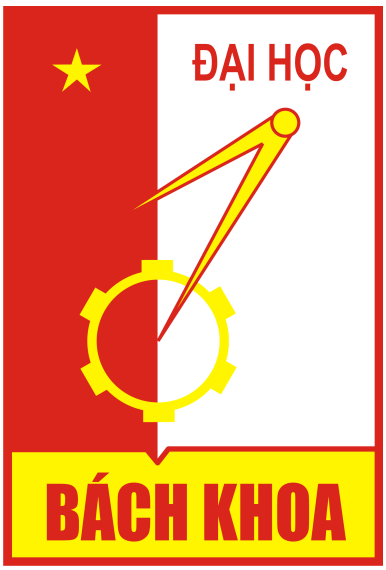
ĐẠI HỌC BÁCH KHOA HÀ NỘI

KHOA TOÁN - TIN



ĐỒ ÁN 2

Ứng dụng mô hình học máy trong phân tích và dự báo rủi ro tín dụng tại Home Credit

GVHD: TS. Phạm Huyền Linh
Sinh viên thực hiện: Lê Thị Hải Anh
MSSV: 20227211
Lớp: Hệ thống thông tin quản lý 02

Hà Nội, 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục đích và nội dung của đồ án

2. Kết quả đạt được

3. Ý thức làm việc của sinh viên

Hà Nội, ngày tháng năm 2025
Giảng viên hướng dẫn
(Ký và ghi rõ họ tên)

Lời mở đầu

Trong bối cảnh nền kinh tế số phát triển mạnh mẽ hiện nay, hoạt động tín dụng tiêu dùng đóng vai trò huyết mạch đối với sự tăng trưởng của các ngân hàng và tổ chức tài chính. Tuy nhiên, đi kèm với cơ hội mở rộng thị phần là thách thức to lớn trong việc quản trị rủi ro nợ xấu. Việc đánh giá chính xác khả năng trả nợ của khách hàng không chỉ giúp tối ưu hóa lợi nhuận mà còn là yếu tố then chốt quyết định sự phát triển bền vững của doanh nghiệp.

Với sự bùng nổ của dữ liệu lớn (Big Data), các phương pháp thẩm định truyền thống đang dần nhường chỗ cho các kỹ thuật Học máy (Machine Learning) tiên tiến. Các mô hình này cung cấp phương pháp luận khoa học để khai thác sâu dữ liệu lịch sử, hành vi khách hàng, từ đó đưa ra các dự báo chính xác và hỗ trợ ra quyết định tự động.

Xuất phát từ niềm đam mê với Khoa học dữ liệu và nhận thức sâu sắc về tầm quan trọng của việc ứng dụng công nghệ trong tài chính, em đã lựa chọn đề tài ”Xây dựng mô hình học máy dự báo rủi ro tín dụng”cho đồ án này. Mục tiêu chính của đồ án là đi sâu tìm hiểu quy trình xử lý dữ liệu thực tế, kỹ thuật trích xuất đặc trưng (Feature Engineering), và xây dựng các mô hình phân loại mạnh mẽ (như XGBoost, LightGBM, Random Forest) để giải quyết bài toán dự báo khả năng vỡ nợ. Để thực hiện mục tiêu đó, báo cáo của em được trình bày theo cấu trúc 6 chương, cụ thể:

- **Chương 1 & 2:** Giới thiệu tổng quan đề tài và Cơ sở lý thuyết về bài toán phân loại cũng như các thuật toán học máy liên quan.
- **Chương 3 & 4:** Tập trung vào việc Khám phá dữ liệu (EDA) và Tiền xử lý dữ liệu – giai đoạn quan trọng nhất bao gồm làm sạch, biến đổi đặc trưng và xử lý mất cân bằng mẫu.
- **Chương 5:** Xây dựng và Tối ưu hóa mô hình – trình bày quá trình huấn luyện, tinh chỉnh tham số (Hyperparameter Tuning), xây dựng mô hình kết hợp (Ensemble Learning) và triển khai ứng dụng thực tế.
- **Chương 6:** Kết luận và hướng phát triển.

Để hoàn thành đồ án này, bên cạnh sự nỗ lực của bản thân, em đã nhận được sự hướng dẫn vô cùng tận tâm, những chỉ dẫn chuyên môn sâu sắc và sự động viên quý báu từ TS. Phạm Huyền Linh. Cô đã định hướng phương pháp nghiên cứu và luôn sẵn lòng giải đáp mọi thắc mắc của em trong suốt quá trình thực hiện. Em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất đến Cô vì tất cả những hỗ trợ và tâm huyết mà Cô đã dành cho em.

Báo cáo này sẽ trình bày một cách hệ thống quy trình từ dữ liệu thô đến việc triển khai một hệ thống hỗ trợ ra quyết định (Dashboard). Mặc dù em đã rất cố gắng, song do kiến thức và kinh nghiệm thực tiễn còn hạn chế, báo cáo chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp từ Thầy/Cô chấm điểm để đồ

án được hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

Danh sách hình vẽ

Danh sách bảng

Danh mục ký hiệu và chữ viết tắt

AUC Area Under the Curve – Diện tích dưới đường cong.

CV Cross-Validation – Kiểm chứng chéo.

EDA Exploratory Data Analysis – Phân tích khám phá dữ liệu.

FN False Negative – Số trường hợp âm tính giả (Bỏ sót nợ xấu).

FP False Positive – Số trường hợp dương tính giả (Báo động giả).

Gini Gini Coefficient – Hệ số Gini (Chỉ số đánh giá xếp hạng tín dụng).

KS Kolmogorov-Smirnov – Chỉ số đo lường độ phân tách giữa hai phân phối.

LGBM LightGBM (Light Gradient Boosting Machine) – Thuật toán tăng cường Gradient nhẹ.

NaN Not a Number – Giá trị khuyết thiếu (Rỗng).

OOF Out-of-Fold – Kết quả dự báo trên tập kiểm thử trong quá trình kiểm chứng chéo.

RF Random Forest – Mô hình Rừng ngẫu nhiên.

ROC Receiver Operating Characteristic – Đường cong đặc trưng hoạt động.

SK_ID_CURR Mã định danh duy nhất của hồ sơ vay hiện tại.

TN True Negative – Số trường hợp âm tính thật (Dự báo đúng khách hàng tốt).

TPE Tree-structured Parzen Estimator – Thuật toán tối ưu hóa tham số (dùng trong Optuna).

TP True Positive – Số trường hợp dương tính thật (Dự báo đúng nợ xấu).

XGB XGBoost (Extreme Gradient Boosting) – Thuật toán tăng cường Gradient cực đại.

Chương 1

Giới thiệu đề tài

1.1 Lý do chọn đề tài

Trong hoạt động của hệ thống ngân hàng thương mại và các tổ chức tài chính, cấp tín dụng luôn được xem là hoạt động cốt lõi mang lại nguồn lợi nhuận chủ yếu, nhưng đồng thời cũng chứa đựng những rủi ro tiềm tàng lớn nhất. Một trong những thách thức lớn nhất mà các ngân hàng phải đối mặt là rủi ro tín dụng – nguy cơ khách hàng không thực hiện được nghĩa vụ trả nợ đúng hạn, dẫn đến gia tăng nợ xấu và ảnh hưởng trực tiếp đến thanh khoản cũng như sự an toàn vốn của tổ chức. Trước đây, quy trình thẩm định và phê duyệt khoản vay thường phụ thuộc nhiều vào phán đoán chủ quan của con người hoặc các hệ thống chấm điểm truyền thống với số lượng biến số hạn chế. Tuy nhiên, trước sự bùng nổ của dữ liệu lớn và sự đa dạng trong hành vi tiêu dùng hiện đại, các phương pháp thủ công này đang dần bộc lộ hạn chế về tốc độ xử lý, chi phí vận hành và đặc biệt là khả năng phát hiện các rủi ro tiềm ẩn phức tạp.

Sự phát triển mạnh mẽ của Khoa học dữ liệu và các thuật toán Học máy đã mở ra một hướng tiếp cận mới, cho phép các tổ chức tài chính tận dụng nguồn dữ liệu lịch sử khổng lồ để xây dựng các mô hình dự báo với độ chính xác cao hơn vượt trội. Việc ứng dụng công nghệ vào quy trình đánh giá tín dụng không chỉ giúp tự động hóa khâu ra quyết định, rút ngắn thời gian phê duyệt mà còn tối ưu hóa chiến lược quản trị rủi ro một cách minh bạch và khoa học. Xuất phát từ thực tiễn đó, em lựa chọn đề tài nghiên cứu về ứng dụng kỹ thuật học máy trong phân tích rủi ro tín dụng nhằm xây dựng một quy trình mô hình hóa hoàn chỉnh, có khả năng phân loại hồ sơ khách hàng hiệu quả để hỗ trợ việc ra quyết định cấp tín dụng trong ngân hàng.

1.2 Mục tiêu nghiên cứu

Mục tiêu tổng quát của đề tài là nghiên cứu, xây dựng và tối ưu hóa một quy trình đánh giá rủi ro tín dụng tự động dựa trên các kỹ thuật Học máy. Kết quả của nghiên cứu là một mô hình phân lớp có khả năng dự báo chính xác xác suất vỡ nợ của khách hàng cá nhân, từ đó cung cấp cơ sở định lượng để hỗ trợ các tổ chức tài chính trong việc ra quyết định phê duyệt khoản vay.

Để đạt được mục tiêu tổng quát trên, đề tài tập trung giải quyết các mục tiêu cụ thể sau:

- Thứ nhất, hệ thống hóa cơ sở lý thuyết liên quan đến quản trị rủi ro tín dụng trong ngân hàng và các thuật toán học máy tiên tiến thường được ứng dụng trong bài toán phân loại trên dữ liệu dạng bảng, trọng tâm là các mô hình học máy tổ hợp.
- Thứ hai, thực hiện quy trình phân tích và xử lý dữ liệu chuyên sâu. Do đặc thù dữ liệu tài chính thường phức tạp, phân tán và nhiễu, nghiên cứu sẽ tập trung vào kỹ thuật làm sạch dữ liệu, xử lý giá trị thiếu và đặc biệt là kỹ thuật trích chọn đặc trưng (Feature Engineering) để khai thác tối đa thông tin từ dữ liệu hành vi và lịch sử giao dịch.
- Thứ ba, xây dựng và huấn luyện các mô hình dự báo. Nghiên cứu sẽ tiến hành thử nghiệm các thuật toán khác nhau trên tập dữ liệu thực nghiệm. Quá trình này bao gồm cả việc tinh chỉnh tham số (Hyperparameter tuning) để tìm ra cấu hình tối ưu nhất cho bài toán.
- Cuối cùng, đánh giá hiệu quả mô hình. Sử dụng các thước đo chuẩn trong ngành khoa học dữ liệu và tài chính để so sánh hiệu năng giữa các mô hình, qua đó đề xuất mô hình phù hợp nhất có khả năng cân bằng giữa độ chính xác và hiệu suất tính toán khi triển khai thực tế.

1.3 Phạm vi và giới hạn

Phạm vi dữ liệu và đối tượng nghiên cứu

Do đặc thù về quy định bảo mật thông tin khách hàng và dữ liệu tài chính tại các ngân hàng thương mại, việc tiếp cận nguồn dữ liệu thực tế để phục vụ mục đích nghiên cứu học thuật gặp nhiều hạn chế. Vì vậy, đề tài sử dụng bộ dữ liệu công khai từ tập đoàn Home Credit làm dữ liệu thực nghiệm thay thế. Đây là bộ dữ liệu có quy mô lớn và cấu trúc phức tạp, bao gồm thông tin định danh, lịch sử tín dụng tại các tổ chức khác (Credit Bureau), lịch sử thẻ tín dụng và quá trình trả góp. Cấu trúc này có độ tương đồng cao với hệ thống dữ liệu tại các ngân hàng, đảm bảo tính đại diện để kiểm chứng hiệu quả của quy trình xây dựng mô hình.

Phạm vi nội dung và kỹ thuật

Nghiên cứu tập trung giải quyết bài toán phân lớp nhị phân (Binary Classification) nhằm dự báo xác suất vỡ nợ của khách hàng. Nội dung thực hiện bao gồm trọn vẹn quy trình từ tiền xử lý dữ liệu , trích chọn đặc trưng đến huấn luyện và đánh giá mô hình. Về mặt công cụ, đề tài sử dụng ngôn ngữ lập trình Python kết hợp với các thư viện phân tích dữ liệu và học máy mã nguồn mở như Random Forest, LightGBM và XGBoost để triển khai các thuật toán.

Giới hạn của đề tài

Nghiên cứu được thực hiện với một số giới hạn nhất định. Thứ nhất, mô hình chỉ dựa trên phân tích định lượng từ dữ liệu lịch sử và hành vi giao dịch, chưa bao gồm các đánh giá định tính thường có trong quy trình thẩm định truyền thống. Thứ hai, dữ liệu sử dụng là dữ liệu quá khứ, có thể chưa phản ánh trọn vẹn các biến động kinh tế vĩ mô mới nhất ảnh hưởng đến khả năng trả nợ. Cuối cùng, do khối lượng dữ liệu lớn và sự phức tạp của các

mô hình học máy hiện đại, việc tối ưu hóa siêu tham số (Hyperparameter tuning) sẽ được thực hiện trong giới hạn cho phép của tài nguyên phần cứng cá nhân nhằm đảm bảo thời gian huấn luyện khả thi.

Chương 2

Cơ sở lý thuyết

2.1 Tổng quan về bài toán phân loại

2.1.1 Bài toán phân loại

Bài toán phân loại là một trong những bài toán cốt lõi và phổ biến nhất trong lĩnh vực Học máy có giám sát (Supervised Learning). Mục tiêu cơ bản của bài toán là dự đoán nhãn lớp của các điểm dữ liệu mới dựa trên việc học các quy luật từ tập dữ liệu đã được gán nhãn trước đó.

Về mặt toán học, giả sử ta có không gian đầu vào $X \subseteq \mathbb{R}^n$ chứa các vector đặc trưng đại diện cho đối tượng, và không gian đầu ra Y là tập hợp hữu hạn các nhãn lớp rời rạc $Y = \{c_1, c_2, \ldots, c_k\}$.

Nhiệm vụ của thuật toán học máy là tìm ra một hàm ánh xạ (mapping function) $f : X \to Y$ sao cho hàm này có thể phân chia không gian đặc trưng thành các vùng quyết định tương ứng với từng lớp, đồng thời tối thiểu hóa sai số dự báo trên tập dữ liệu chưa biết . [3]

Tùy thuộc vào số lượng phần tử trong tập Y, bài toán phân loại thường được chia thành:

- Phân loại nhị phân (Binary Classification): $|Y|$=2
- Phân loại đa lớp (Multi-class Classification): $|Y|$ >2

2.1.2 Bài toán phân loại nhị phân

Phân loại nhị phân là trường hợp cơ bản và phổ biến nhất của bài toán phân loại trong học máy có giám sát. Trong bài toán này, nhiệm vụ đặt ra là phân chia các điểm dữ liệu trong không gian đặc trưng vào một trong hai nhóm nhãn lớp rời rạc và loại trừ nhau.

Định nghĩa toán học: Cho tập dữ liệu huấn luyện $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, trong đó:

- $x_i \in \mathbb{R}^d$ là vector đặc trưng d-chiều của điểm dữ liệu thứ i.
- $y_i \in Y$ là nhãn lớp tương ứng. Trong phân loại nhị phân, tập nhãn đầu ra được định nghĩa là $Y = \{0, 1\}$ (hoặc đôi khi là $\{-1, 1\}$).
  - Lớp 1 thường được gọi là lớp Dương (Positive class).

– Lớp 0 thường được gọi là lớp Âm (Negative class).

Mục tiêu của bài toán:

Mục tiêu là xây dựng (học) một hàm ánh xạ $f : \mathbb{R}^d \to Y$ từ tập dữ liệu huấn luyện sao cho hàm này có thể dự đoán chính xác nhãn y cho một vector đầu vào x mới chưa từng xuất hiện.

Trong thực tế, thay vì gán nhãn cứng trực tiếp, hầu hết các mô hình học máy hiện đại đều ước lượng *xác suất hậu nghiệm* (posterior probability) $P(y = 1 \mid x)$ — xác suất để mẫu x thuộc về lớp Dương. Khi đó, hàm dự báo có dạng:

$$\hat{y} = f(x) = \begin{cases} 1 & \text{nếu } P(y = 1 \mid x) \geq \theta \\ 0 & \text{nếu } P(y = 1 \mid x) < \theta \end{cases}$$

Trong đó, θ là một ngưỡng quyết định, thường được đặt mặc định là 0.5.[3]

2.2 Một số mô hình học máy liên quan

2.2.1 Tổng quan về mô hình học máy

Machine Learning (ML) là một lĩnh vực thuộc trí tuệ nhân tạo (AI), tập trung vào phát triển các thuật toán cho phép máy tính tự động cải thiện và tăng độ chính xác thông qua kinh nghiệm và dựa trên dữ liệu. Nói một cách đơn giản, ML cho phép máy tính học từ dữ liệu và đưa ra quyết định hoặc dự đoán mà không cần được lập trình rõ ràng cho từng tác vụ.

Dựa trên đặc thù của dữ liệu đầu vào và cơ chế phản hồi trong quá trình huấn luyện, các thuật toán học máy thường được chia thành 4 nhóm phương pháp chính:

- Học máy có giám sát (Supervised Learning): Sử dụng tập dữ liệu đã được gán nhãn (có đầu vào và kết quả đầu ra) để huấn luyện mô hình tìm ra quy luật tương quan, từ đó thực hiện phân loại hoặc dự báo cho dữ liệu mới. Đây là phương pháp cốt lõi được áp dụng trong đồ án này.
- Học máy không giám sát (Unsupervised Learning): Làm việc với dữ liệu chưa được gán nhãn nhằm tự động phát hiện các cấu trúc ẩn, phân cụm hoặc tìm ra các mẫu tương đồng trong tập dữ liệu.
- Học bán giám sát (Semi-supervised Learning): Là sự kết hợp giữa một lượng nhỏ dữ liệu có nhãn và lượng lớn dữ liệu không nhãn, giúp tận dụng ưu điểm của cả hai để cải thiện độ chính xác mà không tốn nhiều chi phí gán nhãn thủ công.
- Học tăng cường (Reinforcement Learning): Huấn luyện hệ thống thông qua cơ chế ”thử và sai”trong môi trường tương tác, với mục tiêu tối đa hóa phần thưởng tích lũy từ các quyết định hành động.[3]

2.2.2 Mô hình Rừng ngẫu nhiên (Random Forest)

Random Forest là một thuật toán học máy thuộc nhóm học có giám sát và được sử dụng phổ biến trong các bài toán phân loại và hồi quy . Thuật toán này là một dạng của tập hợp học, nơi mà nhiều mô hình yếu, cụ thể là các cây quyết định, được kết hợp lại để tạo

thành một mô hình mạnh mẽ hơn.

Cơ chế hoạt động

Quá trình huấn luyện của Random Forest bắt đầu bằng việc tạo ra nhiều tập dữ liệu bootstrap từ tập huấn luyện ban đầu. Trên mỗi tập dữ liệu được lấy mẫu ngẫu nhiên có hoàn lại này, mô hình xây dựng một cây quyết định độc lập. Tại mỗi nút của cây, thuật toán lựa chọn ngẫu nhiên một tập con các đặc trưng và chọn điểm tách tối ưu dựa trên chỉ số độ hỗn loạn (impurity), thường dùng chỉ số Gini:

$$Gini(t) = 1 - \sum_{k=1}^{K} p_k^2,$$

với p_k là tỷ lệ mẫu thuộc lớp k trong nút. Sau khi toàn bộ các cây được xây dựng, dự đoán cuối cùng được tổng hợp bằng bỏ phiếu đa số đối với bài toán phân loại:

$$\hat{y} = \text{mode}\{h_b(x)\}_{b=1}^{B},$$

hoặc lấy trung bình đối với bài toán hồi quy:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} h_b(x).$$

Cách xây dựng cây độc lập và kết hợp lại này giúp Random Forest giảm phương sai và tăng tính ổn định của mô hình.[2]

2.2.3 Mô hình LightGBM

Light Gradient Boosting Machine (LightGBM) – là một thư viện học máy nổi bật về hiệu năng. Nó được phát triển với mục đích chính là tạo ra các mô hình học máy dựa trên cấu trúc cây quyết định, sử dụng kỹ thuật boosting tiên tiến.[2]

Cơ chế hoạt động

LightGBM là một phương pháp Gradient Boosting dựa trên việc xây dựng các cây quyết định nhằm tối thiểu hoá hàm mất mát. Ở vòng lặp thứ t, mô hình xây dựng một cây mới để khớp với đạo hàm bậc nhất của hàm mất mát theo đầu ra dự đoán:

$$g_i^{(t)} = \frac{\partial L(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i},$$

đồng thời khai thác cả đạo hàm bậc hai:

$$h_i^{(t)} = \frac{\partial^2 L(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^2}.$$

LightGBM sử dụng cấu trúc Histogram-based Decision Tree, trong đó các đặc trưng liên tục được chia thành các nhóm giá trị (bins) nhằm giảm chi phí tính toán. Khi đánh giá một điểm tách, mô hình sử dụng công thức tăng ích (gain):

$$Gain = \frac{1}{2} \left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) - \gamma,$$

với G_L, H_L và G_R, H_R lần lượt là tổng gradient và Hessian của các nút con. LightGBM phát triển cây theo chiều lá (leaf-wise), giúp giảm nhanh hàm mất mát. Dự đoán sau vòng lặp t được cập nhật như:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta f_t(x).$$

2.2.4 Mô hình XGBoost

XGBoost (Extreme Gradient Boosting) là một trong những thuật toán học máy (học có giám sát) mạnh mẽ và phổ biến nhất hiện nay, đặc biệt hiệu quả trong các bài toán phân loại và hồi quy có dữ liệu lớn và phức tạp. Thuật toán này được đề xuất bởi Chen và Guestrin, dựa trên khung Boosting truyền thống nhưng được cải tiến đáng kể về mặt hiệu suất tính toán và khả năng tổng quát hóa mô hình. XGBoost sử dụng một phương pháp tối ưu hóa độ dốc (gradient) theo hình thức cộng dồn, kết hợp nhiều cây quyết định yếu để xây dựng một mô hình mạnh thông qua việc giảm thiểu hàm tổn thất bằng cách thêm từng cây một cách tuần tự. Điểm nổi bật của XGBoost là khả năng xử lý tốt dữ liệu thiếu, tự động lựa chọn đặc trưng quan trọng và khả năng song song hóa trong quá trình huấn luyện, nhờ đó cải thiện đáng kể tốc độ so với các thuật toán Boosting truyền thống như AdaBoost hay Gradient Boosting Machine.

