Modeling and Predicting Wine Quality: Machine Learning Approach

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)
```

```r
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

```r
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

```r
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
1442           7.4            0.785        0.19            5.2     0.094
522            7.6            0.410        0.49            2.0     0.088
1183          10.2            0.400        0.40            2.5     0.068
1176           6.5            0.610        0.00            2.2     0.095
165            7.3            0.590        0.26            7.2     0.070
512           10.0            0.590        0.31            2.2     0.090
1035           8.9            0.745        0.18            2.5     0.077
1252           7.5            0.580        0.14            2.2     0.077
469           11.4            0.360        0.69            2.1     0.090
160            6.8            0.600        0.18            1.9     0.079
     free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates   alcohol
1442                  19                   98 0.99713 3.16      0.52  9.566667
522                   16                   43 0.99800 3.48      0.64  9.100000
1183                  41                   54 0.99754 3.38      0.86 10.500000
1176                  48                   59 0.99541 3.61      0.70 11.500000
165                   35                  121 0.99810 3.37      0.49  9.400000
512                   26                   62 0.99940 3.18      0.63 10.200000
1035                  15                   48 0.99739 3.20      0.47  9.700000
1252                  27                   60 0.99630 3.28      0.59  9.800000
469                    6                   21 1.00000 3.17      0.62  9.200000
160                   18                   86 0.99680 3.59      0.57  9.300000
     quality  rating
1442       6 average
522        5 average
1183       6 average
1176       6 average
165        5 average
512        6 average
1035       6 average
1252       5 average
```

```
469        6 average
160        6 average
```

```
train_model <- trainControl(method = "repeatedcv", number = 5, repeats=10)

model.cart <- train(rating ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.su
  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, 240, 239, 241, 240, 241, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.01754386  0.8118271  0.2692043
  0.07017544  0.8287916  0.2717075
  0.09649123  0.8157007  0.1993861

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.07017544.
```

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  15     243   31
   good      0       0   11

Overall Statistics

               Accuracy : 0.8467
```

```
               95% CI : (0.8008, 0.8855)
    No Information Rate : 0.81
    P-Value [Acc > NIR] : 0.05832

                  Kappa : 0.2854

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bad Class: average Class: good
Sensitivity                0.00         1.0000     0.26190
Specificity                1.00         0.1930     1.00000
Pos Pred Value              NaN         0.8408     1.00000
Neg Pred Value             0.95         1.0000     0.89273
Prevalence                 0.05         0.8100     0.14000
Detection Rate             0.00         0.8100     0.03667
Detection Prevalence       0.00         0.9633     0.03667
Balanced Accuracy          0.50         0.5965     0.63095
```

