White Wine Quality Exploration

STA141A Final Project Group16

Zhirun Qiu (zrqiu@ucdavis.edu)
Ziqi Dai (ziqdai@ucdavis.edu)
Tianyi Feng (tfeng@ucdavis.edu)
Junyao Lu (jyllu@ucdavis.edu)

1 Introduction

With the improvement of the quality of life, more and more people now like to drink wines. White wine is one of the most popular wines. It is known to benefit heart health and protect people from heart disease when consumed in moderation (Bhatnagar, 2019). The increasing demand for white wines simulates the demand for high-quality white wines. In order to gain a competitive advantage among all the white wine brands, it is inevitable for winemakers to understand the factors that may affect the taste of white wines and then improve the quality of white wines produced. This report aims to explore the impacts of the physicochemical factors on the quality of white wine.

To achieve the goal, we are going to exploit the white wine dataset from the UCI Machine Learning Repository in the following report. We will check the quality of the dataset and if there are any outliers, we will try to figure out why they are outliers. Then, we will examine trends and correlations in the data. After exploring the relationship between the physicochemical variables (inputs) and the quality of white wine (output), we will determine which physicochemical variables significantly impact the white wine quality. Finally, we will choose the statistically significant variables to build a regression model with the greatest accuracy. With the final regression model, we expect to understand which physicochemical variables contribute to high-quality white wines. Thus, the winemakers could adjust the corresponding ingredients in the production process to produce high-quality white wines.

2 Background

2.1 Dataset and Variables

We plan to use the dataset from the UCI Machine Learning Repository collected in 2009 by Paulo Corteza, Antonio Cerdeirab, Fernando Almeida, Telmo Matosb, and José Reisa. This dataset includes 4898 samples from the Portuguese "Vinho Verde" white wines. Each sample is tested for physicochemical properties at the official certification entity CVRVV and the sensory scores were evaluated by at least 3 sensory assessors using blind taste. There are 12 variables in the dataset, including 11 input variables collected from the laboratory tests and 1 output variable collected from average sensory scores.

- Input variables include: (Definitions are collected from the dataset description.)
  - Fixed acidity (g/dm^3): the acids in the wines that do not evaporate readily.
  - Volatile acidity (g/dm^3): the steam distillable acids in the wines; a high level of volatile acidity can lead to an unpleasant, vinegar taste.
  - Citric acid (g/dm^3): the acid that add freshness and flavor to the wines.
  - Residual sugar (g/dm^3): the sugar that remains in the wines after fermentation stops; in general, the amount is greater than 1 g/dm^3; the wines are considered to be sweet wines or dessert wines if the amount is more than 45 g/dm^3.
  - Chloride (g/dm^3): the salt in the wines.
  - Free sulfur dioxide (mg/dm^3): it prevents microbial growth and the oxidation of the wines.

  - Total sulfur dioxide (mg/dm^3): the combination of free sulfur dioxide and bound sulfur dioxide.
  - Density (g/cm^3): the density of wine is close to the density of water depending on the percent of alcohol and sugar.
  - pH: measurement of how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the scale.
  - Sulphates (g/dm3): a wine additive that acts as an antimicrobial and antioxidant.
  - Alcohol (vol%): the ingredient in the wines that causes drunkenness.

- The output variable is:
  - quality: a score ranging from 0 (poor) to 10 (excellent).

In order to fit the data into a logistic regression model, we decide to use the variable quality as a categorical variable and divide the data into two categories by a score of 6.5 according to the tips from the dataset description. If the quality is greater than 6.5, the sample quality is classified as good. If the quality is less than 6.5, the sample quality is classified as bad.

2.2 Questions of Interests

1. Are there any relationships between the physicochemical variables and the white wine quality? We would like to know whether the data is ideal and suitable to do white wine analysis.
2. How effective is the logistic regression model in describing and explaining the relationships between the physicochemical variables and the white wine quality?
3. Build a predictive model for wine quality. According to the model, which variables influence the quality the most and the least?

3 Study Design and Methodology

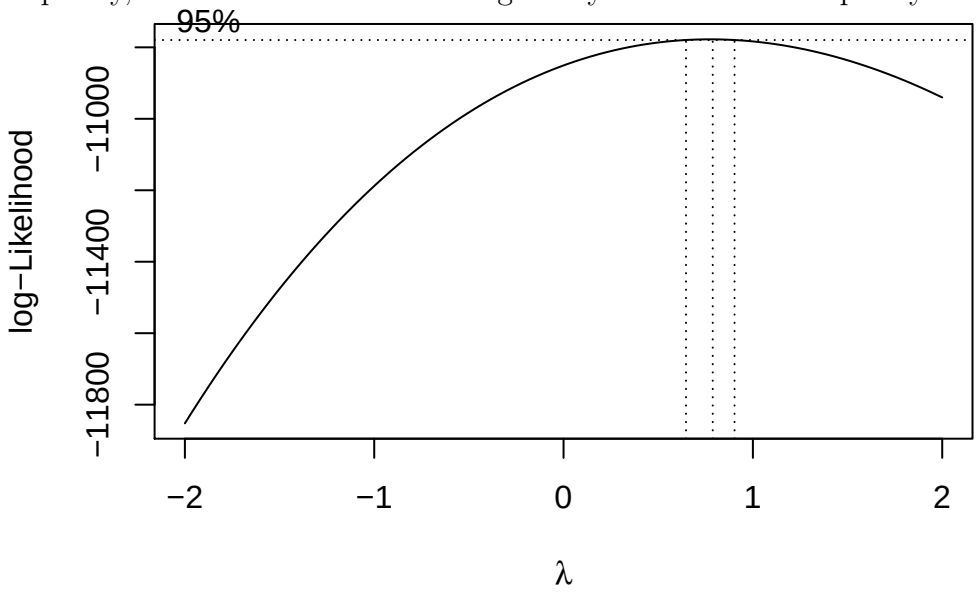
3.1 Data Analysis

First of all, we take a look at the general summary of the dataset. Based on the missingness map (Appendix A-1), there is no missing value in the dataset. We then notice that for some variables, the maximum is much larger than the minimum, but the mean is relatively closed to the median. In order to have a deeper understanding of the data distribution, we decide to find the skewness and kurtosis for each variable. All the physicochemical variables in our dataset are positively skewed. The variable chlorides have the greatest skewness of 5.0233, while the variables total sulfur dioxide and pH have the least skewness. The histograms of the physicochemical variables (Appendix A-2) show the same results. The kurtosis tells us that most of the physicochemical variables have outliers. The variable chlorides has the largest number of outliers, and alcohol has the smallest number of outliers. The right skewness and a large number of outliers in the dataset are noticeable. We consider to use transformations to transform highly skewed variables into more normalized variables when modeling the data. It is possible that the existence of some outliers is not due to errors,

so we decide to keep them for further analysis before deciding whether to remove them or not.

For the output variable quality, we observe that there are no samples with a score of 0, 1, 2, or 10. The distribution is roughly normal.

Then, we compute the correlations between the physicochemical variables and quality. We find that variables pH, sulphates, alcohol, and free sulfur dioxide are all positively correlated with quality, and other variables are negatively correlated with quality.



The box-cox test shows that the optimal λ values is slightly lower than 1. But 1 is not within the confidence interval of 95% significance level, so a transformation is performed with λ equals to 0.7879.

3.2 Classification

Based on the analysis above, the different effects of variables on white wine quality are not so obvious. The classification may make the difference clearer. Thus, we decide to classify the data into two categories, just like what we mentioned before. If the quality is larger than 6.5, the sample is identified as good quality; otherwise, it is identified as a bad quality.

Based on the barplot (Appendix A-3) of the variable quality after classification, we can see that the number of bad quality wines is much larger than that of good quality wines. This information suggests that the data is likely to be biased. Besides, the summary statistics confirm our guess. In the summary, there are 3838 bad quality wine samples and 1060 good quality wine samples. The bias of the variable quality can be one of the reason why there are so many outliers we observe in the previous analysis. Thus, we cannot directly remove the outliers. One possible approach of solving the problem is to use a stratified method. We can split the data into training and testing data, and then we control the number of good and bad quality observations constant in the training data. This method can also help to improve the biased data.

