

```
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
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	\
0	7590-VHVEG	Female	0	Yes	No	1	
1	5575-GNVDE	Male	0	No	No	34	
2	3668-QPYBK	Male	0	No	No	2	
3	7795-CF0CW	Male	0	No	No	45	
4	9237-HQITU	Female	0	No	No	2	
...	
7038	6840-RESVB	Male	0	Yes	Yes	24	
7039	2234-XADUH	Female	0	Yes	Yes	72	
7040	4801-JZAZL	Female	0	Yes	Yes	11	
7041	8361-LTMKD	Male	1	Yes	No	4	
7042	3186-AJIEK	Male	0	No	No	66	

	PhoneService	MultipleLines	InternetService
OnlineSecurity	...	\	
0	No	No phone service	DSL
No	...		
1	Yes	No	DSL
Yes	...		
2	Yes	No	DSL
Yes	...		
3	No	No phone service	DSL
Yes	...		
4	Yes	No	Fiber optic
No	...		
...
...
7038	Yes	Yes	DSL
Yes	...		
7039	Yes	Yes	Fiber optic
No	...		
7040	No	No phone service	DSL
Yes	...		
7041	Yes	Yes	Fiber optic
No	...		
7042	Yes	No	Fiber optic
Yes	...		

	DeviceProtection	TechSupport	StreamingTV	StreamingMovies
Contract	\			
0	No	No	No	No
to-month				Month-

1	Yes	No	No	No	
One year					
2	No	No	No	No	Month-
to-month					
3	Yes	Yes	No	No	
One year					
4	No	No	No	No	Month-
to-month					
...	
...					
7038	Yes	Yes	Yes	Yes	
One year					
7039	Yes	No	Yes	Yes	
One year					
7040	No	No	No	No	Month-
to-month					
7041	No	No	No	No	Month-
to-month					
7042	Yes	Yes	Yes	Yes	
Two year					
PaperlessBilling		PaymentMethod		MonthlyCharges	
TotalCharges \					
0	Yes	Electronic check		29.85	
29.85					
1	No	Mailed check		56.95	
1889.5					
2	Yes	Mailed check		53.85	
108.15					
3	No	Bank transfer (automatic)		42.30	
1840.75					
4	Yes	Electronic check		70.70	
151.65					
...	
...					
7038	Yes	Mailed check		84.80	
1990.5					
7039	Yes	Credit card (automatic)		103.20	
7362.9					
7040	Yes	Electronic check		29.60	
346.45					
7041	Yes	Mailed check		74.40	
306.6					
7042	Yes	Bank transfer (automatic)		105.65	
6844.5					
Churn					
0	No				
1	No				

```

2      Yes
3      No
4      Yes
...    ...
7038   No
7039   No
7040   No
7041   Yes
7042   No

```

[7043 rows x 21 columns]

df.head() *# it will show 5 rows table*

```

   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-CF0CW   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

```
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()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure \
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...
7038	False	False	False	False	False	False
7039	False	False	False	False	False	False
7040	False	False	False	False	False	False

7041	False	False	False	False	False	False
7042	False	False	False	False	False	False
<div>PhoneServiceMultipleLinesInternetServiceOnlineSecurity \</div>						
0	False	False	False	False		
False	...					
1	False	False	False	False		
False	...					
2	False	False	False	False		
False	...					
3	False	False	False	False		
False	...					
4	False	False	False	False		
False	...					
...
.						
7038	False	False	False	False		
False	...					
7039	False	False	False	False		
False	...					
7040	False	False	False	False		
False	...					
7041	False	False	False	False		
False	...					
7042	False	False	False	False		
False	...					
<div>DeviceProtectionTechSupportStreamingTVStreamingMoviesContract \</div>						
0	False	False	False	False	False	
False						
1	False	False	False	False	False	
False						
2	False	False	False	False	False	
False						
3	False	False	False	False	False	
False						
4	False	False	False	False	False	
False						
...	
...						
7038	False	False	False	False	False	
False						
7039	False	False	False	False	False	
False						
7040	False	False	False	False	False	
False						

7041	False	False	False	False
False				
7042	False	False	False	False
False				
	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges
Churn				
0	False	False	False	False
False				
1	False	False	False	False
False				
2	False	False	False	False
False				
3	False	False	False	False
False				
4	False	False	False	False
False				
...
...				
7038	False	False	False	False
False				
7039	False	False	False	False
False				
7040	False	False	False	False
False				
7041	False	False	False	False
False				
7042	False	False	False	False
False				

