EECS 440: Project 1 Writeup

Austin Feydt (apf31) and Mathanki Singaravelan (mxo174)

September 23, 2017

Note: all data for this writeup was generated using the script *writeup.py* in the P1 directory! Feel free to run it using *py writeup.py* to see all of the data.

a) The (average) CV accuracy for *spam:*

Avg Accuracy = 0.67263155922

The (average) CV accuracy for *volcanoes:*

Avg Accuracy = 0.721211665212

The (average) CV accuracy for *voting:*

Avg Accuracy = 0.988505747126

b) From running part a), we can see that for *spam*, the first test picked by our ID3 tree was *OS* 4 of the 5 times, and *geoDistance* the 5th time. For *spam,* all of the attributes were continuous, except for *OS*. Thus, it would make sense that it would be the best first split, since it is able to partition our example set into more than 2 partitions (whereas all of the continuous attributes perform a binary split).

For *voting*, the first test picked by our ID3 tree was always *Repealing-the-Job-Killing-Health-Care-Law-Act*. This seems to be probably one of the most polarizing bills from the list of all bills in *voting.names*, thus it would make sense that this would provide our tree the best first split from all of the attributes.

c) In Figure 1, we can how the CV accuracy of *volcanoes* and *spam* change as the depth of the tree is increased from a depth of 1 to a depth of 9. In general, the accuracy seems to actually worsen as the depth increases. The difference between a depth of 1 and 9 actually decreases the overall accuracy on the test set by almost 10 percent. We think that this is showing that our decision tree is quickly overfitting the data, thus the accuracy of the testing set decreases when the depth increases.

d) For this section, we chose depths of 1, 3, and 5 to test on. Our graphs can be found in Figure 2. For *spam* and *volcanoes*, we can see that in general, Information Gain outperforms Gain Ratio in terms of their accuracy, although both still show an overall decrease in accuracy as depth increases. Remarkably, however, for *voting*, the accuracy for Information Gain and Gain ratio are almost identical with the same downwards trend.

e) The final analysis of the algorithm is to compare the CV accuracy to the accuracy on the full sample. As we can see in Figure 3, there seems to be a variety of correlation amongst the different data sets. For example, *spam* shows that CV accuracy and full accuracy both decrease at about the same rate. However, *volcanoes* and *voting* show us a very different perspective. We see that CV accuracy repeats the common trend of decreasing for both data sets, whereas full accuracy actually grows with depth.

In general, this algorithm does not seem nearly as accurate as we were expecting from it. At least for these much smaller tree sizes, the accuracies seem to have low correlation. From testing the algorithm with larger depths, it seems that accuracy begins to increase (especially for *spam*, which can become a huge tree if you don't bound it at all).

Numpy arrays were a very useful data structure for this algorithm, as they allow you to sort a matrix based on any column value. This allowed for quick sorting when finding continuous splits. Although it took upwards of 45 seconds to convert the *spam* dataset to a numpy array, it ended saving a lot of computation time down the line. However, building a tree for *spam* became very tedious once depth got too big (even at a depth of 7, things take a while). We definitely wish that we had more time to optimize the runtime of the algorithm and explore ways of recycling computations to make it more dynamic.