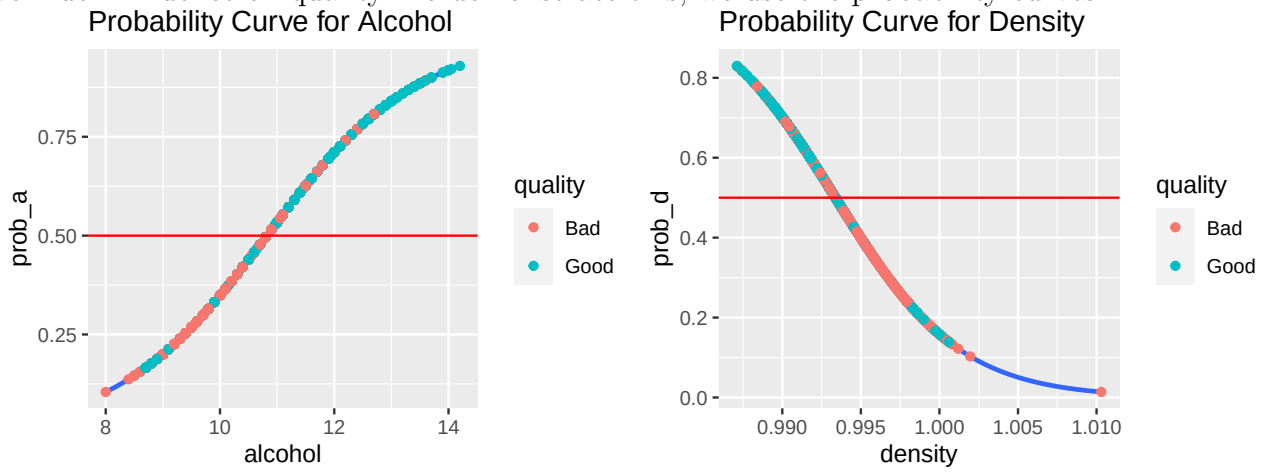
Before creating a stratified sampling to reduce the bias, we need to balance the data

when choosing the sample data because we need to make sure the proportion of good and bad quality in both train and test data should be equal. Then, we choose to randomly select the data of bad quality wines based on the number of good quality wines. In the sample data, we pick 80% of both good and bad quality observations, that is a total of 2,120 observations. 80% of the observations are in the train data, while the remaining 20% are in the test data. Therefore, in the sample data, we have obtained 1,060 good quality wine data and 1,060 bad quality wine data.

Then, from the summary of the variables in the training and testing datasets, we observe that the ranges between the minimum value and maximum value of residual sugar, chlorides, free sulfur dioxide, and total sulfur dioxide become much smaller compared to the ranges from the original dataset.

After solving the potential problem of biased data, we draw the boxplots (Appendix A-4) between the 11 independent variables and quality to see if different effects of variables to the wine quality become more obvious and whether there are still outliers in the dataset. From the boxplots, we observe that variables fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, and density are negatively correlated with the variable quality. For variables such as pH, sulfate and alcohol, they have a positive correlation with the quality of the variable.

Moreover, we find that different data distribution is obvious in the graph about variables like density and alcohol. The distinct data distribution points out that the two variables can have much influence on quality. To demonstrate this, we use the probability curves:



In the above two probability curves, the binomial generalized linear models are successful, which means that they have a great influence on the quality.

In addition, there are too many outliers for the variable chlorides. Through the above method, we have eliminated the negative impact of biased data, so the outlier shows that there may be some errors in the variable chlorides. It is better to remove the variable, but we need to look at the p-value of the variable later in the model to check whether it is significant or not.

4 Clustering Analysis

4.1 K-means (Appendix A-5)

The first method used to identify the clusters in the dataset is k-means. A set of 10 numbers as the parameter k is tried. From the within SS plot and between SS plot, we can see that the biggest change in both of them is when the number of clusters rises from 1 to 2. This indicated that generally the data set is concentrated in one crowd.

To further understand the shape of the k-means clusters, the frequency plot and the scatter plot of the first 2 principal components are drawn. From those figures, we can see that although all the observations are distributed roughly in a crowd, the density of the crowd is not isotropic. The frequency of every cluster in the right of each plot is always lower. Also, there is a spike in the first 2 component plot with relatively low density, which means that describe the shape of the data set as a hyper spindle is more appropriate than a hypersphere. Moreover, as the first component is mainly contributed by total sulfur dioxide, the spikes of the spindle should point approximately in the axis of the variable.

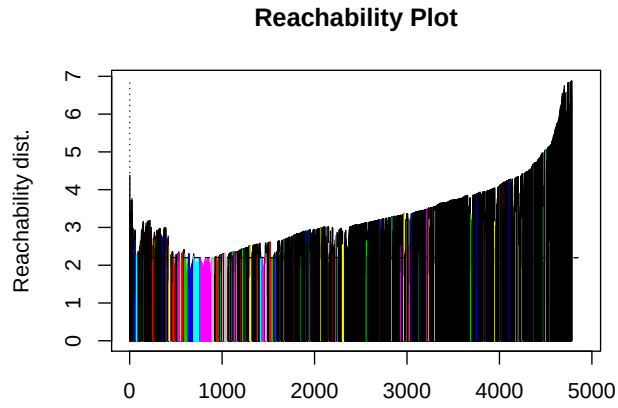
4.2 OPTICS

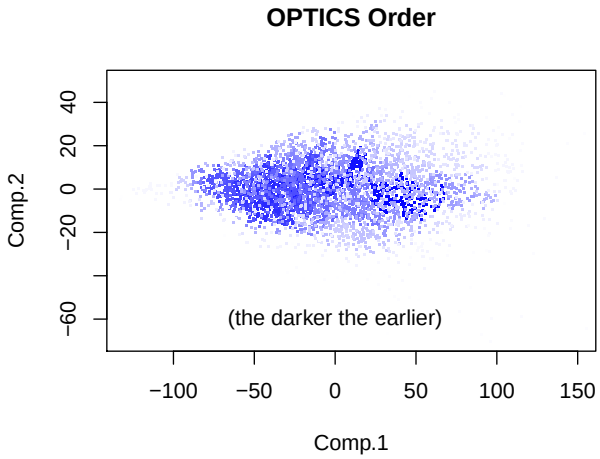
OPTICS is an abbreviation for Ordering Procedure To Identify Cluster Structures. It is an advanced version of the density-based clustering algorithm DBSCAN. In this algorithm, a core point is defined a point that has at least p points in its neighborhood where p is an user input parameter. The reachable distance is the smaller one between the minimum distance that makes a point a core point and the distance between the point and a core point in its neighborhood. This algorithm takes each points in a order that in each iteration the point with the lowest reachable distance among those points that have not been proceed is added to the cluster. When the algorithm is done, we can either plot a reachable plot that represent clusters by valleys or choose a certain eps value and extract a ordinary DBSCAN object from it.

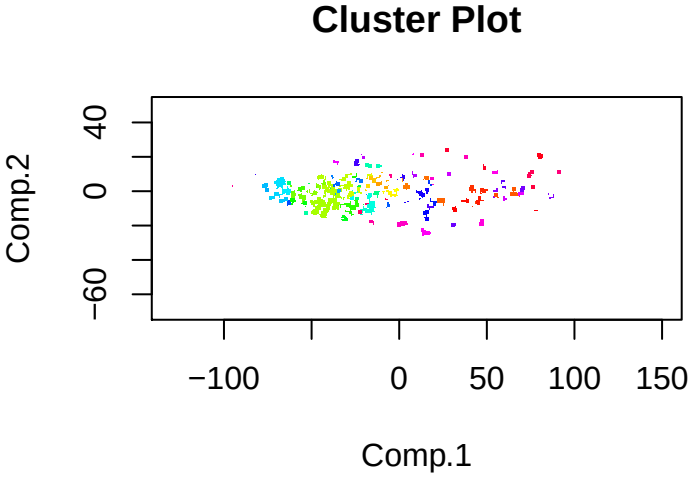
In order to identify the sophisticated structure of the clusters in the dataset, the OPTICS method is implemented. The eps value is set to 5 to calculate the reachable distances of all the observations. Then, a DBSCAN cluster label is subtracted with a threshold of 2.2.

