**Title: Red Wine Quality Prediction Based on Chemical Properties**

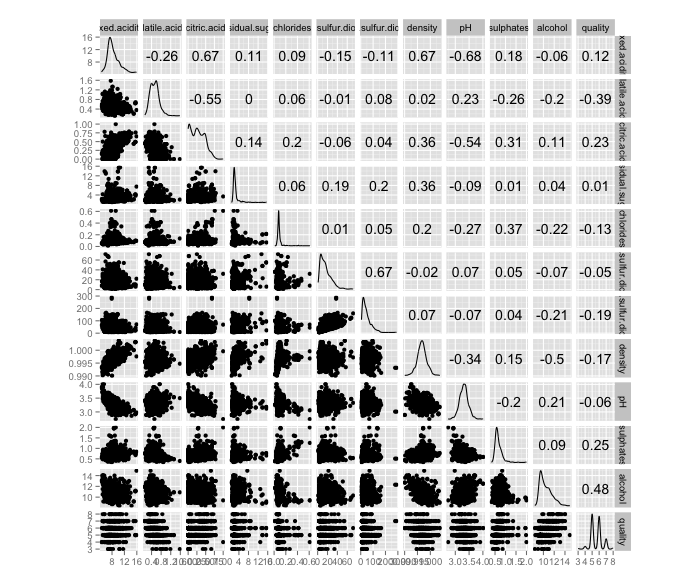
**Name: Tim Lin**

**Abstracts**

The dataset *Wine Quality* [1] contains 1599 observations of red wines. There are total 12 attributes, 11 input variables and 1 output variable. The input variables (*fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol*) are physiochemical test of wines. And the output variable, *quality*, scores between 0 (very bad) and 10(very excellent) based on the median of at least three sensory evaluations made by wine experts. The purpose of this analysis is to find a predictive model and predict the quality of wines without heavily relied on experts.

First, we find the significant variables in for predicting quality of the wine using general linear model. There are only seven significant variables out of eleven variables. Then, we applied statistical methods (generalized linear model, classification tree and random forest) to see if these chemical properties can accurately predict the quality of wine by cross validation. We find that random forest has the least misclassification rate, however the accuracy of classification is relative low. Therefore, we can conclude that classification based on chemicals is not effective in predicting wine quality.

**Data Exploration: Pairs Plots**



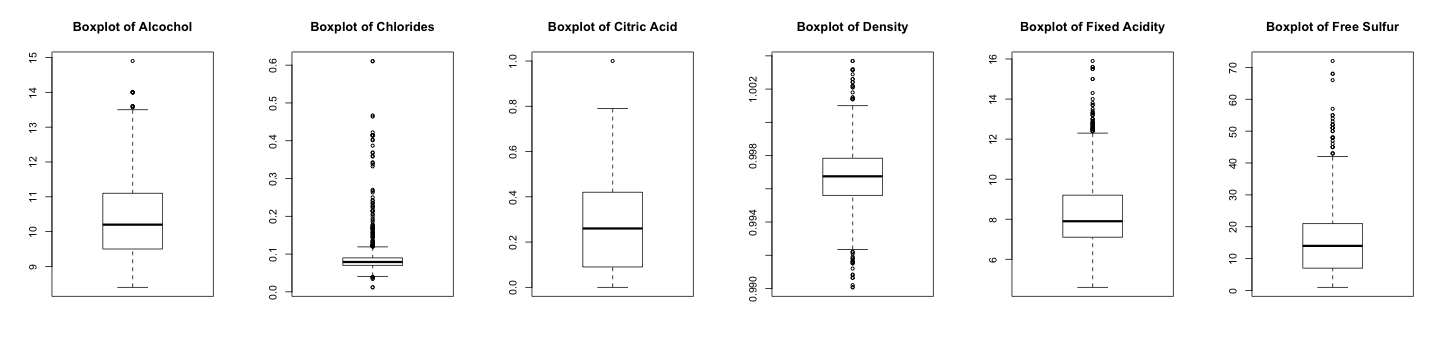
From the histograms of ggpairs plot following observations are made:

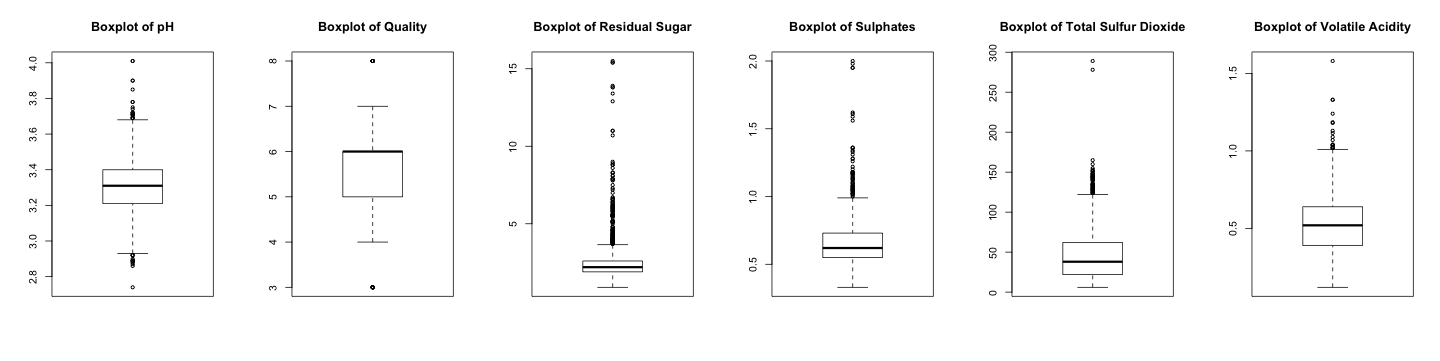
* *Density* and *pH* are normality distributed.
* *Fixed acidity, volatile acidity, sulphates, alcohol* is slightly skewed and possible symmetric might be achieved after removal of outliners.
* *Critic acid* has irregular shape with observations clustering at low values.
* *Residual sugar*, *chlorides, and free sulfur dioxide and total sulfur dioxide* are strong positively skewed. These observations are clustering at lower values.
* *Quality* has high observations at grades 5 and 6.

