Course: ALY6020
Term: A_Fall_2020

Name: Pragati Koladiya
NUID: 001029445

Assignment 3

Introduction:

Taking decisions is crucial in human life. The impact of the decisions varies from scenario to scenario. In business world, decision making is the backbone for business management because each and every second matters. To make right decisions people are taking help of advanced machine learning model which helps them to get confident about their life long decisions.

The objectives of this assignment are as follows,

- To perform data visualization techniques to understand the insight of the data.
- This led us to perform the decision tree and random forest for the chosen dataset.
- Apply various R tools to get a visual understanding of the data
- Clean it to make it ready to apply machine learning model thus predicting the outcome.

Problem Statement:

Consider a wine company that has all the data containing the elements of red wine in a region. It wishes to use the data to improve the quality of red wine al well as predict the quality of wine based on important factors such as alcohol, pH, acidity, etc.

Essentially, the company wants,

- To identify the variables affecting quality of wine,
- To compare several classification algorithms to predict wine quality
- Show some red wine information details with Exploratory Data Analysis(EDA)
- Find best model to predict quality of wine
- The goal of predictive analysis is to avoid overfitting and find the model that gives the highest accuracy

Data Acquisition :

The dataset is chosen from Kaggle with file name “”. This dataset contains an airline passenger satisfaction survey.

The dataset contains 103904 observations(airline customer’s samples) and 25 attributes related to housing.

Data Information:

The data dictionary is provided by the Kaggle with detailed explanation for each variable.

- Link:https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/notebooks?sortBy=relevance&group=everyone&search=R&page=1&pageSize=20&datasetId=4458

The data is about red wine samples (vinho verde) from Portugal.

- The data contains 1599 observations (wine samples) and 12 attributes related to the wine.
- The data was collected from May 2004 to February 2007
- *Vinho Verde* is not a grape variety, it is a DOC for production of wine. The name means "green wine," but translates as "young wine", with wine being released three to six months after the grapes are harvested. They may be red, white or rose and they are usually consumed soon after bottling.

Course: ALY6020
Term: A_Fall_2020

Name: Pragati Koladiya
NUID: 001029445

Data Dictionary: Red Wine Quality

1. ***fixed acidity*** - Most acids involved with wine or fixed or nonvolatile(do not evaporate readily)
2. ***volatile acidity*** - The amount of acetic acid in wine, which at too high of levels can lead to an unplesant, vinegar taste
3. ***citric acid*** - Found in small quantities, citric acid can add 'freshness' and flavour to wines
4. ***residual sugar*** - The amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter. Wines with greater than 45 grams/litter are considered sweet
5. ***chlorides*** - The amount of salt in the wine
6. ***free sulfur dioxide*** - The free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine
7. ***total sulfur dioxide***- Amount of free and bound form of SO2; in low concentrations, SO2 is mostly undetectable in wine,but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine
8. ***density*** - The wine density
9. ***pH*** - Describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale
10. ***sulphates*** - A wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant
11. ***alcohol*** - The percent alcohol content of the wine
12. ***quality*** - output variable (based on sensory data, score between 0 and 10)

All the measurements are (in grams) per decimeter cubed of wine (dm^3) [g/dm^3] accept pH, alcohol and quality columns

Data Overview:

Required packages have been imported before loading the dataset into R.

- Head of the dataset with columns and first few rows. I have removed the space from the variables labels for simplicity.

```
> head(wine_data)
# A tibble: 6 x 13
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide total.sulfur.dioxide density    pH sulphates alcohol quality  good
          <dbl>            <dbl>       <dbl>          <dbl>     <dbl>              <dbl>               <dbl>   <dbl> <dbl>     <dbl>   <dbl> <fct>   <dbl>
1          7.4              0.7           0            1.9     0.076                 11                  34   0.998  3.51      0.56     9.4 5           0
2          7.8              0.88          0            2.6     0.098                 25                  67   0.997  3.2       0.68     9.8 5           0
3          7.8              0.76          0.04         2.3     0.092                 15                  54   0.997  3.26      0.65     9.8 5           0
4         11.2              0.28          0.56         1.9     0.075                 17                  60   0.998  3.16      0.580    9.8 6           0
5          7.4              0.7           0            1.9     0.076                 11                  34   0.998  3.51      0.56     9.4 5           0
6          7.4              0.66          0            1.8     0.075                 13                  40   0.998  3.51      0.56     9.4 5           0
```

