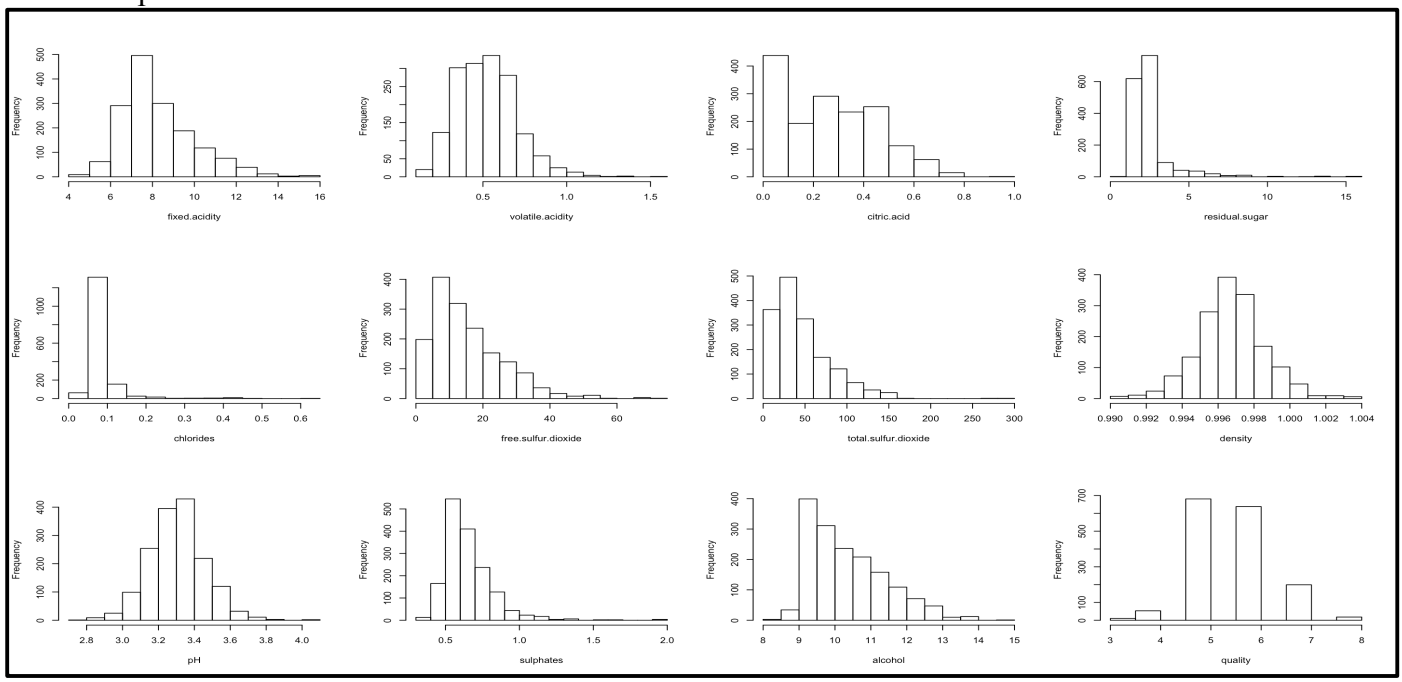
ASSIGNMENT 7

Two datasets selected for the assignment are- **Red & White Wine datasets**

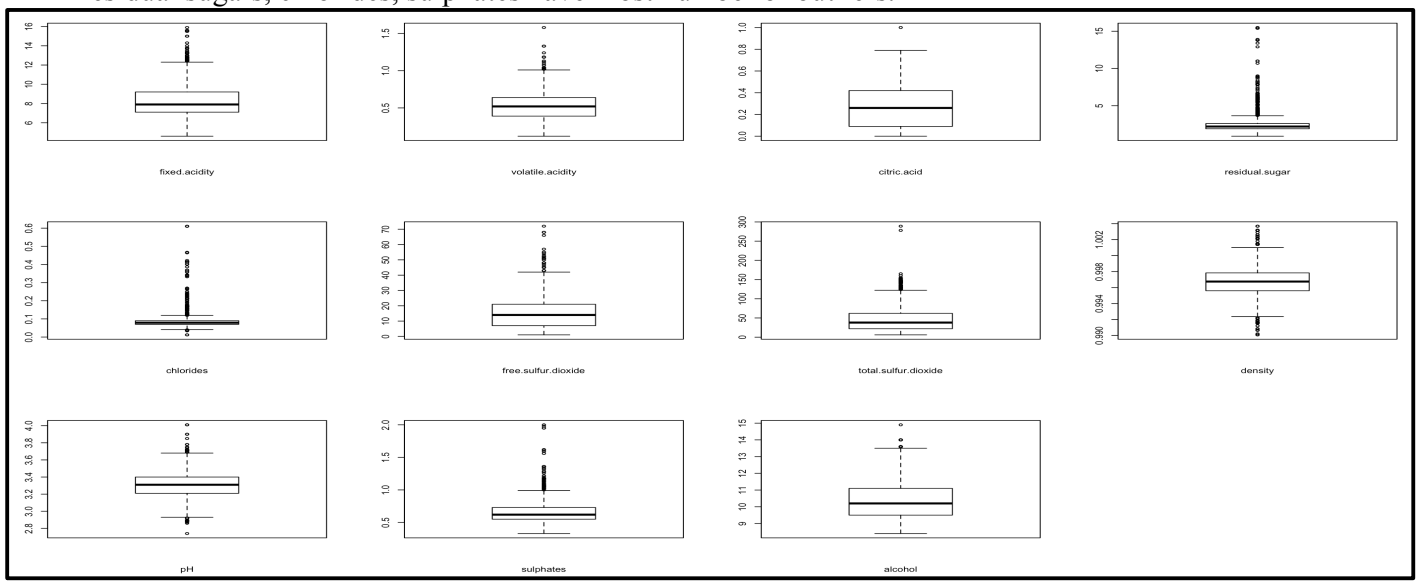
1. Red Wine quality prediction:

I. Exploratory Data Analysis:

- Upon loading the dataset, structure of dataset is checked to ensure its correctness. From this step, we know all predictor variables are of 'num' type and quality variable is of 'int' type.
- First step, is to plot histogram of individual variables to ascertain its respective distributions. From the below plots we can conclude that the distributions for variables are non-normal.

