Introduction to Statistical Machine Learning for Bioinformaticians and Medical Informaticians

SoSe 2022

Tutor: Dana Petracek

1	Σ

Marina Dittschar & Clarissa Auckenthaler

Assignment 7

(Submitted 27.06.2022)

Task 1

Download and inspect the data set Obesity.csv from moodle. It includes data for the estimation of obesity levels in individuals from the countries of Mexico, Peru and Colombia, based on their eating habits and physical condition. The purpose of this assignment is to predict the categorical response ObesityLevel based on the 16 features and to study which of them are most relevant for that purpose. The first 1500 data points in the file are supposed to be used for training, the remaining 611 data points are the test data. To ensure reproducibility of the results, invoke set.seed(42) every time before you call a function that involves randomness

a) **Learn a decision tree on the training data and display it graphically. What do you observe?**

   We can observe that the main decision variables are Weight, Height, Gender and Age (see Figure2).

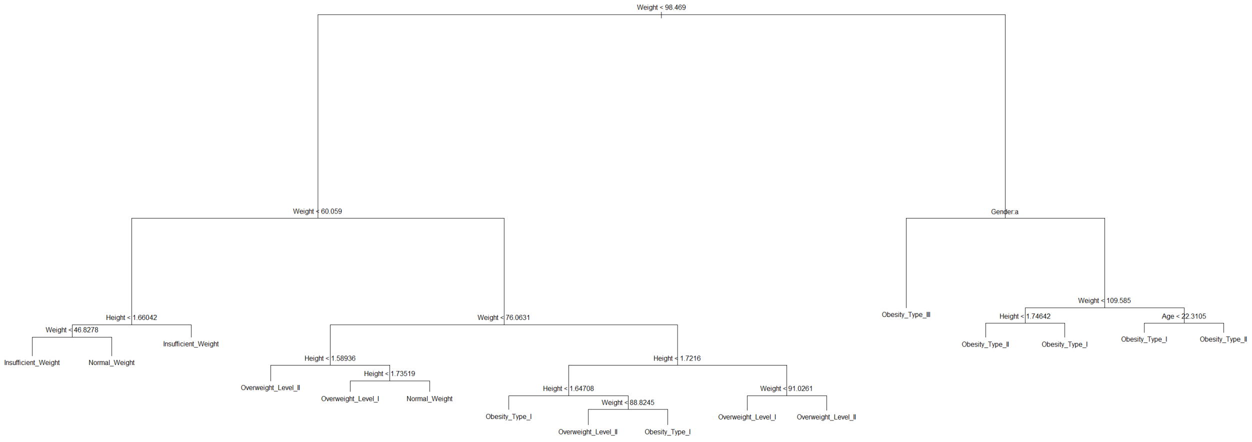
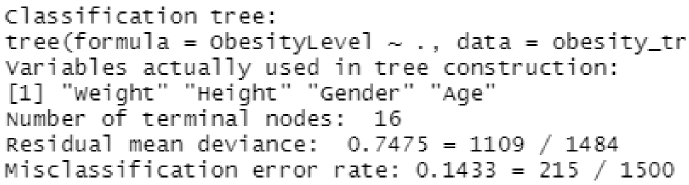


Figure 1: Decision tree for the learned tree model.



```
Classification tree:
tree(formula = ObesityLevel ~ ., data = obesity_train)
Variables actually used in tree construction:
[1] "Weight" "Height" "Gender" "Age"
Number of terminal nodes:  16
Residual mean deviance:  0.7475 = 1109 / 1484
Misclassification error rate: 0.1433 = 215 / 1500
```

Figure 2: Summary of the decision Tree for the learned tree model.

Beside that task we also trained a decision tree with the library *rpart*, which provided slightly different results (Figure 3).

