HW1: ID3 DECISION TREE

BY PAUL AN

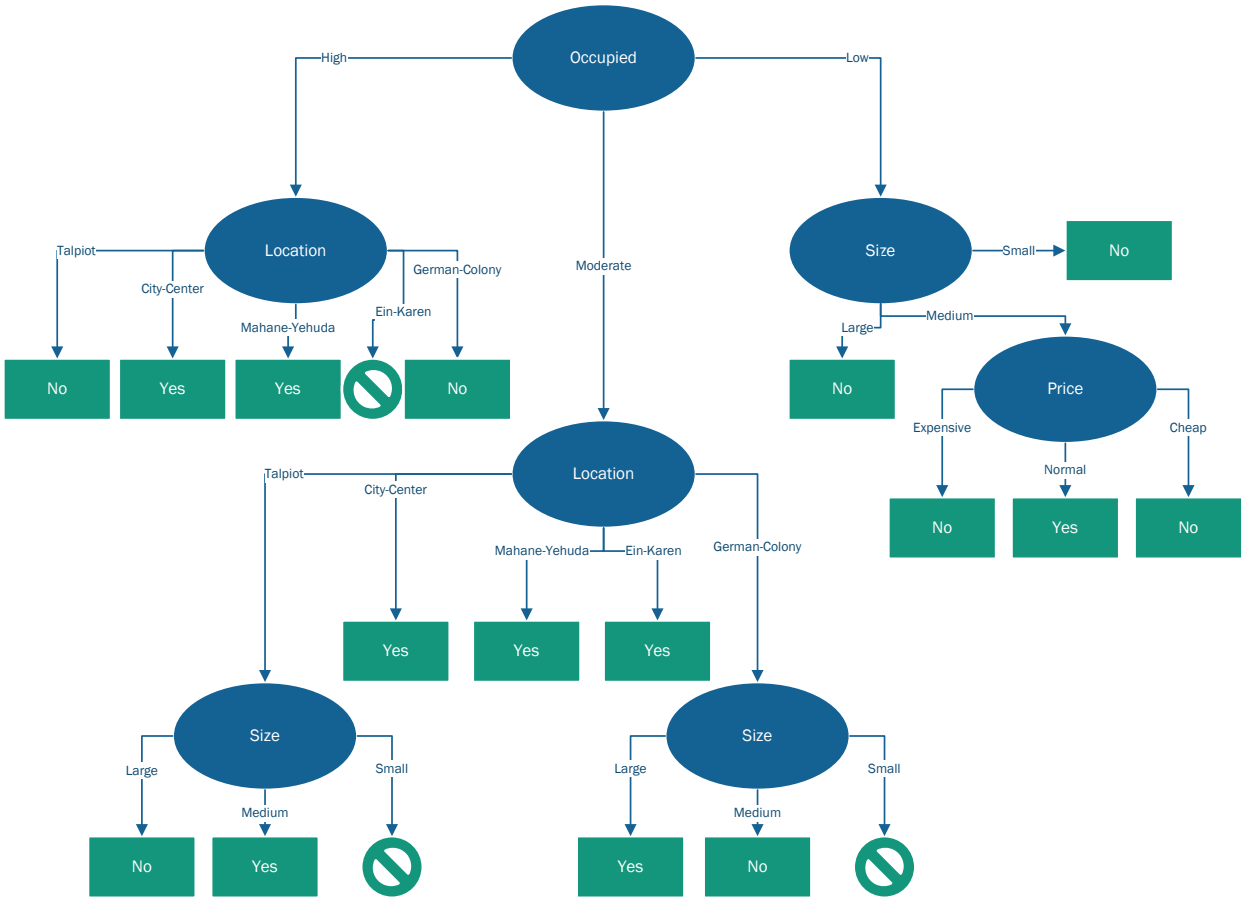
PART 1 IMPLEMENTATION:

Please refer to the project associated with the submission's archive folder for the source code of the ID3 Decision tree algorithm. This tree classifier is written in Python, with a few dependencies in numpy and math.

