Subtitle: Customer Churn Analysis - Improving Retention Through Data-Driven Strategies
Prepared by: Gamuchirai Hlatywayo
Date: 05/05/2025

1. Introduction & Dataset Description

Customer churn—when customers discontinue their relationship with a business—is a pressing issue for companies seeking sustainable growth. For telecommunications providers, churn not only results in lost revenue but also increases costs associated with acquiring new customers. Nonetheless, customer churn is a significant challenge for businesses, particularly in these telecommunication industries. This project aims to predict customer churn using machine learning models, identify key drivers of churn, and provide actionable recommendations to improve customer retention. The purpose of this report is to leverage data analytics and machine learning techniques to predict churn and understand its drivers, paving the way for actionable strategies that increase retention rates.

This project integrates exploratory data analysis (EDA), preprocessing steps, and advanced predictive models such as Logistic Regression, Random Forest, and XGBoost. Additionally, ethical considerations and interpretability efforts ensure responsible deployment of the insights derived from machine learning models.

Significance of Churn Prediction

Effective churn prediction enables businesses to identify high-risk customers early and take proactive measures, such as offering incentives or addressing dissatisfaction. This approach not only reduces churn-related losses but also builds long-term customer loyalty.

Dataset Description

The dataset, sourced from Kaggle's Telco Customer Churn dataset, contains detailed records of customer demographics, service usage, payment methods, and subscription details. These features are critical in identifying patterns and correlations to predict churn effectively.

- **Size:** 7,043 rows and 21 columns.
- **Key Features:**
  - **Demographics:** Gender, SeniorCitizen, Dependents.
  - **Subscription:** Tenure, Contract type, MonthlyCharges, TotalCharges.
  - **Payment Details:** PaymentMethod, PaperlessBilling.
  - **Target Variable:** Churn (binary: Yes/No).

Table Example: Dataset Snapshot

Feature	Description	Example Values
tenure	Duration of service in months	1, 34, 72
Contract	Type of contract	Month-to-Month, One Year
MonthlyCharges	Charges billed per month	29.85, 56.95, 118.75
Churn	Customer churn indicator	Yes, No

2. Exploratory Data Analysis (EDA)

Objective

EDA investigates the dataset to uncover underlying patterns, visualize relationships, and identify inconsistencies such as missing values or anomalies.

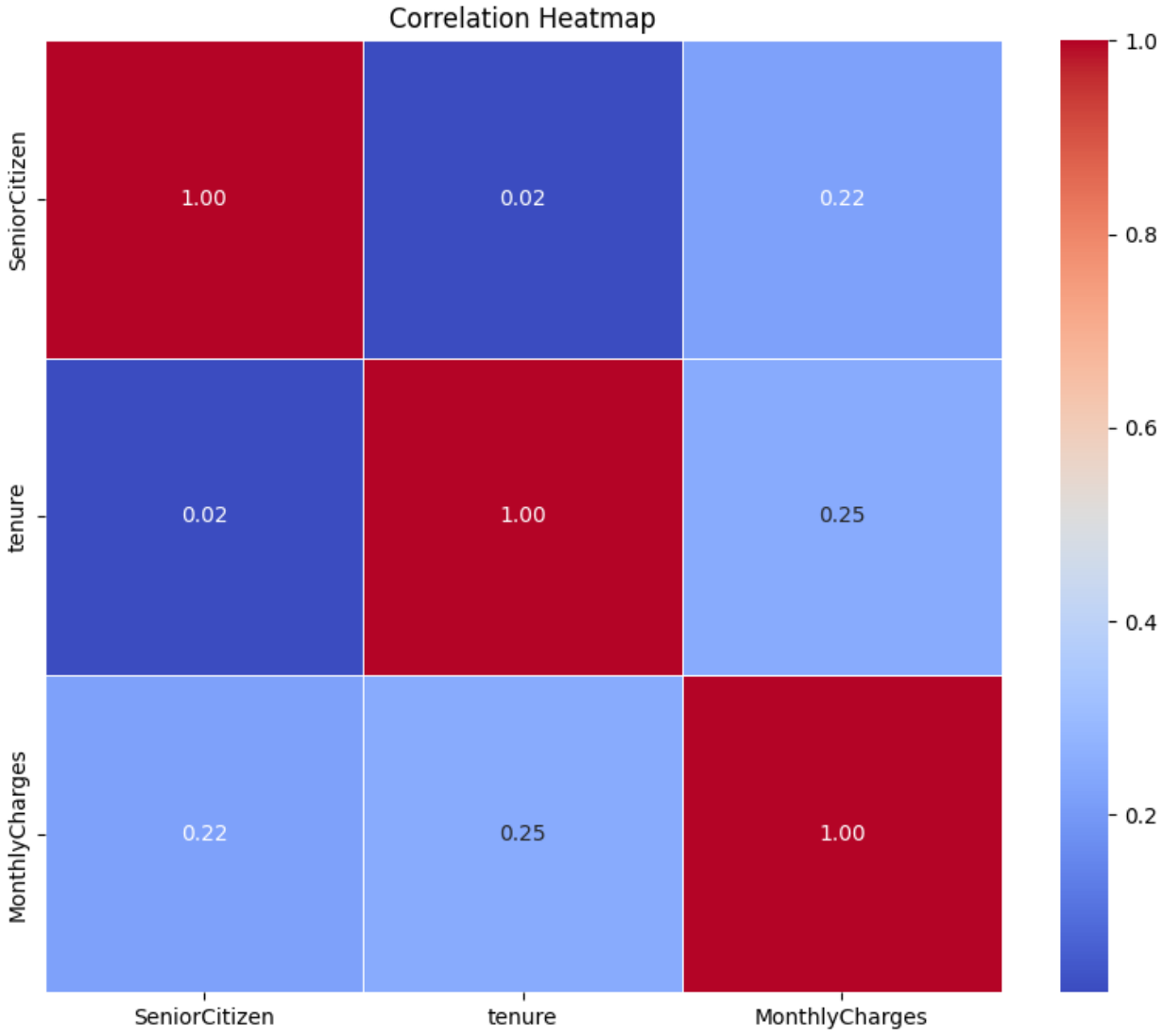
Steps in EDA

1. **Initial Overview:**
   - **Missing Values:** Detected 11 missing entries in the TotalCharges column.
   - **Imbalanced Data:** Churn shows 26% churners vs. 74% non-churners.
2. **Visualization Highlights:**
   - **Histograms:** Display distributions of numerical features (MonthlyCharges, tenure).
   - **Heatmap:** Correlation analysis revealed strong ties between MonthlyCharges and TotalCharges.
   - **Count Plots:** Key categorical features like InternetService and PaymentMethod reveal usage patterns.

Key Insights

- Customers with **higher monthly charges** are more likely to churn.
- **Fiber Optic Internet users** show higher churn than DSL users.
- Month-to-Month contract holders exhibit far greater churn rates than those on longer contracts.
- **Churn Distribution:** Class imbalance observed: fewer churners ("Yes") compared to non-churners ("No").
- **Tenure:** Average tenure is 32.4 months, with many customers leaving early (minimum tenure = 0).
- **Monthly Charges:** Higher charges correlate with premium services, ranging from $18.25 to $118.75.
- **Internet Service:** Fiber Optic users churn more frequently than DSL users.
- **Contract Type:** Month-to-month contracts have higher churn rates compared to 1-year or 2-year contracts.
- **Payment Method:** Electronic check users exhibit higher churn likelihood.

1. *Correlation heatmap for numerical features.*



Key Observations from the Heatmap:

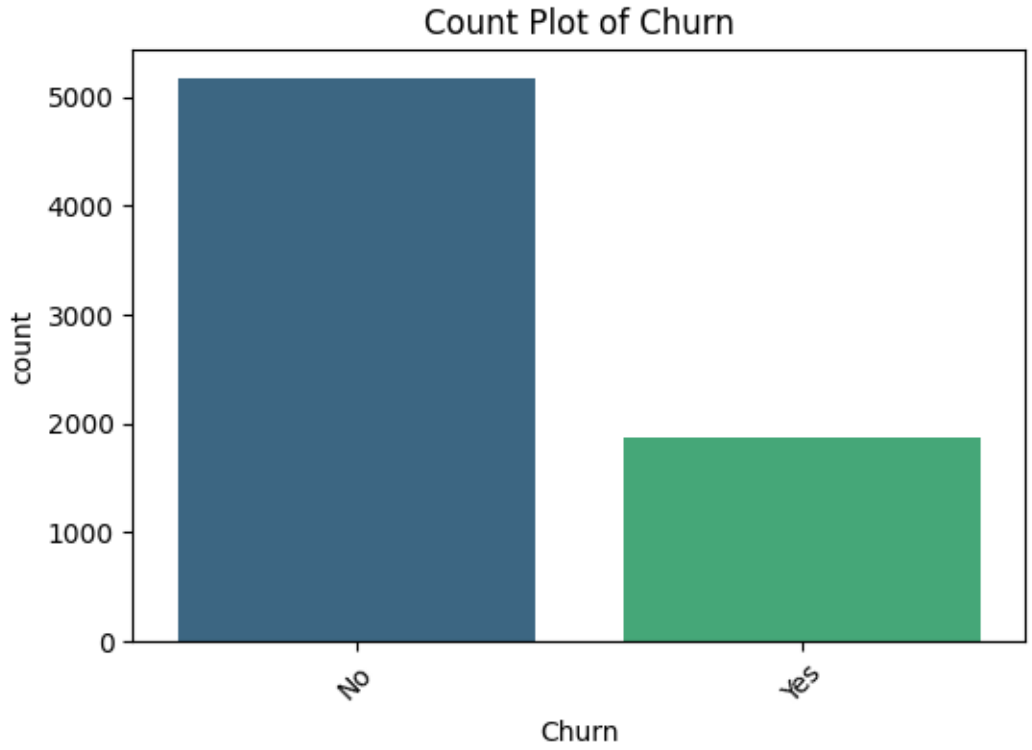
1. **Senior Citizen vs. Tenure (Correlation: 0.02)**
   - Extremely weak correlation, suggesting that being a senior citizen has little impact on how long a customer remains subscribed.
2. **Senior Citizen vs. Monthly Charges (Correlation: 0.22)**
   - A weak positive correlation, meaning that senior citizens tend to have **slightly higher monthly charges**, but the relationship isn't very strong.
3. **Tenure vs. Monthly Charges (Correlation: 0.25)**

- A weak positive correlation, implying that **longer-tenured customers may pay slightly higher monthly charges**, possibly due to contract renewals or added services over time.

