Q1

Dataset description

The dataset contains 1030 instances. A total of 9 attribute breakdowns, 8 quantitative input variables, and 1 quantitative output variable. The missing attribute values are None.

Functions definitions and their usages:

1. **test_train_split(db):**
   This functions splits dataset randomly with ratio of 70:30 into train and test data and data
2. **get_score(y_true,y_pred):**
   Input : Expected output and Predicted output in numpy array type
   Output : Accuracy in percentage
   Accuracy function : acc% = 100 - (100 *(|y_true-y_pred|/y_true))
3. **save_model_tree(model, filename, test):**
   Input : Trained model, filename, test dataset to get accuracy
   Tree Diagram stored in given filename directory of given trained tree model
4. **ten_random_splits():**
   Perform ten random splits and plot accuracy
   Plot PNG File : Accuracy_of_10_random_splits.png
5. **different_limit_size():**
   Perform prediction with different limit sizes and plot accuracy vs limit size graph
   Limit Size : [1,7,13,...,61]
   Plot PNG File : Accuracy_vs_limit_size.png
6. **different_max_depths():**
   Perform prediction with different max depth and plot accuracy vs max depth graph
   For depth -> [1,14]
   Plot PNG file : Accuracy_vs_max_depth.png

RegressionTree Class:

Psedo Code:

Store all necessary values like depth, mean_value etc.

Check for Leaf Node Condition, i.e, Number of datasets <= LIMIT_SIZE or depth >= MAX_DEPTH

If Leaf node, Then isLeaf = True and Return
Else,
Iterate over all attributes

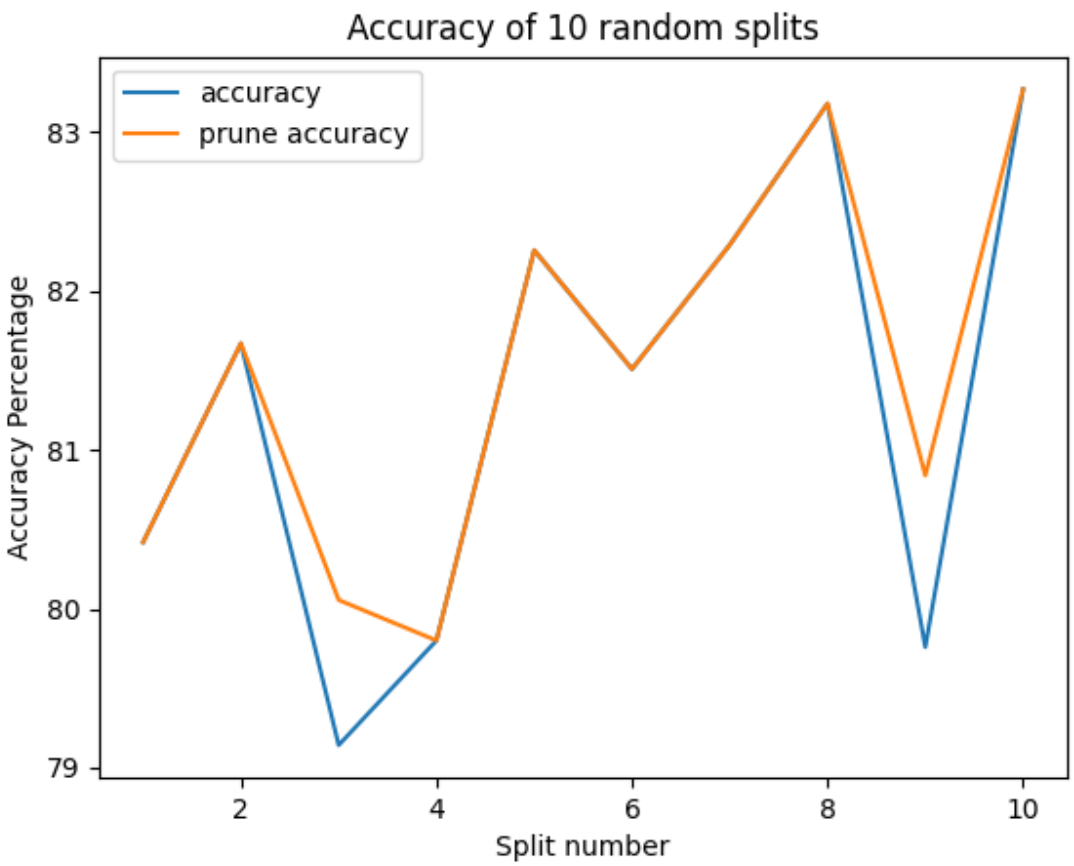
```
Sort dataset with respect to attribute

Take each data,
      Divide dataset into 2 sets
      Find mean of each set
      Calculate sum square error

      Update minimum sum square error
      Save best split
Recursively create left and right tree using best split
```

