Predicting the Quality Categories of Red Wines

1. Introduction

The goal is to predict quality of wine based on chemical measurements.

In this project we will be working with a dataset of 1,599 observations for Portuguese red wines of "Vinho Verde" type, where Vinho Verde refers to the young age of the wine, produced 3 to 6 months after the grapes are harvested in the Minho province of Northern Portugal.

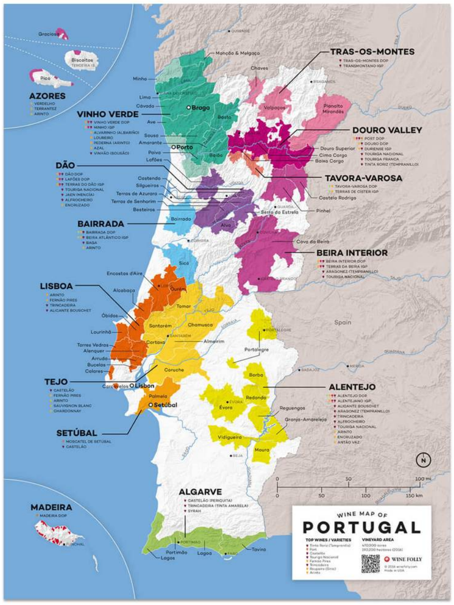


Figure 1: Map of Portugal that shows Vinho Verde Province

The dataset contains twelve variables - eleven for different chemical characteristics, such as pH level and alcohol percentage and the last twelfth variable is the quality of the wine. It will be the outcome variable which we are going to predict based on the chemical characteristics. This dataset is available at the UCI machine learning repository of https://archive.ics.uci.edu/ml/datasets/wine+quality and at the kaggle website: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/tasks?taskId=1738.

The eleven variables measuring chemical properties of the wine are the following:

1. Fixed Acidity: measure of non-volatile acids that do not evaporate readily

2. Volatile Acidity: measure of high acetic acid in wine which leads to an unpleasant vinegar taste
3. Citric Acid: amount of citric acid which acts as a preservative to increase acidity.
4. Residual Sugar: amount of sugar remaining after fermentation
5. Chlorides: the amount of salt in the wine
6. Free Sulfur Dioxide SO2: measure of substance which prevents microbial growth and the oxidation of wine
7. Total Sulfur Dioxide: amount of free SO2 + bound forms of SO2
8. Density: wine density (sweeter wines have a higher density)
9. pH: level of acidity on a scale of 0–14
10. Alcohol
11. Sulphates: wine additives which act as anti-microbials and antioxidants

The standard red wine composition is shown on this pie chart:

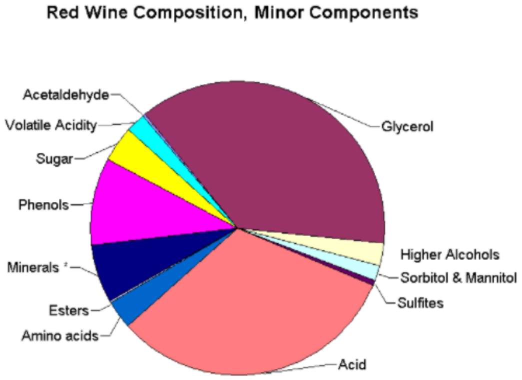


Figure 2: Composition of Red Wine

The twelfth variable in the dataset is the quality of the wine. It is rated on a 0-10 scale. To assign the scale value each wine sample is evaluated by a minimum of three tasters using blind tasting, and then the wine is graded on the 0-10 scale using the median of the scores of the tasters.

The components of wine are measured by a wine analyzer. A wine analyzer is a standalone device that measures characteristics of fifteen or so variables.

Figure 3: Wine Analyzer

The units of measurements are the following: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, sulphates and density are measured in g/dm^3, free sulfur dioxide and total sulfur dioxide are measured in mg/dm^3, alcohol in percent by volume, pH is the measurement of the hydrogen ion concentration pH=log(aH+) and has no units.

The goal of the project is to train some models for predicting how good different wines are and see how they perform. Multiples models will be compared to see how well they do in terms of accuracy and root mean squared error (RMSE).

Section 1: Data Overview

The following histograms show the distribution of all the variables we are initially given:

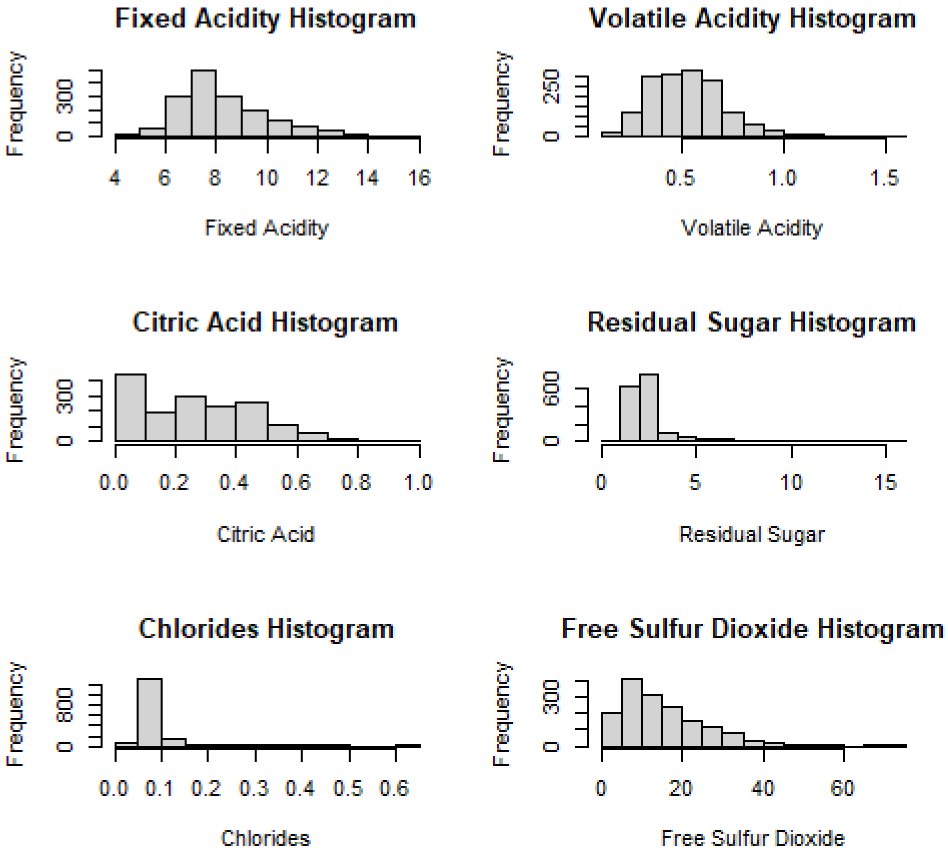


Figure 4: Histograms of the first six variables

Fixed Acidity and Free Sulfur Dioxide have a fairly typical right-skew. Volatile acidity is centered around 0.5 with very few values of <0.3 or >0.7. In the citric acid histogram we can see that the category of 0-0.1 is the most populous and then the overall trend is that the amount of wines decreases as the level of citric acid increases. The overwhelming majority of wines have residual sugar levels between 1 and 3 g/dm^3. The overwhelming majority of wines have chlorides levels between 0.05 and 0.1.

