TELECOM CHURN ANALYSIS

SUMMARY

Cracking the Telecom Customer Retention Code Insights That Reveal Why Customers Leave And How to Make Them Stick

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

```
df=pd.read_csv('Customer_Churn.csv')
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contrac
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Month to month
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One yea
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Month to month
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One yea
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Month to month

5 rows × 21 columns

Replace the blank row with 0 and convert data type of "TotalCharges" from object into Float

In [5]:
```
df['TotalCharges']=df['TotalCharges'].replace(" ","0")
df['TotalCharges']=df['TotalCharges'].astype('float')
```

In [6]:
```
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
```

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

Out[7]:
```
0
```

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

Out[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

In [10]:
```
df.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
df['customerID'].duplicated().sum()
```

Out[11]:
```
0
```

In [12]:
```
def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"
df['SeniorCitizen']= df['SeniorCitizen'].apply(conv)
```

Converted Senior Citizen value 0 and 1 to yes/no to make it easier to understand

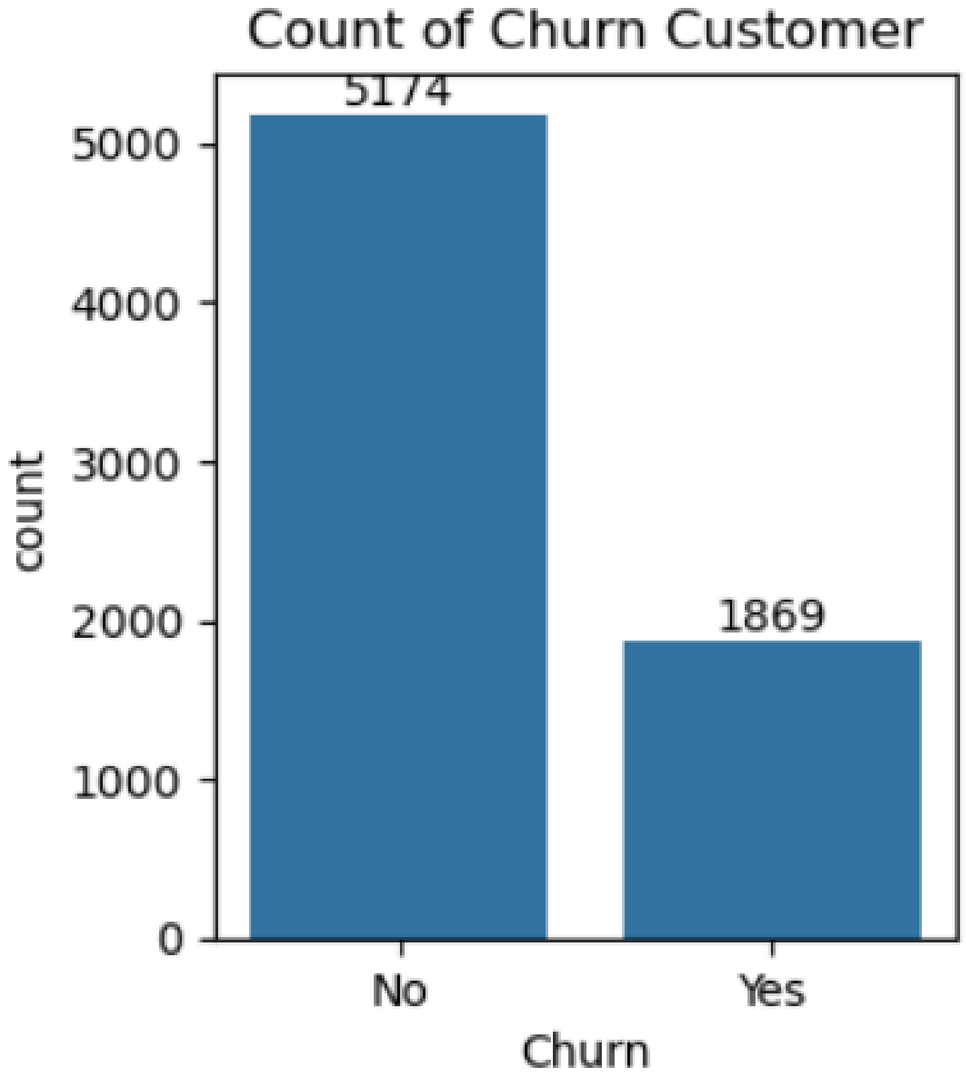
In [13]:
```
df.head()
```

Out[13]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	Pape