From the OPTICS order plot, we can see that the reachability distance is gradually increasing without obvious valleys. After some initial observations with high reachability, the algorithm locates the dense part of the dataset and the largest clusters identified by DBSCAN appear here. The OPTICS order plot with the first principal components clearly shows this process. The initial steps are to the right and the densest part is to the right.

After that, as the order increases, the reachability distance almost increases without a drop. This shows that the general structure of the dataset is a cluster. However, the algorithm can indeed identify many locally dense clusters. Those clusters are generally small in size and never appear in the periphery part of the dataset. The cluster plot reveals all the clusters identified by DBSCAN and supports the conclusion.





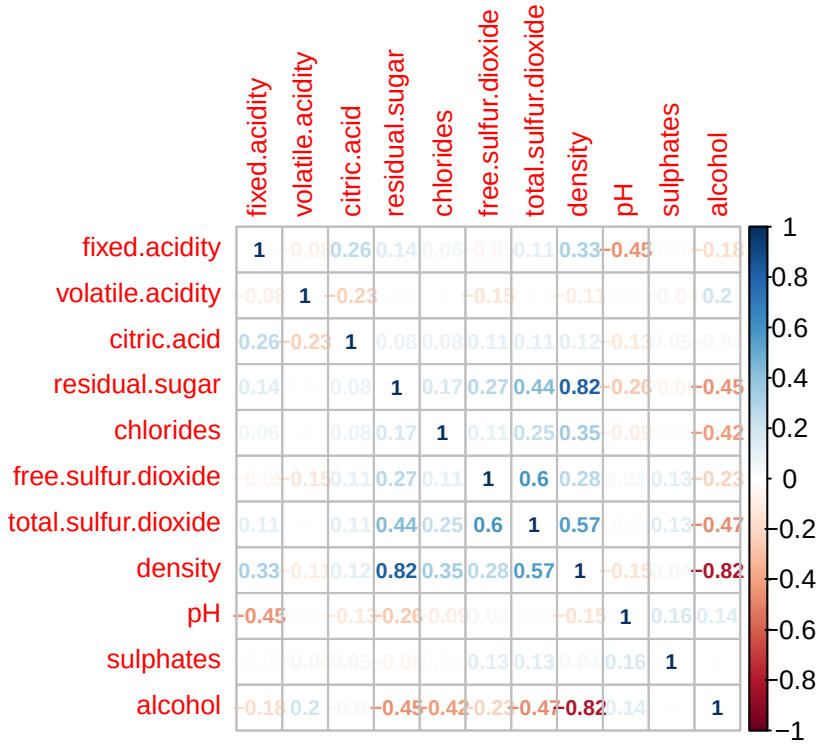


5 Results and Analysis

5.1 Model selection

The correlation matrix below tells us:

- Density is highly correlated to residual sugar (positive) and alcohol (negative).
- Free sulfur dioxide and total sulfur dioxide are highly positively correlated.
- Fixed acidity, volatile acidity, and citric acid have negative relationships with pH, which demonstrates the principle that the pH of acid will be smaller.



Therefore, we may need to delete the variable density, and then choose one between free sulfur dioxide and total sulfur dioxide to prevent multicollinearity when building the model later.

After classifying the data, the variable quality becomes a categorical variable. Therefore, it is better to use the glm model instead of the lm model we mentioned in the proposal. We first use the training data to build a model with all variables.

$$\begin{aligned}\text{quality} = {} & 836.57 + 0.68(\text{fixed. acidity}) - 5.10(\text{volatile. acidity}) - 0.84(\text{citric. acid}) + \\ & 0.39(\text{residual. sugar}) - 7.74(\text{chlorides}) + 0.00(\text{free. sulfur. dioxide}) + 0.00(\text{total. sulfur. dioxide}) - \\ & 860.76(\text{density}) + 3.86(\text{pH}) + 2.34(\text{sulphates}) - 0.03(\text{alcohol}) \ + \epsilon\end{aligned}$$

	Predictied	
True	Bad	Good
Bad	149	66
Good	50	159

	Predicted	
True	Bad	Good
Bad	0.6930233	0.3069767
Good	0.2392344	0.7607656

According to the tables, we conclude that the model can accurately predict 308 values out of 424 observations. The misclassification error rate of good quality is 30.70% and the

rate of bad quality is 23.92%. The overall accuracy of the model is 72.64%.

In the summary of this model, we find that the p-value of the variable chlorides is greater than 0.05, so we fail to reject the null hypothesis, and this variable is not statistically significant. We should exclude it from the model, which proves our previous guess. Then, we need to delete the variables in the new model. In addition, we notice that the p-values of citric acid, free sulfur dioxide, total sulfur dioxide, and alcohol are all larger than 0.05, we can remove citric acid, but for the other three variables, due to the multicollinearity problem, we cannot directly remove them. We should first check the importance of each variable.

We build the model without chlorides and citric acid. After creating the new model, we get the tables below. We find that the misclassification error rate of good quality is 30.70%, and the misclassification error rate of bad quality is 23.92%. The correct prediction rate is 72.64%. These values hardly changed, which indicates that the variables chloride and citric acid are not important.

	Predicted	
True	Bad	Good
Bad	149	66
Good	50	159

	Predicted	
True	Bad	Good
Bad	0.6930233	0.3069767
Good	0.2392344	0.7607656

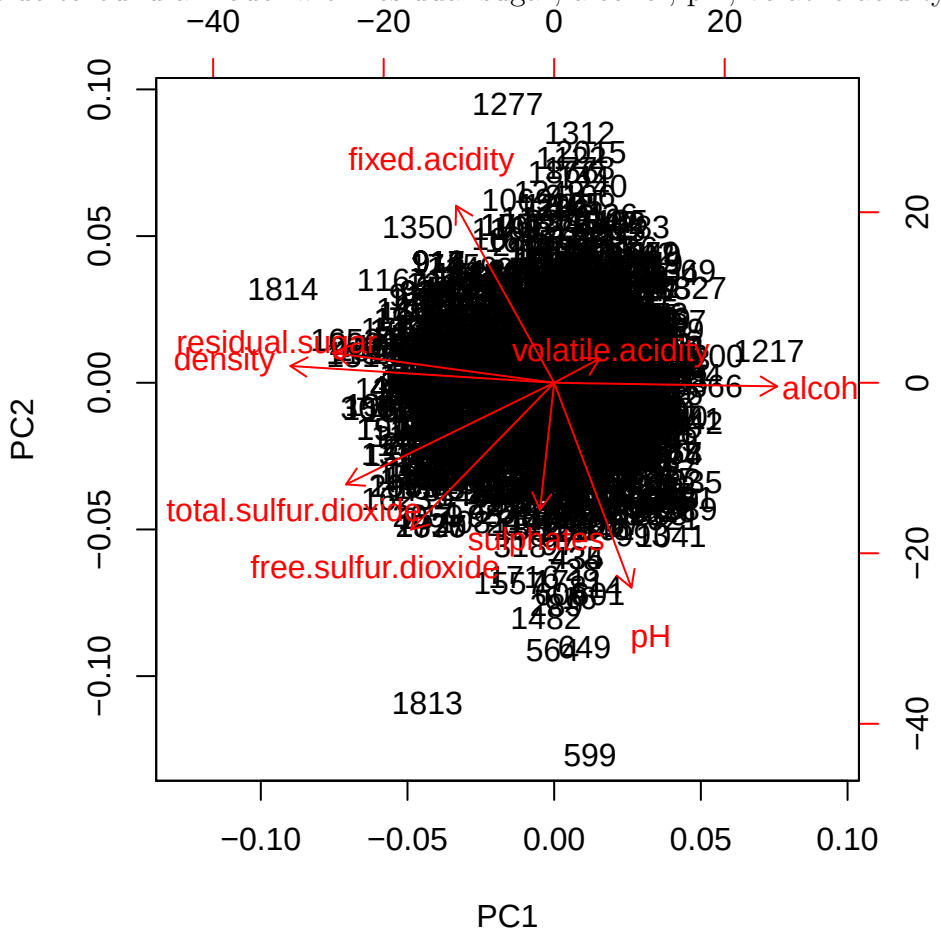
Then, we want to do the model selection to check which variable we can drop based on AIC and BIC. The final model obtained based on AIC and BIC is the same. This may be due to the small size of the predictive variable set. BIC gives more punishment to large models. But when the full model is small, there is little difference between the criteria. These models tell us that the best model based on these criteria also does not include chlorides. However, the correlation matrix shows that the problem of multicollinearity still exists in this model. Thus, we need to find a significant variable and use them to rebuild a model.

