

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```



```
[2]: data=pd.read_csv("C:/Users/anjali/OneDrive/Desktop/Data Analyst/pandas/EDA/customer Analysis/Customer Churn.csv")
data
```

```
[2]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No
...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	No
7040	4801-JJAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	No

Information about dataset

```
[3]: data.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

```
[4]: data["TotalCharges"] = data["TotalCharges"].replace(" ", "0")
data["TotalCharges"] = data["TotalCharges"].astype("float")
```

```
[5]: data.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)
```

Check null values in dataset

```
[6]: data.isnull().sum()
```

```
[6]: customerID      0
      gender         0
      SeniorCitizen  0
      Partner        0
      Dependents     0
      tenure         0
      PhoneService   0
      MultipleLines   0
      InternetService 0
      OnlineSecurity  0
      OnlineBackup    0
      DeviceProtection 0
      TechSupport     0
      StreamingTV     0
      StreamingMovies 0
      Contract        0
      PaperlessBilling 0
      PaymentMethod   0
      MonthlyCharges  0
      TotalCharges    0
      Churn           0
      dtype: int64
```

Display all statistics about dataset

```
[7]: data.describe()
```

```
[7]:
```

	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

Check duplicates in dataset

```
[8]: dup=data.duplicated().any()
print("Are there any duplicated value?",dup)
```

Are there any duplicated value? False

```
[9]: data["customerID"].duplicated().sum()
```

```
[9]: np.int64(0)
```

Convert 0 to no and 1 to yes in SeniorCitizen column

```
[10]: def con(value):
      if value == 1:
          return "yes"
      else:
          return "no"

data["SeniorCitizen"]=data["SeniorCitizen"].apply(con)
```

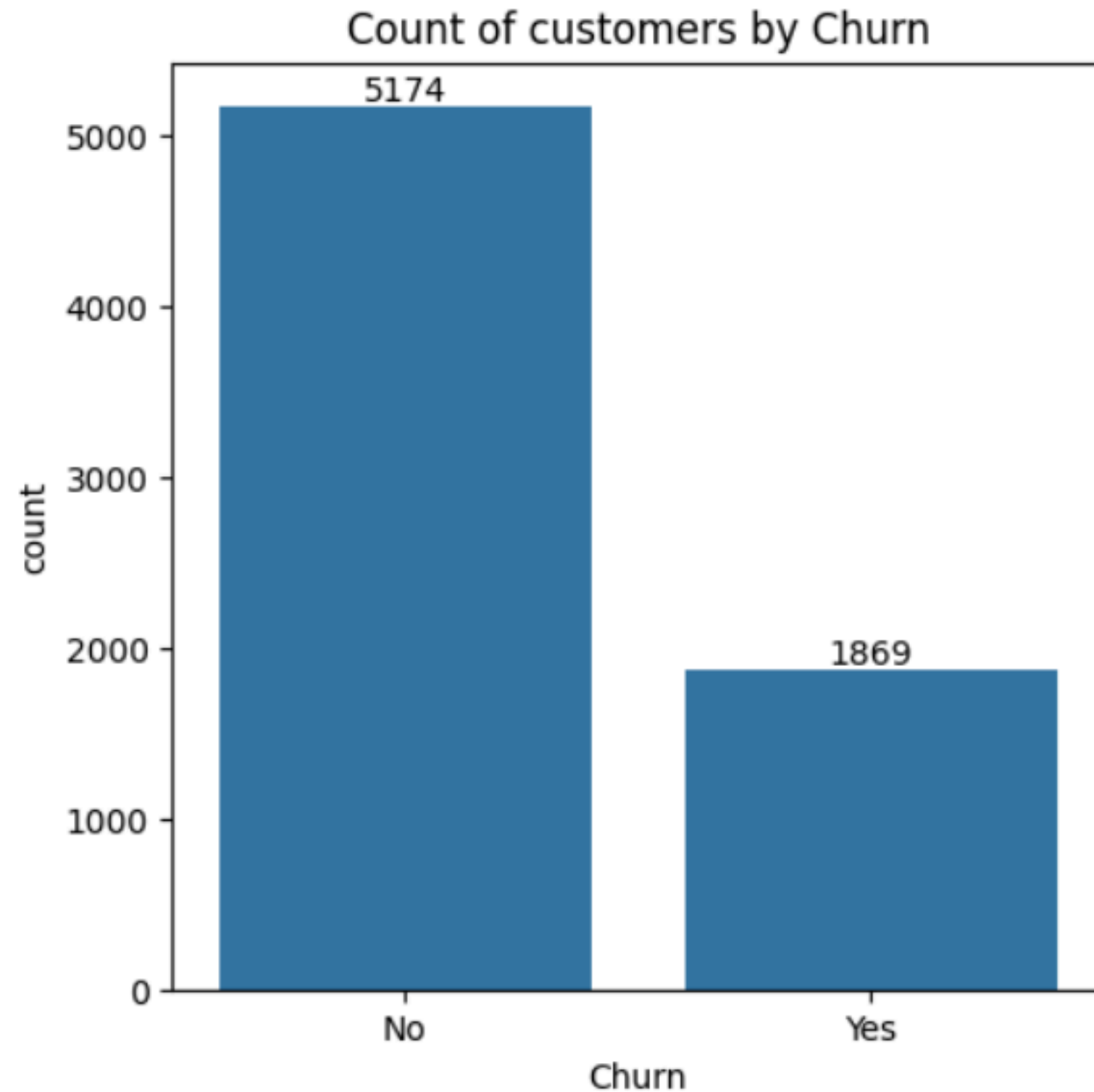
```
[11]: data.head()
```

```
[11]:
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	St
0	7590-VHVEG	Female	no	Yes	No	1	No	No phone service	DSL	No	...	No	No	
1	5575-GNVDE	Male	no	No	No	34	Yes	No	DSL	Yes	...	Yes	No	

Customer churned plots

