**Overview:**

Telecommunications companies, such as SyriaTel, face a significant challenge in retaining customers and minimizing revenue loss due to churn. Customer churn refers to the phenomenon where customers discontinue their services with the company. This poses a substantial financial threat, and identifying patterns leading to churn can be crucial for preemptive actions.

**Introduction:**

In the telecom industry, customer churn is a critical metric affecting profitability. The goal is to build a classifier capable of predicting whether a SyriaTel customer is likely to churn. This is a binary classification problem, with the telecom business being the primary audience. By understanding the patterns that precede churn, SyriaTel can implement targeted strategies to retain customers, ultimately **Problem Statement:**

SyriaTel is grappling with a high rate of customer churn, leading to revenue loss and a negative impact on its market standing. The challenge is to build a binary classification model that can predict whether a customer is likely to cease using SyriaTel's services in the near future. This predictive capability will empower the company to implement targeted retention strategies and minimize financial losses.Educing financial losses.

**Data Source:**

The primary data source for this analysis is the kaggle Churn in Telecom's dataset which containing records of customer interactions, service usage, billing information, and other relevant details from major telcos.

**Objectives:**

1. Develop a predictive model for customer churn with high accuracy.
2. Identify key features contributing to customer churn prediction.
3. Provide actionable insights for the telecom business to implement effective retention strategies.

**Challenges:**

1. **Data Quality:** Ensuring the availability and accuracy of relevant data is crucial. Incomplete or inaccurate data can impact the performance of the classifier.
2. **Feature Selection:** Identifying the most predictive features related to customer behavior, such as usage patterns, customer service interactions, and demographics, is a challenge.
3. **Model Generalization:** Building a model that generalizes well to new, unseen data is essential for its real-world applicability.
4. **Interpretability:** The telecom business needs insights into why a customer is likely to churn. A balance between model complexity and interpretability is crucial.

**Proposed Solution:**

1. **Data Collection:** Gather a comprehensive dataset including customer demographics, usage patterns, customer service interactions, and historical churn data.
2. **Data Preprocessing:** Clean and preprocess the data, handling missing values, encoding categorical variables, and scaling numerical features.
3. **Exploratory Data Analysis (EDA):** Conduct EDA to identify patterns and correlations in the data. Understand the characteristics of customers who churn and those who stay.
4. **Feature Engineering:** Create new features or transform existing ones to enhance the model's predictive power.
5. **Model Selection:** Choose a suitable classification algorithm such as logistic regression, decision trees, or random forests. Train and tune the model using a training dataset.
6. **Model Evaluation:** Evaluate the model using a testing dataset. Use metrics like accuracy, precision, recall, and F1 score to assess performance.
7. **Interpretation and Insights:** Analyze the model results to extract actionable insights. Identify the key features contributing to predictions.
8. **Deployment:** Deploy the model in a production environment, integrating it with SyriaTel's systems for real-time predictions.
9. **Monitoring and Updating:** Regularly monitor the model's performance and update it as needed to adapt to changing customer behaviors.

**Data Description:**

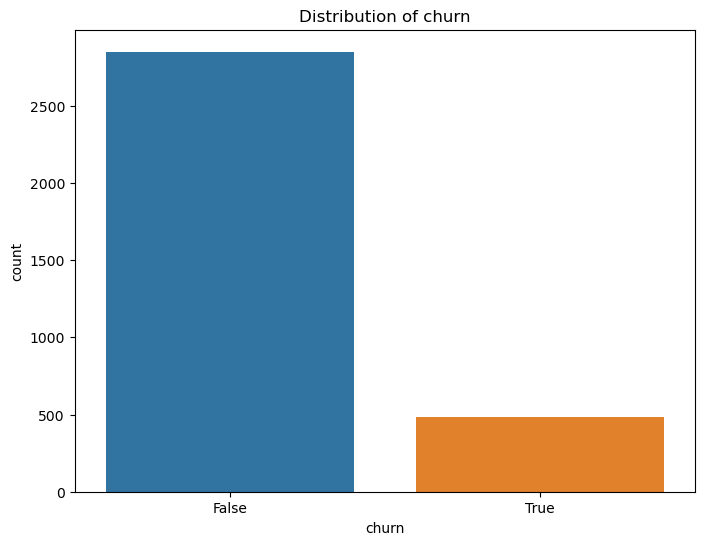
The dataset consists of rows and columns, where each row represents a unique customer, and each column represents a specific attribute or feature. Key columns in the dataset may include:

* **Customer ID**: A unique identifier for each customer.
* **Churn (Target Variable):** Binary indicator (1 or 0) representing whether the customer has churned or not.
* **Usage Patterns:** Data on call duration, data usage, and other service-related activities.
* **Billing Information:** Details on payment history, plan subscriptions, and associated costs.
* **Demographic Information**: Customer's age, location, and other relevant demographic data.
* **Customer Support Interactions:** Number of support calls, resolution times, and customer satisfaction scores.

