Code:

```
# Convert the target variable to a factor
data$y <- as.factor(ifelse(data$y == "yes", 1, 0))

# Handle categorical variables by converting them to factors
categorical_vars <- c('job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month',
'day_of_week', 'poutcome')
data[categorical_vars] <- lapply(data[categorical_vars], as.factor)

# Split the data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(data$y, p = .8, list = FALSE, times = 1)
dataTrain <- data[ trainIndex,]
dataTest  <- data[-trainIndex,]

# Logistic Regression model
logistic_model <- glm(y ~ ., data = dataTrain, family = binomial)
summary(logistic_model)
```

Result:

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = dataTrain)
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.182e+02  4.297e+01  -5.078 3.82e-07 ***
## age                   8.420e-04  2.718e-03   0.310 0.756749
## jobblue-collar       -2.671e-01  8.899e-02  -3.002 0.002683 **
## jobentrepreneur      -1.592e-01  1.381e-01  -1.153 0.249023
## jobhousemaid         -2.131e-02  1.641e-01  -0.130 0.896699
## jobmanagement              -3.507e-02  9.462e-02  -0.371 0.710925
## jobretired                 2.857e-01  1.197e-01   2.387 0.016983 *
## jobself-employed           -9.936e-02  1.282e-01  -0.775 0.438194
## jobservices                -2.115e-01  9.890e-02  -2.138 0.032484 *
## jobstudent                 2.182e-01  1.245e-01   1.752 0.079822 .
## jobtechnician              -3.832e-02  7.956e-02  -0.482 0.630099
## jobunemployed               1.002e-01  1.409e-01   0.711 0.477119
## jobunknown                 -2.366e-01  2.786e-01  -0.849 0.395699
## maritalmarried             -2.214e-02  7.686e-02  -0.288 0.773341
## maritalsingle              5.888e-02  8.769e-02   0.671 0.501944
## maritalunknown             -1.053e-01  4.699e-01  -0.224 0.822734
## educationbasic.6y           6.725e-02  1.364e-01   0.493 0.622119
## educationbasic.9y          -2.870e-02  1.068e-01  -0.269 0.788117
## educationhigh.school        -2.386e-02  1.031e-01  -0.231 0.817001
## educationilliterate          1.335e+00  8.356e-01   1.598 0.110077
## educationprofessional.course  1.216e-01  1.126e-01   1.081 0.279852
## educationuniversity.degree    1.570e-01  1.028e-01   1.528 0.126551
## educationunknown             5.633e-02  1.340e-01   0.420 0.674251
## defaultunknown              -3.019e-01  7.514e-02  -4.018 5.86e-05 ***
## defaultyes                 -7.316e+00  1.135e+02  -0.064 0.948597
## housingunknown              -1.566e-01  1.544e-01  -1.014 0.310683
## housingyes                 -1.407e-02  4.621e-02  -0.305 0.760735
## loanunknown                        NA         NA      NA       NA
## loanyes                    -3.360e-02  6.404e-02  -0.525 0.599783
## contacttelephone            -5.905e-01  8.477e-02  -6.965 3.28e-12 ***
## monthaug                   8.277e-01  1.344e-01   6.157 7.42e-10 ***
## monthdec                   2.283e-01  2.334e-01   0.978 0.327897
## monthjul                   1.290e-01  1.067e-01   1.210 0.226467
## monthjun                   -5.544e-01  1.405e-01  -3.946 7.94e-05 ***
## monthmar                    1.924e+00  1.598e-01  12.046  < 2e-16 ***
## monthmay              -4.946e-01  9.154e-02  -5.403 6.57e-08 ***
## monthnov              -4.640e-01  1.353e-01  -3.430 0.000603 ***
## monthoct               2.160e-01  1.718e-01   1.257 0.208611
## monthsep               3.128e-01  2.005e-01   1.560 0.118758
## day_of_weekmon        -7.727e-02  7.374e-02  -1.048 0.294702
## day_of_weekthu         9.468e-02  7.167e-02   1.321 0.186511
## day_of_weektue         1.118e-01  7.396e-02   1.511 0.130751
## day_of_weekwed         2.141e-01  7.333e-02   2.919 0.003507 **
## duration               4.655e-03  8.320e-05  55.948  < 2e-16 ***
## campaign              -3.956e-02  1.296e-02  -3.053 0.002264 **
## pdays                 -7.899e-04  2.374e-04  -3.327 0.000878 ***
## previous              -2.950e-02  6.741e-02  -0.438 0.661661
## poutcomenonexistent    4.674e-01  1.060e-01   4.409 1.04e-05 ***
## poutcomesuccess        1.072e+00  2.308e-01   4.646 3.38e-06 ***
## emp.var.rate          -1.744e+00  1.579e-01 -11.045  < 2e-16 ***
## cons.price.idx         2.055e+00  2.825e-01   7.274 3.50e-13 ***
## cons.conf.idx          1.492e-02  8.630e-03   1.729 0.083778 .
## euribor3m              3.854e-01  1.463e-01   2.635 0.008423 **
## nr.employed            4.197e-03  3.505e-03   1.197 0.231224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23199  on 32950  degrees of freedom
## Residual deviance: 13673  on 32898  degrees of freedom
## AIC: 13779
##
## Number of Fisher Scoring iterations: 10
```

Interpretation:

The dataset is loaded from the CSV file "bank-additional-full.csv" using read.csv() with a semicolon separator. The column names are checked with colnames(data). The target variable y is converted to a factor, where "yes" is coded as 1 and "no" as 0, making it suitable for classification tasks. Several categorical variables (job, marital, education, default, housing, loan, contact, month, day_of_week, and poutcome) are identified and converted to factors using lapply(), facilitating their use in modeling. Then the dataset is being split into training and testing sets. The set.seed(123) function ensures reproducibility of the results by setting a seed for random number generation. The createDataPartition() function from the caret package creates a partition of the data, with 80% allocated to training (dataTrain) and 20% to testing (dataTest). This ensures that the model can be trained on a subset of the data and then tested on a separate subset to evaluate its performance.

The logistic regression model reveals that the likelihood of subscribing to a term deposit is influenced by various factors. Significant predictors include job type (with blue-collar workers less likely and retirees more likely to subscribe), contact method (telephone contact reduces likelihood), and specific months (higher likelihood in August and March, lower in June, May, and November). Longer call duration strongly increases the likelihood of subscription. Economic indicators such as the employment variation rate and consumer price index also significantly affect the likelihood. While some variables like age, marital status, and certain education levels are not significant, factors such as previous campaign contacts (pdays) and the outcome of previous contacts (poutcome) play a crucial role in predicting subscriptions.

Model Statistics:

- **Null deviance**: 23199, on 32950 degrees of freedom.
- **Residual deviance**: 13673, on 32898 degrees of freedom.
- **AIC**: 13779, suggesting the model's relative quality.
- The number of Fisher Scoring iterations: 10.

The significant variables indicate which factors are influential in predicting whether a client will subscribe to a term deposit. Variables such as job type, contact method, campaign, month, duration of contact, and economic indicators play crucial roles in this prediction.

Code:

```
# Predict and evaluate Logistic Regression model

logistic_pred <- predict(logistic_model, newdata = dataTest, type = "response")

logistic_pred_class <- ifelse(logistic_pred > 0.5, 1, 0)


# Confusion matrix for Logistic Regression
```

```
logistic_conf_matrix <- confusionMatrix(as.factor(logistic_pred_class), dataTest$y)
logistic_conf_matrix
```

Result:

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7115  543
##          1  194  385
##
##                Accuracy : 0.9105
##                  95% CI : (0.9042, 0.9166)
##     No Information Rate : 0.8873
##     P-Value [Acc > NIR] : 3.635e-12
##
##                   Kappa : 0.4646
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9735
##             Specificity : 0.4149
##          Pos Pred Value : 0.9291
##          Neg Pred Value : 0.6649
##              Prevalence : 0.8873
##          Detection Rate : 0.8638
##    Detection Prevalence : 0.9297
##       Balanced Accuracy : 0.6942
##
```

```
##        'Positive' Class : 0
##
```

Interpretation:

Confusion Matrix Analysis:

The logistic regression model's performance is summarized in the confusion matrix and associated statistics. The matrix shows the counts of true positives, true negatives, false positives, and false negatives. Specifically, there are 7115 true negatives (correctly predicted non-subscribers), 543 false negatives (actual subscribers incorrectly predicted as non-subscribers), 194 false positives (non-subscribers incorrectly predicted as subscribers), and 385 true positives (correctly predicted subscribers).

