

**ANAND INSTITUTE OF HIGHER
TECHNOLOGY OLD MAHABALIPURAM
ROAD, KALASALINGAM NAGAR,
KAZHIPATTUR – 603103**



**CUSTOMER CHURN PREDICTION
WITH
DATA ANALYTICS USING COGNOS**

PHASE – 3

NAME : MOUNIKA V
REG No. : 310121104065
BRANCH : COMPUTER SCIENCE & ENGINEERING
YEAR/SEM : III / V

IBM DATA ANALYTICS WITH **COGNOS**

TEAM NAME : Proj_229798_Team_1

PROJECT : 3101-Customer Churn
Prediction

TEAM MEMBERS :

1. Nokitha V M
2. Mounika V
3. Yasmeen U
4. Roshini

LEADER : Nokitha V M

CUSTOMER CHURN PREDICTION

INTRODUCTION

- Churn prediction is predicting which customers are at high risk of leaving your company or canceling a subscription to a service, based on their behavior with your product.
- To predict churn effectively, you'll want to synthesize and utilize key indicators defined by your team to signal when a customer has a probability of churning so that your company can take action.
- At a high level, predicting customer churn requires a detailed grasp of your clientele. Both qualitative and quantitative customer data are usually needed to start building an effective churn prediction model. To ensure that predictions aren't being made by arbitrary human guesses, these models are often built by a data scientist using machine learning.
- In a churn prediction model case, the target variable would be the indicator signifying whether a customer is likely to churn—(yes/no) or (1/0). To obtain this variable, you would need to use historical data of existing customers and previous customers that denotes whether this customer left or stayed; this could be a subscription cancellation, a closed contract, etc.



PHASE - 3 : DEVELOPMENT PART 1

- Start building the customer churn prediction using IBM Cognos for visualization. Define the analysis objectives and collect customer data from the source shared.
- Process and clean the collected data to ensure its quality and accuracy.
- This process involves collecting, cleaning, transforming, reduction of null values, visualization, scalability, efficiency and structuring raw data to make it suitable for analysis.
- Both qualitative and quantitative customer data are usually needed to start building an effective churn prediction model. To ensure that predictions aren't being made by arbitrary human guesses, these models are often built by a data scientist using machine learning.
- The goal of Customer Churn Prediction in this Phase3 is to prepare and begin building your project by loading and preprocessing the dataset.



DATA COLLECTION:

CUSTOMER CHURN PREDICTION is done by using the Dataset of “Telco Customer Churn” provided by the dataset site

www.kaggle.com



Dataset Link:

<https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

DATASET AND ITS DETAILS :

TITLE: CUSTOMER CHURN PREDICTION

LIBRARIES: sklearn, Matplotlib, pandas, seaborn, and NumPy

Context:

The dataset "CUSTOMER CHURN PREDICTION" on Kaggle is a collection of data related to the a Machine Learning Model That Can Predict Customers Who Will Leave The Company "Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs."

Content:

Each row represents a customer, each column contains customer's attributes described on the column Metadata.

The data set includes information about:

Gender -- Whether the customer is a male or a female

SeniorCitizen -- Whether a customer is a senior citizen or not

Partner -- Whether the customer has a partner or not (Yes, No)

Dependents -- Whether the customer has dependents or not (Yes, No)

Tenure -- Number of months the customer has stayed with the company

Phone Service -- Whether the customer has a phone service or not (Yes, No)

MultipleLines -- Whether the customer has multiple lines or not

InternetService -- Customer's internet service provider (DSL, Fiber Optic, No)

OnlineSecurity -- Whether the customer has online security or not (Yes, No, No Internet)

OnlineBackup -- Whether the customer has online backup or not (Yes, No, No Internet)

DeviceProtection -- Whether the customer has device protection or not (Yes, No, No internet service)

TechSupport -- Whether the customer has tech support or not (Yes, No, No internet)

StreamingTV -- Whether the customer has streaming TV or not (Yes, No, No internet service)

StreamingMovies -- Whether the customer has streaming movies or not (Yes, No, No Internet service)

Contract -- The contract term of the customer (Month-to-Month, One year, Two year)

PaperlessBilling -- Whether the customer has paperless billing or not (Yes, No)

Payment Method -- The customer's payment method (Electronic check, mailed check, Bank transfer(automatic), Credit card(automatic))

MonthlyCharges -- The amount charged to the customer monthly

TotalCharges -- The total amount charged to the customer

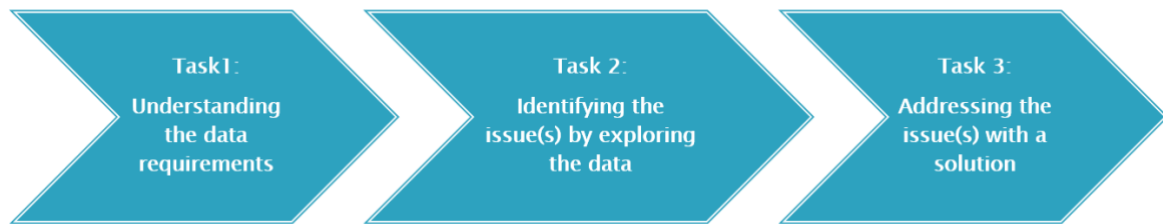
Churn -- Whether the customer churned or not (Yes or No)

SIGNIFICANCE OF LOADING AND PREPROCESSING THE DATASET:

Data Loading is defined as copying data from one electronic file or database into another. Data loading implies converting from one format into another; for example, from one type of production database into a decision support database from a different vendor.

Data preprocessing is essential before its actual use. Data preprocessing is the concept of changing the raw data into a clean data set. The dataset is preprocessed in order to check missing values, noisy data, and other inconsistencies before executing it to the algorithm.

Tasks under data preprocessing:



CHALLENGES INVOLVED IN LOADING AND PREPROCESSING:

1. Irrelevant data:

The most basic yet unavoidable issue that requires data cleaning is the presence of attributes in the data set that are irrelevant to the problem we are trying to solve.

2. Duplicate data:

Integrating data from different sources may result in redundant columns and rows in the data set.

3. Incorrect data type:

A data-set would generally store different types of data such as integer, float, and string. However, the type in which a data column is stored may be wrong.

4. Missing values:

It is seldom a case where values are not missing in a data set.

