

```
from google.colab import drive
drive.mount('/content/drive')
```

↗ Mounted at /content/drive

```
!pip install pandas
!pip install matplotlib
!pip install seaborn
!pip install numpy
```

↗ Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages  
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-p  
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-pa  
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packag  
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-pack  
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dis  
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.11/dist-pa  
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/di  
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/di  
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.11/dist-pac  
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist  
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packa  
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dis  
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packag  
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-package  
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/python3.11  
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.11/dist-pac  
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/local/lib/python3  
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dis  
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.11/dist-pa  
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/di  
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/di  
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist  
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packa  
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dis  
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-pa  
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packag  
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages

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

```
df = pd.read_csv('/content/drive/MyDrive/Customer churn dataset.csv')
df.head()
```

df.head()



	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	M
0	7590-VHVEG	Female	0	Yes	No	1	No	N
1	5575-GNVDE	Male	0	No	No	34	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	
3	7795-CFOCW	Male	0	No	No	45	No	N
4	9237-HQITU	Female	0	No	No	2	Yes	

5 rows × 21 columns

**#replacing blanks with 0 as tenure is 0 and no total charges are recorded**

```
df["TotalCharges"]=df["TotalCharges"].replace(" ", "0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

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   float64
```

```
20 Churn          7043 non-null object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

```
df.isnull().sum().sum()
```

```
➡ np.int64(0)
```

```
df.describe()
```

➡

	tenure	MonthlyCharges	TotalCharges	
count	7043.000000	7043.000000	7043.000000	
mean	32.371149	64.761692	2279.734304	
std	24.559481	30.090047	2266.794470	
min	0.000000	18.250000	0.000000	
25%	9.000000	35.500000	398.550000	
50%	29.000000	70.350000	1394.550000	
75%	55.000000	89.850000	3786.600000	
max	72.000000	118.750000	8684.800000	

```
df["customerID"].duplicated().sum()
```

```
➡ np.int64(0)
```

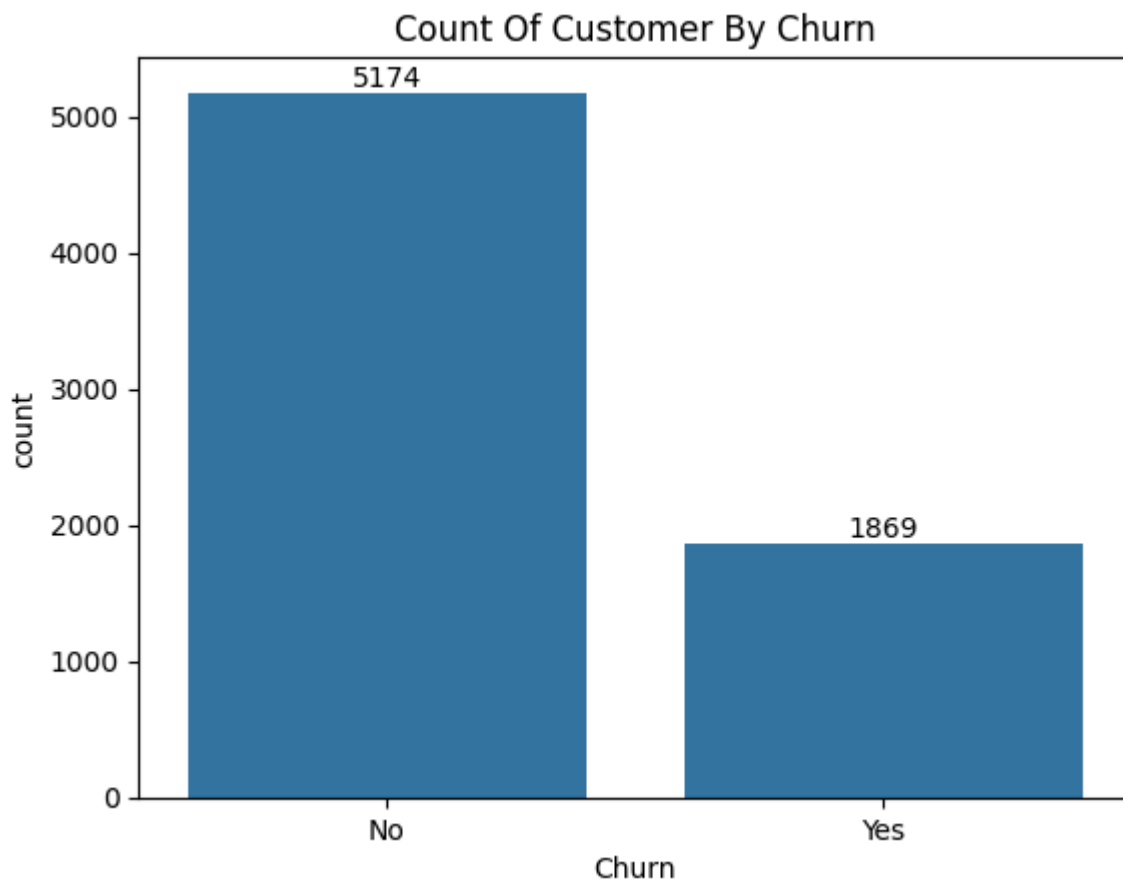
```
def conv(valu):
    if valu ==1:
        return "yes"
    else:
        return "no"
```

