

**PHÂN HIỆU TRƯỜNG ĐẠI HỌC THUỶ LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----🙞🙜🕮🙞🙜----**

****

**BÀI BÁO CÁO MÔN KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: PHÂN LOẠI CREDIT SCORE**

**Giảng viên hướng dẫn: Vũ Thị Hạnh**

**Nhóm thực hiện: Simple Lỏ**

**Thành viên nhóm:**

**1. Phùng Tấn Phương MSSV: 1951065068**

**2. Phạm Hồng Đức MSSV: 1951065077**

**3. Nguyễn Thanh Thùy MSSV: 1951065479**

**TP.HCM, ngày 02 tháng 04 năm 2023**

Mục lục

**[Phần I: Tổng quan về khai phá dữ liệu](#_Toc131371162)** [3](#_Toc131371162)

[**1.** **Phát hiện tri thức và khai phá dữ liệu** 3](#_Toc131371163)

[**2.** **Quy trình khám phá tri thức trong CSDL** 4](#_Toc131371164)

[**3.** **Mô tả bài toán Phân loại Credit score** 4](#_Toc131371165)

[**3.1.** **Tổng quan bài toán** 4](#_Toc131371166)

[**3.2.** **Phân tích dữ liệu thô** 5](#_Toc131371167)

[**Phần II: Tiền xử lý dữ liệu** 7](#_Toc131371168)

[**1.** **Tìm hiểu khái quát dữ liệu gốc** 7](#_Toc131371169)

[**2.** **Làm sạch dữ liệu** 9](#_Toc131371170)

[**2.1** **Tìm hiểu dữ liệu** 9](#_Toc131371171)

[**2.2** **Thay thế các giá trị không hợp lệ** 9](#_Toc131371172)

[**2.3** **Chuyển đổi, sửa và xóa dữ liệu lỗi** 10](#_Toc131371173)

[**2.4** **Kiểm tra dữ liệu** 15](#_Toc131371174)

[**2.5** **Chuẩn hóa dữ liệu** 24](#_Toc131371175)

[**2.6** **Mã hóa** 24](#_Toc131371176)

[**3.** **Chia dữ liệu thành 2 dữ liệu training và testing** 26](#_Toc131371177)

[**Phần III: Tổng quan về các thuật toán sử dụng** 26](#_Toc131371178)

[**1.** **Thuật toán Cây quyết định - Decision Tree** 26](#_Toc131371179)

[**1.1.** **Giới thiệu** 26](#_Toc131371180)

[**1.2.** **Mô tả thuật toán** 26](#_Toc131371181)

[**1.3.** **Ưu điểm & Nhược điểm của thuật toán** 27](#_Toc131371182)

[**1.4.** **Tích hợp dữ liệu** 28](#_Toc131371183)

[**1.5.** **Thực hiện** 28](#_Toc131371184)

[**2.** **Thuật toán Random Forest** 34](#_Toc131371185)

[**2.1.** **Giới thiệu** 34](#_Toc131371186)

[**2.2.** **Mô tả thuật toán** 34](#_Toc131371187)

[**2.3.** **Ưu điểm & Nhược điểm của thuật toán** 35](#_Toc131371188)

[**2.4.** **Thực hiện** 36](#_Toc131371189)

[**3.** **Thuật toán K-Nearest Neighbors.** 42](#_Toc131371190)

[**3.1.** **Giới thiệu** 42](#_Toc131371191)

[**3.2.** **Mô tả thuật toán** 42](#_Toc131371192)

[**3.3.** **Ưu điểm & Nhược điểm của thuật toán** 43](#_Toc131371193)

[**3.4.** **Thực hiện** 44](#_Toc131371194)

[**Phần IV: Kết luận và hướng phát triển** 46](#_Toc131371195)

[**1.** **Kết luận** 46](#_Toc131371196)

[**2.** **So sánh độ chính xác** 46](#_Toc131371197)

[**3.** **Hướng phát triển** 46](#_Toc131371198)

**Danh mục hình ảnh**

[*Hình 1: Quá trình khai phá dữ liệu từ cơ sở dữ liệu* 4](#_Toc131371143)

[*Hình 2: Phân bố khách hàng theo nghề nghiệp* 15](#_Toc131371144)

[*Hình 3: Tỷ lệ lớp tín dụng của khách hàng trong tập dữ liệu* 16](#_Toc131371145)

[*Hình 4: Bản đồ thể hiện mức độ ảnh hưởng giữa các thuộc tính* 17](#_Toc131371146)

[*Hình 5: Hình ảnh phân phối của thuộc tính Age* 18](file:///D:\Nam4\KhaiPhaDuLieu\BaiTapLon\Bài%20báo%20phân%20loại%20tín%20dụng%20V1.docx#_Toc131371147)

[*Hình 6: Hình ảnh phân phối của thuộc tính Occupation* 19](#_Toc131371148)

[*Hình 7: Hình ảnh phân phối của thuộc tính Annual\_Income* 19](#_Toc131371149)

[*Hình 8: Hình ảnh phân phối của thuộc tính Delay\_from\_due\_date* 20](#_Toc131371150)

[*Hình 9: Hình ảnh phân phối của thuộc tính Num\_of\_Delayed\_Payment* 20](#_Toc131371151)

[*Hình 10: Hình ảnh phân phối của thuộc tính Outstanding\_Debt* 21](#_Toc131371152)

[*Hình 11: Hình ảnh phân phối của thuộc tính Credit\_History\_Age* 21](#_Toc131371153)

[*Hình 12: Hình ảnh phân phối của thuộc tính Payment\_of\_Min\_Amoun* 22](#_Toc131371154)

[*Hình 13: Hình ảnh phân phối của thuộc tính Total\_EMI\_per\_month* 22](#_Toc131371155)

[*Hình 14: Hình ảnh phân phối của thuộc tính Payment\_Behaviour* 23](#_Toc131371156)

[*Hình 15: Hình ảnh phân phối của thuộc tính Monthly\_Balance* 23](#_Toc131371157)

[*Hình 16: Hình ảnh phân phối của thuộc tính Credit\_Score* 24](#_Toc131371158)

[Hình 17: Ví dụ cây quyết định 27](#_Toc131371159)

[*Hình 18: Ví dụ Random Forest* 35](#_Toc131371160)

[*Hình 19: Ví dụ thuật toán KNN* 43](#_Toc131371161)

**Danh mục các thuật ngữ**

|  |  |
| --- | --- |
| Thuật ngữ viết tắt | Ý nghĩa |
| CSDL | Cơ sở dữ liệu |
| DT | Decision Tree |
| SVM | Support Vector Machine |
| KNN | K-Nearest Neighbors |
| NN | Neural Network |
| DT | Decision Tree |
| NB | Naive Bayes |

**Lời nói đầu**

Trong thị trường tài chính hiện nay, Credit Score là một yếu tố cực kỳ quan trọng để đánh giá khả năng thanh toán của khách hàng. Nó ảnh hưởng đến quyết định của các tổ chức tài chính trong việc cấp vay tiền và giúp họ xác định nguy cơ của việc cho vay tiền cho một cá nhân hay tổ chức nào đó.

Trong môn học Khai phá dữ liệu, chúng em đã thực hiện nghiên cứu về đề tài Credit Score và xây dựng một mô hình phân loại dựa trên dữ liệu khách hàng để dự đoán khả năng thanh toán của họ. Qua quá trình khai phá dữ liệu và sử dụng các kỹ thuật máy học, chúng em đã xác định các yếu tố ảnh hưởng đến Credit Score và xây dựng một mô hình phân loại chính xác hơn.

Trong báo cáo này, chúng em sẽ trình bày về quá trình khai phá dữ liệu và phân tích dữ liệu để xây dựng mô hình phân loại Credit Score, bao gồm các bước tiền xử lý dữ liệu, phân tích đặc trưng và huấn luyện mô hình. Ngoài ra, chúng em cũng sẽ giới thiệu về các kỹ thuật máy học được sử dụng trong mô hình phân loại và đưa ra những kết quả đạt được từ quá trình khai phá dữ liệu. Hy vọng rằng báo cáo này sẽ cung cấp cho bạn cái nhìn tổng quan về quá trình khai phá dữ liệu và phân loại Credit Score.

# **Phần I: Tổng quan về khai phá dữ liệu**

## **Phát hiện tri thức và khai phá dữ liệu**

Phát hiện tri thức (knowledge discovery) và khai phá dữ liệu (data mining) là hai khái niệm liên quan đến việc tìm kiếm và khai thác tri thức ẩn trong dữ liệu.

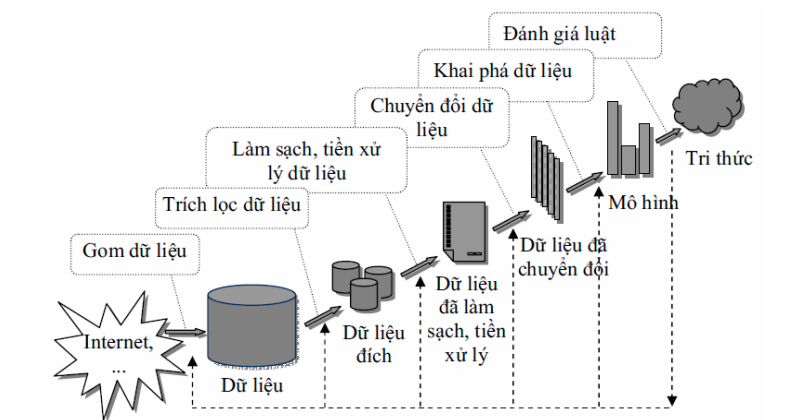
Phát hiện tri thức (knowledge discovery) là quá trình tìm kiếm, khám phá và trích xuất tri thức mới từ dữ liệu, từ đó giúp hiểu rõ hơn về mối quan hệ và luật lệ ẩn sau các dữ liệu đó.

Phát hiện tri thức không chỉ là việc trích xuất thông tin đơn giản từ dữ liệu, mà còn là quá trình phân tích và hiểu được các mẫu và quy luật tồn tại trong dữ liệu.

Khai phá dữ liệu (data mining) là một phần trong quá trình phát hiện tri thức, nó là quá trình áp dụng các phương pháp, kỹ thuật và công cụ để khai thác tri thức ẩn từ dữ liệu. Khai phá dữ liệu giúp phân tích dữ liệu một cách tự động, trích xuất thông tin có giá trị từ những dữ liệu lớn và phức tạp.

Cả hai khái niệm này đều có mục đích tìm kiếm và khai thác tri thức ẩn trong dữ liệu, nhưng phát hiện tri thức tập trung vào khía cạnh phân tích và hiểu các mẫu và quy luật ẩn trong dữ liệu, trong khi đó khai phá dữ liệu tập trung vào việc áp dụng các phương pháp và kỹ thuật để trích xuất thông tin và tri thức ẩn trong dữ liệu.

## **Quy trình khám phá tri thức trong CSDL**

****

*Hình 1: Quá trình khai phá dữ liệu từ cơ sở dữ liệu*

## **Mô tả bài toán Phân loại Credit score**

### **Tổng quan bài toán**

Phân loại Credit score là một bài toán quan trọng và thường xuyên được sử dụng trong lĩnh vực tài chính. Từ những thông tin về tình trạng tài chính của khách hàng, phân loại Credit score giúp dự đoán khả năng trả nợ của họ trong tương lai. Việc phân loại Credit score giúp đánh giá khách hàng và xác định mức lãi suất và giới hạn tín dụng phù hợp cho từng trường hợp.

Tuy nhiên, việc phân loại Credit score là một bài toán khó và phức tạp, đặc biệt là khi có quá nhiều thông tin về khách hàng. Để giải quyết vấn đề này, data-mining cung cấp các công cụ và kỹ thuật để phân tích dữ liệu và đưa ra dự đoán chính xác về Credit score của khách hàng. Trong môn học data-mining, các sinh viên được trang bị các kỹ năng và kiến thức để áp dụng các kỹ thuật như Decision Tree, Support Vector Machine (SVM), để giải quyết bài toán phân loại credit score.

Việc phân loại Credit score thông qua data-mining có thể giúp các tổ chức tài chính cải thiện khả năng quản lý rủi ro tín dụng và đưa ra các quyết định vay tiền một cách hiệu quả và an toàn hơn. Điều này giúp đảm bảo tài chính ổn định của các tổ chức và người dân, đồng thời cũng mang lại lợi ích kinh tế cho toàn xã hội.

### **Phân tích dữ liệu thô**

Đây là bộ dữ liệu mô tả về các mẫu giả định tương ứng với 3 loại thẻ tín dụng. Bộ dữ liệu này nghiên cứu và đưa ra kết luận các loại thẻ tín dụng

Link dataset: <https://www.kaggle.com/datasets/parisrohan/credit-score-classification>

Bộ dữ liệu gồm 100000 dòng dữ liệu và 28 cột thuộc tính.

Mỗi hàng đại diện cho thông tin tín dụng 1 người trong 1 tháng, và nó được dán nhãn là kém, chuẩn hay tốt. Dữ liệu này không rõ ràng, có nhiều sai sót, nhiều cột không cần thiết nên nhóm sẽ phải làm sạch rất nhiều.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Ý nghĩa | Kiểu dữ liệu |
| 1 | Id | Mã số duy nhất cho mỗi bản ghi | String |
| 2 | Customer\_ID | Mã khách hàng | String |
| 3 | Month | Tháng mà bản ghi này được thực hiện | String |
| 4 | Name | Tên của khách hàng | String |
| 5 | Age | Tuổi của khách hàng | Int |
| 6 | SSN | Số an sinh xã hội của khách hàng | String |
| 7 | Occupation | Nghề nghiệp của khách hàng | String |
| 8 | Annual\_Income | Thu nhập hàng năm của khách hàng | Float |
| 9 | Monthly\_Inhand\_Salary | Mức lương cơ sở hàng tháng của khách hàng | Float |
| 10 | Num\_Bank\_Accounts | Số lượng tài khoản ngân hàng mà khách hàng nắm giữ | Int |
| 11 | Num\_Credit\_Card | Số lượng thẻ tín dụng khác được giữ bởi khách hàng | Int |
| 12 | Interest\_Rate | Thể hiện lãi suất trên thẻ tín dụng | Int |
| 13 | Num\_of\_Loan | Thể hiện số lượng các khoản vay từ ngân hàng | Int |
| 14 | Type\_of\_Loan | Các loại khoản vay được thực hiện bởi khách hàng | String |
| 15 | Delay\_from\_due\_date | Số ngày trung bình bị trì hoãn kể từ ngày thanh toán | Int |
| 16 | Num\_of\_Delayed\_Payment | Số lần thanh toán trung bình bị trì hoãn bởi một khách hàng | Int |
| 17 | Changed\_Credit\_Limit | Phần trăm thay đổi trong giới hạn thẻ tín dụng | Float |
| 18 | Num\_Credit\_Inquiries | Số lượng yêu cầu thẻ tín dụng | Float |
| 19 | Credit\_Mix | Tổng hợp các loại khoản vay khác nhau của khách hàng | String |
| 20 | Outstanding\_Debt | Số tiền nợ chưa thanh toán (bằng USD) | Float |
| 21 | Credit\_Utilization\_Ratio | Thể hiện tỷ lệ sử dụng thẻ tín dụng | Float |
| 22 | Credit\_History\_Age | Thời gian sử dụng tín dụng của khách hàng | String |
| 23 | Payment\_of\_Min\_Amount | Số tiền tối thiểu cần thanh toán mỗi tháng | String |
| 24 | Total\_EMI\_per\_month | Tổng số tiền trả hàng tháng của khách hàng (bằng USD) | Float |
| 25 | Amount\_invested\_monthly | Số tiền đầu tư hàng tháng của khách hàng (bằng USD) | Float |
| 26 | Payment\_Behaviour | Hành vi thanh toán của khách hàng (bằng USD) | String |
| 27 | Monthly\_Balance | Số dư hàng tháng của khách hàng (bằng USD) | Float |
| 28 | Credit\_Score | Điểm tín dụng (Kém, Chuẩn, Tốt) | String |

