TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HCM

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ ĐIỆN, ĐIỆN TỬ**



**BÁO CÁO MÔN HỌC**

**NGÔN NGỮ LẬP TRÌNH PYTHON**

**TÊN ĐỀ TÀI**

**DỰ ĐOÁN VỠ NỢ ĐỐI VỚI KHOẢN VAY TIÊU DÙNG**

**GVHD:** Huỳnh Thanh Việt

**SVTH:**

066205007155\_Nguyễn Công Phúc

***Thành phố Hồ Chí Minh, ngày 28 tháng 4 năm 2025***

**Mục Lục**

[LỜI MỞ ĐẦU i](#_Toc196682427)

[LỜI CẢM ƠN ii](#_Toc196682428)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 1](#_Toc196682429)

[1.1. Lý do chọn đề tài 1](#_Toc196682430)

[1.2. Mục tiêu 1](#_Toc196682431)

[1.3. Tổng quan về dữ liệu 1](#_Toc196682432)

[1.4 . Kĩ thuật sử dụng 3](#_Toc196682433)

[CHƯƠNG 2: XỬ LÍ VÀ TRỰC QUAN HÓA DỮ LIỆU 4](#_Toc196682434)

[2.1. Khám phá dữ liệu ban đầu 4](#_Toc196682435)

[2.2. Tiền xử lý dữ liệu 6](#_Toc196682436)

[2.2.1. Thêm tiêu đề cho cột bị thiếu 6](#_Toc196682437)

[2.2.2. Xử lí giá trị thiếu trong mỗi cột 6](#_Toc196682438)

[2.2.3. Tách cột có nhiều giá trị trong 1 ô 8](#_Toc196682439)

[2.2.4. Xử lí dữ liệu trùng nhau 8](#_Toc196682440)

[2.2.5. Xử lí giá trị không hợp lệ 9](#_Toc196682441)

[2.2.6. Xử lí các cột không đồng nhất dữ liệu 9](#_Toc196682442)

[2.2.7. Xử lý các cột không đồng nhất kiểu dữ liệu 11](#_Toc196682443)

[2.2.8. Xóa khoảng trắng đầu, cuối cho các cột dữ liệu object 12](#_Toc196682444)

[2.2.9. Xử lý các ký tự không hợp lệ 12](#_Toc196682445)

[2.2.10. Loại bỏ các cột không cần thiết 13](#_Toc196682446)

[2.2.11. Xem lại thống kê mô tả 13](#_Toc196682447)

[2.2.12. Xử lý mối quan hệ logic giữa các cột dữ liệu 14](#_Toc196682448)

[2.2.13. Phân loại dữ liệu 17](#_Toc196682449)

[2.2.14. Kiểm tra dữ liệu ngoại lai 17](#_Toc196682450)

[2.3. Lưu dữ liệu vào CSDL 19](#_Toc196682451)

[2.4. Trực quan hóa dữ liệu 20](#_Toc196682452)

[2.4.1. Biểu đồ phân phối tỷ lệ vỡ nợ 20](#_Toc196682453)

[2.4.2. Biểu đồ phân phối thu nhập theo nhóm rủi ro 21](#_Toc196682454)

[2.4.3. Biểu đồ tình trạng sở hữu nhà theo tỷ lệ rủi ro 22](#_Toc196682455)

[2.4.4. Biểu đồ tình trạng sở hữu xe theo tỷ lệ rủi ro 23](#_Toc196682456)

[2.4.5. Biểu đồ phân bố tuổi theo tỷ lệ rủi ro 24](#_Toc196682457)

[2.4.6. Phân bố kinh nghiệm theo nhóm rủi ro 27](#_Toc196682458)

[2.4.7. Biểu đồ thể hiện nghề nghiệp có tỷ lệ vỡ nợ cao nhất 28](#_Toc196682459)

[2.4.8. Biểu đồ thể hiện thành phố có tỷ lệ vỡ nợ cao nhất 29](#_Toc196682460)

[2.4.9. Đề xuất giải pháp 30](#_Toc196682461)

[CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU ĐƯA VÀO MÔ HÌNH MÁY HỌC 31](#_Toc196682462)

[3.1. Chuẩn hóa dữ liệu và mã hóa đặc trưng 31](#_Toc196682463)

[3.2. Sử dụng các phương pháp kiểm định để chọn thuộc tính cần thiết cho mô hình 33](#_Toc196682464)

[3.2.1. Kiểm định Chi Square 33](#_Toc196682465)

[3.2.2. Kiểm định ANOVA F-Test 34](#_Toc196682466)

[3.3. Các đặc trưng quan trọng được chọn từ 2 phương pháp trên 35](#_Toc196682467)

[3.4. Đánh labeling 36](#_Toc196682468)

[3.5 Cân bằng dữ liệu 37](#_Toc196682469)

[3.6. Chia tập train và test 38](#_Toc196682470)

[3.7. Xây dựng mô hình để đưa vào máy học 38](#_Toc196682471)

[3.7.1. Lý do chọn mô hình 38](#_Toc196682472)

[3.7.2. Mô hình Decision Tree 38](#_Toc196682473)

[3.7.3. Mô hình RandomForest 39](#_Toc196682474)

[CHƯƠNG 4: ĐÁNH GIÁ CÁC MÔ HÌNH HỌC MÁY 41](#_Toc196682475)

[4.1. Đánh giá mô hình Decision tree 41](#_Toc196682476)

[4.2. Đánh giá mô hình Random Forests 42](#_Toc196682477)

[4.3. So sánh 2 mô hình 44](#_Toc196682478)

[4.4. Tổng kết 44](#_Toc196682479)

[CHƯƠNG 5: KẾT LUẬN 46](#_Toc196682480)

[TÀI LIỆU THAM KHẢO 47](#_Toc196682481)

# **LỜI MỞ ĐẦU**

Với sự phát triển nhanh chóng của nền kinh tế, các sản phẩm tài chính tiêu dùng, đặc biệt là các khoản vay tiêu dùng, ngày càng trở nên phổ biến. Tuy nhiên, trong bối cảnh này, một vấn đề nghiêm trọng mà các tổ chức tín dụng và các cá nhân phải đối mặt là khả năng vỡ nợ của người vay. Dự đoán vỡ nợ đối với khoản vay tiêu dùng không chỉ giúp giảm thiểu rủi ro tài chính mà còn đóng vai trò quan trọng trong việc bảo vệ quyền lợi của các bên liên quan.

Báo cáo này sẽ đi sâu vào việc phân tích và dự đoán khả năng vỡ nợ của các khoản vay tiêu dùng thông qua các phương pháp phân tích dữ liệu và mô hình dự báo. Mục tiêu là cung cấp cái nhìn tổng quan về các yếu tố ảnh hưởng đến khả năng vỡ nợ, từ đó đưa ra các chiến lược giúp các tổ chức tín dụng quản lý rủi ro hiệu quả hơn. Đồng thời, báo cáo cũng sẽ đề cập đến tầm quan trọng của việc áp dụng các công nghệ tiên tiến trong việc phân tích và dự báo các tình huống vỡ nợ trong lĩnh vực tín dụng tiêu dùng.

# **LỜI CẢM ƠN**

Kính gửi thầy Huỳnh Thanh Việt,

Em xin chân thành cảm ơn thầy đã giảng dạy và truyền đạt kiến thức quý báu trong suốt quá trình học môn Ngôn ngữ lập trình Python. Những bài học và kinh nghiệm mà thầy chia sẻ đã giúp em không chỉ hiểu sâu hơn về môn học mà còn phát triển nhiều kỹ năng cần thiết cho tương lai.

Em rất trân trọng sự tận tâm và nhiệt huyết của thầy trong việc hướng dẫn và hỗ trợ em. Những buổi học thú vị và những giờ thảo luận sôi nổi đã để lại ấn tượng sâu sắc trong lòng. Hy vọng rằng trong tương lai, em sẽ có cơ hội học hỏi thêm từ thầy. Chúc thầy sức khỏe và thành công trong sự nghiệp giảng dạy!

Trân trọng.

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

## **1.1. Lý do chọn đề tài**

Bài toán dự đoán khoản vay dựa trên hành vi khách hàng (Loan Prediction Based on Customer Behavior) là một bài toán được quan tâm rất nhiều bởi các ngân hàng và mọi doanh nghiệp liên quan đến hoạt động giao dịch và cho vay. Hoạt động vay của khách hàng là một trong những nguồn thu lớn, ảnh hưởng trực tiếp đến doanh thu và lợi nhuận của các ngân hàng và doanh nghiệp tài chính.

Thông qua việc phát triển bài toán về dự đoán khả năng vỡ nợ của khoản vay bởi khách hàng, doanh nghiệp có thể có các chiến lược cụ thể để từ đó chuẩn bị, tiếp cận, giám sát, đề phòng các trường hợp không mong muốn có thể xảy ra, từ đó giảm thiểu chi phí cho các chiến dịch dư thừa và không hiệu quả.

Nhận thấy sự cần thiết và tầm quan trọng của vấn đề, nhóm quyết định phát triển và tìm hiểu bài toán nêu trên.

## **1.2. Mục tiêu**

* Làm sạch và xử lý dữ liệu bằng ngôn ngữ python
* Trực quan hóa dữ liệu để biểu diễn thông tin dưới dạng hình ảnh, biểu đổ giúp hiểu rõ xu hướng, mẫu hình và mối quan hệ trong dữ liệu, từ đó hỗ trợ đưa ra những chiến lược linh hoạt và hiệu quả.
* Chuẩn hóa các thuộc tính số và mã hóa các thuộc tính phân loại nhằm chuyển đổi dữ liệu về dạng phù hợp với các thuật toán học máy.
* Xây dựng mô hình học máy nhằm dự đoán vỡ nợ đối với khoản vay tiêu dùng.
* Đánh giá hiệu suất mô hình và đề xuất các giải pháp tối ưu hóa

## **1.3. Tổng quan về dữ liệu**

Nguồn dữ liệu: [Loan Prediction Based on Customer Behavior | Kaggle](https://www.kaggle.com/datasets/subhamjain/loan-prediction-based-on-customer-behavior)

Graphical user interface, text, application

Description automatically generated

Thông tin về tác giả

A picture containing shape

Description automatically generated

* Cập nhật lần cuối vào tháng 8 năm 2021
* Hơn 56 150 lượt xem, 6789 lượt tải về, 159 đánh giá tốt

Mô tả dữ liệu : Dữ liệu có 13 cột và 252 128 dòng

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** |
| Id | Id | int |
| Income | Thu nhập | int |
| Age | Tuổi | int |
| Experience | Kinh nghiệm chuyên môn | int |
| Married/Single | Tình trạng hôn nhân | string |
| House\_Ownership | Tình trạng sở hữu nhà | string |
| Car\_Ownership | Tình trạng sỡ hữu xe hơi | string |
| Profession | Nghề nghiệp | string |
| CURRENT\_JOB\_YRS | Kinh nghiệm trong công việc  hiện tại | int |
| CURRENT\_HOUSE\_YRS | Thời gian cư trú ở nơi hiện tại | int |
| Risk\_Flag | Tình trạng vỡ nợ | String |
| Extra\_Notes | Ghi chú | string |
| CITY | Thành phố cư trú | string |

## **1.4 . Kĩ thuật sử dụng**

Công cụ khai thác dữ liệu: Google Colab.

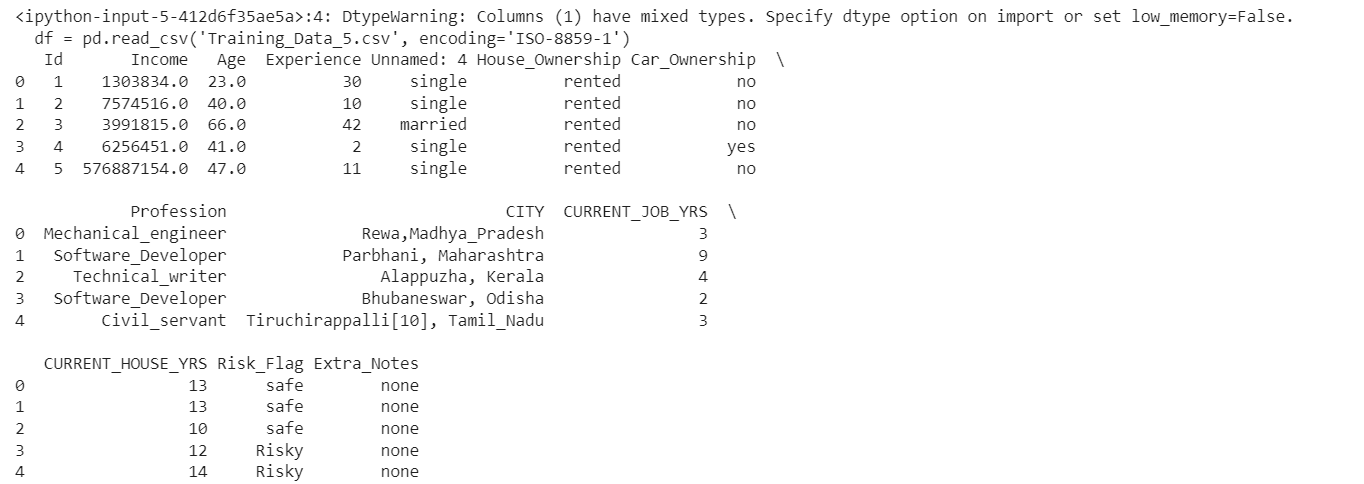
Phiên bản Python: 3.10.5.