0	7590-VHVEG	Female	no	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Month-to-month	
1	5575-GNVDE	Male	no	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One year	
2	3668-QPYBK	Male	no	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Month-to-month	
3	7795-CFOCW	Male	no	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One year	
4	9237-HQITU	Female	no	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Month-to-month	

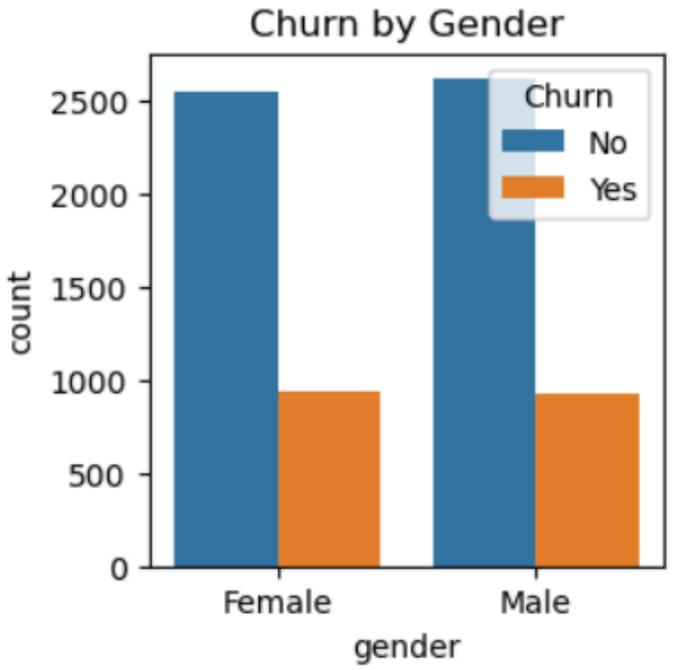
5 rows × 21 columns

In [43]:
```
plt.figure(figsize=(3,3.5))
ax=sns.countplot(x=df['Churn'])
ax.bar_label(ax.containers[0])
plt.title('Count of Churn Customer')
plt.show()
```

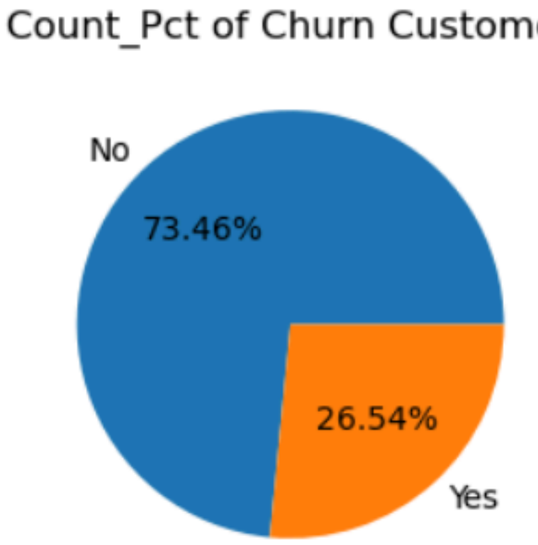


Now the explore reasion behind why the customer churn out or leave the service

In [52]:
```
plt.figure(figsize=(3,3))
