Khoury College of Computer and Information Science,
Northeastern University,
Boston, MA
April 2022

Exploratory Analysis of Wine Quality Evaluation by Physicochemical Factors

Group 4:

Abhishek Reddy Andluru

Bindu Latha Banisetti

Harshitha Somala

Venkata Sai Ujwala Bayana

Prajwal Chinchmalatpure

1. Summary:

1.1. Overview:

In recent years wine is treated as a luxury good and is enjoyed by a wide range of consumers. The Wine industry is investing in new technologies for making and selling wine in order to support its growth. In this context, wine certification and quality assessment are major determining factors. Hence, we are focusing on Quality Evaluation which is often a part of the certification process that involves various factors affecting the quality.

1.2. Goals:

Our aim with this project is to predict the set of factors/attributes that play a major role in determining the quality of the Red and the White wine individually. We intend to traverse through multiple physicochemical and sensory factors that affect wine quality by doing an exploratory analysis for the same. The dataset we have chosen consists of 12 attributes where 11 of them represent the potential quality determining factors for a wine. And the 12th attribute represents the quality of the wine on a scale of 1 to 10. We plan to use different regression techniques to train the model. Furthermore, calculating the error metric (RMSE, MAE) to pick the best fit model that accurately selects the factors affecting wine quality.

1.3. Data Description:

The wine datasets were taken from the UCI Machine Learning Repository. The red wine dataset includes 1599 observations, and the white wine dataset includes 4898 observations. Both the datasets include 11 exploratory variables which focus on the quality (12^{th} variable) of the wine. The attributes of the dataset are as follows:

1. *Fixed Acidity*: Tartaric Acid content in wine (g/dm3).
2. *Volatile Acidity*: Acetic Acid content in wine (g/dm3).
3. *Citric Acid*: Citric acid content in wine (g/dm3). Contributes to the wine's crispness.
4. *Residual Sugar*: Sugar content left in the wine after fermentation (g/dm3).
5. *Chlorides*: Sodium Chloride (salt) content in wine (g/dm3)
6. *Free Sulfur Dioxide*: SO2 content in free form (g/dm3)
7. *Total Sulfur Dioxide*: Total Amount of SO2(g/dm3). SO2 acts as an antioxidant & antimicrobial agent. Too much SO2 leads to deriving a pungent smell.
8. *Density*: Density of Wine (g/dm3)
9. *pH*: pH of Wine on a scale of 0-14. 0 means highly Acidic, while 14 means highly basic.
10. *Sulphates*: Potassium Sulphate content in wine (measured in g/dm3). Contributes to the formation of SO2.
11. *Alcohol*: Alcohol content in wine (in terms of % volume)
12. *Q*uality: Wine Quality is graded on a scale of 1 - 10 (Higher is better)

The Appendix figure 1 and 2 series represent the relationship between the response (quality) and all other predictors individually.

2. Methods:

2.1. Data Preprocessing:

We worked with two files red wine (winequality-red.csv) & white wine (winequality-white.csv). For processing the data, we first checked for the null values in both the red and white wine datasets and concluded that there were no null values present in either of the datasets. Secondly, we drew an inference from the dataset that the three attributes of Fixed Acidity, Volatile acidity, and Citric acidity are in fact a representation of wine samples' Total Acidity. Furthermore, all the three acidity attributes have the same measurement unit of g/dm3, implying that we can augment a new column called Total acidity as the sum of three existing acidity attributes. We were able to reduce three acidity attributes into one total acidity attribute which might facilitate in building of a simple linear model that we perform in further steps. However, to replace these three acidic features with `Total Acidity` we need to check whether all three attributes contribute to the variations in the quality of the wine. This confirmation will be drawn out from the forward selection process of the Linear Regression model.

- Checking NULL values for Red and white wine data frames:

```
which(is.na(red_wine_df))
which(is.na(white_wine_df))
```

```
integer(0)
integer(0)
```

- Data Standardization:
  We also performed standardization to bring down all the attributes to a common scale without distorting the differences in the range of the values.

```
#Standardization of data
red_wine_std <- data.frame(scale(red_wine_df[1:11]))
white_wine_std <- data.frame(scale(white_wine_df[1:11]))
```

- Partition of data for Training, Validation & Testing:
  - Partitioning of the dataset should be done to avoid overfitting of the model. To eliminate the overfitting of the model, the predictive performance of the model should be evaluated on the data which is not used for training the model.
  - We have partitioned the data into 3 subsets for model selection. 50% of the data is used for training the model. 25% of data is used for validating and the remaining 25% is used for testing the model.

- The Appendix figure 3.1. and 3.2 explains the correlation between the features of both red and white wine respectively.

2.2 Data Modelling:

Linear Regression Models:

Linear regression is a statistical model that analyzes the relationship between a scalar response variable (often called y) and one or more independent variables and their interactions (often called x or explanatory variables).

A. Stepwise Model Selection (Forward selection):

- Machine learning, more specifically the field of predictive modeling, is primarily concerned with minimizing the error of a model or making the most accurate predictions possible, at the expense of explainability.
- To select the best fit model, we used the stepwise forward model selection technique and the RMSE error metric, as previously mentioned.
- Forward selection is an iterative technique that begins with an empty model and calculates the individual RMSEs of each variable in each iteration. The variable with the least RMSE is the best predictor and is added to the model in each iteration.
- For this process, we defined a function to which we send the response variable, vector of predictors (initially just 1), the vector of candidate predictors (initially includes all the predictors), and the dataset, which is partitioned for training, validation, and test purposes. Thereby, we have one predictor for our model which is 1. So, in this function, models will be built on each combination of an additional predictor taken from the candidate vector. And then chooses that model whose RMSE is the lowest. So that models' additional predictor variable will be added to our predictor vector.
- Then, we will again call the function with the updated predictor vector and the candidate vector by excluding that predictor which was added to the predictor vector before. So again, the function would build the models and choose the one with the lowest RMSE and add that corresponding variable to the predictor's vector provided that the RMSE value is lesser than that of the previous iteration. So, this process is repeated until the RMSE value is greater than that of its prior iteration, which means the model stopped improving.
- Thereby, the result is the best fit model.
- The plots (Appendix: Figure 4 and 5) depicts the RMSE values as the number of attributes in the model grows for both red and white wine:

The plot (Appendix: Figure 4) depicts the RMSE values of Redwine. The RMSEs decreased from alcohol to fixed acidity and then began to rise from free sulfur dioxide. From pH to fixed acidity there is a very small decrease in the RMSEs which can result in overfitting. As there is a negligible change in the RMSEs after pH and the model stopped improving from free sulfur dioxide, we selected the variables from alcohols to pH as the best predictors for the model.

The next plot (Appendix: Figure 5) depicts RMSE values of White wine. The RMSEs decreased from alcohol to chlorides and then began to rise from citric acid. There is a very small decrease in the RMSEs from pH to chlorides which can result in overfitting. As there is a negligible change in the RMSEs after pH and the model stopped improving from citric acid, we selected the variables from alcohols to pH as the best predictors for the model.

