Customer Churn Analysis

In [2]:
```
#import the required libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:
```
telco_base_data = pd.read_csv(r'F:\NCPL\Project\Python\WA_Fn-UseC_-Telco-Customer-Churn.csv')
```

In [4]:
```
telco_base_data.head()
```

Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contr
Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Mor mo
No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One y
No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Mor mo
No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One y
No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Mor mo

rows and cols

In [5]:
```
telco_base_data.shape
```

Out[5]:
```
(7043, 21)
```

In [6]:
```
telco_base_data.columns.values
```

Out[6]:
```
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 [7]:
```
# Checking the data types of all the columns
telco_base_data.dtypes
```

Out[7]:
```
customerID           object
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure                int64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
TotalCharges         object
Churn                object
dtype: object
```

In [8]:
```
#  Descriptive statistics of numeric variables
telco_base_data.describe()
```

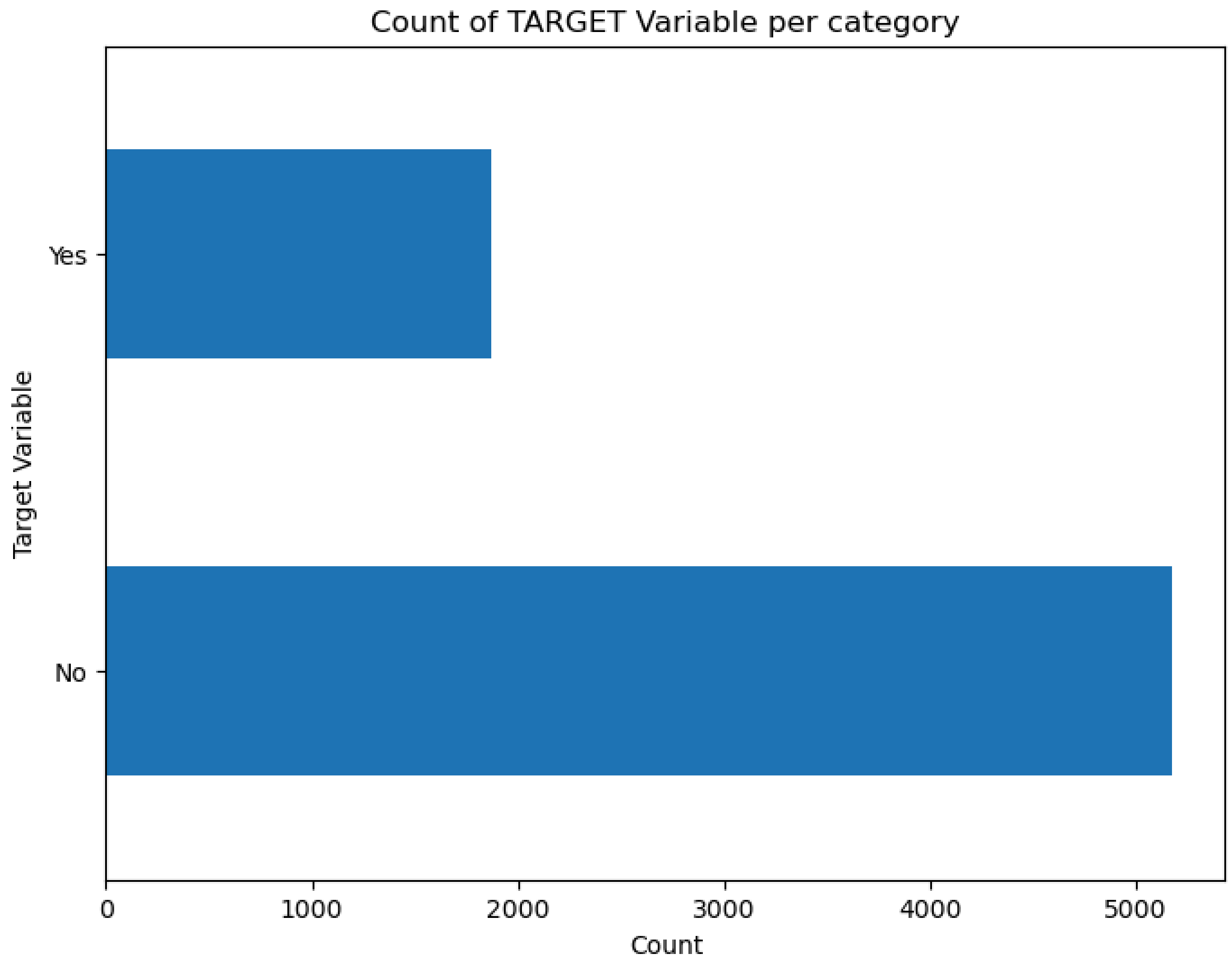
Out[8]:

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

75% of customers have tenure of 55 months

Average Monthly charges are USD 64.76 whereas 75% customers pay USD 89.85 per month

In [9]:
```
telco_base_data['Churn'].value_counts().plot(kind='barh', figsize=(8, 6))
plt.xlabel("Count")
plt.ylabel("Target Variable")
plt.title("Count of TARGET Variable per category");
```



In [10]:
```
100*telco_base_data['Churn'].value_counts()/len(telco_base_data['Churn'])
```

Out[10]:
```
No     73.463013
Yes    26.536987
Name: Churn, dtype: float64
```

In [11]:
```
telco_base_data['Churn'].value_counts()
```

Out[11]:
```
No     5174
Yes    1869
Name: Churn, dtype: int64
```

Data is highly imbalanced, ratio = 73:27

So analysing the data with other features while taking the target values to get some insights.

In [12]:
```
# Summary to check null values
telco_base_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
```

Data Cleaning

1. copying of base data for manupulation & processing

In [13]:
```
telco_data = telco_base_data.copy()
```

2. Total Charges should be numeric amount so converting it to numerical data type

In [14]:
```
telco_data.TotalCharges = pd.to_numeric(telco_data.TotalCharges, errors='coerce')
telco_data.isnull().sum()
```

Out[14]:
```
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        11
Churn                0
dtype: int64
```

3. There are 11 missing values in TotalCharges column.

In [15]:
```
telco_data.loc[telco_data ['TotalCharges'].isnull() == True]
```

Out[15]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtectio
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Ye
753	3115-CZMZD	Male	0	No	Yes	0	Yes	No	No	No internet service	...	No interne servic
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	...	Ye
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No interne servic
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Ye
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	No	No	No internet service	...	No interne servic
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No interne servic
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	No	No	No internet service	...	No interne servic
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	No	No	No internet service	...	No interne servic
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	...	Ye
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	...	N

11 rows × 21 columns

In [16]:

```
#Removing missing values
telco_data.dropna(how = 'any', inplace = True)

#telco_data.fillna(0)
```

5. Divide customers into bins based on tenure

In [17]:
```
#  maximum tenure
print(telco_data['tenure'].max())
```

```
72
```

In [18]:
```
#  tenure in bins of 12 months
labels = ["{0} - {1}".format(i, i + 11) for i in range(1, 72, 12)]

telco_data['tenure_group'] = pd.cut(telco_data.tenure, range(1, 80, 12), right=False, labels=labels)
```

In [19]:
```
telco_data['tenure_group'].value_counts()
```

Out[19]:
```
1 - 12     2175
61 - 72    1407
13 - 24    1024
25 - 36     832
49 - 60     832
37 - 48     762
Name: tenure_group, dtype: int64
```

6. Removed columns which are not not required

In [20]:
```
#drop column customerID and tenure
telco_data.drop(columns= ['customerID','tenure'], axis=1, inplace=True)
telco_data.head()
```

Out[20]:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	Female	0	Yes	No	No	No phone service	DSL	No	Yes	No	N
1	Male	0	No	No	Yes	No	DSL	Yes	No	Yes	N
2	Male	0	No	No	Yes	No	DSL	Yes	Yes	No	N
3	Male	0	No	No	No	No phone service	DSL	Yes	No	Yes	Y
4	Female	0	No	No	Yes	No	Fiber optic	No	No	No	N

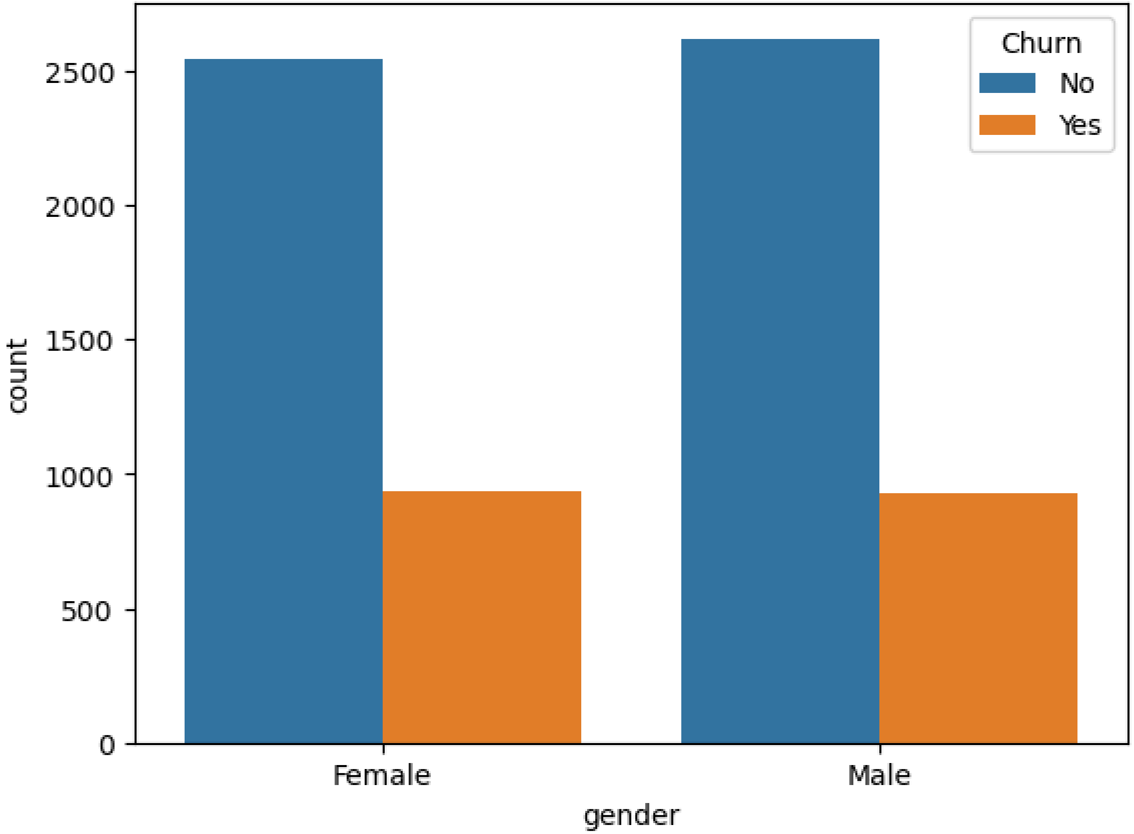
Data Exploration

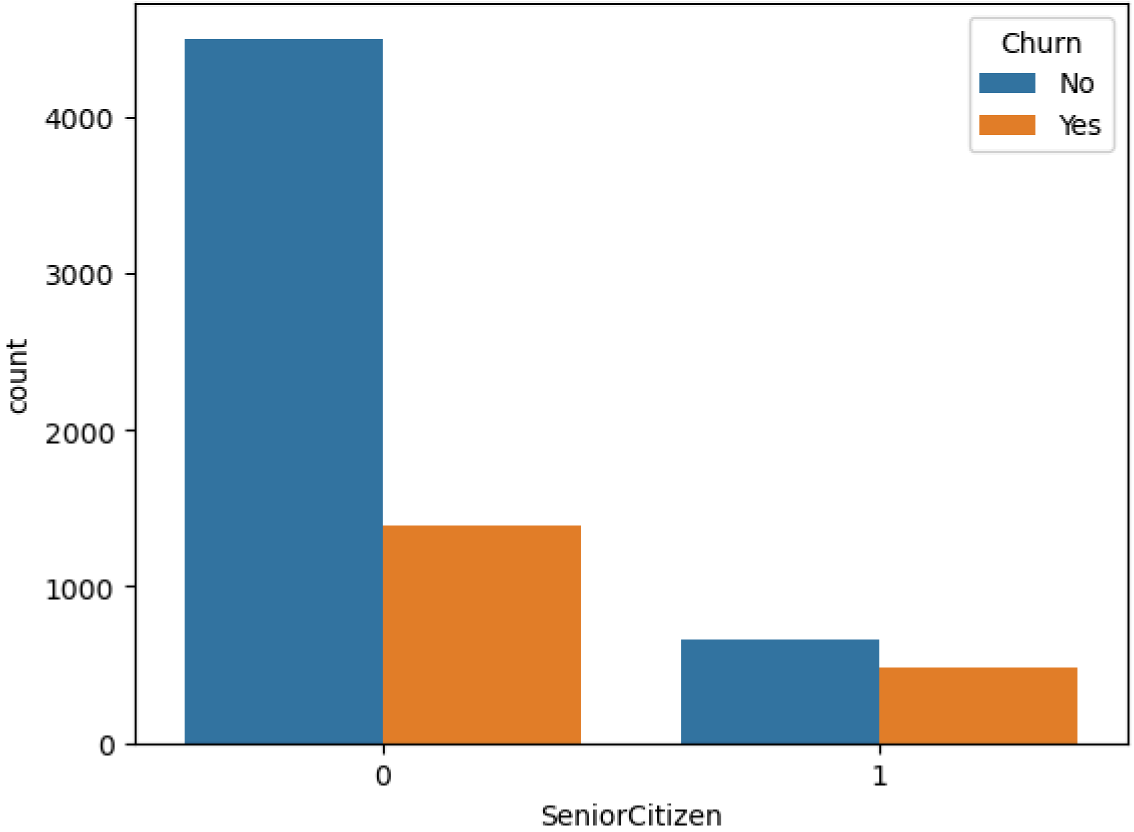
Plot distibution of individual predictors by churn