Cơ chế hoạt động

XGBoost cũng hoạt động theo nguyên lý Gradient Boosting nhưng thêm điều chuẩn L_1 và L_2 nhằm kiểm soát độ phức tạp của mô hình. Tại vòng lặp thứ t, hàm mục tiêu cần tối ưu là:

$$Obj^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t),$$

trong đó g_i và h_i là gradient và Hessian, còn $\Omega(f_t)$ thể hiện độ phức tạp của cây:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2,$$

với T là số lá và w_j là trọng số của lá thứ j. Giá trị tăng ích khi thực hiện tách một nút được xác định bởi:

$$Gain = \frac{1}{2}\left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right) - \gamma.$$

Sau khi cấu trúc cây tối ưu được xác định, trọng số của mỗi lá được tính bằng:

$$w_j = -\frac{G_j}{H_j + \lambda}.$$

Dự đoán cuối cùng của mô hình sau t vòng lặp được xác định bởi:

$$\hat{y}^{(t)} = \sum_{k=1}^{t} \eta f_k(x).$$

XGBoost nhờ vậy đạt khả năng tổng quát hoá tốt, mạnh và hạn chế overfitting hiệu quả.[2]

Chương 3

Tổng quan bài toán phân tích rủi ro tín dụng

3.1 Phát biểu bài toán

Bài toán được đặt ra là xây dựng mô hình dự báo khả năng vỡ nợ của khách hàng cá nhân, dựa trên bộ dữ liệu thực tế từ tập đoàn Home Credit – đơn vị cung cấp dịch vụ tài chính cho phân khúc khách hàng chưa có hoặc ít lịch sử tín dụng ngân hàng. Đây là bài toán phân loại nhị phân điển hình, được giải quyết bằng cách ứng dụng các thuật toán học máy tổ hợp tiên tiến bao gồm Random Forest, XGBoost và LightGBM. Các mô hình này sẽ được huấn luyện trên dữ liệu lịch sử đa chiều, sau đó được đánh giá và so sánh thông qua các chỉ số đo lường hiệu suất chuyên biệt như AUC và Gini để xác định mô hình tối ưu nhất. Mục tiêu chính của bài toán là cung cấp một công cụ định lượng chính xác, hỗ trợ các tổ chức tài chính tự động hóa quy trình phê duyệt, giảm thiểu rủi ro nợ xấu và tối ưu hóa danh mục cho vay.

Việc áp dụng đồng thời các mô hình Random Forest, XGBoost và LightGBM vào bài toán rủi ro tín dụng mang lại góc nhìn khách quan và toàn diện về hiệu năng của các thuật toán học máy hiện đại. Điều này giúp so sánh khả năng xử lý dữ liệu lớn, dữ liệu bị thiếu và dữ liệu mất cân bằng – những đặc điểm cố hữu của dữ liệu tài chính. Hơn nữa, quá trình nghiên cứu không chỉ dừng lại ở việc tìm ra mô hình có độ chính xác cao nhất mà còn đi sâu vào việc phân tích tầm quan trọng của các đặc trưng . Qua đó, nghiên cứu cung cấp những hiểu biết sâu sắc về hành vi khách hàng và các yếu tố then chốt ảnh hưởng đến quyết định cấp tín dụng, tạo tiền đề cho việc xây dựng các hệ thống chấm điểm tín dụng minh bạch và hiệu quả trong thực tế.

3.2 Mô tả dữ liệu

3.2.1 Hệ thống dữ liệu

- Tên bộ dữ liệu: Home Credit Default Risk
- Nguồn: Kaggle (Được cung cấp và tổ chức bởi Home Credit Group).
- Loại dữ liệu: Dữ liệu bảng có cấu trúc quan hệ.
- Kích thước: Khoảng 2.5 GB (Tổng dung lượng sau khi giải nén toàn bộ các bảng).

- Nội dung: Bộ dữ liệu chứa thông tin toàn diện xoay quanh hồ sơ tín dụng của khách hàng cá nhân. Nội dung bao gồm thông tin nhân khẩu học, tình trạng tài chính hiện tại, lịch sử tín dụng tại các tổ chức khác (Credit Bureau), lịch sử sử dụng thẻ tín dụng, lịch sử trả góp và các khoản vay trước đây tại Home Credit.
- Số bảng và định dạng: Bộ dữ liệu bao gồm 7 bảng dữ liệu chính, được lưu trữ dưới dạng 8 tệp tin định dạng .csv (do bảng chính application được chia tách thành tập huấn luyện và kiểm tra).
- Số bản ghi (Bảng chính):
  - Tập huấn luyện (application_train): 307,511 bản ghi.
  - Tập kiểm tra (application_test): 48,744 bản ghi.
  - Các bảng phụ (như bureau_balance, installments_payments...) chứa từ vài triệu đến hàng chục triệu dòng dữ liệu lịch sử.

3.2.2 Giới thiệu các bảng dữ liệu

Bộ dữ liệu "Home Credit Default Risk" bao gồm 8 tệp CSV chính, mỗi tệp cung cấp một khía cạnh khác nhau về thông tin của khách hàng và lịch sử tín dụng của họ. Việc hiểu rõ từng bảng và mối quan hệ giữa chúng là rất quan trọng để có thể tổng hợp dữ liệu một cách hiệu quả.

a. Bảng *application_train.csv application_test.csv*

- **Mô tả:** Đây là các bảng dữ liệu cốt lõi, chứa thông tin chi tiết về các đơn xin vay vốn hiện tại của khách hàng tại Home Credit.
  - *application_train.csv*: Tập huấn luyện, bao gồm biến mục tiêu TARGET (0: khoản vay được trả đúng hạn, 1: khoản vay bị vỡ nợ).
  - *application_test.csv*: Tập kiểm tra, không chứa biến TARGET.
- **Kích thước:**
  - *application_train.csv:* 307,511 dòng và 122 cột.
  - *application_test.csv:* 48,744 dòng và 121 cột.
- **Vai trò:** Cung cấp các đặc trưng cơ bản về khách hàng (thông tin nhân khẩu học, thu nhập, trình độ học vấn, tình trạng nhà ở, v.v.) và các thông số của khoản vay hiện tại (số tiền vay, khoản trả góp hàng năm). Đây là bảng chính mà chúng ta sẽ dự đoán.

b. Bảng *bureau.csv*

- **Mô tả:** Chứa thông tin về lịch sử tín dụng của khách hàng từ các tổ chức tín dụng khác (bên ngoài Home Credit). Mỗi hàng là một khoản vay trước đó của khách hàng tại một tổ chức khác. Một khách hàng có thể có nhiều khoản vay trong bảng này.
- **Kích thước:** 1,716,428 dòng và 17 cột.
- **Vai trò:** Cung cấp cái nhìn toàn diện về hành vi tín dụng của khách hàng ngoài Home Credit, bao gồm trạng thái khoản vay (active, closed, defaulted), số tiền vay, số ngày quá hạn, v.v.

c. Bảng *bureau_balance.csv*

- **Mô tả:** Chứa thông tin chi tiết hàng tháng về trạng thái của các khoản vay trong bureau.csv. Mỗi hàng đại diện cho trạng thái của một khoản vay cụ thể (*SK_ID_BUREAU*) trong một tháng cụ thể (*MONTHS_BALANCE*).
- **Kích thước:** 27,299,925 dòng và 3 cột.
- **Vai trò:** Cung cấp dữ liệu chuỗi thời gian về trạng thái các khoản vay bên ngoài, cho phép phân tích xu hướng quá hạn hoặc thanh toán đúng hạn theo thời gian.

d. Bảng *previous_application.csv*

- **Mô tả:** Chứa thông tin về các khoản vay trước đó mà khách hàng đã nộp đơn tại Home Credit. Mỗi hàng là một đơn xin vay trước đó của khách hàng. Một khách hàng có thể có nhiều đơn xin vay trong bảng này.
- **Kích thước:** 1,670,214 dòng và 37 cột.
- **Vai trò:** Cho biết lịch sử tương tác của khách hàng với Home Credit, bao gồm việc các đơn xin vay trước đó có được chấp thuận, từ chối hay hủy bỏ, cũng như các điều khoản của chúng.

e. Bảng *POS_CASH_balance.csv*

- **Mô tả:** Chứa thông tin hàng tháng về các khoản vay tiền mặt tại điểm bán hàng (Point of Sale - POS) và các khoản vay tiền mặt của khách hàng tại Home Credit.
- **Kích thước:** 10,001,358 dòng và 8 cột.
- **Vai trò:** Cung cấp dữ liệu chuỗi thời gian về trạng thái thanh toán của các khoản vay POS/tiền mặt, bao gồm số ngày quá hạn (DPD) và số kỳ thanh toán còn lại.

f. Bảng *installments_payments.csv*

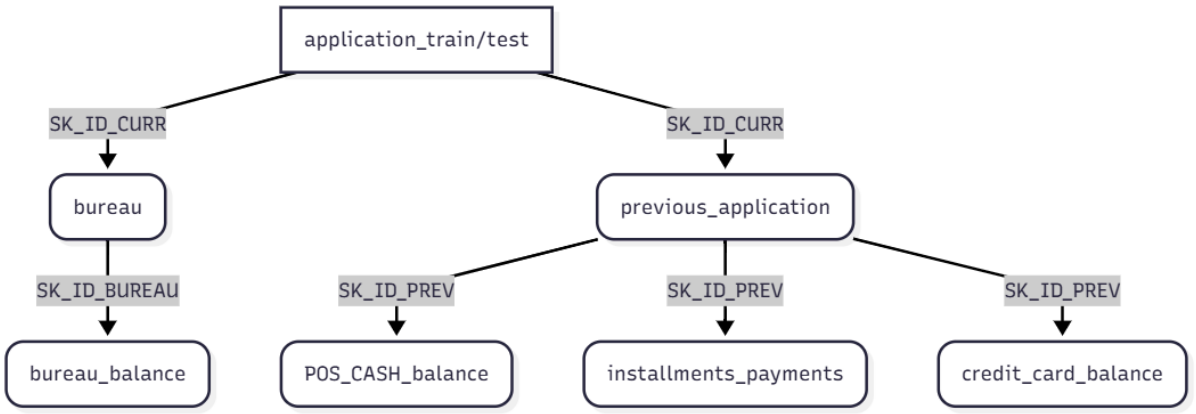
- **Mô tả:** Chứa lịch sử thanh toán các khoản trả góp của khách hàng cho các khoản vay trước đó tại Home Credit. Mỗi hàng là một lần thanh toán trả góp.
- **Kích thước:** 13,605,401 dòng và 8 cột.
- **Vai trò:** Cung cấp thông tin chi tiết về hành vi thanh toán thực tế của khách hàng, bao gồm ngày dự kiến và ngày thực tế thanh toán, số tiền dự kiến và số tiền thực tế đã thanh toán.

g. Bảng *credit_card_balance.csv*

- **Mô tả:** Chứa thông tin hàng tháng về số dư thẻ tín dụng của khách hàng tại Home Credit.
- **Kích thước:** 3,840,312 dòng và 23 cột.
- **Vai trò:** Cung cấp dữ liệu chuỗi thời gian về việc sử dụng thẻ tín dụng, hạn mức, số dư, số tiền thanh toán và trạng thái quá hạn.

h. Sơ đồ mối quan hệ giữa các bảng dữ liệu

Để hình dung rõ hơn cấu trúc của bộ dữ liệu, sơ đồ sau minh họa mối quan hệ giữa các bảng thông qua các khóa liên kết:



Hình 3.1: Sơ đồ mối quan hệ giữa các bảng dữ liệu

- Bảng *application_train/test* là trung tâm, chứa thông tin về khoản vay chính và khách hàng.
- Bảng *bureau* và *previous_application* cung cấp lịch sử tín dụng từ các nguồn khác nhau cho mỗi khách hàng (*SK_ID_CURR*).
- Các bảng *bureau_balance, POS_CASH_balance, installments_payments*, và *credit_card_balance* cung cấp thông tin chi tiết, theo thời gian hoặc theo giao dịch, cho các khoản mục trong bureau (*SK_ID_BUREAU*) hoặc *previous_application* (*SK_ID_PREV*).

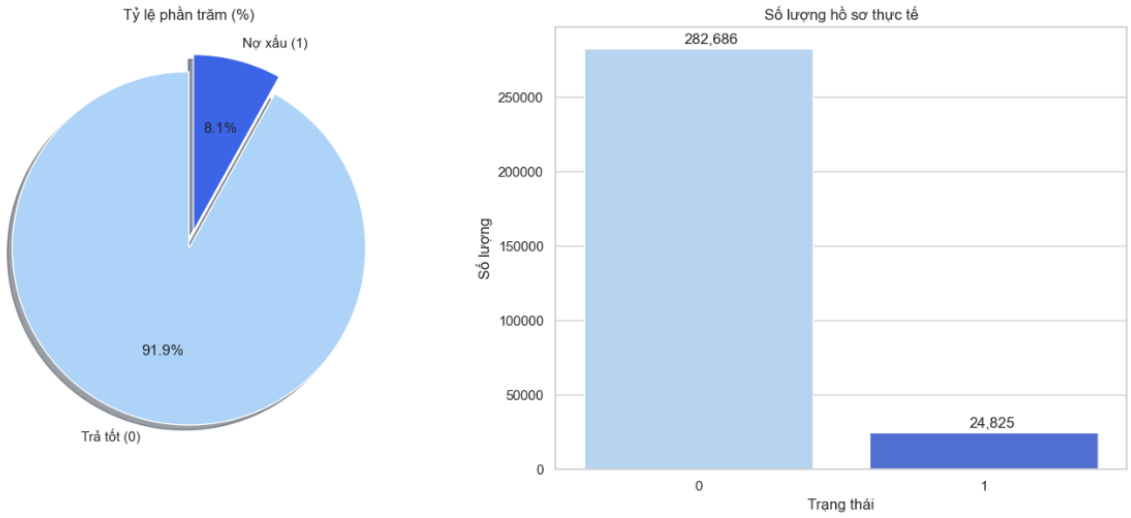
3.3 Khám phá dữ liệu

3.3.1 Tổng quan về bộ dữ liệu

Table Name	Rows	Cols	Duplicates	Total Missing Cells	Missing (%)	Null Cols Count	Memory (MB)
app_train	307511	122	0	9152465	24.400000	67	504.990000
app_test	48744	121	0	1404419	23.810000	64	79.690000
bureau	1716428	17	0	3939947	13.500000	7	472.820000
bureau_bal	27299925	3	0	0	0.000000	0	1718.330000
prev_app	1670214	37	0	11109336	17.980000	16	1703.010000
pos_cash	10001358	8	0	52158	0.070000	2	1060.950000
installments	13605401	8	0	5810	0.010000	2	830.410000
cc_bal	3840312	23	0	5877356	6.650000	9	846.390000

Hình 3.2: Tổng quan về bộ dữ liệu

3.3.2 Phân bố biến mục tiêu TARGET



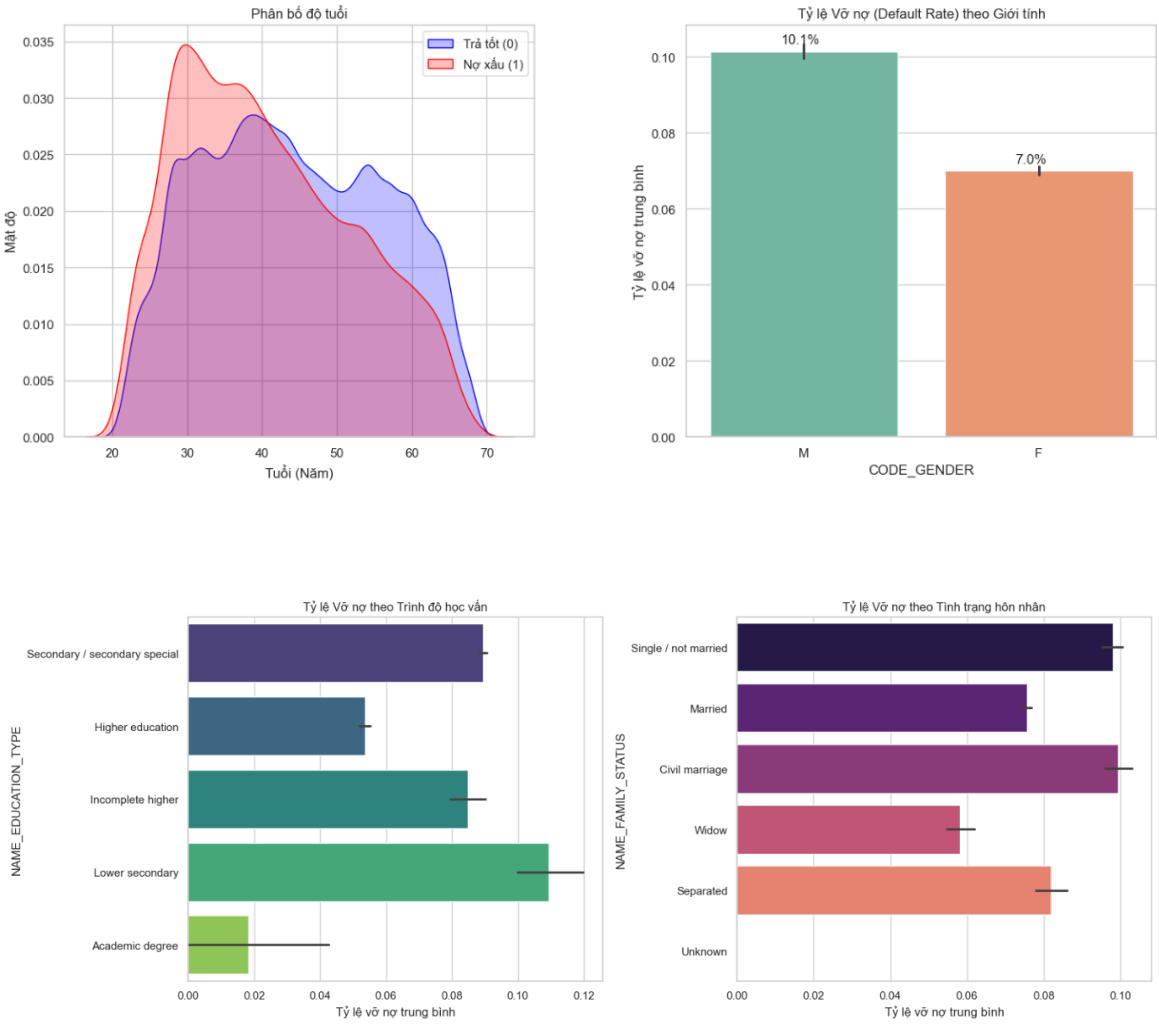
Hình 3.3: Tỷ lệ mất cân bằng

- Mất cân bằng dữ liệu nghiêm trọng: Nhóm khách hàng trả nợ tốt (0) chiếm ưu thế tuyệt đối với 91.9% (hơn $282k$ hồ sơ), trong khi nhóm nợ xấu/rủi ro (1) chỉ chiếm thiểu số 8.1%(khoảng $24k$ hồ sơ).

