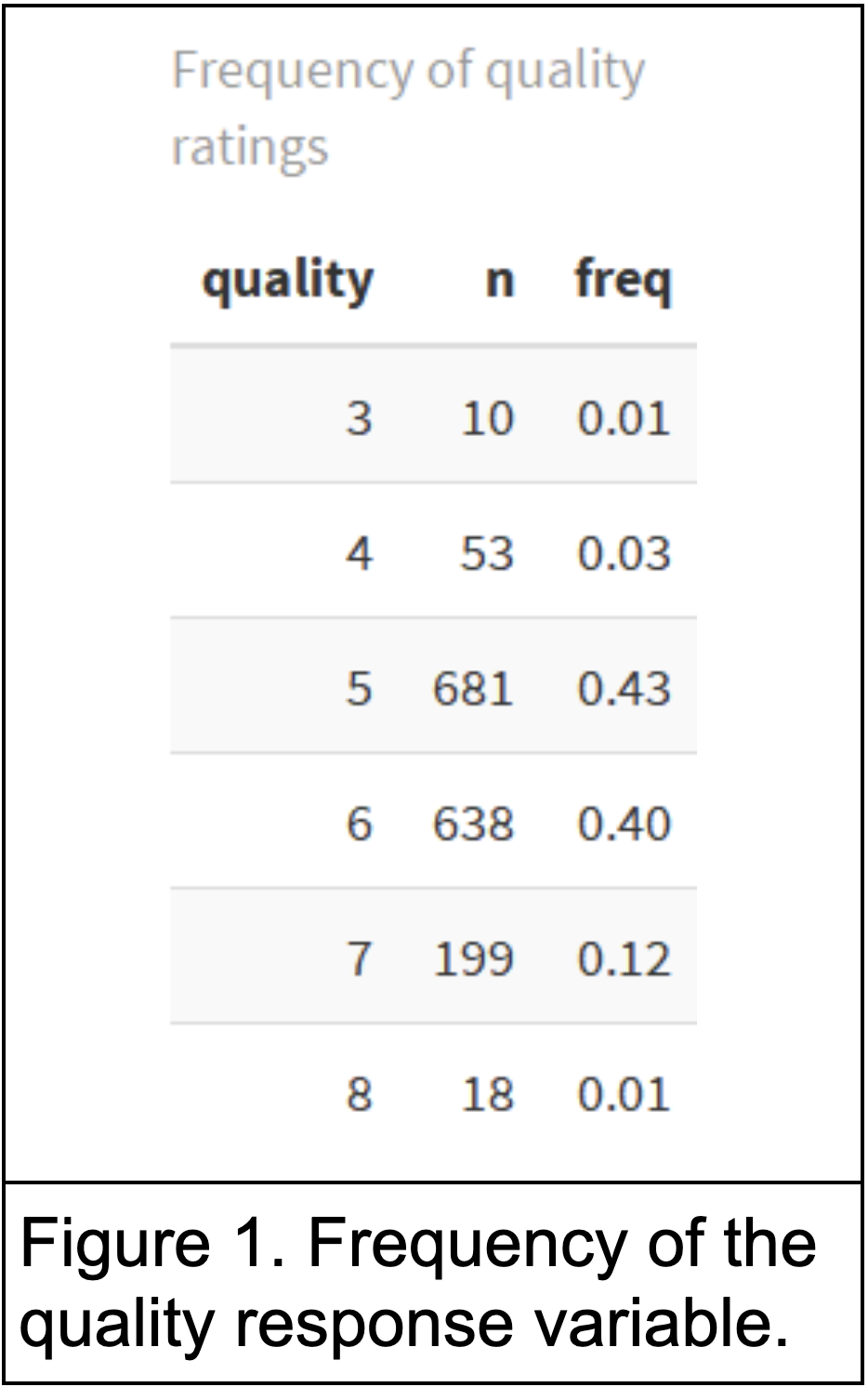
## Introduction

In this report, we will be working with a dataset that contains information about red wines. The dataset contains one observation per row (1599 total) and each observation has the chemical properties of a single wine along with the quality rating of that wine as given by a taste-tester. Our goal is to explore the data, fit various statistical models (both regression and classification) that predict taste-tester quality score given a wine’s chemical properties, and then determine which method has the greatest predictive power and can also provide valuable information for vineyards.

