In [68]:

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

In [69]:

```
df = pd.read_csv("D:\\Python Projects\\1st Project churn analysis\\Customer Churn.csv")

df.head(15)
```

Out[69]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic
5	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic
6	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic
7	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL
8	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic
9	6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL
10	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL
11	7469-LKBCI	Male	0	No	No	16	Yes	No	No
12	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic
13	0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic
14	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic

15 rows × 21 columns

In [70]:
```
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
```

In [71]:
```
# Replacing blanks with 0 as tenure is 0 and no total charges are recorded
# Changed data type of TotalCharges into float from object
```

In [72]:
```
df["TotalCharges"] = df["TotalCharges"].replace(" ", "0")
df["TotalCharges"] = df["TotalCharges"].astype("float")
```

In [73]:
```
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
```

In [74]:
```
df.isnull().sum()
```

Out[74]:
```
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
```

In [75]:
```
# this will show there are any null values in entire dataset

df.isnull().sum().sum()
```

Out[75]:
```
0
```

In [76]:
```
df.describe()
```

Out[76]:

	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

In [77]:
```
df.duplicated().sum()
```

Out[77]:
```
0
```

In [78]:
```
df["customerID"].duplicated().sum()
```

Out[78]:
```
0
```

In [79]:

```
# converting 0 and 1 values of SeniorCitizen to Yes/No to make it easier to understand


# Now let's replace the values in the 'SeniorCitizen' column
df['SeniorCitizen'] = df['SeniorCitizen'].replace({1: 'Yes', 0: 'No'})

# Display the first few rows to confirm the changes
df.head(25)
```

Out[79]:

