**Decision Tree Report**

Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules[1].

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node[2].

The topmost node is the root node and to predict the class a path from the root to leaf node is traversed.

In decision tree all the features are considered and then different types of splits(ordinal/nominal/continuous) are tried using the cost function(Gini Index/Entropy/Misclassification Error). The lowest cost would be selected for the splitting process. The same process for selecting the lowest cost for each attribute is followed. We stop the process when the splitting reaches the leaf node.

Dataset

The adult dataset, hosted by The Machine Learning Group at UCI, contains census information from 1994[4]. With this data, we are tasked of predicting whether a person makes more than $50K/year. The adult dataset is a fairly large set, consisting of 48,842 instances. There are 14 attributes prescribed to each person: {income (‘>50K’ or ‘<=50K’), age, WorkClass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country}[3].

Data Preprocessing

The dataset contained ‘ ?’ which I dropped from the pandas dataframe to get the better accuracy. The dataset was reduced to 30,162 records. I converted the categorical data with the help of LabelEncoder from sklearn import preprocessing. It can also be used to transform non-numerical labels to numerical labels[5].

Implementation

* Loaded the dataset into pandas dataframe using pd.read\_csv.
* Cleaning and preprocessing the data.
* Splitting the data into training and testing using ‘train\_test\_split’ from sklearn.model\_selection.
* Creating Decision Tree Classifier object and using it metrics like ‘Gini’ and ‘Entropy’ to get decision tree accuracy score, classification report and confusion matrix.
* Visualizing the decision tree using ‘graphviz’ library and finally exporting to .PNG file

Results

From my analysis, I got the following accuracy score of ‘Gini’ and ‘Entropy’.

|  |  |  |
| --- | --- | --- |
|  | Gini | Entropy |
| Accuracy Score | 80.41% | 81.03% |

Conclusion

As per my results, Entropy is better than Gini Index. In general, if I run both cost function the accuracy score for both would be same or bit different. This happens due to splitting the data every time after lowest cost is selected until they reach leaf node.

Other Packages

1. Graphviz: It consists of a graph description language named the DOT language[6] and a set of tools that can generate and/or process DOT files[8].

Commands:

* conda install -c anaconda Graphviz
* pip install graphviz

1. PyDotPlus: It is an improved version of the old pydot project that provides a Python Interface to Graphviz’s Dot language[7].

Commands:

* conda install -c conda-forge pydotplus
* pip install pydotplus

Note: Please set environment variables for graphviz before using it.

References

[1] <https://en.wikipedia.org/wiki/Decision_tree>

[2] <https://www.geeksforgeeks.org/decision-tree/>

[3] <http://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/024.pdf>

[4] <https://archive.ics.uci.edu/ml/datasets/census+income>

[5] <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#sklearn.preprocessing.LabelEncoder>

[6] <https://en.wikipedia.org/wiki/Graphviz>

[7] <https://pypi.org/project/pydotplus/>

[8] <https://github.com/xflr6/graphviz>

[9] <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>