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)
```

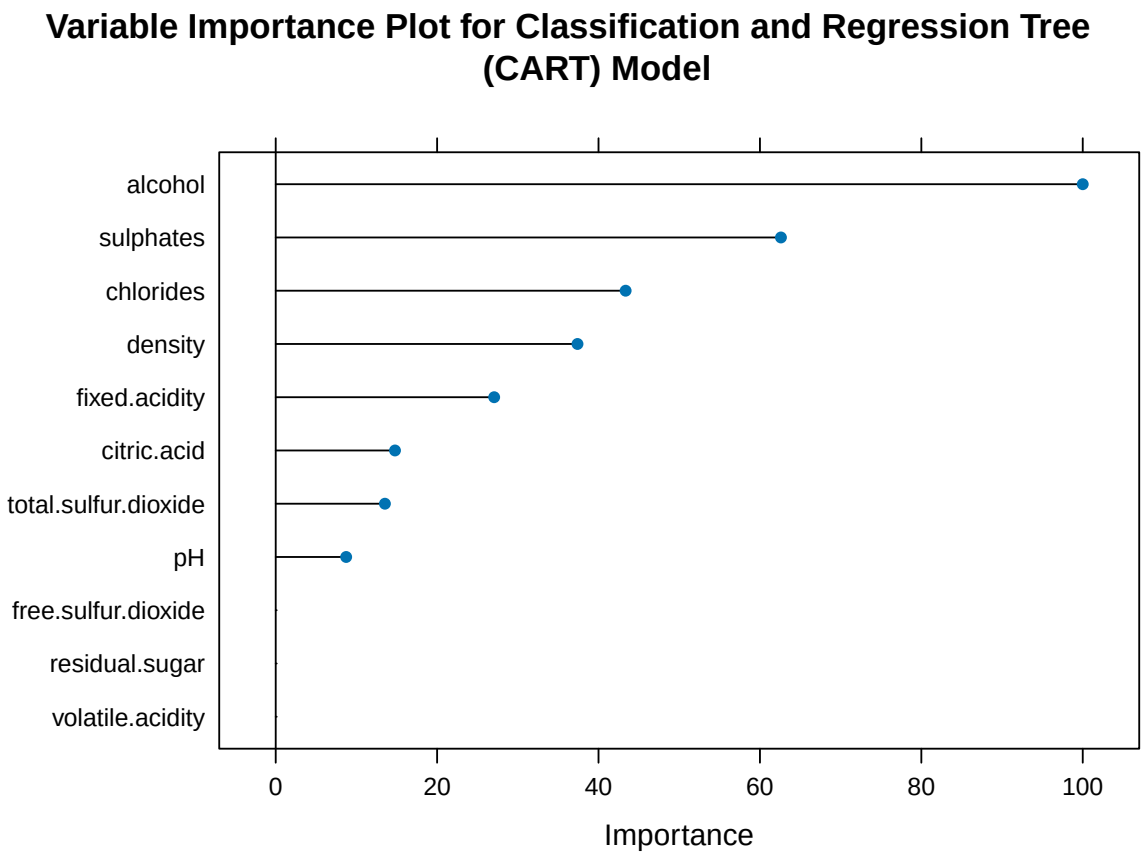
```
rpart variable importance

                    Overall
alcohol              100.00
sulphates             62.60
chlorides             43.36
density               37.39
fixed.acidity         27.07
citric.acid           14.78
total.sulfur.dioxide  13.53
pH                     8.73
free.sulfur.dioxide    0.00
residual.sugar         0.00
volatile.acidity       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.