From the correlation values and pairwise scatterplot:

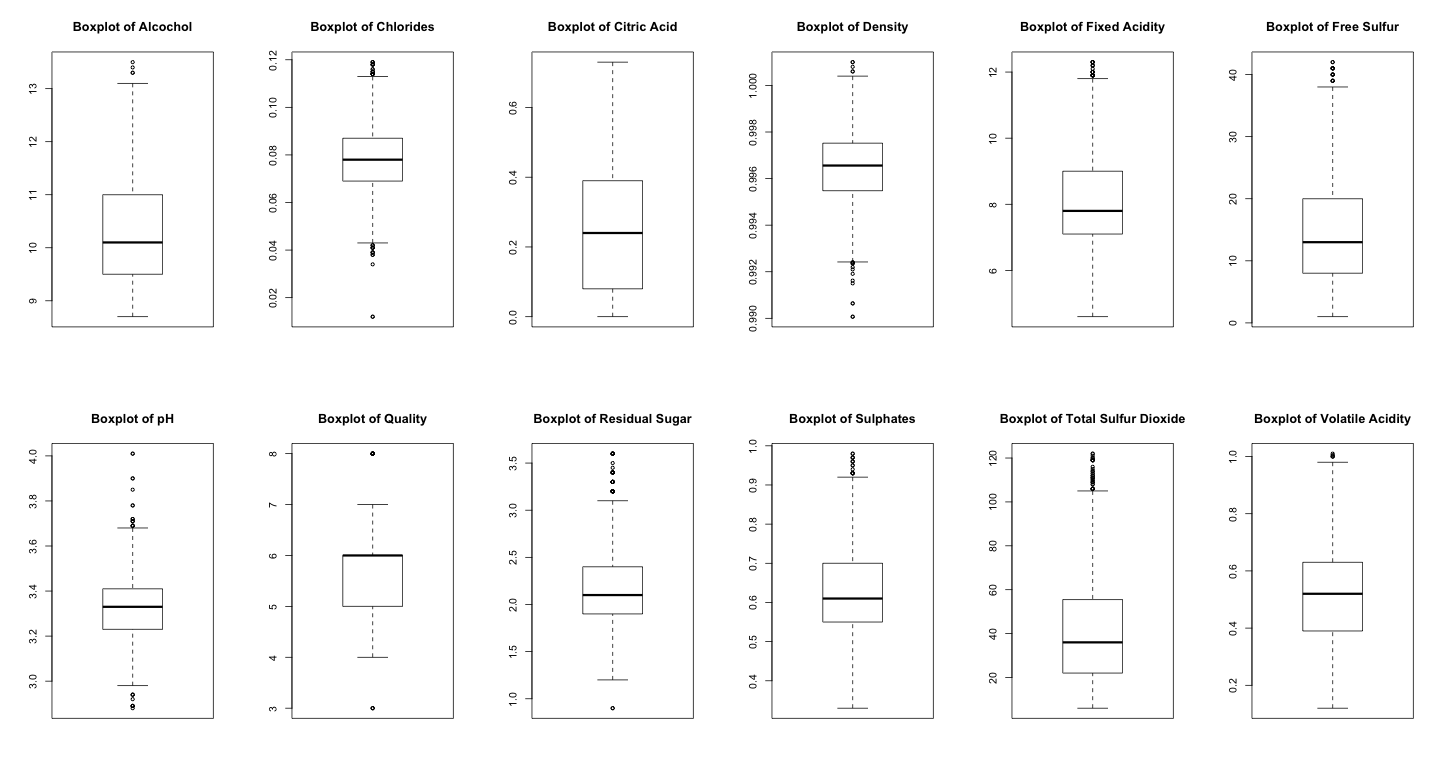
* There are strong correlations (r>0.5) between *fixed acidity vs. citric acid*, *fixed acidity vs.* *density, fixed acidity vs. pH*, *volatile acidity vs. citric acid, citric acid vs. pH.*
* There is a moderate (r = 0.48) correlation between *quality vs. alcoho*l.
* Since many variables are highly correlated, we need to check for possible collinearity.

Boxplot:





From scatter plots, we can observe that there are possible outliners in the dataset. In addition, from the histograms, we can inference that variables contain outliners on the right tail of the curve; this is also confirmed by boxplots. Thus, we consider values as outliners, only on a positively skewed variables (all expect *pH* and *density*), have points greater than third quartile + 1.5 \* interquartile on [2] on the boxplot. The number of observations decreases from 1599 to 1228. The new dataset is more symmetric and normal with some possible outliners as the boxplots shown below:



**Linear Regression**

First, we fit a model with all the variables and exam the fit of the model. Then, we remove any outliners and influential points to increase the model fit. Then remove variables cause collinearity by variance inflation factor (VIF) and select significant variables using step function. Variance inflation factor provides an index of the amount of variance increased due collinearity. And step function automatic select the most significant variables based on AIC.

