Feature Exploration & Analysis on Wine's Quality

Using Machine Learning Techniques

Table of Content

1 Introduction

In this project, we chose the "***winequality-white***" dataset obtained from the UCI Machine Learning Repository (2009) as our research subject. This dataset contains chemical features and compositions of the Portuguese "Vinho Verde" white wine (Cortez et al., 2009), and could provide some insights into how the chemical compositions work and how much does it affect the flavor of the wine.

Particularly, our main objective was to:

1) Predict several necessary indexes including Alcohol and Density with other chemical and sensory variants known and available, in case of their absence in some situations.
2) Classify a white wine's quality with the aid of both provided and estimated variables.

2 Data Description

2.1 Data Overview

Without missing values, the dataset contains 4898 observations and 12 variables. All of them are relevant to our research purposes due to their physicochemical characteristics. The variables consisted in the data are:

a. *fixed acidity* (tartaric acid — g / dm³)
b. *volatile acidity* (acetic acid — g / dm³)
c. *citric acid* (g / dm³)
d. *residual sugar* (g / dm³)
e. *chlorides* (sodium chloride — g / dm³)
f. *free sulfur dioxide* (mg / dm³)
g. *total sulfur dioxide* (mg / dm³)
h. *density* (g / cm³)
i. *pH*
j. *sulphates* (potassium sulphate — g / dm3)
k. *alcohol* (% by volume)
l. *quality* (score between 0 and 10)

The types of the variables are all "numeric", except *fixed acidity*, whose type is "character". It is also remarkable that variable quality has scores reflecting different levels thus should be regarded as categorical. For the convenience of further investigation, we manually converted the types of *fixed acidity* and *quality* into "numeric" and "factor" respectively.

2.2 Visualization & Inferences

2.2.1 Histogram of *quality*.

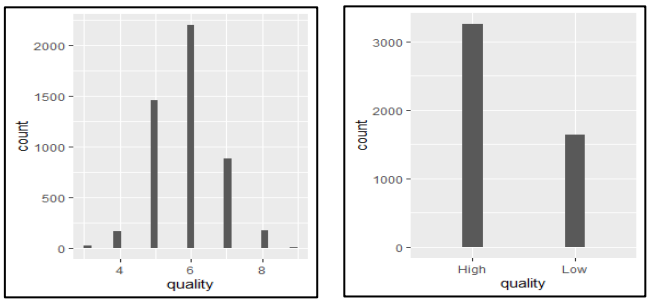


Figure 1 Histogram of pre-classified and re-classified quality distribution

From the histogram (left), we inferred that the initial sample data have *quality* with an imbalanced distribution since the observations for quality levels 5 and 6 have relatively larger frequencies than the rest. A classification result neglecting this feature could be highly biased due to a centralized training dataset. Hence, instead of proceeding with the initial quality levels, we generalized them into two categories, "Low" and "High" (represented by "0" and "1" in code), where "Low" included levels less or equal to 5 and "High" included others. We denoted this new binary variable by "y". The histogram (right) shows that the new groups are more comparable to each other.

2.2.2 Correlation Detection: Matrix Scatter Plot

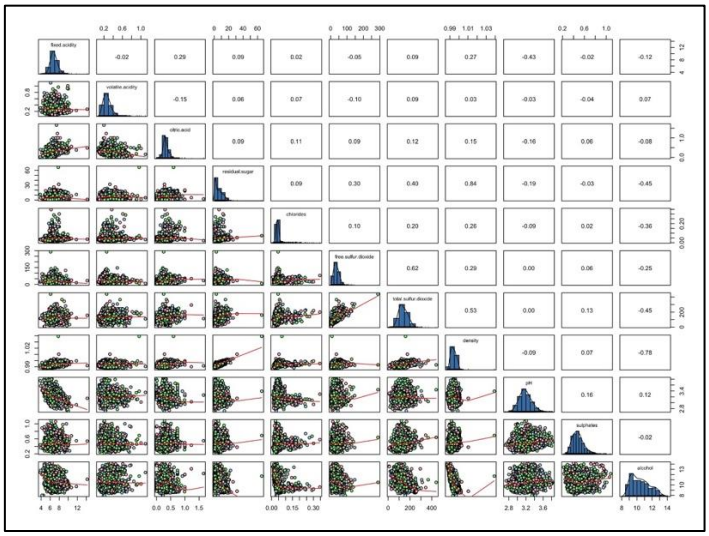


Figure 2 Matrix Scatter Plot

The matrix scatter plot was used for detecting the possible relationships between different indexes. For example, the plot displayed a highly negative correlation between *density* and *alcohol* (−0.78).