To add emphasis on a vital point: Our decision on which model is “best” will be made with predictive power in mind rather than model parsimony. While parsimony is often desirable, we are simply striving for a model that accurately predicts quality based on the 11 chemical property predictor variables and can give us clues into variable importance.



The response variable, quality, is designed to be on an integer scale from 1-10. However, Figure 1 on the left shows that taste-testers didn’t give any ratings of 1, 2, 9, or 10, and only 17% of ratings are outside of a 5 or 6.

For classification purposes, we decided to separate the quality ratings into two categories:

* Low Quality: A rating of 5 or less
* High Quality: A rating of 6 or higher

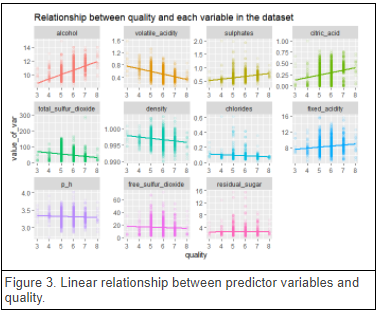
The class threshold decision is often arbitrary, but an unbalanced response variable like this seems to make the decision even trickier. That said, we feel that our High/Low cutoff is appropriate.

Figure 2 below contains information about quality, along with all the independent variables in the dataset**.**

|  |  |
| --- | --- |
|  |  |
| Figure 2. *Left*: Distributions of all variables. *Right:* Linear relationships between variables. | |

From the plot of distributions, we see that a few predictors are approximately normal while others, especially the two sulfur dioxide variables, are skewed right. We also have another look at how skewed the quality variable is towards ratings of 5 and 6.

The correlation matrix shows that there are several highly correlated pairs of predictors, the highest of which make sense intuitively. The free and total sulfur dioxide predictors are highly correlated, as well as pH with the measures of acidity.



Before we build models to determine the precise relationships between wine quality and a wine’s chemical properties, we can look at simple correlations to get an idea of which predictors are most related to quality.

Figure 3 to the left has these correlations and orders the individual plots based on the absolute correlation with quality. Hence, alcohol, volatile acidity, sulphates, and citric acid are the variables most strongly linearly correlated to quality.

### Subset Selection

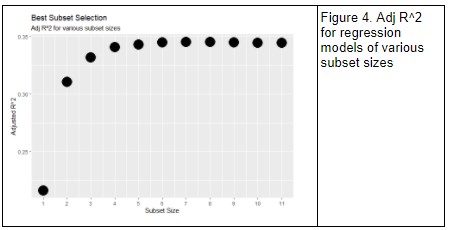
Another way we can analyze the predictors and their relationship to quality is through subset selection techniques. To be clear, the models we run in the following sections are fit on the entire set of predictors. This is an exploratory section to learn more about which variables are most useful.

Figure 4 to the left shows us that the increase in model performance (using Adj R^2) starts to decline sharply after adding the 4th and 5th variables. Additionally, Figure 5 below shows us the optimal set of variables for each subset size, using the best subset selection strategy.

|  |
| --- |
|  |
| Figure 5. Variables selected at each subset size, using best subset selection |