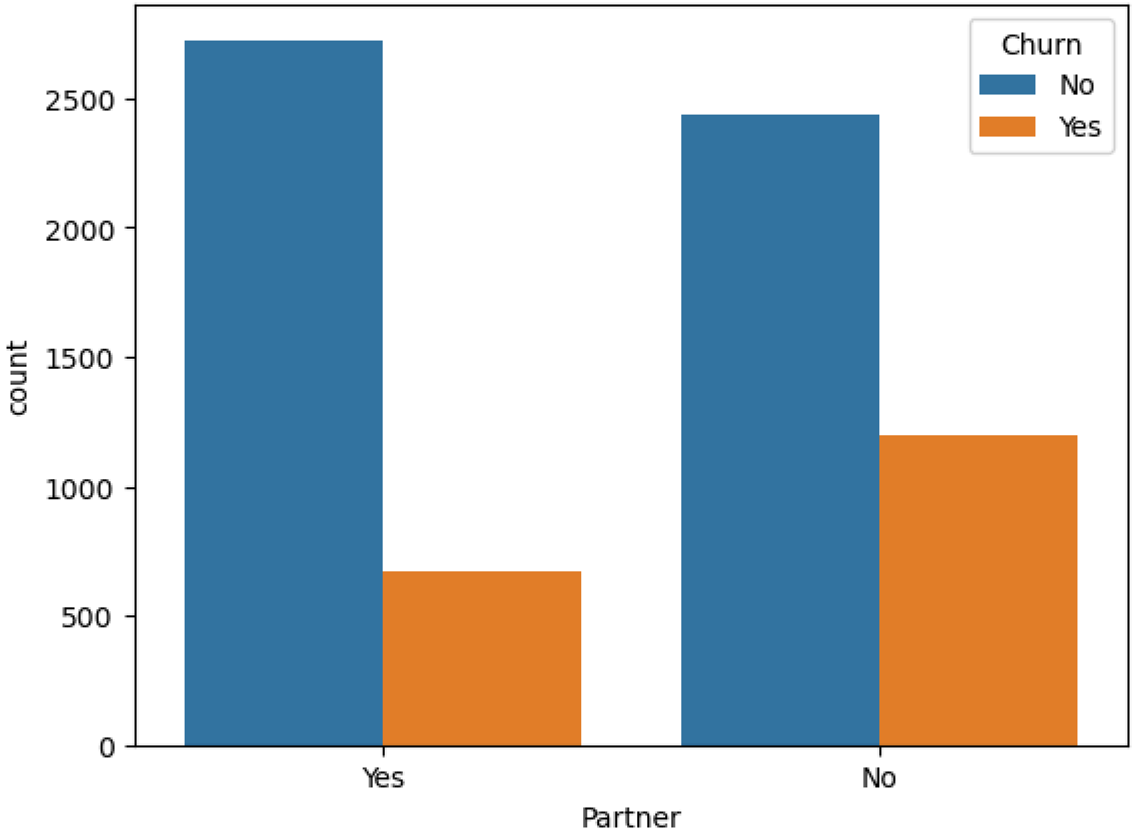
Univariate Analysis

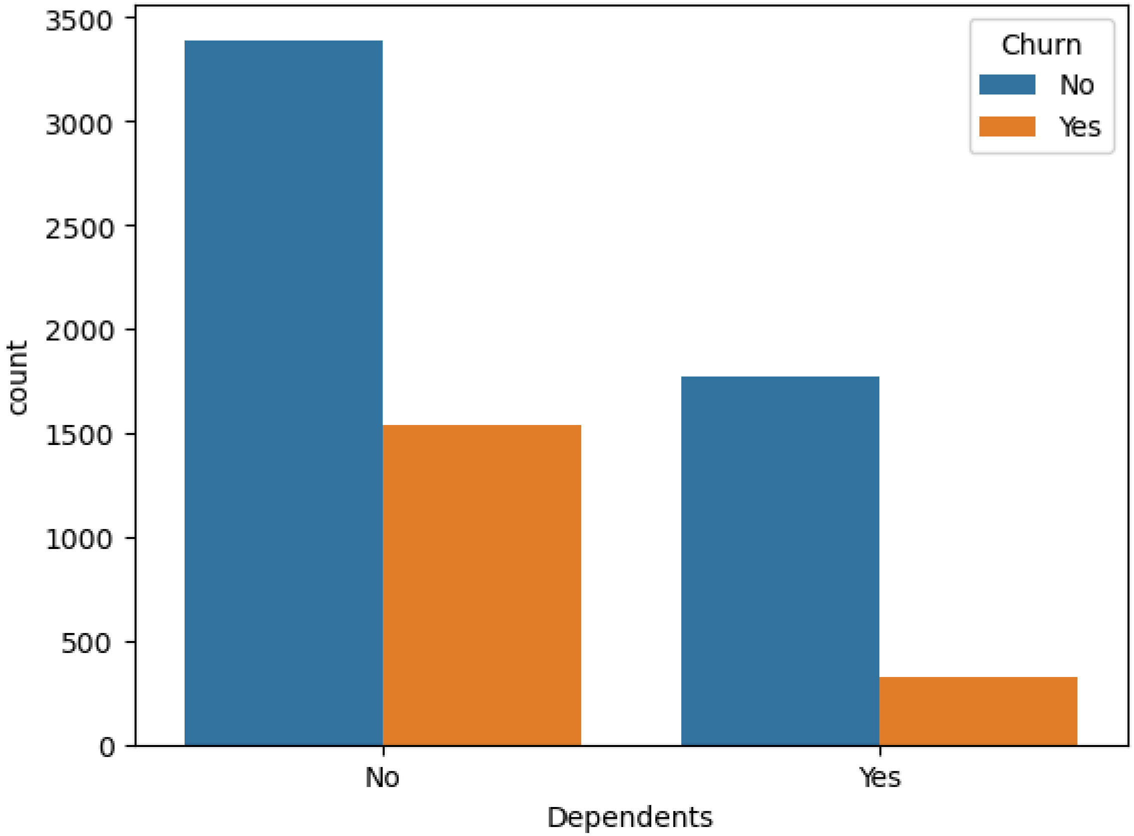
In [21]:

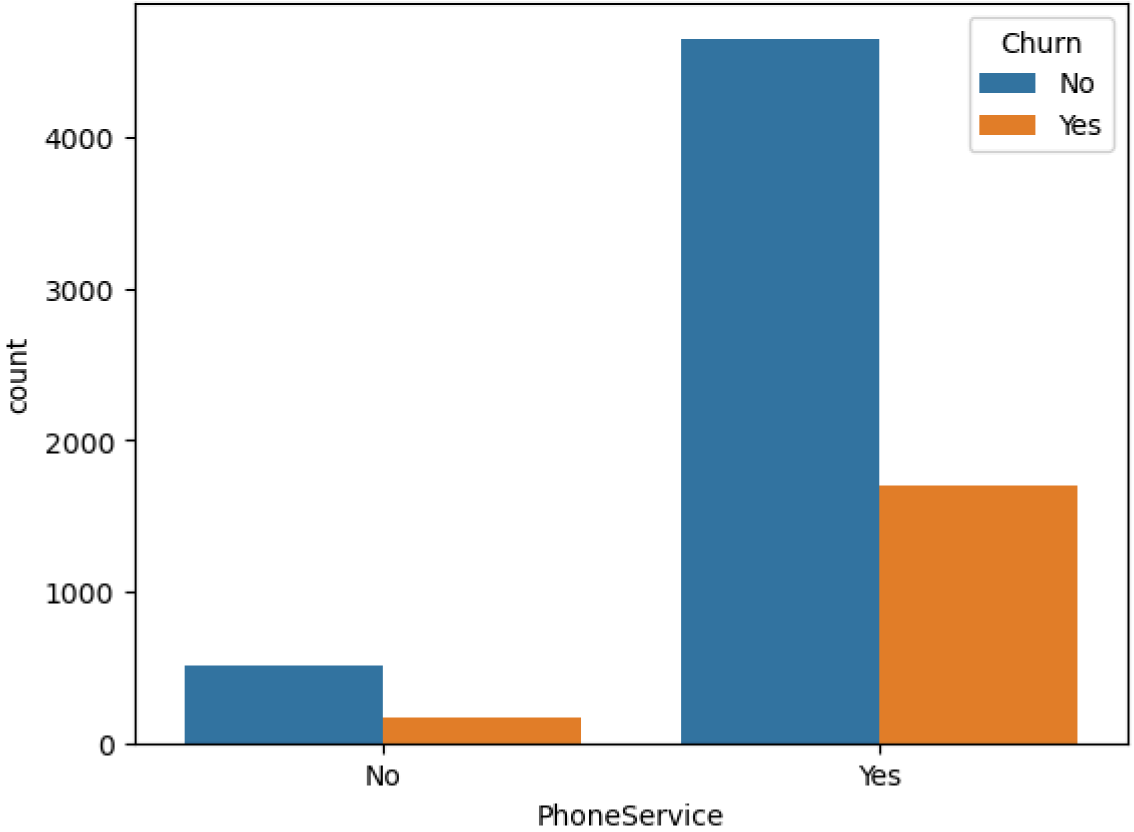
```
for i, predictor in enumerate(telco_data.drop(columns=['Churn', 'TotalCharges', 'MonthlyCharges'])):
    plt.figure(i)
    sns.countplot(data=telco_data, x=predictor, hue='Churn')
```