In addition, density appeared to be highly correlated to *residual sugar* (0.84). Such observations were the intuitive evidence and motivations for further regression analysis conducted in later sections.

2.2.3 Between-group Comparisons: Boxplots

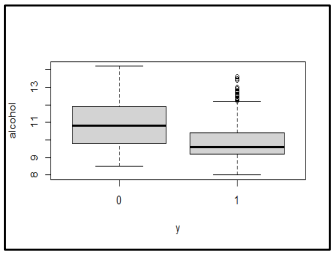


Figure 3 Boxplots for alcohol in the two groups

As defined, *y* is the new variable indicating quality. We conducted between-group comparisons to all numerical variables using boxplots (See Appendix). Most variables seemed not to vary obviously except for *alcohol* (as in Figure 3), on which we witnessed a significant difference between the two groups. To further convince, a two-sample t test was conducted to reject the null hypothesis of equal group means. Therefore, we concluded that alcohol varies across the quality thus *alcohol* must be obtained before starting classification on *quality.*

3 Methods & Model Selection

3.1 Data Split

Using the 80/20 splitting rule, we randomly sampled 80% of the sample data for our model training while employing the remaining data as the test set. By comparing the predicted responses with the real test data, we constantly evaluate and modify our models.

3.2 Regression Models

For this part, we used the "***lm***" and "***train***" functions in R to perform linear and KNN regressions on *density* and *alcohol* separately. Suitable regression models were then derived and compared for the two variables, where our criterion included the RMSE performances as well as the residual behaviors.

3.2.1 Predictions of *density*

In our previous setting and discussion, *density, alcohol* and *wine quality* were assumed unknown, and *density* could be potentially predicted by the given variants due to high correlations.

3.2.1.1 Linear Regression

Linear regression is a fundamental tool to construct linear relationships supported by Least-Square estimators (LSEs). We tried linear regression to predict *density* by the remaining variants. First, we directly included all the other nine predictors in their first-order terms and performed the regression model. We obtained an R-square of 83.44%, an RMSE of 0.001311328. The residual plots are shown below, in which the random scatters formed a band approximately symmetric to the zero line and generally fell on the QQ-normal line.

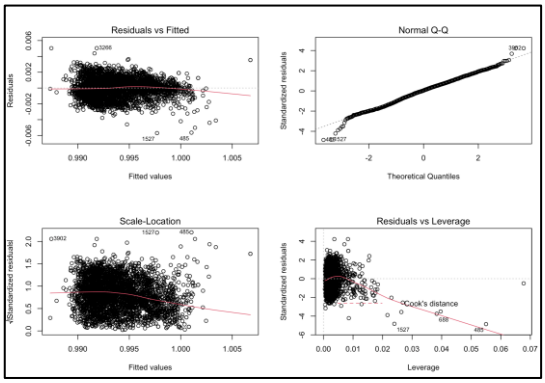


Figure 4 Residual Plot: LR - *density*

We continued removing some outliers appearing in the above residual plot. However, it turned out that the RMSE increased to 0.001330283. To avoid missing information, we decided to keep the raw data unchanged and still inclined to the original model.

3.2.1.2 KNN Regression

KNN regression considers the average of the responses among the K nearest neighbor as the predicted response of a new observation. In this context, the nine predictors were directly used as predictors in the KNN regression model. The optimal K was tuned to be 7 (see appendix) and an RMSE of 0.001572351 was realized, larger than that of the linear regression model. We ended up determining the linear regression as a more appropriate method for the task.

3.2.2 Predictions of *alcohol*

Now that the *density* has been estimated and a high

correlation between *density* and *alcohol* was observed in the Matrix Scatter Plot, it was natural to involve *density* as a regressor for *alcohol*.

