Exercise Lesson06: Random Forest

Machine Learning using R

Exercise 1: Comparison between linear regression and Random Forest

We want to compare the predictive performance of a Random Forest with a standard linear regression model using the (artificial) "rf_lm_data.csv" data set.

a) Read in the dataset using the `read.csv('rf_lm_data.csv', stringsAsFactors=TRUE)` command. Check the structure of the data set, what types of predictor variables are present? How many observations are in the data set?

b) In this exercise we will work with a separate training and test data set (instead of crossvalidation). Randomly divide the observations in our data into a training and test set. The training set should include 70% of the data while the remaining 30% will be our test set. Don't forget to set the random seed (`set.seed(1241)`).

c) Fit a linear regression model to the training data, with `y` as the target variable. Use the command `lm(y ~ ., data = NAME_OF_TRAINDATA)` to fit the model. Look at the results of the linear model (**Hint**: `summary(MODELFIT)`). What can you see?

d) Fit a Random Forest using the `cforest()` function to the training data. The Random Forest should have 500 trees. Set `mtry` to the number of predictors divided by 3.

e) Now we want to see how well the linear model and the Random Forest predict the target variable in the test set. Predict the `y` values in the test set with both models and save the predicted values (**Hint**: Use the `predict(MODELFIT, newdata=...)` function for both models). **Extra**: Calculate the RMSE for both models. The RMSE is the "root-mean-square-error" and is calculated by taking the differences between the predictions and the true `y` values, squaring them, taking the mean of the squared values and finally taking the square-root of the calculated mean value. The RMSE expresses how bad the predictions were, the larger the RMSE, the more imprecise were the predictions.

f) We want to visually compare the predictions made by the Random Forest and the linear regression model. To this end, create two scatterplots. For both models, plot the predicted `y` values (on the x-axis) vs. the true `y` values in the test set (on the y-axis). Compare the two plots, how should the points ideally lie?

Exercise 2: Wholesale customers data

The data set includes measurements regarding the spending behavior of clients of a wholesale distributor. It includes the annual spending in monetary units on diverse product categories. Check the link "https://archive.ics.uci.edu/ml/datasets/wholesale+customers" for more information on the variables' meaning. We will use this dataset to get experience working with Random Forest. Specifically, we will try to model the `Channel` variable (2-level factor, denoting whether the client is active in retail or as a hotel/restaurant/cafe) using the remaining variables as predictors.

a) Read in the data set `Wholesale_customers_data.csv` and get a first impression of the data. Recode categorical variables if necessary.

b) Fit a Random Forest model (**Hint**: `cforest()`) to the data using `Channel` as the target variable and the rest of the variables as predictors. Set the number of trees to 500, and the number of inspected

variables at each split to 3. Create the OOB-confusion table and calculate the OOB-error rate. Since the Random Forest includes random sampling, fix the random seed.

c) **extra:** We now want to compare the Random Forest's performance with the performance of a simple decision tree. Fit a decision tree (**Hint**: `ctree()`) to the data. Since there is no OOB-error in decision trees, you can use our cross-validation function (last exercise) to estimate the test-error of the tree (use e.g. 10-fold cross validation).

d) We want to get an impression of which variable is most important in our Random Forest fit. Compute and plot the variables' (permutation) importance scores. Which variable seems to be most important? (**Hint:** You can use the `implot()` function from the slides)

e) We want to get a better idea of the relation between the target variable `Channel` and the `Detergents_Paper` variable. Can we already see a relation between the two variables when plotting them against each other? Draw a boxplot comparing the `Detergents_Paper` values between the two `Channel` levels.

Exercise 3: Kaggle data set

The present data set is from a Machine Learning competition. In such competitions a data set is provided and participants are invited to train a model of their choice on the training data. Later, a separate test data set is used to score the predictive performance of the models. Because the test-set is not available, we will only work with the training data and compare the performance of a Random Forest (using OOB-error) and a decision tree (using cross-validation).

a) Read in the `kaggle_train.csv` data set. What are the dimensions and the structure of the data set?

b) **extra:** Fit a Random Forest to the data. Since all the predictors are numerical, we do not have to rely on the (more time-consuming) bias-free variable selection. Here we use the "classic" package `randomForest` to fit a Random Forest model based on the Gini-index (**Hint**: library(randomForest); `randomForest(target~., kagg)`). The function includes a default option of 500 trees. The number of variables selected in each split is set automatically depending on the dimension of the data (if no specific value is specified). Look at the (randomForest) output, it contains the OOB-error. Also fit a decision tree to the data and check its performance with 10-fold cross-validation (We will use our self-made cross-validation function, which is based on the `ctree()` function from the `party` package). (**Hint**: Since the data set is quite large, fitting the models will take up a lot of time (ca.1hour), make sure you set the random seed before running your code).

How does the decision tree perform compared to the Random Forest?

c) The `randomForest()` function calculates variable importances. Plot the variable importances using the `varImPlot(YOUR_FOREST_OBJECT)` function included in the package. Which three variables seem to be most important?