Wine Prediction

FRANCIS JINGO

May-10-2021

HarvardX: Data Science: Capstone

Introduction

Wine is a drink consumed at large scale by majority of the global population (https://wineinstitute.org). The makers must be able to ensure quality of their production, otherwise the demand for the particular brand will go down as well as the market share. Based on this, most wine manufacturers have resorted to discovering ways to improve the quality and this has led to most of them using machine learning. The aim is to determine the quality of the product based on the ingredients used during the manufacturing process.We are going to use two types of wine that is red wine and white wine.

The project can be accessed via this link on github (https://github.com/jingof/R_wine_quality_prediction.git).

The ingredients used in this notebook include: `['wine_type','fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol']` and these are considered the predictor features. The output feature is 'quality'.

Methods

In this section, I discuss about the methods used in this project.

Test accuracy

This is used to test the accuracy of our prediction, we look at the values that are exactly the same as in the model, and consider those a success, otherwise our prediction is false, this ensures out accuracy is as factual as possible.

```
test_accuracy <- function(pred,actual)
{
  val<-(actual - pred)
  len<-length(which(val==0))
  len/length(pred)
}
```

Add Elements

This is used to add elements from one list to another list where items in list2 are added to list1.

```
addElements <- function(list1,list2){
  len1 <- length(list1)
  i <- len1+1
  a<-1
  len2 <- length(list2)
  while(a <= len2){
    list1[i] <- list2[a]
    a <-a+1
    i<-i+1
  }
  list1
}
```

Find Outliers

This is used to determine which values in the dataset are considered outliers based on median and standard deviation where dt is the whole dataframe and col is the column with the outliers.

```
FindOutliers <- function(dt,col){
    sd <- as.numeric(sapply(dt[col],sd))
    med <- as.numeric(sapply(dt[col],median))
    lowerq <- med - (sd*3)
    upperq <- med + (sd*3)
    which(dt[col]>upperq | dt[col]<lowerq)
}
```

Data exploration and visualization

The datset is explored to have an understanding of some of its elements. Below are the top rows in the dataset.

```
head(wine_data)
```

```
## # A tibble: 6 x 13
##   wine_type fixed_acidity volatile_acidity citric_acid residual_sugar chlorides
##   <chr>             <dbl>            <dbl>       <dbl>          <dbl>     <dbl>
## 1 Red                 5.4             0.74        0              1.2       0.041
## 2 Red                 5               0.74        0              1.2       0.041
## 3 Red                 5.1             0.47        0.02           1.3       0.034
## 4 Red                 4.6             0.52        0.15           2.1       0.054
## 5 Red                 4.7             0.6         0.17           2.3       0.058
## 6 White               5.3             0.26        0.23           5.15      0.034
## # ... with 7 more variables: free_sulfurdioxide <dbl>,
## #   total_sulfurdioxide <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <dbl>
```

Next we look at the structure of the dataset since we have two wine types, it is better to dig into both of them to understand the dataset better. We start with the information to know how many null values we have as well as the data types.

```r
colSums(is.na(wine_data))
```

```
##           wine_type       fixed_acidity    volatile_acidity         citric_acid
##                   0                   0                   0                   0
##      residual_sugar           chlorides  free_sulfurdioxide total_sulfurdioxide
##                   0                   0                   0                   0
##             density                  pH           sulphates             alcohol
##                   0                   0                   0                   0
##             quality
##                   0
```

Next we look at the shapes of the entire wine dataset. We then look at the white wine data and the red wine data contained in the dataset.

```r
dim(wine_data)
```

```
## [1] 6497   13
```

```r
red_wine <- wine_data %>% filter(wine_type=='Red') %>% select(-wine_type)
dim(red_wine)
```

```
## [1] 1599   12
```

```r
white_wine <- wine_data %>% filter(wine_type=='White') %>% select(-wine_type)
dim(white_wine)
```

```
## [1] 4898   12
```

Looking at the quality summary for the datasets.

```r
summary(wine_data$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.818   6.000   9.000
```

