CUSTOMER CHURN PREDICTION IN SYRIATEL

CUSTOMER CHURN PREDICTION IN SYRIATEL

1.) Business Understanding:

Syriatel is a telecommunications company providing mobile network services in Syria.The company seeks to address the increasing attrition rates observed among its clientele.This issue directly impacts the company's revenue stability and market share, necessitating a comprehensive understanding of its underlying causes. Identifying the underlying factors driving customer defection and implementing proactive strategies to retain valuable subscribers are imperative to sustain profitability and competitiveness in the telecommunications industry.

This project therefore aims at helping SyriaTel tackle this challenge by building a classifier to predict whether a customer will soon stop doing business with the telecommunication company.Predicting customer churn allows SyriaTel to proactively address factors leading to customer dissatisfaction and implement strategies to retain valuable customers and ultimately improve customer satisfaction and loyalty, thereby reducing churn and strengthening its position in the competitive telecommunications landscape.

Problem Statement:

As SyriaTel navigates a competitive telecommunications landscape, the increasing attrition of customers poses a significant challenge to its growth and profitability. This project focuses on developing a churn prediction model to pinpoint customers likely to leave. By analyzing the factors contributing to churn, SyriaTel can adopt proactive measures to enhance customer satisfaction and loyalty, ultimately improving retention rates thereby ensuring sustained profitability in a competitive market.

Objectives:

1. Identify Churn Patterns: Analyze customer data to uncover patterns associated with churn.
2. Develop a Customer Churn Prediction Model: Design a binary classification model employing machine learning techniques to estimate the probability of a customer discontinuing services. The model should be trained on past customer data and evaluated with suitable performance metrics such as accuracy, precision, recall, and F1-score.
3. Deliver Practical Recommendations: Provide practical recommendations and insights derived from the analysis and model outcomes. These insights should enable SyriaTel to make data-driven decisions and formulate targeted strategies for minimizing churn, including enhancing customer service, delivering personalized offers, or improving plan features.

Metrics of Success:

Accuracy: Often reported between 70% to 90% depending on the model and dataset.

Precision: Usually ranges from 60% to 85%, indicating the proportion of correctly predicted churners.

Recall: Commonly falls between 50% to 80%, reflecting the model's ability to identify actual churners.

F1 Score: Typically reported around 0.6 to 0.8, balancing precision and recall.

ROC-AUC Score: Often ranges from 0.7 to 0.9, showing the model's discrimination ability.

Confusion Matrix Values: Specific counts of true positives, false positives, true negatives, and false negatives are usually provided, but exact numbers vary by study.

Conclusion

Creating a churn prediction model for SyriaTel is vital for reducing customer attrition and improving retention efforts. By utilizing machine learning and key performance metrics, the model will identify at-risk customers and provide actionable insights. This proactive approach will enable SyriaTel to enhance customer satisfaction, optimize retention strategies, and strengthen its market position, ultimately driving sustainable growth.

2.) Data Understanding

2.) Data Understanding:

The dataset was obtained from Kaggle Kaggle website. The dataset from Kaggle provides a comprehensive view of various aspects of customer behavior and characteristics within the SyriaTel's services including usage patterns, plan subscriptions, and customer service engagement and is crucial for analyzing factors that contribute to customer churn and for building predictive models.

import My_Functions as myf pd = myf.load_data("data.csv") pd.head(); **Purpose**: This code imports a custom module named `My_Functions`, which contains a function to load data from a CSV file.

Function: `myf.load_data("data.csv")` reads the dataset and loads it into a Pandas DataFrame called `pd`.

Output: `pd.head()` displays the first five rows of the dataset, providing a glimpse of the data structure and content.

The DataFrame consists of 3333 entries and 21 columns, with columns including customer details and usage statistics. Key columns include:

- **State**: Customer's state.
- **Account Length**: Length of the account in months.
- **Churn**: Indicates whether the customer has left (True/False).

Shape: Demonstrates the number of rows (customers) and columns (features).
Size: Total number of elements in the DataFrame.

In this analysis, we loaded a customer churn dataset using a custom function, revealing a structured DataFrame with 3333 customers and 21 features. The data includes essential variables such as account length, usage statistics, and churn status. Descriptive statistics provide insights into customer behavior and service utilization, essential for understanding factors influencing churn.

3.) Data Preparation

3.) Data Preparation:

In this step, we focus on ensuring that our dataset is clean and reliable for further analysis. This involves checking for missing values, duplicate entries, and outliers, all of which can impact the quality of the insights and predictive models.

- **Missing Values**: We performed a check for missing values across all columns in the dataset. The analysis revealed that there are no missing values in any of the fields. This indicates that our dataset is complete and has no missing data entries, ensuring the integrity of the dataset.
- **Duplicate Entries**: A review of the dataset for duplicate records showed that there are no duplicates. This confirms that every row in the dataset is unique, which helps avoid any redundancy or bias in our analysis.
- **Outliers**: We conducted an outlier detection analysis. Outliers can occur due to errors in data collection, faulty measuring tools, or human mistakes during data entry. Understanding the causes of outliers is important because it helps decide if they should be removed, retained, or investigated further. Outliers can influence the outcomes of our analysis and modeling, so addressing them appropriately is key to ensuring the accuracy of the results.

These checks have helped ensure that the dataset is clean and ready for in-depth analysis, which will support the generation of reliable insights and improve the performance of the predictive models.

This preparation step forms the foundation for accurate and effective exploratory data analysis and modeling in the next phase.

4.) Exploratory Data Analysis

4.) Exploratory Data Analysis:

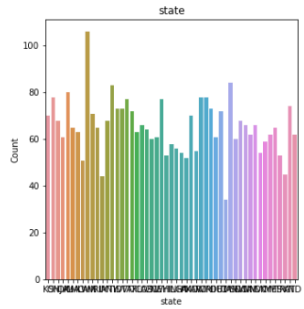
Exploratory Data Analysis (EDA) is an iterative process that helps us better understand the data and guides subsequent steps in the analysis and modeling phases. During this stage, we explore the structure, distribution, and relationships within the data, which will inform the choice of models and features for further analysis.

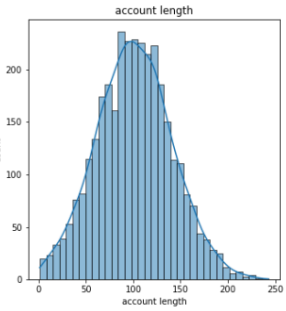
- **Numerical Columns**: We first identified the numerical columns in the dataset. These include:
    - Account length, area code, number of voicemail messages, total minutes and charges for day, evening, night, and international usage, total calls for each of these times of day, and customer service calls.
- **Categorical Columns**: We also identified the categorical columns, which include:
    - State, phone number, international plan, and voice mail plan.

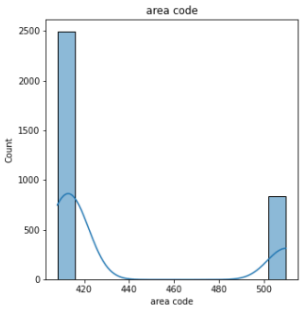
4.1 Univariate Analysis

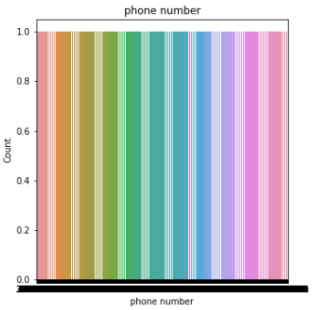
Univariate analysis focuses on the distribution and characteristics of individual variables. Key insights from our univariate analysis include:

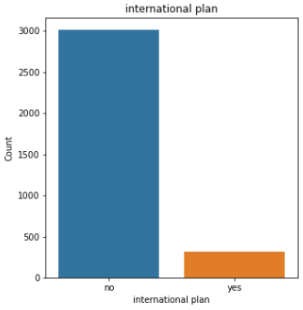
- **State Distribution**: The dataset contains a balanced distribution of customers across various states.
- **Account Length**: Most customers have an account length between 50 to 150 days.
- **Area Code**: The majority of customers are associated with the '415' area code, with '510' and '408' being less common.
- **Phone Number**: The phone number column is excluded from analysis as it serves as a unique identifier.
- **International and Voice Mail Plans**: Most customers do not subscribe to international or voicemail plans, and voicemail usage is low overall.
- **Usage Metrics**: Day, evening, and night usage (in terms of minutes, calls, and charges) are normally distributed. In contrast, international call metrics show skewness, indicating that only a few customers make many international calls.
- **Customer Service Calls**: This feature is right-skewed, with most customers making few calls.
- **Churn**: The churn variable is imbalanced, with the majority of customers not churning, which is an important consideration for predictive modeling.

