

# Telecom Customers Churn Analysis By EDA Perform

An abstract graphic design featuring several white geometric shapes on a yellow background. The shapes include a horizontal rectangle at the top right, a diagonal bar extending from the top right towards the bottom left, and a vertical bar on the far right. These shapes overlap and create a modern, minimalist aesthetic.

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

```
df = pd.read_csv('C:/Users/mohsin/Downloads/Customer Churn.csv')
```

```
df.head()
```

```
✓ [9] 26ms
```

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

5 rows × 21 columns

```
df.info()
```

```
✓ [10] 10ms
```

```
<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

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

ore tool windows blanks with 0 as tenure is 0 and no total charges are recorded

```
1 df["TotalCharges"] = df["TotalCharges"].replace(" ", "0")
2 df["TotalCharges"] = df["TotalCharges"].astype("float")
✓ [11] < 10 ms
```

```
1 df.info()
✓ [12] 10ms
```

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

```
1 df.isnull().sum().sum()
✓ [13] < 10 ms
```

np.int64(0)

```
1 df.describe()
```

✓ [14] 18ms

	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

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

✓ [15] < 10 ms

```
np.int64(0)
```

```
1 def conv(value):
2     if value == 1:
3         return "yes"
4     else:
5         return "no"
6
7 df['SeniorCitizen'] = df["SeniorCitizen"].apply(conv)
✓ [16] < 10 ms
```

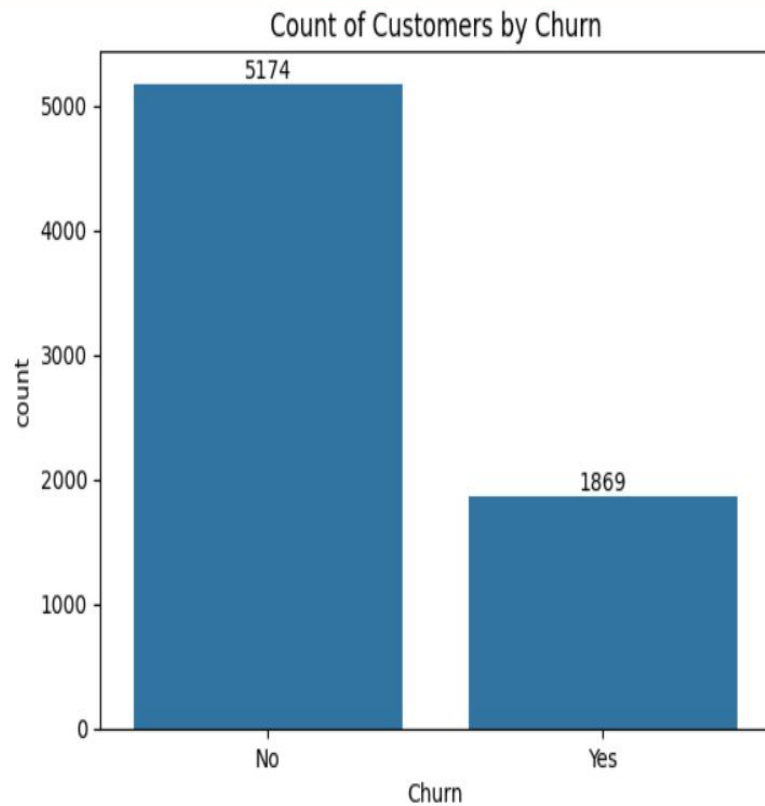
#converted 0 and 1 values of senior citizen to yes/no to make it easier to understand

```
df.columns.values
```

```
✓ [25] < 10 ms
```

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

```
1 ax = sns.countplot(x = 'Churn', data = df)
2 ax.bar_label(ax.containers[0])
3 plt.title("Count of Customers by Churn")
4 plt.show()
✓ [17] 95ms
```

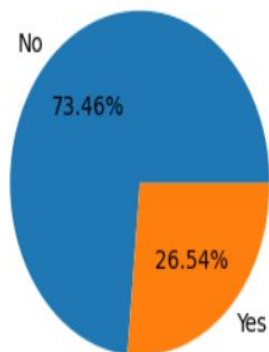


Code Markdown



