VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF ECONOMICS AND LAW

FINAL PROJECT

CASE 2: CUSTOMER RETENTION

Subject:	Data mining
Course:	241CN0601
Instructor:	Master. Phan Huy Tam
Student Name:	Nguyen Lam Xuan Han
Student ID:	K224141662

TABLE OF CONTENT

CHAPTER 1: OVERVIEW

1.1. General Overview

In the competitive telecom industry, retaining customers is vital for sustainable growth, as acquiring new customers is significantly costlier. However, the abundance of attractive offers from competitors enables customers to switch providers easily, challenging companies to maintain loyalty.

Customer churn, defined as the percentage of customers discontinuing services within a given period, poses a major threat to revenue and market share. Understanding the reasons behind churn and predicting potential risks are critical for developing effective retention strategies.

Predictive analytics and machine learning have become essential tools in tackling churn. By analyzing historical data, telecom companies can identify patterns, forecast churn probabilities, and take timely action. Techniques such as Logistic Regression and Decision Trees have demonstrated their effectiveness in understanding customer behavior and improving retention efforts.

1.2. Objectives of the Study

The main objective of this study is to develop a predictive model to identify customers at high risk of churning, enabling telecom companies to create focused retention strategies.

Specific objectives include:

- Analyzing datasets to reveal key factors influencing churn.
- Addressing data issues like missing values and outliers.
- Building and evaluating predictive models using Logistic Regression, KNN, and Random Forest.
- Providing actionable insights to optimize customer retention strategies.

This study showcases the practical application of machine learning to real-world problems, offering a framework to reduce churn and enhance customer satisfaction and profitability.

CHAPTER 2: DATASET DESCRIPTION AND PROBLEM DEFINITION

2.1 Introduction of this dataset

This dataset captures detailed information about customers of a telecom company, with a focus on understanding customer churn. "Churn" refers to customers leaving the service, which is critical for businesses to monitor and predict.

2.2. Nature of this case

This case focuses on analyzing the factors influencing churn decisions, such as customer demographics, service details, and billing information, to build predictive models. By leveraging this dataset, the company can identify high-risk customers, understand their behavior, and implement targeted retention strategies.

The key aspect of this case is to predict customer churn using historical data to proactively identify high-risk customers and take retention measures. The objective is to reduce churn rates by analyzing the key factors influencing customer decisions and leveraging predictive modeling techniques. By doing so, the company can enhance customer satisfaction, improve service offerings, and minimize revenue losses caused by customer attrition.

2.3 Problem Definition:

The telecom company is facing a challenge with customer churn, where customers discontinue their services. Since retaining customers is more cost-effective than acquiring new ones, reducing churn is critical for the company's profitability and growth. The company needs to build a predictive model to classify customers based on their likelihood of churning.

The goal is to analyze customer data, including demographics, service usage patterns, billing details, and contract information, to predict the **Churn** variable (Yes/No). This model will help the company identify high-risk customers and understand the factors driving churn, enabling the design of targeted retention strategies and tailored service offerings to improve customer satisfaction.

This problem is a **binary classification task** in supervised learning. By solving it, the company can not only reduce churn but also optimize marketing and service efforts, leading to enhanced customer loyalty and improved business performance.

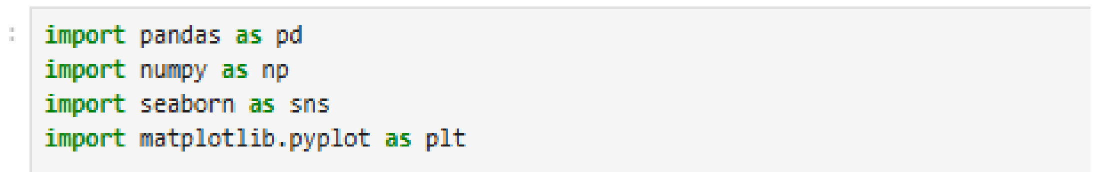
2.4 Some assumption from this case and dataset

Examining how the data features included in the dataset influence customer churn:

- Age: Older customers are less likely to churn due to the perceived effort required to switch providers and adapt to new technologies.
- Cost: Higher monthly costs correlate with increased churn likelihood, as customers become more cost-sensitive and explore cheaper alternatives.
- Contract term: Customers on long-term contracts are less likely to churn as they face fewer opportunities to reconsider or terminate their agreements, enjoying stability and convenience.
- Contract Type: Simple contracts with fewer product components are more prone to churn compared to bundled or complex contracts. Bundles with additional services (e.g., security packages) create dependencies that increase customer retention.
- Family: Customers with families, particularly those with children, have lower churn rates as they seek service continuity to support household needs like education and entertainment.
- Tenure: Longer tenure with the company is associated with lower churn, as customers develop loyalty and familiarity with the service.

CHAPTER 3: DATA PROCESSING

3.1 Declare the library:

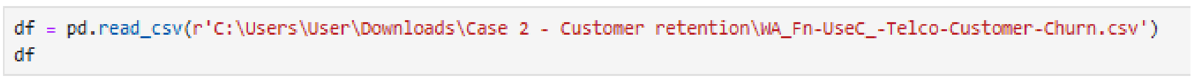


```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

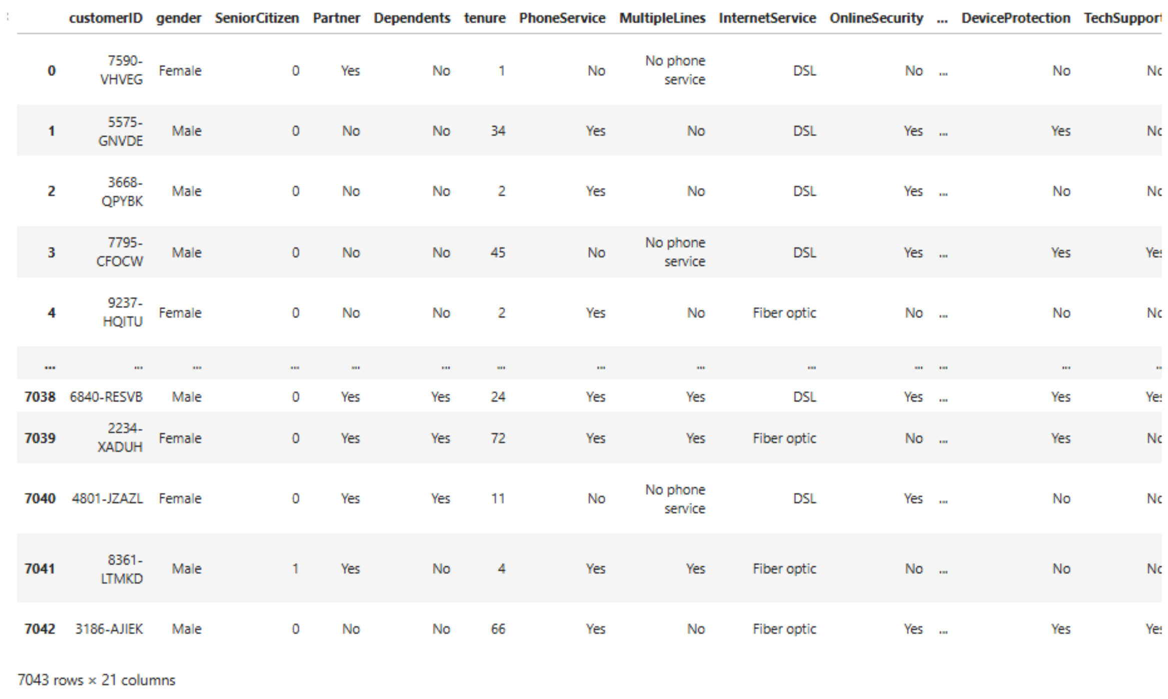
Sets up a Python environment for data analysis and visualization, making it ready to handle data manipulation with Pandas and NumPy and visualize data with Seaborn and Matplotlib.

3.2 Exploratory Data Analysis

Reading the data in csv file



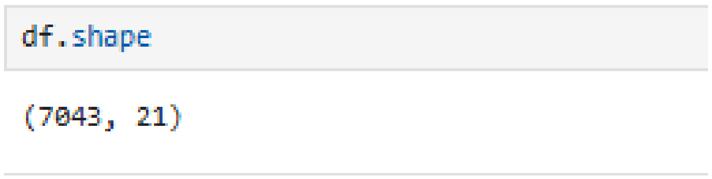
```python
df = pd.read_csv(r'C:\Users\User\Downloads\Case 2 - Customer retention\WA_Fn-UseC_-Telco-Customer-Churn.csv')
df
```



	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSuppor
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	N
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	N
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	N
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Ye
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	N
...	...	...	...	...	...	...	...	...	...	...	...	...	.
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	Ye
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	N
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	N
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	N
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	Ye

7043 rows × 21 columns

Understanding the size of this dataset



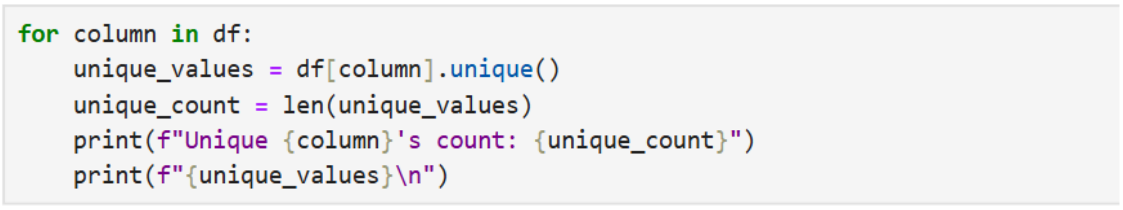
```python
df.shape
```

```
(7043, 21)
```

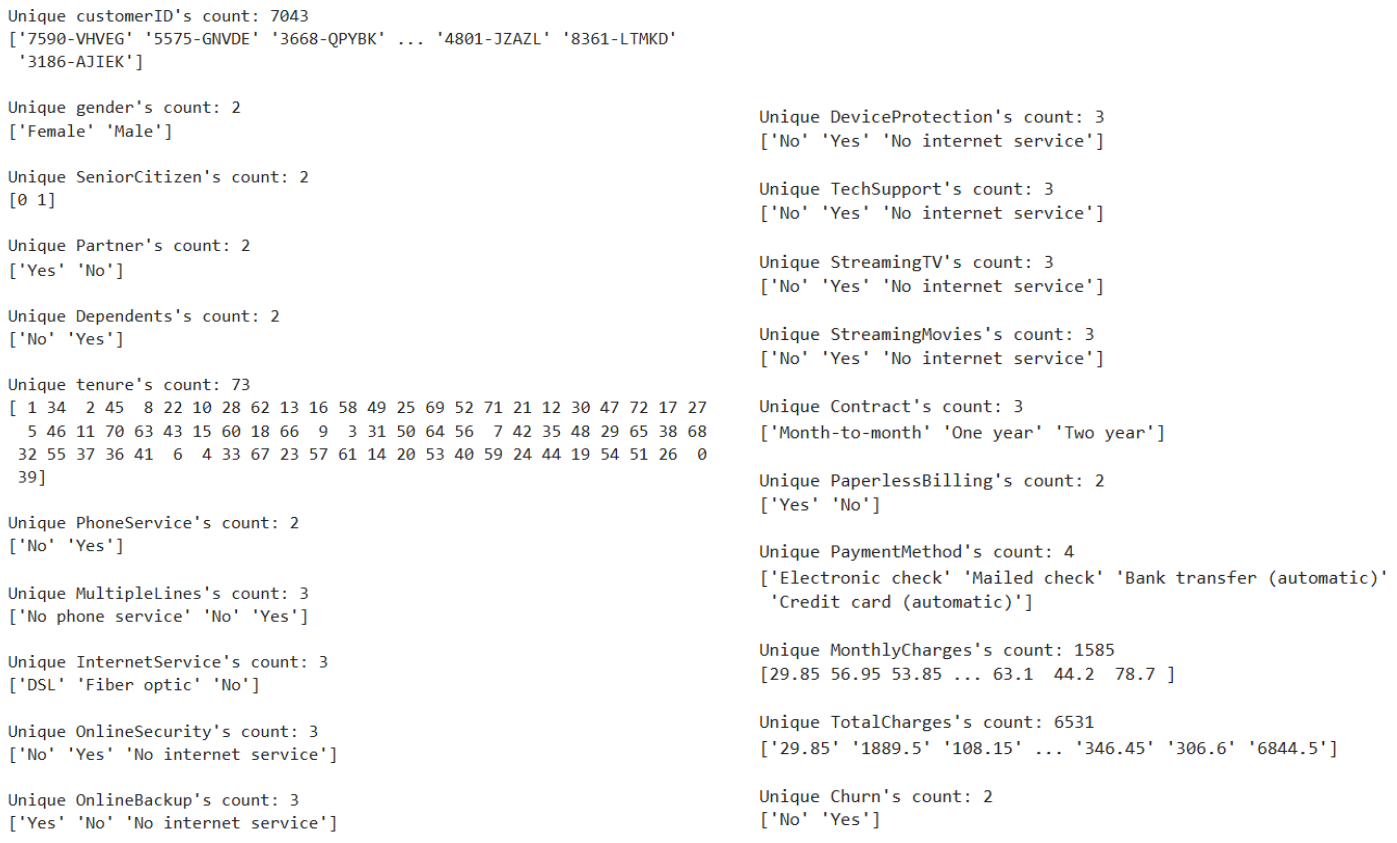
This dataset has 7043 rows and 21 columns

3.3 Meaning of variables:

An effective method for gaining deeper insights into feature values is to display the unique values of each feature in the dataset. This allows for a closer examination of the columns and their distinct entries, providing a clearer understanding of the dataset's structure. Furthermore, features can be categorized or grouped into clusters to uncover potential patterns and relationships.