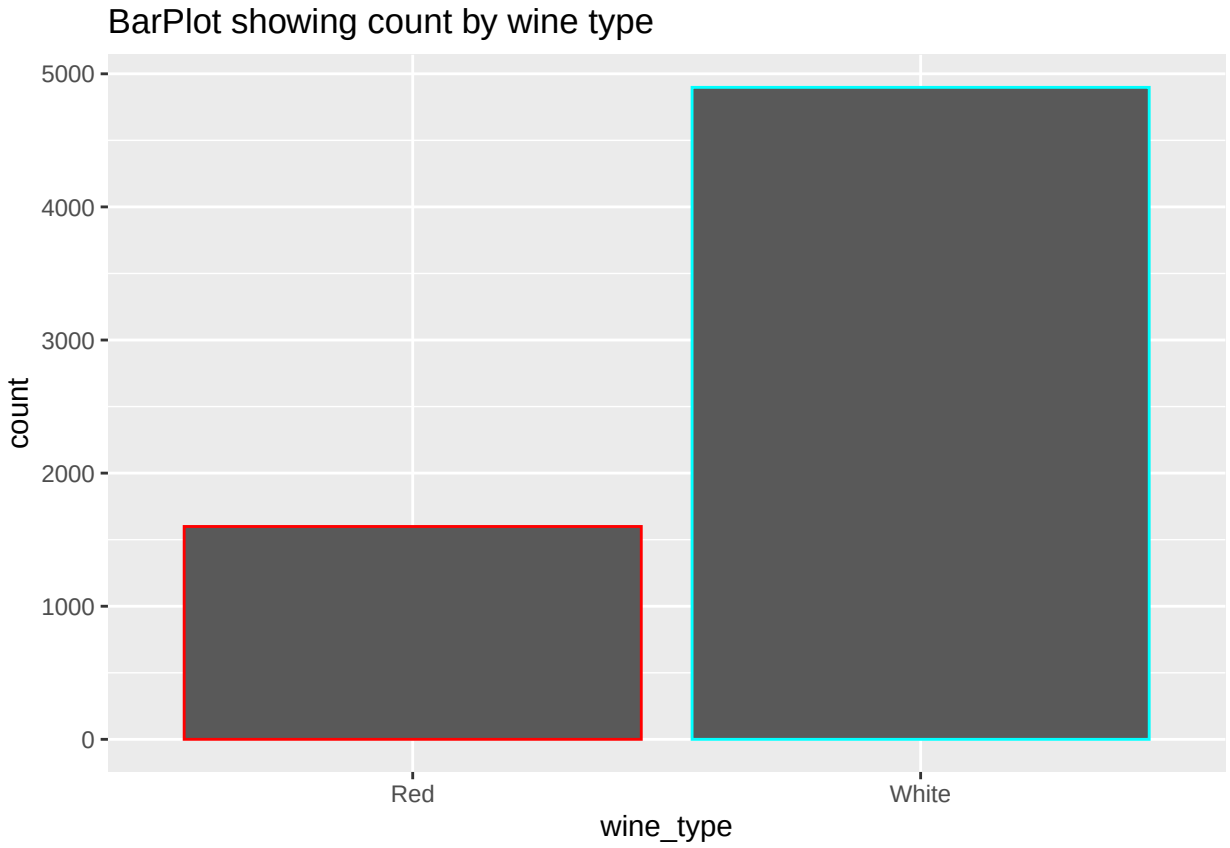
```r
summary(red_wine$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.636   6.000   8.000
```

```r
summary(white_wine$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.878   6.000   9.000
```

Plotting the distribution count of the wine dataset to see the counts between the white wine and the red wine.



From the above, we can see that the white wine dataset has more rows meaning it has more observations. We also see that the data has no null values and has 12 predictor columns and one dependent column. Further, we can tell that all predictor columns are numerical except for wine type, to make this numerical, we shall have to create two categorical features having 1 or 0, one for red wine, another for white wine. We can also see that no column is categorical except for the two we create below.

