Modeling and Predicting Wine Quality: Machine Learning Approach

Lumumba Wandera Victor

2024-05-02

# Modeling and predicting diabetes

## Introduction

The quality of the wine is influenced by a number of factors including but not limited to sensory characteristics and chemical composition. It is in the interest of the manufacture to determine what affects the quality of wine. In this study, we employ machine learning method to model and predict quality of the wine based on the physicochemical attributes. The data set used in this study is obtained from Kaggle website with variable including acidity, volatilize acidity, citric acid, residual sugar, chloride, free sulfuric dioxide, total sulfuric dioxide, density, pH, sulphates and alcohol content. The dependent variable for the study is the quality of the wine.

The application of machine learning in this study provides power tools analyzing and modeling wine quality. By doing so, this paper aims at evaluating the relationship between chemical composition and their effect on the quality of the wine. This algorithm will help determine the most appropriate machine learning technique that can help classify and model wine according their respective quality categories. In this study, wine quality will be categorized into three main categories, that is, “bad”, “average”, and “good”.

## Objectives

This study is guided by the following objectives \* To evaluate the performance of various machine learning algorithms, which include Naive Bayes, k-Nearest Neighbors (kNN), Hierachical clustering and K-Means Clustering in modeling classification of wine quality

* Assessing the effectiveness of the developed models in classifying and predicting the quality of the wine.

## Methodology

This study employed the use of secondary data (Red Wine Data) obtained from Kaggle website. (<http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv>) The dataset used comprised the chemical characteristics of the wine and the quality of the wine. The machine learning applied in this study include the following;

* Classification and Regression Tree (CART): CART is a decision tree algorithm that recursively splits the data into subsets based on the value of predictor variables. At each step, it chooses the variable that best splits the data, resulting in a tree-like structure where the leaves represent the predicted outcome.
* Random Forest: Random Forest is an ensemble machine learning method constructed from various decision trees to create one classification and prediction algorithm.
* k-Nearest Neighbors: This algorithm is a non-parametric machine learning algorithm thta classifies an individual based on the k-Nearest neighbors.
* Support Vector Machine (SVM): SVM is a supervised learning algorithm used for classification and regression tasks. It works by finding the hyper-plane that best separates the classes in the feature space. Using this algorithm, three kernels options are always specified, that is Sigmoid, Linear and Polynomial, however, in many instance, linear kernel has always outperformed the sigmoid and polynomial kernel.
* Naive Bayes Classifier: This is based on Bayes’ theorem to classify individual, holding the assumption that feature are independent

## Results

### Load the Required Libraries

library(ISLR2)  
library(MASS)  
library(caret)  
library(splines)  
library(pROC)  
library(rattle)  
library(recipes)  
library(lava)  
library(sjmisc)  
library(igraph)  
library(e1071)  
library(hardhat)  
library(ipred)  
library(caret)  
library(sjPlot)  
library(nnet)  
library(wakefield)  
library(kknn)  
library(dplyr)  
library(nnet)  
library(caTools)  
library(ROCR)  
library(stargazer)  
library(dplyr)  
library(nnet)  
library(caTools)  
library(ROCR)  
library(stargazer)  
library(ISLR)  
library(ISLR2)  
library(MASS)  
library(splines)  
library(splines2)  
library(pROC)  
library(ISLR)  
library(ISLR2)  
library(MASS)  
library(splines)  
library(splines2)  
library(pROC)  
library(randomForest)  
library(rpart)  
library(rpart.plot)  
library(rattle)  
library(ISLR2)  
library(MASS)  
library(splines)  
library(pROC)  
library(rattle)  
library(rpart)  
library(party)  
library(partykit)  
library(ggplot2)  
library(tune)  
library(TunePareto)  
library(ISLR2)  
library(MASS)  
library(caret)  
library(splines)  
library(pROC)  
library(rattle)  
library(ggplot2)  
library(devtools)  
library(predict3d)  
library(psych)  
library(dplyr)  
library(gtsummary)  
library(DescTools)  
library(nortest)   
library(lmtest)  
library(sandwich)  
library(sjmisc)  
library(ggplot2)  
library(stargazer)