Thư viện: Numpy, Pandas, , Seaborn, sklearn, imblearn, catboost, matplotlib

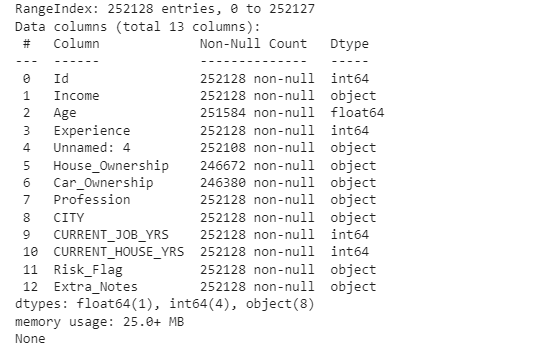
Mô hình sử dụng: RandomForest , CatBoost

# **CHƯƠNG 2: XỬ LÍ VÀ TRỰC QUAN HÓA DỮ LIỆU**

## **2.1. Khám phá dữ liệu ban đầu**

A computer screen with text

AI-generated content may be incorrect.

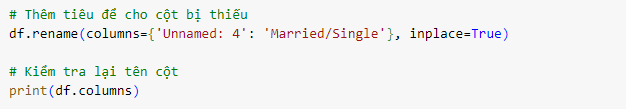


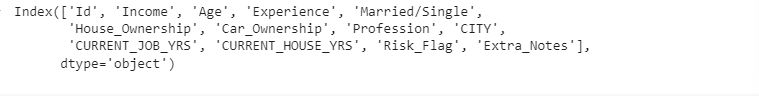
**A screenshot of a computer

AI-generated content may be incorrect.**

## **2.2. Tiền xử lý dữ liệu**

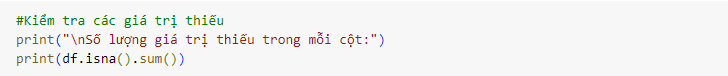
### **2.2.1. Thêm tiêu đề cho cột bị thiếu**

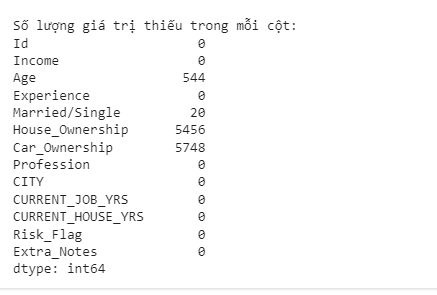




### **2.2.2. Xử lí giá trị thiếu trong mỗi cột**

**-** Kiểm tra các giá trị thiếu:





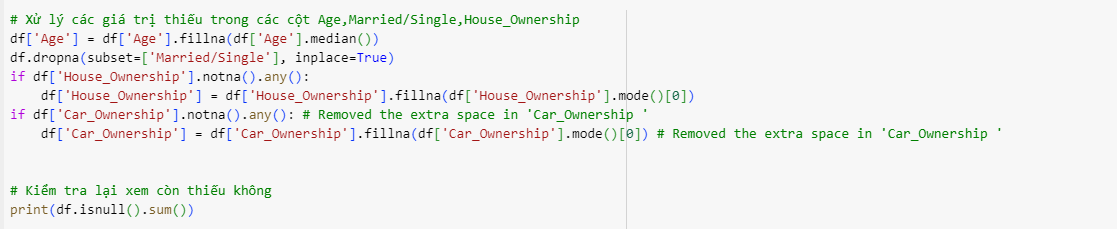
- Phương pháp xử lí:

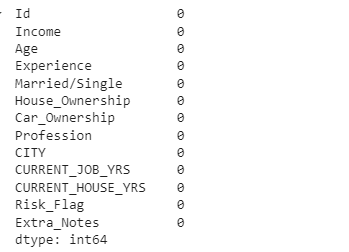
+ Cột Age: Có 544 giá trị thiếu. Vì đây là dữ liệu dạng số liên tục, nên ta thay thế giá trị thiếu bằng trung vị (median) để tránh bị ảnh hưởng bởi các giá trị ngoại lệ (vì trung vị không bị lệch nhiều như giá trị trung bình).

+ Cột Married/Single : Có 20 giá trị thiếu. Cột này mang tính chất phân loại (categorical) (chỉ có 2 giá trị Married hoặc Single). Tuy nhiên, vì số lượng thiếu ít (20 dòng, chiếm 0.00793%) vì thế ta tiến hành xóa những dòng bị thiếu để đảm bảo dữ liệu sau này chính xác hơn.

+ Cột House\_Ownership: Có 5.456 giá trị thiếu: Đây cũng là dữ liệu phân loại (Owned, Rented, hoặc Other). Vì số lượng thiếu nhiều, nếu xóa hết sẽ làm mất rất nhiều dữ liệu.  
→ Do đó, thay thế giá trị thiếu bằng mode (giá trị xuất hiện nhiều nhất) để giữ lại càng nhiều dữ liệu gốc càng tốt.

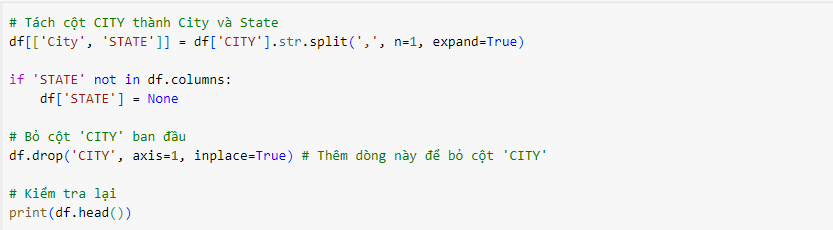
+ Cột Car\_Ownership: Có 5.748 giá trị thiếu: Tương tự như House\_Ownership, vì số lượng thiếu rất lớn nên cũng thay bằng mode để bảo toàn kích thước dữ liệu.

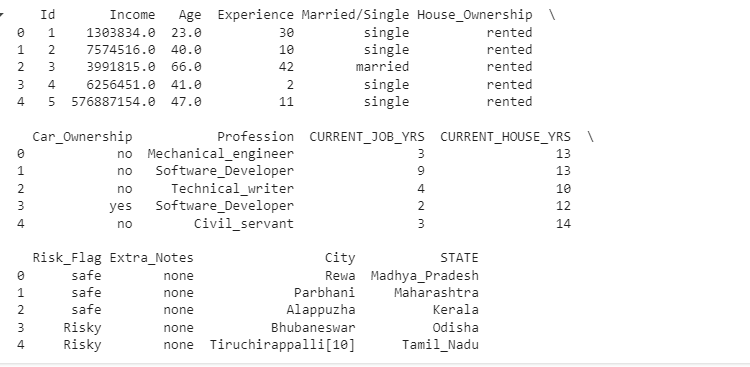




### **2.2.3. Tách cột có nhiều giá trị trong 1 ô**

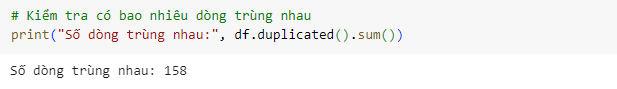
- Phương pháp xử lí: Tiến hành tách cột 'CITY' ban đầu thành hai cột mới là 'City' và 'STATE' bằng cách sử dụng dấu chấm (,) làm ký tự phân tách. Sau khi tạo ra hai cột mới và gán giá trị tương ứng cho chúng, cột 'CITY' gốc sẽ được xóa khỏi bảng dữ liệu để tránh trùng lặp và đảm bảo dữ liệu được chuẩn hóa.





### **2.2.4. Xử lí dữ liệu trùng nhau**

- Kiểm tra các dòng trùng nhau

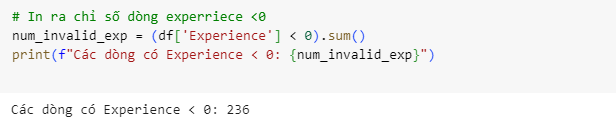


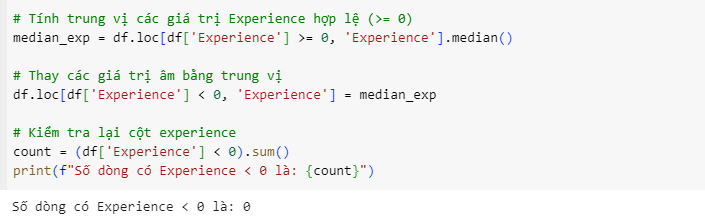
- Phương pháp xử lí: Xóa các dòng trùng nhau, giữ lại dòng đầu tiên



### **2.2.5. Xử lí giá trị không hợp lệ**

- In ra các dòng trong cột Experince <0:

****

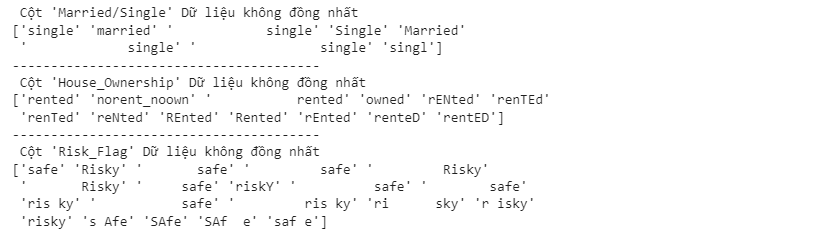
- Phương pháp xử lí: Tính trung vị để thay thế cho các giá trị không hợp lệ

### **2.2.6. Xử lí các cột không đồng nhất dữ liệu**

- Kiểm tra các cột có kiểu dữ liệu object không đồng nhất dữ liệu:



- Kết quả: Các cột Married/Single, House\_Ownerhsip, Risk\_Flag không đồng nhất dữ liệu, có nhiều giá trị lẫn lộn



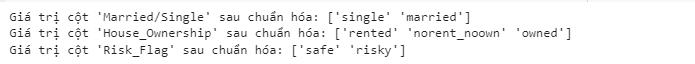
- Phương pháp xử lí: Chuẩn hóa chuỗi về chữ thường, bỏ khoảng trắng đầu cuối, xóa khoảng trắng giữa các chữ, sau đó áp dụng chuẩn hóa cho các cột Married/Single, House\_Ownerhsip, Risk\_Flag

**A computer code with many colorful text

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**



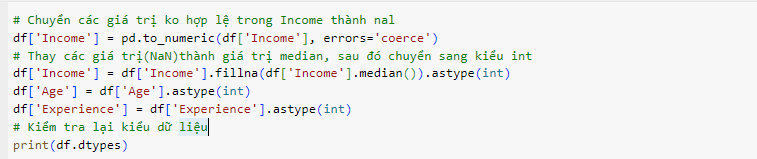
### **2.2.7. Xử lý các cột không đồng nhất kiểu dữ liệu**

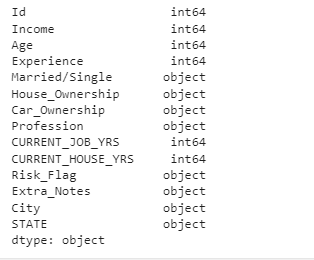
- Phương pháp xử lí:

+ Cột Income:

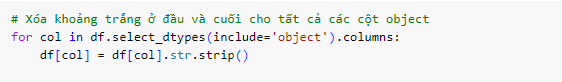
* Do trong cột Income có chứa nhiều kiểu dữ liệu khác nhau (số và chuỗi), nên trước tiên tiến hành chuyển các giá trị không hợp lệ (dạng chuỗi) thành NaN bằng cách dùng pd.to\_numeric(errors='coerce').
* Sau đó, thay thế các giá trị NaN bằng trung vị (median) của cột Income để đảm bảo tính ổn định dữ liệu.
* Cuối cùng, chuyển toàn bộ cột Income về kiểu dữ liệu int để đồng nhất và dễ xử lý.