```python
for column in df:
    unique_values = df[column].unique()
    unique_count = len(unique_values)
    print(f"Unique {column}'s count: {unique_count}")
    print(f"{unique_values}\n")
```



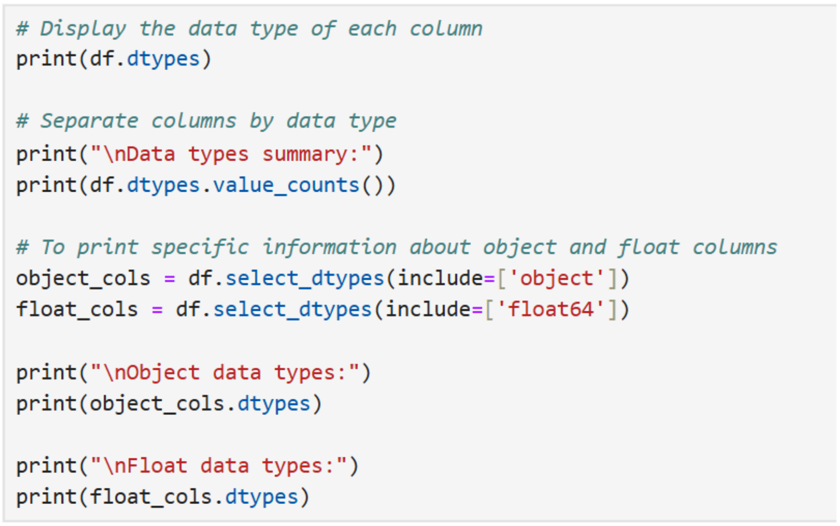
```
Unique customerID's count: 7043
['7590-VHVEG' '5575-GNVDE' '3668-QPYBK' ... '4801-JZAZL' '8361-LTMKD'
 '3186-AJIEK']

Unique gender's count: 2
['Female' 'Male']

Unique SeniorCitizen's count: 2
[0 1]

Unique Partner's count: 2
['Yes' 'No']

Unique Dependents's count: 2
['No' 'Yes']

Unique tenure's count: 73
[ 1 34  2 45  8 22 10 28 62 13 16 58 49 25 69 52 71 21 12 30 47 72 17 27
  5 46 11 70 63 43 15 60 18 66  9  3 31 50 64 56  7 42 35 48 29 65 38 68
 32 55 37 36 41  6  4 33 67 23 57 61 14 20 53 40 59 24 44 19 54 51 26  0
 39]

Unique PhoneService's count: 2
['No' 'Yes']

Unique MultipleLines's count: 3
['No phone service' 'No' 'Yes']

Unique InternetService's count: 3
['DSL' 'Fiber optic' 'No']

Unique OnlineSecurity's count: 3
['No' 'Yes' 'No internet service']

Unique OnlineBackup's count: 3
['Yes' 'No' 'No internet service']

Unique DeviceProtection's count: 3
['No' 'Yes' 'No internet service']

Unique TechSupport's count: 3
['No' 'Yes' 'No internet service']

Unique StreamingTV's count: 3
['No' 'Yes' 'No internet service']

Unique StreamingMovies's count: 3
['No' 'Yes' 'No internet service']

Unique Contract's count: 3
['Month-to-month' 'One year' 'Two year']

Unique PaperlessBilling's count: 2
['Yes' 'No']

Unique PaymentMethod's count: 4
['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']

Unique MonthlyCharges's count: 1585
[29.85 56.95 53.85 ... 63.1  44.2  78.7 ]

Unique TotalCharges's count: 6531
['29.85' '1889.5' '108.15' ... '346.45' '306.6' '6844.5']

Unique Churn's count: 2
['No' 'Yes']
```

Variable Name	**Description**	**Data Type**
customerID	- Unique identifier for each customer. - Used for customer identification.	String
gender	- Gender of the customer. - Can influence customer behavior and preferences	Categorical: Male, Female

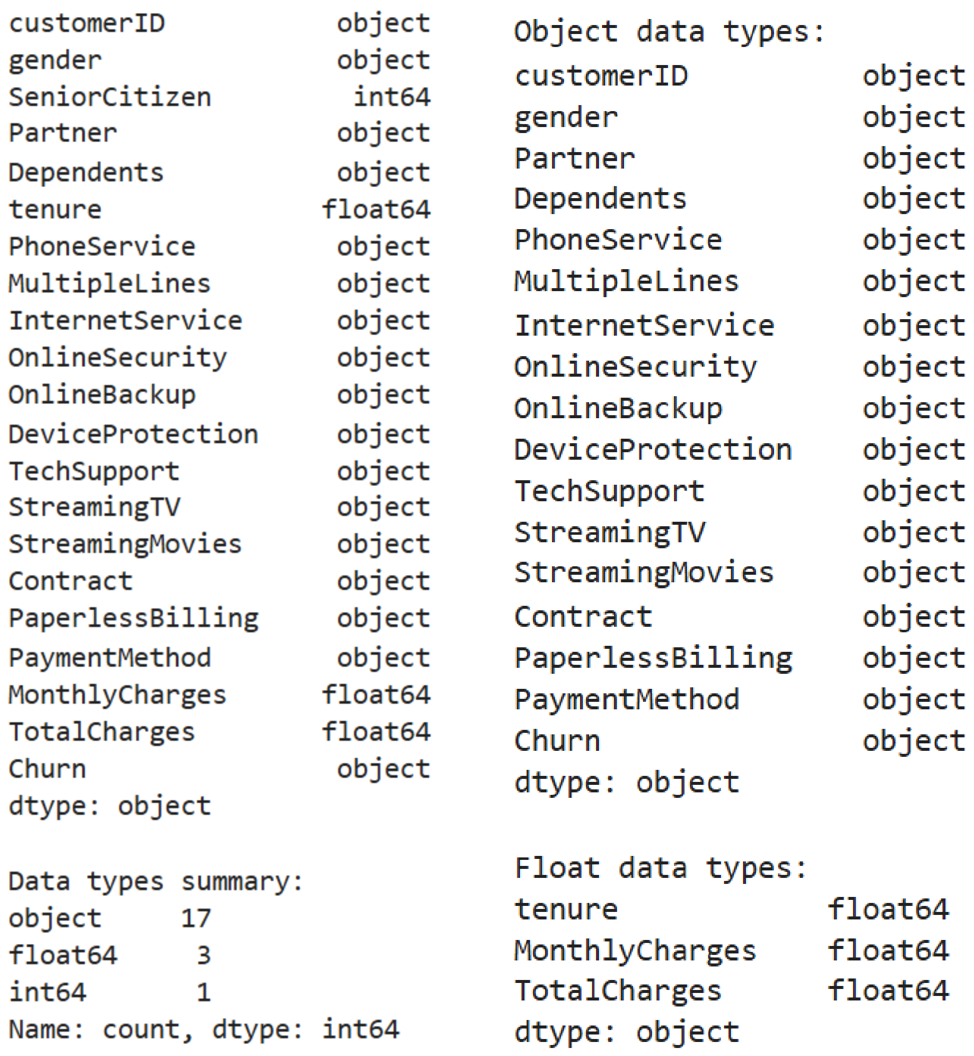
SeniorCitizen	- Senior citizen status. - Identifies senior customers who may have different churn behavior.	Binary: 0: No 1: Yes
Partner	- Marital status - Can impact loyalty and churn rates.	Categorical: Yes: Yes No: No
Dependents	- Dependents status - Affects customer stability and financial responsibility	Categorical: Yes: Yes No: No
tenure	- Duration of service. - Longer tenure typically correlates with lower churn risk.	Integer: months
PhoneService	- Subscription to telephone service - Indicates if the customer subscribes to a phone service	Binary: Yes: Yes No: No
MultipleLines	- Subscription to multiple lines for fixed-line service. - Shows whether the customer has multiple phone lines.	Categorical: Yes: Yes, No: No service, No phone service: No phone service
InternetService	- Subscription to internet service. - Indicates the type of internet service.	Categorical: DSL: Digital Subscriber Line Fiber optic: Fiber optic cable No: No internet service
OnlineSecurity	- Subscription to online security service. - Shows if the customer subscribes to online security	Categorical: Yes, No, No internet service: No internet service
OnlineBackup	- Subscription to online backup service - Indicates if the customer uses online backup services.	Categorical: Yes, No, No internet service: No internet service
DeviceProtection	- Subscription to device protection service	Categorical: Yes,

	- Indicates if the customer subscribes to device protection..	No, No internet service: No internet service
TechSupport	- Subscription to technical support service - Shows if the customer uses technical support services.	Categorical: Yes, No, No internet service: No internet service
StreamingTV	- Subscription to streaming TV service - Indicates if the customer subscribes to streaming TV.	Categorical: Yes, No, No internet service: No internet service
StreamingMovies	- Subscription to streaming movie service - Indicates if the customer subscribes to streaming movies.	Categorical: Yes, No, No internet service: No internet service
Contract	- Type of contract - Reflects the length and type of commitment of the customer.	Categorical: Month-to-month, One year, Two years
PaperlessBilling	- Enrollment in paperless billing. - Indicates whether the customer uses paperless billing.	Binary: Yes: Yes, No: No
PaymentMethod	- Payment method - Reflects the customer’s preferred payment method.	Categorical: Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)
MonthlyCharges	- Monthly charges for services. - The amount billed monthly for telecom services	Float: USD
TotalCharges	- Total amount paid for services - The cumulative amount paid by the customer.	Float: USD
Churn	- Churn status - Indicates if the customer has churned (left the service)	Binary: Yes: Churned, No: Not churned

3.4 Missing value:



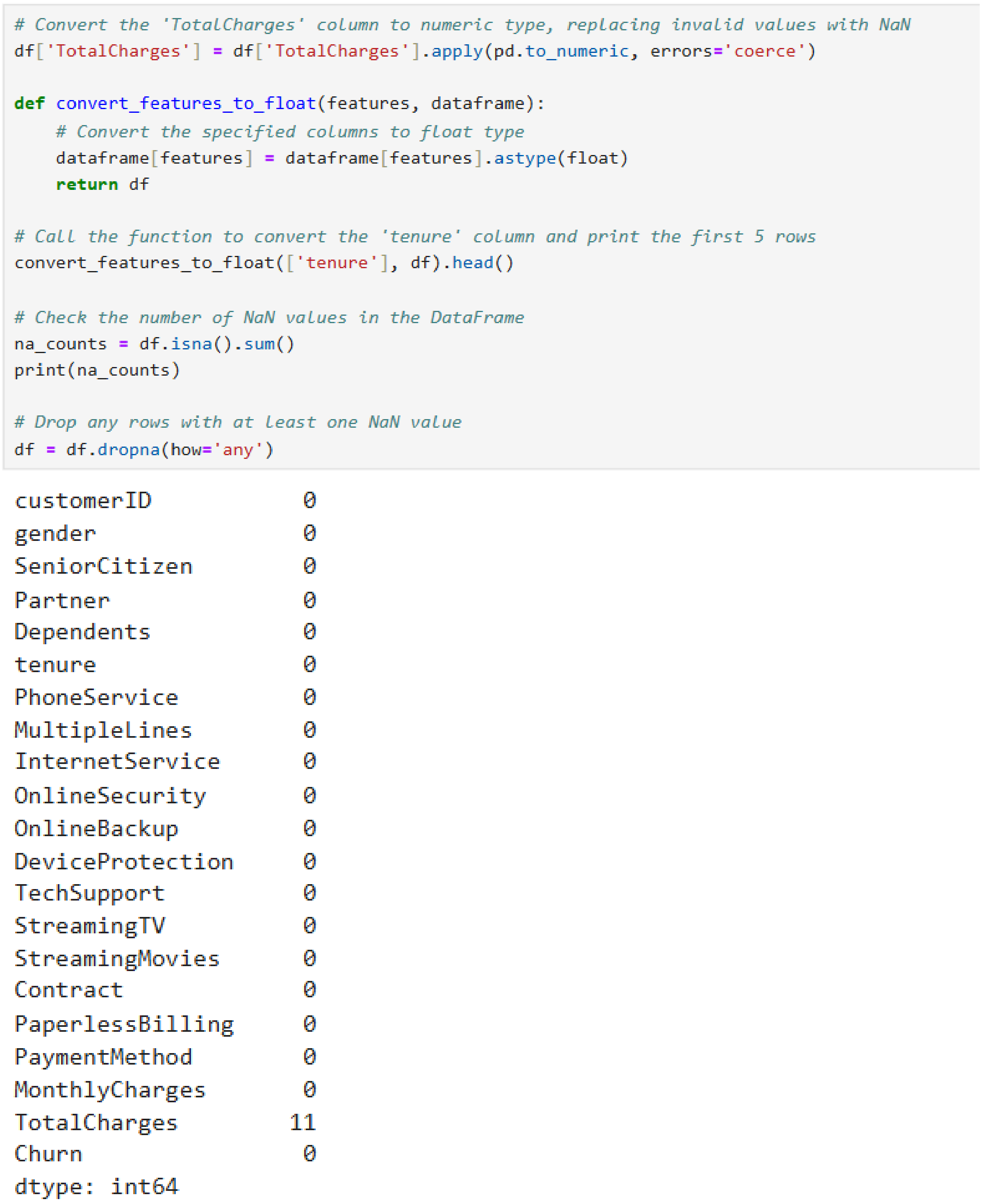
```python
# Display the data type of each column
print(df.dtypes)

# Separate columns by data type
print("\nData types summary:")
print(df.dtypes.value_counts())

# To print specific information about object and float columns
object_cols = df.select_dtypes(include=['object'])
float_cols = df.select_dtypes(include=['float64'])

print("\nObject data types:")
print(object_cols.dtypes)

print("\nFloat data types:")
print(float_cols.dtypes)
```



```
customerID           object
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure              float64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
TotalCharges        float64
Churn                object
dtype: object

Data types summary:
object     17
float64     3
int64       1
Name: count, dtype: int64
```

```
Object data types:
customerID          object
gender              object
Partner             object
Dependents          object
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
Churn               object
dtype: object

Float data types:
tenure            float64
MonthlyCharges    float64
TotalCharges      float64
dtype: object
```

From the given information, it seems that the data types for "TotalCharges" and "tenure" should be float64 rather than object. Hence, we will convert the data type of "TotalCharges" to float64 accordingly.



