exploratory-analysis

Rishika Cherivirala

2025-04-23

R Markdown

```
redwine_df <- read.csv("data/winequality-red.csv", sep = ";")
whitewine_df <- read.csv("data/winequality-white.csv", sep = ";")
names_df <- readLines("data/winequality.names")
```

```
summary(redwine_df)
```

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

```
summary(whitewine_df)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
##  Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
##  Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900
##  Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800
```

```
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  2.00      Min.   :  9.0        Min.   :0.9871
##  1st Qu.:0.03600   1st Qu.: 23.00      1st Qu.:108.0        1st Qu.:0.9917
##  Median :0.04300   Median : 34.00      Median :134.0        Median :0.9937
##  Mean   :0.04577   Mean   : 35.31      Mean   :138.4        Mean   :0.9940
##  3rd Qu.:0.05000   3rd Qu.: 46.00      3rd Qu.:167.0        3rd Qu.:0.9961
##  Max.   :0.34600   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.090   1st Qu.:0.4100   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.180   Median :0.4700   Median :10.40   Median :6.000
##  Mean   :3.188   Mean   :0.4898   Mean   :10.51   Mean   :5.878
##  3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40   3rd Qu.:6.000
##  Max.   :3.820   Max.   :1.0800   Max.   :14.20   Max.   :9.000
```

Red Wine Analysis

Logistic Regression

```
#  If the quality is greater than 7, it gets a label of 1 (good), otherwise 0 (bad)
redwine_df <- redwine_df %>%
  mutate(quality_label = ifelse(quality >= 7, 1, 0)) %>%
  mutate(quality_label = as.factor(quality_label))

# Split data into train and test
set.seed(2950)
train_index <- sample(1:nrow(redwine_df), 0.8 * nrow(redwine_df))
train <- redwine_df[train_index, ]
test <- redwine_df[-train_index, ]

# Fitting logistic regression model
log_model <- glm(quality_label ~ . - quality, data = train, family = "binomial")

# Summary of model
summary(log_model)
```

```
##
## Call:
## glm(formula = quality_label ~ . - quality, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9361  -0.4222  -0.2268  -0.1087   3.0652
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.786e+02  1.184e+02   1.508 0.131470
## fixed.acidity         2.270e-01  1.378e-01   1.648 0.099414 .
## volatile.acidity     -2.581e+00  8.775e-01  -2.941 0.003267 **
## citric.acid           6.350e-01  9.140e-01   0.695 0.487247
```

```
## residual.sugar        2.803e-01  8.393e-02   3.340 0.000838 ***
## chlorides            -8.196e+00  3.339e+00  -2.455 0.014097 *
## free.sulfur.dioxide   2.212e-02  1.401e-02   1.578 0.114455
## total.sulfur.dioxide -2.684e-02  6.363e-03  -4.218 2.46e-05 ***
## density              -1.929e+02  1.210e+02  -1.594 0.111020
## pH                    1.036e-01  1.117e+00   0.093 0.926102
## sulphates             3.926e+00  6.060e-01   6.478 9.32e-11 ***
## alcohol               7.692e-01  1.446e-01   5.318 1.05e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1042.92  on 1278  degrees of freedom
## Residual deviance:  709.27  on 1267  degrees of freedom
## AIC: 733.27
##
## Number of Fisher Scoring iterations: 6
```

