Dillon Ashburn

Xiaoxia Champon

Ying Yu

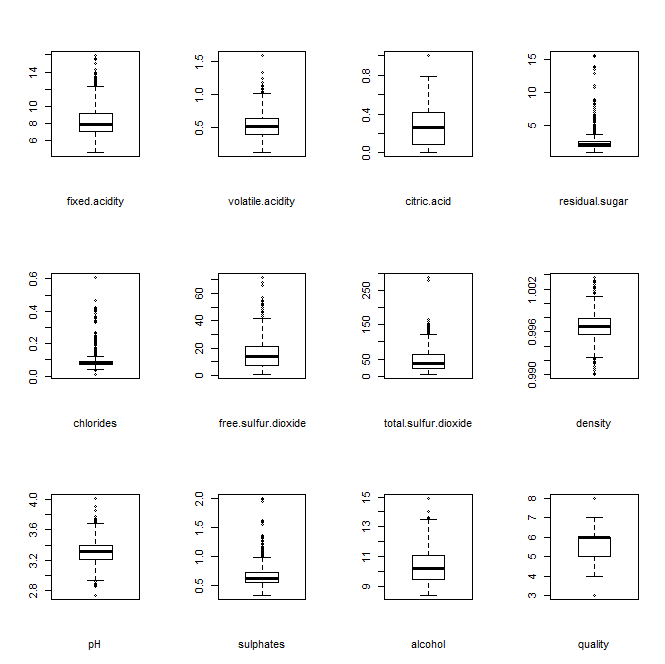
Adam Geringer

**Determining the Quality of Red Wine**

**Introduction:**

Data was collected on 1,599 different red wines in Portugal. The dataset consists of 1,599 observations and 11 predictors with one response variable. The predictors are continuous variables which are chemical properties of the wine varieties and the response variable is the quality of the wine. The quality is determined by three separate wine taste testers who rate the wine from 1 being the worst to 10 being the best. The median of these scores is taken to be the quality rating of that specific variety. From this dataset, predictive analysis and classification methods are performed to determine what makes a high quality wine to eliminate the subjectiveness of the wine taste testers.

**Analysis of Predictors:**

The following histograms and boxplots in Figure 1 show the distribution and spread of the predictor variables. Density and PH appear normally distributed with a symmetric bell curve. Total sulfur dioxide is highly skewed to the right. Residual sugar and chlorides both have a relatively obvious mode. The frequency of citric acid decreases then has a small jump around 0.2 with a decreasing pattern afterwards. All of the observed variables have outliers and an overwhelming majority of them are not normally distributed. The outliers affect the data set but cannot be simply removed for accuracy. Outliers can only be removed unless they are incorrect data entry. The variable “Quality” has the most values concentrated in the ratings of 5 and 6. 63 observations were of quality 3 or 4, 217 observations were of quality 7 or 8, and the rest of the observations were of quality 5 or 6. For classification purposes, 3 and 4 are classified as LOW quality wine, 5 and 6 are classified as MEDIUM quality wine, and 7 and 8 are classified as HIGH quality wine. This division of classification provides the distinct quality levels due to the few observations for LOW and HIGH quality wines.

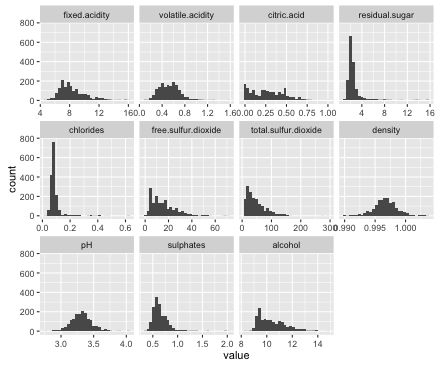


Figure : (Left) Histogram showing the distributions of the predictor variables. (Right) Boxplots showing the spread of the predictor variables.

Correlation was examined between all of the variables. The correlation table is given in figure 2 below. The strongest correlation with quality is alcohol with a correlation value of 0.476. There is a strong positive correlation between fixed acidity and citric acid, total sulfur dioxide and free sulfur dioxide, as well as between density and fixed acidity. There is a strong negative correlation between pH and fixed acidity, alcohol and density, as well as between citric acid and volatile acidity.

