NHÓM 5

PHÂN TÍCH DỮ LIỆU BẰNG PYTHON

Chủ đề: CRM - LR

Thành viên: Trần Thị Mỹ
Trang Trà My
Lê Thị Phương
Dương Thị Kiều Doan
Nguyễn Thị Nhật Phương

MỤC LỤC

A. TỔNG QUAN
B. CƠ SỞ LÝ THUYẾT
C. KẾT QUẢ THỰC NGHIỆM

TỔNG QUAN

Quản lý quan hệ khách hàng (CRM) là một công nghệ được sử dụng để quản lý tất cả các mối quan hệ và tương tác giữa công ty và khách hàng cũng như khách hàng tiềm năng. Mục tiêu của CRM rất đơn giản: cải thiện mối quan hệ kinh doanh. Hệ thống CRM giúp các công ty tạo ra một kết nối chặt chẽ với khách hàng, tối ưu hóa quy trình làm việc và nâng cao lợi nhuận.

TẠI SAO PHẢI PHÂN TÍCH CRM?

DỰ ĐOÁN XU HƯỚNG

Đi sâu vào các xu hướng bán hàng trong lịch sử và so sánh chúng với điều kiện thị trường hiện tại để xác định hành vi mua hàng

CẢI THIỆN SỰ HÀI LÒNG CỦA KHÁCH HÀNG

Doanh nghiệp phân tích dữ liệu được đối chiếu từ nhiều kênh và tận dụng dữ liệu đó để nâng cao sự hài lòng của khách hàng

SÀNG LỌC BÁN HÀNG

Phân tích sở thích và xu hướng của khách hàng để xác định các phương pháp tiếp thị và bán hàng có nhiều khả năng mang lại kết quả tốt nhất

TĂNG CƯỜNG CÁC CHIẾN DỊCH TIẾP THỊ

Đánh giá hiệu suất của các chiến dịch tiếp thị và hợp lý hóa chi tiêu tiếp thị

CƠ SỞ LÝ THUYẾT VỀ PHÂN CỤM

PHÂN CỤM LÀ GÌ?

Phân cụm là quá trình gom các đối tượng tương tự nhau vào cùng một nhóm.

VÌ SAO PHẢI PHÂN CỤM?

- **Phân loại dữ liệu:** Phân loại dữ liệu tự động, ví dụ như phân loại khách hàng, phân loại email, v.v.
- **Tóm tắt dữ liệu:** Đơn giản hóa dữ liệu phức tạp bằng cách sắp xếp nó thành các nhóm có ý nghĩa.
- **Tăng hiệu quả thuật toán:** Giảm kích thước dữ liệu, giúp tăng hiệu quả của các thuật toán học máy khác.

QUY TRÌNH PHÂN CỤM

XỬ LÝ DỮ LIỆU

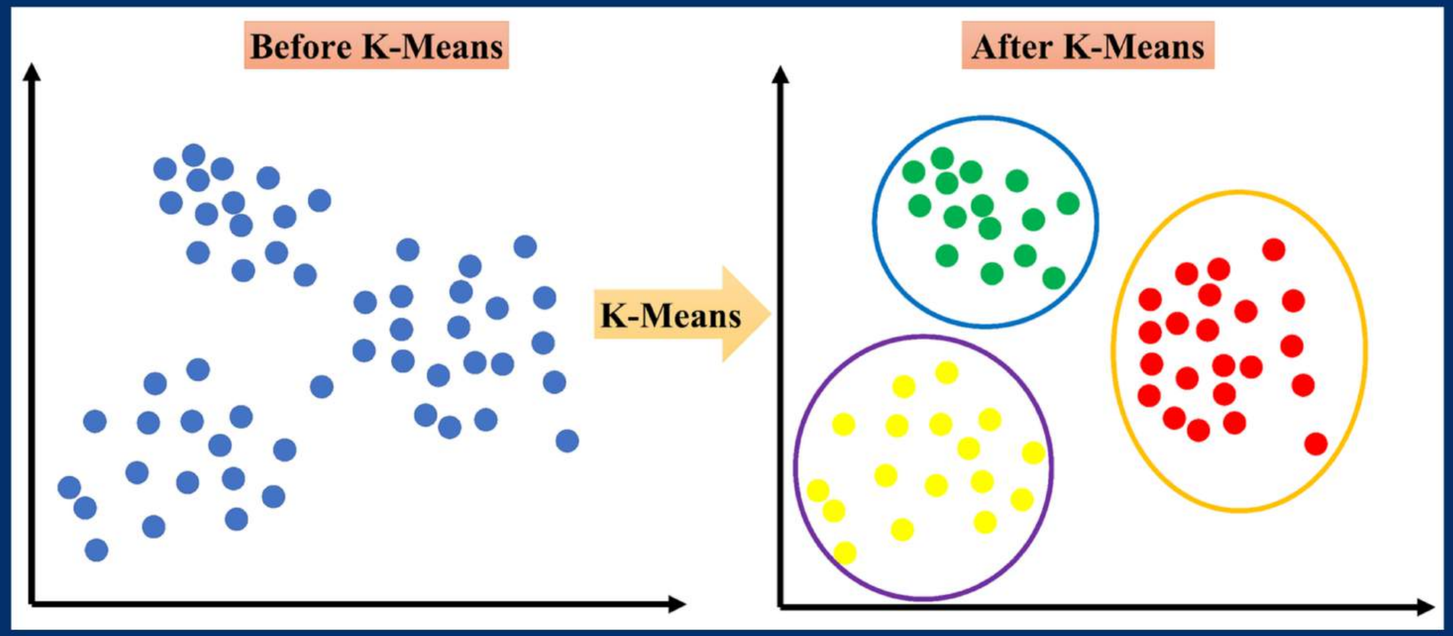
- Xử lý các giá trị thiếu và ngoại lệ.
- Chuẩn hóa dữ liệu để đảm bảo các thuộc tính có cùng đơn vị đo.
- Giảm kích thước dữ liệu nếu cần thiết.

