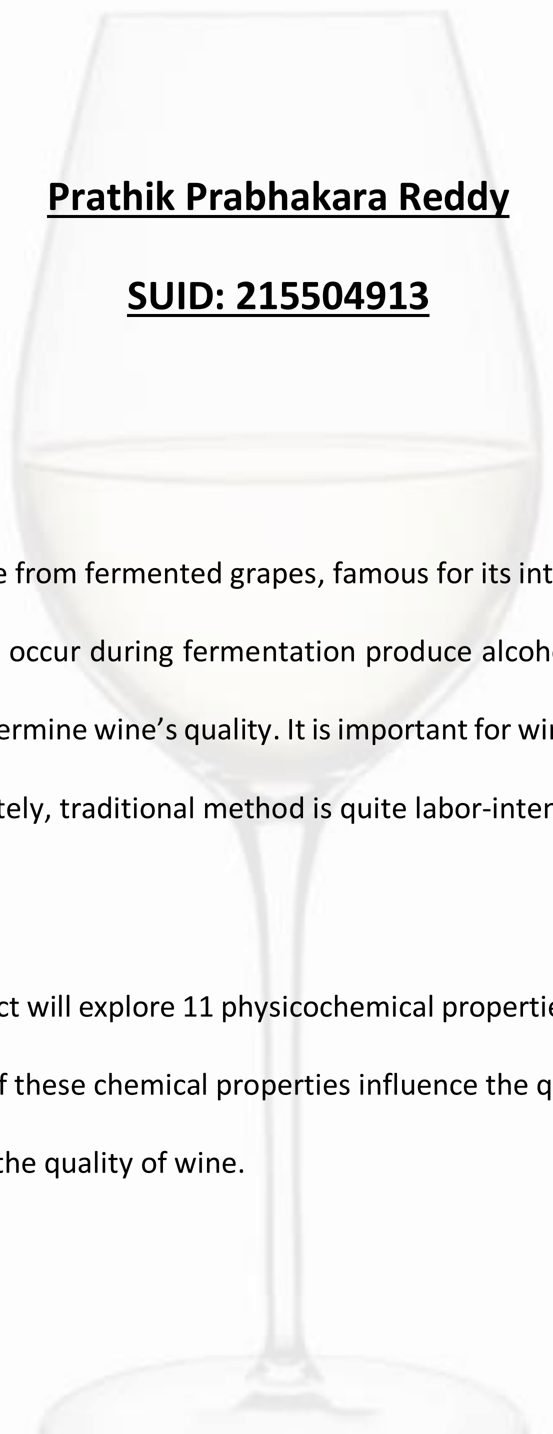
SYRACUSE UNIVERSITY

PAI 793 : PREDICTIVE ANALYTICS

FINAL PROJECT PAPER

WHITE WINE QUALITY PREDICTION



Prathik Prabhakara Reddy

SUID: 215504913

Introduction

Wine, an alcoholic drink made from fermented grapes, famous for its intoxicating effects and delicacy. The biochemical process that occur during fermentation produce alcohol and substances that enrich the flavor of the wine and determine wine's quality. It is important for wine producers to classify wine's quality accurately. Unfortunately, traditional method is quite labor-intensive and time-consuming.

Project Objective

This R machine learning project will explore 11 physicochemical properties of wine. The purpose of this project is to evaluate which of these chemical properties influence the quality of the wines and to find the best approach to predict the quality of wine.

Why Predictive Analytics?

Wine classification is a difficult task since taste is the least understood of the human senses. A good wine quality prediction can be useful in the certification phase, since currently the sensory analysis is performed by human tasters, which is a subjective approach. An automatic prediction system can be integrated into a decision support system, helping the speed and quality of the performance. Moreover, a feature selection process can help to analyze which input variables are highly relevant to predict the wine quality. This can also be used by wine producers to improve the quality based on physiochemical properties.