5. Outliers:

Outliers are extreme data points compared to the rest of the data. They can be detected numerically by calculating the data distribution's inter-quartile range and standard

deviation. They can also be visualized using box plots, histograms, and scatter plots.

6. Unacceptable format:

The data may be in a format that is not acceptable to the machine learning algorithm to be applied later in the data mining stage.

7. Too many categories:

In a categorical variable, there can be many categories. For example, the column storing the name of 'Locality', to which a person belongs, can lead to too many categories. This issue can be observed through descriptive statistics like a bar chart showing the frequency of data category-wise.

8. Class Imbalance: In customer churn prediction, it's common to have an imbalanced dataset, where the number of customers who stayed far exceeds those who left. This imbalance can lead to model bias, where the model may have difficulty in correctly identifying the minority class (churned customers).

Solution - Utilized models like Random Forest (with tuned hyperparameters) that are robust and less prone to overfitting, making the model more reliable when dealing with imbalanced datasets.

IMPORT AND LOAD THE DATASET :

Use Pandas to read the dataset file you downloaded into a DataFrame:

CODING AND ITS OUTPUT:

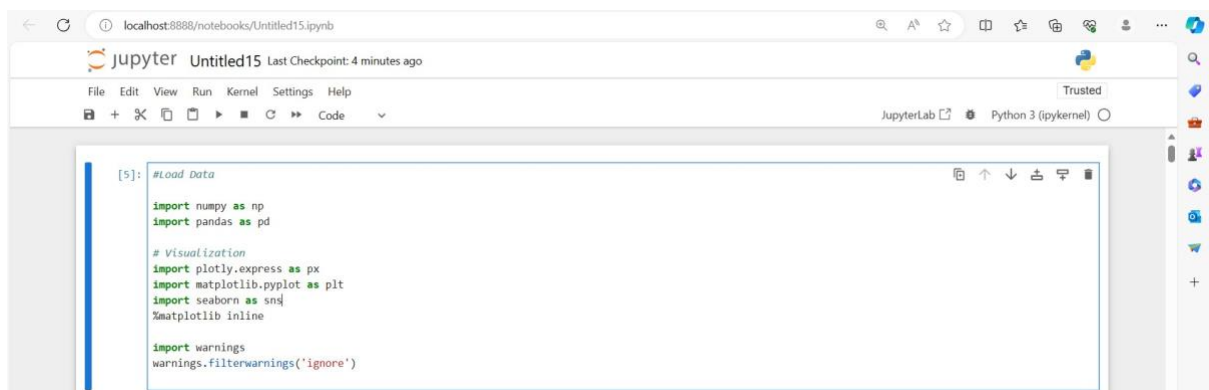
LOAD DATA:

```
import numpy as np
import pandas as pd
```

Visualization

```
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
import warnings
warnings.filterwarnings('ignore')
```



IMPORT DATA:

```
df = pd.read_csv(r"E:\nokitha nan mudhalvan\WA_Fn-UseC_-Telco-
Customer-Churn.csv")
df.head()
```

```
[4]: #Import Data
df = pd.read_csv(r"E:\nokitha nan mudhalvan\WA_Fn-UseC_-Telco-Customer-Churn.csv")
df.head()
```

```
[4]: e OnlineSecurity ... DeviceProtection TechSupport StreamingTV StreamingMovies Contract PaperlessBilling PaymentMethod MonthlyCharges TotalCharges Churn
L No ... No No No No Month-to-month Yes Electronic check 29.85 29.85 No
L Yes ... Yes No No No One year No Mailed check 56.95 1889.5 No
L Yes ... No No No No Month-to-month Yes Mailed check 53.85 108.15 Yes
L Yes ... Yes Yes No No One year No Bank transfer (automatic) 42.30 1840.75 No
c No ... No No No No Month-to-month Yes Electronic check 70.70 151.65 Yes
```

DATA UNDERSTANDING:

`df.info()`

```
[6]: #Data Understanding
df.info()

<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
```

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'],
errors='coerce')
```

Check the Duplicate:

```
print(df.duplicated().value_counts())
```

Check the missing values:

```
df.isnull().values.any()
```

```
[8]: #Check the Duplicate
print(df.duplicated().value_counts())

False    7043
Name: count, dtype: int64

[9]: #Check the missing Values
df.isnull().values.any()

[9]: True
```

Overview:

```
round(df.describe(include='all'),2)
```

```
[12]: #Overview
round(df.describe(include='all'),2)
```

| | e | OnlineSecurity | ... | DeviceProtection | TechSupport | StreamingTV | StreamingMovies | Contract | PaperlessBilling | PaymentMethod | MonthlyCharges | TotalCharges | Churn |
|---|------|----------------|------|------------------|-------------|-------------|-----------------|----------|------------------|---------------|----------------|--------------|-------|
| 3 | 7043 | ... | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043.00 | 7032.00 | 7043 |
| 3 | 3 | ... | 3 | 3 | 3 | 3 | 3 | 2 | 4 | NaN | NaN | 2 | |
| c | No | ... | No | No | No | No | Month-to-month | Yes | Electronic check | NaN | NaN | No | |
| 6 | 3498 | ... | 3095 | 3473 | 2810 | 2785 | 3875 | 4171 | 2365 | NaN | NaN | 5174 | |
| √ | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 64.76 | 2283.30 | NaN | |
| √ | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 30.09 | 2266.77 | NaN | |
| √ | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 18.25 | 18.80 | NaN | |
| √ | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 35.50 | 401.45 | NaN | |
| √ | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 70.35 | 1397.48 | NaN | |
| √ | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 89.85 | 3794.74 | NaN | |
| √ | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 118.75 | 8684.80 | NaN | |

Target value:

Count the oocurance of unique values in the 'Churn' Column

```
churn_counts = df.Churn.value_counts(normalize=True)
churn_counts
```

```
[15]: #Target Value
# Count the oocurance of unique values in the 'Churn' Column

churn_counts = df.Churn.value_counts(normalize=True)
churn_counts

[15]: Churn
No    0.73463
Yes   0.26537
Name: proportion, dtype: float64
```

Calculate the percentage of 'Yes' and 'No' label