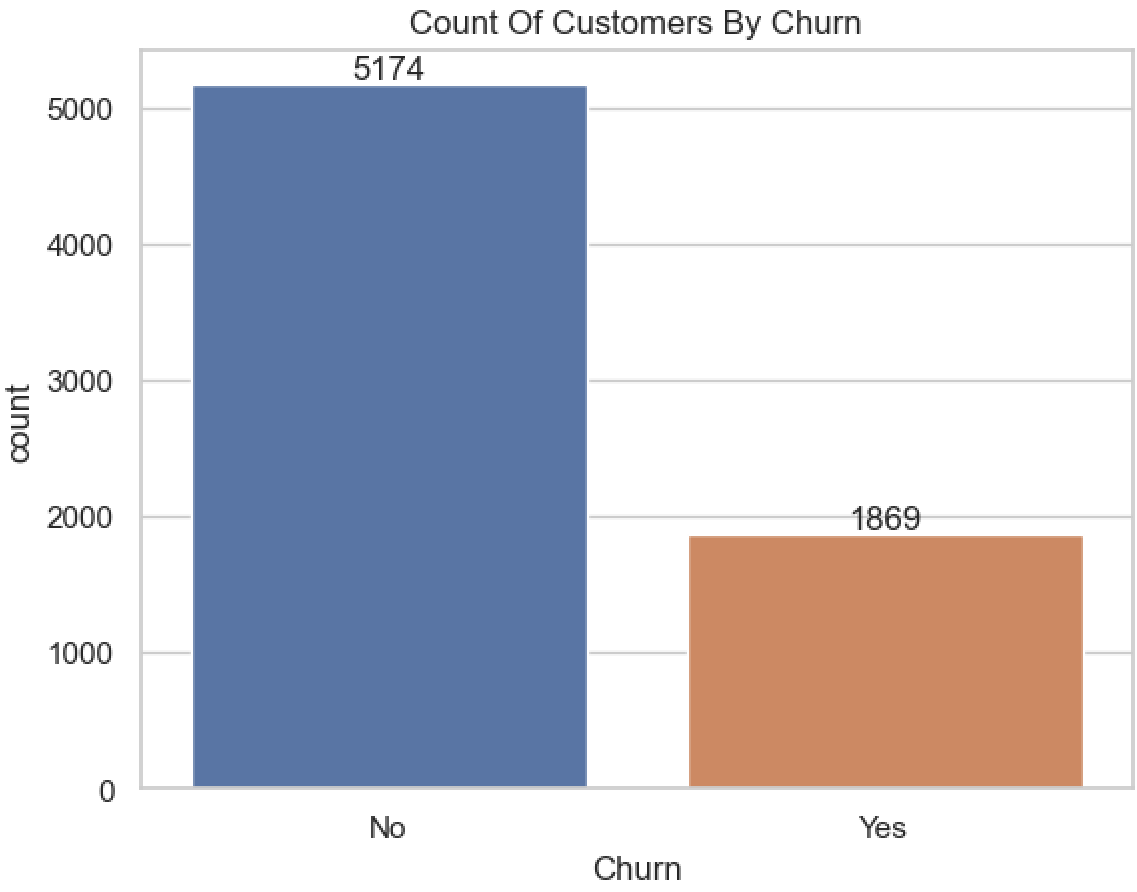
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService
0	7590-VHVEG	Female	No	Yes	No	1	No	No phone service	DSL
1	5575-GNVDE	Male	No	No	No	34	Yes	No	DSL
2	3668-QPYBK	Male	No	No	No	2	Yes	No	DSL
3	7795-CFOCW	Male	No	No	No	45	No	No phone service	DSL
4	9237-HQITU	Female	No	No	No	2	Yes	No	Fiber optic
5	9305-CDSKC	Female	No	No	No	8	Yes	Yes	Fiber optic
6	1452-KIOVK	Male	No	No	Yes	22	Yes	Yes	Fiber optic
7	6713-OKOMC	Female	No	No	No	10	No	No phone service	DSL
8	7892-POOKP	Female	No	Yes	No	28	Yes	Yes	Fiber optic
9	6388-TABGU	Male	No	No	Yes	62	Yes	No	DSL
10	9763-GRSKD	Male	No	Yes	Yes	13	Yes	No	DSL
11	7469-LKBCI	Male	No	No	No	16	Yes	No	No
12	8091-TTVAX	Male	No	Yes	No	58	Yes	Yes	Fiber optic
13	0280-XJGEX	Male	No	No	No	49	Yes	Yes	Fiber optic
14	5129-JLPIS	Male	No	No	No	25	Yes	No	Fiber optic
15	3655-SNQYZ	Female	No	Yes	Yes	69	Yes	Yes	Fiber optic
16	8191-	Female	No	No	No	52	Yes	No	No

	XWSZG								
17	9959-WOFKT	Male	No	No	Yes	71	Yes	Yes	Fiber optic
18	4190-MFLUW	Female	No	Yes	Yes	10	Yes	No	DSL
19	4183-MYFRB	Female	No	No	No	21	Yes	No	Fiber optic
20	8779-QRDMV	Male	Yes	No	No	1	No	No phone service	DSL
21	1680-VDCWW	Male	No	Yes	No	12	Yes	No	No
22	1066-JKSGK	Male	No	No	No	1	Yes	No	No
23	3638-WEABW	Female	No	Yes	No	58	Yes	Yes	DSL
24	6322-HRPFA	Male	No	Yes	Yes	49	Yes	No	DSL

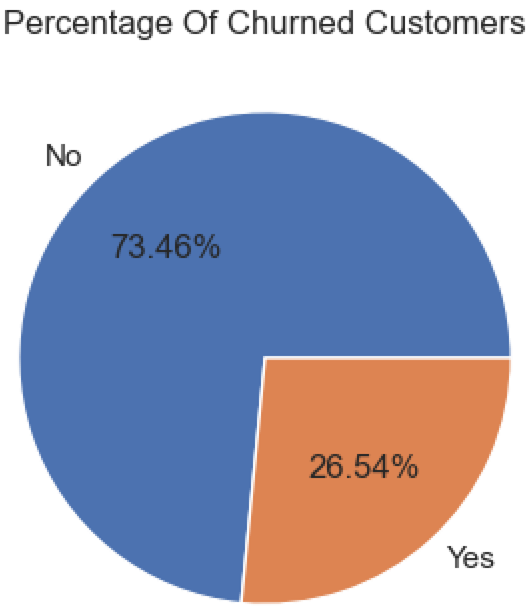
25 rows × 21 columns

In [80]:

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

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

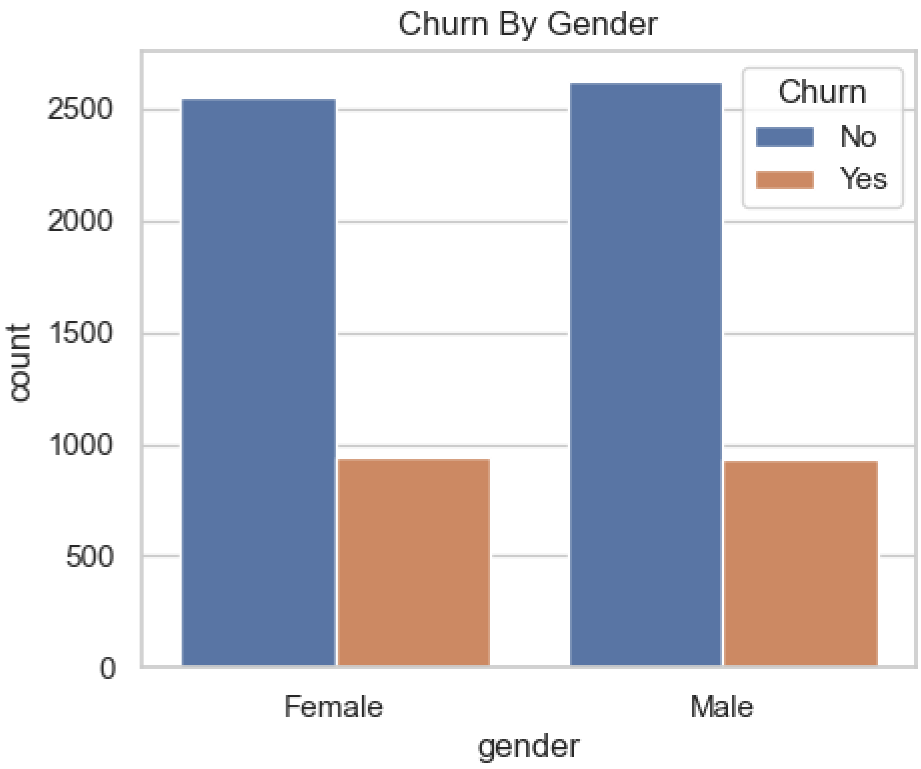


In [81]:
```
plt.figure(figsize = (4,4))
gb = df.groupby("Churn").agg({'Churn':"count"})
plt.pie(gb['Churn'], labels = gb.index, autopct = "%1.2f%%")
plt.title("Percentage Of Churned Customers")
plt.show()
```

