

**UEH UNIVERSITY**  
**College of Economics, Law and Government**



**TIÊU LUẬN CUỐI KỲ**  
**MÔN HỌC: KỸ THUẬT LẬP TRÌNH VỚI STATA VÀ PYTHON**

**Đề tài:**

**Ứng dụng Python vào dự đoán khả năng hoàn trả khoản vay và phân tích các yếu tố ảnh hưởng đến rủi ro tín dụng của khách hàng**

**Giảng viên:**  
**TS.Nguyễn Khánh Duy, TS.Đỗ Như Tài**  
**Mã LHP: 25C1ECO50118801**

**Nhóm sinh viên:**  
**Trịnh Anh Đức – 31231022806**  
**Nguyễn Thái Đông – 31231023068**  
**Lưu Nhật Minh – 31231020989**  
**Nguyễn Hoàng Sơn – 31231023777**  
**Nguyễn Bảo Ngọc – 31231027314**

**Bảng phân công**

Họ và tên	Mức độ hoàn thành
Trịnh Anh Đức	100%
Lưu Nhật Minh	100%
Nguyễn Hoàng Sơn	100%
Nguyễn Bảo Ngọc	100%
Nguyễn Thái Đông	100%

# Mục lục

Contents .....	2
Danh mục hình ảnh .....	2
Danh mục bảng biểu .....	3
Chương I: Giới thiệu .....	4
1. Tổng quan nghiên cứu .....	4
2. Giới thiệu đề tài .....	5
3. Lý do chọn đề tài .....	5
4. Mục tiêu nghiên cứu .....	5
5. Đối tượng nghiên cứu .....	6
6. Nguồn dữ liệu .....	6
Chương II: Tiền xử lý dữ liệu .....	7
1. Mô tả bộ dữ liệu .....	7
2. Tiền xử lý dữ liệu .....	9
2.1. Làm sạch dữ liệu .....	9
2.2. Rút gọn dữ liệu .....	13
2.3. Tích hợp dữ liệu .....	13
2.4. Mã hóa dữ liệu .....	13
2.5. Chuẩn hóa dữ liệu .....	14
Chương III: Phân tích mô tả dữ liệu (Exploratory Data Analysis – EDA) .....	15
1. Thống kê mô tả và phân bố dữ liệu .....	15
2. Phân bố biến mục tiêu .....	16
3. Phân tích đặc trưng phân bố của các biến trong tập dữ liệu .....	17
3.1. Phân tích đặc trưng của các biến định lượng .....	17
3.2. Đặc trưng phân loại của biến định tính .....	18
4. Phân tích mối quan hệ giữa các biến .....	19
4.1. Mối quan hệ tuyến tính giữa các biến định lượng .....	19
4.2. Ma trận tương quan giữa các biến định lượng .....	21
5. Phân tích sâu các yếu tố ảnh hưởng đến khả năng trả nợ .....	23
6. Tổng kết chương .....	24
Chương IV: Tiến hành phân lớp dữ liệu .....	24
1. Kiểm tra tình trạng mất cân bằng dữ liệu .....	24
2. Chọn Phương pháp xử lý bất cân bằng .....	24
2.1. Chọn mô hình Logistic Regression cho giai đoạn đầu .....	24

2.2. Chạy với dữ liệu ban đầu .....	25
2.3. Under sampling(RUS).....	25
2.4. Neermiss(INS) .....	25
2.5. Random Oversampling(ROS) .....	25
2.6. SMOTE.....	26
2.7. Class weight.....	26
3.Phân lớp .....	27
3.1. XGBoost .....	27
3.2. Random Forest.....	32
4.Tổng kết bước phân lớp .....	34
Chương V: Kết luận (Conclusions & Discussion).....	35
1.Trả lời câu hỏi nghiên cứu .....	35
2.Nhận xét về hạn chế và hướng mở rộng. ....	36
2.1. Hạn chế .....	36
2.2. Hướng mở rộng.....	37
Tài liệu tham khảo: .....	38

## Danh mục hình ảnh

Hình 1 Thông tin cơ bản về bộ dữ liệu .....	4
Hình 2 Kết quả xem nhanh bộ dữ liệu .....	4
Hình 3 Kết quả kiểm tra Missing value .....	7
Hình 4 Kết quả kiểm tra giá trị trùng lặp .....	7
Hình 5 Kết quả kiểm tra với biểu đồ Boxplot.....	8
Hình 6 Kết quả kiểm tra bằng biểu đồ Histogram .....	9
Hình 7 Kết quả xử lý Outliers bằng LOF .....	10
Hình 8 Kết quả mã hóa dữ liệu .....	10
Hình 9 Kết quả chuẩn hóa dữ liệu .....	11
Hình 10 Thống kê mô tả các biến định lượng trong tập dữ liệu .....	12
Hình 11 Phân bố các biến định lượng chính trong tập dữ liệu. ....	14
Hình 12 Phân bố điểm tín dụng (FICO) theo trạng thái trả nợ. ....	15
Hình 13 Tỷ lệ không trả đủ nợ theo từng nhóm mục đích vay. ....	16
Hình 14 Mối quan hệ giữa các biến định lượng theo trạng thái trả nợ. ....	17

Hình 15 Xu hướng tuyến tính giữa điểm tín dụng và lãi suất vay theo trạng thái trả nợ. ....	18
Hình 16 Ma trận tương quan giữa các biến định lượng trong tập dữ liệu.....	19
Hình 17 Kết quả sự thay đổi của các biến định lượng theo tình trạng trả nợ .....	20
Hình 18 Tỷ lệ không trả đủ nợ theo mục đích vay .....	20
Hình 19 Kết quả chạy với dữ liệu ban đầu .....	22
Hình 20 Kết quả chạy với dữ liệu xử lý bởi RUS.....	22
Hình 21 Kết quả chạy với dữ liệu xử lý bởi INS .....	22
Hình 22 Kết quả chạy với dữ liệu xử lý bởi ROS.....	23
Hình 23 Kết quả chạy với dữ liệu xử lý bởi SMOTE .....	23
Hình 24 Kết quả chạy với dữ liệu xử lý bởi Class weight .....	23
Hình 25 Kết quả bộ siêu tham số tối ưu.....	24
Hình 26 Kết quả chạy cho mô hình XG Boost .....	26
Hình 27 Ma trận nhầm lẫn chạy với dữ liệu xử lý bởi XG Boost.....	27
Hình 28 Kết quả vẽ đường ROC Curve chạy với mô hình XG Boost.....	28
Hình 29 Kết quả các biến quan trọng chạy với mô hình XG Boost theo trọng số.....	28
Hình 30 Kết quả các biến quan trọng chạy với mô hình XG Boost .....	28
Hình 31 Ma trận nhầm lẫn chạy với dữ liệu xử lý bởi Random Forest .....	30
Hình 32 Kết quả vẽ đường ROC Curve chạy với mô hình Random Forest .....	30
Hình 33 Kết quả các biến quan trọng chạy với mô hình Random Forest theo trọng số .....	30
Hình 34 Kết quả các biến quan trọng chạy với mô hình Random Forest .....	31

## **Danh mục bảng biểu**

Bảng 1 Nội dung các biến đầu vào .....	6
Bảng 2 Kết quả tổng hợp kết quả chạy với dữ liệu được xử lý bởi các phương pháp.....	23
Bảng 3 Kết quả tổng hợp kết quả chạy với mô hình XG Boost và Random Forest .....	32
Biểu đồ 1 Biểu đồ phân bố biến mục tiêu trong tập dữ liệu .....	13

## Chương I: Giới thiệu

### 1. Tổng quan nghiên cứu

Trong bối cảnh chuyển đổi số mạnh mẽ của lĩnh vực tài chính – ngân hàng, việc áp dụng các kỹ thuật học máy (Machine Learning) vào phân tích và dự báo rủi ro tín dụng đang trở thành một hướng nghiên cứu chủ đạo. Rủi ro tín dụng – đặc biệt là khả năng người vay không hoàn trả đầy đủ khoản vay – được xem là một trong những yếu tố ảnh hưởng trực tiếp đến hiệu quả hoạt động, khả năng thanh khoản và sự bền vững tài chính của các tổ chức cho vay.

Các mô hình truyền thống trong đánh giá tín dụng, như mô hình điểm tín dụng tuyến tính (credit scoring models) dựa trên thống kê cổ điển, tuy dễ triển khai nhưng thường hạn chế trong việc xử lý mối quan hệ phi tuyến và phức tạp giữa các biến tài chính. Do đó, xu hướng gần đây trong nghiên cứu tài chính là chuyển dịch từ các mô hình thống kê tuyến tính sang mô hình học máy, cho phép hệ thống học từ dữ liệu để nhận diện các đặc trưng tiềm ẩn trong hành vi tài chính của người vay.

Nhiều công trình thực nghiệm đã chứng minh hiệu quả vượt trội của các thuật toán học máy trong việc dự đoán rủi ro vỡ nợ. Khandani et al. (2010) cũng chỉ ra rằng các mô hình như Support Vector Machine (SVM) và Gradient Boosting giúp cải thiện đáng kể khả năng phân loại giữa nhóm khách hàng có rủi ro cao và thấp.

Bên cạnh đó, các nghiên cứu thực tiễn từ các nền tảng cho vay ngang hàng (peer-to-peer lending) như LendingClub hay Prosper cũng chứng minh rằng các đặc điểm tài chính vi mô chẳng hạn như tỷ lệ nợ trên thu nhập (DTI), điểm tín dụng (FICO), thu nhập hàng năm, lịch sử thanh toán, và thời gian sử dụng tín dụng – có mối liên hệ mạnh mẽ với xác suất khách hàng không hoàn trả đầy đủ khoản vay (not fully paid).

Sự phát triển của Python và các thư viện chuyên biệt như pandas, scikit-learn, matplotlib, seaborn đã thúc đẩy đáng kể khả năng ứng dụng các mô hình học máy vào thực tiễn quản lý rủi ro tín dụng. Python không chỉ là công cụ mạnh mẽ để tiền xử lý và trực quan hóa dữ liệu, mà còn là nền tảng phổ biến cho việc xây dựng, huấn luyện và đánh giá mô hình dự báo trong quy mô lớn.

Từ cơ sở đó, nghiên cứu này kế thừa và phát triển hướng tiếp cận học máy hiện đại, tập trung vào việc dự đoán khả năng người vay không hoàn trả đầy đủ khoản vay dựa trên bộ dữ liệu Loan Data từ Kaggle, đồng thời phân tích các yếu tố ảnh hưởng chính đến rủi ro tín dụng. Cách tiếp cận này không chỉ có ý nghĩa học thuật – khi kiểm chứng hiệu năng giữa các thuật toán – mà còn mang tính thực tiễn cao, hỗ trợ các tổ chức tài chính nâng cao hiệu quả chấm điểm tín dụng và kiểm soát danh mục cho vay trong bối cảnh kinh tế số.

## 2. Giới thiệu đề tài

Đề tài “Ứng dụng Python trong dự đoán khả năng hoàn trả khoản vay và phân tích các yếu tố ảnh hưởng đến rủi ro tín dụng của khách hàng” hướng đến việc sử dụng các công cụ và thư viện trong Python (như pandas, scikit-learn, matplotlib, seaborn) để xử lý dữ liệu tín dụng, xây dựng mô hình học máy, đánh giá hiệu suất dự đoán và phân tích ảnh hưởng của các đặc điểm tài chính đến rủi ro tín dụng.

Bộ dữ liệu được sử dụng trong nghiên cứu được lấy từ các nền tảng cho vay trực tuyến (như LendingClub), bao gồm nhiều đặc trưng tài chính của khách hàng như điểm tín dụng, tỷ lệ nợ, thu nhập, thời gian sở hữu tài khoản tín dụng, và các chỉ số hành vi tài chính khác.

Mục tiêu chính của đề tài là xây dựng mô hình dự báo xác suất hoàn trả lãi vay của khách hàng, đồng thời xác định những yếu tố có ảnh hưởng mạnh nhất đến khả năng hoàn trả thông qua việc phân tích tầm quan trọng của các biến đầu vào.

## 3. Lý do chọn đề tài

Trong bối cảnh hoạt động tín dụng mở rộng nhanh chóng, đặc biệt là tín dụng tiêu dùng và cho vay trực tuyến, việc đánh giá chính xác khả năng hoàn trả của khách hàng trở thành một yêu cầu cấp thiết. Sai lệch trong quá trình thẩm định tín dụng có thể dẫn đến gia tăng nợ xấu, ảnh hưởng đến lợi nhuận và sự ổn định tài chính của các tổ chức cho vay.

Sự phát triển mạnh mẽ của dữ liệu lớn (Big Data) và học máy (Machine Learning) mở ra cơ hội mới trong việc xử lý khối lượng dữ liệu khổng lồ và nhận diện các mô hình hành vi tài chính phức tạp. Bằng cách ứng dụng Python – một ngôn ngữ lập trình mạnh mẽ và linh hoạt trong phân tích dữ liệu – nghiên cứu này không chỉ góp phần nâng cao khả năng dự đoán rủi ro tín dụng mà còn mang lại giá trị thực tiễn trong việc tối ưu hóa quy trình ra quyết định cho vay.

Bên cạnh đó, đề tài cũng giúp sinh viên củng cố kiến thức về khoa học dữ liệu, học máy, và tài chính ứng dụng – qua đó rèn luyện khả năng triển khai các mô hình dự báo trong môi trường thực tế.

## 4. Mục tiêu nghiên cứu

Nghiên cứu này hướng đến việc xây dựng và đánh giá các mô hình phân lớp trong Python để dự đoán khả năng không hoàn trả đầy đủ khoản vay dựa trên bộ dữ liệu Loan Data từ Kaggle, với biến phụ thuộc “not.fully.paid” biểu thị tình trạng khách hàng không hoàn trả đủ khoản vay.

Câu hỏi trung tâm của nghiên cứu được đặt ra là: “Dựa trên các đặc điểm tài chính và lịch sử tín dụng, liệu có thể xây dựng một mô hình học máy có khả năng dự đoán chính xác khả năng người vay không hoàn trả đầy đủ khoản vay hay không, và yếu tố nào có ảnh hưởng lớn nhất đến rủi ro này?”

Từ đó, nghiên cứu được triển khai với ba mục tiêu cụ thể sau:

- + Đánh giá khả năng áp dụng và mức độ chính xác của các mô hình học máy (như Logistic Regression, Random Forest, XGBoost) trong việc dự đoán rủi ro không hoàn trả khoản vay.
- + So sánh hiệu năng dự báo giữa các thuật toán học máy thông qua các chỉ số đánh giá như Accuracy, Precision, Recall, F1-score và ROC-AUC, nhằm xác định mô hình tối ưu nhất.
- + Phân tích tầm quan trọng của các biến đầu vào, nhằm xác định những đặc trưng tài chính có ảnh hưởng lớn nhất đến khả năng người vay không hoàn trả đầy đủ khoản vay.

Kết quả của nghiên cứu kỳ vọng sẽ cung cấp bằng chứng thực nghiệm giúp các tổ chức tài chính và nền tảng cho vay nâng cao hiệu quả quy trình chấm điểm tín dụng, tối ưu hóa quản lý rủi ro, và hỗ trợ ra quyết định cho vay chính xác hơn.

## **5. Đối tượng nghiên cứu**

Nghiên cứu tập trung vào khách hàng vay vốn trên nền tảng LendingClub trong giai đoạn 2007-2010. LendingClub, một trong những nền tảng cho vay ngang hàng lớn nhất tại Mỹ, cung cấp một kho dữ liệu phong phú về hành vi và đặc điểm tín dụng của các cá nhân. Việc phân tích dữ liệu từ LendingClub sẽ giúp chúng ta hiểu rõ hơn về đặc điểm của nhóm khách hàng này và các yếu tố ảnh hưởng đến khả năng trả nợ của họ.

