

```
In [1]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('Customer Churn.csv')
```

Out[1]:

	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
...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber

7043 rows \times 21 columns

In []:

In []:

```
In [2]:
df.shape
```

Out[2]:
(7043, 21)

```
In [3]:
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 blanks with 0 in TotalCharges

In [6]:

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

In [7]:

```
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 [8]:

df.isnull().sum()

Out[8]:

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 [9]:

df.describe()

Out[9]:

	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 [10]:

df.duplicated().sum()

Out[10]:

np.int64(0)

In [11]:

def conv(value):

if value == 1 :

return "yes"

else :

return "no"

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

In [12]:

df.head()

Out[12]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
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-	Male	no	No	No	45	No	No phone	DSL

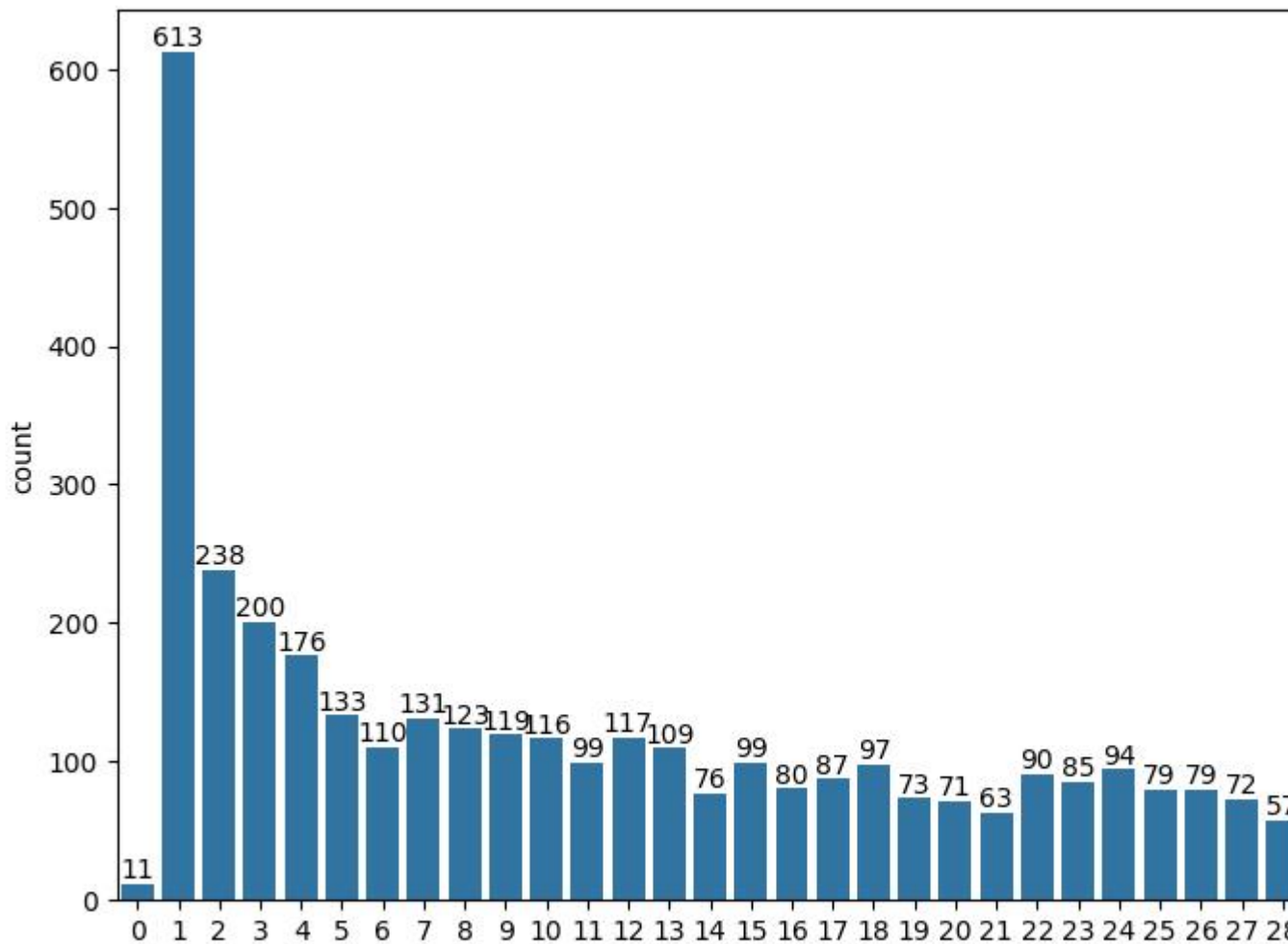
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetS
	CFOCW							service	
4	9237-HQITU	Female	no	No	No	2	Yes	No	Fiber opt

