**10Pearls Pakistan**

**TELECOM CHURN PROJECT**

**ABSTRACT**

The Telecom Customer Churn project aims to address one of the most critical challenges in the telecommunications industry: customer retention. By leveraging advanced data science techniques, this project identifies key factors influencing customer churn and predicts potential churners with high accuracy. The project utilizes a structured dataset of 7032 entries and 21 features, including customer demographics, service usage, and billing information.

Exploratory Data Analysis (EDA) was conducted to understand data distributions and correlations, followed by feature engineering and preprocessing to handle missing values and categorical variables. The dataset’s inherent class imbalance was addressed using the SMOTEENN technique, ensuring balanced model training. Various machine learning algorithms, such as Decision Tree, Random Forest, Gradient Boosting, Ada Boost, Artifical Neural Networks and XGBoost, were evaluated, with XGBoost emerging as the most effective model.

The project integrates a MySQL database for efficient data management and retrieval, while a Flask-based web application facilitates model deployment. Additionally, an RAG (Retrieval-Augmented Generation) pipeline was implemented to enable dynamic SQL query generation and interaction, providing actionable insights into customer churn patterns. For database interaction, Google Gemini Pro was used to convert user questions into SQL queries, which are executed on a normalized MySQL database. This comprehensive solution offers predictive capabilities and supports strategic decision-making to minimize churn and maximize customer retention in the telecom sector.

# TABLE OF CONTENTS

**ABSTRACT ………………………………………………………………………….1**

[TABLE OF CONTENTS 2](#_Toc182857203)

[Introduction 3](#_Toc182857204)

[1.1. Purpose of the Project 3](#_Toc182857205)

[1.2. Scope and Objectives of the Project 3](#_Toc182857206)

[Methodology 4](#_Toc182857207)

[2.1. Introduction 4](#_Toc182857208)

[2.2. Proposed System Framework / Architecture 4](#_Toc182857209)

[2.3. Algorithm Used 12](#_Toc182857210)

[Results and Discussion 15](#_Toc182857211)

[3.1. Introduction 15](#_Toc182857212)

[3.2. Random Forest Classifier result: 15](#_Toc182857213)

[3.3. Decision Tree Classifier Results: 17](#_Toc182857214)

[3.3. Gradient Boost Results: 19](#_Toc182857215)

[3.4. Ada Boost Classifier Results: 23](#_Toc182857216)

[3.5. XGBoost Classifier Results: 25](#_Toc182857217)

[3.6. Artifical Neural Network Results: 30](#_Toc182857218)

[Implementation 33](#_Toc182857222)

[4.1. Introduction 33](#_Toc182857223)

[4.2. Flask App 33](#_Toc182857224)

[Conclusion 38](#_Toc182857225)

# Introduction

## Purpose of the Project

The primary purpose of the Telecom Customer Churn project is to help telecom companies understand the factors driving customer attrition and take proactive measures to retain customers. By identifying high-risk customers and understanding the patterns leading to churn, businesses can make informed decisions, reduce revenue loss, and improve customer satisfaction. This project also serves as a showcase of integrating data science techniques with software development to create a deployable solution that supports real-time predictions and insights.

## Scope and Objectives of the Project

* **Data Analysis**: Understanding the distribution, trends, and correlations within customer demographics, service usage, and account details.
* **Churn Prediction**: Building machine learning models to accurately predict the likelihood of churn, using techniques like SMOTEENN for balancing the dataset and XGBoost for enhanced predictive performance.
* **Database Management**: Creating a normalized MySQL database to store and manage customer-related data efficiently.
* **Web Application Development**: Deploying the churn prediction model via a Flask-based web app, enabling interactive predictions and data retrieval.
* **Generative AI Integration**: Implementing a Retrieval-Augmented Generation (RAG) pipeline to enhance the application with SQL-based insights generation, enabling dynamic query responses and actionable analytics.
  1. **Objective of the Project**
* To identify the key drivers of customer churn through data analysis and feature engineering.
* To develop and evaluate machine learning models for accurate churn prediction, with a focus on explainability and performance.
* To create a structured and scalable database system for managing customer data.
* To build a user-friendly web interface for deploying the churn prediction model and interacting with customer data.
* To integrate advanced Generative AI capabilities for query-based insights and predictions.
* To provide actionable recommendations for telecom companies to minimize churn and maximize customer retention.

# Methodology

## Introduction

The primary goal is to identify the most suitable machine learning model that will be used for churn prediction and EDAthen performing the cleaning process and normalizing the data. After this, it involves evaluating model performance using metrics Precision, Recall, Accuracy, Confusion Matrix, and ROC Curve to conclude which algorithm is best for the prediction of churn. Additionally, the process includes the development of a web application with considerations for front-end development, MySQL Database, and RAG Pipeline.

## Proposed System Framework / Architecture

The proposed system framework for the Telecom Customer Churn project is designed to integrate data processing, predictive modeling, database management, and web application functionalities. Below is the architecture:

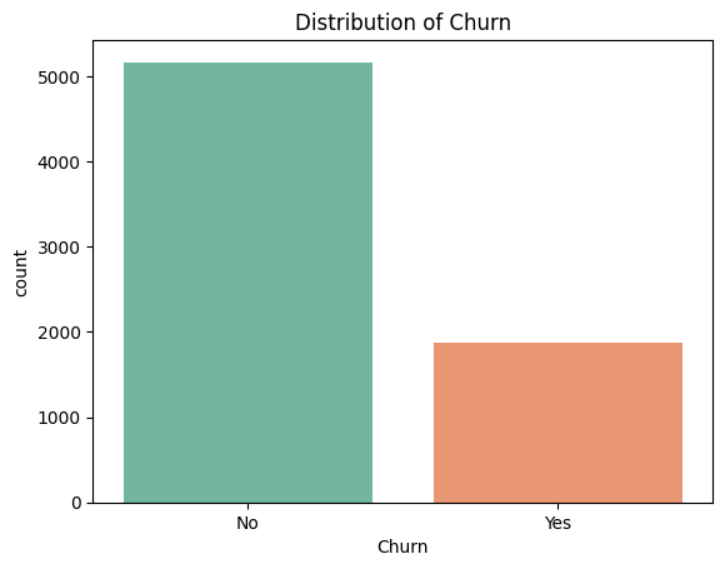
#### **1. Data Source Layer**

* **Dataset**: A structured dataset with 7032 rows and 21 columns containing customer demographics, account details, service usage, and churn information.

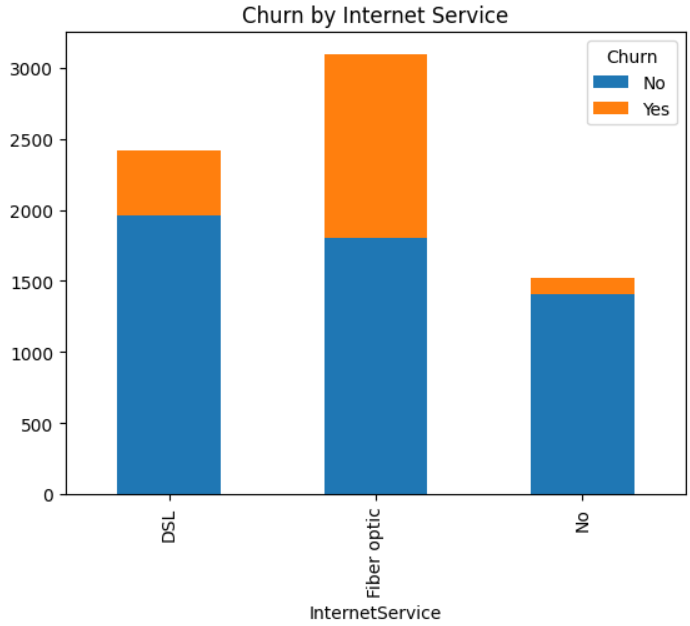
1. **Exploratory Data Analysis(EDA)**

Here is the visualization that provides an analysis on the features that provides an understanding that which leads to customer churn.

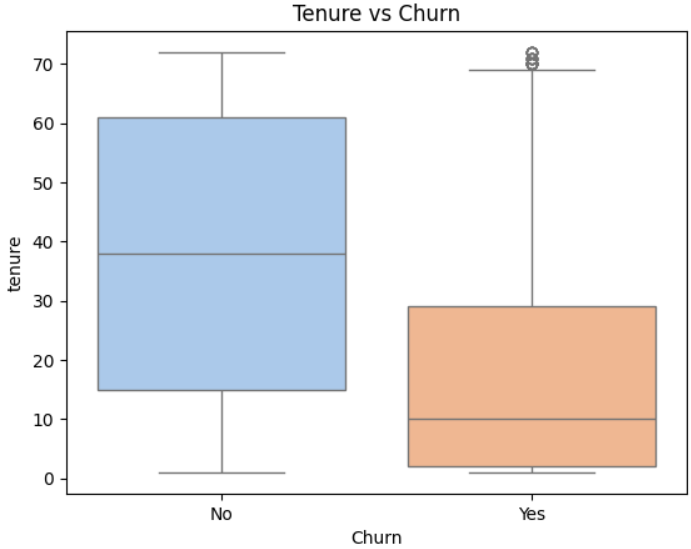
1. **The dataset appears to be imbalanced, with more customers not having churned (No) than churned (Yes).** This is because the height of the bar for "No" is significantly higher than the height of the bar for "Yes".



