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

df = pd.read_csv('Customer churn analysis.csv')
df.head()
```

```
   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

      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

  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

               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 rows x 21 columns]
```

```
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 no total charges are recorded and also convert data object into float

```
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()
```

```
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
```

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

```
0
```

```
df.describe()
```

```
       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
```

#We are checking duplictes by using its unique id like customerID

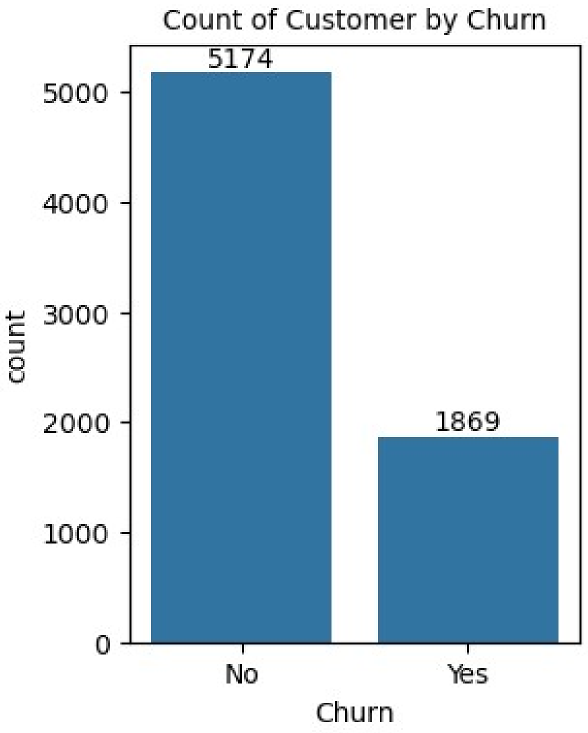
```
df['customerID'].duplicated().sum()
```

```
0
```

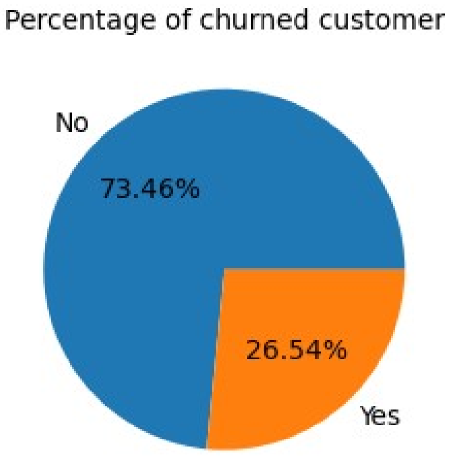
```
def conv(value):
    if value ==1:
        return"Yes"
    else:
        return"No"
df['SeniorCitizen']=df['SeniorCitizen'].apply(conv)
```

#Convert 0 and 1 values of senior citizen to Yes or No to make it easier to understand by using Def function

```
plt.figure(figsize=(3,4))
ax=sns.countplot(x=df['Churn'],data=df)
ax.bar_label(ax.containers[0])
plt.title('Count of Customer by Churn',fontsize=10)
plt.show()
```



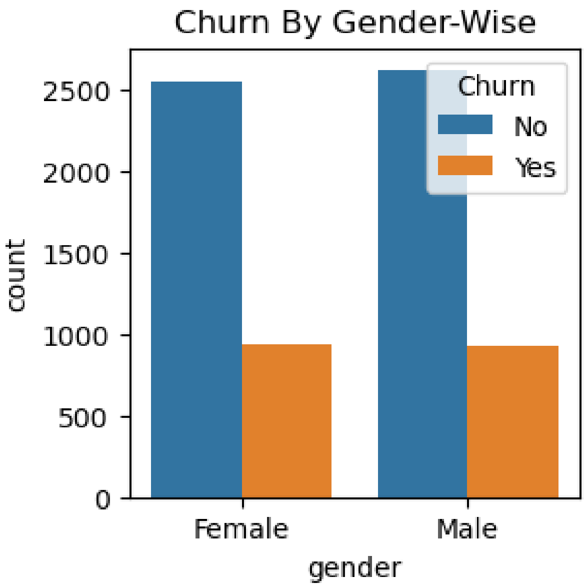
```
plt.figure(figsize=(3,4))
gb=df.groupby('Churn').agg({'Churn':'count'})
plt.pie(gb['Churn'], labels=gb.index, autopct='%1.2f%%')
plt.title('Percentage of Churned Customer', fontsize=10)
plt.show()
```



