Task

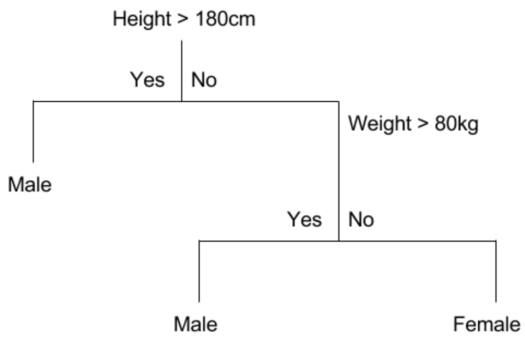
Explore the behavior of the J48 decision tree classifier in different discrete and numeric domains using a 70% training set and 30% testing set split. Additionally, induce several decision trees, each from a different-sized subset of the training set with the same testing set for testing each induced tree. Plot both the changes in induction times and error rates on respective vertical axes against the number of examples used.

Background

A Decision Tree is an easy way to visualize a complex classifier. They consist of nodes, or gates, that test the value of an attribute. These nodes are connected with edges, or branches, that correspond to the outcome of a node, leading either to another node, or a leaf node. Leaf nodes are the end nodes that predict the class label of the examples.

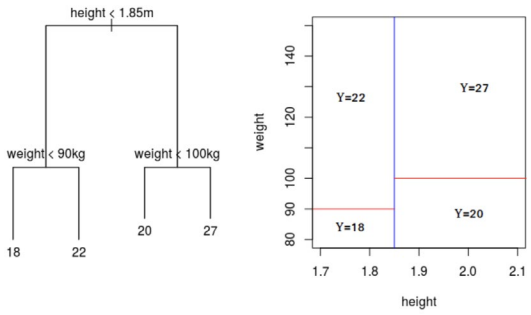
There are two main types of tree classifiers, the first of which is a classification tree. These trees are constructed from domains in which the outcome of the tree is a discrete value. Classification Trees are built through binary recursive partitioning which means that the outcome of a node can only be a yes or a no. Classification trees are used to classify discrete data sets.

Classification Tree



Regression trees are constructed similarly, however they operate on numerical domains with continuous values, such that the outcome of a node is dependent on the value of the attribute and the threshold of the node.

Regression Tree



Classification Procedure

To ensure we obtain the most accurate results, we used the following procedure

- First, we import the necessary Weka Python Wrapper and Java Virtual Machine libraries to run the experiments in a Jupyter Notebook. All codes are implemented in Python.
- The data is then loaded in, and it is specified whether the target class attribute is first or last in the data set.
- The decision tree classifier is then configured to the data set, resulting in the corresponding confusion matrix.

- Then the training set is split into separate portions and the decision tree classifier is then run on each subset, outputting the accuracy, error rate, time to run in nanoseconds, and the size of each subsequent subset is given.
- Using the information gained from the previous step, the Error Rate vs. Number of Examples plot and the Computational Time of Induction vs Number of Example plot are graphed.
- We interpreted the graphs to understand the behavior of the Decision Tree Classifier on each data set.

Goal

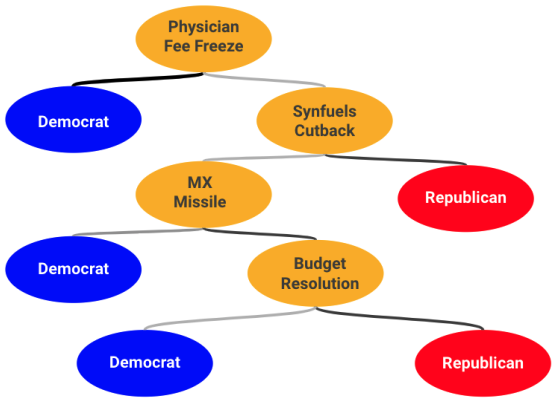
In this task, we will examine four data sets: Congressional Voting Records Data Set, Mushroom Data Set, Dry Beans Data Set, and a Connect 4 Data Set. For each data set, we will perform the above classification procedure and give some analysis based on the plots that are produced.