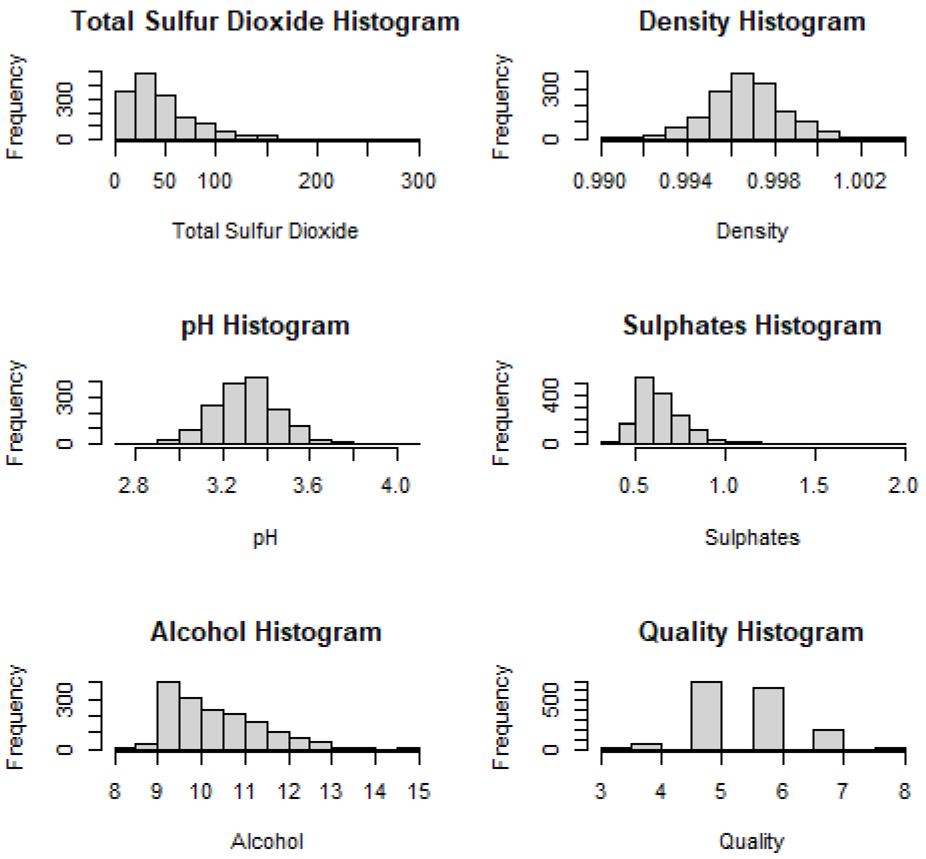
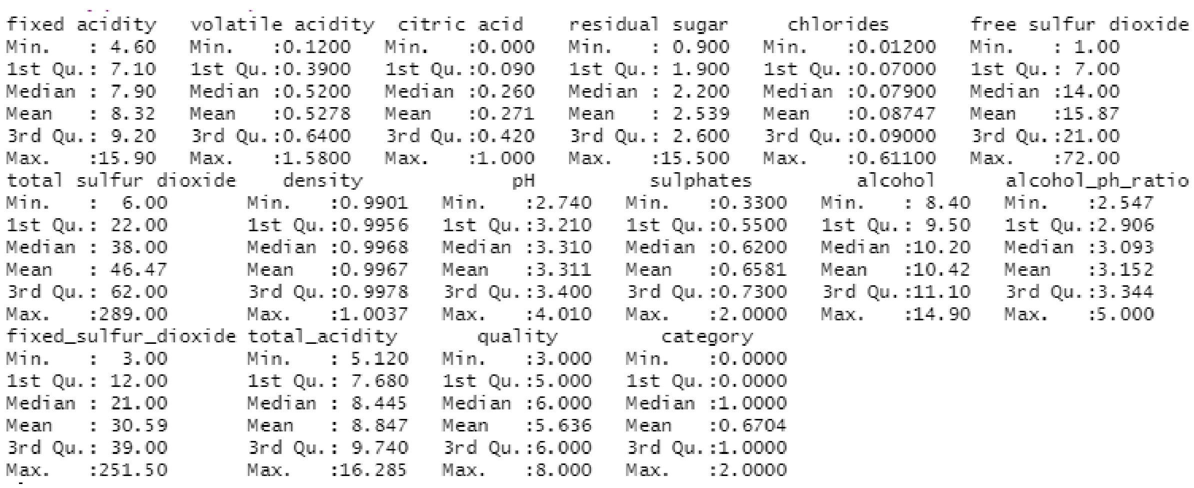


Figure 5: Histograms of the last six variables

Total Sulfur Dioxide, Sulphates and Alcohol are right-skewed in this set. Density, pH and quality are normally distributed. Between 9% and 10% alcohol is the most popular category with the amount of wines steadily declining as the amount of alcohol increases, after the 9% - 10% interval. For quality the overwhelming majority if wines fall into the middle categories with very few in the lower third and very few in the upper third. Most wines have a density of about 0.996.

The below table shows the means and other basic measures about the spread of each of the variables:



```
 fixed acidity   volatile acidity  citric acid    residual sugar     chlorides       free sulfur dioxide
 Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900   Min.   :0.01200   Min.   : 1.00
 1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900   1st Qu.:0.07000   1st Qu.: 7.00
 Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200   Median :0.07900   Median :14.00
 Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539   Mean   :0.08747   Mean   :15.87
 3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600   3rd Qu.:0.09000   3rd Qu.:21.00
 Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500   Max.   :0.61100   Max.   :72.00
 total sulfur dioxide    density             pH          sulphates         alcohol      alcohol_ph_ratio
 Min.   :  6.00       Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :2.547
 1st Qu.: 22.00       1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:2.906
 Median : 38.00       Median :0.9968   Median :3.310   Median :0.6200   Median :10.20   Median :3.093
 Mean   : 46.47       Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :3.152
 3rd Qu.: 62.00       3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:3.344
 Max.   :289.00       Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :5.000
 fixed_sulfur_dioxide total_acidity       quality         category
 Min.   :  3.00       Min.   : 5.120   Min.   :3.000   Min.   :0.0000
 1st Qu.: 12.00       1st Qu.: 7.680   1st Qu.:5.000   1st Qu.:0.0000
 Median : 21.00       Median : 8.445   Median :6.000   Median :1.0000
 Mean   : 30.59       Mean   : 8.847   Mean   :5.636   Mean   :0.6704
 3rd Qu.: 39.00       3rd Qu.: 9.740   3rd Qu.:6.000   3rd Qu.:1.0000
 Max.   :251.50       Max.   :16.285   Max.   :8.000   Max.   :2.0000
```

Figure 6: Summary of all variables

We modify this initial dataset by adding four variables derived from the initial variables:

alcohol_ph_ratio = alcohol/pH

fixed_sulfur_dioxide = total sulfur dioxide - free sulfur dioxide

total_acidity = fixed acidity + volatile acidity

The former three are recommended to use by some wine experts.

We also modify the outcome variable into a categorical variable "category", having values of "bad", "medium" and "good" in regards to the quality of wine.