It is noteworthy to mention that the attributes of Citric Acidity and Fixed Acidity were not having a notable effect on the wine quality and hence were not selected in the Forward Selection. This implies that we are not required to augment the three acidities into a new Total Acidity attribute as all three attributes together are not influencing the quality of the wine.

Output:

	wine_type	rmse	mae	rsquare
1	Red Wine	0.6509610	0.5062295	0.3485918
2	White wine	0.7568375	0.5883841	0.2691790

B. Lasso Regression Model:

The Lasso Regression method can be used to fit a regression model when multicollinearity is present in the data. To be precise, least squares regression tries to find the parameter estimates that minimize the Sum of Squared Residuals (RSS)

$$\mathbf{RSS = \Sigma(y_i - \hat{y}_i)2}$$

However, Lasso regression seeks to minimize the following:

$$\mathbf{RSS + \lambda\Sigma|\beta_j|}$$

Where j ranges from 1 to n predictor variables and lambda >=0
The second term in the equation is also called Shrinkage Penalty. In Lasso Regression we select the lambda value that minimizes the mean squared error.

Following steps can be followed in the Lasso Regression:

1. For Lasso regression we use glmnet package, for which the response variable should be a vector and the set of predictor variables should be of class data.matrix

```
response_rw<-red_wine_df$quality
predictors_rw<-data.matrix(red_wine_df[,c("fixed.acidity","volatile.acidity","citric.acid",
                          "residual.sugar","chlorides","free.sulfur.dioxide",
                          "total.sulfur.dioxide","density","pH","sulphates",
                          "alcohol")])
```

2. Fitting the Lasso Regression model using glmnet package and specifying the alpha=1. Appendix: Figure 6 and Figure 7 depict how the coefficients change for different values of lambda for Red Wine and White wine respectively.

```
red_ls_model<-glmnet(predictors_rw,response_rw,alpha=1)
plot(red_ls_model,xvar="lambda",label=TRUE)
```

3. To determine the value of lambda, we will perform k-fold cross validation to identify the lambda value that produces the least Means Square Error. In K-fold cross validation it automatically performs k=10-fold cross validation. The lambda value that minimizes the Mean Squared Error is: 0.004850794. Appendix Figure 8 and 9 shows the Mean Squared Error values at different lambda values, for Red and White wines respectively.

```
# Using k-fold cv to get best optimal lambda value
red_cv_ls_model<-cv.glmnet(predictors_rw,response_rw,alpha=1)

#produce plot of MSE values for each lambda
plot(red_cv_ls_model)

#find optimal lambda value that minimizes the MSE
best_lambda_rw<-red_cv_ls_model$lambda.min
best_lambda_rw
```

4. We can analyze the final model, by using the optimal lambda obtained. We can get the coefficients of predictor variables of our best model. There the coefficient of density is null, implying that density is relatively not important enough to predict the quality of the wine.

```
best_model_ls_rw<-glmnet(predictors_rw,response_rw,alpha=1,lambda=best_lambda)
coef(best_model_ls_rw)
```

```
12 x 1 sparse Matrix of class "dgCMatrix"
                              s0
(Intercept)           4.311058572
fixed.acidity         0.001590786
volatile.acidity     -1.045885442
citric.acid          -0.056043048
residual.sugar        0.004545097
chlorides            -1.812127522
free.sulfur.dioxide   0.003379133
total.sulfur.dioxide -0.002981089
density               .
pH                   -0.437661346
sulphates             0.850821345
alcohol               0.287823262
```

```
12 x 1 sparse Matrix of class "dgCMatrix"
                               s0
(Intercept)           1.135530e+02
fixed.acidity         3.015296e-02
volatile.acidity     -1.875380e+00
citric.acid           .
residual.sugar        6.637850e-02
chlorides            -3.934651e-01
free.sulfur.dioxide   3.599613e-03
total.sulfur.dioxide -2.344057e-04
density              -1.129171e+02
pH                    5.208469e-01
sulphates             5.556340e-01
alcohol               2.329256e-01
```

5. Finally, we calculate the Rsquared of our model.

```
#Finding the SST and SSE and Rsquare
sst_rw<-sum((actual_quality_rw-mean(actual_quality_rw))^2)
sse_rw<-sum((actual_quality_rw-predicted_quality_rw)^2)
rsq_ls_rw<-1-(sse_rw/sst_rw)
rsq_ls_rw

#Finding the RMSE and MAE
n_rw<-nrow(red_wine_df)
rmse_ls_rw<-sqrt(sum((actual_quality_rw-predicted_quality_rw)^2)/n_rw)
rmse_ls_rw
mae_ls_rw<-sum(abs(actual_quality_rw-predicted_quality_rw))/n_rw
mae_ls_rw
```

Output:

	wine_type	rmse	mae	rsquare
1	Red Wine	0.6460953	0.5020461	0.3595207
2	White wine	0.7506515	0.5841103	0.2814577

C. Ridge Regression Model:

Ridge regression is a model tuning technique for analyzing multicollinear data. L2 regularization is used in this procedure. When there is a problem with multicollinearity, the least-squares method is unbiased, and the variances are enormous, resulting in projected values that are distant from the actual values.
The cost function for ridge regression:

Min (||Y – X(θ)||^2 + λ|| θ ||^2)

The penalty term is lambda. λ given here is denoted by an alpha parameter in the ridge function. We can regulate the penalty term by varying the values of alpha. The greater the alpha value, the greater the penalty, and hence the size of the coefficients is lowered.

The foundation of every regression machine learning model is the standard regression equation, which is stated as:

Y = XB + e

The dependent variable is Y, the independent variables are X, the regression coefficients to be estimated are B, and the errors are residuals are e.

The variance that is not assessed by the general model is taken into account when the lambda function is added to this equation. There are actions that may be taken once the data has been prepared and designated as being part of the L2 regularization process.

The first step in ridge regression is to normalize the variables (both dependent and independent) by dividing by their standard deviations and removing their means. This creates a notation problem since we need to declare whether the variables in a formula are standardized or not. All ridge regression computations are based on standardized variables in terms of standardization. The final regression coefficients are rescaled to their original scale when they are shown. The ridge trace, on the other hand, is on a standardized scale.

When it comes to creating ridge regression models on a real dataset, the trade-off between bias and variance is usually challenging. However, the following is a typical pattern to keep in mind:

As λ grows, the bias rises with it. As λ grows, the variance reduces.

Ridge regression is based on the same assumptions as linear regression: linearity, constant variance, and independence. Because ridge regression does not offer confidence bounds, it is not necessary to assume that the error distribution is normal.

