**685.621: Algorithms for Data Science**

**Rachael Kenney**

**Programming Assignment 2 Analysis**

**Due Date: August 12, 2019**

**Dated Turned In: August 12, 2019**

**Programming Assignment 2 Analysis**

For programming assignment 2, we were assigned to process the iris dataset and perform 4 classifications on it. The iris dataset provided for this assignment had been adjusted to include 2 new features and was “dirtied up” to allow for the processing steps to have an effect on the data. In this analysis, I will discussion some of the choices made in my implementation of this assignment along with insights obtained from the assignment.

To clean the data, we were required to replace the null values places throughout the data. I chose to take the time to split the data by class and replace nulls in each split with the mean from the split. This would preserve any separation between the classes so as to not cause accidental misclassification later. I also took the opportunity to normalize the data in the data transformation step so as to maintain the given features for as long as possible in the processing steps. I wanted to be able to see which features I was working with for as long as possible.

This was useful for my choices in the feature generation step. Here, I wanted to generate my new features straight from the features we have already been using. I’d noticed in previous assignments that the petal width and petal length seemed to have the strongest separation and was curious to see what the petal area versus the sepal area might show about the data. Petal area showed to have very strong separation in the feature ranking step. However, sepal area had the lowest ranking, even including the extraneous features provided for this assignment.

In the feature preprocessing step, I looked into what features had outliers based on the z score of each class. Setosa and virginica both had 2 outliers each but versicolor had no outliers. Feature ranking was performed after the outlier removal. However, it has already been discussed that petal width and petal area (the generated feature) had the highest rankings. The other rankings are not displayed in the output, but they were analyzed while working through the assignment. Lastly, PCA was performed on the total 8 features to transform them down to 2 dimensions.

The classification methods used were all built in methods from the scikit learn python package. This helped to make the code look more uniform overall and also was very simple to use and saved time. The Gaussian mixture method used for the expectation maximization did not report metrics well in the same way as the other classes, but when the predicted data was manually compared to actual output, it appeared to do about as well as the rest of the methods.

After the classification methods were performed on the processed data, they were then performed on the “original” dataset. That is on just the 6 features provided in the csv file without missing data. All scikit learn packages were seeded to the same random state so that performance was consistent. It was found that the processed data had a 100% success rate for all classification methods, but the unprocessed data performed slightly worse. Most methods seemed to be able to get all predictions correct except for the neural network.

Overall, the very mild difference in success between the processed data and the unprocessed data could be attributed to how small the data set. It is clear in a very large data set, these methods could be crucial to achieving even decent classification in the data.