CHURN ANALYSIS

1 PROJECT : CHURN ANALYSIS (EDA)

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df=pd.read_csv("Customer Churn.csv")
print(df.head(10))
```

```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1           No
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
4  9237-HQITU  Female              0      No         No       2          Yes
5  9305-CDSKC  Female              0      No         No       8          Yes
6  1452-KIOVK    Male              0      No        Yes      22          Yes
7  6713-OKOMC  Female              0      No         No      10           No
8  7892-POOKP  Female              0     Yes         No      28          Yes
9  6388-TABGU    Male              0      No        Yes      62          Yes

      MultipleLines InternetService OnlineSecurity  ... DeviceProtection  \
0  No phone service             DSL             No  ...               No
1                No             DSL            Yes  ...              Yes
2                No             DSL            Yes  ...               No
3  No phone service             DSL            Yes  ...              Yes
4                No     Fiber optic             No  ...               No
5               Yes     Fiber optic             No  ...              Yes
6               Yes     Fiber optic             No  ...               No
7  No phone service             DSL            Yes  ...               No
8               Yes     Fiber optic             No  ...              Yes
9                No             DSL            Yes  ...               No

  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
2          No          No              No  Month-to-month              Yes
```

3	Yes	No	No	One year	No
4	No	No	No	Month-to-month	Yes
5	No	Yes	Yes	Month-to-month	Yes
6	No	Yes	No	Month-to-month	Yes
7	No	No	No	Month-to-month	No
8	Yes	Yes	Yes	Month-to-month	Yes
9	No	No	No	One year	No

	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	Electronic check	29.85	29.85	No
1	Mailed check	56.95	1889.5	No
2	Mailed check	53.85	108.15	Yes
3	Bank transfer (automatic)	42.30	1840.75	No
4	Electronic check	70.70	151.65	Yes
5	Electronic check	99.65	820.5	Yes
6	Credit card (automatic)	89.10	1949.4	No
7	Mailed check	29.75	301.9	No
8	Electronic check	104.80	3046.05	Yes
9	Bank transfer (automatic)	56.15	3487.95	No

[10 rows x 21 columns]

[3]:
```
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
```

Replacing blanks with 0 as tenure is 0 and total charges are recorded

[4]:
```
#As totalCharges are of object Dtype s convert it into float
df["TotalCharges"]=df["TotalCharges"].replace(" ", "0")
df["TotalCharges"]=df["TotalCharges"].astype("float")
```

[5]:
```
#monthly charges has now been converted into 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
```

[6]:
```
df.isnull().sum()
```

[6] :
```
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        0
Churn               0
dtype: int64
```

or finding total sum of all the null values

[7] :
```
df.isnull().sum().sum()
```

[7]: 0

[8] :
```
df.describe()
```

[8]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	18.250000	0.000000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

[9] :
```
df["customerID"].duplicated().sum()
```

[9]: 0

Converted 0 and 1 value of seniorcitizen into true or false

[10] :
```
def conv(val):
    if  val==1:
          return  "yes"
      else:
```