3.3.3 Phân tích đặc điểm khách hàng

	AGE_YEARS	CNT_CHILDREN
count	307511	307511
mean	43.936973	0.417052
std	11.956133	0.722121
min	20.517808	0.000000
25%	34.008219	0.000000
50%	43.150685	0.000000
75%	53.923288	0.000000
max	69.120548	0.000000

Bảng 3.1: Thống kê mô tả tuổi và số con



Hình 3.4: Biểu đồ phân tích đặc điểm khách hàng

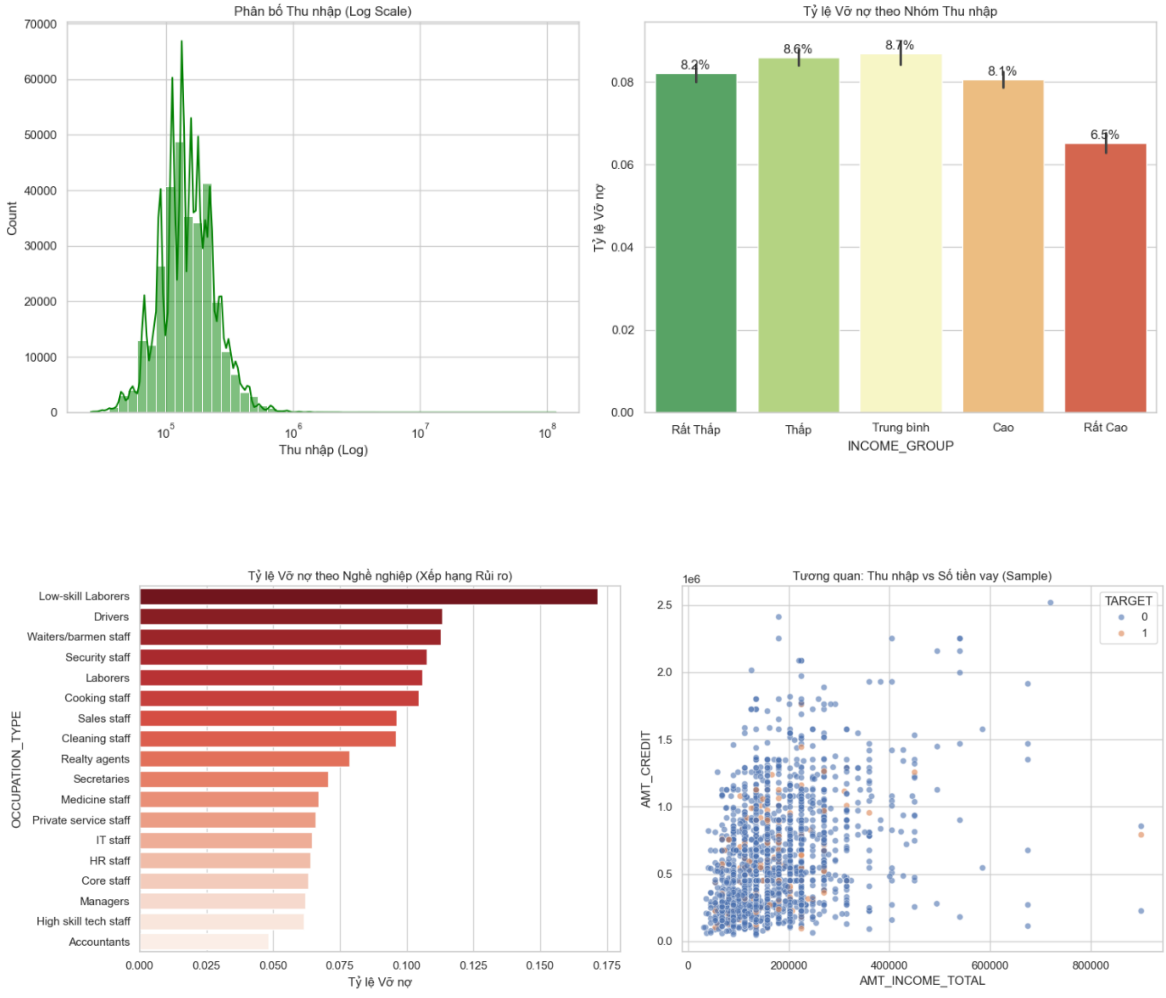
- Thống kê mô tả (Descriptive Statistics):
  - Tuổi (‘AGE_YEARS‘): Độ tuổi trung bình là 44 tuổi, dải tuổi từ 20 đến 69. Dữ liệu này sạch và hợp lý, không có giá trị ngoại lai (outlier) vô lý.
  - Số con (‘CNT_CHILDREN‘): Đa số khách hàng có ít con (trung bình 0.41, 75% số người có ≤ 1 con). Tuy nhiên, giá trị Max = 19 là một outlier cực đoan cần lưu ý xử lý hoặc kiểm tra lại.
- Đường KDE cho thấy nhóm nợ xấu nghiêng về phía tuổi trẻ. Điều này chứng tỏ khách hàng 20–30 tuổi có rủi ro vỡ nợ cao hơn nhóm lớn tuổi.
- Phân tích theo giới tính (Gender): Biểu đồ giới tính cho thấy tỷ lệ vỡ nợ của nam giới cao hơn rõ rệt so với nữ giới. Cột đại diện cho nhóm nam (10.1%) vượt trội hoàn toàn so với nhóm nữ (7.0%). Điều này phản ánh rằng nam giới có mức độ rủi ro tín dụng cao hơn đáng kể và cần được chú ý trong quá trình đánh giá khả năng trả nợ.
- Trình độ học vấn và rủi ro tín dụng (Education): Học vấn càng thấp rủi ro càng cao. Nhóm Lower secondary vượt 10% nợ xấu, trong khi nhóm Academic degree có tỷ lệ thấp nhất.

- Tình trạng hôn nhân (Family Status): Nhóm Độc thân và Kết hôn dân sự có rủi ro cao nhất. Ngược lại, nhóm Góa phụ là nhóm trả nợ tốt và ít rủi ro nhất.

3.3.4 Phân tích thông tin về thu nhập

count	3.075×10^5
mean	1.688×10^5
std	2.371×10^5
min	2.565×10^4
25%	1.125×10^5
50%	1.471×10^5
75%	2.025×10^5
max	1.170×10^8

Bảng 3.2: Thống kê cột thu nhập



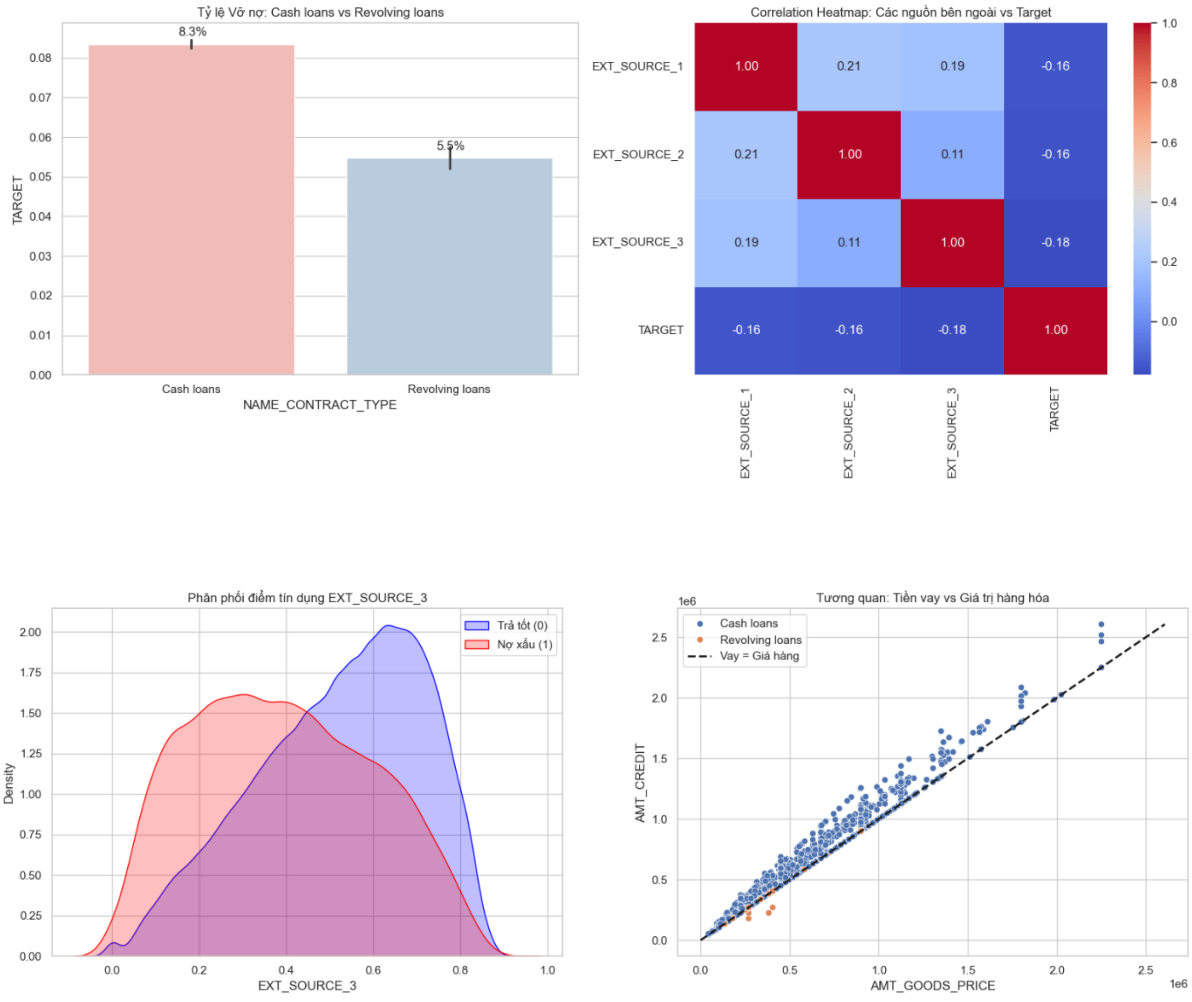
Hình 3.5: Biểu đồ khám phá về thông tin về thu nhập

- Thống kê mô tả cho thấy thu nhập có nhiều giá trị ngoại lai cực đoan: trung bình khoảng 168 nghìn nhưng giá trị lớn nhất lên tới 117 triệu, phản ánh dữ liệu phân tán mạnh và cần xử lý outlier trước khi huấn luyện mô hình. Biểu đồ phân bố thu nhập

dạng logarit cho thấy phân phối gần chuẩn, tập trung quanh mức 10^5, phù hợp cho việc đưa vào mô hình.

- Phân tích tỷ lệ vỡ nợ theo thu nhập cho thấy nhóm "trung bình" có rủi ro cao nhất (8.7%), trong khi rủi ro chỉ giảm đáng kể ở nhóm thu nhập rất cao (6.5%), cho thấy thu nhập đơn lẻ không luôn phản ánh khả năng trả nợ. Nghề nghiệp phân hóa mạnh: lao động tay chân (Low-skill Laborers, Drivers) có rủi ro gấp đôi so với lao động có kỹ năng cao (Accountants, High-skill staff), chứng tỏ đây là biến dự báo quan trọng.
- Tương quan giữa thu nhập và khoản vay thể hiện mối quan hệ tuyến tính dương, với các khoản vay lớn không tương xứng với thu nhập có xu hướng rủi ro cao. Tỷ lệ gánh nặng nợ (Annuity/Income) giữa hai nhóm quá sát nhau (18.0% vs 18.5%), cho thấy biến này đơn lẻ chưa đủ khả năng phân tách rủi ro.

3.3.5 Khám phá về khoản vay

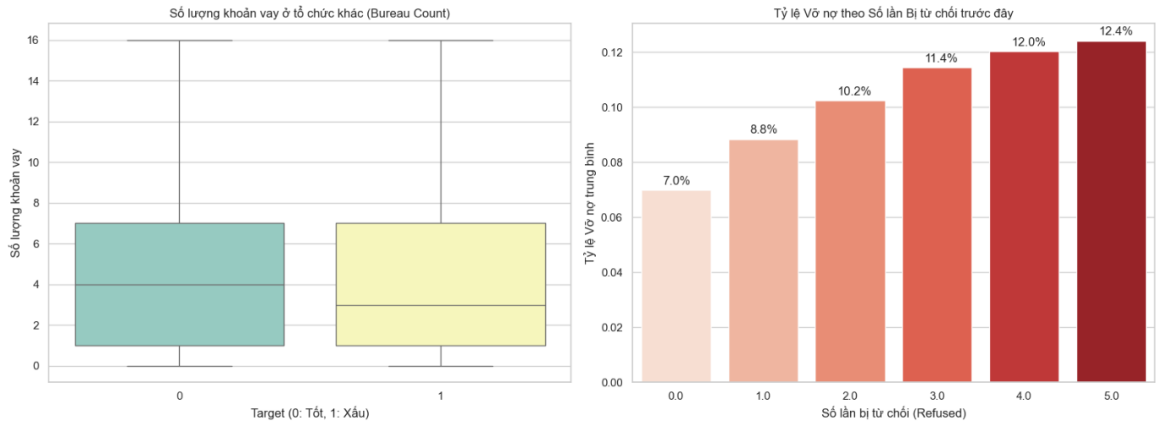


Hình 3.6: Biểu đồ khám phá về tài chính về khoản vay

- Nhóm vay tiền mặt (*Cash loans*) có tỷ lệ vỡ nợ cao hơn đáng kể (8.3%) so với vay quay vòng (*Revolving loans*, 5.5%), phản ánh nhu cầu thanh khoản cấp bách hoặc hồ sơ tín dụng yếu hơn.

- Các biến external (EXT_SOURCE_1, 2, 3) có tương quan nghịch rõ với rủi ro, cho thấy điểm số cao tương ứng với khả năng trả nợ tốt.
- Phân phối KDE của EXT_SOURCE_3 tách biệt rõ ràng: điểm thấp <0.2 rủi ro cao, điểm >0.7 an toàn.
- Quan sát Scatter Plot cho thấy khách hàng thường vay bằng giá trị món hàng; các điểm vượt lên trên đường chéo biểu thị vay lớn hơn giá trị thực, có thể do phí bảo hiểm hoặc tiền mặt bổ sung.

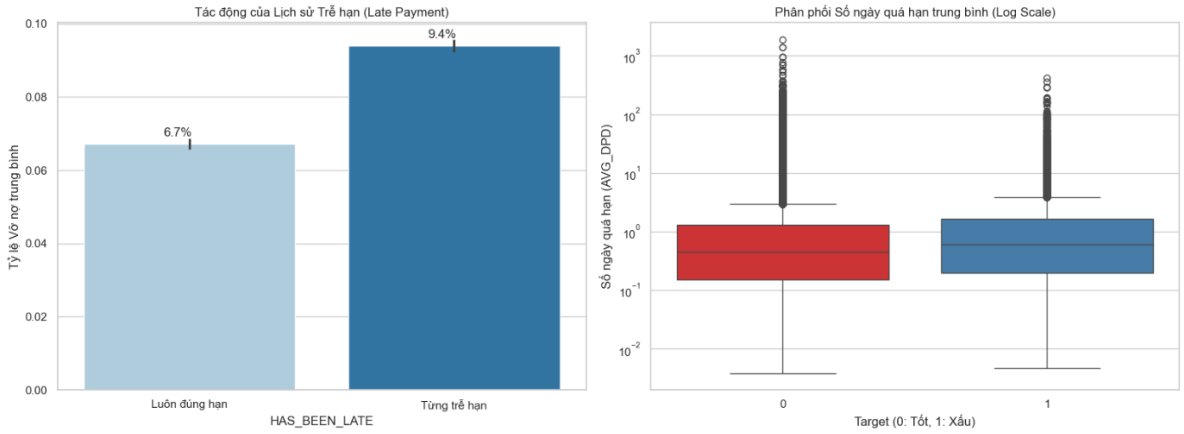
3.3.6 Khám phá lịch sử tín dụng



Hình 3.7: Biểu đồ khám phá về lịch sử khoản vay

- Bảng thống kê trung bình cho thấy số khoản vay tại các tổ chức tín dụng khác (BUREAU_COUNT) gần như không khác biệt giữa hai nhóm khách hàng (nhóm trả tốt: 4.77, nhóm nợ xấu: 4.61), trong khi số lần bị từ chối(PREV_REFUSED_COUNT) lại chênh lệch rõ rệt. Nhóm nợ xấu trung bình từng bị từ chối 1.18 lần, cao hơn 1.5 lần so với nhóm trả tốt (0.76 lần), cho thấy lịch sử từ chối là chỉ báo quan trọng.
- Biểu đồ Boxplot xác nhận rằng số lượng khoản vay tại các tổ chức tín dụng khác không phân hóa khách hàng tốt/xấu, với hình dạng, vị trí trung vị và độ rộng của hai hộp gần như tương đồng. Ngược lại, biểu đồ cột tỷ lệ vỡ nợ theo số lần bị từ chối thể hiện xu hướng tuyến tính dương rõ rệt: tỷ lệ vỡ nợ tăng từ 7.0% (0 lần bị từ chối) lên 12.4% (5 lần bị từ chối). Kết quả này cho thấy "lịch sử bị từ chối" là yếu tố dự báo rủi ro rất mạnh.

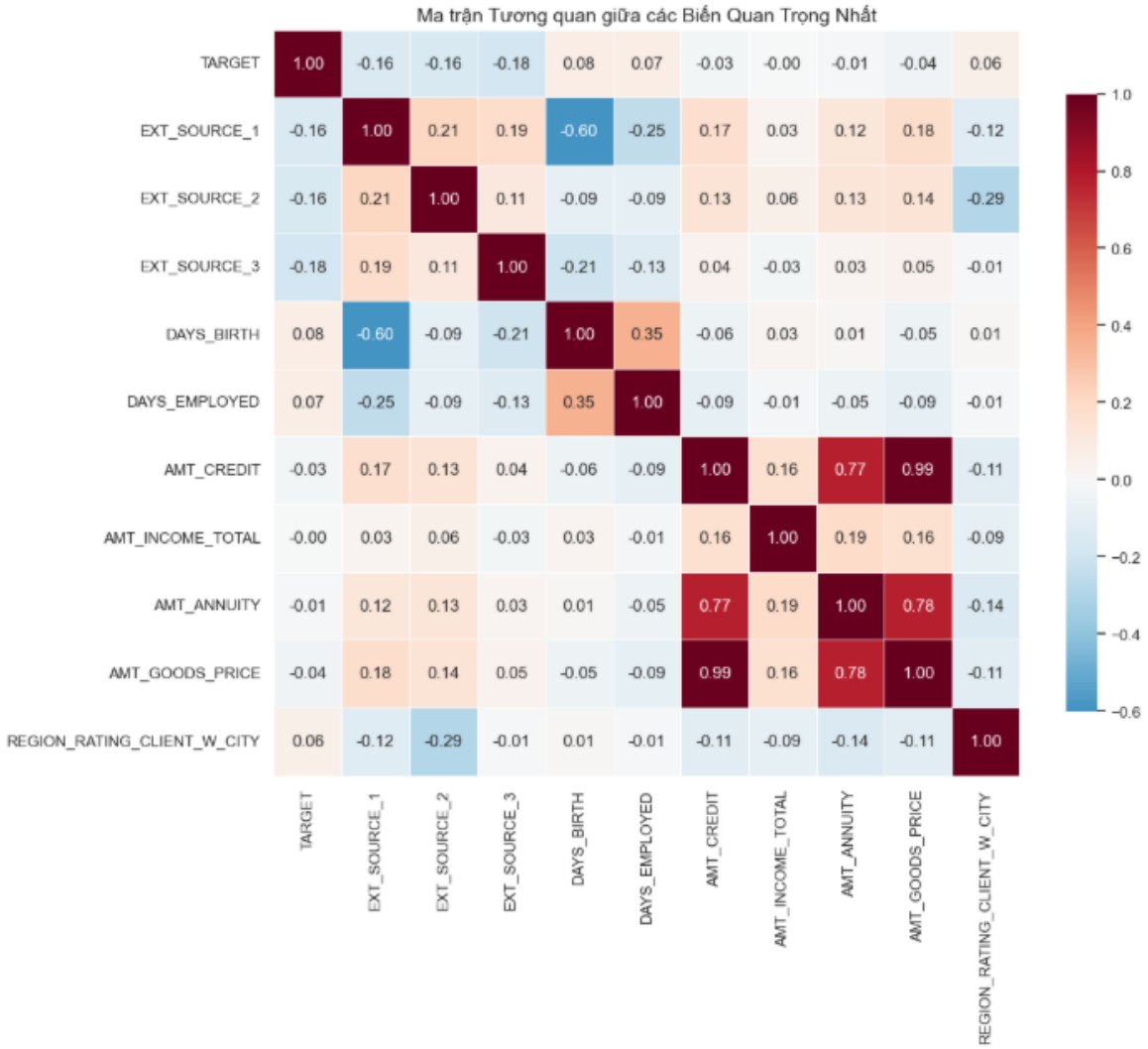
3.3.7 Khám phá hành vi thanh toán



Hình 3.8: Biểu đồ khám phá về hành vi thanh toán

- Bảng thống kê trung bình cho thấy nhóm nợ xấu (Target = 1) có thói quen trễ hạn rõ rệt hơn nhóm trả tốt: tỷ lệ trễ hạn trung bình đạt 9.9% so với 6.9% của nhóm trả tốt. Không chỉ trễ nhiều lần hơn, nhóm nợ xấu còn trễ lâu hơn (1.28 ngày so với 0.98 ngày) và trả thiếu nhiều hơn (trung bình 741 đơn vị so với 538 đơn vị), phản ánh mức độ nghiêm trọng cao hơn.
- Biểu đồ cột minh họa tác động của lịch sử trễ hạn: khách hàng từng trễ hạn có tỷ lệ vỡ nợ 9.4%, cao hơn đáng kể so với nhóm luôn trả đúng hạn (6.7%). Điều này nhấn mạnh rằng thói quen trả nợ trong quá khứ là chỉ báo mạnh mẽ cho rủi ro hiện tại.
- Biểu đồ Boxplot (số ngày quá hạn, Log Scale) cho thấy nhóm nợ xấu có phân phối dịch chuyển lên cao hơn, với đường trung vị nằm trên nhóm trả tốt. Kết quả này xác nhận rằng khách hàng rủi ro thường kéo dài thời gian quá hạn hơn, trong khi khách hàng trả tốt nếu trễ cũng nhanh chóng hoàn thành nghĩa vụ.