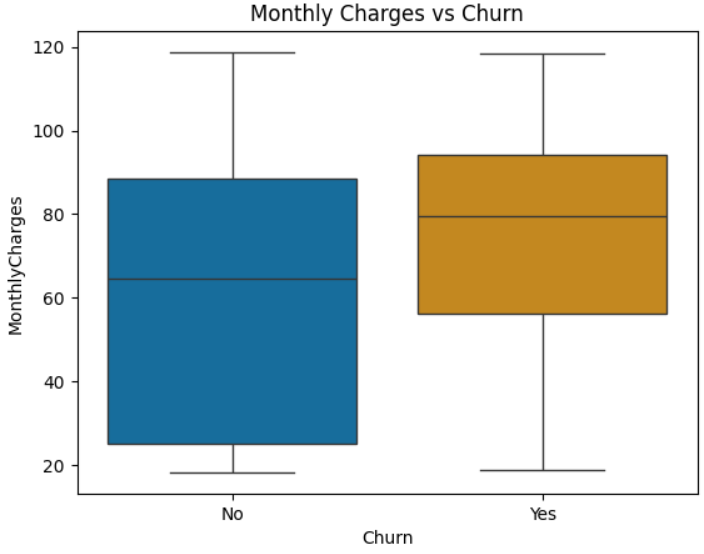
1. Fiber optic internet service has the highest churn rate. This suggests that customers with fiber optic connections are more likely to switch providers compared to those with DSL or no internet service.



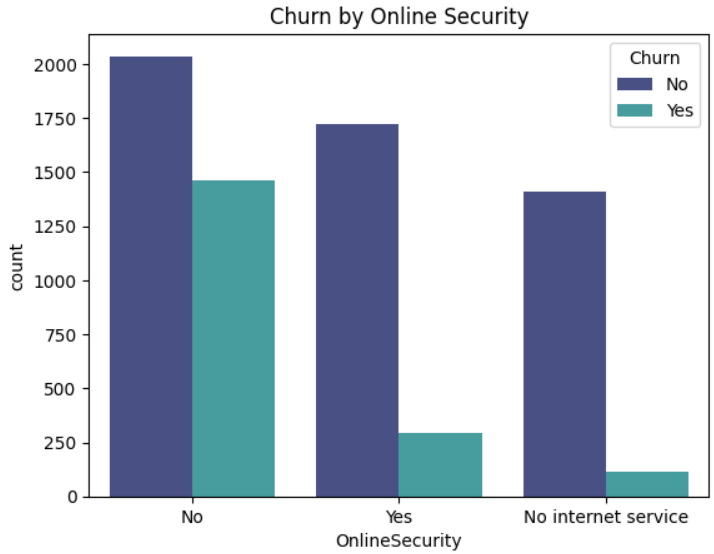
1. **There might be a slight trend towards higher tenures for non-churned customers.** The median (horizontal line within the box) for the "No" churn category is slightly higher than for "Yes".



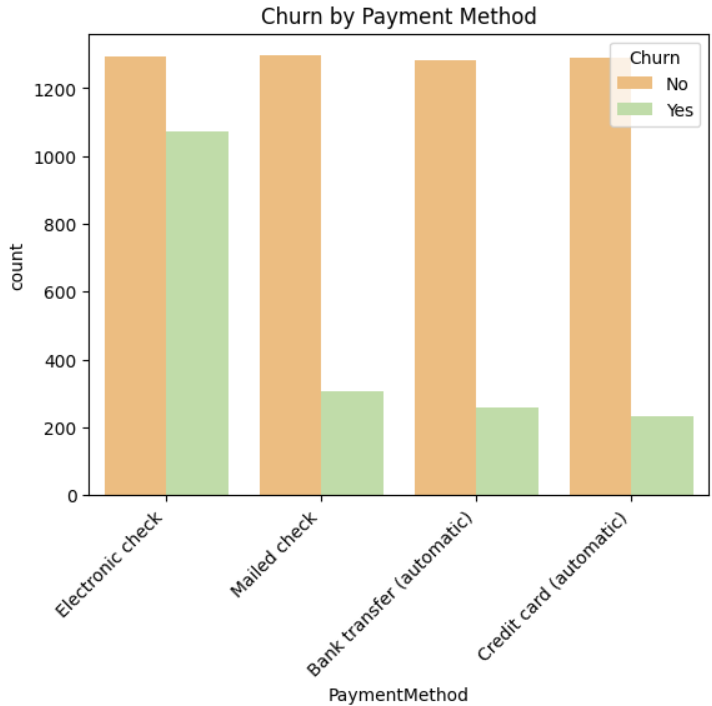
1. **The interquartile range (IQR), represented by the size of the boxes, is larger for churned customers compared to non-churned customers.** This indicates a greater spread of monthly charges among those who churned.



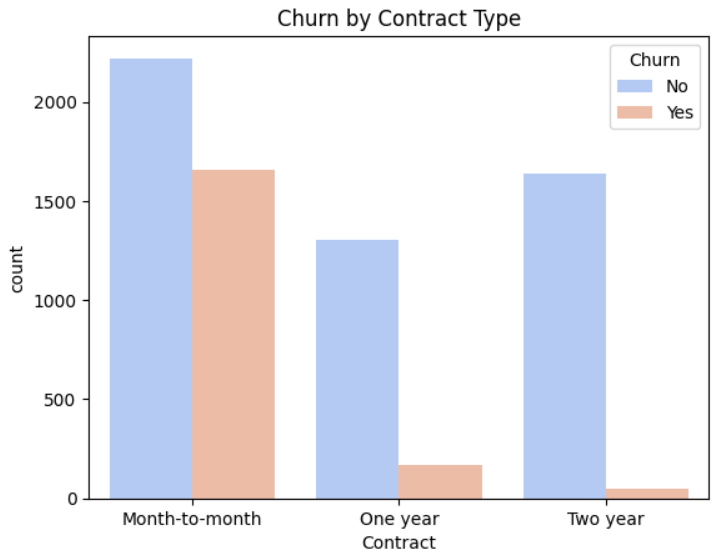
1. **There is a slightly higher proportion of churned customers among those who don't have Online Security.** This is because the count of "Yes" churn seems a bit higher for the "No" category on Online Security compared to "Yes".

****

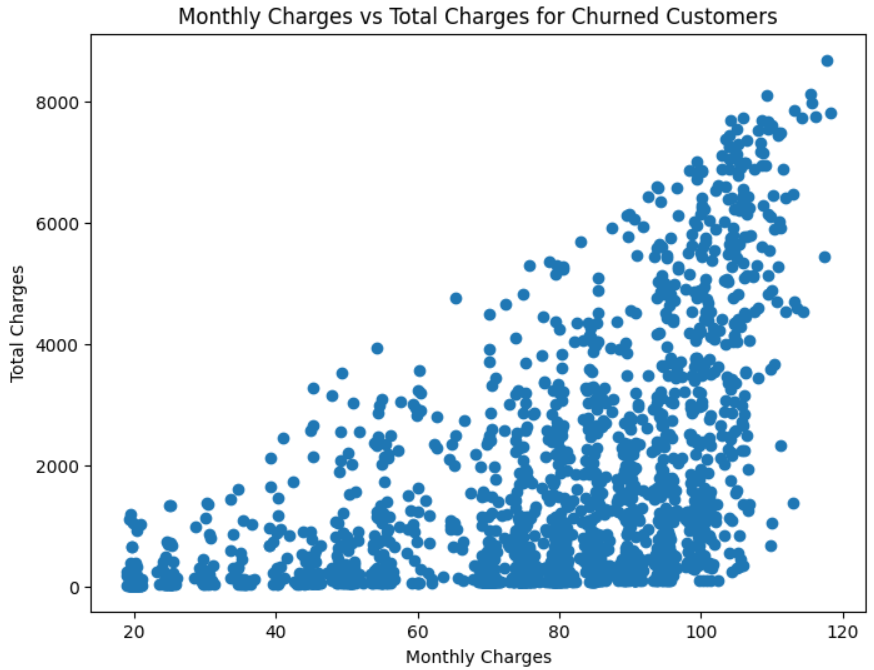
1. **Customers who pay with Credit Card (automatic) or Bank transfer (automatic) churn at a lower rate.**

****

1. **The churn rate appears to be highest for customers with Month-to-month contracts. Two-year contracts seem to have the lowest churn rate.** This suggests that customers who commit to longer contracts churn less, possibly due to price incentives or feeling more locked in.

