HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA CÔNG NGHỆ THÔNG TIN**

------------------------------

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỌC: NHẬP MÔN KHOA HỌC DỮ LIỆU**

**Tên đề tài: Xây dựng model dự đoán trả nợ vay**

**Giảng viên hướng dẫn: Vũ Hoài Nam**

**Nhóm bài tập lớn: 10**

**Thành viên nhóm:**

**Phạm Trọng Tiến - B20DCCN597**

**Nguyễn Hoàng Hà - B20DCCN213**

**Hoàng Văn An - B20DCCN045**

**Hà Nội, tháng 11 năm 2023**

# **LỜI MỞ ĐẦU**

Ngày nay cùng với sự phát triển của xã hội khoa học hiện đại công nghệ khoa học đã và đang có những bước phát triển mạnh mẽ không ngừng, ngày càng thâm nhập sâu, trở thành một công cụ hỗ trợ đắc lực cho con người trong mọi lĩnh vực. Song song với sự phát triển đó là nhu cầu áp dụng khoa học dữ liệu trong thực tiễn ngày càng tăng, nhất là trong lĩnh vữc tài chính. Trong bối cảnh ngân hàng và tổ chức tài chính ngày càng chú trọng vào việc đánh giá rủi ro và quản lý nợ,, việc áp dụng khoa học dữ liệu trong lĩnh vực tài chính, đặc biệt là trong việc đánh giá rủi ro và dự đoán khả năng trả nợ vay, đã trở thành một phần quan trọng và hấp dẫn.

Trong thời gian học tập tìm hiểu về môn "Nhập môn khoa học dữ liệu", chúng em đã hiểu thế mạnh mà các hệ dự đoán đem lại. Vì vậy, nhóm em chọn đề tài bài tập lớn với nội dung “Xây dựng model dự đoán trả nợ vay”.

Mục tiêu của báo cáo là chia sẻ kinh nghiệm và kiến thức của chúng em trong việc xây dựng hệ thống dự đoán, đồng thời cung cấp một cái nhìn tổng quan về môn học nhập môn khoa học dữ liệu . Chúng em hy vọng rằng báo cáo này sẽ giúp cho các độc giả có cái nhìn rõ ràng hơn về xây dựng hệ thống, cách áp dụng nó vào các dự án thực tế và đóng góp một phần nhỏ vào việc nâng cao hiểu biết về xử lý dữ liệu và xây dựng mô hình trong lĩnh vực tài chính.

Dưới sự giúp đỡ và hướng dẫn tận tình của thầy Vũ Hoài Nam chúng em đã tìm hiểu và hoàn thành chủ đề của mình. Tuy nhiên với lượng kiến thức có hạn, thời gian còn hạn chế và tìm hiểu chưa được sâu nên bài còn nhiều thiếu sót.Chúng em rất mong nhận được sự đóng góp nhiệt tình của các thầy cô và bạn bè để đề tài của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

**Nhóm sinh viên 10**

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 2](#_Toc152179134)

[**I.** **GIỚI THIỆU BÀI TOÁN:** 4](#_Toc152179135)

[1. Vấn đề đặt ra : dự đoán trả nợ vay(Loan repayment prediction) 4](#_Toc152179136)

[2. Mục tiêu: 4](#_Toc152179137)

[3. Ý nghĩa của đề tài: 5](#_Toc152179138)

[**4. Phạm vi nghiên cứu:** 5](#_Toc152179139)

[**II.** **DỮ LIỆU** 5](#_Toc152179140)

[1. Nguồn gốc tập dữ liệu: 5](#_Toc152179141)

[2. Mô tả các tính năng 5](#_Toc152179142)

[3. Số lượng dữ liệu 6](#_Toc152179143)

[4. Tiền xử lý dữ liệu: 6](#_Toc152179144)

[5. Xây Dựng Mô Hình 8](#_Toc152179145)

[6. Công cụ, ngôn ngữ sử dụng. 9](#_Toc152179146)

[**III.** **CHƯƠNG TRÌNH CHẠY:** 9](#_Toc152179147)

[1. Chuẩn bị dữ liệu 9](#_Toc152179148)

[2. Setup 9](#_Toc152179149)

# **GIỚI THIỆU BÀI TOÁN:**

* + 1. **Vấn đề đặt ra : dự đoán trả nợ vay(Loan repayment prediction)**
* Tập trung vào một vấn đề quan trọng trong lĩnh vực tài chính và cho vay, đó là việc xây dựng một hệ thống dự đoán rủi ro tín dụng. Được phát triển dựa trên dữ liệu thu thập từ Lending-Club.com, nền tảng cho vay trực tuyến lớn, đề tài này hứa hẹn mang lại cái nhìn sâu sắc về khả năng thanh toán của khách hàng và giúp đánh giá rủi ro tín dụng hiệu quả.



* + 1. **Mục tiêu:**
* Mục tiêu chính của dự án là phân loại khách hàng theo khả năng thanh toán và đánh giá rủi ro tín dụng. Qua việc kết hợp nhiều yếu tố quan trọng như thu nhập, mục đích vay, lãi suất, và các chỉ số tín dụng, chúng ta sẽ xây dựng một mô hình dự đoán chính xác và có ứng dụng thực tế. Bằng cách này, hệ thống sẽ hỗ trợ quyết định về cấp tín dụng, giảm thiểu rủi ro mà tổ chức cho vay có thể phải đối mặt.



* + 1. **Ý nghĩa của đề tài:**
* Tăng hiệu quả cho vay: Hệ thống dự đoán rủi ro tín dụng giúp tổ chức cho vay xác định khách hàng có khả năng thanh toán cao, từ đó tối ưu hóa quy trình cho vay và giảm thiểu rủi ro tín dụng.
* Bảo vệ lợi ích người vay: Bằng cách phân loại khách hàng một cách chính xác, đề tài đảm bảo rằng người vay có khả năng thanh toán sẽ được cung cấp các sản phẩm tài chính với điều kiện thuận lợi, bảo vệ họ khỏi áp lực tài chính không cần thiết.
* Áp dụng trí tuệ nhân tạo cho tài chính: Sử dụng mô hình học máy và trí tuệ nhân tạo đưa vào ngành tài chính, đề tài này còn là một ví dụ minh họa cho việc ứng dụng công nghệ để cải thiện quy trình quyết định tín dụng truyền thống.

## **4. Phạm vi nghiên cứu:**

* Dữ liệu: Dữ liệu sẽ được thu thập từ Lending-Club.com, bao gồm các biến quan trọng như mục đích vay, lãi suất, thu nhập, điểm FICO, và các thông tin liên quan.
* Phân loại khách hàng: Dự án sẽ tập trung vào việc phân loại khách hàng thành các nhóm có khả năng thanh toán khác nhau, dựa trên các yếu tố ảnh hưởng đến khả năng này.
* Xây dựng mô hình dự đoán: Sử dụng các mô hình học máy để xây dựng một hệ thống dự đoán rủi ro tín dụng hiệu quả.

1. **DỮ LIỆU**
2. **Nguồn gốc tập dữ liệu:**

* Tập dữ liệu được thu thập từ Lending Club, một nền tảng tài chính trực tuyến kết nối người vay và nhà đầu tư. Dữ liệu này cung cấp thông tin về các khoản vay và các yếu tố liên quan đến khả năng trả nợ của người vay.

