```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (2.3.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from pandas)
(2.3.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from pandas)
(2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from pandas)
(2025.2)
Requirement already satisfied: six>=1.5 in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (from python-
dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (3.10.6)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from matplotlib)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib) (4.60.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib) (1.4.9)
Requirement already satisfied: numpy>=1.23 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from matplotlib)
(2.3.2)
Requirement already satisfied: packaging>=20.0 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from matplotlib)
(25.0)
Requirement already satisfied: pillow>=8 in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib) (3.2.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (from python-
dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
seaborn) (2.3.2)
Requirement already satisfied: pandas>=1.2 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from seaborn)
(2.3.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
seaborn) (3.10.6)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.3.3)
Requirement already satisfied: cycler>=0.10 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (4.60.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.9)
Requirement already satisfied: packaging>=20.0 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (25.0)
Requirement already satisfied: pillow>=8 in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (3.2.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\pc\
appdata\local\programs\python\python313\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from pandas>=1.2-
>seaborn) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\pc\appdata\
local\programs\python\python313\lib\site-packages (from pandas>=1.2-
>seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (from python-
dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\pc\appdata\local\
programs\python\python313\lib\site-packages (2.3.2)
Note: you may need to restart the kernel to use updated packages.
```

```python
import pandas as pd
```

```python
import matplotlib.pyplot as plt
```

```python
import seaborn as sns
```

```python
import numpy as np
```

```python
df=pd.read_csv('customer_churn.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   float64
 20  Churn             7043 non-null   object 
dtypes: float64(2), int64(2), object(17)
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().sum().sum()
```

```
np.int64(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
```

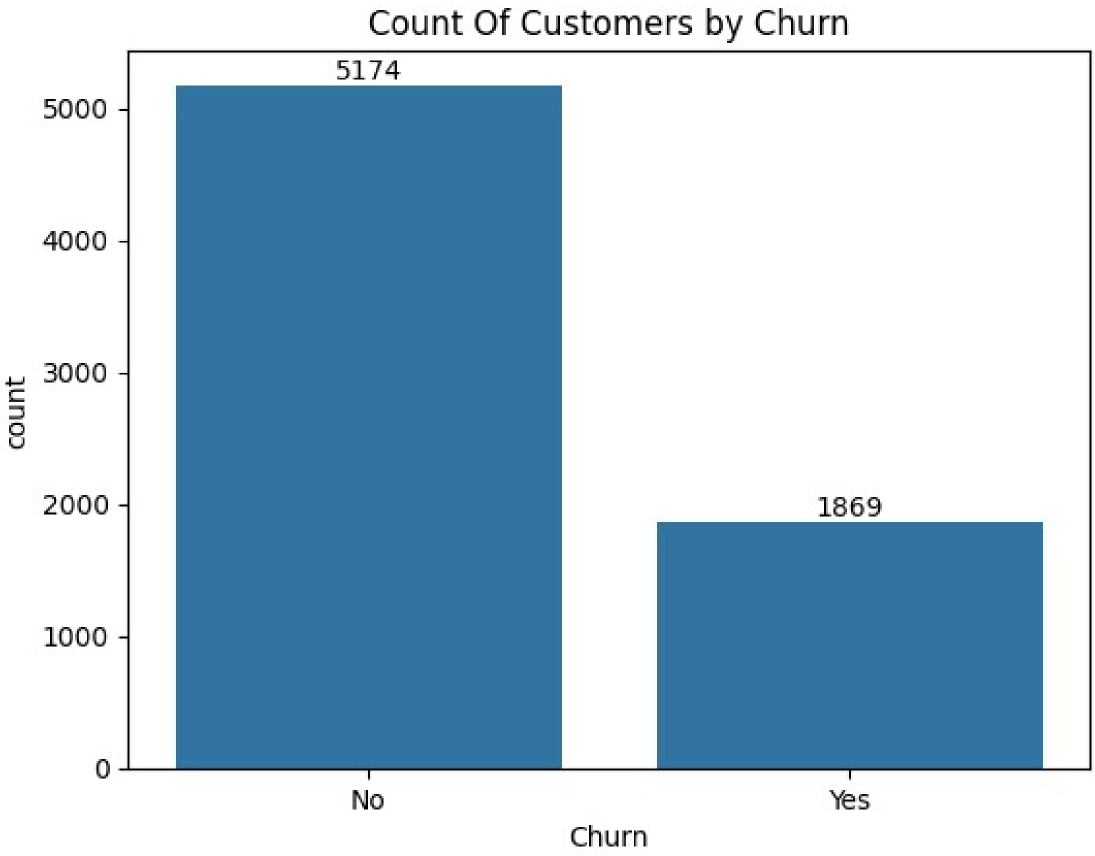
```
df["customerID"].duplicated().sum()
```