****

1. **Positive Correlation:** There likely exists a positive correlation between monthly charges and total charges for churned customers. This means that as the monthly charges increase, the total charges also tend to increase.

****

1. Correlation with respct to churn

**Strong Positive Correlations with Churn**:

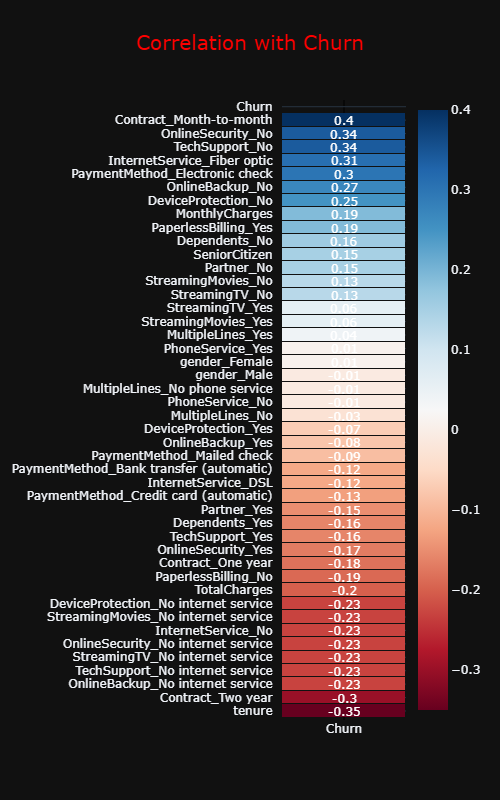
* **Contract Month-to-Month (0.40)**: Customers with month-to-month contracts are more likely to churn compared to those with longer-term contracts.
* **Online Security No (0.34)** and **Tech Support No (0.34)**: Lack of online security and tech support is strongly associated with higher churn, indicating these services are critical in retaining customers.
* **Internet Service Fiber Optic (0.31)**: Customers using fiber optic internet services have a higher churn tendency, possibly due to competition or dissatisfaction with pricing or service.
* **Payment Method Electronic Check (0.30)**: Customers using electronic checks as a payment method are more likely to churn.

**Negative Correlations with Churn**:

* **Tenure (-0.35)**: The strongest negative correlation indicates that longer tenure decreases the likelihood of churn, highlighting the importance of customer loyalty programs.
* **Contract Two-Year (-0.30)**: Customers with long-term contracts are less likely to churn, showcasing the value of offering incentives for long-term commitments.
* **Total Charges (-0.20)**: While somewhat intuitive, higher total charges correlate with lower churn as they often reflect customers who stay longer.

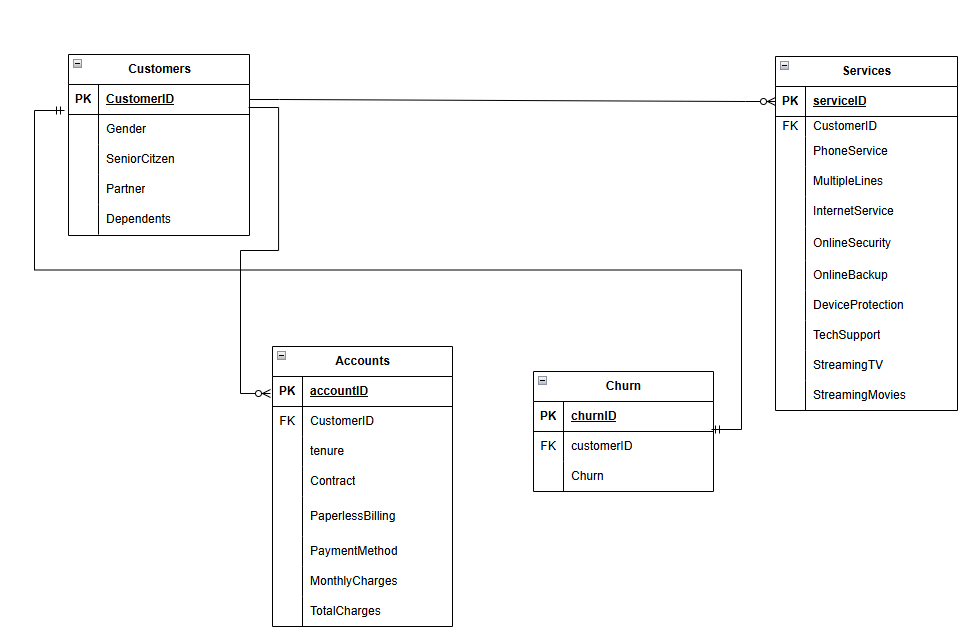
**Actionable Insights**:

* Focus on reducing churn for customers with month-to-month contracts by offering long-term discounts or incentives.
* Improve or bundle critical services like online security and tech support, as their absence is a significant churn driver.
* Address price sensitivity for customers with higher monthly charges by offering customized plans or loyalty rewards.
* Pay special attention to customers using electronic checks as a payment method and investigate underlying reasons for dissatisfaction.



#### **3. Data Management Layer**

* **Preprocessing**:
  1. Handling missing values (e.g., replacing empty strings in TotalCharges with NaN).
  2. Encoding categorical variables using one-hot encoding.
  3. Addressing class imbalance with SMOTEENN.
* **Database**:
  1. MySQL database for efficient storage and retrieval of structured data.
  2. Tables include Customer, Services, Accounts, and Churn, normalized for efficient querying.



#### **4. Modeling Layer**

* **Algorithms Used**:
  + **Random Forest, Decision Tree, Gradient Boosting, AdaBoost, XGBoost**: Used for experimentation, with XGBoost providing the best results due to its handling of non-linearity and imbalanced data.
  + **Artificial Neural Network (ANN)**: Built as a binary classifier with an architecture of 64, 32 nodes in the hidden layers, and a sigmoid-activated output layer. Used for comparative evaluation.
* **Pipeline**: A pipeline built to handle preprocessing, feature transformation, and model prediction seamlessly, deployed using joblib.

#### **5. Prediction and Insights Layer**

* **Churn Prediction**: Predicting the likelihood of customer churn with high accuracy.
* **Feature Importance**: Identifying the most significant factors influencing churn (e.g., contract type, monthly charges, tenure).
* **Generative AI**: A Retrieval-Augmented Generation (RAG) pipeline integrated using Google Gemini Pro APIs to enable SQL-based insights and provide interpretive summaries of the results.

#### **6. Deployment Layer**

* **Flask Web Application**:
  + Interactive interface for users to input customer data and get churn predictions.
  + Displaying SQL query results with insights like churn rates by tenure or payment method.
* **API Integration**: RESTful APIs developed for seamless interaction between the web app and the model/database.

## Algorithm Used

For each algorithm:

* The dataset was preprocessed using techniques like SMOTEENN to address class imbalance before training.
* Metrics like accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC were used for comprehensive evaluation.
* Visualizations such as the ROC curve and confusion matrix were plotted for interpretability.

The diversity of models implemented provided robust insights into their strengths and weaknesses, enabling the selection of the most effective algorithm for deployment. **XGBoost** was ultimately chosen for its superior performance and scalability.

* + 1. **Random Forest Classifier:**

A Random Forest classifier was implemented to leverage the power of ensemble learning for churn prediction.

* **Configuration**:
  1. Number of trees (n\_estimators): 100
  2. Splitting criterion: Gini Index
  3. Maximum tree depth: 6 (to avoid overfitting)
  4. Minimum samples per leaf: 8 (to ensure balanced leaf nodes)
  5. Random state: 100 (to maintain reproducibility)
* **Pipeline**: A Pipeline was used to streamline the workflow, encapsulating the Random Forest model and preprocessing steps.
* **Evaluation**:
  1. Training and testing accuracy scores were calculated to evaluate model performance.
  2. The classification report provided precision, recall, and F1 scores for each class.
  3. A confusion matrix visualized the model’s predictions, highlighting true positives, false positives, true negatives, and false negatives.
  4. The ROC curve and AUC score assessed the model’s ability to distinguish between churners and non-churners.
     1. **Decision Tree Classifier:**

The Decision Tree model was implemented for its simplicity and interpretability:

* **Configuration**:
  + Splitting criterion: Gini Index
  + Maximum tree depth: 6
  + Minimum samples per leaf: 8
  + Random state: 100
* **Evaluation**: The model was evaluated using the same metrics as the Random Forest, and feature importance was analyzed to understand the decision-making process.
  + 1. **Gradient Boost Classifier:**

Gradient Boosting was implemented for its ability to optimize prediction accuracy through iterative boosting:

* **Configuration**:
  1. Number of estimators: 50
  2. Maximum depth of trees: 4
  3. Minimum samples per leaf: 8
  4. Random state: 42
* **Evaluation**: The algorithm’s iterative approach was particularly effective in minimizing error, and it performed well in capturing non-linear relationships.
  + 1. **Ada Boost Classifier:**

AdaBoost was utilized to boost the performance of weak learners:

* **Configuration**:
  1. Number of estimators (n\_estimators): 40
  2. Random state: 42
* **Functionality**:
  1. AdaBoost sequentially trains weak classifiers and assigns higher weights to misclassified instances, improving overall model accuracy.
* **Evaluation**: Accuracy, classification metrics, and the ROC-AUC curve were used to compare its performance against other algorithms.
  + 1. **XGBoost Classifier:**

XGBoost emerged as the best-performing model, known for its speed and accuracy:

* **Functionality**:
  1. Combines gradient boosting with regularization techniques to prevent overfitting.
  2. Handles class imbalance effectively by optimizing the objective function.
* **Evaluation**: XGBoost achieved the highest accuracy and AUC scores, making it the chosen model for deployment.
  + 1. **Artifical Neural Network:**

An Artificial Neural Network was built as a binary classification model to explore deep learning’s potential for churn prediction:

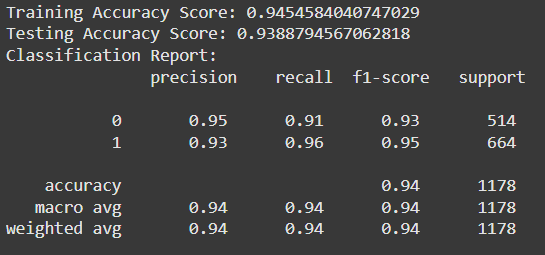
* **Architecture**:
  1. Input layer: 64 nodes
  2. Hidden layers: Two layers with 64 and 32 nodes, respectively, using ReLU activation.
  3. Output layer: One node with a sigmoid activation function to output probabilities.
* **Training**: The model used binary cross-entropy as the loss function and tracked accuracy as the evaluation metric.
* **Evaluation**: Compared with other models, ANN provided competitive results but required more computational resources

# Results and Discussion

## Introduction

This chapter presents a comparative analysis of machine learning models developed for the "Telecom Churn" project. It evaluates each algorithm's performance using ROC Curves and Classification Rpeort. This comprehensive assessment highlights the strengths and weaknesses of each model, guiding the selection of the most effective approach for predicting car prices

## Random Forest Classifier result:

****

#### **1. Training and Testing Accuracy**

* **Training Accuracy**: 94.55%
* **Testing Accuracy**: 93.89%

The high training accuracy indicates that the model effectively learns patterns from the training data. The slightly lower testing accuracy suggests that the model generalizes well to unseen data with minimal overfitting. This balance between training and testing accuracy highlights the model’s robustness and reliability.

#### **2. Classification Report Analysis**

 **Precision**:

* Class 0 (No Churn): 95% of customers predicted as "No Churn" were correctly classified.
* Class 1 (Churn): 93% of customers predicted as "Churn" were correct.
* The high precision for both classes indicates the model’s ability to minimize false positives effectively.

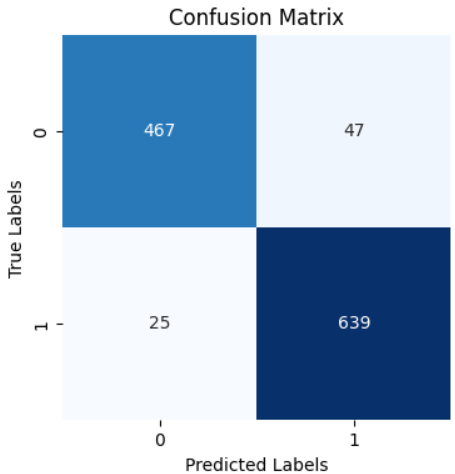
 **Recall**:

* Class 0 (No Churn): The model correctly identified 91% of non-churners.
* Class 1 (Churn): The model captured 96% of actual churners, showcasing its strength in detecting churn, which is critical for this business problem.

 **F1-Score**:

* The F1-scores for both classes are high, with Class 1 (Churn) achieving 0.95. This reflects a good balance between precision and recall, making the model suitable for applications where false negatives (missed churners) can have significant costs.

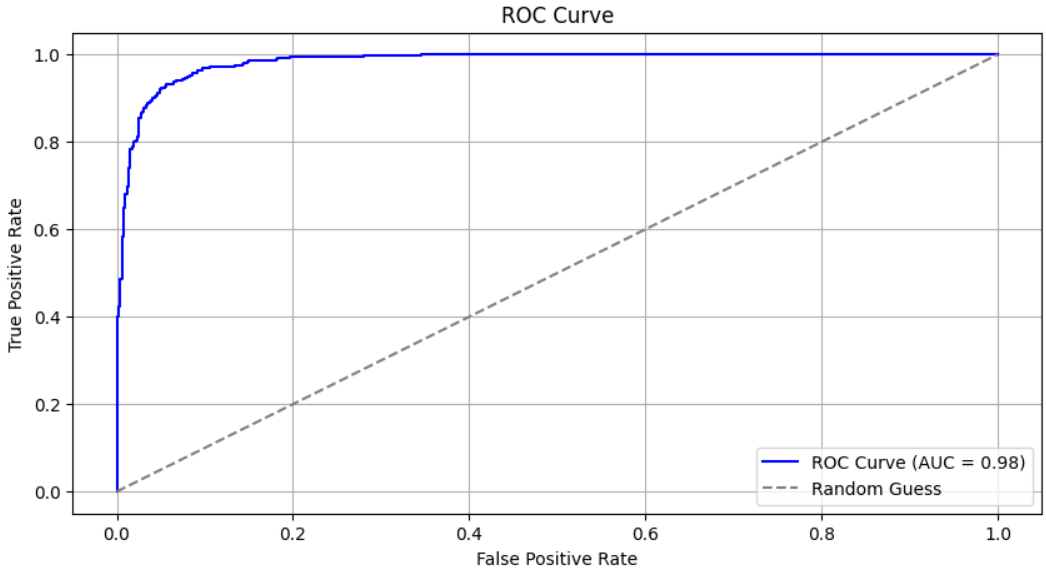
1. Confusion Matrix Analysis



* **True Positives (639)**: The model correctly identified 639 churners.
* **True Negatives (467)**: The model accurately predicted 467 non-churners.
* **False Positives (47)**: 47 non-churners were incorrectly classified as churners.
* **False Negatives (25)**: Only 25 churners were missed, demonstrating strong sensitivity to churn detection.

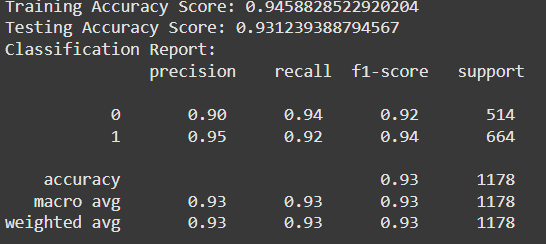
The confusion matrix reflects that the model is highly effective in minimizing false negatives, which is critical for customer retention strategies.

1. **ROC Curve**



* **High AUC Score:** The Area Under the Curve (AUC) score of 0.98 indicates excellent model performance. A higher AUC suggests better classification accuracy. In this case, the model is highly effective in distinguishing between positive and negative classes.
* **Steep Curve:** The steepness of the curve signifies that the model can accurately classify instances with minimal false positives and false negatives.
* **Close to the Top-Left Corner:** The curve's proximity to the top-left corner implies that the model has a high true positive rate (sensitivity) and a low false positive rate (specificity). This is desirable as it indicates the model's ability to correctly identify positive instances while minimizing incorrect classifications.

## Decision Tree Classifier Results:



#### **1. Training and Testing Accuracy**

* **Training Accuracy**: 94.59%
* **Testing Accuracy**: 93.12%

The Decision Tree classifier demonstrates high training accuracy, indicating strong learning capabilities from the dataset. A slightly lower testing accuracy suggests the model generalizes well, though slightly less effectively compared to Random Forest. This performance shows the Decision Tree model is both powerful and interpretable, making it a good fit for churn prediction.

#### **2. Classification Report Analysis**

 **Precision**:

* **Class 0 (No Churn)**: 90% of predictions for non-churners were accurate.
* **Class 1 (Churn)**: 95% of churn predictions were correct, which is crucial for identifying at-risk customers.

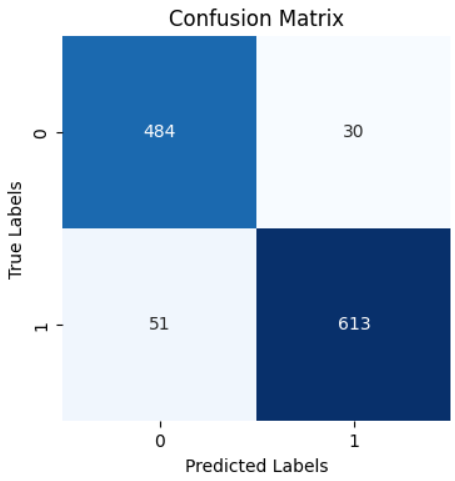
 **Recall**:

* **Class 0 (No Churn)**: 94% of actual non-churners were correctly classified.
* **Class 1 (Churn)**: 92% of churners were identified correctly, slightly lower than Random Forest but still effective for customer retention.