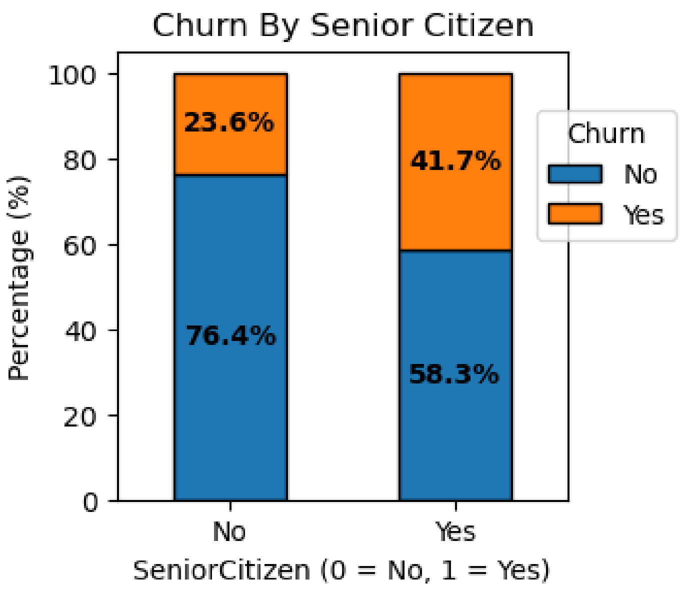
#From the given pie chart we can conclude that 26.54% of our customers have churned out.

```
plt.figure(figsize=(3,3))
sns.countplot(x=df['gender'],data=df,hue='Churn')
```

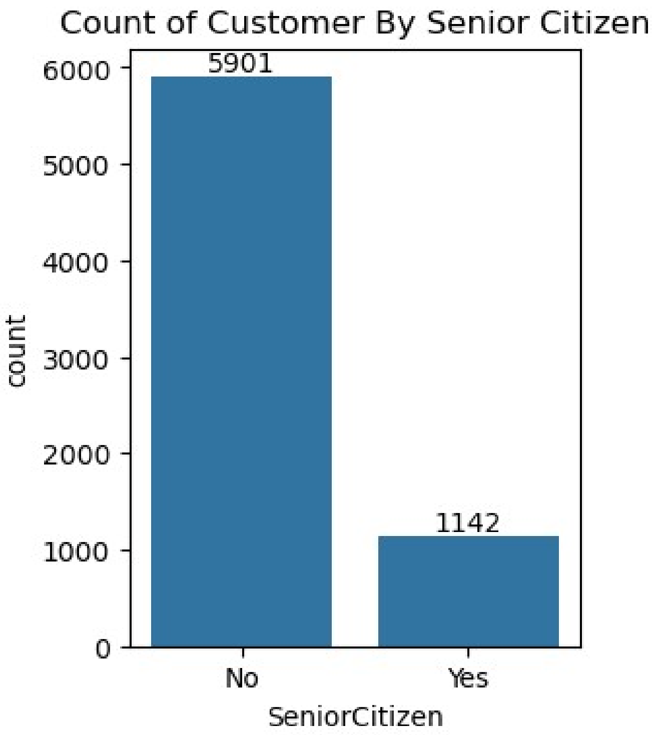
```
plt.title('Churn By Gender-Wise')
plt.show()
```



```
ct = pd.crosstab(df['SeniorCitizen'], df['Churn'], normalize='index')
* 100

# Plot stacked bar chart
ax = ct.plot(kind='bar', stacked=True, figsize=(3,3),
color=['#1f77b4','#ff7f0e'], edgecolor='black')

# Add percentage labels
for p in ax.patches:
    if p.get_height() > 0:
        ax.text(p.get_x() + p.get_width()/2, p.get_y() +
p.get_height()/2, f'{p.get_height():.1f}%',
                ha='center', va='center', fontsize=10, color='black',
fontweight='bold')

# Customize plot
plt.title('Churn By Senior Citizen')
plt.xlabel('SeniorCitizen (0 = No, 1 = Yes)')
plt.ylabel('Percentage (%)')
plt.xticks(rotation=0)
plt.legend(title='Churn', bbox_to_anchor=(0.9,0.9))
plt.show()
```