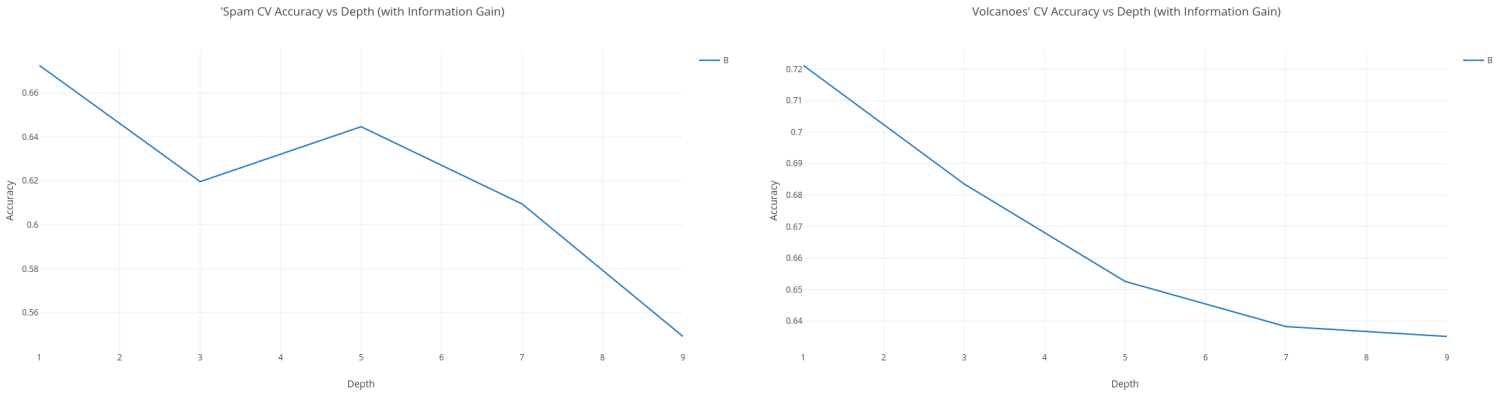

Figure 1: Part C: Accuracy vs. Depth for some data sets

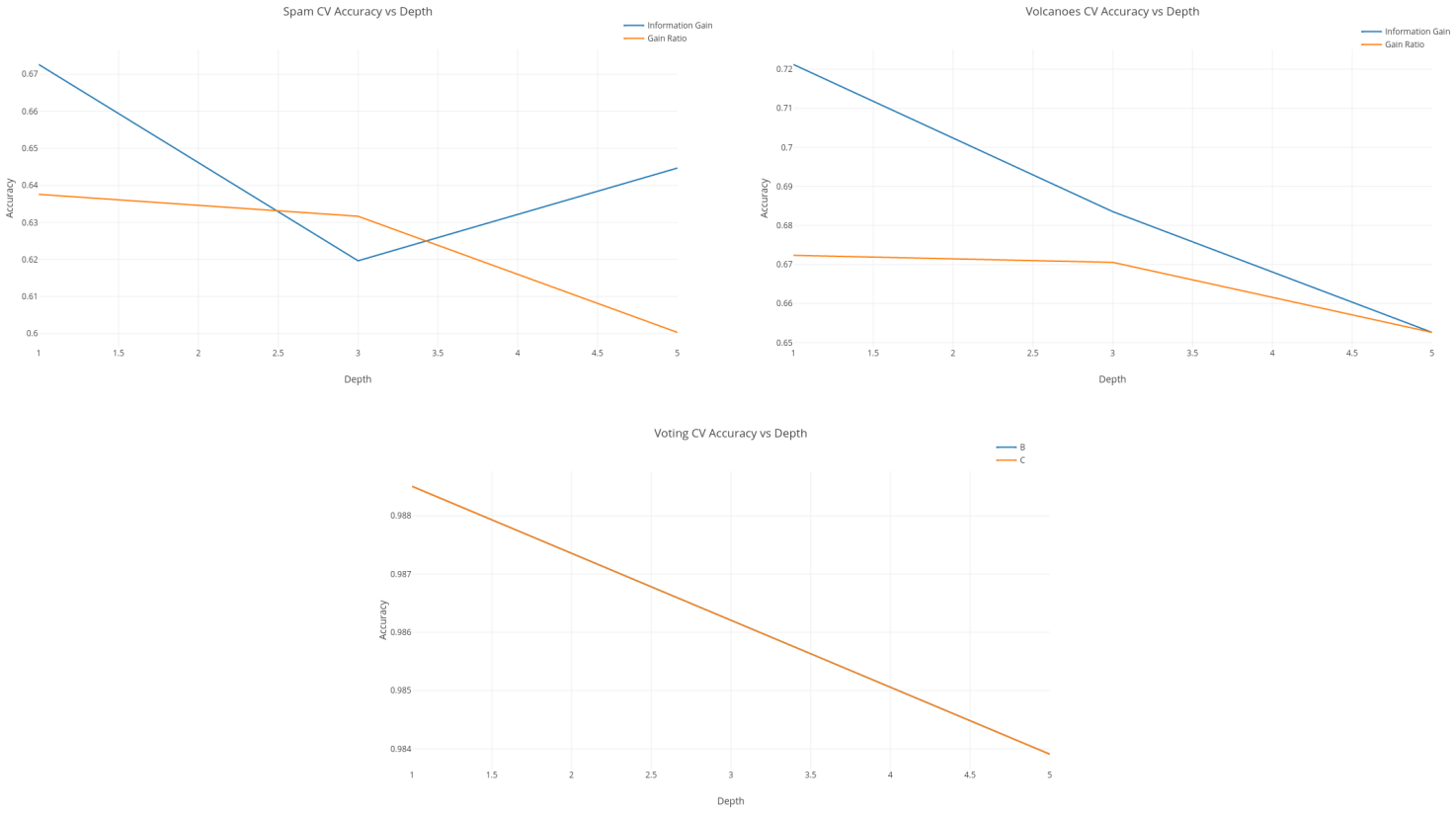