Implications for Customer Churn Analysis

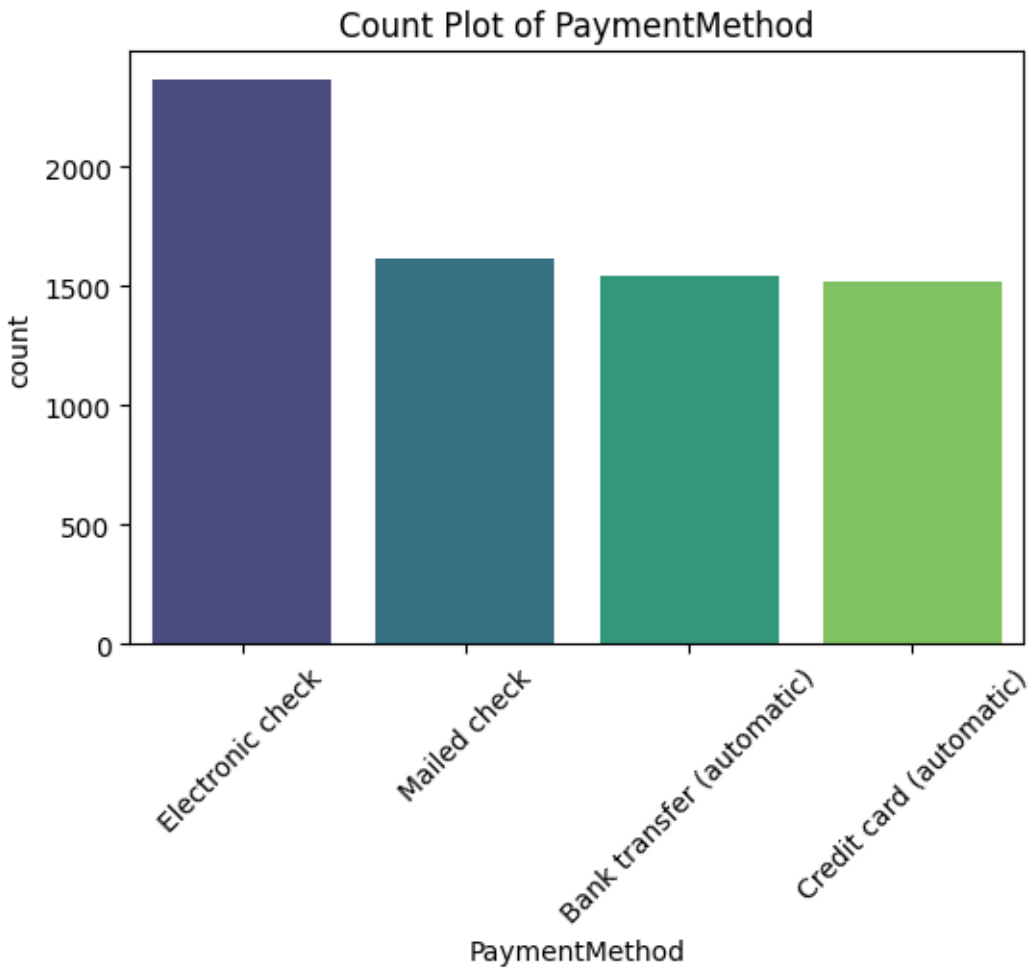
- Since the correlation values are relatively low, none of these variables strongly dictate churn behavior.
- Other features (e.g., contract type or payment method) may be **more significant predictors of churn** than tenure or senior citizen status.
- If optimizing pricing strategies, **monthly charges could be explored further** to understand its interaction with other variables impacting churn.

2. *Count plots*



Insights from the Churn Count Plot

- **Churn is Imbalanced:** Since the majority of customers did not churn, the dataset is **imbalanced**, which could affect machine learning models. Techniques like **SMOTE (Synthetic Minority Oversampling Technique)** or weighting methods may help balance predictions.
- **Business Implications:** Given that **26% of customers churn**, retention efforts should focus on understanding the drivers behind churn and implementing strategies to increase loyalty.
- **Next Steps:** Analyzing churn rates across different features (e.g., contract type, payment method) could reveal high-risk customer segments.



Observations from the Chart:

1. **Dominance of Electronic Check:**
    - The highest bar represents **Electronic Check**, with a count slightly above **2,000** customers.
    - This suggests it is the most popular payment method but might also correlate with higher churn rates (as observed in previous analysis).
2. **Other Payment Methods:**

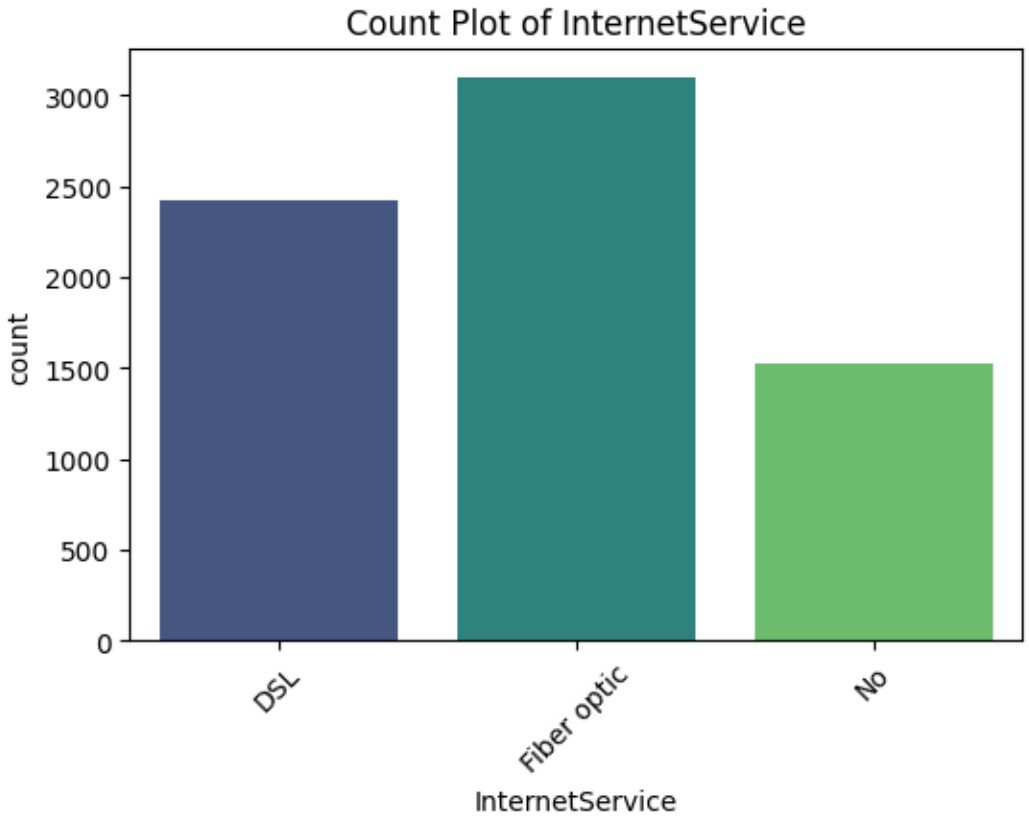
- **Mailed Check, Bank Transfer (automatic), and Credit Card (automatic)** have relatively similar frequencies, each around **1,500** customers.
- This indicates a more evenly spread preference among these payment types.

Implications for Business Strategy

- **Electronic Check customers may require retention efforts** if this method is linked to higher churn.

 Encouraging customers to shift to automated payment methods (Bank Transfers or Credit Cards) might improve retention.

- **Further segmentation analysis** on customer behavior by payment type could uncover deeper patterns influencing churn



Key Observations from the Chart:

1. **Fiber Optic is the Most Popular Service**
   - The highest bar represents **Fiber Optic**, with a count of approximately **3100** customers.
   - This suggests it is the dominant choice but may also correlate with higher churn rates.
2. **DSL Usage**

    - **DSL** is the second most used service, with **around 2500 customers**.
    - This could indicate that DSL is an alternative for users who prefer lower-cost internet services.
3. **No Internet Service Category**

- The **No Internet Service** bar, at **approximately 1500 customers**, is the lowest.
- Customers in this category likely use other telecom services without internet.

Implications for Customer Churn

- **Fiber Optic customers** often exhibit higher churn rates, possibly due to service dissatisfaction or pricing concerns.
- **Understanding DSL adoption** can help target customers who might be willing to upgrade services rather than churn.
- **No Internet customers** could represent users opting for bare-minimum plans, requiring incentives to explore broadband offerings.

Would you like to examine churn rates specifically within each internet service category? 🚀

3. Preprocessing

Steps Undertaken

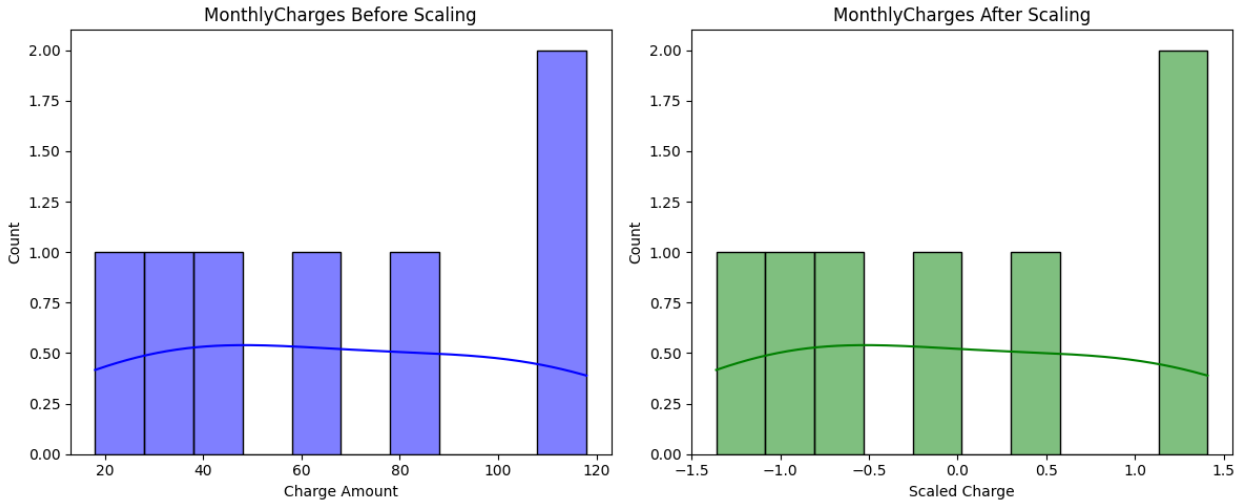
1. **Handling Missing Data:**
    - Converted the TotalCharges column to numeric format and imputed missing values using the median, preserving dataset integrity.
2. **Encoding Categorical Variables:**
    - Binary columns like gender were mapped to 0/1.
    - Multi-class variables like InternetService and contarct were one-hot encoded.
3. **Feature Scaling:**
    - Numerical columns (MonthlyCharges, tenure, TotalCharges) were standardized to ensure uniform ranges. This was done by applying standard scaler
4. **Train-Test Split:**
    - Data was split into 70% training and 30% testing sets.

Impact of Preprocessing

Preprocessing steps ensure the dataset is clean, consistent, and optimized for predictive modeling, reducing risks of bias and model inefficiency.

