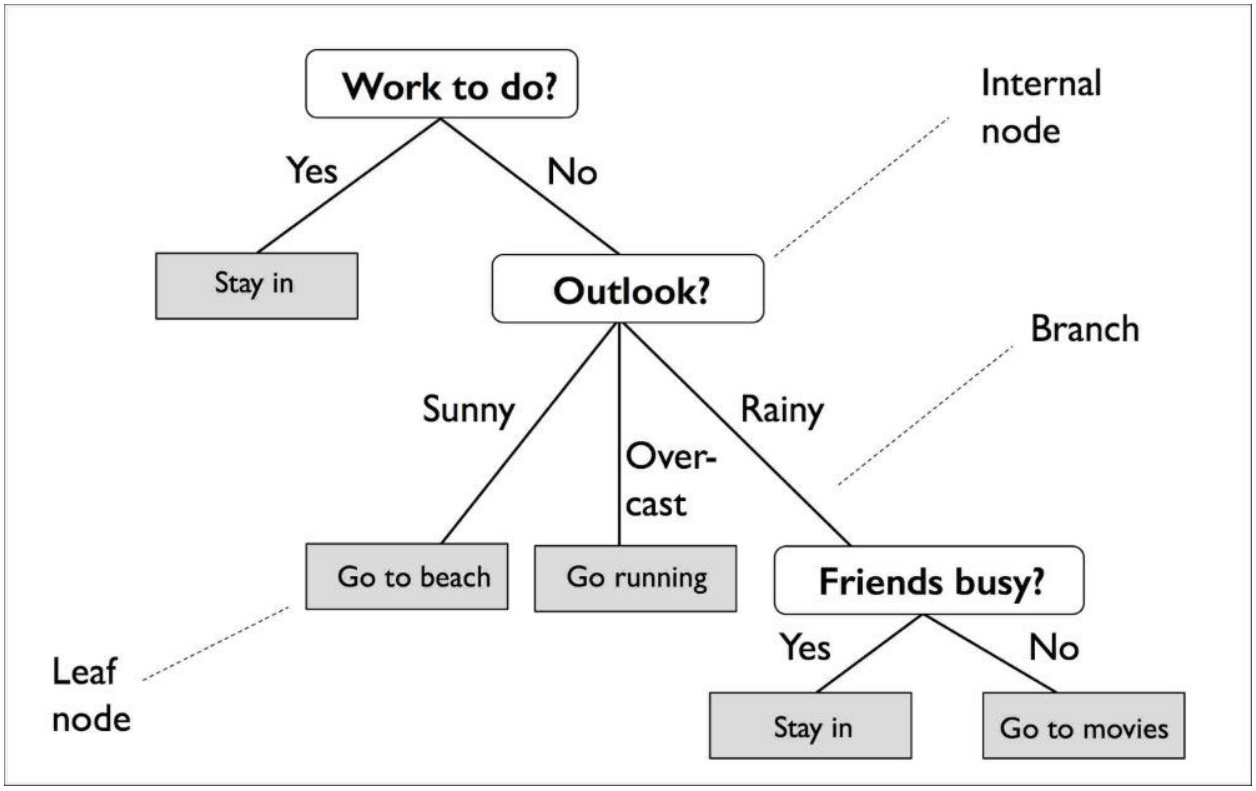
Question #2 Report

Implementation of Decision Tree Classifier



Abhinav Khanna(2017CSB1071)

CS524: Data Mining

INTRODUCTION

In this report, I will discuss the implementation and results of the decision tree classification algorithm. The purpose of this project is to implement these algorithms on

various datasets, analyze their space and time complexity, and conclude which algorithm is better for which dataset. The scope of this project is limited by the hardware capabilities, such as RAM, number of cores/processors, and CPU clock cycles, availability and usage of GPUs et al.

THEORETICAL FOUNDATION

Decision trees are the most intuitive form of classification. The dataset is partitioned recursively to form a tree. Partitioning is done so as to maximize the information gain. If we achieve a minimum threshold of purity of class labels of data points, or the number of points at that point in less than a certain lower limit of number of points, then we form a leaf of the tree, and assign a majority class to the leaf. To perform classification of a given point, we traverse the tree following the split point conditions in the internal nodes until we reach a leaf, which is then our predicted class label for that point.

For internal nodes, we have to choose a split point. The split point is chosen so as to maximize the information gain, which is calculated as the difference between the original entropy of the dataset, and the weighted mixture of the entropies of the new datasets formed based on the split.

IMPLEMENTATION

The decision tree is built following the same rules as specified in the theoretical foundation. The decision tree has two hyperparameters, min_pts i.e. minimum number of points below which we will form a leaf, and purity. Since I cannot directly control the number of nodes in the tree, to plot the graph of test error vs number of nodes in the tree, I vary min_pts from 1 to 30, whilst keeping the purity at 0.9 .

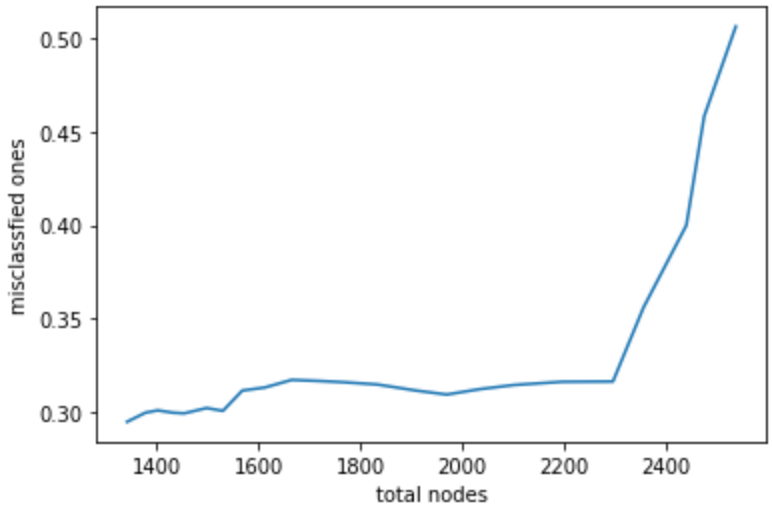
Another aspect to keep in mind is splitting categorical attributes. Although we can consider evaluating all subsets of the domain for maximum information gain, that would make the function of exponential order. Thus, for categorical attributes, I have only taken one value of the categorical attribute at a time into consideration.

OUTPUTS AND OBSERVATIONS

Once again, the data is skewed towards class “<=50k”.

Thus, to properly evaluate the data, I have plotted only misclassified “>50k”s against the

total number of nodes.



# of training data points: 23649

# of test data points: 6497

Note: On the y-axis, it says “misclassified ones”. This is because the class label “<=50k” was encoded as 0, while class label “>50k” was encoded as 1.

CONCLUSION

We can clearly see that the misclassification increases as the number of nodes increases. This is due to the fact that more nodes here means that the model is overfitted. As the number of min_pts decreases, more nodes(and more leaf nodes) are formed, leading to overfitting. Thus, the optimal decision tree should be one with around 1400 nodes(This includes both internal as well as leaf nodes).

REFERENCES

1. Data Mining and Analysis: Fundamental Concepts and Algorithms: Textbook by Mohammed J. Zaki and Wagner Meira
2. Class notes