Group 7:

Danh sách hành viên:

- *Lương Trọng Khôi - 20120514*
- *Lê Minh Trí - 20120601*
- *Nguyễn Minh Trí - 20120602*
- *Nguyễn Thanh Tùng - 20120618*
- *Nguyễn Hoàng Vinh - 20120628*

PHÂN CÔNG CÔNG VIỆC			
MSSV	HỌ TÊN	NHIỆM VỤ	ĐÁNH GIÁ
20120514	Lương Trọng Khôi	Phần A. Thu thập dữ liệu	100%
20120601	Lê Minh Trí	Phần B. Khám phá dữ liệu	100%
20120602	Nguyễn Minh Trí	Phần C. Khám phá các mối quan hệ trong dữ liệu	100%
20120618	Nguyễn Thanh Tùng	Phần C. Khám phá các mối quan hệ trong dữ liệu	100%
20120628	Nguyễn Hoàng Vinh	Phần C. Khám phá các mối quan hệ trong dữ liệu	100%

BẢNG TỰ ĐÁNH GIÁ

Tiêu chí	Tự đánh giá
Thu thập và tiền xử lý dữ liệu.	100%
Chọn lựa, giải thích, trực quan các trường và các mối quan hệ giữa chúng.	100%
Rút ra ý nghĩa hợp lý sau mỗi dữ liệu được trực quan.	100%
Xem xét trên nhiều quan hệ, nhiều góc nhìn khác nhau.	100%
Báo cáo trình bày bố cục và định dạng hợp lý, rõ ràng.	100%
Có những phân tích, trực quan hóa bằng những biểu đồ mới lạ và rút ra những thông tin hữu ích. Sử dụng mô hình học máy cơ bản.	100%

A. Thu thập dữ liệu

Ngữ cảnh và động cơ lựa chọn dữ liệu

- Ngữ cảnh: Tập dữ liệu "customer_shopping_data.csv" được sử dụng để phân tích hành vi mua sắm của khách hàng, từ đó có thể đưa ra những quyết định kinh doanh hợp lý.
- Động cơ lựa chọn tìm kiếm dữ liệu: Bằng việc thu thập dữ liệu liên quan đến thông tin khách hàng và hành vi mua sắm của họ, các doanh nghiệp có thể hiểu rõ hơn về nhu cầu và sở thích của khách hàng, từ đó tối ưu hóa chiến lược marketing và đưa ra những sản phẩm phù hợp.

Chủ đề của dữ liệu và nguồn thu thập

- Chủ đề của dữ liệu: Tập dữ liệu này liên quan đến hành vi mua sắm của khách hàng, bao gồm các biến như độ tuổi, giới tính, thu nhập, số lần mua hàng, tổng chi phí mua hàng, v.v.
- Nguồn thu thập dữ liệu: Dữ liệu đã được thu thập từ cửa hàng thời trang.

License và phương pháp thu thập

- Không có thông tin về license của tập dữ liệu này.
- Phương pháp thu thập dữ liệu của tập dữ liệu "customer_shopping_data.csv" không được cung cấp. Tuy nhiên, với các cửa hàng thời trang hiện đại, thường sử dụng các phần mềm CRM hoặc POS để quản lý thông tin khách hàng và hành vi mua sắm của họ. Các phần mềm này có thể ghi lại các thông tin như độ tuổi, giới tính, số lần mua hàng, tổng chi phí mua hàng, v.v. khi khách hàng mua sắm tại cửa hàng.

B. Khám phá dữ liệu và tiền xử lý dữ liệu

- Phân tích ý nghĩa của mỗi dòng và mỗi cột trong bảng dữ liệu.
- Xác định kiểu dữ liệu của các cột và kiểm tra tính phù hợp để tiền xử lý dữ liệu.
- Khám phá phân bố giá trị của các cột và xử lý dữ liệu nếu (cụ thể là chuẩn hóa, xử lý giá trị khuyết để dữ liệu trở nên chuẩn mực và dễ sử dụng).

Import thư viện

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Đọc dữ liệu

```
url =
'https://raw.githubusercontent.com/nmtrihcmus/DV_Lab01/main/customer_s
hopping_data.csv'
df = pd.read_csv(url)
# df = pd.read_csv('./data/customer_shopping_data.csv')
df
```

```
      invoice_no customer_id  gender  age         category  quantity
price  \
0        I138884     C241288  Female   28         Clothing         5
1500.40
1        I317333     C111565    Male   21            Shoes         3
1800.51
2        I127801     C266599    Male   20         Clothing         1
300.08
3        I173702     C988172  Female   66            Shoes         5
3000.85
4        I337046     C189076  Female   53            Books         4
60.60
...          ...         ...     ...  ...              ...       ...
...
99452    I219422     C441542  Female   45         Souvenir         5
58.65
99453    I325143     C569580    Male   27  Food & Beverage         2
10.46
99454    I824010     C103292    Male   63  Food & Beverage         2
10.46
99455    I702964     C800631    Male   56       Technology         4
4200.00
99456    I232867     C273973  Female   36         Souvenir         3
35.19

      payment_method invoice_date     shopping_mall
0        Credit Card     5/8/2022           Kanyon
1         Debit Card   12/12/2021    Forum Istanbul
2               Cash    9/11/2021        Metrocity
3        Credit Card   16/05/2021      Metropol AVM
4               Cash   24/10/2021           Kanyon
...              ...          ...              ...
99452    Credit Card   21/09/2022           Kanyon
99453           Cash   22/09/2021    Forum Istanbul
99454     Debit Card   28/03/2021        Metrocity
99455           Cash   16/03/2021      Istinye Park
99456    Credit Card   15/10/2022  Mall of Istanbul

[99457 rows x 10 columns]
```

```
df.shape
```

```
(99457, 10)
```

```
df.size
```

```
994570
```

```
df.isnull().sum()
```

```
invoice_no        0
customer_id       0
gender            0
age               0
category          0
quantity          0
price             0
payment_method    0
invoice_date      0
shopping_mall     0
dtype: int64
```

```
df.isna().sum()
```

```
invoice_no        0
customer_id       0
gender            0
age               0
category          0
quantity          0
price             0
payment_method    0
invoice_date      0
shopping_mall     0
dtype: int64
```

```
df.duplicated().value_counts()
```

```
False    99457
dtype: int64
```

```
df.describe()
```

```
                age      quantity         price
count  99457.000000  99457.000000  99457.000000
mean      43.427089      3.003429    689.256321
std       14.990054      1.413025    941.184567
min       18.000000      1.000000      5.230000
25%       30.000000      2.000000     45.450000
50%       43.000000      3.000000    203.300000
75%       56.000000      4.000000   1200.320000
max       69.000000      5.000000   5250.000000
```

```
df.columns
```

```
Index(['invoice_no', 'customer_id', 'gender', 'age', 'category',
'quantity',
       'price', 'payment_method', 'invoice_date', 'shopping_mall'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99457 entries, 0 to 99456
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   invoice_no      99457 non-null  object
 1   customer_id     99457 non-null  object
 2   gender          99457 non-null  object
 3   age             99457 non-null  int64
 4   category        99457 non-null  object
 5   quantity        99457 non-null  int64
 6   price           99457 non-null  float64
 7   payment_method  99457 non-null  object
 8   invoice_date    99457 non-null  object
 9   shopping_mall   99457 non-null  object
dtypes: float64(1), int64(2), object(7)
memory usage: 7.6+ MB

df.nunique()

invoice_no        99457
customer_id       99457
gender                2
age                  52
category              8
quantity              5
price                40
payment_method        3
invoice_date        797
shopping_mall        10
dtype: int64
```

Tiền xử lý

```
# Chuyển đổi thành dạng datetime
df['invoice_date'] = df['invoice_date'].astype('datetime64');

# Sắp xếp giảm dần theo datetime
# df = df.sort_values(by='invoice_date', ascending=False)

# Xóa dữ liệu từ tháng 3/2023
df = df.loc[df['invoice_date'] < '2023-03-01'];

# Lọc theo năm 2023
df_2023 = df[df['invoice_date'].dt.year == 2023];
df_2023

      invoice_no customer_id  gender  age         category  quantity
price  \
20       I183746     C220180    Male   23         Clothing         1
300.08
51       I202367     C317478  Female   41            Books         3
```

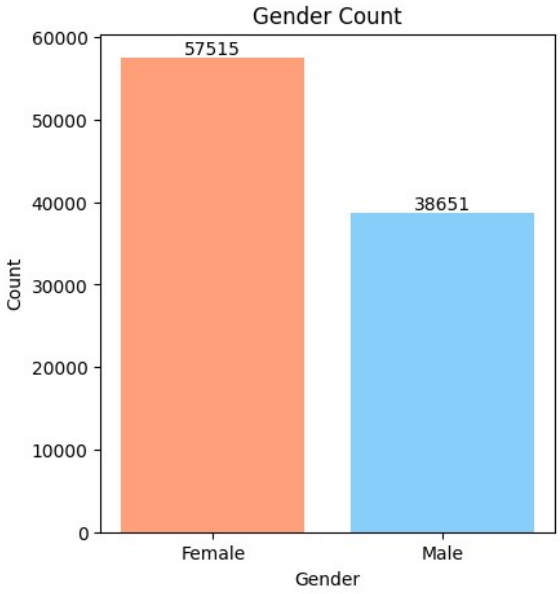
```
45.45
89       I195567     C992677    Male   65         Clothing         4
1200.32
102      I985478     C324683    Male   55         Clothing         4
1200.32
117      I134452     C112750  Female   32         Clothing         1
300.08
...          ...         ...     ...  ...              ...       ...
...
99400    I289616     C149476  Female   55            Shoes         1
600.17
99429    I208840     C219131  Female   58             Toys         1
35.84
99441    I203187     C235554    Male   38  Food & Beverage         4
20.92
99449    I134399     C953724    Male   65         Clothing         1
300.08
99450    I170504     C226974  Female   28            Books         1
15.15

      payment_method invoice_date      shopping_mall  total_price
age_group  \
20       Credit Card   2023-02-15  Emaar Square Mall       300.08
18-24
51             Cash   2023-02-24       Istinye Park       136.35
35-44
89        Debit Card   2023-01-22       Metropol AVM      4801.28
65+
102      Credit Card   2023-01-24             Kanyon      4801.28
65+
117      Credit Card   2023-01-15     Forum Istanbul       300.08
25-34
...              ...          ...                ...          ...
...
99400     Debit Card   2023-01-30             Kanyon       600.17
65+
99429    Credit Card   2023-02-18       Istinye Park        35.84
55-64
99441            Cash   2023-02-03       Zorlu Center        83.68
35-44
99449            Cash   2023-01-01             Kanyon       300.08
65+
99450            Cash   2023-02-28       Zorlu Center        15.15
25-34

       year  month
20     2023      2
51     2023      2
89     2023      1
102    2023      1
```

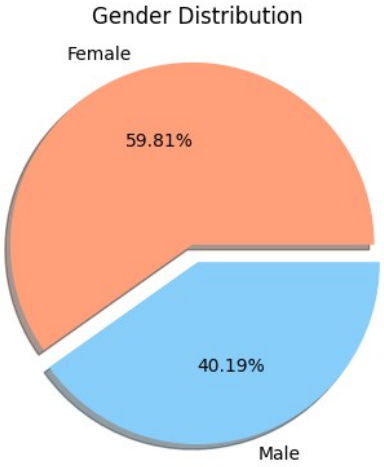
```
117    2023      1
...     ...    ...
99400  2023      1
99429  2023      2
99441  2023      2
99449  2023      1
99450  2023      2

[5233 rows x 14 columns]
```

C. Khám phá mối quan hệ trong dữ liệu

Nhóm giới tính nào mua sắm nhiều hơn?

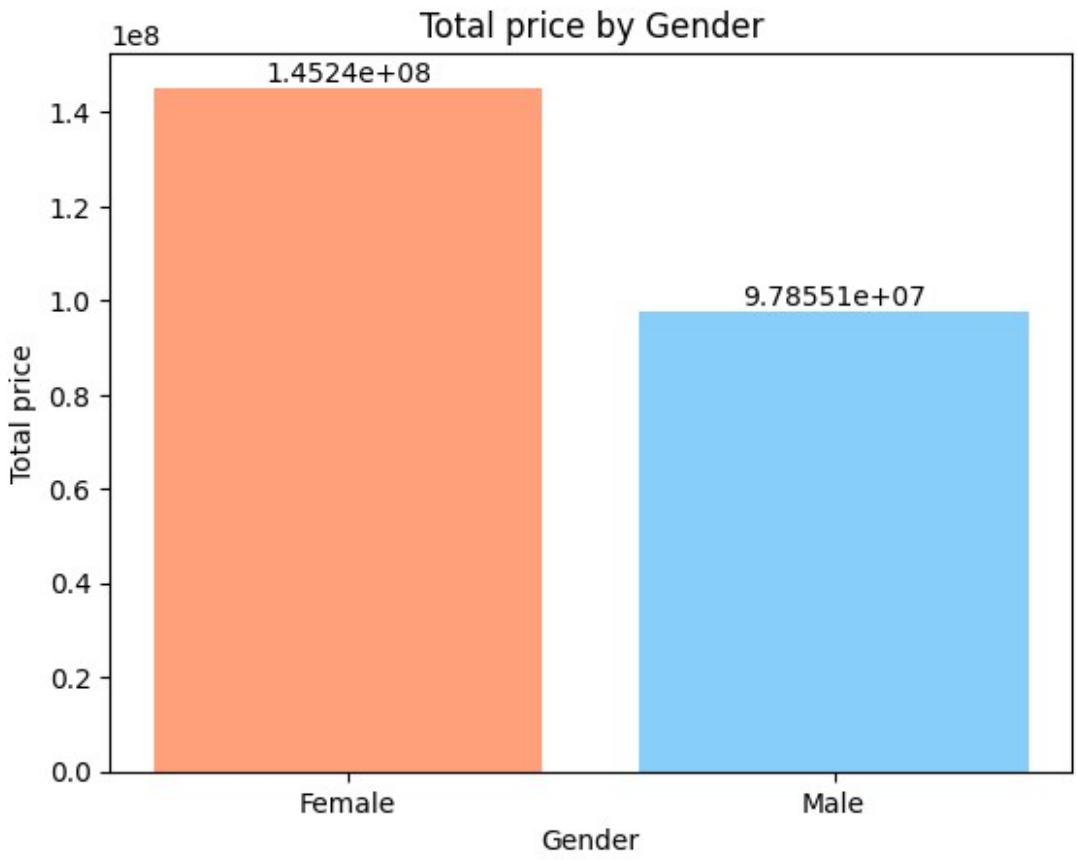
```
#Tính sô´ lượng khách hàng cu̇a từng giới tính
gender_counts =df['gender'].value_counts()

# Tạo khung hình với hai phâ`n
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

# Vẽ biê̇u đô` cột trong phâ`n bên trái
ax1.bar(gender_counts.index, gender_counts.values, color=['#FFA07A',
'#87CEFA'])
for i in ax1.containers:
    ax1.bar_label(i)
ax1.set_title("Gender Count")
ax1.set_xlabel("Gender")
ax1.set_ylabel("Count")

# Vẽ biê̇u đô` tròn trong phâ`n bên phȧi
colors = ['#FFA07A', '#87CEFA']
explode = (0.1, 0)
ax2.pie(gender_counts, labels=gender_counts.index, autopct='%1.2f%%',
colors=colors, explode=explode, shadow=True)
ax2.set_title("Gender Distribution")

# Hiê̇n thị hai biê̇u đô` cạnh nhau
plt.show()
```





- Nhóm sử dụng hai loại biểu đồ khác nhau để trực quan hóa phân bố giới tính của khách hàng trong tập dữ liệu.
- Trong phần bên trái, nhóm sử dụng biểu đồ cột (bar chart) để hiển thị số lượng khách hàng của từng giới tính. Đây là loại biểu đồ phù hợp để so sánh giá trị định lượng (số lượng khách hàng) giữa các nhóm (nam và nữ). Chúng ta có thể thấy rõ ràng rằng số lượng nữ khách hàng cao hơn số lượng nam khách hàng.
- Trong phần bên phải, nhóm sử dụng biểu đồ tròn (pie chart) để trực quan hóa tỷ lệ phần trăm của từng giới tính. Đây là loại biểu đồ phù hợp để trình bày phân phối tần suất của các nhóm và phần trăm mỗi nhóm so với tổng thể. Tỷ lệ phần trăm nữ khách hàng cao hơn tỷ lệ phần trăm nam khách hàng.
- Từ hai biểu đồ, nhóm có thể kết luận rằng phân phối giới tính trong tập dữ liệu không đồng đều, với tỷ lệ nữ khách hàng cao hơn nam khách hàng. Từ đó cũng cho thấy nhu cầu mua sắm của nữ nhiều hơn nam.

```
# Tính tổng giá tiền mua sắm theo giới tính
df['total_price'] = df['quantity']*df['price']
total_price_by_gender = df.groupby('gender')['total_price'].sum()

# Vẽ biểu đồ cột tổng giá tiền mua sắm theo giới tính
fig, ax = plt.subplots()
ax.bar(total_price_by_gender.index, total_price_by_gender.values,
color=['#FFA07A','#87CEFA'])
for i in ax.containers:
    ax.bar_label(i)
# Đặt tên cho trục tung và trục hoành
plt.title('Total price by Gender')
plt.xlabel('Gender')
plt.ylabel('Total price')
# Hiển thị biểu đồ
plt.show()
```

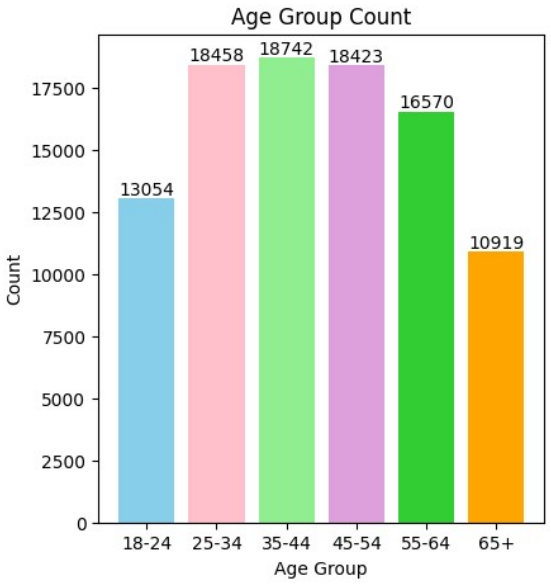


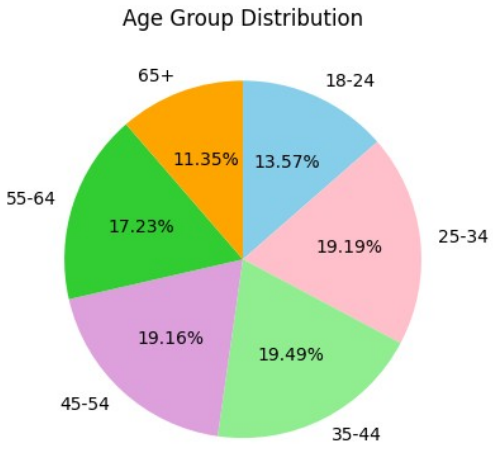
- Biểu đồ cột trực quan này thể hiện tổng giá tiền mua sắm theo giới tính. Từ biểu đồ, nhóm có thể nhận thấy rằng giá trị giá tiền mua sắm của nữ khách hàng cao hơn nam khách hàng. Điều này có thể cho thấy phái nữ có xu hướng tiêu dùng nhiều hơn phái nam hoặc các sản phẩm dành cho nữ có giá trị cao hơn so với các sản phẩm dành cho nam. Điều này có thể giúp cho các doanh nghiệp tập trung vào các sản phẩm hoặc dịch vụ phù hợp để thu hút và phục vụ tốt hơn các khách hàng nữ.

Phân phối theo nhóm tuổi

```
# Hàm phân chia tuổi theo từng nhóm tuổi
def set_age_category(df):
    if df <= 24 :
        df= '18-24'
    elif df >24 and df <=34:
        df= '25-34'
    elif df >34 and df <=44:
        df= '35-44'
    elif df >44 and df <=54:
        df= '45-54'
    elif df >55 and df <=64:
        df= '55-64'
    else:
```

```
        df= '65+'
    return df

df['age_group']=df['age'].apply(set_age_category)
age_group_counts= df['age_group'].value_counts().sort_index()

# Tạo khung hình với hai phần
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))
colors = ['skyblue', 'pink', 'lightgreen',
'plum','limegreen','orange']

# Vẽ biểu đồ cột trong phần bên trái
ax1.bar(age_group_counts.index, age_group_counts.values, color=colors)
for i in ax1.containers:
    ax1.bar_label(i)
ax1.set_title("Age Group Count")
ax1.set_xlabel("Age Group")
ax1.set_ylabel("Count")

# Vẽ biểu đồ tròn trong phần bên phải
# colors = ['skyblue', 'pink', 'lightgreen',
'plum','limegreen','orange']
ax2.pie(age_group_counts, labels=age_group_counts.index,
autopct='%1.2f%%', colors=colors, startangle=-90)
ax2.set_title("Age Group Distribution")
ax2.invert_yaxis()

# Hiển thị hai biểu đồ cạnh nhau
plt.show()
```



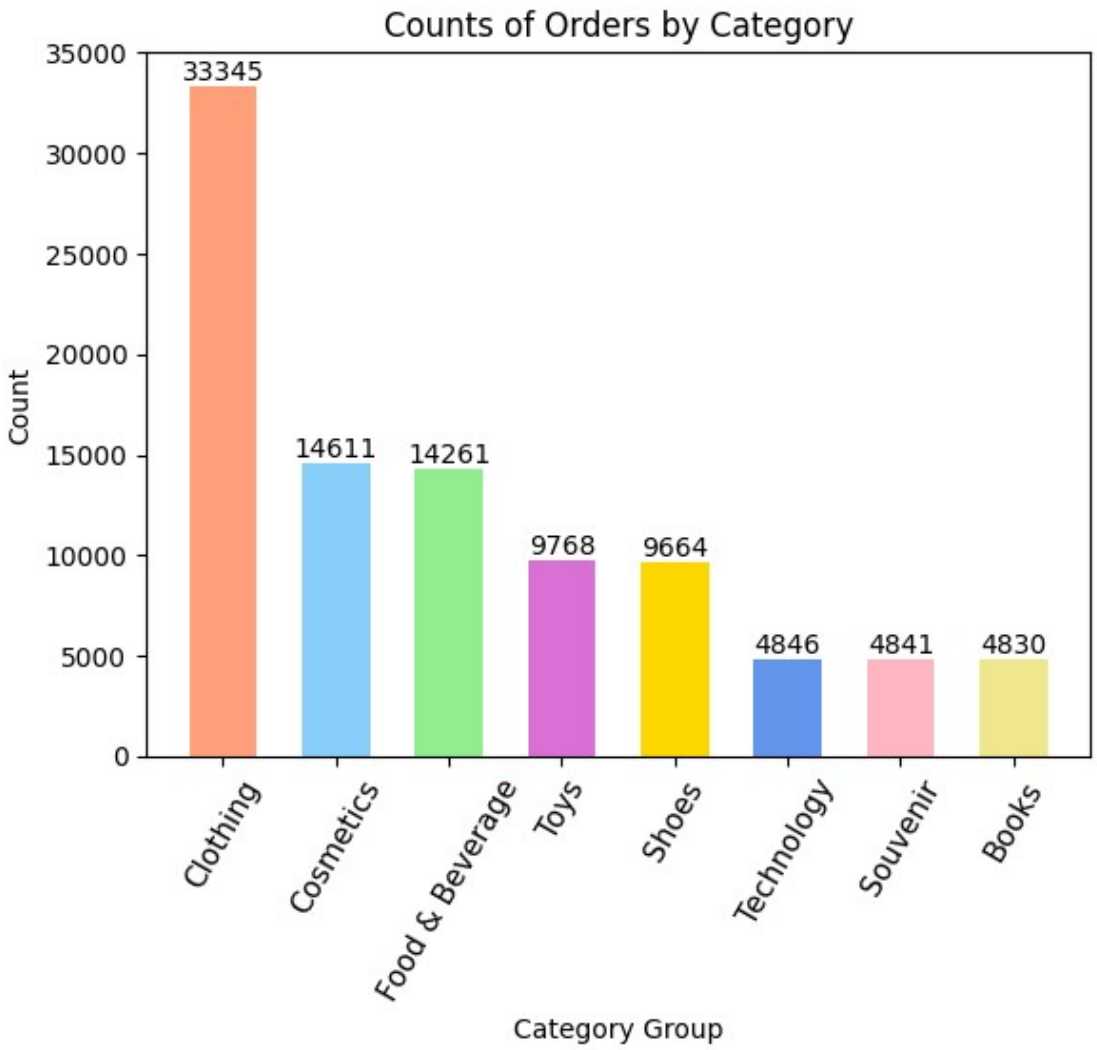


- Hai biểu đồ trên cho thấy phân bố và số lượng khách hàng theo nhóm tuổi.

- Biểu đồ cột bên trái cho thấy số lượng khách hàng trong mỗi nhóm tuổi, với độ tuổi từ 35 đến 44 chiếm số lượng lớn nhất, đến sau đó là độ tuổi từ 25 đến 34 và từ 45 đến 54. Biểu đồ tròn bên phải cho thấy tỷ lệ phần trăm khách hàng trong mỗi nhóm tuổi, với độ tuổi từ 35 đến 44 chiếm số lượng lớn nhất, đến sau đó là độ tuổi từ 25 đến 34 và từ 45 đến 54.
- Từ hai biểu đồ, ta có thể kết luận rằng đối tượng khách hàng chủ yếu của cửa hàng là những người trong độ tuổi từ 25 đến 54, đặc biệt là trong độ tuổi từ 35 đến 44. Cho thấy nhu cầu mua sắm của nhóm tuổi tử trưởng thành tới trung niên chiếm tỉ lệ cao.

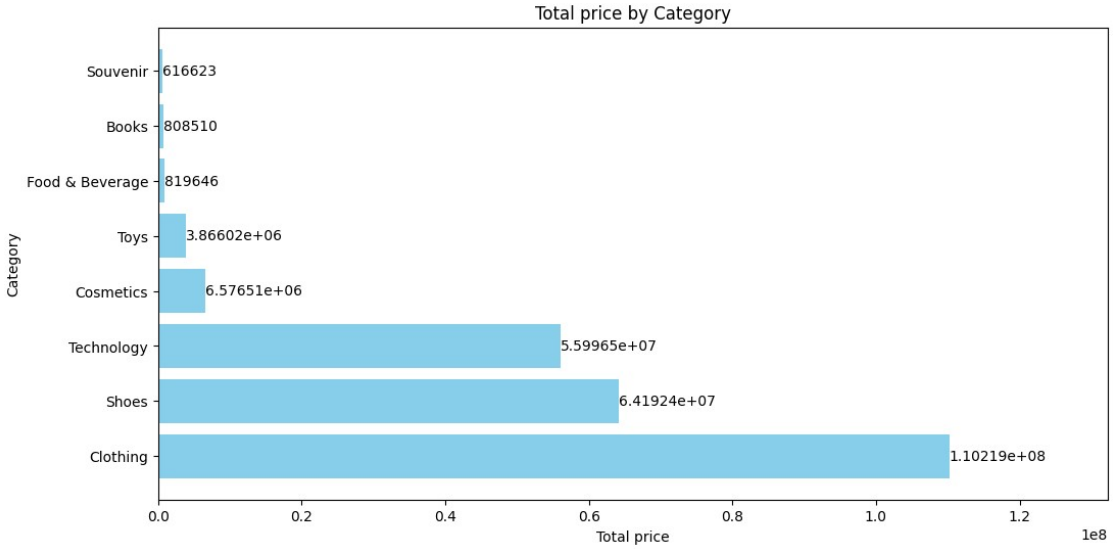
Khách hàng hay mua sắm loại mặt hàng nào nhất?

```
category_counts=
df['category'].value_counts().sort_values(ascending=False)
#Vẽ biểu đồ
fig, ax = plt.subplots()
colors =
['#FFA07A','#87CEFA','#90EE90','#DA70D6','#FFD700','#6495ED','#FFB6C1'
,'#F0E68C']
ax.bar(category_counts.index, category_counts.values, color=colors,
width=0.6)
plt.xticks(rotation=60, fontsize=11)
for i in ax.containers:
    ax.bar_label(i)
plt.title('Counts of Orders by Category')
plt.xlabel("Category Group")
plt.ylabel("Count")
plt.show()
```



- Biểu đồ cột trên trực quan và cho thấy phân bố số lượng đơn hàng theo từng nhóm sản phẩm. Màu sắc được sử dụng cho mỗi cột giúp tạo ra một hiệu ứng tương phản và giúp dễ dàng phân biệt giữa các nhóm sản phẩm.
- Dựa trên biểu đồ, ta có thể thấy nhóm sản phẩm "Clothing" có số lượng đơn hàng cao nhất *(hơn gấp đôi mặt hàng cao thứ hai)* , tiếp theo là nhóm sản phẩm "Cosmetics" và "Food & Beverage". Trong khi đó, các nhóm sản phẩm khác có số lượng đơn hàng thấp hơn đáng kể so với các nhóm trên.
- Từ kết quả trên, ta có thể suy ra một số kết luận, ví dụ như:
  - Doanh số của cửa hàng tập trung chủ yếu vào các nhóm sản phẩm "Clothing", "Cosmetics" và "Food & Beverage" là các mặt hàng liên quan đến sinh hoạt.
  - Nên tập trung quảng cáo và phát triển sản phẩm cho các nhóm sản phẩm trên để tăng doanh số.
  - Các nhóm sản phẩm khác có thể được đưa vào chương trình khuyến mãi để thu hút khách hàng mua sắm.

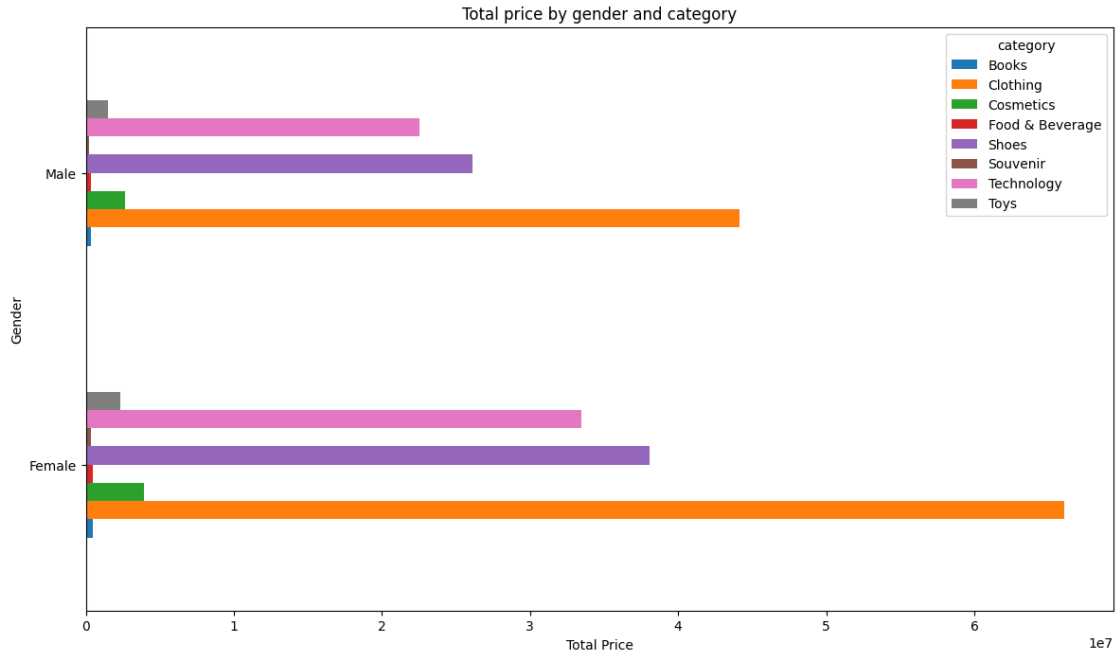
```
# Tính tổng giá tiền mua sắm theo từng loại sản phẩm
total_price_by_category = df.groupby('category')
['total_price'].sum().sort_values(ascending=False)
# Vẽ biểu đồ cột tổng giá tiền mua sắm theo loại sản phẩm
fig, ax = plt.subplots(figsize = (12, 6))

ax.barh(total_price_by_category.index, total_price_by_category.values,
color='skyblue')

for i in ax.containers:
    ax.bar_label(i)
# Điều chỉnh kích thước label trên trục x
ax.set_xlim([0, max( total_price_by_category.values) * 1.2])
# Đặt tên cho trục tung và trục hoành
plt.ylabel('Category')
plt.xlabel('Total price')
plt.title('Total price by Category')
# Hiển thị biểu đồ
plt.show()
```



- Biểu đồ cột ngang (horizontal bar chart) được chọn để hiển thị tổng giá tiền mua sắm theo từng loại sản phẩm vì nó giúp cho việc so sánh giá trị giữa các loại sản phẩm dễ dàng hơn so với biểu đồ cột dọc.
- Từ kết quả trực quan, ta thấy được loại sản phẩm "Clothing" có tổng giá trị mua sắm cao nhất, theo sau là "Shoes" và "Technology". Trong khi đó, loại sản phẩm "Books" và "Souvenir" có tổng giá trị mua sắm thấp nhất.
- Từ kết quả này, ta có thể rút ra kết luận rằng:
  - Doanh thu có được từ mặt hàng quần áo rất cao, sau đó là từ mặt hàng giày. Cho thấy nhu cầu mua các mặt hàng thời trang là rất cao.
  - Doanh thu từ công nghệ đứng thứ 3, mặc dù đây có vẻ là mặt hàng có giá đắt nhưng cũng cho thấy nhu cầu mua sắm về mặt hàng này cũng khá cao.

- Các mặt hàng khác có doanh thu khá thấp, đặc biệt là sách và đồ lưu niệm, cho thấy nhu cầu đọc sách là rất thấp, và nhu cầu mua đồ lưu niệm cũng rất thấp.

```
# Nhóm dữ liệu theo từng giới tính và từng nhóm sản phẩm
grouped = df.groupby(['gender', 'category'])
['total_price'].sum().unstack()

# Vẽ biểu đồ
fig, ax = plt.subplots(figsize=(14, 8))
grouped.plot(kind='barh', ax=ax)

# Cấu hình thêm cho biểu đồ
ax.set_ylabel('Gender')
ax.set_xlabel('Total Price')
ax.set_title('Total price by gender and category')
plt.show()
```

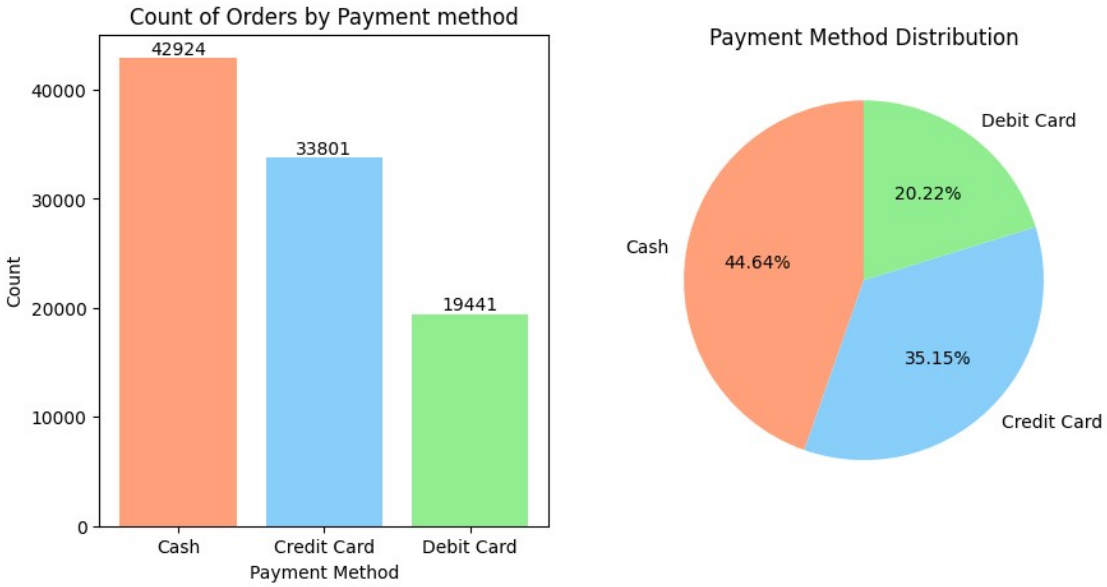


- Biểu đồ vẽ tổng giá trị các sản phẩm đã mua theo từng nhóm sản phẩm và từng giới tính khác nhau. Ta có thể thấy rằng giá trị trung bình của sản phẩm mà phái nữ mua đa phần cao hơn so với phái nam trong hầu hết các nhóm sản phẩm, đặc biệt là ở các nhóm sản phẩm quần áo, giày dép và công nghệ. Điều này là thông tin hữu ích trong việc định hướng kinh doanh và marketing cho các nhà bán lẻ.

Phương thức thanh toán nào được khách hàng sử dụng nhiều nhất?

```
# Tính toán số lượng khách hàng theo từng phương thức thanh toán
payment_method_count = df['payment_method'].value_counts()

# Vẽ biểu đồ
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10,5))
```

```
colors=['#FFA07A', '#87CEFA', '#90EE90']

# Vẽ biểu đồ cột
ax1.bar(payment_method_count.index, payment_method_count.values,
color=colors)

ax1.set_title('Count of Orders by Payment method')
ax1.set_xlabel('Payment Method')
ax1.set_ylabel('Count')

for i in ax1.containers:
    ax1.bar_label(i)

# vẽ biểu đồ tròn
ax2.pie(payment_method_count, labels= payment_method_count.index,
colors=colors, autopct='%1.2f%%',  startangle=90)

ax2.set_title('Payment Method Distribution')

plt.show()
```

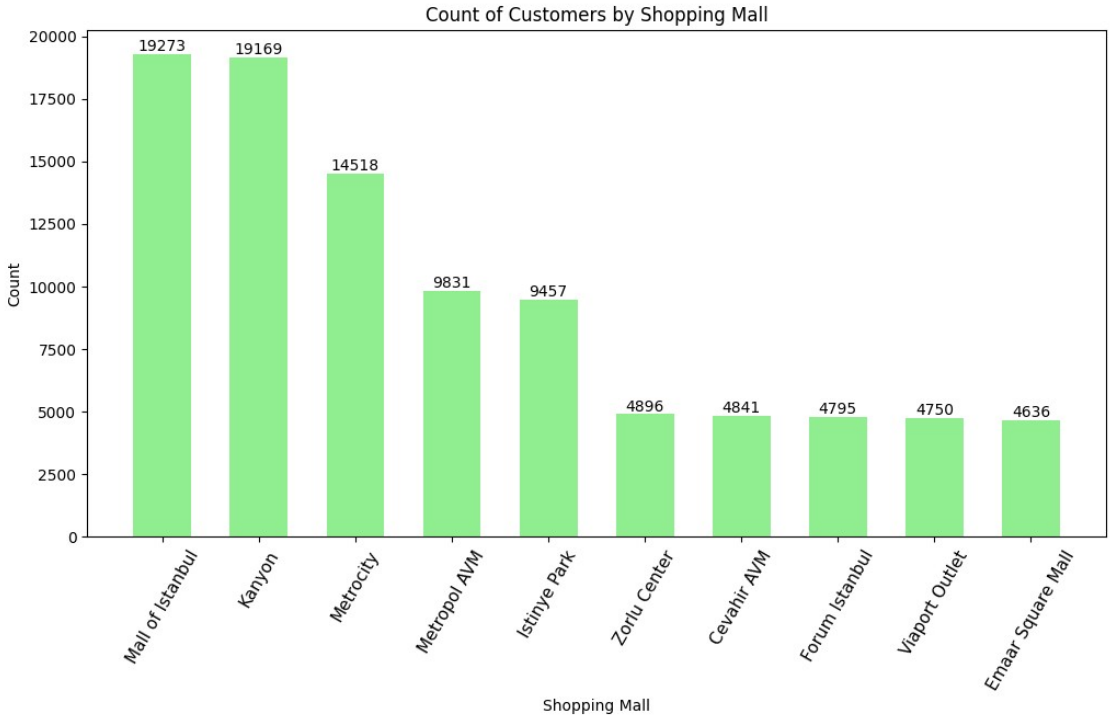


- Biểu đồ trực quan cho thấy phương thức thanh toán phổ biến nhất là *Cash (chiếm khoảng 45% tổng số phương thức thanh toán)*, theo sau đó là Credit Card và cuối cùng là Debit Cash. Biểu đồ cột và biểu đồ tròn đều cho thấy thông tin tương tự nhau, tuy nhiên biểu đồ cột cho ta cái nhìn tổng quát hơn về số lượng của từng phương thức thanh toán. Trong khi đó, biểu đồ tròn thích hợp hơn để thể hiện tỷ lệ phần trăm của từng phương thức thanh toán trong tổng số.

Trung tâm mua sắm nào được ghé thăm nhiều nhất?

```
# Gom nhóm sô´ lượng khách hàng theo từng shopping mall và să´p xê´p
giảm dần
```

```
shopping_mall_count =
df['shopping_mall'].value_counts().sort_values(ascending= False)
# Vẽ biểu đồ
fig, ax = plt.subplots(figsize=(12,6))
ax.bar(shopping_mall_count.index , shopping_mall_count.values,
color='lightgreen', width=0.6)
plt.xticks(rotation=60, fontsize=11)
for i in ax.containers:
    ax.bar_label(i)
plt.xlabel('Shopping Mall')
plt.ylabel('Count')
plt.title('Count of Customers by Shopping Mall')
plt.show()
```



- Biểu đồ cột cho thấy số lượng khách hàng của các trung tâm thương mại, với trục x là tên trung tâm thương mại và trục y là số lượng khách hàng. Biểu đồ cho thấy rằng *trung tâm thương mại Istanbul có số lượng khách hàng cao nhất, tiếp theo là trung tâm thương mại Kanyon, Metrocity, Metropol AVM và Istinye Park. Các trung tâm thương mại còn lại đều có số lượng khách hàng tương đối gần nhau*. Biểu đồ này có thể giúp cho các nhà quản lý các trung tâm thương mại hiểu hơn về lượng khách hàng và đưa ra các quyết định về quảng cáo, khuyến mãi, và phát triển các dịch vụ để tăng khách hàng của mình.

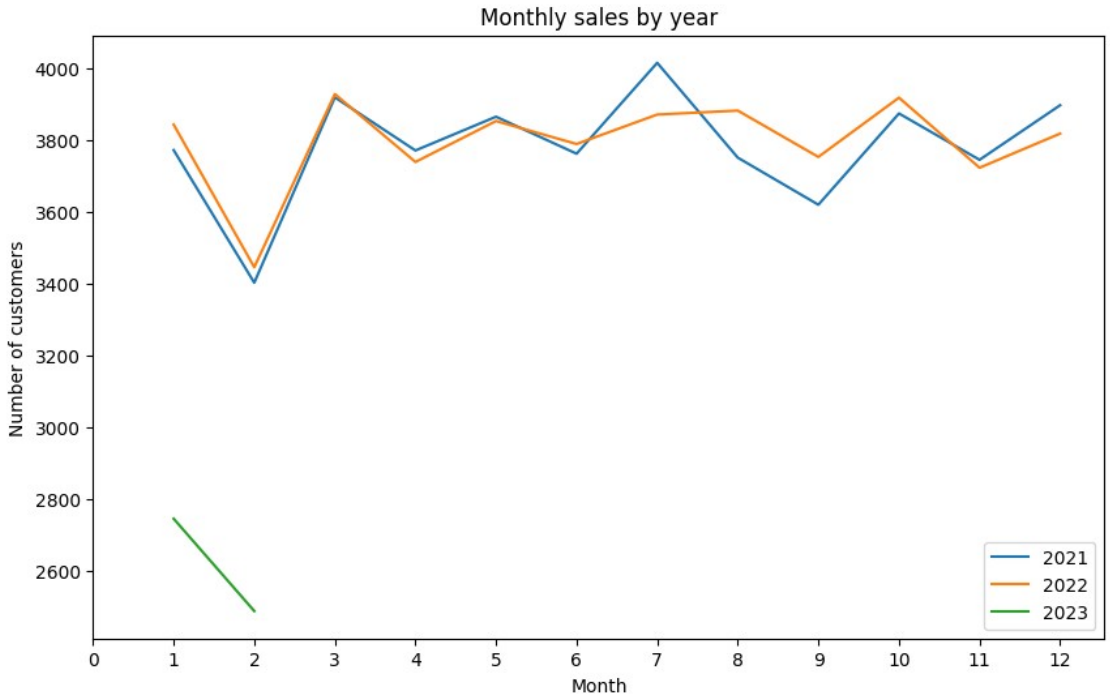
```
# Tính tổng giá trị theo từng shopping mall và sắp xếp giảm dần
shopping_mall_revenue = df.groupby('shopping_mall')
['total_price'].sum().sort_values(ascending=False)
# Vẽ biểu đồ
fig, ax = plt.subplots(figsize=(14,6))
```

```
ax.barh(shopping_mall_revenue.index , shopping_mall_revenue.values,
color='salmon')
for i in ax.containers:
    ax.bar_label(i)
ax.set_xlim([0, max( shopping_mall_revenue.values) * 1.15])
plt.ylabel('Shopping Mall')
plt.xlabel('Revenue')
plt.title('Revenue by Shopping Mall')
plt.show()
```

- Biểu đồ này cho thấy doanh thu của từng trung tâm mua sắm trong tập dữ liệu. Các trung tâm mua sắm được ghé thăm nhiều đều đóng góp một số lượng doanh thu lớn hơn so với các trung tâm mua sắm có lượt ghé thăm ít hơn. Kết quả này có thể giúp cho các nhà quản lý trung tâm mua sắm hiểu rõ hơn về sự phân bố doanh thu của các trung tâm mua sắm và đưa ra các quyết định phù hợp để tăng doanh thu cho các trung tâm mua sắm ít được ghé thăm.

```
# Tạo cột year lưu năm cuả từng đơn hàng
df["year"] = df["invoice_date"].dt.year;

# Tạo cột month lưu tháng cuả từng đơn hàng
df["month"] = df["invoice_date"].dt.month;


# Gom nhóm dữ liệu theo năm và tháng, tính tổng số người mua sắm
hàng tháng
monthly_sales = df.groupby(["year", "month"])["customer_id"].nunique()

# Chuyển đổi dữ liệu từ series sang dataframe
monthly_sales_df = monthly_sales.reset_index(name="num_customers")

# Tạo biểu đồ đường số lượng người mua sắm hàng tháng theo từng
năm
fig, ax = plt.subplots(figsize=(10, 6))
```

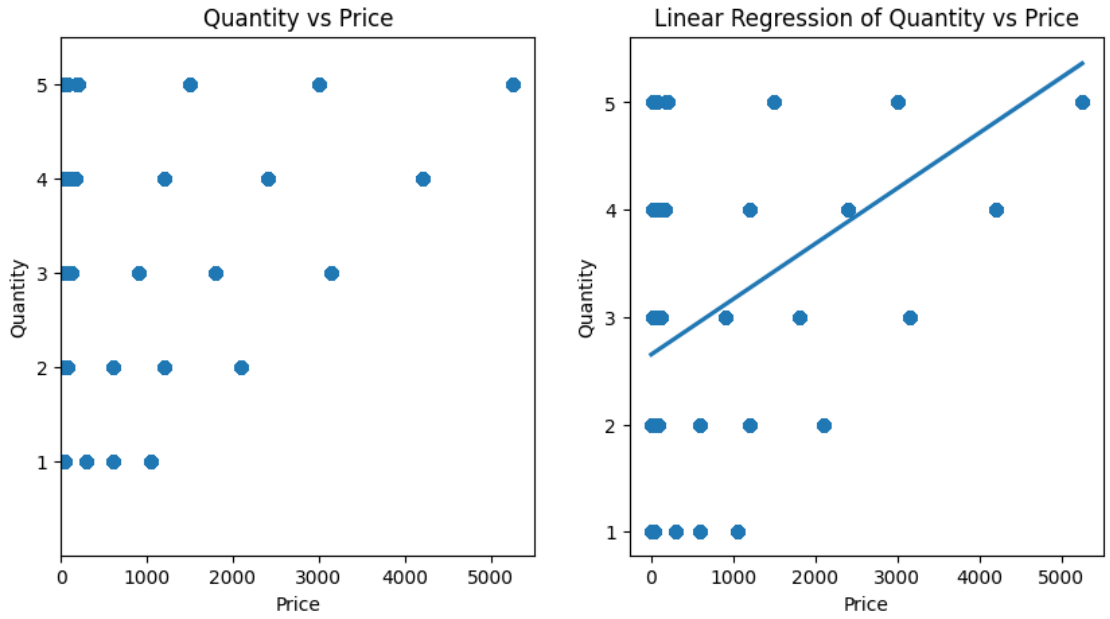
```
for year in monthly_sales_df["year"].unique():
    year_data = monthly_sales_df[monthly_sales_df["year"] == year]
    ax.plot(year_data["month"], year_data["num_customers"],
label=str(year))
plt.xticks(np.arange(0, 13, 1))
# Đặt tên cho các trục và tiêu đề cho biểu đồ
ax.set_xlabel("Month")
ax.set_ylabel("Number of customers")
ax.set_title("Monthly sales by year")

# Đặt legend cho biểu đồ
ax.legend()

# Hiển thị biểu đồ
plt.show()
```



- Biểu đồ cho thấy số lượng người mua sắm hàng tháng trong các năm khác nhau. Ta có thể quan sát thấy rằng số lượng người mua sắm thường tăng và giảm liên tục, giảm vào đầu năm (từ tháng 1 đến tháng 2) và sau đó nhìn chung là tăng. Ngoài ra, ta có thể thấy rằng số lượng người mua sắm giảm dần qua các năm (giảm mạnh ở 2023).

```
# Vẽ biểu đồ Scatter plot
fig, axs = plt.subplots(ncols=2, figsize=(10, 5))

# Scatter plot giữa giá và số lượng
axs[0].scatter(df['price'], df['quantity'])
axs[0].set_yticks(range(int(min(df['quantity'])),
int(max(df['quantity']))+1))
```

```
axs[0].set_ylim((0, max(df['quantity']) + 0.5))
axs[0].set_xlim(left=0)
axs[0].set_ylabel('Quantity')
axs[0].set_xlabel('Price')
axs[0].set_title('Quantity vs Price')

# Biểu đồ hồi quy tuyến tính giữa giá và số lượng
sns.regplot(x='price', y='quantity', data=df, ax=axs[1])
axs[1].set_ylabel('Quantity')
axs[1].set_xlabel('Price')
axs[1].set_title('Linear Regression of Quantity vs Price')

# Hiển thị biểu đồ
plt.show()
```



- Scatter plot và biểu đồ hồi quy tuyến tính được dùng để trực quan hóa quan hệ giữa hai biến liên tục trong tập dữ liệu. Trong trường hợp này, giá cả (price) và số lượng sản phẩm (quantity) là hai biến liên tục. Scatter plot cho phép quan sát mối quan hệ giữa hai biến liên tục và phát hiện ra sự phân bố của dữ liệu. Biểu đồ hồi quy tuyến tính cho thấy mối quan hệ giữa giá cả và số lượng sản phẩm bằng một đường thẳng hồi quy tuyến tính. Kết hợp hai biểu đồ này giúp ta có cái nhìn tổng quan hơn về quan hệ giữa giá cả và số lượng sản phẩm trong tập dữ liệu.
- Bằng cách vẽ Scatter plot giữa giá và số lượng, ta có thể thấy được mối quan hệ giữa giá và số lượng sản phẩm. Từ biểu đồ, có thể thấy rằng có một số sản phẩm với giá rất cao, số lượng sản phẩm bán được vẫn rất cao. Nhưng nhìn chung, số lượng sản phẩm bán được nhiều nhất đều ở mức giá trung bình và thấp.
- Biểu đồ hồi quy tuyến tính giữa giá và số lượng cho thấy có một mối quan hệ tương đối tuyến tính giữa giá và số lượng sản phẩm, nghĩa là khi giá tăng thì số lượng sản phẩm bán ra cũng tăng và ngược lại.

- Từ kết quả trực quan, có thể suy ra rằng giá cả của sản phẩm có ảnh hưởng rất lớn đến số lượng sản phẩm được bán ra. Cần phải chú ý đến việc đặt giá cả phù hợp để tối đa hóa doanh số bán hàng.

Tài liệu tham khảo

https://www.kaggle.com/code/ilyai332/customer-shopping

https://www.kaggle.com/code/drfrank/customer-shopping-eda

https://www.kaggle.com/code/mostafaabdelbadie/customer-shopping-dataset-retail-sales-data-eda

https://www.kaggle.com/code/chloe912/customer-shopping-analysis-eda

https://www.kaggle.com/code/mdforiduzzamanzihad/customer-shopping-retail-sales-analysis