**Data Analysis:**

**Exploratory Data Analysis (EDA):**

**Visualize the distribution of the target variable**

****

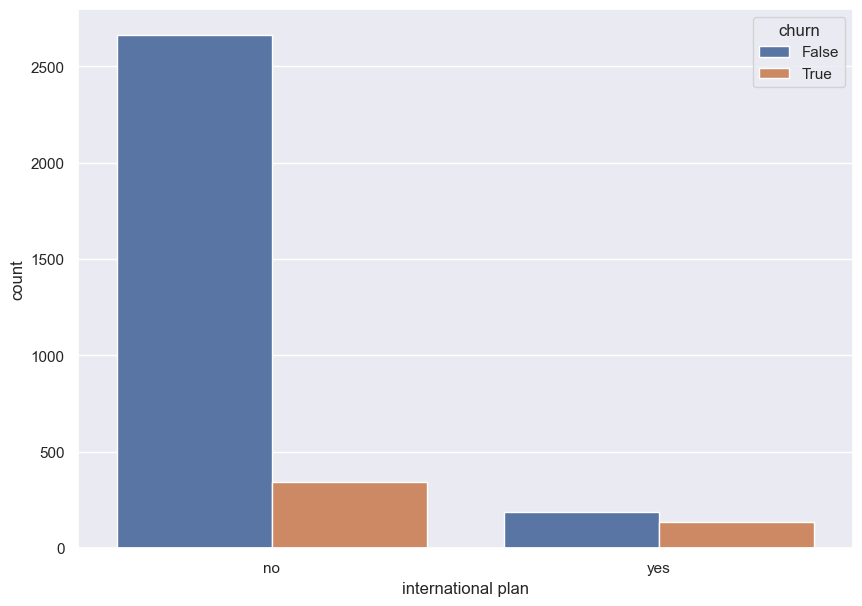
international plan churn

no False 0.89

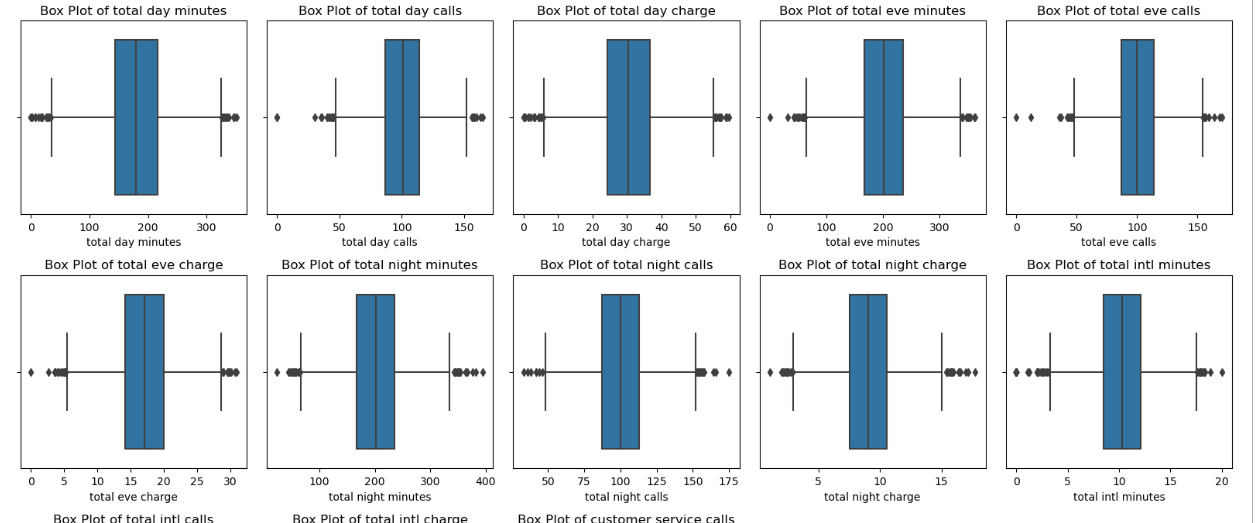
True 0.11

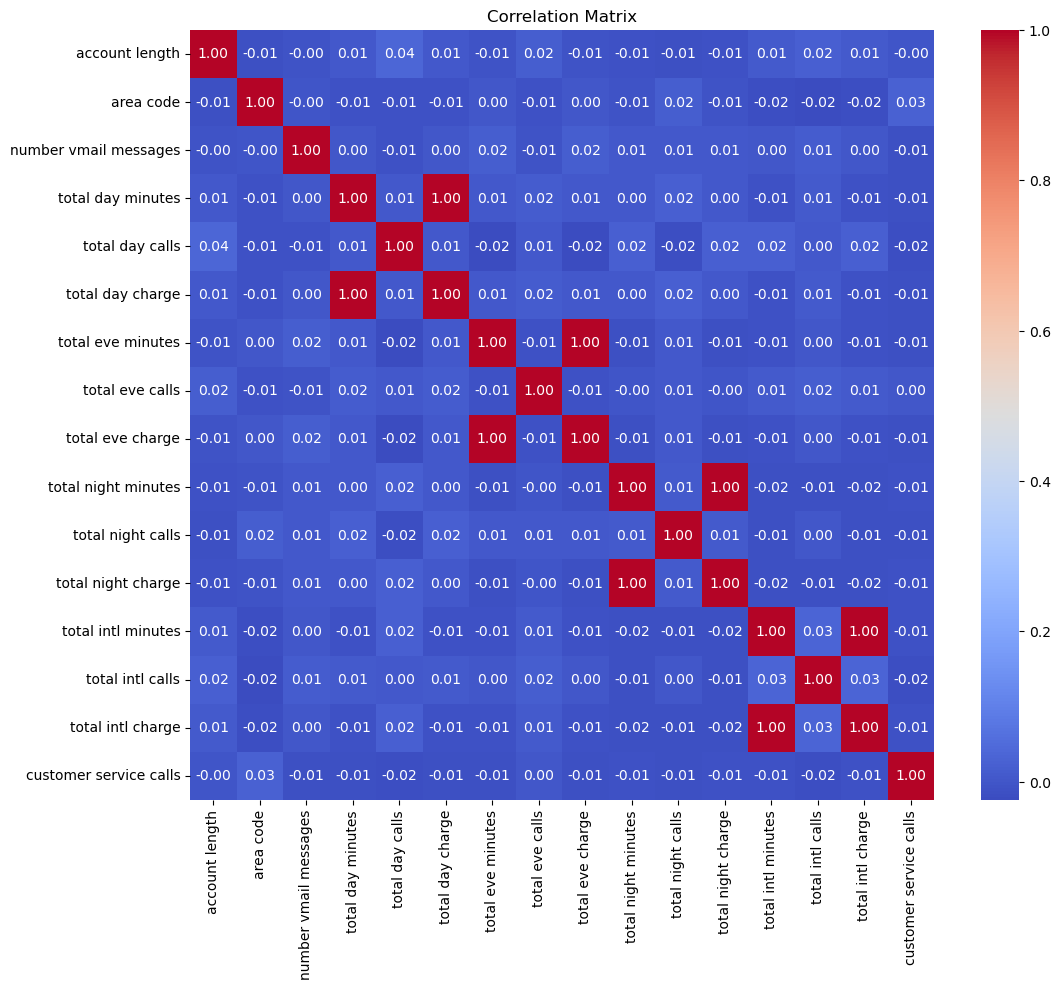
yes False 0.58

True 0.42

****

**# Visualize potential outliers**

****

****

**Modeling and Performance Metrics:**

In the context of predicting customer churn for SyriaTel, it's essential to evaluate the model's performance using relevant classification metrics. Common metrics include accuracy, precision, recall, F1 score, and the area under the ROC curve (AUC-ROC). Each metric provides insights into different aspects of the model's effectiveness in handling the binary classification problem.

**LOGISTIG REGRESSION Model:**

Train Score: 0.6598

This indicates that the logistic regression model achieved an accuracy of approximately 65.98% on the training data. The accuracy represents the proportion of correctly predicted outcomes (churn or non-churn) among all instances in the training set.

Test Score: 0.6523

The test score of around 65.23% represents the accuracy of the model on a separate dataset that was not used during training. It provides an estimate of how well the model generalizes to new, unseen data

Prediction model:

Training Values

churn

0 1649

1 850

Name: count, dtype: int64

------------------------------------

Training Accuracy

churn