# **Phần II: Tiền xử lý dữ liệu**

## **Tìm hiểu khái quát dữ liệu gốc**

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Các giá trị | |
| Dữ liệu gốc | Ý nghĩa |
| Id |  | Mã số duy nhất cho mỗi bản ghi |
| Customer\_ID |  | Mã khách hàng |
| Month |  | Tháng mà bản ghi này được thực hiện |
| Name |  | Tên của khách hàng |
| Age |  | Tuổi của khách hàng |
| SSN |  | Số an sinh xã hội của khách hàng |
| Occupation |  | Nghề nghiệp của khách hàng |
| Annual\_Income |  | Thu nhập hàng năm của khách hàng |
| Monthly\_Inhand\_Salary |  | Mức lương cơ sở hàng tháng của khách hàng |
| Num\_Bank\_Accounts |  | Số lượng tài khoản ngân hàng mà khách hàng nắm giữ |
| Num\_Credit\_Card |  | Số lượng thẻ tín dụng khác được giữ bởi khách hàng |
| Interest\_Rate |  | Thể hiện lãi suất trên thẻ tín dụng |
| Num\_of\_Loan |  | Thể hiện số lượng các khoản vay từ ngân hàng |
| Type\_of\_Loan |  | Các loại khoản vay được thực hiện bởi khách hàng |
| Delay\_from\_due\_date |  | Số ngày trung bình bị trì hoãn kể từ ngày thanh toán |
| Num\_of\_Delayed\_Payment |  | Số lần thanh toán trung bình bị trì hoãn bởi một khách hàng |
| Changed\_Credit\_Limit |  | Phần trăm thay đổi trong giới hạn thẻ tín dụng |
| Num\_Credit\_Inquiries |  | Số lượng yêu cầu thẻ tín dụng |
| Credit\_Mix |  | Tổng hợp các loại khoản vay khác nhau của khách hàng |
| Outstanding\_Debt |  | Số tiền nợ chưa thanh toán (bằng USD) |
| Credit\_Utilization\_Ratio |  | Thể hiện tỷ lệ sử dụng thẻ tín dụng |
| Credit\_History\_Age |  | Thời gian sử dụng tín dụng của khách hàng |
| Payment\_of\_Min\_Amount |  | Số tiền tối thiểu cần thanh toán mỗi tháng |
| Total\_EMI\_per\_month |  | Tổng số tiền trả hàng tháng của khách hàng (bằng USD) |
| Amount\_invested\_monthly |  | Số tiền đầu tư hàng tháng của khách hàng (bằng USD) |
| Payment\_Behaviour |  | Hành vi thanh toán của khách hàng (bằng USD) |
| Monthly\_Balance |  | Số dư hàng tháng của khách hàng (bằng USD) |
| Credit\_Score | Good | Thể hiện khung điểm tín dụng tốt |
| Standard | Thể hiện khung điểm tín dụng tiêu chuẩn |
| Poor | Thể hiện khung điểm tín dụng kém |

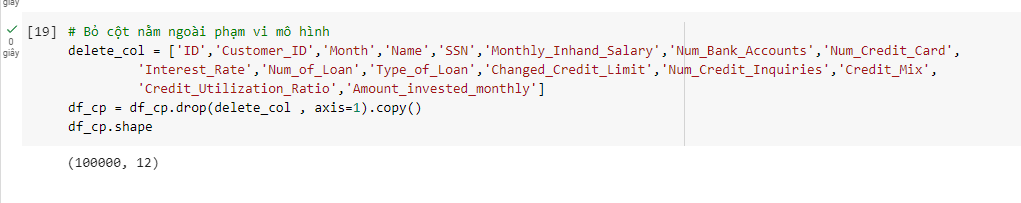
## **Làm sạch dữ liệu**

Làm sạch dữ liệu là quá trình xử lý, kiểm tra và chuẩn bị dữ liệu trước khi áp dụng các phương pháp phân tích dữ liệu. Điều này bao gồm loại bỏ các giá trị bị thiếu, sửa chữa các giá trị không chính xác hoặc bất thường và chuyển đổi dữ liệu sang định dạng phù hợp để sử dụng cho các phương pháp phân tích dữ liệu. Quá trình làm sạch dữ liệu có ý nghĩa quan trọng đối với việc đạt được kết quả phân tích chính xác và đáng tin cậy. Dữ liệu không được làm sạch có thể chứa các giá trị bất thường hoặc thiếu sót, dẫn đến kết quả phân tích sai lệch hoặc không chính xác. Ngoài ra, việc làm sạch dữ liệu cũng giúp giảm thiểu thời gian và chi phí cho việc phân tích dữ liệu.

Trong bộ dữ liệu, chúng em đã tiến hành:

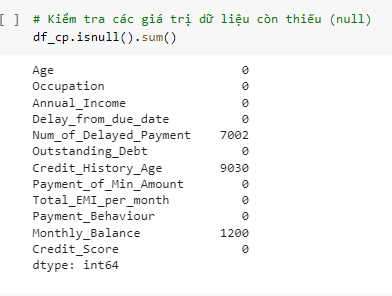
* 1. **Tìm hiểu dữ liệu**

Nhóm em tiến hành loại bỏ các cột nằm ngoài phạm vi mô hình, gồm các cột sau:

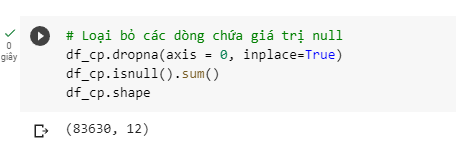


Dữ liệu sau khi xóa còn 12 cột

* 1. **Thay thế các giá trị không hợp lệ**

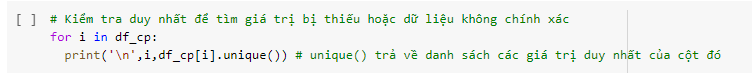
****

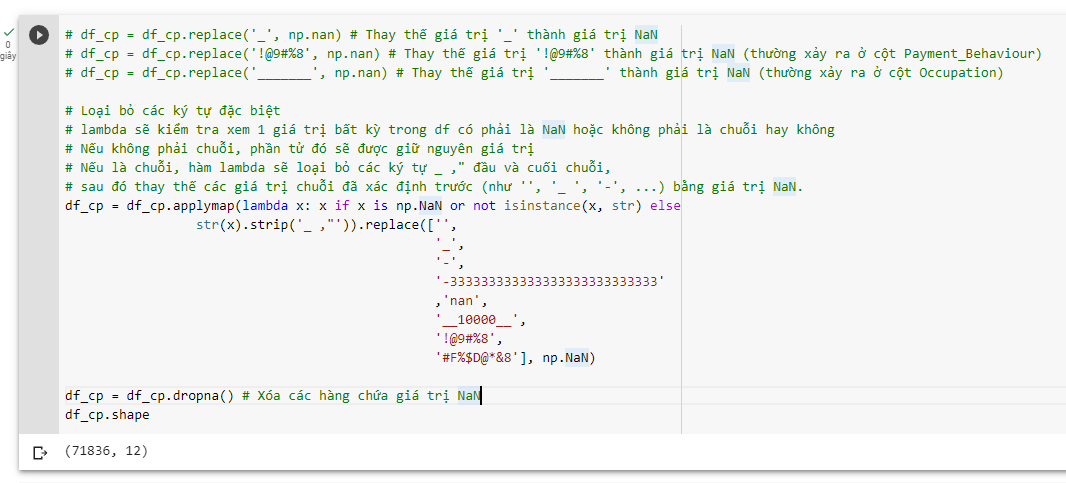
Ta thấy cột Num\_of\_Delayed\_Payment, Credit\_History\_Age, Monthly\_Balance có các dữ liệu null. Tiến hành loại bỏ các dòng đó.

****

Kết quả: Từ 100000 dòng dữ liệu, giảm xuống 83630 dòng (16370 dòng có giá trị null bị loại bỏ).

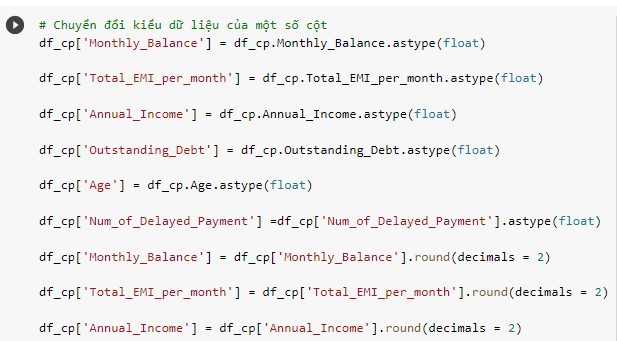
### **Chuyển đổi, sửa và xóa dữ liệu lỗi**



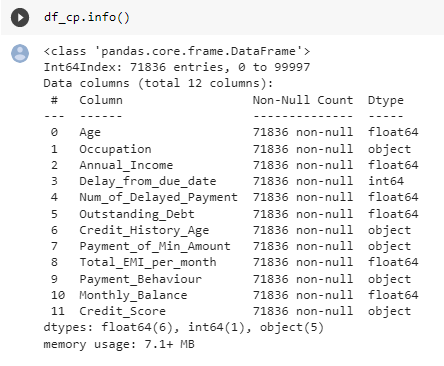


Ta thấy, dữ liệu từ 83630 dòng giảm xuống còn 71836 dòng (11794 dòng chứa giá trị không xác định bị loại bỏ)

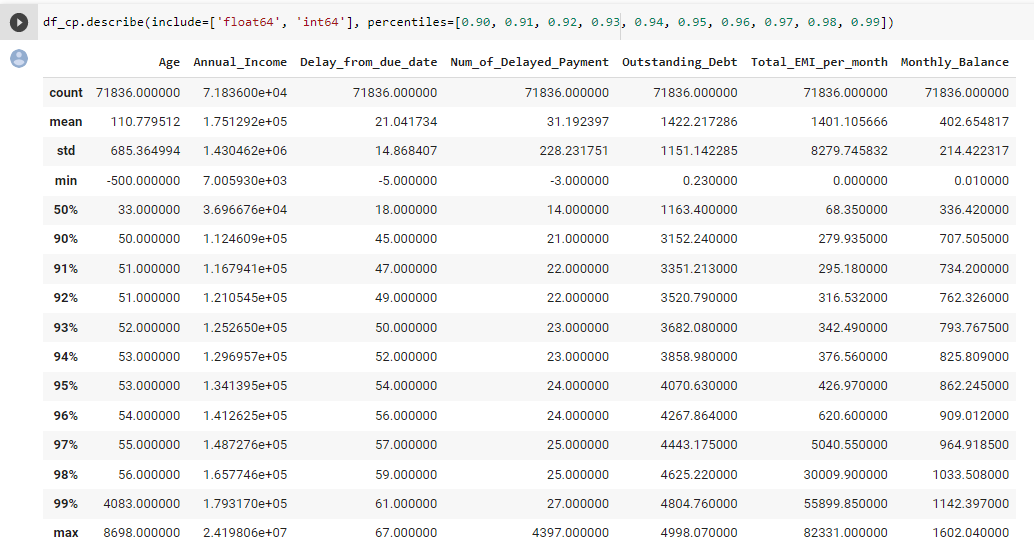
Tiếp theo, nhóm em chuyển đổi kiểu dữ liệu của 1 số cột.



Kết quả sau khi chuyển đổi:



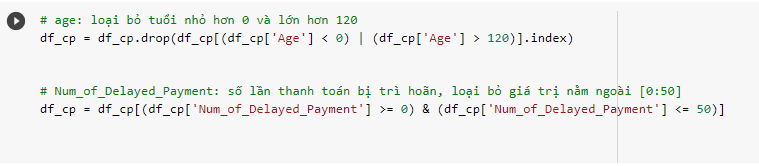
Qua quan sát mô tả các cột:



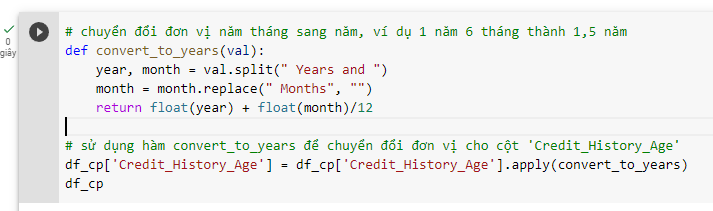
Ta thấy:

* Cột Age có giá trị [-500, 8698]: thực tế tuổi con người chỉ nằm trong khoảng [0, 120] tuổi.
* Cột Num\_of\_Delayed\_Payment có trị trị [-3, 4397]: trong khi đó 99% dữ liệu <= 27, rất có thể các giá trị lớn là sai, nhóm em giới hạn cột này trong khoảng [0, 50]

Tiến hành loại bỏ các dòng chứa giá trị nằm ngoài phạm vi.