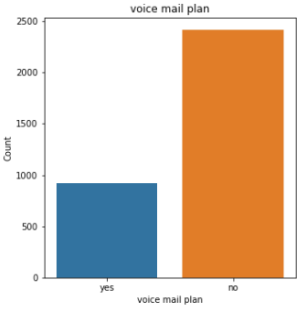

state
Count
state

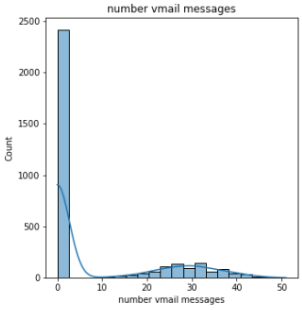

account length
Count
account length

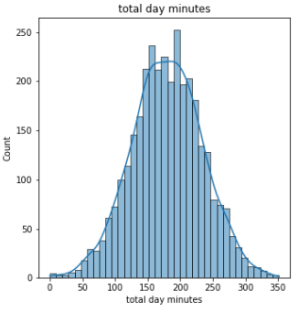

area code
Count
area code

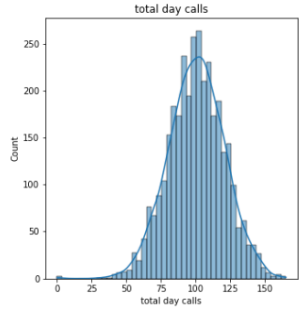

phone number
Count
phone number

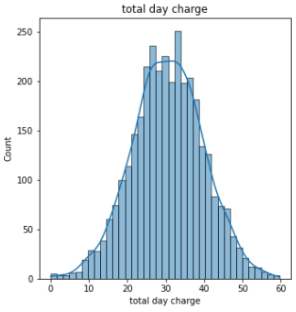

international plan
Count
no
yes
international plan

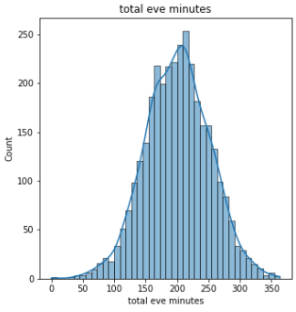

voice mail plan
Count
yes
no
voice mail plan

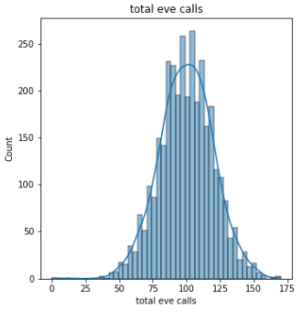

number vmail messages
Count
number vmail messages

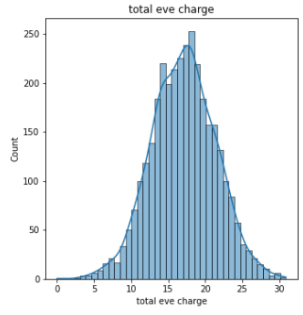

total day minutes
Count
total day minutes

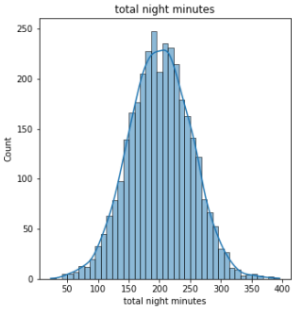

total day calls
Count
total day calls

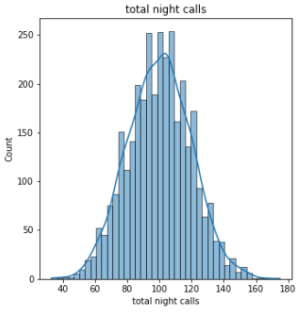

total day charge
Count
total day charge

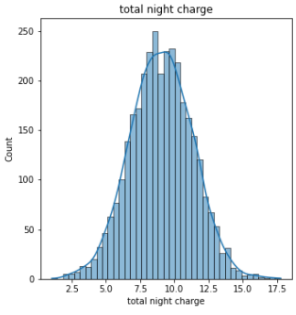

total eve minutes
Count
total eve minutes

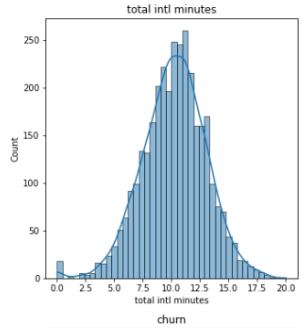

total eve calls
Count
total eve calls

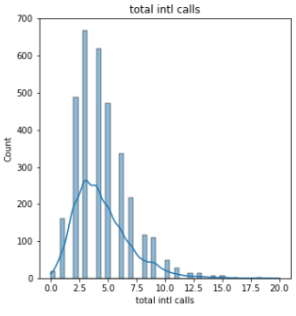

total eve charge
Count
total eve charge

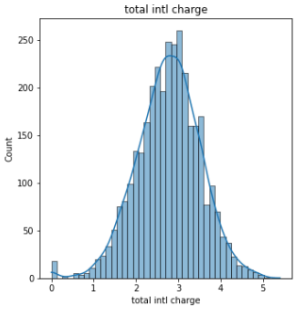

total night minutes
Count
total night minutes

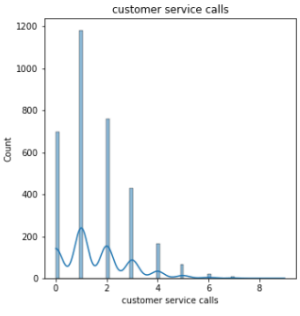

total night calls
Count
total night calls

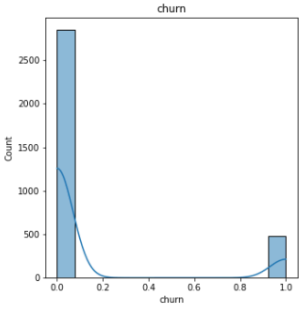

total night charge
Count
total night charge


total intl minutes
Count
total intl minutes


total intl calls
Count
total intl calls


total intl charge
Count
total intl charge


customer service calls
Count
customer service calls


churn
Count
churn

4.2 Bivariate Analysis

In bivariate analysis, we examine the relationship between two variables to identify correlations that may inform predictive models:

State: Churn rates vary by state, indicating that state could be a significant predictive feature.

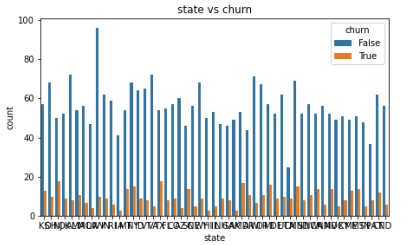
Account Length and Area Code: These variables show little correlation with churn, suggesting minimal predictive value.

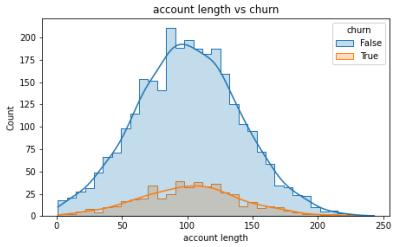
International Plan and Voice Mail Plan: Customers without an international plan or a voice mail plan tend to have higher churn rates. This suggests these features may be important for predicting churn.