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

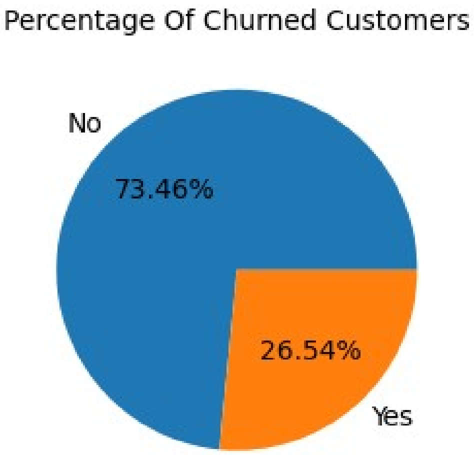
```
def conv(value):
    if value == 1:
        return "yes"
    else:
        return "no"

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

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

```
ax=sns.countplot(x= 'Churn', data = df)

ax.bar_label(ax.containers[0])
plt.title("Count Of Customers by Churn")

plt.show()
```



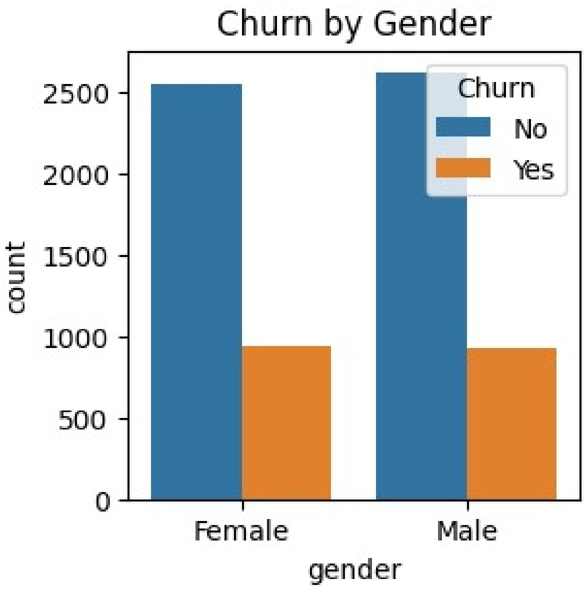
```
plt.figure(figsize = (3,4))
