PROJECT REPORT – CUSTOMER CHURN PREDICTION

Project Overview

The primary goal of this project is to predict customer churn for a telecommunications company by leveraging machine learning techniques, specifically a Logistic Regression model. Customer churn refers to the phenomenon where customers discontinue their association with a service provider. For businesses in highly competitive industries like telecommunications, understanding the reasons behind churn and predicting which customers are likely to leave can significantly improve retention strategies, customer satisfaction, and revenue generation.

The dataset for this project comprises 5,880 customer records, each represented by 21 distinct features. These features capture vital information about the customers, such as their demographic details, account status, service usage patterns, and payment behaviors. The target variable, 'Churn,' is a binary classification that indicates whether a customer has churned (Yes) or remained with the company (No). The features in the dataset span both categorical and numerical types. Categorical variables include attributes like gender, partner status, and internet service type, while numerical variables include tenure (number of months a customer has stayed with the company), monthly charges, and total charges.

The primary challenge of this project lies in preparing the raw data for machine learning. The dataset initially contains a mix of missing values, imbalanced classes, and categorical data that must be encoded into numerical representations for the model to process. To address these challenges, various data preprocessing techniques were applied, such as handling missing values, encoding categorical variables, and scaling numerical features to ensure uniformity. Additionally, the dataset was divided into training and testing subsets to evaluate the performance of the model on unseen data.

The Logistic Regression model was chosen for its simplicity and interpretability, particularly in identifying the relative importance of features contributing to churn. After preprocessing the data and training the model, its performance was assessed using various metrics, including accuracy, precision, recall, F1-score, and the confusion matrix. The evaluation revealed an accuracy of 49%, which highlights the complexity of the churn prediction problem and indicates room for improvement through advanced modeling techniques or feature engineering.

Ultimately, the insights derived from this project aim to provide actionable recommendations for the telecommunications company. By identifying the key factors driving churn, the business can take targeted steps to improve customer satisfaction, enhance service quality, and design retention strategies tailored to high-risk customer groups. This project not only emphasizes the predictive capability of machine learning but also underlines the importance of understanding customer behavior and translating data into meaningful business decisions.

Data Exploration

Data exploration is an essential step in understanding the characteristics of the dataset and identifying potential relationships between features. In this project, the exploration of the dataset was conducted in a structured manner, allowing for a comprehensive analysis of the customer data. Below is a detailed exploration, divided into key areas of focus.

1. Overview of the Dataset

The dataset contains a total of 5,880 customer records, each characterized by 21 features. These features include both categorical and numerical variables, capturing various aspects of the customers' demographics, service subscriptions, and usage patterns. The target variable, Churn, indicates whether a customer has left the service (Yes) or is still active (No). Here's a brief overview of the columns:

- **Customer ID**: Unique identifier for each customer.
- **Demographic Features**: Gender, Senior Citizen, Partner, Dependents.
- **Service Features**: Tenure, Phone Service, Multiple Lines, Internet Service, Online Security, Device Protection, Tech Support, Streaming TV, Streaming Movies, etc.
- **Billing Features**: Monthly Charges, Total Charges.
- **Target Variable**: Churn (indicating whether the customer has left the service or not).

2. Categorical Features

Categorical features help describe customer characteristics and service-related behaviors. In this dataset, many features were categorical, such as gender, Partner, Dependents, PhoneService, MultipleLines, and InternetService. A detailed analysis of these features provided some key insights:

- **Gender**: The dataset had a fairly balanced gender distribution, with approximately equal numbers of male and female customers.
- **SeniorCitizen**: This feature had a binary encoding (0 for non-senior citizens and 1 for senior citizens). It was found that about half of the customers were seniors, reflecting the 0.5 mean value for this feature.
- **Partner and Dependents**: A significant number of customers did not have partners or dependents, which could suggest that many customers are either living alone or do not have familial obligations.
- **PhoneService and MultipleLines**: Most customers who had phone service also had multiple lines, highlighting the preference for bundled services.
- **InternetService**: A high proportion of customers had fiber-optic internet service, with DSL being the next most popular option.

- **Service Preferences (OnlineSecurity, TechSupport, StreamingTV, etc.)**: These features displayed a varied customer preference. For instance, many customers did not subscribe to additional services like online security or tech support, while others had streaming services like TV and movies.

