Wines LDA QDA Classification

Isabella Chen

2025-04-30

```
library(MASS)  # For LDA and QDA
library(dplyr) # For data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2) # For visualization
library(caret)  # For data partitioning
```

```
## Loading required package: lattice
```

```
wine_data <- read.csv("/Users/isabellachen/Downloads/wine-quality-white-and-red.csv")
```

```
wine_data$quality_bin <- ifelse(wine_data$quality >= 7, "high", "low")
wine_data$quality_bin <- factor(wine_data$quality_bin)

# Split the data into training and testing sets (70% train, 30% test)
set.seed(1)  # For reproducibility
train_indices <- createDataPartition(wine_data$quality_bin, p = 0.7, list = FALSE)
train_data <- wine_data[train_indices, ]
test_data <- wine_data[-train_indices, ]

# Define the feature set (all variables except 'quality', 'quality_bin', and 'type')
features <- setdiff(names(wine_data), c("quality", "quality_bin", "type"))

# Create formula for the models
formula <- as.formula(paste("quality_bin ~", paste(features, collapse = " + ")))
```

```r
# Train LDA model
lda_model <- lda(formula, data = train_data)

# Train QDA model
qda_model <- qda(formula, data = train_data)

# Make predictions on test data
lda_pred <- predict(lda_model, test_data)
qda_pred <- predict(qda_model, test_data)
```

```r
# Evaluate LDA model
lda_conf_matrix <- confusionMatrix(lda_pred$class, test_data$quality_bin)
print("LDA Model Performance:")
```

```
## [1] "LDA Model Performance:"
```

```r
print(lda_conf_matrix)
```

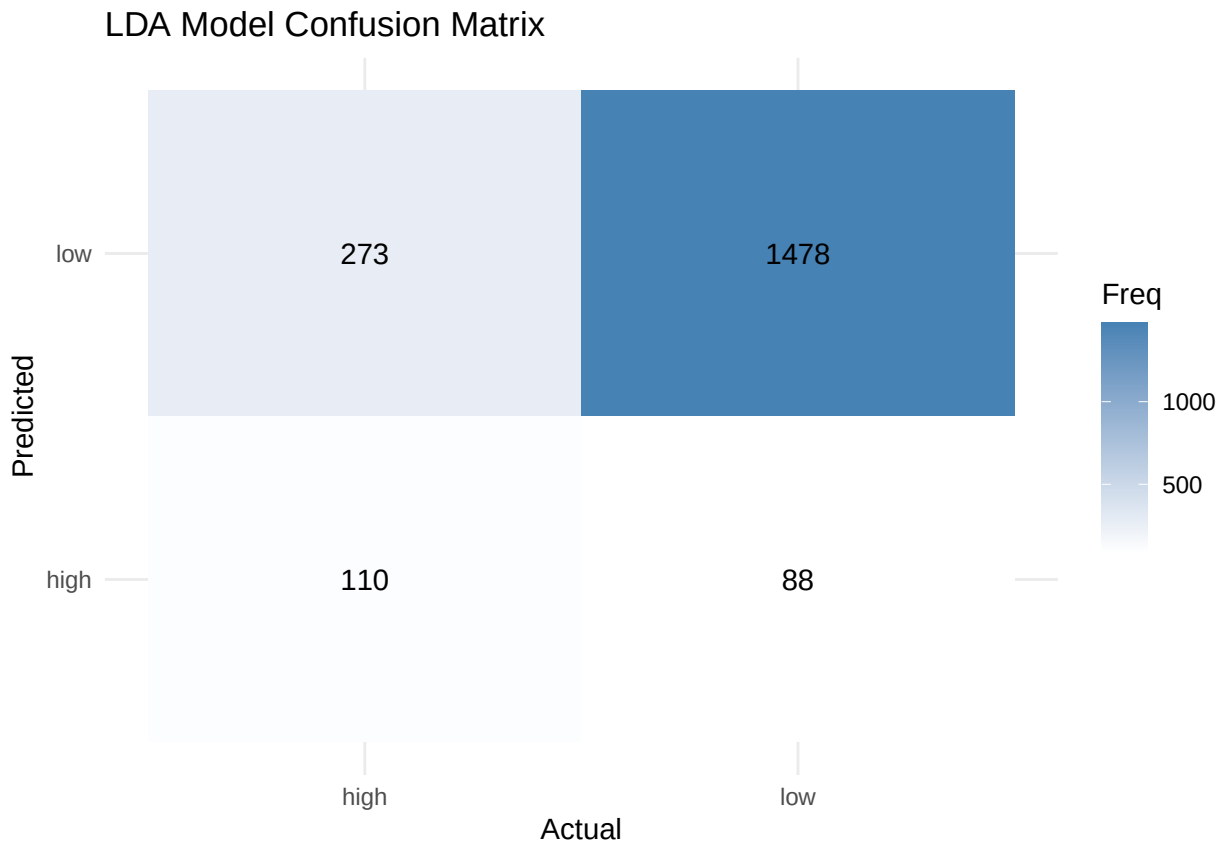
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high  low
##       high  110   88
##       low   273 1478
##
##                Accuracy : 0.8148
##                  95% CI : (0.7968, 0.8318)
##     No Information Rate : 0.8035
##     P-Value [Acc > NIR] : 0.1096
##
##                   Kappa : 0.2826
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.28721
##             Specificity : 0.94381
##          Pos Pred Value : 0.55556
##          Neg Pred Value : 0.84409
##              Prevalence : 0.19651
##          Detection Rate : 0.05644
##    Detection Prevalence : 0.10159
##       Balanced Accuracy : 0.61551
##
##        'Positive' Class : high
##
```

