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

Importing Data as DataFrame

```
df = pd.read_csv("Customer Churn.csv")
print(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-CFOCW    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  Month-
to-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]
```

Replacing blank values from Total Charges column to 0 as tenure is 0 and no total charges are recorded

```
df["TotalCharges"] = df["TotalCharges"].replace(" ","0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

Checking null values in the dataframe

```
check_null = df.isnull().sum().sum()
print(check_null)
```

```
0
```

```
describe = df.describe()
print(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
```

Checking Duplicates in the Data

```
duplicates = df["customerID"].duplicated().sum()
print(duplicates)
```

```
0
```

Function for creating yes and no values for SeniorCitizen column

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

df["SeniorCitizen"] = df["SeniorCitizen"].apply(convert)
print(df.head())
```

```
   customerID  gender SeniorCitizen Partner Dependents  tenure
PhoneService  \
0  7590-VHVEG  Female            no     Yes         No       1
No
1  5575-GNVDE    Male            no      No         No      34
Yes
2  3668-QPYBK    Male            no      No         No       2
Yes
3  7795-CFOCW    Male            no      No         No      45
No
4  9237-HQITU  Female            no      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
```

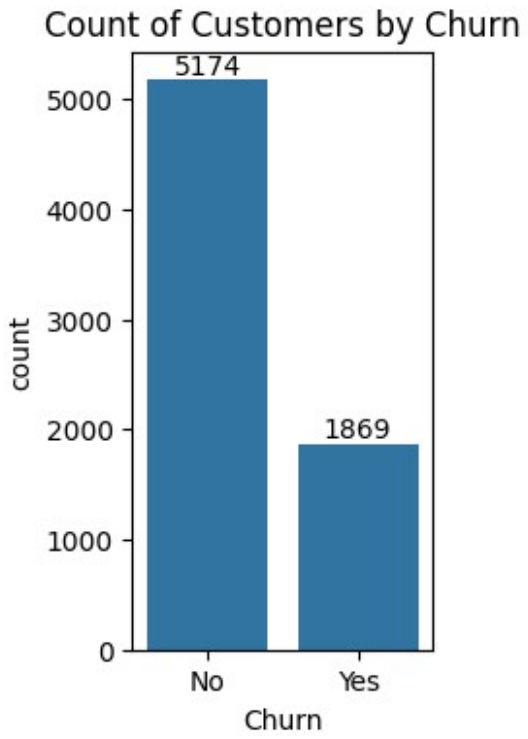
```
Yes
```

	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	Electronic check	29.85	29.85	No
1	Mailed check	56.95	1889.50	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 rows x 21 columns]

Creating count Plot to check the count of people churned out

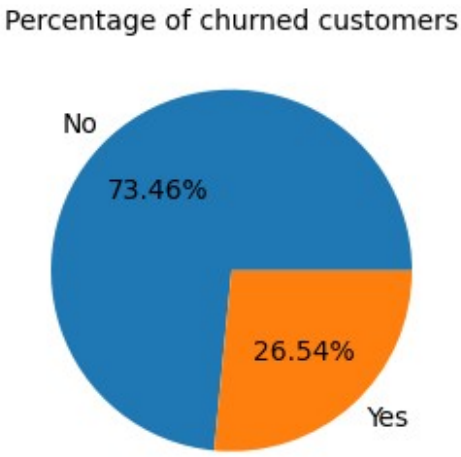
```
plt.figure(figsize = (2,4))
ax = sns.countplot(x = 'Churn', data = df)

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



Plotting a Pie Chart

