```
import numpy as np
import pandas as pd
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-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]
```

```
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 blank space with 0 as tensure is Zero no total charges is 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()
```

```
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()
```

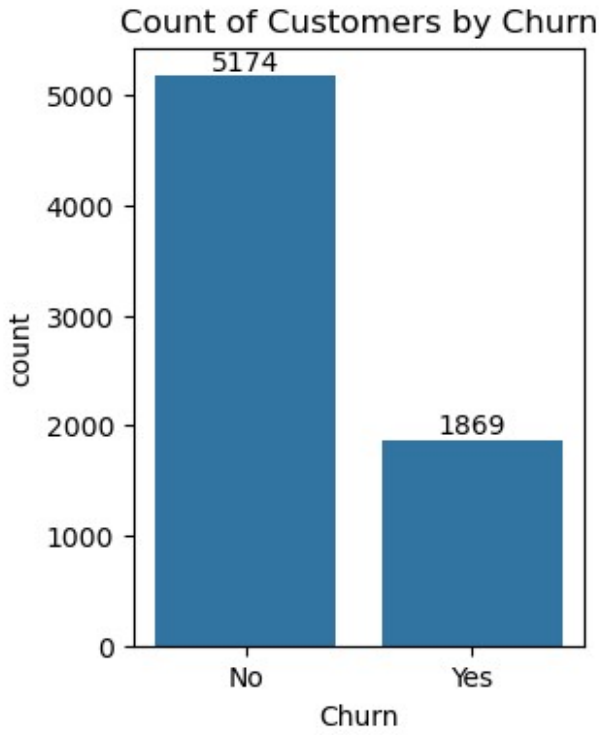
```
0
```

```
def conv(value):
    if value == 1:
```

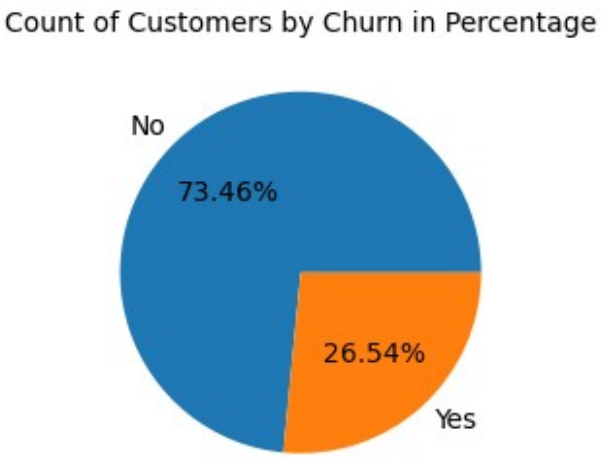
```
        return "yes"
    else:
        return "no"

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

converted 0 and 1 vlaue of senior citizne into yes and no for easy

```
plt.figure(figsize = (3,4))

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

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

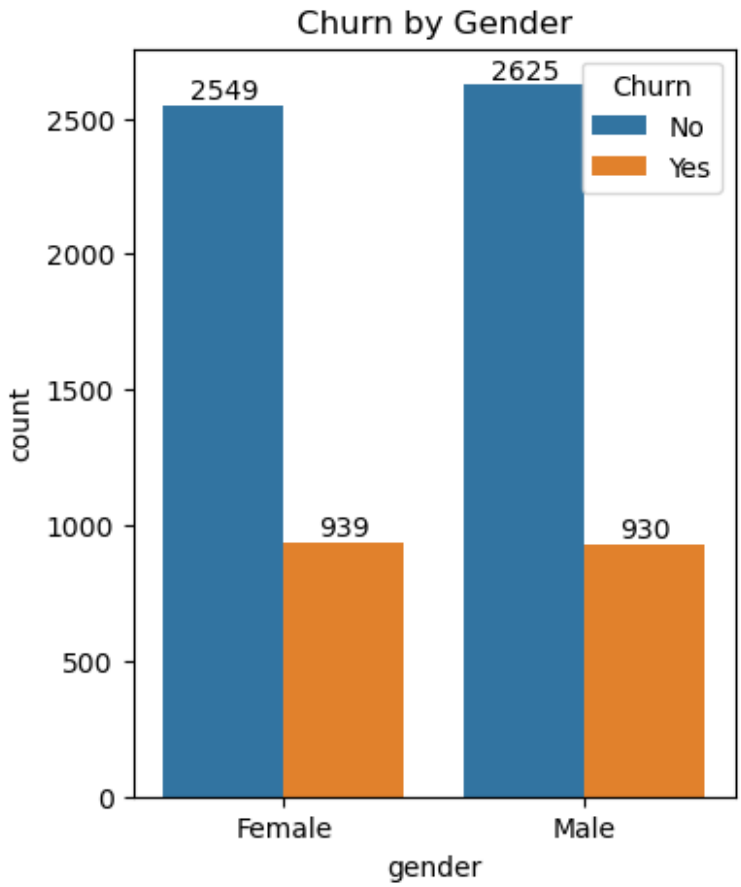



```
plt.figure(figsize = (3,4))
gb = df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")
plt.title("Count of Customers by Churn in Percentage", fontsize = 10)
plt.show()
```