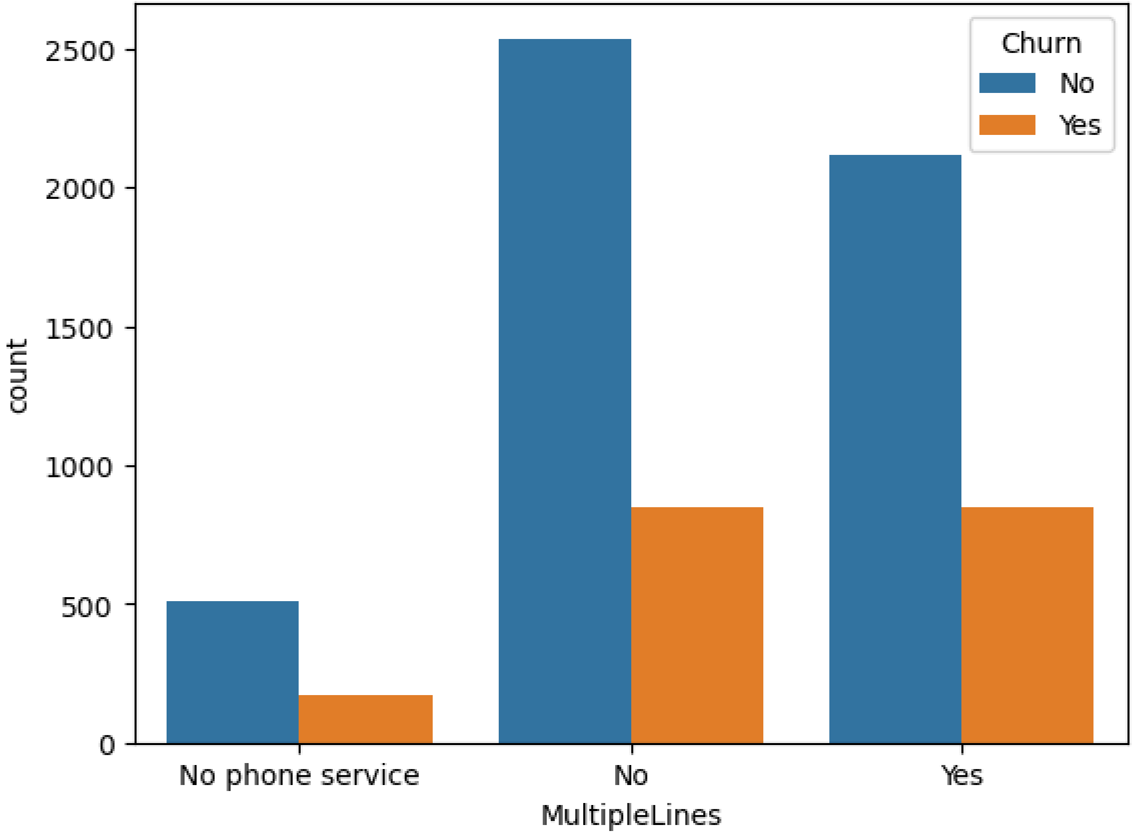

Churn
No
Yes
count
2500
2000
1500
1000
500
0
Female
Male
gender

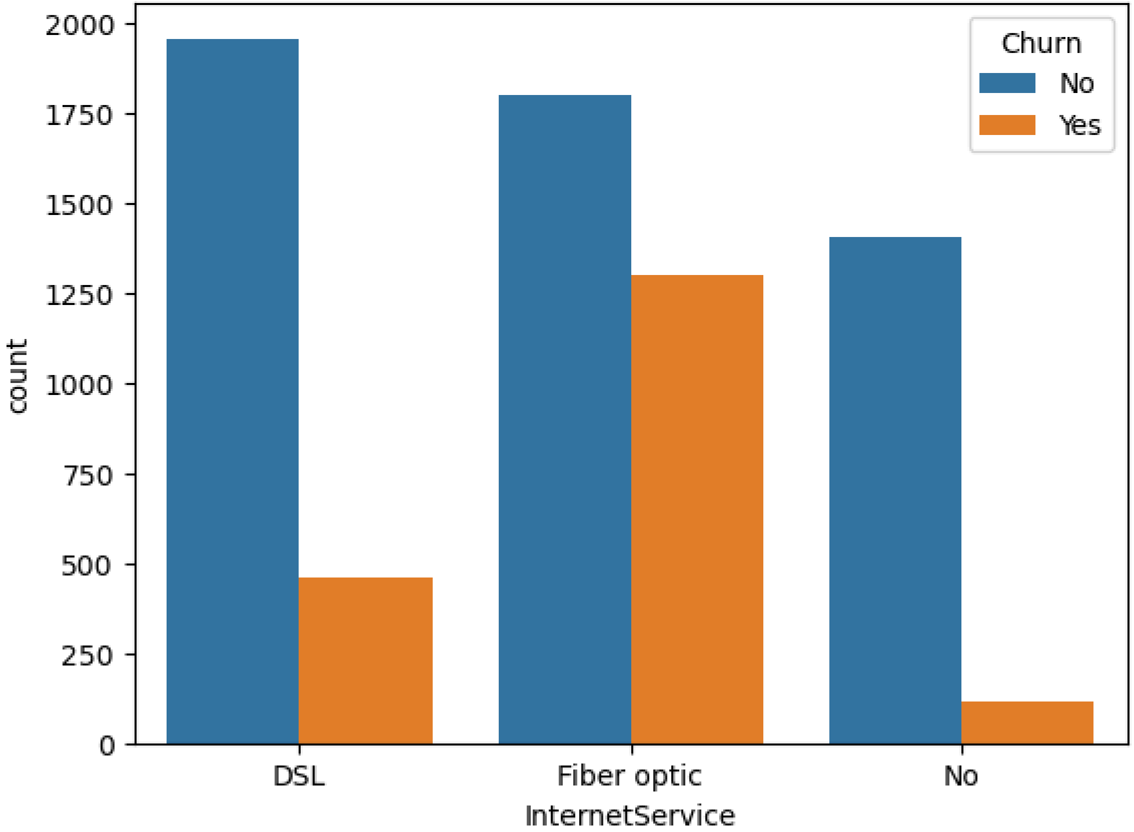

Churn
No
Yes
count
SeniorCitizen
0
1000
2000
3000
4000
0
1

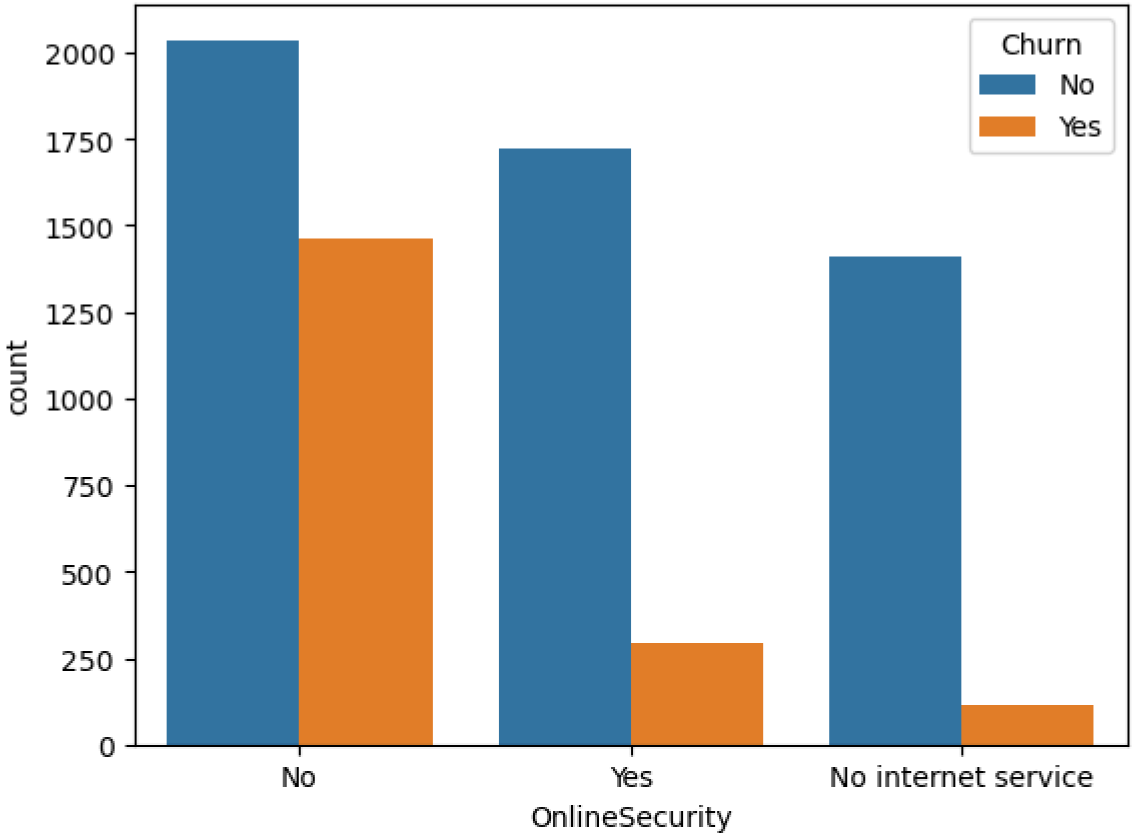

Churn
No
Yes
2500
2000
1500
1000
500
0
count
Yes
No
Partner

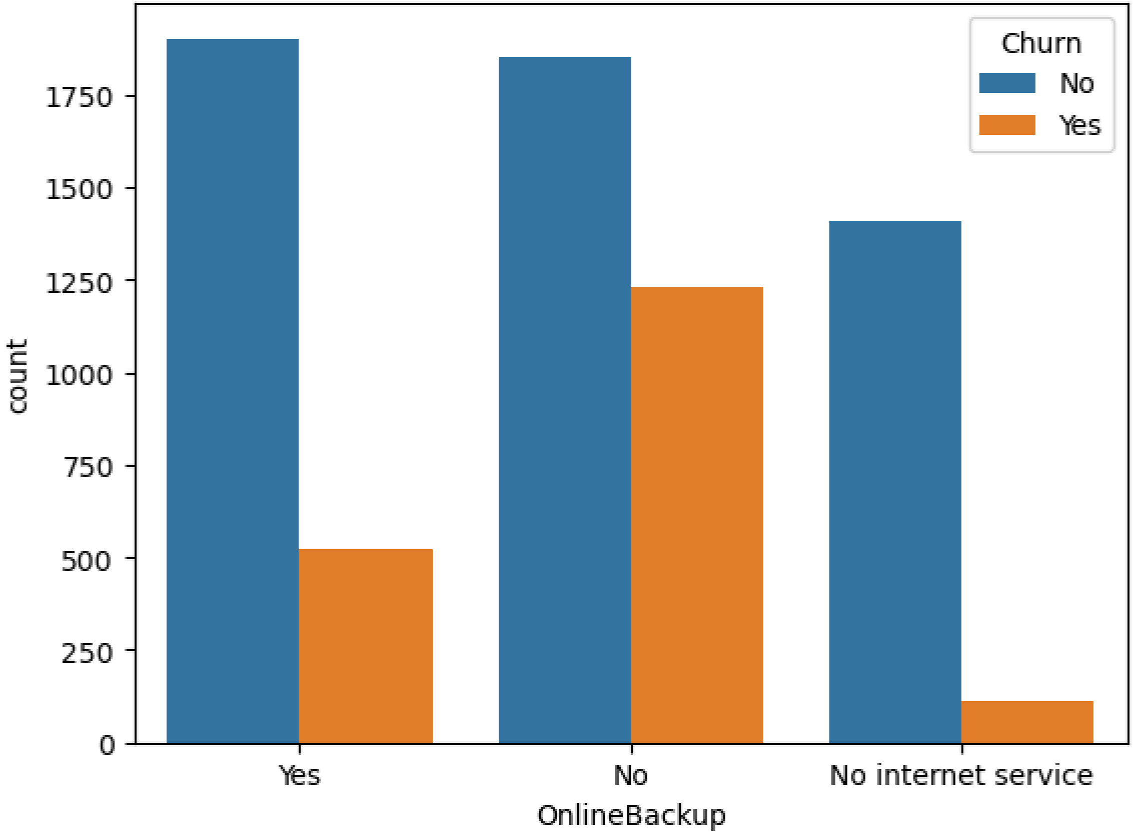

Churn
No
Yes
count
3500
3000
2500
2000
1500
1000
500
0
No
Yes
Dependents

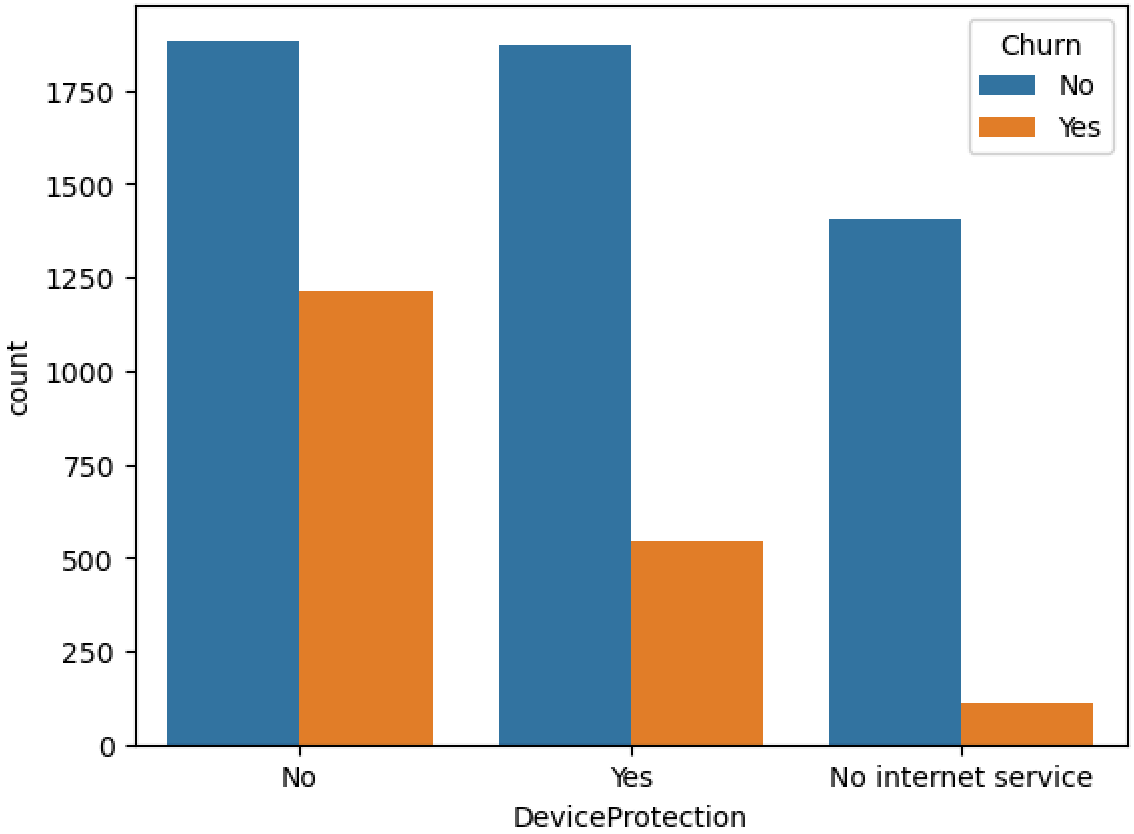

