# Decision Trees and Random Forests

## Problem I: Decision Trees, Application to Real Data

In this exercise, you will experiment with implementing a decision tree for binary classification.

1. Begin with an exploratory descriptive analysis of your dataset (variable typology, missing values, observation distributions for each variable, boxplots, etc.).
2. Use a decision tree implementation in R to build a classifier aimed at predicting the dependent variable's class in your dataset based on the independent variables. Follow these guidelines:
   (a) Optimize your classification tree by employing techniques covered in class (e.g., hyperparameter optimization via cross-validation, pruning, etc.).
   (b) Visualize the optimal tree generated according to the operational procedure.
3. You will be evaluated based on the performance of your optimal model on test data:
   (a) Explore the classification results:
      i. Detailed Accuracy By Class (Precision, Recall, etc.)
      ii. Confusion Matrix
      iii. Calculate as many performance metrics as possible and justify each result.
   (b) Include in the report the ROC curve and AUC for your optimal model, along with the corresponding R code.
   (c) Provide a general conclusion/interpretation of the results. Feel free to go beyond the requested questions. Any added-value analysis will be highly appreciated.

## Problem II: Random Forest, Application to Real Data

In this exercise, you will experiment with implementing a random forest of decision trees.

1. Consider the same data sample from Problem I. Use different implementations (at least two) of decision tree forests in R, and for each implementation, clearly present the reasoning steps that led to the optimal model (Out-of-sample estimation, Cross Validation, Pruning, hyperparameter optimization, etc.).
2. Analyze and compare the classification errors:
   - Detailed Accuracy By Class (Precision, Recall, etc.)
   - Confusion Matrix
   - ROC Curves, AUC, etc.
3. Interpret the performances of the different random forest implementations from parts (1 and 2).