Data Description

The dataset used is Wine Quality Dataset from UCI Machine Learning Repository and is related to the white wine variants of the Portuguese Vin ho Verde variety of wine. Due to privacy & logistics issues the dataset only contains **11** physiochemical (input) and **1** sensory (output) variables, with **4898** rows.

https://archive.ics.uci.edu/ml/datasets/wine+quality

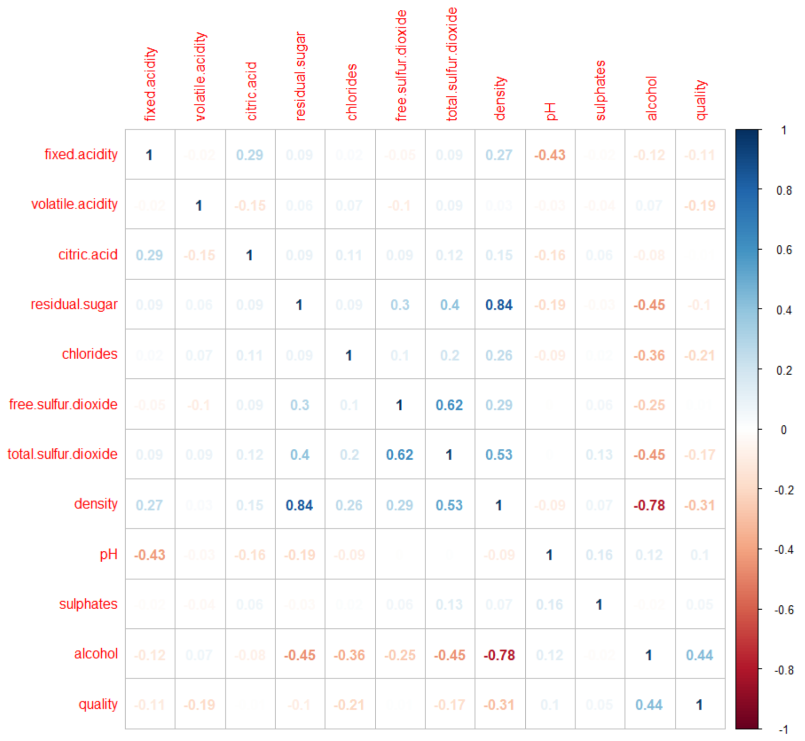
https://www.kaggle.com/piyushagni5/white-wine-quality

Attribute Description

Attribute	**Description**
fixed acidity	Acidity is a fundamental property of wine, imparting sourness, and resistance to microbial infection. Fixed acidity is the no. of grams of tartaric acid per dm3
volatile acidity	Wine spoilage is legally defined by volatile acidity which is calculated as no. of grams of acetic acid per dm3 of wine
citric acid	It is the no. of grams of citric acid per dm3 of wine

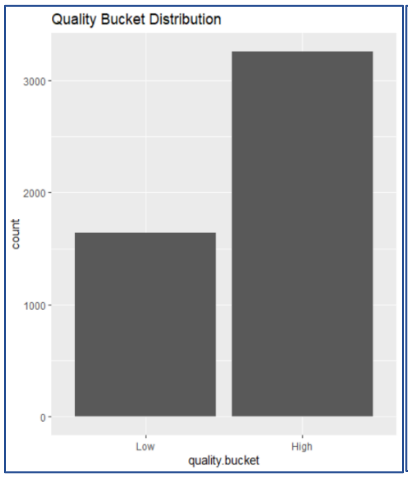
residual sugar	Sugar remaining after fermentation stops. Given as no. of grams per dm3
chlorides	It is the no. of grams of sodium chloride per dm3 of wine
free sulfur dioxide	It is the no. of grams of free sulfites per dm3 of wine
total sulfur dioxide	It is the no. of grams of total sulfite in per dm3 of wine
density	It gives the density of the wine in grams per cm3
pH	It gives the pH of the wines, used to measure ripeness in relation to acidity
sulphates	It gives the no. of grams of potassium sulphate per dm3 of wine
alcohol	It gives the volume of alcohol in percentage
quality bucket (Target variable)	Here the wine is rated from 1- 10 based on the quality. 1-5: Low Quality, 6-10: High Quality

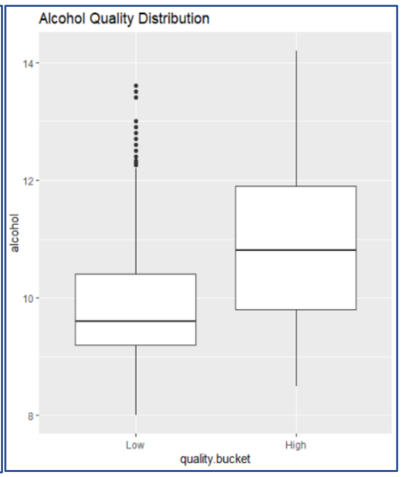
Correlation Matrix



The above correlation matrix indicates that variables density and residual sugar have high positive correlation, and variables density and alcohol have high negative correlation. Correlated variables in a regression setting gives impaired output, hence we will drop the less important column before analysis.