```r
print(paste("LDA Accuracy:", round(lda_conf_matrix$overall["Accuracy"] * 100, 2), "%"))
```

```
## [1] "LDA Accuracy: 81.48 %"
```

```
lda_cm_df <- as.data.frame(lda_conf_matrix$table)
colnames(lda_cm_df) <- c("Predicted", "Actual", "Freq")

ggplot(lda_cm_df, aes(x = Actual, y = Predicted, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "black") +
  scale_fill_gradient(low = "white", high = "steelblue") +
  labs(title = "LDA Model Confusion Matrix") +
  theme_minimal()
```




```
# Evaluate QDA model
qda_conf_matrix <- confusionMatrix(qda_pred$class, test_data$quality_bin)
print("QDA Model Performance:")
```

```
## [1] "QDA Model Performance:"
```

```
print(qda_conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high  low
##       high  262  341
##       low   121 1225
```

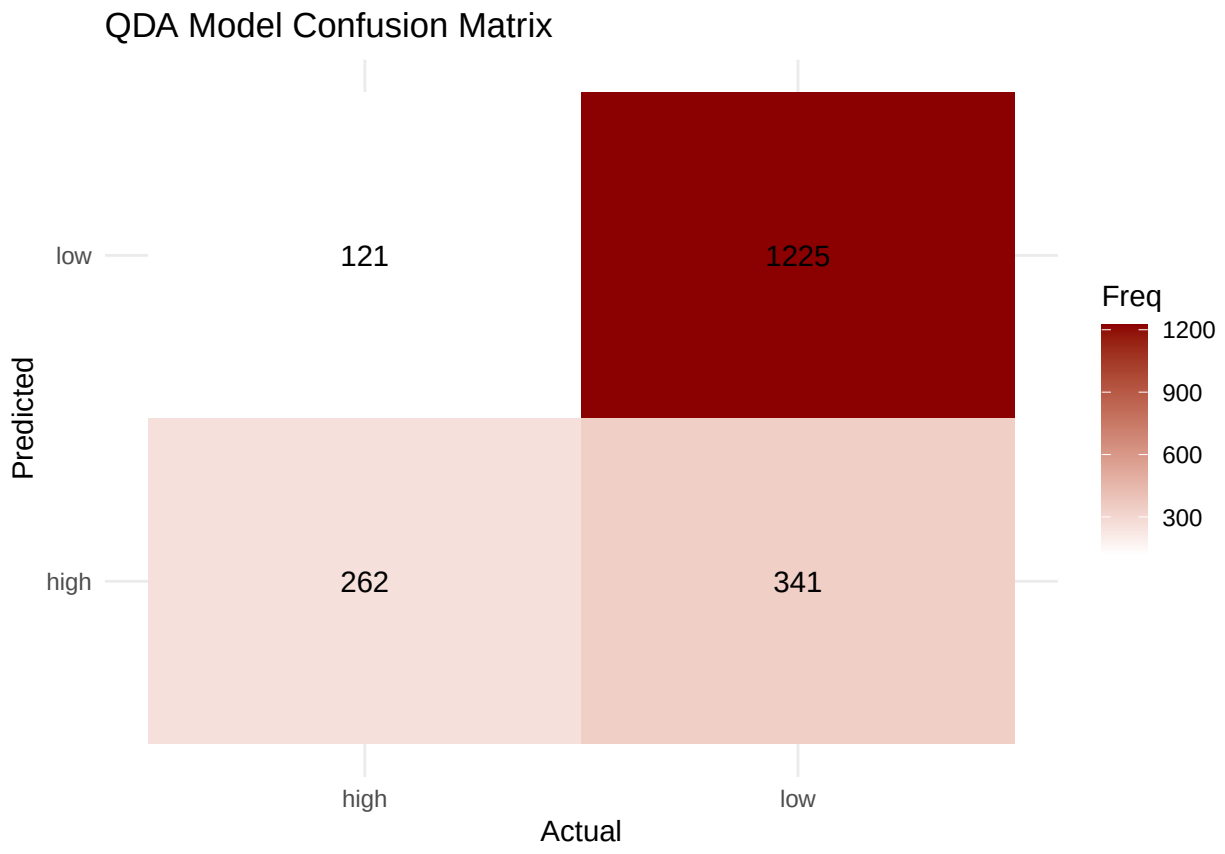
```
##
##                Accuracy : 0.763
##                  95% CI : (0.7434, 0.7817)
##     No Information Rate : 0.8035
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3832
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6841
##             Specificity : 0.7822
##          Pos Pred Value : 0.4345
##          Neg Pred Value : 0.9101
##              Prevalence : 0.1965
##          Detection Rate : 0.1344
##    Detection Prevalence : 0.3094
##       Balanced Accuracy : 0.7332
##
##        'Positive' Class : high
##
```