```
total_count = churn_counts.sum()
```

```
percentage_yes = (churn_counts['Yes']/ total_count) * 100
```

```
percentage_no = (churn_counts['No']/ total_count) *100
```

```
# Plot the target value
```

```
ax = churn_counts.plot(kind='bar')
```

```
# Annotate the bars with percentages
```

```
for i, count in enumerate(churn_counts):
```

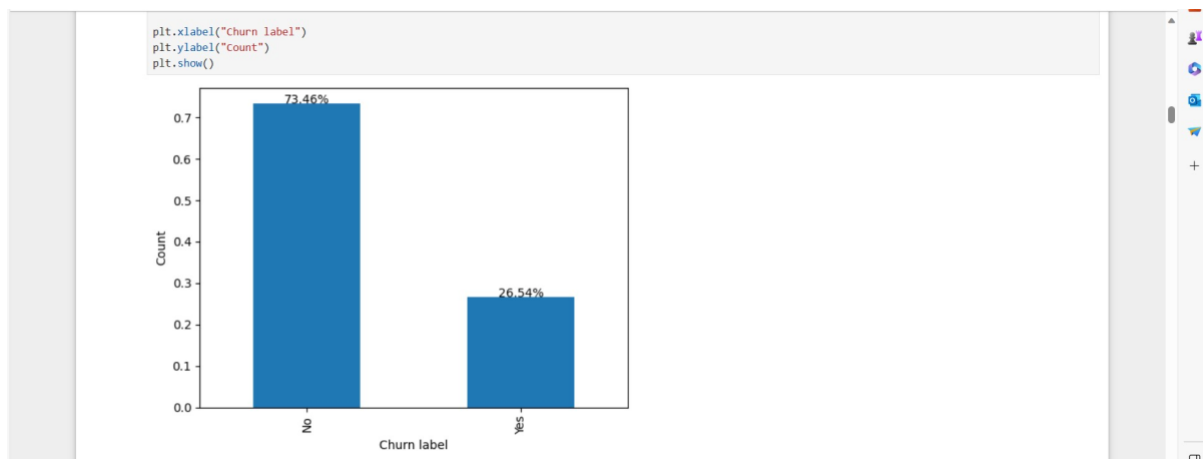
```
    percentage = percentage_yes if i == 1 else percentage_no
```

```
    ax.annotate(f'{percentage:.2f}%', xy=(i, count), ha='center')
```

```
plt.xlabel("Churn label")
```

```
plt.ylabel("Count")
```

```
plt.show()
```



EDA - Exploratory Data Analysis on each feature:

It is observed that the dataset exhibits a significant class imbalance, with a larger amount of data representing non-churners.

1 . Is there a correlation between churn and factors such as monthly charges and total charges?

```
df[['MonthlyCharges','TotalCharges']]
```

```
df.groupby(['MonthlyCharges','TotalCharges'])['Churn'].size()
```

```
sns.kdeplot(data=df,  
x="MonthlyCharges",hue="Churn",multiple="stack")
```

```
# Customize the plot appearance
```

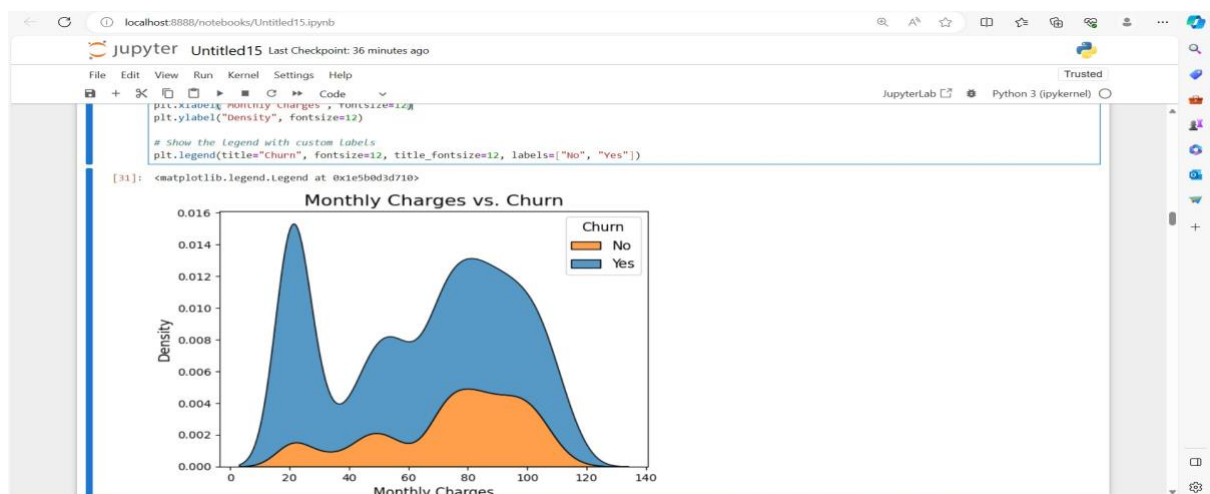
```
plt.title("Monthly Charges vs. Churn", fontsize=16)
```

```
plt.xlabel("Monthly Charges", fontsize=12)
```

```
plt.ylabel("Density", fontsize=12)
```

```
# Show the legend with custom labels
```

```
plt.legend(title="Churn", fontsize=12, title_fontsize=12, labels=["No",  
"Yes"])
```



- Note: it's noticeable that as monthly charges increase within the range of 60 to 120, the density also rises. This trend indicates a higher rate of churn as monthly charges increase.

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'],
errors='coerce')
```

```
df['TotalCharges']
```

```
print(df['TotalCharges'].dtype)
```



```
[34]: df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df['TotalCharges']

[34]: 0      29.85
1    1889.50
2     108.15
3    1840.75
4     151.65
...
7038  1990.50
7039  7362.90
7040   346.45
7041   306.60
7042  6844.50
Name: TotalCharges, Length: 7043, dtype: float64

[35]: print(df['TotalCharges'].dtype)
float64
```

```
sns.kdeplot(data=df,
x="TotalCharges",hue="Churn",multiple="stack")
```

```
# Customize the plot appearance
```