3.2.2.1 Linear Regression

The linear regression model considering each first-order predictor gave an R-square of 90.48% and an RMSE of 0.6576248. As shown in Figure 6, a slight quadratic trend can be observed in the residual plot and there were points departing from the normal line in the right of Q-Q plot, suggesting residual behaviors against model assumptions.

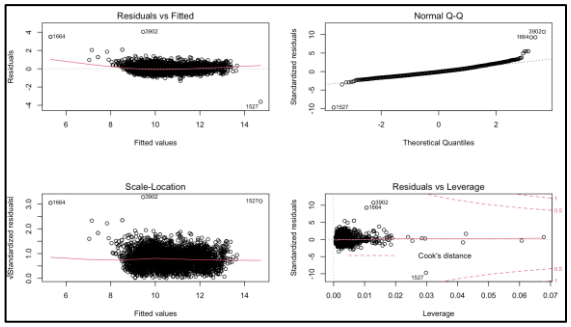


Figure 5 Residual Plot: LR - *alcohol*

Due to the quadraticity observed in residuals, we proposed an adjusted model discarding obvious outliers and interacting with some highly related second-order terms in the following correlation plot. We determined the engagement of those second order terms by looking at the correlation plot produced by ***corplot()*** (Figure 6). Specifically, the square and interaction terms of density, total sulfur dioxide and residual sugar were included.

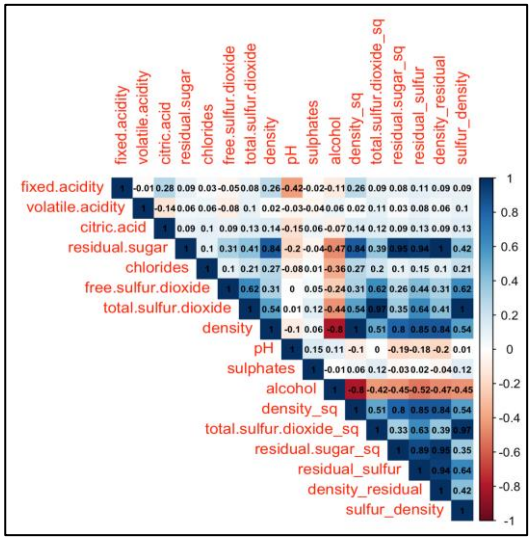


Figure 6 Correlation Plot

The adjusted model seemed better as an R-square of 87.01% as well as an RMSE of 0.4438751 were given, both of which outperformed the raw model. Note that the new RMSE was 32.5% down from the previous one. The residual plot was satisfactory without any abnormal pattern around the base line, and the adjusted model's residuals' normality is also stronger than the original model according to the Q-Q plot. Both plots are shown as below (Figure 7).

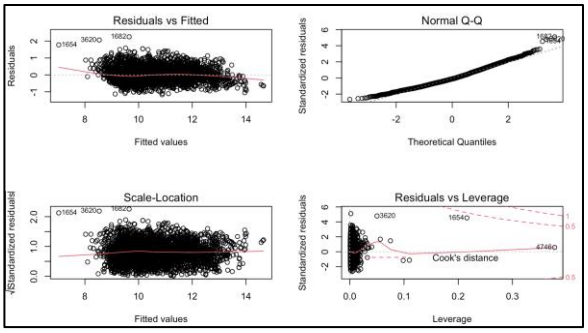


Figure 5 Residual Plot: adjusted LR - *alcohol*

3.2.2.2 KNN Regression

The ten predictors were directly used as predictors in the KNN regression model. The optimal K was tuned to be 5 (see appendix) and an RMSE of 0.5628379 was realized, larger than that of the linear regression model. Therefore, we ended up accepting the adjusted linear regression model as the best model for the task.

3.3 Classification Models

In this section, we gathered all available variables to build up models for classifying *quality* levels (high or low). The model selection was referred to the average consistency between expected and real categories of our test data. Possible candidates included logistic regression, KNN and SVM.

