University of Catania

Department of Economy

Master of Data Science for Management

Analysis and Prediction Report On
Red Wine Quality Dataset

Anahita Esfandiaryfard

Thamires de Souza Oliveira

Prof. S. Ingrassia

Academic Year 2020- 2021

Introduction

This report is based on the *red wine quality dataset* available on this link from UCI machine learning repository: https://archive.ics.uci.edu/ml/datasets/wine+quality

The dataset is included, related to red wine samples from the north of Portugal, that produces wines of exceptional quality, equal to a distinct and diversified gastronomy. It is in this region that the world famous Wine of Porto is made, and also the much appreciated DOC douro.

The purpose of the results of this report is to help *winemakers*, *oenophiles* and *sommeliers* from around the world to understand the best physical factors that lead wines to be classified on a scale from good to bad, in relation to their quality.

- In this report we are looking for the research and comprehension of which factors lead the wine to be classified as high or low quality.
- We will also experiment with different classifications methods that yelds the highest accuracy,
- We will also provide the best techniques to calculate it.
- We will evaluate what factors should a company increase or reduce in order to have high quality wine.
- We will also give advice to professionals in the field so that they have the best criteria when it comes to classifying red wines.

With all the objectives and guidelines explained, let's begin the analysis:

Data Description

The dataset that we will use is "Wine Quality" dataset, which exists in the project folder. The data set is divided into three data sets:

+ Train data: about 60% of the units of the original dataset

+ Validation data: about 20% of the units of the original dataset

+ Test data: about 20% of the units of the original dataset.

This data will allow us to create different regression models to determine how different independent variables help predict our determined dependent variable, which is *quality*.

Fields include

Fixed Acidity: (tartaric acid - g / dm^3) - non-volatile acids that do not evaporate readily;

Volatile Acidity: (acetic acid - g / dm^3) - high acetic acid in wine which leads to an unpleasant vinegar taste;

Citric Acid: (g / dm^3) - acts as a preservative to increase acidity. When in small quantities, adds freshness and flavor to wines;

Residual Sugar: (g / dm^3) - amount of sugar remaining after fermentation stops.

Chlorides: (sodium chloride - g / dm^3) - the amount of salt in the wine;

Free Sulfur Dioxide: (mg / dm^3) - it prevents microbial growth and the oxidation of wine;

Total Sulfur Dioxide: (mg / dm^3) - amount of free + bound forms of SO2;

Density: (g / cm^3) - mass per unit volume of wine;

pH: describes the level of acidity on a scale of 0–14. Most wines are always between 3–4 on the pH scale;

Alcohol: (potassium sulphate - g / dm3) - available in small quantities in wines makes the drinkers sociable;

Sulphates: (potassium sulphate - g / dm3) - additive that contributes to SO2 levels and acts as an antimicrobial and antioxidant

Quality: which is the output variable/predictor. Integer numbers ranging in a scale from 3 to 8.

We start our work by importing train data set and apply some data analysis on it, the data set contained 975 observations of 13 variables, where no noun value exits in the data

Data Exploration and Transformation

Lets take a look at the summary of data we have, In the below table we can check some parameters of each variable, such as mean, minimum and maximum value.

Variables	N.	Mean	SD	Min	Q1	Median	Q3	Max
fixed.acidity	957	8.32	1.71	4.70	7.10	8.00	9.20	15.50
volatile.acidity	957	0.53	0.18	0.12	0.40	0.52	0.64	1.58
citric.acid	957	0.27	0.19	0.00	0.10	0.26	0.42	1.00
residual.sugar	957	2.50	1.25	1.20	1.90	2.20	2.60	13.80
chlorides	957	0.09	0.04	0.01	0.07	0.08	0.09	0.61
free.sulfur.dioxide	957	16.08	10.13	1.00	8.00	14.00	22.00	66.00
total.sulfur.dioxide	957	47.21	33.11	6.00	23.00	39.00	64.00	289.00
density	957	1.00	0.00	0.99	1.00	1.00	1.00	1.00
pH	957	3.31	0.16	2.74	3.21	3.31	3.40	4.01
sulphates	957	0.66	0.17	0.33	0.55	0.62	0.73	2.00
alcohol	957	10.39	1.05	8.40	9.50	10.10	11.00	14.00
quality	957	5.63	0.76	4.00	5.00	6.00	6.00	7.00

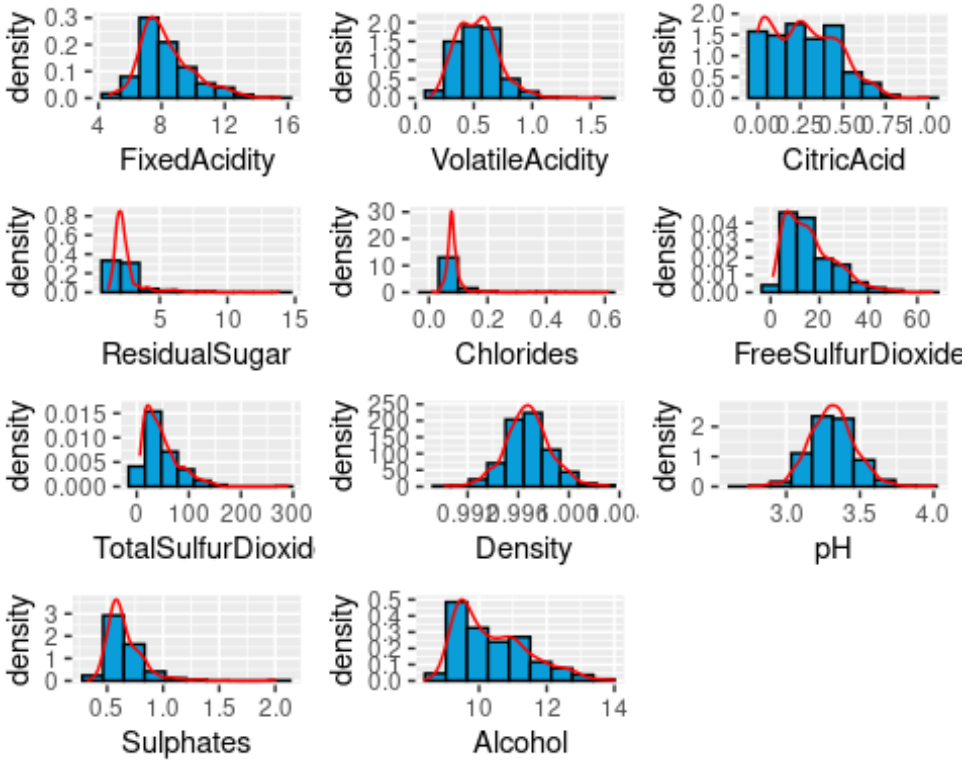
As we can check, we have some interesting insights from it like:

- The *volatile.acidity* that leads to an unpleasant taste of wine have a mean of 0.53 which is really low with a standard deviation of only 0.18, but the minimum value we can find of this characteristic on this dataset is 0.12. Which bring us a question if this wine, combined with other good characteristics could bring to it a high quality.
  Looking at the dataset, the wines with that characteristics are with the ID *950* and *949* and both have quality 7, what consists in a good quality wine. But two only two units are not enough to come up with an conclusion, so we need to check other features.