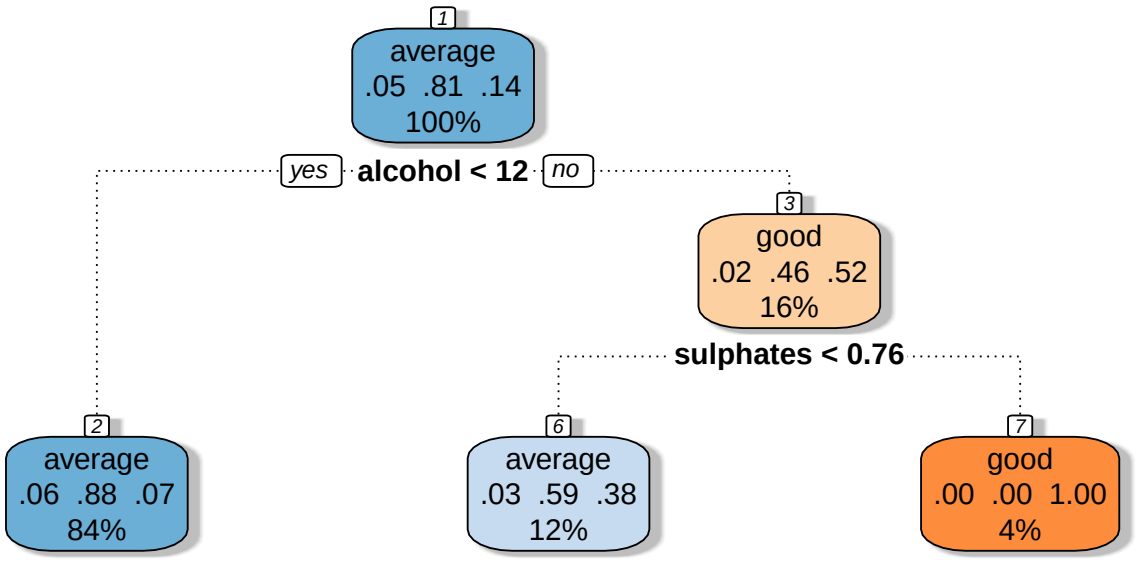
Plot the Variable Importance

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



Plot the Classification Model

```r
fancyRpartPlot(model.cart$finalModel)
```



Rattle 2024−May−02 08:32:40 LUMUMBA

Model Two: Random Forest

```
model.rf <- train(
 rating ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total
  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, 240, 240, 241, 239, 240, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.8220395  0.2390306
   6    0.8236896  0.3008550
```

```
  11    0.8197392  0.3045753

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      15       0    0
   average   0     243    0
   good      0       0   42