+ Chuyển cả hai cột Age và Experience sang kiểu dữ liệu int nhằm đảm bảo dữ liệu đúng dạng số nguyên, phù hợp với bản chất thực tế của các thông tin này.



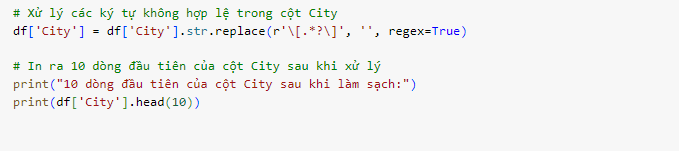


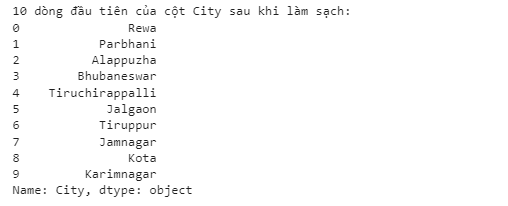
### **2.2.8. Xóa khoảng trắng đầu, cuối cho các cột dữ liệu object**



### **2.2.9. Xử lý các ký tự không hợp lệ**

- Phương pháp xử lí: Trong cột City xuất hiện các ký tự không hợp lệ, ví dụ như "Rewa[10]", "Delhi[5]", v.v.. Tiến hành làm sạch dữ liệu bằng cách xóa toàn bộ các ký tự không hợp lệ bằng phương thức str.replace() với biểu thức chính quy (regex)

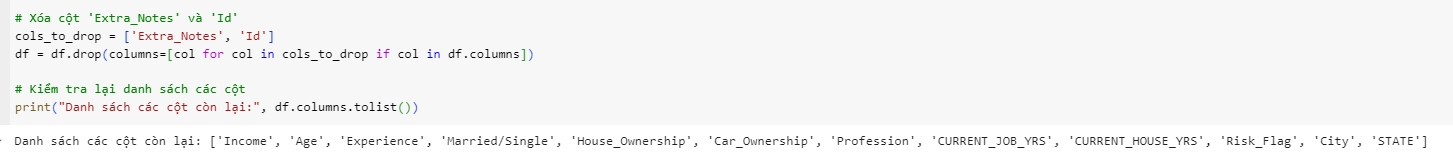
****



### **2.2.10. Loại bỏ các cột không cần thiết**

Các cột như 'Extra\_Notes' và 'Id' không mang giá trị phân tích:

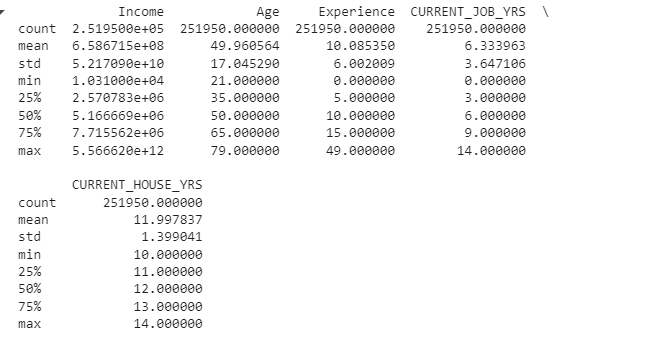
* 'Id' chỉ là mã định danh duy nhất cho từng dòng dữ liệu, không có ý nghĩa thống kê hay ảnh hưởng đến mô hình phân tích.
* 'Extra\_Notes' chứa thông tin không liên quan hoặc không có giá trị sử dụng trong phân tích dữ liệu hiện tại.

****

### **2.2.11. Xem lại thống kê mô tả**

**A close up of words

AI-generated content may be incorrect.**

****

### **2.2.12. Xử lý mối quan hệ logic giữa các cột dữ liệu**

**- Kiểm tra và xử lý lỗi giữa cột Experience và Age:**

+ Lọc ra các dòng Experience >= Age.

+ Xóa toàn bộ các dòng dữ liệu có Experience >= Age ra khỏi bộ dữ liệu. Vì lỗi chỉ có 9 dòng, nó chỉ chiếm khoảng 0.0035% trên tổng số 252,128 dòng, nên việc loại bỏ những dòng này không gây ảnh hưởng đáng kể đến chất lượng và độ đại diện của tập dữ liệu.

+ Sau khi xóa, kiểm tra lại số dòng còn lại để đảm bảo rằng không còn dòng lỗi nào.

****

**- Kiểm tra điều kiện Age - Experience < 18 và xử lí:**

**+** Lọc ra các dòng có điều kiện Age - Experience < 18

+ Tiến hành điều chỉnh giá trị Experience nhằm đảm bảo logic hợp lý. Cụ thể, cập nhật lại Experience = Age - 18 cho những dòng vi phạm. Vì

* Số dòng lỗi nhiều (32,800 dòng), ảnh hưởng độ đại diện và giảm độ tin cậy khi phân tích hoặc huấn luyện mô hình.
* Không thể gán bằng 0 hoặc null vì cột CURRENT\_JOB\_YRS đang > 0, nên nếu làm vậy sẽ phá vỡ tính logic giữa số năm làm việc hiện tại và tổng kinh nghiệm.
* Không thể thay thế bằng trung vị của cột Experience vì trung vị cột (=10) nên khi thay thế thì vẫn sai logic giữa cột Age và Experience ( Age-Experience <18)
* Việc điều chỉnh giúp giữ lại dữ liệu và đảm bảo tính hợp lý, vì người đi làm thường bắt đầu từ 18 tuổi.
* Khó xây dựng mô hình dự đoán Experience (hồi quy dựa trên Age, Education, Job Role...)=> Phức tạp, tốn công xây dựng mô hình riêng và kiểm định độ chính xác

**+** Sau khi điều chỉnh, kiểm tra lại số dòng còn lại để đảm bảo rằng không còn dòng lỗi nào.



**- Kiểm tra và xử lí lỗi giữa CURRENT\_JOB\_YRS và Experience**

+ Lọc ra các dòng CURRENT\_JOB\_YRS > Experience

+ Tiến hành xử lí gán CURRENT\_JOB\_YRS = Experience. Vì

* Không thể xóa các dòng lỗi vì số lượng lên tới 15 965 dòng (~6.33% tổng dữ liệu).  
  → Nếu xóa, sẽ làm mất dữ liệu nhiều, ảnh hưởng độ đại diện và độ tin cậy của tập dữ liệu.
* Không thể gán bằng 0 hoặc Null cho Experience, vì:

+ Những dòng đó có CURRENT\_JOB\_YRS > 0, tức là họ đang có kinh nghiệm làm việc thực tế.

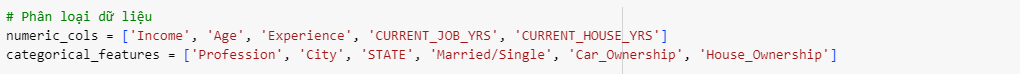
+ Nếu gán Experience = 0 hoặc null, thì sẽ mâu thuẫn: làm việc nhiều năm nhưng tổng kinh nghiệm lại bằng 0 → dữ liệu bị phi lý.

* Không thể thay thế bằng trung vị của cột Experience vì trung vị cột (=10) nên khi thay thế thì vẫn sai logic giữa cột Experience và CURRENT\_JOB\_YRS (CURRENT\_JOB\_YRS> Experience)
* …..

+ Sau khi điều chỉnh, kiểm tra lại số dòng còn lại để đảm bảo rằng không còn dòng lỗi nào.



### **2.2.13. Phân loại dữ liệu**

****

### **2.2.14. Kiểm tra dữ liệu ngoại lai**

**Lý do chọn phương pháp IQR:**

Phân phối dữ liệu:

-Income**:** Mean (6.59e8) > Median (5.17e6), khoảng từ median đến max (5.57e12 - 5.17e6 ≈ 5.57e12) cực lớn so với từ min đến median (5.17e6 - 1.03e4 ≈ 5.16e6)=> Dữ liệu lệch phải

- Age: Mean (49.96) ≈ Median (50): Gần đối xứng

- Experience: Mean (10,08) ≈ Median (10), khoảng từ median đến max (49 - 10 = 39) lớn hơn từ min đến median (10 - 0 = 0). => Lệch phải….

=> Vì những thuộc tính có phân phối hỗn hợp nên dùng phương pháp IQR để kiểm tra dữ liệu ngoại lai thay vì Z-score

**Cơ sở lý thuyết**

- IQR (Interquartile Range) là khoảng tứ phân vị, được tính bằng hiệu số giữa phân vị 75% (Q3) và phân vị 25% (Q1) của tập dữ liệu:

+ Phù hợp với dữ liệu có phân phối hỗn hợp

+ Tập trung vào vùng dữ liệu trung tâm, không bị ảnh hưởng bởi giá trị cực đại

+ Phù hợp với tập dữ liệu lớn

+ Tính toán nhanh mà không cần mô hình phức tạp

+ Xác định các giá trị bất thường trong một cột mà không cần xem xét mối quan hệ giữa các cột

A computer screen shot of a code

AI-generated content may be incorrect.

Sau khi áp dụng phương pháp IQR (Interquartile Range), kết quả phát hiện dữ liệu ngoại lai như sau:

- Cột Income: Có 7.790 giá trị ngoại lai, chiếm khoảng 3,09% tổng dữ liệu. Dữ liệu này có phân phối lệch phải rõ rệt, do đó việc loại bỏ các giá trị cực lớn giúp giảm ảnh hưởng đến mô hình học máy.

- Cột Experience: Có 3 giá trị ngoại lai, không đáng kể so với tổng thể. Phân phối cũng có xu hướng lệch phải nhẹ.

A white background with black text

AI-generated content may be incorrect.

**Xử lý ngoại lai**

Phương pháp xử lý: Các ngoại lệ được thay thế bằng trung vị. Vì :

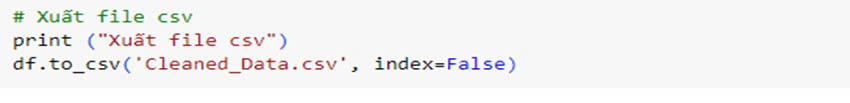
* Trung vị (median) là một giá trị mạnh mẽ (robust) đối với phân phối lệch, không bị ảnh hưởng bởi các giá trị cực đại như trung bình (mean).
* Cột Income có phân phối lệch phải mạnh (mean = 6.59e8, median = 5.17e6), nên trung vị đại diện tốt hơn cho giá trị trung tâm của dữ liệu.
* Thay thế ngoại lai bằng trung vị giúp giữ dữ liệu trong phạm vi hợp lý mà không làm thay đổi đáng kể phân phối tổng thể.
* Phương pháp này đơn giản, dễ triển khai, và phù hợp khi số lượng ngoại lai không quá lớn (3.09%).

**A screenshot of a computer program

AI-generated content may be incorrect.**

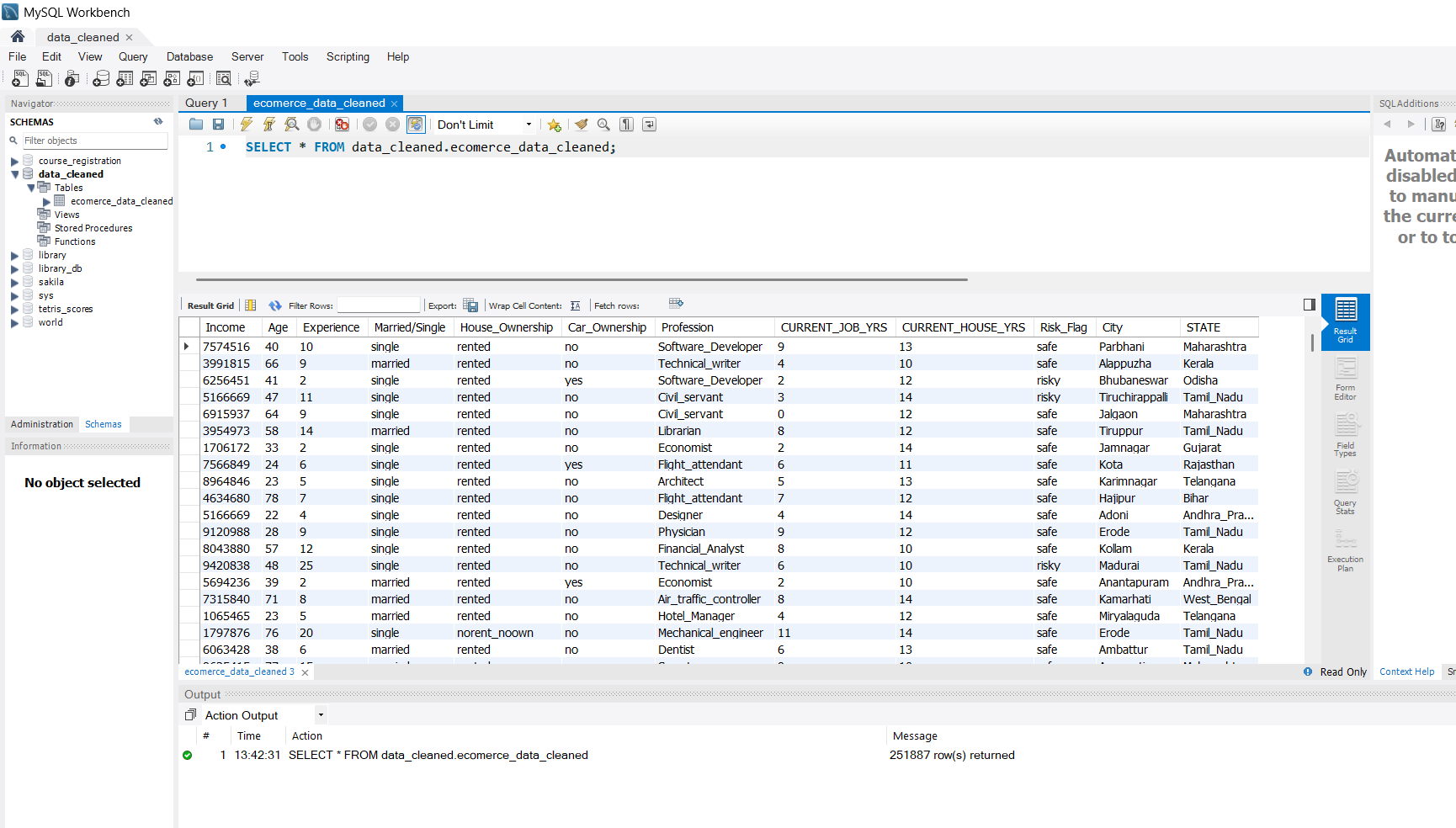
## **2.3. Lưu dữ liệu vào CSDL**

**- Xuất file đã làm sạch**

****

**- Tiến hành đưa file lên python để lưu vào SQL**

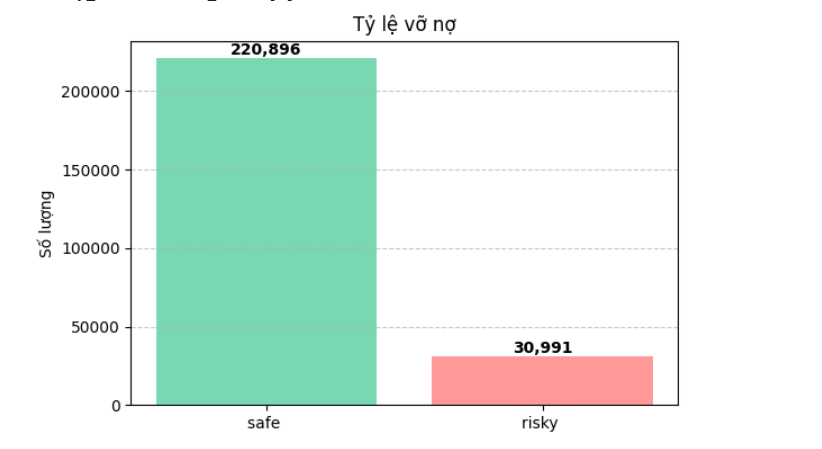
****



## **2.4. Trực quan hóa dữ liệu**

### **2.4.1. Biểu đồ phân phối tỷ lệ vỡ nợ**

****

****

**Hình 1: Biểu đồ phân phối tỷ lệ vỡ nợ**

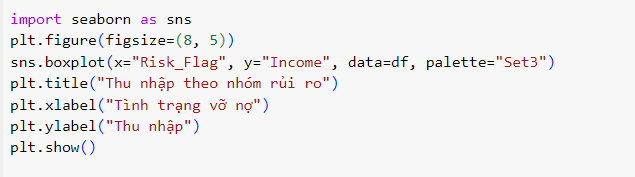
**Kết luận:**

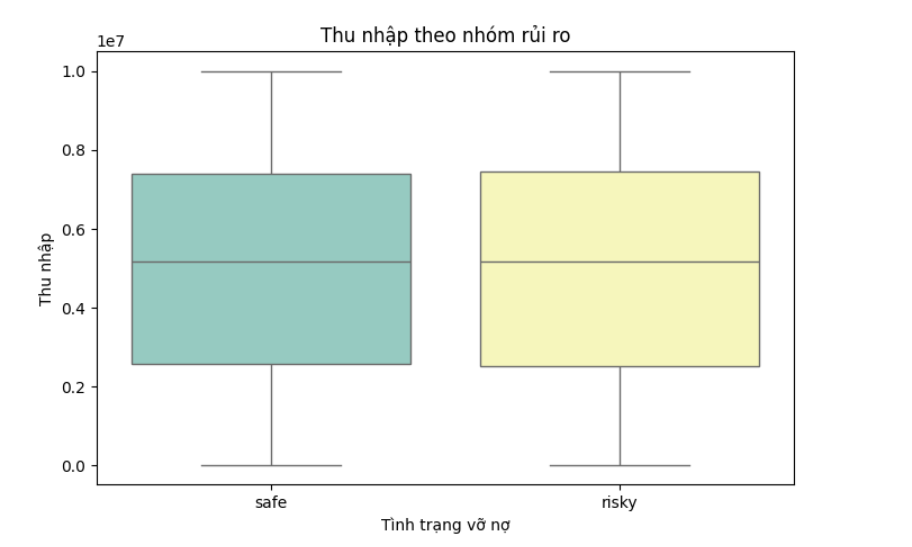
Có 30,991 người đang trong tình trạng có rủi ro vỡ nợ ( chiếm ~12.3%).

Không những thế 220,896 người đang ở trong ngưỡng an toàn ( chiếm ~ 87,7%).

=> Những người đang ở trong ngưỡng an toàn gấp gần 7 lần so với những người có tình trạng vỡ nợ.

### **2.4.2. Biểu đồ phân phối thu nhập theo nhóm rủi ro**

****

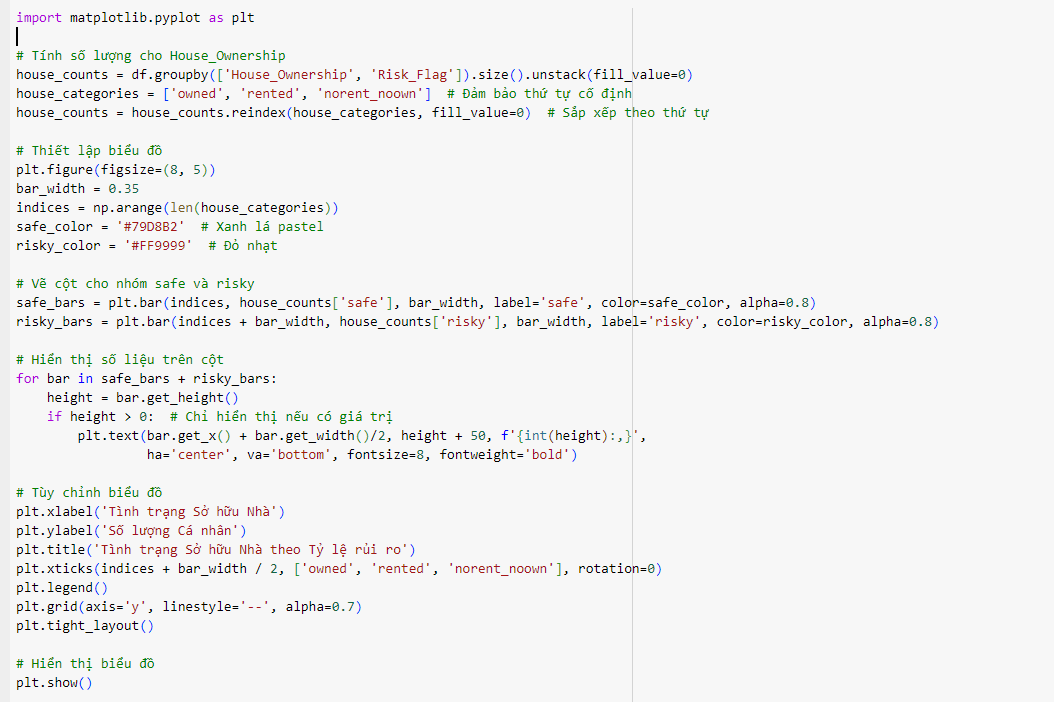
****

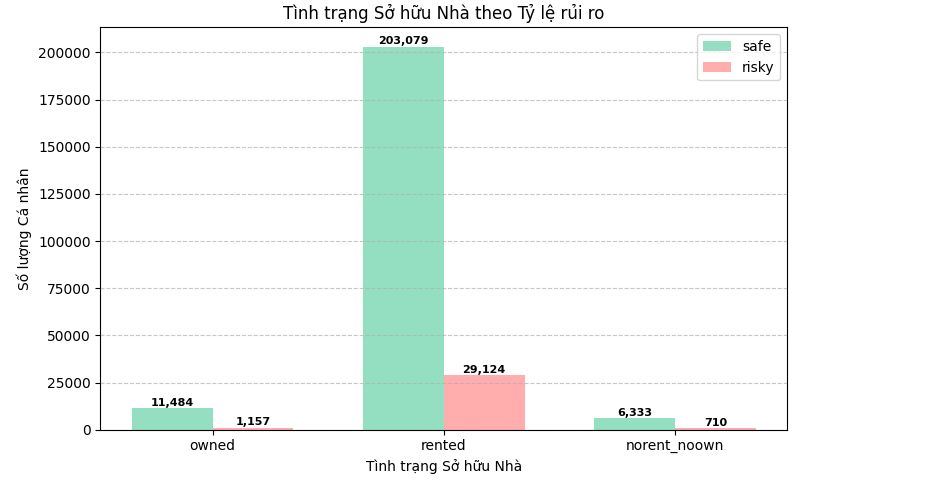
**Kết luận:**

Cả 2 nhóm đều có tỉ lệ an toàn và rủi ro trong mức từ 0.25\*1e7 đến 0.75\*1e7 (1e7 = 1\*10^7)

Từ đó có thể kết luận thì tỉ lệ an toàn và rủi ro không được quyết định rõ ràng thông qua thu nhập.