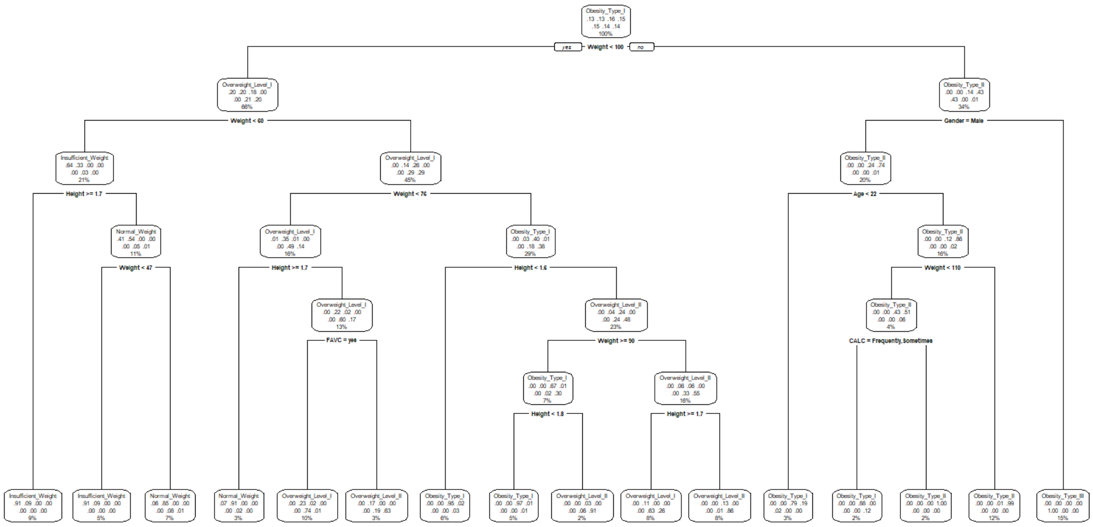
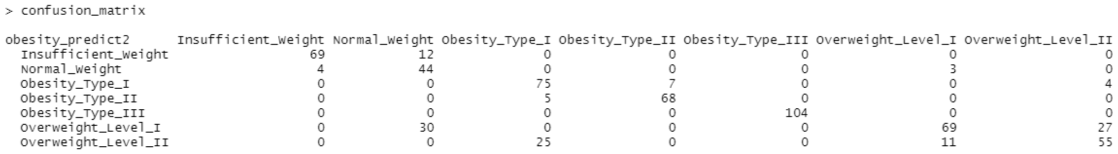


Figure 3: Decision Tree for the learned Tree model

b) **Predict the *ObesityLevel* of the test data points using the learned decision tree, comp ute the accuracy and show the confusion matrix. Which classes are hard to predict?**

Using the tree in Figure 1, we predicted "`ObesityLevel`" of the test set. The computed accuracy was 79.09%.

The confusion matrix is displayed in Figure 4. One can see that "Obesity_Type_III" was especially well predicted, with no false classifications. Meanwhile, "Insufficient_Weight" and "Obesity_Type_II" had some misclassifications, while "Overweight_Level_I" had over 10% misclassifications. For "Overweight_Level_II" and "Obesity_Type_I" over 30% of samples were being misclassified. "Normal_Weight", with over 50% of samples being misclassified, had the lowest classification accuracy. In summary, we can say that especially the classes "Normal_Weight", "Obesity_Type_I" and "Overweight_Level_II" were hard to predict.



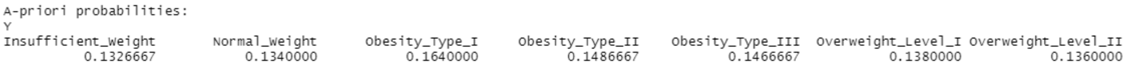
```
> confusion_matrix

obesity_predict2      Insufficient_Weight Normal_Weight Obesity_Type_I Obesity_Type_II Obesity_Type_III Overweight_Level_I Overweight_Level_II
  Insufficient_Weight                  69            12              0               0                0                  0                   0
  Normal_Weight                         4            44              0               0                0                  3                   0
  Obesity_Type_I                        0             0             75               7                0                  0                   4
  Obesity_Type_II                       0             0              5              68                0                  0                   0
  Obesity_Type_III                      0             0              0               0              104                  0                   0
  Overweight_Level_I                    0            30              0               0                0                 69                  27
  Overweight_Level_II                   0             0             25               0                0                 11                  55
```

Figure 4: Confusion matrix for the computed decision tree.

c) **Learn a Naive Bayes classifier on the training data, inspect the parameters (no need to include all of them into the report), calculate test data accuracy, and compare the results to the decision tree. What do you observe? What is a likely explanation for your observation?**

The model provides the following parameters for the a-priori probabilities (Figure 5) for each response class: Insufficient_Weight 13.27%, Normal_Weight 13.40%, Obesity_Type_I 16.4% , Obesity_Type_II 14.87% , Obesity_Type_III 14.67%, Overweight_Level_I 13.8%, Overweight_Level_II 13.6%. We can observe that nearly all classes have approximately the same probabilities to occur, only Obesity_Type_I has a higher probability compared to the other classes. In addition to the a-priori probabilities, the model also provides conditional probabilities for each parameter and class. For each categorical variable the parameter table gives, for each attribute level, the conditional probabilities given by the target class (Gender, Smoke,...). For the numeric variable, the table gives, for each target class, mean and standard deviation of the (sub-)variable (example Weight, Height,...).