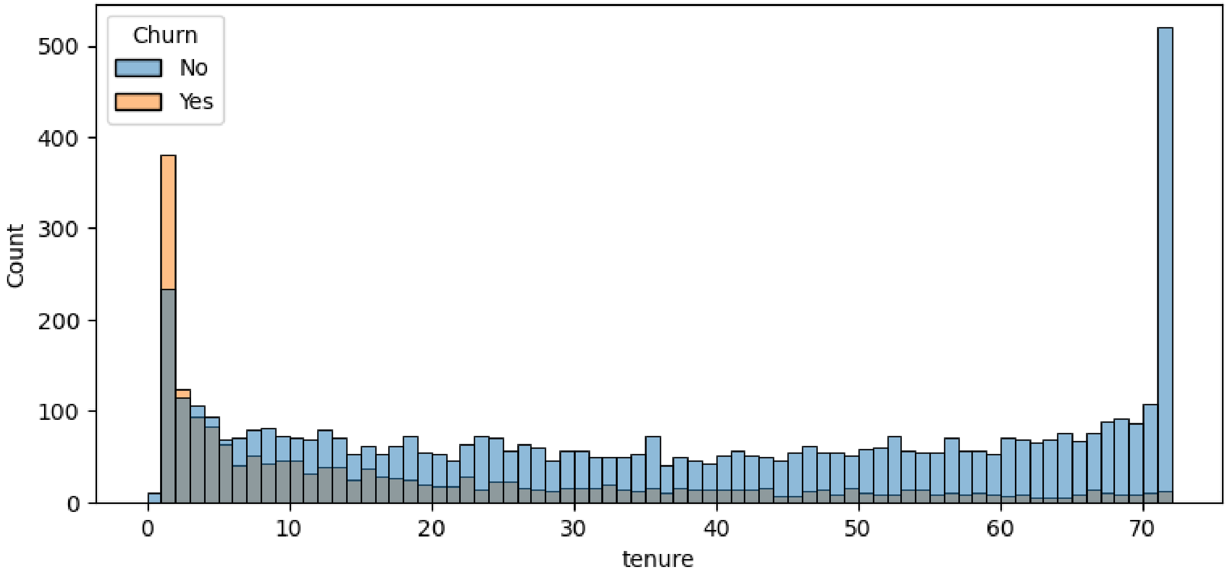


```
plt.figure(figsize=(3,4))
ax=sns.countplot(x=df['SeniorCitizen'],data=df)
ax.bar_label(ax.containers[0])
plt.title('Count of Customer By Senior Citizen')
plt.show()
```



#Comparative a Greater Percentage of People In Senior Citizen Category Have Chured.

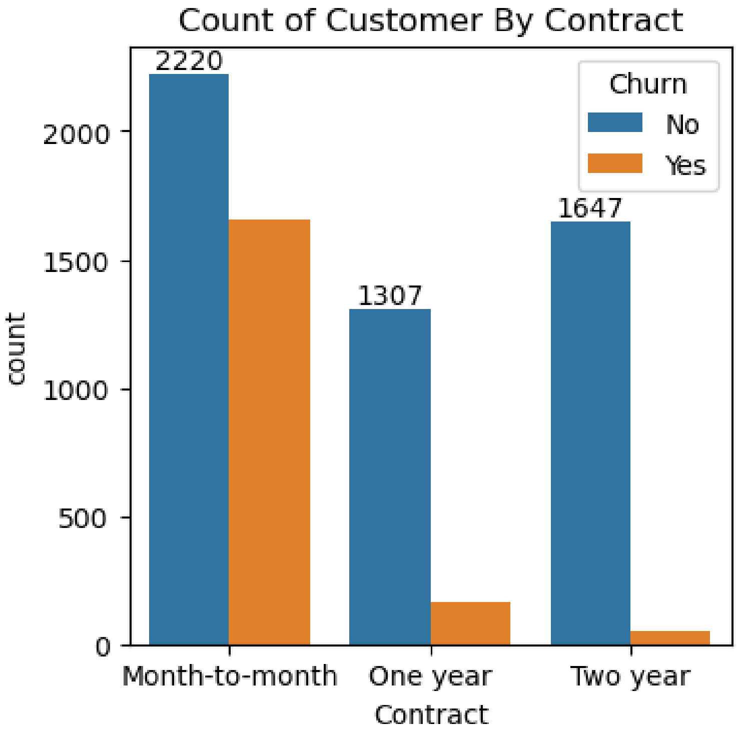
```
plt.figure(figsize=(9,4))
sns.histplot(x='tenure',data=df, bins=72, hue='Churn')
plt.show()
```