Churn
No
Yes
4000
3000
2000
1000
0
count
No
Yes
PhoneService

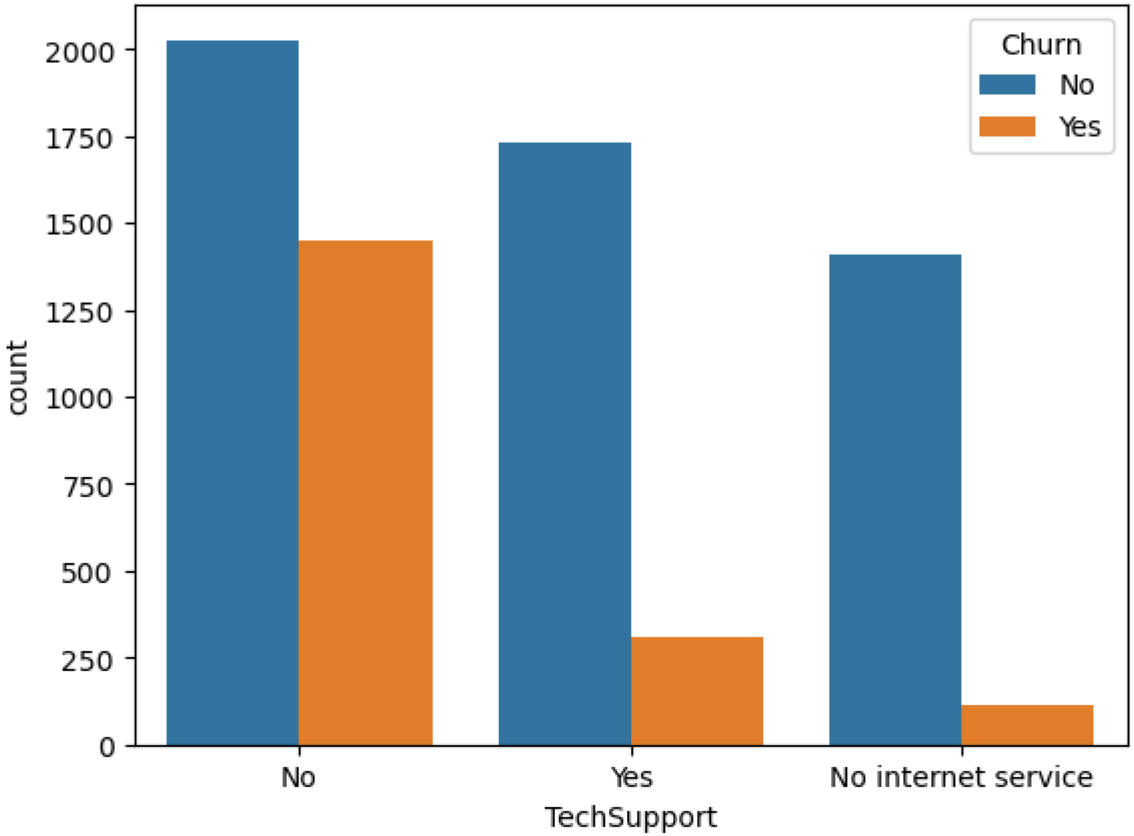

Churn
No
Yes
count
2500
2000
1500
1000
500
0
No phone service
No
Yes
MultipleLines

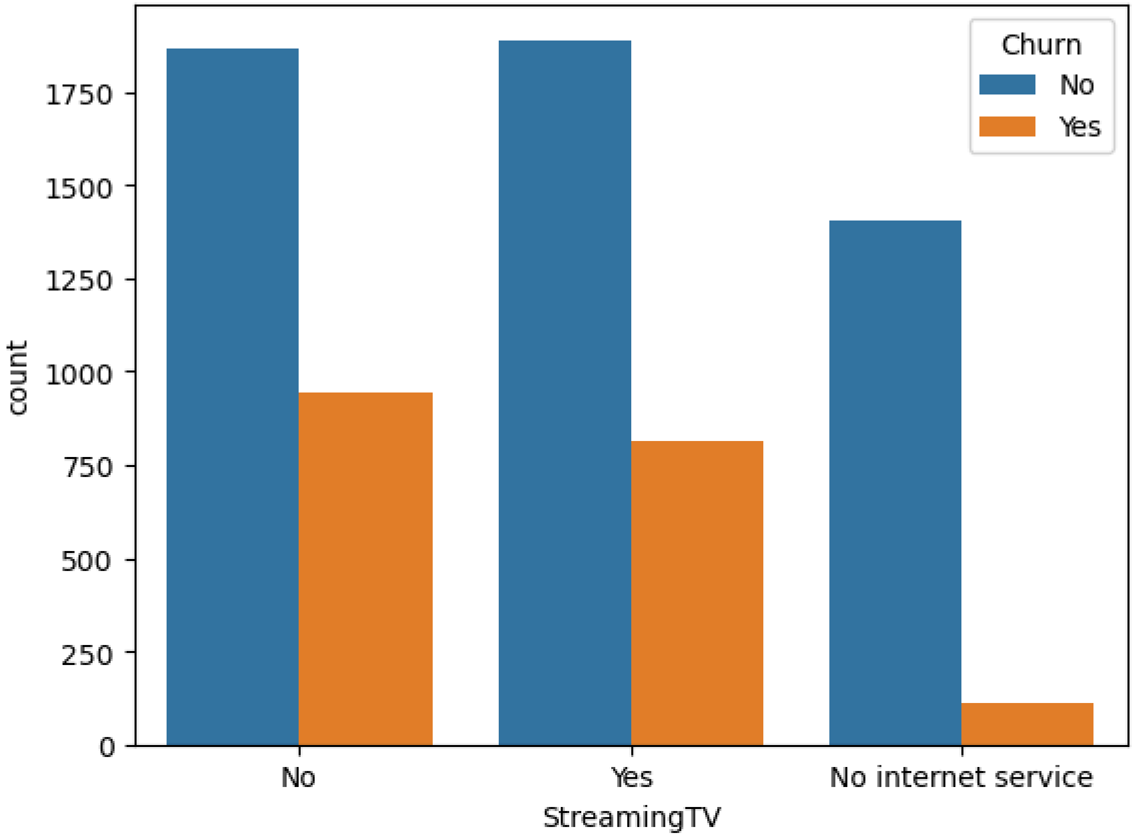

Churn
No
Yes
count
2000
1750
1500
1250
1000
750
500
250
0
DSL
Fiber optic
No
InternetService

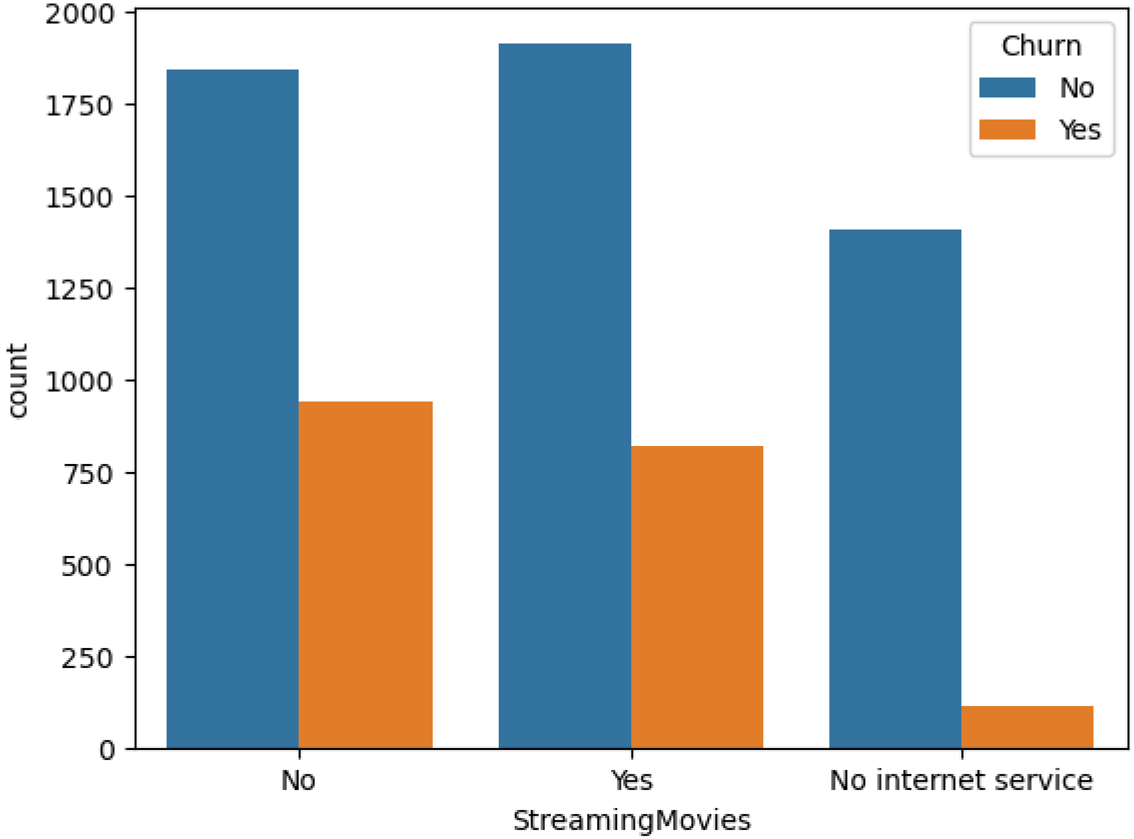

Churn
No
Yes
count
2000
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
OnlineSecurity