gb = df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")
plt.title("Percentage Of Churned Customers", fontsize = 10)
plt.show()
gb
```



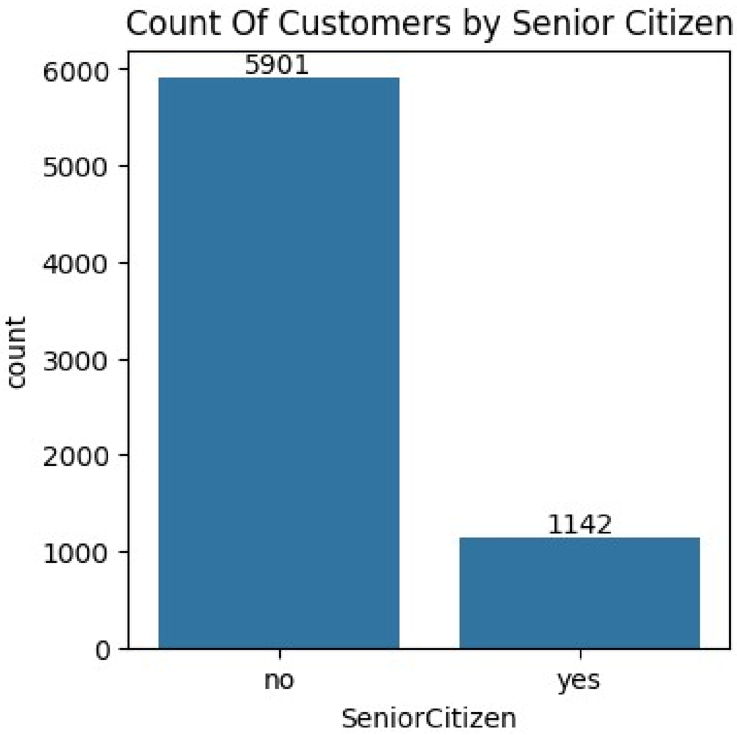
```
       Churn
Churn
No      5174
Yes     1869
```

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

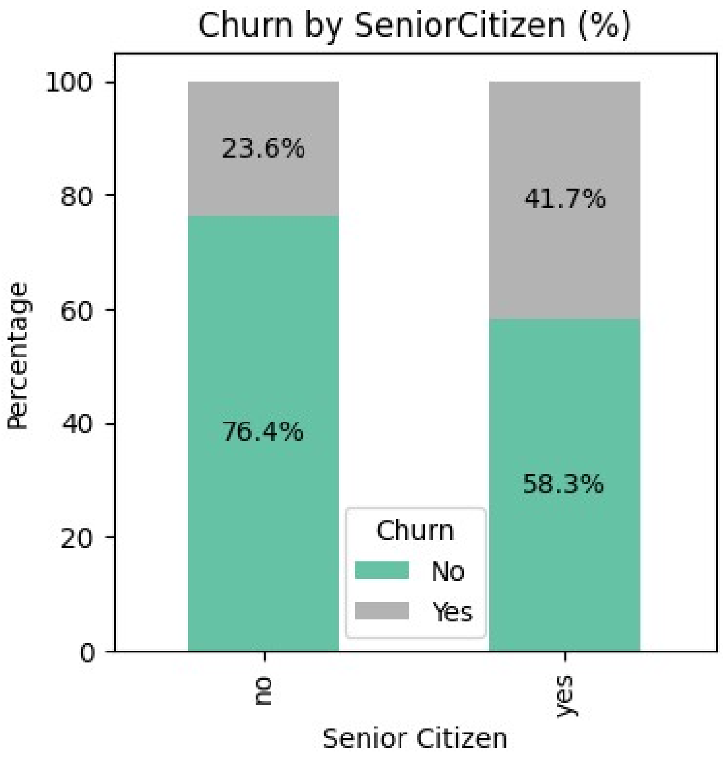
```
plt.figure(figsize = (3,3))
sns.countplot(x= "gender", data= df, hue = "Churn")
plt.title("Churn by Gender")
plt.show()
```



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

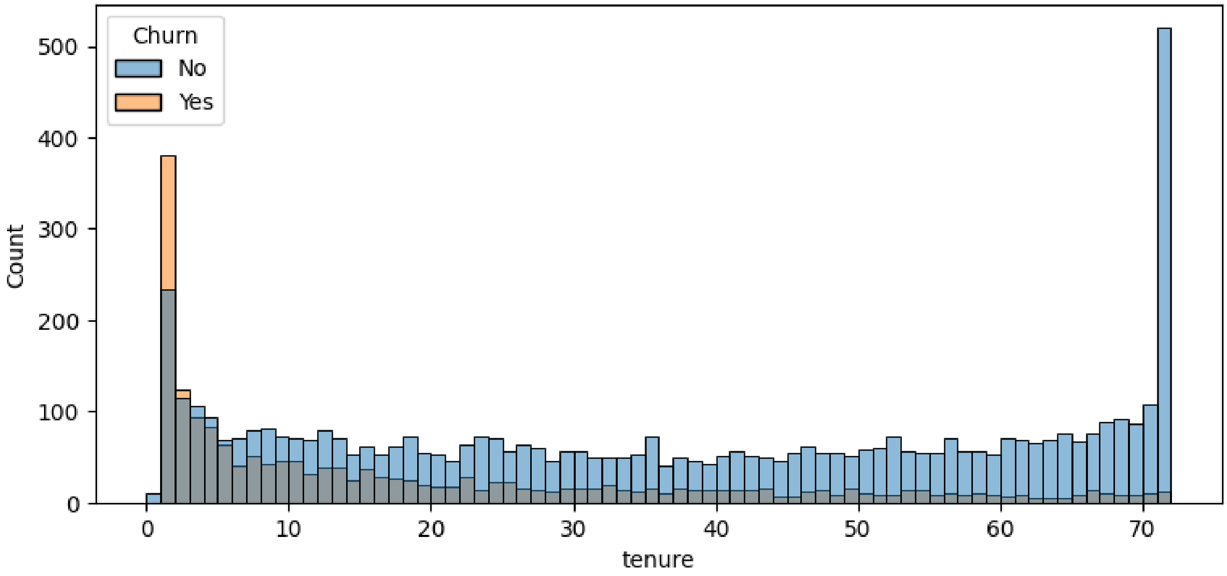