	(Bad)0	(Good)1
fixed.acidity	28.77450	28.77450
volatile.acidity	39.04025	39.04025
residual.sugar	27.42128	27.42128
free.sulfur.dioxide	29.66970	29.66970
total.sulfur.dioxide	27.57055	27.57055
density	28.02091	28.02091
pH	30.09542	30.09542
sulphates	24.87078	24.87078
alcohol	52.01239	52.01239

We use randomForest and the function varImp to find the variable importance for re-

gression. We find that three variables are most important to the regression: volatile acidity, alcohol, and pH. Based on the above findings, the high correlation between density and alcohol and residual sugar makes us believe that it is better to remove the variable density. Then, we use the model selection again to find the best model based on AIC and BIC if removing the variable density. The final models given by AIC and BIC are different, and BIC penalizes larger models more. The differences between the two models are free sulfur dioxide, total sulfur dioxide, and sulphates.

Furthermore, we apply principal component analysis to reduce the dimensionality of our data. Based on the biplot, we find that the first loading vector places roughly equal large weight on density, residual sugar, and alcohol with a small weight on other variables. The second loading vector places most of its weight on pH. The third loading vector places most of its weight on volatile acidity. The fourth loading vector places most of its weight on sulphates. Thus, we guess that residual sugar, alcohol, pH, volatile acidity, and sulphates are important variables. For free sulfur dioxide and total sulfur dioxide, they are not so significant in the random forest test and PCA. In addition, their coefficient values are almost zero. Thus, we decide to build a model with residual sugar, alcohol, pH, volatile acidity, and sulphates.



$$\text{quality} = -14.14 - 5.24(\text{volatile. acidity}) + 0.08(\text{residual. sugar}) + 1.16(\text{alcohol}) + 1.16(\text{pH}) + 1.15(\text{sulphates}) + \epsilon$$

In this model, the misclassification rate for bad quality is 34.42% and the misclassification

rate for good quality is 28.23%. The overall correct prediction rate is 68.63% and the AUC is 0.758. See Appendix A-6 for the ROC curve for this model.

Then, we use LDA method and KNN method to determine which method is effective to predict the classification of the white wine quality. With glm, the correct prediction rate is 68.63%; with LDA, the rate is 68.87%; with KNN when k = 1, the rate is 73.82%; with KNN when k = 5, the rate is 72.88%. It seems that the KNN method is better to show the accuracy of the model. And the result shows that with the model, we can only ensure that around 70% of the data can be correctly explained by the model. See Appendix A-7 for the ROC curve for LDA.

6 Conclusion

In conclusion, the data here is not ideal. We use Box-Cox to transform the data so that we can achieve the requirements of building models. In addition, most variables have outliers, and the data is biased due to much higher number of bad quality wines than that of good quality wines. Thus, we do the classification: choosing the sample data, and picking 80% of the observations in the train data while the remaining 20% in the test data. To improve the data, we finally remove the variable chloride which has too many outliers.

There are some kind of relationships between the physicochemical variables and the white wine quality. However, the relationships are not significant for some of the physicochemical variables, such as fixed acidity, citric acid and chlorides. Based on our analysis, the best model we get is about the relationship between quality and five variables, volatile.acidity, residual.sugar, pH, sulphates and alcohol.

$$\begin{aligned}\text{quality} = &-14.14 - 5.24(\text{volatile. acidity}) + 0.08(\text{residual. sugar}) + 1.16(\text{alcohol}) + \\ &1.16(\text{pH}) + 1.15(\text{sulphates}) + \epsilon\end{aligned}$$

According to the model, the variable volatile acidity influences the quality the most, and the variable residual sugar influences the quality the least. There are 73.82% of the variations (based on KNN when k=1) in the white wine quality are explained by the best model, which is relatively accepted. The failure to explain more variations by the final model could due to the sensory assessors have different tastes in white wines, and the ratings are based on their subjective preferences. Another reason could be the quality is affected by other variables, such as the types of the grapes used, the local temperature, and so on. Thus, more information and study are required to make a further conclusion. In order to provide the winemakers more accurate and professional suggestions, further investigations are required. One possible approach is to explore the white wines produced from other regions, as we only focus on the Portuguese "Vinho Verde" white wines in our analysis. Also, since there are a lot of types of white wines, we can consider to group the white wines based on their types in the further study.

7 Contributions

Zhirun Qiu: Data Analysis and Model Selection
Ziqi Dai: Data Summary and Visualization
Tianyi Feng: Clustering Analysis and Proofreading
Junyao Lu: Formatting and Reporting

8 References

1. Ankerst, Mihael, et al. "OPTICS: ordering points to identify the clustering structure." ACM Sigmod record 28.2 (1999): 49-60

2. Bhatnagar, Shubham. "Red Wine Or White Wine: Which Is Better For Your Health?" *NDTV Food*, 19 Mar. 2019, food.ndtv.com/food-drinks/red-wine-or-white-wine-which-is-better-for-your-health-1834678.

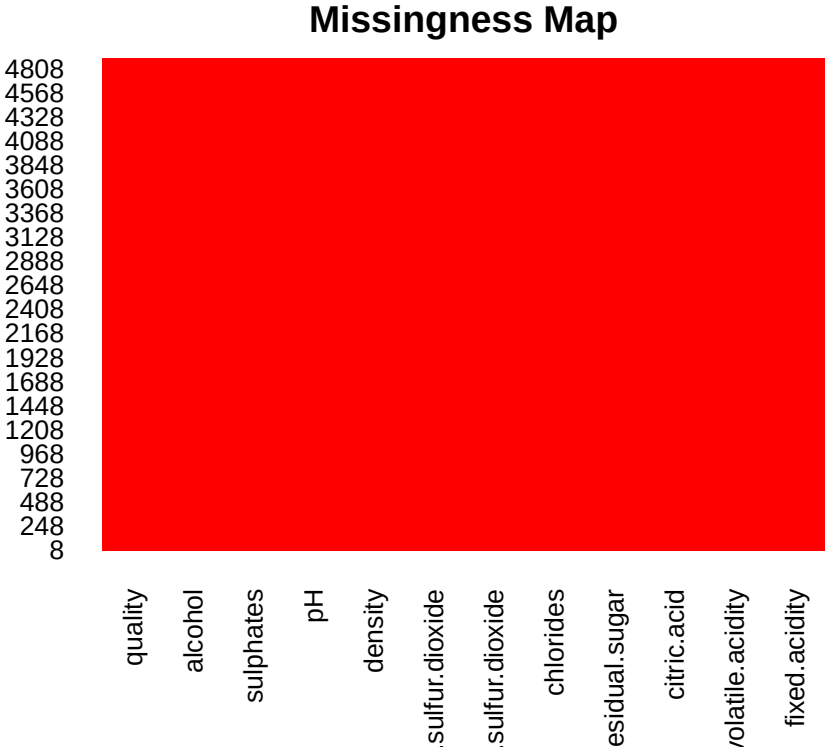
3. P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

4. "Wine Quality Data Set." *UCI Machine Learning Repository*, archive.ics.uci.edu/ml/datasets/Wine+Quality.