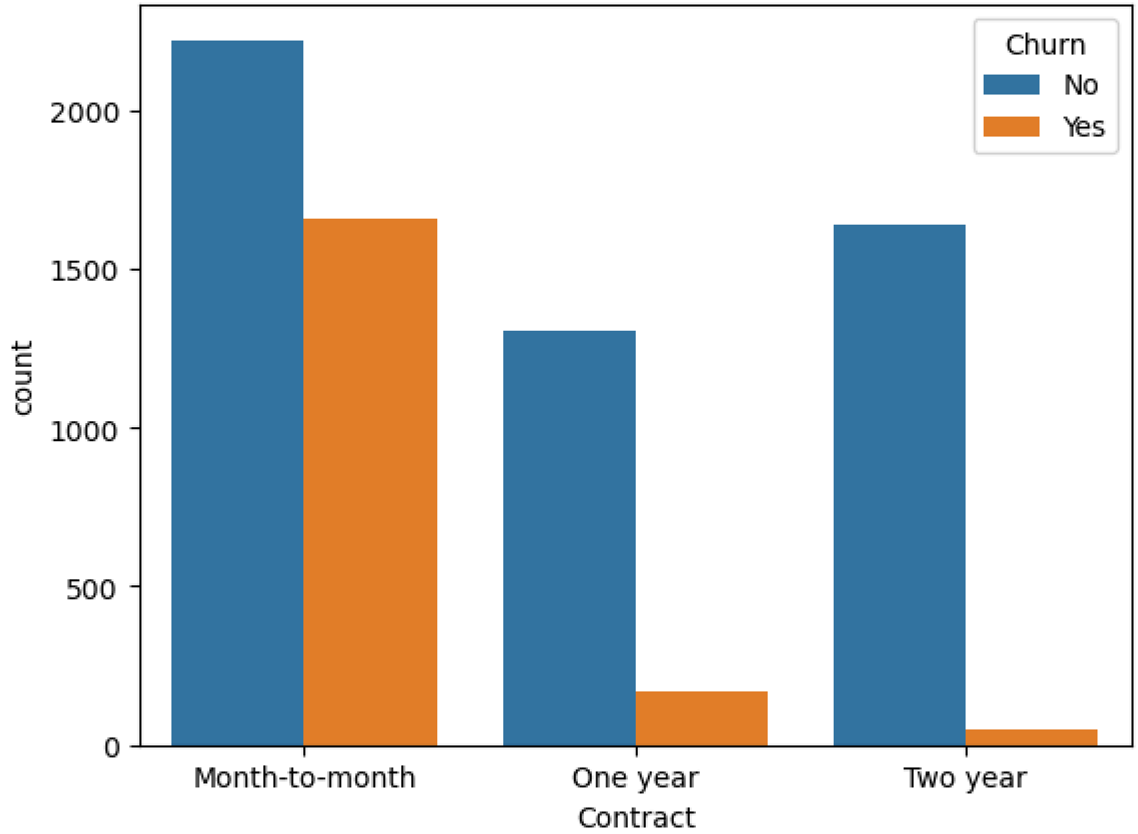

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
Yes
No
No internet service
OnlineBackup

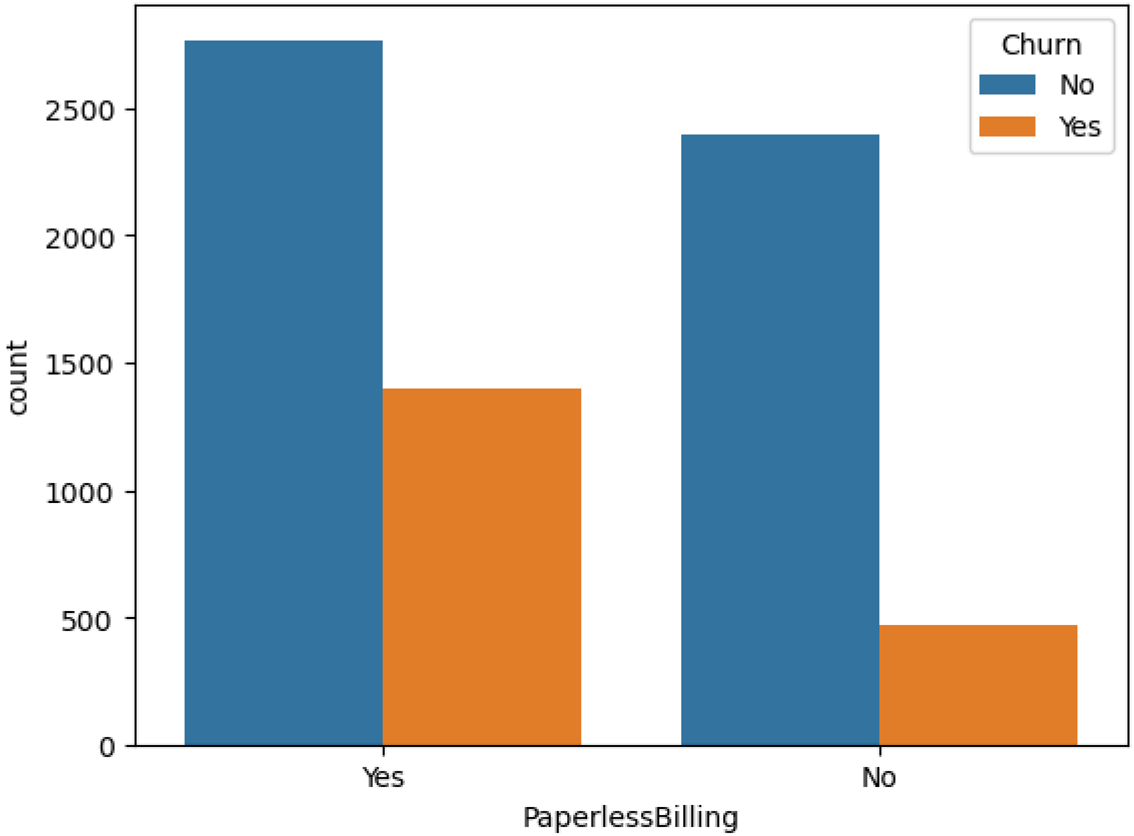

Churn
No
Yes
count
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
DeviceProtection

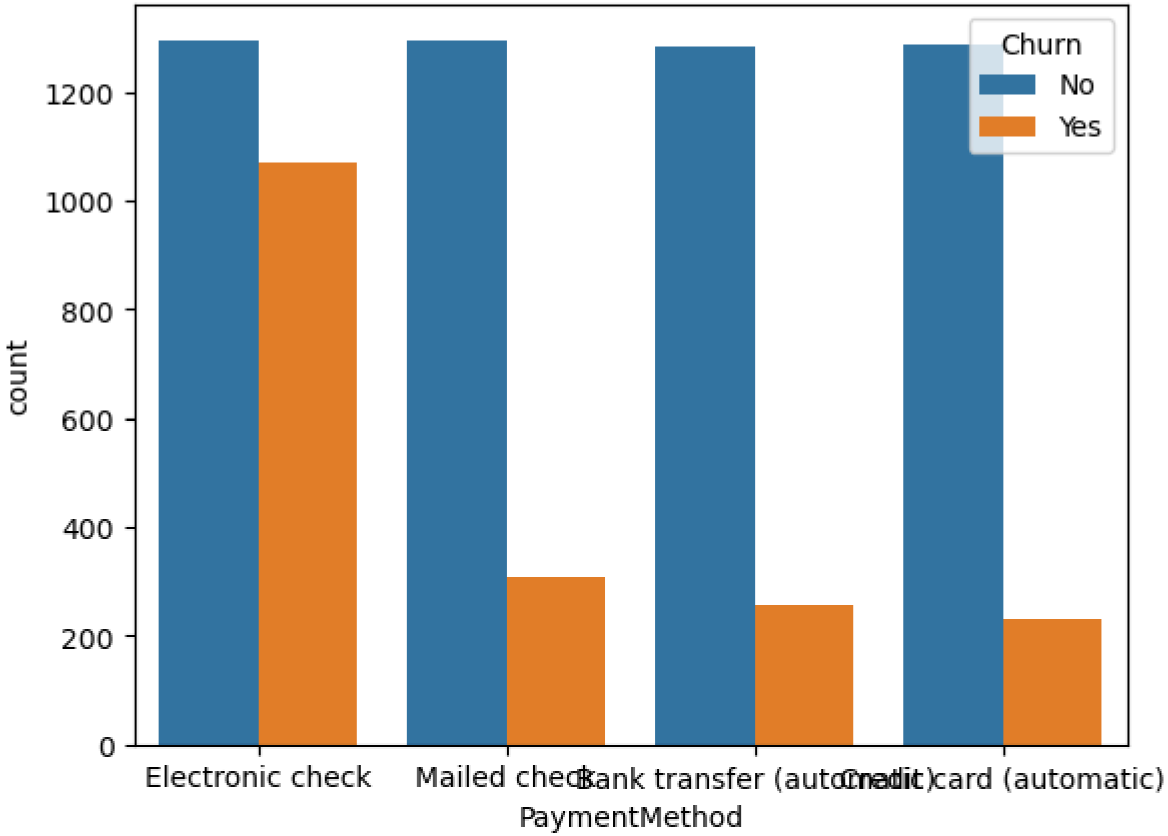

Churn
No
Yes
count
2000
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
TechSupport

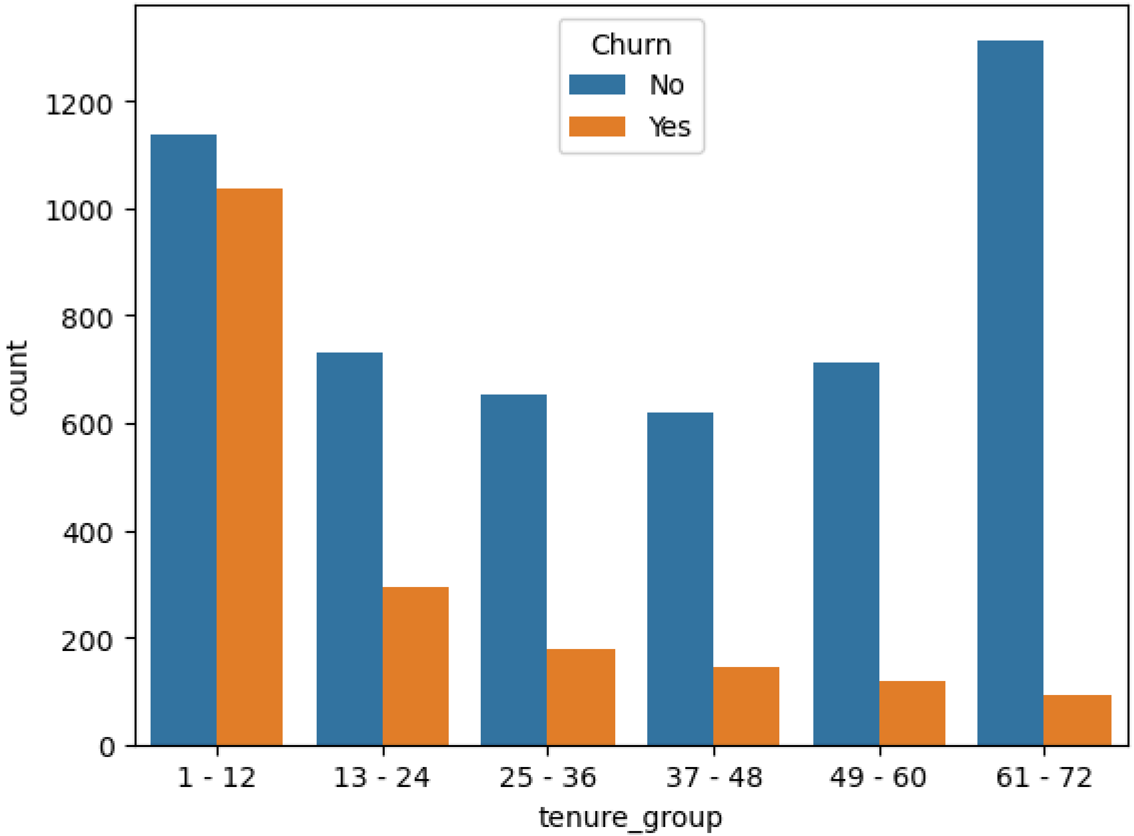

Churn
No
Yes
count
0
250
500
750
1000
1250
1500
1750
No
Yes
No internet service
StreamingTV


Churn
No
Yes
count
2000
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
StreamingMovies


Churn
No
Yes
2000
1500
1000
500
0
count
Month-to-month
One year
Two year
Contract


count
2500
2000
1500
1000
500
0
Yes
No
PaperlessBilling
Churn
No
Yes


Churn
No
Yes
count
0
200
400
600
800
1000
1200
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod



Convering the target variable 'Churn' in a binary numeric variable i.e. Yes=1 ; No = 0

In [22]:
```
telco_data['Churn'] = np.where(telco_data.Churn == 'Yes',1,0)
```

In [23]:
```
telco_data.head()
```

Out[23]:

	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	Female	0	Yes	No	No	No phone service	DSL	No	Yes	No	N
1	Male	0	No	No	Yes	No	DSL	Yes	No	Yes	N
2	Male	0	No	No	Yes	No	DSL	Yes	Yes	No	N
3	Male	0	No	No	No	No phone service	DSL	Yes	No	Yes	Y
4	Female	0	No	No	Yes	No	Fiber optic	No	No	No	N

3. Converting all the categorical variables into dummy variables

In [24]:
```
telco_data_dummies = pd.get_dummies(telco_data)    # One Hot Encoding
telco_data_dummies.head()
```

Out[24]:

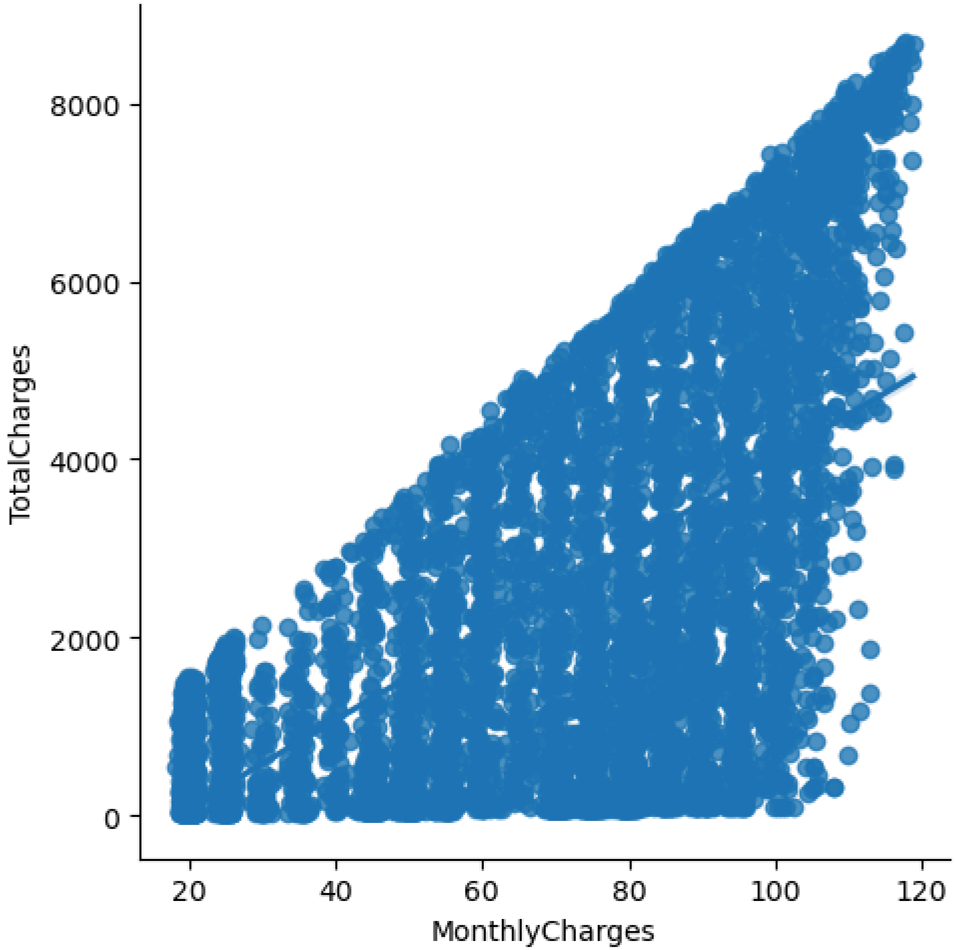
	SeniorCitizen	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	Dependents_Yes	...	Pay t
0	0	29.85	29.85	0	1	0	0	1	1	0	...	
1	0	56.95	1889.50	0	0	1	1	0	1	0	...	
2	0	53.85	108.15	1	0	1	1	0	1	0	...	
3	0	42.30	1840.75	0	0	1	1	0	1	0	...	
4	0	70.70	151.65	1	1	0	1	0	1	0	...	

5 rows × 51 columns

Relationship between Monthly Charges and Total Charges

In [25]:
```
sns.lmplot(data=telco_data_dummies, x='MonthlyCharges', y='TotalCharges')
```

Out[25]: <seaborn.axisgrid.FacetGrid at 0x1c619da2200>

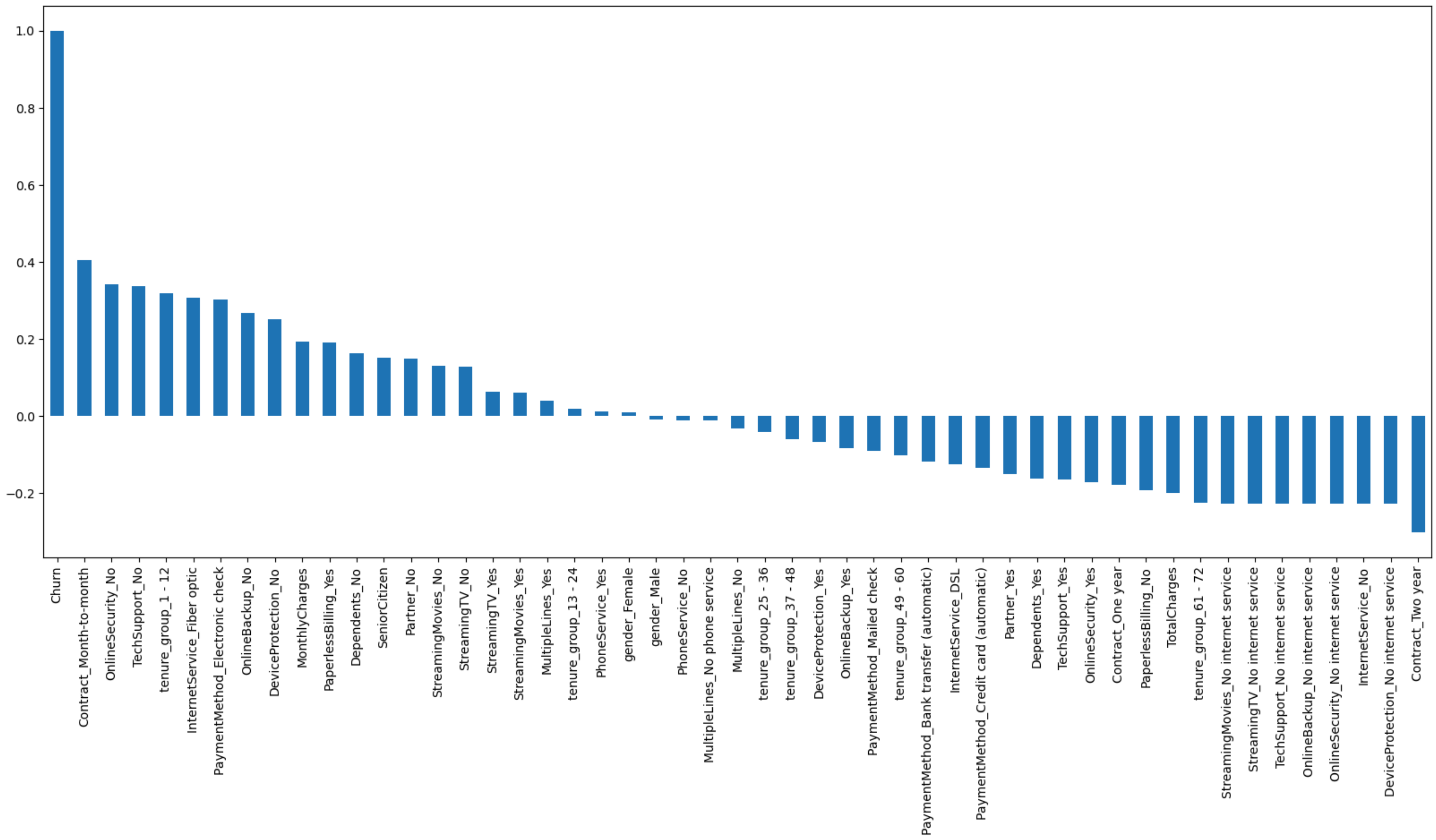


Total Charges increase as Monthly Charges increase

corelation of all predictors with 'Churn'

In [26]:
```
plt.figure(figsize=(20,8))
telco_data_dummies.corr()['Churn'].sort_values(ascending = False).plot(kind='bar')
```

Out[26]: <Axes: >



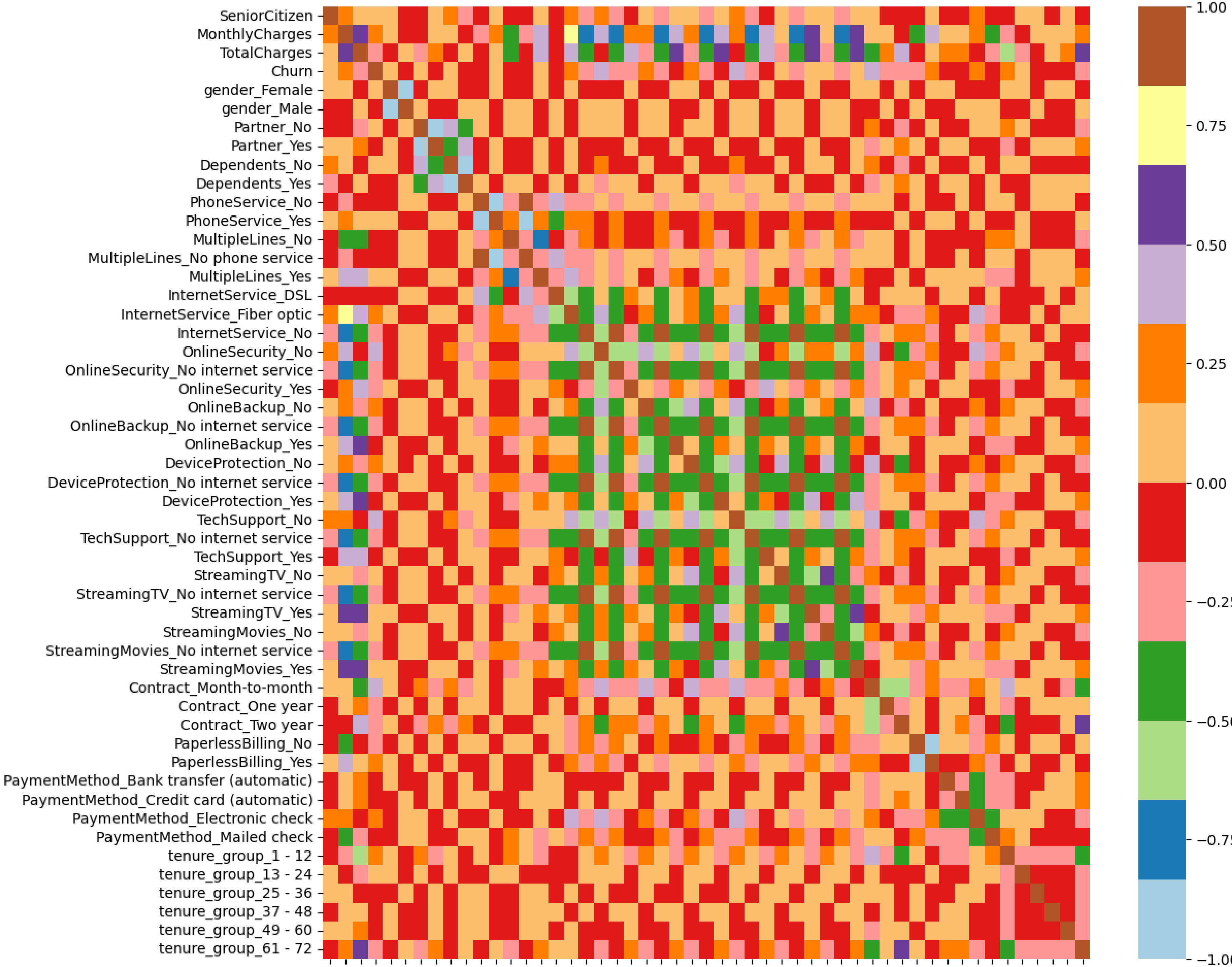
Insights:

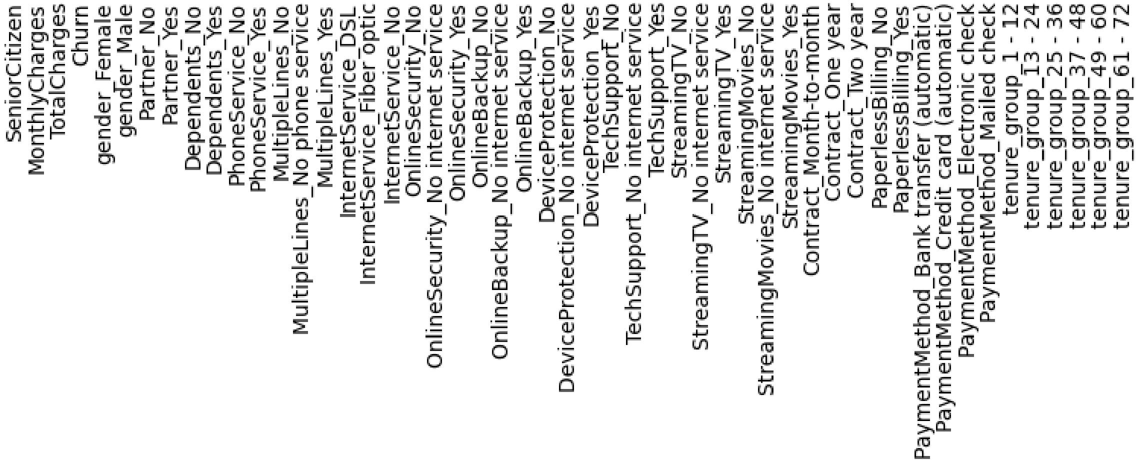
HIGH Churn seen in case of **Month to month contracts**, **No online security**, **No Tech support**, **First year of subscription** and **Fibre Optics Internet**

LOW Churn is seens in case of **Long term contracts**, **Subscriptions without internet service** and **The customers engaged for 5+ years**

In [27]:
```
plt.figure(figsize=(12,12))
sns.heatmap(telco_data_dummies.corr(),cmap="Paired")
```

Out[27]:
```
<Axes: >
```


SeniorCitizen
MonthlyCharges
TotalCharges
Churn
gender_Female
gender_Male
Partner_No
Partner_Yes
Dependents_No
Dependents_Yes
PhoneService_No
PhoneService_Yes
MultipleLines_No
MultipleLines_No phone service
MultipleLines_Yes
InternetService_DSL
InternetService_Fiber optic
InternetService_No
OnlineSecurity_No
OnlineSecurity_No internet service
OnlineSecurity_Yes
OnlineBackup_No
OnlineBackup_No internet service
OnlineBackup_Yes
DeviceProtection_No
DeviceProtection_No internet service
DeviceProtection_Yes
TechSupport_No
TechSupport_No internet service
TechSupport_Yes
StreamingTV_No
StreamingTV_No internet service
StreamingTV_Yes
StreamingMovies_No
StreamingMovies_No internet service
StreamingMovies_Yes
Contract_Month-to-month
Contract_One year
Contract_Two year
PaperlessBilling_No
PaperlessBilling_Yes
PaymentMethod_Bank transfer (automatic)
PaymentMethod_Credit card (automatic)
PaymentMethod_Electronic check
PaymentMethod_Mailed check
tenure_group_1 - 12
tenure_group_13 - 24
tenure_group_25 - 36
tenure_group_37 - 48
tenure_group_49 - 60
tenure_group_61 - 72
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

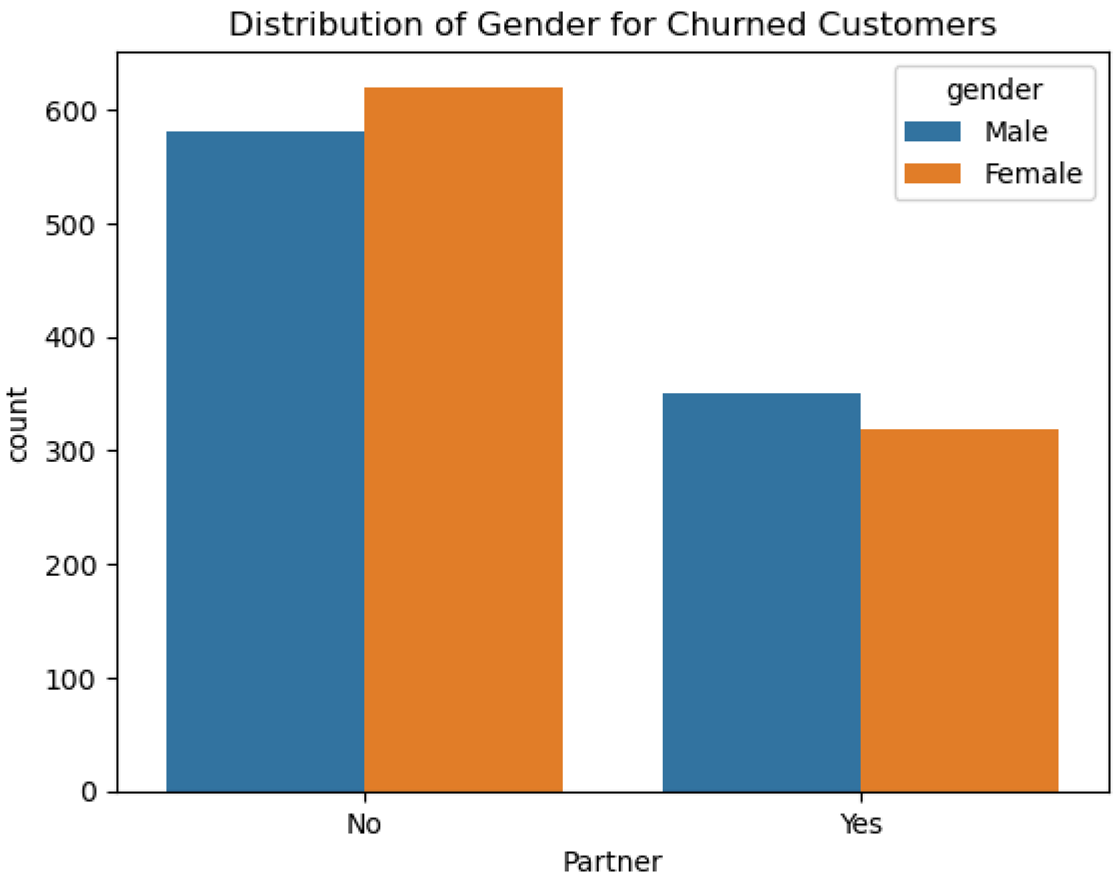


Bivariate Analysis

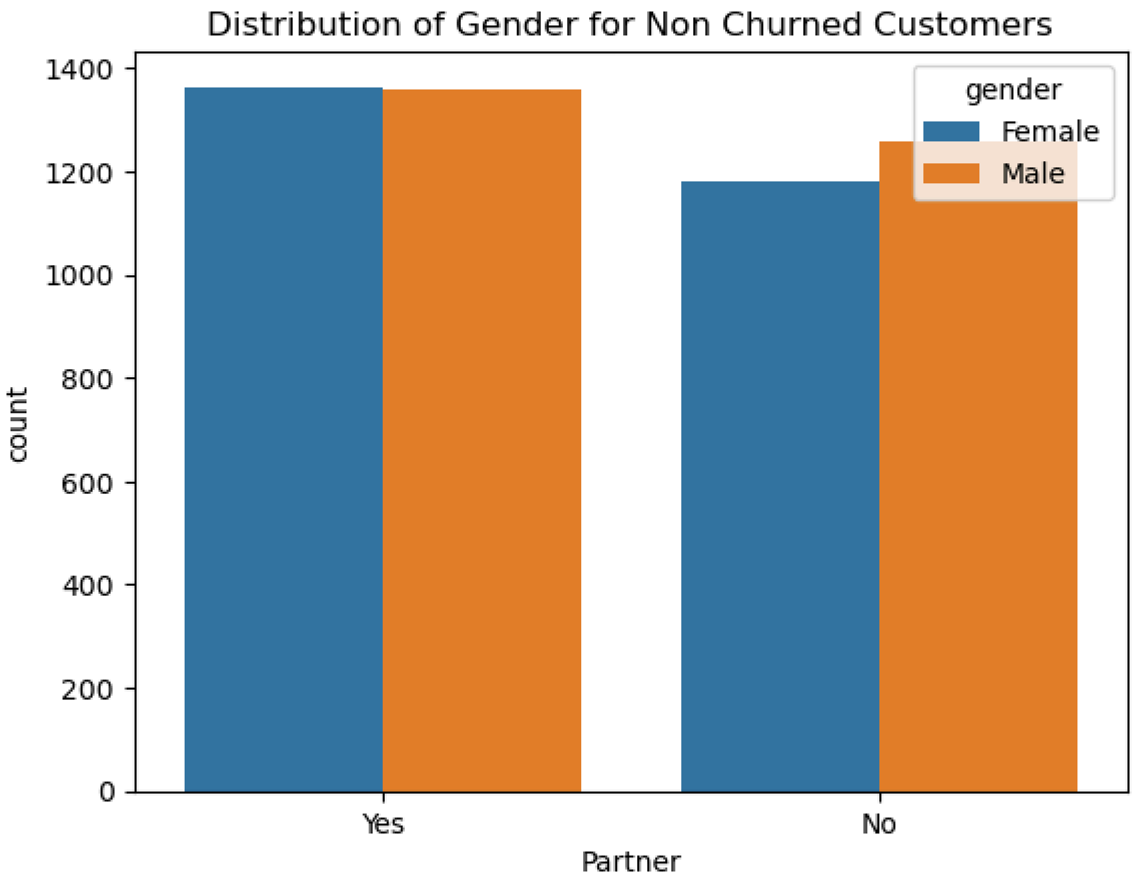
In [28]:
```
new_df1_target0=telco_data.loc[telco_data["Churn"]==0]    # Non churners
new_df1_target1=telco_data.loc[telco_data["Churn"]==1]     # Churners
```

In [29]:
```
def uniplot(df,col,title,hue =None):
    plt.title(title)
    ax = sns.countplot(data = df, x= col, order=df[col].value_counts().index,hue = hue)
    plt.show()
```

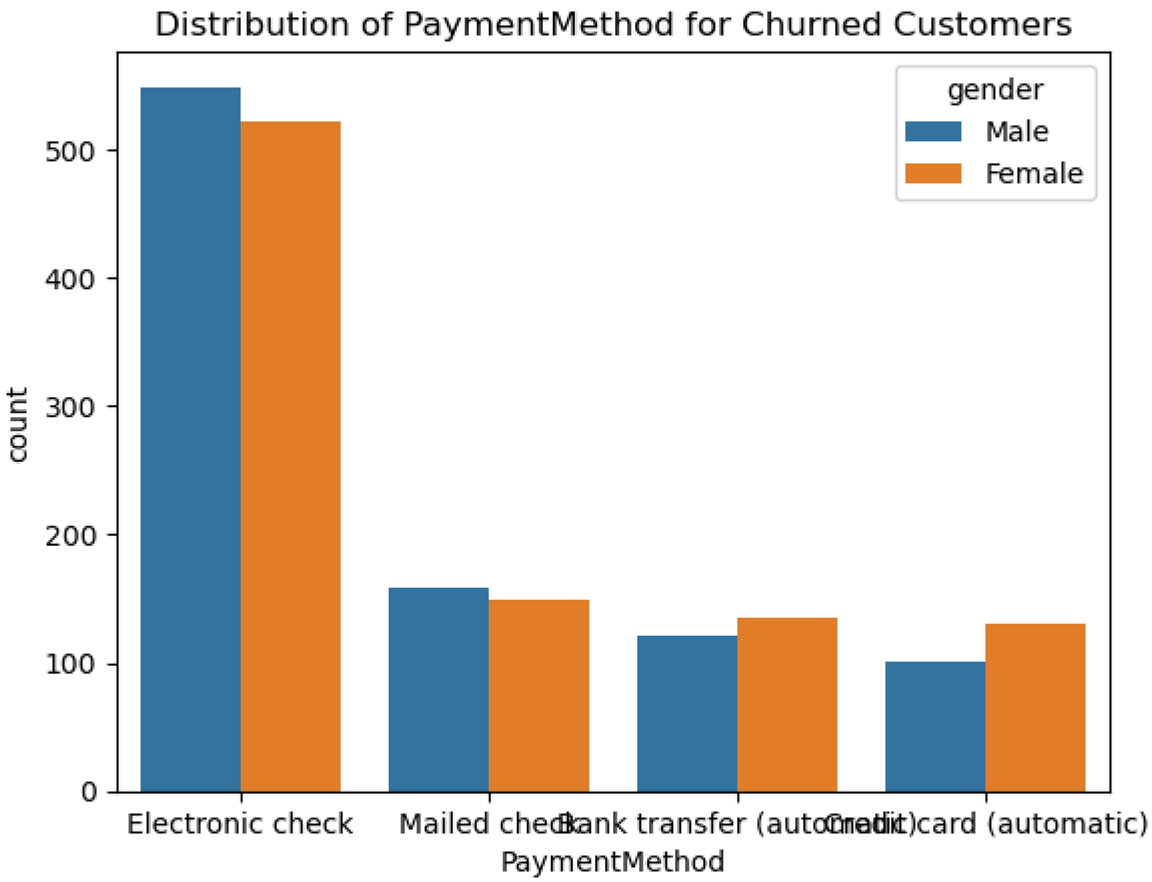
In [30]:
```
uniplot(new_df1_target1,col='Partner',title='Distribution of Gender for Churned Customers',hue='gender')
```



