Red wine Quality

2024-11-11

Set Up and Load Packages

Load and Inspect the Dataset

```
# Load the dataset
wine_data <- read.csv("~/Downloads/winequality-red.csv")

# Preview the data
head(wine_data)
```

```
##   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
## 6           7.4             0.66        0.00            1.8     0.075
##   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
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

```
summary(wine_data)
```

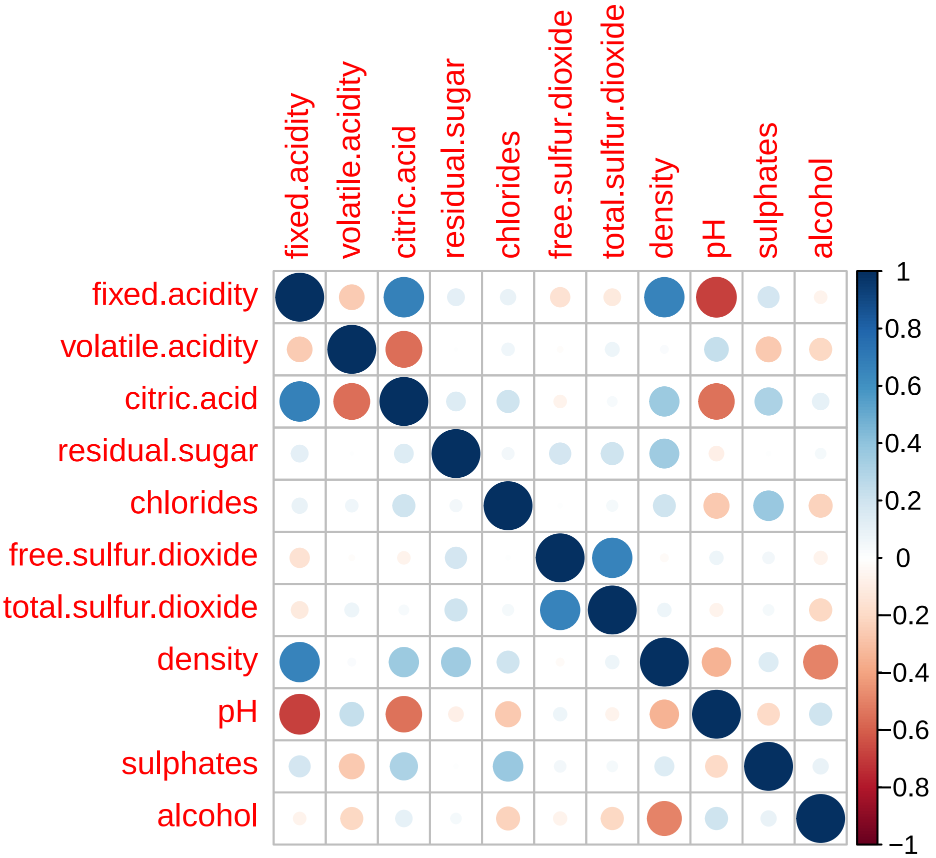
```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :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
```