```
1 plt.figure(figsize = (3,4))
2 gb = df.groupby("Churn").agg({'Churn':"count"})
3 plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")
4 plt.title("Percentage of Churned Customeres", fontsize = 10)
5 plt.show()
✓ [18] 31ms
```

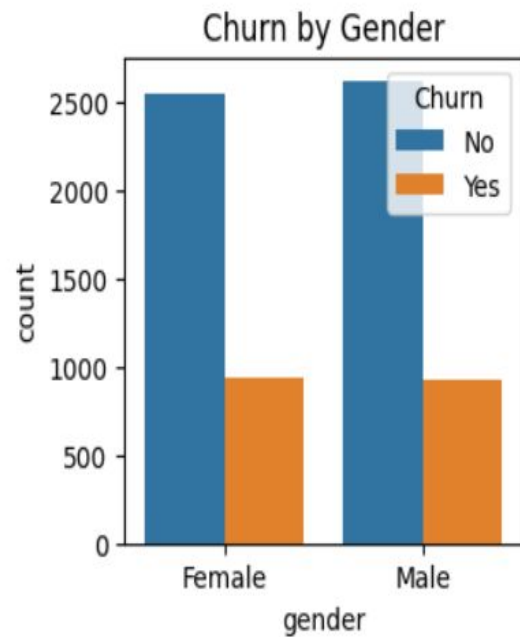
Percentage of Churned Customeres



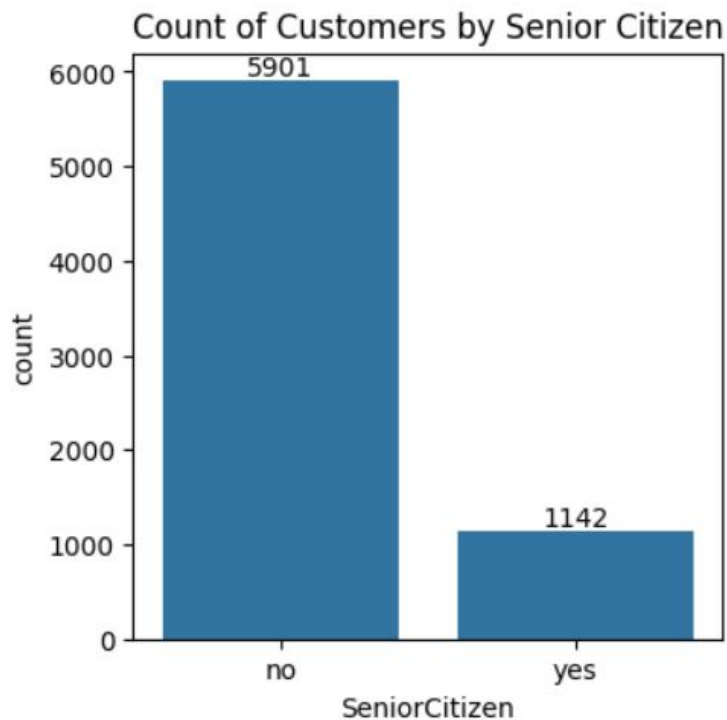