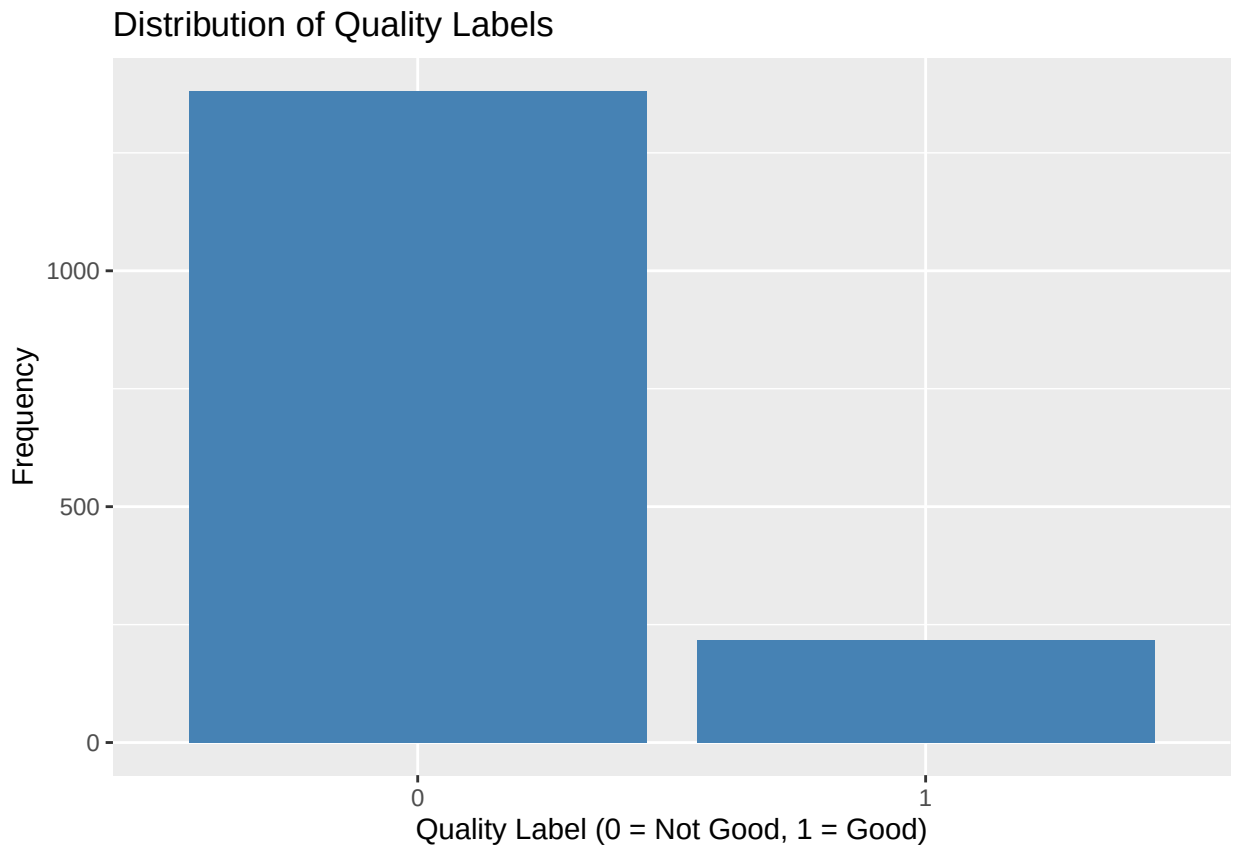
```r
# Predicting on test data
pred_probs <- predict(log_model, test, type = "response")

# Evaluating model
pred_labels <- ifelse(pred_probs > 0.5, 1, 0) %>% as.factor()

confusionMatrix(pred_labels, test$quality_label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 271  23
##          1  13  13
##
##                Accuracy : 0.8875
##                  95% CI : (0.8477, 0.92)
##     No Information Rate : 0.8875
##     P-Value [Acc > NIR] : 0.5442
##
##                   Kappa : 0.3589
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 0.9542
##             Specificity : 0.3611
##          Pos Pred Value : 0.9218
##          Neg Pred Value : 0.5000
##              Prevalence : 0.8875
##          Detection Rate : 0.8469
##    Detection Prevalence : 0.9187
##       Balanced Accuracy : 0.6577
##
##        'Positive' Class : 0
##
```

```r
table(redwine_df$quality_label)
```

```
##
##    0    1
## 1382  217
```

```r
ggplot(redwine_df, aes(x = quality_label)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Distribution of Quality Labels",
       x = "Quality Label (0 = Not Good, 1 = Good)",
       y = "Frequency")
```



Volitale acidity, residual sugar, chlorides, total sulfur dioxide, sulfates, and alcohol seem to be statistically significant since their p-values are less than 0.05. The model has an accuracy of Accuracy : 0.8589.