sns.countplot(x=df['gender'],data=df,hue='Churn')
plt.title('Churn by Gender')
plt.show()
```

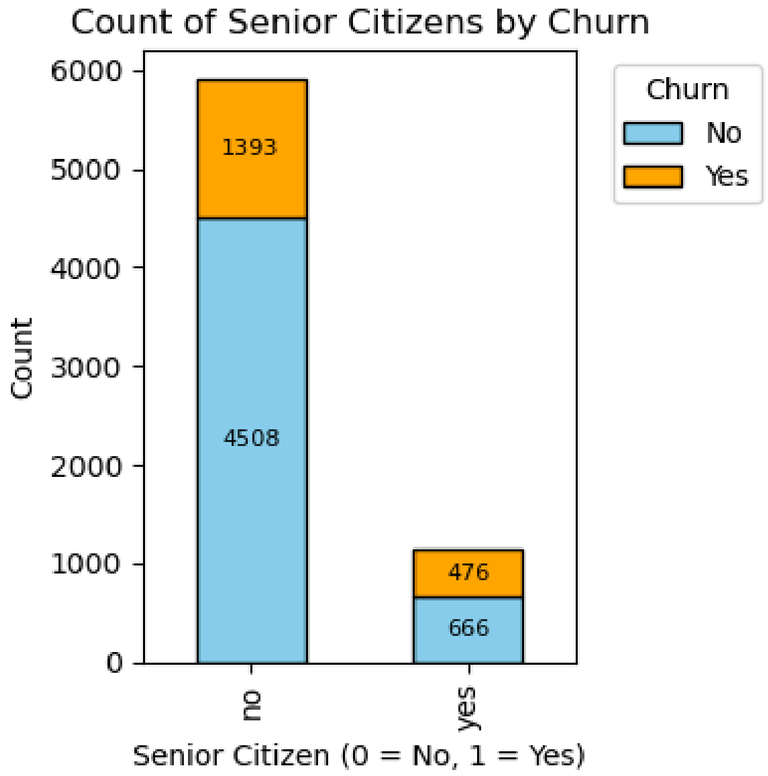


In [44]:
```
plt.figure(figsize=(3,3))
gb=df.groupby('Churn').agg({'Churn':'count'})
plt.pie(gb['Churn'],labels=gb.index,autopct="%1.2f%%")
plt.title('Count_Pct of Churn Customer')
plt.show()
```

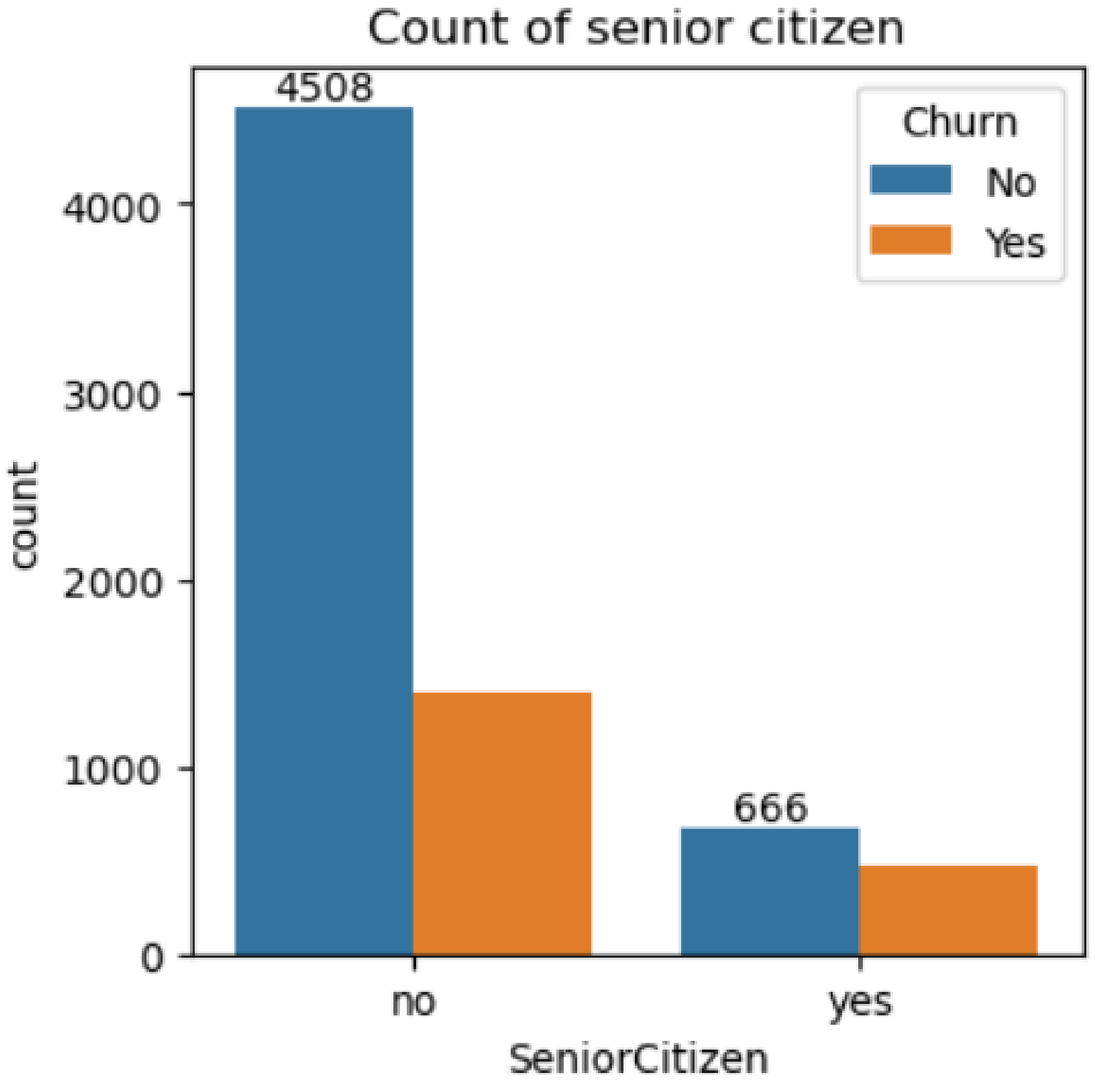


From the given pic chart we can clearly see that almost 27% customer leave t
service due to some reasion

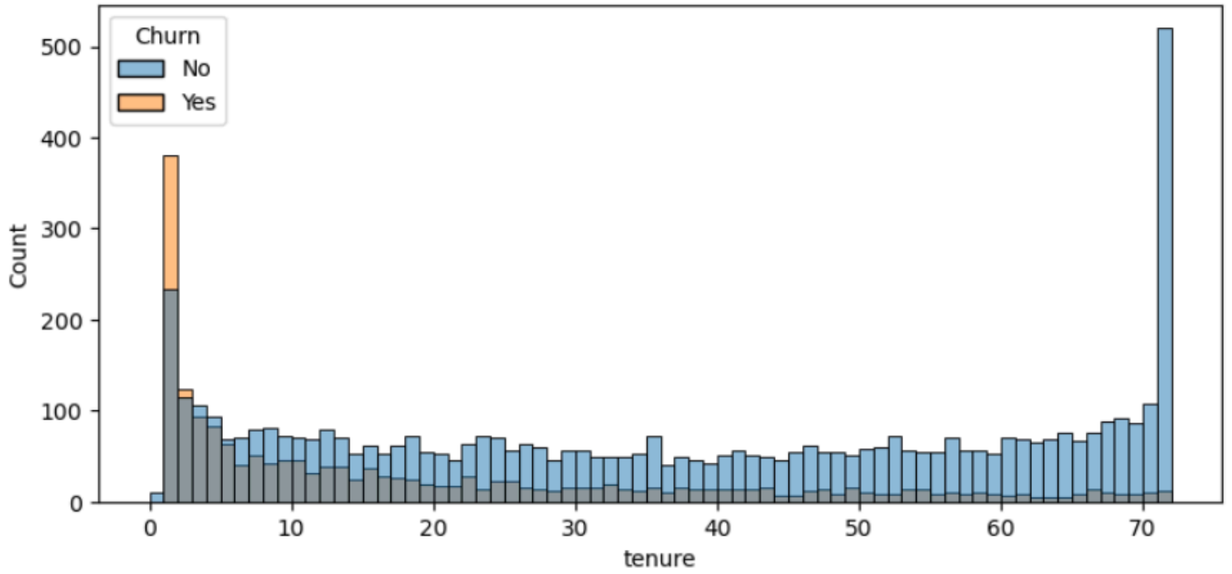
```
    for bar in container:
        height = bar.get_height()
        if height > 0:
            ax.text(
                bar.get_x() + bar.get_width() / 2,
                bar.get_y() + height / 2,
                f'{int(height)}',
                ha='center', va='center', fontsize=8, color='black'
            )