```
        return "no"
df['SeniorCitizen']=df["SeniorCitizen"].apply(conv)
```

[11] :
```
df.head(10)
```

[11]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	\
0	7590-VHVEG	Female	no	Yes	No	1	No	
1	5575-GNVDE	Male	no	No	No	34	Yes	
2	3668-QPYBK	Male	no	No	No	2	Yes	
3	7795-CFOCW	Male	no	No	No	45	No	
4	9237-HQITU	Female	no	No	No	2	Yes	
5	9305-CDSKC	Female	no	No	No	8	Yes	
6	1452-KIOVK	Male	no	No	Yes	22	Yes	
7	6713-OKOMC	Female	no	No	No	10	No	
8	7892-POOKP	Female	no	Yes	No	28	Yes	
9	6388-TABGU	Male	no	No	Yes	62	Yes	

	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	\
0	No phone service	DSL	No	...	No	
1	No	DSL	Yes	...	Yes	
2	No	DSL	Yes	...	No	
3	No phone service	DSL	Yes	...	Yes	
4	No	Fiber optic	No	...	No	
5	Yes	Fiber optic	No	...	Yes	
6	Yes	Fiber optic	No	...	No	
7	No phone service	DSL	Yes	...	No	
8	Yes	Fiber optic	No	...	Yes	
9	No	DSL	Yes	...	No	

	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	\
0	No	No	No	Month-to-month	Yes	
1	No	No	No	One year	No	
2	No	No	No	Month-to-month	Yes	
3	Yes	No	No	One year	No	
4	No	No	No	Month-to-month	Yes	
5	No	Yes	Yes	Month-to-month	Yes	
6	No	Yes	No	Month-to-month	Yes	
7	No	No	No	Month-to-month	No	