```
# Check for missing values
sum(is.na(wine_data))
```

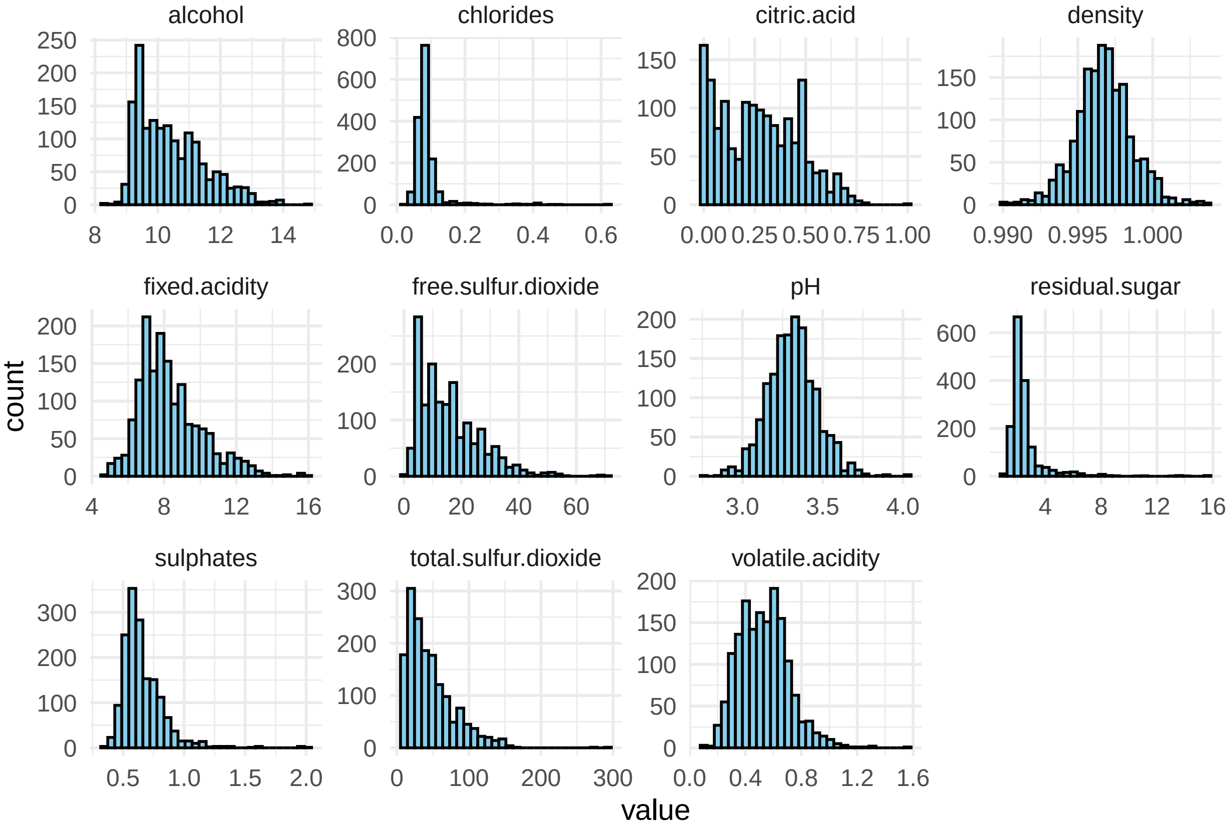
```
## [1] 0
```

Exploratory Data Analysis (EDA)

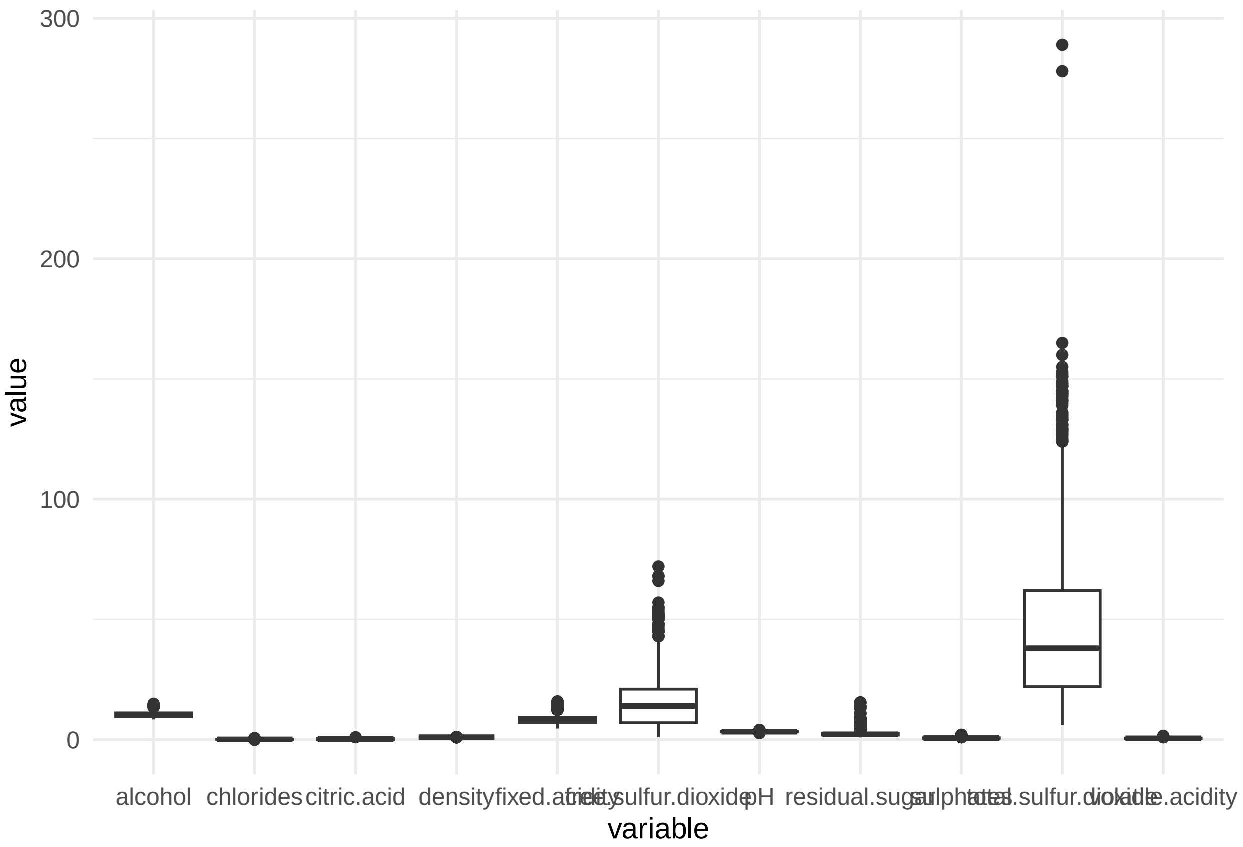
```
# Correlation matrix
cor_matrix <- cor(wine_data %>% select(-quality))
corrplot::corrplot(cor_matrix, method = "circle")
```