```r
print(paste("QDA Accuracy:", round(qda_conf_matrix$overall["Accuracy"] * 100, 2), "%"))
```

```
## [1] "QDA Accuracy: 76.3 %"
```

```r
qda_cm_df <- as.data.frame(qda_conf_matrix$table)
colnames(qda_cm_df) <- c("Predicted", "Actual", "Freq")

ggplot(qda_cm_df, aes(x = Actual, y = Predicted, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "black") +
  scale_fill_gradient(low = "white", high = "darkred") +
  labs(title = "QDA Model Confusion Matrix") +
  theme_minimal()
```




```
# Compare models
models <- c("LDA", "QDA")
accuracies <- c(
  lda_conf_matrix$overall["Accuracy"],
  qda_conf_matrix$overall["Accuracy"]
)
comparison <- data.frame(Model = models, Accuracy = accuracies)
print(comparison)
```

```
##   Model  Accuracy
## 1   LDA 0.8147768
## 2   QDA 0.7629554
```

```
set.seed(1)

# Define the training control using LOOCV
train_control <- trainControl(method = "LOOCV")

# Train LDA model with LOOCV
lda_model_loocv <- train(formula,
                         data = train_data,
                         method = "lda",
                         trControl = train_control)

# Train QDA model with LOOCV
```

```r
qda_model_loocv <- train(formula,
                         data = train_data,
                         method = "qda",
                         trControl = train_control)

print(lda_model_loocv)
```

```
## Linear Discriminant Analysis
##
## 4548 samples
##   11 predictor
##    2 classes: 'high', 'low'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 4547, 4547, 4547, 4547, 4547, 4547, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8172823  0.3063653
```

```r
print(qda_model_loocv)
```

```
## Quadratic Discriminant Analysis
##
## 4548 samples
##   11 predictor
##    2 classes: 'high', 'low'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 4547, 4547, 4547, 4547, 4547, 4547, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.769569  0.4014461
```

```r
set.seed(1)

# Define the training control using 10-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)

# Train LDA model with 10-fold CV
lda_model_cv <- train(formula,
                      data = train_data,
                      method = "lda",
                      trControl = train_control)

# Train QDA model with 10-fold CV
qda_model_cv <- train(formula,
                      data = train_data,
                      method = "qda",
```

```
                      trControl = train_control)

print(lda_model_cv)
```

```
## Linear Discriminant Analysis
##
## 4548 samples
##   11 predictor
##    2 classes: 'high', 'low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4094, 4093, 4094, 4093, 4092, 4093, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8174951  0.3068626
```

```
print(qda_model_cv)
```

```
## Quadratic Discriminant Analysis
##
## 4548 samples
##   11 predictor
##    2 classes: 'high', 'low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4093, 4092, 4093, 4094, 4093, 4093, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7704505  0.402677
```

```
# Features are most important in the LDA model
lda_coeffs <- data.frame(Feature = features,
                         Coefficient = abs(lda_model$scaling[,1]))
lda_coeffs <- lda_coeffs[order(lda_coeffs$Coefficient, decreasing = TRUE), ]
print(lda_coeffs)
```

```
##                                 Feature  Coefficient
## density                         density 2.823293e+02
## sulphates                     sulphates 1.901737e+00
## pH                                   pH 1.718736e+00
## chlorides                     chlorides 1.631052e+00
## volatile.acidity     volatile.acidity 1.464407e+00
## alcohol                       alcohol 5.157228e-01
## fixed.acidity            fixed.acidity 3.434493e-01
## residual.sugar          residual.sugar 1.499258e-01
## citric.acid                citric.acid 1.282984e-01
## free.sulfur.dioxide   free.sulfur.dioxide 1.095375e-02
## total.sulfur.dioxide total.sulfur.dioxide 4.446837e-03
```

```r
# Features are most important in the QDA model
qda_coeffs <- data.frame(Feature = features,
                         Coefficient = apply(qda_model$scaling, 1, function(x) sum(abs(x))))
qda_coeffs <- qda_coeffs[order(qda_coeffs$Coefficient, decreasing = TRUE), ]
print(qda_coeffs)
```

```
##                                 Feature  Coefficient
## density                         density 4.938124e+03
## chlorides                     chlorides 1.920972e+02
## volatile.acidity     volatile.acidity 4.939625e+01
## citric.acid               citric.acid 3.534119e+01
## pH                                   pH 2.853784e+01
## sulphates                   sulphates 2.120719e+01
## fixed.acidity           fixed.acidity 9.404075e+00
## alcohol                       alcohol 3.894990e+00
## residual.sugar         residual.sugar 2.892464e+00
## free.sulfur.dioxide free.sulfur.dioxide 2.736342e-01
## total.sulfur.dioxide total.sulfur.dioxide 9.594711e-02
```