## **6. Nguồn dữ liệu**

Bộ dữ liệu “Loan Data” là một tập dữ liệu lịch sử về khoản vay cá nhân, được công khai trên nền tảng Kaggle. Dữ liệu bao gồm các thông tin chi tiết về người đi vay như: Có đáp ứng tiêu chí thẩm định tín dụng không, mục đích khoản vay, lãi suất, khoản thanh toán hàng tháng, thu nhập năm của người vay, tỷ lệ nợ trên thu nhập, số lần chậm thanh toán... cùng biến mục tiêu có có thanh toán đầy đủ hay không.

Bộ dữ liệu này phù hợp cho phân tích tín dụng và xây dựng mô hình phân lớp nhằm dự đoán khả năng người đi vay không hoàn trả đầy đủ, đồng thời giúp khám phá các đặc điểm tài chính và lịch sử tín dụng nào có ảnh hưởng lớn tới rủi ro vỡ nợ.

Vì tính minh bạch và chi tiết trong các biến tài chính và tín dụng, bộ dữ liệu này cho phép thực hiện xuyên suốt các bước từ phân tích khám phá dữ liệu (EDA), tiền xử lý, kỹ thuật tạo biến, xử lý mất cân bằng lớp, huấn luyện mô hình và giải thích kết quả. Do đó, nó là lựa chọn lý tưởng cho nghiên cứu liên quan đến “credit risk modelling” trong lĩnh vực tín dụng tiêu dùng hoặc cho vay ngân hàng.

Link dữ liệu: [Loan data \(Kaggle\)](#)



## Chương II: Tiền xử lý dữ liệu

### 1. Mô tả bộ dữ liệu

Bộ dữ liệu này có 9578 quan sát và 14 thuộc tính, trong đó 13 thuộc tính đóng vai trò như các yếu tố giải thích cho biến mà chúng ta muốn dự đoán. Tập dữ liệu cụ thể gồm các thông tin sau:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   credit.policy          9578 non-null   int64
1   purpose                9578 non-null   object
2   int.rate               9578 non-null   float64
3   installment            9578 non-null   float64
4   log.annual.inc         9578 non-null   float64
5   dti                    9578 non-null   float64
6   fico                   9578 non-null   int64
7   days.with.cr.line      9578 non-null   float64
8   revol.bal              9578 non-null   int64
9   revol.util             9578 non-null   float64
10  inq.last.6mths         9578 non-null   int64
11  delinq.2yrs            9578 non-null   int64
12  pub.rec                9578 non-null   int64
13  not.fully.paid         9578 non-null   int64
dtypes: float64(6), int64(7), object(1)
memory usage: 1.0+ MB
```

Hình 1 Thông tin cơ bản về bộ dữ liệu

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs	pub.rec	not.fully.paid
0	1	debt_consolidation	0.1189	829.10	11.350407	19.48	737	5639.958333	28854	52.1	0	0	0	0
1	1	credit_card	0.1071	228.22	11.082143	14.29	707	2760.000000	33623	76.7	0	0	0	0
2	1	debt_consolidation	0.1357	366.86	10.373491	11.63	682	4710.000000	3511	25.6	1	0	0	0
3	1	debt_consolidation	0.1008	162.34	11.350407	8.10	712	2699.958333	33667	73.2	1	0	0	0
4	1	credit_card	0.1426	102.92	11.299732	14.97	667	4066.000000	4740	39.5	0	1	0	0

Hình 2 Kết quả xem nhanh bộ dữ liệu

Thuộc tính	Kiểu dữ liệu	Khoảng giá trị	Ý nghĩa
credit.policy	integer	“0”: Không đáp ứng “1”: Đáp ứng	Tiêu chí thẩm định tín dụng
purpose	object	all_other credit_card    debt_consolidation educational    home_improvement major_purchase small_business	Mục đích của khoản vay
int.rate	float	[0.06:0.22]	Lãi suất của khoản vay

installment	float	[15,7:940]	Các khoản trả góp hàng tháng mà người vay phải trả nếu khoản vay được giải ngân
log.annual.inc	float	[7.55:14.5]	Logarit tự nhiên của thu nhập hàng năm
dti	float	[0:30]	Tỷ lệ nợ trên thu nhập của người vay
fico	integer	[612:827]	Điểm tín dụng FICO của người vay
days.with.cr.line	float	[179:17600]	Số ngày kể từ khi người vay được cấp hạn mức tín dụng đến thời điểm hiện tại
revol.bal	integer	[0:1210000]	Số dư luân chuyển của người vay
revol.util	float	[0:119]	Tỷ lệ sử dụng hạn mức tín dụng luân chuyển của người vay
inq.last.6mths	integer	[0:33]	Số lần người vay bị chủ nợ yêu cầu thanh toán trong 6 tháng qua
delinq.2yrs	integer	[0:13]	Số lần người vay quá hạn thanh toán 30 ngày trở lên trong 2 năm qua.
pub.rec	integer	[0:5]	Số hồ sơ công khai có dấu hiệu bất lợi của người vay

not.fully.paid	integer	“0”:Không “1”: Có	Trạng thái không thanh toán đầy đủ
----------------	---------	----------------------	---------------------------------------

*Bảng 1 Nội dung các biến đầu vào*

Bộ dữ liệu cho vay này có 9.578 quan sát và 14 biến, với một ưu điểm lớn là không có bất kỳ giá trị thiếu (missing values) nào trên tất cả các cột. Về kiểu dữ liệu, phần lớn là các biến số định lượng (float64 và int64), bao gồm các chỉ số tài chính và rủi ro quan trọng như int.rate (lãi suất), fico (điểm tín dụng), và log.annual.inc (thu nhập). Tuy nhiên, có một cột danh mục là purpose (kiểu object), cột này sẽ cần được mã hóa (như One-Hot Encoding) thành định dạng số trước khi đưa vào các mô hình học máy. Cuối cùng, biến mục tiêu not.fully.paid thuộc kiểu int64 (chứa giá trị 0/1), xác nhận việc có thể dùng thuật toán phân lớp để dự đoán rủi ro vỡ nợ và xem xét đặc tính nào ảnh hưởng lớn nhất đến rủi ro đó.

## 2. Tiền xử lý dữ liệu

### 2.1. Làm sạch dữ liệu

#### 2.1.1. Xử lý giá trị bị thiếu

Xử lý giá trị bị thiếu (Missing Value Handling) là quá trình phát hiện và khắc phục các ô dữ liệu trống, null hoặc NaN trong bộ dữ liệu. Đây là bước rất quan trọng vì giá trị bị thiếu có thể làm sai lệch kết quả phân tích, giảm độ chính xác của mô hình học máy hoặc khiến một số thuật toán không thể hoạt động. Việc xử lý có thể được thực hiện bằng nhiều cách khác nhau như: loại bỏ các hàng hoặc cột chứa nhiều giá trị thiếu, thay thế (impute) giá trị bị thiếu bằng trung bình, trung vị, mode hoặc giá trị dự đoán, hoặc giữ nguyên nếu giá trị thiếu mang ý nghĩa đặc biệt.


	Số lượng thiếu	Tỷ lệ thiếu (%)
credit.policy	0	0.0
purpose	0	0.0
int.rate	0	0.0
installment	0	0.0
log.annual.inc	0	0.0
dti	0	0.0
fico	0	0.0
days.with.cr.line	0	0.0
revol.bal	0	0.0
revol.util	0	0.0
inq.last.6mths	0	0.0
delinq.2yrs	0	0.0
pub.rec	0	0.0
not.fully.paid	0	0.0

Hình 3 Kết quả kiểm tra Missing value

Kết quả kiểm tra cho thấy bộ dữ liệu Loan\_data là bộ dữ liệu tốt và không bị thiếu các giá trị. Vì vậy không cần tiến hành xử lý Missing value.

### 2.1.2. Xử lý giá trị trùng lặp

Xử lý giá trị trùng lặp (Duplicate Value Handling) là quá trình phát hiện và loại bỏ các bản ghi giống hệt nhau hoặc gần giống nhau trong bộ dữ liệu. Đây là bước quan trọng trong tiền xử lý dữ liệu vì các giá trị trùng lặp có thể làm sai lệch kết quả thống kê, ảnh hưởng đến hiệu suất và độ chính xác của các mô hình phân tích hoặc học máy. Việc xử lý có thể thực hiện bằng cách xác định và loại bỏ các hàng trùng hoàn toàn, hoặc kiểm tra các trường hợp trùng lặp theo một hoặc một vài cột quan trọng. Mục tiêu của quá trình này là đảm bảo dữ liệu chỉ phản ánh mỗi quan sát một lần, giúp kết quả phân tích trở nên đáng tin cậy, chính xác và có ý nghĩa hơn.

 Bộ dữ liệu không có hàng nào bị trùng lặp hoàn toàn.

Hình 4 Kết quả kiểm tra giá trị trùng lặp

Kết quả kiểm tra cho thấy bộ dữ liệu không tồn tại các quan sát có dữ liệu giống nhau hoàn toàn, nên không cần phải xử lý dữ liệu trùng lặp.

### 2.1.3. Kiểm tra và xử lý giá trị ngoại lai (outliers)

Các giá trị ngoại lai (outliers) là những điểm dữ liệu nằm ngoài phạm vi của phần lớn dữ liệu trong một tập dữ liệu cụ thể. Những giá trị này có thể cao hơn hoặc thấp hơn đáng kể so với các điểm dữ liệu khác và có thể ảnh hưởng đến kết quả phân tích dữ liệu theo cách làm sai lệch mẫu dữ liệu. Bằng cách học cách xác định và xử lý các giá trị ngoại lai, các nhà phân tích dữ liệu có thể tăng khả năng phân tích phản ánh chính xác tính hợp lệ và độ tin cậy của kết quả.

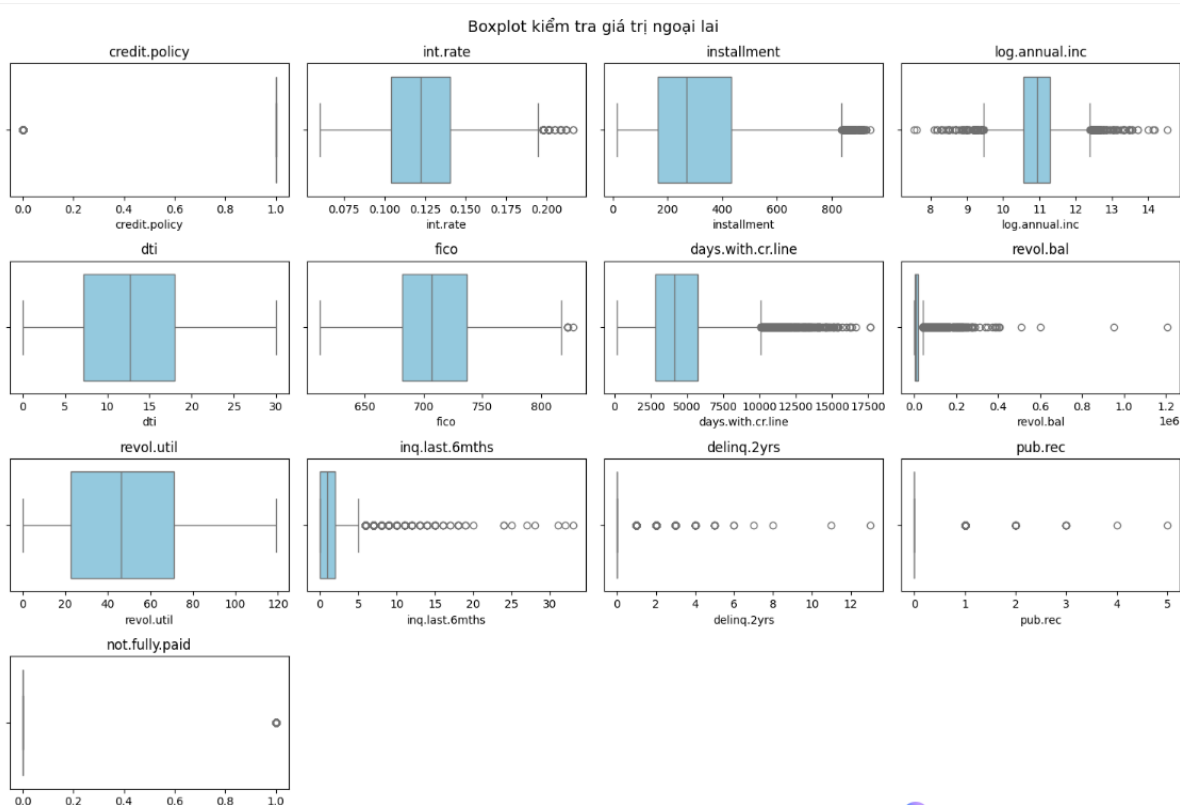
Các cách xử lý giá trị ngoại lai :

- + SVM một lớp với kernel phi tuyến (RBF): Phân loại dữ liệu thành giống hoặc khác so với lớp chính. Phương pháp này hoạt động tốt nhất với bộ dữ liệu có phân phối không chuẩn.
- + Ước tính hiệp phương sai (Covariance estimator): Phù hợp với việc tập trung các điểm trung tâm bằng cách sử dụng thước đo khoảng cách Mahalanobi. Phương pháp này hoạt động tốt nhất với bộ dữ liệu có phân phối chuẩn.
- + Hệ số ngoại lai cục bộ (Local Outlier Factor): Tính toán mật độ cục bộ từ các điểm lân cận gần nhất (k-nearest neighbors). Đây là phương pháp hiệu quả thường sử dụng để phát hiện giá trị ngoại lai trên các tập dữ liệu đa chiều và được sử dụng nhiều nhất.

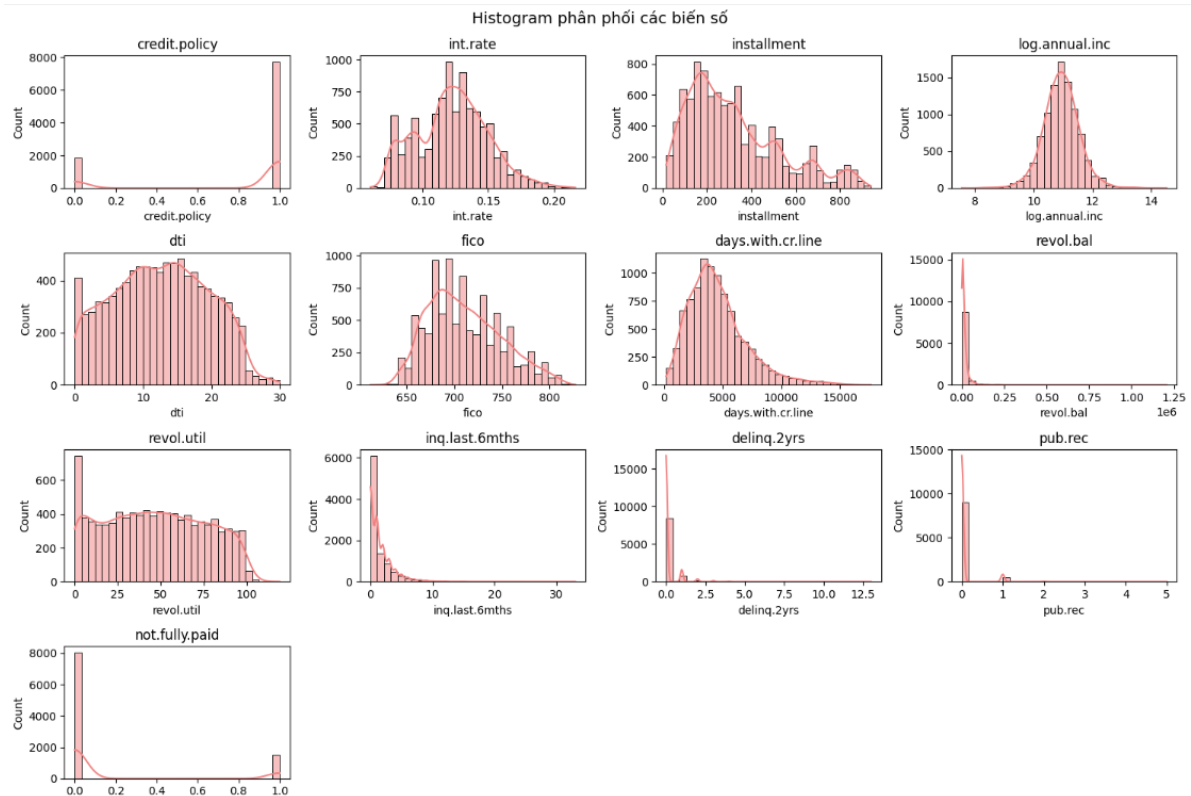
Để lựa chọn phương pháp hiệu quả và tối ưu, ta sẽ phân tích sơ bộ về đặc điểm của bộ dữ liệu

Các bước thực hiện

1. Kiểm tra phân bố của dữ liệu bằng Biểu đồ Boxplot và Biểu đồ Histogram để xem xét xem bộ dữ liệu có tồn tại giá trị ngoại lai không.



Hình 5 Kết quả kiểm tra với biểu đồ Boxplot



*Hình 6 Kết quả kiểm tra bằng biểu đồ Histogram*

Qua quan sát biểu đồ boxplot, ta nhận thấy nhiều biến như installment, int.rate, days.with.cr.line, inq.last.6mths, và revol.bal có xuất hiện nhiều điểm ngoại lai nằm xa khỏi hộp dữ liệu chính, cho thấy sự tồn tại của các giá trị bất thường trong tập dữ liệu. Bên cạnh đó, biểu đồ phân phối (histogram) cho thấy phần lớn các biến có phân phối lệch phải (right-skewed) hoặc không tuân theo phân phối chuẩn, điều này khiến cho các phương pháp truyền thống như IQR (Interquartile Range) hay Z-score kém hiệu quả, vì chúng giả định dữ liệu có phân phối chuẩn hoặc đối xứng.

Do đó, nhóm lựa chọn phương pháp hệ số ngoại lai cục bộ (Local Outlier Factor – LOF) để xử lý giá trị ngoại lai. Phương pháp này không dựa trên phân phối toàn cục của dữ liệu mà đánh giá mức độ “cô lập” của từng điểm dữ liệu so với mật độ lân cận cục bộ. Cách tiếp cận này giúp phát hiện các điểm bất thường trong những cụm có mật độ khác nhau, phù hợp với đặc điểm phân phối không đồng nhất của dữ liệu hiện tại. Đồng thời, LOF có khả năng xử lý các trường hợp ngoại lai phức tạp đa chiều, giúp duy trì cấu trúc dữ liệu và giảm thiểu mất mát thông tin so với việc loại bỏ giá trị bằng ngưỡng cứng (như IQR). Từ đó, nhóm quyết định áp dụng LOF làm phương pháp tối ưu để phát hiện và loại bỏ các giá trị ngoại lai trước khi tiến hành huấn luyện mô hình.

## 2. Loại Outliers