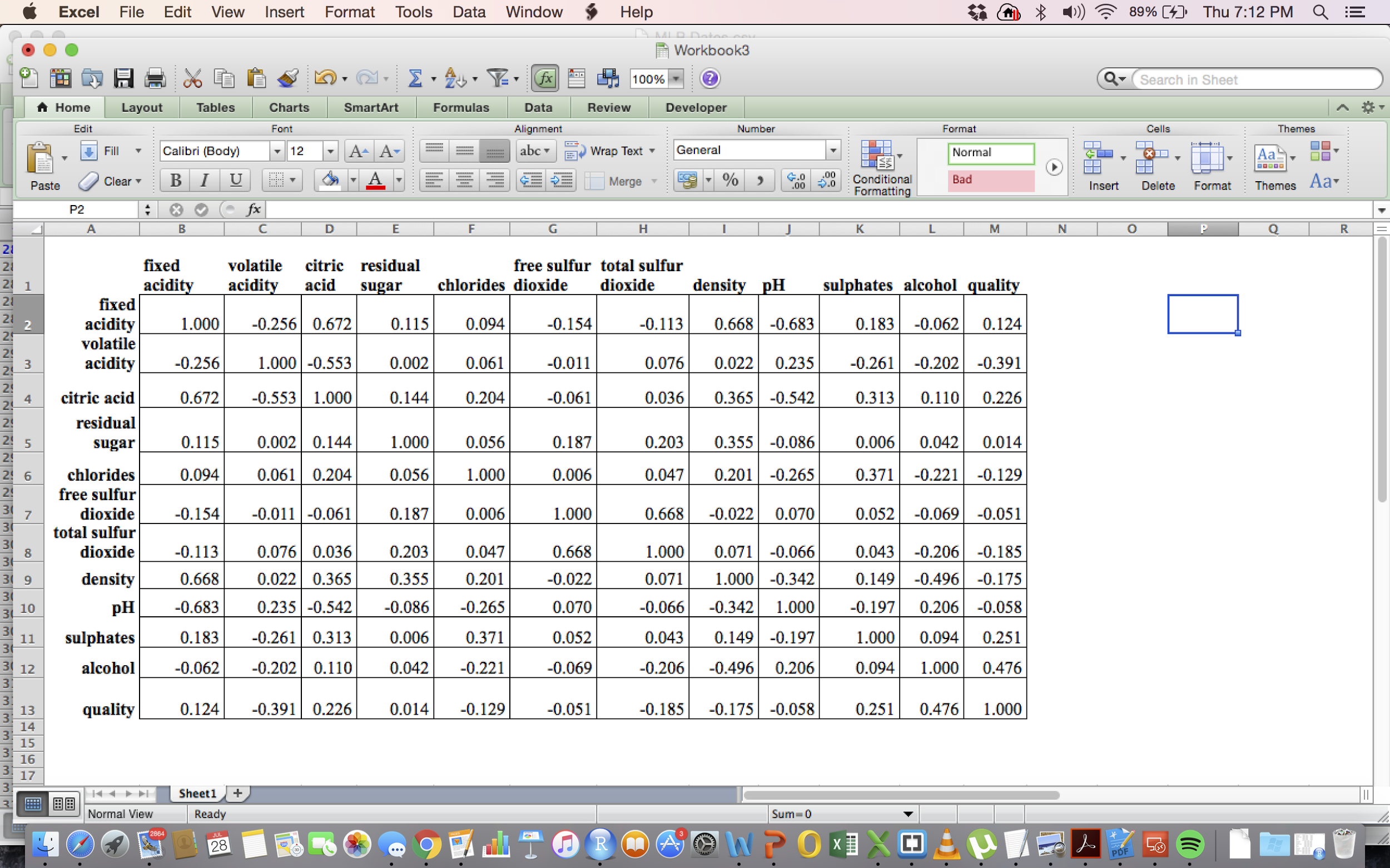


Figure : The correlation values between all of the variables.

**Regression Methods:**

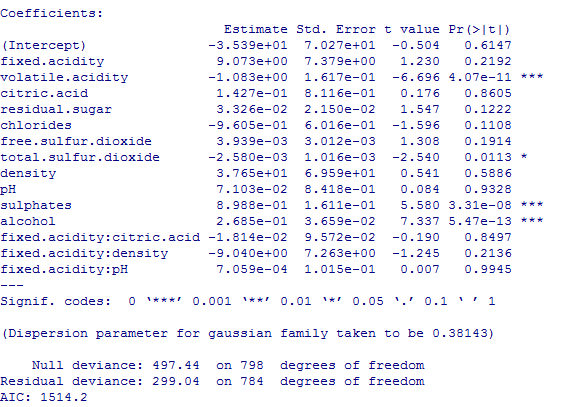
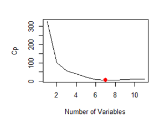
The dataset was randomly divided into training data and test data of equal sizes (50% each). Model I Linear regression is fitted on all the training data with all of the predictors and the main interaction terms. The following figure 3 shows the results of linear regression. 

Figure : Linear regression with all predictors.