```
[25]: plt.figure(figsize=(5,5))
      ax=sns.countplot(x="Churn",data=data)
      ax.bar_label(ax.containers[0])
      plt.title("Count of Customers by Churn")
      plt.show()
```



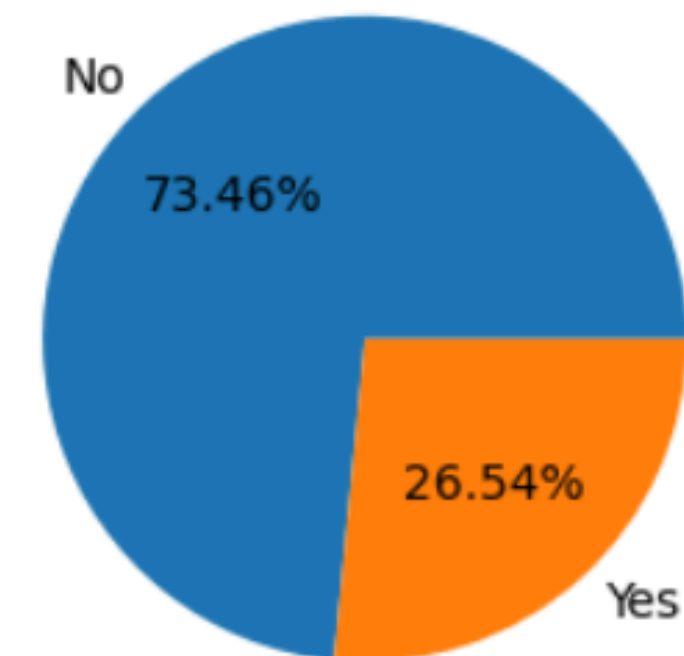
```
[17]: gb=data.groupby("Churn").agg({"Churn":"count"})
      gb
```

[17]:

Churn	
No	5174
Yes	1869

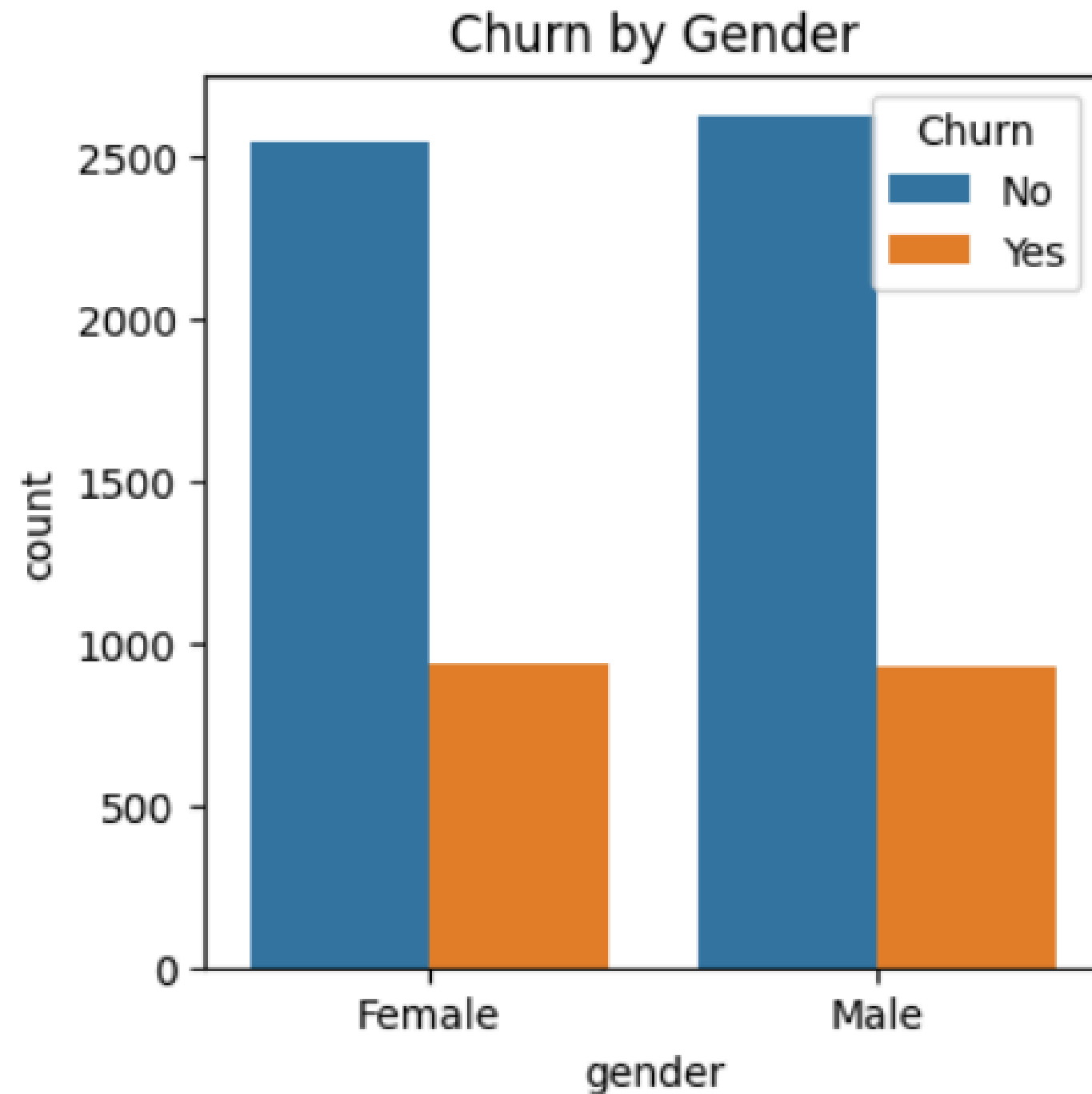
```
[26]: plt.figure(figsize=(3,4))
      plt.pie(gb["Churn"],labels=gb.index,autopct="%1.2f%%")
      plt.title("Percentage of Churned Customers",fontsize=10)
      plt.show()
```

Percentage of Churned Customers



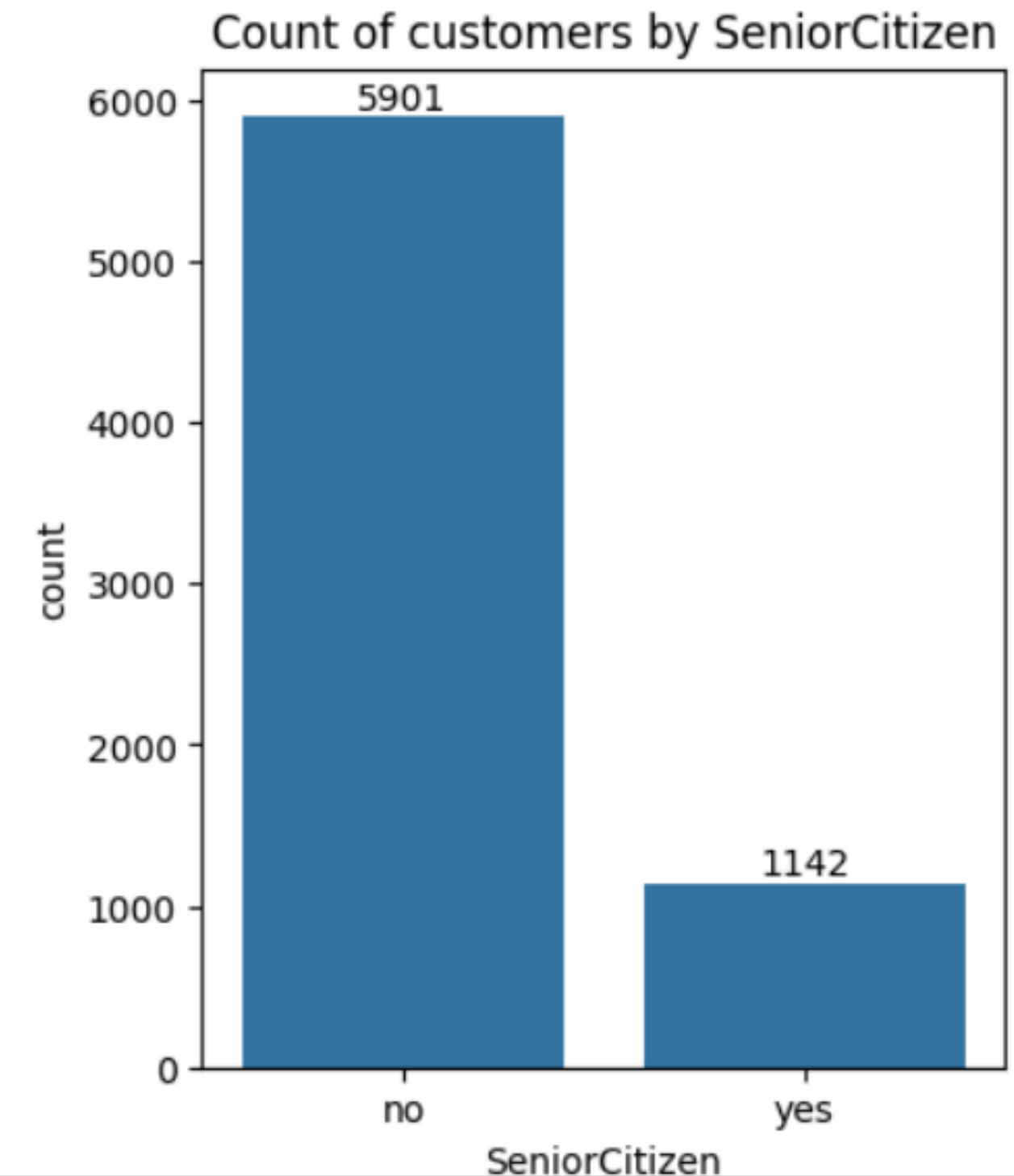
▼ plots based on Gender