Output:

	wine_type	rmse	mae	rsquare
1	Red Wine	0.6328168	0.4952300	0.3767629
2	White wine	0.7352006	0.5791992	0.2795611

D. Random Forest Regression Model:

Random forests, also known as random decision forests, are an ensemble learning approach for classification, regression, and other problems that works by training a large number of decision trees. For classification tasks, the random forest's output is the class chosen by majority of trees. The mean or average forecast of the individual trees is returned for regression tasks.
A random forest is made up of Decision Trees, each of which makes its own prediction. The values are then averaged (Regression) or max voted (Classification) to arrive at the final result.
The power of this model comes from the ability to create several trees with various sub-features from the features. Because the features chosen for each tree are random, the trees do not grow in depth and are just focused on the set of features.
Finally, we develop an ensemble of Decision Trees that offers a well-learned forecast when they are combined.

What distinguishes it from the Decision Tree?
A decision tree provides a single path that takes into account all of the characteristics at the same time. As a result, deeper trees may be created, causing the model to overfit. The trees in a random forest have a variety of random traits and aren't particularly tall.
Including an Ensemble option for decision trees increases efficiency by averaging the outcomes and delivering generic findings.

While the structure of a decision tree is mainly dependent on the training data and can change dramatically even with little changes in the training data, the random selection of features allows for little variation in structure modification with data change.

This can be further reduced by employing techniques such as tagging for data selection.
Random Forests, on the other hand, demand more storage and processing power than a decision tree.

In summary, Random Forest outperforms decision trees in terms of accuracy and efficiency, but at the cost of storage and computation power.

Output:

	wine_type	rmse	mae	rsquare
1	Red Wine	0.5743168	0.4200583	0.4869987
2	White wine	0.6161880	0.4630049	0.4943317

3. Results:

- The following tables represent the integrated results of all the models for both red and white wine:

Red Wine			
Model	**RMSE**	**MAE**	**Rsquared**
Forward Selection	0.651	0.506	0.349
Lasso Regression	0.646	0.502	0.359
Ridge Regression	0.632	0.495	0.377
Random forest regression	0.574	0.420	0.487

White Wine			
Model	**RMSE**	**MAE**	**Rsquared**
Forward Selection	0.756	0.588	0.269
Lasso Regression	0.750	0.584	0.281
Ridge Regression	0.735	0.579	0.279
Random forest regression	0.616	0.463	0.494

- The Appendix Figure 10 depicts the integrated results of all the models for both red and white wine respectively:
  - In the RMSE plot (Appendix Figure 10.2), the error is less for the Random Forest model when compared to other models.
  - In the MAE plot (Appendix Figure 10.3), the error is again less for the Random Forest model when compared to other models.
  - The R-squared value explains the variation i.e., the model's accuracy representing how well our model best fits the data. The random forest model has the highest accuracy when compared to other models. (Appendix Figure 10.1)
- Appendix Figure 11 depicts the Real Vs Predicted values of the model.

4. Discussions:

- The aim was to predict the set of factors/attributes that play a key role in determining the quality of Red and White wine individually. By using forward selection, we were able to select the best set of predictors that play a key role in determining the quality of the wine.
- After performing all the models, the random forest model is the best fit model for the data and has acquired a variation of around 49% for red wine data and 50% for white wine data.
- The accuracy is less because of the dataset's size. Though the variation explained is 49-50%, the model was able to predict the results more accurately for the qualities 5,6,7 of both the wine populations. This can be explained accurately using the Real Vs Predicted response variable plots, where the qualities 5,6,7 of both the wines have a considerable number of observations when compared to other quality wines as shown in the Appendix: Figure 12.
- Hence, we can infer from the Real Vs Predicted response variable plots (Appendix: Figure 13) that the model is working fine for the data with more observations. Therefore, if the dataset is more balanced with a greater number of observations, it would be more feasible for the data to train the model and produce more accurate results for the test data.

5. Statement of Contribution

- Abhishek Reddy Andluru : Ridge Regression
- Bindu Latha Banisetti: Lasso Regression
- Harshitha Somala: Linear Regression – Stepwise Model Selection
- Prajwal Chinchmalatpure: Random Forest Regression
- Venkata Sai Ujwala Bayana: EDA, Data Preprocessing

6. References:

- Data Source: https://archive.ics.uci.edu/ml/datasets/wine+quality
- Research paper: https://www.sciencedirect.com/science/article/pii/S0167923609001377?via%3Dihub

7. Appendix:

Response Vs Predictors – Red Wine:

Fig 1: Alcohol Vs Quality

Fig 1.2: Chlorides Vs Quality

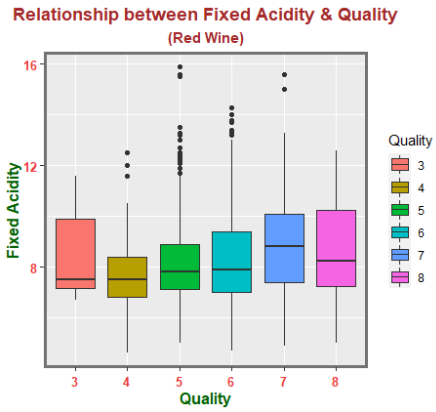

Fig 1.3: Fixed Acidity Vs Quality

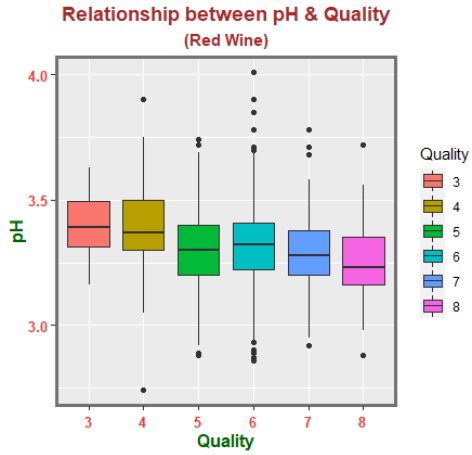

Fig 1.4: pH Vs Quality

Fig 1.5: Sulphates Vs Quality

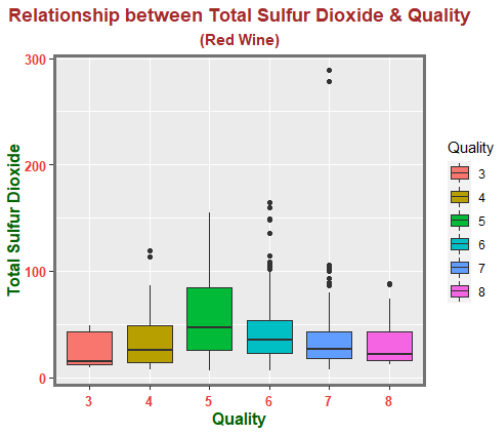

Fig 1.6: Total Sulfur Dioxide Vs Quality

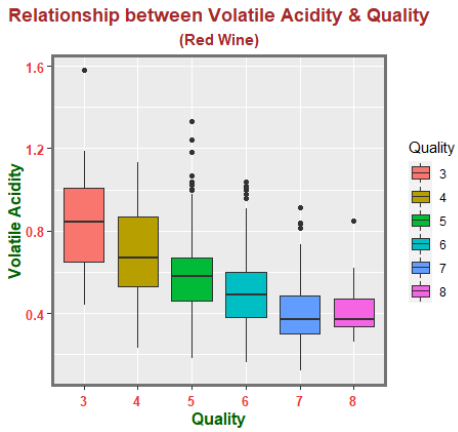

Fig 1.7: Volatile Acidity Vs Quality

Response Vs Predictors – White Wine:

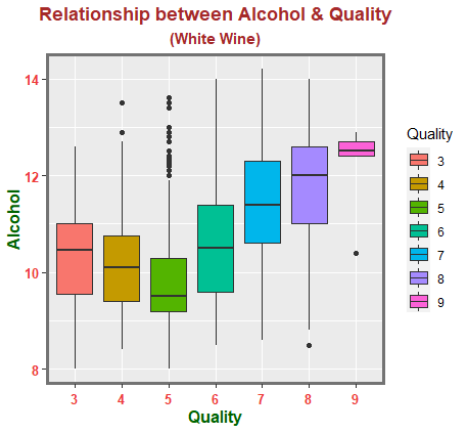

Fig 2.1 : Alcohol Vs Quality

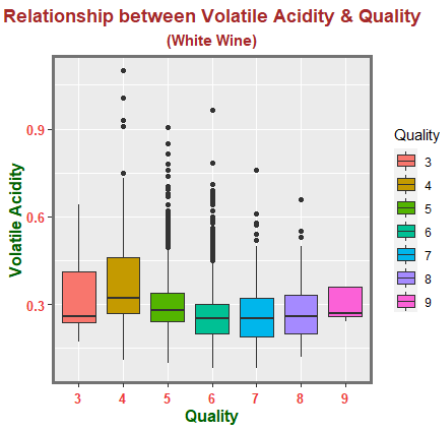

Fig 2.2: Volatile Acidity Vs Quality

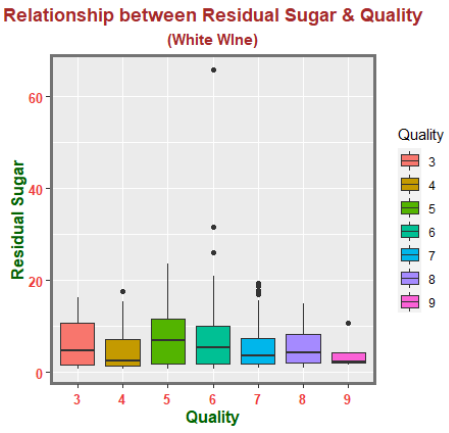

Fig 2.3: Residual Sugar Vs Quality

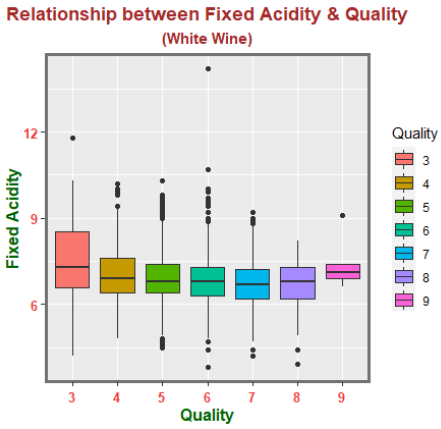

Fig 2.4: Fixed Acidity Vs Quality

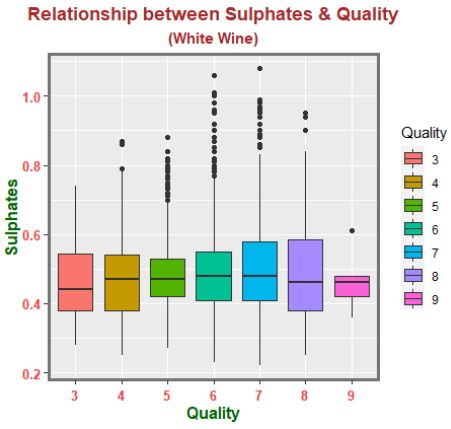