The computed test accuracy for the naive Bayes classifier is 68.14%. This is much lower than the accuracy obtained with the tree model described in a). Because the Bayes classifier, in providing conditional probabilities instead of single decision boundaries, is more closely adapted to the training data, the test accuracy for the test data set could have suffered.



```
A-priori probabilities:
Y
Insufficient_Weight       Normal_Weight      Obesity_Type_I     Obesity_Type_II    Obesity_Type_III  Overweight_Level_I Overweight_Level_II
          0.1326667           0.1340000           0.1640000           0.1486667           0.1466667           0.1380000           0.1360000
```

Figure 5: Priori probabilities for the Bayes classifier model.

d) **Learn a bagged tree ensemble and a random forest with default hyperparameters on the training data and calculate the test data accuracy for both models. What do you observe?**

We can observe a test accuracy of 95.59% for the random forest model and a test accuracy of 95.42%, associated with the bagged tree model. This means the random forest model and the bagged tree model have approximately the same test accuracy.

e) **Tune the mtry-parameter of the random forest by five-fold cross validation. For this purpose, split the training data in consecutive blocks (first block contains data points 1-300, second block 301-600, ...; the data is already shuffled). Plot mean cross validation accuracy (incl. standard errors) for each mtry value and mark the maximal value, $\hat{mtry}$.**

In Figure 6, we can see that the maximum accuracy occurs at mtry = 9 with a value of 95.60%.

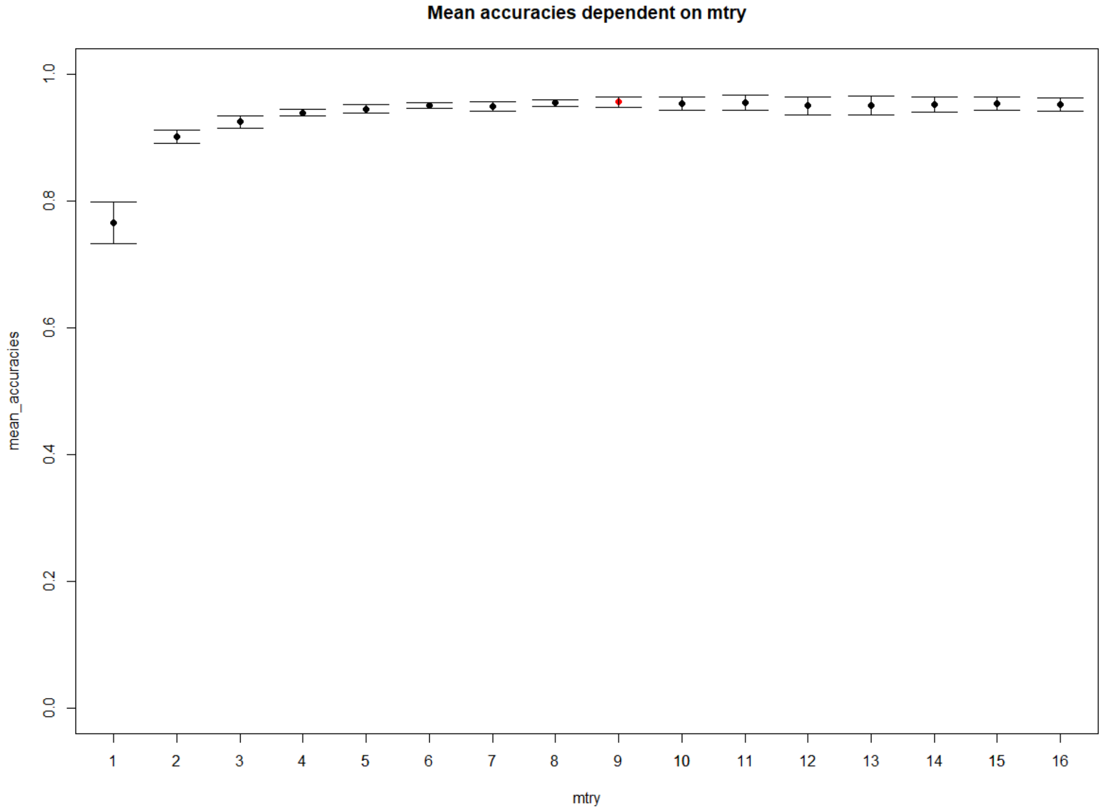
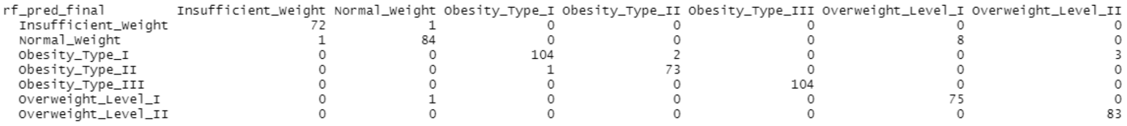