#from the given pie chart we can conclude that 26.54% of our customers have churned out. #not let's explore the reason behind it



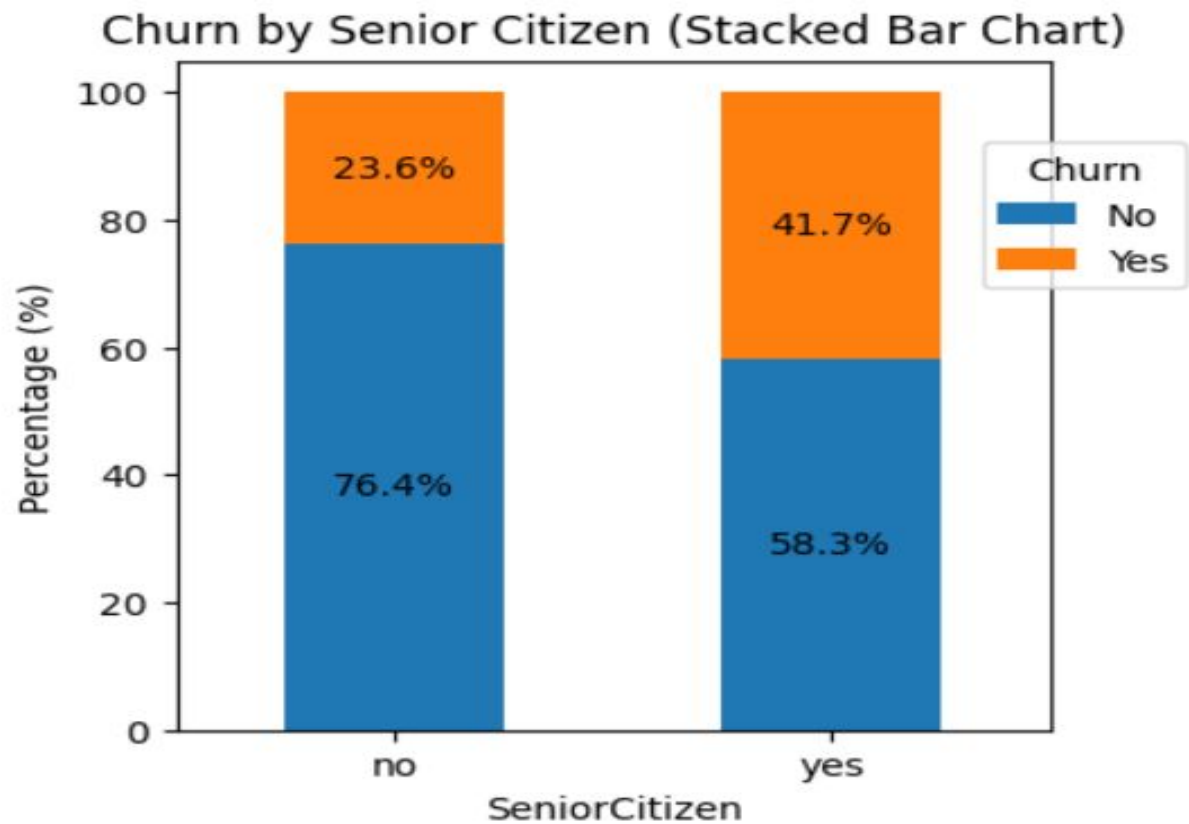
```
1 plt.figure(figsize = (3,3))
2 sns.countplot(x = "gender", data = df, hue = "Churn")
3 plt.title("Churn by Gender")
4 plt.show()
✓ [19] 75ms
```



