-Value : 0.1198

Sensitivity : 0.8229

Specificity : 0.8492

Pos Pred Value : 0.8834

Neg Pred Value : 0.7754

Prevalence : 0.5814

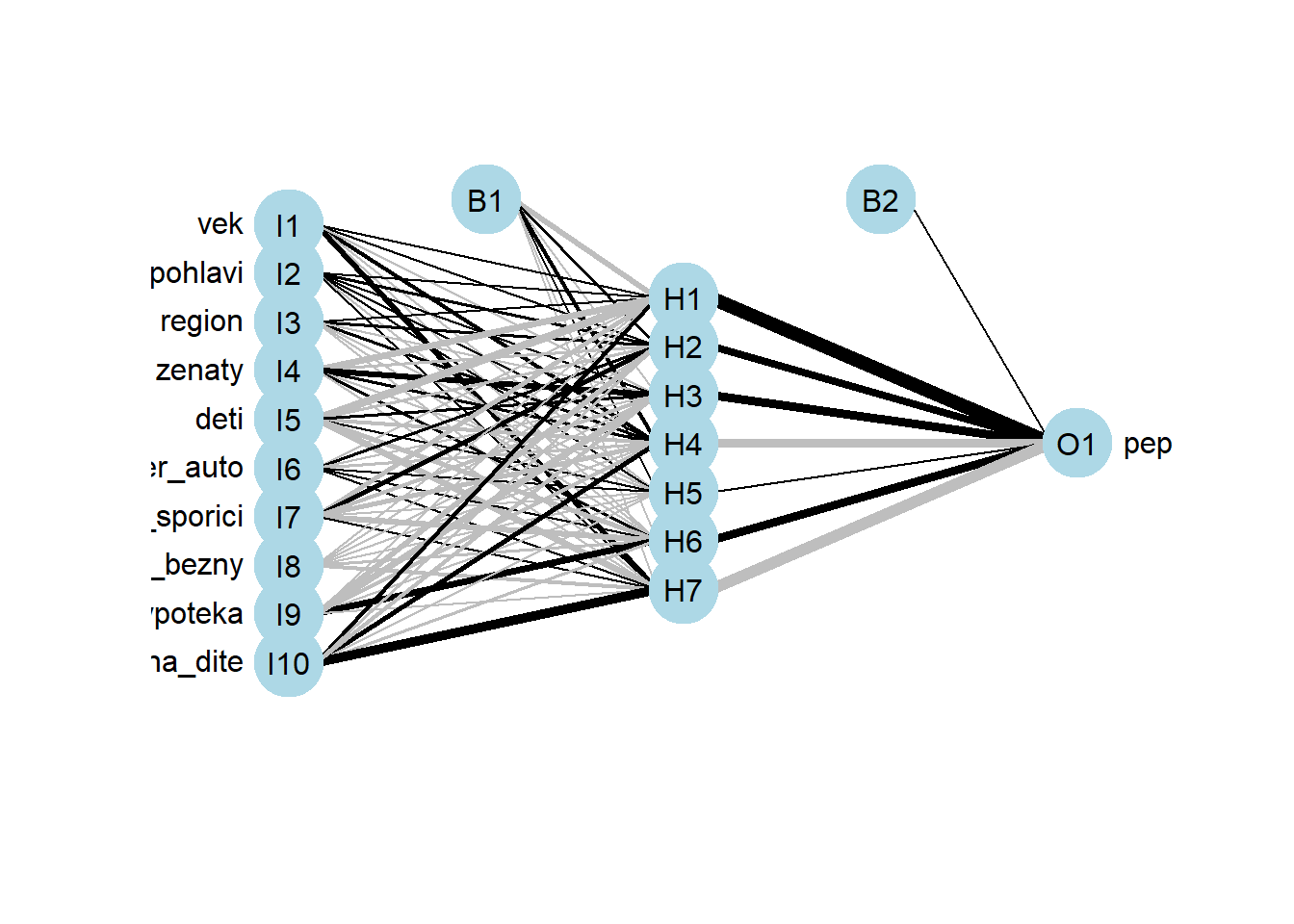
Detection Rate : 0.4784

Detection Prevalence : 0.5415

Balanced Accuracy : 0.8360

'Positive' Class : ano

| plotnet(neurn) *# Neuronová síť* |
| --- |



| varImp(neurn) *# Významnost prediktorů* |
| --- |

Overall

vek 5.831822

pohlavi 3.116713

region 4.192338

zenaty 14.844848

deti 15.707544

uver\_auto 5.936171

ucet\_sporici 14.249732

ucet\_bezny 5.860807

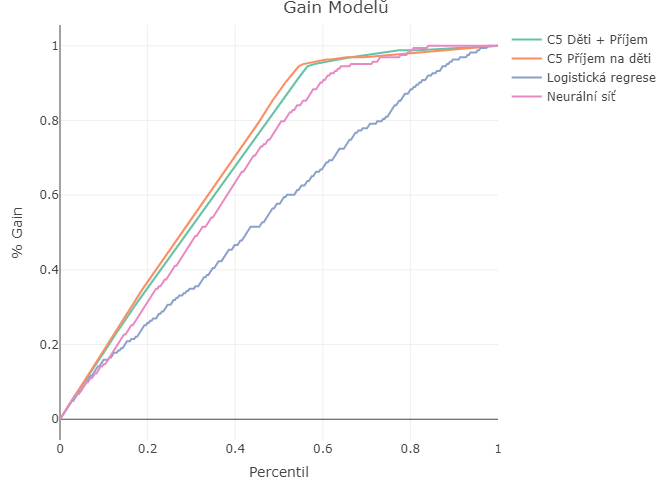
hypoteka 14.273482

prijem\_na\_dite 15.986542

Neuronová síť dopadla dobře - přesnost 83.39%. Po poměrně dlouhé době hraní si s parametry a různými knihovnami, dopadla nejlépe aktuální neuronová síť.

## Gains

| true\_labels <- test\_matrix$pep  *# Model 0: C5 Děti + Příjem* prob\_c5c <- 1 - prob\_c5c pred\_c5c <- prediction(prob\_c5c, true\_labels) gain\_c5c <- performance(pred\_c5c, "tpr", "rpp")  *# Model 1: C5 Děti + Příjem + Příjem na dítě* prob\_c5m <- 1 - prob\_c5m pred\_c5m <- prediction(prob\_c5m, true\_labels) gain\_c5m <- performance(pred\_c5m, "tpr", "rpp")  *# Model 2: Logistická regrese* prob\_lr <- prob\_lr pred\_lr <- prediction(prob\_lr, true\_labels) gain\_lr <- performance(pred\_lr, "tpr", "rpp")  *# Model 3: Neurální síť* pred\_n <- prediction(prob\_n, true\_labels) gain\_n <- performance(pred\_n, "tpr", "rpp")   *# Příprava dat pro plotly* data\_c5m <- data.frame(cumulative\_percent = gain\_c5m@x.values[[1]], cumulative\_gain = gain\_c5m@y.values[[1]], model = "C5 Příjem na děti") data\_c5c <- data.frame(cumulative\_percent = gain\_c5c@x.values[[1]], cumulative\_gain = gain\_c5c@y.values[[1]], model = "C5 Děti + Příjem") data\_lr <- data.frame(cumulative\_percent = gain\_lr@x.values[[1]], cumulative\_gain = gain\_lr@y.values[[1]], model = "Logistická regrese") data\_n <- data.frame(cumulative\_percent = gain\_n@x.values[[1]], cumulative\_gain = gain\_n@y.values[[1]], model = "Neurální síť")  *# vytvoření jedné datové matice se všemi daty* gains\_df <- rbind(data\_c5c, data\_c5m, data\_lr, data\_n)  *# Vykreslení grafu* plot\_ly(data = gains\_df, x = ~cumulative\_percent, y = ~cumulative\_gain, color = ~model, type = 'scatter', mode = 'lines',  line = list(width = 2)) %>%  layout(title = "Gain Modelů",  xaxis = list(title = "Percentil"),  yaxis = list(title = "% Gain"),  showlegend = TRUE) |
| --- |



Opět bych chtěl podotknout, že v HTML dokumentaci je tento graf interaktivní.

# 

# Deployment

Pro deployment byl zvolen model C5, kvůli jeho kvalitním výsledkům při testování. Přesněji se jedná o verzi s příjmem na dítě. Data na kterých byl model využit jsou v databázi “modeler” na serveru 147.230.21.38 v tabulce “Data”.

Tato tabulka má následující strukturu:

| **Proměnná** | **Předpokládaná proměnná** |
| --- | --- |
| id | id |
| vek | vek |
| pohlavi | pohlavi |
| region | region |
| prijem | prijem |
| stav | zenaty |
| deti | deti |
| uver\_auto | uver\_auto |
| ucet\_sporici | ucet\_sporici |
| uset\_beznyt | ucet\_bezny |
| hypoteka | hypoteka |

Pro prediktory je tedy nutné některé proměnné přejmenovat a dopočítat hodnoty “prijem\_na\_dite”. Výsledky predikcí byly přidány k původním datům a zapsány do tabulky DM\_24\_KNESPL\_SKORE2.

V následujícím kódu je připojení se k databázi, načtení dat z databáze a příprava dat pro aplikaci modelu.

| used\_model <- c5\_model *#model, který bude použit pro predikci nad daty z DB*  *# Připojení k DB* con <- dbConnect(odbc(),  Driver = "ODBC Driver 17 for SQL Server", *#ODBC driver, který se má použít pro připojení k serveru*  Server = "147.230.21.38",  Database = "modeler",  UID = "student",  PWD = "student",  Port = 1433) *# Defaultní SQL Server TCP port*  *#Načtení tabulky z databáze* df <- dbReadTable(con, "Data")  *# Očištění dat od diakritiky a od parametru id (není prediktorem)* data\_predictors <- df %>%  mutate(across(where(is.character), ~stri\_trans\_general(., "latin-ascii"))) %>%  select(-id) %>%  mutate(across(where(is.character), as.factor))   *# Přejmenování prediktorů tak, aby na data fungoval můj model* names(data\_predictors)[names(data\_predictors) == "stav"] <- "zenaty" names(data\_predictors)[names(data\_predictors) == "uset\_beznyt"] <- "ucet\_bezny"  *# Dopočítání prijem\_na\_dite prediktoru* data\_predictors$prijem\_na\_dite <- ifelse(data\_predictors$deti == 0,  data\_predictors$prijem,  data\_predictors$prijem /  df$deti) |
| --- |

Zde je vidět využití modelu, příprava dat pro zápis do databáze, vlastní zápis do databáze a ukončení komunikace s databází.

| *# Udělání predikcí* df$prediction <- predict(used\_model, data\_predictors) probability <- predict(used\_model, data\_predictors, type = "prob")  *# Nastavení úrovní predikce* invisible(factor(df$prediction, levels = c("ano", "ne")))  *# Připojení pravděpodobností do tabulky* df <- cbind(df, probability)  *# Přejmenování sloupců s pravděpodobnostmi* names(df)[names(df) == "ano"] <- "probability-ano" names(df)[names(df) == "ne"] <- "probability-ne"  *# Zápis do tabulky*  *# con - jaká databáze*  *# "DM\_24\_KNESPL\_SKORE2" - jaká tabulka*  *# df - co ukládám*  *# overwrite = TRUE - přepisování povoleno*  *# row.names = FALSE - zamezení uložení hlavičky jako záznamu* dbWriteTable(con, "DM\_24\_KNESPL\_SKORE2", df, overwrite = TRUE, row.names = FALSE) dbDisconnect(con) *# Odpojení se z DB* |
| --- |