Figure 2: Part D: Accuracy vs. Depth for Information Gain and Gain Ratio

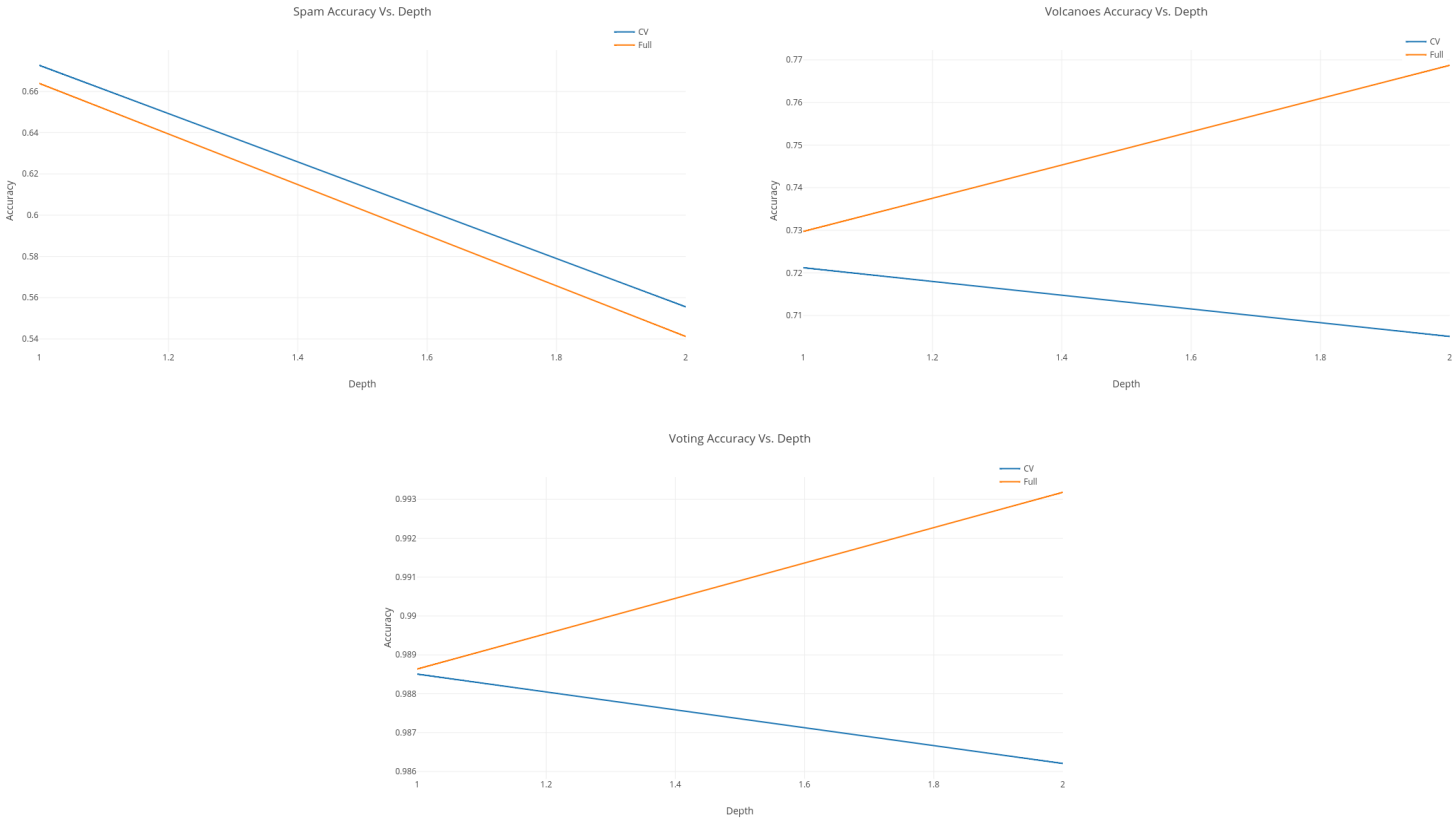

Figure 3: Part E: Accuracy vs. Depth for CV and Full Data sets