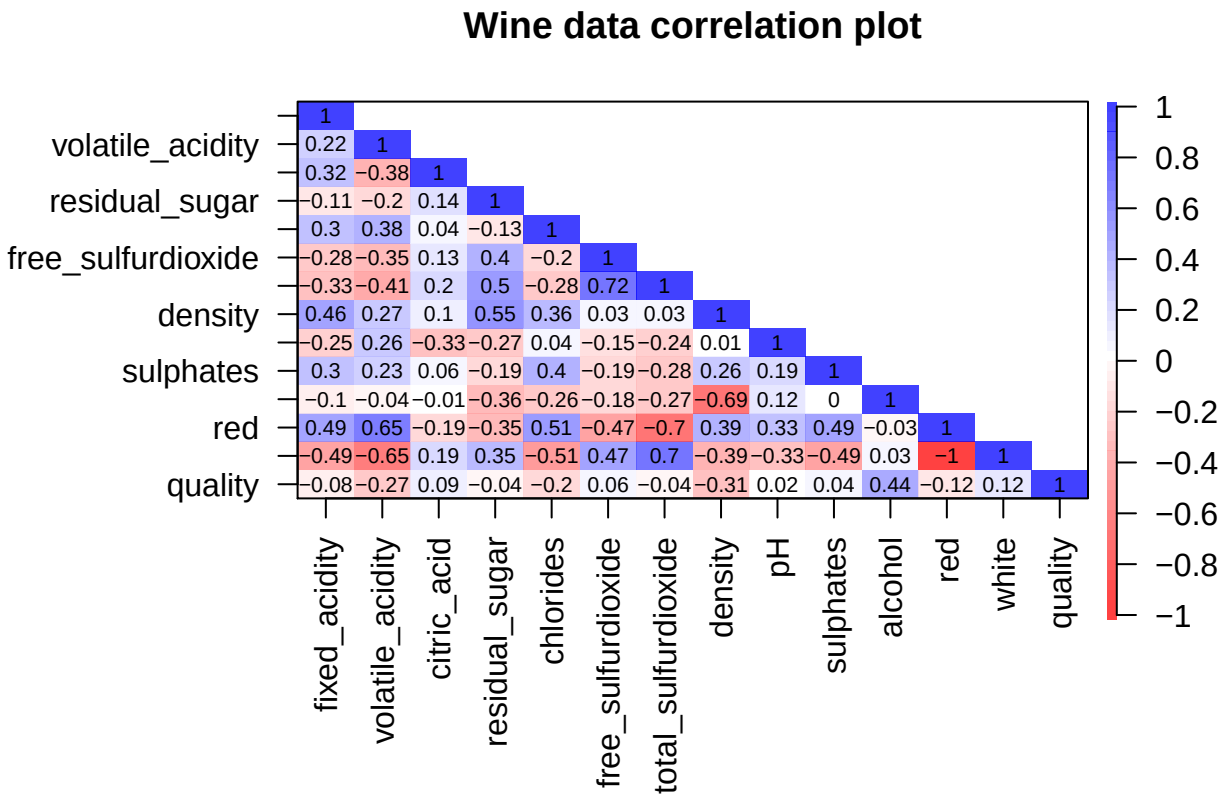
Next, we look at the dependent column to determine the nature of the values in the column. From the below, we can tell that it is an integer rank ranging from 0 of 10.

```
unique(wine_data$quality)
```

```
## [1] 6 4 7 5 8 3 9
```

We now look at correlation, we could print out the `correlation matrix` of the columns however i think using a `heatmap` might be prefferable since it is easier to spot very highly correlated and very lowly correlated columns than on the matrix, lets have a look.

```
corPlot(wine_data,scale=FALSE,xlas=2,upper=FALSE,main='Wine data correlation plot')
```