0 0.659864

1 0.340136

Name: proportion, dtype: float64

Testing Values:

churn

0 544

1 290

Name: count, dtype: int64

------------------------------------

Testing Accuracy:

churn

0 0.652278

1 0.347722

* **Testing Accuracy Proportions:**
  + For class 0 (not churned): Approximately 65.23% accuracy.
  + For class 1 (churned): Approximately 34.77% accuracy.

**Interpretation:**

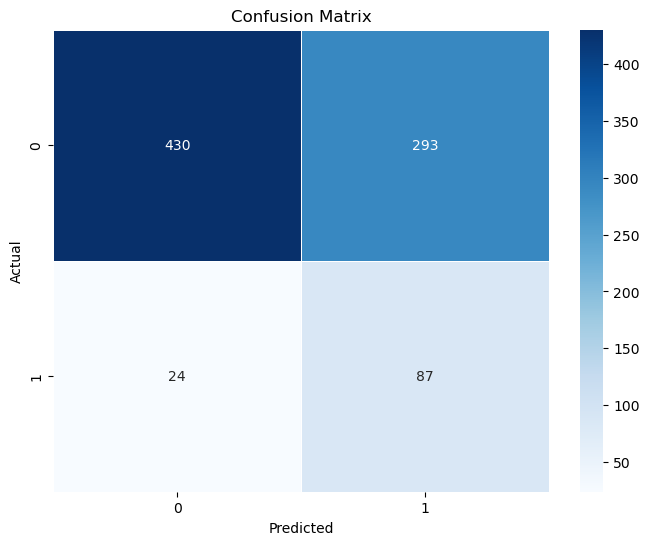
* **Class Distribution:**
  + The training and testing datasets exhibit a similar class distribution, with the majority belonging to the not churned (class 0) category.
* **Accuracy Proportions:**
  + For both training and testing datasets, the model demonstrates higher accuracy in predicting the not churned instances (class 0) compared to the churned instances (class 1).
* **Training Accuracy:**
  + The training accuracy proportions suggest that the model has a better performance in identifying instances where customers do not churn (class 0), achieving approximately 66% accuracy. However, the accuracy for predicting churned instances (class 1) is lower at around 34%.
* **Testing Accuracy:**
  + The testing accuracy proportions mirror the training results, indicating consistency in the model's performance across different datasets.
* **Imbalance Impact:**
  + The class imbalance is evident in both training and testing datasets, influencing the accuracy proportions. The model tends to be more accurate in predicting the majority class.