The variable "category" groups the wine qualities into categories with the high quality category consisting of ratings 7 and 8 (no wines in this dataset have ratings 9 or 10), the "bad" category consists of qualities 5, 4 and 3 (no wines in this dataset have a rating of 2 or less) and the medium category consists of rating 6. The numeric values corresponding to these categories are 0 for bad, 1 for medium and 2 for good. This categorization is natural as the scale has very few really bad wines with the rating under 5, and very few excellent wines with the rating 8, as will be seen later. There are only 10 wines of quality 3 and 53 wines of quality 4. In the meanwhile here is the distribution of the three newly created categories:

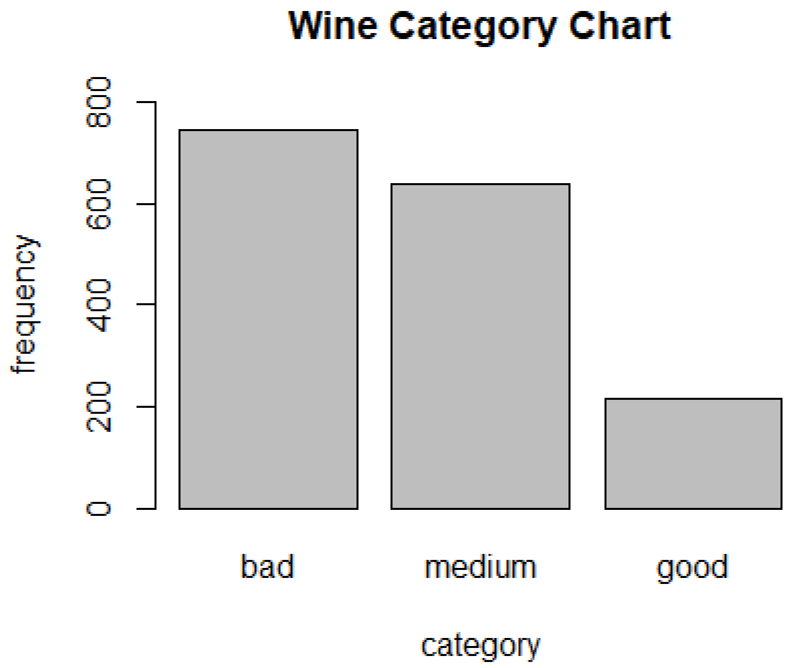


Figure 7: Wine Category Bar Chart

By the above-mentioned categorization good wines constitute a small percentage of all of them, 13.6%. The bad category is the most populous containing 46.5% with the medium category having 39.9%.

Figure 8 graphically displays a correlation matrix where blue color is used for positive correlations and red color for negative correlations. Circle size and color intensity are proportional to the size of the correlation coefficients.

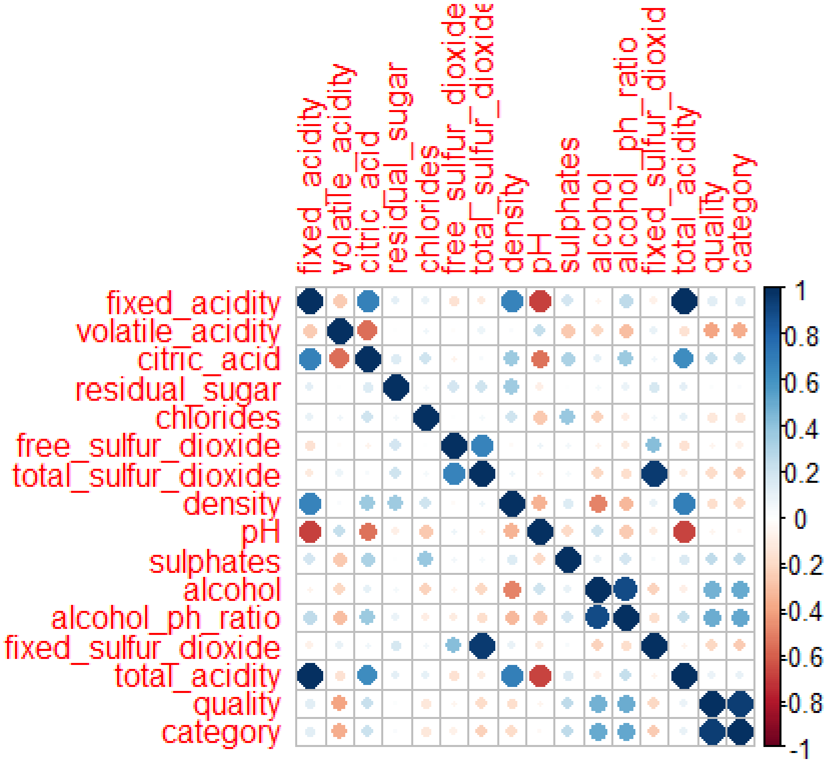
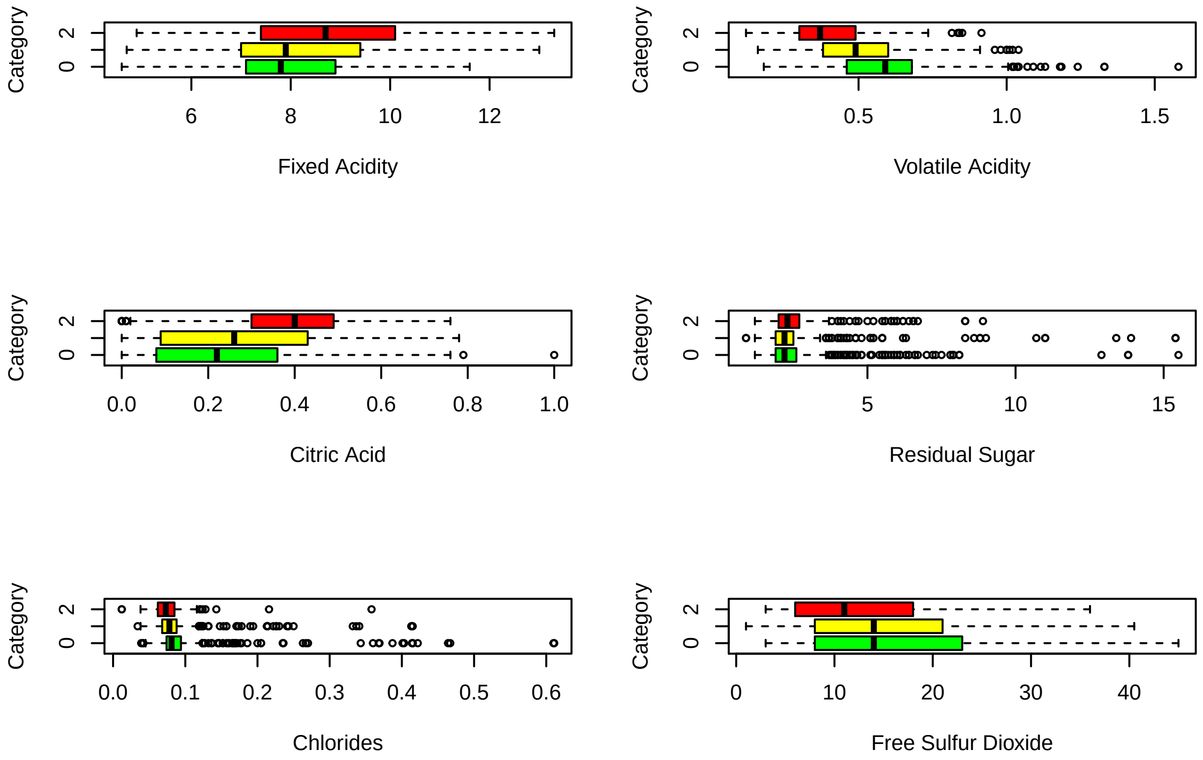


Figure 8: Correloplot

We see that the level of alcohol correlates with quality more than any of the originally provided variables. Free sulfur dioxide and total sulfur dioxides are highly correlated with each other, and fixed acidity and volatile acidity have a very strong correlation with one another. Strong negative correlation exists between pH and many of the acidities, not surprisingly.