The linear model with all variables:

lm(formula = quality ~ ., data = newdata)

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4.398e+01 2.690e+01 1.635 0.10225

fixed.acidity 3.086e-02 3.037e-02 1.016 0.30984

volatile.acidity -7.793e-01 1.462e-01 -5.332 1.16e-07 \*\*\*

citric.acid -2.385e-01 1.660e-01 -1.437 0.15098

residual.sugar 4.272e-02 5.021e-02 0.851 0.39506

chlorides -1.540e+00 1.338e+00 -1.151 0.24980

free.sulfur.dioxide 4.886e-03 2.618e-03 1.866 0.06225 .

total.sulfur.dioxide -2.800e-03 9.337e-04 -2.999 0.00276 \*\*

density -4.029e+01 2.743e+01 -1.469 0.14214

pH -5.410e-01 2.144e-01 -2.523 0.01177 \*

sulphates 1.784e+00 1.687e-01 10.574 < 2e-16 \*\*\*

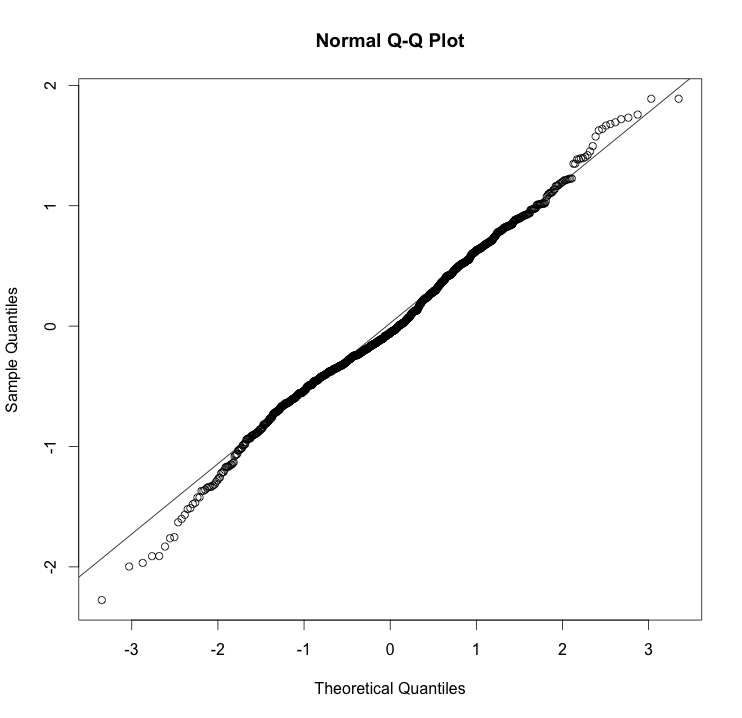
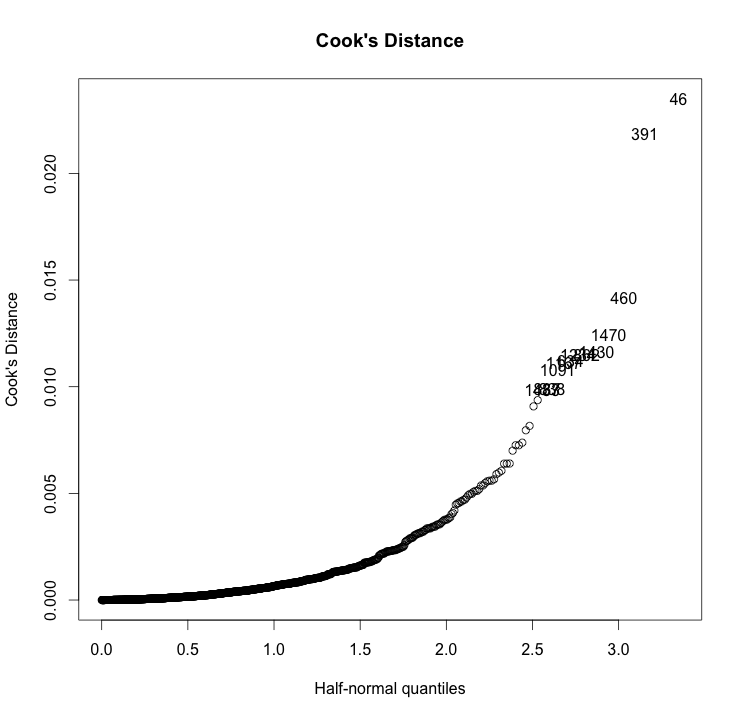
alcohol 2.672e-01 3.312e-02 8.069 1.68e-15 \*\*

Residual standard error: 0.6021 on 1216 degrees of freedom

Multiple R-squared: 0.3895, Adjusted R-squared: 0.384

F-statistic: 70.53 on 11 and 1216 DF, p-value: < 2.2e-16

The model contains 11 variables with a low R2 = 0.3895. We can see that not all variables are significant. But first, we want to see if we can improve the model fit by remove possible outliners and influential points using Q-Q plot, cook’s distance, and Jackknifes.