Fig 2.5: Sulphates Vs Quality

Fig 2.6: pH Vs Quality

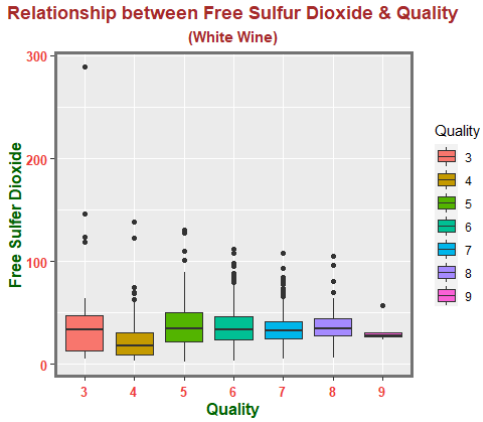

Fig 2.8: Free Sulfur Dioxide Vs Quality

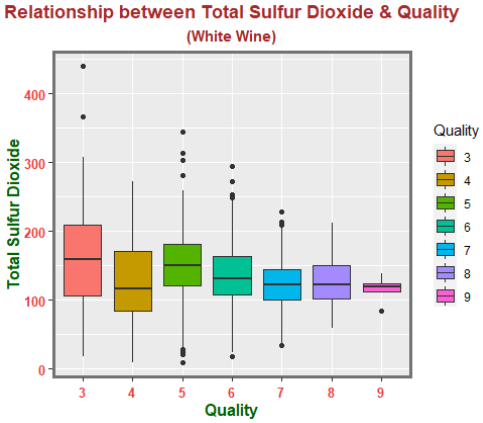

Fig 2.9: Total Sulfur Dioxide Vs Quality

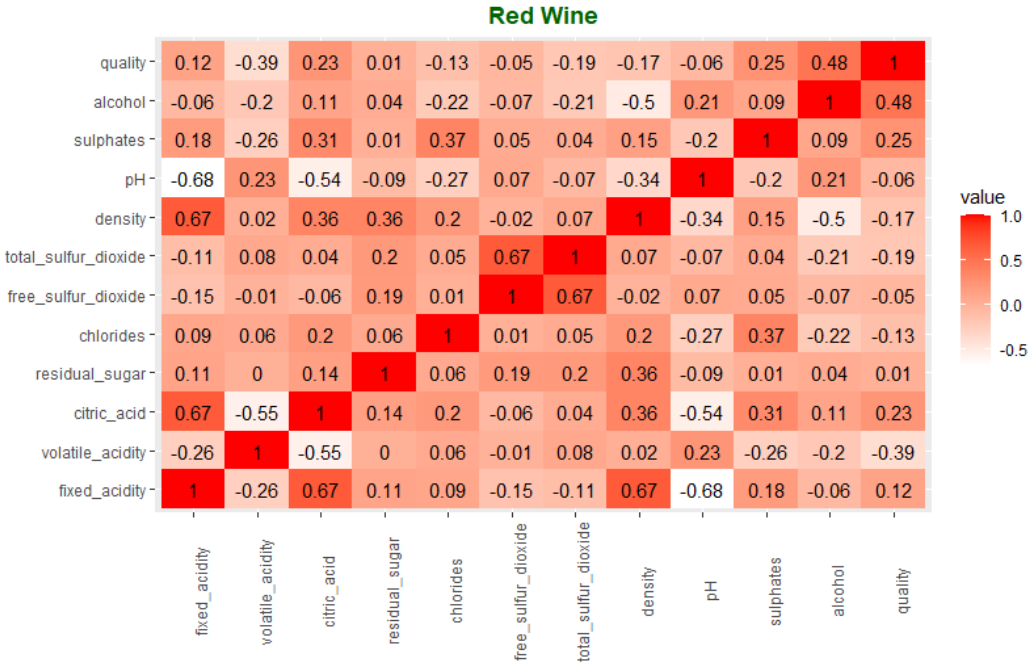

Fig 3.1: Correlation heat map of Red Wine

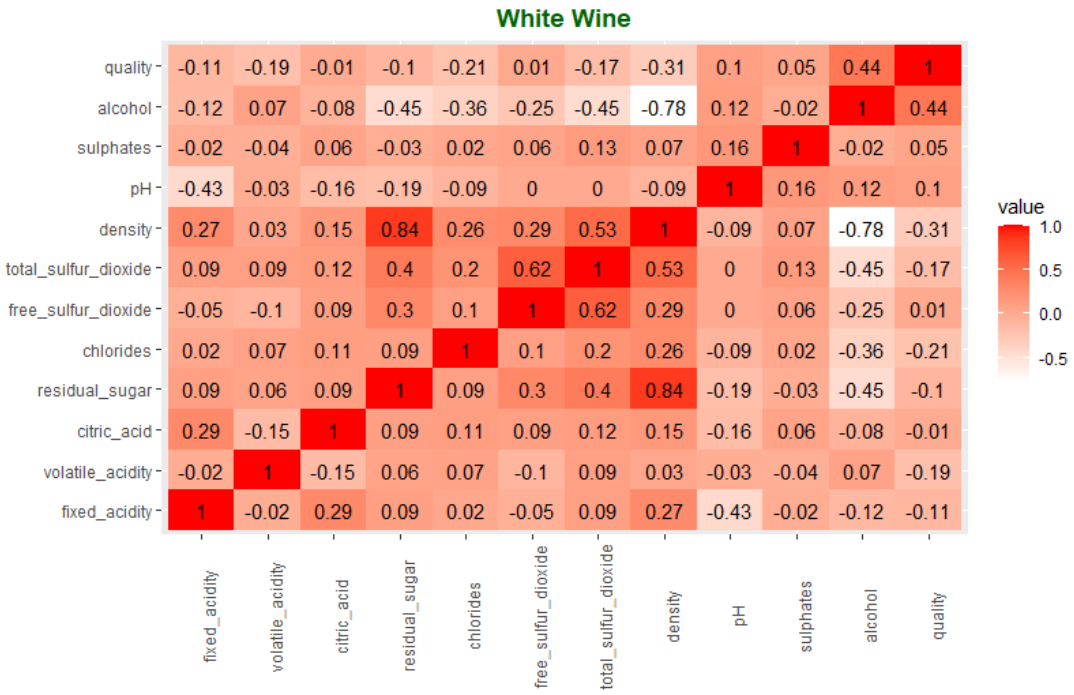

Fig 3.2: Correlation heat map of White wine