```python
# Convert the 'TotalCharges' column to numeric type, replacing invalid values with NaN
df['TotalCharges'] = df['TotalCharges'].apply(pd.to_numeric, errors='coerce')

def convert_features_to_float(features, dataframe):
    # Convert the specified columns to float type
    dataframe[features] = dataframe[features].astype(float)
    return df

# Call the function to convert the 'tenure' column and print the first 5 rows
convert_features_to_float(['tenure'], df).head()

# Check the number of NaN values in the DataFrame
na_counts = df.isna().sum()
print(na_counts)

# Drop any rows with at least one NaN value
df = df.dropna(how='any')
```

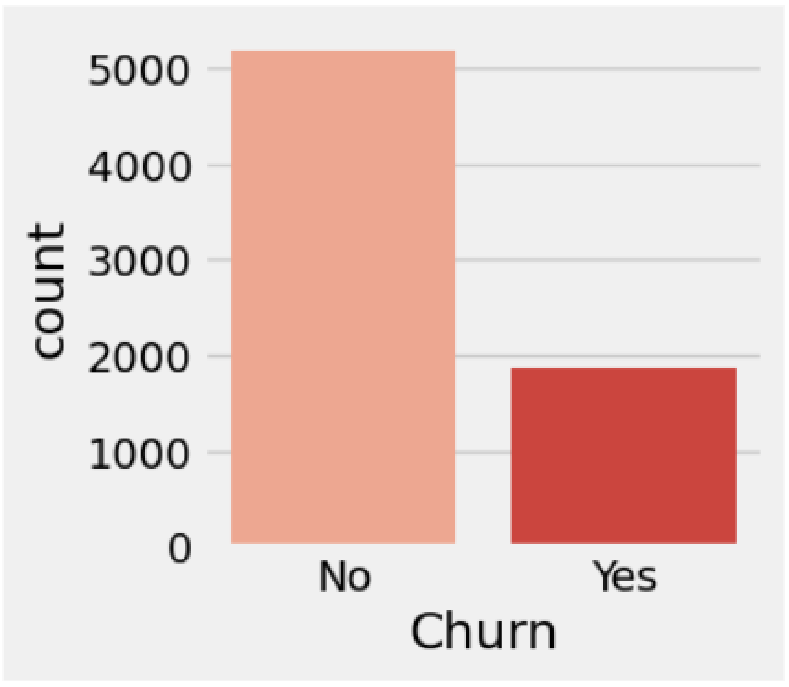
```
customerID           0
gender               0
SeniorCitizen        0
Partner              0
Dependents           0
tenure               0
PhoneService         0
MultipleLines        0
InternetService      0
OnlineSecurity       0
OnlineBackup         0
DeviceProtection     0
TechSupport          0
StreamingTV          0
StreamingMovies      0
Contract             0
PaperlessBilling     0
PaymentMethod        0
MonthlyCharges       0
TotalCharges        11
Churn                0
dtype: int64
```

The analysis displays 11 missing values of “TotalCharges”. So, removing missing values “TotalCharges”.

3.4 Exploratory analysis

3.4.1 Fivethirtyeight

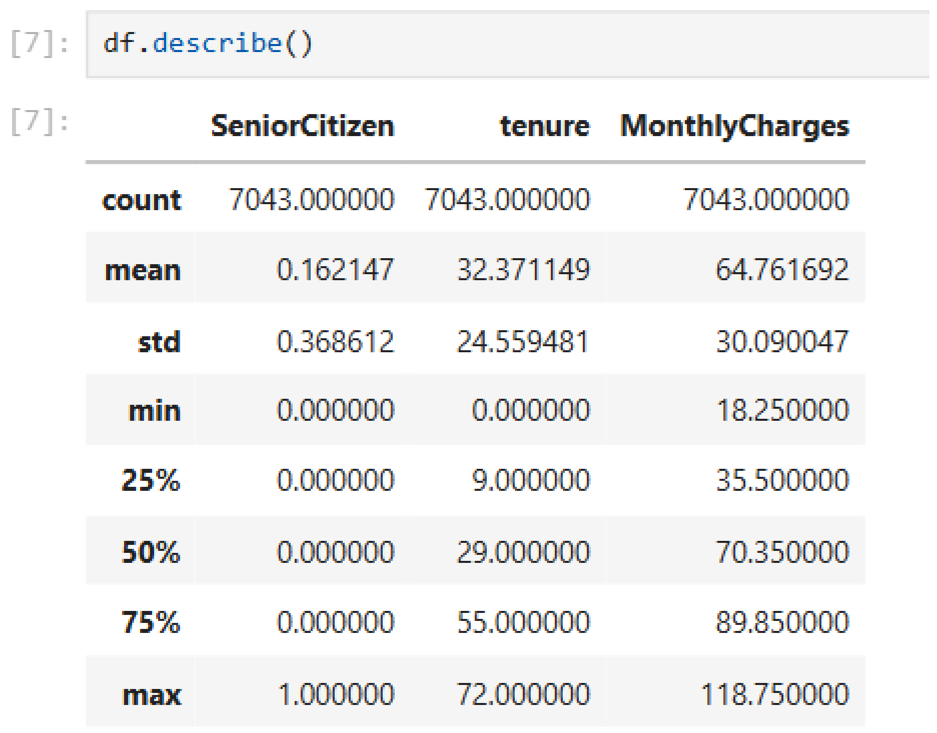
First, the Fivethirtyeight style will be applied to all visualizations. This style is popular in data analysis for its clean design, featuring high-contrast colors and clear, readable fonts. Next, a graph will be created to show the frequency distribution of churn. This visualization will offer valuable insights into the different churn categories and their associated frequencies.



- The graph highlights a noticeable imbalance between customers who churn and those who remain. One potential solution to address this issue is resampling the data.
- However, to maintain simplicity in this analysis, the class imbalance will be left as is. Instead, the focus will shift to using carefully chosen evaluation metrics to assess the models.

3.4.2 Descriptive statistics

Perform descriptive statistics for continuous variables in the data set.



[7]:
```
df.describe()
```

[7]:

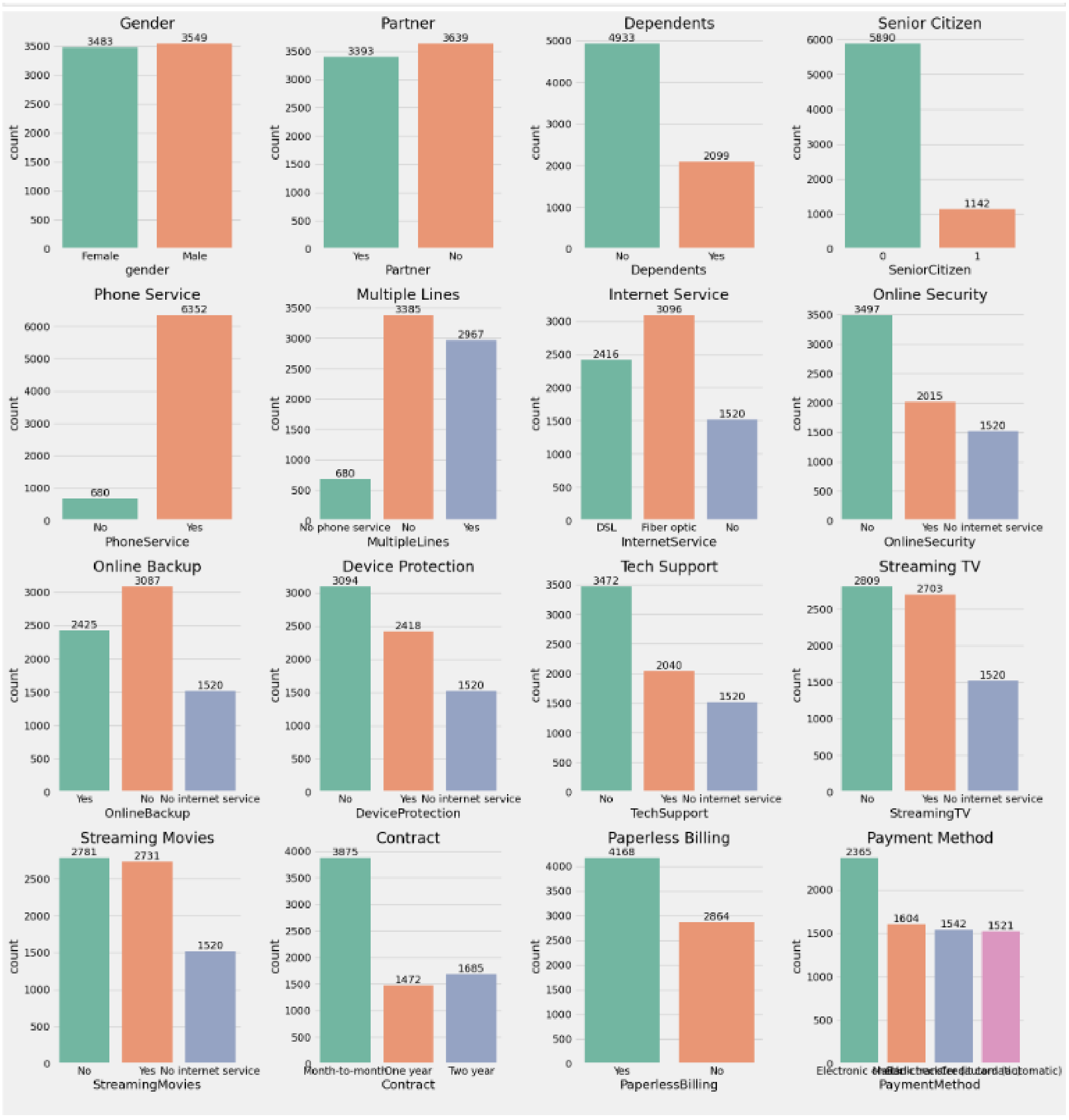
	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

- The data indicates that the majority of customers are not senior citizens, as the 25th, 50th, and 75th percentiles all have a value of 0. Only about 16.2% of the customers are senior citizens, based on the mean value of 0.162.
- The tenure column highlights the diversity in the duration of service usage. It ranges from 0 months (new customers) to 72 months (long-term users). The median tenure is 29 months, and about 75% of the customers have been using the service for less than 55 months.
- The average monthly charge is approximately 64.76, with a wide spread as indicated by the standard deviation of 30.09. The charges range from a minimum of 18.25 to a maximum of 118.75, reflecting the availability of different service packages catering to various customer needs and preferences.

3.4.3 Stacked bar chart:

Visualize the distribution of the data for each attribute using a chart:

```python
# List of column names and titles for countplots
columns = [
    'gender', 'Partner', 'Dependents', 'SeniorCitizen',
    'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
    'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
    'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod'
]

titles = [
    'Gender', 'Partner', 'Dependents', 'Senior Citizen',
    'Phone Service', 'Multiple Lines', 'Internet Service', 'Online Security',
    'Online Backup', 'Device Protection', 'Tech Support', 'Streaming TV',
    'Streaming Movies', 'Contract', 'Paperless Billing', 'Payment Method'
]

# Create figure and axes
fig, ax = plt.subplots(5, 4, figsize=(20, 25))

# Loop through columns and plot countplots
for i, (col, title) in enumerate(zip(columns, titles)):
    row, col_pos = divmod(i, 4)  # Determine row and column position
    # Generate a unique palette for each column based on its unique values
    unique_values = df[col].unique()
    palette = sns.color_palette("Set2", len(unique_values))

    # Plot countplot with the generated palette, using hue
    sns.countplot(data=df, x=col, hue=col, ax=ax[row][col_pos], palette=palette, dodge=False, legend=False)
    ax[row][col_pos].set_title(title)

    # Add counts on bars
    for patch in ax[row][col_pos].patches:
        x = patch.get_x() + patch.get_width() / 2
        y = patch.get_height()
        ax[row][col_pos].annotate(int(y), (x, y), ha='center', va='bottom')

# Hide unused subplots
for i in range(len(columns), 20):
    row, col_pos = divmod(i, 4)
    ax[row][col_pos].axis('off')

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



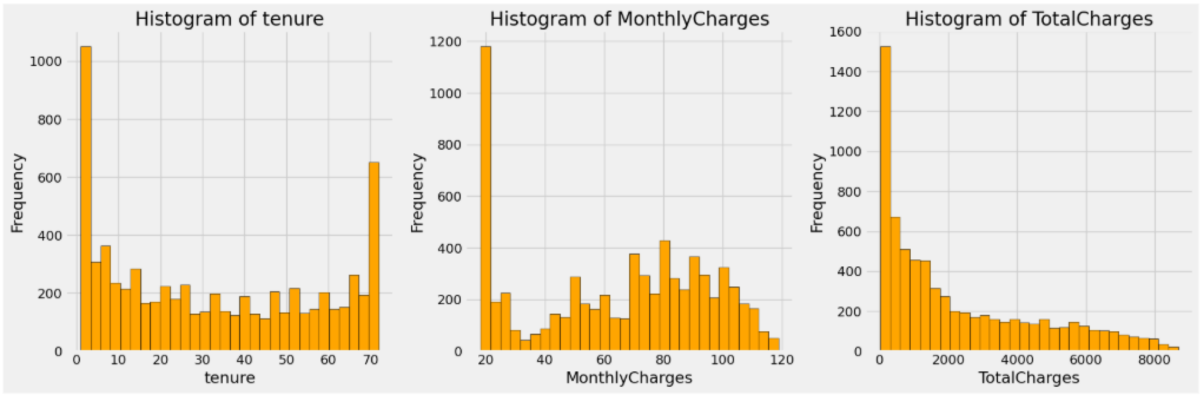
Here's a revised version of your analysis in a more streamlined format:

- **Gender Chart**: The gender distribution is almost balanced, with a slight majority of male customers (3549 males vs. 3483 females), indicating that the company's services appeal equally to both genders. There is no need for gender-specific marketing strategies as both groups show similar usage patterns.
- **Partner Chart**: A higher proportion of customers are not married (3639 non-married vs. 3393 married), suggesting that unmarried individuals make up a larger portion of the customer base. Marketing strategies could be tailored to focus more on unmarried customers, who may have different service preferences compared to married customers.

- **Dependents Chart**: More customers are independent (4933 independent vs. 2099 dependent), likely reflecting a customer base that is financially independent and capable of making decisions without family support. The company could target working professionals and independent individuals for tailored service offerings.
- **SeniorCitizen Chart**: A large majority of the customers are not senior citizens (5901 non-senior citizens vs. 1142 senior citizens), indicating that the company's services are more popular among younger or middle-aged individuals. While senior citizens represent a smaller group, the company could develop specialized services to increase engagement with this demographic.
- **PhoneService Chart**: The majority of customers (6352) use the phone service, with only 680 opting out. This suggests that phone service is a core offering that attracts most of the customer base. The company should continue promoting its phone services as a primary feature and ensure its reliability to retain customers.
- **MultipleLines Chart**: Most customers (3385) do not use multiple lines, with only 680 customers opting for this service. Since only a small portion of people do not use the company's phone service, the company could explore ways to encourage customers to opt for multiple lines, potentially offering discounts or incentives.
- **InternetService Chart**: A significant portion of customers (around 3000) use fiber optic internet, followed by 2300 using DSL. The rest of the customers do not use internet services. Fiber optic internet is the preferred choice, and the company should focus on expanding and improving this service to meet customer demand.
- **OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies Charts**: The majority of customers do not use these services, with usage rates falling below 50%. This could indicate that these services are perceived as optional or unnecessary by many customers. The company might consider bundling these services with core offerings or enhancing their value propositions to increase adoption.
- **Contract Chart**: The majority of customers prefer month-to-month contracts (5000), followed by 2-year contracts. A smaller portion opts for 1-year contracts, suggesting that customers value flexibility. The company should continue to offer flexible contract options while finding ways to incentivize longer-term commitments.
- **PaperlessBilling Chart**: More customers do not use paperless billing (4176 non-users vs. 2664 users), suggesting a preference for traditional paper billing. The company may need to promote the advantages of paperless billing, such as environmental impact and convenience, to encourage greater adoption.
- **PaymentMethod Chart**: A significant portion of customers use electronic payment methods like credit cards and automatic bank transfers (1604 and 1542 customers, respectively). This trend toward digital payments highlights the need for the company to maintain and improve its digital payment options for convenience and security.