```
[56]: plt.figure(figsize=(4,4))
sns.countplot(x="gender",data=data,hue="Churn")
plt.title("Churn by Gender")
plt.show()
```



plot based on seniorcitizen

```
[52]: plt.figure(figsize=(4,5))
ax=sns.countplot(x="SeniorCitizen",data=data)
ax.bar_label(ax.containers[0])
plt.title("Count of customers by SeniorCitizen")
plt.show()
```

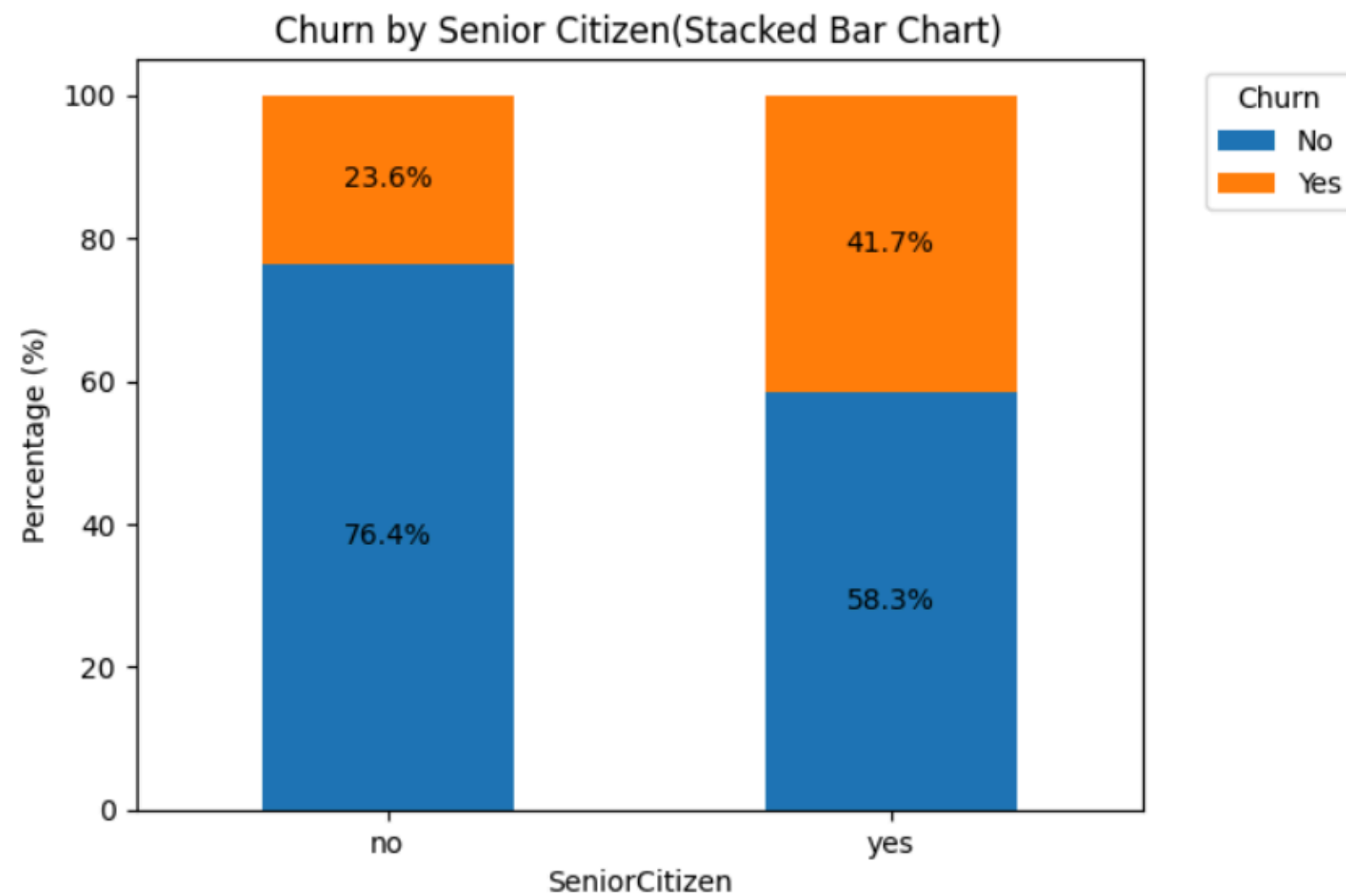


using "stacked bar chart"

```
[54]: total_counts=data.groupby("SeniorCitizen")["Churn"].value_counts(normalize=True).unstack()*100

#plot
ax=total_counts.plot(kind="bar",stacked=True)
#add labels
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")

