BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

HỒ QUANG HUY

KHÓA LUẬN TỐT NGHIỆP

MÔ HÌNH CHẤM ĐIỂM TÍN DỤNG SỬ DỤNG CÁC THUẬT TOÁN HỌC MÁY

Ngành: Khoa học máy tính

Chuyên ngành: Khoa học dữ liệu

Giảng viên hướng dẫn: TS. Nguyễn Chí Kiên

TP.HỒ CHÍ MINH, THÁNG 05 NĂM 2023

MINISTRY OF INDUSTRY AND TRADE

INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY

FACULTY OF INFORMATION TECHNOLOGY

HO QUANG HUY

GRADUATION THESIS

CREDIT SCORING MODEL
USING MACHINE LEARNING ALGORITHMS

Major: Data Science

Instructor: PhD. Nguyen Chi Kien

HO CHI MINH CITY, MAY 2023

CONTENT SUMMARY

Title: Credit scoring model using machine learning algorithms

Abstract:

- Reason for writing: Credit scoring is a crucial topic in the financial industry. The aim of credit scoring is to evaluate the creditworthiness of a borrower. Traditional credit scoring models have been the industry standard for many years and generally rely on a borrower's credit history, payment behavior, and other financial data to generate a credit score. The problem with traditional credit scoring models is that they rely on limited and static features, which can lead to inaccurate predictions and give an unfair assessment of creditworthiness. This has led to the adoption of machine learning algorithms in credit scoring.
- Problem: Building a credit scoring model for individual customers.
- Method: This process includes:
    - Techniques related to data preprocessing, feature engineering, model selection, and model evaluation.
    - Apply machine learning algorithms: Logistic regression, Random forest, Neural Network, XGBoost, LightGBM, CatBoost to build models.
- Results:
    - Successfully trained all models, with the best model achieving a quality evaluation index AUC of over 0.8.
    - Creating a credit score based on the output results of the prediction model.
- Conclusion:
    - The feature extraction process from the original data is a very important process. It is the quality foundation for the model training process.

- The CatBoost model has the best quality among the prediction models. In addition, the LightGBM and XGBoost models also have good results.
- Due to imbalanced data, the Logistic regression and Random forest models did not achieve high results.
- Deep learning is not suitable for this type of data as the Neural network model did not achieve good results.
- Ensemble learning helps improve the prediction quality of the model, although the quality improvement is not yet significantly outstanding.

LỜI CẢM ƠN

Lời đầu tiên em xin chân thành bày tỏ lòng biết ơn đến thầy **TS. Nguyễn Chí Kiên** người đã hết lòng giúp đỡ, hướng dẫn, truyền đạt kinh nghiệm, góp ý và tạo mọi điều kiện tốt nhất cho em hoàn thành khóa luận tốt nghiệp.

Em xin cảm ơn thầy **ThS. Trương Vĩnh Linh** và thầy **ThS. Lưu Giang Nam** đã đồng ý phản biện đề tài khóa luận tốt nghiệp của em. Em tin rằng những đánh giá phản biện của các thầy sẽ góp phần quan trọng trong việc hoàn thiện đề tài này.

Em xin cảm ơn thầy **ThS. Nguyễn Hữu Tình**, giáo viên chủ nhiệm lớp DHKHDL15A đã giúp đỡ, hỗ trợ, truyền đạt kinh nghiệm về cả kiến thức và tinh thần trong quá trình học tập và nghiên cứu.

Em xin cảm ơn đến toàn thể quý thầy cô trong khoa Công Nghệ Thông Tin – Trường Đại học Công nghiệp Thành phố Hồ Chí Minh đã tận tình truyền đạt những kiến thức quý báu cũng như tạo điều kiện thuận lợi cho em trong suốt quá trình học tập nghiên cứu và cho đến khi thực hiện khóa luận tốt nghiệp.

Em xin cảm ơn đến quý công ty SmartNet và các anh chị đồng nghiệp trong phòng Business Intelligence đã tạo điều kiện cho em học tập và thực hành trên những dự án thực tế. Điều này giúp em tích lũy được rất nhiều kinh nghiệm để áp dụng vào quá trình nghiên cứu và thực hiện khóa luận tốt nghiệp.

Mặc dù đã nỗ lực cố gắng cùng với sự tận tâm của thầy giáo hướng dẫn nhưng do trình độ còn hạn chế, nội dung đề tài còn khá mới mẻ với em nên khó tránh khỏi những sai sót trong quá trình tiếp nhận kiến thức. Em rất mong nhận được góp ý từ phía thầy cô để em có thể hoàn thiện đề tài.

Xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP.Hồ Chí Minh, Tháng 05 Năm 2023

GIÁO VIÊN HƯỚNG DẪN

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP.Hồ Chí Minh, Tháng 05 Năm 2023

GIÁO VIÊN PHẢN BIỆN

MỤC LỤC

MỤC LỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ VÀ VIẾT TẮT

TỪ NGỮ	Ý NGHĨA
Activation function	Hàm kích hoạt
Bin	Ngăn chứa dữ liệu
Binary tree	Cây nhị phân
Bootstrap	Lấy mẫu tái lập
Categorical feature	Thuộc tính phân loại
CNN - convolutional neural networks	mạng nơron tích chập
Concatenate layer	Lớp nối của mạng neural network truyền thẳng
CV	Cross-validation
Decision trees	Thuật toán cây quyết định
Dense layer	Lớp mạng của mạng mạng neural network truyền thẳng
Ensemble learning	Phương pháp học đồng bộ
Feed-forward neural network	Mạng neural network truyền thẳng
FN	False negative
FP	False positive
FPR	False positive rate
GAN - generative adversarial networks	Mạng đối nghịch
GBDT	Gradient boosting decision tree - Cây quyết định tăng cường độ dốc

GD - Gradient descent	Thuật toán tối ưu hàm mất mát được sử dụng trong các bài toán Machine Learning và Deep Learning
GOSS	Gradient-based one-side sampling - Lấy mẫu một phía dựa trên độ dốc
Hidden layers	Lớp ẩn của neural network
Input layer	Lớp đầu vào của neural network
Iteration	Vòng lặp
IV	Information value - Chỉ số giá trị thông tin
Label	Nhãn
Loss function	Hàm mất mát
LR - Learning rate	Tốc độ học
Max-depth	Độ sâu lớn nhất
NN	Neural network
Node	Nốt
Output layer	Lớp đầu ra của neural network
Overfitting - Trang bị quá mức	Là hiện tượng mô hình tìm được quá khớp với dữ liệu training
RNN - recurrent neural networks	Mạng nơron hồi quy
SGD - Stochastic Gradient Descent	Thuật toán giảm độ dốc ngẫu nhiên
std	Standard deviation - Độ lệch chuẩn

Test set	Tập dữ liệu kiểm thử
Threshold	Ngưỡng phân loại
TN	True negative
TP	True positive
TPR	True positive rate
Train set	Tập dữ liệu huấn luyện
WOE	Weight of evidence - Trọng số dấu hiệu

CHƯƠNG 1. GIỚI THIỆU

1.1. Tổng quan

1.1.1. Bối cảnh

Tín dụng đóng một vai trò quan trọng trong hệ thống tài chính. Hiện nay, tín dụng đang tăng trưởng cực kỳ mạnh mẽ với sự gia tăng liên tục về số lượng khách hàng đi và nhu cầu tín dụng. Xu hướng này không chỉ giới hạn ở các nước phát triển mà còn được thấy ở các thị trường mới nổi. Tại Việt Nam, số lượng người sử dụng tín dụng đã tăng nhanh trong những năm gần đây, nhờ tăng trưởng kinh tế của đất nước và sự gia tăng của tầng lớp trung lưu. Theo Ngân hàng Nhà nước Việt Nam, tính đến cuối năm 2020, dư nợ tín dụng đạt 8.000 nghìn tỷ đồng, tăng 10,1% so với cuối năm 2019 [1]. Sự tăng trưởng này được thúc đẩy bởi một số yếu tố, bao gồm sự cạnh tranh ngày càng tăng giữa những tổ chức tín dụng, sự gia tăng của các nền tảng cho vay trực tuyến và sự phát triển của các sản phẩm và dịch vụ tín dụng mới. Các tổ chức cho vay hiện đang cung cấp nhiều loại sản phẩm tín dụng hơn để đáp ứng các nhu cầu khác nhau của khách hàng, chẳng hạn như cho vay cá nhân, cho vay mua ô tô, cho vay mua nhà… Ngoài ra, việc áp dụng các dịch vụ cho vay dựa trên các nền tảng số đã giúp khách hàng tiếp cận tín dụng dễ dàng hơn, giờ đây khách hàng có thể đăng ký khoản vay trực tuyến và nhận tiền trong vòng vài ngày.

Đi kèm với sự phát triển mạnh của các hoạt động tín dụng đó là vấn đề kiểm soát rủi ro, đặc biệt trong bối cảnh kinh tế hiện đại, khi thế giới vừa trải qua đại dịch COVID-19 với tình trạng mất việc làm lan rộng và tình trạng bất ổn kinh tế. Những tổ chức hoặc cá nhân cho vay đang đối mặt với rủi ro vỡ nợ gia tăng.

Chấm điểm tín dụng là một giai đoạn quan trọng trong quy trình quản lý rủi ro của các ngân hàng, các định chế tài chính, các tổ chức tín dụng. Chấm điểm tín dụng tốt sẽ góp phần làm cho chất lượng cho vay tốt hơn. Chất lượng cho vay là yếu tố quyết định hàng đầu đến sự cạnh tranh, tồn tại và lợi nhuận của các ngân hàng, các định chế tài chính, các tổ chức tín dụng [2].

Các mô hình chấm điểm tín dụng được ra đời nhằm xác định mức độ tin cậy về tín dụng của các cá nhân cũng như doanh nghiệp từ đó tối thiểu hóa rủi ro khoản vay. Theo Muhammad Azeem Qureshi [3], chấm điểm tín dụng là một công cụ quan trọng được các ngân hàng và tổ chức tài chính sử dụng để đánh giá mức độ tin cậy của người đi vay. Nó liên quan đến việc sử dụng các mô hình thống kê để phân tích lịch sử tín dụng, hành vi thanh toán và các dữ liệu tài chính khác của người vay để tạo điểm tín dụng. Mục đích của việc chấm điểm tín dụng là giảm rủi ro vỡ nợ và tối đa hóa lợi nhuận cho tổ chức tín dụng. Ví dụ, điểm tín dụng FICO dựa trên các yếu tố lịch sử thanh toán, các khoản nợ, độ dài lịch sử tín dụng, cơ cấu tín dụng, tín dụng mới để đánh giá điểm tín dụng của người đi vay [4]. Hiện nay, với với những tiến bộ công nghệ, các mô hình chấm điểm tín dụng đã trở nên tự động hơn và dựa trên các nguồn dữ liệu thay thế. Các thuật toán học máy ngày càng được sử dụng nhiều hơn trong các mô hình chấm điểm tín dụng nhằm phân tích lượng dữ liệu khổng lồ và xác định các mẫu dữ liệu không rõ ràng. Việc sử dụng các thuật toán học máy (Machine learning - ML) có khả năng cải thiện đáng kể độ chính xác của các mô hình chấm điểm tín dụng. Điều này sẽ mở rộng khả năng tiếp cận tín dụng cho những cá nhân có thể không có thông tin về lịch sử tín dụng [5].

1.1.2. Lý do chọn đề tài

Điểm tín dụng đóng một vai trò quan trọng trong việc xác định mức độ tin cậy của người đi vay và là điều cần thiết để các ngân hàng, các định chế tài chính, các tổ chức tín dụng đưa ra các quyết định cho vay đúng đắn.

Các mô hình chấm điểm tín dụng truyền thống thường dựa vào các thuộc tính cố định và hạn chế như lịch sử tín dụng, hành vi thanh toán và các dữ liệu tài chính khác để tạo điểm tín dụng. Cách tiếp cận này có thể dẫn đến những dự đoán không chính xác và đưa ra đánh giá không công bằng về mức độ tín nhiệm, vì nó không nắm bắt được bản chất phức tạp tín dụng của từng cá nhân. Hơn nữa, các mô hình chấm điểm tín dụng truyền thống không tính đến các yếu tố phi tài chính như lịch sử việc làm, giáo dục và các chỉ số kinh tế xã hội khác, điều này có thể dẫn đến những quyết định tín dụng rủi ro hoặc làm giảm khả năng tiếp cận tín dụng của một cá nhân. Ngoài ra,

các mô hình chấm điểm tín dụng truyền thống dễ bị chi phối bởi các yếu tố liên quan đến giá trị đạo đức. Theo Hội đồng Quan hệ Đối ngoại (CFR), Ba cơ quan xếp hạng tín dụng là Moody's Investor Services, Standard and Poor's (S&P), Fitch Group đã bị cáo buộc góp phần chính gây ra cuộc khủng hoảng tài chính toàn cầu năm 2008 khi các cơ quan này đã xếp hạng tín dụng sai lệch cho các khoản tín dụng [6]. Các nhược điểm của mô hình chấm điểm tín dụng truyền thống có thể gây những ảnh hưởng rất nặng nề đến hệ thống tài chính.

Các thuật toán học máy có thể được áp dụng để giải quyết những vấn đề này. Các thuật toán học máy có thể phân tích với khối lượng dữ liệu lớn, bao gồm cả những yếu tố phi tài chính để tạo ra điểm tín dụng chính xác và mang tính dự đoán hơn [7]. Các thuật toán này cũng có thể đưa ra những thuộc tính quan trọng của mô hình từ đó cung cấp cái nhìn sâu sắc hơn về các yếu tố ảnh hưởng đến quyết định tín dụng. Chính vì thấy được những ưu điểm của phương pháp này, em chọn nghiên cứu "Xây dựng mô hình chấm điểm tín dụng sử dụng các thuật toán học máy" làm đề tài khóa luận tốt nghiệp của mình.

1.2. Mục tiêu nghiên cứu

- Hiểu về các kiến thức tài chính liên quan đến hoạt động cho vay tín dụng.
- Nắm rõ được kiến thức, cách hoạt động và ứng dụng của các thuật toán học máy:
    - Logistic regression
    - Random forest
    - XGBoost
    - LightGBM
    - CatBoost
    - Neural Networks
- Phân tích và khám phá dữ liệu, thu thập thông tin chi tiết từ quá trình phân tích khám phá.
- Trích xuất và đánh giá được các đặc trưng từ dữ liệu.

- Xây dựng và huấn luyện mô hình dữ liệu từ các thuật toán học máy đã tìm hiểu.
- Kết hợp mô hình đã huấn luyện bằng các phương pháp học đồng bộ.
- So sánh hiệu suất giữa các mô hình đã đào tạo, đưa ra đánh giá.
- Chấm điểm tín dụng sử dụng dữ liệu có sẵn từ mô hình đã huấn luyện có hiệu suất tốt nhất.
- Đưa ra các thuộc tính quan trọng trong dữ liệu.

1.3. Phạm vi nghiên cứu

- Nghiên cứu đào tạo mô hình dữ liệu sử dụng các thuật toán:
    - Logistic regression
    - Random forest
    - XGBoost
    - LightGBM
    - CatBoost
    - Neural Networks
- Sử dụng bộ dữ liệu "Home Credit Default Risk" có sẵn và được công khai trên Kaggle [8].

1.4. Ý nghĩa khoa học và thực tiễn

Từ góc độ khoa học, việc phát triển các mô hình chấm điểm tín dụng chính xác và đáng tin cậy là một lĩnh vực nghiên cứu tích cực trong lĩnh vực học máy và khoa học dữ liệu. Các thuật toán học máy ngày càng được sử dụng để phát triển các mô hình chấm điểm tín dụng chính xác hơn khi có thể kết hợp được nhiều nguồn dữ liệu và thuộc tính hơn. Đề tài này có khả năng đóng góp vào việc phát triển các thuật toán, kỹ thuật và phương pháp mới để các mô hình đánh giá rủi ro tín dụng ngày càng chính xác, công bằng và minh bạch hơn.

Từ góc độ thực tế, các mô hình chấm điểm tín dụng được các tổ chức tín dụng và các tổ chức tài chính sử dụng để đánh giá mức độ tin cậy của người vay và đưa ra các quyết định cho vay đúng đắn. Sử dụng các mô hình chấm điểm tín dụng chính xác có thể giúp người cho vay xác định rủi ro tín dụng của những người đăng

ký khoản vay. Việc sử dụng các thuật toán học máy trong chấm điểm tín dụng cũng có thể giúp người cho vay tự động hóa quy trình đăng ký khoản vay, giảm chi phí và nâng cao hiệu quả tổng thể của hoạt động cho vay của họ.

Khi kết hợp với các mô hình chấm điểm tín dụng truyền thống, việc sử dụng các thuật toán học máy có thể nâng cao độ chính xác và độ tin cậy của các đánh giá rủi ro tín dụng. Bằng cách kết hợp với các mô hình chấm điểm tín dụng truyền thống có thể cung cấp bức tranh toàn diện hơn về mức độ tin cậy của người đăng ký khoản vay, từ đó có thể xác định được các mẫu và mối quan hệ dữ liệu mà các mô hình chấm điểm tín dụng truyền thống có thể đã bỏ qua. Điều này có thể giúp các quyết định cho vay đúng hơn và giảm thiểu rủi ro tín dụng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Chấm điểm tín dụng

2.1.1. Điểm tín dụng

Điểm tín dụng là một thang điểm số từ 300 đến 850 phản ánh mức độ đáng tin cậy của cá nhân hoặc tổ chức vay nợ. Điểm tín dụng ra đời vào năm 1989 khi tập đoàn Fair Isaac (nay là FICO) đã dựa trên thông tin từ các báo cáo tín dụng của người tiêu dùng để tạo ra khung điểm số đánh giá khả năng trả nợ của một cá nhân. Từ điểm tín dụng, bên cho vay có thể xác định được mức độ rủi ro mà bên vay nợ có thể có, từ đó xác định các yếu tố cho vay và xác định bên vay nợ sẽ trả lại khoản vay đúng hạn hay không [9].

Theo nghiên cứu do Cục Bảo vệ Tài chính Người tiêu dùng (CFPB) thực hiện, điểm tín dụng có tính dự báo cao về rủi ro tín dụng [10]. Nghiên cứu cho thấy những người vay có điểm tín dụng cao hơn ít có khả năng vỡ nợ hơn đáng kể, trong khi những người vay có điểm tín dụng thấp hơn có nhiều khả năng vỡ nợ hơn đáng kể. Ngoài ra, nghiên cứu cho thấy rằng điểm tín dụng có khả năng dự đoán rủi ro tín dụng cao hơn các yếu tố khác, chẳng hạn như thu nhập hoặc lịch sử việc làm.

Điểm tín dụng cũng có thể có tác động lớn đến tình trạng tài chính của một cá nhân. Nghiên cứu được thực hiện bởi Ngân hàng Dự trữ Liên bang Philadelphia [11] cho thấy những cá nhân có điểm tín dụng cao hơn sẽ có khả năng được chấp thuận cho các khoản vay và thẻ tín dụng hơn, đồng thời được cung cấp mức lãi suất thấp hơn và các điều khoản tốt hơn. Mặt khác, những cá nhân có điểm tín dụng thấp hơn có nhiều khả năng bị từ chối tín dụng hoặc đưa ra mức lãi suất cao hơn và các điều khoản kém thuận lợi.

2.1.2. Chấm điểm tín dụng

2.1.2.1. Khái niệm

Chấm điểm tín dụng là một phương pháp thống kê được sử dụng để đánh giá mức độ tín nhiệm của một cá nhân hoặc doanh nghiệp liên quan đến các hoạt động

tín dụng. Hand & Jacka là hai nhà kinh tế học và thống kê kê tài chính đã tuyên bố rằng: "quá trình (của các tổ chức tài chính) lập mô hình mức độ tin cậy được gọi là chấm điểm tín dụng" [12].

Chấm điểm tín dụng liên quan đến việc phân tích lịch sử tín dụng, thu nhập, tỷ lệ nợ trên thu nhập và các dữ liệu tài chính khác của người đi vay để dự đoán khả năng trả nợ đúng hạn của họ. Kết quả của quá trình chấm điểm tín dụng là điểm tín dụng nằm trong khoảng từ 300 đến 850 ở Hoa Kỳ.

Chấm điểm tín dụng là một công cụ quan trọng đối với người cho vay và các tổ chức tài chính vì sẽ giúp đưa ra quyết định sáng suốt về việc cho vay, quyết định hạn mức và lãi suất cho vay.

2.1.2.2. Các mô hình chấm điểm tín dụng

Hiện nay, có nhiều mô hình tính điểm tín dụng khác nhau được sử dụng, nhưng mô hình phổ biến nhất là điểm FICO. Điểm FICO do Fair Isaac Corporation phát triển. Mô hình tính điểm này được sử dụng bởi đại đa số những tổ chức tín dụng ở Hoa Kỳ [13].

Điểm FICO nằm trong khoảng từ 300 đến 850, điểm số càng cao cho thấy rủi ro tín dụng càng thấp. Điểm FICO từ 740 trở lên thường được coi là rất tốt, trong khi điểm dưới 580 được coi là kém [14]. Mô hình chấm điểm này dựa trên nhiều thông tin khác nhau đối với từng cá nhân. Tuy nhiên, để đánh giá điểm tín dụng, một số yếu tố chung được đặt ra với trọng số tương ứng như lịch sử thanh toán (35%), số tiền nợ (30%), độ dài lịch sử tín dụng (15%), kết hợp tín dụng (10%), tín dụng mới (10%).

Điểm FICO được cập nhật thường xuyên để phản ánh những thay đổi trong lịch sử tín dụng của người vay. Ví dụ: nếu người vay trả hết số dư thẻ tín dụng, điểm FICO của họ có thể tăng lên. Tương tự, nếu người vay bỏ lỡ khoản thanh toán hoặc nhận khoản nợ mới, điểm FICO của họ có thể giảm.

VantageScore cũng là mô hình chấm điểm tín dụng phổ biến hiện nay ngoài điểm FICO. Mô hình VantageScore được giới thiệu vào năm 2006 bởi ba văn phòng

tín dụng chính là Equifax, Experian và TransUnion [15]. Giống như điểm FICO, VantageScore được các tổ chức tín dụng sử dụng để đánh giá mức độ tin cậy của người đi vay và đưa ra quyết định cho vay sáng suốt. Các yếu tố đá giá chính đi kèm với trọng số của mô hình này bao gồm lịch sử thanh toán (41%), tuổi và loại tín dụng (20%), tỷ lệ hạn mức tín dụng sử dụng (20%), tổng số dư (6%), hành vi tín dụng gần đây (11%), tín dụng khả dụng (2%) [16].

Ngoài 2 mô hình chấm điểm tín dụng này còn có nhiều mô hình khác tuy nhiên đây là những mô hình được sử dụng rộng rãi trong hoạt động tín dụng và được đa phần các tổ chức tín dụng sử dụng.

2.1.3. Nguyên cứu liên quan

Nghiên cứu của Zhu và cộng sự [17] đã khám phá tác động của các nguồn dữ liệu thay thế đối với các mô hình chấm điểm tín dụng. Các tác giả nhận thấy rằng việc kết hợp dữ liệu thay thế, chẳng hạn như thanh toán tiền thuê nhà và hóa đơn tiện ích, đã cải thiện độ chính xác của các mô hình chấm điểm tín dụng và có thể giúp mở rộng khả năng tiếp cận tín dụng cho những người dân chưa được phục vụ đầy đủ. Các tác giả gợi ý rằng việc sử dụng các nguồn dữ liệu thay thế có thể là một công cụ quan trọng để giảm sai lệch trong các mô hình chấm điểm tín dụng.

Trong một nghiên cứu của Lin và các cộng sự [18] đã điều tra tác động của dữ liệu truyền thông xã hội đối với các mô hình chấm điểm tín dụng. Các tác giả nhận thấy rằng việc kết hợp dữ liệu truyền thông xã hội (chẳng hạn như các hoạt động trên nền tảng Facebook) vào các mô hình chấm điểm tín dụng đã cải thiện độ chính xác của mô hình trong việc dự đoán rủi ro tín dụng. Các tác giả gợi ý rằng dữ liệu truyền thông xã hội có thể là một công cụ hữu ích để giảm sự sai lệch trong các mô hình chấm điểm tín dụng và mở rộng khả năng tiếp cận tín dụng cho những người chưa có hoặc rất ít lịch sử tín dụng.

Một nghiên cứu của Bhattacharya và Bose [19] đã điều tra tác động của các yếu tố kinh tế vĩ mô đến rủi ro tín dụng. Các tác giả nhận thấy rằng các yếu tố kinh tế vĩ mô, chẳng hạn như tăng trưởng GDP và lạm phát, là những yếu tố dự báo đáng kể về

rủi ro tín dụng. Các tác giả gợi ý rằng việc kết hợp các yếu tố kinh tế vĩ mô vào các mô hình chấm điểm tín dụng có thể cải thiện độ chính xác của chúng và giúp các cá nhân, tổ chức cho vay đưa ra quyết định cho vay đúng đắn hơn.