3. Numerical Features

Numerical features like tenure, MonthlyCharges, and TotalCharges provide valuable insights into customer behavior, payment patterns, and loyalty. Here's a closer look at these features:

- **Tenure**: The tenure of customers ranged from 1 to 72 months, with an average tenure of around 36.5 months. This indicates a mix of both new and long-term customers, which is important for understanding customer retention.
- **MonthlyCharges**: This feature ranged from 20.00 to 119.99, with an average value of 70.16. A significant portion of customers paid lower monthly charges, while others subscribed to premium services.
- **TotalCharges**: Ranging from 20.03 to 8589.60, this feature showed a wide disparity, primarily influenced by the tenure feature. Customers who had been with the service for longer periods generally had higher total charges due to accumulated monthly payments.

4. Churn Distribution

One of the most critical aspects of the dataset is the target variable Churn, which indicates whether a customer has canceled their subscription. The dataset was found to be **slightly imbalanced**, with a marginally higher number of customers who had not churned (label No) compared to those who had churned (label Yes). This imbalance is important because it can affect the model's performance and might require techniques like resampling, class weighting, or using specific evaluation metrics to avoid bias towards the majority class.

5. Missing Values and Data Quality

Before proceeding with any machine learning tasks, it was important to ensure that the dataset was clean and free of any issues such as missing values or incorrect data types. A comprehensive check revealed that:

- **No Missing Values**: The dataset had no missing values in any of the columns, making it easier to proceed with preprocessing and model building without the need for imputation or removal of rows.

6. Data Transformation

After ensuring that the data was free of missing values, some data transformation was performed to prepare it for the model:

- **Categorical to Numeric**: Several categorical features like gender, Partner, Dependents, and InternetService were transformed into numeric codes for compatibility with

machine learning algorithms. For example, gender was encoded as 0 for male and 1 for female.

- **Scaling**: Features such as MonthlyCharges and TotalCharges were already in numerical formats, but scaling was performed on these continuous features to ensure that they contributed equally to the model.

7. Key Observations

- **High correlation between TotalCharges and tenure**: A significant relationship was observed between these two features. Customers with longer tenures generally had higher total charges due to their long-term subscriptions and accumulated monthly payments.
- **Churned customers**: When analyzing churned customers, certain service preferences, such as having no online security or tech support, seemed to correlate with higher churn rates. These features may be valuable predictors of customer churn.

In conclusion, the data exploration process provided valuable insights into the characteristics of the customers, their preferences, and factors influencing churn. The next step involves transforming the data and applying machine learning techniques to predict churn. By understanding the relationships in the data, we can improve model performance and gain actionable insights for business decisions.

Data Preprocessing and Feature Engineering

Data Preprocessing and Feature Engineering

Data preprocessing is an integral part of any machine learning project, as it ensures that the raw data is in the correct format for analysis and model building. In this project, the preprocessing steps were aimed at preparing the dataset for training the predictive model, while feature engineering involved enhancing the features to better capture patterns in customer behavior.

1. Handling Missing Data

One of the first steps in preprocessing was to check for any missing values in the dataset. After performing a thorough inspection, it was confirmed that there were **no missing values** across any of the features. This allowed for a smooth continuation of the data preparation process, as there was no need for imputation or row deletion.

2. Converting Categorical Variables

Many features in the dataset were categorical, including gender, Partner, Dependents, InternetService, and others. These categorical features were initially in string format, making them incompatible with machine learning algorithms, which require numerical data for model building.

- **Label Encoding**: To convert categorical variables into numerical values, we used **label encoding**, where each category in a feature was mapped to an integer. For example:
    - gender: Female -> 0, Male -> 1
    - Partner: No -> 0, Yes -> 1
    - Dependents: No -> 0, Yes -> 1
    - Churn: No -> 0, Yes -> 1

This transformation allowed the machine learning models to process the data without issues related to categorical encoding.

3. Feature Scaling

For features like MonthlyCharges and TotalCharges, which were continuous variables with varying scales, **feature scaling** was applied. Scaling is important in machine learning because it ensures that features with larger numerical ranges do not disproportionately affect the performance of the model.

- **Standardization**: A **standard scaler** was applied to the continuous features to transform them into a standard normal distribution, ensuring that they all had a mean of 0 and a standard deviation of 1. This step helped in speeding up the convergence during model training and made the model more efficient.