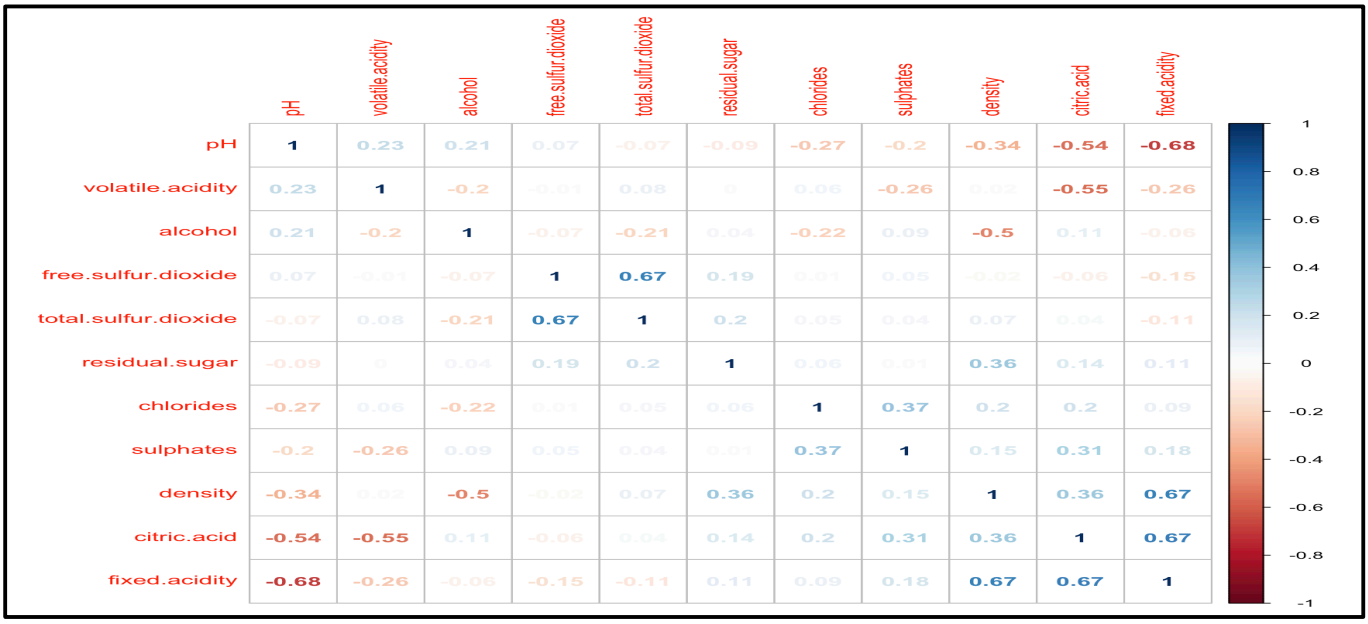


- Boxplots are plotted for each of the variables (excluding quality) as spread indicator and for presence of visual inspection of outliers. Some observations from below boxplots are as follows:
- All variables present in the dataset have outliers and most are present on larger side.
- Some of the variables such as Alcohol, citric acid, pH, density have relatively fewer outliers. While, residual sugars, chlorides, sulphates have most number of outliers.

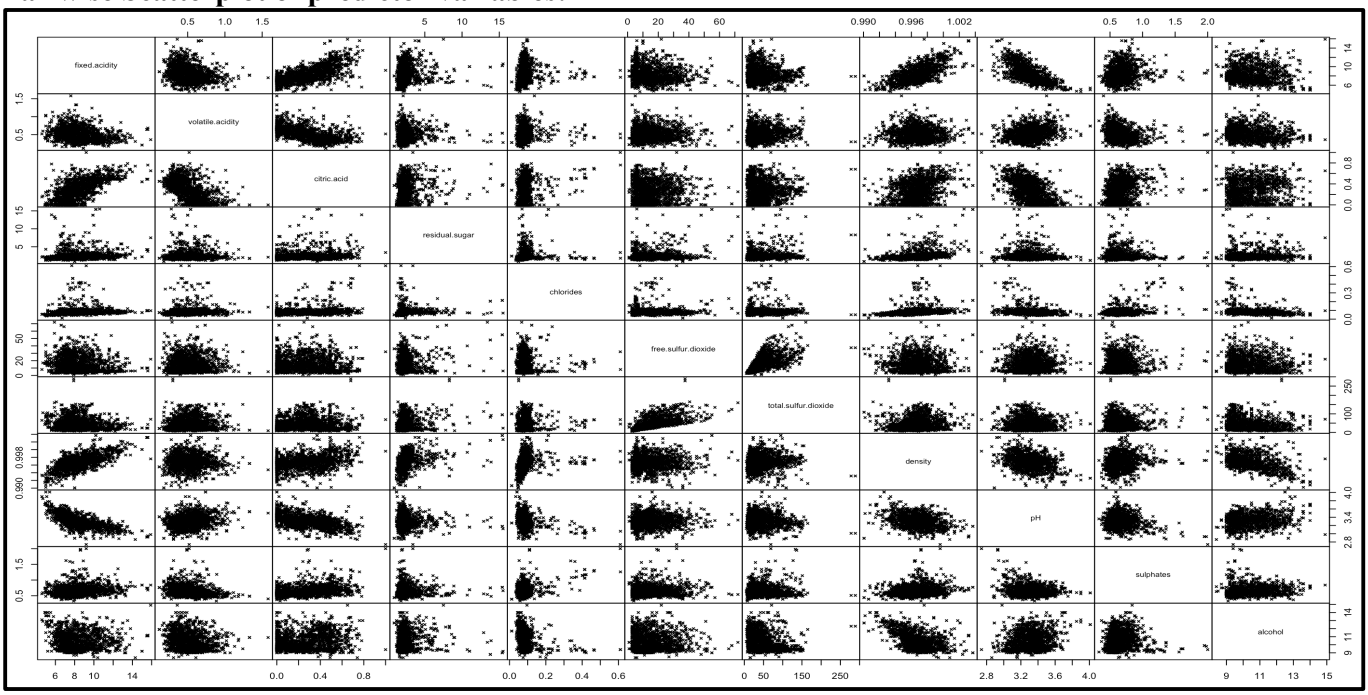


- After inspecting individual variables, bivariate analysis is carried out using correlation matrix & scatter plots for the predictor variables of the dataset.
- For correlation matrix, first Pearson method was used. Since, the variables follow non-normal distributions Spearman rank correlation was carried out. But, both the correlation matrices were very close, hence only Pearson correlation is considered.
- Correlation matrix is plotted using the *corrplot()* library. Positive correlations are displayed in blue while negative correlations in red. Color intensity is proportional to the strength of correlation coefficients.