```
Tổng số hàng được LOF xác định là ngoại lai (chỉ dựa trên các cột số): 958
Kích thước DataFrame gốc (df): (9578, 14)
Kích thước DataFrame sau khi loại bỏ ngoại lai:(8620, 14)
```

*Hình 7 Kết quả xử lý Outliers bằng LOF*

Quá trình làm sạch này đã loại bỏ thành công 858 điểm dữ liệu bất thường dựa trên mật độ cục bộ. Bộ dữ liệu (8620, 14) hiện đã sạch hơn và sẵn sàng cho các bước tiền xử lý quan trọng tiếp theo.

## 2.2. Rút gọn dữ liệu

Với bộ dữ liệu chứa 8763 quan sát, số lượng này đủ lớn để sử dụng trong quá xây dựng và đánh giá mô hình. Vậy nên nhóm quyết định không thực hiện bước rút gọn dữ liệu này.

## 2.3. Tích hợp dữ liệu

Qua quá trình đánh giá, nhóm kết luận rằng bộ dữ liệu gồm 8767 quan sát, không tồn tại giá trị bị thiếu đã đáp ứng đủ các tiêu chí về chất lượng và số lượng để tiến hành phân tích. Việc bổ sung thêm dữ liệu vào thời điểm này có thể gây ra tình trạng nhiễu dữ liệu, ảnh hưởng đến độ tin cậy của kết quả nghiên cứu. Vì vậy, nhóm quyết định không tích hợp thêm dữ liệu.

## 2.4. Mã hóa dữ liệu

Trong bộ dữ liệu, nhóm nhận thấy biến `purpose` là biến phân loại (categorical), thể hiện mục đích vay vốn của khách hàng (như vay để mua nhà, vay tiêu dùng, vay học tập,...). Do các mô hình học máy chỉ có thể xử lý dữ liệu dạng số, nên việc mã hóa biến phân loại là bước cần thiết để chuyển đổi thông tin này sang dạng phù hợp cho quá trình huấn luyện mô hình.

Có nhiều phương pháp mã hóa dữ liệu như Label Encoding, One-Hot Encoding hay Target Encoding. Tuy nhiên, vì biến `purpose` không có quan hệ thứ tự giữa các giá trị (ví dụ “vay mua nhà” không hơn “vay học tập”), nên nhóm đã lựa chọn phương pháp One-Hot Encoding. Phương pháp này giúp tạo ra các biến giả (dummy variables) đại diện cho từng mục đích vay, đảm bảo rằng mô hình không hiểu sai các giá trị phân loại như các giá trị có tính thứ tự.

Việc áp dụng One-Hot Encoding cho biến `purpose` giúp mô hình hiểu rõ hơn về mối quan hệ giữa mục đích vay và khả năng hoàn trả, đồng thời giữ nguyên bản chất định tính của dữ liệu mà không gây sai lệch trong quá trình phân tích hay dự báo.

kích thước DataFrame gốc (df\_cleaned): (8620, 14)  
kích thước DataFrame sau khi mã hóa: (8620, 17)

5 hàng đầu tiên của DataFrame đã được mã hóa:

	credit.policy	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	log.last.6mths	delinq.2yrs	pub.rec	net.fully.paid	purpose_credit_card	purpose_debt_consolidation	purpose_educational	purpose_home_improvement	purpose_major_purchase	purpose_small_business
0	1	0.1189	829.10	11.350407	19.43	737	5639.950333	20854	52.1	0	0	0	0	False	True	False	False	False	False
2	1	0.1357	368.96	10.373491	11.63	682	4710.000000	3511	25.6	1	0	0	0	False	True	False	False	False	False
4	1	0.1426	162.92	11.299732	14.97	667	4066.000000	4740	39.5	0	1	0	0	True	False	False	False	False	False
5	1	0.0788	125.13	11.904968	16.98	727	6120.041667	56807	51.0	0	0	0	0	True	False	False	False	False	False
6	1	0.1496	194.02	10.714410	4.09	667	3180.041667	3839	76.0	0	0	1	1	False	True	False	False	False	False

*Hình 8 Kết quả mã hóa dữ liệu*

## 2.5. Chuẩn hóa dữ liệu.

Trong quá trình tiền xử lý, nhóm tiến hành chuẩn hóa dữ liệu để đảm bảo các biến có quy mô (scale) tương đồng, tránh tình trạng những biến có giá trị lớn chi phối mô hình. Việc chuẩn hóa đặc biệt quan trọng trong các mô hình dựa trên khoảng cách hoặc có trọng số (như hồi quy logistic, SVM, KNN, hay mô hình phân lớp tuyến tính).

Cụ thể, nhóm lựa chọn chuẩn hóa các biến số liên tục, tức là những biến có thể nhận bất kỳ giá trị nào trong một khoảng và có phạm vi giá trị rộng. Các biến được chuẩn hóa gồm:

- + Lãi suất (int.rate)
- + Khoản trả góp (installment)
- + Thu nhập hằng năm (log.annual.inc)
- + Tỷ lệ nợ trên thu nhập (dti)
- + Điểm tín dụng (fico)
- + Số ngày có hạn mức tín dụng (days.with.cr.line)
- + Số dư quay vòng (revol.bal)
- + Tỷ lệ sử dụng quay vòng (revol.util)

Nhóm sử dụng phương pháp chuẩn hóa Z-score, giúp các biến có trung bình bằng 0 và độ lệch chuẩn bằng 1 (mean=0, std=1). Cách chuẩn hóa này giúp mô hình học máy đánh giá các biến một cách công bằng, không bị ảnh hưởng bởi độ lớn tuyệt đối của giá trị. Đối với các biến không được chuẩn hóa, nhóm giữ nguyên giá trị gốc vì chúng mang ý nghĩa phân loại hoặc định lượng rời rạc:

- + Biến nhị phân (0/1): credit.policy, not.fully.paid — đã thể hiện trạng thái rõ ràng, việc chuẩn hóa có thể làm mất ý nghĩa logic của 0 và 1.
- + Biến đếm rời rạc: inq.last.6mths, delinq.2yrs, pub.rec — thể hiện số lần xảy ra của một sự kiện, nên không cần chuẩn hóa để tránh mất đi khả năng diễn giải.
- + Các biến mã hóa từ One-Hot Encoding: các cột purpose\_... — là biến nhị phân đại diện cho từng mục đích vay vốn, việc chuẩn hóa sẽ làm sai lệch bản chất nhị phân này.

Như vậy, việc chỉ chuẩn hóa các biến liên tục và giữ nguyên các biến phân loại hoặc rời rạc giúp đảm bảo độ chính xác, tính diễn giải và hiệu quả huấn luyện mô hình.

Kích thước dataframe cuối cùng: (8820, 29)

5 hàng đầu tiên (kiểm tra các cột số đã được chuẩn hóa):

	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	credit.policy	inq.last.6mths	delinq.2yrs	pub.rec	not.fully.paid	purpose_credit_card	purpose_debt_consolidation	purpose_educational	purpose_home_improvement	purpose_major_purchase	purpose_small_business
0	-0.100786	2.794789	0.731371	0.864714	0.703853	0.515203	0.555306	0.160593	1	0	0	0	0	False	True	False	False	False	False
2	0.531025	0.263875	-0.679439	-0.161799	-0.754209	0.099773	-0.580379	-0.750302	1	1	0	0	0	False	True	False	False	False	False
4	0.739653	-1.025382	0.647816	0.326816	-1.154705	-0.187973	-0.449166	-0.290353	1	0	1	0	0	True	False	False	False	False	False
5	-1.606804	-0.900598	1.645772	0.819582	0.437001	0.729795	1.470143	0.130447	1	0	0	0	0	True	False	False	False	False	False
6	1.053782	-0.540157	-0.317294	-1.276101	-1.154705	-0.503029	-0.486795	1.025147	1	0	0	1	1	False	True	False	False	False	False

Hình 9 Kết quả chuẩn hóa dữ liệu



### Chương III: Phân tích mô tả dữ liệu (Exploratory Data Analysis – EDA)

Phân tích mô tả dữ liệu (EDA) được tiến hành nhằm khám phá đặc điểm tổng thể của tập dữ liệu, xác định xu hướng, mối tương quan giữa các biến, và phát hiện những quy luật tiềm ẩn có thể ảnh hưởng đến khả năng khách hàng trả nợ. Bước này đóng vai trò nền tảng cho mô hình dự báo được xây dựng ở phần sau, đồng thời cung cấp bằng chứng sơ bộ cho câu hỏi nghiên cứu về việc những yếu tố nào có ảnh hưởng đáng kể đến khả năng không trả đủ nợ của khách hàng.

#### 1. Thống kê mô tả và phân bố dữ liệu

Kết quả thống kê mô tả cho thấy các biến định lượng trong tập dữ liệu có sự phân tán đáng kể. Điểm tín dụng (fico) trung bình đạt 710 và dao động trong khoảng từ 612 đến 827, phản ánh mức độ tín nhiệm tương đối cao của đa số khách hàng. Lãi suất vay (int.rate) trung bình khoảng 12,2%, với giá trị nhỏ nhất là 6% và cao nhất lên đến 21,6%, thể hiện sự khác biệt rõ rệt về rủi ro giữa các nhóm khách hàng. Tỷ lệ nợ trên thu nhập (dti) trung bình là 12,7%, với độ lệch chuẩn lớn cho thấy khả năng tài chính giữa các khách hàng không đồng đều. Thu nhập hàng năm (log.annual.inc) có giá trị trung bình khoảng 10,9 (tương ứng với khoảng 55.000 đến 60.000 USD/năm) và biến động trong phạm vi hợp lý.

	Trung bình	Trung vị	Độ lệch chuẩn	Giá trị nhỏ nhất	Giá trị lớn nhất
credit.policy	0.814	1.000	0.389	0.000	1.000
int.rate	0.122	0.122	0.027	0.060	0.216
installment	300.213	258.570	191.708	15.670	926.830
log.annual.inc	10.909	10.915	0.607	7.548	14.528
dti	12.728	12.800	6.855	0.000	29.960
fico	710.558	707.000	37.496	612.000	827.000
days.with.cr.line	4503.152	4140.042	2268.517	178.958	14167.000
revol.bal	15698.093	8652.000	24600.446	0.000	242194.000
revol.util	47.241	46.700	28.829	0.000	119.000
inq.last.6mths	1.549	1.000	2.140	0.000	33.000
delinq.2yrs	0.167	0.000	0.553	0.000	13.000
pub.rec	0.064	0.000	0.267	0.000	5.000
not.fully.paid	0.155	0.000	0.362	0.000	1.000

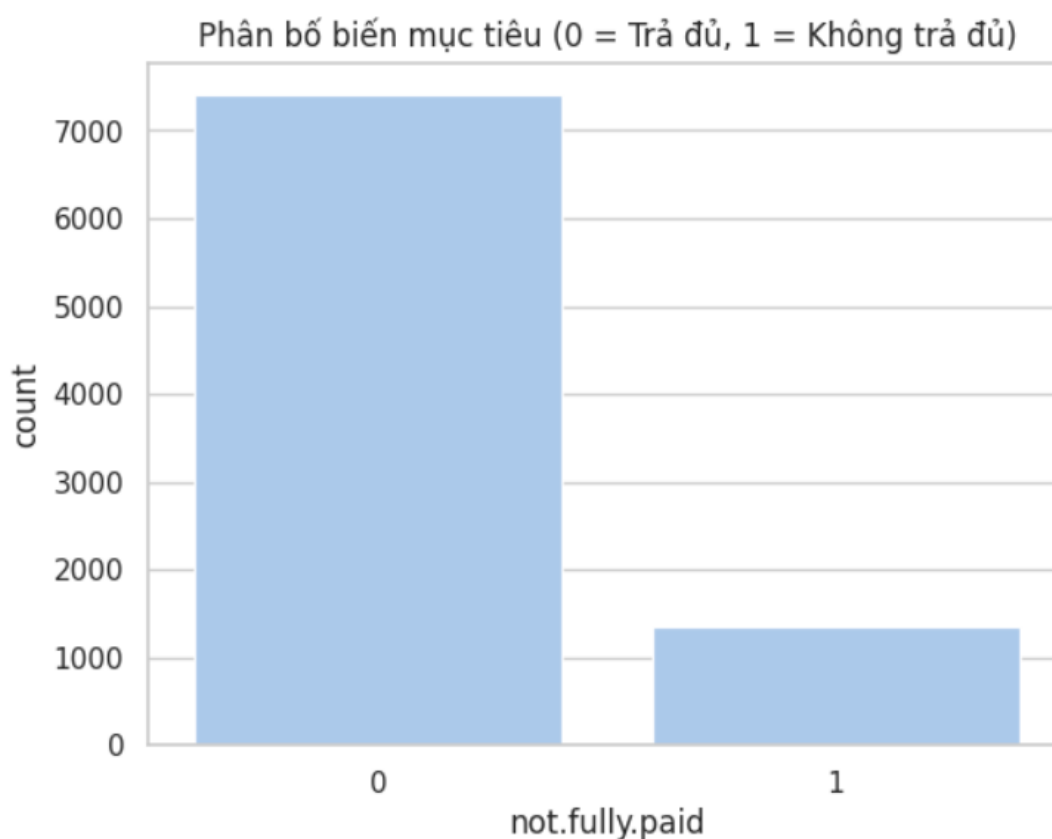
Hình 10 Thống kê mô tả các biến định lượng trong tập dữ liệu

Những kết quả trên phản ánh đặc điểm đa dạng của tập khách hàng vay vốn. Sự khác biệt rõ rệt về điểm tín dụng và lãi suất cho thấy đây có thể là hai yếu tố quan trọng quyết

định khả năng hoàn trả nợ, đồng thời gợi ý rằng việc phân tích mối quan hệ giữa chúng với biến mục tiêu là cần thiết để kiểm chứng giả thuyết nghiên cứu.

## 2. Phân bố biến mục tiêu

Phân tích phân bố biến mục tiêu cho thấy tỷ lệ khách hàng trả nợ đầy đủ chiếm khoảng 84,5%, trong khi nhóm không trả đủ nợ chỉ chiếm khoảng 15,5%. Điều này cho thấy tập dữ liệu có hiện tượng mất cân bằng lớp, tức là nhóm khách hàng rủi ro thấp (trả nợ đủ) chiếm đa số.



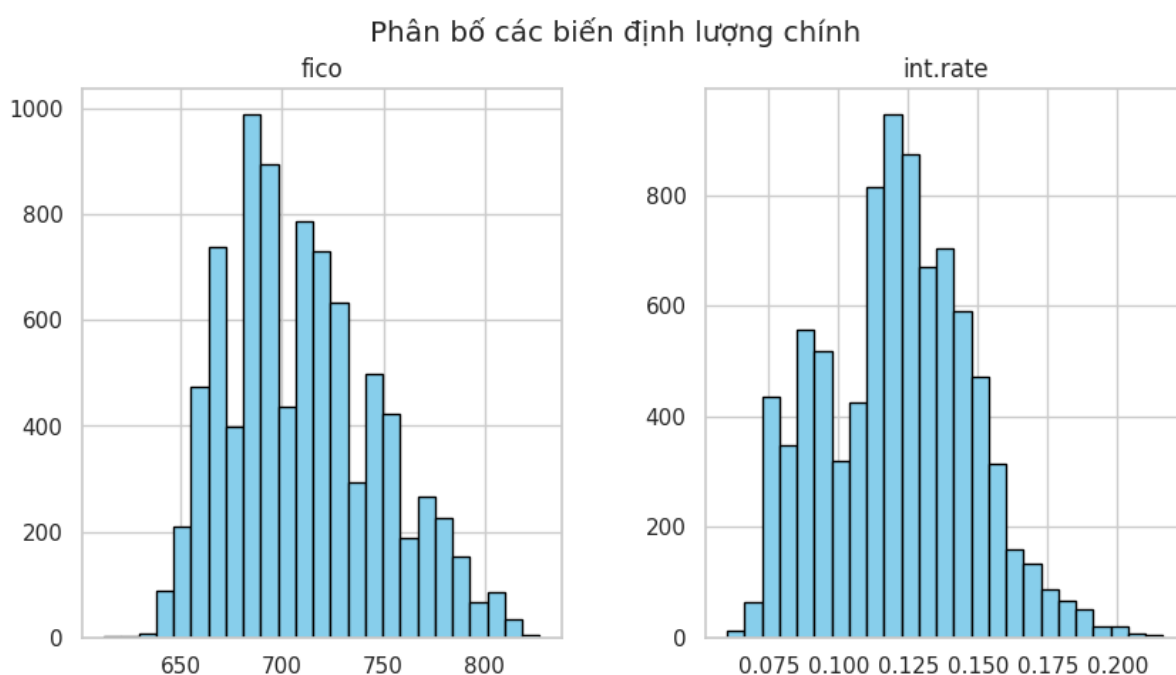
*Biểu đồ 1 Biểu đồ phân bố biến mục tiêu trong tập dữ liệu*

Về mặt phân tích, sự mất cân bằng này mang ý nghĩa quan trọng. Nếu không được xử lý trong mô hình dự đoán, nó có thể khiến thuật toán học máy thiên lệch về nhóm khách hàng “an toàn” và bỏ qua các trường hợp rủi ro cao – là nhóm mà nghiên cứu đặc biệt quan tâm. Tuy nhiên, điều này cũng phản ánh một thực tế trong tín dụng: phần lớn khách hàng đều trả nợ đúng hạn, và chỉ một tỷ lệ nhỏ không hoàn thành nghĩa vụ tài chính.

### 3. Phân tích đặc trưng phân bố của các biến trong tập dữ liệu

#### 3.1. Phân tích đặc trưng của các biến định lượng

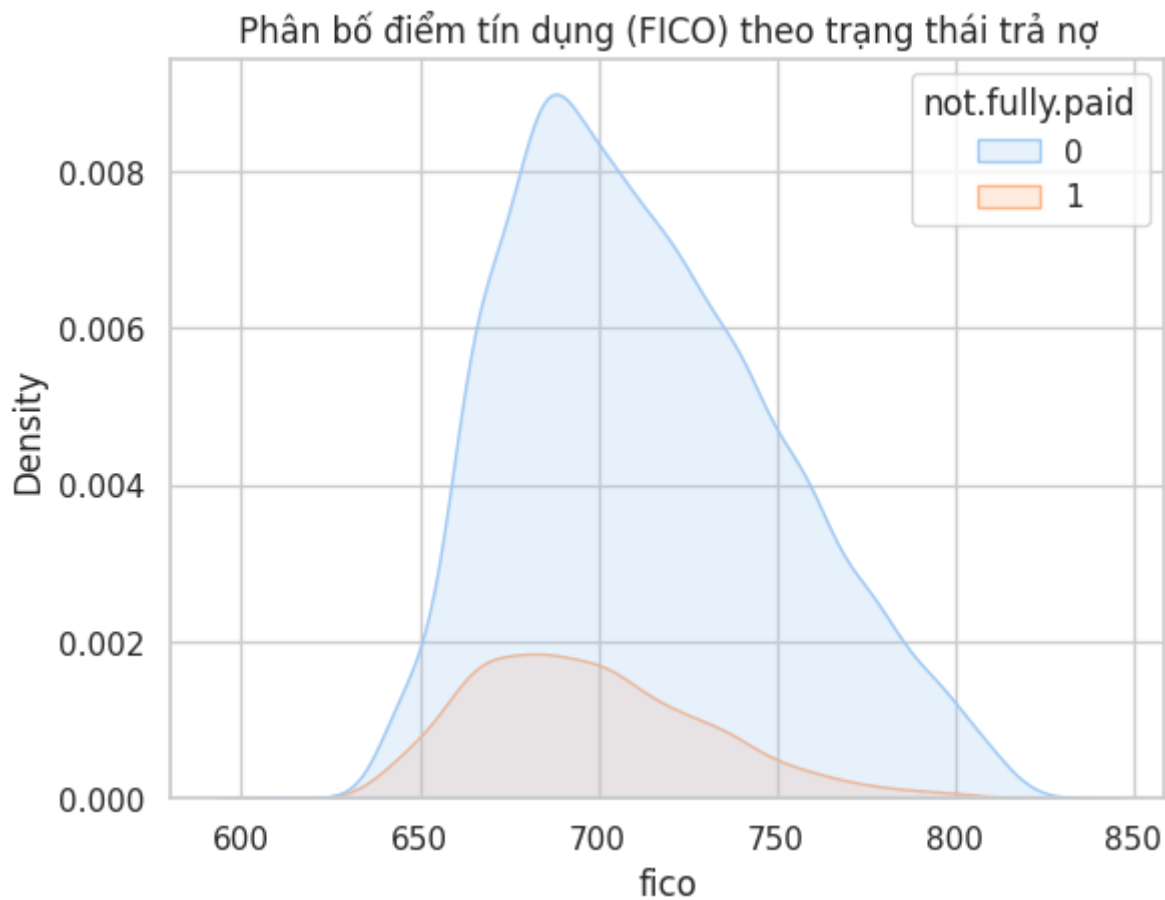
