**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC KINH TẾ TP HỒ CHÍ MINH**

**TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ**



**ĐỒ ÁN MÔN HỌC: BIỂU DIỄN TRỰC QUAN DỮ LIỆU**

**ĐỀ TÀI:**

**BIỂU DIỄN TRỰC QUAN DỮ LIỆU, KIỂM ĐỊNH VÀ THỰC HIỆN PCA VỚI BỘ DỮ LIỆU DATA CUSTOMERS**

**Học phần**: INF50908201

**Nhóm Sinh Viên 1**:

1. PHAN TRẦN SƠN BẢO
2. TRƯƠNG THẾ BẢO
3. BÙI THÀNH CÔNG
4. TRẦN GIA ĐỊNH
5. ĐỖ XUÂN ĐỨC

**Chuyên Ngành**: KHOA HỌC DỮ LIỆU

**Khóa**: K46

**Giảng Viên**: TS. Nguyễn An Tế

**TP. Hồ Chí Minh, Ngày 04 tháng 12 năm 2022**

[PHÂN CÔNG CÔNG VIỆC 2](#_Toc122222787)

[CHƯƠNG 1. GIỚI THIỆU VỀ DATASET 3](#_Toc122222788)

[1.1 Tổng quát về dataset: 3](#_Toc122222789)

[1.2 Các thuộc tính của dataset: 3](#_Toc122222790)

[CHƯƠNG 2. TIỀN XỬ LÍ DỮ LIỆU 4](#_Toc122222791)

[2.1 Quy trình tiền xử lí dữ liệu. 4](#_Toc122222792)

[CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU VÀ BIỂU DIỄN TRỰC QUAN DỮ LIỆU 6](#_Toc122222793)

[3.1 Phân tích dữ liệu và biểu diễn trực quan kết quả phân tích. 6](#_Toc122222794)

[3.2 Phân tích theo mô hình RFM và biểu diễn trực quan kết quả phân tích 14](#_Toc122222795)

[CHƯƠNG 4. ÁP DỤNG CÁC THUẬT TOÁN MÁY HỌC VÀ BIỂU DIỄN TRỰC QUAN 25](#_Toc122222796)

[4.1 K-means 25](#_Toc122222797)

[4.1.1 Giới thiệu về Kmeans. 25](#_Toc122222798)

[4.1.2 Áp dụng thuật toán và biễu diên trực quan. 25](#_Toc122222799)

[4.2 HAC 29](#_Toc122222800)

[4.2.1 Giới thiệu về HAC: 29](#_Toc122222801)

[4.2.2 Áp dụng thuật toán và biểu diễn trực quan dữ liệu. 30](#_Toc122222802)

[CHƯƠNG 5: KIỂM ĐỊNH VÀ GIẢM CHIỀU DỮ LIỆU 37](#_Toc122222803)

[5.1. Kiểm định. 37](#_Toc122222804)

[5.1.1 Kiểm định Chi-Square. 37](#_Toc122222805)

[5.1.2 Kiểm định one-way ANNOVA. 38](#_Toc122222806)

[5.2 Giảm chiều dữ liệu. 39](#_Toc122222807)

[5.2.1 Giới thiệu và phương pháp giảm chiều dữ liệu PCA. 39](#_Toc122222808)

[5.2.1 Áp dụng phương pháp với dữ liệu sau khi áp dụng mô hình RFM (với TARGET là Customer Type). 40](#_Toc122222809)

# PHÂN CÔNG CÔNG VIỆC

* BDTQDL: Biểu diễn trực quan dữ liệu.
* LTPTDL: Lập trình phân tích dữ liệu.

|  |  |  |
| --- | --- | --- |
| Họ và tên | Công việc | Tỷ lệ đóng góp |
| Phan Trần Sơn Bảo | - BDTQDL: data visualization, PCA.  - LTPTDL: Tiền xử lí, phân tích dữ liệu, data visualization, RFM score, HAC. | 100% |
| Trương Thế Bảo | - BDTQDL: kiểm định chi square.  - LTPTDL: Phân tích dữ liệu. | 90% |
| Bùi Thành Công | - BDTQDL: kiểm định Anova, data visualization.  - LTPTDL: Phân tích RFM, mô hình Kmeans. | 100% |
| Trần Gia Định | - BDTQDL: kiểm định chi square.  - LTPTDL: Áp dụng thuật toán HAC. | 90% |
| Đỗ Xuân Đức | - BDTQDL: kiểm định Anova.  (không hợp tác làm bài)  - LTPTDL: Áp dụng thuật toán HAC. | 50% |

# CHƯƠNG 1. GIỚI THIỆU VỀ DATASET

## 1.1 Tổng quát về dataset:

Sự tăng trưởng của các siêu thị ở hầu hết các thành phố đông dân cư ngày càng tăng và sự cạnh tranh thị trường cũng cao. Bộ dữ liệu là một trong những doanh số lịch sử của công ty siêu thị đã ghi nhận tại 3 chi nhánh khác nhau trong dữ liệu 3 tháng. Các phương pháp phân tích dữ liệu dự đoán rất dễ áp dụng với tập dữ liệu này.

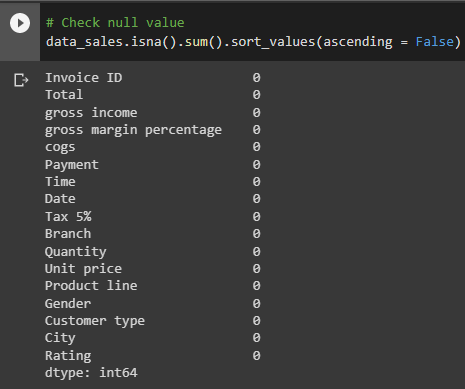
## 1.2 Các thuộc tính của dataset:

1. Invoice ID: Phiếu bán hàng do máy tính tạo hóa đơn mã số.
2. Branch: Chi nhánh siêu trung tâm (3 chi nhánh có sẵn do A, B và C xác định).
3. City: Vị trí của các siêu trung tâm.
4. Customer type: Loại khách hàng, được Hội viên ghi nhận cho khách hàng sử dụng thẻ hội viên và Bình thường đối với khách hàng không có thẻ hội viên.
5. Gender: Giới tính loại khách hàng.
6. Product line: Nhóm phân loại mặt hàng chung - Phụ kiện điện tử, Phụ kiện thời trang, Thực phẩm và đồ uống, Sức khỏe và sắc đẹp, Nhà cửa và lối sống, Thể thao và du lịch.
7. Unit price: Giá của từng sản phẩm tính bằng $.
8. Quantity: Số lượng sản phẩm mà khách hàng mua .
9. Tax: 5% phí thuế cho khách hàng mua .
10. Total: Tổng giá bao gồm cả thuế khách hàng chịu khi mua sản phẩm.
11. Date: Ngày mua (Hồ sơ có sẵn từ tháng 2019 năm 2019 đến tháng 2019 năm XNUMX) .
12. Time: Thời gian mua hàng (10 giờ sáng đến 9 giờ tối) .
13. Payment: hình thức thanh toán được khách hàng sử dụng để mua hàng (có sẵn 3 phương thức - Cash, Credit và Ewallet) .
14. COGS: Giá vốn hàng bán .
15. Gross margin percentage: Tỷ lệ phần trăm lợi nhuận gộp .
16. Gross Income: thu nhập gộp.
17. Rating: khách hàng chấm điểm dựa trên trải nghiệm mua sắm tổng thể của họ (Trên thang điểm từ 1 đến 10).

# CHƯƠNG 2. TIỀN XỬ LÍ DỮ LIỆU

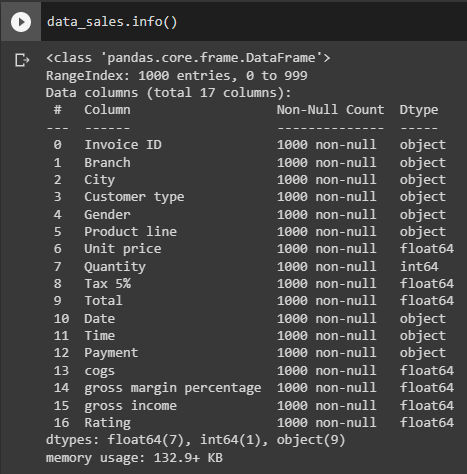
## 2.1 Quy trình tiền xử lí dữ liệu.

- Kiểm tra bộ dữ liệu có missing value không.



Vì bộ dữ liệu được chọn không có missing value nên ta không cần xử lí missing value, nếu có missing value thì ta cần chọn ra phương pháp tối ưu nhất cho bộ dữ liệu để xử lí missing value.

- Tiếp theo, ta kiểm tra kiểu dữ liệu của các columns có đúng không.



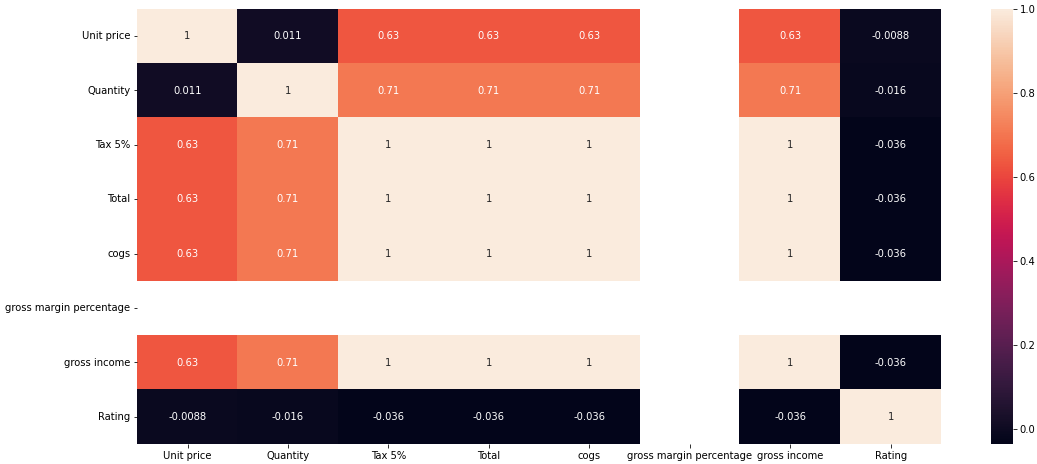
Ở đây, ta thấy dữ liệu của cột Date bị sai (đang là object) nên ta cần chuyển kiểu dữ liệu của cột Date về kiểu datetime.