Figure 6: Mean cross validation accuracy (incl. standard errors) for each mtry value.

f) **Learn a random forest with the selected $\hat{mtry}$ from the previous subtask on the entire training data set. Calculate test data accuracy, confusion matrix, and variable importance (mean decrease Gini). Compare with all previous results.**

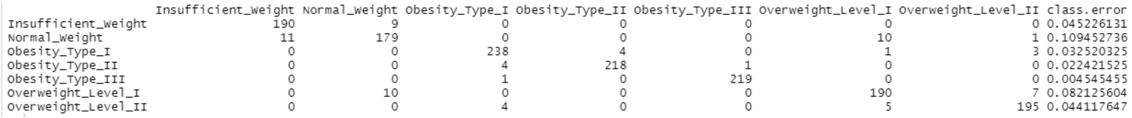
The test accuracy of the model trained with the optimal *mtry* has a value of 97.06%. Compared to the other models created in e), c), and a), the accuracy of the model calculated in f) is the highest. The confusion matrix for the test set (see Figure 7) shows us, that nearly all samples are classified correctly. For the class "Obesity_Type_III", all samples are correctly classified (0% misclassifications), only

"Overweight_Level_I" has over 10% misclassifications, the other classes have misclassification rates between 0.9% and 3.6%. Compared to the confusion matrix from the model obtained in b), the misclassification rate is much lower on the test set. Because we were uncertain whether we should provide the confusion matrix on the trained model or on the test performance, in Figure 8 you can see the confusion matrix obtained directly after training the model. Here, the ranges of the classification errors are similar, with only "Normal_Weight" exceeding a classification error of 10% and all other classification errors ranging below that.



```
rf_pred_final          Insufficient_Weight Normal_Weight Obesity_Type_I Obesity_Type_II Obesity_Type_III Overweight_Level_I Overweight_Level_II
  Insufficient_Weight                   72             1              0               0                0                  0                   0
  Normal_Weight                          1            84              0               0                0                  8                   0
  Obesity_Type_I                         0             0            104               2                0                  0                   3
  Obesity_Type_II                        0             0              1              73                0                  0                   0
  Obesity_Type_III                       0             0              0               0              104                  0                   0
  Overweight_Level_I                     0             1              0               0                0                 75                   0
  Overweight_Level_II                    0             0              0               0                0                  0                  83
```

Figure 7: Confusion matrix for the random forest model with the selected $\hat{mtry}$.



```
                    Insufficient_Weight Normal_Weight Obesity_Type_I Obesity_Type_II Obesity_Type_III Overweight_Level_I Overweight_Level_II class.error
Insufficient_Weight                 190             9              0               0                0                  0                   0 0.045226131
Normal_Weight                        11           179              0               0                0                 10                   1 0.109452736
Obesity_Type_I                        0             0            238               4                0                  1                   3 0.032520325
Obesity_Type_II                       0             0              4             218                1                  0                   0 0.022421525
Obesity_Type_III                      0             0              1               0              219                  0                   0 0.004545455
Overweight_Level_I                    0            10              0               0                0                190                   7 0.082125604
Overweight_Level_II                   0             0              4               0                0                  5                 195 0.044117647
```

Figure 8: Confusion matrix obtained directly from the model after training

The values of the mean decrease Gini are shown in Figure 10. On the right side of Figure 9, the mean decrease Gini of each parameter after training the model is displayed. "Weight" can be clearly identified as the most important variable. "Height" and "Gender" are the second and third most important variables. Regarding variable importance, we see similarities to previous subtasks. For example, the illustrated decision tree in a) shows most decisions to occur in the categories that are shown to be the most important in this random forest model, namely "Weight", "Height", "Gender" and "Age".