Pearson correlation matrix is as follows:



Pairwise Scatterplot of predictor variables:



Data Preparation:

Using boxplot, we identified the presence of outliers in the dataset. But, there is no indication if it is bad data or incorrect values. Thus, none of outliers from variables were removed from the dataset. Data is randomly divided into Training data and Test Data of sizes 60% and 40% respectively.

II. Model Development, Validation, Optimization and Tuning:

The 3 models used for this dataset are as follows:

1. **Stepwise Multiple Regression**

- Applying multiple regression with all the predictor variables, gives the following output:

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.4893 -0.5216 -0.0484  0.5216  3.0216

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.363e+02  5.433e+01   4.349 1.52e-05 ***
fixed.acidity         1.663e-01  5.324e-02   3.123  0.00185 **
volatile.acidity     -1.830e+00  2.665e-01  -6.868 1.18e-11 ***
citric.acid          -1.788e-01  2.073e-01  -0.862  0.38872
residual.sugar        1.011e-01  2.040e-02   4.958 8.44e-07 ***
chlorides            -4.002e-01  1.177e+00  -0.340  0.73384
free.sulfur.dioxide   9.043e-03  2.285e-03   3.958 8.14e-05 ***
total.sulfur.dioxide -4.644e-04  9.084e-04  -0.511  0.60933
density              -2.391e+02  5.501e+01  -4.347 1.53e-05 ***
pH                    1.332e+00  2.672e-01   4.985 7.36e-07 ***
sulphates             1.051e+00  2.501e-01   4.201 2.91e-05 ***
alcohol               1.142e-01  6.696e-02   1.706  0.08835 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8082 on 947 degrees of freedom
Multiple R-squared:  0.2954,        Adjusted R-squared:  0.2872
F-statistic:  36.1 on 11 and 947 DF,  p-value: < 2.2e-16
```

- To improve the model, I use vif from cars library. VIF is variance inflation factor that helps us reduce detect multicollinearity between predictor variables. We get following results:

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
3.252230	1.162167	1.289004	15.287979	1.213947	2.062968
total.sulfur.dioxide	**density**	**pH**	**sulphates**	**alcohol**	
2.298508	32.804706	2.604048	1.233585	8.709529	

- For extremely high VIF, density was removed from the model. The results below show that VIFs improved after density is removed.

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
1.581513	1.159709	1.278304	1.681277	1.176287	2.002006
total.sulfur.dioxide	**pH**	**sulphates**	**alcohol**		
2.121429	1.546447	1.112795	1.549152		

- Not all predictors are significant and needed to predict the quality. A backward selection method is employed to build a working model. StepAIC from MASS library was used to determine the predictor variable required for regression model.

- **Final Step: AIC=-382.45**

quality ~ volatile.acidity + residual.sugar + free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates + alcohol

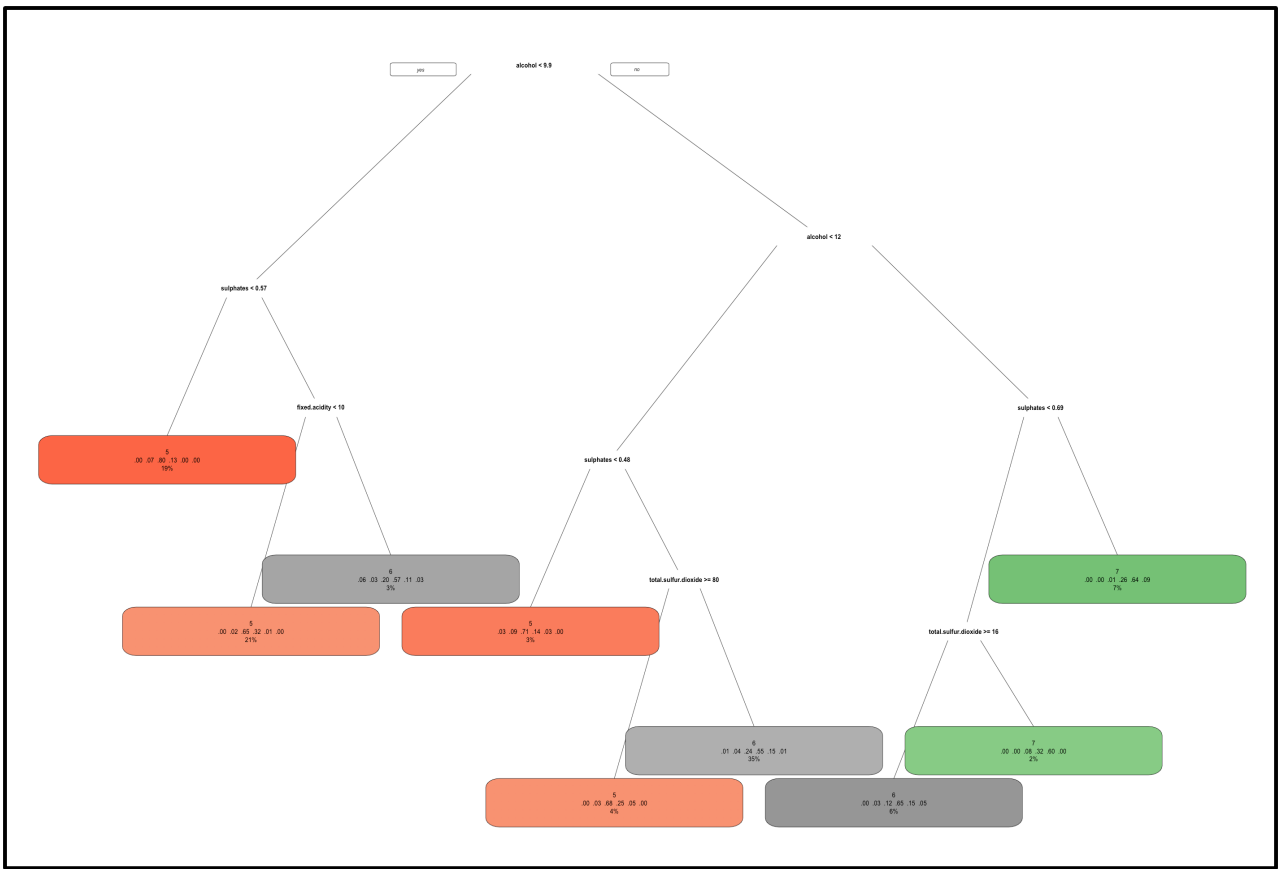
```
Residuals:
```

```
   Min      1Q  Median      3Q     Max
-3.6790 -0.4769 -0.0455  0.4645  2.6499