### Load the Data

winequality <- read.csv("winequality-red.csv")  
attach(winequality)  
head(winequality,5)

fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
1 7.4 0.70 0.00 1.9 0.076  
2 7.8 0.88 0.00 2.6 0.098  
3 7.8 0.76 0.04 2.3 0.092  
4 11.2 0.28 0.56 1.9 0.075  
5 7.4 0.70 0.00 1.9 0.076  
 free.sulfur.dioxide total.sulfur.dioxide density pH sulphates alcohol  
1 11 34 0.9978 3.51 0.56 9.4  
2 25 67 0.9968 3.20 0.68 9.8  
3 15 54 0.9970 3.26 0.65 9.8  
4 17 60 0.9980 3.16 0.58 9.8  
5 11 34 0.9978 3.51 0.56 9.4  
 quality  
1 5  
2 5  
3 5  
4 6  
5 5

### Summary Statistics

options(scipen = 999)  
knitr::kable(  
 describeBy(winequality[]) %>% round(2)   
)

|  | vars | n | mean | sd | median | trimmed | mad | min | max | range | skew | kurtosis | se |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fixed.acidity | 1 | 1599 | 8.32 | 1.74 | 7.90 | 8.15 | 1.48 | 4.60 | 15.90 | 11.30 | 0.98 | 1.12 | 0.04 |
| volatile.acidity | 2 | 1599 | 0.53 | 0.18 | 0.52 | 0.52 | 0.18 | 0.12 | 1.58 | 1.46 | 0.67 | 1.21 | 0.00 |
| citric.acid | 3 | 1599 | 0.27 | 0.19 | 0.26 | 0.26 | 0.25 | 0.00 | 1.00 | 1.00 | 0.32 | -0.79 | 0.00 |
| residual.sugar | 4 | 1599 | 2.54 | 1.41 | 2.20 | 2.26 | 0.44 | 0.90 | 15.50 | 14.60 | 4.53 | 28.49 | 0.04 |
| chlorides | 5 | 1599 | 0.09 | 0.05 | 0.08 | 0.08 | 0.01 | 0.01 | 0.61 | 0.60 | 5.67 | 41.53 | 0.00 |
| free.sulfur.dioxide | 6 | 1599 | 15.87 | 10.46 | 14.00 | 14.58 | 10.38 | 1.00 | 72.00 | 71.00 | 1.25 | 2.01 | 0.26 |
| total.sulfur.dioxide | 7 | 1599 | 46.47 | 32.90 | 38.00 | 41.84 | 26.69 | 6.00 | 289.00 | 283.00 | 1.51 | 3.79 | 0.82 |
| density | 8 | 1599 | 1.00 | 0.00 | 1.00 | 1.00 | 0.00 | 0.99 | 1.00 | 0.01 | 0.07 | 0.92 | 0.00 |
| pH | 9 | 1599 | 3.31 | 0.15 | 3.31 | 3.31 | 0.15 | 2.74 | 4.01 | 1.27 | 0.19 | 0.80 | 0.00 |
| sulphates | 10 | 1599 | 0.66 | 0.17 | 0.62 | 0.64 | 0.12 | 0.33 | 2.00 | 1.67 | 2.42 | 11.66 | 0.00 |
| alcohol | 11 | 1599 | 10.42 | 1.07 | 10.20 | 10.31 | 1.04 | 8.40 | 14.90 | 6.50 | 0.86 | 0.19 | 0.03 |
| quality | 12 | 1599 | 5.64 | 0.81 | 6.00 | 5.59 | 1.48 | 3.00 | 8.00 | 5.00 | 0.22 | 0.29 | 0.02 |

The mean fixed acidity was 6.85 (SD = 0.84), volatile acidity was 0.28 (SD = 0.10), citric acid was 0.33 (SD = 0.12), residual sugar was 6.39 (SD = 5.07), chlorides were 0.05 (SD = 0.02), free sulfur dioxide was 35.31 (SD = 17.01), total sulfur dioxide was 138.36 (SD = 42.50), density was 0.99 (SD = 0.00), pH was 3.19 (SD = 0.15), sulphates were 0.49 (SD = 0.11), alcohol content was 10.51 (SD = 1.23), and wine quality was 5.88 (SD = 0.89).