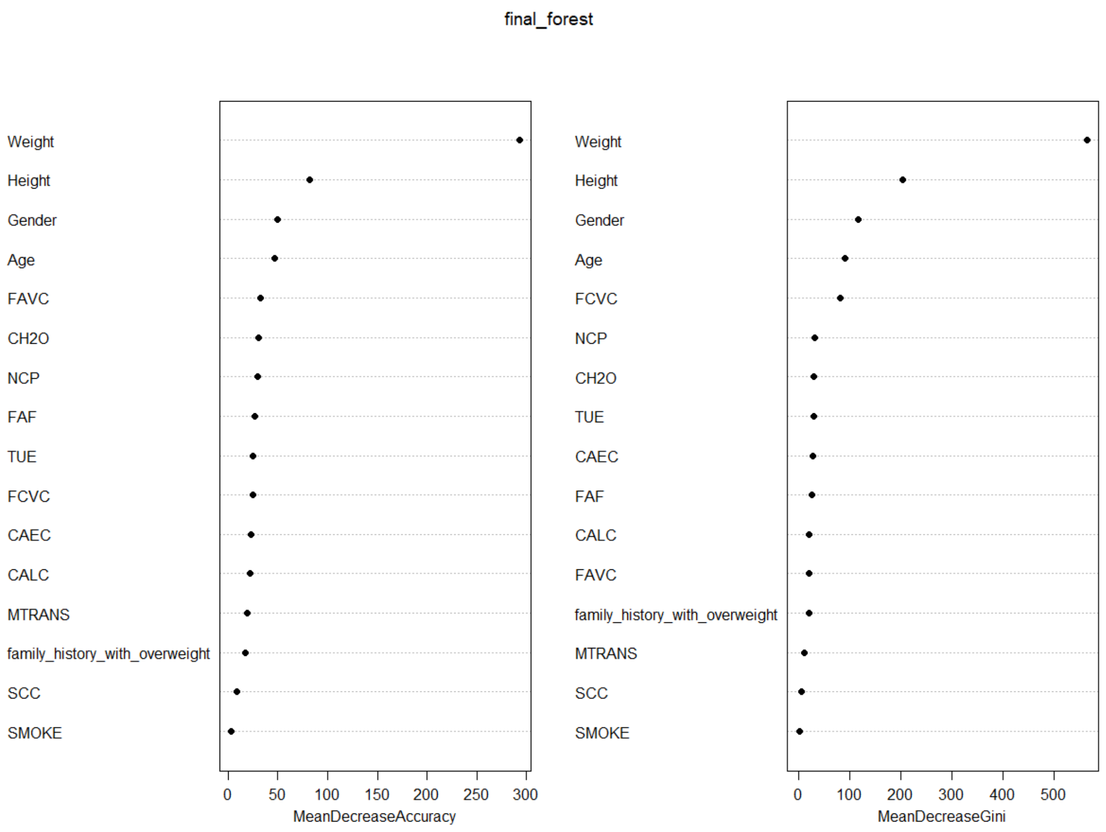
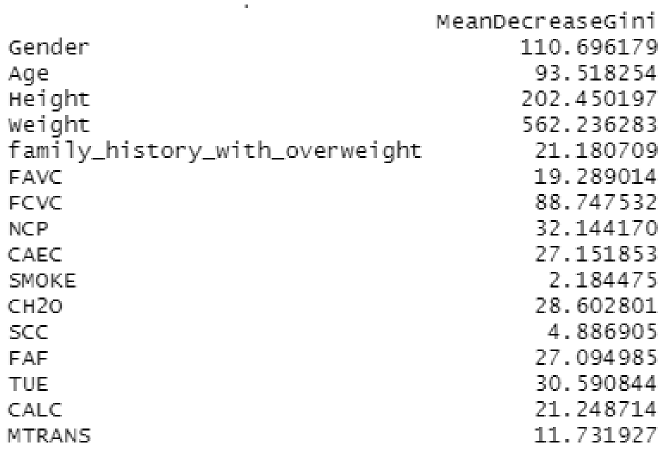