### **2.4.3. Biểu đồ tình trạng sở hữu nhà theo tỷ lệ rủi ro**

****

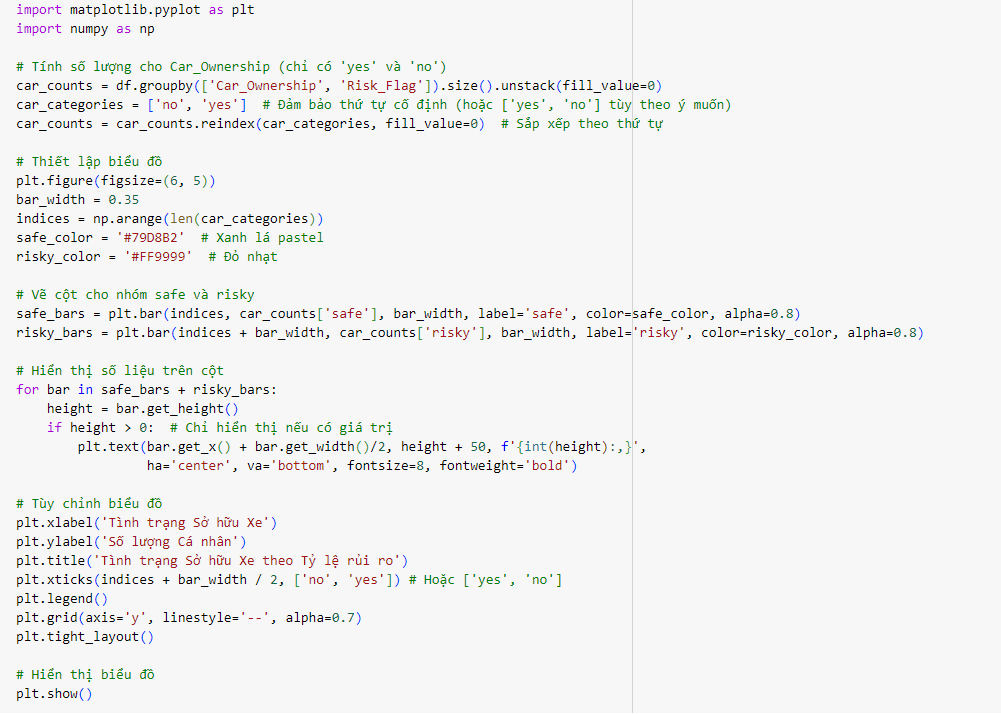
****

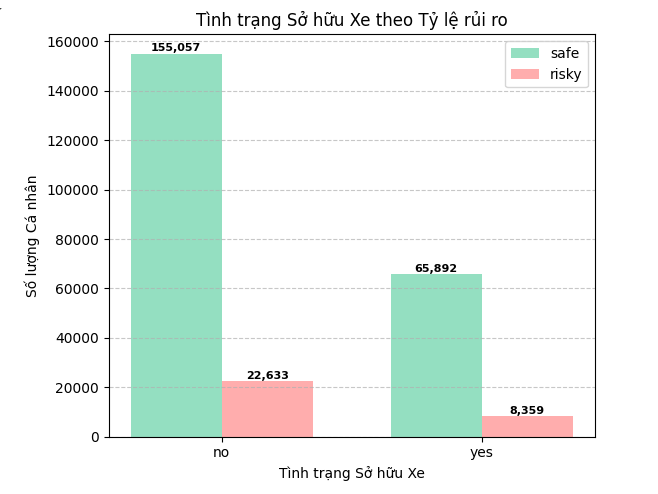
**Kết luận:**

Đối với những người đang thuê nhà thì tỉ lệ rủi ro có tới 29,124 người ( chiếm đến ~94% so với tổng 30,991 người đang trong tình trạng vỡ nợ).

Do đó có thể kết luận những người có tỉ lệ vỡ nợ cao nhất là những người đang thuê nhà.

### **2.4.4. Biểu đồ tình trạng sở hữu xe theo tỷ lệ rủi ro**

****

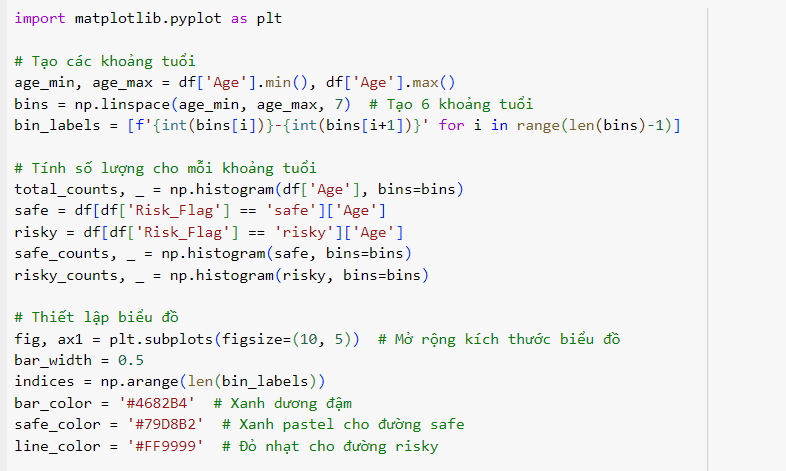
****

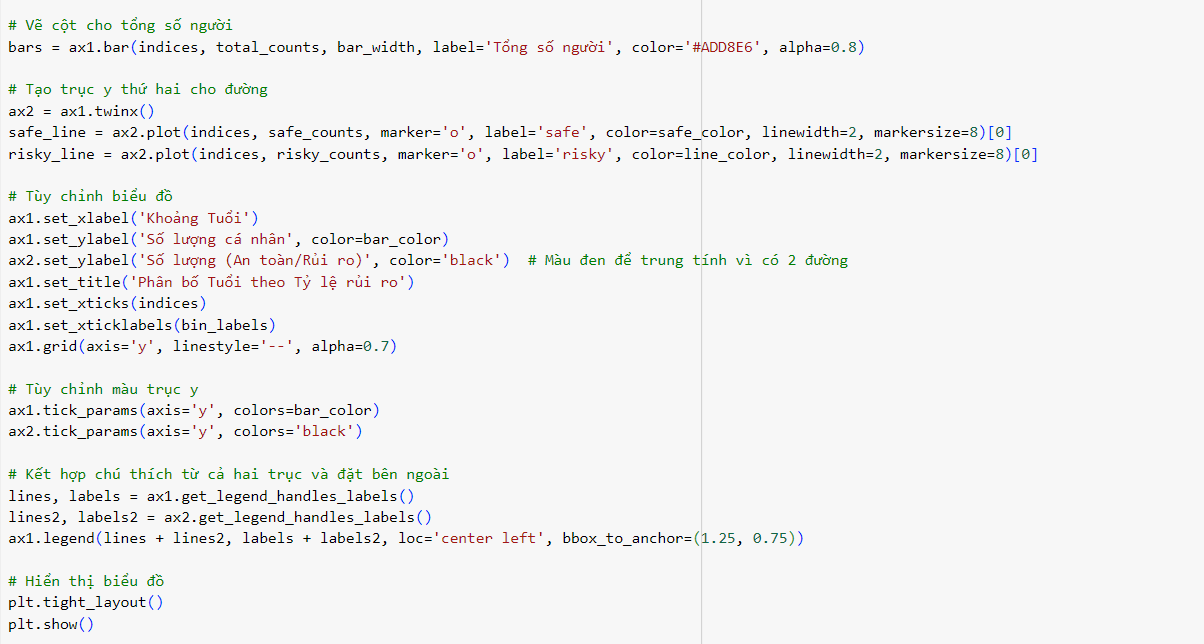
**Kết luận:**

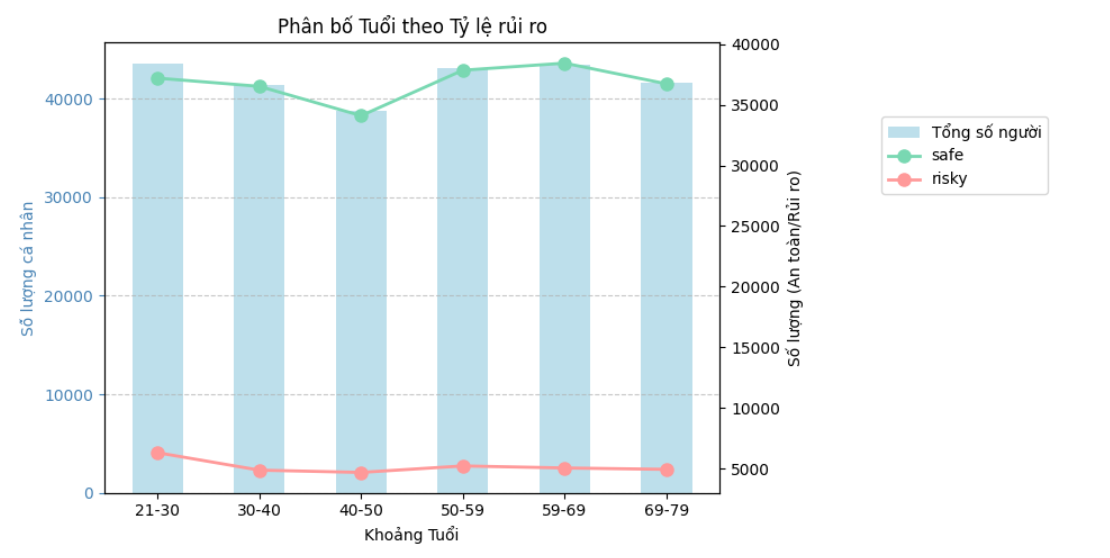
Đối với những người đang không có xe thì tỉ lệ rủi ro có tới 22,633 người ( chiếm đến ~73% so với tổng 30,991 người đang trong tình trạng vỡ nợ).

Do đó có thể kết luận những người có tỉ lệ vỡ nợ cao nhất là những người đang không xe.

### **2.4.5. Biểu đồ phân bố tuổi theo tỷ lệ rủi ro**

****

****

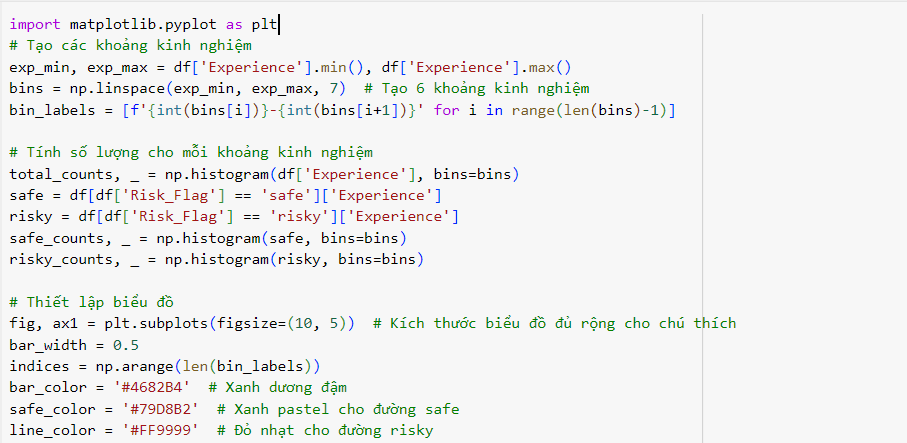
****

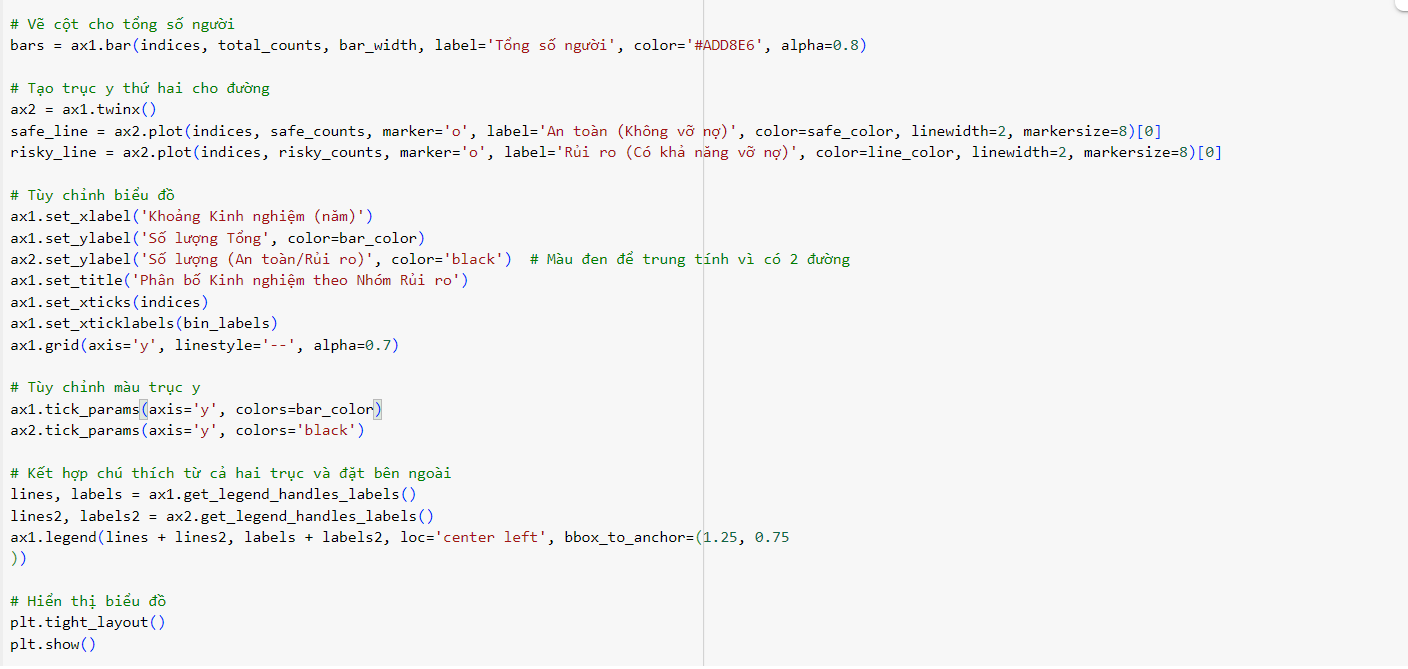
**Kết luận:**

Theo biểu đồ ta có thể thấy độ tuổi có tỉ lệ rủi ro cao nhất là những người nằm trong độ tuổi 21-30 tuổi.