LỰA CHỌN THUẬT TOÁN PHÂN CỤM

- K-means
- Hierarchical

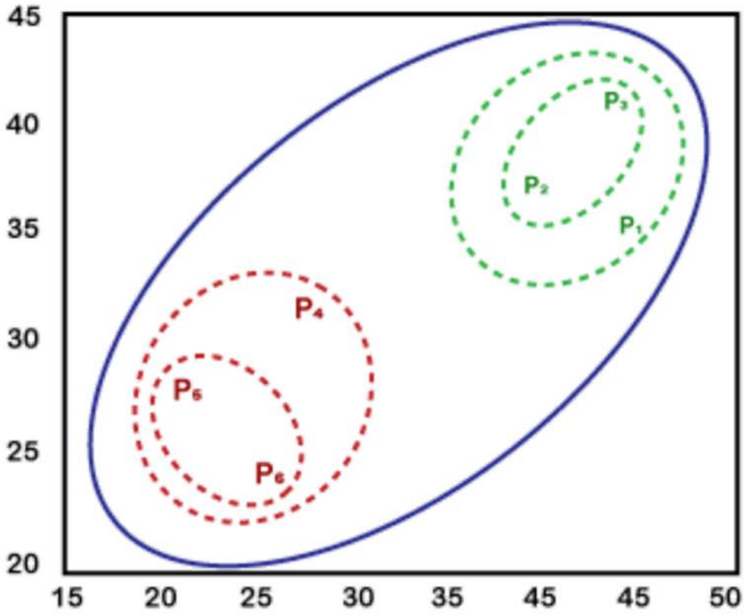
K-MEANS

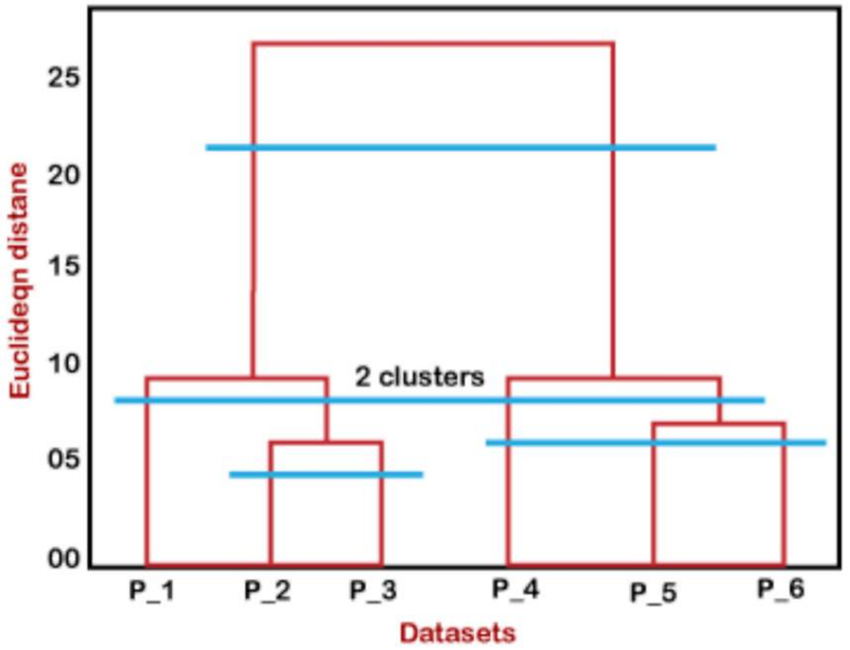
K-means là thuật toán rất quan trọng và được sử dụng phổ biến trong kỹ thuật phân cụm. Thuật toán K-Means là tìm cách phân nhóm các đối tượng (objects) sao cho tổng bình phương khoảng cách giữa các đối tượng đến tâm nhóm (centroid) là nhỏ nhất.



HIERARCHICAL

Hierarchical algorithm là phương pháp phân cụm dựa vào khoảng cách giữa các đối tượng để tạo ra một cây đối tượng có thứ bậc thể hiện mối quan hệ giữa các đối tượng và cho thấy cách các đối tượng được nhóm lại ở các cấp độ khác nhau.





QUY TRÌNH PHÂN CỤM

CHẠY THUẬT TOÁN PHÂN CỤM

Chọn các tham số cho thuật toán phân cụm và chạy thuật toán trên tập dữ liệu đã chuẩn bị.

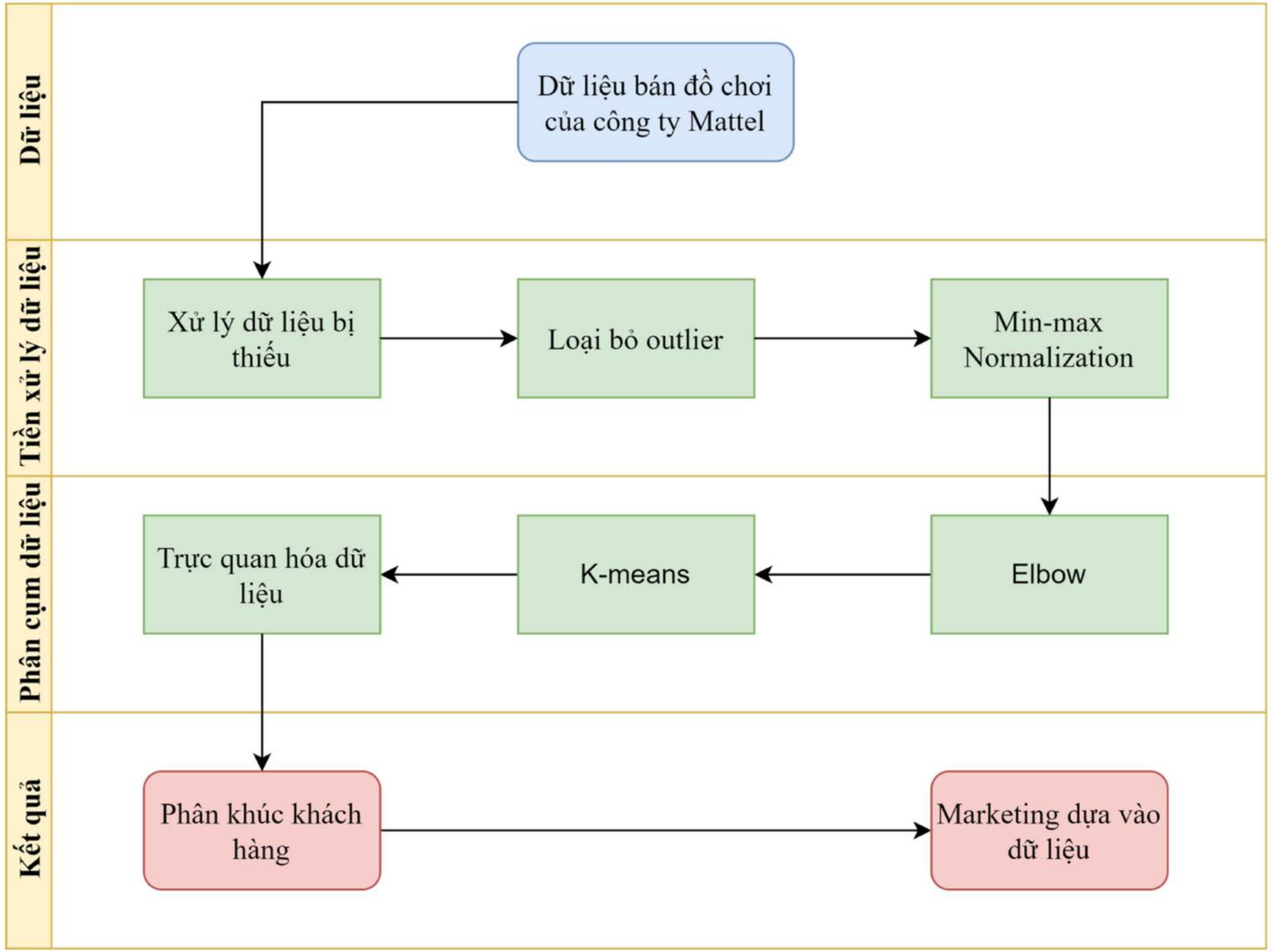
ĐÁNH GIÁ KẾT QUẢ PHÂN CỤM

Sử dụng các chỉ số đánh giá để đánh giá chất lượng của kết quả phân cụm

- Silhouette coefficient
- Elbow method

KẾT QUẢ THỰC NGHIỆM

FRAMEWORK


Dữ liệu
Dữ liệu bán đồ chơi của công ty Mattel
Tiền xử lý dữ liệu
Xử lý dữ liệu bị thiếu
Loại bỏ outlier
Min-max Normalization
Phân cụm dữ liệu
Trực quan hóa dữ liệu
K-means
Elbow
Kết quả
Phân khúc khách hàng
Marketing dựa vào dữ liệu

MÔ TẢ DATASET

COLUMN NAME	DESCRIPTION
ORDERNUMBER	Cột này thể hiện số ID nhận dạng duy nhất được gán cho mỗi đơn hàng.
QUANTITYORDERED	Nó cho biết số lượng mặt hàng được đặt hàng trong mỗi đơn hàng.
PRICEEACH	Cột này chỉ định giá của từng mặt hàng trong đơn hàng.
ORDERLINENUMBER	Nó đại diện cho số dòng của từng mục trong một đơn hàng.
SALES	Cột này biểu thị tổng số tiền bán được cho mỗi đơn hàng, được tính bằng cách nhân số lượng đặt hàng với giá của từng mặt hàng.
ORDERDATE	Nó biểu thị ngày đặt hàng.
DAYS_SINCE_LASTORDER	Cột này thể hiện số ngày đã trôi qua kể từ đơn hàng cuối cùng của mỗi khách hàng. Nó có thể được sử dụng để phân tích mô hình mua hàng của khách hàng.
STATUS	Nó cho biết trạng thái của đơn đặt hàng, chẳng hạn như "Shipped," "In Process," "Cancelled," "Disputed," "On Hold," or "Resolved."
PRODUCTLINE	Cột này chỉ định danh mục dòng sản phẩm mà mỗi mặt hàng thuộc về.
MSRP	Nó là viết tắt của Giá bán lẻ đề xuất của nhà sản xuất và thể hiện giá bán đề xuất cho từng mặt hàng.
PRODUCTCODE	Cột này thể hiện mã duy nhất được gán cho mỗi sản phẩm.
CUSTOMERNAME	Nó biểu thị tên của khách hàng đã đặt hàng.
PHONE	Cột này chứa số điện thoại liên hệ của khách hàng.
ADDRESSLINE1	Nó đại diện cho dòng đầu tiên trong địa chỉ của khách hàng.
CITY	Cột này chỉ định thành phố nơi khách hàng sinh sống.
POSTALCODE	Nó biểu thị mã bưu chính hoặc mã ZIP được liên kết với địa chỉ của khách hàng.
COUNTRY	Cột này cho biết quốc gia nơi khách hàng sinh sống.
CONTACTLASTNAME	Nó đại diện cho họ của người liên hệ có liên quan đến khách hàng
CONTACTFIRSTNAME	Cột này biểu thị tên của người liên hệ được liên kết với khách hàng.
DEALSIZE	Nó cho biết quy mô của giao dịch hoặc đơn đặt hàng, đó là các danh mục "Small," "Medium," or "Large."

Tập dữ liệu chứa dữ liệu Bán hàng của một Công ty đồ chơi

ĐỌC DỮ LIỆU

```
# Khai báo thư viện pandas
import pandas as pd

# Đọc file csv và lưu vào biến data
data=pd.read_csv('C:\\Users\\ADMIN\\Downloads\\HK2 NĂM 3\\Auto-Sales-data.csv')
print(data)
```

XỬ LÝ DỮ LIỆU

```
# Xóa dữ liệu NaN
# Xử lý dữ liệu bị thiếu bằng hàm dropna() loại bỏ các hàng có giá trị thiếu trong DataFrame data
data=data.dropna(axis=0)
print(data)
```

```
# Xóa dữ liệu trùng
# Sử dụng hàm drop_duplicates() để loại bỏ dữ liệu trùng lặp
data=data.drop_duplicates()
print(data)
```

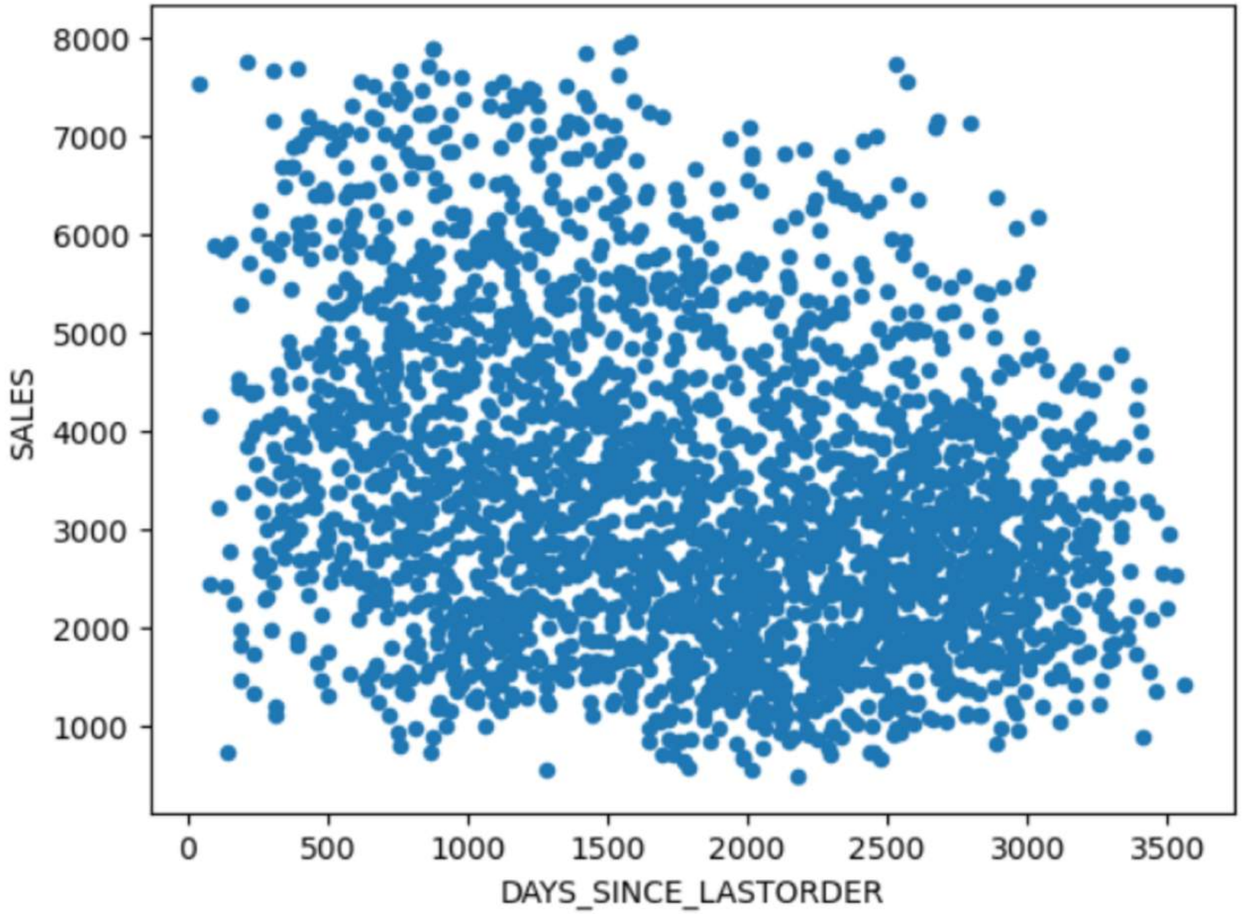
XỬ LÝ DỮ LIỆU

```
# Tính tứ phân vị lọc outliers

# Lấy những cột cần sử dụng để phân cụm để lọc outlier
columns=['DAYS_SINCE_LASTORDER','SALES']

# Tính toán Quantile và IQR:
# sử dụng vòng lặp for để duyệt qua 2 cột khai ở trên
for column in columns:
    q1 = data[column].quantile(0.25) # tính toán tứ phân vị thứ nhất (Q1) cho mỗi cột. Q1 là giá trị mà 25% dữ liệu nằm dưới.
    q3 = data[column].quantile(0.75) # tính toán tứ phân vị thứ ba (Q3) cho mỗi cột.  Q3 là giá trị mà 75% dữ liệu nằm trên.
    iqr = q3 - q1 # tính IQR
    # sử dụng hàm loc() để lọc theo nhãn và chỉ giữ lại các hàng mà giá trị trong mỗi cột nằm trong một phạm vi khoảng trả giữa:
    data = data.loc[(data[column] > q1 - 1.5 * iqr) & (data[column] < q3 + 1.5 * iqr)]

# Đặt lại index cho dữ liệu liền mạch để sử dụng cho các bước sau
data.reset_index(drop=True,inplace=True)
print(data)
```

```
# chuẩn hóa min-max
# Dòng này tạo một DataFrame mới tên df bằng cách dùng loc() để lọc theo nhãn tất cả các hàng của hai cột 'SALES'
# và 'DAYS_SINCE_LASTORDER' từ DataFrame đã lọc outlier.
df = data.loc[:, [ 'SALES','DAYS_SINCE_LASTORDER']]

# Định nghĩa một hàm tên min_max_scaling để chuẩn hoá min-max
def min_max_scaling(data):
  for column in data.columns: # Sử dụng vòng lặp for để duyệt qua từng cột
    min_value = data[column].min() # Tính giá trị nhỏ nhất và lưu vào biến min_value
    max_value = data[column].max() # Tính giá trị lớn nhất và lưu vào biến max_value
    data[column] = (data[column] - min_value) / (max_value - min_value) # Công thức chuẩn hoá min-max
  return data # Trả về dataframe đã được chuẩn hoá

# Truyền tham số dataframe df cho hàm tính chuẩn hoá min-max
df_scaled = min_max_scaling(df)
print(df_scaled)
```

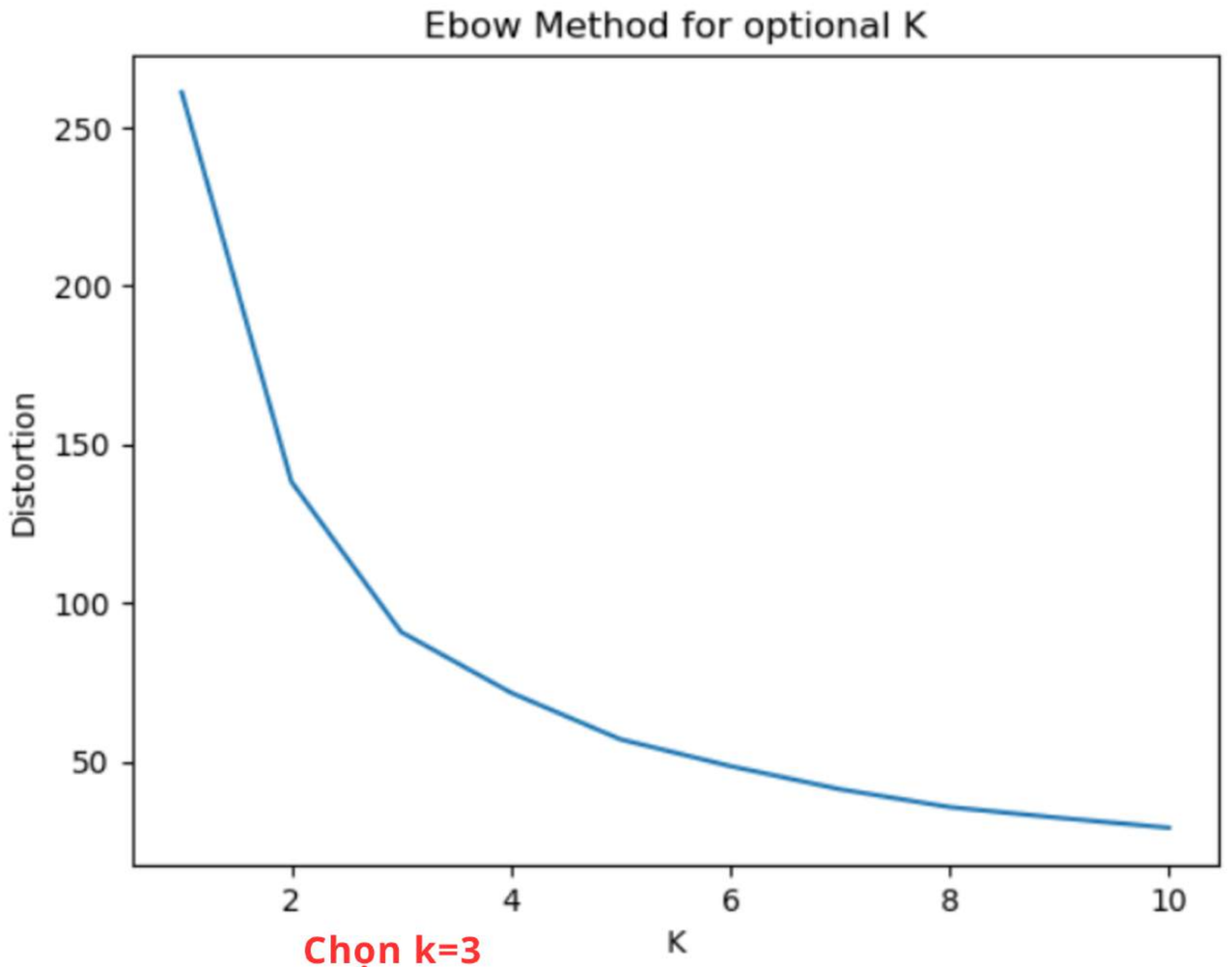
BIỂU ĐỒ PHÂN TÁN

```
# Vẽ biểu đồ phân tán trước khi phân cụm
# import thư viện matplotlib để tạo biểu đồ phân tán
import matplotlib.pyplot as plt

# sử dụng loại biểu đồ scatter với
# trục hoành là cột 'DAYS_SINCE_LASTORDER'
# và trục tung là cột 'SALES'
data.plot(kind = 'scatter',
          x = 'DAYS_SINCE_LASTORDER',
          y = 'SALES')

plt.show() # Hiển thị biểu đồ đã tạo
```



ELBOW

```python
# Xây dựng elbow
import numpy as np # khai báo thư viện numpy

# Sử dụng thư viện sklearn.cluster để tìm số cụm tối ưu
from sklearn.cluster import KMeans

# Tạo một danh sách k_range chứa các giá trị k từ 1 đến 10
k_range = range(1, 11)

# tạo 1 list rỗng
inertias = []

# Vòng lặp này sẽ thử nghiệm thuật toán K-Means với các
# giá trị k khác nhau để tìm ra k tối ưu.
for k in k_range:
  kmeans = KMeans(n_clusters=k)
  kmeans=kmeans.fit(df_scaled.to_numpy())
  inertias.append(kmeans.inertia_)

# Vẽ đồ thị elbow
plt.plot(k_range, inertias)
plt.xlabel('K')
plt.ylabel('Distortion')
plt.title('Ebow Method for optional K')
plt.show()
```



Chọn k=3

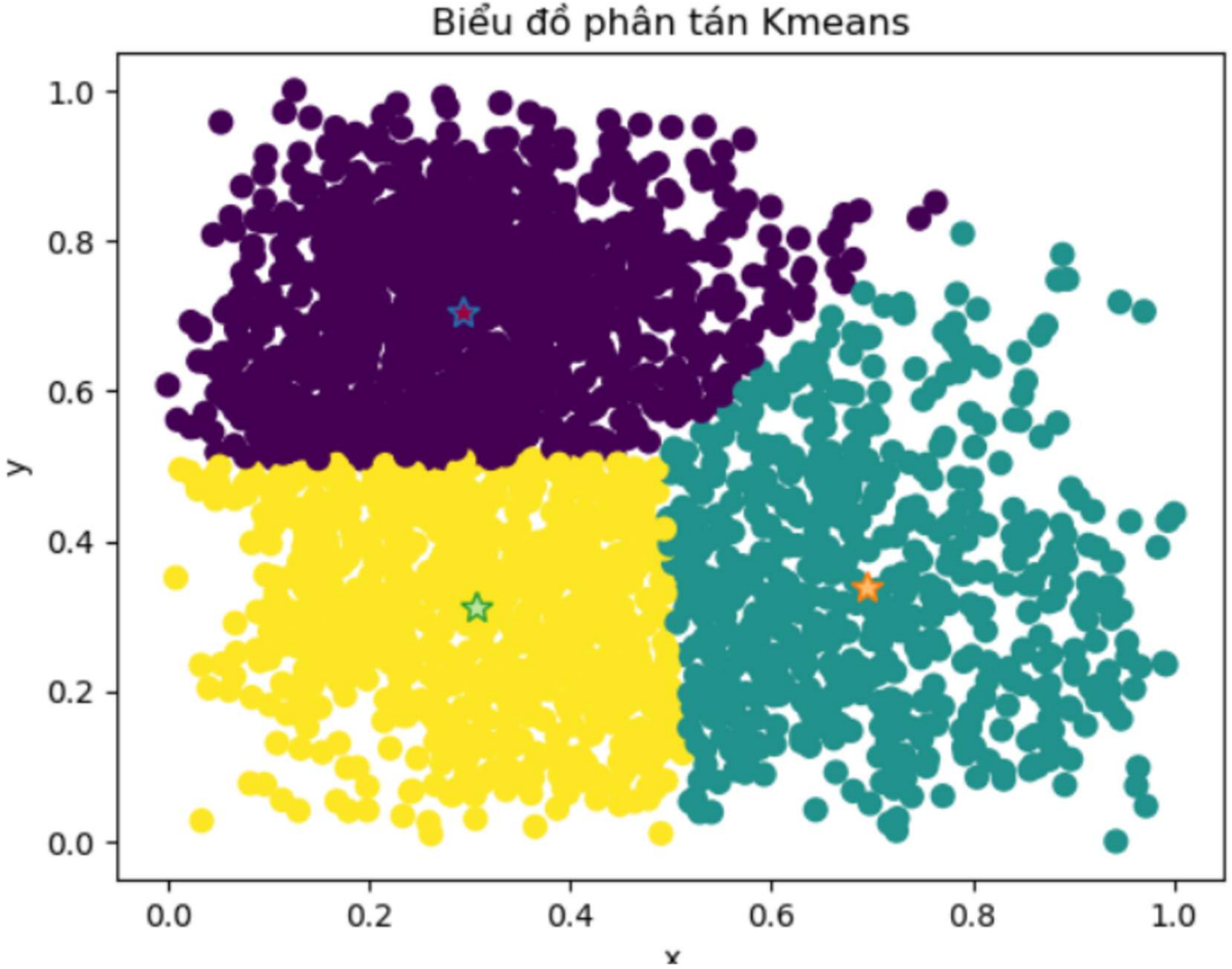
K-MEANS

```
# Dựa vào biểu đồ elbow ta chọn số cụm tối ưu k=3
# Khởi tạo đối tượng Kmeans với số cụm bằng 3
Kmean=KMeans(n_clusters=3)
Kmean.fit(df_scaled)

#Dòng này trích xuất nhãn cụm (cluster labels) từ đối tượng KMeans.
labels=Kmean.labels_

# Tạo ra cột Cluster
labels=pd.Series(labels,name='CLUSTER')

# Nối cột Cluster và dữ liệu đã chuẩn hoá
df_clustered=pd.concat([df_scaled,labels], axis=1)
print(df_clustered)
```

```
         SALES  DAYS_SINCE_LASTORDER  CLUSTER
0     0.320002              0.223295        2
1     0.305923              0.203125        2
2     0.455744              0.187784        2
3     0.437306              0.172443        2
4     0.401548              0.154545        2
...        ...                   ...      ...
2663  0.236065              0.818750        0
2664  0.468358              0.801989        0
2665  0.661128              0.793750        0
2666  0.218887              0.786364        0
2667  0.347923              0.775568        0

[2668 rows x 3 columns]
```

K-MEANS

```python
# import thư viện matplotlib để tạo biểu đồ phân tán
import matplotlib.pyplot as plt

# Trích xuất cột 'SALES' từ DataFrame df_clustered và lưu trữ trong biến x_data.
x_data = df_clustered['SALES']

# Trích xuất cột 'DAYS_SINCE_LASTORDER' từ DataFrame df_clustered và lưu trữ trong biến y_data.
y_data = df_clustered['DAYS_SINCE_LASTORDER']

# Trích xuất cột 'CLUSTER' từ DataFrame df_clustered và lưu trữ trong biến cluster_labels.
cluster_labels = df_clustered['CLUSTER']

# Tạo biểu đồ phân tán theo cụm, trong đó: c=cluster_labels sử dụng nhãn cụm (cluster_labels) để tô màu cho các điểm theo cụm,
# s=50 thiết lập kích thước điểm là 50.
plt.scatter(x_data, y_data, c=cluster_labels,s=50)

# Thêm tiêu đề và nhãn cho các trục
plt.title('Biểu đồ phân tán Kmeans')
plt.xlabel('x')
plt.ylabel('y')


# Tính toán tâm cụm
centroids = df_clustered.groupby('CLUSTER').mean()

# Vẽ tâm cụm
for i in range(3):
    plt.plot(centroids.loc[i, 'SALES'], centroids.loc[i, 'DAYS_SINCE_LASTORDER'], '*',  markersize=10,
             markerfacecolor=plt.cm.Spectral(i/3))

# Hiển thị biểu đồ
plt.show()
```


Biểu đồ phân tán Kmeans
1.0
0.8
0.6
0.4
0.2
0.0
y
0.0
0.2
0.4
0.6
0.8
1.0
x

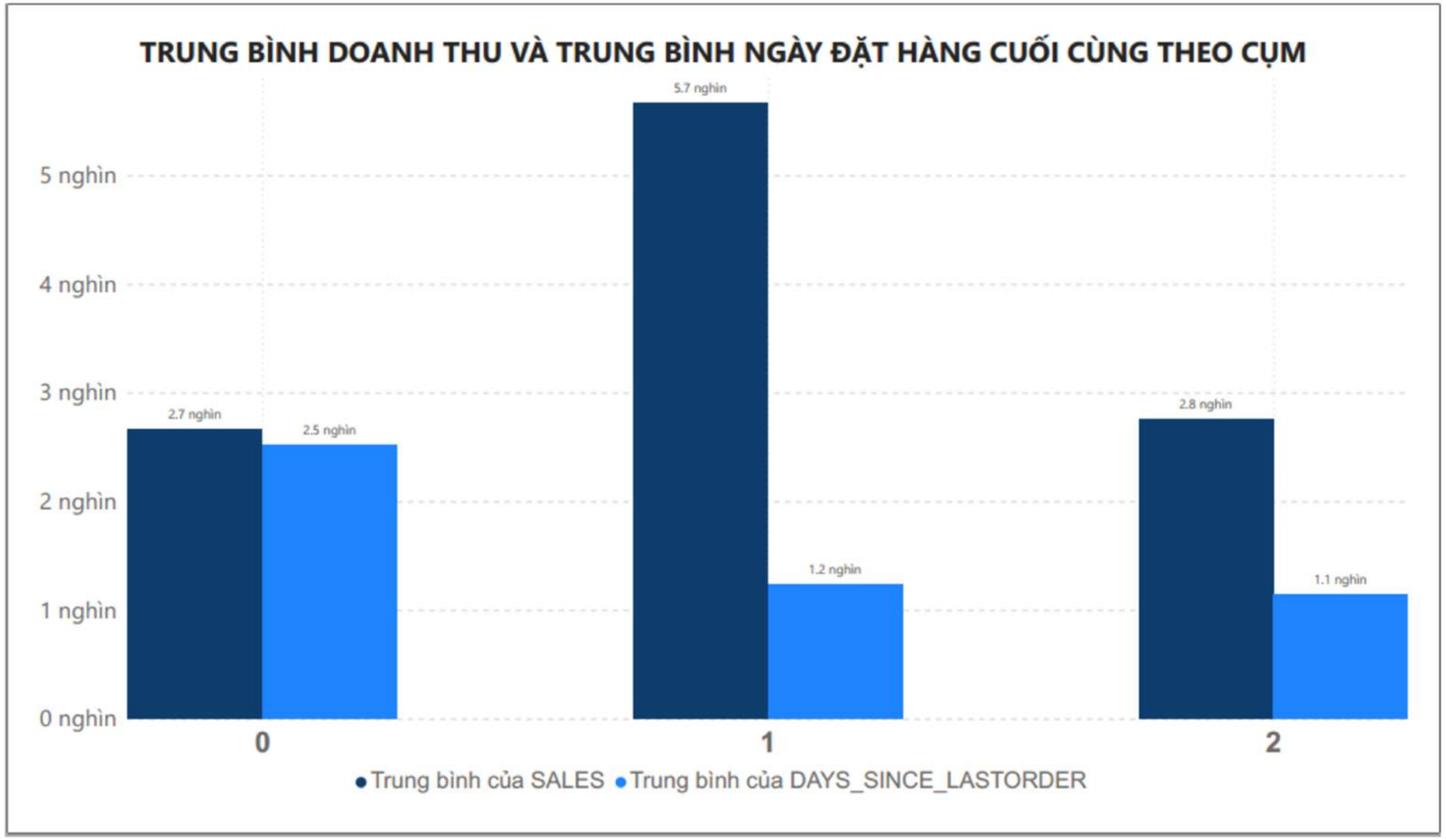
TRỰC QUAN HÓA DỮ LIỆU

Phân khúc khách hàng

Cụm 0: Khách hàng tiềm năng

Cụm 1: Khách hàng VIP

Cụm 2: Khách hàng trung thành



CHIẾN LƯỢC MARKETING

NHÓM KHÁCH HÀNG TIỀM NĂNG

Mục tiêu: Tăng nhận diện thương hiệu, thu hút khách hàng mới

Chiến lược:

- Tiếp thị qua các trang mạng xã hội: Facebook, TikTok,...
- Tổ chức các chương trình khuyến mãi: giảm giá, tặng quà, miễn phí vận chuyển,...

NHÓM KHÁCH HÀNG VIP

Mục tiêu: Duy trì mối quan hệ, tăng tỷ lệ giữ chân khách hàng và khuyến khích họ mua hàng thường xuyên hơn

Chiến lược:

- Chương trình khách hàng thân thiết
- Tổ chức các sự kiện dành riêng cho khách hàng VIP
- Hỗ trợ khách hàng VIP nhanh chóng và hiệu quả

NHÓM KHÁCH HÀNG TRUNG THÀNH

Mục tiêu: Khuyến khích khách hàng trung thành mua hàng thường xuyên hơn

Chiến lược:

- Gửi email marketing: Cung cấp thông tin về sản phẩm mới,...
- Tiếp thị cá nhân hóa
- Tặng voucher, quà cho khách hàng trung thành

Thank You

Nhóm 5

Phân tích dữ liệu bằng Python