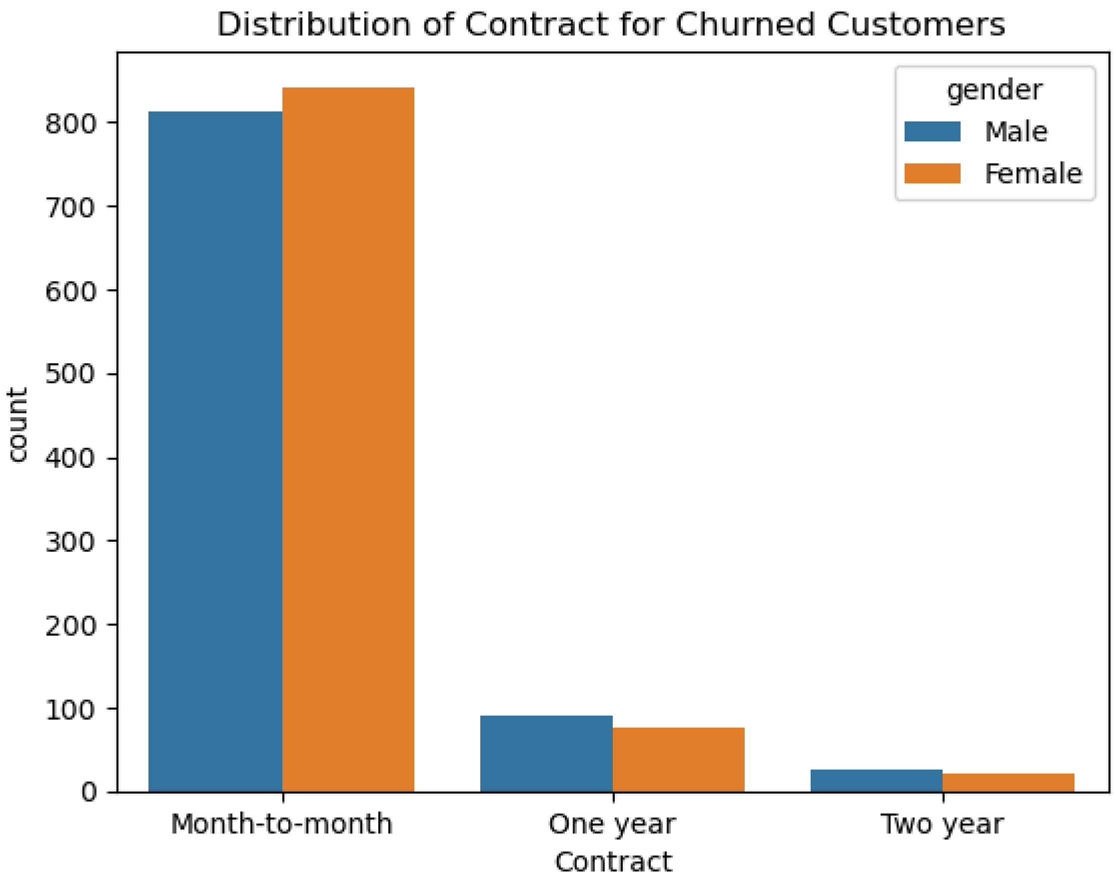
In [31]:
```
uniplot(new_df1_target0,col='Partner',title='Distribution of Gender for Non Churned Customers',hue='gender')
```



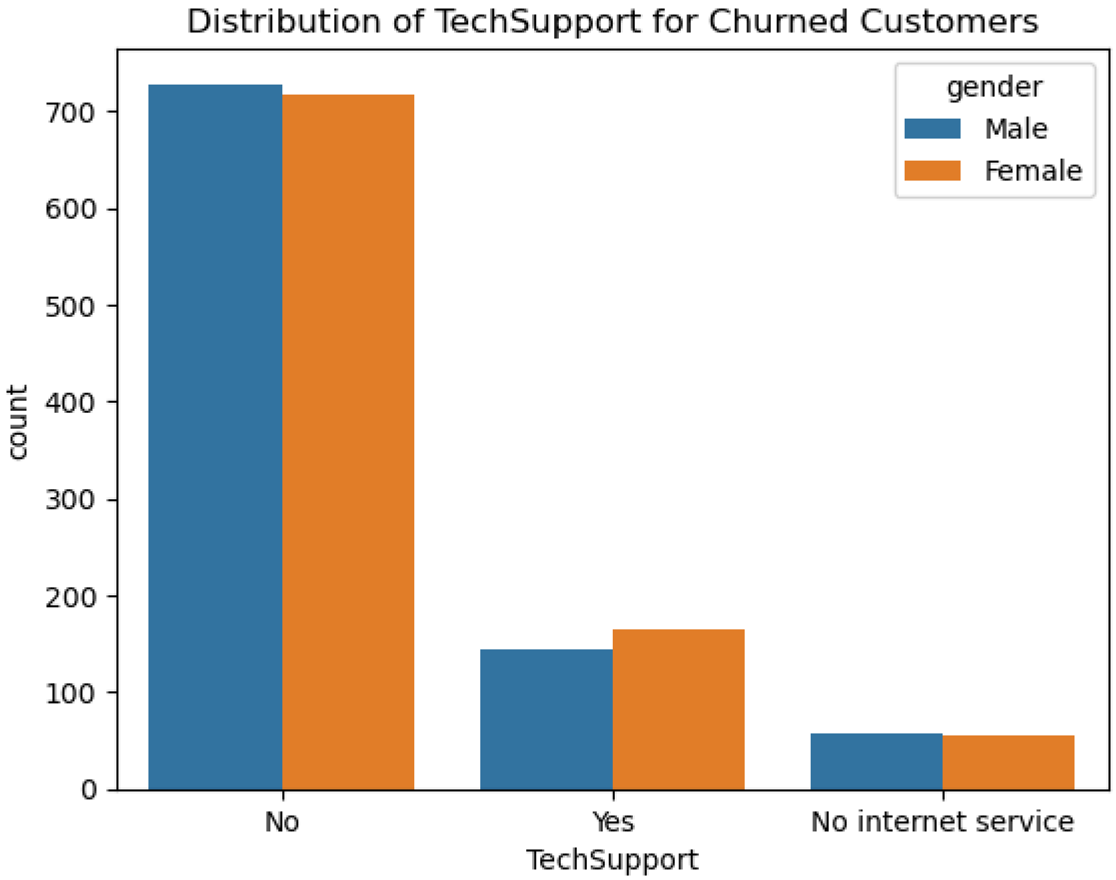
In [32]:
```
uniplot(new_df1_target1,col='PaymentMethod',title='Distribution of PaymentMethod for Churned Customers',hue='gender')
```



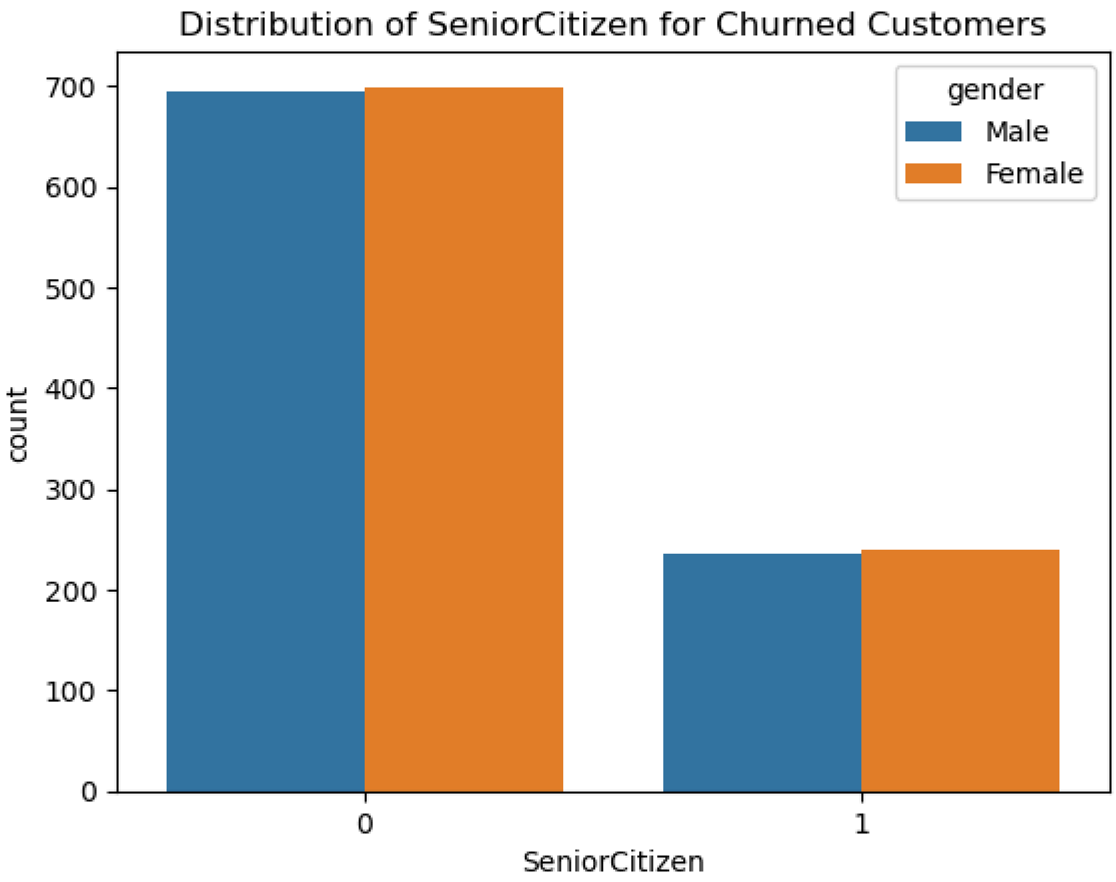
In [33]:
```
uniplot(new_df1_target1,col='Contract',title='Distribution of Contract for Churned Customers',hue='gender')
```



In [34]:
```
uniplot(new_df1_target1,col='TechSupport',title='Distribution of TechSupport for Churned Customers',hue='gender')
```



In [35]:
```
uniplot(new_df1_target1,col='SeniorCitizen',title='Distribution of SeniorCitizen for Churned Customers',hue='gender')
```



CONCLUSION

1. Electronic check medium are the highest churners
2. Contract Type - Monthly customers are more likely to churn because of no contract terms, as they are free to go customers.
3. No Online security, No Tech Support category are high churners
4. Non senior Citizens are high churners

In [37]:
```
telco_data_dummies.to_csv(r'F:\NCPL\Project\Python\tel_churn.csv')
```

In [ ]:

```
```