Congressional Voting Records Data Set

The Congressional Voting Records Data Set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. Here is a breakdown of the dataset.

- 435 instances
- 16 attributes

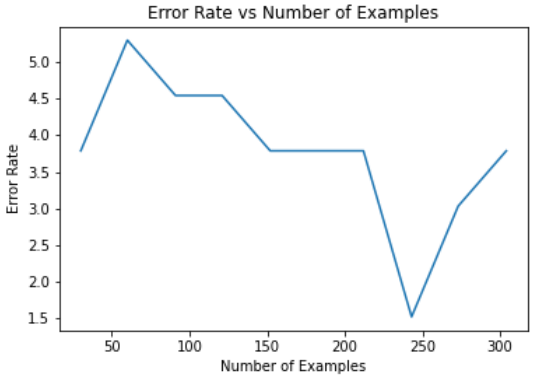
It proved to be a very simple binary data set that once visualized as a tree, was easy to follow. The illustration below shows what an induced tree on the entire dataset looks like with each leaf representing voting outcomes of whether a congressperson is a Democrat or Republican. Black lines represent a “No” vote while grey lines represent a “Yes” vote.

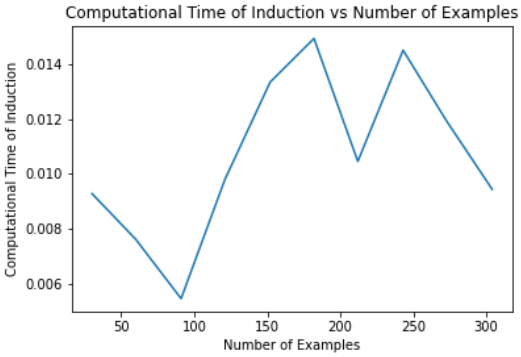


The dataset was then divided into 10 subsets and a classifier was induced from each. The graph on the left shows the error rates as examples were added for each induced classifier. We observe a peak error rate at around 75 examples and subsequent drops until a steady error rate (3.75%) from 150 to 200 examples. Our lowest error rate is around 240 examples when approximately 80% of votes have been accounted for. The error rates then rise until they reach the 3.75% steady rate and 100% of congressional votes have been accounted for.

The graph on the right shows the computational induction time as more congressperson examples are added. We observe a massive increase from 100 to 190 examples as well as a steady incline from 190 to 200 examples and 250 to 300 examples. These steady inclines are very minute (between 0.010 - 0.014 seconds) hence can be attributed to changes in computational load.

Since congresspeople tend to vote either independently, bipartisan, or across party lines, it is observed that in this dataset, each subsequent example, is increasingly complex in example size, attribute count, and class distribution. This adversely affects the error rate and computational induction time.

