Statistical Learning Lab

Assignment - 3

LDA, QDA and KNN Assignment

Submitted by,

Ben Abraham Biju

22IM10048

1. Load the dataset "diabetes.csv". Display the first few rows of the dataset.

The dataset was loaded using the read.csv() function and stored in the variable diabetes.

```
> diabetes <- read.csv("C:/Users/benab/OneDrive - iitkgp.ac.in/Desktop/Sem 6/SL Lab/Lab 3/diabetes.csv")
> head(diabetes, 10)
   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI DiabetesPedigreeFunction Age Outcome
1            6     148            72            35       0 33.6                    0.627  50       1
2            1      85            66            29       0 26.6                    0.351  31       0
3            8     183            64             0       0 23.3                    0.672  32       1
4            1      89            66            23      94 28.1                    0.167  21       0
5            0     137            40            35     168 43.1                    2.288  33       1
6            5     116            74             0       0 25.6                    0.201  30       0
7            3      78            50            32      88 31.0                    0.248  26       1
8           10     115             0             0       0 35.3                    0.134  29       0
9            2     197            70            45     543 30.5                    0.158  53       1
10           8     125            96             0       0  0.0                    0.232  54       1
```

2. Perform preliminary analysis to show how the variables are related to each other. Use scatter plot, box plot etc. to visualize how different variables impact the "Outcome" variable.

To generate the scatter plot of predictor variables, the Outcome variable was converted into a categorical factor with values labeled as "Diabetic" and "Non-Diabetic."

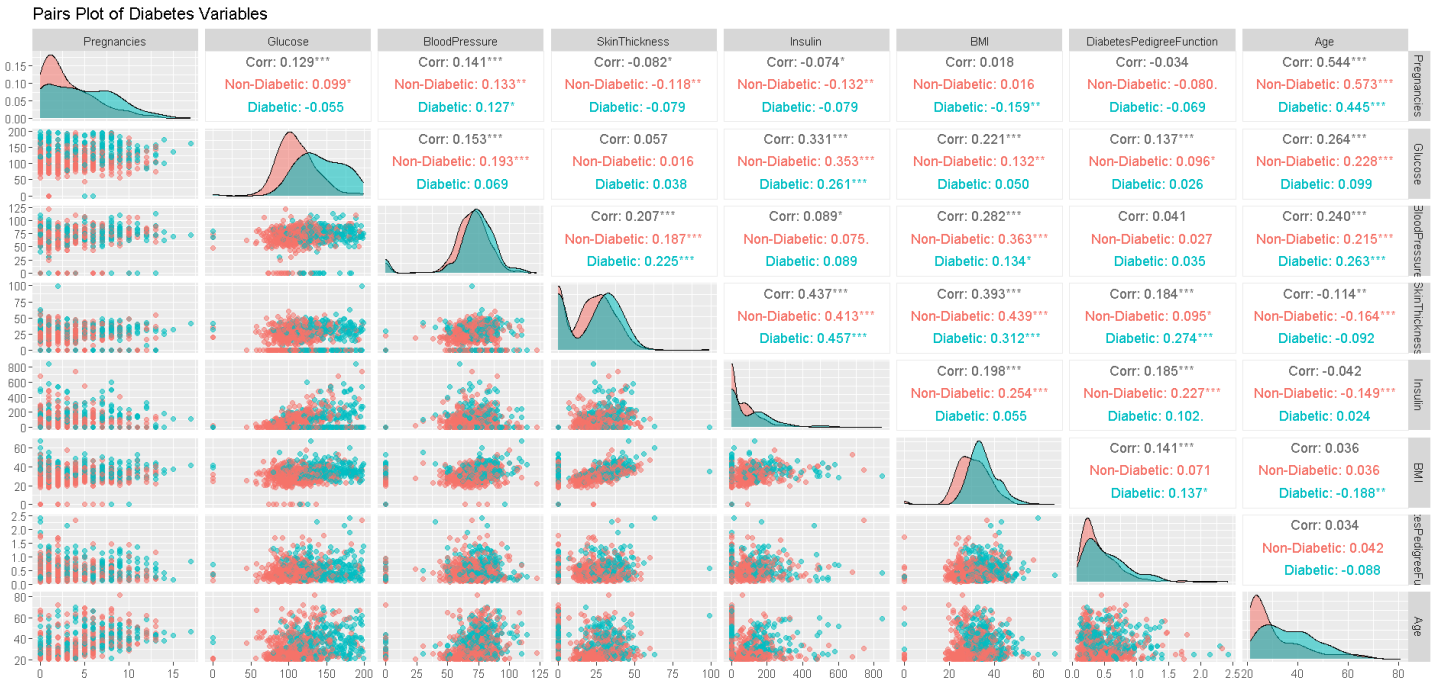
```
> diabetes$Outcome <- factor(diabetes$Outcome, labels = c("Non-Diabetic", "Diabetic"))
> head(diabetes, 10)
   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI DiabetesPedigreeFunction Age      Outcome
1            6     148            72            35       0 33.6                    0.627  50     Diabetic
2            1      85            66            29       0 26.6                    0.351  31 Non-Diabetic
3            8     183            64             0       0 23.3                    0.672  32     Diabetic
4            1      89            66            23      94 28.1                    0.167  21 Non-Diabetic
5            0     137            40            35     168 43.1                    2.288  33     Diabetic
6            5     116            74             0       0 25.6                    0.201  30 Non-Diabetic
7            3      78            50            32      88 31.0                    0.248  26     Diabetic
8           10     115             0             0       0 35.3                    0.134  29 Non-Diabetic
9            2     197            70            45     543 30.5                    0.158  53     Diabetic
10           8     125            96             0       0  0.0                    0.232  54     Diabetic
>
```