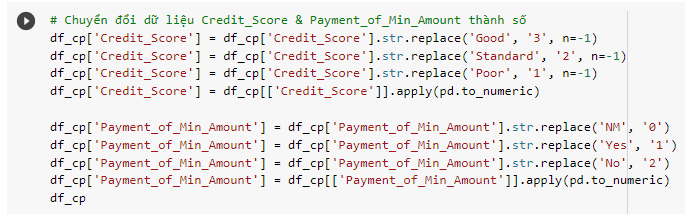
Tiếp theo đổi giá trị trong cột Credit\_History\_Age sang đơn vị năm, tức “22 Years and 1 Months” sẽ thành 22.0833, hay “10 Years and 6 Months” sẽ thành 10.5



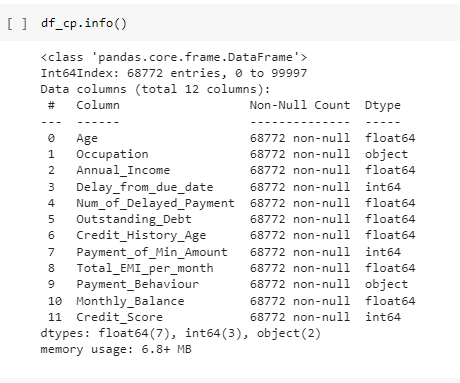
Tiếp theo chuyển đổi cột Payment\_Behaviour theo cấp độ 1 -> 6



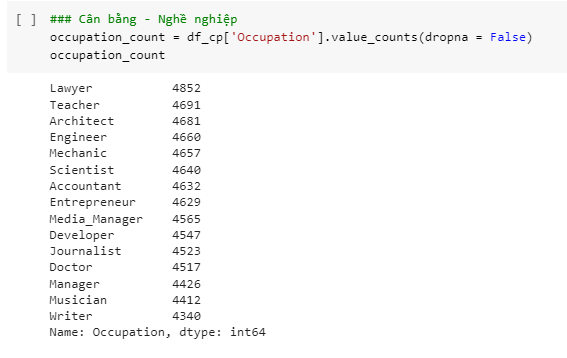
Tiếp theo chuyển chuyển đổi cột Credit\_Score và cột Payment\_of\_Min\_Amount thành số

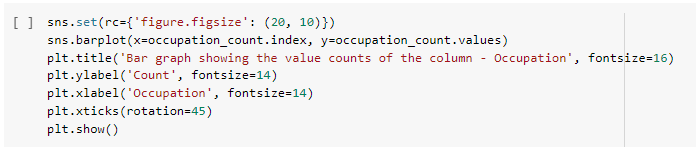


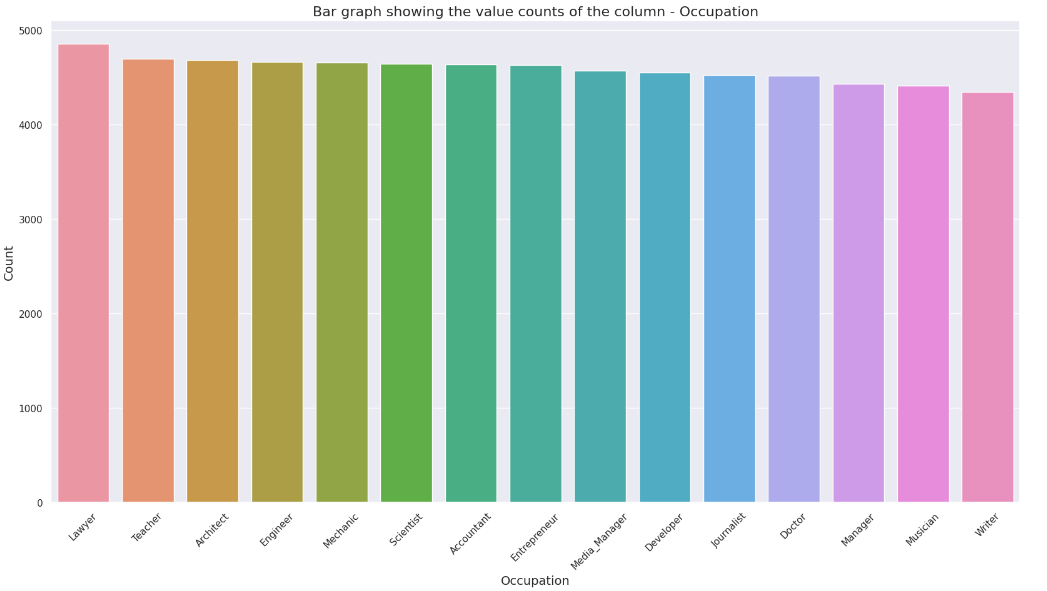
Thông tin dữ liệu sau khi làm sạch



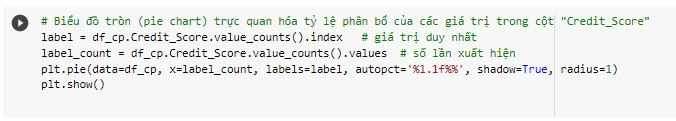
### **Kiểm tra dữ liệu**

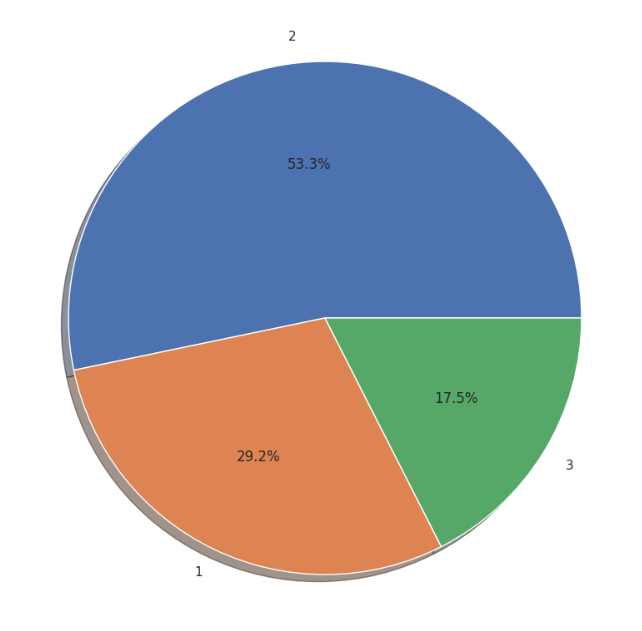




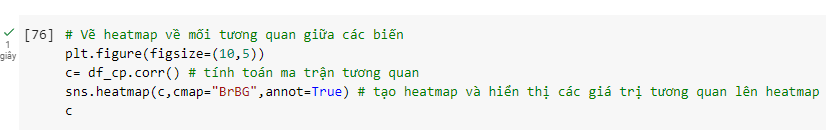


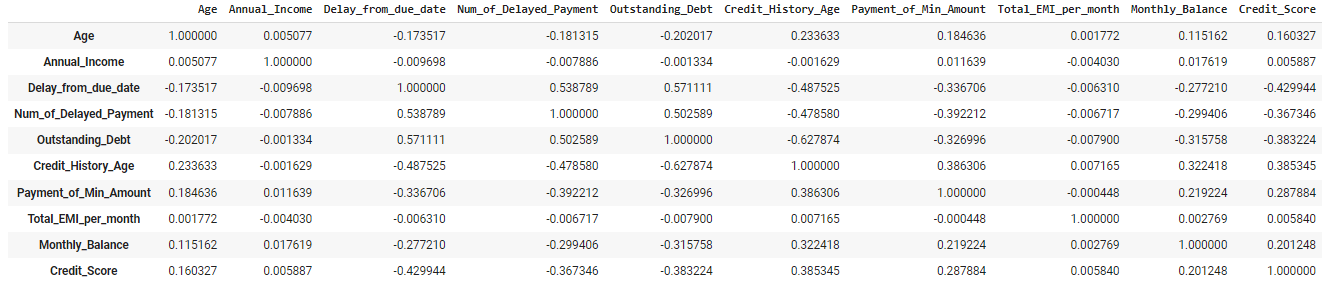
*Hình 2: Phân bố khách hàng theo nghề nghiệp*

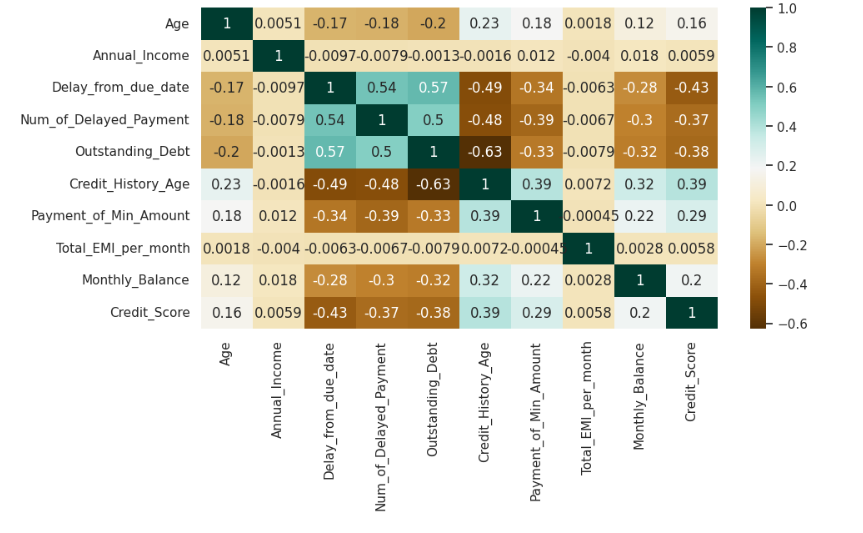




*Hình 3: Tỷ lệ lớp tín dụng của khách hàng trong tập dữ liệu*

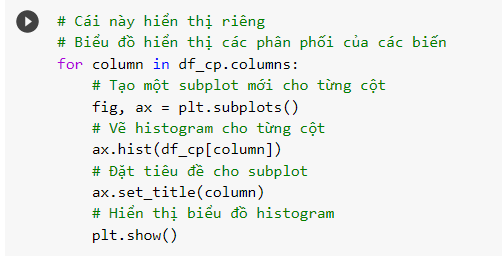


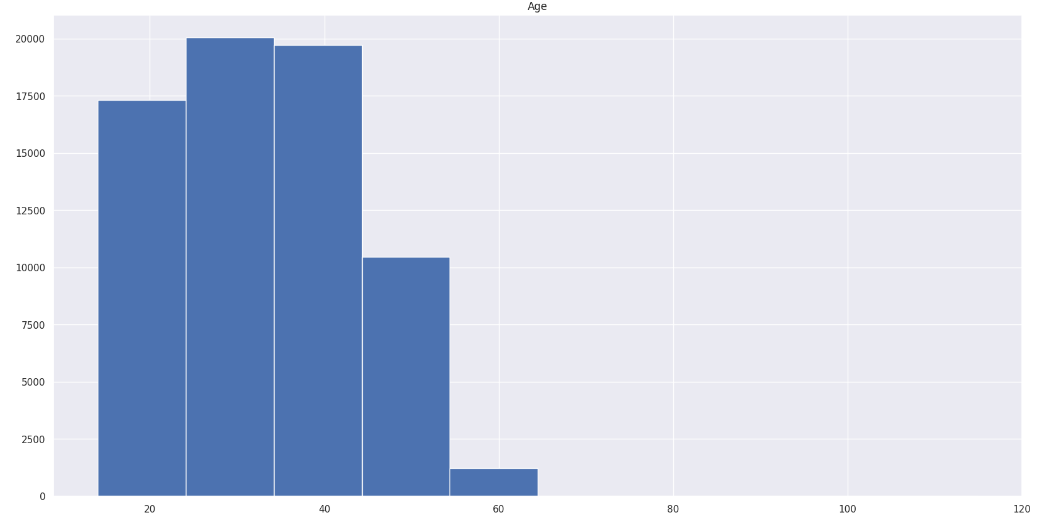




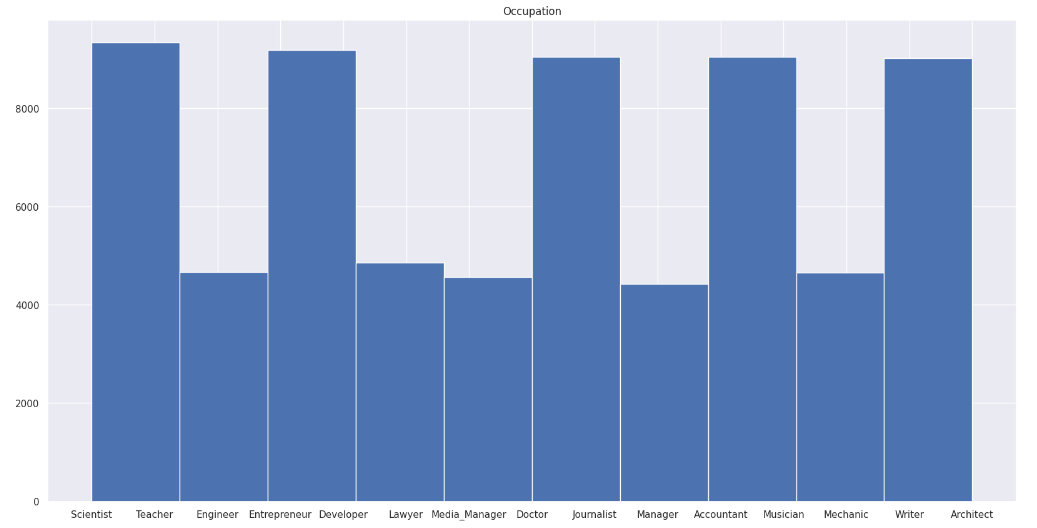
*Hình 4: Bản đồ thể hiện mức độ ảnh hưởng giữa các thuộc tính*

Biểu đồ phân phối của các thuộc tính

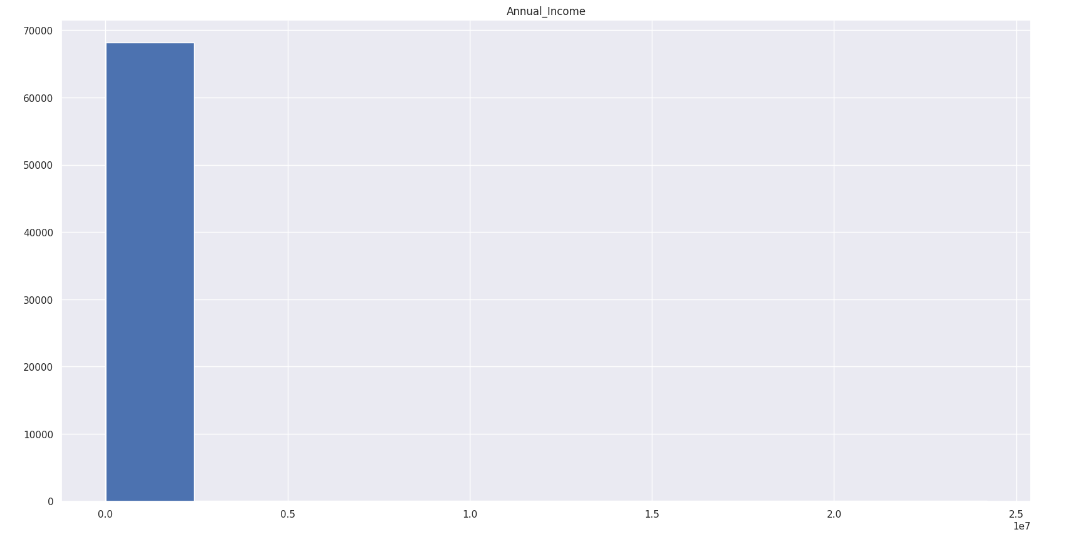




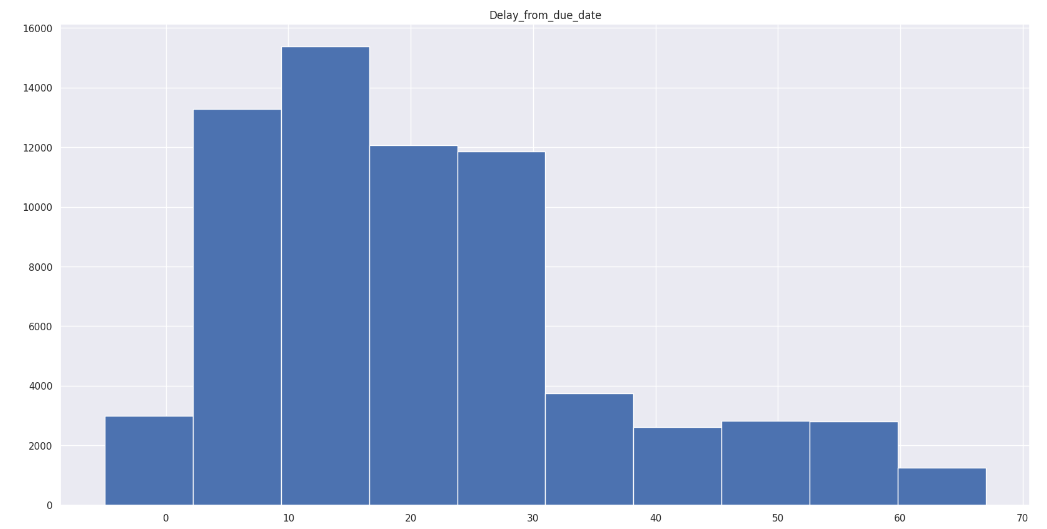
*Hình 5: Hình ảnh phân phối của thuộc tính Age*