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.5827028  0.2870031   5.515 3.72e-08 ***
volatile.acidity    -1.9465660  0.1231936 -15.801  < 2e-16 ***
residual.sugar       0.0285843  0.0027220  10.501  < 2e-16 ***
free.sulfur.dioxide  0.0044537  0.0008894   5.007 5.76e-07 ***
total.sulfur.dioxide -0.0011070  0.0004060  -2.726 0.006434 **
pH                   0.1706523  0.0818111   2.086 0.037049 *
sulphates            0.3602988  0.1059060   3.402 0.000675 ***
alcohol              0.3716076  0.0115337  32.219  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7388 on 3931 degrees of freedom
Multiple R-squared:  0.3558,        Adjusted R-squared:  0.3617
F-statistic: 213.8 on 7 and 3931 DF,  p-value: < 2.2e-16
```

- After carrying out steps mentioned above, we improved the adjusted R^2 from 29% to 36%.
- Applying this model on test data, using *predict()* I find the mean difference between quality variable of test dataset and squared value of predicted quality. I obtained a mean squared error of 0.42.

2. Decision Trees(Classification):

- For decision trees, we use 3 levels to predict quality which is **good, bad and normal**. I made a new variable that stores response of these 3 levels. Here, quality is made a factor for the model. I plot the decision tree using rpart() library available.



- Apply decision tree to my test data, we get,

Confusion Matrix and Statistics:

wine_redPred	3	4	5	6	7	8
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	3	6	122	56	2	0
6	1	3	73	135	35	2
7	0	1	2	19	18	2
8	0	0	0	0	0	0

Overall Statistics

Accuracy : 0.7529
95% CI : (0.5273, 0.6176)
No Information Rate : 0.4375
P-Value [Acc > NIR] : 1.768e-09

Kappa : 0.2899
Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: 3	Class: 4	Class: 5	Class: 6	Class: 7	Class: 8
Sensitivity	0.000000	0.00000	0.6193	0.6429	0.3273	0.000000
Specificity	1.000000	1.00000	0.7633	0.5778	0.9435	1.000000
Pos Pred Value	NaN	NaN	0.6455	0.5422	0.4286	NaN
Neg Pred Value	0.991667	0.97917	0.7423	0.6753	0.9155	0.991667

Prevalence	0.008333	0.02083	0.4104	0.4375	0.1146	0.008333
Detection Rate	0.000000	0.00000	0.2542	0.2812	0.0375	0.000000
Detection Prevalence	0.000000	0.00000	0.3937	0.5188	0.0875	0.000000
Balanced Accuracy	0.500000	0.50000	0.6913	0.6103	0.6354	0.500000

- From this confusion matrix, we know that the accuracy for my model is 75.29%. Further, I applied tree pruning to try improve my results. I found it does not significantly change my accuracy.

3. Random Forests:

Decision trees are simplistic and most of times outperformed by other algorithms. Random Forests are one way to improve its performance. The algorithm starts by building out trees similar to the way a normal decision tree algorithm works. However, every time a split has to made, it uses only a small random subset of features to make the split instead of the full set of features. It builds multiple trees using the same process, and then takes the average of all the trees to arrive at the final model. This works by reducing the amount of correlation between trees, and thus helping reduce the variance of the final tree.

