**Morgan Gere, Dan Burke, Nick Tinsley, Natali Newman**

**IST 707- Final Project Report**

**Wine Quality Prediction**

For our machine learning topic we will explore the Wine Quality data set found on the University of Irvine’s Machine Learning Repository. The data for can be found at: <https://archive.ics.uci.edu/ml/datasets/wine+quality> . The objective of this analysis is to predict the quality of wine, and in particular, the physiochemical traits that define the high-quality wines in this wine variety.

Correlation

The highest correlations were between 1) residual sugar & density and 2) total sulfur dioxide & free sulfur dioxide (which would make intuitive sense). The correlations output from RStudio were plotted in Microsoft Excel (below) for easier visualization using conditional formatting.

Calendar

Description automatically generated with medium confidence

The output plot from RStudio is here:

Chart, bubble chart

Description automatically generated

Decision Tree Prediction

Diagram

Description automatically generatedIn the Decision Tree prediction model, alcohol followed by volatile acidity were the biggest factors in the decision on the quality of the wine. However, the accuracy of the prediction of quality was only 54.06% as seen in the output below.

## Accuracy : 0.5406   
## 95% CI : (0.516, 0.5651)  
## No Information Rate : 0.4366   
## P-Value [Acc > NIR] : < 2.2e-16   
##

Support Vector Machine Prediction

Between Decision Tree, SVM Linear and SVM non-linear with radial kernel, the best performing model was SVM non-linear with radial kernel when the quality variable was discretized to “high”, “medium”, and “low” categories. The prediction accuracy was approximately 77.36% accurate with this model.

Graphical user interface

Description automatically generated

**Figure 1**: Graph of SVM linear and non-linear with raw data and scaled data. In this dataset, the SVM non-linear algorithm performed the best. Scaling the data did not make a difference in model performance as seen in the prediction values (SVM\_comparison) below.

SVM\_comparison

## SVM.Linear SVM.Linear.Scaled SVM.RBF SVM.RBF.Scaled  
## [1,] 0.5210472 0.5210472 0.6196099 0.6196099

The prediction variable was then discretized into “high”, “medium”, and “low” categories and the best performing model (SVM with RBF) was applied to the data. This yielded in the best performance

Graphical user interface

Description automatically generated

**Figure 2**: SVM prediction model performance with raw, scaled, and discretized data.

SVM\_comparison2

## DecisionTree SVM.Linear SVM.Linear.Scaled SVM.RBF SVM.RBF.Scaled  
## [1,] 0.5282341 0.5210472 0.5210472 0.6196099 0.6196099  
## SVM.Discretize  
## [1,] 0.773614

In the discretized data predicting high, medium and low quality categories, the most important variables were chlorides, alcohol and density as seen in Figure 3 below.