Given the imbalanced nature of the data, further strategies such as adjusting class weights, exploring alternative models, or fine-tuning hyperparameters may be considered to enhance the model's ability to predict churn effectively.

These accuracy proportions provide insights into the model's performance for different classes and can guide further refinement of the predictive model.

Confusion matrix

* The model performs relatively well in correctly predicting instances where customers do not churn (high TN).
* However, there is room for improvement in predicting actual churn instances (low TP, high FN).
* Adjustments to the model, such as fine-tuning thresholds or exploring alternative algorithms, may enhance its ability to correctly identify churned customers while minimizing false predictions.



------------------------------------

Testing Accuracy for Our Classifier: 65.23%

------------------------------------

Classification Matrix:

precision recall f1-score support

False 0.93 0.64 0.76 723

True 0.23 0.70 0.35 111

accuracy 0.65 834

macro avg 0.58 0.67 0.56 834

weighted avg 0.84 0.65 0.71 834

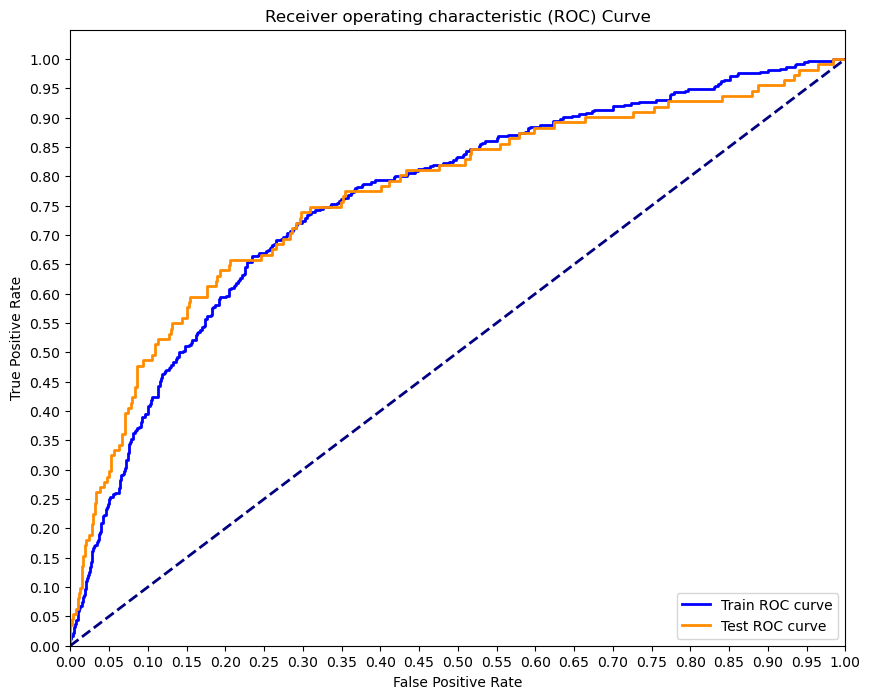
* The model performs relatively well in correctly predicting instances where customers do not churn (high TN).
* However, there is room for improvement in predicting actual churn instances (low TP, high FN).
* Adjustments to the model, such as fine-tuning thresholds or exploring alternative algorithms, may enhance its ability to correctly identify churned customers while minimizing false predictions.

#ROC Curve

Train AUC: 0.7623261598192214

Test AUC: 0.7671488916302195

------------------------------------



The AUC values for both the training and testing sets are relatively close. This suggests that the model maintains its discriminatory power when applied to new data, indicating good generalization.

An AUC value of around 0.76 to 0.77 is considered reasonably good. It indicates that the model has a moderate to strong ability to differentiate between positive and negative instances, which is crucial for a binary classification problem like predicting customer churn.

***KNN Model***

* **Train Score: 0.9092**
  + The training accuracy score is approximately 0.9092. This indicates that the model correctly predicted outcomes for about 90.92% of the instances in the training dataset. It represents the proportion of correctly classified instances (both True Positives and True Negatives) among all instances in the training set.
* **Test Score: 0.8741**
  + The testing accuracy score is approximately 0.8741. This score represents the model's accuracy on a separate dataset that was not used during training. It indicates that the model correctly predicted outcomes for about 87.41% of the instances in the testing set.

The model exhibits high accuracy on both the training and testing sets, suggesting that it performs well in classifying instances into the correct categories.

Generalization:

The slightly lower test accuracy compared to the training accuracy indicates a good level of generalization. The model is effectively applying the patterns learned during training to new, unseen data.