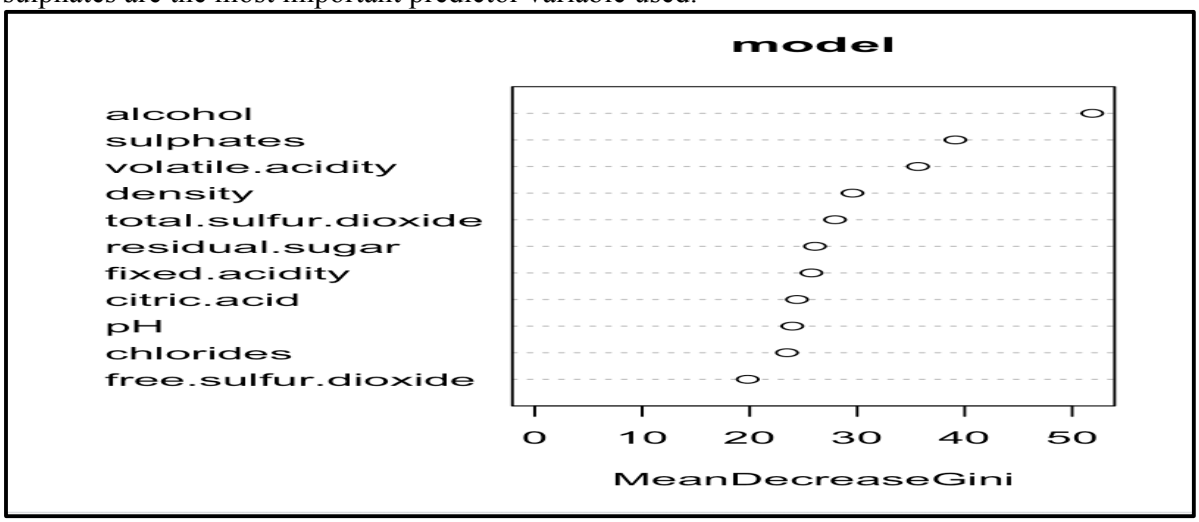
- First we provide 70% of data as test data for training. We get the following output:
  randomForest(taste ~ . - quality, data = train, ntree=1000, mtry=5)

  Call:
  randomForest(formula = taste ~ . - quality, data = train, ntree = 1000, mtry = 5)
  Type of random forest: classification
  Number of trees: 1000
  No. of variables tried at each split: 3

  OOB estimate of error rate: 13.32%
  Confusion matrix:

	bad	good	normal	class.error
bad	0	0	44	1.00000000
good	0	68	79	0.53741497
normal	1	25	902	0.02801724

- To improve the model, I tried different values ranging from 3 to 7 for 'mtry'. *Mtry* refers to the number of variables randomly sampled as candidates at each split. The best output received at *mtry* value of 5.
- I have plotted variable importance plot for my train data model. From this we interpret that alcohol and sulphates are the most important predictor variable used.



III. How should we use this information?

- After apply test data created to the random forest model, we get the following output:

pred	bad	good	normal
bad	0	0	0
good	0	31	8
normal	19	39	383

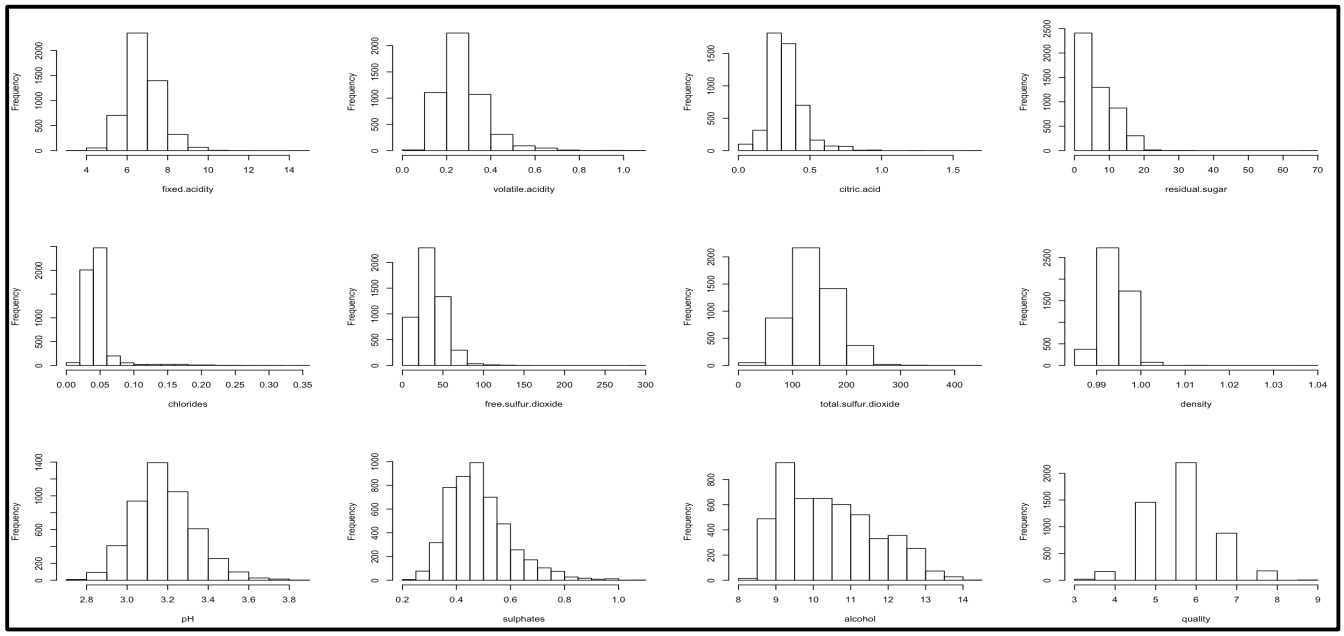
Accuracy is calculated as follows:
Accuracy = sum of diagonal elements divide by total number of elements:
Accuracy = 414/480 = 86.25%
Thus, accuracy of correcting predicting the quality of wine using random forests is 86.25%.

- Once we have some idea on which of the 11 properties are more strongly correlated with the wine quality ratings, then we can use them to serve us on many directions
  1. we can use the list of strong predictors jointly into a model to predict the quality of red wine.
  2. Out of 3 models, random forests are the best model with 86.25% accuracy.
  3. one can better control the quality of wines during their whole making processes.

2. White wine quality prediction:

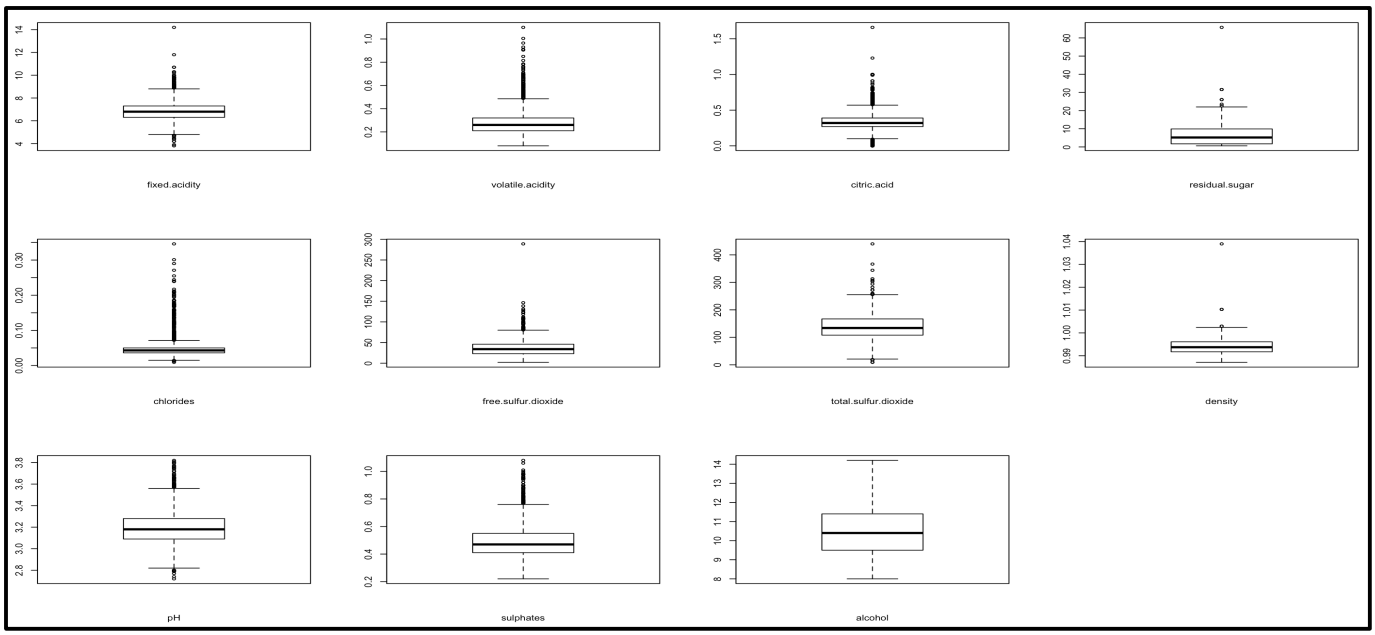
I. Exploratory Data Analysis:

- Upon loading the dataset, structure of dataset is checked to ensure its correctness. From this step, we know all predictor variables are of 'num' type and quality variable is of 'int' type.
- First step, is to plot histogram of individual variables to ascertain its respective distributions. From the below plots we can conclude that the most distributions for variables are non-normal.



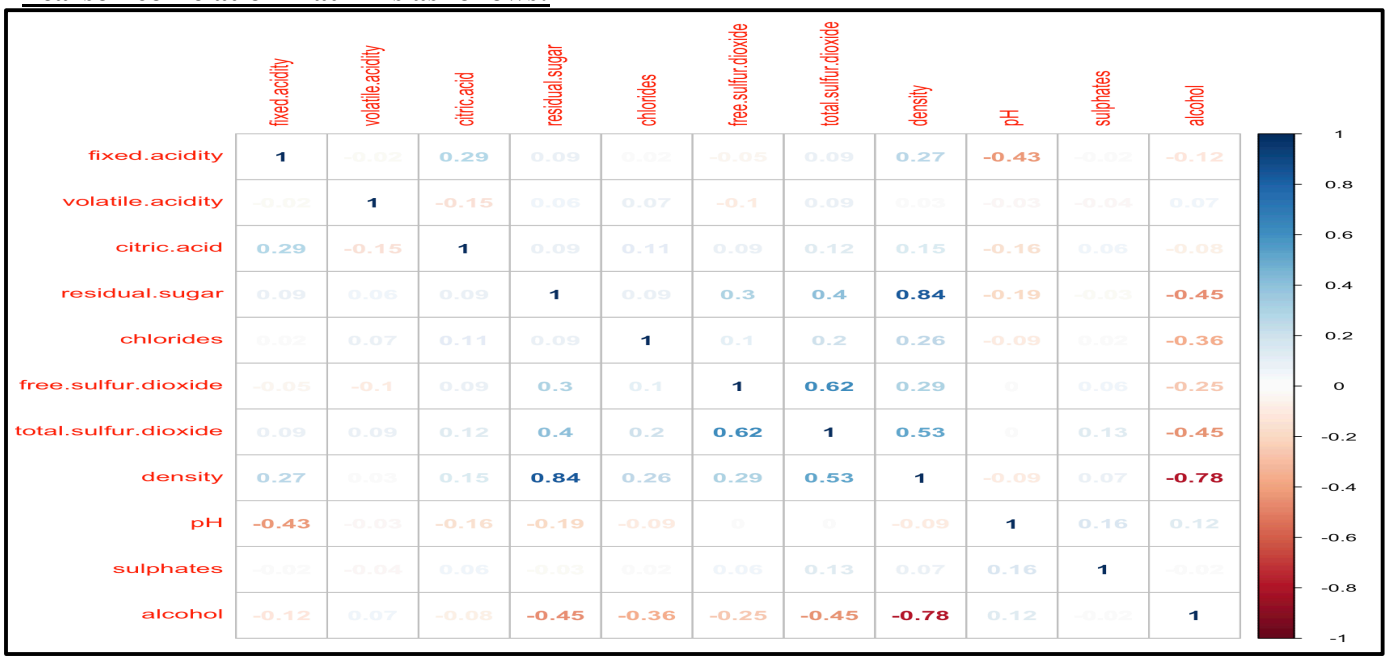
Boxplots are plotted for each of the variables (excluding quality) as spread indicator and for presence of visual inspection of outliers. Some observations from below boxplots are as follows:

- All variables present in the dataset have outliers and most are present on larger side.
- Variables such as volatile acidity, chlorides, sulphates, citric acid has most number of outliers. Residual sugars and density have very few outliers and alcohol has no outliers present in it.

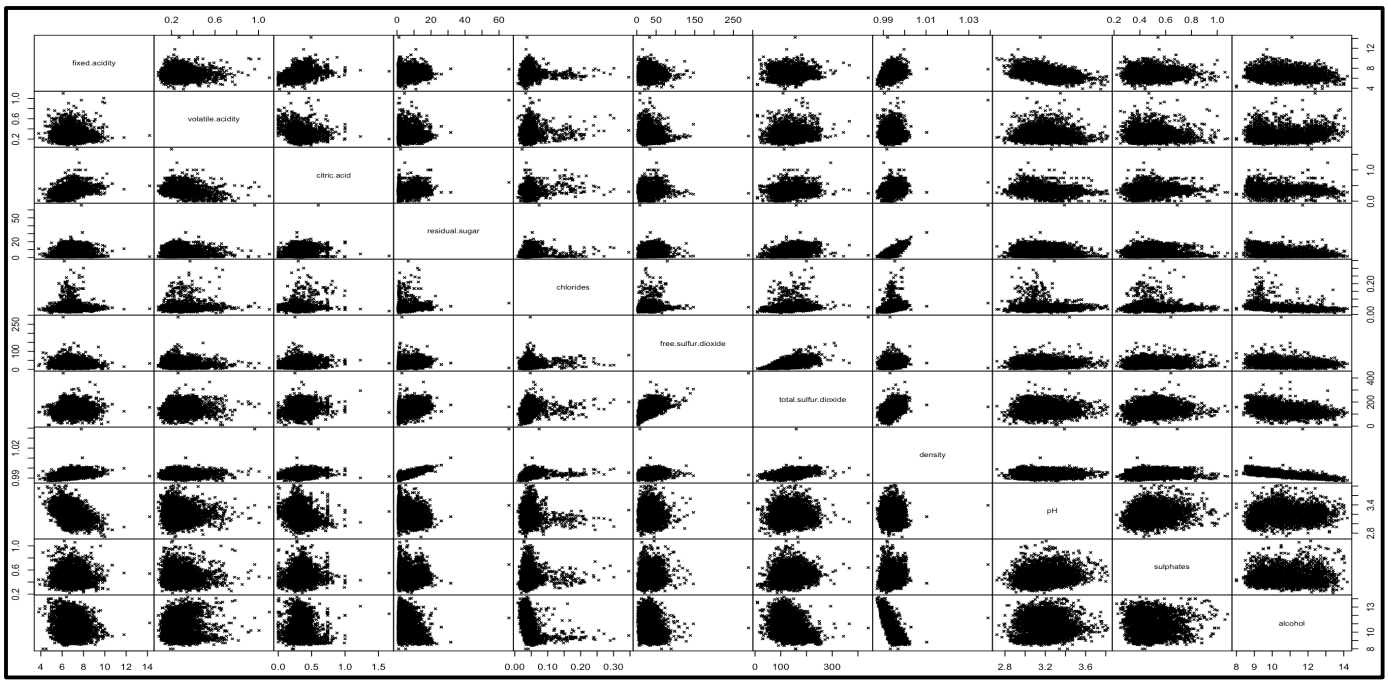


- After inspecting individual variables, bivariate analysis is carried out using correlation matrix & scatter plots for the predictor variables of the dataset.
- For correlation matrix, first Pearson method was used. Since, the variables follow non-normal distributions Spearman rank correlation was carried out. But, both the correlation matrices were very close, hence only Pearson correlation is considered.
- Correlation matrix is plotted using the *corrplot()* library. Positive correlations are displayed in blue while negative correlations in red. Color intensity is proportional to the strength of correlation coefficients.

Pearson correlation matrix is as follows:



Pairwise Scatterplot of predictor variables:



Data Preparation:

Using boxplot, we identified the presence of outliers in the dataset. But, there is no indication if it is bad data or incorrect values. Thus, none of outliers from variables were removed from the dataset. Data is randomly divided into Training data and Test Data of sizes 60% and 40% respectively.

II. Model Development, Validation, Optimization and Tuning:

The 3 models used for this dataset are as follows:

1. **Stepwise Multiple Regression**

- Applying multiple regression with all the predictor variables, gives the following output:

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.8076 -0.4978 -0.0203  0.4686  3.1445

Coefficients:     Estimate Std.   Error        t value   Pr(>|t|)
(Intercept)         2.089e+02  2.962e+01   7.052    2.20e-12 ***
fixed.acidity        7.963e-02  3.013e-02   2.643    0.00827 **
volatile.acidity    -1.878e+00  1.555e-01 -12.082  < 2e-16 ***
citric.acid          1.259e-01    1.306e-01   0.965   0.33480
residual.sugar     1.015e-01  1.117e-02     9.083   < 2e-16 ***
chlorides          -1.723e-01  7.356e-01    -0.234   0.81482
free.sulfur.dioxide 3.052e-03  1.078e-03   2.832   0.00466 **
total.sulfur.dioxide  1.856e-04  4.993e-04   0.372  0.71014
density            -2.093e+02     3.001e+01  -6.974  3.80e-12 ***
pH                  8.490e-01      1.447e-01      5.869    4.89e-09 ***
sulphates           5.692e-01    1.328e-01      4.287    1.87e-05 ***
alcohol             1.165e-01    3.765e-02      3.094     0.00199 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7575 on 2926 degrees of freedom
Multiple R-squared:  0.2648,  Adjusted R-squared:  0.262
F-statistic: 95.78 on 11 and 2926 DF,  p-value: < 2.2e-16
```