Results:

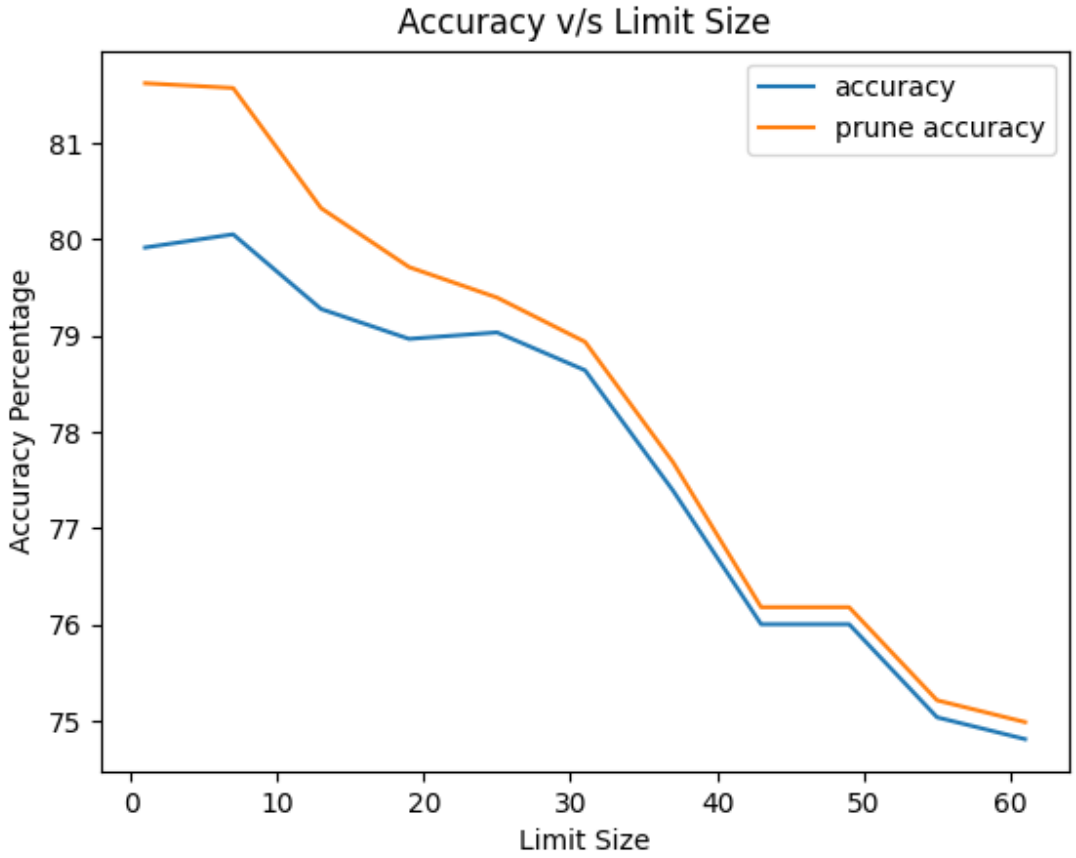
1.



Accuracy Without Pruning : [80.41947082868083, 81.67057617910446, 79.14191017745534, 79.80041389158497, 82.25760563731467, 81.50912178411565, 82.29137916604053, 83.18048251640225, 79.75891537472909, 83.27066790240411]

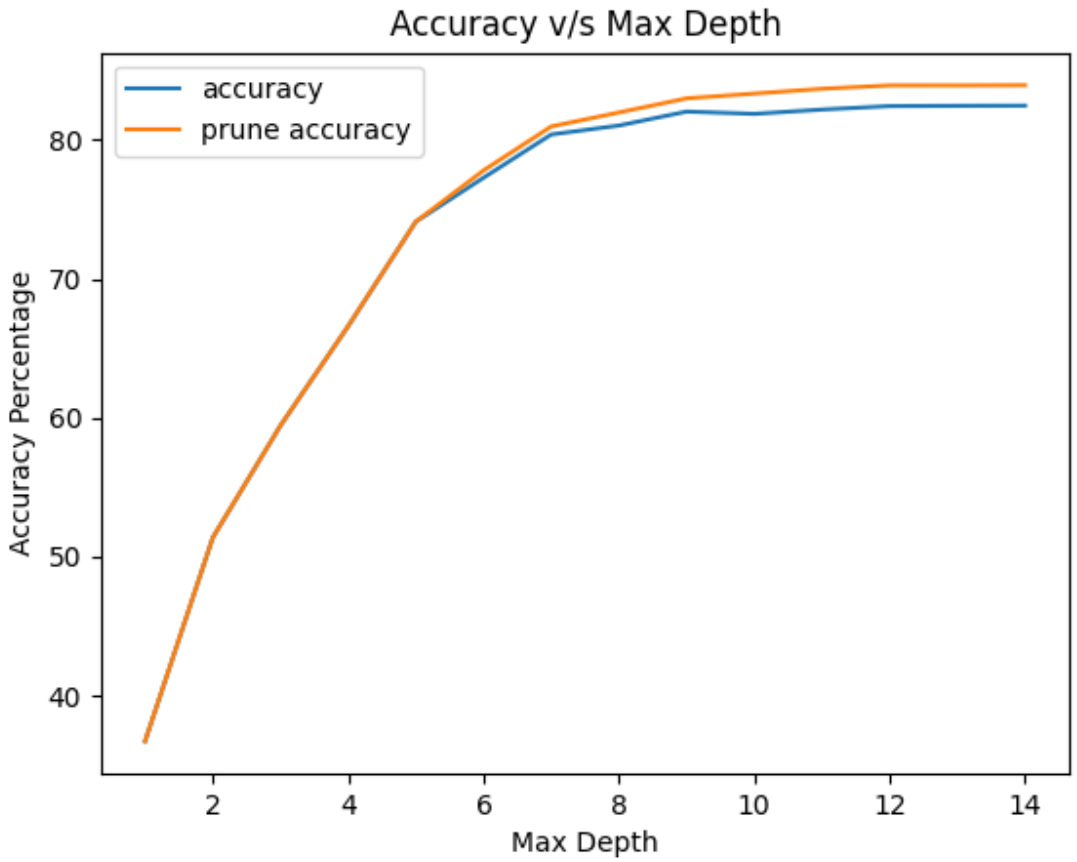
Accuracy With Pruning : [80.41947082868083, 81.67057617910446, 80.05556423678743, 79.80041389158497, 82.25760563731467, 81.50912178411565, 82.29137916604053, 83.18048251640225, 80.83905315296468, 83.27066790240411]

2.



Limit Size : [1, 7, 13, 19, 25, 31, 37, 43, 49, 55, 61]
Accuracy Without Pruning : [79.91670967877134, 80.05358469389681, 79.27836248448682, 78.96831829304392, 79.03550698435134, 78.64197856301323, 77.39217135668041, 76.0062564135433, 76.0062564135433, 75.03971054184564, 74.81348311504472]
Accuracy With Pruning : [81.62207257795521, 81.57278972395883, 80.32547113962305, 79.71334892059555, 79.39683721558809, 78.93716198747369, 77.68735478114087, 76.18294228660704, 76.18294228660704, 75.21639641490938, 74.99016898810848]

3.



Therefore, for depth = 10, model overfits

Max Depth : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
Accuracy Without Pruning : [36.721347480599945, 51.3728842762679, 59.437572728326586, 66.59620081731182, 74.12122998487285, 77.27513328370239, 80.39063075779758, 81.0346427622943, 82.03875365961981, 81.87400005623167, 82.184176230982, 82.43452962016391, 82.4545024809265, 82.46633509844776]
Accuracy With Pruning : [36.721347480599945, 51.3728842762679, 59.437572728326586, 66.59620081731182, 74.12122998487285, 77.80269032984557, 80.97033130487932, 81.97348888004646, 82.99102876013613, 83.33621378087446, 83.68043491294475, 83.92429901707231, 83.92187424199811, 83.93568514889816]