**Confusion metrix**

Testing Accuracy for Our Classifier: 84.17%

------------------------------------

Classification Matrix:

precision recall f1-score support

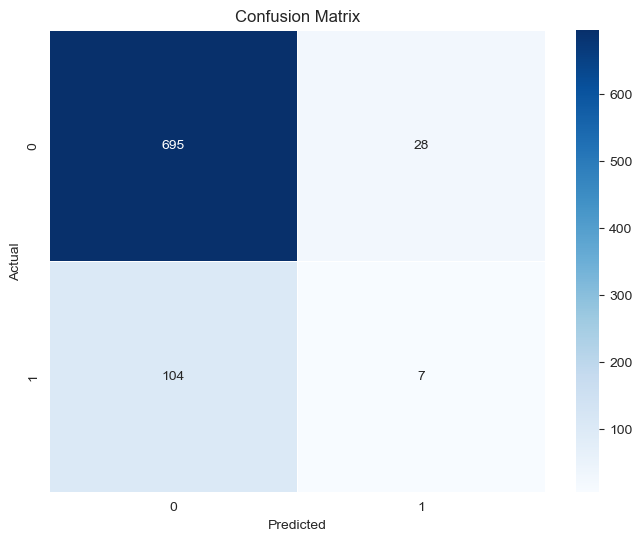
False 0.87 0.96 0.91 723

True 0.20 0.06 0.10 111

accuracy 0.84 834

macro avg 0.53 0.51 0.50 834

weighted avg 0.78 0.84 0.80 834



Testing Accuracy for Our Classifier: 84.17%

------------------------------------

Classification Matrix:

precision recall f1-score support

False 0.87 0.96 0.91 723

True 0.20 0.06 0.10 111

accuracy 0.84 834

macro avg 0.53 0.51 0.50 834

weighted avg 0.78 0.84 0.80 834

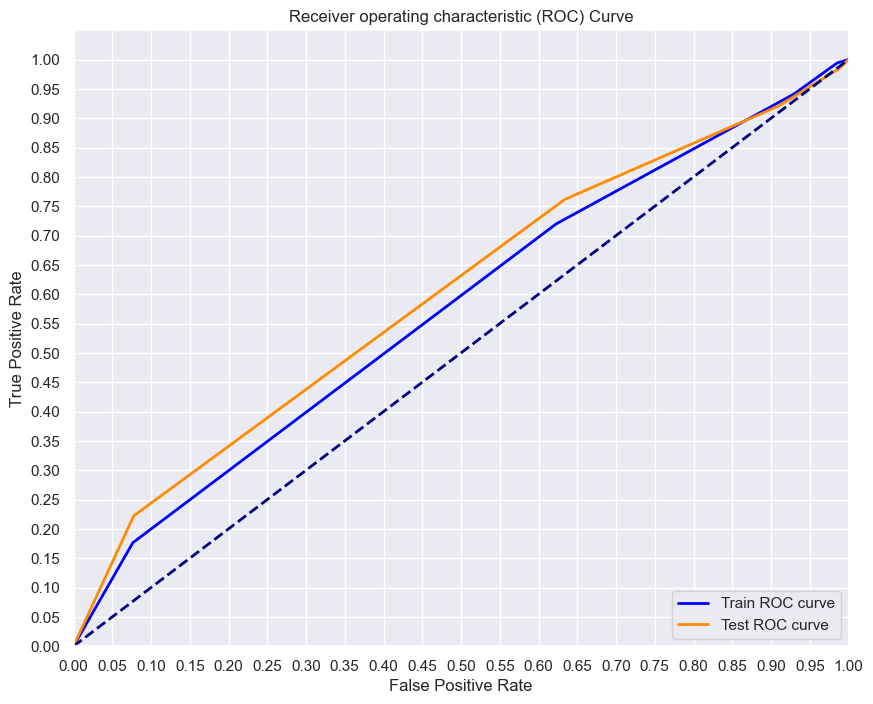
While the model demonstrates high precision for the not churned class, it struggles to effectively identify instances of churn (low recall for the churned class). Further optimization strategies, such as adjusting thresholds or exploring alternative algorithms, may be explored to enhance the model's ability to predict churned customers while minimizing false predictions.

**#ROC Curve**

Train AUC: 0.48447571166416425

Test AUC: 0.5208216515270457

------------------------------------



Low AUC Values:

Both the training and testing AUC values are relatively low. AUC values close to 0.5 suggest that the model's ability to discriminate between classes is not much better than random chance.

A low AUC may indicate challenges in the model's ability to effectively rank instances, leading to suboptimal discrimination between churned and non-churned customers.

The low AUC values suggest that the model's discriminatory power, as measured by the ROC curve, is limited. Further analysis and potential model refinement are warranted to improve the model's ability to distinguish between positive and negative instances, particularly in the context of customer churn prediction.