Below are the boxplots that show how the predictor variables depend on the wine quality category. I will be sometimes using boxplots with outliers and sometimes without outliers, depending on which one is more informative for the predictor. Outliers on the boxplots will be displayed as points. Anything falling outside of the first quartile minus 1.5*IQR and third quartile plus 1.5*IQR is classified as an outlier.



As the quality of wine increases fixed acidity also increases a little bit with a more pronounced leap from the medium category to the good category than from the bad category to the medium category. The spread in fixed acidity is also larger in medium and good categories.

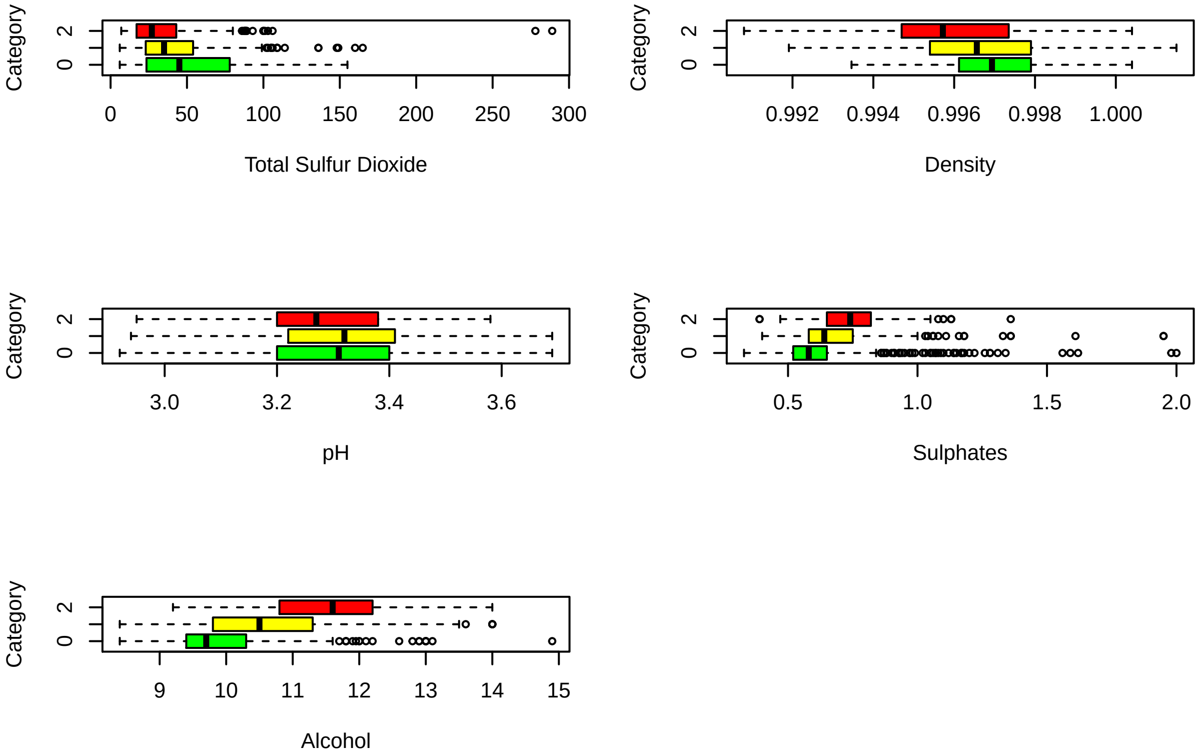
Here we see an inverse correlation between category and volatile acidity. The median level of volatile acidity steadily decreases as the category betters. We can also see that the bad wine category has a bunch of outliers with unusually high levels of volatile acidity.

As the wine quality betters the amount of citric acid slightly increases generally speaking. the IQR for the good category is notably smaller than for the other categories.

With residual sugar there does not seem to be much of a relation between its median and category. The thing that stands out here is the huge amount of outliers that are way higher than the median. This applies for each of the categories.

With chlorides the median slightly decreases with higher categories and the IQR remains roughly the same. We can see that the size of outliers decreases as the category value increases.

The free sulfur dioxide median as well as the variance is slightly smaller for good wines than for the two other categories.



Total sulfur dioxide slightly decreases as the category goes up, and so does the variance. The variance for the bad wines is much bigger than for the other categories.

Density slightly decreases as the category value increases and the variance in density for bad wines is smaller than for the rest.

There is very little change in ph from category to category. The variance of it for the good category is slightly smaller.

The higher the category the more sulphates typically. With that being said the wines with the greatest amount of sulphates are also not good. The bad and medium categories have more outliers when it comes to high levels of sulphates.

Alcohol on average increases notably as the quality increases. The graph shows a clear relationship. The bad category has the greatest amount of outliers and among those outliers lies a wine with the highest amount of alcohol out of all wines in the dataset.

Section 2: Methods/Analysis

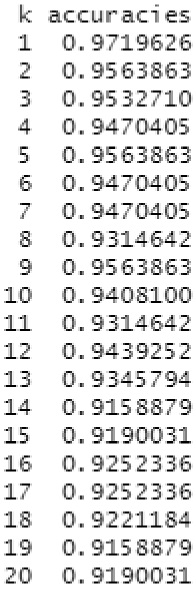
After performing exploratory analysis I split the modified dataset of 1,599 rows with an 80%/20% split to divide it into a training and testing part, and applied four different algorithms to select the best model for predicting wine quality category. The final selection was done after running the models on the testing set. The idea is to compare the results and see which does the best job in terms of confusion matrix accuracy and RMSE.

The four methods are K-Nearest Neighbors (KNN), Classification And Regression Trees (CART) method, Random Forest (RF) and an ensemble using KNN, CART and RF.

2.1. KNN method

The KNN algorithm is used for classification and regression. In KNN classification the output is a categorical variable which defines a class. Each observation is assigned to a class by a plurality vote of its k nearest neighbors, where k is usually small. It can be equal to 1, in which case the observation is assigned to the same class as the nearest neighbor. This algorithm uses distance between the observations for classification. If the variables are measured on different scales then normalizing the data greatly improves the algorithm accuracy. The distance between two observations is calculated as euclidean distance between two points where the formula is the square root of the sum of squared differences of the observation measurements. To apply the KNN method I normalized all of the predictors.