```
df ['SeniorCitizen'] = df ["SeniorCitizen"].apply(conv)
```

**#converted 0 and 1 values of senior citizen to yes/no to make it easier to understand**

```
ax= sns.countplot(x='Churn', data = df )
```

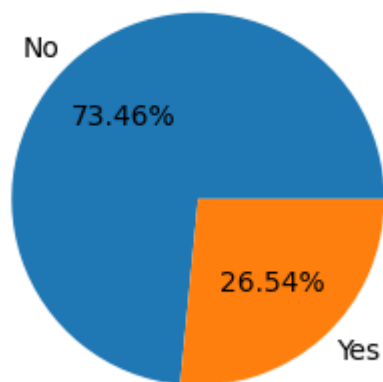
```
ax.bar_label(ax.containers[0])
plt.title("Count Of Customer By Churn")
plt.show()
```



```
plt.figure(figsize =(3,4))
gb=df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels =gb.index, autopct= "%1.2f%%")
plt.title("Percentage of Churned Customer",fontsize= 10)
plt.show()
```

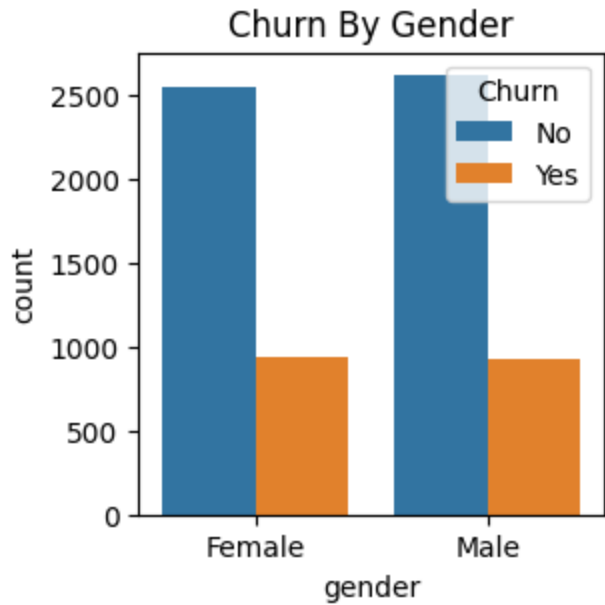


Percentage of Churned Customer

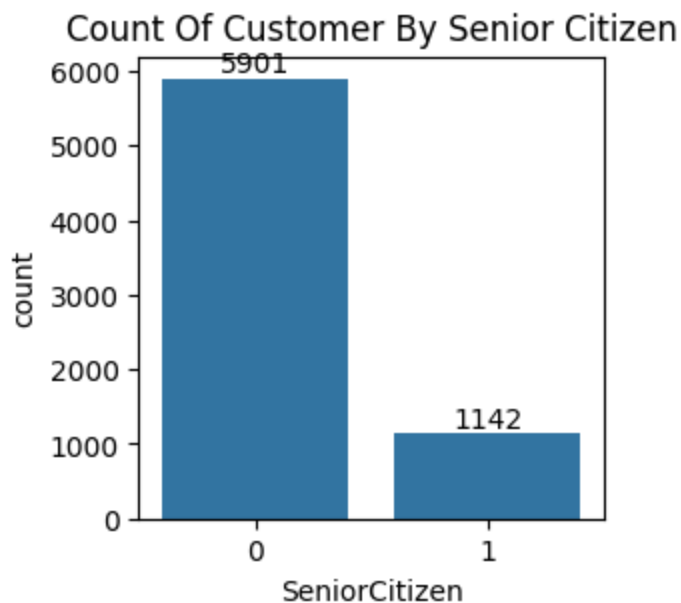


#from the given pie chart we can conclude that 26.54% of our customers have churned out #now let's explore the reason behind it