### Label the Level the Dependent Variable

#Transforming Quality from an Integer to a Factor  
winequality$quality <- factor(winequality$quality, ordered = T)  
  
#Creating a new Factored Variable called 'Rating'  
  
winequality$rating <- ifelse(winequality$quality < 5, 'bad', ifelse(  
 winequality$quality < 7, 'average', 'good'))  
  
winequality$rating <- ordered(winequality$rating,  
 levels = c('bad', 'average', 'good'))  
head(winequality,5)

fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
1 7.4 0.70 0.00 1.9 0.076  
2 7.8 0.88 0.00 2.6 0.098  
3 7.8 0.76 0.04 2.3 0.092  
4 11.2 0.28 0.56 1.9 0.075  
5 7.4 0.70 0.00 1.9 0.076  
 free.sulfur.dioxide total.sulfur.dioxide density pH sulphates alcohol  
1 11 34 0.9978 3.51 0.56 9.4  
2 25 67 0.9968 3.20 0.68 9.8  
3 15 54 0.9970 3.26 0.65 9.8  
4 17 60 0.9980 3.16 0.58 9.8  
5 11 34 0.9978 3.51 0.56 9.4  
 quality rating  
1 5 average  
2 5 average  
3 5 average  
4 6 average  
5 5 average

# Model Estimation

## Create a Sample of 300 Observations

n <- 300  
random\_indices <- sample(1:nrow(winequality), size = n, replace = FALSE)  
winequality <- winequality[random\_indices, ]  
head(winequality,10)

fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
486 10.2 0.670 0.39 1.9 0.054  
304 7.4 0.670 0.12 1.6 0.186  
329 13.4 0.270 0.62 2.6 0.082  
1401 7.9 0.690 0.21 2.1 0.080  
1321 9.7 0.660 0.34 2.6 0.094  
1462 6.2 0.785 0.00 2.1 0.060  
776 7.2 1.000 0.00 3.0 0.102  
1186 7.0 0.430 0.30 2.0 0.085  
1357 6.1 0.340 0.25 1.8 0.084  
477 9.3 0.270 0.41 2.0 0.091  
 free.sulfur.dioxide total.sulfur.dioxide density pH sulphates alcohol  
486 6 17 0.99760 3.17 0.47 10.0  
304 5 21 0.99600 3.39 0.54 9.5  
329 6 21 1.00020 3.16 0.67 9.7  
1401 33 141 0.99620 3.25 0.51 9.9  
1321 12 88 0.99796 3.26 0.66 10.1  
1462 6 13 0.99664 3.59 0.61 10.0  
776 7 16 0.99586 3.43 0.46 10.0  
1186 6 39 0.99346 3.33 0.46 11.9  
1357 4 28 0.99464 3.36 0.44 10.1  
477 6 16 0.99800 3.28 0.70 9.7  
 quality rating  
486 5 average  
304 5 average  
329 6 average  
1401 5 average  
1321 5 average  
1462 4 bad  
776 5 average  
1186 6 average  
1357 5 average  
477 5 average

train\_model <- trainControl(method = "repeatedcv", number = 5, repeats=10)  
  
model.cart <- train(rating ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol,  
 data = winequality,  
 method = "rpart",  
 trControl = train\_model)

### Model Summary

model.cart

CART   
  
300 samples  
 11 predictor  
 3 classes: 'bad', 'average', 'good'   
  
No pre-processing  
Resampling: Cross-Validated (5 fold, repeated 10 times)   
Summary of sample sizes: 240, 241, 240, 240, 239, 240, ...   
Resampling results across tuning parameters:  
  
 cp Accuracy Kappa   
 0.03 0.8173798 0.1887057  
 0.04 0.8173798 0.1887057  
 0.06 0.8153631 0.1491500  
  
Accuracy was used to select the optimal model using the largest value.  
The final value used for the model was cp = 0.04.