```
ct = pd.crosstab(df["SeniorCitizen"], df["Churn"], normalize="index")
* 100

# Plot stacked bar chart
ax = ct.plot(kind="bar", stacked=True, figsize=(4,4), colormap="Set2")

# Add labels as % to total
for p in ax.patches:
    width = p.get_width()
    height = p.get_height()
    x, y = p.get_xy()
    if height > 0:  # Avoid 0% labels
        ax.text(x + width/2,
                y + height/2,
                f"{height:.1f}%",
                ha="center", va="center", fontsize=10, color="black")

plt.title("Churn by SeniorCitizen (%)")
plt.xlabel("Senior Citizen")
plt.ylabel("Percentage")
plt.legend(title="Churn")
plt.show()
```



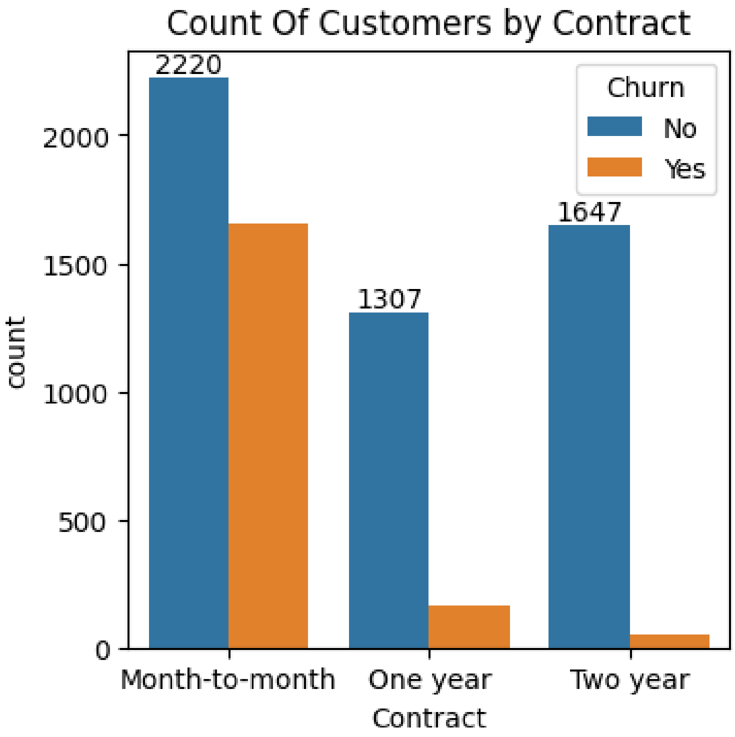
Comparative a greater percentage of people in senior citizen category have churned.

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



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()
```