1. **Mô tả các tính năng**

* **credit\_policy:** Trường này nhận giá trị 1 nếu khách hàng đáp ứng đủ các tiêu chí đánh giá rủi ro tín dụng của Lending-Club.com, và 0 là ngược lại.
* **purpose:** Đây là mục đích của khoản vay, bao gồm các danh mục như mở thẻ tín dụng, mua xe hơi, trang thiết bị gia đình, và nhiều mục khác.
* **int\_rate:** Là lãi suất của khoản vay, là một trong những yếu tố quan trọng đánh giá chi phí vay.
* **installment**: Số tiền trả góp hàng tháng mà người vay phải thanh toán.
* **log\_annual\_inc:** Là logarit của thu nhập hàng năm của người vay, giúp chuẩn hóa dữ liệu thu nhập.
* **dti:** Hệ số nợ trên thu nhập (Debt to Income Ratio), là tỉ lệ phần trăm của tổng thu nhập hàng tháng để trả các khoản thanh toán nợ hàng tháng.
* **fico:** Điểm FICO là một yếu tố quan trọng để đánh giá rủi ro tín dụng. Các khoảng điểm được phân loại từ tệ đến xuất sắc.
* **days\_with\_cr\_line:** Số ngày mà người vay đã được cấp hạn mức tín dụng.
* **revol\_bal:** Số dư tín dụng quay vòng, đại diện cho hạn mức tín dụng cá nhân.
* **revol\_util:** Tỉ lệ sử dụng nợ tín dụng, là tỉ lệ phần trăm của tổng nợ tín dụng hiện có của người vay đang được sử dụng.
* **inq\_last\_6mths:** Số lượng các câu hỏi của người vay đối với tổ chức tài chính trong 6 tháng qua.
* **delinq\_2yrs:** Số lần người vay đã quá 30 ngày đến hạn thanh toán trong 2 năm qua.
* **pub\_rec:** Thông tin về lịch sử tín dụng của bên vay mà bên cho vay có thể sử dụng để đánh giá rủi ro tín dụng.
* **not\_fully\_paid:** Trường này cho biết liệu khoản vay đã được hoàn trả đầy đủ hay không (1: đã hoàn trả đầy đủ, 0: chưa hoàn trả đầy đủ).

1. **Số lượng dữ liệu**

* Tập dữ liệu bao gồm một số lượng lớn các mẫu, mỗi mẫu có thông tin chi tiết về các tính năng đã nêu trên. Việc này cung cấp đủ dữ liệu để xây dựng và đào tạo mô hình dự đoán trả nợ khoản vay.

1. **Tiền xử lý dữ liệu:**

* Dữ liệu được tải từ Kaggle, và nó được lưu trữ trong dataframe của thư viện Pandas.
* Các tính năng và nhãn được xác định, trong đó "not\_fully\_paid" là nhãn cần được dự đoán.
* Dữ liệu được hiển thị thông qua hàm head() để kiểm tra cấu trúc và giá trị ban đầu.
* Khám phá dữ liệu (EDA):
* Sử dụng hàm info() để kiểm tra thông tin về dữ liệu, bao gồm kiểu dữ liệu và có bao nhiêu giá trị không rỗng.
* Sử dụng hàm describe() để kiểm tra các thống kê chính của dữ liệu như giá trị trung bình, độ lệch chuẩn, và phân vị.
* Dữ liệu được kiểm tra để xác định xem có giá trị NaN (rỗng) nào không và được loại bỏ nếu cần thiết.
* Tiền xử lý dữ liệu (tiếp theo):
* Cột "purpose" là một biến phân loại, và nó được chuyển đổi thành biến số bằng phương pháp mã hóa one-hot (hoặc có thể dùng phương pháp mã hóa số hóa).
* Cột "credit\_policy" không cung cấp thông tin hữu ích, vì nó chỉ chứa giá trị "1". Do đó, cột này đã được loại bỏ.
* Kiểm tra độ tương quan:
* Sử dụng hàm corr() để tạo ma trận tương quan giữa các tính năng.
* Ma trận tương quan được hiển thị bằng heatmap để có cái nhìn trực quan về mức độ tương quan giữa các cặp tính năng.
* Chọn tính năng:
* Sử dụng mô hình RandomForest để đánh giá độ quan trọng của từng tính năng.
* Tính năng được chọn bao gồm "credit.policy", "int.rate", "installment", "log.annual.inc", và "fico".
* Chia dữ liệu và chuẩn hóa:
* Sử dụng hàm train\_test\_split() để chia dữ liệu thành tập huấn luyện và tập kiểm tra.
* Sử dụng StandardScaler để chuẩn hóa giá trị của các tính năng, giúp mô hình học tốt hơn.
* Xử lý mất cân bằng dữ liệu:
* Sử dụng kỹ thuật SMOTE để tăng cường dữ liệu cho lớp thiếu sót, đảm bảo rằng mô hình được huấn luyện trên một tập dữ liệu cân bằng hơn.

1. **Xây Dựng Mô Hình**
2. **Xây dựng và đào tạo mô hình Logistic Regression:**

* Sử dụng mô hình Logistic Regression từ thư viện scikit-learn.
* Mô hình được đào tạo trên tập huấn luyện đã được chuẩn hóa và tăng cường.

1. **Train 1:Mô Hình Logistic Regression**

Trong quá trình phát triển và tối ưu hóa mô hình cho dự án của chúng tôi, chúng tôi đã chọn áp dụng mô hình Logistic Regression, một phương pháp phân loại tuyến tính phổ biến và mạnh mẽ.

* Logistic Regression: Một Phương Pháp Phân Loại Độc Đáo.Logistic Regression là một mô hình thống kê được sử dụng rộng rãi trong các bài toán phân loại nhị phân, nơi mục tiêu là dự đoán xác suất thuộc về một lớp cụ thể. Mặc dù có từ "regression" trong tên, Logistic Regression thực sự là một mô hình phân loại, không phải là mô hình hồi quy. Phương pháp này được ưa chuộng vì tính đơn giản và khả năng giải thích cao.
* Tính linh hoạt và hiệu quả:Mô hình Logistic Regression hoạt động bằng cách ánh xạ tổng trọng số của các đặc trưng đầu vào thông qua hàm logistic, chuyển đổi đầu ra thành một giá trị xác suất nằm trong khoảng từ 0 đến 1. Điều này làm cho nó rất linh hoạt trong việc xử lý các tình huống mà mối quan hệ giữa đặc trưng và kết quả là tuyến tính hoặc gần tuyến tính.
* Dễ hiểu và dễ giải thích:Một ưu điểm lớn của Logistic Regression là tính dễ hiểu của nó. Các trọng số được gán cho mỗi đặc trưng cung cấp một cách rõ ràng để hiểu tác động của từng yếu tố lên dự đoán. Điều này làm cho Logistic Regression trở thành một công cụ mạnh mẽ cho việc hiểu biến chấm dứt (categorical variables) và tạo ra quyết định dựa trên dữ liệu.
* Ứng dụng rộng rãi trong ngôn ngữ máy tính và dự Báo:Với ưu điểm về tính linh hoạt, đơn giản, và khả năng giải thích cao, Logistic Regression thường được sử dụng trong nhiều lĩnh vực, bao gồm y học, tài chính, xã hội học, và ngôn ngữ máy tính. Sự kết hợp giữa hiệu suất và tính dễ giải thích của mô hình này làm cho nó trở thành lựa chọn ưa thích khi xây dựng các hệ thống dự đoán và phân loại.

1. **Train 2:**

* **Không Upsampling**:
* Upsampling là quá trình tăng cường số lượng mẫu của lớp thiểu số trong tập dữ liệu để cân bằng với lớp đa số.
* Không thực hiện upsampling có thể ám chỉ rằng bạn chấp nhận sự mất cân bằng và muốn mô hình học từ dữ liệu không cân bằng.
* **XGBoost:**
* XGBoost là một thuật toán học máy dựa trên cây quyết định, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy.
* XGBoost tự nó có khả năng xử lý dữ liệu không cân bằng, do có các tham số có thể được điều chỉnh để đối phó với vấn đề này.