# Customize the chart
plt.title('Count of Senior Citizens by Churn')
plt.xlabel('Senior Citizen (0 = No, 1 = Yes)')
plt.ylabel('Count')
plt.legend(title='Churn', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



In [64]:
```
plt.figure(figsize=(4,4))
ax=sns.countplot(x='SeniorCitizen',data=df,hue='Churn')
ax.bar_label(ax.containers[0])
plt.title('Count of senior citizen')
plt.show()
```

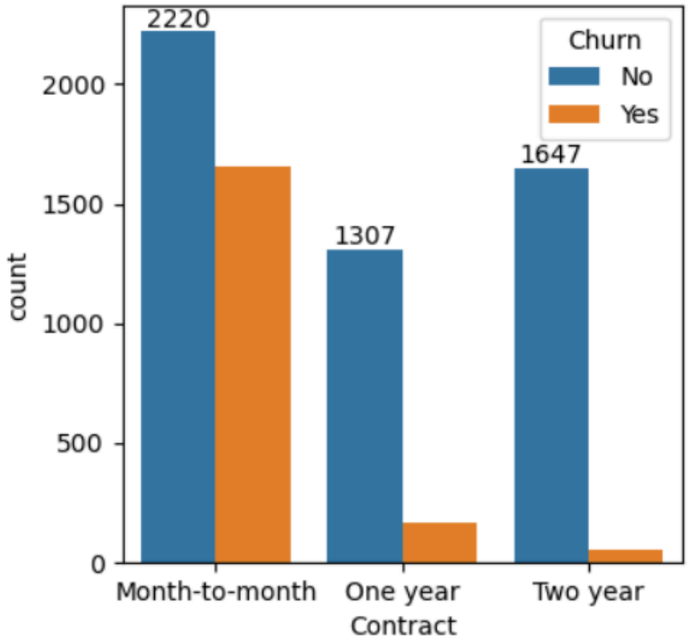


In [74]:
```
plt.figure(figsize=(9,4))
sns.histplot(x='tenure',data=df,bins=72,hue='Churn')
plt.show()
```



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

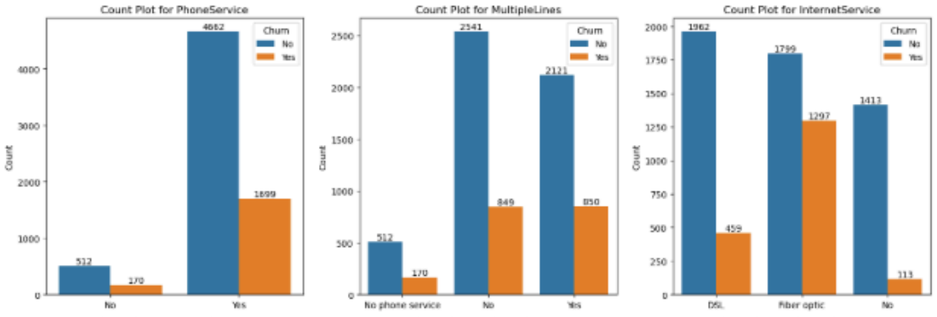
In [80]:
```
plt.figure(figsize=(4,4))
ax=sns.countplot(x='Contract',data=df,hue='Churn')
ax.bar_label(ax.containers[0])
plt.show()
```