Scatter Plots

```
ggpairs(diabetes[, -9], aes(color = diabetes$Outcome, alpha = 0.5)) +
  ggtitle("Pairs Plot of Diabetes Variables")
```

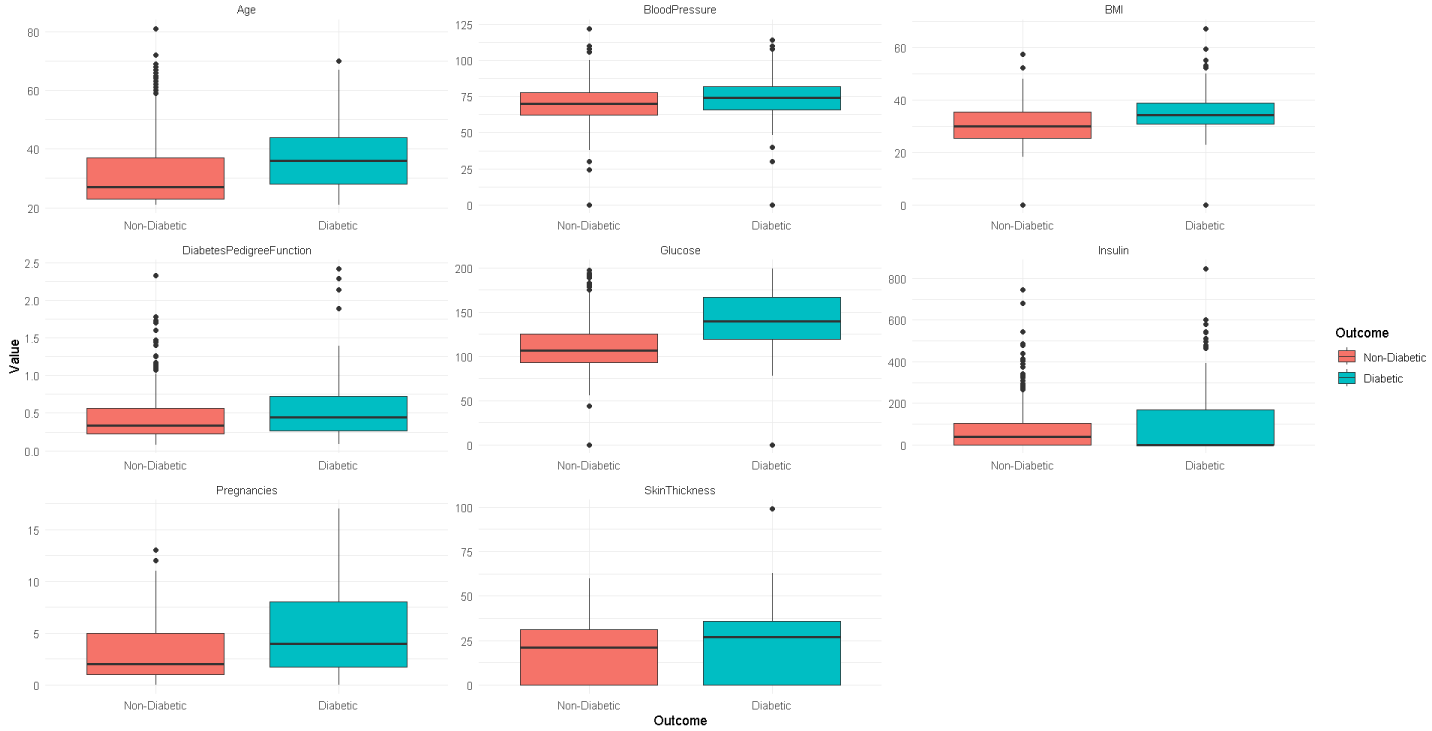
Observations from the scatter plot:

- The scatter plots differentiate between the Diabetic (blue) and Non-Diabetic (red) groups.
- Glucose, BMI, Insulin and Age tend to have higher values for the Diabetic group (shown in blue color), which indicates that they can be strong indicators.
- Blood Pressure and Skin Thickness do not seem to have a strong individual correlation with diabetes from the plots, but can have a collective impact on the outcome.



Box Plots

```
ggplot(gather(diabetes, key = "Predictor", value = "Value", -Outcome),
       aes(x = Outcome, y = Value, fill = Outcome)) +
  geom_boxplot() +
  facet_wrap(~ Predictor, scales = "free") +
  theme_minimal()
```



Observations from the Box Plots:

- For the Diabetic group, the mean values of Glucose, BMI, Age, Pregnancies are higher than those in the Non-Diabetic group.
- There are several outliers present in the columns for Age, DiabetesPedigreeFunction, and Insulin. This suggests the presence of extreme values in these variables, which could significantly affect the analysis and interpretation of the dataset.

3. Randomly sample 80% of the data as training data and rest as test data. Fit a LDA model and interpret the result.

The diabetes dataset is divided into **training and testing sets** using the *split(..)* function, based on the Outcome variable. This ensures that both datasets maintain similar proportions of the target variable.

```
> set.seed(123)
> split <- sample.split(diabetes$Outcome, SplitRatio = 0.8)
> train <- subset(diabetes, split == TRUE)
> test <- subset(diabetes, split == FALSE)
> head(train)
  Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI DiabetesPedigreeFunction Age      Outcome
1           6     148            72            35       0 33.6                    0.627  50     Diabetic
2           1      85            66            29       0 26.6                    0.351  31 Non-Diabetic
3           8     183            64             0       0 23.3                    0.672  32     Diabetic
4           1      89            66            23      94 28.1                    0.167  21 Non-Diabetic
5           0     137            40            35     168 43.1                    2.288  33     Diabetic
8          10     115             0             0       0 35.3                    0.134  29 Non-Diabetic
>
```

The training dataset is used to train a LDA model, with the Outcome as the target variable. The prior probabilities in the training dataset show a high imbalance in the data, with the value of the Non-Diabetic group being almost double that of the Diabetic group.

```
> lda_model <- lda(Outcome ~ ., data = train)
> lda_model
Call:
lda(Outcome ~ ., data = train)

Prior probabilities of groups:
Non-Diabetic     Diabetic
   0.6514658    0.3485342

Group means:
             Pregnancies  Glucose BloodPressure SkinThickness  Insulin      BMI DiabetesPedigreeFunction     Age
Non-Diabetic    3.187500 109.6650      68.71500      19.30500 68.26000 30.37150                0.4349750 30.9075
Diabetic        4.817757 141.5561      70.57477      21.85981 93.06542 35.02477                0.5411542 37.0000

Coefficients of linear discriminants:
                                LD1
Pregnancies               0.100735313
Glucose                   0.027532709
BloodPressure            -0.011924931
SkinThickness             0.005520363
Insulin                  -0.001685088
BMI                       0.059186395
DiabetesPedigreeFunction  0.510757708
Age                       0.012592865
```

- Diabetic individuals tend to have higher values for variables like *Pregnancies*, *Glucose*, *Insulin*, *BMI*, and *Age* compared to non-diabetic individuals.
- *Diabetes Pedigree Function* has the highest coefficient (**0.5108**), which shows that genetic features play a crucial role.
- Positive coefficients (e.g., *Pregnancies, BMI, DiabetesPedigreeFunction, Age*) suggest that higher values of these variables are associated with a higher likelihood of being diabetic, whereas negative coefficients (e.g., *BloodPressure*) suggest that higher values of these variables decrease the likelihood of being diabetic.