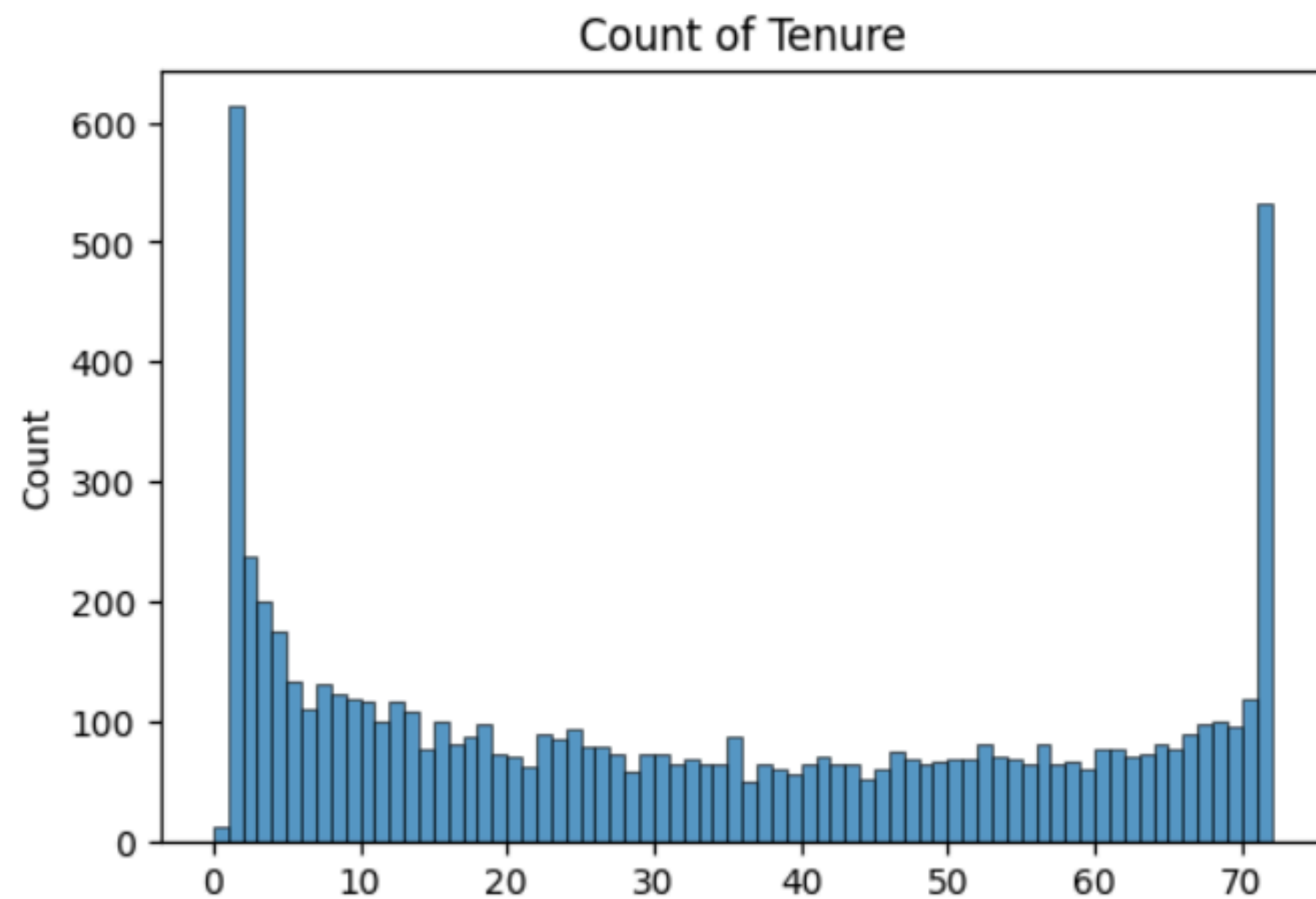
plt.title("Churn by Senior Citizen(Stacked Bar Chart)")
plt.xlabel("SeniorCitizen")
plt.ylabel("Percentage (%)")
plt.legend(title="Churn",bbox_to_anchor=(1.05,1),loc=2)
plt.xticks(rotation=0)
plt.show()
```



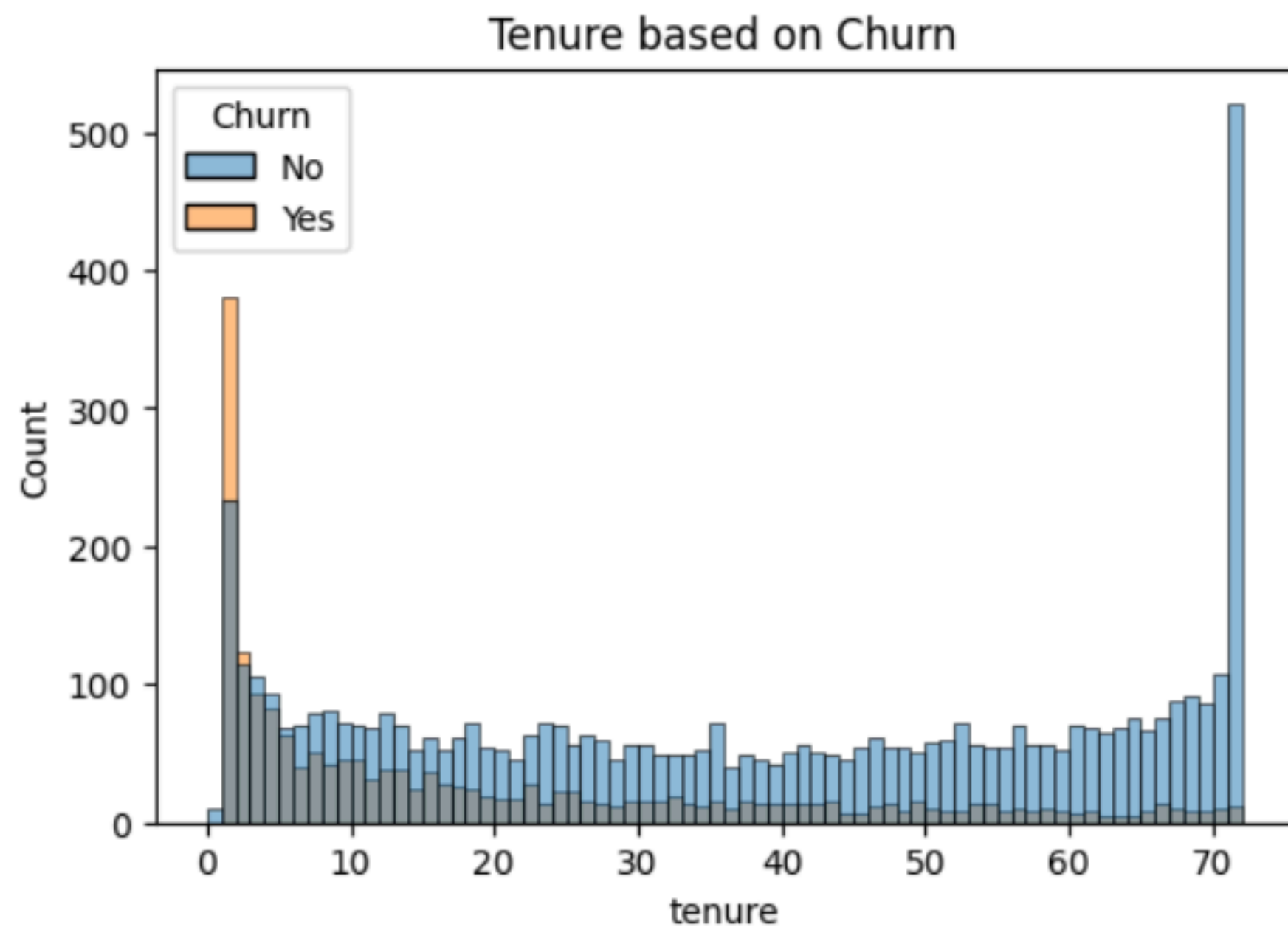
▼ Count of Tenure