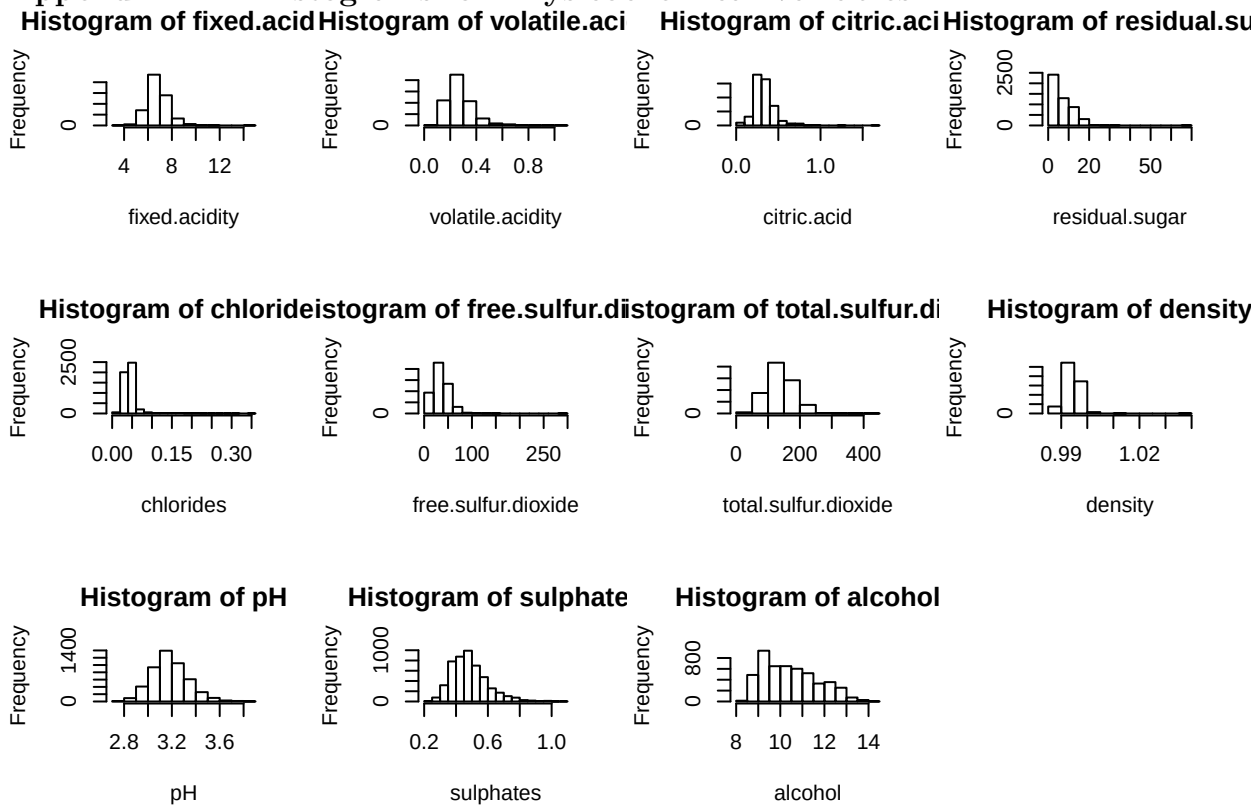
9 Appendices

9.1 Appendix A: Figure and Plots

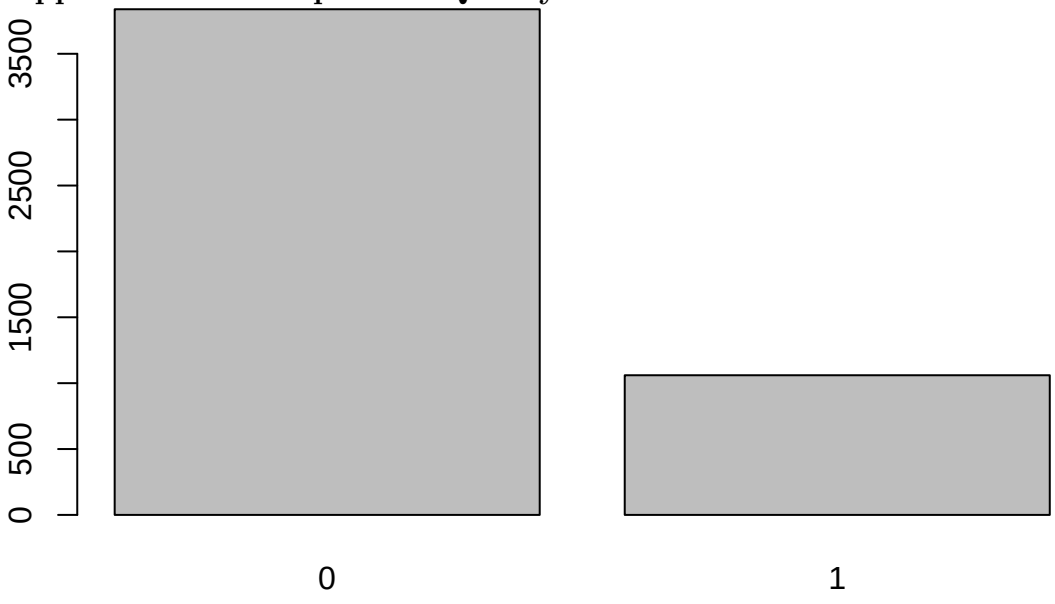
Appendix A-1: Missingness Map



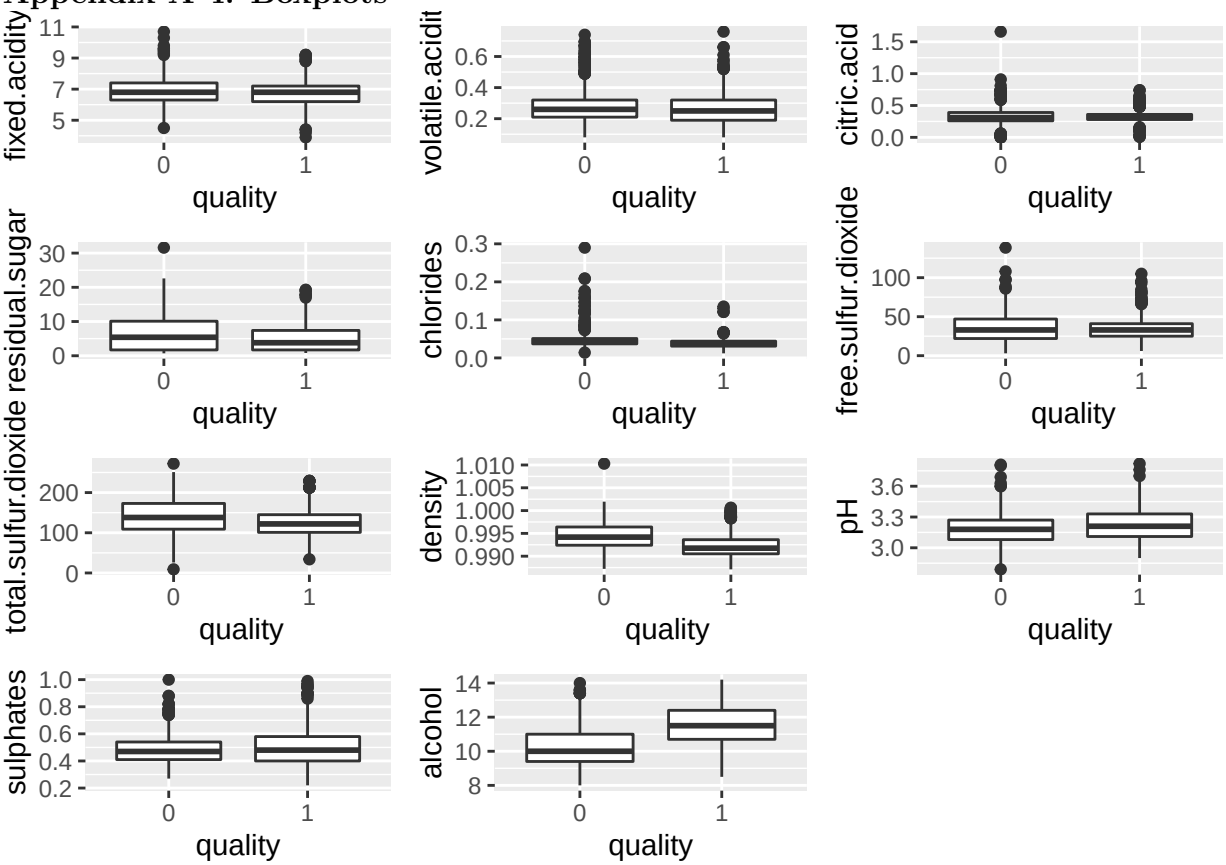
Appendix A-2: Histograms for Physicochemical Variables