3.3.1 Logistic Regression

Logistic Regression extends linear regression to probabilities and gives binary outcomes. In this section, we constructed a logistic regression model via the function ***glm()*** to classify the *quality*. Based on the confusion matrix, we got the accuracy of the classification to be 73.2%.

y_pred	0	1
0	159	83
1	180	558

Table 1 Confusion Matrix: logistic regression

3.3.2 KNN Classification

KNN algorithm calculates the K smallest distance

between the input vector and other points in the training data. The most represented class by these K neighbors is considered as the class of the input value.

In this subsection, we normalized our data and built up the simple model using ***knn()*** function from R packages "**class**", where we firstly set K = 2. The accuracy of this situation was 69.6%. To find the optimal K value constructing best classifier, we made K iterate from 1 to 30. In the end, we get the summary accuracy output and an accuracy plot.

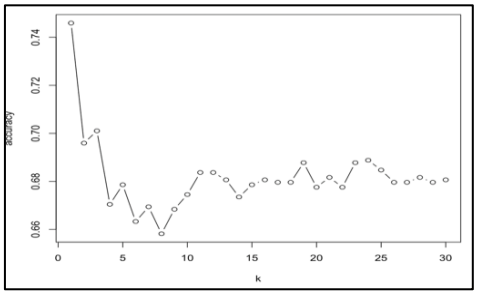


Figure 6 KNN Tuning

A small value of K can make the prediction result extremely sensitive to the neighboring points, which can be easily misled by noises. Hence, rather than choosing K = 1, we tended to select the next optimal parameter K = 3, resulting in the best accuracy of 70.1%.

3.3.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) generates optimal boundary hyperplanes to maximize the margins between the classifier and the support vectors. Particularly, we applied SVM to our data thus constructing high-dimensional boundaries between the two categories of *quality*. We implemented the SVM model via the function ***svm()*** in the "**e1071**" packages of R.

pred.svm1	0	1
0	143	75
1	196	566

pred.svm.tune1	0	1
0	197	59
1	142	582

Table 2 Confusion Matrices: linear SVM (left) & radial SVM (right)

Considering in all numerical components, we first tried a linear decision boundary to see whether it sufficed to give ideal classifications with respect to our test data. The accuracy for the linear SVM was 72.3%, close to that of logistic regression. To further improve, radial kernel with parameters tuned was proposed to conduct appropriate transformations. The rate of accuracy, as expected, grew to 79.5%, and the contingency table (Table 2) had fewer errors. This model prevailed all previous ones on the grouping performance.

4 Conclusion & Discussions

4.1 Summary of Results

In a nutshell, using available variables in our setting, we successfully generated models with satisfactory accuracy to predict *density* and *alcohol*, which proved to be key in classifying the final *quality* level. Linear regression containing all first-order predicting variables was selected to estimate density with RMSE = 0.0013, while the optimal model for regressing on alcohol contains several second-order terms and gave RMSE = 0.4439. The prioritized classification model to identify qualities was the SVM model with 79.5% accuracy.

4.2 Discussion & Potential Improvements

In general, the performances given by the optimal models were acceptable, in both regression and classification tasking. However, to further improve the modelling, one may consider the following aspects:

1) The overfitting problem. Although the linear regression model gave impressive prediction accuracy, the model might be overfitted as all variables were included. One may check whether there will be equivalently good performance with less regressors involved.
2) Ordinal categorical variables. The *quality* in this data could be regarded as ordinal (according to its definition and distribution). Hence, models like ordinal regression can be introduced to address such phenomenon.

5 References

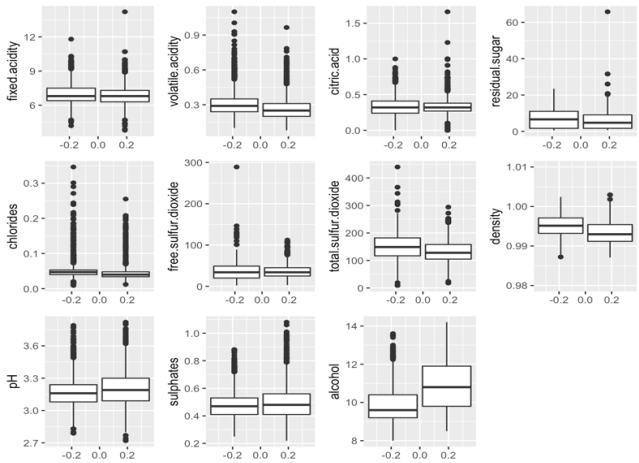
Cortez P, Cerdeira A, Almeida F, Matos T and Reis J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems, Elsevier*. 47(4):547-553.