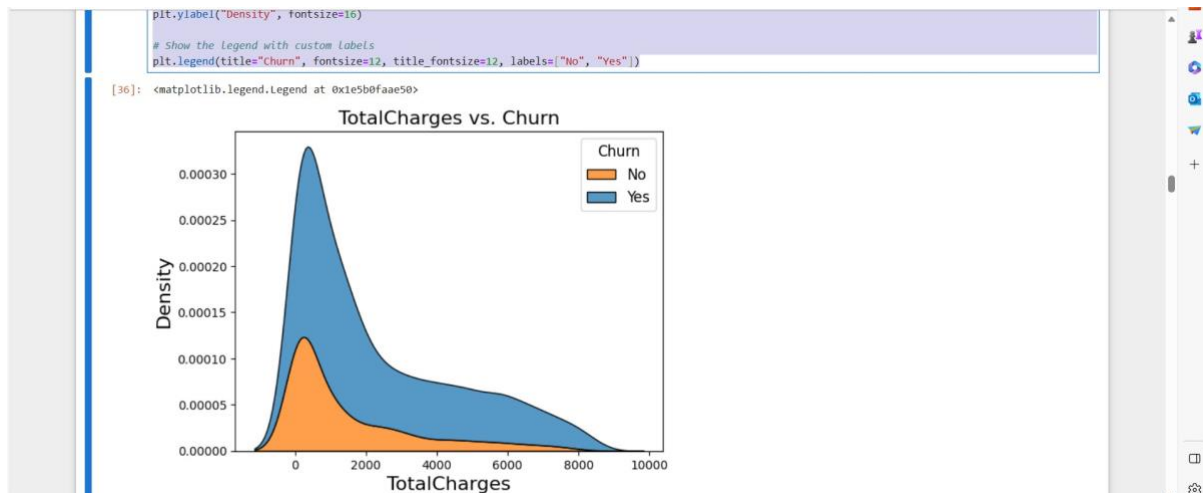
```
plt.title("TotalCharges vs. Churn", fontsize=16)
```

```
plt.xlabel("TotalCharges", fontsize=16)
```

```
plt.ylabel("Density", fontsize=16)
```

```
# Show the legend with custom labels
```

```
plt.legend(title="Churn", fontsize=12, title_fontsize=12, labels=["No",
"Yes"])
```



Note: High churn rates are associated with lower total charges, with the highest churning occurring in the 0-2000 total charges range.

2. How does the length of a customer's tenure with the company influence their likelihood of churning?

```
grouped_data = df.groupby(['tenure',
'Churn']).size().reset_index(name='count')
```

```
grouped_data[20:30]
```

```
[37]: # How does the length of a customer's tenure with the company influence their Likelihood of churning?
grouped_data = df.groupby(['tenure', 'Churn']).size().reset_index(name='count')
grouped_data[20:30]
```

[37]:

| | tenure | Churn | count |
|----|--------|-------|-------|
| 20 | 10 | Yes | 45 |
| 21 | 11 | No | 68 |
| 22 | 11 | Yes | 31 |
| 23 | 12 | No | 79 |
| 24 | 12 | Yes | 38 |
| 25 | 13 | No | 71 |
| 26 | 13 | Yes | 38 |
| 27 | 14 | No | 52 |
| 28 | 14 | Yes | 24 |
| 29 | 15 | No | 62 |

```
grouped_data = df.groupby(['tenure',
'Churn']).size().reset_index(name='count')
```

```
grouped_data[30:40]
```



```
# Separate churn and non-churn counts
```

```
churn_data = grouped_data[grouped_data['Churn'] == 'Yes']
```

```
non_churn_data = grouped_data[grouped_data['Churn'] == 'No']
```

```
# Create a line chart for churn and non-churn counts
```

```
plt.figure(figsize=(5, 4))
```

```
plt.plot(churn_data['tenure'], churn_data['count'], label='Churn',  
marker='*')
```

```
plt.plot(non_churn_data['tenure'], non_churn_data['count'],  
label='Non-Churn', marker='.')
```

```
plt.xlabel('Tenure')
```

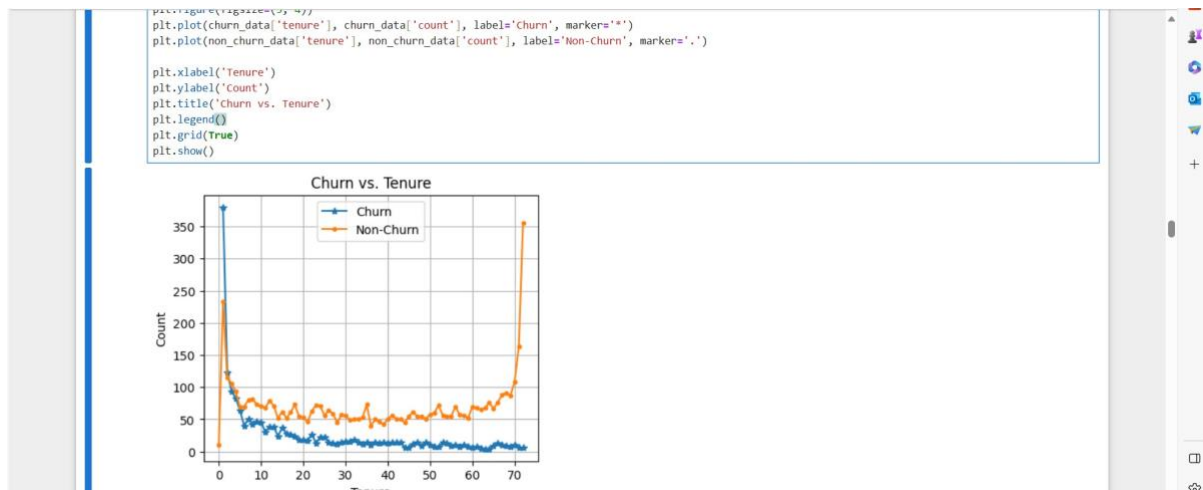
```
plt.ylabel('Count')
```

```
plt.title('Churn vs. Tenure')
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```



Note:

Customers with the low tenure "**0-10**" has the highest rate of churning, these range can be crucial for business decisions.

In general the **Non-Churn** line has small fluctuation and remains **relatively stable**, with longer tenure tend to stay with the company.

The Churn line decreases or remains relatively stable as tenure increases, it suggests that **customer loyalty** increases with longer tenure

3. connection between gender, partner status, and churn

```
df.gender.value_counts(normalize=True)
```

```
# Create the DataFrame
```

```
df_grouped = df.groupby(['gender', 'Partner', 'Dependents',  
'Churn']).size().reset_index(name='Count')
```