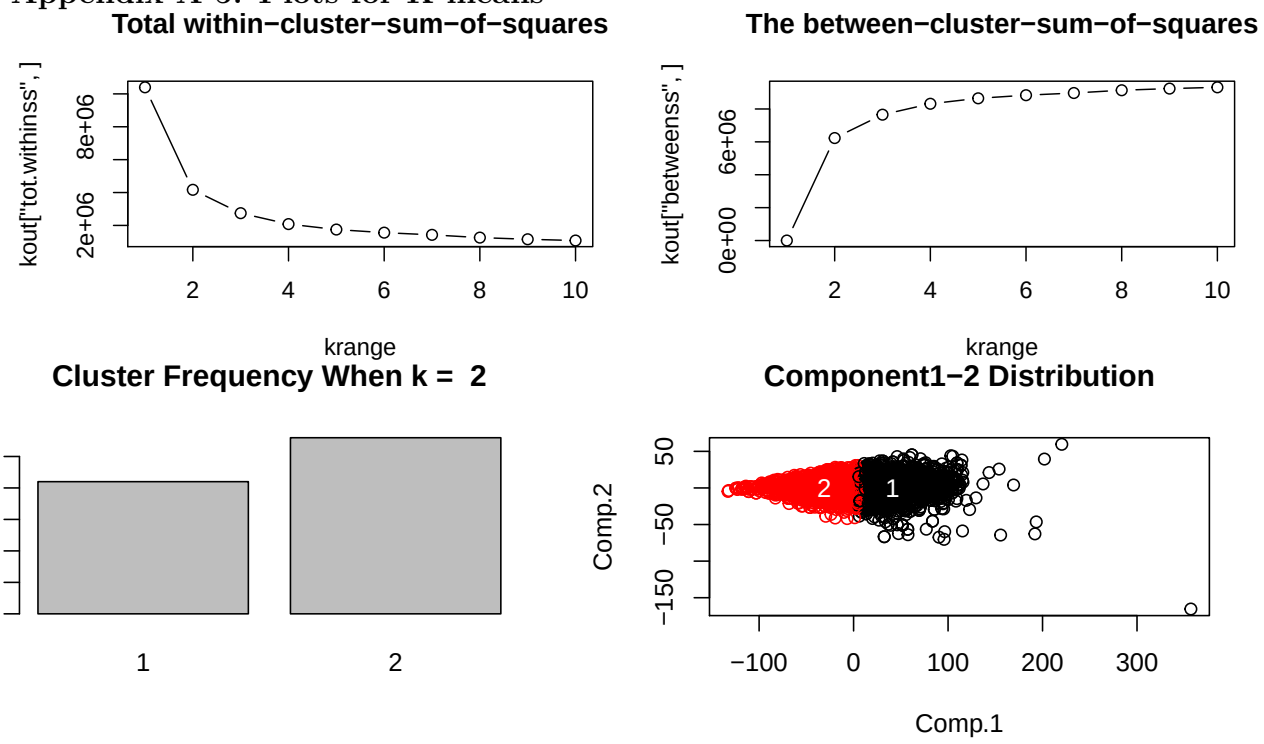
Appendix A-3: Barplot for Quality

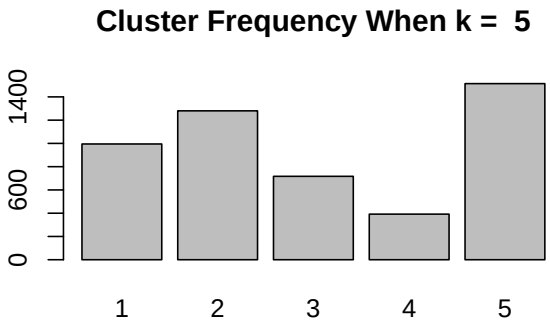


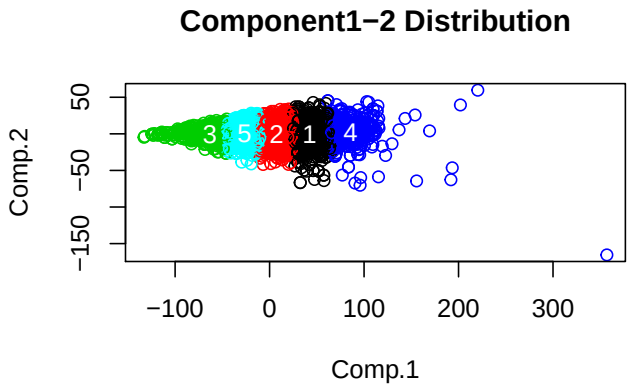
Appendix A-4: Boxplots

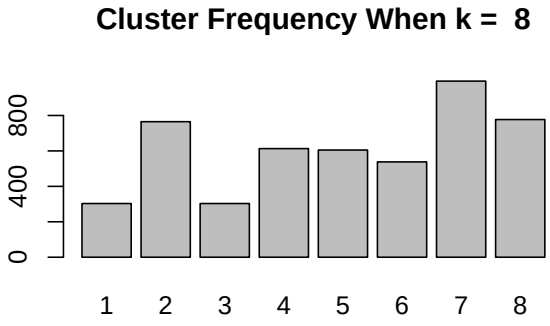


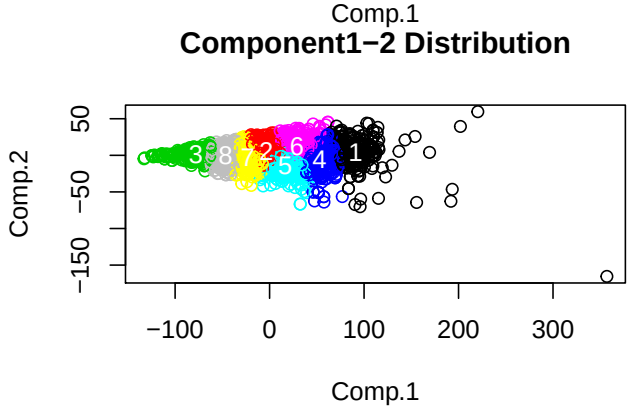
Appendix A-5: Plots for K-means



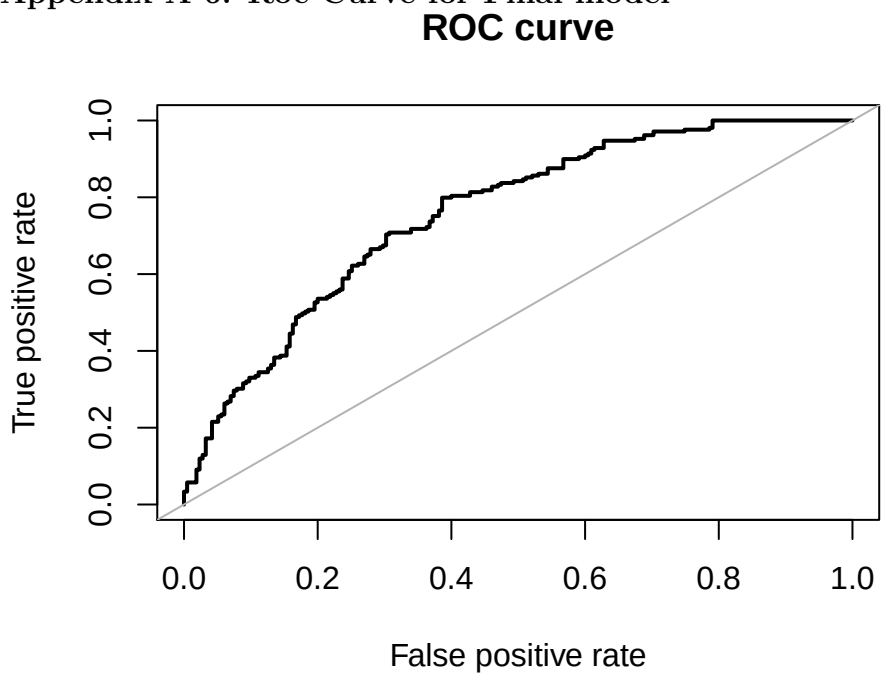




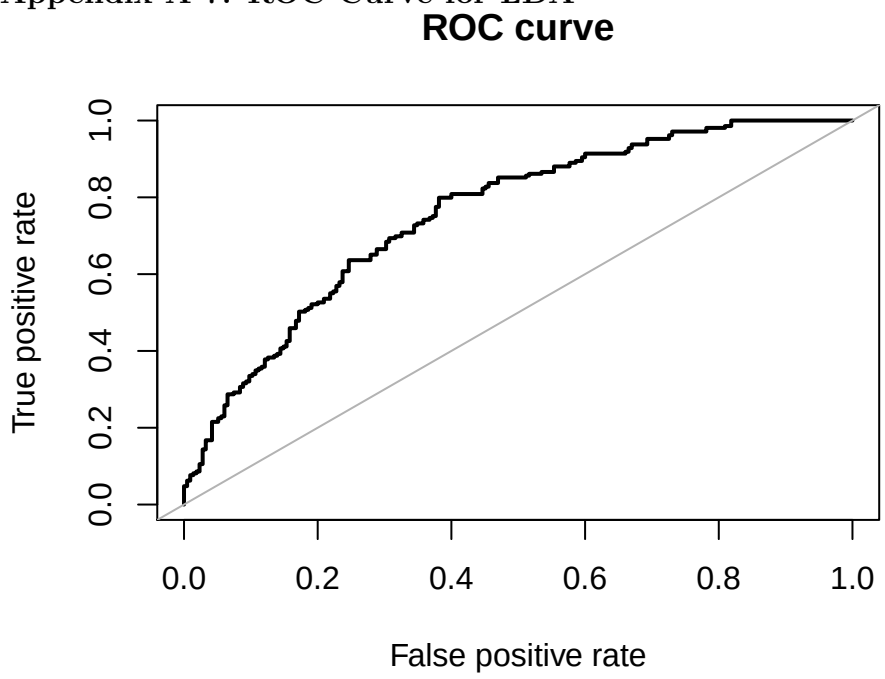




Appendix A-6: Roc Curve for Final model



Appendix A-7: ROC Curve for LDA



9.2 Appendix B: Codes

```
# Header-----------------------------------------------------------------
rm(list=ls())
graphics.off()
cat("\014")
set.seed(1)
# Data-------------------------------------------------------------------
df = read.csv("winequality-white.csv", header = TRUE, sep = ";")
str(df)
summary(df)
library(Amelia)
library(mlbench)
missmap(df, col = c("blue", "red"), legend = FALSE)
n = dim(df)[1]
# Exploratory Analysis---------------------------------------------------
skewness_f <- function(x)
  {
  x = as.numeric(x)
  m <- mean(((x-mean(x))/sd(x)/(n-1)*n)^3)
  return(m)
}
t(t(apply(df,2,skewness_f)))
kurtosis_f <- function(x)
```

```
  {
  x = as.numeric(x)
  m <- mean(((x-mean(x))/sd(x)/(n-1)*n)^4)-3
  return(m)
}
t(t(apply(df,2,kurtosis_f)))
library(ggplot2)
qplot(quality, data=df) +
  scale_x_continuous(breaks = seq(3, 9, 1), lim = c(3, 9))
par(mfrow=c(3,4))
for (i in 1: (length(df)-1)){
  hist(x = df[[i]],
       main = sprintf('Histogram of %s', colnames(df[i])),
       xlab = colnames(df[i]))
}
corr <- cor(df[,1:12])
corr[12,]
# Box-Cox-----------------------------------------------------------------
library(MASS)
bc = boxcox(quality~.,data = df)
lambda = bc$x[which.max(bc$y)]
df$quality = (df$quality^lambda-1)/lambda
# Classification----------------------------------------------------------
df$quality <- ifelse(df$quality > (6.5^lambda-1)/lambda, 1, 0)
df$quality <- factor(df$quality, levels=c(0,1))
summary(df)
quality <- table(df$quality)
barplot(quality)
# Train-Test Split--------------------------------------------------------
df_good <- df[which(df$quality == 1), ]
df_bad <- df[which(df$quality == 0), ]
n_good = dim(df_good)[1]
n_bad = dim(df_bad)[1]
df_bad = df_bad[sample(1:n_bad,n_good),]
df1 = rbind(df_good,df_bad)

n1 = 2*n_good
n1==dim(df1)[1] # should be equal
rownames(df1) = 1:n1

ratio = 0.8
split = sample(1:n1,n1*ratio)
df_train = df1[split, ]
df_test = df1[-split, ]
```

```r
summary(df_train)
summary(df_test)
# Boxplots-----------------------------------------------------------------
a <- ggplot(data = df_train,
            mapping = aes(x = quality, y = fixed.acidity,
                          group = quality)) +
  geom_boxplot()
b <- ggplot(data = df_train,
            mapping = aes(x = quality, y = volatile.acidity,
                          group = quality)) +
  geom_boxplot()
c <- ggplot(data = df_train,
            mapping = aes(x = quality, y = citric.acid,
                          group = quality)) +
  geom_boxplot()
d <- ggplot(data = df_train,
            mapping = aes(x = quality, y = residual.sugar,
                          group = quality)) +
  geom_boxplot()
e <- ggplot(data = df_train,
            mapping = aes(x = quality, y = chlorides,
                          group = quality)) +
  geom_boxplot()
f <- ggplot(data = df_train,
            mapping = aes(x = quality, y = free.sulfur.dioxide,
                          group = quality)) +
  geom_boxplot()
g <- ggplot(data = df_train,
            mapping = aes(x = quality, y = total.sulfur.dioxide,
                          group = quality)) +
  geom_boxplot()
h <- ggplot(data = df_train,
            mapping = aes(x = quality, y = density,
                          group = quality)) +
  geom_boxplot()
i <- ggplot(data = df_train,
            mapping = aes(x = quality, y = pH,
