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

Telecom Churn Dataset

Types of churn

1) Tariff Plan Churn = Like one person use 750 plan of jio now he is using 395.

2) Service Churn = Like one person is using Yearly PLan now he start using Monthly Plan.

3) Product Churn = Now one person using Pospaid sim and now he shift to Prepaid.It will be loss for company like company generate more revenue from pospaid service charged.

4) Usage Churn = In This case person stop using Product and service.

*****Scenario and Condition of Churners

WHO DONT WANT TO CHURN

1)INERT SUBSCRIBER:

Because its too complex or i don't have time or its nor worth it.

UNCONDITIONALLY LOYAL:

2)Because my operator is the best

Who Thought about churning

1)IOCKED IN CONTRACT AND SUBSCRIBER:

I want to churn but i locked in a contract.

2)Conditionally loyal:

I want to churn but now i will wait give them some chance.

3) Conditional Churner:

Because i found a better offer.

4) LIFESTYLE MIGRATOR:

Because my needs have changed.

5) Unsatisfied Churner:

Beacause i am not satisfied with the service of the product.

Load Dataset

In [2]:
```
ch =  pd.read_csv("D:\Python\WA_Fn-UseC_-Telco-Customer-Churn.csv")
ch
```

Out[2]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLin
0	7590-VHVEG	Female	0	Yes	No	1	No	No pho servi
1	5575-GNVDE	Male	0	No	No	34	Yes	I
2	3668-QPYBK	Male	0	No	No	2	Yes	I
3	7795-CFOCW	Male	0	No	No	45	No	No pho servi
4	9237-HQITU	Female	0	No	No	2	Yes	I
...	...	...	...	...	...	...	...	
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Y
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Y
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No pho servi
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Y
7042	3186-AJIEK	Male	0	No	No	66	Yes	I

7043 rows × 21 columns

In [3]:
```
ch.describe()
```

Out[3]:

	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

In [4]:
```
top_tenure = ch['tenure'].nlargest(n=1000).index
top_tenure
```

Out[4]:
```
Int64Index([  28,   35,   59,   62,   94,  106,  109,  127,  140,  167,
            ...
            3503, 3779, 3783, 3800, 3979, 3986, 4265, 4287, 4301, 4375],
           dtype='int64', length=1000)
```

In [5]:
```
top_tenure_df = ch.iloc[top_tenure]
top_tenure_df
```

Out[5]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLin
28	5248-YGIJN	Male	0	Yes	No	72	Yes	Y
35	6234-RAAPL	Female	0	Yes	Yes	72	Yes	Y
59	5954-BDFSG	Female	0	No	No	72	Yes	Y
62	0526-SXDJP	Male	0	Yes	No	72	No	No pho servi
94	9848-JQJTX	Male	0	No	No	72	Yes	Y
...	...	...	...	...	...	...	...	
3986	6242-FEGFD	Male	0	Yes	No	66	Yes	Y
4265	2632-UCGVD	Male	1	Yes	No	66	Yes	Y
4287	6425-YQLLO	Female	1	Yes	No	66	Yes	Y
4301	7729-XBTWX	Male	0	Yes	Yes	66	Yes	I
4375	9896-UYMIE	Male	0	No	No	66	Yes	Y

1000 rows × 21 columns

In [6]:
```
numerical_column = [feature for feature in ch.columns if ch[feature].dtypes !=
print("Number of numerical column:", len(numerical_column))
ch[numerical_column].head()
```

```
Number of numerical column: 21
```

Out[6]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service
1	5575-GNVDE	Male	0	No	No	34	Yes	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service
4	9237-HQITU	Female	0	No	No	2	Yes	No

5 rows × 21 columns

In [7]:
```
sns.heatmap(ch.corr(numeric_only = True),annot = True)
```

Out[7]:
```
<Axes: >
```

In [8]:
```
ch.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]:
```
ch.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 [10]:
```
ch.shape
```

Out[10]:
```
(7043, 21)
```

Contract Analysis

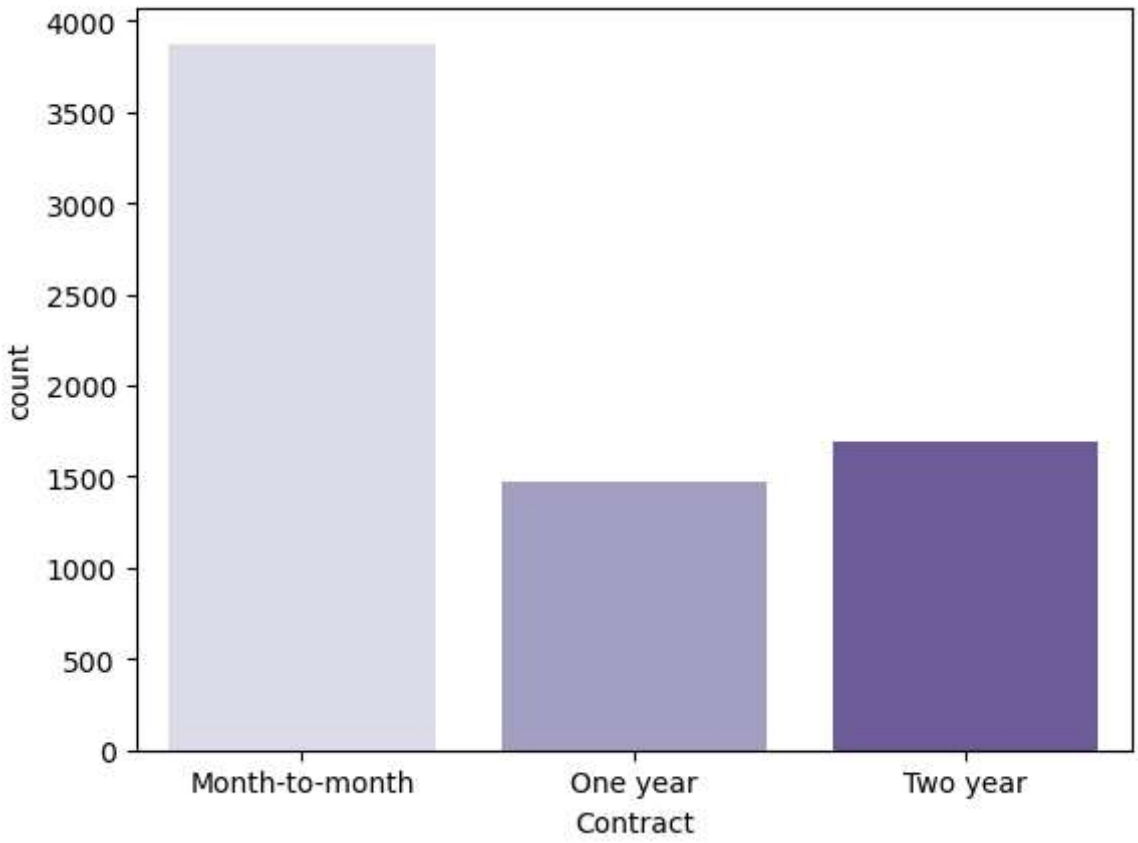
In [11]:
```
ch.groupby("Contract").size()
```

Out[11]:
```
Contract
Month-to-month    3875
One year          1473
Two year          1695
dtype: int64
```

Most likely Month-to-Month customer are most churners because they are not binded with any contract

In [12]:
```
sns.countplot(x = "Contract",palette = "Purples" ,data = ch)
```

Out[12]:
```
<Axes: xlabel='Contract', ylabel='count'>
```



Churn analysis

In [13]:
```
ch["Contract_binary"] = ch["Contract"].map({"Month-to-month":0,"One year":1,"T
# Month-to-Month = 0, one year = 1,Two year = 2
```

In [14]:
```
ch["Contract_binary"].isnull().sum()
```

Out[14]:
```
0
```

In [15]:
```
ch["Churn"].unique()
```

Out[15]:
```
array(['No', 'Yes'], dtype=object)
```

In [16]:
```
ch["Churn"].value_counts()
```

Out[16]:
```
No     5174
Yes    1869
Name: Churn, dtype: int64
```

In [17]:
```
type(ch["Churn"]).info
```

Out[17]:
```
<function pandas.core.series.Series.info(self, verbose: 'bool | None' = None,
buf: 'IO[str] | None' = None, max_cols: 'int | None' = None, memory_usage: 'b
ool | str | None' = None, show_counts: 'bool' = True) -> 'None'>
```

In [18]:
```
ch['Churn'] = np.where(ch['Churn']=='Yes',1,0)
```

In [19]:
```
#ch["Churn"] = ch["Churn"].map({"yes":1,"No":0})
# yes = 1 and No =0
```

In [20]:
```
ch["Churn"].isnull().sum()
```

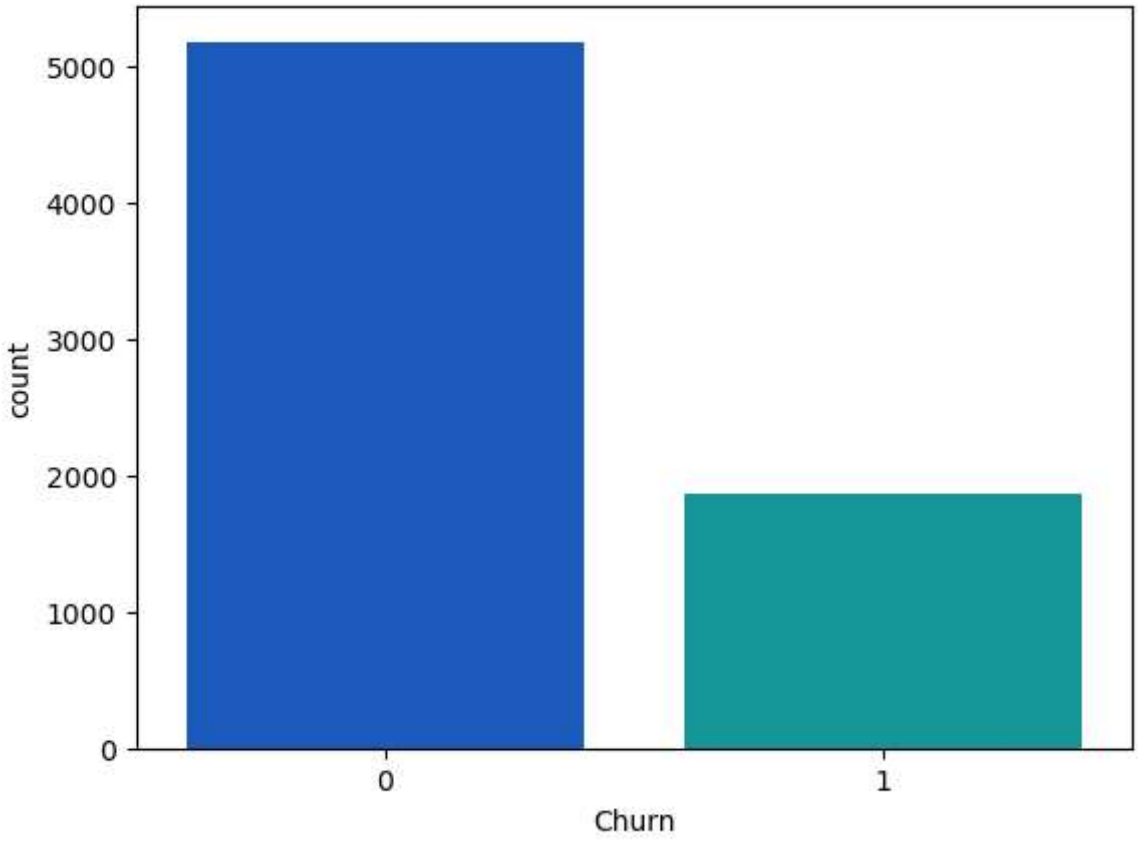
Out[20]:
```
0
```

In [21]:
```
ch.groupby("Churn").size()
```

Out[21]:
```
Churn
0    5174
1    1869
dtype: int64
```

In [22]:
```
sns.countplot(x = "Churn",palette = "winter",data = ch)
```

Out[22]:
```
<Axes: xlabel='Churn', ylabel='count'>
```



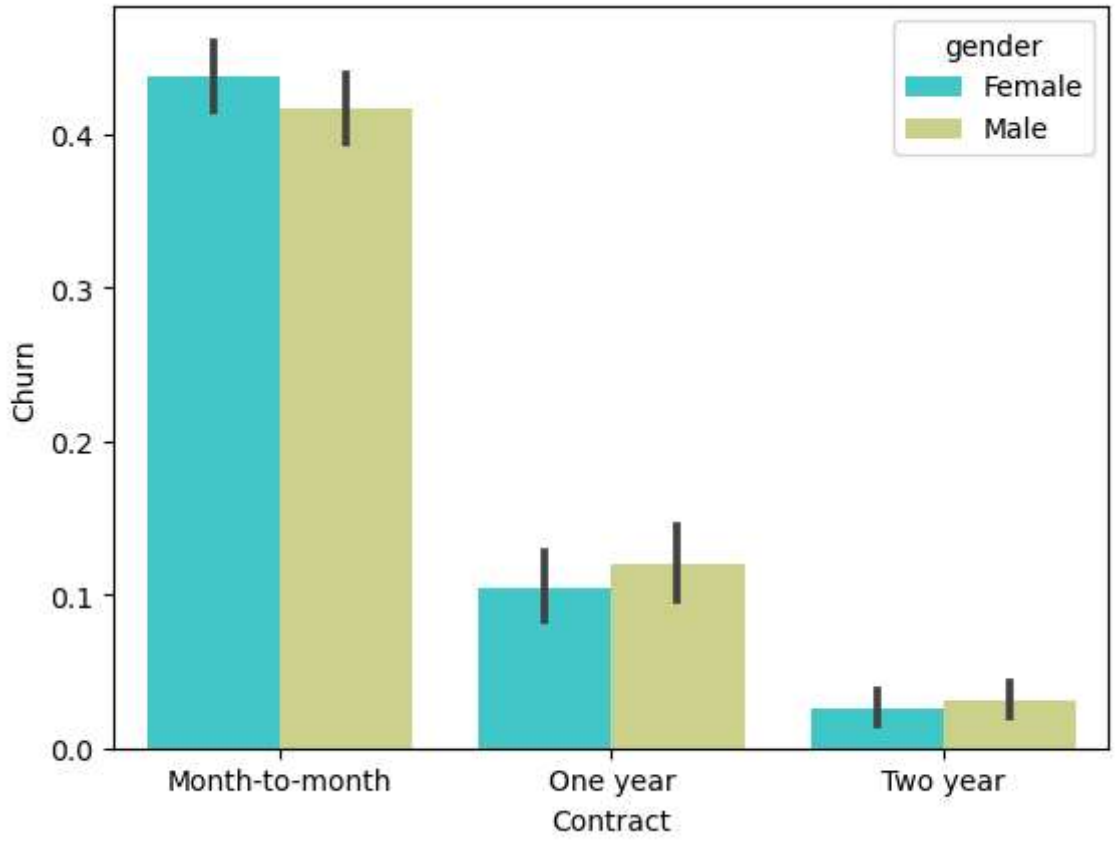
Contract vs Churner

According to this analysis most churners are belongs to one monthly contracts

** In monthly contract womens has done more churn than men

** IN one year or Two year contract mens are doing more churn than women

In [23]:
```
sns.barplot(x = "Contract", y = "Churn", data = ch,palette = "rainbow",hue = "
plt.show()
```

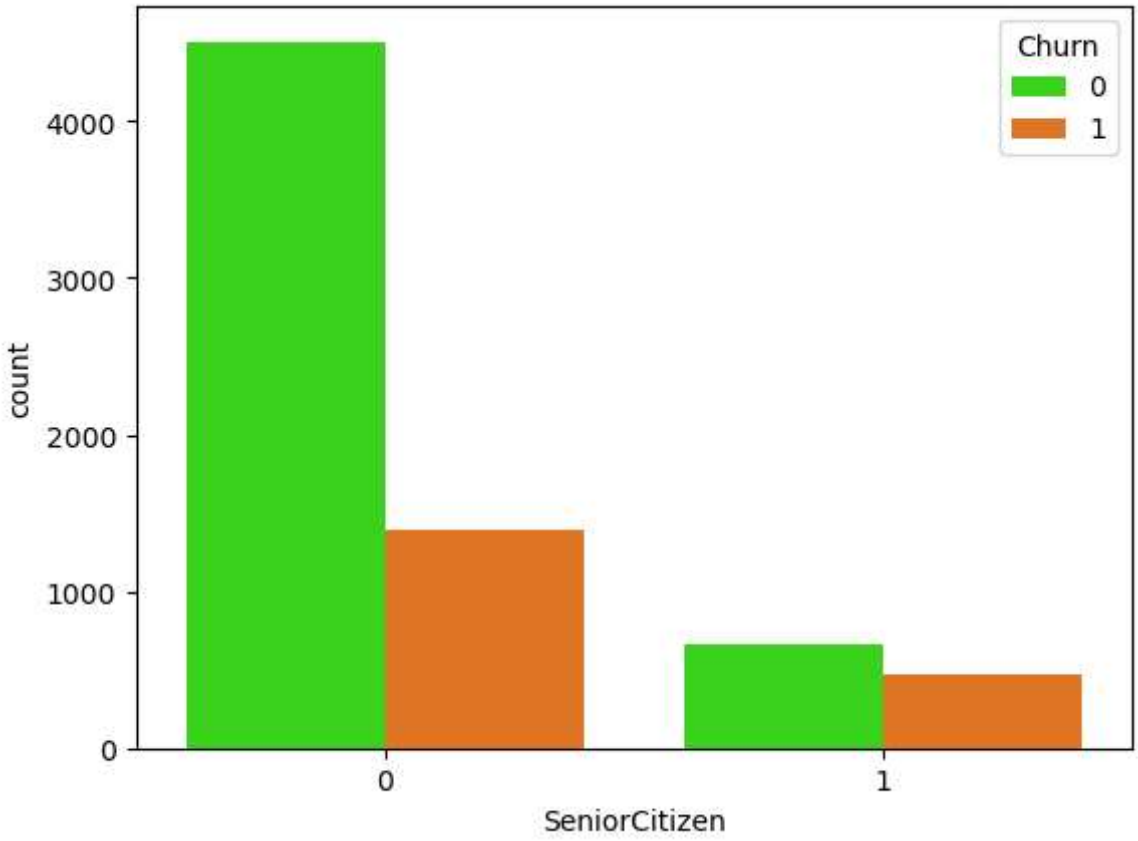


In [24]:
```
type("SeniorCitizen")
```

Out[24]:
```
str
```

In [25]:
```
sns.countplot(x ="SeniorCitizen",palette = "gist_ncar", hue = "Churn",data =ch
```

Out[25]:
```
<Axes: xlabel='SeniorCitizen', ylabel='count'>
```



Partner

In [26]:
```
ch["Partner"].unique()
```

Out[26]:
```
array(['Yes', 'No'], dtype=object)
```

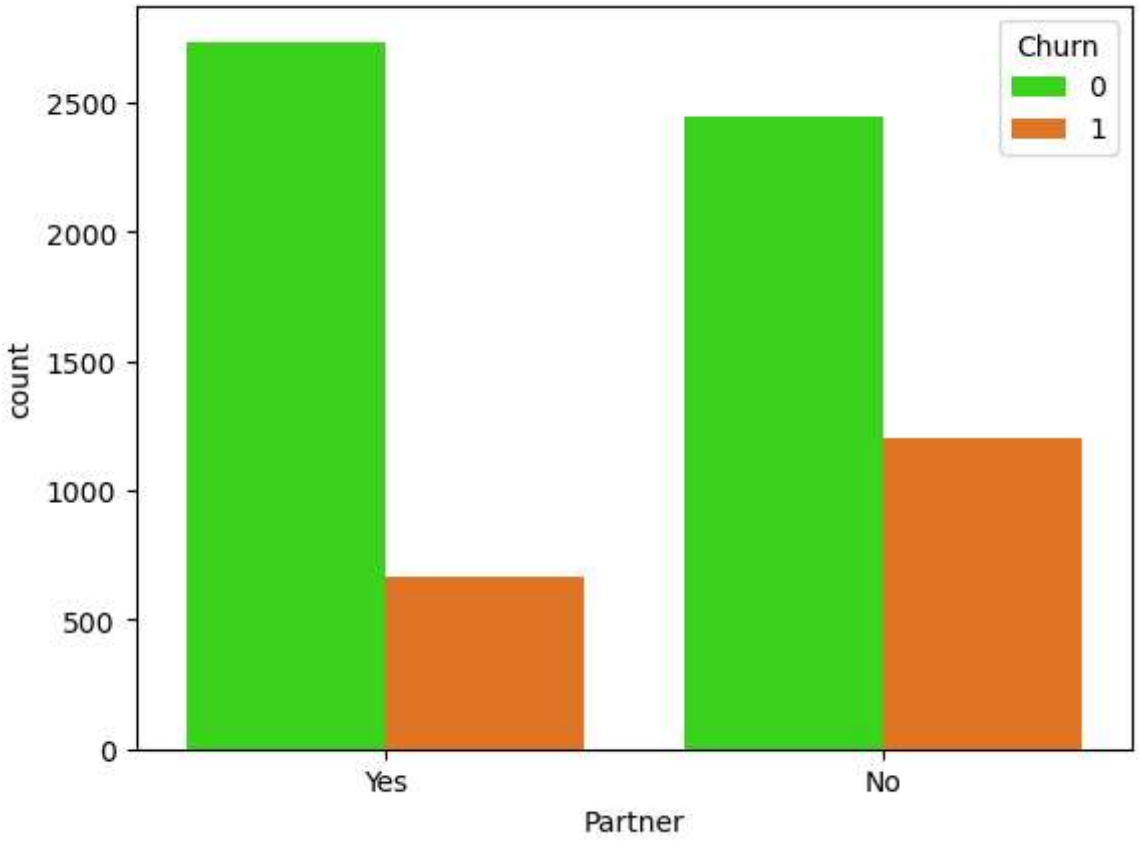
In [27]:
```
ch.groupby("Partner").size()
```

Out[27]:
```
Partner
No     3641
Yes    3402
dtype: int64
```

According to this analysis mostly unmarried people stop using this product

In [28]:
```
sns.countplot(x = "Partner",hue = "Churn",palette = "gist_ncar",data = ch)# Ac
# Mostly married people are well settled and mature  so they want stable life
```

Out[28]: <Axes: xlabel='Partner', ylabel='count'>

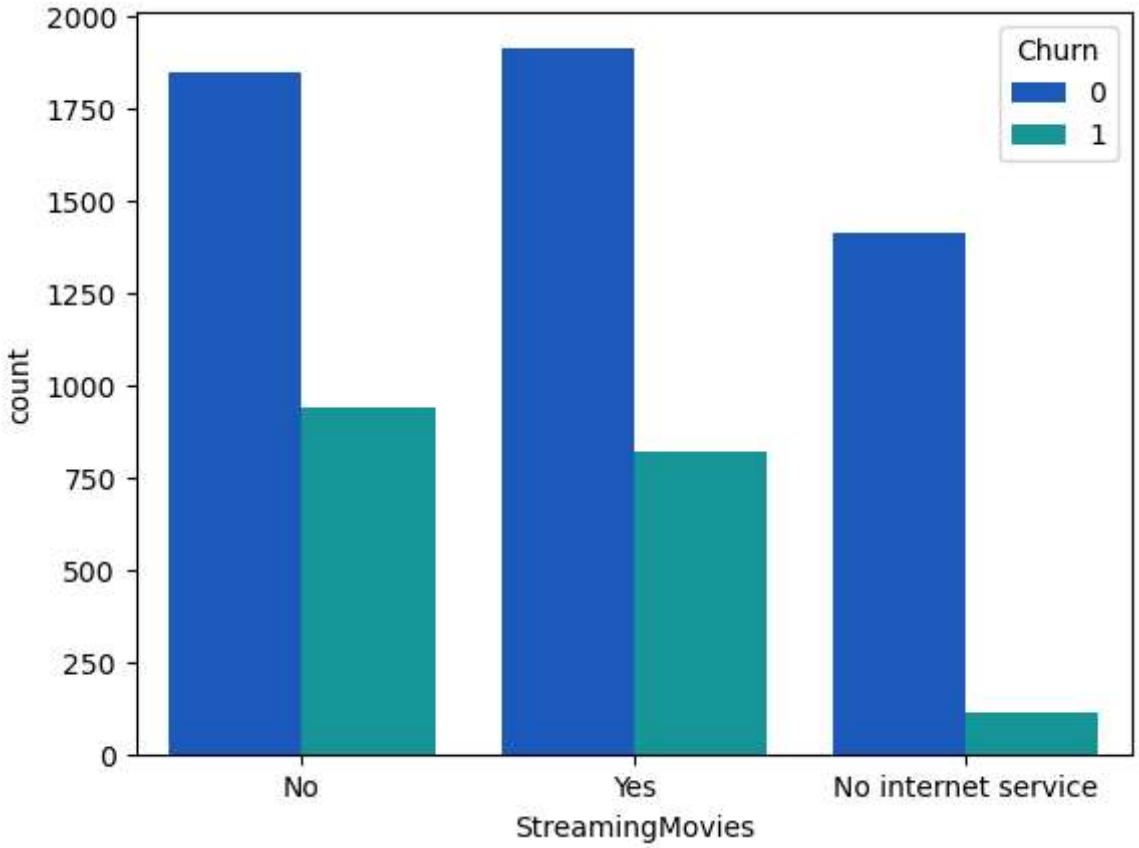


StreamingMovies

The people Who is not watching movie most likely stop using the product

In [29]:
```
sns.countplot(x = "StreamingMovies",hue = "Churn",palette = "winter", data  =
```

Out[29]:
```
<Axes: xlabel='StreamingMovies', ylabel='count'>
```



Data Cleaning

1)Create a copy of base data for manupulation and processing

In [30]:
```
ch_base_data = ch.copy()
```

In [31]:
```
ch["TotalCharges"] = pd.to_numeric(ch["TotalCharges"], errors = "coerce")
```

In [32]:
```
ch_base_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 22 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   int32  
 21  Contract_binary   7043 non-null   int64  
dtypes: float64(1), int32(1), int64(3), object(17)
memory usage: 1.2+ MB
```

In [33]:
```
ch.isnull().sum()
```

```
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
Contract_binary      0
dtype: int64
```

In [34]:
```
ch.value_counts().sum()
```

Out[34]: 7032

In [35]:
```
ch["TotalCharges"].value_counts().sum()
```

Out[35]: 7032

Only 11 null values in total charges

In [36]:
```
ch.loc[ch['TotalCharges'].isnull() == True]
```

Out[36]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLin
488	4472-LVYGI	Female	0	Yes	Yes	0	No	No pho servi
753	3115-CZMZD	Male	0	No	Yes	0	Yes	I
936	5709-LVOEQ	Female	0	Yes	Yes	0	Yes	I
1082	4367-NUYAO	Male	0	Yes	Yes	0	Yes	Y
1340	1371-DWPAZ	Female	0	Yes	Yes	0	No	No pho servi
3331	7644-OMVMY	Male	0	Yes	Yes	0	Yes	I
3826	3213-VVOLG	Male	0	Yes	Yes	0	Yes	Y
4380	2520-SGTTA	Female	0	Yes	Yes	0	Yes	I
5218	2923-ARZLG	Male	0	Yes	Yes	0	Yes	I
6670	4075-WKNIU	Female	0	Yes	Yes	0	Yes	Y
6754	2775-SEFEE	Male	0	No	Yes	0	Yes	Y

11 rows × 22 columns

In [37]:
```
ch["TotalCharges"].unique()
```

Out[37]: array([29.85, 1889.5 , 108.15, ..., 346.45, 306.6 , 6844.5])

In [38]:
```
ch["TotalCharges"] = ch["TotalCharges"].fillna(0)
```

In [39]:
```
ch["TotalCharges"].isnull().sum()
```

Out[39]:
```
0
```

In [40]:
```
# Group the tenure in bins of 12 months
labels = ["{0} - {1}".format(i, i + 11) for i in range(1, 72, 12)]

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

Tenure Analysis

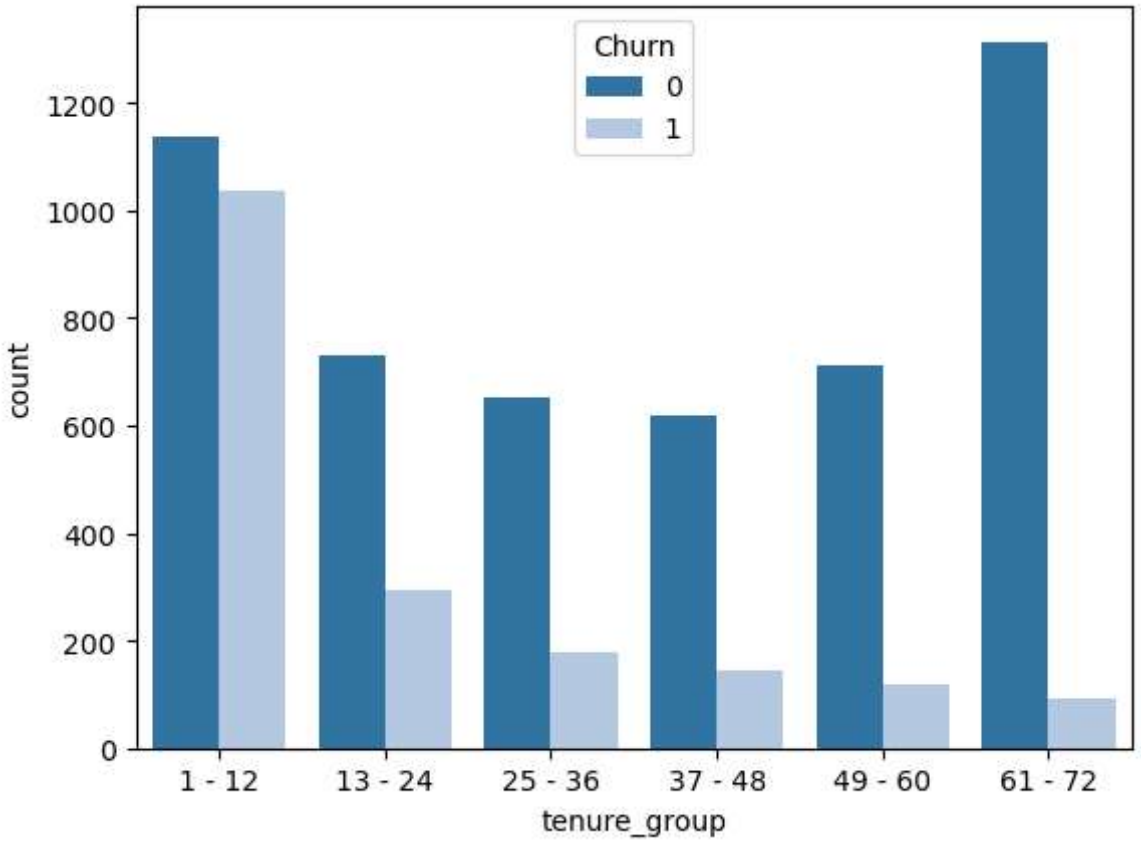
In [41]:
```
ch["tenure_group"].value_counts()
```

Out[41]:
```
1 - 12     2175
61 - 72    1407
13 - 24    1024
25 - 36     832
49 - 60     832
37 - 48     762
Name: tenure_group, dtype: int64
```

Most churners belongs to 1-12 months using period

In [42]:
```
sns.countplot(x = "tenure_group",hue = "Churn",palette = "tab20",data = ch)
```

Out[42]:
```
<Axes: xlabel='tenure_group', ylabel='count'>
```



In [43]:
```
tn = ch.loc[ch['tenure'].nlargest(n=5)]
tn
```

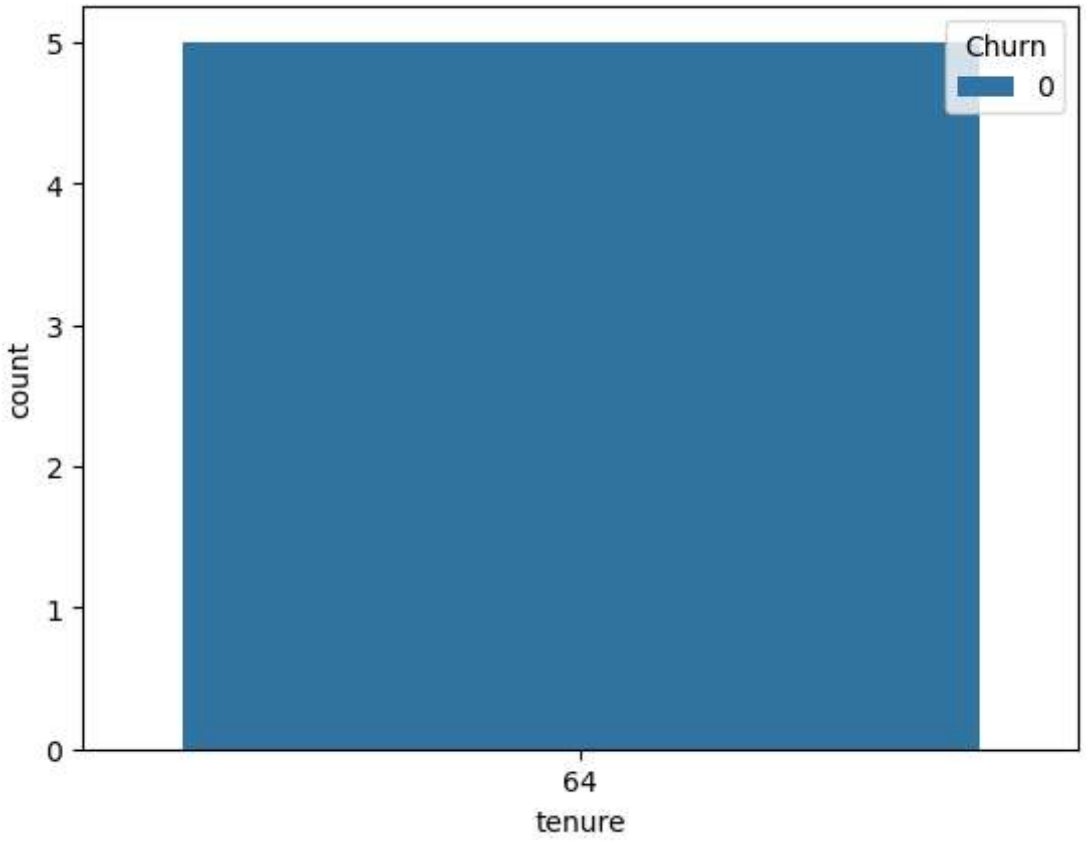
Out[43]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
72	1891-QRQSA	Male	1	Yes	Yes	64	Yes	Yes
72	1891-QRQSA	Male	1	Yes	Yes	64	Yes	Yes
72	1891-QRQSA	Male	1	Yes	Yes	64	Yes	Yes
72	1891-QRQSA	Male	1	Yes	Yes	64	Yes	Yes
72	1891-QRQSA	Male	1	Yes	Yes	64	Yes	Yes

5 rows × 23 columns

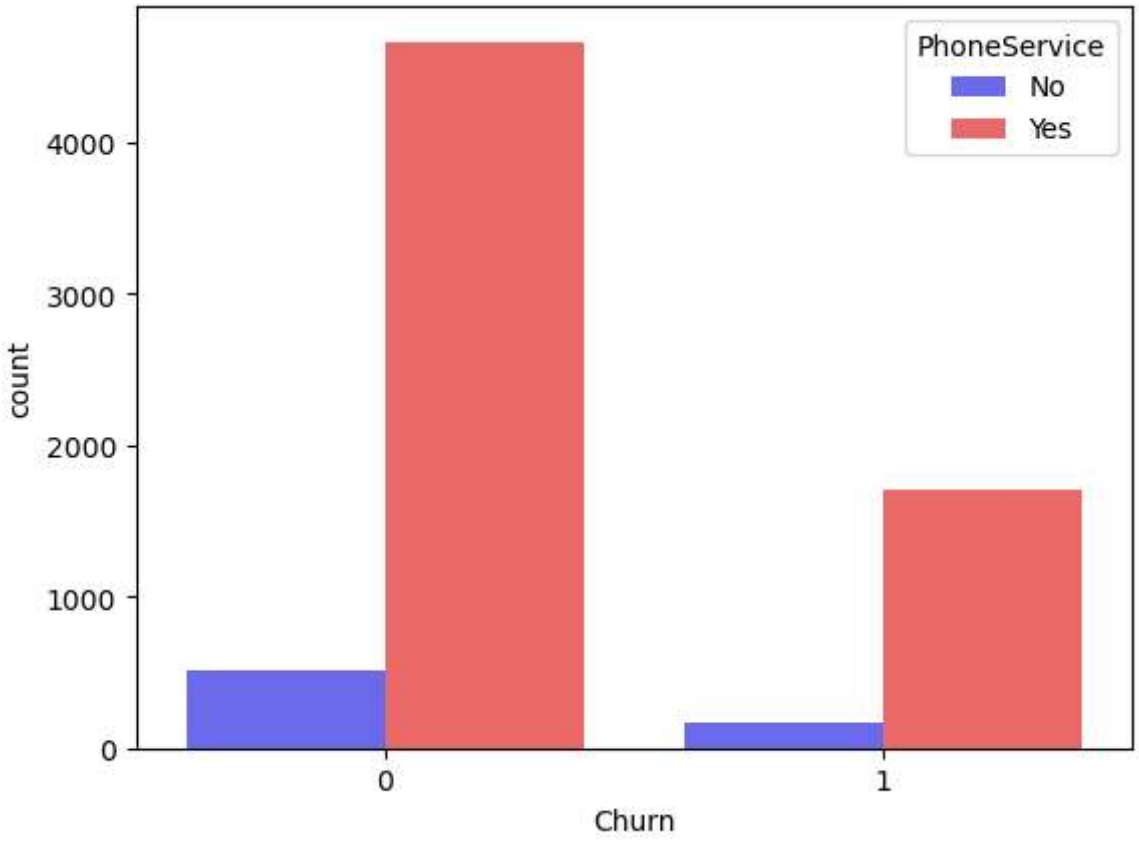
In [44]:
```
sns.countplot(x = "tenure",palette = "tab20",hue = "Churn",data = tn)
```

Out[44]:
```
<Axes: xlabel='tenure', ylabel='count'>
```



In [45]:
```
sns.countplot(x = "Churn",hue = "PhoneService",palette = "seismic",data = ch)
```

Out[45]:
```
<Axes: xlabel='Churn', ylabel='count'>
```



In [46]:
```
ch.columns
```

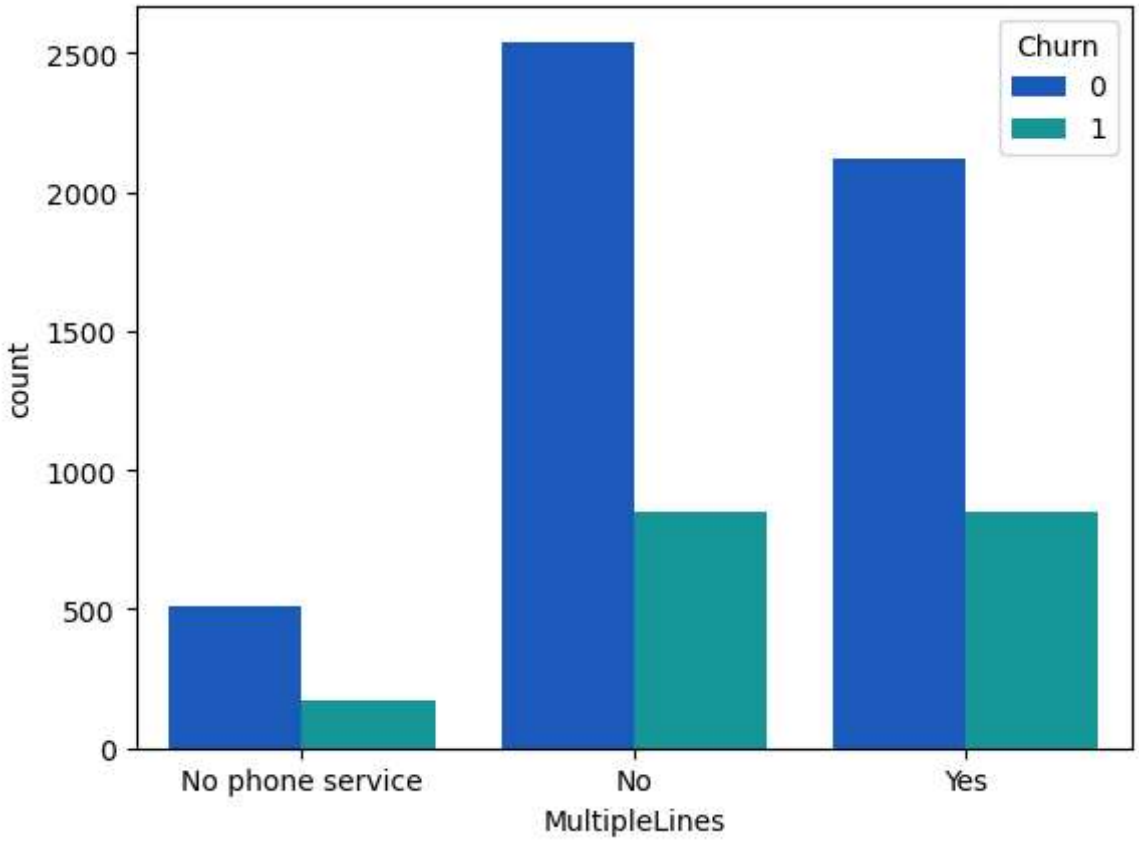
Out[46]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn',
       'Contract_binary', 'tenure_group'],
      dtype='object')
```

MultipleLines Analysis

most churners as not Multiplelines user

In [47]:
```
sns.countplot(x = "MultipleLines",hue = "Churn",palette = "winter",data = ch)
```

Out[47]:
```
<Axes: xlabel='MultipleLines', ylabel='count'>
```



In [48]:
```
ch["Churn"].value_counts()
```

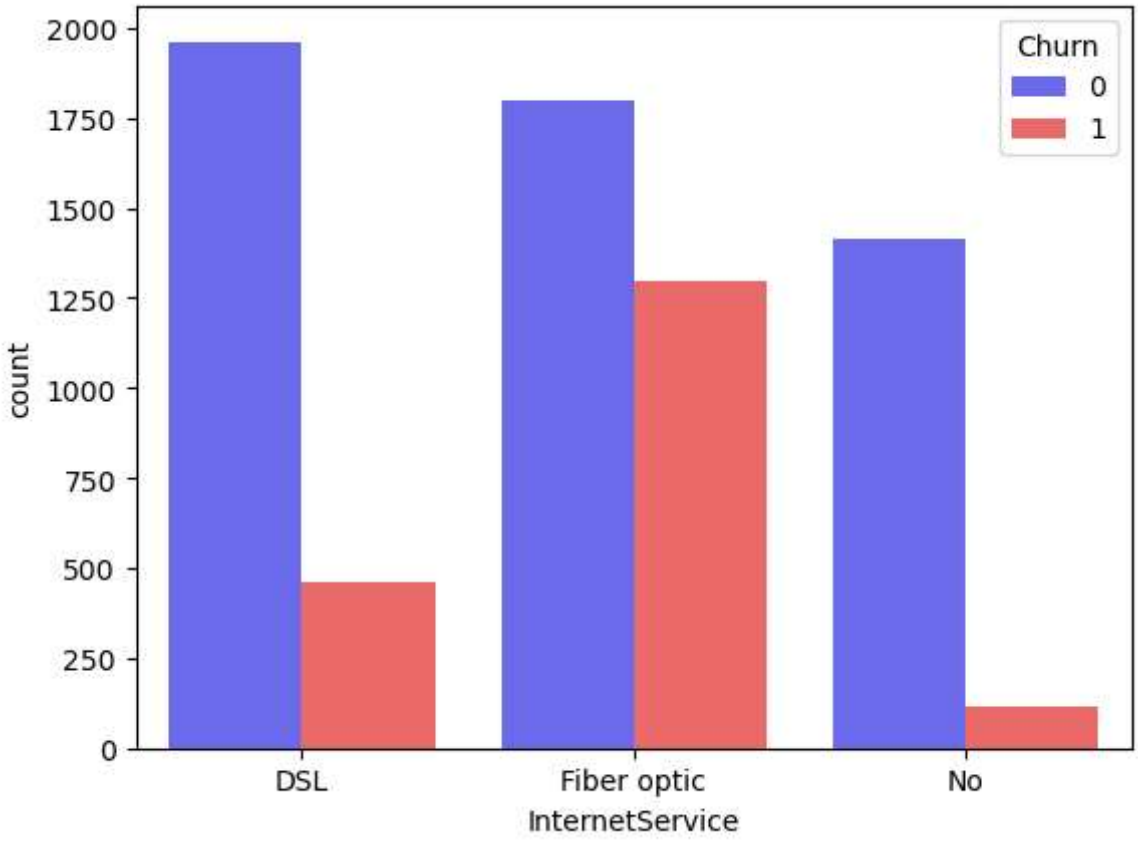
Out[48]:
```
0    5174
1    1869
Name: Churn, dtype: int64
```

Internet Service Analysis

According to this analysis whosoever using internetservice with Fiber optic are Churns more.

In [49]:
```
sns.countplot(x = "InternetService",hue = "Churn",palette = "seismic",data = c
```

Out[49]:
```
<Axes: xlabel='InternetService', ylabel='count'>
```

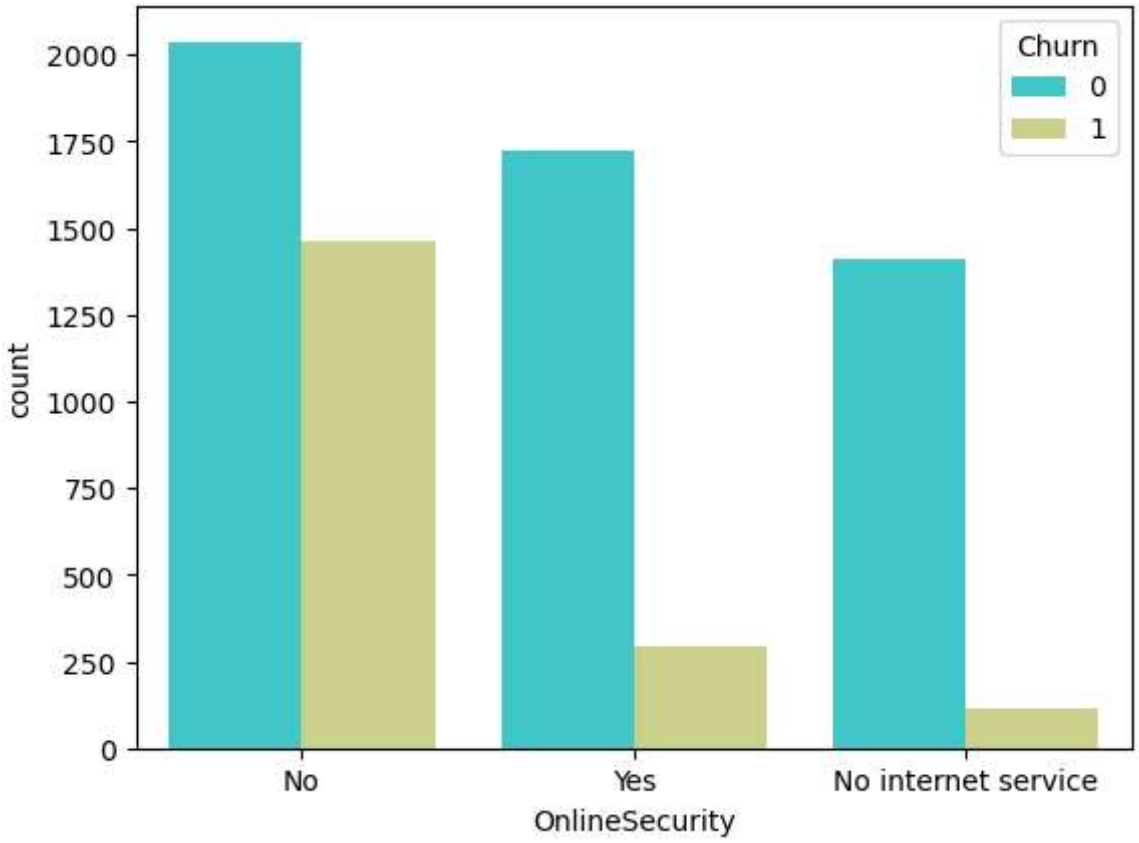


OnlineSecurity Analysis

According to this analysis whosoever is not using OnlineSecurity are Churns more.

In [50]:
```
sns.countplot(x = "OnlineSecurity",hue = "Churn",palette = "rainbow",data = ch
```

Out[50]:
```
<Axes: xlabel='OnlineSecurity', ylabel='count'>
```



In [51]:
```
ch.columns
```

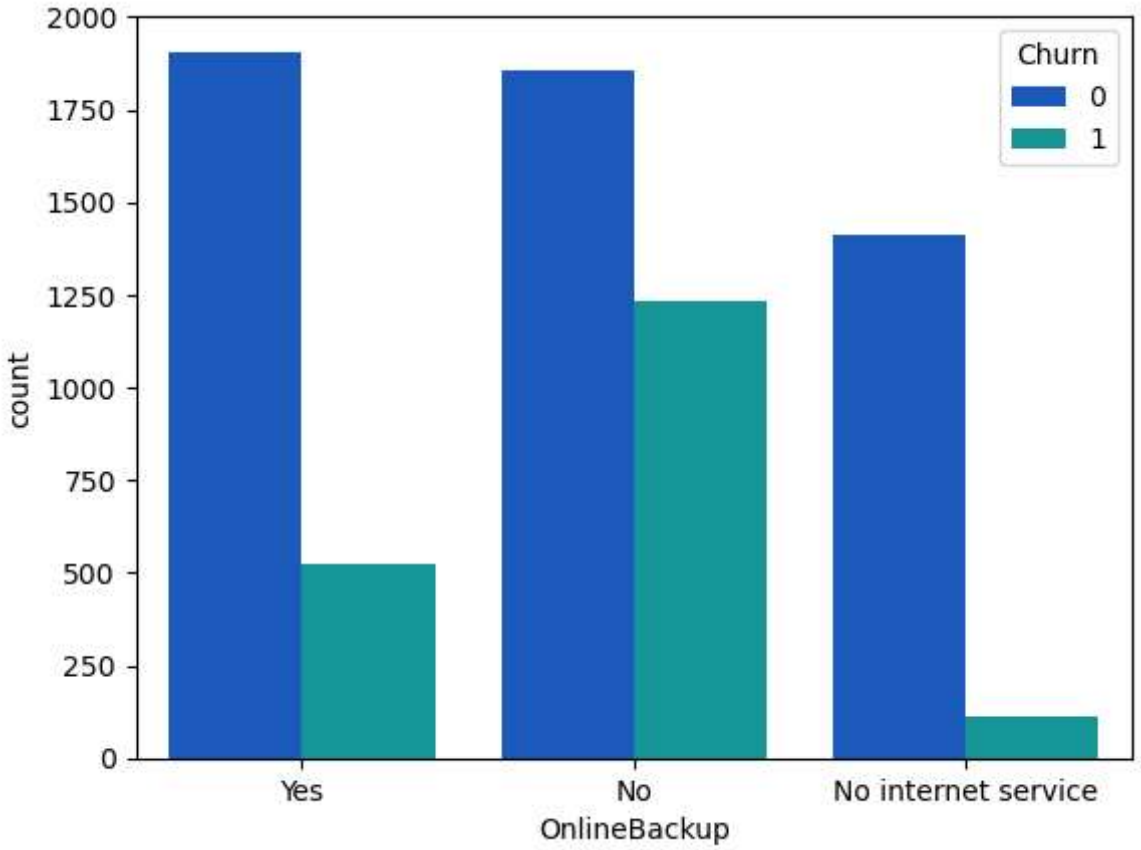
Out[51]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn',
       'Contract_binary', 'tenure_group'],
      dtype='object')
```

Online Backups Analysis

According to this analysis whosoever is not having OnlineBackups are Churns more.

In [52]:
```
sns.countplot(x = "OnlineBackup",hue = "Churn",palette = "winter",data = ch)
```

Out[52]:
```
<Axes: xlabel='OnlineBackup', ylabel='count'>
```

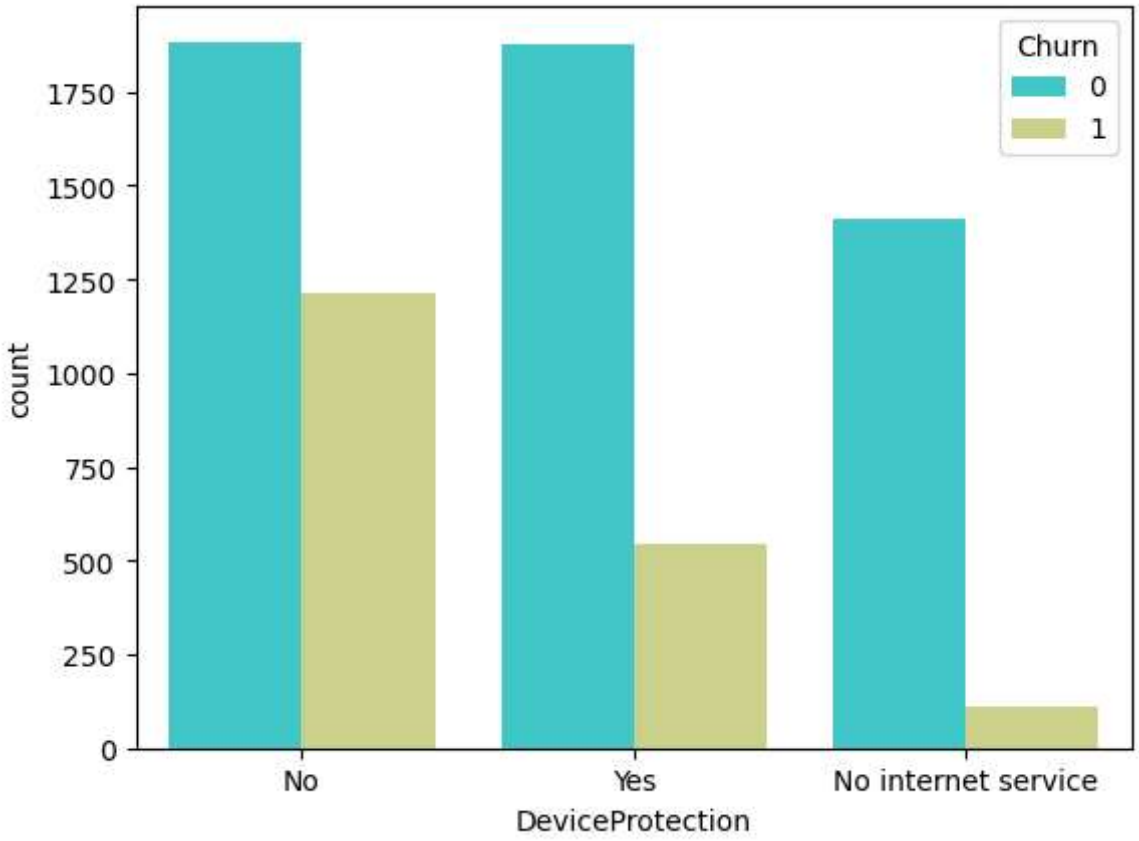


DeviceProtection Analysis

According to this analysis the person dont have device Protection Churns more.

In [53]:
```
sns.countplot(x = "DeviceProtection",hue = "Churn",palette = "rainbow",data =
```

Out[53]:
```
<Axes: xlabel='DeviceProtection', ylabel='count'>
```

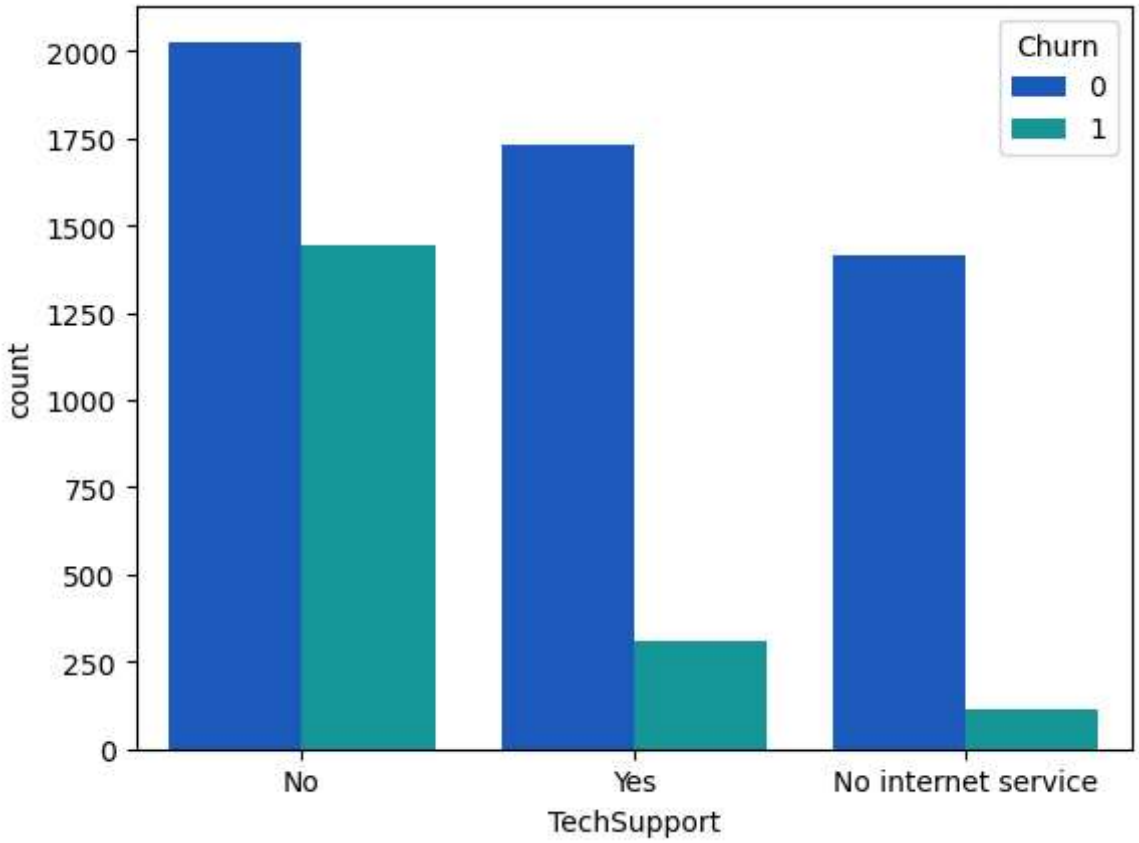


TechSupport

According to this analysis the person dont have TechSupport Churns more.

In [54]:
```
sns.countplot(x = "TechSupport",hue = "Churn",palette = "winter",data = ch)
```

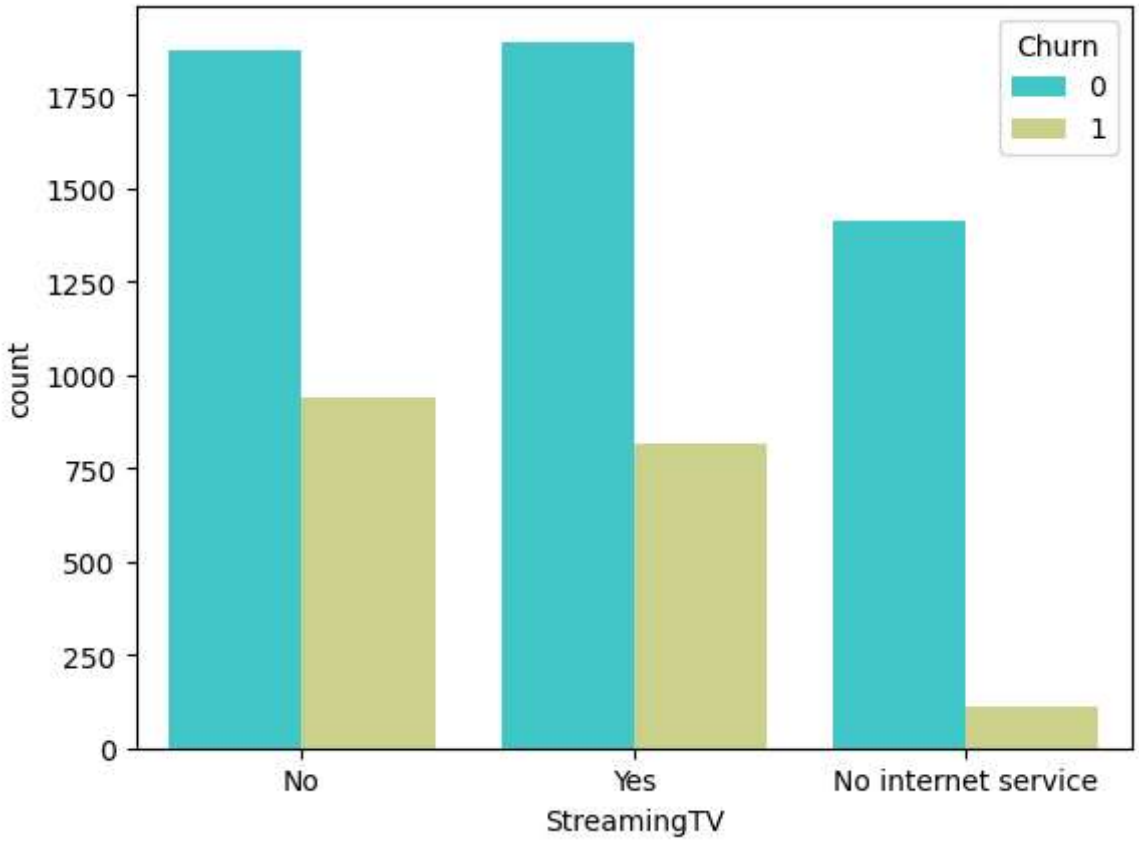
Out[54]:
```
<Axes: xlabel='TechSupport', ylabel='count'>
```



StreamingTV Analysis

In [55]:
```
sns.countplot(x = "StreamingTV",hue = "Churn",palette = "rainbow",data = ch)
```

Out[55]:
```
<Axes: xlabel='StreamingTV', ylabel='count'>
```



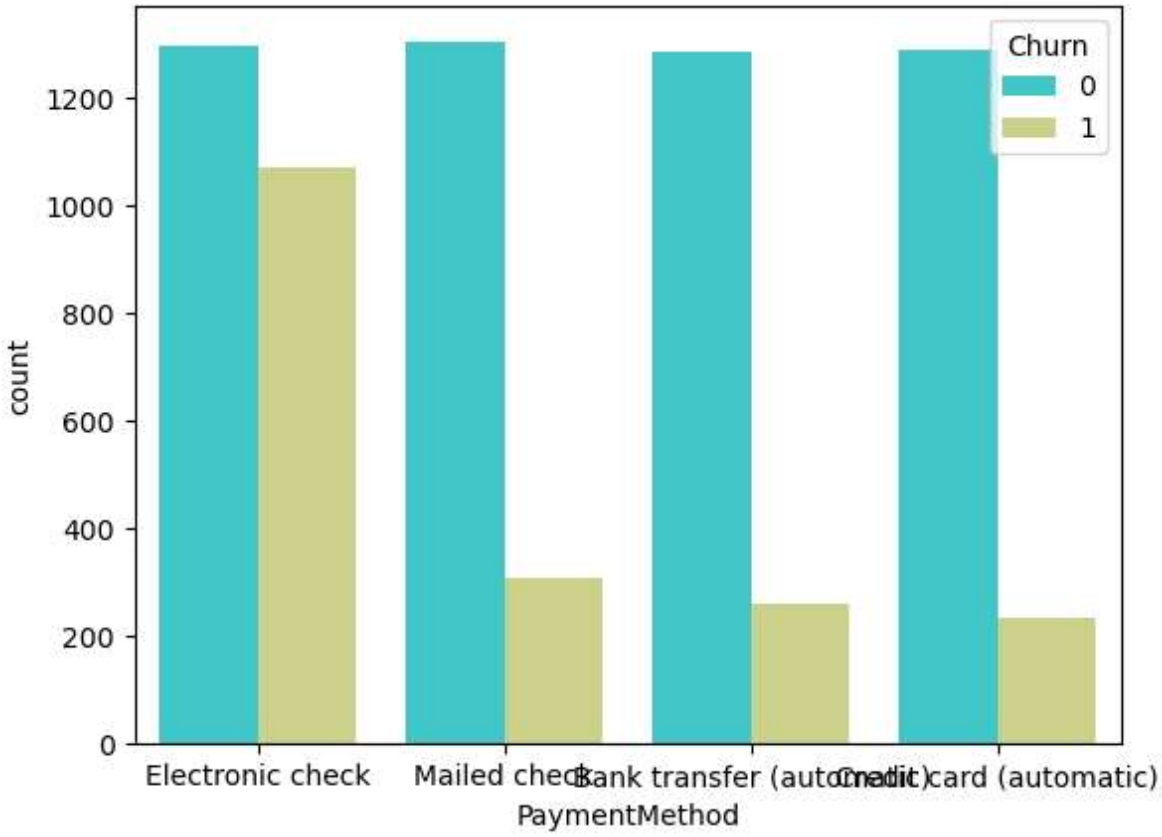
In [56]:
```
ch.columns
```

Out[56]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn',
       'Contract_binary', 'tenure_group'],
      dtype='object')
```

PaymentMethod Analysis

In [57]:
```
sns.countplot(x = "PaymentMethod",hue = "Churn",palette = "rainbow",data = ch)
```

Out[57]:
```
<Axes: xlabel='PaymentMethod', ylabel='count'>
```



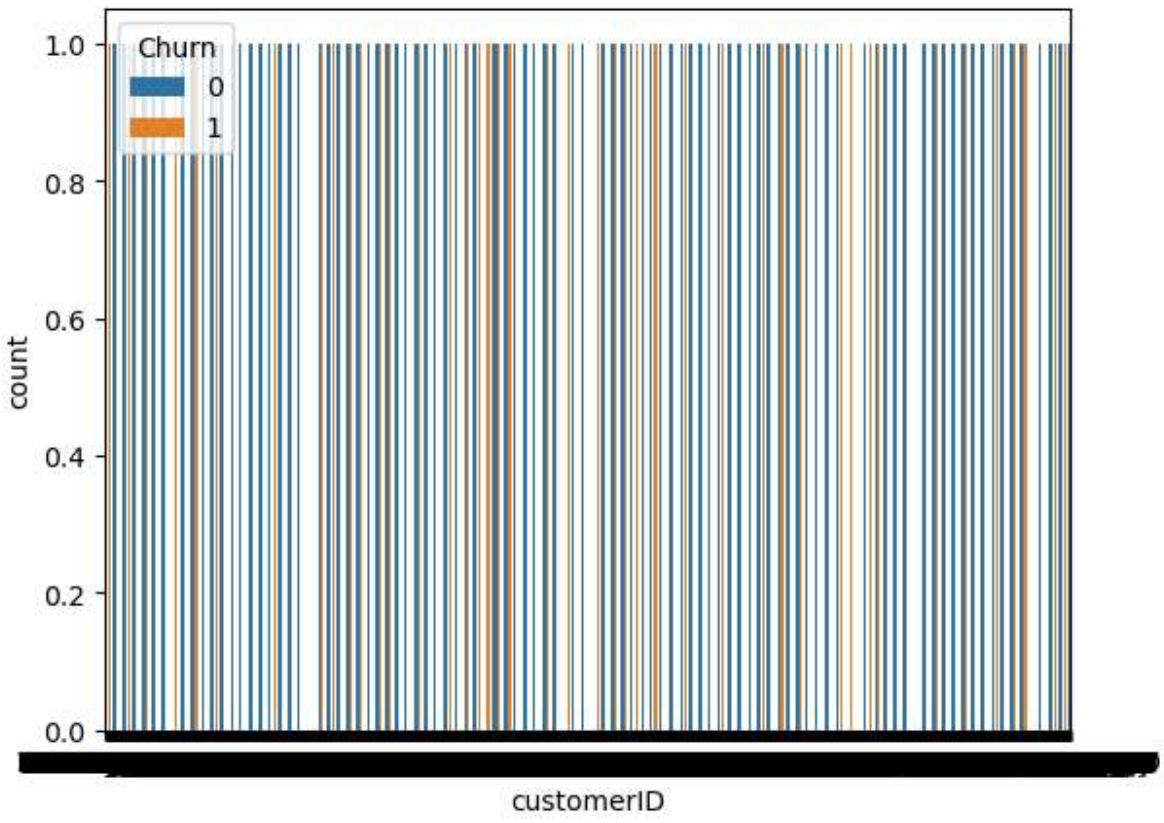
In [58]:
```
ch["MonthlyCharges"].unique()
```

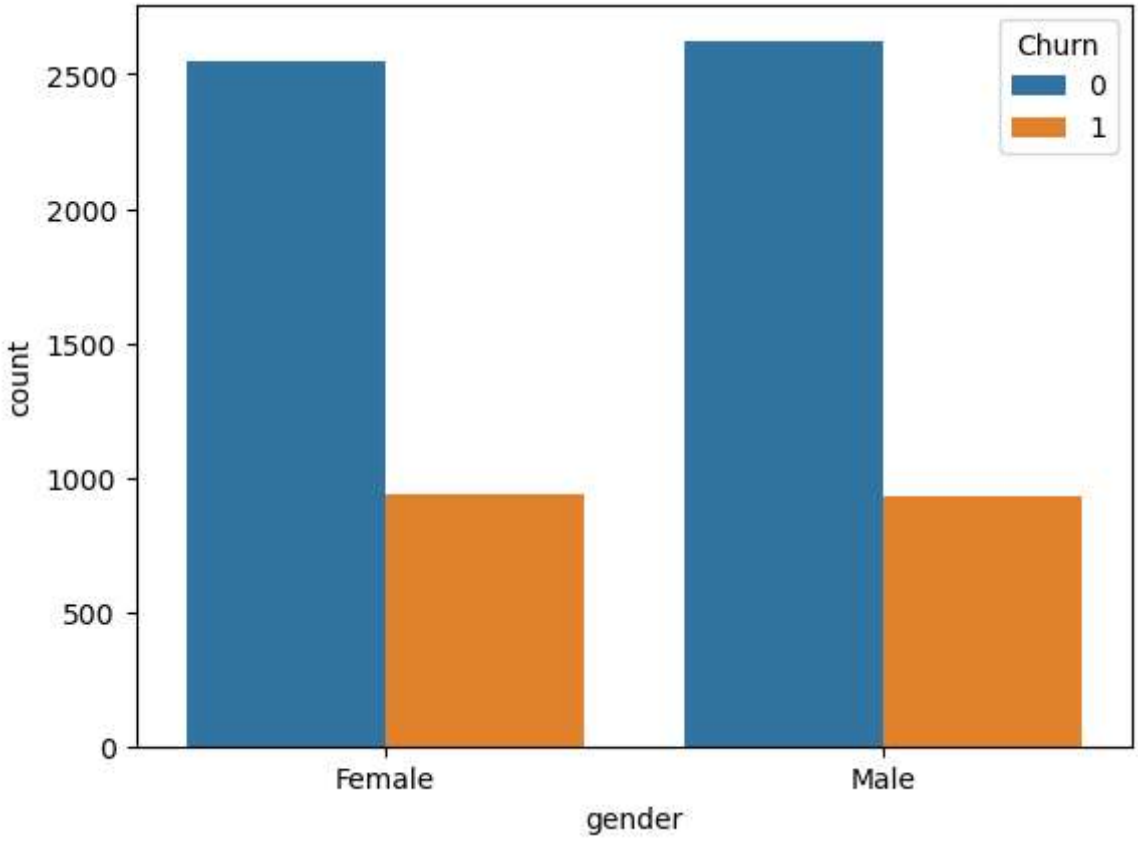
Out[58]:
```
array([29.85, 56.95, 53.85, ..., 63.1 , 44.2 , 78.7 ])
```

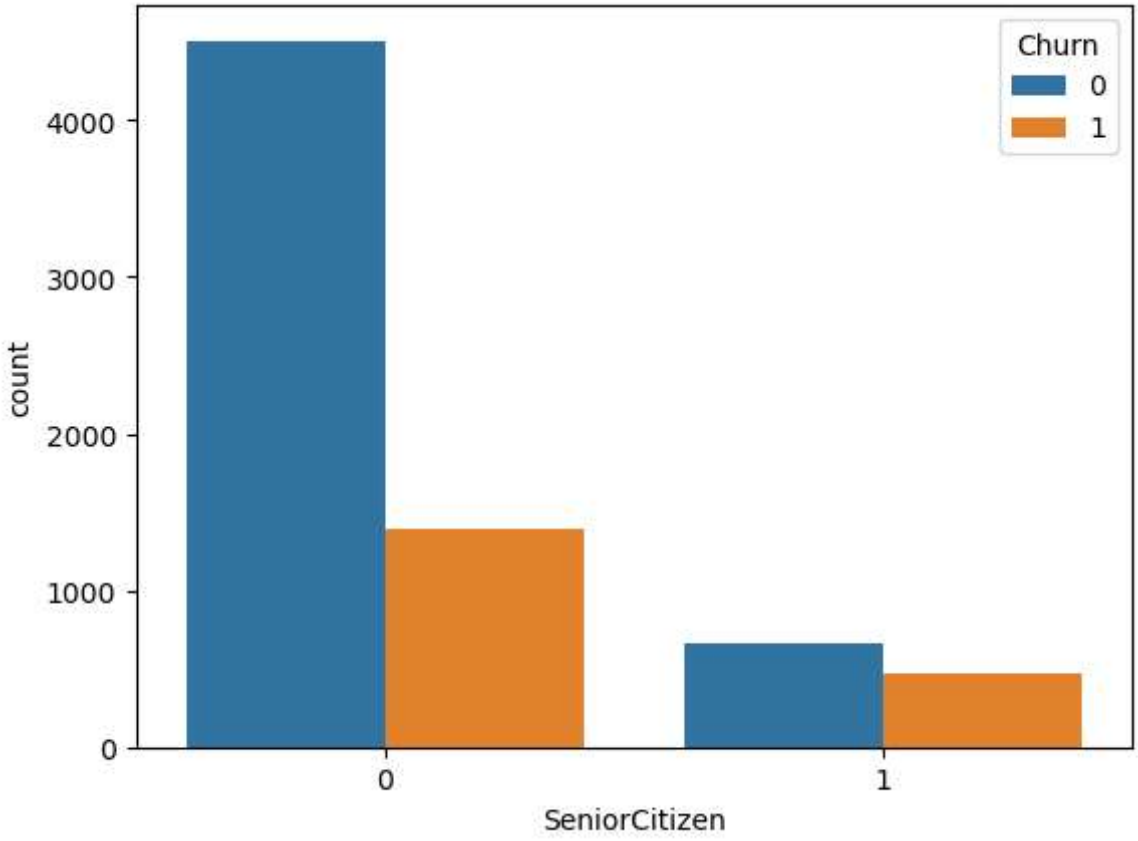
WE can plot all columns in a single step

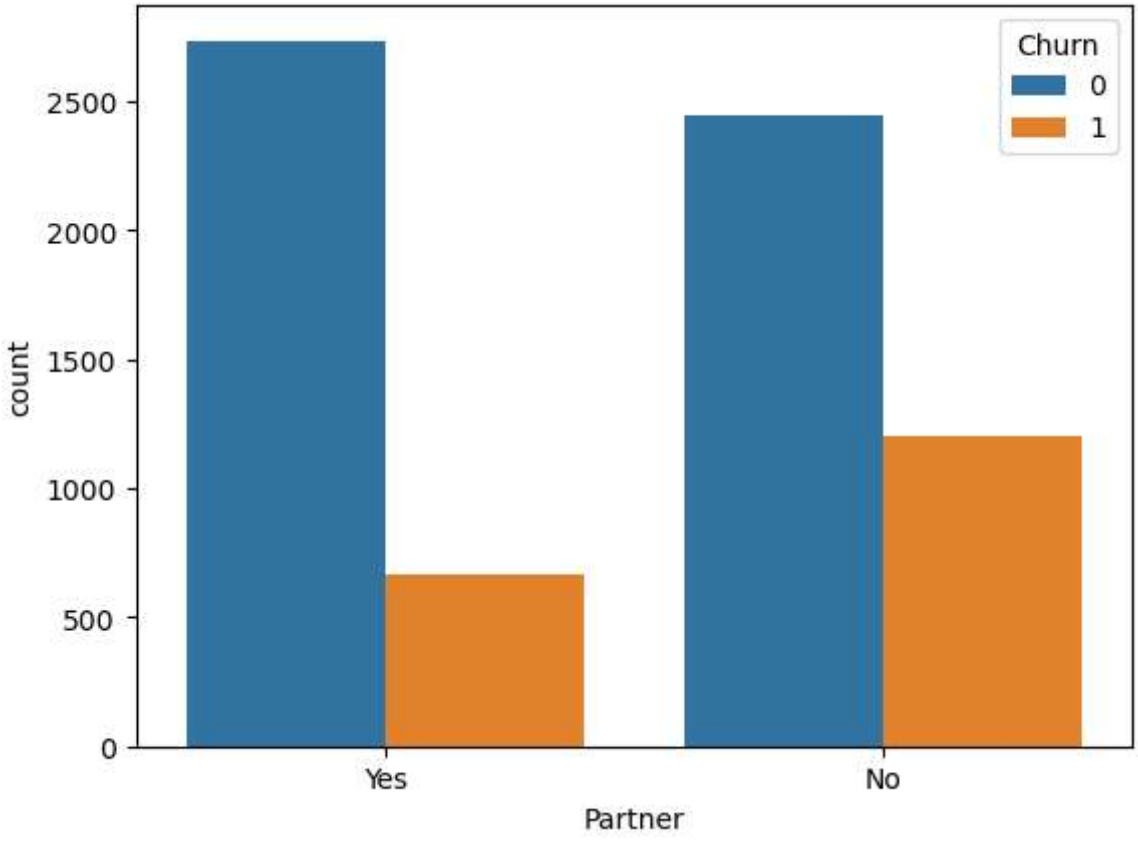
In [59]:
```
for i, predictor in enumerate(ch):
    plt.figure(i)
    sns.countplot(data=ch, x=predictor, hue='Churn')
```

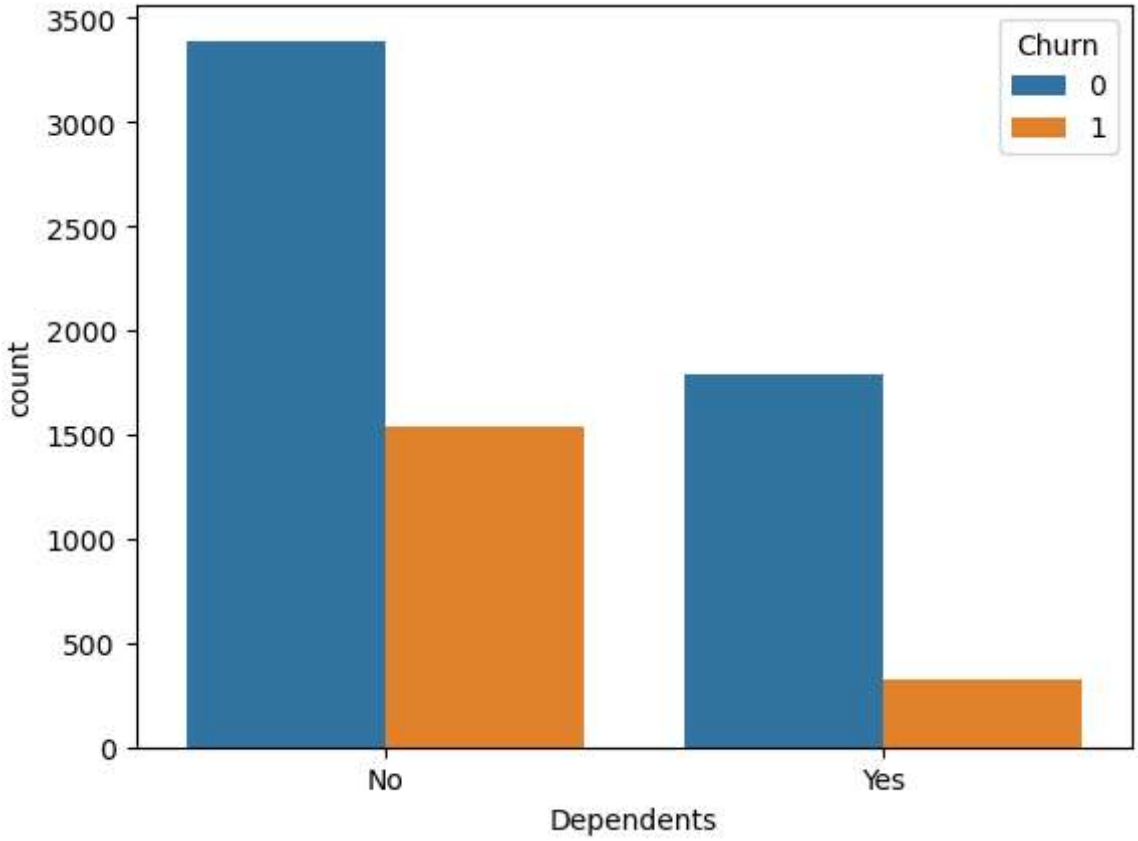
```
C:\Users\Subham Ranjan\AppData\Local\Temp\ipykernel_18980\3484135066.py:2: Ru
ntimeWarning: More than 20 figures have been opened. Figures created through
the pyplot interface (`matplotlib.pyplot.figure`) are retained until explicit
ly closed and may consume too much memory. (To control this warning, see the
rcParam `figure.max_open_warning`). Consider using `matplotlib.pyplot.close()
`.
  plt.figure(i)
```

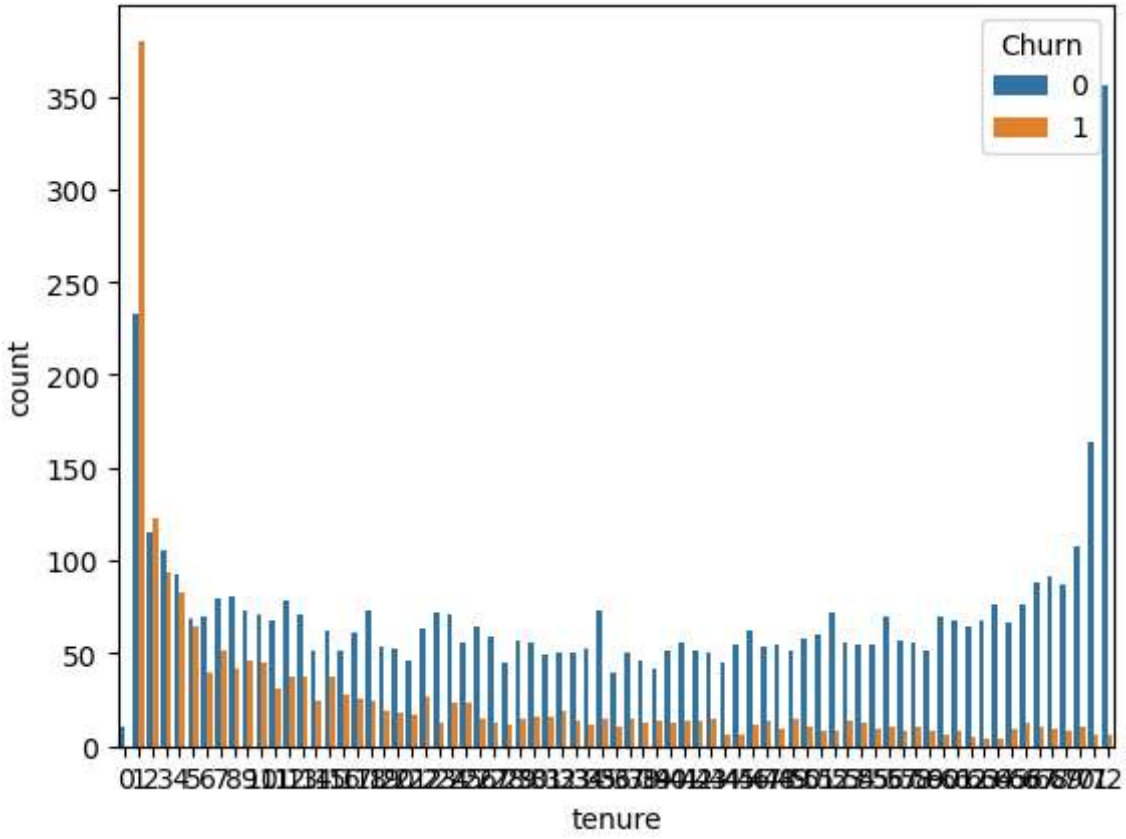


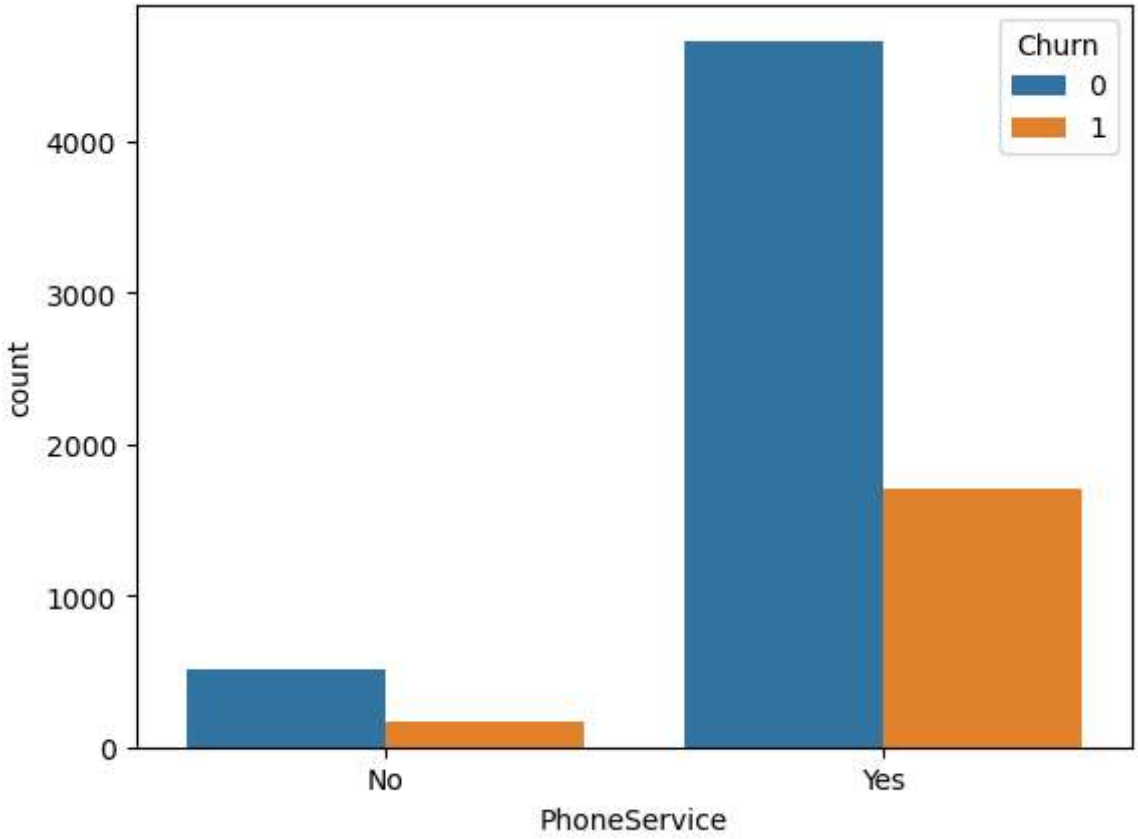

Churn
0
1
2500
2000
1500
1000
500
0
count
Female
Male
gender

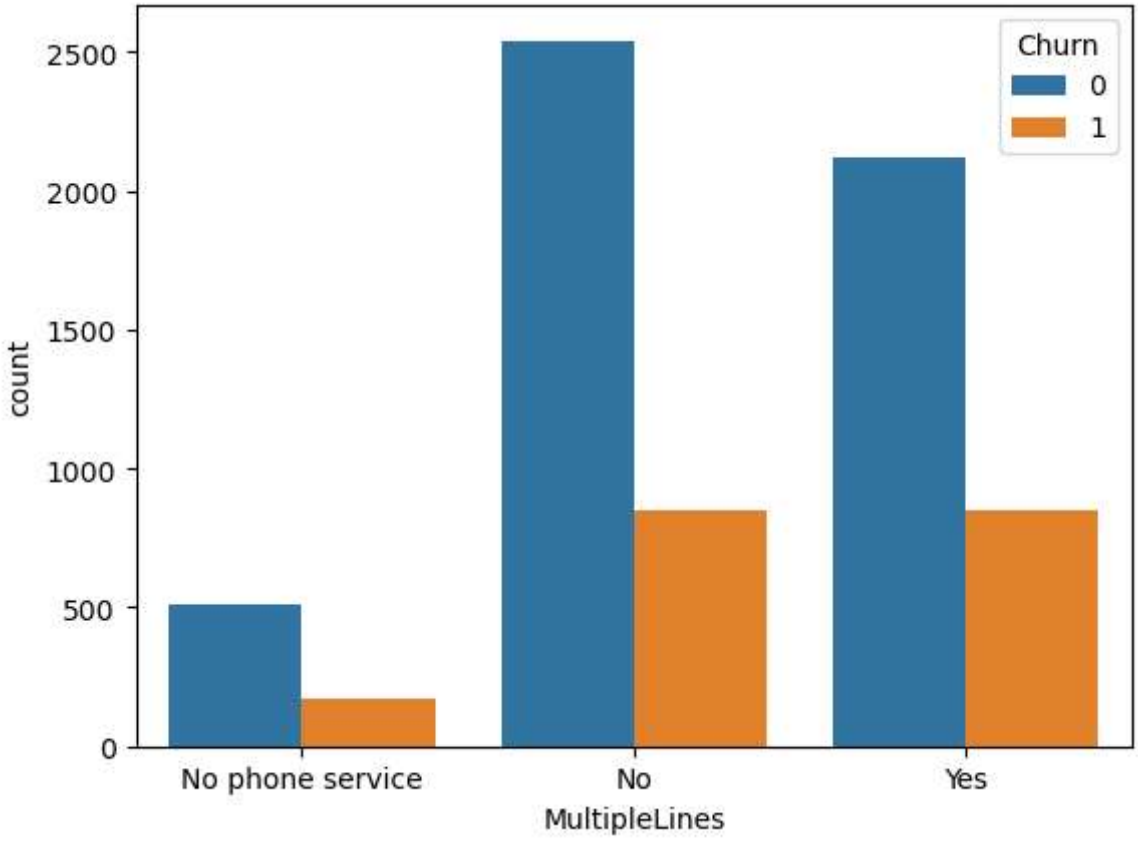

Churn
0
1
4000
3000
2000
1000
0
count
0
1
SeniorCitizen

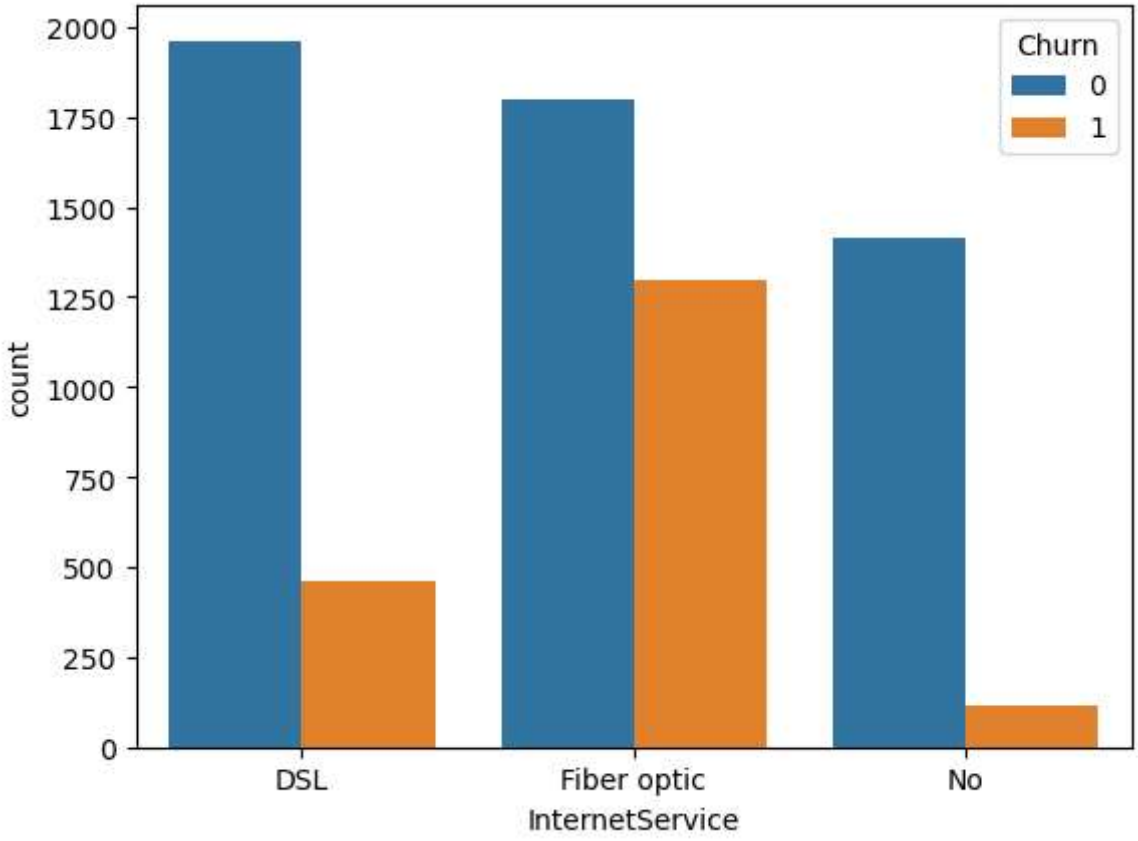

Churn
0
1
2500
2000
1500
1000
500
0
count
Yes
No
Partner

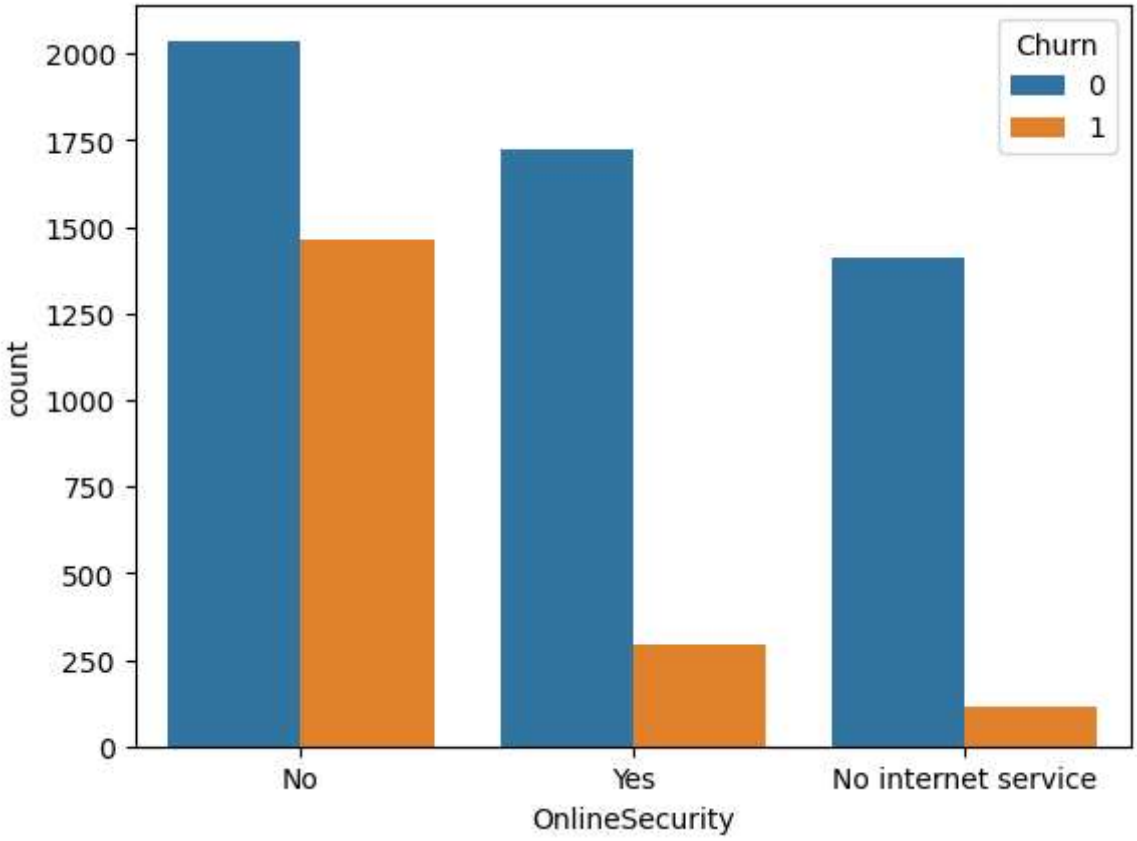

Churn
0
1
3500
3000
2500
2000
1500
1000
500
0
count
No
Yes
Dependents

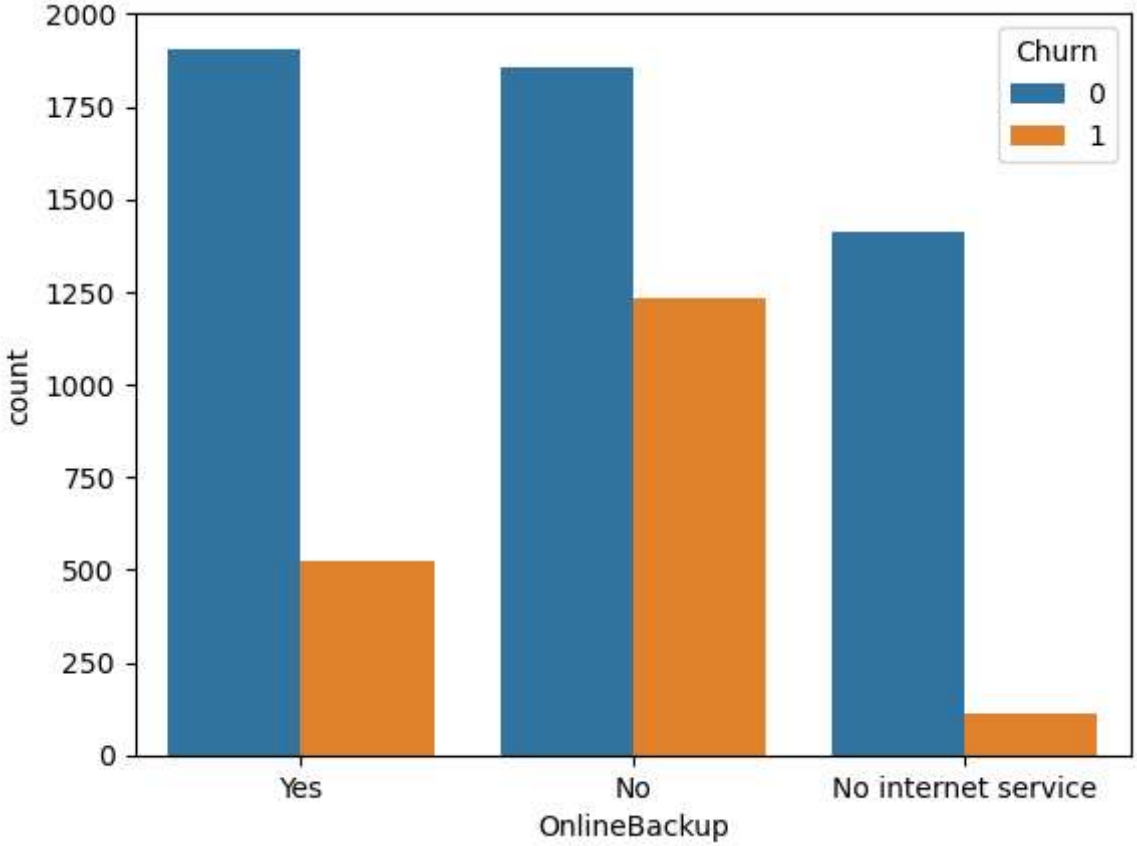

Churn
0
1
count
tenure

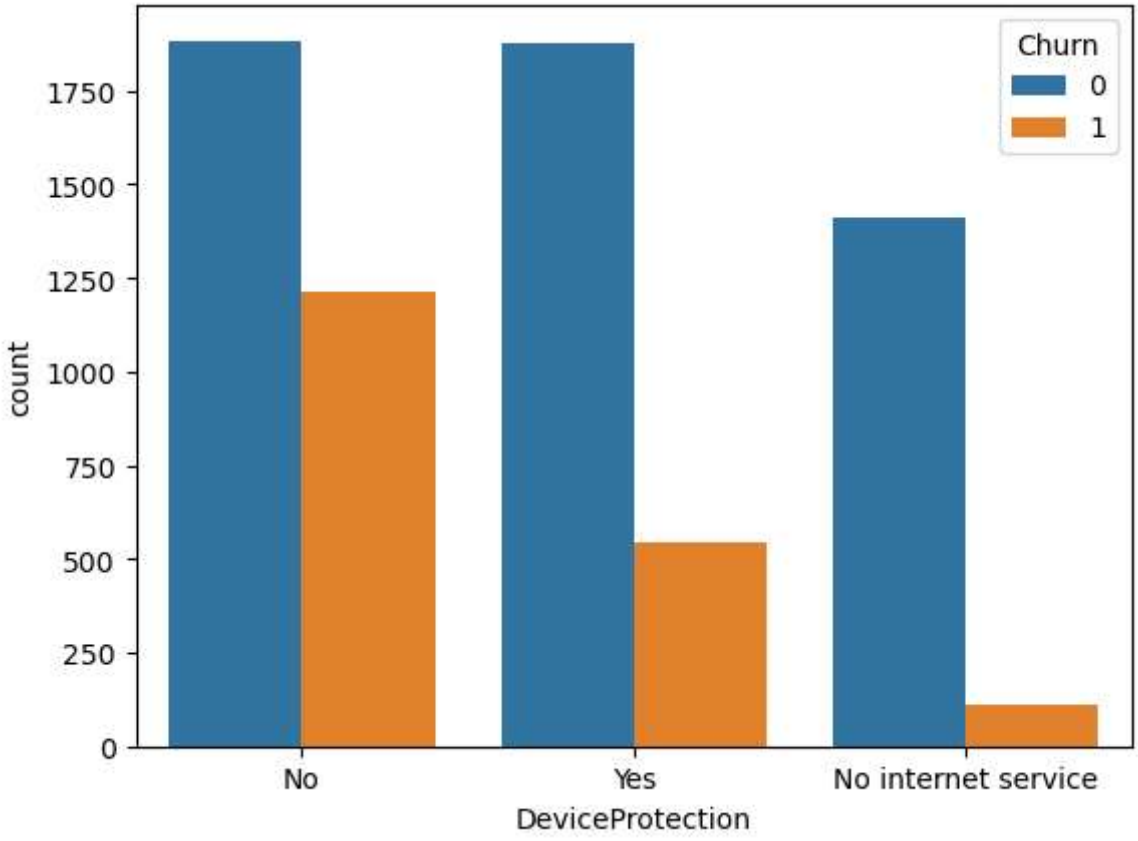

Churn
0
1
count
No
Yes
PhoneService

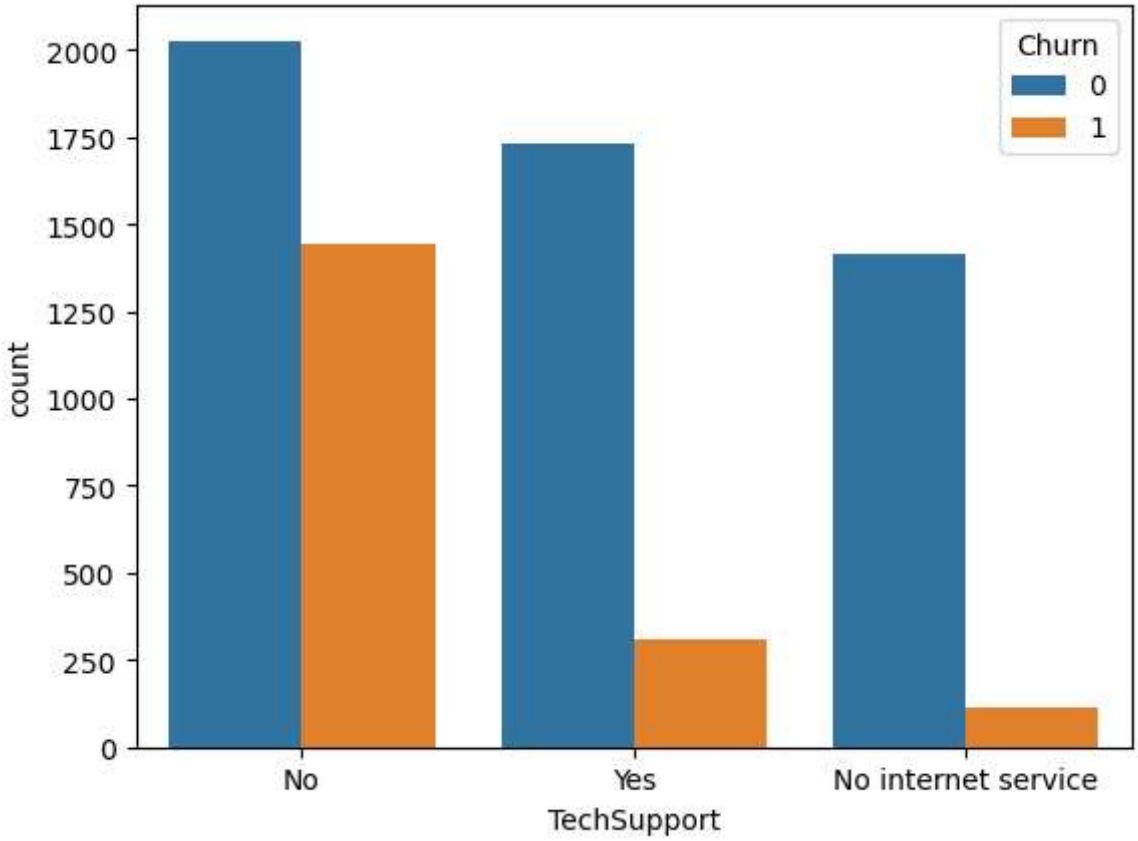

Churn
0
1
2500
2000
1500
1000
500
0
count
No phone service
No
Yes
MultipleLines

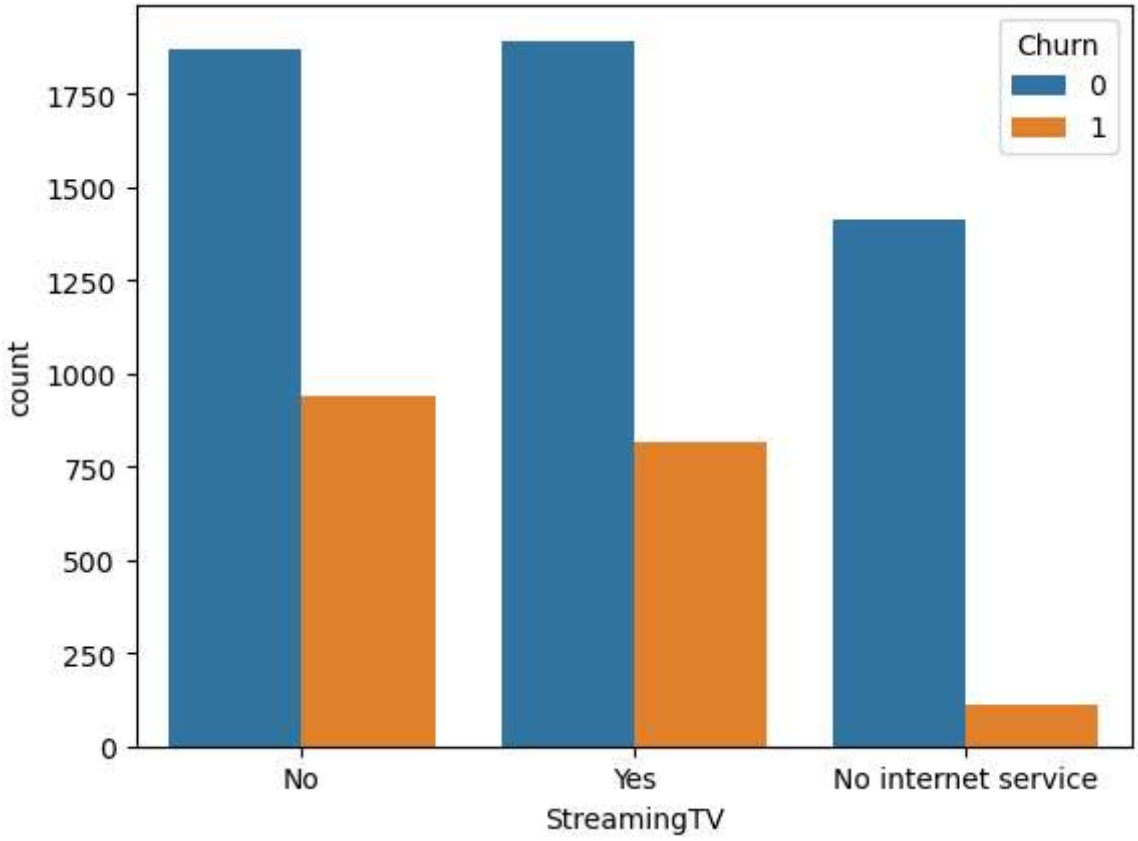

Churn
0
1
2000
1750
1500
1250
1000
750
500
250
0
count
DSL
Fiber optic
No
InternetService

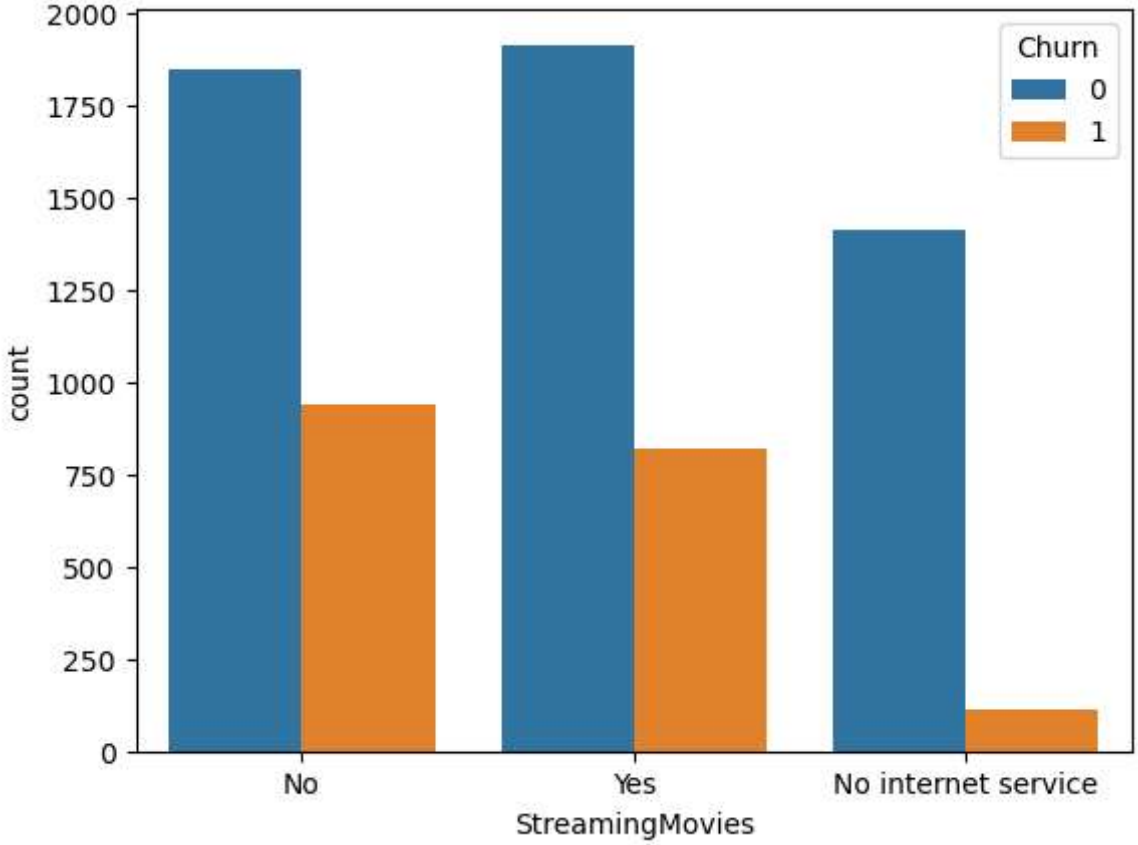

Churn
0
1
2000
1750
1500
1250
1000
750
500
250
0
count
No
Yes
No internet service
OnlineSecurity

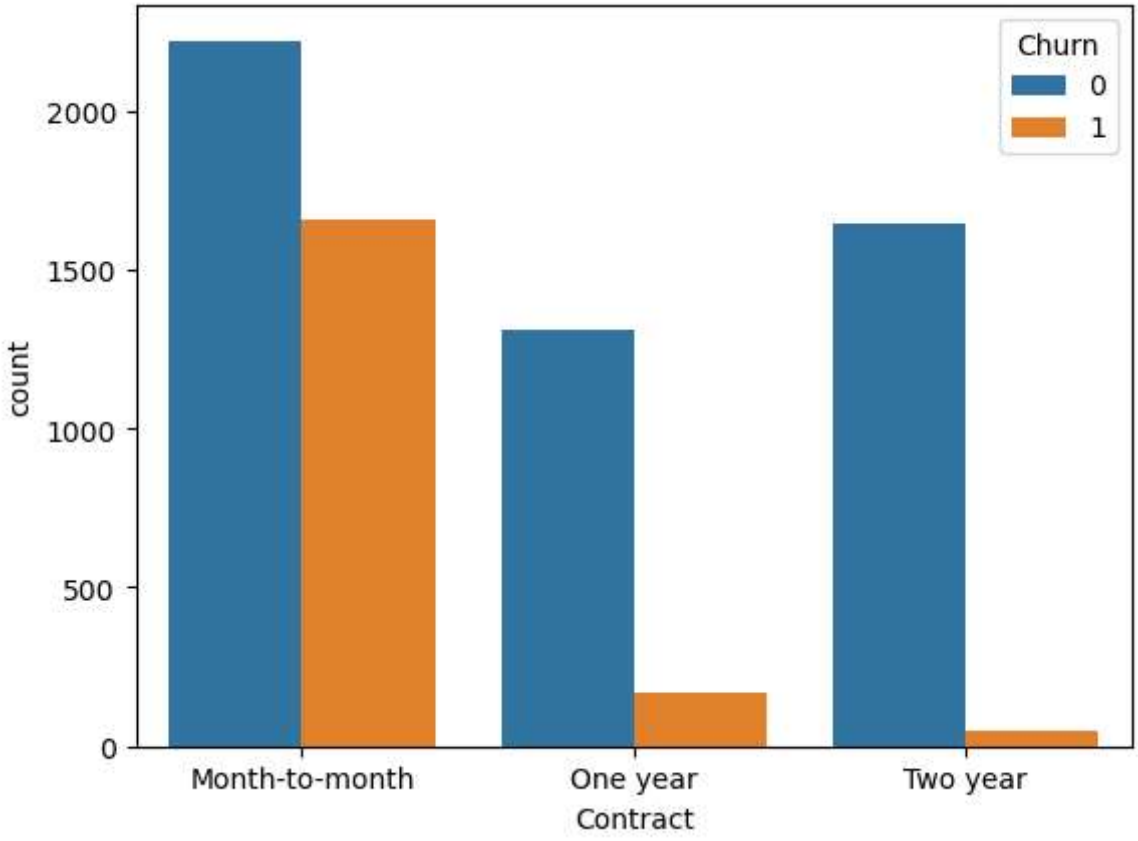

Churn
0
1
2000
1750
1500
1250
1000
750
500
250
0
count
Yes
No
No internet service
OnlineBackup

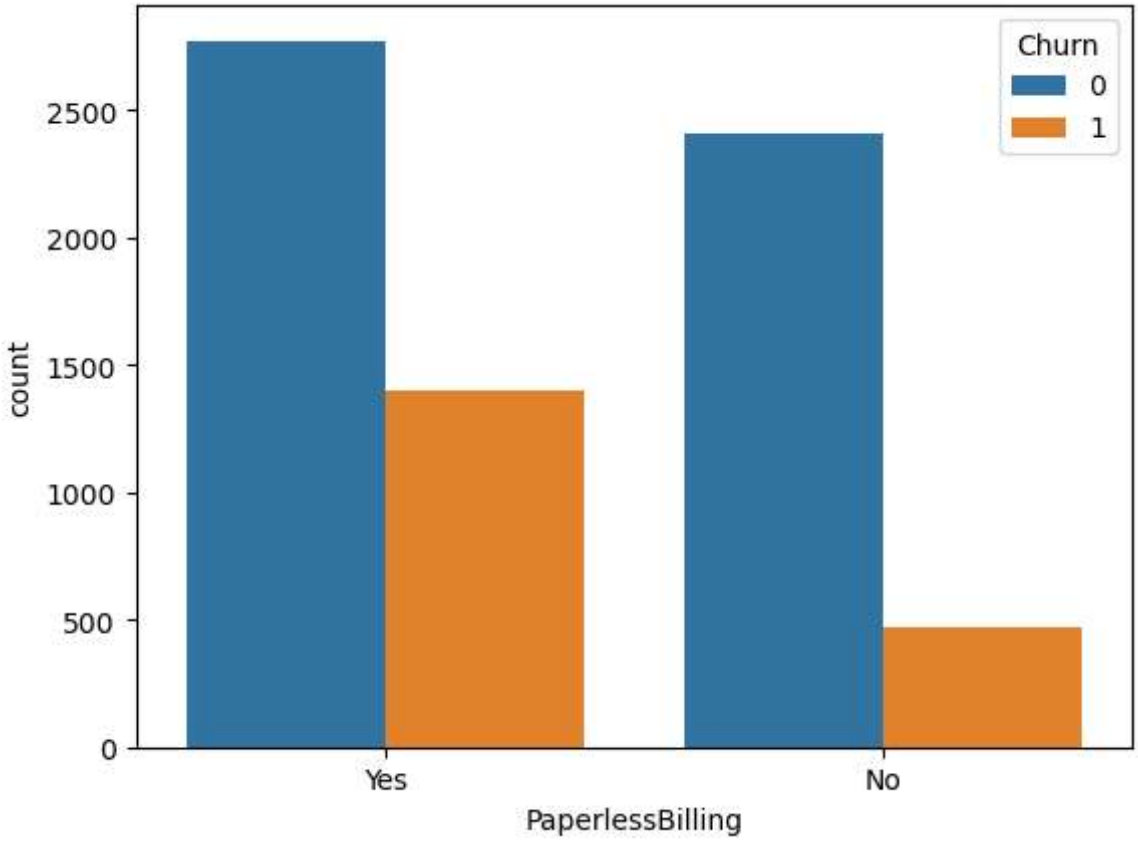

Churn
0
1
1750
1500
1250
1000
750
500
250
0
count
No
Yes
No internet service
DeviceProtection

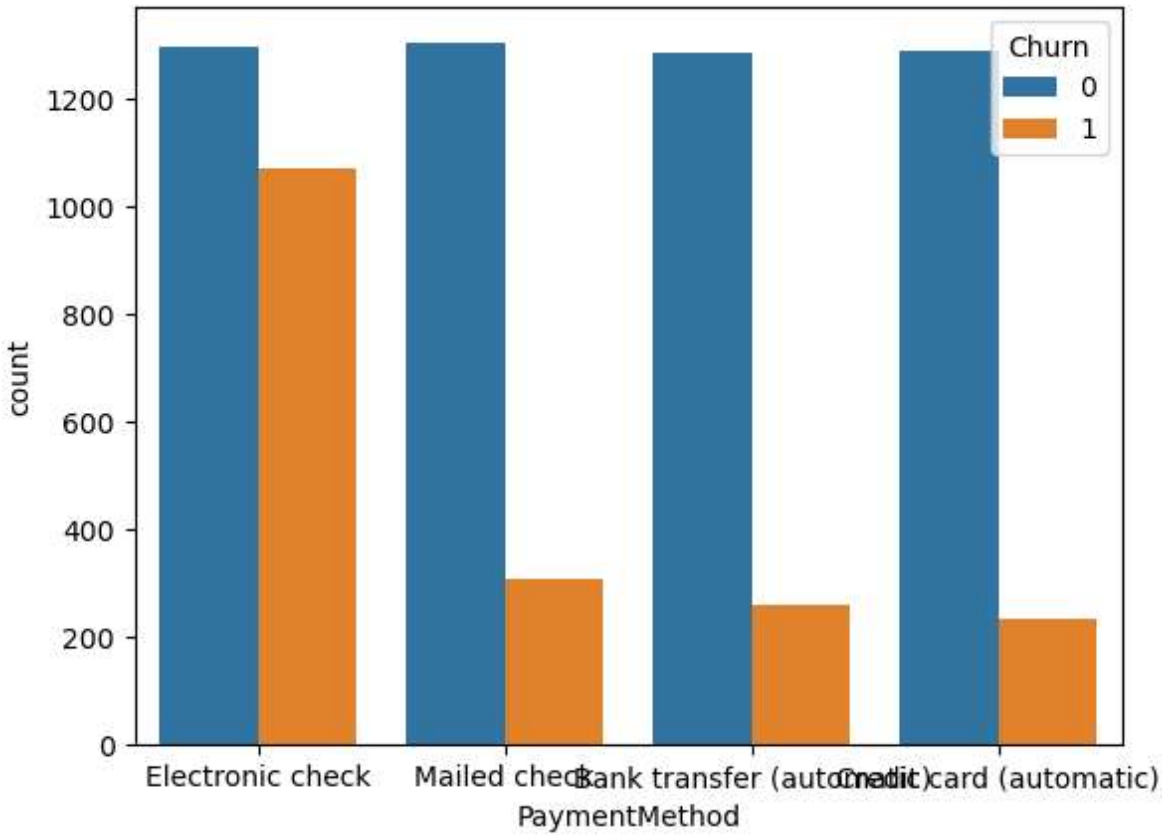

Churn
0
1
2000
1750
1500
1250
1000
750
500
250
0
count
No
Yes
No internet service
TechSupport