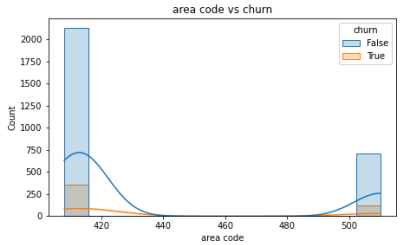
Voicemail Usage: Customers with higher voicemail usage tend to have lower churn rates, indicating a potential negative relationship with churn.

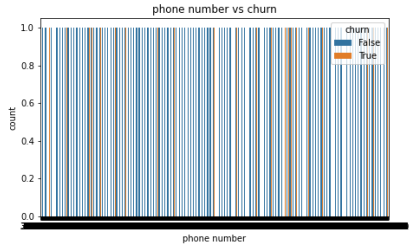
Usage Metrics: Day, evening, and night usage metrics (minutes, calls, charges) do not strongly predict churn, showing that these features are less important.

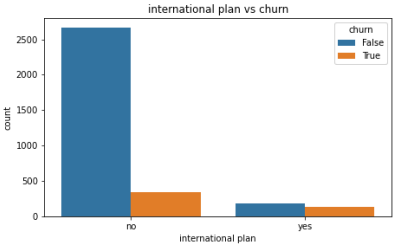
Customer Service Calls: A lower number of customer service calls correlates with lower churn rates, suggesting feature might be a significant factor in predicting churn.

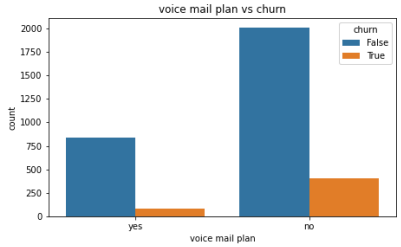

state vs churn
churn
False
True
count
state

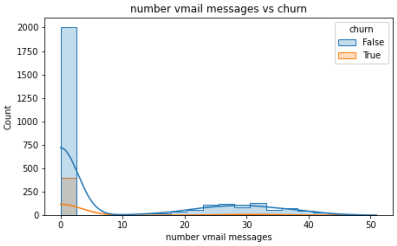

account length vs churn
churn
False
True
Count
account length

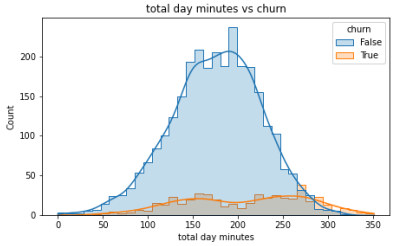

area code vs churn
churn
False
True
Count
area code

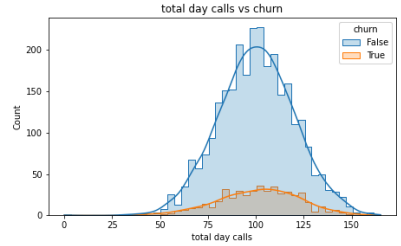

phone number vs churn
churn
False
True
count
phone number

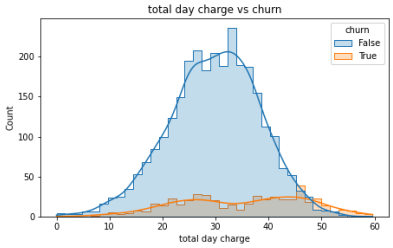

international plan vs churn
churn
False
True
count
no
yes
international plan

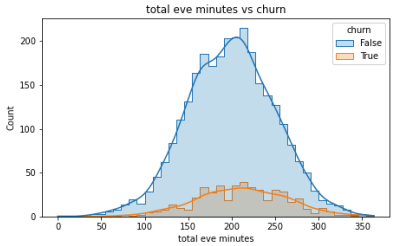

voice mail plan vs churn
churn
False
True
count
yes
no
voice mail plan

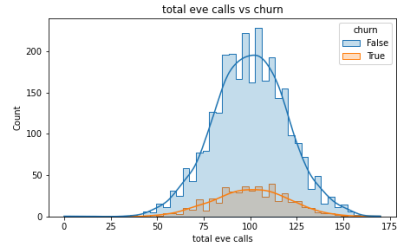

number vmail messages vs churn
churn
False
True
Count
number vmail messages

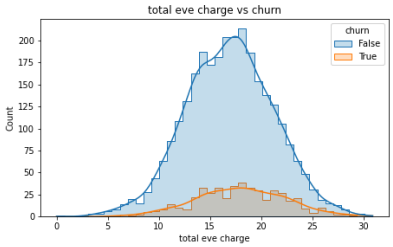

total day minutes vs churn
churn
False
True
Count
total day minutes

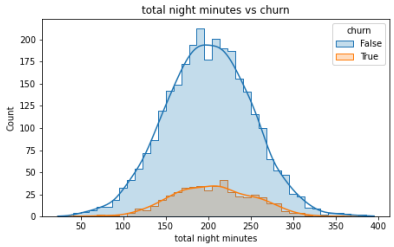

total day calls vs churn
churn
False
True
Count
total day calls

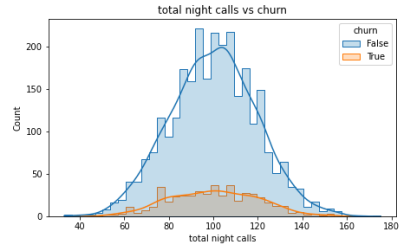

total day charge vs churn
churn
False
True
Count
total day charge

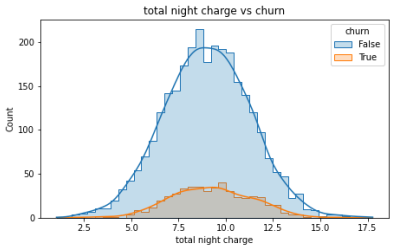

total eve minutes vs churn
churn
False
True
Count
total eve minutes

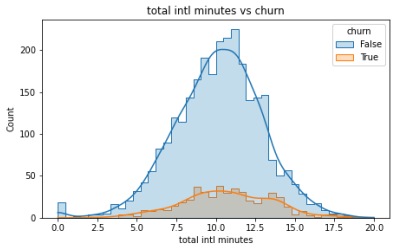

total eve calls vs churn
churn
False
True
Count
total eve calls

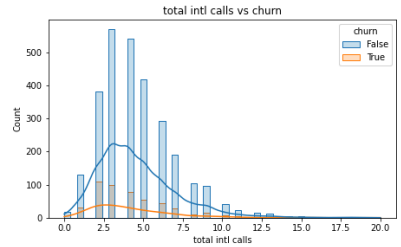

total eve charge vs churn
churn
False
True
Count
total eve charge

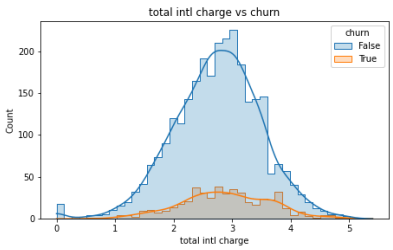

total night minutes vs churn
churn
False
True
Count
total night minutes

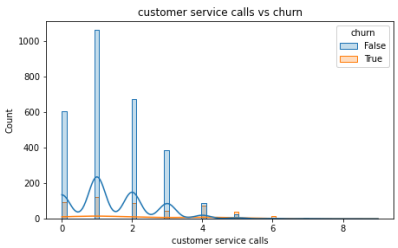

total night calls vs churn
churn
False
True
Count
total night calls


total night charge vs churn
churn
False
True
Count
total night charge


total intl minutes vs churn
churn
False
True
Count
total intl minutes


total intl calls vs churn
churn
False
True
Count
total intl calls


total intl charge vs churn
churn
False
True
Count
total intl charge


customer service calls vs churn
churn
False
True
Count
customer service calls

4.3 Multivariate Analysis

Multivariate analysis helps to identify relationships between multiple variables at once and their combined impact on churn:

- **Correlation Matrix**: The correlation matrix reveals that account length and area code have weak correlations with other variables, indicating minimal impact on churn.
- **Usage Metrics**: Total day, evening, and night metrics (minutes, calls, and charges) show strong correlations with each other, suggesting that customers exhibit consistent usage patterns across different times of day.
- **International Call Metrics and Customer Service Calls**: These features exhibit weak correlations with other variables, indicating they may not be significant predictors of churn or customer behavior.
- **Key Insights**: Overall, usage patterns are more important in understanding churn than account length or area code. Therefore, the focus should be on customer usage behaviors, such as day, evening, and night metrics, and service-related factors like customer service calls.