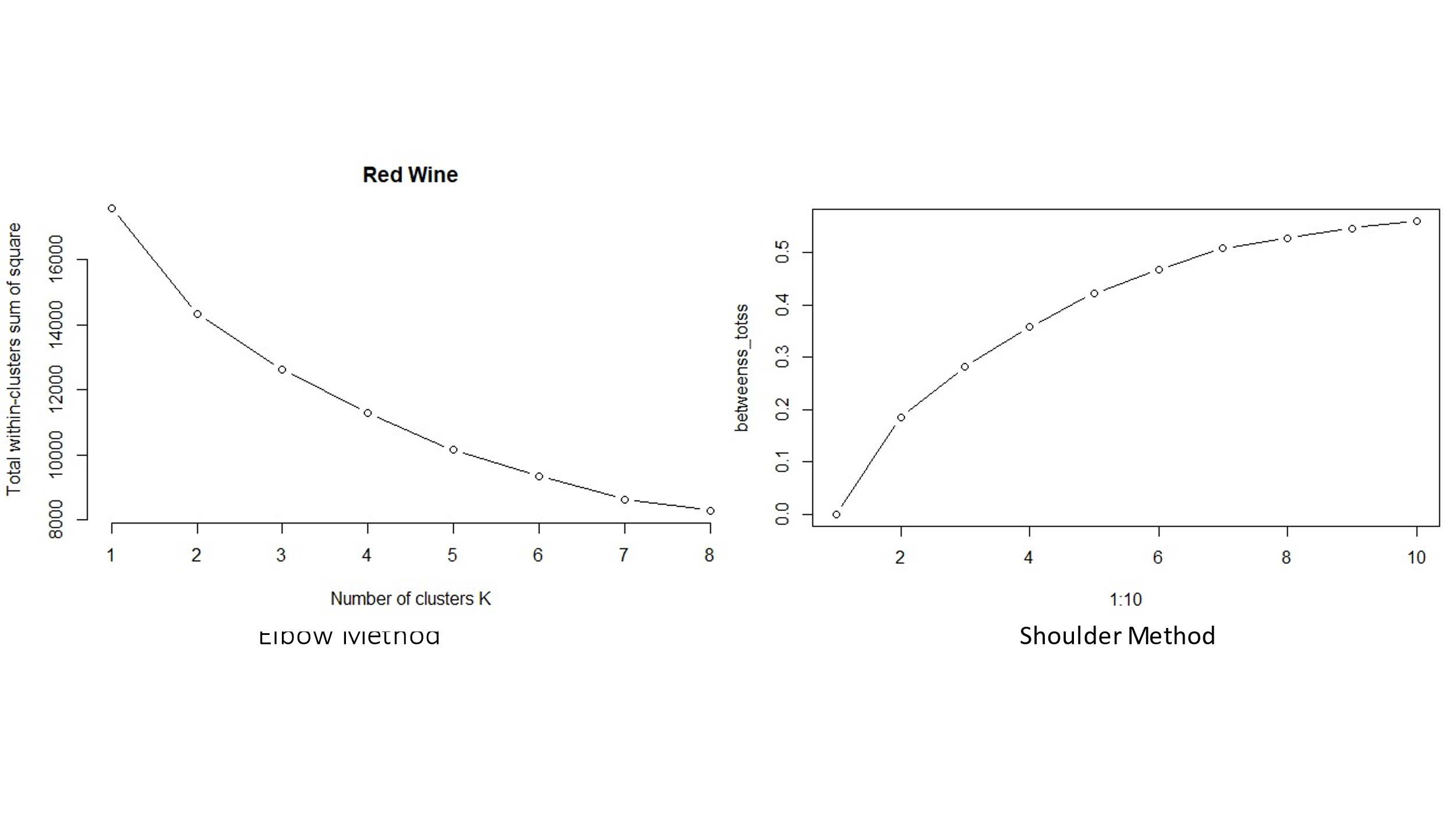
Chart

Description automatically generated

**Figure 3**: Variable importance in the SVM non-linear with radial kernel model with a discretized prediction variable. Of note, volatile acidity was not as important in performance prediction in the high quality category as compared to the medium and low categories.

K-means Clustering

To gain an understanding of the data set using clustering, k-means algorithm was used on the full data set. The elbow method and shoulder method were both implemented on the red wine data set. Finding that it should be used to find 5 clusters. This was found to be problematic as the clusters were very random with what quality of wines were in each one. After attempting to use specific attributes, while eliminating others, it was not yielding good results.

Chart, scatter chart

Description automatically generated

**Figure 4:** Elbow and shoulder method used for Red Wine data. **Figure 5**: K-means of full data set results of k=5 clusters.

The quality numbers were very skewed towards ratings 5 and 6. Outside those ranges the data was quite limited only having 10 examples of wine with a quality of 3. To observe high quality versus low quality the quality data was discretized so any wine with quality rating of 5 or lower was marked as “low quality” and any above was marked as “high quality”. The data set was broken apart into the different levels of quality and 10 random samples were taken from each of the levels. This new dataset was clustered into two clusters attributes that were highly correlated to others as well as attributes that were proving to have no change on the clustering were removed. This led to having two clusters one comprised of red wines with high quality and one with red wines that contained low quality. Accuracy, precision, recall, and F measure were all found to be 80%.

With the same attribute columns and the discretized quality of low and high the entire data set was then used with the k-means algorithm using 2 clusters. This yielded some much better results than previously. The clusters were close to one another but no overlapping and when checking one appeared to high quality wine and one low. Checking the Accuracy was 65%, the precision was 55.9% the recall was 72.4% and the F measure was 63.1%. This tells us that some red wines are being placed into high quality that are not at a higher rate, but it is less likely for the clusters to place a high-quality wine into the low quality.

Chart

Description automatically generatedChart, scatter chart

Description automatically generated

**Figure 6**:Equal data points and removed attributes **Figure 7**: Full data set with same parameters as Fig.6.

The same procedure was done with the white wine dataset. The white wine dataset has larger range of quality ratings but has less of quantity in those ranges. When the full dataset was clustered using the high and low discretized data it was found to have many high or low errors. To investigate this the quality was discretized into low being a rating of 5 or less mid being a rating of 6 and high being 7 or greater. The K-means was run but still set to 2 clusters attempting to predict high and low quality and showing how the “mid” quality rating wines were being handled by the algorithm. As was suspected the mid quality was being divided into the two clusters. 1049 in high quality and 1149 in low quality. This places our accuracy very high assuming the white wine with a rating of 6 is being divided correctly. This leads to some follow up questions about how the quality ratings were obtained.

Chart, scatter chart

Description automatically generated

**Figure 8**: Full white data set using the K-means cluster and parameters as fig.6

KNN was used to predict the quality of the wine. The wine datasets were split into Training and validation sets. The accuracy was found to be low only ~61.2% the quality data was discretized into high and low using the same parameters as before (low <=5 and high >5) The KNN was rerun finding ~75.6% accuracy and a kappa of 50.8%. This was followed up with predicting the validation set and a confusion matrix was used to find the accuracy of 76.8% and a kappa of 53.11%. This is a very fast method to achieve good results.