2.2. Học máy và học sâu

2.2.1. Học máy

Học máy là một lĩnh vực khoa học máy tính đang phát triển nhanh chóng, tập trung vào việc phát triển các thuật toán và mô hình cho phép máy tính học hỏi từ dữ liệu và đưa ra dự đoán hoặc quyết định. Theo IBM [20], học máy là một nhánh của trí tuệ nhân tạo (AI) và khoa học máy tính tập trung vào việc sử dụng dữ liệu và thuật toán để bắt chước cách con người học, dần dần cải thiện độ chính xác của nó.

Có thể chia hệ thống của các thuật toán học máy thành 3 thành phần chính:

- Quy trình quyết định (Decision process): Các thuật toán dựa trên dữ liệu đầu vào có thể gắn nhãn hoặc không gắn nhãn. Dựa trên dữ liệu đầu vào này thuật toán sẽ ước tính về mẫu dữ liệu từ đó đưa ra dự đoán hoặc phân loại.
- Hàm lỗi (Loss function): Loss function đánh giá dự đoán của mô hình. Nếu có các mẫu dữ liệu đã biết, loss function có thể so sánh để đánh giá độ chính xác của mô hình.
- Quy trình tối ưu hóa mô hình (Model optimization): Nếu mô hình có thể phù hợp hơn với các mẫu dữ liệu trong tập huấn luyện, thì các trọng số sẽ được điều chỉnh để giảm sự chênh lệch giữa dữ liệu đã biết và dữ liệu dự đoán của mô hình. Thuật toán sẽ lặp lại quy trình đánh giá, tối ưu hóa và cập nhật các trọng số của mô hình một cách tự động cho đến khi đạt đến ngưỡng tối ưu kỳ vọng.

Các thuật học máy được chia thành ba loại chính:

- Học máy có giám sát (Supervised machine learning): Supervised machine learning được xác định bằng cách sử dụng các bộ dữ liệu được gắn nhãn để huấn luyện các thuật toán nhằm phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác.

- Học máy không giám sát (Unsupervised machine learning): Unsupervised machine learning sử dụng các thuật toán học máy để phân tích và phân cụm các bộ dữ liệu không được gắn nhãn. Các thuật toán này khám phá các mẫu hoặc nhóm dữ liệu ẩn mà không cần sự can thiệp của con người.
- Học máy bán giám sát (Semi-supervised machine learning): Trong quá trình đào tạo, semi-supervised machine learning sử dụng tập dữ liệu được gắn nhãn nhỏ hơn để huấn luyện khả năng phân loại phân loại của mô hình và trích xuất đặc trưng từ tập dữ liệu lớn hơn, không được gắn nhãn. Semi-supervised machine learning có thể giải quyết vấn đề không có đủ dữ liệu được gán nhãn cho thuật toán học có giám sát.

Hiện nay, đã có nhiều thuật toán học máy được nghiên cứu và phát triển ứng dụng. Một số thuật toán học máy thường được sử dụng như: Linear regression, Logistic regression, Clustering, Decision trees, Random forests, Neural networks.

Một trong những điểm mạnh chính của học máy là khả năng xác định các mẫu và mối quan hệ phức tạp trong các tập dữ liệu lớn mà con người khó hoặc không thể phát hiện được [21]. Học máy cũng có khả năng tự động hóa nhiều nhiệm vụ hiện đang được thực hiện bởi con người, giải phóng thời gian và nguồn lực cho các nhiệm vụ phức tạp hoặc sáng tạo hơn.

Tuy nhiên, cũng có những thách thức liên quan đến học máy, đặc biệt là liên quan đến các vấn đề về sai lệch, công bằng và minh bạch. Các thuật toán học máy chỉ tốt khi dữ liệu được đào tạo đầy đủ và đúng đắn. Nếu dữ liệu này chứa các sai lệch hoặc không chính xác, điều này có thể dẫn đến các dự đoán sai lệch hoặc không chính xác. Vì vậy, những nghiên cứu và ứng dụng cần nhận thực được những thách thức này và hướng tới phát triển các thuật toán mạnh mẽ, công bằng và minh bạch.

2.2.2. Học sâu

Học sâu là một nhánh của học máy liên quan đến việc sử dụng mạng nơron nhân tạo (artificial neural networks) nhiều lớp, artificial neural networks là một mô hình tính toán được lấy cảm hứng từ cấu trúc và chức năng của bộ não con người.

Các mạng này được thiết kế để tìm hiểu và mô hình hóa các mẫu phức tạp trong dữ liệu bằng cách xử lý lượng lớn dữ liệu và điều chỉnh các kết nối giữa các lớp để đáp ứng với phản hồi [22]. Học sâu đã được sử dụng trong nhiều ứng dụng, chẳng hạn như nhận dạng hình ảnh, nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên.

Học sâu loại bỏ một số bước tiền xử lý dữ liệu. Các thuật toán này có thể xử lý dữ liệu phi cấu trúc, chẳng hạn như văn bản và hình ảnh, đồng thời các mô hình học sâu cũng có thể tự động trích xuất đặc trưng [23].

Quá trình đào tạo mô hình học sâu thường bao gồm việc xây dựng các lớp mạng nơron và cung cấp cho mô hình một lượng lớn dữ liệu, mỗi lớp được xây dựng dựa trên lớp trước đó để tinh chỉnh và tối ưu hóa dự đoán hoặc phân loại. Quá trình tính toán này thông qua mạng được gọi là lan truyền thuận (forward propagation). Sau đó, mô hình điều chỉnh các kết nối giữa các lớp để đáp ứng với phản hồi. Quá trình này, được gọi là lan truyền ngược (backpropagation) bằng cách sử dụng các thuật toán như gradient descent (GD), liên quan đến việc tính toán sai số giữa đầu ra dự đoán của mô hình và đầu ra thực tế, sau đó sử dụng sai số này để cập nhật trọng số và độ lệch của các kết nối giữa các nơron trong mạng. Quá trình này được lặp đi lặp lại nhiều lần, với mục tiêu giảm thiểu sai số giữa đầu ra dự đoán và đầu ra thực tế.

Có nhiều loại thuật toán học sâu, bao gồm mạng nơron tích chập (Convolutional Neural Network - CNN), mạng nơron hồi quy (Recurrent Neural Network - RNN) và mạng đối nghịch (Generative Adversarial Network - GAN). CNN thường được sử dụng để nhận dạng hình ảnh và video, trong khi RNN thường được sử dụng để xử lý ngôn ngữ tự nhiên và dự đoán trình tự. GAN là một loại thuật toán học không giám sát có thể được sử dụng cho các tác vụ như tạo hình ảnh, dữ liệu mới và tổng hợp dữ liệu.

Học sâu đã được chứng minh là một công cụ mạnh mẽ. Các nghiên cứu đang được tiếp tục tiến hành để khám phá những tiềm năng của các mô hình học sâu nhằm giải quyết các vấn đề phức tạp trong nhiều lĩnh vực khác nhau.

2.3. Bài toán ứng dụng các thuật toán học máy vào mô hình chấm điểm tín dụng

2.3.1. Tổng quan

Áp dụng các thuật toán học máy vào mô hình chấm điểm tín dụng đề cập đến việc sử dụng các thuật toán học máy để phát triển các mô hình chấm điểm tín dụng chính xác và hiệu quả hơn [24]. Các mô hình chấm điểm tín truyền thống dựa trên các tính năng cố định và hạn chế, chẳng hạn như lịch sử tín dụng và hành vi thanh toán của người đi vay để tạo ra điểm tín dụng. Tuy nhiên, các thuật toán học máy có thể phân tích các tập dữ liệu lớn và phức tạp, bao gồm các nguồn dữ liệu phi truyền thống như hoạt động trên mạng xã hội, hành vi mua sắm trực tuyến, … để tạo ra điểm tín dụng chính xác hơn. Điều này có thể giúp các tổ chức tài chính đưa ra quyết định cho vay tốt hơn, cải thiện và quản lý rủi ro, đồng thời thúc đẩy sự ổn định và tăng trưởng tài chính.

Việc sử dụng các thuật toán học máy trong chấm điểm tín dụng đã thu hút được sự chú ý đáng kể trong những năm gần đây do khả năng cải thiện tính chính xác, công bằng và minh bạch của các mô hình chấm điểm tín dụng. Tuy nhiên, cũng có một số thách thức liên quan đến việc áp dụng các thuật toán học máy cho các mô hình chấm điểm tín dụng. Một trong những thách thức chính là vấn đề về khả năng diễn giải. Các thuật toán học máy như Neural Networks, Random Forest, … thường được coi là "black box", nghĩa là hoạt động bên trong của thuật toán không thể nhìn thấy và khó hiểu. Điều này có thể dẫn đến những khó khăn trong việc giải thích quá trình ra quyết định của mô hình và có thể dẫn đến sự không tin tưởng từ cả người đi vay và người cho vay. Một số nghiên cứu đã đề xuất cách tiếp cận để giải quyết vấn đề này bằng cách kết hợp các thuật toán học máy khác nhau để tạo ra các mô hình dễ hiểu hơn.

Một thách thức khác là vấn đề thiên vị. Các thuật toán học máy chỉ khách quan khi dữ liệu mà chúng được đào tạo đầy đủ và chính xác. Nếu dữ liệu đào tạo bị sai lệch, mô hình kết quả cũng sẽ bị sai lệch. Điều này có thể dẫn đến việc đối xử không công bằng đối với một số nhóm người vay nhất định, chẳng hạn như những người đến từ các cộng đồng ít được đại diện. Những thách thức này cần phải được xem xét cẩn

thận khi áp dụng các thuật toán học máy cho các mô hình chấm điểm tín dụng và phải thực hiện các biện pháp thích hợp để đảm bảo rằng các mô hình thu được là chính xác, công bằng và minh bạch.

Ngoài các vấn đề về sai lệch và minh bạch, vấn đề overfitting cũng là một thách thức khác liên quan đến các thuật toán học máy trong chấm điểm tín dụng. Overfitting xảy ra khi mô hình quá phức tạp và quá khớp với dữ liệu huấn luyện, dẫn đến khả năng dự đoán kém đối với dữ liệu mới. Overfitting có thể dẫn đến dự đoán không chính xác và giảm hiệu suất của mô hình. Để giải quyết vấn đề này, một số kỹ thuật sẽ được đề xuất trong báo cáo bao gồm chuẩn hóa dữ liệu và xác thực chéo (Cross-validation).

2.3.2. Nghiên cứu liên quan

Nghiên cứu được thực hiện bởi Jiang và cộng sự [25] đã so sánh độ chính xác của một số mô hình chấm điểm tín dụng khác nhau, bao gồm Logistic Regression, Decision Trees và Neural Networks. Nghiên cứu cho thấy mạng nơron (Neural Networks) là mô hình chính xác nhất để dự đoán rủi ro tín dụng, với tỷ lệ chính xác là 90%. Các tác giả gợi ý rằng Neural Networks có thể là một thuật toán hữu ích cho những tổ chức cho vay muốn cải thiện mô hình chấm điểm tín dụng của họ.

Trong một nghiên cứu của Bouzouita và các cộng sự [26] đã sử dụng phương pháp học máy để phát triển mô hình chấm điểm tín dụng cho các doanh nghiệp vừa và nhỏ (SME). Các tác giả nhận thấy rằng mô hình này có độ chính xác cao trong việc dự đoán rủi ro tín dụng cho các doanh nghiệp vừa và nhỏ, với tỷ lệ chính xác là 92,5%. Các tác giả gợi ý rằng mô hình này có thể là một công cụ hữu ích cho những người cho vay muốn đưa ra quyết định cho vay đúng đắn đối với các doanh nghiệp vừa và nhỏ.

Một thuật toán học máy phổ biến được sử dụng trong chấm điểm tín dụng là Decision Trees. Một nghiên cứu của Zhang và các cộng sự [27] đã đề xuất một mô hình sử dụng thuật toán Decision Trees để chấm điểm tín dụng từ một bộ dữ liệu lớn chứa rất nhiều thuộc tính, bao gồm cả dữ liệu tài chính truyền thống và dữ liệu

phi truyền thống như sử dụng điện thoại di động và các hoạt động trên mạng xã hội. Nghiên cứu cho thấy mô hình sử dụng thuật toán Decision Trees hoạt động tốt hơn các mô hình tính điểm tín dụng truyền thống, với độ chính xác và điểm AUC cao hơn.

Ngoài ra, một nghiên cứu của Lachos-Perez và các cộng sự [28] đã đề xuất một mô hình chấm điểm tín dụng dựa trên lựa chọn và phân loại biến Bayes bằng cách sử dụng Logistic Regression và phân tích biệt thức tuyến tính. Mô hình đạt được độ chính xác cao và cho thấy những cải tiến đáng kể về khả năng diễn giải, tính công bằng và minh bạch. Các tác giả cũng nhấn mạnh về tầm quan trọng của khả năng diễn giải và tính minh bạch trong các mô hình chấm điểm tín dụng, đặc biệt là trong việc tránh thiên vị và phân biệt đối xử đối với một số nhóm người đi vay.

Việc sử dụng các thuật toán học máy trong chấm điểm tín dụng đã cho thấy kết quả đầy hứa hẹn trong việc cải thiện tính chính xác, công bằng và minh bạch của các mô hình chấm điểm tín dụng. Random Forest, Gradient Boosted Decision Trees, Neural Networks là các thuật toán học máy phổ biến được sử dụng để chấm điểm tín dụng. Các thuật toán này sử dụng các nguồn dữ liệu truyền thống và phi truyền thống để tạo điểm tín dụng có thể cung cấp cái nhìn sâu sắc hơn về mức độ tin cậy của người đi vay.

2.4. Thuật toán học máy được ứng dụng vào xây mô hình chấm điểm tín dụng

2.4.1. Logistic regression

2.4.1.1. Tổng quan

Logistic regression là một thuật toán học máy có giám sát được sử dụng cho các tác vụ phân loại, logistic regression dự đoán xác suất của một biến mục tiêu [29]. Thuật toán logistic regression mô hình hóa mối quan hệ giữa các thuộc tính dữ liệu đầu vào và phân lớp đầu ra bằng cách sử dụng hàm logistic, còn được gọi là hàm sigmoid.

Hàm logistic nhận bất kỳ số có giá trị thực nào và đầu ra là giá trị trong khoảng (0,1). Thuật toán học các tham số của hàm logistic bằng cách giảm thiểu hàm chi phí, điển hình là cross-entropy loss với việc sử dụng thuật toán tối ưu hóa như

gradient descent. Thuật toán logistic regression có thể xử lý cả tính năng đầu vào phân loại và liên tục và thường được sử dụng trong các lĩnh vực khác nhau, bao gồm tài chính, chăm sóc sức khỏe và tiếp thị, …. Thuật toán này dễ diễn giải và có thể cung cấp thông tin chi tiết về mối quan hệ giữa các tính năng đầu vào và lớp đầu ra.

2.4.1.2. Lý thuyết liên quan

Đầu ra dự đoán của logistic regression:

$$f(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x}) \qquad \textit{PT 2.1}$$

Trong đó:

- **x** là dữ liệu đầu vào.
- θ là hàm logistic.
- **w** là các tham số của thuật toán.

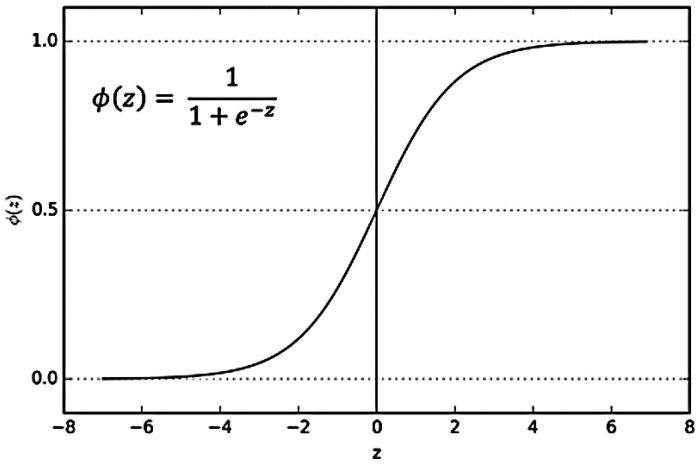
Đầu ra dự đoán của logistic regression là giá trị xác suất của biến mục tiêu thuộc về lớp positive (lớp dữ liệu quan trọng hơn cần được xác định đúng của bài toán), với các thuộc tính dữ liệu đầu vào. Hàm logistic, còn được gọi là hàm sigmoid, được sử dụng để chuyển đổi tổ hợp tuyến tính của các thuộc tính đầu vào thành giá trị xác suất trong khoảng $(0,1)$. Công thức hàm sigmoid được định nghĩa như sau:

$$\phi(z) = \frac{1}{1 + e^{-z}} \qquad \textit{PT 2.2}$$

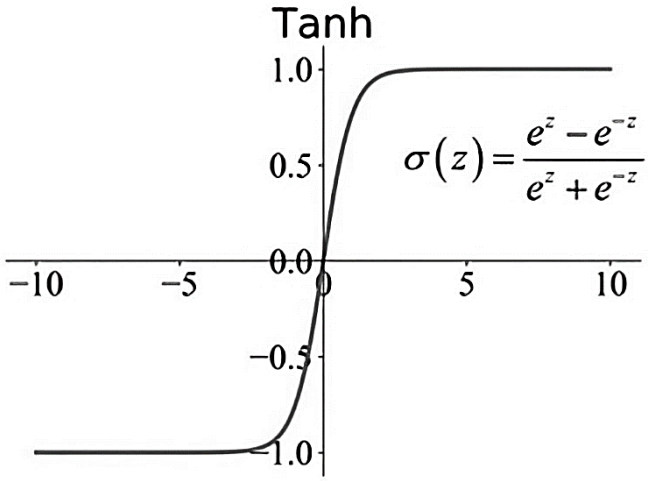
Ngoài ra, hàm tanh cũng có thể được sử dụng được sử dụng:

$$\tanh(z) = f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \qquad \textit{PT 2.3}$$

$$f'(z) = 1 - \tanh^2(z) \qquad \textit{PT 2.4}$$



Hình 2.1 Biểu diễn đồ thị hàm Sigmoid



Hình 2.2 Biểu diễn đồ thị hàm Tanh

Hàm Tanh nhận giá trị trong khoảng $(-1, 1)$ nhưng các giá trị có thể dễ dàng đưa về khoảng $(0, 1)$ để phù hợp với thuật toán.

Hàm mất mát của thuật toán logistic regression được sử dụng để đo lỗi hoặc sự khác biệt giữa đầu ra dự đoán và đầu ra thực tế đối với tập hợp các tham số của mô hình. Trong logistic regression, mục tiêu là tối đa hóa khả năng xảy ra của dữ liệu huấn luyện với tham số mô hình. Hàm mất mát được sử dụng cho hồi quy logistic là log-likelihood, còn được gọi là hàm mất mát cross-entropy [30].

Công thức cho hàm mất mát của thuật toán hồi quy logistic:

$$\mathcal{J}(\theta) = -\frac{1}{m}\Big[\sum y^{(i)}\log\Big(h_\theta\big(x^{(i)}\big)\Big) + \big(1-y^{(i)}\big)\log\Big(1 - h_\theta\big(x^{(i)}\big)\Big)\Big] \qquad PT\ 2.5$$

Trong đó:

- $\mathcal{J}(\theta)$: Hàm mất mát.
- θ: Các tham số của mô hình.
- m: Số lượng mẫu dữ liệu.
- $y^{(i)}$: Giá trị đầu ra thực tế của mẫu dữ liệu $x^{(i)}$.
- $h_\theta\big(x^{(i)}\big)$: Giá trị đầu ra dự đoán của mẫu dữ liệu $x^{(i)}$.

Hàm mất mát tính toán sự khác biệt giữa giá trị đầu ra được dự đoán và giá trị đầu ra thực tế cho từng mẫu đào tạo, tổng hợp trên tất cả các mẫu đào tạo. Mục tiêu của thuật toán logistic regression là tìm các giá trị của θ nhằm tối ưu hàm mất mát. Quá trình này thường sử dụng thuật toán tối ưu giảm độ dốc ngẫu nhiên (SGD). Công thức tổng quát của thuật toán SGD được đưa ra như sau:

$$\theta_j = \theta_j - \alpha\Big(y^{(i)} - h_\theta\big(x^{(i)}\big)\Big)x_j^{(i)} \qquad PT\ 2.6$$

Trong đó:

- θ_j: Tham số thứ j được cập nhật.
- α: Tốc độ học (LR).
- $y^{(i)}$: Giá trị đầu ra thực tế của mẫu dữ liệu huấn luyện thứ i.
- $h_\theta\big(x^{(i)}\big)$: Giá trị đầu ra dự đoán của mẫu dữ liệu huấn luyện thứ i.
- $x_j^{(i)}$: Thuộc tính thứ j của mẫu dữ liệu đào tạo thứ i.

Công thức trên tính toán sự khác biệt giữa các giá trị đầu ra được dự đoán và giá trị đầu ra thực tế cho một mẫu dữ liệu đào tạo duy nhất và cập nhật từng tham số theo giá trị thuộc tính tương ứng của nó và sự khác biệt giữa các giá trị đầu ra được

dự đoán và giá trị đầu ra thực tế. Tốc độ học α kiểm soát kích thước bước giảm của quá trình cập nhật tham số, với giá trị α lớn quá trình hội tụ của mô hình sẽ diễn ra nhanh hơn, nhưng cũng có nguy cơ vượt quá giá trị tham số tối ưu. Thuật toán SGD áp dụng quá trình lặp lại công thức cập nhật này cho từng mẫu dữ liệu huấn luyện trong tập dữ liệu, cho đến khi đạt được sự hội tụ hoặc số lần lặp tối đa.

Bằng cách tối ưu hàm mất mát, thuật toán có thể tìm được tập hợp tham số mô hình tốt nhất để tối đa hóa khả năng dự đoán của mô hình thông qua dữ liệu huấn luyện từ đó cải thiện độ chính xác của các dự đoán trên dữ liệu mới.

2.4.2. Random forest

2.4.2.1. Tổng quan

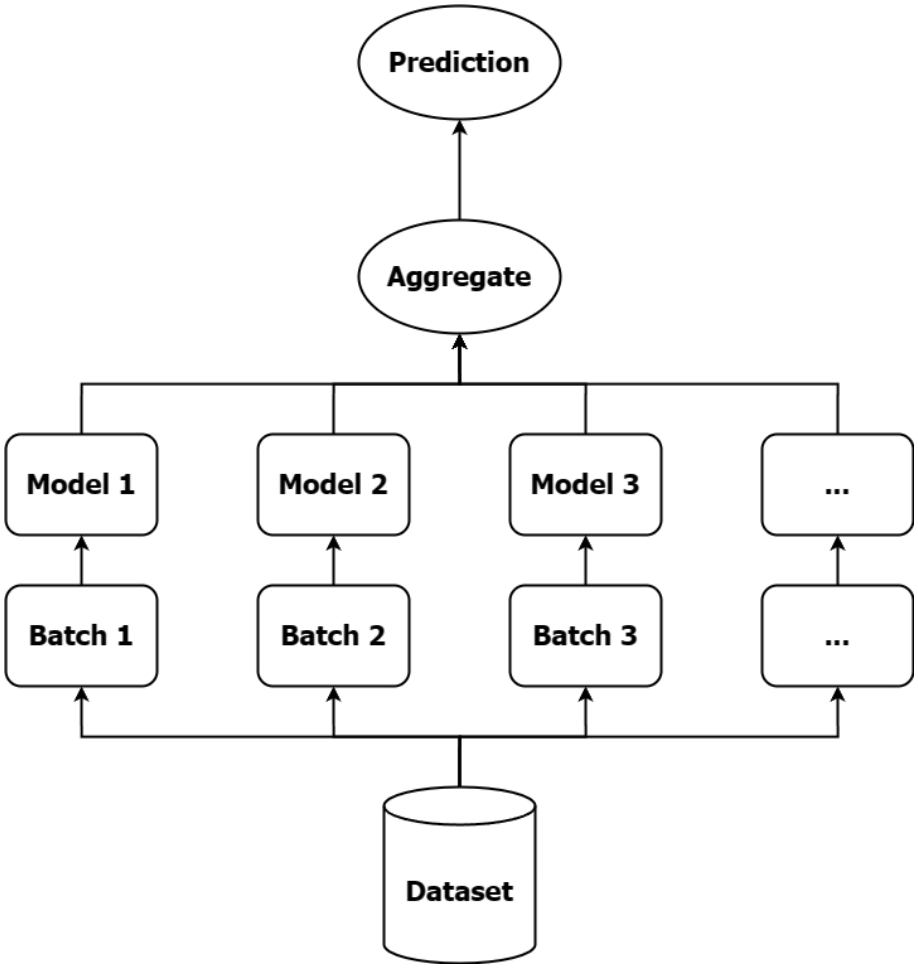
Random forest là một thuật học có giám sát được sử dụng phổ biến trong học máy. Thuật toán này dựa trên thuật toán cây quyết định (decision trees), là thuật toán phân loại dữ liệu dựa trên một loạt các quyết định phân loại. Thuật random forest xây dựng nhiều cây quyết định và kết hợp chúng để cải thiện độ chính xác của mô hình.