The quality variable has some positive correlation with alcohol level. The bar plot on the right shows the distribution of quality bucket variable and boxplot shows distribution of alcohol according to the quality bucket.





Models Implemented

1. Logistic Regression
2. Linear Discriminant Analysis
3. K-Nearest Neighbors
4. Decision Tree
5. Bagging
6. Random Forest
7. Boosting

The response variable in this dataset is qualitative with 2 levels. Here, I want to predict if a wine with given physiochemical properties is of low or high quality. To predict a qualitative variable, we use classification models, which are best to answer yes or no questions, providing broad analysis that is helpful for guiding decisive action. The above-mentioned models are significant in resolving a classification problem. Logistic Regression and Decision Trees provided more interpretability indicating important properties that affect the quality. Random forests and ensemble methods provided higher prediction accuracy but lower interpretability. One of the simplest and best-known non-parametric methods is K-nearest neighbor classification (KNN).

Cross-Validation Approach and Normalization

All the classification models in this project have implemented with K-fold cross validation, with keeping K at the optimal values (5 or 10), to compare the model accuracy. The mean miscalculation error rate for each model has been calculated and compared for best model selection.

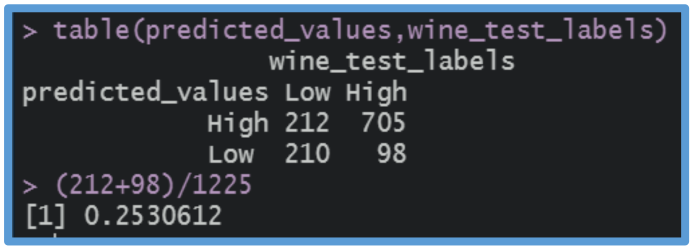
The dataset has been divided at 75:25 to training dataset and testing dataset. So, all the classification models can be first trained and then tested to find the accuracy of prediction.

The KNN classifier predicts the class of a given test observation by identifying the observations that are nearest to it, the scale of the variables matters. I have handled this problem by standardizing the data so that all standardized variables are given a mean of zero and a standard deviation of one.

For Boosting models, Cross Validation method is used to find the best iteration number, which is used as number of trees while predicting.

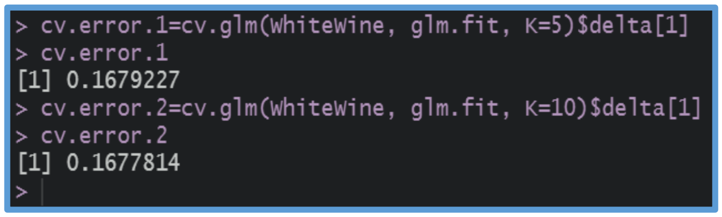
Logistic Regression

Logistic Regression models the probability that Y (response variable) belongs to a particular category. It is used for regression and classification on a



```
> table(predicted_values,wine_test_labels)
                wine_test_labels
predicted_values Low High
            High 212  705
            Low  210   98
> (212+98)/1225
[1] 0.2530612
```

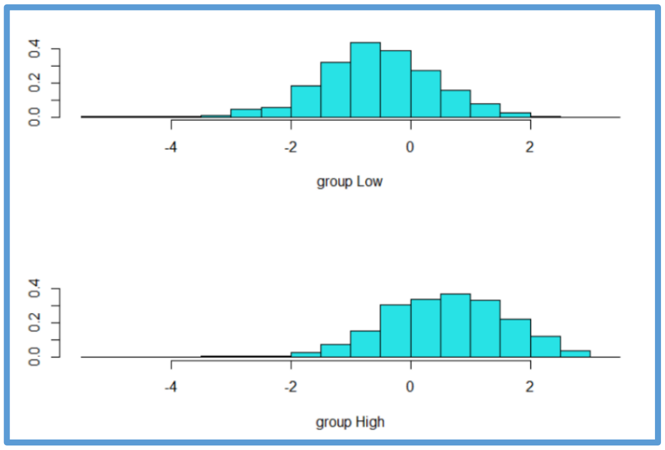
binomial dependent variable. Here, Logistic Regression has predicted test values with a mis - -classification rate of **0.2531.** After implementing a 5–fold and 10–fold cross validation the mean error rate is **0.1679**