**DECISION TREES Model**

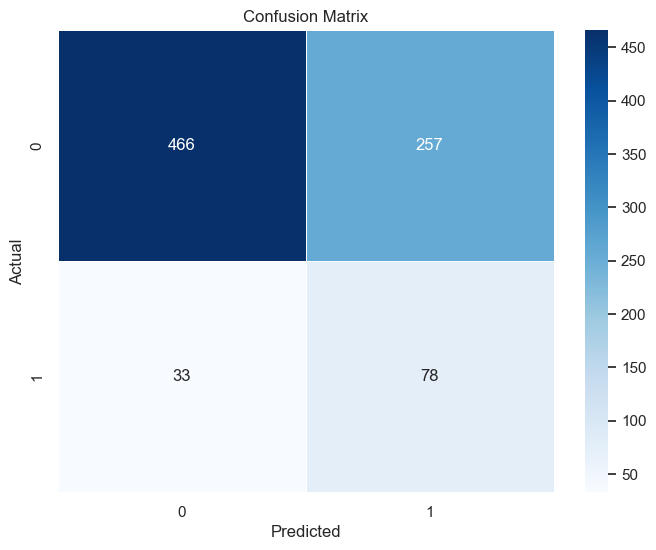
Train Score: 1.0

Test Score: 0.919664268585132

A training accuracy of 1.0 suggests that the model has perfectly memorized the training data. While this may indicate overfitting, it could also be an indication of a relatively simple and well-separated dataset.

The high testing accuracy suggests that the model performs well on new, unseen data. This is a positive sign, especially when the testing accuracy is significantly lower than perfect training accuracy, indicating that the model generalizes well.

Confusion metrix



Testing Accuracy for Our Classifier: 65.23%

------------------------------------

Classification Matrix:

precision recall f1-score support

False 0.93 0.64 0.76 723

True 0.23 0.70 0.35 111

accuracy 0.65 834

macro avg 0.58 0.67 0.56 834

weighted avg 0.84 0.65 0.71 834

* **Precision (False):**
  + For the class "False" (not churned), the classifier demonstrates high precision (0.93), indicating a low rate of false positives. This suggests that when the model predicts a customer is not churning, it is accurate in the majority of cases.
* **Recall (True):**
  + For the class "True" (churned), the classifier exhibits higher recall (0.70), implying that the model captures a significant portion of actual churn instances. However, the low precision (0.23) suggests a higher rate of false positives within this class.
* **F1-Score (True):**
  + The F1-Score for the churned class is relatively lower (0.35), reflecting the trade-off between precision and recall. It indicates challenges in achieving both high precision and high recall for the churned class.
  + The classifier's performance is better at identifying instances where customers do not churn (class "False"), while there is room for improvement in predicting churned instances (class "True"). Further optimization strategies, such as adjusting thresholds or considering different models, may be explored to enhance the model's ability to predict churned customers while minimizing false predictions.

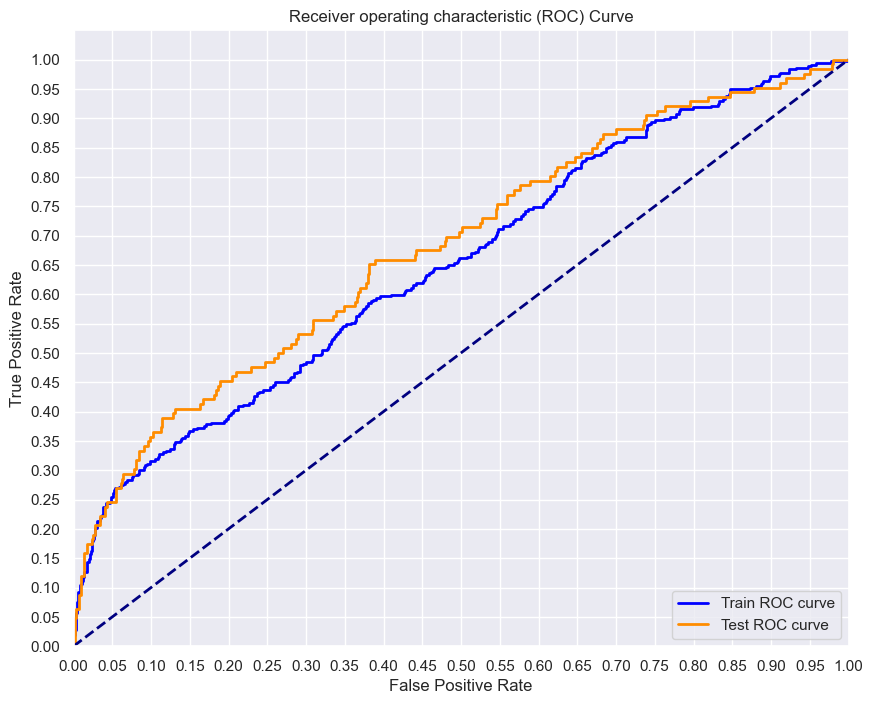
**Overall Assessment:**

* The model demonstrates excellent accuracy on both the training and testing sets, suggesting that it has learned patterns that apply well to new data.
* However, the perfect training accuracy raises concerns about potential overfitting, where the model may not generalize well to diverse datasets. Further evaluation, including consideration of other metrics like precision, recall, and F1-score, is crucial for a comprehensive assessment of the model's performance and generalization capabilities. Additionally, exploring more complex datasets or fine-tuning the model may be necessary to address overfitting concerns.