3.4.4 Histogram chart:

Create a histogram chart for the three variables: "Tenure", "MonthlyCharges", and "TotalCharges" to visualize the data distribution.

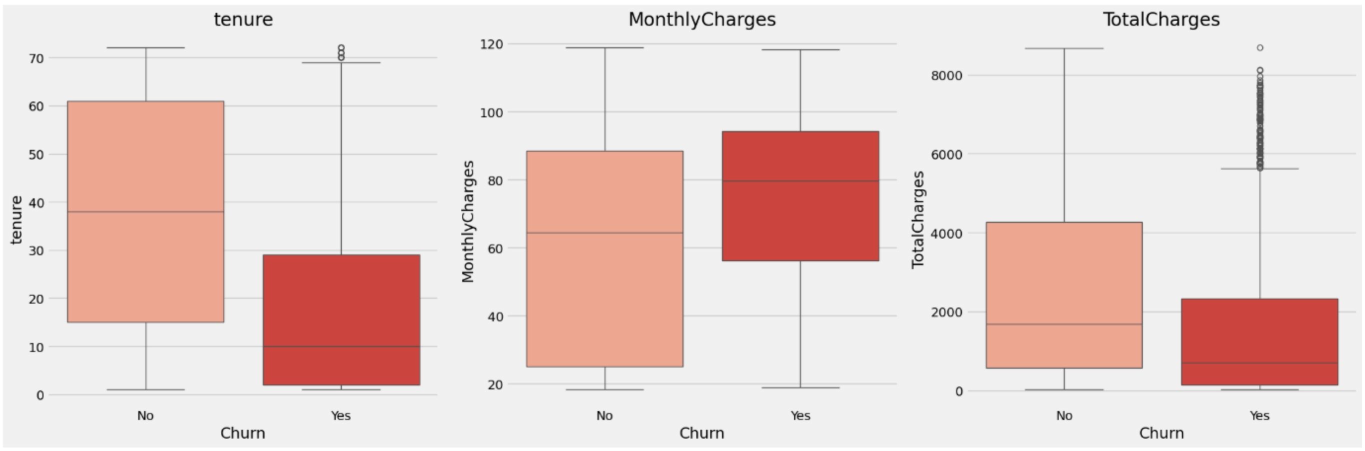


- **Tenure**: The "tenure" variable exhibits a left-skewed distribution with a pronounced peak at the 0-10 range, followed by a smaller peak around the 70 mark. This suggests that there are two distinct groups of customers: those who are relatively new to the service and those who have been using it for a longer period. The data indicates a higher concentration of newer customers compared to long-term users.
- **MonthlyCharges**: The "MonthlyCharges" variable shows a right-skewed distribution with a peak on the lower end and a long tail extending toward higher values. This indicates that most customers pay a low or moderate monthly charge, but there are a smaller number of customers who pay significantly higher fees. The majority of customers seem to fall within a typical price range, while a smaller proportion of customers incur higher charges.
- **TotalCharges**: The "TotalCharges" variable follows an exponential decay distribution, with a higher frequency of customers reporting lower or moderate total charges and a much lower frequency of customers with higher total charges. This indicates that the majority of customers have spent relatively less on the service, while only a small fraction have incurred significantly higher costs. The data suggests that most customers are either new or have low monthly charges, resulting in lower overall spending.

3.4.5. Boxplot:

Create boxplot charts for three variables: tenure, MonthlyCharges, and TotalCharges with respect to customer churn status (Yes or No).

```python
def boxplot(x, y, df):
    # Calculate the number of rows and columns needed for subplots
    num_plots = len(y)
    rows = (num_plots - 1) // 3 + 1  # Determine the number of rows (3 plots per row)
    columns = min(3, num_plots)      # Maximum 3 columns

    plt.figure(figsize=(7 * columns, 7 * rows))

    for i, j in enumerate(y):
        # Create a subplot at position i+1
        plt.subplot(rows, columns, i + 1)

        # Create the boxplot
        ax = sns.boxplot(x=x, y=j, data=df, hue=x, palette="Reds", linewidth=1, dodge=False)
        ax.set_title(j)

        # Check and remove the legend if it exists
        if ax.legend_ is not None:
            ax.legend_.remove()

    plt.tight_layout()  # Adjust the layout to avoid overlap
    plt.show()

# Call the function with the list of y columns
boxplot("Churn", ["tenure", "MonthlyCharges", "TotalCharges"], df)
```



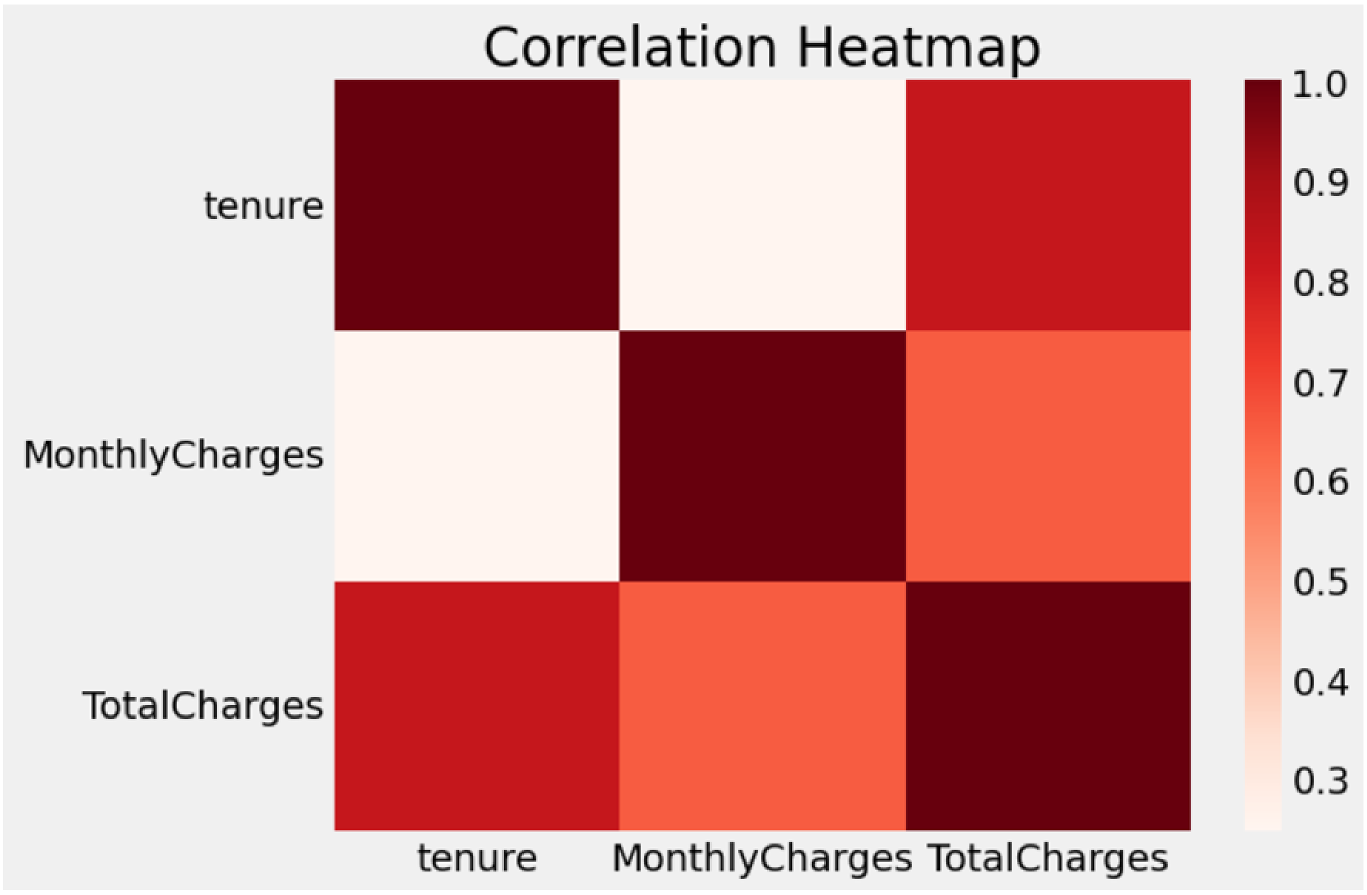
- **Tenure**: For customers who did not churn (No), the tenure tends to be higher, with the median tenure being around 40-50 months. In contrast, customers who churned (Yes) generally have a lower tenure, with the median closer to 10 months. This indicates that customers who have been with the company for a longer period are less likely to churn.
- **MonthlyCharges**: The boxplot for "MonthlyCharges" reveals that customers who churned tend to pay a lower monthly fee, with the median around 60-70. Non-churning customers, on the other hand, have a higher median monthly charge of around 80-90. This suggests that

customers with lower monthly charges are more likely to churn, possibly indicating a higher proportion of budget-conscious or less engaged customers.

- **TotalCharges**: The "TotalCharges" boxplot shows that non-churning customers generally have higher total charges, with a median around 2000, compared to customers who churned, who have a much lower median total charge (around 1500). Additionally, there are some outliers for both groups, especially for non-churning customers, with a few who have paid significantly higher total charges.

3.4.6 Correlation heatmap:

```
corr = df[numerical_vars].corr()

# Draw heatmap with red
sns.heatmap(corr, cmap='Reds')

plt.title('Correlation Heatmap')
plt.show()
```

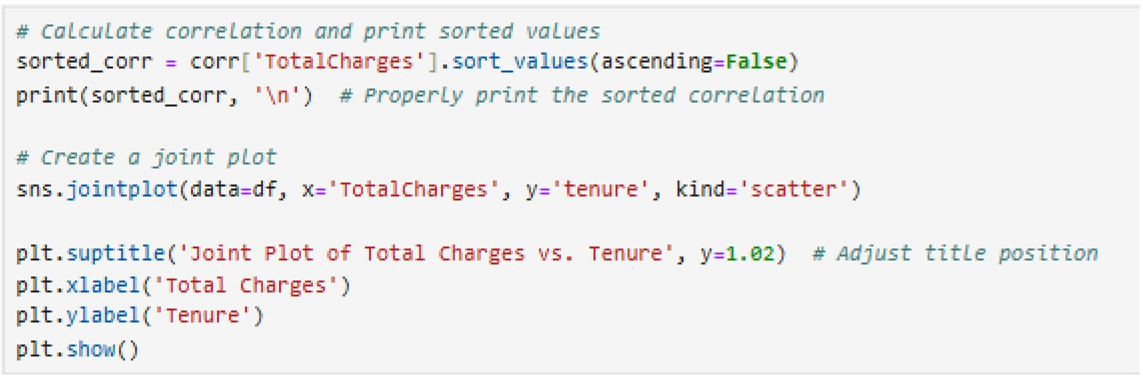


Tenure and Monthly Charges: The dark color in this cell indicates a strong positive correlation between the two variables. This means that customers who have been using the service for a longer period tend to pay higher monthly fees.

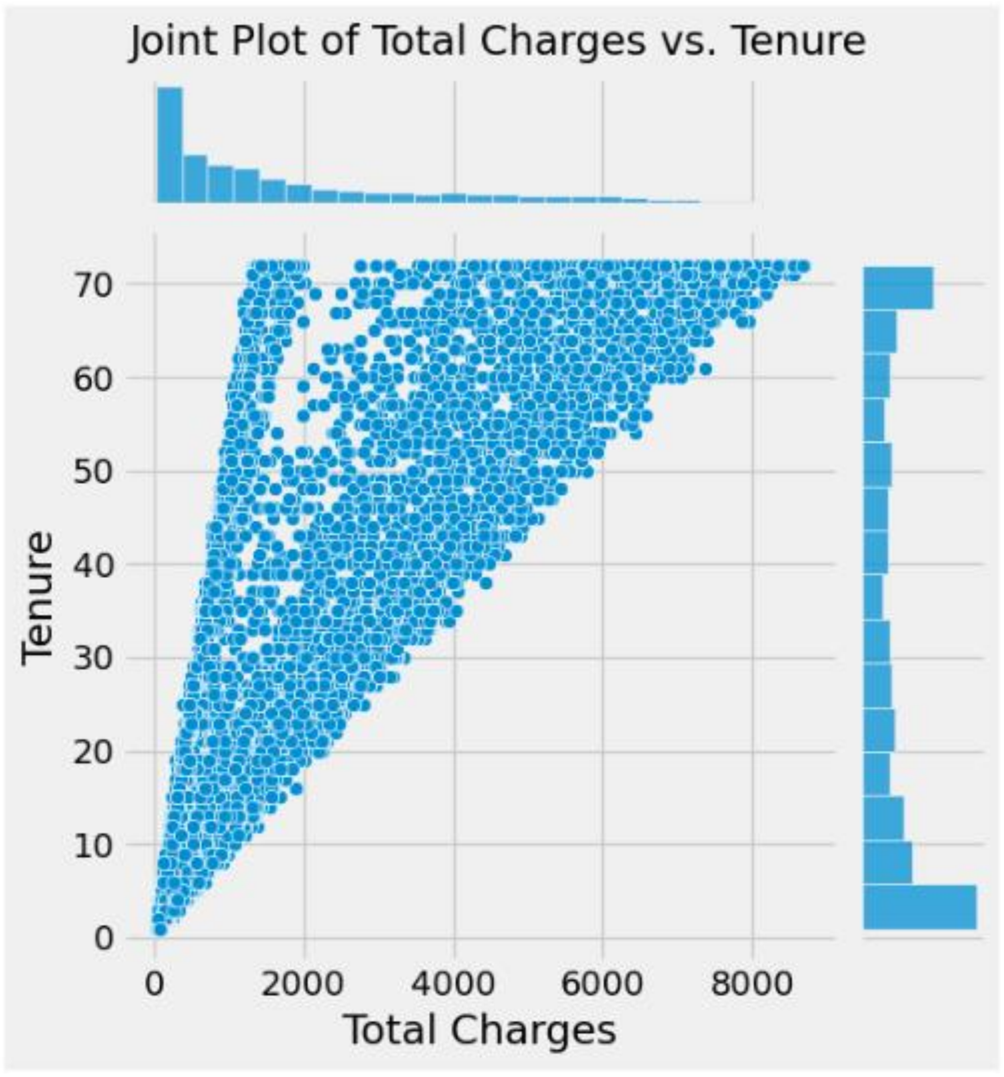
Tenure and Total Charges: Similarly, the correlation between tenure and total charges is also very strong. This is logical because as tenure increases, so does the total amount charged.

Monthly Charges and Total Charges: The correlation between monthly charges and total charges is also very high. This is understandable since total charges are the accumulation of monthly charges over time.