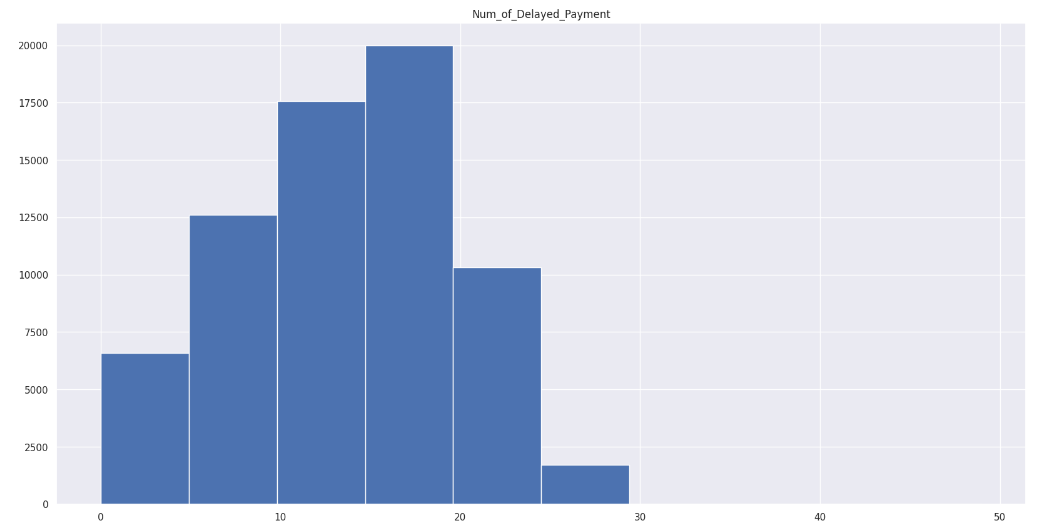
*Hình 6: Hình ảnh phân phối của thuộc tính Occupation*



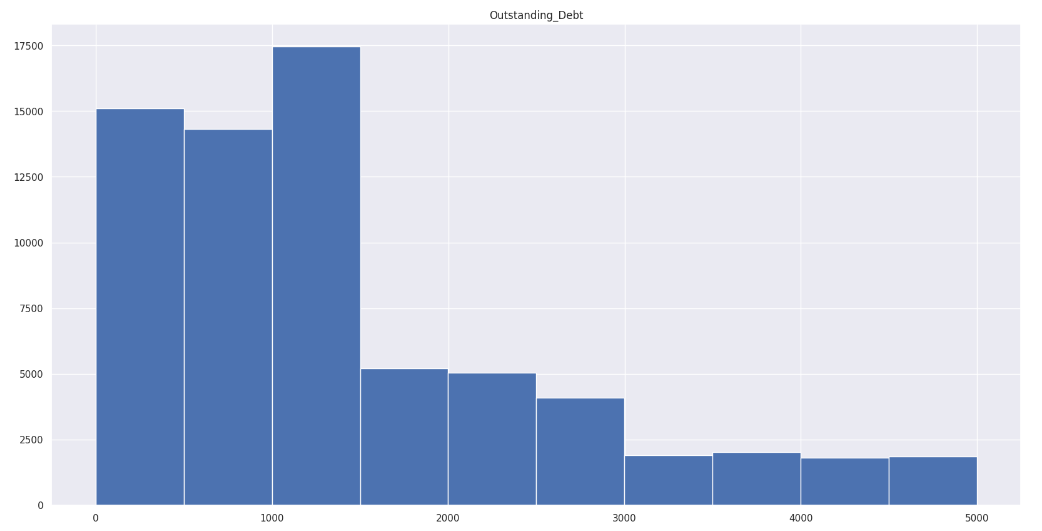
*Hình 7: Hình ảnh phân phối của thuộc tính Annual\_Income*



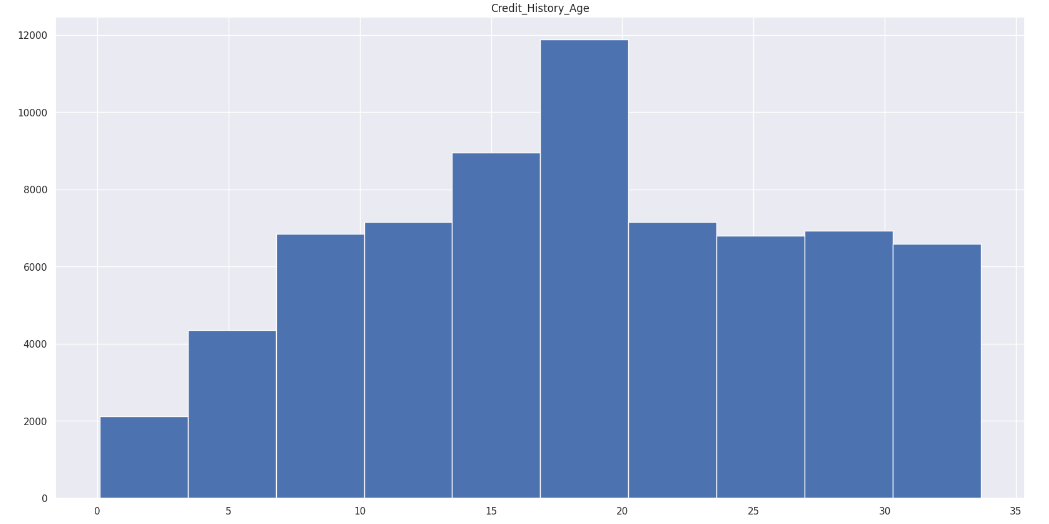
*Hình 8: Hình ảnh phân phối của thuộc tính Delay\_from\_due\_date*



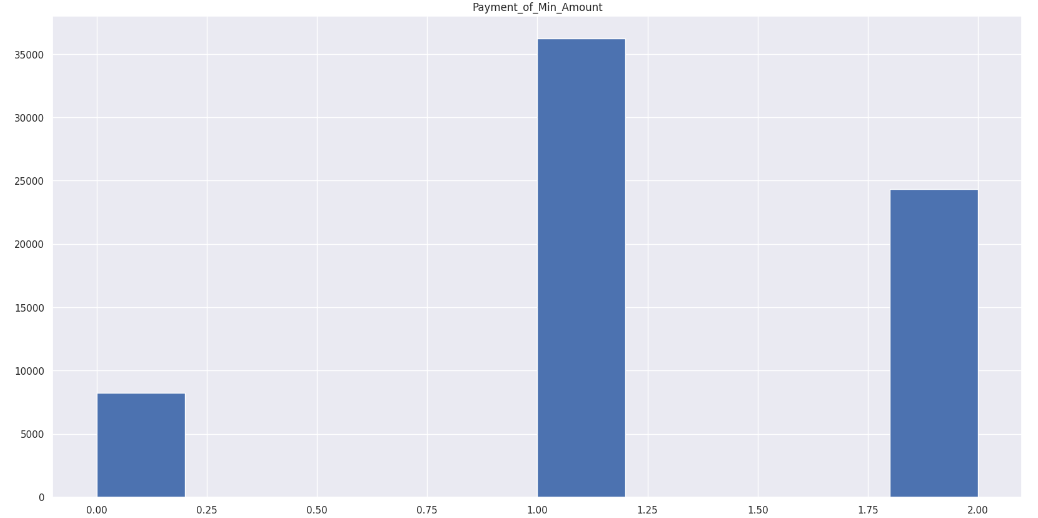
*Hình 9: Hình ảnh phân phối của thuộc tính Num\_of\_Delayed\_Payment*



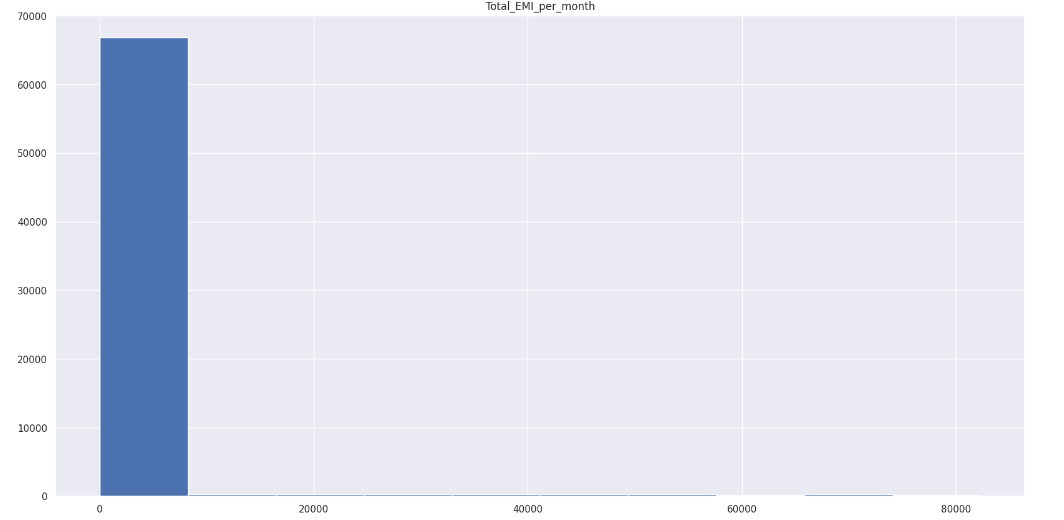
*Hình 10: Hình ảnh phân phối của thuộc tính Outstanding\_Debt*



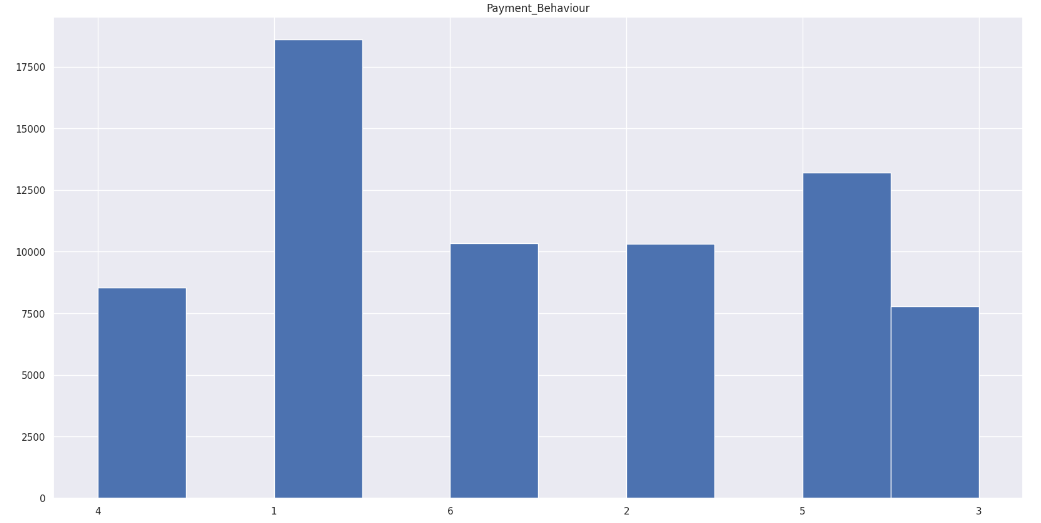
*Hình 11: Hình ảnh phân phối của thuộc tính Credit\_History\_Age*



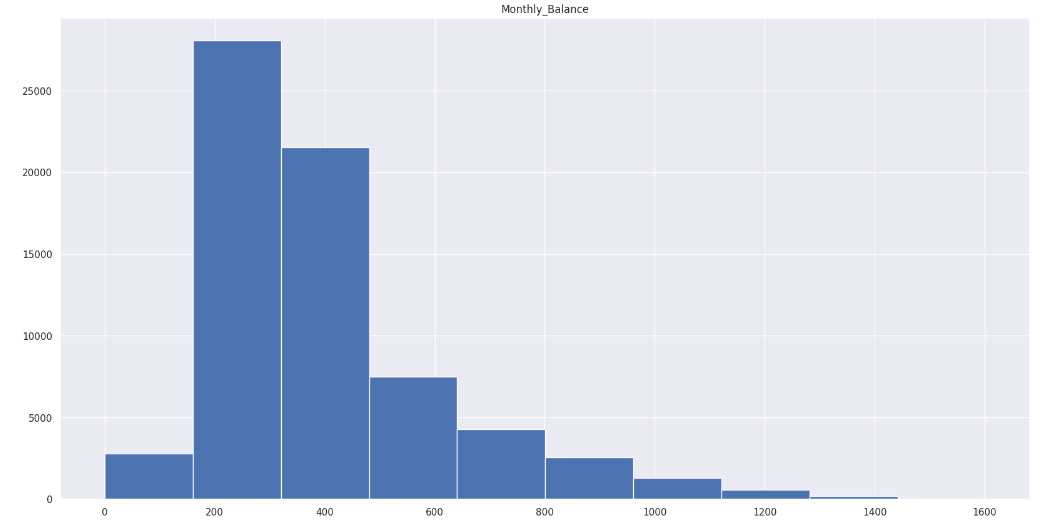
*Hình 12: Hình ảnh phân phối của thuộc tính Payment\_of\_Min\_Amoun*



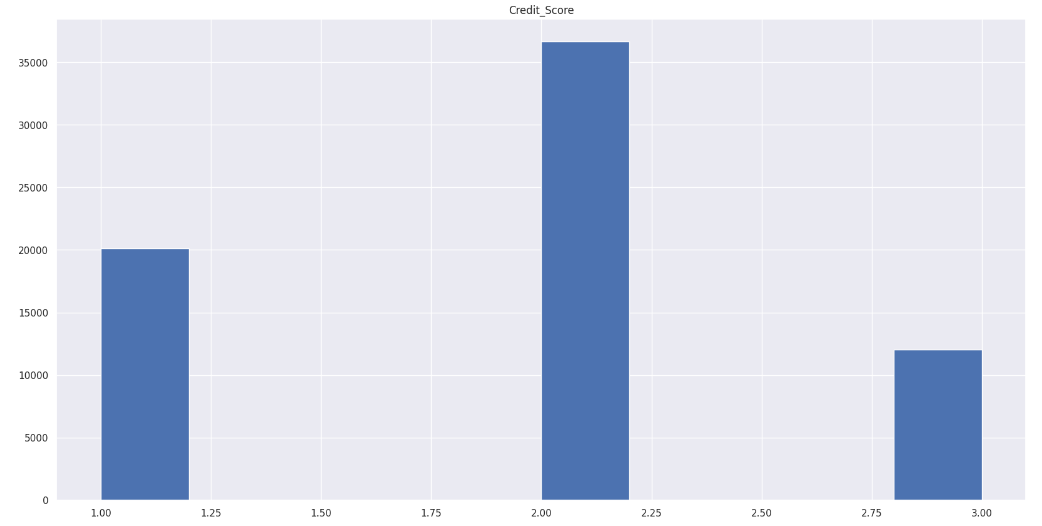
*Hình 13: Hình ảnh phân phối của thuộc tính Total\_EMI\_per\_month*