1. **Đánh giá mô hình:**

* Sử dụng các độ đo đánh giá như precision, recall, và f1-score để đánh giá hiệu suất của mô hình trên tập kiểm tra.
* Ma trận nhầm lẫn (confusion matrix) được tạo để thấy rõ hiệu suất của mô hình trong việc dự đoán nhãn của các mẫu.

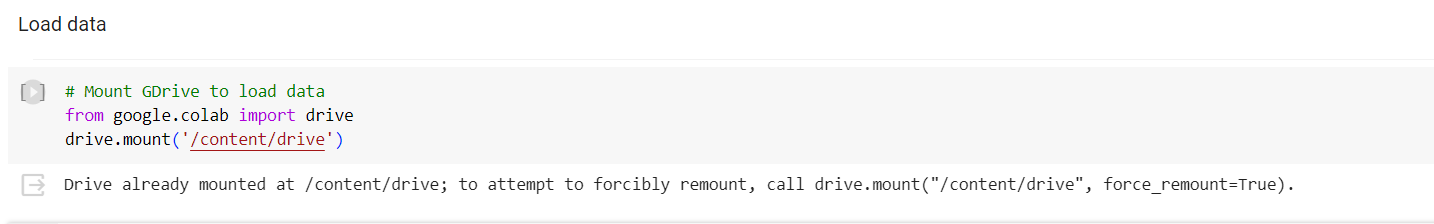
1. **Công cụ, ngôn ngữ sử dụng.**

* Ngôn ngữ: sử dụng các thư viện của python để huấn luyện mô hình(xgboot, category\_encoders, imbalanced-learn,..)
* Công cụ: Sử dụng gg.Colab để build và ggDrive để lưu trữ data

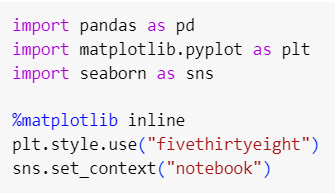
1. **CHƯƠNG TRÌNH CHẠY:**
   * 1. **Chuẩn bị dữ liệu**

Dữ liệu được sử dụng trong dự án được lấy từ tập tin "loan\_data.csv". Đây là nguồn thông tin chính giúp chúng tôi hiểu và dự đoán trạng thái trả nợ của các khoản vay.

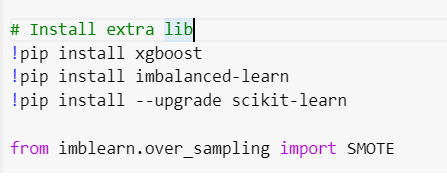
* + 1. **Setup**
* Thiết lập kết nối với Google Drive trong môi trường Google Colab.
* Import thư viện: Mã nhập mô-đun drive từ thư viện google.colab, cung cấp các chức năng liên quan đến Google Colab.
* Mount Google Drive: Lệnh drive.mount('/content/drive') được sử dụng để gắn kết Google Drive. Điều này có nghĩa là nội dung của Google Drive sẽ có sẵn trong môi trường Colab và sẽ được gắn kết tại thư mục /content/drive.
* Quyền ứng dụng: Sau khi chạy ô mã này, một liên kết sẽ được cung cấp và bạn sẽ được yêu cầu nhập mã ủy quyền. Đây là quy trình một lần để cho phép Colab truy cập vào Google Drive của bạn.



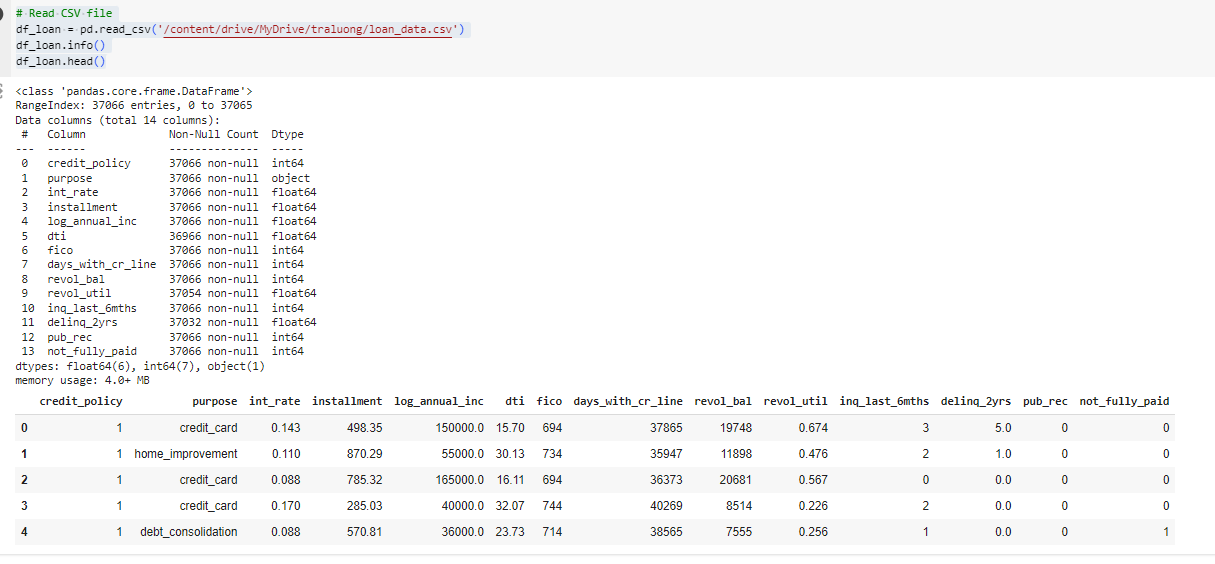
* Sử dụng thư viện Python để thực hiện các nhiệm vụ liên quan đến xử lý và trực quan hóa dữ liệu
* Import Thư viện Pandas, Matplotlib và Seaborn:
* import pandas as pd: Nhập thư viện Pandas và đặt tên viết tắt là pd. Pandas được sử dụng để làm việc với dữ liệu dưới dạng DataFrame.
* import matplotlib.pyplot as plt: Nhập thư viện Matplotlib để tạo các biểu đồ và đặt tên viết tắt là plt.
* import seaborn as sns: Nhập thư viện Seaborn để trực quan hóa dữ liệu một cách đẹp mắt và hiệu quả.
* Thiết lập Hiển thị đồ thị trong Jupyter Notebook:
* %matplotlib inline: Dòng lệnh này là "magic command" trong Jupyter Notebook để hiển thị đồ thị ngay tại nơi mã được chạy.
* Thiết lập Phong cách và Ngữ cảnh của Seaborn:
* plt.style.use("fivethirtyeight"): Chọn phong cách đồ thị "fivethirtyeight" từ Matplotlib để áp dụng cho các đồ thị được tạo ra sau đó.
* sns.set\_context("notebook"): Thiết lập ngữ cảnh của Seaborn cho "notebook", đảm bảo rằng các đồ thị sẽ được hiển thị đẹp mắt trong môi trường notebook.



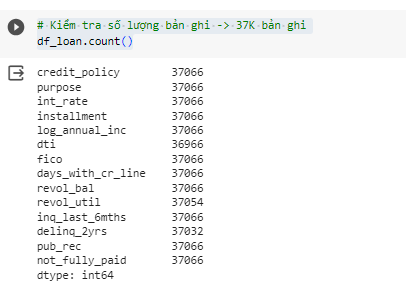
* Cài đặt thư viện bổ sung:
* !pip install xgboost: Cài đặt thư viện XGBoost, một thư viện mạnh mẽ được sử dụng cho các mô hình học máy dựa trên cây quyết định.
* !pip install imbalanced-learn: Cài đặt thư viện Imbalanced-learn, được sử dụng để xử lý vấn đề mất cân bằng dữ liệu, đặc biệt là khi có sự chênh lệch giữa các lớp trong tập dữ liệu.
* !pip install --upgrade scikit-learn: Cập nhật thư viện scikit-learn lên phiên bản mới nhất.