This comprehensive exploration of the data enables us to identify key features and relationships that will drive the next steps in our analysis and predictive modeling.

Dropping Highly-Correlated Features:

In the data preparation process, it is essential to identify and drop features that are highly correlated with each other, as they may lead to multicollinearity and affect the performance of predictive models. We define highly-correlated features as those with a correlation of 0.9 or above.

In this dataset, we observed that the charge features, such as total day charge, total evening charge, total night charge, and total international charge, are strongly correlated with the corresponding minutes features (total day minutes, total eve minutes, total night minutes, and total intl minutes). Since charges are directly proportional to the minutes used, keeping both charge and minutes as separate features could lead to redundancy in the model.

Process:

We chose to drop the charge-related columns from the dataset, specifically:

- `total day charge`
- `total eve charge`
- `total night charge`
- `total intl charge`

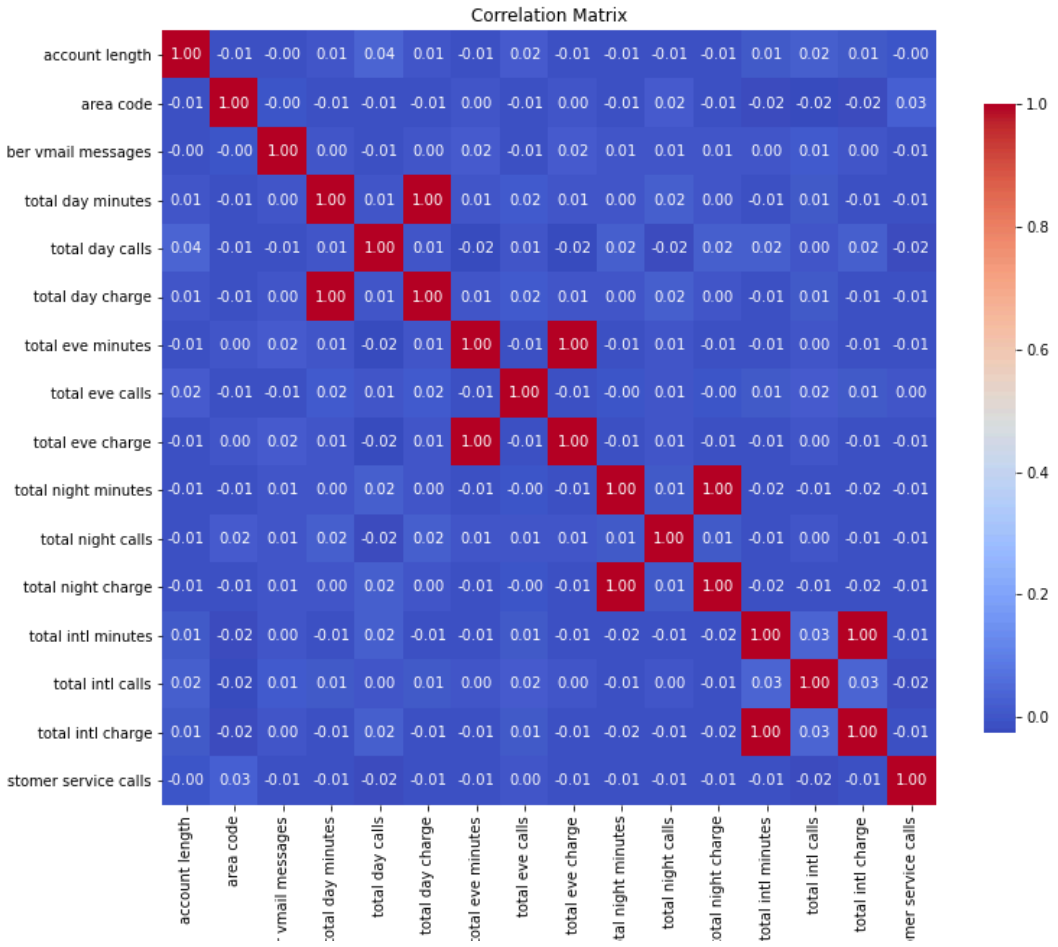
Additionally, we removed other features that were either irrelevant or highly correlated with other columns, such as:

- `state`
- `area code`
- `account length`
- `phone number`

After dropping these columns, the resulting dataset, now referred to as `trimmed_pd`, includes the following features:

- `international plan`
- `voice mail plan`
- `number vmail messages`
- `total day calls`
- `total eve calls`
- `total night calls`
- `total intl calls`
- `customer service calls`
- `churn`

This streamlined dataset retains the essential variables that are less likely to introduce redundancy, allowing for a more efficient analysis and model building process.


Correlation Matrix
account length 1.00 -0.01 -0.00 0.01 0.04 0.01 -0.01 0.02 -0.01 -0.01 -0.01 -0.01 0.01 0.02 0.01 -0.00
area code -0.01 1.00 -0.00 -0.01 -0.01 -0.01 0.00 -0.01 0.00 -0.01 0.02 -0.01 -0.02 -0.02 -0.02 0.03
ber vmail messages -0.00 -0.00 1.00 0.00 -0.01 0.00 0.02 -0.01 0.02 0.01 0.01 0.01 0.00 0.01 0.00 -0.01
total day minutes 0.01 -0.01 0.00 1.00 0.01 1.00 0.01 0.02 0.01 0.00 0.02 0.00 -0.01 0.01 -0.01 -0.01
total day calls 0.04 -0.01 -0.01 0.01 1.00 0.01 -0.02 0.01 -0.02 0.02 -0.02 0.02 0.02 0.00 0.02 -0.02
total day charge 0.01 -0.01 0.00 1.00 0.01 1.00 0.01 0.02 0.01 0.00 0.02 0.00 -0.01 0.01 -0.01 -0.01
total eve minutes -0.01 0.00 0.02 0.01 -0.02 0.01 1.00 -0.01 1.00 -0.01 0.01 -0.01 -0.01 0.00 -0.01 -0.01
total eve calls 0.02 -0.01 -0.01 0.02 0.01 0.02 -0.01 1.00 -0.01 -0.00 0.01 -0.00 0.01 0.02 0.01 0.00
total eve charge -0.01 0.00 0.02 0.01 -0.02 0.01 1.00 -0.01 1.00 -0.01 0.01 -0.01 -0.01 0.00 -0.01 -0.01
total night minutes -0.01 -0.01 0.01 0.00 0.02 0.00 -0.01 -0.00 -0.01 1.00 0.01 1.00 -0.02 -0.01 -0.02 -0.01
total night calls -0.01 0.02 0.01 0.02 -0.02 0.02 0.01 0.01 0.01 0.01 1.00 0.01 -0.01 0.00 -0.01 -0.01
total night charge -0.01 -0.01 0.01 0.00 0.02 0.00 -0.01 -0.00 -0.01 1.00 0.01 1.00 -0.02 -0.01 -0.02 -0.01
total intl minutes 0.01 -0.02 0.00 -0.01 0.02 -0.01 -0.01 0.01 -0.01 -0.02 -0.01 -0.02 1.00 0.03 1.00 -0.01
total intl calls 0.02 -0.02 0.01 0.01 0.00 0.01 0.00 0.02 0.00 -0.01 0.00 -0.01 0.03 1.00 0.03 -0.02
total intl charge 0.01 -0.02 0.00 -0.01 0.02 -0.01 -0.01 0.01 -0.01 -0.02 -0.01 -0.02 1.00 0.03 1.00 -0.01
stomer service calls -0.00 0.03 -0.01 -0.01 -0.02 -0.01 -0.01 0.00 -0.01 -0.01 -0.01 -0.01 -0.01 -0.02 -0.01 1.00
account length
area code
vmail messages
otal day minutes
total day calls
total day charge
otal eve minutes
total eve calls
total eve charge
tal night minutes
total night calls
otal night charge
total intl minutes
total intl calls
total intl charge
mer service calls
1.0
0.8
0.6
0.4
0.2
0.0

5.) Data Preprocessing

5.) Data Preprocessing

Data preprocessing is a crucial step to ensure that the dataset is in an optimal format for analysis and modeling. One important part of preprocessing is label encoding, which involves converting categorical or boolean values into numerical values. This simplifies the data handling, enhances model compatibility, and can improve the interpretability and performance of machine learning models.