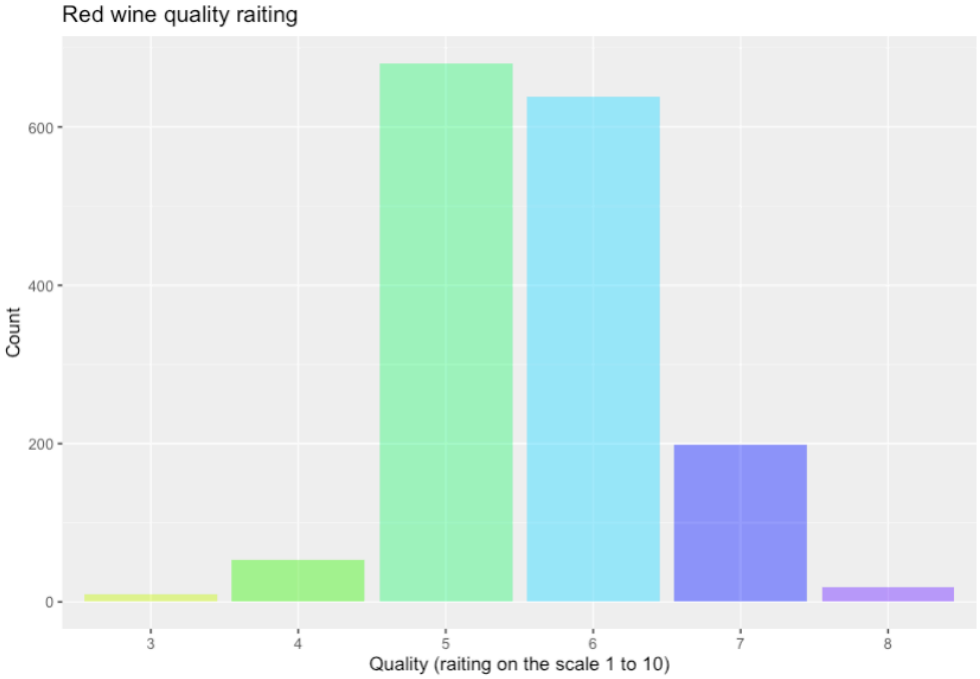
Course: ALY6020
Term: A_Fall_2020

Name: Pragati Koladiya
NUID: 001029445

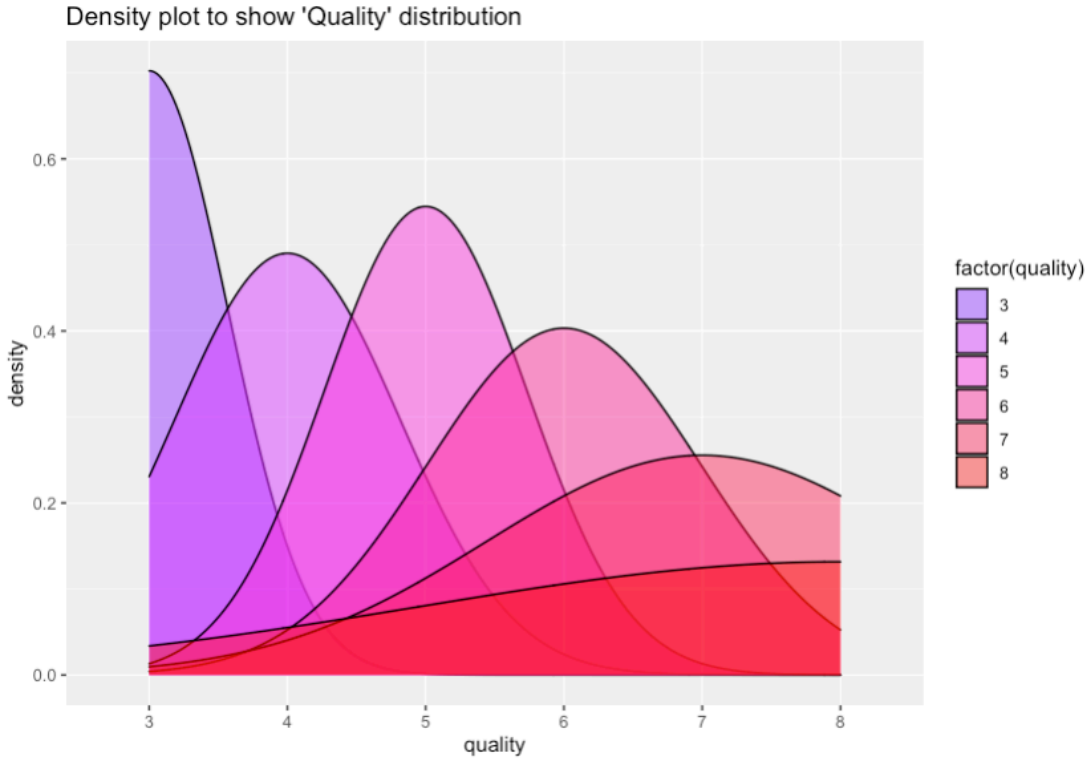
Exploratory Data Analysis(EDA):

The most important step is to understand the data and identify if there is some obvious multicollinearity present. Here's where we will also identify if predictors have a strong association with the outcome variable.

1) Distribution of red wine quality ratings using bar plot and density plot



- Maximum quality counts are dense around 5 and 6 while quality count 3,4,7 and 8 are relatively low.



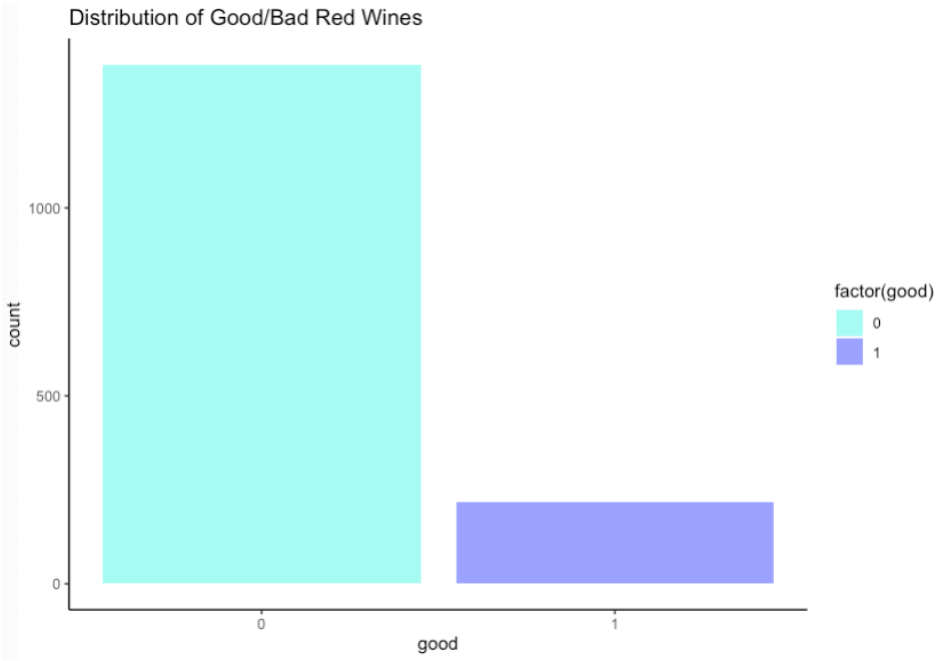
- The data is not equally distributed, so we will consider some actions in future so that results may not bias to majority.

Name: Pragati Koladiya
NUID: 001029445

2) Distribution of good / bad red wine quality

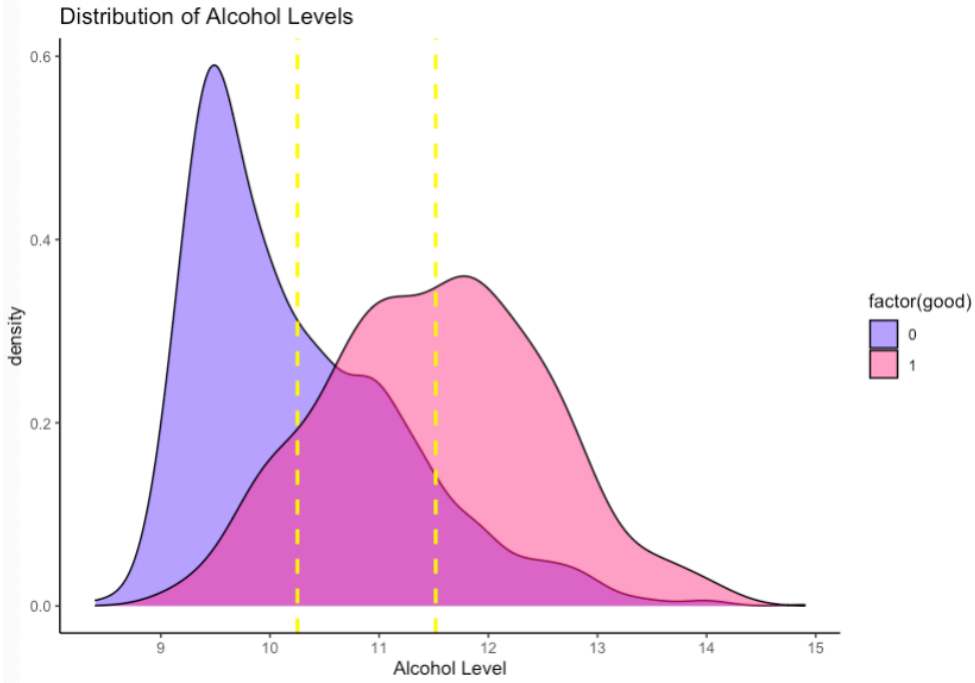
```
#-- Create a variable indicating if a wine is good or bad
wine_data$good <- ifelse(wine_data$quality > 6,1,0)
```

- The red wine quality rating is between 3 and 8. In order to avoid overfitting the dependent variable needs to be equally distributed so the rating more than 6 is considered to be a good wine else bad wine quality.



- Above bar graph shows that the good wines were outnumbered by bad wines by a huge margin. Most wines are average(rated 5 or 6), but we could also observe that there are some poor wines (rated 3 or 4). A vast majority of good wines has a quality rating of 7.