```
1 plt.figure(figsize = (4,4))
2 ax = sns.countplot(x = "SeniorCitizen", data = df)
3 ax.bar_label(ax.containers[0])
4 plt.title("Count of Customers by Senior Citizen")
5 plt.show()
✓ [20] 61ms
```



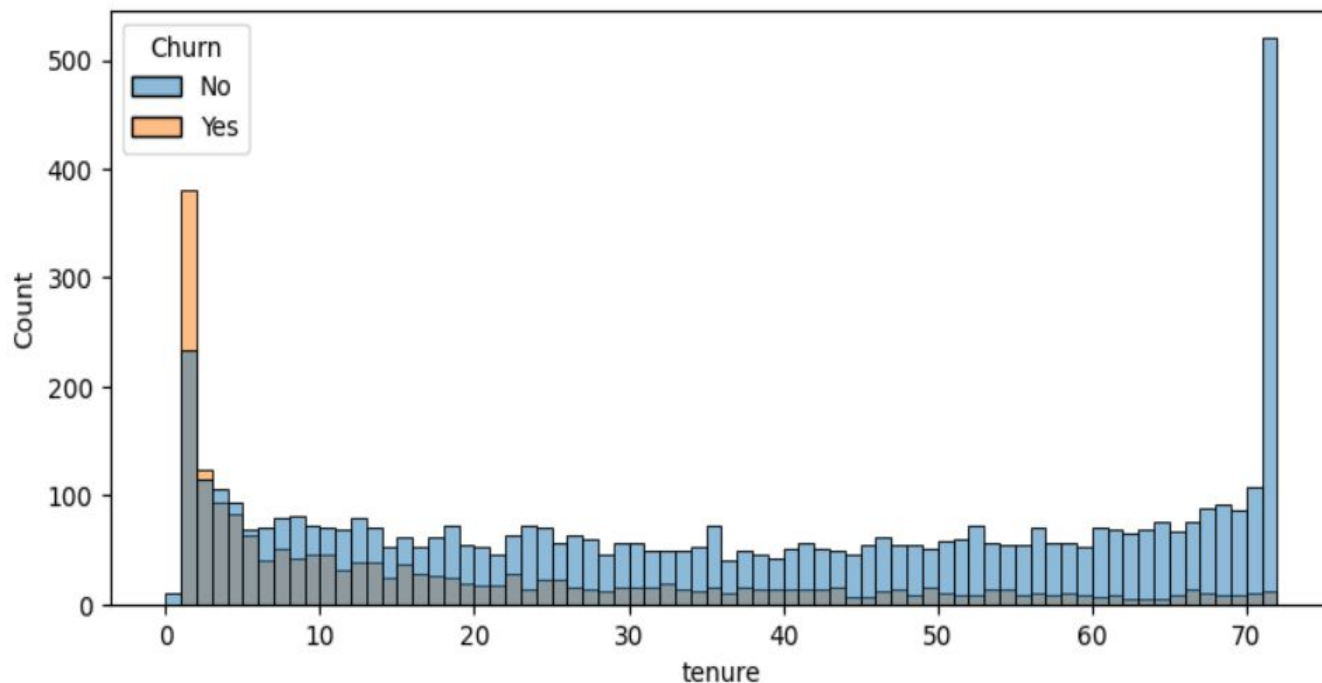
```
total_counts.plot(kind='bar', stacked=True, ax=ax, color=[ '#1f77b4', '#ff7f0e' ]) # customize colors if desired
5 # Add percentage labels on the bars
6 for p in ax.patches:
7     width, height = p.get_width(), p.get_height()
8     x, y = p.get_xy()
9     ax.text(x + width / 2, y + height / 2, f'{height:.1f}%', ha='center', va='center')
10 plt.title('Churn by Senior Citizen (Stacked Bar Chart)')
11 plt.xlabel('SeniorCitizen')