White Wine Analysis

Logistic Regression

```r
# If the quality is greater than 7, it gets a label of 1 (good), otherwise 0 (bad)
whitewine_df <- whitewine_df %>%
  mutate(quality_label = ifelse(quality >= 7, 1, 0)) %>%
```

```r
  mutate(quality_label = as.factor(quality_label))

# Split data into train and test
set.seed(2950)
train_index <- sample(1:nrow(whitewine_df), 0.8 * nrow(whitewine_df))
train <- whitewine_df[train_index, ]
test <- whitewine_df[-train_index, ]

# Fitting logistic regression model
log_model <- glm(quality_label ~ . - quality, data = train, family = "binomial")

# Summary of model
summary(log_model)
```

```
##
## Call:
## glm(formula = quality_label ~ . - quality, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3092  -0.6701  -0.4066  -0.1772   2.8297
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           6.624e+02  1.048e+02   6.320 2.62e-10 ***
## fixed.acidity         6.133e-01  1.010e-01   6.075 1.24e-09 ***
## volatile.acidity     -3.966e+00  5.557e-01  -7.137 9.56e-13 ***
## citric.acid          -7.841e-01  4.439e-01  -1.766  0.07734 .
## residual.sugar        3.095e-01  3.978e-02   7.780 7.27e-15 ***
## chlorides            -1.069e+01  4.112e+00  -2.600  0.00934 **
## free.sulfur.dioxide   8.234e-03  3.473e-03   2.371  0.01776 *
## total.sulfur.dioxide -1.845e-04  1.672e-03  -0.110  0.91211
## density              -6.867e+02  1.062e+02  -6.464 1.02e-10 ***
## pH                    3.544e+00  4.760e-01   7.445 9.68e-14 ***
## sulphates             2.319e+00  3.872e-01   5.990 2.10e-09 ***
## alcohol               1.380e-01  1.271e-01   1.086  0.27741
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4075.1  on 3917  degrees of freedom
## Residual deviance: 3293.8  on 3906  degrees of freedom
## AIC: 3317.8
##
## Number of Fisher Scoring iterations: 5
```

```r
# Predicting on test data
pred_probs <- predict(log_model, test, type = "response")

# Evaluating model
pred_labels <- ifelse(pred_probs > 0.5, 1, 0) %>% as.factor()
```