3.3.8 Phân tích tương quan



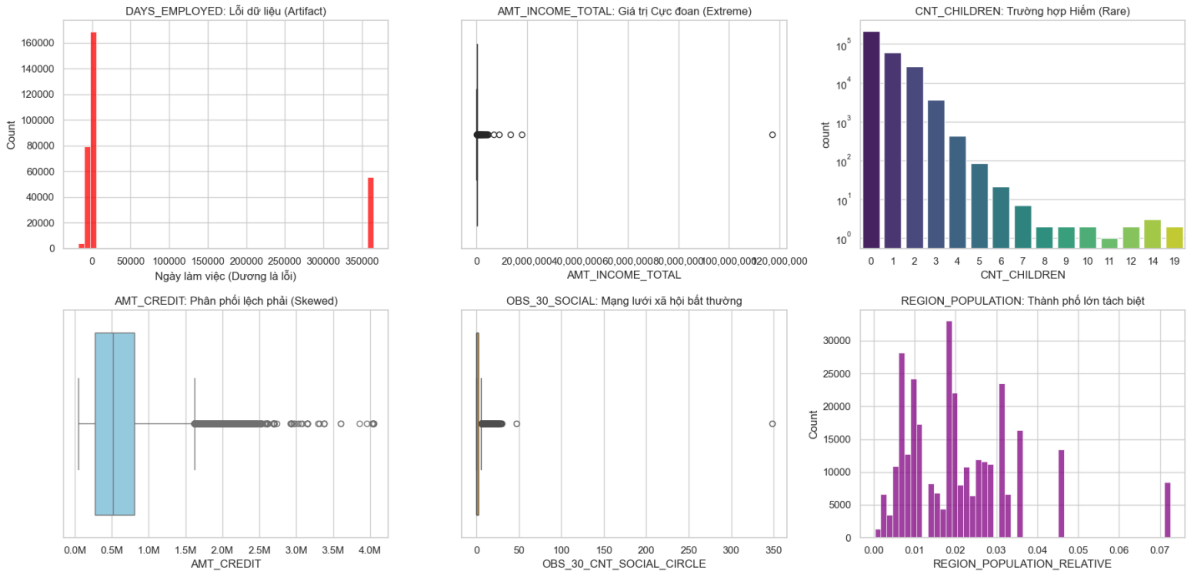
Hình 3.9: Heatmap giữa các biến quan trọng

- Các biến external (EXT_SOURCE_1, 2, 3) có tương quan nghịch mạnh nhất với rủi ro, cho thấy điểm tín dụng bên ngoài càng cao thì khả năng vỡ nợ càng thấp. Ngược lại, DAYS_BIRTH và một số chỉ số khu vực có tương quan dương, nhấn mạnh rằng khách hàng trẻ tuổi hoặc sống ở khu vực điểm thấp có rủi ro cao hơn.
- Bất ngờ là thu nhập tổng (AMT_INCOME_TOTAL) và số tiền vay (AMT_CREDIT) hầu như không liên quan tuyến tính đến rủi ro, cho thấy hành vi trả nợ quan trọng hơn số lượng thu nhập hay khoản vay.
- Cảnh báo đa cộng tuyến: AMT_CREDIT và AMT_GOODS_PRICE gần như trùng nhau (0.99), nên cần loại bớt một biến khi sử dụng mô hình tuyến tính. DAYS_BIRTH và EXT_SOURCE_1 tương quan -0.60, phản ánh rằng điểm external phụ thuộc đáng kể vào độ tuổi khách hàng.

3.3.9 Phát hiện giá trị ngoại lai

	Feature	Min	Median	Max	Outliers Count	Outliers (%)	Detection Rule
0	DAYS_EMPLOYED	-17912.000000	-1213.000000	365243.000000	55374	18.010000	Giá trị == 365243
1	AMT_INCOME_TOTAL	25650.000000	147150.000000	117000000.000000	14035	4.560000	> 337,500.00
2	CNT_CHILDREN	0.000000	0.000000	19.000000	4272	1.390000	> 2.50
3	AMT_CREDIT	45000.000000	513531.000000	4050000.000000	6562	2.130000	> 1,616,625.00
4	AMT_ANNUITY	1615.500000	24903.000000	258025.500000	7504	2.440000	> 61,704.00
5	AMT_GOODS_PRICE	40500.000000	450000.000000	4050000.000000	14728	4.790000	> 1,341,000.00
6	OBS_30_CNT_SOCIAL_CIRCLE	0.000000	0.000000	348.000000	19971	6.520000	> 5.00
7	REGION_POPULATION_RELATIVE	0.000290	0.018850	0.072508	8412	2.740000	> 0.06

Hình 3.10: Tổng hợp tình trạng ngoại lai



Hình 3.11: Boxplot và biểu đồ cột khám phá về giá trị ngoại lai

- Nhóm Lỗi dữ liệu: Trong nhóm này, DAYS_EMPLOYED có 55.374 dòng (18,01%) mang giá trị 365243, biểu đồ Histogram cho thấy cột đỏ tách biệt hoàn toàn ở phía dương, trong khi dữ liệu thực tế nằm ở phía âm.
- Nhóm Giá trị Cực đoan: Các biến như AMT_INCOME_TOTAL có giá trị lớn nhất 117 triệu trong khi Median chỉ 147k, và OBS_30_CNT_SOCIAL_CIRCLE có giá trị Max là 348, rất vô lý. Giải pháp: xóa dòng Max của AMT_INCOME_TOTAL (vì chỉ có 1 người), với các giá trị cao khác dùng Winsorization hoặc chặn trên; với OBS_30_CNT_SOCIAL_CIRCLE, chặn trên ở mức hợp lý.
- Nhóm Phân phối: Biến AMT_CREDIT và AMT_ANNUITY có nhiều outlier bên phải biểu đồ, Max AMT_CREDIT lên tới 4 triệu. Đây là dữ liệu hợp lệ, phản ánh khách hàng giàu và vay nhiều. Giải pháp: không xóa, sử dụng biến đổi Logarit (np.log1p) để làm mềm dữ liệu trước khi đưa vào mô hình.
- Nhóm Đặc thù: Biến REGION_POPULATION_RELATIVE có một cột cao tách biệt ở giá trị 0,0725, phản ánh khách hàng sống ở thủ đô/thành phố lớn; đây là thông

tin dự báo giá trị, nên giữ nguyên. Biến CNT_CHILDREN có giá trị cực cao 19 con, giải pháp là gom nhóm thành 0,1,2,3,4,5+ (tất cả >5 gộp vào nhóm này).

Chương 4

Tiền xử lý dữ liệu

Quy trình tiền xử lý dữ liệu được thiết kế nhằm mục đích làm sạch, chuẩn hóa và tối ưu hóa nguồn dữ liệu thô từ nhiều bảng khác nhau, tạo tiền đề vững chắc cho việc xây dựng các mô hình học máy. Dựa trên mã nguồn thực tế, em đã chia quy trình thành 4 giai đoạn chính:

4.1 Xử lý Sơ bộ và Làm sạch Dữ liệu

Trước khi tích hợp dữ liệu, em tiến hành xử lý các giá trị dị thường và lỗi định dạng trên bảng chính (`application_train/test`) và bảng lịch sử nộp đơn (`previous_application`).

Xử lý định dạng và lỗi dữ liệu:

- **Biến `DAYS_EMPLOYED`:** Phát hiện giá trị bất thường `365243` (tương đương 1000 năm, thường ám chỉ người đã nghỉ hưu hoặc lỗi hệ thống). Thay thế giá trị này bằng `NaN` để tránh làm lệch phân phối dữ liệu, đồng thời tạo thêm biến cờ `DAYS_EMPLOYED_ANOM` để giữ lại thông tin về trạng thái bất thường này.
- **Biến `CODE_GENDER`:** Loại bỏ hoặc gán lại các giá trị 'XNA' (không xác định) để đảm bảo tính nhất quán.

Xử lý lỗi toán học và định dạng

- **Xử lý giá trị vô cực:** Các phép toán tạo đặc trưng mới (feature engineering) như phép chia có thể sinh ra giá trị vô cực (`np.inf`, `-np.inf`) nên thay thế toàn bộ các giá trị này bằng `NaN` ở bước cuối cùng.
- **Chuẩn hóa tên cột:** Để đảm bảo tương thích với thư viện LightGBM và XGBoost, sử dụng biểu thức chính quy (Regex) để loại bỏ các ký tự đặc biệt (khoảng trắng, dấu ngoặc, dấu phẩy) trong tên cột, thay thế bằng dấu gạch dưới '_'.

Loại bỏ đặc trưng rác:

- Thực hiện loại bỏ các cột có phương sai bằng 0 – tức các cột chỉ chứa một giá trị duy nhất cho toàn bộ tập dữ liệu.

4.2 Biến đổi Đặc trưng và Gom nhóm Dữ liệu

4.2.1 Bảng chính (Application Train/Test)

a. Tạo 6 đặc trưng tài chính

Dựa trên bảng chính, tính được các chỉ số tài chính phản ánh khả năng trả nợ:

1. Tỷ lệ tín dụng trên thu nhập:

$$\texttt{NEW\_CREDIT\_TO\_INCOME\_RATIO} = \frac{\texttt{AMT\_CREDIT}}{\texttt{AMT\_INCOME\_TOTAL}}.$$

   Ý nghĩa: Khách hàng vay gấp bao nhiêu lần thu nhập hàng năm

2. Tỷ lệ gánh nặng trả nợ:

$$\texttt{NEW\_ANNUITY\_TO\_INCOME\_RATIO} = \frac{\texttt{AMT\_ANNUITY}}{\texttt{AMT\_INCOME\_TOTAL}}.$$

   Ý nghĩa: Số tiền phải trả hàng tháng chiếm bao nhiêu % lương

3. Tỷ lệ hàng hóa trên tín dụng:

$$\texttt{NEW\_GOODS\_TO\_CREDIT\_RATIO} = \frac{\texttt{AMT\_GOODS\_PRICE}}{\texttt{AMT\_CREDIT}}$$

   Ý nghĩa: Nếu < 1, khách hàng vay nhiều hơn giá trị món hàng → Có thể vay thêm tiền mặt

4. Số tiền trả trước:

$$\texttt{NEW\_DOWN\_PAYMENT} = \texttt{AMT\_GOODS\_PRICE-AMT\_CREDIT}$$

   Ý nghĩa : Ước tính số tiền khách hàng đã trả trước (nếu số dương) hoặc vay lố (nếu số âm).

5. Thời gian trả nợ giả định:

$$\texttt{NEW\_CREDIT\_TO\_ANNUITY\_RATIO} = \frac{\texttt{AMT\_CREDIT}}{\texttt{AMT\_ANNUITY}}$$

   Ý nghĩa: Biến này ước tính khoảng thời gian (số tháng/kỳ) cần thiết để trả hết khoản vay dựa trên mức trả góp hiện tại.

6. Tỷ lệ thâm niên công tác:

$$\texttt{NEW\_EMPLOYED\_TO\_BIRTH\_RATIO} = \frac{\texttt{DAYS\_EMPLPOYED}}{\texttt{DAYS\_BIRTHDAY}}$$

	NEW_DOWN_PAYMENT	NEW_GOODS_TO_CREDIT_RATIO	NEW_CREDIT_TO_INCOME_RATIO	NEW_ANNUITY_TO_INCOME_RATIO	NEW_CREDIT_TO_ANNUITY_RATIO	NEW_EMPLOYED_TO_BIRTH_RATIO
0	-55597.5	0.863262	2.007889	0.121978	16.461104	0.067329
1	-164002.5	0.873211	4.790750	0.132217	36.234085	0.070862
2	0.0	1.000000	2.000000	0.100000	20.000000	0.011814
3	-15682.5	0.949845	2.316167	0.219900	10.532818	0.159905
4	0.0	1.000000	4.222222	0.179963	23.461618	0.152418

Hình 4.1: 5 dòng đầu dữ liệu về 6 đặc trưng tài chính được tạo

b. Mã hóa One-Hot (One-Hot Encoding):

Các biến phân loại được tách thành nhiều biến nhị phân (0/1), gây ra sự gia tăng lớn về số cột:

- `ORGANIZATION_TYPE` (Loại hình tổ chức): Tách thành 58 cột.
- `OCCUPATION_TYPE` (Nghề nghiệp): Tách thành 18 cột.
- `NAME_INCOME_TYPE` (Nguồn thu nhập): Tách thành 8 cột.
- `NAME_EDUCATION_TYPE` (Học vấn): Tách thành 5 cột.
- `NAME_FAMILY_STATUS` (Gia đình): Tách thành 6 cột.
- `NAME_HOUSING_TYPE` (Nhà ở): Tách thành 6 cột.
- `NAME_TYPE_SUITE` (Người đi cùng khi nộp đơn): Tách thành 7 cột.
- `NAME_CONTRACT_TYPE` (Loại hợp đồng: Cash/Revolving): Tách thành 2 cột.
- `CODE_GENDER` (Giới tính: M/F/XNA): Tách thành 3 cột.
- `FLAG_OWN_CAR` (Sở hữu xe): Tách thành 2 cột.
- `FLAG_OWN_REALTY` (Sở hữu nhà): Tách thành 2 cột.
- `WEEKDAY_APPR_PROCESS_START` (Ngày nộp đơn trong tuần): Tách thành 7 cột.
- `FONDKAPREMONT_MODE`: Tách thành 4 cột.
- `HOUSETYPE_MODE`: Tách thành 3 cột.
- `WALLSMATERIAL_MODE` (Vật liệu tường): Tách thành 7 cột.
- `EMERGENCYSTATE_MODE` (Tình trạng khẩn cấp): Tách thành 2 cột.

4.2.2 Bảng bureau & bureau_balance: Lịch sử tín dụng bên ngoài

Dữ liệu từ CIC (Credit Bureau) được xử lý qua 3 tầng để nắm bắt toàn diện lịch sử tín dụng:

a. Xử lý chi tiết hàng tháng (bureau_balance):

- Biến `STATUS` (Trạng thái nợ: 0, 1, 2, 3, 4, 5 C, X) được One-Hot Encoding tách thành 8 cột.
- Gom nhóm theo `SK_ID_BUREAU` và tính trung bình (Mean) tạo thành đặc trưng mới (ghép vào bảng Bureau) về tỷ lệ xuất hiện của từng trạng thái. Ví dụ: `STATUS_0_MEAN`: Tỷ lệ số tháng trả nợ đúng hạn; `STATUS_1_MEAN`: Tỷ lệ số tháng bị trễ hạn mức 1; `STATUS_C_MEAN`: Tỷ lệ thời gian khoản vay đã đóng;...

b. Gom nhóm tổng quát (bureau):

Thực hiện gom nhóm theo `SK_ID_CURR` cho toàn bộ các khoản vay. Các đặc trưng số học được sinh ra chính xác theo cấu hình từ điển **num_aggregations**:

- `DAYS_CREDIT` (Số ngày từ khi vay): Tính Min, Max, Mean, Var.
- `DAYS_CREDIT_ENDDATE` (Ngày kết thúc): Tính Min, Max, Mean.

- DAYS_CREDIT_UPDATE (Ngày cập nhật gần nhất): Tính Mean.
- AMT_CREDIT_MAX_OVERDUE (Nợ quá hạn cao nhất): Tính Mean.
- AMT_CREDIT_SUM (Tổng hạn mức): Tính Max, Mean, Sum.
- AMT_CREDIT_SUM_DEBT (Dư nợ hiện tại): Tính Max, Mean, Sum.
- AMT_CREDIT_SUM_OVERDUE (Tiền quá hạn): Tính Mean.
- AMT_CREDIT_SUM_LIMIT (Hạn mức thẻ tín dụng): Tính Mean, Sum.
- AMT_ANNUITY (Số tiền trả hàng năm): Tính Max, Mean.
- CREDIT_DAY_OVERDUE (Số ngày quá hạn): Tính Max, Mean.
- CNT_CREDIT_PROLONG (Số lần gia hạn): Tính Sum.
- Tính giá trị trung bình (Mean) cho tất cả các cột One-Hot được sinh ra từ CREDIT_ACTIVE, CREDIT_CURRENCY, CREDIT_TYPE và các cột STATUS_MEAN từ mục a.

→ Tổng số lượng đặc trưng số học nhóm BURO_ sinh ra là 23 đặc trưng và tổng số cột tỷ lệ phân loại sinh ra từ việc One-Hot Encoding là 31 đặc trưng.

c. Gom nhóm theo trạng thái

Để phân biệt rủi ro hiện tại và quá khứ, tách dữ liệu và tính toán riêng:

- Nhóm Khoản vay đang hoạt động (Active Loans).
  Điều kiện: CREDIT_ACTIVE_Active == 1.
  → Sinh ra 6 đặc trưng cụ thể:
  1. ACTIVE_DAYS_CREDIT_MEAN: Trung bình số ngày vay.
  2. ACTIVE_DAYS_CREDIT_VAR: Phương sai số ngày vay.
  3. ACTIVE_AMT_CREDIT_SUM_SUM: Tổng hạn mức tín dụng đang có.
  4. ACTIVE_AMT_CREDIT_SUM_DEBT_SUM: Tổng dư nợ thực tế đang nợ.
  5. ACTIVE_AMT_CREDIT_SUM_OVERDUE_MEAN: Trung bình nợ quá hạn hiện tại.
  6. ACTIVE_CNT_CREDIT_PROLONG_SUM: Tổng số lần gia hạn nợ.
- Nhóm Khoản vay đã tất toán (Closed Loans).
  Điều kiện: CREDIT_ACTIVE_Closed == 1.
  → Sinh ra 2 đặc trưng cụ thể:
  1. CLOSED_AMT_CREDIT_SUM_SUM: Tổng số tiền đã từng vay và trả xong trong quá khứ.
  2. CLOSED_DAYS_CREDIT_VAR: Độ biến động thời gian của các khoản vay cũ.

```
================================================================================
2. LỊCH SỬ TÍN DỤNG (BUREAU) | Tổng số cột: 62
================================================================================
```

	BURO_DAYS_CREDIT_MIN	BURO_DAYS_CREDIT_MAX	BURO_DAYS_CREDIT_MEAN	BURO_DAYS_CREDIT_VAR	BURO_DAYS_CREDIT_ENDDATE_MIN	BURO_DAYS_CREDIT_ENDDATE_MAX
0	-1437.0	-103.0	-874.00	186150.000000	-1072.0	780.0
1	-2586.0	-606.0	-1400.75	827783.583333	-2434.0	1216.0
2	-1326.0	-408.0	-867.00	421362.000000	-595.0	-382.0
3	NaN	NaN	NaN	NaN	NaN	NaN
4	-1149.0	-1149.0	-1149.00	NaN	-783.0	-783.0

5 rows × 62 columns

Hình 4.2: Dữ liệu 5 dòng đầu bảng bureau & bureau_balance sau xử lý để nối vào bảng chính

4.2.3 Tích hợp Lịch sử hồ sơ vay (Previous Application)

Dữ liệu về các lần nộp đơn trước đây tại Home Credit được xử lý như sau:
a. Tạo đặc trưng mới:

- Tỷ lệ được duyệt vay:

$$\texttt{APP\_CREDIT\_PERC} = \frac{\texttt{AMT\_APPLICATION}}{\texttt{AMT\_CREDIT}}$$

  Ý nghĩa: đánh giá mức độ uy tín của hồ sơ.

b. Chiến lược gom nhóm:

- Thống kê chung (cột tiền tốPREV_):
  - Tính Min, Max, Mean cho các biến: AMT_ANNUITY, AMT_APPLICATION, AMT_CREDIT, AMT_DOWN_PAYMENT, AMT_GOODS_PRICE, RATE_DOWN_PAYMENT, HOUR_APPR_PROCESS_START, DAYS_DECISION.
  - Tính Min, Max, Mean, Var cho biến APP_CREDIT_PERC (Tỷ lệ duyệt).
  - Tính Mean, Sum cho biến CNT_PAYMENT (Kỳ hạn vay).
  - Tính giá trị trung bình (Mean) cho tất cả các cột One-Hot được sinh ra từ NAME_CONTRACT_TYPE, WEEKDAY_APPR_PROCESS_START, FLAG_LAST_APPL_PER_CONTRACT, NAME_CASH_LOAN_PURPOSE, NAME_CONTRACT_STATUS, NAME_PAYMENT_TYPE, CODE_REJECT_REASON, NAME_TYPE_SUITE, NAME_CLIENT_TYPE, NAME_GOODS_CATEGORY, NAME_PORTFOLIO, NAME_PRODUCT_TYPE, CHANNEL_TYPE, NAME_SELLER_INDUSTRY, NAME_YIELD_GROUP, PRODUCT_COMBINATION.[1]

  → Tổng số lượng đặc trưng số học nhóm PREV_ sinh ra là: 29 đặc trưng và sau One-hot các cột nêu trên tách thêm được 145 đặc trưng.