I ran the KNN algorithm for each observation in the test set by finding the closest observation(s) to it from the training set. I tested the accuracy for k values 1 through 20 to determine which k value gives the most accurate result. Looking at the table of ks matched with their corresponding accuracies as well as looking at the graph of k vs accuracy the clear winner is k = 1, so this is the k value we will stick with. The graph displays a fairly straight forward decrease in accuracy as the number of neighbors increases. The obtained accuracy is 0.972 for k = 1. Below is the table and graph of the accuracy results for each of the ks:



```
  k accuracies
  1  0.9719626
  2  0.9563863
  3  0.9532710
  4  0.9470405
  5  0.9563863
  6  0.9470405
  7  0.9470405
  8  0.9314642
  9  0.9563863
 10  0.9408100
 11  0.9314642
 12  0.9439252
 13  0.9345794
 14  0.9158879
 15  0.9190031
 16  0.9252336
 17  0.9252336
 18  0.9221184
 19  0.9158879
 20  0.9190031
```

Figure 9: Accuracy by k Table

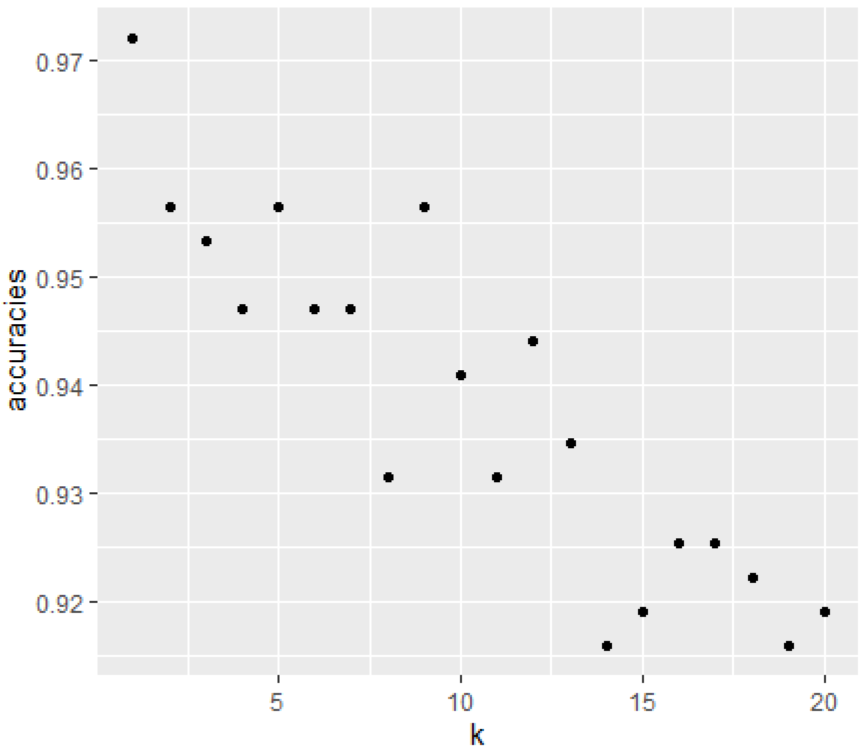
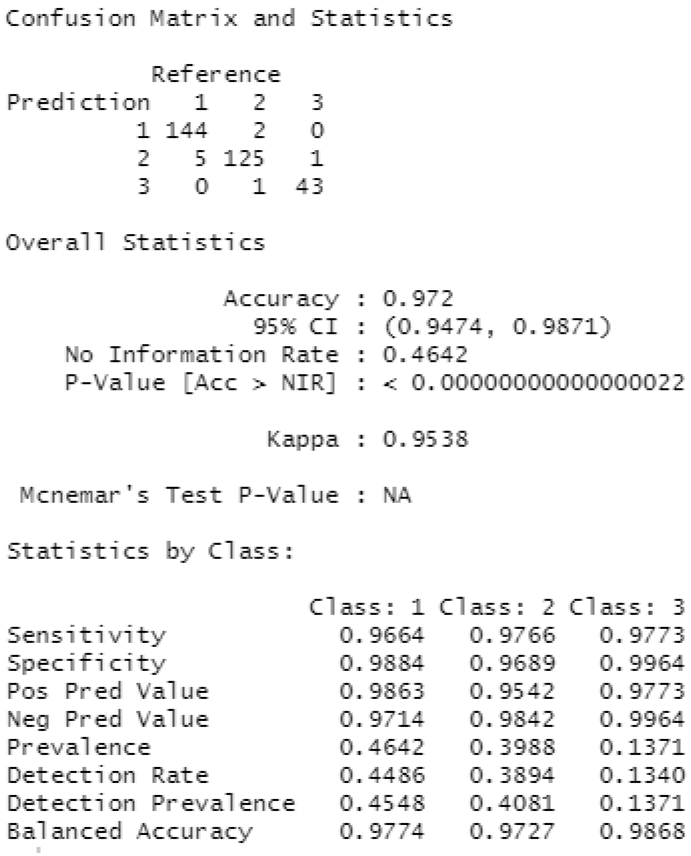


Figure 10: k vs Accuracy Graph

Here are the results of the confusion matrix:



```
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3
         1 144   2   0
         2   5 125   1
         3   0   1  43

Overall Statistics

               Accuracy : 0.972
                 95% CI : (0.9474, 0.9871)
    No Information Rate : 0.4642
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.9538

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9664   0.9766   0.9773
Specificity            0.9884   0.9689   0.9964
Pos Pred Value         0.9863   0.9542   0.9773
Neg Pred Value         0.9714   0.9842   0.9964
Prevalence             0.4642   0.3988   0.1371
Detection Rate         0.4486   0.3894   0.1340
Detection Prevalence   0.4548   0.4081   0.1371
Balanced Accuracy      0.9774   0.9727   0.9868
```

Figure 11: Confusion Matrix for KNN with k = 1

2.2. CART method

A decision tree is another classification algorithm. A decision tree or a classification tree is built by splitting the source set. The algorithm starts from the root node of the tree, it compares the values of the selected for the root variable with the pre-selcted cutoff, and based on this comparison branches the corresponding to the next level node. The process continues until the node has all the same values of the classified variable, or when splitting no longer improves the GINI index or entropy which are measuring the impurity of the node (a node with multiple classes is impure, and a node with only one class is pure).

This method faired a little worse than KNN, resulting in the lowest accuracy and highest RMSE.