*Hình 14: Hình ảnh phân phối của thuộc tính Payment\_Behaviour*

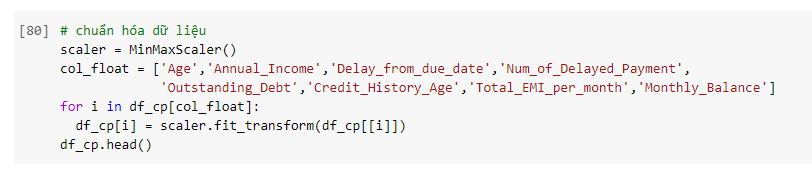


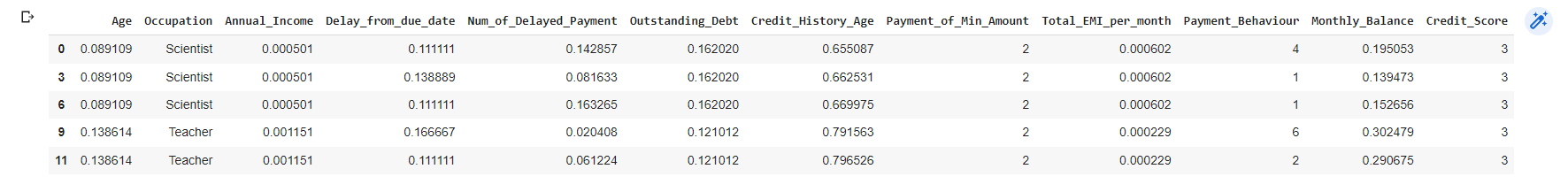
*Hình 15: Hình ảnh phân phối của thuộc tính Monthly\_Balance*



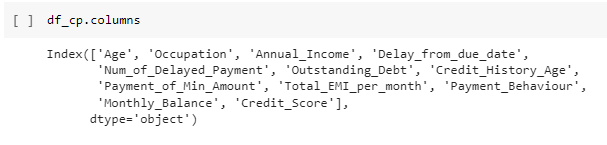
*Hình 16: Hình ảnh phân phối của thuộc tính Credit\_Score*

### **Chuẩn hóa dữ liệu**

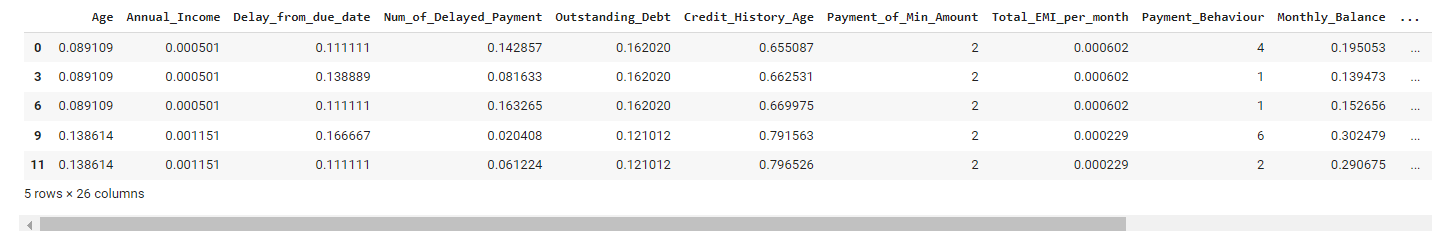


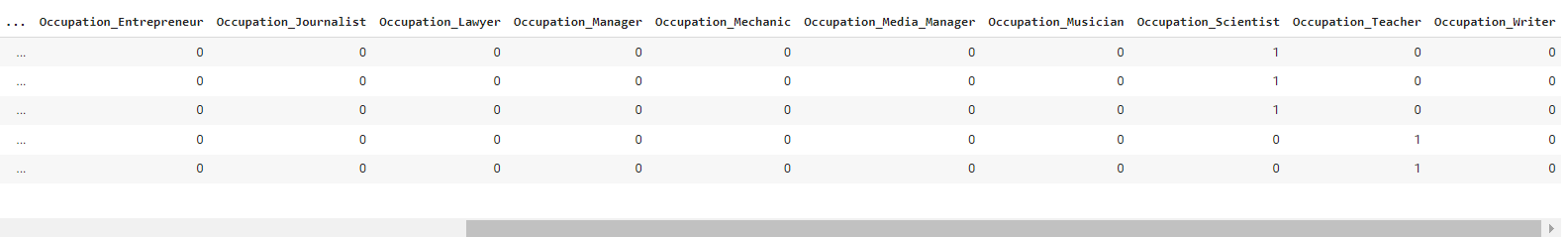


### **Mã hóa**



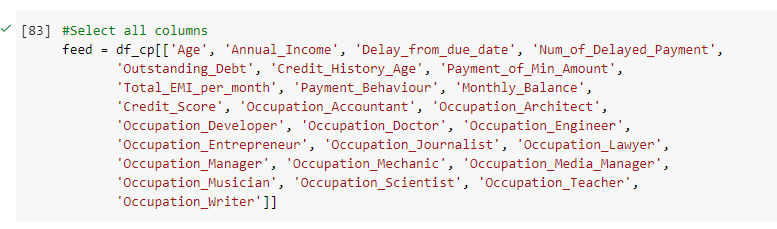


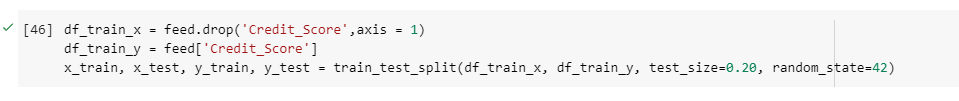




## **Chia dữ liệu thành 2 dữ liệu training và testing**

Chia tỷ lệ thành 80% training, 20% testing





# **Phần III: Tổng quan về các thuật toán sử dụng**

## **Thuật toán Cây quyết định - Decision Tree**

### **Giới thiệu**

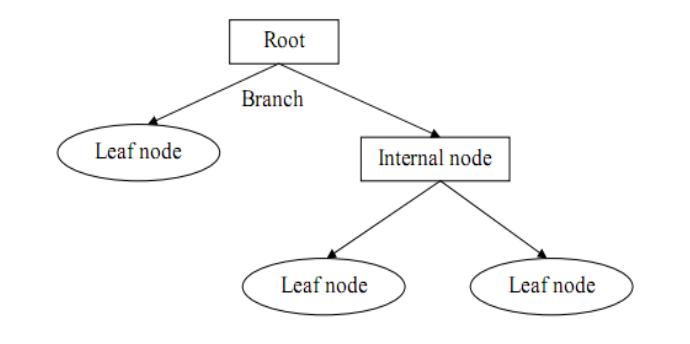
Thuật toán Cây quyết định (Decision Tree) là một phương pháp học có giám sát trong Machine Learning, được sử dụng cho cả bài toán phân loại và dự đoán. Thuật toán này tạo ra một cây quyết định dựa trên các thuộc tính của dữ liệu và mục tiêu đạt được để tìm ra quyết định tốt nhất.

Thuật toán Cây quyết định sử dụng các phép đo độ tương đồng như Entropy và Gini để tính toán sự tương đồng của các mẫu dữ liệu và chọn thuộc tính phân chia tốt nhất. Khi cây quyết định được xây dựng, nó có thể được sử dụng để phân loại hoặc dự đoán các mẫu dữ liệu mới.

Cây quyết định là một thuật toán đơn giản và hiệu quả trong việc giải quyết các bài toán Machine Learning. Nó rất phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), phân tích tín hiệu và trong các bài toán khai thác dữ liệu.

### **Mô tả thuật toán**

Cây quyết định bao gồm các nút (node) biểu diễn các thuộc tính, các cạnh biểu diễn các quan hệ giữa các thuộc tính và các giá trị của chúng, và các lá (leaf) biểu diễn kết quả của quyết định. Các nút có thể được chia thành hai loại: nút quyết định (decision node) và nút lá (leaf node).



Hình 17: Ví dụ cây quyết định

* Root: điểm ngọn chứa giá trị của biến đầu tiên được dùng để phân nhánh.
* Internal node: các điểm bên trong thân cây là các biến chứa các thuộc tính, gía trị dữ liệu được dùng để xét cho các phân nhánh tiếp theo.
* Leaf node: là các lá cây chứa giá trị của biến phân loại sau cùng.
* Branch: là quy luật phân nhánh, hay nói đơn giản là mối quan hệ giữa giá trị của biến độc lập (Internal node) và giá trị của biến mục tiêu (Leaf node)

### **Ưu điểm & Nhược điểm của thuật toán**

* Ưu điểm:
* Dễ hiểu và dễ giải thích: Cây quyết định là một thuật toán đơn giản và dễ hiểu, giúp người dùng dễ dàng giải thích kết quả dự đoán và quá trình ra quyết định.
* Tính linh hoạt: Cây quyết định có thể được sử dụng cho cả bài toán phân loại và dự đoán, và có thể xử lý dữ liệu định tính và định lượng.
* Có thể xử lý dữ liệu thiếu: Cây quyết định có khả năng xử lý dữ liệu thiếu bằng cách sử dụng các kỹ thuật như phân tán giá trị trung bình (mean imputation) và phương pháp điền giá trị trung vị (median imputation).
* Không yêu cầu chuẩn hóa dữ liệu: Thuật toán Cây quyết định không yêu cầu chuẩn hóa dữ liệu trước khi sử dụng, điều này giúp tiết kiệm thời gian và công sức trong quá trình tiền xử lý dữ liệu.
* Hiệu quả tính toán: Cây quyết định có thể được xây dựng và sử dụng nhanh chóng trên dữ liệu lớn, điều này giúp tiết kiệm thời gian và công sức trong quá trình đào tạo mô hình.
* Nhược điểm:
* Dễ bị quá khớp: Cây quyết định có xu hướng quá khớp khi tập dữ liệu huấn luyện có nhiều biến hoặc thuộc tính. Điều này dẫn đến cây quyết định quá phức tạp và không thể tổng quát hóa cho dữ liệu mới.
* Nhạy cảm với nhiễu: Nếu tập dữ liệu huấn luyện có nhiễu, cây quyết định sẽ dễ bị lỗi trong việc phân loại hoặc dự đoán giá trị của biến mục tiêu.
* Không phù hợp với dữ liệu liên tục: Thuật toán Cây quyết định không phù hợp với dữ liệu liên tục, vì nó yêu cầu các biến phải có giá trị rời rạc hoặc định danh.
* Không đảm bảo tối ưu toàn cục: Cây quyết định được xây dựng dựa trên các quyết định tại mỗi nút, vì vậy không đảm bảo tối ưu toàn cục cho các vấn đề phức tạp.
* Khó giải thích: Cây quyết định có thể trở nên rất phức tạp và khó hiểu đối với người dùng cuối. Một số cây quyết định có thể có hàng trăm hoặc hàng ngàn nút, gây khó khăn trong việc giải thích quyết định được đưa ra.

### **Tích hợp dữ liệu**

* Là quá trình trộn dữ liệu từ các nguồn khác nhau vào một kho dữ liệu sẵn sàng cho quá trình khai phá dữ liệu.

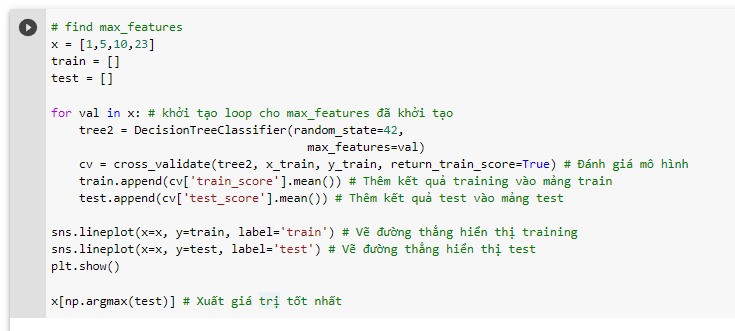
(1) Tích hợp lược đồ và so trùng đối tượng.

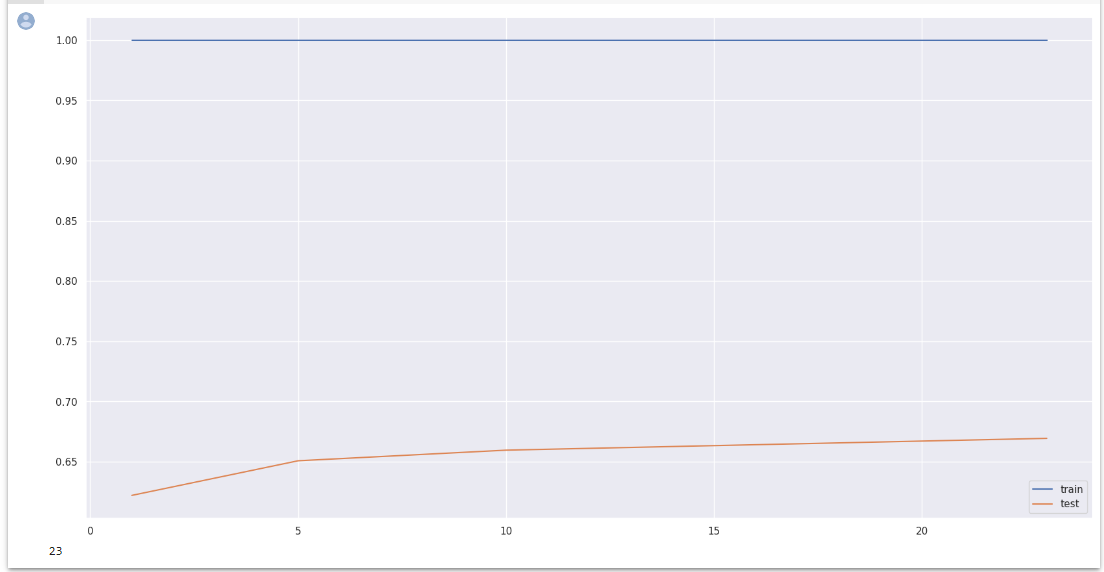
(2) Vấn đề dư thừa.

(3) Phát hiện và xử lý mâu thuẫn giá trị dữ liệu.