3.4.5 Joint Scatter Plot:



```python
# Calculate correlation and print sorted values
sorted_corr = corr['TotalCharges'].sort_values(ascending=False)
print(sorted_corr, '\n')  # Properly print the sorted correlation

# Create a joint plot
sns.jointplot(data=df, x='TotalCharges', y='tenure', kind='scatter')

plt.suptitle('Joint Plot of Total Charges vs. Tenure', y=1.02)  # Adjust title position
plt.xlabel('Total Charges')
plt.ylabel('Tenure')
plt.show()
```

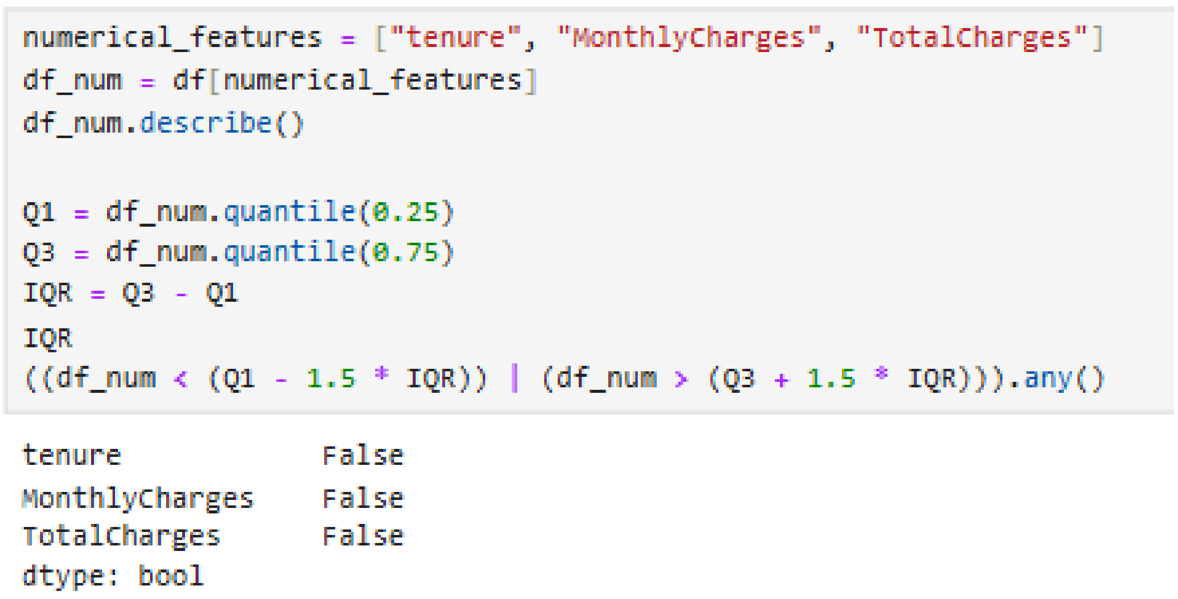


The joint plot between Total Charges (X-axis) and Tenure (Y-axis) shows that Total Charges range from 0 to around 8000, with some customers paying significantly more. The Tenure ranges from 0 to about 80 months, indicating varying lengths of service usage. Scatter points show a concentration of higher charges with longer tenures, suggesting a positive relationship between the two variables. Marginal histograms reveal that most customers have lower Total Charges and Tenure, with fewer customers having high values in both.

The analysis highlights a strong positive correlation between Total Charges and Tenure, indicating that customers with longer service durations tend to pay more. However, most customers fall within the lower ranges for both metrics, suggesting that businesses could focus on retaining short-term customers who may potentially increase their payments over time.

3.4.6 IQR:

Analyze a dataset for outliers based on the interquartile range method



```
numerical_features = ["tenure", "MonthlyCharges", "TotalCharges"]
df_num = df[numerical_features]
df_num.describe()

Q1 = df_num.quantile(0.25)
Q3 = df_num.quantile(0.75)
IQR = Q3 - Q1
IQR
((df_num < (Q1 - 1.5 * IQR)) | (df_num > (Q3 + 1.5 * IQR))).any()
```

```
tenure            False
MonthlyCharges    False
TotalCharges      False
dtype: bool
```

Based on the analysis, the dataset for these three features does not contain any outliers, as all results are False. This suggests that the values in these columns are within the expected range, and there is no need to perform any outlier removal for these variables.

3.4.7. Special point or potential issue that the analyst must pay attention to:

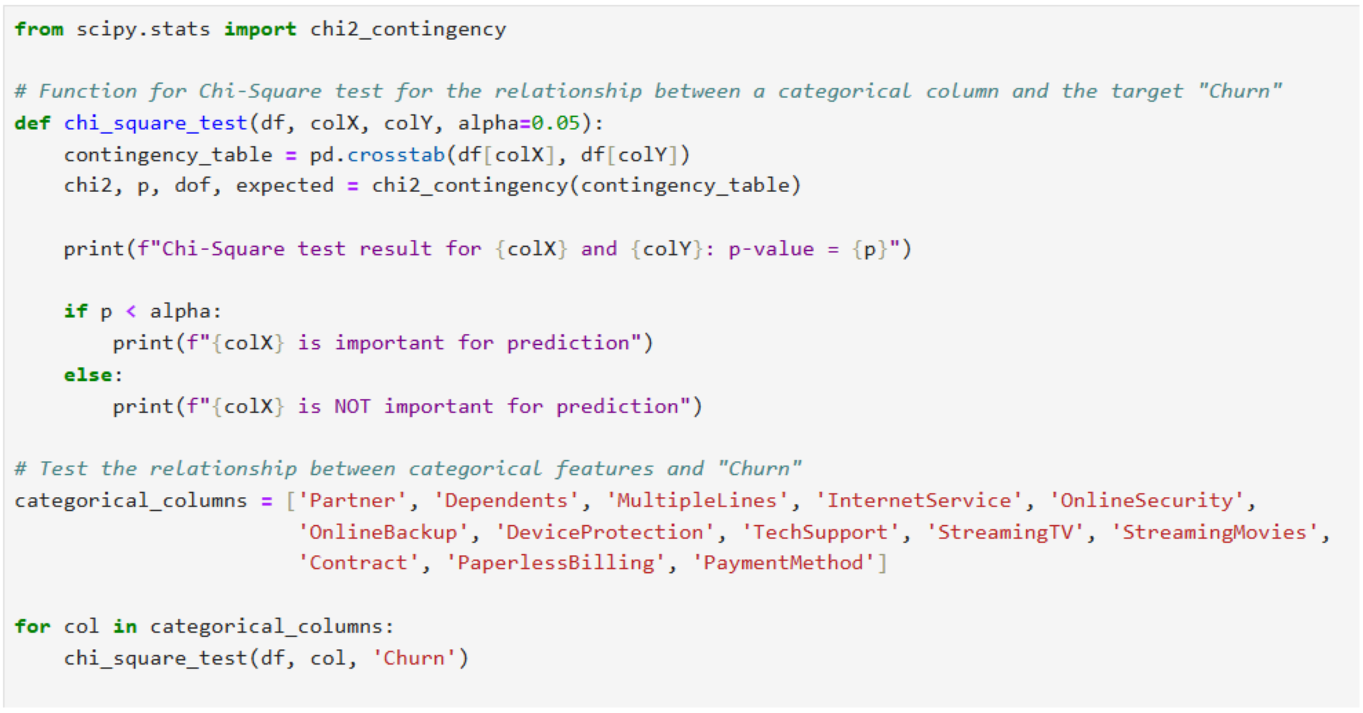
When analyzing the data, there are some **Special points and potential issues** need to be addressed:

- **Data Cleaning**: The 'TotalCharges' column may contain non-numeric values that should be either converted to numeric format or treated as missing data to ensure the integrity of the analysis.
- **Class Imbalance**: An imbalance between the churned and retained customers can distort model performance and limit its ability to generalize across different customer segments.
- **Feature Selection**: Certain features may have minimal impact on the model's predictive accuracy. Removing these irrelevant features can help simplify the model, improving both performance and efficiency.
- **Model Interpretability**: Depending on the model selected, its interpretability can vary. If stakeholders need to understand the model's decisions, this should guide the choice of modeling technique.
- **Evaluation Metrics**: Choosing the right evaluation metrics is critical. Metrics should align with the business objectives, with a particular emphasis on balancing precision and recall, especially in cases where the cost of false negatives is high.
- **Temporal Dynamics**: Customer behaviors and churn factors evolve over time. Therefore, the model should be updated regularly to reflect these changes and maintain its accuracy.

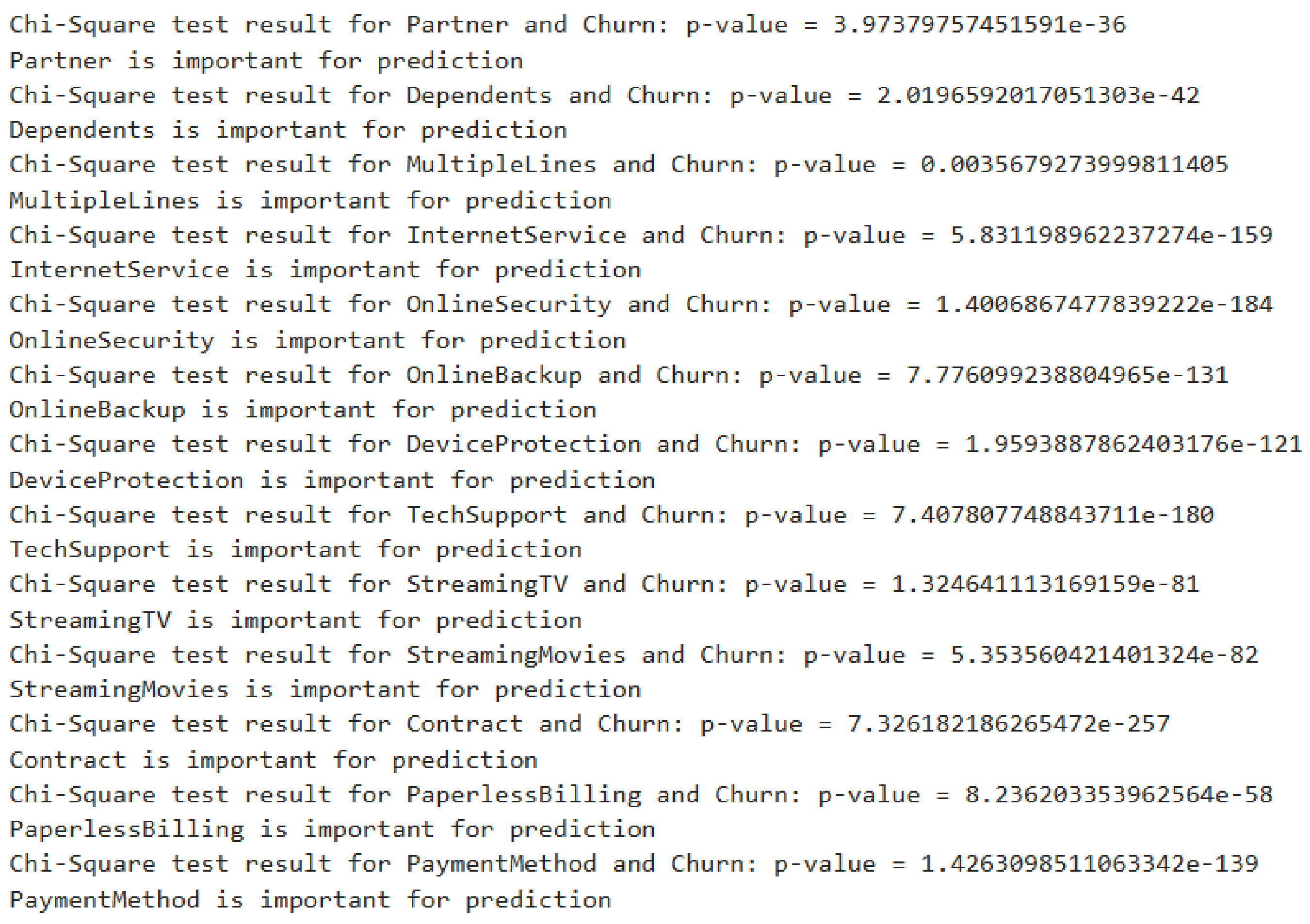
By considering these aspects, the analysis will be more robust, and the resulting model will better align with the business objectives and adapt to changing conditions.

3.5. Feature inspections:

3.5.1 Chi-square testing



```python
from scipy.stats import chi2_contingency

# Function for Chi-Square test for the relationship between a categorical column and the target "Churn"
def chi_square_test(df, colX, colY, alpha=0.05):
    contingency_table = pd.crosstab(df[colX], df[colY])
    chi2, p, dof, expected = chi2_contingency(contingency_table)

    print(f"Chi-Square test result for {colX} and {colY}: p-value = {p}")

    if p < alpha:
        print(f"{colX} is important for prediction")
    else:
        print(f"{colX} is NOT important for prediction")

# Test the relationship between categorical features and "Churn"
categorical_columns = ['Partner', 'Dependents', 'MultipleLines', 'InternetService', 'OnlineSecurity',
                       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
                       'Contract', 'PaperlessBilling', 'PaymentMethod']

for col in categorical_columns:
    chi_square_test(df, col, 'Churn')
```



```
Chi-Square test result for Partner and Churn: p-value = 3.97379757451591e-36
Partner is important for prediction
Chi-Square test result for Dependents and Churn: p-value = 2.0196592017051303e-42
Dependents is important for prediction
Chi-Square test result for MultipleLines and Churn: p-value = 0.0035679273999811405
MultipleLines is important for prediction
Chi-Square test result for InternetService and Churn: p-value = 5.831198962237274e-159
InternetService is important for prediction
Chi-Square test result for OnlineSecurity and Churn: p-value = 1.4006867477839222e-184
OnlineSecurity is important for prediction
Chi-Square test result for OnlineBackup and Churn: p-value = 7.776099238804965e-131
OnlineBackup is important for prediction
Chi-Square test result for DeviceProtection and Churn: p-value = 1.9593887862403176e-121
DeviceProtection is important for prediction
Chi-Square test result for TechSupport and Churn: p-value = 7.407807748843711e-180
TechSupport is important for prediction
Chi-Square test result for StreamingTV and Churn: p-value = 1.324641113169159e-81
StreamingTV is important for prediction
Chi-Square test result for StreamingMovies and Churn: p-value = 5.353560421401324e-82
StreamingMovies is important for prediction
Chi-Square test result for Contract and Churn: p-value = 7.326182186265472e-257
Contract is important for prediction
Chi-Square test result for PaperlessBilling and Churn: p-value = 8.236203353962564e-58
PaperlessBilling is important for prediction
Chi-Square test result for PaymentMethod and Churn: p-value = 1.4263098511063342e-139
PaymentMethod is important for prediction
```

The Chi-Square test results reveal that all the categorical features tested have a significant relationship with the target variable "Churn" ($p\text{-value} < 0.05$). This indicates that these features play an important role in predicting customer churn. Notably:

- **Partner**, **Dependents**, **Contract**: These features strongly influence whether customers are likely to churn, with longer contracts and the presence of dependents reducing churn.
- **InternetService**, **TechSupport**, and **OnlineSecurity**: Services related to internet access and tech support significantly affect churn rates, suggesting that service quality impacts customer retention.