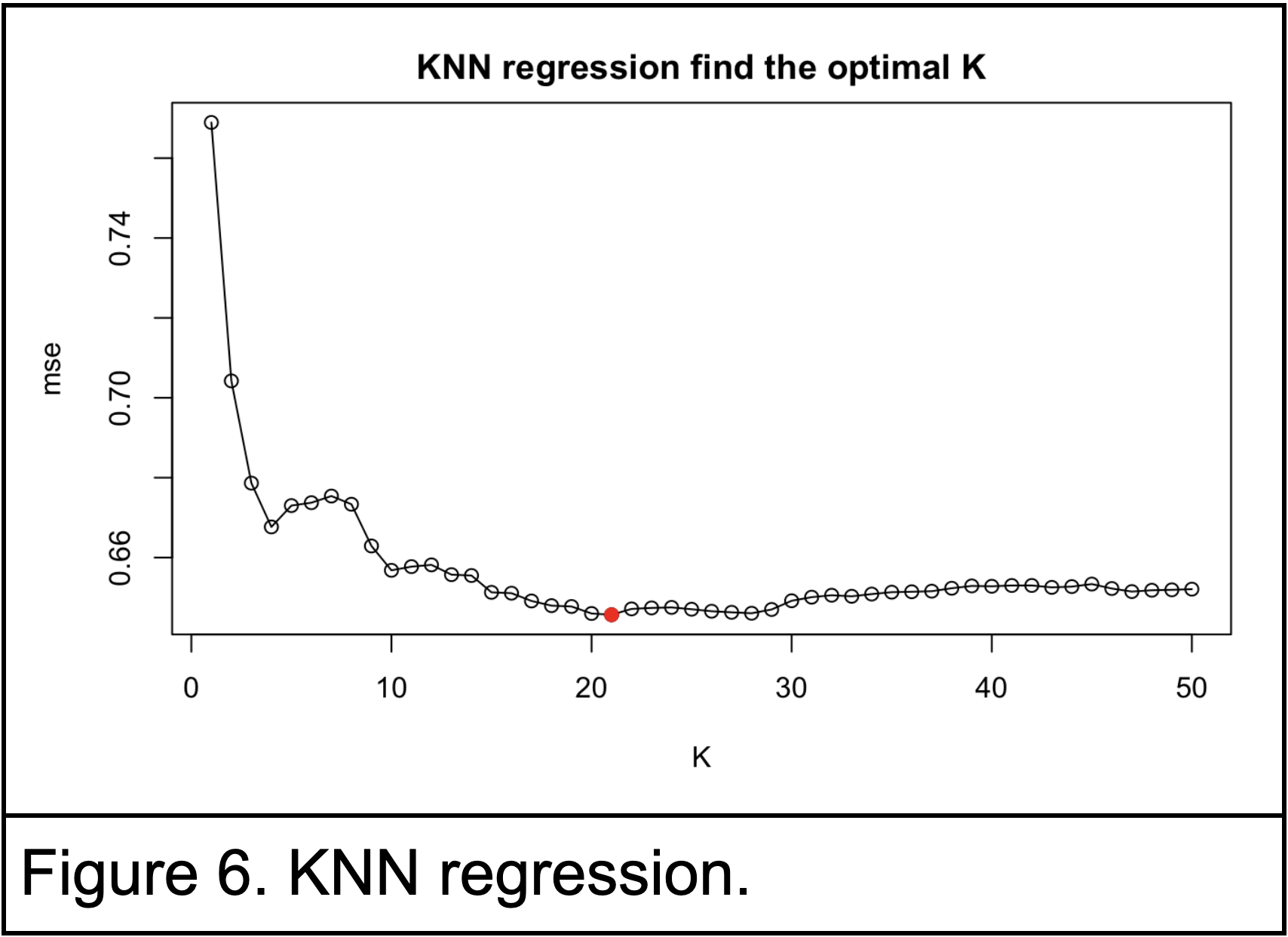
Forward and backward selection techniques led to differences in the models of subset size 8 and larger, though we know from Figure 4 that after 4 or 5 variables the models will only be marginally different. As Figure 3 suggests, alcohol, volatile acid, and sulphates will likely be the most important variables in these models.

## Regression Methods

In order to properly evaluate and compare the following methods, we split the dataset in half into a training set and a test set; the training data ends up at 799 observations while the test set has 800 observations. The same training set and test set are used for each of the following methods, including the classification models.