3) Distribution of alcohol levels in wine using density plot



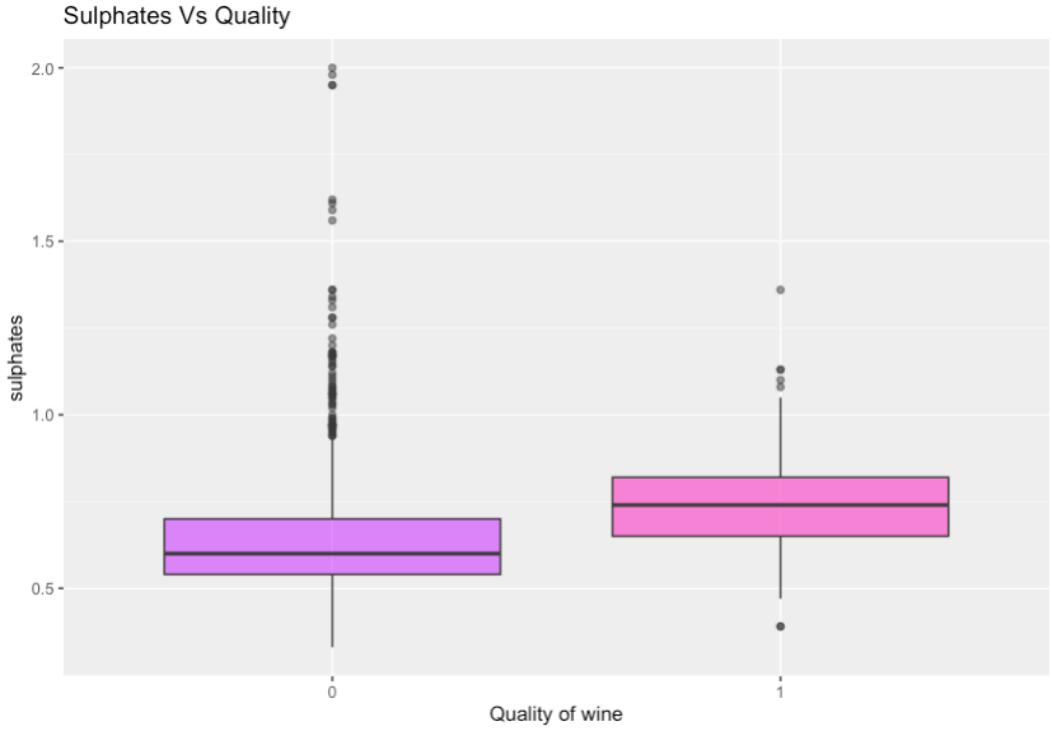
- Above graph shows that good and bad wines has very similar distribution of their corresponding physiochemical properties.

Name: Pragati Koladiya
NUID: 001029445

4) Alcohol Vs Density grouped by quality of wine using scatter plot

- The alcohol is denser between 9 and 10 while the density is denser between 1 and 0.996.
- The density column has equally distributed data while alcohol has right-skewed data.
- The green points indicates that the bad wine quality has lower amount of alcohol in it where as good wine quality has alcohol level above 10.

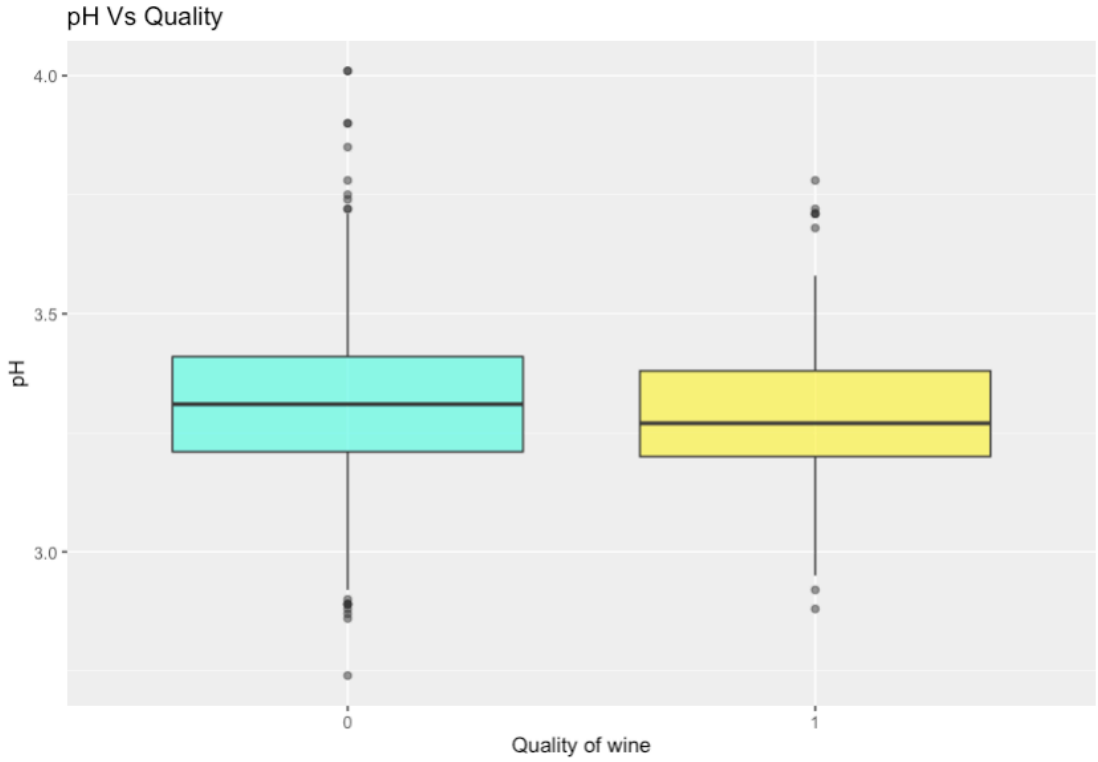
5) Sulphates Vs Quality reviews using box plot



- Wine to being a prestigious quality requires an average sulphates level to be around 0.75 grams. The bad wine contains sulphate level approximately below 0.55 grams.
- The box plot also helps to identify the outliers.

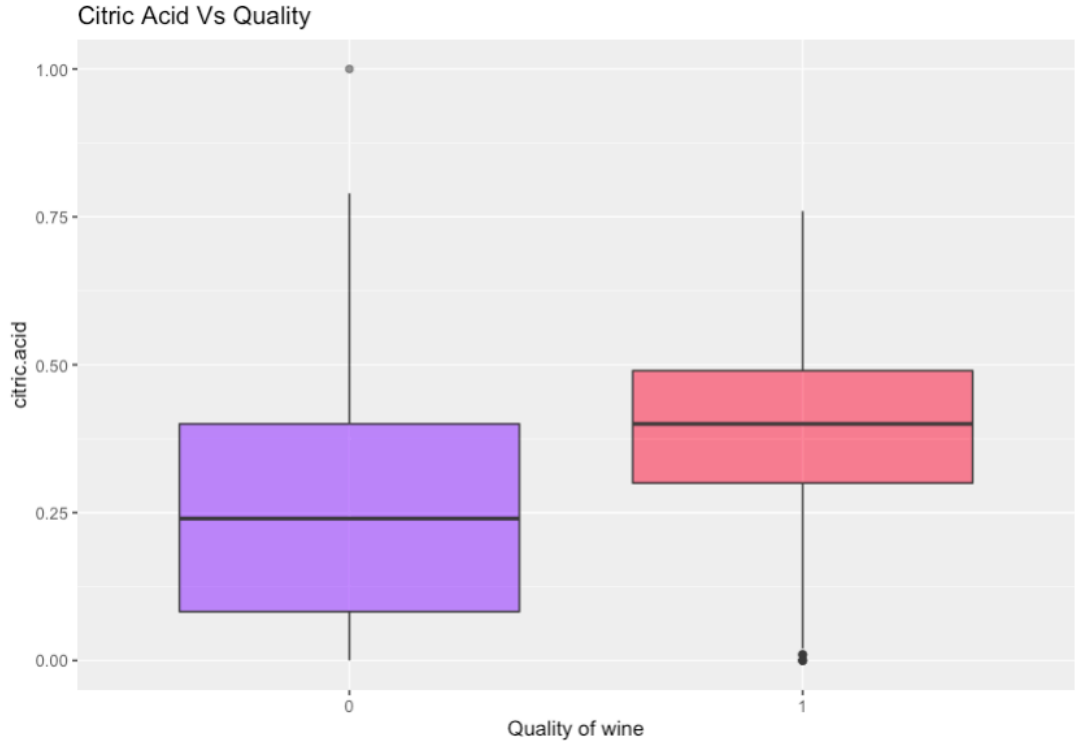
Name: Pragati Koladiya
NUID: 001029445

6) pH Vs Reviews of red wine using box plot



- The pH shows how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic)
- Most wines are between 3-4 on the pH scale
- The excellent wine contains pH, which is approximately an average of 3.2 grams per decimetre whereas bad wine contains pH approximately 3.4 grams per decimetre.