```
plt.figure(figsize=(3,3))
sns.countplot(x= "gender" , data =df , hue= "Churn")
plt.title("Churn By Gender")
plt.show()
```



```
plt.figure(figsize=(3,3))
ax= sns.countplot(x= "SeniorCitizen" , data =df )
ax.bar_label(ax.containers[0])
plt.title("Count Of Customer By Senior Citizen")
plt.show()
```



```

grouped = df.groupby(['SeniorCitizen', 'Churn']).size().unstack(fill_value=0)
percent = grouped.div(grouped.sum(axis=1), axis=0) * 100

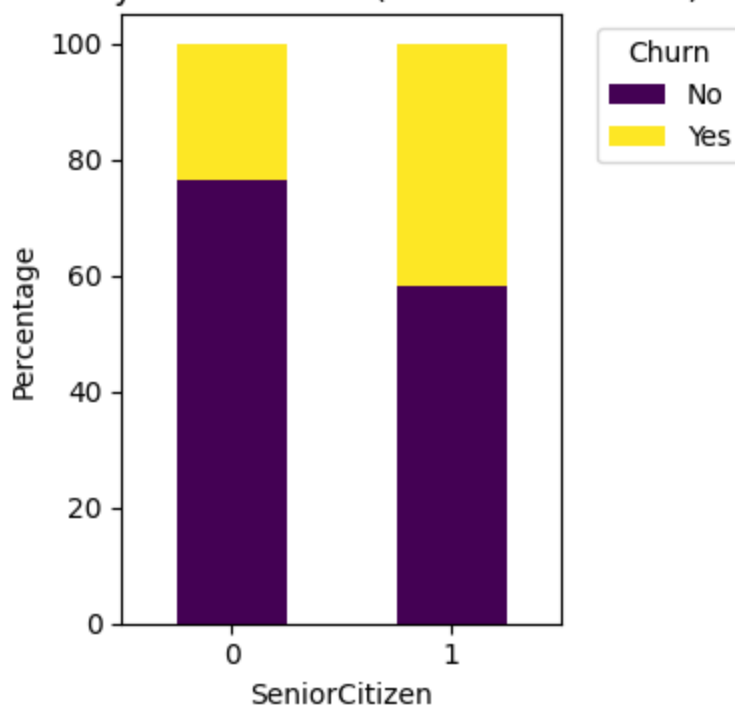
# Step 2: Plot as a stacked bar chart
percent.plot(kind='bar', stacked=True, figsize=(4, 4), colormap='viridis')

plt.title('Churn by SeniorCitizen (Stacked Bar Chart)')
plt.ylabel('Percentage')
plt.xlabel('SeniorCitizen')
plt.legend(title='Churn', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()

```



Churn by SeniorCitizen (Stacked Bar Chart)

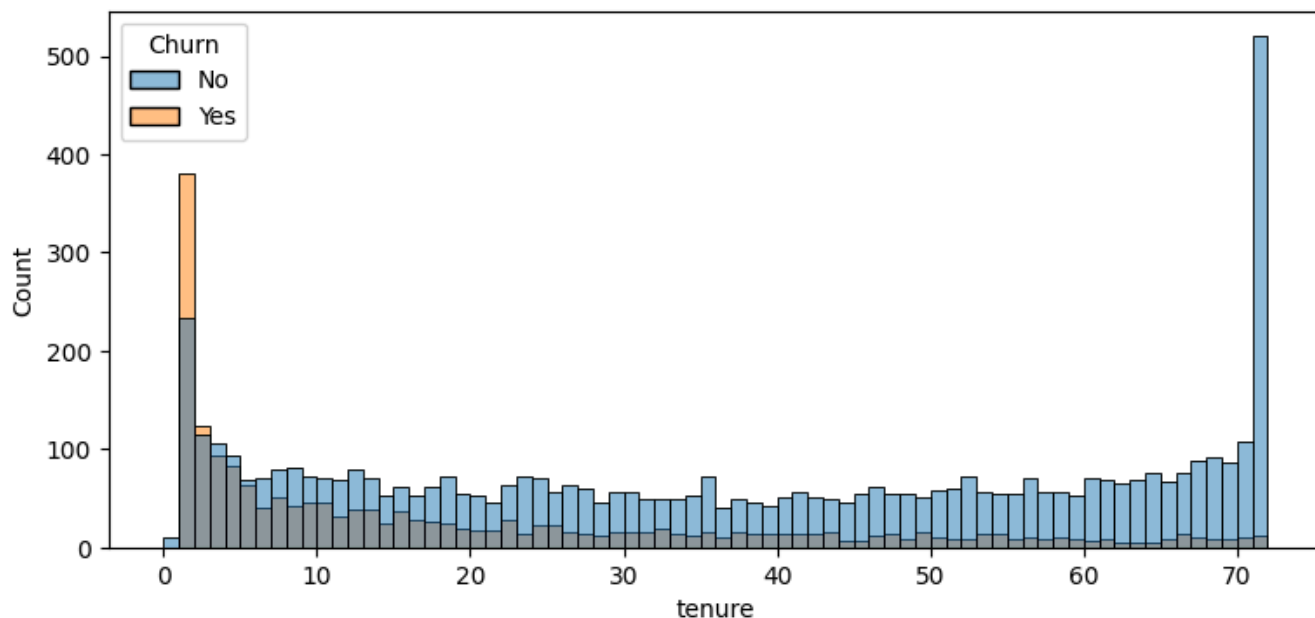


**#comparative a greater percentage of people in senior citizen category have churned**