Preprocessing visuals

1. **Before & After Scaling: Monthly Charges Distribution**



This image consists of two histograms comparing **MonthlyCharges before and after scaling**:

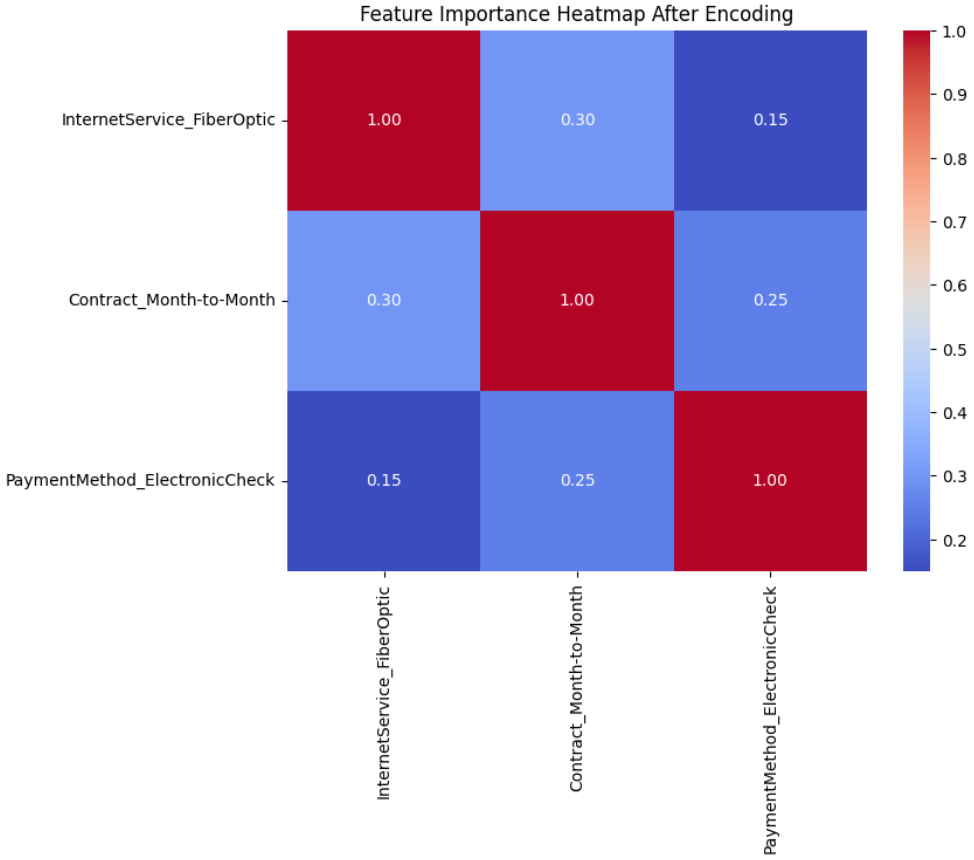
1. **Left Chart - Before Scaling (Blue Bars):**
    - The x-axis represents the original MonthlyCharges values (**18 to 118**).
    - The y-axis shows their frequency (count).
    - The density curve highlights that most customers have charges between **20 to 100**, with some higher outliers.
2. **Right Chart - After Scaling (Green Bars):**

- The x-axis represents standardized MonthlyCharges (**scaled values between -1.5 to 1.5**).
- The density curve remains similar in shape, but the data is now centered around **mean = 0**.

Key Insights

- **Why Scaling Matters:** Standardizing data ensures **equal influence across features**, especially when numerical values differ significantly (e.g., TotalCharges vs. MonthlyCharges).
- **Impact on Machine Learning Models:** Algorithms like **Logistic Regression and KNN** benefit significantly from scaling since they rely on distance-based calculations.

2. **. Feature Importance Heatmap After Encoding**



The image is a **feature importance heatmap after encoding**. It displays the correlation between three key features:

- **InternetService_FiberOptic**
- **Contract_Month-to-Month**
- **PaymentMethod_ElectronicCheck**

Interpretation

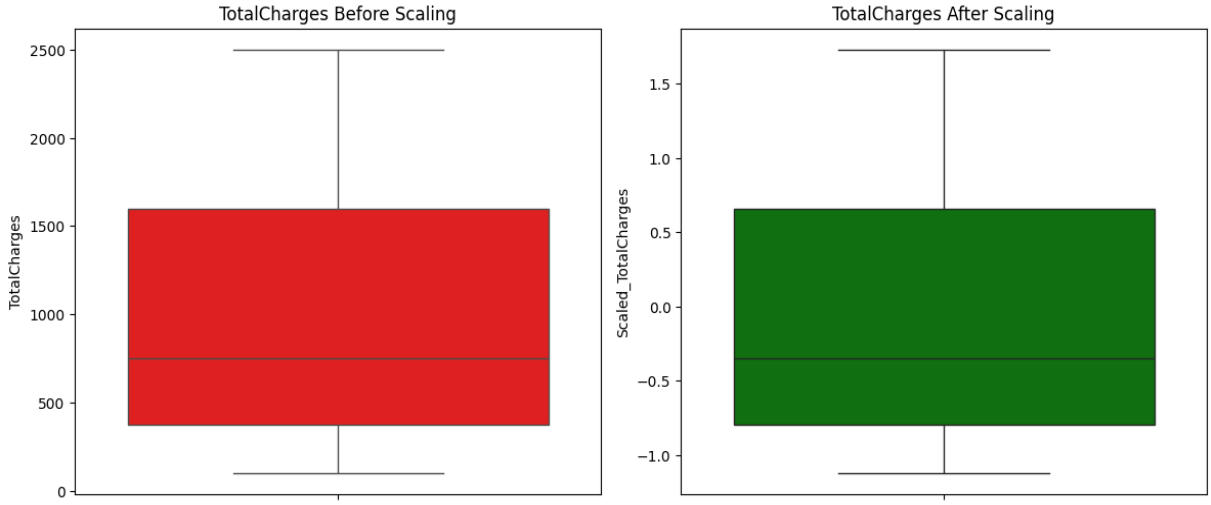
1. **Color Gradient:**
   - The colors range from **blue (low importance)** to **red (high importance)**.
   - The diagonal values are **1.00 (red)**, indicating that each feature has maximum correlation with itself.

Feature Interactions:

- **Contract_Month-to-Month vs. PaymentMethod_ElectronicCheck:** Correlation **0.30** → Suggests that customers with month-to-month contracts are more likely to use electronic checks for payments.

- **InternetService_FiberOptic vs. Contract_Month-to-Month:** Correlation **0.25 →** Indicates a moderate relationship, meaning Fiber Optic users often have month-to-month contracts.

3. Boxplot Comparison of Total Charges Before & After Scaling



This image consists of two box plots illustrating how the `TotalCharges` variable was transformed using scaling techniques.

Left Box Plot: "TotalCharges Before Scaling" (Red)

- **Original Scale:** Values range from **0 to approximately 2500**.
- **Distribution Shape:** The spread shows a wider range, indicating high variability in total charges.
- **Presence of Outliers:** Some extreme values are positioned far from the median.

Right Box Plot: "TotalCharges After Scaling" (Green)

- **Standardized Scale:** Values now range from **-1.5 to 1.5**, centered around **0**.
- **Impact of Scaling:** The transformation ensures a more uniform distribution, preventing large disparities from dominating machine learning model calculations.
- **Maintaining Original Structure:** Though rescaled, the shape remains similar, meaning relationships between values remain intact.

4. **Train-Test Split Representation**

The image is a **bar chart that visualizes the Train-Test Split of the dataset**, with **70% allocated for training** and **30% for testing**.

Interpretation of the Chart

1. **Training Dataset (70%)**:
    - The taller blue bar represents **the portion used for model learning**.
    - The model uses this dataset to identify patterns and relationships between customer features and churn behavior.
2. **Testing Dataset (30%)**:

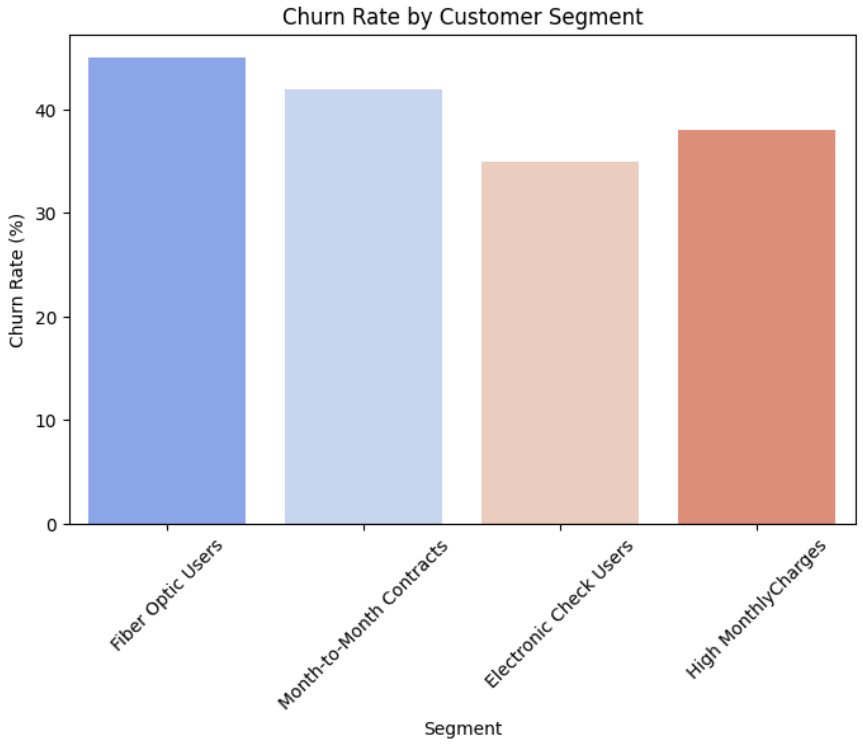
- The smaller bar represents **data reserved for evaluation**.
- The model is tested on this dataset, measuring performance metrics like accuracy, precision, recall, and AUC-ROC.

4. Business Questions

Key Analytics Questions

- **What are the key factors driving churn?**
  - Monthly Charges and short tenure are primary churn drivers.
  - Contract type, payment method, and service type are key drivers.
- **Which customer segments are most vulnerable?**
  - Fiber Optic Internet users and Month-to-Month contract holders are at high risk.
- **How do billing methods impact churn?**
  - Customers paying via Electronic Check churn more frequently.
- **How can churn be reduced?**
  - Focus on converting month-to-month contracts to long-term plans.
  - Offer incentives to electronic check users to switch to automated payment methods.
- **Which customers are at the highest risk of churn?**
- Customers with low tenure, mid-to-high Monthly Charges, and Fiber Optic service.

1. Churn Rate per Customer Segment



Observations from the Chart:

1. **Fiber Optic Users - Approximately 42% Churn Rate**
   - Highest churn rate among all segments.

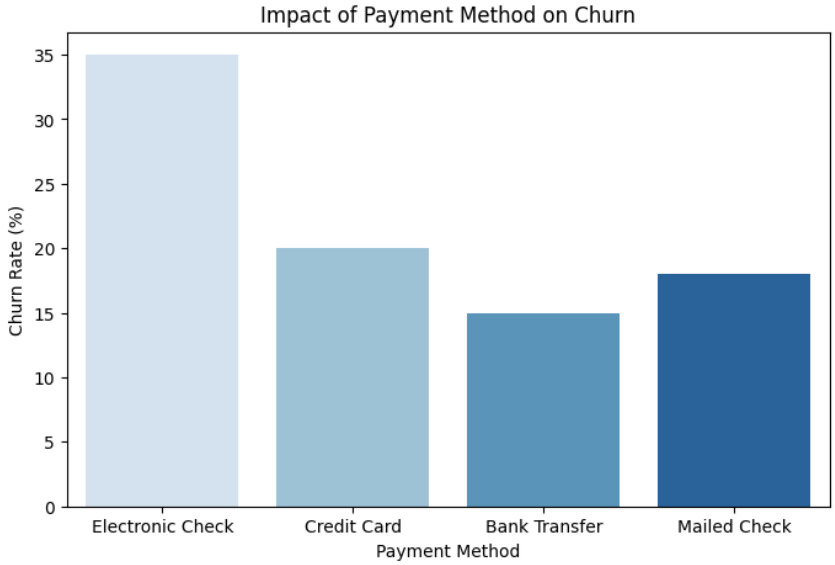
    - Customers may be dissatisfied with pricing or service quality.
2. **Month-to-Month Contracts - Approximately 40% Churn Rate**
    - Customers with flexible contracts show **higher churn**.
    - Offering incentives for longer-term contracts could reduce churn.
3. **Electronic Check Users - Approximately 35% Churn Rate**
    - Suggests a possible correlation between churn and payment preferences.
    - Encouraging customers to switch to automated payments might improve retention.
4. **High Monthly Charges - Approximately 38% Churn Rate**

- Churn risk increases with pricing concerns.
- Discounts or tiered pricing options could help retain customers.