Có thể nhận xét rằng những người trong độ tuổi 21-30 tuổi là những người có khả năng vỡ nợ cao nhất

### **2.4.6. Phân bố kinh nghiệm theo nhóm rủi ro**

****

****

**A graph with a green line and red line

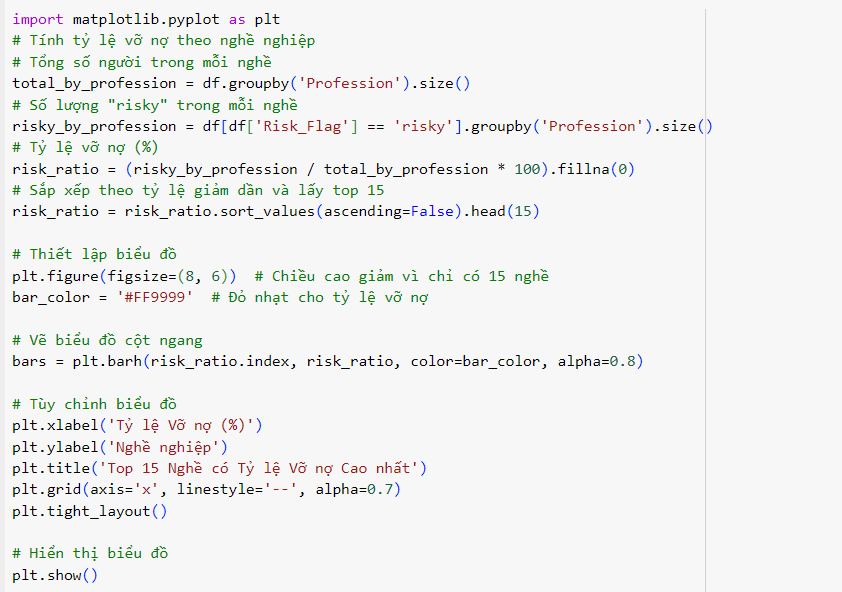
AI-generated content may be incorrect.**

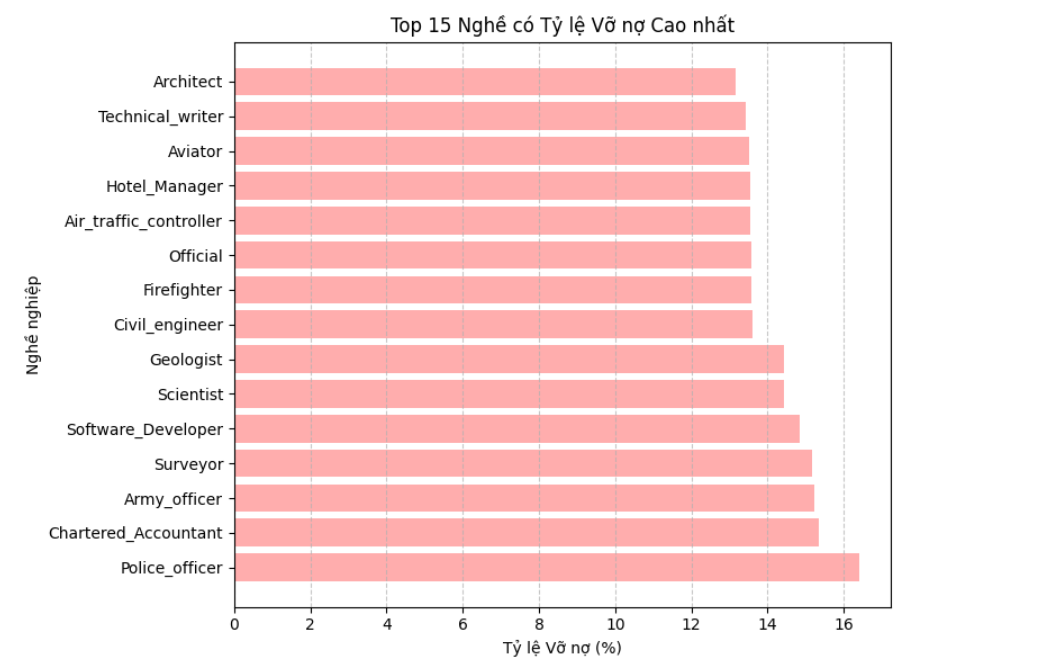
**Kết luận:**

Theo biểu đồ ta có thể thấy độ tuổi kinh nghiệm có tỉ lệ rủi ro cao nhất là những người có kinh nghiệm từ 0-4 năm.

Có thể nhận xét rằng những người có kinh nghiệm từ 0-4 năm là những người có khả năng vỡ nợ cao nhất.

### **2.4.7. Biểu đồ thể hiện nghề nghiệp có tỷ lệ vỡ nợ cao nhất**

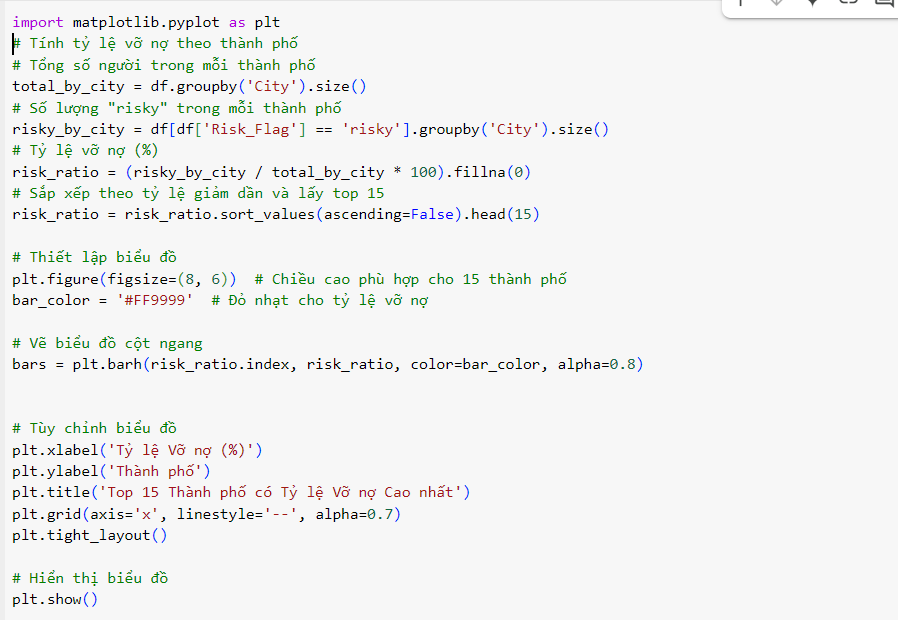
****

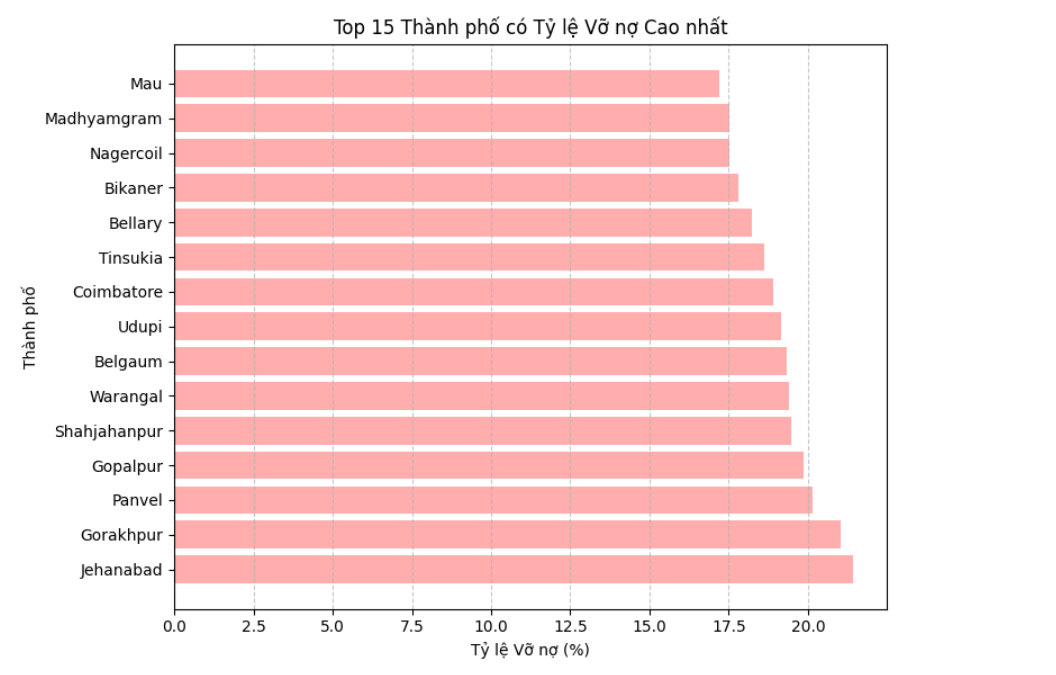
****

**Kết luận:**

Theo biểu đồ thì ta thấy ngành nghề có tỉ lệ vỡ nợ cao nhất là Police-officer với tỉ lệ cao ngất ngưỡng lên tới trên 16%. Các ngành đứng thứ 2,3 cũng chỉ có tỉ lệ vỡ nợ từ 15% đổ xuống.

### **2.4.8. Biểu đồ thể hiện thành phố có tỷ lệ vỡ nợ cao nhất**

****

****

**Kết luận:**

Theo biểu đồ thì ta thấy ngành nghề có tỉ lệ vỡ nợ cao nhất là Jehanabad vớI tỉ lệ khá là cao lên tới trên 22%

### **2.4.9. Đề xuất giải pháp**

Tăng cường kiểm soát rủi ro với nhóm thuê nhà và thuê xe:

* Áp dụng tiêu chí tín dụng nghiêm ngặt hơn, yêu cầu tài sản đảm bảo hoặc lịch sử tín dụng tốt.
* Cung cấp chương trình giáo dục tài chính để quản lý nợ hiệu quả.

Tập trung vào nhóm tuổi 21-30 và nghề nghiệp rủi ro cao:

* Cung cấp chính sách tín dụng linh hoạt (giảm lãi suất, tái cơ cấu nợ) cho nhóm tuổi 40-50 và nghề như kiến trúc sư, phi công.
* Tư vấn tài chính cá nhân cho các nhóm có thu nhập không ổn định.

Giảm thiểu rủi ro ở thành phố có tỷ lệ vốn nợ rủi ro cao:

* Tại Mau, Madhyamgram, Nagercoil, tăng giám sát tín dụng và nâng cao nhận thức tài chính.
* Hợp tác với tổ chức địa phương cung cấp khoản vay lãi suất ưu đãi.