Phân tích các biến định lượng chính cho thấy cấu trúc dữ liệu phản ánh khá rõ đặc điểm tín dụng của khách hàng vay. Biến fico (điểm tín dụng) tập trung chủ yếu trong khoảng từ 680 đến 740, thể hiện phần lớn khách hàng có mức độ tín nhiệm trung bình đến khá. Tuy nhiên, vẫn tồn tại một nhóm nhỏ khách hàng có điểm tín dụng dưới 660, phản ánh khả năng rủi ro cao hơn trong việc hoàn trả các khoản vay. Biến int.rate (lãi suất vay) có phân bố lệch phải, chủ yếu tập trung trong khoảng 10% đến 13%, trong khi một số ít khoản vay chịu lãi suất trên 18%. Kết quả này cho thấy các tổ chức cho vay có xu hướng điều chỉnh mức lãi suất tương ứng với mức độ tín nhiệm của người vay, phù hợp với cơ chế định giá rủi ro tín dụng.



Hình 11 Phân bố các biến định lượng chính trong tập dữ liệu.

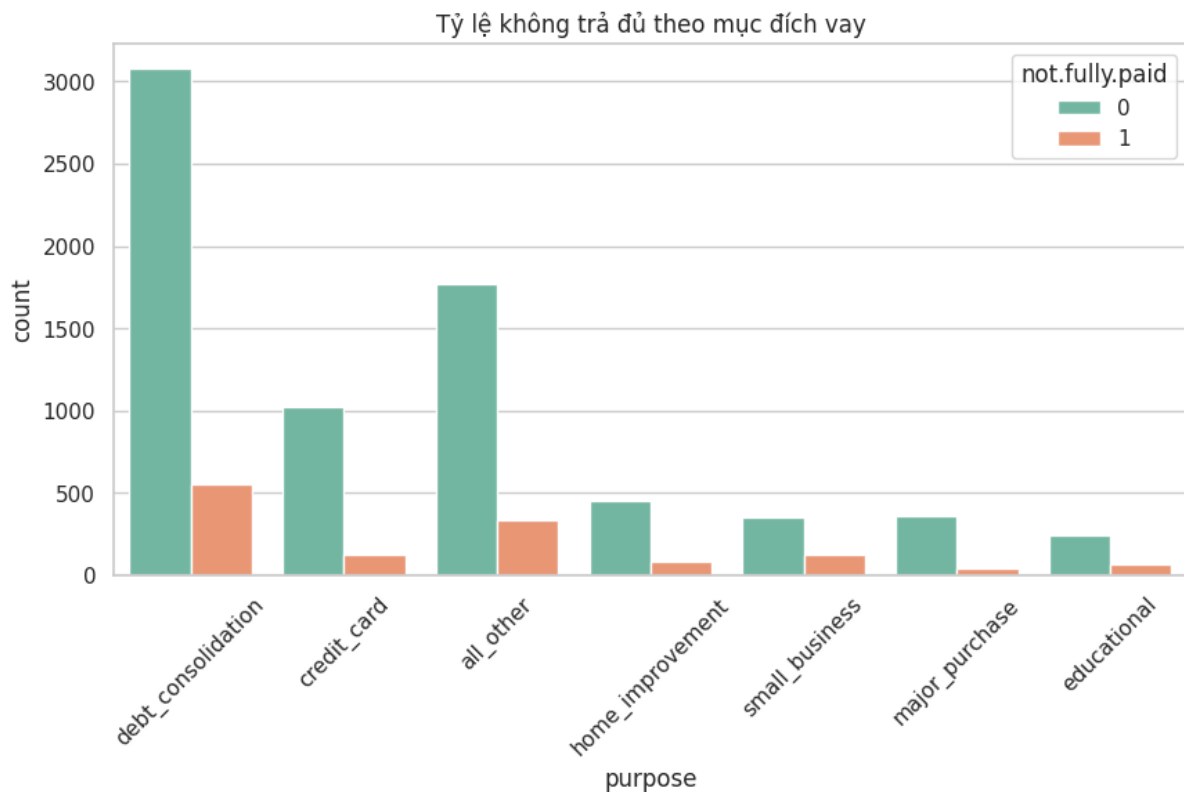
Đặc biệt, khi xem xét mối quan hệ giữa fico và biến mục tiêu not.fully.paid, có thể thấy sự khác biệt rõ rệt trong hành vi trả nợ của hai nhóm khách hàng. Nhóm trả nợ đầy đủ (giá trị 0) có phân bố điểm fico cao hơn, tập trung quanh 700–740, trong khi nhóm không trả đủ nợ (giá trị 1) chủ yếu tập trung ở mức dưới 700. Điều này gợi ý rằng điểm tín dụng thấp có liên hệ chặt chẽ với khả năng không hoàn thành nghĩa vụ trả nợ, qua đó củng cố giả thuyết rằng chất lượng tín dụng là một yếu tố quan trọng ảnh hưởng đến rủi ro tín dụng cá nhân.

### 3.2 Đặc trưng phân loại của biến định tính



Hình 12 Phân bố điểm tín dụng (FICO) theo trạng thái trả nợ.

Ở nhóm biến định tính, biến purpose (mục đích vay vốn) thể hiện sự chênh lệch đáng kể giữa các nhóm. Nhóm vay với mục đích hợp nhất nợ (debt\_consolidation) chiếm tỷ trọng lớn nhất, tiếp theo là credit\_card và all\_other. Khi đối chiếu với biến mục tiêu not.fully.paid, tỷ lệ khách hàng không trả đủ nợ cao hơn ở các nhóm vay có tính chất tiêu dùng và xoay vòng, như credit\_card hoặc debt\_consolidation. Điều này cho thấy mục đích vay vốn có thể là yếu tố phản ánh hành vi tài chính và mức độ rủi ro của khách hàng.



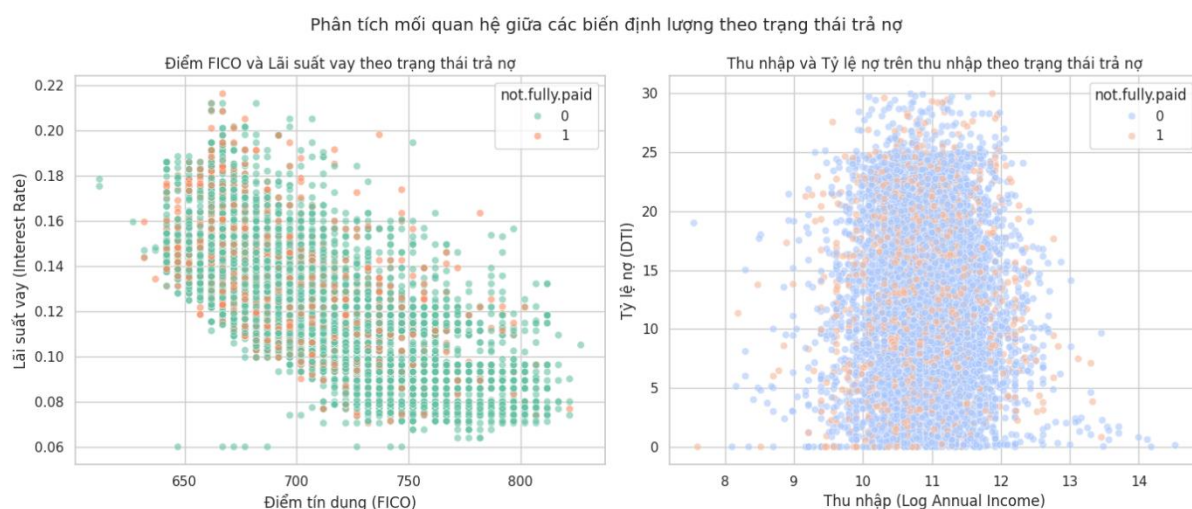
Hình 13 Tỷ lệ không trả đủ nợ theo từng nhóm mục đích vay.

Nhìn chung, kết quả phân tích mô tả cho thấy sự khác biệt đáng kể giữa các nhóm khách hàng về điểm tín dụng, lãi suất và mục đích vay vốn. Những phát hiện này đóng vai trò như bằng chứng ban đầu cho giả thuyết nghiên cứu, rằng khả năng không trả đủ nợ của khách hàng chịu ảnh hưởng đồng thời bởi đặc điểm tín dụng và hành vi vay vốn. Phần tiếp theo sẽ tập trung phân tích sâu hơn mối quan hệ giữa các biến này nhằm làm rõ cơ chế ảnh hưởng đến rủi ro tín dụng.

#### 4. Phân tích mối quan hệ giữa các biến

##### 4.1. Mối quan hệ tuyến tính giữa các biến định lượng

Phân tích mối quan hệ giữa các biến định lượng giúp xác định những đặc điểm tài chính có khả năng ảnh hưởng trực tiếp đến rủi ro tín dụng của khách hàng. Hình dưới đây thể hiện đồng thời hai cặp biến quan trọng: (i) điểm tín dụng (fico) và lãi suất vay (int.rate), (ii) thu nhập (log.annual.inc) và tỷ lệ nợ trên thu nhập (dti), được phân tích theo trạng thái trả nợ (not.fully.paid).

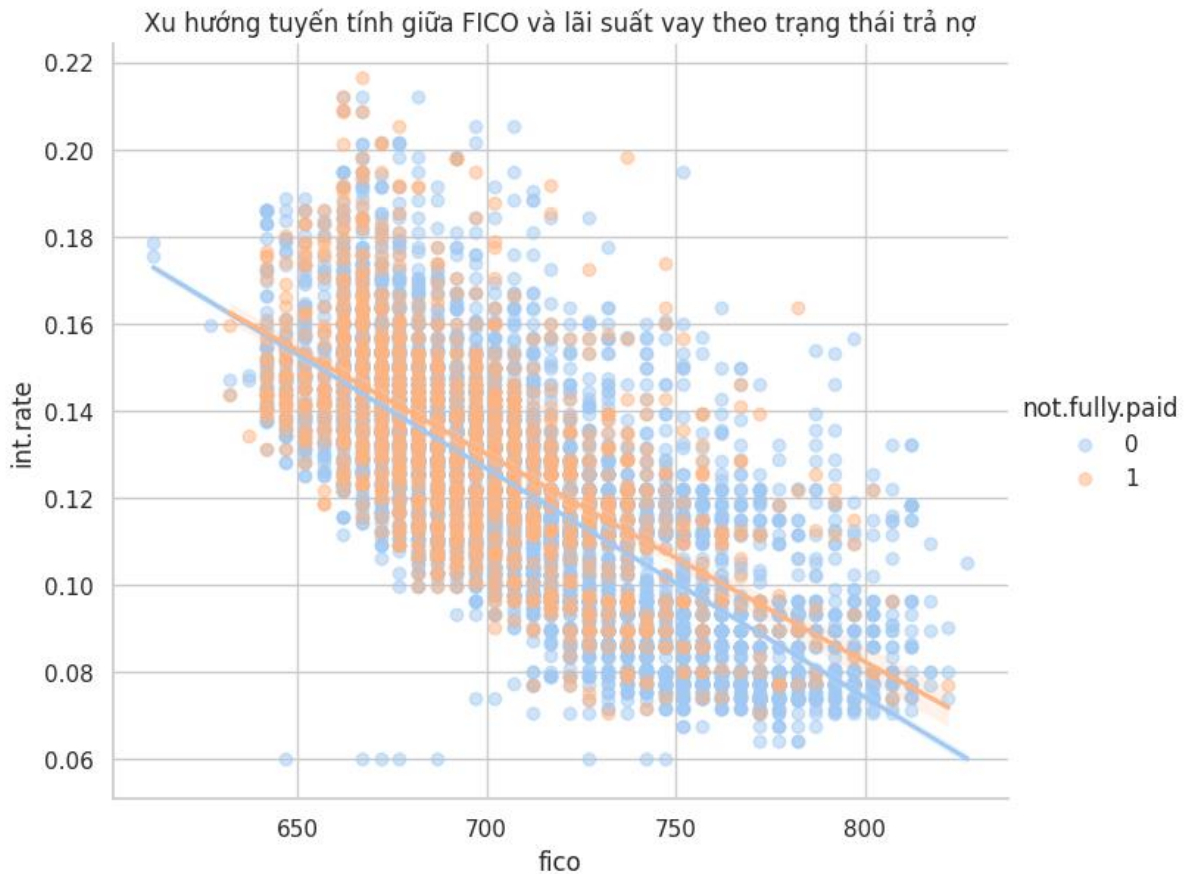


*Hình 14 Mối quan hệ giữa các biến định lượng theo trạng thái trả nợ.*

Quan sát biểu đồ cho thấy mối quan hệ nghịch chiều rõ rệt giữa điểm tín dụng và lãi suất vay. Các cá nhân có điểm fico thấp thường phải chịu lãi suất cao hơn, và chính họ cũng chiếm phần lớn trong nhóm không trả đủ nợ. Xu hướng này phản ánh cơ chế định giá rủi ro trong hoạt động cho vay: lãi suất đóng vai trò bù đắp rủi ro, do đó những người có hồ sơ tín dụng yếu (rủi ro cao) sẽ phải trả chi phí vốn lớn hơn. Tuy nhiên, mức lãi suất cao cũng có thể làm tăng áp lực trả nợ, khiến khả năng vỡ nợ trở nên rõ rệt hơn – gợi ý về một vòng xoáy rủi ro tín dụng tiềm ẩn.

Đối với cặp biến `log.annual.inc` và `dti`, dữ liệu cho thấy tương quan tuyến tính yếu, song vẫn thể hiện một xu hướng đáng chú ý về khả năng chịu nợ của khách hàng. Nhóm có tỷ lệ `dti` cao – tức gánh nặng nợ lớn so với thu nhập – xuất hiện nhiều hơn trong nhóm không trả đủ nợ, dù mức thu nhập tuyệt đối của họ không hẳn thấp. Điều này hàm ý rằng khả năng hoàn trả không chỉ phụ thuộc vào thu nhập, mà còn vào cấu trúc chi tiêu và nghĩa vụ nợ hiện hữu – yếu tố mà các mô hình chấm điểm tín dụng cần cân nhắc song song với điểm fico.

Để kiểm chứng xu hướng này, biểu đồ hồi quy tuyến tính giữa `fico` và `int.rate` được trình bày dưới đây.



Hình 15 Xu hướng tuyến tính giữa điểm tín dụng và lãi suất vay theo trạng thái trả nợ.

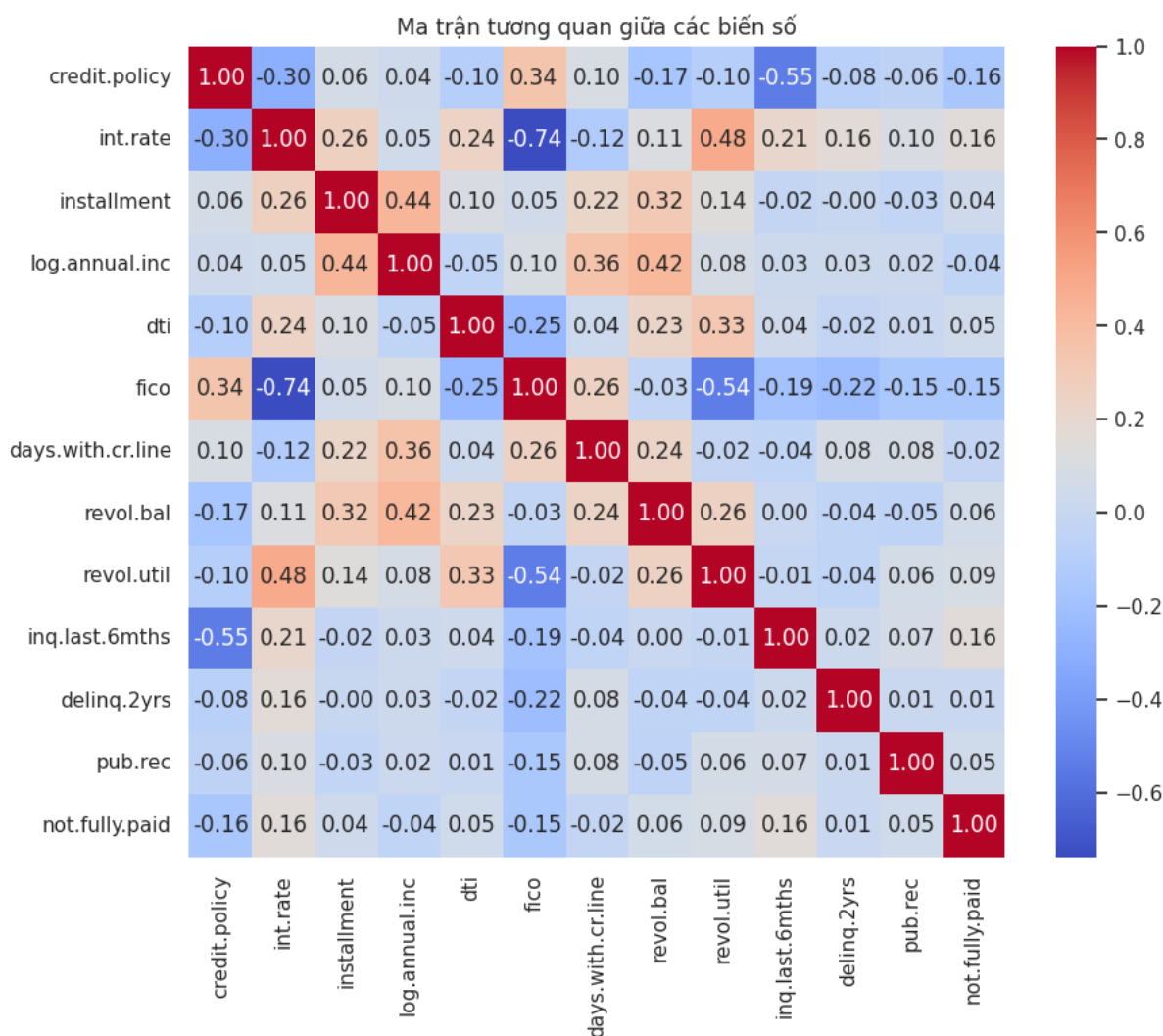
Đường hồi quy thể hiện xu hướng tuyến tính âm ổn định, cho thấy lãi suất giảm dần khi điểm tín dụng tăng. Nhóm khách hàng không trả đủ nợ (màu cam) tập trung chủ yếu ở vùng điểm tín dụng thấp và lãi suất cao – khẳng định rằng rủi ro tín dụng có thể được mô tả bằng mối quan hệ tuyến tính giữa chất lượng tín dụng và chi phí vay vốn.

Kết quả này không chỉ củng cố ý nghĩa kinh tế của biến fico và int.rate, mà còn chỉ ra rằng những mối quan hệ tuyến tính trong dữ liệu phản ánh logic vận hành thực tế của thị trường tín dụng tiêu dùng. Do đó, các biến này được xem là những yếu tố đầu vào có ý nghĩa trong việc dự báo xác suất không trả đủ nợ ở giai đoạn mô hình hóa tiếp theo.

#### 4.2. Ma trận tương quan giữa các biến định lượng

Phân tích tương quan giúp xác định mối quan hệ tuyến tính giữa các biến định lượng và kiểm tra khả năng xảy ra đa cộng tuyến trong dữ liệu.





Hình 16 Ma trận tương quan giữa các biến định lượng trong tập dữ liệu

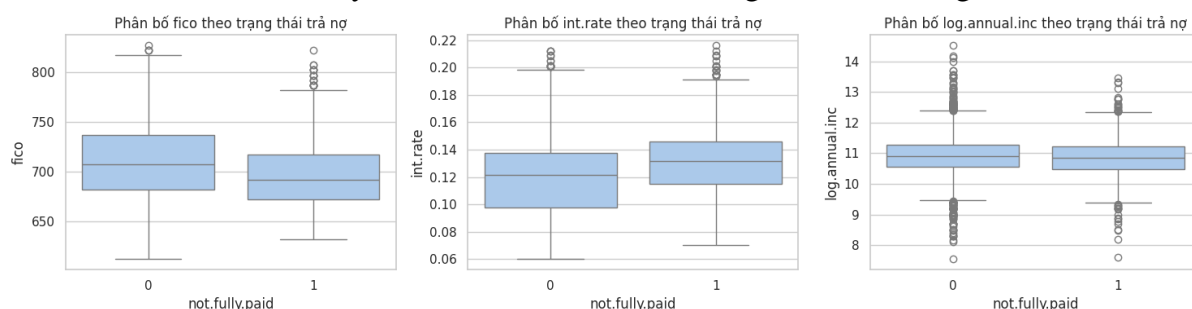
Kết quả cho thấy điểm tín dụng (fico) và lãi suất (int.rate) có tương quan âm mạnh ở mức -0,74, nghĩa là khách hàng có điểm tín dụng càng thấp thì lãi suất vay càng cao, phù hợp với nguyên tắc định giá rủi ro của các tổ chức tín dụng. Biến mục tiêu not.fully.paid có tương quan dương nhẹ với int.rate và tương quan âm nhẹ với fico, cho thấy những khách hàng phải chịu mức lãi suất cao và có điểm tín dụng thấp thường có xu hướng không trả đủ nợ. Ngoài ra, không có cặp biến nào có hệ số tương quan tuyệt đối vượt 0,8, do đó hiện tượng đa cộng tuyến trong dữ liệu là không đáng kể.

Kết quả này góp phần khẳng định giả định ban đầu của nghiên cứu rằng các yếu tố về tín dụng, đặc biệt là lãi suất và điểm tín dụng, có thể ảnh hưởng mạnh đến khả năng trả nợ của khách hàng.



## 5. Phân tích sâu các yếu tố ảnh hưởng đến khả năng trả nợ

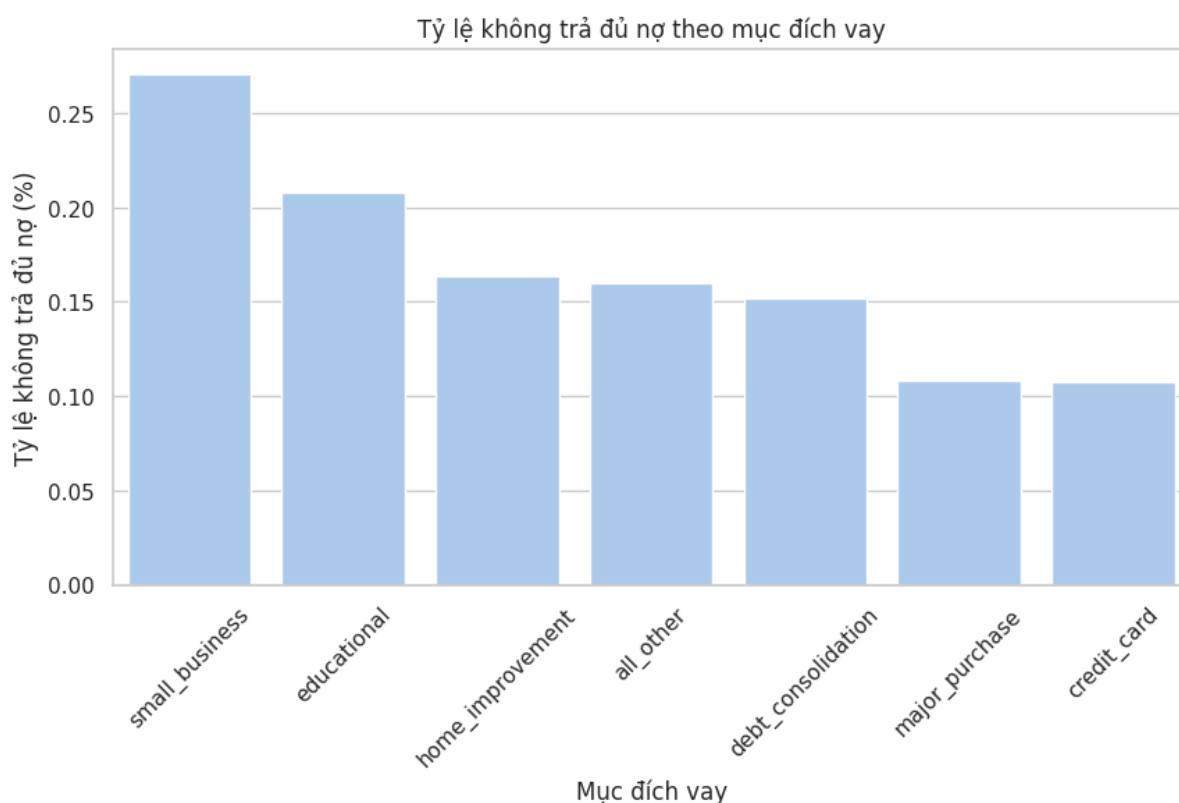
Để xem xét rõ hơn sự khác biệt giữa hai nhóm khách hàng, biểu đồ hộp được sử dụng để mô tả sự thay đổi của các biến định lượng theo tình trạng trả nợ.



Hình 17 Kết quả sự thay đổi của các biến định lượng theo tình trạng trả nợ

Kết quả cho thấy nhóm không trả đủ nợ thường có điểm FICO thấp hơn đáng kể và lãi suất cao hơn so với nhóm trả nợ đầy đủ. Điều này phản ánh đúng logic của thị trường tín dụng: những khách hàng bị đánh giá rủi ro cao sẽ phải chịu chi phí vay vốn cao hơn, đồng thời cũng là nhóm dễ gặp khó khăn trong quá trình trả nợ. Ngược lại, biến thu nhập (log.annual.inc) không có sự khác biệt rõ ràng giữa hai nhóm, cho thấy yếu tố thu nhập không phải lúc nào cũng là chỉ báo tốt cho khả năng trả nợ nếu không đi kèm với mức độ tín nhiệm.

Xét theo mục đích vay (purpose), tỷ lệ không trả đủ nợ cao nhất tập trung ở các khoản vay phục vụ kinh doanh nhỏ (*small business*) và giáo dục (*e3.3educational*), trong khi các khoản vay tiêu dùng như hợp nhất nợ hoặc thẻ tín dụng lại có rủi ro thấp hơn.



Hình 18 Tỷ lệ không trả đủ nợ theo mục đích vay

Kết quả này gợi ý rằng mục đích vay vốn có tác động đáng kể đến rủi ro tín dụng: các khoản vay mang tính đầu tư, sinh lợi tiềm năng cao thường đi kèm rủi ro tài chính lớn hơn.

## **6. Tổng kết chương**

Phân tích mô tả dữ liệu đã giúp hình thành bức tranh tổng thể về tập dữ liệu và làm rõ mối quan hệ giữa các yếu tố tài chính với khả năng trả nợ của khách hàng. Kết quả cho thấy tập dữ liệu có chất lượng tốt, không gặp vấn đề về đa cộng tuyến, và các biến như `fico`, `int.rate`, `dti`, và `purpose` có mối liên hệ chặt chẽ với biến mục tiêu `not.fully.paid`. Những phát hiện này là cơ sở vững chắc cho bước tiếp theo trong nghiên cứu – xây dựng mô hình phân loại nhằm dự đoán xác suất khách hàng không trả đủ nợ. Phần tiếp theo sẽ trình bày chi tiết quá trình mô hình hóa, bao gồm lựa chọn thuật toán, huấn luyện và đánh giá hiệu quả dự báo.

## **Chương IV: Tiến hành phân lớp dữ liệu**

### **1. Kiểm tra tình trạng mất cân bằng dữ liệu**

Từ kết quả khi phân tích mô tả dữ liệu ở trên đã cho ra kết quả cho thấy dữ liệu của biến mục tiêu bị mất cân bằng nghiêm trọng. Điều này đặt ra yêu cầu cần có biện pháp để xử lý nếu không muốn thuật toán xảy ra thiên lệch và bỏ qua các khách hàng có mức độ rủi ro cao.

### **2. Chọn Phương pháp xử lý mất cân bằng**

#### **2.1. Chọn mô hình Logistic Regression cho giai đoạn đầu**

Nhóm lựa chọn mô hình Logistic Regression làm bước thử nghiệm đầu tiên vì đây là mô hình nền tảng, dễ diễn giải, hoạt động tốt với dữ liệu tuyến tính và cho phép kiểm soát trọng số lớp (class weight) một cách trực tiếp. Logistic Regression thường được sử dụng như một mô hình baseline để đánh giá tác động của các kỹ thuật xử lý mất cân bằng dữ liệu trước khi áp dụng cho các thuật toán phức tạp hơn như XGBoost hay Random Forest.

Việc áp dụng Logistic Regression trước còn giúp nhóm đánh giá hiệu quả thực tế của từng kỹ thuật cân bằng dữ liệu (oversampling, undersampling, class weight) trên cùng một khung mô hình đơn giản. Kết quả từ mô hình này là cơ sở để lựa chọn phương pháp xử lý tối ưu trước khi triển khai cho các mô hình có cấu trúc cây.

## 2.2. Chạy với dữ liệu ban đầu

In [ ]:

```
print_scores( y_test, y_pred)
```

	precision	recall	f1-score	support
0	0.85	1.00	0.92	1456
1	0.59	0.04	0.07	268
accuracy			0.85	1724
macro avg	0.72	0.52	0.49	1724
weighted avg	0.81	0.85	0.78	1724

Hình 19 Kết quả chạy với dữ liệu ban đầu

## 2.3. Under sampling(RUS)

	precision	recall	f1-score	support
0	0.89	0.64	0.74	1456
1	0.22	0.56	0.32	268
accuracy			0.63	1724
macro avg	0.55	0.60	0.53	1724
weighted avg	0.78	0.63	0.68	1724

Hình 20 Kết quả chạy với dữ liệu xử lý bởi RUS

## 2.4. Neermis(INNS)

	precision	recall	f1-score	support
0	0.91	0.40	0.56	1456
1	0.19	0.77	0.31	268
accuracy			0.46	1724
macro avg	0.55	0.59	0.43	1724
weighted avg	0.79	0.46	0.52	1724

Hình 21 Kết quả chạy với dữ liệu xử lý bởi INS

## 2.5. Random Oversampling(ROS)

	precision	recall	f1-score	support
0	0.89	0.64	0.75	1456
1	0.23	0.58	0.33	268
accuracy			0.63	1724
macro avg	0.56	0.61	0.54	1724
weighted avg	0.79	0.63	0.68	1724

Hình 22 Kết quả chạy với dữ liệu xử lý bởi ROS

## 2.6. SMOTE

	precision	recall	f1-score	support
0	0.88	0.66	0.75	1456
1	0.21	0.49	0.29	268
accuracy			0.63	1724
macro avg	0.54	0.57	0.52	1724
weighted avg	0.77	0.63	0.68	1724

Hình 23 Kết quả chạy với dữ liệu xử lý bởi SMOTE

## 2.7. Class weight

	precision	recall	f1-score	support
0	0.89	0.65	0.75	1456
1	0.23	0.57	0.33	268
accuracy			0.63	1724
macro avg	0.56	0.61	0.54	1724
weighted avg	0.79	0.63	0.68	1724

Hình 24 Kết quả chạy với dữ liệu xử lý bởi Class weight

### Bảng tổng hợp

Phương pháp	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	Weighted Avg F1-Score
<b>Class weighted</b>	0.23	0.57	0.33	0.68
<b>RUS (Undersampling)</b>	0.22	0.56	0.32	0.68
<b>ROS (Oversampling)</b>	0.23	0.58	0.33	0.68
<b>SMOTE</b>	<b>0.21</b>	<b>0.49</b>	<b>0.29</b>	0.68
<b>Neermis(Undersampli ng)</b>	0.19	<b>0.77</b>	<b>0.31</b>	0.52

Bảng 2 Kết quả tổng hợp kết quả chạy với dữ liệu được xử lý bởi các phương pháp

Kết luận:

Nhóm lựa chọn phương pháp SMOTE (Synthetic Minority Oversampling Technique) để xử lý vấn đề mất cân bằng dữ liệu do kỹ thuật này tạo ra các mẫu thiểu số mới bằng

cách nội suy giữa các điểm gần nhau, thay vì chỉ sao chép ngẫu nhiên như ROS. Cách tiếp cận này giúp mở rộng không gian đặc trưng của lớp thiểu số một cách hợp lý hơn, đồng thời giảm nguy cơ overfitting khi mô hình học lặp lại dữ liệu trùng lặp. So với các phương pháp undersampling như RUS hay NeerMiss, SMOTE có ưu thế là giữ nguyên toàn bộ dữ liệu của lớp đa số, tránh làm mất thông tin quan trọng trong quá trình huấn luyện. Kết quả cho thấy mô hình sử dụng SMOTE đạt Weighted Avg F1-Score ở mức 0.68, tương đương với các phương pháp khác, cho thấy hiệu quả phân loại tổng thể vẫn được duy trì ổn định. Dù F1-Score của lớp thiểu số chưa cao nhất, song SMOTE giúp cân bằng tương đối giữa precision và recall, phản ánh khả năng phát hiện các trường hợp thiểu số mà không đánh đổi quá nhiều độ chính xác. Bên cạnh đó, SMOTE là một phương pháp phổ biến, dễ áp dụng và được công nhận rộng rãi trong xử lý mất cân bằng dữ liệu, do đó việc lựa chọn phương pháp này giúp quy trình nghiên cứu đảm bảo tính minh bạch, khả năng tái lập và giá trị so sánh học thuật.

### 3. Phân lớp

#### 3.1. XGBoost

Sau khi dữ liệu đã được cân bằng bằng SMOTE, nhóm tiến hành xây dựng mô hình XGBoost – một thuật toán mạnh trong việc xử lý dữ liệu phi tuyến tính và thường đạt hiệu suất cao trong bài toán phân lớp.

Các bước thực hiện gồm:

1. Import thư viện và đọc dữ liệu.

Trong bước đầu tiên, các thư viện cần thiết được nạp vào, bao gồm: *NumPy*, *Pandas*, *Matplotlib* và *Seaborn*. Các module từ *scikit-learn* phục vụ cho việc chia dữ liệu, chuẩn hóa, huấn luyện mô hình SVM, và đánh giá kết quả. Môi trường hiển thị đồ họa được cấu hình để các biểu đồ xuất hiện trực tiếp trong notebook, đồng thời tinh chỉnh một số thông số giúp hình ảnh hiển thị rõ nét và trực quan hơn.

2. Chia tập dữ liệu thành tập huấn luyện (train) và tập kiểm tra (test) theo tỷ lệ thích hợp và áp dụng SMOTE cho tập huấn luyện để cân bằng dữ liệu.

Trong đó, thực hiện xác định biến mục tiêu (label): Hệ thống tự động nhận diện biến *not.fully.paid* làm biến đầu ra. Sau đó thực hiện bước Tách tập dữ liệu: Biến mục tiêu được tách riêng khỏi các biến đầu vào.

3. Xây dựng mô hình XGBoost và dùng GridSearchCV để tìm bộ siêu tham số tối ưu (*learning\_rate*, *max\_depth*, *n\_estimators*,...).