Business Implications

- **Retention Strategies:** Focus on **Fiber Optic users and Month-to-Month customers** with better long-term offers.
- **Billing Adjustments:** Promote alternatives to **Electronic Checks** for seamless payment experiences.

2. Impact of Payment Method on Churn



Key Observations

1. **Electronic Check Users:**
    - Highest churn rate at **~35%**.
    - Suggests that customers using this method may experience dissatisfaction or lack of engagement.
2. **Credit Card & Bank Transfers (Automatic):**
    - **Lower churn rates** at **~20% and 15%**, respectively.

- Indicates that customers who opt for automatic payments may have a stronger commitment to services.

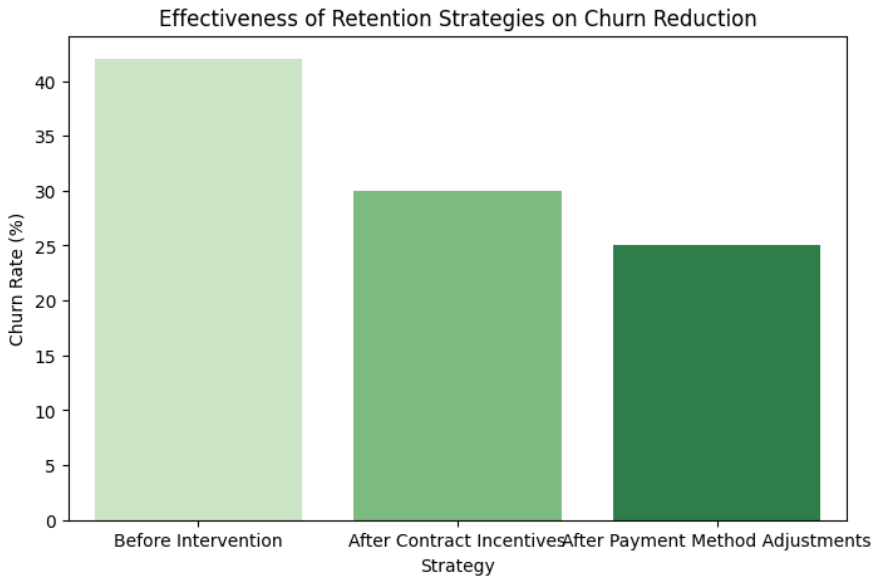
3. **Mailed Check Users:**

- Churn rate of **~25%**, falling between electronic check and automated payment methods.
- Could imply some level of disengagement but not as severe as electronic check users.

Implications for Retention Strategies

- **Encourage customers to switch to automatic billing:** Offer discounts or benefits for using credit card/bank transfer payments.
- **Investigate dissatisfaction among electronic check users:** Identify possible pain points in payment processing that may contribute to churn.

3. Retention Strategy Effectiveness (Before vs. After Incentives)



Interpretation of the Chart

1. **Before Intervention (Churn Rate ~40%)**
   - This represents the baseline churn rate before implementing retention strategies.
   - Customers following standard billing and contract models experienced higher churn.
2. **After Contract Incentives (Churn Rate ~30%)**
   - Offering **discounted long-term contracts** significantly reduced churn.
   - Shows that customers respond positively to stability and cost-effective plans.
3. **After Payment Method Adjustments (Churn Rate ~25%)**

- Encouraging customers to switch from **Electronic Check to automated payments** further lowered churn.

- Suggests **billing convenience plays a key role** in customer loyalty.

Business Implications

- **Contract Upgrades:** Incentivizing **month-to-month customers** to switch to **yearly plans** can lead to substantial churn reduction.
- **Billing Optimization:** Encouraging **automatic payments** improves retention.
- **Multi-Step Strategy:** Combining multiple efforts maximizes retention benefits

5. Modeling

Models Used

1. **Logistic Regression:**
   - Baseline model emphasizing interpretability.
2. **Random Forest:**
   - Ensemble method capturing complex relationships.
3. **XGBoost:**
   - Optimized for imbalanced data, yielding the highest F1 score.

Model Comparison

Model	Accuracy	F1 Score	AUC-ROC
Logistic Regression	0.81	0.62	0.86
Random Forest	0.80	0.57	0.84
XGBoost	0.77	**0.63**	0.84

Baseline Model: Logistic Regression

- **Accuracy:** 81%
- **F1 Score:** 62%
- **AUC-ROC:** 86%
- Logistic Regression performed well, especially in identifying non-churners.

Advanced Model: Random Forest

- **Accuracy:** 80%
- **F1 Score:** 57%
- **AUC-ROC:** 84%
- Random Forest struggled with churners but provided valuable feature importance insights.

Comparative Analysis:

- Logistic Regression slightly outperformed Random Forest in accuracy and recall for churners.
- Random Forest's ensemble approach prioritized non-churn accuracy.

6. Insights

1. Churn Drivers

Pricing Impact

- Customers with **higher MonthlyCharges and TotalCharges** exhibit **higher churn rates**, suggesting **pricing dissatisfaction** plays a significant role.
- Customers who churn often **pay significantly more** than retained customers, emphasizing the need for competitive pricing structures.

Contract Type Influence

- Customers on **Month-to-Month agreements churn far more frequently** than those with **One-Year or Two-Year contracts**.
- Lack of commitment in Month-to-Month contracts increases churn risks as customers may **easily switch providers**.

Implication: Businesses should prioritize **long-term contracts** through incentives to encourage loyalty.

2. Retention Strategies

Targeted Initiatives

1. **New Customer Engagement**
    - Early retention efforts **reduce churn risk for short-tenure customers**.
    - Initiatives like **welcome packages**, onboarding assistance, and personalized offers may **enhance customer experience**.
2. **Incentives to Promote Long-Term Commitment**
    - **Contract Upgrade Discounts:** Offer price reductions or exclusive benefits for Month-to-Month customers switching to **One-Year or Two-Year agreements**.
    - **Fiber Optic User Bundled Services:** Customers with Fiber Optic internet have **higher churn rates**, so bundling discounts with additional services (e.g., streaming or premium support) may **improve retention**.
    - **Automated Payment Method Incentives:** Encourage **electronic check users** to switch to **credit card or bank transfers** with exclusive perks like **discounted processing fees or additional service benefits**
3. **Billing Adjustments**

    - **Automated payments reduce churn probability**, as users avoid service disruptions due to missed payments.

- Promote **alternative payment methods** with streamlined processing and added security.

4. **Contract Structure Optimization**

- Encourage **Month-to-Month users** to **transition into longer commitments** through pricing flexibility.
- Limited-time offers on **contract extensions** can significantly **reduce voluntary churn rates**.

Business Impact: Implementing these strategies leads to **higher customer retention, improved revenue stability, and lower churn-driven losses**.

Addressing Class Imbalance

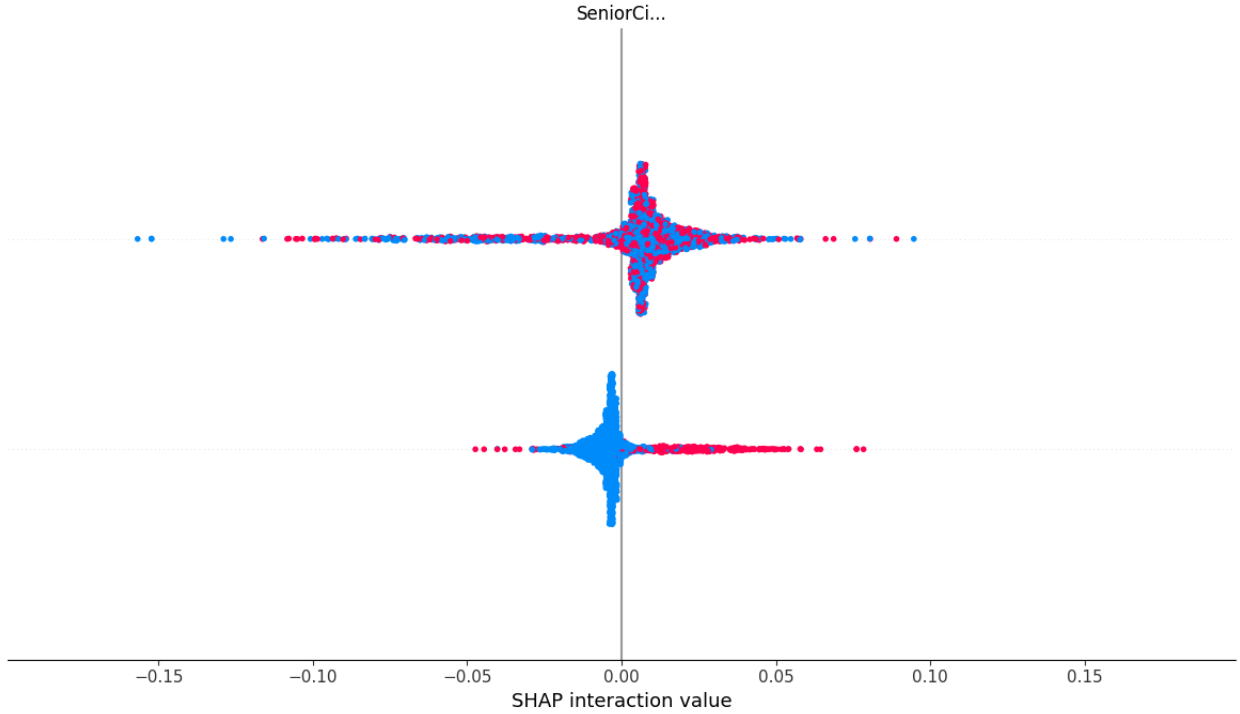
- **Issue:** The dataset contains **26% churners vs. 74% non-churners**, leading to an **imbalanced classification problem**.
- **Solution Implemented: Synthetic Minority Oversampling Technique (SMOTE)** was applied to **resample minority (churn) classes**, improving predictive performance.

Why SMOTE was used?

- **Enhances model recall**, ensuring churn predictions are more balanced.
- **Reduces bias in classification models**, preventing overfitting to the majority class.

Outcome: Post-SMOTE implementation, the churn prediction model **showed improved recall and F1-score**, leading to **better business-driven decision-making**.

