# Machine learning project

Gemei Li

# Quality of Wine

Wine an alcoholic beverage has been around for a very long time and has a very large market. In order to stay in the market wine producers need to be able to produce wine of certain quality in order to stay competitive. We can measure the quality of wine by analyzing the physiochemical properties of wine. In this project we used a dataset that had samples of white wine from a Portuguese wine called "Vinho Verde". There were 11 chemicals and a quality variable that was created based on sensory data, which was not explained due to logistical reasons.

We wished to see which chemicals were important in predicting the quality of wine using multiple methods. These methods include knn, tree, logistic regression(ordinal), lasso, and linear regression.

## Variables

There were 12 variables in the data set:

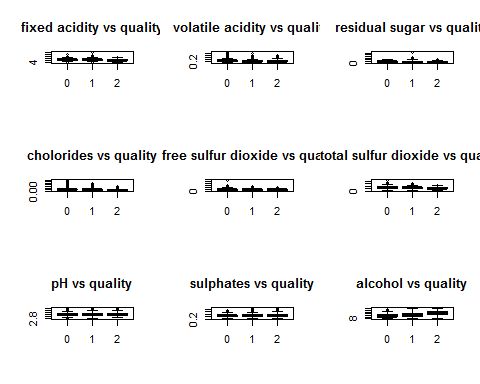
* **Fixed Acidity**:This is majority of the acid in the wine(considered fixed)
* **Volatile Acidity**:The amount of acetic acid found in the wine(high amounts leads to a vinegar taste
* **Citric Acid**:An organic acid thats adds flavor to the wine.
* **Residual Sugar**:The amount of sugar left after fermentation, the amount determines type of wine(Dry/Sweet)
* **Chlorides**:The amount of salt in the wine
* **Free Sulfur Dioxide**:The amount of SO2 found in the wine(Gas), helps prevent oxidation
* **Total Sulfur Dioxide**:The total amount both gas and bound forms of SO2, high levels leads to undesirables tastes in the wine.
* **Denisty**:Density of the wine( practically as dense as water)
* **Ph**:Describes how acidic or basic wine is (Most fall between 3 and 4)
* **Sulfates**:An additive that affects the levels of SO2 in wine
* **Alcohol**: Percent of Alcohol in wine(THE BEST PART)
* **Quality**:Quality of wine ranges from 3 to 9 in this dataset
* **Y**: Created based on the quality variable(Low=0,Average=1,and High=2)

## Analysis

wine.white$y <- ifelse(wine.white$quality==3 | wine.white$quality==4| wine.white$quality==5 ,0,ifelse(wine.white$quality==6,1,2))  
set.seed(001)  
train <- sample(length(wine.white$fixed.acidity),4000)  
white.test <- wine.white[-train,]  
white.train <- wine.white[train,]  
table(wine.white$y)

##   
## 0 1 2   
## 1640 2198 1060

We created a variable 'Y' that we wanted to classify with 3 levels. Due to preliminary analysis and our understanding of the variable density, we decided to drop it from the data set as we believe that it would not add much information as all the values are close to 1. We also created a training and test sets for validation. Since our data set it relatively large all our modeling will be based on the training set. ## Descriptive Data



From the boxplots above, there are no obvious differences bewteen the different catergories, the means are relatively close knowing that the count for average quality wine is signifcantly higher than the other 2 levels of quality. Interestingly enough the mean for alcohol is higher in the high quality when compared to the other 2.

## KNN

attach(wine.white)  
white.ytrain<-y[train]  
white.ytest<-y[-train]  
knn.result=knn(white.train,white.test,white.ytrain,k=3)  
table(white.ytest,knn.result)

## knn.result  
## white.ytest 0 1 2  
## 0 201 101 14  
## 1 101 265 43  
## 2 4 53 116

mean(white.ytest==knn.result)

## [1] 0.6481069

We classfied correctly about 65% of the time when k=3. Which is not a terrible classification rate given that our sample is based on 4000 observations. We will now check to see if there is better k.

class.rate = rep(0,20)  
for (K in 1:20) {  
 knn.result = knn( white.train,white.test,white.ytrain, K )  
 class.rate[K]=mean(white.ytest == knn.result )  
 }  
class.rate