4. Feature Engineering

Feature engineering is the process of creating new features or modifying existing ones to improve the performance of machine learning models. While no new features were created in this project, certain important aspects were focused on:

- **Tenure and Total Charges**: There was a strong correlation between the tenure feature and TotalCharges. As customers with longer tenures tended to have higher total charges, this relationship was considered when analyzing patterns in customer retention. The tenure feature was used directly, and no additional transformations were made to this feature, as it was already providing useful information.
- **Interaction Between Features**: Although interaction terms (combinations of multiple features) were not explicitly created in this case, it would be an interesting area for future exploration. For instance, combining features like PhoneService and MultipleLines could capture patterns specific to customers who subscribe to these services together.

5. Data Splitting

Once the preprocessing was completed, the next step was to **split** the dataset into training and testing sets. A common practice in machine learning is to reserve a portion of the data for testing the model's performance after training.

- **Training Set**: The majority of the data, **4704 samples**, was used for training the model.

- **Test Set**: The remaining **1176 samples** were reserved for testing the model's performance and ensuring that it could generalize well to unseen data.

By splitting the dataset in this way, we ensured that the model could be evaluated fairly, and the risk of overfitting to the training data was minimized.

6. Data Transformation Recap

- **No missing values** were found, allowing for direct use of the data without any imputation.
- **Categorical features** were converted into numerical values using label encoding.
- **Feature scaling** was applied to continuous variables to standardize their ranges.
- **Data splitting** ensured that we had distinct training and testing sets, minimizing the risk of overfitting.

7. Key Observations After Preprocessing

- The dataset was now in a clean and standardized format, suitable for feeding into machine learning models.
- The Churn column, which is the target variable, was clearly defined, allowing us to proceed with training a classification model to predict customer churn.
- The preprocessing steps maintained the integrity of the original data while transforming it into a form that could better inform the model's predictions.

In conclusion, the preprocessing steps played a vital role in transforming the raw dataset into a format that was ready for machine learning. The careful handling of categorical variables, feature scaling, and data splitting ensured that the model could be trained efficiently and that its performance could be assessed accurately.

Model Selection and Training

After the dataset was pre-processed, the next step was to select an appropriate model and train it to predict customer churn. The goal was to create a model that could effectively classify customers into two categories: those likely to churn and those likely to stay.

1. Model Selection

For this project, the **Logistic Regression** model was selected. Logistic Regression is a statistical method that is widely used for binary classification tasks, making it a natural fit for this problem where the target variable, Churn, has two possible outcomes: "Yes" or "No".

Reasons for Choosing Logistic Regression:

- **Simplicity**: Logistic Regression is a simple model that is easy to implement and understand, which made it a good choice to start with.

- **Interpretability**: Logistic Regression provides insights into the relationships between the features and the target variable through coefficients, which makes it easier to understand the model's decision-making process.
- **Performance as a Baseline**: As a common method in binary classification, Logistic Regression can serve as a baseline model. If it performs well, it can indicate that a simple model is sufficient, and if it performs poorly, it will prompt exploring more complex models.

2. Model Training

Once the Logistic Regression model was selected, the next step was to train it using the data. The training process involved using the preprocessed dataset to "teach" the model how to recognize patterns that predict customer churn.

The training process involved providing the model with input data, including features like gender, SeniorCitizen, tenure, MonthlyCharges, TotalCharges, and others. The model then used this data to learn the relationship between these features and the target variable, which is the likelihood of a customer churning (Yes or No).

3. Model Performance Evaluation

After training the model, its performance was evaluated using the test data to understand how well it generalized to new, unseen data. Several evaluation metrics were used to assess the model's performance:

3.1. Accuracy

The **accuracy** of the model was calculated by comparing the predicted churn status with the actual churn status in the test set. Accuracy reflects the overall success of the model in predicting both churn and non-churn customers correctly. In this case, the accuracy was approximately **49%**, which indicates that the model was able to correctly predict the churn status of nearly half of the customers in the test set.

However, this relatively low accuracy suggested that the model was not capturing the patterns needed to predict churn effectively.

3.2. Classification Report

The **classification report** provided detailed performance metrics, including precision, recall, and F1-score, for each of the two classes (churn vs. no churn).

- **Precision** measures how many of the predicted churn customers actually churned.
- **Recall** measures how many of the actual churn customers were correctly identified by the model.
- **F1-score** is a harmonic mean of precision and recall, giving a balanced measure of the model's performance.