Figure 9: Variable importance mean decrease accuracy and mean decrease Gini.



```
                                MeanDecreaseGini
Gender                                110.696179
Age                                    93.518254
Height                                202.450197
Weight                                562.236283
family_history_with_overweight         21.180709
FAVC                                   19.289014
FCVC                                   88.747532
NCP                                    32.144170
CAEC                                   27.151853
SMOKE                                   2.184475
CH2O                                   28.602801
SCC                                     4.886905
FAF                                    27.094985
TUE                                    30.590844
CALC                                   21.248714
MTRANS                                 11.731927
```

Figure 10: Mean decrease Gini for the random forest model with the selected $\hat{mtry}$.

g) **Tune mtry according to the out-of-bag (OOB) error and compare to the CV-based tuning, both in terms of CV/OOB error for each mtry-value and performance of the optimal models on the final test data set. Hint: Remember the inverse relationship between (classification) error and accuracy.**

When we perform model training with different values for *mtry*, we can observe that the best-performing model in regards to the OOB-error also has an *mtry*-value of 9 (see Figure 11). The OOB-error for this model is 2.8%. In Figure 12, one can see that the error for the OOB-tuned model is consistently lower than that of the CV-tuned model, but the curve follows the same path. While the displayed errors are different however, the accuracies on the test data between different models are equivalent. For the OOB-validated model the accuracy is 97.22%, while the model accuracy for the best-performing CV-validated model is 97.08%. The difference in errors in Figure 12 could be explained because the validation sets are different in the CV vs. the OOB procedure (the CV-models are evaluated by folds of the training set while the OOB model is evaluted on the whole test set).