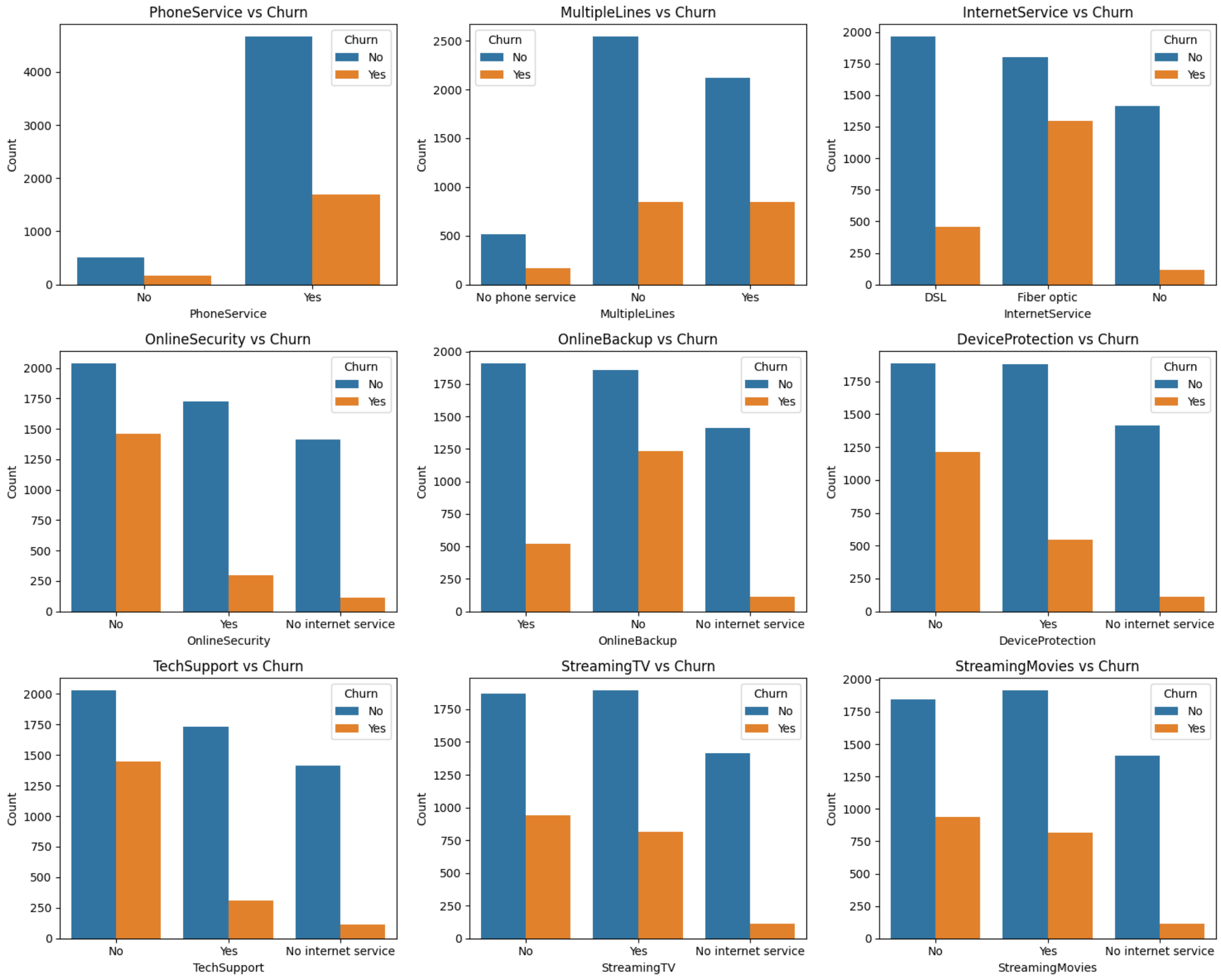
People who have month to month contract are likely to churn then from those who have 1 or 2 years of contract.

```
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)
```