5.1 Perform Label Encoding

The dataset contains several categorical features, such as `international plan`, `voice mail plan`, and `churn`, which hold boolean values ("yes"/"no" or "True"/"False"). These values need to be transformed into integers for better model compatibility. For example:

- "yes" can be mapped to 1 and "no" to 0.
- "True" can be mapped to 1 and "False" to 0.

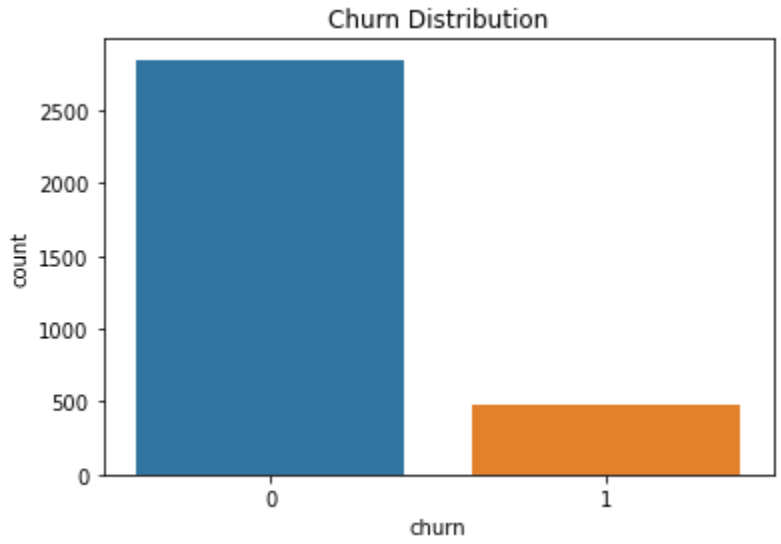
This encoding allows the machine learning model to process the features as numerical data rather than categorical text, improving computational efficiency and predictive performance.

Step-by-Step Process:

1. **Checking the Distribution of Categorical Features:**
    - The `international plan` column has 3,010 instances of "no" and 323 instances of "yes".
    - The `voice mail plan` column has 2,411 instances of "no" and 922 instances of "yes".
    - The `churn` column has 2,850 instances of "False" and 483 instances of "True".
2. **Applying Label Encoding:** We used the `.map()` function to replace categorical values with integers:
    - For the `international plan`, we mapped "yes" to 1 and "no" to 0.
    - Similarly, we mapped "yes" to 1 and "no" to 0 for the `voice mail plan`.
    - For the `churn` variable, we mapped "True" to 1 and "False" to 0.
3. **Verification:** After performing label encoding, we confirmed the transformation by printing the value counts for each of the columns:
    - The `international plan` now shows 3,010 "0"s and 323 "1"s.
    - The `voice mail plan` now shows 2,411 "0"s and 922 "1"s.
    - The `churn` column now shows 2,850 "0"s and 483 "1"s.
4. **Visualization:** We visualized the distribution of the `churn` variable using a count plot to better understand the imbalance in churned vs. non-churned customers. The plot revealed that the majority of customers did not churn, with fewer customers exhibiting a churned status.

This label encoding step has successfully converted the categorical boolean values into integers, preparing the dataset for further analysis and machine learning model development.

By encoding these columns, the dataset is now in a format that is ready for modeling, and the categorical data is appropriately converted to numerical values for analysis.



6.) Logistic Regression

6.) Logistic Regression Analysis:

6.1 Train-Test Split

To begin, we split the dataset into training and testing subsets. We used 80% of the data for training and 20% for testing, ensuring that the distribution of the target variable churn is consistent across both sets. This is accomplished by using the `stratify` parameter, which ensures proportional representation of churn in both subsets.

- **Training set size:** 2666 samples
- **Test set size:** 667 samples
- **Churn distribution:**
    - Train set churn percentage: 14.48%
    - Test set churn percentage: 14.54%

The churn distribution is almost identical across both sets, confirming a representative split.

6.2 Build and Evaluate a Baseline Model

Before diving into hyperparameter tuning, we first build a baseline logistic regression model. The preprocessing steps include handling missing values using the `SimpleImputer` and scaling the features using `StandardScaler`. We then train the model using the default parameters and evaluate it with cross-validation.

- **Log Loss (Cross-validated on training set):** 0.329
- **Log Loss (Test set):** 0.325

The close log loss values on both the training and test sets suggest that the model generalizes well.

6.3 Custom Cross-Validation Function

To ensure that preprocessing (such as feature scaling and handling class imbalance via SMOTE) is correctly applied during cross-validation, we created a custom cross-validation function. This function applies SMOTE during each fold of cross-validation, balances the classes, scales the features, and fits the model.

- **Custom Cross-Validated Log Loss:** 0.523

The custom cross-validation results show that the model performs with an average log loss of 0.523, indicating a modest predictive performance.

6.4 Evaluate Multiple Logistic Regression Models

We evaluated several logistic regression models with varying hyperparameters (C values and penalty types) to minimize log loss. The parameters tested included different regularization strengths (C values) and penalty types (L1, L2, and ElasticNet).

- **Best Model (C=0.1, penalty='l2')**
    - **Log Loss:** 0.5224

The model with C=0.1 and L2 regularization resulted in the lowest log loss, suggesting it offers the best performance.

6.5 Final Model Selection and Evaluation

After selecting the best hyperparameters, we train a final logistic regression model using C=0.1 and L2 regularization. The model is evaluated using various metrics, including log loss, accuracy, precision, recall, F1-score, and ROC AUC.

- **Log Loss:** 0.3255
- **Accuracy:** 85.46%
- **Precision:** 50.00%
- **Recall:** 13.40%
- **F1 Score:** 21.14%
- **ROC AUC:** 0.8264

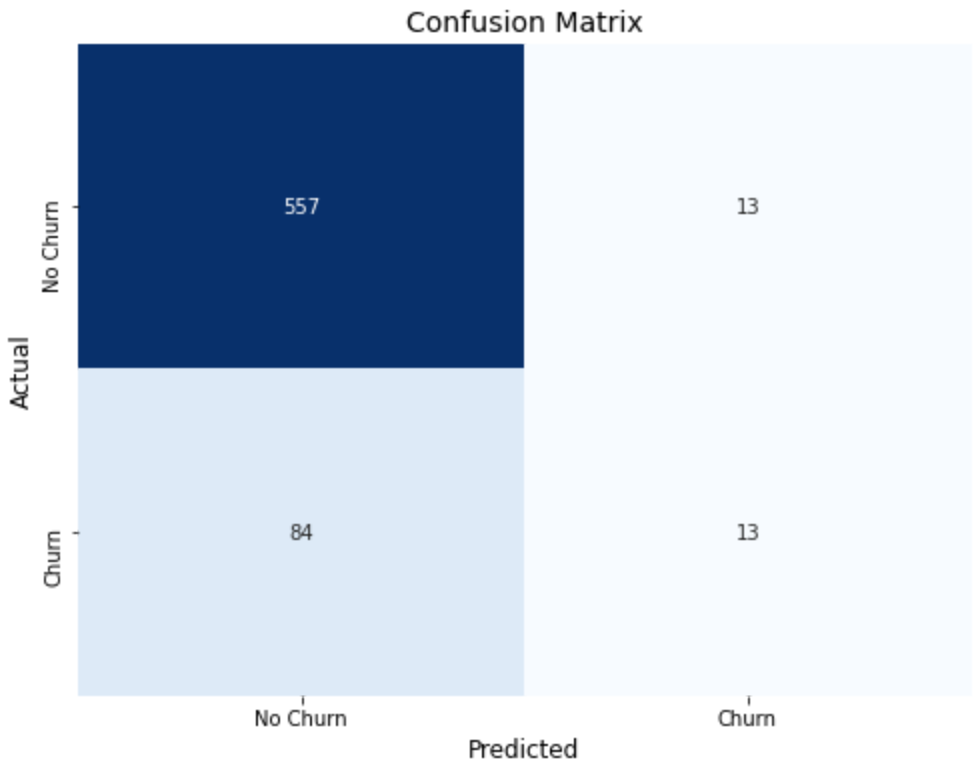
While the model achieves high accuracy, its precision and recall are lower, suggesting it struggles to correctly identify churn cases (low recall).