5 rows × 21 columns

In [92]:

```
plt.figure(figsize=(20,6))d = sns.countplot( x ='tenure' , data = df)for bars in
d.containers:
```

```
    d.bar_label(bars)plt.xlabel('total months')plt.ylabel('count')plt.show()
```



In []:

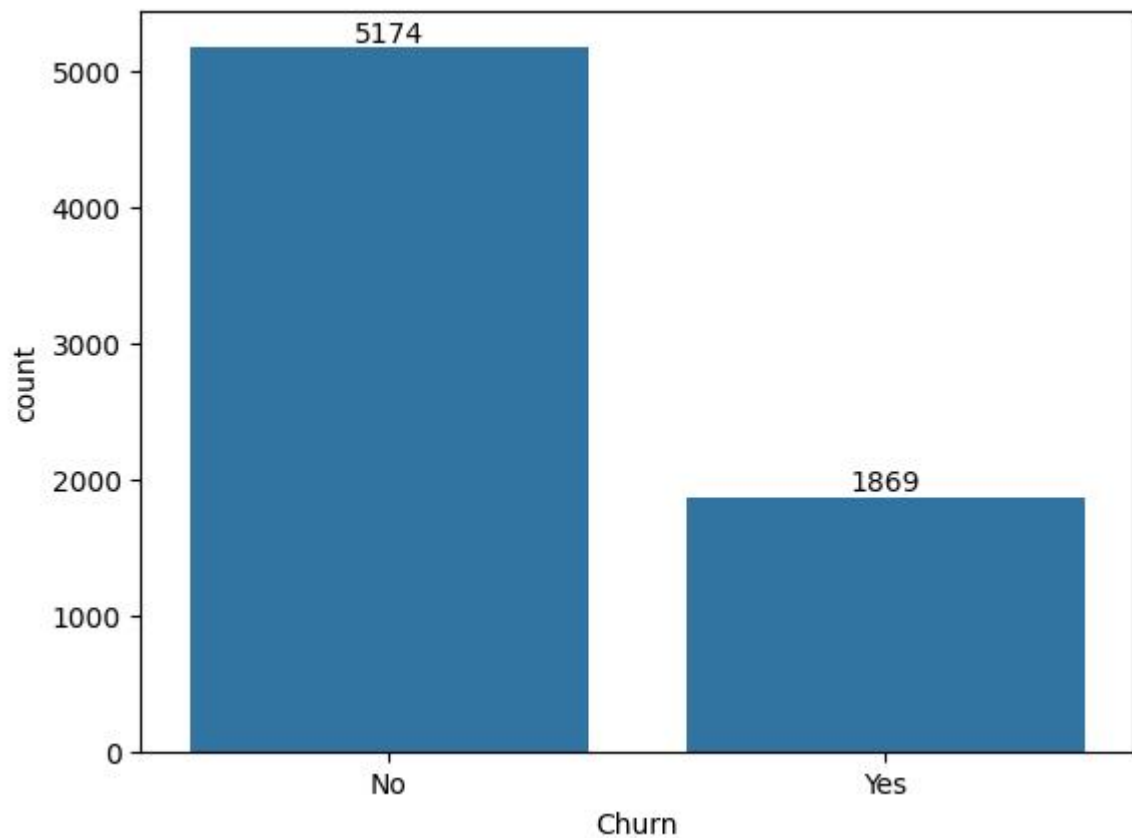
With this graph, you can directly observe:

For each tenure value, how many customers churned versus stayed. Whether churn occurs more frequently for shorter tenures, longer tenures, or is evenly distributed.