 **F1-Score**:

* The F1-scores for both classes reflect a good balance between precision and recall. The churn class (Class 1) achieved an F1-score of 0.94, showcasing the model’s reliability in churn prediction.

3. Confusion Matrix Analysis



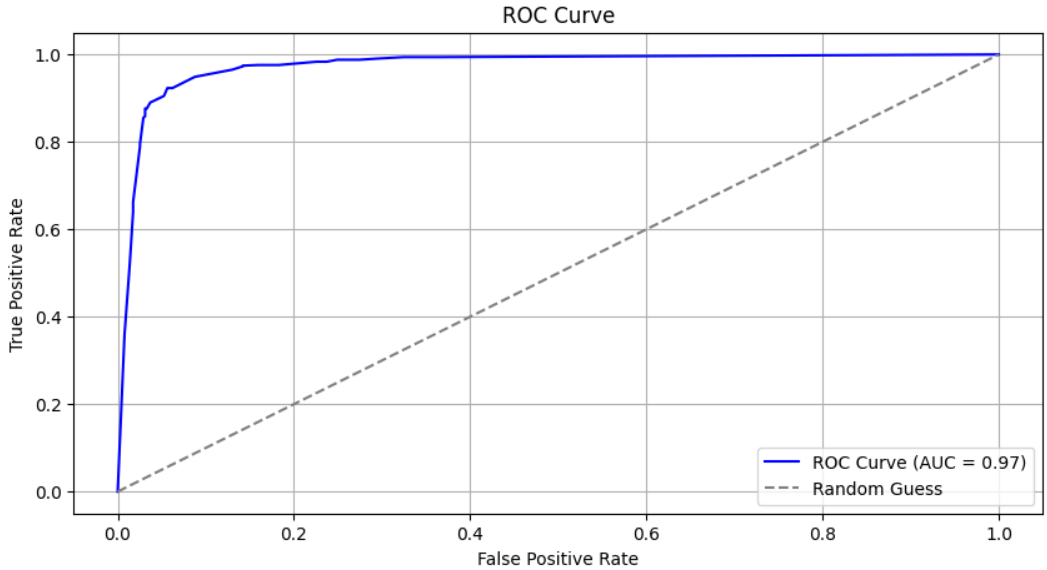
 **True Positives (613)**: The model correctly identified 613 churners.

 **True Negatives (484)**: Non-churners were correctly predicted in 484 cases.

 **False Positives (30)**: Only 30 non-churners were mistakenly classified as churners.

 **False Negatives (51)**: 51 churners were misclassified as non-churners.

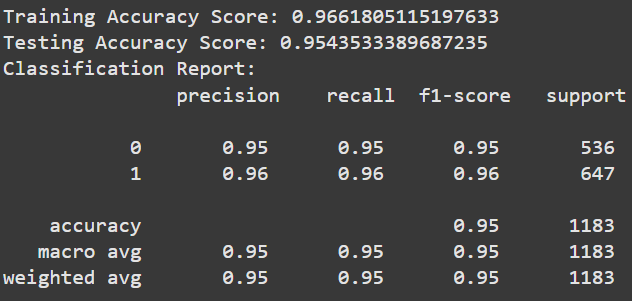
1. **ROC Curve**

****

 **High AUC Score:** The AUC score of 0.97 indicates excellent model performance. This means the model effectively distinguishes between positive and negative classes.

 **Steep Curve:** The curve's steepness signifies that the model can accurately classify instances with minimal false positives and false negatives.

## Gradient Boost Results:



#### **1. Training and Testing Accuracy**

* **Training Accuracy**: 96.61%
* **Testing Accuracy**: 95.43%

The model demonstrates high accuracy on both training and testing datasets, with minimal variance between them. This indicates strong generalization capabilities and effective learning from the training data.

#### **2. Classification Report Analysis**

 **Precision**:

* **Class 0 (No Churn)**: 95% of non-churn predictions were correct.
* **Class 1 (Churn)**: 96% of churn predictions were accurate, showcasing excellent precision.

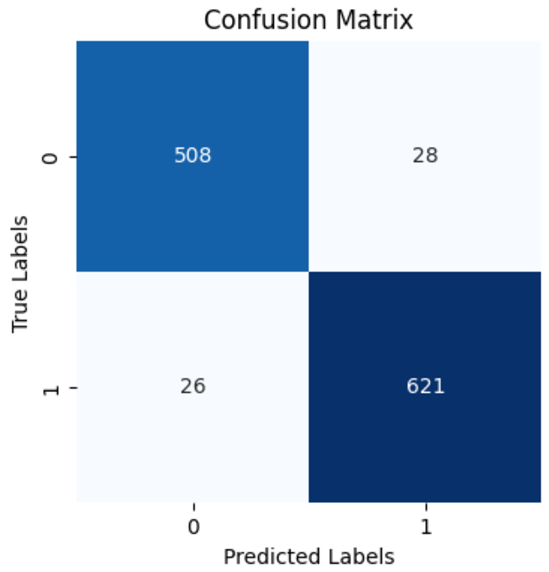
 **Recall**:

* **Class 0 (No Churn)**: 95% of actual non-churners were identified.
* **Class 1 (Churn)**: The model correctly identified 96% of churners, making it highly effective in detecting at-risk customers.

 **F1-Score**:

* F1-scores for both classes are high, indicating a perfect balance between precision and recall.

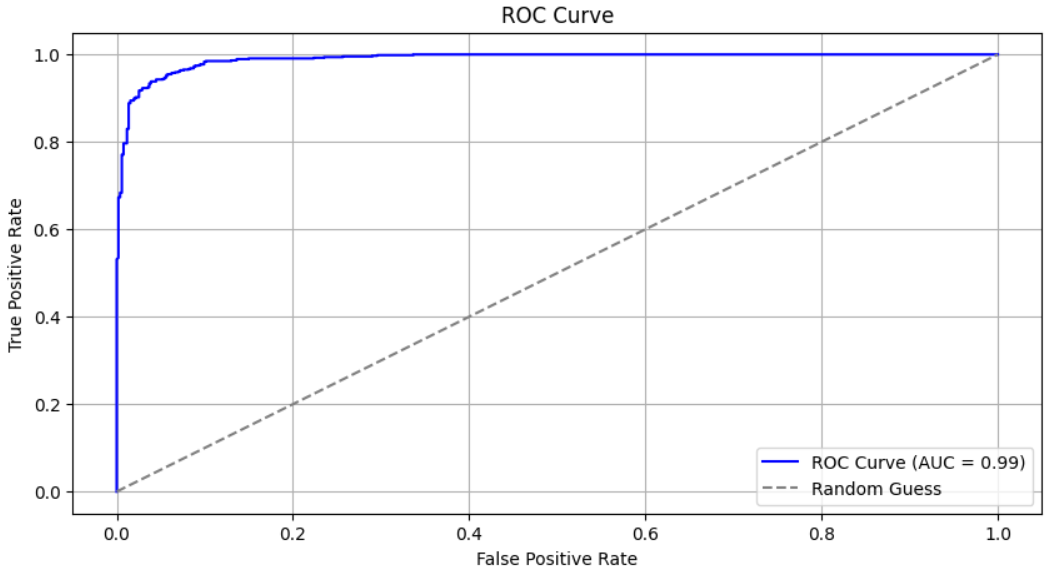
3. Confusion Matrix Analysis



* **True Positives**: Correctly identified 621 churners.
* **True Negatives**: Accurately predicted 508 non-churners.
* **False Positives**: Misclassified 28 non-churners as churners.
* **False Negatives**: Missed 26 churners, a low rate that demonstrates the model's sensitivity.

