Isai Mercado Oliveros
March 3, 2016

**Decision Tree**

**- Cars data set and the voting data set**
**- Do not use a stopping criteria, but induce the tree as far as it can go**
**- Why would you and in what cases would you not get 100% accuracy?**

I think that even if a tree is left to grow as far as it can, it will not classify new data with 100% accuracy because most of the time, the leaves are not pure, which means that the data set might lack some chunks of information contained in features that were not recorded to begin with. In addition, there is noise in data sets, so noise cannot be learned.

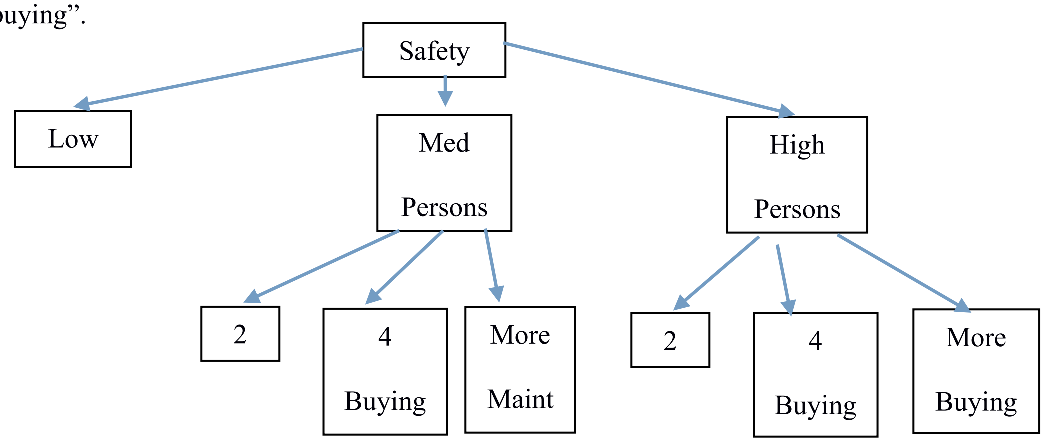
**- Use 10-fold CV on each data set.**
**- Report the training and test classification accuracy, and average**
**- Create a table summarizing these accuracy results, and discuss what you observed.**

| | Cars Data Set | Voting Data Set |
|---|---|---|
| Fold 1 | 0.9127 | 0.9767 |
| Fold 2 | 0.9130 | 1.0 |
| Fold 3 | 0.9130 | 0.9302 |
| Fold 4 | 0.9130 | 0.9545 |
| Fold 5 | 0.9884 | 0.9767 |
| Fold 6 | 0.9651 | 0.8863 |
| Fold 7 | 0.9653 | 0.8604 |
| Fold 8 | 0.9421 | 0.9545 |
| Fold 9 | 0.9306 | 0.9534 |
| Fold 10 | 0.9248 | 0.9090 |
| Average | 0.9230 | 0.9402 |

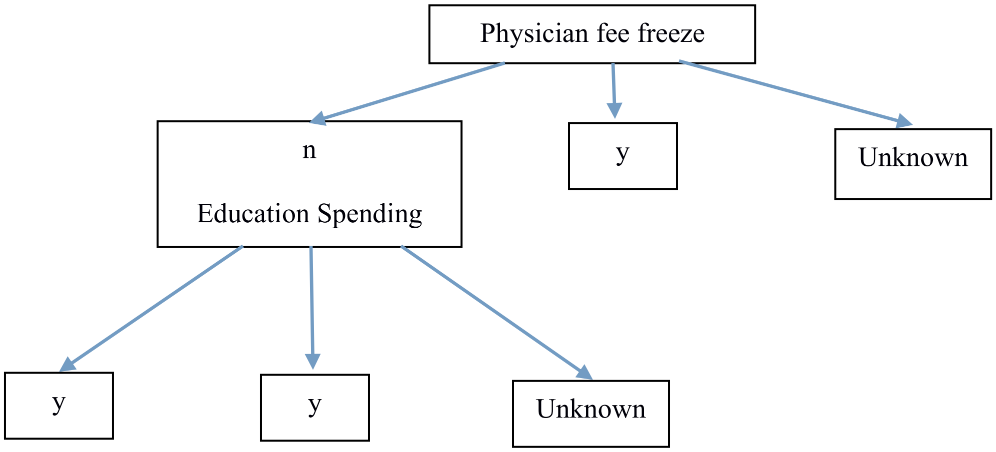
I observed that it is hard to get %100 accuracy with no pruning although it is possible as it is shown in fold number 2 of the voting data set.

**- Look at the induced tree and describe what rules it has discovered to try to solve each task**

For the Cars data set I saw that the most important attributes are first “safety”, second “persons”, and third “buying”.