- Thống kê theo kết quả thẩm định:
  - Nhóm Approved (Được duyệt): Tính toán riêng để đánh giá hành vi trả nợ của các hợp đồng thành công. Tiền tố APPROVED_.
    Điều kiện: NAME_CONTRACT_STATUS_Approved == 1.
    → Sinh ra 7 đặc trưng cụ thể:
    1. APPROVED_AMT_ANNUITY_MEAN: Trung bình số tiền trả góp của các hợp đồng thành công
    2. APPROVED_AMT_CREDIT_MEAN: Trung bình hạn mức được cấp.
    3. APPROVED_AMT_CREDIT_MAX: Hạn mức cao nhất từng được cấp.
    4. APPROVED_DAYS_DECISION_MEAN: Trung bình thời gian nộp đơn thành công.
    5. APPROVED_DAYS_DECISION_MAX: Lần được duyệt gần nhất.
    6. APPROVED_CNT_PAYMENT_MEAN: Kỳ hạn vay trung bình.
    7. APPROVED_CNT_PAYMENT_SUM: Tổng số kỳ hạn đã vay.
  - Nhóm Refused (Bị từ chối): Tính toán riêng để đếm số lần bị từ chối và phân tích các khoản vay thường bị từ chối. Tiền tố REFUSED_.
    Điều kiện: NAME_CONTRACT_STATUS_Refused == 1.
    → Sinh ra 4 đặc trưng cụ thể (có xử lý điều kiện cột cnt_payment tồn tại hay không):

1. REFUSED_AMT_APPLICATION_MEAN: Trung bình số tiền xin vay bị từ chối.
2. REFUSED_AMT_APPLICATION_MAX: Số tiền lớn nhất từng bị từ chối.
3. REFUSED_DAYS_DECISION_MEAN: Trung bình thời gian bị từ chối.
4. REFUSED_DAYS_DECISION_MAX: Lần bị từ chối gần nhất.

```
========================================================================
3. LỊCH SỬ NỘP ĐƠN (PREV APP) | Tổng số cột: 186
========================================================================
```

	PREV_AMT_ANNUITY_MIN	PREV_AMT_ANNUITY_MAX	PREV_AMT_ANNUITY_MEAN	PREV_AMT_APPLICATION_MIN	PREV_AMT_APPLICATION_MAX	PREV_AMT_APPLICATION_MEAN
0	9251.775	9251.775	9251.775	179055.0	179055.0	179055.00
1	6737.310	98356.995	56553.990	68809.5	900000.0	435436.50
2	5357.250	5357.250	5357.250	24282.0	24282.0	24282.00
3	2482.920	39954.510	23651.175	0.0	688500.0	272203.26
4	1834.290	22678.785	12278.805	17176.5	247500.0	150530.25

5 rows × 186 columns

Hình 4.3: Dữ liệu 5 dòng đầu bảng Previous sau xử lý để ghép vào bảng chính

4.2.4 Phân tích Hành vi trả nợ (Installments Payments)

Đây là bảng dữ liệu quan trọng nhất để phát hiện nợ xấu. Quy trình gồm 2 bước:

1. Tạo ra 5 biến mới phản ánh mức độ vi phạm:
   - Tỷ lệ số tiền thực trả so với phải trả:
   $$\texttt{PAYMENT\_PERC} = \frac{\texttt{AMT\_PAYMENT}}{\texttt{AMT\_INSTALLMENT}}.$$
   Nếu tỷ lệ <1 là trả thiếu.
   - Số tiền trả thiếu :
   $$\texttt{PAYMENT\_DIFF} = \texttt{AMT\_INSTALLMENT} - \texttt{AMT\_PAYMENT}.$$
   - Số ngày chênh lệch:
   $$\texttt{DAYS\_DIFF} = \texttt{DAYS\_ENTRY\_PAYMENT-DAYS\_INSTALMENT}.$$
   - Số ngày quá hạn (Days Past Due):
   $$\texttt{DPD} = \max(0, \texttt{DAYS\_DIFF}).$$
   - Số ngày trả trước hạn (Days Before Due):
   $$\texttt{DBD} = \max(0, \texttt{DAYS\_DIFF}).$$
2. Aggregation (Gom nhóm theo khách hàng): Áp dụng các hàm thống kê lên 4 biến trên để sinh ra 24 cột chi tiết:
   - PAYMENT_PERC (Tỷ lệ trả): Tính Max, Mean, Var. → Sinh ra 3 cột.
   - PAYMENT_DIFF (Trả thiếu): Tính Max, Mean, Sum, Var: Sinh ra 4 cột.
   - DPD (Quá hạn): Tính Max, Mean, Sum. → Sinh ra 3 cột.
   - DBD (Trả sớm): Tính Max, Mean, Sum. → Sinh ra 3 cột.
   - AMT_PAYMENT (Số tiền trả): Tính Min, Max, Mean, Sum. → Sinh ra 4 cột.

- AMT_INSTALMENT (Số tiền phải trả): Tính Max, Mean, Sum. → Sinh ra 3 cột.
- DAYS_ENTRY_PAYMENT: Tính Max, Mean, Min. → Sinh ra 3 cột.
- INSTAL_COUNT_LATE: Tổng số lần trả chậm. → Sinh ra 1 cột.

```
================================================================================
4. LỊCH SỬ TRẢ GÓP (INSTALLMENTS) | Tổng số cột: 24
================================================================================
```

	INSTAL_PAYMENT_PERC_MAX	INSTAL_PAYMENT_PERC_MEAN	INSTAL_PAYMENT_PERC_VAR	INSTAL_PAYMENT_DIFF_MAX	INSTAL_PAYMENT_DIFF_MEAN	INSTAL_PAYMENT_DIFF_VAR
0	1.0	1.000000	0.000000	0.000	0.000000	0.000000e+00
1	1.0	1.000000	0.000000	0.000	0.000000	0.000000e+00
2	1.0	1.000000	0.000000	0.000	0.000000	0.000000e+00
3	1.0	1.000000	0.000000	0.000	0.000000	0.000000e+00
4	1.0	0.954545	0.043995	22655.655	452.384318	8.084830e+06

Hình 4.4: Dữ liệu 5 dòng đầu bảng Instalments sau xử lý để nối vào bảng chính

4.2.5 Dữ liệu Vay tiền mặt tại điểm bán (Bảng POS_CASH)

Đây là dữ liệu về các khoản vay tiêu dùng hoặc vay tiền mặt tại các điểm bán hàng. Quy trình xử lý tập trung vào việc đánh giá mức độ tuân thủ hợp đồng thông qua các biến trạng thái và thời gian.

a. Xử lý biến phân loại

- Cột NAME_CONTRACT_STATUS (Active, Completed, Demand...) được mã hóa One-Hot.
- Tính trung bình (Mean) để xác định tỷ lệ trạng thái hợp đồng của khách hàng.

b. Gom nhóm thống kê:

Áp dụng các hàm thống kê lên các biến số học quan trọng:

- SK_DPD & SK_DPD_DEF (Số ngày quá hạn):
- Tính Max, Mean để xác định mức độ trễ hạn nghiêm trọng nhất và trung bình: Tính Min, Max, Mean cho biến CNT_INSTALMENT(Kỳ hạn vay); Tính Min, Max, Mean **CNT_INSTALMENT_FUTURE** (Số kỳ còn lại phải trả) biến này phản ánh gánh nặng nợ trong tương lai gần.

→ Sinh ra khoảng 19 cột đặc trưng.

```
--------------------------------------------------------------------------------------------------
5. VAY TIỀN MẶT TẠI QUẦY (POS CASH)
   Tổng số cột: 19
--------------------------------------------------------------------------------------------------
```

	POS_SK_DPD_MAX	POS_SK_DPD_MEAN	POS_SK_DPD_DEF_MAX	POS_SK_DPD_DEF_MEAN	POS_CNT_INSTALMENT_MIN	POS_CNT_INSTALMENT_MAX	POS_CNT_INSTALMENT_MEAN
0	0.0	0.0	0.0	0.0	24.0	24.0	24.000000
1	0.0	0.0	0.0	0.0	6.0	12.0	10.107143
2	0.0	0.0	0.0	0.0	3.0	4.0	3.750000
3	0.0	0.0	0.0	0.0	1.0	48.0	12.000000
4	0.0	0.0	0.0	0.0	10.0	24.0	15.333333

Hình 4.5: Dữ liệu 5 dòng đầu bảng POS_CASH sau xử lý

4.2.6 Tích hợp Lịch sử thẻ tín dụng (Bảng credit_card_balance)

a. Xử lý biến phân loại:

- `NAME_CONTRACT_STATUS` (Trạng thái hợp đồng): Thực hiện One-Hot Encoding cho 7 trạng thái (Active, Completed, Demand, Signed, Sent proposal, Refused, Approved), sau đó tính Mean cho từng cột để ra tỷ lệ. → Sinh ra 7 cột.
- `CC_COUNT` (Thâm niên): Đếm số lượng bản ghi bằng hàm Size. → Sinh ra 1 cột.

b. Gom nhóm thống kê:
Dựa trên cấu hình từ điển aggregations trong mã nguồn, sinh ra 45 cột số học từ các biến sau:

- `AMT_BALANCE` (Dư nợ hàng tháng): Tính Min, Max, Mean, Sum, Var.
- `AMT_CREDIT_LIMIT_ACTUAL` (Hạn mức thẻ): Tính Min, Max, Mean, Sum.
- `AMT_TOTAL_RECEIVABLE` (Tổng tiền phải thu): Tính Min, Max, Mean.
- `AMT_DRAWINGS_ATM_CURRENT` (Tiền rút tại ATM): Tính Min, Max, Mean, Sum.
- `AMT_DRAWINGS_CURRENT` (Tổng tiền chi tiêu): Tính Min, Max, Mean, Sum.
- `AMT_DRAWINGS_POS_CURRENT` (Tiền quẹt thẻ POS): Tính Min, Max, Mean, Sum.
- `CNT_DRAWINGS_ATM_CURRENT` (Số lần rút tiền): Tính Min, Max, Mean, Sum.
- `CNT_DRAWINGS_CURRENT` (Tổng số lần dùng thẻ): Tính Min, Max, Mean, Sum.
- `CNT_DRAWINGS_POS_CURRENT` (Số lần quẹt thẻ): Tính Mean.
- `AMT_INST_MIN_REGULARITY` (Trả tối thiểu): Tính Min, Max, Mean.
- `AMT_PAYMENT_TOTAL_CURRENT` (Tiền thực trả): Tính Min, Max, Mean.
- `SK_DPD` (Số ngày quá hạn): Tính Mean, Max, Sum.
- `SK_DPD_DEF` (Số ngày quá hạn mức độ cao): Tính Mean, Max, Sum.

```
==================================================================================================
6. THẺ TÍN DỤNG (CREDIT CARD)
   Tổng số cột: 53
==================================================================================================
```

	CC_AMT_BALANCE_MIN	CC_AMT_BALANCE_MAX	CC_AMT_BALANCE_MEAN	CC_AMT_BALANCE_SUM	CC_AMT_BALANCE_VAR	CC_AMT_CREDIT_LIMIT_ACTUAL_MIN	CC_AMT_CREDIT_LIMIT_ACTUAL_MAX
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	0.0	NaN	270000.0	270000.0
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Hình 4.6: Dữ liệu 5 dòng đầu bảng Credit_card sau xử lý

4.3 Chuẩn hóa và Mã hóa Dữ liệu Cuối cùng

Sau quá trình biến đổi và gom nhóm, bộ dữ liệu đã mở rộng lên 473 đặc trưng. Trước khi đưa vào huấn luyện cần thực hiện bước rà soát kỹ thuật cuối cùng để đảm bảo tính tương thích tuyệt đối với thuật toán.

a. Kiểm tra lại và Chuẩn hóa kỹ thuật sau biến đổi:

- **Xử lý giá trị vô cực** (`inf`): Các phép toán tỷ lệ (ví dụ: chia cho số tiền bằng 0) trong quá trình Feature Engineering có thể sinh ra giá trị vô cực. Thực hiện quét lại toàn bộ dữ liệu và thay thế `np.inf` bằng `NaN` để tránh gây lỗi gradient khi huấn luyện.
- **Chuẩn hóa tên đặc trưng**: Quá trình gom nhóm tạo ra các tên cột phức tạp hoặc chứa ký tự đặc biệt. Sử dụng **Regex** để loại bỏ hoàn toàn khoảng trắng và ký tự lạ (như **(**, **)**, **,**), thay thế bằng dấu gạch dưới _. Việc xử lý rất quan trọng vì thư viện LightGBM/XGBoost không chấp nhận tên cột chứa ký tự đặc biệt (định dạng JSON).
- **Loại bỏ đặc trưng rác**: Sau khi One-Hot Encoding và gom nhóm, có thể xuất hiện các cột chỉ chứa một giá trị duy nhất (do không có biến động trong tập dữ liệu). Các cột này được loại bỏ để giảm tải bộ nhớ.

b. Phân tách và Định dạng dữ liệu

Dữ liệu bảng tổng hợp được phân tách dựa trên trạng thái của biến mục tiêu:

- Tách tập Train/Test:
  - Tập Train: Bao gồm các bản ghi có giá trị TARGET (Dùng để huấn luyện và kiểm thử chéo).
  - Tập Test: Bao gồm các bản ghi có TARGET là NaN (Dùng để dự báo nộp kết quả).
- Tách biệt biến mục tiêu: Tách cột TARGET ra khỏi ma trận đặc trưng đầu vào (X) và lưu vào biến nhãn (y).
- Lưu trữ định danh: Tách riêng cột `SK_ID_CURR` để phục vụ cho việc truy xuất kết quả sau này.

c. Mã hóa biến phân loại còn lại (Label Encoding)

Đối với các biến dạng chuỗi còn sót lại hoặc các biến phân loại nhị phân chưa được One-Hot ở bước trước, áp dụng Label Encoding (chuyển đổi thành các số nguyên 0, 1, 2...). Quá trình này được thực hiện trên tập dữ liệu gộp (Train + Test) trước khi tách để đảm bảo tính nhất quán của bộ mã hóa.

4.4 Phân nhánh xử lý và Cân bằng Mẫu

4.4.1 Phân nhánh xử lý

Dữ liệu sau khi xử lý được tách thành hai luồng riêng biệt (`X_lgbm` và `X_rf`) để phù hợp với yêu cầu đầu vào của từng nhóm thuật toán:

Nhánh A: Xử lý cho XGBoost và LightGBM (`X_lgbm`).

- Giữ nguyên giá trị khuyết thiếu (`NaN`). Không thực hiện điền khuyết.
- Các thuật toán Gradient Boosting hiện đại (như XGBoost, LightGBM) có cơ chế nội tại (Sparsity-aware Split Finding) cho phép tự động học hướng đi tối ưu cho các giá trị bị thiếu trong quá trình xây dựng cây quyết định. Việc giữ nguyên NaN giúp bảo toàn thông tin về sự thiếu khuyết của dữ liệu, vốn có thể là một tín hiệu rủi ro.

Nhánh B: Xử lý cho Random Forest (`X_rf`).

- Thực hiện điền khuyết và sử dụng Giá trị trung vị (Median).
  - Thư viện Scikit-learn (cài đặt Random Forest) không hỗ trợ đầu vào chứa giá trị NaN, bắt buộc phải thay thế giá trị thiếu.
  - Lựa chọn Median thay vì Mean (Trung bình) vì dữ liệu tài chính (thu nhập, dư nợ...) thường chứa các giá trị ngoại lai rất lớn. Median bền vững hơn với nhiễu, giúp dữ liệu đầu vào của Random Forest ổn định hơn.

4.4.2 Xử lý mất cân bằng mẫu

Dữ liệu rủi ro tín dụng có đặc thù mất cân bằng nghiêm trọng (số lượng hồ sơ tốt áp đảo hồ sơ nợ xấu). Thay vì sử dụng kỹ thuật lấy mẫu (Sampling) làm thay đổi phân phối dữ liệu gốc thì em đã sử dụng phương pháp Cost-sensitive Learning:

- Tính toán trọng số: Thiết lập tham số `scale_pos_weight` dựa trên tỷ lệ thực tế của dữ liệu:

$$\texttt{scale\_pos\_weight} = \frac{\text{Số lượng mẫu đa số (Target = 0)}}{\text{Số lượng mẫu thiểu số (Target = 1)}}.$$

  $\rightarrow$ Tham số này được đưa trực tiếp vào hàm mất mát (Loss Function) của mô hình, giúp thuật toán xử lý khi dự báo sai các trường hợp nợ xấu, từ đó tăng cường độ nhạy (Recall) của mô hình.

4.4.3 Thiết lập quy trình đánh giá

Sử dụng **Stratified K-Fold Cross-Validation** (Kiểm chứng chéo phân tầng) với $K = 5$. Đảm bảo tỷ lệ nợ xấu (Target=1) trong mỗi tập kiểm thử (Fold) luôn được giữ nguyên tương đồng với tập dữ liệu gốc, giúp kết quả đánh giá ổn định và tin cậy hơn.

- Tập dữ liệu huấn luyện được chia thành 5 phần (folds) riêng biệt. Kỹ thuật Stratified đảm bảo tỷ lệ nợ xấu (Target=1) trong mỗi fold luôn xấp xỉ tỷ lệ của tập dữ liệu gốc ($\sim 8\%$), giúp tránh trường hợp một fold chỉ toàn hồ sơ tốt.
- Quá trình huấn luyện diễn ra 5 lần. Ở mỗi lần, mô hình sử dụng 4 phần (80%) để học và 1 phần (20%) còn lại để kiểm thử (Validation).
- Dự báo OOF (Out-of-Fold): Kết quả dự báo trên 5 tập kiểm thử rời rạc được ghép lại để tạo thành một bộ dự báo hoàn chỉnh cho toàn bộ tập dữ liệu. Đây là cơ sở tin cậy nhất để đánh giá hiệu suất mô hình.

Chương 5

Xây dựng mô hình

5.1 Chuẩn bị tập huấn luyện và kiểm tra

5.1.1 Phân chia dữ liệu

Sử dụng kỹ thuật Stratified K-Fold Cross-Validation (Kiểm chứng chéo phân tầng) với số lượng $K = 5(5 - Folds)$. Phương pháp này chia tập dữ liệu huấn luyện thành 5 phần bằng nhau, trong đó đảm bảo tỷ lệ nhãn mục tiêu (Target 0 và 1) trong mỗi phần là tương đồng với tập dữ liệu gốc. Điều này đặc biệt quan trọng đối với bài toán mất cân bằng dữ liệu như Home Credit.

5.1.2 Các chỉ số đánh giá

1. **Chỉ số đánh giá hiệu suất mô hình** [2]

   (a) **Ma trận nhầm lẫn**

   Để tính toán các chỉ số đánh giá, trước hết ta cần xác định các thành phần trong Ma trận nhầm lẫn (Confusion Matrix):

   - TP (True Positive): Số trường hợp dự báo đúng là rủi ro (Dương tính thật).
   - TN (True Negative): Số trường hợp dự báo đúng là an toàn (Âm tính thật).
   - FP (False Positive): Số trường hợp dự báo sai là rủi ro (Dương tính giả).
   - FN (False Negative): Số trường hợp dự báo sai là an toàn (Âm tính giả).

   (b) **Chỉ số độ chính xác (Accuracy)**

   Chỉ số Accuracy là thước đo cơ bản nhất, thể hiện tỷ lệ phần trăm các dự báo đúng trên tổng số các quan sát. Accuracy được xác định dựa theo công thức:

   $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

   Trong đó:

   - Trong đó: TP, TN, FP, FN là các giá trị thu được từ ma trận nhầm lẫn.
   - $TP + TN$ là tổng số lượng mẫu được mô hình dự đoán đúng.
   - $TP + TN + FP + FN$ là tổng số lượng mẫu trong tập dữ liệu.

   Tuy nhiên, trong bài toán rủi ro tín dụng với dữ liệu mất cân bằng, chỉ số này thường không phản ánh chính xác hiệu quả mô hình vì nó bị chi phối bởi lớp đa số (khách hàng tốt) vậy nên không được đưa vào để đánh giá mô hình.

(c) **Chỉ số Độ chính xác dự báo dương (Precision)**

Chỉ số Precision (còn gọi là Positive Predictive Value) đo lường tỷ lệ các mẫu thực sự thuộc lớp Dương trong tổng số các mẫu được mô hình dự báo là lớp Dương. Precision được xác định bởi công thức:

$$Precision = \frac{TP}{TP + FP}$$

Trong đó:

- TP là số lượng mẫu Dương tính thật.
- FP (Dương tính giả) là số lượng mẫu thực tế là Âm nhưng bị mô hình gán nhãn Dương.

Chỉ số này càng cao chứng tỏ mô hình có độ tin cậy cao khi đưa ra dự báo Dương tính, hạn chế việc báo động sai.

(d) **Chỉ số Độ nhạy (Recall)**

Chỉ số Recall (còn gọi là Sensitivity hoặc True Positive Rate) đo lường khả năng của mô hình trong việc phát hiện các mẫu thuộc lớp Dương. Chỉ số này thể hiện tỷ lệ các mẫu Dương được dự báo đúng trên tổng số mẫu Dương thực tế. Recall được tính bằng công thức:

$$Recall = \frac{TP}{TP + FN}$$

Trong đó:

- TP là số lượng mẫu Dương tính thật được phát hiện.
- FN (Âm tính giả) là số lượng mẫu thực tế là Dương nhưng bị mô hình bỏ sót (gán nhãn Âm).

Chỉ số Recall càng cao cho thấy mô hình có khả năng bao quát tốt và ít bỏ sót các đối tượng quan trọng cần phát hiện.

(e) **Chỉ số F1-Score**

Chỉ số F1-Score là trung bình điều hòa (Harmonic Mean) của Precision và Recall. Chỉ số này được sử dụng để đánh giá tổng hợp khi cần cân bằng giữa độ chính xác của dự báo và độ phủ của mô hình. F1-Score được tính theo công thức:

$$\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Trong đó:

- Giá trị F1-Score nằm trong khoảng $[0, 1]$.
- F1-Score đạt giá trị cao khi cả Precision và Recall đều cao. Chỉ số này đặc biệt hữu ích trong các bài toán mà sự phân bố giữa các lớp dữ liệu không đồng đều.

(f) **ROC-AUC**

ROC Curve (Receiver Operating Characteristic) là đồ thị biểu diễn mối quan hệ giữa Tỷ lệ Dương tính thật (TPR) và Tỷ lệ Dương tính giả (FPR) tại các ngưỡng phân loại khác nhau.

$$TPR = Recall = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Chỉ số AUC là diện tích nằm dưới đường cong ROC.

Ý nghĩa của chỉ số:

- AUC < 0.5: Hiệu suất kém hơn cả đoán ngẫu nhiên.
- $AUC = 0.5$: Mô hình không tốt hơn đoán ngẫu nhiên.
- $AUC = 1$: Mô hình có khả năng phân loại tốt, phân tách rõ ràng giữa hai lớp Dương và Âm.

2. **Chỉ số đánh giá hiệu quả nghiệp vụ** [10]

(a) **Hệ số Gini (Gini Coefficient)**

Hệ số Gini (Gini Coefficient) là thước đo phổ biến trong mô hình phân loại nhị phân, đặc biệt trong các bài toán chấm điểm tín dụng. Chỉ số này phản ánh mức độ phân tách giữa nhóm tốt và nhóm xấu của mô hình. Về bản chất, Gini được tính trực tiếp từ ROC–AUC theo công thức:

$$Gini = 2 \times AUC - 1$$

(b) **Chỉ số KS (Kolmogorov-Smirnov Statistic)**

Hệ số KS (Kolmogorov–Smirnov) là chỉ số đánh giá mức độ phân biệt của mô hình phân loại nhị phân bằng cách đo khoảng cách lớn nhất giữa hai phân phối tích lũy: phân phối dự đoán của nhóm dương (bad) và nhóm âm (good). Cụ thể, KS được tính bằng hiệu lớn nhất giữa TPR (tỷ lệ bắt đúng bad) và FPR (tỷ lệ nhầm good thành bad) tại một ngưỡng bất kỳ.