```
Best params: {'colsample_bytree': 1.0, 'learning_rate': 0.05, 'max_depth': 4, 'n_estimators': 300, 'subsample': 0.8}
```

Hình 25 Kết quả bộ siêu tham số tối ưu

Trong bước này, nhóm tiến hành huấn luyện mô hình XGBoost (Extreme Gradient Boosting) để phân loại rủi ro khách hàng không trả được nợ. Mô hình XGBoost được lựa chọn nhờ khả năng xử lý tốt các dữ liệu mất cân bằng, tính linh hoạt trong điều chỉnh siêu tham số và hiệu quả cao trong nhiều bài toán dự báo rủi ro tín dụng. Trong phần khởi tạo, tham số `eval_metric='logloss'` được sử dụng nhằm đo lường mức độ sai lệch giữa giá trị dự báo và thực tế trong quá trình huấn luyện; chỉ số này càng nhỏ thì mô hình càng chính xác. Tham số `random_state=42` giúp đảm bảo kết quả có thể tái lập, trong khi `n_jobs=-1` cho phép tận dụng toàn bộ lõi CPU để tăng tốc độ huấn luyện.

Sau đó, nhóm thiết lập một tập hợp các siêu tham số (`param_grid`) để thực hiện quá trình tìm kiếm tối ưu bằng phương pháp Grid Search. Cụ thể, `n_estimators` được thử nghiệm với các giá trị 200 và 300 nhằm xác định số lượng cây quyết định phù hợp — giá trị này càng cao có thể giúp mô hình học tốt hơn nhưng cũng dễ dẫn đến quá khớp. Tham số `max_depth` được chọn trong khoảng từ 3 đến 4 nhằm kiểm soát độ sâu của cây, qua đó giới hạn độ phức tạp của mô hình. Tốc độ học (`learning_rate`) được thử với hai mức 0.03 và 0.05; đây là những giá trị nhỏ giúp mô hình học dần dần, tránh việc cập nhật trọng số quá nhanh gây dao động. Hai tham số `subsample` và `colsample_bytree` lần lượt được đặt trong khoảng từ 0.8 đến 1.0, nhằm xác định tỷ lệ mẫu và tỷ lệ đặc trưng được sử dụng trong mỗi lần huấn luyện cây, giúp mô hình giảm thiểu hiện tượng overfitting và tăng khả năng tổng quát hóa.

Quá trình Grid Search được thực hiện bằng `GridSearchCV`, trong đó tiêu chí đánh giá là chỉ số F1-score — một thước đo cân bằng giữa Precision và Recall, đặc biệt phù hợp khi dữ liệu có sự mất cân bằng giữa hai lớp (ví dụ: khách hàng trả nợ và không trả nợ). Việc chia dữ liệu thành ba phần trong tham số `cv=3` giúp đảm bảo mô hình được kiểm định chéo, giảm sai lệch trong đánh giá hiệu suất.

Sau khi chạy Grid Search, mô hình sẽ tự động thử nghiệm tất cả các tổ hợp siêu tham số có thể có, huấn luyện trên tập dữ liệu `X_train_res` và `y_train_res`, rồi chọn ra bộ tham số tối ưu dựa trên F1-score cao nhất. Kết quả cuối cùng hiển thị thông qua dòng lệnh `print("Best params:", grid.best_params_)`, cho biết tổ hợp siêu tham số giúp mô hình đạt hiệu suất dự báo tốt nhất. Đây là bước quan trọng để đảm bảo mô hình XGBoost không chỉ đạt độ chính xác cao mà còn duy trì được sự cân bằng giữa khả năng phát hiện khách hàng rủi ro và hạn chế dự báo sai đối với các khách hàng an toàn.

#### 4. Đánh giá và trực quan hóa kết quả mô hình XGBoost

Mô hình sau khi huấn luyện được đánh giá qua các chỉ tiêu Accuracy, Precision, Recall, F1-score, và AUC-ROC. Ngoài ra, nhóm còn trực quan hóa ma trận nhầm lẫn (Confusion Matrix) để xem mức độ phân biệt giữa hai lớp.

Tính toán và in ra màn hình Accuracy và ROC AUC của mô hình, lấy 3 số sau dấu phẩy. Sau đó, in báo cáo chi tiết và ma trận nhầm lẫn.

Vẽ đường cong ROC curve để thể hiện trade-off giữa recall và false positive rate khi thay đổi threshold. Dùng để chọn threshold hợp lý nếu bạn muốn cân bằng TPR và FPR theo chi phí thực tế; so sánh mô hình (model có AUC lớn hơn thường tốt hơn về phân biệt).

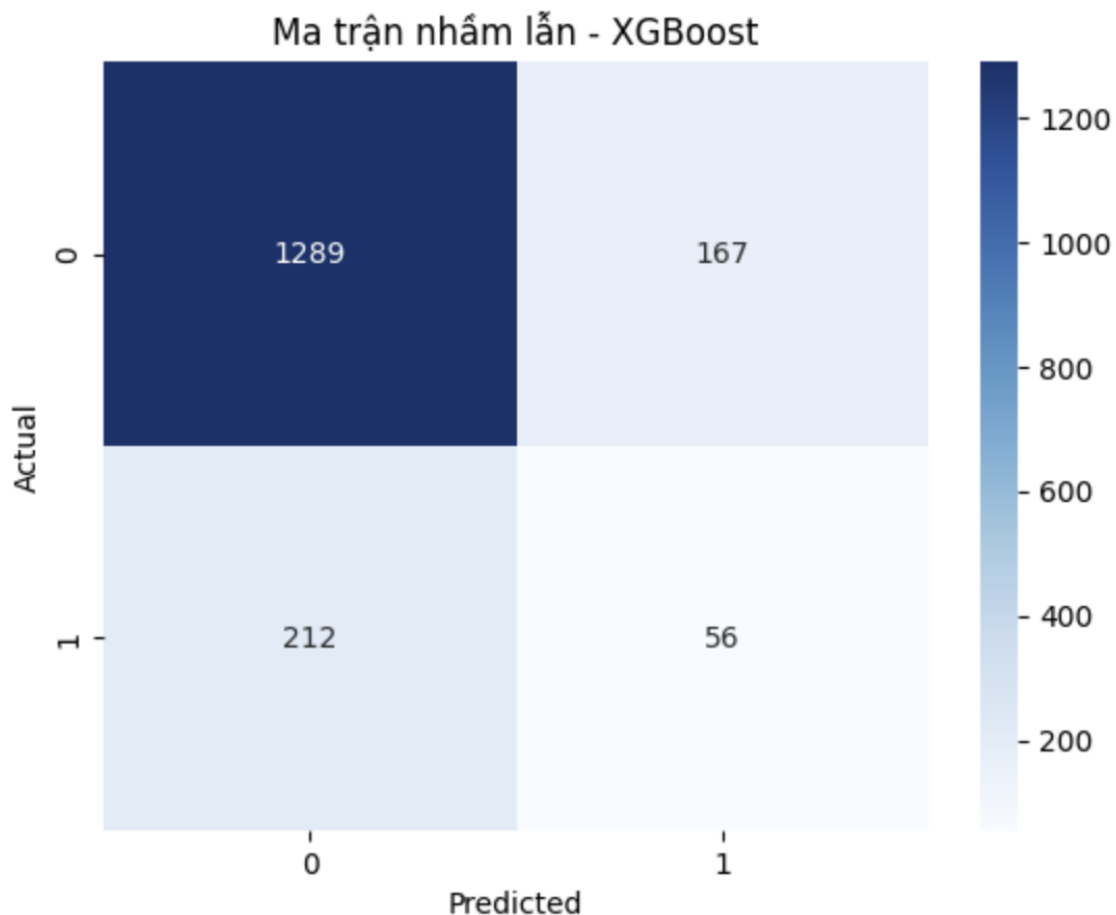
Cho biết tầm quan trọng tương đối của biến theo đóng góp vào split/gain (mặc định XGBoost có nhiều loại importance: gain, weight, cover). Mục đích là hiểu biến nào ảnh hưởng nhiều nhất tới dự đoán (hữu ích cho giải thích mô hình, giảm chiều - feature selection, hoặc kiểm tra tính hợp lý theo domain knowledge).