### **Thực hiện**

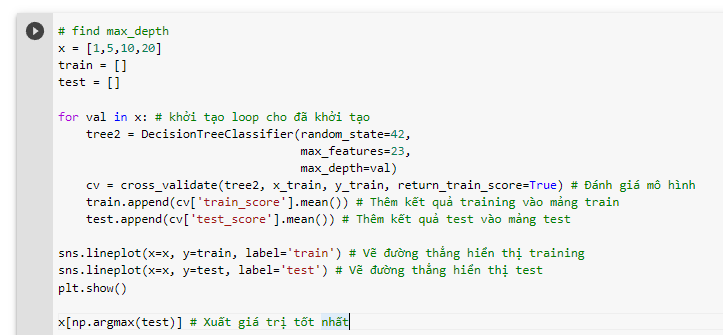
**1.5.1 Tìm số lượng đặc trưng**

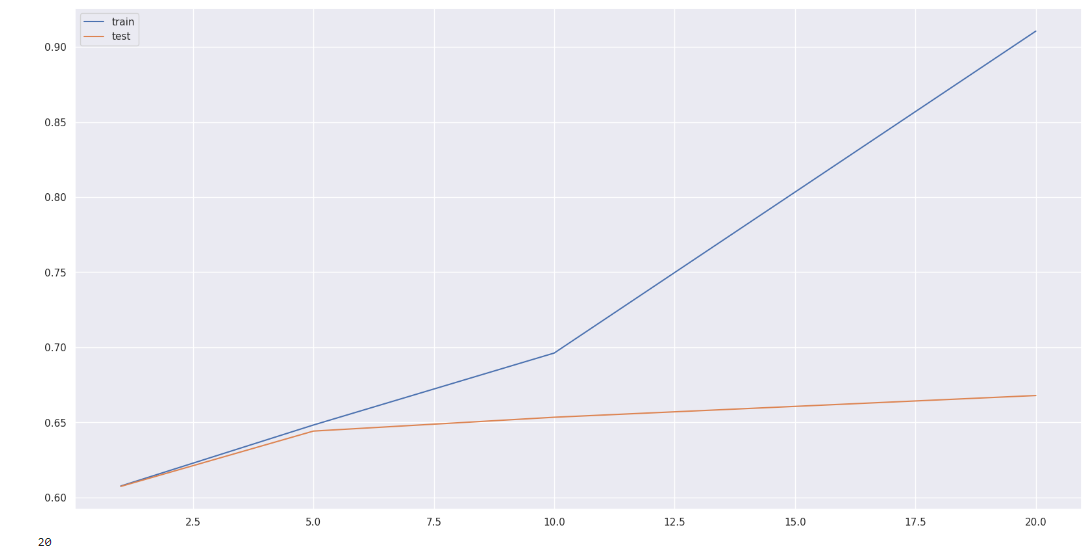




Kết quả trả về 23

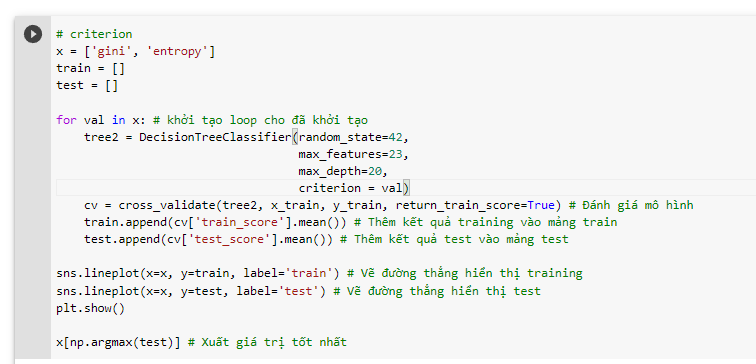
**1.5.2 Tìm độ sâu tối đa của cây quyết định**





Kết quả trả về 20

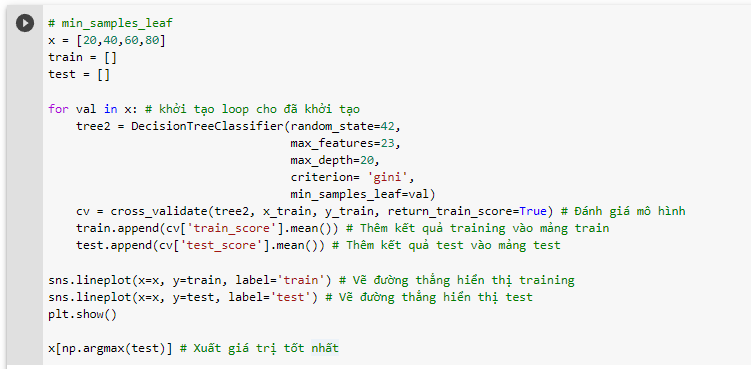
1.5.3 Tìm criterion

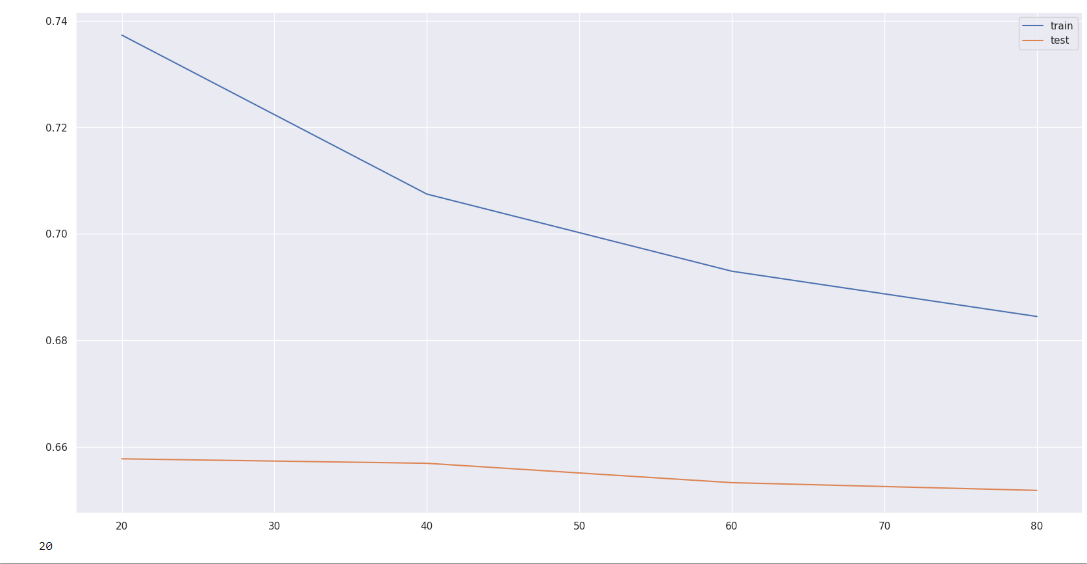




Kết quả giá trị gini phù hợp nhất

1.5.4 Tìm số lượng mẫu tối thiểu cần có

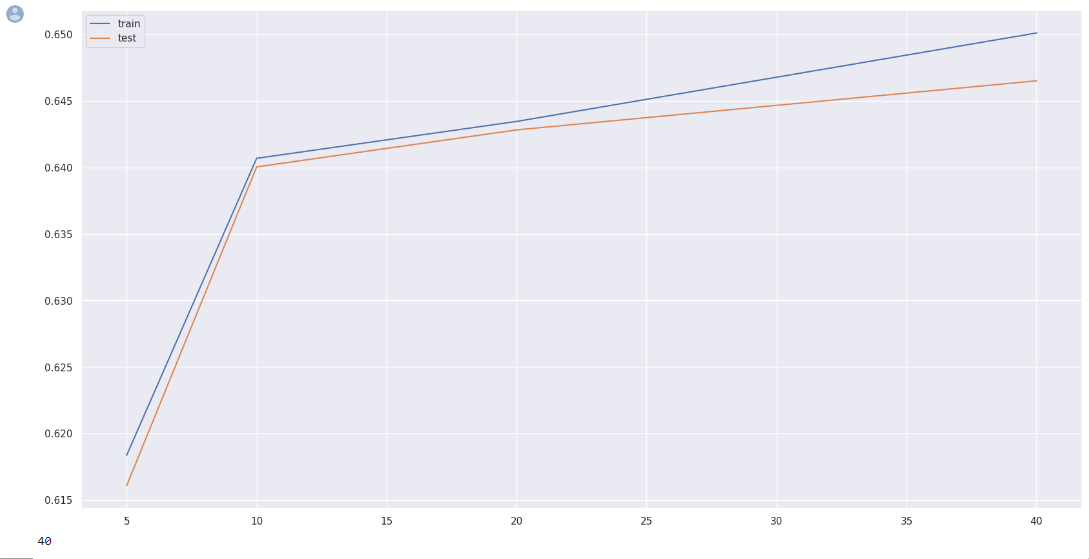




Kết quả trả về 20

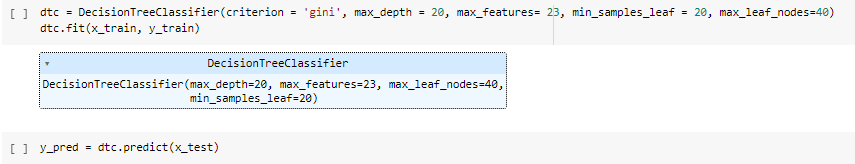
1.5.6 Tìm số lượng lá (leaf) tối đa được phép



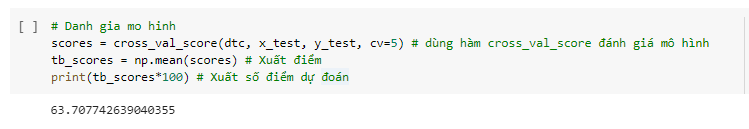


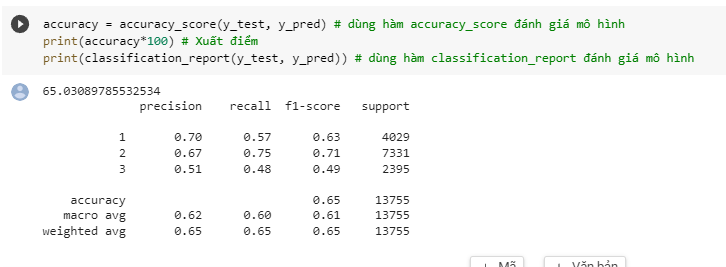
Kết quả trả về 40

1.5.7 Training data



1.5.8 Đánh giá mô hình





Hàm accuracy\_score có độ chính xác cao hơn với 65.03%

## **Thuật toán Random Forest**

### **Giới thiệu**

Thuật toán Random Forest là một thuật toán học máy dựa trên phương pháp Ensemble Learning (học tổ hợp). Đây là một phương pháp kết hợp nhiều mô hình học máy đơn giản để tạo ra một mô hình học máy phức tạp hơn, giúp cải thiện độ chính xác của mô hình.

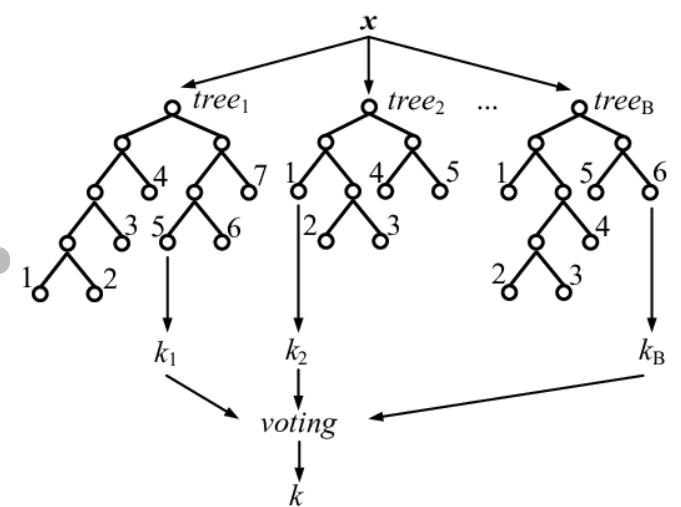
Thuật toán Random Forest hoạt động bằng cách tạo ra nhiều cây quyết định (Decision Trees) và kết hợp chúng để tạo ra một mô hình dự đoán. Mỗi cây quyết định trong Random Forest được huấn luyện trên một phần dữ liệu ngẫu nhiên và một phần các thuộc tính (features) ngẫu nhiên của tập dữ liệu.

Khi có một mẫu dữ liệu mới, mỗi cây quyết định trong Random Forest sẽ đưa ra một dự đoán và dự đoán cuối cùng được tính bằng cách lấy trung bình hoặc phương pháp biểu quyết đa số (voting) của các dự đoán cá nhân của các cây quyết định.

### **Mô tả thuật toán**

Thuật toán Random Forest hoạt động bằng cách tạo ra nhiều cây quyết định (Decision Trees) và kết hợp chúng để tạo ra một mô hình dự đoán. Cụ thể, các bước trong thuật toán Random Forest như sau:

* Tạo ngẫu nhiên nhiều tập con (subsets) từ tập dữ liệu ban đầu.
* Đối với mỗi tập con, huấn luyện một cây quyết định trên tập dữ liệu đó.
* Khi huấn luyện các cây quyết định, mỗi lần chọn một thuộc tính (feature) để phân tách cây, chỉ chọn ngẫu nhiên một số thuộc tính.
* Khi dự đoán, mỗi cây quyết định trong Random Forest sẽ đưa ra một dự đoán và dự đoán cuối cùng được tính bằng cách lấy trung bình hoặc phương pháp biểu quyết đa số (voting) của các dự đoán cá nhân của các cây quyết định.