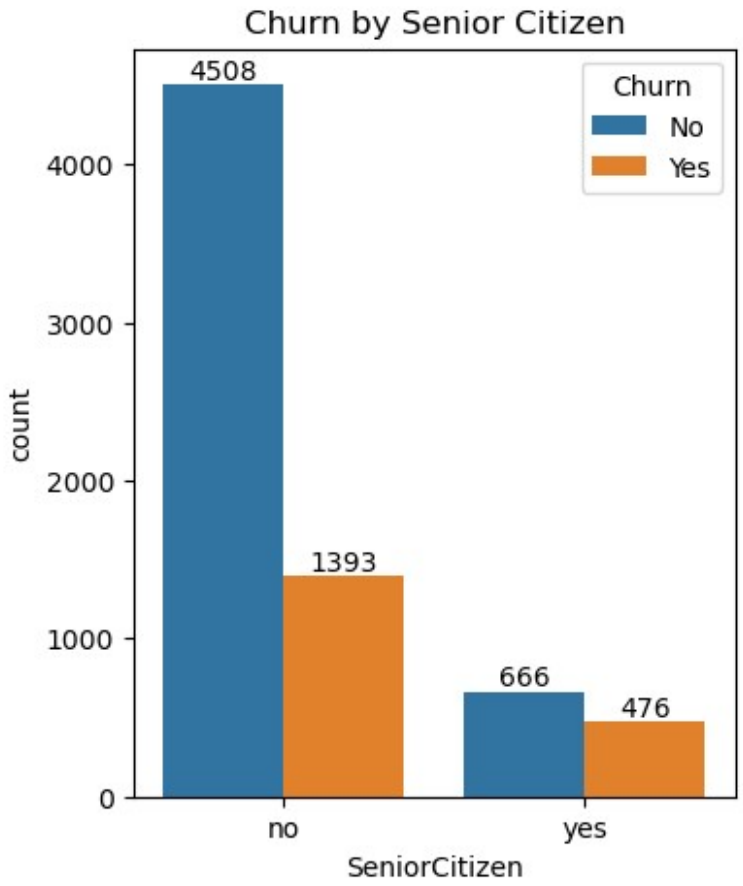


The above graph shows that 26.54% of the customer has churn out. Lets figure out the reason

```
plt.figure(figsize = (4,5))
# Capture the returned axes object from sns.countplot
ax = sns.countplot(x= "gender", data = df, hue = "Churn")
# Now we can access ax.containers
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Churn by Gender")
plt.show()
```