The report showed that the model had similar performance for both classes (churn and no churn), with precision and recall for churn being relatively low, especially when compared to non-churn predictions.

3.3. Confusion Matrix

The **confusion matrix** gave us a clearer picture of the model's predictions. It showed the following outcomes:

- **True Negatives (TN)**: The customers predicted not to churn who indeed did not churn.
- **False Positives (FP)**: The customers predicted to churn who did not churn.
- **False Negatives (FN)**: The customers predicted not to churn who actually churned.
- **True Positives (TP)**: The customers predicted to churn who did churn.

The confusion matrix highlighted the model's struggle to correctly identify churners (class 1), as the number of false negatives (FN) was quite high.

3.4. ROC-AUC Score

The **ROC-AUC** score was another important evaluation metric. This score reflects the model's ability to distinguish between the two classes. A value close to **1** would indicate a good model, while a value close to **0.5** suggests random guessing.

The **ROC-AUC score** for this model was **0.48**, which is very close to random guessing. This indicates that the model was not effectively differentiating between customers who would churn and those who would not, further pointing to areas for improvement.

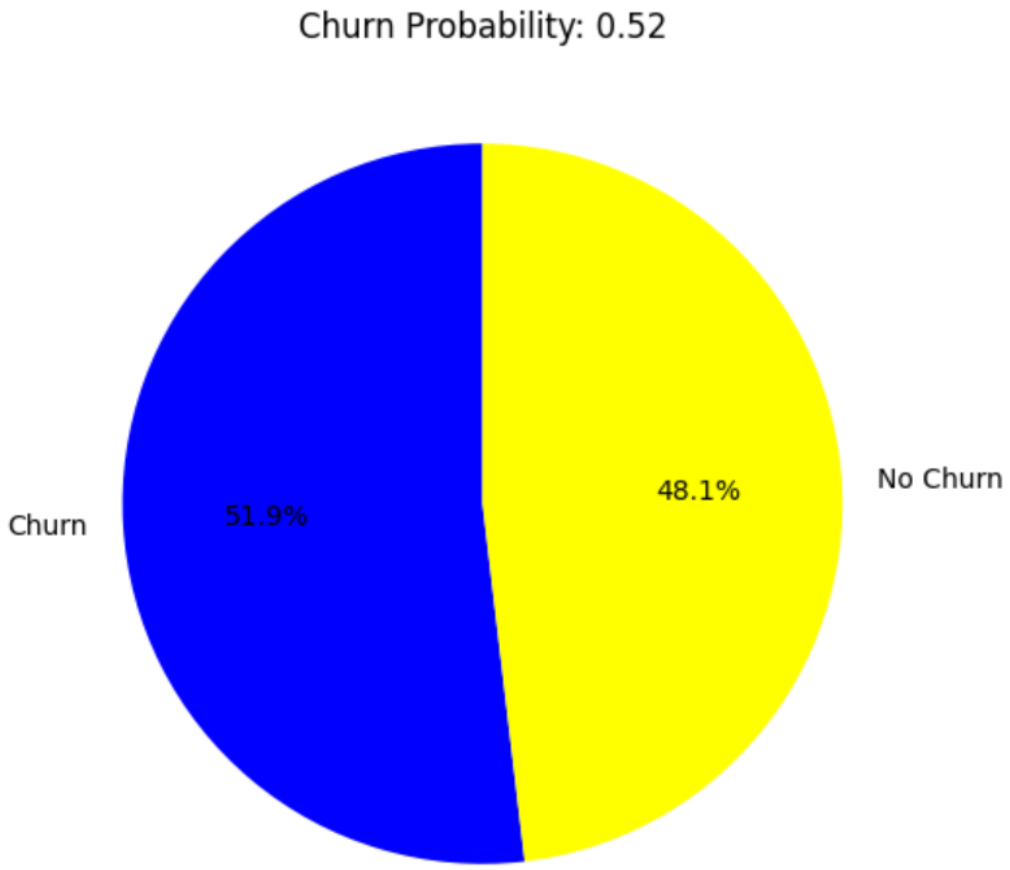
4. Summary of Model Training

To summarize, the Logistic Regression model provided a baseline for churn prediction, but its performance was suboptimal. The accuracy and evaluation metrics indicated that the model struggled to make reliable predictions, especially for predicting customer churn. The confusion matrix and low ROC-AUC score highlighted the model's inability to effectively distinguish between the churn and non-churn classes.