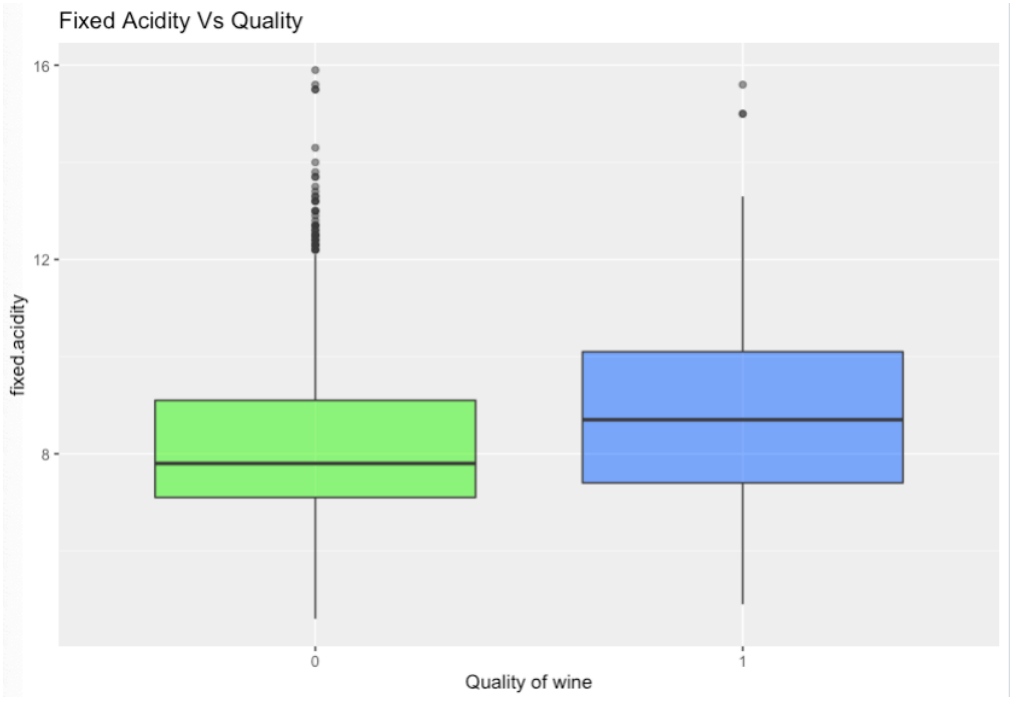
7) Citric acid Vs Quality of red wine using box plot



- It found in small quantities, citric acid can add 'freshness' and flavour to wines
- In order to be qualifies as better wine, it should have higher amount of citric acid in it.

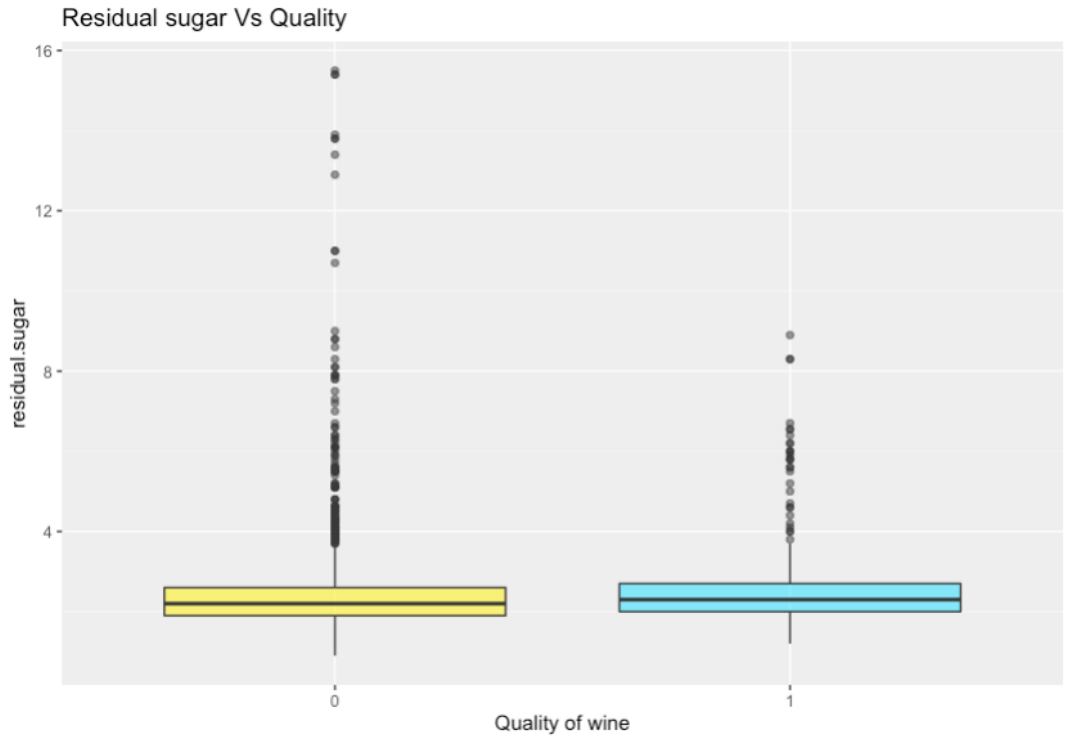
Name: Pragati Koladiya
NUID: 001029445

8) Fixed Acidity Vs Quality of wine



- The bad wine has average fixed acidity below 8 grams per decimetre where as good wine quality has average 9 grams per decimetre fixed acidic content.

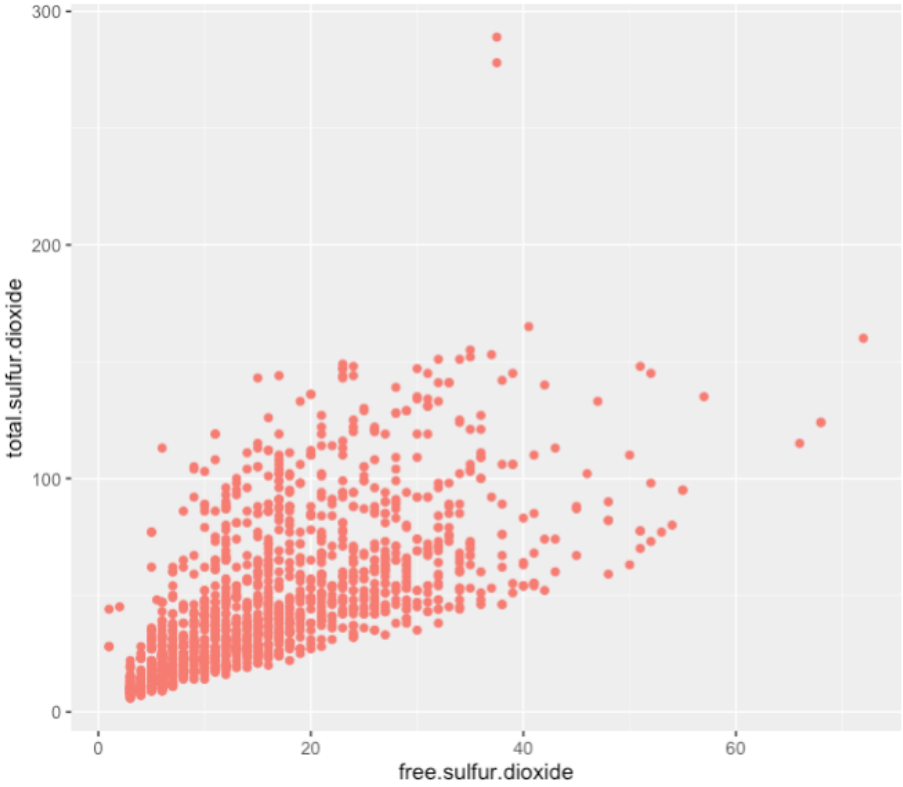
9) Residual sugar Vs Quality reviews



- It is difficult to find a wine with sugar level less than 1 gram/litter .
- Wines with greater than 45grams/litter considered to be a sweet wine.
- It is apparent that wine quality does not affect by the sugar level as far as the sugar level is close around 2.5 grams per decimetre