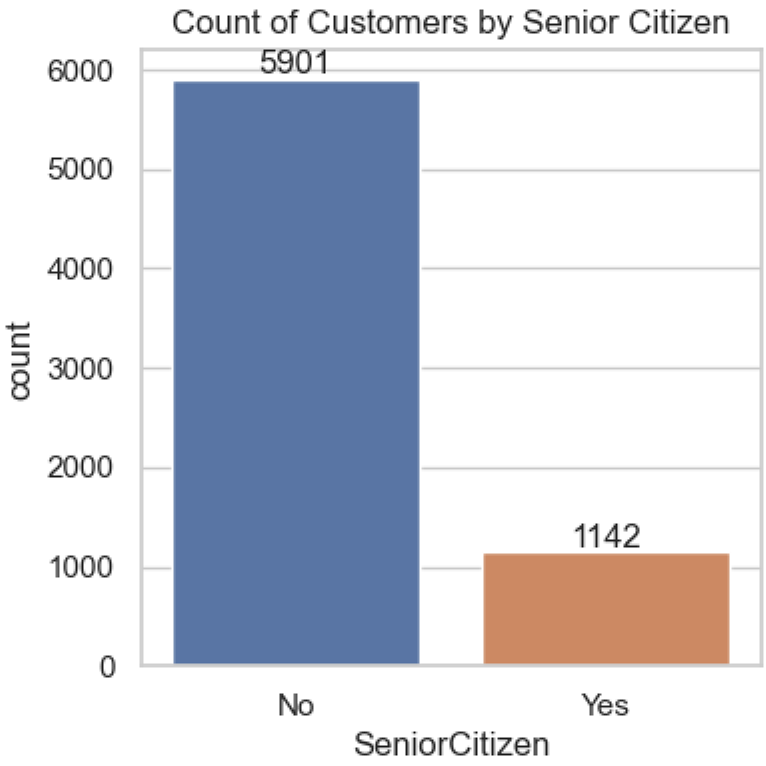


In [82]:
```
# from the given pie chart we can conclude that 26.54% of our customer have churned out
# now lets explore the reason behind it
```

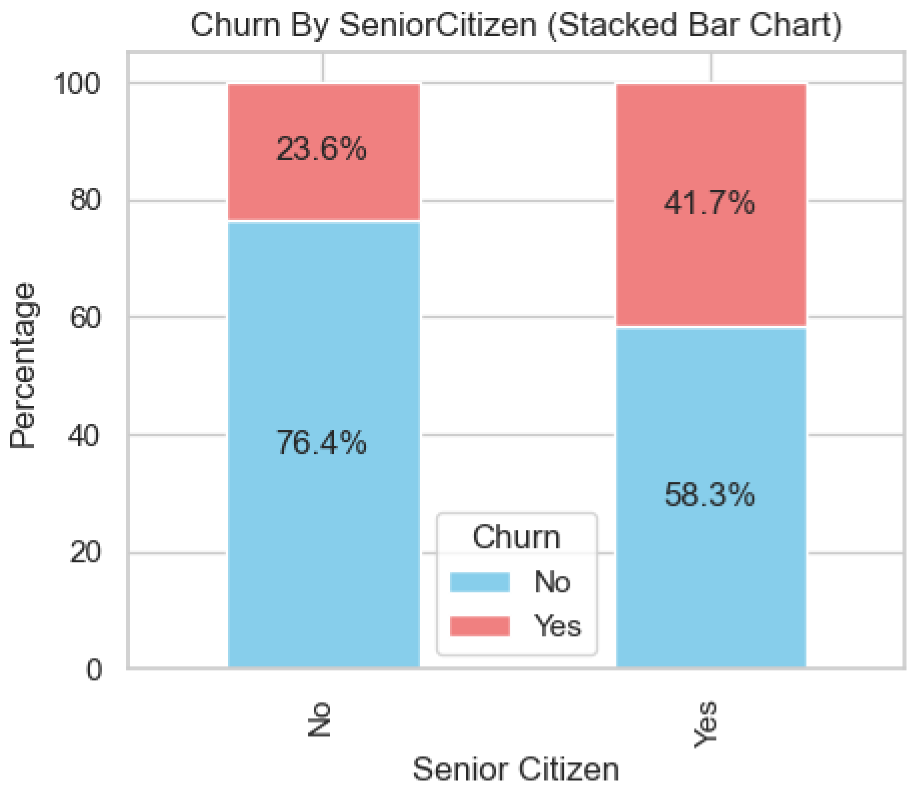
In [83]:
```
plt.figure(figsize = (5,4))
sns.countplot(x = "gender", data = df, hue = "Churn")
plt.title("Churn By Gender")
plt.show()
```



```
In [84]: 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()
```