## [1] 0.7449889 0.6625835 0.6492205 0.6425390 0.6369710 0.6135857 0.5991091  
## [8] 0.6035635 0.5812918 0.5846325 0.5968820 0.5902004 0.5946548 0.5779510  
## [15] 0.5857461 0.5857461 0.5779510 0.5690423 0.5623608 0.5723831

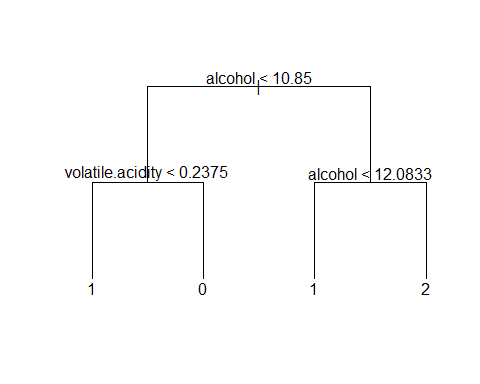
which.max(class.rate)

## [1] 1

So we find that our best k when k=1 but we wouldn't want to use k=1 since it is too flexible since it would classify at each point. Since k=3 is a better classification rate then k=2 our best k is when k=3.

## Tree

white.tree<-tree(as.factor(y)~fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+pH+sulphates+alcohol,data=white.train)  
plot(white.tree,type="uniform")  
text(white.tree)



summary(white.tree)

##   
## Classification tree:  
## tree(formula = as.factor(y) ~ fixed.acidity + volatile.acidity +   
## citric.acid + residual.sugar + chlorides + free.sulfur.dioxide +   
## total.sulfur.dioxide + pH + sulphates + alcohol, data = white.train)  
## Variables actually used in tree construction:  
## [1] "alcohol" "volatile.acidity"  
## Number of terminal nodes: 4   
## Residual mean deviance: 1.824 = 7288 / 3996   
## Misclassification error rate: 0.452 = 1808 / 4000

So we did a classification tree and ended up with a very simple tree with 4 terminal nodes. Earlier we stated that the data set does not clarify how the quality variable which is based off sensory data was catergozied. We just know that it ranged from 3 to 9 going from low to high, respectively. With all the chemicals the percentage of alcohol is the most important variable by the tree method. Essentially we anything greater than 12.083 was considered high quality. If the alcohol percentage was less than 10.85 then the amount of volatile acidity was then considered. From our study we can see that if you had more than 0.2375 then the quality of wine was classified as low. High amounts of volatile acidity can cause wine to have a vinegar taste which makes this tree very sensible. Now we will check the classification rate for the tree.

y.predict=predict(white.tree,wine.white,type="class")  
table(y.predict[-train],y[-train])

##   
## 0 1 2  
## 0 203 131 12  
## 1 107 226 106  
## 2 6 52 55

mean(y.predict[-train]==y[-train])

## [1] 0.5389755

Our classification rate using the tree method is .54 which is less than KNN when k=3. We can see from the table that it missclassfied high quality wine as average more than it did for KNN which can be due to the simplicity of our tree.

cv<-cv.tree(white.tree)  
names(cv)

## [1] "size" "dev" "k" "method"

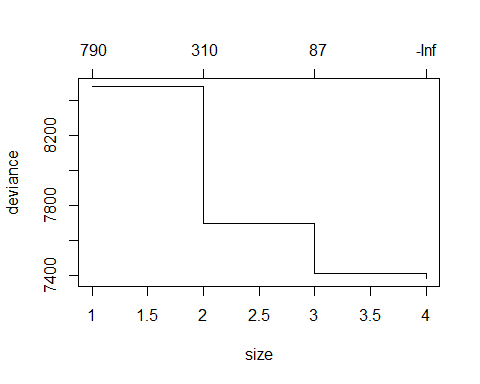
cv$size #number of terminal nodes

## [1] 4 3 2 1

cv$dev #deviance: #4 is the smallest

## [1] 7380.512 7408.044 7692.682 8481.579

plot(cv)