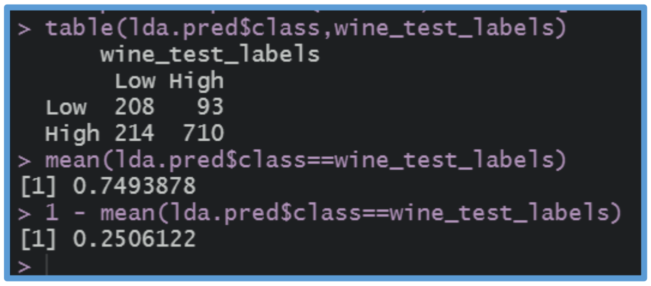


```
> cv.error.1=cv.glm(whiteWine, glm.fit, K=5)$delta[1]
> cv.error.1
[1] 0.1679227
> cv.error.2=cv.glm(whiteWine, glm.fit, K=10)$delta[1]
> cv.error.2
[1] 0.1677814
>
```

Linear Discriminant Analysis

LDA is popular if there are more than two responses, but let us check the model performance with binomial response variable as compared to logistic regression. The LDA output indicates that 66.83% of the training observations corresponds to High wine and 33.16% correspond to low quality wine. It also provides group means and can create clear distinct groups for Low and High wines. The mis-classification error rate for LDA on testing set is **0.2506**, which is more than that of Logistic Regression.



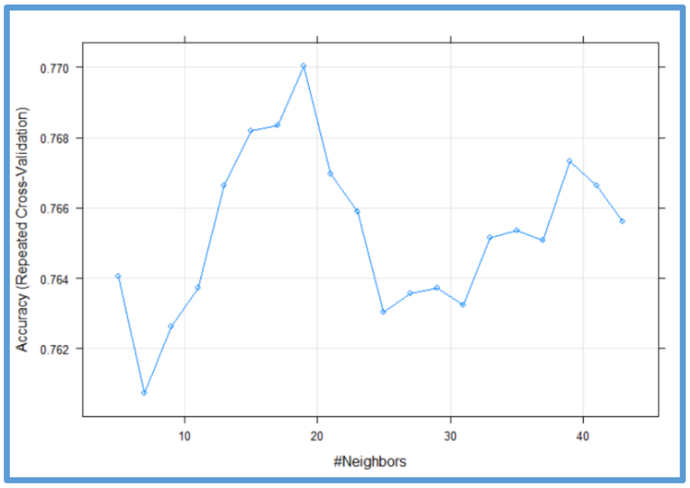


```
> table(lda.pred$class,wine_test_labels)
      wine_test_labels
       Low High
  Low  208   93
  High 214  710
> mean(lda.pred$class==wine_test_labels)
[1] 0.7493878
> 1 - mean(lda.pred$class==wine_test_labels)
[1] 0.2506122
>
```

K – Nearest Neighbor (KNN)

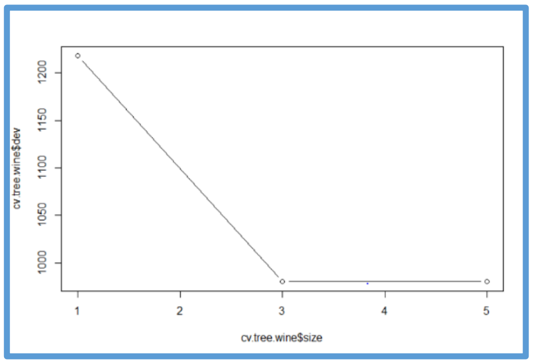
Here we consider one of the simplest and best-known non-parametric methods, K-nearest neighbor classification (KNN). This algorithm is simple to implement, robust to noisy training data. Classification is computed from a simple majority vote of the k nearest neighbors of each point.