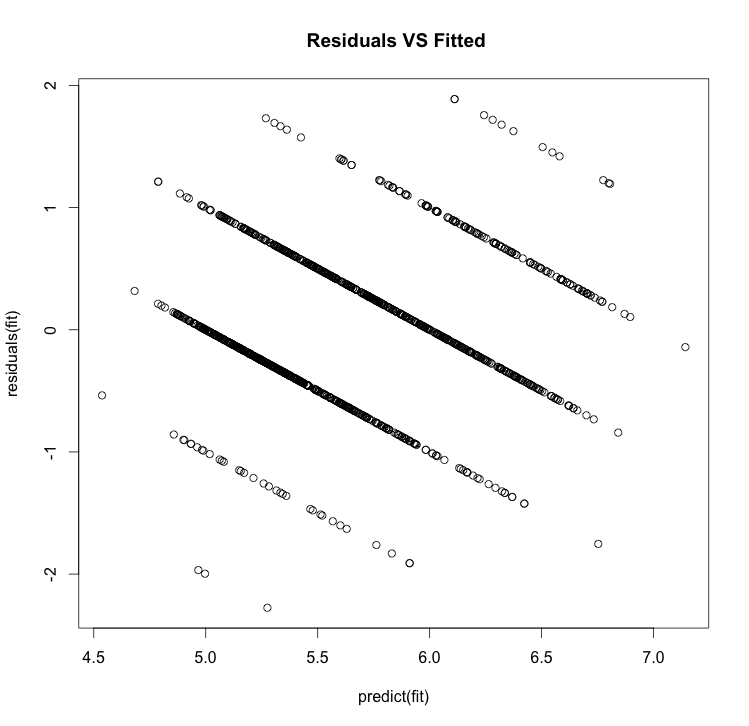
 

> qt = abs((qt(.05/(1228\*2),(1228+12-1))))

> jack[which(abs(jack) >= qt)]

named numeric(0)

Based on these three tests, there is no outliners or influential points.



Model Diagnostics:

Based on the Q-Q plot, the residuals are about normal. Residuals are not independent show on Residuals vs Fitted plot since each response only takes on finite values. Each line on the graph is corresponding to a different possible value.

From the correlation test, we suspect there is possible collinearity, so we use VIF to remove variables contributed most to variance due to collinearity.

VIF:

> vif(newdata)

fixed.acidity volatile.acidity citric.acid residual.sugar

6.939119 2.020394 3.062418 1.706242

chlorides free.sulfur.dioxide total.sulfur.dioxide density

1.358471 1.826812 2.049089 7.209567

pH sulphates alcohol quality

3.279189 1.433074 3.956742 1.638008

Based on the variance inflation factor, *density* (VIF = 7.2) contributed greatest variance to the model due of collinearity. Therefore, we remove *density.*  The VIF improve after *density* is removed as following:

> vif(newdata[,-c(8)])

fixed.acidity volatile.acidity citric.acid residual.sugar

3.236917 1.987341 3.048606 1.231061

chlorides free.sulfur.dioxide total.sulfur.dioxide pH

1.273330 1.825240 2.049029 2.167222

sulphates alcohol quality

1.329678 1.632975 1.635107

Then, we use step function (both backward and forward) to select significant variables. The backward stepwise function uses drop one variable at a time from a full model and select significant variables based on lowest possible AIC. The forward stepwise function adds one variable to null model one at a time until full model and select variables based on lowest AIC. Both result return the same variables as following:

lm(formula = quality ~ volatile.acidity + citric.acid + chlorides + free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates + alcohol, data = newdata)

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4.4524480 0.5198373 8.565 < 2e-16 \*\*\*

volatile.acidity -0.8099431 0.1399567 -5.787 9.10e-09 \*\*\*

citric.acid -0.2610026 0.1388821 -1.879 0.06044 .

chlorides -2.0188455 1.2510963 -1.614 0.10686

free.sulfur.dioxide 0.0049922 0.0026118 1.911 0.05619 .

total.sulfur.dioxide -0.0027706 0.0008996 -3.080 0.00212 \*\*

pH -0.7174565 0.1417224 -5.062 4.78e-07 \*\*\*

sulphates 1.7157545 0.1602057 10.710 < 2e-16 \*\*\*

alcohol 0.3064889 0.0195248 15.697 < 2e-16 \*\*\*

Residual standard error: 0.6019 on 1219 degrees of freedom

Multiple R-squared: 0.3884, Adjusted R-squared: 0.3844

F-statistic: 96.77 on 8 and 1219 DF, p-value: < 2.2e-16

Based on the model, variable *chloride, critic acid, and free sulfur dioxide* are not significant. Then we use drop1 function to remove the variable that is least significant.

Single term deletions

Model:

quality ~ volatile.acidity + citric.acid + chlorides + free.sulfur.dioxide +

total.sulfur.dioxide + pH + sulphates + alcohol

Df Sum of Sq RSS AIC F value Pr(>F)

<none> 441.58 -1238.0

volatile.acidity 1 12.132 453.71 -1206.7 33.4905 9.097e-09 \*\*\*

citric.acid 1 1.279 442.86 -1236.4 3.5318 0.060440 .

chlorides 1 0.943 442.52 -1237.3 2.6039 0.106860

free.sulfur.dioxide 1 1.323 442.91 -1236.3 3.6534 0.056188 .

total.sulfur.dioxide 1 3.436 445.02 -1230.5 9.4854 0.002118 \*\*

pH 1 9.284 450.87 -1214.4 25.6280 4.777e-07 \*\*\*

sulphates 1 41.549 483.13 -1129.5 114.6977 < 2.2e-16 \*\*\*

alcohol 1 89.261 530.84 -1013.9 246.4086 < 2.2e-16 \*\*\*

Based on the drop1() function, we find that deletion of *chlorides*, reduced the most in RSS and with least AIC value . Therefore, we can conclude that *chlorides* is not significant. All variables are statistical significant after *chlorides* is removed (see final model).