Ý tưởng cơ bản của thuật toán random forest là tạo ra một số lượng lớn cây quyết định bằng cách chọn ngẫu nhiên một tập hợp con các thuộc tính và huấn luyện từng cây trên một tập hợp con ngẫu nhiên của dữ liệu huấn luyện. Mỗi cây trong cụm sẽ dự đoán nhãn lớp của một đầu vào nhất định và dự đoán cuối cùng được thực hiện bằng cách kết hợp các dự đoán của tất cả các cây đã được xây dựng. Quá trình này được gọi là học đồng bộ (ensemble learning), trong đó nhiều mô hình được sử dụng để đưa ra dự đoán chính xác hơn các mô hình riêng lẻ [31] [32].

Thuật toán random forest có một số lợi thế so với các thuật toán học máy khác. Thuật toán có thể xử lý dữ liệu nhiều chiều, dữ liệu nhiễu và các giá trị bị thiếu và cũng có thể cung cấp các ước lượng về tầm quan trọng của thuộc tính. Hơn nữa, nó ít bị overfitting hơn so với một cây quyết định duy nhất vì có xu hướng ghi nhớ dữ liệu huấn luyện [33].

2.4.2.2. Lý thuyết liên quan

Học đồng bộ (ensemble learning) là một kỹ thuật trong đó kết hợp nhiều mô hình riêng lẻ để tạo ra một mô hình dự đoán mạnh mẽ và chính xác hơn. Ensemble learning được sử dụng rộng rãi trong học máy vì chúng có thể cải thiện độ chính xác và hiệu suất của các mô hình riêng lẻ. Ensemble learning có thể giúp giảm việc trang bị quá mức (overfitting) và nắm bắt các mẫu đa dạng trong dữ liệu từ đó cho ra mô hình dự đoán mạnh mẽ và chính xác hơn [34] [35] [36].

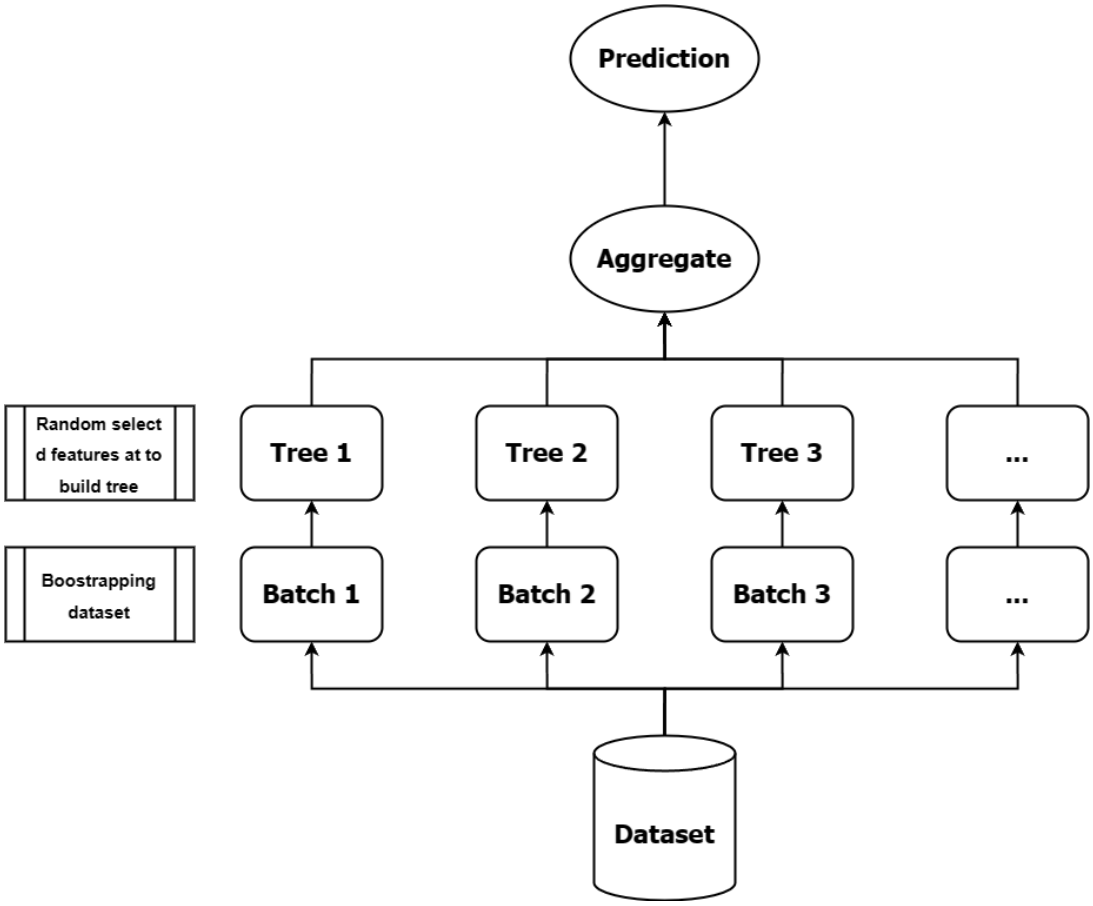


Hình 2.3 Kiến trúc tổng quan về mô hình kết hợp

Lấy mẫu tái lập (bootstrap) phương pháp lấy mẫu thống kê được sử dụng trong học máy, có liên quan chặt chẽ với thuật toán random forest. Kỹ thuật này liên quan đến việc lấy mẫu dữ liệu với sự thay thế từ tập dữ liệu gốc để tạo tập dữ liệu mới, sau đó được sử dụng để huấn luyện nhiều mô hình. Phương pháp lấy mẫu tái lập tạo ra nhiều bộ dữ liệu tương tự như dữ liệu ban đầu, nhưng có chút thay đổi nhỏ,

nhằm giảm tình trạng overfitting và cải thiện độ chính xác của mô hình [37] [38] [39] [40].

Mô hình random forest sẽ áp dụng cả hai phương pháp ensemble learning và bootstrap. Mỗi cây được đào tạo trên một mẫu bootstrap khác nhau của dữ liệu gốc. Mẫu bootstrap được lấy bằng phương pháp bootstrap đã đề cập ở phần trên kết quả của quá trình này là một tập dữ liệu mới có cùng số lượng quan sát như tập dữ liệu gốc, nhưng với một số quan sát được lặp lại và những quan sát khác bị loại trừ.



Hình 2.4 Quá trình tạo thành một mô hình của thuật toán Random forest

Các bước tạo mô hình từ thuật toán Random forest:

1. Lấy **N** mẫu ngẫu nhiên với từ tập dữ liệu gốc để tạo mẫu bootstrap.
2. Huấn luyện mô hình decision trees trên mẫu bootstrap bằng cách sử dụng ngẫu nhiên **d** thuộc tính.
3. Lặp lại bước 1 và bước 2 để xây dựng **M** mô hình decision trees.

4. Khi đã xây dựng đủ số lượng mô hình decision trees đã đặt ra. Tiến hành dự đoán giá trị biến mục tiêu bằng cách tổng hợp các dự đoán của tất cả **M** mô hình decision trees với phương pháp lấy giá trị trung bình của các dự đoán (đối với mô hình dự báo) hoặc bằng cách sử dụng biểu quyết đa số (đối với mô hình phân loại).

Kết quả dự đoán từ mô hình random forest là sự kết hợp của nhiều mô hình decision trees nên sẽ giúp cải thiện độ chính xác so với chỉ sử dụng một mô hình decision trees. Đồng thời giúp cho kết quả ít bị chệch, giảm thiểu được hiện tượng overfitting ở mô hình decision trees, một điều mà mô hình decision trees thường xuyên gặp phải [41].

2.4.3. Neural Network

2.4.3.1. Tổng quan

Neural Network (NN) là một mô hình xử lý thông tin mô phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Đó là một hệ thống gồm các nút hoặc nơron được kết nối với nhau, hoạt động cùng nhau để tìm hiểu các mẫu và mối quan hệ trong dữ liệu [42].

NN bao gồm ba thành phần chính: lớp đầu vào (input layer), lớp ẩn (hidden layers) và lớp đầu ra (output layer). Input layer lấy dữ liệu thô và đưa vào lớp ẩn. Trong bài nghiên cứu này dữ liệu vào input layer là dữ liệu về khách hàng, chẳng hạn như tuổi, thu nhập, lịch sử tín dụng, … Các lớp ẩn là nơi diễn ra phần lớn quá trình tính toán, lớp này sử dụng các hàm toán học phức tạp để xác định các mẫu và mối quan hệ trong dữ liệu. Cuối cùng, lớp đầu ra tạo ra kết quả phân tích của mạng thần kinh nhằm phân loại tín dụng khách hàng và điểm tín dụng [43].

Một lợi thế của việc sử dụng thuật toán NN là khả năng học hỏi và cải thiện theo thời gian. Khi có nhiều dữ liệu hơn được đưa vào thuật toán, nó có thể điều chỉnh trọng số và kết nối để xác định các mẫu tốt hơn và đưa ra dự đoán chính xác hơn. Điều này giúp cải thiện hiệu suất tổng thể của mô hình.

Có nhiều hình trạng NN khác nhau, bao gồm: mạng truyền thẳng (Feed-forward neural network), mạng hồi quy (Recurrent neural network) và mạng tích chập (Convolutional neural networks), mỗi loại được tối ưu hóa cho các tác vụ và loại dữ liệu cụ thể. Trong đề tài nghiên cứu này, feed-forward neural network sẽ được sử dụng để thực nghiệm.

2.4.3.2. Lý thuyết liên quan

2.4.3.2.1. Perceptron

NN được cấu thành bởi các nơron đơn lẻ được gọi là các perceptron. Một perceptron sẽ nhận một hoặc nhiều đầu vào dạng nhị phân $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_N}]$ và cho ra một kết quả nhị phân $\mathbf{o}$ duy nhất. Các đầu vào được điều phối tầm ảnh hưởng bởi các tham số tương ứng là $\mathbf{w}$. Kết quả đầu ra được quyết định dựa vào một ngưỡng quyết định b.

$$\mathbf{o} = \begin{cases} 0 & \sum_i \mathrm{w}_i \mathrm{x}_i \leq b \\ 1 & \sum_i \mathrm{w}_i \mathrm{x}_i > b \end{cases} \qquad \textit{PT 2.7}$$

2.4.3.2.2. Hàm kích hoạt (activation function)

Với đầu vào và đầu ra dạng nhị phân nên rất khó có thể điều chỉnh một lượng nhỏ đầu vào để đầu ra thay đổi. Để trở nên linh động, đầu vào được mở rộng ra khoảng

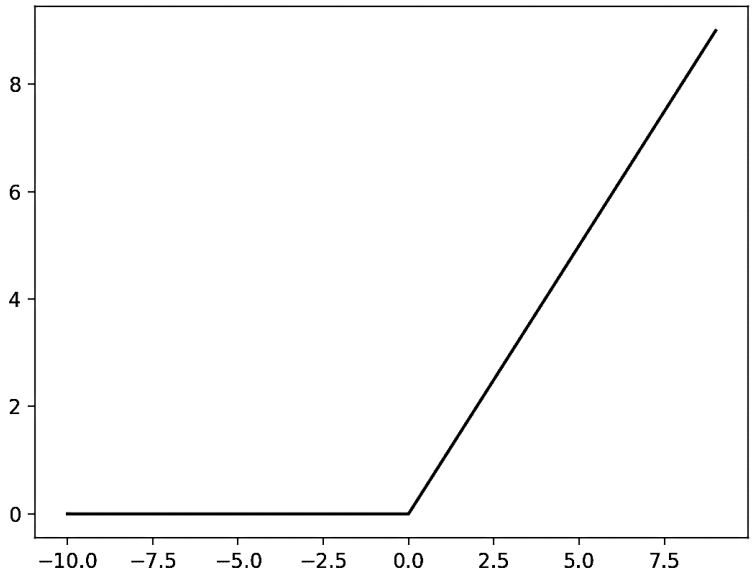
$(0,1)$. Lúc này, đầu vào và đầu ra được quyết định bởi các activation function [44]. Một số hàm kích hoạt xử lý dữ đầu vào và đầu ra như: hàm sigmoid, hàm tanh, hàm ReLU. Hai hàm là hàm sigmoid và hàm tanh đã được trình bày ở phần 2.4.1 với phương trình *PT 2.2* và *PT 2.3*.

Trong NN, hàm kích hoạt ReLU (Rectified Linear Unit) được định nghĩa là phần dương của đối số của nó:

$$f(x) = x^+ = \max(0, x) = \begin{cases} x & \text{nếu } x > 0, \\ 0 & \text{khác} \end{cases} \qquad \textit{PT 2.8}$$

$$f'(x) = \begin{cases} 1 & \text{nếu } x > 0 \\ 0 & \text{nếu } x < 0 \end{cases} \qquad \textit{PT 2.9}$$

Trong đó x là đầu vào của perceptron. Hàm ReLU trả về giá trị đầu vào nếu nó dương và 0 nếu ngược lại [45]. Điều này tạo ra tính phi tuyến tính đơn giản và hiệu quả, cho phép NN mô hình hóa các mối quan hệ phức tạp giữa dữ liệu đầu vào và đầu ra. Một trong những lợi ích chính của hàm kích hoạt ReLU là nó hiệu quả về mặt tính toán, giúp đào tạo và đánh giá NN nhanh hơn.



Hình 2.5 Biểu diễn đồ thị hàm ReLU

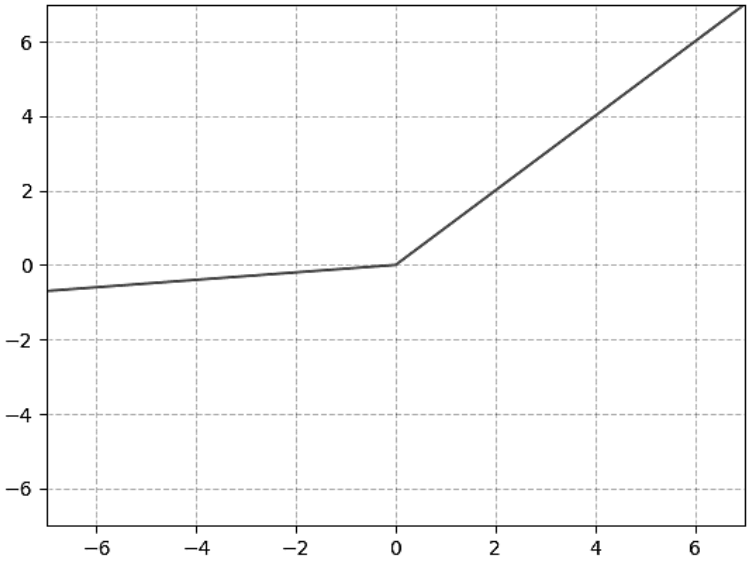
Tuy nhiên, một nhược điểm của hàm kích hoạt ReLU là nó có thể gây ra vấn đề "dying ReLU", trong đó, điểm dữ liệu có giá trị âm thì giá trị của ReLU sẽ bằng 0 và nó sẽ không có đạo hàm tại các điểm có giá trị bằng 0. Để giải quyết vấn đề này, các nhà nghiên cứu đã đề xuất các biến thể của ReLU, chẳng hạn như Leaky ReLU:

$$f(x) = \begin{cases} x & \text{nếu } x > 0, \\ ax & \text{khác} \end{cases} \qquad \textit{PT 2.10}$$

$$f'(x) = \begin{cases} 1 & \text{nếu } x > 0 \\ a & \text{khác} \end{cases} \qquad \textit{PT 2.11}$$

Trong đó a là một hằng số rất nhỏ (Ví dụ: $a = 0.01$) đại diện cho hệ số góc của hàm đối với đầu vào là giá trị âm [46] [47].

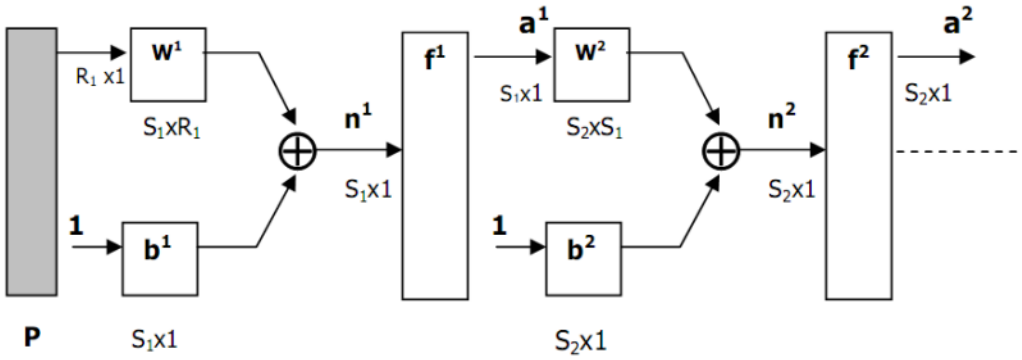
Bằng cách cho phép các giá trị âm, Leaky ReLU giúp ngăn chặn sự cố "dying ReLU" và có thể giúp cải thiện hiệu suất trong NN [48] [49].



Hình 2.6 Biểu diễn đồ thị hàm Leaky ReLU

2.4.3.2.3. Feed-forward neural network

Feed-forward neural network là một loại hình trạng trong NN trong đó thông tin chỉ truyền theo một hướng, từ lớp đầu vào đến lớp đầu ra, không có bất kỳ kết nối vòng lặp hoặc phản hồi nào. Việc xử lý và tính toán dữ liệu có thể mở rộng ra nhiều perceptron, nhưng không có các liên kết phản hồi [50]. Nghĩa là, các liên kết mở rộng từ các đơn vị đầu ra tới các đơn vị đầu vào trong cùng một lớp hay các lớp trước đó là không cho phép.



Hình 2.7 Cấu trúc hoạt động của Feed-forward neural network

Trong đó:

- **P**: Vector đầu vào.
- $\mathbf{W^i}$: Ma trận trọng số của các nơron lớp thứ i.
  ($\mathbf{S^i} \times \mathbf{R^i}$: S hàng - R cột)
- $\mathbf{b^i}$: Vector độ lệch (bias) của lớp thứ i ($\mathbf{S^i} \times 1$: cho **S** nơron).
- $\mathbf{n^i}$: Input ($\mathbf{S^i} \times 1$).
- $\mathbf{f^i}$: Hàm chuyển (Hàm kích hoạt).
- $\mathbf{a^i}$: Output ($\mathbf{S^i} \times 1$).
- $\oplus$: Hàm tổng.

Các nơron đầu vào không thực hiện bất kỳ một tính toán nào trên dữ liệu vào mà tiếp nhận các dữ liệu vào và chuyển cho các lớp kế tiếp. Các nơron ở lớp ẩn và lớp đầu ra mới thực sự thực hiện các tính toán, kết quả được chuyển đổi thông qua hàm chuyển.

Mỗi liên kết gắn với một trọng số, trọng số này được thêm vào trong quá trình dữ liệu đi qua liên kết đó. Các trọng số có thể dương, thể hiện trạng thái kích thích, hoặc âm, thể hiện trạng thái kiềm chế. Mỗi nơron tính toán mức kích hoạt của chúng bằng cách cộng tổng các đầu vào và đưa ra hàm chuyển. Một khi đầu ra của tất cả các nơron trong một lớp mạng cụ thể đã thực hiện xong tính toán thì lớp kế tiếp có thể bắt đầu thực hiện tính toán của mình bởi vì đầu ra của lớp hiện tại tạo ra đầu vào của lớp kế tiếp. Khi tất cả các nơron đã thực hiện tính toán thì kết quả được trả lại bởi các nơron đầu ra.

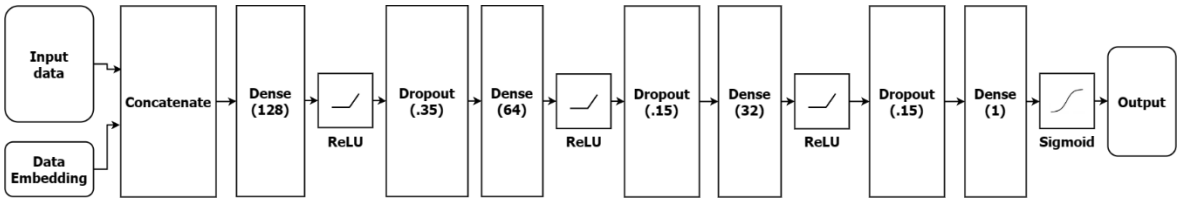
Công thức tính toán cho đầu ra:

$$\boldsymbol{a^2 = f^2\left(W^2\left(f^1(W^1P + b^1)\right) + b^2\right)} \qquad \textit{PT 2.12}$$

Mạng có nhiều lớp có khả năng sẽ có hiệu suất tốt hơn các mạng chỉ có một lớp, chẳng hạn như mạng hai lớp với lớp thứ nhất sử dụng hàm sigmoid và lớp thứ hai dùng hàm đồng nhất (identity function) có thể áp dụng để xấp xỉ các hàm toán học khá tốt, trong khi các mạng chỉ có một lớp thì không có khả năng này [51].

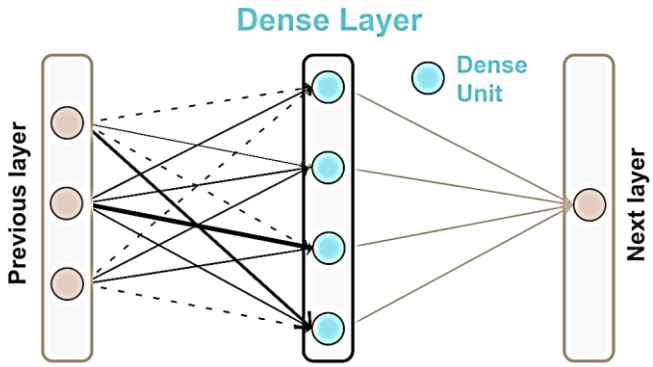
2.4.3.2.4. Kiến trúc mô hình

Trong bài đề tài nghiên cứu này sẽ sử dụng feed-forward neural network để thực nghiệm trên bộ dữ liệu đã được chọn và xử lý. Các thành phần mô hình bao gồm: Đầu vào, lớp nối (concatenate layer), các lớp mạng dày đặc (dense layer), các hàm kích hoạt, các lớp dropout và đầu ra.



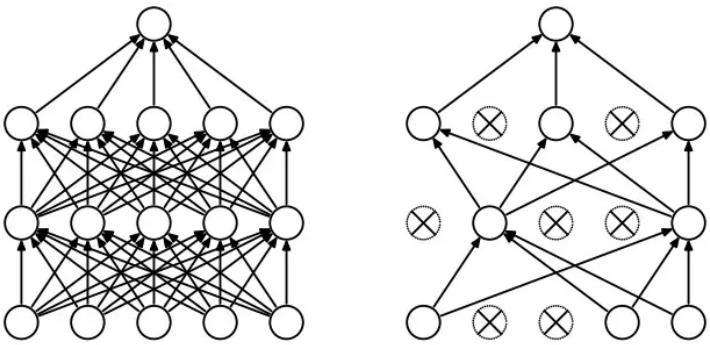
Hình 2.8 Kiến trúc mô hình

Dữ liệu đầu vào bao gồm 2 phần là input data và data embedding với input data là dữ liệu gốc đã qua quá trình xử lý và chuẩn bị dữ liệu, embedding là quá trình chuyển các dữ liệu sau quá trình xử lý có số chiều lớn và thưa thớt qua dạng véc-tơ dày đặc [52]. Hai phần dữ liệu này sẽ được nối lại với nhau bằng lớp nối (concatenate layer) và đưa vào lớp mạng (dense layer). Dense layer bao gồm các nút (node) perceptron nhận và xử lý thông tin từ dữ liệu các lớp trước đó trong mạng với tham số unit là số lượng node và cũng là kích thước không gian đầu ra [53] [54]. Ví dụ, với Dense(128) sẽ có 128 node trong mạng và kích thước không gian đầu ra là 128.



Hình 2.9 Dense layer [55]

Dữ liệu sau khi qua dense layer sẽ vào lớp dropout, lớp này bỏ qua một phần unit bên trong nhằm ngăn chặn tình trạng overfitting với tỷ lệ được định sẵn [56] [57]. Ví dụ, với Dropout(.35) lớp này sẽ bỏ qua ngẫu nhiên 35% unit ở lần cập nhật sau đó.



Hình 2.10 Mạng tiêu chuẩn (phải) và mạng sử dụng dropout (trái) [58]

Dữ liệu sau khi đi qua lớp dense và lớp dropout được chuyển đổi phi tuyến thông qua hàm kích hoạt ReLU và tiếp tục đi vào lớp densen tiếp theo. Khi dữ liệu đi qua lớp dense cuối cùng hàm kích hoạt được sử dụng là Sigmoid nhằm chuyển đổi kết quả đầu ra của mô hình về khoảng (0,1). Đầu ra của mô hình sẽ là giá trị xác suất trong khoảng (0,1).

