# Lab 5 - Decision Trees and Random Forests

*Lab 5 on Decision trees and random forest classification for DS3010 - Machine Learning*

## OVERVIEW & PURPOSE

In this lab, you will experiment with Decision Trees and Random Forests

## Instructions

1. Please submit the assignment through Moodle in .ipynb format (python notebook)
2. The submission should contain a single notebook containing all the solutions, including the requested documentation, observations, and findings.
3. The naming convention for the notebook is <firstname>_<lastname>_<rollnumber>.ipynb
4. You must adequately comment on the code to improve its readability.
5. The lab is worth 5 points
6. This graded lab is due on October 6th at 11.59 pm

## Lab

**1. Data Generation (0.5 mark)**

a. Generate a random 3-class classification problem using make_classification method from sklearn.datasets with 1000 samples and 5 features in which 3 among them are informative and there should not be any redundant and repeated columns
b. Convert the features and labels into data frames named features_df and labels_df respectively
c. Split the data into train and test sets with 30% is of test data

**2. Multi-class Classification**

## Decision tree Classifier (0.75 mark)

a. Create an instance of a decision tree classifier and fit the model.
b. Predict the labels for both train and test data.
c. Store the predictions of test data in a new CSV file named ‘test_predictions’ with a column name ‘DT_test_predicted’
d. Print accuracy, confusion_matrix for both train and test data
e. Print precision, recall, f1-score for each class with the help of ‘average’ parameter for both train and test data.

## Random Forest Classifier (0.75 mark)

a. Create an instance of a random forest classifier and fit the model.
b. Predict the labels for both train and test data.
c. Store the predictions of test data in the above CSV file with a column name ‘RF_test_predicted’
d. Print accuracy, confusion_matrix for both train and test data
e. Print precision, recall, f1-score for each class with the help of ‘average’ parameter for both train and test data.

**3. Observations (1 mark)**

a. Refer to the documentation of decision tree and use methods to print the ‘depth of the tree’, ‘number of leaves’ for the above learned decision tree
b. Write your observations on the predictions of the above models

**4. Hyper-Parameter Tuning**

## Decision tree classifier (1 mark)

a. Refer to the documentation and use GridSearchCV method to tune the parameters.
b. Define a param_grid dictionary with the list of permissible values of your choice for the hyper-parameters "criterion”, "splitter", "max_depth", "min_samples_split", "min_samples_leaf", "max_leaf_nodes", "max _features"

c. Print the best parameters
d. Now train the classifier with best parameters.
e. Predict the labels for both train and test data.
f. Store the predictions of test data in the above CSV file with a column name ‘Tuned_DF_test_predicted’
g. Print accuracy, confusion_matrix for both train and test data
h. Print precision, recall, f1-score for each class with the help of ‘average’ parameter for both train and test data.

## Random Forest Classifier (1 mark)

a. Follow the same steps as above to tune the hyper parameters for Random Forest but use RandomisedSearchCV method with hyper parameters “n_estimators”, “max_features”, “max_depth”, “min_samples_split”, “min_samples_leaf”, “bootstrap” and Store the predictions in the above csv file with column name ‘Tuned_RF_test_predicted’
b. Download the final csv file “test_predictions.csv” and submit it along with the .ipynb file.