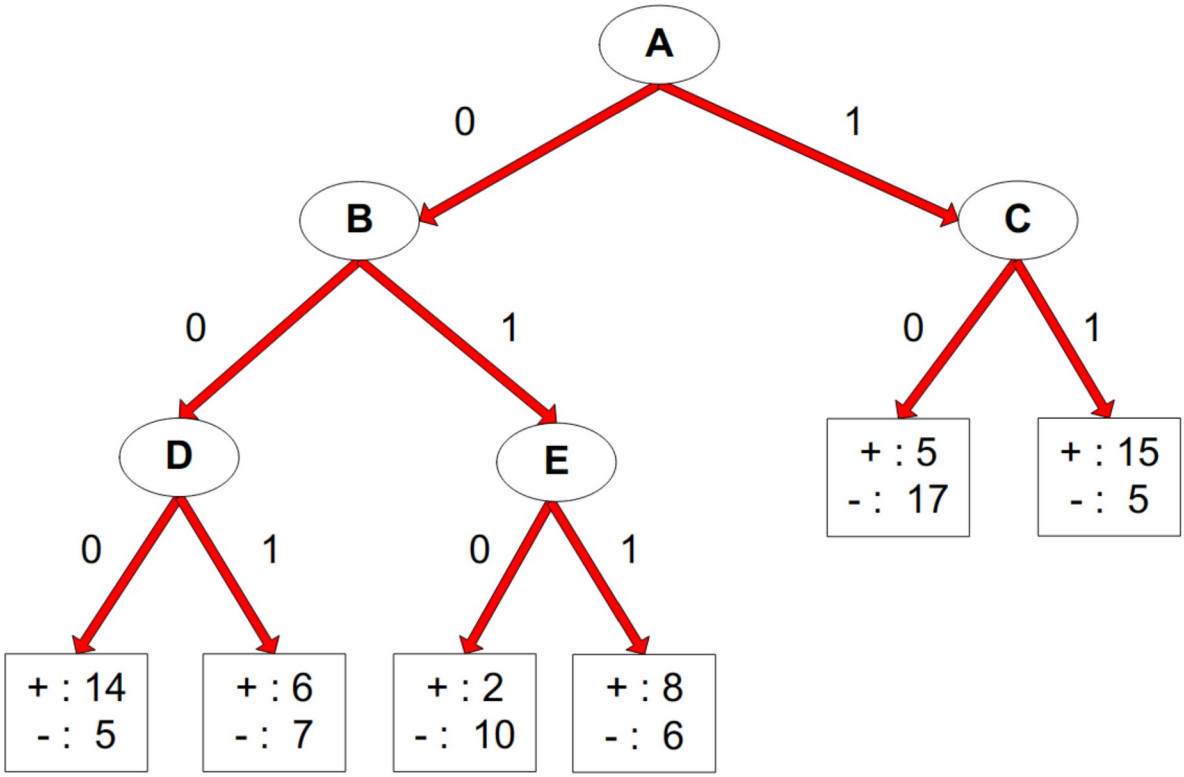
Homework 02
Name: Lam Nguyen
Student ID: la815794
Due 18Feb2024

Problem 1 (Written Question 10 points). Consider the decision tree shown in the diagram below. The counts shown in the leaf nodes correspond to the number of training records associated with the nodes.



(1) What is the training error rate for the tree? Explain how you get the answer.

Classification error for single node: Error(t) = 1-max[p(t)]
Classification error for each node:

Error(D_0) = 1-14/19 = .263
Error(D_1) = 1-7/13 = .46
Error(E_0) = 1-10/12 = .167
Error(E_1) = 1-8/14 = .429
Error(C_0) = 1-17/22 = .23
Error(C_1) = 1-15/20 = .25

Classification Error for entire tree is the weighted average of the errors for each leaf node.

Classification Error Rate = .2(.263)+.13(.46)+.12(.167)+.14(.429)+.22(.23)+.2(.25)
Classification Error = .293 = 29.3%

(2) Given a test instance T={A=0, B=1, C=1, D=1, E=0}, what class would the decision tree above be assigned to T? Explain how you get the answer.