2.4.4. XGBoost

2.4.4.1. Tổng quan

XGBoost (Extreme Gradient Boosting) [59] là một thuật toán học máy phổ biến được sử dụng rộng rãi cho các bài toán hồi quy và phân loại. Thuật toán sử dụng tăng cường

độ dốc (gradient boosting) và được thiết kế để có hiệu quả cao và có thể mở rộng, làm cho nó trở nên lý tưởng cho các bộ dữ liệu lớn. Gradient boosting là một thuật toán học có giám sát, cố gắng dự đoán chính xác một biến mục tiêu bằng cách kết hợp các ước tính của một tập hợp các mô hình yếu hơn, đơn giản hơn [60].

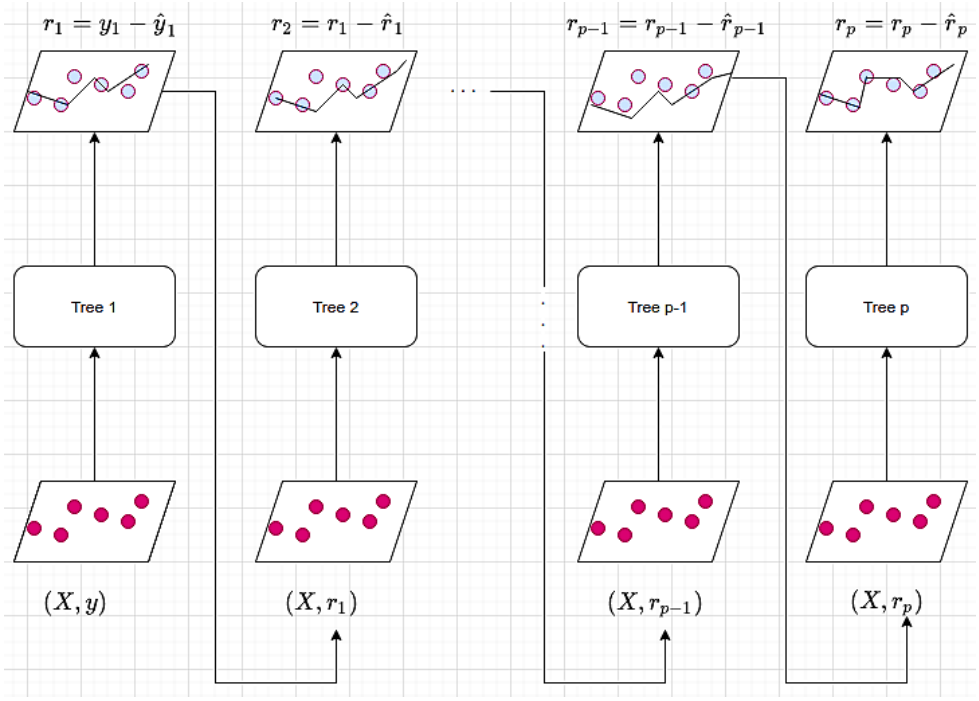
XGBoost hoạt động bằng cách tạo một chuỗi các cây quyết định (decision trees), trong đó mỗi cây tiếp theo được tạo để sửa lỗi của các cây trước đó. Thuật toán có nhiều siêu tham (hyperparameters) số có thể được điều chỉnh để tối ưu hóa hiệu suất của thuật toán, bao gồm tốc độ học, số lượng cây và độ sâu tối đa của mỗi cây.

XGBoost đã trở thành một thuật toán phổ biến trong nhiều lĩnh vực khác nhau, bao gồm tài chính, chăm sóc sức khỏe, tiếp thị, …. Nó đã được sử dụng để giải quyết các vấn đề phức tạp, chẳng hạn như dự đoán các giao dịch gian lận, chẩn đoán tình trạng y tế và xác định khả năng rời bỏ của khách hàng. Tính linh hoạt, độ chính xác và khả năng mở rộng đã khiến XGBoost trở thành một trong những thuật toán học máy được sử dụng rộng rãi nhất hiện nay.

2.4.4.2. Lý thuyết liên quan

XGBoost là thuật toán với mục đích đưa ra một mô hình dự đoán dưới dạng một tập hợp các mô hình dự đoán yếu, thường là các cây quyết định (decision trees) [61] [62]. Thuật toán này được các nghiên cứu đánh giá có hoạt động vượt trội hơn so với thuật toán random forest [61] [62] [63].

Thuật toán không sử dụng sai số của mô hình để tính toán trọng số cho dữ liệu mà sử dụng phần dư. Với mô hình tập hợp từ các mô hình cây quyết định, Mỗi cây quyết định sẽ được thành lập phụ thuộc vào kết quả dự báo của cây quyết định liền trước. Tại một cây quyết định mô hình sẽ tìm cách khớp phần dư từ cây quyết định trước đó.



Hình 2.11 Huấn luyện mô hình XGBoost

Với dữ liệu đầu vào $\mathbf{X}$ và biến mục tiêu là $\boldsymbol{y}$, thuật toán gradient boosting cố gắng tạo ra hàm $\hat{\boldsymbol{f}}(\boldsymbol{x})$ với mục tiêu dự đoán. Tại mô hình thứ $\boldsymbol{b}$ trong chuỗi các mô hình dự đoán hàm mục tiêu tại mô hình đó là $\hat{\boldsymbol{f}}^{\boldsymbol{b}}$. Mô hình tìm cách khớp giá trị phần dư $\mathbf{r^i}$ từ cây quyết định trước là $\hat{\boldsymbol{f}}^{\boldsymbol{b-1}}$. Quá trình được mô tả như sau:

1. Thiết lập hàm mục tiêu $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \mathbf{0}$ và phần dư $\mathbf{r_0} = \boldsymbol{y}$.
2. Lặp lại quá trình huấn luyện cây quyết định theo chuỗi tương ứng với $b = 1,2,3, \dots, B$. Với các bước nhỏ:
    1. Khớp cây quyết định $\hat{\boldsymbol{f}}^{\boldsymbol{b}}$ có độ sâu cây là $\boldsymbol{d}$ trên tập huấn luyện $(\mathbf{X}, \mathbf{r_b})$.
    2. Cập nhật $\boldsymbol{f}$ bằng cách cộng thêm vào giá trị dự báo của một cây quyết định, giá trị này được nhân với hệ số co $\boldsymbol{\lambda}$ (Hệ số này gần giống như learning rate có tác dụng kiểm soát tỷ lệ mà gradient boosting cập nhật phần dư):

$$\hat{\boldsymbol{f}}(\mathbf{x}) = \hat{\boldsymbol{f}}(\mathbf{x}) + \boldsymbol{\lambda}\hat{\boldsymbol{f}}^{\boldsymbol{b}}(\mathbf{x}) \qquad \textit{PT 2.13}$$

3. Cập nhật phần dư cho mô hình:

$$\mathbf{r}_{b+1} \coloneqq \mathbf{r}_b - \boldsymbol{\lambda}\hat{\boldsymbol{f}}^{\boldsymbol{b}}(\mathbf{x}) \qquad \textit{PT 2.14}$$

3. Kết quả dự báo sẽ là kết quả kết hợp từ các mô hình con:

$$\hat{f}(\mathbf{x}) = \sum_{b=1}^{B} \lambda \hat{f}^{b}(\mathbf{x}) \qquad \textit{PT 2.15}$$

XGBoost cải thiện thuật toán tăng cường độ dốc (gradient boosting) về mặt tốc độ và quy mô tính toán [64]. XGBoost sử dụng nhiều nhân CPU để quá trình học có thể diễn ra song song trong lúc đào tạo. XGBoost là một thuật toán tăng cường có thể xử lý các tập dữ liệu mở rộng, khiến nó trở nên hấp dẫn đối với những ứng dụng dữ liệu lớn.

2.4.5. LightGBM

2.4.5.1. Tổng quan

LightGBM là một khung gradient boosting mã nguồn mở sử dụng các thuật toán tree based learning. Nó được thiết kế để hoạt động hiệu quả, có thể mở rộng và chính xác, khiến nó trở thành lựa chọn phổ biến cho nhiều tác vụ học máy, bao gồm phân loại, hồi quy và xếp hạng. LightGBM được thiết kế để hoạt động hiệu quả với những ưu điểm sau [65]:

4. Tốc độ đào tạo nhanh hơn và hiệu quả cao hơn.
5. Sử dụng bộ nhớ thấp hơn.
6. Độ chính xác tốt hơn.
7. Hỗ trợ học song song, phân tán dựa trên quá trình sử dụng GPU.
8. Có khả năng xử lý dữ liệu quy mô lớn.

Mô hình LightGBM dựa trên thuật toán cây quyết định tăng cường độ dốc (gradient boosting decision tree - GBDT), thuật toán này xây dựng một tập hợp các cây quyết định dự đoán biến mục tiêu bằng cách kết hợp các dự đoán của nhiều mô hình yếu hơn. Mỗi cây quyết định được xây dựng bằng cách lặp đi lặp lại quá trình thêm các nhánh mới vào cây hiện có với thuật toán tối ưu hàm mất mát. Thuật toán cũng kết hợp một kỹ thuật gọi là lấy mẫu một phía dựa trên độ dốc (Gradient-based one-side sampling - GOSS), GOSS lấy mẫu dữ liệu bằng cách ưu tiên các trường hợp có độ dốc lớn hơn, giúp giảm chi phí tính toán và cải thiện độ chính xác của mô hình.

2.4.5.2. Lý thuyết liên quan

GBDT là một mô hình tập hợp các cây quyết định được đào tạo theo trình tự. Trong mỗi lần lặp, GBDT học từ các cây quyết định bằng cách khớp các phần dư (lỗi cho đến lần lặp hiện tại). Khi kết thúc vòng lặp huấn luyện đầu tiên sẽ tiếp tục huấn luyện lần 2, trong lần huấn luyện này mô hình sẽ cố gắng tìm hiểu sự khác biệt giữa đầu ra thực tế và tổng trọng số của các dự đoán cho đến lần lặp trước đó. Các lỗi được giảm thiểu bằng cách sử dụng phương pháp gradient.

Điều phức tạp trong mô hình GBDT là quá trình tìm được điểm phân chia tối ưu trong một nhánh của cây quyết định. Các thuật toán tìm kiếm điểm phân chia được sử dụng để tìm một số điểm phân chia tốt. Một trong những thuật toán tìm điểm phân chia phổ biến nhất là thuật toán pre-sorted, thuật toán này liệt kê tất cả các điểm chia có thể có trên các giá trị được sắp xếp trước. Mặc dù pre-sorted giúp quá trình phân chia đơn giản hơn tuy nhiên thuật toán này không mang lại hiệu quả cao về sức mạnh tính toán và bộ nhớ. Phương pháp thứ hai là thuật toán Histogram based. Histogram based nhóm các giá trị thuộc tính liên tục vào các ngăn chứa dữ liệu (bin) riêng biệt để xây dựng biểu đồ thuộc tính trong quá trình đào tạo. Chi phí $\mathbf{O}(\#\mathbf{data} * \#\mathbf{feature})$ để xây dựng biểu đồ và $\mathbf{O}(\#\mathbf{bin} * \#\mathbf{feature})$ để tìm điểm phân tách. Điều này tăng độ phức tạp và quá trình đào tạo sẽ chậm.

LightGBM đã giảm độ phức tạp của thuật toán GBDT với việc xây dựng histogram-based. Histogram-based giảm dữ liệu mẫu và các thuộc tính bằng cách sử

dụng GOSS. Điều này làm giảm độ phức tạp xuống còn $\mathbf{O}(\#\mathbf{data2} * \#\mathbf{feature2})$ trong đó $\#\mathbf{data2} < \#\mathbf{data}$ và $\#\mathbf{feature2} \ll \#\mathbf{feature}$.

Input: I: training data, d : iterations
Input: a : sampling ratio of large gradient data
Input: b : sampling ratio of small gradient data
Input: loss: loss function, L : weak learner

models ← {}, fact ← $\frac{1-a}{b}$
topN ← a × len (I), randN ← b × len (I)

for i = 1 to d do

    preds ← models.predict (I)
    g ← loss (I, preds), w ← {1,1, ... }
    sorted ← GetSortedIndices (abs (g))
    topSet ← sorted[1:topN]
    randSet ← RandomPick(sorted[topN:len(I)],
    randN)
    usedSet ← topSet + randSet
    w[randSet] ×= fact ▷ Assign weight fact to the small gradient data.
    newModel ← L(I [usedSet], −g [usedSet]
    w[usedSet])
    models.append(newModel)

Hình 2.12 Mã minh họa thuật toán GOSS

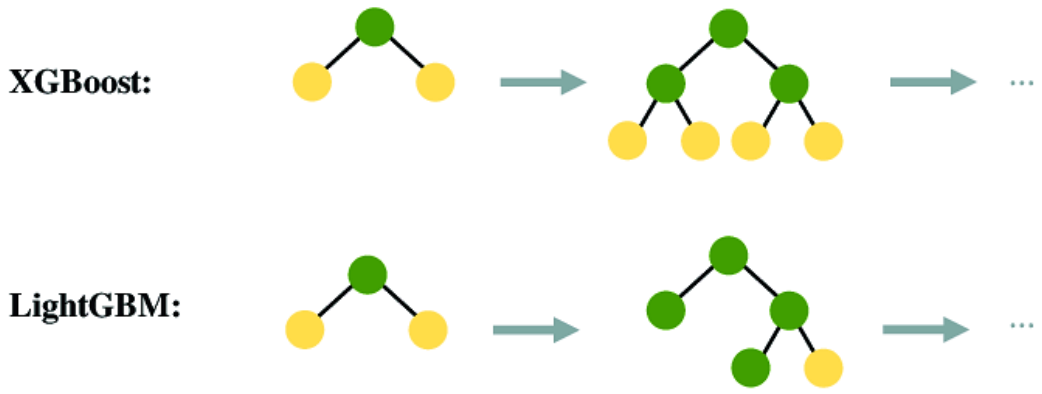
GOSS là một phương pháp lấy mẫu mới giúp lấy mẫu xuống (down samples) dựa trên gradients. GOSS giữ lại các trường hợp phân loại có độ dốc lớn trong khi thực hiện lấy mẫu ngẫu nhiên trên các trường hợp phân loại có độ dốc nhỏ trong mô hình.

Quá trình thực hiện của GOSS bao gồm:

1. Sắp xếp các trường hợp phân loại với thứ tự giảm dần theo gradient.
2. Chon ra (a × 100%) trường hợp được xếp trên cùng.
3. Lấy mẫu ngẫu nhiên các trường hợp (b × 100%) từ phần còn lại của dữ liệu. Điều này sẽ làm giảm sự đóng góp của các mẫu dữ liệu đào tạo tốt theo hệ số b.

Bằng cách sử dụng thuật toán GOSS và GBDT để lấy mẫu dữ liệu và đào tạo LightGBM đã giúp cho mô hình dữ liệu tăng tốc độ đào tạo nhưng vẫn giữ nguyên được chất lượng đầu ra [66] [67].

Ngoài ra một điều giúp cho LightGBM có tốc độ đào tạo nhanh là do LightGBM thực hiện phân tách các nốt lá theo chiều dọc (leaf-wise) dẫn đến giảm thiểu tổn thất chi phí hơn các thuật toán phân chia nốt lá theo cấp độ (level-wise), chẳng hạn như thuật toán XGBoost thực hiện phân chia các nốt lá theo level-wise [68].



Hình 2.13 Quá trình phân chia các nốt trong hai mô hình thuật toán XGBoost và LightGBM

Từ những yếu tố trên đã giúp LightGBM có tốc độ đào tạo nhanh nhưng vẫn giữ được chất lượng của mô hình đào tạo so với các thuật toán gradient boosting khác. Nhưng điều này cũng có thể làm cho mô hình gặp vấn đề overfitting tuy nhiên vấn đề này có thể được giải quyết bằng việc tối ưu độ sâu lớn nhất (max-depth) của mô hình [69].

2.4.6. CatBoost

2.4.6.1. Tổng quan

CatBoost [70] là một khung gradient boosting mã nguồn mở phát triển bởi Yandex [71]. CatBoost được tạo ra với mục tiêu cung cấp thuật toán học máy hiệu suất cao, dễ sử dụng và có thể hoạt động với các loại dữ liệu khác nhau, bao gồm dữ liệu số, dữ liệu phân loại, văn bản, …. CatBoost được ứng dụng nhiều vào các bài toán như phân loại hình ảnh, xử lý ngôn ngữ tự nhiên và hệ thống đề xuất.

Một trong những tính năng chính của CatBoost là nó có thể xử lý rất tốt các thuộc tính phân loại (categorical feature) mà không yêu cầu mã hóa trước. Điều này đạt được thông qua một thuật toán mới sử dụng kết hợp thuật toán gradient boosting và mã hóa phân loại (categorical encoder). Cách tiếp cận này cho phép CatBoost tự động phát hiện và xử lý dữ liệu phân loại, giúp làm việc dễ dàng hơn nhiều và giảm nhu cầu về trích xuất đặc trưng.

CatBoost cũng có một số tính năng độc đáo khác khiến nó khác biệt với các thuật toán gradient boosting khác. Ví dụ: nó sử dụng cấu trúc cây đối xứng (symmetric tree) thay vì cây nhị phân (binary tree) giúp giảm tình trạng overfitting và cải thiện khả năng tổng hợp thông tin từ mô hình. Ngoài ra, CatBoost cũng sử dụng thêm phương pháp tăng cường theo thứ tự (ordered boosting), phương pháp này ưu tiên các trường hợp khó phân loại hơn, từ đó tối ưu độ chính xác đối với các bài toán phân loại khó [72].

Về hiệu suất, CatBoost được biết đến với tốc độ và khả năng mở rộng. Thuật toán này được thiết kế để hoạt động với các bộ dữ liệu lớn và có thể sử dụng hiệu quả các CPU và GPU đa lõi [73]. Ngoài ra, CatBoost có thể đưa ra các thuộc tính dữ liệu quan trọng của mô hình đào tạo để có thể giúp xác định những thuộc tính dữ liệu nào là quan trọng nhất để đưa ra dự đoán.

2.4.6.2. Lý thuyết liên quan

2.4.6.2.1. Thuật toán kết hợp gradient boosting và categorical encode

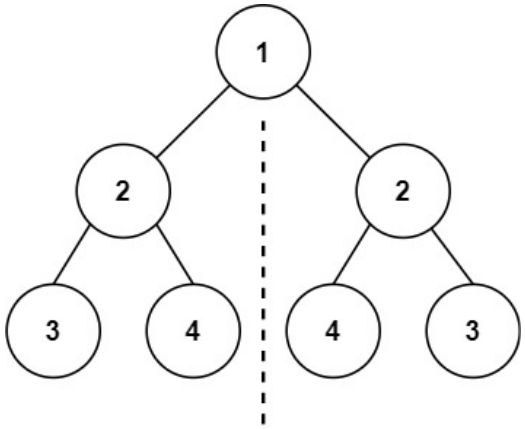
Một cách tiếp cận phổ biến để xử lý dữ liệu phân loại là mã hóa one-hot, trong đó mỗi danh mục dữ liệu đầu vào được chuyển đổi thành một vectơ nhị phân cho biết sự hiện diện của danh mục dữ liệu đó. Tuy nhiên, mã hóa one-hot có thể dẫn đến quá tải về số lượng thuộc tính mới sau quá trình mã hóa, khiến mô hình trở nên tốn kém về mặt tính toán và sử dụng nhiều bộ nhớ, đặc biệt đối với các bộ dữ liệu có nhiều thuộc tính phân loại.

Để giải quyết vấn đề này, CatBoost sử dụng một thuật toán mới kết hợp gradient boosting với categorical encode. Thuật toán này hoạt động bằng cách nắm bắt mối quan hệ giữa các biến phân loại với biến mục tiêu. CatBoost chuyển đổi từng giá trị của biến phân loại thành các giá trị số, giá trị này sẽ dựa trên thứ hạng của biến đối với biến mục tiêu. Quá trình mã hóa được học lặp đi lặp lại trong suốt quá trình huấn luyện nhằm tăng chất lượng mã hóa [73].

2.4.6.2.2. Symmetric tree

Trong các thuật toán gradient boosting truyền thống, cây quyết định được xây dựng ở định dạng nhị phân (binary), trong đó mỗi nốt bên trong cây có hai nhánh tương ứng với hai quyết định phân loại. Tuy nhiên, cấu trúc binary tree có thể khiến mô hình gặp vấn đề overfitting.

Để giải quyết vấn đề này, CatBoost sử dụng cấu trúc symmetric tree, trong đó mỗi nốt bên trong cây có một số nhánh cố định tương ứng với một quyết định phân loại. Cách tiếp cận này cho phép cấu trúc cây cân bằng và đối xứng hơn và có thể mô hình tránh tình trạng overfitting và cải thiện hiệu suất tổng quát của mô hình [74] [75].



Hình 2.14 Cấu trúc symmetric tree [76]

Trong đó mỗi nốt bên cấu trúc symmetric tree sẽ biểu thị một quyết định phân loại từng phần dựa trên một tập hợp con các thuộc tính, sau đó chọn phần phân tách tốt nhất trong số tất cả các phần tách có thể bằng việc sử dụng thuật toán tham lam (greedy algorithm) và tối ưu hàm mất mát tại mỗi nút, từ đó đưa ra mẫu phân loại [74] [75].

2.4.6.2.3. Ordered boosting

Ordered boosting là phương pháp được CatBoost sử dụng nhằm giúp quá trình đào tạo dễ dàng hơn trên các dữ liệu khó phân loại. Cách tiếp cận này giúp cải thiện độ chính xác và hiệu suất phân loại của mô hình, bằng cách tập trung vào các thuộc tính trọng nhất để cải thiện hiệu suất tổng thể.

Ordered boosting gán trọng số cho từng mẫu đào tạo, dựa trên độ khó hoặc mức độ quan trọng của nó. Các trọng số sau đó được sử dụng trong quá trình huấn luyện, để giúp mô hình tập trung hơn hơn vào các mẫu đào tạo khó. Các trọng số được cập nhật ở mỗi lần lặp của quá trình gradient boosting dựa trên khoảng lỗi do các lần lặp trước đó [77] [78].

Ưu điểm của ordered boosting là nó giúp mô hình giảm tình trạng overfitting. Bằng cách gán thêm trọng số cho các mẫu đào tạo khó, từ đó mô hình sẽ khái quát được toàn bộ dữ liệu và giúp dự đoán tốt hơn với các dữ liệu mới [77].

2.5. Kỹ thuật và phương pháp sử dụng trong nghiên cứu và thực nghiệm

2.5.1. Feature engineering

Feature engineering là một kỹ thuật học máy tận dụng dữ liệu để tạo các biến mới không có trong tập huấn luyện. Nó có thể tạo ra các tính năng mới cho cả mô hình học có giám sát và không giám sát, với mục tiêu đơn giản hóa và tăng tốc độ chuyển đổi dữ liệu, đồng thời nâng cao độ chính xác của mô hình [79].

Các kỹ thuật trong feature engineering được sử dụng trong quá trình nghiên cứu và thực nghiệm bao gồm: Imputation, Feature extraction, One-hot encoding, Feature scaling, Combine.

2.5.1.1. Imputation

Imputation là quá trình điền vào các giá trị còn thiếu trong tập dữ liệu. Có nhiều cách để điền dữ liệu bị thiếu, có thể điền giá trị trung bình, trung vị hoặc sử dụng các kỹ thuật imputation nâng cao như K-nearest neighbor (KNN) hoặc hồi quy (regression-based). Kỹ thuật imputation ngăn ngừa việc mất mát dữ dữ liệu, giúp cho mô hình huấn luyện tiếp nhận được đầy đủ thông tin trong quá trình huấn luyện từ đó cải thiện chất lượng huấn luyện của mô hình.

2.5.1.2. Feature extraction

Feature extraction sử dụng các thuộc tính hiện có trong dữ liệu thô để tạo các thuộc tính mới cho quá trình huấn luyện mô hình. Điều này liên quan đến việc chọn các thông tin có liên quan từ dữ liệu thô và chuyển đổi thành một số biểu diễn hoặc thông tin mới có ý nghĩa hơn để mô hình nắm bắt được tổng quát hơn về dữ liệu. Ví dụ như trích xuất "tỷ lệ khoản vay" từ hai thuộc tính là "khoản vay tín dụng yêu cầu" và "khoản vay tín dụng được cấp".

Để định lượng khả năng dự đoán của một thuộc tính trong quá trình trích xuất đặc trưng thì trọng số dấu hiệu (weight of evidence – WOE) và chỉ số giá trị thông tin (information value - IV) được sử dụng nhằm tính toán lượng thông tin mà một thuộc tính cung cấp về biến mục tiêu trong tập dữ liệu [80] [81] [82].

WOE (weight of evidence) là một trong những kĩ thuật feature engineering được áp dụng trong việc đánh giá chất lượng của thuộc tính dữ liệu. Trong mô hình dự đoán rủi ro tín dụng, WOE được tính toán dựa trên tỷ lệ khách hàng xấu (khách hàng không trả hoặc trả chậm khoản vay) và khách hàng tốt (khách hàng hoàn trả được khoản vay):

$$\mathbf{WOE} = \mathbf{ln}\left(\frac{\%\ \text{of non} - \text{events}}{\%\ \text{of events}}\right) \qquad PT\ 2.16$$

Với "% of non − events" là phân phối phi sự kiện trong dữ liệu (với dữ liệu cho mô hình rủi ro tín dụng sẽ là phân phối khách hàng tốt), "% of events" là phân phối của các sự kiện trong dữ liệu (khách hàng xấu).