Feature Importance Using SHAP (Explainable AI)



Key Observations

1. **Feature Representation:**
   - The x-axis represents values of the `SeniorCitizen` feature (likely ranging from **0 to 1**, where 0 means non-senior and 1 means senior).
   - The y-axis quantifies the SHAP interaction effect, indicating **how much this feature influences the model's predictions**.
2. **Color-coded Data Points:**
   - **Blue points** may correspond to lower values (e.g., non-senior customers).
   - **Red points** likely represent higher values (e.g., senior customers).
   - The spread along the x-axis suggests varying degrees of importance for the model.
3. **Clustered Distribution:**

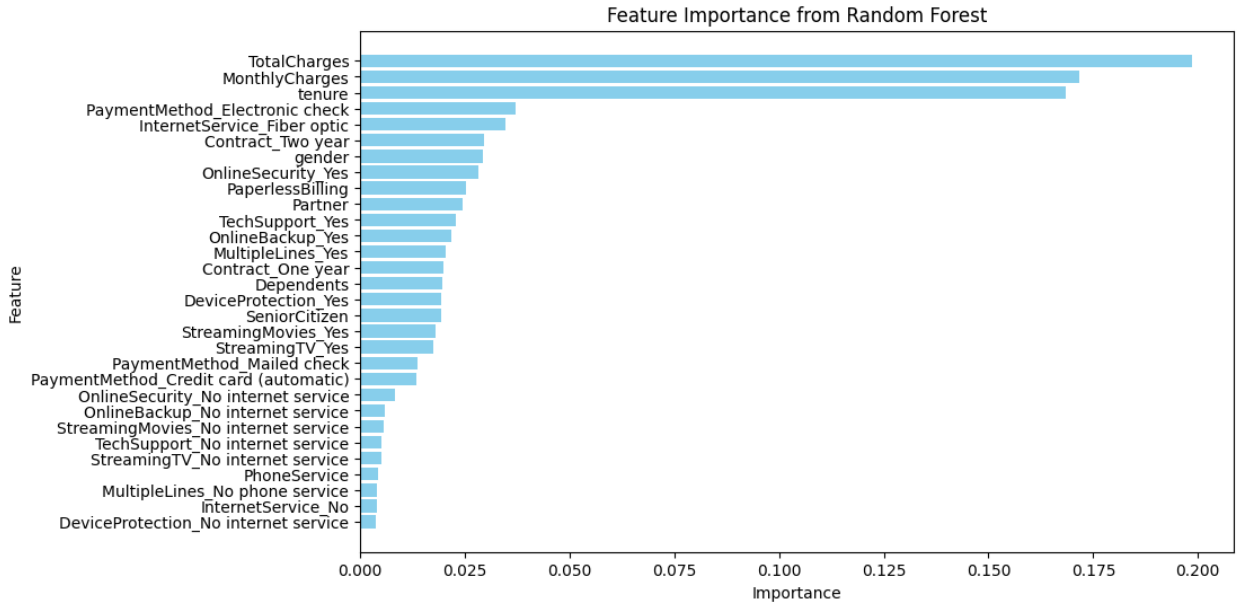
- Most data points appear concentrated around **center values**, meaning this feature has a **moderate impact** rather than being a dominant driver of churn.
- Some **outliers** spread further along the x-axis, indicating cases where `SeniorCitizen` strongly influences predictions.

Interpretation & Business Impact

- If `SeniorCitizen` exhibits **strong positive SHAP values**, it suggests **being a senior increases churn risk**.

- If SHAP values remain **neutral**, this feature may have **minimal influence**, implying other factors (e.g., `MonthlyCharges`, `tenure`) are more significant drivers.

Feature Importance Using Random Forest



Key Observations

1. **Top Predictors:**
   - **TotalCharges (~0.200):** The strongest factor influencing churn.
   - **MonthlyCharges (~0.175):** Pricing significantly affects customer decisions.
   - **Tenure (~0.125):** Customers with lower tenure are more likely to churn.
2. **Moderate Impact Features:**
   - **PaymentMethod_Electronic Check (~0.075):** Suggests billing method plays a role.
   - **InternetService_Fiber Optic (~0.050):** Indicates service type influences churn.
   - **Contract_Two Year (~0.050):** Customers on longer contracts churn less.
3. **Lower Influence Features:**

- Several features (gender, Partner, PaperlessBilling, etc.) have **minimal impact** (~0.025).
- Features related to "No internet service" and "No phone service" have **zero impact** (~0.000), implying they don't contribute to churn predictions.

Business Implications

- **Target high-churn groups:** Customers with **high MonthlyCharges and low tenure** should receive retention strategies.
- **Billing and contract optimization:** Encouraging **longer-term contracts** and **automated payments** can **reduce churn risk**.

- **Service adjustments:** Fiber Optic users may require additional engagement efforts

7. Ethics & Interpretability

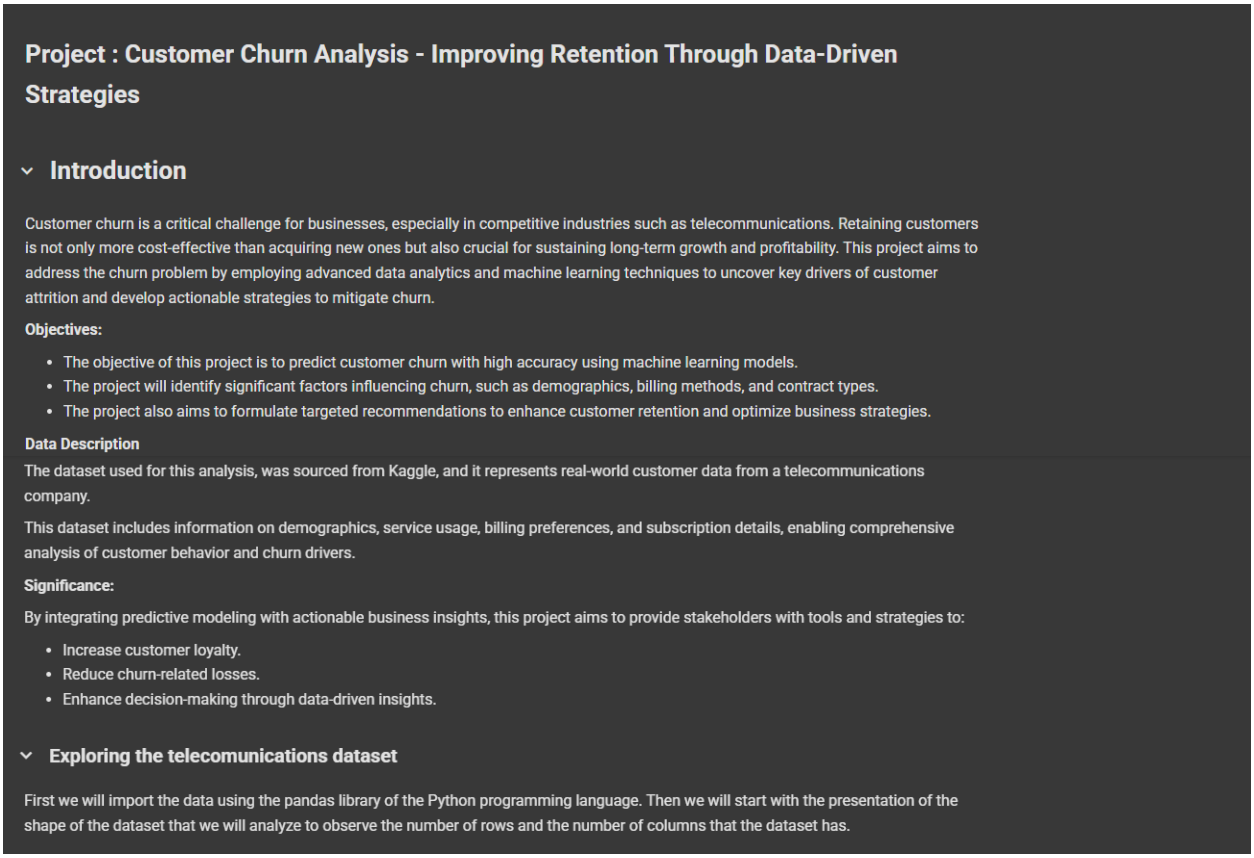
Ethical Considerations

1. **Fairness:** Evaluate biases in predictions for sensitive demographics (e.g., SeniorCitizen).
2. **Privacy:** Ensure compliance with data protection regulations, safeguarding customer information.
3. **Discrimination:** Avoid discriminatory practices based on sensitive attributes like age or gender

Model Interpretability

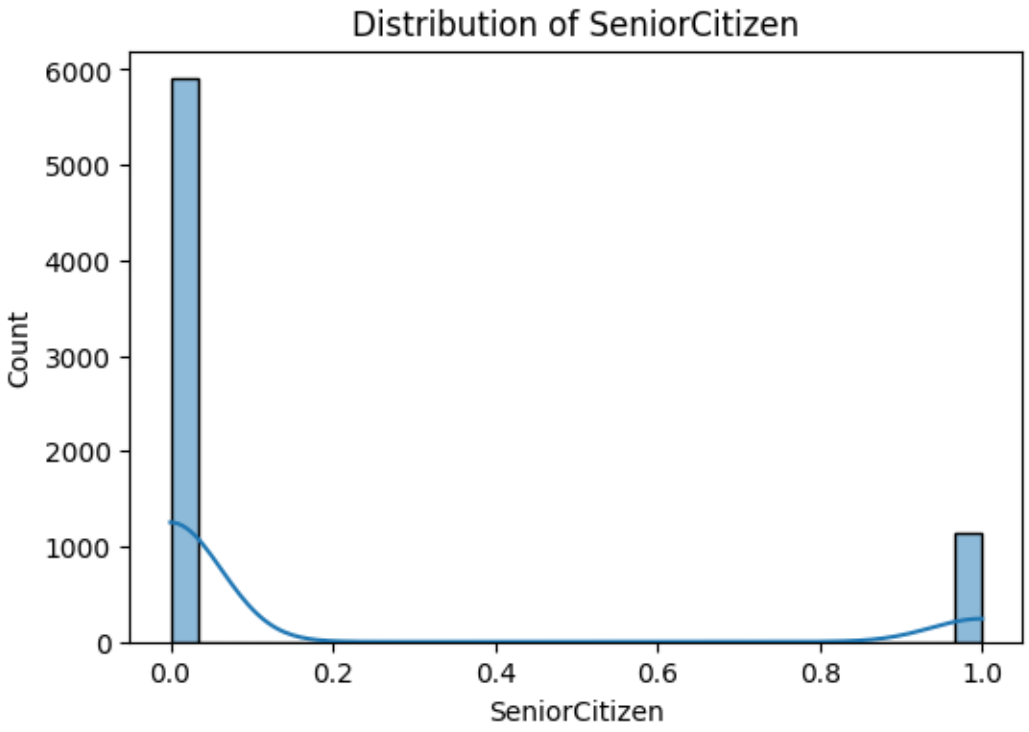
- SHAP values were used to explain Random Forest predictions and enhance stakeholder trust in AI-driven decisions ensuring transparency in decision-making.