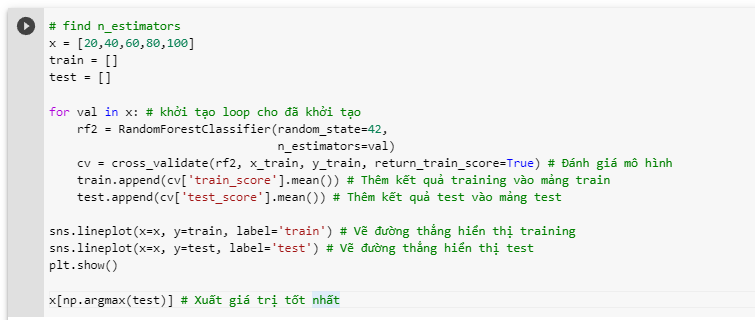
*Hình 18: Ví dụ Random Forest*

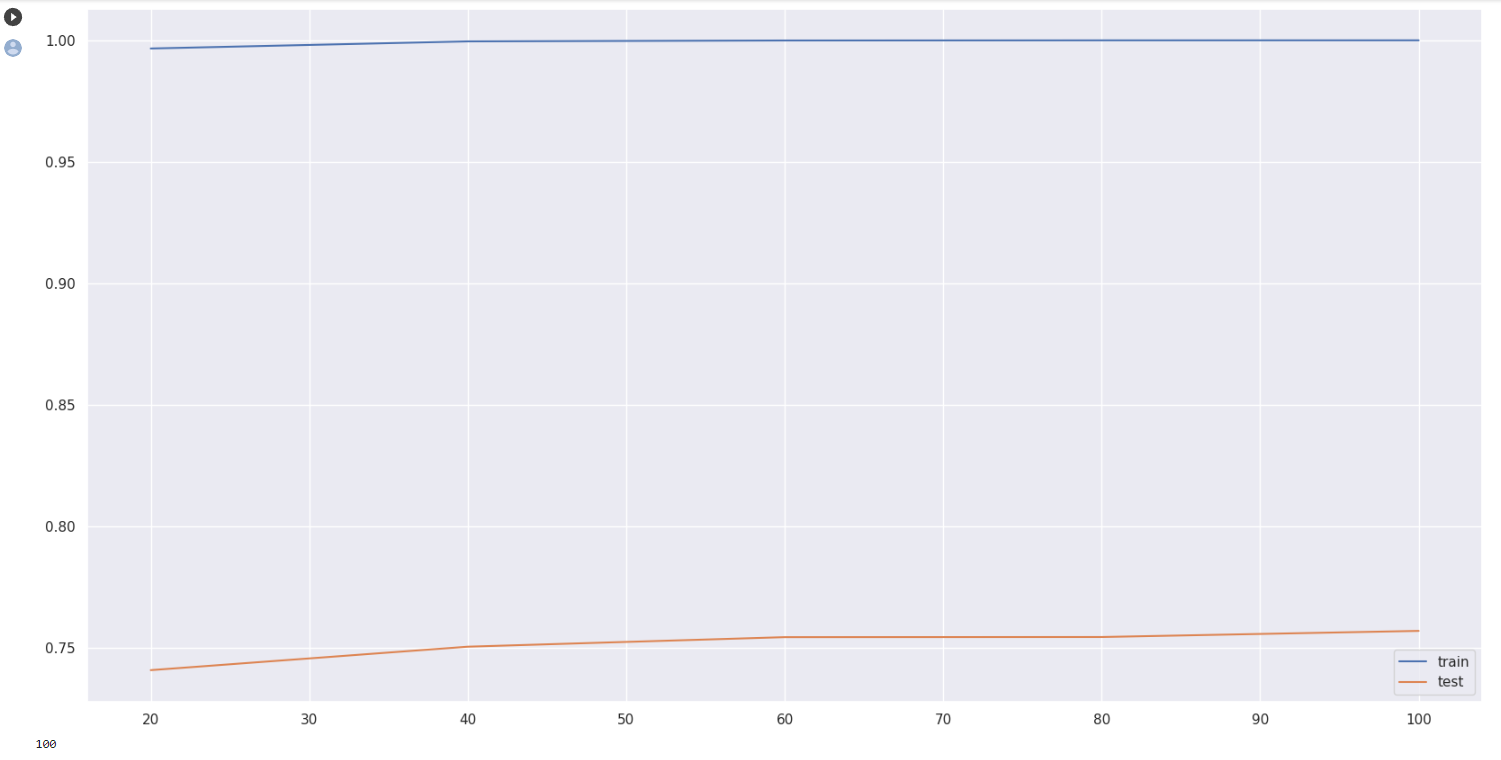
### **Ưu điểm & Nhược điểm của thuật toán**

* Ưu điểm:
* Độ chính xác cao: Random Forest là một phương pháp Ensemble Learning, kết hợp nhiều cây quyết định để tạo ra một mô hình phức tạp hơn, giúp cải thiện độ chính xác của mô hình. Điều này đặc biệt hữu ích khi sử dụng với các tập dữ liệu lớn và phức tạp.
* Khả năng xử lý được các tập dữ liệu lớn: Random Forest có thể xử lý các tập dữ liệu lớn với nhiều thuộc tính (features) và có thể được huấn luyện một cách song song (parallel) trên nhiều CPU.
* Linh hoạt trong việc chọn thuộc tính: Random Forest cho phép chọn ngẫu nhiên một số thuộc tính để phân tách cây, giúp tránh hiện tượng overfitting và giúp tăng độ đa dạng của các cây quyết định trong mô hình.
* Tránh được hiện tượng overfitting: Random Forest có khả năng giảm thiểu hiện tượng overfitting do việc kết hợp nhiều cây quyết định trong mô hình.
* Có thể được sử dụng để giải quyết cả các bài toán phân loại và dự đoán: Random Forest có thể được sử dụng cho cả các bài toán phân loại và dự đoán, và cũng có thể được áp dụng cho nhiều loại dữ liệu khác nhau.
* Nhược điểm:
* Tốc độ huấn luyện chậm hơn so với các thuật toán học máy khác: Do Random Forest phải huấn luyện nhiều cây quyết định, nên tốc độ huấn luyện sẽ chậm hơn so với các thuật toán học máy khác.
* Khó khăn trong việc diễn giải kết quả: Random Forest có nhiều cây quyết định được kết hợp với nhau để tạo thành một mô hình phức tạp hơn, việc diễn giải kết quả trở nên khó khăn hơn.
* Không phù hợp với các dữ liệu có tính nhất quán cao: Random Forest không phù hợp với các tập dữ liệu có tính nhất quán cao, nghĩa là các điểm dữ liệu có sự tương đồng lớn nhau, vì các cây quyết định sẽ cho ra kết quả tương tự nhau và dẫn đến sự trùng lặp trong mô hình.
* Có thể không thích hợp với các bài toán có số lượng lớn các lớp phân loại: Random Forest có thể không phù hợp với các bài toán có số lượng lớn các lớp phân loại, vì sẽ dẫn đến sự phân tán dữ liệu và làm giảm độ chính xác của mô hình.
* Tính khả diễn giải thấp: Vì Random Forest kết hợp nhiều cây quyết định để tạo ra một mô hình phức tạp hơn, nên độ khả diễn giải của mô hình thường thấp hơn so với các thuật toán học máy đơn giản hơn.

### **Thực hiện**

2.4.1 Tìm số lượng cây quyết định

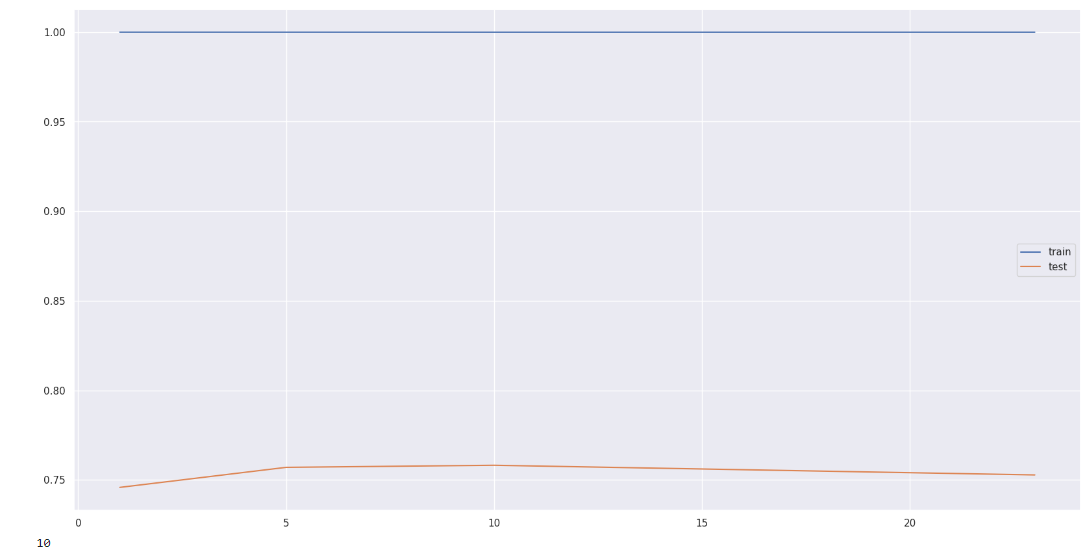




Kết quả trả về 100

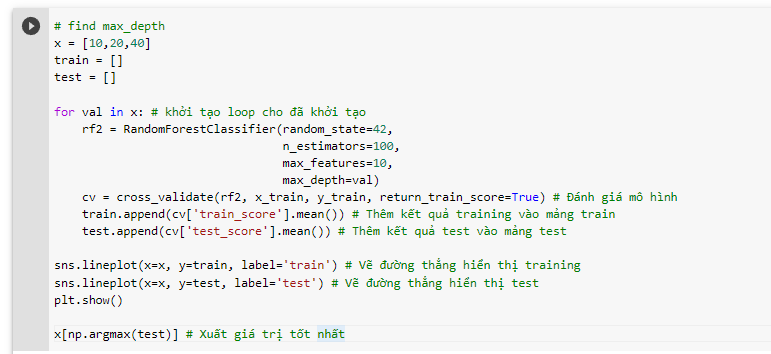
2.4.2 Xác định số lượng đặc trưng (features)

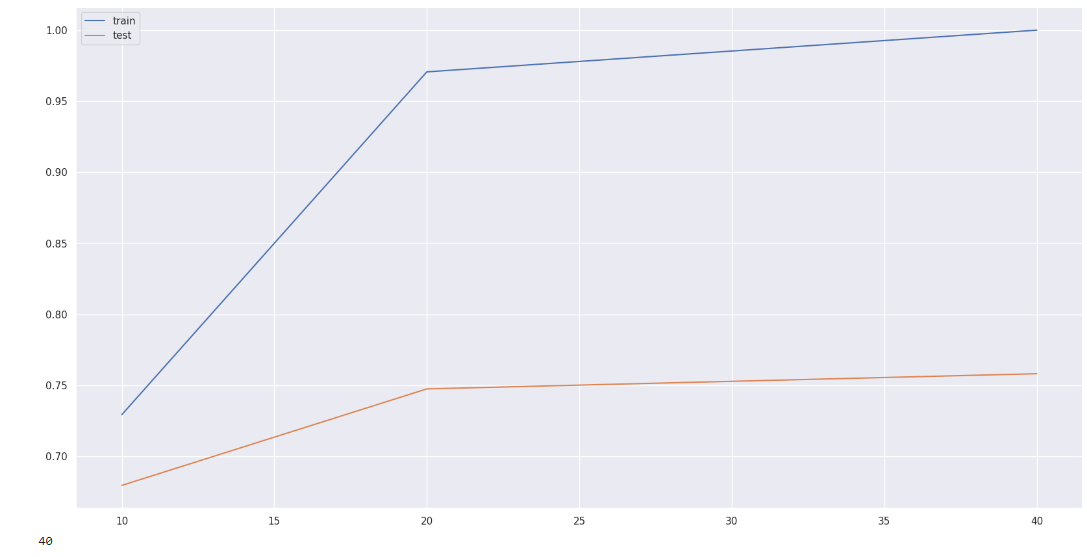




Kết quả trả về 10

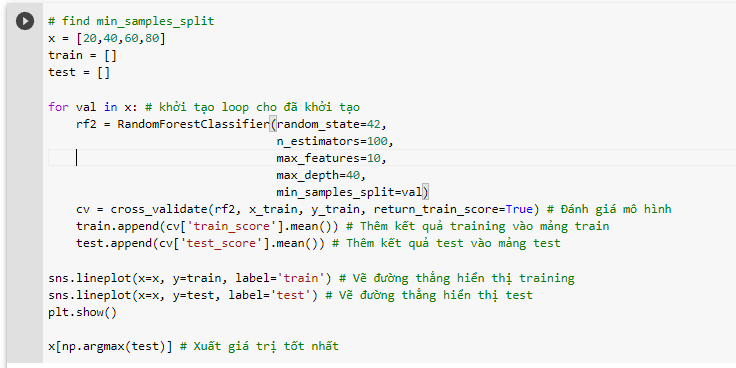
2.4.3 Xác định độ sâu tối đa

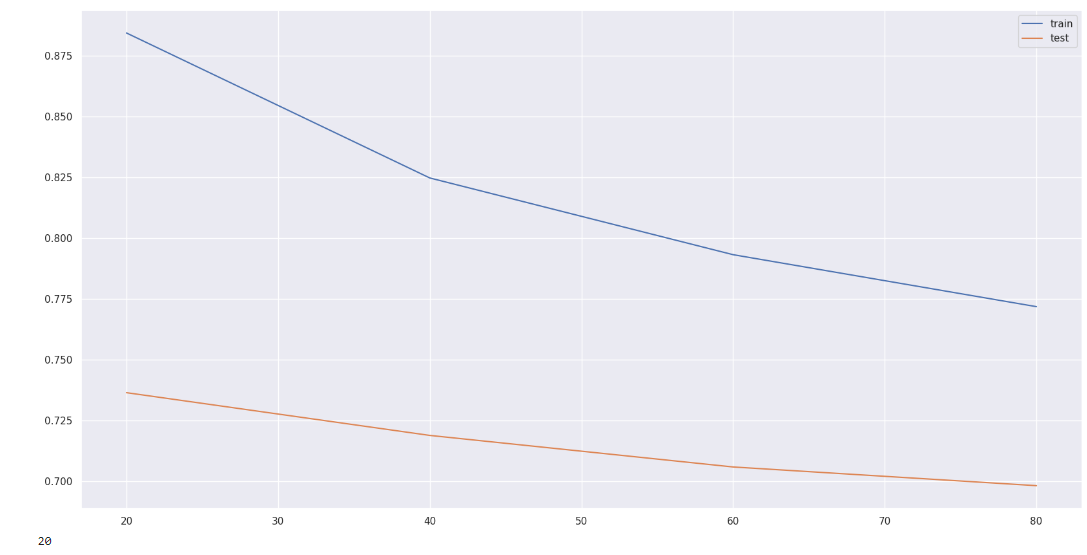




Kết quả trả về 40

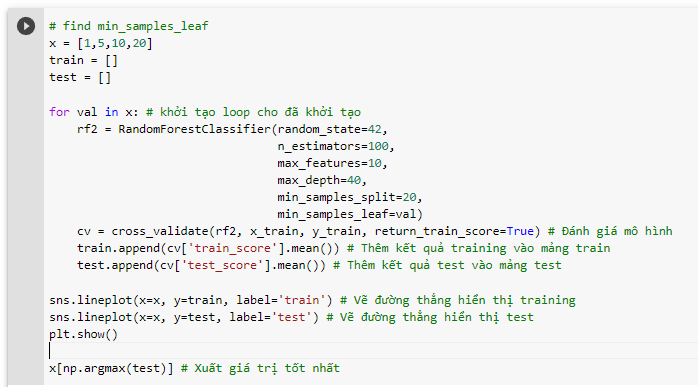
2.4.4 Xác định số lượng mẫu tối thiểu

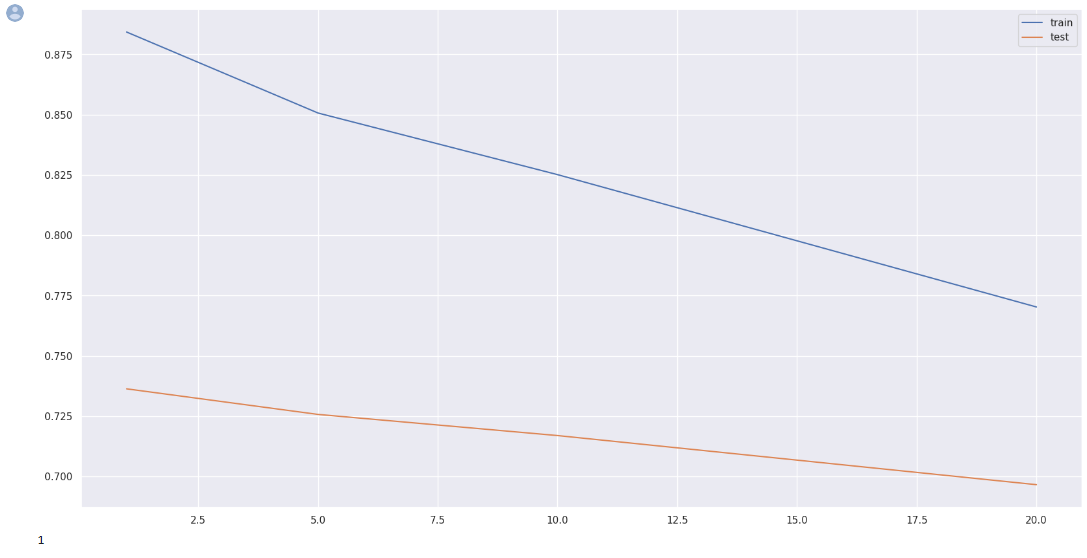




Kết quả trả về 20

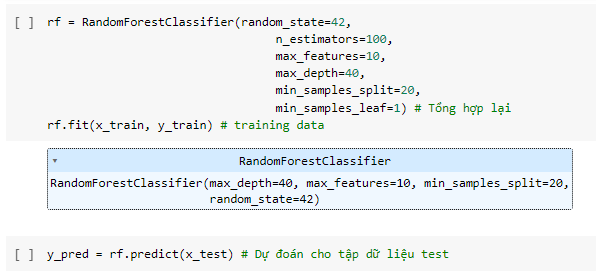
2.4.5 Xác định số lượng mẫu tối thiểu cần có trong một lá (leaf)