These insights can help telecommunications companies focus on the most influential factors to improve retention strategies and enhance service offerings.

3.5.2 ANOVA test

```python
from scipy import stats

# Function for ANOVA test for numerical features and the target "Churn"
def anova_test(df, col, target='Churn'):
    churn_yes = df[df[target] == 'Yes'][col]
    churn_no = df[df[target] == 'No'][col]

    f_stat, p_value = stats.f_oneway(churn_yes, churn_no)

    print(f"ANOVA test result for {col}: F-statistic = {f_stat}, p-value = {p_value}")

    if p_value < 0.05:
        print(f"{col} is important for prediction")
    else:
        print(f"{col} is NOT important for prediction")

# Test ANOVA for numerical features
numerical_columns = ['tenure', 'TotalCharges', 'MonthlyCharges']
for col in numerical_columns:
    anova_test(df, col)
```

```
ANOVA test result for tenure: F-statistic = 1007.5094314093412, p-value = 9.437650217603554e-207
tenure is important for prediction
ANOVA test result for TotalCharges: F-statistic = 291.3448623664935, p-value = 4.8768656897080145e-64
TotalCharges is important for prediction
ANOVA test result for MonthlyCharges: F-statistic = 271.5769897682043, p-value = 6.760843117999019e-60
MonthlyCharges is important for prediction
```

- **tenure**: The very high **F-statistic** (1007.51) and an extremely low **p-value** (9.44e-207) suggest a strong relationship between the **tenure** of a customer and the likelihood of churn. This indicates that **tenure** is a key feature, with longer-tenured customers more likely to stay. This statistical significance makes **tenure** a very important feature in churn prediction models.
- **TotalCharges**: The **F-statistic** (291.34) for **TotalCharges** is also high, and the **p-value** (4.88e-64) is extremely low. This further confirms that **TotalCharges** plays a significant role in predicting customer churn. It suggests that customers with higher total charges, possibly implying long-term or more extensive use of services, are less likely to churn.
- **MonthlyCharges:** Similar to the other features, **MonthlyCharges** has a high **F-statistic** (271.58) and an extremely low **p-value** (6.76e-60). The result indicates that the amount customers pay monthly has a statistically significant impact on churn. Customers with higher monthly charges are likely more invested in the service, making them less prone to leaving.
-

3.5.3 T-test

```python
from scipy.stats import ttest_ind

# Function for T-Test for numerical features and the target "Churn"
def t_test(df, col, target='Churn'):
    churn_yes = df[df[target] == 'Yes'][col]
    churn_no = df[df[target] == 'No'][col]

    t_stat, p_value = ttest_ind(churn_yes, churn_no, equal_var=False)

    print(f"T-test result for {col}: t-statistic = {t_stat}, p-value = {p_value}")

    if p_value < 0.05:
        print(f"{col} is important for prediction")
    else:
        print(f"{col} is NOT important for prediction")

# Test T-Test for numerical features
for col in numerical_columns:
    t_test(df, col)
```

```
T-test result for tenure: t-statistic = -34.97187009750348, p-value = 2.3470747188949526e-234
tenure is important for prediction
T-test result for TotalCharges: t-statistic = -18.80076821738761, p-value = 1.1524944112838114e-75
TotalCharges is important for prediction
T-test result for MonthlyCharges: t-statistic = 18.34091879095257, p-value = 2.6573571445160277e-72
MonthlyCharges is important for prediction
```

- **Tenure** shows a strong relationship with churn, with a very low p-value (2.35e-234), indicating its importance in the prediction model. The negative t-statistic (-34.97) further supports the hypothesis that longer tenure is associated with lower churn.
- **TotalCharges** also has a low p-value (1.15e-75) and a significant t-statistic (-18.80), suggesting that the total amount a customer has been charged plays a significant role in predicting churn.
- **MonthlyCharges** has an even higher t-statistic (18.34) and a p-value of 2.66e-72, indicating that monthly payments are a significant factor in churn prediction as well.

3.5.4 VFI

```
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Calculate VIF for numerical features
numerical_columns = ['tenure', 'TotalCharges', 'MonthlyCharges']
X_numerical = df[numerical_columns]

# Adding a constant for VIF calculation
X_numerical = sm.add_constant(X_numerical)
vif_data = pd.DataFrame()
vif_data["Feature"] = X_numerical.columns
vif_data["VIF"] = [variance_inflation_factor(X_numerical.values, i) for i in range(X_numerical.shape[1])]
print(vif_data)
```

```
          Feature        VIF
0           const  14.973839
1          tenure   5.844646
2    TotalCharges   9.526697
3  MonthlyCharges   3.225293
```

The Variance Inflation Factor (VIF) results show the degree of multicollinearity among the numerical features. A higher VIF indicates greater multicollinearity, which can make it harder to determine the individual effect of each predictor variable in a regression model.

- **const (14.97)**: The constant term typically has a high VIF because it's not a feature but is necessary for the regression model.
- **tenure (5.84)**: This suggests moderate multicollinearity with other features, but it's still acceptable since values below 10 are typically considered tolerable.
- **TotalCharges (9.53)**: This feature has a relatively high VIF, indicating it is somewhat collinear with other variables, particularly tenure, as both are related to customer longevity.
- **MonthlyCharges (3.23)**: This value is relatively low, indicating minimal multicollinearity.

3.5.5 Feature Importance

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

#Prepare data
X = pd.get_dummies(df.drop(columns=['Churn', 'customerID']), drop_first=True)
y = df['Churn'].map({'Yes': 1, 'No': 0})

# Decision Tree
model = DecisionTreeClassifier(random_state=42)
model.fit(X, y)

# Feature Importance
importance = pd.Series(model.feature_importances_, index=X.columns).sort_values(ascending=False)
print(importance)
```

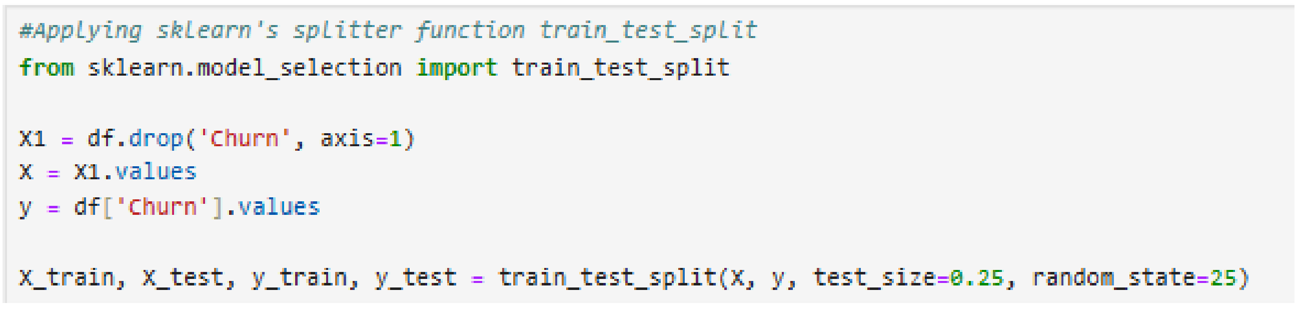
```
TotalCharges                              0.209149
tenure                                    0.206291
MonthlyCharges                            0.177939
InternetService_Fiber optic               0.107369
gender_Male                               0.025474
PaymentMethod_Electronic check            0.024623
Partner_Yes                               0.023731
SeniorCitizen                             0.020231
MultipleLines_Yes                         0.019450
Dependents_Yes                            0.019313
PaperlessBilling_Yes                      0.018392
OnlineSecurity_Yes                        0.018146
OnlineBackup_Yes                          0.014923
TechSupport_Yes                           0.014751
DeviceProtection_Yes                      0.014349
PaymentMethod_Mailed check                0.012833
Contract_One year                         0.012525
StreamingTV_Yes                           0.011575
PaymentMethod_Credit card (automatic)     0.011495
Contract_Two year                         0.010077
StreamingMovies_Yes                       0.009795
OnlineSecurity_No internet service        0.007045
TechSupport_No internet service           0.005004
PhoneService_Yes                          0.002785
MultipleLines_No phone service            0.002735
DeviceProtection_No internet service      0.000000
StreamingTV_No internet service           0.000000
OnlineBackup_No internet service          0.000000
StreamingMovies_No internet service       0.000000
InternetService_No                        0.000000
dtype: float64
```

This analysis highlights which customer attributes matter most for churn prediction and can inform strategies to target at-risk customers.

CHAPTER 4: MODEL BUILDING

4.1 Definition of model evaluation metrics:

Before building and testing the model, the dataset will be split into 75% training data and 25% test data. The "Churn" column will serve as the target variable (the "y"), while the other columns will be used as features (the "X").



```python
#Applying sklearn's splitter function train_test_split
from sklearn.model_selection import train_test_split

X1 = df.drop('Churn', axis=1)
X = X1.values
y = df['Churn'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=25)
```

To evaluate the performance of the selected models, various metrics are considered. Below is a breakdown of each metric used:

Metric	**Description**
Accuracy Score	Measures the overall correctness of the model's predictions. It calculates the ratio of correct predictions to total instances in the dataset. This is commonly used for classification models.
Confusion Matrix	A grid that compares the model's predictions with actual values, showing true positive, true negative, false positive, and false negative predictions. This helps assess how well the model classifies each class.
Feature Weights	Indicates the importance or contribution of each feature in the model's predictions. It highlights which features have the greatest impact on the target variable.
ROC Curve	Plots the true positive rate (TPR) against the false positive rate (FPR) at various thresholds, providing insight into the model's diagnostic ability.
AUC (Area Under the ROC Curve)	Quantifies the model's overall performance by measuring the area under the ROC curve. A higher AUC value signifies better model performance, indicating a higher probability of ranking a positive instance above a negative one.

```python
from sklearn.metrics import (precision_recall_curve, average_precision_score,
                             roc_curve, roc_auc_score, confusion_matrix,
                             accuracy_score, classification_report)
from sklearn.preprocessing import LabelEncoder
from mlxtend.plotting import plot_confusion_matrix

# Function to prepare the data
def prepare_data(X_train, X_test):
    X_train_df = pd.DataFrame(X_train)
    X_test_df = pd.DataFrame(X_test)

    # Convert categorical features to numeric (if any)
    X_train_encoded = pd.get_dummies(X_train_df)
    X_test_encoded = pd.get_dummies(X_test_df)

    # Align the train and test sets
    return X_train_encoded.align(X_test_encoded, join='left', axis=1, fill_value=0)
```

```python
# Function to plot Precision-Recall curve
def plot_precision_recall(y_true, y_scores, model_name):
    precision, recall, _ = precision_recall_curve(y_true, y_scores)
    average_precision = average_precision_score(y_true, y_scores)

    plt.figure(figsize=(8, 6))
    plt.plot(recall, precision, color='blue', label=f'AP = {average_precision:.2f}')
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title(f'Precision-Recall Curve - {model_name}')
    plt.legend(loc='lower left')
    plt.grid()
    plt.show()
```

```python
# Function to plot ROC curve
def plot_roc_curve(y_true, y_scores, model_name):
    fpr, tpr, _ = roc_curve(y_true, y_scores)
    roc_auc = roc_auc_score(y_true, y_scores)

    plt.figure(figsize=(8, 6))
    plt.plot(fpr, tpr, color='blue', label=f'ROC Curve (AUC = {roc_auc:.2f})')
    plt.plot([0, 1], [0, 1], color='red', linestyle='--')  # Diagonal line
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.0])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'ROC Curve - {model_name}')
    plt.legend(loc='lower right')
    plt.grid()
    plt.show()
```

```python
# Function to plot confusion matrix
def plot_confusion_matrix_func(y_true, y_pred, model_name):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    plot_confusion_matrix(conf_mat=cm, show_normed=True, colorbar=True)
    plt.title(f"Confusion Matrix - {model_name}")
    plt.xlabel("Predicted Labels")
    plt.ylabel("True Labels")
    plt.show()
```

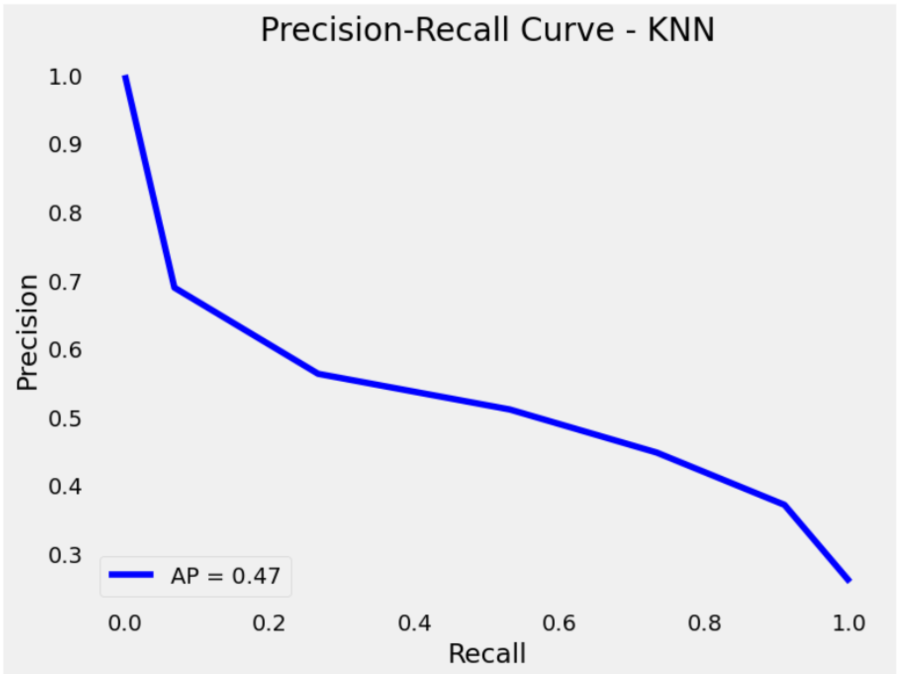
```python
# Main function for training and evaluating models
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test, model_name):
    # Prepare data
    X_train_encoded, X_test_encoded = prepare_data(X_train, X_test)

    # Train the model
    model.fit(X_train_encoded, y_train)

    # Predictions and probabilities
    y_pred_prob = model.predict_proba(X_test_encoded)[:, 1]  # Probabilities for the positive class

    # Convert y_test to binary format
    label_encoder = LabelEncoder()
    y_test_binary = label_encoder.fit_transform(y_test)