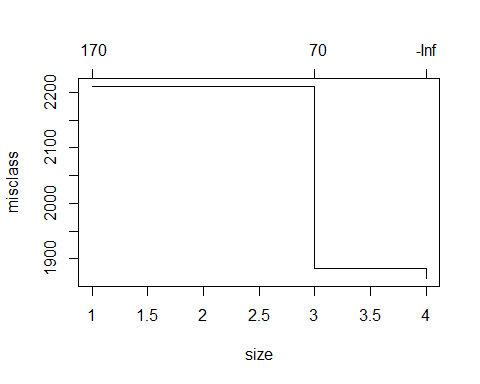
cv2=cv.tree(white.tree,FUN=prune.misclass)   
cv2$size #number of terminal nodes

## [1] 4 3 1

cv2$dev #now this is the number of mis classified units

## [1] 1865 1883 2211

plot(cv2)



white.tree.pruned=prune.misclass(white.tree,best=4)   
summary(white.tree.pruned)

##   
## Classification tree:  
## tree(formula = as.factor(y) ~ fixed.acidity + volatile.acidity +   
## citric.acid + residual.sugar + chlorides + free.sulfur.dioxide +   
## total.sulfur.dioxide + pH + sulphates + alcohol, data = white.train)  
## Variables actually used in tree construction:  
## [1] "alcohol" "volatile.acidity"  
## Number of terminal nodes: 4   
## Residual mean deviance: 1.824 = 7288 / 3996   
## Misclassification error rate: 0.452 = 1808 / 4000

Since the optimal was 4 which matches our previous result we can still see that tree classified less than KNN.

## Logistic Regression(Ordinal)

ww.lr.full <- lrm(y~fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+pH+sulphates+alcohol,data=white.train)  
ww.lr.full

## Logistic Regression Model  
##   
## lrm(formula = y ~ fixed.acidity + volatile.acidity + citric.acid +   
## residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +   
## pH + sulphates + alcohol, data = white.train)  
##   
## Model Likelihood Discrimination Rank Discrim.   
## Ratio Test Indexes Indexes   
## Obs 4000 LR chi2 1377.52 R2 0.331 C 0.771   
## 0 1324 d.f. 10 g 1.447 Dxy 0.542   
## 1 1789 Pr(> chi2) <0.0001 gr 4.250 gamma 0.544   
## 2 887 gp 0.255 tau-a 0.348   
## max |deriv| 2e-11 Brier 0.166   
##   
## Coef S.E. Wald Z Pr(>|Z|)  
## y>=1 -10.1119 1.0100 -10.01 <0.0001   
## y>=2 -12.6841 1.0191 -12.45 <0.0001   
## fixed.acidity -0.0718 0.0430 -1.67 0.0954   
## volatile.acidity -5.3971 0.3777 -14.29 <0.0001   
## citric.acid -0.1992 0.2749 -0.72 0.4687   
## residual.sugar 0.0663 0.0076 8.74 <0.0001   
## chlorides -2.8574 1.7314 -1.65 0.0989   
## free.sulfur.dioxide 0.0154 0.0026 6.01 <0.0001   
## total.sulfur.dioxide -0.0036 0.0011 -3.25 0.0011   
## pH 0.5780 0.2357 2.45 0.0142   
## sulphates 1.1591 0.2893 4.01 <0.0001   
## alcohol 0.9822 0.0367 26.78 <0.0001   
##

From initial summary we can see that fixed acidity,citric acid, and chlorides are insignificant. We proceeded to use fastbw which stands for fast backward variable selection, which is also found in the "rms" package. This method computes fitted values from a complete model and computes approximate Wald statistics by computing conditional mles assuming multivariate normaility of estimates.

fastbw(ww.lr.full)

##   
## Deleted Chi-Sq d.f. P Residual d.f. P AIC   
## citric.acid 0.53 1 0.4687 0.53 1 0.4687 -1.47  
## chlorides 3.01 1 0.0826 3.54 2 0.1704 -0.46  
##   
## Approximate Estimates after Deleting Factors  
##   
## Coef S.E. Wald Z P  
## y>=1 -10.529984 0.981644 -10.727 0.000e+00  
## y>=2 -13.100443 0.991246 -13.216 0.000e+00  
## fixed.acidity -0.076306 0.041702 -1.830 6.728e-02  
## volatile.acidity -5.411710 0.373251 -14.499 0.000e+00  
## residual.sugar 0.067541 0.007510 8.993 0.000e+00  
## free.sulfur.dioxide 0.015250 0.002561 5.954 2.622e-09  
## total.sulfur.dioxide -0.003665 0.001090 -3.361 7.760e-04  
## pH 0.606758 0.235175 2.580 9.879e-03  
## sulphates 1.143814 0.288646 3.963 7.411e-05  
## alcohol 0.999681 0.034877 28.663 0.000e+00  
##   
## Factors in Final Model  
##   
## [1] fixed.acidity volatile.acidity residual.sugar   
## [4] free.sulfur.dioxide total.sulfur.dioxide pH   
## [7] sulphates alcohol