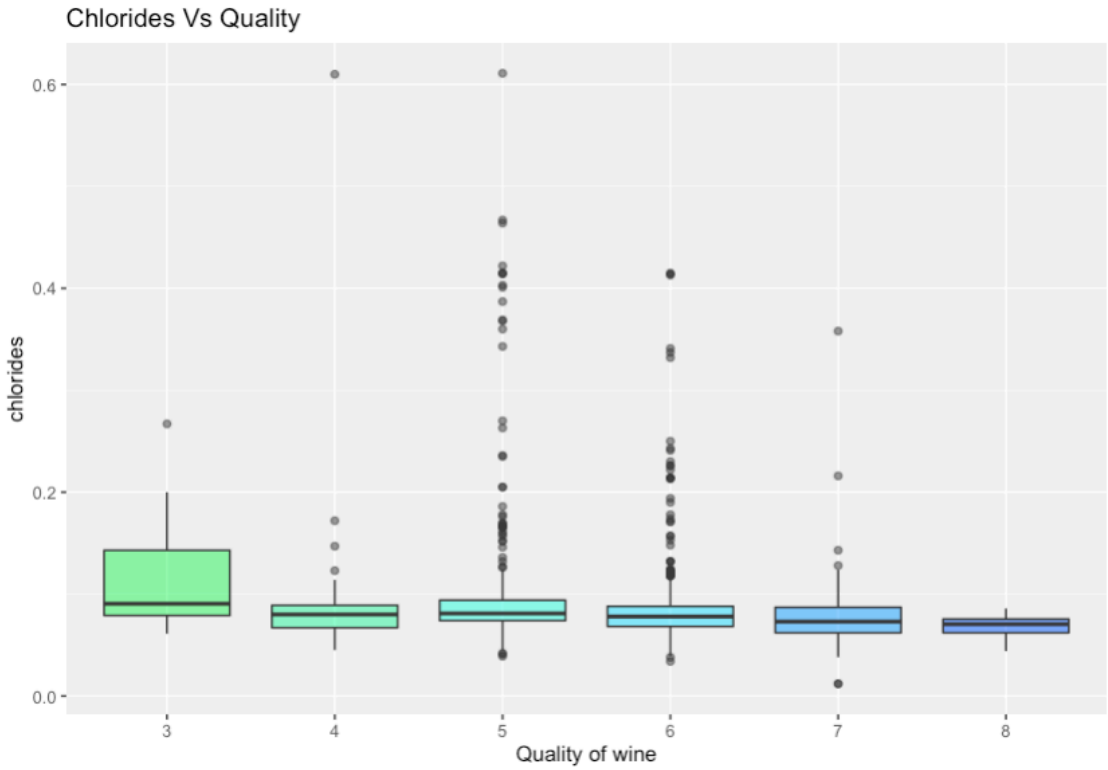
Name: Pragati Koladiya
NUID: 001029445

10) Total sulfur dioxide Vs Free Sulfur dioxide



- The above scatter plot shows the total sulfur and free sulfur dioxide has the positive correlation.

11) Chlorides Vs Quality



- Chlorides is the amount of salt in the wine.
- Chlorides and quality appear a negative relationship for red wines. which can be interpreted as salt is not much required for a good quality.

Data Preparation:

The first step is to check for null values in the dataset.

```
> sum(is.na(wine_data))
[1] 0
```

- We see that the sum of null values is 0 which means there are no null values in our dataset.

Decision Tree Implemented:

Decision tree is a tree shaped algorithm used to determine a course of action. Each branch of the tree represents a possible decision, occurrence or reaction.

In this step, we will implement the decision tree model that will be appropriate for the given dataset. As discussed earlier the aim is to predict the quality of red wine. Considering that independent variables(X) would be all the features except the quality. The dependent variable(Y) will be quality.

I have implemented three different trees with different variables. They are as listed below,

- Decision tree without splitting data
  - Good Vs alcohol, sulphates (Selected this two features because the most discriminating attributes we can observe are sulphates and alcohol level of the red wine.
  - Good Vs alcohol, volatile acidity, citric acidity, and sulphates
- Decision tree with splitting data
  - Quality Vs all independent variable

Decision tree1 without splitting data:

For the tree1, we are going to use good as dependent variable as good column has 0 and 1 as a unique values, 0 represents bad wine quality and 1 represents good wine quality. The 'good' feature is derived from the 'quality' where red wine quality ratings more than 6 are considered as good else bad.

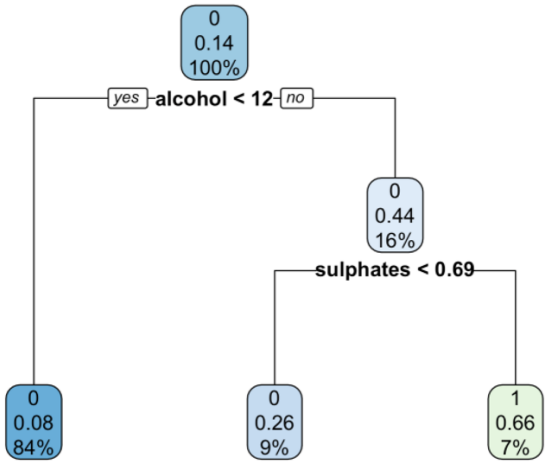
```
tree1 <- rpart(good ~ alcohol + sulphates, data = wine_data, method="class")
summary(tree1)
rpart.plot(tree1)
pred1 <- predict(tree1,newdata=wine_data,type="class")
```

Decision tree has various parameters that control aspects of the fit. In the rpart library, can control the parameters using the rpart.control() function.

The model is built using two independent variables as the most discriminating attributes we can observe are Sulphates and Alcohol level of the wine.

Name: Pragati Koladiya
NUID: 001029445

Using rpart.plot() function, the decision tree is plotted as below,



- From above graph shows the decision tree, shows the percentage of wine quality to be based on given independent features. It infers that 84% chances that red wine quality will be bad whereas only 7% chances of good wine quality

```
> table(wine_data$good,pred1)
   pred1
       0    1
  0 1343   39
  1  142   75
```

- The table shows confusion matrix of tree1. 0 indicates bad wine quality and 1 indicates good wine quality.

```
> Accuracy_t1 = ((1343+75)/(1343+39+142+75))
> Accuracy_t1
[1] 0.8868043
```

- The model's accuracy is 89% which is quite good.

Decision tree2 without splitting data:

For the tree2, we are going to use good as dependent variable and alcohol, volatile acidity, citric acide and sulphate as independent variable. This features were selected after analysing the data during EDA.

```
tree2 <- rpart(good ~ alcohol + volatile.acidity + citric.acid + sulphates, data = wine_data, method="class")
summary(tree2)
rpart.plot(tree2)
pred2 <- predict(tree2,newdata=wine_data,type="class")
```

As explained earlier the model is built using rpart() function and it has dependent variable good and four independent variables namely, alcohol, volatile acidity, citric acid, and sulphates.