```

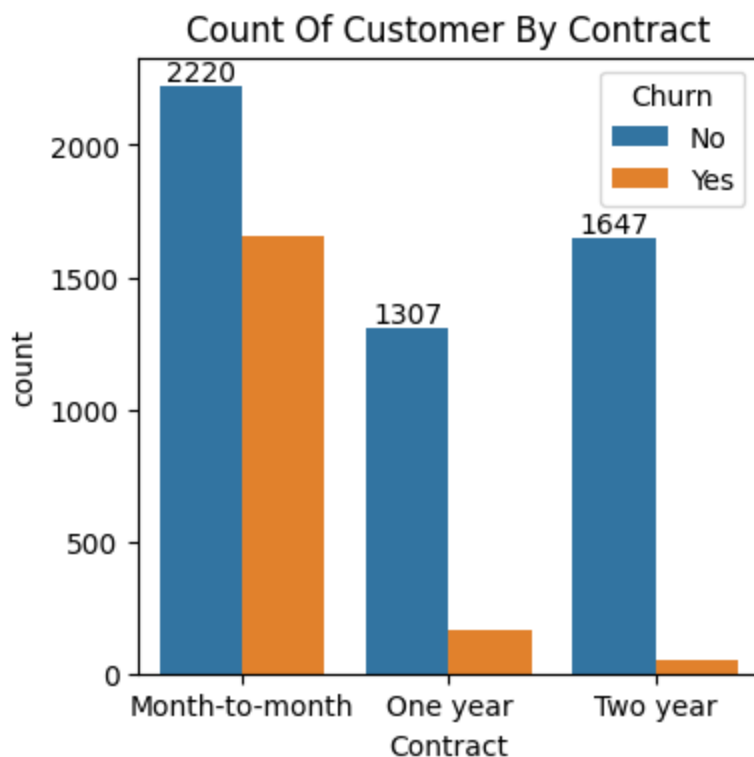
plt.figure(figsize=(9,4))
sns.histplot(x="tenure", data = df, bins =72, hue ="Churn")
plt.show()

```



#people who have used our services for a long time have stayed and people who have used our services #1 or 2 months have churned

```
plt.figure(figsize=(4,4))
ax= sns.countplot(x= "Contract" , data =df, hue = 'Churn' )
ax.bar_label(ax.containers[0])
plt.title("Count Of Customer By Contract")
plt.show()
```



**#people who have month contract are likely to churn then from those who have 1 or 2 years of contract**

```
df.columns.values
```

```
⇒ array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',  
        'tenure', 'PhoneService', 'MultipleLines', 'InternetService',  
        'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',  
        'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',  
        'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',  
        'TotalCharges', 'Churn'], dtype=object)
```

```
# List of categorical columns you want to plot
```

```
columns = ['PhoneService', 'MultipleLines', 'InternetService',  
          'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',  
          'TechSupport', 'StreamingTV', 'StreamingMovies']
```