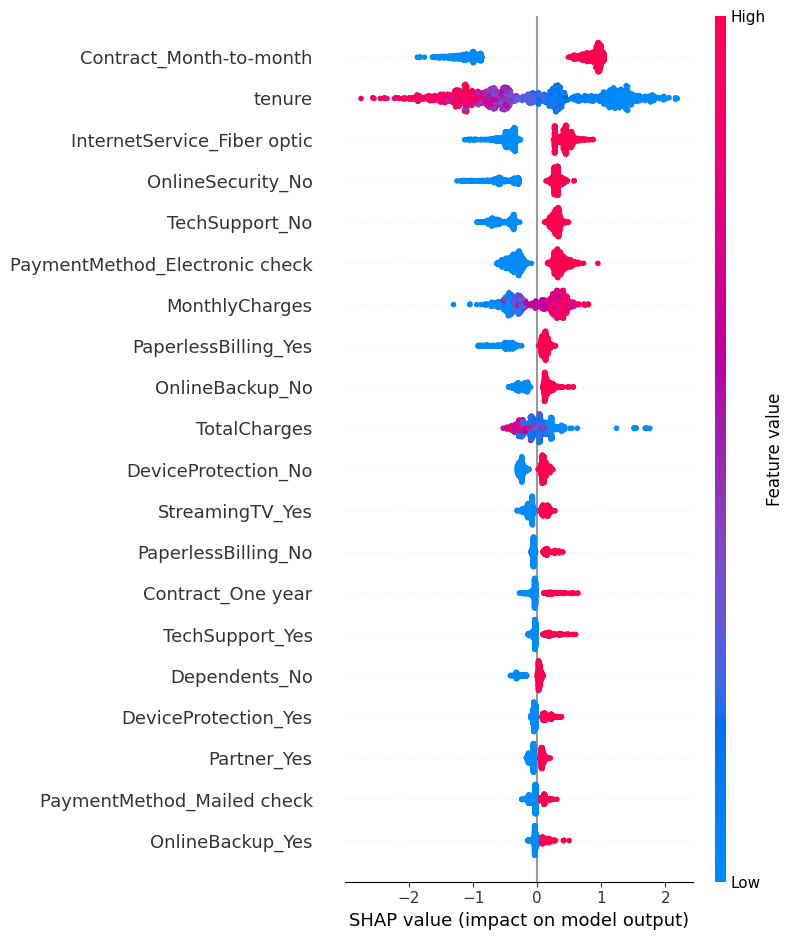
The confusion matrix highlights balanced performance across both classes, with low misclassification rates.

1. **ROC Curve**



* **High AUC Score:** The AUC score of 0.99 indicates an exceptionally strong model performance. This means the model is highly effective in distinguishing between positive and negative classes.
* **Steep Curve:** The steepness of the curve signifies that the model can accurately classify instances with minimal false positives and false negatives.

1. **Shapely Values**



The SHAP summary plot provides a visual representation of the feature importance and direction of their impact on the model's output. Here are some key insights from the plot:

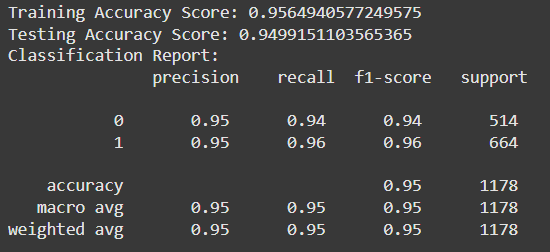
**Important Features:**

* **Contract\_Month-to-month:** Customers with month-to-month contracts have a higher likelihood of churning, as indicated by the positive SHAP values.
* **tenure:** Customers with lower tenure are more likely to churn, as shown by the negative SHAP values.
* **InternetService\_Fiber optic:** Customers with fiber optic internet service have a higher tendency to churn, as indicated by the positive SHAP values.

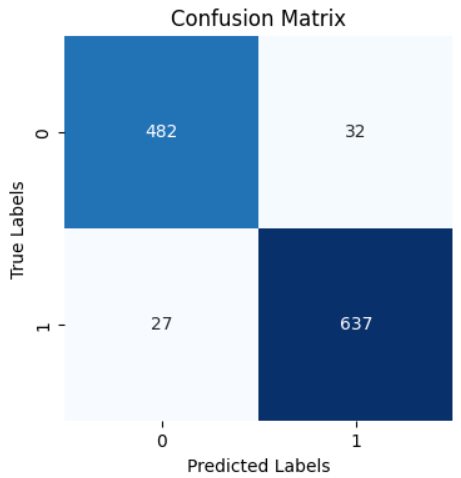
**Other Notable Features:**

* **OnlineSecurity\_No, TechSupport\_No, PaymentMethod\_Electronic check, PaperlessBilling\_Yes, OnlineBackup\_No, DeviceProtection\_No:** These features also contribute positively to the churn prediction, suggesting that customers without these services or with specific payment methods are more likely to churn.
* **Contract\_One\_year,TechSupport\_Yes,Dependents\_No,DeviceProtection\_Yes, Partner\_Yes, PaymentMethod\_Mailed check, OnlineBackup\_Yes:** These features have negative SHAP values, indicating that customers with these characteristics are less likely to churn.

## Ada Boost Classifier Results:

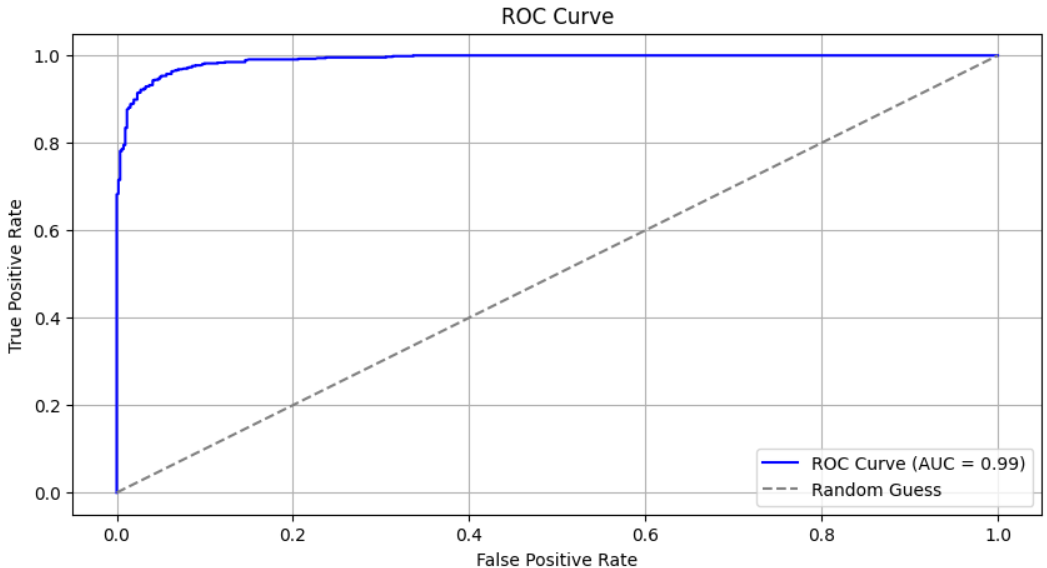


1. **Accuracy**:
   * **Training Accuracy**: 95.65%
   * **Testing Accuracy**: 94.99%
   * The model generalizes well from training to testing data, showing no significant overfitting.
2. **Classification Report**:
   * **Precision**: High precision (0.95 for both classes) indicates the model effectively avoids false positives.
   * **Recall**: High recall (0.94 for class 0 and 0.96 for class 1) shows the model captures most true positives.
   * **F1-Score**: Balanced f1-scores (0.94 and 0.96) indicate a strong balance between precision and recall.
3. **Confusion Matrix**:



* + Class 0: Out of 514 actual negatives, 482 are correctly predicted (TN), and 32 are misclassified (FP).
  + Class 1: Out of 664 actual positives, 637 are correctly predicted (TP), and 27 are misclassified (FN).
  + The matrix suggests slightly more misclassifications in class 0 but still excellent overall performance.

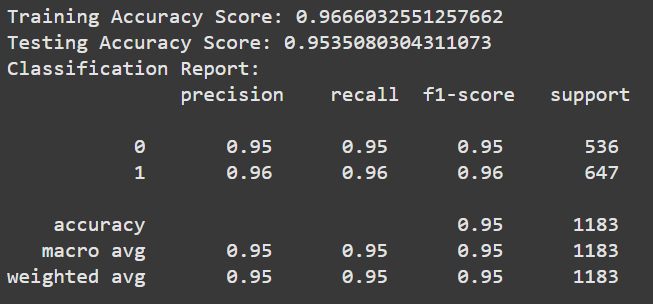
1. **ROC Curve** :



 **High AUC Score:** A score of 0.99 indicates an exceptionally strong model performance. This means the model effectively distinguishes between positive and negative classes.

 **Steep Curve:** The curve's steepness signifies that the model can accurately classify instances with minimal false positives and false negatives.

## XGBoost Classifier Results:



#### **1. Training and Testing Accuracy**

* **Training Accuracy**: 96.66%
* **Testing Accuracy**: 95.35%

The model demonstrates high accuracy on both training and testing datasets, with minimal variance between them. This indicates strong generalization capabilities and effective learning from the training data.

#### **2. Classification Report Analysis**

 **Precision**:

* **Class 0 (No Churn)**: 95% of non-churn predictions were correct.
* **Class 1 (Churn)**: 96% of churn predictions were accurate, showcasing excellent precision.