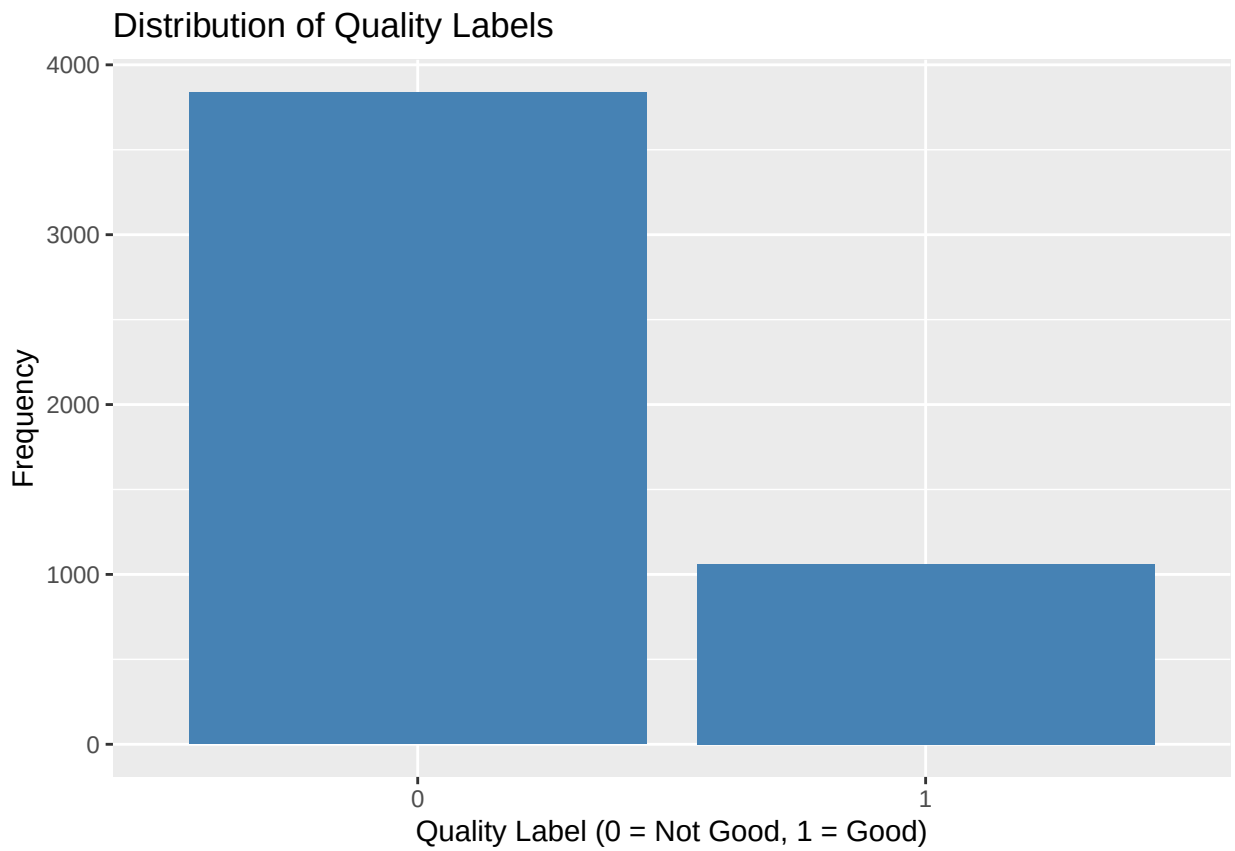
```r
confusionMatrix(pred_labels, test$quality_label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 709 152
##          1  52  67
##
##                Accuracy : 0.7918
##                  95% CI : (0.765, 0.8169)
##     No Information Rate : 0.7765
##     P-Value [Acc > NIR] : 0.1327
##
##                   Kappa : 0.2837
##
##  Mcnemar's Test P-Value : 4.167e-12
##
##             Sensitivity : 0.9317
##             Specificity : 0.3059
##          Pos Pred Value : 0.8235
##          Neg Pred Value : 0.5630
##              Prevalence : 0.7765
##          Detection Rate : 0.7235
##    Detection Prevalence : 0.8786
##       Balanced Accuracy : 0.6188
##
##        'Positive' Class : 0
##
```

```r
table(whitewine_df$quality_label)
```

```
##
##    0    1
## 3838 1060
```

```r
ggplot(whitewine_df, aes(x = quality_label)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Distribution of Quality Labels",
       x = "Quality Label (0 = Not Good, 1 = Good)",
       y = "Frequency")
```



Fixed Acidity, Volitale Acidity, Citric Acid, residual sugars, free sulfur dioxide, density, pH, sulphates, alcohol all seem to be statistically significant. The model seems to have an accuracy of 0.7819.

Red & White Wine Analysis

```
redwine_df$wine_type <- 1
whitewine_df$wine_type <- 0

# Combine the red and white wine datasets into one
wine_df <- rbind(redwine_df, whitewine_df)

summary(wine_df)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
```

```
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality      quality_label
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000   0:5220
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000   1:1277
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
##    wine_type
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2461
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

Random Forest

```r
wine_df$quality_numeric <- as.numeric(as.character(wine_df$quality))
wine_df <- wine_df %>%
  filter(!is.na(quality_numeric))

# If quality is greater than 7, label as 1 (good), otherwise 0 (bad)
wine_df <- wine_df %>%
  mutate(quality_label = ifelse(quality_numeric >= 7, 1, 0)) %>%
  mutate(quality_label = as.factor(quality_label))

# Convert quality column to factor for classification
wine_df$quality <- as.factor(wine_df$quality)

# Split the data into training and testing sets
set.seed(2950)
train_index <- sample(1:nrow(wine_df), 0.8 * nrow(wine_df))
train_data <- wine_df[train_index, ]
test_data <- wine_df[-train_index, ]

# Train a random forest model
rf_model <- randomForest(quality ~ . - quality_label - quality_numeric, data = train_data)
print(rf_model)
```

```
##
## Call:
##  randomForest(formula = quality ~ . - quality_label - quality_numeric,      data = train_data)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 30.52%
## Confusion matrix:
```

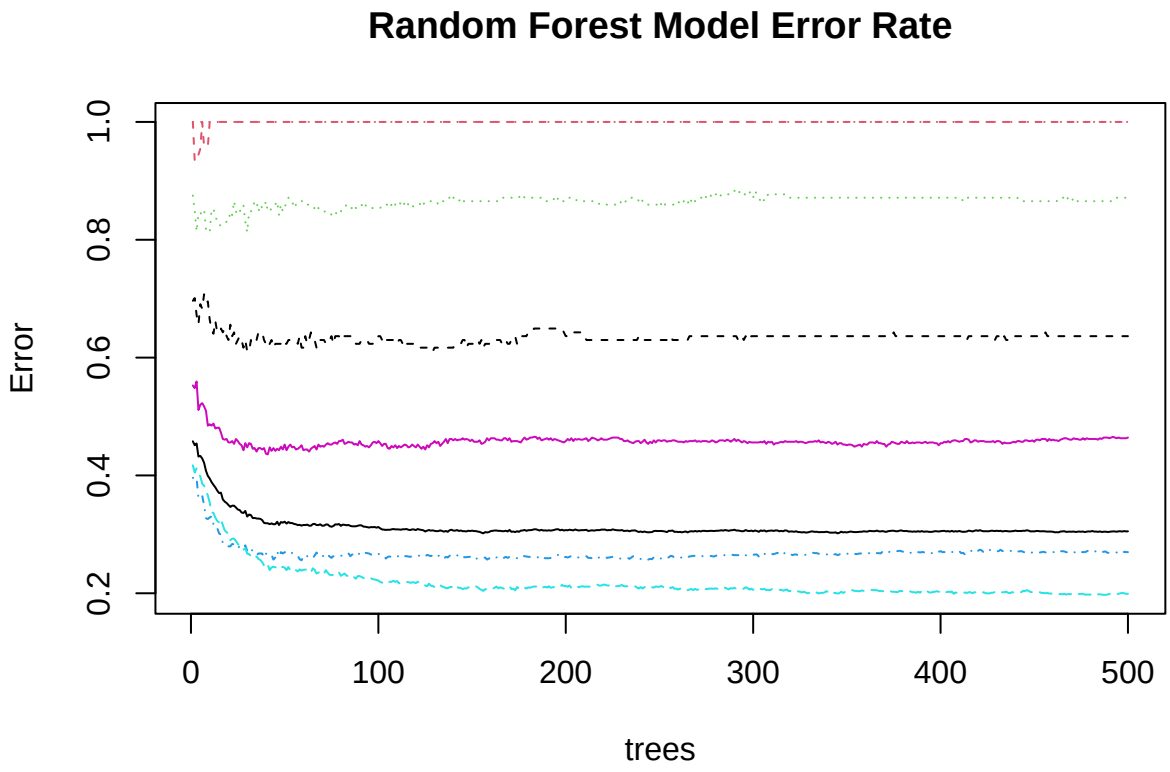
```
##   3  4    5    6   7  8 9 class.error
## 3 0  1   12   12   0  0 0   1.0000000
## 4 1 22   95   51   2  0 0   0.8713450
## 5 0  6 1269  450  13  0 0   0.2698504
## 6 0  3  325 1814 121  2 0   0.1991170
## 7 0  0   21  362 450  7 0   0.4642857
## 8 0  0    0   62  36 56 0   0.6363636
## 9 0  0    0    1   3  0 0   1.0000000
```

```r
# Predict on the test data
predictions <- predict(rf_model, test_data)

