**Assignment 03 – Random Forest and Stacking**

There are two steps in this assignment: 1) Training and testing most of the algorithms over the given dataset, and 2) Using the best algorithms (classifiers) to generate a stacked model. You will also export the model in a pickle file, load it again and test it on one of the feature vectors from the dataset. You will submit both the ipynb and pdf files for the ensemble export. Total points = 75 (no extra points in this assignment).

You will use the credit card dataset uploaded here. Please start with a random.seed(last four digits of your GUIDg). The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced. The positive class (frauds) accounts for 0.172% of all transactions.

All the variables are continuous and have been transformed by using PCA. Features V1, V2, … V28 are the principal components obtained with PCA. The only features that have not been transformed with PCA are 'Time' and 'Amount.' Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount. This feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise. The goal is to predict the class.

Since this is a classification dataset, for the first part, you will follow the steps in assignment 2 (and some more) as shown below. The code is already with you, and you can use that.

**I have intentionally misspelled some explanations in assignments. When you copy the text, make sure to spell check it. I will drop points for incorrect spelling and grammar (-5 points). Please also provide explanations/ reasoning for utilizing a command and describe the output. Failure to describe/ explain will result in lower points. Also, make sure to add a g#groupnumber (for example, g01) suffix to each variable, so we will know you are not just copy-pasting the given example code.**

# Load the dataset. (1)

Load the csv file as a pandas data frame. And display the first six points

# Show first 6 data points using head(). (1)

# Describe pandas Dataframe by using describe. (1)

# Show correlation heat plot of the entire dataset using matplotlib and sns, choose any color pallet (except blue) you like (experiment). (1)

# Show the Scatterplot matrix for the dataframe (avoid matplotlib and sns for this assignment). You can use [Scatterplot Matrix Plotly](https://plotly.com/python/splom/). Use the code for the second image that shows different colors for classes. In this case, you will get two colors since we have two classes. Also, experiment with visual aspects of the image (not a lot, but an excellent visual will always leave a better impression. you can change color, thickness, font, font size, font color, etc.). No need to explain the plots but do save them in a pdf/svg/png with either static export function or html export function from plotly [Interactive HTML Export Plotly](https://plotly.com/python/interactive-html-export/). (10)

# Split the dataset into the Training set and Test set. Choose your preferred split and justify the rationale. (1)

# Perform classification routine by using LogisticRegression(), KNeighborsClassifier(), DecisionTreeClassifier(), SVC(), GaussianNB(), RandomForestClassifier(), BaggingClassifier(), GradientBoostingClassifier(), XGboostclassifier. Output the accuracy box plot as we have seen in the class (make sure to change regressmod df to classmod. And use an appropriate metric for classification evaluation, for example, accuracy, precision,recall etc). Remember to use the object oriented approach and develop a function (def…), this will be very helpful for the next assignment. (40)

# Select the best classifier for level o classifier. Use logistic regression as a second level classifier. Similar to 5 generate the box plot and show the accuracy of each algorithm as well as stacked classifier. Also show the confusion metrices of the above algorithms .(15)

# Export the Pickle model and import it back. Use the imported model to predic the y\_test from x\_test and report the confusion metrix (3)

# Show both text and visual confusion Matrices using scikit learn and matplotlib and explain what the graph tells you and what you did. (2)