6.6 Confusion Matrix

The confusion matrix provides valuable insights into the performance of the model. It shows that the model correctly identifies most non-churn customers but misses many churn cases.

- **True Negatives (TN):** 557
- **False Positives (FP):** 13
- **False Negatives (FN):** 84
- **True Positives (TP):** 13

The high number of false negatives indicates that the model misses a significant number of churn cases, which is critical for customer retention efforts.

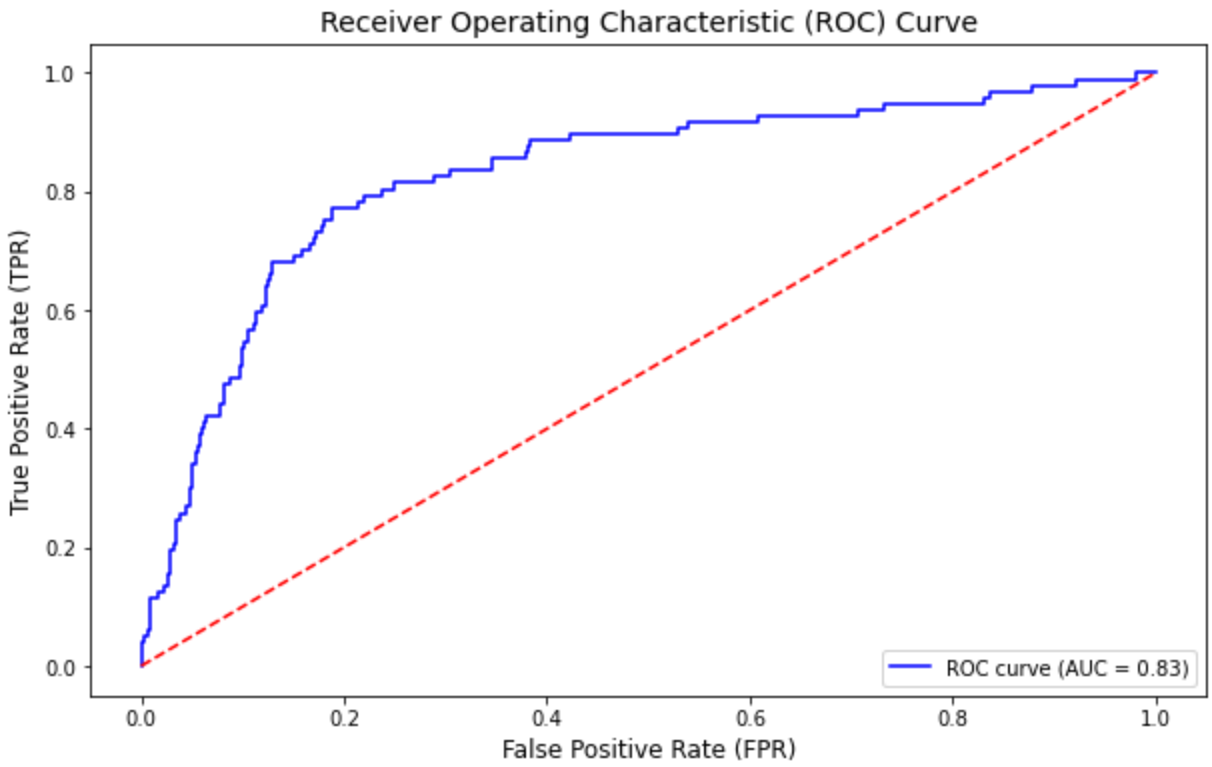


6.7 ROC Curve

The ROC curve evaluates the model's ability to distinguish between churn and non-churn customers. The area under the curve (AUC) is a key metric, with a higher AUC indicating better model performance.

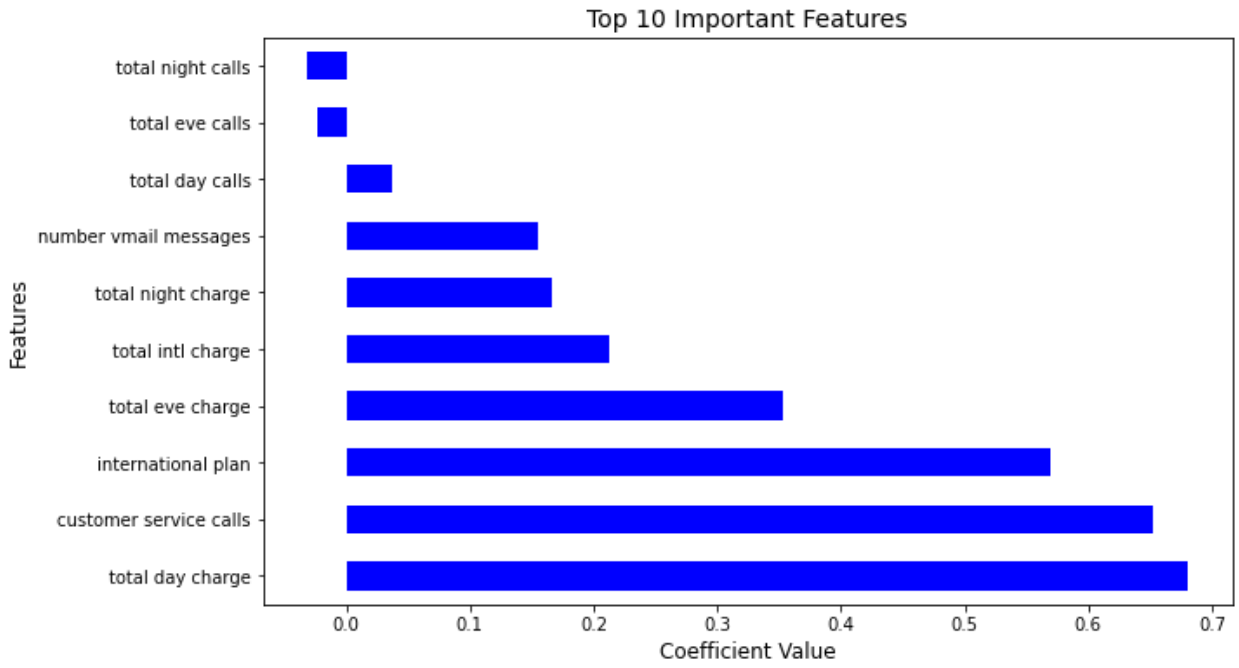
- **ROC AUC:** 0.83

This AUC value demonstrates that the model performs well in distinguishing between churn and non-churn cases.



6.8 Feature Importance

Feature importance is derived from the coefficients of the logistic regression model. Features with larger absolute coefficients have more influence on the model's predictions.



- **Top Features Influencing Churn Prediction:**
    1. **Customer Service Calls:** Strongest positive influence on churn. More calls correlate with higher churn likelihood.
    2. **International Plan:** Significant impact on churn behavior.
    3. **Total Night Calls** and **Total Day Calls** also contribute to churn predictions.

These features are positively correlated with churn, meaning that higher values in these attributes increase the likelihood of customer churn.

Hyperparameter Tuning with Grid Search

To optimize the logistic regression model, a grid search was performed to find the best combination of hyperparameters.