- The *citric.acid* in this dataset, that when in small quantities add frashness and flavor to wines has its minimun value as 0.00 what consists in a wine that doesnt have this characteristic. When looking into the dataset, there is 71 wines that have 0 *citric.acid* with qualities that goes from 4 to 7. So by now, it means that this field is not a a strong adviser for quality classification.

- The *alcohol* has its maximun value as 14, when we look at the data the 2 wines with this numbers have qualities 6 and 7, which is consider high. What indicates that alcohol may be a good characteristic related to quality, but probably not the most important one.

- Another good observation is that is this dataset he mean is usually greater than the median, this kind of observations indicates that there are *outliers* in the data.

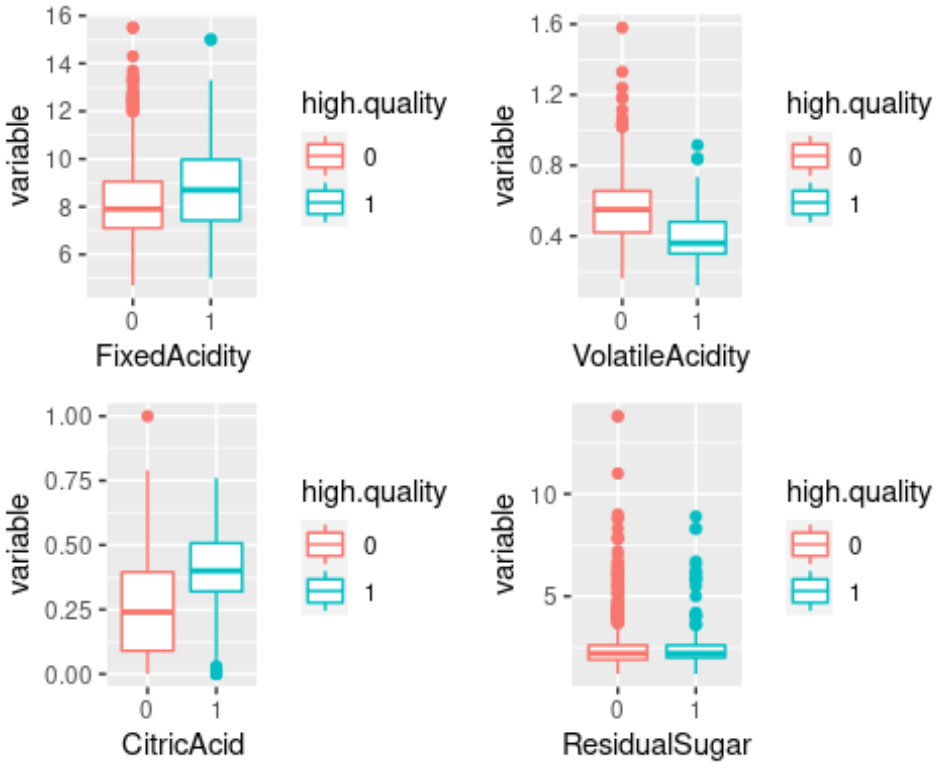


The above histograms shows us distribution of data in our dataset. As we can see there are lots of outliers in the dataset which may cause on wrong prediction for our models. So we will test our model on a validation dataset to make sure about the result.

Also we can see pH and Density has normal distribution but other variables are skewed right.

We continue with the box plot, because the purpose of this analysis is to find high quality wine, we need to define what quality consider high and what low, so we add a "high.quality" field to our data. Here we consider wines with quality less than 6 as low quality wine.

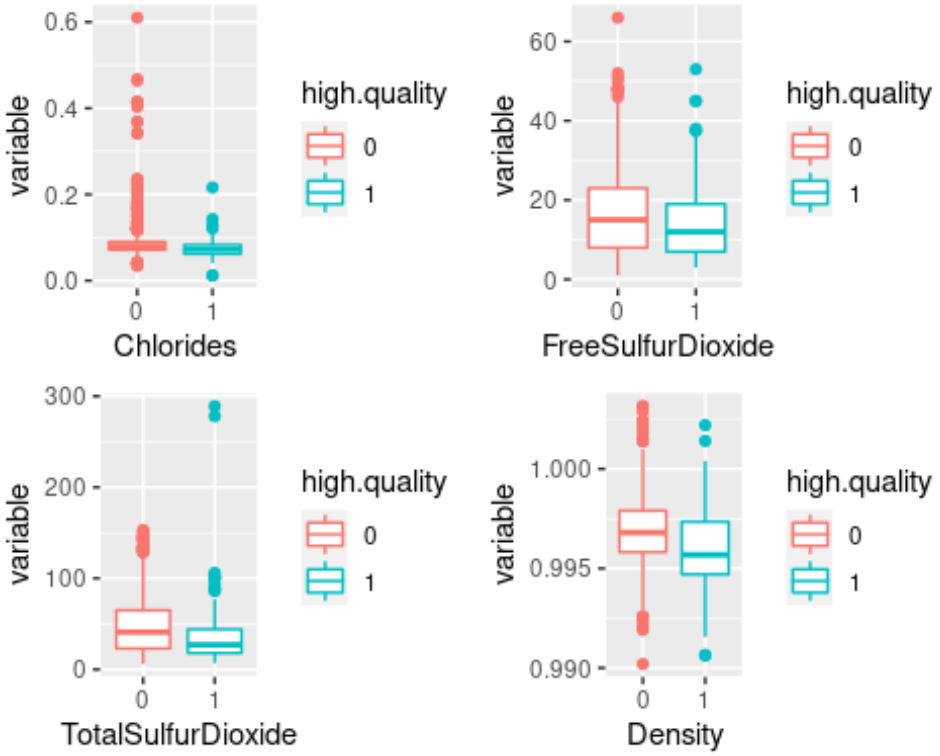
Now, below box plot can shows us the distribution of each variable in both high and low quality wines.