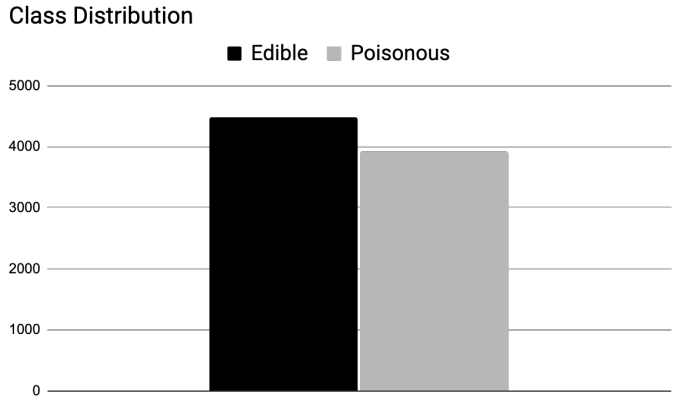


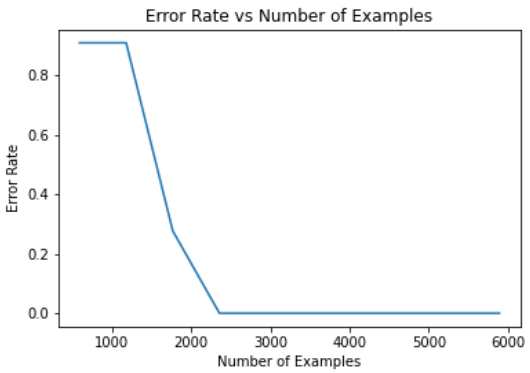


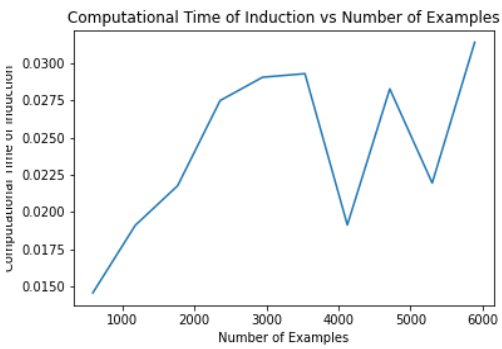
Mushroom Data Set

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. Here is a breakdown of the dataset:

- 8416 instances
- 23 attributes







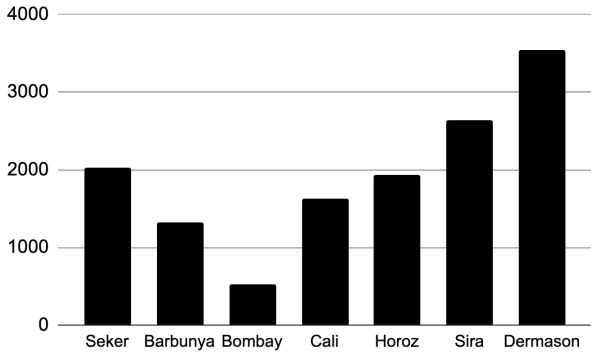
From the left diagram, we observe that perfect accuracy is achieved with only 30% of the training set. This is because the dataset contains a few highly informative attributes, the root of the tree can classify 55% of the data while depth can classify an additional 37%. Because of this, the resulting tree is small and only requires 5 of the 23 total attributes.

The highly relevant attributes of this dataset cause the decision tree algorithm to uncover the most effective tree in under 30 milliseconds. Although it may seem volatile on the chart, these apparent changes are only due to the minuscule increments of the y-axis. These changes are so insignificant it is likely due to the computational load of the machine at any given moment rather than the efficiency of the algorithm. The computational increase is practically insignificant, particularly since anything beyond 2500 examples is unnecessary.

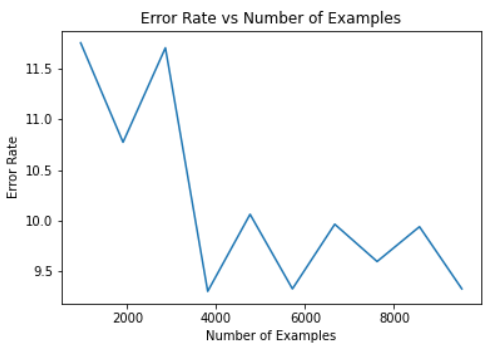
Dry Beans Data Set

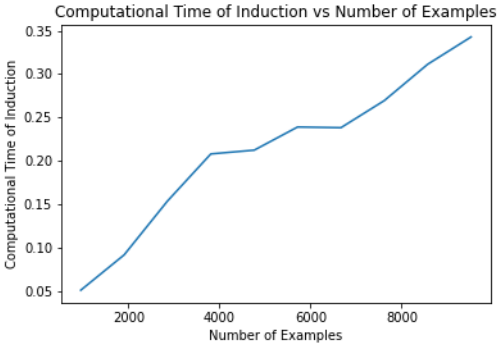
The Dry Beans Data Set consists of continuous numerical measurements of 7 different types of beans. The data set can be broken down into

- 13611 instances
- 17 Attributes



After running the J48 classifier we were able to find that the error rate remains relatively consistent regardless of the number of examples within the training set. The error rate only fluctuated 2% as it dropped to its lowest at the 30% training size.