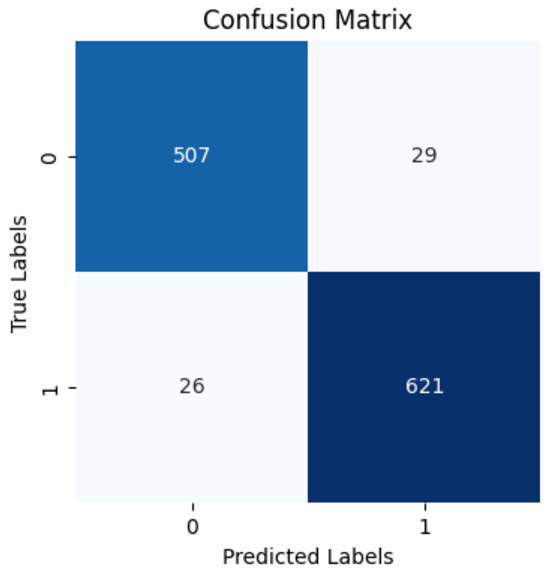
 **Recall**:

* **Class 0 (No Churn)**: 95% of actual non-churners were identified.
* **Class 1 (Churn)**: The model correctly identified 96% of churners, making it highly effective in detecting at-risk customers.

 **F1-Score**:

* F1-scores for both classes are high, indicating a perfect balance between precision and recall.

1. **Confusion Matrix**:

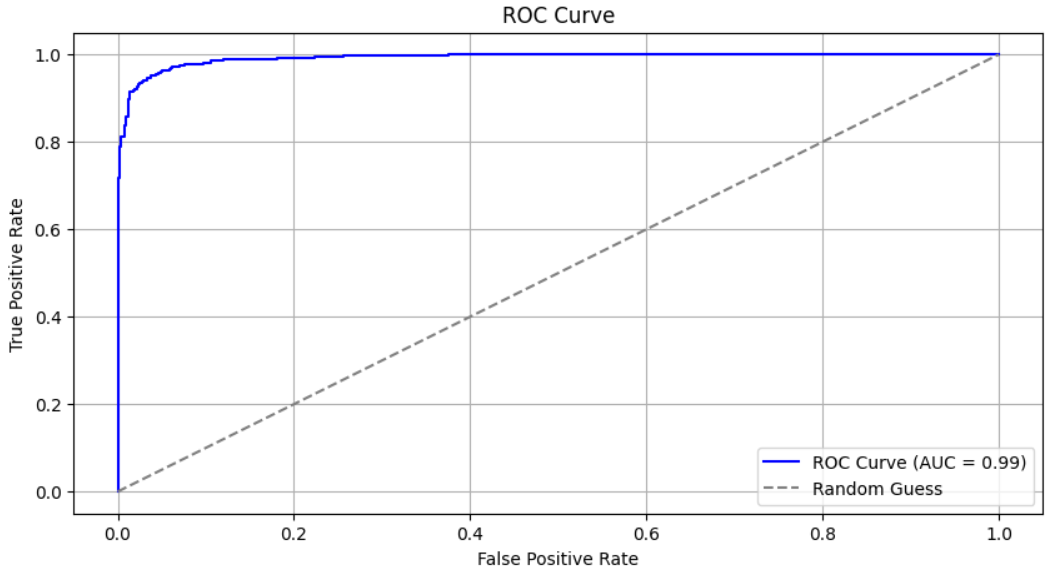


* **True Positives**: Correctly identified 621 churners.
* **True Negatives:** Accurately predicted 507 non-churners.
* **False Positives**: Misclassified 29 non-churners as churners.
* **False Negatives**: Missed 26 churners, a low rate that demonstrates the model's sensitivity.

The confusion matrix highlights balanced performance across both classes, with low misclassification rates.

1. **ROC Curve** :

*  **High AUC Score:** The AUC score of 0.99 indicates an exceptionally strong model performance. This means the model is highly effective in distinguishing between positive and negative classes.
*  **Steep Curve:** The steepness of the curve signifies that the model can accurately classify instances with minimal false positives and false negatives.



1. Shapely Values

**Interpreting the SHAP Summary Plot**

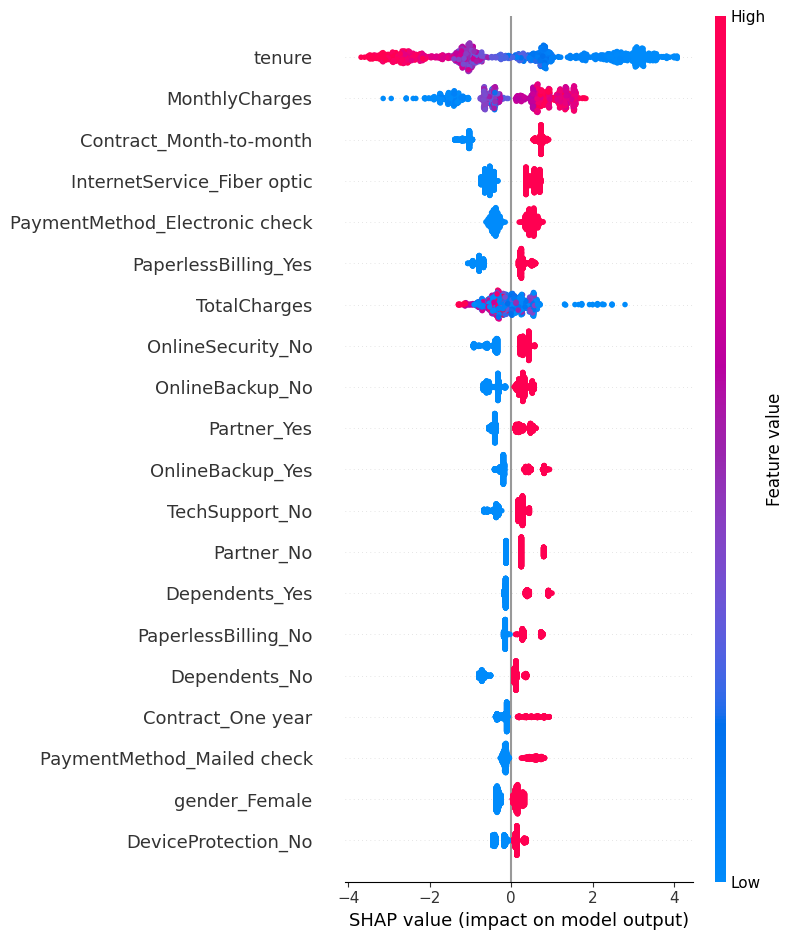
The SHAP summary plot provides a visual representation of the feature importance and direction of their impact on the model's output. Here are some key insights from the plot:

**Important Features:**

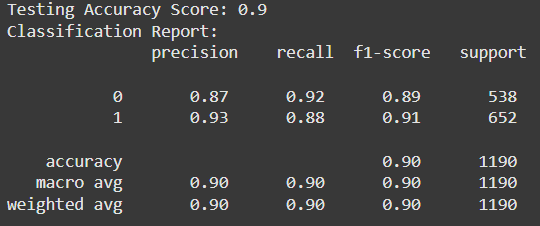
* **tenure:** Customers with lower tenure are more likely to churn, as shown by the negative SHAP values.
* **MonthlyCharges:** Higher monthly charges are associated with a higher likelihood of churn, as indicated by the positive SHAP values.
* **Contract\_Month-to-month:** Customers with month-to-month contracts have a higher likelihood of churning, as indicated by the positive SHAP values.
* **InternetService\_Fiber optic:** Customers with fiber optic internet service have a higher tendency to churn, as indicated by the positive SHAP values.

**Other Notable Features:**

* **PaymentMethod\_Electroniccheck,PaperlessBilling\_Yes,OnlineSecurity\_No, OnlineBackup\_No:** These features also contribute positively to the churn prediction, suggesting that customers without these services or with specific payment methods are more likely to churn.
* **Partner\_Yes, TechSupport\_Yes, Dependents\_Yes:** These features have negative SHAP values, indicating that customers with these characteristics are less likely to churn.



## Artifical Neural Network Results:



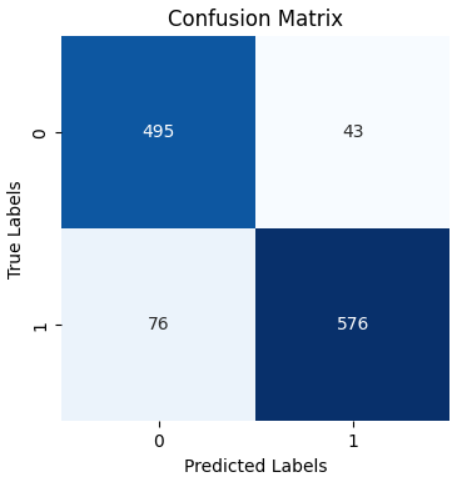
### 1. ****Testing Accuracy****:

* **Accuracy Score**: 0.90, meaning that 90% of the predictions made by the model on the test set were correct.