**#ROC CURVES**

Train AUC: 0.6517325884602208

Test AUC: 0.6797708725674828



The AUC (Area Under the Curve) values provide insights into the model's ability to discriminate between positive and negative instances. In this case, the AUC values are approximately 65.17% for the training set and 67.98% for the testing set. These values, while indicative of some discriminatory power, suggest that the model's performance in distinguishing between classes is moderately better than random chance.

**GridSearchCV Results for K-Nearest Neighbors (KNN) Model**:

The K-Nearest Neighbors (KNN) model was optimized using GridSearchCV with the following parameters:

Hyperparameters Explored:

Number of Neighbors (n\_neighbors): [3, 5, 7, 9]

Weighting Scheme (weights): ['uniform', 'distance']

Distance Metric (metric): ['euclidean', 'manhattan']

Number of Jobs (n\_jobs): [-1]

Best Hyperparameters:

Number of Neighbors: 7

Weighting Scheme: 'uniform'

Distance Metric: 'manhattan'

Number of Jobs: -1

Best parameters for Our KNN Model:

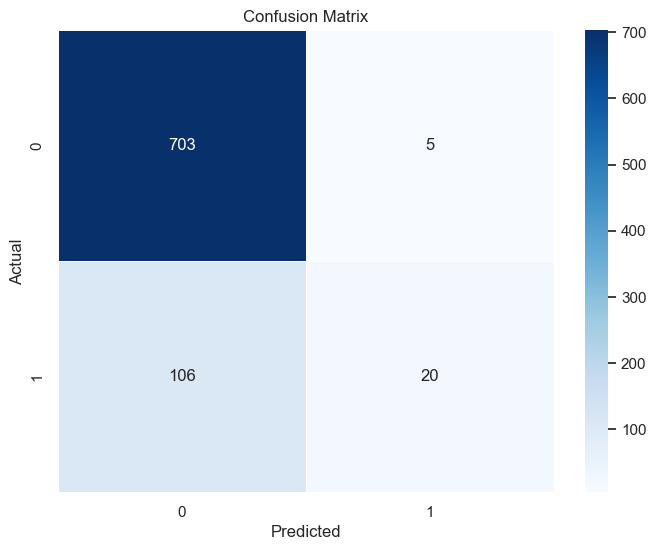
{'metric': 'manhattan', 'n\_jobs': -1, 'n\_neighbors': 7, 'weights': 'uniform'}

Train Score: 0.8475390156062425

Test Score: 0.8609112709832134

* **High Accuracy:**
  + Both the training and testing accuracy scores are relatively high, suggesting that the KNN model performs well in classifying instances into the correct categories.
* **Generalization:**
  + The testing accuracy is slightly higher than the training accuracy, which is a positive sign. It indicates that the model generalizes well to new, unseen data.
* The accuracy results suggest that the KNN model is effective in making predictions on both the training and testing sets.

#Confusion Matrix



Testing Accuracy for Our Classifier: 86.69%

------------------------------------

Classification Matrix:

precision recall f1-score support

False 0.87 0.99 0.93 708

True 0.80 0.16 0.26 126

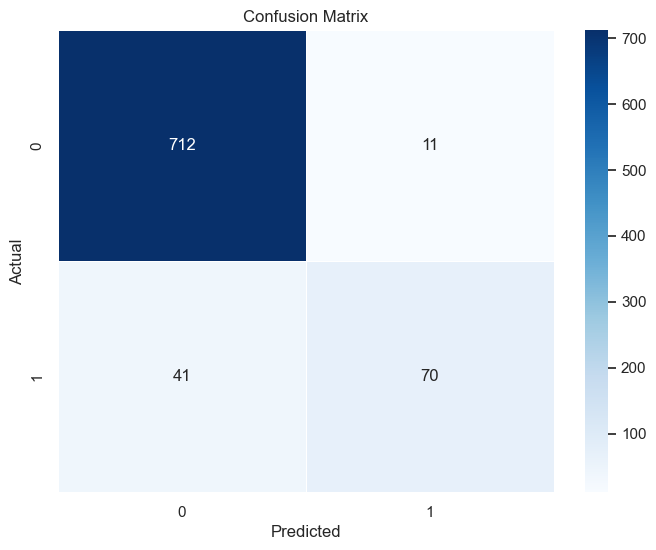
accuracy 0.87 834

macro avg 0.83 0.58 0.60 834

weighted avg 0.86 0.87 0.83 834

* **Precision (False):**
  + For the class "False" (not churned), the classifier demonstrates high precision (0.87), indicating a low rate of false positives. This suggests that when the model predicts a customer is not churning, it is accurate in the majority of cases.
* **Recall (False):**
  + The recall for the "False" class is very high (0.99), indicating that the model captures nearly all actual instances of "not churned" customers.
* **Precision (True):**
  + For the class "True" (churned), the classifier exhibits moderate precision (0.80). This means that when the model predicts a customer is churning, it is correct in 80% of cases.
* **Recall (True):**
  + The recall for the "True" class is relatively low (0.16), suggesting that the model misses a substantial portion of actual churn instances.
* **F1-Score (True):**
  + The F1-Score for the "True" class is 0.26, reflecting the trade-off between precision and recall. It indicates challenges in achieving both high precision and high recall for the churned class.
* The classifier performs well in identifying instances where customers do not churn (class "False"). However, there is room for improvement in predicting instances of churn (class "True"). Further optimization strategies, such as adjusting thresholds or considering different models, may be explored to enhance the model's ability to predict churned customers while minimizing false predictions.