ww.lr1 <- lrm(y~fixed.acidity+volatile.acidity+residual.sugar+free.sulfur.dioxide+total.sulfur.dioxide+pH+sulphates+alcohol,data=white.train,x=T,y=T)

Surprisingly, fixed acidity ended up staying in the final model. Then we checked for significance.

DevChi<-ww.lr1$deviance[1] - ww.lr1$deviance[2]   
c("DevChiSq"=DevChi, "p-value"=1-pchisq(DevChi, 10))

## DevChiSq p-value   
## 1373.874 0.000

Because the p-value is 0.000, our model is siginificant. Below are the 2 models from this regression.

Model 1:log(Y>=1)/log(Y<1)= -10.55-0.076(fixedacidity)-5.42(volatile acidity)+0.068(residual sugar)-2.9911(chlorides)+0.015(free sulfur dioxide)-0.0036(total sulfur dioxide)+0.61(pH)+1.14(sulphates)+1(alcohol).

Model 2: log(Y>=2)/log(Y<2)= -13.12-0.076(fixedacidity)-5.42(volatile acidity)+0.068(residual sugar)-2.9911(chlorides)+0.015(free sulfur dioxide)-0.0036(total sulfur dioxide)+0.61(pH)+1.14(sulphates)+1(alcohol).

pred <- predict(ww.lr1,white.test[,1:11],type='fitted.ind')  
class.predictions <- apply(pred,1,which.max)  
class.predictions[class.predictions ==1] <- 0  
class.predictions[class.predictions ==2] <- 1  
class.predictions[class.predictions ==3] <- 2  
table(class.predictions,white.test$y)

##   
## class.predictions 0 1 2  
## 0 176 82 9  
## 1 132 277 117  
## 2 8 50 47

mean(white.test$y==class.predictions)

## [1] 0.5567929

Our classification rate using a proportional odds model is 56% which is slightly above the tree's classification rate of 55%.

## Linear Regression and Lasso

From this point we treat Y as a numerical variable. We will compare a linear regession to lasso.

reg <- lm(y~.-quality,data=white.train)  
summary(reg)

##   
## Call:  
## lm(formula = y ~ . - quality, data = white.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.00457 -0.47566 -0.02263 0.44005 1.86515   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.6022890 0.3124629 -8.328 < 2e-16 \*\*\*  
## fixed.acidity -0.0191098 0.0134786 -1.418 0.15633   
## volatile.acidity -1.5872451 0.1038820 -15.279 < 2e-16 \*\*\*  
## citric.acid -0.0785556 0.0852480 -0.921 0.35685   
## residual.sugar 0.0202741 0.0023313 8.696 < 2e-16 \*\*\*  
## chlorides -0.8794579 0.5024065 -1.750 0.08011 .   
## free.sulfur.dioxide 0.0043884 0.0007707 5.694 1.33e-08 \*\*\*  
## total.sulfur.dioxide -0.0010251 0.0003397 -3.017 0.00257 \*\*   
## pH 0.1680085 0.0738737 2.274 0.02300 \*   
## sulphates 0.3500183 0.0897155 3.901 9.72e-05 \*\*\*  
## alcohol 0.3118634 0.0101369 30.765 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6191 on 3989 degrees of freedom  
## Multiple R-squared: 0.2931, Adjusted R-squared: 0.2914   
## F-statistic: 165.4 on 10 and 3989 DF, p-value: < 2.2e-16

reg2 <- step(reg)

