Project Checkpoint 2

Jakob Horvath - u1092049

November 13, 2020

My second submission to the *Old Bailey* project used the **averaged perceptron** algorithm in combination with the *tfidf* data set. Examples in the training, test, and evaluation data were modified to include all attribute vectors measured within the training data. This ensured that every attribute vector was of the same length, which made it easier to fit the data to my previous algorithmic implementation. The overall time complexity was much lower than that of the previously submitted decision tree algorithm.

The two hyper-parameters were the learning rate and the number of epochs. After running 5-fold cross-validation across a number of different combinations for these parameters, it was determined that a learning rate of **1** and an epoch count of **5** (reporting an average accuracy of 0.70400) would be used for final testing and evaluation. Generating this classifier on the whole training set took 24302 updates and returned an accuracy of 0.69333 on the test set. Finally, the Kaggle submission reported a score of 0.70552.

For the third submission, I implemented a support vector machine that makes use of **stochastic sub-gradient descent** as well as **bagging**. This algorithm also used the *tfidf* files, and similarly ran preprocessing on the data to create full attribute vectors for each example. This meant that the hyper-parameters to be considered were the learning rate, the number of epochs, the tradeoff constant, and the number of 'bags' to create (weight vectors drawn from running the **svm** function on randomized subsets of the training data).

After numerous trials, I was ultimately disappointed in being unable to make this classifier perform better than the perceptron algorithm. A learning rate of **0.01**, tradeoff constant of **0.0001**, epoch count of **3**, and a 'bag' count of **17** was used for my hyper-parameters. In addition, each bag utilized $\frac{1}{7}$ of the training folds. These constants achieved the highest discovered 5-fold accuracy of 0.68154. Running with these parameters on the whole training set, the training accuracy was 0.69669, the test accuracy was 0.65600, and the evaluation accuracy as reported by Kaggle was 0.68038.

After learning about boosting and ensembles, I plan on using relevant learning methods such as **AdaBoost** and/or **Random Forests** as part of my final three submissions to the project. In addition, I anticipate incorporating the *misc-attributes* data, either standalone or in conjunction with one of the other data sets. With this, I expect these new classifiers to perform significantly better than my past three submissions, hopefully reaching beyond 80% accuracy on the evaluation set.