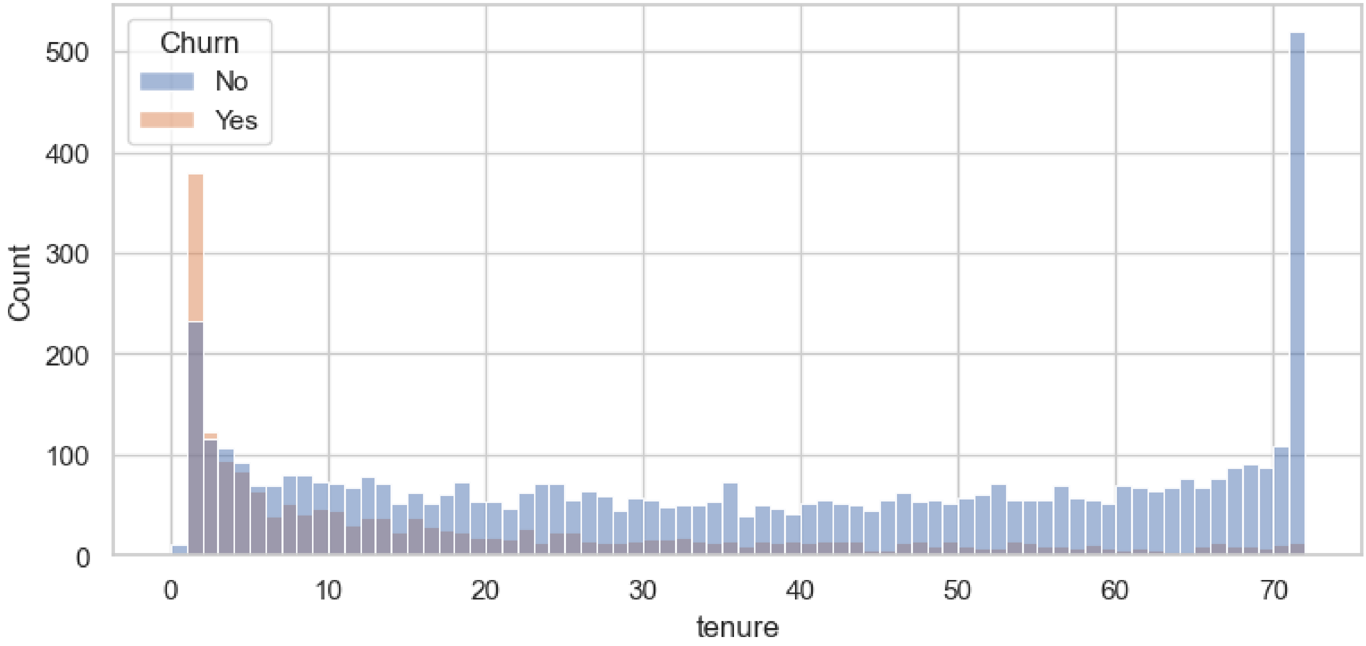


```
In [85]: # Calculate percentage
         senior_churn = df.groupby(['SeniorCitizen', 'Churn']).size().reset_index(name='counts')
         senior_churn['percent'] = senior_churn.groupby('SeniorCitizen')['counts'].transform(lamb

         # Pivot to make it suitable for stacked bar plot
         pivot_table = senior_churn.pivot(index='SeniorCitizen', columns='Churn', values='percent
         # Plot the stacked bar chart
         pivot_table.plot(kind='bar', stacked=True, figsize=(5, 4), color=['skyblue', 'lightcoral

         # Add labels and title
         plt.title("Churn By SeniorCitizen (Stacked Bar Chart)")
         plt.ylabel("Percentage")
         plt.xlabel("Senior Citizen")

         # Add percentage labels on the bars
         for i, bar in enumerate(plt.gca().patches):
             height = bar.get_height()
             plt.gca().text(
                 bar.get_x() + bar.get_width() / 2,
                 bar.get_y() + height / 2,
                 f'{height:.1f}%',
                 ha='center', va='center'
             )

         plt.show()
```



