MỤC LỤC

Lời nói đầu

Sự phát triển của công nghệ thông tin và việc ứng dụng công nghệ thông tin trong nhiều lĩnh vực của đời sống, kinh tế xã hội trong nhiều năm qua cũng đồng nghĩa với lượng dữ liệu đã được các cơ quan thu thập và lưu trữ ngày một tích luỹ nhiều lên. Họ lưu trữ các dữ liệu này vì cho rằng trong nó ẩn chứa những giá trị nhất định nào đó. Mặt khác, trong môi trường cạnh tranh, người ta ngày càng cần có nhiều thông tin với tốc độ nhanh để trợ giúp việc ra quyết định và ngày càng có nhiều câu hỏi mang tính chất định tính cần phải trả lời dựa trên một khối lượng dữ liệu khổng lồ đã có. Với những lý do như vậy, các phương pháp quản trị và khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế đã làm phát triển một khuynh hướng kỹ thuật mới đó là Kỹ thuật phát hiện tri thức và khai phá dữ liệu.

Khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau ở các nước trên thế giới, tại Việt Nam kỹ thuật này tương đối còn mới mẻ tuy nhiên cũng đang được nghiên cứu và dần đưa vào ứng dụng. Khai phá dữ liệu là một bước trong qui trình phát hiện tri thức gồm có các thuật toán khai thác dữ liệu chuyên dùng dưới một số qui định về hiệu quả tính toán chấp nhận được để tìm ra các mẫu hoặc các mô hình trong dữ liệu. Nói một cách khác, mục đích của phát hiện tri thức và khai phá dữ liệu chính là tìm ra các mẫu và/hoặc các mô hình đang tồn tại trong các cơ sở dữ liệu nhưng vẫn còn bị che khuất bởi hàng núi dữ liệu.

Trong bài viết này, em sẽ trình bày một cách tổng quan về Kỹ thuật khai phá dữ liệu. Trên cơ sở đó đưa ra một bài toán dự báo về khả năng chơi thể thao dựa vào thời tiết và giải quyết bài toán bằng phương pháp phân lớp nhằm cung cấp cho bạn đọc một cách nhìn khái quát về kỹ thuật mới này cũng như mối tương quan với phương pháp thống kê truyền thống.

I. Lý do chọn đề tài

Black Friday là ngày của mua sắm vào ngày thứ 6 ngay sau ngày Lễ Tạ Ơn và được coi là ngày mở hàng cho mùa mua sắm tấp nập nhất ở Mỹ.Trong ngày này hầu hết các cửa hàng đều mở cửa rất sớm và kèm theo những khuyến mãi hấp dẫn.Bởi vì điều đó mà có rất nhiều người đi mua sắm vào ngày này và dẫn tới tình trạng kẹt xe và hàng trăm người chen chút đen kịt trên các con phố và vỉa hè để tranh nhau mua sắm chuẩn bị cho Lễ Noel sắp đến.Nắm bắt được thị hiếu và nhu cầu lớn của người tiêu dùng vào thời gian này nên các cửa hàng quảng cáo rầm rộ trên các phương tiện truyền thông,đồng loạt giảm giá, khuyến mã để thu hút khách hàng.Nhận thấy được sự lan rộng và tiềm năng từ ngày này nên nhóm em chọn khai thác dữ liệu của một cửa hàng bán lẻ trong ngày này để hiểu hơn về ngày Black Friday.

II. Thông tin dữ liệu

- Tên dataset : Black Friday
- Data sourses: BlackFriday.csv
- Nguồn dataset: https://www.kaggle.com/mehdidag/black-friday
- Mô tả dữ liệu:

  Dataset này là dữ liệu các giao dịch được thu thập trong một cửa hàng bán lẻ. Cửa hàng muốn biết thêm về hành vi mua hàng của khách hàng đối với những sản phẩm khác nhau.
- Thông số dataset:
  - Số thuộc tính:12
  - Số dòng:537577
  - Dung lượng file:5MB
  - Bảng thông tin của thuộc tính

STT	Tên thuộc tính	Kiểu dữ liệu	Loại dữ liệu	Mô tả thuộc tính	Phân khúc
1	User_ID	Numeric	Discrete	Id khách hàng	Khách hàng
2	Product_ID	Numeric	Discrete	Id sản phẩm	Sản phẩm
3	Gender	Categorical	Nominal	Giới tính	Khách hàng
4	Age	Categorical	Ordinal	Tuổi	Khách hàng
5	Occupation	Categorical	Nominal	Nghề nghiệp (Được giấu)	Khách hàng
6	City_Category	Categorical	Ordinal	Danh mục thành phố(A,B,C)	Thành phố
7	Stay_In_Current_City_Years	Categorical	Ordinal	Số năm ở tại thành phố hiện tại	Thành phố
8	Marital_Status	Categorical	Ordinal	Tình trạng hôn nhân	Khách hàng
9	Product_Category_1	Categorical	Nominal	Danh mục sản phẩm(được giấu)	Sản phẩm
10	Product_Category_2	Categorical	Nominal	Sản phẩm cũng có thể thuộc danh mục khác(được giấu)	Sản phẩm
11	Product_Category_3	Categorical	Nominal	Sản phẩm cũng có thể thuộc danh mục khác(được giấu)	Sản phẩm
12	Purchase	Numeric	Continuous	Số tiền mua	Sản phẩm

III. Mô tả bài toán

Với dữ liệu được dùng cho đồ án là thông tin các giao dịch của một của hàng bán lẻ vào ngày Black Friday(file BlackFriday.csv) nhóm tiền hành đặt ra bài toán cần giải quyết với dữ liệu trên.Tiến hành phân tích dữ liệu cho việc nghiên cứu:

- Ai là người chi tiêu nhiều hơn trong ngày Black Friday?
  - Nam hay Nữ
  - Đã kết hôn hay còn độc thân
  - Cư dân cũ hay cư dân mới
- Loại sản phẩm nào được bán nhiều trong ngày Black Friday?

và áp dụng các thuật toán trong khai thác dữ liệu để đưa ra các dự đoán như "Dự đoán số tiền mua của khách hàng dựa trên các dữ liệu còn lại trong dataset".

IV. Thuật toán khai thác dữ liệu

1. Cấu trúc dữ liệu

1.1 Nhập thư viện và đọc dữ liệu

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import os
print(os.listdir("../test/input"))
dataset=pd.read_csv('../test/input/BlackFriday.csv')
dataset.head()
```

Kết quả:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Produc
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN	
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN	
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	
4	1000002	P00285442	M	55+	16	C	4+	0	8	NaN	

1.2 Xem thông tin DataFrame

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 537577 entries, 0 to 537576
Data columns (total 12 columns):
User_ID                     537577 non-null int64
Product_ID                  537577 non-null object
Gender                      537577 non-null object
Age                         537577 non-null object
Occupation                  537577 non-null int64
City_Category               537577 non-null object
Stay_In_Current_City_Years  537577 non-null object
Marital_Status              537577 non-null int64
Product_Category_1          537577 non-null int64
Product_Category_2          370591 non-null float64
Product_Category_3          164278 non-null float64
Purchase                    537577 non-null int64
dtypes: float64(2), int64(5), object(5)
memory usage: 39.0+ MB
```

1.3 Xem các thống kê cơ bản của dữ liệu

```
dataset.describe()
```

	User_ID	Occupation	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
count	5.375770e+05	537577.00000	537577.000000	537577.000000	370591.000000	164278.000000	537577.000000
mean	1.002992e+06	8.08271	0.408797	5.295546	9.842144	12.669840	9333.859853
std	1.714393e+03	6.52412	0.491612	3.750701	5.087259	4.124341	4981.022133
min	1.000001e+06	0.00000	0.000000	1.000000	2.000000	3.000000	185.000000
25%	1.001495e+06	2.00000	0.000000	1.000000	5.000000	9.000000	5866.000000
50%	1.003031e+06	7.00000	0.000000	5.000000	9.000000	14.000000	8062.000000
75%	1.004417e+06	14.00000	1.000000	8.000000	15.000000	16.000000	12073.000000
max	1.006040e+06	20.00000	1.000000	18.000000	18.000000	18.000000	23961.000000

1.4 Kiểm tra giá trị thiếu

```
dataset.isna().any()
```

```
User_ID                       False
Product_ID                    False
Gender                        False
Age                           False
Occupation                    False
City_Category                 False
Stay_In_Current_City_Years    False
Marital_Status                False
Product_Category_1            False
Product_Category_2             True
Product_Category_3             True
Purchase                      False
dtype: bool
```

2. Phân tích dữ liệu

2.1. Phân tích từng biến độc lập

2.1.1. Phân tích các biến là số

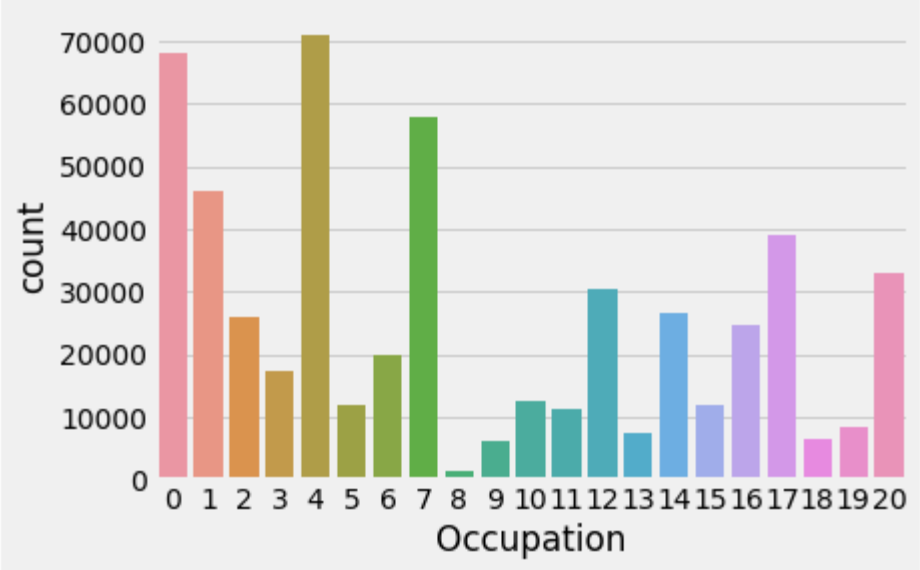
```
numeric_features = dataset.select_dtypes(include=[np.number])
numeric_features.dtypes
```

```
User_ID                 int64
Occupation              int64
Marital_Status          int64
Product_Category_1      int64
Product_Category_2    float64
Product_Category_3    float64
Purchase                int64
dtype: object
```

2.1.1.1. Biến Occupation

```
sns.countplot(dataset.Occupation)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xfd017b0>
```

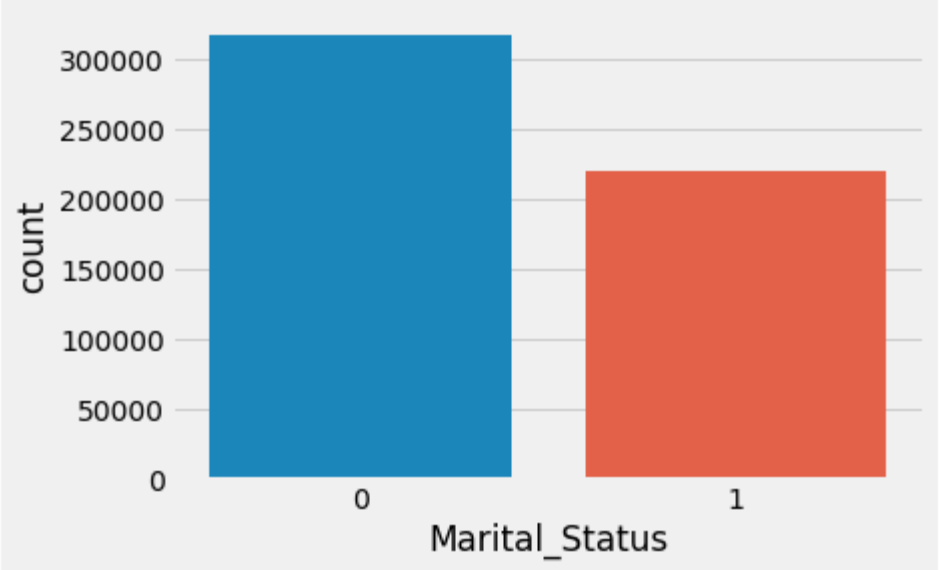


Như chúng ta thấy ở biểu đồ trên, Occupation có đến 20 giá trị khác nhau.Đây là 20 nghề nghiệp khác nhau ứng với từng con số mà dữ liệu đã giấu đi.Ta thấy được trong ngày Black Friday thì nghề nghiệp của khách hàng rất đa dạng không phân biệt ngành nghề nào(ở đây là có 20 loại nghề nghiệp).

2.1.1.2. Biến Marital_Status

```
sns.countplot(dataset.Marital_Status)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xfd6b790>
```

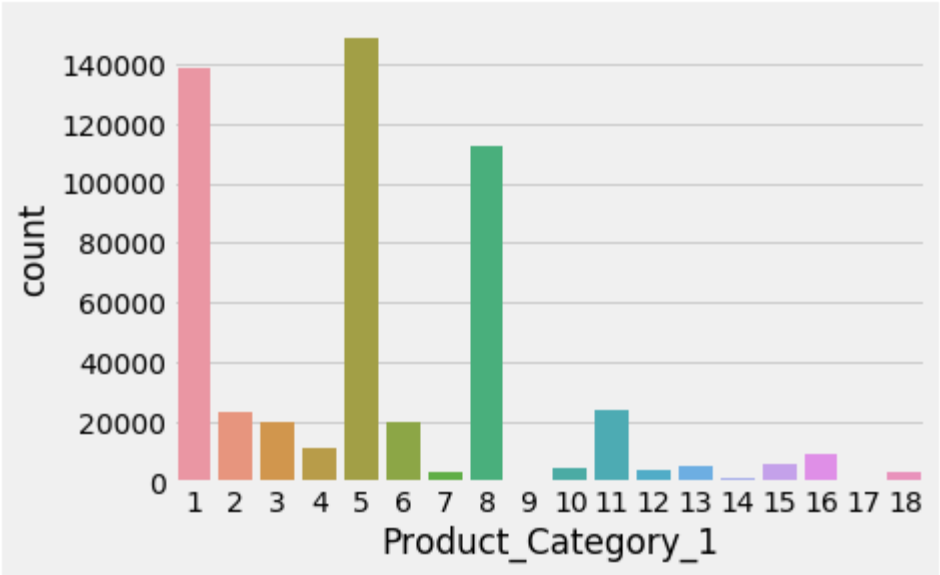


Ta thấy được rõ ràng nhóm người độc thân mua sắm trong ngày Black Friday nhiều hơn nhóm người đã kết hôn.

2.1.1.3. Biến Product_Category_1

```
sns.countplot(dataset.Product_Category_1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xfd7e2f0>
```

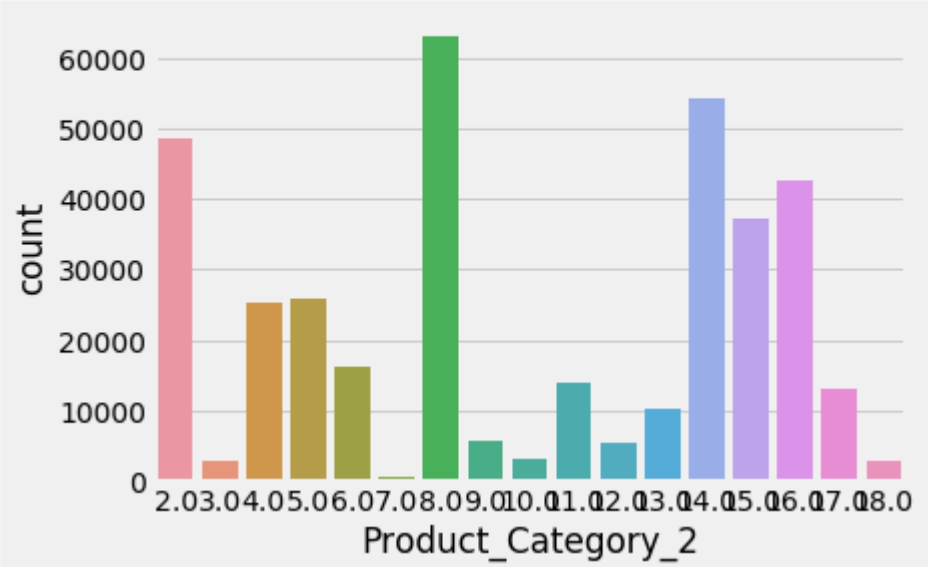


Trong danh mục sản phẩm thứ 1 thì có 3 sản phẩm nổi bật nhất là 1 , 5 và 8.

2.1.1.4. Biến Product_Category_2

```
sns.countplot(dataset.Product_Category_2)
```

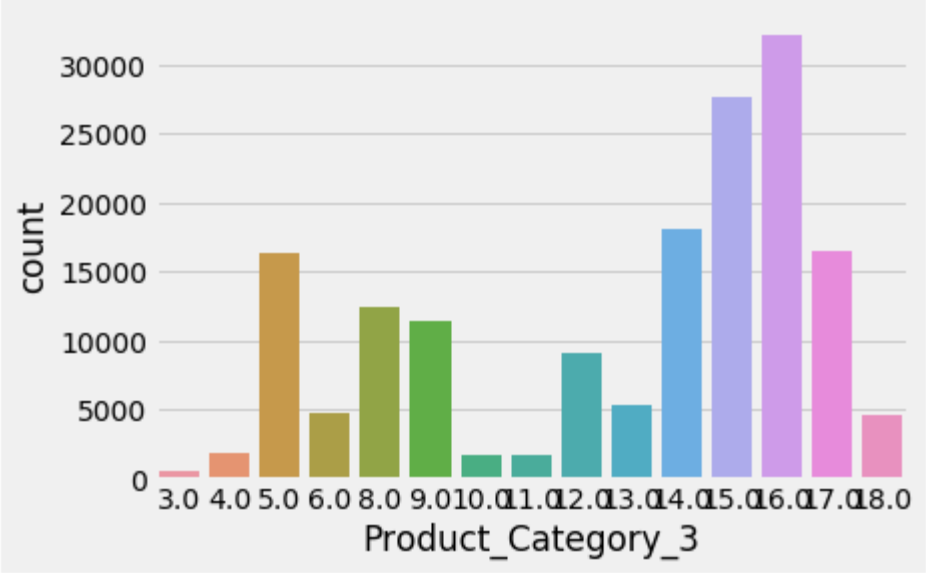
```
<matplotlib.axes._subplots.AxesSubplot at 0xfbf4ab0>
```



2.1.1.5. Biến Product_Category_3

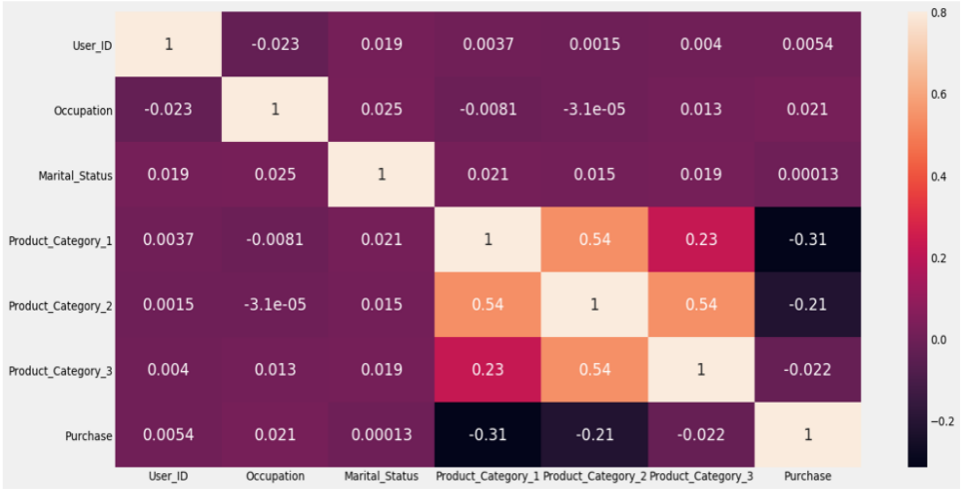
```
sns.countplot(dataset.Product_Category_3)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x976f30>
```



2.1.1.6. Mối tương quan giữa các biến là số và biến Purchase

```
corr = numeric_features.corr()
f, ax = plt.subplots(figsize=(20, 9))
sns.heatmap(corr, vmax=.8,annot_kws={'size': 20}, annot=True);
```



Ta thấy được là không có bất cứ biến nào có ảnh hưởng cao đối với biến Purchase, trong đó hệ số tương quan cao nhất là

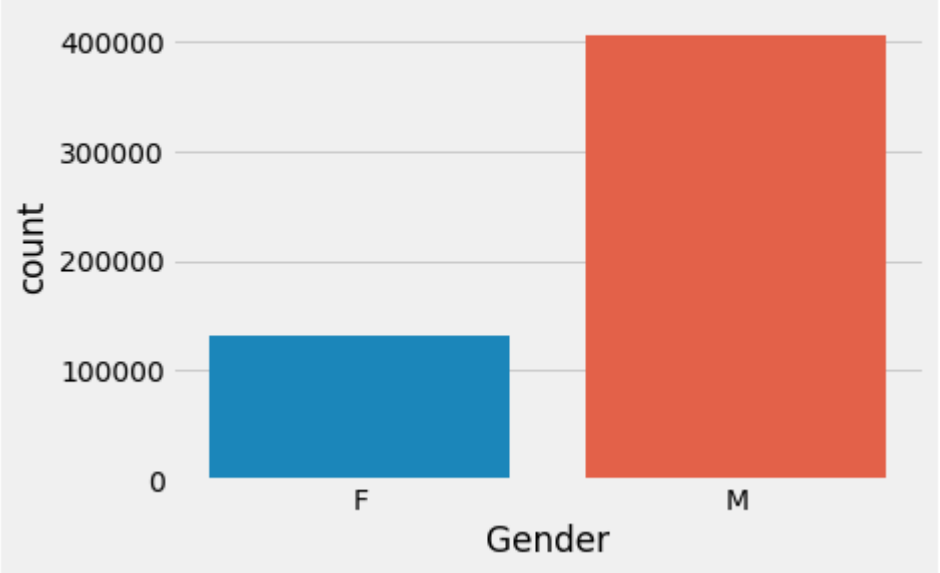
của biến 'Occupation' là 0,021.Mặt khác thì các biến 'Product_Category_1', 'Product_Category_2' và Product_Category_3' lại tương quan nghịch với biến Purchase với hệ số tương quan là -0.31 (Product_Category_1), -0.21 (Product_Category_2) , -0.022 (Product_Category_3).

2.1.2 Phân tích các biến phân loại

2.1.2.1 Biến Gender

```
sns.countplot(dataset.Gender)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1000ee90>
```

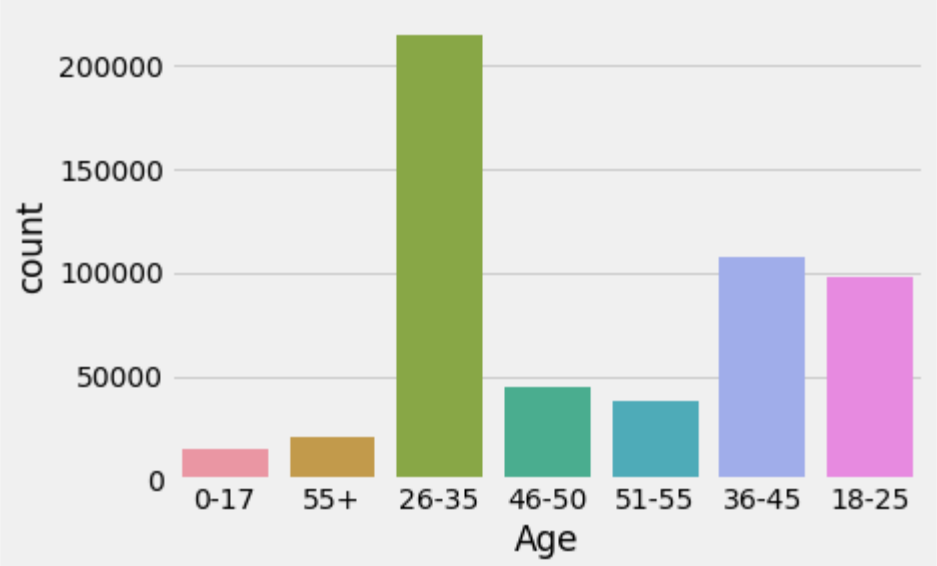


Nhìn vào biểu đồ ta thấy hầu hết người mua sắm là Nam.Nhưng cũng có thể là người mua là Nữ mà người trả tiền lại là người chồng của họ và ngược lại.

2.1.2.2 Biến Age

```
sns.countplot(dataset.Age)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x130b4c70>
```

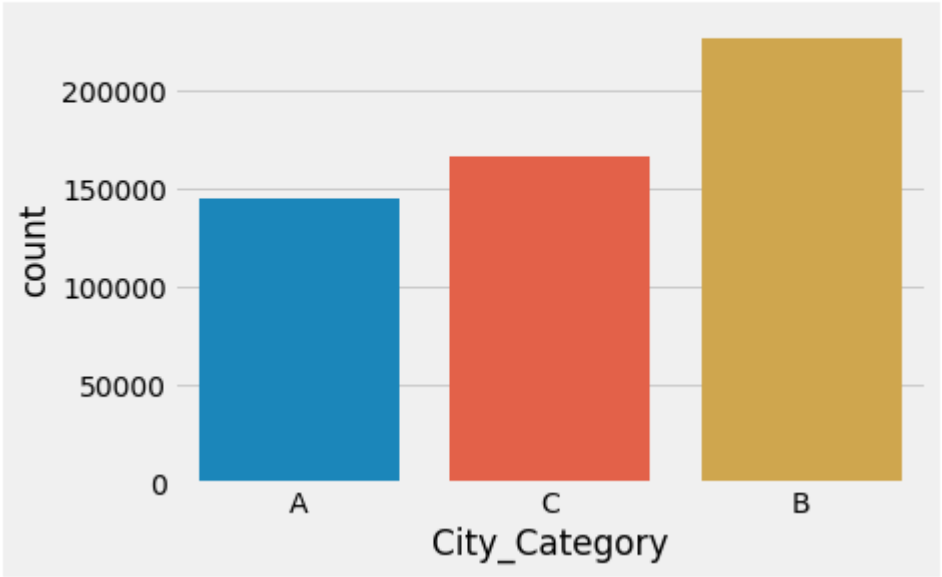


Hầu hết các giao dịch được thực ở những khách hàng có độ tuổi từ 18 tới 45 .

2.1.2.3 Biến City_Category

```
sns.countplot(dataset.City_Category)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x130ef950>
```

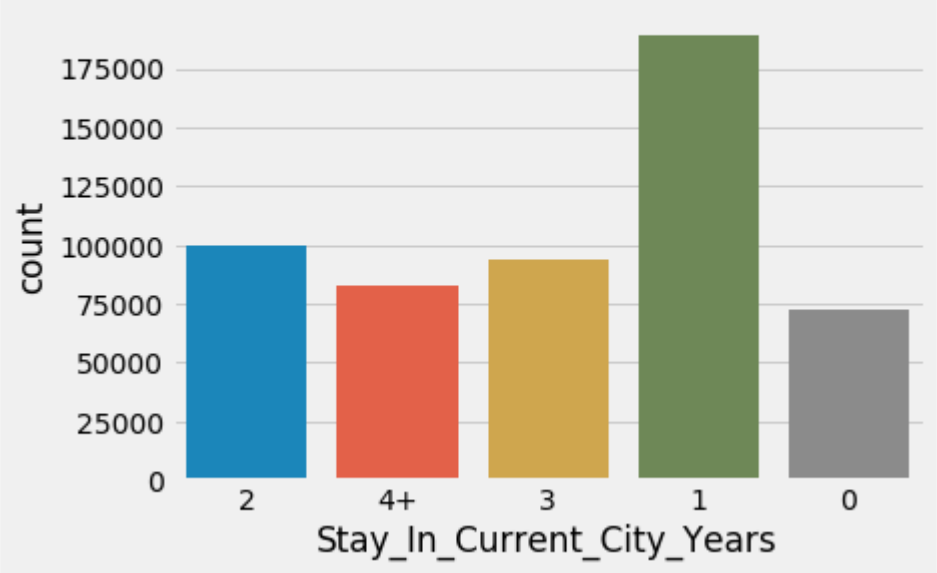


Ta thấy sức mua sắm ở thành phố B là cao hơn A và C.Do không biết chính xác cụ thể tên của từng thành phố nên khó kết luật được.

2.1.2.4 Biến Stay_In_Current_City_Years

```
sns.countplot(dataset.Stay_In_Current_City_Years)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x126f2cb0>
```

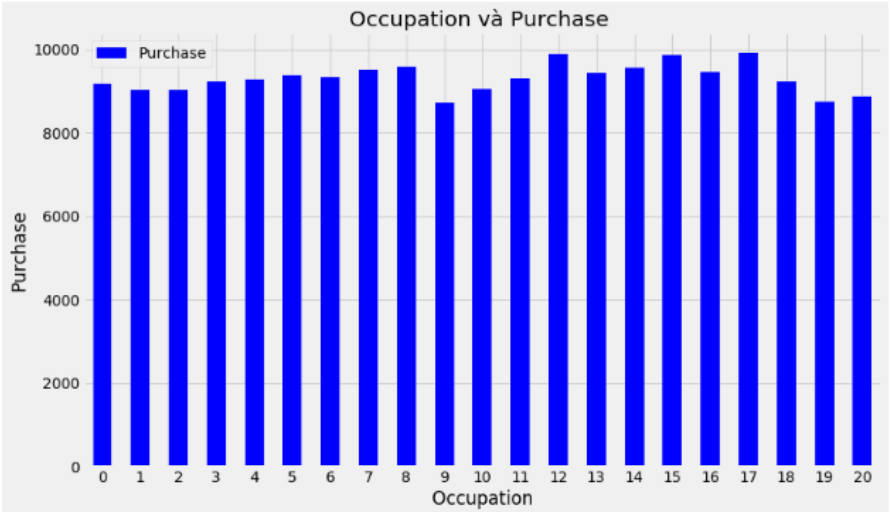


Từ biểu đồ ta thấy ,những người ở lâu năm tại thành phố đó thì xu hướng mua đồ mới thì ít hơn những người mới chuyển tới.Những người mới chuyển tới và ở đó 1 năm thì sẽ có nhu cầu mua sắm cao hơn nhân ngày Black Friday để mua sắm những vật dụng mới cho căn hộ,đồ cá nhân cho họ.

2.2 Phân tích từng biến với biến phụ thuộc (Purchase)

2.2.1 Occupation và Purchase

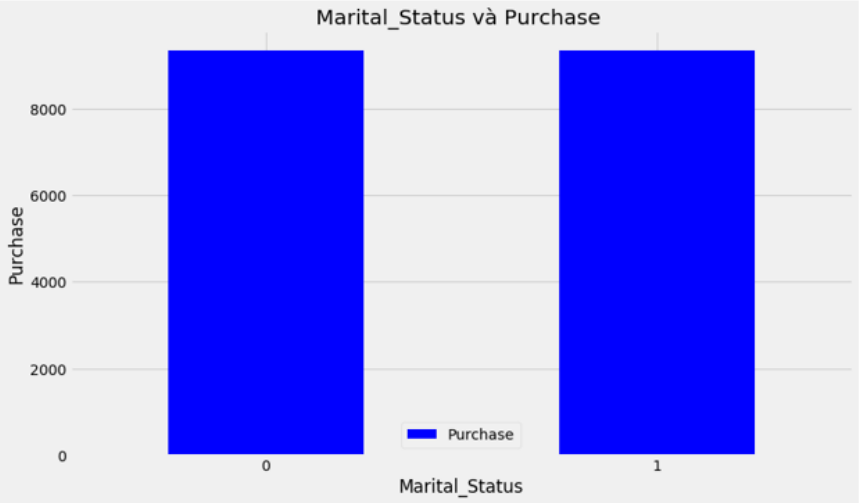
```
Occupation_pivot = dataset.pivot_table(index='Occupation', values='Purchase', aggfunc=np.mean)
Occupation_pivot.plot(kind='bar', color='blue',figsize=(12,7))
plt.xlabel('Occupation')
plt.ylabel('Purchase')
plt.title('Occupation và Purchase ')
plt.xticks(rotation=0)
plt.show()
```



Mặc dù có một vài nghề nghiệp có ảnh hưởng cao tới Purchase nhưng ta thấy được số tiền trung bình mà mỗi user bỏ ra thì gần giống nhau.Cuối cùng thì thuộc tính Occupation cũng có ảnh hưởng cao tới Purchase.

2.2.2 Marital_Status và Purchase

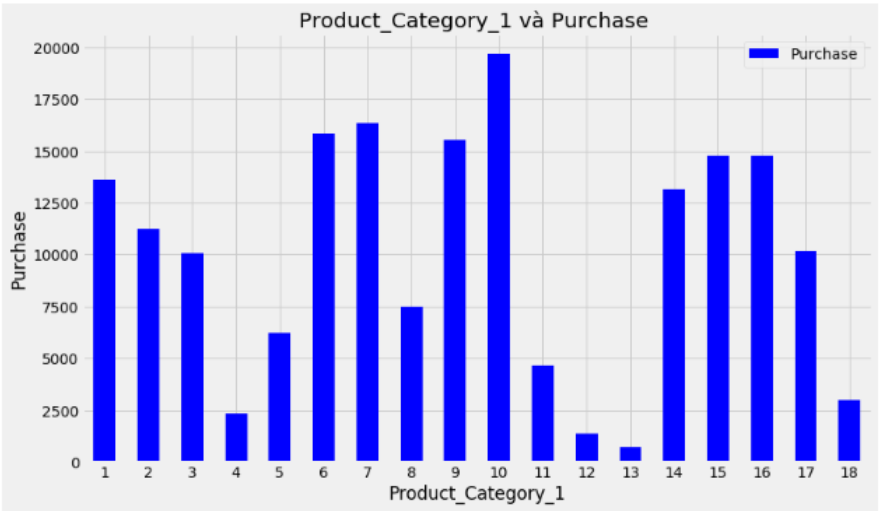
```
Marital_Status_pivot = dataset.pivot_table(index='Marital_Status', values='Purchase', aggfunc=np.mean)
Marital_Status_pivot.plot(kind='bar', color='blue',figsize=(12,7))
plt.xlabel('Marital_Status')
plt.ylabel('Purchase')
plt.title('Marital_Status và Purchase ')
plt.xticks(rotation=0)
plt.show()
```



Ta có thể thấy là số lượng đối tượng độc thân thì nhiều hơn là đã kết hôn.Nhưng trung bình mỗi cá nhân thì lại có xu hướng trả cùng một số tiền mà không phụ thuộc vào đã kết hôn hay chưa.

2.2.3 Product_Category_1 và Purchase

```
Product_Category_1_pivot = dataset.pivot_table(index='Product_Category_1', values='Purchase', aggfunc=np.mean)
Product_Category_1_pivot.plot(kind='bar', color='blue',figsize=(12,7))
plt.xlabel('Product_Category_1')
plt.ylabel('Purchase')
plt.title('Product_Category_1 và Purchase ')
plt.xticks(rotation=0)
plt.show()
```



Như phân tích trên về biến Product_Category_1 thì sản phẩm 1, 5 và 8 thì được mua rất nhiều.Nhưng trung bình chi phí bỏ ra để mua các sản phẩm này thì lại không cao.Mặc khác thì các sản phẩm được mua ít hơn thì lại có giá trị Purchase là cao hơn.

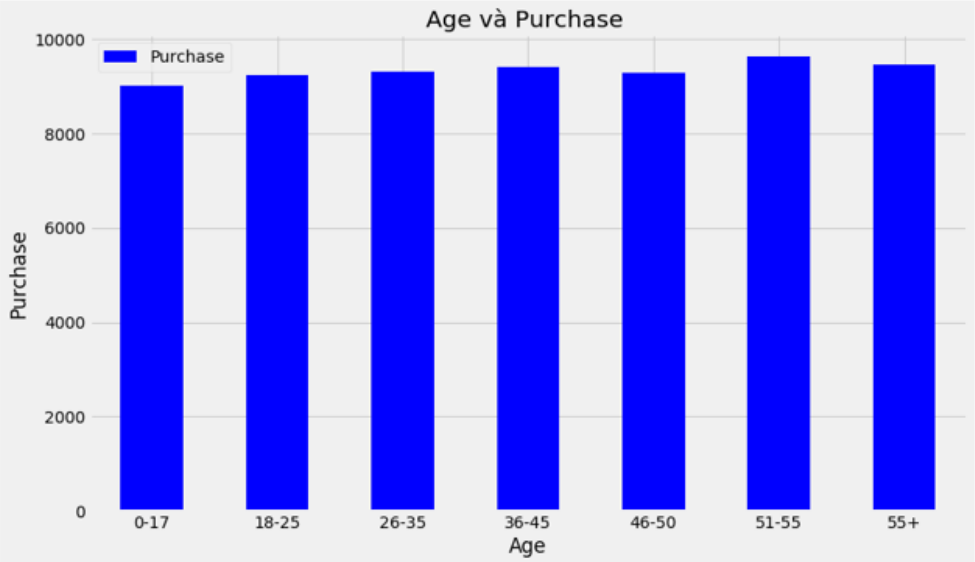
2.2.4 Gender và Purchase

```
Gender_pivot = dataset.pivot_table(index='Gender', values='Purchase', aggfunc=np.mean)
Gender_pivot.plot(kind='bar', color='blue',figsize=(12,7))
plt.xlabel('Gender')
plt.ylabel('Purchase')
plt.title('Gender và Purchase ')
plt.xticks(rotation=0)
plt.show()
```

Ta cũng có thể thấy trung bình số tiền bỏ ra đề mua sắm của giới tính Nam vẫn cao hơn Nữ.

2.2.5 Age và Purchase

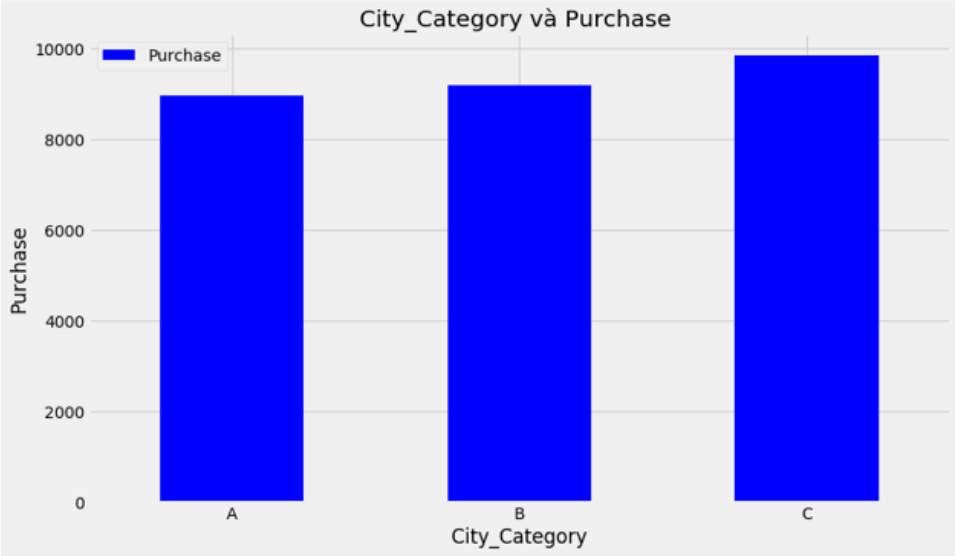
```
Age_pivot = dataset.pivot_table(index='Age', values='Purchase', aggfunc=np.mean)
Age_pivot.plot(kind='bar', color='blue',figsize=(12,7))
plt.xlabel('Age')
plt.ylabel('Purchase')
plt.title('Age và Purchase ')
plt.xticks(rotation=0)
plt.show()
```



Ta có thể thấy trung bình số tiền bỏ ra của từng nhóm tuổi gần như bằng nhau.Đặt biệt nhóm tuổi trên 50 lại có số tiền trung bình cao nhất.

2.2.6 City_Category và Purchase

```
City_Category_pivot = dataset.pivot_table(index='City_Category', values='Purchase', aggfunc=np.mean)
City_Category_pivot.plot(kind='bar', color='blue',figsize=(12,7))
plt.xlabel('City_Category')
plt.ylabel('Purchase')
plt.title('City_Category và Purchase ')
plt.xticks(rotation=0)
plt.show()
```



Như ta đã biết thì số lượng người mua ở thành phố B là nhiều nhất nhưng ở thành phố C thì người mua lại có số tiền bỏ ra là cao nhất.

2.2.7 Stay_In_Current_City_Years và Purchase

```
Stay_In_Current_City_Years_pivot = dataset.pivot_table(index='Stay_In_Current_City_Years', values='Purchase', aggfunc=np.mean)
Stay_In_Current_City_Years_pivot.plot(kind='bar', color='blue',figsize=(12,7))
plt.xlabel('Stay_In_Current_City_Years')
plt.ylabel('Purchase')
plt.title('Stay_In_Current_City_Years và Purchase ')
plt.xticks(rotation=0)
plt.show()
```

Ta cũng có thể thấy là có sự khác biệt giữa những người mới ở tại thành phố và những người ở lâu năm nhưng với trung bình chi phí bỏ ra thì lại không có sự khác biệt nhiều giữa các nhóm.

3. Tiền xử lý dữ liệu

Sau khi phân tích dữ liệu,rút ra một số kết luận:

- **Age:** Thuộc tính Age hiện tại là gồm những nhóm tuổi.Có thể biến đổi thành những con số thay thế các nhóm tuổi.
- **City_Category:** Có thể chuyển các danh mục thành phố ‘A’ , ‘B’ , ‘ C’ thành những con số thay thế.
- **Gender:** Có 2 kiểu giới tính là ‘M’ và ’F’.Có thể chuyển sang dạng Binary
- **Stay_In_Current_City_Years:** Có các giá trị ‘0’ , ’1’ , ’2’ , ’3’ , ’4+’. Có thể xóa dấu ‘+’ .
- **Product_Category_2** and **Product_Category_3:** Có giá trị thiếu.

Các kỹ thuật tiền xử lý dữ liệu được chọn:

3.1. Làm sạch dữ liệu (data cleaning)

3.1.1. Tổng quát về kỹ thuật

Đối với dữ liệu thu thập được, cần xác định các vấn đề ảnh hưởng là cho nó không sạch. Bởi vì, dữ liệu không sạch (có chứa lỗi, nhiễu, không đầy đủ, có mâu thuẫn) thì các tri thức khám phá được sẽ bị ảnh hưởng và không đáng tin cậy, sẽ dẫn đến các quyết định không chính xác. Do đó, cần gán các giá trị thuộc tính còn thiếu;

sửa chữa các dữ liệu nhiễu/lỗi; xác định hoặc loại bỏ các ngoại lai (outliers); giải quyết các mâu thuẫn dữ liệu.

Các vấn đề của dữ liệu:

- Không hoàn chỉnh (incomplete)
- Nhiễu/lỗi (noise/error)
- Mâu thuẫn (inconsistent)

3.1.2. Thực hiện trên dataset

Áp dụng kỹ thuật làm sạch dữ liệu đối với thuộc tính ‘Product_Category_2’ and ‘Product_Category_3’.Có thể là do khách hàng không mua những sản phẩm trong 2 danh mục này nên dữ liệu bị thiếu.Ta có thể thay thế các giá trị thiếu này bằng 0.

```
dataset.isna().any()
```

```
User_ID                     False
Product_ID                  False
Gender                      False
Age                         False
Occupation                  False
City_Category               False
Stay_In_Current_City_Years  False
Marital_Status              False
Product_Category_1          False
Product_Category_2           True
Product_Category_3           True
Purchase                    False
dtype: bool
```

```
dataset.fillna(value=0,inplace=True)
```

Kết quả:

Ta có thể thấy số lượng ở 2 thuộc tính ‘Product_Category_2’ and ‘Product_Category_3’ đã bằng với số lường dòng của dataset.

```
dataset.describe()
```

	User_ID	Occupation	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
count	5.375770e+05	537577.00000	537577.000000	537577.000000	537577.000000	537577.000000	537577.000000
mean	1.002992e+06	8.08271	0.408797	5.295546	6.784907	3.871773	9333.859853
std	1.714393e+03	6.52412	0.491612	3.750701	6.211618	6.265963	4981.022133
min	1.000001e+06	0.00000	0.000000	1.000000	0.000000	0.000000	185.000000
25%	1.001495e+06	2.00000	0.000000	1.000000	0.000000	0.000000	5866.000000
50%	1.003031e+06	7.00000	0.000000	5.000000	5.000000	0.000000	8062.000000
75%	1.004417e+06	14.00000	1.000000	8.000000	14.000000	8.000000	12073.000000
max	1.006040e+06	20.00000	1.000000	18.000000	18.000000	18.000000	23961.000000

Và ở những giá trị bị thiếu đã được thay bằng giá trị 0.0.

```
dataset.head()
```

)	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
2	F	0-17	10	A	2	0	3	0.0	0.0	8370
2	F	0-17	10	A	2	0	1	6.0	14.0	15200
2	F	0-17	10	A	2	0	12	0.0	0.0	1422
2	F	0-17	10	A	2	0	12	14.0	0.0	1057
2	M	55+	16	C	4+	0	8	0.0	0.0	7969

3.2. Biến đổi dữ liệu (data transformation)

3.2.1. Tổng quát về kỹ thuật

Biến đổi dữ liệu là việc chuyển toàn bộ tập giá trị của một thuộc tính sang một tập các giá trị thay thế, sao cho mỗi giá trị cũ tương ứng với một trong các giá trị mới.

Các phương pháp biến đổi dữ liệu:

- Làm trơn (smoothing): Loại bỏ nhiễu/lỗi khỏi dữ liệu.
- Kết hợp (aggregation): Sự tóm tắt dữ liệu, xây dựng các khối dữ liệu.
- Khái quát hóa (generalization): Xây dựng các phân cấp khái niệm.

- Chuẩn hóa (normalization): Đưa các giá trị về một khoảng được chỉ định

3.2.2. Thực hiện trên dataset

3.2.2.1. Chuyển thuộc tính Age sang giá trị số

```
age_dict = {'0-17':0,'18-25':1,'26-35':2,'36-45':3,'46-50':4,'51-55':5,'55+':6, }
dataset['Age']=dataset['Age'].apply(lambda line: age_dict[line])
dataset['Age'].value_counts()
```

```
2    214690
3    107499
1     97634
4     44526
5     37618
6     20903
0     14707
Name: Age, dtype: int64
```

Kết quả:

- Kiểu dữ liệu của Age đã chuyển sang int64
- Các nhóm tuổi qua chuyển sang các số tương ứng

```
dataset.head(10)
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Produ
0	1000001	P00069042	F	0	10	A	2	0	3	
1	1000001	P00248942	F	0	10	A	2	0	1	
2	1000001	P00087842	F	0	10	A	2	0	12	
3	1000001	P00085442	F	0	10	A	2	0	12	
4	1000002	P00285442	M	6	16	C	4+	0	8	
5	1000003	P00193542	M	2	15	A	3	0	1	
6	1000004	P00184942	M	4	7	B	2	1	1	
7	1000004	P00346142	M	4	7	B	2	1	1	
8	1000004	P0097242	M	4	7	B	2	1	1	
9	1000005	P00274942	M	2	20	A	1	1	8	

3.2.2.2. Chuyển thuộc tính Gender sang số

```
gender_dict = {'F':0, 'M':1}
dataset['Gender'] = dataset['Gender'].apply(lambda line: gender_dict[line])
dataset['Gender'].value_counts()
```

```
1    405380
0    132197
Name: Gender, dtype: int64
```

Kết quả:

- Kiểu dữ liệu của Gender đã chuyển sang int64

- Các giá trị ‘F’ và ‘M’ đã lần lượt chuyển sang ‘0’ và ‘1’

```
dataset.head(10)
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years
0	1000001	P00069042	0	0	10	A	2
1	1000001	P00248942	0	0	10	A	2
2	1000001	P00087842	0	0	10	A	2
3	1000001	P00085442	0	0	10	A	2
4	1000002	P00285442	1	6	16	C	4+
5	1000003	P00193542	1	2	15	A	3
6	1000004	P00184942	1	4	7	B	2
7	1000004	P00346142	1	4	7	B	2
8	1000004	P0097242	1	4	7	B	2
9	1000005	P00274942	1	2	20	A	1

3.2.2.3. Chuyển thuộc tính City_Category sang số

```
city_dict = {'A':0, 'B':1, 'C':2}
dataset['City_Category'] = dataset['City_Category'].apply(lambda line: city_dict[line])
dataset['City_Category'].value_counts()
```

```
1    226493
2    166446
0    144638
Name: City_Category, dtype: int64
```

Kết quả:

- Kiểu dữ liệu của City_Category đã chuyển sang int64
- Các giá trị ‘A’ , ‘B’ và ‘C’ đã lần lượt chuyển sang ‘0’, ‘1’ và ‘2’.

3.2.2.4. Chuyển thuộc tính Stay_In_Current_City_Years sang kiểu số

Loại bỏ dấu ‘+’ và chuyển kiểu dữ liệu sang float

```
dataset['Stay_In_Current_City_Years']=(dataset['Stay_In_Current_City_Years'].str.strip('+').astype('float'))
```

Kết quả:

Đã mất dấu '+' và tất cả giá trị đã chuyển sang kiểu float.

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Produ
0	1000001	P00069042	0	0	10	0	2.0	0	3	
1	1000001	P00248942	0	0	10	0	2.0	0	1	
2	1000001	P00087842	0	0	10	0	2.0	0	12	
3	1000001	P00085442	0	0	10	0	2.0	0	12	
4	1000002	P00285442	1	6	16	2	4.0	0	8	
5	1000003	P00193542	1	2	15	0	3.0	0	1	
6	1000004	P00184942	1	4	7	1	2.0	1	1	
7	1000004	P00346142	1	4	7	1	2.0	1	1	
8	1000004	P0097242	1	4	7	1	2.0	1	1	
9	1000005	P00274942	1	2	20	0	1.0	1	8	

3.3. Thu giảm dữ liệu (data reduction)

3.3.1. Tổng quát kỹ thuật

Một kho dữ liệu lớn có thể chứa lượng dữ liệu lên đến terabytes sẽ làm cho quá trình khai phá dữ liệu chạy rất mất thời gian, do đó nên thu giảm dữ liệu.

Việc thu giảm dữ liệu sẽ thu được một biểu diễn thu gọn, mà nó vẫn sinh ra cùng (hoặc xấp xỉ) các kết quả khai phá như tập dữ liệu ban đầu.

Các chiến lược thu giảm:

- Giảm số chiều (dimensionality reduction), loại bỏ bớt các thuộc tính không (ít) quan trọng.
- Giảm lượng dữ liệu (data/numberosity reduction)

3.3.2. Thực hiện trên dataset

Sau khi phân tích dữ liệu ,ta thấy 2 thuộc tính 'User_ID' và 'Product_ID' không cần thiết nên có thể xóa.

```
dataset.drop(['User_ID','Product_ID'], axis=1, inplace=True)
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 537577 entries, 0 to 537576
Data columns (total 10 columns):
Gender                        537577 non-null int64
Age                           537577 non-null int64
Occupation                    537577 non-null int64
City_Category                 537577 non-null int64
Stay_In_Current_City_Years    537577 non-null float64
Marital_Status                537577 non-null int64
Product_Category_1            537577 non-null int64
Product_Category_2            537577 non-null float64
Product_Category_3            537577 non-null float64
Purchase                      537577 non-null int64
dtypes: float64(3), int64(7)
memory usage: 41.0 MB
```

4. Lưu file dữ liệu đã xử lý

```
dataset.to_csv('../test/input/BlackFriday_modified.csv',index=False)
```

5. Xây dựng model với phương pháp hồi quy- Linear Regression

5.1.. Tổng quan hồi qui

5.1.1. Khái niệm

- Hồi qui là kỹ thuật thống kê cho phép dự đoán các trị (số) liên tục. J.Han et al(2001, 2006).
- Hồi qui (Phân tích hồi quy – regression analysis) là kỹ thuật thống kê cho phép ước lượng các mối liên kết giữa các biến. Wiki(2009)
- Hồi qui (Phân tích hồi quy) là kỹ thuật thống kê trong lĩnh vực phân tích dữ liệu và xây dựng các mô hình từ thực nghiệm, cho phép mô hình hồi qui vừa được khám phá được dùng cho mục đích dự báo (prediction), điều khiển

(control), hay học (learn) cơ chế đã tạo ra dữ liệu. R.D.Snee(1977)v

5.1.2. Mô hình Hồi qui

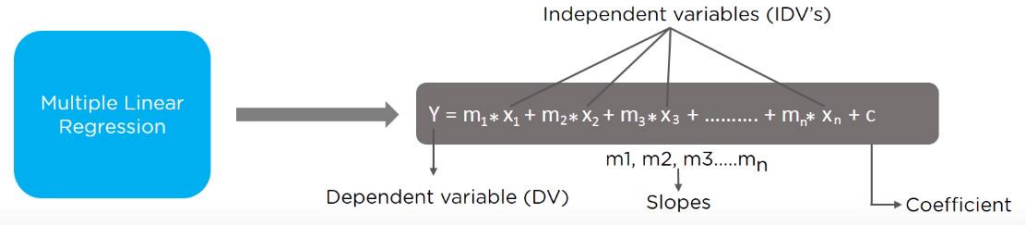
Mô hình mô tả mối liên kết (relationship) giữa một tập các biến dự báo (predictor variables/independent variables) và một hay nhiều đáp ứng (responses/dependent variables).

5.1.3. Phân loại

- Hồi qui tuyến tính (linear) và phi tuyến (nonlinear).
- Hồi qui đơn biến (single) và đa biến (multiple).
- Hồi qui có thông số (parametric), phi thông số (nonparametric), và thông số kết hợp (semiparametric).
- Hồi qui đối xứng (symmetric) và bất đối xứng (asymmetric).

5.2. Xây dựng mô hình

5.2.1. Dạng tổng quát



Trong đó:

- Y : là biến phụ thuộc (Purchase)
- C : là sai số
- m1,m2 ,m3 ,…m_n là hệ số hồi qui
- x1,x2,x3 ,…x_n là các biến dự đoán

5.2.2. Import thư viện

```
import numpy as np
import pandas as pd
import os
print(os.listdir("../test/input"))
import matplotlib.pyplot as plt
import seaborn as sns
```

5.2.3. Load dataset và chọn biến dự đoán và biến độc lập

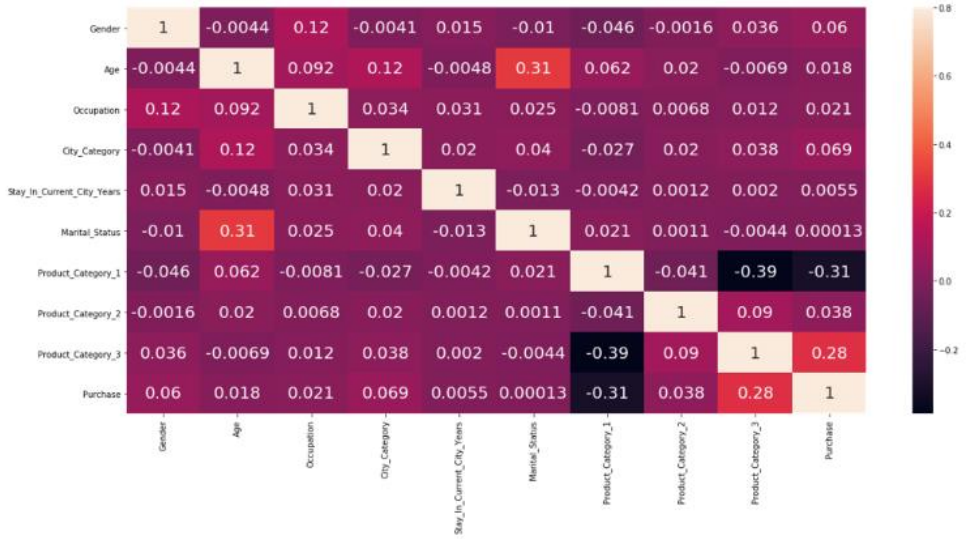
```
dataset=pd.read_csv('../test/input/BlackFriday_modified.csv')

X=dataset[['Gender','Age','Occupation','City_Category','Stay_In_Current_City_Years','Marital_Status','Product_Category_1','Produ
Y=dataset['Purchase']
dataset.head()
```

	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchas
0	0	0	10	0	2.0	0	3	0.0	0.0	837
1	0	0	10	0	2.0	0	1	6.0	14.0	1520
2	0	0	10	0	2.0	0	12	0.0	0.0	142
3	0	0	10	0	2.0	0	12	14.0	0.0	105
4	1	6	16	2	4.0	0	8	0.0	0.0	796

5.2.4. Data visualization

```
corr = dataset.corr()
f, ax = plt.subplots(figsize=(20, 9))
sns.heatmap(corr, vmax=.8,annot_kws={'size': 20}, annot=True);
```



5.2.5. Chia dự liệu thành 2 tập: Train set và Test set

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
```

5.2.6. Sử dụng Multiple Linear Regression lên tập Huấn luyện(Train set)

```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(X_train, Y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

5.2.7. Dự đoán kết quả của tập dữ liệu kiểm nghiệm(Test set)

```
Y_pred = lm.predict(X_test)
Y_pred
```

Kết quả:

```
array([ 7657.14518007,  8672.65100773, 10505.27435169, ...,
       11621.88037493,  9418.50845378,  9307.45046938])
```

5.2.8. Tính sai số và các hệ số hồi qui

5.2.8.1. Sai số (Coefficients)

```
print(lm.coef_)
```

```
[ 483.43171241  108.03049682    5.62812692  333.26455655    7.38571934
  -55.14235338 -318.19331716    8.31925039  149.38230824]
```

5.2.8.2. Các hệ số hồi qui (Intercepts)

```
print(lm.intercept_)
```

```
9373.223899652607
```

5.3. Đánh giá mô hình với các độ đo

5.3.1. R-square

5.3.1.1. Tổng quan về R-Square

R-Square xuất phát từ ý tưởng:Toàn bộ sự biến thiên của biến phụ thuộc được chia làm 2 phần:Phần biến thiên do hồi qui và phần biến thiên do hồi qui.Như trên phần xây dựng mô hình thì chúng ta chia dữ liệu ra làm 2 phần là phần dữ liệu Train(tức là phần áp dụng hồi qui) và phần dữ liệu Test(tức không áp dụng hồi qui).

5.3.1.2. Công thức tính

$$R^2 = 1 - (ESS/TSS)$$

Trong đó:

- Regression Sum of Squares(RSS):Tổng độ lệch bình phương giải thích từ hồi qui
- Residual Sum of Squares(ESS): Tổng các độ lệch bình phương phần dư
- Total Sum of Squares(TSS) : Tổng các độ lệch bình phương toàn bộ.

5.3.1.3. Ý nghĩa

Giá trị R dao động từ 0 đến 1.R-square càng gần 1 thì mô hình đã xây dựng càng phù hợp với tập dữ liệu dùng chạy hồi qui.R-Square càng gần 0 thì mô hình đã xây dựng càng kém phù hợp với tập dữ liệu chạy hồi qui.

5.3.1.4. Đánh giá kết quả mô hình vừa xây dựng

5.3.1.4.1. Tính R-Square

```
from sklearn.metrics import r2_score
r2_score(Y_test,Y_pred)
```

5.3.1.4.2. Kết quả

```
0.13202924305187025
```

5.3.1.4.3. Đánh giá

Kết quả R-Square gần bằng 13,2%.Ta có thể kết luật là mô hình vừa xây dựng kém phù hợp với tập dữ liệu này.

5.3.2. RMSE

5.3.2.1. Tổng quan về RMSE

Root Mean Square Error(RMSE) là độ lệch chuẩn của phần dư (lỗi dự đoán). Phần dư là thước đo khoảng cách từ các điểm dữ liệu đường hồi quy; RMSE là thước đo mức độ lan truyền của những phần dư này. Nói cách khác, nó cho bạn biết mức độ tập trung của dữ liệu xung quanh dòng phù hợp nhất. Lỗi bình phương trung bình thường được sử dụng trong khí hậu học, dự báo và phân tích hồi quy để xác minh kết quả thí nghiệm.

5.3.2.2. Công thức tính

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y)^2}{n}}$$

Trong đó:

- n: Tổng số quan sát
- y^: là vector của n trị dự báo
- y: là là vector các trị quan sát được
- *trung bình* $(\frac{1}{n}\sum_{i=1}^{n})$ *của bình phương các sai số* $((Y_i - \hat{Y}_i)^2)$.

5.3.2.3. Ý nghĩa

Nó cho bạn biết mức độ tập trung của dữ liệu xung quanh đường hồi qui, kết quả càng lớn mức độ biến thiên của dữ liệu càng cao. Kết quả càng nhỏ thì mức độ dự báo càng chính xác.

5.3.2.4. Đánh giá kết quả mô hình vừa xây dựng

5.3.2.4.1. Tính RMSE

```
from sklearn.metrics import mean_squared_error
mean_squared_error(Y_test,Y_pred)
```

5.3.2.4.2. Kết quả

```
21399636.520310994
```

5.3.2.4.3. Đánh giá

Kết quả RMSE rất lớn.Ta có thể kết luật là sự biến thiên của dữ liệu so với đường hồi quy là chênh lệch quá nhiêu, mô hình vừa xây dựng có độ chính xác trong dự đoán rất thấp.

V. Kết luận

1. Ưu điểm

Việc làm khai thác dữ liệu mang đến những kinh nghiệm trong việc dự đoán xu hướng, giá trị, cũng như tìm ra những điểm tương đồng kết suất thành kết quả mà chúng ta mong muốn, xong bên cạnh đó đòi hỏi tính kiên trị, chịu khó, tìm hiểu các mô hình khai thác dữ liệu, kiến thức máy học và sự hiểu biết về chuyên ngành liên quan đến dữ liệu cần phân tích. Black Friday là tập dự liệu phù hợp cho sự học tập cơ bản khai thác dữ liệu.

2. Nhược điểm

Xong bên cạnh những kết quả đạt được là những thiếu xót về kiến thức các mô hình khác nhau, nhằm phân tích đạt được nhiều kết quả khai thác mong muốn.Thời gian tìm hiểu còn ít nên kiến thức tìm hiểu được chỉ ở mức cơ bản.

3. Hướng phát triển đồ án

Do kết quả khi đánh giá mô hình vừa xây dựng không được như mong muốn.Nên nhóm đề ra ột số bước tiếp theo cần làm để cải thiện mô hình:

- Tạo thêm những biến mới để tìm những biến có ảnh hưởng đến biến Purchase.Ví dụ như: User_ID_Count,Product_ID_Count,…
- Sử dụng thêm các phương pháp khác để tìm ra phương pháp phù hợp với dataset.Ví dụ như Ridge Regression,Decision Tre,Classification,Cluster,..

VI. Bảng phân công công việc

	Lựa chọn dữ liệu	Lựa chọn thuật toán,công cụ	Phân tích dữ liệu(tiền xử lý)	Làm sạch dữ liệu(tiền xử lý)	Biến đổi dữ liệu(tiền xử lý)	Xây dựng model	Rút kết kết quả	Chỉnh sửa lần cuối	Báo cáo
Hữu	**X**	**X**	**X**			**X**	**X**	**X**	
Ngọc	**X**	**X**		**X**	**X**				**X**

VII. Bảng đánh giá chéo các thành viên

	Đúng thời gian	**Mức độ hoàn thành**	**Đóng góp tri thức**
Hữu	**90%**	**95%**	**55%**
Ngọc	**90%**	**95%**	**45%**

VIII. Tài liệu tham khảo

- **https://www.youtube.com/watch?v=NUXdtN1W1FE&list=LLX9aIZX9UDH4WkU-klksS3Q&index=2&t=0s**
- **https://viblo.asia/p/linear-regression-hoi-quy-tuyen-tinh-trong-machine-learning-4P856akRlY3**
- **https://www.kaggle.com/mehdidag/black-friday**
- **http://bis.net.vn/forums/t/366.aspx**
- **https://www.youtube.com/channel/UCh9nVJoWXmFb7sLApWGcLPQ**
- **https://www.youtube.com/channel/UCsvqVGtbbyHaMoevxPAq9Fg**