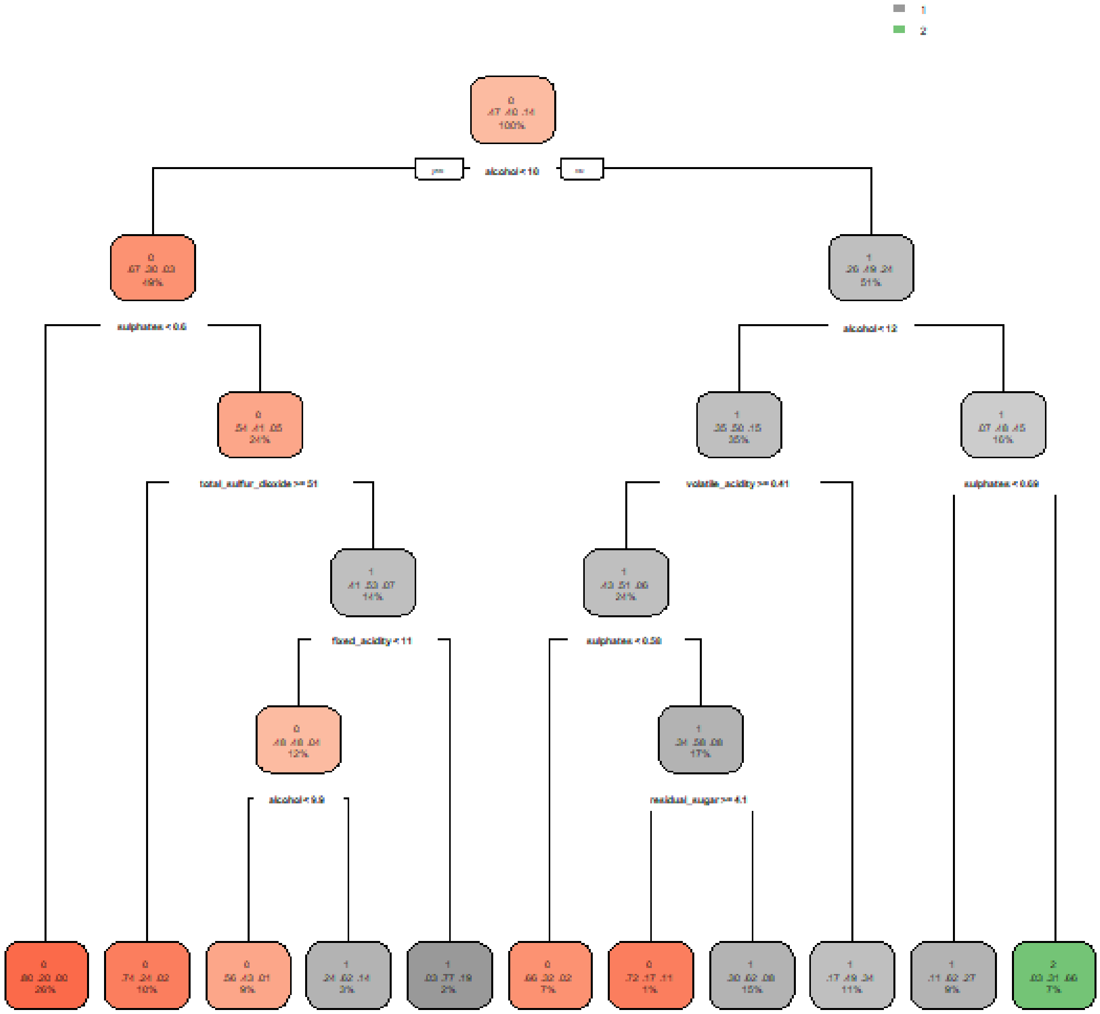
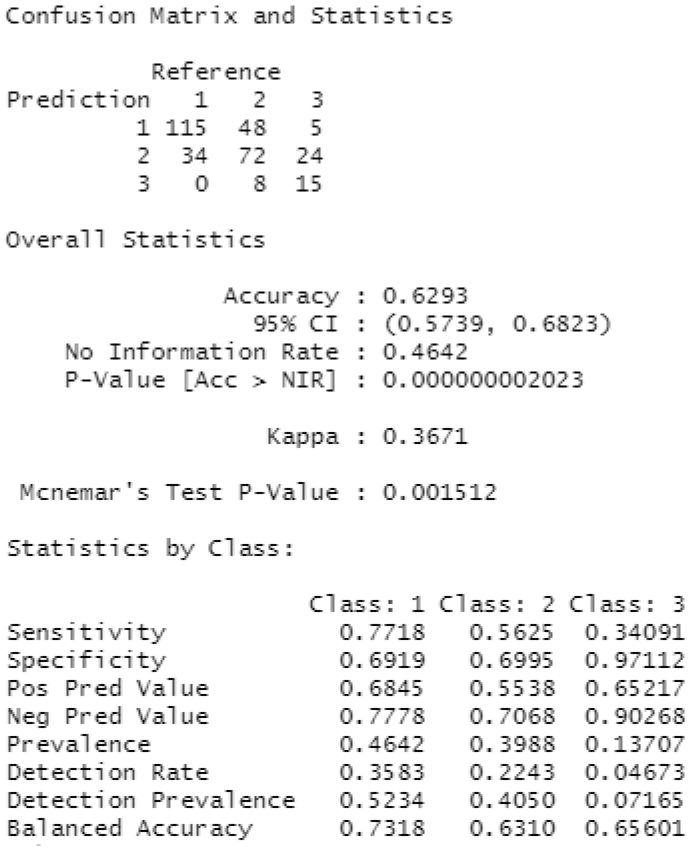


Figure 12: Decision Tree Diagram

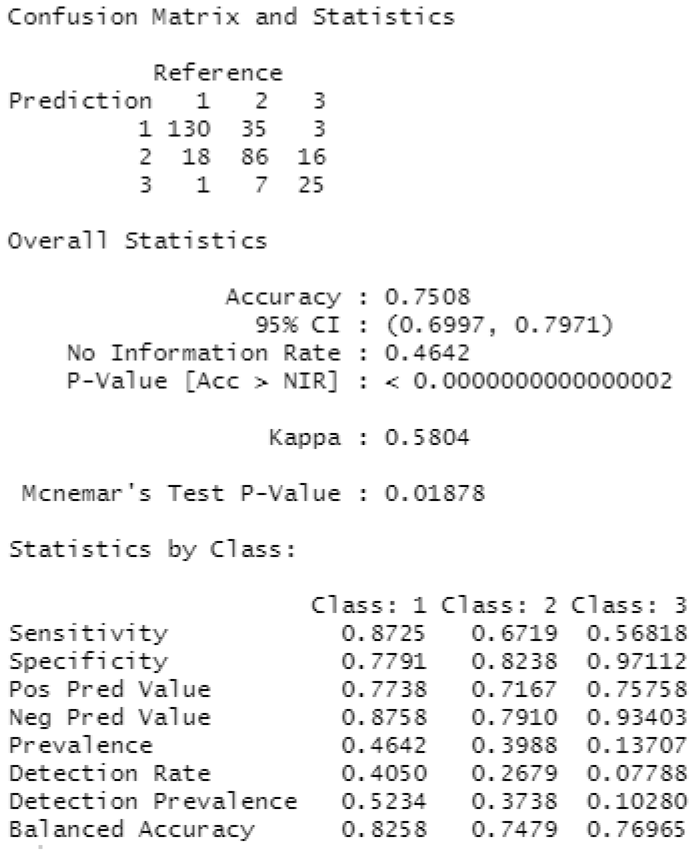


```
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3
         1 115  48   5
         2  34  72  24
         3   0   8  15

Overall Statistics

               Accuracy : 0.6293
                 95% CI : (0.5739, 0.6823)
    No Information Rate : 0.4642
    P-Value [Acc > NIR] : 0.000000002023

                  Kappa : 0.3671

 Mcnemar's Test P-Value : 0.001512

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.7718   0.5625  0.34091
Specificity            0.6919   0.6995  0.97112
Pos Pred Value         0.6845   0.5538  0.65217
Neg Pred Value         0.7778   0.7068  0.90268
Prevalence             0.4642   0.3988  0.13707
Detection Rate         0.3583   0.2243  0.04673
Detection Prevalence   0.5234   0.4050  0.07165
Balanced Accuracy      0.7318   0.6310  0.65601
```

Figure 13: Confusion Matrix for Decision Tree Algorithm

2.3. Random Forest method

Random forest is a version of a decision tree ensemble method. It creates multiple decision trees using multiple samples with replacement from the training dataset, and then uses a plurality vote of all built trees. This works well as a single decision tree often is prone to overfitting the data. The accuracy of random forest is generally better than the accuracy of one decision tree. The disadvantage of random forest is that comparing to decision tree is not easily interpretable, and more similar to a “blackbox” model.



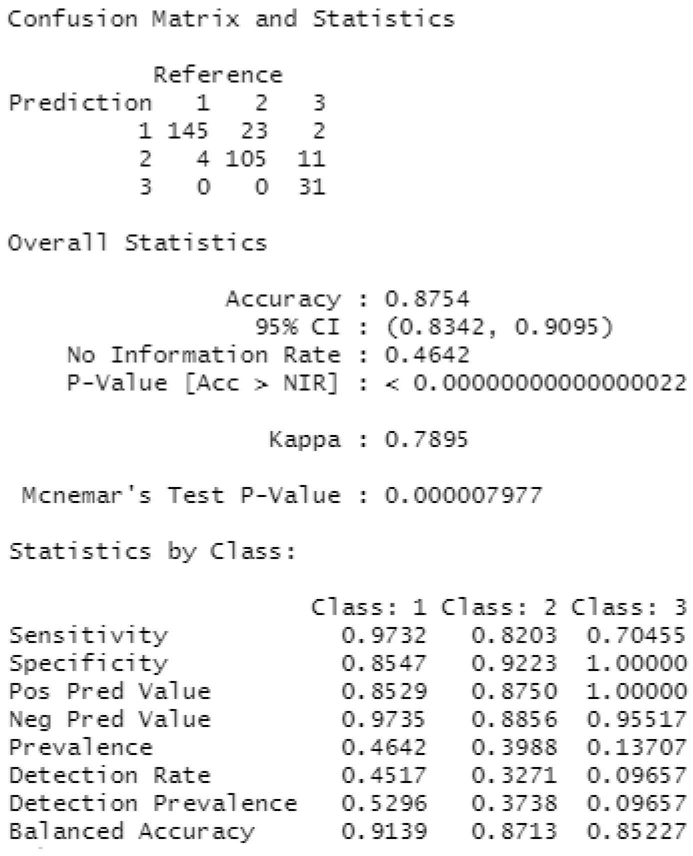
```
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3
         1 130  35   3
         2  18  86  16
         3   1   7  25

Overall Statistics
                                          
               Accuracy : 0.7508          
                 95% CI : (0.6997, 0.7971)
    No Information Rate : 0.4642          
    P-Value [Acc > NIR] : < 0.0000000000000002
                                          
                  Kappa : 0.5804          
                                          
 Mcnemar's Test P-Value : 0.01878         

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.8725   0.6719  0.56818
Specificity            0.7791   0.8238  0.97112
Pos Pred Value         0.7738   0.7167  0.75758
Neg Pred Value         0.8758   0.7910  0.93403
Prevalence             0.4642   0.3988  0.13707
Detection Rate         0.4050   0.2679  0.07788
Detection Prevalence   0.5234   0.3738  0.10280
Balanced Accuracy      0.8258   0.7479  0.76965
```

Figure 14: Random Forest

Random Forest produced an accuracy and an RMSE that is between those of KNN and decision tree.

2.4. Ensemble method

The final method that I used was an ensemble method of KNN, decision tree and random forest. This gave the second best result after KNN. I thought it was worth a try since some of the methods may work well for some wines and other methods for other wines.

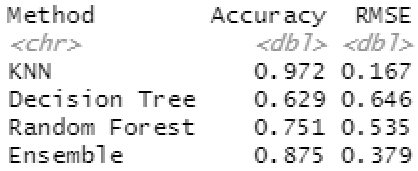


```
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3
         1 145  23   2
         2   4 105  11
         3   0   0  31

Overall Statistics

               Accuracy : 0.8754
                 95% CI : (0.8342, 0.9095)
    No Information Rate : 0.4642
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.7895

 Mcnemar's Test P-Value : 0.000007977

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            0.9732   0.8203  0.70455
Specificity            0.8547   0.9223  1.00000
Pos Pred Value         0.8529   0.8750  1.00000
Neg Pred Value         0.9735   0.8856  0.95517
Prevalence             0.4642   0.3988  0.13707
Detection Rate         0.4517   0.3271  0.09657
Detection Prevalence   0.5296   0.3738  0.09657
Balanced Accuracy      0.9139   0.8713  0.85227
```

Figure 15: Ensemble Results

Section 3: Results

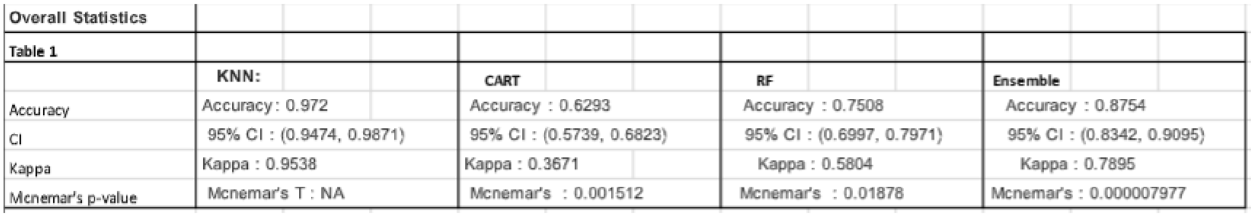
The following table shows the performance of all methods used.



```
Method         Accuracy  RMSE
<chr>             <dbl> <dbl>
KNN               0.972 0.167
Decision Tree     0.629 0.646
Random Forest     0.751 0.535
Ensemble          0.875 0.379
```

Figure 16: Results Table

The next table shows the accuracy, kappa and McNemar's test results.



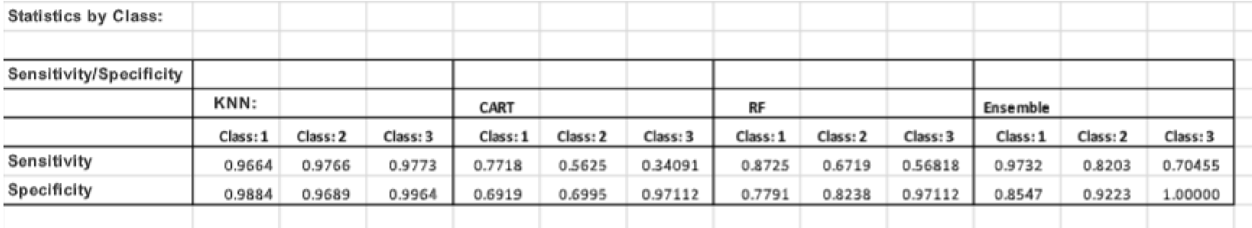
Overall Statistics				
Table 1				
	KNN:	CART	RF	Ensemble
Accuracy	Accuracy: 0.972	Accuracy : 0.6293	Accuracy : 0.7508	Accuracy : 0.8754
CI	95% CI : (0.9474, 0.9871)	95% CI : (0.5739, 0.6823)	95% CI : (0.6997, 0.7971)	95% CI : (0.8342, 0.9095)
Kappa	Kappa : 0.9538	Kappa : 0.3671	Kappa : 0.5804	Kappa : 0.7895
Mcnemar's p-value	Mcnemar's T : NA	Mcnemar's : 0.001512	Mcnemar's : 0.01878	Mcnemar's : 0.000007977

Figure 17: Overall Statistics

Using the criteria of Cohen's Kappa the results of KNN are almost perfect (0.81 - 1), the results of the decision tree method are fair (0.21 - 0.4), the results of random forest are moderate (0.41 - 0.6) and the results of the ensemble method are substantial (0.61 - 0.8). Kappa = (observed accuracy - expected accuracy)/(1 - expected accuracy). Is is a measure of how much the observed accuracy differs from the expected accuracy.

