**CSCE 623 Spring 2020 - Machine Learning. In Class Work, Day 15**

From Chapter 8: Tree-Based Methods

For the purposes of this exercise, you may assume all dataframes have 2 real number features (‘X1’,’X2’) and a real number label (‘Y’). The dataframes will have *n* observations.

Visualization plots will be used to present the data and the feature sets. In 2D feature-presentation scatterplots, the acutal datapoint value is not shown and the horizontal axis will represent the X1 feature and the vertical axis will represent the X2 feature. 3D plots will show the features and the data: X1 on the “x” axis, X2 on the “y” axis and the value of the observation label value on the “z” axis

In this exercise, you will write code to determine how to split a region (represented by a dataframe) to achieve the best possible combined RSS from the resulting 2 dataframes.

1. Write a function to compute the RSS for a dataframe, assuming every observation was predicted to have the mean of the ‘Y’ value for the entire dataframe. Try to do this with a matrix multiply instead of a for loop.

def computeRSSonMean(df):

…

return(RSS)

2. Split a dataframe by feature split value: Create a function to compute two dataframes which partition the original dataframe lower and higher than the split value of the split feature. Return the two resulting dataframes

def divideFrame(df,splitFeature,splitVal):

…

return(dfLow,dfHigh)

3: Write a function to determine the feature and value to split a dataframe which minimizes the combined RSS of all of the resulting subframes. For building a regression tree, one step is to write a function which makes the split decision for a region. Given a region with a dataframe df (containing *n* observations and 2 features ‘X1’ and ‘X2’ and real output values (column ‘Y’)), write code to determine & return the best feature and threshold split value for that predictor. Your best split feature and value is the one which minimizes the overall RSS of the two resulting regions. Your code should return the best feature to split on, the value to split that feature at, and the new RSS for the subregions prediction the mean of the ‘Y’ values in each of those regions. There is no need to try to make this as fast as possible, and it is ok to call the previous functions you wrote in parts 1 and 2 above. You can use a for loop – can you accomplish this task by checking only 2 x n possible splits?

def pickSplit(df):

…

return(splitFeature,splitValue,bestRSS)

Note: When working with the shell code provided, after you have implemented your code portions for the 3 steps above, set the flag finishedCode=True This will allow the instructor code to run which tests your decision system on a 1st and second split

Assuming you don’t change the seed on the dataset generator from “42” you will get these outcomes:

Best 1st split: X1 1.9981694881817487 , RSS = 16346.739993900164

Best 2nd Split: X2 3.4962847016552057 , RSS = 4799.969513184029