```
plt.figure(figsize = (3,4))
gb = df.groupby("Churn").agg({'Churn':"count"})
```

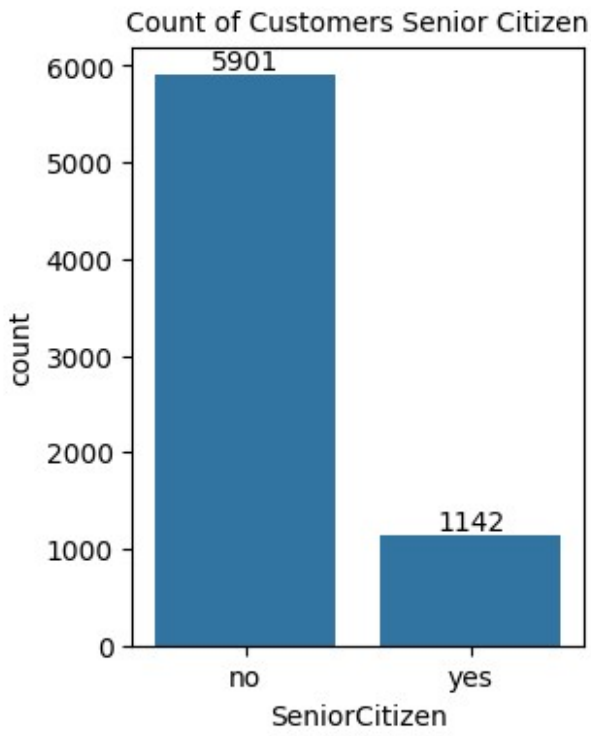
```
plt.pie(gb["Churn"], labels = gb.index, autopct = "%1.2f%%")
plt.title("Percentage of Churned Customers", fontsize = 10)
plt.show()
```



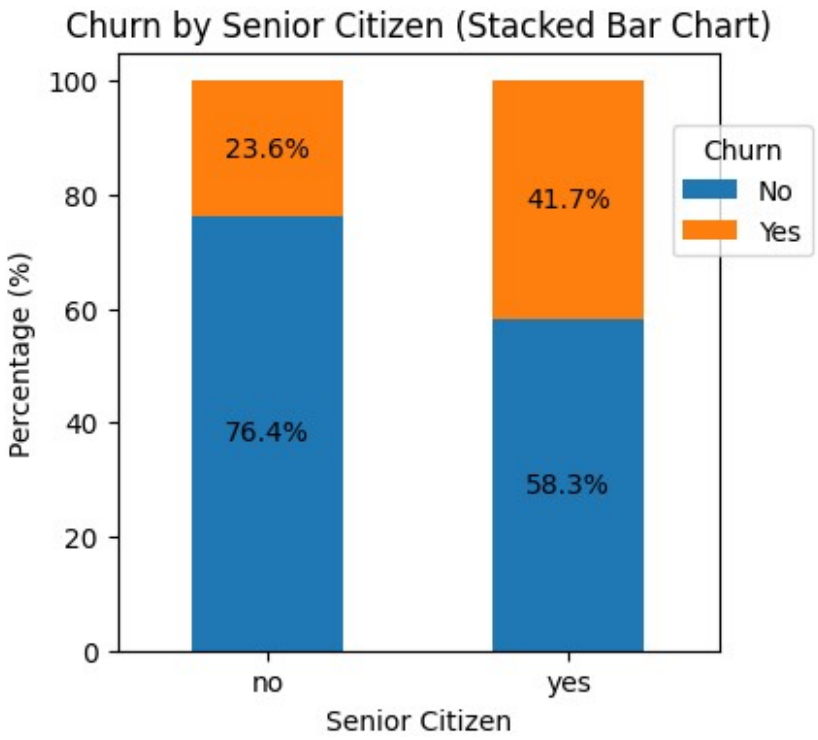
From the given pie chart we conclued that 26.54% of our customers have churned out

Now lets explore the reason behind it

```
plt.figure(figsize = (3,4))
ax = sns.countplot(x = "SeniorCitizen", data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers Senior Citizen", fontsize = 10)
plt.show()
```

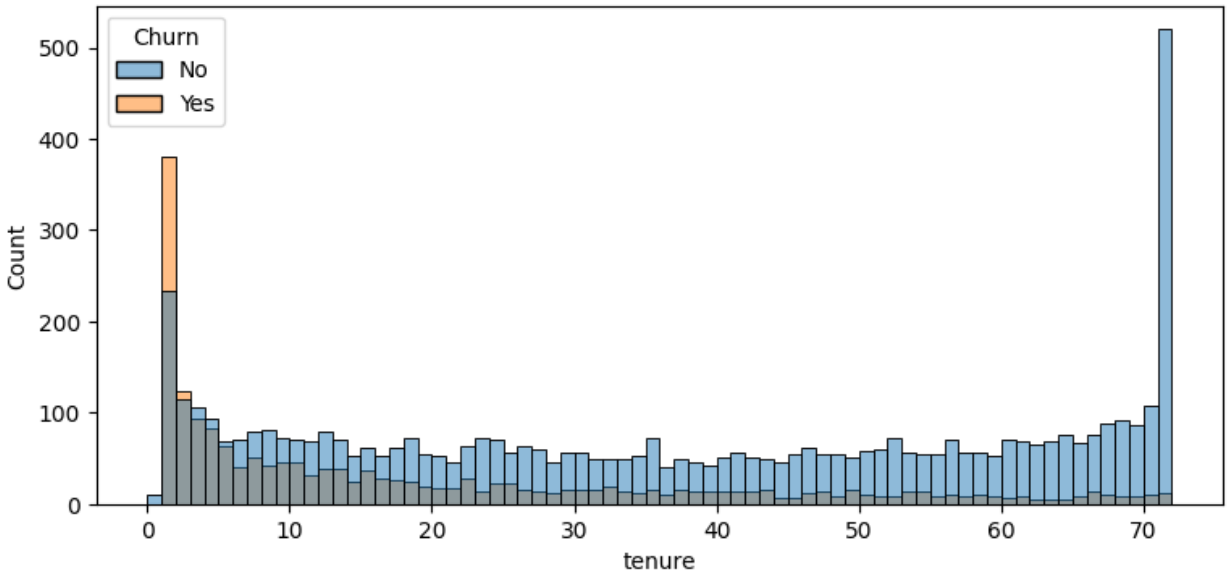