### Linear Regression

To start, we fit a few multiple linear regression models to the red wine training data. Initially, we fit a linear regression model using all the variables. Among all the variables, volatile acidity, total sulphur dioxide, sulphates and alcohol were significant. For this model, we got the value of R square as 0.34 and the mean square error was 0.4172. The output from the multiple regression fits can be found in the appendix (Figure A.1 in Appendix)

Next, we fit the second linear regression model using significant variables. R squared was similar to the first model. Mean squared error increased to 0.427. We fit the third linear regression model using interaction effect. R-square increased to 0.35. There was a decrease in mean squared error to 0.40. The output of the interactions effect linear regression model can be found in the appendix (Figure A.2 in Appendix).

### KNN Regression

We use KNN in both classification and regression. As KNN is a distance-based model, we normalized all the predictors data in order to have them in the same scale. We apply the KNN model on the train dataset using different K and get MSE in the regression model. For KNN regression, we can find in the plot that when K=21( Figure 6), we have the lowest MSE 0.647.

### Ridge Regression

Ridge regression adds penalty to the ordinary least squares loss function in order to prevent the overfitting. We fit the data to the ridge regression model using different lambda, and find out the best lambda is 0.053 which has a MSE of 0.395. The coefficients of predictors are as follows in Figure 7.

### Lasso Regression

Lasso regression adds penalty to the ordinary least squares loss function in order to prevent overfitting. We fit the data to the lasso regression model using different values of lambda, and find out the best lambda is 0.01 which has a MSE of 0.395. The Lasso could also make some coefficients of the model to zero which is able to reduce dimension. The coefficients of predictors are as follows, and we can see that the coefficients of fixed acidity, citric acid residual sugar and density became 0.

|  |  |
| --- | --- |
|  | Figure 7. *Left:* Coefficients from the Ridge Regression fit. *Right:* Coefficients from the LASSO Regression fit. |

### Principal Components Regression

PCR is a dimension reduction technique that involves creating a set of *M* principal components, which are linear combinations of the original variables. These principal components are then used as the predictor variables in a linear regression model. PCR assumes that the directions in which the original predictor variables show the most variation are most important for predicting the response. Since PCR involves fitting a model with fewer predictors, it decreases the likelihood of overfitting. However, a drawback of PCR is that it does not perform variable selection or provide coefficients for the predictor variables, so we cannot use it to determine which variables are most important in predicting the response.

To apply PCR to the wine quality data, we first standardized the data so that all the predictor variables are on the same scale. Then, we ran cross-validation using the training data set. The resulting cross-validation MSEs are shown in Figure 8. We chose 3 principal components for the model because the cross-validation MSE at *M*=3 is nearly as low as at any of the higher values of *M*. We next fit this 3-component model on the test data data, and found that it has a test MSE of 0.436. This error appears reasonable, and can later be compared with the results of other methods. Last, we can also fit the 3-component model on the full data set. With all 3 components, the model explains 59.78% of the variance in the predictor variables, but only 32.12% of the variance in the response variable (quality).

|  |
| --- |
|  |
| Figure 8. *Left*: PCR ten-fold cross-validation MSE as a function of the number of components.   *Right*: PLS ten-fold cross-validation MSE as a function of the number of components. |

### Partial Least Squares

Partial least squares (PLS) is a dimension reduction method, similar to PCR. Unlike the PCR method, PLS chooses the principal components based on the directions that best explain the predictors and the response, instead of only the predictors.

To begin, we again standardized the wine quality data, putting all the predictor and response variables on the same scale. We ran ten-fold cross validation on the training data set to identify the best number of components. The resulting cross-validation MSEs are shown in Figure 8. We chose 2 components for the PLS model because it has a near-minimum cross-validation MSE value. Fitting the 2-component model on the test data results in a test MSE of 0.422. This is slightly less than that of the PCR model. Fitting this model on the full data set we observed that it explains 40.77% of the variance in the predictor variables and 34.83% of the variance in the response variable (quality). This is also a slight improvement over the PCR model for explanation of variance in the response variable, even though the PLS model uses only 2 components instead of 3.

