Homework 4: Decision Trees **CSCE 633**
Due: 11:59pm on March 8, 2024

Instructions for homework submission
a) Please write your code in a Jupyter Notebook. Please explain your thought process, results, and observations in markdown cells. Please do not just include your code without justification.
b) Your submission should be a Jupyter Notebook that can be run seamlessly and performs all the required steps one after another. Any submission with a runtime error would result in lost points.
c) **You can not use available ML libraries (like scikit-learn) for building trees in this homework.** Allowed libraries include NumPy, Pandas, and Matplotlib.
d) Please start early :)
e) Total: 100 points

Part A - Data Pre-processing (20 points)

Our dataset is Loan Dataset. Since labels in column *Loan_Status* only include *Y* and *N*, you can try to use your trees as binary classifier.

1. Read Loan Dataset into a *pandas* dataframe.
2. Explore the data. How does the data look like? What's the shape of the data? Drop all the rows with any missing values.
3. Extract the features and the label from the data, assuming the label is *Loan_Status* and the others are considered features. Then do one-hot encoding for categorical features: (1) separate numerical columns from nonnumerical columns; (2) use *get_dummies* or *replace* for transforming to categorical; (3) concatenate both parts. Transform the output into numerical format. Examples may include transforming 'N' in label to 0 and 'Y' to 1; replacing '3+' by 3; changing 'Urban' to 0, 'Rural' to 1 and 'Semiurban' to 2, etc.
4. Split the data into three sets - train, validation, and test sets. You should have the following six splits: X_train, X_val, X_test and y_train, y_val, y_test.

Part B - Decision Tree Implementation (80 points)

In this problem, you will be coding up a regression tree and a classification tree from scratch. Trees are a special class of graphs with only directed edges without any cycles. They fall under the category of directed acyclic graphs or DAGs. So, trees are DAGs where each child node has only one parent node.

Since trees are easy to design **recursively**, it is super important that you are familiar with recursion. So, it is highly recommended that you brush up on recursion and tree-based search algorithms such as depth-first search (DFS) and breadth-first search (BFS).

Your submission should include a script that can be run seamlessly and performs all the following steps one after another. Submission with a runtime error would result in lost points.

1. Using the data you pre-processed above, implement both a maximum-depth regression tree and a two-class classification tree from scratch for prediction. You are NOT allowed to use machine learning libraries like scikit-learn here.

2. Train the models using training data, and use validation data to validate the trained models. What are the performances (accuracies) of the two trees on data_val?

3. Using the trained models, conduct inference on the test data and save the predicted results in separate files called Test_Result_1/2.csv.

4. Below are suggested steps you may want to consider.

   Define a splitting criteria: 1) this criteria assigns a score to a split; 2) this criteria might be MSE for regression tree or the Gini Index for classification tree.

   Create the split: 1) split the dataset by iterating over all the rows and feature columns; 2) evaluate all the splits using the splitting criteria; 3) choose the best split.

   Build the tree: 1) decide when to stop growing (when the tree reaches the maximum allowed depth or when a leaf is empty or has only 1 element); 2) split recursively by calling the same splitting function; 3) create a root node and apply recursive splitting.

   Predict with the tree: For a given data point, make a prediction using the tree.