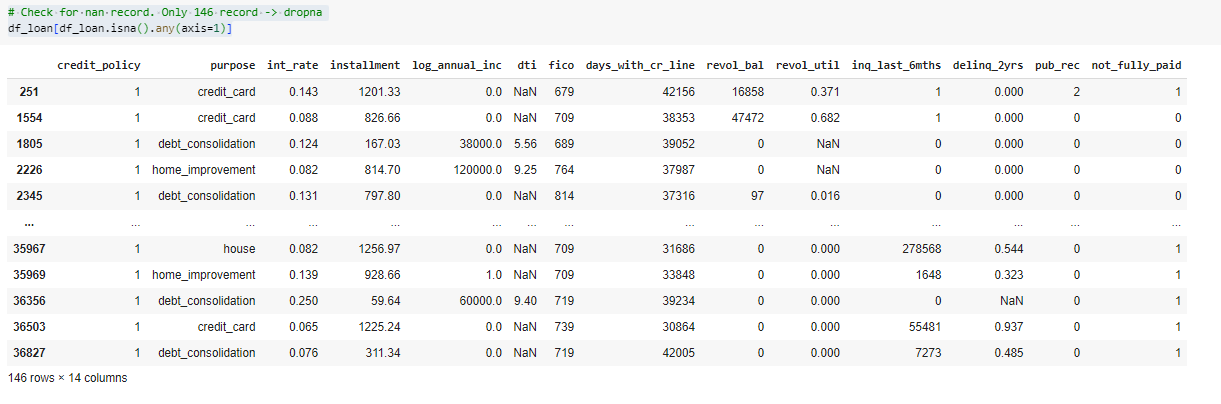
* Đọc dữ liệu từ CSV File:
* df\_loan = pd.read\_csv('/content/drive/MyDrive/traluong/loan\_data.csv'): Đọc dữ liệu từ file CSV có đường dẫn là '/content/drive/MyDrive/traluong/loan\_data.csv' và lưu kết quả vào DataFrame df\_loan. Đây là bước quan trọng để tải dữ liệu vào môi trường làm việc của bạn.
* Hiển Thị Thông Tin Dữ Liệu:
* df\_loan.info(): Hiển thị thông tin về DataFrame df\_loan, bao gồm tên cột, số lượng giá trị không rỗng, kiểu dữ liệu của mỗi cột, và dung lượng bộ nhớ sử dụng. Điều này giúp kiểm tra sơ bộ về dữ liệu và xác định xem có cần thực hiện các bước tiền xử lý không.
* Hiển Thị Năm Dòng Đầu Tiên của Dữ Liệu:
* df\_loan.head(): Hiển thị năm dòng đầu tiên của DataFrame df\_loan, giúp xem xét cấu trúc và dạng của dữ liệu để có cái nhìn tổng quan.



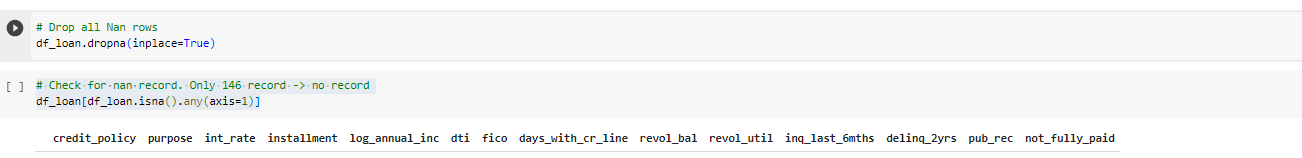
* df\_loan.count() được sử dụng để kiểm tra số lượng bản ghi trong DataFrame df\_loan. Nếu kết quả là 37,000, đó có thể là số lượng bản ghi trong tập dữ liệu.



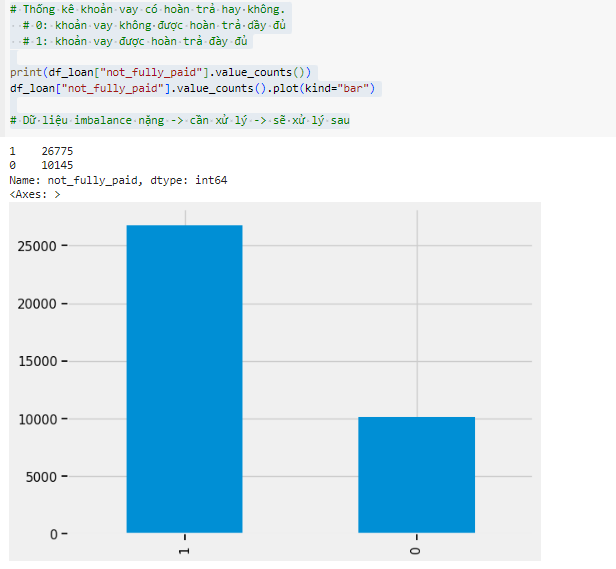
* df\_loan[df\_loan.isna().any(axis=1)] được sử dụng để kiểm tra và hiển thị các bản ghi trong DataFrame df\_loan có chứa giá trị thiếu (NaN) ở ít nhất một cột nào đó. Nếu kết quả chỉ có 146 bản ghi, đó là số lượng bản ghi trong DataFrame mà có ít nhất một giá trị thiếu.



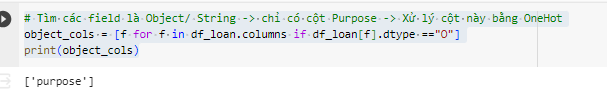
* df\_loan.dropna(inplace=True) được sử dụng để loại bỏ (drop) tất cả các dòng trong DataFrame df\_loan mà chứa ít nhất một giá trị thiếu. Tham số inplace=True có nghĩa là thực hiện thay đổi trực tiếp trên DataFrame hiện tại mà không cần gán cho một biến mới.
* df\_loan[df\_loan.isna().any(axis=1)] cho thấy không còn bản ghi nào chứa giá trị thiếu (NaN). Điều này đảm bảo rằng dữ liệu hiện tại không có giá trị thiếu và có thể được sử dụng cho các bước tiếp theo trong quá trình xử lý và xây dựng mô hình.



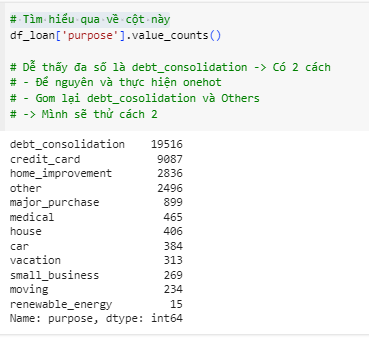
* Thống kê khoản vay có hoàn trả hay không:



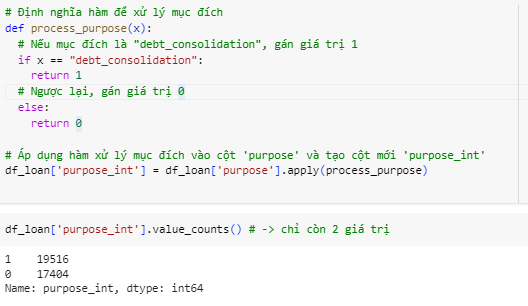
* Tìm các field là Object/ String -> chỉ có cột Purpose -> Xử lý cột này bằng OneHot