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

# Create subplots
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12))
```

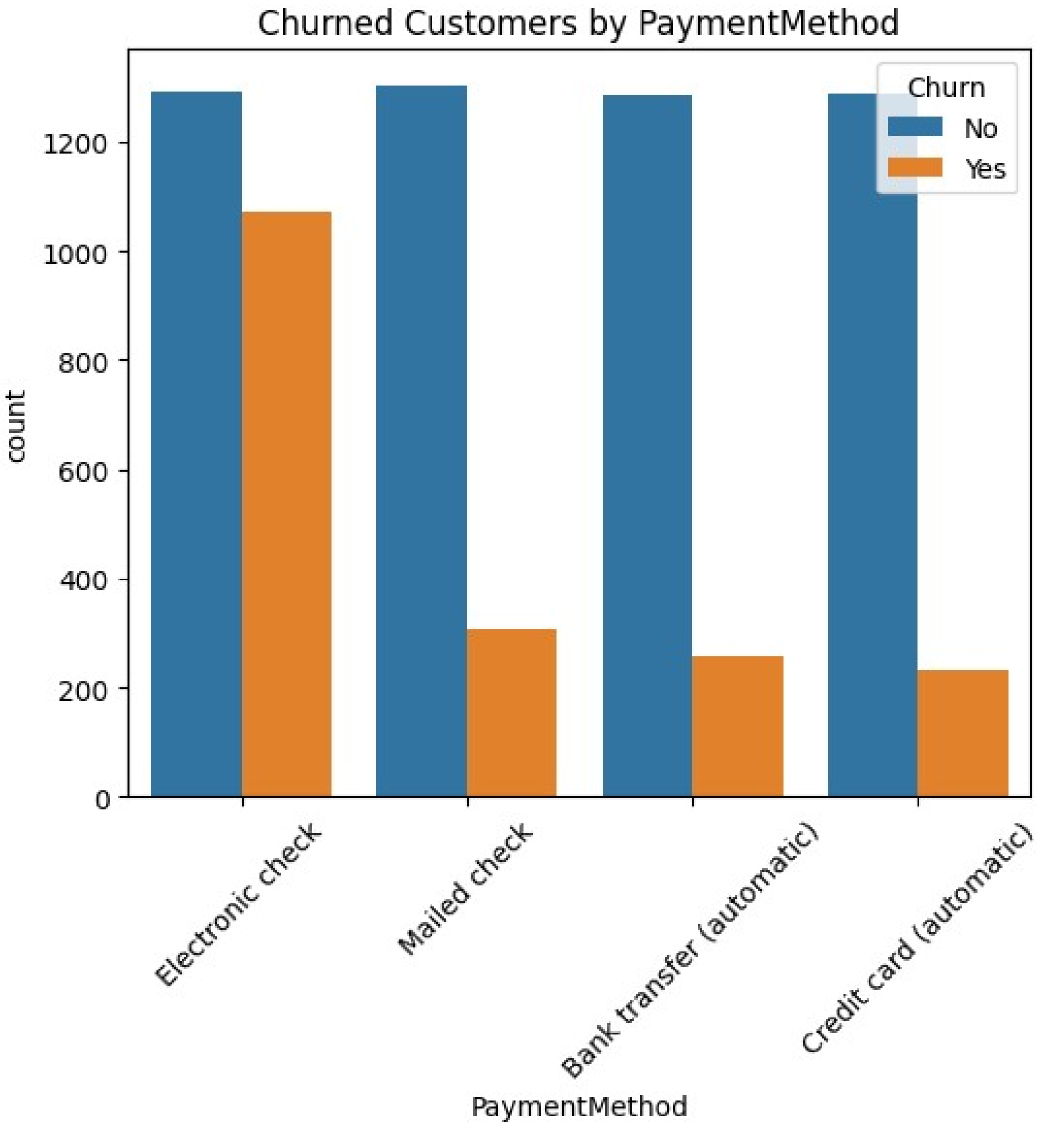
```
axes = axes.flatten()

for i, col in enumerate(features):
    sns.countplot(x=col, data=df, hue="Churn", ax=axes[i])
    axes[i].set_title(f"{col} vs Churn")
    axes[i].set_xlabel(col)
    axes[i].set_ylabel("Count")

# Remove empty subplot if number of features < grid size
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

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



The majority of customers who do not churn tend to have services like phone service, internet service (particularly DSL), and online Security enabled.

For services like online backup, Tech support, and Streaming Tv,churn rates are noticeably higher when these services are not used or are unavailable.

```python
plt.figure(figsize=(6,5))
sns.countplot(x="PaymentMethod", data=df, hue="Churn")

plt.title("Churned Customers by PaymentMethod")

# Rotate x-axis labels
plt.xticks(rotation=45)   # Rotate by 45 degrees
# plt.xticks(rotation=90) # If you want vertical labels

plt.show()
```



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