Britton Gaul
u0915408

CS 5350 Midterm Report

Competitive Project

Preprocessing

To begin this project I had to process the data from the csv file in an organized manner. There are a couple different methods I have used so far to accomplish this.

The first way the data processed was for the decision tree implementation. The data was read into a dictionary that held each entry, and each entry had a corresponding list that held the values for each row. The position of each of the attributes could then be easily found by the pos function, that returns the index of the given attribute name. A attribute dictionary was also created to be able to easily lookup the different labels and their possible values.

The second method I used was to store the data in the pandas library DataFrame class. This was the approach taken for the perceptron method. The functionality in this class made it easy to organize the data by the attributes for each row. After the data was loaded into the proper columns it was required to check for unknown values in columns. By checking the occurrences of '?' in each column. The next step was to transform the non-numerical columns into numerical representations that can be used in perceptron algorithm. The final step of preprocessing for this approach was to set the feature selection. The feature selection of the training data that was selected was the income>50K label, because this is the label that will be predicted.

Algorithms Implemented Thus Far

The first method used was the decision tree algorithm from the beginning of the semester. First a decision tree needed to be built for the label to be predicted. The first step in building the tree was to find the best root for the tree to split on. Then a metric was to decide what attribute to split on. The different metrics used were information gain, majority error, and Gini Index. After using all three it seems that the Gini Index metric received the highest score, followed by information gain, and then majority error receiving the lowest score. After the root was selected the child nodes were found. To decide the child nodes a similar approach was taken as the root. The child nodes would find the next best attribute to split on and select it for the leaves of the tree. It would repeat this process until the max depth of the tree was reached. The depth of the tree chosen for this was thirteen. Thirteen was chosen because after increasing the depth of the tree by one, the error began to stabilize and would no longer decrease after thirteen.

The second method used to predict the income level was the perceptron method. This is currently being worked on. The method I am using is the voted perceptron algorithm. It begins by going through the training data and updating the predicted value until the desired value is reached. Then the predict method uses this information to predict the unknown income value. The prediction always results in true for the income at the moment. I believe this is because of the way I am setting up the data. I am having trouble loading in the testing data set properly, because of the missing income column. It does not align with the training data set that already has this column. This is one of the problems I will be working moving forward.

Moving Forward

As I continue to work on this project the first thing to accomplish is getting my perceptron algorithm working properly, as mentioned above. The next step will be to implement future algorithms and methods that will be mentioned moving forward in this class. Two that will be looked into is support vector machines and logistic regression. Ada Boosting and Random Forrest are two more approaches that I plan on implementing in the future. I would like to see how accuracy improves when adding these methods to the decision tree implementation.