12 plt.ylabel('Percentage (%)')
13 plt.xticks(rotation=0)
14 plt.legend(title='Churn', bbox_to_anchor = (0.9,0.9)) # Customize legend location
15 plt.show()
✓ [21] 87ms
```



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

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

✓ [22] 138ms



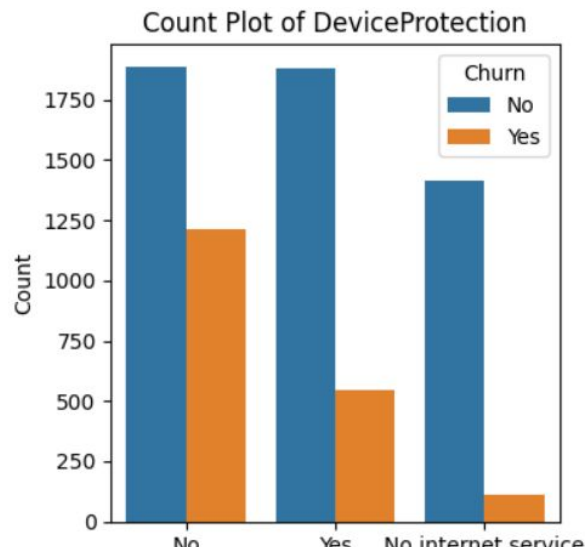
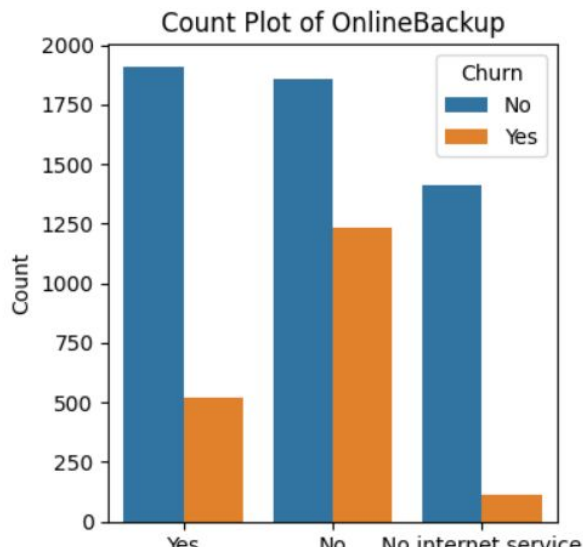
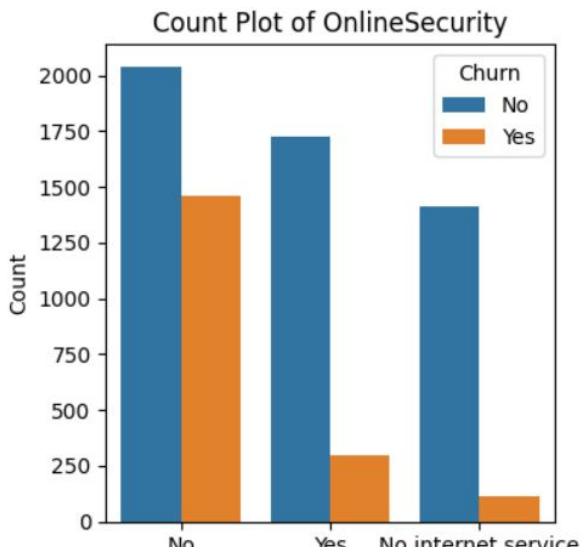
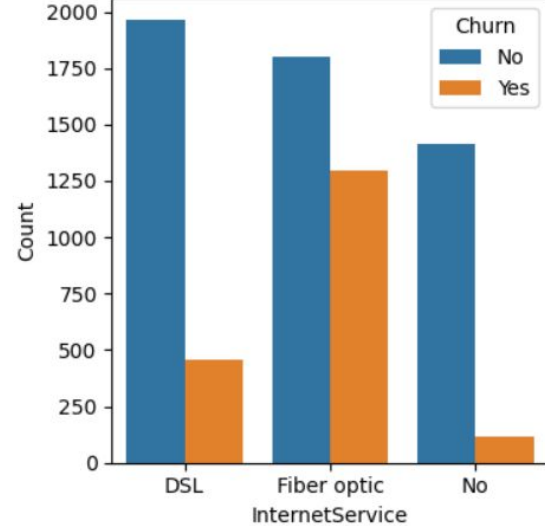
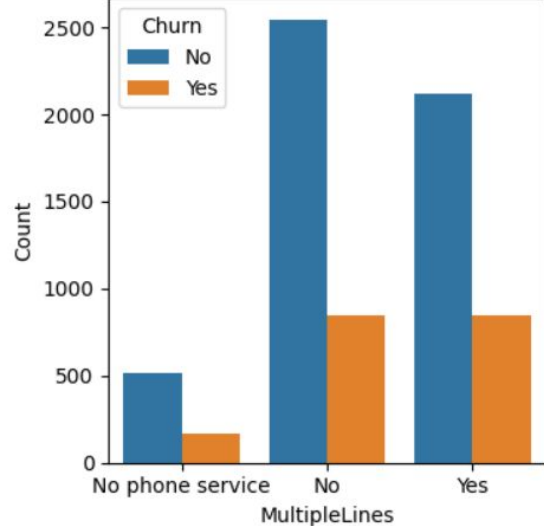
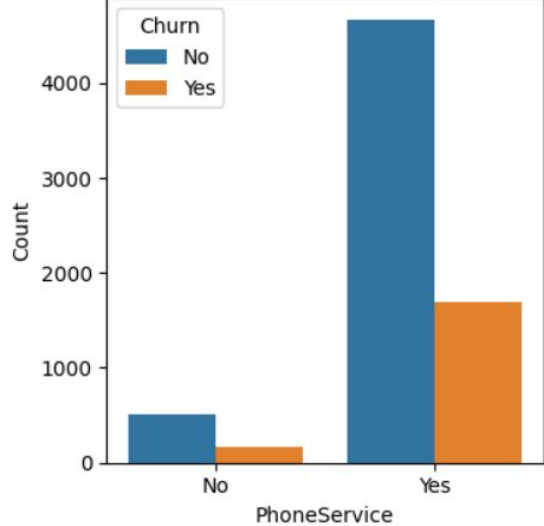
#people who have used our services for a long time have stayed and people who have used our services #1 or 2 months have churned

