CS 3753 & 5163 Data Science
Homework 5 (100 + 20 points)

Submission:

1. The instruction is the same as the ones previous assignments. Please submit a single python script (abc123_hw#.ipynb) through blackboard. All the results are outputted from your Python code.
2. For this assignment, you **are encouraged to** collaborate with up to three members in a team. You can help each other to understand the questions and discuss the issues in the assignment. However, you cannot copy solutions from your team members. You still need to submit your solution independently.

Questions

1. (20 points) We do the linear regression on three points (0.5, 1), (2, 2.5), and (3, 3). Please calculate the SSEs of the four linear regression. Which is the best linear regression using SSE? Output all steps through your python code.
   (a) y = x + 0.5, SSE = (0.5+0.5 - 1)**2 + (2+0.5 – 2.5)**2 + (3+0.5 - 3)**2 = 0.25
   (b) y = x + 1, SSE = (0.5 +1 - 1)**2 + (2+1-2.5)**2 + (3+1 - 3)**2 = 1.5.
   (c) y = 0.8*x + 0.3, SSE = (0.8*0.5+0.3 - 1)**2 + (0.8*2+0.3 – 2.5)**2 + (0.8*3+0.3 - 3)**2 = 0.54
   (d) y = 0.8*x + 0.7, SSE = (0.8*0.5 + 0.7 - 1)**2 + (0.8*2 + 0.7 – 2.5)**2 + (0.8*3 +0.7 - 3)**2 = 0.06

   (d) is the best since the SSE is the smallest.

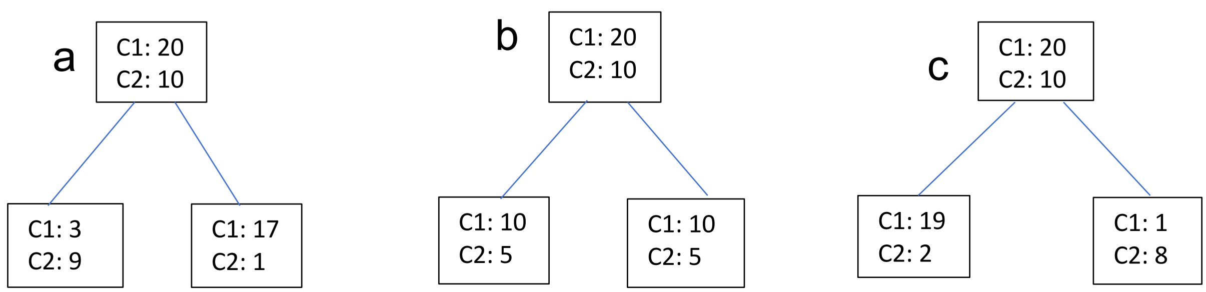
2. (20 points) What is the problem solved by Lasso and Ridge regression? What is the major difference between the two regression? Please discuss the advantages and disadvantages of them.
   Overfitting. The penalty terms are different. It is the sum of squares of all coefficients in Ridge regression and it is the sum of the absolute value of the coefficients in Lasso regression.

   Lasso regression can remove useless or irrelevant variables from regression equations. In contrast, Ridge regression does a little better when most variables are useful or relevant since it cannot remove them by the penalty term.

3. (30 pints) Decision Tree
There are various ways to decide on the metric to choose the variable on which splitting for a node is done. Different algorithms deploy different metrices to decide which variable splits the dataset best.

Let's say we have a sample of 30 records. There are two classes C1 and C2. We have three possible splits a, b, and c (see figure below). The number of records in each class is shown in every node.

a. Write a Python code to measure the node impurity using Gini Index, Entropy Gain, and Misclassification Error, respectively.
b. Evaluate the quality of the three splitting and report the best one.
c. The code should print out the quality of each splitting and your best choice. You also can print out any other information freely.
d. Finally, print out your conclusion about whether all three methods have the same best choice.



The output information of node impurities is in the following table

Impurity	a		b		c	
	left	right	left	right	left	right
Gini index	0.375	0.105	0.444	0.444	0.172	0.198
Entropy	0.811	0.31	0.918	0.918	0.454	0.503
Misclass error	0.25	0.056	0.333	0.333	0.095	0.111

The output qualities of the splitting are in the following table

Splitting	a	b	c
Gini index	0.213	0.444	0.18
Entropy	0.408	0	0.45
Misclass error	0.2	0	0.233

All three methods have the same best choice c.