Models summary is detailed analysis of tree2 with each variables importance used during model building as follows,

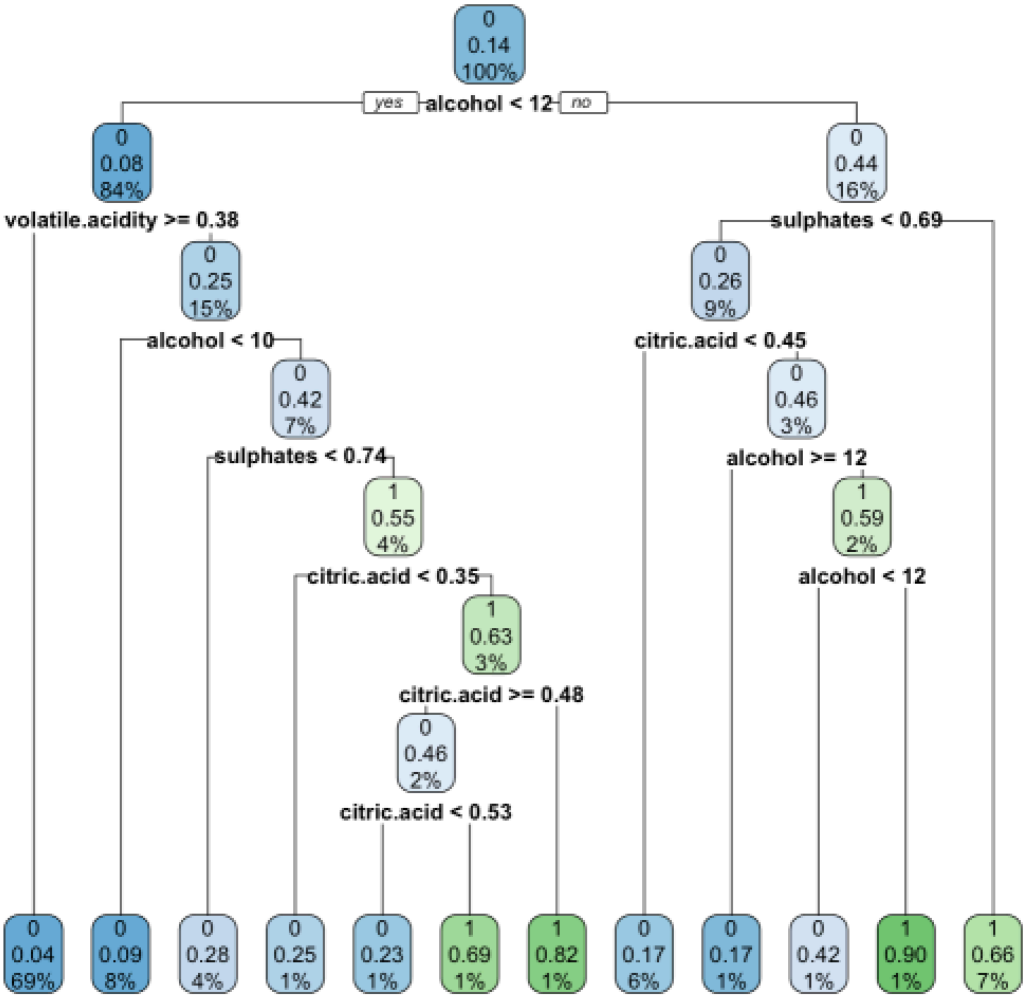
Name: Pragati Koladiya
NUID: 001029445

```
> summary(tree2)
Call:
rpart(formula = good ~ alcohol + volatile.acidity + citric.acid +
    sulphates, data = wine_data, method = "class")
  n= 1599

          CP nsplit rel error    xerror       xstd
1 0.08294931      0 1.0000000 1.0000000 0.06311025
2 0.01382488      2 0.8341014 0.8663594 0.05935516
3 0.01152074      8 0.7465438 0.9032258 0.06043283
4 0.01000000     11 0.7096774 0.9032258 0.06043283

Variable importance
        alcohol        sulphates volatile.acidity
             51              20              16
    citric.acid
             14
```

- As we can observe the highest variable importance is alcohol than sulphates, volatile acidity, and citric acid.



- Above decision tree is showing detailed decision made at each step with given conditions. Our model indicates that the higher chances that the wine quality will be bad highlighted with blue at leaf node and green represents chances of good wine quality are almost near the zero.

```
> table(wine_data$good,pred2)
   pred2
       0    1
  0 1334   48
  1  106  111
```

```
> Accuracy_t2=((1334+111)/(1334+48+106+111))
> Accuracy_t2
[1] 0.9036898
```

- The above matrix shows the important statistics. From the matric we can find the accuracy of the model using formula TP + TN/(TP+TN+FP+FN). It gives 90% accuracy we can definitely trust the model as it predicts that based on historical data the wine quality is going to be bad.

Decision tree after splitting data:

The first step in the model implementation is to split the dataset to eliminate the bias to training data in machine learning algorithms.

```
set.seed(123)
sample = sample.split(wine_data$quality, SplitRatio = .80)
train = subset(wine_data, sample == TRUE)
test = subset(wine_data, sample == FALSE)
```

```
> dim(train)
[1] 1278   13
> dim(test)
[1] 321  13
```

Train set has 1278 x 13 and the test set has 321 x 13. The data is split into a 80:20 ratio.

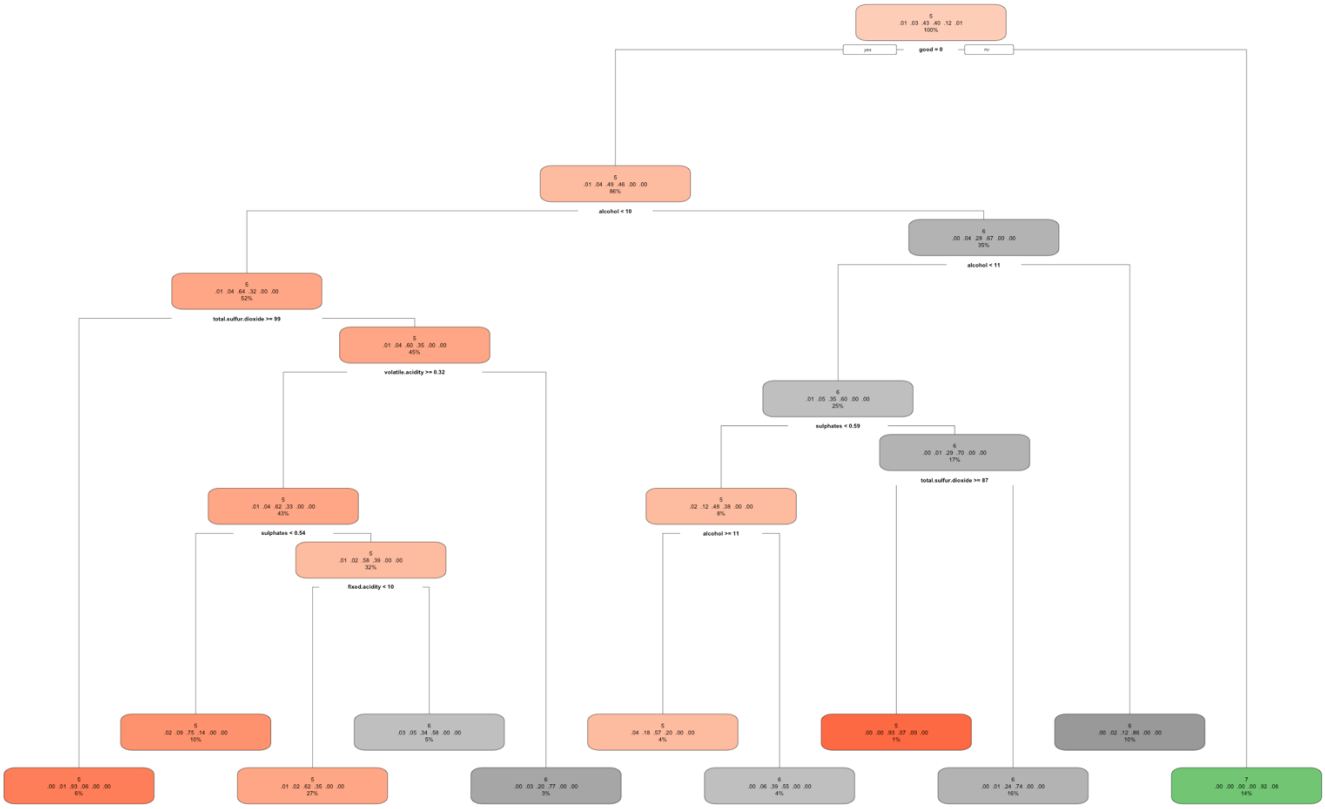
The caTools package provides a method sample.split() for partitioning our data into train and test sets. We are passing 2 parameters. The “y” parameter takes the value of the variable according to which data needs to be partitioned. In our case, the target variable is quality, so we are passing wine_data $quality and the SplitRatio is .80 meaning it splits into 80:20.

```
#Training the decision tree classifier
tree <- rpart(quality ~ ., data = train, method = 'class')
```