### Confusion Matrix

confusionMatrix(predict(model.cart, winequality),   
 reference=winequality$rating, positive="good")

Confusion Matrix and Statistics  
  
 Reference  
Prediction bad average good  
 bad 0 0 0  
 average 11 244 18  
 good 0 6 21  
  
Overall Statistics  
   
 Accuracy : 0.8833   
 95% CI : (0.8415, 0.9174)  
 No Information Rate : 0.8333   
 P-Value [Acc > NIR] : 0.00995   
   
 Kappa : 0.4927   
   
 Mcnemar's Test P-Value : NA   
  
Statistics by Class:  
  
 Class: bad Class: average Class: good  
Sensitivity 0.00000 0.9760 0.5385  
Specificity 1.00000 0.4200 0.9770  
Pos Pred Value NaN 0.8938 0.7778  
Neg Pred Value 0.96333 0.7778 0.9341  
Prevalence 0.03667 0.8333 0.1300  
Detection Rate 0.00000 0.8133 0.0700  
Detection Prevalence 0.00000 0.9100 0.0900  
Balanced Accuracy 0.50000 0.6980 0.7577

The classification and regression model estimated shows that the model has an accuracy of approximately 75.33%. This shows that the model correctly classifies the wine qualities into their respective qualities (bad, average and good), 75.33% of the time.

### Variable Importance

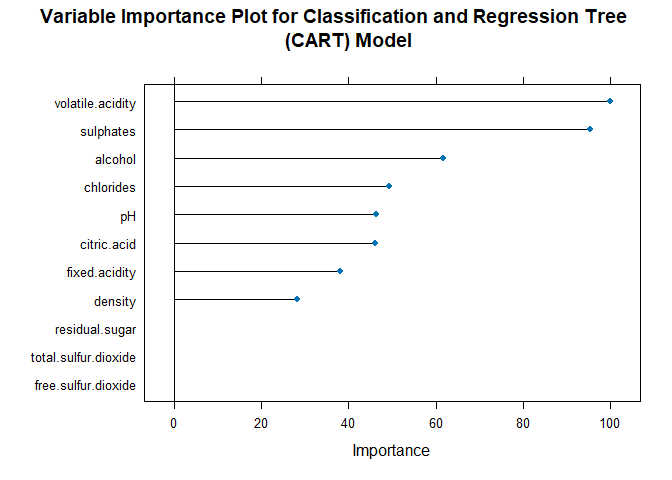
varImp(model.cart)

rpart variable importance  
  
 Overall  
volatile.acidity 100.00  
sulphates 95.41  
alcohol 61.68  
chlorides 49.29  
pH 46.37  
citric.acid 46.06  
fixed.acidity 38.14  
density 28.11  
residual.sugar 0.00  
total.sulfur.dioxide 0.00  
free.sulfur.dioxide 0.00

The results shows that the most important and significant variable in the classification and regression trees model developed are alcohol, chlorides, density, total sulfuric dioxide and residual sugar ans so on. From the results, alcohol has 100% importance, followed by density with 80.67%, chlorides with 53.96% and total sulfuric dioxide with 40.13%%. Consider the plot below to visualize the results above.

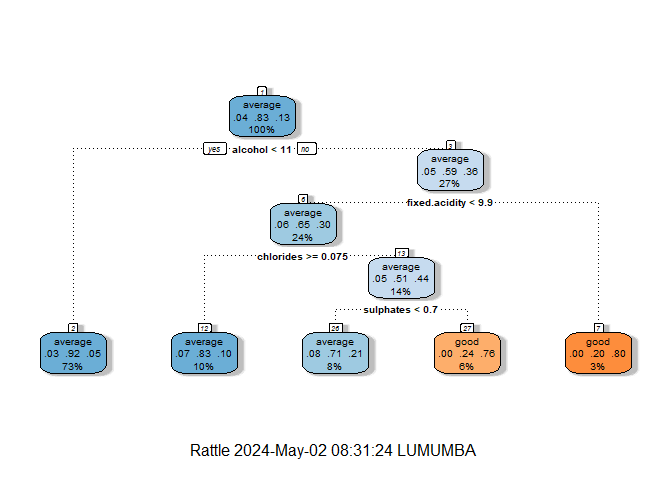
### Plot the Variable Importance

plot(varImp(model.cart), main = "Variable Importance Plot for Classification and Regression Tree   
 (CART) Model")



### Plot the Classification Model

fancyRpartPlot(model.cart$finalModel)



## Model Two: Random Forest

model.rf <- train(  
 rating ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol,  
 data = winequality,  
 method = "rf",  
 trControl = train\_model)  
model.rf

