Decision Tree Report

Decision Tree

Decision Tree(DT) is one of the classification methods. In decision analysis, a DT can be used to visually and explicitly represent decisions and decision making. DT has a flow-chart like structure (tree-like model) in which each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label.

In this method all the features are considered and different split points are tried and tested using a cost function. The best cost (or lowest) would be selected for the split.

There are three types of cost function for Decision Tree. They are Gini Index, Entropy and Misclassification Error. Through these functions, we can measure the impurity of the node/attribute. The attribute with the lowest impurity would be selected as a split. After selecting such node, the same procedure would be repeated for the remaining attributes. This procedure would be repeated until we are able to reach all the leaf nodes (class label) in the tree.

The topmost node in a tree is the root node. A path is traced from the root to a leaf node, which holds the class prediction for that tuple.

Data Set

Data Set which is used for this task is wine data set which is imported from 'sklearn.datasets'. The data in this Data Set is the results of a chemical analysis of wines grown in the same region in Italy by three different cultivators. The data set determines the quantities of 13 constituents found in each of the three types of wines. This data set has a categorical label for each wine along with thirteen continuous-valued features. There are three different wine 'categories' and our goal will be to classify an unlabeled wine according to its characteristic features such as alcohol content, flavor, hue etc.

Data Pre-Processing

As per my understanding, there is no need to perform data pre-processing for Decision Tree(DT) method as DT finds the split from the data based on the number of output classes assigned to particular attribute in training data. It has no dependency on the values of the attribute as it is only associated with the count of the classes.

Code

This assignment has been done in the below sequence of code.

- Load the data set from the sklearn data set.
- Create dataframe object from the data set object bunch using inbuilt pandas function.
- Split the train and test data using “train_test_split” function of “sklearn.model_selection”.
- Create Decision Tree Classifier Object using “DecisionTreeClassifier” method of “sklearn.tree”. Give the different test functions like ‘Gini’ and ‘Entropy’ as an argument of this method to get different decision trees.
- Print the graph of the above output using “export_graphviz” method of “graphviz” library.

Results Interpretation

Below are the two matrices derived using ‘Gini’ and ‘Entropy’ method in DT classifier.

*Gini:

```
[0.65608741 0.         0.59070295 0.11072664 0.         0.
 0.2688     0.         0.        ]
```

*Entropy:

```
[1.56169517 0.99620888 0.         0.32275696 0.         0.
 0.78894066 0.37123233 0.         0.         0.        ]
```

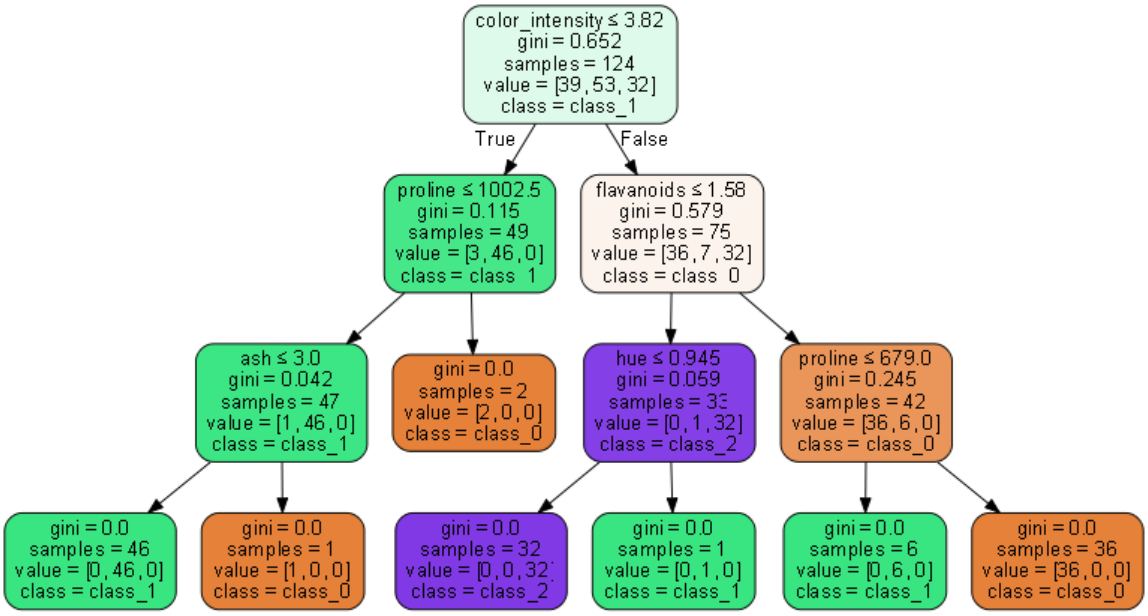
	Gini	**Entropy**
Accuracy	94.44 %	92.59 %

[*Note: Values in the above matrices are impurity of nodes in the tree. Going from top to bottom in the tree, the values in the matrices represent an impurity from each node traversing from left to right at a particular branch].