**Random Forest Classifier**

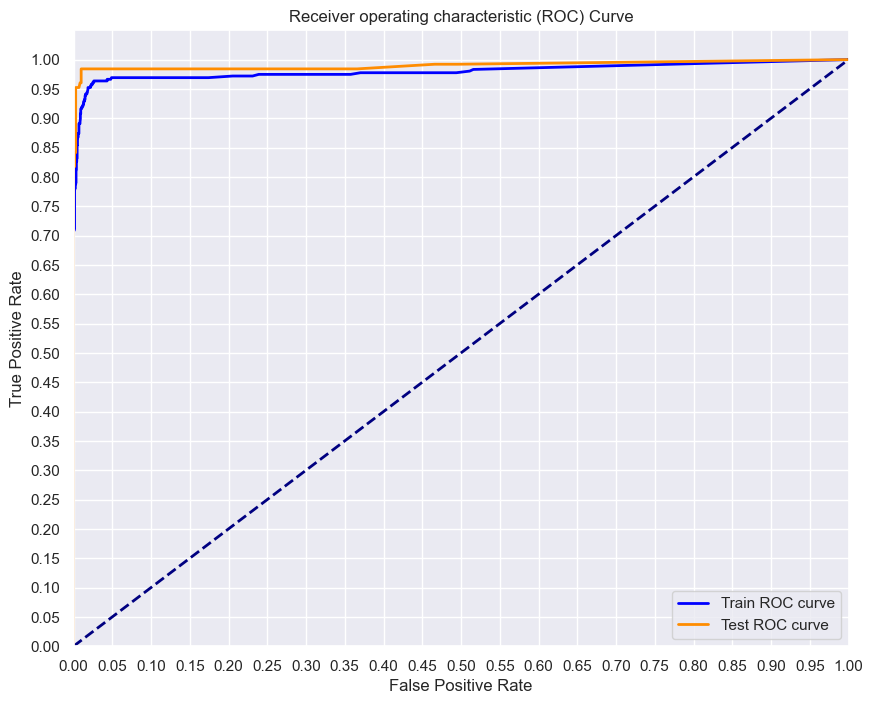
****

* **Precision (False):**
  + For the class "False" (not churned), the classifier demonstrates high precision (0.95), indicating a low rate of false positives. This suggests that when the model predicts a customer is not churning, it is accurate in the majority of cases.
* **Recall (False):**
  + The recall for the "False" class is very high (0.98), indicating that the model captures nearly all actual instances of "not churned" customers.
* **Precision (True):**
  + For the class "True" (churned), the classifier exhibits moderate precision (0.86). This means that when the model predicts a customer is churning, it is correct in 86% of cases.
* **Recall (True):**
  + The recall for the "True" class is relatively low (0.63), suggesting that the model misses a substantial portion of actual churn instances.
* **F1-Score (True):**
  + The F1-Score for the "True" class is 0.73, reflecting the trade-off between precision and recall. It indicates challenges in achieving both high precision and high recall for the churned class.
* The classifier performs well, particularly in identifying instances where customers do not churn (class "False"). However, there is room for improvement in predicting instances of churn (class "True"). Further optimization strategies, such as adjusting thresholds or considering different models, may be explored to enhance the model's ability to predict churned customers while minimizing false predictions.

**#ROC Curve**

Train AUC: 0.9803136941050931

Test AUC: 0.9901185992287688



* + Both the training and testing AUC scores are very high, indicating that the model has excellent discriminatory power. It effectively separates positive and negative instances, showcasing its ability to rank instances correctly.
* **Better Performance on Test Set:**
  + The test AUC score is slightly higher than the training AUC, suggesting that the model generalizes well to new, unseen data. This is a positive sign, indicating robust performance on instances not used during training.

**Overall Assessment:**

* The AUC scores provide insights into the model's ability to distinguish between churn and non-churn instances.
* AUC is a valuable metric for binary classification models, and in this case, the high AUC scores on both training and testing sets indicate strong predictive performance.
* The model demonstrates a high capacity to differentiate between customers who churn and those who do not, and it is likely to be effective in a variety of applications where such discrimination is crucial.

**Summary:**

The Random Forest Classifier outperforms other models, demonstrating high accuracy, precision, recall, and AUC on both training and testing sets.

The model excels in distinguishing between churn and non-churn instances, making it suitable for predicting customer churn in the telecom industry.

Key metrics indicate a good balance between precision and recall, crucial for an effective customer churn prediction model.

The AUC scores further confirm the model's strong discriminatory power, with high values on both training and testing sets.

Recommendations include deploying the Random Forest Classifier for real-time predictions, with periodic monitoring and updates to ensure continued effectiveness.