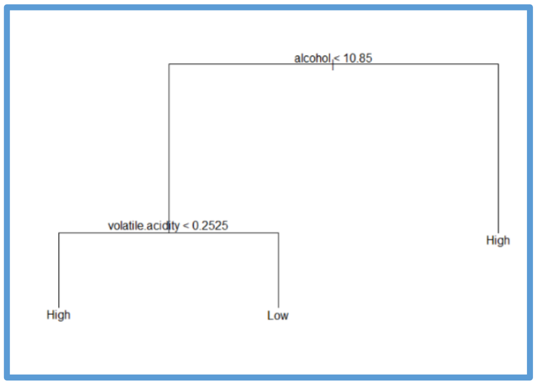
The first step in adjusting the k-nearest neighbor model was to fix the number of neighbors k. For that, we used 10- fold cross validation and chose k such that the CV residual mean error is minimized. This yielded to a result of k = 19 as the final value used for the model. This can be seen in the figure on right. The misclassification error rate of KNN with is **0.2211**

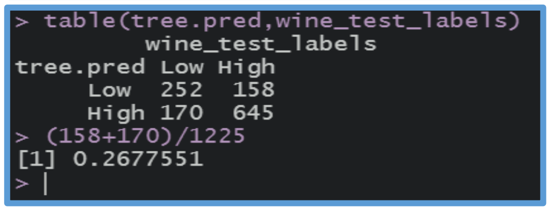


Decision Tree Classifier

Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data. Decision Tree is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data. First, trained and tested tree model to get a misclassification error rate of **0.2677551**. Next, pruned this tree using cross validation. The plot is shown on the right. Three nodes result in the lowest cross-validation error. Based on the results from cross validation, pruned the tree to the smallest number of terminal nodes that reduces the deviance. The error rate did not change. The most important features to predict quality of wine are alcohol and volatile acidity





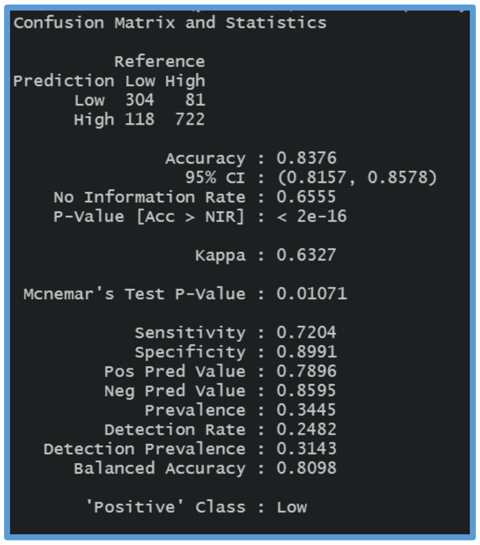


```
> table(tree.pred,wine_test_labels)
         wine_test_labels
tree.pred Low High
     Low  252  158
     High 170  645
> (158+170)/1225
[1] 0.2677551
> |
```

Bagging

The previous models (decision trees) suffer from high variance.

Bagging models can reduce this by taking multiple training sets and building separate trees for each training set and average the resulting predictions. The confusion matrix results on the right gives the accuracy (**0.8376)** of bagging on testing dataset. The misclassification error rate is **0.1624.** Which means, bagging model can predict the quality of wine with 83.76% accuracy.

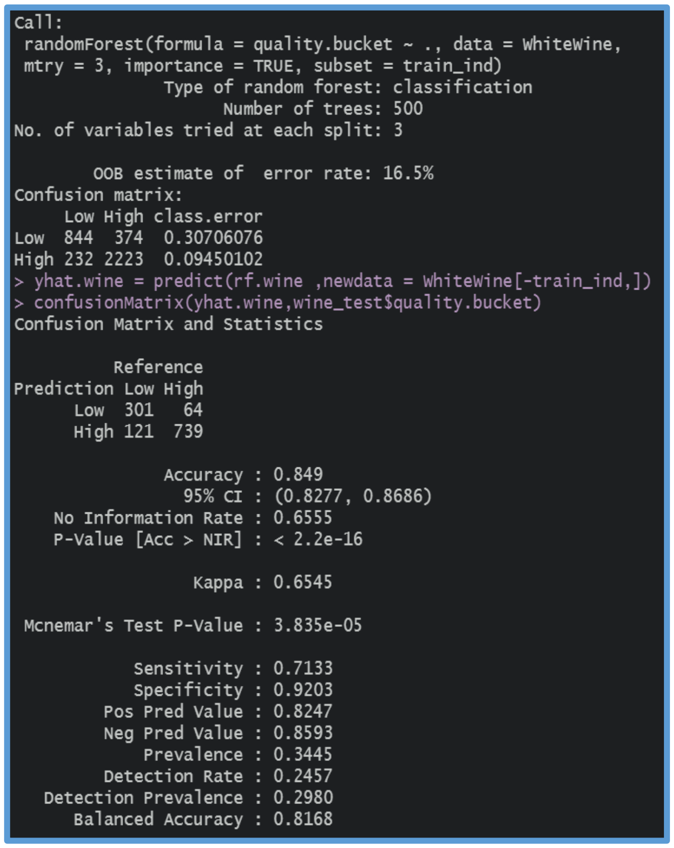


```
Confusion Matrix and Statistics

          Reference
Prediction Low High
      Low  304   81
      High 118  722

               Accuracy : 0.8376
                 95% CI : (0.8157, 0.8578)
    No Information Rate : 0.6555
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.6327

 Mcnemar's Test P-Value : 0.01071

            Sensitivity : 0.7204
            Specificity : 0.8991
         Pos Pred Value : 0.7896
         Neg Pred Value : 0.8595
             Prevalence : 0.3445
         Detection Rate : 0.2482
   Detection Prevalence : 0.3143
      Balanced Accuracy : 0.8098

       'Positive' Class : Low
```