Random Forest   
  
300 samples  
 11 predictor  
 3 classes: 'bad', 'average', 'good'   
  
No pre-processing  
Resampling: Cross-Validated (5 fold, repeated 10 times)   
Summary of sample sizes: 240, 239, 240, 240, 241, 240, ...   
Resampling results across tuning parameters:  
  
 mtry Accuracy Kappa   
 2 0.8440486 0.2366232  
 6 0.8497437 0.3303788  
 11 0.8440876 0.3364278  
  
Accuracy was used to select the optimal model using the largest value.  
The final value used for the model was mtry = 6.

### Obtain the Confusion Matrix

confusionMatrix(predict(model.rf, winequality),   
 reference=winequality$rating, positive="good")

Confusion Matrix and Statistics  
  
 Reference  
Prediction bad average good  
 bad 11 0 0  
 average 0 250 0  
 good 0 0 39  
  
Overall Statistics  
   
 Accuracy : 1   
 95% CI : (0.9878, 1)   
 No Information Rate : 0.8333   
 P-Value [Acc > NIR] : < 0.00000000000000022  
   
 Kappa : 1   
   
 Mcnemar's Test P-Value : NA   
  
Statistics by Class:  
  
 Class: bad Class: average Class: good  
Sensitivity 1.00000 1.0000 1.00  
Specificity 1.00000 1.0000 1.00  
Pos Pred Value 1.00000 1.0000 1.00  
Neg Pred Value 1.00000 1.0000 1.00  
Prevalence 0.03667 0.8333 0.13  
Detection Rate 0.03667 0.8333 0.13  
Detection Prevalence 0.03667 0.8333 0.13  
Balanced Accuracy 1.00000 1.0000 1.00

The random forest model developed shows that the model has an accuracy of approximately 100%. This shows that the model classifies wine qualities correctly into their respective wine qualities as either bad, average, or good, 100% of the time.

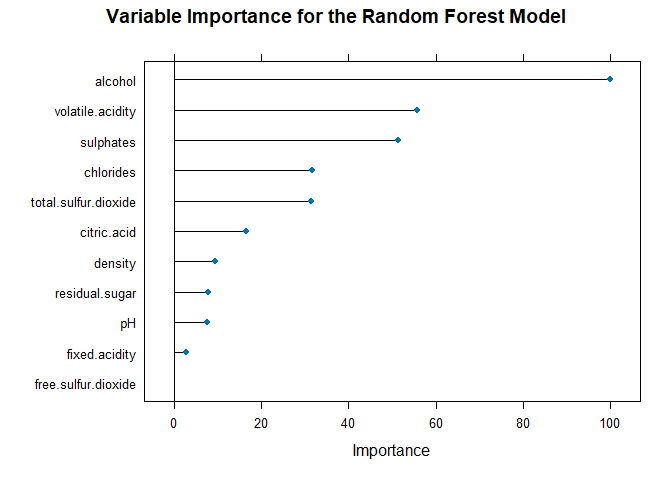
### Obtain variable importance

varImp(model.rf)

rf variable importance  
  
 Overall  
alcohol 100.000  
volatile.acidity 55.669  
sulphates 51.302  
chlorides 31.585  
total.sulfur.dioxide 31.343  
citric.acid 16.433  
density 9.444  
residual.sugar 7.692  
pH 7.445  
fixed.acidity 2.687  
free.sulfur.dioxide 0.000

The results above shows relative importance of each variable in helping predicting and classify wine qualities. From the results, alcohol has a higher relative importance of 100%, followed by density with 59.37%, chlorides with 43.36%, and total sulphuric with dioxide with 38.62% and so so. The figure below shows the relative importance of each variable in our model

plot(varImp(model.rf), main = "Variable Importance for the Random Forest Model")



## Model Three: k-Nearest Neighbors

trControl <- trainControl(method = "repeatedcv",number = 5,repeats = 10)  
model.knn <- train(rating ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol,  
 data = winequality,  
 method = "knn",  
 tuneLength = 5,  
 trControl = trControl,  
 preProc = c("center", "scale")  
)