In [93]:

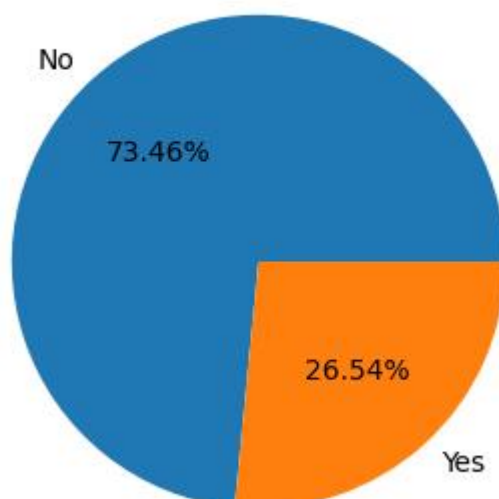
```
c = sns.countplot(x = 'Churn' , data =df )for bars in c.containers:
```

```
    c.bar_label(bars)plt.show()
```



```
In [27]:  
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 Customer')plt.show()
```

Percentage of Churned Customer

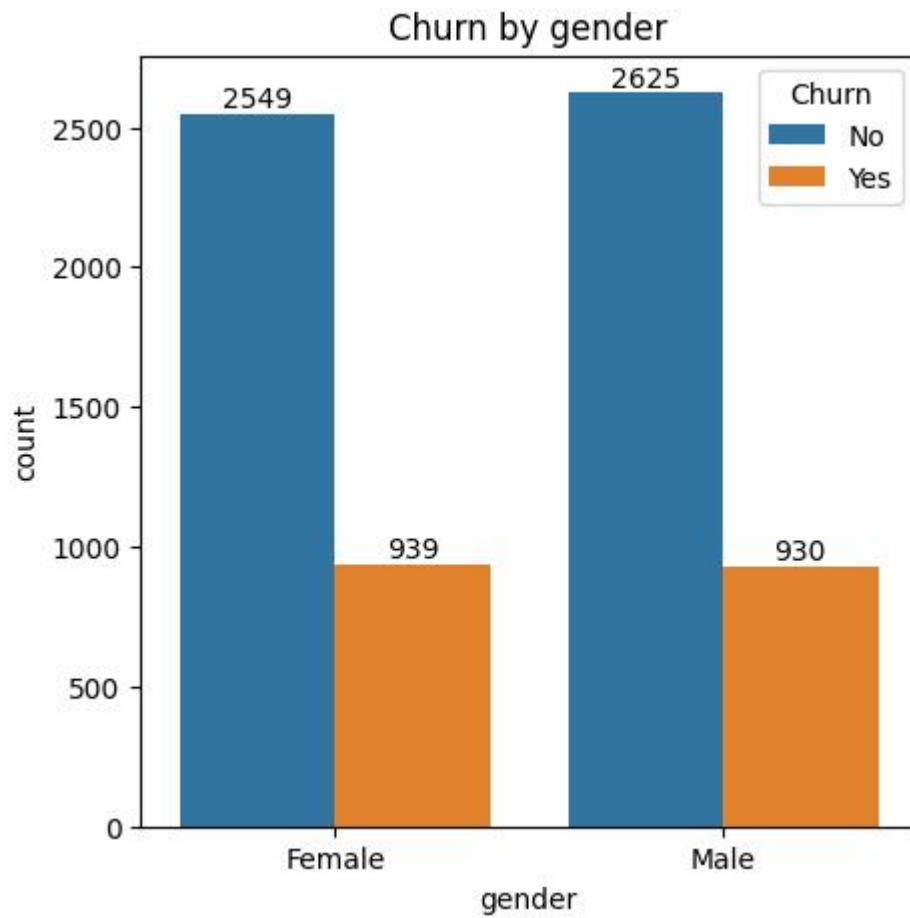


from above pie chart we can conclude 26.54 people churned out .
In [38]:

```
plt.figure(figsize=(5,5))g= sns.countplot(x = 'gender',data =df , hue = 'Churn')for bars
```

in g.containers:

```
g.bar_label(bars)plt.title('Churn by gender')plt.show()
```