The UCI Machine Learning Repository. (2009). Wine Quality Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/wine+quality

Appendix

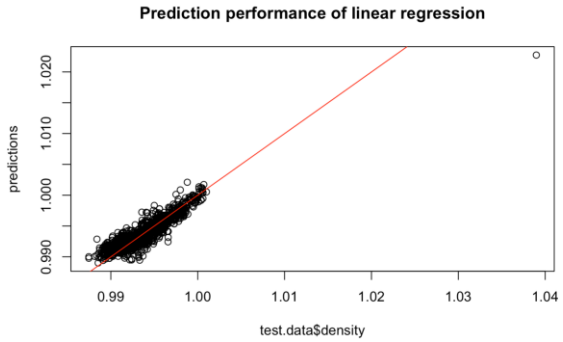
I. Relevant Figures

i. Boxplots for Between-Group Comparisons

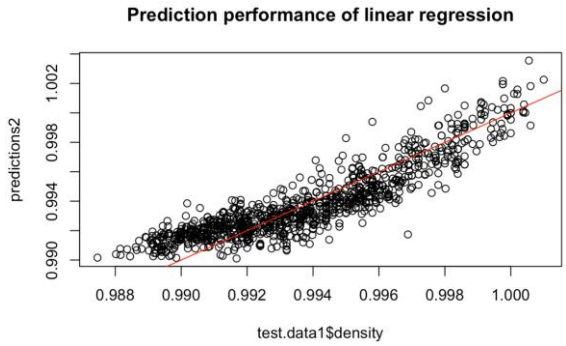


ii. Linear Regression Prediction Performances

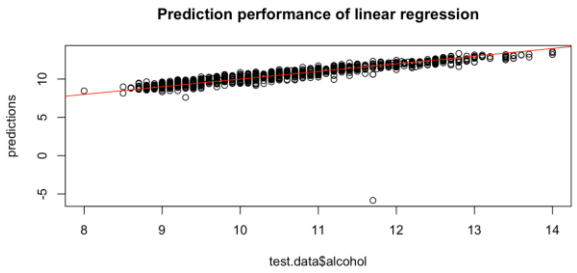
a) Original LR: Density



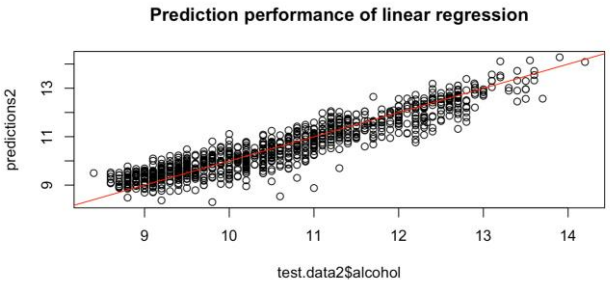
b) Adjusted LR: Density



c) Original LR: Alcohol

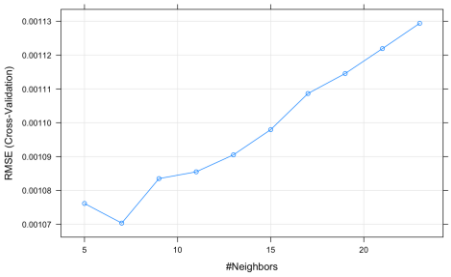


d) Adjusted LR: Alcohol

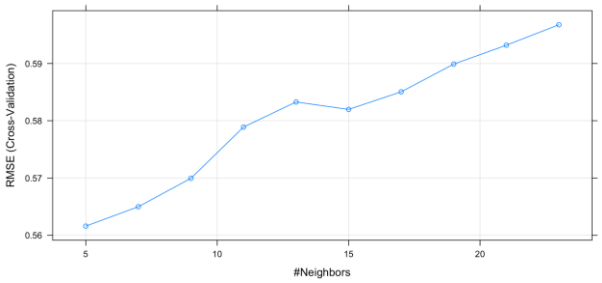


iii. KNN Regression Tuning

a) Density

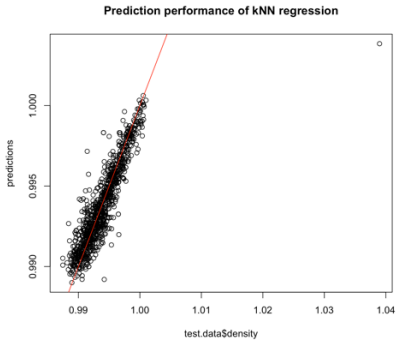


b) Alcohol

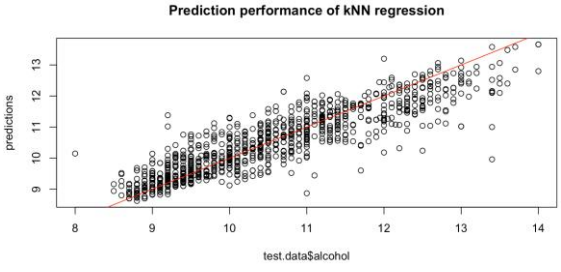


iv. KNN Regression Prediction Performances

a) Density



b) Alcohol



II. Code Availability (R Code)

```
library(psych)
library(e1071)
library(dplyr)
library(ggplot2)
library(class)
library(caret)
library(corrplot)
# install.packages("gridExtra")
library(gridExtra)
#data preparation
wine <- read.csv("/Users/Hello/Downloads/winequality-white.csv", sep = ";", head = TRUE)
head(wine)
dim(wine)
attach(wine)
wine[,1]<-sapply(wine[,1],as.numeric)
# Reclassify quality into 2 categories
wine %>% ggplot(aes(x = quality))+ geom_histogram()
wine <- wine %>% mutate(y = as.factor((ifelse(quality <= 5, 0, 1))))
wine %>% ggplot(aes(x = y))+ geom_bar(width = 0.2)

colnames(wine)

# Matrix Scatter plot
pairs.panels(wine[, 1:11],
                    method = "pearson", # correlation method
                    hist.col = "steelblue",
                    pch = 21, bg = c("pink", "light green", "light blue")[(iris$Species)], density = TRUE, # show
density plots
                    ellipses = FALSE # show correlation ellipses
)

# Boxplots
p1 <- wine %>% ggplot(aes(group = y, y = fixed.acidity)) + geom_boxplot()
p2 <- wine %>% ggplot(aes(group = y, y = volatile.acidity)) + geom_boxplot()
p3 <- wine %>% ggplot(aes(group = y, y = citric.acid)) + geom_boxplot()
p4 <- wine %>% ggplot(aes(group = y, y = residual.sugar)) + geom_boxplot()
p5 <- wine %>% ggplot(aes(group = y, y = chlorides)) + geom_boxplot()
p6 <- wine %>% ggplot(aes(group = y, y = free.sulfur.dioxide)) + geom_boxplot()
p7 <- wine %>% ggplot(aes(group = y, y = total.sulfur.dioxide)) + geom_boxplot()
p8 <- wine %>% ggplot(aes(group = y, y = density)) + geom_boxplot() + ylim(0.98, 1.01)
p9 <- wine %>% ggplot(aes(group = y, y = pH)) + geom_boxplot()
p10 <- wine %>% ggplot(aes(group = y, y = sulphates)) + geom_boxplot()