```
# Set up the figure and axes (adjust rows and columns as needed)
```

```
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12)) # 3x3 grid for 9 plots  
axes = axes.flatten() # Flatten to 1D array for easier iteration
```

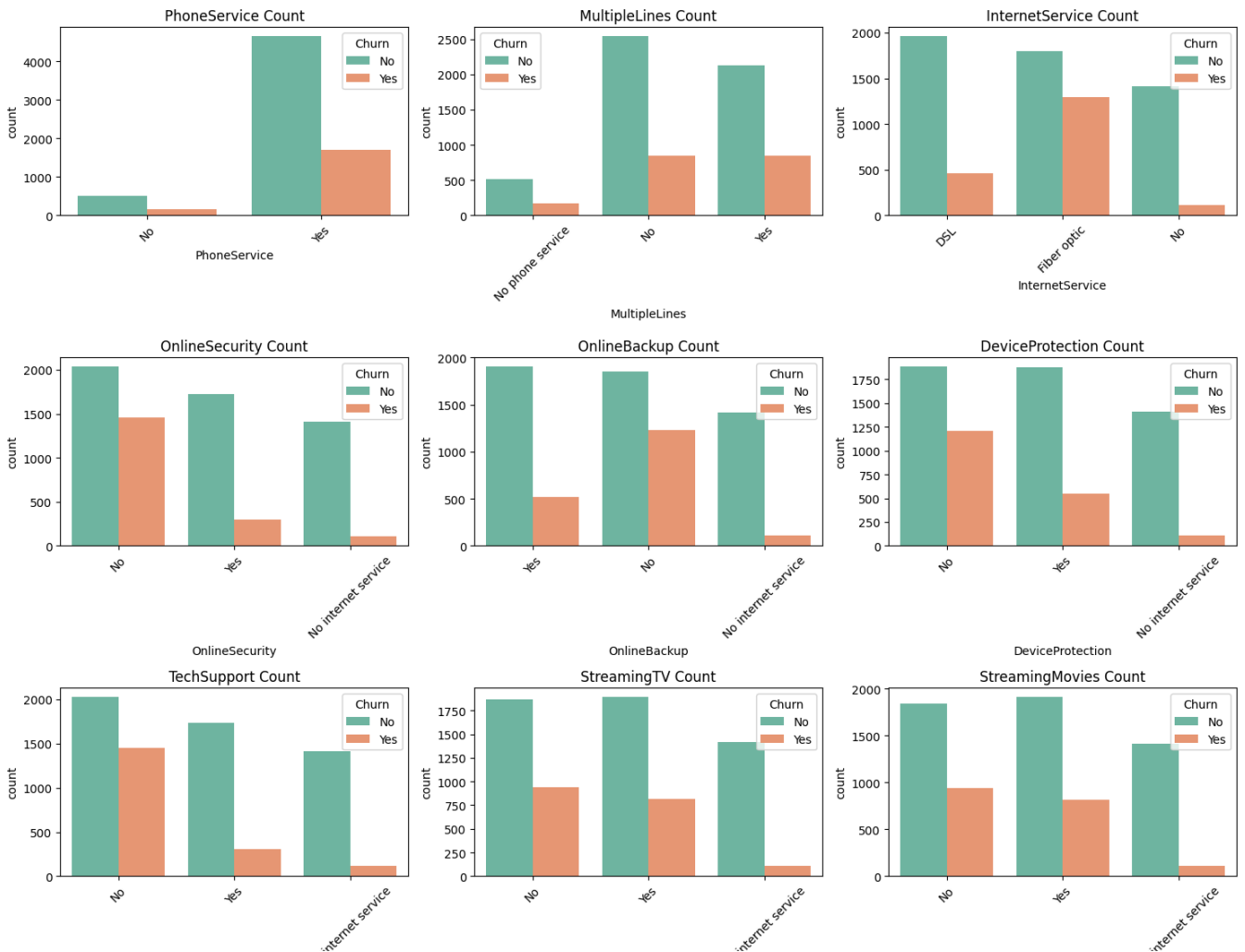
```
# Loop through each column and corresponding subplot
```

```
for idx, col in enumerate(columns):  
    sns.countplot(x=col, data=df, ax=axes[idx], palette="Set2" , hue = 'Churn')  
    axes[idx].set_title(f"{col} Count")  
    axes[idx].tick_params(axis='x', rotation=45)
```

```
# Adjust layout
```

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





**#Customers without internet-related services (like OnlineSecurity, OnlineBackup, or TechSupport) show relatively lower churn rates, likely due to their limited engagement. Features like Fiber Optic Internet, Multiple Lines, and lack of Device Protection are associated with higher churn rates. Conversely, users with PhoneService and Streaming services are more likely to stay, although churn is still present. Overall, digital service engagement shows a strong relationship with customer churn.**

```
plt.figure(figsize=(6,4))
ax= sns.countplot(x= "PaymentMethod" , data =df, hue = 'Churn' )
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Churned Customer By Payment Method")
plt.xticks(rotation=45)
plt.show()
```