### Regression Trees

In this section the regression tree method is used to analyze the data.

The following Figure 9 left shows a regression tree fit to the red wine data. The output tree is with 15 terminal nodes with test MSE 0.4998054.

|  |  |
| --- | --- |
|  |  |
| Figure 9. *Left*: A regression tree for predicting the quality based on other predictors.   *Right* : Cross-validation size vs. dev. | |

Tree method is useful for interpretation. Based on Figure 9 left, we could interpret the regression tree as follows: Alcohol is the most important factor in determining quality; and sulphates and volatile.acidity are also important. For example, the 2 most left leaf nodes indicate that the wines with alcohol < 10.45 and sulphates < 0.685 and volatile.acidity < 0.75, the predicted quality is 5. We used cross-validation (Figure 9 right) and tried pruning the tree to 12 terminal nodes, but pruning doesn’t improve the performance (i.e. test MSE) in this case.

#### Bagging

Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance.

When applying bagging to the regression trees, the test MSE is 0.363. We see that the bagging test error rate is lower than the test error rate obtained from a single tree.

#### Random Forest

Random forests build decision trees on bootstrapped training samples similar to bagging, but the split is allowed to use only one of *m* predictors randomly sampled at each split. Applying Random Forest to the regression tree with mtry(*m*)=4 (the closest value of p/3 where p=11), the test MSE is 0.359. It is slightly better than the result of bagging.

#### **Boosting**

In boosting method, the trees are grown sequentially: each tree is grown using information from previously grown trees by fitting on a modified version of the original data set. With some probing, we use shrinkage parameter λ = 0.001 and interaction depth d = 5 and the number of trees B = 4863 to apply boosting to the regression tree, the resulting test MSE is 0.3840237. The result is not as good as the random forest method nor bagging in this case, but better than the single tree.

#### **Summary**

Random Forest has the best result among the regression trees with test MSE = 0.359. All the tree-based methods show the most 3 important variables are alcohol, sulphates, volatile.acidity.

### Generalized Additive Models

Here we use GAM to apply non-linear functions of alcohol, sulphates, and volatile.acidity (The 3 most important variables indicated by subset selection and tree-based methods). One possible best test MSE is 0.408 with 5 degrees of freedom for sulphates, 4 degrees of freedom for volatile.acidity and linear function of year. The nice thing about GAM is we can examine the effect of alcohol, sulphates, and volatile.acidity on quality individually. Figure 10 gives us useful representation for inference.

|  |
| --- |
|  |
| Figure 10. Plots of the relationship between the predictors alcohol, sulphates, volatile.acidity (from left to right) and the response quality in the fitted GAM model. Each plot displays the fitted function and pointwise standard errors. The right two functions are smoothing splines in year and age, with 5 and 4 degree of freedom |