                          group = quality)) +
  geom_boxplot()
j <- ggplot(data = df_train,
            mapping = aes(x = quality, y = sulphates,
                          group = quality)) +
  geom_boxplot()
k <- ggplot(data = df_train,
```

```
            mapping = aes(x = quality, y = alcohol,
                          group = quality)) +
  geom_boxplot()
library(ggpubr)
appendix_a_4 <- ggarrange(a, b, c, d, e, f, g, h, i, j, k, ncol=3, nrow=4)
appendix_a_4
# Probability Curves-------------------------------------------------------
library(gridExtra)
logreg_a <- glm(quality ~ alcohol, data = df_train, family=binomial)
prob_a<-predict(logreg_a, type = "response")
plot1 = ggplot(data = df_train) +
    geom_smooth(mapping = aes (x = alcohol, y = prob_a),
                method = "glm", method.args = list(family = "binomial"),
                se = FALSE) +
    geom_point(mapping = aes (x = alcohol, y = prob_a, colour = quality)) +
    geom_hline(yintercept = .5, colour = "red") +
  scale_colour_discrete(labels = c("Bad", "Good")) +
  ggtitle("Probability Curve for Alcohol")
logreg_d <- glm(quality ~ density, data=df_train, family=binomial)
prob_d<-predict(logreg_d, type="response")
plot2 = ggplot(data = df_train) +
    geom_smooth(mapping = aes (x = density, y = prob_d),
                method = "glm", method.args = list(family = "binomial"),
                se = FALSE) +
    geom_point(mapping = aes (x = density, y = prob_d, colour = quality)) +
    geom_hline(yintercept = .5, colour = "red") +
  scale_colour_discrete(labels = c("Bad", "Good")) +
  ggtitle("Probability Curve for Density")
grid.arrange(plot1, plot2, ncol=2)
library(dbscan)
par(mfrow=c(1, 2))
pri = princomp(df[-12])
clu1 = optics(df[1:11],eps = 7)
clu1 = extractDBSCAN(clu1, eps_cl = 2.2)
plot(clu1)
plot(pri$scores[,1:2][clu1$order,],
     col = colorRampPalette(c("blue", "white"))(n),pch = '.',
     cex = 2,xlim = c(-130,150), ylim = c(-70,50),
     main = 'OPTICS Order')
text(0,-60,'(the darker the earlier)')
par(mfrow=c(1, 1))
plot(pri$scores[,1:2],
     col = c("#ffffff",
             rainbow(length(unique(clu1$cluster))-1))[clu1$cluster+1],
```

```
     cex = 2,pch = '.',
     xlim = c(-130,150), ylim = c(-70,50), main = 'Cluster Plot' )
# Cor Plot Matrix-----------------------------------------------------------
par(mfrow=c(1, 1))
corr1 <- cor(df_train[,1:11])
library(corrplot)
corrplot(corr1, method = "number", tl.cex = 0.8, number.cex = 0.7)
# GLM 1---------------------------------------------------------------------
logreg <- glm(quality ~ ., data = df_train, family = binomial)
summary(logreg)
contrasts(factor(df_train$quality))
prob<-predict(logreg, df_test, type = "response")
predicted<-ifelse(prob < .5, 0, 1)
cm <- table(df_test[,12], predicted, dnn = c("True", "Predicted"))
sum(diag(cm))/sum(cm)
rownames(cm) = c("Bad", "Good")
colnames(cm) = c("Bad", "Good")
prob = prop.table(cm,1)
rownames(prob) = c("Bad", "Good")
knitr::kable(cm)
knitr::kable(prob)
# GLM 2---------------------------------------------------------------------
logreg1 <- glm(quality ~ . -chlorides - citric.acid,
               data = df_train, family = binomial)
summary(logreg1)
prob1<-predict(logreg1, df_test, type = "response")
predicted1<-ifelse(prob1 < .5, 0, 1)
cm1 <- table(df_test[,12], predicted1, dnn = c("True", "Predicted"))
sum(diag(cm1))/sum(cm1)
rownames(cm1) = c("Bad", "Good")
colnames(cm1) = c("Bad", "Good")
prob1 = prop.table(cm1,1)
rownames(prob1) = c("Bad", "Good")
colnames(prob1) = c("Bad", "Good")
knitr::kable(cm1)
knitr::kable(prob1)
# Stepwise GLM--------------------------------------------------------------
library(MASS)
step(logreg1, direction = "both",trace = 0)
step(logreg1, direction = "both",trace = 0, k = log(n1*ratio))
# Variable Importance-------------------------------------------------------
library(caret)
library(randomForest)
regressor <- randomForest(quality ~ . - chlorides - citric.acid,
```

```
                          data= df_train, importance=TRUE)
knitr::kable(varImp(regressor, conditional=TRUE))
logreg_density = glm(quality ~ . - chlorides - citric.acid - density,
                     data = df_train, family = binomial)
step(logreg_density, direction = "both",trace = 0)
step(logreg_density, direction = "both",trace = 0,k = log(n1*ratio))
# PCA-----------------------------------------------------------------------
pc_train <- prcomp(df_train[,-c(3,5,12)], center = TRUE, scale. = TRUE)
summary(pc_train)
pc_train$rotation
biplot(pc_train)
# ROC -----------------------------------------------------------------------
logreg2 <- glm(quality ~ volatile.acidity + residual.sugar +
                 alcohol + pH + sulphates,
               data=df_train, family=binomial)
summary(logreg2)
prob2<-predict(logreg2, df_test, type = "response")
predicted2<-ifelse(prob2 < .5, 0, 1)
cm2 <- table(df_test[,12], predicted2)
cm2
prop.table(cm2,1)
sum(diag(cm2))/sum(cm2)
library(ROSE)
roc.curve(df_test$quality, prob2,n.thresholds = n)
# LDA-----------------------------------------------------------------------
library(MASS)
lda_fit <- lda(quality ~ volatile.acidity + residual.sugar +
                 alcohol + pH + sulphates, df_train)
lda_pred <- predict(lda_fit, df_test)
lda_cm <- table(df_test$quality, lda_pred$class,
                dnn = c("True", "Predicted"))
lda_cm
sum(diag(lda_cm))/sum(lda_cm)
roc.curve(df_test$quality, lda_pred$posterior[,2],n.thresholds = n)
# KNN-----------------------------------------------------------------------
library(class)
variable_train <- data.frame(df_train[, c(2,4,9,10,11)])
variable_test <- data.frame(df_test[, c(2,4,9,10,11)])
knn.pred_1 <- knn(variable_train, variable_test, df_train$quality, k=1)
cm_1 <- table(True = df_test$quality, Prediction = knn.pred_1)
cm_1
sum(diag(cm_1))/sum(cm_1)
knn.pred_5 <- knn(variable_train, variable_test, df_train$quality, k=5)
cm_5 <- table(True = df_test$quality, Prediction = knn.pred_5)
```

```r
cm_5
sum(diag(cm_5))/sum(cm_5)
# Appendix A-1---------------------------------------------------------------
par(mfrow=c(1, 1))
missmap(df, col = c("blue", "red"), legend = FALSE)
# Appendix A-2---------------------------------------------------------------
par(mfrow=c(3,4))
for (i in 1: (length(df)-1)){
  hist(x = df[[i]],
       main = sprintf('Histogram of %s', colnames(df[i])),
       xlab = colnames(df[i]))
}
# Appendix A-3---------------------------------------------------------------
barplot(quality)
# Appendix A-4---------------------------------------------------------------
par(mfrow=c(1, 1))
appendix_a_4
# Appendix A-6---------------------------------------------------------------
library(stats)
par(mfrow=c(1, 2))
krange = 1:10
kout = list()
for (k in krange){
  kout = cbind(kout,kmeans(df[-12],k))
}
plot(krange, kout["tot.withinss",], type = 'b',
     main = 'Total within-cluster-sum-of-squares' )
plot(krange, kout["betweenss",], type = 'b',
     main = 'The between-cluster-sum-of-squares')
psub = c(1,2)
for (i in seq(2,9,3)){
  barplot(table(kout[1,i]$cluster),
          main = paste('Cluster Frequency When k = ',as.character(i)))
  plot(pri$scores[,psub], col = kout[1,i]$cluster,
       main = 'Component1-2 Distribution')
  text(t(t(kout[2,i]$centers)-apply(df[-12],2,mean))%*%pri$loadings[,psub],
       labels = as.character(1:i), col = 'white')
}
# Appendix A-6---------------------------------------------------------------
par(mfrow=c(1, 1))
roc.curve(df_test$quality, prob2,n.thresholds = n)
# Appendix A-7---------------------------------------------------------------
par(mfrow=c(1, 1))
roc.curve(df_test$quality, lda_pred$posterior[,2],n.thresholds = n)
```