- **Best Hyperparameters from Grid Search:** C=100, penalty='l2'

This combination of hyperparameters yielded the best performance during grid search.

Cross-Validation Scores

Cross-validation scores were calculated to assess the stability and generalization of the final model.

- **Cross-Validation Scores:** 86.5% average accuracy across 5 folds.

These results indicate that the model performs consistently across different subsets of the data.

Conclusion

The logistic regression model provides a solid baseline for predicting customer churn, with an AUC of 0.83 indicating strong discriminatory power. However, the model's ability to identify churners (recall) is lower than desired, suggesting room for improvement, particularly in reducing false negatives. Hyperparameter tuning, cross-validation, and feature importance analysis were essential steps in optimizing model performance.

7.) Decision Trees

7.) Decision Trees:

Decision trees are a powerful tool for predictive modeling due to their ease of interpretation, ability to handle non-linear relationships, and versatility with both numerical and categorical data. They don't require feature scaling and can automatically select important features. Additionally, decision trees are robust to outliers, making them a strong choice for many classification problems.

7.1 Model Initialization and Training

To begin with decision trees, we initialize a basic decision tree classifier. The model is trained using the training data that has been preprocessed (scaled). After training, the decision tree learns patterns in the data that allow it to predict outcomes on unseen data..

7.2 Model Evaluation

Once the model is trained, we evaluate its performance on the test set. We measure several important metrics, including accuracy, precision, recall, F1 score, and ROC AUC score.

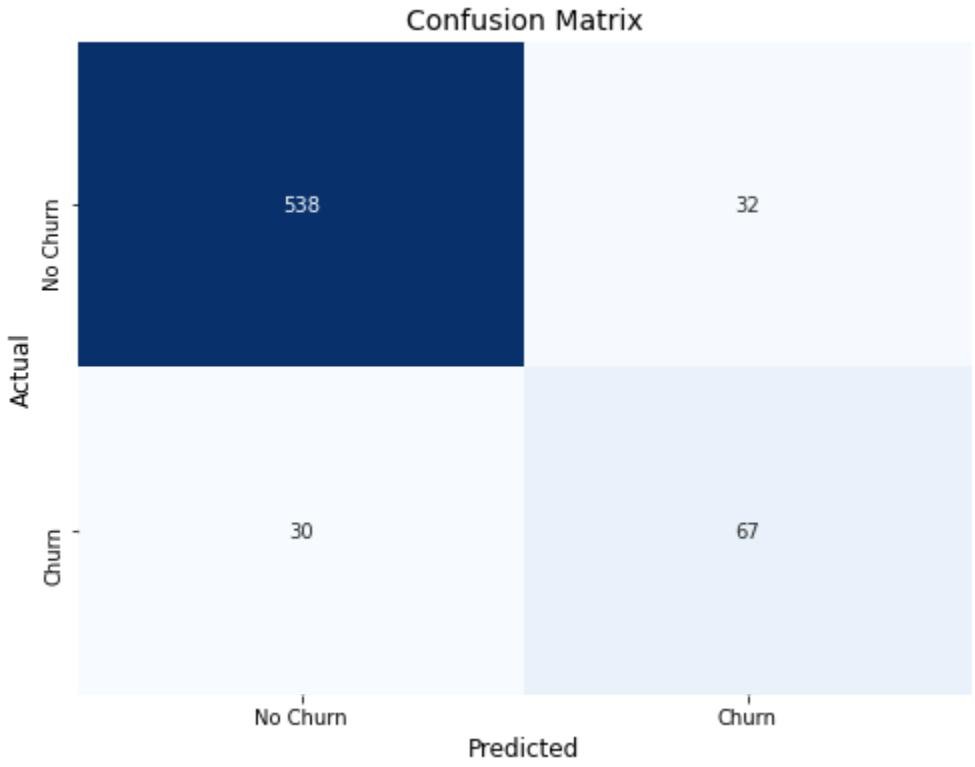
- **Performance Metrics:**
    - **Accuracy (90.7%)**: The model correctly predicts the outcome 90.7% of the time.
    - **Precision (67.7%)**: When the model predicts churn, it is correct 67.7% of the time.
    - **Recall (69.1%)**: The model correctly identifies 69.1% of actual churn cases.
    - **F1 Score (68.4%)**: A balanced measure of precision and recall.
    - **ROC AUC (81.7%)**: The model has a strong ability to discriminate between churn and non-churn, with an 81.7% area under the ROC curve.

7.3 Confusion Matrix

A confusion matrix is used to visualize the performance of the decision tree model in classifying churn vs. non-churn cases. It reveals misclassifications and shows where the model is performing well and where improvements are needed.

- **Confusion Matrix Breakdown:**
    - **True Negatives (538):** Correctly predicted "No Churn".
    - **False Positives (32):** Incorrectly predicted "Churn" when it was actually "No Churn".
    - **False Negatives (30):** Missed churn predictions (predicted "No Churn").
    - **True Positives (67):** Correctly predicted "Churn".

This confusion matrix indicates that the model is better at predicting "No Churn" than "Churn."



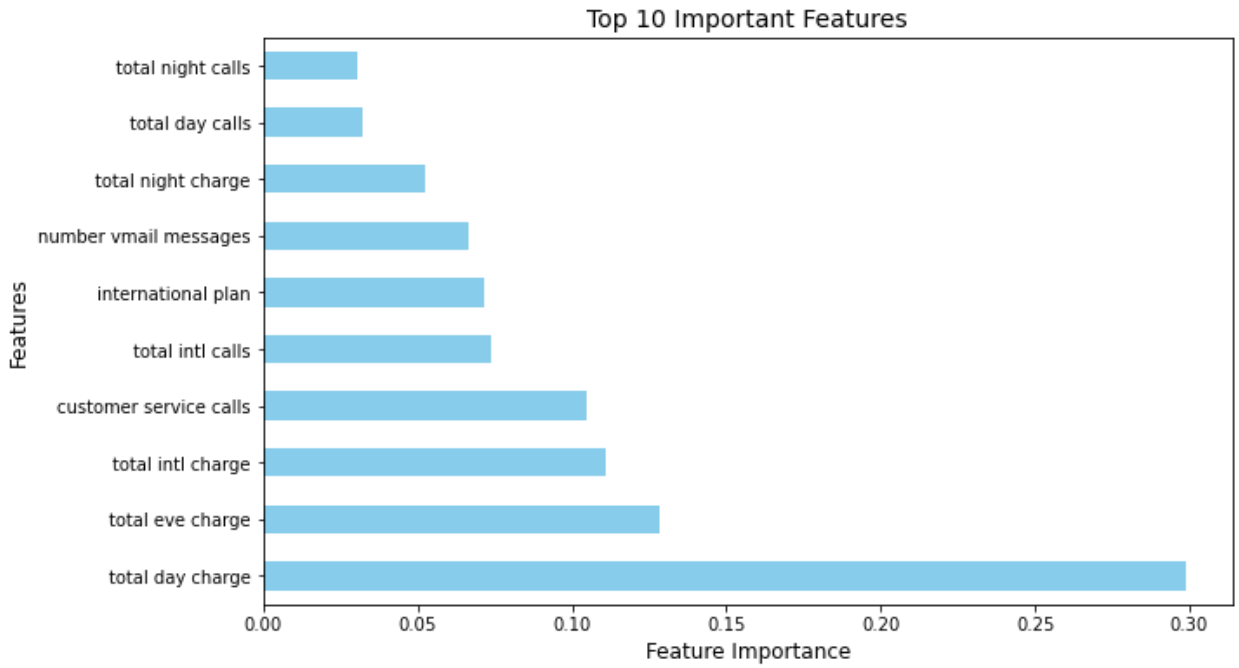
7.4 Model Tuning

We fine-tune the decision tree model's hyperparameters to improve its performance. Using grid search, we adjust parameters such as `max_depth`, `min_samples_split`, and `min_samples_leaf`. These adjustments help the model avoid overfitting while enhancing its predictive accuracy.