### 2. ****Classification Report****:

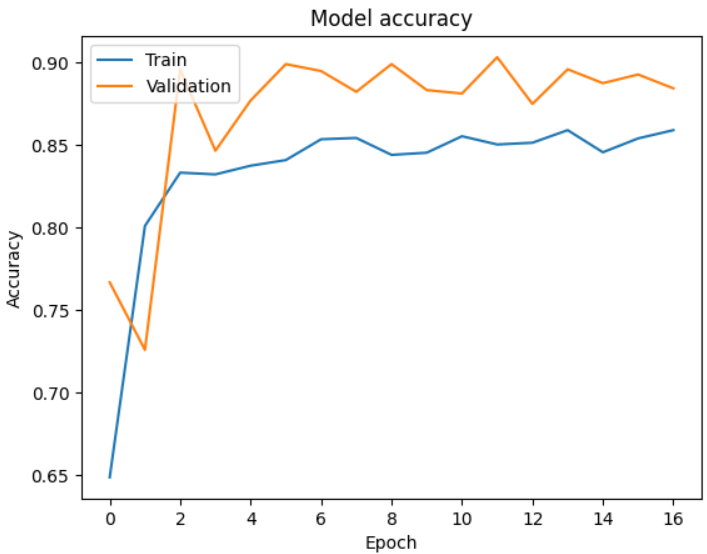
* **Precision**:
  + **Class 0 (Churn = 0)**: 0.87, meaning 87% of the instances predicted as not churned were correct.
  + **Class 1 (Churn = 1)**: 0.93, meaning 93% of the instances predicted as churned were correct.
* **Recall**:
  + **Class 0 (Churn = 0)**: 0.92, meaning 92% of the actual non-churned instances were correctly predicted.
  + **Class 1 (Churn = 1)**: 0.88, meaning 88% of the actual churned instances were correctly predicted.
* **F1-Score**:
  + **Class 0**: 0.89 (harmonic mean of precision and recall for non-churned customers).
  + **Class 1**: 0.91 (harmonic mean of precision and recall for churned customers).
  + The F1-scores are well-balanced across both classes, showing the model's good performance in distinguishing between churn and non-churn customers.
* **Macro Average**: The average precision, recall, and F1-score across both classes, which are all 0.90, indicating balanced performance across the classes.
* **Weighted Average**: Also 0.90, which accounts for the imbalance in the class distribution (since Class 1 has more samples than Class 0) and gives a weighted average of precision, recall, and F1-score.

### 3. ****Confusion Matrix****:



* **True Negatives (TN)**: 495 (The number of instances that were correctly predicted as non-churned).
* **False Positives (FP)**: 43 (The number of non-churned instances incorrectly predicted as churned).
* **False Negatives (FN)**: 76 (The number of churned instances incorrectly predicted as non-churned).
* **True Positives (TP)**: 576 (The number of churned instances correctly predicted as churned).
* The confusion matrix further confirms that the model is performing well, with fewer false positives (43) and false negatives (76), which is indicative of its ability to identify churn and non-churn customers with reasonable accuracy.

1. **Validation Accuracy:**

****

The validation accuracy also increases initially and then plateaus, indicating that the model is generalizing well to unseen data.

**Final Result:**

Gradient Boosting and XGBoost, both models performed exceptionally well, achieving testing accuracies of 95.43% and 95.35%, respectively, with an identical ROC AUC of 0.99. However, **Gradient Boosting** demonstrated slightly better performance in terms of false positive reduction (28 compared to 29), making it the preferred choice for deployment.

# Implementation

## Introduction

The implementation of telecom churn prediction leverages a Flask web application framework to serve two main functionalities: predicting whether the customer churns or not based on its attributes and an LLM model that gives an output to a question asked in natural language to process a result from a MySQL database.

## Flask App

#### 1. **Setup and Initialization**

* **Flask Framework**: Flask was chosen as the framework for this project due to its simplicity and flexibility in building web applications. It provides a lightweight way to create REST APIs and manage web routes.
* **Environment Setup**: The environment variables, such as MySQL database credentials and API keys, were stored in a **.env** file to ensure security and ease of configuration. The **dotenv** package was used to load these environment variables into the Flask app.
* **CORS Handling**: The CORS (Cross-Origin Resource Sharing) was enabled using the **flask\_cors** package to ensure that the Flask app could be accessed by different clients (like a front-end application or an API client).

#### **2.** **Integrating Machine Learning Model for Churn Prediction**

The Flask app is responsible for serving the churn prediction model, which was trained using the Telecom Customer Churn dataset. The machine learning model pipeline is loaded using the **joblib** package to make predictions on customer data.

* **Model Loading**: The model was saved as a .pkl file during the training phase and is loaded into the Flask app at runtime.
* **Prediction Endpoint**: The /predict endpoint accepts POST requests containing customer data, validates the data, preprocesses it (e.g., one-hot encoding), and feeds it into the machine learning pipeline to make churn predictions.
* **Prediction Logic**: The pipeline, which includes preprocessing steps (such as encoding categorical variables) and the model itself (an Gradient Boosting classifier), processes the input and returns a prediction.

#### **3**. **Natural Language Query to SQL Conversion (RAG Pipeline)**

To interact with the database via natural language questions, a **Retrieval-Augmented Generation (RAG)** pipeline was implemented. The app uses **Google Gemini Pro**, an advanced generative AI model, to convert user questions into SQL queries, which are then executed on the MySQL database.

* **Google Gemini Pro**: This model was configured using an API key, and its purpose is to generate SQL queries based on natural language inputs.
* **Database Interaction**: Once the SQL query is generated, the Flask app connects to the MySQL database using pymysql.connect, executes the query, and returns the result to the user.

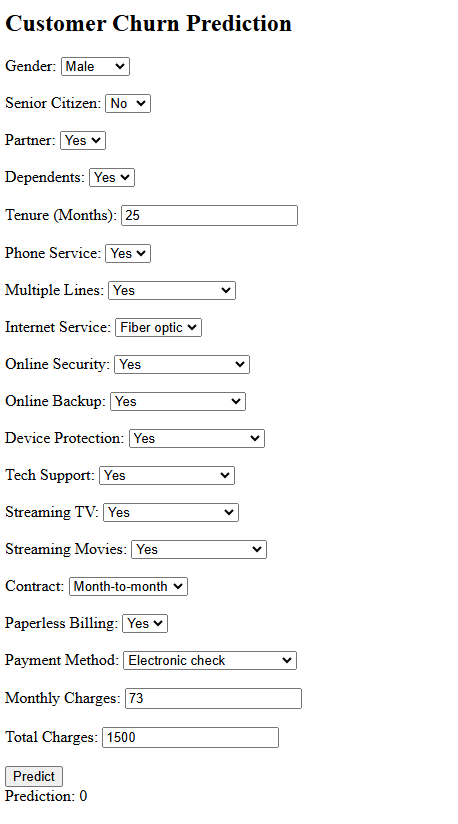
#### **4. Database Interaction**

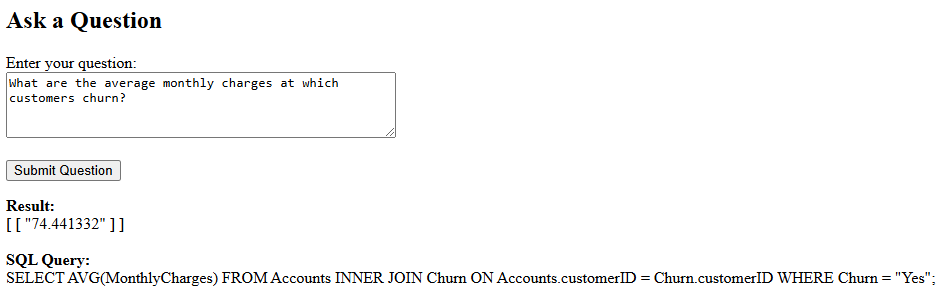
The Flask app connects to a **MySQL database** to retrieve and manipulate customer data, which is stored in tables like Customers, Accounts, Services, and Churn. A SQL query is generated dynamically based on the user’s question and executed using mysql.connector.

* **SQL Query Execution**: The read\_sql\_query function connects to the database, executes the query, and returns the results. If any errors occur during the database operation, the app handles the exception and returns an error message.
  1. **User Interface**

The user interface (UI) for the Telecom Customer Churn Prediction application is designed to allow users to interact with the churn prediction model and query the database using natural language.

1. **Churn Prediction Form:**
   * The main section features a form where users can input various customer details such as gender, senior citizen status, contract type, internet service, and charges.
   * Upon form submission, the system processes the data and provides a prediction on whether the customer is likely to churn.
2. **Question Submission:**
   * A second section allows users to submit a question regarding the telecom database.
   * A textarea input field lets users type natural language questions, and a button triggers the backend to generate an SQL query and retrieve results.
3. **Result Display:**
   * After submitting the churn prediction form, the predicted outcome is displayed on the page.
   * Similarly, when the user asks a question, the SQL query and its corresponding result are displayed below the question form.

****

****

# Conclusion

Working on this project has been immensely beneficial for me. It allowed me to explore and implement concepts that were new to me, such as SHAP for explainable AI, creating a Retrieval-Augmented Generation (RAG) pipeline, and building a custom LLM. These experiences not only expanded my technical knowledge but also enhanced my problem-solving abilities. Ultimately, this project has been a significant milestone in polishing my skills and boosting my confidence in tackling complex, real-world challenges.