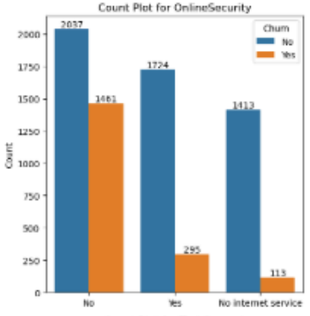


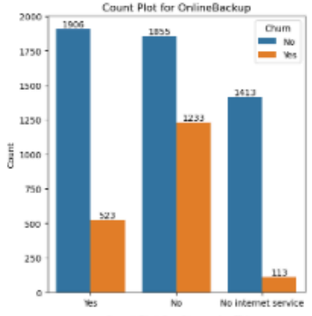
people who have month to month contract are likely to churn then from those who have 1 or 2 year contract

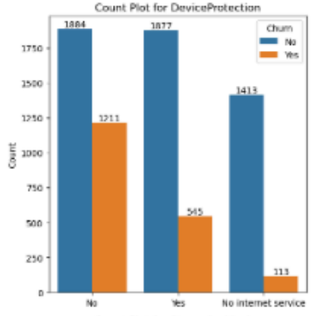
In [84]:

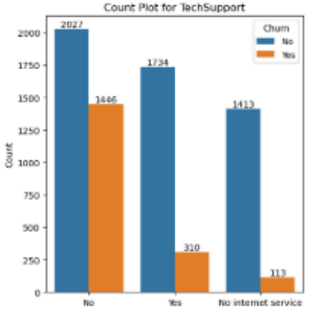
```
# List of columns to plot
columns = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

# Create a grid for subplots
n_cols = 3  # Number of columns per row in the grid
n_rows = (len(columns) + n_cols - 1) // n_cols  # Calculate the number of rows needed

fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 5 * n_rows))
axes = axes.flatten()  # Flatten the 2D array of axes to 1D for easier iteration

# Loop through columns and create count plots
for idx, col in enumerate(columns):
    sns.countplot(data=df, x=col, ax=axes[idx],hue="Churn")
    axes[idx].set_title(f'Count Plot for {col}')
    axes[idx].set_xlabel('')
    axes[idx].set_ylabel('Count')
    # Add counts on the bars