Các bước tính chỉ số WOE:

- Bước 1: Đối với biến liên tục sẽ chia dữ liệu thành k nhóm (bin). Đối với các biến phân loại sẽ bỏ qua bước này.
- Bước 2: Tính số lượng "events" và "non − events" trong mỗi bin.
- Bước 3: Tính "% of non − events" và "% of events" trên trong mỗi nhóm.
- Bước 4: Tính chỉ số WOE bằng công thức ở phương trình *PT 2.16*

Các quy tắc liên quan:

- Số lượng mẫu trong bin phải lớn hơn 5% trên tổng lượng mẫu dữ liệu.
- Số lượng "events" và "non − events" phải khác nhau trong trong mỗi bin.

Từ chỉ số WOE ta có thể tính được chỉ số IV. IV là chỉ số rất hữu ích trong việc chọn lọc các biến quan trọng trong mô hình dự đoán. IV được tính theo công thức:

$$\mathbf{IV} = \sum\left(\frac{\%\ \text{of non} - \text{events}}{\%\ \text{of events}}\right) \times \mathbf{WOE} \qquad PT\ 2.17$$

Bảng 2.1 Bảng phân chia giá trị thông tin của các thuộc tính dựa trên khoảng giá trị của chỉ số IV

Giá trị thông tin (IV)	**Khả năng dự đoán của thuộc tính**
Nhỏ hơn 0.02	Thuộc tính không hữu ích cho mô hình
0.02 – 0.1	Khả năng dự đoán yếu
0.1 – 0.3	Khả năng dự đoán trung bình
0.3 – 0.5	Khả năng dự đoán mạnh
Lớn hơn 0.5	Thuộc tính đáng ngờ (Cần kiểm tra)

Chỉ số trọng số dấu hiệu (WOE) và chỉ số giá trị thông tin (IV) là thước đo hữu ích trong việc lựa chọn và trích xuất đặc trưng. Những chỉ số này cung cấp thước đo định lượng về khả năng dự đoán thuộc tính dữ liệu. Bằng cách chọn các tính năng có giá trị thông tin cao nhất, có thể xây dựng các mô hình dự đoán chính xác và hiệu quả hơn.

2.5.2. Gradient descent

Gradient descent là một thuật toán tối ưu được sử dụng để giảm thiểu hàm mất mát (loss function) trong các mô hình học máy. Mục tiêu của thuật toán này là tìm giá trị của các tham số mô hình mà ở đó sẽ tối thiểu được loss function, bằng cách điều chỉnh lặp đi lặp lại các giá trị tham số theo hướng dốc nhất của đạo hàm loss function [83].

Giả sử, trong trường hợp mô hình logistic regression, loss function của mô hình logistic regression sẽ là:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{L_{CE}}\left(f\left(x^{(i)};\theta\right), y^{(i)}\right)$$

PT 2.18

$$\mathrm{L_{CE}}\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) = \begin{cases} -\log\left(f\left(x^{(i)};\theta\right)\right) & \text{nếu } y^{(i)} = 1 \\ -\log\left(1 - f\left(x^{(i)};\theta\right)\right) & \text{nếu } y^{(i)} = 0 \end{cases}$$

Mục tiêu của gradient descent là tìm được bộ tham số θ để tối ưu loss function:

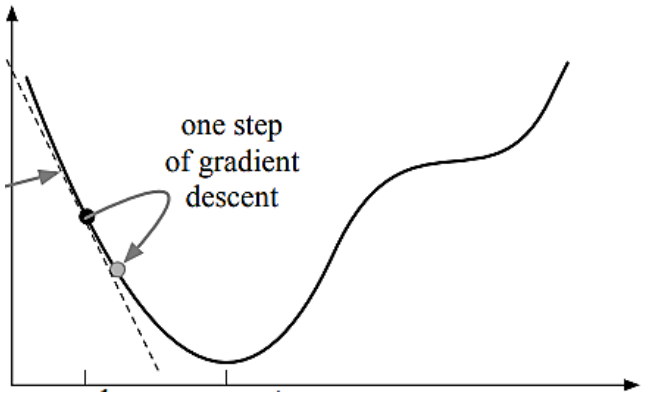
$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \mathrm{L_{CE}}\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right) \quad \textit{PT 2.19}$$

Thuật toán Gradient descent bắt đầu với việc lấy ngẫu nhiên cho giá trị tham số θ và lặp lại cập nhật các giá trị tham số theo hướng đạo hàm mang giá trị âm của loss function, quá trình giảm này sử dụng thêm chỉ số α được gọi là tốc độ học (learning rate) nhằm tăng tốc độ giảm. Công thức để cập nhật các tham số:

$$\theta_i = \theta_i - \alpha \frac{\partial J(\theta)}{\partial \theta_i} \quad \textit{PT 2.20}$$

Trong đó:

- θ_i đại diện cho tham số thứ j được cập nhật.
- $\frac{\partial J(\theta)}{\partial \theta_i}$ là đạo hàm riêng (gradient - ∇) của loss function đối với tham số θ_j.



Hình 2.15 Minh họa một bước giảm của loss function [84]

Các bước tối ưu tham số θ của thuật toán gradient descent [84]:

1. Lấy ngẫu nhiên tham số θ_0 khi đó $\theta = \theta_0$.
2. Quá trình lặp bao gồm:
    1. Xác định $\nabla\theta_i = \nabla_\theta \mathrm{L_{CE}}\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right)$.

2. Cập nhật $\theta_{i+1} = \theta_i - \alpha \nabla\theta_i$.

3. Dừng vòng lặp khi giá trị được cập nhật không còn thay đổi hoặc khoảng cách giữa 2 giá trị liên tiếp đủ nhỏ khi đó:

$$\|\theta_{i-1} - \theta_i\| \leq \varepsilon \qquad \textit{PT 2.21}$$

Sau quá trình thực hiện thuật toán gradient descent sẽ thu được bộ tham số θ mà tại đó loss function đạt giá trị tối ưu nhất.

2.5.3. Cross-validation

Cross-validation (CV) là một kỹ thuật được sử dụng trong học máy để đánh giá hiệu suất của mô hình dự đoán. Kỹ thuật này liên quan đến việc phân vùng tập dữ liệu thành nhiều tập hợp con, đào tạo mô hình trên một số tập hợp con và đánh giá mô hình trên tập hợp con còn lại. Cross-validation được sử dụng để ước tính hiệu suất của một mô hình và tránh tình trạng mô hình gặp vấn đề overfitting [85].

Kỹ thuật này bao gồm các bước:

1. Xáo trộn dataset một cách ngẫu nhiên.
2. Chia dataset thành **k** nhóm (fold).
3. Với mỗi fold:
    1. Sử dụng $\mathbf{k-1}$ fold hiện tại để đánh giá hiệu quả mô hình.
    2. Fold còn lại được sử dụng để huấn luyện mô hình.
    3. Huấn luyện mô hình.
    4. Đánh giá và sau đó hủy mô hình.
4. Tổng hợp hiệu quả của mô hình dựa trên các số liệu đánh giá.

Cách tiếp cận này liên quan đến việc chia ngẫu nhiên tập hợp dữ liệu thành **k** fold có kích thước xấp xỉ bằng nhau. Fold đầu tiên được sử dụng làm tập đánh giá và huấn luyện mô hình với $\mathbf{k-1}$ fold còn lại [86].

Kết quả đánh giá của quá trình sử dụng CV với k-fold để huấn luyện mô hình sẽ sử dụng giá trị trung bình và độ lệch chuẩn của các chỉ số đánh giá hiệu suất mô hình

phân loại. Ví dụ, mô hình sử dụng chỉ số AUC ROC để đánh giá hiệu suất, sau quá trình huấn luyện mô hình sử dụng kỹ thuật CV sẽ thu được trung bình và độ lệch chuẩn AUC trên tất cả các vòng lặp huấn luyện từ đó đánh giá được hiệu suất dự đoán tổng thể của mô hình.

Trong bài nghiên cứu này sẽ tính toán trung bình và độ lệch chuẩn của chỉ số AUC sau quá trình huấn luyện mô hình bằng kỹ thuật CV.

Công thức tính trung bình và độ lệch chuẩn của chỉ số AUC được đưa ra như sau:

$$\text{mean}_{\text{AUC}} = \frac{1}{\text{n}}\sum_{i=1}^{n} \text{A}_i \qquad \textit{PT 2.22}$$

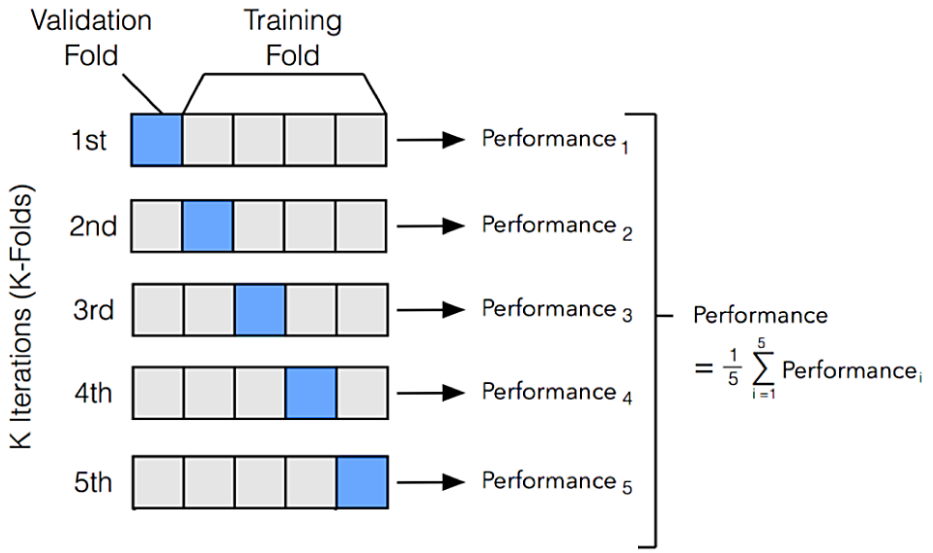
Trong đó:

- mean_{AUC}: Trung bình AUC của tất cả vòng lặp huấn luyện.
- n: Số lượng fold
- A_i: Chỉ số AUC của mỗi vòng lặp huấn luyện

$$\text{std}_{\text{AUC}} = \sqrt{\frac{\sum_{i=1}^{n}(\text{A}_i - \text{mean}_{\text{AUC}})^2}{\text{n}}} \qquad \textit{PT 2.23}$$

Trong đó:

- std_{AUC}: Độ lệch chuẩn AUC tính trên tất cả vòng lặp huấn luyện.
- n: Số lượng fold.
- A_i: Chỉ số AUC của mỗi vòng lặp huấn luyện.
- mean_{AUC}: Trung bình AUC của tất cả vòng lặp huấn luyện.



Hình 2.16 Mô phỏng cấu trúc huấn luyện mô hình sử dụng kỹ thuật CV

Có một số cách để lựa chọn số fold cho quá trình huấn luyện:

- Chọn **k** sao cho mỗi nhóm có dữ liệu đào tạo/đánh giá đủ lớn để mang tính đại diện thống kê cho tập dữ liệu chung.
- Giá trị của **k** được cố định là 10 hoặc 5. Khi **k** càng lớn, sự khác biệt về kích thước giữa tập huấn luyện và tập đánh giá càng nhỏ. Khi sự khác biệt này tăng lên thì độ lệch và phương sai sẽ tăng theo [87] [88].

Sau quá trình huấn luyện mô hình với kỹ thuật CV nếu mô hình đạt được kết quả kỳ vọng sẽ có 2 cách chính để tạo ra mô hình cuối cùng:

1. Trong quá trình huấn luyện các fold, sẽ lưu lại mô hình, sau đó chọn ra mô hình tốt nhất. Cách này có ưu điểm là không cần huấn luyện lại mô hình, tuy nhiên, lại có nhược điểm là mô hình sẽ không bao quát được tất cả dữ liệu và có thể không dự đoán tốt với các dữ liệu trong thực tế.
2. Huấn luyện lại mô hình với toàn bộ dữ liệu, sau đó lưu lại mô hình và dự đoán với tập kiểm thử để kiểm tra kết quả.

Trong bài nghiên cứu này, các mô hình được huấn luyện với kỹ thuật CV sẽ sử dụng cách 2 để đánh giá mô hình và sử dụng.

CV là một kỹ thuật mạnh mẽ để đánh giá hiệu suất của các mô hình dự đoán và giảm vấn đề overfitting. Tuy nhiên, điều quan trọng là kỹ thuật CV chỉ đáng tin cậy khi tập huấn luyện và tập kiểm thử đại diện cho toàn bộ tập dữ liệu [89]. Bằng cách phân chia dữ liệu thành nhiều tập hợp con, CV cung cấp ước lượng rất tốt về hiệu suất tổng quát của mô hình.

2.5.4. Phương pháp học đồng bộ

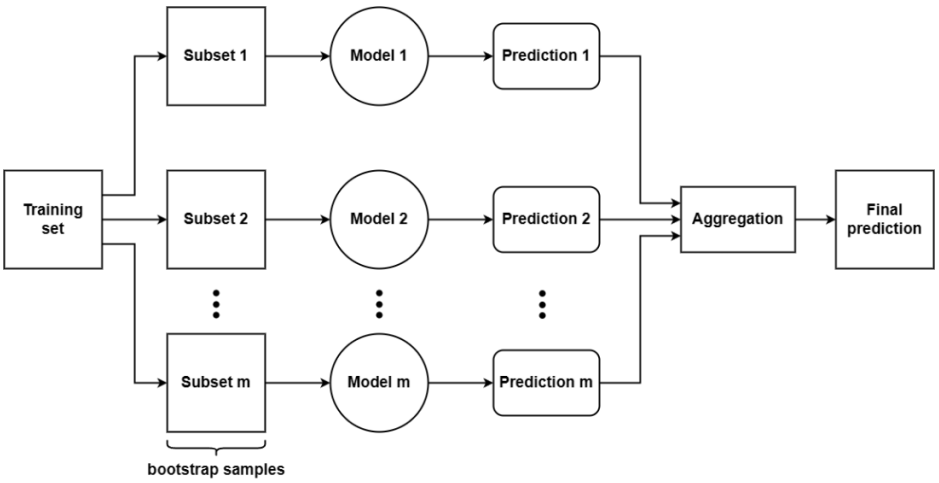
Phương pháp học đồng bộ (Ensemble learning) là phương pháp tổng hợp kết quả dự đoán của nhiều mô hình riêng lẻ để tạo ra kết quả mới đạt được hiệu suất dự đoán tốt hơn so với bất kỳ mô hình đơn lẻ nào khác [90] [91].

Ý tưởng cơ bản của phương pháp ensemble learning là huấn luyện nhiều mô hình độc lập trên cùng một bộ dữ liệu và sau đó kết hợp các dự đoán của mô hình để tạo ra dự đoán cuối cùng. Ensemble learning sẽ tận dụng được điểm mạnh của các mô hình riêng lẻ đồng thời giảm thiểu các điểm yếu của những mô hình riêng lẻ này. Bằng cách kết hợp nhiều mô hình, phương pháp ensemble learning có thể nắm bắt các khía cạnh khác nhau của dữ liệu, tăng tính ổn định của dự đoán và giảm tác động của các dữ liệu nhiễu hoặc ngoại lệ, từ đó giúp giảm nguy cơ gặp vấn đề overfitting và tăng khả năng khái quát thông tin dữ liệu [92].

Phương pháp ensemble learning kết hợp các mô hình độc lập bằng nhiều cách, với mỗi cách sẽ nhằm phục vụ cho những vấn đề khác nhau trong xây dựng và huấn luyện mô hình học máy. Tuy nhiên, mục tiêu cuối cùng đều nhằm tạo ra kết quả mới tốt hơn các kết quả của các mô hình độc lập. Một số phương pháp ensemble learning phổ biến thường được sử dụng:

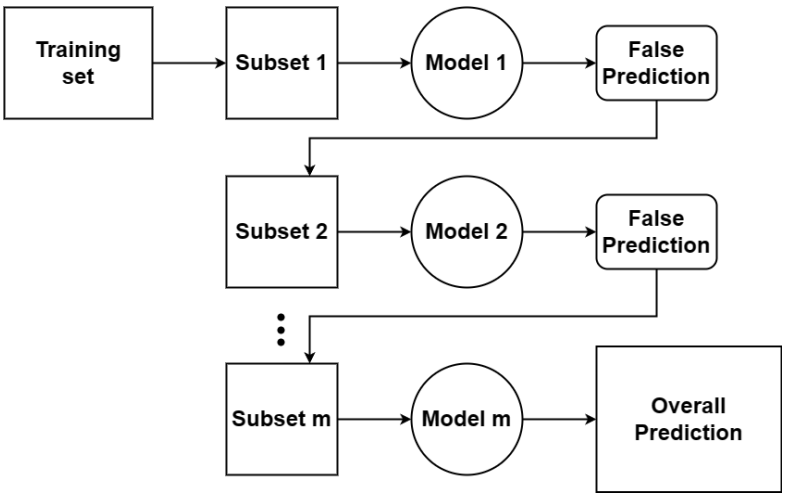
- Bagging: Phương pháp này xây dựng một lượng lớn mô hình (thường sẽ cùng loại với nhau) trên những mẫu dữ liệu nhỏ từ tập dữ liệu huấn luyện. Những mô hình này sẽ được huấn luyện độc lập và song song với nhau. Dự đoán của

các mô hình sẽ được tổng hợp và đưa ra kết quả dựa trên các hàm tổng hợp (chẳng hạn như hàm trung bình cộng).



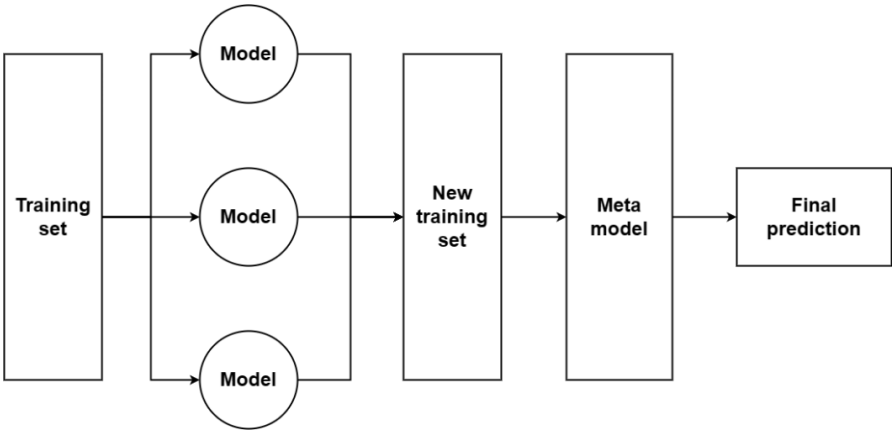
Hình 2.17 Quá trình xây dựng mô hình và tạo kết quả dựa trên phương pháp Bagging

- Boosting: Phương pháp này xây dựng một lượng lớn mô hình (thường sẽ cùng loại với nhau) mỗi mô hình sẽ được kế thừa từ mô hình đã xây dựng trước và sẽ tập trung vào các mẫu dự đoán sai từ các mô hình trước đó, từ đó học cách sửa lỗi. Phương pháp này sẽ tạo một chuỗi các mô hình, trong đó mô hình sau sẽ tốt hơn mô hình trước bởi quá trình cập nhật trọng số thông qua mỗi lần huấn luyện (trọng số của những mẫu dự đoán đúng sẽ không đổi, còn trọng số của những mẫu dự đoán sai sẽ tăng thêm). Kết quả của phương thức này là kết quả dự đoán của mô hình cuối cùng trong chuỗi.



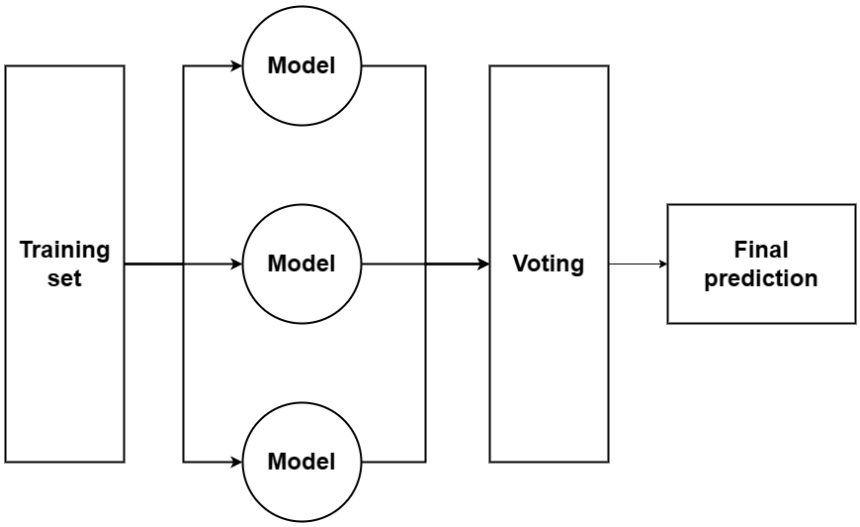
Hình 2.18 Quá trình xây dựng mô hình và tạo kết quả dựa trên phương pháp Boosting

- Stacking: Phương pháp này xây dựng một số mô hình độc lập (thường sẽ khác loại) và một mô hình thay thế (meta-model), sau đó meta-model sẽ học cách kết hợp các kết quả dự đoán của các mô hình độc lập một cách tốt nhất. Quá trình huấn luyện những mô hình độc lập sẽ dựa trên toàn bộ dữ liệu huấn luyện (khác với phương pháp bagging sẽ chia nhỏ dữ liệu).



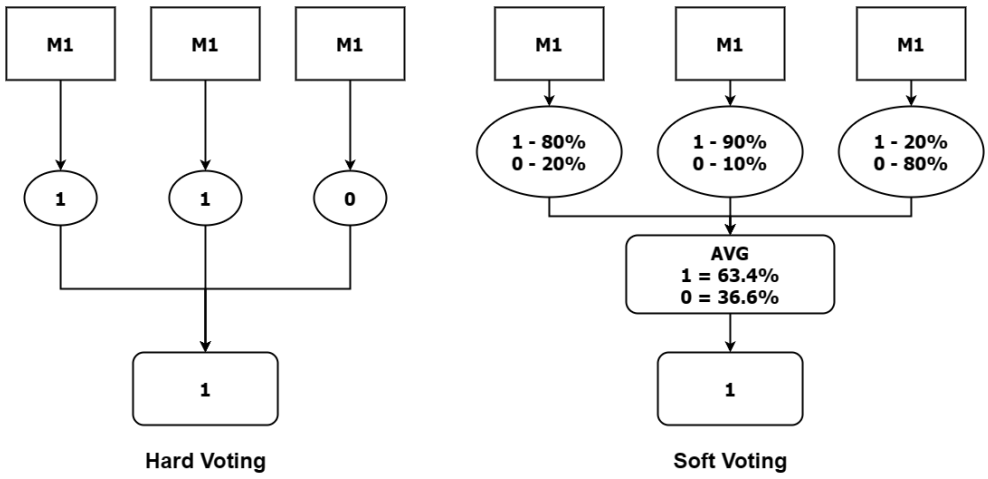
Hình 2.19 Quá trình xây dựng mô hình và tạo kết quả dựa trên phương pháp Stacking

- Voting: Phương pháp này tương tự với phương pháp stacking, tuy nhiên, quá trình thực nghiệm thay vì sử dụng meta-model để học cách kết hợp các dự đoán thì sẽ sử dụng mô hình biểu quyết (voting model), đây là một biến thể của phương pháp stacking trong đó voting model sẽ sử dụng các phương pháp thống kê để đưa ra kết quả từ các mô hình huấn luyện độc lập.



Hình 2.20 Quá trình xây dựng mô hình và tạo kết quả dựa trên phương pháp Voting

Voting model sẽ sử dụng hard-voting (lớp được dự đoán nhiều nhất) hoặc soft-voting (lớp có tổng xác suất được dự đoán là cao nhất).



Hình 2.21 Ví dụ về 2 phương thức hard-voting (trái) và soft-voting (phải)

Trong hình 2.20, phương thức hard-voting lấy ra kết quả từ lớp được dự đoán nhiều nhất từ các mô hình độc lập, khi đó:

$$\hat{y} = \mathbf{mode}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n) \qquad \textit{PT 2.24}$$

Trong đó:

- $\hat{y}$: Lớp phân loại được dự đoán.
- $(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$: Lớp phân loại được dự đoán từ các mô hình độc lập.

Với phương pháp soft-voting, kết quả cuối cùng thu được bằng cách lấy trung bình trọng số của các giá trị xác suất dự đoán từ các các mô hình độc lập:

$$\hat{y} = \underset{i}{\mathbf{argmax}} \frac{1}{\mathrm{N}} \sum_{j=1}^{m} \mathrm{P}(C_j(x) = i) \qquad \textit{PT 2.25}$$

Trong đó:

- $\hat{y}$: Lớp phân loại được dự đoán.
- N: Số lớp của mô hình dự đoán.