Final Model:

lm(formula = quality ~ volatile.acidity + citric.acid + free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates + alcohol, data = newdata)

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4.1561516 0.4866481 8.540 < 2e-16 \*\*\*

volatile.acidity -0.8410561 0.1387133 -6.063 1.78e-09 \*\*\*

citric.acid -0.2843710 0.1382157 -2.057 0.03986 \*

free.sulfur.dioxide 0.0051572 0.0026115 1.975 0.04852 \*

total.sulfur.dioxide -0.0028829 0.0008975 -3.212 0.00135 \*\*

pH -0.6928734 0.1409938 -4.914 1.01e-06 \*\*\*

sulphates 1.7048171 0.1601674 10.644 < 2e-16 \*\*\*

alcohol 0.3149751 0.0188156 16.740 < 2e-16 \*\*\*

Residual standard error: 0.6023 on 1220 degrees of freedom

Multiple R-squared: 0.3871, Adjusted R-squared: 0.3836

F-statistic: 110.1 on 7 and 1220 DF, p-value: < 2.2e-16

The final model contains seven significant variables (*volatile acidity, citric acid, free sulfur dioxide, total sulfur dioxide, pH, sulphates, and alcohol*) with R2 = 0.3871. The final model removed four variables with slightly decrease R2 (from 0.389 to 0.387). Thus, we can conclude that *chlorides, residual sugar, density, and fixed acidity* are not significant in the model. Then, we want to classify quality of the wine using these variables using different models. Since the linear model violated the independent of errors we use generalized linear model instead to classify and compare the accuracy with tree-based models.

**Classification**

From data exploratory, we observed that most of the quality value is centered on grade 5 and 6. The proportional table of the quality confirms this:

> round(prop.table(ftable(newdata$quality)),3)

3 4 5 6 7 8

0.002 0.028 0.424 0.420 0.116 0.010

Since the proportion of grade 3 and 4 are low, we combined these grades with grade 5 into one group. This is same for grades 7 and 8; therefore, we combined these grades with grade 6. Therefore, we get an obvious 50-50 cutoff between grades less or equal to 5 and grades greater than 5. Any grades less than or equal to 5 are consider as low quality wine and any grades greater 5 are considered as high quality wine. This new variable is called *factorQ*.

Then, we use cross validation to test the performance of the model. First, we separate the dataset without outliners into 2 independent datasets, training and testing. The training dataset contains 818 observations and testing dataset contains 410 observations. We use training dataset to build binomial generalized linear model, classification tree, and random forest and then use testing dataset to test its performance based on misclassification error rate. A suitable model should have low misclassification error rate.

Generalized linear model:

glm(formula = factorQ ~ volatile.acidity + citric.acid + free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates + alcohol, family = binomial, data = training)

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 5.63910 2.33592 2.414 0.015775 \*

volatile.acidity 1.76145 0.71585 2.461 0.013869 \*

citric.acid 1.32929 0.68828 1.931 0.053443 .

free.sulfur.dioxide -0.02524 0.01337 -1.888 0.059081 .

total.sulfur.dioxide 0.01819 0.00469 3.879 0.000105 \*\*\*

pH 2.03655 0.69178 2.944 0.003241 \*\*

sulphates -5.51092 0.87300 -6.313 2.74e-10 \*\*\*

alcohol -1.04319 0.10636 -9.808 < 2e-16 \*\*\*

Null deviance: 1126.54 on 817 degrees of freedom

Residual deviance: 816.82 on 810 degrees of freedom

AIC: 832.82

> pchisq(deviance(glm), df.residual(glm), lower.tail = F)

[1] 0.4264482

> (glm.table = table(yhat.glm , testing$factorQ))

yhat.glm High Low

High 66 134

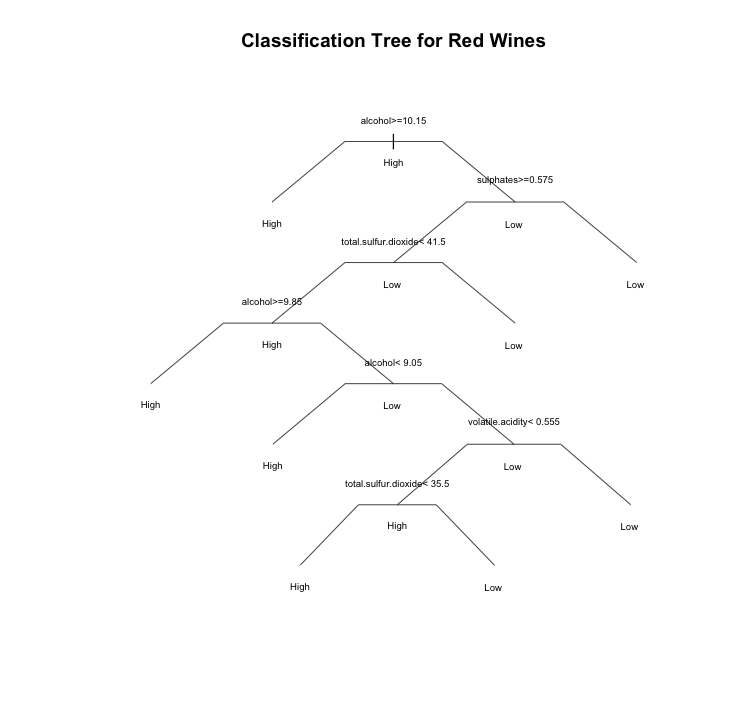
Low 156 54

> (glm.mis = 1-sum(diag(glm.table))/sum(glm.table))

[1] 0.7073171

Since *factorQ* is binomial and ni is large, so the deviance is approximate chi-squared with n –s. The p-value is 0.426; therefore the fit of the model is not adequate. Then, we classify the quality of wine being low quality if p < 0.5 and high quality if p > 0.5. The misclassification error for testing dataset is 0.707.