The KNN was run on the White data set and found similar results from the model. Accuracy of ~79.9% and a Kappa of 54.5%. The validation set prediction confusion matrix found ~80% accuracy and a kappa of ~54.9.

Both datasets were combined adding a column named type. This identified the wine as either red or white. A KNN was run on this in the same way and found to be ~78.3% accuracy on the model with a kappa of 52.9%. While the confusion matrix run on the predictions of the validation set were found to be ~79% with a kappa of 54.58%. This tells that combining all the data gives us a fast way of identifying new batches created.

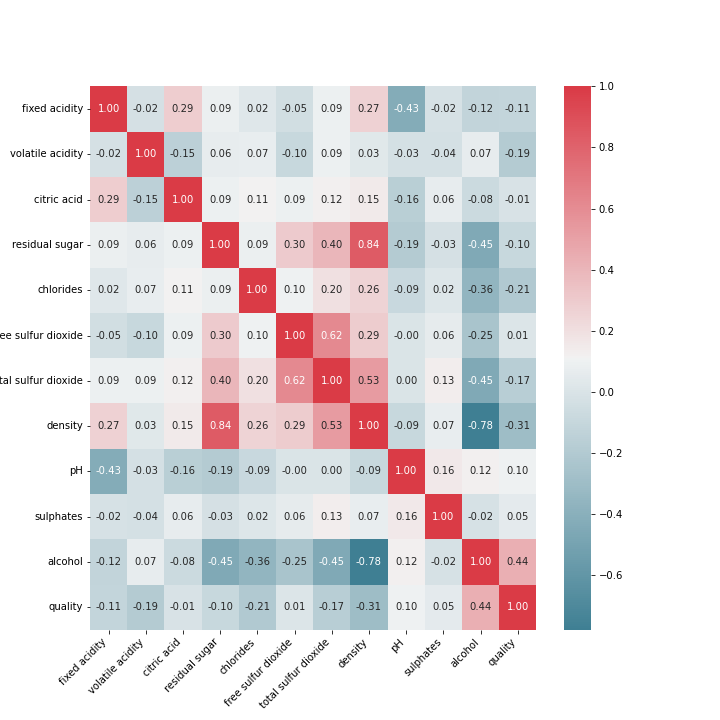
**Python Implementation: Random Forest & Naïve Bayes**

Chart

Description automatically generated Initially, a Random Forest and Naïve Bayes models were generated with R Studio, however, to explore both the potential differences in models and workflow, a Python implementation was utilized; leveraging the “scikit-learn” libraries/package.

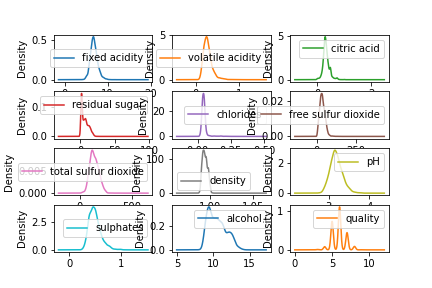
F**igure 10** - Attribute Exploration

Like other segments in this analysis, before any model generation we examine each attribute of every record contained within the dataset compared against its “quality” rating to construct a correlation matrix. Then further explore by attribute density (see figures 9,10).

After examining all attributes, of both red and white wine separately; in addition to the shape (record count) of each dataset, it was apparent that we possessed more data on white wine (4898 records) than red wine (1599 records). With this record count disparity, we proceeded to build models specifically for red and white, and a third for a concatenated dataset.

While building random forest models for both the separate datasets and the concatenated dataset, it became apparent that memory management issues were avoided with the python scikit-learn implementation. In our first R implementation we encountered memory utilization issues which may interactive development both time consuming and impractical due to RStudio crashing. After this development, it became clear that the python implementation is better suited for our analysis.

**Figure 11** Correlation Matrix

Each of these models were generated utilizing a test and train split. Various splits were tested through iteration, however we limited this to not exceed an 80/20 (train/test) split. Exceeding this 80/20 split was tested, however it produced unrealistic accuracy when predicting the test dataset. Though we do not have additional data to test these models, we concluded that we had encountered a situation where overfitting should not only be suspected but expected.

Like our Support Vector implementation within this analysis, we decided to segment the wine quality into three (3) groups, low, medium and high, with their respective ranges of 3-5,6-7 and 8-9 (see figure 12).

Figure 12 Density Plot

Text

Description automatically generated

**Figure 13** Grouping of Classification into Bins

**Random Forest**

|  |  |
| --- | --- |
| Dataset | Observed Accuracy |
| Red Wine | 78% |
| White Wine | 80% |
| Red & White Wine | 81% |

**Naïve Bayes (Gaussian)**

|  |  |
| --- | --- |
| Dataset | Observed Accuracy |
| Red Wine | 68.6% |
| White Wine | 66.3% |
| Red & White Wine | 65.4% |