## Classification Methods

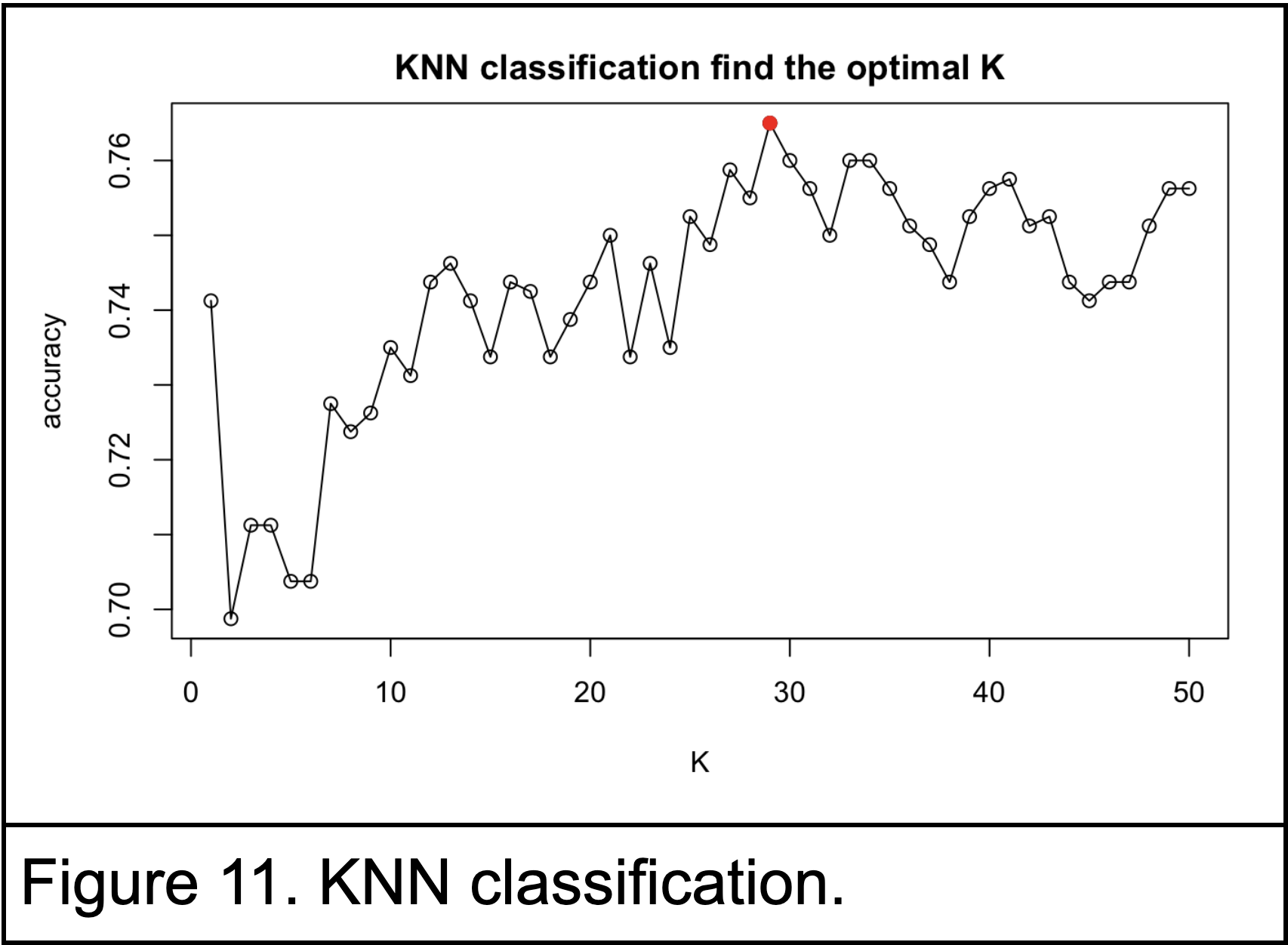
### Logistic Regression

We fit the logistic regression to the dataset and the accuracy of the prediction is 0.73.

### Linear Discriminant Analysis

LDA follows the classification rule that is based on Bayes theorem. Each class is normally distributed with different means but same variance. We fit the train data to the LDA model and predict the testing data using the model. And the accuracy of the prediction is 0.7675.

### **Quadratic Discriminant Analysis**

QDA has the same classification rule as LDA. However, each class has different covariance matrices. We fit the train data to the LDA model and predict the testing data using the model. And the accuracy of the prediction is 0.729.

### **K-nearest Neighbors**

We use KNN in both classification and regression. As KNN is a distance-based model, we normalized all the predictors data in order to have them in the same scale. We apply the KNN model on the train dataset using different K and get accuracy of the prediction of testing data in the classification model. We can find in the plot that when K=29 (Figure 11), we have the highest accuracy 76.5%.