Classification Tree:

printcp(tree) 

Classification tree:

rpart(formula = factorQ ~ ., data = training, method = "class")

Variables actually used in tree construction:

[1] alcohol sulphates total.sulfur.dioxide volatile.acidity

Root node error: 370/818 = 0.45232

n= 818

CP nsplit rel error xerror xstd

1 0.416216 0 1.00000 1.00000 0.038473

2 0.022973 1 0.58378 0.61351 0.034612

3 0.010811 3 0.53784 0.61081 0.034565

4 0.010000 7 0.48378 0.60000 0.034373

­

> tree

n= 818

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 818 370 High (0.5476773 0.4523227)

2) alcohol>=10.15 416 92 High (0.7788462 0.2211538) \*

3) alcohol< 10.15 402 124 Low (0.3084577 0.6915423)

6) sulphates>=0.575 214 92 Low (0.4299065 0.5700935)

12) total.sulfur.dioxide< 41.5 113 48 High (0.5752212 0.4247788)

24) alcohol>=9.85 26 4 High (0.8461538 0.1538462) \*

25) alcohol< 9.85 87 43 Low (0.4942529 0.5057471)

50) alcohol< 9.05 7 0 High (1.0000000 0.0000000) \*

51) alcohol>=9.05 80 36 Low (0.4500000 0.5500000)

102) volatile.acidity< 0.555 46 20 High (0.5652174 0.4347826)

204) total.sulfur.dioxide< 35.5 36 12 High (0.6666667 0.3333333) \*

205) total.sulfur.dioxide>=35.5 10 2 Low (0.2000000 0.8000000) \*

103) volatile.acidity>=0.555 34 10 Low (0.2941176 0.7058824) \*

13) total.sulfur.dioxide>=41.5 101 27 Low (0.2673267 0.7326733) \*

7) sulphates< 0.575 188 32 Low (0.1702128 0.8297872) \*

>

> (tree.table = table(yhat.tree, testing$factorQ))

yhat.tree High Low

High 168 67

Low 54 121

> (tree.mis = 1-sum(diag(tree.table))/sum(tree.table))

[1] 0.295122

Based on the complexity parameter for optimal pruning of the classification tree, we can see that *xerror* = 0.60 is lowest with 7 splits. Therefore, the classification tree is not over-fitted and no pruned tree is needed. The fitted tree uses four variables (*alcohol, sulphates, total sulfur dioxide, and volatile acidity*). Based on the model, the misclassification error testing dataset is 29.5%.

Random Forest with 100 trees:

randomForest(formula = factorQ ~ ., data = training, ntree = 100, importance = T, proximity = T)