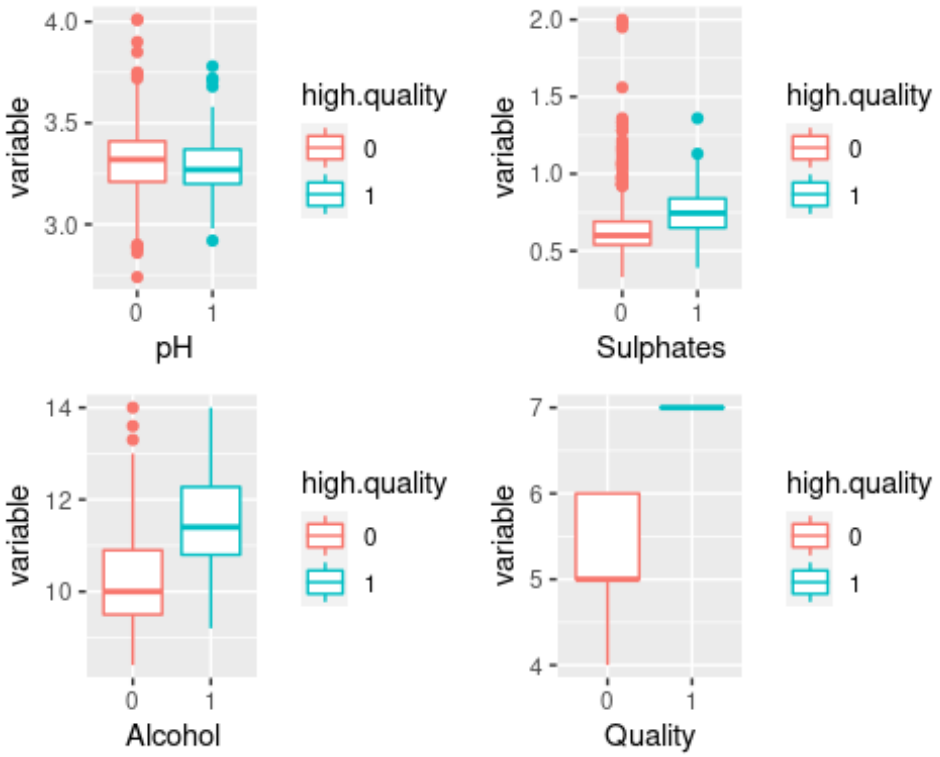


Acids are one of 4 fundamental traits in wine. Acidity gives a wine its tart and sour taste. Fundamentally speaking, all wines lie on the acidic side of the pH spectrum, and most range from 2.5 to about 4.5 pH (7 is neutral).

This is a important factor in quality of wine. In this data set we have the data for Fixed and Volatile acidity. Most acids involved with wine are fixed which has the effect on different taste of wine on the other hand volatile acidity is the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste.

As the graph shows us in Fixed acidity the median line in box plot in group high quality is higher than the other group it means that more Fixed acidity leads to higher quality and in the Volatile acidity plot the median line of high quality box plot is lower than the other one it means that the higher Volatile acidity leads to lower quality.



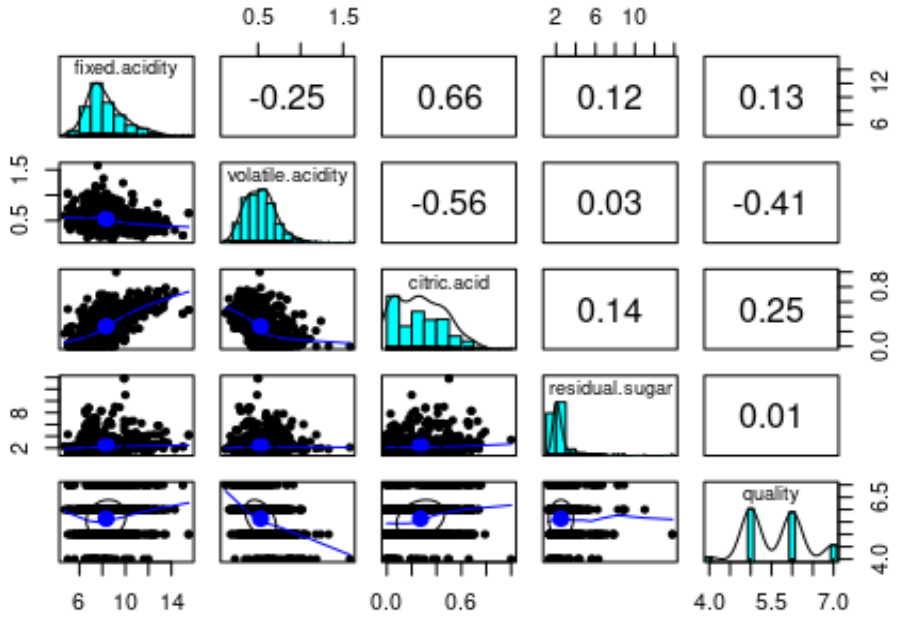


- As we can check, all the variables contains outliers, as we previously suspected.
- *Quality* has most values concentrated in the categories 5, 6 and 7.
- Some of the variables, like *free sulphur dioxide* and *density*, have a few outliers but these are very different from the rest.
- Mostly outliers are on the larger side.
- Alcohol has an irregular shaped distribution but it does not have pronounced outliers.

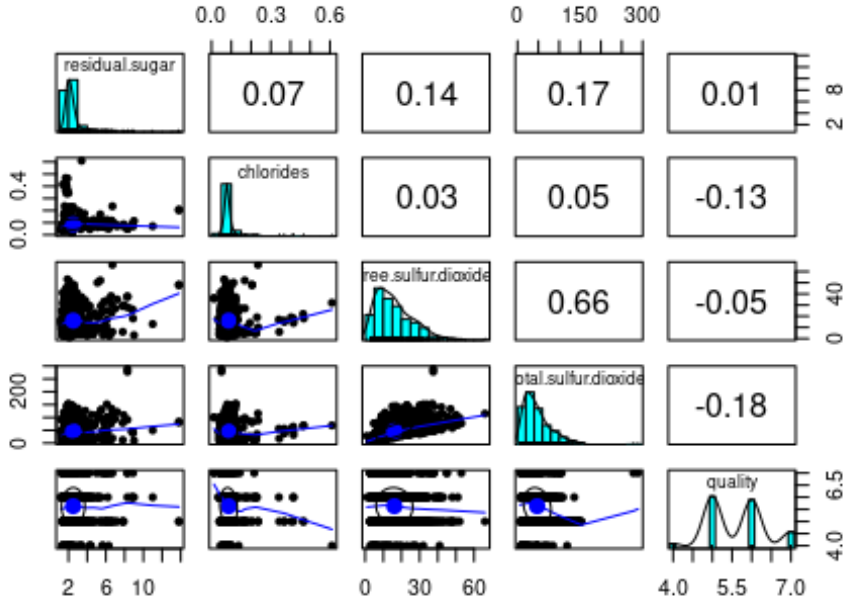
Looking at the data we can see the density of each variable in each quality group, also we can assume if the higher rate of each variable is related to high quality or not.For example in pH, we can see that the median value in group 0 is higher than group 1 so we can say that lower pH can results better quality

Now lets run a correlation matrix function to check which variables are most likely to affect the quality of red wine the most.