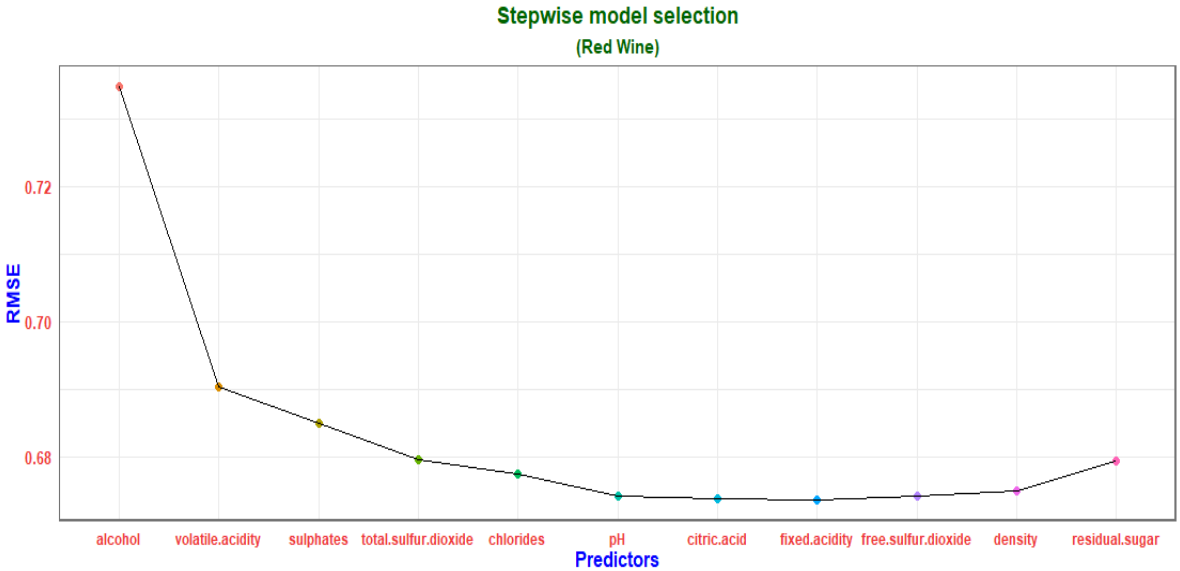

Fig 4: RMSE vs Predictors for Red Wine

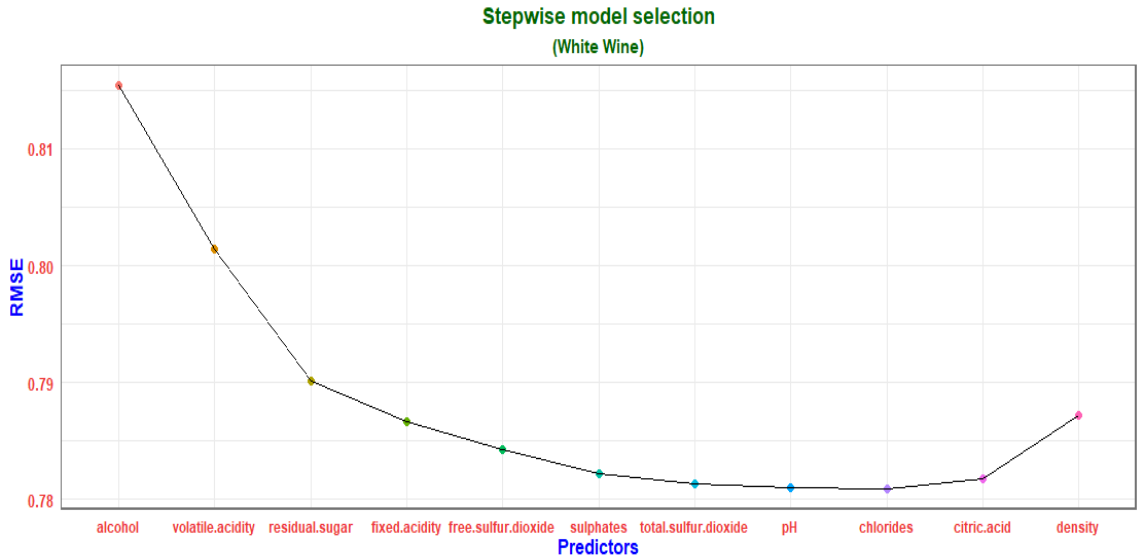

Fig 5: RMSE vs Predictors for White Wine

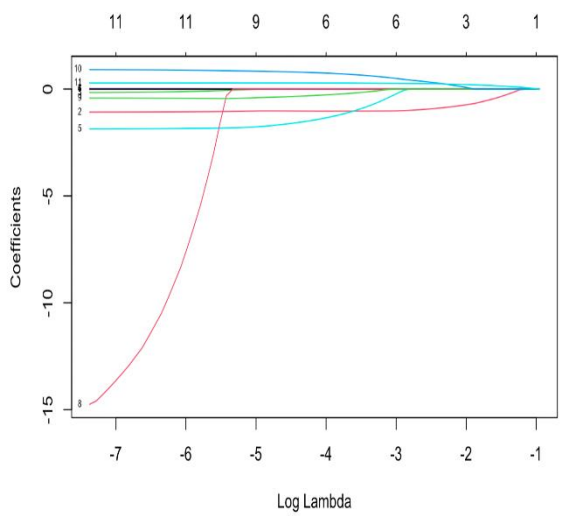

Fig 6: Red Wine- Coeff Vs Lambda

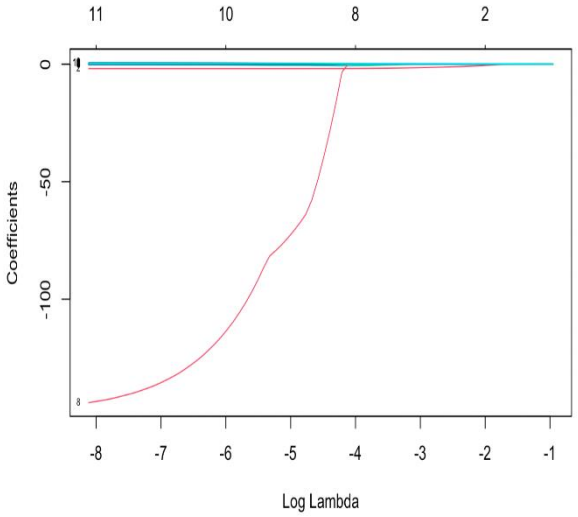

Fig 7: White Wine - Coeff vs Lambda