8	Yes	Yes	Yes	Month-to-month	Yes	
9	No	No	No	One year	No	

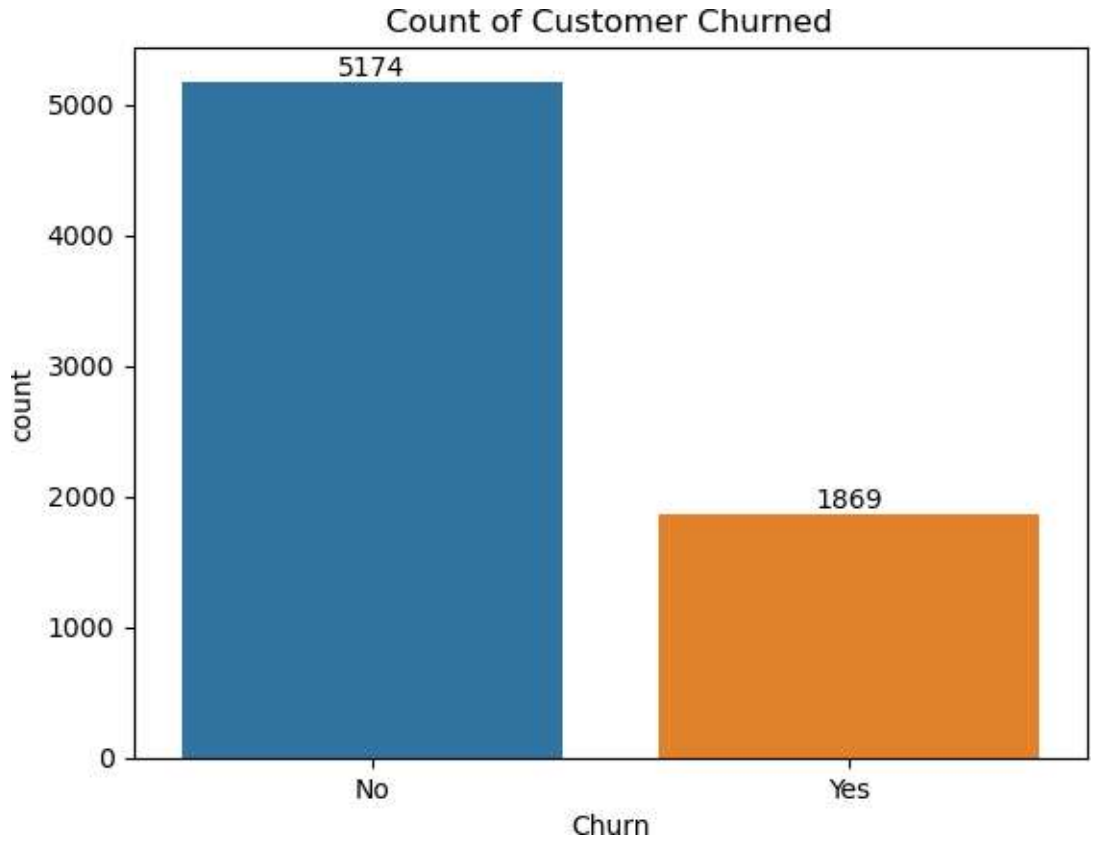
	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	Electronic check	29.85	29.85	No
1	Mailed check	56.95	1889.50	No
2	Mailed check	53.85	108.15	Yes
3	Bank transfer (automatic)	42.30	1840.75	No
4	Electronic check	70.70	151.65	Yes

5	Electronic check	99.65	820.50	Yes
6	Credit card (automatic)	89.10	1949.40	No
7	Mailed check	29.75	301.90	No
8	Electronic check	104.80	3046.05	Yes
9	Bank transfer (automatic)	56.15	3487.95	No

[10 rows x 21 columns]

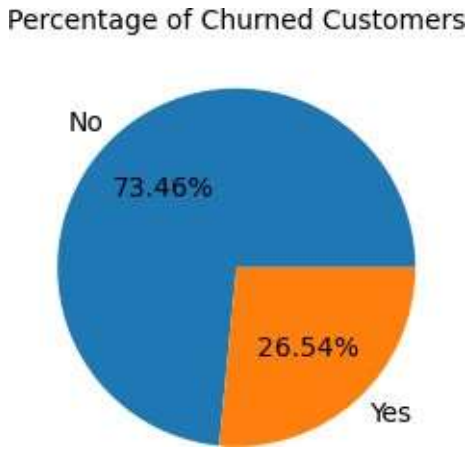
Insight :1 How many customers have churn out ?

[12] :
```
ax=sns.countplot(x="Churn", data =df)
ax.bar_label(ax.containers[0]) #for labels on bars
plt.title("Count of Customer Churned")
plt.show()
```



2 Insight: 2 Percentage of Churned Customers

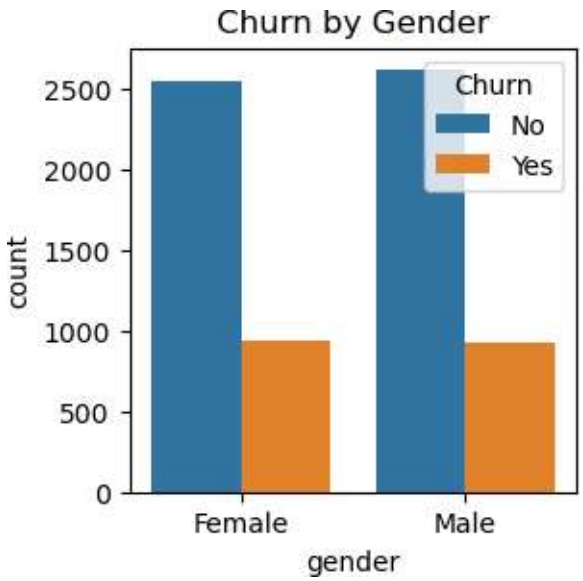
[13]:
```
plt.figure(figsize=(3,4)) # to set the size of chart (height , width)
gb=df.groupby("Churn").agg({'Churn':'count'})
plt.title("Percentage of Churned Customers" , fontsize=10)
plt.pie(gb["Churn"] , labels=gb.index , autopct="%1.2f%%" ) # labels for yes
↪and no and autopercentage as autopct