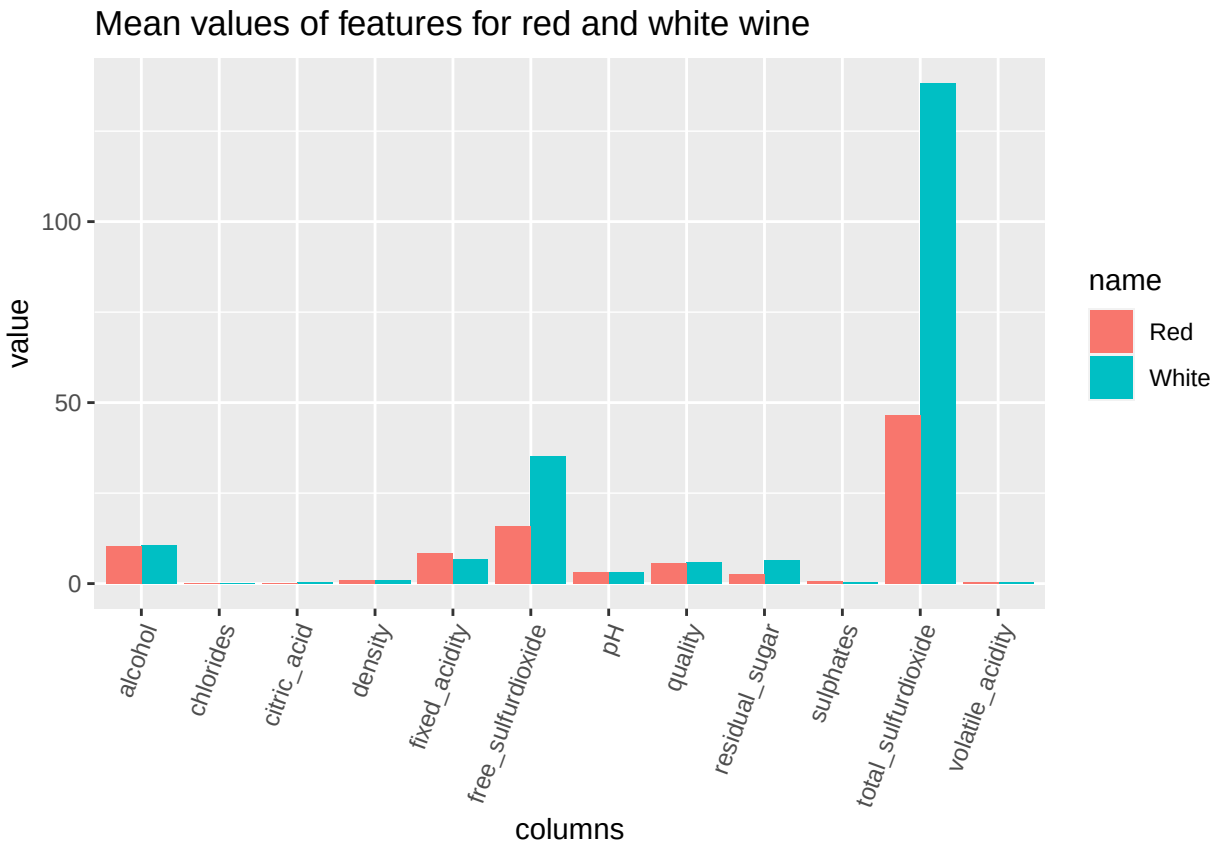


We then calculate the average of each feature for the red and white wines separately using mean() function and plot bar graph to show comparison. This will help us understand the basic average between the quantities of ingredients used for the two different wine datasets. From below we can see that most features have an almost similar mean 2 for the two datasets except for free sulfurdioxide and total sulfur dioxide which have a significant difference between the two means. We can also plot these to have a visualization.

```
red_means <- red_wine %>% colMeans()
white_means <- white_wine %>% colMeans()
wine_data2 <- wine_data %>% select(-red,-white)

wine_means <- structure(list(columns = c(colnames(wine_data2)),
                White = c(white_means),Red = c(red_means)),
                 class = "data.frame",
                 row.names = c('1','2','3','4','5','6','7','8','9','10','11', '12'))

wine_means %>%
  pivot_longer(cols = -columns) %>%
  ggplot(aes(x = columns, y = value, fill = name)) +
  geom_col(position = 'dodge')+
  theme(axis.text.x = element_text(angle = 70, hjust = 1))+
  ggtitle('Mean values of features for red and white wine')
```