- P($\hat{y}_j = i$): Xác suất dự đoán lớp i của mô hình độc lập j

Trong quá trình thực nghiệm, sau khi hoàn thành huấn luyện và đánh giá các mô hình độc lập sẽ sử dụng phương pháp voting để kết hợp một số mô hình có hiệu suất tốt từ đó đưa ra kết quả dự đoán mới và đánh giá quá trình thực nghiệm với phương pháp voting dựa trên kết quả có được.

2.5.5. Phương pháp đánh giá mô hình

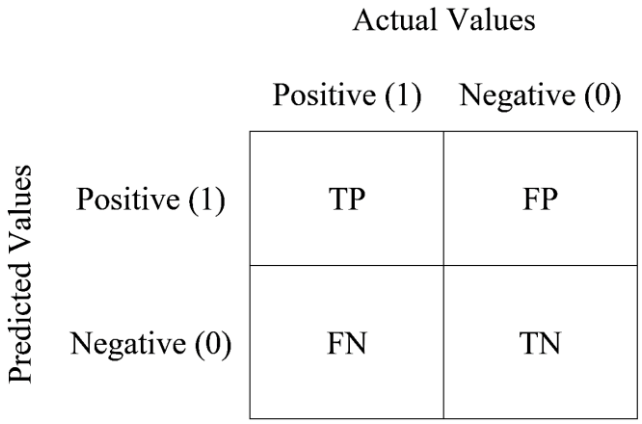
2.5.5.1. Confusion matrix

Trong học máy, các mô hình phân loại được sử dụng để phân chia dữ liệu thành các danh mục khác nhau. Theo Jianfeng và các cộng sự [93], Confusion matrix là phương pháp để tóm tắt hiệu suất của mô hình phân loại. Nó là một công cụ cần thiết để đánh giá độ chính xác của một mô hình và xác định mức độ phân loại dữ liệu, đặc biệt là các mô hình đánh giá rủi ro tín dụng khi mà các dữ liệu thường có xu hướng mất cân bằng rất lớn [94] [95] khiến cho kết quả của chỉ số đánh giá độ chính xác phân loại (classification accuracy) không còn đánh giá đúng chất lượng của mô hình huấn luyện [96].

Giả sử, với mô hình phân loại khách hàng rủi ro, nếu xem khách hàng rủi ro là "positive" và khách hàng không có rủi ro được xem là "negative". Quá trình xây dựng confusion matrix cho mô hình phân loại 2 lớp bao gồm các bước sau:

1. Tạo tập dữ liệu kiểm thử với các biến mục tiêu phân loại ứng với các mẫu dữ liệu.
2. Đưa ra dự đoán cho từng mẫu dữ liệu trong tập dữ liệu kiểm thử bằng đầu ra của mô hình phân loại đã được huấn luyện.
3. Từ kết quả thật và kết quả dự đoán của các dữ liệu trong dữ liệu kiểm thử.
4. Đưa ra số lượng cho các lớp TP, TN, FP, FN. Trong đó:
    - TP (True Positive): Số lượng mẫu dữ liệu được mô hình dự đoán là "positive" và thực tế nhãn của mẫu dữ liệu là "positive".
    - TN (True Negative) : Số lượng mẫu dữ liệu được mô hình dự đoán là "negative" và thực tế nhãn của mẫu dữ liệu là "negative".

- FP (False Positive): Số lượng mẫu dữ liệu được mô hình dự đoán là "positive" tuy nhiên nhãn thực tế của mẫu là "negative". (Những trường hợp dự đoán này được phân loại là sai lầm loại I).
- FN (False Negative): Số lượng mẫu dữ liệu được mô hình dự đoán là "negative" tuy nhiên nhãn thực tế của mẫu là "positive" (Những trường hợp dự đoán này được phân loại là sai lầm loại II).



Hình 2.22 Confusion Matrix

Để biết mức độ chính xác của mô hình sẽ cần thêm một số chỉ số nhằm xác định hiệu suất phân loại của mô hình thông qua các chỉ số TP, TN, FP, FN. Các chỉ số đánh giá liên quan đến confusion matrix:

Recall (Sensitivity - TPR): Tỷ lệ của các kết quả dự đoán là "positive" trên tổng số mẫu "positive" thực tế. Trong mô hình phân loại sẽ kỳ vọng chỉ số này lớn nhất.

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{PT 2.26}$$

Precision: Tỷ lệ của các kết quả dự đoán "positive" là đúng trên tổng số mẫu dự đoán là "positive". Trong mô hình phân loại sẽ kỳ vọng chỉ số này lớn nhất.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \textit{PT 2.27}$$

Accuracy: Tỷ lệ các dự đoán chính xác trên tổng số mẫu dự đoán.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad \textit{PT 2.28}$$

F1 score: Sử dụng "harmonic mean" để tính toán giữa 2 chỉ số recall và precision giúp việc so sánh giữa 2 mô hình có recall thấp, precision cao và ngược lại được thuận lợi hơn.

$$\text{F1 score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad \textit{PT 2.29}$$

FPR: Tỷ lệ dự đoán sai các trường hợp "positive" trên tổng số mẫu dự đoán là "negative".

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad \textit{PT 2.30}$$

Specificity: Tỷ lệ dự đoán đúng các trường hợp "negative" trên tổng số trường hợp "negative" thực tế

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} = 1 - \text{FPR} \quad \textit{PT 2.31}$$

Confusion matrix cung cấp trực quan kết quả dự đoán của mô hình. Một trong những lợi ích chính của việc sử dụng confusion matrix là nó giúp xác định điểm mạnh và điểm yếu của mô hình. Bằng cách xem xét confusion matrix, có thể biết mô hình đang mắc phải loại lỗi nào và điều chỉnh cách tiếp cận phù hợp. Ví dụ: nếu chỉ số FP quá cao, thì cần điều chỉnh ngưỡng dự đoán để giảm số lượng dự đoán FP.

2.5.5.2. AUC-ROC curve

AUC-ROC curve là một phương pháp tính toán hiệu suất của một mô hình phân loại theo các ngưỡng phân loại khác nhau. Với bài toán phân loại nhị phân kết quả đầu ra

của mô hình sẽ là xác suất trong khoảng $(0,1)$ và việc chọn ngưỡng để phân lớp đầu ra này rất quan trọng.

Đường cong ROC (ROC curve) đồ thị của tỷ lệ TPR (*PT 2.26*) so với tỷ lệ FPR (*PT 2.30*) với các ngưỡng phân loại khác nhau. Để tạo ROC curve, xác suất dự đoán của mô hình cho lớp "positive" được sắp xếp theo thứ tự giảm dần. Sau đó, một ngưỡng dự đoán được đặt ra và tất cả các trường hợp có xác suất dự đoán lớn hơn hoặc bằng ngưỡng đó và được phân loại là "positive", trong khi các trường hợp khác được phân loại là "negative". TPR và FPR được tính toán dựa trên các phân loại này và quy trình được lặp lại cho các ngưỡng khác nhau [97] [98].

AUC là vùng bên dưới đường cong ROC, biểu thị xác suất mà một trường hợp "positive" được chọn ngẫu nhiên sẽ được xếp hạng cao hơn một trường hợp "negative" được chọn ngẫu nhiên theo mô hình. AUC nằm trong khoảng [0,1], với $\mathrm{AUC} = 0{,}5$ cho biết mô hình không tốt và hoàn toàn không có khả năng phân loại giữa 2 lớp, $\mathrm{AUC} = 1$ cho biết mô hình phân loại rất tốt [99].

2.5.5.3. Điểm LB

Điểm LB là phương thức đánh giá kết quả của các mô hình huấn luyện trong những cuộc thi trên Kaggle. Phương thức này nhằm đánh giá và sắp xếp thứ hạng cho các mô hình đã huấn luyện bằng các phương thức và thuật toán khác nhau dựa trên bộ dữ liệu mà mỗi cuộc thi cung cấp, các cuộc thi này được tạo ra nhằm đưa ra được các kết quả cho những vấn đề liên quan đến trí tuệ nhân tạo và khoa học máy tính. Ví dụ như cuộc thi do Home Credit tổ chức với bài toán đặt ra là dự đoán khả năng trả nợ của khách hàng đăng ký vay [100].

Các cuộc thi được tổ chức sẽ sử dụng phương thức đánh giá LB này sẽ tạo ra hai bộ dữ liệu được sử dụng để đánh giá hiệu suất của người tham gia: bộ dữ liệu công khai và bộ dữ liệu ẩn. Bộ dữ liệu công khai sẽ được cung cấp công khai trong quá trình cuộc thi diễn ra để giúp đánh giá được hiệu suất của các mô hình. Bộ dữ liệu ẩn sẽ chỉ hiển thị kết quả đánh giá khi cuộc thi đã kết thúc nhằm xác định thứ hạng.

Khi tạo được bộ dữ liệu dự đoán cho mô hình và nộp lên hệ thống cuộc thi sẽ nhận được 2 chỉ số đánh giá là public score và private score tương ứng với bộ dữ liệu công khai và bộ dữ liệu ẩn. Kaggle sử dụng các thước đo đánh giá được xác định trước do ban tổ chức cuộc thi chỉ định để đánh giá và đưa ra chỉ số đánh giá. Chỉ số đánh giá trên bộ dữ liệu ẩn sẽ khắt khe hơn so với bộ dữ liệu công khai. Chỉ số public score và private score càng cao thì hiệu suất mô hình huấn luyện sẽ càng tốt.

2.5.6. Phương pháp tìm ngưỡng phân loại

Trong đề tài nghiên cứu này, đầu ra của các mô hình dữ liệu là giá trị xác suất trong khoảng (0,1) chỉ báo xác suất vỡ nợ của khách hàng. Từ giá trị xác suất này cần tìm được một ngưỡng giá trị mà tại đó các giá trị xác suất được chuyển đổi chính xác nhất thành giá trị nhị phân nhằm phân loại khách hàng có rủi ro tín dụng và khách hàng không có rủi ro tín dụng. Tìm kiếm ngưỡng phân loại tốt nhất cho mô hình dự đoán có thể dựa vào một trong số các số chỉ số đánh giá sau: accuracy, precision, recall, F1-Score, sensitivity(TPR), specificity, roc_auc.

Quá trình tìm ngưỡng phân loại sẽ sử dụng phương pháp thống kê Youden's J [101] với mục tiêu tìm giá trị lớn nhất của $\boldsymbol{\mathcal{J}}$:

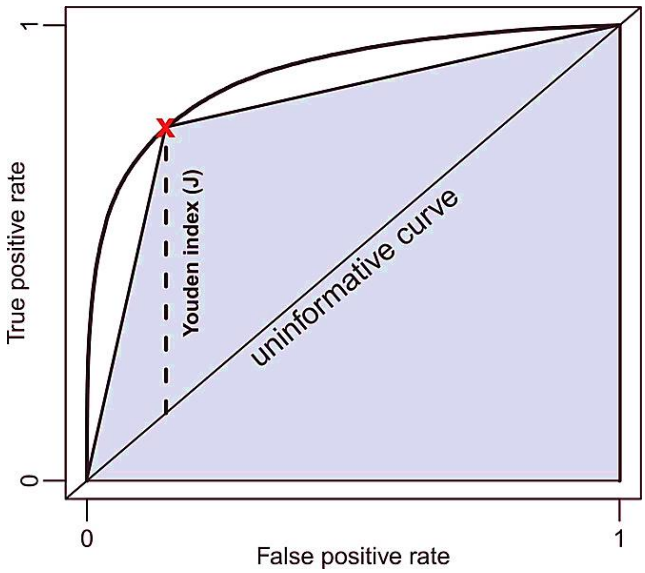
$$\begin{aligned} \boldsymbol{\mathcal{J}} &= \text{sensitivity} + \text{specificity} - 1 \\ &= \text{TPR} - (1 - \text{FPR}) - 1 \\ &= \text{TPR} - \text{FPR} \end{aligned} \qquad \textit{PT 2.29}$$

Các giá trị sensitivity(TPR), specificity, FPR là các giá trị đã được đề cập trong phần *Phương pháp đánh giá mô hình.*

Các bước tìm ngưỡng phân loại bao gồm:

1. Xây dựng đường cong ROC bằng cách sử dụng xác suất dự đoán của mô hình và nhãn thực tế.
2. Lấy ra các bộ giá trị bao gồm (FPR, TPR, threshold) quá trình xây dựng đường cong ROC.

3. Tính toán chỉ số thống kê $\boldsymbol{J}$ cho mỗi bộ giá trị (FPR, TPR, threshold).
4. Với bộ giá trị có chỉ số $\boldsymbol{J}$ lớn nhất. Giá trị threshold là giá trị ngưỡng phân loại tốt nhất của mô hình.



Hình 2.23 Minh họa giá trị ngưỡng phân loại tốt nhất trên đường cong ROC [102]

2.5.7. Phương pháp chuyển đổi điểm tín dụng từ kết quả đầu ra mô hình

Trong chủ đề nghiên cứu này mục tiêu của mô hình chấm điểm tín dụng sẽ là điểm tín dụng với các thuộc tính dữ liệu người dùng ở đầu vào. Tuy nhiên, mô hình dự đoán với đầu ra sẽ là xác xuất trong khoảng (0,1) thể hiện xác suất vỡ của khách hàng. Mục tiêu của phương pháp chuyển đổi điểm tín dụng từ kết quả đầu ra mô hình sẽ chuyển đổi xác suất từ đâu ra của mô hình thành điểm tín dụng trong khoảng (300,850).

Với mỗi mẫu dự đoán sẽ được chuyển đổi với công thức:

$$\mathbf{CS} = \alpha \times 300 + (1 - \alpha) \times 850 \qquad \textit{PT 2.32}$$

Trong đó:

- **CS**: Điểm tín dụng
- α: Giá trị dự đoán của mô hình

Ví dụ, nếu giá trị dự đoán của mô hình là 0.5, khi đó điểm tín dụng của khách hàng ứng với mẫu dữ liệu mới sẽ là: $\mathbf{CS} = (0.5 \times 300 + (1 - 0.5) \times 850) = 575$. Như vậy điểm tín dụng của khách hàng này sẽ là 575 điểm.

CHƯƠNG 3. DỮ LIỆU

3.1. Tổng quan dữ liệu

Bộ dữ liệu "Home Credit Default Risk" được sử dụng rộng rãi trong ngành tài chính, đặc biệt là để đánh giá rủi ro tín dụng và dự đoán vỡ nợ. Bộ dữ liệu được tổng hợp và công khai bởi Home Credit Group, một tổ chức tài chính phi ngân hàng quốc tế chuyên cung cấp các khoản vay cho các cá nhân ít hoặc không có lịch sử tín dụng và chưa được tiếp cận đầy đủ các dịch vụ từ ngân hàng. Bộ dữ liệu được cung cấp công khai trên Kaggle [8], bao gồm nhiều tệp dữ liệu định dạng CSV, mỗi tệp cung cấp thông tin về các khía cạnh khác nhau liên quan đến dữ liệu tín dụng của mỗi khách hàng đăng ký khoản vay.

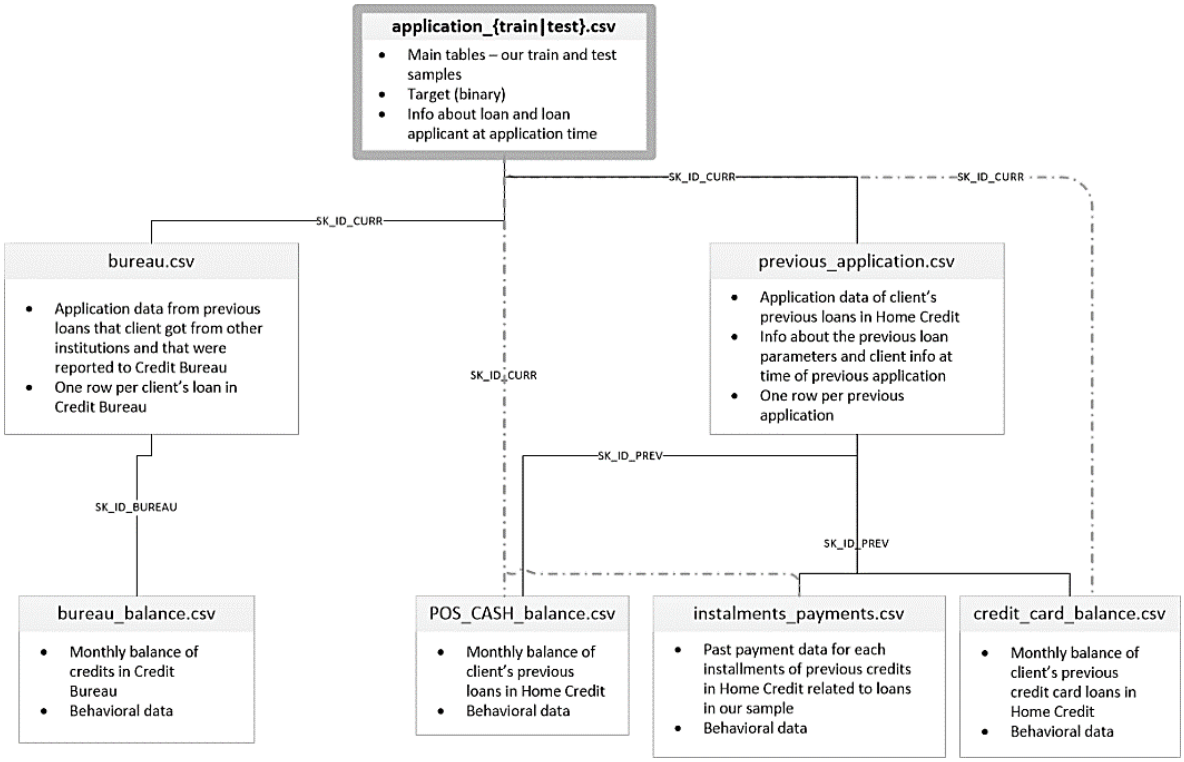
Hình 3.1 Logo Home Credit Group

Tệp dữ liệu chính trong bộ dữ liệu là tệp dữ liệu "application_train. csv", tệp dữ liệu này chứa thông tin về khách hàng đăng ký khoản vay và biến mục tiêu nhị phân (0 hoặc 1) cho biết liệu rằng người đăng ký khoản vay có gặp khó khăn trong việc thanh toán các khoản vay trước đó đúng hạn hay không với giá trị 0 thể hiện khách hàng đăng ký vay không gặp khó khăn trong việc hoàn trả nợ và 1 thể hiện khách hàng đăng ký vay gặp khó khăn trong việc thanh toán nợ và đây là khách hàng có rủi ro tín dụng cao. Các tệp khác trong tập dữ liệu cung cấp thông tin bổ sung về dữ liệu tín dụng có liên quan đến khách hàng đăng ký khoản vay, các khoản vay đã đăng ký trước đó trước đó, số dư thẻ tín dụng và các thông tin liên quan khác.

Trong bộ dữ liệu này có đề cập thêm thông tin dữ liệu tín dụng từ các tổ chức Credit Bureau. Đây là những văn phòng tín dụng thu thập thông tin tín dụng từ nhiều tổ chức tài chính và phi tài chính, bao gồm các tổ chức tài chính vi mô và công ty tín dụng, đồng thời cung cấp thông tin tín dụng tiêu dùng toàn diện. Những văn phòng tín dụng này có xu hướng thu thập dữ liệu rất chi tiết về các khách hàng cá nhân, do đó các thông tin thường toàn diện hơn và được thiết kế tốt hơn để đánh giá và giám sát mức độ tín nhiệm của các khách hàng cá nhân [103]. Trong bộ dữ liệu Home Credit Default Risk cũng có sử dụng các thông tin về những khoản tín dụng trước đây của các khách hàng đăng ký khoản vay ở Home Credit.

Bộ dữ liệu chứa nhiều biến số về thông tin khách hàng đăng ký khoản vay như dữ liệu nhân khẩu học, lịch sử tín dụng, thu nhập, lịch sử việc làm và nhiều yếu tố khác có thể được sử dụng để dự đoán mức độ tin cậy của khách hàng đăng ký khoản vay.

3.2. Mô tả dữ liệu



Hình 3.2 Tổng quan mối quan hệ giữa các tệp dữ liệu trong bộ dữ liệu Home Credit Default Risk

Bảng 3.1 Thông tin về các tập dữ liệu trong bộ dữ liệu Home Credit Default Risk

STT	Tên tệp dữ liệu	Số lượng hàng	Số lượng cột	Định dạng	Số lượng thuộc tính categorical	Số lượng thuộc tính numeric
1	application_train	307,511	122	CSV	16	106
2	application_test	48,744	121	CSV	16	105
3	bureau	1,716,428	17	CSV	3	14
4	bureau_balance	27,299,925	3	CSV	1	2
5	previous_application	1,670,214	37	CSV	16	21
6	POS_CASH_balance	10,001,203	8	CSV	4	4
7	credit_card_balance	38,067,841	23	CSV	4	19
8	installments_payments	13,605,401	8	CSV	1	7

Dữ liệu application{train|test}.csv:

- Đây là tập dữ liệu chính đã được chia thành 2 tập dữ liệu dành cho quá trình huấn luyện (có biến mục tiêu "TARGET") và tập dữ liệu kiểm thử (không có biến mục tiêu "TARGET")
- Đây là dữ liệu cố định cho các khách hàng đăng ký khoản vay. Mỗi hàng dữ liệu đại diện một khoản vay.

Dữ liệu bureau.csv:

- Dữ liệu về các khoản tín dụng trước đây của các khách hàng đăng ký vay được cung cấp bởi các phòng tín dụng bao gồm thông tin về loại khoản vay, số tiền cho vay, số ngày quá hạn thanh toán, …

- Đối với mỗi khách hàng đăng ký vay sẽ có nhiều hàng thể hiện số lượng tín dụng mà khách hàng đã có trong báo cáo từ phòng tín dụng trước ngày nộp đơn đăng ký vay ở Home Credit.

Dữ liệu bureau_balance. csv

- Dữ liệu về số dư hàng tháng của khách hàng đăng ký vay tín dụng trước đây và có thông tin trong dữ liệu bureau.csv.
- Với mỗi khoản tín dụng, mỗi hàng trong dữ liệu thể hiện một tháng trong lịch sử của các khoản tín dụng.

Dữ liệu POS_CASH_balance. csv:

- Dữ liệu chứa thông tin về số dư hàng tháng của các điểm bán hàng (POS) và các khoản vay tiền mặt mà khách hàng đã đăng ký với Home Credit trước đây.
- Mỗi hàng dữ liệu thể hiện thông tin giao dịch tín dụng mỗi tháng của tất cả các khoản tín dụng trước đây (tín dụng tiêu dùng và cho vay tiền mặt).

Dữ liệu credit_card_balance. csv:

- Dữ liệu chứa dữ liệu số dư hàng tháng của các giao dịch thẻ tín dụng mà khách hàng đã đăng ký với Home Credit trước đây.
- Mỗi hàng dữ liệu thể hiện thông tin giao dịch tín dụng mỗi tháng của tất cả các khoản tín dụng trước đây của khách hàng đã từng đăng ký (tín dụng tiêu dùng và cho vay tiền mặt).

Dữ liệu previous_application. csv:

- Dữ liệu chứa thông tin về các khoản tín dụng mà khách hàng đã đăng ký trước đây bao gồm mã khách hàng đăng ký, số tiền vay, ngày nộp đơn, trạng thái phê duyệt, …
- Mỗi hàng trong dữ liệu thể hiện một khoản tín dụng mà khách hàng đã đăng ký trước đó.

Dữ liệu installments_payments.csv:

- Dữ liệu chứa thông tin về lịch sử thanh toán nợ cho các khoản tín dụng đã giải ngân trước đây mà khách hàng đã từng đăng ký với Home Credit.
- Mỗi hàng tương ứng với một lần thanh toán của các khoản tín dụng thanh toán một lần hoặc một lần trả góp cho các khoản tín dụng trả góp.

3.3. Chuẩn bị dữ liệu

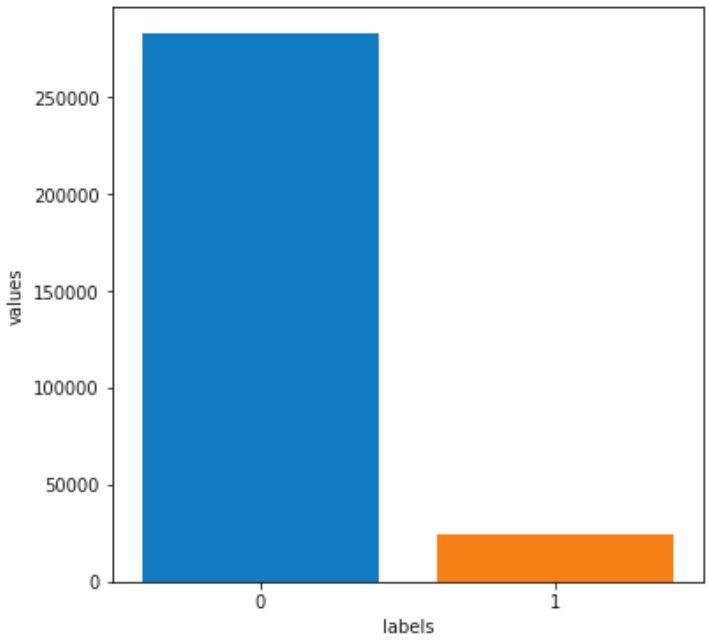
3.3.1. Phân tích khám phá dữ liệu

Quá trình phân tích dữ liệu trên bộ dữ liệu Home Credit Default Risk liên quan đến việc tạo các biểu đồ trực quan để khám phá dữ liệu và xác định các mẫu ngoại lệ. Những thông tin trực quan này sẽ giúp xác định xu hướng, mối quan hệ và sự bất thường trong dữ liệu từ đó cung cấp thông tin về bộ dữ liệu, phục vụ cho quá trình xử lý và trích xuất đặc trưng [104].