```
total_counts = df.groupby('SeniorCitizen')
['Churn'].value_counts(normalize = True).unstack() * 100

#plot
fig, ax = plt.subplots(figsize = (4,4))  # Adjudting figsize for
better visualization

# plotting the bars
total_counts.plot(kind = 'bar', stacked = True, ax = ax, color =
['#1f77b4', '#ff7f0e'])  # color customization

# Adding percentage labels on the bars
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y= p.get_xy()
    ax.text(x + width / 2, y+ height / 2, f'{height:.1f}%',
ha='center', va='center')

plt.title("Churn by Senior Citizen (Stacked Bar Chart)")
plt.xlabel('Senior Citizen')
plt.ylabel("Percentage (%)")
plt.xticks(rotation=0)
plt.legend(title="Churn", bbox_to_anchor = (0.9,0.9)) # Customizing
Legend Location
plt.show()
```



Comparatively a greater percentage of people in senior citizen category have Churned out

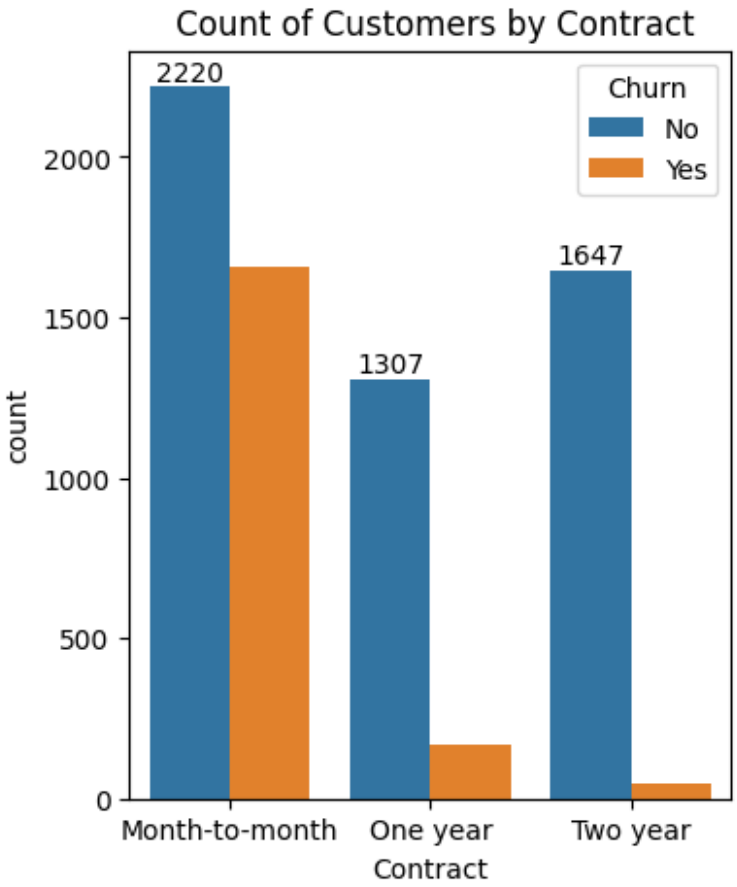
```
plt.figure(figsize=(9,4))
sns.histplot(x = "tenure", data=df, bins= 72, hue="Churn")
plt.show()
```



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