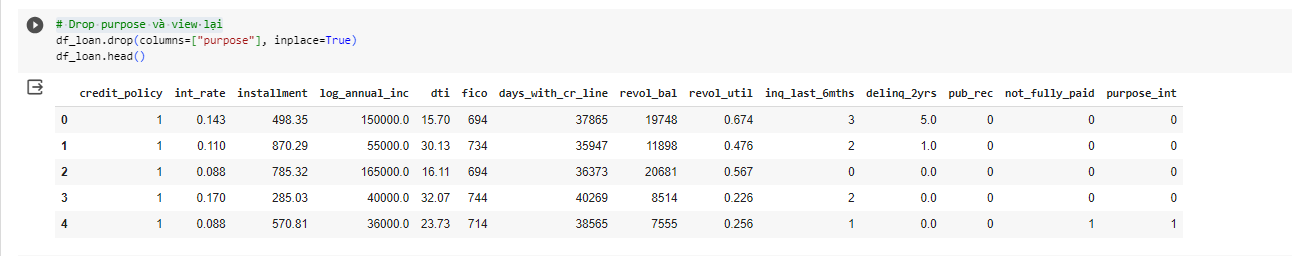
* Tìm hiểu qua về cột purpose



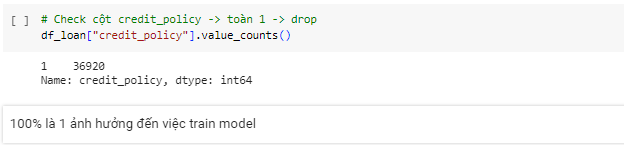
* Định nghĩa một hàm có tên là process\_purpose để xử lý cột "purpose". Hàm này nhận vào một giá trị và kiểm tra xem giá trị đó có phải là "debt\_consolidation" không. Nếu là "debt\_consolidation", hàm trả về 1, ngược lại trả về 0.
* Áp dụng hàm này cho cột "purpose" bằng cách sử dụng phương thức apply và tạo một cột mới có tên là "purpose\_int" để lưu trữ kết quả xử lý. Cụ thể, nếu mục đích là "debt\_consolidation", giá trị tương ứng trong cột mới sẽ là 1; ngược lại, sẽ là 0. Điều này giúp chuyển đổi dữ liệu mục đích thành dạng số, thuận tiện cho quá trình xây dựng mô hình.



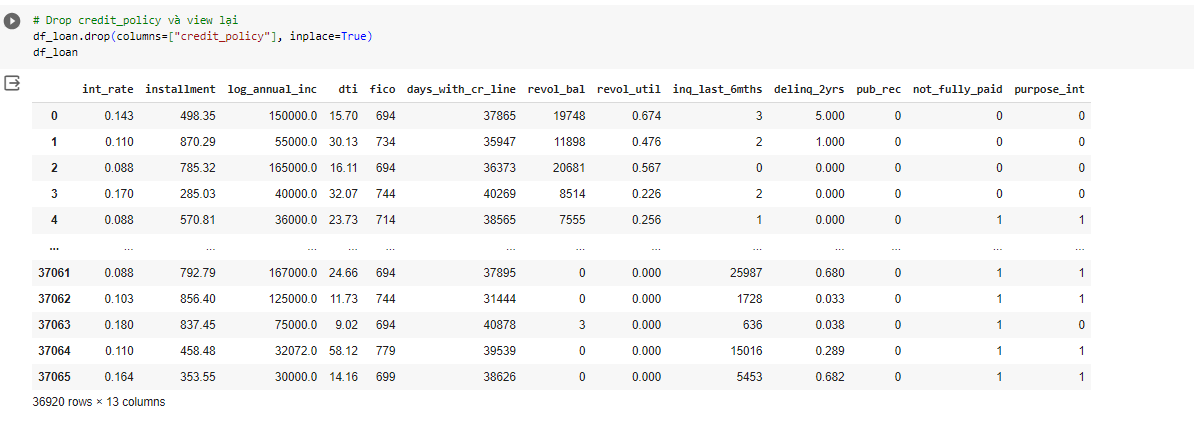
* Drop purpose và view lại



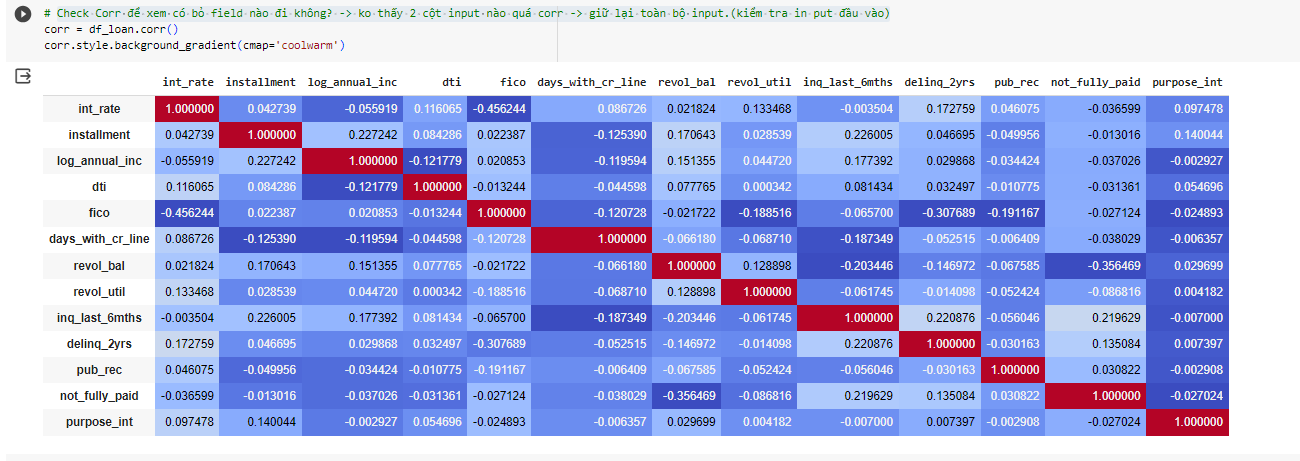
* Check cột credit\_policy -> toàn 1 -> drop



* Drop credit\_policy và view lại

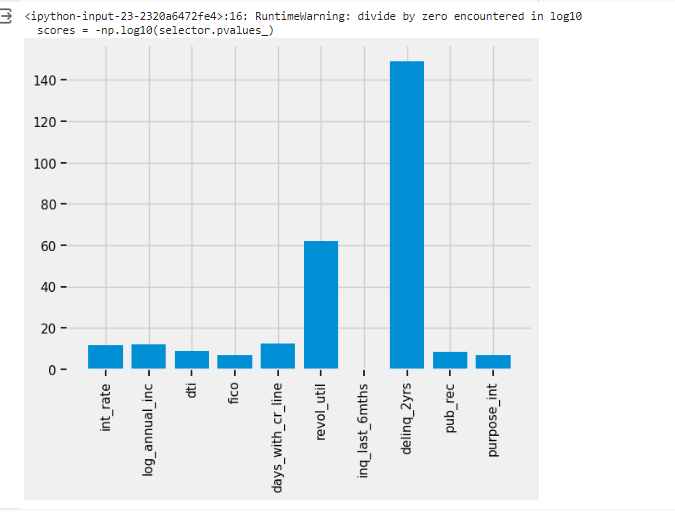


* Check Corr để xem có bỏ field nào đi không? -> ko thấy 2 cột input nào quá corr -> giữ lại toàn bộ input.(kiểm tra in put đầu vào)