4. (30 points) KNN: this section applies the KNN algorithm to the Iris flowers dataset. The first step is to load the dataset in "iris.csv" and convert the loaded data to numbers that we can use with the mean and standard deviation calculations. For this we will use the helper function load_csv() to load the file, str_column_to_float() to convert string numbers to floats and str_column_to_int() to convert the class column to integer values.

You do not need to import any libraries or modules about KNN because you will implement the KNN from scratch. The template of the code is provided and you just need to complete the functions Euclidean_distance(), get_neighbors(), and predict_classification(). The mean accuracy is around 97% (96.667%).

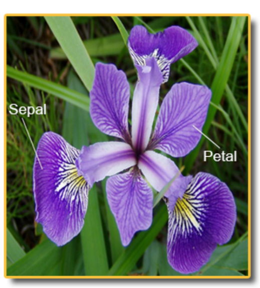
We will evaluate the algorithm using k-fold cross-validation with 5 folds. This means that 150/5=30 records will be in each fold. We will use the helper functions evaluate_algorithm() to evaluate the algorithm with cross-validation and accuracy_metric() to calculate the accuracy of predictions.

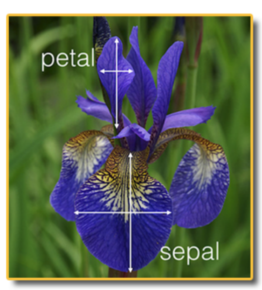
A new function named k_nearest_neighbors() was developed to manage the application of the KNN algorithm, first learning the statistics from a training dataset and using them to make predictions for a test dataset.

Download the dataset and save it into your current working directory with the filename "iris.csv". The Iris Flower Dataset involves predicting the flower species given measurements of iris flowers.

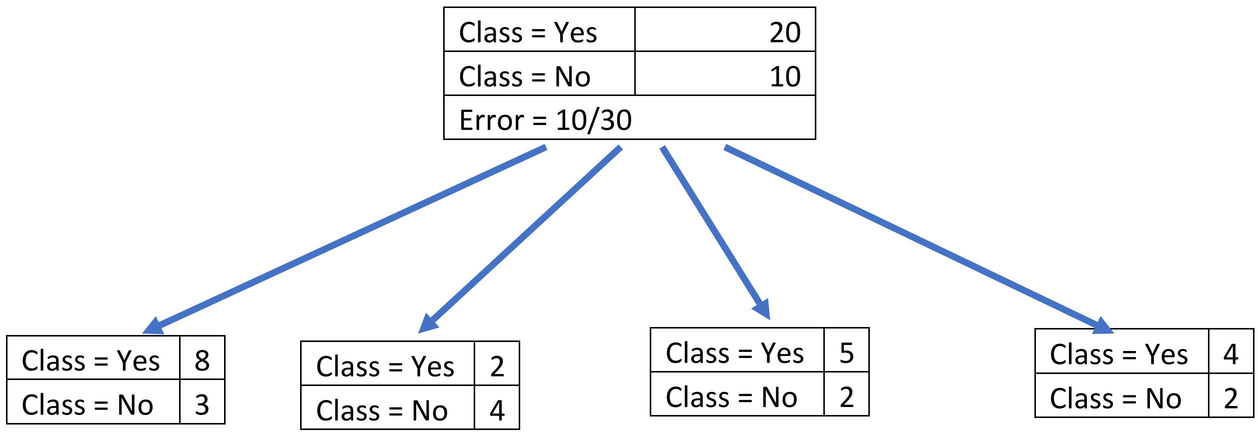
It is a multiclass classification problem. The number of observations for each class is balanced. There are 150 observations with 4 input variables and 1 output variable. The variable names are as follows:

a. Sepal length in cm.
b. Sepal width in cm.
c. Petal length in cm.
d. Petal width in cm.
e. Class





5. (Extra credits: 20 pts) Post pruning of decision tree by pessimistic approach. In this approach, the error in a leaf node is $e'(t) = e(t) + 0.5$, where $e(t)$ is the training error in a node. Please calculate the pessimistic error at the parent node A and leaf nodes. Should this tree be pruned?



Training Error (Before splitting) = 10/30 = 0.3333
Pessimistic Error (Before splitting) = (10+0.5)/30 = 10.5/30 = 0.35

Training Error (After splitting) = (3+2+2+2)/30 = 0.3
Pessimistic Error (After splitting) = (3+2+2+2+4*0.5)/30=0.3667

Because the pessimistic error after splitting is higher than the error before splitting, we should prune the tree.