In this model we are going to use entire dataset so that can will be able to compare all the models. The model has used rpart() function with dependent variable as quality – column contains rating of red wine on the scale of 1 to 10. And independent features are all(remaining 11 features) other than quality.

Let’s plot the tree so that we can visualize each steps with conditions.

Name: Pragati Koladiya
NUID: 001029445



- Above image is quite big and difficult to read but here I would like to highlight the important decisions. The model indicates only 7% chances that customers are going to give reviews as good win quality whereas more than 90% shows the wine quality will be bad.

```
#predictions
tree.quality.predicted <- predict(tree, newdata = test, type = 'class')
```

Here we will predict the quality of wine using test data. The predict() function will helps to do so. The below table shows the actual values as column name `test.quality` and predicted values in column name `tree.quality.predicted`.

	tree.quality.predicted	test.quality
1	5	5
2	5	5
3	5	5
4	5	5
5	5	6
6	6	6
7	5	5

As we see out of 7 observations 6 observations shows the wine quality will be bad. Wine reviews above 6 is good and below that are bad so only one of them shows good wine quality. There are many other observations listed in the table but here we have shown first few of them.

After predicting the wine quality lets view the confusion matrix with accuracy so that we can decide whether we will trust the model or not.

```
> #confusion matrix for evaluating the model
> confusionMatrix(table(tree.quality.predicted ,test$quality))
Confusion Matrix and Statistics


tree.quality.predicted   3   4   5   6   7   8
                     3   0   0   0   0   0   0
                     4   2  11   0   0   0   0
                     5   0   0 103  49   0   0
                     6   0   0  33  79   0   0
                     7   0   0   0   0  40   4
                     8   0   0   0   0   0   0

Overall Statistics

               Accuracy : 0.7259
                 95% CI : (0.6736, 0.7739)
    No Information Rate : 0.4237
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5728

 Mcnemar's Test P-Value : NA


 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 3 Class: 4 Class: 5 Class: 6 Class: 7
Sensitivity          0.000000  1.00000   0.7574   0.6172   1.0000
Specificity          1.000000  0.99355   0.7351   0.8290   0.9858
Pos Pred Value            NaN  0.84615   0.6776   0.7054   0.9091
Neg Pred Value       0.993769  1.00000   0.8047   0.7656   1.0000
Prevalence           0.006231  0.03427   0.4237   0.3988   0.1246
Detection Rate       0.000000  0.03427   0.3209   0.2461   0.1246
Detection Prevalence 0.000000  0.04050   0.4735   0.3489   0.1371
Balanced Accuracy    0.500000  0.99677   0.7462   0.7231   0.9929
                     Class: 8
Sensitivity           0.00000
Specificity           1.00000
Pos Pred Value            NaN
Neg Pred Value        0.98754
Prevalence            0.01246
Detection Rate        0.00000
Detection Prevalence  0.00000
Balanced Accuracy     0.50000
```

- The confusion matrix provides the matrix with accuracy 73%, with 95% confidence interval between 0.6736 and 0.7739. It also indicates the statistics by class. It shows eight different classes which has been used during decision making.

For more clear understanding I have plotted a confusion matrix table which shows the detail row and column count with total observation.

```
> CrossTable(x = tree.quality.predicted, y = test$quality,
+            prop.chisq = FALSE)
```

Name: Pragati Koladiya
NUID: 001029445

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  321
```

tree.quality.predicted	test$quality 3	4	5	6	7	8	Row Total
5	0	5	93	36	2	0	136
	0.000	0.037	0.684	0.265	0.015	0.000	0.424
	0.000	0.455	0.684	0.281	0.050	0.000	
	0.000	0.016	0.290	0.112	0.006	0.000	
6	2	5	42	85	29	1	164
	0.012	0.030	0.256	0.518	0.177	0.006	0.511
	1.000	0.455	0.309	0.664	0.725	0.250	
	0.006	0.016	0.131	0.265	0.090	0.003	
7	0	1	1	7	9	3	21
	0.000	0.048	0.048	0.333	0.429	0.143	0.065
	0.000	0.091	0.007	0.055	0.225	0.750	
	0.000	0.003	0.003	0.022	0.028	0.009	
Column Total	2	11	136	128	40	4	321
	0.006	0.034	0.424	0.399	0.125	0.012	

The table has total observations 321. Column(horizontal) shows the wine quality(`test$quality`) on the scale of 1 to 10. People has given maximum ratings between 3 and 8. The `test.quality.predicted` (vertical) shows the predicted values for given test data. If we observe the column total it shows how many total wine quality were predicted as given rating meaning that wine quality 3, model has predicted twice. Model has predicted most of the time the wine rating will be 5 as column count is 136. The wine quality being 6 is 128 times being predicted which is less then rating 5. As we know above 6 shows wine will be good and below that will be bad. Over all we can say the model has predicted wine quality will be bad with 77% accuracy.

Random Forest Implemented:

Random Forest is a method that operates by constructing multiple decision tree during the training phase. The decision of the majority of the tree is chosen by the random forest as the final decision.

The first step in the model implementation is to split the dataset to eliminate the bias to training data in machine learning algorithms.

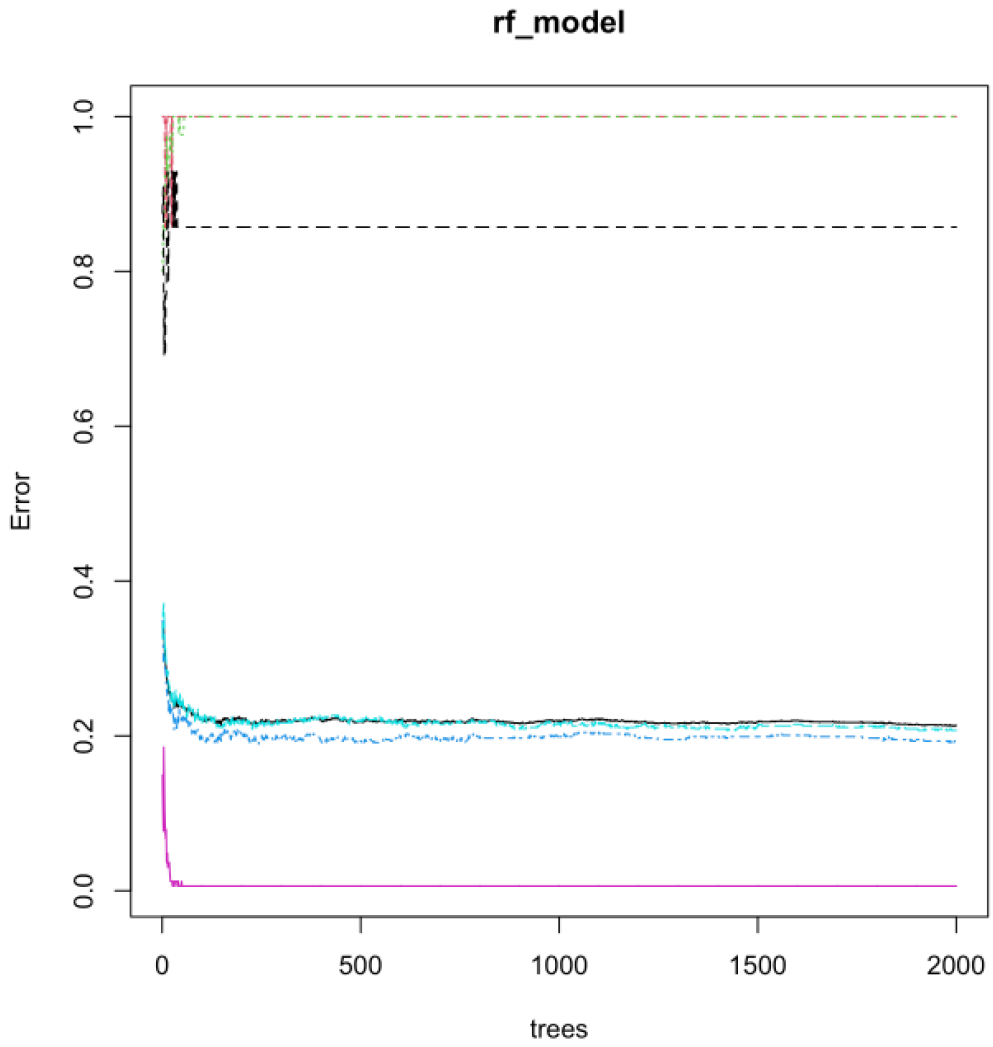
```
set.seed(123)
index <- sample(1:nrow(wine_data),size = 0.8*nrow(wine_data))
train <- wine_data[index,]
test <- wine_data[-index,]
```

- The data is being split into 80:20 ration