```
df_grouped.head()
```

```
[45]: #connection between gender, partner status, and churn
df.gender.value_counts(normalize=True)

[45]: gender
Male    0.584756
Female  0.415244
Name: proportion, dtype: float64

[46]: # Create the DataFrame
df_grouped = df.groupby(['gender', 'Partner', 'Dependents', 'Churn']).size().reset_index(name='Count')
df_grouped.head()
```

| | gender | Partner | Dependents | Churn | Count |
|---|--------|---------|------------|-------|-------|
| 0 | Female | No | No | No | 1068 |
| 1 | Female | No | No | Yes | 587 |
| 2 | Female | No | Yes | No | 112 |
| 3 | Female | No | Yes | Yes | 33 |
| 4 | Female | Yes | No | No | 618 |

```
# Create the DataFrame
```

```
df_grouped = df.groupby(['gender', 'Partner', 'Dependents',
'Churn']).size().reset_index(name='Count')
```

```
# Create a subplot with two subplots (one for Dependents and one
for Partner)
```

```
fig = px.bar(df_grouped, x='Count', y='Churn', color='Dependents',
facet_col='Partner',
```

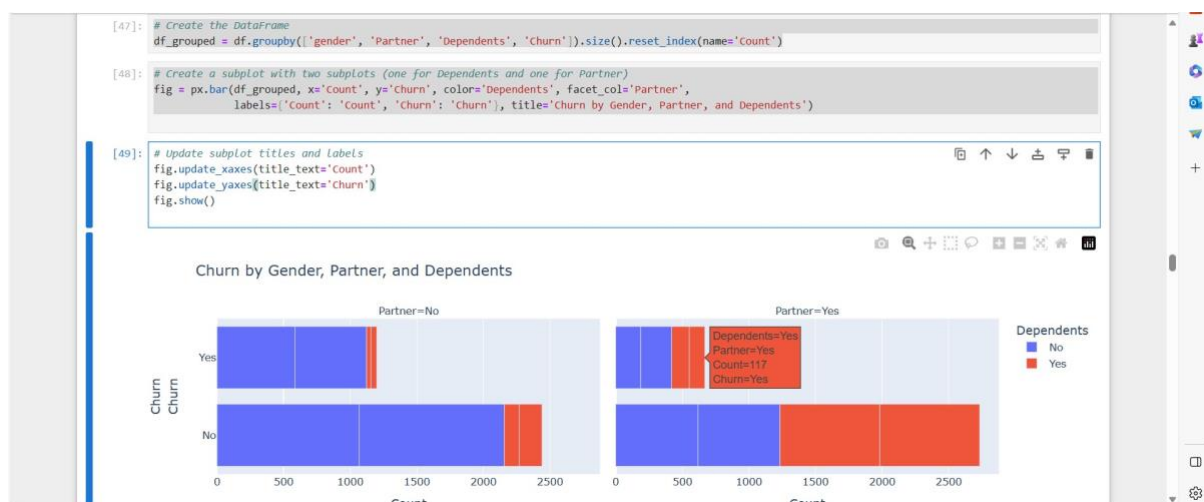
```
labels={'Count': 'Count', 'Churn': 'Churn'}, title='Churn by
Gender, Partner, and Dependents')
```

```
# Update subplot titles and labels
```

```
fig.update_xaxes(title_text='Count')
```

```
fig.update_yaxes(title_text='Churn')
```

```
fig.show()
```



Note:

Regardless of gender, individuals who lack a partner or dependents are at a higher risk of churning.

4. Does the availability of technical support play a role in influencing customer churn? and does the duration of being customer

```
df.TechSupport.value_counts()
```

```
# Create two dataframe for TechSupport categories
```

```
Receive_techSup = df[df['TechSupport'] == 'Yes']
```

```
Receive_techSup.head(2)
```

```
[50]: # Does the availability of technical support play a role in influencing customer churn? and does the duration of being customer
df.TechSupport.value_counts()

[50]: TechSupport
No      3473
Yes     2044
No internet service  1526
Name: count, dtype: int64

[51]: # create two dataframe for TechSupport categories
Receive_techSup = df[df['TechSupport'] == 'Yes']
Receive_techSup.head(2)
```

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... | DeviceProtection | TechSupport | Stu |
|---|------------|--------|---------------|---------|------------|--------|--------------|------------------|-----------------|----------------|-----|------------------|-------------|-----|
| 3 | 7795-CFOCW | Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | ... | Yes | Yes | |
| 8 | 7892-POOKP | Female | 0 | Yes | No | 28 | Yes | Yes | Fiber optic | No | ... | Yes | Yes | |

2 rows x 21 columns

```
NotReceive_techSup = df[df['TechSupport'] == 'No']
```

```
NotReceive_techSup.head(2)
```

```
[52]: NotReceive_techSup = df[df['TechSupport'] == 'No']
NotReceive_techSup.head(2)
```

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... | DeviceProtection | TechSupport | Stu |
|---|------------|--------|---------------|---------|------------|--------|--------------|------------------|-----------------|----------------|-----|------------------|-------------|-----|
| 0 | 7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | ... | No | No | |
| 1 | 5575-GNVDE | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | ... | Yes | No | |

2 rows x 21 columns

```
# Create subplots to compare the distribution

fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# Plot the distribution of 'TechSupport' for 'Yes' category

sns.countplot(data=Receive_techSup, x='TechSupport',
hue='Churn', ax=axes[0])

axes[0].set_title('TechSupport = Yes')

axes[0].set_xlabel('TechSupport')

axes[0].set_ylabel('Count')

# Plot the distribution of 'TechSupport' for 'No' category

sns.countplot(data=NotReceive_techSup, x='TechSupport',
hue='Churn', ax=axes[1])

axes[1].set_title('TechSupport = No')

axes[1].set_xlabel('TechSupport')

axes[1].set_ylabel('Count')