3.3.1.1. Cân bằng dữ liệu trên biến mục tiêu

Biến mục tiêu trong bộ dữ liệu đào tạo có 2 nhãn (label) với:

- 0: khoản vay đã được hoàn trả.
- 1: khoản vay không được hoàn trả.

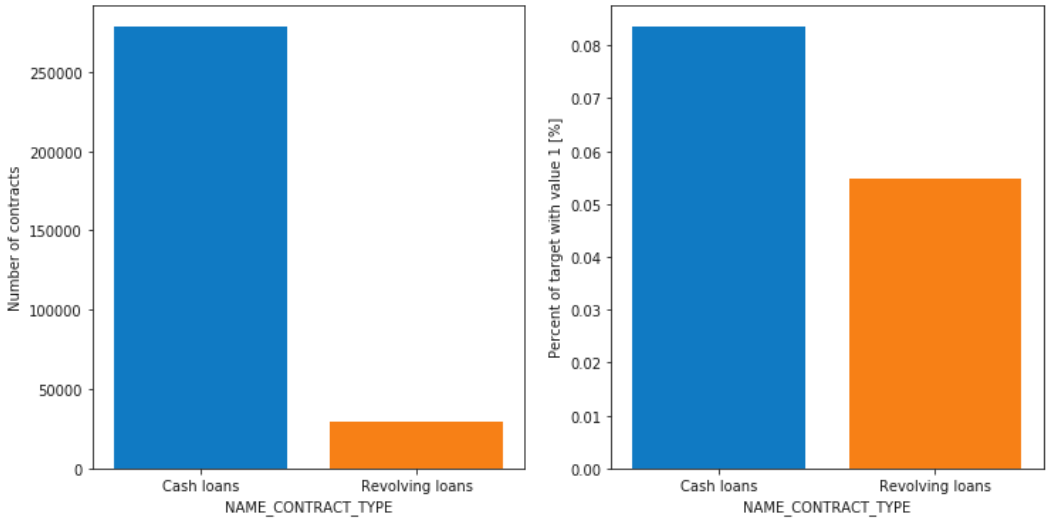


Hình 3.3 Đồ thị biểu diễn số lượng nhãn phân loại của 2 lớp trong biến mục tiêu

Biểu đồ cho thấy rằng dữ liệu có sự mất cân bằng rất lớn giữa 2 lớp phân loại. Vấn đề này có thể dẫn đến việc mô hình có độ chính xác cao đối với lớp 0 nhưng hoạt động kém đối với lớp 1 [105].

3.3.1.2. Loại khoản vay

Tìm hiểu các loại khoản vay được thực hiện và tỷ lệ của các loại khoản vay trên các trường hợp không thể trả nợ (label=1).

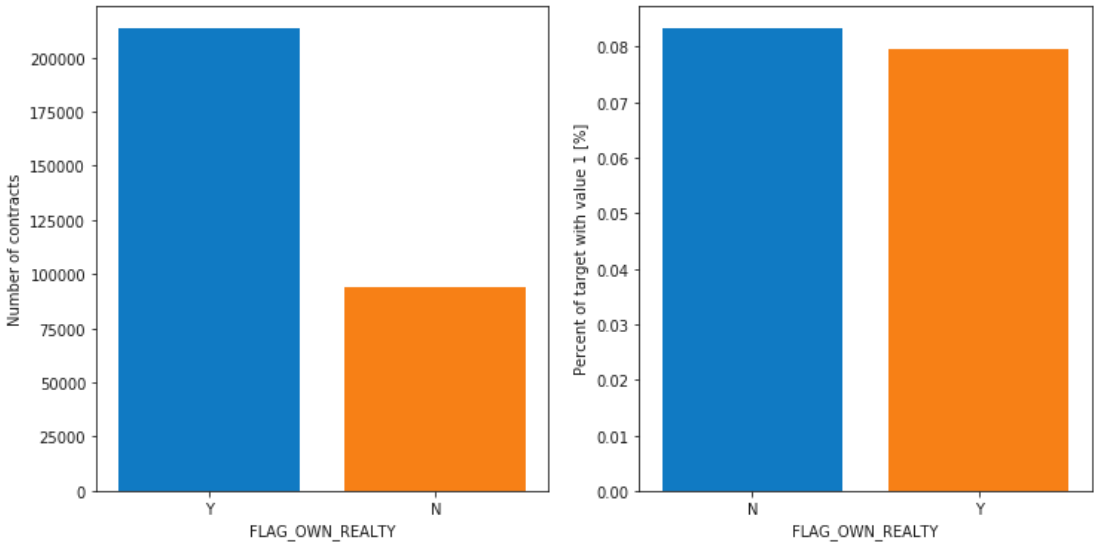


Hình 3.4 Biểu đồ số lượng loại khoản vay được thực hiện (phải) và tỷ lệ của các loại khoản vay trên các trường hợp không thể trả nợ (trái)

Loại khoản vay "Revolving loans" chỉ chiếm một phần nhỏ trong tổng số các khoản vay (khoảng 10%), chiếm phần lớn là "Cash loans" (Cho vay tiền mặt). Tuy nhiên các khoản vay "Revolving loans" vẫn chiếm một phần lớn trong các khoản vay không thể thanh toán.

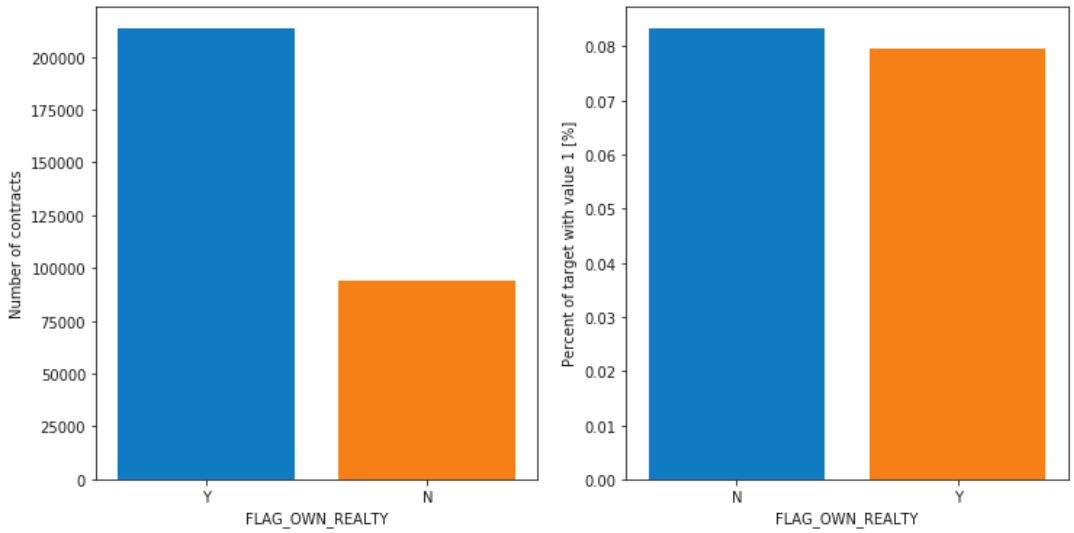
3.3.1.3. Dữ liệu khách hàng có sở hữu ô tô hoặc bất động sản

Tìm hiểu số lượng lượng khách hàng có sở hữu ô tô hoặc bất động sản trong bộ dữ liệu và tỷ lệ trong các khoản vay không thể trả (label=1).



Hình 3.5 Biểu đồ số lượng khách hàng sở hữu ô tô (phải) và tỷ lệ trên các khoản vay không thể trả (trái)

Những khách hàng sở hữu ô tô gần gấp đôi số khách hàng không sở hữu ô tô. Tuy nhiên, cả hai loại (sở hữu ô tô hoặc không sở hữu) đều có tỷ lệ không trả nợ khoảng 8%.

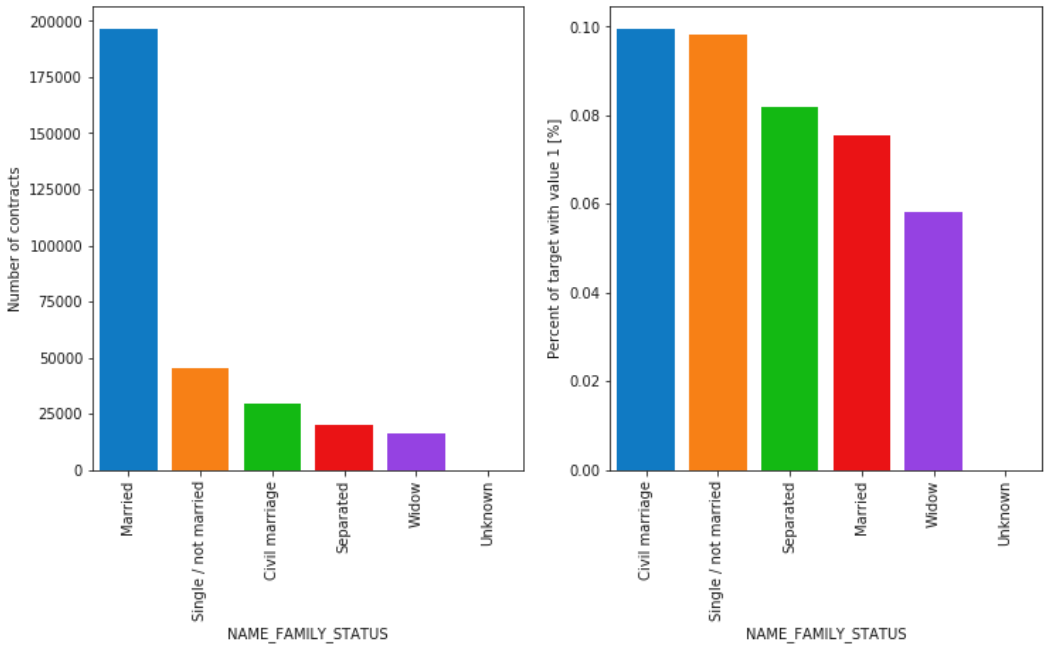


Hình 3.6 Biểu đồ số lượng khách hàng sở hữu bất động sản (phải) và tỷ lệ trên các khoản vay không thể trả (trái)

Những khách hàng sở hữu bất động sản nhiều hơn gấp đôi so với những khách hàng không sở hữu. Cả hai loại (sở hữu bất động sản hoặc không sở hữu) đều có tỷ lệ không trả nợ xấp xỉ 8%.

3.3.1.4. Tình trạng gia đình

Tìm hiểu về tình trạng gia đình của các khách hàng vay và tỷ lệ trên các khoản vay không thể thanh toán (label=1).



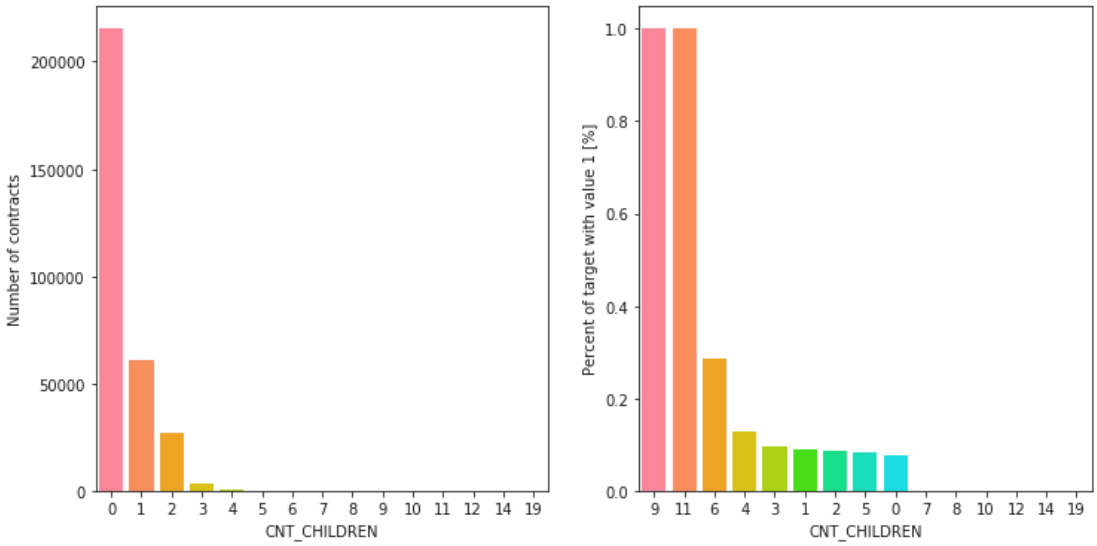
Hình 3.7 Biểu đồ tình trạng gia đình của các khách hàng vay (phải) và tỷ lệ trên các khoản vay không thể thanh toán (trái)

Hầu hết các khách hàng đã kết hôn (Married), tiếp theo là độc thân/chưa kết hôn (Single/not married) và hôn nhân dân sự (Civil marriage).

Xét trên tỷ lệ không hoàn trả nợ, hôn nhân dân sự (Civil marriage) có tỷ lệ không trả được nợ cao nhất (10%), khách hàng là góa phụ (Widow) có tỷ lệ thấp nhất.

3.3.1.5. Thành viên gia đình

Tìm hiểu về phân phối số lượng thành viên trong gia đình của các khách hàng vay và tỷ lệ trên các khoản vay không thể thanh toán (label=1).

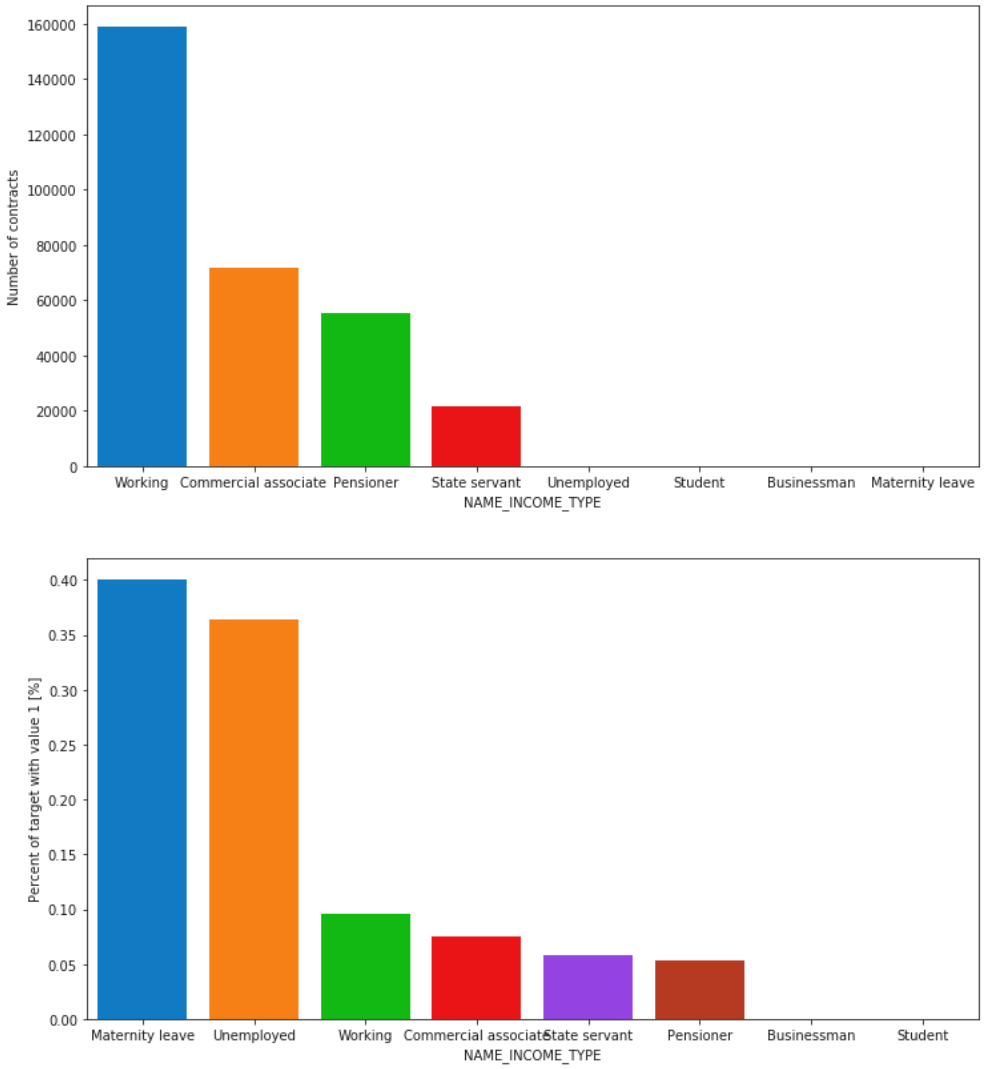


Hình 3.8 Biểu đồ phân phối số lượng thành viên trong gia đình (phải) của các khách hàng vay và tỷ lệ trên các khoản vay không thể thanh toán (trái)

Khách hàng có thành viên gia đình gồm 2 người có số lượng khoản vay nhiều nhất, tiếp theo là gia đình có 1 người (độc thân), 3 người và 4 người.

Các gia đình có 10 hoặc 8 thành viên có tỷ lệ không hoàn trả nợ trên 30%. Các gia đình có từ 6 thành viên trở xuống có tỷ lệ không hoàn trả nợ ở mức trung bình 10%. Khách hàng có số lượng người trong gia đình từ 11 đến 13 người có tỷ lệ không hoàn trả nợ là 100% (Đây có thể là dữ liệu sai lệch hoặc ngoại lai cần xử lý).

3.3.1.6. Loại thu nhập

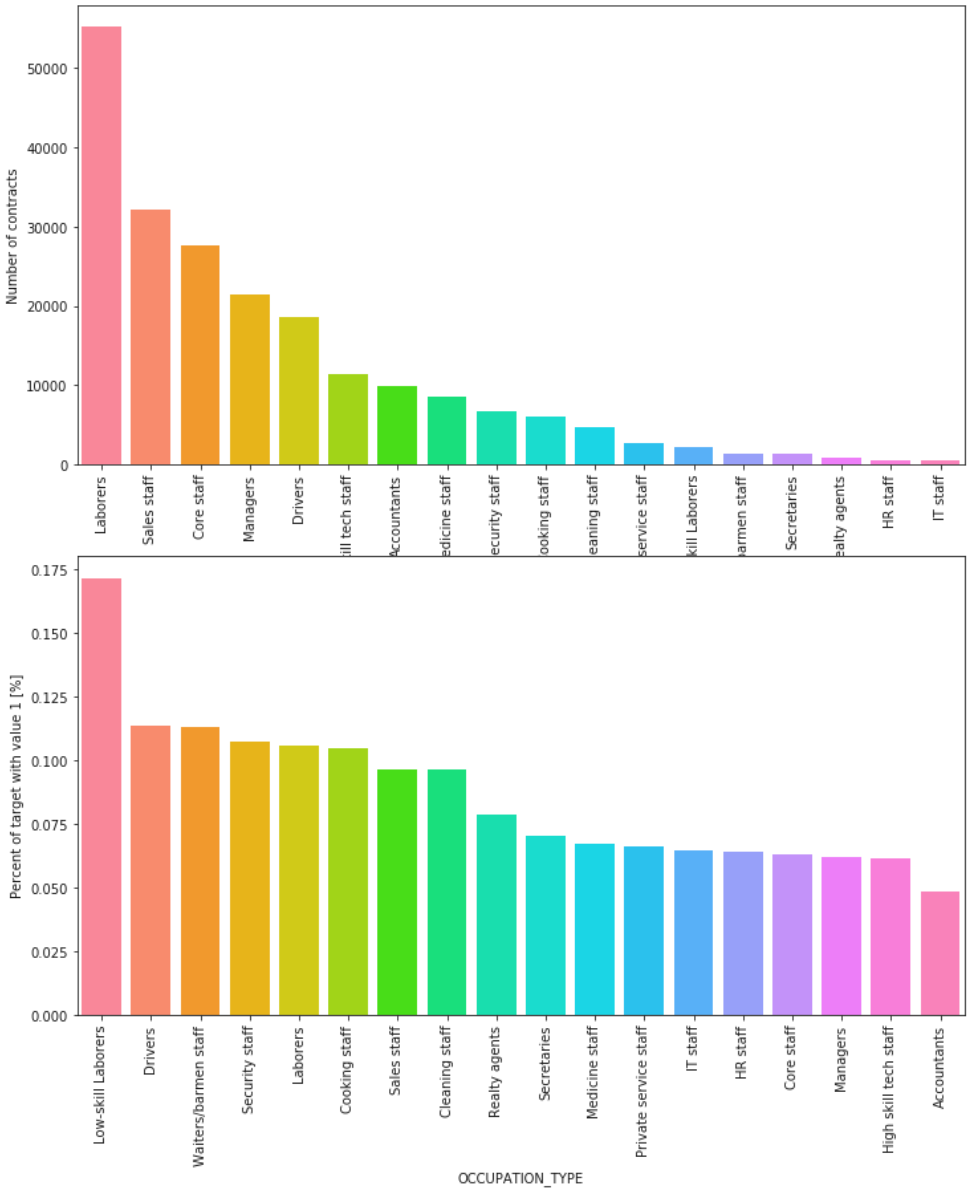


Hình 3.9 Biểu đồ thể hiện số lượng khách hàng vay với các loại thu nhập khác nhau và tỷ lệ trên các khoản vay không hoàn trả

Hầu hết những khách hàng đăng ký vay đều có thu nhập từ làm việc (Working), tiếp theo là cộng tác viên thương mại (Commercial associate), hưu trí (Pensioner) và công chức nhà nước (State servant).

Các khách hàng có thu nhập từ trợ cấp nghỉ thai sản (Maternity leave) có tỷ lệ không trả nợ cao nhất (gần 40%), tiếp theo là thất nghiệp (Unemployed) với tỷ lệ 37%. Các loại thu nhập còn lại dưới mức trung bình 10% đối với các khoản vay không hoàn trả nợ.

3.3.1.7. Ngành nghề làm việc

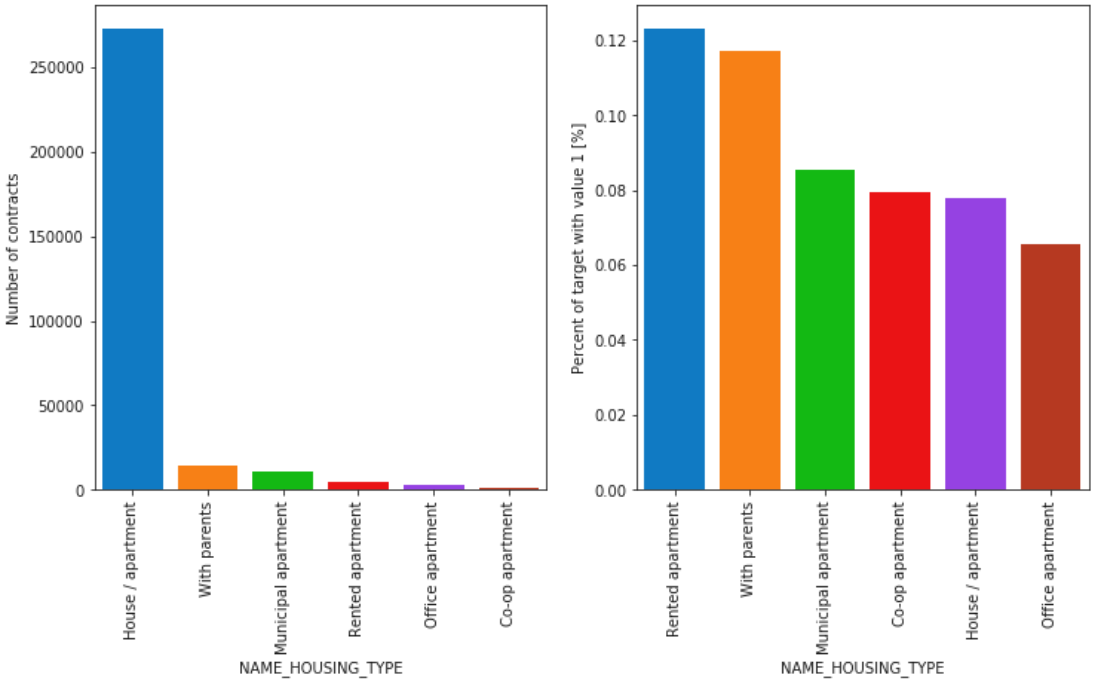


Hình 3.10 Biểu đồ phân phối ngành nghề làm việc của khách hàng vay và tỷ lệ trên số lượng không hoàn trả nợ

Hầu hết các khoản vay được vay bởi người lao động (Laborers), tiếp theo là nhân viên kinh doanh (Sales staff). Nhân viên công nghệ thông tin có số lượng đăng ký vay thấp nhất (IT staff).