- **Best Hyperparameters Found:**
  - `max_depth = 10`: Allows for deeper trees, improving model fitting.
  - `min_samples_leaf = 5`: Requires at least 5 samples in each leaf node.
  - `min_samples_split = 2`: Sets the minimum number of samples required to split a node.

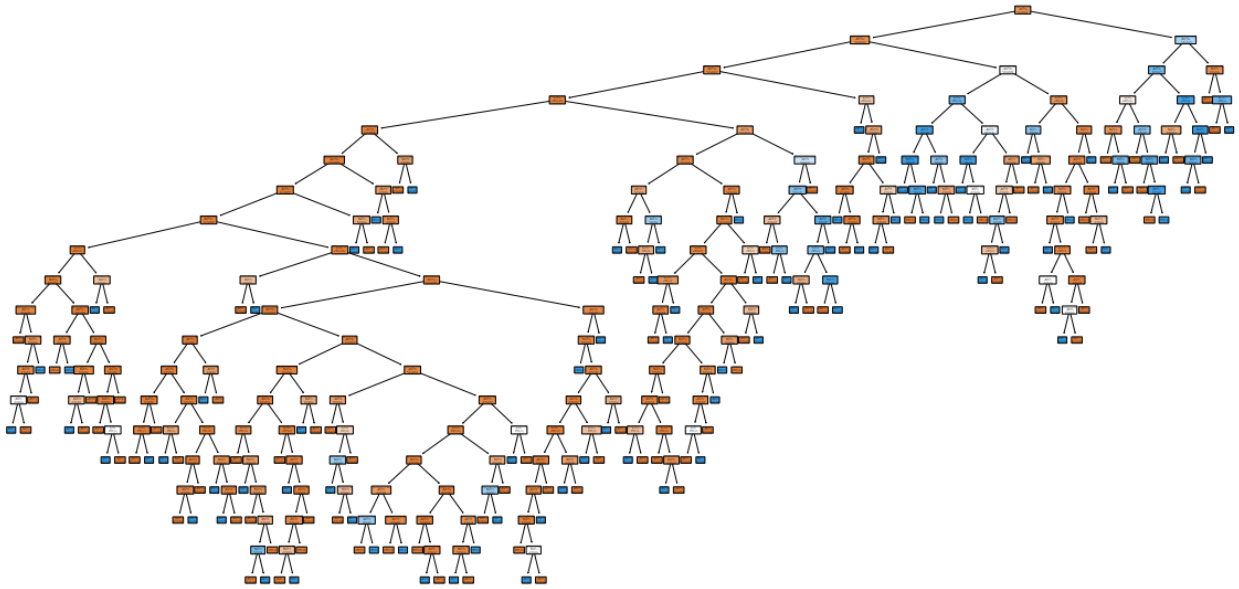
7.5 Feature Importance

One of the great advantages of decision trees is their ability to directly provide feature importance. Feature importance tells us which features are most important for making predictions. This can be useful for understanding what factors are most relevant in predicting churn, and can guide decisions on feature engineering or further analysis.



7.6 Model Visualization

Visualizing a decision tree helps us understand how the model makes decisions based on the features. Each node in the tree represents a decision point, where the data is split based on a particular feature. The branches represent possible outcomes based on the decision at each node, and the leaf nodes represent the final prediction or class. This visualization helps in interpreting the decision-making process and understanding the logic behind predictions.



7.7 Testing Precision and Accuracy on Unseen Data

After the model is trained and tuned, we test it on new, unseen data. This allows us to check how well the model generalizes to new instances. Precision and accuracy are two key metrics that can help us understand how reliable the model is on real-world data.

8.) Evaluation

8.) Evaluation:

Metrics Comparison: Accuracy:

LR: 0.8546 DT: 0.9070 Rank: DT > LR Precision:

LR: 0.5000 DT: 0.6768 Rank: DT > LR Recall:

LR: 0.1340 DT: 0.6907 Rank: DT > LR F1 Score:

LR: 0.2114 DT: 0.6837 Rank: DT > LR ROC AUC:

LR: 0.8264 DT: 0.8173 Rank: LR > DT

- DT outperforms LR across most metrics, especially in terms of accuracy, precision, recall, and F1 score, indicating it's more reliable at distinguishing between churn and non-churn cases.
- LR slightly outperforms DT in ROC AUC, suggesting it might have a marginally better ability to differentiate between the classes at different thresholds, though both models perform similarly in this aspect.
- DT is a better overall model in this case, especially considering its higher accuracy and ability to identify both churners and non-churners effectively.

9.)Recommendations

9.) Recommendations:

Improve Feature Engineering: Customer Service Calls and International Plan are key features influencing churn. Consider creating new features or using feature interactions (e.g., customer service calls combined with tenure) to improve model performance.

Class Imbalance: The Logistic Regression model struggles with recall, as it fails to predict many churn cases. Ensure that SMOTE or other balancing techniques are consistently applied during model training to address class imbalance, particularly for models with poor recall.

Threshold Adjustment: Consider adjusting the decision threshold for the Logistic Regression model to improve recall without significantly hurting precision. A lower threshold could help in capturing more churn cases (FN) while balancing the precision-recall trade-off.

Hyperparameter Tuning: Grid Search or Random Search could be employed to further tune hyperparameters for Logistic Regression (C, penalty type) and Decision Trees (max depth, min samples split, max features). Hyperparameter optimization is essential to avoid overfitting and improve generalization.

Feature Importance: Decision Trees show strong performance with a clear interpretation of feature importance. Focus on improving features such as Customer Service Calls and International Plan, which have the highest impact on churn prediction.

Customer Retention Strategy: Based on model insights, focus retention efforts on customers with high churn likelihood (identified by both models), particularly those showing frequent customer service calls and high usage of certain features like International Plan.

10.)Conclusions

10.) Coclusions:

Model Performance: Logistic Regression performed reasonably well in terms of accuracy (85.46%) but struggled with recall (13.4%), meaning it missed a large portion of the actual churn cases. This indicates the model's difficulty in identifying churners, which is critical in a customer retention context. Decision Tree showed superior performance in terms of recall and precision, though it had some limitations regarding generalization (tendency to overfit), suggesting that while it performs well on training data, its ability to generalize to unseen data could be improved.

Key Features: The most important features influencing customer churn are Customer Service Calls and International Plan. These features are strongly correlated with churn, indicating that customers who interact more with customer service or have international plans are more likely to churn. Feature Engineering improvements, such as creating interaction terms or new derived features, could help improve model performance further.

Class Imbalance: Both models were impacted by class imbalance, where the number of churn cases is much lower than non-churn cases. While SMOTE (Synthetic Minority Over-sampling Technique) was applied to balance the data, it's clear that better handling of this imbalance is critical to improving recall.

Model Interpretability: Decision Trees offer better interpretability with clear insights into which features contribute most to the prediction. This is important for understanding customer behavior and informing business decisions, especially in churn prediction.

11.)Next Steps

11.) Next Steps:

Deployment for Access to End Users: Create user access dashboards or reports using tools like Power BI, Tableau, or Dashboards in Python (using libraries like Streamlit or Dash) that display key churn prediction insights and trends, such as the number of at-risk customers or the impact of specific features on churn.

Collecting More Data Points: Collect more detailed customer behavior data such as usage patterns, customer complaints, account changes, and service issues. This data can provide more insights into what drives churn beyond basic features like customer service calls or international plan usage. Integrate survey responses or direct feedback from customers to capture qualitative factors contributing to churn, such as dissatisfaction with product quality, pricing, or customer support.

Feature Engineering: Use this new data to enhance feature engineering efforts. Derive new features that might be more predictive of churn, such as customer engagement scores, social media sentiment, or usage decay patterns.

Hyperparameter Tuning: Conduct hyperparameter tuning for both the Logistic Regression and Decision Tree models to optimize their performance. This includes experimenting with different settings for the regularization of Logistic Regression and pruning strategies for Decision Trees.

Scalability: Scale the Model: As the company grows, scale the model to handle a larger customer base by optimizing the deployment infrastructure, for example, moving to cloud-based solutions.