    for container in axes[idx].containers:
        axes[idx].bar_label(container, fmt='%d')

# Hide any unused subplots
for i in range(len(columns), len(axes)):
    fig.delaxes(axes[i])

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

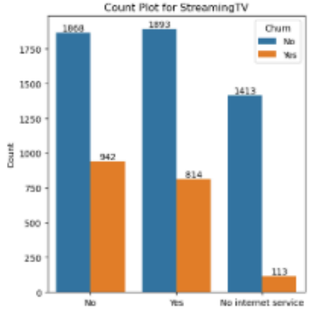


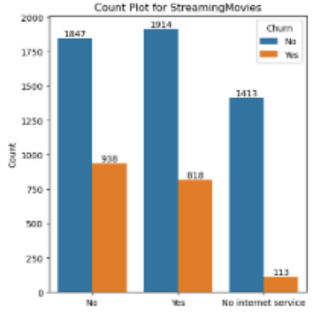






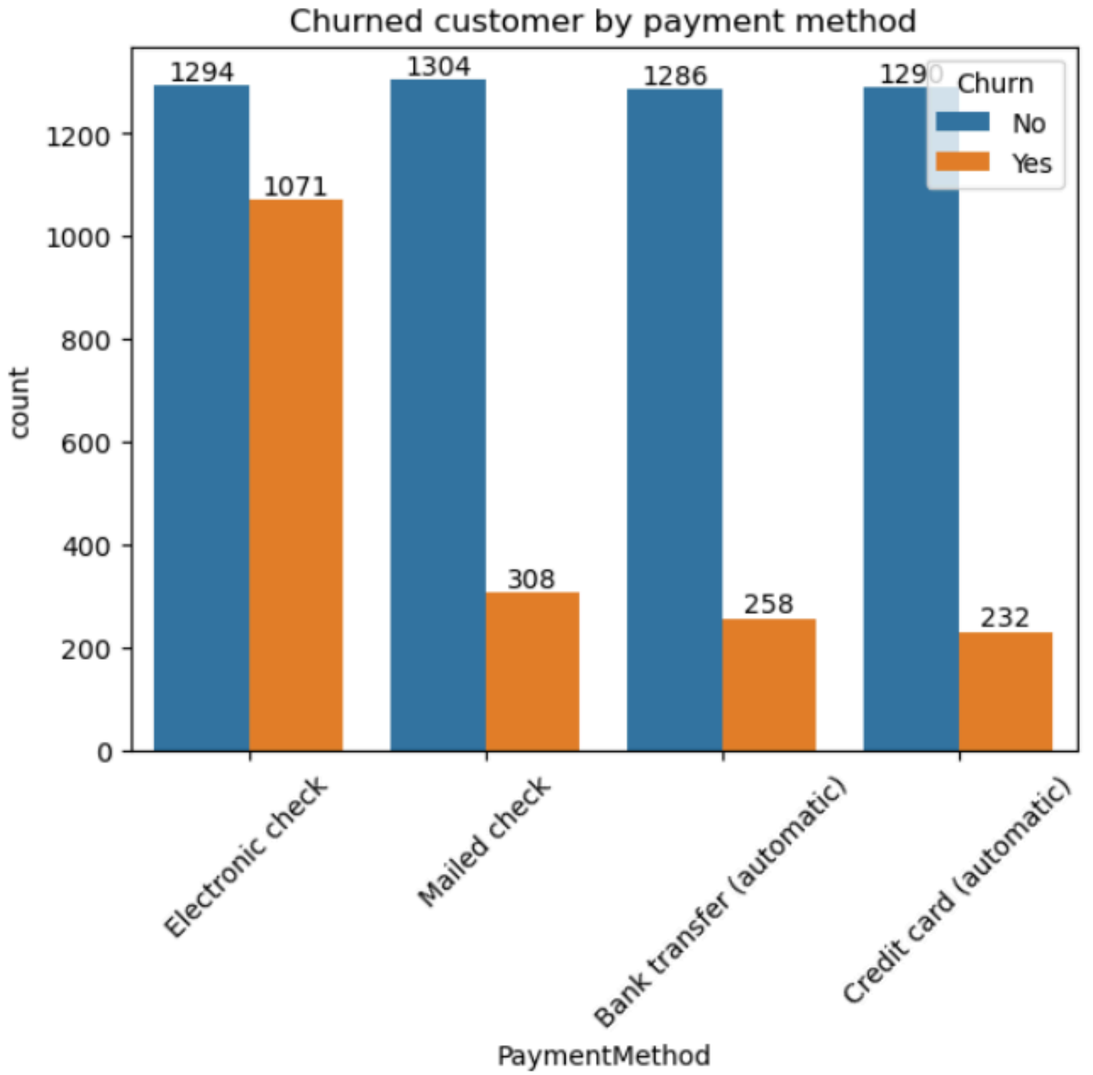






The majority of customers have phone services and prefer DSL or Fiber optic for internet. Optional services like Online Security, Backup, and Tech Support show a balanced split, but churn is higher among those not using these services. Enhancing optional services could help reduce churn rates.

In [89]:
```
ax=sns.countplot(x='PaymentMethod',data=df,hue='Churn')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.xticks(rotation=45)
plt.title("Churned customer by payment method")
plt.show()
```



customer is likely to churn when he is using electronic check as paymentmethod