```
[62]: plt.figure(figsize=(6,4))
sns.histplot(x="tenure",data=data,bins=72)
plt.title("Count of Tenure")
plt.show()
```



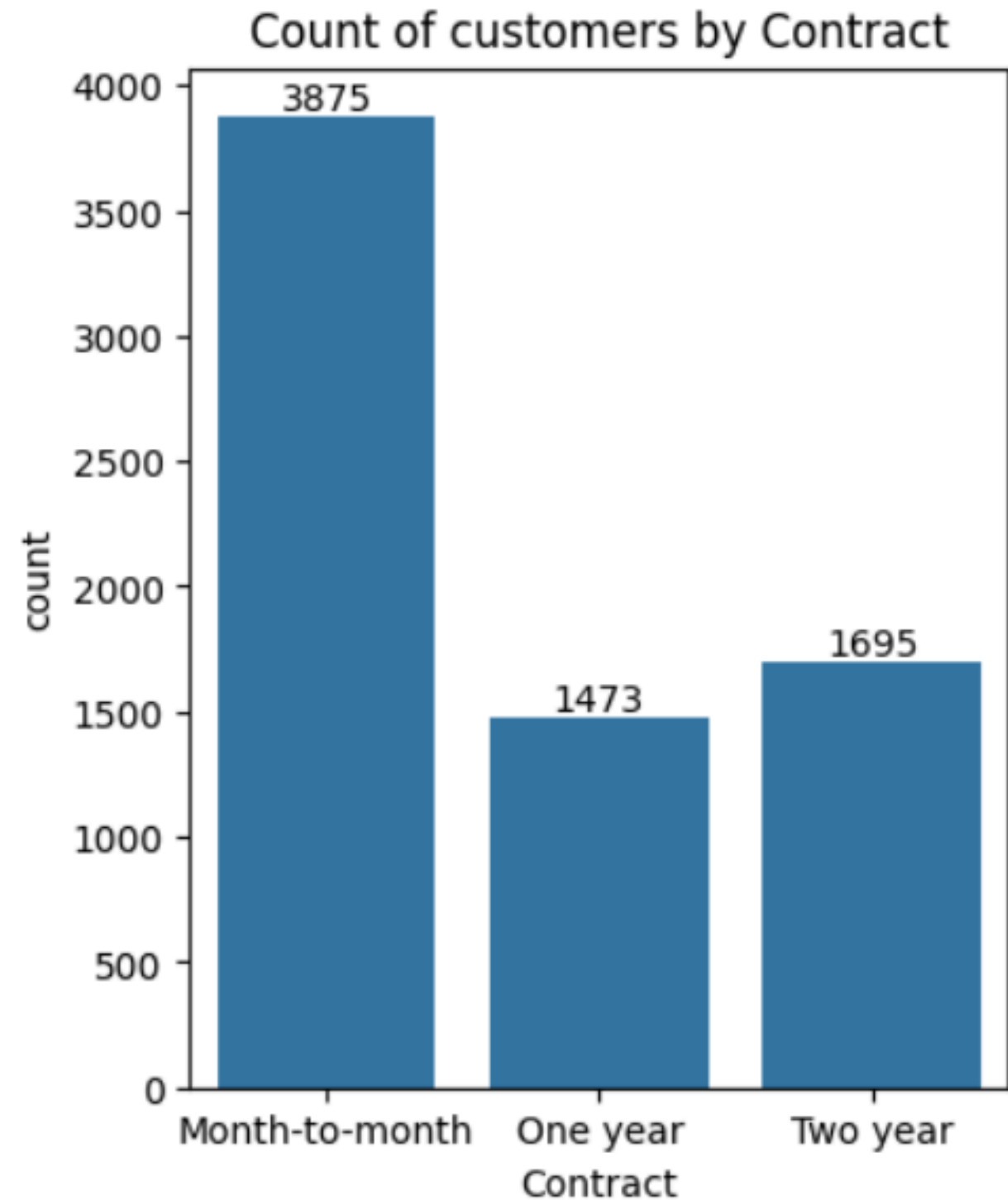

