Problem Description:

What algorithms perform best on sparse and binary classification data sets, when aided with pipes and grid searches to fully optimize said algorithms?

Descriptions:

K-Nearest Neighbors: Algorithm which makes predictions for the testing set based on the k nearest data points.

Random Forests: A extension of decision trees, which constructs multiple trees and returns the mean prediction of the individual trees.

Support Vector Machines: Creates boundaries along the categories of the training examples to separate data into the corresponding category, with an emphasis on creating large gaps. Employs the kernel trick to add a nonlinear feature to the representation to make the prediction more powerful.

Multilayer Perceptron: An algorithm which employs nodes that represent the input features, each using a nonlinear activation function, which together represent the weighted sum of the inputs to give a final output based on the weights calculated by the hidden layer.

Experimental Results:

Attached on back

Analysis of results:

Due to the large imbalance of the RNA folding data set none of the algorithms were able to learn enough to make informed and meaningful predictions. The f1 score never reached 15% for any of the algorithms with the highest retuned score being 12% by kNN when preprocessed with a standardizer. The standard accuracy score returned in the nineties for some of the algorithms but that could very well be attributed to random chance guessing as well, as the imbalanced nature of the data simply means that a majority class guesser could potentially also score just as high. The data frames support this as all the test run with all the different combination of attributes not being able to get meaningful f1 scores across all algorithms. KNN perfomed best most likely only because it takes all the attributes into account, instead of trying to find which affects the data the most which might be impossible in such unbalanced data, as it could vary wildly among the largest class, and might not exist within the smaller class. RF preformed so poorly because of this most likely, as the splits became meaningless and reflected more a majority class predictor more than splits based on the importance of the attributes. SVM and MLP sufferd as well for the same reason then, as no amount of weight fixing of support vectors could accurately split the data based on so little information on the minority class. As such nothing was learned from the data set because there was very little to learn.

The molecule data set on the other hand had plenty of information due to being fifty times larger than the RNA data set. So much so that the entire time taken to get the best attributes and score for the mlpc took six hours to run. The returned f1 scores were 93% and 92% respectively for the mlpc and kNNC, greatly contrasting that of the RNA folding data set. With a testing set alone that is ten times larger than the RNA data set in its entirety, it stands to reason that the algorithms were able to get more meaningful information, and take a much longer time, if nothing for the size alone.

Conclusion:

Both data sets highlighted two very distinct but equally important aspects of machine learning. The RNA folding data set showed that the kind of data collected very greatly impacts the accuracy of your predictions and that even the best algorithms won’t be able to make meaningful predictions even if fully optimized to be at their best. The molecule data set compounded this, being just as unbalanced, but having much more information to use to make more meaningful predictions. The much larger size has the drawback of taking much more time to run, with the mlpc taking six hours to run on the molecule data set when it took twenty minutes on the RNA data set. While nothing new was learned about the algorithms on this lab, the lab did show one of the fundamental aspects of machine learning. The algorithms aren’t perfect, and their predictions can wildly vary depending on the information fed to them. A greatly unbalanced data set will return great accuracy for just about any algorithm rendering their predictions meaningless as the f1 score showed. The parameters that differentiate the algorithms start meaning less when the data isn’t enough as shown by the fact that a majority predictor will preform just as well as these algorithms. On the other hand, the parameters that make each algorithm different start to mean much more on a data set like the molecules data set as there is more data to analyze. This makes optimization important as each algorithms score will reflect the parameters more. As such the testing time is much larger and shows the disparity between the time taken to run the molecule and RNA data sets. In conclusion, truly understanding why the parameters affect an algorithm will determine whether the information returned can be meaningful.