For the Voting data set I saw that the most important attributes are first “Physician fee freeze”, second “Education Spending”




**- How did you handle unknown attributes? Why did you choose this approach?**

I handled unknown values as if they were another feature for every attribute. I chose this way because it was simple. It also made sense since the features in the voting data set are yes or no, it makes no sense to replace unknown values with something like a mean. In addition, choosing another technique such as treating unknown values as the majority feature, introduces bias for that feature. In my opinion, if representatives did not want to vote, that is a good indicator of political beliefs, so it should be used as a feature.

**- Create a table comparing the original trees created with no overfit avoidance in item 2 above and the trees you create with pruning.**

**a) the # of nodes (including leaf nodes) and tree depth of the final decision trees**

**b) b) the generalization (test set) accuracy**

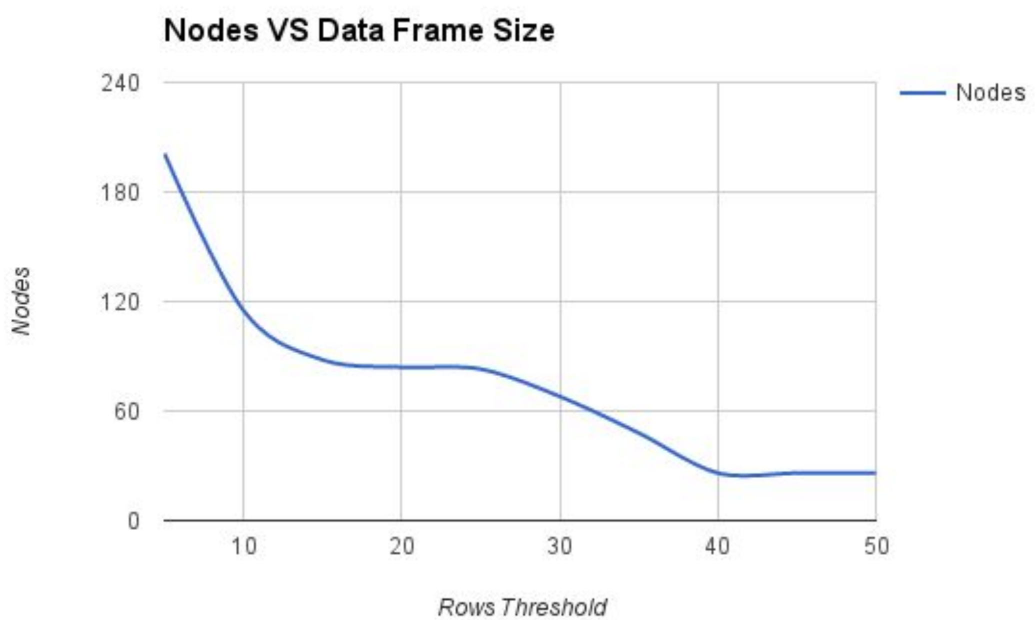
| | Nodes | Accuracy |
|---|---|---|
| Cars with no pruning | 259 | 0.9230 |
| Cars with pruning | 60 | 0.8385 |
| Voting with no pruning | 37 | 0.9402 |
| Voting with pruning | 7 | 0.9057 |

I thought that by letting the tree grow without pruning, the test set accuracy was going to decrease because the tree was very over fitted to the training set, but if we observe this table, the accuracy is greater when both trees are not pruned. On the other hand, I think that in real life, smaller trees are more useful because one of the main reason decision trees are used, is that their decisions are easy to track by the nodes. Thus, if the tree is very deep, and it does not generalize its learning, the purpose of looking into its nodes is no longer useful, so the smaller the tree the better.

**- Do an experiment of your own regarding decision trees. You can be as creative as you would like on this. Experiments could include such things as modifying the algorithm, modifying the measure of best attribute, comparing information gain and gain ratio, supporting real valued attributes, comparing different stopping criteria and/or pruning approaches, etc.**

I experimented with different stopping criteria where I stopped when the data frame in leave nodes had less rows that minimum threshold. This are my results for the cars data set

| Data Frame Rows Threshold | Nodes | Accuracy |
|---|---|---|
| 5 | 201 | 0.9166 |
| 10 | 115 | 0.8947 |
| 15 | 88 | 0.8860 |
| 20 | 84 | 0.8842 |
| 25 | 83 | 0.8825 |
| 30 | 68 | 0.8200 |
| 35 | 48 | 0.7633 |
| 40 | 26 | 0.7268 |
| 45 | 26 | 0.7320 |
| 50 | 26 | 0.7245 |




It is interesting to see that the number of nodes remains more or less constant while decreasing the data frame threshold until a certain threshold where the nodes drop.