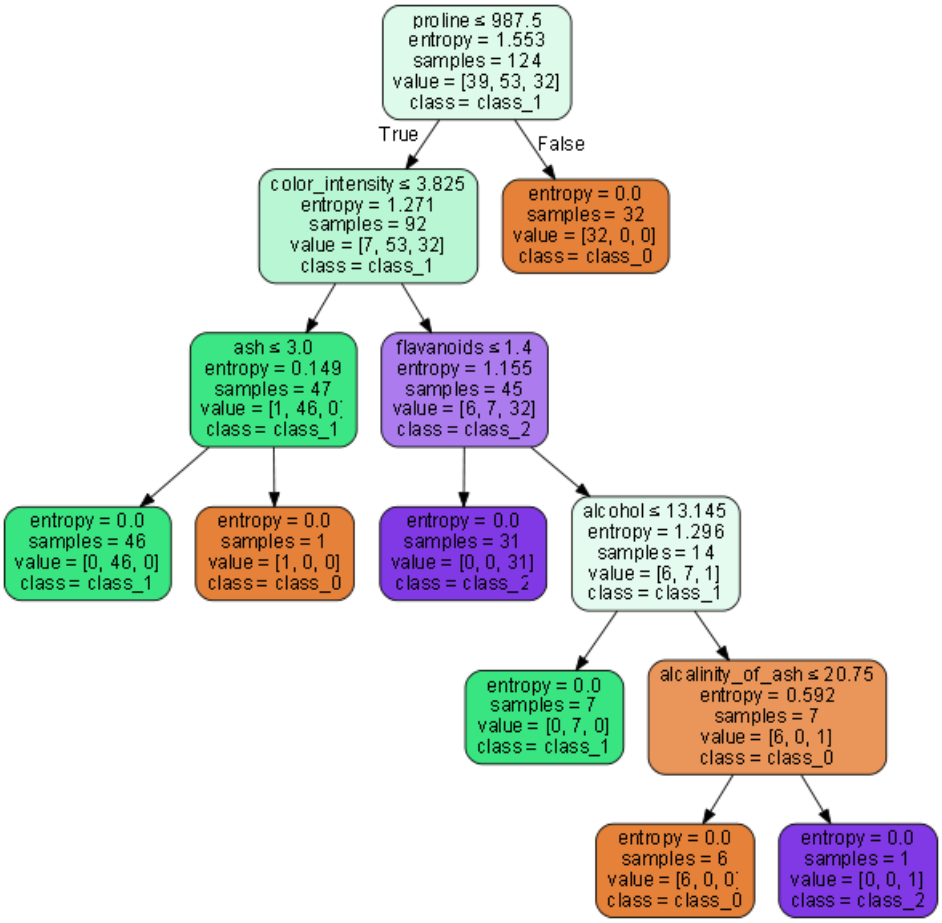
In the DT Classifier method, the impurity would be find using above methods and the node with the lowest impurity would be selected as split.

For splitting the train and test data in the assignment, ‘train_test_split’ is used from “sklearn.model_selection”. This function splits the data every time in the random order and hence the output of the code would be different every time because of this function. The accuracy of Gini and Entropy would be either be same or might sometimes differ from each other. As a general “Gini will tend to find the largest class, and entropy tends to find groups of classes that make up ~50% of the data”.

Decision Tree Using Gini



Decision Tree using Entropy



Extra Imported Packages

1. **Graphviz:** This package facilitates the creation and rendering of graph descriptions in the DOT language of the Graphviz graph drawing software (master repo) from Python.

   Commands:

   conda install -c anaconda Graphviz

   pip install graphviz

2. **PyDotPlus:** PyDotPlus is an improved version of the old pydot project that provides a Python Interface to Graphviz's Dot language.

   Commands:

   conda install -c conda-forge pydotplus

   pip install pydotplus

References

i. https://en.wikipedia.org/wiki/Decision_tree

ii. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

iii. Book: Data Mining Concepts and Techniques by Jiawei Han, Micheline Kamber, Jian Pei

iv. http://scikit-learn.org/stable/datasets/index.html

v. https://archive.ics.uci.edu/ml/datasets/wine

vi. https://jonathonbechtel.com/blog/2018/02/06/wines/

vii. http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

viii. https://m.garysieling.com/blog/sklearn-gini-vs-entropy-criteria

ix. https://github.com/xflr6/graphviz

x. https://github.com/carlos-jenkins/pydotplus