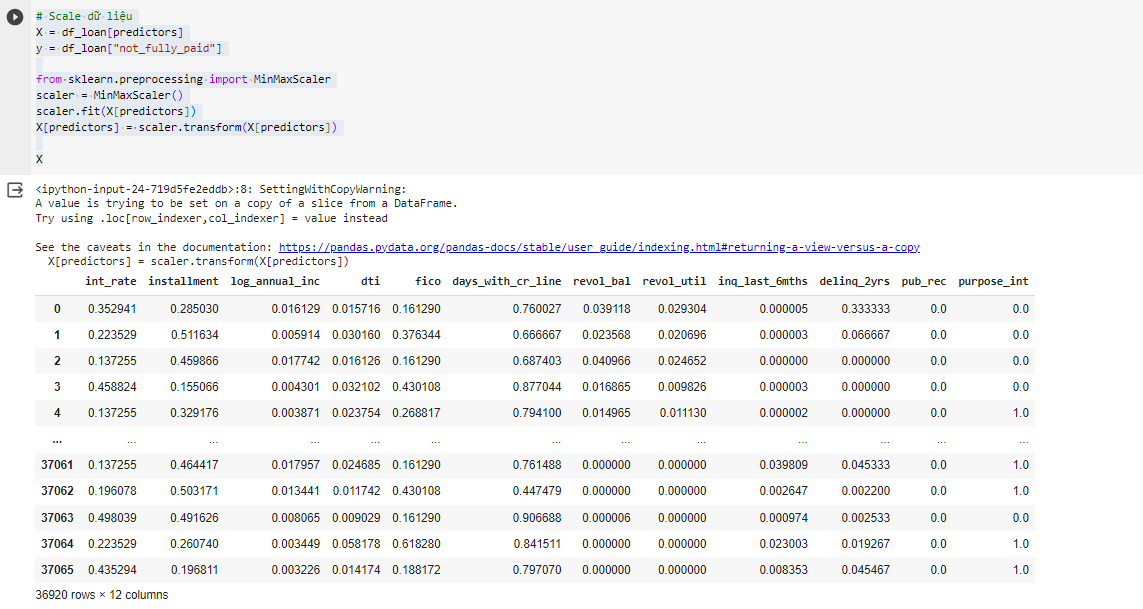


* thực hiện một quá trình lựa chọn đặc trưng (feature selection) bằng cách sử dụng phương pháp ANOVA (Analysis of Variance) thông qua lớp SelectKBest từ thư viện scikit-learn.
* Xác định đặc trưng dự đoán (predictors):
* Đặc trưng dự đoán là tất cả các cột trong DataFrame df\_loan ngoại trừ cột "not\_fully\_paid", tức là nhãn mục tiêu.
* Thực hiện lựa chọn đặc trưng:
* Sử dụng SelectKBest với phương pháp ANOVA và chọn 5 đặc trưng tốt nhất dựa trên các giá trị p.
* Chuyển đổi giá trị p thành điểm số và vẽ biểu đồ:

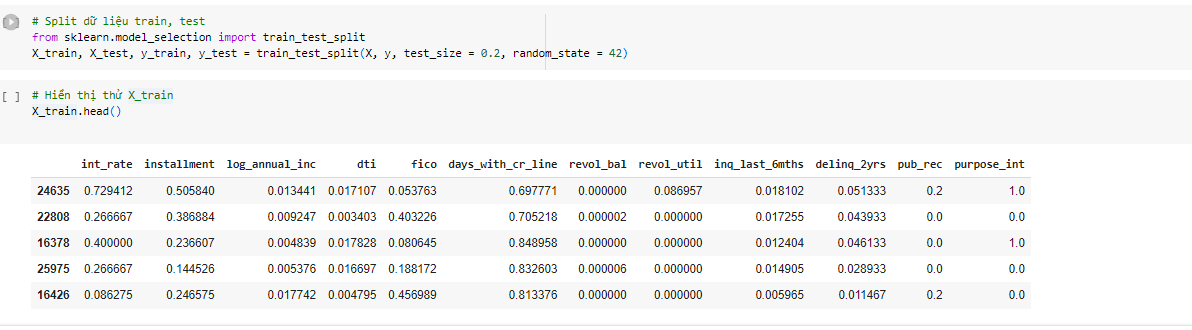




* thực hiện quá trình chuẩn hóa dữ liệu bằng cách sử dụng Min-Max Scaling. Dưới đây là chi tiết chức năng của mã:
* Chia dữ liệu thành đặc trưng (X) và nhãn (y):
* Tách DataFrame thành một tập hợp các đặc trưng (X) và một chuỗi nhãn (y).
* Chuẩn hóa dữ liệu:
* Sử dụng MinMaxScaler từ scikit-learn để thực hiện quá trình chuẩn hóa Min-Max trên tập hợp các đặc trưng.
* scaler.fit(X[predictors]) tính toán giá trị cần thiết để chuẩn hóa (min, max).
* X[predictors] = scaler.transform(X[predictors]) áp dụng quá trình chuẩn hóa cho các đặc trưng.

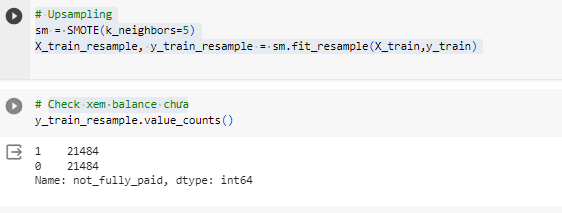


* Split dữ liệu train, test
* Hiển thị thử X\_train



1. **Train 01. Upsampling bằng SMOTE, train Logistic Regression**

* Sử dụng kỹ thuật Upsampling bằng phương pháp SMOTE (Synthetic Minority Over-sampling Technique). Dưới đây là mô tả về chức năng của mã:
* Import thư viện và khởi tạo SMOTE:
* SMOTE là một lớp từ thư viện imbalanced-learn được sử dụng để thực hiện kỹ thuật SMOTE.
* k\_neighbors=5 xác định số lượng láng giềng được sử dụng để tạo dữ liệu tổng hợp mới cho mỗi mẫu thiểu số.
* Thực hiện Upsampling:
* fit\_resample là phương thức của đối tượng SMOTE, thực hiện quá trình upsampling.
* X\_train\_resample và y\_train\_resample là dữ liệu đào tạo sau khi được upsampling.



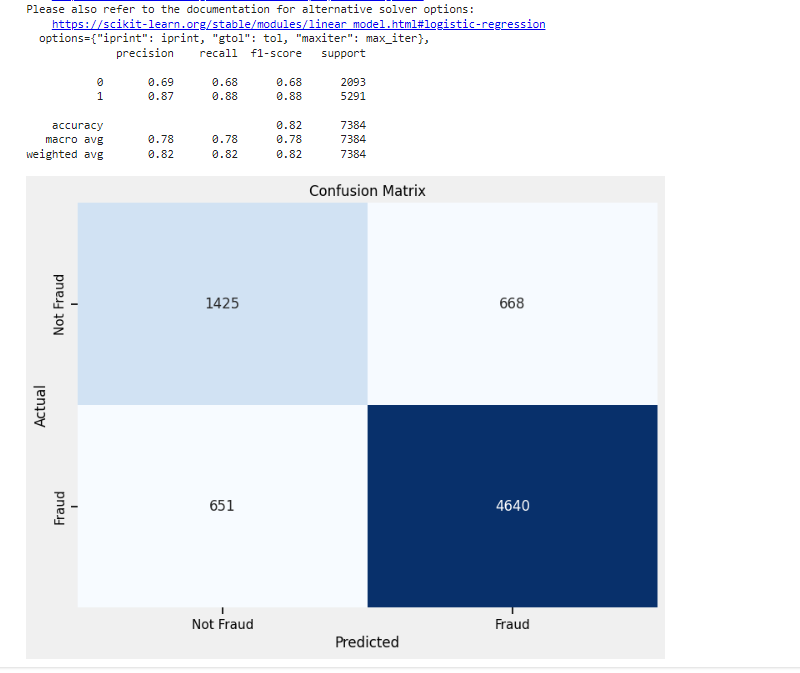
* Cài đặt thư viện seaborn, scikit-learn, và matplotlib bằng cách sử dụng lệnh pip trong môi trường Colab.



