Instructions

- This homework assignment is worth 130 points.
- Please submit a **.ipynb** file to Blackboard.
- **Please strive for clarity and organization.**
- **Due Date: February 17, 2023 by 11:59 pm.**

For this homework assignment and for future one, we will work on the challenge presented in the `data mining cup 2019`. Please read the task and get familiar with the data. For this week homework assignment, answer the following:

Exercise 1

(5 points) Using the bucket, that you create in the last homework assignment, and the pandas library, read the `train.csv` and `test.csv` data files and create two data-frames called `train` and `test`, respectively.

Exercise 2

(55 points) Using the `train` data-frame, engineer at least 7 different features, that can help to predict `fraud`, using the given input variables. Engineer the same features, that you engineer on the `train` data-frame, on the `test` data-frame. These are the rules to engineer the features:

- You can use the Box-Cox transformation only one time.
- You can't use neither 0-1 scaling nor z-score standardization.

Exercise 3

(70 points) Using the `train` data-frame, and the features that were engineered in Exercise 2, engineer at least four interactions as follows:

(a) Engineer three feature by using the *strong heredity* principle. That is, identify the top three features and engineer the interaction between them. For examples, let's assume X_3, X_{12} and X_8 are the top three features, then the interactions would be $X_3 \times X_{12}$, $X_3 \times X_8$, and $X_{12} \times X_8$. Notice that you can use a tree-based model (since you can extract feature importance) to identify important features. You can follow these steps:

(i) Split the `train` data-frame (including the features that were engineered in homework assignment 4) intro `training` (80%) and `testing` (20%) (taking into account the proportions of 0s and 1s).

(ii) Build a random forest model with 500 trees and depth equal to 3 on the `training` data-frame. Extract and store the importance of each of the features.

Repeat (i)-(ii) 100 times. Then, compute the average importance for each of the features.

(b) Engineer at least one feature by building a decision tree model (with depth at most equal to 4) on the `train` data-frame and identifying interesting patterns in the decision tree. Run this procedure a few times (with different samples of the `train` data-frame) to make sure that pattern in consistent and not just random chance.

(c) Engineer the same features from parts (a) and (b), on the `test` data-frame.