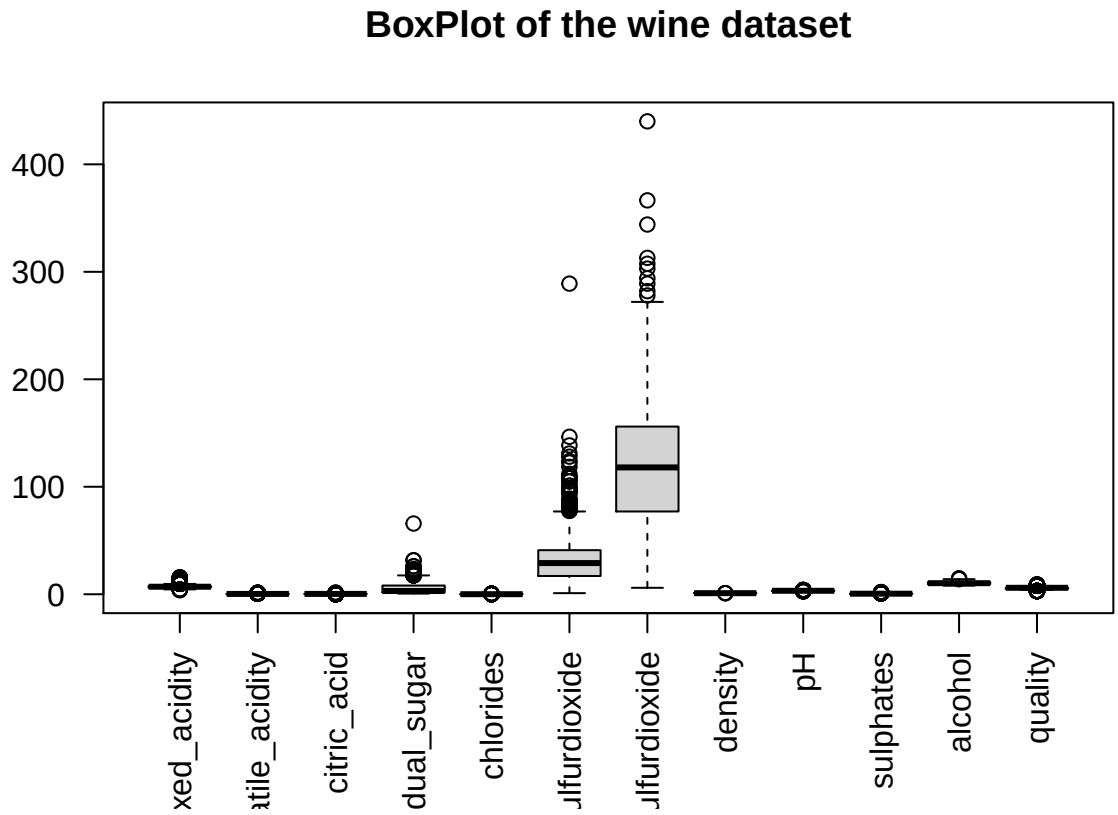


Graph Description

White wine generally has a higher mean for most of the values meaning that it requires more ingredients and therefore is more expensive to make. However, the cost does not necessarily lead to better quality since the mean value for the quality is the same.

Outliers

These are biased observations that may make our data analysis inaccurate. They are 3 standard deviations away from the median. We can perform a box plot to check if there are any outliers.



We can also have the outliers count in a data frame to correctly see how many outliers each column has. From the below we can see that 8 columns have less than 100 rows with outliers.

```
##                columns outliers
## 1        fixed_acidity      148
## 2     volatile_acidity      128
## 3          citric_acid       29
## 4       residual_sugar      145
## 5            chlorides      122
## 6   free_sulfurdioxide       41
## 7  total_sulfurdioxide        8
## 8              density        3
## 9                   pH       35
## 10           sulphates       88
## 11             alcohol       18
## 12             quality       35
```

Looking at the `outlier_index` to determine how many unique values are there to determine how many rows will actually fall under outliers.

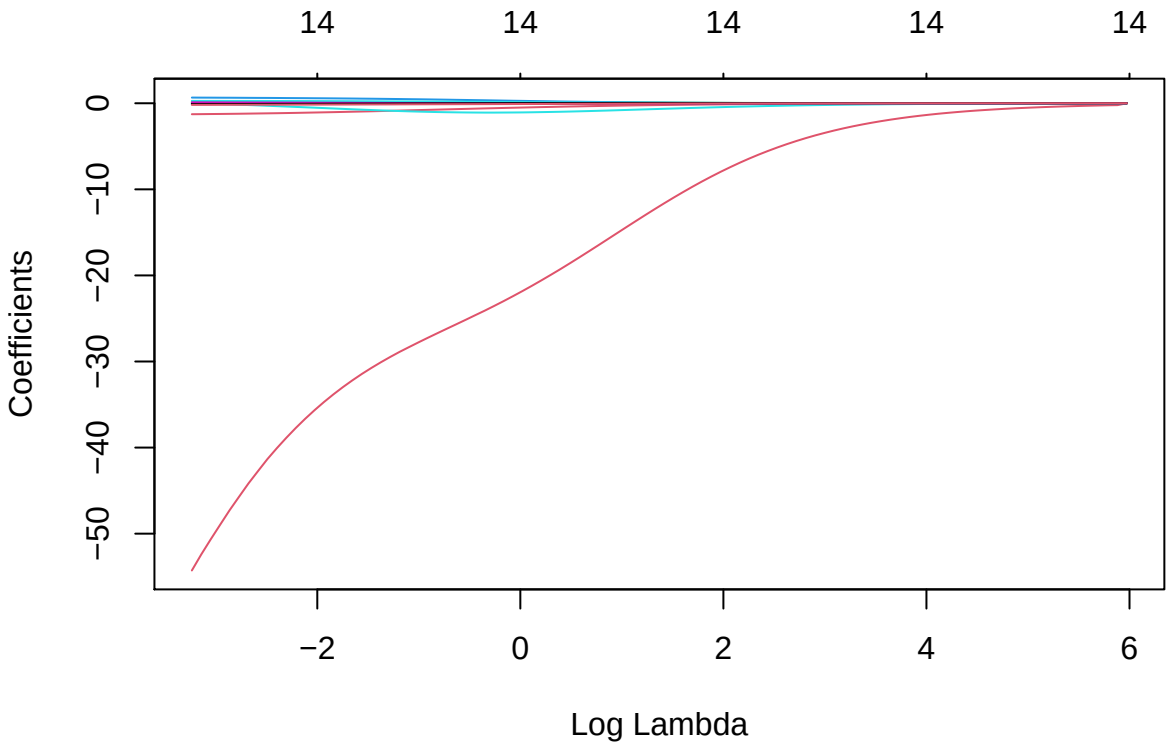
We have `714 rows` as outliers, this is `10.99%` percent of the data, which is a huge number Instead of dropping these rows, we can have it as a categorical feature.

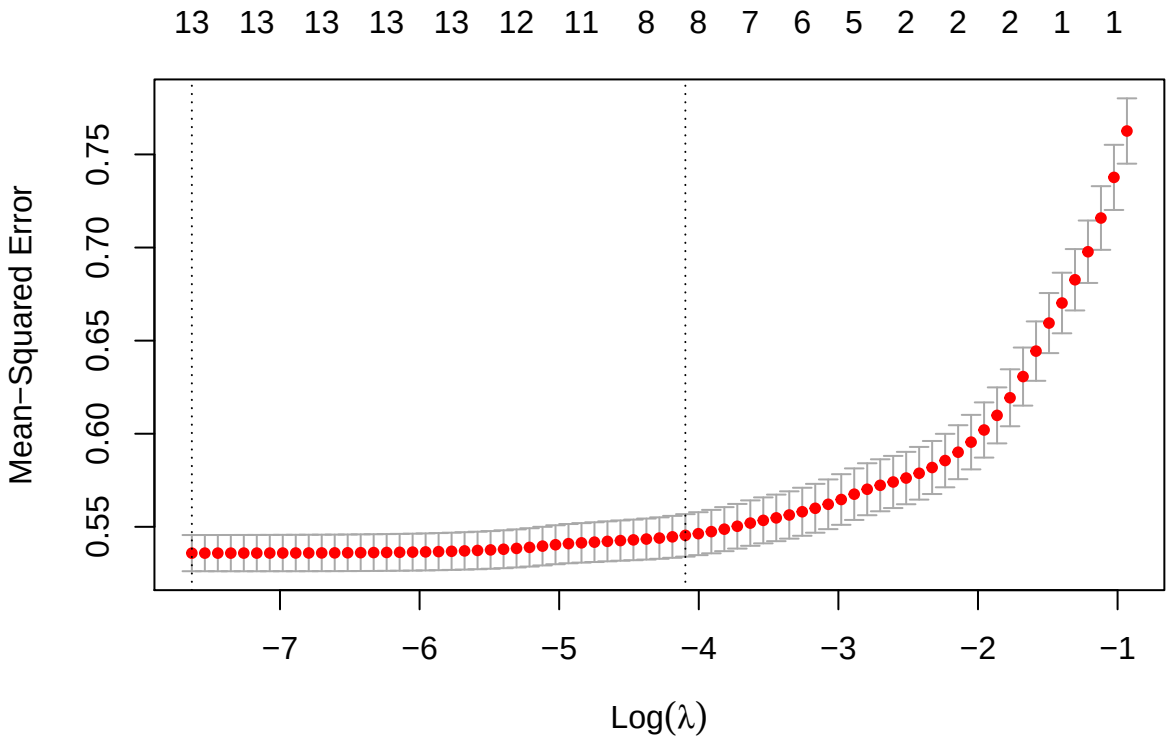
Data train test split.

We split the dataset into train and test datasets into portions of `66%` to `33%` respectively. The train dataset has `4288 rows` and the test dataset has `2209 rows`.

Feature Importance and Selection.

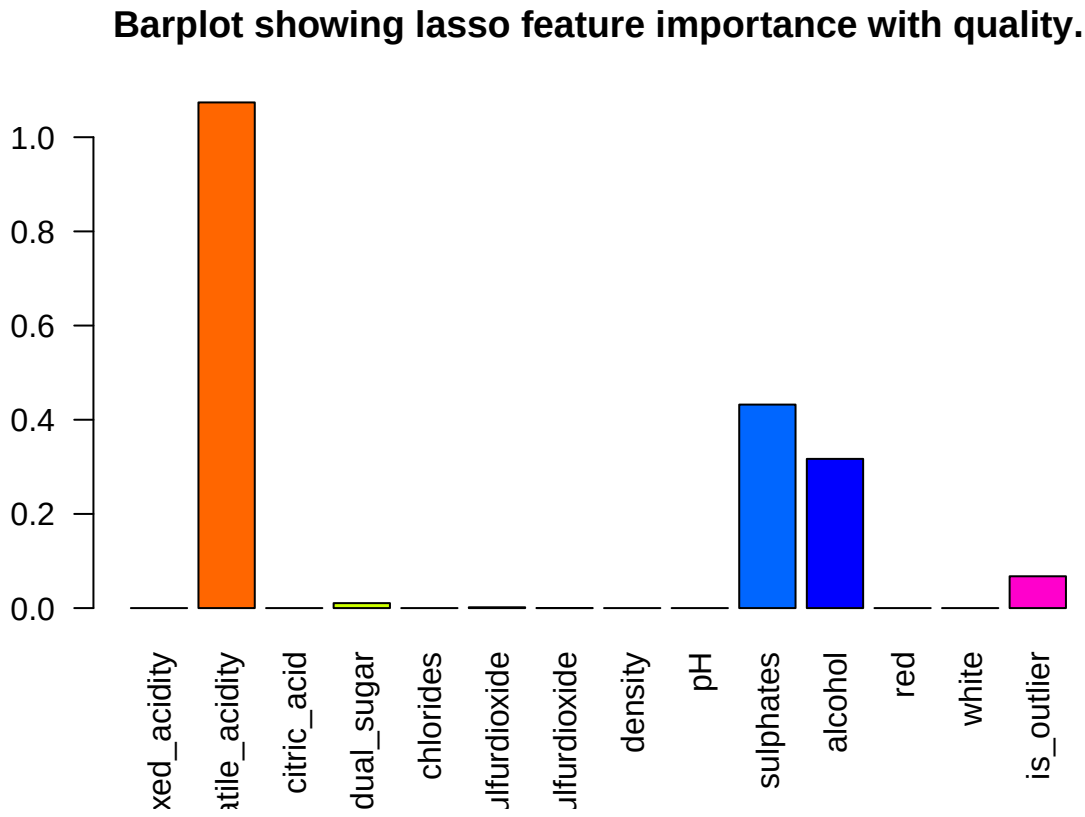
I opted to use Lasso to study the features, their importance to the dependent variable and for feature selection. I use Lasso and cross-validation to provide a plot of MSE for the wine data. An explanation on how LASSO selects features is also given below.

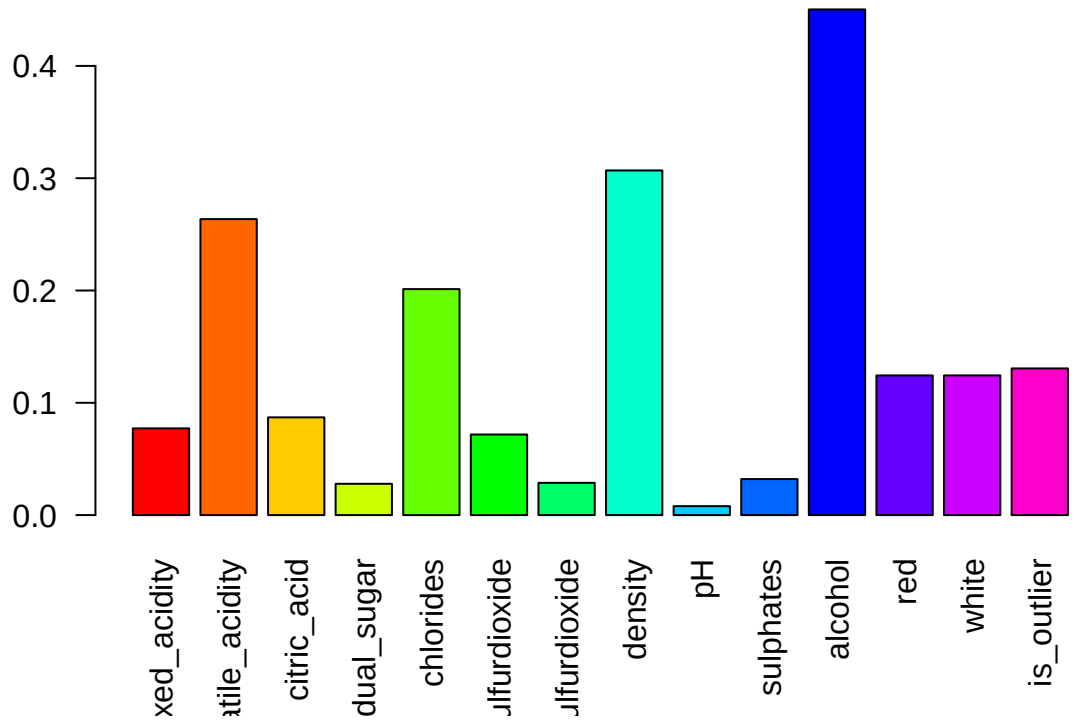




For feature selection, I use Lasso to get features with the hisghest Importance and compare this with theie respective correlation with the dependent variable(`quality`).

```
## [1] "volatile_acidity"    "residual_sugar"      "free_sulfurdioxide"
## [4] "total_sulfurdioxide" "sulphates"           "alcohol"
## [7] "is_outlier"
```


Barplot showing lasso feature importance with quality.
1.0
0.8
0.6
0.4
0.2
0.0
xed_acidity
atile_acidity
citric_acid
dual_sugar
chlorides
lfurdioxide
lfurdioxide
density
pH
sulphates
alcohol
red
white
is_outlier




```
##              Features        lasso correlation
## 1       fixed_acidity 0.0000000000 0.077221205
## 2    volatile_acidity 1.0738333956 0.263584911
## 3         citric_acid 0.0000000000 0.087024532
## 4      residual_sugar 0.0105500017 0.027847213
## 5           chlorides 0.0000000000 0.201219355
## 6  free_sulfurdioxide 0.0016959099 0.071744696
## 7 total_sulfurdioxide 0.0001060559 0.028720545
## 8             density 0.0000000000 0.306893726
## 9                  pH 0.0000000000 0.007955032
## 10          sulphates 0.4321362069 0.032083357
## 11            alcohol 0.3168533984 0.450283207
## 12                red 0.0000000000 0.124366528
## 13              white 0.0000000000 0.124366528
## 14         is_outlier 0.0675653375 0.130603142
```

Lasso Explanation.

Using Linear Regression with L1 regularization is called Lasso Regularization. Given a dataset with features and dependent variables We have different features or variables in our data which we denote by x1, x2, . . . , xn.

The learnable or trainable parameters in our models are A_0, A_1, . . . , A_n. After training some values will be assigned to A parameters. If we observe the solution the larger the values of A the larger effect it will have on the solution and vice versa. Thus, we can decide upon some threshold value and we can keep all

those variables for which the corresponding A values are larger than the threshold value and discard the others. The larger the threshold values the smaller the number of parameters we will keep and vice versa. The selected features can then be used to train the models. Of course, there are other things as well to improve the dataset like normalizing the values and handling the missing data before proceeding to train the model and there are other ways of feature selection. But I liked this idea of using regularization for feature selection. This is a nice way of reducing dimensionality by removing not so important features.

LASSO.

The advantages.

- As any regularization method, it can avoid overfitting. It can be applied even when number of features is larger than number of data.
- It can do feature selection.
- It is fast in terms of inference and fitting.

The disadvantages.

- It ignores nonsignificant variables that may, nevertheless, be interesting or important.
- It doesn't follow the hierarchy principle.
- The model selected by lasso is not always stable.
- When there are highly correlated features, lasso may randomly select one of them of part of them.
- It is an automatic model selection technique and is susceptible to error.

Because of the above facts, I decided to use the variables from the Lasso Regressor. We can go on to compare the Lasso regressor to other regressor models, we are going to use only the features selected by Lasso, to compare the Lasso regressor to:-

- KNN regressor
- Decision Tree
- Random Forest

```
##          MODELS ACCURACYS
## 1         lasso 0.5201449
## 2 random forest 0.6468990
## 3        knn cv 0.4477139
## 4 decision tree 0.5273880
```

Conclusion

From the above, we can see that the overall best model that gives the best accuracy is is the Random Forest. Therefore in conclusion, to be able to get the best prediction for the wine quality based on the quantity of ingredients, we can use the Lasso regularization model to perform the feature selection after which we use the Random Forest to predict the quality, we can also always improve our model by turning the parameters for the Random Forest differently.