



ปัจจัยที่ส่งผลต่อการเลือกใช้บริการ ด้วย LOGISTIC REGRESSION

แรงบรรดาลใจในการเลือกหัวข้อนี้

เพื่อจะศึกษาพฤติกรรมลูกค้าว่ามีลักษณะเป็นเช่นไรและหาปัจจัยที่สำคัญที่ส่งผลต่อการเลิกใช้บริการ (Customer Churn)

Interested Questions

1

ลูกค้าที่อยู่กับเรา(เดือน)มีแนวโน้มที่จะเลิกใช้บริการหรือไม่

2

ลูกค้าที่มีค่าบริการรายเดือนต่ำ มีโอกาสที่จะเลิกใช้บริการหรือไม่

3

ลูกค้าที่เป็นผู้สูงอายุมีแนวโน้มที่จะเลิกใช้บริการหรือไม่

4

ลูกค้าที่มีผู้อุปการะมีแนวโน้มที่จะเลิกใช้บริการหรือไม่

5

ลูกค้าที่มีหลาย Line มีแนวโน้มที่จะเลิกใช้บริการหรือไม่

ข้อมูลที่เกี่ยวข้องในการใช้ตอบคำถามที่สนใจ

1

tenure จำนวนเดือนที่
ลูกค้าอยู่กับบริษัท

2

MonthlyCharges: จำนวนเงิน
ที่เรียกเก็บจากลูกค้ารายเดือน

3

SeniorCitizen: ลูกค้าเป็น
ผู้สูงอายุหรือไม่ (1,0)

4

Dependents ลูกค้ามีผู้อยู่
ในอุปการะหรือไม่ (ใช่,ไม่ใช่)

5

MultipleLines ลูกค้ามีหลายLinesหรือ
ไม่ (ใช่,ไม่ใช่,ไม่มีโทรศัพท์)

ข้อมูลที่จะนำมาวิเคราะห์

ตัวแปร

customer ID: รหัสลูกค้า

gender: ลูกค้าจะเป็นชายหรือหญิง(0,1)

SeniorCitizen: ลูกค้าเป็นผู้สูงอายุหรือไม่ (1, 0)

Partner: ลูกค้าจะมีหุ้นส่วนหรือไม่ (ใช่,ไม่ใช่)

Dependents: ลูกค้ามีผู้อยู่ในอุปการะหรือไม่ (ใช่,ไม่ใช่)

tenure: จำนวนเดือนที่ลูกค้าอยู่กับบริษัท

PhoneService: ลูกค้าจะมีบริการโทรศัพท์หรือไม่ (ใช่,ไม่ใช่)

MultipleLines: ลูกค้ามีหลายLinesหรือไม่ (ใช่,ไม่ใช่,ไม่มีโทรศัพท์)

InternetService: ผู้ให้บริการอินเทอร์เน็ตของลูกค้า (DSL, Fiber optic, No)

ข้อมูลที่จะนำมาวิเคราะห์

ตัวแปร

OnlineSecurity: ลูกค้ามีความปลอดภัยทางออนไลน์หรือไม่ (ใช่, ไม่ใช่, ไม่มีบริการอินเทอร์เน็ต)

OnlineBackup: ลูกค้ามีการสำรองข้อมูลออนไลน์หรือไม่ (ใช่, ไม่ใช่, ไม่มีบริการอินเทอร์เน็ต)

DeviceProtection: ลูกค้าจะมีการป้องกันอุปกรณ์หรือไม่ (ใช่, ไม่ใช่, ไม่มีบริการอินเทอร์เน็ต)

TechSupport: ลูกค้ามีการสนับสนุนด้านเทคนิคหรือไม่ (ใช่, ไม่ใช่, ไม่มีบริการอินเทอร์เน็ต)

StreamingTV: ลูกค้าจะสตรีมทีวีหรือไม่ (ใช่, ไม่ใช่, ไม่มีบริการอินเทอร์เน็ต)

ข้อมูลที่จะนำมาวิเคราะห์

ตัวแปร

StreamingMovies: ลูกค้ามีการสตรีมภาพยนตร์หรือไม่ (ใช่,ไม่ใช่,ไม่มีบริการอินเทอร์เน็ต)

Contract : ระยะเวลาของลูกค้า (เดือนต่อเดือนหนึ่งปีสองปี)

PaperlessBilling: ลูกค้ามีการเรียกเก็บเงินแบบไม่ใช้กระดาษหรือไม่ (ใช่,ไม่ใช่)

PaymentMethod: วิธีการชำระเงินของลูกค้า (เช็คอิเล็กทรอนิกส์,เช็คไปรษณีย์โอนเงิน,ผ่านธนาคาร (อัตโนมัต) ,บัตรเครดิต (อัตโนมัต))

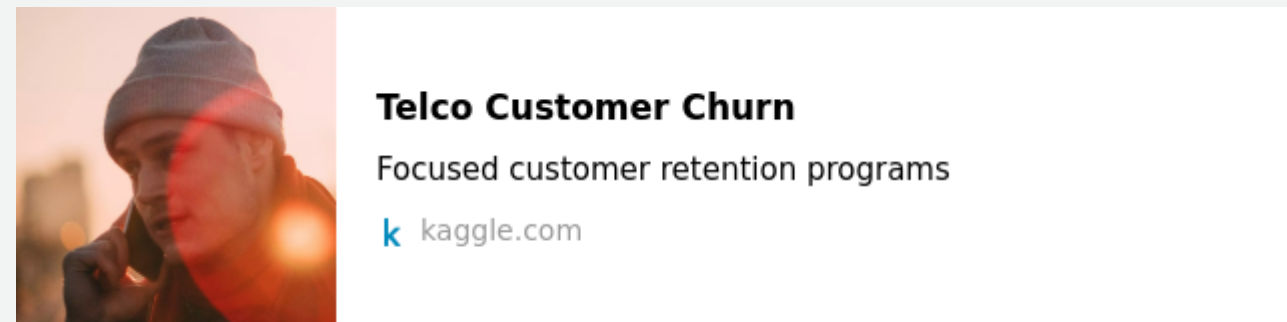
MonthlyCharges: จำนวนเงินที่เรียกเก็บจากลูกค้ารายเดือน

TotalCharges: จำนวนเงินทั้งหมดที่เรียกเก็บจากลูกค้า

Churn: ลูกค้าจะเลิกใช้บริการหรือไม่ (ใช่,ไม่)

ข้อมูลที่จะนำมาวิเคราะห์

แหล่งที่มาของข้อมูล



ที่มา <https://www.kaggle.com/blastchar/telco-customer-churn/notebooks>

Data acquisition process

```
▶ import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="ticks")
from scipy import stats
from scipy.stats import mannwhitneyu
from scipy.stats import shapiro
from pycaret.classification import *
```

Data acquisition process

```
from google.colab import drive
drive.mount('/content/drive')

#df = pd.read_csv('/content/drive/My Drive/Project/WA_Fn-UseC_-Telco-Customer-Churn.csv')
df = pd.read_csv('/content/drive/My Drive/WA_Fn-UseC_-Telco-Customer-Churn.csv')
df.head()
```

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

Data acquisition process

DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
Yes	No	No	No	One year	No	Mailed check	56.95	1889.5	No
No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

Data cleaning process

```
In [6]: 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
```

Data cleaning process

```
In [7]: df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce', downcast='float')
df['SeniorCitizen'] = df['SeniorCitizen'].astype(np.object)
```

```
In [8]: 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  object 
 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           7032 non-null  float32 
20  Churn                  7043 non-null  object 
dtypes: float32(1), float64(1), int64(1), object(18)
memory usage: 1.1+ MB
```

Data cleaning process

```
In [9]: df.isnull().sum()
```

```
Out[9]: 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
```

```
In [10]: df['TotalCharges'].fillna(value=df['TotalCharges'].mean(), inplace=True)
df.isnull().sum()
```

```
Out[10]: 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
```

Explore Global Properties

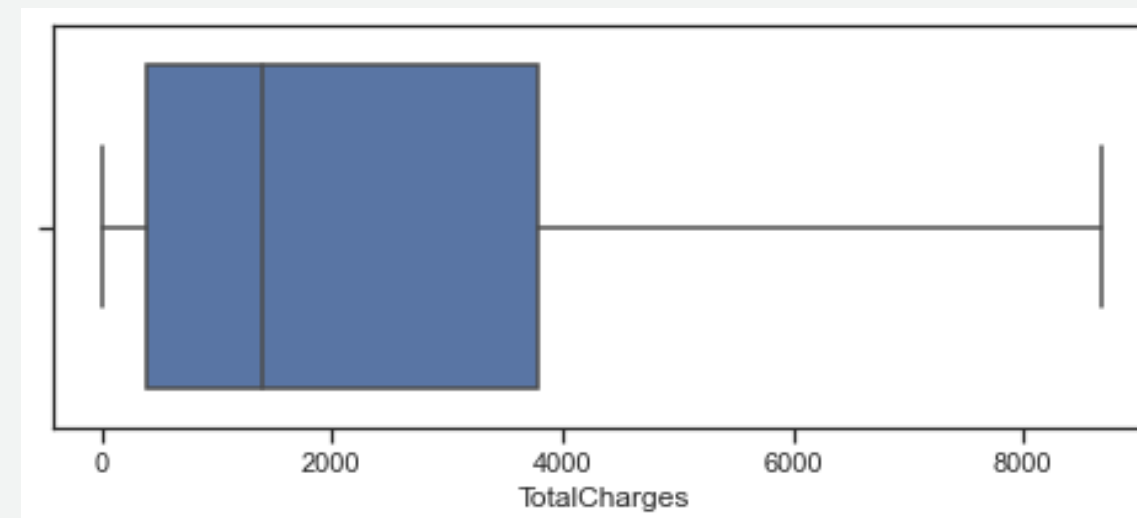
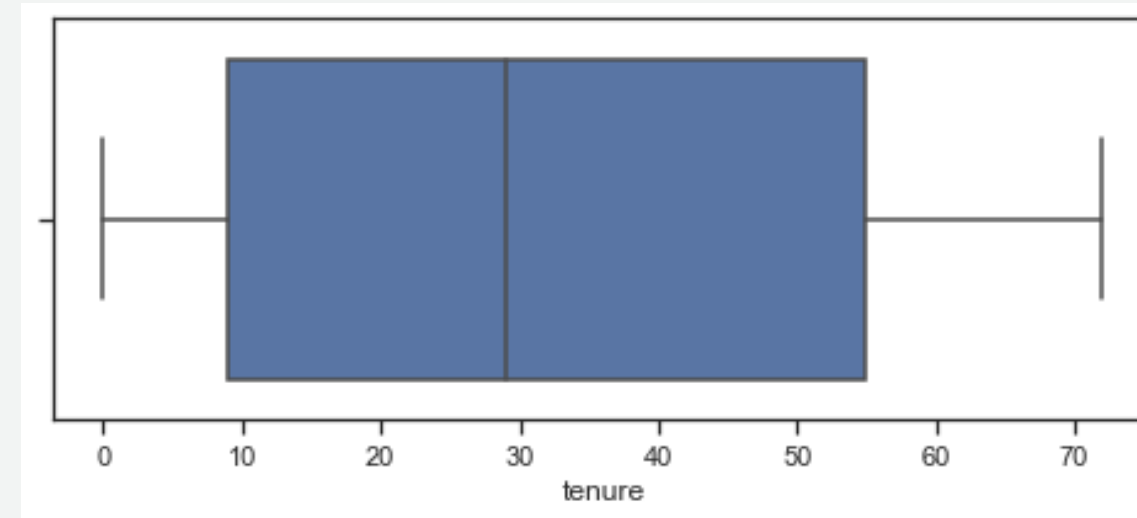
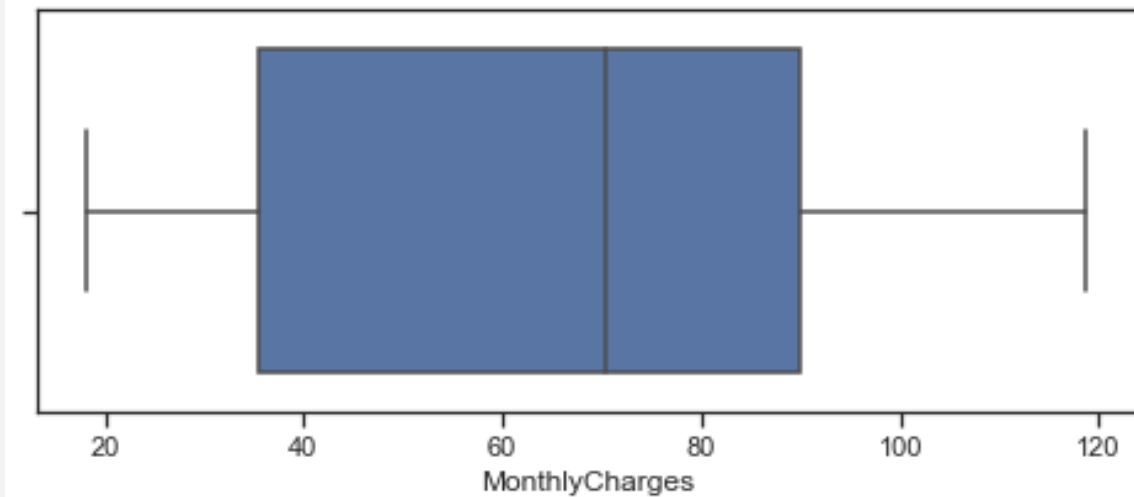
```
In [11]: df.describe()
```

```
Out[11]:
```

	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000
mean	32.371149	64.761692	2283.300537
std	24.559481	30.090047	2265.000244
min	0.000000	18.250000	18.799999
25%	9.000000	35.500000	402.225006
50%	29.000000	70.350000	1400.550049
75%	55.000000	89.850000	3786.599976
max	72.000000	118.750000	8684.799805

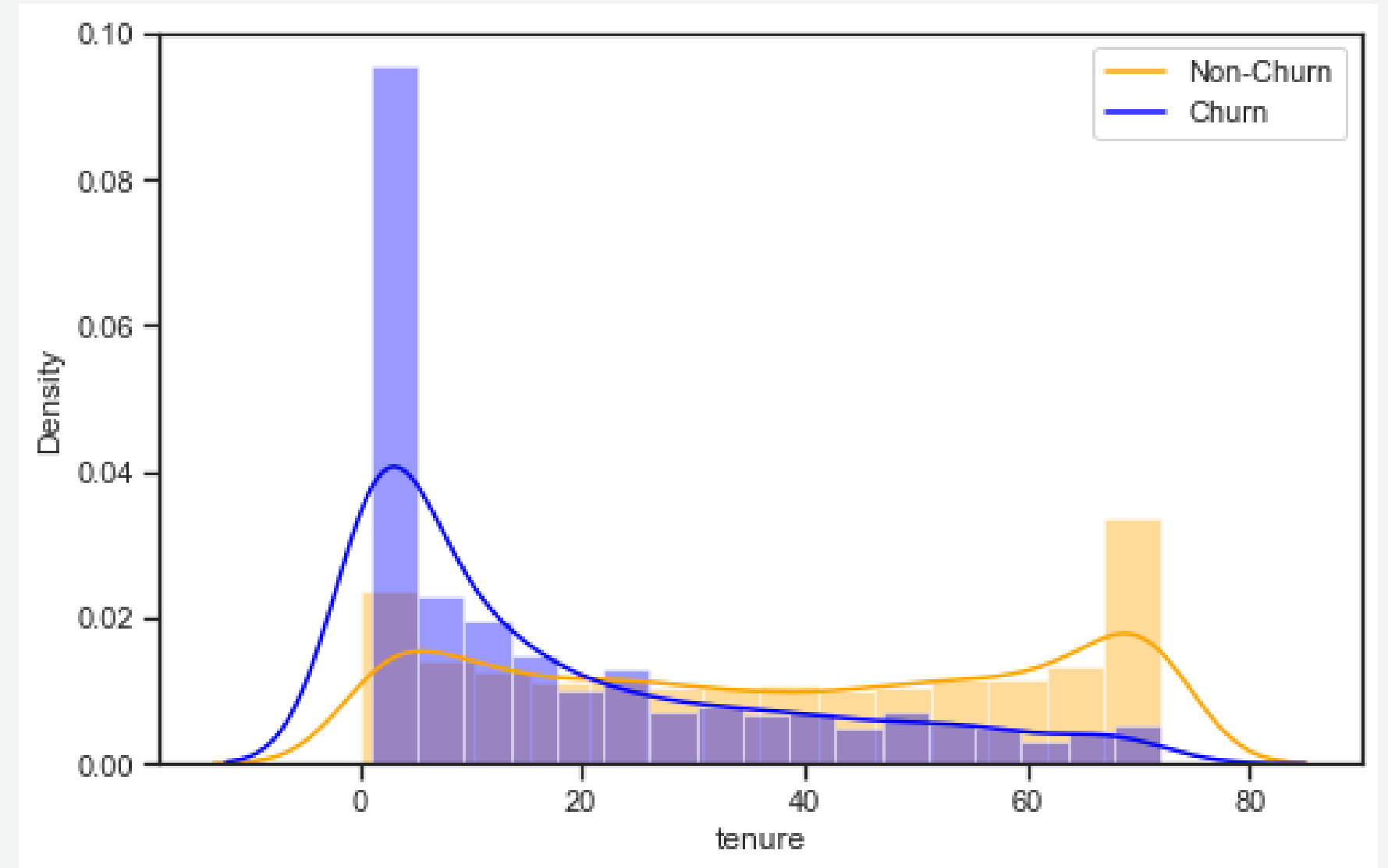
Explore Global Properties

```
In [13]: for col in ['tenure', 'MonthlyCharges', 'TotalCharges']:  
         fig = plt.figure(figsize=(8,3))  
         sns.boxplot(df[col])
```

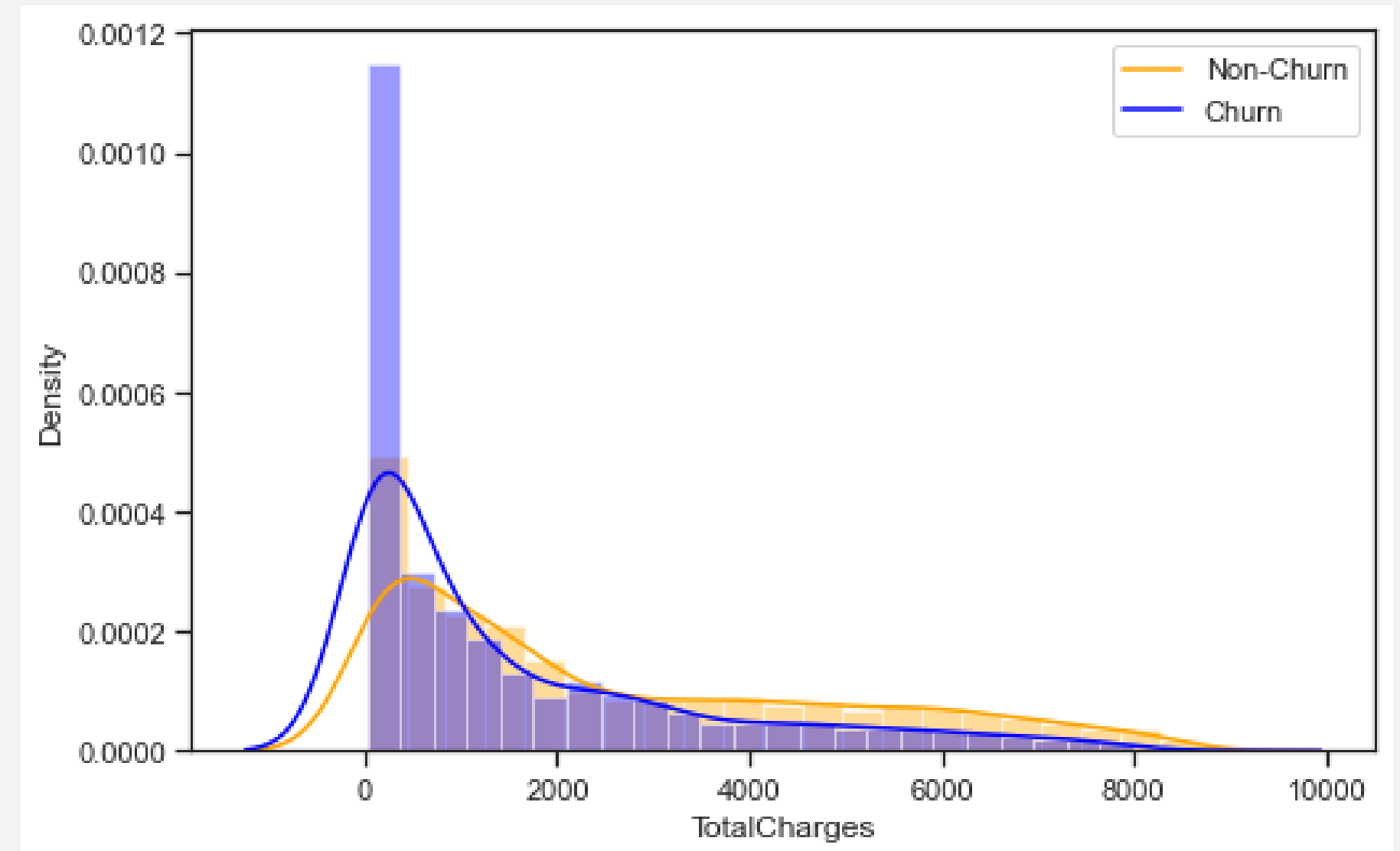
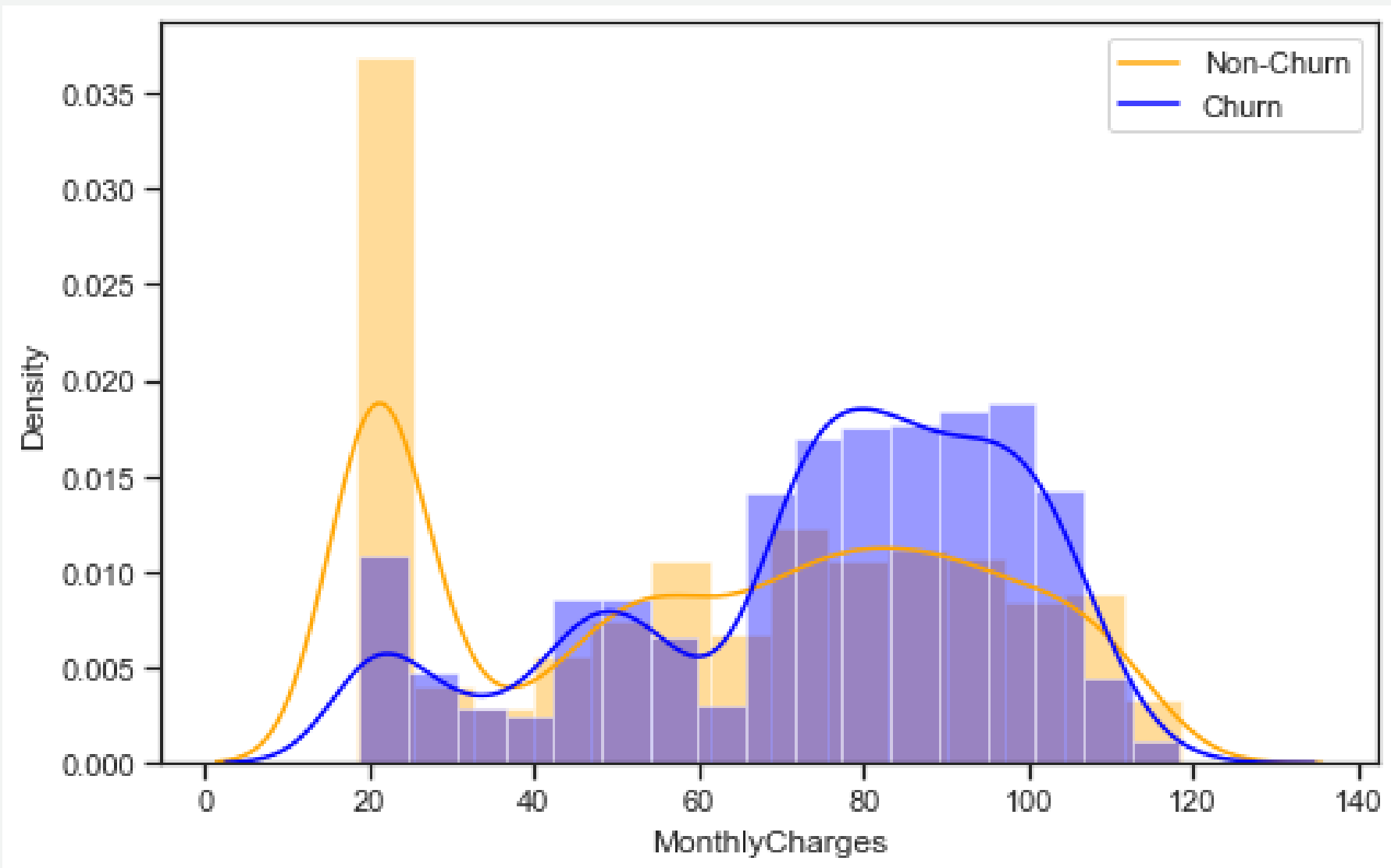


Explore Global Properties

```
In [17]: labels = 'Non-Churn', 'Churn'
for col in ['tenure', 'MonthlyCharges', 'TotalCharges']:
    fig = plt.figure(figsize=(8,5))
    sns.distplot(df[df.Churn == 'No'][col],
                  color='orange',
                  label='Non-Churn',
                  kde=True)
    sns.distplot(df[df.Churn == 'Yes'][col],
                  color='blue',|
                  label='Churn',
                  kde=True)
    plt.legend(labels)
```



Explore Global Properties



Explore Global Properties

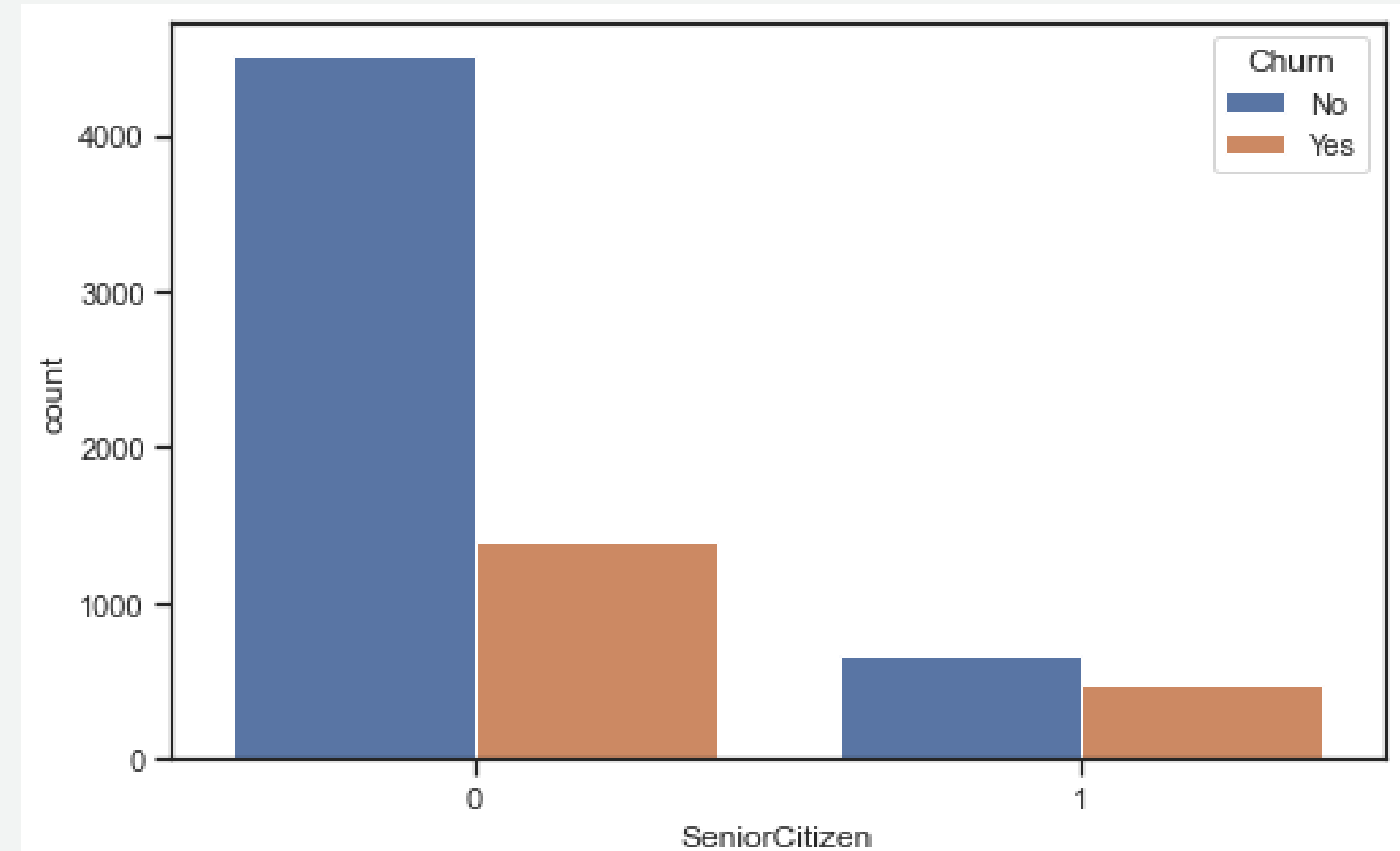
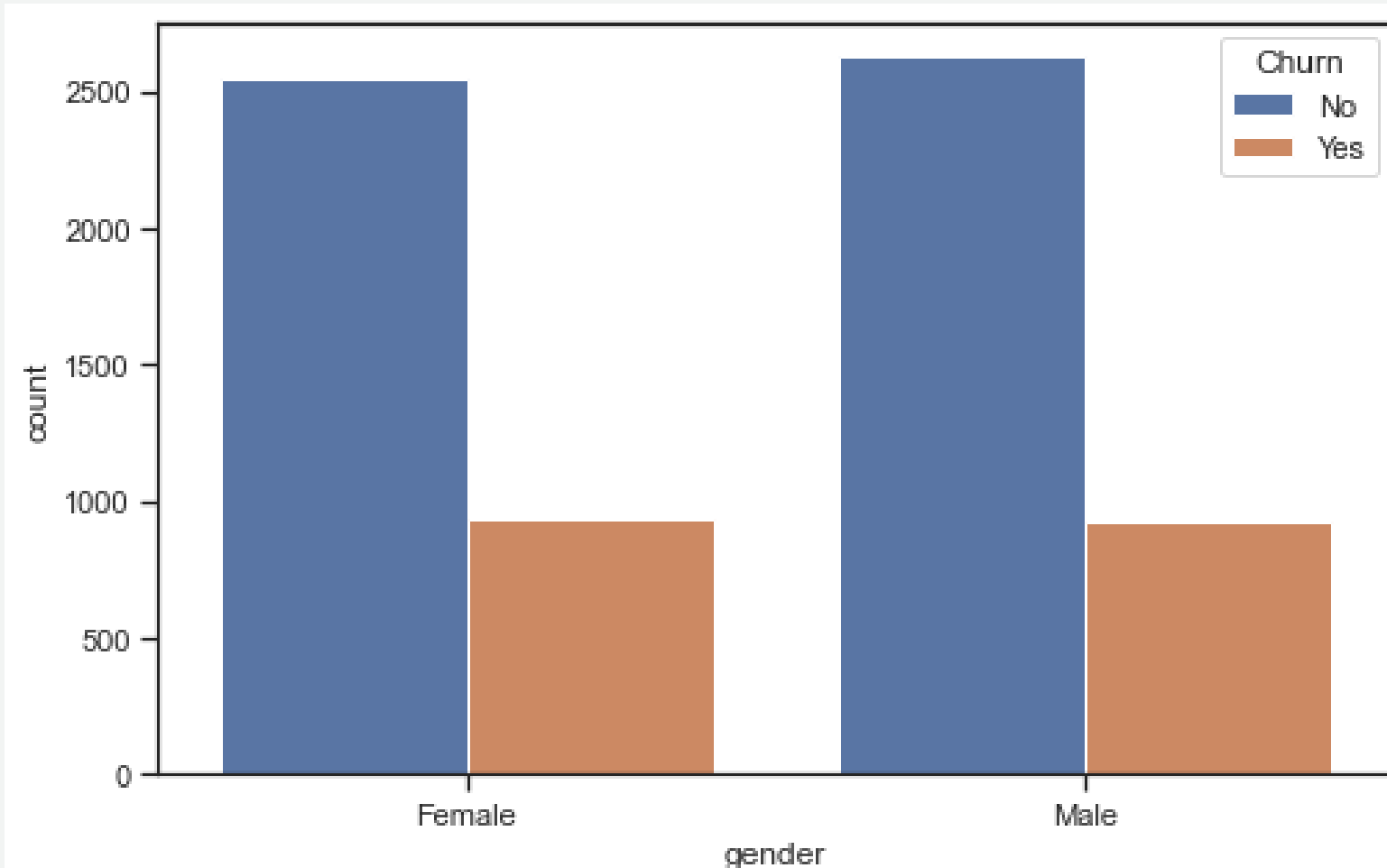
```
In [12]: df.describe(include=[object] )
```

	customerID	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup
count	7043	7043	7043	7043	7043	7043	7043	7043	7043	7043
unique	7043	2	2	2	2	2	3	3	3	3
top	2223-KAGMX	Male	0	No	No	Yes	No	Fiber optic	No	No
freq	1	3555	5901	3641	4933	6361	3390	3096	3498	3088

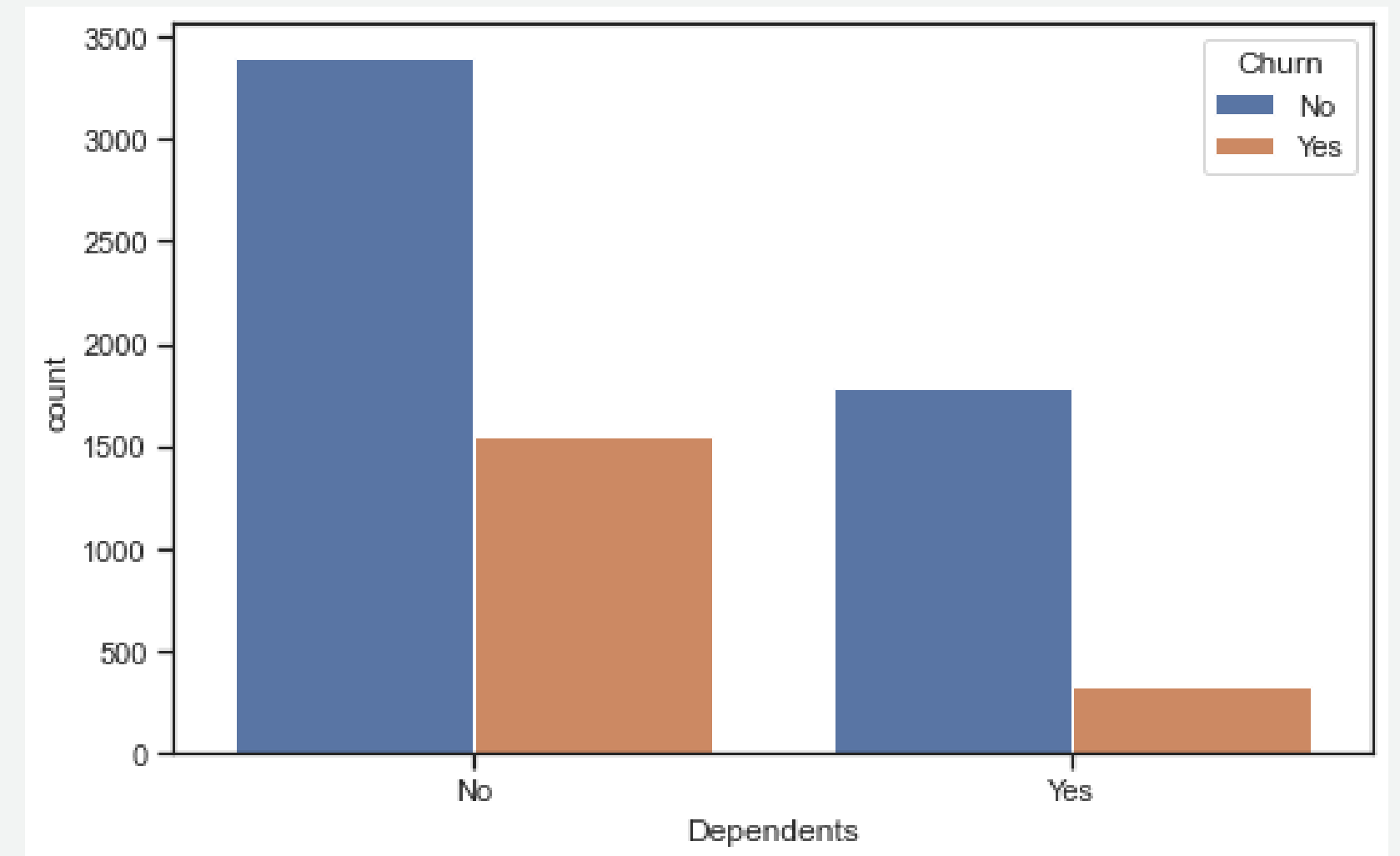
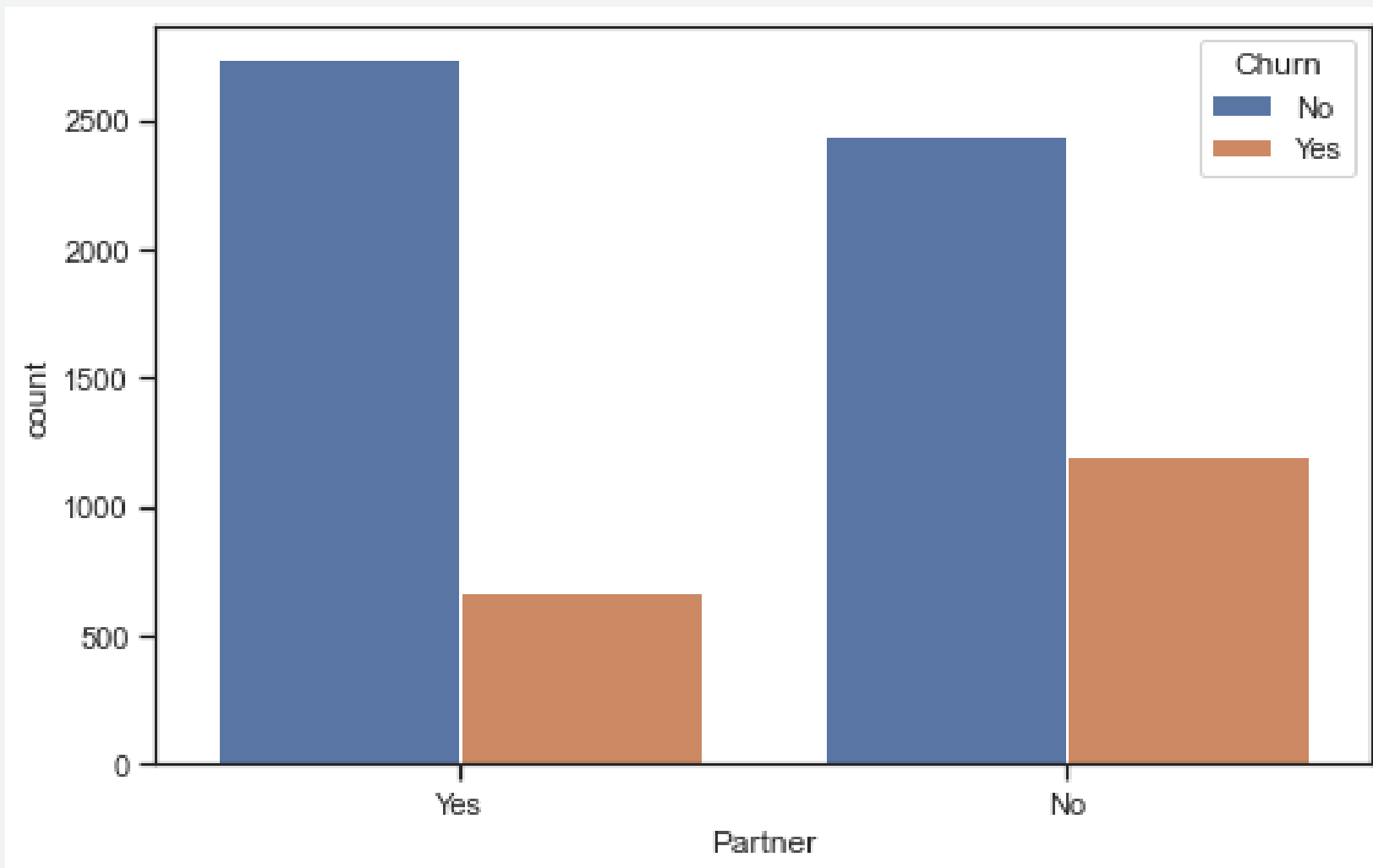
DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	Churn
7043	7043	7043	7043	7043	7043	7043	7043
3	3	3	3	3	2	4	2
No	No	No	No	Month-to-month	Yes	Electronic check	No
3095	3473	2810	2785	3875	4171	2365	5174

Explore Global Properties

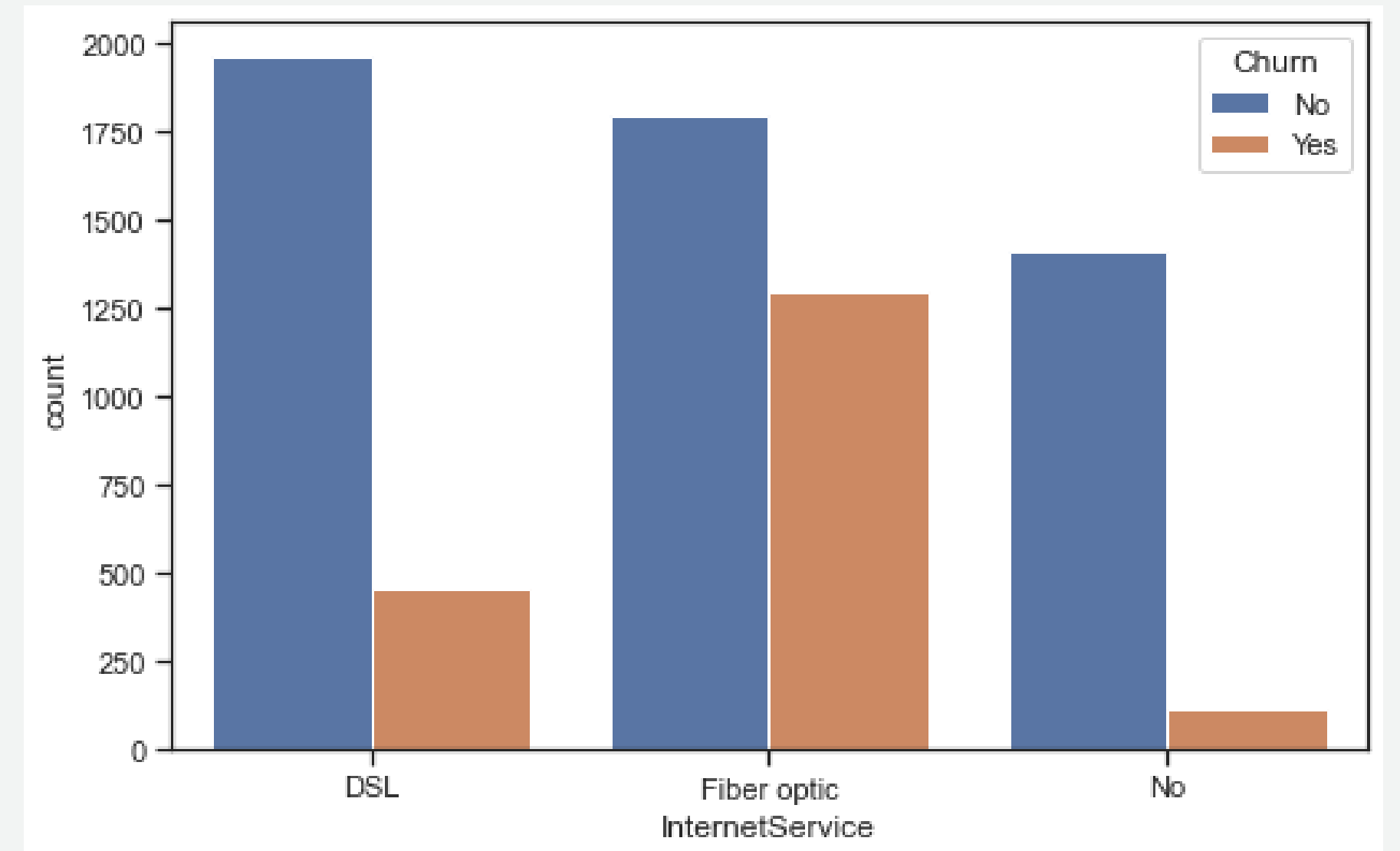
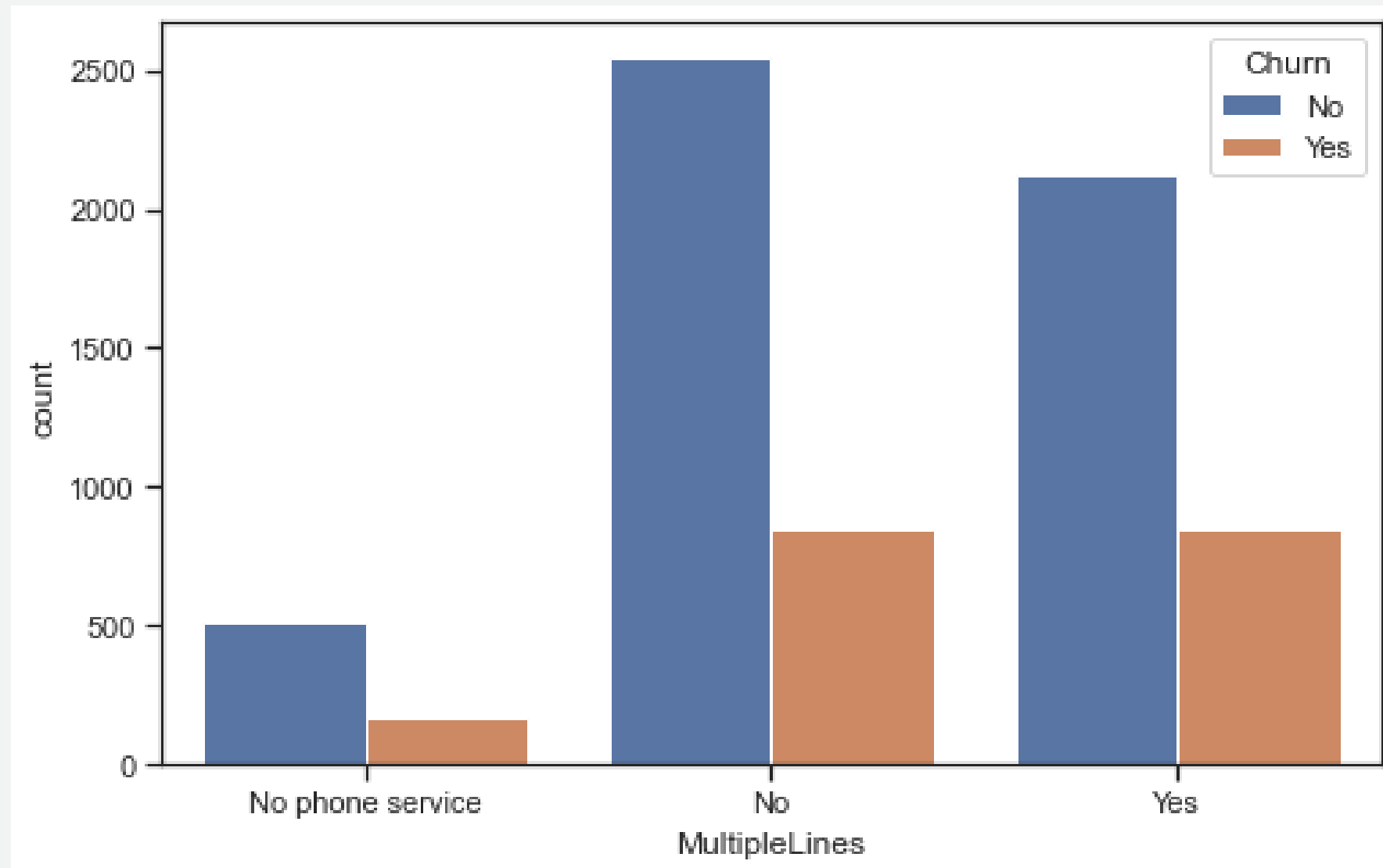
```
In [14]: for col in ['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'MultipleLines',  
                  , 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',  
                  'TechSupport', 'StreamingTV', 'StreamingMovies', 'PaperlessBilling', 'PaymentMethod', 'Contract']:  
    plt.figure(figsize=(8,5))  
    sns.countplot(x=col, hue='Churn', data=df)  
    plt.show()
```



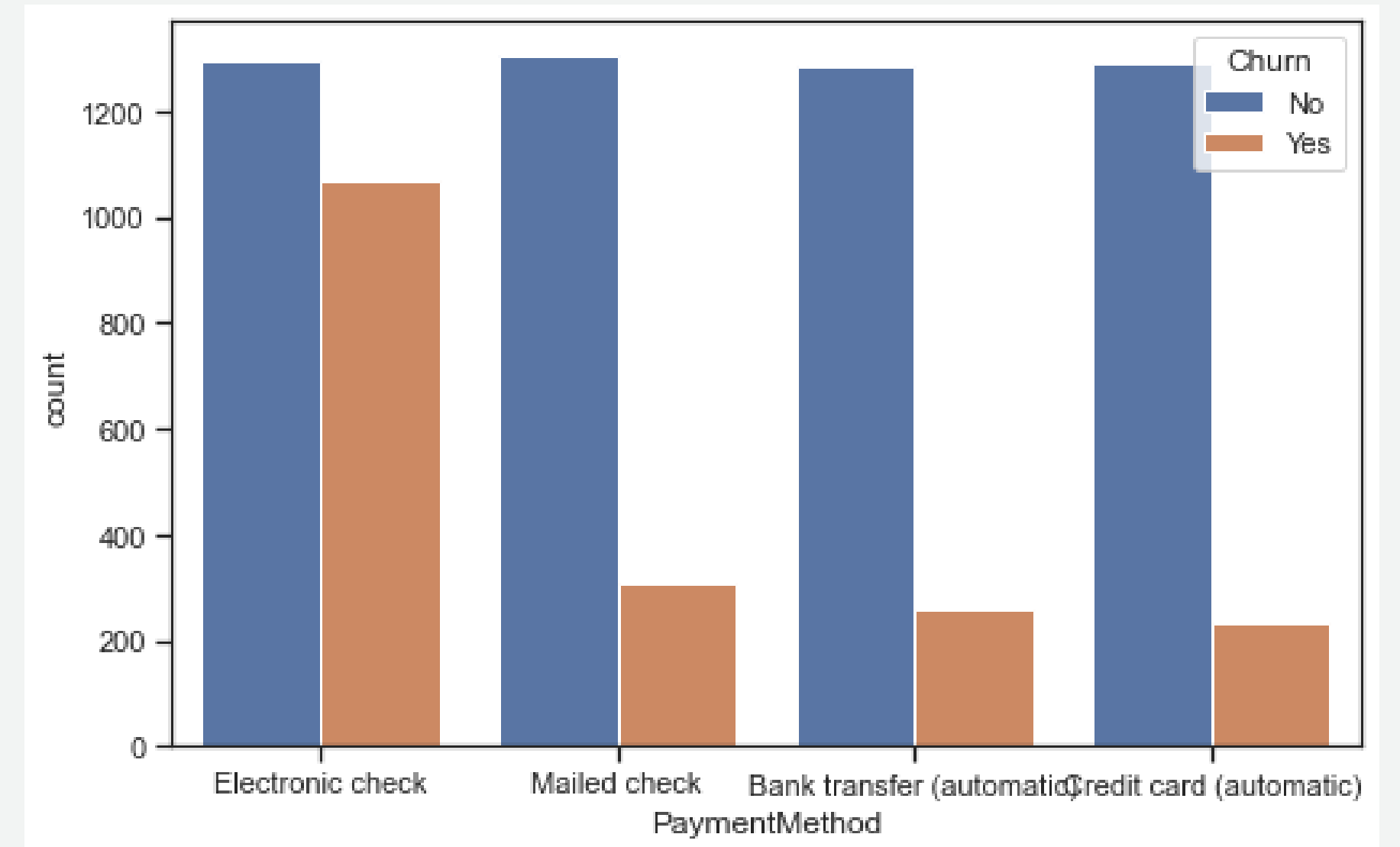
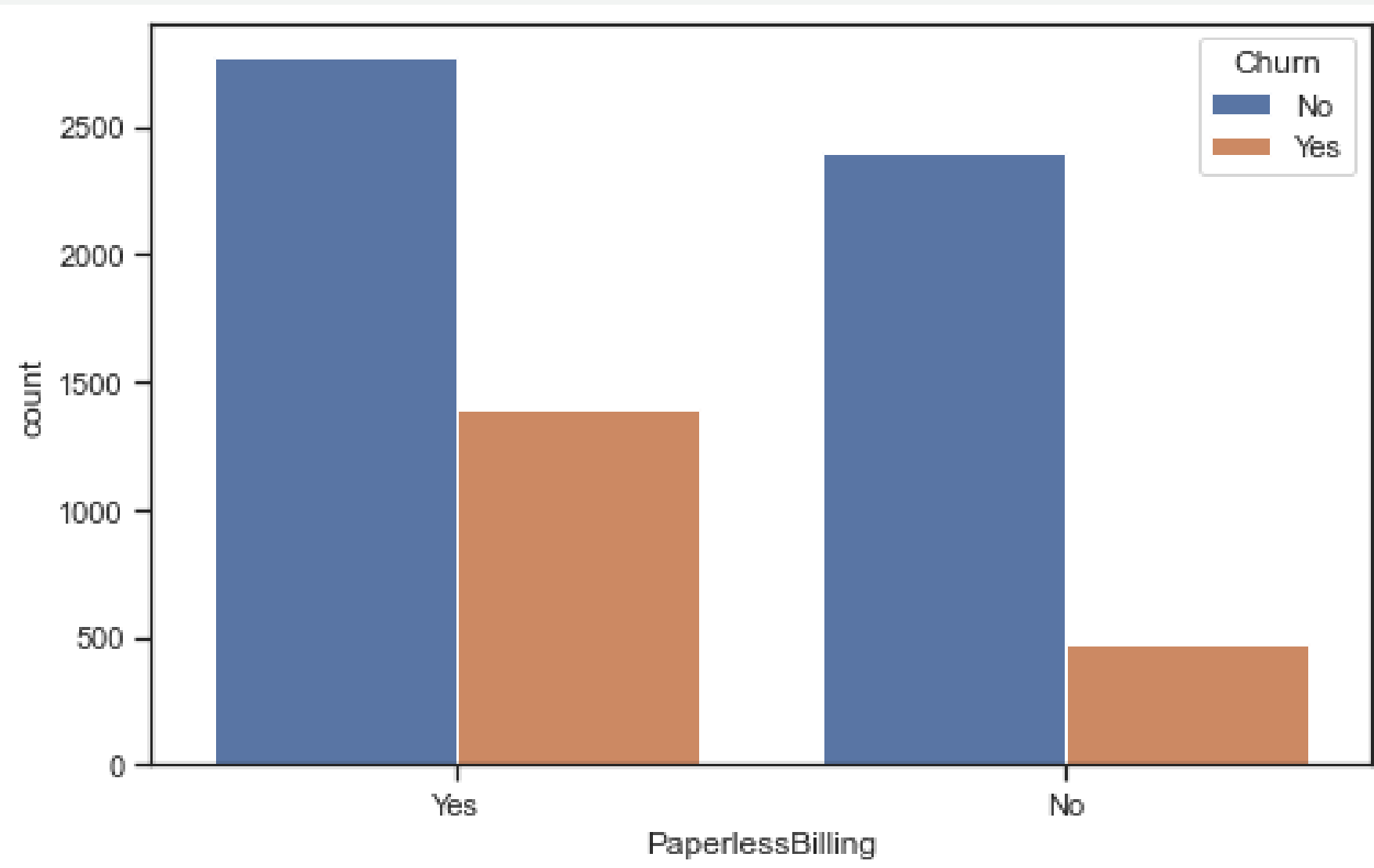
Explore Global Properties



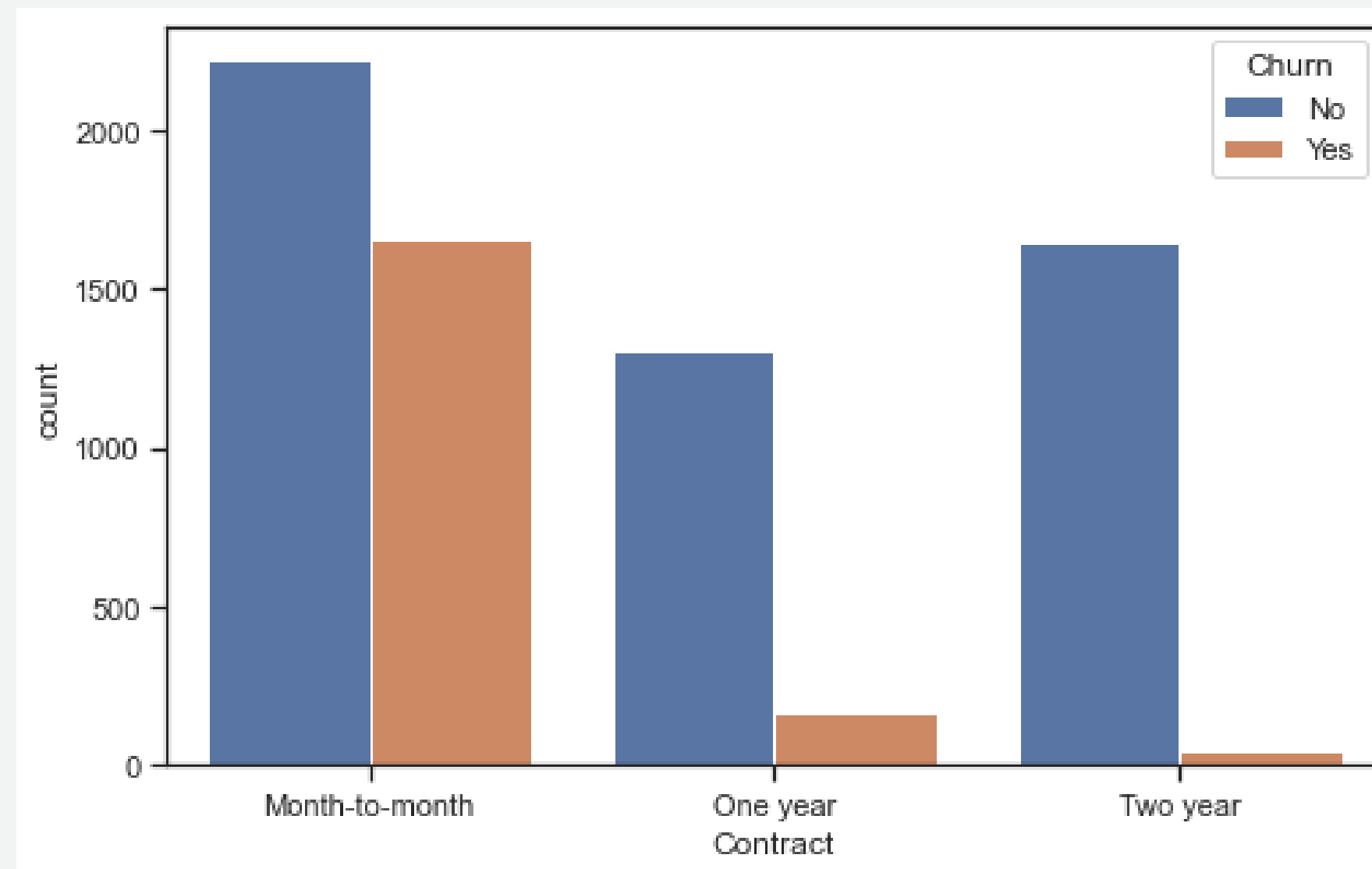
Explore Global Properties



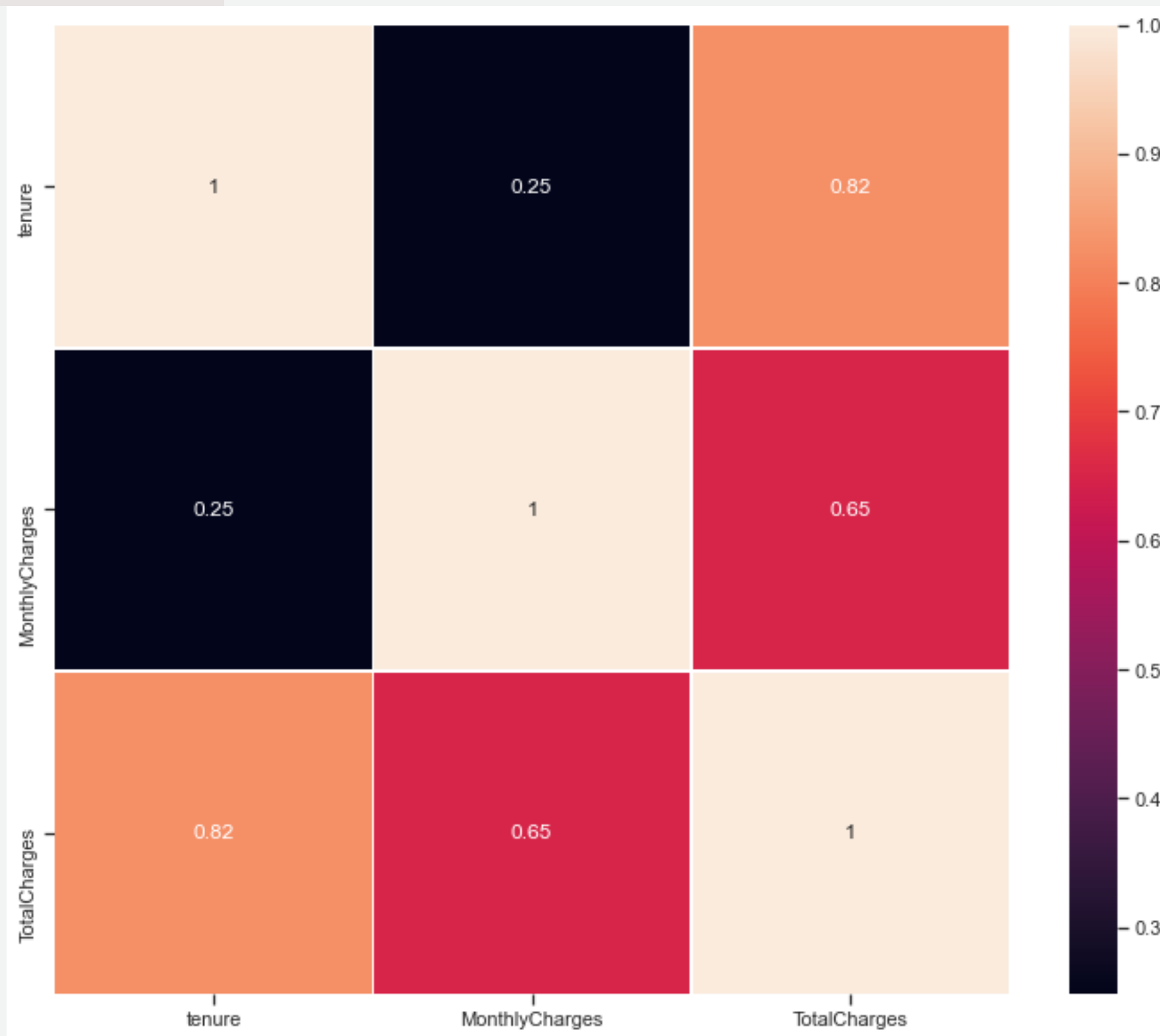
Explore Global Properties



Explore Global Properties



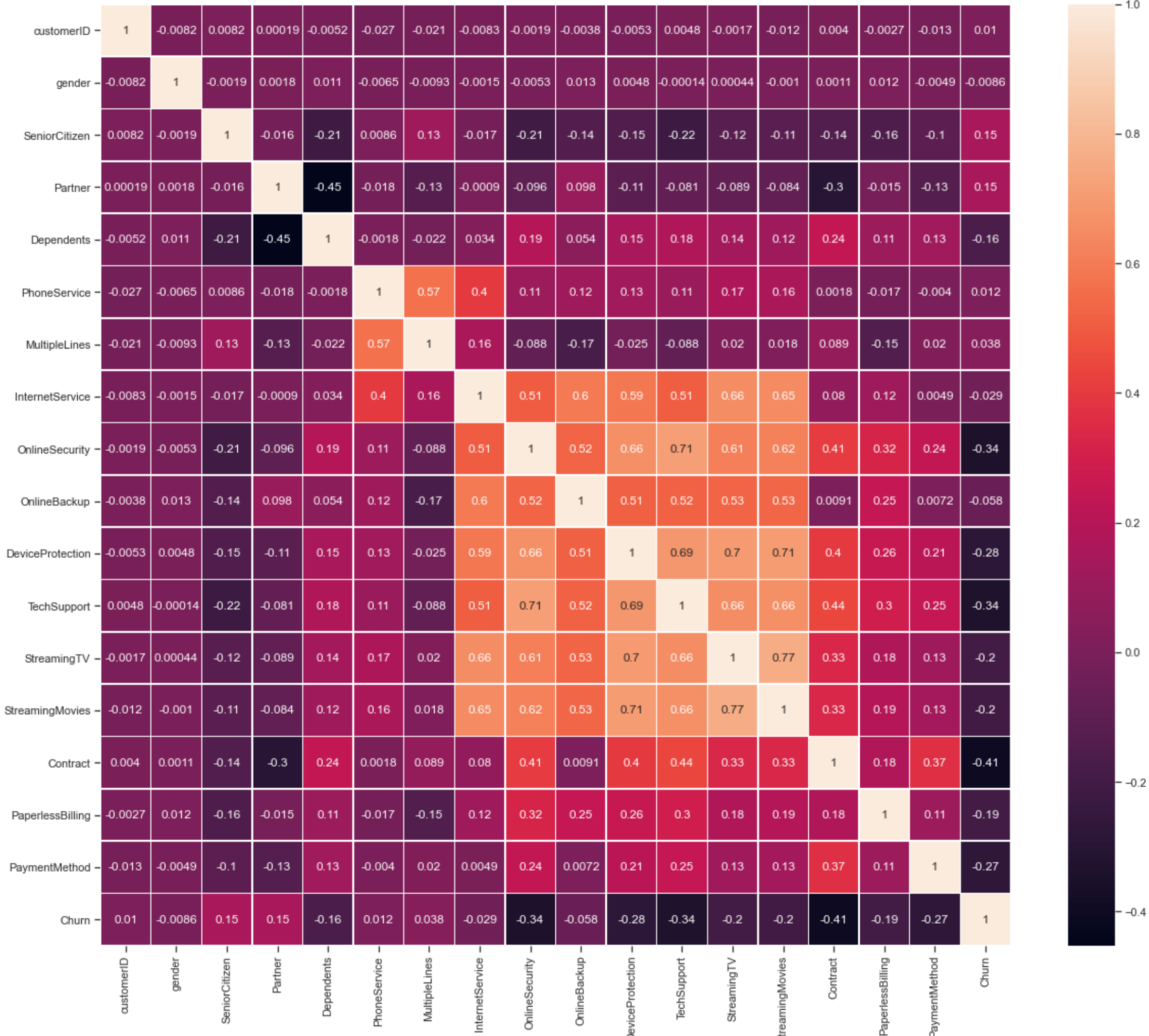
Explore Global Properties



```
In [19]: f, ax = plt.subplots(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, linewidths=.5, ax=ax)
```

```
In [20]: corr =df.drop(['tenure', 'MonthlyCharges'
                        , 'TotalCharges'] , 1).apply(lambda x : pd.factorize(x)[0]).corr(method='spearman'
                                                , min_periods=1)

f, ax = plt.subplots(figsize=(20, 17))
sns.heatmap(corr, annot=True, linewidths=.5, ax=ax)
```



Explore Global Properties

hypothesis testing

```
In [21]: def normality(x):  
    stat, p = shapiro(df[x])  
    print('Statistics=%.3f, p=%.3f' % (stat, p))  
    # interpret  
    alpha = 0.05  
    if p > alpha:  
        print('มีการแจกแจงแบบปกติ')  
    else:  
        print('ไม่มีการแจกแจงแบบปกติ')
```

```
In [22]: def twotest(x):  
    stat, p = mannwhitneyu(df[x][df['gender'] == 'Male'], df[x][df['gender'] == 'Female'])  
    print('Statistics=%.3f, p=%.3f' % (stat, p))  
    # interpret  
    alpha = 0.05  
  
    if p > alpha:  
        print('กลุ่มตัวอย่าง 2 กลุ่มที่มาจากประชากรไม่แตกต่างกัน')  
    else:  
        print('กลุ่มตัวอย่าง 2 กลุ่มที่มาจากประชากรแตกต่างกัน')
```

Explore Global Properties

normality test

```
In [29]: normality("tenure")
```

```
Statistics=0.904, p=0.000  
ไม่มีการแจกแจงแบบปกติ
```

```
In [24]: normality("TotalCharges")
```

```
Statistics=0.861, p=0.000  
ไม่มีการแจกแจงแบบปกติ
```

```
In [25]: normality("MonthlyCharges")
```

```
Statistics=0.921, p=0.000  
ไม่มีการแจกแจงแบบปกติ
```

Explore Global Properties

เปรียบเทียบความแตกต่าง 2 กลุ่ม

```
In [26]: twotest("tenure")
```

```
Statistics=6174973.000, p=0.385  
กลุ่มตัวอย่าง 2 กลุ่มที่มาจากประชากรไม่แตกต่างกัน
```

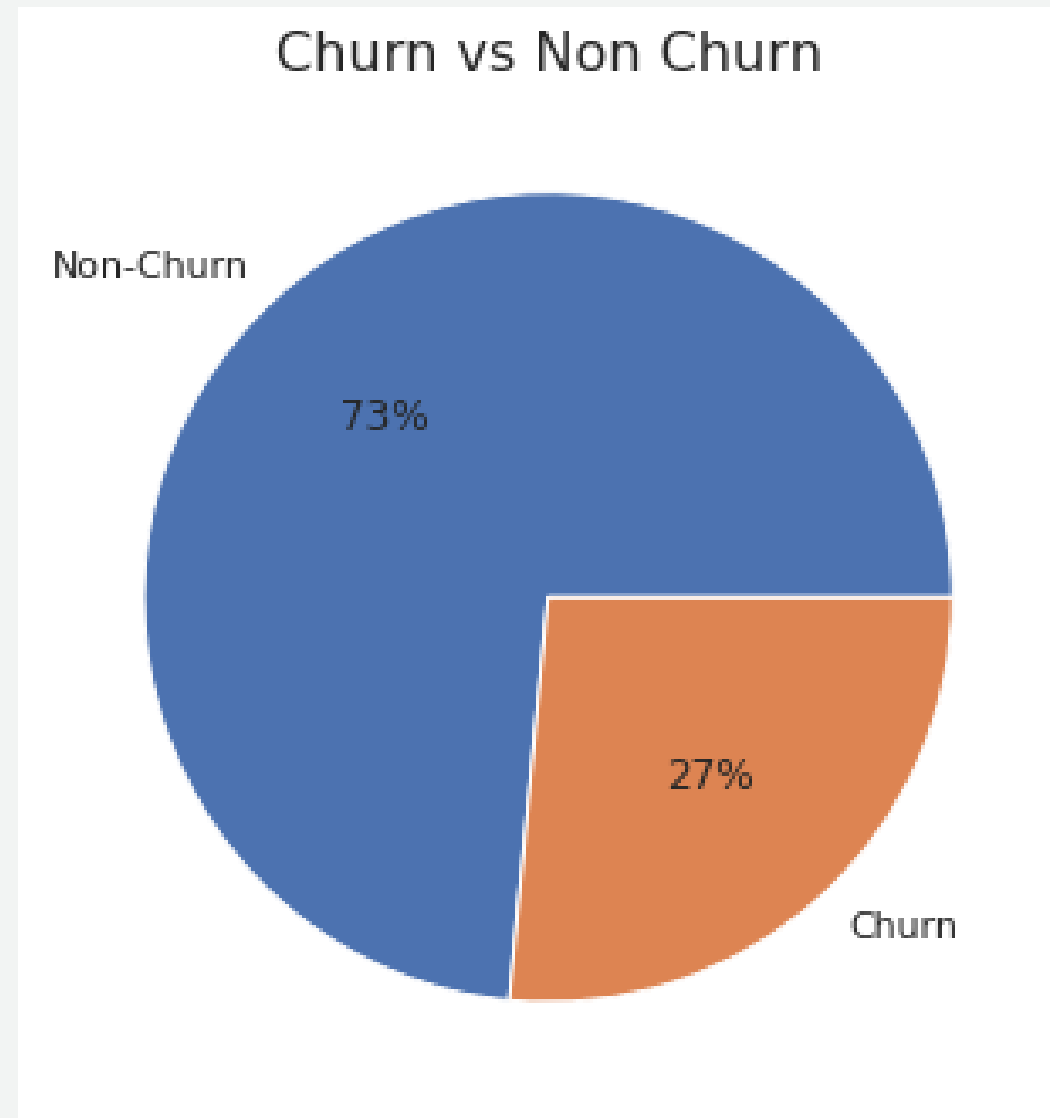
```
In [27]: twotest("TotalCharges")
```

```
Statistics=6192024.000, p=0.463  
กลุ่มตัวอย่าง 2 กลุ่มที่มาจากประชากรไม่แตกต่างกัน
```

```
In [28]: twotest("MonthlyCharges")
```

```
Statistics=6101575.500, p=0.125  
กลุ่มตัวอย่าง 2 กลุ่มที่มาจากประชากรไม่แตกต่างกัน
```

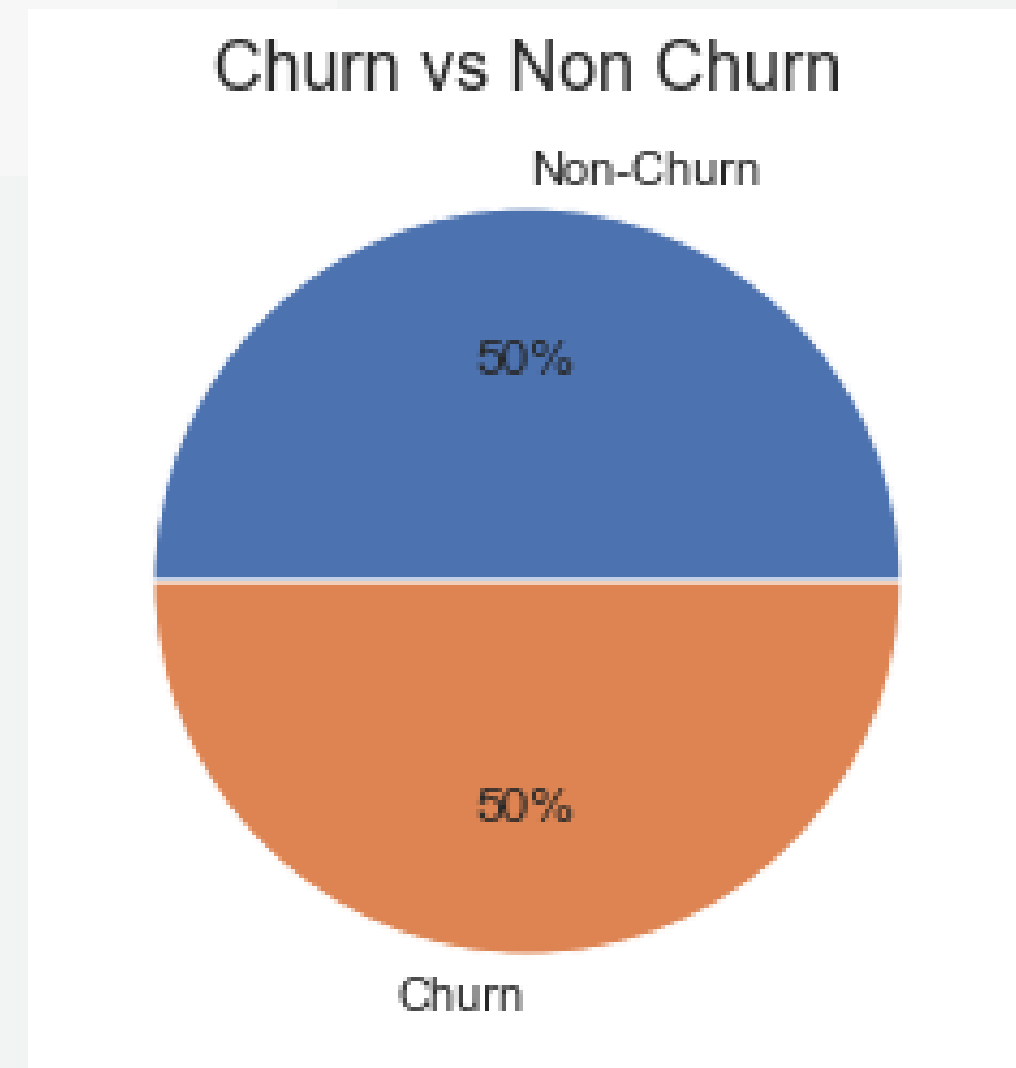
Modeling



Modeling

```
In [30]: df_Yes=df[df.Churn == 'Yes']  
df_No = df[df.Churn == 'No'].drop_duplicates().sample(n=1869, random_state=1)  
data_ML = pd.concat([df_No , df_Yes] , axis=0)
```

```
In [33]: churn_rate = data_ML.Churn.value_counts() / len(data_ML.Churn)  
labels = 'Non-Churn', 'Churn'  
  
fig, ax = plt.subplots()  
ax.pie(churn_rate, labels=labels, autopct='%f%%')  
ax.set_title('Churn vs Non Churn', fontsize=16)
```



Modeling

```
[34] exp_clf = setup(data_ML, target = 'Churn' , normalize = True , feature_selection = True
                    , remove_multicollinearity = True, multicollinearity_threshold = 0.75
                    , remove_outliers = True )
```

	Description	Value
0	session_id	5237
1	Target	Churn
2	Target Type	Binary
3	Label Encoded	No: 0, Yes: 1
4	Original Data	(3738, 19)
5	Missing Values	False
6	Numeric Features	3
7	Categorical Features	15
8	Ordinal Features	False
9	High Cardinality Features	False
10	High Cardinality Method	None
11	Transformed Train Set	(2485, 21)
12	Transformed Test Set	(1122, 21)

Modeling

▶ best_model = compare_models()



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ada	Ada Boost Classifier	0.7678	0.8431	0.8152	0.7641	0.7886	0.5317	0.5334	0.148
gbc	Gradient Boosting Classifier	0.7606	0.8385	0.8015	0.7614	0.7805	0.5175	0.5191	0.223
ridge	Ridge Classifier	0.7586	0.0000	0.8144	0.7523	0.7818	0.5125	0.5152	0.018
lda	Linear Discriminant Analysis	0.7586	0.8391	0.8144	0.7523	0.7818	0.5125	0.5152	0.025
lr	Logistic Regression	0.7582	0.8385	0.8000	0.7585	0.7783	0.5126	0.5142	0.332
catboost	CatBoost Classifier	0.7565	0.8284	0.7902	0.7615	0.7753	0.5099	0.5108	2.176
nb	Naive Bayes	0.7469	0.8251	0.8144	0.7375	0.7736	0.4882	0.4922	0.020
rf	Random Forest Classifier	0.7453	0.8105	0.7705	0.7561	0.7627	0.4878	0.4887	0.595
lightgbm	Light Gradient Boosting Machine	0.7384	0.8154	0.7811	0.7410	0.7603	0.4729	0.4740	0.083
xgboost	Extreme Gradient Boosting	0.7284	0.7959	0.7591	0.7375	0.7479	0.4536	0.4541	0.668
svm	SVM - Linear Kernel	0.7272	0.0000	0.7318	0.7686	0.7295	0.4538	0.4770	0.028
knn	K Neighbors Classifier	0.7244	0.7776	0.7932	0.7190	0.7537	0.4426	0.4461	0.125
et	Extra Trees Classifier	0.7107	0.7837	0.7402	0.7236	0.7312	0.4180	0.4189	0.571
dt	Decision Tree Classifier	0.6765	0.6731	0.6894	0.6986	0.6934	0.3509	0.3515	0.024
qda	Quadratic Discriminant Analysis	0.5303	0.5198	0.6886	0.4577	0.5404	0.0464	0.0807	0.022

Modeling

lr = create_model('lr')



	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7470	0.8423	0.7652	0.7594	0.7623	0.4919	0.4919
1	0.7671	0.8582	0.8409	0.7500	0.7929	0.5288	0.5333
2	0.7510	0.8311	0.7955	0.7500	0.7721	0.4982	0.4993
3	0.7590	0.8364	0.8182	0.7500	0.7826	0.5135	0.5159
4	0.7590	0.8469	0.8106	0.7535	0.7810	0.5140	0.5157
5	0.7258	0.7941	0.7424	0.7424	0.7424	0.4493	0.4493
6	0.7944	0.8465	0.8333	0.7914	0.8118	0.5855	0.5864
7	0.8145	0.8872	0.8485	0.8116	0.8296	0.6263	0.6271
8	0.7419	0.8251	0.7424	0.7656	0.7538	0.4828	0.4830
9	0.7218	0.8172	0.8030	0.7114	0.7544	0.4362	0.4405
Mean	0.7582	0.8385	0.8000	0.7585	0.7783	0.5126	0.5142
SD	0.0272	0.0236	0.0367	0.0258	0.0260	0.0547	0.0545

Modeling

▶ tuned_lr = tune_model(lr)



	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7590	0.8417	0.7500	0.7857	0.7674	0.5177	0.5183
1	0.7952	0.8578	0.8258	0.7956	0.8104	0.5879	0.5883
2	0.7550	0.8308	0.7727	0.7669	0.7698	0.5080	0.5080
3	0.7671	0.8367	0.7955	0.7721	0.7836	0.5315	0.5318
4	0.7631	0.8467	0.7879	0.7704	0.7790	0.5237	0.5238
5	0.7298	0.7942	0.7197	0.7600	0.7393	0.4594	0.4601
6	0.7903	0.8466	0.8030	0.8030	0.8030	0.5789	0.5789
7	0.8185	0.8869	0.8333	0.8271	0.8302	0.6354	0.6354
8	0.7379	0.8249	0.7197	0.7724	0.7451	0.4761	0.4773
9	0.7258	0.8171	0.7652	0.7319	0.7481	0.4476	0.4481
Mean	0.7642	0.8383	0.7773	0.7785	0.7776	0.5266	0.5270
SD	0.0284	0.0235	0.0376	0.0247	0.0284	0.0566	0.0564

Modeling

▶ tuned_lr = tune_model(lr)



	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7590	0.8417	0.7500	0.7857	0.7674	0.5177	0.5183
1	0.7952	0.8578	0.8258	0.7956	0.8104	0.5879	0.5883
2	0.7550	0.8308	0.7727	0.7669	0.7698	0.5080	0.5080
3	0.7671	0.8367	0.7955	0.7721	0.7836	0.5315	0.5318
4	0.7631	0.8467	0.7879	0.7704	0.7790	0.5237	0.5238
5	0.7298	0.7942	0.7197	0.7600	0.7393	0.4594	0.4601
6	0.7903	0.8466	0.8030	0.8030	0.8030	0.5789	0.5789
7	0.8185	0.8869	0.8333	0.8271	0.8302	0.6354	0.6354
8	0.7379	0.8249	0.7197	0.7724	0.7451	0.4761	0.4773
9	0.7258	0.8171	0.7652	0.7319	0.7481	0.4476	0.4481
Mean	0.7642	0.8383	0.7773	0.7785	0.7776	0.5266	0.5270
SD	0.0284	0.0235	0.0376	0.0247	0.0284	0.0566	0.0564

Modeling



```
tuned_lr
```

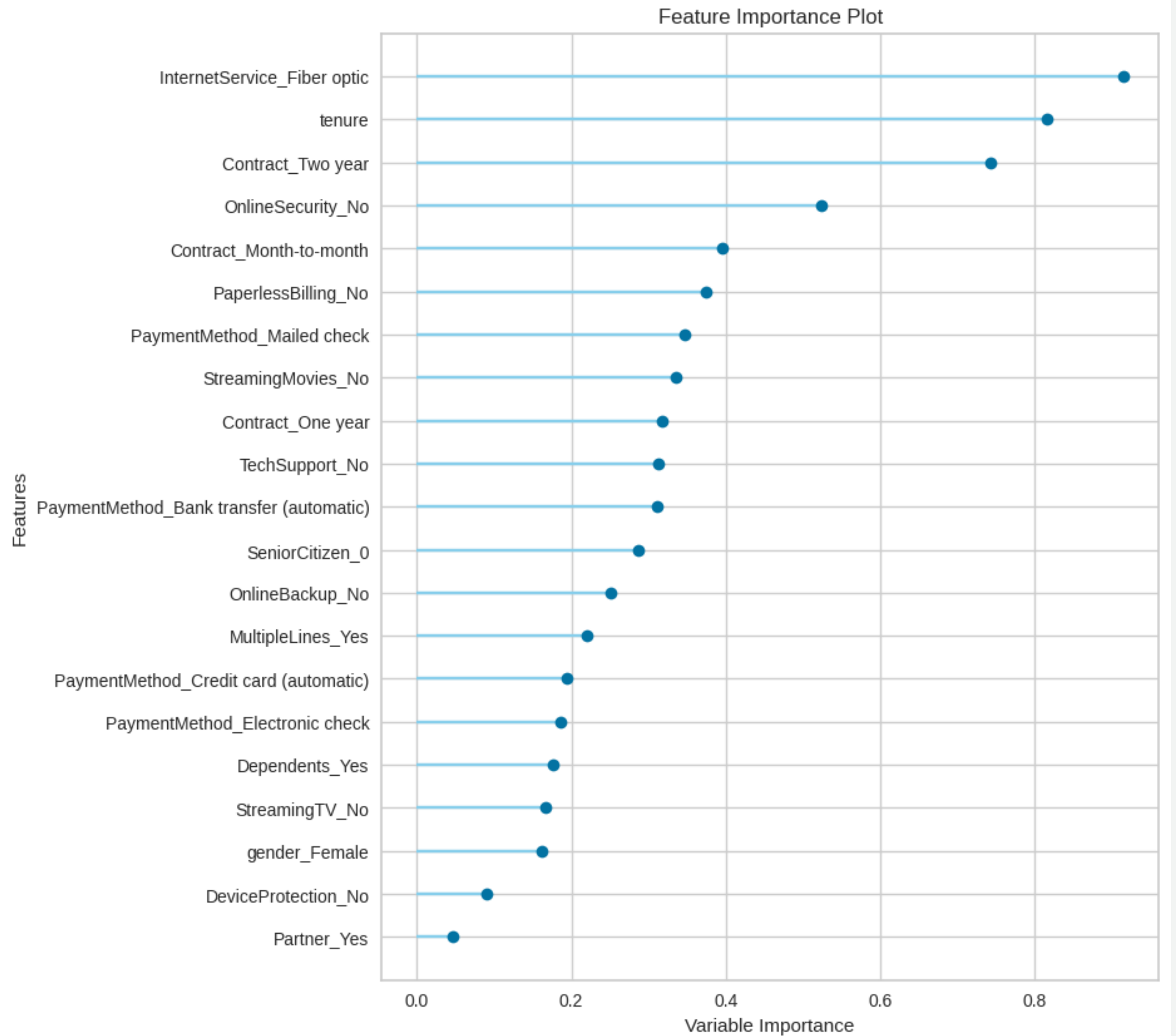
```
LogisticRegression(C=3.006, class_weight='balanced', dual=False,  
                    fit_intercept=True, intercept_scaling=1, l1_ratio=None,  
                    max_iter=1000, multi_class='auto', n_jobs=None,  
                    penalty='none', random_state=5237, solver='lbfgs',  
                    tol=0.0001, verbose=0, warm_start=False)
```



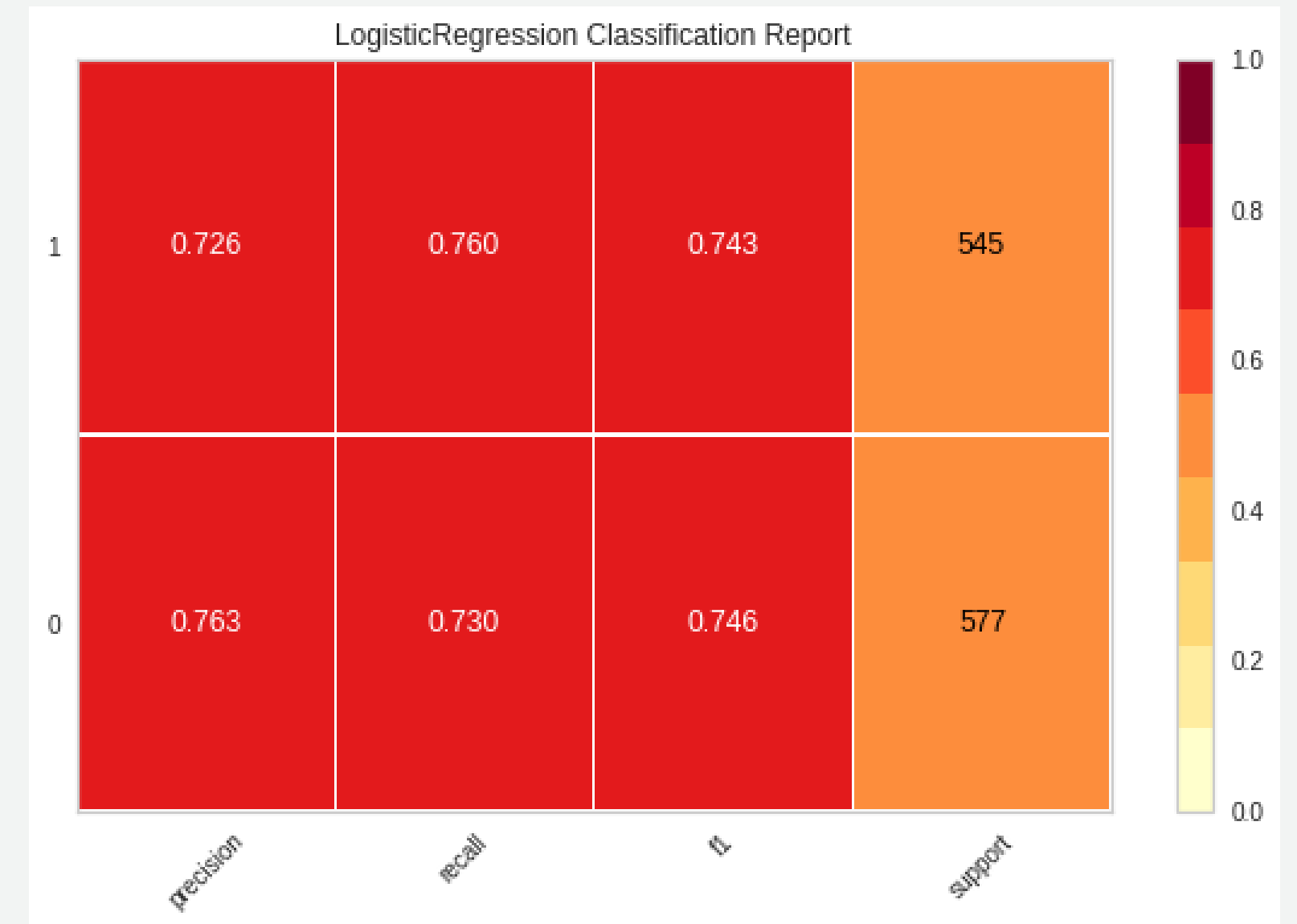
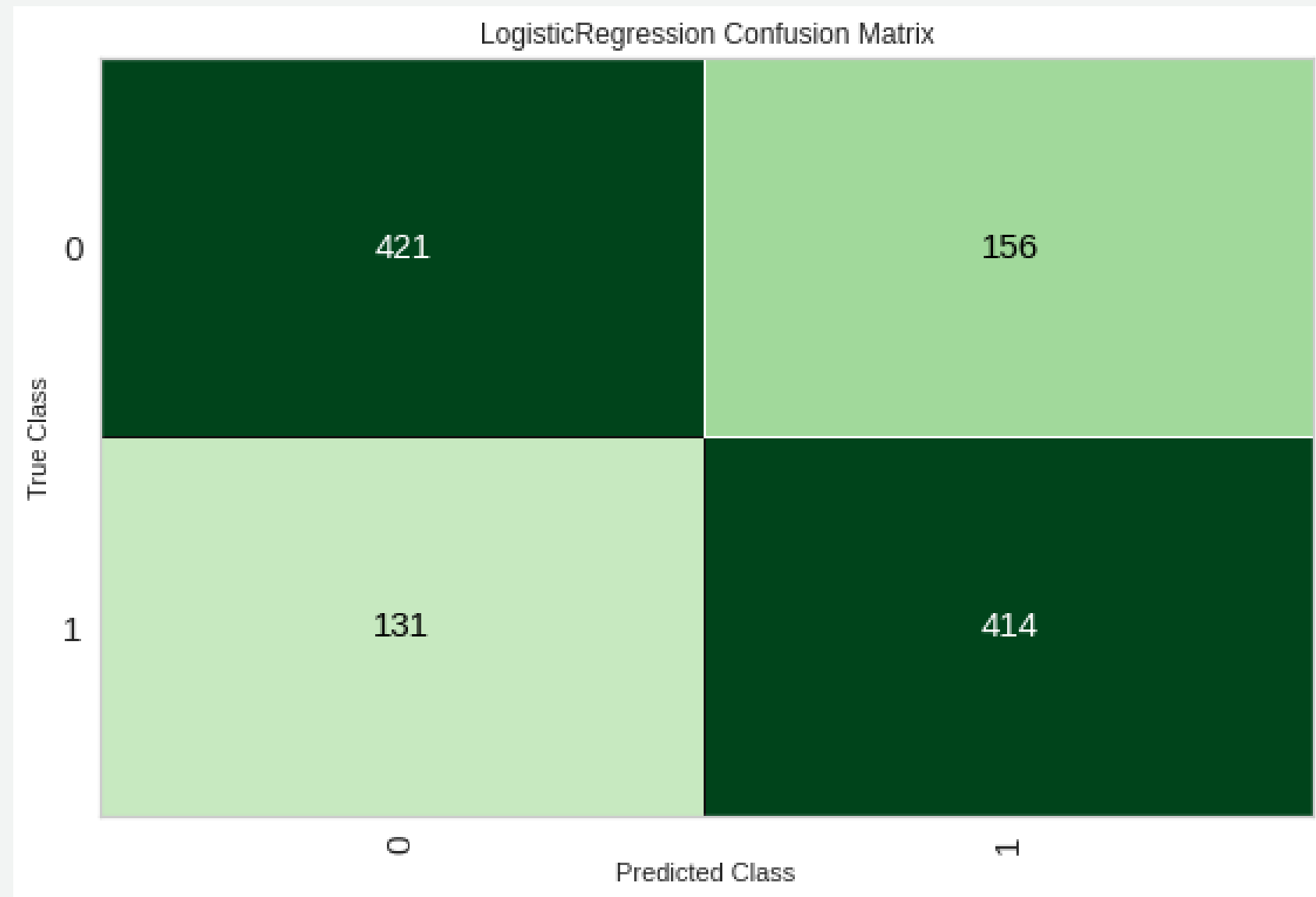
```
pred_holdout = predict_model(tuned_lr)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Logistic Regression	0.7442	0.8258	0.7596	0.7263	0.7426	0.4886	0.4891

Modeling



Modeling



Story Telling

EDA

1. ลูกค้าที่อยู่กับเราน้อย(เดือน)มีแนวโน้มที่จะเลิกใช้บริการ
2. ลูกค้าที่มีค่าบริการรายเดือนต่ำมีโอกาสที่จะใช้บริการ มากกว่าลูกค้าที่มีค่าบริการรายเดือนสูง
3. อัตราการเลิกใช้บริการ ระหว่างเพศชายและเพศหญิงมีความใกล้เคียงกัน
4. ลูกค้าที่เป็นผู้สูงอายุมีแนวโน้มที่จะเลิกใช้บริการมากกว่าลูกค้าที่ไม่เป็นผู้สูงอายุ
5. ลูกค้าที่ไม่เป็น **Partner** มีแนวโน้มที่จะเลิกใช้บริการ
6. ลูกค้าที่ไม่มีผู้อยู่ในอุปการะมีแนวโน้มที่จะซื้อสัตย์
7. ลูกค้าที่มี **Multiple Lines** และไม่มี **Multiple Lines** มีแนวโน้มที่จะเลิกใช้บริการ เท่าๆกันซึ่งมากกว่า **No phone service**
8. ลูกค้าที่ใช้บริการอินเทอร์เน็ตเป็น **Fiber optic** มีแนวโน้มที่จะเลิกใช้บริการมากที่สุด
9. ลูกค้าที่มีการเรียกเก็บเงินแบบใช้กระดาษมีแนวโน้มที่จะเลิกใช้บริการ
10. ลูกค้าที่ใช้เช็คอิเล็กทรอนิกส์ในการชำระค่าใช้จ่ามีแนวโน้มที่จะเลิกใช้บริการ มากกว่าผู้ที่ใช้วิธีการชำระเงินแบบอื่น
- 11 . อัตราการเลิกใช้บริการจะลดลงเมื่อระยะเวลาของสัญญาเพิ่มขึ้น

Story Telling

hypothesis testing

ระหว่างเพศชายและเพศหญิง พบว่า จำนวนเดือนที่ลูกค้าใช้ จำนวนเงินทั้งหมดที่เรียกเก็บจากลูกค้า และจำนวนเงินที่เรียกเก็บจากลูกค้ารายเดือนไม่แตกต่างกันที่ความเชื่อมั่น 95%

model

- internertService_Fiber optic , tenure , Contract_Two year , OnlineSecurity_No เป็นปัจจัยที่มีความสำคัญต่อการเลิกใช้บริการ (Customer Churn)
- model จะมีค่า Accuracy ที่สูงถึง 74.42 % ส่วน Confusion Matrix จะเห็นได้ว่าการทำนายมีความแม่นยำระดับปานกลางไปถึงดี ซึ่งยังสามารถดูได้จาก Class_Report ใน class 0 ,1 ของ column ของ f1-score จะได้ 74.3%และ 74.6% ตามลำดับ จะทำให้ model มีประสิทธิภาพปานกลางจนถึงดีในการทำนาย Churn