Key Performance Metrics:

The model achieves an accuracy of 91.05%, meaning it correctly classifies whether a client will subscribe to a term deposit 91.05% of the time. The 95% confidence interval for accuracy ranges from 90.42% to 91.66%, indicating a high level of confidence in the model's performance. The No Information Rate, which is the accuracy obtained by always predicting the majority class (no subscription), is 88.73%. The P-Value [Acc > NIR] is 3.635e-12, showing that the model's accuracy is significantly better than random guessing.

The Kappa statistic, which measures agreement between predicted and actual classes adjusted for chance, is 0.4646, indicating moderate agreement. McNemar's Test P-Value is less than 2.2e-16, suggesting a significant difference between the number of false positives and false negatives, highlighting potential areas for model improvement.

Sensitivity, Specificity, and Predictive Values:

Sensitivity, or the true positive rate, is very high at 97.35%, indicating that the model is excellent at identifying clients who will not subscribe to a term deposit. However, the specificity, or true negative rate, is quite low at 41.49%, suggesting that the model struggles to correctly identify clients who will subscribe. This imbalance indicates that while the model is good at predicting non-subscribers, it has difficulty with subscribers.

The positive predictive value (PPV) is 92.91%, meaning that when the model predicts no subscription, it is correct 92.91% of the time. The negative predictive value (NPV) is 66.49%, indicating that when the model predicts a subscription, it is correct about two-thirds of the time. These values show that the model is more reliable in predicting non-subscribers than subscribers.

Prevalence and Detection Rates:

The prevalence of non-subscribers in the dataset is 88.73%. The detection rate, which is the proportion of actual positives detected by the model, is 86.38%. The detection prevalence, or the proportion of predicted positives, is 92.97%. These metrics provide insight into the model's ability to capture the actual distribution of classes within the data.

Balanced Accuracy:

The balanced accuracy, which is the average of sensitivity and specificity, is 69.42%. This metric provides a more nuanced view of the model's performance, taking into account the imbalance between sensitivity and specificity. While the model excels at identifying non-subscribers, its lower specificity suggests that there is room for improvement in accurately predicting subscribers.

Code:

```
# AUC-ROC for Logistic Regression
logistic_roc <- roc(dataTest$y, logistic_pred)
```

Result:

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

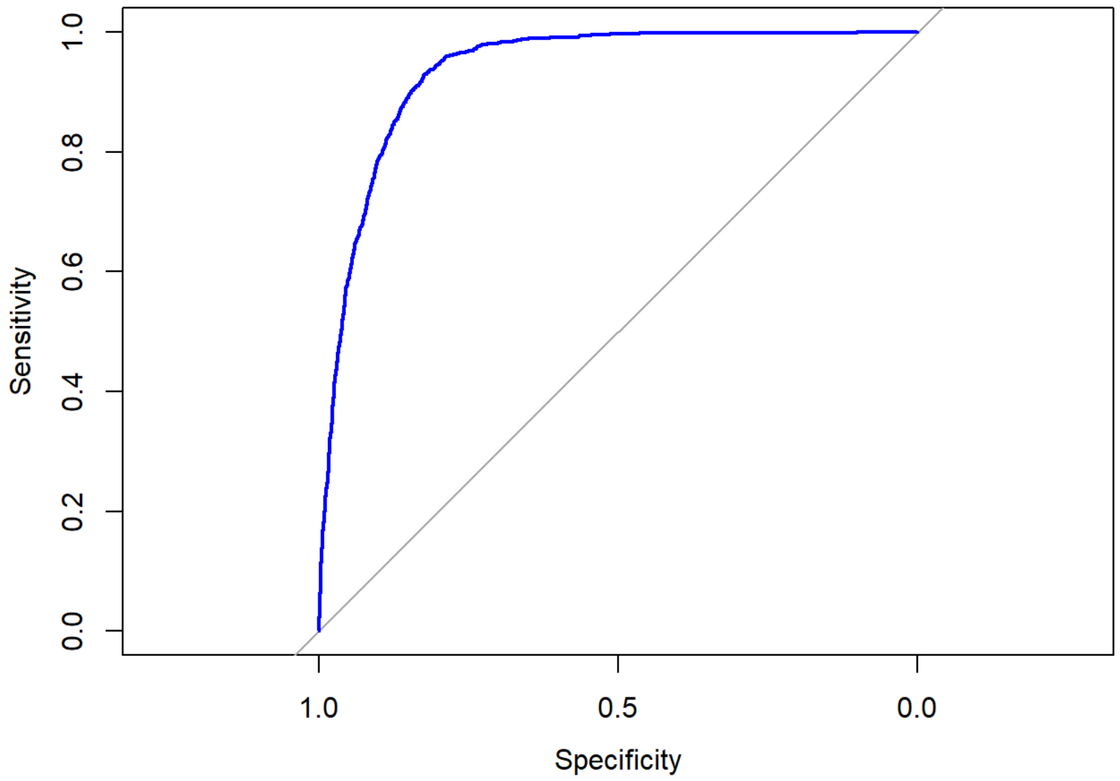
Interpretation:

The Area Under the Receiver Operating Characteristic (AUC-ROC) curve is calculated and plotted for the logistic regression model to evaluate its performance in distinguishing between the two classes (subscribing and not subscribing to a term deposit).

- The ROC curve helps visualize the model's ability to discriminate between the two classes.
- The AUC (Area Under the Curve) value, which will be calculated in the next step, quantifies this ability. An AUC of 0.5 indicates no discriminative ability (random guessing), while an AUC of 1 indicates perfect discrimination.
- A high AUC value close to 1 signifies that the model has a strong ability to distinguish between clients who will subscribe to a term deposit and those who will not.

Code:

```
plot(logistic_roc, col = "blue")
```



Interpretation:

The ROC (Receiver Operating Characteristic) curve for the logistic regression model is displayed in the plot.

The x-axis represents **1 - Specificity** (False Positive Rate), which is the proportion of actual negatives that are incorrectly identified as positives. The y-axis represents **Sensitivity** (True Positive Rate), which is the proportion of actual positives correctly identified by the model.

The blue curve represents the ROC curve of the logistic regression model. It plots the trade-off between sensitivity and specificity at various threshold levels. The closer the ROC curve follows the left-hand border and then the top border of the ROC space, the more accurate the model.

The diagonal line represents a random classifier with no discriminative ability (AUC = 0.5). Points along this line indicate that the model's predictions are no better than random chance.

The ROC curve is well above the diagonal line, indicating that the logistic regression model performs significantly better than random guessing. The curve approaches the top left corner, suggesting high sensitivity and specificity, particularly at lower thresholds. The area under the curve (AUC) will quantify this performance. An AUC closer to 1 indicates excellent model performance. The logistic regression model has strong discriminative power, effectively distinguishing between clients who will and will not subscribe to a term deposit. The high placement of the ROC curve reflects the model's ability to achieve high sensitivity while maintaining reasonable specificity, making it a reliable predictor in this context.

Code:

```
auc(logistic_roc)
```

Result:

```
## Area under the curve: 0.9353
```

The Area Under the Curve (AUC) for the logistic regression model's ROC curve is 0.9353. This value is a key metric that quantifies the overall ability of the model to discriminate between the two classes: clients who will subscribe to a term deposit and those who will not.