Random Forest Classifier

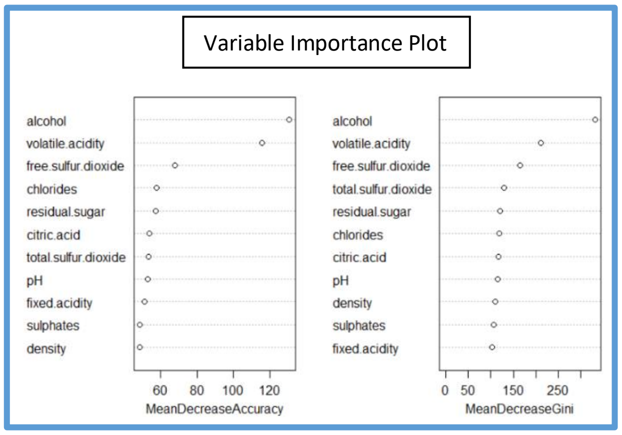
This classifier is an improvement from Bagging as Random Forest decorrelates the trees. The table below gives the misclassification rates of Random Forest model on testing data for different m predictors. The best model, with least test misclassification error was obtained with m = 3, with error of **0.1510.** Output is shown on the right

M (predictors at split)	Misclassification Error
2	**190/1225 = 0.1551**
3	**185/1225 = 0.1510**
4	**194/1225 = 0.1583**



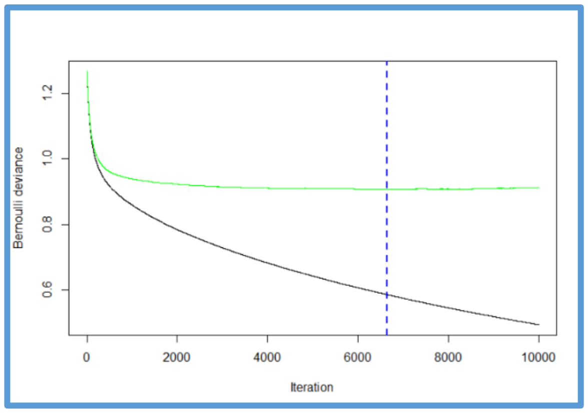
```
Call:
 randomForest(formula = quality.bucket ~ ., data = WhiteWine,
 mtry = 3, importance = TRUE, subset = train_ind)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 16.5%
Confusion matrix:
     Low High class.error
Low  844  374  0.30706076
High 232 2223  0.09450102
> yhat.wine = predict(rf.wine ,newdata = WhiteWine[-train_ind,])
> confusionMatrix(yhat.wine,wine_test$quality.bucket)
Confusion Matrix and Statistics

          Reference
Prediction Low High
      Low  301   64
      High 121  739

               Accuracy : 0.849
                 95% CI : (0.8277, 0.8686)
    No Information Rate : 0.6555
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6545

 Mcnemar's Test P-Value : 3.835e-05

            Sensitivity : 0.7133
            Specificity : 0.9203
         Pos Pred Value : 0.8247
         Neg Pred Value : 0.8593
             Prevalence : 0.3445
         Detection Rate : 0.2457
   Detection Prevalence : 0.2980
      Balanced Accuracy : 0.8168
```