plt.show()
```



3 Insight: 3 Churn by Gender

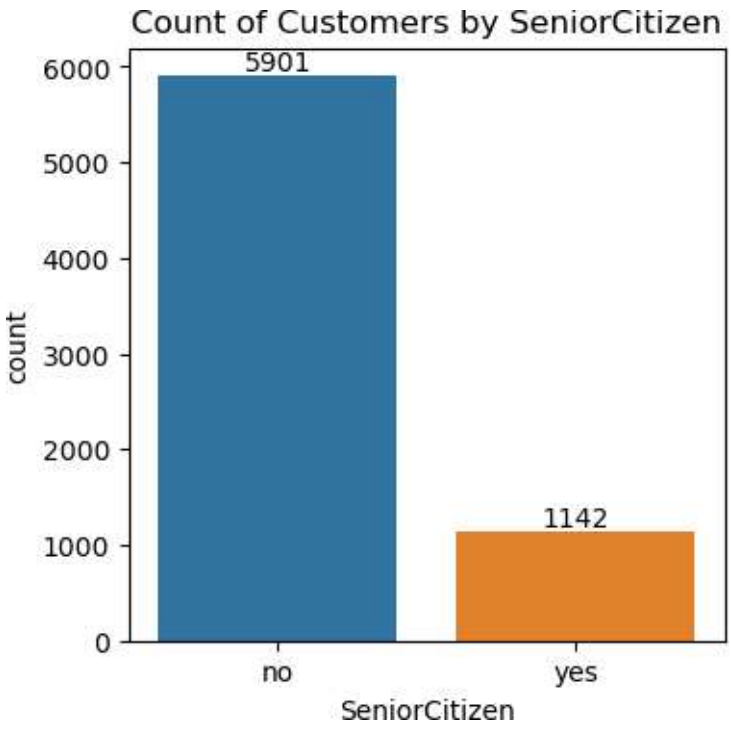
From the given pie chart we can conclude that 26.54% of our customers have churned out #Now let's explore the reason behind it

[14]:
```
plt.figure(figsize=(3,3))
sns.countplot(x="gender" , data =df , hue="Churn")
plt.title("Churn by Gender")
plt.show()
```



4 Insight:4 Count of Customers by SeniorCitizen

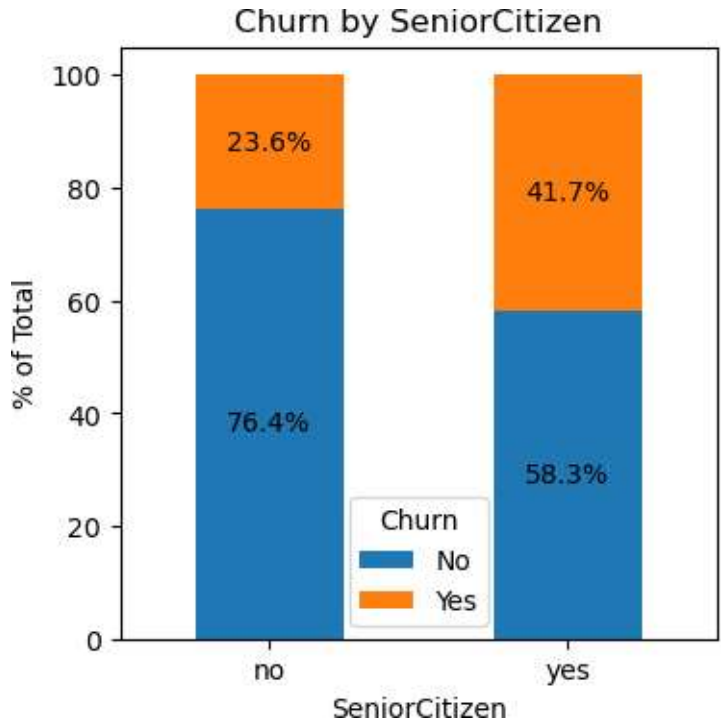
[15]:
```
plt.figure(figsize=(4,4))
ax= sns.countplot(x="SeniorCitizen" , data =df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by SeniorCitizen")
plt.show()
```



5 Insight:5 Churn by SeniorCitizen

[16]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Calculate counts and percentage of Churn by SeniorCitizen
counts = df.groupby(['SeniorCitizen', 'Churn']).size().unstack()
totals = counts.sum(axis=1)

# Normalize to get percentages
percentages = counts.divide(totals, axis=0) * 100

# Plot the stacked bar chart
fig, ax = plt.subplots(figsize=(4,4))
percentages.plot(kind='bar', stacked=True, ax=ax)

# Add labels
for i in ax.containers:
    ax.bar_label(i, label_type='center', fmt='%.1f%%')

# Customize the chart
plt.title('Churn by SeniorCitizen')
```