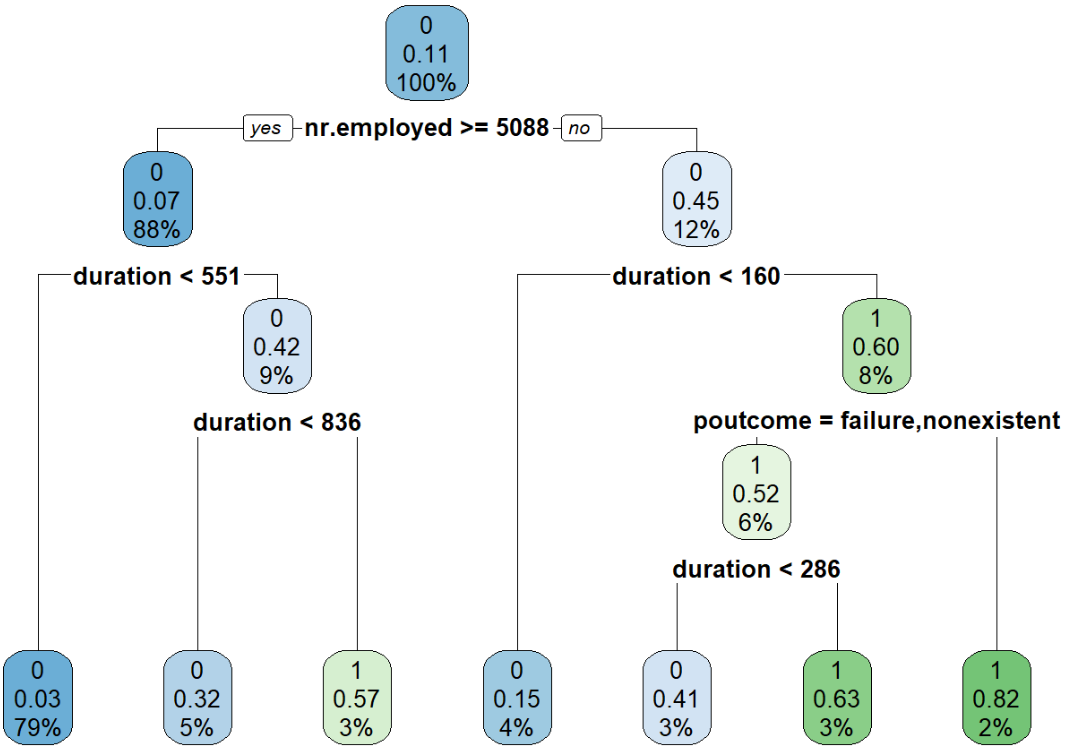
The AUC value ranges from 0 to 1. An AUC of 0.5 indicates a model with no discriminative ability, equivalent to random guessing. In contrast, an AUC closer to 1 signifies excellent discriminative ability. With an AUC of 0.9353, the logistic regression model demonstrates a very high level of performance. A high AUC value of 0.9353 suggests that the logistic regression model has a strong capacity to correctly distinguish between clients who are likely to subscribe to a term deposit and those who are not. This means that the model is able to achieve a good balance between sensitivity (true positive rate) and specificity (true negative rate) across various threshold settings.

Decision Tree

```
# Decision Tree model

tree_model <- rpart(y ~ ., data = dataTrain, method = "class")

rpart.plot(tree_model)
```



The tree splits the data at each node based on the most significant feature that reduces impurity, ultimately leading to a decision (0 for no subscription and 1 for subscription).

nr.employed: The number of employed individuals is a crucial predictor. A higher number of employed individuals generally leads to no subscription, indicating economic stability might reduce the need for term deposits.

Duration: The length of the call significantly influences the prediction. Longer calls are more likely to result in a subscription, possibly indicating more effective communication or interest from the client.

Previous Outcome: The past outcome of marketing efforts (poutcome) plays a role. If the previous outcome was a failure or nonexistent and the call duration is adequately long, the chances of subscription increase.

The decision tree effectively captures these patterns, allowing for a clear understanding of how various features interact to influence the likelihood of subscribing to a term deposit.

```
# Predict and evaluate Decision Tree model

tree_pred <- predict(tree_model, newdata = dataTest, type = "class")

tree_conf_matrix <- confusionMatrix(tree_pred, dataTest$y)

tree_conf_matrix
```

```
# AUC-ROC for Decision Tree

tree_pred_prob <- predict(tree_model, newdata = dataTest, type = "prob")[,2]

tree_roc <- roc(dataTest$y, tree_pred_prob)

plot(tree_roc, col = "red")

auc(tree_roc)
```

Interpretation:

The performance of the decision tree model can be evaluated using the confusion matrix and the AUC-ROC curve. The confusion matrix shows that the model correctly classified 7055 instances as class 0 and 458 instances as class 1. However, it also misclassified 254 instances as class 0 and 470 instances as class 1. The overall accuracy of the model is 91.21%, with a 95% confidence interval ranging from 90.58% to 91.81%. This high accuracy indicates that the model performs well in predicting the target variable.

The sensitivity of the model, which measures the proportion of actual positives correctly identified, is 96.52%. This means the model is highly effective at identifying instances of class 0. However, the specificity, which measures the proportion of actual negatives correctly identified, is relatively low at 49.35%. This indicates that the model is less effective at identifying instances of class 1, which may be due to an imbalance in the dataset where class 0 is more prevalent. The positive predictive value (precision) is 93.75%, indicating that when the model predicts class 0, it is correct 93.75% of the time. The negative predictive value is 64.33%, indicating that when the model predicts class 1, it is correct 64.33% of the time.

The balanced accuracy, which considers both sensitivity and specificity, is 72.94%, showing that the model performs moderately well across both classes. The Kappa statistic, which measures the agreement between the predicted and actual classifications, is 0.5107, indicating a moderate level of agreement.