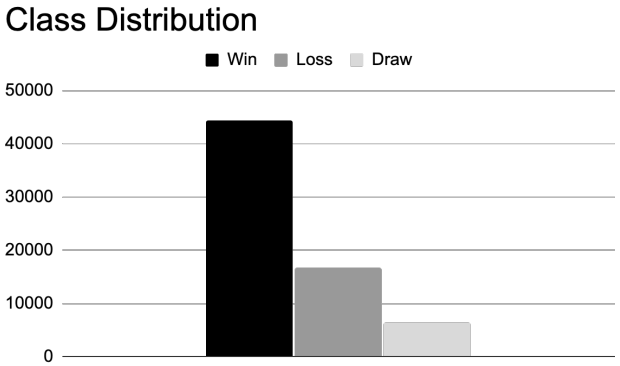


The computational demand of the J48 classifier increased quickly with the number of examples used in the training set, displaying a 100-millisecond increase each time the number of examples used doubled. With this data set, the computational cost becomes unnecessarily expensive past the 30% training subset.

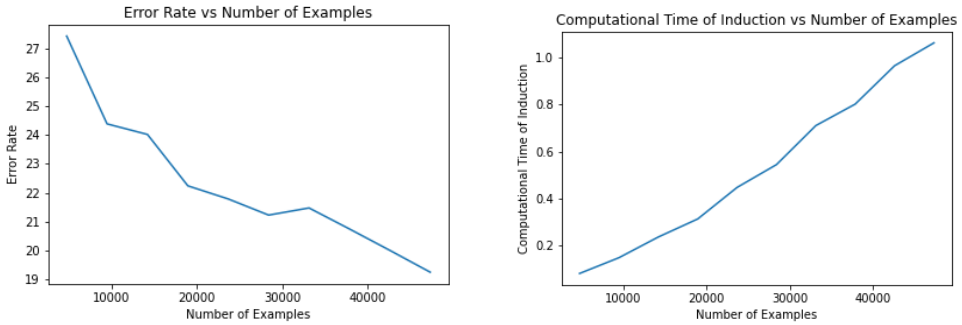
Connect 4 Data Set

The Connect 4 Data Set contains all legal 8-ply positions in which the game has not been won, and the next move is not forced.

- 67557 instances
- 43 attributes



The diagram below to the left is a plot of the Error rate in regard to the number of examples trained from in the training set. The graph shows a strong negative connection between the two variables, as the error rate falls about 4% from a 30% training set to a 50% training set. This correlation suggests that as more examples are added the error rate would continue to fall.



The diagram above to the right contains the plot of the computational time of induction as compared to the number of examples used in the training subset. The computational induction time of the classifier behaved almost linearly. The time to compute increased by about 200 milliseconds every example increase of 10,000 demonstrating a trade-off between the performance of the classifier and the accuracy in this domain.

Conclusion

Decision trees are best used when the number of relevant attributes is low. They can also be used to determine the relevant attributes in a given dataset. The depth of the decision tree is therefore related to how many attributes truly influence a classification.

From the above trials, it is observed that additional examples are not always necessary if few relevant attributes carry enough weight. Adding more examples can adversely affect the accuracy rate in a positive or negative manner. In some cases, the accuracy rates remain constant with the addition of more examples.

Balanced datasets also result in improved accuracy and faster tree induction. Additionally, increased complexity (poor class distribution, large amounts of relevant attributes, etc.) demands larger, more intricate trees with higher computational costs.

Data Sources

Google Collaboratory Implementation
Congressional Voting Records Data Set
Mushroom Data Set
Dry Beans Data Set
Connect 4 Data Set
Tic-Tac-Toe Endgame Data Set