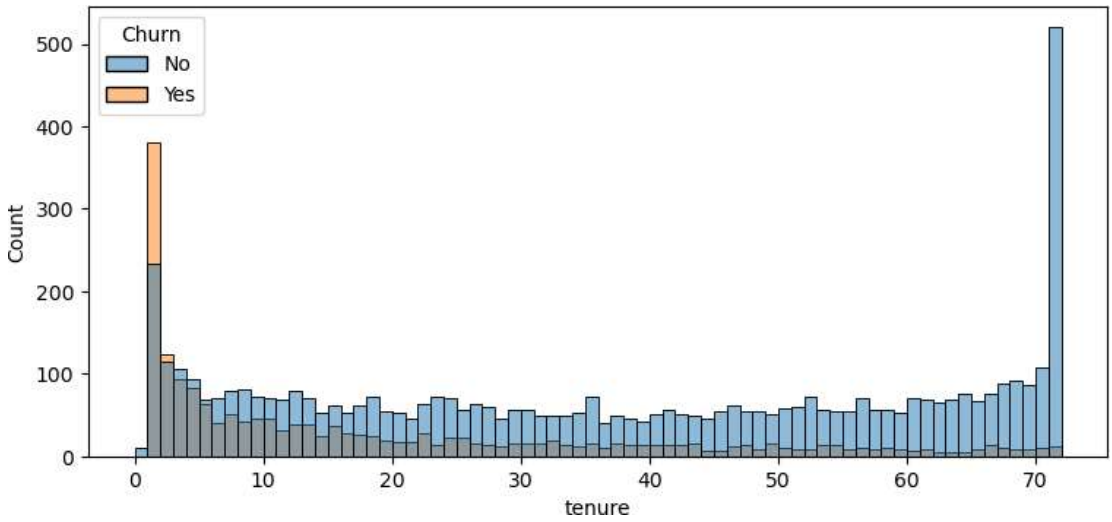
```
plt.ylabel('% of Total')
plt.xlabel('SeniorCitizen')
plt.xticks(rotation=0)
plt.show()
```