```
# Histograms for each variable
wine_data %>%
  gather(key = "variable", value = "value", -quality) %>%
  ggplot(aes(x = value)) +
  facet_wrap(~ variable, scales = "free") +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  theme_minimal()
```



```r
# Boxplot to identify outliers
wine_data %>%
  gather(key = "variable", value = "value", -quality) %>%
  ggplot(aes(x = variable, y = value)) +
  geom_boxplot() +
  theme_minimal()
```



Data Preprocessing

```r
# Scale features
wine_data_scaled <- as.data.frame(scale(wine_data %>% select(-quality)))
wine_data_scaled$quality <- wine_data$quality

# Convert quality to binary
wine_data_scaled$quality_binary <- ifelse(wine_data_scaled$quality >= 7, 1, 0)
table(wine_data_scaled$quality_binary)  # Check distribution
```

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

```r
# Set a seed for reproducibility
set.seed(123)

# Split the data
trainIndex <- createDataPartition(wine_data_scaled$quality_binary, p = 0.75, list = FALSE)
train_data <- wine_data_scaled[trainIndex, ]
test_data <- wine_data_scaled[-trainIndex, ]
```

Building a model

```r
# Convert quality to a binary factor (classification)
wine_data_scaled$quality_binary <- as.factor(ifelse(wine_data_scaled$quality >= 7, 1, 0))

# Split data again if needed, keeping quality_binary as the target
trainIndex <- createDataPartition(wine_data_scaled$quality_binary, p = 0.75, list = FALSE)
```

```r
train_data <- wine_data_scaled[trainIndex, ]
test_data <- wine_data_scaled[-trainIndex, ]

# Train the Random Forest model for classification
rf_model <- randomForest(quality_binary ~ . - quality, data = train_data, ntree = 100)

# Predict on test data with probability output
pred_rf <- predict(rf_model, newdata = test_data, type = "prob")[, 2]

# Check the first few predictions to confirm
head(pred_rf)
```

```
##    8    9   11   14   17   18
## 0.08 0.01 0.00 0.05 0.16 0.00
```

Evaluating model