4. From the model fitted in problem 3, derive confusion matrix, accuracy, and F1-score on test data.

The predictions made by the LDA model are used to calculate the confusion matrix. The ***Diabetic*** group is the positive group and the ***Non-Diabetic*** group is the negative group in the matrix, as shown in the code.

```
> confusion_lda <- confusionMatrix(lda_pred$class, test$Outcome, positive ="Diabetic")
> confusion_lda
Confusion Matrix and Statistics

             Reference
Prediction     Non-Diabetic Diabetic
  Non-Diabetic           85       25
  Diabetic               15       29

               Accuracy : 0.7403
                 95% CI : (0.6635, 0.8075)
    No Information Rate : 0.6494
    P-Value [Acc > NIR] : 0.01009

                  Kappa : 0.4043

 Mcnemar's Test P-Value : 0.15473

            Sensitivity : 0.5370
            Specificity : 0.8500
         Pos Pred Value : 0.6591
         Neg Pred Value : 0.7727
             Prevalence : 0.3506
         Detection Rate : 0.1883
   Detection Prevalence : 0.2857
      Balanced Accuracy : 0.6935

       'Positive' Class : Diabetic
```

- From the confusion matrix, we can see that the model correctly classified 85 Non-Diabetic cases and 29 Diabetic cases. It misclassified 25 Non-Diabetic cases as Diabetic and 15 Diabetic cases as Non-Diabetic.
- The **sensitivity of 53.70%** indicates that the model correctly identifies 53.70% of Diabetic cases (true positives). This suggests the model has room for improvement in identifying Diabetic individuals.
- The No Information Rate (NIR), which represents the accuracy obtained by always predicting the majority class (Non-Diabetic), is 64.94%. The model's accuracy is significantly higher than this baseline, with a p-value of 0.01009, suggesting that the model performs better than random guessing.

```
> accuracy_lda <- confusion_lda$overall['Accuracy']
> accuracy_lda
 Accuracy
0.7402597
> f1_lda <- confusion_lda$byClass['F1']
> f1_lda
       F1
0.5918367
```

The overall **accuracy of the model is 74.03%**, meaning the model correctly predicted 74.03% of the cases in the test set.

The **F1 score** for the Diabetic class is approximately **0.5925**. This indicates a moderate balance between precision and recall for the Diabetic category.

5. Fit QDA and KNN (K = 5) models on training data. Compare the metrics in problem 4 for LDA, QDA and KNN models for test data and discuss the results.

A Quadratic Discriminant Analysis (QDA) model and a K-Nearest Neighbours (KNN)model,with k-5 are fitted on the training dataset. The values in the training dataset are scaled prior to training for the KNN model to prevent distortion in distance classification and ensure uniformity of features.

```
# QDA Model
qda_model <- qda(Outcome ~ ., data = train)
qda_pred <- predict(qda_model, newdata = test)

# KNN Model
train_scale <- scale(train[, -9])
test_scale <- scale(test[, -9])

knn_pred <- knn(train_scale, test_scale,
                cl = train$Outcome, k = 5)
```

The confusion matrices are calculated for both the models, based on the predictions of the test data, with the positive class being 'Diabetic'.

Confusion Matrix of QDA model

```
> confusion_qda <- confusionMatrix(qda_pred$class, test$Outcome, positive ="Diabetic")
> confusion_qda
Confusion Matrix and Statistics

              Reference
Prediction     Non-Diabetic Diabetic
  Non-Diabetic           76       19
  Diabetic               24       35

               Accuracy : 0.7208
                 95% CI : (0.6429, 0.79)
    No Information Rate : 0.6494
    P-Value [Acc > NIR] : 0.03637

                  Kappa : 0.3996

 Mcnemar's Test P-Value : 0.54187

            Sensitivity : 0.6481
            Specificity : 0.7600
         Pos Pred Value : 0.5932
         Neg Pred Value : 0.8000
             Prevalence : 0.3506
         Detection Rate : 0.2273
   Detection Prevalence : 0.3831
      Balanced Accuracy : 0.7041

       'Positive' Class : Diabetic
```

Confusion Matrix of KNN model

```
> confusion_knn <- confusionMatrix(knn_pred, test$Outcome, positive ="Diabetic")
> confusion_knn
Confusion Matrix and Statistics

              Reference
Prediction     Non-Diabetic Diabetic
  Non-Diabetic           82       28
  Diabetic               18       26

               Accuracy : 0.7013
                 95% CI : (0.6224, 0.7723)
    No Information Rate : 0.6494
    P-Value [Acc > NIR] : 0.1016

                  Kappa : 0.3149

 Mcnemar's Test P-Value : 0.1845

            Sensitivity : 0.4815
            Specificity : 0.8200
         Pos Pred Value : 0.5909
         Neg Pred Value : 0.7455
             Prevalence : 0.3506
         Detection Rate : 0.1688
   Detection Prevalence : 0.2857
      Balanced Accuracy : 0.6507

       'Positive' Class : Diabetic
```

The metrics for all the three models are combined to a dataframe for analysis.

```
> metrics <- data.frame(
+   Model = c("LDA", "QDA", "KNN (k=5)"),
+   Accuracy = c(accuracy_lda,
+                confusion_qda$overall['Accuracy'],
+                confusion_knn$overall['Accuracy']),
+   F1_Score = c(f1_lda,
+                confusion_qda$byClass['F1'],
+                confusion_knn$byClass['F1'])
+ )
> print(metrics)
      Model  Accuracy  F1_Score
1       LDA 0.7402597 0.5918367
2       QDA 0.7207792 0.6194690
3 KNN (k=5) 0.7012987 0.5306122
```

- Out of the three models, **LDA** has the **highest accuracy of 74.03%**
- LDA provides the best overall accuracy, but **QDA has a slightly better F1 score of 0.6194**, showing that it may have a **better balance between precision and recall**, especially for the Diabetic class.
- **KNN** with k=5 appears to **underperform relative to both LDA and QDA**, with lower accuracy and F1 score. Different values for k or parameter tuning may be needed to improve its performance.

6. Plot ROC curve for LDA and QDA models using the test data.

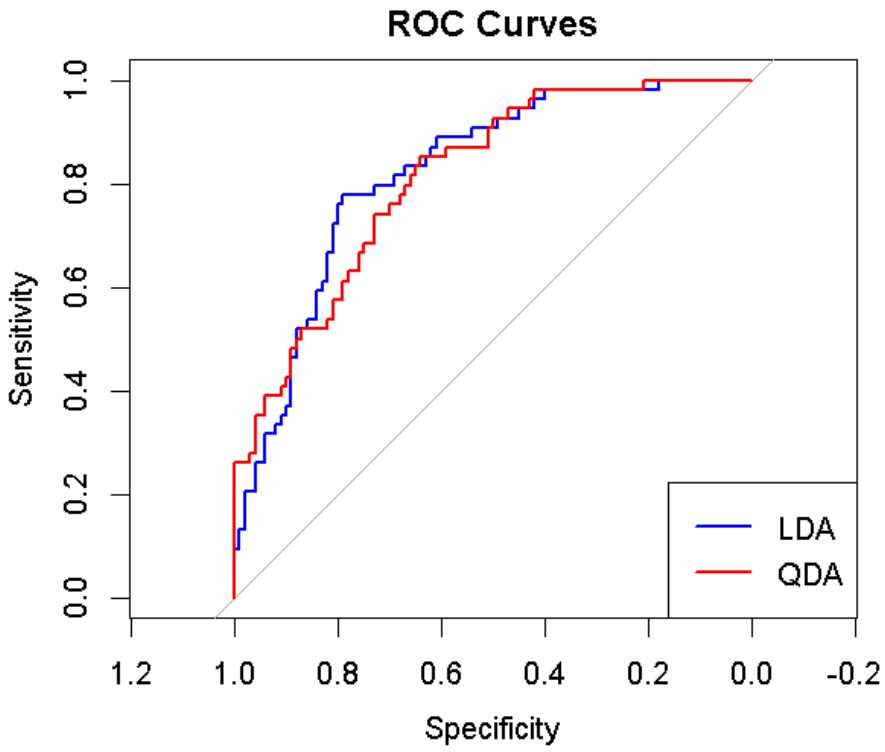
The Receiver Operating Characteristic (ROC) curve is fitted for the LDA and QDA models.

```
roc_lda <- roc(response = test$Outcome,
               predictor = as.numeric(lda_pred$posterior[,"Diabetic"]),
               levels = c("Non-Diabetic", "Diabetic"))  # Negative first, positive second

roc_qda <- roc(response = test$Outcome,
               predictor = as.numeric(qda_pred$posterior[,"Diabetic"]),
               levels = c("Non-Diabetic", "Diabetic"))

plot(roc_lda, col = "blue", main = "ROC Curves")
lines(roc_qda, col = "red")
legend("bottomright", legend = c("LDA", "QDA"), col = c("blue", "red"), lwd = 2)
```