### View the Final Model

model.knn

k-Nearest Neighbors   
  
300 samples  
 11 predictor  
 3 classes: 'bad', 'average', 'good'   
  
Pre-processing: centered (11), scaled (11)   
Resampling: Cross-Validated (5 fold, repeated 10 times)   
Summary of sample sizes: 239, 240, 240, 241, 240, 240, ...   
Resampling results across tuning parameters:  
  
 k Accuracy Kappa   
 5 0.8036141 0.13332119  
 7 0.8130485 0.13796466  
 9 0.8167156 0.11590268  
 11 0.8170485 0.07082575  
 13 0.8173928 0.04433794  
  
Accuracy was used to select the optimal model using the largest value.  
The final value used for the model was k = 13.

### Classification Accuracy

confusionMatrix(predict(model.knn, winequality),   
 reference=winequality$rating, positive="good")

Confusion Matrix and Statistics  
  
 Reference  
Prediction bad average good  
 bad 0 0 0  
 average 11 245 31  
 good 0 5 8  
  
Overall Statistics  
   
 Accuracy : 0.8433   
 95% CI : (0.7972, 0.8826)  
 No Information Rate : 0.8333   
 P-Value [Acc > NIR] : 0.3548   
   
 Kappa : 0.2053   
   
 Mcnemar's Test P-Value : NA   
  
Statistics by Class:  
  
 Class: bad Class: average Class: good  
Sensitivity 0.00000 0.9800 0.20513  
Specificity 1.00000 0.1600 0.98084  
Pos Pred Value NaN 0.8537 0.61538  
Neg Pred Value 0.96333 0.6154 0.89199  
Prevalence 0.03667 0.8333 0.13000  
Detection Rate 0.00000 0.8167 0.02667  
Detection Prevalence 0.00000 0.9567 0.04333  
Balanced Accuracy 0.50000 0.5700 0.59299

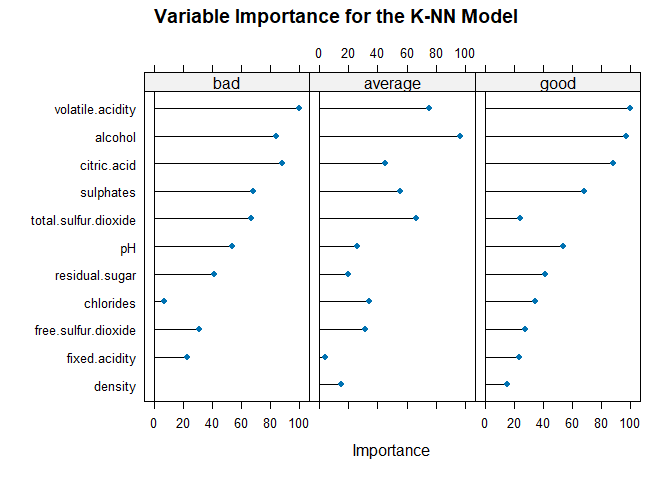
k-Nearest Neighbors performed slightly better classification and regression tree. However, the model performs slightly poorer in prediction as compared to random forest. From the above algorithm, the k-Nearest Neighbors accuracy is approximately 80% implying that the model correctly predict and classify wine quality in their correct wine qualities 80% of the time. The algorithm has a higher mis-classification error than that of random forest and a lower mis-classification error as compared to classification and regression tress (CART).

varImp(model.knn)

ROC curve variable importance  
  
 variables are sorted by maximum importance across the classes  
 bad average good  
volatile.acidity 100.00 75.586 100.00  
alcohol 84.11 97.123 97.12  
citric.acid 87.98 45.491 87.98  
sulphates 68.23 55.409 68.23  
total.sulfur.dioxide 66.85 66.848 24.00  
pH 54.06 25.922 54.06  
residual.sugar 41.61 19.760 41.61  
chlorides 7.10 34.155 34.16  
free.sulfur.dioxide 31.35 31.347 27.44  
fixed.acidity 23.14 4.003 23.14  
density 0.00 15.055 15.06

The relative importance of each variable in the classification and prediction of wine quality is as shown above, with free sulfuric dioxide as the overall important variable in the classification and prediction of bad, average and good wine, followed by pH in classifying bad wine, alcohol in classifying average wine quality and pH in classifying good quality wine. This is also indicated in the plot below

plot(varImp(model.knn), main = "Variable Importance for the K-NN Model")



