Exercise 10

Advanced Methods for Regression and Classification

January 10, 2019

Random forests: function `randomForest()` from the R package `randomForest`

Take the bank data set (see previous exercises) for random forest classification. The goal is to predict if the client will subscribe a term deposit or not. This information is represented by the binary variable y (last one). Select randomly a training set of a reasonable size, compute the classifer, and evaluate the classifier based on the test set.

1. Use the option `importance=TRUE` in the function `randomForest()`, and plot the result object with `plot()` and `varImpPlot()`. How can you interpret these plots?

2. Try to improve the misclassification error of the "yes" clients (by keeping the overall misclassificatione error still small) with different strategies.

   (a) Oversampling, undersampling, same-size-sampling.

   (b) Modify the parameter `sampsize` in the `randomForest()` function. What is it doing?

   (c) Modify the parameter `classwt` in the `randomForest()` function. What is it doing?

   (d) Modify the parameter `cutoff` in the `randomForest()` function. What is it doing?

   (e) Modify the parameter `strata` in the `randomForest()` function. What is it doing?

   (f) Use the function `SMOTE` from the package `DMwR` to generate new artificial observations for the smaller class. Afterwards, apply `randomForest()` to the new data set. Does the performance improve?

   (g) Does it make sense to combine some of the above approaches? Any other ideas?

   Which approach leads to the overall best solution (and is simple to implement)? Apply the best strategy also on the whole data set *bank-full.csv*.

Save your (successful) R code together with short documentations and interpretations of results in a text file (= R script file), named as *Matrikelnummer_10.R* (no word document, no plots). Submit this file to Exercise 10 of our tuwel course (deadline January 9).