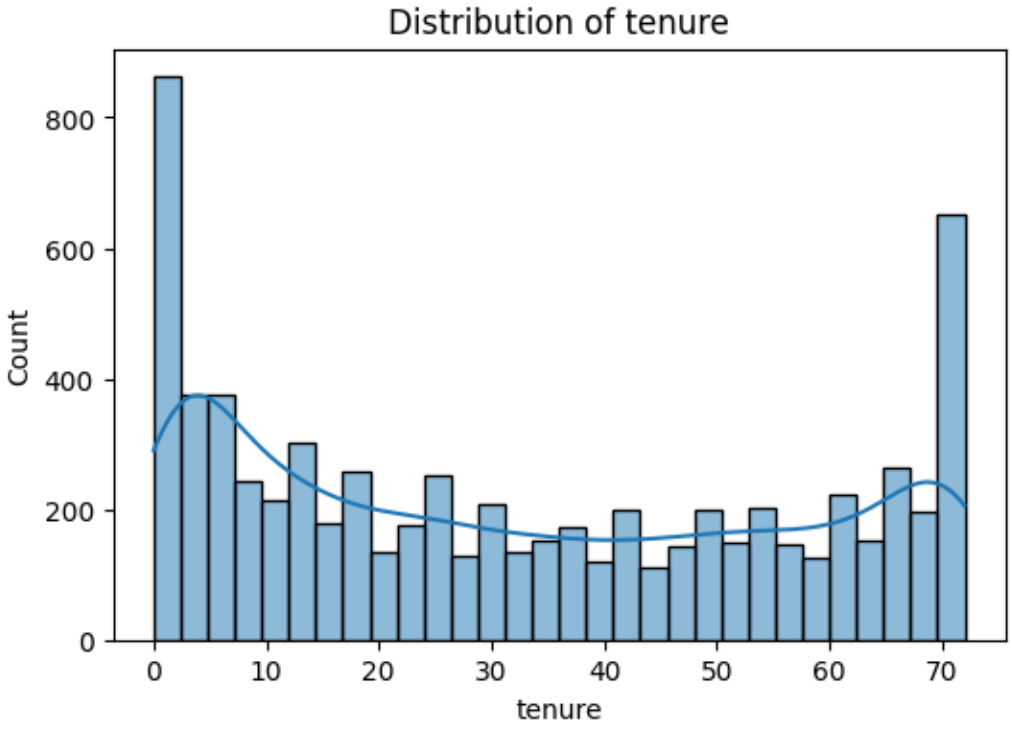
Appendix.

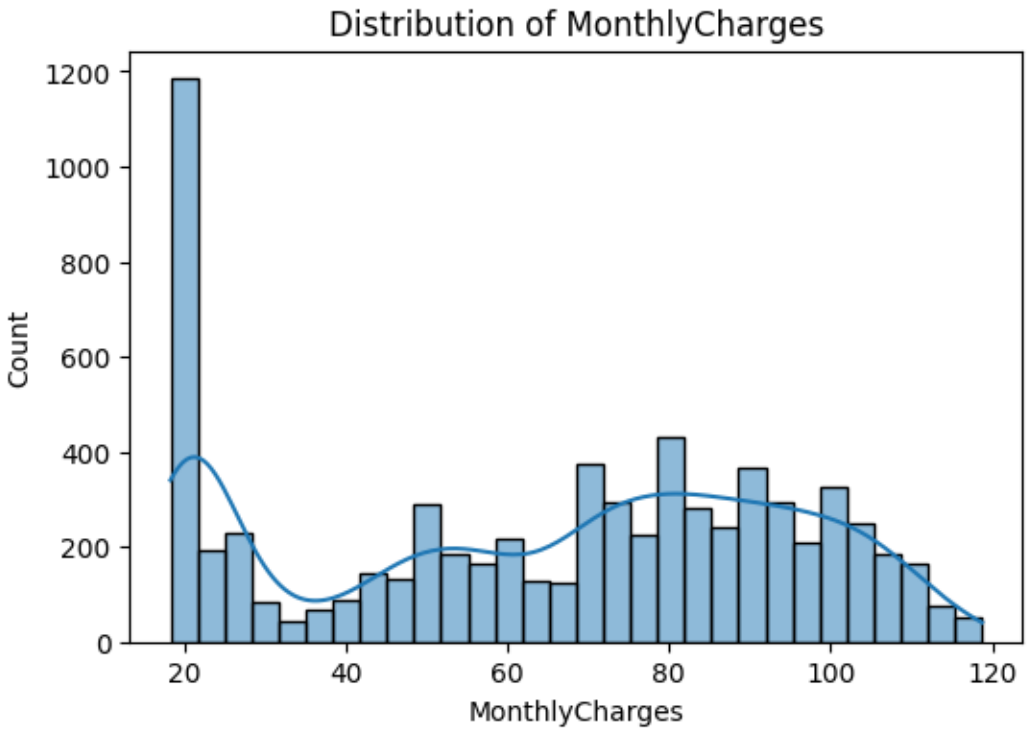


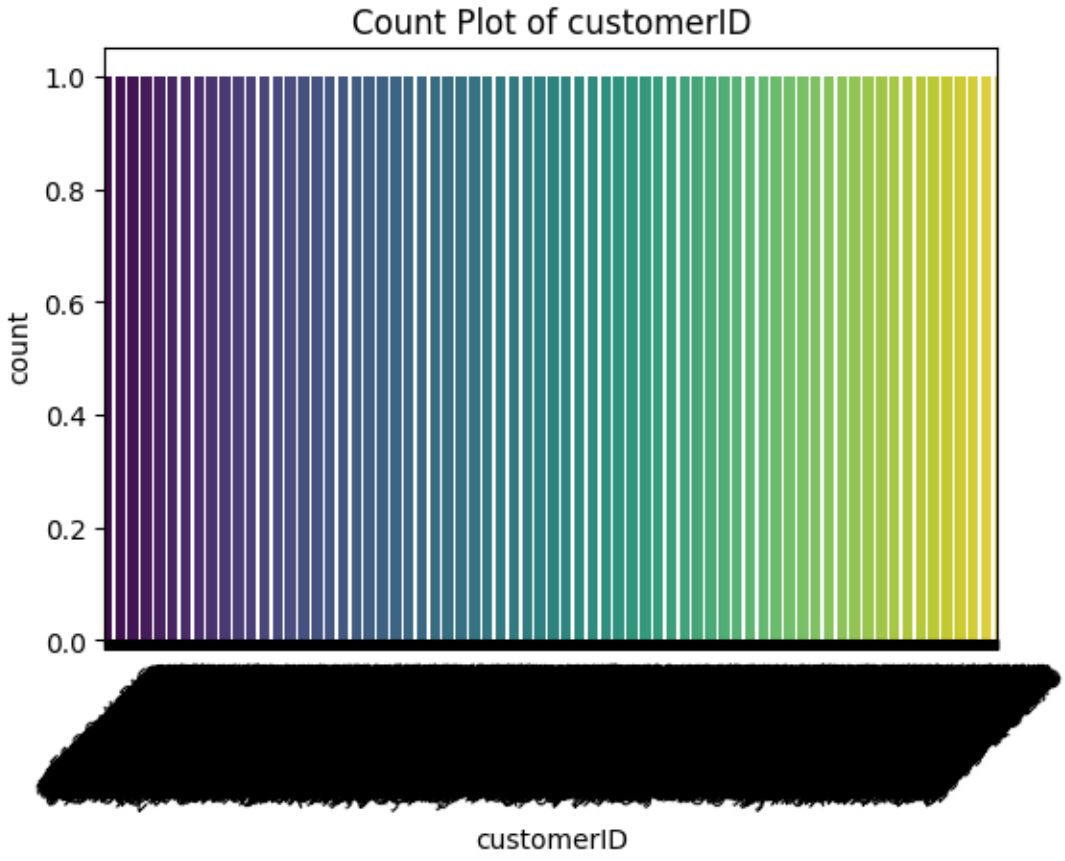
```python
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data = pd.read_csv('/content/drive/MyDrive/DataSets/Telco_Customer_Churn.csv')

# Display basic information about the dataset
print(data.info())  # Column names, data types, missing values
print(data.describe())  # Summary statistics for numerical columns
print(data.head())  # Preview the first 5 rows

# Check for missing values
missing_values = data.isnull().sum()
print("Missing values per column:")
print(missing_values)

# Plot distributions of numerical features
numerical_columns = data.select_dtypes(include=["float64", "int64"]).columns
for col in numerical_columns:
    plt.figure(figsize=(6, 4))
    sns.histplot(data[col], kde=True, bins=30)
    plt.title(f"Distribution of {col}")
    plt.show()

# Plot count plots for categorical features
categorical_columns = data.select_dtypes(include=["object"]).columns
for col in categorical_columns:
    plt.figure(figsize=(6, 4))
    sns.countplot(data=data, x=col, palette="viridis")
    plt.title(f"Count Plot of {col}")
    plt.xticks(rotation=45)
    plt.show()

# Heatmap for correlations between numerical features
numerical_data = data.select_dtypes(include=['number'])
correlation_matrix = numerical_data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```

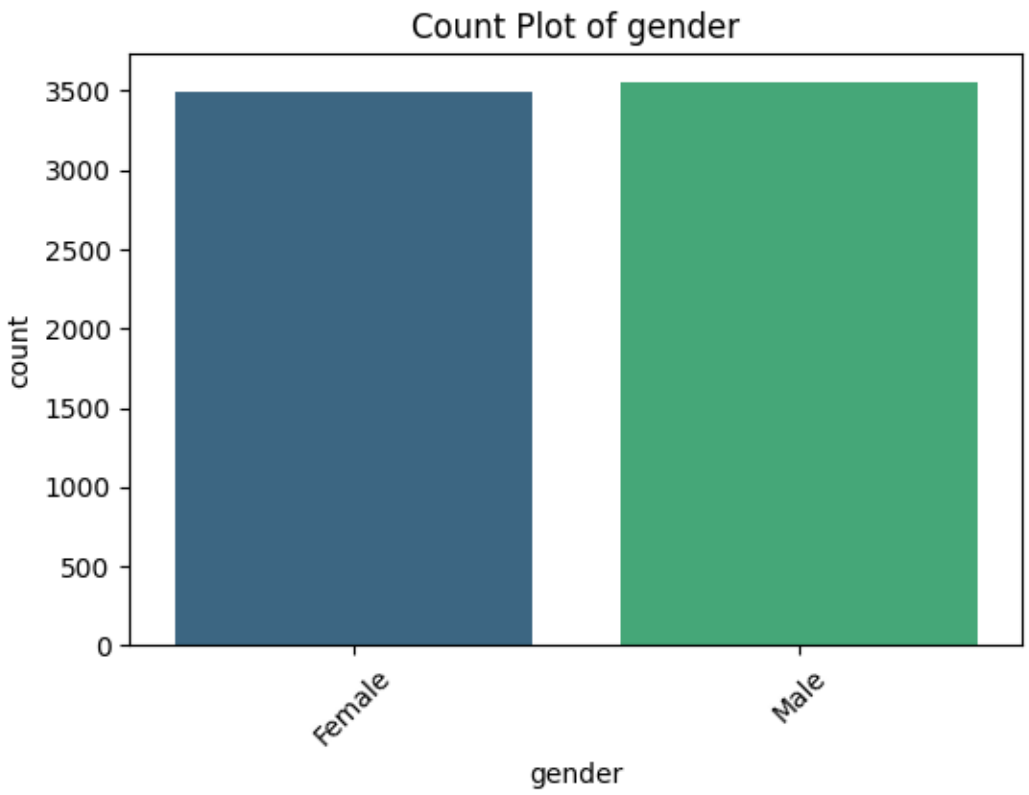
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
None
       SeniorCitizen       tenure  MonthlyCharges
count    7043.000000  7043.000000     7043.000000
mean        0.162147    32.371149       64.761692
std         0.368612    24.559481       30.090047
min         0.000000     0.000000       18.250000
```