## Model Four: Naive Bayes

trControl <- trainControl(method = "repeatedcv",number = 5,repeats = 10)  
model.nb <- train(rating ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol,  
 data = winequality,  
 method = "naive\_bayes",  
 tuneLength = 5,  
 trControl = trControl,  
 preProc = c("center", "scale")  
)

### View the Model

model.nb

Naive Bayes   
  
300 samples  
 11 predictor  
 3 classes: 'bad', 'average', 'good'   
  
Pre-processing: centered (11), scaled (11)   
Resampling: Cross-Validated (5 fold, repeated 10 times)   
Summary of sample sizes: 240, 240, 240, 239, 241, 241, ...   
Resampling results across tuning parameters:  
  
 usekernel Accuracy Kappa   
 FALSE 0.7250025 0.3088958  
 TRUE 0.7987210 0.3562510  
  
Tuning parameter 'laplace' was held constant at a value of 0  
Tuning  
 parameter 'adjust' was held constant at a value of 1  
Accuracy was used to select the optimal model using the largest value.  
The final values used for the model were laplace = 0, usekernel = TRUE  
 and adjust = 1.

### Prediction and Classification Accuracy

confusionMatrix(predict(model.nb, winequality),   
 reference=winequality$rating, positive="good")

Confusion Matrix and Statistics  
  
 Reference  
Prediction bad average good  
 bad 5 3 0  
 average 6 224 7  
 good 0 23 32  
  
Overall Statistics  
   
 Accuracy : 0.87   
 95% CI : (0.8266, 0.9059)  
 No Information Rate : 0.8333   
 P-Value [Acc > NIR] : 0.04857   
   
 Kappa : 0.5897   
   
 Mcnemar's Test P-Value : NA   
  
Statistics by Class:  
  
 Class: bad Class: average Class: good  
Sensitivity 0.45455 0.8960 0.8205  
Specificity 0.98962 0.7400 0.9119  
Pos Pred Value 0.62500 0.9451 0.5818  
Neg Pred Value 0.97945 0.5873 0.9714  
Prevalence 0.03667 0.8333 0.1300  
Detection Rate 0.01667 0.7467 0.1067  
Detection Prevalence 0.02667 0.7900 0.1833  
Balanced Accuracy 0.72208 0.8180 0.8662

Similar to the k-nearest neighbors, naive bayes performs slightly poor in the classification and prediction of the wine quality. From the model above,naive bayes correctly predict and classify wine qualities in their respective wine quality categories as either bad, average and good, 77% of the time, which lower as compared to random forest and above that of CART model.

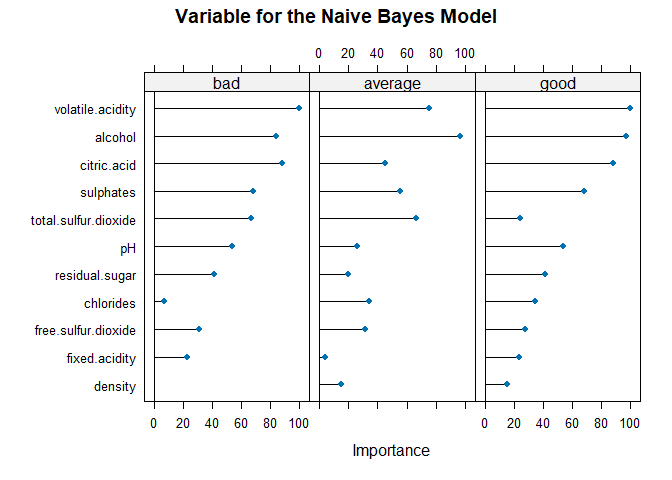
### Variable Importance

varImp(model.nb)

ROC curve variable importance  
  
 variables are sorted by maximum importance across the classes  
 bad average good  
volatile.acidity 100.00 75.586 100.00  
alcohol 84.11 97.123 97.12  
citric.acid 87.98 45.491 87.98  
sulphates 68.23 55.409 68.23  
total.sulfur.dioxide 66.85 66.848 24.00  
pH 54.06 25.922 54.06  
residual.sugar 41.61 19.760 41.61  
chlorides 7.10 34.155 34.16  
free.sulfur.dioxide 31.35 31.347 27.44  
fixed.acidity 23.14 4.003 23.14  
density 0.00 15.055 15.06