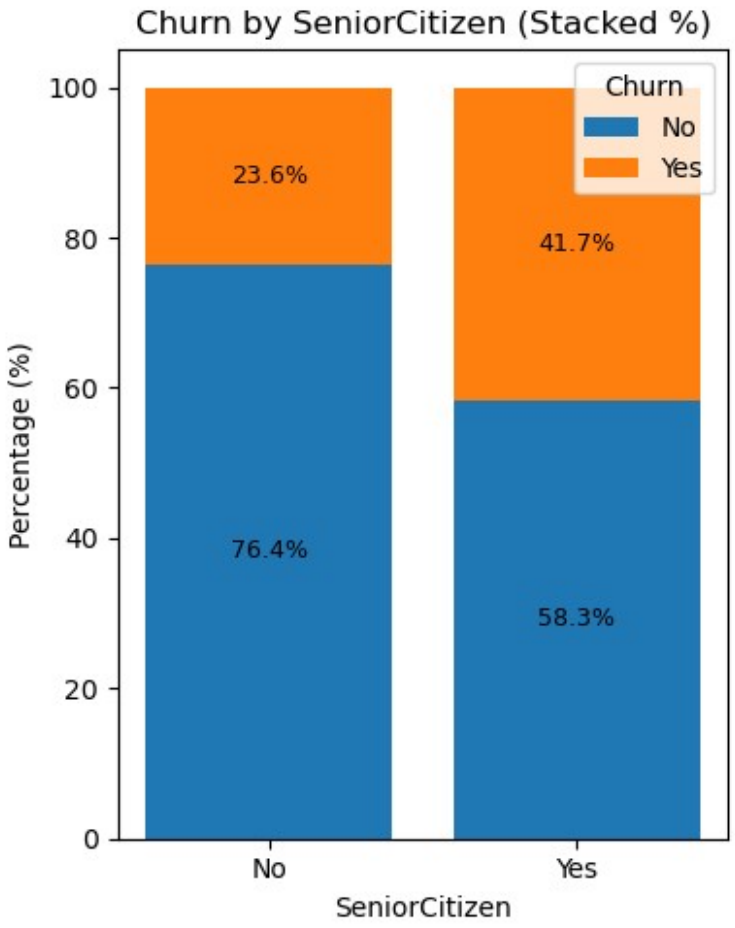



```
plt.figure(figsize = (4,5))
# Capture the returned axes object from sns.countplot
ax = sns.countplot(x= "SeniorCitizen", data = df, hue = "Churn")
# Now we can access ax.containers
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Churn by Senior Citizen")
plt.show()
```



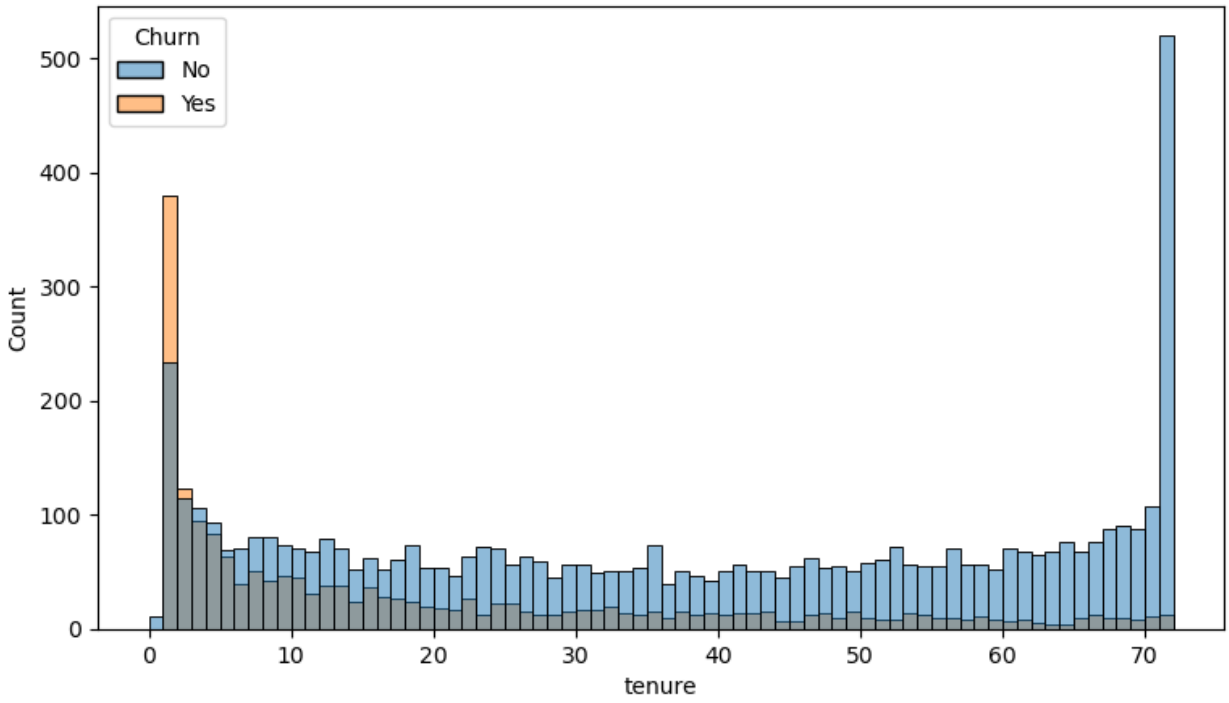

```
grouped = df.groupby(['SeniorCitizen',
'Churn']).size().unstack(fill_value=0)

# Calculate percentages
percentages = grouped.div(grouped.sum(axis=1), axis=0) * 100

# Plot
fig, ax = plt.subplots(figsize=(4, 5))

# Define colors - this was missing in the original code
colors = ['#1f77b4', '#ff7f0e']  # Default matplotlib colors, you can
customize these

# Bottoms for stacking
bottom_vals = [0] * len(percentages)

# Plot bars
for i, col in enumerate(percentages.columns):
    ax.bar(percentages.index, percentages[col], bottom=bottom_vals,
label=col, color=colors[i])

    # Add percentage labels
```

```
    for j, val in enumerate(percentages[col]):
        ax.text(j, bottom_vals[j] + val / 2, f'{val:.1f}%',
ha='center', va='center', color='black', fontsize=9)

    bottom_vals = [bottom_vals[k] + percentages[col].iloc[k] for k in
range(len(bottom_vals))]

# Labels and legend
ax.set_title("Churn by SeniorCitizen (Stacked %)")
ax.set_xlabel("SeniorCitizen")
ax.set_ylabel("Percentage (%)")
ax.set_xticks([0, 1])
ax.set_xticklabels(['No', 'Yes'])  # if 0 = No, 1 = Yes
ax.legend(title='Churn')
plt.tight_layout()
plt.show()
```