# Evaluating model
confusion_matrix <- confusionMatrix(predictions, test_data$quality)
print(confusion_matrix)
```

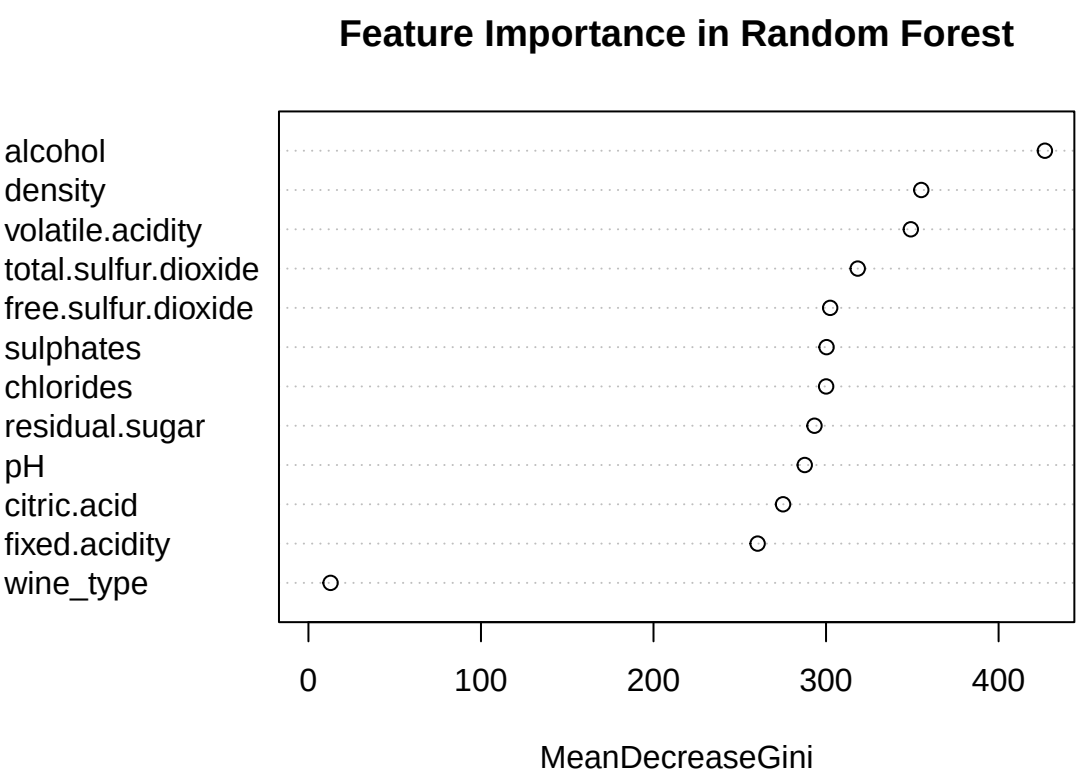
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   3   4   5   6   7   8   9
##          3   0   0   0   0   0   0   0
##          4   0   4   0   1   0   0   0
##          5   3  25 286  88   6   1   0
##          6   2  15 113 454  95  16   1
##          7   0   1   1  27 134   8   0
##          8   0   0   0   1   4  14   0
##          9   0   0   0   0   0   0   0
##
## Overall Statistics
##
##                Accuracy : 0.6862
##                  95% CI : (0.6601, 0.7113)
##     No Information Rate : 0.4392
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5121
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
## Sensitivity          0.000000 0.088889   0.7150   0.7951   0.5607  0.35897
## Specificity          1.000000 0.999203   0.8633   0.6680   0.9651  0.99603
## Pos Pred Value            NaN 0.800000   0.6993   0.6523   0.7836  0.73684
## Neg Pred Value       0.996154 0.968340   0.8721   0.8063   0.9070  0.98048
## Prevalence           0.003846 0.034615   0.3077   0.4392   0.1838  0.03000
## Detection Rate       0.000000 0.003077   0.2200   0.3492   0.1031  0.01077
## Detection Prevalence 0.000000 0.003846   0.3146   0.5354   0.1315  0.01462
## Balanced Accuracy    0.500000 0.544046   0.7892   0.7316   0.7629  0.67750
##                      Class: 9
## Sensitivity          0.0000000
## Specificity          1.0000000
## Pos Pred Value             NaN
```

```
## Neg Pred Value       0.9992308
## Prevalence           0.0007692
## Detection Rate       0.0000000
## Detection Prevalence 0.0000000
## Balanced Accuracy    0.5000000
```