```
1 plt.figure(figsize = (4,4))
2 ax = sns.countplot(x = "Contract", data = df, hue = "Churn")
3 ax.bar_label(ax.containers[0])
4 plt.title("Count of Customers by Contract")
5 plt.show()
✓ [23] 87ms
```



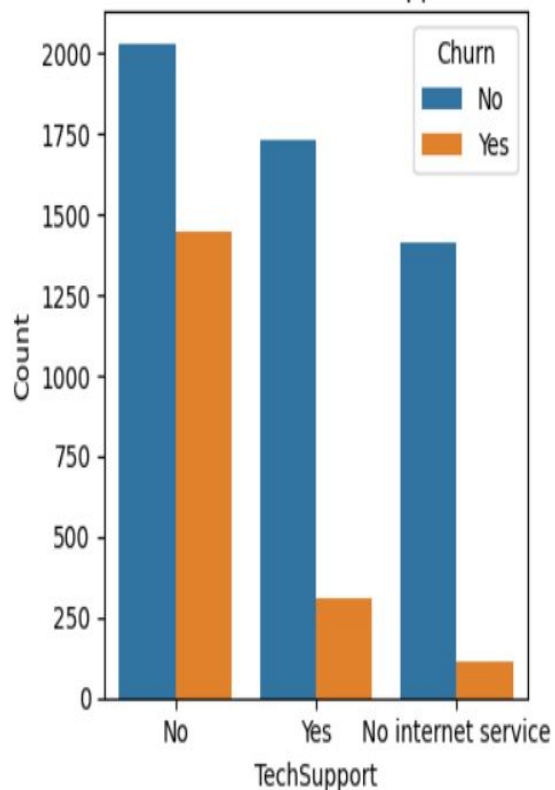
```
1 #people who have month to month contract are likely to churn then from those who have 1 or 2 years or contract.
✓ [24] < 10 ms
```

```
columns = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',  
           'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']  
  
# Number of columns for the subplot grid (you can change this)  
n_cols = 3  
n_rows = (len(columns) + n_cols - 1) // n_cols # Calculate number of rows needed  
  
# Create subplots  
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 4)) # Adjust figsize as needed  
  
# Flatten the axes array for easy iteration (handles both 1D and 2D arrays)  
axes = axes.flatten()  
  
# Iterate over columns and plot count plots  
for i, col in enumerate(columns):  
    sns.countplot(x=col, data=df, ax=axes[i], hue = df["Churn"])  
    axes[i].set_title(f'Count Plot of {col}')  
    axes[i].set_xlabel(col)  
    axes[i].set_ylabel('Count')  
  
# Remove empty subplots (if any)  
for j in range(1, len(axes)):  
    fig.delaxes(axes[j])  
  
plt.tight_layout()  
plt.show()  
✓ [26] 739ms
```

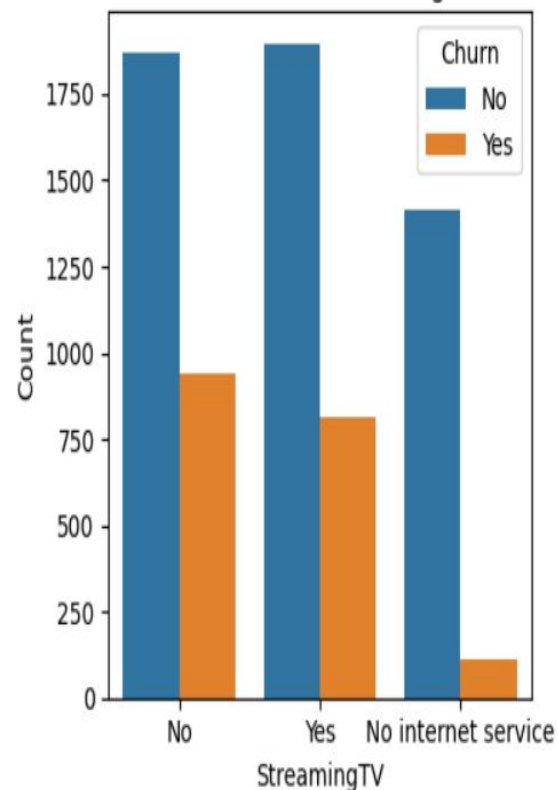




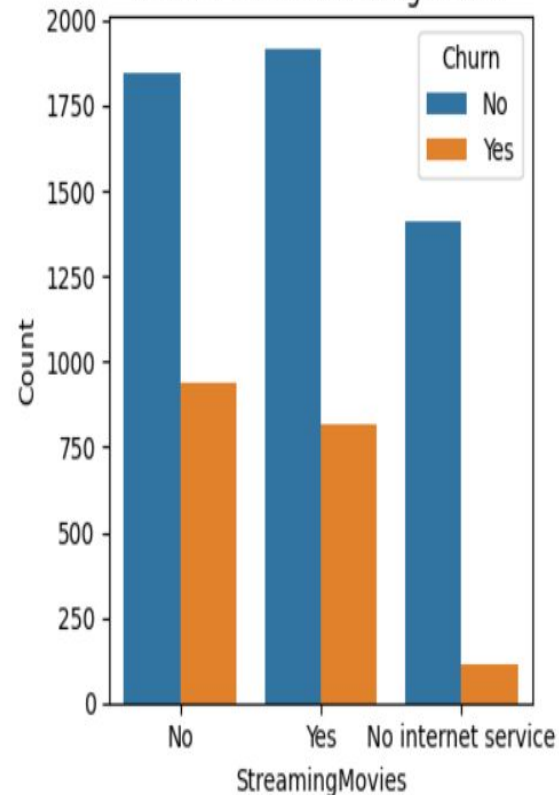
Count Plot of TechSupport



Count Plot of StreamingTV



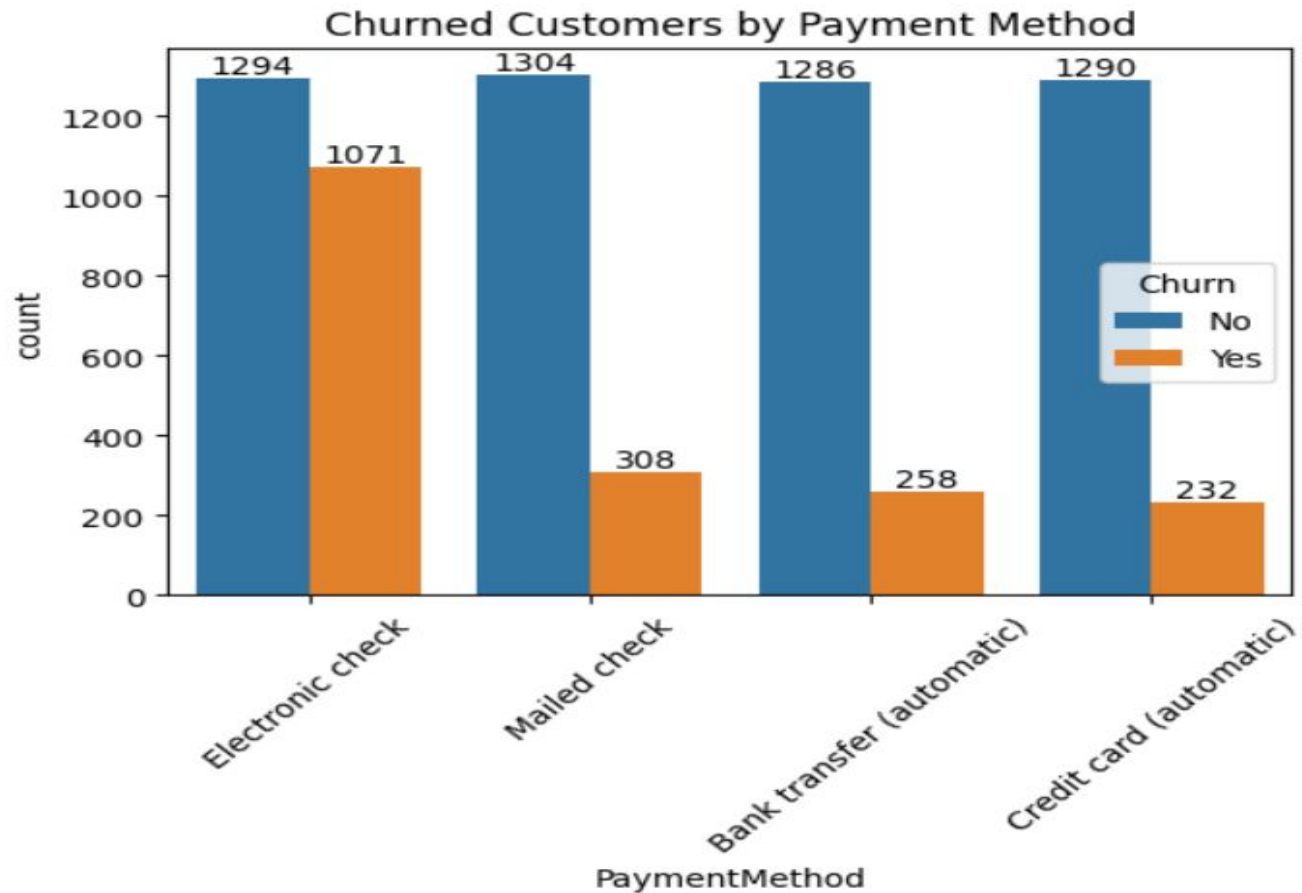
Count Plot of StreamingMovies



#The majority of customers who do not churn tend to have services like PhoneService, InternetService (particularly DSL), and OnlineSecurity enabled. For services like OnlineBackup, TechSupport, and StreamingTV, churn rates are noticeably higher when these services are not used or are unavailable.

```
1 plt.figure(figsize = (6,4))
2 ax = sns.countplot(x = "PaymentMethod", data = df, hue = "Churn")
3 ax.bar_label(ax.containers[0])
4 ax.bar_label(ax.containers[1])
5 plt.title("Churned Customers by Payment Method")
6 plt.xticks(rotation = 45)
7 plt.show()
```

✓ [27] 108ms



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