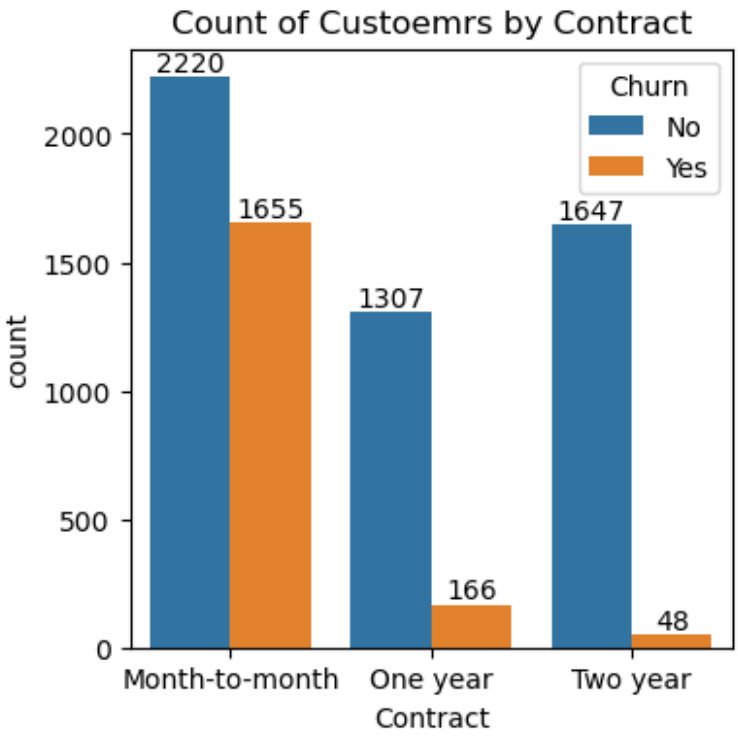
In the chart we can see more senior citizen category have churn out

```
plt.figure(figsize=(9, 5))
sns.histplot(x = "tenure", data = df, bins = 72, hue = "Churn")
plt.show()
```



people who have used are services for a long time have stayed and people who have used are servies for 1 or 2 month have churned out

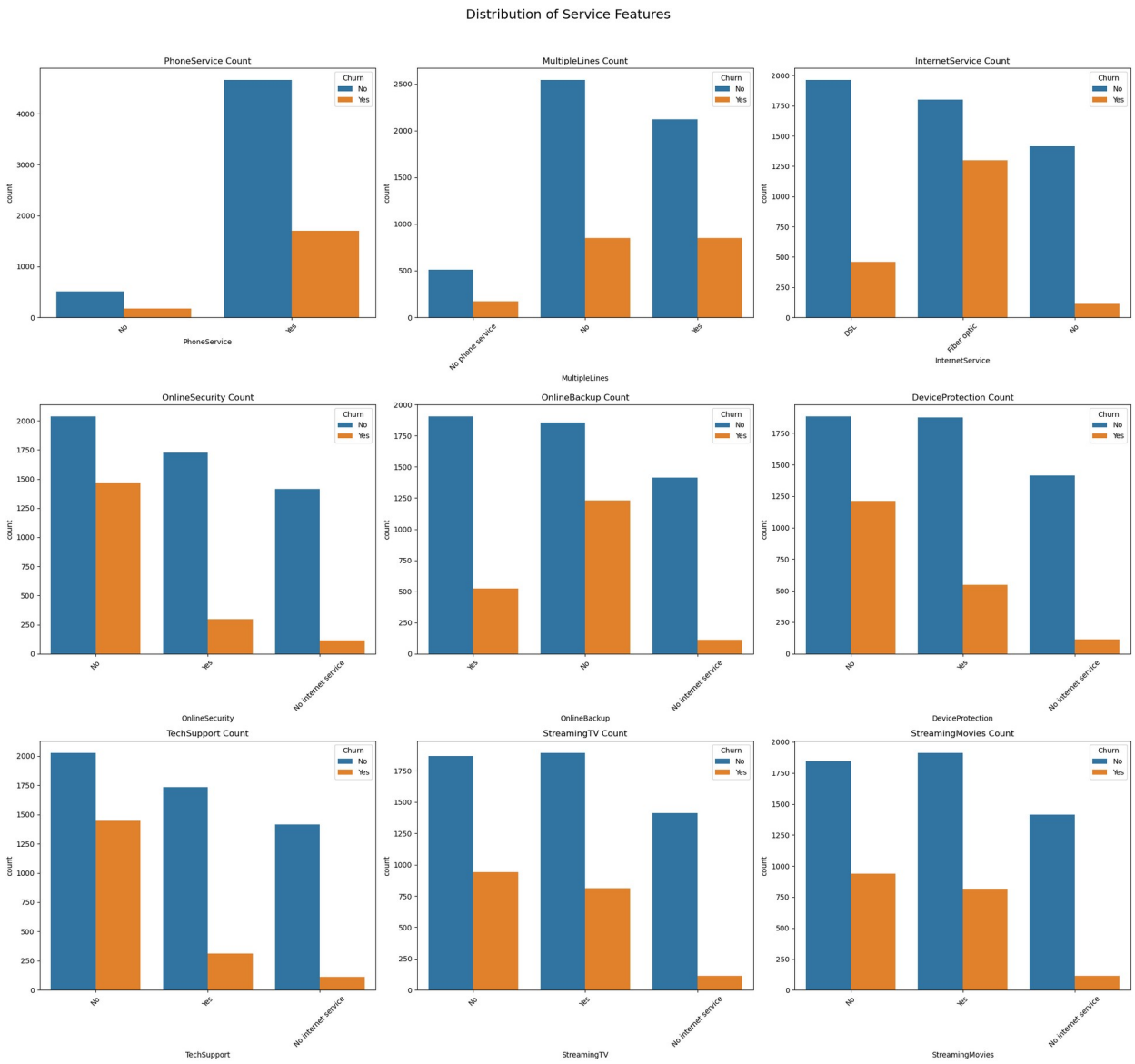
```
plt.figure(figsize = (4,4))
ax = sns.countplot (x= "Contract", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title("Count of Custoemrs by Contract")
plt.show()
```



People who have monthly contract churn our more comparative to the people who have yearly contract

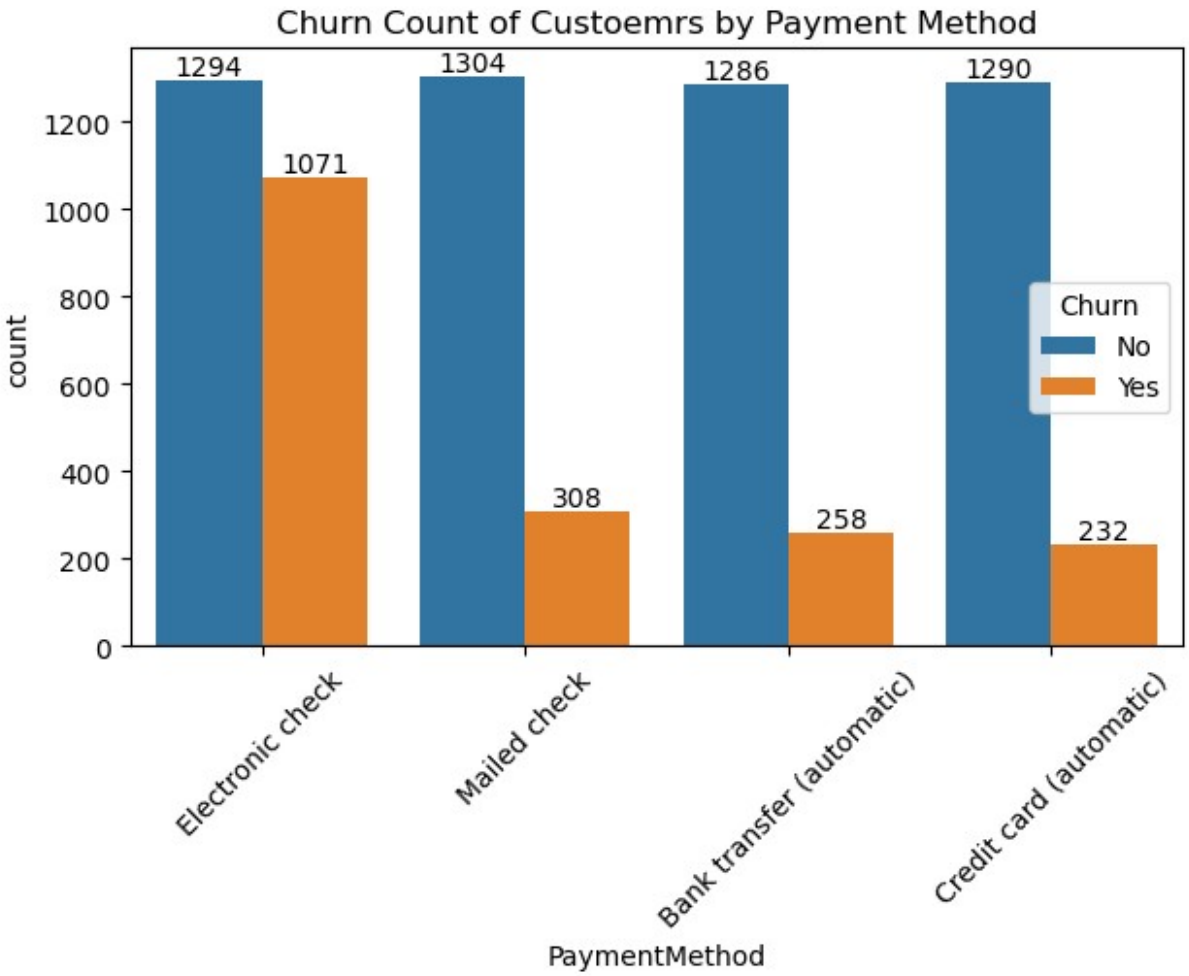
```
df.columns.values

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 to visualize
cols = ['PhoneService', 'MultipleLines', 'InternetService',
        'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
        'TechSupport', 'StreamingTV', 'StreamingMovies']

# Grid setup
n_cols = 3
n_rows = (len(cols) + n_cols - 1) // n_cols
```

```
# Figure size
plt.figure(figsize=(22, 20))

# Create subplots
for i, col in enumerate(cols, 1):
    plt.subplot(n_rows, n_cols, i)
    sns.countplot(x=col, data=df, hue = "Churn")
    plt.title(f'{col} Count')
    plt.xticks(rotation=45)
    plt.tight_layout()

# Final adjustments
plt.suptitle("Distribution of Service Features", fontsize=18, y=1.02)
plt.tight_layout()
plt.show()
```



#The data shows that customers who do not subscribe to value-added services like OnlineSecurity, TechSupport, DeviceProtection, and OnlineBackup are more likely to churn. Fiber optic internet users also show higher churn compared to DSL users. Streaming services do not significantly reduce churn, suggesting they are less critical for retention. Overall, lack of service engagement is strongly linked to higher churn, highlighting the importance of cross-selling core services.

```
plt.figure(figsize = (7,4))
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("Churn Count of Custoemrs by Payment Method")
plt.show()
```



Customers likely to churn when he is using electronic check as a payment menthod