McNemar's test compares the rates of false positives and false negatives where small values for the test result show that the values occur at statistically different rates. For KNN the test is inapplicable and for all the other methods the result is statistically significant at the 5% level.

The following table summarizes the results for specificity and sensitivity.

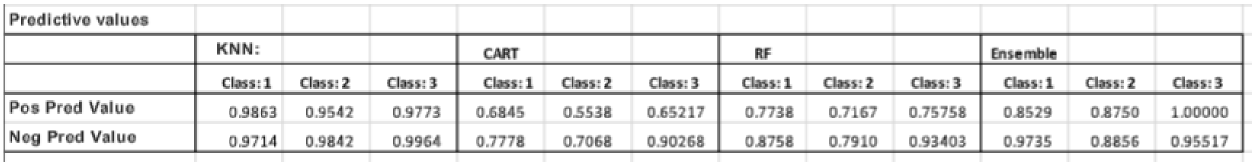


Statistics by Class:												
Sensitivity/Specificity												
	KNN:			CART			RF			Ensemble		
	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3
Sensitivity	0.9664	0.9766	0.9773	0.7718	0.5625	0.34091	0.8725	0.6719	0.56818	0.9732	0.8203	0.70455
Specificity	0.9884	0.9689	0.9964	0.6919	0.6995	0.97112	0.7791	0.8238	0.97112	0.8547	0.9223	1.00000

Figure 18: Sensitivity and Specificity by Class

Sensitivity is the true positive rate, (true positives) / (true positives + false positives). It is the best for KNN and the lowest for the decision tree method.

Specificity is the true negative rate, (true negatives) / (true negatives + false negatives). Overall KNN fairs the best for specificity with some noticeable variation in specificity for the decision tree method. The following table shows the positive predictive values and negative predicted values.



Predictive values												
	KNN:			CART			RF			Ensemble		
	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3
Pos Pred Value	0.9863	0.9542	0.9773	0.6845	0.5538	0.65217	0.7738	0.7167	0.75758	0.8529	0.8750	1.00000
Neg Pred Value	0.9714	0.9842	0.9964	0.7778	0.7068	0.90268	0.8758	0.7910	0.93403	0.9735	0.8856	0.95517

Figure 19: Predictive Values by Class

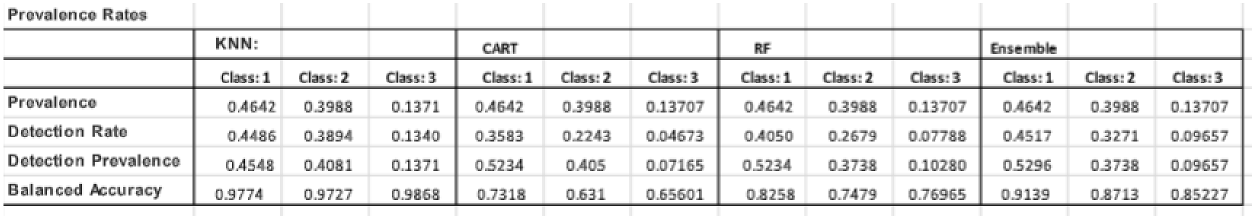
The positive predicted value is defined as (sensitivity x prevalence) / ((sensitivity x prevalence) + ((1 – specificity) x (1 – prevalence)))

The positive predicted value fairs best for KNN and ensemble with KNN having less variance among categories.

Negative predicted value is defined as (specificity x (1 – prevalence)) / ((specificity x (1 – prevalence)) + ((1 – sensitivity) x prevalence))

The same can be said about the negative predicted value as was said about positive predicted value.

The last table shows the prevalence rates.



Prevalence Rates												
	KNN:			CART			RF			Ensemble		
	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3	Class: 1	Class: 2	Class: 3
Prevalence	0.4642	0.3988	0.1371	0.4642	0.3988	0.13707	0.4642	0.3988	0.13707	0.4642	0.3988	0.13707
Detection Rate	0.4486	0.3894	0.1340	0.3583	0.2243	0.04673	0.4050	0.2679	0.07788	0.4517	0.3271	0.09657
Detection Prevalence	0.4548	0.4081	0.1371	0.5234	0.405	0.07165	0.5234	0.3738	0.10280	0.5296	0.3738	0.09657
Balanced Accuracy	0.9774	0.9727	0.9868	0.7318	0.631	0.65601	0.8258	0.7479	0.76965	0.9139	0.8713	0.85227

Figure 20: Prevalence Rates by Class

Detection rate is (true positives) / (total number of cases). It is generally rather uneven by category but on average again fairs best among KNN and then ensemble.

Detection prevalence is the number of cases predicted to be positive divided by the total of cases. For detection prevalence it is a little less clear which method does best. KNN does the best in detective prevalence for class 2 but the worst in detective prevalence for class 1. All the methods do significantly worse for category 3 in temrs of detective prevalence than for the other classes.

Balanced accuracy is the average value of sensitivity and specificity. Again the most often encountered pattern emerges with KNN fairing the best, then the ensemble, then random forest and then decision tree.

Conclusion

After loading in the dataset and doing some wrangling the data was split up into a training and testing part. All the methods used in this project are well known in machine learning. Four approaches were tested out and compared. For this task the decision tree algorithm turned out less effective than the others while KNN of the four turned out to be the best. The results were neatly gathered together into a table for easy comparison.

KNN was the suggested method for this project on the kaggle site. The model performed best with one nearest neighbor and showed a fairly linear negative correlation between accuracy and the amount of nearest neighbors. Aggregating the qualities together turned out to give a better accuracy then using KNN for predicting the originally given variable of quality.

Aggregating wine qualities into categories did improve the accuracy. The accuracy I obtained for the aggregated categories was 0.972 while predicting the original quality variable the obtained accuracy was 0.875. For both cases k = 1 gave the best results with a clear negative correlation between k and accuracy.

Future Work and Limitations

Potentially it may be interesting to see how multivariate logistic regression would work for the task even though for this project I chose more complex methods. Also this analysis was done only on red wines from a specific region of Portugal. If say the wines were to be taken from France or Georgia it is quiet possible that some of the relations seen here would not hold up, perhaps due to different natural environments or wine production practices.

References

1. Breiman, L (2002), “Manual On Setting Up, Using, And Understanding Random Forests V3.1”, https://www.stat.berkeley.edu/~breiman/Using_random_forests_V3.1.pdf

2. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

3. Map: https://winefolly.com/deep-dive/what-wines-to-drink-from-portugal-by-region/

4. https://waterhouse.ucdavis.edu/whats-in-wine/red-wine-composition

5. https://en.wikipedia.org/wiki/Confusion_matrix