## Start: AIC=-3824.38  
## y ~ (fixed.acidity + volatile.acidity + citric.acid + residual.sugar +   
## chlorides + free.sulfur.dioxide + total.sulfur.dioxide +   
## pH + sulphates + alcohol + quality) - quality  
##   
## Df Sum of Sq RSS AIC  
## - citric.acid 1 0.33 1529.5 -3825.5  
## <none> 1529.1 -3824.4  
## - fixed.acidity 1 0.77 1529.9 -3824.4  
## - chlorides 1 1.17 1530.3 -3823.3  
## - pH 1 1.98 1531.1 -3821.2  
## - total.sulfur.dioxide 1 3.49 1532.6 -3817.3  
## - sulphates 1 5.83 1535.0 -3811.2  
## - free.sulfur.dioxide 1 12.43 1541.6 -3794.0  
## - residual.sugar 1 28.99 1558.1 -3751.3  
## - volatile.acidity 1 89.49 1618.6 -3598.9  
## - alcohol 1 362.83 1892.0 -2974.7  
##   
## Step: AIC=-3825.53  
## y ~ fixed.acidity + volatile.acidity + residual.sugar + chlorides +   
## free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +   
## alcohol  
##   
## Df Sum of Sq RSS AIC  
## <none> 1529.5 -3825.5  
## - fixed.acidity 1 1.10 1530.5 -3824.7  
## - chlorides 1 1.35 1530.8 -3824.0  
## - pH 1 2.05 1531.5 -3822.2  
## - total.sulfur.dioxide 1 3.57 1533.0 -3818.2  
## - sulphates 1 5.69 1535.1 -3812.7  
## - free.sulfur.dioxide 1 12.28 1541.7 -3795.5  
## - residual.sugar 1 28.72 1558.2 -3753.1  
## - volatile.acidity 1 89.90 1619.4 -3599.1  
## - alcohol 1 363.19 1892.7 -2975.3

reg3 <-lm(y~1,data=white.train)  
anova(reg3,reg2)

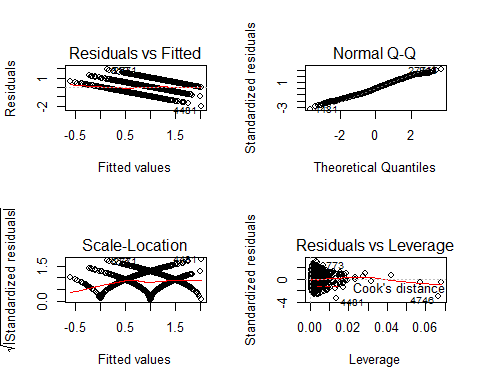
## Analysis of Variance Table  
##   
## Model 1: y ~ 1  
## Model 2: y ~ fixed.acidity + volatile.acidity + residual.sugar + chlorides +   
## free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +   
## alcohol  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 3999 2163.3   
## 2 3990 1529.5 9 633.8 183.72 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(reg2)

##   
## Call:  
## lm(formula = y ~ fixed.acidity + volatile.acidity + residual.sugar +   
## chlorides + free.sulfur.dioxide + total.sulfur.dioxide +   
## pH + sulphates + alcohol, data = white.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.00789 -0.47580 -0.02194 0.44038 1.86483   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.6042177 0.3124500 -8.335 < 2e-16 \*\*\*  
## fixed.acidity -0.0221307 0.0130736 -1.693 0.090575 .   
## volatile.acidity -1.5728996 0.1027069 -15.314 < 2e-16 \*\*\*  
## residual.sugar 0.0201387 0.0023267 8.656 < 2e-16 \*\*\*  
## chlorides -0.9344667 0.4988380 -1.873 0.061102 .   
## free.sulfur.dioxide 0.0043589 0.0007700 5.661 1.61e-08 \*\*\*  
## total.sulfur.dioxide -0.0010360 0.0003395 -3.052 0.002291 \*\*   
## pH 0.1705246 0.0738218 2.310 0.020942 \*   
## sulphates 0.3448297 0.0895369 3.851 0.000119 \*\*\*  
## alcohol 0.3111763 0.0101093 30.781 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6191 on 3990 degrees of freedom  
## Multiple R-squared: 0.293, Adjusted R-squared: 0.2914   
## F-statistic: 183.7 on 9 and 3990 DF, p-value: < 2.2e-16