```
# simplest random forest
rf_model <- randomForest(quality ~., data = train, mtry=4, ntree=2000, importance=TRUE)
plot(rf_model)
```

- The random forest is being implemented using randomForest() function from library randomForest.
- The first argument shows which column are we going to predict here it's quality with independent variables other than quality.
- We are using training data and mtry depends upon how many feature are we using to predict the dependent variable.

- Here we have chosen 4 as mtry. The ntree is number of trees with default value 500 or 501. It says that ntree should be odd in case there is a tie but we have used even value 2000.
- Last argument is importance set to True which is used to make sure there is some corelation between features.



The above graph shows the error rate Vs trees. The x axis shows the number of tree as we discussed earlier we have opted ntree = 2000 and y axis shows the error rate at each variable.

It is clear that as we are building more and more trees the error rate is reducing to almost none. From the above graph we can also decide how many ntree should be plotted to get accurate results. Even if we have selected ntree = 500 we might get the accurate reslt.

After model implementation we are predicting the quality of wine which is stored in the rf_predict variable.

```
result <- data.frame(test$quality, predict(rf_model, test[,1:11], type="response"))
View(data.frame(rf_prediction, test$quality))
```

The below table shows the predicted rating for red wine using test quality data.

Name: Pragati Koladiya
NUID: 001029445

	rf_prediction	test.quality
1	5	5
2	5	5
3	5	5
4	5	5
5	5	5
6	5	6
7	5	5

- The above table shows first few line of data as it has 321 results predicted for red wine quality. This results are almost identical to what we have predicted using decision tree using all the feature.

```
rf_matrix <- confusionMatrix(rf_prediction, test$quality)
rf_matrix
```

```
> rf_matrix
Confusion Matrix and Statistics

          Reference
Prediction   3   4   5   6   7   8
         3   0   0   0   0   0   0
         4   3  10   0   0   0   0
         5   0   0 123  35   0   0
         6   0   0  20  91   0   0
         7   0   0   0   0  34   4
         8   0   0   0   0   0   0

Overall Statistics

               Accuracy : 0.8062
                 95% CI : (0.7586, 0.8481)
    No Information Rate : 0.4469
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6919

 Mcnemar's Test P-Value : NA


Statistics by Class:

                     Class: 3 Class: 4 Class: 5 Class: 6 Class: 7
Sensitivity          0.000000  1.00000   0.8601   0.7222   1.0000
Specificity          1.000000  0.99032   0.8023   0.8969   0.9860
Pos Pred Value            NaN  0.76923   0.7785   0.8198   0.8947
Neg Pred Value       0.990625  1.00000   0.8765   0.8325   1.0000
Prevalence           0.009375  0.03125   0.4469   0.3937   0.1062
Detection Rate       0.000000  0.03125   0.3844   0.2844   0.1062
Detection Prevalence 0.000000  0.04063   0.4938   0.3469   0.1187
Balanced Accuracy    0.500000  0.99516   0.8312   0.8096   0.9930
                     Class: 8
Sensitivity            0.0000
Specificity            1.0000
Pos Pred Value            NaN
Neg Pred Value         0.9875
Prevalence             0.0125
Detection Rate         0.0000
Detection Prevalence   0.0000
Balanced Accuracy      0.5000
```

- The random forest gives accuracy 80% with 95% confidence interval between 0.7586 and 0.8481.

Name: Pragati Koladiya
NUID: 001029445

Conclusion:

Even though the quality of a wine is a subjective matter, prediction can still help tremendously to make a better choice. We were able to predict the wine quality using the best model (i.e. random forest), with accuracy score 80% using all the features.

After implementing different decision trees, it is clear that the prediction is mostly depend on the features which has higher correlation meaning that red wine quality is highly depends on the alcohol level, it's acidic factors and sugar content. In real world people do judge wine based on its smell and test. The decision tree(tree2) has also predicted wine quality with accuracy 90 percent using the most important four features.

Discussion:

What makes the problem interesting from the viewpoint of analytics?

- The decision tree creates multiple trees by conditional node. Based on that the tree splits into branches or edges. When the entropy is null and stops further splitting then the final output of the decision tree occurs. this final output becomes the input for the random forest. The random forest predicts the output based on the majority class.
- The model will solve the problem by not overfitting the data (Instead use multiple trees to reduce the risk of over fitting. This leads to reduced training time. It also gives higher accuracy by executing effectively on large datasets.

How did the chosen technique help to illuminate, or solve the problem?

- Data analysis and its technique is solving big-time medical problems. Taking all the heart fatalities into the account, analysing the patterns of the pulsing rate and the heart rhythm will be a helping hand. Random forest's technique of bagging and using random building thrives to give accurate results. It then creates an individual tree from it and then an uncorrelated forest makes an accurate prediction.

What analysis do you think should be conducted next?

- It can be analysed that how many people are interested in particular brand. What is their basic requirement. After analysing all the features of red wine the company must be able to understand the pattern. It will be able to improve their wine quality, for being one of the best red wine company.

Reference:

1] https://www.datacamp.com/community/tutorials/decision-trees-R

2] https://www.slideshare.net/Simplilearn/decision-tree-in-r-decision-tree-algorithm-data-science-tutorial-machine-learning-simplilearn?qid=e01009b9-ede4-4ff3-b1ab-f6c8ec5933e3&v&b&from_search=1

3] https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0