```
[7043 rows x 21 columns]
```

```
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

```
df.duplicated().sum()
```

```
0
```

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

```
0
```

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

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

```
df.head(10)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure
0	7590-VHVEG	Female	no	Yes	No	1
1	5575-GNVDE	Male	no	No	No	34
2	3668-QPYBK	Male	no	No	No	2
3	7795-CF0CW	Male	no	No	No	45
4	9237-HQITU	Female	no	No	No	2
5	9305-CDSKC	Female	no	No	No	8
6	1452-KIOVK	Male	no	No	Yes	22
7	6713-OKOMC	Female	no	No	No	10
8	7892-P00KP	Female	no	Yes	No	28
9	6388-TABGU	Male	no	No	Yes	62

	MultipleLines	InternetService	OnlineSecurity	...
0	No phone service	DSL	No	...
1	No	DSL	Yes	...
2	No	DSL	Yes	...
3	No phone service	DSL	Yes	...
4	No	Fiber optic	No	...
5	Yes	Fiber optic	No	...

6	Yes	Fiber optic	No	...
No				
7	No phone service	DSL	Yes	...
No				
8	Yes	Fiber optic	No	...
Yes				
9	No	DSL	Yes	...
No				

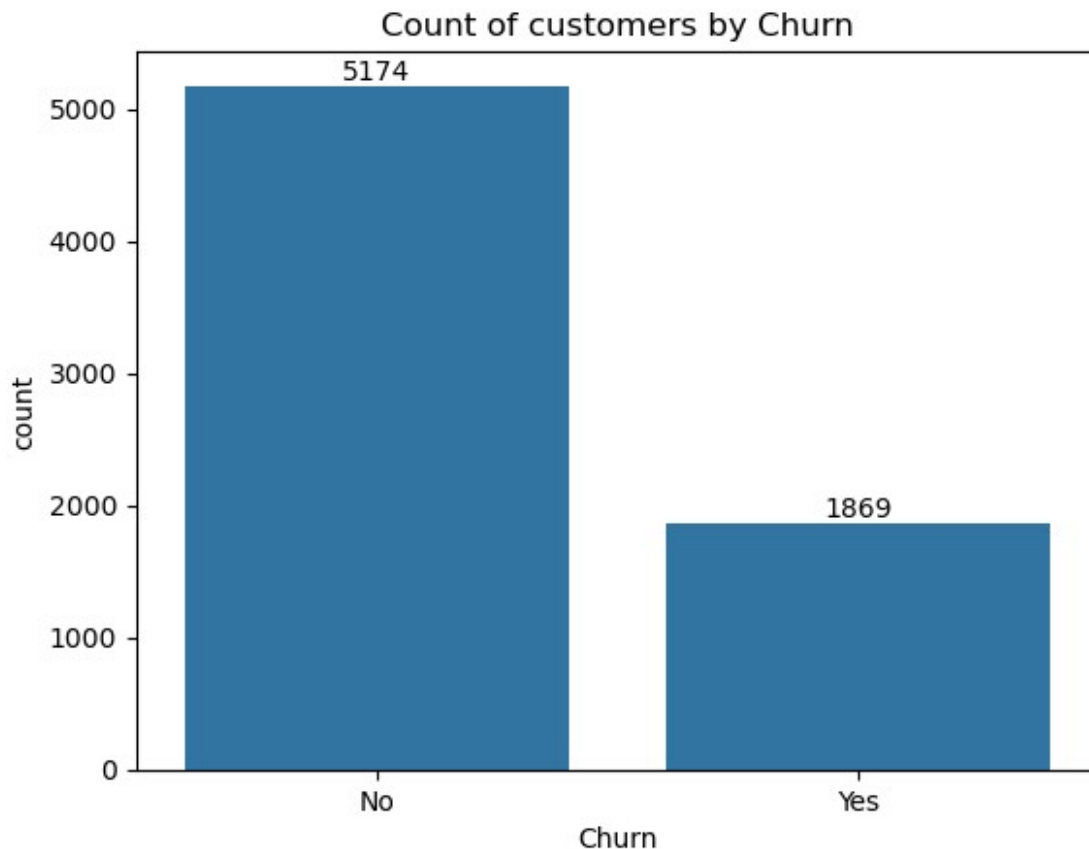
	TechSupport	StreamingTV	StreamingMovies	Contract
PaperlessBilling \				
0	No	No	No	Month-to-month
Yes				
1	No	No	No	One year
No				
2	No	No	No	Month-to-month
Yes				
3	Yes	No	No	One year
No				
4	No	No	No	Month-to-month
Yes				
5	No	Yes	Yes	Month-to-month
Yes				
6	No	Yes	No	Month-to-month
Yes				
7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

converted 0 and 1 value of senior citizen to yes/no to make to understand