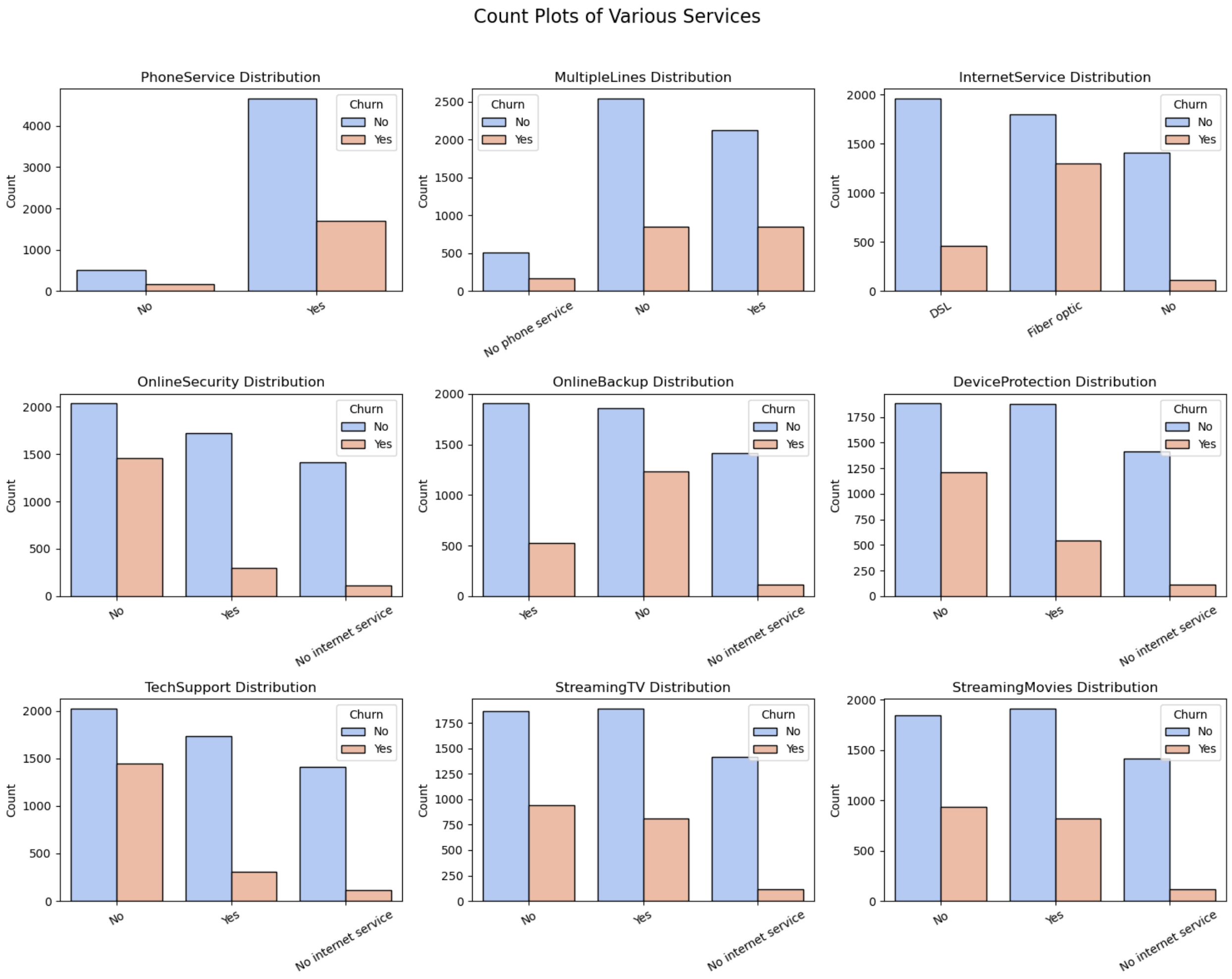
#People Who Have Used Our Services For a Long Time Have Stayed.

#People Who Have Used Our Services For 1 or 2 Months Have Churned.