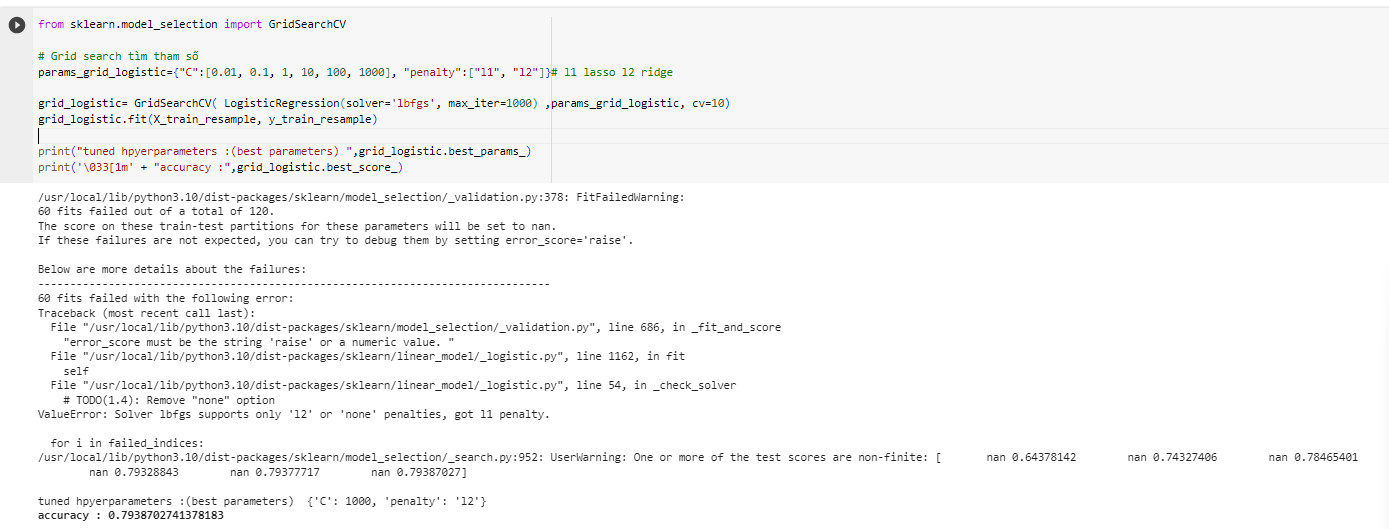


* Mô hình hóa và đánh giá mô hình Logistic Regression trong bối cảnh dữ liệu đã được upsampling bằng kỹ thuật SMOTE.
* Import thư viện và module cần thiết:
* from sklearn.metrics import classification\_report, confusion\_matrix: Import các hàm để đánh giá hiệu suất của mô hình.
* from sklearn.linear\_model import LogisticRegression: Import mô hình Logistic Regression.
* from sklearn.model\_selection import train\_test\_split: Import hàm để chia dữ liệu thành tập huấn luyện và tập kiểm tra.
* import seaborn as sns, matplotlib.pyplot as plt: Import thư viện để vẽ đồ thị.
* Chia dữ liệu thành tập huấn luyện và tập kiểm tra:

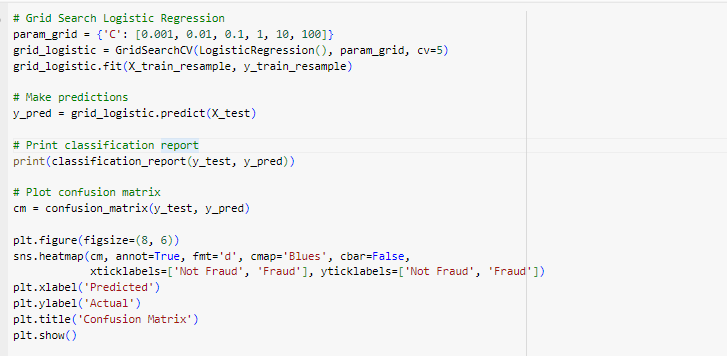


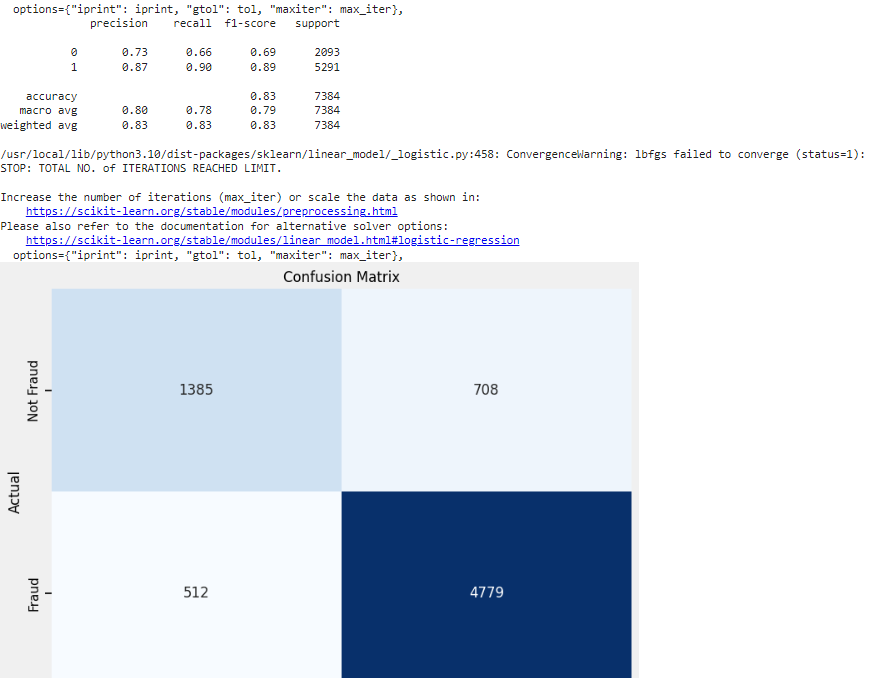


* Thực hiện quá trình tìm kiếm lưới (Grid Search) để tìm ra các tham số tối ưu cho mô hình Logistic Regression.



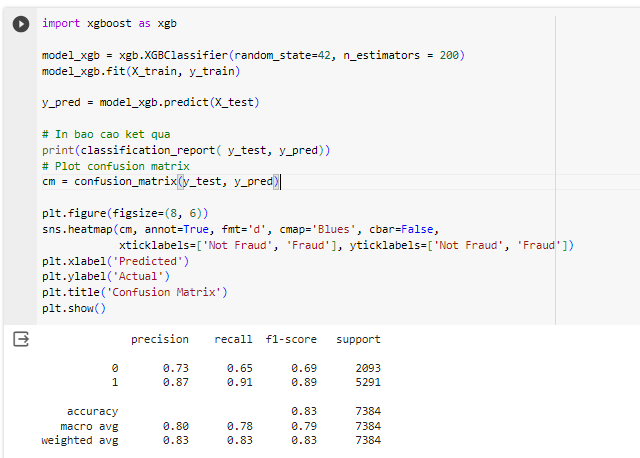
* Thực hiện Grid Search cho Logistic Regression với các giá trị khác nhau của tham số C và sau đó in ra báo cáo phân loại và ma trận nhầm lẫn.