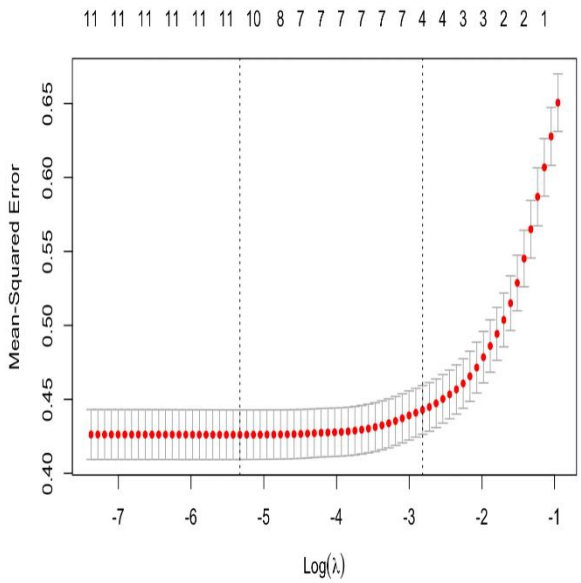

Fig 8: Red Wine - RMSE vs Lambda

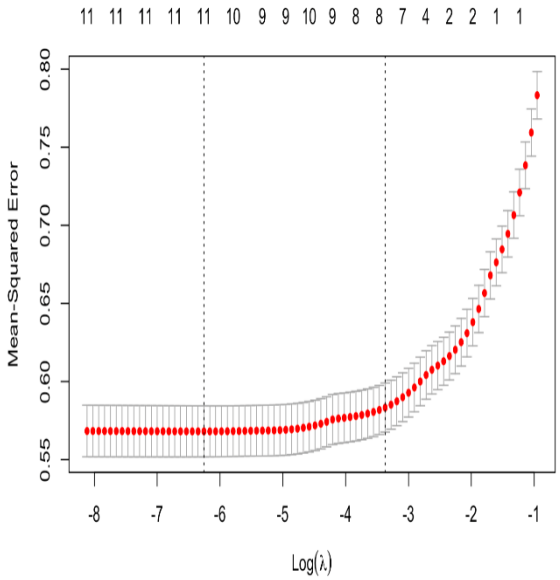

Fig 9: Red Wine- RMSE vs Lambda

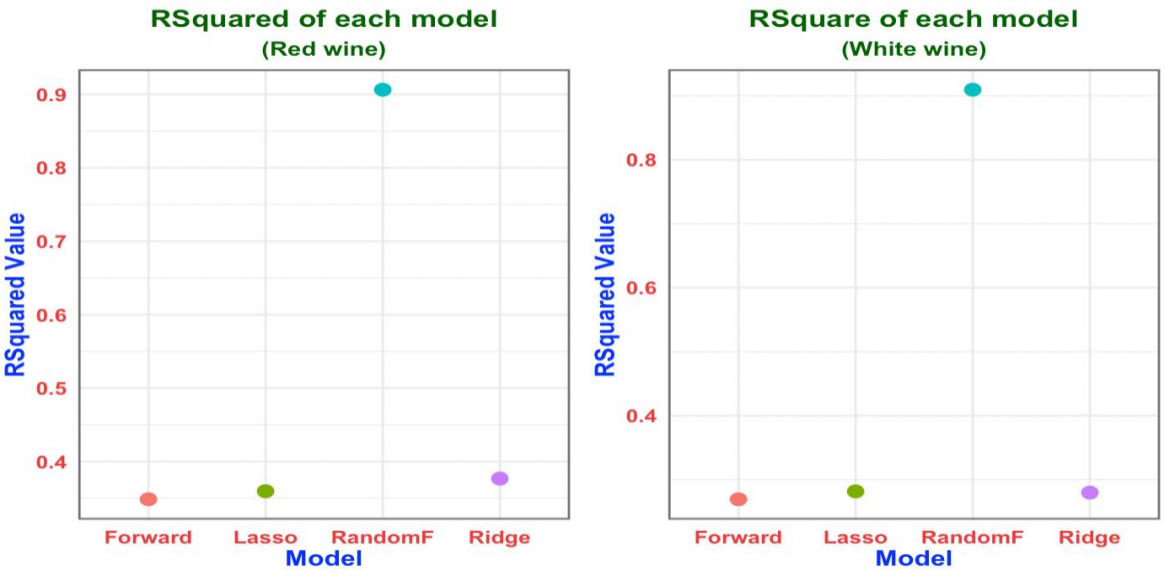

Fig 10.1: Integrated result for all models - Red and White wine

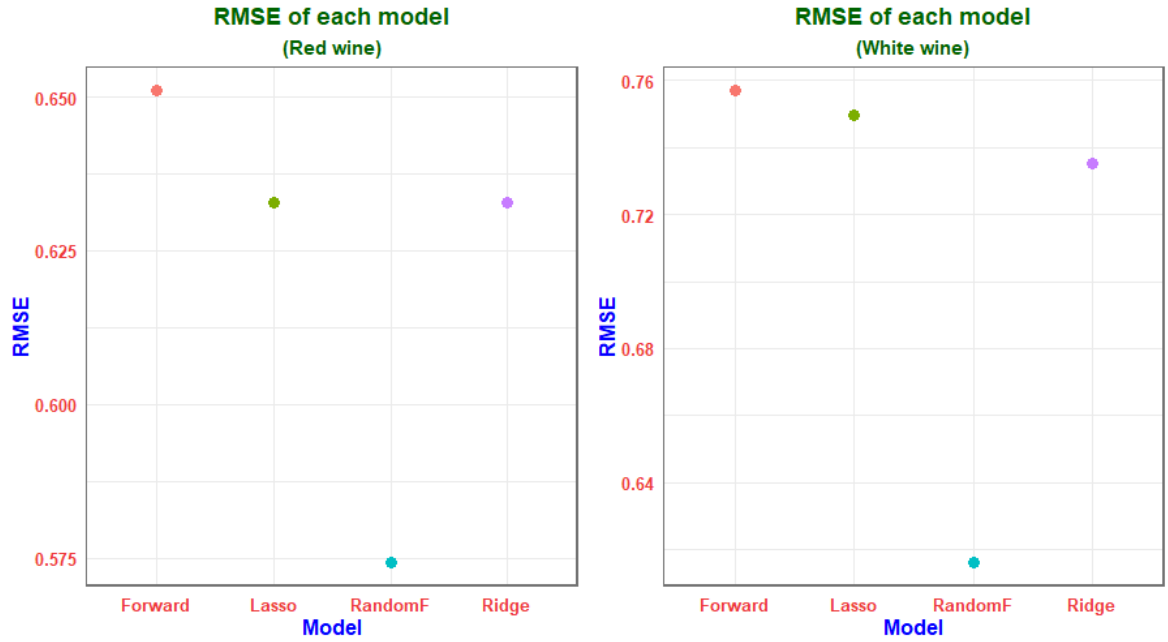

Figure 10.2: RMSE for each model-Red and White wine

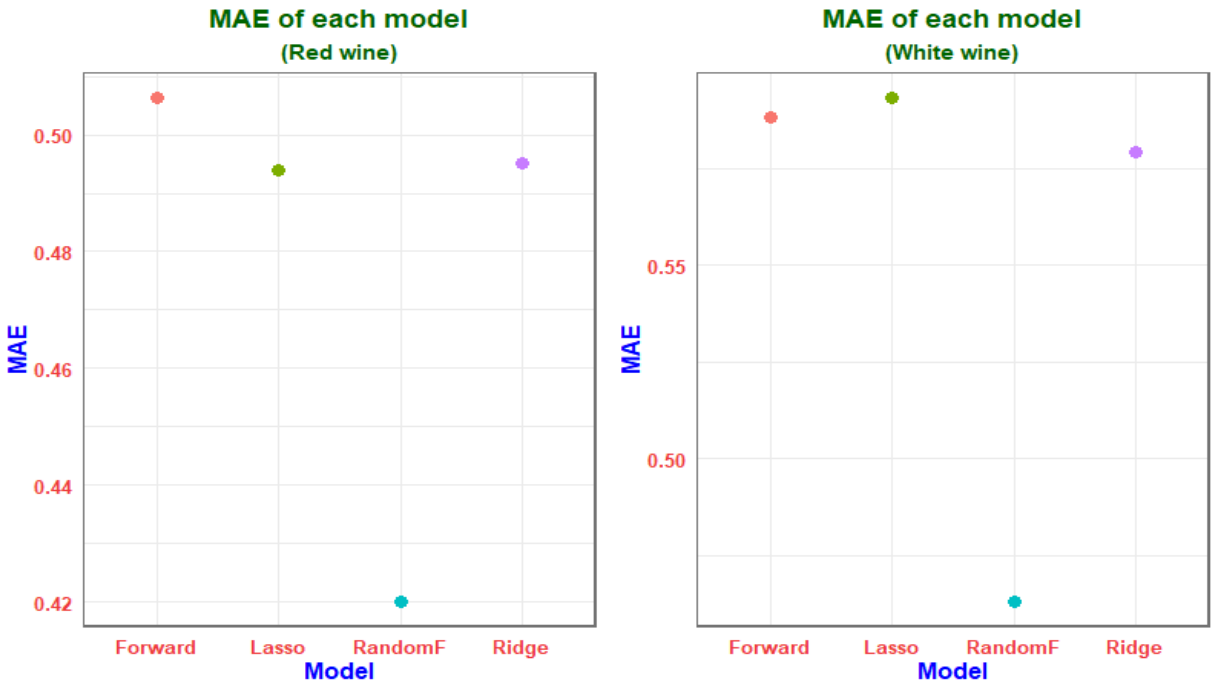