```
[61]: plt.figure(figsize=(6,4))
sns.histplot(x="tenure",data=data,bins=72,hue="Churn")
plt.title("Tenure based on Churn")
plt.show()
```



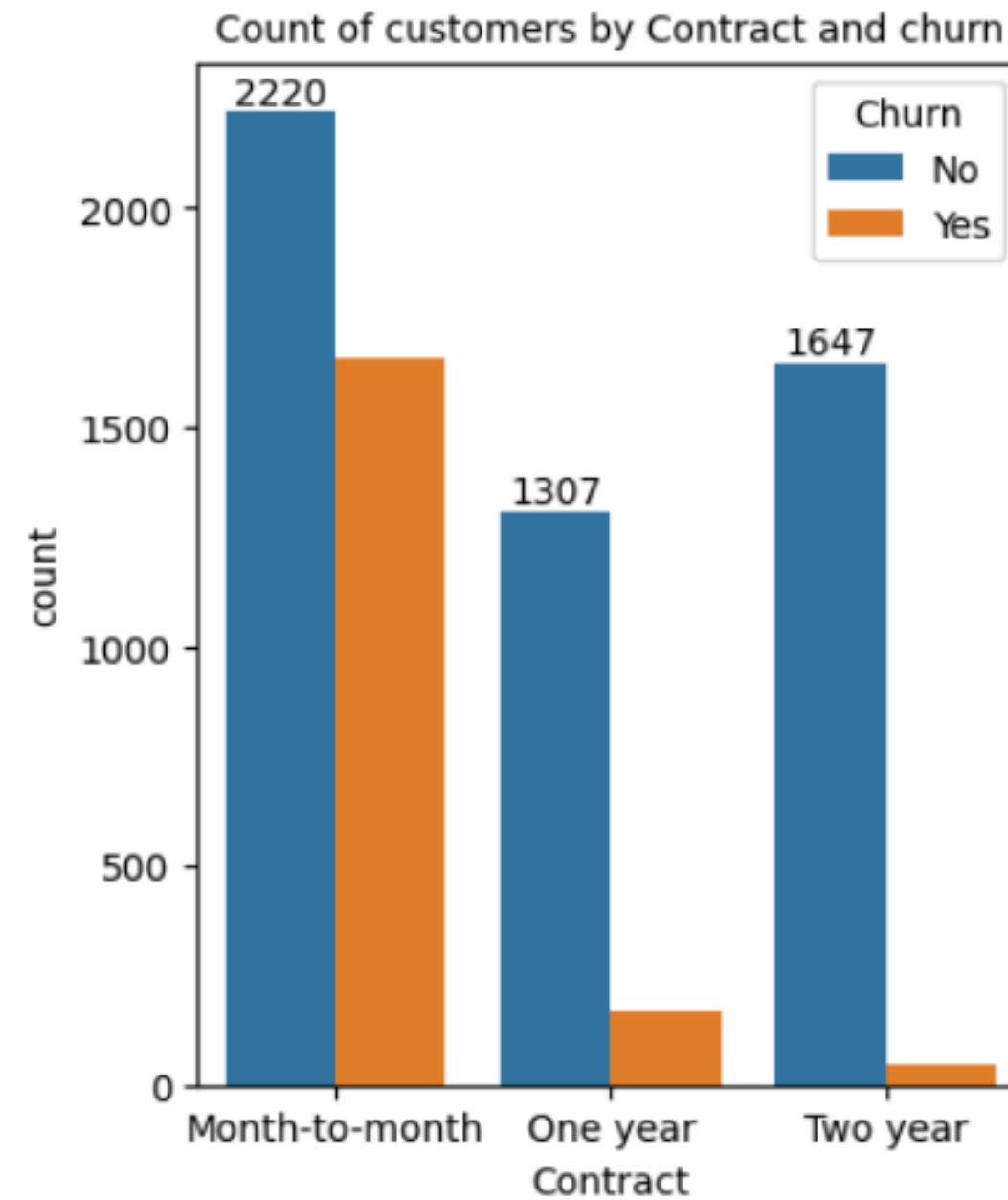
#people who have used our services for a long time have stayed and people who have used our services for 1 or 2 months have churned.

Count of people based on contract

```
[63]: plt.figure(figsize=(4,5))
      ax=sns.countplot(x="Contract",data=data)
      ax.bar_label(ax.containers[0])
      plt.title("Count of customers by Contract")
      plt.show()
```