These results suggest that improvements could be made by either tuning the Logistic Regression model, trying more complex models, or addressing any class imbalance issues that might have impacted the model's performance.

Model Evaluation

After training the Logistic Regression model, it was essential to evaluate its performance using various metrics. This allowed us to understand how well the model predicted customer churn, and whether it could provide valuable insights for decision-making.



1. Accuracy

The first metric used to evaluate the model was **accuracy**, which measures the proportion of correct predictions made by the model. For this dataset, the accuracy of the Logistic Regression model was approximately **49%**.

While this accuracy might seem low, it is important to note that the dataset contains a highly imbalanced target variable. The churn rate in the dataset is relatively low, meaning that the model could simply predict the majority class (no churn) and achieve a higher accuracy without necessarily capturing the patterns leading to actual churn. As a result, **accuracy** alone isn't sufficient to judge the model's effectiveness in this case.

2. Precision and Recall

To gain deeper insights into the model's performance, we turned to **precision** and **recall**, which are crucial metrics for imbalanced classification problems like churn prediction.

- **Precision** measures the proportion of predicted positive outcomes (churn) that were actually correct. For this model, precision for predicting churn was low, at **49%**, meaning that about half of the predicted churns were incorrect.

- **Recall**, on the other hand, measures how many of the actual positive outcomes (real churners) were correctly predicted by the model. The recall for predicting churn was also low at **35%**, meaning that a significant number of actual churners were not correctly identified.

A **low precision** combined with **low recall** suggests that the model struggled to identify true churners effectively and often made false predictions about customers who were not likely to churn.