```
Accuracy: 0.780 | ROC AUC: 0.620
```

```
=== Báo cáo phân lớp ===
```

	precision	recall	f1-score	support
0	0.859	0.885	0.872	1456
1	0.251	0.209	0.228	268
accuracy			0.780	1724
macro avg	0.555	0.547	0.550	1724
weighted avg	0.764	0.780	0.772	1724

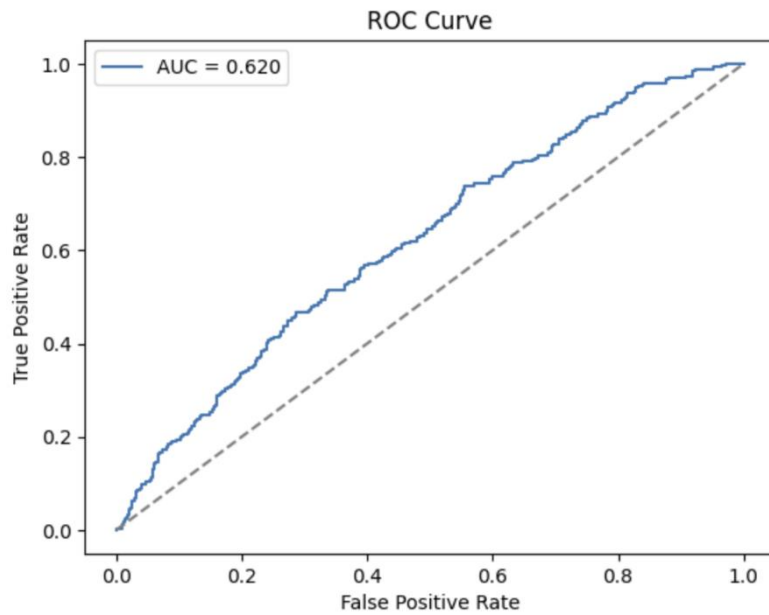
*Hình 26 Kết quả chạy cho mô hình XG Boost*



*Hình 27 Ma trận nhầm lẫn chạy với dữ liệu xử lý bởi XG Boost*

Mô hình XGBoost sau khi tối ưu đạt độ chính xác (Accuracy) 0.780 và ROC AUC 0.620, cho thấy khả năng phân biệt giữa hai nhóm khách hàng chỉ ở mức trung bình. Nhìn vào bảng phân loại, mô hình dự đoán tốt với lớp “0” (khách hàng trả nợ đầy đủ) với precision 0.859 và recall 0.885, trong khi hiệu quả dự báo lớp “1” (không trả nợ đầy đủ) còn thấp, với precision chỉ 0.251 và recall 0.209. Điều này cho thấy mô hình vẫn chưa nhận diện tốt nhóm khách hàng rủi ro cao, nguyên nhân chủ yếu là do dữ liệu bị mất cân bằng (tỷ lệ khách hàng không trả nợ rất thấp). Tuy vậy, các chỉ số macro và weighted trung bình đều quanh mức 0.55–0.77, chứng tỏ mô hình có tiềm năng cải thiện thêm nếu cân bằng lại dữ liệu hoặc tinh chỉnh thêm ngưỡng phân loại.



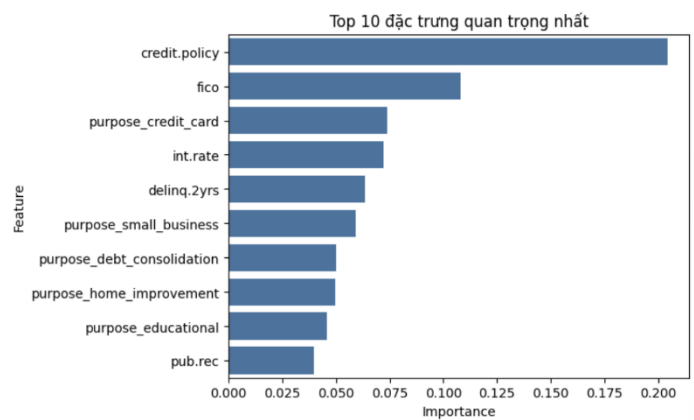


Hình 28 Kết quả vẽ đường ROC Curve chạy với mô hình XG Boost

Top 10 đặc trưng quan trọng nhất:

	Feature	Importance
8	credit.policy	0.204250
4	fico	0.108208
12	purpose_credit_card	0.073912
0	int.rate	0.072022
10	delinq.2yrs	0.063429
17	purpose_small_business	0.059194
13	purpose_debt_consolidation	0.050227
15	purpose_home_improvement	0.049789
14	purpose_educational	0.045864
11	pub.rec	0.039799

Hình 29 Kết quả các biến quan trọng chạy với mô hình XG Boost theo trọng số



Hình 30 Kết quả các biến quan trọng chạy với mô hình XG Boost

Kết quả cho thấy biến `credit.policy` có mức độ quan trọng cao nhất (0.204), vượt trội so với các biến còn lại, cho thấy chính sách tín dụng của tổ chức cho vay là yếu tố then chốt ảnh hưởng đến khả năng khách hàng không trả được nợ. Tiếp theo là biến `fico` (0.108), phản ánh điểm tín dụng của khách hàng – yếu tố đánh giá trực tiếp mức độ uy tín trong vay mượn. Các biến như `purpose_credit_card`, `int.rate` và `delinq.2yrs` cũng có ảnh hưởng đáng kể, lần lượt thể hiện mục đích vay, lãi suất và số lần chậm trả trong hai năm gần nhất. Nhìn chung, các đặc trưng liên quan đến chính sách tín dụng, lịch sử tín dụng và điều kiện vay mượn đóng vai trò quyết định trong dự báo rủi ro tín dụng, phù hợp với thực tiễn trong lĩnh vực tài chính – ngân hàng.

### 3.2. Random Forest

Sau khi có kết quả từ XGBoost, nhóm tiếp tục triển khai mô hình Random Forest để so sánh hiệu suất. Random Forest cũng là mô hình dựa trên cây quyết định, hoạt động tốt với dữ liệu phi tuyến tính và có khả năng chống overfitting nhờ cơ chế lấy mẫu ngẫu nhiên.

Nhóm áp dụng cùng quy trình xử lý dữ liệu (SMOTE) và tối ưu tham số tương tự như XGBoost để đảm bảo tính đồng nhất khi so sánh. Sau khi huấn luyện, nhóm tiếp tục đánh giá mô hình qua các chỉ tiêu tương tự (Accuracy, Precision, Recall, F1-score, ROC-AUC) và trực quan hóa kết quả.