Fig 10.3: MAE for each model- Red and White Wine

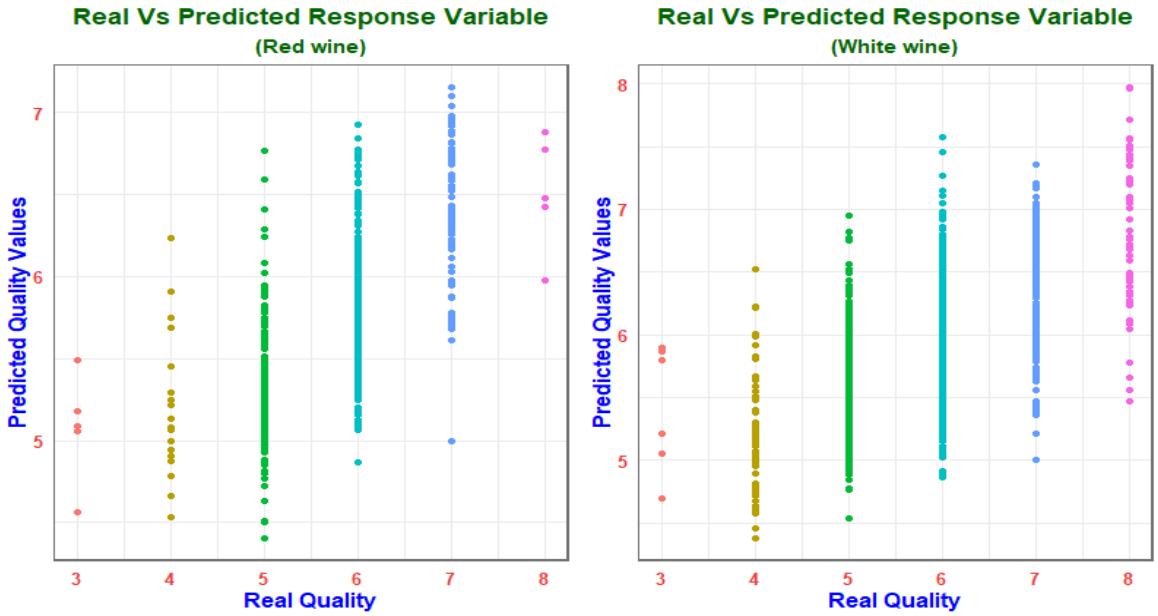

Fig 11: Real Vs Predicted values of the model

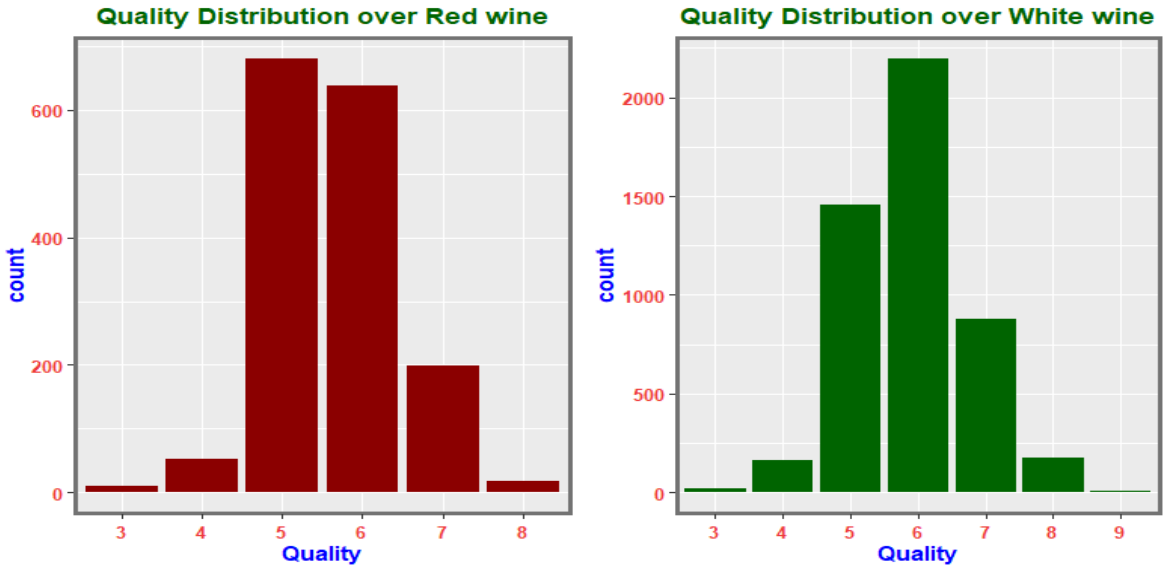

Fig 12: Quality Distribution graph- Red and White wine

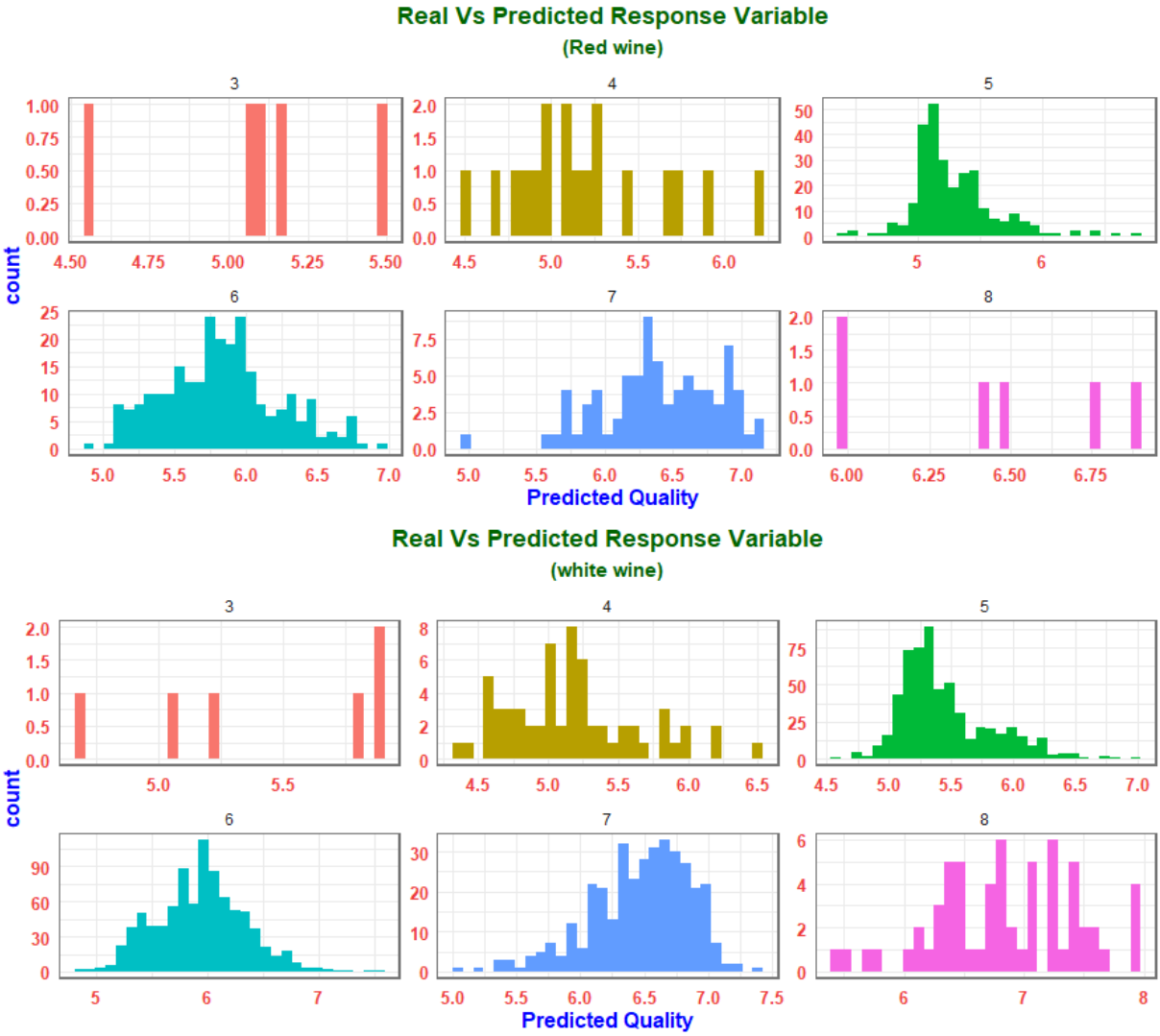

Fig 13: Real Vs Predicted response variable for Red and White wine