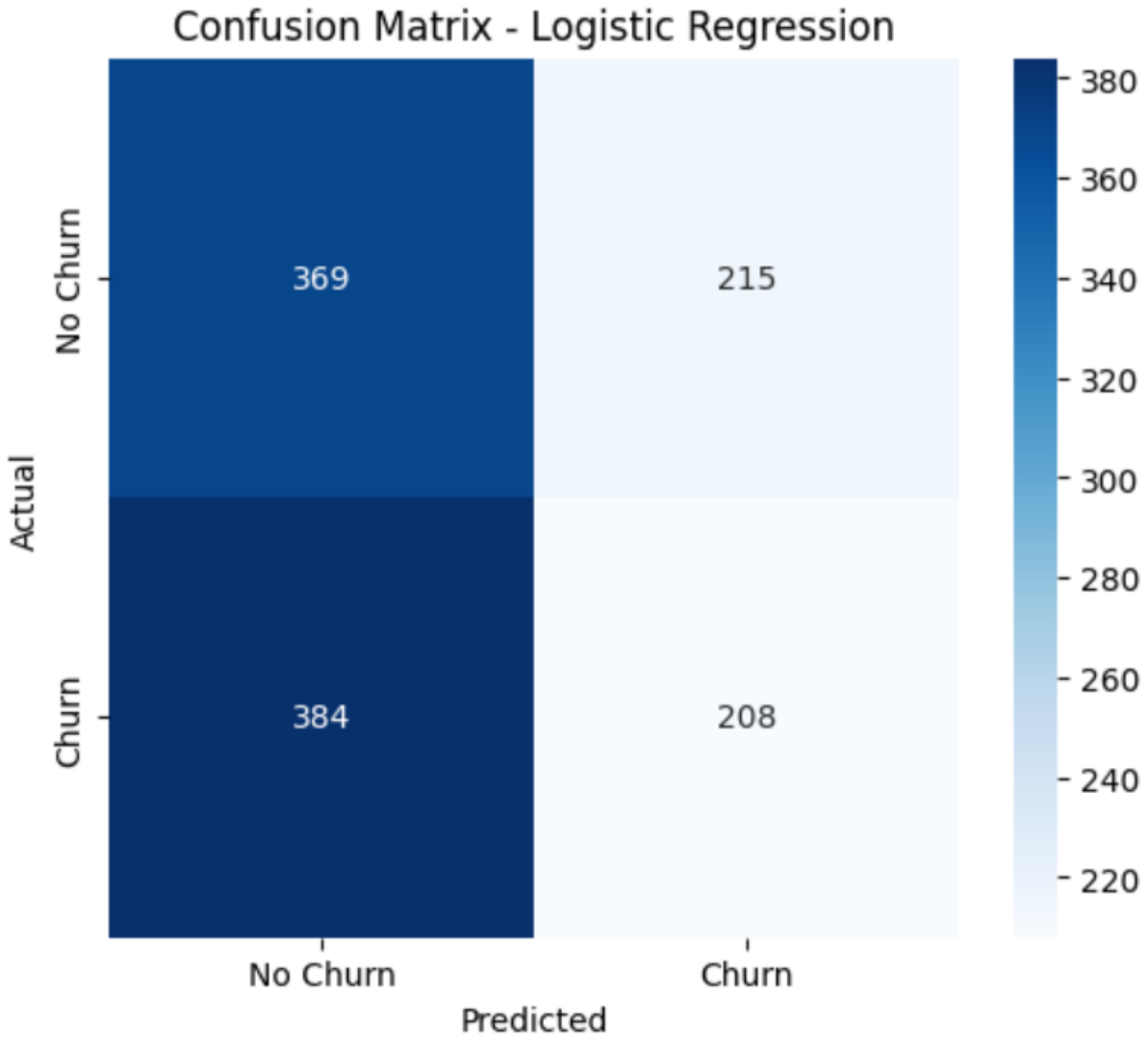
3. F1-Score

To balance the trade-off between precision and recall, the **F1-score** was computed. This metric provides a harmonic mean of precision and recall, giving a single value to represent both metrics. In this case, the F1-score for predicting churn was **0.41**, which is relatively low. A low F1-score indicates that the model's overall performance in classifying churn and non-churn customers was subpar.

4. Confusion Matrix

The **confusion matrix** provided a detailed breakdown of the model's performance by showing how many churn and non-churn cases were correctly or incorrectly predicted.

- **True Negatives (TN)**: The number of customers correctly predicted not to churn.
- **False Positives (FP)**: The number of customers incorrectly predicted to churn.
- **False Negatives (FN)**: The number of customers incorrectly predicted not to churn.
- **True Positives (TP)**: The number of customers correctly predicted to churn.

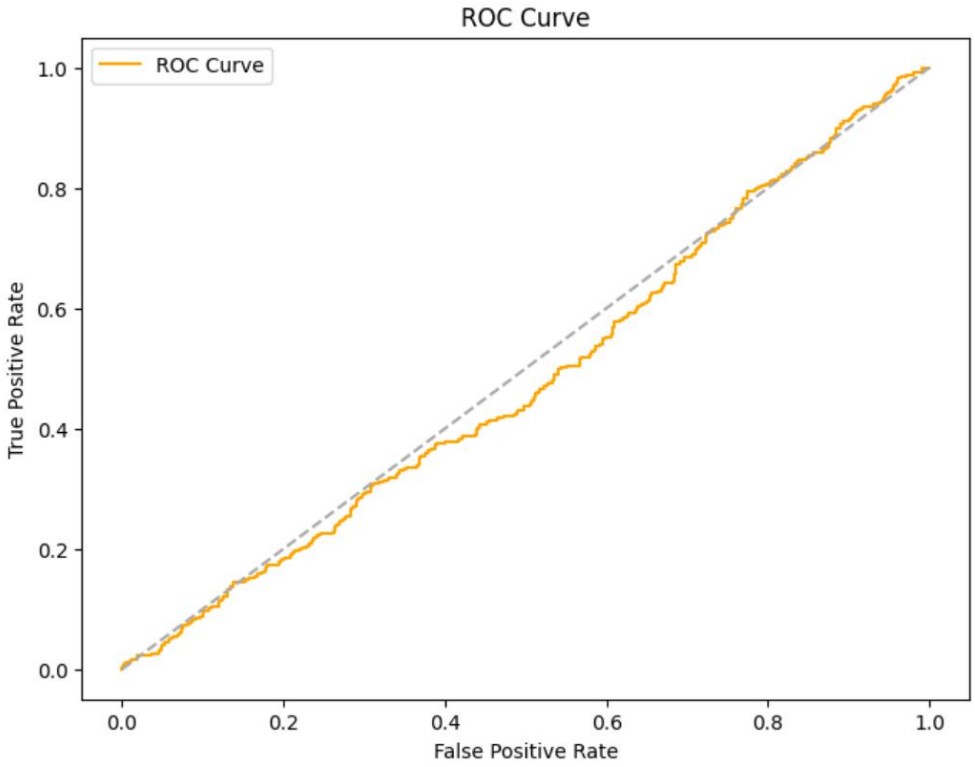


For this model, the confusion matrix showed a significant number of **False Negatives (FN)**, meaning that the model failed to identify many churners. This highlighted that, while the model was good at predicting non-churn customers, it struggled to capture the more critical churn predictions.