- To improve the model, I use vif from cars library. VIF is variance inflation factor that helps us reduce detect multicollinearity between predictor variables. We get following results:

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
3.117348	1.142577	1.150855	15.724936	1.233290	1.776700

total.sulfur.dioxide	density	pH	sulphates	alcohol
2.296963	38.226609	2.452310	1.170184	10.758485

- For extremely high VIF, density was removed from the model. The results below show that VIFs improved after density is removed.

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
1.330562	1.134158	1.148733	1.452498	1.201208	1.732925

total.sulfur.dioxide	pH	sulphates	alcohol
2.159559	1.327984	1.055366	1.667012

- Not all predictors are significant and needed to predict the quality. A backward selection method is employed to build a working model. StepAIC from MASS library was used to determine the predictor variable required for regression model.
- Start: AIC=-1573.25
  quality ~ (fixed.acidity + volatile.acidity + citric.acid + residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide + density + pH + sulphates + alcohol) - density

  Final Step: AIC=-1575.23
  quality ~ fixed.acidity + volatile.acidity + residual.sugar + free.sulfur.dioxide + pH + sulphates + alcohol

- One observation made here is that white wine dataset requires more predictors compared to red wine dataset. Final model using fixed.acidity ,volatile.acidity,residual.sugar, free.sulfur.dioxide, pH, sulphates & alcohol is received as follows:

  Residuals:

Min	1Q	Median	3Q	Max
-3.2405	-0.4909	-0.0737	0.4597	2.7347

  Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.478272	0.520565	2.840	0.004562	**
fixed.acidity	-0.024095	0.021481	-1.122	0.262134	
volatile.acidity	-2.015837	0.159107	-12.670	< 2e-16	***
residual.sugar	0.022742	0.003773	6.027	1.99e-09	***
free.sulfur.dioxide	0.003859	0.001100	3.510	0.000458	***
pH	0.149390	0.127739	1.169	0.242347	
sulphates	0.578610	0.149834	3.862	0.000116	***
alcohol	0.387474	0.015349	25.244	< 2e-16	***

  Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

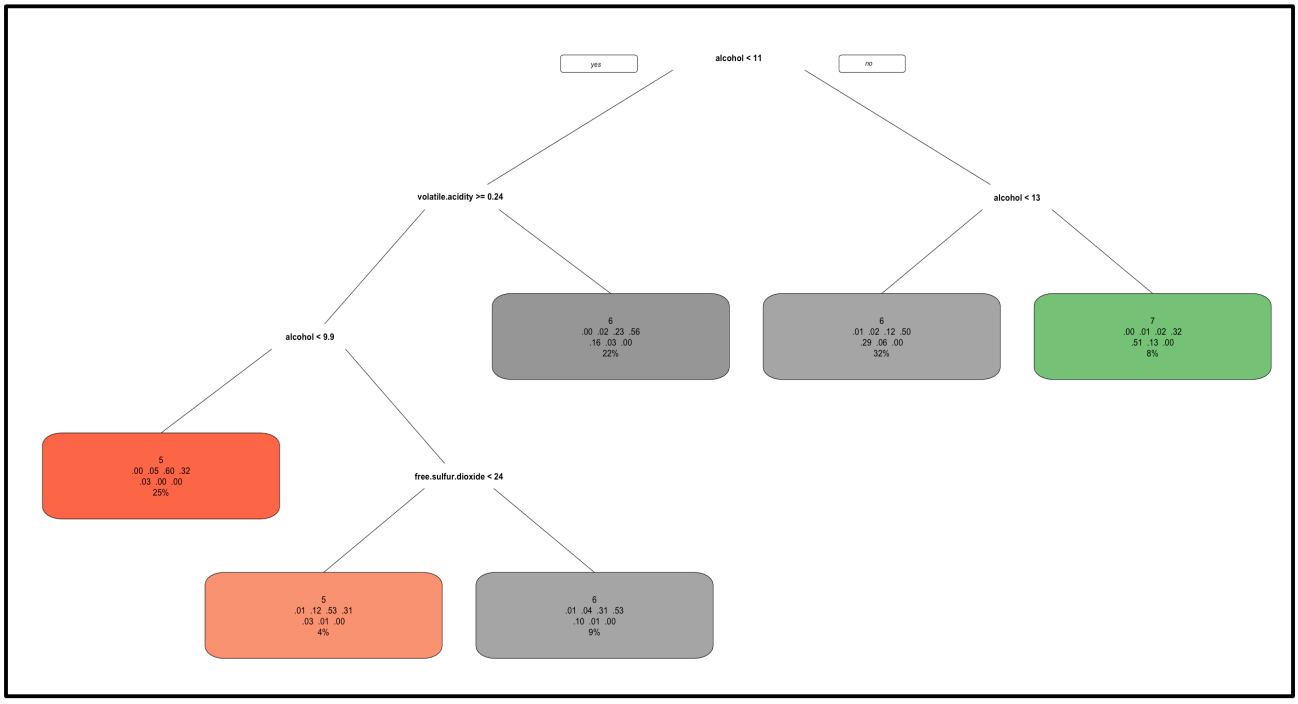
  Residual standard error: 0.7452 on 1952 degrees of freedom
  Multiple R-squared: 0.3027, Adjusted R-squared: 0.3002
  F-statistic: 121.1 on 7 and 1952 DF, p-value: < 2.2e-16

- After carrying out steps mentioned above, we improved the adjusted R^2 from 26.2% to 30.02%.