```

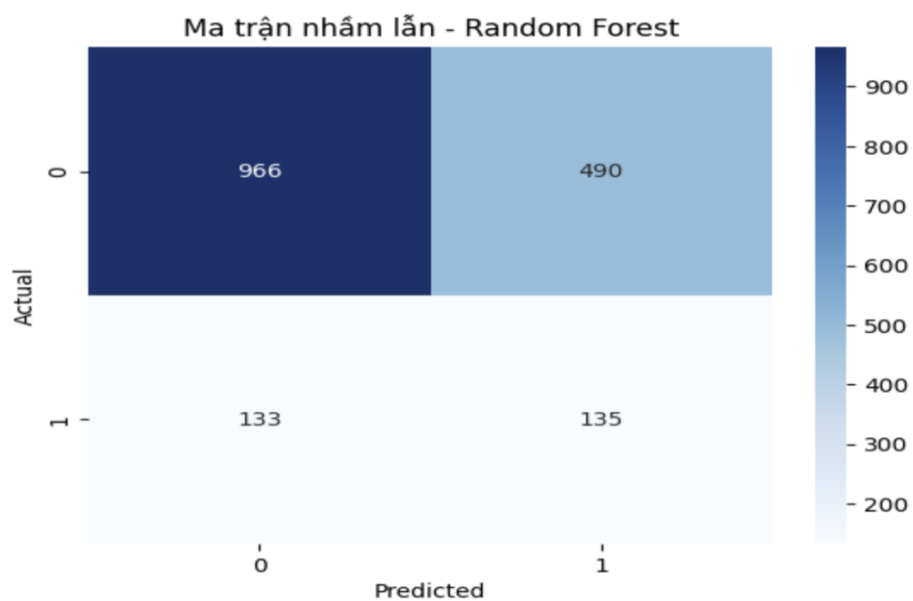
=== Báo cáo phân lớp - Random Forest ===

```

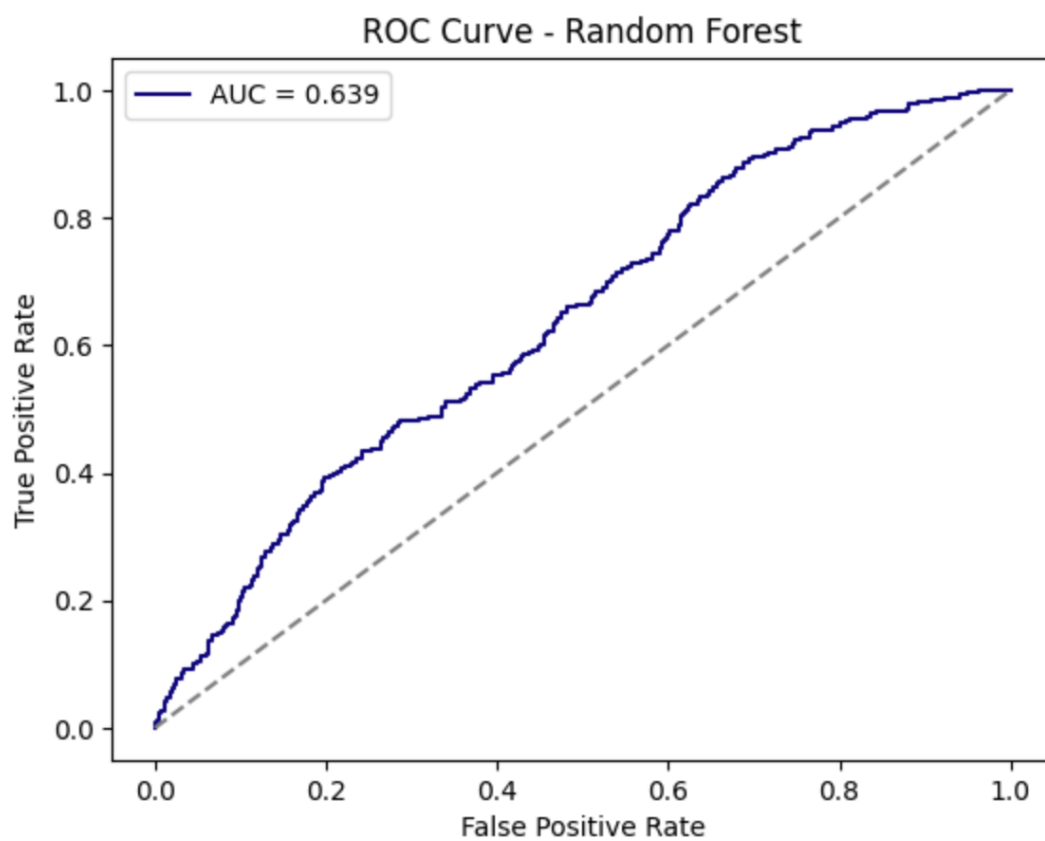
	precision	recall	f1-score	support
0	0.879	0.663	0.756	1456
1	0.216	0.504	0.302	268
accuracy			0.639	1724
macro avg	0.547	0.584	0.529	1724
weighted avg	0.776	0.639	0.686	1724

#### *Kết quả chạy cho mô hình Random Forest*

Kết quả phân lớp bằng mô hình Random Forest cho thấy độ chính xác (accuracy) đạt 0.639, tức mô hình dự đoán đúng khoảng 63.9% tổng số quan sát. Tuy nhiên, khi đi sâu vào từng lớp, có thể thấy sự mất cân bằng rõ rệt trong khả năng nhận diện. Với lớp “0” (khách hàng trả nợ đầy đủ), mô hình đạt precision 0.879 và recall 0.663, nghĩa là dự đoán khá tốt ở nhóm chiếm tỷ lệ cao trong dữ liệu. Ngược lại, với lớp “1” (khách hàng không trả đầy đủ), các chỉ số đều ở mức thấp (precision chỉ 0.216, recall 0.504, f1-score 0.302), cho thấy mô hình vẫn còn hạn chế trong việc phát hiện đúng nhóm khách hàng rủi ro. Điều này phần lớn bắt nguồn từ hiện tượng mất cân bằng dữ liệu – khi số lượng khách hàng không trả nợ quá ít so với số còn lại. Nhìn chung, các giá trị macro avg (0.529) và weighted avg (0.686) cho thấy mô hình có mức độ hiệu quả trung bình, cần cải thiện thêm thông qua việc cân bằng dữ liệu hoặc điều chỉnh tham số.



Hình 31 Ma trận nhầm lẫn chạy với dữ liệu xử lý bởi Random Forest

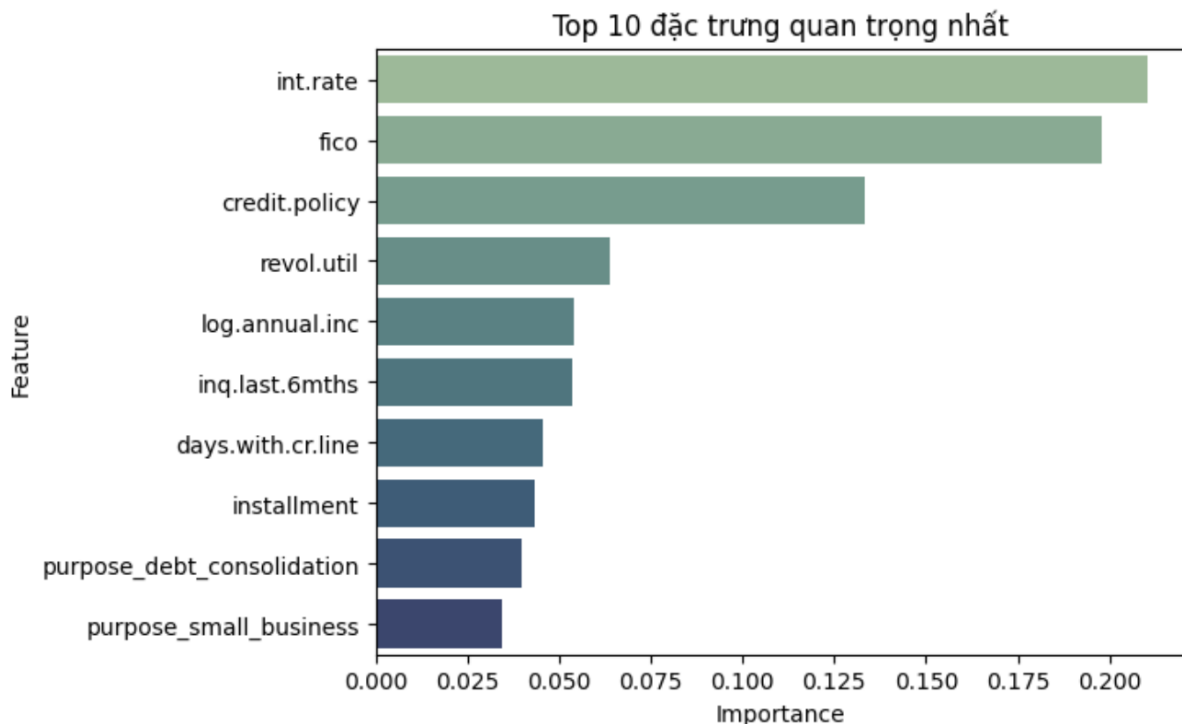


Hình 32 Kết quả vẽ đường ROC Curve chạy với mô hình Random Forest

Top 10 đặc trưng quan trọng nhất:

	Feature	Importance
0	int.rate	0.210495
4	fico	0.198091
8	credit.policy	0.133479
7	revol.util	0.063649
2	log.annual.inc	0.054219
9	inq.last.6mths	0.053594
5	days.with.cr.line	0.045351
1	installment	0.043423
13	purpose_debt_consolidation	0.039695
17	purpose_small_business	0.034350

Hình 33 Kết quả các biến quan trọng chạy với mô hình Random Forest theo trọng số



Hình 34 Kết quả các biến quan trọng chạy với mô hình Random Forest

Phân tích mức độ quan trọng của các đặc trưng trong mô hình Random Forest cho thấy int.rate (lãi suất) là yếu tố có tác động lớn nhất đến khả năng không trả nợ, với độ quan trọng 0.210. Tiếp theo là fico (điểm tín dụng) với 0.198, phản ánh trực tiếp uy tín tài chính của khách hàng. Biến credit.policy (0.133) cũng thể hiện vai trò đáng kể, cho thấy điều kiện và chính sách cho vay của tổ chức tín dụng có ảnh hưởng mạnh đến rủi ro tín dụng. Các yếu tố khác như revol.util (tỷ lệ sử dụng tín dụng quay vòng), log.annual.inc (thu nhập hàng năm), inq.last.6mths (số lần hỏi vay trong 6 tháng gần nhất) và days.with.cr.line (số ngày có đường tín dụng) cũng góp phần nhất định vào khả năng dự đoán. Hai biến purpose\_debt\_consolidation và purpose\_small\_business tuy có trọng số nhỏ hơn nhưng vẫn được xếp vào nhóm quan trọng, cho thấy mục đích vay vốn cũng là yếu tố không thể bỏ qua. Như vậy, mô hình Random Forest cho thấy khả năng học được các quy luật thực tế, trong đó lãi suất, điểm tín dụng và chính sách cho vay là ba nhân tố chủ đạo chi phối rủi ro không trả được nợ.

#### 4. Tổng kết bước phân lớp

Chỉ số	XGBoost	Random Forest
Accuracy	0.780	0.639
ROC AUC	0.620	0.639
Precision (Class 1)	0.251	0.216
Recall (Class 1)	0.209	0.504
F1-Score (Class 1)	0.228	0.302
Weighted Avg F1	0.772	0.686

*Bảng 3 Kết quả tổng hợp kết quả chạy với mô hình XG Boost và Random Forest*

Sau khi áp dụng kỹ thuật SMOTE kết hợp GridSearchCV để xử lý mất cân bằng dữ liệu và tối ưu siêu tham số, hai mô hình XGBoost và Random Forest được huấn luyện nhằm dự báo rủi ro khách hàng không trả được nợ. Kết quả cho thấy XGBoost đạt Accuracy 0.780, Weighted F1-Score 0.772, và ROC AUC 0.620, trong khi Random Forest đạt Accuracy 0.639, Weighted F1-Score 0.686, nhưng ROC AUC cao hơn, ở mức 0.639. Mặc dù XGBoost có độ chính xác tổng thể cao hơn, song Random Forest lại cho recall lớp thiểu số vượt trội (0.504 so với 0.209), cho thấy mô hình này nhận diện được nhiều trường hợp khách hàng có rủi ro không trả được nợ hơn.

Mô hình Random Forest có một số ưu điểm nổi bật, bao gồm khả năng xử lý tốt dữ liệu phi tuyến và nhiễu, khả năng giảm overfitting nhờ kết hợp ngẫu nhiên nhiều cây quyết định, và độ ổn định cao khi dữ liệu được mở rộng. Ngoài ra, Random Forest còn cung cấp chỉ số tầm quan trọng của biến (feature importance), hỗ trợ quá trình phân tích và giải thích các yếu tố tác động đến khả năng không trả nợ của khách hàng — đây là lợi thế quan trọng trong các bài toán tài chính yêu cầu tính minh bạch.

Tuy nhiên, mô hình vẫn còn một số hạn chế. Precision cho lớp thiểu số ở mức thấp (0.216) cho thấy tỷ lệ báo sai vẫn còn khá cao, nghĩa là một số khách hàng được dự báo “có rủi ro” thực tế vẫn trả được nợ. Điều này có thể gây tổn kém chi phí giám sát hoặc đánh giá lại tín dụng. Ngoài ra, độ chính xác tổng thể của mô hình chưa cao, phản ánh rằng vẫn còn sự chồng lấn nhất định giữa hai nhóm khách hàng, ngay cả sau khi đã áp dụng SMOTE.

### Chương V: Kết luận (Conclusions & Discussion)

#### 1. Trả lời câu hỏi nghiên cứu

Dựa trên kết quả phân tích từ hai mô hình Random Forest và XGBoost, có thể thấy rằng việc xây dựng mô hình phân lớp nhằm dự đoán khả năng người vay không thanh toán

đầy đủ khoản vay là hoàn toàn khả thi, tuy nhiên độ chính xác hiện tại vẫn ở mức trung bình và cần được cải thiện thêm.

Cụ thể, mô hình XGBoost đạt độ chính xác 0.780 và ROC AUC 0.620, trong khi mô hình Random Forest có độ chính xác 0.639. Cả hai mô hình đều dự đoán khá tốt với nhóm khách hàng trả nợ đầy đủ (class 0), nhưng hiệu quả nhận diện nhóm không trả nợ đầy đủ (class 1) còn hạn chế – thể hiện qua precision thấp (chỉ 0.21–0.25) và recall chưa cao (0.20–0.50). Nguyên nhân chính xuất phát từ sự mất cân bằng dữ liệu, khi số lượng khách hàng không trả nợ chiếm tỷ lệ rất nhỏ so với tổng thể. Dù vậy, kết quả vẫn cho thấy các mô hình học máy có thể nhận diện được các tín hiệu rủi ro tín dụng dựa trên các đặc trưng tài chính và lịch sử tín dụng.

Về yếu tố ảnh hưởng lớn nhất đến rủi ro không thanh toán, cả hai mô hình đều thống nhất rằng lãi suất (int.rate), điểm tín dụng (fico) và chính sách tín dụng (credit.policy) là ba yếu tố có tác động mạnh nhất. Trong đó, lãi suất có độ quan trọng cao nhất, cho thấy khi chi phí vay tăng, khả năng không trả nợ đầy đủ cũng tăng theo. Điểm tín dụng phản ánh mức độ uy tín và lịch sử vay mượn của khách hàng, càng thấp thì rủi ro vỡ nợ càng cao. Trong khi đó, chính sách tín dụng thể hiện mức độ thắt chặt hay nới lỏng của tổ chức cho vay – yếu tố có thể làm thay đổi đáng kể xác suất rủi ro.

## **2. Nhận xét về hạn chế và hướng mở rộng.**

### **2.1. Hạn chế**

Hạn chế lớn nhất và rõ ràng nhất là hiệu suất dự đoán tổng thể của các mô hình. Mặc dù nhóm đã áp dụng các kỹ thuật xử lý mất cân bằng đa dạng (bao gồm over-sampling và under-sampling) và lựa chọn SMOTE làm giải pháp cuối cùng, đồng thời thực hiện tinh chỉnh siêu tham số nghiêm ngặt bằng GridSearchCV, các chỉ số đánh giá quan trọng (đặc biệt là F1-Score: 0.228 và Recall: 0.209 cho XGBoost; F1-Score: 0.302 cho Random Forest) vẫn ở mức thấp.

Để kiểm chứng, nhóm đã triển khai một loạt các thuật toán với các nguyên lý hoạt động khác nhau (bao gồm Logistic Regression, SVM, Random Forest và XGBoost). Kết quả nhất quán là không có mô hình nào đạt được hiệu suất vượt trội, tất cả đều gặp khó khăn trong việc nhận diện Lớp 1.

Điều này là một minh chứng mạnh mẽ, gợi ý rằng hạn chế không nằm ở việc lựa chọn mô hình hay quy trình tinh chỉnh, mà nằm ở bản chất của dữ liệu. Rất có khả năng tồn tại một sự chồng lấn (overlapping) cao giữa hai lớp trong không gian đặc trưng. Nói cách khác, các đặc trưng hiện có (dù đã qua bước kỹ thuật đặc trưng - feature engineering) không đủ sức mạnh để phân biệt rõ ràng giữa một khách hàng rủi ro (Lớp 1) và một khách hàng an toàn (Lớp 0). Kết quả này cho thấy rằng việc mở rộng hoặc tái

cấu trúc bộ biến đầu vào (thu thập thêm dữ liệu mới) là yêu cầu gần như bắt buộc để có thể cải thiện hiệu quả mô hình trong tương lai.

Yếu điểm này cũng có thể đến từ việc sử dụng Gridsearch CV, khi các giá trị lớp 1 thật đã nằm sâu trong vùng của lớp 0. Điều này không chỉ không làm rõ ranh giới mà còn gây thêm nhiễu (noise) cho thuật toán. Đây có thể là nguyên nhân lý giải kết quả của Random Forest: để cố gắng bao phủ các mẫu SMOTE "nhiều" này, mô hình đạt được Recall (0.504) tương đối, nhưng phải đánh đổi bằng một chỉ số Precision (0.216) cực kỳ thấp, dẫn đến F1-Score không được cải thiện.

Bên cạnh đó, hiệu suất dự báo bị hạn chế đáng kể do sự tương tác phi tuyến tính và không đồng nhất giữa các đặc điểm tài chính, hành vi và tâm lý đã tạo ra sự chồng lấn lớn giữa hai lớp rủi ro (vỡ nợ và không vỡ nợ) trong không gian đặc trưng. Chính những yếu tố này kết hợp với sự thiếu bao quát của các biến độc lập đã tạo rào cản nền tảng đối với khả năng dự báo của mô hình.

## **2.2. Hướng mở rộng**

Với kết quả hiện tại, có thể tin tưởng tương đối vào các yếu tố được mô hình đánh giá có tầm quan trọng cao trong việc dự báo khả năng không trả được nợ. Tuy nhiên, cần lưu ý rằng độ tin cậy tuyệt đối của insight còn phụ thuộc vào khả năng bao phủ dữ liệu thực tế và chất lượng đặc trưng đầu vào. Do đó, để củng cố kết quả, nhóm nên thực hiện thêm phân tích SHAP hoặc Partial Dependence Plot (PDP) nhằm làm rõ mức độ ảnh hưởng của từng biến độc lập đến khả năng vỡ nợ, từ đó giúp ra quyết định tín dụng chính xác và có căn cứ hơn trong thực tế. Hoặc, thử các biến thể như Balanced Random Forest hoặc kết hợp thêm ADASYN, SMOTEENN, hay thuật toán boosting để nâng cao độ nhạy mà không làm giảm mạnh độ chính xác.

**Tài liệu tham khảo:**

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. SSRN. <https://doi.org/10.2139/ssrn.1568864>