The output tree structure of the tree is the following:

```
Occupied,
Location,Location,Size,
No,Yes,Yes,None,No,Size,Yes,Yes,Yes,Size,No,Price,No,
No,Yes,None,Yes,No,None,No,Yes,No,
```

A visualization of the tree is shown here:

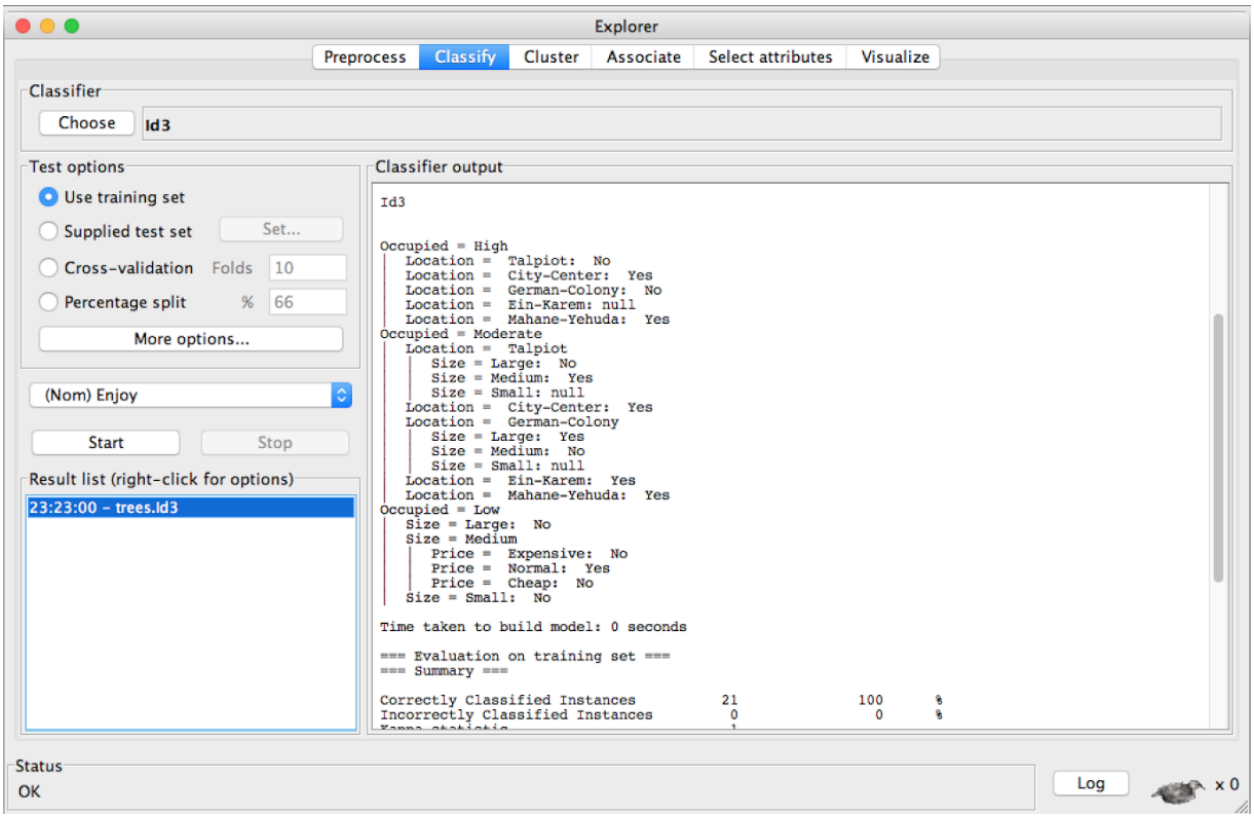


I have also taken the liberty of feeding the training set back into the classifier to verify E_{in}. Since there was a 100% accuracy in predictions with the training set, this means $E_{in} = 0$.

As for our prediction for Size = Large, Occupied = Moderate, Price = Cheap, Music = Loud, Location = City-Center, VIP = No, and Favorite Beer = No…This classifier will predict that **Yes** you will likely enjoy your night out with such given conditions.