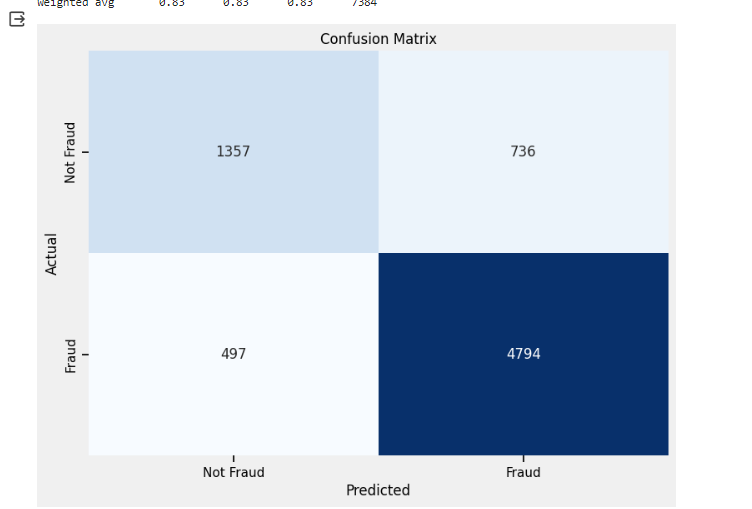




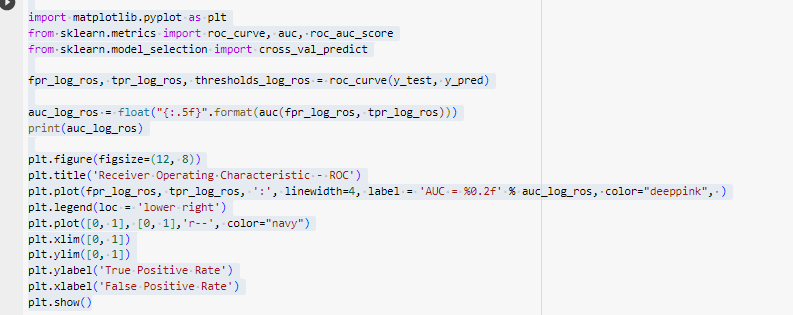
1. **Train 2. Không upsampling. XGBoost**

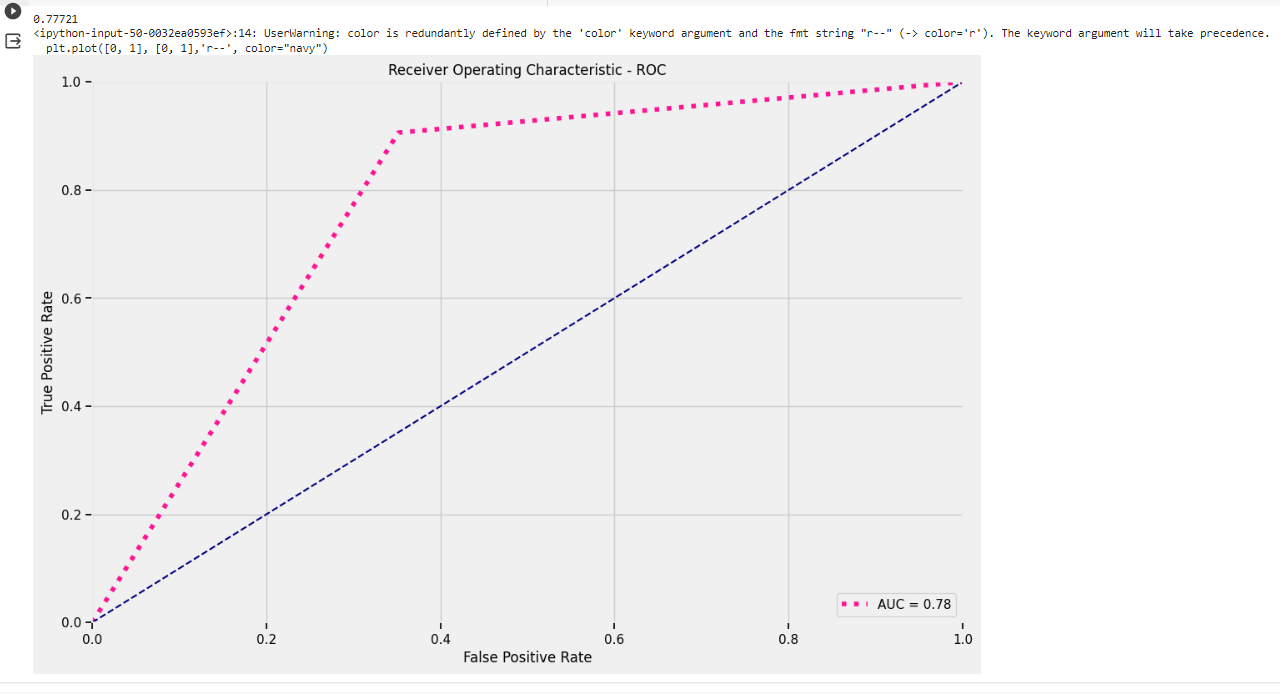
* Thực hiện việc sử dụng XGBoost để xây dựng mô hình phân loại và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.



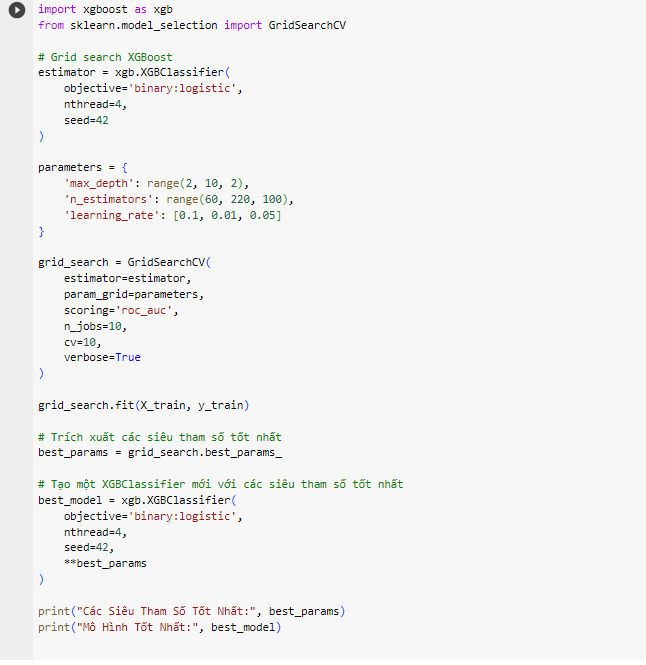


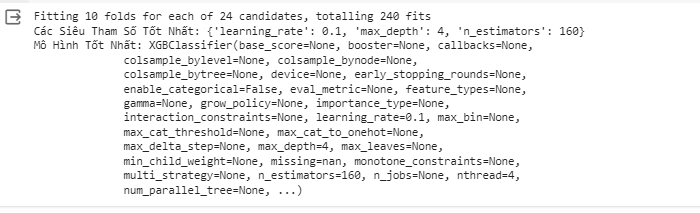
* Thực hiện việc vẽ đường cong ROC (Receiver Operating Characteristic) và tính diện tích dưới đường cong (AUC - Area Under the Curve) để đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.





* Thực hiện quá trình tìm kiếm siêu tham số tốt nhất cho mô hình XGBoost thông qua kỹ thuật Grid Search.





* Thực hiện lưu trữ mô hình XGBoost tốt nhất vào một file sử dụng thư viện pickle và sau đó tải lại mô hình từ file.
* Lưu trữ mô hình XGBoost tốt nhất vào Google Drive.