    # Plot Precision-Recall curve
    plot_precision_recall(y_test_binary, y_pred_prob, model_name)

    # Plot ROC curve
    plot_roc_curve(y_test_binary, y_pred_prob, model_name)

    # Make predictions for confusion matrix
    y_pred = model.predict(X_test_encoded)
    y_pred_binary = label_encoder.transform(y_pred)  # Convert predictions to binary format

    # Plot confusion matrix
    plot_confusion_matrix_func(y_test_binary, y_pred_binary, model_name)

    # Print accuracy and classification report
    accuracy = accuracy_score(y_test_binary, y_pred_binary)
    print(f"Accuracy ({model_name}): {accuracy:.4f}")
    print(f"Classification Report ({model_name}):")
    print(classification_report(y_test_binary, y_pred_binary))
```

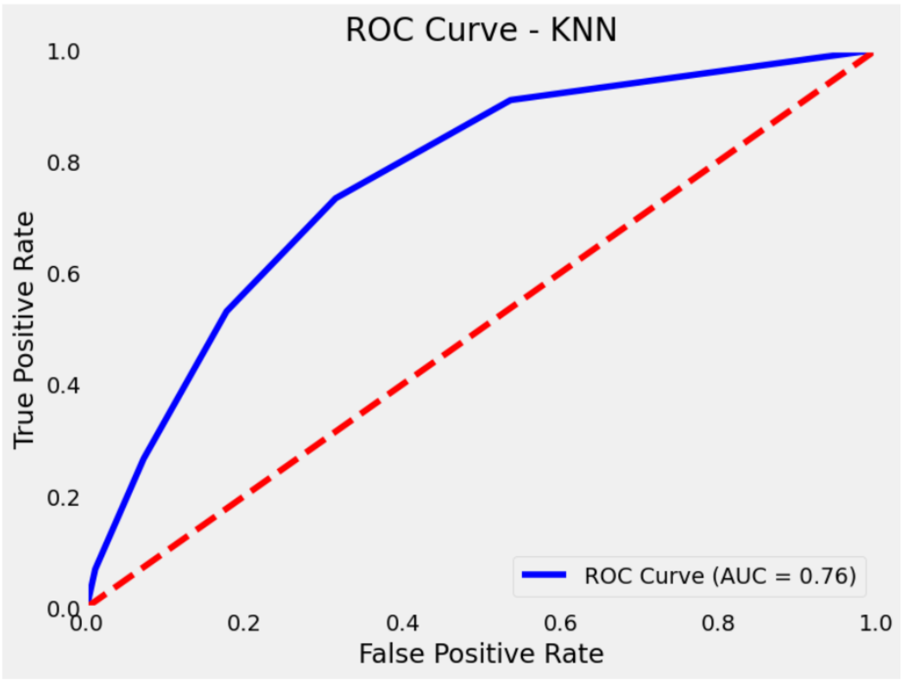
4.2 Model Selection, Prediction and Evaluation:

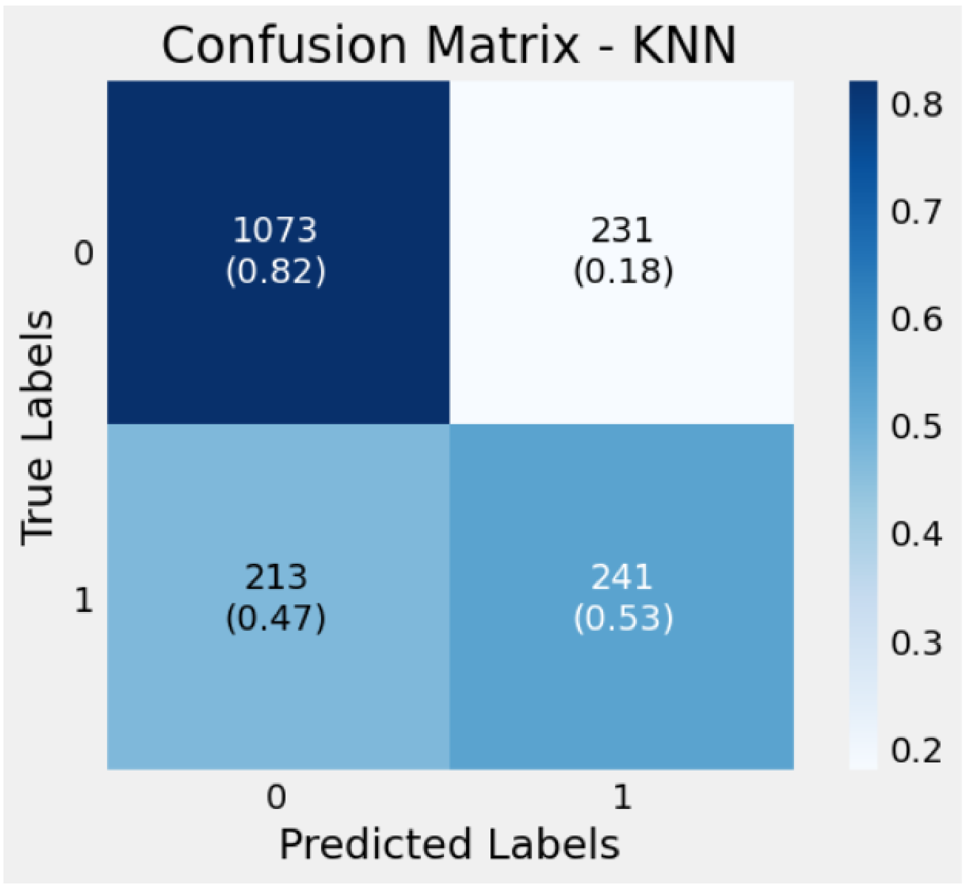
4.2.1 K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a non-parametric algorithm used for classification and regression by determining the majority vote of the nearest neighbors in the feature space. It's simple and effective for small datasets but can struggle with large or high-dimensional data due to its computational cost. KNN is sensitive to irrelevant features, making feature selection important for improving accuracy. The choice of **K**, the number of neighbors, is crucial—too few may lead to overfitting, while too many can cause underfitting. KNN performs best with small datasets and when appropriate distance metrics are selected, such as Euclidean distance

```
from sklearn.neighbors import KNeighborsClassifier

# Define and train the KNN model
knn = KNeighborsClassifier(n_neighbors=5)

# Train and evaluate KNN model
train_and_evaluate_model(knn, X_train, y_train, X_test, y_test, 'KNN')
```







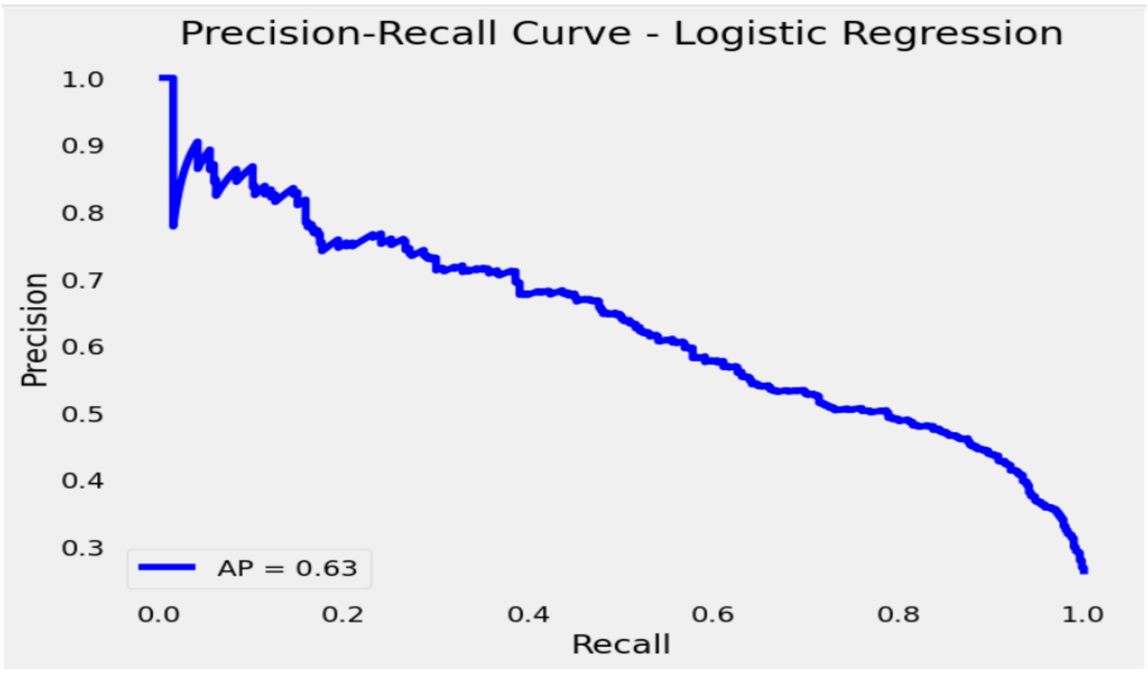
```
Accuracy (KNN): 0.7474
Classification Report (KNN):
              precision    recall  f1-score   support

           0       0.83      0.82      0.83      1304
           1       0.51      0.53      0.52       454

    accuracy                           0.75      1758
   macro avg       0.67      0.68      0.67      1758
weighted avg       0.75      0.75      0.75      1758
```

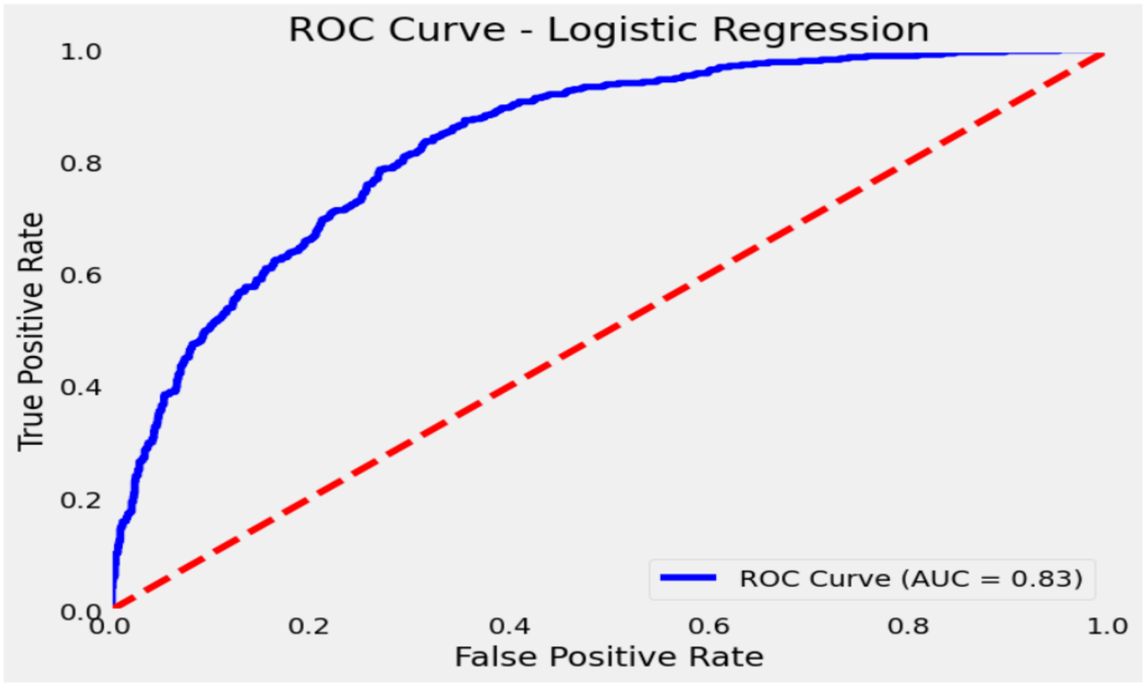
KNN performs reasonably well with a high precision and recall for class 0 (no churn), but the model struggles with class 1 (churn). The lower performance in class 1 (precision 0.51 and recall 0.53) suggests that KNN might not be capturing churn predictions as accurately.

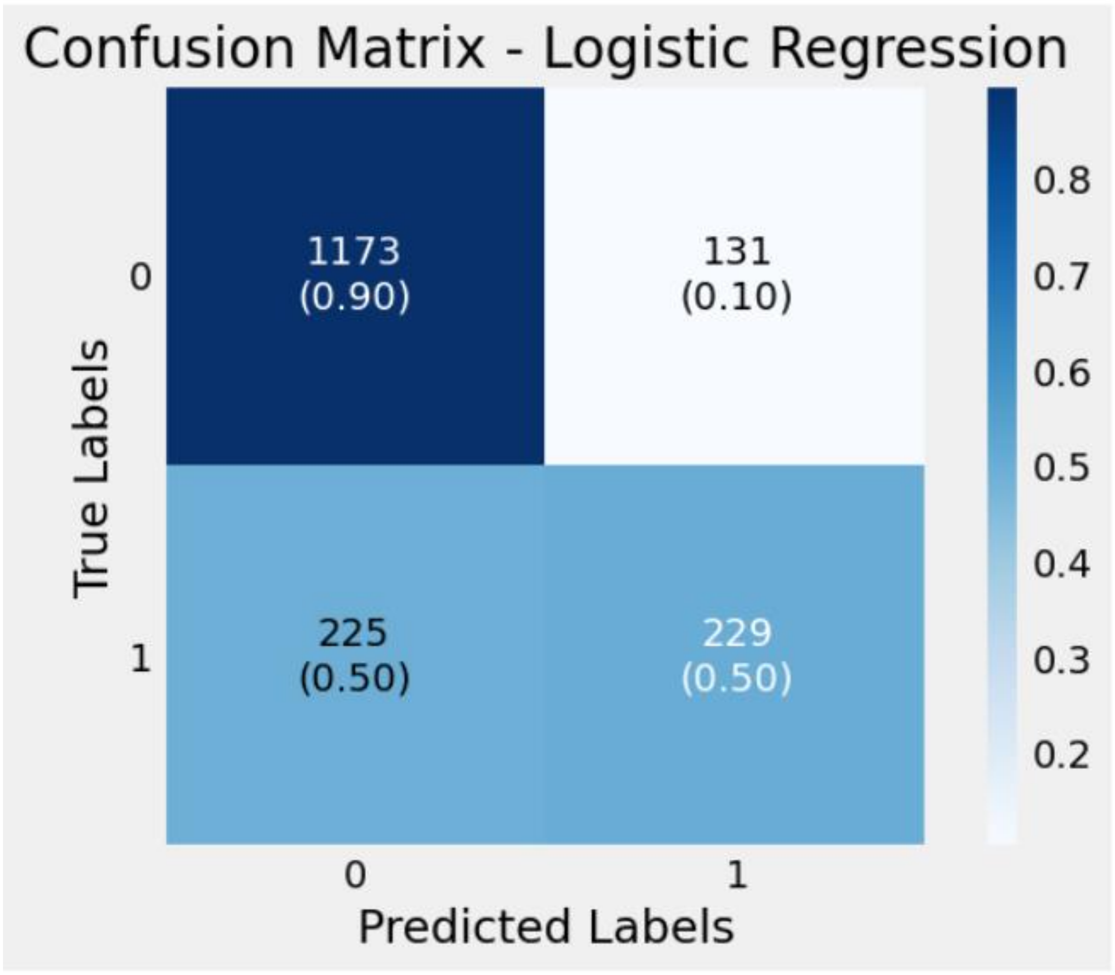
4.2.2 Logistic Regression

Logistic Regression is a statistical model commonly used for binary classification tasks. It models the relationship between input features and the probability of an event (e.g., churn) occurring by applying the logistic function, which produces an output between 0 and 1. This makes it ideal for predicting binary outcomes. Logistic Regression is interpretable, as it provides coefficients that represent the strength and direction of the relationship between each feature and the target variable. It works well when the relationship between features and the target is approximately linear, but may not perform as effectively when the relationship is complex or non-linear.