The AUC-ROC curve provides a graphical representation of the model's performance across different threshold values. The area under the curve (AUC) is 0.8724, indicating that the model has a high ability to distinguish between the two classes. The AUC value closer to 1 signifies a better-performing model. The ROC curve itself, plotted in red, shows a steep rise towards the top-left corner, reflecting the high sensitivity of the model. Overall, the decision tree model demonstrates strong performance in terms of accuracy and sensitivity, with some room for improvement in specificity. The AUC-ROC value further supports the model's effectiveness in distinguishing between the two classes.

```
# Display metrics for Logistic Regression

cat("Logistic Regression Metrics:\n")

## Logistic Regression Metrics:

cat("Accuracy: ", logistic_conf_matrix$overall['Accuracy'], "\n")
```

```
## Accuracy:  0.9105257

cat("Precision: ", logistic_conf_matrix$byClass['Pos Pred Value'], "\n")

## Precision:  0.9290938

cat("Recall: ", logistic_conf_matrix$byClass['Sensitivity'], "\n")

## Recall:  0.9734574

cat("F1 Score: ", logistic_conf_matrix$byClass['F1'], "\n")

## F1 Score:  0.9507583

cat("AUC: ", auc(logistic_roc), "\n")

## AUC:  0.9353211
```

The performance metrics for the Logistic Regression model are as follows:

- **Accuracy**: The model achieves an accuracy of approximately 91.05%, indicating that it correctly classifies 91.05% of the instances.
- **Precision**: The precision is around 92.91%, meaning that when the model predicts a positive class (subscription), it is correct 92.91% of the time.
- **Recall**: The recall (sensitivity) is high at 97.35%, indicating that the model correctly identifies 97.35% of the actual positive instances.
- **F1 Score**: The F1 score, which balances precision and recall, is 95.08%, reflecting the model's strong performance in both metrics.
- **AUC**: The area under the ROC curve (AUC) is 0.9353, suggesting that the model has a very good ability to distinguish between the positive and negative classes.

```
# Display metrics for Decision Tree

cat("Decision Tree Metrics:\n")

## Decision Tree Metrics:

cat("Accuracy: ", tree_conf_matrix$overall['Accuracy'], "\n")

## Accuracy:  0.9121039

cat("Precision: ", tree_conf_matrix$byClass['Pos Pred Value'], "\n")

## Precision:  0.9375415

cat("Recall: ", tree_conf_matrix$byClass['Sensitivity'], "\n")

## Recall:  0.9652483
```

```
cat("F1 Score: ", tree_conf_matrix$byClass['F1'], "\n")
## F1 Score:  0.9511932
cat("AUC: ", auc(tree_roc), "\n")
## AUC:  0.8723852
```

The performance metrics for the Decision Tree model are as follows:

- **Accuracy**: The model achieves an accuracy of approximately 91.21%, indicating that it correctly classifies 91.21% of the instances.
- **Precision**: The precision is around 93.75%, meaning that when the model predicts a positive class (subscription), it is correct 93.75% of the time.
- **Recall**: The recall (sensitivity) is 96.52%, indicating that the model correctly identifies 96.52% of the actual positive instances.
- **F1 Score**: The F1 score, which balances precision and recall, is 95.12%, reflecting the model's strong performance in both metrics.
- **AUC**: The area under the ROC curve (AUC) is 0.8724, suggesting that the model has a good ability to distinguish between the positive and negative classes.

Conclusion:

Which is a better model – Logistic Regression

- **Accuracy**: Both models have similar accuracy, with the Decision Tree being slightly higher.
- **Precision**: The Decision Tree has a slightly higher precision than Logistic Regression, meaning it is slightly better at predicting true positives.
- **Recall**: Logistic Regression has a higher recall, indicating it is better at identifying actual positive instances.
- **F1 Score**: Both models have very similar F1 scores, with the Decision Tree being marginally higher.
- **AUC**: Logistic Regression has a significantly higher AUC, indicating better overall performance in distinguishing between the classes.

Conclusion:

While both models perform well, Logistic Regression slightly edges out the Decision Tree in terms of AUC and recall, making it a better model for this specific task. Logistic regression outperforms the decision tree in terms of AUC (0.94 vs. 0.87), indicating better overall classification performance and ability to discriminate between classes. The higher AUC indicates that Logistic Regression has a better ability to discriminate between the positive and negative classes, and the higher recall suggests it is better at identifying true positives. Therefore, Logistic Regression would be the preferred model.