- In the graph, we can see that LDA is initially below QDA initially but marginally improves and goes above QDA later. This shows that the LDA model may be better at classifying negative cases (Non- Diabetic) at higher thresholds, while QDA might be better at identifying positive cases (Diabetic) at lower thresholds.
- The area under the ROC curve (AUC) would provide a more precise measure of overall performance, but this pattern suggests that LDA and QDA have different strengths at different threshold values. LDA appears to be marginally better compared to QDA.



7. Plot accuracy and f1-score by varying the neighbourhood size from K=1 to K=20 and interpret the results.

The k-value is varied from 1 to 20 and each time, the accuracy and the F1-Score are recorded in a dataframe to see the optimal fit.

```
k_values <- 1:20
results <- data.frame(k = k_values, Accuracy = numeric(20), F1 = numeric(20))

for (k in k_values) {
  knn_pred_temp <- knn(train_scale, test_scale,
                       cl = train$Outcome, k = k)
  cm <- confusionMatrix(knn_pred_temp, test$Outcome)
  results$Accuracy[k] <- cm$overall['Accuracy']
  results$F1[k] <- cm$byClass['F1']
}

best_k_accuracy <- results$k[which.max(results$Accuracy)]
best_k_f1 <- results$k[which.max(results$F1)]

cat("Best k for Accuracy:", best_k_accuracy, "\n")
cat("Best k for F1 Score:", best_k_f1, "\n")
```

The highest accuracy and F1-Score is obtained for **k = 18.**

```
> cat("Best k for Accuracy:", best_k_accuracy, "\n")
Best k for Accuracy: 18
> cat("Best k for F1 Score:", best_k_f1, "\n")
Best k for F1 Score: 18
```

The corresponding accuracy and F1-score for k=18 are shown below.

```
> best_accuracy <- max(results$Accuracy)
> best_f1 <- max(results$F1)
> cat("Highest Accuracy:", best_accuracy, "\n")
Highest Accuracy: 0.7662338
> cat("Highest F1 Score:", best_f1, "\n")
Highest F1 Score: 0.8252427
```

Graph plot of Accuracy and F1 Score with varying k values.

```
# Plot accuracy and f1 score vs K
ggplot(results, aes(x = k)) +
  geom_line(aes(y = Accuracy, color = "Accuracy")) +
  geom_line(aes(y = F1, color = "F1 Score")) +
  scale_color_manual(values = c("Accuracy" = "blue", "F1 Score" = "red")) +
  labs(title = "KNN Performance vs Neighborhood Size",
       x = "Number of Neighbors (K)",
       y = "Metric Value") +
  theme_minimal()
```

Conclusions and Discussion:

- LDA demonstrated the best overall accuracy (74.03%) among the models, suggesting it was the most reliable in terms of correctly classifying both Diabetic and Non-Diabetic cases.
- QDA slightly outperformed LDA in terms of the F1 score (0.6195 vs. 0.5918), indicating it achieved a better balance between precision and recall, particularly for the Diabetic class.
- The ROC curve analysis showed that LDA initially underperformed compared to QDA, but as the threshold increased, LDA marginally surpassed QDA, highlighting its better ability to

classify Non-Diabetic cases at higher thresholds.

- On training the KNN model on different values of K, the accuracy and F1 score significantly increased and reached the highest at k=18. This shows the earlier value of k=5 was not optimal and was underfitted leading to poor accuracy in predictions. The accuracy for k=18, was as good as the LDA model, showing the flexibility of KNN models.
- Overall, LDA provided the most balanced and reliable results for the classification task, while QDA offered a better trade-off between precision and recall