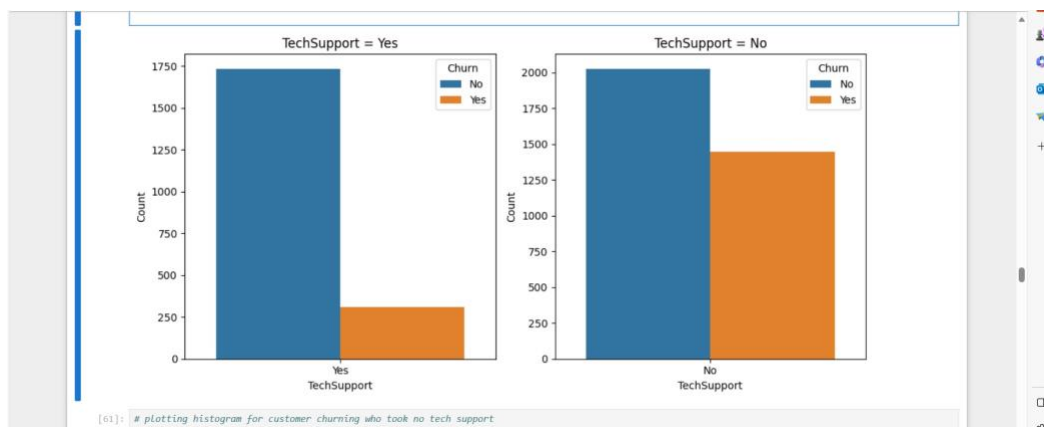
# Adjust layout

plt.tight_layout()

plt.show()
```

Note:

Customers who do not receive tech support are more likely to churn.



plotting histogram for customer churning who took no tech support

```
fig = px.histogram(NotReceive_techSup.groupby(['tenure',
'Churn']).size().reset_index(name='count'),

                    x='tenure', y='count',color='Churn', marginal='rug',
                    color_discrete_map={"Yes":"Green", "No":"Yellow"},

                    title="Customers NOT Receive tech Support")

fig.show()
```



```
# plotting histogram for customer churning who took no tech support
```

```
fig = px.histogram(Receive_techSup.groupby(['tenure',  
'Churn']).size().reset_index(name='count'),  
                  x='tenure', y='count', color='Churn', marginal='rug',  
                  color_discrete_map={"Yes": "Green", "No": "Yellow"},  
                  title="Customers NOT Receive tech Support")
```

```
fig.show()
```



Note:

The data indicates that churn rates are highest within the first year of service, especially among customers without tech support. Conversely, longer tenure is associated with increased customer loyalty.

5 .Which aspect of the contract has the most significant impact on the business?

```
df.Contract.value_counts(normalize =True)
```

```
Contract_condition =
```

```
df.groupby(['Churn','Contract']).size().reset_index(name='count')
```

```
Contract_condition
```

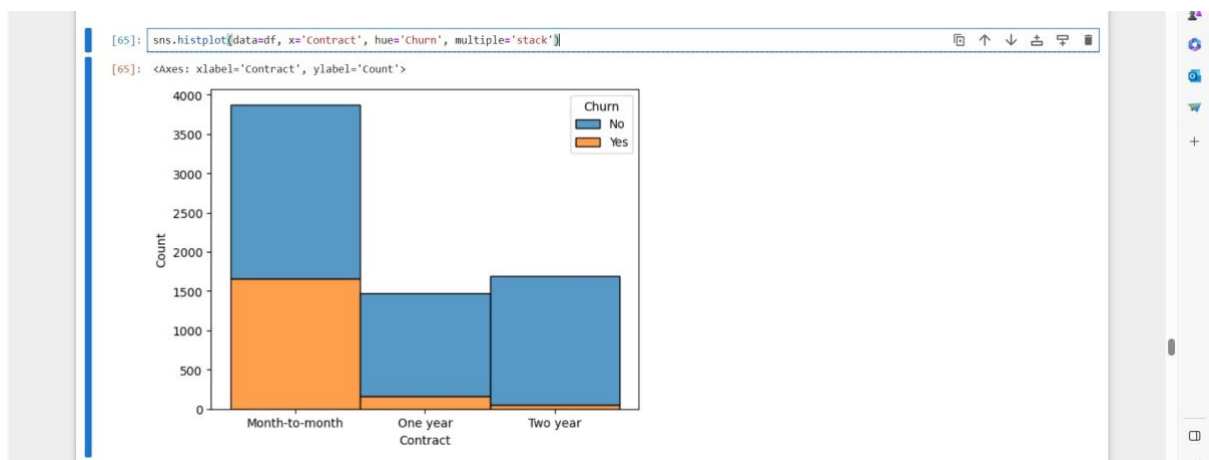
```
[63]: # Which aspect of the contract has the most significant impact on the business?
      | df.Contract.value_counts(normalize=True)

[63]: Contract
      Month-to-month    0.550192
      Two year         0.240664
      One year         0.209144
      Name: proportion, dtype: float64

[64]: Contract_condition = df.groupby(['Churn', 'Contract']).size().reset_index(name='count')
      Contract_condition

[64]:   Churn    Contract  count
0     No  Month-to-month  2220
1     No    One year    1307
2     No    Two year    1647
3     Yes  Month-to-month  1655
4     Yes    One year     166
5     Yes    Two year      48
```

`sns.histplot(data=df, x='Contract', hue='Churn', multiple='stack')`



Note:

It is evident that customers with month-to-month contracts have the highest churn rates.

6 .How does the quality of service differ for customers who have opted for streaming services?

`df.StreamingTV.value_counts()`

```
Streaming = df.groupby
([ 'Churn', 'StreamingTV']).size().reset_index(name='count')
Streaming
```



```
[66]: # How does the quality of service differ for customers who have opted for streaming services?
df.StreamingTV.value_counts()
```

```
[66]: StreamingTV
No      2810
Yes     2707
No internet service 1526
Name: count, dtype: int64
```

```
[67]: Streaming = df.groupby(['Churn', 'StreamingTV']).size().reset_index(name='count')
Streaming
```