Mean test error of the linear model with all predictors is 0.47. It is seen from the R output that not all the predictors are significant based on the P-value. Volatile acidity, sulphates, and alcohol are highly significant to the quality of the red wine and followed by total sulfur dioxide. Best subset selection is performed on training data set and minimum validation errors are identified from the test data set. Seven variables are included in the model with lowest validation error rate equal to 0.3985. This model also has the smallest Cp value. The following Figure 4 is the working model with corresponding coefficients.

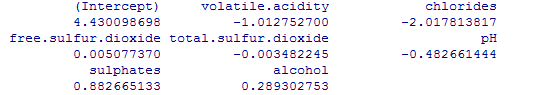


Figure : Working model and Cp plot based on the number of variables.

In order to investigate whether a polynomial relationship fits the model better, an alternative model with squared terms of the most significant variables is tried. This approach improves validation error rate to 0.3923. The following figure 5 is the coefficients of linear regression model with higher orders which has the smallest validation error.

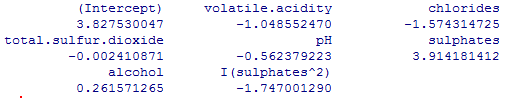


Figure : Linear regression model with higher orders.

**Applying Tree Based Methods:**

Tree function is used to fit a classification tree in order to predict the quality of the wine using all variables but quality. The test error rate for red wine quality classification is 1-(31+1+625)/800=17.875% before pruning (See Figure 6 for the R output). The cross-validation method is performed in order to determine the optimal level of tree complexity. The tree with five terminal nodes results in the lowest cross-validation error rate (this was plotted in Figure 7). The test error rate is 1-(36+626)/800=17.25% after pruning and this actually does not improve so much (See Figure 8 for the R output). Unsupervised random forest is also used on the training data set and the test error rate was reduced to 1-(52+635)/800=14.12% (Figure 9). The variable importance is identified and the results indicate that across all of the trees considered in the random forest, the alcohol and sulphates are by far the two most important variables (Figure 10).

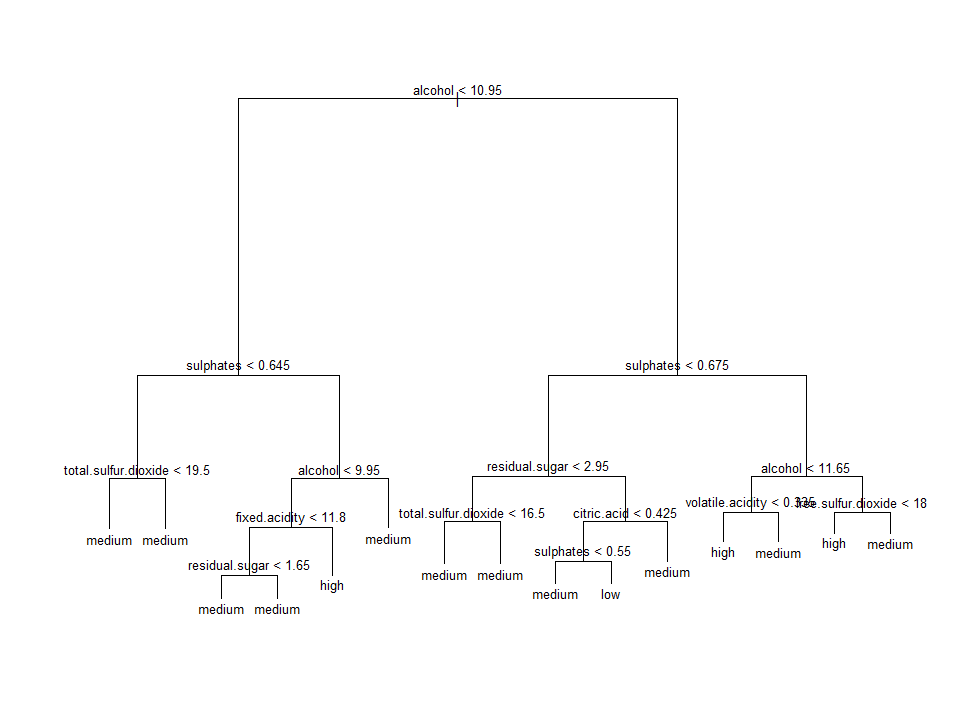
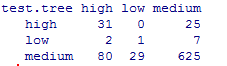


Figure : (Left) Classification tree before pruning. (Right) Test error using original tree.

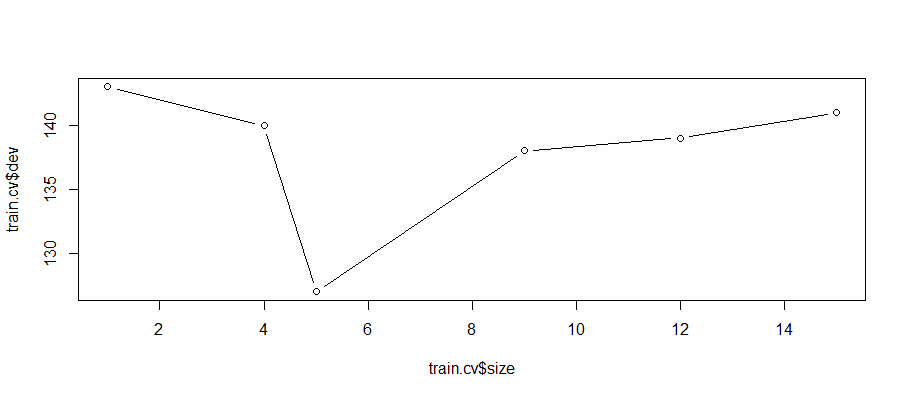


Figure : Test error vs. number of tree nodes by CV.

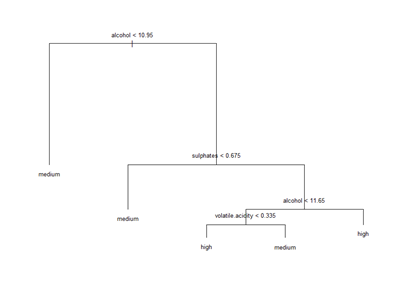
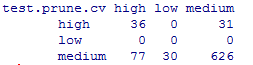


Figure : (Left) Classification tree after pruning by CV. (Right) Test error using pruned tree.

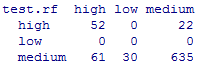


Figure : Test error for random forest method.

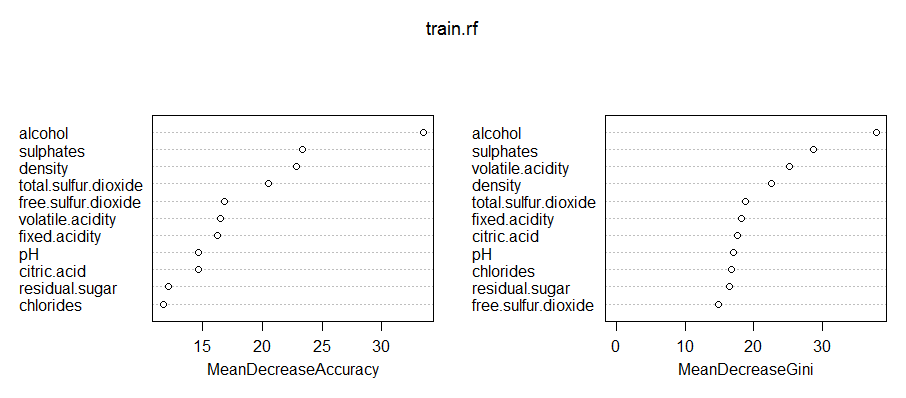


Figure : Variables importance plot using random forest.

**KNN, LDA, QDA Classification Methods:**

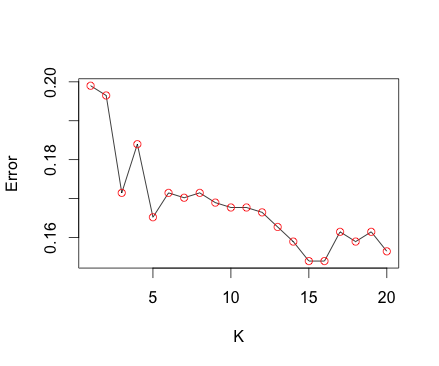
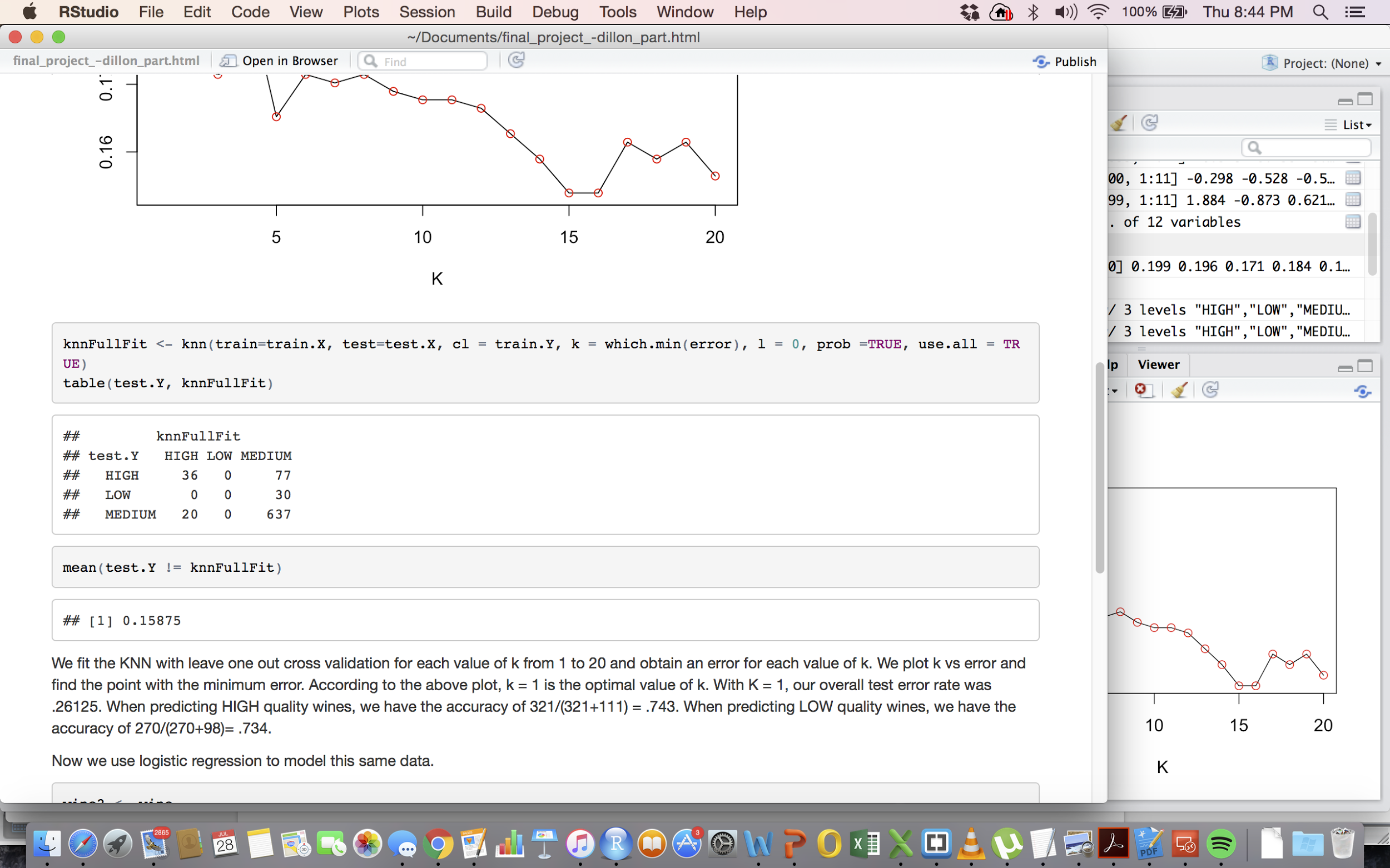
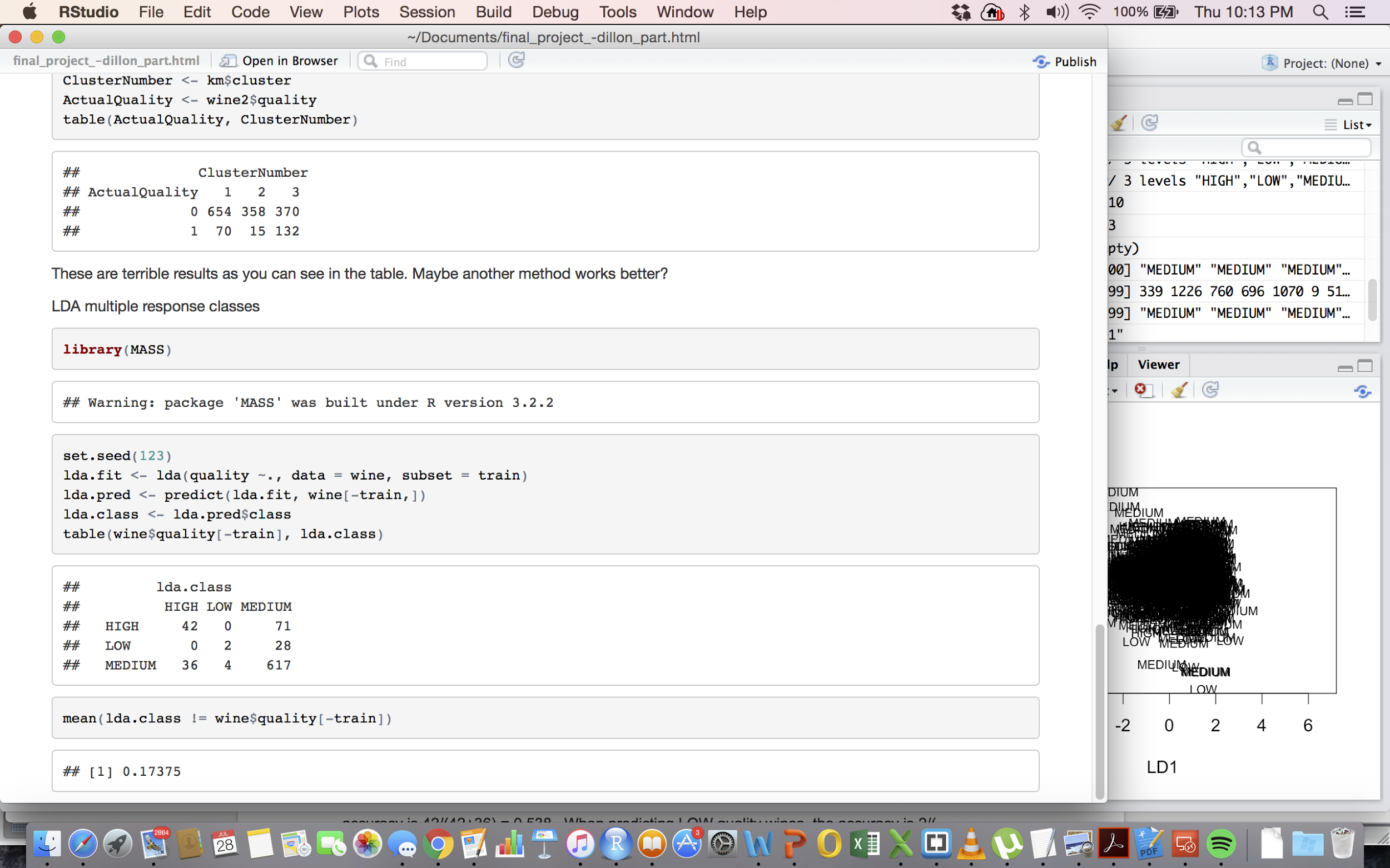
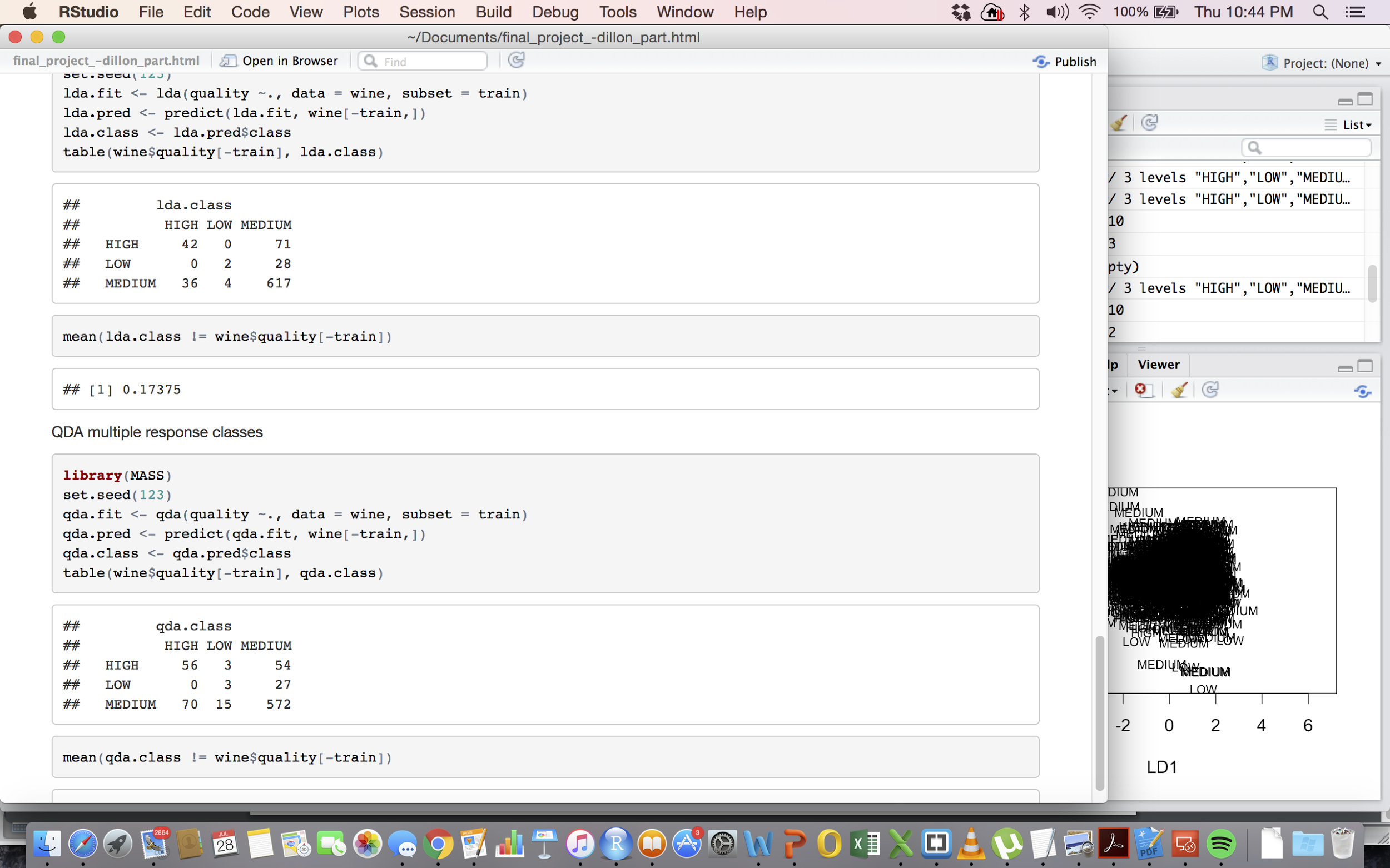
Supervised classification methods used were K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA). Running KNN, the predictor variables were standardized so that each predictor has mean zero and a standard deviation of one. This standardization eliminates the effects of different scales used to measure each individual variable. Leave one out cross validation was used to select the optimal value of K for KNN. The value of K versus the error is plotted in the figure to the right (Figure 11). The error is minimized at the value of K equals to 15. The KNN model was refit to the training data then applied to the test data. With K = 15, the overall test error rate was 0.15875. The results of the fit on the test data is shown in the figure to the right (Figure 12). When predicting HIGH quality wines, the accuracy is 36/(36+77) = 0.318. When predicting LOW quality wines, the accuracy is 0/30 = 0. When predicting MEDIUM quality wines, the accuracy is 637/(637+20) = 0.970. To see if this is the best classifications rates, another classification method was ran. Logistic Regression is not used because we we have more than two response classes, therefore we used linear discriminant analysis (LDA). The LDA model was fit on the training data then formed predictions on the test data. The results from LDA are displayed in the following figure to the right (Figure 13). The overall error mean is 0.17375. When predicting HIGH quality wines, the accuracy is 42/(42+71) = 0.372. When predicting LOW quality wines, the accuracy is 2/30 = 0.067. When predicting MEDIUM quality wines, the accuracy is 617/(617+4+36) = 0.939. Quadratic Linear Analysis was also run and the results are in the following figure to the right (Figure 14). The overall error rate was 0.211. The accuracy in predicting HIGH quality wines was 56/(56+3+54) = 0.496 and the accuracy in predicting LOW quality wines was 3/(27+3) = .10. Therefore, KNN is better than the LDA model based solely on the overall error rate. But, when predicting LOW quality wines, QDA was more accurate. When predicting MEDIUM quality wines, KNN was more accurate and when predicting HIGH quality wines, QDA was more accurate.

Figure :Test error rate vs. K values

Figure : Test error using KNN, K=15

Figure : Test error using LDA.

Figure : Test error using QDA.

**Conclusion**

In general, all the regression and classification methods have a relatively high error rate. The chemical properties of the red wine such as alcohol and sulphates do not necessarily contribute to the level of quality. The prediction accuracy of the model varies at different quality level. The classification methods were far more accurate for the MEDIUM wine categories but were much less accurate for predicting two other classes.

**Appendix:**

R Code:

Data File:

wine <- read.table('/Users/dashburn/Documents/winequality-red.csv', sep = ';', header = TRUE)

Histograms:

library(reshape2)

library(ggplot2)

d <- melt(wine[,])

ggplot(d,aes(x = value)) + facet\_wrap(~variable,scales = "free\_x") + geom\_histogram()

BoxPlots:

boxplot(fixed.acidity,xlab="fixed.acidity") (do the same for each variable)

The number of quality level  
nrow(subset(redwine,quality==8))  
[1] 18  
> nrow(subset(redwine,quality==3))  
[1] 10  
> nrow(subset(redwine,quality==4))  
[1] 53  
nrow(subset(redwine,quality==7))  
[1] 199

Correlations:

cor(wine,method="pearson")

Linear regression

set.seed(123)  
> train=sample(nrow(redwine),nrow(redwine)/2)  
>linear=glm(quality~.+fixed.acidity:citric.acid+fixed.acidity:density+fixed.acidity:pH,data=redwine,subset=train)   
> summary(linear)  
mean((quality-predict(linear,redwine))[-train]^2)  
Linear regression bestsubset  
> set.seed(123)  
> train=sample(c(TRUE,FALSE),nrow(redwine)/2,rep=TRUE)  
> test=(!train)  
> library(leaps)  
> regfit.best=regsubsets(quality~.,data=redwine[train,],nvmax=11)  
> test.mat=model.matrix(quality~.,data=redwine[test,])  
> val.errors=rep(NA,11)  
> for(i in 1:11){coefi=coef(regfit.best,id=i)  
+ pred=test.mat[,names(coefi)]%\*%coefi  
+ val.errors[i]=mean((quality[test]-pred)^2)}  
> val.errors  
> which.min(val.errors)

Linear regression with higher orders

set.seed(123)  
> train=sample(c(TRUE,FALSE),nrow(redwine)/2,rep=TRUE)  
> test=(!train)  
> library(leaps)  
> regfit.best=regsubsets(quality~.,data=redwine[train,],nvmax=11)  
regfit.best=regsubsets(quality~.+I(volatile.acidity^2)+I(sulphates^2)+I(alcohol^2),data=redwine[train,],nvmax=11)  
> test.mat=model.matrix(quality~.,data=redwine[test,])  
test.mat=model.matrix(quality~.+I(volatile.acidity^2)+I(sulphates^2)+I(alcohol^2),data=redwine[test,])  
> val.errors=rep(NA,11)  
> attach(redwine)  
> for(i in 1:11){coefi=coef(regfit.best,id=i)  
+ pred=test.mat[,names(coefi)]%\*%coefi  
+ val.errors[i]=mean((quality[test]-pred)^2)}   
> val.errors  
> which.min(val.errors)  
> coef(regfit.best,7)

Tree and random forest

wine <- read.csv("winequality-red.csv", header = T, sep = ";", na.strings = "NA")  
attach(wine)  
quality.class <- ifelse(quality<=4, "low", ifelse(quality>=7, "high", "medium"))  
wine <- data.frame(wine, quality.class)  
set.seed(123)  
train <- sample(1:nrow(wine), nrow(wine)/2)  
wine.train <- wine[train,]  
wine.test <- wine[-train,]  
  
library(tree)  
train.tree <- tree(quality.class ~.-quality, data = wine.train)  
plot(train.tree)  
text(train.tree,pretty = 0, cex=0.8)  
test.tree <- predict(train.tree, wine.test, type = "class")  
table(test.tree, wine.test$quality.class)  
1-(31+1+625)/800 # =0.17875

## prune tree using CV  
set.seed(123)  
train.cv <- cv.tree(train.tree, FUN=prune.misclass)  
train.cv # terminal node = 5 is the best can be found in output  
  
plot(train.cv$size, train.cv$dev, type="b")  
train.prune.cv <- prune.misclass(train.tree, best=5)  
plot(train.prune.cv)  
text(train.prune.cv, pretty = 0, cex=0.8)  
  
test.prune.cv <- predict(train.prune.cv, wine.test, type = "class")  
table(test.prune.cv, wine.test$quality.class)  
1-(36+626)/800 # =0.1725 Actually does not improve so much.  
  
## random forests  
library(randomForest)  
set.seed(123)  
train.rf <- randomForest(quality.class~.-quality, data=wine.train, mtry=3, importance=T)  
train.rf  
importance(train.rf)  
varImpPlot(train.rf)  
test.rf <- predict(train.rf, newdata = wine.test)  
table(test.rf, wine.test$quality.class)  
1-(52+635)/800 #=0.1412 got improved

KNN:

library(class)

wine <- read.table('/Users/dashburn/Documents/winequality-red.csv', sep = ';', header = TRUE)

wine$quality <- ifelse(wine$quality %in% c(3,4), 'LOW', ifelse(wine$quality %in% c(5,6),

'MEDIUM', 'HIGH'))

set.seed(123)

standardized.X <- scale(wine[,-12])

train <- sample(1:nrow(wine), nrow(wine)/2)

train.X <- standardized.X[train,]

test.X <- standardized.X[-train,]

train.Y <- wine$quality[train]

test.Y <- wine$quality[-train]

error <- NULL

for (k in 1:20){

knncv <- NULL

knncv <- knn.cv(train=train.X, cl = train.Y, k = k, l = 0, prob =TRUE, use.all = TRUE)

error[k] <- sum(knncv != train.Y) / length(train.Y)

}

plot(1:20, error, xlab = 'K', ylab = 'Error', type = 'l')

points(x = 1:20, y = error, col = 'red')

knnFullFit <- knn(train=train.X, test=test.X, cl = train.Y, k = which.min(error), l = 0, prob =TRUE, use.all = TRUE)

table(test.Y, knnFullFit)

mean(test.Y != knnFullFit)

LDA:

library(MASS)

set.seed(123)

lda.fit <- lda(quality ~., data = wine, subset = train)

lda.pred <- predict(lda.fit, wine[-train,])

lda.class <- lda.pred$class

table(wine$quality[-train], lda.class)

mean(lda.class != wine$quality[-train])

QDA:

library(MASS)

set.seed(123)

qda.fit <- qda(quality ~., data = wine, subset = train)

qda.pred <- predict(qda.fit, wine[-train,])

qda.class <- qda.pred$class

table(wine$quality[-train], qda.class)

mean(qda.class != wine$quality[-train])