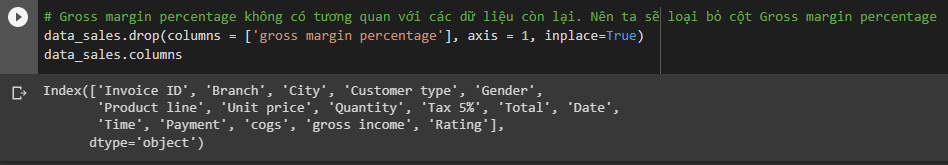
# CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU VÀ BIỂU DIỄN TRỰC QUAN DỮ LIỆU

## 3.1 Phân tích dữ liệu và biểu diễn trực quan kết quả phân tích.

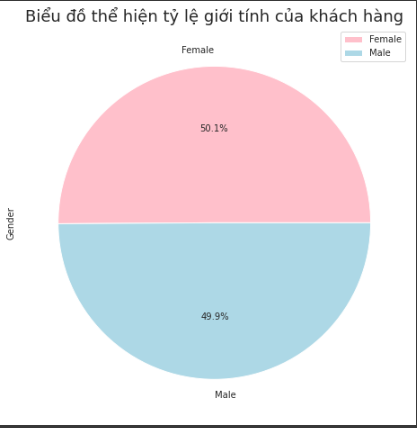
- Ta xem độ tương quan của các columns trong bộ dữ liệu



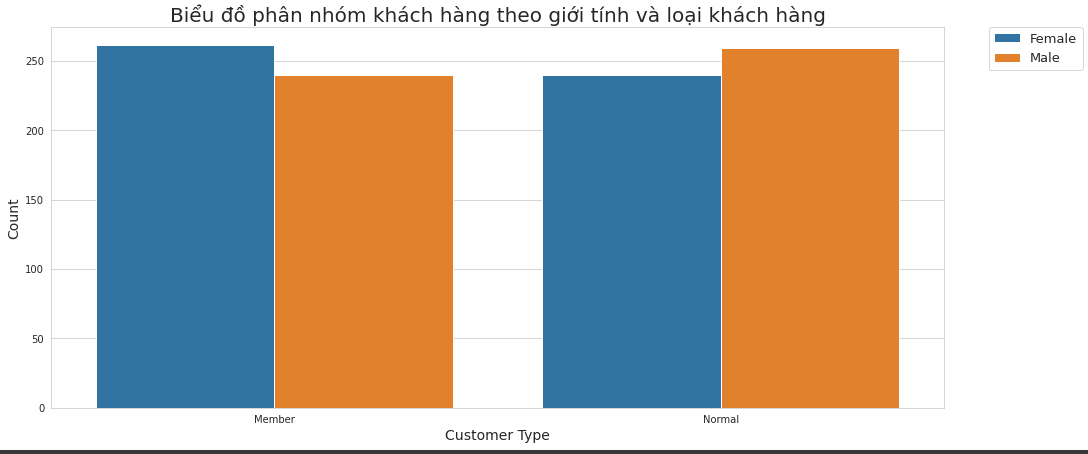
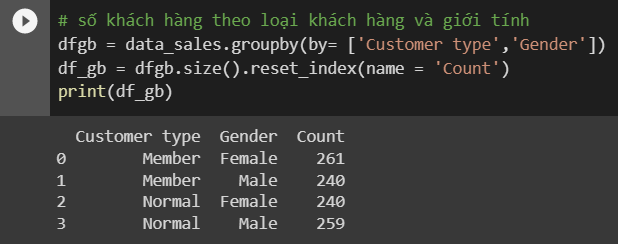
Ta thấy cột gross margin percentage không tương quan với các column còn lại nên ta có thể bỏ cột này đi.



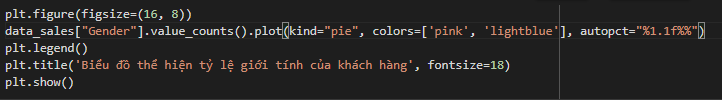
- - Biểu đồ thể hiện tỷ lệ giới tính của khách hàng.



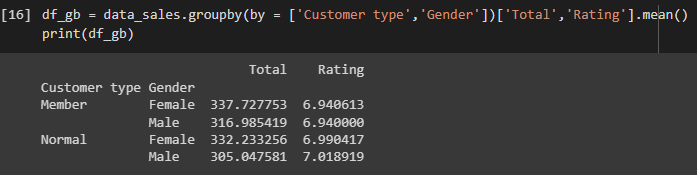
- Sau đó, ta tiến hành phân nhóm khách hàng theo loại khách hàng và giới tính.



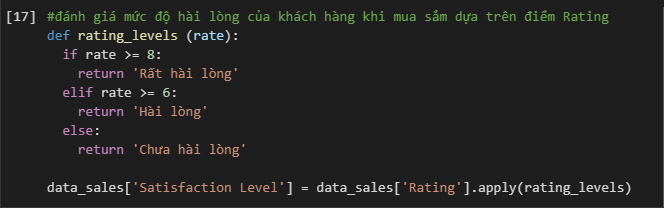
-> Ta thấy được khách hàng thành viên chủ yếu là nữ còn khách hàng thông thường là nam.



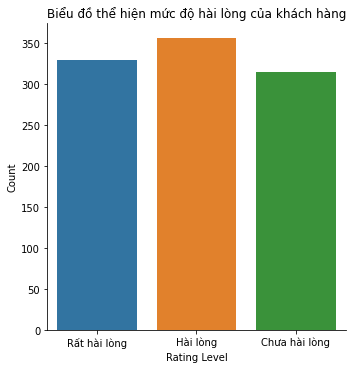
- Ta tính tổng doanh thu và đánh giá của từng nhóm khách hàng theo từng loại khách hàng và giới tính.



- Dựa vào cột Rating ta có thể đánh giá được mức độ hài lòng của khách hàng khi mua sắm ở 3 chi nhánh. Ta cho Rating nhỏ hơn 6 thì khách hàng vẫn còn chưa hài lòng khi mua sắm của siêu thị ở 3 chi nhánh; từ 6 đến 7.9 thì khách hàng hài lòng khi mua sắm ở siêu thị và từ 8 đổ lên thì khách hàng có trải nghiệm rất tuyệt với khi mua sắm.

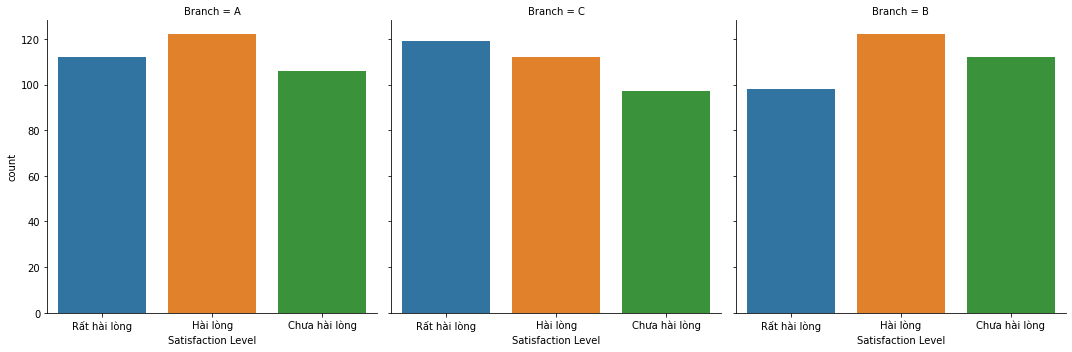


Ta xem biểu đồ thể hiện mức độ hài lòng của khách hàng

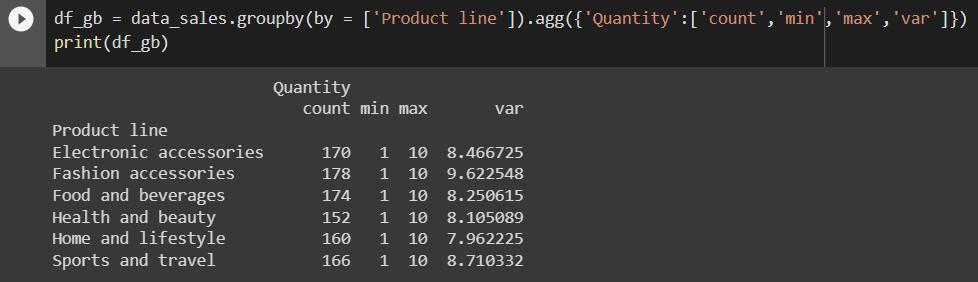


-> Các khách hàng thuộc mức độ chưa hài lòng chiếm tỷ lệ khá nhiều (gần bằng 1/3 tổng khách hàng trong 3 tháng).

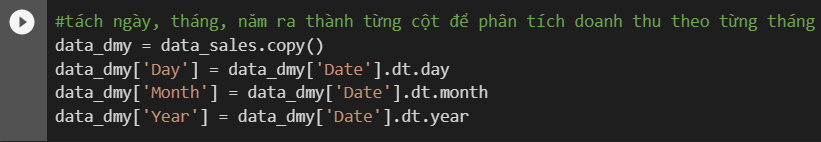
Biểu đồ thể hiện mức độ hài lòng theo từng chi nhánh



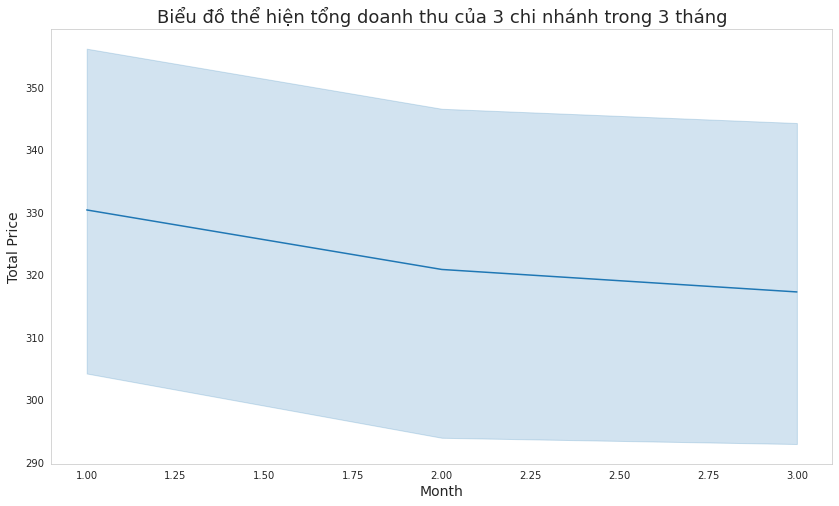
- Ta tìm số lượng sản phẩm được bán, min, max, var của các dòng sản phẩm