- Applying this model on test data, using *predict()* I find the mean difference between quality variable of test dataset and squared value of predicted quality. I obtained a mean squared error of 0.53 which is high but still acceptable.

2. Decision Trees(Classification):

- For decision trees, we use 3 levels to predict quality which is good, bad and normal. I made a new variable that stores response of these 3 levels. Here, quality is made a factor for the model. I plot the decision tree using rpart() library available.



- **Confusion Matrix and Statistics:**

wine_whitePred	3	4	5	6	7	8	9
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	3	23	217	111	10	1	0
6	3	16	167	412	167	23	0
7	0	1	2	28	32	8	1
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0

Overall Statistics:

Accuracy : 0.6396
95% CI : (0.5112, 0.5678)
No Information Rate : 0.4498
P-Value [Acc > NIR] : 1.823e-10

Kappa : 0.2412
Mcnemar's Test P-Value : NA

Statistics by Class:

Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8 Class: 9

Sensitivity	0.000000	0.00000	0.5622	0.7477	0.15311	0.00000	0.0000000
Specificity	1.000000	1.00000	0.8236	0.4421	0.96063	1.00000	1.0000000
Pos Pred Value	NaN	NaN	0.5945	0.5228	0.44444	NaN	NaN
Neg Pred Value	0.995102	0.96735	0.8035	0.6819	0.84649	0.97388	0.9991837
Prevalence	0.004898	0.03265	0.3151	0.4498	0.17061	0.02612	0.0008163
Detection Rate	0.000000	0.00000	0.1771	0.3363	0.02612	0.00000	0.0000000
Detection Prevalence	0.000000	0.00000	0.2980	0.6433	0.05878	0.00000	0.0000000
Balanced Accuracy	0.500000	0.50000	0.6929	0.5949	0.55687	0.50000	0.5000000

- From this confusion matrix, we know that the accuracy for my model is 63.96%. Further, I applied tree pruning to try improving my results. I found it does not significantly change my accuracy.

3. Random Forests:

- Decision trees are simplistic and most of times outperformed by other algorithms. Random Forests are one way to improve its performance. The algorithm starts by building out trees similar to the way a normal decision tree algorithm works. However, every time a split has to made, it uses only a small random subset of features to make the split instead of the full set of features. It builds multiple trees using the same process, and then takes the average of all the trees to arrive at the final model. This works by reducing the amount of correlation between trees, and thus helping reduce the variance of the final tree.
- Following output is received for taste variable after classing quality in to three levels.
  table(wine_white$taste)

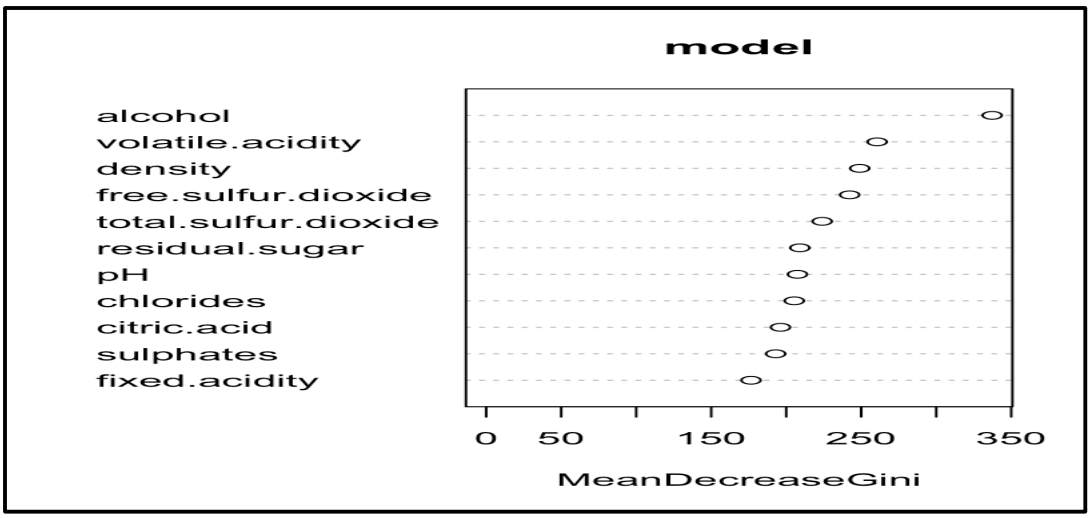
bad	good	normal
1640	1060	2198

- First we provide 80% of data as test data for training. We get the following output:

```
Call:
 randomForest(formula = taste ~ . - quality, data = train, ntree = 1002,      mtry = 5)
               Type of random forest: classification
                     Number of trees: 1002
No. of variables tried at each split: 5

        OOB estimate of  error rate: 27.62%
Confusion matrix:
       bad good normal class.error
bad    967   13    349   0.2723853
good    17  509    304   0.3867470
normal 262  137   1360   0.2268334
```

- To improve the model, I tried different values ranging from 3 to 7 for 'mtry'. *Mtry* refers to the number of variables randomly sampled as candidates at each split. The best output received at *mtry* value of 5.

4. I have plotted variable importance plot for my train data model. From this we interpret that alcohol and volatile.acidity are the most important predictor variable used.



III. How should we use this information?

- After apply test data created to the random forest model, we get the following output:

pred	bad	good	normal
bad	233	5	67
good	1	136	29
normal	77	89	343

Accuracy is calculated as follows:
Accuracy = sum of diagonal elements divide by total number of elements:
Acc = (233+136+343)/980 = 82.65%

- Thus, accuracy of correcting predicting the quality of wine using random forests is 82.65%.

- Once we have some idea on which of the 11 properties are more strongly correlated with the wine quality ratings, then we can use them to serve us on many directions.
- Out of 3 models, random forests are the best model with 82.65% accuracy.
- we can use the list of strong predictors jointly into a model to predict the quality of a specific variant of white wine.
- one can better control the quality of wines during their whole making processes.