```r
# Convert probabilities to binary predictions (0 or 1) with threshold 0.5
pred_class <- ifelse(pred_rf > 0.5, 1, 0)

# View the first few predictions to confirm
head(pred_class)
```

```
##  8  9 11 14 17 18
##  0  0  0  0  0  0
```

```r
# Load the caret package if not already loaded
library(caret)

# Confusion matrix to evaluate the performance of the model
confusion <- confusionMatrix(factor(pred_class), factor(test_data$quality_binary))
print(confusion)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 338  26
##          1   7  28
##
##                Accuracy : 0.9173
##                  95% CI : (0.8858, 0.9424)
##     No Information Rate : 0.8647
##     P-Value [Acc > NIR] : 0.0007493
##
##                   Kappa : 0.585
##
##  Mcnemar's Test P-Value : 0.0017280
##
##             Sensitivity : 0.9797
##             Specificity : 0.5185
##          Pos Pred Value : 0.9286
##          Neg Pred Value : 0.8000
##              Prevalence : 0.8647
##          Detection Rate : 0.8471
##    Detection Prevalence : 0.9123
##        Balanced Accuracy : 0.7491
##
##         'Positive' Class : 0
##
```

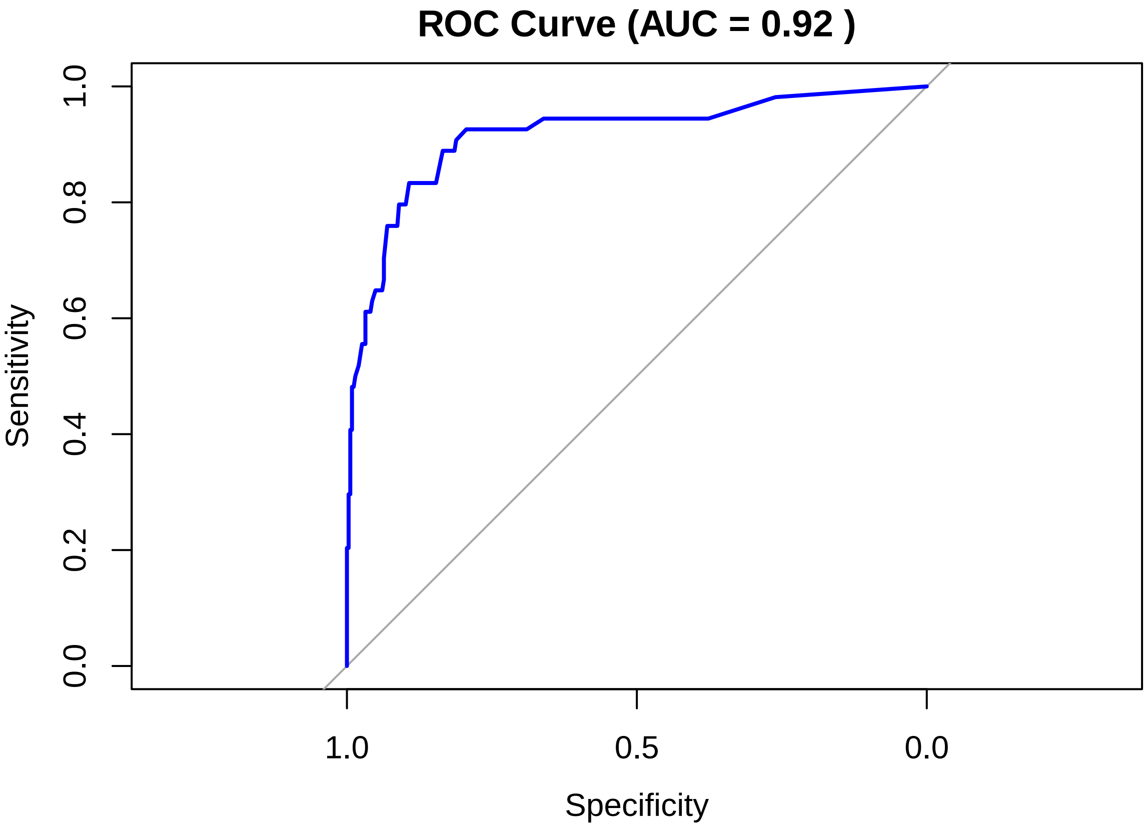
```
# Load the pROC library if not already loaded
library(pROC)

