Exercise:
Decision Trees

Machine Learning and Prediction Modelling

Exercise 1: Diabetes in Pima Indian women

This data set includes the test-results of women who are of Pima Indian heritage and were tested for diabetes according to Wold Health Organization criteria.

a) The `Pima.tr` data set is available in the `MASS` package. Load the package to be able to access the data. Get an overview of the data and use the command `?Pima.tr` to see the individual variables' meaning.

b) We want to model the variable `type` (does the woman have diabetes: Yes/No) using a decision tree. Fit a decision tree to the data using the `ctree()` function (**Hint**:`library(partykit)`). Look at the output of the created object.

c) Plot the tree structure. Look at all the information printed on the generated figure. Which variables were used to split the data? What is the meaning of the p-values printed below the splitting variables?

d) The `MASS` package also contains a `Pima.te` data frame which is supposed to serve as a test data set for the `Pima.tr` data. Calculate the test error (misclassification rate) of our fitted tree using `Pima.te` as the test data. How does the tree perform?

Exercise 2: Plasma glucose in Pima Indian women

In this exercise we work again with the `Pima.tr` and `Pima.te` data from the `MASS` package. This time, we want to predict the numeric variable `glu` which indicates the plasma glucose concentration in an oral glucose tolerance test. Therefore, we are performing a regression task and not a classification task like in the previous exercise.

a) Fit a decision tree with the `ctree` function to the `Pima.tr` data. Plot the created tree and look at its structure. Which variables were used for splitting? What do the boxplots in the end nodes show?

b) Using the fitted decision tree, predict the `glu` values in the Pima.te data set. Since we are not doing classification we cannot produce a confusion matrix or calculate a misclassification rate. To compare the predicted `glu` values with the true `glu` values in `Pima.te` we instead calculate the mean squared error (MSE) of the predictions. The mean squared error is simply the mean of the squared differences between the predicted and the true `glu` values (formula below). Therefore, the smaller the MSE the closer are the predictions to the true values.

$$MSE = \frac{1}{n}\sum_{i}^{n}(\widehat{glu_i} - glu_i)^2$$

c) We want to compare the performance of the decision tree with the performance of a simple linear model. Fit a linear regression model to the `Pima.tr` data with `glu` as the target variable. (**Hint**: `lm()`)

d) Using the fitted linear model, predict the `glu` values in the `Pima.te` data and calculate the corresponding MSE. Looking at the MSE, how did the linear model perform compared to the decision tree? (**Hint**: `predict()`)

Exercise 3: Crossvalidation with decision tree

In the first exercise, we have predicted the binary variable `type` and have estimated the test error by calculating the missclassification rate on the test data (`Pima.te`). In this exercise, we want to estimate the test error using cross validation. Try to adapt the cross validation function which we wrote for the k-nearest-neighbor classifier in the previous session, so that it can be used for a decision tree. Use the adpated function to estimate the test error for the prediction of `type`. For the cross validation, you can combine `Pima.tr` and `Pima.te` into one data frame.