In this assignment, students will implement the decision tree classifier and AdaBoost algorithms using the Titanic data set provided by Kaggle. The dataset can be downloaded using the provided link here: [Dataset link](https://www.kaggle.com/c/titanic/data)

1. **[10 pts] Data Preparation**: Since the Kaggle does not provide the true label in the test dataset, we will only use the training set.
   1. [15] Do EDA and pre-processing (do feature extraction and engineering if needed). Do not use PCA to reduce the dimension. Explain the workflow.
   2. [5] Split the provided training set into new training and test sets with a ratio of 20%. Hereafter, the new training set will be called the training set, and the new test set will be the test set.
2. **[30 pts] Decision Tree Implementation using NumPy**:

Build a tree on the training data and evaluate the classification performance. The majority of implementation codes are provided in the lecture notebook file. Use them.

1. [20] Try the Gini index and Information Gain as the attribute selection in building the decision tree and compare their results on the test set.
2. [5] Report your best accuracy on the test data set.
3. [5] Give a brief description of your observations.
4. **[40 pts] AdaBoost Implementation using NumPy:**

You can use package/tools to implement your **decision tree classifiers only**. You are **not allowed** to use AdaBoost directly from sklearn. The fit function of DecisionTreeClassifier in sklearn has a parameter: sample weight, which you can use to weigh training examples differently during various rounds of AdaBoost.

1. [25] Implement the AdaBoost algorithm, AdaBoost(X, y, M)
   * 1. Write a method that calculates error and alpha, and updates weights.
        1. def compute\_error (y, y\_pred, w\_i)
        2. def compute\_alpha(error)
        3. def update\_weights(w\_i, alpha, y\_pred)
     2. For each weak classifier, h, fit the train data using DecisionTreeClassifier.
        1. h=DecisionTreeClassifier(max\_depth=1)
        2. h.fit(X, y, sample\_weight=weight)
     3. Write a prediction algorithm that classifies the label.
2. [5] Plot the train and test errors as a function of the number of rounds (M) from 1 through 500.
3. [5] Report your best accuracy on the test data set.
4. [5] Give a brief description of your observations.