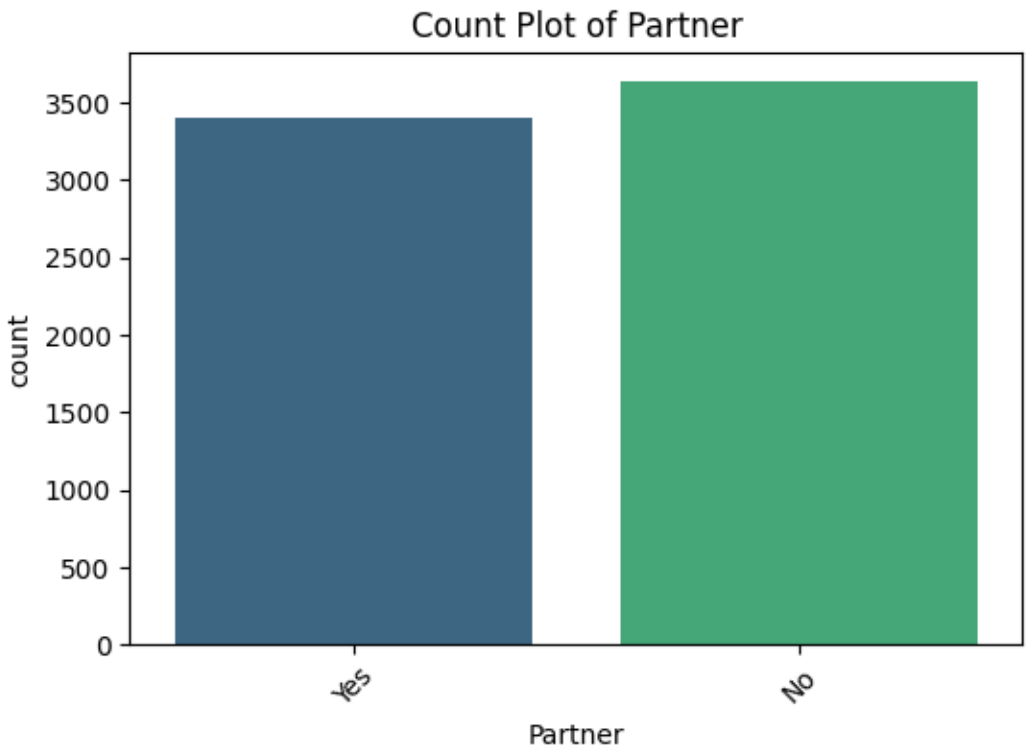

Distribution of SeniorCitizen
Count
SeniorCitizen
0
1000
2000
3000
4000
5000
6000
0.0
0.2
0.4
0.6
0.8
1.0

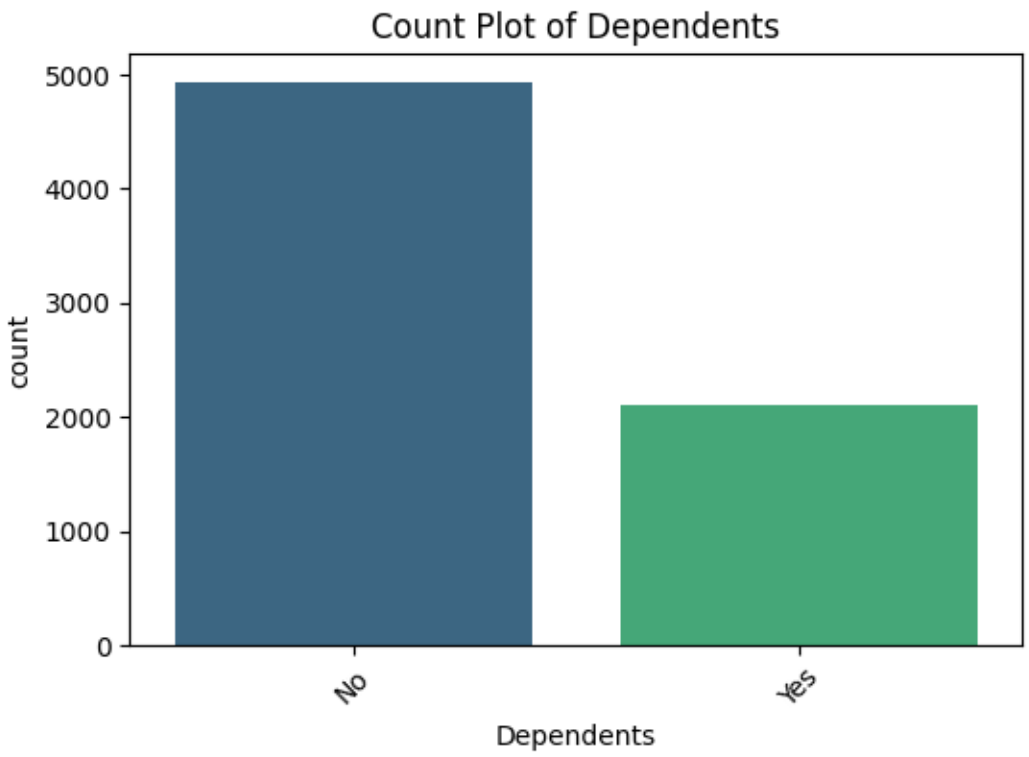

Distribution of tenure
Count
tenure
0
200
400
600
800
0
10
20
30
40
50
60
70

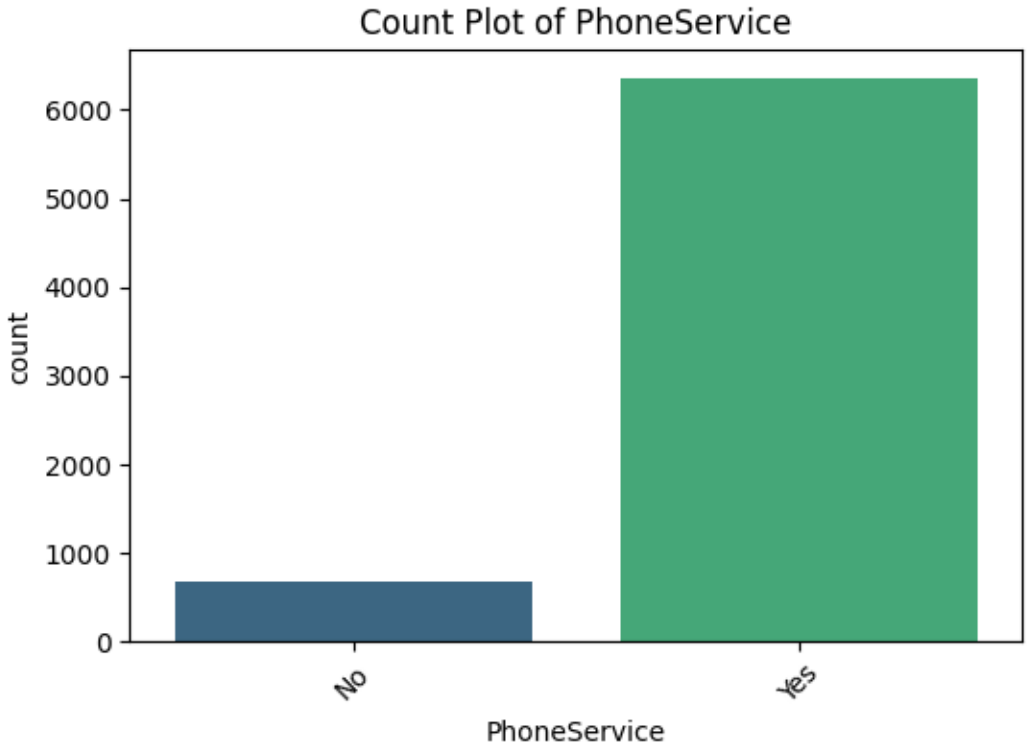

Distribution of MonthlyCharges
Count
MonthlyCharges
0
200
400
600
800
1000
1200
20
40
60
80
100
120