In [86]:
```
# comparative a greated percentage of people in senior citizen category have churned
```

In [87]:
```
plt.figure(figsize = (9,4))
sns.histplot(x = "tenure", data = df, bins = 72, hue = "Churn")
plt.show()
```

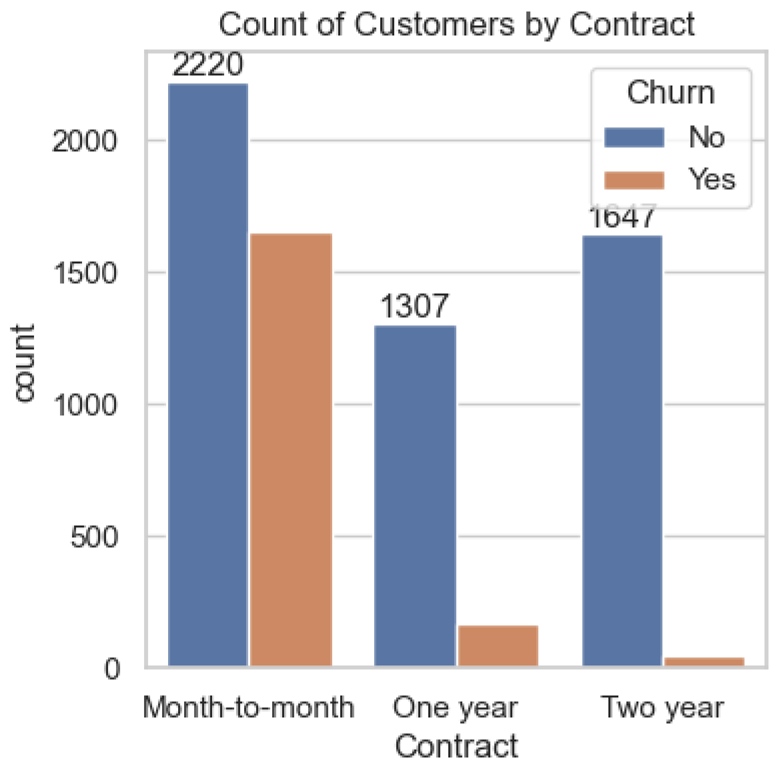


In [88]:

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

In [89]:

```
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()
```



In [90]:

```
# people who have month to month contract are likely to churn then from those have 1 or
```

In [91]:

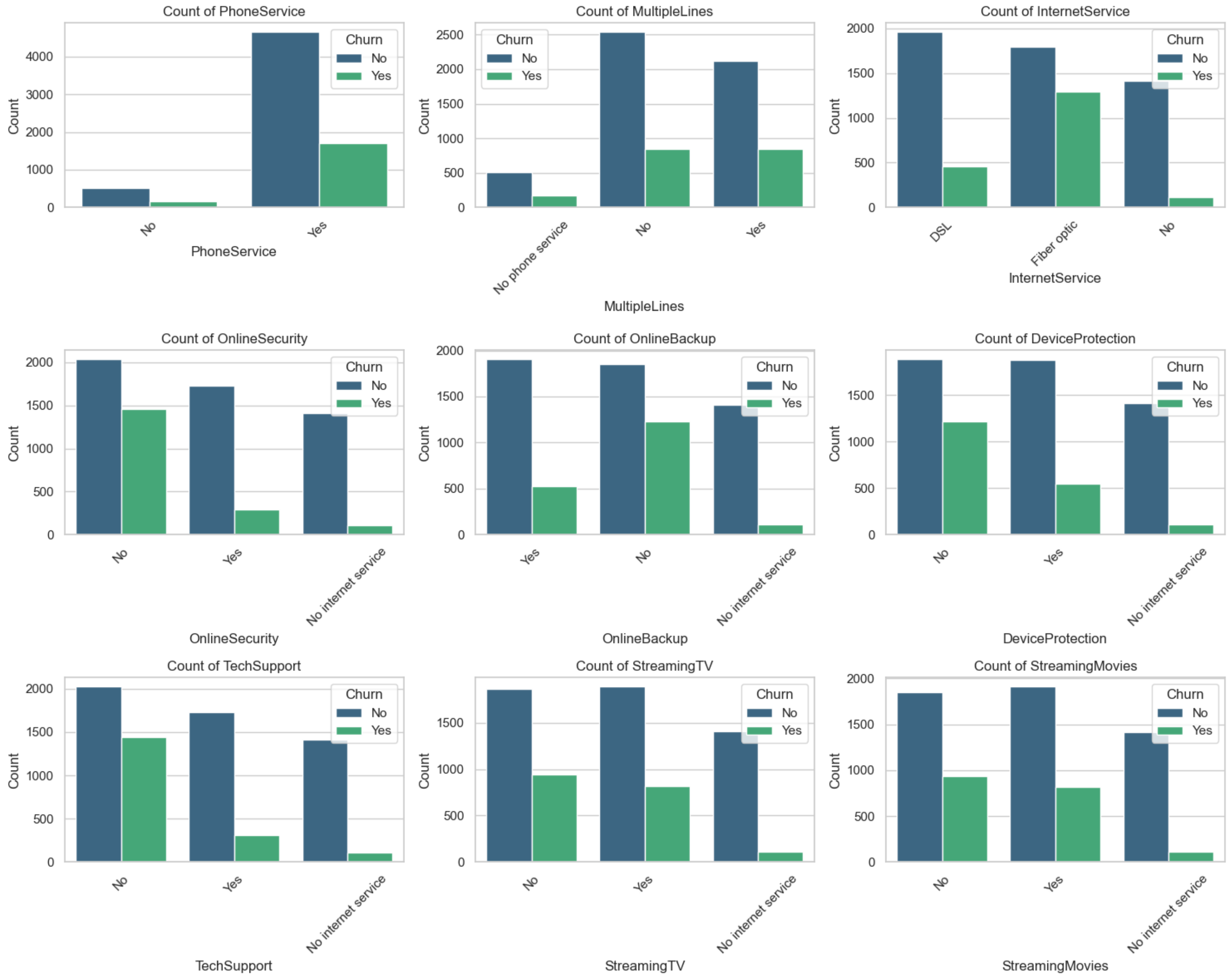
```
df.columns.values
```

Out[91]:

```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

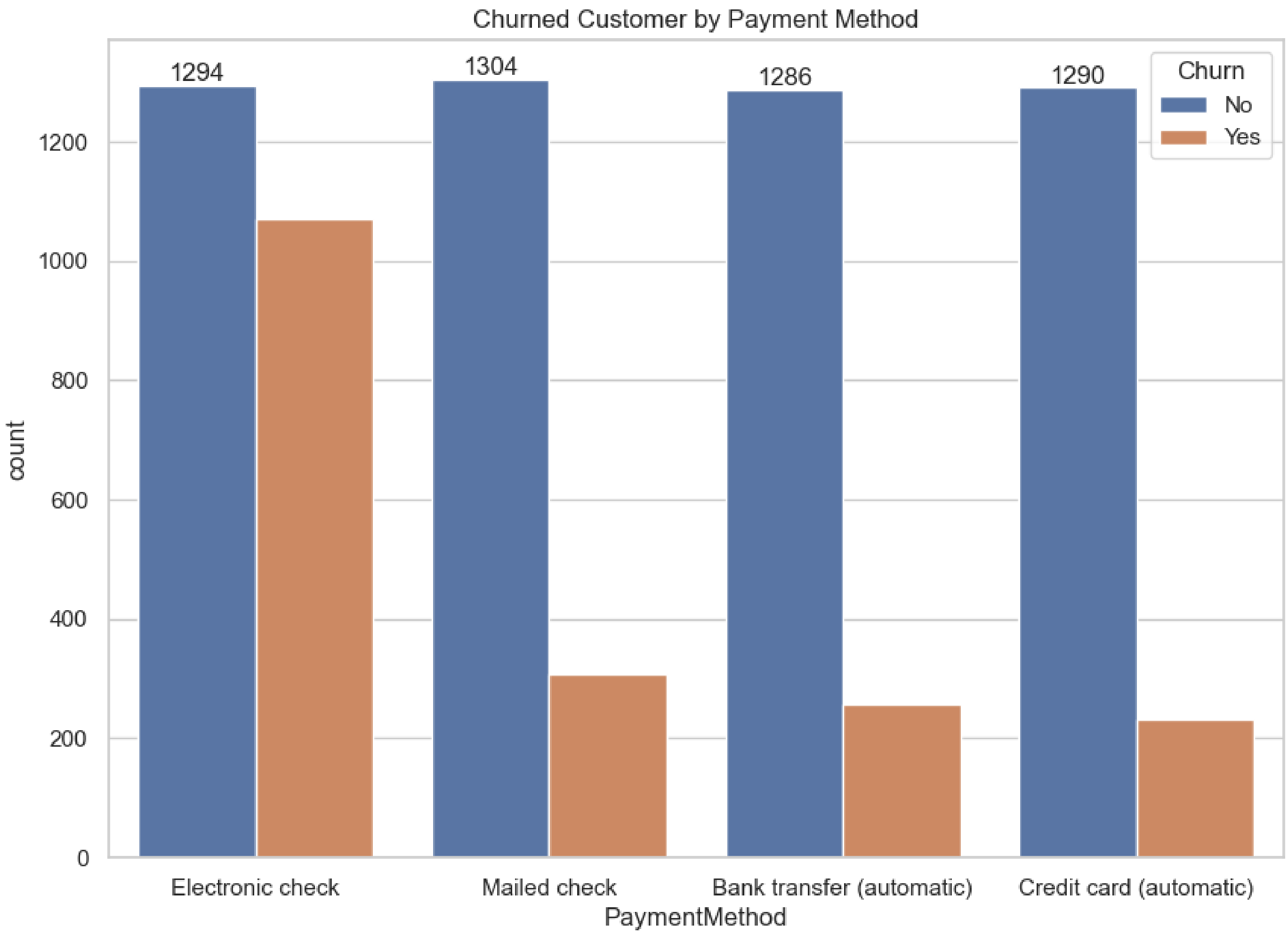
In [92]:

```
# Example DataFrame (replace with your actual data)
# df = pd.read_csv('your_data.csv')

columns = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

sns.set(style="whitegrid")

num_cols = len(columns)
n_rows = 3
n_cols_plot = 3

plt.figure(figsize=(n_cols_plot * 5, n_rows * 4))

for i, column in enumerate(columns):
    plt.subplot(n_rows, n_cols_plot, i + 1)
    sns.countplot(data=df, x=column, palette="viridis", hue = df["Churn"])
    plt.title(f'Count of {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.xticks(rotation=45)

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



In [93]:

```
plt.figure(figsize = (10,7))
ax = sns.countplot(x = "PaymentMethod", data = df, hue = "Churn")
ax.bar_label(ax.containers[0])
plt.title("Churned Customer by Payment Method")
plt.show()
```



In [94]:

```
# Customer is likely to churn whenn he is using electric check as a payment
```

Here’s the output form for Telecom churn data analysis:

1. Churn Distribution: 26.54% of customers have churned.

2. Churn by Gender: Similar churn rates between males and females.

3. Senior Citizens: Higher churn rates among senior citizens.

4. Contract Type: Month-to-month contracts see significantly higher churn.

5. Tenure: Short-tenure customers are more likely to churn.

6. Payment Method: Customers using electronic checks have higher churn rates.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: