

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**MÔN HỌC: KHAI PHÁ DỮ LIỆU**

**BÁO CÁO CUỐI KỲ**

**ĐỀ TÀI: CUSTOMER SEGMENTAITION EXPLORE**

**GVHD: Trần Trọng Bình**

**MÃ HP: DAMI330484\_23\_2\_02**

|  |  |
| --- | --- |
| **SVTH:** | **MSSV** |
| **Lê Hoàng Đức Duy** | **21133016** |
| **Nguyễn Tấn Khang** | **21133105** |
| **Nguyễn Thị Thùy Linh** | **21133051** |
| **Võ Hoài Trọng** | **21133112** |
|  |  |

*TP. Hồ Chí Minh, tháng 05 năm 2024*

**NHẬN XÉT CỦA GIÁO VIÊN**

**ĐIỂM GV KÝ TÊN**

**LỜI CẢM ƠN**

Lời nói đầu tiên, nhóm chúng em xin được gửi đến thầy Trần Trọng Bình – giảng viên bộ môn *Khai phá dữ liệu* lời cảm ơn chân thành và sâu sắc nhất.

Nhóm thực hiện xin cảm ơn sự quan tâm và giúp đỡ tận tình của thầy trong suốt quá trình giảng dạy. Cảm ơn thầy đã luôn giải đáp những thắc mắc cũng như đưa ra những nhận xét, góp ý giúp nhóm thực hiện cải thiện chất lượng công việc của nhóm.

Vì khả năng còn hạn chế nên trong quá trình thực hiện báo cáo không tránh khỏi sai sót, kính mong nhận được những ý kiến đóng góp từ thầy để nhóm có thể cải thiện hơn sau này.

Nhóm thực hiện xin chân thành cảm ơn thầy!

Nhóm 8

[**CHƯƠNG I. TỔNG QUAN 6**](#_Toc168005670)

[**1.1 Mục tiêu kinh doanh 6**](#_Toc168005671)

[**1.2 Mục tiêu học tập 6**](#_Toc168005672)

[**CHƯƠNG II. GIỚI THIỆU TẬP DỮ LIỆU 8**](#_Toc168005673)

[**2.1 Mô tả tập dữ liệu 8**](#_Toc168005674)

[**2.2 Thư viện 9**](#_Toc168005675)

[**2.3 Xem cấu trúc tập dữ liệu 9**](#_Toc168005676)

[**CHƯƠNG III. XỬ LÝ DỮ LIỆU 12**](#_Toc168005677)

[**3.1 Tiền xử lý dữ liệu 12**](#_Toc168005678)

[**3.1.1 Xử lí giá trị NULL 12**](#_Toc168005679)

[**3.1.2 Xử lí giá trị ngoại lệ 14**](#_Toc168005680)

[**3.1.3 Kiểm tra lại tập dữ liệu 19**](#_Toc168005681)

[**3.2 EDA 21**](#_Toc168005682)

[**3.2.1 Phân tích đơn biến 21**](#_Toc168005683)

[**3.2.2 Phân tích đa biến 32**](#_Toc168005684)

[**3.3 Các phương pháp phân tích dữ liệu 43**](#_Toc168005685)

[**3.3.1 Kiểm định thống kê 43**](#_Toc168005686)

[**3.3.2 Phân tích luật kết hợp 46**](#_Toc168005687)

[**3.3.3 Giảm chiều dữ liệu với PCA 53**](#_Toc168005688)

[**CHƯƠNG IV. MÔ HÌNH 57**](#_Toc168005689)

[**4.1 Giải quyết bài toán phân cụm, phân cấp khách hàng 57**](#_Toc168005690)

[**4.1.1 K-mean 57**](#_Toc168005691)

[**4.1.2 DBscan 58**](#_Toc168005692)

[**4.1.3 GMM 59**](#_Toc168005693)

[**4.1.4 Hierarchy 60**](#_Toc168005694)

[**4.2 Phân tích kết quả phân cụm / phân cấp khách hàng 61**](#_Toc168005695)

[**4.3 Giải quyết bài toán phân loại khách hàng dựa trên các phân cụm 63**](#_Toc168005696)

[**4.3.1 Logistic Regression 63**](#_Toc168005697)

[**4.3.2 K-Nearest Neighbors (KNN) 63**](#_Toc168005698)

[**4.3.3 Decision Trees 65**](#_Toc168005699)

[**4.3.4 Random Forest 67**](#_Toc168005700)

[**4.3.5 ANN - Artificial Neural Networks 69**](#_Toc168005701)

[**CHƯƠNG V. ĐÁNH GIÁ VÀ KẾT LUẬN 72**](#_Toc168005702)

[**THAM KHẢO 73**](#_Toc168005703)

# CHƯƠNG I. TỔNG QUAN

## Mục tiêu kinh doanh

- Khai thác các mẫu và thông tin có ý nghĩa từ dữ liệu khách hàng mà không dựa vào các nhãn hoặc danh mục định sẵn.

- Bằng cách sử dụng các kỹ thuật học không giám sát (unsupervised learning),nhằm phân nhóm (Phân cụm / Phân cấp) khách hàng dựa trên sự tương đồng về hành vi, sở thích hoặc đặc điểm của họ mà cụ thể ở đây các giao dịch tín dụng.

- Phân khúc này sau đó có thể được sử dụng cho các mục đích kinh doanh khác nhau như tiếp thị mục tiêu, đề xuất cá nhân hóa, phát triển sản phẩm, hoặc tối ưu hóa dịch vụ.

## Mục tiêu học tập

* **Khám phá một số phương pháp phân cụm, phân cấp khách hàng, bao gồm:**

K-means clustering: Một thuật toán phổ biến để phân chia dữ liệu thành các cụm dựa trên sự tương đồng, trong đó mỗi cụm được đại diện bởi một centroid.

Hierarchical clustering: Một phương pháp xây dựng thứ bậc các cụm bằng cách gộp hoặc tách chúng dựa trên một thước đo khoảng cách.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Một thuật toán nhóm các điểm gần nhau và xác định các điểm nhiễu như là nhiễu.

Gaussian Mixture Models (GMM): Một mô hình xác suất đại diện cho hỗn hợp các phân phối Gaussian, thường được sử dụng để phân cụm dữ liệu.

Principal Component Analysis (PCA): Một kỹ thuật giảm chiều dữ liệu, giúp trực quan hóa và hiểu cấu trúc của dữ liệu có chiều cao.

* **Khám phá một số phương pháp phân loại các khách hàng, bao gồm:**

Logistic Regression Một mô hình phân loại thống kê đơn giản nhưng mạnh mẽ. Logistic Regression được sử dụng để ước lượng xác suất của một biến phụ thuộc nhị phân dựa trên các biến độc lập.

K-Nearest Neighbors (KNN): Một phương pháp phân loại đơn giản dựa trên việc đo khoảng cách giữa các điểm dữ liệu và xác định lớp của một điểm dữ liệu mới dựa trên lớp của các điểm dữ liệu lân cận. KNN không cần huấn luyện mô hình và thường dễ triển khai, nhưng có thể đòi hỏi nhiều tài nguyên tính toán khi kích thước của tập dữ liệu lớn.

Decision Trees: Một mô hình phân loại dựa trên cấu trúc cây quyết định, trong đó mỗi nút trong cây đại diện cho một biến độc lập, mỗi cạnh đại diện cho một quyết định và mỗi lá đại diện cho một lớp hoặc nhãn. Decision Trees có thể dễ dàng diễn giải và hiểu, làm cho chúng thích hợp cho việc khám phá dữ liệu và phân loại.

Random Forest: Là một mô hình ensemble (tổ hợp) được tạo ra từ việc kết hợp nhiều cây quyết định (Decision Trees) khác nhau. Khi cần phân loại một mẫu mới, Random Forest kết hợp kết quả từ tất cả các cây quyết định thành một dự đoán cuối cùng.

Mạng Nơ-ron Nhân Tạo (ANN - Artificial Neural Networks): Một phương pháp phân loại mạnh mẽ trong học máy và học sâu. ANN là một mô hình được lấy cảm hứng từ cấu trúc nơ-ron của não người và bao gồm một số lớp nơ-ron được kết nối với nhau.

# CHƯƠNG II. GIỚI THIỆU TẬP DỮ LIỆU

## Mô tả tập dữ liệu

- Bộ dữ liệu này tóm tắt hành vi sử dụng của khoảng 9000 chủ thẻ tín dụng đang hoạt động trong 6 tháng qua. Mục đích của phân tích là phân chia khách hàng sử dụng thẻ tín dụng thành các nhóm theo hành vi tương tự và cung cấp thông tin chi tiết dựa trên dữ liệu về việc sử dụng thẻ tín dụng giữa các nhóm nhằm phân khúc khách hàng để xác định chiến lược tiếp thị.

Nguồn: <https://www.kaggle.com/datasets/arjunbhasin2013/ccdata>

- Tập dữ liệu có 8950 dòng và 18 cột:

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| CUST\_ID | Nhận dạng chủ thẻ tín dụng (Phân loại) |
| BALANCE | Số dư còn lại trong tài khoản để mua hàng |
| BALANCE\_FREQUENCY | Tần suất cập nhật Số dư, điểm từ 0 đến 1 (1 = cập nhật thường xuyên, 0 = không cập nhật thường xuyên) |
| PURCHASES | Số lượng mua hàng được thực hiện từ tài khoản |
| ONEOFF\_PURCHASES | Số tiền mua tối đa được thực hiện trong một lần |
| INSTALLMENTS\_PURCHASES | Số tiền mua trả góp |
| CASH\_ADVANCE | Tiền mặt người dùng ứng trước |
| PURCHASES\_FREQUENCY | Tần suất mua hàng được thực hiện, điểm từ 0 đến 1 (1 = mua thường xuyên, 0 = không mua thường xuyên) |
| ONEOFF\_PURCHASES\_FREQUENCY | Tần suất mua hàng diễn ra một lần (1 = mua thường xuyên, 0 = không mua thường xuyên) |
| PURCHASES\_INSTALLMENTS\_FREQUENCY | ần suất mua hàng trả góp được thực hiện như thế nào (1 = thực hiện thường xuyên, 0 = không thực hiện thường xuyên) |
| CASH\_ADVANCE\_FREQUENCY | Tần suất thanh toán tiền mặt trước |
| CASH\_ADVANCETRX | Tổng số giao dịch thanh toán ứng trước tiền được thực hiện bằng thẻ tín dụng |
| PURCHASES\_TRX | Số lượng giao dịch mua hàng được thực hiện |
| CREDIT\_LIMIT | Hạn mức thẻ tín dụng cho người dùng |
| PAYMENTS | Số tiền thanh toán được thực hiện bởi người dùng |
| MINIMUM\_PAYMENTS | Số tiền thanh toán tối thiểu do người dùng thực hiện |
| PRC\_FULL\_PAYMENT | Tỷ lệ phần trăm số lần thanh toán toàn bộ của người dùng |
| TENURE | Thời hạn sử dụng dịch vụ thẻ tín dụng của người sử dụng |

## Thư viện

|  |
| --- |
| import os  import pandas as pd  import numpy as np  import seaborn as sns  import math  #plotly has been used for visualization  import matplotlib.pyplot as plt  import plotly as py  from plotly.subplots import make\_subplots  import plotly.graph\_objects as go  from plotly.offline import init\_notebook\_mode, iplot, plot  import plotly.express as px |

## Xem cấu trúc tập dữ liệu

* **Load Data**

|  |
| --- |
| df = pd.read\_csv("./CC GENERAL.csv")  df |

**A screenshot of a computer screen

Description automatically generated**

* **Kích thước**

|  |
| --- |
| df.shape |

* **Cấu trúc**

|  |
| --- |
| df.info() |

**A screenshot of a computer

Description automatically generated**

* **Mô tả**

|  |
| --- |
| df.describe() |

**A screenshot of a white and black screen

Description automatically generated**

Số dư trung bình của khách hàng là $1564 và tần suất cập nhật số dư trung bình là khoảng 0,9. Giá trị trung bình của các giao dịch mua hàng là $1000, trong khi giá trị trung bình của các khoản mua sắm một lần là khoảng $600. Tần suất mua sắm trung bình là khoảng 0,5. Tần suất trung bình của các khoản mua sắm một lần, mua trả góp và tạm ứng tiền mặt nói chung là thấp. Hạn mức tín dụng trung bình của khách hàng là khoảng $4500. Tỷ lệ phần trăm số lần thanh toán toàn bộ của người dùng là 15%. Thời gian trung bình khách hàng sử dụng dịch vụ là 11 năm.

# CHƯƠNG III. XỬ LÝ DỮ LIỆU

## Tiền xử lý dữ liệu

### 3.1.1 Xử lí giá trị NULL

* **Kiểm tra NULL**

|  |
| --- |
| df.isnull().sum() |

**A screenshot of a computer screen

Description automatically generated**

* **Tỷ lệ phần trăm giá trị thiếu trong từng cột**

|  |
| --- |
| percentage\_missing = round(df.isnull().sum() \* 100 / len(df),4)  missing\_value\_df = pd.DataFrame({'Missing\_Percentage' : percentage\_missing})  missing\_value\_df.sort\_values(by='Missing\_Percentage', ascending = False).head() |

**A screenshot of a missing person

Description automatically generated**

* **Loại bỏ các hàng có giá trị thiếu trong cột CREDIT\_LIMIT**

(Do có tỉ lệ NULL ít)

|  |
| --- |
| df.dropna(subset=['CREDIT\_LIMIT'], inplace=True, axis=0) |

* **Điền giá trị thiếu trong cột MINIMUM\_PAYMENTS bằng giá trị trung vị**

- Ta cần kiểm tra mối quan hệ giữa MINIMUM\_PAYMENTS và PAYMENTS (Vì thông thường MINIMUM\_PAYMENTS <= PAYMENTS)

|  |
| --- |
| number = df[(df['MINIMUM\_PAYMENTS'] > df['PAYMENTS']) & (df['MINIMUM\_PAYMENTS'].notna())]  print(f"Số dòng MINIMUM\_PAYMENTS > PAYMENTS (Không tính giá trị thiếu): {number.shape[0]} dòng!") |

- Nhận xét về kết quả trên một số giải thích có thể cho hiện tượng này:

* Khách hàng Không Thanh Toán Đủ Số Tiền Tối Thiểu

Một lý do phổ biến là khách hàng đã không thanh toán đủ số tiền tối thiểu yêu cầu. Trong trường hợp này, số tiền thanh toán (PAYMENTS) có thể nhỏ hơn số tiền thanh toán tối thiểu (MINIMUM\_PAYMENTS).

* Lỗi Nhập Dữ liệu hoặc Xử lý Dữ liệu

Có thể có lỗi nhập liệu hoặc lỗi trong quá trình xử lý dữ liệu, dẫn đến tình trạng MINIMUM\_PAYMENTS lớn hơn PAYMENTS. Điều này có thể xảy ra do sai sót trong việc nhập liệu ban đầu hoặc trong các bước xử lý dữ liệu.

* Chính Sách Tài Chính Của Công Ty

Các tổ chức tài chính đôi khi có thể có các chính sách phức tạp về việc thanh toán và lãi suất. Ví dụ, họ có thể tính lãi suất bổ sung hoặc các khoản phí khác mà khách hàng không thanh toán đủ số tiền tối thiểu.

* Thời Gian và Kỳ Thanh Toán Khác Nhau

Cũng có thể các số liệu này được ghi nhận ở các thời điểm khác nhau hoặc cho các kỳ thanh toán khác nhau. Số tiền thanh toán có thể là cho một kỳ trước, trong khi số tiền thanh toán tối thiểu có thể là yêu cầu cho kỳ sau.

- Ta sẽ xử lý giá trị thiếu cột MINIMUM\_PAYMENTS như sau:

* Với PAYMENTS = 0 thì MINIMUM\_PAYMENTS = 0
* Với 0 < PAYMENTS < mean(PAYMENTS) thì MINIMUM\_PAYMENTS = PAYMENTS
* Với PAYMENTS > mean(PAYMENTS) thì MINIMUM\_PAYMENTS = mean(PAYMENTS)

A close up of words

Description automatically generated

* **Loại bỏ cột CUST\_ID**

|  |
| --- |
| df.drop(columns=['CUST\_ID'],axis=1,inplace=True) |

### 3.1.2 Xử lí giá trị ngoại lệ

* **Trực quan hóa các ngoại lệ**

**A collage of blue squares

Description automatically generated**

Nhìn vào các biểu đồ Boxplot trên, chỉ có PURCHASES\_FREQUENCY và PURCHASES\_FREQUENCY là không chứa ngoại lệ.

* **Số lượng ngoại lệ trong mỗi cột**

**A screenshot of a computer

Description automatically generated**

Dựa vào kiểu thống kê này, dễ dàng thấy được ngoại lệ giữa các cột là bao nhiêu, từ đó đưa ra biện pháp loại bỏ.

* **Biểu đồ so sánh ngoại lệ giữa các cột**

**A graph with pink bars

Description automatically generated with medium confidence**

Dựa vào biểu đồ cột, ta thấy có 2 cột không có ngoại lai, cột BALANCE\_FREQUENCY, PRC\_FULL\_PAYMENT, TENURE có mức ngoại lai khá cao so với tổng thể.

* **Bỏ qua các giá trị ngoại lệ**

|  |
| --- |
| df = df[(df['BALANCE']<15000)]  df = df[(df['PURCHASES']<40000)]  df = df[(df['ONEOFF\_PURCHASES']<30000)]  df = df[(df['INSTALLMENTS\_PURCHASES']<20000)]  df = df[(df['CASH\_ADVANCE']<40000)]  df = df[(df['MINIMUM\_PAYMENTS']<60000)] |

**A group of blue squares with black numbers

Description automatically generated with medium confidence**

- Ta nhận thấy có rất nhiều giá trị ngoại lai trên các cột. Tuy nhiên việc ta loại bỏ hoàn toàn chúng thì là điều không được khuyến khích hoặc không thể:

* Thông tin giá trị: Các giá trị ngoại lai có thể chứa thông tin quan trọng về các mẫu hoặc trường hợp đặc biệt mà không nên bỏ qua.
* Mất dữ liệu: Loại bỏ quá nhiều giá trị có thể làm giảm kích thước của tập dữ liệu, dẫn đến mất mát thông tin và làm giảm độ tin cậy của các phân tích thống kê.
* Bóp méo kết quả: Nếu các giá trị ngoại lai có nguyên nhân xác đáng, việc loại bỏ chúng có thể dẫn đến kết quả sai lệch hoặc không phản ánh đúng thực tế.
* Sinh ra các giá trị sai sót khác

- Đề xuất:

1. Biến đổi dữ liệu:

* Log Transformation: Chuyển đổi dữ liệu bằng cách lấy logarit của các giá trị để làm giảm sự ảnh hưởng của các giá trị ngoại lai lớn.
* Z-score Normalization: Sử dụng chuẩn hóa Z-score để xác định và điều chỉnh các giá trị ngoại lai.

2. Chỉ loại bỏ các giá trị ngoại lai cực đoan

3. Sử dụng mô hình robust:

Một số mô hình học máy có khả năng xử lý tốt các giá trị ngoại lai mà không cần phải loại bỏ chúng. Ví dụ: Random Forest, Gradient Boosting.

4. Phân tích sâu hơn:

Thực hiện phân tích chi tiết để xác định nguyên nhân của các giá trị ngoại lai và xem liệu chúng có thể được giải thích bởi các yếu tố khác trong dữ liệu hay không.

- Lựa chọn của nhóm:

Nhóm sẽ loại bỏ các giá trị ngoại lai cực đoan và song song với đó sử dụng các cách biến đổi dữ liệu, và các mô hình phù hợp để giải quyết vấn đề ngoại lai này!

### 3.1.3 Kiểm tra lại tập dữ liệu

* **Kiểm tra NULL lại**

**A list of items with text

Description automatically generated with medium confidence**

* **Kiểm tra xem có hàng trùng lặp không**

|  |
| --- |
| df.duplicated().any() |

* **Xem lại data sau khi xử lí**

**A screenshot of a computer

Description automatically generated**

Sau khi xử lí xong, tập dữ liệu còn 8931 dòng và 17 cột.

## EDA

### Phân tích đơn biến

* **Biểu đồ phân bố mật độ xác suất (KDE)**

**A graph of energy and energy

Description automatically generated with medium confidence**

Area Under the cureve of BALANCE between 0.0 and 3645.4: 0.7156

Area Under the cureve of BALANCE between 3645.4 and 7290.7: 0.0983

Area Under the cureve of BALANCE between 7290.7 and 10936.1: 0.0216

Area Under the cureve of BALANCE between 10936.1 and 14581.5 mAh: 0.004

A graph of energy consumption

Description automatically generated

Area Under the cureve of PURCHASES between 0.0 and 8134.9: 0.7029

Area Under the cureve of PURCHASES between 8134.9 and 16269.9: 0.0099

Area Under the cureve of PURCHASES between 16269.9 and 24404.8: 0.0009

Area Under the cureve of PURCHASES between 24404.8 and 32539.8 mAh: 0.0009

A graph of a graph showing the amount of energy

Description automatically generated

Area Under the cureve of ONEOFF\_PURCHASES between 0.0 and 6636.9: 0.6884

Area Under the cureve of ONEOFF\_PURCHASES between 6636.9 and 13273.7: 0.0072

Area Under the cureve of ONEOFF\_PURCHASES between 13273.7 and 19910.6: 0.0009

Area Under the cureve of ONEOFF\_PURCHASES between 19910.6 and 26547.4 mAh: 0.0008

A graph of energy consumption

Description automatically generated

Area Under the cureve of INSTALLMENTS\_PURCHASES between 0.0 and 3671.5: 0.6337

Area Under the cureve of INSTALLMENTS\_PURCHASES between 3671.5 and 7343.0: 0.0085

Area Under the cureve of INSTALLMENTS\_PURCHASES between 7343.0 and 11014.6: 0.001

Area Under the cureve of INSTALLMENTS\_PURCHASES between 11014.6 and 14686.1 mAh: 0.0008

A graph of a graph showing the value of a money

Description automatically generated with medium confidence

Area Under the cureve of CASH\_ADVANCE between 0.0 and 7320.5: 0.5979

Area Under the cureve of CASH\_ADVANCE between 7320.5 and 14641.1: 0.0171

Area Under the cureve of CASH\_ADVANCE between 14641.1 and 21961.6: 0.002

Area Under the cureve of CASH\_ADVANCE between 21961.6 and 29282.1 mAh: 0.0006

A graph of a graph

Description automatically generated

Area Under the cureve of CASH\_ADVANCE\_TRX between 0 and 30.8: 0.5744

Area Under the cureve of CASH\_ADVANCE\_TRX between 30.8 and 61.5: 0.007

Area Under the cureve of CASH\_ADVANCE\_TRX between 61.5 and 92.2: 0.0007

Area Under the cureve of CASH\_ADVANCE\_TRX between 92.2 and 123 mAh: 0.0004

A graph of energy

Description automatically generated

Area Under the cureve of PURCHASES\_TRX between 0 and 86.8: 0.7405

Area Under the cureve of PURCHASES\_TRX between 86.8 and 173.5: 0.0179

Area Under the cureve of PURCHASES\_TRX between 173.5 and 260.2: 0.0025

Area Under the cureve of PURCHASES\_TRX between 260.2 and 347 mAh: 0.0006

A graph of energy efficiency

Description automatically generated

Area Under the cureve of CREDIT\_LIMIT between 50.0 and 7537.5: 0.7979

Area Under the cureve of CREDIT\_LIMIT between 7537.5 and 15025.0: 0.1595

Area Under the cureve of CREDIT\_LIMIT between 15025.0 and 22512.5: 0.0164

Area Under the cureve of CREDIT\_LIMIT between 22512.5 and 30000.0 mAh: 0.0004

A graph of energy

Description automatically generated

Area Under the cureve of PAYMENTS between 0.0 and 12680.4: 0.8744

Area Under the cureve of PAYMENTS between 12680.4 and 25360.7: 0.0081

Area Under the cureve of PAYMENTS between 25360.7 and 38041.1: 0.0016

Area Under the cureve of PAYMENTS between 38041.1 and 50721.5 mAh: 0.0003

A graph of energy

Description automatically generated

Area Under the cureve of MINIMUM\_PAYMENTS between 0.0 and 14092.5: 0.6929

Area Under the cureve of MINIMUM\_PAYMENTS between 14092.5 and 28185.0: 0.0032

Area Under the cureve of MINIMUM\_PAYMENTS between 28185.0 and 42277.5: 0.0005

Area Under the cureve of MINIMUM\_PAYMENTS between 42277.5 and 56370.0 mAh: 0.0003

A graph of energy

Description automatically generated

Area Under the cureve of TENURE between 6 and 7.5: 0.0321

Area Under the cureve of TENURE between 7.5 and 9.0: 0.0309

Area Under the cureve of TENURE between 9.0 and 10.5: 0.0356

Area Under the cureve of TENURE between 10.5 and 12 mAh: 0.4489

**Nhận xét:**

Các biểu đồ trên thể hiện sự phân phối tập trung nằm ở các khoảng đã được chia và thể hiện rõ các mức độ phổ biến của các giá trị. Sau đó thực hiện tính toán AUC (diện tích dưới đường cong) để xem rõ hơn việc phân bổ các giá trị.

* **Biểu đồ Histogram**

**A group of graphs showing the same size

Description automatically generated with medium confidence**

**Nhận xét:**

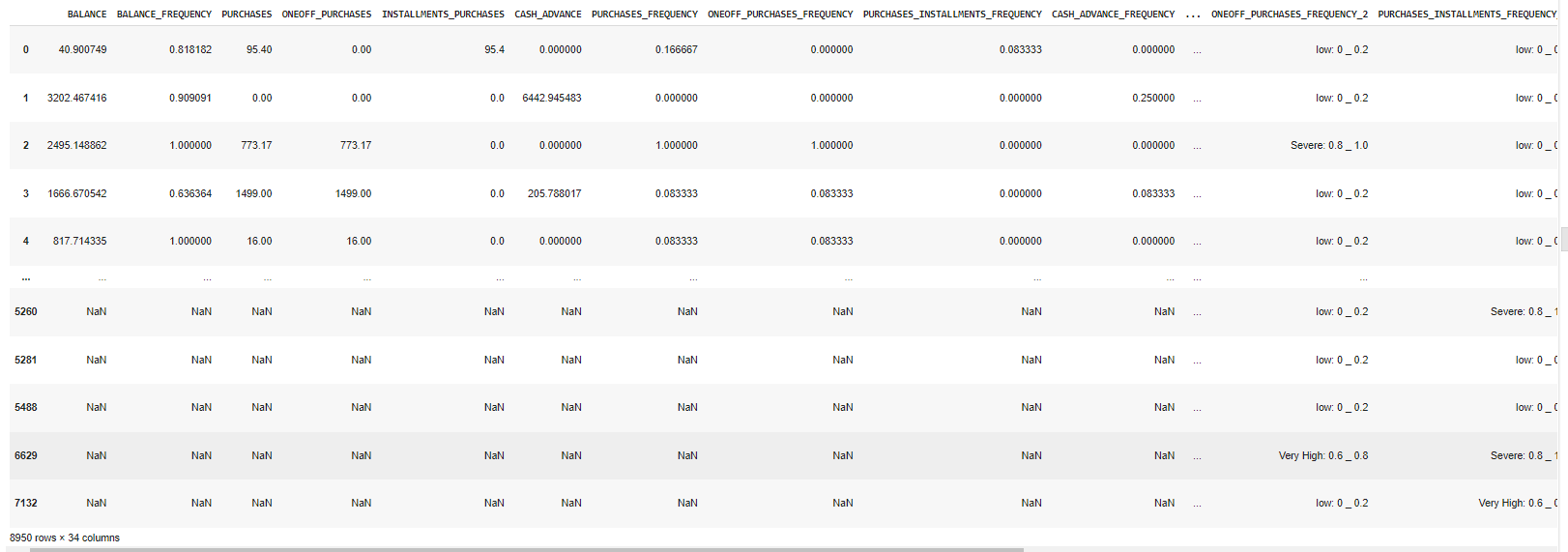
### Phân tích đa biến

* Tiến hành phân loại giá trị từng cột của data ban đầu thành các nhóm (low, moderate, high, very high, severe) dựa trên khoảng giá trị của chúng. Sau đó sẽ lưu trữ các nhãn phân loại này thành 'df2'. Cuối cùng sẽ kết hợp với data ban đầu để tạo thành 'df3' có chứa cả giá trị gốc và nhãn phân loại.

A screenshot of a computer

Description automatically generated

* Kết hợp hai DataFrames



* Biểu đồ

A screenshot of a graph

Description automatically generated

A red and grey bars

Description automatically generated

A graph of red and grey bars

Description automatically generated

A screenshot of a graph

Description automatically generated

A graph of different colored squares

Description automatically generated with medium confidence

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A graph of cash advance frequency

Description automatically generated

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

A graph of red and grey bars

Description automatically generated

A graph of credit limit

Description automatically generated with medium confidence

A red and grey striped chart

Description automatically generated with medium confidence

A graph of a diagram

Description automatically generated with medium confidence

Ta có thể nhìn vào bảng dưới đây để xem rõ ràng số liệu hơn

A table with numbers and symbols

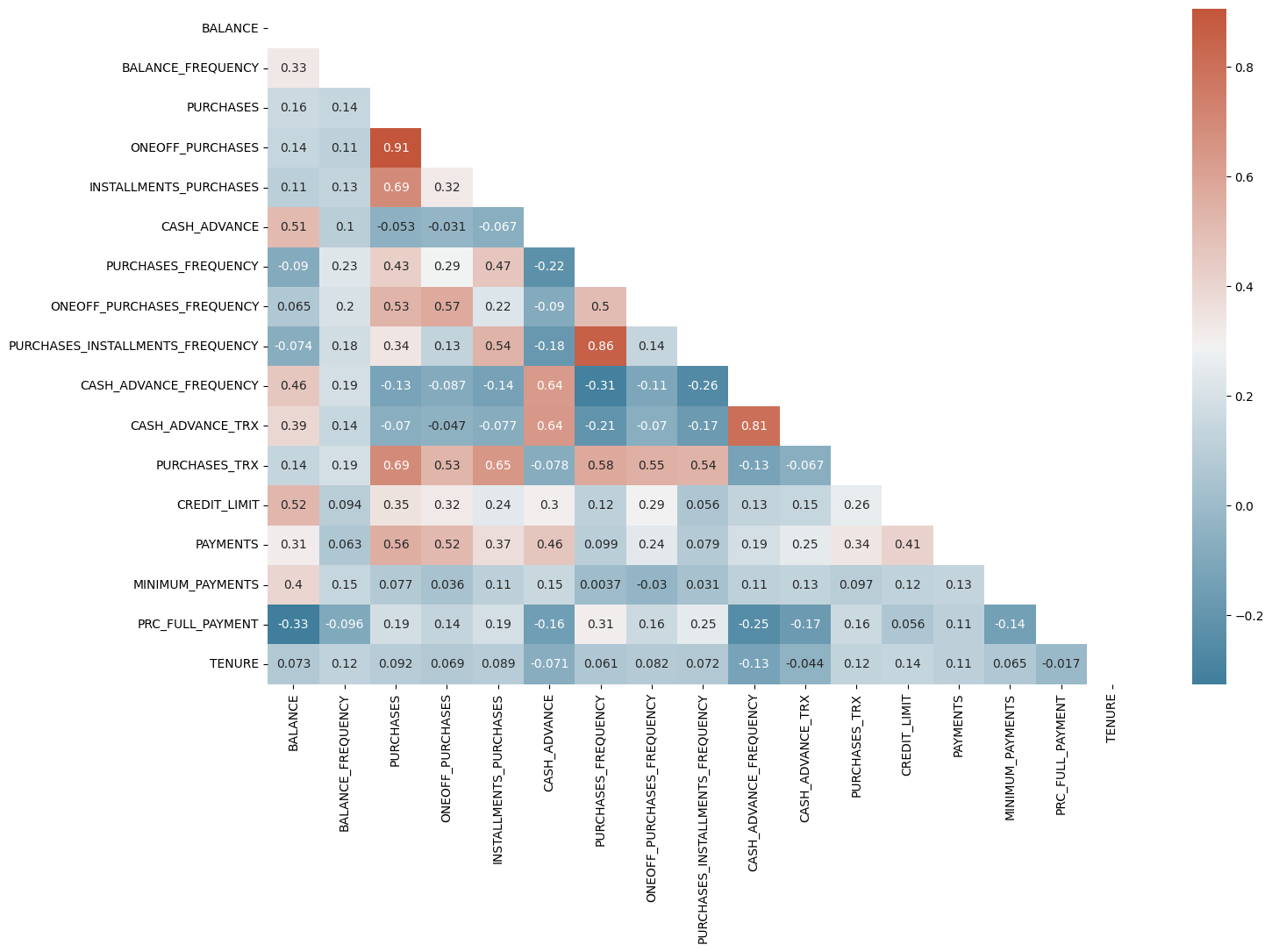
Description automatically generated

* Tổng kết lại các giá trị

A screenshot of a graph

Description automatically generated

* **Biểu đồ tương quan (heatmap)**



**Nhận xét:**

* Tương quan dương cao:

- ONEOFF\_PURCHASES và PURCHASES (0.91): Điều này chỉ ra rằng có một tương quan dương mạnh mẽ, gợi ý rằng khi các giao dịch mua một lần tăng lên, tổng số giao dịch mua cũng tăng theo.

- PURCHASES\_INSTALLMENTS\_FREQUENCY và PURCHASES\_FREQUENCY (0.86) : Có một tương quan mạnh giữa tần suất các giao dịch mua trả góp và tần suất các giao dịch mua, cho thấy rằng khi tần suất các giao dịch mua trả góp tăng, tần suất các giao dịch mua cũng tăng.

* Tương Quan Âm Cao:

- PRC\_FULL\_PAYMENT và BALANCE (-0.33): Có một tương quan âm đáng kể, nghĩa là khi phần trăm thanh toán đầy đủ tăng lên, số dư tài khoản giảm.

- PRC\_FULL\_PAYMENT và BALANCE\_FREQUENCY (-0.096): Một tương quan âm nhỏ hơn, nhưng vẫn cho thấy rằng khi tần suất cân đối tài khoản tăng, phần trăm thanh toán đầy đủ giảm.

* Tương Quan Trung Bình:

- BALANCE và CASH\_ADVANCE (0.51): Có một tương quan dương trung bình, gợi ý rằng khi số dư tài khoản tăng, các giao dịch ứng tiền mặt cũng tăng.

- PAYMENTS và PURCHASES (0.52): Tương quan dương này cho thấy rằng khi số lượng thanh toán tăng, tổng số giao dịch mua cũng tăng.

* Tương Quan Yếu hoặc Không Đáng Kể:

- TENURE và các đặc trưng khác: Đa số các tương quan giữa TENURE và các đặc trưng khác là yếu hoặc không đáng kể, với các giá trị dao động quanh 0

**Kết luận:**

- Tương quan giữa tần suất giao dịch mua và loại giao dịch: Các đặc trưng liên quan đến tần suất giao dịch mua và loại giao dịch thường có mối tương quan cao với nhau. Điều này ngụ ý rằng mức độ hoạt động mua sắm của khách hàng thường ảnh hưởng đến loại hình giao dịch mà họ thực hiện.

- Tương quan âm giữa PRC\_FULL\_PAYMENT và số dư tài khoản: Sự tương quan âm giữa PRC\_FULL\_PAYMENT và các đặc trưng liên quan đến số dư tài khoản cho thấy rằng người dùng có xu hướng thanh toán toàn bộ số dư khi số dư tài khoản của họ ít hơn. Điều này có thể cho thấy một xu hướng của người dùng muốn trả nợ hoặc quản lý tài chính cẩn thận khi số dư tài khoản giảm.

- Mối quan hệ trung bình giữa ứng tiền mặt và số dư tài khoản: Các đặc trưng liên quan đến ứng tiền mặt và số dư tài khoản thường có mối quan hệ trung bình, cho thấy một mối liên hệ giữa việc ứng tiền mặt và số dư tài khoản. Điều này có thể gợi ý rằng một số khách hàng có thể ứng tiền mặt từ số dư tài khoản của họ để thực hiện các giao dịch cụ thể.

## Các phương pháp phân tích dữ liệu

### Kiểm định thống kê

Sau khi phân tích về những phân phối, những mối quan hệ đơn biến cũng như là đa biến ta có một số giả thuyết cần được chứng minh:

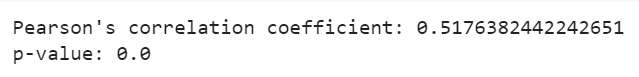
* **Câu hỏi:**

Liệu những khách hàng sử dụng thẻ theo mục đích riêng và nhu cầu của họ? Tức là số dư trong tài khoản của họ có mối tương quan với hạn mức thẻ tín dụng hay không? Hay họ chỉ mở thẻ vì một lí do nào đó ?

- Giả thuyết

H0: Không có mối tương quan giữa số dư còn lại trong tài khoản (BALANCE) và hạn mức thẻ tín dụng (CREDIT\_LIMIT) (hệ số tương quan = 0).

Ha: Có mối tương quan giữa số dư còn lại trong tài khoản (BALANCE) và hạn mức thẻ tín dụng (CREDIT\_LIMIT) (hệ số tương quan ≠ 0).



Ta thấy p-value gần bằng 0 chứng tỏ ta hoàn toàn bác bỏ được giả thuyết H0 để chấp nhận giả thuyết Ha rằng Có mối tương quan giữa số dư còn lại trong tài khoản (BALANCE) và hạn mức thẻ tín dụng (CREDIT\_LIMIT)

Chứng tỏ rằng khách hàng sử dụng thẻ theo mục đích riêng và nhu cầu riêng của họ.

* **Câu hỏi:**

Khi khách hàng sử dụng thẻ để mua hàng, liệu có sự khác biệt về số tiền mua giữa mua hàng một lần và mua hàng trả góp? Họ sẽ trả tiền một lần hay trả góp dựa vào số tiền họ có phải hay không ?

- Giả thuyết

H0 : Không có sự khác biệt về số tiền mua giữa mua hàng một lần (ONEOFF\_PURCHASES) và mua hàng trả góp (INSTALLMENTS\_PURCHASES).

Ha : Có sự khác biệt về số tiền mua giữa mua hàng một lần (ONEOFF\_PURCHASES) và mua hàng trả góp (INSTALLMENTS\_PURCHASES).

A number and a number

Description automatically generated with medium confidence

Ta thấy p-value gần bằng 0 chứng tỏ ta hoàn toàn bác bỏ được giả thuyết H0 để chấp nhận giả thuyết Ha rằng có sự khác biệt về số tiền mua giữa mua hàng một lần (ONEOFF\_PURCHASES) và mua hàng trả góp (INSTALLMENTS\_PURCHASES).

Cho thấy rằng khách hàng có xu hướng chi tiêu khác nhau khi mua hàng một lần so với khi mua hàng trả góp. Cụ thể, số tiền mua hàng một lần khác biệt đáng kể so với số tiền mua hàng trả góp.

* **Câu hỏi:**

Khi khách hàng sử dụng thẻ tín dụng, liệu có sự khác biệt về tần suất giữa việc mua hàng và ứng trước tiền mặt? Tức là việc mua hàng nhiều có đi cùng với việc ứng trước tiền mặt (trả trước) nhiều hay ngược lại hay không theo nhu cầu của khách hàng dùng thẻ ?

- Giả thuyết

H0 : Không có sự khác biệt về tần suất giữa mua hàng (PURCHASES\_FREQUENCY) và ứng trước tiền mặt (CASH\_ADVANCE\_FREQUENCY).

Ha: Có sự khác biệt về tần suất giữa mua hàng (PURCHASES\_FREQUENCY) và ứng trước tiền mặt (CASH\_ADVANCE\_FREQUENCY).

A black text on a white background

Description automatically generated

Ta thấy p-value bằng 0 chứng tỏ ta hoàn toàn bác bỏ được giả thuyết H0 để chấp nhận giả thuyết Ha  rằng có sự khác biệt về tần suất giữa mua hàng (PURCHASES\_FREQUENCY) và ứng trước tiền mặt (CASH\_ADVANCE\_FREQUENCY).

Cho thấy rằng tần suất mua hàng và tần suất ứng trước tiền mặt của khách hàng có sự khác biệt đáng kể. Điều này có thể gợi ý rằng khách hàng sử dụng thẻ tín dụng theo các mục đích khác nhau (mua hàng so với ứng trước tiền mặt) và có các hành vi tiêu dùng khác nhau cho mỗi loại giao dịch này.

### Phân tích luật kết hợp

* **Phân khúc dữ liệu:**

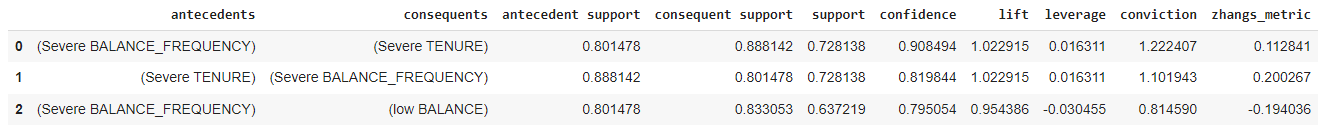
|  |
| --- |
| df2=pd.DataFrame()  column\_name={}  for i in df.columns:  column\_name= []  x= df[i].max() - df[i].min()  c= round(x / 5, 2)  d= df[i].min()  for j in df[i]:  if j <= d+c:  column\_name.append( f"low {i}" )  if (d+c)< j <= d+(2\*c) :  column\_name.append(f"Moderate {i}" )  if (d+2\*c) < j <= d+3\*c:  column\_name.append(f"High {i}" )  if d+3\*c < j <= d+4\*c:  column\_name.append(f"Very High {i}")  if d+4\*c < j <= 1+d+5\*c:  column\_name.append(f"Severe {i}")  se = pd.Series(column\_name)  df2[i]= se.values  df2 = df2.add\_suffix('\_2')  df2.head(3) |

A group of text boxes

Description automatically generated

* **Dùng thuật toán apriori để sinh ra các Rules:**

|  |
| --- |
| # Chuyển đổi dữ liệu sang dạng giao dịch  transactions = df2.astype(str).values.tolist()  # Sử dụng TransactionEncoder  te = TransactionEncoder()  te\_ary = te.fit(transactions).transform(transactions)  df\_trans = pd.DataFrame(te\_ary, columns=te.columns\_)  # Áp dụng thuật toán Apriori  frequent\_itemsets = apriori(df\_trans, use\_colnames=True)  # Khai phá luật kết hợp  rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=0.5)  rules.head(3) |



* **Quy trình Mua sắm và Thanh toán:**

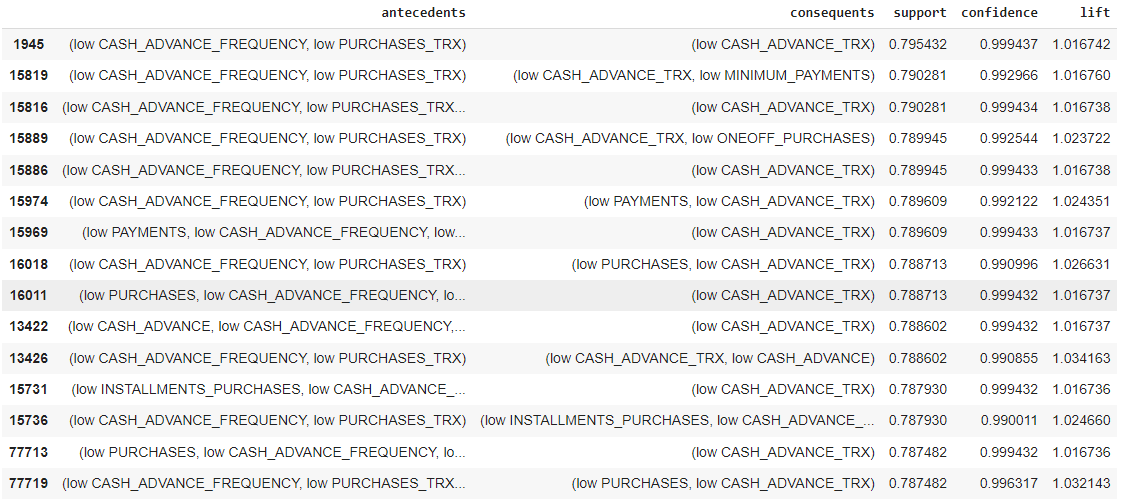
Nghi ngờ: Người dùng có xu hướng mua hàng trả góp sau khi đã thực hiện mua hàng trả hết một lần không?

|  |
| --- |
| # Lọc các luật có liên quan đến PURCHASES\_FREQUENCY và ONEOFF\_PURCHASES\_FREQUENCY  purchase\_rules = rules[rules['antecedents'].astype(str).str.contains('PURCHASES\_FREQUENCY') & rules['antecedents'].astype(str).str.contains('ONEOFF\_PURCHASES\_FREQUENCY')]  # Lọc tiếp các luật có consequents là PURCHASES\_INSTALLMENTS\_FREQUENCY  purchase\_installment\_rules = purchase\_rules[purchase\_rules['consequents'].astype(str).str.contains('PURCHASES\_INSTALLMENTS\_FREQUENCY')]  # Sắp xếp theo support giảm dần  purchase\_installment\_rules\_sorted = purchase\_installment\_rules.sort\_values(by='support', ascending=False)  purchase\_installment\_rules\_sorted |

* Ta thấy không có một luật nào được tìm thấy cho ta rút ra được không có mối quan hệ rõ ràng: Không tìm thấy bằng chứng thống kê nào cho thấy người dùng có xu hướng mua hàng trả góp sau khi đã thực hiện mua hàng một lần.
* Hoặc chỉ đơn giản là cũng có khả năng dữ liệu không đủ lớn hoặc không có thông tin đầy đủ để phát hiện ra các mối quan hệ như đã nghi ngờ. Việc thu thập thêm dữ liệu hoặc cải thiện chất lượng dữ liệu có thể cần thiết để kiểm tra lại giả thuyết này trong tương lai.
* **Quy trình ứng tiền trước:**

Nghi ngờ: Người dùng thường mua hàng bằng tiền ứng trước sau khi thực hiện giao dịch mua hàng không?

|  |
| --- |
| cash\_advance\_rules = rules[rules['antecedents'].astype(str).str.contains('CASH\_ADVANCE\_FREQUENCY') & rules['antecedents'].astype(str).str.contains('PURCHASES\_TRX')]  # Lọc tiếp các luật có consequents là CASH\_ADVANCETRX  cash\_advancetrx\_rules = cash\_advance\_rules[cash\_advance\_rules['consequents'].astype(str).str.contains('CASH\_ADVANCE\_TRX')]  # Sắp xếp theo support giảm dần  cash\_advancetrx\_rules\_sorted = cash\_advancetrx\_rules.sort\_values(by='support', ascending=False)  cash\_advancetrx\_rules\_sorted.head(15) |



* Ta nhận thấy có một mối quan hệ rõ rệt trong các luật kết hợp:
  + Các luật kết hợp cho thấy khi tần suất giao dịch mua hàng thấp (PURCHASES\_TRX) và tần suất ứng trước tiền mặt thấp (CASH\_ADVANCE\_FREQUENCY), thì số lượng giao dịch ứng trước tiền mặt (CASH\_ADVANCETRX) cũng có xu hướng thấp.
  + Điều này được thể hiện qua các luật có độ tin cậy rất cao (gần 1) và giá trị hỗ trợ tương đối cao, cho thấy rằng đây là một xu hướng phổ biến trong dữ liệu.
* Ý nghĩa thực tiễn:
  + Các luật này gợi ý rằng khách hàng có tần suất mua hàng thấp và ít sử dụng ứng trước tiền thì cũng sẽ ít có số lượng giao dịch ứng trước tiền mặt.
  + Các chính sách hoặc chiến lược tiếp thị có thể nhắm vào nhóm khách hàng này để khuyến khích sử dụng thêm các dịch vụ khác, nếu mục tiêu là tăng cường sự tương tác và sử dụng thẻ tín dụng.
* **Quy trình Thanh toán và Hạn mức tín dụng:**

Nghi ngờ: Người dùng có xu hướng thanh toán ít hơn số tiền tối thiểu yêu cầu khi hạn mức tín dụng thấp không?

|  |
| --- |
| # Lọc các luật có liên quan đến CREDIT\_LIMIT và MINIMUM\_PAYMENTS  credit\_limit\_rules = rules[rules['antecedents'].astype(str).str.contains('CREDIT\_LIMIT') & rules['antecedents'].astype(str).str.contains('MINIMUM\_PAYMENTS')]  # Lọc tiếp các luật có consequents là PRC\_FULL\_PAYMENT  prc\_full\_payment\_rules = credit\_limit\_rules[credit\_limit\_rules['consequents'].astype(str).str.contains('PRC\_FULL\_PAYMENT')]  # Sắp xếp theo support giảm dần  prc\_full\_payment\_rules\_sorted = prc\_full\_payment\_rules.sort\_values(by='support', ascending=False)  prc\_full\_payment\_rules\_sorted.head(15) |

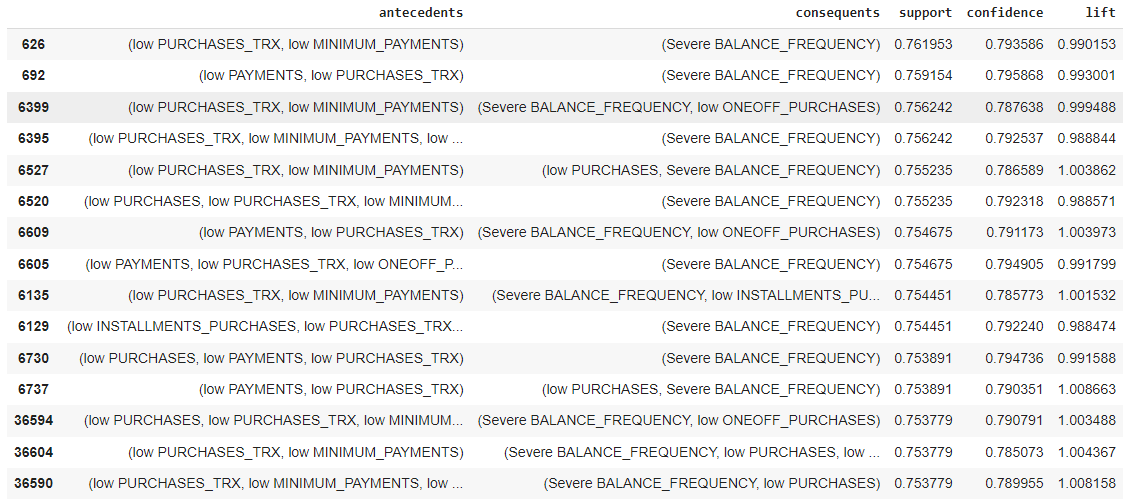
A screenshot of a computer

Description automatically generated

* Ta nhận thấy có một mối quan hệ rõ rệt trong các luật kết hợp:
  + Các luật kết hợp cho thấy khi hạn mức tín dụng (CREDIT\_LIMIT) và số tiền thanh toán tối thiểu (MINIMUM\_PAYMENTS) thấp, tỷ lệ thanh toán đầy đủ (PRC\_FULL\_PAYMENT) cũng có xu hướng thấp.
  + Các luật này có độ tin cậy (confidence) dao động quanh mức 0.79, cho thấy rằng có một mối quan hệ khá mạnh giữa các yếu tố này.
  + Các luật có giá trị hỗ trợ (support) khoảng 0.588, nghĩa là khoảng 58.8% của dữ liệu tuân theo các luật này. Đây là một tỷ lệ khá cao, cho thấy rằng những luật này thường xuyên xuất hiện trong dữ liệu.
  + Các giá trị lift quanh mức 1.01, cho thấy rằng mặc dù có mối quan hệ, nhưng sự tăng lên trong khả năng xuất hiện của các kết quả (consequents) khi các điều kiện (antecedents) thỏa mãn không phải là quá lớn. Điều này chỉ ra rằng mặc dù có mối quan hệ, nhưng không phải là một sự phụ thuộc mạnh mẽ.
* Ý nghĩa thực tiễn:
* Kết quả này gợi ý rằng khách hàng có hạn mức tín dụng thấp và thanh toán tối thiểu thấp có xu hướng không thanh toán đầy đủ hóa đơn. Đây có thể là một nhóm khách hàng có nguy cơ cao hơn về khả năng không trả nợ đúng hạn.
* Các chiến lược quản lý rủi ro hoặc các chương trình khuyến khích thanh toán đầy đủ có thể được tập trung vào nhóm khách hàng này để cải thiện tỷ lệ thanh toán và giảm thiểu rủi ro.
* **Quy trình tương tác với dịch vụ:**

Nghi ngờ: Người dùng có xu hướng tăng cường tương tác với dịch vụ sau khi mua hàng nhiều lần không?

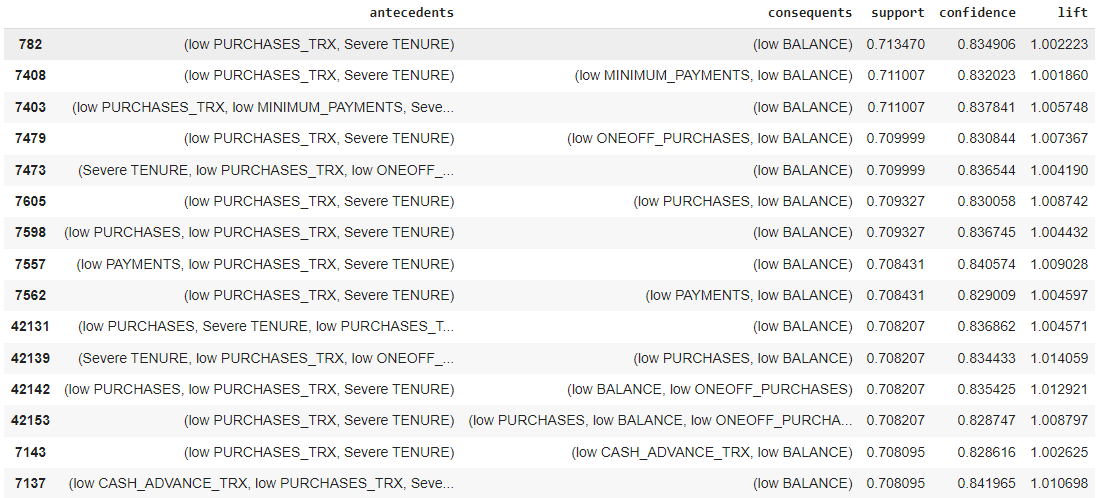
|  |
| --- |
| # Lọc các luật có liên quan đến PURCHASES\_TRX và PAYMENTS  interaction\_rules = rules[rules['antecedents'].astype(str).str.contains('PURCHASES\_TRX') & rules['antecedents'].astype(str).str.contains('PAYMENTS')]  # Lọc tiếp các luật có consequents là BALANCE\_FREQUENCY  balance\_frequency\_rules = interaction\_rules[interaction\_rules['consequents'].astype(str).str.contains('BALANCE\_FREQUENCY')]  # Sắp xếp theo support giảm dần  balance\_frequency\_rules\_sorted = balance\_frequency\_rules.sort\_values(by='support', ascending=False)  balance\_frequency\_rules\_sorted.head(15) |



* Ta nhận thấy có một mối quan hệ rõ rệt trong các luật kết hợp:
  + Kết quả cho thấy một số luật kết hợp với Severe BALANCE\_FREQUENCY (tần suất cập nhật số dư cao) như là kết quả, điều này gợi ý rằng có mối quan hệ giữa các yếu tố như số lần giao dịch mua (PURCHASES\_TRX), số tiền thanh toán tối thiểu (MINIMUM\_PAYMENTS), và tần suất cập nhật số dư.
  + Các luật này có giá trị hỗ trợ (support) khoảng từ 0.753 đến 0.762, nghĩa là khoảng 75.3% đến 76.2% của dữ liệu tuân theo các luật này. Đây là một tỷ lệ tương đối cao, cho thấy rằng những luật này thường xuyên xuất hiện trong dữ liệu.
* Độ tin cậy (confidence) của các luật dao động từ 0.785 đến 0.795, cho thấy rằng khi các điều kiện (antecedents) thỏa mãn, kết quả (Severe BALANCE\_FREQUENCY) xảy ra với xác suất khá cao.
* Các giá trị lift dao động quanh mức 0.99 đến 1.01, cho thấy rằng mặc dù có mối quan hệ, nhưng sự tăng lên trong khả năng xuất hiện của các kết quả (consequents) khi các điều kiện (antecedents) thỏa mãn không phải là quá lớn. Điều này chỉ ra rằng mặc dù có mối quan hệ, nhưng không phải là một sự phụ thuộc mạnh mẽ.
* Ý nghĩa thực tiễn:
  + Kết quả này gợi ý rằng khi số lượng giao dịch mua hàng (PURCHASES\_TRX) thấp và số tiền thanh toán tối thiểu (MINIMUM\_PAYMENTS) thấp, người dùng có xu hướng tăng cường tần suất cập nhật số dư. Điều này có thể phản ánh một nhu cầu quản lý tài chính cá nhân chặt chẽ hơn trong nhóm khách hàng này.
  + Các chiến lược quản lý dịch vụ có thể tập trung vào nhóm khách hàng này để cung cấp các công cụ và dịch vụ giúp họ quản lý tài chính hiệu quả hơn.
* **Quy trình Sử dụng thẻ tín dụng trong thời gian dài:**

Nghi ngờ: Người dùng có xu hướng sử dụng thẻ tín dụng lâu dài sau khi đã thực hiện nhiều giao dịch mua hàng không?

|  |
| --- |
| # Lọc các luật có liên quan đến PURCHASES\_TRX và TENURE  long\_term\_usage\_rules = rules[rules['antecedents'].astype(str).str.contains('PURCHASES\_TRX') & rules['antecedents'].astype(str).str.contains('TENURE')]  # Lọc tiếp các luật có consequents là BALANCE  balance\_rules = long\_term\_usage\_rules[long\_term\_usage\_rules['consequents'].astype(str).str.contains('BALANCE')]  # Sắp xếp theo support giảm dần  balance\_rules\_sorted = balance\_rules.sort\_values(by='support', ascending=False)  balance\_rules\_sorted.head(15) |



* Ta nhận thấy có một mối quan hệ rõ rệt trong các luật kết hợp:
  + Kết quả cho thấy một số luật kết hợp với low BALANCE (Số dư thấp) như là kết quả, điều này gợi ý rằng có mối quan hệ giữa các yếu tố như thời gian sử dụng thẻ tín dụng (TENURE), số lượng giao dịch mua hàng được thực hiện (PURCHASES\_TRX), và số dư.
  + Các luật này có giá trị hỗ trợ (support) khoảng 0.71, nghĩa là khoảng 71% của dữ liệu tuân theo các luật này. Đây là một tỷ lệ tương đối cao, cho thấy rằng những luật này thường xuyên xuất hiện trong dữ liệu.
  + Độ tin cậy (confidence) của các luật dao động từ 0.82 đến 0.84, cho thấy rằng khi các điều kiện (antecedents) thỏa mãn, kết quả (Severe BALANCE\_FREQUENCY) xảy ra với xác suất khá cao.
* Ý nghĩa thực tiễn:

Kết quả này gợi ý rằng khi thời gian sử dụng thẻ (TENURE) lâu năm và số lượng giao dịch mua hàng được thực hiện (PURCHASES\_TRX) thấp, Số dư (BALANCE) có xu hướng thấp. Điều này có thể phản ánh hành vi tiêu dùng của người dùng thẻ tín dụng. Cụ thể, những người sử dụng thẻ lâu dài nhưng ít tiêu dùng có thể giữ mức số dư thấp trên tài khoản của họ. Điều này có thể do họ tích lũy số dư trong thời gian dài hoặc thực hiện ít giao dịch mua hàng, dẫn đến mức số dư thấp hơn.

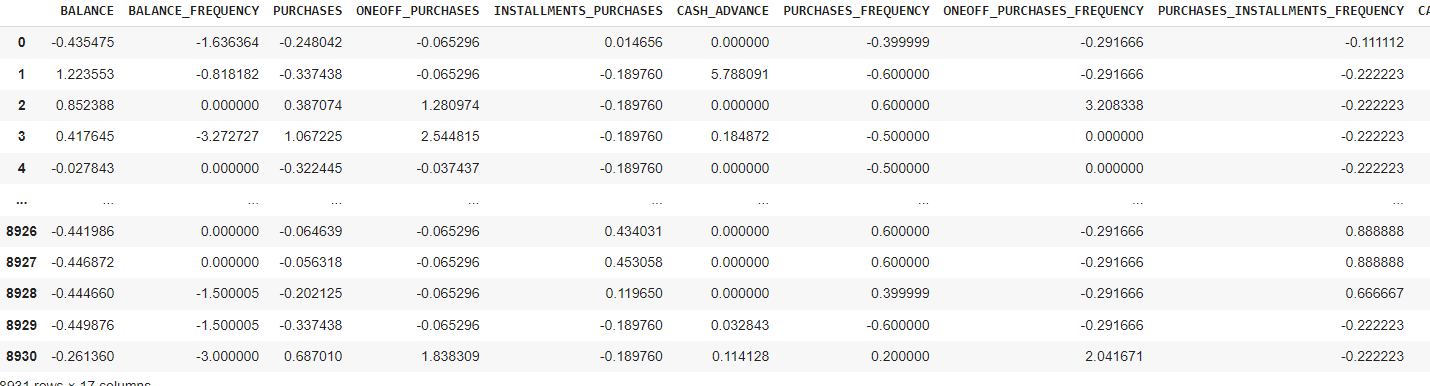
Tuy nhiên, cần lưu ý rằng mối quan hệ này cần được xác nhận thông qua các nghiên cứu chi tiết hơn và có thể bị ảnh hưởng bởi các yếu tố khác như thu nhập, mục tiêu tài chính cá nhân, hoặc thói quen tiêu dùng cá nhân của mỗi người dùng.

### Giảm chiều dữ liệu với PCA

#### **3.3.3.1. Chuẩn hoá dữ liệu** (RobustScaler)

RobustScaler loại bỏ giá trị trung bình và chia tỷ lệ dữ liệu theo phạm vi lượng tử (mặc định là IQR: Phạm vi liên tứ phân vị). IQR là khoảng giữa tứ phân vị thứ 1 và tứ phân vị thứ 3. Việc chuẩn hóa tập dữ liệu là quá trình tiền xử lý phổ biến. Tuy nhiên, các giá trị ngoại lệ thường có thể ảnh hưởng theo hướng tiêu cực. Trong trường hợp như thế, việc sử dụng dải phân vị thường cho kết quả ổn định hơn.

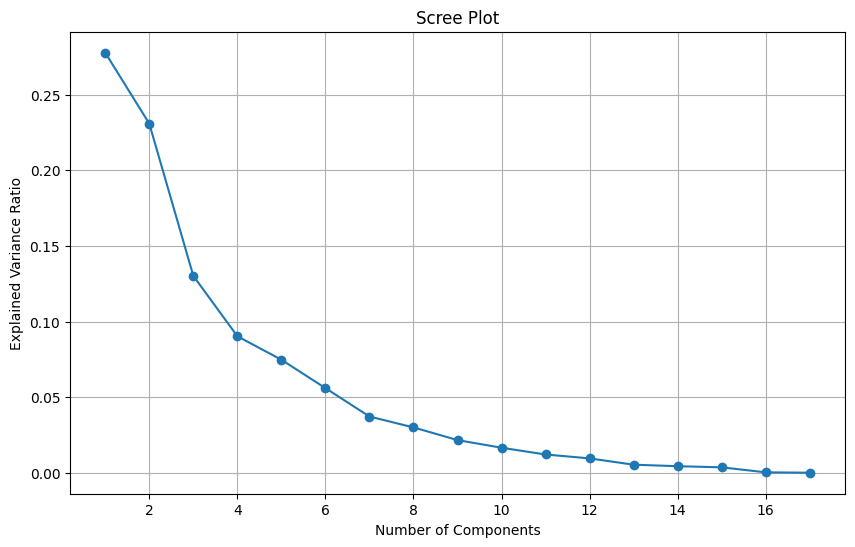
Chuẩn hoá RobustScaler sẽ cho df như sau:



#### 3.3.3.2. Principal Component Analysis (PCA)

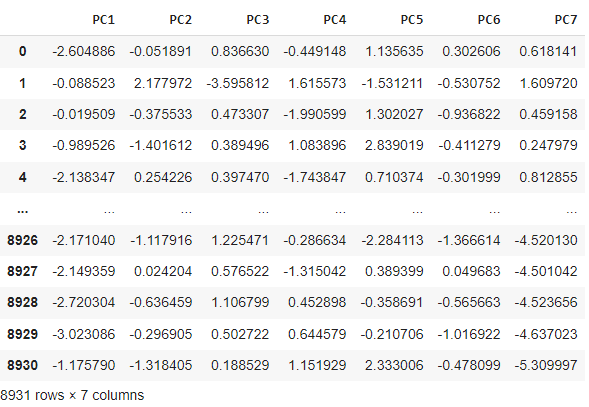
PCA thực ra là một ứng dụng đơn giản của SVD. Trong khoa học máy tính, các thuật toán nhận dạng khuôn mặt đầu tiên được sử dụng PCA và SVD để biểu diễn khuôn mặt như là một sự kết hợp tuyến tính của “eigenfaces”, làm giảm kích thước, và sau đó kết hợp khuôn mặt với các tính chất thông qua các phương pháp đơn giản. Mặc dù các phương pháp hiện đại phức tạp hơn nhiều, nhiều người vẫn còn phụ thuộc vào các kỹ thuật tương tự.

Để xác định được số nhân tố chính cần giữ lại, ta xem xét đồ thị sau:

Scree plot là một biểu đồ quan trọng trong phân tích thành phần chính (PCA) giúp xác định số lượng thành phần chính (principal components) cần giữ lại. Nhận Dạng Điểm Gãy (Elbow Point)

Điểm gãy (elbow point): Là vị trí trên biểu đồ nơi mà đường cong bắt đầu dốc xuống ít hơn, tức là tốc độ giảm của eigenvalues giảm đi. Ý nghĩa: Điểm gãy chỉ ra số lượng thành phần chính cần giữ lại. Trước điểm gãy, các eigenvalues giảm mạnh, nghĩa là mỗi thành phần chính tiếp theo giải thích một lượng lớn phương sai. Sau điểm gãy, các eigenvalues giảm chậm hơn, nghĩa là các thành phần chính tiếp theo giải thích ít phương sai hơn và có thể bỏ qua.

Ở đây, ta thấy elbow nằm ở số 7, nên ta sẽ chọn n\_component = 7 cho PCA, vì sau 7 thì càng ngày các eigenvalue càng giảm đi, nên ta có thể lược bớt chúng.

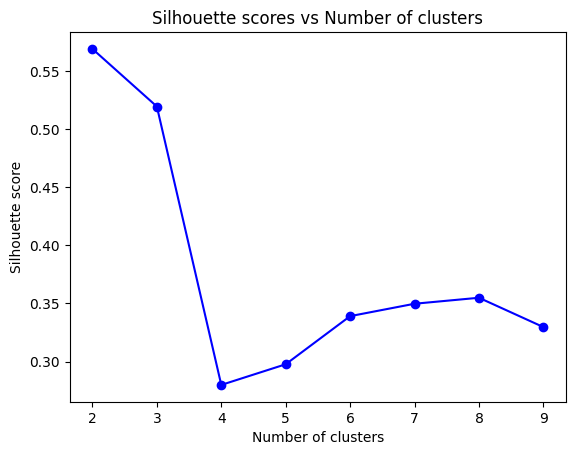


# CHƯƠNG IV. MÔ HÌNH

## Giải quyết bài toán phân cụm, phân cấp khách hàng

### K-mean

K-means là phương pháp phân cụm dựa trên việc chia dữ liệu thành k cụm sao cho tổng bình phương khoảng cách từ các điểm dữ liệu đến trung tâm cụm (centroid) là nhỏ nhất. Thuật toán bắt đầu bằng cách chọn ngẫu nhiên k centroids, sau đó lặp lại việc gán các điểm dữ liệu vào cụm gần nhất và cập nhật lại centroids dựa trên trung bình các điểm dữ liệu trong từng cụm cho đến khi hội tụ.



Khi chọn số lượng cụm k=3 thay vì k=2, ta đã dựa trên biểu đồ Silhouette score. Dù Silhouette score cao nhất khi k=2, nhưng giá trị này không chênh lệch quá nhiều so với khi k=3 (chỉ giảm một ít). Điều này cho thấy rằng việc tăng số cụm từ 2 lên 3 vẫn mang lại một mức độ nhóm tương đối tốt.

Lý do chính để chọn k=3 thay vì k=2 là vì với k=3, chúng ta có thể phân loại dữ liệu thành nhiều nhóm hơn, giúp hiểu rõ hơn về cấu trúc và đặc điểm của dữ liệu. Trong khi đó, với k=2, dữ liệu có thể bị chia một cách quá tổng quát và không phản ánh đầy đủ các nhóm nhỏ bên trong dữ liệu.

Tóm lại, dù Silhouette score cao hơn khi k=2, nhưng việc chọn k=3 vẫn hợp lý hơn vì nó cân bằng giữa chất lượng phân cụm và sự chi tiết trong phân tích dữ liệu.

Bắt đầu xây dựng lại thuật toán phân cụm kMeans với k=3 và tính toán silhouette\_score

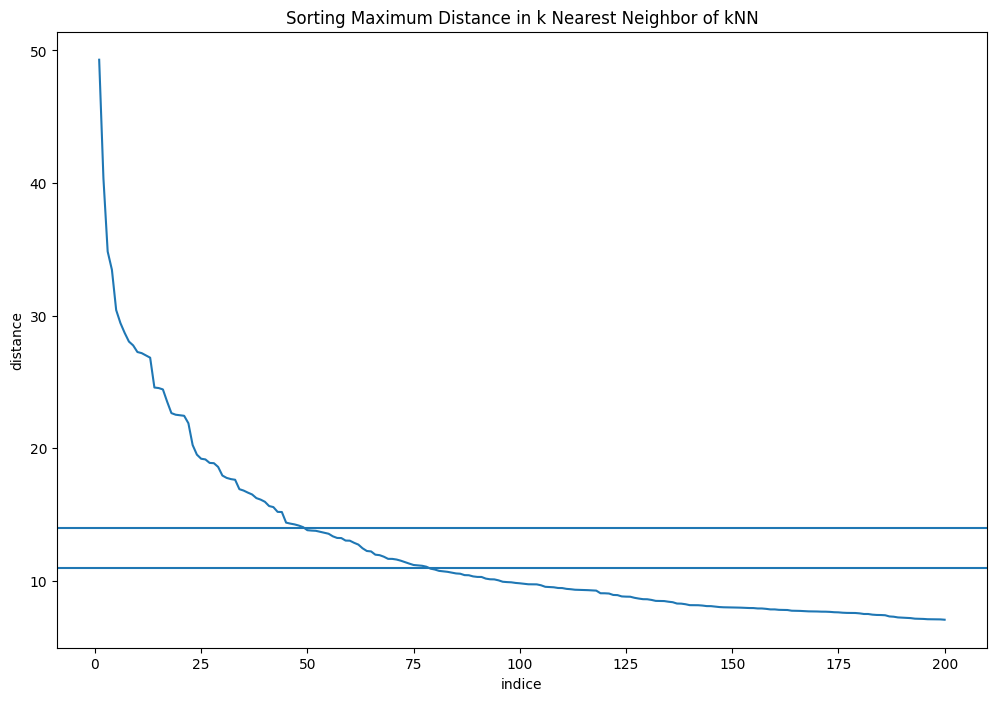


### DBscan

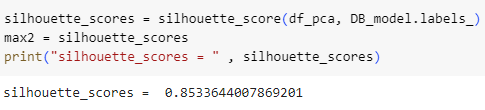
DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là phương pháp phân cụm dựa trên mật độ điểm dữ liệu. Nó xác định các cụm dựa trên vùng mật độ cao, có thể tìm ra các cụm có hình dạng bất kỳ và xử lý tốt với nhiễu. DBSCAN sử dụng hai thông số chính:

* ϵ (khoảng cách tối thiểu giữa các điểm)
* minPts (số điểm tối thiểu để hình thành một cụm). Các điểm được phân loại là core points, border points, hoặc noise points.

Trước khi bước vào thuật toán phân cụm này, ta cần xác định ϵ tối ưu qua đồ thị k-Distance để xác định ϵ cho phù hợp, Còn về minPnts, thì ta cần chọn tối thiểu là 2 x <số chiều của tập>, mà tập dữ liệu PCA của chúng ta có 7 cột, nên minPnts = 14 <=> tương đương với việc vẽ biểu đồ k-Distance với k = 15, sau đây chúng ta xem qua biểu đồ k-Distance với k = 15:

Ta dễ dàng nhận thấy ϵ nằm trong khoảng [11;14], nên ở đây ta sẽ chọn ϵ = 12,5 để tiến hành thuật toán.

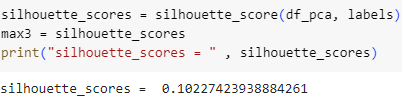
Bắt đầu xây dựng thuật toán phân cụm DBSCAN và tính silhouette\_score:



### GMM

GMM là phương pháp phân cụm dựa trên mô hình thống kê, giả định dữ liệu được sinh ra từ một tập hợp các phân phối Gaussian. Mỗi cụm được mô tả bằng một phân phối Gaussian với trung bình và ma trận hiệp phương sai riêng.

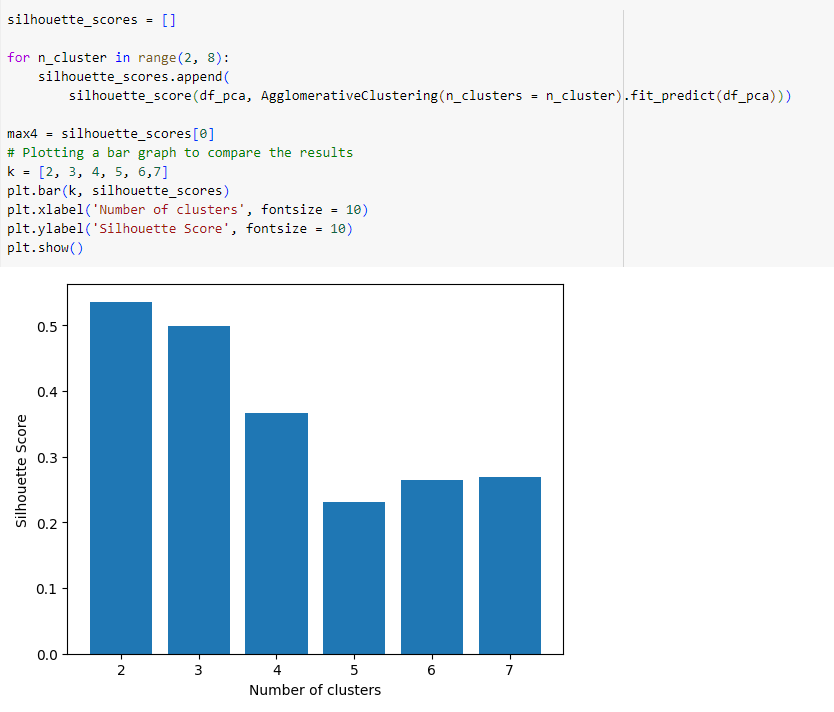
Tiến hành xây dựng thuật toán phân cụm GMM và tính điểm silhouette cho thuật toán:



### Hierarchy

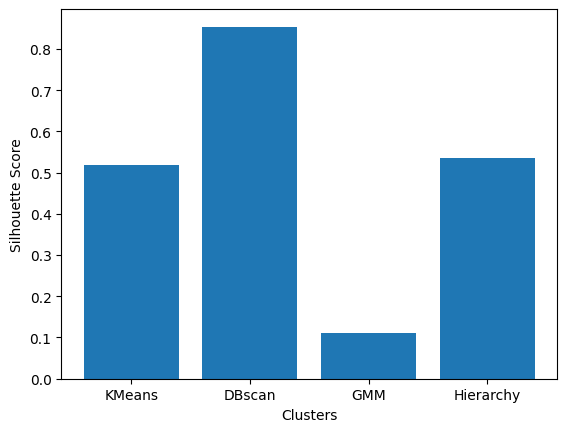
Hierarchical clustering là phương pháp phân cụm tạo ra một cấu trúc cây (dendrogram) để biểu diễn mối quan hệ phân cấp giữa các điểm dữ liệu. Có hai cách tiếp cận chính: Agglomerative (dựa dưới lên) bắt đầu với mỗi điểm là một cụm và hợp nhất các cụm gần nhất dần dần, và Divisive (dựa trên xuống) bắt đầu với một cụm chứa tất cả các điểm và chia nhỏ dần. Kết quả có thể được cắt tại mức phân cấp mong muốn để tạo ra các cụm cuối cùng.

Tiến hành xây dựng thuật toán phân cụm GMM và tính điểm silhouette cho thuật toán (lấy 1 tập k từ 2 đến 7 để so sánh số điểm từ đó chọn ra số cụm tối ưu nhất):



## Phân tích kết quả phân cụm / phân cấp khách hàng

Trước tiên, ta hãy so sánh silhouette\_score của 4 mô hình với nhau:



Dưới đây là một số lý do có thể giải thích về việc chọn **KMeans** mặc dù điểm silhouette của **DBScan** cao hơn:

* Hiệu suất và tốc độ

**KMeans** thường nhanh hơn và hiệu quả hơn so với **DBScan**, đặc biệt đối với các tập dữ liệu lớn. **DBScan** có thể gặp khó khăn trong việc xử lý các tập dữ liệu lớn hoặc có nhiều chiều.

* Đơn giản và dễ hiểu

**KMeans** là một thuật toán đơn giản và dễ hiểu, dễ triển khai và điều chỉnh. Điều này có thể quan trọng nếu bạn cần giải thích kết quả của mình cho những người không có nền tảng kỹ thuật sâu.

* Sự ổn định của kết quả

**KMeans** thường tạo ra các kết quả ổn định hơn nếu bạn có một số lượng cụm đã biết trước. **DBScan** có thể dẫn đến số lượng cụm không thể dự đoán trước và có thể thay đổi khi bạn điều chỉnh các tham số như epsilon và min\_samples.

* Tính dễ điều chỉnh và khả năng mở rộng

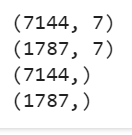
**KMeans** dễ dàng điều chỉnh số lượng cụm bằng cách thay đổi giá trị của K. Trong khi đó, **DBScan** yêu cầu điều chỉnh nhiều tham số hơn và kết quả có thể thay đổi đáng kể khi thay đổi các tham số này.

Mặc dù điểm silhouette của **DBScan** cao hơn, nhưng các yếu tố trên đã khiến nhóm quyết định chọn **KMeans** cho bài toán cụ thể này.

## Giải quyết bài toán phân loại khách hàng dựa trên các phân cụm

Tiến hành chia tập dữ liệu ra để xây dựng mô hình

|  |
| --- |
| X = np.array(df\_pca.drop('cluster',axis=1))  y = np.array(df\_pca['cluster'])  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)  for i in [X\_train, X\_test, y\_train, y\_test]:      print(i.shape) |



### Logistic Regression

|  |
| --- |
| from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error  lr = LinearRegression()  lr.fit(X\_train,y\_train)  y\_pred = lr.predict(X\_test)  r2 = r2\_score(y\_test,y\_pred)  rmse = np.sqrt(mean\_squared\_error(y\_test,y\_pred))  mae = mean\_absolute\_error(y\_test, y\_pred)  mse = mean\_squared\_error(y\_test, y\_pred) |

A number on a white background

Description automatically generated

### K-Nearest Neighbors (KNN)

|  |
| --- |
| from sklearn.metrics import classification\_report, confusion\_matrix  from sklearn.neighbors import KNeighborsClassifier  knn = KNeighborsClassifier(n\_neighbors=1)  knn.fit(X\_train, y\_train)  y\_pred = knn.predict(X\_test)  accuracy = knn.score(X\_test, y\_test)  cm = confusion\_matrix(y\_test, y\_pred) |

A screenshot of a computer

Description automatically generated

Ma trận hỗn loạn:

A graph with a blue square

Description automatically generated

### Decision Trees

|  |
| --- |
| from sklearn import tree  from sklearn import metrics  import pandas as pd  model = tree.DecisionTreeClassifier()  model.fit(X\_train, y\_train)  y\_pred = model.predict(X\_test)  accuracy = metrics.accuracy\_score(y\_test, y\_pred)  classification\_report = metrics.classification\_report(y\_test, y\_pred)  confusion\_matrix = metrics.confusion\_matrix(y\_test, y\_pred)  confusion\_matrix\_df = pd.DataFrame(confusion\_matrix, index=model.classes\_, columns=model.classes\_) |

A screenshot of a computer

Description automatically generated

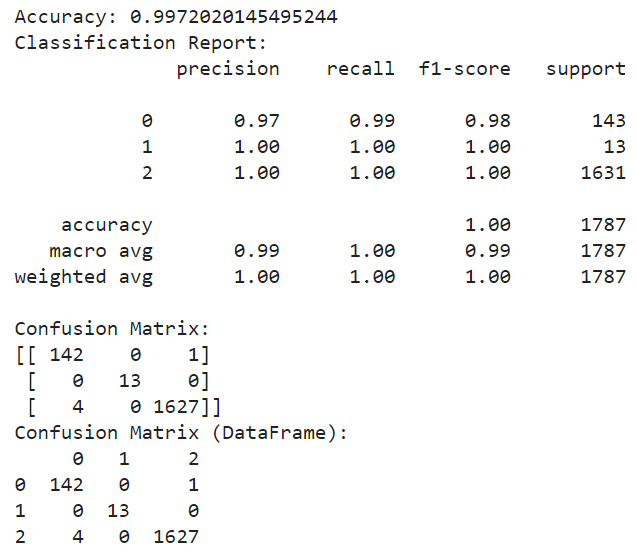
Ma trận hỗn loạn:

A graph with a blue square

Description automatically generated

### Random Forest

|  |
| --- |
| from sklearn.ensemble import RandomForestClassifier  from sklearn import metrics  import pandas as pd  import seaborn as sns  import matplotlib.pyplot as plt  model = RandomForestClassifier()  model.fit(X\_train, y\_train)  y\_pred = model.predict(X\_test)  accuracy = metrics.accuracy\_score(y\_test, y\_pred)  classification\_report = metrics.classification\_report(y\_test, y\_pred)  confusion\_matrix = metrics.confusion\_matrix(y\_test, y\_pred)  confusion\_matrix\_df = pd.DataFrame(confusion\_matrix, index=model.classes\_, columns=model.classes\_) |



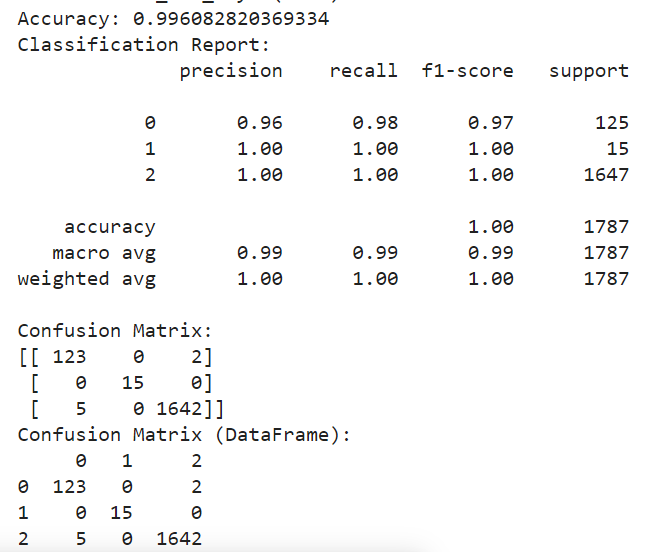
Ma trận hỗn loạn

A graph with a blue square

Description automatically generated

### ANN - Artificial Neural Networks

|  |
| --- |
| import tensorflow as tf  from tensorflow.keras import layers, models  model = models.Sequential([      layers.Dense(64, activation='relu', input\_shape=(X\_train.shape[1],)),      layers.Dense(32, activation='relu'),      layers.Dense(3, activation='softmax')  ])  model.compile(optimizer='adam',                loss='sparse\_categorical\_crossentropy',                metrics=['accuracy'])  model.fit(X\_train, y\_train, epochs=12, batch\_size=32, validation\_data=(X\_test, y\_test))  y\_pred = model.predict(X\_test)  y\_pred = np.argmax(y\_pred, axis=1)  accuracy = metrics.accuracy\_score(y\_test, y\_pred)  classification\_report = metrics.classification\_report(y\_test, y\_pred)  confusion\_matrix = metrics.confusion\_matrix(y\_test, y\_pred)  confusion\_matrix\_df = pd.DataFrame(confusion\_matrix, index=[0,1,2], columns=[0,1,2]) |



Ma trận hỗn loạn:

A graph with a blue square

Description automatically generated

# CHƯƠNG V. ĐÁNH GIÁ VÀ KẾT LUẬN

**Kết luận**

1. Ý nghĩa và mục tiêu kinh doanh:

- Khai thác thông tin từ dữ liệu khách hàng: Việc không dựa vào nhãn hoặc danh mục định sẵn giúp mở rộng khả năng phát hiện các mẫu hành vi mới, không bị giới hạn bởi các giả định trước.

- Phân nhóm khách hàng: Sử dụng các kỹ thuật học không giám sát để phân nhóm khách hàng dựa trên hành vi, sở thích hoặc đặc điểm, đặc biệt là các giao dịch tín dụng. Điều này giúp công ty hiểu rõ hơn về cơ sở khách hàng của mình và tùy chỉnh các chiến lược kinh doanh phù hợp.

2. Phương pháp học không giám sát:

- K-means clustering: Phân chia dữ liệu thành các cụm dựa trên sự tương đồng, giúp xác định các nhóm khách hàng khác nhau.

- Hierarchical clustering: Xây dựng thứ bậc các cụm, cung cấp cái nhìn sâu sắc về mối quan hệ giữa các cụm và khả năng quan sát phân cấp.

-DBSCAN: Phân cụm dựa trên mật độ, giúp phát hiện các điểm nhiễu và các cụm có hình dạng không gian phức tạp.

- Gaussian Mixture Models (GMM): Sử dụng mô hình xác suất để phân cụm, cho phép phân tích dữ liệu với các phân phối phức tạp.

-Principal Component Analysis (PCA): Giảm chiều dữ liệu, giúp trực quan hóa và hiểu cấu trúc dữ liệu, làm tiền đề cho các kỹ thuật phân cụm khác.

3. Phương pháp phân loại:

- Logistic Regression: Phương pháp phân loại đơn giản, mạnh mẽ, thích hợp cho các biến phụ thuộc nhị phân.

- K-Nearest Neighbors (KNN): Phân loại dựa trên khoảng cách, thích hợp cho việc phân loại nhanh và đơn giản.

- Decision Trees: Dễ dàng diễn giải và hiểu, giúp khai phá dữ liệu và phân loại.

-Random Forest: Mô hình ensemble kết hợp nhiều cây quyết định, tăng cường độ chính xác và khả năng dự đoán.

-Mạng Nơ-ron Nhân Tạo (ANN): Phương pháp phân loại mạnh mẽ, phù hợp với các bài toán phức tạp và dữ liệu lớn.

**Hướng phát triển đồ án**

1. Phân tích dữ liệu thực tế:

- Thu thập và tiền xử lý dữ liệu khách hàng và giao dịch tín dụng.

- Áp dụng các phương pháp học không giám sát để phân nhóm khách hàng và phân tích kết quả.

2. So sánh và đánh giá phương pháp phân cụm:

- Thực hiện và so sánh các phương pháp phân cụm như K-means, hierarchical clustering, DBSCAN, và GMM.

- Sử dụng PCA để giảm chiều dữ liệu và trực quan hóa các cụm.

3. Xây dựng mô hình phân loại:

- Áp dụng các phương pháp phân loại như Logistic Regression, KNN, Decision Trees, Random Forest, và ANN trên các nhóm khách hàng đã phân cụm.

- So sánh và đánh giá hiệu suất của các mô hình phân loại.

4. Ứng dụng kinh doanh:

- Sử dụng kết quả phân cụm và phân loại để xây dựng các chiến lược tiếp thị mục tiêu, đề xuất cá nhân hóa, phát triển sản phẩm mới và tối ưu hóa dịch vụ.

- Đánh giá hiệu quả của các chiến lược trên thông qua các chỉ số kinh doanh cụ thể.

5. Tối ưu hóa và mở rộng:

- Tối ưu hóa các mô hình dựa trên phản hồi và kết quả kinh doanh.

- Mở rộng nghiên cứu để bao gồm thêm các yếu tố và dữ liệu khác, như dữ liệu xã hội và dữ liệu thị trường, để cải thiện độ chính xác và ứng dụng của các mô hình.

Thông qua việc thực hiện đồ án này, học viên không chỉ nắm vững các kỹ thuật khai phá dữ liệu mà còn hiểu rõ cách ứng dụng chúng vào thực tế kinh doanh, từ đó nâng cao khả năng phân tích và ra quyết định.

# THAM KHẢO

[1]. <https://machinelearningcoban.com/2017/01/01/kmeans/> ,(26/5/2024)

[2]. <https://www.databricks.com/glossary/machine-learning-models> ,(26/5/2024)

[3]. Slide thầy Trần Trọng Bình

[4]. <https://vietnambiz.vn/khai-pha-luat-ket-hop-trong-co-so-du-lieu-association-rule-in-data-mining-la-gi-2020052614221747.htm> ,(26/5/2024)