# Calculate the ROC curve
roc_curve <- roc(test_data$quality_binary, pred_rf)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

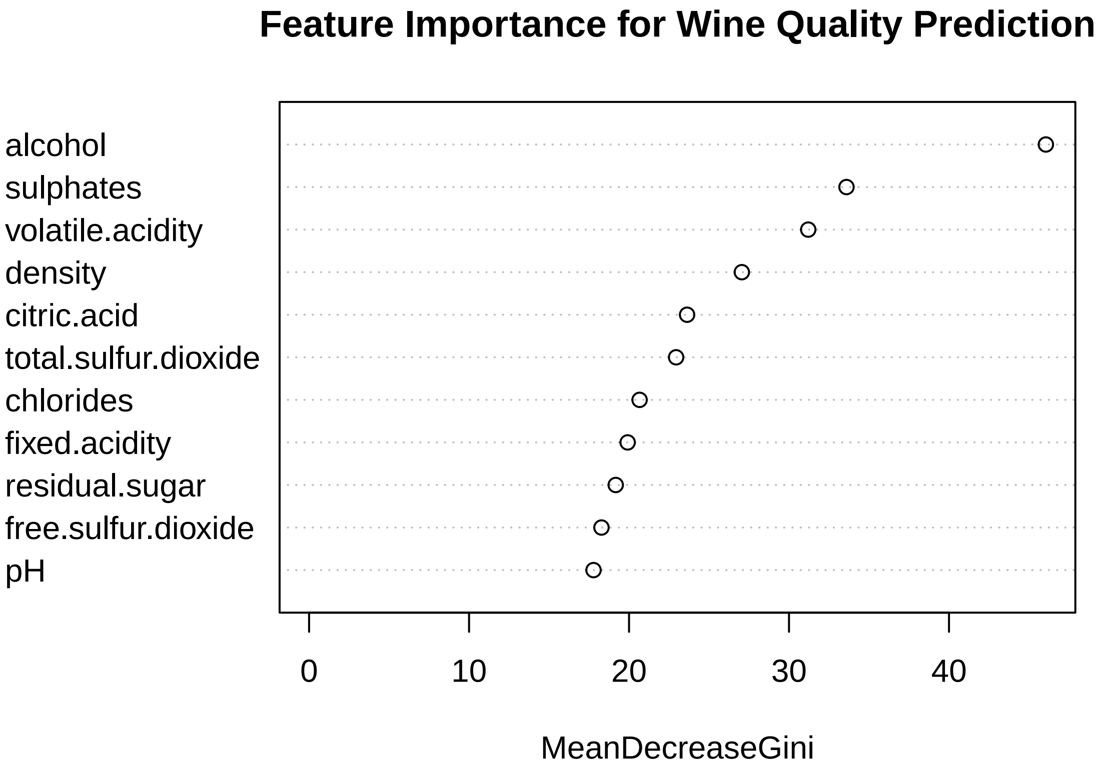
```
# Plot the ROC curve
plot(roc_curve, col = "blue", main = paste("ROC Curve (AUC =", round(auc(roc_curve), 2), ")"))
```



```
# Display the AUC value
auc_value <- auc(roc_curve)
print(paste("AUC:", round(auc_value, 2)))
```

```
## [1] "AUC: 0.92"
```

```
# Plot feature importance
varImpPlot(rf_model, main = "Feature Importance for Wine Quality Prediction")
```


Feature Importance for Wine Quality Prediction
alcohol
sulphates
volatile.acidity
density
citric.acid
total.sulfur.dioxide
chlorides
fixed.acidity
residual.sugar
free.sulfur.dioxide
pH
0
10
20
30
40
MeanDecreaseGini