```
[67]:   Churn  StreamingTV  count
0     No           No    1868
1     No  No internet service 1413
2     No           Yes    1893
3     Yes           No    942
4     Yes  No internet service   113
5     Yes           Yes    814
```

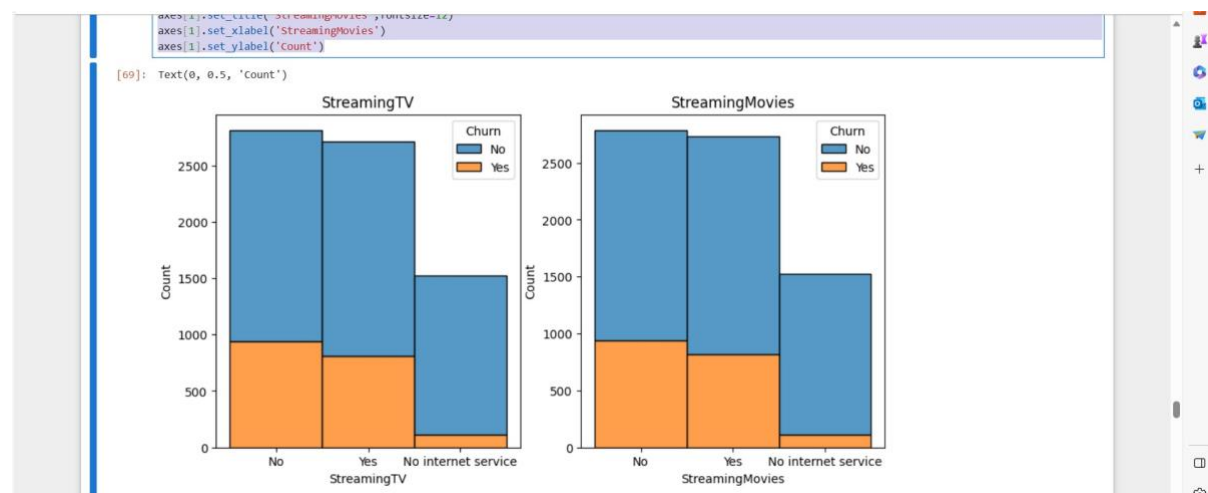
ig, axes = plt.subplots(1,2,figsize = (10,5))

Plot the distribution of 'StreamingTV'

```
sns.histplot(data=df, x='StreamingTV', hue='Churn',
multiple='stack',ax=axes[0])
axes[0].set_title('StreamingTV',fontsize=12)
axes[0].set_xlabel('StreamingTV')
axes[0].set_ylabel('Count')
```

Plot the distribution of 'StreamingMovies'

```
sns.histplot(data=df, x='StreamingMovies', hue='Churn',
multiple='stack',ax=axes[1])
axes[1].set_title('StreamingMovies',fontsize=12)
axes[1].set_xlabel('StreamingMovies')
axes[1].set_ylabel('Count')
```



Note: Churn rates are similar for both the 'Yes' and 'No' groups in terms of whether customers are connected to StreamingTV and StreamingMovies.

7. Given that the dataset pertains to the telecom industry, what insights can we uncover regarding phone and internet services?

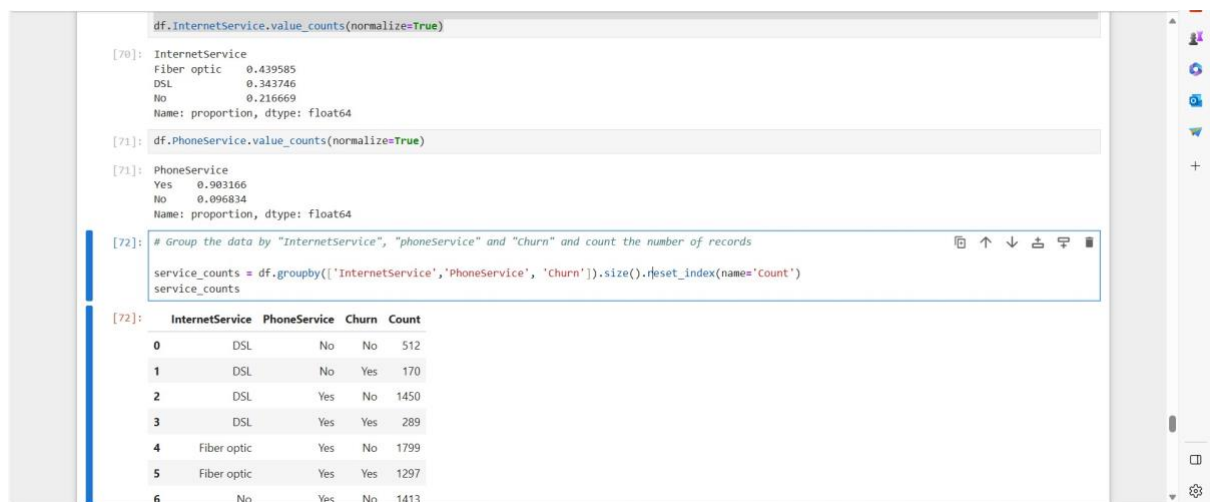
```
df.InternetService.value_counts(normalize=True)
```

```
df.PhoneService.value_counts(normalize=True)
```

```
# Group the data by "InternetService", "phoneService" and "Churn" and count the number of records
```

```
service_counts = df.groupby(['InternetService', 'PhoneService', 'Churn']).size().reset_index(name='Count')
```

```
service_counts
```



```
df.InternetService.value_counts(normalize=True)
```

```
[70]: InternetService
      Fiber optic    0.439585
      DSL           0.343746
      No            0.216669
      Name: proportion, dtype: float64
```

```
[71]: df.PhoneService.value_counts(normalize=True)
```

```
[71]: PhoneService
      Yes    0.903166
      No     0.096834
      Name: proportion, dtype: float64
```

```
[72]: # Group the data by "InternetService", "phoneService" and "churn" and count the number of records
      service_counts = df.groupby(['InternetService', 'PhoneService', 'Churn']).size().reset_index(name='Count')
      service_counts
```

| | InternetService | PhoneService | Churn | Count |
|---|-----------------|--------------|-------|-------|
| 0 | DSL | No | No | 512 |
| 1 | DSL | No | Yes | 170 |
| 2 | DSL | Yes | No | 1450 |
| 3 | DSL | Yes | Yes | 289 |
| 4 | Fiber optic | Yes | No | 1799 |
| 5 | Fiber optic | Yes | Yes | 1297 |
| 6 | No | Yes | No | 1413 |