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

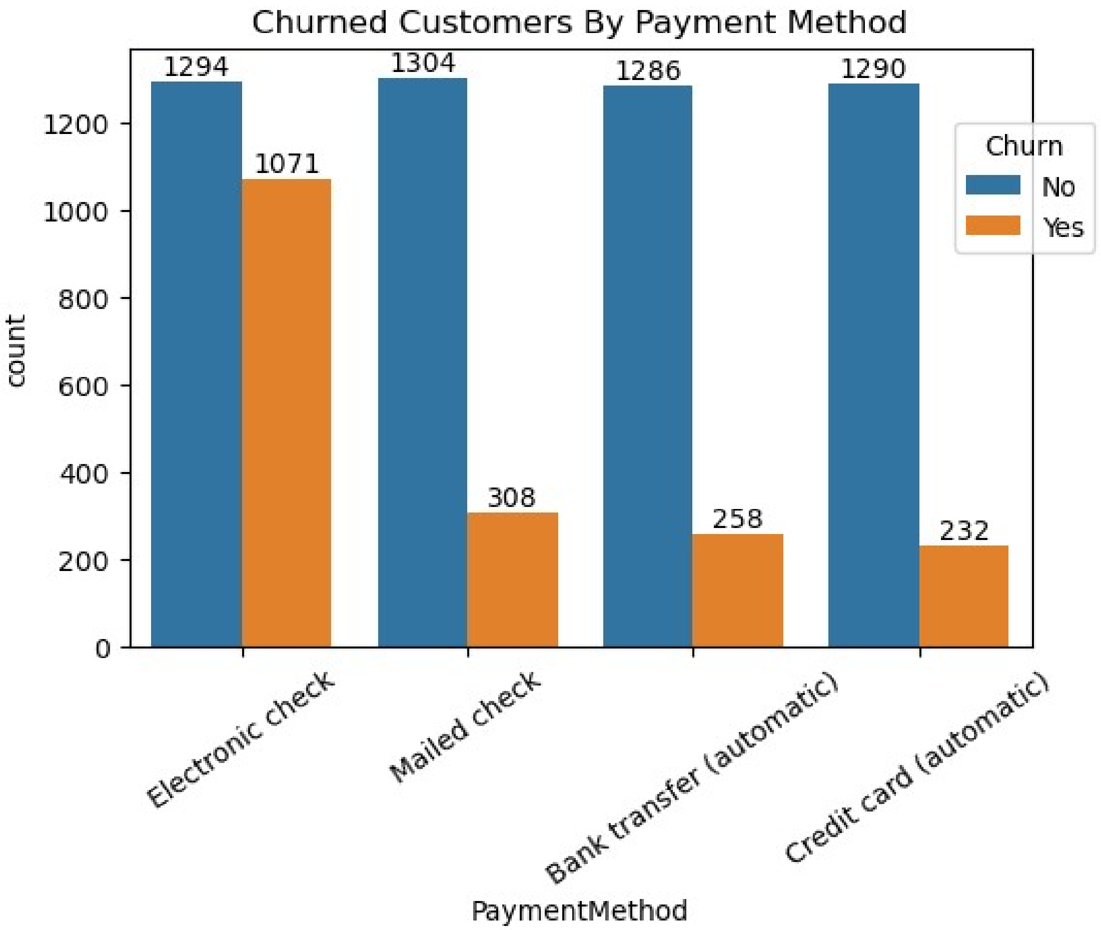


```
#People Who Have Month To Month Contract Are Likely To Churn.
#Who Have 1 or 2 Years Contract Are Likely To Have Very Less Chances
To Churn.

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)

# Define the columns to plot
cols = ['PhoneService', 'MultipleLines', 'InternetService',
        'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
        'TechSupport', 'StreamingTV', 'StreamingMovies']

# Create subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12))  # 3x3
grid
fig.suptitle("Count Plots of Various Services", fontsize=16)

# Loop through each column and plot
for col, ax in zip(cols, axes.flatten()):
    sns.countplot(x=df[col], data=df, ax=ax, palette='coolwarm',
edgecolor='black',hue=df['Churn'])
    ax.set_title(f"{col} Distribution")
    ax.set_xlabel('')
    ax.set_ylabel('Count')
    ax.tick_params(axis='x', rotation=30)

# Adjust layout
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



#It highlights differences in subscription patterns between churned and retained customers, with noticeable variations in services like OnlineSecurity and TechSupport, where churn rates appear higher for non-subscribers.

```
plt.figure(figsize=(6,4))
ax=sns.countplot(x=df['PaymentMethod'],data=df,hue='Churn')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title('Churned Customers By Payment Method')
plt.xticks(rotation=35)
plt.legend(title='Churn',bbox_to_anchor=(0.9,0.9))
plt.show()
```



#Customer Are Likely To Churned When Customer Using Electronic Check as Payment Method