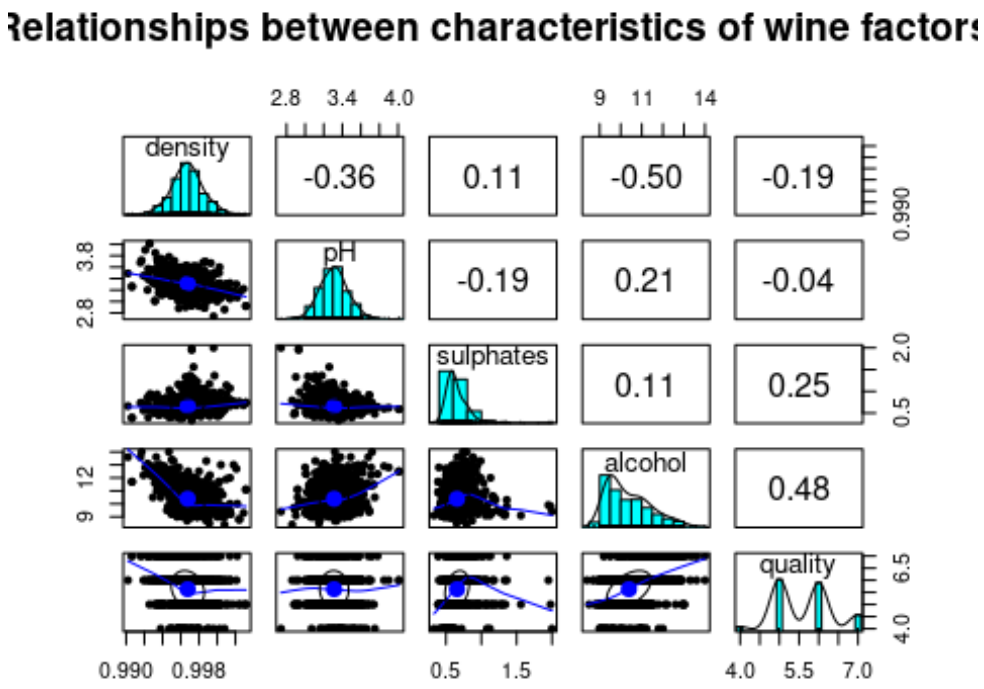
This is a preliminary analysis of the data in the original space. The upper triangle of the matrix there are the coefficients of correlation between variables, In the lower triangle there are the scatterplots of data and on the main diagonal there is the non-parametric density of the data. If we look at the plot, we can see that there some variables has no correlation to each other as their correlation is less than 0.1 and is close 0 like correlation between residual sugar and choloride (0.07) but some variables are quite related. In particular, the variables fixed acidity and density are the most positively correlated (0.67) and the variables fixed acidity and pH are the most negatively correlated (-0.69).



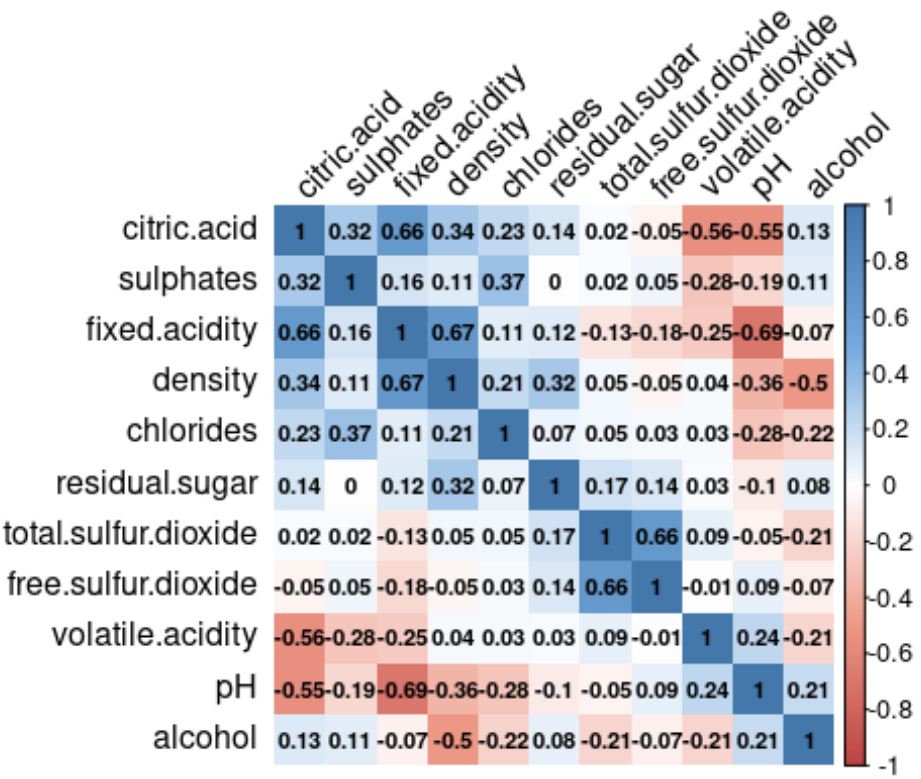
As we can see in this plot Volatile acid has a negative corelation with quality which it means when we increase this acid quality decrease



This plot also shows us that chlorides, free sulfur dioxide and total sulfur dioxide have negative correlation with quality.



And this plot shows us a negative correlation between density and ph.



As we can check with the graph and numbers above, the variable that has the most strong and positive correlation with our predictor *quality* is *alcohol*. Doing some researches about the relation of alcohol and the quality and acceptance of a wine taste, in reality, the alcoholic strength of wine is just one of the characteristics that must be observed. The important thing is that it is tasty and complete, and this can happen both in bottles with a graduation above 14% and in the lower ones.