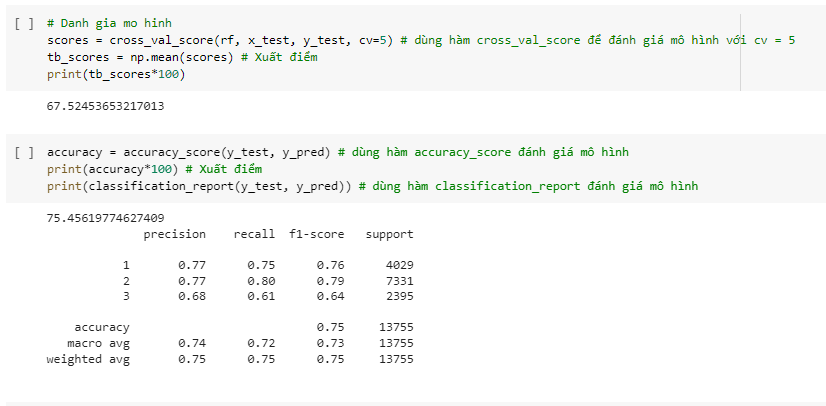


Kết quả trả về 1

2.4.6 Training data



2.4.7 Đánh giá mô hình



Trong thuật toán này, hàm accuracy\_score trả về độ chính xác tốt hơn với 75.45%

## **Thuật toán K-Nearest Neighbors.**

### **Giới thiệu**

Thuật toán K-Nearest Neighbors (K-NN) là một thuật toán học có giám sát được sử dụng để phân loại và dự đoán giá trị của một mẫu dữ liệu mới dựa trên các mẫu dữ liệu đã được gán nhãn trong tập dữ liệu huấn luyện.

Thuật toán KNN là một thuật toán đơn giản và dễ triển khai, đặc biệt là trong các bài toán phân loại dữ liệu. Tuy nhiên, nó cũng có một số hạn chế như cần phải lưu toàn bộ tập huấn luyện trong bộ nhớ, độ phức tạp tính toán và thời gian huấn luyện tăng lên khi số lượng mẫu dữ liệu tăng lên.

### **Mô tả thuật toán**

Thuật toán KNN thực hiện việc phân loại một mẫu dữ liệu mới bằng cách tìm K mẫu dữ liệu có khoảng cách gần nhất với nó trong tập dữ liệu huấn luyện. K là một số nguyên dương được định trước và được chọn sao cho số lượng mẫu dữ liệu có khoảng cách gần nhất với mẫu dữ liệu mới là đủ để đưa ra dự đoán chính xác.

- Các bước thực hiện của thuật toán K-NN như sau:

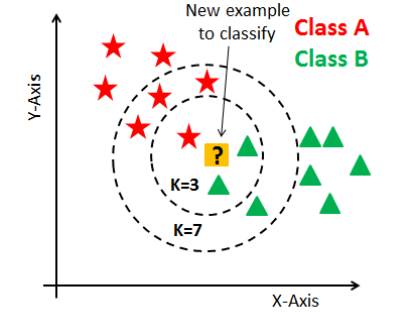
**Bước 1:** Xác định số lượng k lân cận gần nhất (tức là số lượng mẫu dữ liệu gần nhất) mà ta sẽ sử dụng để đưa ra quyết định cho mẫu dữ liệu mới.

**Bước 2:** Tính khoảng cách giữa mẫu dữ liệu mới và tất cả các mẫu dữ liệu trong tập huấn luyện bằng cách sử dụng một phương thức đo khoảng cách như khoảng cách Euclide, khoảng cách Manhattan, hay khoảng cách Cosine.

**Bước 3:** Lựa chọn k lân cận gần nhất từ tập huấn luyện bằng cách sắp xếp các mẫu dữ liệu dựa trên khoảng cách tính được và chọn ra k mẫu dữ liệu có khoảng cách nhỏ nhất với mẫu dữ liệu mới.

**Bước 4:** Phân loại mẫu dữ liệu mới bằng cách sử dụng đa số phiếu bầu của k mẫu dữ liệu gần nhất đã chọn. Nếu các lớp đếm được của các mẫu dữ liệu gần nhất là như nhau, ta có thể áp dụng phương thức lá phiếu (voting) ngẫu nhiên.

**Bước 5:** Dự đoán giá trị của mẫu dữ liệu mới bằng cách tính trung bình (mean) giá trị của k mẫu dữ liệu gần nhất đã chọn. Đây là phương pháp sử dụng cho bài toán hồi quy (regression).



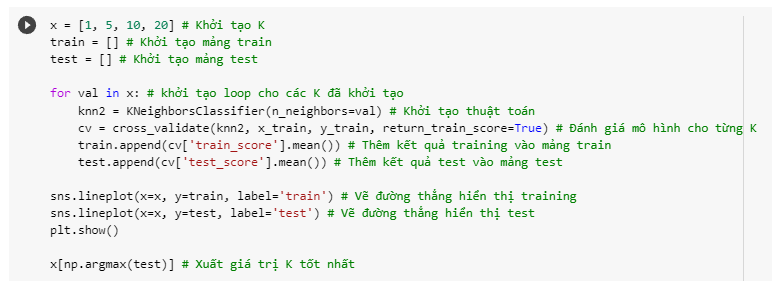
*Hình 19: Ví dụ thuật toán KNN*

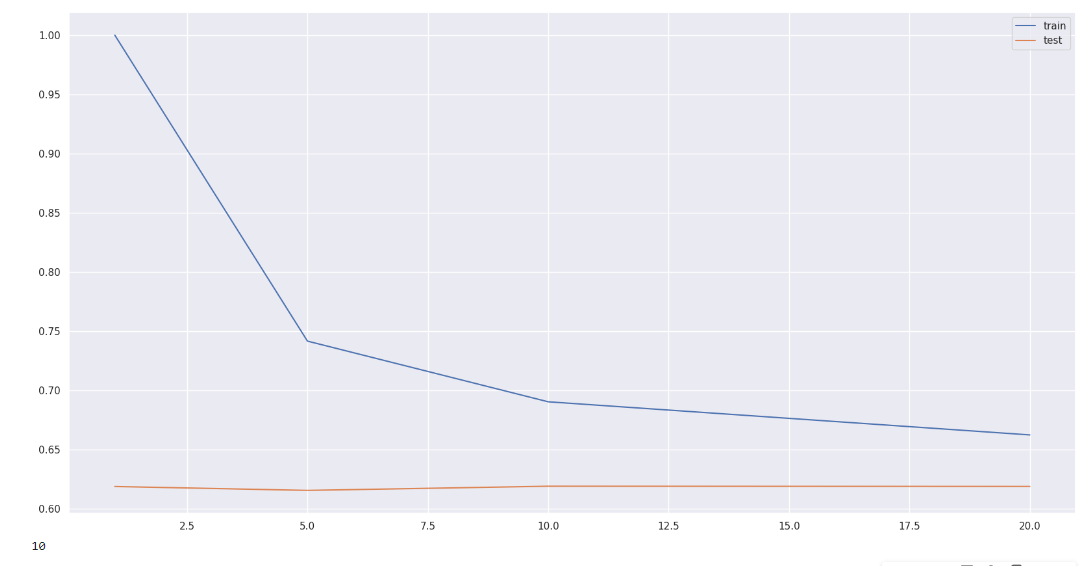
### **Ưu điểm & Nhược điểm của thuật toán**

* Ưu điểm:
* Đơn giản và dễ triển khai: KNN là một thuật toán đơn giản và dễ hiểu, không đòi hỏi kiến thức chuyên môn sâu về toán học hay lập trình nhiều.
* Không yêu cầu giả định về phân phối dữ liệu: KNN không yêu cầu phân phối dữ liệu tuân theo bất kỳ giả định nào, điều này cho phép KNN có thể được áp dụng cho nhiều loại dữ liệu khác nhau.
* Có thể được sử dụng cho cả bài toán phân loại và hồi quy: K-NN có thể được sử dụng cho cả bài toán phân loại và hồi quy, tùy thuộc vào cách ta đưa ra quyết định dựa trên k lân cận gần nhất.
* Nhược điểm:
* Độ phức tạp tính toán tăng khi kích thước tập dữ liệu tăng: K-NN cần phải tính toán khoảng cách giữa mẫu dữ liệu mới và tất cả các mẫu dữ liệu trong tập huấn luyện, điều này tạo ra độ phức tạp tính toán tăng khi kích thước tập dữ liệu tăng.
* Yêu cầu lưu toàn bộ tập huấn luyện trong bộ nhớ: K-NN cần phải lưu toàn bộ tập huấn luyện trong bộ nhớ, điều này làm tăng yêu cầu về bộ nhớ.
* Nhạy cảm với dữ liệu nhiễu: K-NN có thể bị ảnh hưởng bởi dữ liệu nhiễu, điều này có thể làm giảm độ chính xác của thuật toán.
* Việc lựa chọn k là một vấn đề quan trọng: Việc lựa chọn k là một vấn đề quan trọng đối với K-NN, nếu k quá nhỏ sẽ làm giảm độ chính xác của thuật toán, còn k quá lớn có thể làm tăng độ phức tạp tính toán và làm giảm khả năng tổng quát hóa của thuật toán.

### **Thực hiện**

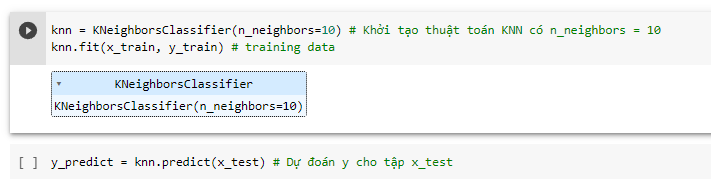
3.4.1 Tìm giá trị k tốt nhất



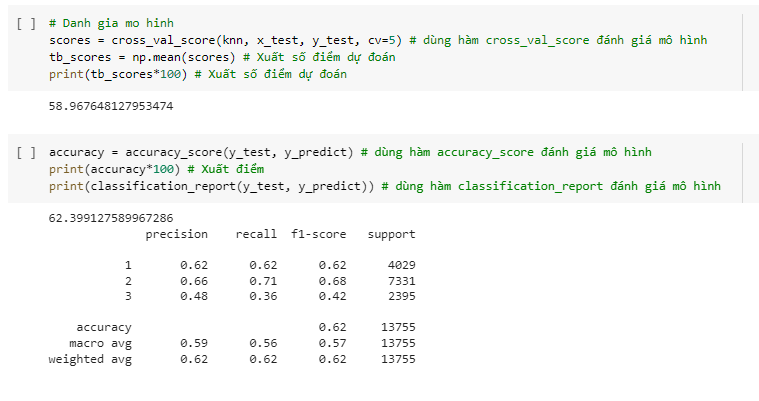


Kết quả k = 10

3.4.2 Training data



3.4.3 Đánh giá mô hình



Hàm accuracy\_score có độ chính xác cao hơn với 62.4%

# **Phần IV: Kết luận và hướng phát triển**

## **Kết luận**

Dự án phân loại Credit Score là một công việc quan trọng và phức tạp trong lĩnh vực tài chính, đòi hỏi sự hiểu biết sâu sắc về các khía cạnh tài chính và các thuật toán học máy. Trong dự án này, chúng em đã sử dụng thuật toán Random Forest để phân loại khách hàng dựa trên các yếu tố như thu nhập, số tiền vay và tuổi.

Kết quả cho thấy, thuật toán Random Forest đã cho độ chính xác cao và đáng tin cậy trong việc phân loại khách hàng, giúp đưa ra quyết định cho việc cấp vay một cách hiệu quả. Đồng thời, Random Forest cũng cho phép chọn ngẫu nhiên một số thuộc tính để phân tách cây, giúp tránh hiện tượng overfitting và giúp tăng độ đa dạng của các cây quyết định trong mô hình.

Tuy nhiên, Random Forest cũng có một số nhược điểm, bao gồm tốc độ huấn luyện chậm và khả năng diễn giải kết quả thấp.

1. **So sánh độ chính xác**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | Random Forest | K-Nearest Neighbors |
| cross\_val\_scores | 63.7077426390403 | 67.524536532170 | 58.9676481279534 |
| accuracy\_score | 65.0308978553253 | 75.456197746274 | 62.3991275899672 |

## **Hướng phát triển**

Trong tương lai, chúng em sẽ tiếp tục nghiên cứu và áp dụng các thuật toán học máy khác để tăng độ chính xác và khả diễn giải của mô hình. Tổng kết lại, dự án phân loại Credit Score đã giúp chúng em hiểu rõ hơn về các thuật toán học máy và áp dụng chúng vào thực tế để giúp các đơn vị tài chính đưa ra quyết định cho việc cấp vay một cách chính xác và hiệu quả.

**Lời cảm ơn**

Kính gửi cô Vũ Thị Hạnh,

Chúng em xin gửi lời cảm ơn chân thành đến cô vì sự hỗ trợ và giúp đỡ của cô trong dự án này. Những lời khuyên và sự chỉ dẫn của cô đã giúp chúng tôi hoàn thành dự án một cách hiệu quả và thành công.

Cô là người đã truyền đạt cho chúng em những kiến thức quý báu về lĩnh vực của mình. Sự tận tâm và nhiệt tình của cô trong việc hướng dẫn chúng em đã giúp chúng em hiểu rõ hơn về các vấn đề liên quan đến dự án và đưa ra các quyết định và giải pháp phù hợp nhất.

Chúng em rất biết ơn sự đóng góp của cô và sẽ luôn nhớ đến những điều mà cô đã giảng dạy cho chúng em. Chúng em tin rằng những kiến thức và kinh nghiệm mà cô truyền đạt sẽ giúp chúng em phát triển và thành công trong tương lai. Một lần nữa, chúng em xin chân thành cảm ơn cô vì sự giúp đỡ và hỗ trợ của cô trong dự án này. Chúc cô luôn thành công và hạnh phúc trong cuộc sống.

Trân trọng !

**Tài liệu tham khảo**

Th.S Vũ Thị Hạnh, (2021), Slide Bài giảng Khai Phá dữ liệu - Datamining.

<https://www.kaggle.com/datasets/parisrohan/credit-score-classification>

[Khai phá dữ liệu – Wikipedia tiếng Việt](https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u)