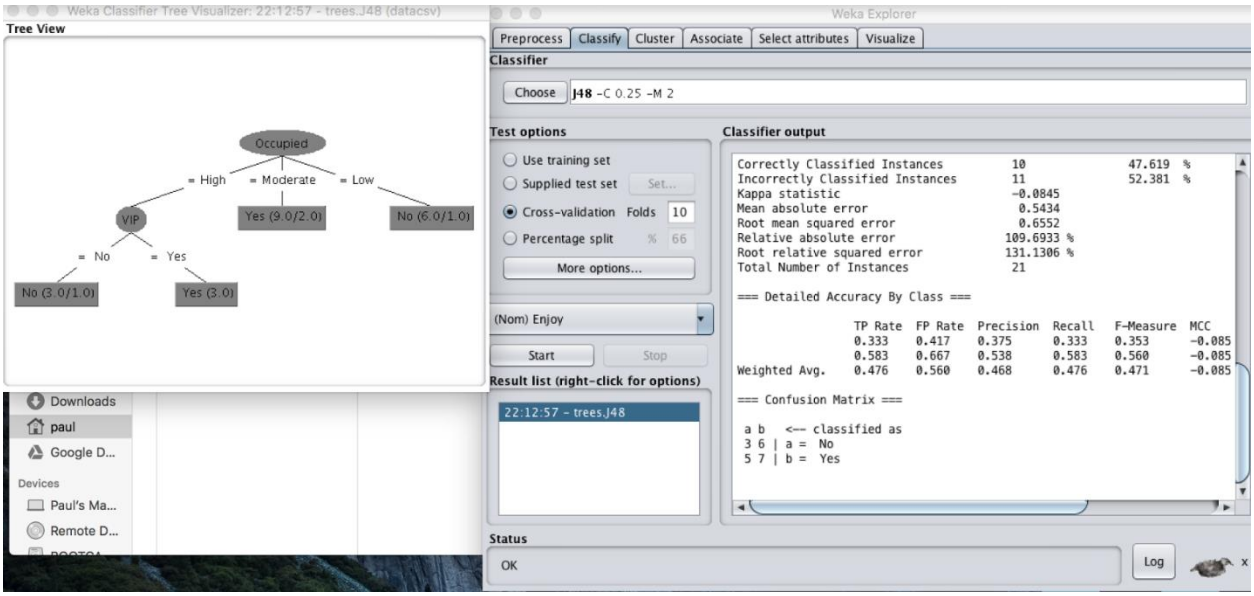
PART 2 SOFTWARE FAMILIARIZATION

Weka is a java-based machine learning software. Not only does it include a .jar library, which can be used for your own programming needs and customization, it also has some GUI baked into an app, which can be used to free-hand explore some data without having to create your own java code. To check my ID3 tree, I transformed the data text file into the appropriate format for Weka consumption (.arff) and ran the ID3 classifier in Weka. The following are the results:



Although there is no great visualization for ID3 in Weka, you can see the construction of the tree. I have also used the training set to test the tree, and got the same correct/incorrect as my own tree. In fact, looking at the text file, you can see that the tree constructed is exactly the same as the solution our ID3 implementation arrived at.

Interestingly, ID3 tree classifier was not available in the latest version of Weka. In fact, I had to download a much older version of Weka dating back to 2005 in order to obtain the solution. Currently, Weka's suggestion is to use the J48 tree classifier, which I believe is a variant of the C4.5 algorithms. The output and visualization of that tree is given in the following:



Interestingly, the tree looks completely different from the ID3 implementation, and uses the VIP Attribute while ID3 did not even consider that attribute.

We also saw in our tree that there were a few decision nodes where 'null' was incurred. I was taken aback by this when I first ran my project, but it was confirmed by Weka that it is indeed the correct solution. Perhaps this is one of the reason ID3 has fell out of favor. If the training set is not extensive enough, it seems sometimes the classifier won't know what to do with specific situations.

PART 3 APPLICATIONS

During my day job, I am a software engineer for a Software company that provides genomic analysis and visualization tools for both research and clinical use. In this software, thousands to millions of mutations, or 'variants from reference' is annotated as Pathogenic, Likely Pathogenic, 'Benign', etc. Although within the software we currently use, the classification is either done upstream, or decision tree is maintained semi-manually by a curator. However, such classification and curation is no doubt obtained from data scientists using a classifying machine learning algorithm.