But we cannot deny the strong correlation between alcohol and the wine quality, even if we cannot put it as the only strong characteristic.

Following with a still positive, but not that strong relation with *sulphates* and *citric.acid*. As we checked, we can even have good qualities wine with no *citric.acid* at all.

We also can see that the variable *volatile.acidity* has a strong but negative correlation with *quality*. It makes sense cause this is a high acetic acid in wine which leads to an unpleasant vinegar taste.

Wines with high volatile acidity are commonly considered to be undesirable because of their marked sour taste. Although unfavorable harvests can lead to the production of a wine with high volatile acidity, the problem actually arises during fermentation. Certain microorganisms present in wine can generate an excess of acid during the aging process. The bacteria that metabolize alcohol and convert it to acetic acid (such as Acetobacter aceti, A. pasteurianus and Gluconobacter oxydans) are primarily responsible for volatile acidity.

Many *winemakers* are using techniques to decrease the volatile acidity in a good way and it makes really sense in order to increase the wine quality.

To finalize the correlation analysis, we also have the information that *residual.sugar* and *PH* almost doesn't have any correlation with quality at all.

Modeling

Logistic Regression

To first model our data, we will use the generalized linear model that is used to express the relation between covariates X and response Y in a linear, additive manner. We apply *glm* to model our data and find out what variables are related to our purpose, which is finding high quality wine. With this, lets fit our data using all the variables first:

```
Call:
glm(formula = high.quality ~ ., family = binomial, data = wine)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-1.9874  -0.4136  -0.2007  -0.1163   3.2015

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           4.271e+02  1.412e+02   3.025 0.002489 **
## fixed.acidity         5.214e-01  1.609e-01   3.240 0.001194 **
## volatile.acidity     -3.420e+00  1.049e+00  -3.262 0.001108 **
## citric.acid           4.044e-02  1.125e+00   0.036 0.971317
## residual.sugar        3.302e-01  1.074e-01   3.076 0.002101 **
## chlorides            -1.226e+01  5.138e+00  -2.387 0.016977 *
## free.sulfur.dioxide   6.525e-03  1.563e-02   0.418 0.676259
## total.sulfur.dioxide -9.084e-03  5.180e-03  -1.754 0.079497 .
## density              -4.468e+02  1.442e+02  -3.098 0.001945 **
## pH                    1.367e+00  1.303e+00   1.050 0.293877
## sulphates             3.775e+00  7.011e-01   5.384 7.27e-08 ***
## alcohol               6.081e-01  1.751e-01   3.474 0.000513 ***

AIC: 526.89
```

This initial model, containing all the variables has an AIC equal to 526.89 and we can check that the variable *citric.acid* has a p-value really high and not statistically significant. Because of that we model again our data taking off this variable and the results are below:

```
Call:
glm(formula = high.quality ~ . - citric.acid, family = binomial, data = wine)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.9822	-0.4133	-0.2008	-0.1158	3.2024

Coefficients:

```
                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           4.264e+02  1.399e+02   3.048 0.002304 **
## fixed.acidity         5.228e-01  1.563e-01   3.345 0.000824 ***
## volatile.acidity     -3.441e+00  8.619e-01  -3.993 6.53e-05 ***
## residual.sugar        3.303e-01  1.073e-01   3.079 0.002074 **
## chlorides            -1.224e+01  5.087e+00  -2.406 0.016136 *
## free.sulfur.dioxide   6.478e-03  1.557e-02   0.416 0.677349
## total.sulfur.dioxide -9.056e-03  5.122e-03  -1.768 0.077023 .
## density              -4.461e+02  1.428e+02  -3.123 0.001791 **
## pH                    1.358e+00  1.279e+00   1.062 0.288200
## sulphates             3.773e+00  6.999e-01   5.391 7.00e-08 ***
## alcohol               6.100e-01  1.674e-01   3.645 0.000267 ***
## ---
```

AIC: 524.89

This new model without the variable *citric.acid* has a lower AIC which indicates that is a better-fit model, but we can check that the variable *free.sulfur.dioxide* has a high p value that is not statically significant, so we need to do another model without this variable, as we can see below:

Call:

```
glm(formula = high.quality ~ . - citric.acid - free.sulfur.dioxide, family =
binomial, data = wine)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.9683	-0.4158	-0.2011	-0.1186	3.1915

Coefficients:

```
                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           4.295e+02  1.396e+02   3.077 0.002094 **
## fixed.acidity         5.256e-01  1.561e-01   3.368 0.000757 ***
## volatile.acidity     -3.463e+00  8.606e-01  -4.024 5.73e-05 ***
## residual.sugar        3.268e-01  1.075e-01   3.041 0.002359 **
## chlorides            -1.212e+01  5.066e+00  -2.392 0.016748 *
## total.sulfur.dioxide -7.678e-03  3.878e-03  -1.980 0.047739 *
## density              -4.494e+02  1.425e+02  -3.153 0.001615 **
## pH                    1.447e+00  1.259e+00   1.149 0.250468
## sulphates             3.777e+00  6.992e-01   5.401 6.62e-08 ***
## alcohol               6.040e-01  1.667e-01   3.622 0.000292 ***
```

AIC: 523.06

In this case we have a drop from AIC of 524.89 to 523.06, which is not so much, and that means that the models are similar in terms of AIC. And in this case we can note that we still have one variable with a high p value, this variable is *ph.* So we need to fit another model removing this variable:

Call:

```
glm(formula = high.quality ~ . - citric.acid - free.sulfur.dioxide - pH,
family = binomial, data = wine)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9751  -0.4255  -0.2013  -0.1186   3.2058

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           3.525e+02  1.215e+02   2.903 0.003699 **
## fixed.acidity         3.930e-01  1.039e-01   3.783 0.000155 ***
## volatile.acidity     -3.481e+00  8.656e-01  -4.022 5.78e-05 ***
## residual.sugar        2.860e-01  1.015e-01   2.818 0.004836 **
## chlorides            -1.309e+01  5.059e+00  -2.588 0.009663 **
## total.sulfur.dioxide -8.103e-03  3.862e-03  -2.098 0.035892 *
## density              -3.668e+02  1.220e+02  -3.007 0.002638 **
## sulphates             3.596e+00  6.788e-01   5.297 1.18e-07 ***
## alcohol               6.870e-01  1.498e-01   4.587 4.49e-06 ***

AIC: 522.37
```

By doing steps above now we have a model with AIC 522.37 which is smaller than the AIC of initial model, we can see the p value of all coefficients are significantly small so we can reject null hypothesis and realize these variables are completely related.