### Classification Trees

We fit the classification tree to the red wine data. The result plot is as below (Figure 12 left) with correct prediction 70.625% for the test data set. From the plot we can see that “alcohol” is the most important indicator, and then sulphates.

|  |
| --- |
|  |
| Figure 12. *Left:* A classification tree for predicting the quality class based on other predictors (1 High/0 Low).   *Right*: Tree after pruning to 8 terminal nodes |

We used cross-validation and tried pruning to 8 terminal nodes (Figure 12. right), the resulting correct prediction accuracy is the same as 12 terminal nodes.

Similar to what we mentioned in the regression tree section, the classification tree is very useful to interpret the data. We can predict the class of quality by just using some of the predictors to achieve 70.625% correct prediction in this case.

#### Bagging

Applying bagging to the regression tree, the correct prediction is 78.375% for the test data set.

#### Random Forest

Applying random forest with mtry=4 to the regression tree, the correct prediction is 79.5% for the test data set. ( We also tried m=3, the result is slightly worse but still better than bagging. The correct prediction is 78.5%)

#### Boosting

Applying boosting to the regression tree, the correct prediction is 78.125% for the test data set with shrinkage parameter λ = 0.01 and interaction depth d = 3 and the number of trees B = 5000.

#### Summary

Random Forest has the best result among the classification trees with correct prediction 79.5%.

All the tree methods show that the most 3 important variables are alcohol, sulphates, volatile.acidity.

### Generalized Additive Models

We apply GAM for classification similar to what we did for regression, the prediction accuracy as 72.86%.

## Unsupervised Methods

### Principal Components Analysis

Principal components analysis is an unsupervised learning method, which uses only a set of features without any response variable. Here, we can apply it as a tool for visualizing the wine quality data. The PCA method was applied to a data set containing only the predictor variables, which were scaled to have standard deviation one. From the biplot in Figure 13 (left), we can see that some of the variables have the same orientation across the plot of the first two principal components. For instance, the “citric.acid”, “fixed.acidity”, and “sulphates” arrows are similar, as are the “total.sulfur.dioxide” and “free.sulfur.dioxide”, and the “chlorides” and “density”. This indicates that there may be some correlation between those variables, which could be explored further.

|  |
| --- |
|  |
| Figure 13. *Left*: The first two principal components with arrows scaled to represent loadings. *Middle*: scree plot; the proportion of variance explained by each principal component. *Right*: the cumulative proportion of variance explained by the principal components. |

From the plot of the cumulative proportion of variance explained by each component (Figure 13), we can see that the contribution of each additional component quickly diminishes after the first 3-5 components, after which each component adds only about 5% or less. Finally, we can plot each observation as a projection onto the first three principal components, and color the points by the quality score. As shown in Figure 14, this allows us to visualize some separation in the data for certain score values. For instance, the observations with scores of 7 (blue) or 8 (pink) (the two highest scores assigned) appear to cluster mostly in the lower half of the points in each plot; and the observations with scores of 3 (red) (the lowest score assigned) appear only in the far upper portion of the plot. This shows that principal components are able to separate the data somewhat, although the separation is far from perfect, at least across the first three principal components.

|  |
| --- |
|  |
| Figure 14. Projections of the wine observations onto the first three principal components. |

### K-Means Clustering

In k-means clustering, the observations are assigned to a pre-specified number of clusters. When clustering, we’ll use multiple values of K, since it’s not possible to pre-determine the optimal value of K. We’ll first cluster the observations based on the scaled predictor variables, which should illustrate any subgroups in the observations. Shown in Figure 15 left is an example table of the clusters from K=8 compared to the observations grouped by quality. From this table, we see that the observations with quality 7 and 8 (the highest assigned quality values) tend to fall into clusters 4, 5, and 6; while quality 3, 4, and 5 (the lower assigned quality values) tend to call into clusters 2, 7, and 8. There is still overlap between these clusters, but this indicates that the observations will be able to at least partially separate the data by quality.

|  |  |
| --- | --- |
|  |  |
| Figure 15. *Left*: K-means clustering of the observations with K=8. Quality values as column names, and cluster ids as row names. *Right*: K-means cluster assignments for K=2 to K=10, when clustering variables based on the observations. | |