#comparatively a greater percentage of people in Senior citizen category has churned

6 Insight :6 Churn on the basis of Tenure

[17] :
```
plt.figure(figsize=(9,4))
sns.histplot(x="tenure", data=df, bins=72, hue="Churn")
plt.show()
```



7 Insight: 7 Count of Customers by Contract

#people who have used our services for a long time have stayed and people who have used our services for 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 Customers by Contract") plt.show()

#people who have month-to month are likely to churn for those who ahave one or 2 year

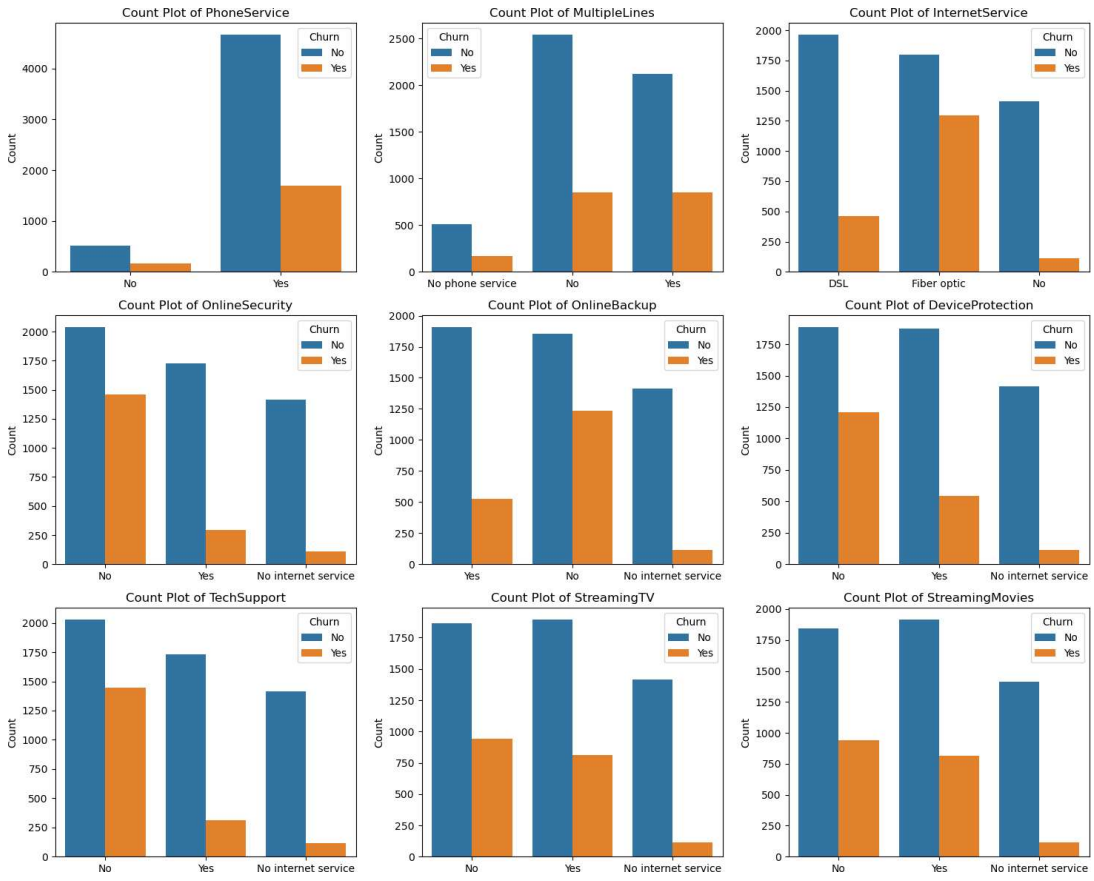
[19]:
```
df.columns.values
```

[19]:
```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

8 Insight 8 : Churn on the basis of different services

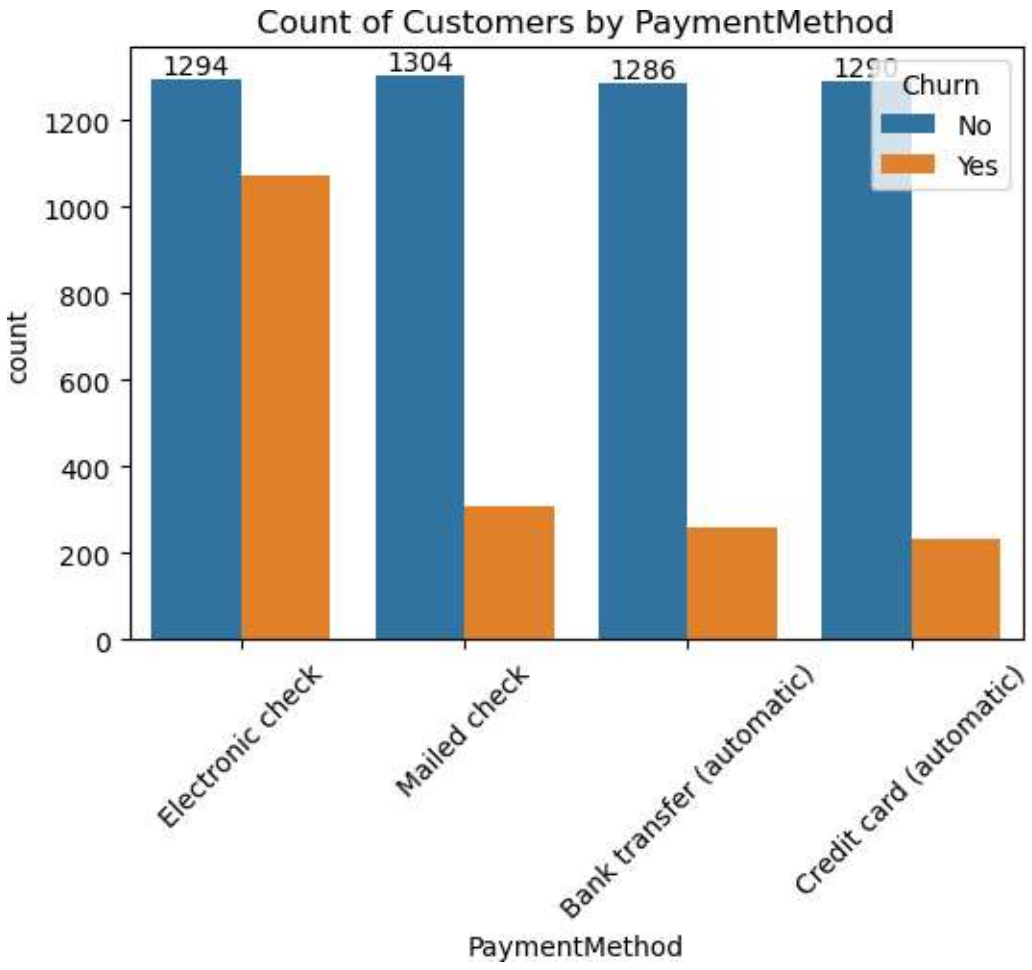
[21]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Define the columns you want to plot
columns = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
           'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',␣
↪'StreamingMovies']
```

```
# Set up the figure and axes
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12))   # 3x3 grid

# Flatten the axes array for easier indexing
axes = axes.flatten()

# Loop through each column and create a count plot with hue='Churn'
for i, col in enumerate(columns):
    sns.countplot(x=col, hue='Churn', data=df, ax=axes[i])
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')

# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```



9 Insight 9 : Count of Customers by PaymentMethod

[24]:
```
plt.figure(figsize=(6,4))
ax= sns.countplot(x="PaymentMethod" , data =df , hue="Churn")
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by PaymentMethod")
plt.xticks(rotation =45)
plt.show()
```



#customer is likely to churn when he is using electronic check as a payment method

10 THANKYOU !!

[ ]: