

HIỂU DỮ LIỆU

Mô tả tập dữ liệu sử dụng: Tập dữ liệu chứa thông tin về tất cả 7.043 khách hàng từ một công ty Viễn thông ở California trong quý 2 năm 2022. Mỗi bản ghi đại diện cho một khách hàng và chứa thông tin chi tiết về nhân khẩu học, vị trí, thời hạn sử dụng, dịch vụ đăng ký, trạng thái của quý (đã tham gia, ở lại hoặc rời dịch vụ).

Các trường thông tin bao gồm:

- Customer ID: Một ID duy nhất xác định từng khách hàng
- Gender: Giới tính của khách hàng
- Age: Tuổi
- Married: Tình trạng hôn nhân
- Number of Dependents: Số lượng phụ thuộc
- City: Thành phố nơi sống
- Zip Code: Mã bưu điện
- Latitude: Vĩ độ của nơi cư trú chính của khách hàng
- Longitude: Kinh độ của nơi cư trú chính của khách hàng
- Number of Referrals: SSố lần khách hàng đã giới thiệu bạn bè hoặc thành viên gia đình đến công ty này cho đến nay
- Tenure in Months: Cho biết tổng số tháng mà khách hàng đã làm việc với công ty
- Offer: Xác định ưu đãi tiếp thị cuối cùng mà khách hàng chấp nhận
- Phone Service: Cho biết khách hàng có đăng ký dịch vụ điện thoại tại nhà với công ty hay không: Có, Không.
- Avg Monthly Long Distance Charges: Cho biết phí đường dài trung bình của khách hàng, được tính đến cuối quý.
- Multiple Lines: Cho biết nếu khách hàng đăng ký nhiều đường dây điện thoại với công ty: Có, Không.
- Internet Service: Indicates if the customer subscribes to Internet service with the company: Yes, No
- Internet Type: Cho biết loại kết nối internet của khách hàng: DSL, Cáp quang, Cáp.
- Avg Monthly GB Download: Cho biết khối lượng tải xuống trung bình của khách hàng tính bằng gigabyte, được tính đến cuối quý
- Online Security: Indicates if the customer subscribes to an additional online security service provided by the company: Yes, No.
- Online Backup: Cho biết liệu khách hàng có đăng ký dịch vụ sao lưu trực tuyến bổ sung do công ty cung cấp hay không: Có, Không
- Device Protection Plan: Cho biết liệu khách hàng có đăng ký gói bảo vệ thiết bị bổ sung cho thiết bị Internet của họ do công ty cung cấp hay không.
- Premium Tech Support: Cho biết nếu khách hàng đăng ký gói hỗ trợ kỹ thuật bổ sung từ công ty.
- Streaming TV: Cho biết liệu khách hàng có sử dụng dịch vụ Internet của họ để truyền phát chương trình truyền hình từ nhà cung cấp bên thứ ba hay không.
- Streaming Movies: Cho biết liệu khách hàng có sử dụng dịch vụ Internet của họ để phát trực tuyến phim từ nhà cung cấp bên thứ ba mà không phải trả thêm phí hay không: Có, Không.
- Streaming Music: Cho biết liệu khách hàng có sử dụng dịch vụ Internet của họ để truyền phát nhạc từ nhà cung cấp bên thứ ba mà không mất thêm phí hay không: Có, Không.
- Unlimited Data: Cho biết liệu khách hàng có trả thêm phí hàng tháng để tải xuống / tải lên dữ liệu không giới hạn hay không
- Contract: Cho biết loại hợp đồng hiện tại của khách hàng: Hàng tháng, Một năm, Hai năm.
- Paperless Billing: Cho biết nếu khách hàng đã chọn thanh toán không cần hóa đơn: Có, Không

- Payment Method: Cho biết cách khách hàng thanh toán hóa đơn của họ.
- Monthly Charge: Cho biết tổng phí hàng tháng hiện tại của khách hàng cho tất cả các dịch vụ của họ từ công ty.
- Total Charges: Cho biết tổng các khoản phí của khách hàng, được tính đến cuối quý đã nêu ở trên.
- Total Refunds: Cho biết tổng số tiền hoàn lại của khách hàng, được tính đến cuối quý được chỉ định ở trên
- Total Extra Data CCharges Cho biết tổng số phí của khách hàng cho các lần tải xuống dữ liệu bổ sung cao hơn những khoản được chỉ định trong gói của họ.
- Total Long Distance Charges: Cho biết tổng các khoản phí của khách hàng cho quãng đường dài cao hơn những khoản phí được chỉ định trong gói của họ
- Total Revenue: Cho biết tổng doanh thu của công ty từ khách hàng này, được tính đến cuối quý đã nêu ở trên
- Customer Status: Cho biết trạng thái của khách hàng vào cuối quý (Churned, Stayed, or Joined)
- Churn Category: Một danh mục cho trước lý do khách hàng rời bỏ dịch vụ
- Churn Reason: Lý do cụ thể của khách hàng để rời khỏi công ty, được hỏi khi họ rời công ty.

1. Phát biểu và hiểu bài toán

Bài toán dự đoán khách hàng rời dịch vụ. Đầu vào đầu ra của mô hình:

- **Input:** các thuộc tính nhà mạng đã cung cấp
- **Output:** khả năng rời dịch vụ của khách hàng

2. Phân tích và trực quan hóa dữ liệu

2.1. Thư viện và nạp dữ liệu

```
In [1]: import numpy as np
import pandas as pd
import pickle

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import OrdinalEncoder, OneHotEncoder, label_binarize
from sklearn.preprocessing import MinMaxScaler, StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
```

```
In [2]: filename = 'telecom_customer_churn.csv'
print(f'Load data from {filename}.')
df = pd.read_csv(filename)
```

Load data from telecom_customer_churn.csv.

```
In [3]: df.columns
```

```
Out[3]: Index(['Customer ID', 'Gender', 'Age', 'Married', 'Number of Dependents',
              'City', 'Zip Code', 'Latitude', 'Longitude', 'Number of Referrals',
              'Tenure in Months', 'Offer', 'Phone Service',
              'Avg Monthly Long Distance Charges', 'Multiple Lines',
              'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
              'Online Security', 'Online Backup', 'Device Protection Plan',
              'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
              'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless Billing',
              'Payment Method', 'Monthly Charge', 'Total Charges', 'Total Refunds',
              'Total Extra Data Charges', 'Total Long Distance Charges',
              'Total Revenue', 'Customer Status', 'Churn Category', 'Churn Reason'],
              dtype='object')
```

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

Out[4]:

	Customer ID	Gender	Age	Married	Number of Dependents	City	Zip Code	Latitude	Longitude	Number of Referrals
0	0002-ORFBO	Female	37	Yes	0	Frazier Park	93225	34.827662	-118.999073	0
1	0003-MKNFE	Male	46	No	0	Glendale	91206	34.162515	-118.203869	0
2	0004-TLHLJ	Male	50	No	0	Costa Mesa	92627	33.645672	-117.922613	0
3	0011-IGKFF	Male	78	Yes	0	Martinez	94553	38.014457	-122.115432	0
4	0013-EXCHZ	Female	75	Yes	0	Camarillo	93010	34.227846	-119.079903	0

5 rows × 38 columns

2.2. Phân tích và xử lý missing

tương quan biến, thống kê đơn biến, trực quan hóa

In [5]:


df.describe()

Out[5]:

	Age	Number of Dependents	Zip Code	Latitude	Longitude	Number of Referrals	Tenure in Months
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000
mean	46.509726	0.468692	93486.070567	36.197455	-119.756684	1.951867	32.386767
std	16.750352	0.962802	1856.767505	2.468929	2.154425	3.001199	24.542061
min	19.000000	0.000000	90001.000000	32.555828	-124.301372	0.000000	1.000000
25%	32.000000	0.000000	92101.000000	33.990646	-121.788090	0.000000	9.000000
50%	46.000000	0.000000	93518.000000	36.205465	-119.595293	0.000000	29.000000
75%	60.000000	0.000000	95329.000000	38.161321	-117.969795	3.000000	55.000000
max	80.000000	9.000000	96150.000000	41.962127	-114.192901	11.000000	72.000000

```
In [6]: df.isna().sum()
```

```
Out[6]: Customer ID          0
Gender          0
Age             0
Married         0
Number of Dependents  0
City            0
Zip Code        0
Latitude        0
Longitude       0
Number of Referrals  0
Tenure in Months  0
Offer           0
Phone Service    0
Avg Monthly Long Distance Charges  682
Multiple Lines   682
Internet Service  0
Internet Type    1526
Avg Monthly GB Download  1526
Online Security  1526
Online Backup    1526
Device Protection Plan  1526
Premium Tech Support  1526
Streaming TV     1526
Streaming Movies  1526
Streaming Music  1526
Unlimited Data    1526
Contract         0
Paperless Billing  0
Payment Method   0
Monthly Charge   0
Total Charges    0
Total Refunds    0
Total Extra Data Charges  0
Total Long Distance Charges  0
Total Revenue    0
Customer Status  0
Churn Category    5174
Churn Reason      5174
dtype: int64
```

```
In [7]:  # fill null
#Avg Monthly Long Distance Charges = mean
df['Avg Monthly Long Distance Charges'] = df['Avg Monthly Long Distance Charges']

# Multiple Lines = 'No'
df['Multiple Lines'] = df['Multiple Lines'].fillna('No')

#Internet Type = 'None'
df['Internet Type'] = df['Internet Type'].fillna('None')

#Avg Monthly GB Download = mean
df['Avg Monthly GB Download'] = df['Avg Monthly GB Download'].fillna(df['Avg Monthly GB Download'].mean())

#Online Security = 'No'
df['Online Security'] = df['Online Security'].fillna('No')

#Online Backup = 'No'
df['Online Backup'] = df['Online Backup'].fillna('No')

#Device Protection Plan = 'No'
df['Device Protection Plan'] = df['Device Protection Plan'].fillna('No')

#Premium Tech Support = 'No'
df['Premium Tech Support'] = df['Premium Tech Support'].fillna('No')

#Streaming TV = 'No'
df['Streaming TV'] = df['Streaming TV'].fillna('No')

#Streaming Movies = 'No'
df['Streaming Movies'] = df['Streaming Movies'].fillna('No')

#Streaming Music = 'No'
df['Streaming Music'] = df['Streaming Music'].fillna('No')

#Unlimited Data = 'No'
df['Unlimited Data'] = df['Unlimited Data'].fillna('No')

#Churn Category = 'Other'
df['Churn Category'] = df['Churn Category'].fillna('Other')

#Churn Reason = 'Other'
df['Churn Reason'] = df['Churn Reason'].fillna('Other')
```

```
In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 38 columns):
#   Column                                          Non-Null Count  Dtype
---  -
0   Customer ID                                   7043 non-null   object
1   Gender                                         7043 non-null   object
2   Age                                             7043 non-null   int64
3   Married                                        7043 non-null   object
4   Number of Dependents                         7043 non-null   int64
5   City                                           7043 non-null   object
6   Zip Code                                       7043 non-null   int64
7   Latitude                                       7043 non-null   float64
8   Longitude                                      7043 non-null   float64
9   Number of Referrals                          7043 non-null   int64
10  Tenure in Months                             7043 non-null   int64
11  Offer                                           7043 non-null   object
12  Phone Service                                  7043 non-null   object
13  Avg Monthly Long Distance Charges            7043 non-null   float64
14  Multiple Lines                                7043 non-null   object
15  Internet Service                             7043 non-null   object
16  Internet Type                                 7043 non-null   object
17  Avg Monthly GB Download                     7043 non-null   float64
18  Online Security                             7043 non-null   object
19  Online Backup                                7043 non-null   object
20  Device Protection Plan                      7043 non-null   object
21  Premium Tech Support                        7043 non-null   object
22  Streaming TV                                 7043 non-null   object
23  Streaming Movies                            7043 non-null   object
24  Streaming Music                             7043 non-null   object
25  Unlimited Data                              7043 non-null   object
26  Contract                                      7043 non-null   object
27  Paperless Billing                           7043 non-null   object
28  Payment Method                              7043 non-null   object
29  Monthly Charge                              7043 non-null   float64
30  Total Charges                               7043 non-null   float64
31  Total Refunds                               7043 non-null   float64
32  Total Extra Data Charges                    7043 non-null   int64
33  Total Long Distance Charges                 7043 non-null   float64
34  Total Revenue                              7043 non-null   float64
35  Customer Status                             7043 non-null   object
36  Churn Category                             7043 non-null   object
37  Churn Reason                               7043 non-null   object
dtypes: float64(9), int64(6), object(23)
memory usage: 2.0+ MB
```

```
In [9]: df['Zip Code']=df['Zip Code'].astype('object')
```

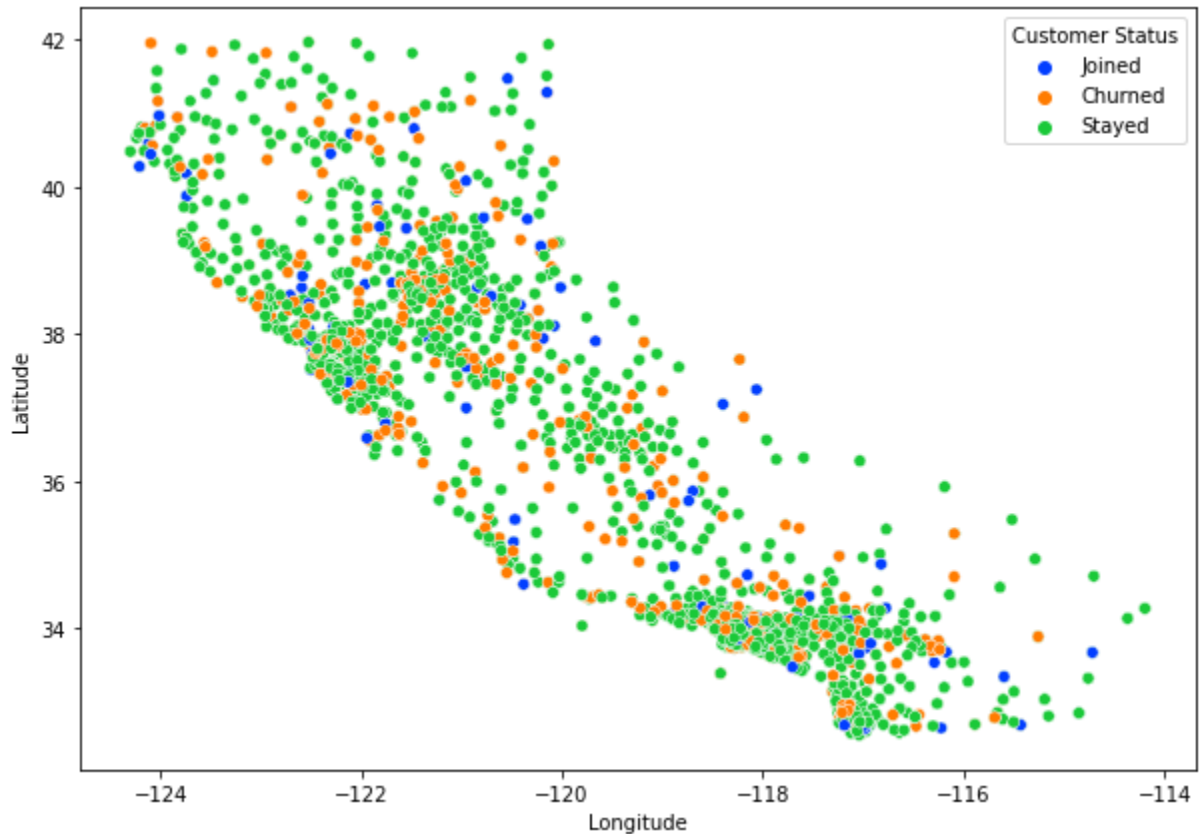
```
In [10]: df_train, df_test = train_test_split(df, test_size=0.2, random_state=42)
```

2.3. Trực quan hóa dữ liệu

Lon Lat

```
In [11]: fig, ax = plt.subplots(figsize=(10,7))
sns.scatterplot(data=df_train, x='Longitude', y='Latitude', hue='Customer Status')
```

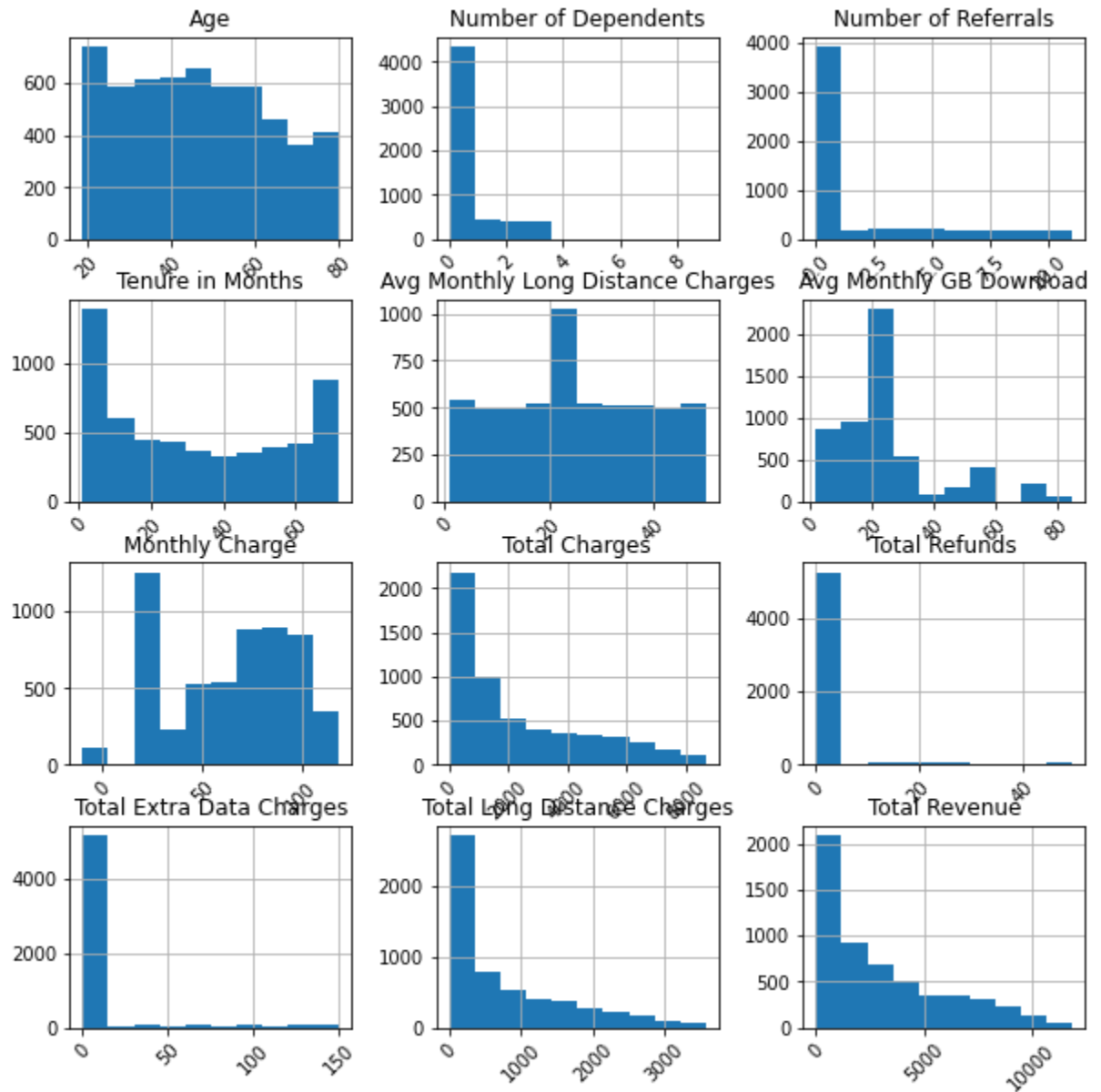
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f79226cd790>



```
In [12]: df_num = df_train.select_dtypes('number')
df_num = df_num.drop(['Longitude', 'Latitude'], axis=1)
df_object = df_train.select_dtypes('object')
df_object = df_object.drop(['City', 'Zip Code', 'Customer ID', 'Churn Category', ''])
```

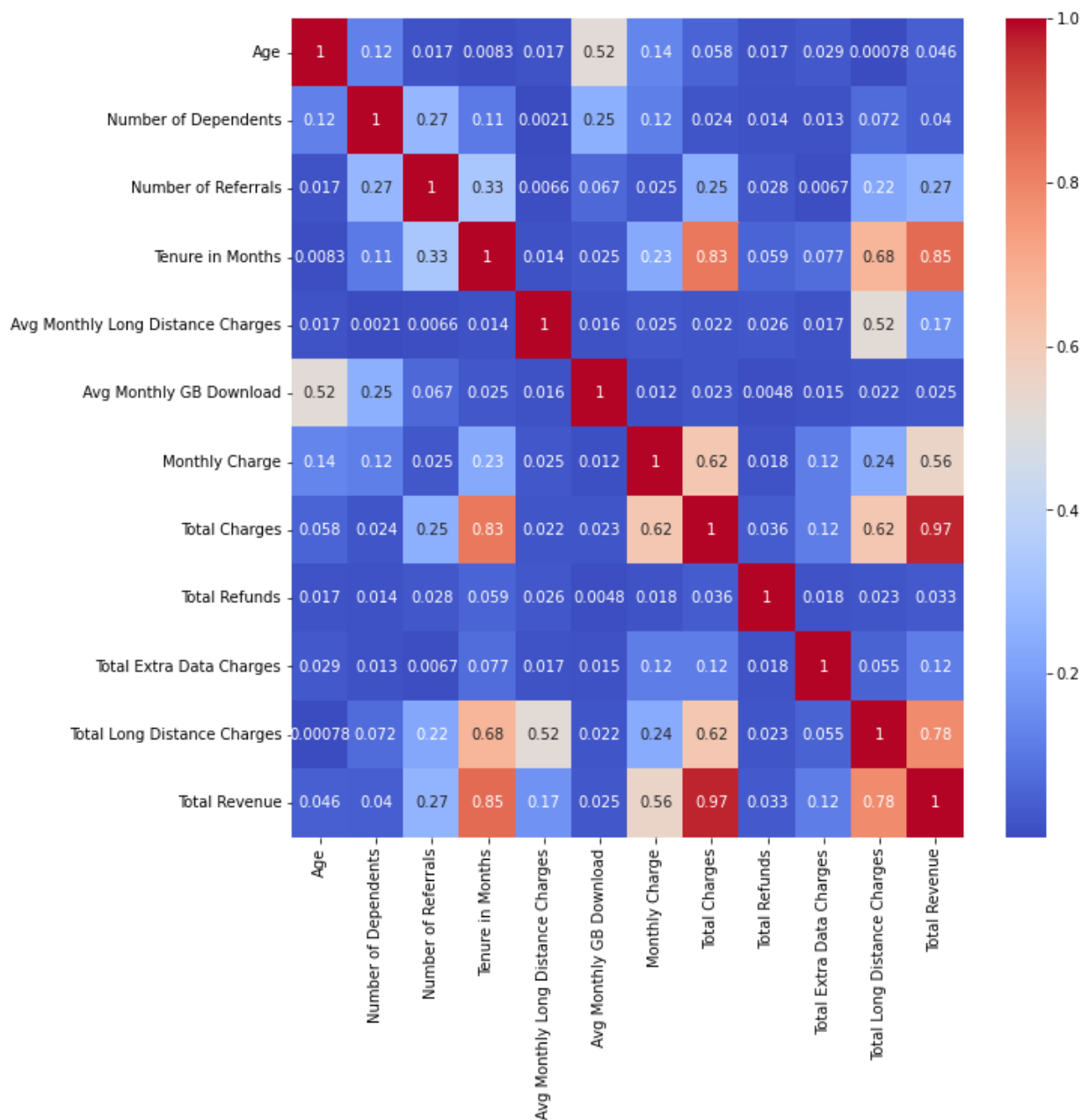
Biểu đồ Histogram của dữ liệu kiểu số:


```
In [13]: df_num.hist(figsize=(10,10), xrot=45);
```



Tương quan biến

```
In [14]: cor = df_num.corr().abs()
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(cor, annot=True, ax = ax, cmap='coolwarm');
```

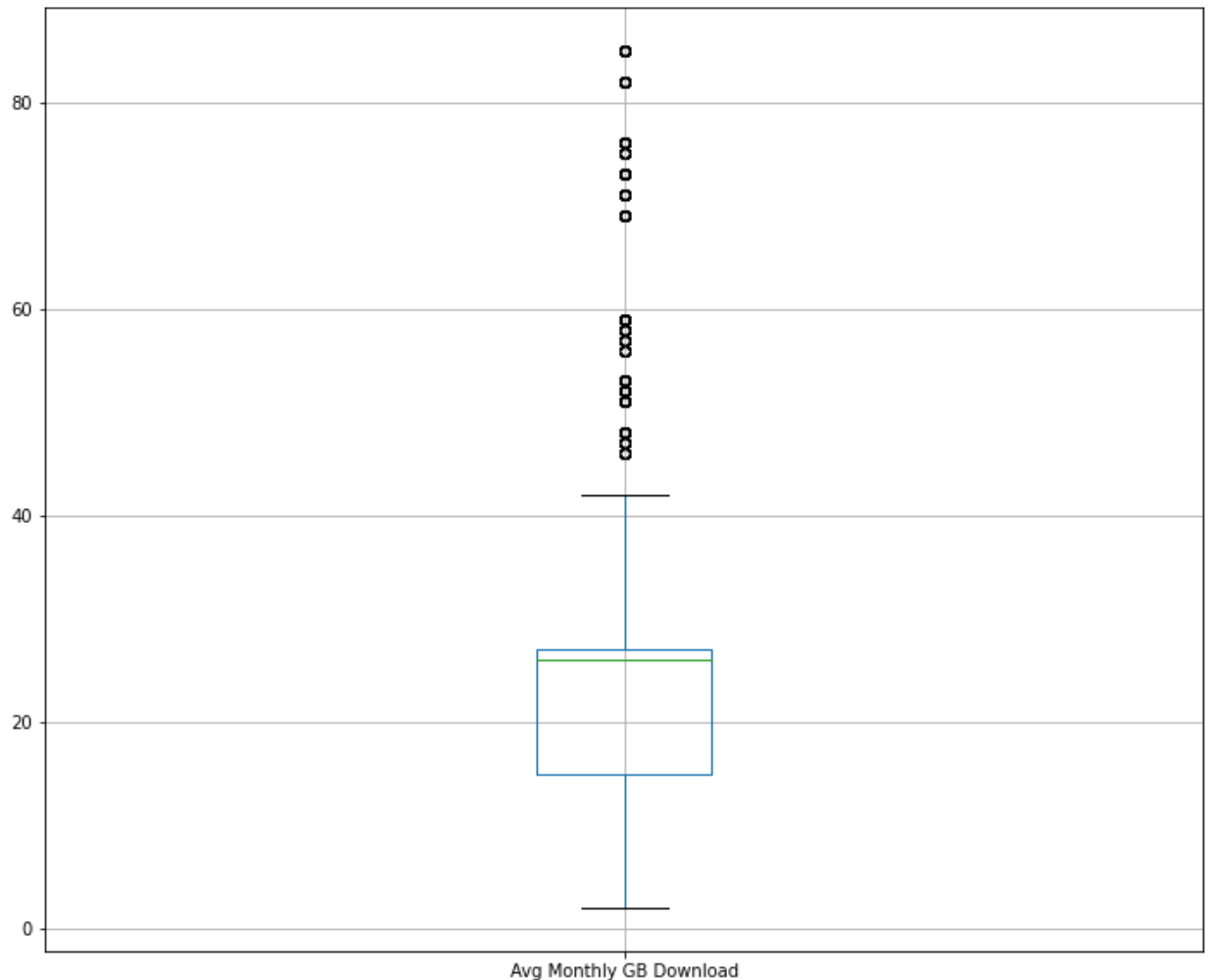


Dựa vào biểu đồ ta thấy, các trường dữ liệu thể hiện các loại phí có tương quan cao với nhau. Đặc biệt Total Charges và Total Long Distance Charges có tương quan cao với Total Revenue, v.v.

```
In [15]: > yn_feature = ['Married', 'Phone Service', 'Multiple Lines', 'Internet Service',  
                        'Online Security', 'Online Backup', 'Device Protection Plan',  
                        'Premium Tech Support', 'Streaming TV', 'Streaming Movies',  
                        'Streaming Music', 'Unlimited Data', 'Paperless Billing',]
```

```
In [16]: > df_num[['Avg Monthly GB Download']].boxplot(figsize=(12,10))
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7920348410>
```

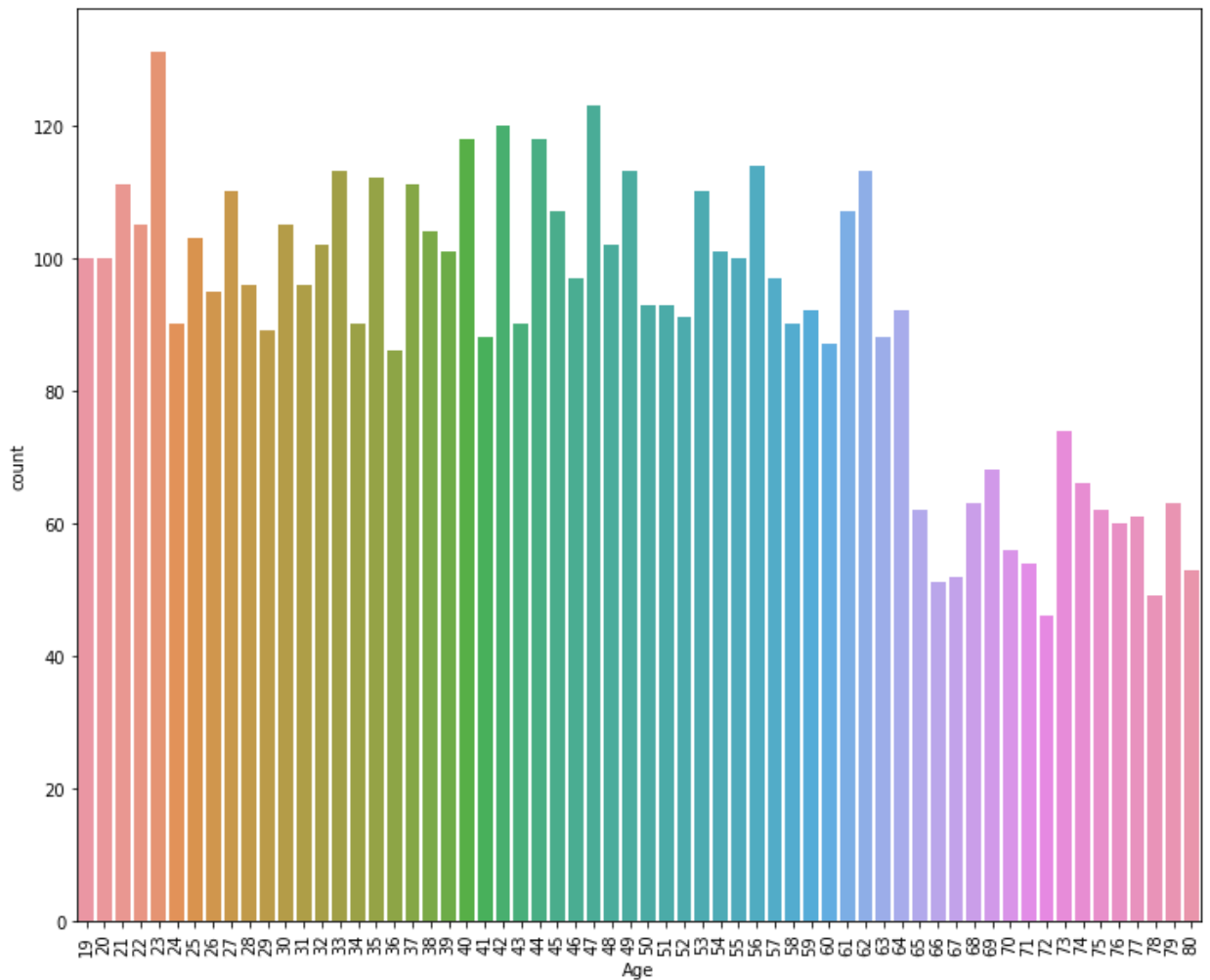


Cần phải phân tích xem các ngoại lệ mà chúng ta thấy trong biểu đồ trên có ý nghĩa hay không. Chúng ta cần phải tìm lý do tải xuống cao như vậy để đưa ra phương án xử lý ngoại lệ.

```
In [17]: ▶ plt.figure(figsize=(12,10))
sns.countplot(df_num['Age'])
plt.xticks(rotation=90);
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

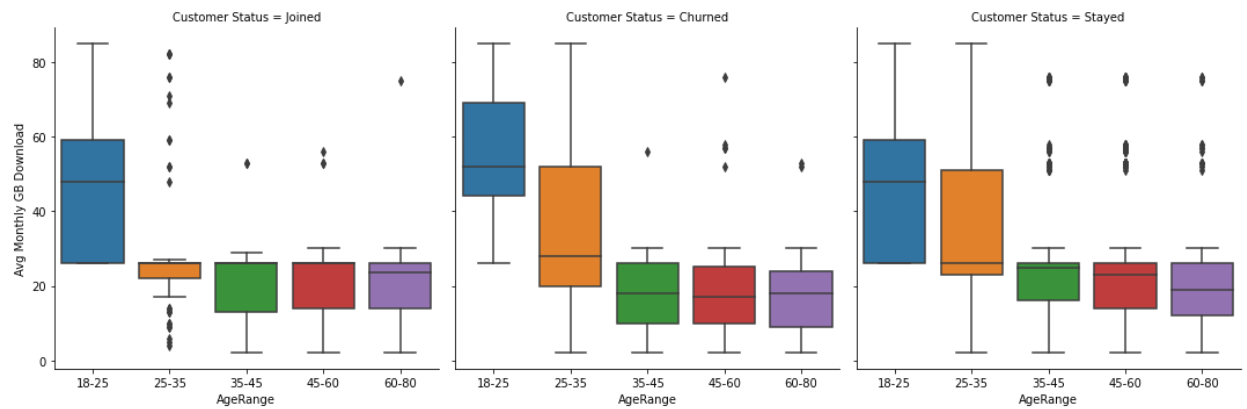


```
In [18]: ▶ bins = [18,25,35,45,60,80]
names = ['18-25','25-35','35-45','45-60','60-80']
df_object['AgeRange'] = pd.cut(df_num['Age'],bins, labels=names,include_lowest=True)
```

```
In [19]: ▶ plt.figure(figsize=(10,10))
sns.catplot(data=df_object.join(df_num),x='AgeRange',y='Avg Monthly GB Download',
            col='Customer Status',kind='box')
```

Out[19]: <seaborn.axisgrid.FacetGrid at 0x7f792067fbd0>

<Figure size 720x720 with 0 Axes>



Những người trẻ có xu hướng download nhiều hơn người già

```
In [20]: ▶ money_feature = ['Avg Monthly Long Distance Charges', 'Monthly Charge',
                             'Total Refunds', 'Total Extra Data Charges', 'Total Revenue']
```

```
In [21]: fig, axes = plt.subplots(2,3,sharex=True,sharey=False,figsize=(20,10))
df_num.join(df_train['Customer Status']).boxplot(money_feature,'Customer Status',
```

/usr/local/lib/python3.7/dist-packages/pandas/plotting/_matplotlib/boxplot.py:405: UserWarning: When passing multiple axes, sharex and sharey are ignored. These settings must be specified when creating axes

```
**kws,
```

/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.

```
X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```

/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.

```
X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```

/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.

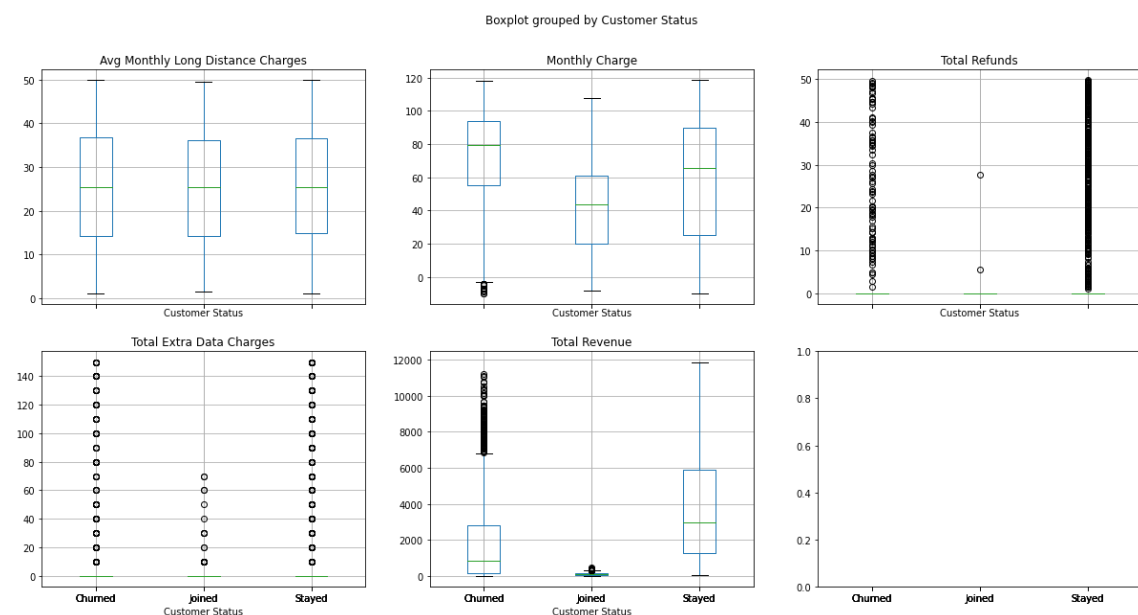
```
X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```

/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.

```
X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```

/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.

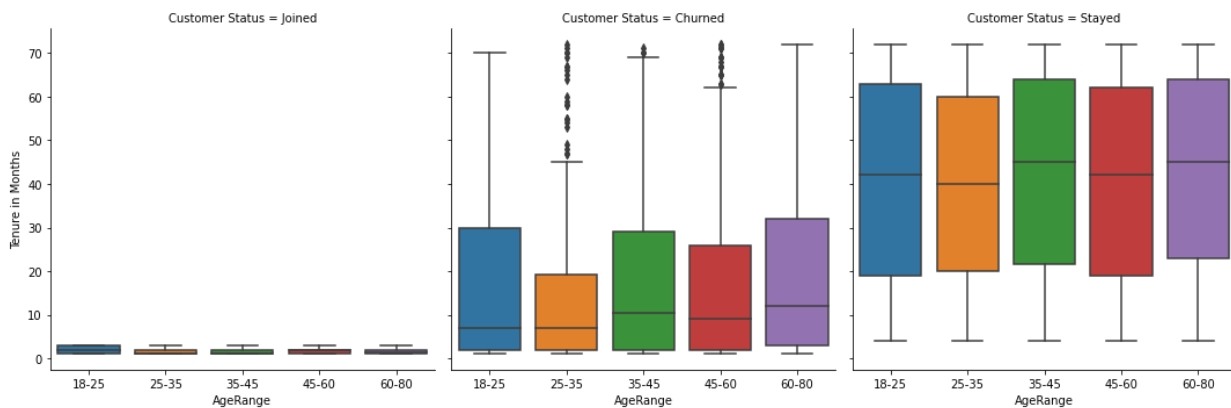
```
X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```



```
In [22]: ▶ plt.figure(figsize=(10,10))
sns.catplot(data=df_object.join(df_num),x='AgeRange',y='Tenure in Months',kind='box')
```

Out[22]: <seaborn.axisgrid.FacetGrid at 0x7f7920634a90>

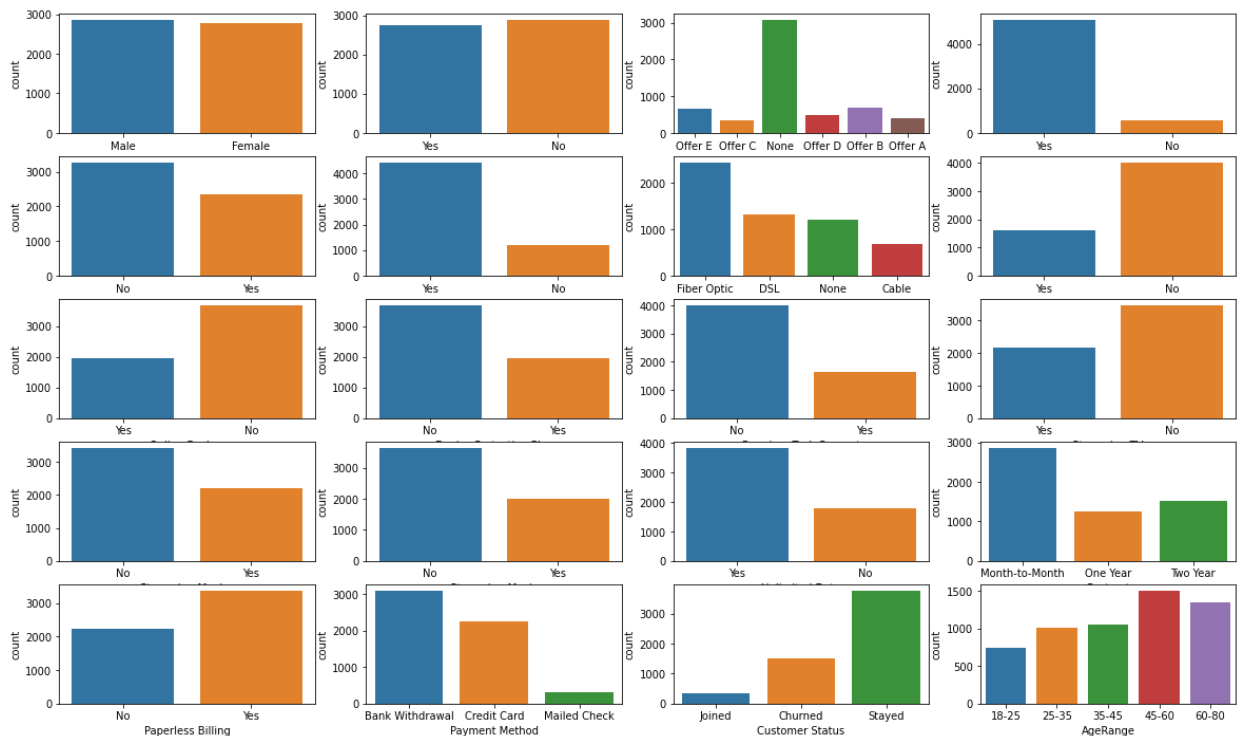
<Figure size 720x720 with 0 Axes>



Những người dời dịch vụ thường có phí tiêu hàng tháng trung bình cao hơn. Và đa phần các khách hàng dùng dịch vụ >30 sẽ trở thành khách hàng trung thành.

Dữ liệu kiểu categorical

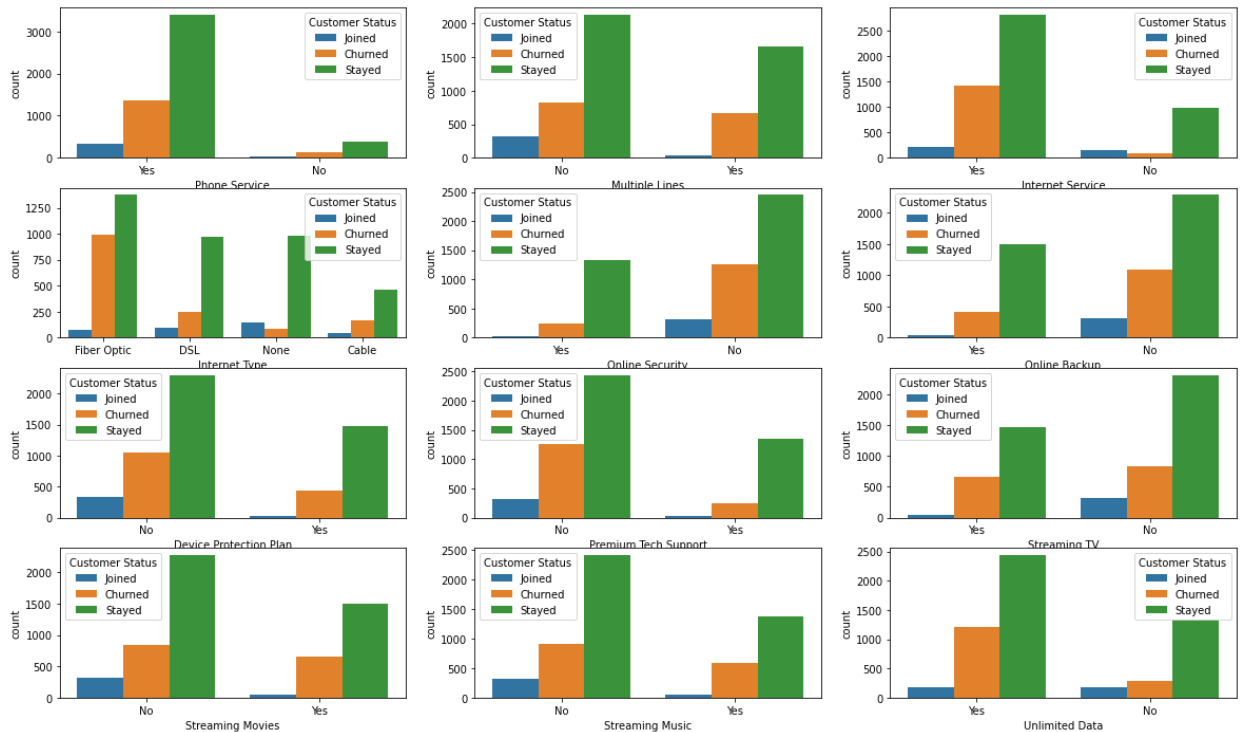
```
In [23]: ▶ #Categorical
fig, axes = plt.subplots(5,4,figsize=(20,12))
for i,col in enumerate(df_object.columns):
    sns.countplot(data=df_object,x=col,ax=axes.flat[i])
```



```
In [24]: ▶ yn_feature = ['Phone Service','Multiple Lines','Internet Service','Internet Type',
                        'Online Security','Online Backup','Device Protection Plan',
                        'Premium Tech Support','Streaming TV','Streaming Movies','Streaming
```



```
In [25]: #Categorical
fig, axes = plt.subplots(4,3,figsize=(20,12))
for i,col in enumerate(yn_feature):
    sns.countplot(data=df_train, x=col, ax=axes.flat[i], hue='Customer Status')
```



3. Tiền xử lý và biểu diễn dữ liệu

3.1. Nội suy dữ liệu

```
In [26]: df_num['AvgMonthlyCharge'] = df_num['Total Revenue']/df_num['Tenure in Months']
```

Vì đây là các trường thông tin đã được tổng hợp và quý II năm 2022. Trong bài toán thực tế, chúng ta cần thông tin nhiều hơn là chỉ 1 tháng. Chúng ta có thể lấy dữ liệu 6-12 tháng trước để dự đoán, những trường có nhiều bản ghi chúng ta có thể nội suy thêm dữ liệu: min, max, avg, độ lệch chuẩn, delta, v.v.

3.2. Xử lý dữ liệu categorical/string

```
In [27]: > # Huấn Luyện mô hình encode
encoder = OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-1)
encoder.fit(df_object[df_object.columns])

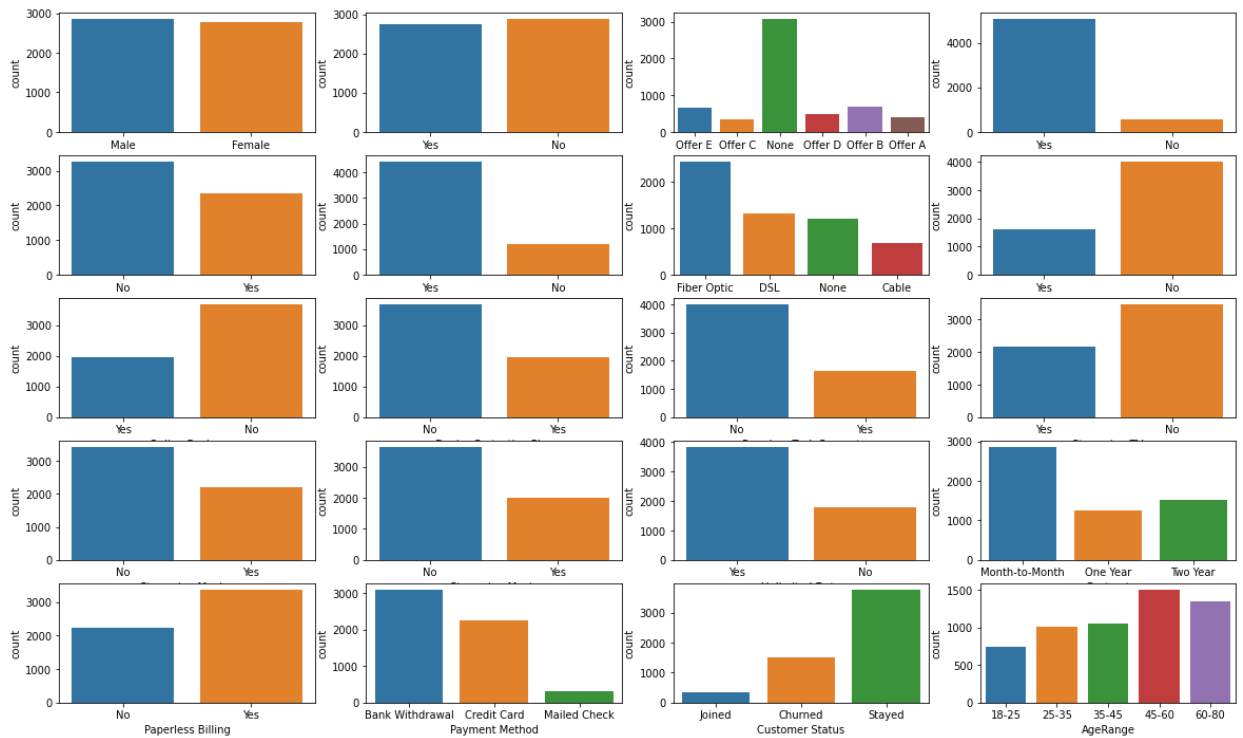
# Lưu mô hình encode Lại để dùng cho lần sau
pickle.dump(encoder, open('string_encoder.pkl', 'wb'), pickle.HIGHEST_PROTOCOL)

# Load Lại hình encode đã được Lưu trước đó
encoder = pickle.load(open('string_encoder.pkl', 'rb'))
```

```
In [28]: > encoder.categories_
```

```
Out[28]: [array(['Female', 'Male'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['None', 'Offer A', 'Offer B', 'Offer C', 'Offer D', 'Offer E'],
dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['Cable', 'DSL', 'Fiber Optic', 'None'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['Month-to-Month', 'One Year', 'Two Year'], dtype=object),
array(['No', 'Yes'], dtype=object),
array(['Bank Withdrawal', 'Credit Card', 'Mailed Check'], dtype=object),
array(['Churned', 'Joined', 'Stayed'], dtype=object),
array(['18-25', '25-35', '35-45', '45-60', '60-80'], dtype=object)]
```

```
In [29]: fig, axes = plt.subplots(5,4,figsize=(20,12))
for i,col in enumerate(df_object.columns):
    sns.countplot(data=df_object,x=col,ax=axes.flat[i])
```



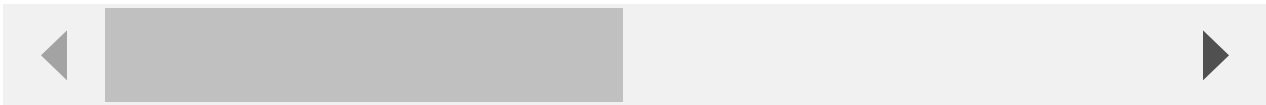
```
In [30]: # Áp dụng mô hình encoder
df_object[df_object.columns] = encoder.transform(df_object[df_object.columns])
```

```
In [31]: df_object
```

Out[31]:

	Gender	Married	Offer	Phone Service	Multiple Lines	Internet Service	Internet Type	Online Security	Online Backup	Device Protection Plan	Pr S
2142	1.0	1.0	5.0	1.0	0.0	1.0	2.0	1.0	1.0	0.0	
1623	1.0	0.0	3.0	1.0	1.0	1.0	2.0	0.0	0.0	0.0	
6074	0.0	1.0	0.0	1.0	1.0	1.0	2.0	0.0	0.0	0.0	
1362	0.0	0.0	0.0	1.0	0.0	1.0	2.0	0.0	0.0	0.0	
6754	0.0	1.0	5.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	
...	
3772	1.0	0.0	0.0	1.0	0.0	0.0	3.0	0.0	0.0	0.0	
5191	0.0	0.0	0.0	1.0	0.0	0.0	3.0	0.0	0.0	0.0	
5226	0.0	0.0	5.0	1.0	1.0	0.0	3.0	0.0	0.0	0.0	
5390	1.0	0.0	5.0	1.0	0.0	0.0	3.0	0.0	0.0	0.0	
860	1.0	1.0	2.0	1.0	1.0	0.0	3.0	0.0	0.0	0.0	

5634 rows × 20 columns



3.2. Xử lý dữ liệu số

Binning

```
In [32]: df_object['AgeRange']
```

Out[32]:

2142	2.0
1623	0.0
6074	3.0
1362	3.0
6754	2.0
...	
3772	3.0
5191	1.0
5226	1.0
5390	1.0
860	2.0

Name: AgeRange, Length: 5634, dtype: float64

```
In [33]: df_num['Age']
```

```
Out[33]: 2142    38
         1623    22
         6074    53
         1362    54
         6754    38
         ..
         3772    51
         5191    35
         5226    33
         5390    35
         860    42
         Name: Age, Length: 5634, dtype: int64
```