```
[67]: plt.figure(figsize=(4,5))
      ax=sns.countplot(x="Contract",data=data,hue="Churn")
      ax.bar_label(ax.containers[0])
      plt.title("Count of customers by Contract and churn",fontsize=10)
      plt.show()
```



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

count plot of servies

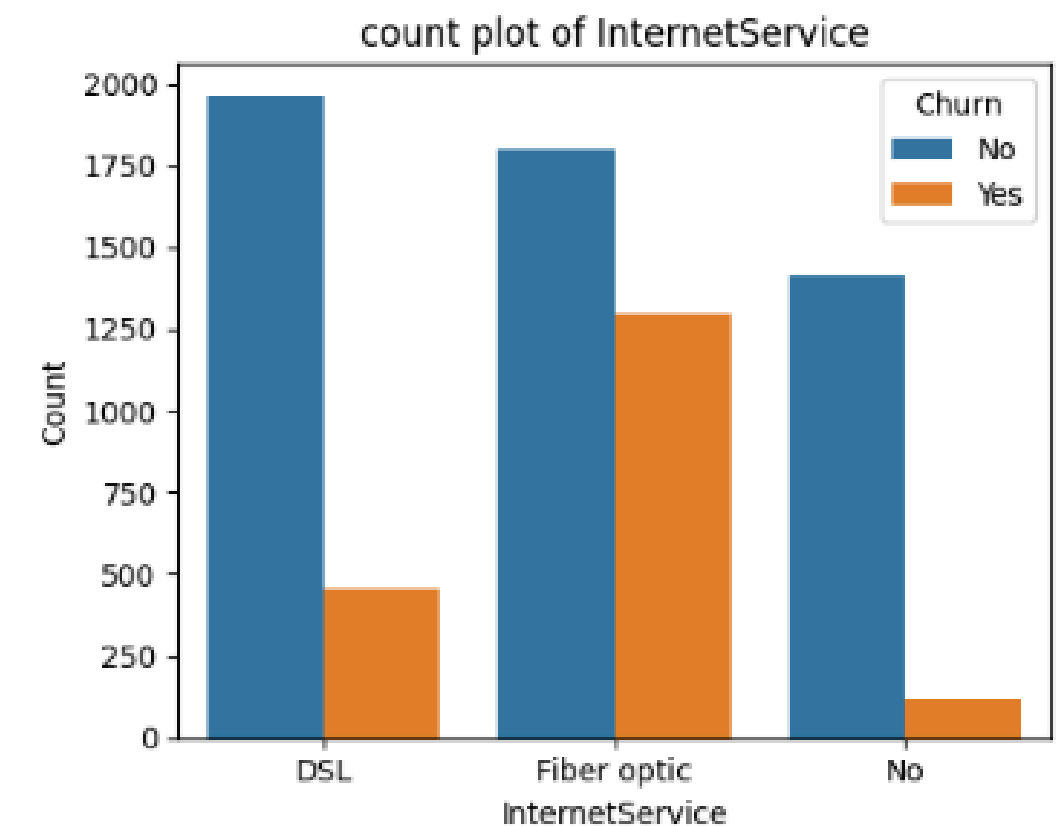
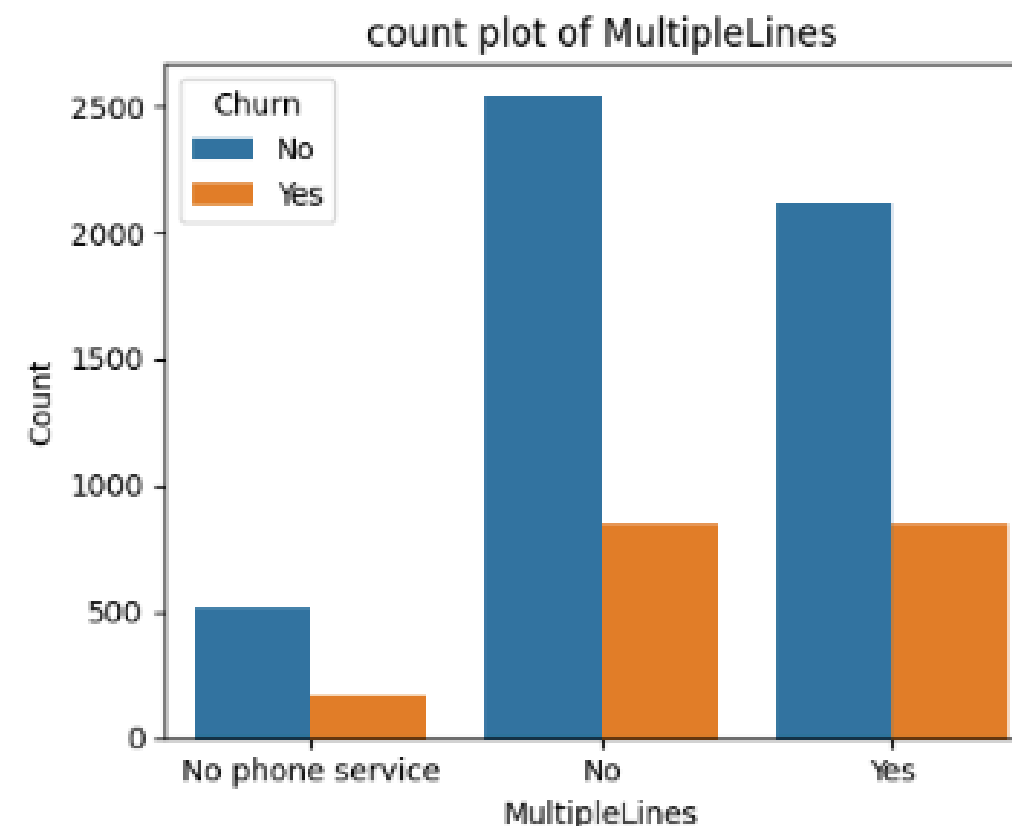
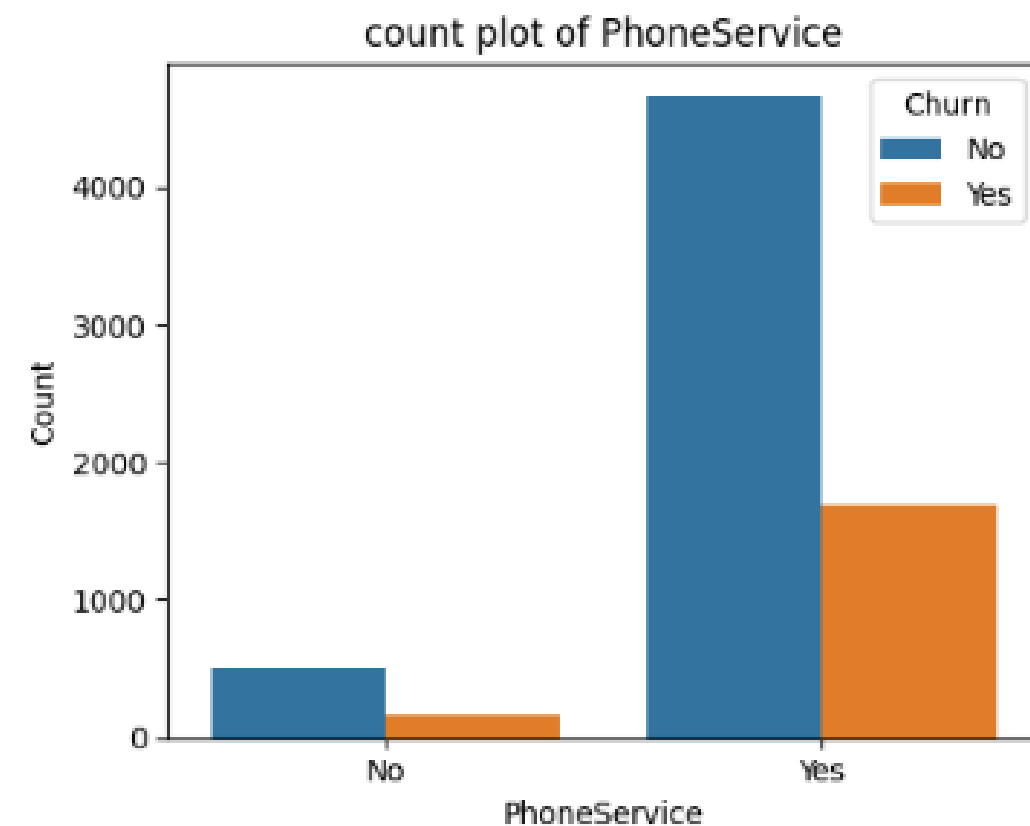
```
[76]: columns=['PhoneService', 'MultipleLines', 'InternetService',
            'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
            'StreamingTV', 'StreamingMovies']
n_cols=3
n_rows=(len(columns)+n_cols -1)//n_cols

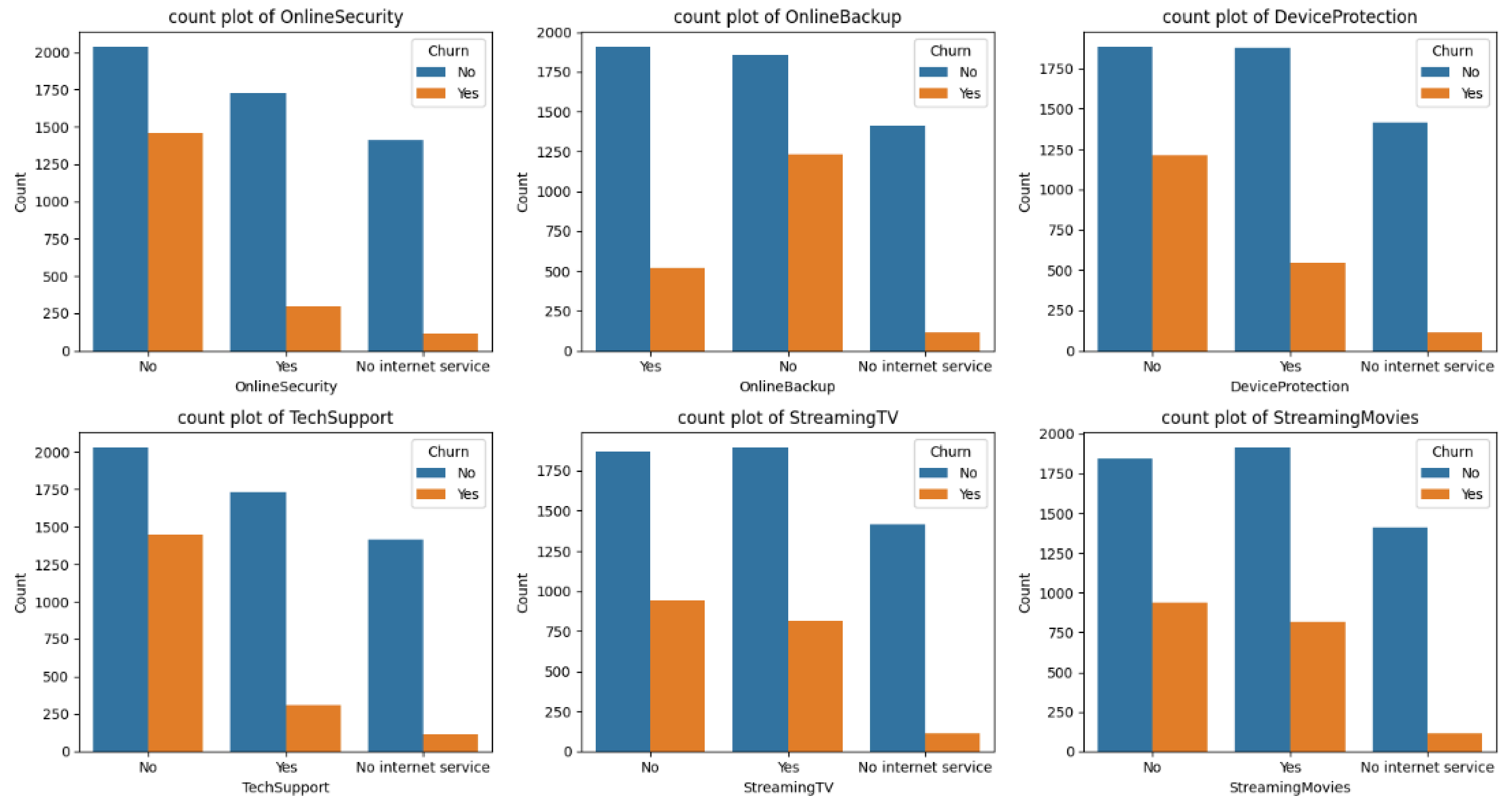
fig,axes=plt.subplots(n_rows,n_cols,figsize=(15,n_rows*4))

axes = axes.flatten()
for i,col in enumerate(columns):
    sns.countplot(x=col,data=data,ax=axes[i],hue="Churn")
    axes[i].set_title(f'count plot of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel("Count")

for j in range(i+1,len(axes)):
    fig.delaxes(axes[j])

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

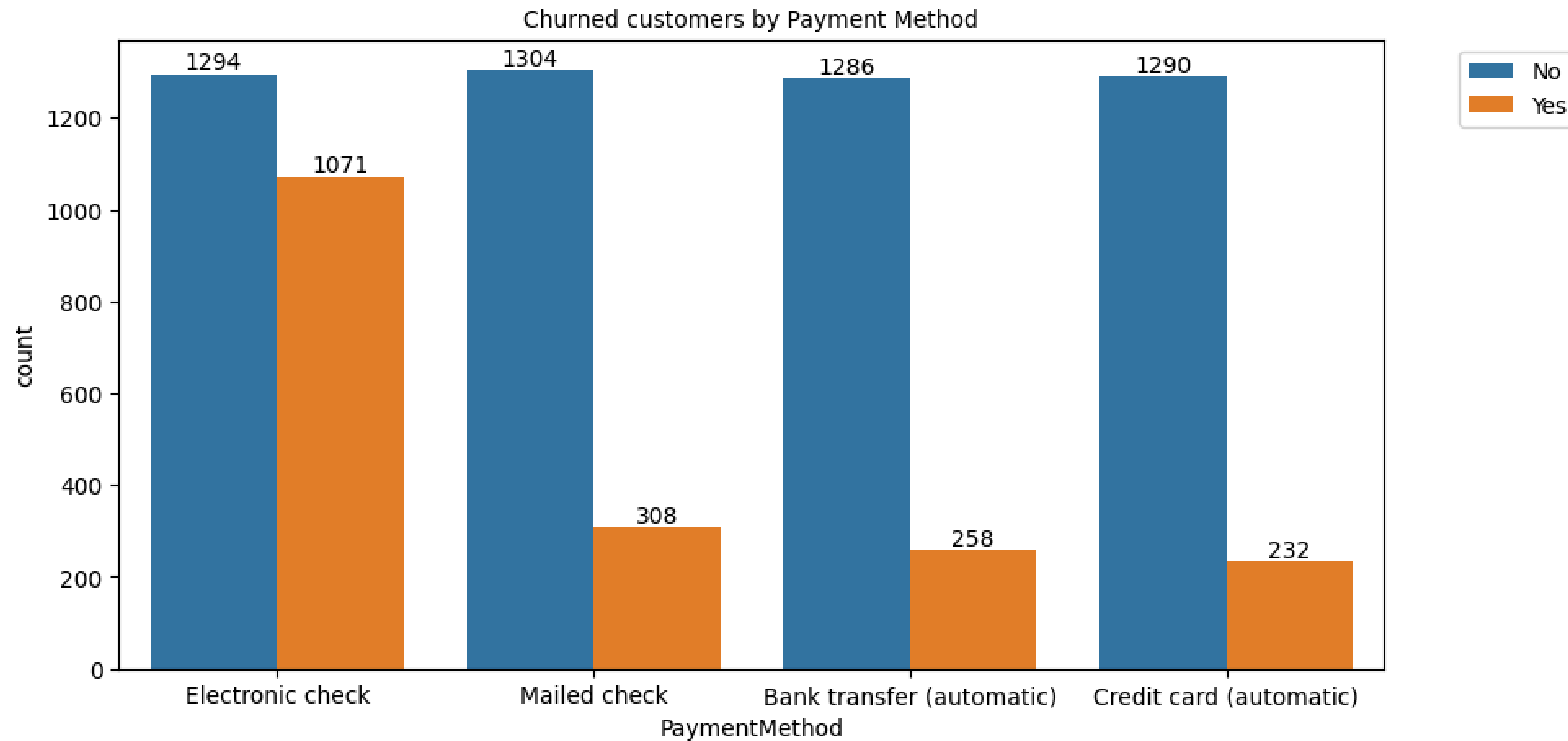




#The majority of customers who do not churn tend to have services like PhonService,InternetService (particularly DSL),and OnlineSecurity enabled.For Services like OnlineBackup,TechSupport and StreamingTV,churn rated are noticeably higher when these services are not used or are unavaliabe.

Plot based on Payment Method and churn

```
•[6]: plt.figure(figsize=(10,5))
      ax=sns.countplot(x="PaymentMethod",data=data,hue="Churn")
      ax.bar_label(ax.containers[0])
      ax.bar_label(ax.containers[1])
      plt.title("Churned customers by Payment Method",fontsize=10)
      plt.legend(bbox_to_anchor=(1.05,1),loc=2)
      plt.show()
```



#Customer is likely to churn when he is using Electronic Check as a payment method.