With that, we will take off our model the variables *citric acid, free sulfur dioxide and Ph,* because the smallest p-value associated with it are not statistically significant.

Lets see the accuracy of our model like that in train dataset, looking at the confusion matrix below:

		True Values		Total
		High	Low	
Predicted Values	High	797	79	876
	Low	30	51	81
Total		827	130	957

```
delta = 11.38976
```

The above tables shows that 79 wrong prediction for low quality and 30 wrong prediction for high quality which in general 11.38% error we have in glm which is quite good. Lets test our model on validation dataset and see the accuracy:

		True Values		Total
		High	Low	
Predicted Values	High	270	26	296
	Low	10	18	28
Total		280	44	324

```
delta = 11.11111
```

Looking at confusion matrix above, we can see that the diagonal elements of the confusion matrix indicate correct predictions, while the off-diagonals represent incorrect predictions. In general 270 +18 data predicted correctly and 26+10 data is wrongly predicted. and delta is 11.1 which is small and its close to the train delta. We can say this result is good enough so we can say citric acid, free sulfur dioxide, pH are not really related to our result so we can make our models without using them.

Linear Discriminant Analysis

The Linear Discriminant Analysis, or more commonly names as LDA, is a method used in statistics and other fields, to find a linear combination of features that characterizes or separates two or more classes of objects or events. LDA is not just a dimension reduction tool, but also a robust classification method.

We will model this method in our data, taking off the not significant variables that we previously checked on glm:

Call:

lda(high.quality ~ . - citric.acid - free.sulfur.dioxide - pH, data = wine)

Prior probabilities of groups:

```
     0         1
0.8641588 0.1358412
```

Coefficients of linear discriminants:

```
                        LD1
## fixed.acidity         0.34084076
## volatile.acidity     -1.64001287
## residual.sugar        0.21623773
## chlorides            -5.29607760
## total.sulfur.dioxide  -0.00242651
## density             -345.31401117
## sulphates             2.34569120
## alcohol               0.43587051
```

The LDA output indicates that 86% of the training observation correspond to low quality wine and 13% of data is related to high quality data. The coefficients of linear discriminant output show how factors can effect on better quality. Like for example coefficients with positive value can increase the quality if we increase that value as much as calculated. and vice versa. These values could suggest that the variable volatile. Acidity for example might have a slightly greater influence on low quality (0.55) than on high quality group (0.39).

Let's see the accuracy of our model in train dataset:

		True Values	
		High	**Low**
Predicted Values	**High**	789	70
	Low	38	60

```
[1] 11.28527
```

The above tables show that 70 wrong predictions for low quality and 38 wrong predictions for high quality which in general 11.28% error we have in lda which is quite good. Let's test our model on validation dataset and see the accuracy:

	True Values	
	High	**Low**

Predicted Values	**High**	267	24
	Low	13	20

```
[1] 11.41975
```

Confusion matrix above shows us the summary of prediction using LDA, we can see that delta is not much different than glm method. Also model predicted 267+20 correct predictions which is not bad in compare to 24+13 wrong prediction. delta is 11.41 which is small and its close to the train delta. We can say this model is quite good .Let see if QDA can create a better model:

Call:
qda(high.quality ~ . - citric.acid - free.sulfur.dioxide - pH, data = wine)

Prior probabilities of groups:

0	1
0.8641588	0.1358412

Lets see the accuracy of our model in train dataset:

		True Values	
		High	**Low**
Predicted Values	**High**	762	54
	Low	65	76

```
[1] 12.43469
```

The above tables shows that 54 wrong prediction for low quality and 65 wrong prediction for higtquality which in general 12.43% error we have in qda which is quite good. Lets test our model on validation dataset and see the accuracy:

	True Values

		High	Low
Predicted Values	High	247	21
	Low	33	23

```
[1] 16.66667
```

Confusion matrix above shows us the summary of prediction using QDA, Model predicted 247+23 correct predictions which is not bad in compare to 21+33 wrong prediction. delta is 16.67 which is small but its quite more than the train delta which is 12.43. We can say this model is not good .Also the result from LDA is better than this result. So far we can say LDA model our data better. We continue or modeling with other methods to see if we can have better results.

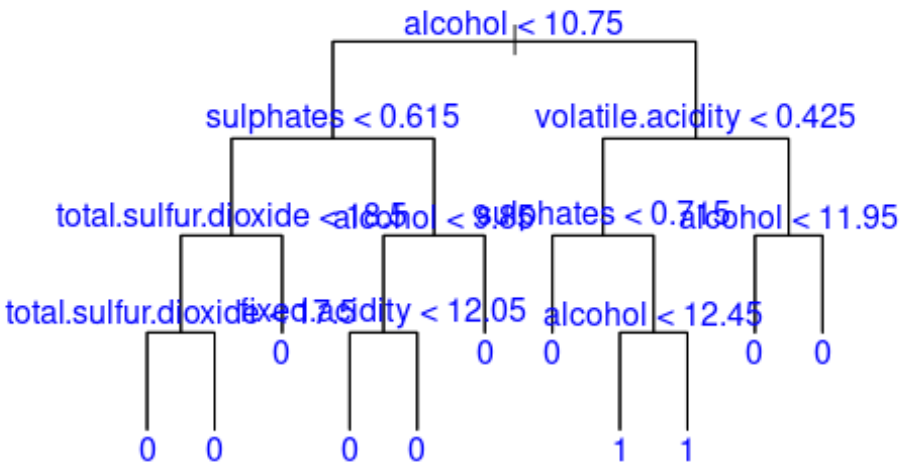
Random Forest

Random forests are a joint learning technique that builds on decision trees. Random forests involve creating multiple decision trees using datasets initialized from the original data and randomly selecting a subset of variables at each step of the decision tree. The model then selects the mode of all predictions from each decision tree.

Classification tree:
tree(formula = high.quality ~ . - citric.acid - free.sulfur.dioxide - pH, data = wine, control = setup)

Variables actually used in tree construction:
[1] "alcohol" "sulphates" "total.sulfur.dioxide"
[4] "fixed.acidity" "volatile.acidity"

Number of terminal nodes: 11
Residual mean deviance: 0.4925 = 465.9 / 946
Misclassification error rate: 0.1129 = 108 / 957



So Lets see the accuracy of our model in train dataset:

		True Values	
		High	**Low**
Predicted Values	**High**	801	79
	Low	26	51

```
[1] 10.97179
```

The above tables shows that 79 wrong prediction for low quality and 26 wrong prediction for high quality which in general 10.91% error we have in tree which is quite good.