The decision tree would be assigned to class E_0. A= 0 leads to node B=1 leads to node E=0 leads to the node E_0.

Problem 2 (Written Question 15 points).
(1) Are decision trees a linear classifier?

No. It is a non-linear classifier

(2) What are the weaknesses of decision trees?
First, interacting tributes that can distinguish things together but not individually, may be passed over in favor of attributes that are less selective. Next, each decision boundary involves only one attribute, when in reality, sometimes a boundary requires multiple attributes to form a decision.

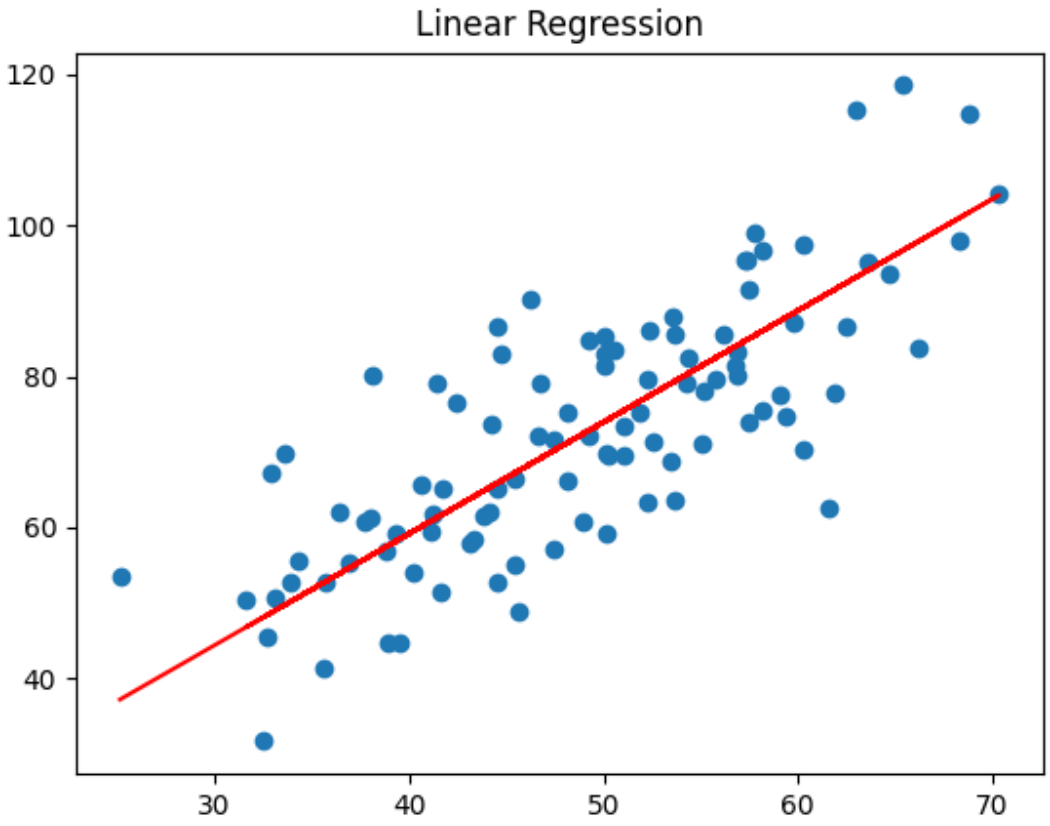
(3) Is Misclassification errors better than Gini index as the splitting criteria for decision trees?

No. Gini Index improves as splits occur while misclassification error remains the same.

Note that: For Coding Questions, please **do not** directly call linear regression and non-linear regression built-in functions in existing library packages such as scikit-learn. You may call basic computation functions built in Numpy.