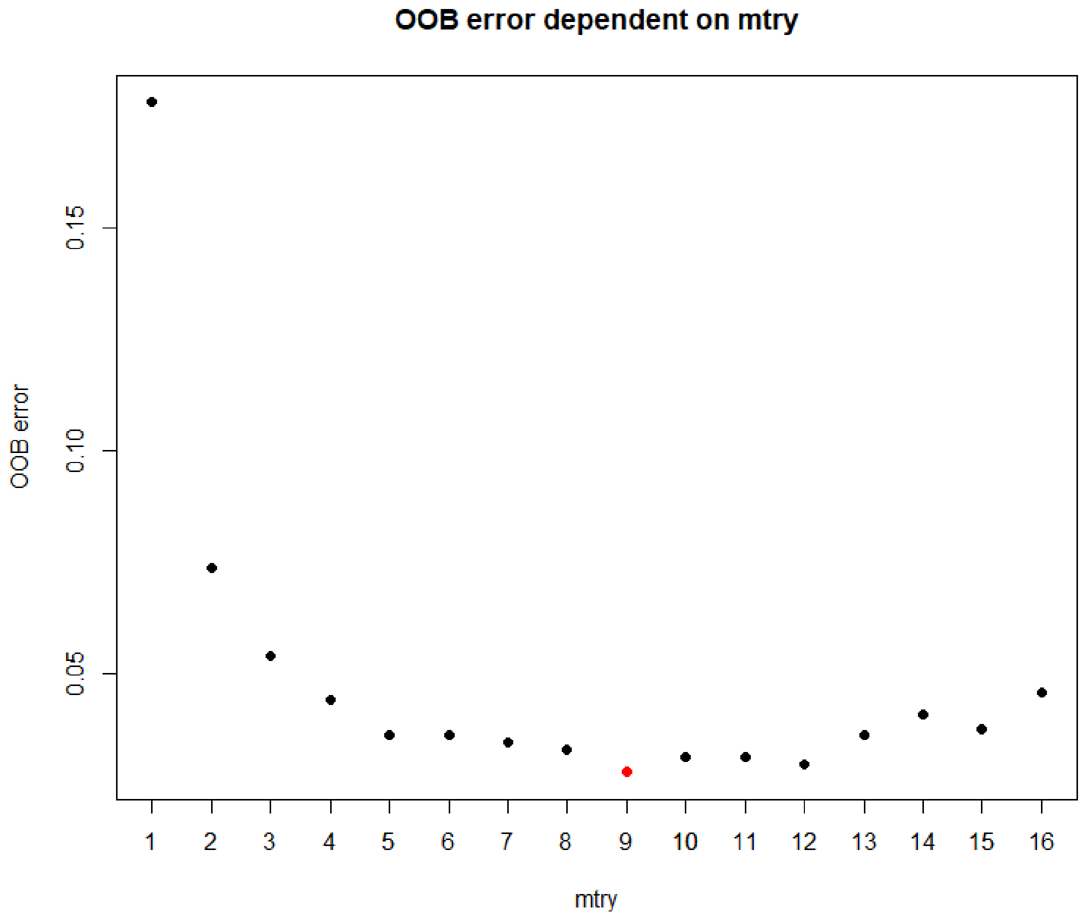


Figure 11: OOB error for each $m\hat{t}ry$, with the lowest value marked

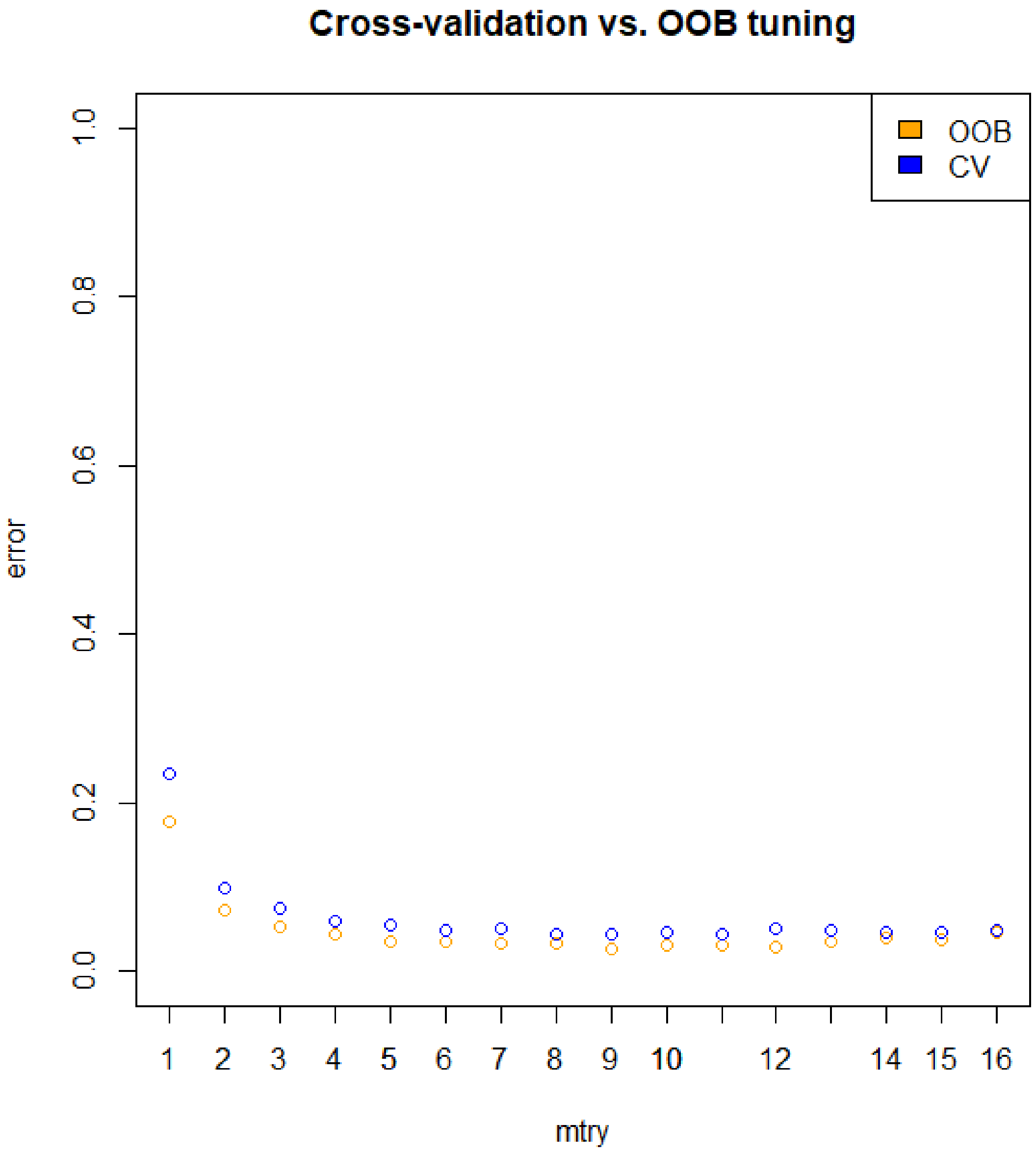


Figure 12: CV error vs. OOB error