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9878, 1)
    No Information Rate : 0.81
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bad Class: average Class: good
Sensitivity                1.00           1.00        1.00
Specificity                1.00           1.00        1.00
Pos Pred Value             1.00           1.00        1.00
Neg Pred Value             1.00           1.00        1.00
Prevalence                 0.05           0.81        0.14
Detection Rate             0.05           0.81        0.14
Detection Prevalence       0.05           0.81        0.14
Balanced Accuracy          1.00           1.00        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

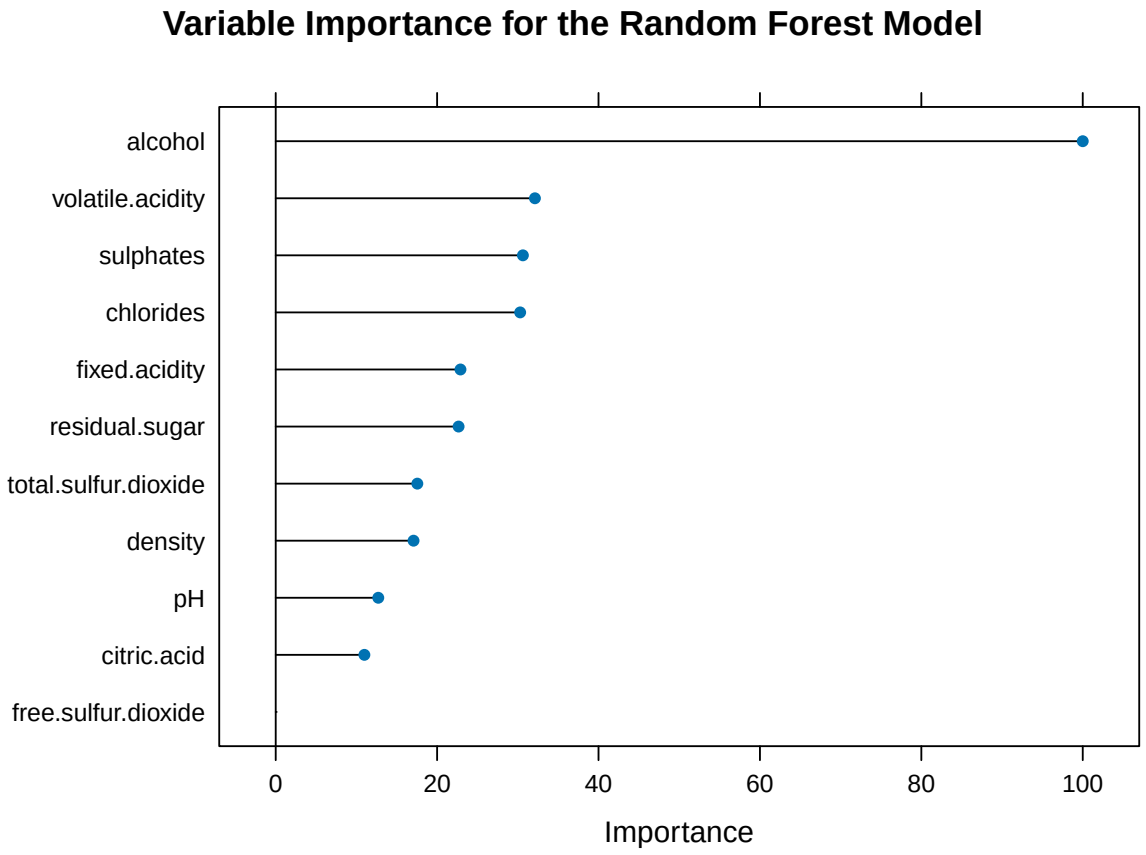
```
varImp(model.rf)
```

```
rf variable importance

                     Overall
alcohol               100.00
volatile.acidity       32.12
sulphates              30.63
chlorides              30.29
fixed.acidity          22.89
residual.sugar         22.66
total.sulfur.dioxide   17.55
density                17.08
pH                     12.71
citric.acid            10.99
free.sulfur.dioxide     0.00
```

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.su
  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, 239, 241, 241, 240, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.7966610  0.19748718
   7  0.8062902  0.17275361
   9  0.8043902  0.11540579
  11  0.8083791  0.09843255
  13  0.8107405  0.08953840

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  15     239   35
   good      0       4    7
```

```
Overall Statistics

               Accuracy : 0.82
                 95% CI : (0.7718, 0.8618)
    No Information Rate : 0.81
    P-Value [Acc > NIR] : 0.3614

                  Kappa : 0.1611

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bad Class: average Class: good
Sensitivity                0.00         0.9835     0.16667
Specificity                1.00         0.1228     0.98450
Pos Pred Value              NaN         0.8270     0.63636
Neg Pred Value             0.95         0.6364     0.87889
Prevalence                 0.05         0.8100     0.14000
Detection Rate             0.00         0.7967     0.02333
Detection Prevalence       0.00         0.9633     0.03667
Balanced Accuracy          0.50         0.5532     0.57558
```

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).

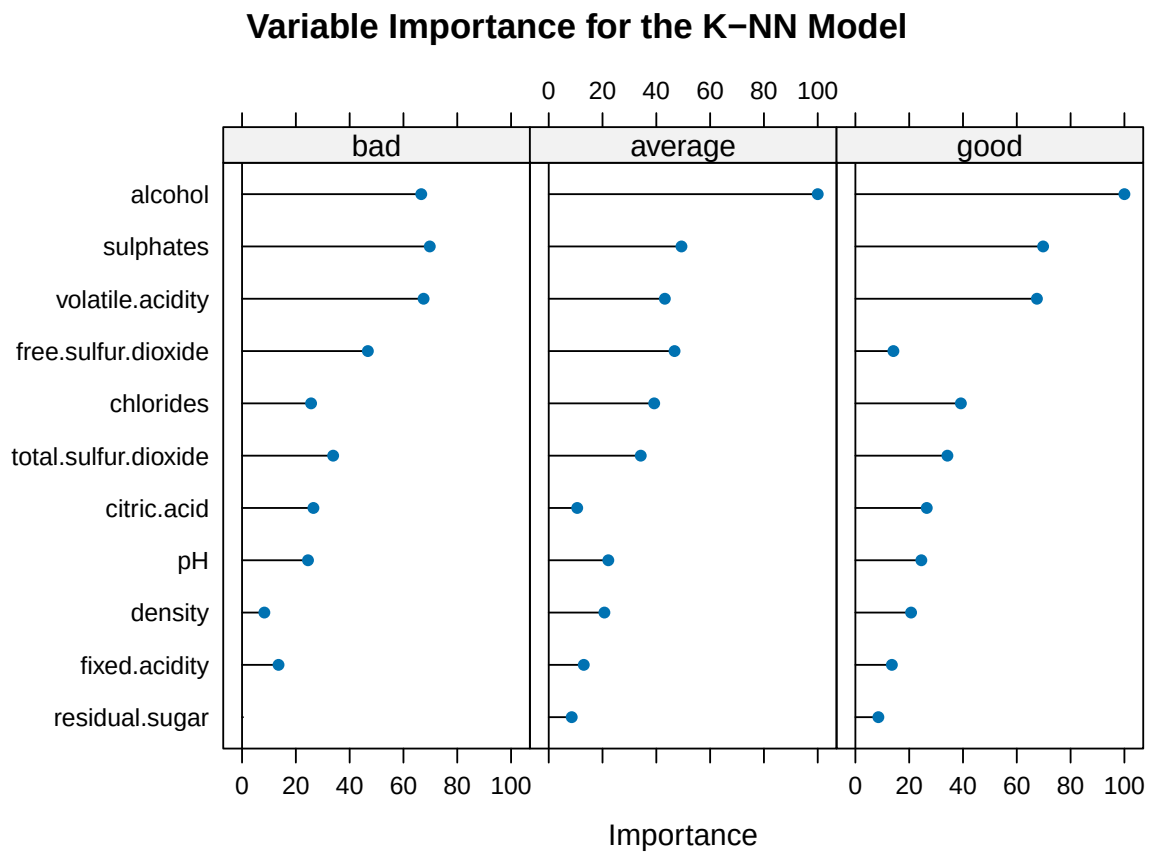
```
varImp(model.knn)
```

```
ROC curve variable importance

  variables are sorted by maximum importance across the classes
                        bad average    good
alcohol              66.613 100.000 100.000
sulphates            69.785  49.340  69.785
volatile.acidity     67.478  43.163  67.478
free.sulfur.dioxide  46.769  46.769  14.130
chlorides            25.665  39.211  39.211
total.sulfur.dioxide 33.860  34.209  34.209
citric.acid          26.530  10.588  26.530
pH                   24.511  22.147  24.511
density               8.292  20.698  20.698
fixed.acidity        13.553  13.026  13.553
residual.sugar        0.000   8.541   8.541
```

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.sul
  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, 241, 240, 239, 241, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.7212752  0.2502993
   TRUE      0.7626862  0.2980370

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       9       3    2
   average   6     216   11
   good      0      24   29

Overall Statistics

               Accuracy : 0.8467
                 95% CI : (0.8008, 0.8855)
    No Information Rate : 0.81
    P-Value [Acc > NIR] : 0.05832

                  Kappa : 0.554

 Mcnemar's Test P-Value : 0.04969

Statistics by Class:

                     Class: bad Class: average Class: good
Sensitivity             0.60000         0.8889     0.69048
Specificity             0.98246         0.7018     0.90698
Pos Pred Value          0.64286         0.9270     0.54717
Neg Pred Value          0.97902         0.5970     0.94737
Prevalence              0.05000         0.8100     0.14000
Detection Rate          0.03000         0.7200     0.09667
```

```
Detection Prevalence    0.04667         0.7767     0.17667
Balanced Accuracy       0.79123         0.7953     0.79873
```

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

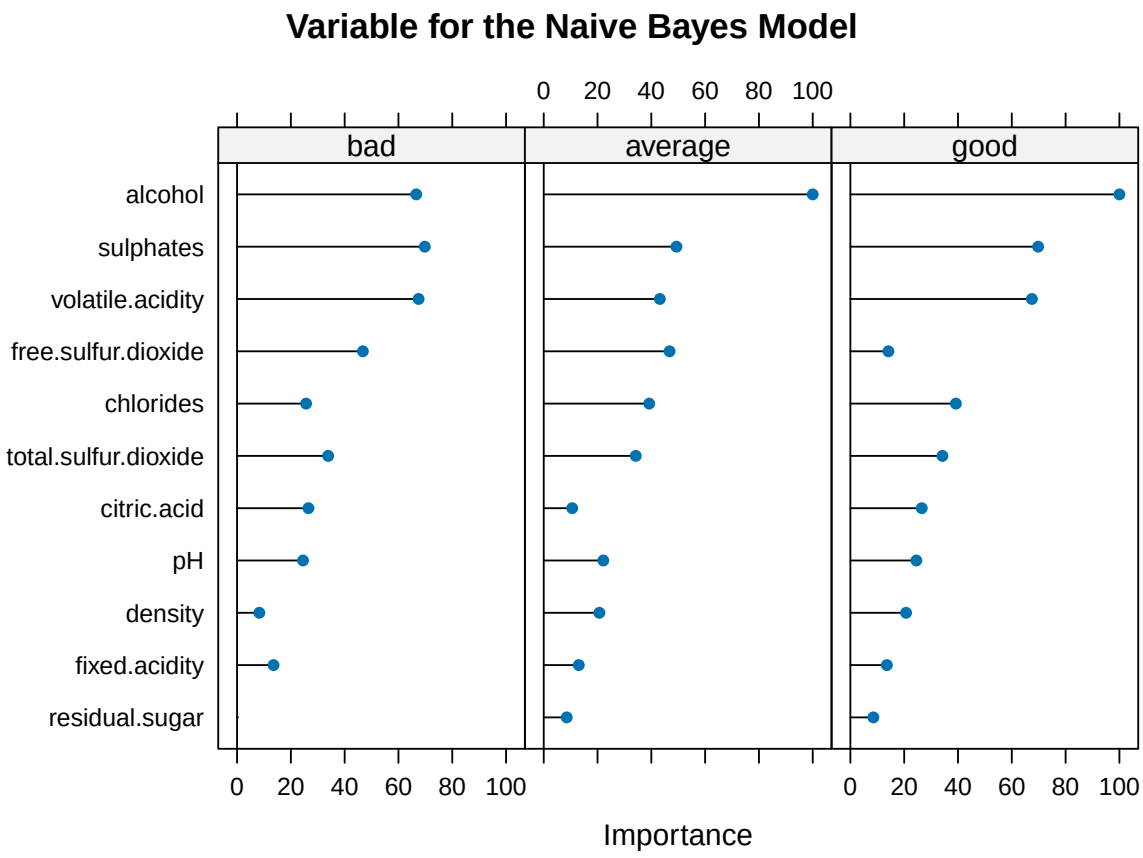
```
varImp(model.nb)
```

```
ROC curve variable importance

  variables are sorted by maximum importance across the classes
                        bad average    good
alcohol              66.613 100.000 100.000
sulphates            69.785  49.340  69.785
volatile.acidity     67.478  43.163  67.478
free.sulfur.dioxide  46.769  46.769  14.130
chlorides            25.665  39.211  39.211
total.sulfur.dioxide 33.860  34.209  34.209
citric.acid          26.530  10.588  26.530
pH                   24.511  22.147  24.511
density               8.292  20.698  20.698
fixed.acidity        13.553  13.026  13.553
residual.sugar        0.000   8.541   8.541
```

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.su
  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: 241, 241, 239, 240, 239, 240, ...
Resampling results:

  Accuracy   Kappa
  0.8064622  0.2203108

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       1       1    0
   average  14     236   28
   good      0       6   14

Overall Statistics

               Accuracy : 0.8367
                 95% CI : (0.7899, 0.8767)
    No Information Rate : 0.81
    P-Value [Acc > NIR] : 0.134

                  Kappa : 0.3187

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bad Class: average Class: good
Sensitivity            0.066667         0.9712     0.33333
Specificity            0.996491         0.2632     0.97674
Pos Pred Value         0.500000         0.8489     0.70000
Neg Pred Value         0.953020         0.6818     0.90000
Prevalence             0.050000         0.8100     0.14000
Detection Rate         0.003333         0.7867     0.04667
Detection Prevalence   0.006667         0.9267     0.06667
Balanced Accuracy      0.531579         0.6172     0.65504
```

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

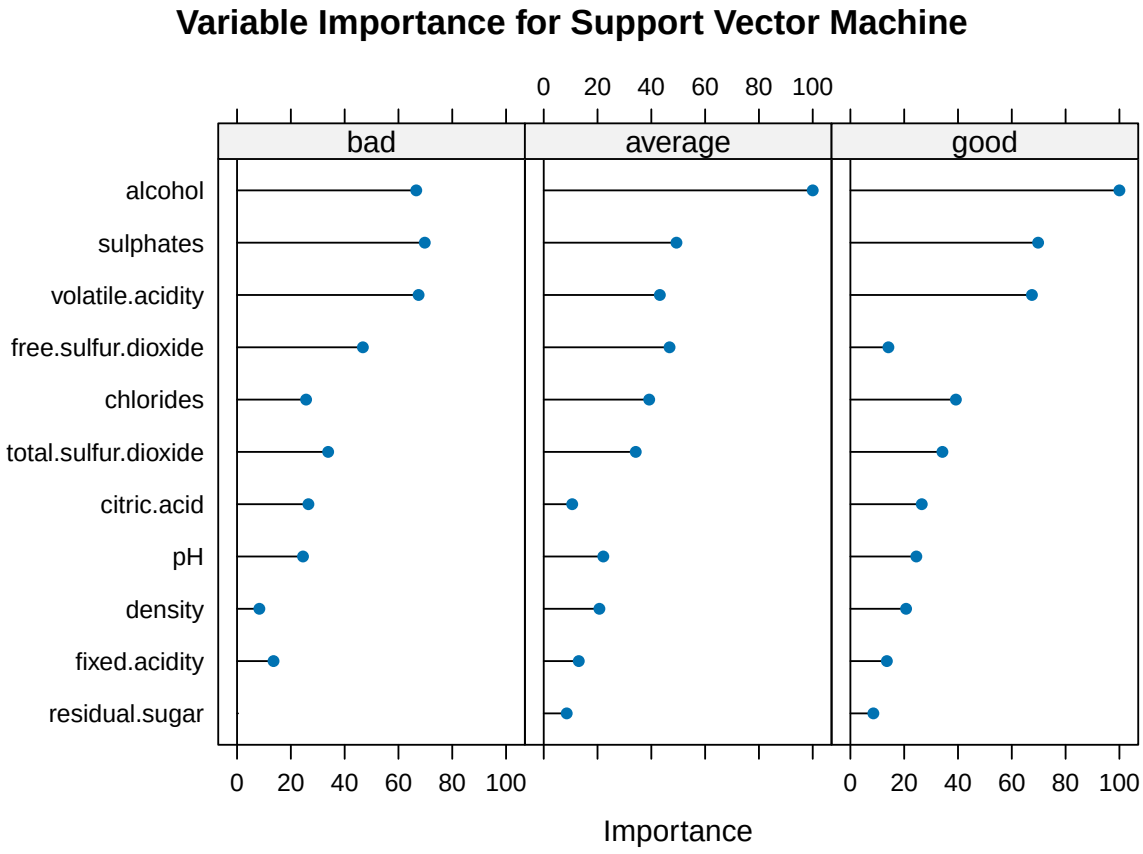
```
varImp(model.svm)
```

```
ROC curve variable importance

  variables are sorted by maximum importance across the classes
                        bad average    good
alcohol              66.613 100.000 100.000
sulphates            69.785  49.340  69.785
volatile.acidity     67.478  43.163  67.478
free.sulfur.dioxide  46.769  46.769  14.130
chlorides            25.665  39.211  39.211
total.sulfur.dioxide 33.860  34.209  34.209
citric.acid          26.530  10.588  26.530
pH                   24.511  22.147  24.511
density               8.292  20.698  20.698
fixed.acidity        13.553  13.026  13.553
residual.sugar        0.000   8.541   8.541
```

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")
```


Variable Importance for Support Vector Machine
bad
average
good
alcohol
sulphates
volatile.acidity
free.sulfur.dioxide
chlorides
total.sulfur.dioxide
citric.acid
pH
density
fixed.acidity
residual.sugar
0
20
40
60
80
100
Importance