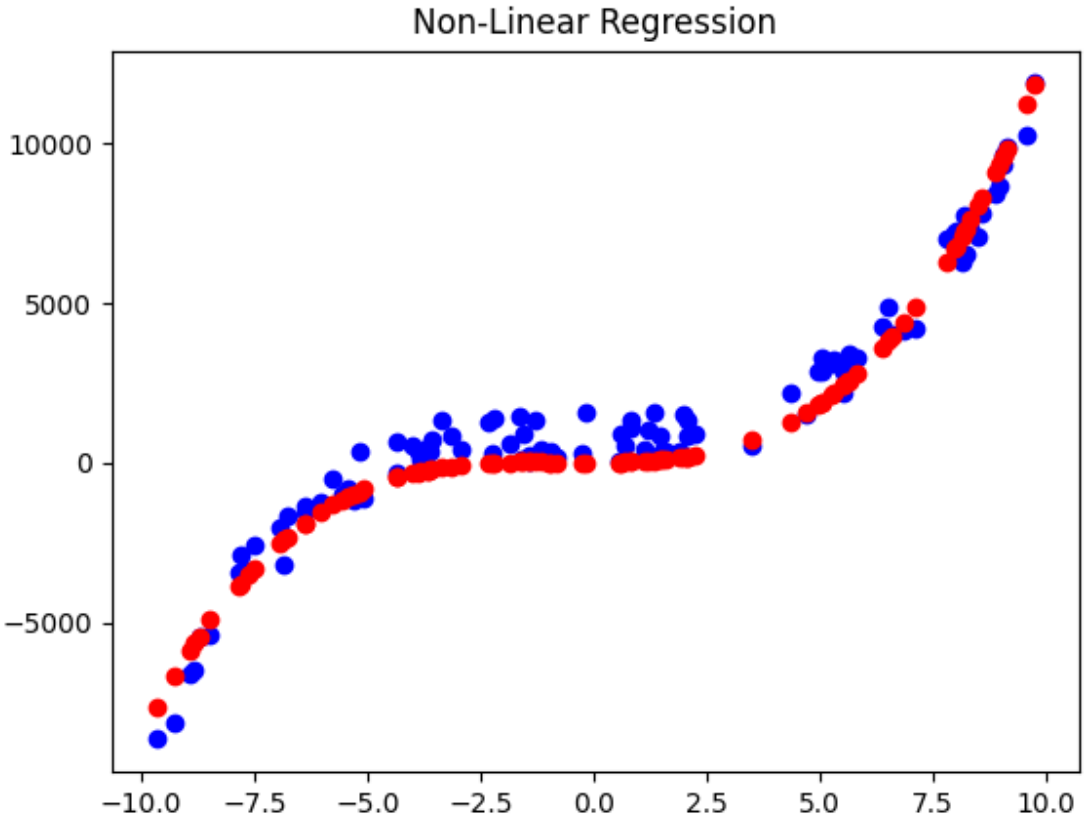
Problem 3 (Coding Question 25 points)
Please implement a Linear Regression to find the best linear model for the provided HW2_linear_data.csv. Please plot the result using “matplotlib.pyplot”.
Note that

(1) The linear model is in the following format Y=mX+c
(2) Use MSE as the loss function
(3) You may use “pandas” to read the csv file and load the values into two vectors X and Y.
(4) Use Gradient Descent for the training. You may choose fixed learning rate (such as 0.0001) and epochs (such as 1000) without considering mini-batch.
(5) The result will look like the following image.



Problem 4 (Coding Question 25 points):
Please implement a non-linear regression to find the best cubic function model for the provided HW2_nonlinear_data.csv. Please plot the result, too.

(1) The cubic function is in the following format: Y=aX^3+bX^2+cX+d
(2) Use MSE as the loss function.
(3) Use Gradient Descent for the training. You may choose fixed learning rate (such as 0.000001 (1e-6)) and epochs (such as 10000) without considering mini-batch. It may take 10-15 seconds to finish the running for 10000 steps. Please be patient.
(4) The result will look like the following


Non-Linear Regression
10000
5000
0
−5000
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0