Normalization

```
In [34]: def normalize(df, method='minmax'):
         if method == 'standard':
             print('start Standard norming')
             standard_scaler = StandardScaler()
             return standard_scaler.fit_transform(df)
         elif method == 'minmax':
             print('start MinMax norming')
             min_max_scaler = MinMaxScaler()
             return min_max_scaler.fit_transform(df)
         else:
             return None
```

```
In [35]: df_num_norm = normalize(df_num)
```

```
start MinMax norming
```

4. Một số kĩ thuật khác

4.1 Giảm chiều dữ liệu

Theo chiều rộng

```
In [36]: ▶ def tsne_reduction(data):
    tsne = TSNE(n_components=2)
    _2d_data = tsne.fit_transform(data)

    return _2d_data

def pca_reduction(data):
    pca = PCA(n_components=2)
    _2d_data = pca.fit_transform(data)

    return _2d_data

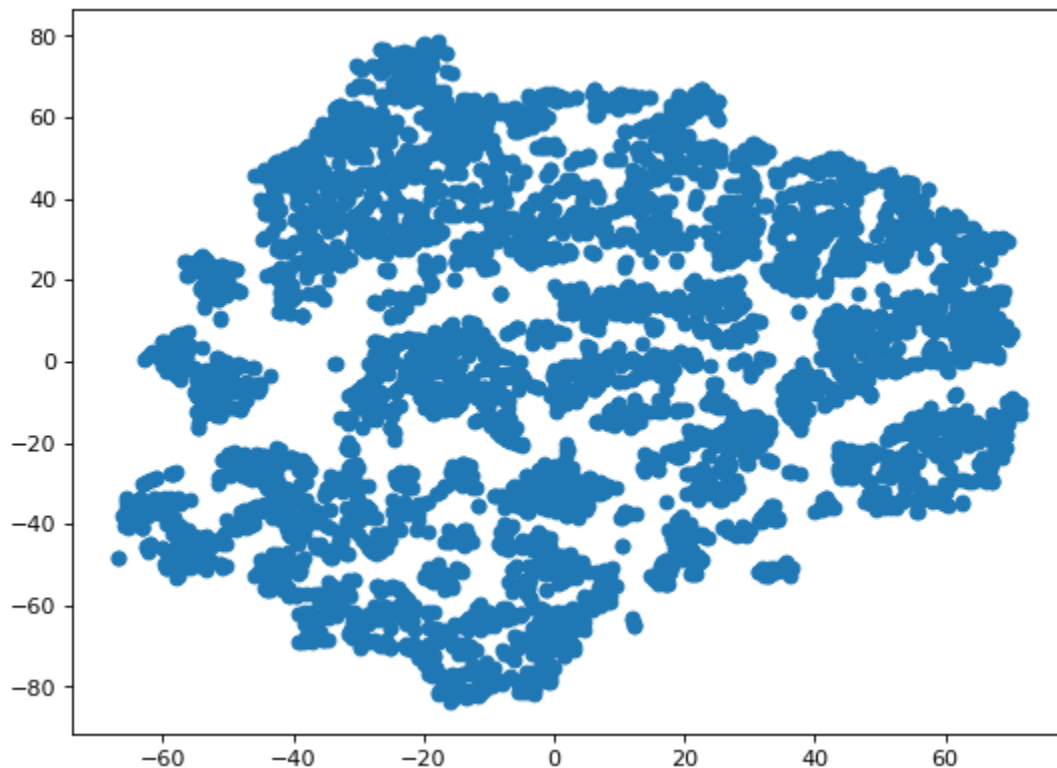
def scatter_2d(_2d_data):
    plt.figure(figsize=(8, 6), dpi=80)
    plt.scatter(_2d_data[:, 0], _2d_data[:, 1])
    plt.grid(False)
    plt.show()
```

```
In [37]: ▶ tsne_num_2d_data = tsne_reduction(normalize(df_num))
    scatter_2d(tsne_num_2d_data)
```

start MinMax norming

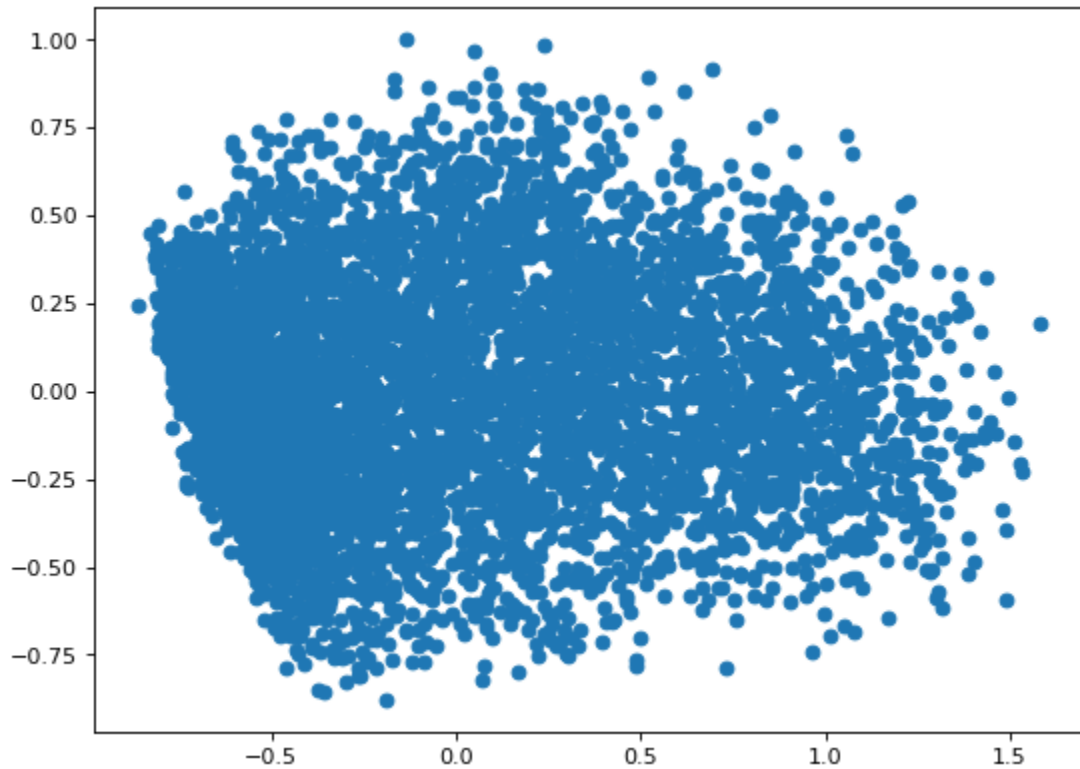
/usr/local/lib/python3.7/dist-packages/sklearn/manifold/_t_sne.py:783: FutureWarning: The default initialization in TSNE will change from 'random' to 'pca' in 1.2.

FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/manifold/_t_sne.py:793: FutureWarning: The default learning rate in TSNE will change from 200.0 to 'auto' in 1.2.
FutureWarning,



```
In [38]: ▶ pca_num_2d_data = pca_reduction(normalize(df_num))
scatter_2d(pca_num_2d_data)
```

start MinMax norming



Theo chiều sâu

```
In [39]: ▶ df_num.duplicated().any()
```

Out[39]: False

4.2. Ném dữ liệu:

Thường sử dụng để xử lý dữ liệu thưa

VD: biểu diễn text:

- one-hot
- w2v

4.3. Tổng quát hóa

Thường sử dụng với dữ liệu thưa

VD: biểu diễn one-hot tuổi -> khoảng tuổi như trên

VD: abc@gmail.com (<mailto:abc@gmail.com>), dsg@gmail.com (<mailto:dsg@gmail.com>) -> 'email'

tên riêng, tên tổ chức, v.v.

5. Bài tập về nhà

Thực hiện các công việc tương tự với bài tập lớn.