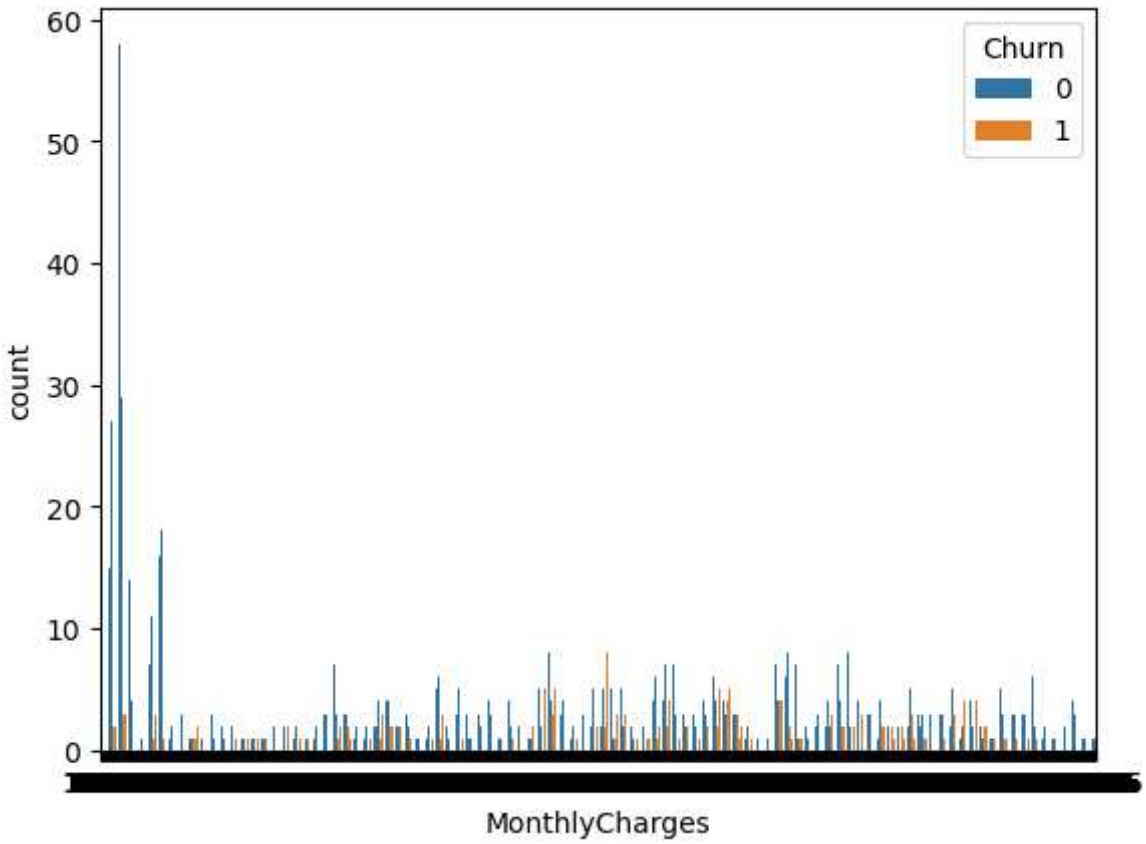

Churn
0
1
count
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
StreamingTV

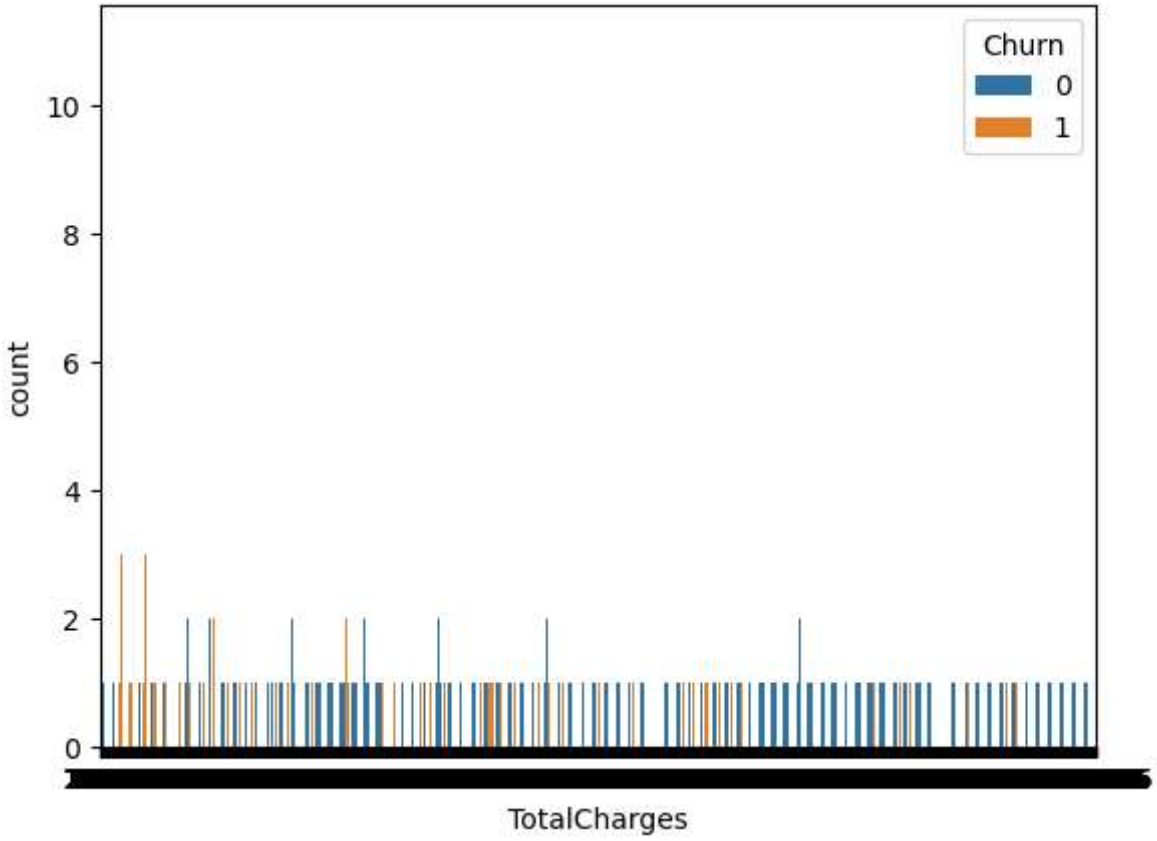

Churn
0
1
count
2000
1750
1500
1250
1000
750
500
250
0
No
Yes
No internet service
StreamingMovies

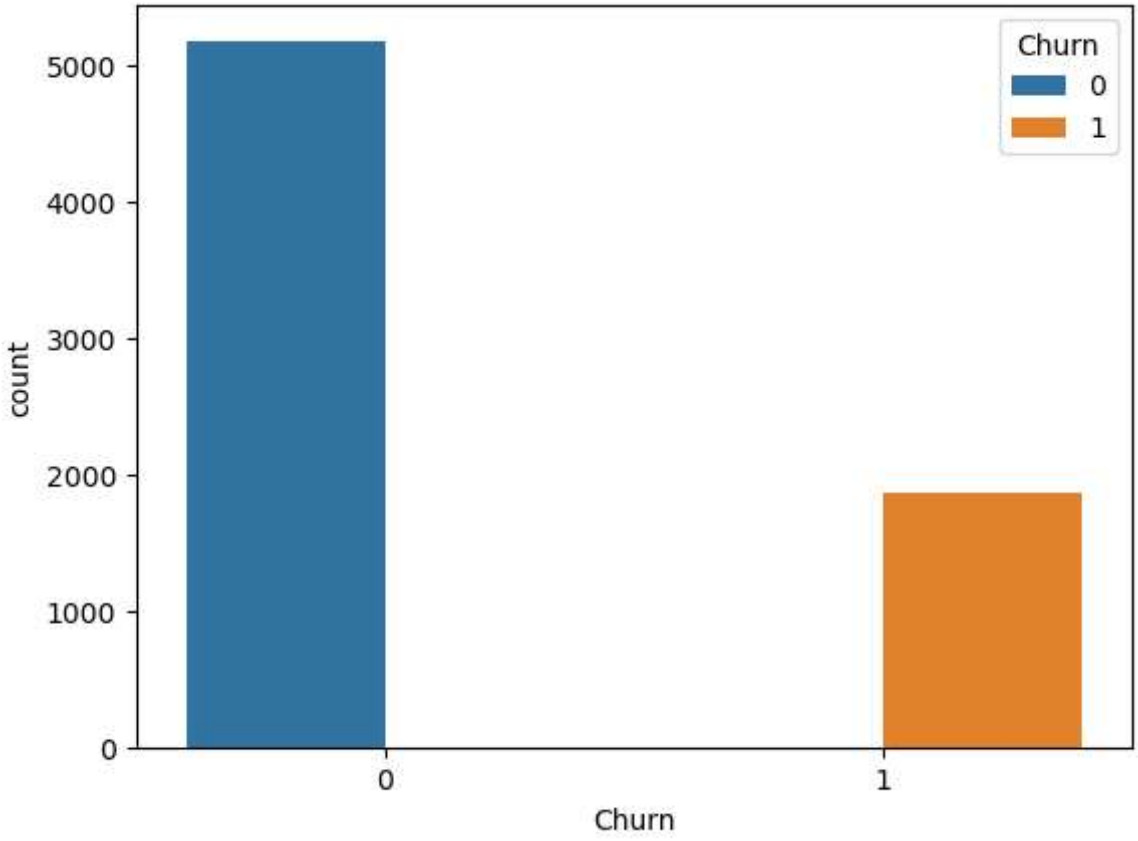

Churn
0
1
2000
1500
1000
500
0
count
Month-to-month
One year
Two year
Contract

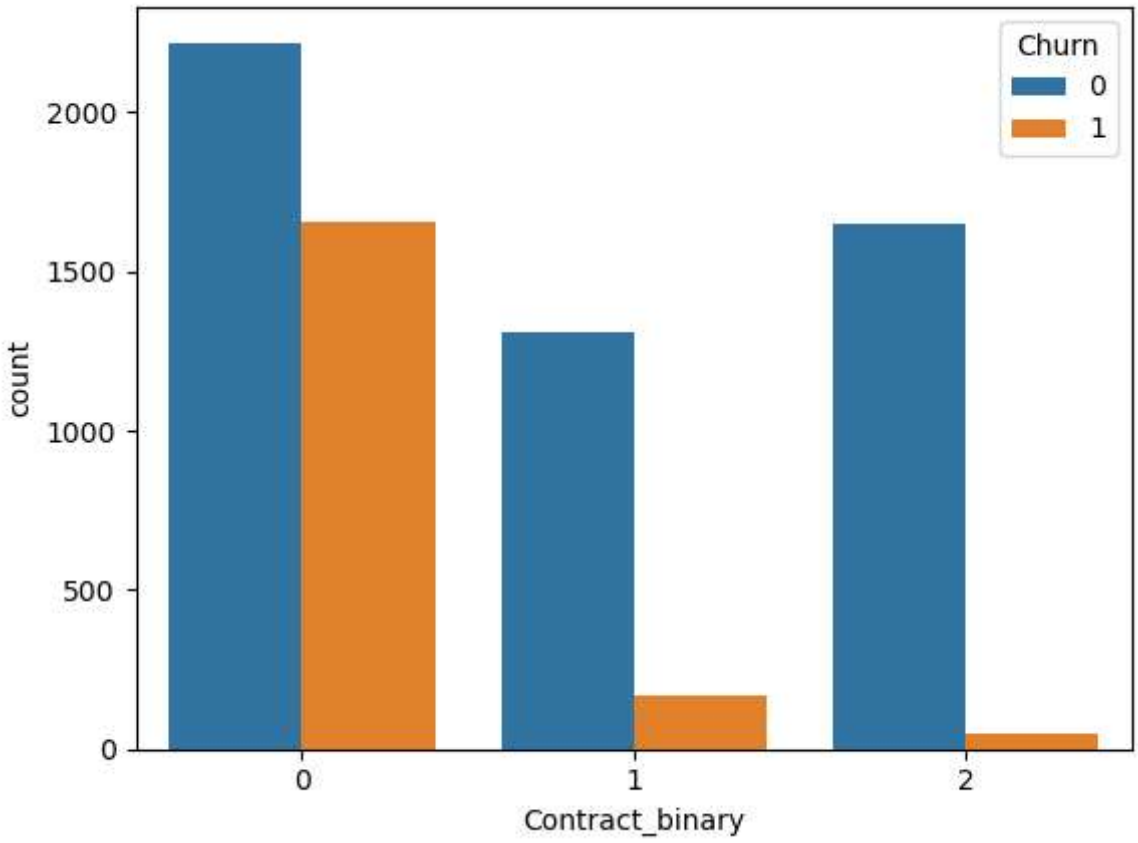

Churn
0
1
2500
2000
1500
1000
500
0
count
Yes
No
PaperlessBilling

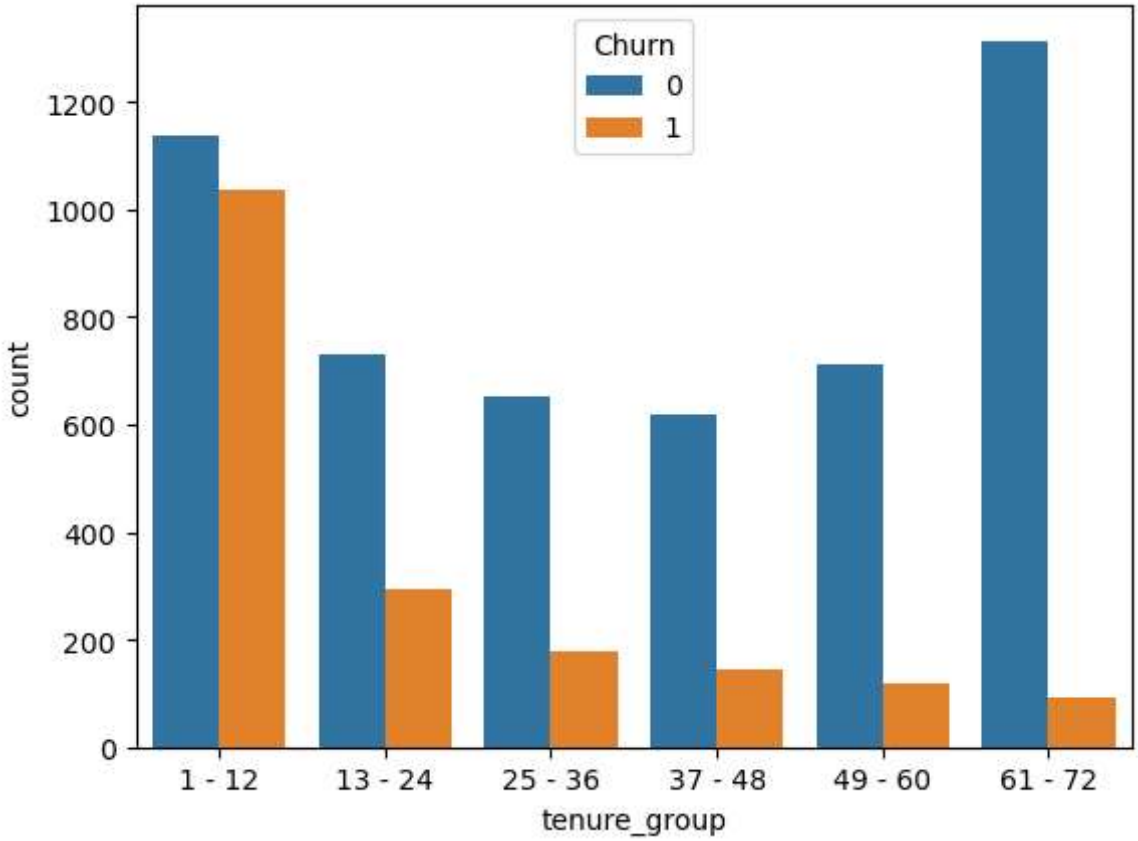

Churn
0
1
1200
1000
800
600
400
200
0
count
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod


Churn
0
1
60
50
40
30
20
10
0
count
MonthlyCharges


Churn
0
1
10
8
6
4
2
0
count
TotalCharges


Churn
0
1
5000
4000
3000
2000
1000
0
count
0
1
Churn


Churn
0
1
2000
1500
1000
500
0
count
0
1
2
Contract_binary


Churn
0
1
1200
1000
800
600
400
200
0
count
1 - 12
13 - 24
25 - 36
37 - 48
49 - 60
61 - 72
tenure_group