p11 <- wine %>% ggplot(aes(group = y, y = alcohol)) + geom_boxplot()
grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, nrow = 3) # Put boxplots on one page

t.test(filter(wine, y == 1)$alcohol, filter(wine, y == 0)$alcohol, paired = FALSE)
```

```
# Split the data
set.seed(100)
training.idx <- sample(1: nrow(wine), size=nrow(wine)*0.8)
train.data <-wine[training.idx, ]
test.data <- wine[-training.idx, ]
dim(train.data)
dim(test.data)

# Estimate Density
## KNN Regression
set.seed(101)
knn_model1 <- train(
   density~
      fixed.acidity+
      volatile.acidity+
      citric.acid+
      residual.sugar+
      chlorides+
      free.sulfur.dioxide+
      total.sulfur.dioxide+
      pH+
      sulphates,
   data = train.data, method = "knn",
   trControl = trainControl("cv", number = 100),
   preProcess = c("center","scale"),
   tuneLength = 10)

plot(knn_model1)
knn_model1$bestTune
predictions <- predict(knn_model1, test.data)
head(predictions)
RMSE(predictions, test.data$density)

par(mfrow=c(1,1))
plot(test.data$density, predictions,main="Prediction performance of kNN regression")
abline(0,1, col="red")

## Linear Regression
lmodel <- lm(density~fixed.acidity+ volatile.acidity+ citric.acid+
                     residual.sugar+ chlorides+ free.sulfur.dioxide+ total.sulfur.dioxide+
                     pH+ sulphates, data = train.data)
summary(lmodel)

predictions <-predict(lmodel, test.data)
plot(test.data$density, predictions, main="Prediction performance of linear regression")
abline(0,1, col="red")
RMSE(predictions, test.data$density)
par(mfrow=c(2,2))
plot(lmodel)
```

```
# Improved: Eliminate outliers and attempt on second-order terms
par(mfrow=c(1,1))
corrplot(cor(train.data[, 1:11] %>%
                        mutate(residual.sugar_sq = residual.sugar^2,
                                total.sulfur.dioxide_sq = total.sulfur.dioxide^2,
                                residual_sulfur = residual.sugar*total.sulfur.dioxide)),
             type="upper", method="color",addCoef.col = "black",number.cex = 0.6)

# Eliminate Outliers
wine1 <- wine[-c(1527, 3266, 485, 3902, 4481),]
set.seed(105)
training.idx1 <- sample(1: nrow(wine1), nrow(wine1)*0.8)
train.data1 <- wine1[training.idx1, ]
test.data1 <- wine1[-training.idx1, ]

lmodel2 <- lm(density~fixed.acidity+ volatile.acidity+ citric.acid+
                        residual.sugar+ chlorides+ free.sulfur.dioxide+ total.sulfur.dioxide+
                        pH+ sulphates, data = train.data1)
summary(lmodel2)
predictions2 <-predict(lmodel2, test.data1)
RMSE(predictions2, test.data1$density)
plot(test.data1$density, predictions2, main="Prediction performance of linear regression")
abline(0, 1, col = "red")
par(mfrow = c(2, 2))
plot(lmodel2)

# Estimate Alcohol
# KNN Regression
set.seed(101)
knn_model2 <- train(
   alcohol~
      fixed.acidity+
      volatile.acidity+
      citric.acid+
      residual.sugar+
      chlorides+
      free.sulfur.dioxide+
      total.sulfur.dioxide+
      pH+
      density+
      sulphates,
   data = train.data, method = "knn",
   trControl = trainControl("cv", number = 100),
   preProcess = c("center","scale"),
   tuneLength = 10)

plot(knn_model2)
knn_model2$bestTune
```

```
predictions <- predict(knn_model2, test.data)
head(predictions)
RMSE(predictions, test.data$alcohol)

par(mfrow=c(1,1))
plot(test.data$alcohol, predictions,main="Prediction performance of kNN regression")
abline(0,1, col="red")


# Linear Regression
lmodel <- lm(alcohol~fixed.acidity+ volatile.acidity+ citric.acid+
                        residual.sugar+ chlorides+ free.sulfur.dioxide+ total.sulfur.dioxide+
                        density+ pH+ sulphates, data = train.data)
summary(lmodel)

predictions <-predict(lmodel, test.data)
plot(test.data$alcohol, predictions, main="Prediction performance of linear regression")
abline(0,1, col="red")
RMSE(predictions, test.data$alcohol)
par(mfrow=c(2,2))
plot(lmodel)

par(mfrow=c(1,1))
# corrplot(cor(train.data[, 1:11]), type="upper", method="color",addCoef.col = "black",number.cex = 0.6)
corrplot(cor(train.data[, 1:11] %>%
                        mutate(density_sq = density^2,
                                total.sulfur.dioxide_sq = total.sulfur.dioxide^2,
                                residual.sugar_sq = residual.sugar^2,
                                residual_sulfur = residual.sugar*total.sulfur.dioxide,
                                density_residual = density*residual.sugar,
                                sulfur_density = density*total.sulfur.dioxide)),
             type="upper", method="color",addCoef.col = "black",number.cex = 0.6)

wine2 <- wine[-c(1527,1664,3902,2782),]
set.seed(106)
training.idx2 <- sample(1: nrow(wine2), nrow(wine2)*0.8)
train.data2 <- wine1[training.idx2, ]
test.data2 <- wine1[-training.idx2, ]

lmodel2 <- lm(alcohol~fixed.acidity+volatile.acidity+citric.acid+residual.sugar+
                        chlorides+total.sulfur.dioxide+I(density^2)+
                        sulphates+I(total.sulfur.dioxide^2)+I(residual.sugar^2)+
                        residual.sugar*total.sulfur.dioxide+density*residual.sugar+
                        density*total.sulfur.dioxide, data = train.data2)
summary(lmodel2)
predictions2 <-predict(lmodel2, test.data2)
RMSE(predictions2, test.data2$alcohol)
par(mfrow = c(2, 2))
```

```
plot(lmodel2)
par(mfrow = c(1, 1))
plot(test.data2$alcohol, predictions2, main="Prediction performance of linear regression")
abline(0,1, col="red")


# Logistic Regression (All variables)
mlogit<-glm(y~.,data=train.data %>% select(-quality),family="binomial")
summary(mlogit)
pred.p<-predict(mlogit,newdata=test.data,type="response")
y_pred_num<-ifelse(pred.p>0.5,1,0)
y_pred<-factor(y_pred_num,levels=c(0,1))
tab<-table(y_pred,test.data$y)
tab
mean(y_pred==test.data$y)

# logistic regression (selected variables)
mlogitn<-glm(y~ volatile.acidity + residual.sugar + free.sulfur.dioxide + density + pH +
                        sulphates + alcohol, data = train.data, family="binomial")
summary(mlogitn)
pred.p<-predict(mlogitn,newdata=test.data,type="response")
y_pred_numn<-ifelse(pred.p>0.5,1,0)
y_pred<-factor(y_pred_numn,levels=c(0,1))
mean(y_pred==test.data$y)

# KNN
nor<-function(x){(x-min(x))/(max(x)-min(x))}
nor_wine <- nor(wine[, 1:11])
wine1 <- cbind(wine, nor_wine)
# dim(wine1): 4898      24
set.seed(110)
knn1<-knn(wine1[, 14:24][training.idx, ], wine1[, 14:24][-training.idx, ], cl=wine1[training.idx, ]$y, k=2)
mean(knn1==test.data$y)
#find the best k
ac<-rep(0,30)
for(i in 1:30){
   set.seed(110)
   knn.i<-knn(wine1[, 14:24][training.idx, ], wine1[, 14:24][-training.idx, ], cl=wine1[training.idx, ]$y, k=i)
   ac[i]<-mean(knn.i==test.data$y)
   cat("k=",i,"accuracy=",ac[i],"\n")
}
par(mfrow = c(1, 1))
plot(ac,type="b",xlab="k",ylab="accuracy")



# SVM
set.seed(100)
m.svm1<-svm(y ~ ., data = train.data %>% dplyr::select(-quality), type = "C-classification",
```

```
                    kernel = "linear")
summary(m.svm1)
pred.svm1 <- predict(m.svm1, newdata=test.data %>% dplyr::select(-quality))
table(pred.svm1, test.data$y)
mean(pred.svm1 == test.data$y)

set.seed(2)
m.svm.tune1<-tune.svm(y ~ ., data=train.data %>% dplyr::select(-quality), type = "C-classification",
                                  kernel="radial", cost=10^(-1:2), gamma=c(.1,.5,1,2))
summary(m.svm.tune1)
plot(m.svm.tune1)
best.svm1 = m.svm.tune1$best.model
best.svm1
pred.svm.tune1 = predict(best.svm1, newdata=test.data %>% dplyr::select(-quality))
table(pred.svm.tune1, test.data$y)
mean(pred.svm.tune1 == test.data$y)
```