- Ta tách ngày, tháng, năm ra thành từng cột để phân tích doanh thu theo từng tháng

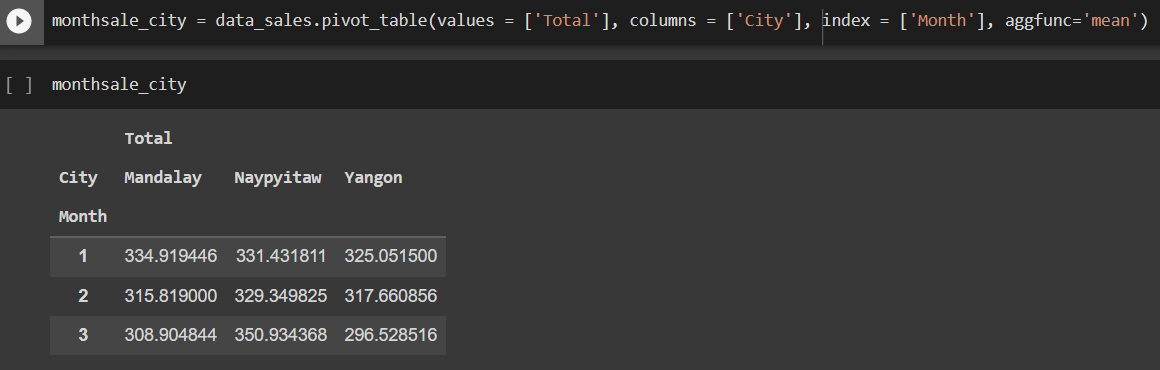


Sau đó ta vẽ biểu đồ thể hiện tổng doanh thu của các chi nhánh trong 3 tháng

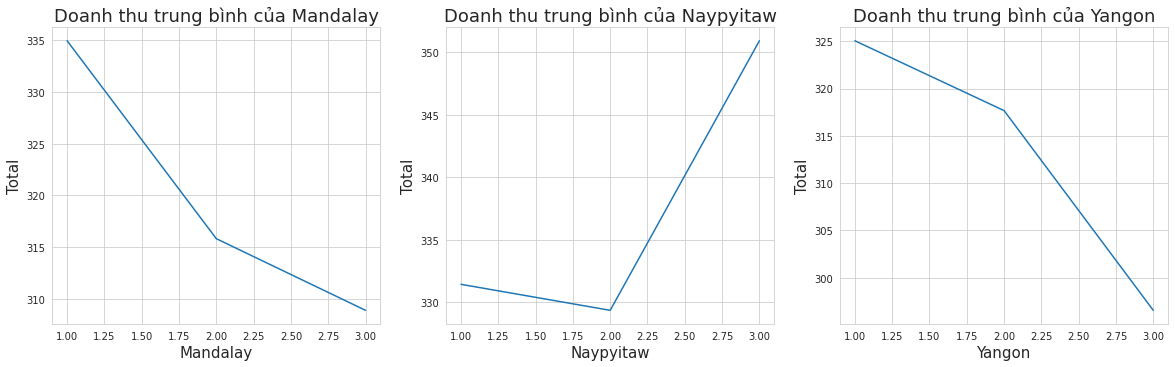


-> Từ tháng 2 thì doanh thu đã giảm 1 lượng nhỏ so với tháng 1.

Tiếp theo, ta tính doanh thu trung bình của 3 chi nhánh trong 3 tháng

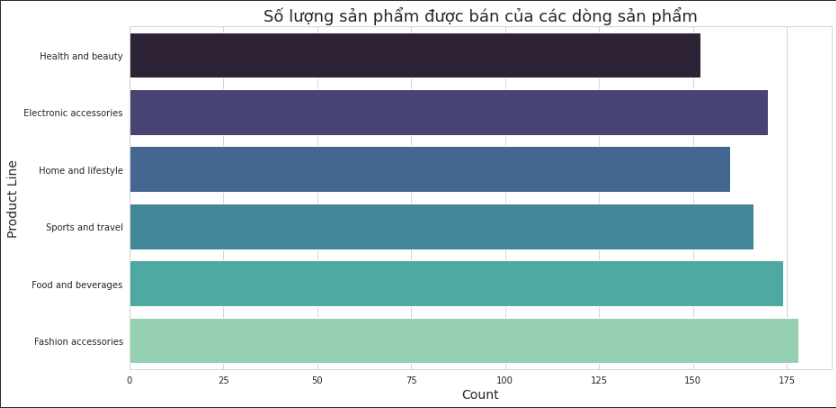


Biểu đồ thể hiện doanh thu trung bình của từng chi nhánh trong 3 tháng



-> Dựa vào biểu đồ, ở Mandalay và Yangon thì doanh thu trung bình liên tục giảm trong ba tháng (nên ta cần tìm ra giải pháp để khắc phục và gia tăng doanh thu.). Tuy nhiên ở Naypyitaw thì doanh thu trung bình có giảm nhẹ vào tháng 2 nhưng tăng lên đáng kể vào tháng 3.

- Ta xem biểu đồ thể hiện số lượng sản phẩm được bán của các dòng sản phẩm



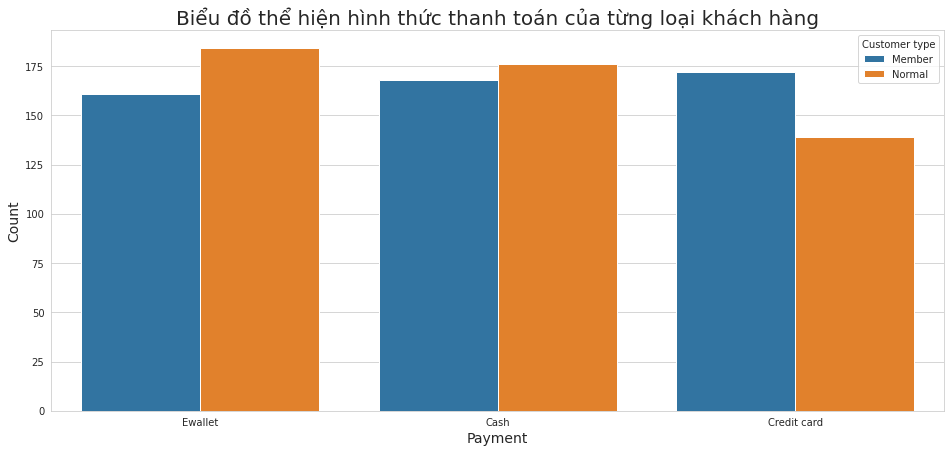
-> Các dòng sản phẩm đều được khách hàng lựa chọn mua khá nhiều và số lượng các dòng sản phẩm được khách hàng mua giữa các dòng sản phẩm cũng không chênh nhau quá nhiều. Nổi bật trong đó là dòng sản phẩm Electronic accessories và Fashion accessories.

- Biểu đồ thể hiện hình thức thanh toán ở mỗi chi nhánh

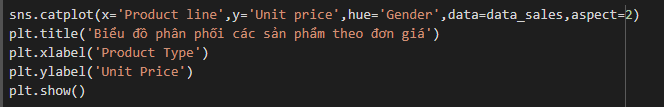


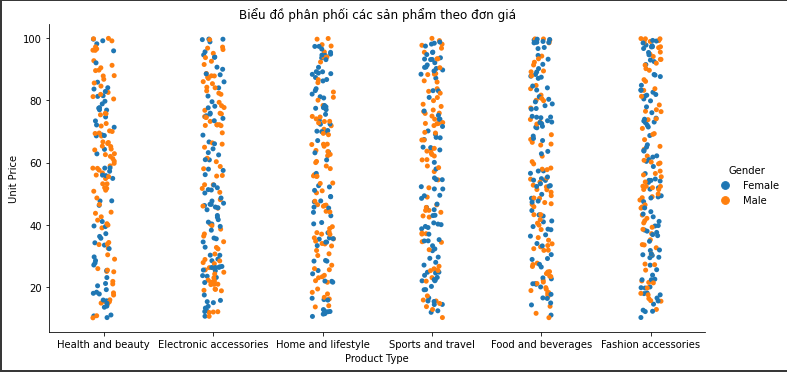
-> 3 hình thức thanh toán ở 3 chi nhánh đều không chênh lệch quá nhiều nhưng ở chi nhánh A (hay ở Yangon) thì khách hàng ưa chuộng hình thức thanh toán bằng Ewallet (ví điện tử) hơn 2 hình thức thanh toán còn lại. Tuy nhiên ở chi nhánh B (hay ở Mandalay) thì khách hàng lại thích thanh toán bằng credit card (thẻ tín dụng) hơn, và cuối cùng ở chi nhánh C (hay ở Naypyitaw) thì khách hàng lại thì thích dùng cash (tiền mặt) để thanh toán khi đi mua sắm.

- Biểu đồ thể hiện hình thức thanh toán của từng loại khách hàng.

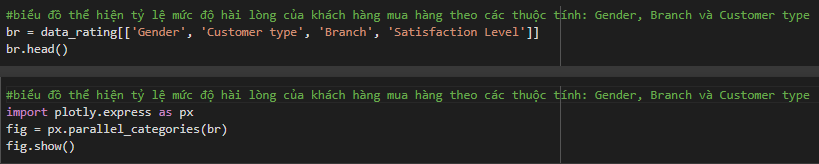
-> Các khách hàng Member (thành viên) lựa chọn thanh toán bằng hình thức Credit card (thẻ tín dụng) hơn 2 thành 2 thức thanh toán còn lại, trong khi khi hàng Normal (thường) lựa chọn hình thức thanh toán bằng Ewallet (ví điện tử) hoặc Cash (tiền mặt) để thanh toán.

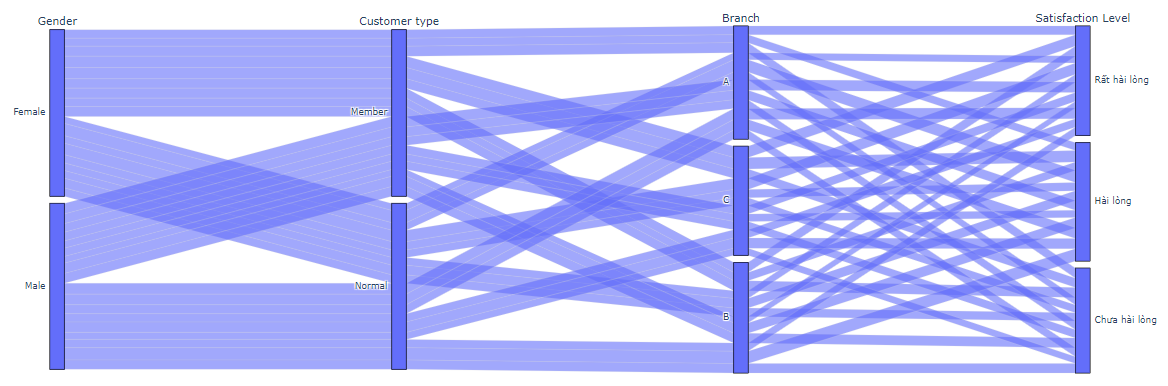
- Biểu đồ thể hiện phân phối các dòng sản phẩm (Product Line) theo đơn giá (unit price)





- Biểu đồ thể hiện tỷ lệ mức độ hài lòng của khách hàng theo các thuộc tính: Gender, Branch và Customer type.





## 3.2 Phân tích theo mô hình RFM và biểu diễn trực quan kết quả phân tích

- Do ở đây dữ liệu bài chưa có Customer ID nên ở đây đầu tiên em sẽ tạo ra Customer ID bằng hàm random

id\_lst = []

for i in data\_sales.values:

  if i[4] == 'Female':

    if i[3] == 'Member':

      id = np.random.randint(1,50)

    else:

      id = np.random.randint(50,100)

  else:

    if i[3] == 'Member':

      id = np.random.randint(100,150)

    else:

      id = np.random.randint(150,201)

  id\_lst.append(id)

print(id\_lst)

- Ở dữ liệu này có cột loại khách hàng (Customer Type) và giới tính (Gender) do đó ở đây đầu tiên em sẽ chia làm 2 loại khách hàng sau đó tới giới tính và dùng hàm Random để tạo ngẫu nhiên các Customer ID

- Sau khi đã tạo được Customer ID ta sẽ add dữ liệu này vào data.

data\_sales.drop(columns = 'Satisfaction Level', axis = 1, inplace = True)

new\_df = pd.concat([data\_sales,pd.DataFrame(id\_lst)],axis =1)

new\_df.head()

- Đổi lại tên cột vừa add vào bảng thành Customer ID

new\_df.rename(columns = {0:'Customer ID'}, inplace=True)

new\_df.head()

Graphical user interface, text, application

Description automatically generated

- Sau khi đã có đẩy đủ để có những thứ cần, thì ta sẽ tiến hành thực hiện RFM với bộ dữ liệu Supermarket sale:

* R - Recency: Khoảng thời gian mua hàng gần nhất của khách hàng
* F – Frequency: Tần suất mua hàng của khách hàng
* M – Monetary: Giá trị mỗi lần mua hàng

- Sau khi tính toán 3 biến RFM thì ta sẽ group by lại 3 biến này

# Lấy ngày lớn nhất cộng 1

current\_date = max(new\_df['Date']) + dt.timedelta(days = 1)

# Groupby Customer ID để tính RFM

df\_customers = new\_df.groupby(['Customer ID']).agg(

    {

        'Date': lambda x: (current\_date - x.max()).days,

        'Invoice ID': 'count',

        'Total': 'sum'

    }

)

df\_customers.head()

- Hiển thị kết quả sau group by:

Graphical user interface, application

Description automatically generated

- Để cho đẹp hơn thì ở đây ta sẽ đổi tên lại 3 cột Date, Invoice ID và Total thành Recency, Frequency và Monetary

df\_customers.rename(columns = {'Date': 'Recency', 'Invoice ID': 'Frequency', 'Total': 'Monetary'}, inplace = True)

df\_customers.info()

Text

Description automatically generated

- Tiếp đó, ta vẽ phân phối dữ liệu 3 biến RFM ra để xem thử

# Vẽ phân phối 3 biến RFM

plt.figure(figsize = (24,16))

plt.subplot(2,3,1)

sns.distplot(df\_customers['Recency'])

plt.title('Biểu đồ phân phối biến Recency', fontsize = 16)

plt.xlabel('Recency', fontsize = 14)

plt.ylabel('Density', fontsize = 14)

plt.subplot(2,3,2)

sns.distplot(df\_customers['Frequency'])

plt.title('Biểu đồ phân phối biến Frequency', fontsize = 16)

plt.xlabel('Frequency', fontsize = 14)

plt.ylabel('Density', fontsize = 14)

plt.subplot(2,3,3)

sns.distplot(df\_customers['Monetary'])

plt.title('Biểu đồ phân phối biến Monetary', fontsize = 16)

plt.xlabel('Monetary', fontsize = 14)

plt.ylabel('Density', fontsize = 14)

plt.show()

Chart, histogram

Description automatically generated

- Ở đây, ta thấy dữ liệu 3 biến RFM không đều (lệch phải), nên ta sẽ tiến hành transform.

- Ta có 3 cách tranform dữ liệu là xài: log, sqrt và box cox

from scipy import stats

def analyze\_skewness(x):

    fig, ax = plt.subplots(2, 2, figsize=(12,8))

    sns.distplot(df\_customers[x], ax=ax[0,0])

    sns.distplot(np.log(df\_customers[x]), ax=ax[0,1])

    sns.distplot(np.sqrt(df\_customers[x]), ax=ax[1,0])

    sns.distplot(stats.boxcox(df\_customers[x])[0], ax=ax[1,1])

    plt.tight\_layout()

    plt.show()

    print(df\_customers[x].skew().round(2))

    print(np.log(df\_customers[x]).skew().round(2))

    print(np.sqrt(df\_customers[x]).skew().round(2))

    print(pd.Series(stats.boxcox(df\_customers[x])[0]).skew().round(2))

- Ở đây ta tiến hành xài từng cách một và so sánh xem cách nào tối ưu nhất thì ta sẽ chọn

- Kết quả:

* Recency

Chart, line chart

Description automatically generated

* Frequency

Chart, histogram

Description automatically generated

* Monetary

Chart, histogram

Description automatically generated

- Ở đây sau khi tính toán ta thấy sau khi tranform bằng boxcox thì ta thấy dữ liệu của 3 biến gần bằng 0 nhất. Vì vậy ta sẽ chọn boxcox để transform data

#tiến hành transform

df\_customer\_tf = pd.DataFrame()

df\_customer\_tf['Recency'] = stats.boxcox(df\_customers['Recency'])[0]

df\_customer\_tf['Frequency'] = stats.boxcox(df\_customers['Frequency'])[0]

df\_customer\_tf['Monetary'] = stats.boxcox(df\_customers['Monetary'])[0]

df\_customer\_tf.head(10)

Text

Description automatically generated

- Tiếp theo, ta tiến hành scale dữ liệu các biến về 1 khoảng tỉ lệ quy đổi bằng Standard Scaler

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(df\_customer\_tf)

df.customer\_tf = scaler.transform(df\_customer\_tf)

pd.DataFrame(df.customer\_tf).head(10)

Text

Description automatically generated

- Tạo bản copy để tính điểm theo mô hình RFM

#tạo 1 bản copy của df\_customers để tính điểm theo mô hình RFM

dfc\_copy = df\_customers.copy()

#tính điểm rfm

dfc\_copy["recency\_score"] = pd.qcut(dfc\_copy['Recency'], 5, labels = [5, 4, 3, 2, 1])

dfc\_copy["frequency\_score"] = pd.qcut(dfc\_copy['Frequency'].rank(method = "first"), 5, labels = [1, 2, 3, 4, 5])

dfc\_copy["monetary\_score"] = pd.qcut(dfc\_copy['Monetary'], 5, labels = [1, 2, 3, 4, 5])

dfc\_copy['score'] = dfc\_copy['recency\_score'].astype(int) + dfc\_copy['frequency\_score'].astype(int) + dfc\_copy['monetary\_score'].astype(int)

- Tạo ra hàm rfm\_scores\_ranking để đánh giá hạng khách hàng

* [1-4): Khách hàng ít tiềm năng
* [4-7): Khách hàng thông thường
* [7-11): Khách hàng tiềm năng
* Còn lại: Khách hàng vip

def rfm\_scores\_ranking(score):

  if ((score) >= 1 and (score < 4)):

    return 'Khách hàng ít tiềm năng'

  elif ((score >= 4) and (score < 7)):

    return 'Khách hàng thông thường'

  elif ((score >= 7) and (score < 11)):

    return 'Khách hàng tiềm năng'

  else:

    return 'Khách hàng V.I.P'

dfc\_copy['Customer Type'] = dfc\_copy['score'].apply(lambda score : rfm\_scores\_ranking(score))

dfc\_copy.head()

- Hiển thị:

Graphical user interface, application

Description automatically generated

- Một số biểu đồ về chỉ số RFM

* Biều đồ thể hiện điểm khách hàng theo mô hình RFM
* Biểu đồ phân loại khách hàng theo mô hình RFM

plt.figure(figsize = (30,20))

plt.subplot(2,3,1)

plt.title('Biểu đồ thể hiện điểm khách hàng theo mô hình RFM', fontsize = 20)

plt.xlabel('Điểm RFM', fontsize = 14)

plt.ylabel('Count', fontsize = 14)

sns.countplot(dfc\_copy['score'])

plt.subplot(2,3,2)

plt.title('Biểu đồ phân loại khách hàng', fontsize = 20)

sns.countplot(dfc\_copy['Customer Type'])

plt.xlabel('Loại khách hàng', fontsize = 14)

plt.ylabel('Count', fontsize = 14)

plt.show()

Chart, bar chart, histogram

Description automatically generated

# CHƯƠNG 4. ÁP DỤNG CÁC THUẬT TOÁN MÁY HỌC VÀ BIỂU DIỄN TRỰC QUAN

## 4.1 K-means

### 4.1.1 Giới thiệu về Kmeans.

K-mean clustering là một phương pháp để tìm các cụm và hạt nhân - trung tâm của cụm trong một tập hợp dữ liệu không được gắn nhãn. Người ta chọn số lượng hạt nhân cụm mong muốn phân chẳng hạn như k cụm. Thuật toán K-mean di chuyển lặp đi lặp lại các hạt nhân để giảm thiểu tổng số trong phương sai cụm. Với một tập hợp các hạt nhân ban đầu, thuật toán Kmeans lặp lại hai bước:

* Đối với mỗi hạt nhân, tính toán khoảng cách giữa các training ponit với nó và nếu gần nó hơn -> sẽ gán là cụm của hạt nhân đấy
* Sau khi phân được cụm như ở bước trên, thì tiếp theo các training point của các cụm tính toán vector trung bình để được vị trí của hạt nhân mới và lặp lại bước trên đến khi không thể thay đổi được vị trí hạt nhân nữa.

Phân cụm có nhiều hữu ích đặc biệt và cực kỳ phổ biến trong ngành khoa học dữ liệu. Trong đó như:

* Phân tích cụm được sử dụng rộng rãi trong nghiên cứu thị trường, nhận dạng mẫu, phân tích dữ liệu và xử lý ảnh.
* Phân tích cụm cũng có thể giúp các nhà khoa học dữ liệu khám phá ra các nhóm khác hàng của họ. Và họ có thể mô tả đặc điểm nhóm khách hàng của mình dựa trên lịch sử mua hàng.

### 4.1.2 Áp dụng thuật toán và biễu diên trực quan.

- Đầu tiên, trước khi tiến hành phân cụm bằng Kmeans thì ta sẽ dùng elbow để xác định chọn số k tối ưu trước khi phân cụm

# Chọn số cụm k bằng Elbow

from sklearn.cluster import KMeans

sse = {}

for k in range(1, 11):

  kmeans = KMeans(n\_clusters = k, random\_state=42)

  kmeans.fit(df.customer\_tf)

  sse[k] = kmeans.inertia\_

plt.figure(figsize = (16,8))

plt.title('The Elbow', fontsize = 20)

plt.xlabel('k', fontsize = 15)

plt.ylabel('SSE', fontsize = 15)

sns.pointplot(x=list(sse.keys()), y = list(sse.values()))

plt.show()

Chart, line chart

Description automatically generated

- Ở đây, ta sẽ chọn k = 4 vì số lượng cụm lớn hơn 4 thì tốc độ của hàm thay đổi dường như không đáng kể so với trước

- Tiến hành phân cụm bằng Kmeans

model = KMeans(n\_clusters=4, random\_state=42)

model.fit(df.customer\_tf)

df\_customers['Cluster'] = model.labels\_

df\_customers.head()

Graphical user interface, application

Description automatically generated

- Thử thống kê trung bình các cụm theo RFM

df\_customers.groupby('Cluster').agg(

    {

        'Recency':'mean',

        'Frequency':'mean',

        'Monetary':'mean'

    }

).round(2)

A picture containing table

Description automatically generated

- Vẽ các cụm theo các biến RFM

plt.figure(figsize = (20, 15))

plt.subplot(2, 3, 1)

sns.scatterplot(data = df\_customers, x = df\_customers['Recency'], y = df\_customers['Frequency'], hue = df\_customers['Cluster'], palette = "flare")

plt.xlabel('Recency', fontsize = 12)

plt.ylabel('Frequency', fontsize = 12)

plt.subplot(2, 3, 2)

sns.scatterplot(data = df\_customers, x = df\_customers['Recency'], y = df\_customers['Monetary'], hue = df\_customers['Cluster'], palette = "flare")

plt.xlabel('Recency', fontsize = 12)

plt.ylabel('Monetary', fontsize = 12)

plt.subplot(2, 3, 3)

sns.scatterplot(data = df\_customers, x = df\_customers['Frequency'], y = df\_customers['Monetary'], hue = df\_customers['Cluster'], palette = "flare")

plt.xlabel('Frequency', fontsize = 12)

plt.ylabel('Monetary', fontsize = 12)

plt.show()

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

## 4.2 HAC

### 4.2.1 Giới thiệu về HAC:

Thuật toán phân cụm K-means cho thấy cần phải cấu hình trước số lượng cụm cần phân chia. Ngược lại, phương pháp *phân cụm phân cấp* (*Hierachical Clustering*) không yêu cầu khai báo trước số lượng cụm. Thay vào đó, thuật toán chỉ yêu cầu xác định trước thước đo về sự khác biệt giữa các cụm (không giao nhau), dựa trên sự khác biệt từng cặp giữa các quan sát trong hai cụm. Theo phương pháp này, chúng tạo ra những biểu diễn phân cấp trong đó các cụm ở mỗi cấp của hệ thống phân cấp được tạo bằng cách hợp nhất các cụm ở cấp độ thấp hơn bên dưới. Ở cấp thấp nhất, mỗi cụm chứa một quan sát. Ở cấp cao nhất, chỉ có một cụm chứa tất cả dữ liệu.

Các chiến lược *phân cụm phân cấp* chia thành hai mô hình cơ bản: Hợp nhất (*agglomerative*) và phân chia (*divisive*). Trước khi tìm hiểu về hai chiến lược này, tôi khuyến nghị bạn đọc ôn tập lại kiến thức cây quyết định để nắm rõ các thành phần trong cây quyết định. Trục hoành thể hiện index của các quan sát trong nhóm được phân vào một cụm, trong khi tục tung là gía trị thước đo sự khác biệt giữa các cụm.Một cụm được đại diện bởi một node mà toàn bộ các quan sát khác nếu thuộc cụm thì đều liên kết tới node đó. Như vậy chúng ta có thể nhận thấy rằng các cụm có sự phân cấp dựa vào level của node. Khi kẻ một đường thẳng nằm ngang cắt toàn bộ các đường thẳng thẳng đứng ta sẽ thu được các cụm tương ứng với các node nằm gần nhất bên dưới đường thẳng. Bất kì hai cụm nào trong số chúng sẽ không chồng lấn nhau.

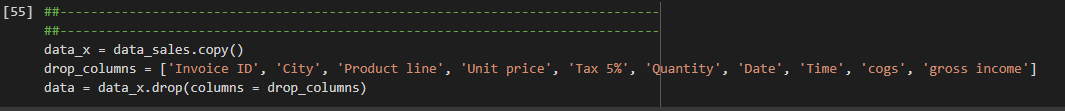
Thuật toán *phân cụm phân cấp* được xây dựng trên bộ dữ liệu có kích thước *N* thì sẽ trải qua tổng cộng *N* bước phân chia. Có hai chiến lược phân chia chính phụ thuộc vào chiều di chuyển trên biểu đồ *dendrogram* mà chúng ta sẽ tìm hiểu bên dưới:

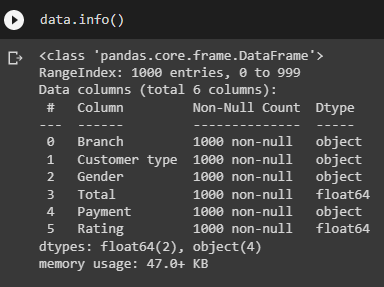
* **Chiến lược hợp nhất**: Chiến lược này sẽ đi theo chiều *bottum-up* (từ dưới lên trên). Quá trình phân cụm bắt đầu ở dưới cùng tại các node lá (còn gọi là *leaf node* hoặc *termial node*). Ban dầu mỗi quan sát sẽ được xem là một cụm tách biệt được thể hiện bởi một node lá. Ở mỗi level chúng ta sẽ tìm cách hợp một cặp cụm thành một cụm duy nhất nhằm tạo ra một cụm mới ở level cao hơn tiếp theo. Cụm mới này tương ứng với các node quyết định (*non-leaf node*). Như vậy sau khi hợp cụm thì số lượng cụm ít hơn. Một cặp được chọn để hợp nhất sẽ là những cụm trung gian không giao nhau.
* **Chiến lược phân chia**: Chiến lược này sẽ thực hiện theo chiều *top-down*. Tức là phân chia bắt đầu từ node gốc của đồ thị. Node gốc bao gồm toàn bộ các quan sát, tại mỗi level chúng ta phân chia một cách đệ qui các cụm đang tồn tại tại level đó thành hai cụm mới. Phép phân chia được tiến hành sao cho tạo thành hai cụm mới mà sự tách biệt giữa chúng là lớn nhất. Sự tách biệt này sẽ được đo lường thông qua một thước đo khoảng cách mà ta sẽ tìm hiểu kĩ hơn bên dưới.

Như vậy đồ thị của *chiến lược phân chia* và *chiến lược hợp nhất* lđều là cây nhị phân, chúng chỉ khác biệt về chiều thực hiện thuật toán. Node gốc của cây nhị phân sẽ bao gồm toàn bộ các quan sát và cây nhị phân bao gồm *N* node lá đại diện cho *N* quan sát từ bộ dữ liệu. Mỗi một node quyết định bao gồm hai node con. Quá trình phân chia thì hai node con thể hiện kết quả được phân chia từ node cha và quá trình hợp nhất thì node cha là thể hiện kết quả sau khi gộp hai node con.

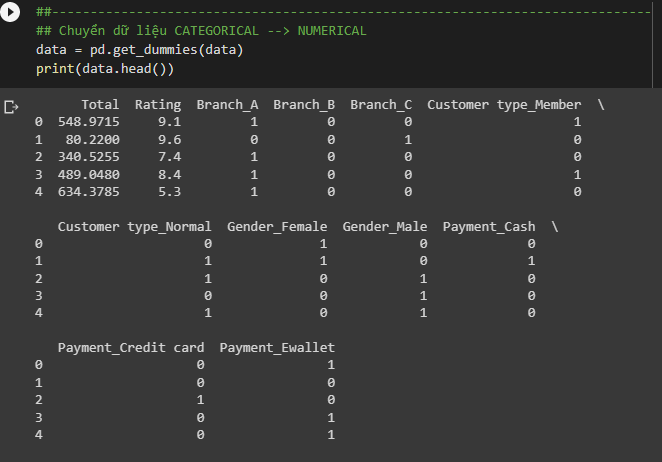
### 4.2.2 Áp dụng thuật toán và biểu diễn trực quan dữ liệu.

Đầu tiên ta loại đi những cột mà ta không cần

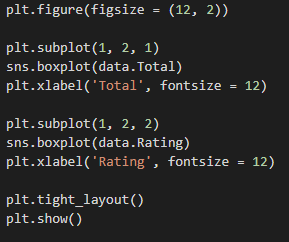


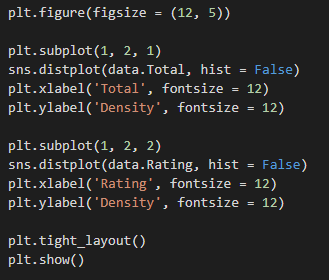


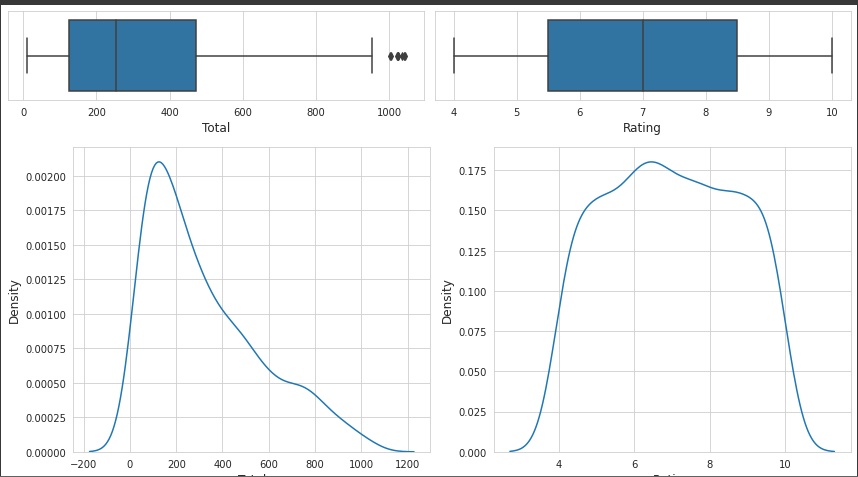
Ta phải chuyển dữ liệu Categorical thành dữ liệu Numerical



Kiểm tra outlier(s) dựa trên boxplot đối với các biến LIÊN TỤC qua đó ta có thể chọn phương pháp chuẩn hóa dữ liệu và chúng ta phải kiểm tra biến có phân phối chuẩn không

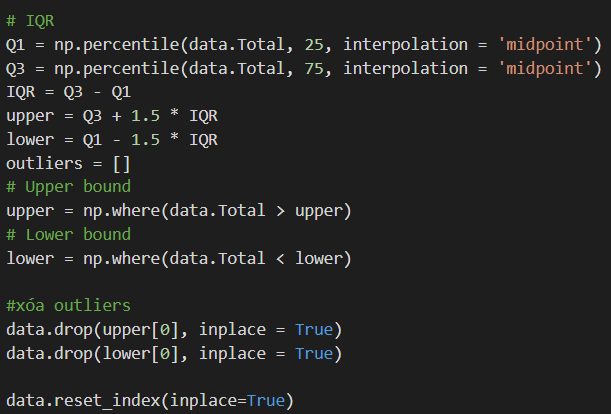




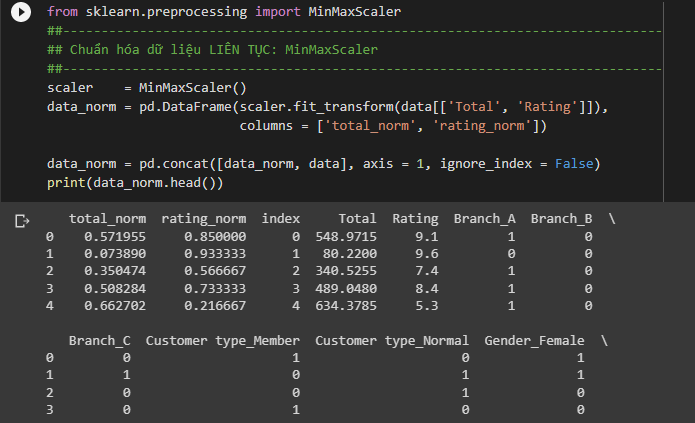


- Qua biểu đồ ta có thể thấy các features 'Quantity' và 'Total' và 'Rating' không có phân phối chuẩn. 2 features 'Quantity' và 'Rating' không có outliers, 'Total' có outliers nên ta cần loại bỏ outliers.

- Ta loại bỏ outliers bằng IQR



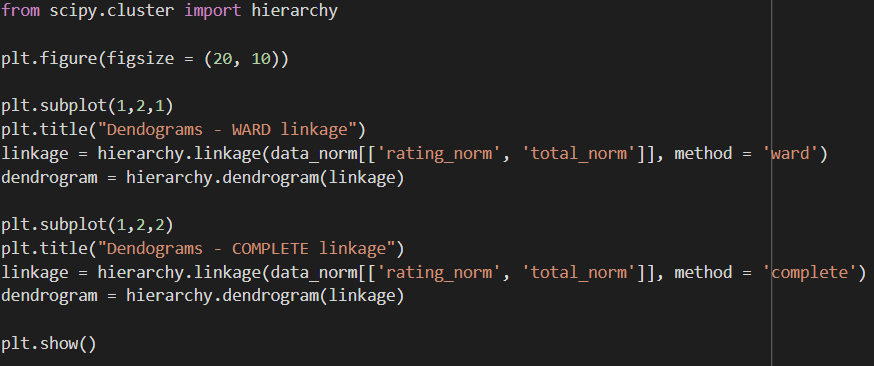
- Từ box plot ta chọn chuẩn hóa dữ liệu bằng phương pháp MinMaxScaler

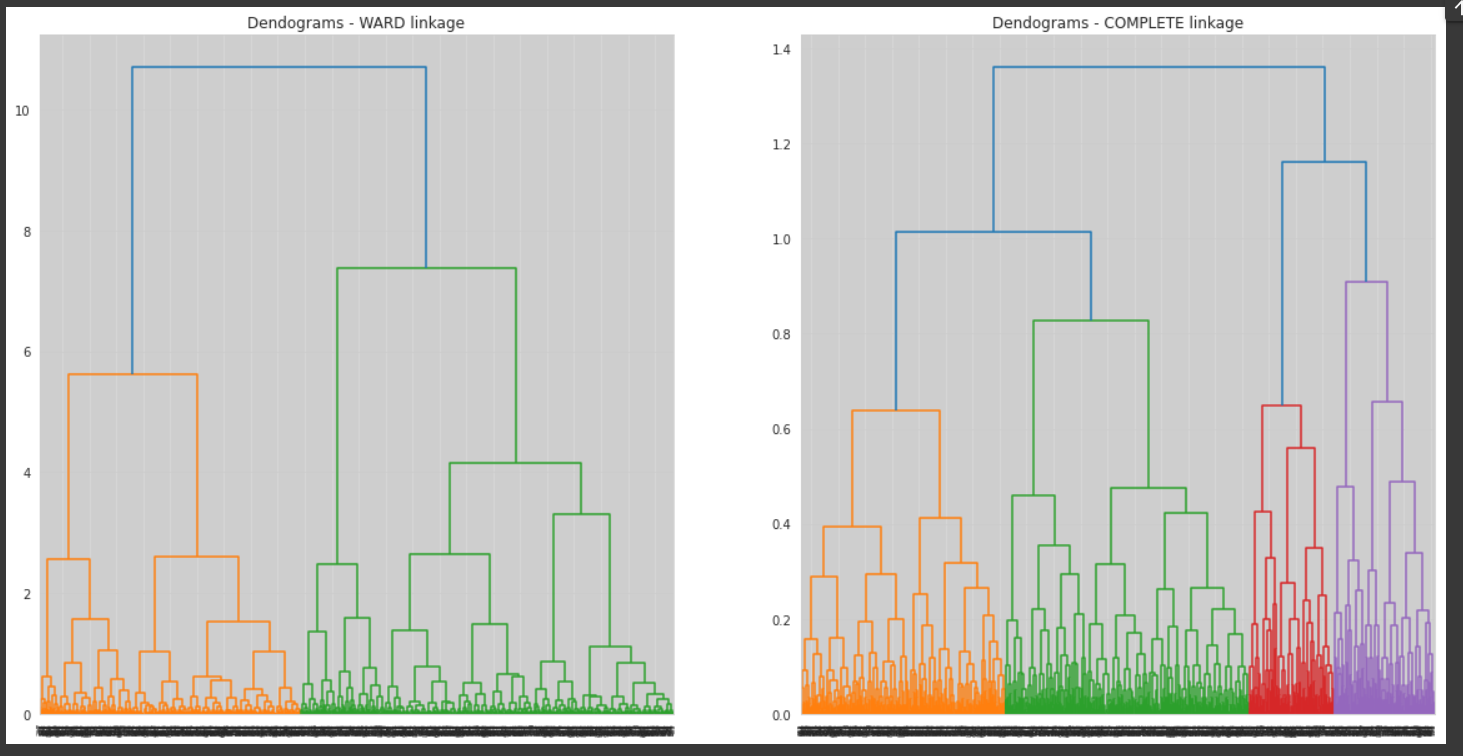


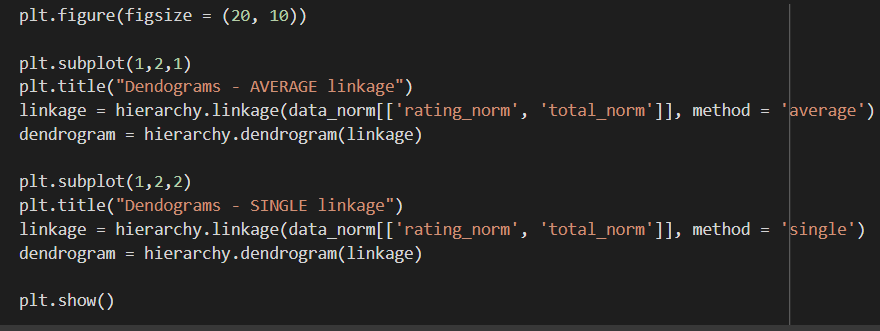
- Sau đó ta biểu diễn trực quan dữ liệu của Rating - Total

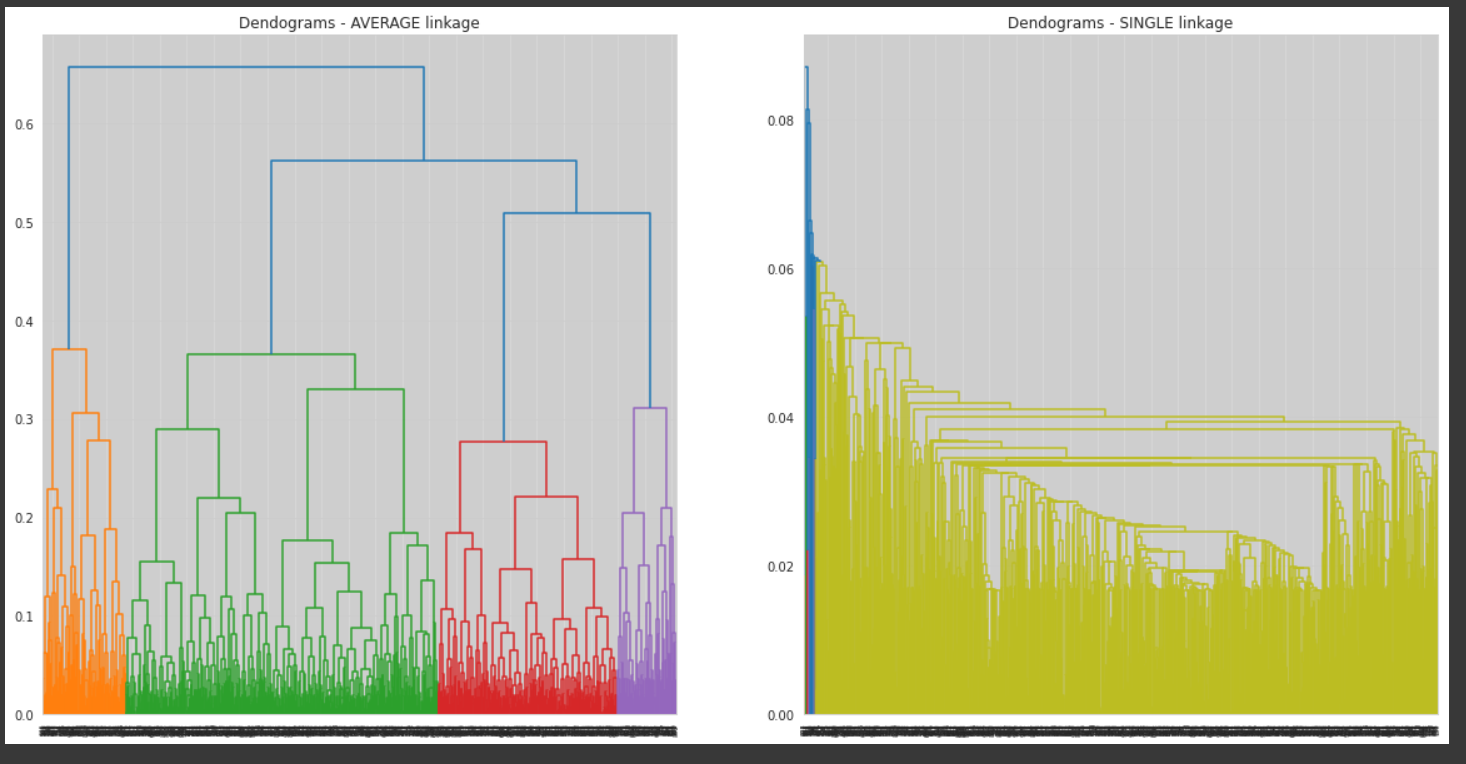


- Ta bắt đầu gom cụm bằng thuật toán HAC. Ta bắt đầu bằng việc vẽ dendrogram



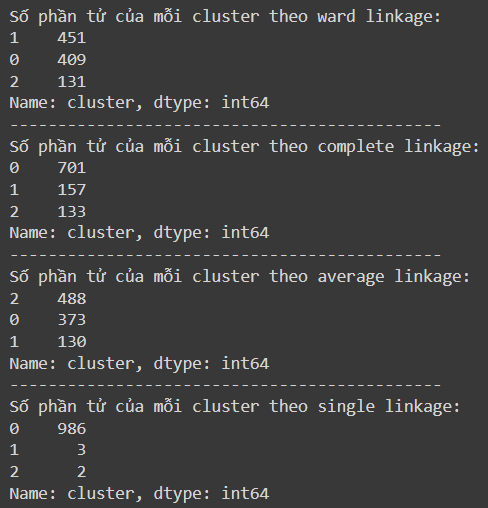




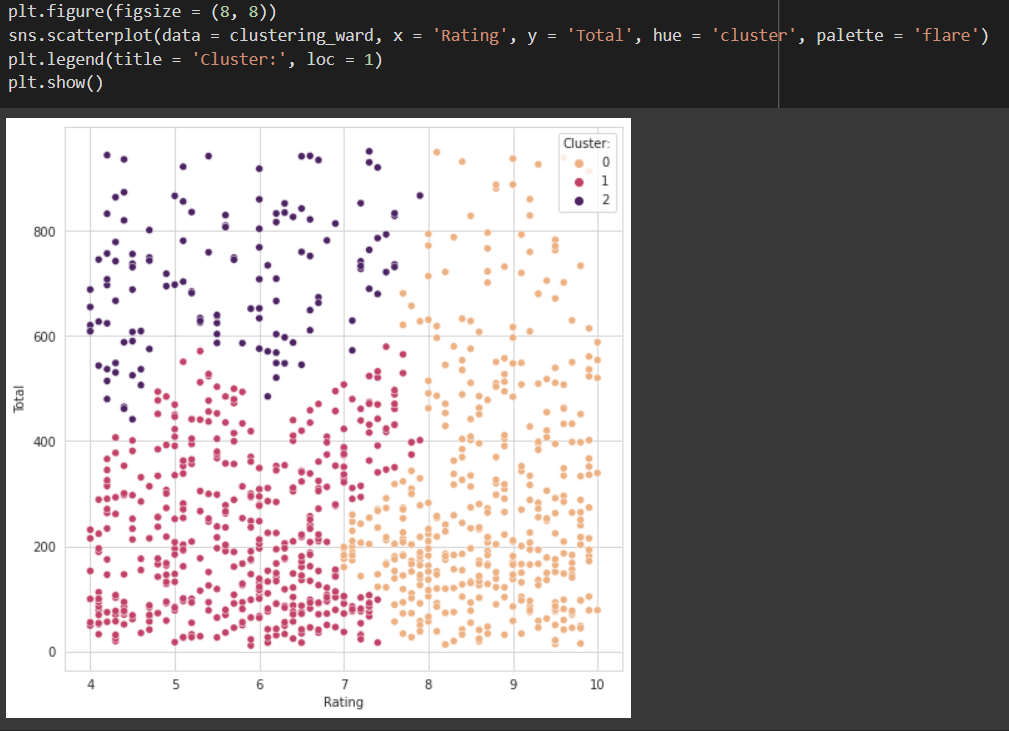


- Ta chọn k dựa trên biểu đồ scatter plot





- Vì phân cụm theo ward linkage cho ra 3 cụm tốt hơn so với các phương pháp còn lại nên ta chọn ward linkage và ta biểu diễn kết quả phân cụm



# CHƯƠNG 5: KIỂM ĐỊNH VÀ GIẢM CHIỀU DỮ LIỆU

## 5.1. Kiểm định.

### 5.1.1 Kiểm định Chi-Square.

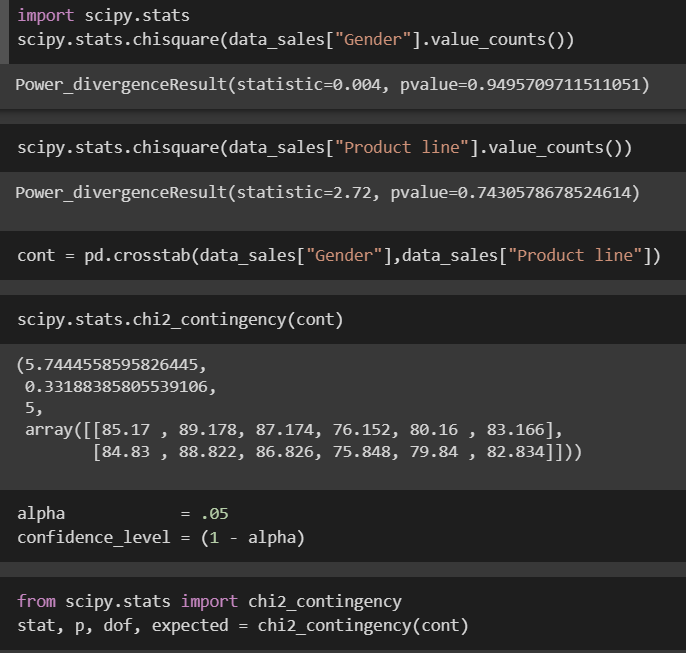
**-** Giới thiệu về kiểm định Chi Square: (kiểm định chi bình phương) là một phép thử thống kê được sử dụng để so sánh kết quả quan sát và kết quả mong đợi giữa các biến phân loại trong tổng thể dữ liệu.

- Áp dụng:

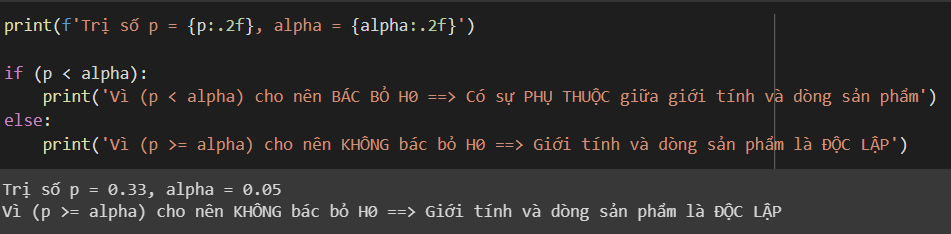
Ta tiến hành kiểm định với H0 và Ha như sau:



Ta lấy 2 biến Gender và Product line để làm 2 biến mẫu để tính giá trị quan sát và giá trị kỳ vọng



- Từ đó ta kiểm định giả thuyết (H0: giới tính và dòng sản phẩm độc lập, H1: có sự phục thuộc giữa giới tính và dòng sản phẩm)



### 5.1.2 Kiểm định one-way ANNOVA.

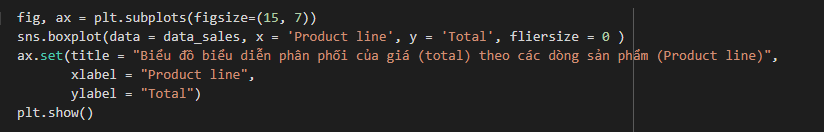
- Giới thiệu về kiểm định ANOVA: ANOVA test, hay còn được gọi là phân tích phương sai (Analysis of Variance), là một kỹ thuật thống kê tham số được sử dụng để phân tích sự khác nhau giữa giá trị trung bình của các biến phụ thuộc với nhau

- Áp dụng vào bài toán:

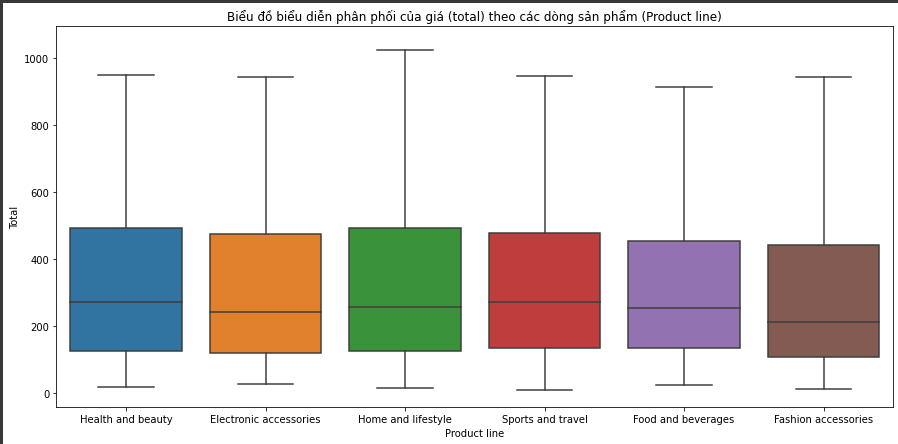
Ta tiến hành kiểm định với H0 và Ha như sau:



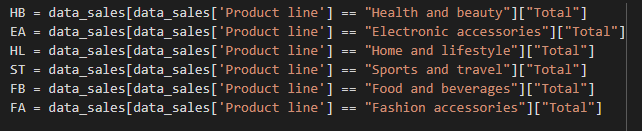
Đầu tiên, ta sẽ vẽ biểu đồ boxplot để biểu diễn sự phụ thuộc của biến total so với biển các dòng sản phẩm (Product Line)



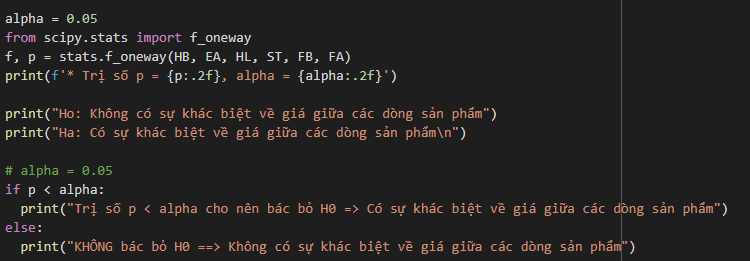
Ở biểu đồ boxplot này, nhóm em có biến fliersize = 0 để loại bỏ hết các outliner đi cho ta dễ thấy được sự tương quan của các biến.



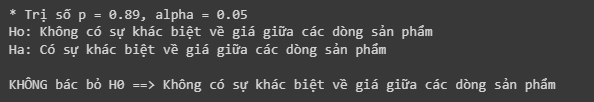
Do ở đây, data của nhóm em đang ở dạng **DataFrame** nên em sẽ đọc dữ liệu các biến ra bằng cách:



Sau khi đã đọc được dữ liệu các biến, ta sẽ tiến hành thực hiện kiểm định ANOVA với alpha = 0.05



Kết quả nhận được sẽ là:



Do ở đây, ta không bác bỏ H0 nên chúng ta sẽ không cần phải tiến hành thêm Hậu kiểm Tuskey HSD.

## 5.2 Giảm chiều dữ liệu.

### 5.2.1 Giới thiệu và phương pháp giảm chiều dữ liệu PCA.

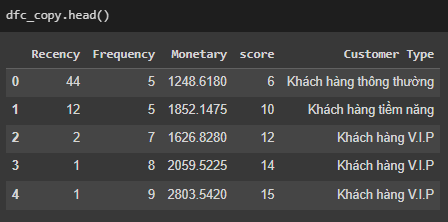
- Phép phân tích thành phần chính (Principal Components Analysis - PCA) là một [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) [thống kê](https://vi.wikipedia.org/wiki/Th%E1%BB%91ng_k%C3%AA) sử dụng phép biến đổi [trực giao](https://vi.wikipedia.org/wiki/Tr%E1%BB%B1c_giao) để biến đổi một [tập hợp](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p) [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u) từ một [không gian](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_gian) nhiều chiều sang một không gian mới ít chiều hơn (2 hoặc [3 chiều](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_gian_ba_chi%E1%BB%81u)) nhằm tối ưu hóa việc thể hiện [sự biến thiên](https://vi.wikipedia.org/w/index.php?title=S%E1%BB%B1_bi%E1%BA%BFn_thi%C3%AAn&action=edit&redlink=1) của dữ liệu.

- Phép biến đổi tạo ra những ưu điểm sau đối với dữ liệu:

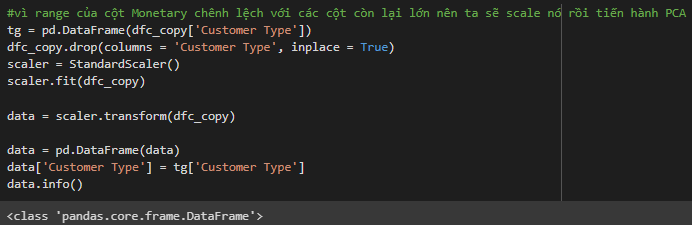
* Giảm số chiều của không gian chứa dữ liệu khi nó có số chiều lớn, không thể thể hiện trong không gian 2 hay 3 chiều.
* Xây dựng những [trục tọa độ](https://vi.wikipedia.org/w/index.php?title=Tr%E1%BB%A5c_t%E1%BB%8Da_%C4%91%E1%BB%99&action=edit&redlink=1) mới, thay vì giữ lại các trục của không gian cũ, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương, và đảm bảo độ biến thiên của dữ liệu trên mỗi chiều mới.
* Tạo điều kiện để các liên kết tiềm ẩn của dữ liệu có thể được khám phá trong không gian mới, mà nếu đặt trong không gian cũ thì khó phát hiện vì những liên kết này không thể hiện rõ.
* Đảm bảo các trục tọa độ trong không gian mới luôn trực giao đôi một với nhau, mặc dù trong không gian ban đầu các trục có thể không trực giao.

### 5.2.1 Áp dụng phương pháp với dữ liệu sau khi áp dụng mô hình RFM (với TARGET là Customer Type).

- Ta xem dữ liệu trước khi áp dụng phương pháp PCA.

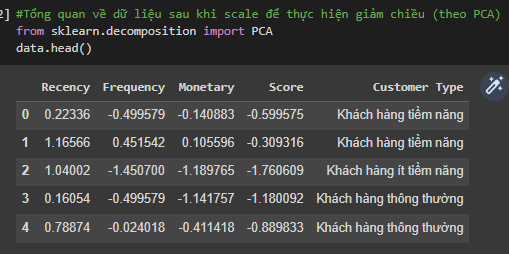


- Ta thấy range của Monetary chênh lệch khá lớn với range của Recency, Frequency và score nên ta sẽ tiến hành scale dữ liệu (Monetary, Recency, Frequency và score) trước khi áp dụng phương pháp PCA.

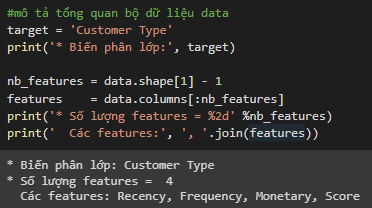


- Sau khi đã scale dữ liệu xong ta sẽ đặt lại tên các cột như ban đầu và được bộ dữ liệu như hình dưới đây.

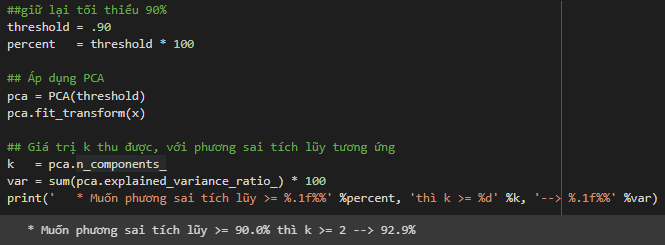




- Mô tả tổng quan về bộ dữ liệu:

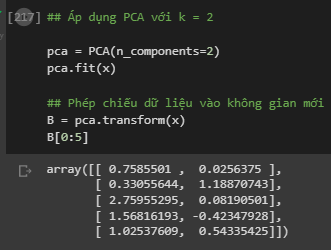


- Nhóm em muốn ngưỡng phương sai là 90% nên sẽ tiến hành chọn k từ ngưỡng phương sai này:

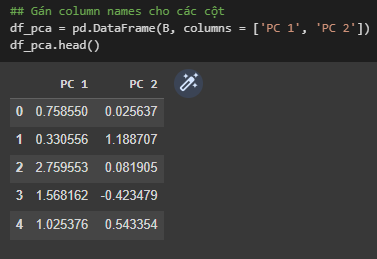


- Ta thấy, muốn phương sai tích lũy >= 90% thì k >= 2 nên nhóm chúng em sẽ chọn k = 2 và tiến hành giảm chiều dữ liệu với PCA.

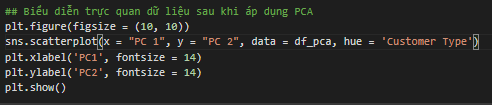
- Áp dụng pca với k = 2, fit bộ dữ liệu data (đã bỏ cột Customer Type). Sau đó, chiếu dữ liệu vào không gian mới.

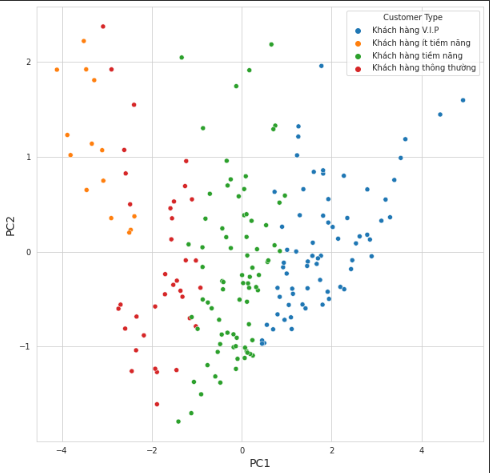


- Chuyển thành dataframe và đặt tên cho các cột.



- Ta tiến hành biễu diễn tương quan giữa 2 biến sau khi giảm chiều dữ liệu xuống còn 2 với hue là Customer Type.





Trích từ:

-<https://phamdinhkhanh.github.io/deepai-book/ch_ml/index_HierarchicalClustering.html>

- <https://msdnvietnam.net/tai-lieu-hoc-tap-kiem-dinh-chi-square/>