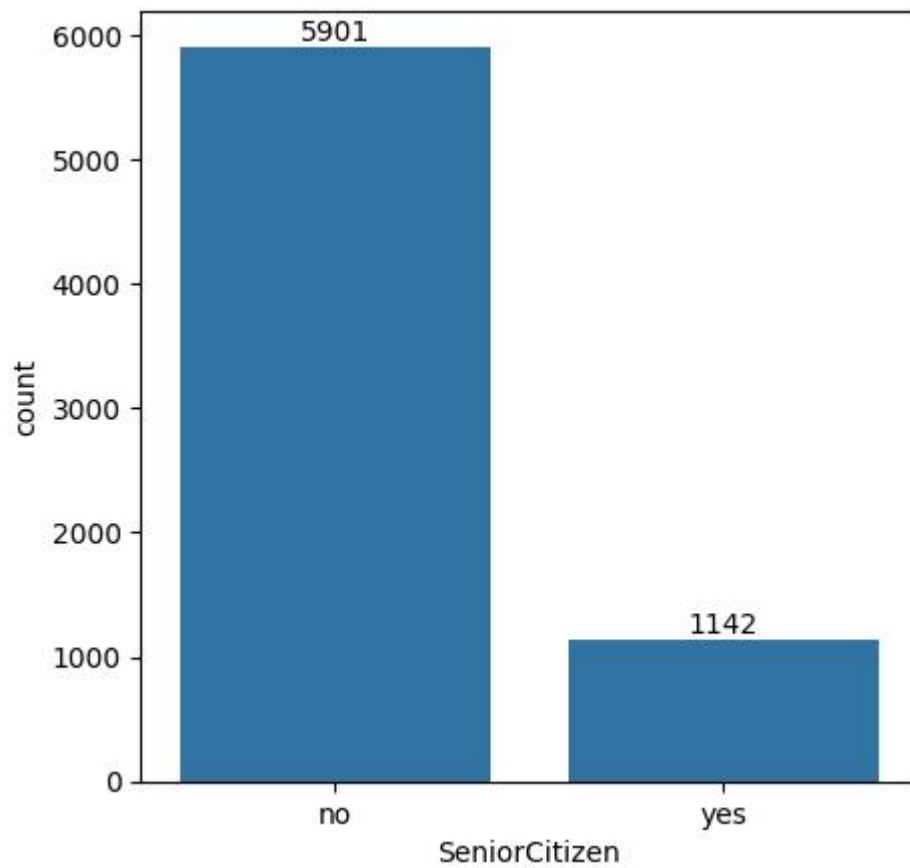


In [97]:

```
plt.figure(figsize=(5,5))g= sns.countplot(x = 'SeniorCitizen',data =df )for bars in
```

g.containers:

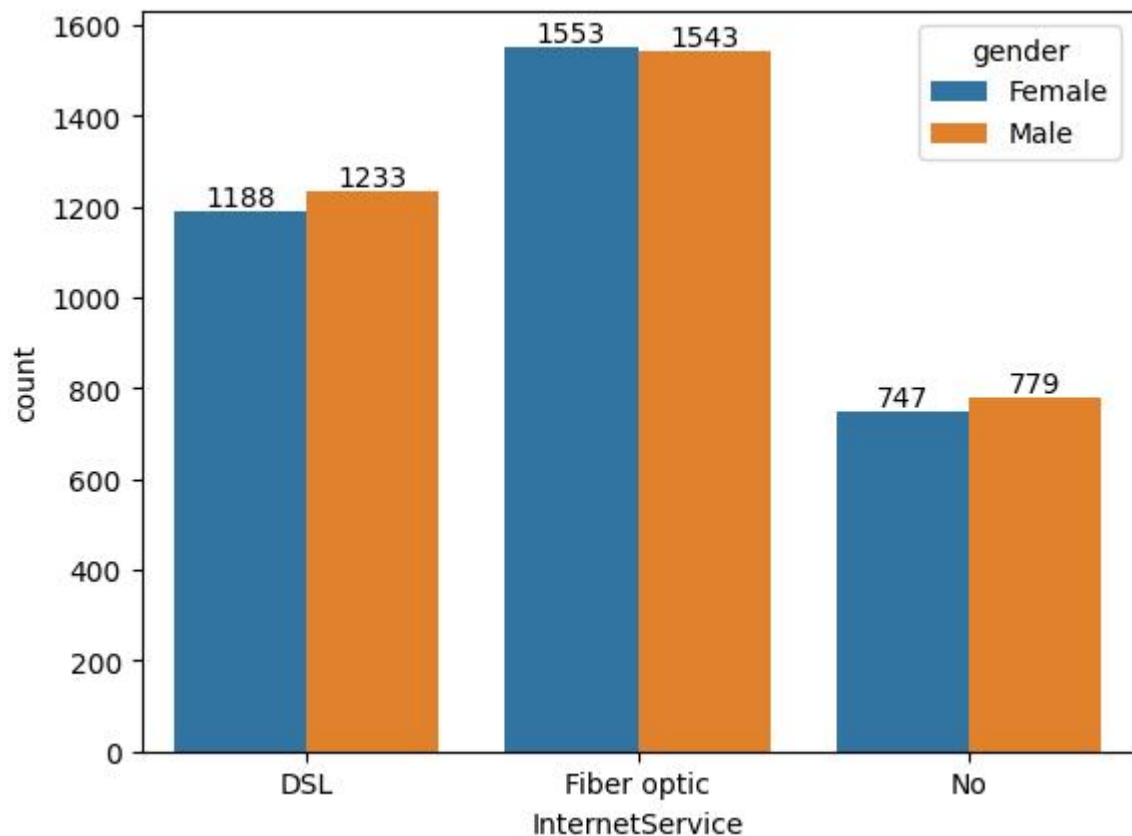
```
g.bar_label(bars)plt.show()
```



In [49]:

```
h = sns.countplot(x = 'InternetService' , data = df , hue = 'gender')for bars in h.containers:
```

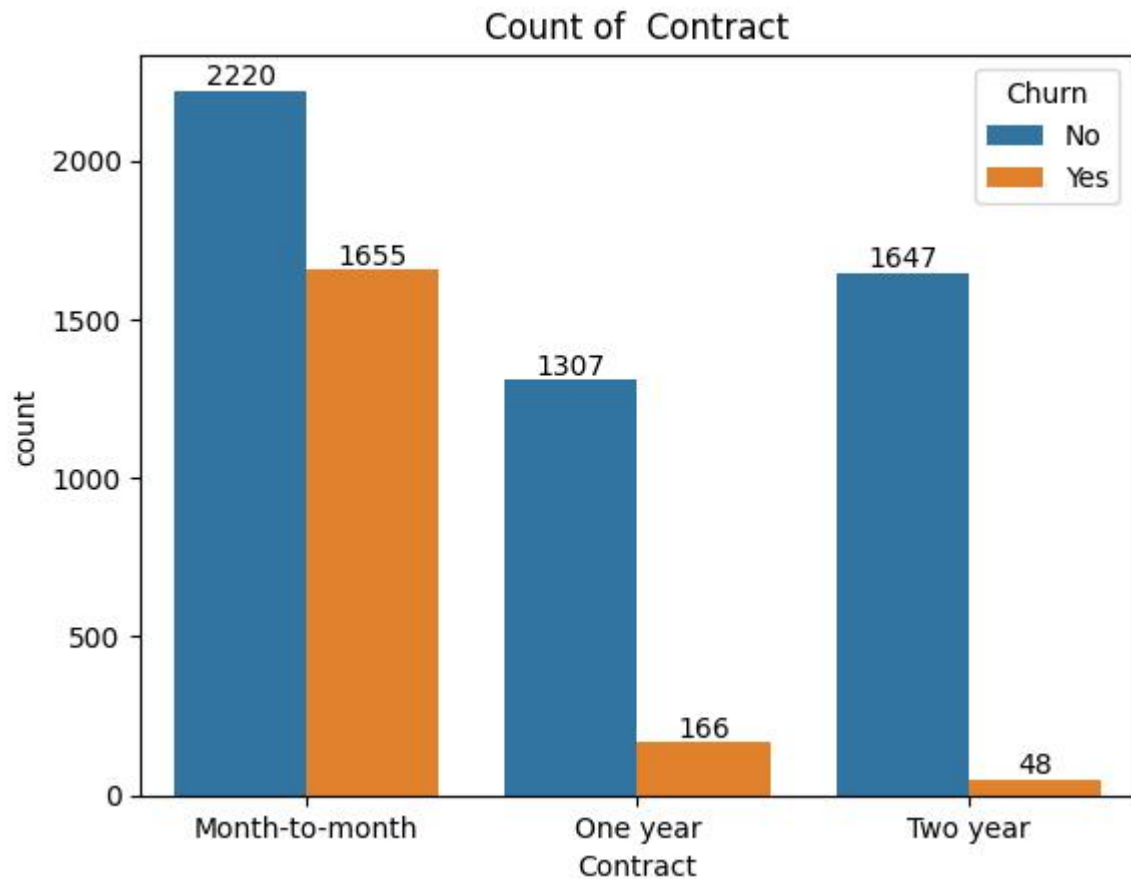
```
    h.bar_label(bars)plt.show()
```

With the help of above figure we can conclude that most INTERNET SERVICE are of Fiber optic most by Female than male and male with DSL followed by female and no IS by male

In [51]:

```
c = sns.countplot(x = 'Contract', data = df, hue = 'Churn') plt.title('Count of Contract')  
for bars in c.containers:  
    c.bar_label(bars) plt.show()
```

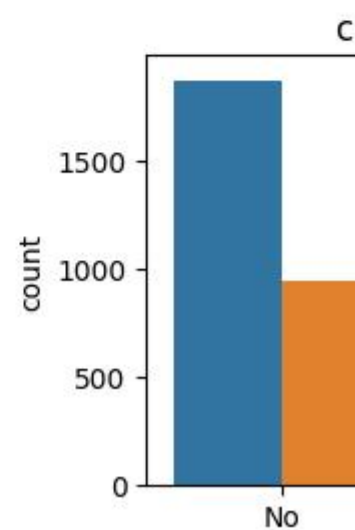
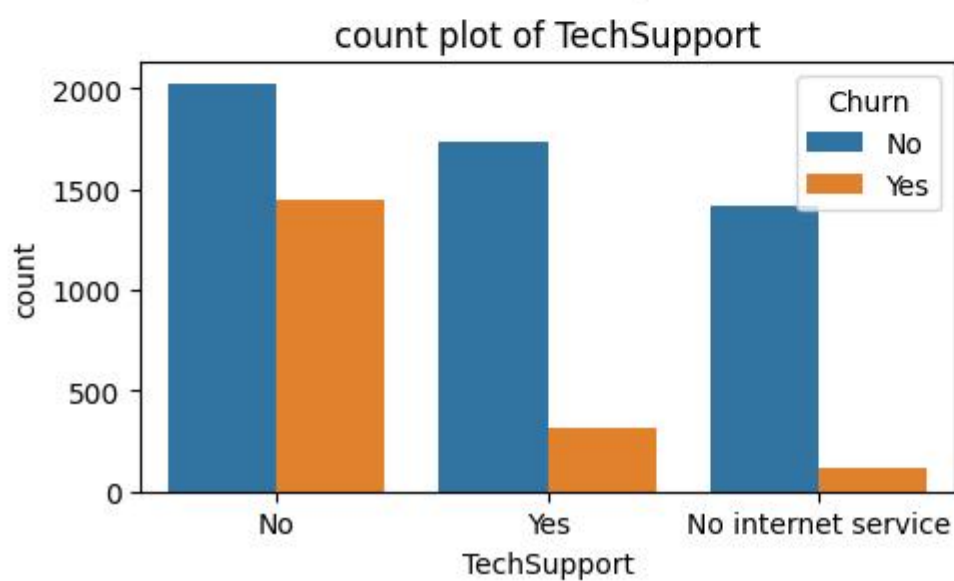
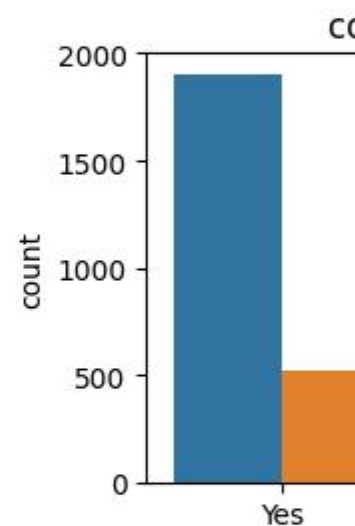
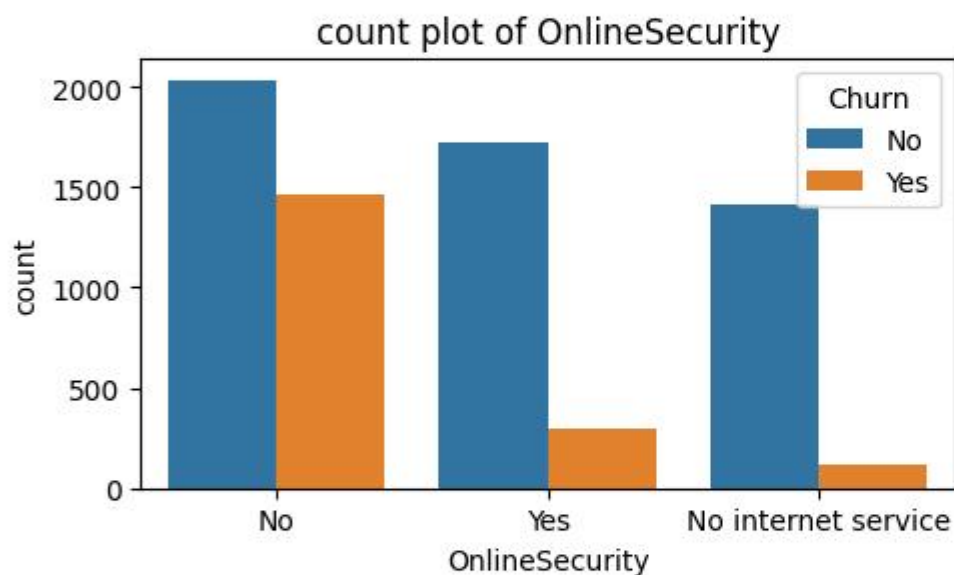
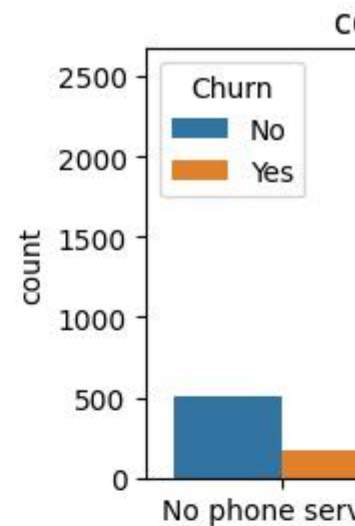
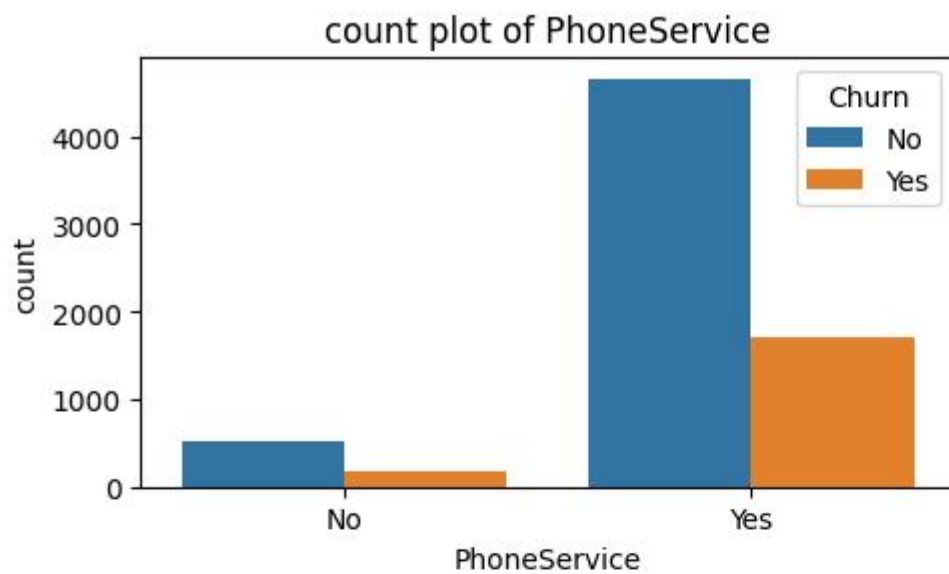


In the above graph we can clearly see total contracts of people on monthly to yearly basis mostly on month to month basis where people month to month to contract more people are churning out than that of year to two . # SO, try to make people in longer plans

In [65]:

```
columns = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
           'OnlineBackup', 'DeviceProtection', 'TechSupport',
           'StreamingTV', 'StreamingMovies']
# Number of subplots (3 columns per row)n_cols = 3n_rows = (len(columns) + n_cols
- 1) // num_cols
fig,axes = plt.subplots(n_rows,n_cols ,figsize=(15, num_rows * 3))
axes = axes.flatten()
for i ,col in enumerate(columns):
    sns.countplot(x=col , data = df , ax= axes[i] , hue = df["Churn"])
    axes[i].set_title(f'count plot of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel(f'count')

for j in range(i+1,len(axes)):
    fig.delaxes(axes[j])
# Adjust layoutplt.tight_layout()plt.show()
```



majority customs have not churn out in DSL but for internet services , phone services with online services enableed where as customer who churned out are online support , techsupport and streaming services

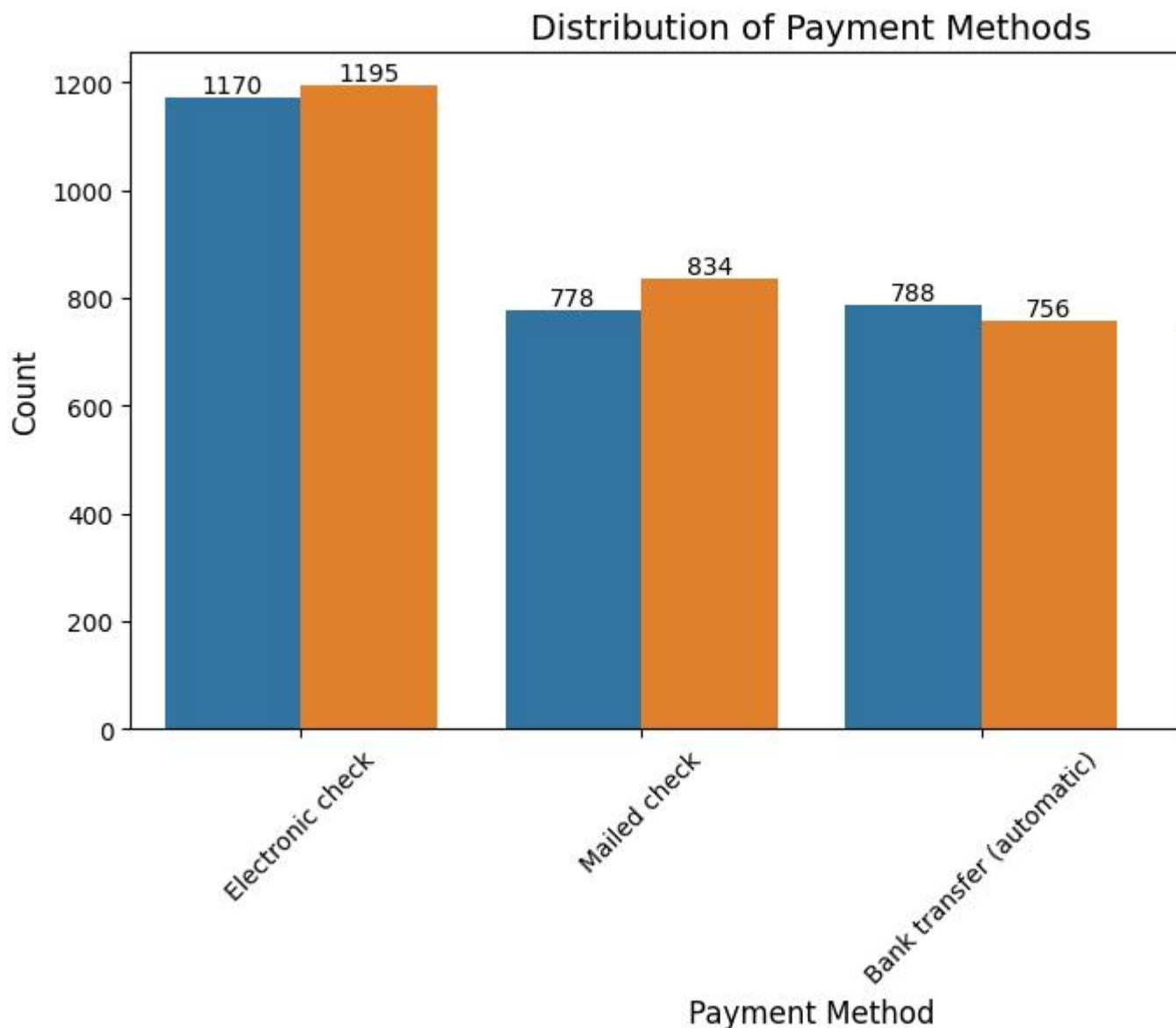
In [96]:

```
plt.figure(figsize=(10, 5))
```

```
p = sns.countplot(x='PaymentMethod', data=df , hue = 'gender')
```

```
for bar in p.containers:
```

```
    p.bar_label(bar)plt.xlabel('Payment Method', fontsize=12)plt.ylabel('Count',  
    fontsize=12)plt.title('Distribution of Payment Methods',  
    fontsize=14)plt.xticks(rotation=45)plt.show()
```



most of payment is done in electronic check and done by male and most payment are done by male than female

In [45]:

```
df.to_csv('new churn.csv')
```