HW06_720_DecTrees_2191

by Aseem Mehta

1. Code Structure:

   Code HW06_Mehta_Aseem is divided into 3 parts

   I. Read datasets using pandas into python
   II. Creating Decision Tree Classifier

      Important parts in creation of decision tree classifier

   a. Find count of classes
   b. Checking if stopping criteria is met

      3 Stopping criteria are used in the code

      1. Stop if a class count is more than 90%
      2. Stop if a class has less than 23 instances in the dataset
      3. Stop if depth of decision tree has reached 10
   c. If a stop criteria is met tree structure is updated with the return class value, a label as Leaf Node(None) and the depth of the tree in with root being labeled as 10 going towards 0 (check section f for tree structure works specification)
   d. Finding the correct attribute to split and the value to split on

      Split criteria are chosen based on ranges, so if an attribute is between 0.1 to 9.9, all values between 0 to 10 are considered with an interval of 1.

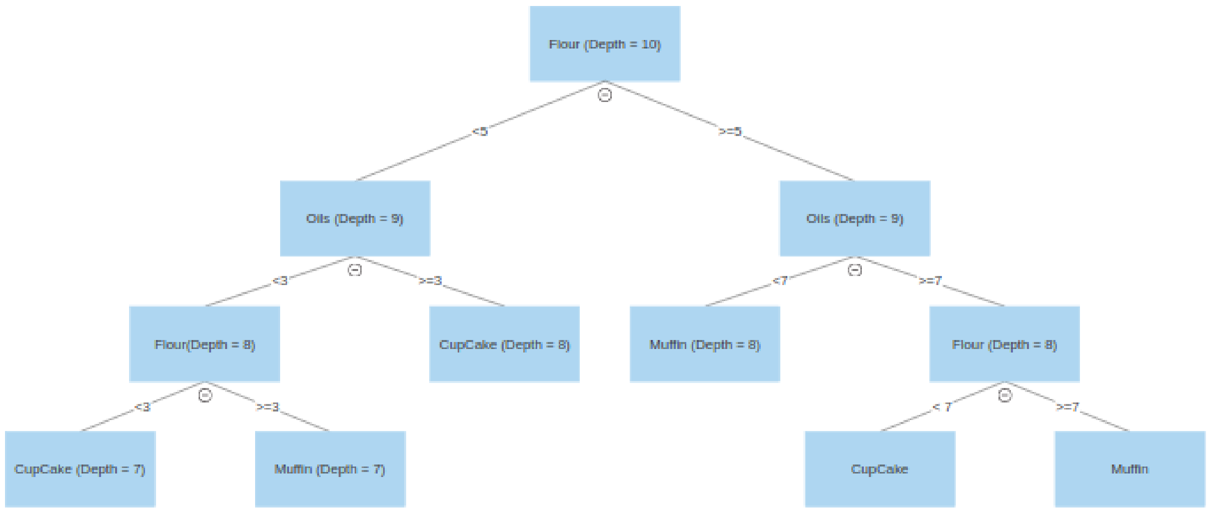
      Weighted entropy is calculated for every split criteria of an attribute and split criteria corresponding to the minimum weighted entropy is chosen.

      This activity is performed for all attributes of the dataset the attribute and the split criteria corresponding to the minimum weighted entropy is chosen.
   e. If a split criteria is found the Attribute name, split value and the depth of the tree in with root being labeled as 10 going towards 0 (check section f for tree structure works specification)
   f. Tree Structure defines the tree order traversal and values in preorder format. TreeStructure consists of a list of nodes, these nodes define every node of the program's classifier tree.

      Structure of node

      e.g. 1 ['Flour', 5, 10], Flour is the Attribute to check, 5 is the splitting criteria and 10 is the tree level in desecnding order, starting at 10 (see figure modify in writeup)

      e.g. 2 ['CupCake', None, 7], None specifies it is a leaf node, CupCake is the return class and 7 is the level of the tree.



Final Classifier figure

III. Writing a program to use the classifier on the testing dataset

Code HW06_Mehta_Aseem_Classifier

Previous code generates this code and is divided in 3 parts

I. Read datasets using pandas into python
II. Using Tree structure predict values on the test dataset
III. Write the predictions in a csv file

2. Accuracy

Classifier provided an accuracy of 93% on when ran on the training dataset.

3. Challenges

a. Instead of putting multiple nested if-else in the classifier code (HW06_Mehta_Aseem_Classifier.py), I decided to create a tree structure consisting of list of nodes. This part was very difficult to visualize and addressing the correct node was difficult. Initially I had not used depth as a factor to traverse through the tree structure, once this was added the program became simpler to traverse and code.
b. Making sure all edge cases were covered inside the program.

4. Discussion

Number 23 is a is a prime number, because of which you cannot get a tie between 2 classes. Multiple studies have found number 23 as very significant in decision making. In US Grand Jury also consist of 23 people because of this reason. Multiple mathematicians and historians have studied this number.

5. Conclusion and Results

The program provided an Accuracy of 93% on the training dataset.

e.g. treeStructure = [['Flour', 5, 10], ['Oils', 3, 9], ['Flour', 3, 8], ['CupCake', None, 7], ['Muffin', None, 7], ['CupCake', None, 8], ['Oils', 7, 9], ['Muffin', None, 8], ['Flour', 7, 8], ['CupCake', None, 7], ['Muffin', None, 7]]

Tree structure is the classifier and is written in second code. Benefit of using this structure is even if the tree is very large, second code would not take a huge amount of space in nested if-else statements. The code size will remain the similar, only changing factor would be treeStructure.

As the classifier was a decision tree once traversal was sorted it was very simple to go through the data to see if the results were accurate as per the structure.

Another approach would be to use a doubly Linked list for storage and traversal purposes.