Lets test our model on validation dataset and see the accuracy:

		True Values	
		High	**Low**
Predicted Values	**High**	262	33
	Low	18	11

```
[1] 15.74074
```

Confusion matrix above shows us model predicted 262+11 correct predictions which is not bad in compare to 33+18 wrong prediction. Delta is 15.74 which is small but its quite more than the train delta which is 10.91. We can say this model is not good.

Lets perform random forest with effective variable in tree:

Call:
randomForest(formula = high.quality ~ alcohol + sulphates + total.sulfur.dioxide + fixed.acidity + volatile.acidity, data = wine, mtry = 6, importance = TRUE)

Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 5

OOB estimate of error rate: 9.93%

Confusion Matrix:

		True Values		
		High	**Low**	**Class. Error**
Predicted Values	**High**	262	33	0.039
	Low	18	11	0.476

Lets see the accuracy of our model in train dataset:

		True Values		
		High	**Low**	**Class. Error**
Predicted Values	**High**	794	33	0.039
	Low	62	68	0.476

```
[1] 9.926855
```

The above tables shows that 33 wrong prediction for low quality and 62 wrong prediction for high quality which in general 9.92% error we have in random forest which is quite good. Lets test our model on validation dataset and see the accuracy:

		True Values	
		High	Low
Predicted Values	High	268	24
	Low	12	20

```
## [1] 11.11111
```

Confusion matrix above shows us model predicted 268+20 correct predictions which is not bad in compare to 24+12 wrong predictions. delta is 11.11 which is small and its quite close to the train delta which is 11.11 We can say this model is a good model. In addition, random forest gives us a delta which is much better in compare to discriminate analysis.

Neural Network

Neural Network modeling is essentially a Machine Learning model that is used in unsupervised learning. A Neural Network is a web of interconnected entities known as nodes wherein each node is responsible for a simple computation.

So lets use this type of modeling in our data:

```
##                     Length Class      Mode
## call                   6   -none-     call
## response             957   -none-     numeric
## covariate           4785   -none-     numeric
## model.list             2   -none-     list
## err.fct                1   -none-     function
## act.fct                1   -none-     function
## output.act.fct         1   -none-     function
## linear.output          1   -none-     logical
## data                  12   data.frame list
## exclude                0   -none-     NULL
## net.result             1   -none-     list
## weights                1   -none-     list
## generalized.weights    1   -none-     list
## startweights           1   -none-     list
## result.matrix         39   -none-     numeric
```

Lets see the accuracy of our model in train dataset:

		True Values		Total
		High	Low	
Predicted Values	High	797	30	827
	Low	72	58	130
Total		869	88	957

```
## [1] 10.65831
```

The above tables shows that 30 wrong prediction for low quality and 72 wrong prediction for high quality which in general 10.65% error we have in neural network which is quite good.

Lets test our model on validation dataset and see the accuracy:

		True Values		Total
		High	Low	
Predicted Values	High	268	12	280
	Low	21	23	44
Total		289	35	324

```
delta = 10.18519
```

Confusion matrix above shows us model predicted 268+23 correct predictions which is not bad in compare to 12+21 wrong prediction. Delta is 10.18 which is small and its close to the train delta which is 10.65. We can say this model is a good model. In addition, this accuracy is smallest among other models.

3. Compare the results and choose the best model

We run 3 algorithms in our dataset to modeling our data based on three approaches the first approach was modeling data using discriminant analysis. We run LDA and QDA and we calculated the accuracy of each model LDA returned delta of 11.28 and QQA returned 12.43. Which we can say between these two LDA modeled our data better.

In random forest our delta is 11.19 which is better than discriminate analysis.

Neural network gives us the delta of 10.18.

So the best model to run our data between these three method is Neural Network.

4. Apply the model to the test data and predict the target values

Lets apply our neural network model to test dataset.

pr.nn <- **predict**(nn,wine_test, rep=imin_rep)
head(pr.nn)

```
##          [,1]
## [1,] 0.096642101
## [2,] 0.058471481
## [3,] 0.099833498
## [4,] 0.059504437
## [5,] 0.049448602
## [6,] 0.007017716
```

yhat_test = **round**(pr.nn)
head(yhat_test)

```
##     [,1]
## [1,]    0
## [2,]    0
## [3,]    0
## [4,]    0
## [5,]    0
## [6,]    0
```

The above values shows us the predicted value for test dataset.

5. Provide the file of the test data with the predicted values

Now we add these predicted value to the test dataset and save it in a file for validation.

```
wine_test$high.quality <- yhat_test
write.csv(wine_test,"winequality-red_predicted.csv", row.names = FALSE)

Note that the predicted file is available with this project under the name of
“winequality-red_predicted.csv”
```

6. Conclusion

By analyzing the data of red wines, it is able to create a model that can help *winemakers, oenophiles* and *sommeliers* from around the world to understand the best physical factors that lead wines to be classified on a scale from good to bad, in relation to their quality.

With that, we can come with 3 important conclusions:

1. The best model to fit this data is neural network, following by random forest and the logistic regression.
2. The features *citric acid, free sulfur dioxide* and *Ph* are not related to the quality classification of wine.
3. Based on this analysis, we can come to a ranking of importance of features related to wine qualities. The 5 most important features are, in order: **alcohol**, **sulphates**, **total.sulfur.dioxide**, **fixed.acidity** , **volatile.acidity**

7. Appendix

Importing the dataset

```
 wine <- winequality.red_train
wine <- wine[2:13]
N<-dim(wine)[1]
wine_validation <- winequality.red_validation
wine_validation <- wine_validation[2:13]
N_V<-dim(wine_validation)[1]
```

Creating Summary Table

```
library("devtools")
library("papeR")
summarize(wine, type = "numeric")
```