```
plt.figure(figsize=(10, 5))
```

```
# Create two subplots
```

```
plt.subplot(121)
```

1 row, 2 columns, the first plot for Phone Service

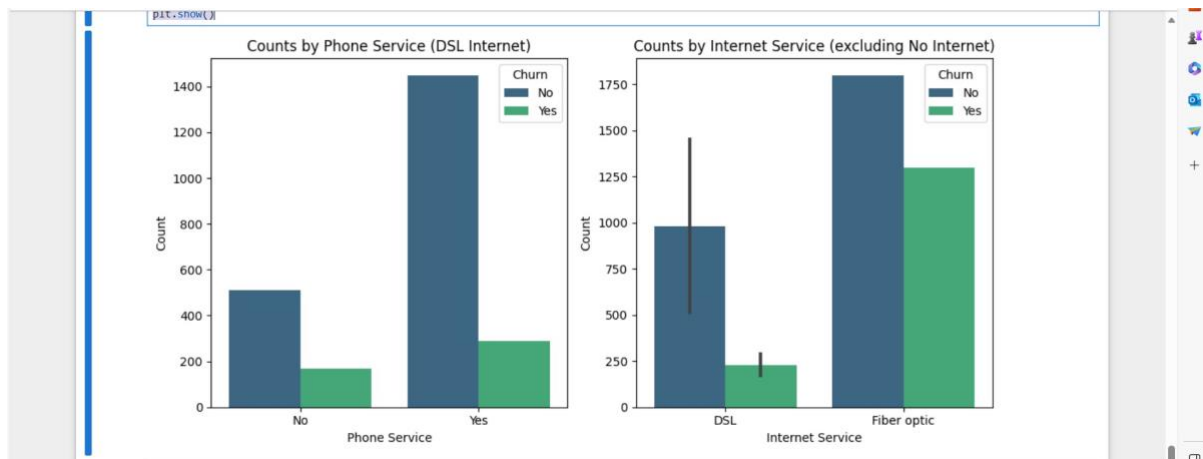
```
sns.barplot(data=service_counts[service_counts['InternetService'] ==  
'DSL'], x='PhoneService', y='Count', hue='Churn', palette='viridis')  
plt.xlabel('Phone Service')  
plt.ylabel('Count')  
plt.title('Counts by Phone Service (DSL Internet)')
```

```
plt.subplot(122)
```

1 row, 2 columns, the second plot for Internet Service

```
sns.barplot(data=service_counts[service_counts['InternetService'] !=  
'No'], x='InternetService', y='Count', hue='Churn', palette='viridis')  
plt.xlabel('Internet Service')  
plt.ylabel('Count')  
plt.title('Counts by Internet Service (excluding No Internet)')
```

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



Note:

The plot suggests that the customers with phone service are loyal and have not churned. (the blue bar is significantly taller than green).

For customers without phone service, the blue bar for non-churn is still higher than the green bar. This indicates that even among those without phone service, a significant portion has not churned, and the absence of phone service is associated with a lower churn rate.

In summary, the chart illustrates that customers with phone service, as well as some customers without phone service, are less likely to churn. This suggests that both groups have a relatively lower churn rate, with a particularly strong retention rate among customers with phone service.

8. Does the PaymentMethod has an impact on churn?

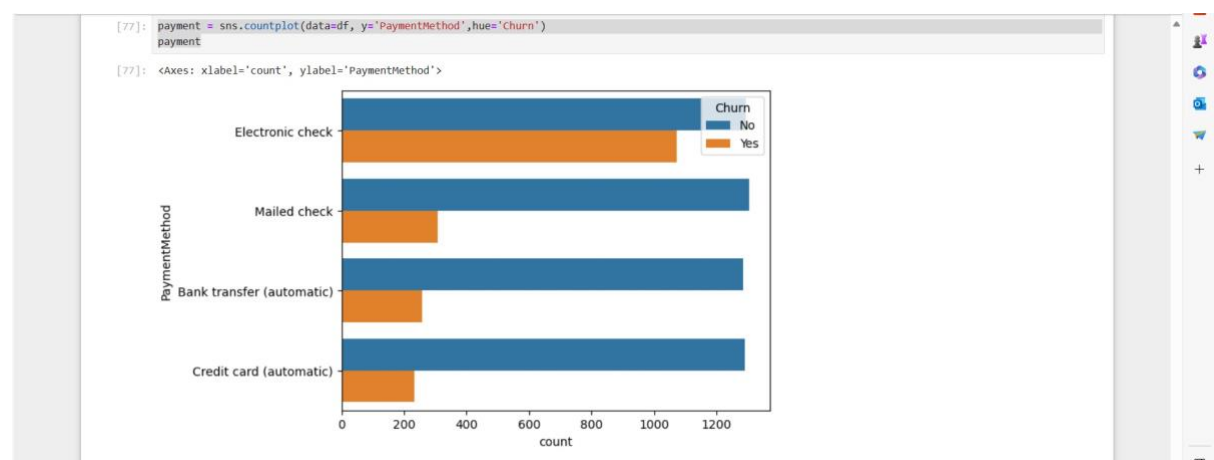
```
df.PaymentMethod.value_counts(normalize=True)
```

```
[76]: # Does the PaymentMethod has an impact on churn?
df.PaymentMethod.value_counts(normalize=True)

[76]: PaymentMethod
Electronic check    0.335794
Mailed check       0.228880
Bank transfer (automatic)  0.219225
Credit card (automatic)  0.216101
Name: proportion, dtype: float64
```

```
payment = sns.countplot(data=df, y='PaymentMethod', hue='Churn')
```

```
payment
```



CONCLUSION:

EDA Outcome:

Observation:

- The data reveals that the highest churn rate occurs within the first 0-10 years of customer tenure, suggesting that customers are less inclined to switch telecom providers as they become more familiar with their current one.
- Churning tends to increase with rising monthly charges, particularly in the 60-120 range, indicating a correlation between higher charges and increased churn.
- In contrast to monthly charges, the most significant churning occurs in the early phases, with the 0-2000 total charges bracket experiencing the highest churn rate.
- Regardless of gender, customers without partners or dependents are more likely to churn.
- Churning is more prevalent within the first 10 years of tenure for customers with or without tech support, with a higher rate among those without tech support.
- Customers who have both Phone services (yes) and 'Fiber optic' Internet Service are more prone to churn.
- The presence of StreamingTV, whether 'Yes' or 'No,' does not significantly affect the rate of churn; it appears to be relatively consistent
- The availability of StreamingMovies (either 'Yes' or 'No') does not appear to have a substantial impact on churn.
- Customers with month-to-month contracts are the most frequent churners, highlighting the importance of contract type in predicting churn.

- Senior citizens exhibit a lower likelihood of churning compared to non-senior citizens. It's worth noting that the dataset comprises a significantly higher proportion of non-senior citizens at a 5:1 ratio.