```
ax=sns.countplot(x='Churn',data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of customers by Churn")
plt.show()
```

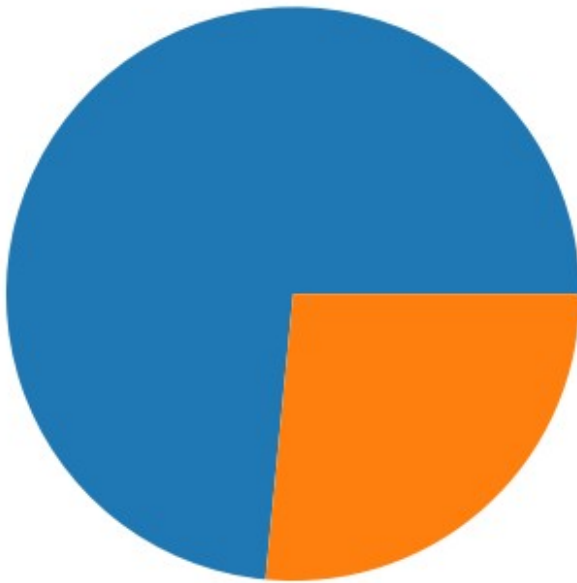


From The Above Figure The number of customers who did not churn is 5,174, which is significantly higher than those who churned (1,869).

```
gb=df.groupby("Churn").agg({'Churn': "count"})
gb
```

Churn	
Churn	
No	5174
Yes	1869

```
gb=df.groupby("Churn").agg({'Churn': "count"})  
plt.pie(gb['Churn'])  
plt.show()
```



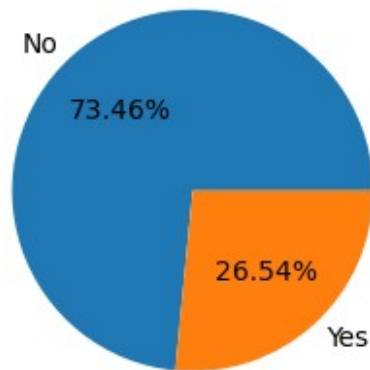
OR

The pie chart shows that most customers did not churn, while a smaller portion did. This suggests a retention advantage, but understanding churn reasons can help improve customer loyalty.

```
plt.figure(figsize=(3,4))  
gb=df.groupby("Churn").agg({'Churn': "count"})  
plt.pie(gb['Churn'], labels=gb.index, autopct="%1.2f%%")
```

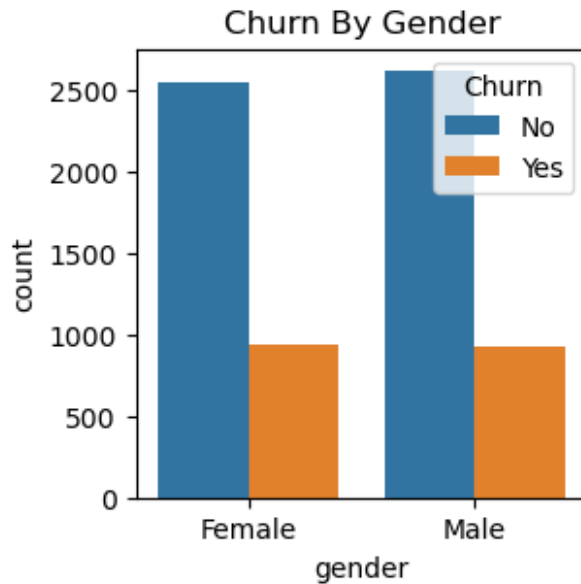
```
plt.title("percentage of Churned Customers",fontsize=10)  
plt.show()
```

percentage of Churned customers



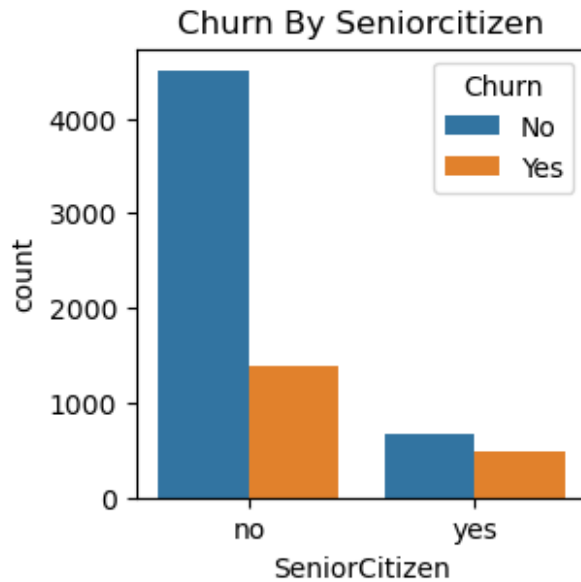
from the Above Figure pie chart we can conclude that 26.54% of our customers have churned out not Let's explore the reason behind it.

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



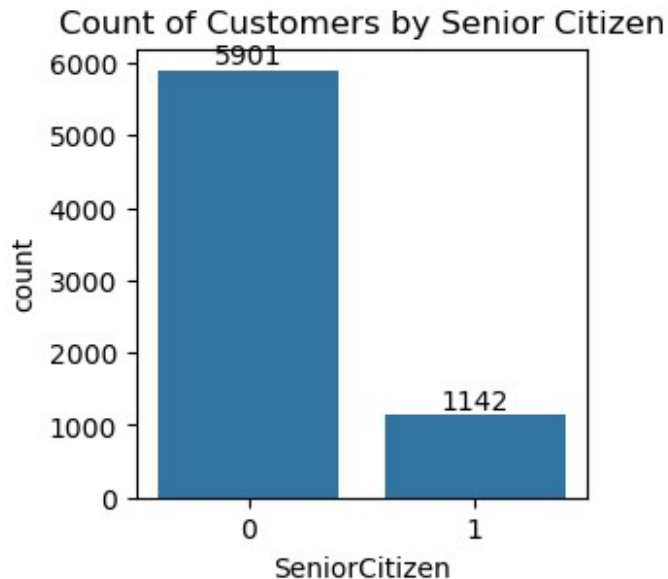
The chart shows that both male and female customers have similar churn rates, with more customers staying than leaving. This suggests that gender does not play a significant role in customer churn.

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



From The Above Figure Senior citizens have fewer customers, but their churn rate appears higher. Retention efforts may be needed for this group.

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



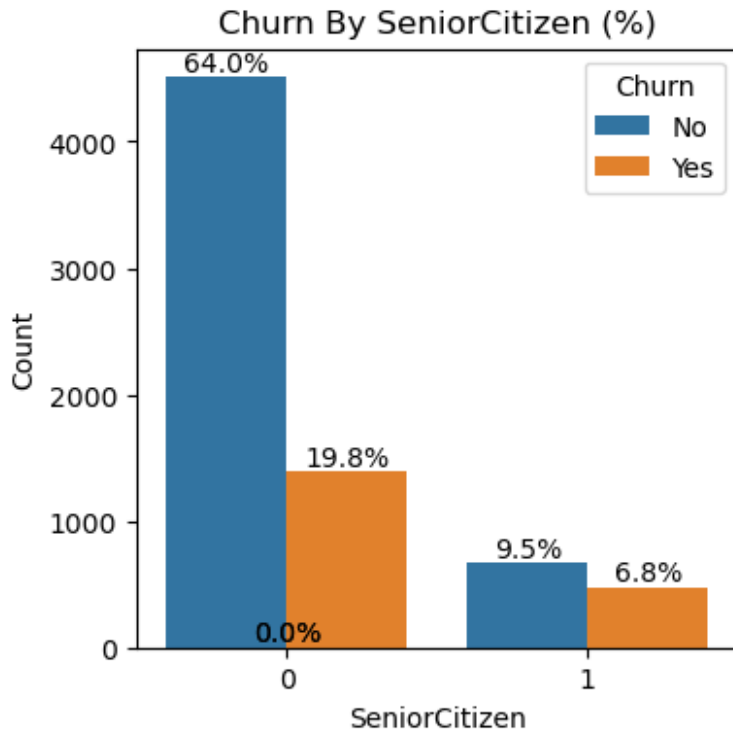
From The Above Figure The chart shows that the majority of customers are non-senior citizens (5,901), while senior citizens make up a smaller portion (1,142).

```
# Ensure you have a DataFrame named df before running this
plt.figure(figsize=(4, 4))
ax = sns.countplot(x='SeniorCitizen', data=df, hue='Churn')

# Calculate total counts for percentages
total_counts = len(df)

# Add percentage labels
for p in ax.patches:
    percentage = f'{100 * p.get_height() / total_counts:.1f}%'
    ax.annotate(percentage, (p.get_x() + p.get_width() / 2,
p.get_height()),
                ha='center', va='bottom', fontsize=10)

plt.title("Churn By SeniorCitizen (%)")
plt.ylabel("Count")
plt.show()
```



From The Above Figure Senior citizens have a lower overall count, but their churn percentage (6.8%) is relatively high compared to their total population.

```
# Calculate counts
counts = df.groupby(['SeniorCitizen', 'Churn']).size().unstack()

# Convert counts to percentages
percentages = counts.div(counts.sum(axis=1), axis=0) * 100

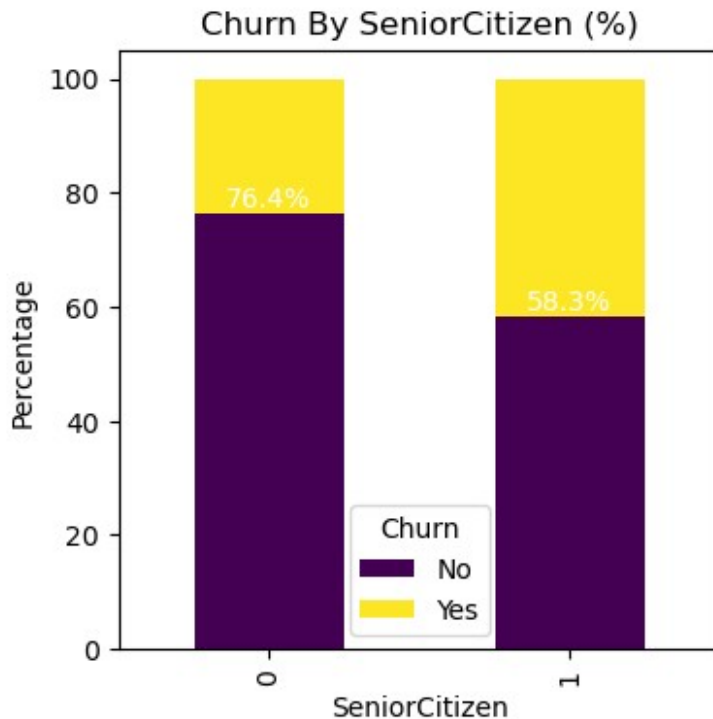
# Plot stacked bar chart
ax = percentages.plot(kind='bar', stacked=True, figsize=(4, 4),
colormap='viridis')

# Add percentage labels
for c in ax.containers:
    ax.bar_label(c, fmt='%.1f%%', label_type='edge', color='white',
    fontsize=10)

plt.title("Churn By SeniorCitizen (%)")
plt.xlabel("SeniorCitizen")
```

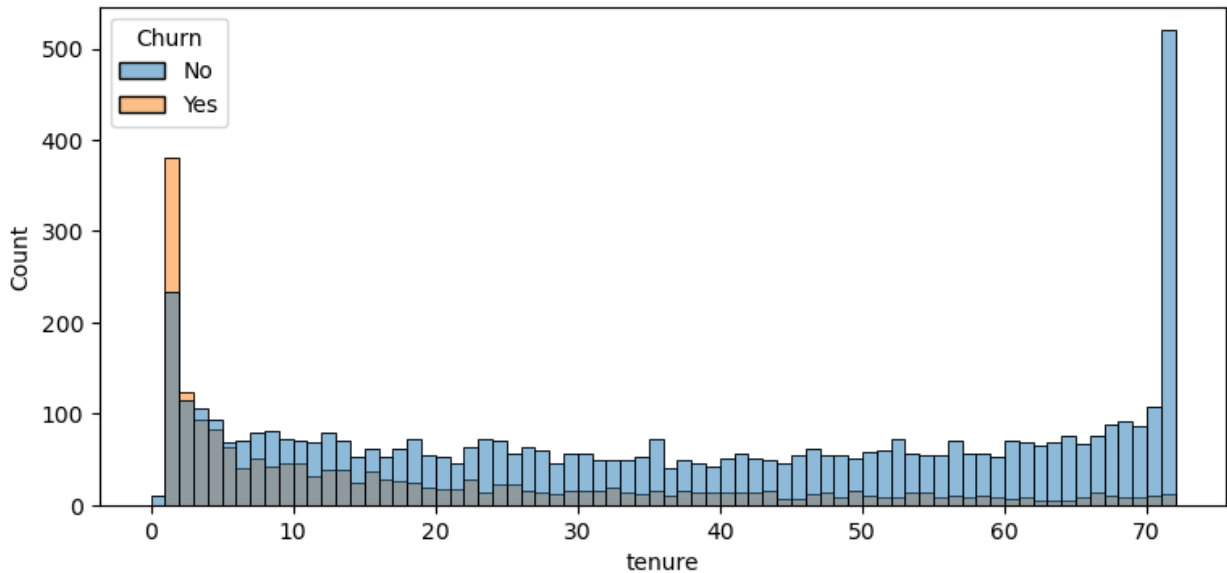


```
plt.ylabel("Percentage")
plt.legend(title="Churn")
plt.show()
```



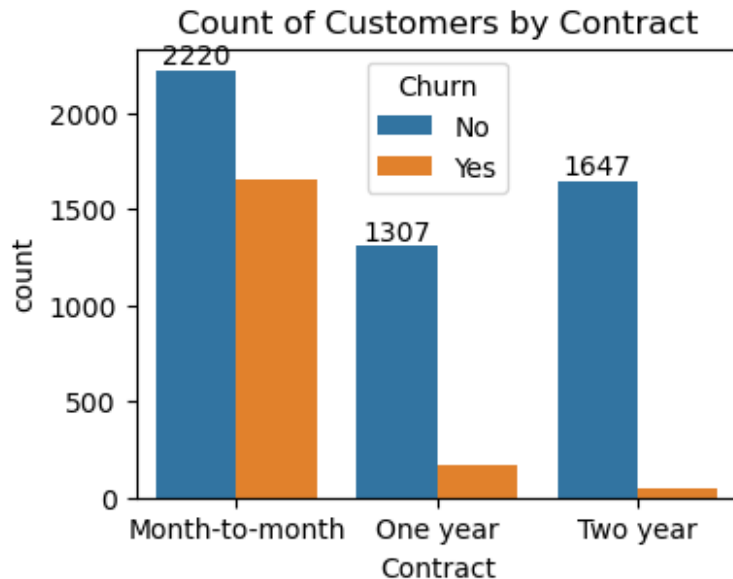
From The Figure Comarative a greated percantage of people in senior citizen category have churned

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



From The Above Figure People Who used our service for a long time have stayed and who have used our service 1 or 2 months have churned

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



From The Above The Chart people who have month contract are likely to churn then 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)

# Sample DataFrame (Replace with your actual dataset)
df = pd.DataFrame({
    'PhoneService': ['Yes', 'No', 'Yes', 'Yes', 'No'],
    'MultipleLines': ['No', 'Yes', 'No', 'No', 'Yes'],
    'InternetService': ['DSL', 'Fiber optic', 'DSL', 'None', 'Fiber
optic'],
    'OnlineSecurity': ['No', 'Yes', 'Yes', 'No', 'No'],
    'OnlineBackup': ['Yes', 'No', 'Yes', 'No', 'Yes'],
    'DeviceProtection': ['No', 'Yes', 'No', 'Yes', 'No'],
    'TechSupport': ['Yes', 'No', 'No', 'Yes', 'No'],
    'StreamingTV': ['No', 'Yes', 'Yes', 'No', 'No'],
    'StreamingMovies': ['Yes', 'No', 'No', 'Yes', 'No']
})
```

```

}))

# Define the number of rows and columns for the subplots
num_cols = 3 # Number of columns in the subplot grid
num_rows = -(-len(df.columns) // num_cols) # Calculate rows needed

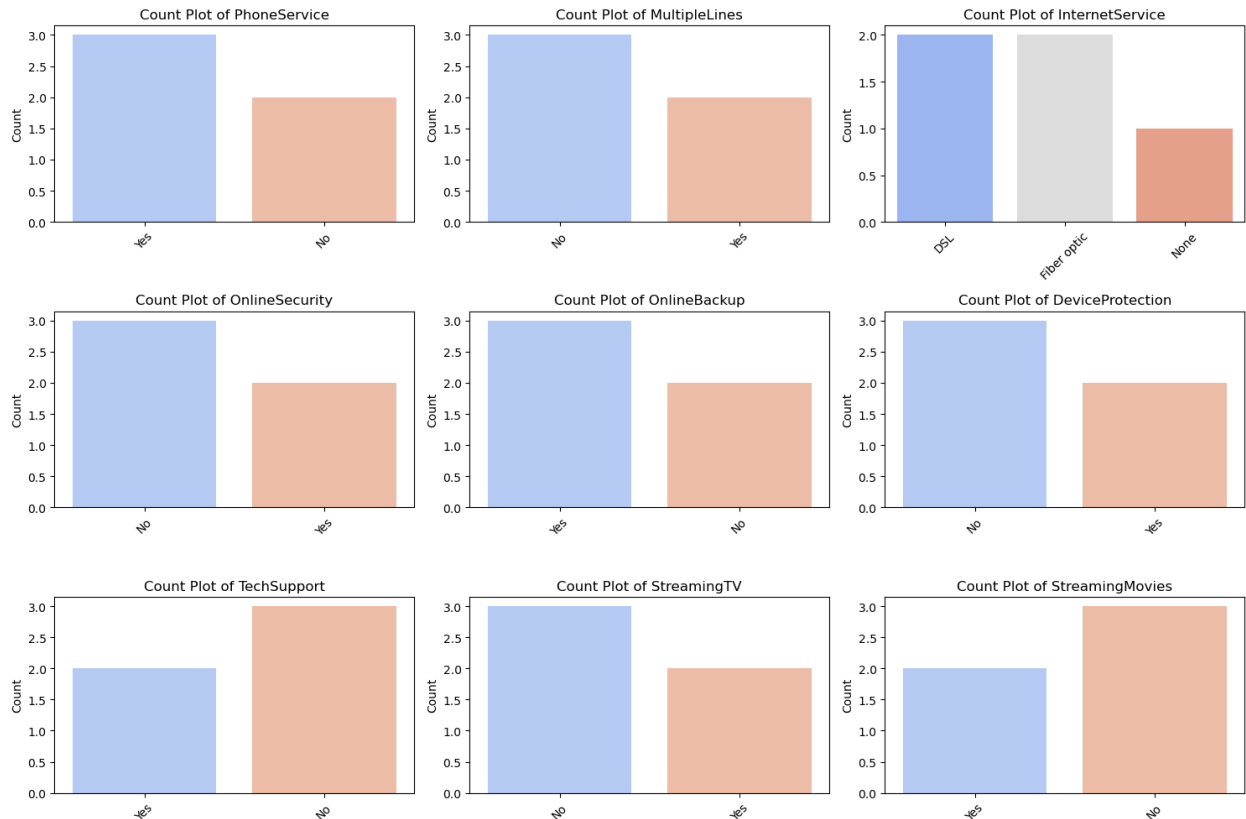
# Create subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 10))
axes = axes.flatten() # Flatten for easy indexing

# Loop through columns and create count plots
for i, col in enumerate(df.columns):
    sns.countplot(data=df, x=col, ax=axes[i], hue=col,
palette="coolwarm", legend=False)
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')
    axes[i].tick_params(axis='x', rotation=45) # Rotate x-axis labels
for readability

# Remove empty subplots if any
for i in range(len(df.columns), len(axes)):
    fig.delaxes(axes[i])

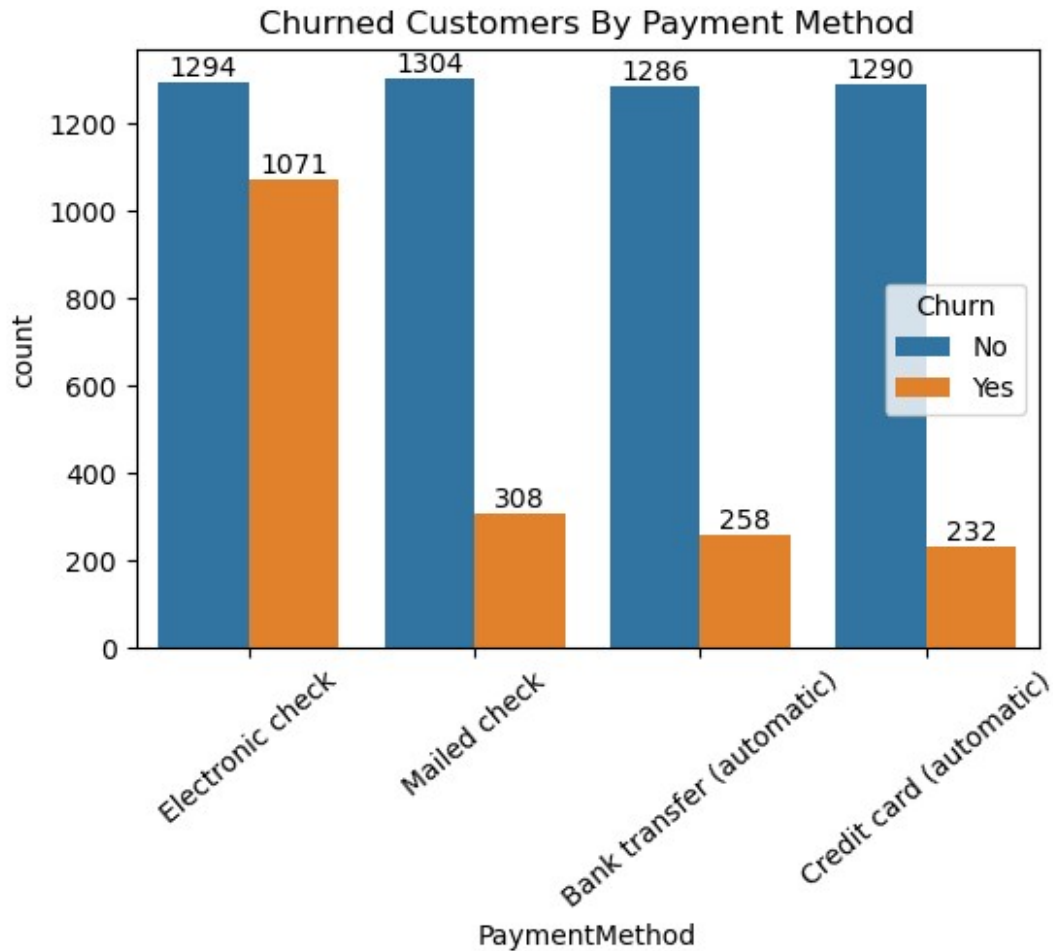
plt.tight_layout()
plt.show()

```



From The Above Figure The majority of customers who do not churn to have services like PhoneServices InternetService(particularly DSL),and Online Security enabled OnlineBackup TechSupport and Streaming Tv.churn rates are noticeably higher when services are not used or are unvaialiable

```
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 Customers By Payment Method")
plt.xticks(rotation=40)
plt.show()
```



From The above figure shows that Customer is likely to churn when he is using electronic check as a payment method