Our final model using the step function and checking for overall significance included all the variables except for citric acid. The final model is Y=-2.60-0.022(fixed acidity)-1.57(volatile acidity)+0.02(residual sugar)-0.93(chlorides)+0.004(free sulfur dioxide)-0.001(total sulfur dioxide)+0.17(ph)+.34(sulphates)+0.31(alcohol)

par(mfrow=c(2,2))  
plot(reg2)



vif(reg2)

## fixed.acidity volatile.acidity residual.sugar   
## 1.278810 1.095717 1.441650   
## chlorides free.sulfur.dioxide total.sulfur.dioxide   
## 1.189748 1.757165 2.172095   
## pH sulphates alcohol   
## 1.323425 1.052295 1.648371

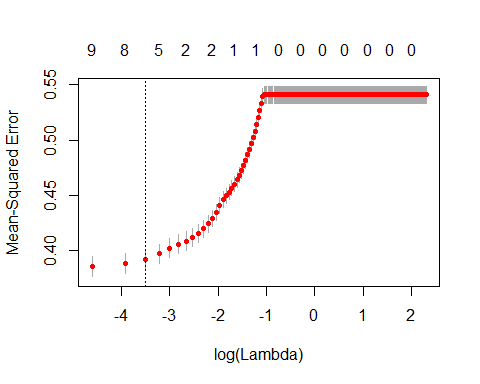
From our plots we can see some interesting information. Our qq plot shows that our data is practically normal with a few data points at the tails that may cause for concern but for a large data set this may not be an issue. The residual vs fitted show 3 distinct groups of points that are all parallel showing that our model can distinguish between the 3 catergories. Doing a quick vif function we can see that we are not worried about multicollinearity between the variables.

yhat <- predict(reg2,white.test[,1:10])  
lin.mse<-mean((yhat-white.test$y)^2)  
lin.mse

## [1] 0.3854574

Our MSE is 0.38

X <- model.matrix(y~.-quality,data=white.train)  
X2<- model.matrix(y~.-quality,data=white.test)  
lasso <- glmnet(X,white.train$y,alpha=1,lambda=seq(0,10,0.01))  
cv.lasso <- cv.glmnet(X,white.train$y,alpha=1,lambda=seq(0,10,0.01))  
plot(cv.lasso)



yhat2 = predict( lasso, cv.lasso$lambda.min, newx= X2)  
lasso.mse<-mean((yhat2 - white.test$y)^2)  
lasso.mse

## [1] 0.3853438

c(lin.mse,lasso.mse)

## [1] 0.3854574 0.3853438

After finding the mse using the lasso method we can see that lasso very narrowly out performed the linear regression since they both essentially round to either 0.39 either method is worth using but as Y is a catergorical method we would prefer to stick to classification methods.

## Conclusion

From our classification methods KNN classfied correctly 65% of the time while, ordinal logistic regression classfied properly 56% of the time with tree classifying at 55%. Our classification methods were relatively weak but there are some things we need to note about the method in which the variable "Y"" was created. Given what we know about the chemical properties that wine provides we cannot be conclusive about the quality of wine as the information surrounding the quality of wine is limited but also quite subjective. First, we must acknowledge that the quality variable in this dataset is not fleshed out in the sense that we do not know exactly how the quality was quantified. Due to logistic issues the company that released this dataset did not explain how quality was categorized. This would impact how we had chosen to split the quality variable. From the original distribution of the quality variable we can assume that most wine are of average quality. Most of the counts of the data can be found between 5 and 7 leading us to believe that assumption. From the previous boxplots we can see that the mean for the predictors did not differ greatly between the levels that we assigned. The chemical properties only tell us a portion of the story they are some other factors that can be considered the dataset does not include. They type of grapes, price of wine, storage of the wine and other properties that aren't limited to the chemical properties can also affect the quality of wine. Yet, we must also think about as an average consumer most wine found in stores today can be considered of the average quality, as wine that would be of low quality may just have been a bad batch. While on the other hand, high quality wine may not always be accessible due to either price or availability of said wine.

There are some things that we could have done differently that may change our prediction pow the of our model. For simplistic reasons we worked with only 3 levels for quality, but we could have increased the number of levels to show a larger spread. We could have checked for outliers in the x observations that may affect the classification of some samples but as we do not know exactly how quality was done we can't know for sure if by industry standards if some values are truly outliers. This dataset let us learn a little about how a "subjective" attribute can be tested with science but that it only tells us a piece of the story.