Type of random forest: classification

Number of trees: 100

No. of variables tried at each split: 3

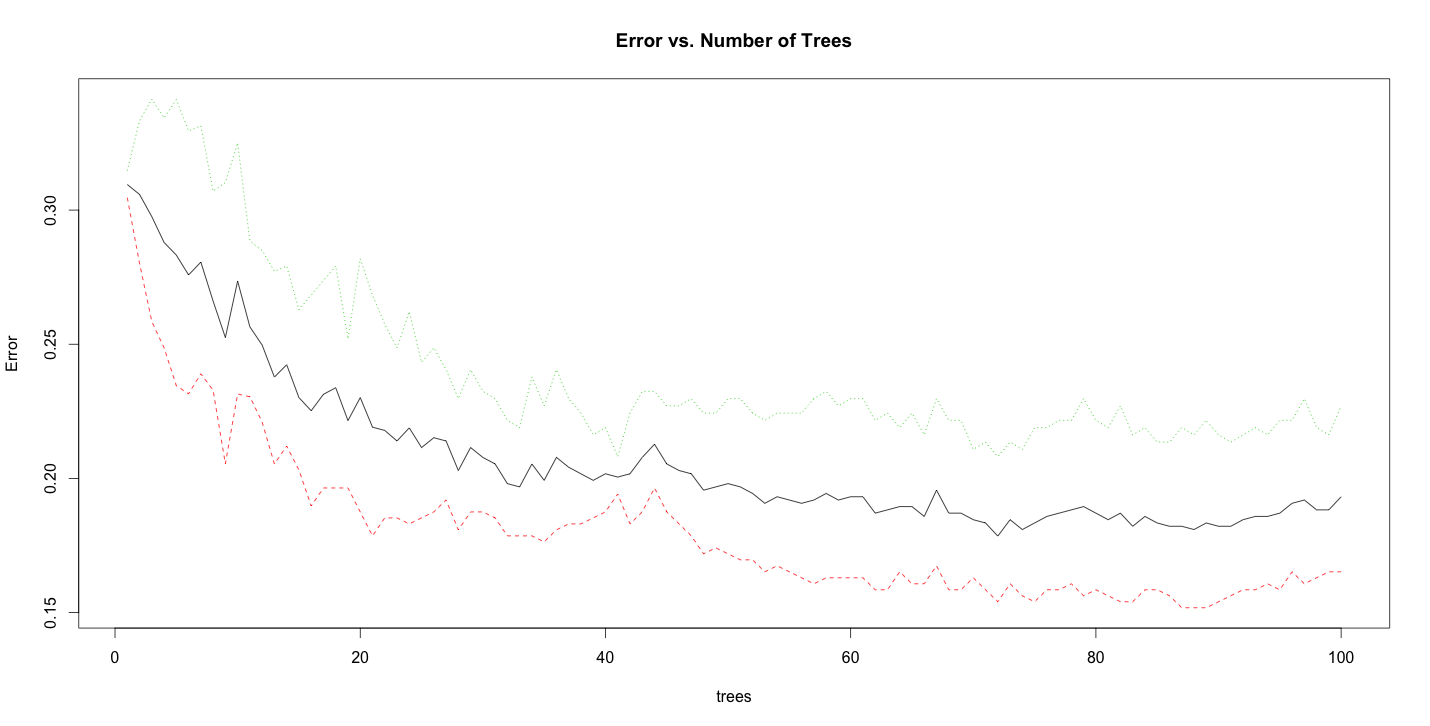
OOB estimate of error rate: 19.32%

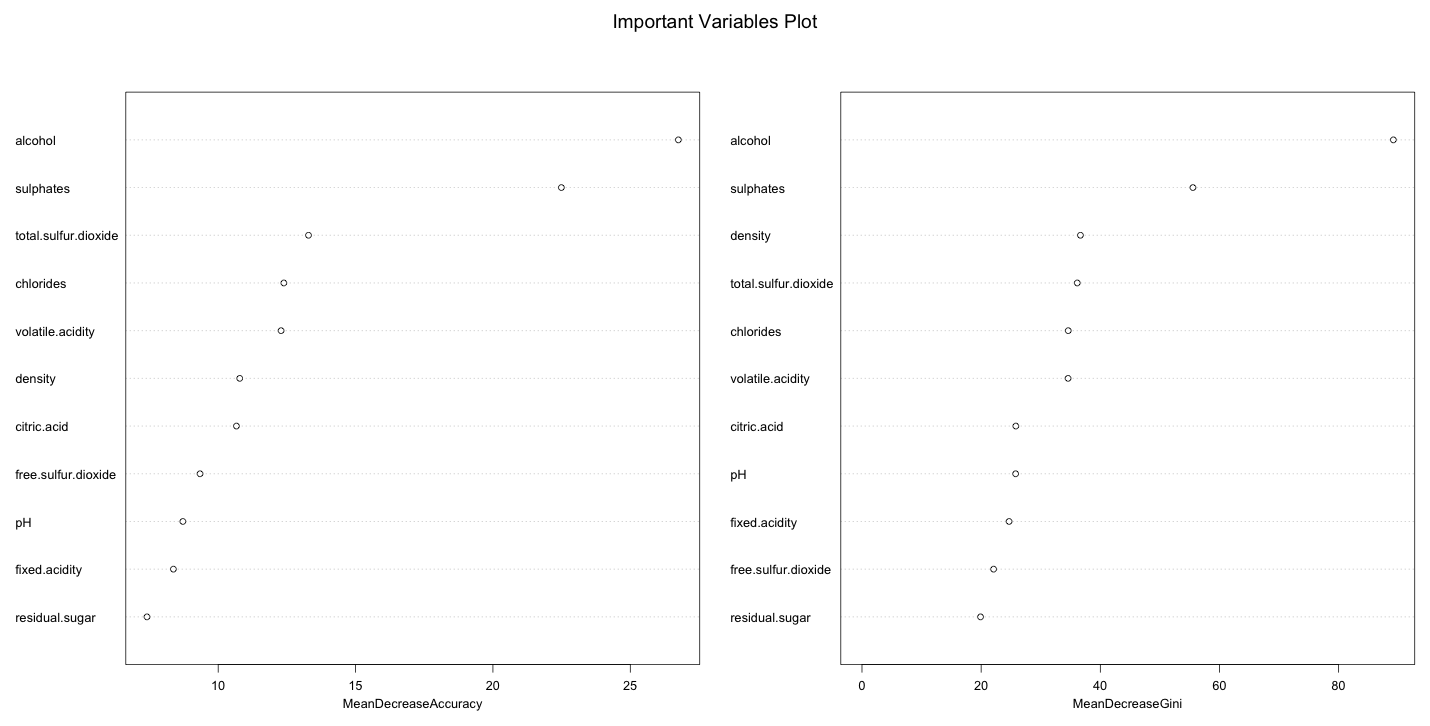
Confusion matrix:

High Low class.error

High 374 74 0.1651786

Low 84 286 0.2270270





> (forest.table = table(yhat.forest, testing$factorQ))

yhat.forest High Low

High 156 42

Low 66 146

> (forest.mis = 1-sum(diag(forest.table))/sum(forest.table))

[1] 0.2634146

Based on the error vs. number of trees, we can see that about 80 trees are sufficient for the forest model. From the important Variables Plot, the model has greatest decrease in accuracy and Gini index with *alcohol* and *sulphates.* Therefore, alcohol content and sulphate value are critical variables in the forest model. The misclassification error for the forest model for test dataset is 26.3%.

**Result**

From the linear model, we find that only seven out of eleven variables are significant for wine quality. These variables are *volatile acidity, citric acid, free sulfur dioxide, total sulfur dioxide, pH, sulphates,* and *alcohol.*  However, model doesn’t fit adequately with R2 = 0.3871. When we use generalized linear model, the misclassification error is 71% with cutoff at p= 0.5. This is expected because the model also fit inadequately with p-value = 0.426. Thus, it is not a sufficient model for predicting wine quality based on these variables. Therefore, we choose tree based classification methods. From the tree-based method, the misclassification error decreases from 29.5% to 26.5% when random forest is used against classification tree. However, the misclassification error is too large (>25%) and accuracy is to low (<75%). Therefore, tree-based models are not suitable models to predict wine quality based on chemical properties.

**Conclusion**

In conclusion, physiochemical properties are not sufficient to predict the quality of the wines. This might because the predictors are highly varied (mostly strong positively skewed) and the quality groups are concentrated at grades 5 and 6. If we want to further investigate this subject, we can include other variables, such as the age of the wine, type of yeast was used, the properties of the grape that was used. Many these factors are relevant to the quality of the wine. However, we must include alcohol content because from our analysis we see that alcohol is an essential variable in each model.

Reference

[1] 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. ISSN: 0167-9236.