```
plt.figure(figsize=(4,5))
ax = sns.countplot(x="Contract", data = df, hue="Churn")

ax.bar_label(ax.containers[0])
plt.title(" Count of Customers by Contract")
plt.show()
```



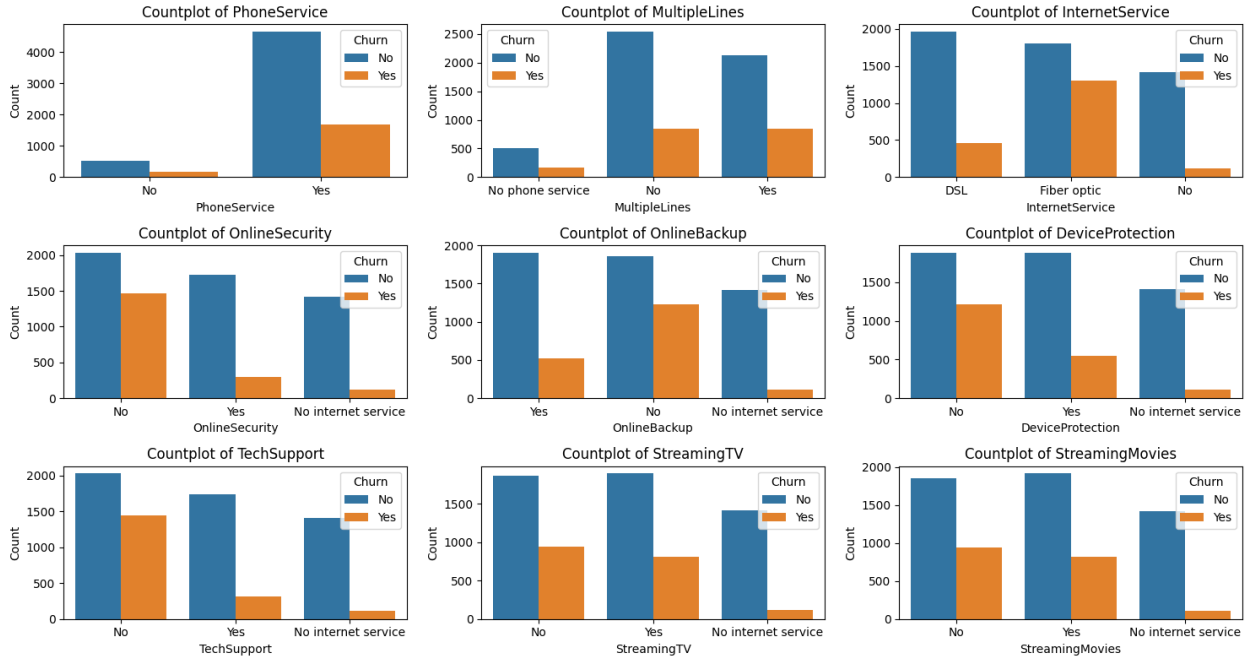
People who have Month-to-month contract are likely to churn more then from those who

have 1 or 2 years of contract

```
print(df.columns.values)

columns = ['PhoneService', 'MultipleLines', 'InternetService',
'OnlineSecurity',
```

```
           'OnlineBackup', 'DeviceProtection', 'TechSupport',
'StreamingTV', 'StreamingMovies']

# Number of rows and columns for the subplots (3 rows x 3 columns in
this case for 9 plots)
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 8))

# Flatten the axes array for easier iteration
axes = axes.flatten()

# # Looping through each column and creating a count plot for each
for i, col in enumerate(columns):
    sns.countplot(x=col, data=df, ax=axes[i], hue=df["Churn"])  #
Create countplot
    axes[i].set_title(f"Countplot of {col}")  # Set title for each
plot
    axes[i].set_xlabel(col)  # Set x-axis label
    axes[i].set_ylabel('Count')  # Set y-axis label

# # Adjusting the layout to prevent overlapping
plt.tight_layout()

# Display the plots
plt.show()
```

```
['customerID' 'gender' 'SeniorCitizen' 'Partner' 'Dependents' 'tenure'
 'PhoneService' 'MultipleLines' 'InternetService' 'OnlineSecurity'
 'OnlineBackup' 'DeviceProtection' 'TechSupport' 'StreamingTV'
 'StreamingMovies' 'Contract' 'PaperlessBilling' 'PaymentMethod'
 'MonthlyCharges' 'TotalCharges' 'Churn']
```

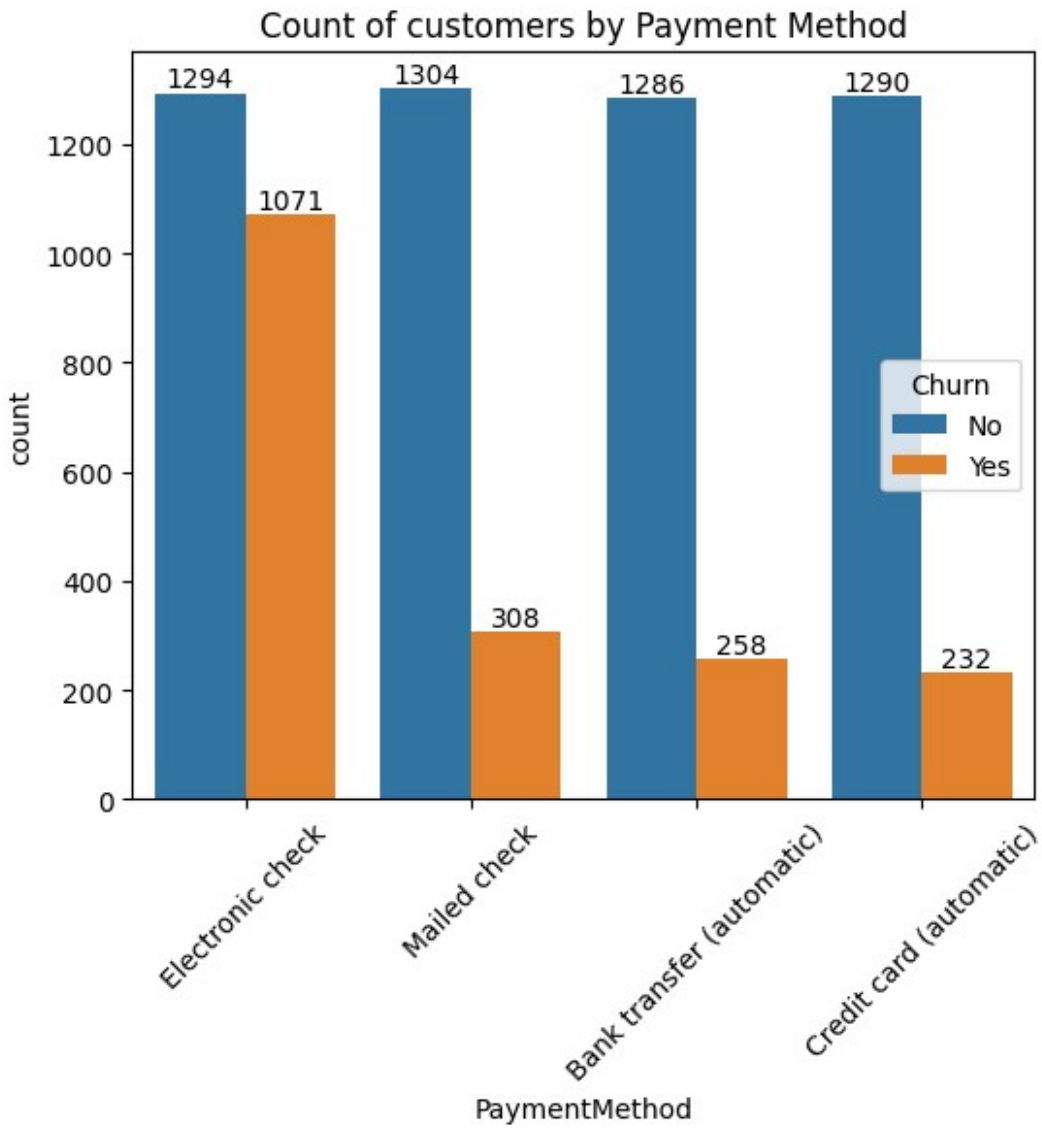


The "Fiber optic" internet service category shows a noticeable churn rate compared to "DSL."

Additionally, services like OnlineBackup, DeviceProtection, StreamingTV, and

StreamingMovies indicate that customers lacking these features are more prone to churn.

```
plt.figure(figsize=(6,5))
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("Count of customers by Payment Method")
plt.show()
```



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