Next, we can transpose (and scale) the data and cluster the variables based on the observations, illustrating any subgroups in the variables. The resulting cluster assignments are shown in Figure 15 right. This can be used as part of an exploratory analysis to build hypotheses about any predictor variables that are potentially correlated. We see that the “total.sulfur.dioxide” always clusters separately from all other variables, indicating that it contains different information. The “free.sulfur.dioxide” also tends to cluster separately from the other variables. The following sets of variables tend to cluster together, indicating that they may be correlated: “residual.sugar” and “pH”; “citric.acid” and “chlorides”; “volatile.acidity” and “sulphates”.

### Hierarchical Clustering

Hierarchical clustering is another method of clustering the data which does not require that we pre-specify a number of clusters. The three linkage methods that we applied for hierarchical clustering are Complete, Average, and Single, each using Euclidean distance and applied on the scaled data. We observed that single linkage results in trailing clusters and thus is not meaningfully clustering the data. Average linkage likewise does not seem to cluster the data very well, as it also results in clusters that are somewhat trailing, though to a lesser extent than single linkage. Complete linkage is the most effective method at splitting the data into clusters; the resulting dendrogram is shown in Figure 16.

|  |  |
| --- | --- |
|  | Figure 16. Hierarchical clustering with complete linkage and euclidean distance. |

## Conclusion

### Comparing Classification Models

Our primary goal was to identify the models with the best predictive power, so we’ll focus our comparison on the test set error rate each model produced. The left table shows each model’s test set mis-classification rate as well as the false positive (predicted high quality, actual low quality) and false negative rates. We included the false positive/negative rates in case extra measures were needed to differentiate models with the same mis-classification rate tie that would require other intuitive tie-breaker measures, but the table shows that the **random forest** model performed the best on the test set.

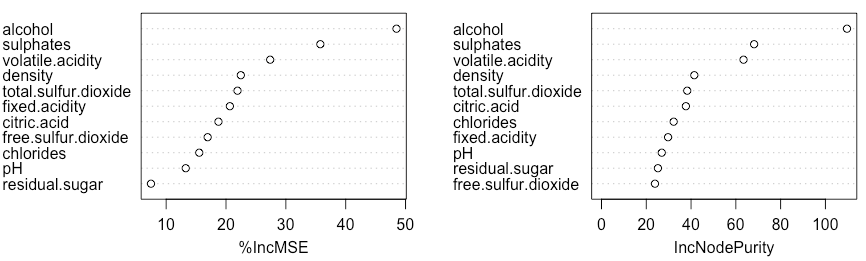
Since a random forest is non-parametric, we can’t provide vineyard analysts with any coefficient estimates. We can, however, point them towards the variables that our final model deems most important. That figure is in the next section.

### Comparing Regression Models

The random forest model also performed the best of the regression models we tested. The left table is the test MSE’s for each regression model we fit. We include the plots of variable importance rankings from the random forest regression model below.

As we found in our exploratory analysis and in other models like multiple regression, the most important chemical properties for vineyards to consider are alcohol, sulphates, and volatile\_acidity.

After this analysis, we have fairly compelling information to share with vineyards who are interested. Given the right data on a wine’s chemical properties, we can predict whether it’ll be rated as above- or below-average with some certainty (~75-80%). We’ve also gathered information on what generally leads to higher ratings. Domain knowledge might already have suggested more alcohol and less volatile acidity, but now we have evidence!



## Appendix

### R code

Link to R code for the report (group member initials are in front of the scripts they worked on): <https://github.com/dgdulaney1/st-563/tree/master/final-project>

### Figures

|  |  |
| --- | --- |
|  |  |
| Figure A.1. *Left:* Output of multiple regression fit using all predictors. *Right:* Output of multiple regression fit using only significant variables from the full model. | |

|  |
| --- |
|  |
| Figure A.2. Output of multiple regression fit with some interaction effects. |