[2] <https://onlinecourses.science.psu.edu/stat857/node/223>

Related R Code

red <- read.csv("~/Downloads/winequality/winequality-red.csv", sep=";")

library(GGally)

library(ggplot2)

ggpairs(red)

attach(red)

## Data Explore

par(mfrow = c(1,6))

boxplot(alcohol, main = "Boxplot of Alcochol")

boxplot(chlorides, main = "Boxplot of Chlorides")

boxplot(citric.acid,main = "Boxplot of Citric Acid")

boxplot(density,main = "Boxplot of Density")

boxplot(fixed.acidity,main = "Boxplot of Fixed Acidity")

boxplot(free.sulfur.dioxide,main = "Boxplot of Free Sulfur")

boxplot(pH, main = "Boxplot of pH")

boxplot(quality,main = "Boxplot of Quality")

boxplot(residual.sugar, main = "Boxplot of Residual Sugar")

boxplot(sulphates,main = "Boxplot of Sulphates")

boxplot(total.sulfur.dioxide,main = "Boxplot of Total Sulfur Dioxide")

boxplot(volatile.acidity,main = "Boxplot of Volatile Acidity")

par(mfrow = c(1,1))

## Data Cleaning

red2 = red[,-c(8,9,12)]

a = 0

for( i in 1:(ncol(red2))){

a[i] = (quantile(red2[,i],0.75) + 1.5\*IQR(red2[,i], 0.75))

}

index = matrix(0, nrow(red2), ncol(red2))

for (i in 1:nrow(red2))

for(j in 1:(ncol(red2))){

if(red2[i,j] > a[j]) index[i,j] = 1}

tot = apply(index, 1, sum)

b = 0

j = 1

for(i in 1:nrow(red2)) { if( tot[i]> 0){ b[j]=i

j=j+1} else j = j}

newdata = red[-b,]

##Check cleaned dataset

par(mfrow = c(2,6))

boxplot(newdata$alcohol, main = "Boxplot of Alcochol")

boxplot(newdata$chlorides, main = "Boxplot of Chlorides")

boxplot(newdata$citric.acid,main = "Boxplot of Citric Acid")

boxplot(newdata$density,main = "Boxplot of Density")

boxplot(newdata$fixed.acidity,main = "Boxplot of Fixed Acidity")

boxplot(newdata$free.sulfur.dioxide,main = "Boxplot of Free Sulfur")

boxplot(newdata$pH, main = "Boxplot of pH")

boxplot(newdata$quality,main = "Boxplot of Quality")

boxplot(newdata$residual.sugar, main = "Boxplot of Residual Sugar")

boxplot(newdata$sulphates,main = "Boxplot of Sulphates")

boxplot(newdata$total.sulfur.dioxide,main = "Boxplot of Total Sulfur Dioxide")

boxplot(newdata$volatile.acidity,main = "Boxplot of Volatile Acidity")

par(mfrow = c(1,1))

par(mfrow = c(1,1))

attach(newdata)

#full model

fit = lm(quality~., data = newdata)

summary(fit)

## Outlier and influential points

plot(predict(fit), residuals(fit), main = " Residuals VS Fitted")

qqnorm(residuals(fit))

qqline(residuals(fit))

library(faraway)

cook = cooks.distance(fit)

halfnorm(cook,13, labs =names(cook), ylab = "Cook's Distance", main = "Cook's Distance")

## 46, 319

jack = rstudent(fit)

qt = abs((qt(.05/(1228\*2),(1228+12-1))))

jack[which(abs(jack) >= qt)]

##VIF

which(sqrt(vif(newdata))>2)

vif(newdata[,-c(1,8)])

fit2 = lm(quality~. -density-fixed.acidity, data = newdata)

summary(fit2)

##Step function

fit3 = step(fit2)

null=lm(quality~1, data=newdata)

fit4 = step(null, scope=list(lower=null, upper= fit2, direction="forward"))

summary(fit3)

summary(fit4)

## both return the same model

## Drop1()

drop1(fit3, test = "F")

fit5 = update(fit3,~.-chlorides )

summary(fit5)

## Proportion of quality

round(prop.table(ftable(newdata$quality)),3)

##Assign factor of quality

newdata$factorQ = ifelse(newdata$quality <= 5, "Low", "High")

newdata$factorQ = factor(newdata$factorQ)

## sample 2/3 of data as training set and 1/3 of data as testing set

indexes = sample(1:nrow(newdata), (2/3)\*nrow(newdata), replace = F)

training = newdata[indexes, -c(12)]

testing = newdata[-indexes, -c(12)]

##generialized linear model

glm = glm(factorQ ~ volatile.acidity + citric.acid + free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates + alcohol, data = training, family = binomial)

summary(glm)

pchisq(deviance(glm), df.residual(glm), lower.tail = F)

probs = predict(glm, testing, type = "response")

yhat.glm = ifelse(probs <= 0.5, "Low", "High")

(glm.table = table(yhat.glm , testing$factorQ))

(glm.mis = 1-sum(diag(glm.table))/sum(glm.table))

# Classification Tree

library(rpart)

tree = rpart(factorQ ~., method = "class", training)

plot(tree,compress=T, uniform=T, branch= 0.4, margin = 0.1, main="Classification Tree for Red Wines")

text(tree, all=TRUE, cex = .6)

printcp(tree)

yhat.tree = predict(tree, testing, type = "class")

(tree.table = table(yhat.tree, testing$factorQ))

(tree.mis = 1-sum(diag(tree.table))/sum(tree.table))

# Random Forest

library(randomForest)

forest = randomForest(factorQ ~., data = training, ntree = 100, importance = T, proximity =T)

print(forest)

yhat.forest = predict(forest, testing)

(forest.table = table(yhat.forest, testing$factorQ))

(forest.mis = 1-sum(diag(forest.table))/sum(forest.table))

plot(forest, main="Error vs. Number of Trees")

varImpPlot(forest, main="Important Variables Plot", cex=0.8)