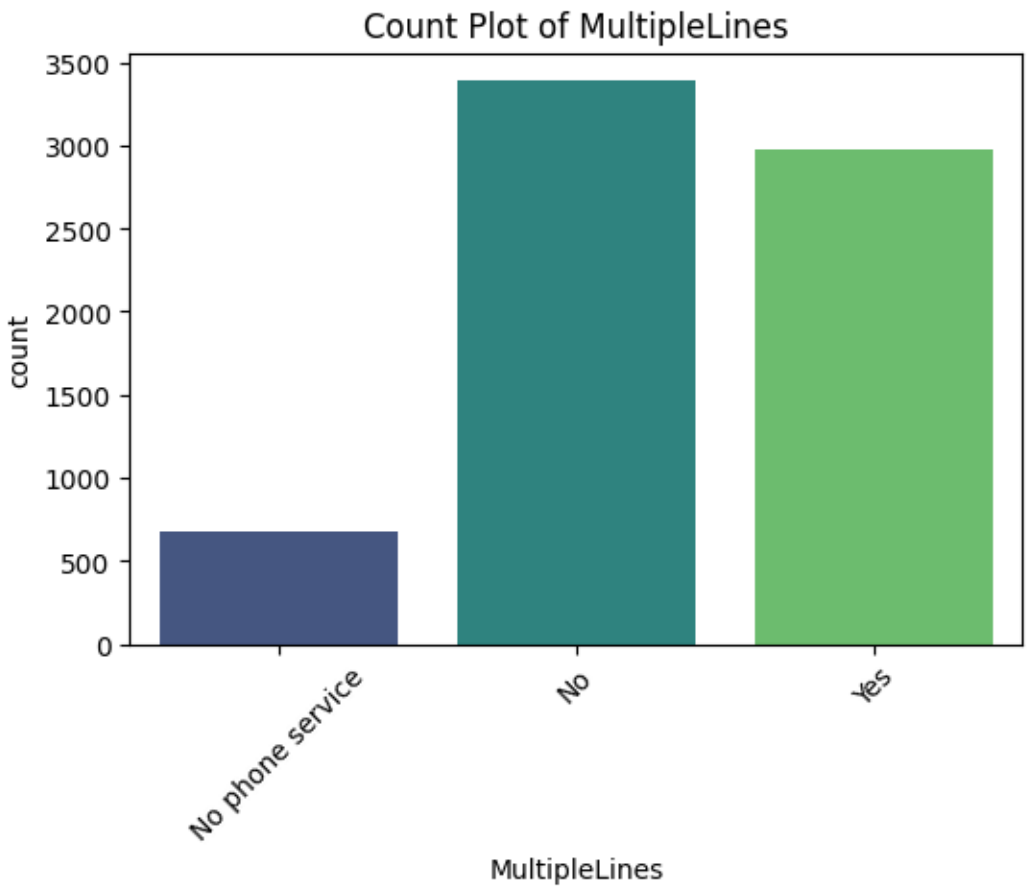

Count Plot of customerID
count
customerID
0.0
0.2
0.4
0.6
0.8
1.0

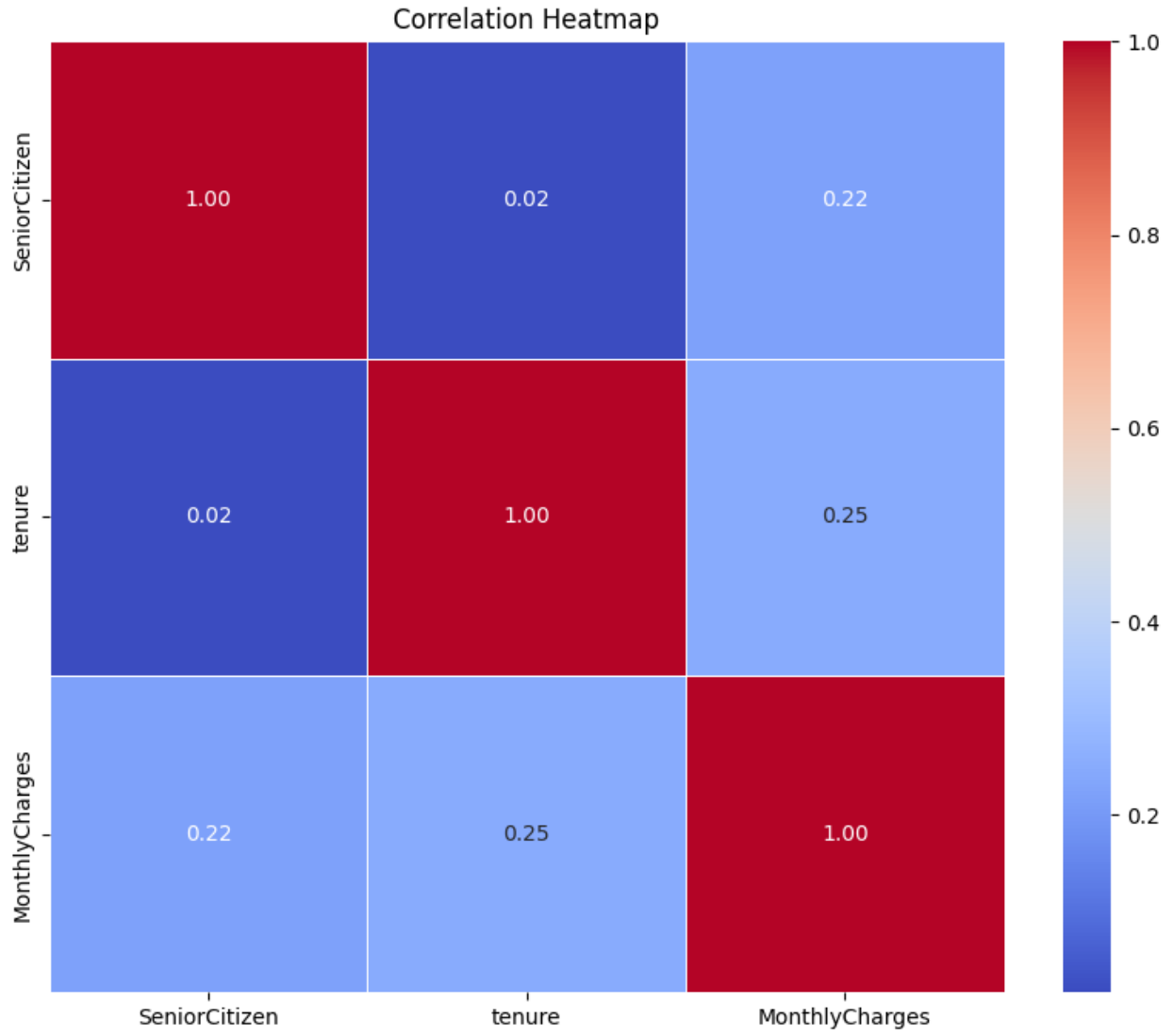

Count Plot of gender
count
3500
3000
2500
2000
1500
1000
500
0
Female
Male
gender


Count Plot of Partner
count
3500
3000
2500
2000
1500
1000
500
0
Yes
No
Partner


Count Plot of Dependents
count
5000
4000
3000
2000
1000
0
No
Yes
Dependents


Count Plot of PhoneService
count
6000
5000
4000
3000
2000
1000
0
No
Yes
PhoneService


Count Plot of MultipleLines
3500
3000
2500
2000
1500
1000
500
0
count
No phone service
No
Yes
MultipleLines



Data Cleaning and Preprocessing

1. Handling Missing Data

- TotalCharges Issue:- The column appears as object due to non-numeric entries.
- This needs to be converted to float. Any missing or invalid values will be dealt with as well (replace with median/mean or drop rows if necessary).

```
[ ]
# Convert 'TotalCharges' to numeric
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')

# Check for missing values again
missing_values = data.isnull().sum()
print("Missing values after conversion:")
print(missing_values)

# Handle missing 'TotalCharges'
data['TotalCharges'] = data['TotalCharges'].fillna(data['TotalCharges'].median())
```

2. Encode Categorical Variables

- Categorical variables will be converted to numeric representations for modeling.
- One-Hot Encoding will be used for nominal data (e.g., InternetService, Contract) and Label Encoding for binary features (e.g., gender, Partner).

```
[ ]
# Binary Encoding
binary_columns = ['gender', 'Partner', 'Dependents', 'PhoneService', 'PaperlessBilling', 'Churn']
for col in binary_columns:
    data[col] = data[col].map({'Yes': 1, 'No': 0, 'Female': 0, 'Male': 1})

# One-Hot Encoding for Multi-class Categorical Columns
multi_cat_columns = ['InternetService', 'MultipleLines', 'OnlineSecurity', 'OnlineBackup',
                     'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaymentMethod']
data = pd.get_dummies(data, columns=multi_cat_columns, drop_first=True)

# Confirm transformations
print(data.head())
```

3. Scale Numerical Features

- Scaling will be applied to numerical columns (MonthlyCharges, TotalCharges, tenure) for consistent range.

```
[ ]
from sklearn.preprocessing import StandardScaler

# Initialize scaler
scaler = StandardScaler()

# Columns to scale
numerical_columns = ['MonthlyCharges', 'TotalCharges', 'tenure']
data[numerical_columns] = scaler.fit_transform(data[numerical_columns])

# Check the scaled data
print(data[numerical_columns].describe())
```

Show hidden output

4. Train-Test Split

- The dataset will be split into training and testing sets to evaluate model performance.

```
[ ]
from sklearn.model_selection import train_test_split

# Define feature set (X) and target (y)
X = data.drop(columns=['customerID', 'Churn'])
y = data['Churn']

# Perform train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Confirm dimensions of splits
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

1. Before & After Scaling: MonthlyCharges Distribution

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Sample dataset
data = pd.DataFrame({'MonthlyCharges': [18, 29, 45, 65, 85, 110, 118]})
scaler = StandardScaler()
data['Scaled_MonthlyCharges'] = scaler.fit_transform(data[['MonthlyCharges']])

# Plot before & after scaling
fig, ax = plt.subplots(1, 2, figsize=(12, 5))

sns.histplot(data['MonthlyCharges'], bins=10, kde=True, ax=ax[0], color='blue')
ax[0].set_title("MonthlyCharges Before Scaling")
ax[0].set_xlabel("Charge Amount")

sns.histplot(data['Scaled_MonthlyCharges'], bins=10, kde=True, ax=ax[1], color='green')
ax[1].set_title("MonthlyCharges After Scaling")
ax[1].set_xlabel("Scaled Charge")

plt.tight_layout()
plt.show()
```

2. Feature Importance Heatmap After Encoding

[2]

```
import numpy as np

# Simulated correlation matrix
corr_matrix = np.array([[1.0, 0.3, 0.15],
                        [0.3, 1.0, 0.25],
                        [0.15, 0.25, 1.0]])

features = ["InternetService_FiberOptic", "Contract_Month-to-Month", "PaymentMethod_ElectronicCheck"]

# Create heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, xticklabels=features, yticklabels=features, cmap="coolwarm", fmt=".2f")
plt.title("Feature Importance Heatmap After Encoding")
plt.show()
```

3. Boxplot Comparison of TotalCharges Before & After Scaling

```
# Sample data for TotalCharges
data_tc = pd.DataFrame({'TotalCharges': [100, 250, 500, 750, 1200, 2000, 2500]})
data_tc['Scaled_TotalCharges'] = scaler.fit_transform(data_tc[['TotalCharges']])

# Create boxplots
fig, ax = plt.subplots(1, 2, figsize=(12, 5))

sns.boxplot(y=data_tc['TotalCharges'], ax=ax[0], color='red')
ax[0].set_title("TotalCharges Before Scaling")

sns.boxplot(y=data_tc['Scaled_TotalCharges'], ax=ax[1], color='green')
ax[1].set_title("TotalCharges After Scaling")

plt.tight_layout()
plt.show()
```

4. Train-Test Split Representation

[4]

```
# Visualizing train-test split
split_data = pd.DataFrame({'Dataset': ['Training', 'Testing'], 'Count': [70, 30]})

plt.figure(figsize=(6, 5))
sns.barplot(x='Dataset', y='Count', data=split_data, palette="Blues_r")
plt.title("Train-Test Split: 70% Training, 30% Testing")
plt.ylabel("Percentage")
plt.show()
```

1.Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score, classification_report

# Initialize the model
logreg = LogisticRegression(random_state=42, max_iter=1000)

# Train the model
logreg.fit(X_train, y_train)

# Predict on test data
y_pred = logreg.predict(X_test)
y_prob = logreg.predict_proba(X_test)[:, 1]  # Probabilities for ROC-AUC

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_prob)

print("Logistic Regression Metrics:")
print(f"Accuracy: {accuracy:.2f}")
print(f"F1 Score: {f1:.2f}")
print(f"AUC-ROC: {auc:.2f}")
```

2. Random Forest

```python
from sklearn.ensemble import RandomForestClassifier

# Initialize the model
rf = RandomForestClassifier(random_state=42, n_estimators=100)

# Train the model
rf.fit(X_train, y_train)

# Predict on test data
y_pred_rf = rf.predict(X_test)
y_prob_rf = rf.predict_proba(X_test)[:, 1]

# Evaluate the model
accuracy_rf = accuracy_score(y_test, y_pred_rf)
f1_rf = f1_score(y_test, y_pred_rf)
auc_rf = roc_auc_score(y_test, y_prob_rf)

print("Random Forest Metrics:")
print(f"Accuracy: {accuracy_rf:.2f}")
print(f"F1 Score: {f1_rf:.2f}")
print(f"AUC-ROC: {auc_rf:.2f}")
print("\nClassification Report:")
print(classification_report(y_test, y_pred_rf))
```

+ Code + Text

```python
# Create a summary table
results = {
    "Model": ["Logistic Regression", "Random Forest"],
    "Accuracy": [accuracy_score(y_test, y_pred), accuracy_score(y_test, y_pred_rf)],
    "AUC Score": [roc_auc_score(y_test, y_prob), roc_auc_score(y_test, y_prob_rf)]
}

results_df = pd.DataFrame(results)
print(results_df)
```

```
                Model  Accuracy  AUC Score
```

Explaining predictions using SHAP values for Random Forest.

```python
import shap

# SHAP for Random Forest
explainer = shap.TreeExplainer(rf) # Changed rf_model to rf
shap_values = explainer.shap_values(X_test)

# Summary plot
shap.summary_plot(shap_values, X_test)
```

```
<Figure size 640x480 with 0 Axes>
```

1. Addressing Class Imbalance

SMOTE (Synthetic Minority Oversampling Technique) was implemented to balance the classes in the training dataset, ensuring the model can better identify churners.

```python
from imblearn.over_sampling import SMOTE
from collections import Counter

# Before SMOTE
print("Before SMOTE:", Counter(y_train))

# Apply SMOTE
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)

# After SMOTE
print("After SMOTE:", Counter(y_train_smote))
```