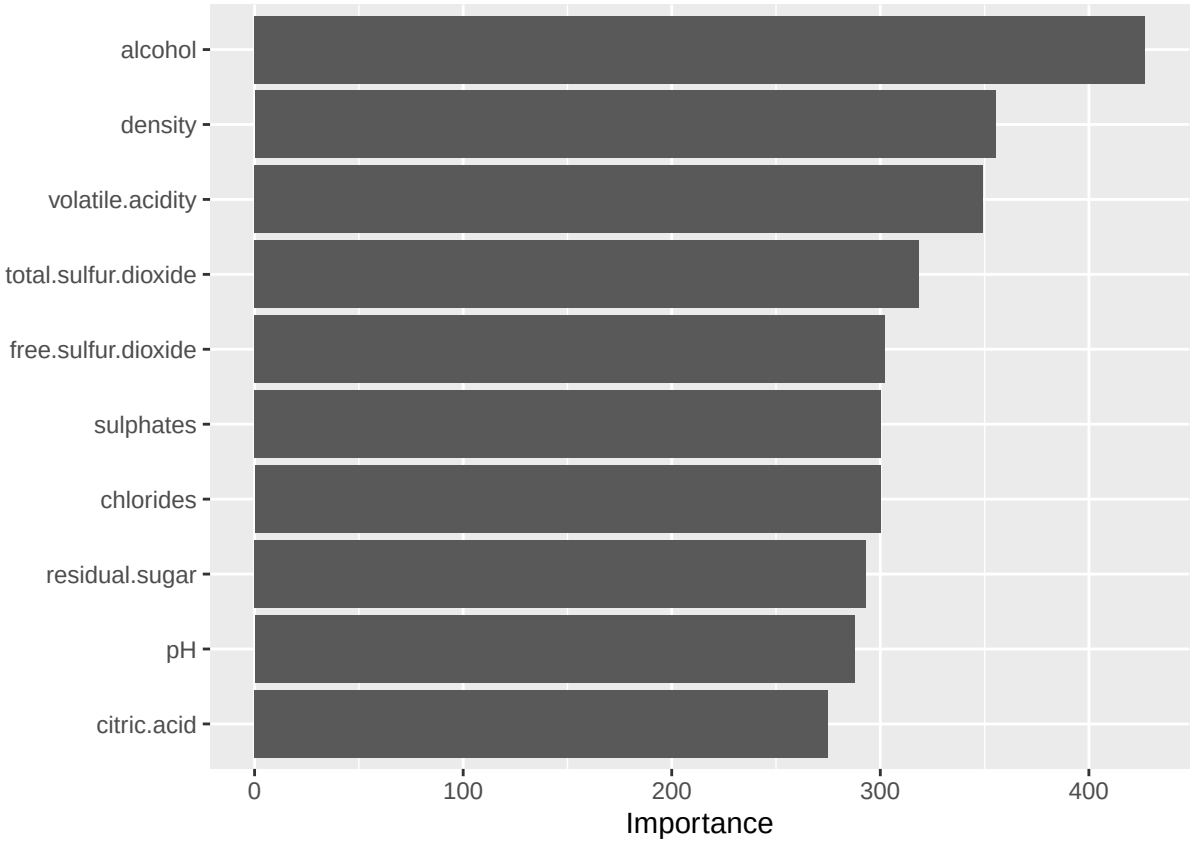
```r
plot(rf_model, main = "Random Forest Model Error Rate")
```




```r
importance_rf <- randomForest::importance(rf_model)
varImpPlot(rf_model, main = "Feature Importance in Random Forest")
```




```
vip(rf_model, num_features = 10)
```


alcohol
density
volatile.acidity
total.sulfur.dioxide
free.sulfur.dioxide
sulphates
chlorides
residual.sugar
pH
citric.acid
0
100
200
300
400
Importance