ML Final Project Report

Avishek Choudhury - u1428226

December 2023

1. Introduction:

   As of now I have tried to two Machine Learning models to train the data and predict the results.

   Model 1: Single Full Grown Decision Tree using ID3 Algorithm

   Model 2: Random Forest using ID3 Algorithm

2. Steps Taken:

   Step 1:
   First few steps involved analysing and prepossessing the data for it to trained using the suitable model.

   Step 2:
   Going through each features and their attributes, then deciding whether to group a few attributes to one attribute for features which has a large number of attributes. But, I moved on with all the attributes without any grouping for any of the features.

   Step 3:
   Transforming the continuous values to discreet data so that each feature has fixed amount of attributes. To perform this transformation I used the information given in the column tab of the project.

   Step 4:
   I split most of the continuous data column into low, medium and high except a few columns. The columns in which the data was well spread out I used four attributes very low, low, medium and high. And the columns where the data is majorly clustered in two different places, I have used to two attributes low and high.

   Step 5:
   Same approach has been used for both the training models.

   Step 6:

   Processing the missing data in the dataset:

   The data which was missing(?), I replaced it with the attribute which is in majority in that column(feature). Used the same approach for both the models.

   While training the models I used both sets of data - One with the missing values replaced with ? and another with the majority attribute.

In the case of the missing values dataset I used "?" as an attribute for the features where data was missing.

Note: Here it was observed that the model was performing better when the missing cell was not replaced with the majority value. Hence, i moved on with the "?" as attribute.

Step 7:

Once the data was processed I used two models with multiple trails changing various parameters to get the slightly better results.

Step 8:

Single Decision Tree:
Performed multiple iterations by varying the depth of the decision tree.

Random Forest:
Performed multiple iterations by varying the size of the forest i.e. the numbers of trees used to make the average prediction.
Performed multiple iterations by varying the number to subset of features to be picked to find the feature with the lowest entropy.

3. Summary:

After multiple trials of the single decision tree it can be concluded that it performed better when the depth was slightly less than the maximum depth of the tree.

It can be confirmed that the Random Forest was performing slightly better than the Single Decision tree.

After performing multiple iterations by varying the size of the forest, it can be concluded that the performance improved up to a certain point when the size of the forest was increased but eventually it reached the saturation point at around 300-400 iterations and there was no improvement.

4. Further Approach:

I plan on to use multiple training models I have used so far with several variations to each model.

Based on the previous observation I will be picking the models which performed fairly better like the Random Forest, Bagging and Adaboost for now and as we move forward in the class, I will be implementing the new models that I learn.

I will also true to use the combination of these models to improve the performance.

The data-processing approach also needs to be improved to handle the continuous data.

Missing data can also be replaced by better values like taking the median or the average of the data.

5. Final Submission:

- For the final submission I have used neural networks to get the prediction.
- First I have grouped continuous features to discreet groups.

- Then, I have used sklearn's LabelEncoder library to convert string name of the groups for each features to integer values so that it can be converted to Tensor data.
- I used the Pytorch library to get the final prediction result by passing the Tensor data.
- I tried 'RELU' and 'TANH' activation function and observed that 'RELU' gives highest level of accuracy.
- After trying multiple combinations of Depth and Width, the best results were observed when Depth = 9 and Width = 100.
- Hence, I picked the results of the above mentioned combination for the final submission with the accuracy: 0.88398.