5. ROC-AUC Score

The **ROC-AUC** score was another important evaluation metric, which provides insight into the model's ability to distinguish between the two classes. A higher AUC score generally indicates a better ability to discriminate between the target classes.

The **ROC-AUC score** for this model was **0.48**, which is very close to 0.5, indicating that the model's predictions were almost no better than random guessing. This suggested that the Logistic Regression model was not effectively distinguishing between customers who would churn and those who would stay.



6. Model Limitations

While the model did provide some insight into customer churn, the evaluation results indicated several limitations:

- **Imbalanced Classes**: The dataset likely suffers from an imbalanced class distribution, where the majority of customers do not churn. This imbalance may have made it difficult for the model to learn the features that separate churners from non-churners effectively.
- **Model Choice**: Logistic Regression, although a simple and interpretable model, might not have been complex enough to capture the nuanced patterns within the data. More advanced models like Random Forest, Gradient Boosting, or Neural Networks could potentially provide better performance.
- **Feature Engineering**: Although the dataset was preprocessed and categorical variables were encoded, additional feature engineering and domain knowledge might help improve the model's predictions. For instance, more interactions between features or transformations might reveal hidden patterns.

7. Summary of Model Evaluation

In summary, while the Logistic Regression model was able to make some predictions about customer churn, its performance was suboptimal, with an accuracy of **49%**, a low F1-score of **0.41**, and an ROC-AUC score of **0.48**. These results highlighted the model's struggles, particularly with identifying churners accurately.

Conclusion

The goal of this project was to build a model to predict customer churn based on a variety of customer characteristics, such as tenure, service usage, and payment methods. Despite the Logistic Regression model's relatively low performance, the process provided valuable insights and highlighted areas for improvement.

Key Insights from the Dataset

1. **Customer Behavior**: A significant portion of customers has a relatively long tenure, which indicates that many customers tend to stay with the service for several years. However, the churn rate, while relatively low, presents an opportunity to focus on retaining customers who might be on the verge of leaving.
2. **Influence of Service Features**: Services like Internet Service, Online Security, and Device Protection seemed to play an important role in customer retention. This suggests that offering value-added services or improving the quality of existing services could reduce churn.
3. **Financial Factors**: The analysis of monthly and total charges revealed that higher monthly charges were correlated with a higher churn rate. This suggests that customers who feel they are paying too much may be more likely to leave the service. Pricing strategies or offering customized plans could help retain such customers.
4. **Contract Types**: The type of contract—whether month-to-month or longer-term—also had a noticeable effect on churn. Customers with month-to-month contracts were more likely to churn, while those with longer-term contracts tended to stay. Offering attractive incentives for long-term plans might help reduce churn.

Challenges and Limitations

The Logistic Regression model, though a good starting point, struggled to provide strong predictive power due to:

- **Imbalanced Data**: The dataset had an imbalanced distribution of churn and non-churn customers, which can significantly impact the performance of classification models. Techniques like resampling, or using models designed for imbalanced datasets, might improve results.
- **Model Complexity**: Logistic Regression, while interpretable and simple, might not have been complex enough to capture the nuances in the data. More advanced models like Random Forests, Gradient Boosting, or even Neural Networks could provide better results in terms of accuracy and predictive power.
- **Feature Engineering**: The model's performance could likely be improved with more advanced feature engineering. In this project, the dataset was preprocessed, but incorporating interactions between features, or even external data, could help uncover deeper patterns in customer behavior.

Potential Future Work

1. **Advanced Modeling Techniques**: Exploring more sophisticated models such as Random Forest, Gradient Boosting, or Neural Networks could improve prediction accuracy and the ability to detect churn more effectively.
2. **Hyperparameter Tuning**: Fine-tuning the hyperparameters of the chosen model could help optimize its performance, especially in complex models like Random Forest or XGBoost.
3. **Handling Class Imbalance**: Implementing techniques such as oversampling or undersampling to balance the dataset, or using algorithms designed to handle imbalanced data, could improve the model's ability to predict churn accurately.
4. **Feature Engineering**: Further exploration of feature engineering techniques, including creating interaction terms or using domain knowledge to create new features, could uncover hidden patterns in the data and improve model performance.