Giá trị KS càng cao cho thấy mô hình càng phân tách rõ ràng giữa hai nhóm; trong thực tế, KS từ 20–40% được xem là tốt trong mô hình tín dụng. Chỉ số này đặc biệt được ưa chuộng vì đơn giản, dễ giải thích và phản ánh trực tiếp mức độ tách biệt giữa hai nhóm khách hàng.

5.2 Huấn luyện các mô hình

Trong phần này,3 mô hình học máy khác nhau để thiết lập mức hiệu suất cơ sở (Baseline). Các mô hình được khởi tạo với bộ tham số dựa trên kinh nghiệm thực nghiệm, chưa qua các bước tinh chỉnh sâu, nhằm mục đích đánh giá sơ bộ khả năng học của từng thuật toán trên tập dữ liệu Home Credit.

5.2.1 Mô hình Random Forest

Mô hình Random Forest (Rừng ngẫu nhiên) được xây dựng dựa trên nguyên lý Bagging, kết hợp kết quả của nhiều cây quyết định độc lập để đưa ra dự báo cuối cùng.[7]

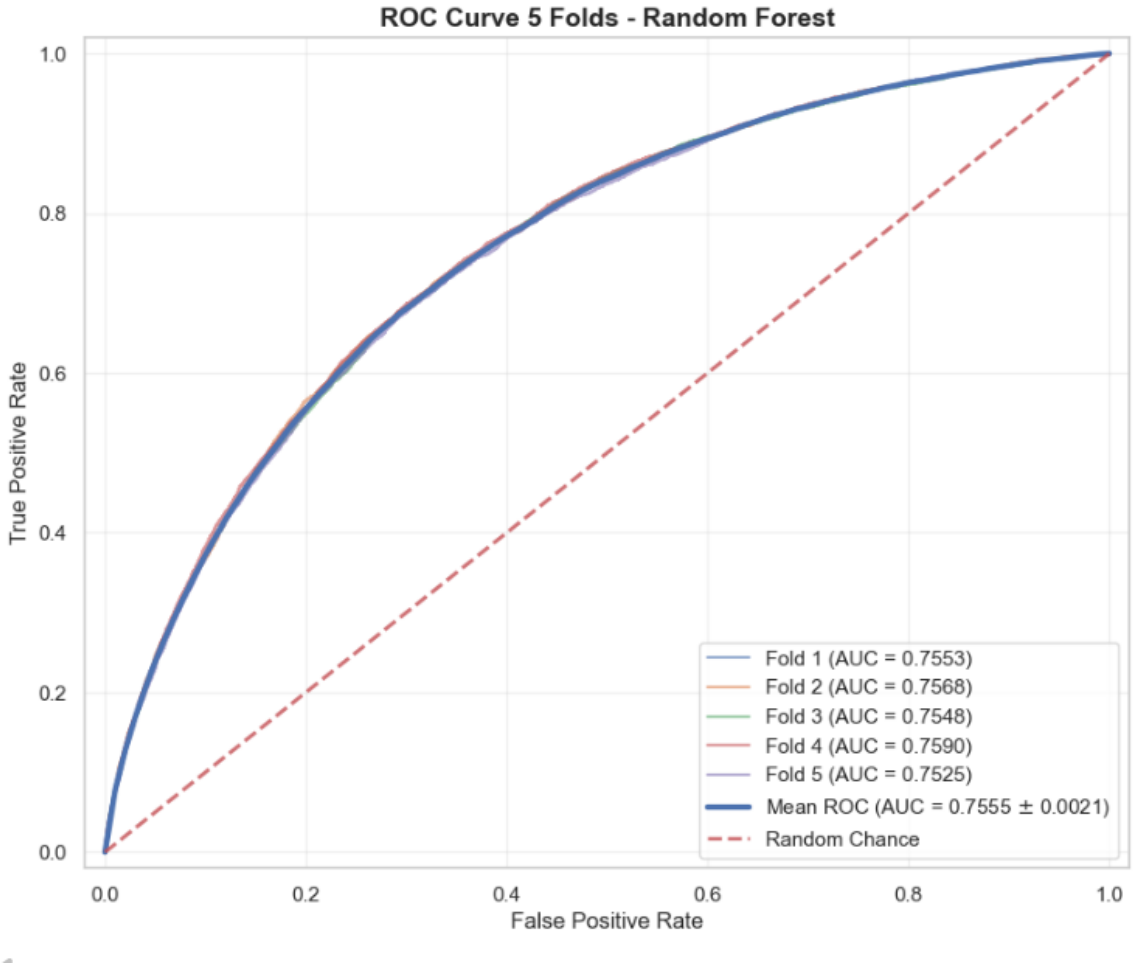
Do đặc thù của thư viện Scikit-learn, mô hình này không thể xử lý trực tiếp các giá trị khuyết thiếu (NaN). Vì vậy, mô hình được huấn luyện trên tập dữ liệu X_rf (đã được xử lý điền khuyết bằng trung vị ở chương trước). Cấu trúc huấn luyện được thiết lập như sau:

1. Mô hình khởi tạo 200 cây quyết định hoạt động song song.
2. Để kiểm soát kích thước mô hình và tránh hiện tượng tràn bộ nhớ (Out of Memory) với bộ dữ liệu lớn, độ sâu của cây được giới hạn và quy định số lượng mẫu tối thiểu tại mỗi nút lá.
3. Sử dụng cơ chế trọng số lớp (Class Weight) để tự động cân bằng sự chênh lệch giữa nhóm khách hàng nợ xấu và nợ tốt.

Các siêu tham số (Hyperparameters) thiết lập cho mô hình:

- **n_estimators**: 200 (Số lượng cây trong rừng).
- **max_depth**: 12 (Độ sâu tối đa của mỗi cây, giúp tránh Overfitting).
- **min_samples_leaf**: 30 (Số lượng mẫu tối thiểu tại mỗi nút lá).
- **class_weight**: 'balanced' (Tự động điều chỉnh trọng số dựa trên tần suất nhãn).
- **n_jobs**: -1 (Sử dụng toàn bộ nhân CPU để tăng tốc độ huấn luyện).

Kết quả thực nghiệm mô hình Random Forest Baseline:



Hình 5.1: Biểu đồ ROC Curve đánh giá mô hình RF

5.2.2 Mô hình XGBoost

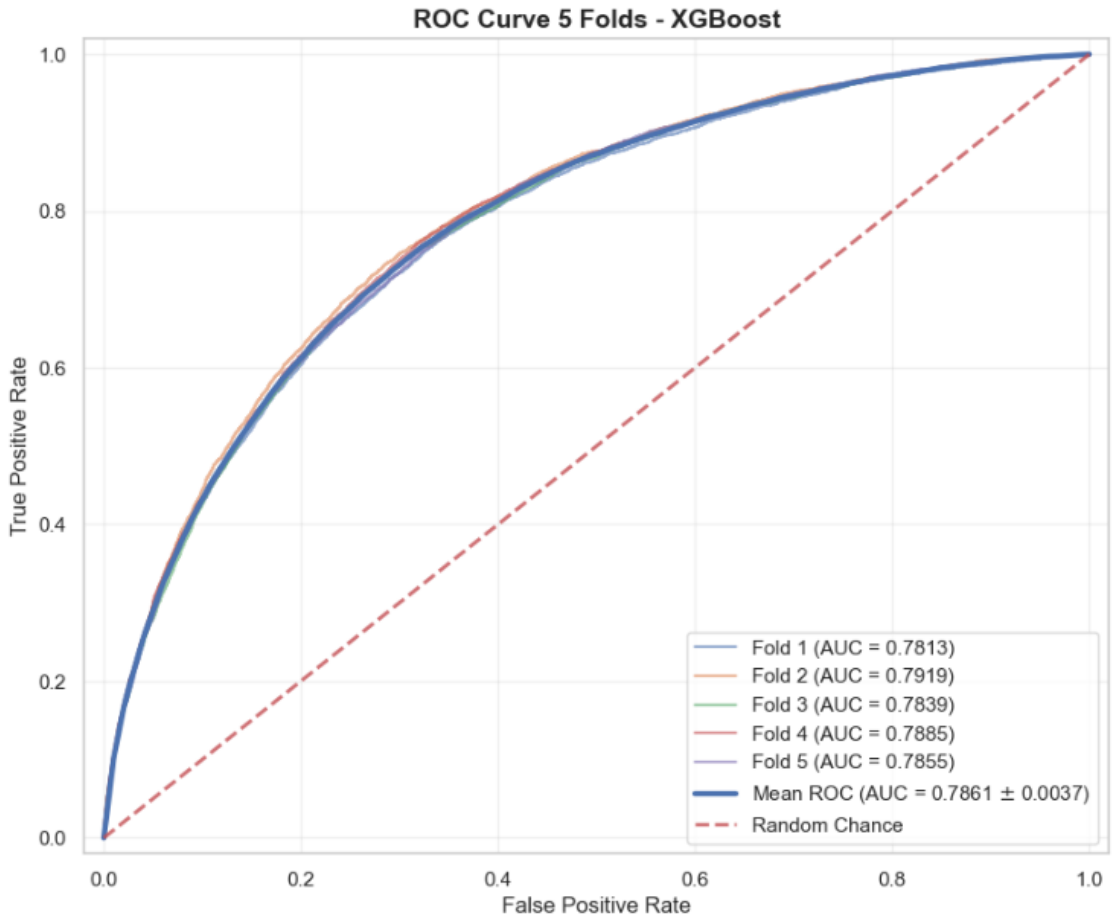
Mô hình XGBoost được xây dựng dựa trên nguyên lý Boosting, trong đó các cây quyết định được thêm vào tuần tự để sửa chữa sai số của các cây trước đó.[5] Khác với Random Forest, XGBoost được huấn luyện trên tập dữ liệu X_lgbm (giữ nguyên các giá trị NaN) để tận dụng thuật toán xử lý khuyết thiếu nội tại. Cấu trúc mô hình được tối ưu hóa cho tốc độ xử lý dữ liệu lớn như sau:

1. Sử dụng phương pháp xây dựng cây dựa trên Histogram ($tree\_method =' hist'$) giúp giảm đáng kể thời gian huấn luyện mà vẫn giữ nguyên độ chính xác.
2. Áp dụng kỹ thuật lấy mẫu con theo cả dòng và cột để tăng tính đa dạng và giảm thiểu quá khớp.
3. Sử dụng cơ chế $scale\_pos\_weight$ để trực tiếp trừng phạt các sai số dự báo trên nhóm nợ xấu (nhóm thiểu số).

Các siêu tham số thiết lập cho mô hình:

- **n_estimators**: 1000 (Số lượng cây tối đa).
- **learning_rate**: 0.05 (Tốc độ học).
- **max_depth**: 6 (Độ sâu tối đa của cây).
- **tree_method**: 'hist' (Chế độ Histogram tối ưu tốc độ).
- **subsample**: 0.8 (Sử dụng 80% dữ liệu ngẫu nhiên cho mỗi cây).
- **colsample_bytree**: 0.7 (Sử dụng 70% đặc trưng ngẫu nhiên cho mỗi cây).
- **scale_pos_weight**: Giá trị tính toán từ tỷ lệ mất cân bằng mẫu thực tế.

Kết quả thực nghiệm mô hình XGBoost Baseline:



Hình 5.2: Biểu đồ ROC Curve đánh giá mô hình XGBoost

5.2.3 Mô hình LightGBM

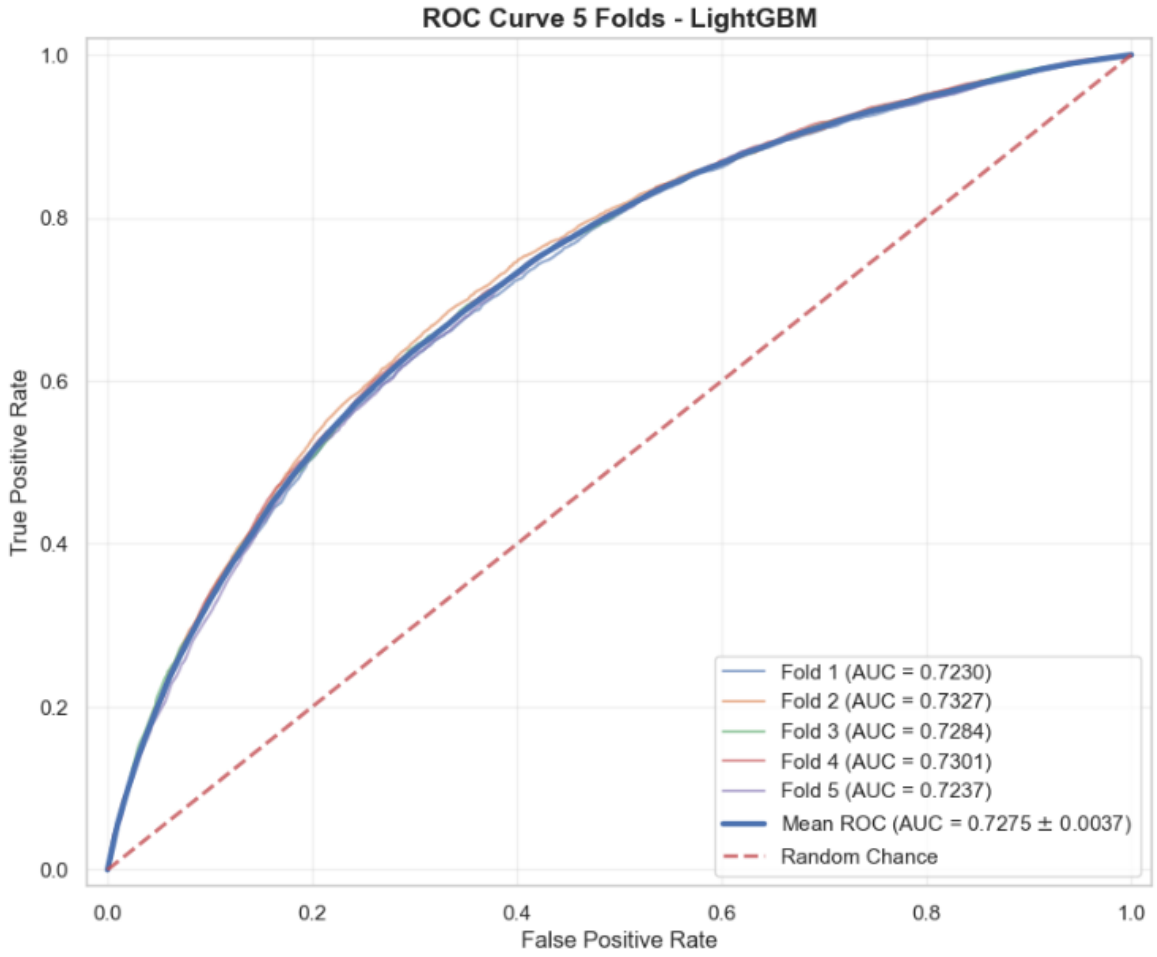
Mô hình LightGBM (Light Gradient Boosting Machine) cũng thuộc họ Gradient Boosting nhưng sử dụng cơ chế phát triển cây theo chiều lá thay vì theo chiều sâu như truyền thống.

Mô hình này được thiết kế đặc biệt để xử lý lượng dữ liệu lớn với tốc độ cao và mức tiêu thụ bộ nhớ thấp. Tương tự XGBoost, mô hình nhận đầu vào là tập X_lgbm chứa các giá trị NaN. Quy trình huấn luyện bao gồm cơ chế dừng sớm (Early Stopping) để tự động ngắt quá trình học nếu hiệu suất không cải thiện sau một số vòng lặp nhất định.[6]

Các siêu tham số thiết lập cho mô hình

- **n_estimators**: 2000 (Số lượng cây tối đa cho phép).
- **learning_rate**: 0.02 (Tốc độ học thấp để mô hình hội tụ chi tiết hơn).
- **num_leaves**: 34 (Số lượng lá tối đa trên một cây - tham số quan trọng nhất kiểm soát độ phức tạp của LightGBM).
- **max_depth**: 8 (Giới hạn độ sâu để tránh quá khớp).
- **scale_pos_weight**: Giá trị tính toán từ tỷ lệ mất cân bằng mẫu.
- **importance_type**: 'gain' (Phương pháp tính độ quan trọng của biến).

Kết quả thực nghiệm mô hình LightGBM Baseline:



Hình 5.3: Biểu đồ ROC Curve đánh giá mô hình LightGBM

5.3 Tối ưu hóa mô hình

Sau khi đánh giá hiệu quả của các mô hình cơ sở, thực hiện các bước tối ưu hóa chuyên sâu nhằm loại bỏ nhiễu, giảm độ phức tạp tính toán và tìm kiếm cấu hình tham số tốt nhất để nâng cao chỉ số AUC.

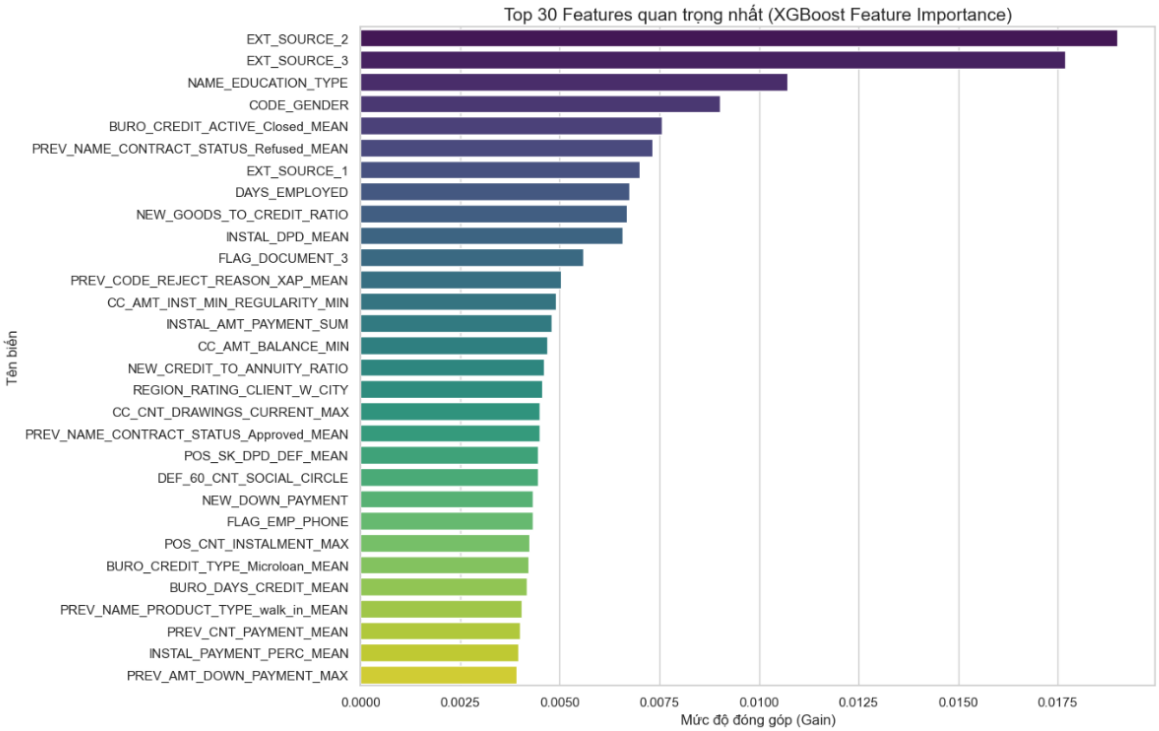
5.3.1 Lựa chọn đặc trưng

Do bộ dữ liệu có rất nhiều biến (471 đặc trưng), nếu giữ lại toàn bộ sẽ làm mô hình phức tạp và thời gian huấn luyện tăng lên. Vì vậy thực hiện chọn lọc đặc trưng dựa trên độ quan trọng của biến (Feature Importance) được trích xuất từ mô hình XGBoost, là mô hình cho kết quả tốt nhất ở giai đoạn baseline.

Quy trình thực hiện:

1. Trích xuất giá trị Gain (mức độ đóng góp trung bình vào việc giảm hàm mất mát) của từng biến trong quá trình xây dựng cây.
2. Xếp hạng các biến từ quan trọng nhất đến ít quan trọng nhất.

3. Loại bỏ hoàn toàn các biến có Importance = 0 (Zero Importance). Đây là các biến không được bất kỳ cây quyết định nào sử dụng để phân loại, do đó chúng là dư thừa và gây nhiễu.



Hình 5.4: Top 30 đặc trưng quan trọng nhất đối với mô hình

Kết quả thu được:

- Số lượng đặc trưng ban đầu: 471.
- Số lượng đặc trưng bị loại bỏ (Rác/Nhiễu): 51.
- Số lượng đặc trưng giữ lại cho mô hình cuối cùng: 420.

5.3.2 Tinh chỉnh tham số (Hyperparameter Tuning)

Sau khi xác định được tập đặc trưng tối ưu, bước tiếp theo là tìm kiếm cấu hình siêu tham số phù hợp nhất cho các mô hình:

- Tập trung vào nhóm Boosting (XGBoost, LightGBM): Đây là các thuật toán có tiềm năng đạt hiệu suất cao nhất nhưng lại rất nhạy cảm với các siêu tham số (như `learning_rate`, `num_leaves`). Do đó, việc áp dụng dò tìm tự động là bắt buộc để khai thác hết hiệu suất của mô hình.
- Đối với Random Forest: Do đặc thù thuật toán Bagging có tính ổn định cao, ít bị ảnh hưởng mạnh bởi thay đổi nhỏ của siêu tham số, đồng thời chi phí tính toán rất lớn trên tập dữ liệu này ($\sim$ 300.000 dòng) quyết định giữ nguyên bộ tham số chuẩn dựa trên kinh nghiệm thực nghiệm (Best Practices) thay vì chạy dò tìm tự động.

