# Lab03: Classification and Regression Prediction Models

**Handed out:** Wednesday, March 8, 2023

**Return date:** Friday, March 24, 2023, at the eLearning link **Lab03Submit** in the **Lab03** folder.

**Objectives:** Comparison of different classification as well as regression tree models and the evaluation of their predictive properties.

**Grades:** This lab counts 9 % towards your final grade

**Format of answer:** Your answers (statistical figures and verbal description) should be submitted electronically as Word document. Add a running title with the following information: Lab03, your name and page numbers. Use this document as template: add your answers for each subtask, i.e., 1 (a) etc., in a red color as well as any requested statistical figures. Trial and error answers will lead to a deduction of points. You are expected to hand in professionally formatted answers: use a fixed pitch font, like **Courier New**, for any  code and output.

## Part 1: Classification trees [6 points]

You will be using for this part the dataset **mushrooms.csv** and split it into a stratified ***training*** data-frame with 2/3 of the observations and ***test*** data-frame with 1/3 observations. The dependent variable is **type**. Remove the variable **veil\_type** with **mushrooms$veil\_type <- NULL** because it is constant over all observations.

**Task 1:** Build a classification tree, properly prune the tree, and interpret the pruned tree. Show both the pruned and unpruned trees. Use the ***training*** data-frame. [1 points]

**Task 2:** Build a predictive model using the **randomForest** function with bagging based on the parameter **ntree**. Evaluate the relevance of the features. Use the ***training*** data-frame. [1 point]

**Task 3:** Build a predictive random forest model using the **randomForest** function and find the optimal hyper-parameter **mtry** for the number of features explored at each steps. Use the ***training*** data-frame. [1 point]

**Task 4:** Build a predictive boosted tree model using the function **gbm** and find the optimal depth hyper-parameter **interaction.depth**. Use the ***training*** data-frame. [1 point]

**Task 5:** Compare the models from tasks 1 to 4 for the ***test*** data-frame by using their **ROC** curves, the **auc** statistic, their prediction error rate and their overall node purity. Justify which model you would use to avoid mushroom poisoning. [2 points]

## Part 2: Regression trees [3 point]

You will be using for this part the dataset **redwines.csv** and split it stratified with just 3 matching strata into 2/3 ***training*** data and 1/3 ***test*** data. The dependent variable is **quality**.

**Task 6:** Build a pruned regression tree with all feature variables and interpret the pruned tree. Show the pruned and unpruned trees. For model calibration use the ***training*** data-frame. Calculate the model fit for the ***test*** data-frame. [1 point]

**Task 7:** Calibrate for the ***training*** data-frame with all feature variables a random forest model and identify its optimal hyper-parameters **ntree** and **mtry**. Evaluate the variable importance. Calculate the model fit for the ***test*** data-frame. [1 point]

**Task 8:** Calibrate for the ***training*** data-frame with all feature variables a boosted model and identify its optimal depth hyper-parameter **interaction.depth**. Calculate the model fit for the ***test*** data-frame. [1 point]