Khuyến khích quản lý vốn nợ an toàn:

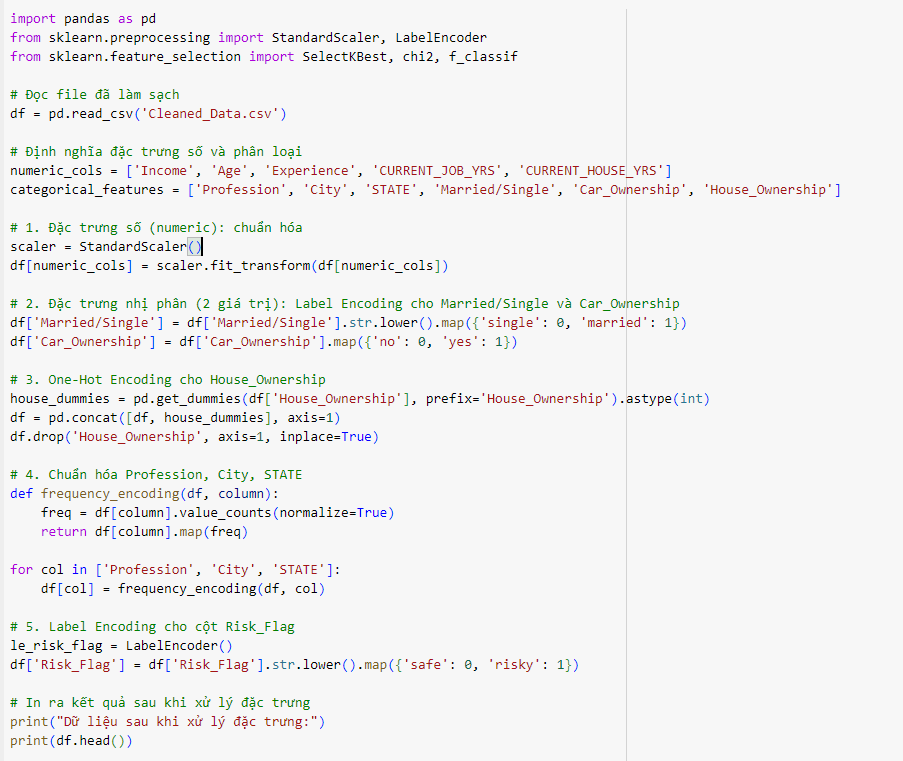
* Thưởng cho khách hàng duy trì vốn nợ an toàn (giảm lãi suất, tăng hạn mức).
* Xây dựng công cụ trực tuyến để theo dõi và quản lý vốn nợ.

Nghiên cứu nhóm có tỷ lệ rủi ro thấp:

* Tìm hiểu lý do nhóm sở hữu nhà/xe và nghề cảnh sát có rủi ro thấp, áp dụng kinh nghiệm cho nhóm khác.
* Khuyến khích nhóm rủi ro cao học thói quen tài chính từ nhóm rủi ro thấp qua đào tạo, hội thảo.

# **CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU ĐƯA VÀO MÔ HÌNH MÁY HỌC**

## **3.1. Chuẩn hóa dữ liệu và mã hóa đặc trưng**



- numeric\_cols=[ Income, Age, Experience, CURRENT\_JOB\_YRS, CURRENT\_HOUSE\_YRS]

* Đây là các cột chứa giá trị số, và các mô hình học máy cần chuẩn hóa chúng để tất cả có phạm vi tương đương. Nếu không, các đặc trưng có phạm vi rộng hơn (ví dụ: Income) sẽ chi phối quá trình học.
* Xử lý: Dùng StandardScaler để chuẩn hóa các giá trị này, chuyển chúng về phân phối chuẩn với trung bình = 0 và độ lệch chuẩn = 1.

- categorical\_features = ['Profession', 'City', 'STATE', 'Married/Single', 'Car\_Ownership', 'House\_Ownership']

* Đây là các cột chứa giá trị phân loại, các mô hình học máy yêu cầu dữ liệu dạng số. Nếu không xử lý, dữ liệu sẽ không thể đưa vào mô hình.

+ Label Encoding: Cột: Married/Single, Car\_Ownership

* Lý do xử lý: Đây là các cột phân loại với chỉ 2 giá trị, ví dụ: "single" hoặc "married" trong Married/Single, và "yes" hoặc "no" trong Car\_Ownership.
* Xử lý: Dùng Label Encoding để chuyển đổi các giá trị chuỗi thành số. Ví dụ: "single" -> 0, "married" -> 1; "no" -> 0, "yes" -> 1.

Cột: Risk\_Flag

* Lý do xử lý: Cột này có 2 giá trị phân loại, ví dụ: "safe" và "risky".
* Xử lý: Dùng Label Encoding để mã hóa "safe" thành 0 và "risky" thành 1.

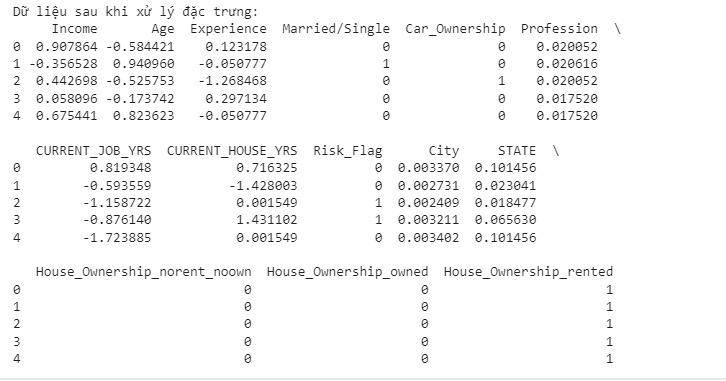
+ One-Hot Encoding: Cột: House\_Ownership

* Lý do xử lý: Cột này chứa nhiều giá trị khác nhau, ví dụ: "Owned", "Rented", "Norent\_noown". Mô hình học máy không thể xử lý trực tiếp các giá trị chuỗi.
* Xử lý: Dùng One-Hot Encoding để chuyển các giá trị này thành các cột nhị phân. Nếu có 3 giá trị như "Owned", "Rented", và "Norent\_noown", ta sẽ tạo ra 3 cột: House\_Ownership\_norent\_noown House\_Ownership\_owned House\_Ownership\_rented, mỗi cột sẽ có giá trị 1 hoặc 0.

+ Frequency Encoding**:** Cột: Profession, City, STATE

* Lý do xử lý: Đây là các cột chứa giá trị chuỗi, như "Engineer", "Teacher", "New York", "California", v.v. Mô hình học máy không thể làm việc trực tiếp với các giá trị chuỗi.
* Xử lý: Sử dụng Frequency Encoding để thay thế mỗi giá trị bằng tỷ lệ xuất hiện của nó trong bộ dữ liệu. Ví dụ, nếu "Engineer" xuất hiện 30%, ta sẽ thay thế "Engineer" bằng 0.3.

- Kết quả:



## **3.2. Sử dụng các phương pháp kiểm định để chọn thuộc tính cần thiết cho mô hình**

### **3.2.1. Kiểm định Chi Square**

**- Mục đích phương pháp**

Chi-Square Test (Kiểm định Chi Bình Phương) là một phương pháp thống kê được sử dụng để kiểm tra sự độc lập giữa các biến phân loại. Mục đích của phương pháp này trong bài toán của bạn là xác định xem các đặc trưng phân loại có liên quan đến biến mục tiêu Risk\_Flag (biến phân loại với các giá trị như "safe" và "risky") hay không.

**- Cách thức hoạt động:**

+ Đặc trưng phân loại và biến mục tiêu

+ Tính toán Chi-Square

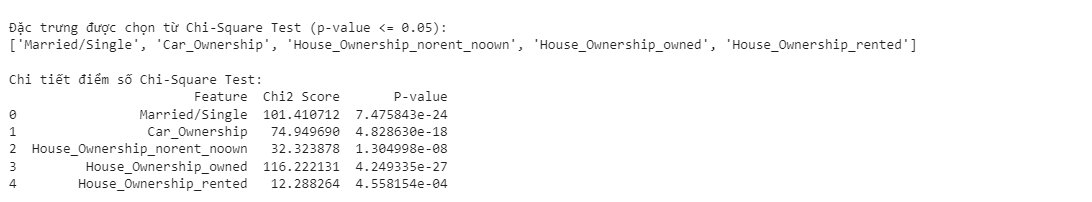
+ Xử lý dữ liệu trước khi kiểm định

+ Áp dụng Chi-Square Test trong thực tế

**- Kết quả đầu ra:**

A screenshot of a computer code

AI-generated content may be incorrect.

****

### **3.2.2. Kiểm định ANOVA F-Test**

**- Mục đích phương pháp**

Phương pháp ANOVA (Analysis of Variance) hoặc F-test là một bài kiểm tra thống kê dùng để xác định sự khác biệt có ý nghĩa giữa các nhóm dữ liệu. Trong bài toán của bạn, ANOVA sẽ được sử dụng để kiểm tra mối quan hệ giữa các đặc trưng số và biến mục tiêu Risk\_Flag, nhằm xác định xem các đặc trưng số có ảnh hưởng đáng kể đến khả năng phân loại "safe" hay "risky".

**- Cách thức hoạt động:**

+ Đặc trưng số và biến mục tiêu

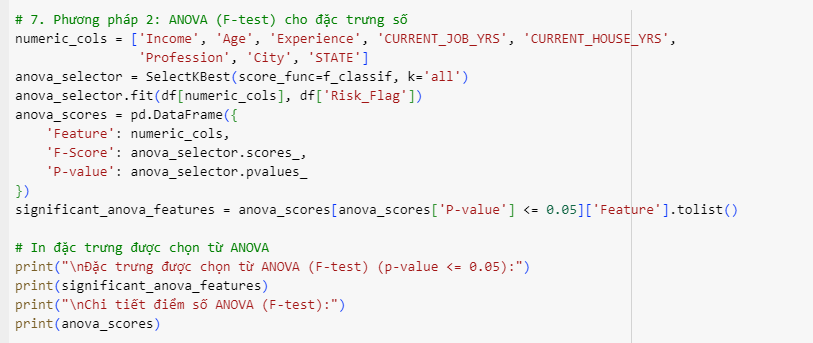
+ Tính toán F-Score

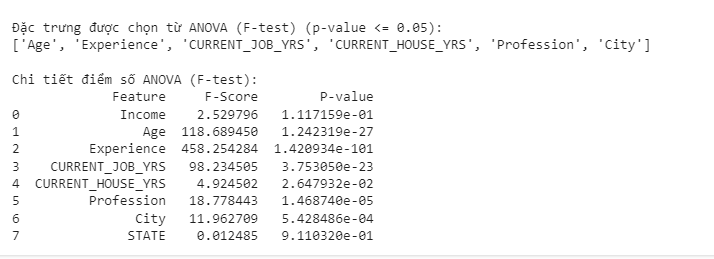
+ P-value

+ Xử lý dữ liệu

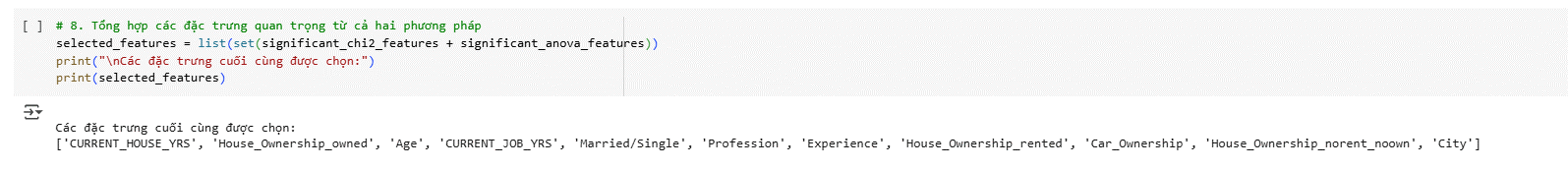
+ Áp dụng ANOVA trong thực tế

**- Kết quả đầu ra**

****

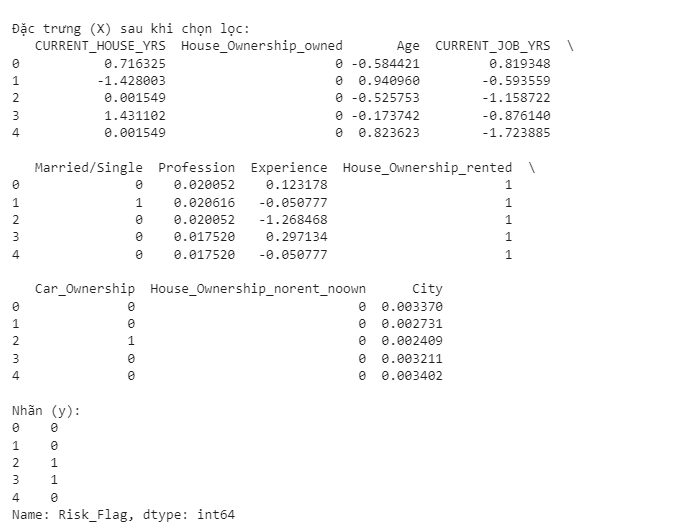
****

## **3.3. Các đặc trưng quan trọng được chọn từ 2 phương pháp trên**

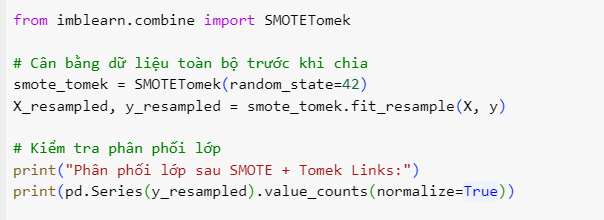
****

## **3.4. Đánh labeling**





## **3.5 Cân bằng dữ liệu**



SMOTETomek là phương pháp kết hợp giữa:

* SMOTE (Synthetic Minority Oversampling Technique): Tạo mẫu giả lập cho lớp rủi ro (1) bằng cách nội suy giữa các điểm dữ liệu, tăng số mẫu để cân bằng với lớp an toàn (0).
* Tomek Links: Loại bỏ các cặp mẫu nhiễu ở biên giới giữa hai lớp, giúp phân tách lớp rõ ràng hơn.
* random\_state=42: Đảm bảo kết quả tái lập, tránh dao động chỉ số (F1-score, Recall, v.v.) qua mỗi lần chạy.
* fit\_resample(): Học cấu trúc dữ liệu (X, y), sau đó áp dụng SMOTETomek để cân bằng tập dữ liệu đầu ra.

- Ưu điểm:

+ Kết hợp oversampling (SMOTE) và làm sạch nhiễu (Tomek Links).

+ Giảm nguy cơ overfitting

+ Hiệu quả với dữ liệu lớn, tỷ lệ mất cân bằng cao

- Nhược điểm:

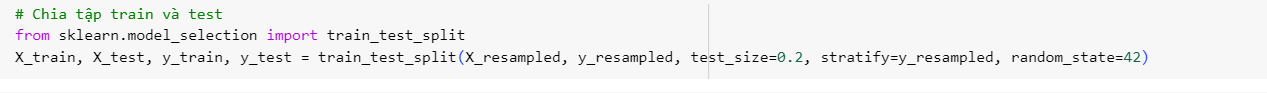
* Tốn thời gian tính toán (tìm hàng xóm, nhiễu).
* Có thể tạo/xóa nhiễu không mong muốn nếu dữ liệu chồng lấn nặng.
* Phụ thuộc vào tham số k, cần thử nghiệm.

A white background with black text

AI-generated content may be incorrect.- Kết quả:

## **3.6. Chia tập train và test**

Sử dụng train\_test\_split để chia dữ liệu thành tập huấn luyện (80%) và kiểm tra (20%), với stratify=y\_resampled để giữ tỷ lệ lớp cân bằng (gần 50:50) và random\_state=42 để đảm bảo tính tái lập. Điều này giúp mô hình học máy được đánh giá chính xác, tổng quát hóa tốt, và phù hợp với mục tiêu phát hiện lớp rủi ro trong bài toán dự đoán vỡ nợ.



## **3.7. Xây dựng mô hình để đưa vào máy học**

### **3.7.1. Lý do chọn mô hình**

Bài toán dự đoán vỡ nợ mang tính phân loại, có dữ liệu phức tạp, đòi hỏi độ chính xác cao nên Decision Tree và Random Forest là hai lựa chọn rất phù hợp.

### **3.7.2. Mô hình Decision Tree**

#### **3.7.2.1. Cơ sở lý thuyết**

Decision tree là một mô hình Supervised learning, có thể được áp dụng vào cả hai bài toán Classification và Regression. Việc xây dựng một Decision tree trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của decision tree là nó có thể làm việc với các đặc trưng (trong các tài liệu về Decision tree, các đặc trưng thường được gọi là thuộc tính – attribute) dạng categorical, thường là rời rạc và không có thứ tự. Ví dụ, mưa, nắng hay xanh, đỏ, v.v.

Decision Tree cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng categorical và liên tục (numeric).

**Ưu điểm**

* Cây quyết định dễ hiểu
* Ít yêu cầu việc chuẩn hoá dữ liệu.
* Cây quyết định có thể xử lý cả dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại và kể cả dữ liệu bị thiếu
* Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn

**Nhược điểm**

* Khi được áp dụng với bộ dữ liệu phức tạp, nhiều biến và thuộc tính khác nhau có thể dẫn đến mô hình bị overfitting, quá khớp với dữ liệu training. Từ đó không đưa ra kết quả phân loại chính xác khi áp dụng cho dữ liệu test và dữ liệu mới.
* Khi có sự thay đổi nhỏ trong bộ dữ liệu có thể gây ảnh hưởng đến cấu trúc của mô hình.
* Thuật toán cây quyết định có khả năng “bias” hay thiên vị nếu bộ dữ liệu không được cân bằng.

#### **3.7.2.2. Huấn luyện mô hình Decision tree**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**- Dự đoán trên tập TEST**

****

### **3.7.3. Mô hình RandomForest**

#### **3.7.3.1. Cơ sở lý thuyết**

Random Forests là thuật toán học có giám sát (supervised learning), có thể được sử dụng cho cả Classfication và Regression. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất.

Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Random forests có nhiều ứng dụng, chẳng hạn như công cụ đề xuất, phân loại hình ảnh và lựa chọn tính năng. Nó có thể được sử dụng để phân loại các ứng viên cho vay trung thành, xác định hoạt động gian lận và dự đoán các bệnh. Nó nằm ở cơ sở của thuật toán Boruta, chọn các tính năng quan trọng trong tập dữ liệu.

**Ưu điểm :**

* Không bị vấn đề Overfitting.
* Thích ứng nhanh với tập dữ liệu, thích hợp cho việc phân tích Dữ liệu lớn
* Có thể xử lý hàng ngàn tính năng đầu vào và các biến tại một thời điểm.
* Có thể được áp dụng cho các vấn đề học không được giám sát.

**Nhược điểm:**

Chậm tạo dự đoán bởi vì nó có nhiều cây quyết định.

Thời gian thực thi lâu vì sử dụng nhiều cây quyết định.

Mô hình khó hiểu hơn so với Cây quyết định.

#### **3.7.3.2. Huấn luyện mô hình Random Forests**

A screen shot of a computer code

AI-generated content may be incorrect.

**- Dự đoán trên tập TEST**

****

# **CHƯƠNG 4: ĐÁNH GIÁ CÁC MÔ HÌNH HỌC MÁY**

## **4.1. Đánh giá mô hình Decision tree**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**- Nhận xét:**

Độ chính xác tổng thể (accuracy):

* Mô hình đạt 92%, nghĩa là trong tổng số 87,102 mẫu, có 92% mẫu được dự đoán đúng. Đây là một độ chính xác cao.

Precision (độ chính xác theo lớp):

* Lớp 0: 94% — trong số các mẫu mà mô hình dự đoán là 0, có 94% là đúng.
* Lớp 1: 90% — trong số các mẫu mà mô hình dự đoán là 1, có 90% là đúng.  
  → Nhìn chung, precision của hai lớp khá đồng đều, chênh lệch không quá lớn.

Recall (độ nhạy theo lớp):

* Lớp 0: 90% — mô hình dự đoán đúng 90% các mẫu thực sự là 0.
* Lớp 1: 94% — mô hình dự đoán đúng 94% các mẫu thực sự là 1.  
  → Recall cho lớp 1 cao hơn một chút so với lớp 0.

F1-score (trung bình hài hòa giữa precision và recall):

* Cả lớp 0 và lớp 1 đều đạt 0.92 F1-score.  
  → Điều này cho thấy mô hình cân bằng tốt giữa precision và recall.

Macro avg và weighted avg đều bằng 0.92 → chứng tỏ mô hình khá cân bằng giữa hai lớp (không bị thiên lệch về một lớp nào).

Thời gian thực thi của mô hình: 6.59 giây

## **4.2. Đánh giá mô hình Random Forests**

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

**- Nhận xét:**

Độ chính xác tổng thể (accuracy):

* Mô hình đạt 94%, tức là trong tổng số 87,102 mẫu, có 94% mẫu được dự đoán đúng.  
  → So với bảng trước (92%), mô hình này tốt hơn.

Precision (độ chính xác theo lớp):

* Lớp 0: 96% — trong số các mẫu dự đoán là 0, có 96% là đúng.
* Lớp 1: 92% — trong số các mẫu dự đoán là 1, có 92% là đúng.  
  → Precision lớp 0 nhỉnh hơn lớp 1.

Recall (độ nhạy theo lớp):

* Lớp 0: 92% — mô hình dự đoán đúng 92% các mẫu thực sự là 0.
* Lớp 1: 96% — mô hình dự đoán đúng 96% các mẫu thực sự là 1.  
  → Recall lớp 1 nhỉnh hơn lớp 0.

F1-score (trung bình giữa precision và recall):

* Cả hai lớp đều đạt 0.94 F1-score.  
  → Rất cân bằng và cao.

Macro avg và weighted avg đều là 0.94 → mô hình vẫn duy trì sự cân bằng tốt giữa hai lớp.

Thời gian thực thi: 82.10 giây

## **4.3. So sánh 2 mô hình**

**A table with numbers and text

AI-generated content may be incorrect.**

## **4.4. Tổng kết**

Mô hình Random Forests cho độ chính xác (accuracy 94%) và F1-score cao hơn, đồng thời cân bằng tốt giữa precision và recall.

Mô hình Decision tree tuy độ chính xác thấp hơn (92%), nhưng thời gian thực thi nhanh hơn nhiều (chỉ 6.59 giây so với 82.10 giây).

Nhìn chung, cả Decision Tree và Random Forest đều cho kết quả dự đoán rất tốt. Tuy nhiên, do bài toán dự đoán vỡ nợ yêu cầu độ chính xác cao để hạn chế rủi ro tài chính, nên việc lựa chọn mô hình 2 sẽ là phương án hợp lý hơn.

# **CHƯƠNG 5: KẾT LUẬN**

Trong quá trình thực hiện đề tài "Dự đoán khoản vay dựa trên hành vi khách hàng", nhóm đã tiến hành toàn bộ các bước từ khảo sát dữ liệu, tiền xử lý, trực quan hóa, đến xây dựng và đánh giá các mô hình máy học. Đây là một đề tài thực tiễn, có ý nghĩa lớn đối với lĩnh vực tài chính - ngân hàng, giúp các doanh nghiệp chủ động hơn trong việc đánh giá rủi ro khoản vay và đề xuất các chiến lược phù hợp nhằm tối ưu hóa hoạt động kinh doanh.

Trước tiên, chúng em đã tiến hành xử lý dữ liệu thô bằng cách loại bỏ các giá trị thiếu, không hợp lệ, đồng thời chuẩn hóa và mã hóa các thuộc tính để dữ liệu sẵn sàng cho quá trình xây dựng mô hình. Các biểu đồ trực quan đã được sử dụng nhằm làm nổi bật mối quan hệ giữa các yếu tố như thu nhập, độ tuổi, tình trạng sở hữu nhà/xe và tỷ lệ vỡ nợ, từ đó đưa ra những phân tích trực quan và dễ hiểu.

Ở giai đoạn tiếp theo, nhóm áp dụng các phương pháp kiểm định như Chi-square và ANOVA F-test để chọn lọc những đặc trưng quan trọng nhất, đồng thời thực hiện cân bằng dữ liệu nhằm cải thiện độ chính xác của mô hình. Các mô hình Decision Tree và Random Forest đã được triển khai để thực hiện dự đoán, trong đó Random Forest cho thấy hiệu quả cao hơn về cả độ chính xác và khả năng tổng quát hóa dữ liệu.

Kết quả đánh giá dựa trên các tiêu chí về thời gian thực thi, độ chính xác (Accuracy) và sai số trung bình tuyệt đối (MAE) đã cho thấy các mô hình xây dựng đạt hiệu suất khá tốt, hoàn thành mục tiêu đã đề ra.

Thông qua quá trình thực hiện đề tài, nhóm không chỉ củng cố kiến thức về xử lý và phân tích dữ liệu mà còn tiếp cận và áp dụng thành công các kỹ thuật học máy vào một bài toán thực tiễn. Đây là bước nền tảng quan trọng cho việc nghiên cứu sâu hơn và ứng dụng dữ liệu trong lĩnh vực quản lý thông tin cũng như các lĩnh vực liên quan trong tương lai.

# **TÀI LIỆU THAM KHẢO**

[1]. Decision Tree, [Decision Tree algorithm — Machine Learning cho dữ liệu dạng bảng (machinelearningcoban.com)](https://machinelearningcoban.com/tabml_book/ch_model/decision_tree.html)

[2]. [Decision Tree - GeeksforGeeks](https://www.geeksforgeeks.org/decision-tree/)

[3.] Random Forest, <https://www.datacamp.com/community/tutorials/random-forests-classifier-python>