Úvod do strojového učení v systému R (mh-eHW3)

Jakub Genči

13.5.2022

Dáta

- Auto dataset
  - Atribút ‘name’ sa nepoužíva

```
'data.frame':   392 obs. of  9 variables:
 $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
 $ weight      : num  3504 3693 3436 3433 3449 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
 $ origin      : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2 ...
```

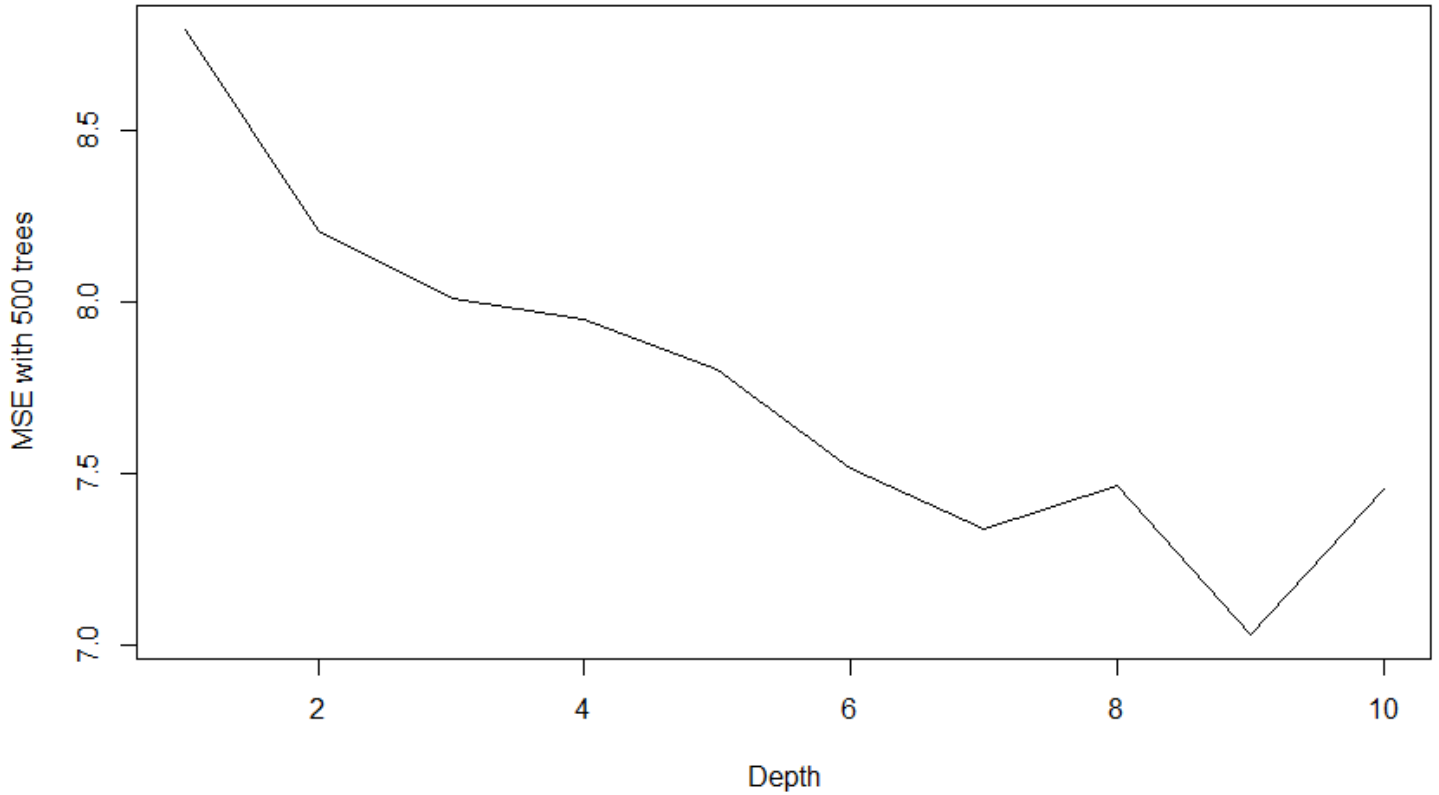
Úloha 1

Task 1 – Boosting Trees and the number of splits d
Build a BT model and experiment with parameter d (`interaction.depth`). Take into consideratin also $d = 1$ (stumps). For different values of d make plots to show how the model performace depends on the number of trees. To estimate generalization error use 8-fold or 4-fold cross-validation.

- Použijeme knižnicu (package) ‘glm’
- Budeme robiť 8-fold cross-validation (parameter cv.folds)

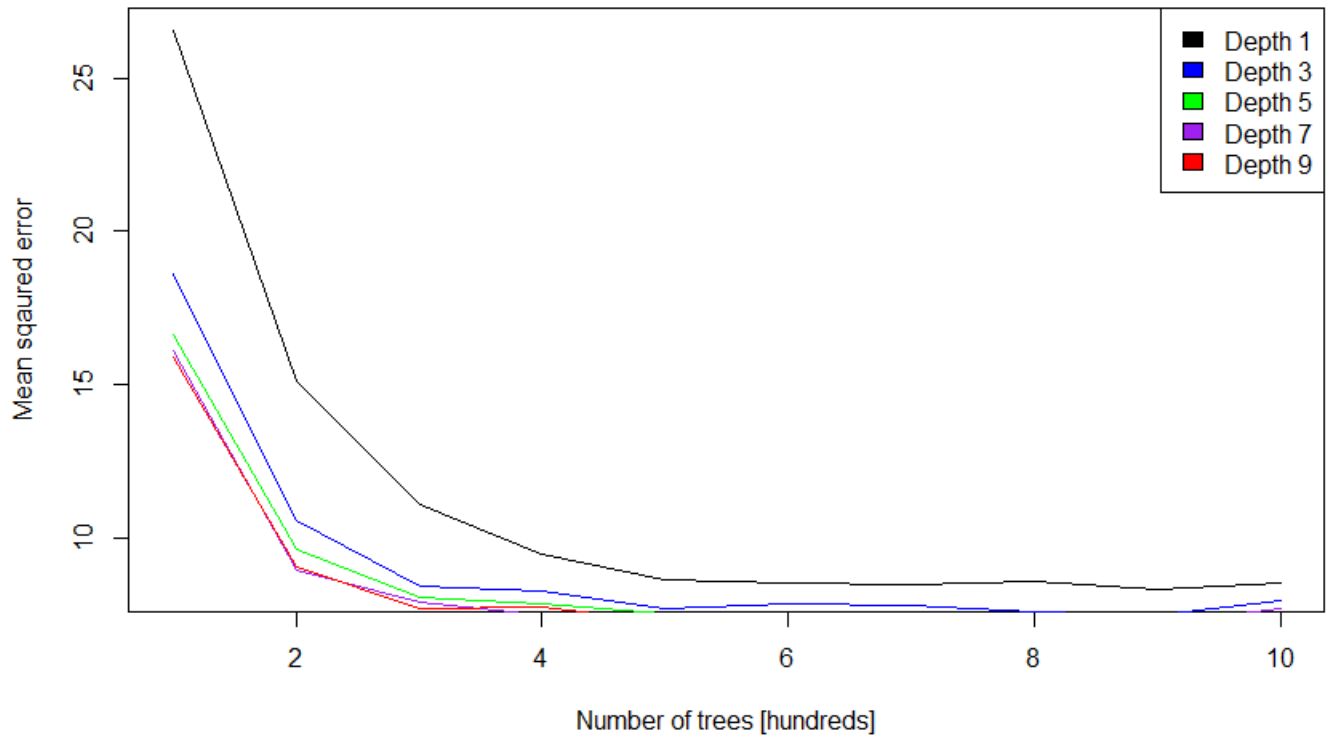
Úloha 1 – parameter depth

- Mean sqaured error
- 500 stromov



Úloha 1 – hĺbka a počet stromov

- Počet stromov od 100 do 1000 (po 100 stromoch)
- Hĺbky od 1 do 10



Úloha 3

Task 3 – Boosting trees and Random Forest
Compare Boosting Trees and Random Forest models with 1000 trees. Try to tune other paramaters to get best performance. To estimate generalization error use 8-fold or 4-fold cross-validation. Which of the two methods is better for the given data?

- Knižnice ‘glm’ a ‘randomForest’
- 8-fold cross-validation

Úloha 3 – parameter tuning

- Boosting trees
  - Žiadna tune funkcia
  - Nebudeme pracovať s parametrami z úloh 1 a 2
  - Zostáva:
    - Minimum pozorovaní v liste
    - Parameter ‘bag.fraction’
- Random Forest
  - nodesize, maxnodes – súvisia spolu

`nodesize`	Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). Note that the default values are different for classification (1) and regression (5).
`maxnodes`	Maximum number of terminal nodes trees in the forest can have. If not given, trees are grown to the maximum possible (subject to limits by `nodesize`). If set larger than maximum possible, a warning is issued.

Výsledky

- Boosting trees:

```
> # Best boosting model
> best_boost_model = gbm(mpg ~ cylinders + displacement + horsepower + weight +
+                acceleration + year + origin,
+              data = d, distribution = "gaussian",
+              n.trees = 1000, shrinkage = 0.01, interaction.depth = 5, cv.folds = 8, n.minobsinnode = 6)
>
> boost_error = mean((model$cv.fitted - d$mpg)^2)
> print(boost_error)
[1] 7.789556
```

- Random forest:

```
> # Best random forest
> nodesize_error = numeric(0)
> for (fold in 1:8) {
+   cv.train <- d[ - cv.index[fold,][cv.index[fold,] > 0], ]
+   cv.test  <- d[ cv.index[fold,][cv.index[fold,] > 0], ]
+
+   RFmodel = randomForest(mpg ~ cylinders + displacement + horsepower + weight +
+                            acceleration + year + origin,
+                          data = cv.train, ntree = 1000, nodesize = 1)
+
+   predictions = predict(RFmodel, cv.test)
+   nodesize_error = c(nodesize_error, (predictions - cv.test$mpg)^2)
+ }
> error = mean(nodesize_error)
> print(error)
[1] 7.274119
```

Užitočné zdroje

- Zadanie:
  - https://ufal.mff.cuni.cz/~holub/2022/docs/MH-eHW3.2022.pdf

- Môj kód
  - https://github.com/GenciJakub/MLinR