Thực nghiệm ở giai đoạn Baseline cho thấy:

- XGBoost hoạt động khá tốt với tham số mặc định (AUC $\sim$ 0.78).
- LightGBM có hiệu suất thấp bất thường (AUC $\sim$ 0.73), cho thấy bộ tham số mặc định không phù hợp với cấu trúc dữ liệu sau khi xử lý.

Do đó, mục tiêu của giai đoạn này là tối ưu hóa XGBoost để đạt độ ổn định cao nhất và cải thiện LightGBM để nâng cao năng lực dự báo, tạo tiền đề cho kỹ thuật kết hợp mô hình sau này.

a. Phương pháp thực hiện:

Phương pháp được thực hiện là sử dụng Optuna [8] – một framework tối ưu hóa siêu tham số tự động. Optuna sử dụng thuật toán TPE (Tree-structured Parzen Estimator), một dạng tối ưu hóa Bayes. Thuật toán này học từ kết quả của các lần thử nghiệm trước để đoán vùng tham số tiềm năng cho lần thử tiếp theo, giúp quá trình hội tụ nhanh hơn và tìm được kết quả tốt hơn.

Quá trình tối ưu được thực hiện thông qua chiến lược kiểm chứng chéo (Cross-Validation 3-Folds) để tiết kiệm thời gian, với hàm mục tiêu là cực đại hóa chỉ số AUC. Không gian tìm kiếm tập trung vào 3 nhóm tham số chính:

1. Nhóm cấu trúc cây:
   - **num_leaves** (LightGBM) và **max_depth** (XGBoost/LightGBM): Kiểm soát độ phức tạp của mô hình. Với dữ liệu Home Credit phải giới hạn độ sâu (depth) trong khoảng $[3, 8]$ để tránh Overfitting.
   - **min_child_samples** / **min_child_weight**: Số lượng mẫu tối thiểu cần thiết để tạo một nút lá.
2. Nhóm tốc độ học:
   - **learning_rate**: Tìm kiếm trong khoảng logarit [0.01, 0.1]. Tốc độ học thấp thường yêu cầu số lượng cây (**n_estimators**) lớn hơn nhưng cho kết quả chính xác hơn.
3. Nhóm điều chuẩn& Mẫu:
   - **subsample** và **colsample_bytree**: Tỷ lệ lấy mẫu dữ liệu và đặc trưng (từ 0.6 đến 1.0) để tăng tính đa dạng.
   - **reg_alpha** (L1) và **reg_lambda** (L2): Điều chỉnh các trọng số lớn để giảm nhiễu.

b. Kết quả thu được:

Quá trình tinh chỉnh đã đem lại hiệu quả rõ rệt, đặc biệt là đối với mô hình LightGBM.

- Đối với LightGBM: Quá trình tối ưu giúp chỉ số AUC tăng vọt từ mức cơ sở 0.7327 lên mức 0.7884 (tại thời điểm thử nghiệm tốt nhất). Đây là mức cải thiện rất lớn (+5.57%), đưa LightGBM từ kém hiệu quả trở thành mô hình có hiệu suất tương đương.
- Đối với XGBoost: Tiếp tục khẳng định vị thế là mô hình mạnh nhất với chỉ số AUC đạt 0.7896 (tăng $\sim$ 0.0035 so với mức cơ sở 0.7861). Bộ tham số tìm được giúp mô hình học chậm hơn (learning_rate thấp) nhưng chi tiết và sâu sắc hơn.

Dưới đây là bộ tham số tối ưu cuối cùng được lựa chọn cho hai mô hình:

1. **LightGBM:**
    - **learning_rate**: 0.0101 (Tốc độ học rất chậm, yêu cầu số lượng cây lớn).
    - **num_leaves**: 26 (Số lá thấp giúp tránh Overfitting).
    - **max_depth**: 6
    - **reg_alpha**: 4.628 (Chấp nhận Regularization L1 mạnh để lọc nhiễu).
    - **subsample**: 0.835 (Lấy mẫu dòng).
    - **colsample_bytree**: 0.634 (Lấy mẫu cột).
    - **reg_lambda**: 9.99.
2. **XGBoost:**
    - **learning_rate**: 0.0126.
    - **max_depth**: 4.
    - **min_child_weight**: 36.
    - **subsample**: 0.793.
    - **colsample_bytree**: 0.857.
    - **gamma**: 2.968 (Tham số giảm nhiễu đặc trưng của XGBoost).
    - **reg_alpha**: 4.555.
    - **reg_lambda**: 7.787.

5.3.3 Xây dựng mô hình kết hợp

Để khai thác tối đa hiệu suất của các thuật toán khác nhau và giảm thiểu rủi ro của việc phụ thuộc vào một mô hình duy nhất, áp dụng kỹ thuật Ensemble Learning ở bước cuối cùng của quá trình xây dựng.[9]

Phương pháp: Sử dụng kỹ thuật Soft Voting (Trung bình có trọng số). Kết quả dự báo cuối cùng là tổng hợp xác suất của các mô hình thành phần dựa trên độ tin cậy của chúng. Thiết lập trọng số

Dựa trên kết quả thực nghiệm sau khi tinh chỉnh tham số thiết lập tỷ lệ như sau:

- XGBoost (40%): Đóng vai trò chủ đạo nhờ độ chính xác cao nhất.
- LightGBM (40%): Đối trọng với XGBoost, có tốc độ tốt và hiệu suất tương đương.
- Random Forest (20%): Đóng vai trò ổn định hóa (Regularizer). Do cơ chế Bagging khác biệt hoàn toàn với Boosting, Random Forest giúp mô hình tổng thể không bị quá khớp (Overfitting) vào các đặc điểm nhiễu mà 2 mô hình Boosting có thể mắc phải.

Công thức:

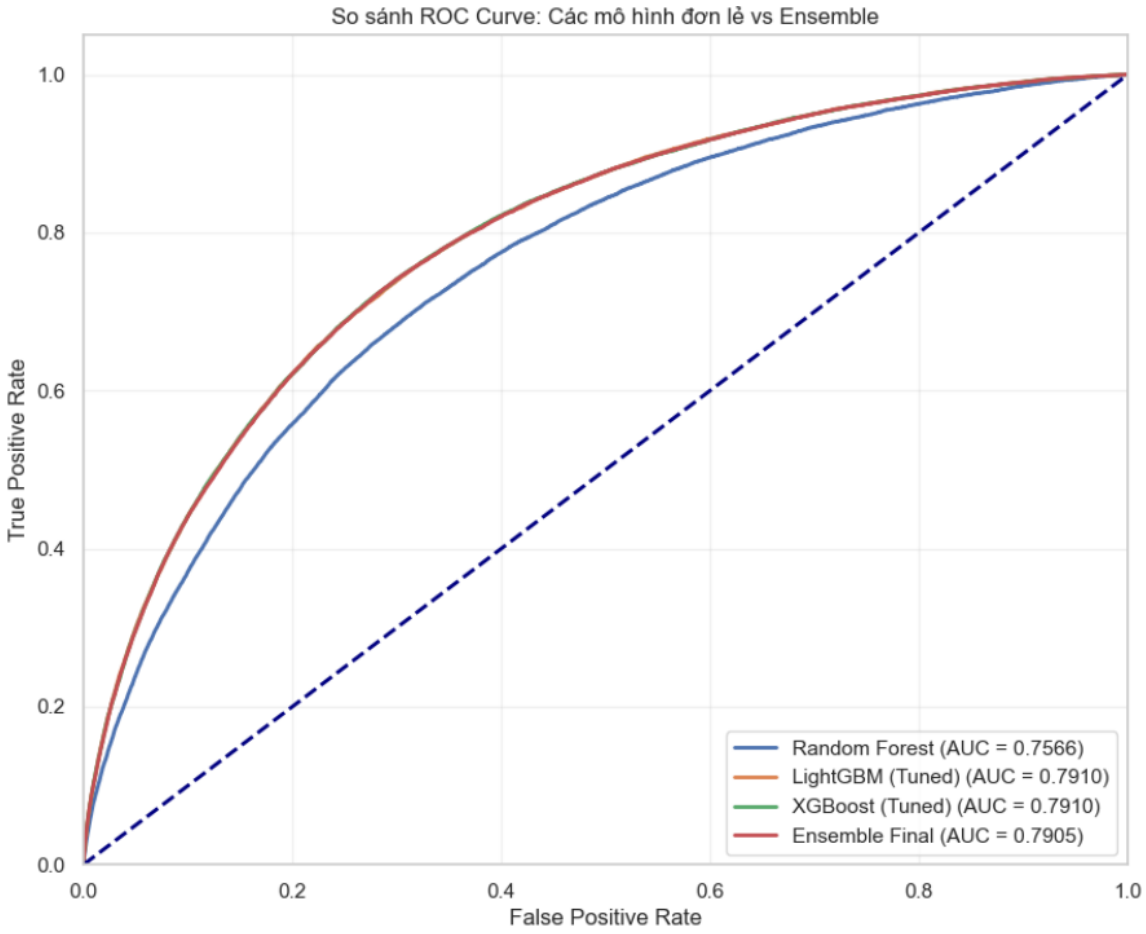
$$P_{Final} = 0.4 \times P_{XGBoost} + 0.4 \times P_{LightGBM} + 0.2 \times P_{RandomForest}$$

5.4 Đánh giá mô hình

5.4.1 Đánh giá hiệu suất tổng thể

Mô hình	ROC-AUC	Gini	KS	F1-Score	Precision	Recall
Random Forest	0.7566	0.5133	0.3840	0.2995	0.2204	0.4672
XGBoost	0.7910	0.5820	0.4415	0.3426	0.2852	0.4290
LightGBM	0.7910	0.5820	0.4396	0.3428	0.2811	0.4392
Ensemble Model	0.7905	0.5811	0.4415	0.3421	0.2867	0.4242

Bảng 5.1: Hiệu suất trên các mô hình



Hình 5.5: Biểu đồ đường cong ROC của mô hình Ensemble so với các mô hình đơn lẻ

Nhận xét:

- XGBoost và LightGBM cho khả năng phân loại tổng quát tốt hơn Random Forest (AUC ~0.79 so với ~0.76); các chỉ số nghiệp vụ Gini (~0.58) và KS (~0.44) đạt mức tốt, phù hợp với dữ liệu tín dụng phức tạp.
- Random Forest có Recall cao nhất (46.72%), giúp phát hiện nợ xấu tốt; đưa vào Ensemble (trọng số 20%) để tăng tính đa dạng và tránh bỏ sót rủi ro.
- Mô hình Ensemble đạt chỉ số AUC = 0.7905, đạt gần bằng mô hình tốt nhất nhưng cân bằng hơn với Precision cao và KS tốt; Đường cong ROC của Ensemble (màu

đỏ) bao trùm và tiệm cận tốt về góc trái phía trên. Mặc dù chỉ số AUC thấp hơn không đáng kể so với mô hình LightGBM/XGBoost đơn lẻ (0.7910) do ảnh hưởng của việc trung bình hóa với Random Forest (0.7566), nhưng việc kết hợp này mang lại tính ổn định cao hơn, giảm thiểu phương sai và rủi ro quá khớp (Overfitting) khi áp dụng vào thực tế.

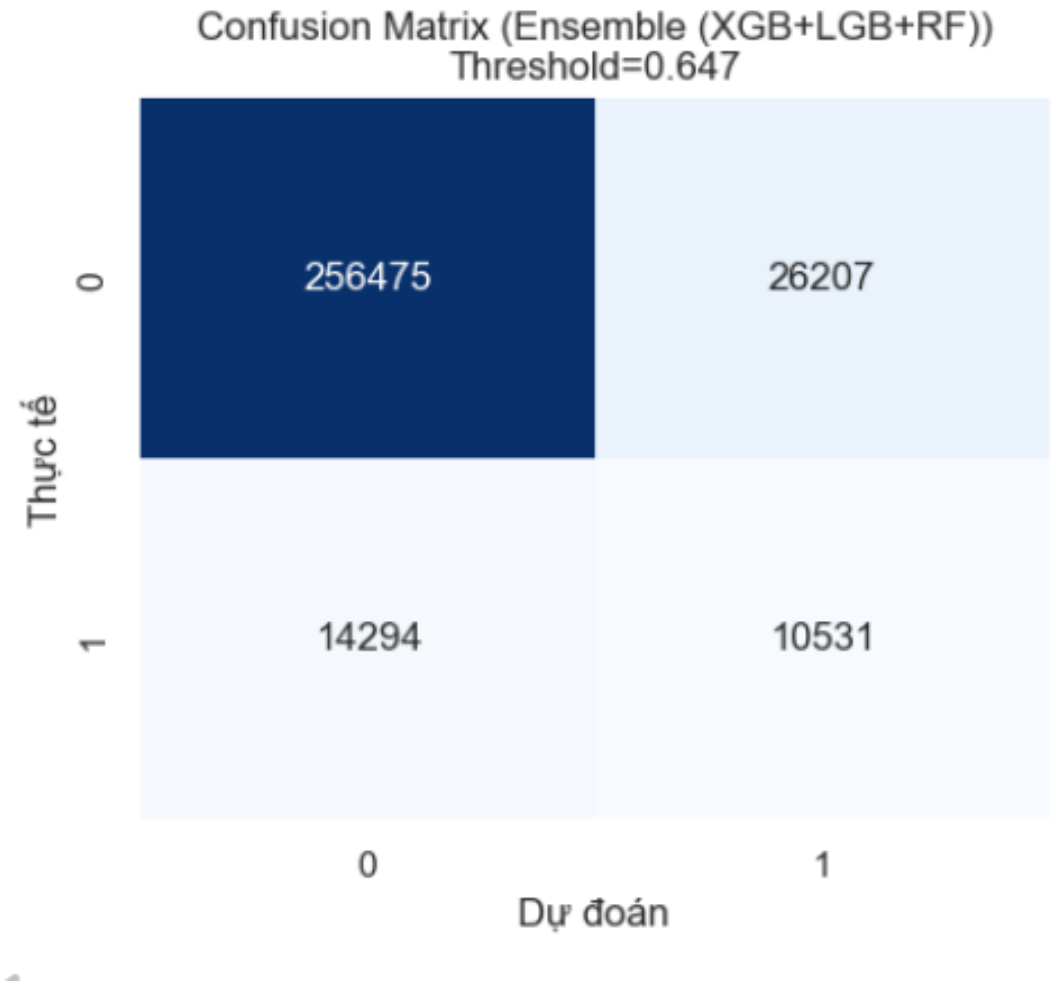
→ Ensemble Model là lựa chọn tối ưu nhờ kết hợp độ chính xác của Boosting và độ nhạy của Bagging, phù hợp triển khai thực tế.

5.4.2 Phân tích chi tiết mô hình kết hợp

a. Đánh giá hiệu suất mô hình:

Ma trận nhầm lẫn (Confusion Matrix):

Tại ngưỡng cắt tối ưu (Threshold = 0.647) được xác định để tối đa hóa điểm F1-Score phân tích chi tiết khả năng dự báo của mô hình:



Hình 5.6: Ma trận nhầm lẫn

Dựa trên ma trận nhầm lẫn, với tổng mẫu kiểm thử khoảng 307.507 hồ sơ:

- **True Negative** (256.475): Số lượng khách hàng tốt được dự báo đúng.
- **False Positive** (26.207): Số lượng khách hàng tốt bị nhầm là xấu (Báo động giả).
- **False Negative** (14.294): Số lượng khách hàng nợ xấu bị bỏ sót.
- **True Positive** (10.531): Số lượng khách hàng nợ xấu được phát hiện chính xác.

Các chỉ số chi tiết:

- Precision (Độ chính xác) = 28.67%: Trong số các hồ sơ mô hình cảnh báo rủi ro, có $\sim$29% là nợ xấu thực sự. Tỷ lệ này là chấp nhận được đối với bài toán mất cân bằng dữ liệu nghiêm trọng (nợ xấu chỉ chiếm $\sim$8%).
- Recall (Độ nhạy) = 42.42%: Mô hình đã phát hiện được hơn 42% tổng số lượng khách hàng nợ xấu trong tệp dữ liệu. Đây là chỉ số quan trọng giúp ngân hàng ngăn chặn thất thoát vốn.
- F1-Score = 0.3421: Điểm trung hòa giữa Precision và Recall, đạt mức cao nhất tại ngưỡng cắt đã chọn.

b. Đánh giá hiệu quả nghiệp vụ

- Hệ số Gini: 0.5881
  $\rightarrow$ Mô hình có Gini > 0.4 được coi là ”Tốt, và Gini tiệm cận 0.6 được coi là ”Xuất sắc”. Kết quả 0.5811 cho thấy mô hình có khả năng phân loại hồ sơ tín dụng rất mạnh mẽ.
- Chỉ số KS: 44.15%
  $\rightarrow$ Chỉ số KS đo lường khoảng cách lớn nhất giữa phân phối xác suất của nhóm khách hàng Tốt và Xấu. Mức KS > 40% cho thấy mô hình tách biệt hai nhóm khách hàng này rất rõ ràng, giúp ngân hàng dễ dàng thiết lập các điểm cắt để phê duyệt vay tự động với rủi ro thấp nhất.[11]

5.5 Ứng dụng và triển khai mô hình

5.5.1 Thực nghiệm dự báo trên tập dữ liệu mới

Trong giai đoạn này, mô hình Ensemble được sử dụng để chấm điểm tín dụng cho tập dữ liệu Test (application_test.csv), bao gồm 48.744 hồ sơ khách hàng chưa có lịch sử trả nợ.

Quy trình thực hiện:

1. Đồng bộ hóa dữ liệu: Tập dữ liệu mới được đưa qua quy trình tiền xử lý và kỹ thuật đặc trưng hoàn toàn tương đồng với tập huấn luyện.
2. Tính toán xác suất: Dữ liệu được đưa qua 3 mô hình thành phần. Kết quả xác suất cuối cùng (P_{Final} được tổng hợp theo công thức trọng số của mô hình Ensemble.
3. Phân loại nhãn dự báo : Áp dụng ngưỡng cắt tối ưu (Threshold = 0.647) đã được xác định tại giai đoạn đánh giá (Mục 5.4) để đưa ra quyết định nhị phân:
   - Nếu Xác suất $\geq$ 0.647: Dự báo là Nợ xấu $\Rightarrow$ Từ chối hoặc xem xét kỹ.

- Nếu Xác suất < 0.647: Dự báo là An toàn ⇒ Chấp thuận.

Kết quả dự báo:

Dưới đây là bảng trích xuất kết quả dự báo cho nhóm khách hàng có nguy cơ rủi ro cao nhất và nhóm an toàn nhất từ hệ thống:

```
Top 5 khách hàng có nguy cơ rủi ro cao nhất:
        SK_ID_CURR    TARGET
330322      265895  0.925817
309442      113627  0.924458
312032      132358  0.915568
318034      176483  0.909335
345576      379119  0.905696

Top 5 khách hàng an toàn nhất:
        SK_ID_CURR    TARGET
318283      178290  0.030822
312876      139208  0.032824
337112      315499  0.036135
354456      443039  0.036670
351559      421868  0.037990
```

Hình 5.7: Kết quả dự báo top 5 khách hàng nguy cơ rủi ro và an toàn

5.5.2 Xây dựng dashboard hỗ trợ ra quyết định

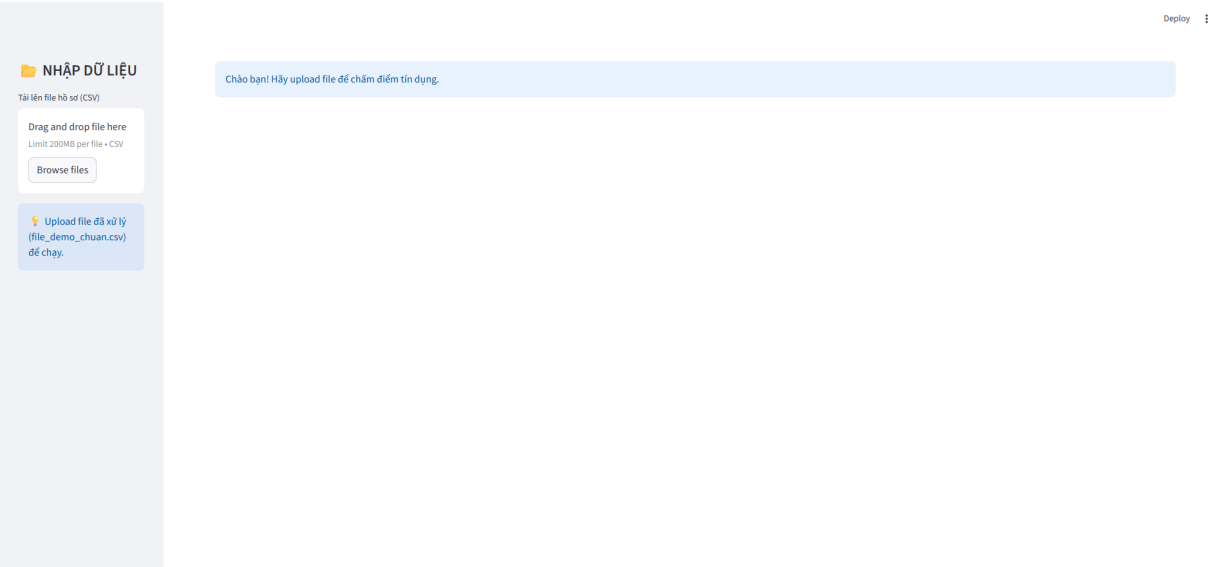
Để chuyển đổi kết quả từ mô hình học máy thành công cụ thực tiễn cho bộ phận tín dụng em đã phát triển hệ thống Dashboard tương tác sử dụng nền tảng **Streamlit**.

a. Kiến trúc và Quy trình vận hành:

Hệ thống hoạt động theo cơ chế Input-Process-Output khép kín:

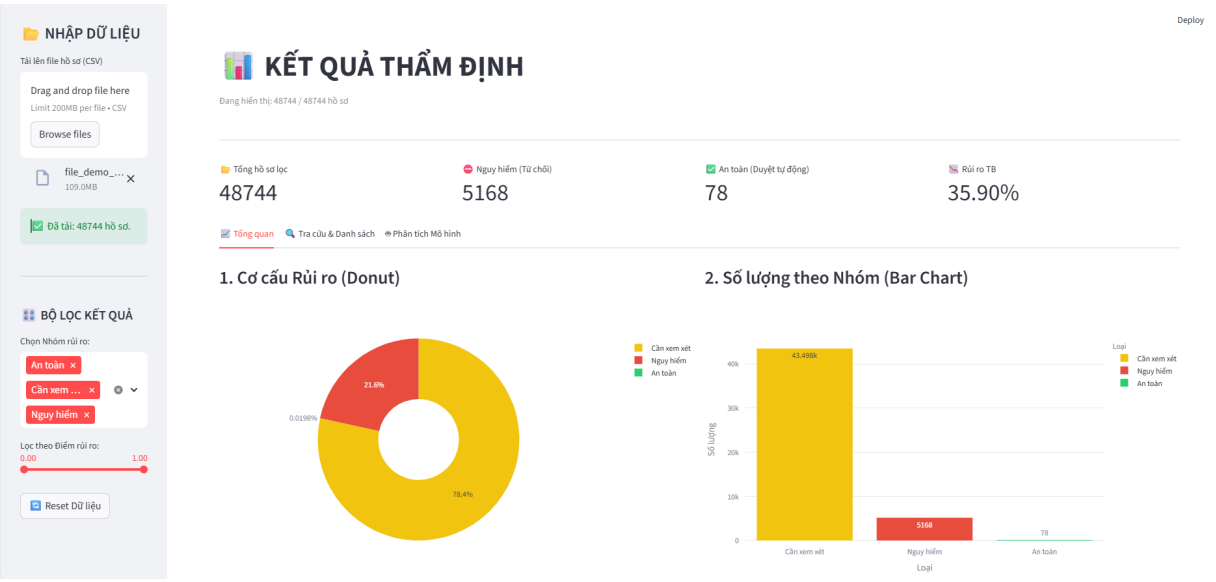
- Đầu vào : Nhân viên tín dụng tải lên tệp tin danh sách hồ sơ khách hàng mới (định dạng CSV/Excel) đã qua xử lý đặc trưng.
- Xử lý lỗi :Hệ thống tự động kích hoạt 3 mô hình thành phần (XGBoost, LightGBM, Random Forest) đã được huấn luyện và đóng gói (.pkl). Thực hiện tính toán xác suất song song và tổng hợp kết quả theo cơ chế Ensemble.
- Đầu ra : Hiển thị kết quả trên giao diện Dashboard với 3 phân hệ chức năng: Tổng quan, Tra cứu chi tiết, và Phân tích mô hình.

b. Phân tích các thành phần hiển thị



Hình 5.8: Giao diện đầu vào

Giao diện khởi động của hệ thống được thiết kế theo phong cách tối giản và trực quan, tập trung tối đa vào trải nghiệm người dùng với bố cục chia làm hai khu vực rõ rệt. Thanh điều hướng bên trái đóng vai trò là cổng nhập liệu chính, tích hợp công cụ tải tệp tin hỗ trợ kéo thả tiện lợi và chấp nhận định dạng chuẩn .CSV, đi kèm với khung thông báo hướng dẫn cụ thể (màu xanh) giúp nhân viên tín dụng dễ dàng nhận biết định dạng file chuẩn để giảm thiểu sai sót đầu vào. Tại trạng thái chờ, màn hình chính hiển thị thông báo chào mừng và hướng dẫn hành động tiếp theo, tuân thủ nguyên tắc hống lỗi trong thiết kế hệ thống bằng cách yêu cầu người dùng bắt buộc phải hoàn tất việc tải lên dữ liệu hợp lệ trước khi được phép truy cập vào các tính năng phân tích và dashboard chuyên sâu.

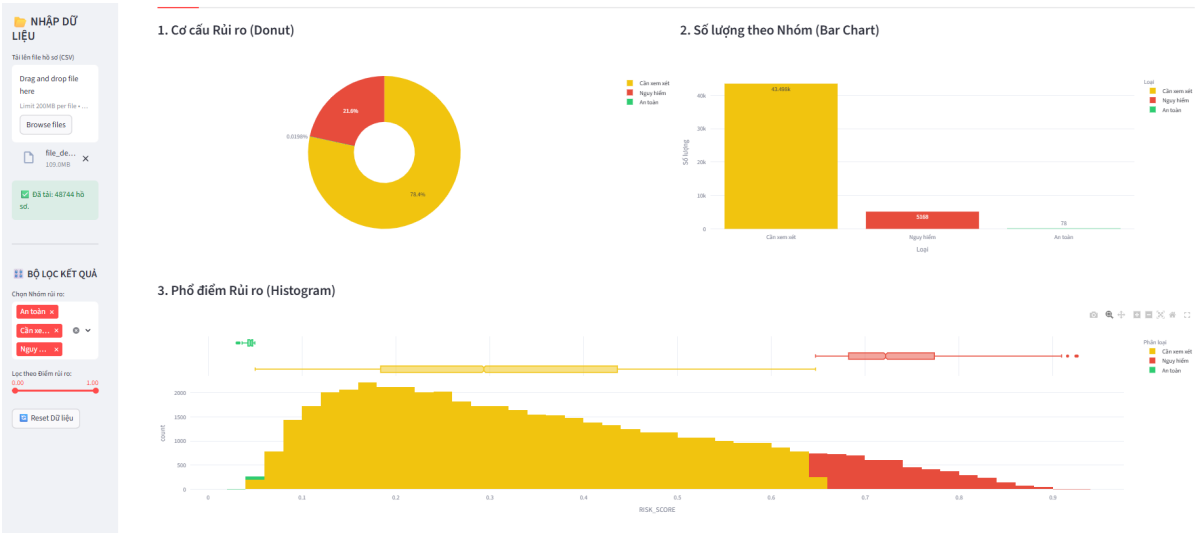


Hình 5.9: Giao diện hiển thị ban đầu

Ngay sau khi quá trình tính toán của mô hình hoàn tất, hệ thống tự động chuyển sang giao diện Dashboard phân tích với bố cục trực quan, được chia thành 3 thẻ chức năng: Tab Tổng quan, Tab Tra cứu & Danh sách , Tab Phân tích mô hình chuyên biệt để phục

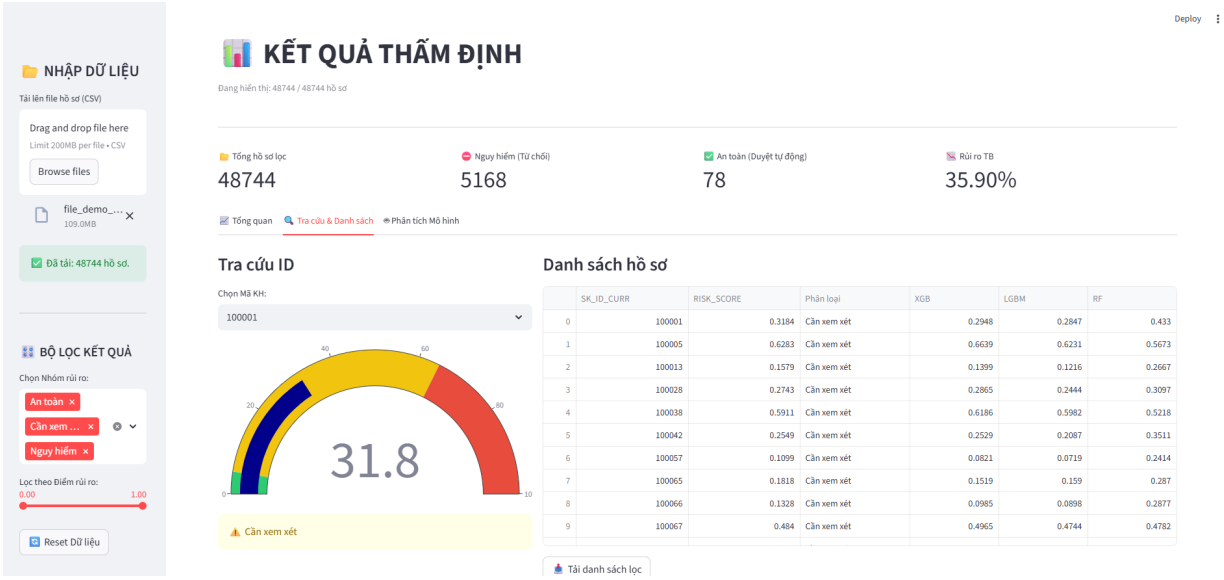
vụ các nhu cầu nghiệp vụ khác nhau. Tại Tab Tổng quan:

- **Bộ chỉ số KPI:** Cung cấp cái nhìn tức thì về tình trạng rủi ro của toàn bộ danh mục hồ sơ vừa tải lên. Với tệp dữ liệu 48.744 hồ sơ, tỷ lệ rủi ro trung bình là 35.90%. Số lượng hồ sơ ”Nguy hiểm”(bị từ chối tự động) là 5.168, giúp tiết kiệm thời gian thẩm định cho hơn 5.000 trường hợp rủi ro cao này.



Hình 5.10: Biểu đồ đánh giá điểm tín dụng

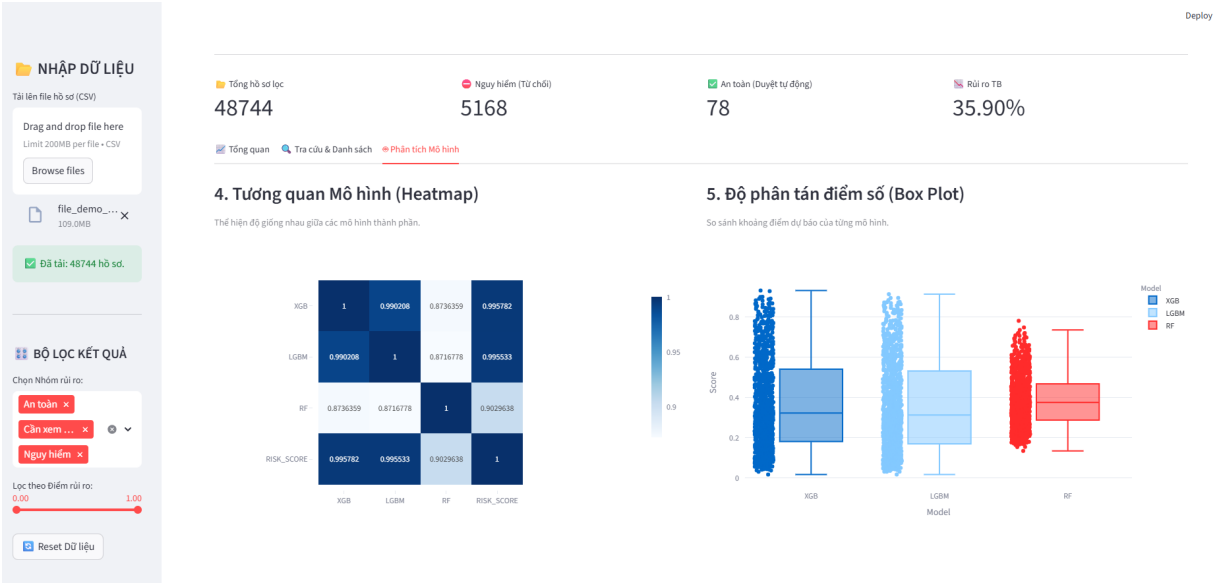
- **Donut Chart:** Trực quan hóa tỷ trọng của 3 nhóm khách hàng theo chiến lược phân tầng. Nhóm ”Cần xem xét”(Vàng) chiếm tỷ trọng lớn nhất (78.4%). Điều này phản ánh đúng thực tế là đa số khách hàng nằm trong vùng ”xám”, đòi hỏi sự kết hợp giữa máy móc và con người (thẩm định) để ra quyết định chính xác, tránh bỏ sót khách hàng tiềm năng.
- **Histogram Chart:** Cho biết sự phân bố của điểm số rủi ro (Risk Score).Phổ điểm có xu hướng tập trung ở khoảng 0.2 - 0.4 (rủi ro thấp đến trung bình), cho thấy chất lượng tệp khách hàng này tương đối tốt. Các hồ sơ rủi ro cao (Đỏ) nằm tách biệt rõ ràng ở phía đuôi phải (score > 0.65), chứng tỏ mô hình có khả năng phân tách tốt.
- **Bar chart:** Trực quan hóa số lượng tuyệt đối của từng nhóm rủi ro. Trong khi biểu đồ Donut cho thấy tỷ lệ phần trăm, thì biểu đồ cột giúp nhà quản lý ước lượng được khối lượng công việc thực tế cần xử lý. Cột ”Cần xem xét”(Màu vàng) cao vượt trội với khoảng hơn 43.000 hồ sơ. Điều này báo hiệu cho bộ phận vận hành rằng khối lượng hồ sơ cần thẩm định thủ công là rất lớn, đòi hỏi sự chuẩn bị kỹ lưỡng về nhân sự. Ngược lại, cột ”Nguy hiểm”(Màu đỏ) với hơn 5.000 hồ sơ cho thấy hệ thống đã tự động lọc bỏ một lượng lớn khách hàng kém chất lượng ngay từ cửa ngõ, giúp tiết kiệm đáng kể chi phí thẩm định cho ngân hàng.



Hình 5.11: Tra cứu và danh sách điểm tín dụng

Giao diện ở tab **Tra cứu & Danh sách** này được thiết kế để phục vụ quy trình thẩm định sâu đối với từng khách hàng cụ thể hoặc xem xét danh sách tổng hợp. Các thành phần chính bao gồm:

- **Gauge Chart:** Công cụ này đóng vai trò trực quan hóa điểm số tín dụng cho từng hồ sơ khách hàng cụ thể đang được tra cứu, giúp nhân viên thẩm định định vị ngay lập tức mức độ rủi ro mà không cần phân tích số liệu thô. Biểu đồ được thiết kế với ba vùng màu đặc trưng tương ứng với các ngưỡng quyết định: vùng Xanh (An toàn), vùng Vàng (Cần xem xét) và vùng Đỏ (Nguy hiểm). Ví dụ, với khách hàng mã 100001 trong hình minh họa đạt điểm rủi ro 31.8%, kim chỉ thị nằm sâu trong vùng màu vàng báo hiệu đây là trường hợp chưa đủ điều kiện phê duyệt tự động và bắt buộc phải chuyển sang quy trình thẩm định thủ công kỹ lưỡng hơn.
- **Bảng dữ liệu:** Bảng danh sách hồ sơ không chỉ đơn thuần hiển thị kết quả cuối cùng mà còn đảm bảo tính minh bạch của hệ thống bằng cách công khai các điểm số thành phần từ ba mô hình con là XGBoost, LightGBM và Random Forest. Cơ chế hiển thị đa chiều này cho phép chuyên viên rủi ro đối chiếu mức độ đồng thuận giữa các thuật toán; chẳng hạn nếu cả ba mô hình đều đưa ra điểm số cao thì quyết định từ chối sẽ có độ tin cậy rất lớn, ngược lại nếu có sự chênh lệch thì hồ sơ cần được chú ý đặc biệt. Ngoài ra, bảng dữ liệu này hỗ trợ sắp xếp, lọc và trích xuất báo cáo ra file lưu trữ, phục vụ đắc lực cho công tác kiểm soát nội bộ và quản lý danh mục vay.



Hình 5.12: Phân tích mô hình

Khác với hai giao diện trước tập trung vào nghiệp vụ thẩm định, Tab Phân tích Mô hình cung cấp góc nhìn kỹ thuật chuyên sâu dành cho bộ phận Kiểm soát rủi ro hoặc Quản trị hệ thống. Giao diện này giúp giám sát hiệu suất và hành vi của các thuật toán AI.

- **Ma trận Tương quan Mô hình (Heatmap):** Tương quan giữa XGBoost và LightGBM rất cao (0.99) do cùng họ Boosting. Tuy nhiên, tương quan giữa Random Forest với hai mô hình trên thấp hơn (~ 0.87). Sự khác biệt này chứng tỏ Random Forest mang lại ”góc nhìn lạ”, giúp hệ thống ổn định hơn và tránh bị thiên kiến bởi một loại thuật toán duy nhất.
- **Biểu đồ hộp (Boxplot):** Random Forest (RF) có dải điểm phân tán rộng hơn và trung vị cao hơn so với XGB/LGBM. Điều này cho thấy RF có xu hướng khắt khe hơn trong việc đánh giá rủi ro, đóng vai trò là chốt chặn an toàn cho hệ thống.

Chương 6

Kết luận và hướng phát triển

Kết luận

Đồ án đã xây dựng một hệ thống đánh giá rủi ro tín dụng hoàn chỉnh, giải quyết bài toán rủi ro tín dụng với bộ dữ liệu quan hệ phức tạp và mất cân bằng nghiêm trọng của Home Credit. Điểm nhấn quan trọng nhất của nghiên cứu nằm ở quy trình Kỹ thuật đặc trưng chuyên sâu, khi đã chuyển đổi thành công từ 7 bảng dữ liệu rời rạc thành một không gian dữ liệu đồng nhất với 473 đặc trưng đồng thời tối ưu hóa đặc trưng bằng cách lựa chọn đặc trưng để giảm thiểu còn 420 đặc trưng chất lượng, phản ánh toàn diện hành vi tài chính của khách hàng. Trên cơ sở dữ liệu đó, việc ứng dụng thuật toán tối ưu hóa Bayes (Optuna) kết hợp với kỹ thuật học tổ hợp (Ensemble Learning) giữa XGBoost, LightGBM và Random Forest đã giúp mô hình đạt được hiệu suất vượt trội với chỉ số AUC 0.7905 và hệ số Gini 0.58.

Bên cạnh các kết quả định lượng, giá trị thực tiễn của đồ án được khẳng định thông qua việc xây dựng thành công ứng dụng Web (Dashboard) hỗ trợ ra quyết định. Hệ thống không chỉ dừng lại ở việc dự báo xác suất vỡ nợ mà còn thực hiện phân tầng rủi ro tự động thành các nhóm An toàn, Cần thẩm định và Nguy hiểm. Kết quả kiểm thử cho thấy mô hình có khả năng phân tách rõ ràng giữa tập khách hàng tốt và xấu (chỉ số KS đạt 44%), đồng thời phát hiện được hơn 42% lượng nợ xấu tiềm ẩn tại ngưỡng cắt tối ưu. Đây là cơ sở vững chắc để khẳng định tính khả thi và hiệu quả của hệ thống khi đưa vào quy trình thẩm định tín dụng thực tế, giúp ngân hàng vừa tối ưu hóa thời gian phê duyệt, vừa kiểm soát chặt chẽ rủi ro mất vốn.

Hạn chế và hướng phát triển

Mặc dù mô hình đã đạt được độ ổn định cao, nghiên cứu vẫn còn dư địa để phát triển nhằm đáp ứng tốt hơn các yêu cầu khắt khe của ngành tài chính hiện đại. Một trong những thách thức lớn nhất cần giải quyết trong giai đoạn tiếp theo là cải thiện độ chính xác (Precision) đối với nhóm khách hàng nợ xấu nhằm giảm tải áp lực cho bộ phận thẩm định thủ công. Để làm được điều này, hướng phát triển tiềm năng là tích hợp các kỹ thuật ”Giải thích mô hình”(Explainable AI) như SHAP hoặc LIME trực tiếp vào Dashboard. Việc này sẽ giúp minh bạch hóa thuật toán, chỉ rõ lý do cụ thể tại sao một hồ sơ bị đánh giá là rủi ro (ví dụ: do lịch sử trả chậm hay do thu nhập thấp), từ đó hỗ trợ nhân viên tín dụng ra quyết định tự tin hơn.

Về mặt hệ thống và nghiệp vụ, định hướng tương lai là chuyển đổi kết quả dự báo từ

dạng xác suất sang dạng Thẻ điểm tín dụng với thang điểm chuẩn (ví dụ 300-850) để phù hợp với thói quen sử dụng của các tổ chức tài chính truyền thống. Đồng thời, hệ thống có thể được nâng cấp để xử lý dữ liệu lớn theo thời gian thực bằng cách ứng dụng các công nghệ Feature Store hiện đại, cho phép cập nhật điểm số tín dụng ngay khi khách hàng phát sinh giao dịch mới, thay vì chỉ đánh giá dựa trên dữ liệu lịch sử tĩnh như hiện tại.

Tài liệu tham khảo

[1] Tống Đình Quỳ. (2020). *Giáo trình Xác suất thống kê*. Đại học Bách Khoa Hà Nội.

[2] Trần Ngọc Thăng. (2025). *Slide học máy cơ bản*. Tài liệu giảng dạy nội bộ, Đại học Bách Khoa Hà nội

[3] Đinh Viết Sang. (2022). *Bài giảng Học máy (Machine Learning)*. Viện Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội.

[4] Thân Quang Khoát. (2021). *Nhập môn Trí tuệ nhân tạo*. Đại học Bách Khoa Hà Nội.

[5] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.

[6] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc.

[7] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32. Springer.

[8] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2623–2631).

[9] Zhou, Z. H. (2012). *Ensemble Methods: Foundations and Algorithms*. CRC Press.

[10] Siddiqi, N. (2017). *Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards* (2nd ed.). John Wiley & Sons.

[11] Babaev, A., Savchenko, M., Tuzhilin, A., & Umerenkov, D. (2019). E.T.-RNN: Applying Deep Learning to Credit Loan Applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2183–2190).