Drawing histograms

```
library(gridExtra)
library(ggplot2)
draw_hist <- function(variable, name)
{
  plot <- ggplot(data = wine, aes(x = variable)) +
        geom_histogram(aes(y=..density..), color = 'black', fill = '#099DD9',bins=10) +
        xlab(deparse(substitute(name))) + geom_density(alpha=.3, colour = 'red')
  return(plot)
}
grid.arrange(draw_hist(wine$fixed.acidity, FixedAcidity),
           draw_hist(wine$volatile.acidity, VolatileAcidity),
           draw_hist(wine$citric.acid, CitricAcid),
           draw_hist(wine$residual.sugar, ResidualSugar),
           draw_hist(wine$chlorides, Chlorides),
           draw_hist(wine$free.sulfur.dioxide, FreeSulfurDioxide),
           draw_hist(wine$total.sulfur.dioxide, TotalSulfurDioxide),
           draw_hist(wine$density, Density),
           draw_hist(wine$pH, pH),
           draw_hist(wine$sulphates, Sulphates),
```

```
        draw_hist(wine$alcohol, Alcohol),
        ncol = 3)
```

Transforming values into factors

```
wine$high.quality <- factor (as.integer(wine$quality>6))
wine_validation$high.quality <- factor (as.integer(wine_validation$quality>6))
```

Ploting Boxplots

```
draw_box <- function(variable, name)
{
 plot <- ggplot(data = wine, aes(x = high.quality, y = variable, color = high.quality)) +
 geom_boxplot() +
       xlab(deparse(substitute(name)))
 return(plot)
}
grid.arrange(draw_box(wine$fixed.acidity, FixedAcidity),
        draw_box(wine$volatile.acidity, VolatileAcidity),
        draw_box(wine$citric.acid,CitricAcid),
        draw_box(wine$residual.sugar,ResidualSugar),
        ncol = 2)
```

Correlation Matrices Graphs

```
library(psych)
pairs.panels(wine[c(1:4, 12)], main="Relationships between characteristics of wine factors",
pch=20,
col="blue",
)

pairs.panels(wine[c(4:7, 12)], main="Relationships between characteristics of wine factors",
pch=20,
col="blue",
)

pairs.panels(wine[8:12], main="Relationships between characteristics of wine factors",
pch=20,
col="blue",
)
```

```
library(corrplot)
correlation1<-cor(wine[1:11])
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(correlation1,

    method="shade", # visualisation method

      shade.col=NA, # colour of shade line

      tl.col="black", # colour of text label

      tl.srt=45, # text label rotation

      col=col(200), # colour of glyphs

      addCoef.col="black", # colour of coefficients

      order="AOE", # ordering method

      number.cex=0.7
```

Glm Fit

```
> glm.fit <- step(glm(high.quality~.,family = binomial,data = wine), direction="forward")

Start:  AIC=526.89
high.quality ~ fixed.acidity + volatile.acidity + citric.acid +
   residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
   density + pH + sulphates + alcohol
```

The Summery of The Tree

```
> setup<-tree.control(nrow(wine), mincut = 5, minsize = 10, mindev = 0.015)
> tree.fit = tree(high.quality ~.-citric.acid-free.sulfur.dioxide-pH, data=wine, control = setup)
> summary(tree.fit)

Classification tree:
tree(formula = high.quality ~ . - citric.acid - free.sulfur.dioxide -
   pH, data = wine, control = setup)
Variables actually used in tree construction:
[1] "alcohol"          "sulphates"          "total.sulfur.dioxide"
[4] "fixed.acidity"    "volatile.acidity"
Number of terminal nodes:  11
```

Residual mean deviance: 0.4925 = 465.9 / 946
Misclassification error rate: 0.1129 = 108 / 957

> tree.fit

node), split, n, deviance, yval, (yprob)
** denotes terminal node*

1) root 957 760.500 0 (0.864159 0.135841)
2) alcohol < 10.75 627 240.900 0 (0.952153 0.047847)
4) sulphates < 0.615 350 34.530 0 (0.991429 0.008571)
8) total.sulfur.dioxide < 18.5 44 21.900 0 (0.931818 0.068182)
*16) total.sulfur.dioxide < 17.5 38 0.000 0 (1.000000 0.000000) **
*17) total.sulfur.dioxide > 17.5 6 8.318 0 (0.500000 0.500000) **
*9) total.sulfur.dioxide > 18.5 306 0.000 0 (1.000000 0.000000) **
5) sulphates > 0.615 277 177.000 0 (0.902527 0.097473)
10) alcohol < 9.85 148 36.780 0 (0.972973 0.027027)
*20) fixed.acidity < 12.05 140 11.880 0 (0.992857 0.007143) **
*21) fixed.acidity > 12.05 8 10.590 0 (0.625000 0.375000) **
*11) alcohol > 9.85 129 120.900 0 (0.821705 0.178295) **
3) alcohol > 10.75 330 404.900 0 (0.696970 0.303030)
6) volatile.acidity < 0.425 159 219.900 0 (0.528302 0.471698)
*12) sulphates < 0.715 85 106.300 0 (0.682353 0.317647) **
13) sulphates > 0.715 74 95.950 1 (0.351351 0.648649)
*26) alcohol < 12.45 62 84.330 1 (0.419355 0.580645) **
*27) alcohol > 12.45 12 0.000 1 (0.000000 1.000000) **
7) volatile.acidity > 0.425 171 142.300 0 (0.853801 0.146199)
*14) alcohol < 11.95 131 70.670 0 (0.923664 0.076336) **
*15) alcohol > 11.95 40 52.930 0 (0.625000 0.375000) **

Above results shows us terminal nodes of the tree which is indicated by *. Total sulfur dioxide, fixed acidity, sulphates and alcohol are terminal nodes here.
As we can see node numbers are 15

The Summary of Random Forest

> rand.wine=randomForest(
+ high.quality~ alcohol + sulphates + total.sulfur.dioxide + fixed.acidity + volatile.acidity
,data=wine,mtry=5,importance=TRUE)
> rand.wine

Call:
randomForest(formula = high.quality ~ alcohol + sulphates + total.sulfur.dioxide +
fixed.acidity + volatile.acidity, data = wine, mtry = 5, importance = TRUE)
Type of random forest: classification
Number of trees: 500

No. of variables tried at each split: 5

OOB estimate of error rate: 10.14%
Confusion matrix:
0 1 class.error
0 793 34 0.04111245
1 63 67 0.48461538

> importance(rand.wine)

	0	*1*	*MeanDecreaseAccuracy*	*MeanDecreaseGini*
alcohol	*10.043724*	*60.61710*	*43.35550*	*56.97706*
sulphates	*4.086005*	*46.86809*	*29.18832*	*44.54419*
total.sulfur.dioxide	*8.870041*	*30.04780*	*23.65847*	*39.77195*
fixed.acidity	*3.171308*	*31.02641*	*21.38642*	*36.28692*
volatile.acidity	*6.644531*	*35.78262*	*24.13330*	*46.24119*