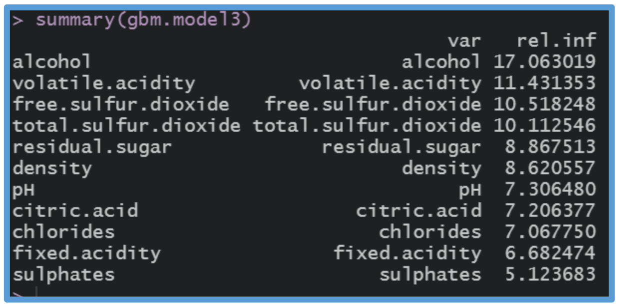
After training a random forest, it is natural to ask which variables have the most predictive power. Variables with high importance are drivers of the outcome and their values have a significant impact on the outcome values. By contrast, variables with low importance might be omitted from a model, making it simpler and faster to fit and predict. The first measure in the plot on the right is based on how much the accuracy decreases when the variable is excluded. The second measure, when a tree is built, the decision about which variable to split at each node uses a calculation of the Gini impurity. The most important variables to predict wine quality are alcohol, volatile.acidity, and free.sulphur.dioxide.



Boosting

Boosting works in a similar way to Bagging, except that the trees are grown sequentially: each tree is grown using information from previously grown trees. Boosting does not involve bootstrap sampling; instead, each tree is fit on a modified version of the original data set. Here, I used cross validation method to check the best iteration number. According to CV, boosting tree with **6648** gives the best boosting model with least misclassification error. The test misclassification error for different values of depth and shrinkage are shown in table below.



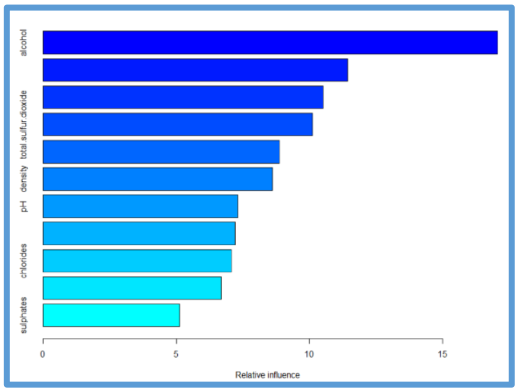


```
> summary(gbm.model3)
                                      var   rel.inf
alcohol                           alcohol 17.063019
volatile.acidity         volatile.acidity 11.431353
free.sulfur.dioxide   free.sulfur.dioxide 10.518248
total.sulfur.dioxide total.sulfur.dioxide 10.112546
residual.sugar             residual.sugar  8.867513
density                           density  8.620557
pH                                     pH  7.306480
citric.acid                   citric.acid  7.206377
chlorides                       chlorides  7.067750
fixed.acidity               fixed.acidity  6.682474
sulphates                       sulphates  5.123683
```

The Boosting model with interaction depth 3 and shrinkage 0.01 has the lowest test misclassification error of **0.1910**

Interaction Depth	Shrinkage	Error
2	0.01	0.2065
2	0.001	0.2192
3	0.01	0.1910

Plots the marginal effect of the selected variables by "integrating" out the other variables. How your features affect the dependent variable and shows the importance level of features in the model.



Model Results Comparison

By comparing the Cross-Validation mis-classification error rates of all the models, Random Forest works the best for this classification problem. Random Forest with 3 predictors at the tree split (m=3) can predict the quality of wine (Low or High) with **84.90%** accuracy.

MODEL	Misclassification Error
Logistic Regression	**0.1679**
Linear Discriminant Analysis	**0.2506**
KNN	**0.2211**
Decision Tree	**0.2677**
Bagging	**0.1624**
Random Forest	**0.1510**
Boosting	**0.1910**

Conclusion

Based on the analysis in this project, I was able to find out the important features that influence the quality of wine. The top three features are alcohol, volatile acidity, and Free Sulphur dioxide.

For classifying the wine quality, I have implemented multiple algorithms, out of which, Random Forest had the most accurate wine quality prediction with 84.92% accuracy, followed by Bagging and Logistic Regression with 83.76% and 83.21% prediction accuracies.

At the end of the story I can say that Wine quality is a very complex study. Good wine is more than perfect combination of different chemical components. Future improvement can be made if more data can be collected on both low-quality and high-quality wine. If the data set has more records on both the low end and high end, the quality of analysis can be improved. We can be more certain about whether there is a significant correlation between a chemical component and the wine quality