```python
from sklearn.linear_model import LogisticRegression

# Define and train the Logistic Regression model
logreg = LogisticRegression(max_iter=1000)

# Train and evaluate Logistic Regression model
train_and_evaluate_model(logreg, X_train, y_train, X_test, y_test, 'Logistic Regression')
```





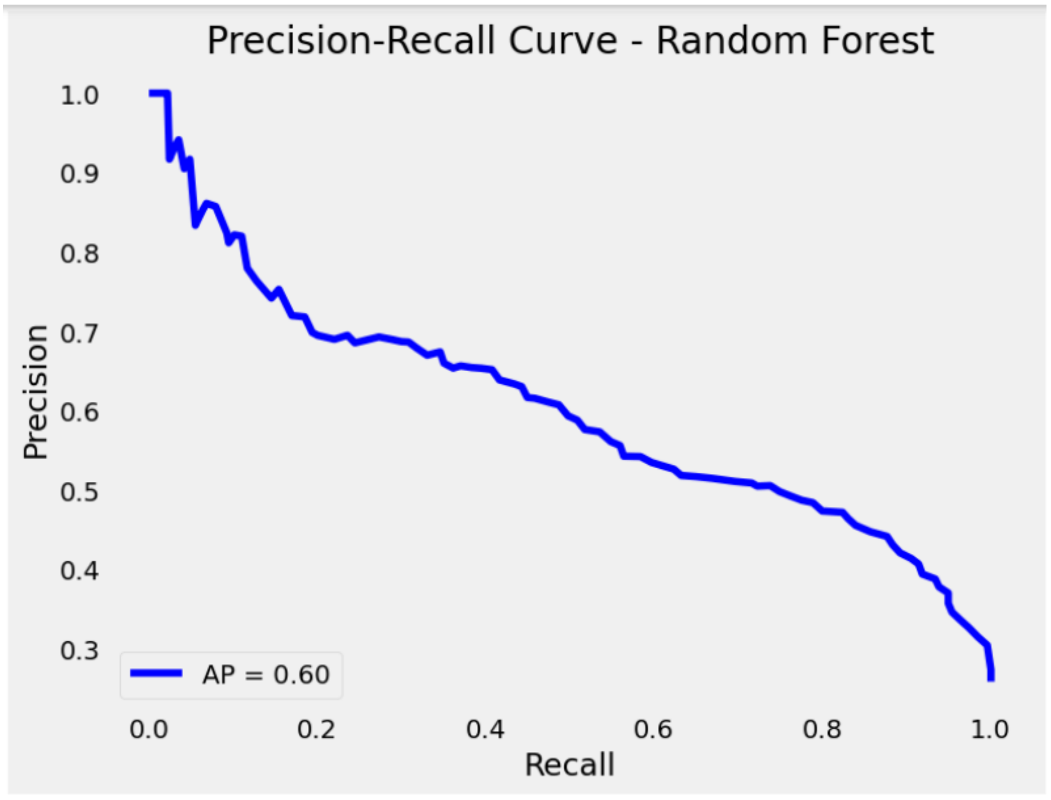


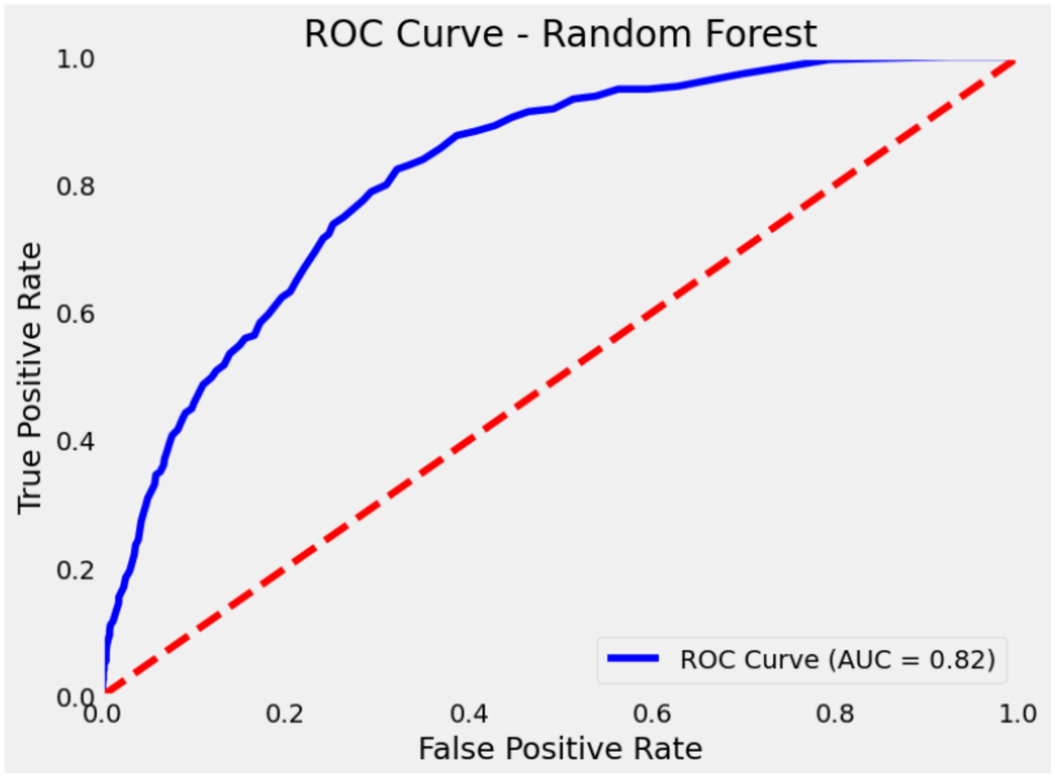
```
Accuracy (Logistic Regression): 0.7975
Classification Report (Logistic Regression):
              precision    recall  f1-score   support

           0       0.84      0.90      0.87      1304
           1       0.64      0.50      0.56       454

    accuracy                           0.80      1758
   macro avg       0.74      0.70      0.72      1758
weighted avg       0.79      0.80      0.79      1758
```

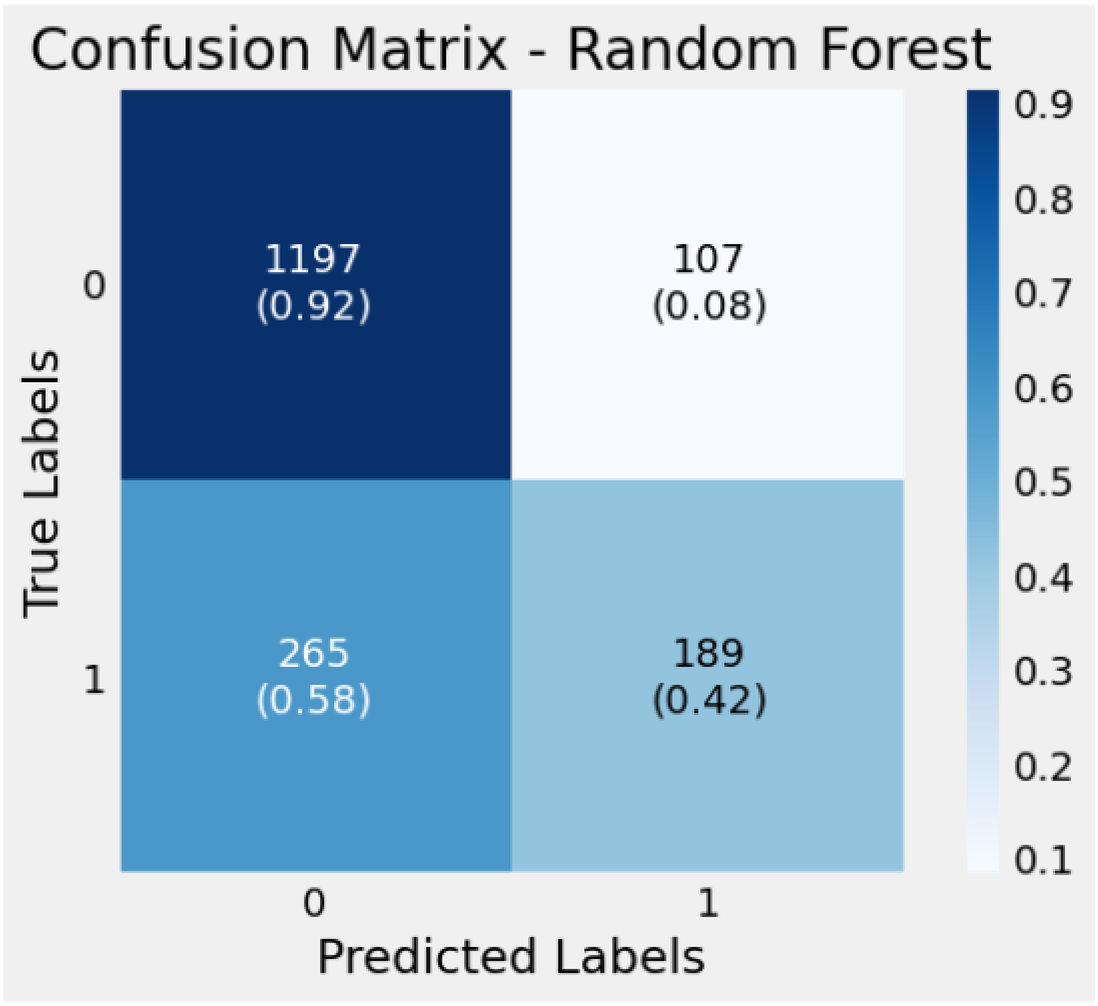
Logistic Regression has a higher accuracy (0.7975) compared to KNN, with better recall for class 0 (0.90). However, class 1 (churn) still suffers from lower recall (0.50), indicating the model isn't as effective at detecting churn.

4.2.3 Random Forest

Random Forest is an ensemble learning technique that combines multiple decision trees to make predictions. By averaging the predictions of many individual trees, it reduces the likelihood of overfitting compared to a single decision tree, making it more resilient to noise in the data. Random Forest is known for its ability to handle complex relationships in data and does not require as much feature engineering as some other models. It is highly effective for both classification and regression tasks, and its robustness often results in strong performance across a wide range of datasets.







```
Accuracy (Random Forest): 0.7884
Classification Report (Random Forest):
              precision    recall  f1-score   support

           0       0.82      0.92      0.87      1304
           1       0.64      0.42      0.50       454

    accuracy                           0.79      1758
   macro avg       0.73      0.67      0.68      1758
weighted avg       0.77      0.79      0.77      1758
```

Random Forest also has an accuracy of 0.7884, similar to Logistic Regression. It performs very well with class 0 (precision 0.82, recall 0.92) but struggles with class 1 (churn), similar to the other models, with a low recall of 0.42.

4.2.3 Model Selection:

Logistic Regression is the most suitable model in this case for several reasons:

- **Accuracy**: Logistic Regression has the highest accuracy (79.75%) compared to the other models. Accuracy is a straightforward metric that measures how well the model performs overall, and this model performs better than both K-Nearest Neighbors (74.74%) and Random Forest (78.84%).

- **Precision and Recall**:
  - **Precision** for class 0 (non-churn) is 0.84, which is comparable to the precision of the other models.
  - **Recall** for class 0 is 0.90, which is higher than that of both KNN (0.82) and Random Forest (0.92). This shows that Logistic Regression correctly identifies churned customers (class 0) more effectively.
  - For **class 1 (churned customers)**, Logistic Regression has a recall of 0.50, which is lower than that of KNN (0.53) and Random Forest (0.42). However, the overall performance still justifies Logistic Regression's selection because of the higher accuracy and balanced performance across classes.
- **F1-score**: The **weighted F1-score** of Logistic Regression is 0.79, the highest among the three models. The F1-score provides a balance between precision and recall and is crucial when dealing with class imbalance (which is the case here, since churned customers are less frequent).
- **Macro vs. Weighted Average**:
  - The **macro average** precision, recall, and F1-score for Logistic Regression (0.74, 0.70, 0.72) is also the most balanced among the three models, suggesting a well-rounded performance across both classes.

Given these points, **Logistic Regression** is the best choice as it provides a good balance of accuracy, precision, recall, and interpretability, making it more suitable for this task compared to KNN and Random Forest.

CHAPTER 5: CONCLUSION & RECOMMENDATION

5.1 CONCLUSION

The analysis of customer churn in the telecommunications sector revealed several key insights regarding the factors that influence customer retention. The most prominent drivers of churn were contract duration and customer tenure, where longer contracts and extended tenure significantly reduced churn likelihood. This aligns with industry practices, where fostering long-term relationships through contract commitments can mitigate churn rates. Additionally, the analysis found that high monthly payments and the number of services customers subscribed to were not as strongly linked to churn as initially expected. This suggests that churn behaviors are multifaceted and cannot be solely explained by financial commitments or service offerings.

Exploratory Data Analysis (EDA) provided a deeper understanding of customer behavior, shedding light on some assumptions that were not supported by the data, such as the role of senior citizens in churn rates and the influence of monthly payments on churn. The insights drawn from this analysis are crucial for refining churn prediction strategies and informing decision-making processes within the industry.

5.2 RECOMMENDATION

Given the insights gained, it is recommended that telecommunications companies focus on nurturing long-term customer relationships, especially by offering incentives for customers with long-term contracts and those who have been with the company for an extended period. These customers should be prioritized for retention campaigns, as they are less likely to churn, but may need reassurance through targeted loyalty programs.

Furthermore, companies should adopt a more nuanced approach in understanding churn by expanding their data analysis to include a wider range of customer behaviors, such as service usage patterns and customer service interactions. This data, when integrated into churn prediction models, can allow companies to intervene more proactively before churn occurs. Regular model updates with fresh data are also crucial to ensuring the accuracy and effectiveness of these retention strategies.

By combining data-driven insights with proactive customer service and tailored retention efforts, telecom companies can reduce churn, enhance customer satisfaction, and ultimately improve profitability.