The relative importance of each variable in the classification and prediction of wine quality is as shown above, with free sulfuric dioxide as the overall important variable in the classification and prediction of bad 100%), average (100%) and good (100%) wine, followed by pH in classifying bad wine (69.12%), alcohol in classifying average (65.35%) wine quality and pH in classifying good quality wine (69.12%). This is also indicated in the plot below

plot(varImp(model.nb), main = "Variable for the Naive Bayes Model")



## Model Five: Support Vector Machine (SVM)

trControl <- trainControl(method = "repeatedcv",number = 5,repeats = 10)  
model.svm <- train(rating ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol,  
 data = winequality,  
 method = "svmLinear",  
 tuneLength = 5,  
 trControl = trControl,  
 preProc = c("center", "scale")  
)

### View the Model

model.svm

Support Vector Machines with Linear Kernel   
  
300 samples  
 11 predictor  
 3 classes: 'bad', 'average', 'good'   
  
Pre-processing: centered (11), scaled (11)   
Resampling: Cross-Validated (5 fold, repeated 10 times)   
Summary of sample sizes: 239, 240, 240, 241, 240, 240, ...   
Resampling results:  
  
 Accuracy Kappa   
 0.8190975 0.1303667  
  
Tuning parameter 'C' was held constant at a value of 1

### Prediction and Classification Accuracy

confusionMatrix(predict(model.svm, winequality),   
 reference=winequality$rating, positive="good")

Confusion Matrix and Statistics  
  
 Reference  
Prediction bad average good  
 bad 0 0 0  
 average 11 245 27  
 good 0 5 12  
  
Overall Statistics  
   
 Accuracy : 0.8567   
 95% CI : (0.8118, 0.8943)  
 No Information Rate : 0.8333   
 P-Value [Acc > NIR] : 0.1568   
   
 Kappa : 0.306   
   
 Mcnemar's Test P-Value : NA   
  
Statistics by Class:  
  
 Class: bad Class: average Class: good  
Sensitivity 0.00000 0.9800 0.30769  
Specificity 1.00000 0.2400 0.98084  
Pos Pred Value NaN 0.8657 0.70588  
Neg Pred Value 0.96333 0.7059 0.90459  
Prevalence 0.03667 0.8333 0.13000  
Detection Rate 0.00000 0.8167 0.04000  
Detection Prevalence 0.00000 0.9433 0.05667  
Balanced Accuracy 0.50000 0.6100 0.64427

The support vector machine model estimated above shows that the model has an accuracy of approximately 75.33%. This shows that the model correctly classifies the wine qualities into their respective qualities (bad, average and good), 78.67% of the time with a relatively higher mis-classification error of approximately 24.67%

### Variable Importance

varImp(model.svm)

ROC curve variable importance  
  
 variables are sorted by maximum importance across the classes  
 bad average good  
volatile.acidity 100.00 75.586 100.00  
alcohol 84.11 97.123 97.12  
citric.acid 87.98 45.491 87.98  
sulphates 68.23 55.409 68.23  
total.sulfur.dioxide 66.85 66.848 24.00  
pH 54.06 25.922 54.06  
residual.sugar 41.61 19.760 41.61  
chlorides 7.10 34.155 34.16  
free.sulfur.dioxide 31.35 31.347 27.44  
fixed.acidity 23.14 4.003 23.14  
density 0.00 15.055 15.06

free sulfuric dioxide has 100% importance in classifying and predicting bad quality wine and 100% importance in classifying and predicting average quality wine and finally 98.33% importance in classifying and predicting good quality. On the other hand, pH has 69.12% importance in classifying bad quality wine, 52% importance in classifying average quality wine and 60.12% importance in classifying good quality wine. The remaining variables and their importance in classifying wine qualities into their respective wine quality categories is as shown in the table above. Besides, the results can be visualized as shown below.

plot(varImp(model.svm), main = "Variable Importance for Support Vector Machine")