Nhóm có tỷ lệ không hoàn trả nợ cao nhất là lao động kỹ năng thấp (Low-skill Laborers) với trên 17%, tiếp theo là nhân viên lái xe (Drivers) và nhân viên phục vụ (Waiters/barmen staff), nhân viên bảo vệ (Security staff), người lao động (Laborers) và Nhân viên nấu ăn (Cooking staff).

3.3.1.8. Loại nhà ở và cư trú



Hình 3.11 Biểu đồ phân phối số lượng của các loại nhà ở/cư trú của các khách hàng vay và tỷ lệ trên số lượng không hoàn trả nợ

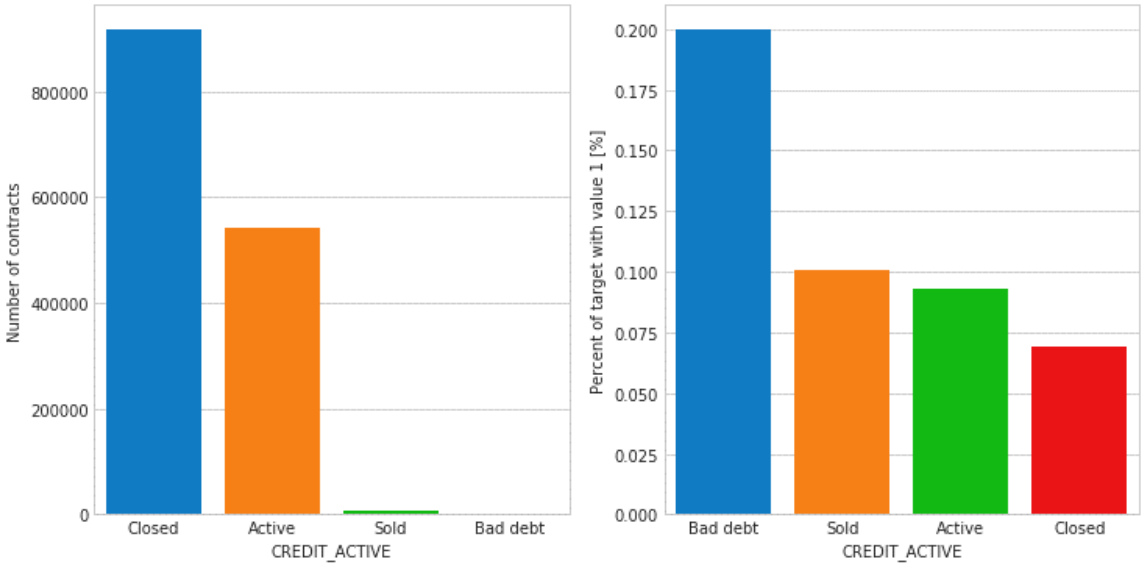
Hơn 250.000 khách hàng đăng ký vay tín dụng với loại nhà ở là nhà/căn hộ (House/apartment). Các danh mục sau có số lượng khách hàng rất nhỏ (Ở với cha mẹ (With parents), căn hộ thành phố (Municipal apartment).

Các khách hàng ở căn hộ cho thuê (Rented apartment) và ở cùng cha mẹ (With parents) có tỷ lệ không trả nợ cao hơn 10%.

3.3.1.9. Tình trạng tín dụng

Tìm hiểu sự phân bố tình trạng tín dụng với:

- Closed: Tín dụng đã đóng.
- Active: Tín dụng đang hoạt động.
- Sold: Tín dụng đã được bán.
- Bad debt: Tín dụng xấu.



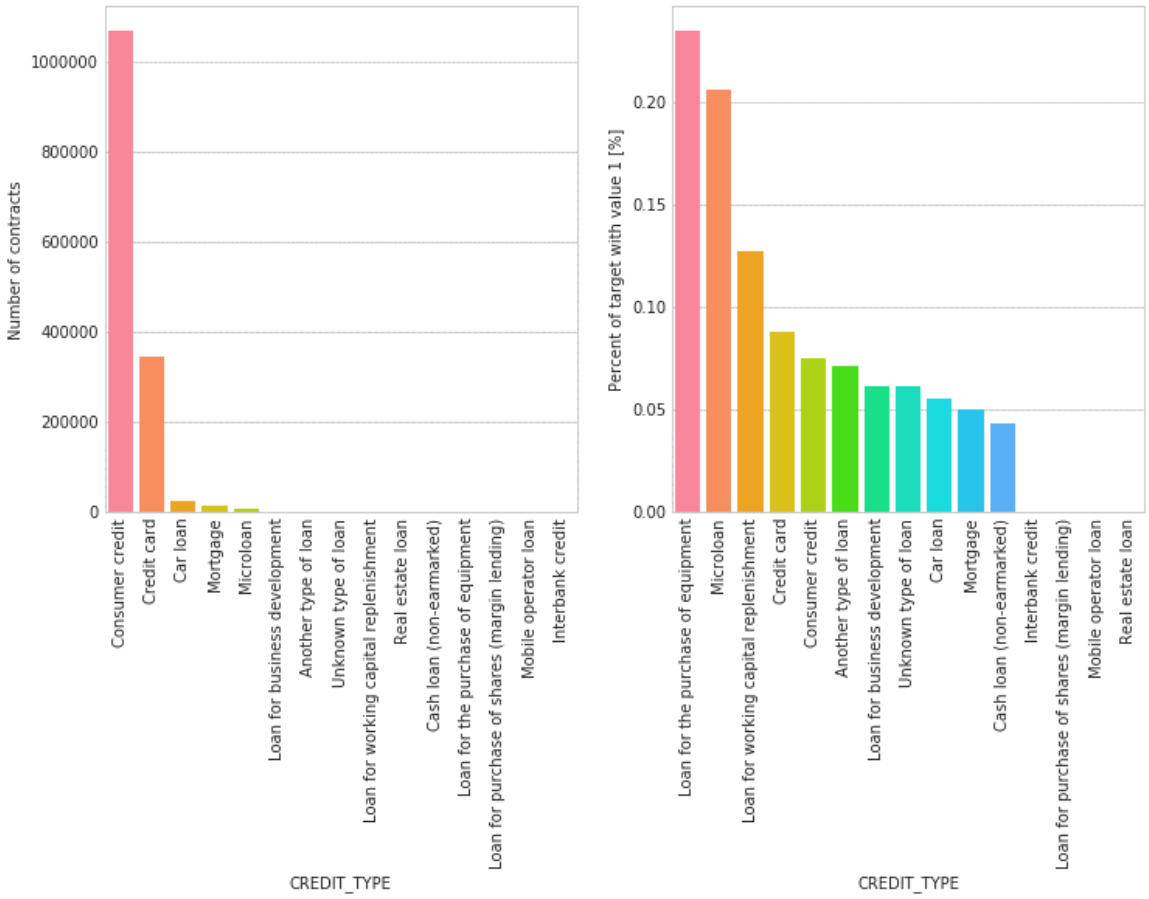
Hình 3.12 Phân phối loại tín dụng và tỷ lệ trên các khoản vay không hoàn trả nợ

Hầu hết các khoản tín dụng đã đăng ký tại phòng tín dụng đều ở trạng thái "Closed" (Khoản 900 nghìn khoản tín dụng). Tiếp theo là các khoản tín dụng "Active" (khoảng 600 ngàn khoản tín dụng). Các khoản tín dụng có trạng thái "Sold" và "Bad debt" chiếm số lượng ít.

Các khách hàng có tín dụng đã đăng ký với phòng tín dụng có trạng thái "Bad debt" có tỷ lệ không hoàn trả nợ cao (Khoảng 20%). Khách hàng có trạng thái tín dụng "Sold", "Active" và "Closed" có tỷ lệ không hoàn trả nợ nhỏ hơn 10%.

Như vậy, lịch sử tín dụng đã đăng ký trước đây là một yếu tố dự đoán mạnh mẽ cho các khoản tín dụng sai phạm.

3.3.1.10. Loại tín dụng



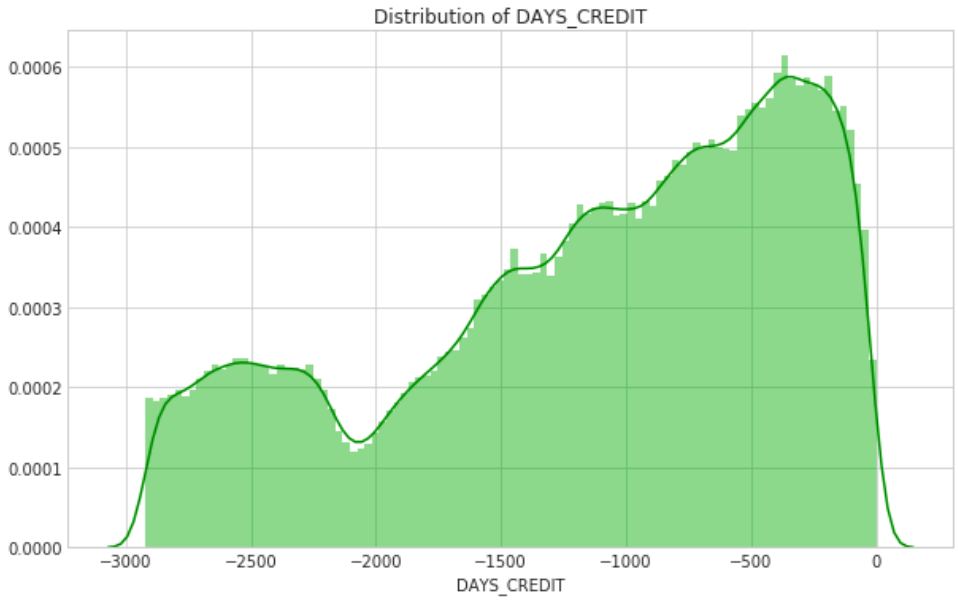
Hình 3.13 Biểu đồ phân phối của các loại tín dụng khác nhau và tỷ lệ trên các khoản vay không hoàn trả nợ

Phần lớn các khoản tín dụng được đăng ký tại phòng tín dụng là "Consumer credit" (tín dụng tiêu dùng) và "Credit card" (thẻ tín dụng). Số lượng tín dụng chiếm số lượng nhỏ hơn là "Car loan" (Tín dụng mua/thuê ô tô), "Mortgage" (Tín dụng thế chấp tài sản) và "Microloan" (Tín dụng với các khoản vay rất nhỏ).

Với các loại tín dụng đã đăng ký tại phòng tín dụng trước đây, có một số loại có tỷ lệ vỡ nợ tín dụng cao như:

- "Loan for the purchase of equipment" (Khoản vay mua thiết bị) và "Microloan", mỗi loại chiếm hơn 20% số lượng các khoản vay không hoàn trả nợ.
- "Loan for working capital replenishment" (Khoản vay bổ sung vốn lưu động) với hơn 12% trên các khoản nợ không hoàn trả nợ.

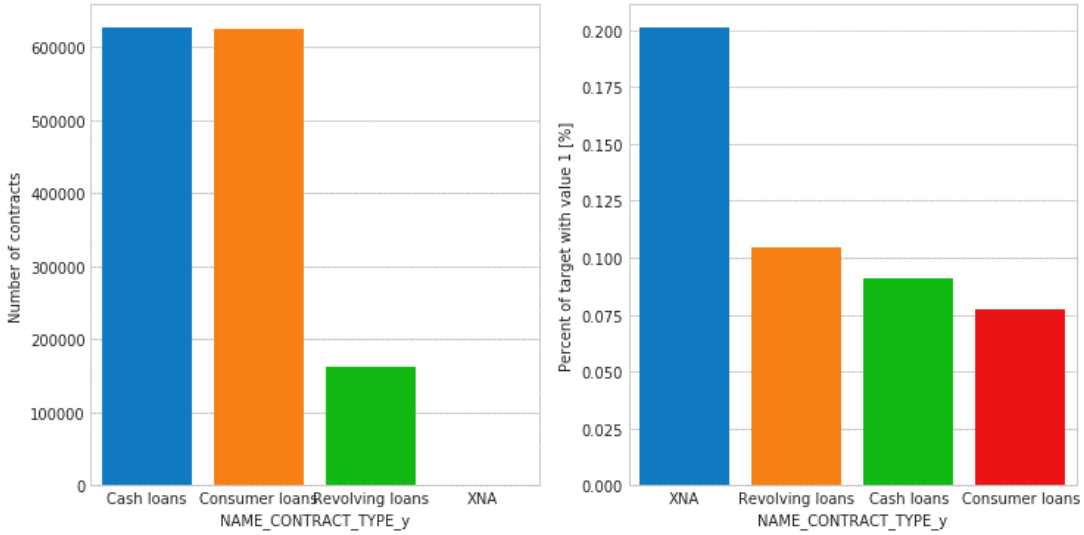
3.3.1.11. Thời hạn tín dụng



Hình 3.14 Phân phối thời hạn tín dụng của các khoản tín dụng đã vay trước đây từ dữ liệu của phòng tín dụng

Thời hạn tín dụng được tính theo ngày dao động từ 500 ngày đến 3000 ngày. Từ biểu đồ có thể thấy phân phối các các khoản vay nhiều nhất ở khoảng từ 200 đến 1000 ngày và mức cao nhất ở khoảng 300 ngày. Các khoản vay thường được đăng ký với thời hạn tín dụng từ 200 đến 100 ngày.

3.3.1.12. Loại hợp đồng tín dụng



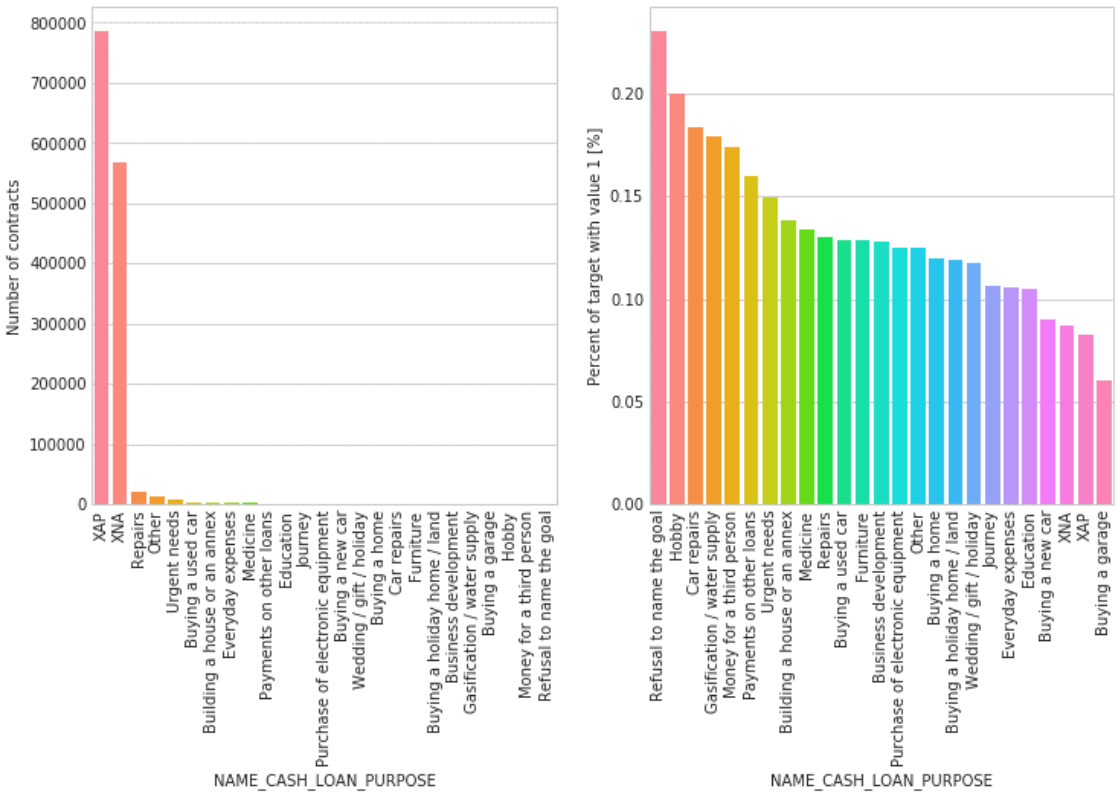
Hình 3.15 Phân bố số lượng của các loại hợp đồng tín dụng khác nhau và tỷ lệ trên các khoản vay không hoàn trả nợ

Có 3 loại hợp đồng trong dữ liệu đăng ký tín dụng trước đây bao gồm: "Cash loans" (Vay tiền mặt), "Consumer loans" (vay tiêu dùng), "Revolving loans" (Vay quay vòng). Số lượng khoản vay tiền mặt và vay tiêu dùng gần bằng nhau (Khoảng 600 nghìn khoản vay) trong khi số lượng khoản vay quay vong khá thấp (150 nghìn khoản vay).

Tỷ lệ các khoản vay không hoàn trả nợ đối với các khách hàng đăng ký khoản vay trước đó khác nhau đối với loại hợp đồng, giảm từ 10% đối với khoản vay quay vòng, 9,5% đối với khoản vay tiền mặt và 8% đối với khoản vay tiêu dùng.

3.3.1.13. Mục đích vay tiền mặt

Tìm hiểu về mục đích vay tiền mặt đối với các khoản vay tiền mặt.



Hình 3.16 Phân bố số lượng của các mục đích vay tiền mặt khác nhau và tỷ lệ trên số lượng khoản vay không hoàn trả nợ

Trừ các khoản vay không xác định (XAP và XNA), các khoản vay "Repairs" (khoản vay dùng để sửa chữa), "Urgent needs" (khoản vay dùng cho nhu cầu cấp thiết), "Buying a used car" (Khoản vay dùng để mua xe cũ) và "Building a house

or an annex" (khoản vay dùng để xây nhà hoặc công trình phụ) chiếm số lượng lớn các khoản vay tín dụng.

Tỷ lệ các khoản vay không hoàn trả nợ cao nhất ở khoản vay "Refusal to name the goal" (Khoản vay không có lý do) chiếm 23%, "Hobby" (khoản vay phục vụ sở thích) chiếm (20%) và "Car repairs" (khoản vay để sử xe) chiếm 18%.

Từ các biểu đồ và phân tích [106] thông qua các thuộc tính trong bộ dữ liệu Home Credit Default Risk có thể nắm bắt được các thuộc tính quan trọng ảnh hưởng đến tỷ lệ không hoàn trả nợ vay. Từ quá trình này, tiếp tục thực hiện trích xuất các tính năng từ những thông tin đã nắm bắt.

3.3.2. Trích xuất đặc trưng dữ liệu

Sau khi hoàn thành các bước phân tích và khám phá dữ liệu ban đầu, bước quan trọng tiếp theo là trích xuất tính năng. Trong bước này, từ thuộc tính có liên quan nhất đến các các yếu tố thanh toán khoản vay trong bộ dữ liệu tiến hành tạo thêm các thuộc tính mới bằng cách chọn các tính năng mang tính dự báo, cung cấp nhiều thông tin và sử dụng các kiến thức có liên quan đến tài chính tín dụng [107], từ đó có thể khai thác đầy đủ các giá trị ẩn trong dữ liệu phục vụ quá trình xây dựng các mô hình [108].

3.3.2.1. Tập dữ liệu POS_CASH_balance

Sử dụng các hàm tổng hợp để lấy ra:

- **POS_CNT_INSTALMENT_FUTURE_MIN**: Số tiền tối thiểu còn lại để thanh toán cho khoản tín dụng.
- **POS_NAME_CONTRACT_STATUS_Completed_SUM**: Số lượng trạng thái hợp đồng là "complete".
- Nếu đồng thời **POS_CNT_INSTALMENT_FUTURE_MIN** và **POS_NAME_CONTRACT_STATUS_Completed_SUM** đều bằng 0 thì sẽ đánh nhãn cho các khoản tín dụng này không trả đúng hạn **POS_NEW_IS_CREDIT_NOT_COMPLETED_ON_TIME** sẽ là:
    - 1: Khoản vay chưa được đóng đúng hạn.

- 0: Khoản vay đã được đóng đúng hạn.

Các thuộc tính còn lại của tập dữ liệu **POS_CASH_balance** sẽ sử dụng các hàm aggregate (min, max, mean, count, sum) để tạo ra các thuộc tính mới.

3.3.2.2. Tập dữ liệu credit_card_balance

Tạo thuộc tính **SK_DPD**: Số ngày mà khoản vay đó bị trễ hạn (day past due). Tính số lần xảy ra các khoản thanh toán chậm dựa trên thuộc tính SK_DPD (Giá trị khác 0).

Tạo thuộc tính **AMT_INST_MIN_REGULARITY**: Khoản thanh toán tối thiểu cho khoản tín dụng vào tháng đó.

Tạo thuộc tính **AMT_PAYMENT_CURRENT**: Số tiền mà khách hàng đã thanh toán cho khoản tín dụng của tháng đó.

Với mỗi khách hàng tính tổng số lượng giao dịch có khoảng thanh toán nhỏ hơn khoản thanh toán tối thiểu mà khoản tín dụng đó đặt ra (**AMT_PAYMENT_CURRENT < AMT_INST_MIN_REGULARIT**).

Tạo thuộc tính **PERCENTAGE_MIN_MISSED_PAYMENTS** (tỷ lệ phần trăm các khoản thanh toán nhỏ hơn khoản thanh toán tối thiểu):

$$\frac{\sum(\textbf{AMT\_PAYMENT\_CURRENT} < \textbf{AMT\_INST\_MIN\_REGULARIT})}{\text{number of instalments (số lần trả góp)}}$$

Tạo thuộc tính **AMT_DRAWINGS_ATM_CURRENT**: Số tiền rút trong tháng tại ATM. Tính tổng cho mỗi SK_ID_CURR.

Tạo thuộc tính **AMT_DRAWINGS_CURRENT**: Số tiền rút ra trong tháng. Tính tổng cho mỗi SK_ID_CURR.

Tạo thuộc tính $\textbf{CASH\_CARD\_RATIO} = \frac{\text{AMT\_DRAWINGS\_ATM\_CURRENT}}{\text{AMT\_DRAWINGS\_CURRENT}}$

Các thuộc tính còn lại của tập dữ liệu **credit_card_balance** sẽ sử dụng các hàm aggregate (min, max, mean, count, sum) để tạo ra các thuộc tính mới.

3.3.2.3. Tập dữ liệu previous_application

Tạo thuộc tính $\mathbf{NEW\_LOAN\_RATE} = \frac{\text{Tỷ lệ khoản vay yêu cầu (AMT\_APPLICATION)}}{\text{Khoản tín dụng được cấp (AMT\_CREDIT)}}$

Từ các thuộc tính:

- **AMT_ANNUITY**: Khoản thanh toán cố định hằng năm (niên kim).
- **CNT_PAYMENT**: Kỳ hạn của khoản tín dụng.
- **AMT_CREDIT**: Khoản tiền tín dụng được cấp.

Tạo thuộc tính **INTEREST_RATE**:

$$= ((\mathbf{AMT\_ANNUITY} \times \mathbf{CNT\_PAYMENT})/(\mathbf{AMT\_CREDIT}))^{\frac{12}{\mathbf{CNT\_PAYMENT}}} - 1$$

Các thuộc tính còn lại của tập dữ liệu **credit_card_balance** sẽ sử dụng các hàm aggregate (min, max, mean, count, sum) để tạo ra các thuộc tính mới.

3.3.2.4. Tập dữ liệu installments_payments

Từ **DAYS_INSTALMENT** (Thời gian của đợt trả góp của khoản tín dụng đó tính đến ngày application) và **DAYS_ENTRY_PAYMENT** (Thời gian từ lúc chi trả đợt trả góp đó đến ngày application). Nếu khoảng thời gian:

$\mathbf{DAYS\_INSTALMENT} - \mathbf{DAYS\_ENTRY\_PAYMENT} > \mathbf{0}$ gán nhãn (label) là 0 (trả sớm). Ngược lại gán nhãn 1 (trả muộn)

Ví dụ: $\mathbf{DAYS\_INSTALMENT} = -1180$

$\mathbf{DAYS\_ENTRY\_PAYMENT} = -1187$

⇨ Đợt trả góp này là trả sớm (label: 0)

Thêm các thuộc tính về tỷ lệ thanh toán khoản vay và dư nợ còn lại của mỗi khoản vay trước đó:

- Tỷ lệ thanh toán khoản vay $= \frac{\sum(\mathbf{AMT\_PAYMENT}\ \text{(Số tiền trả cho từng đợt trả góp)})}{\sum(\mathbf{AMT\_INSTALMENT}\ \text{(Số tiền phải trả cho từng đợt trả góp)})}$
- Dư nợ $=$ Tổng($\mathbf{AMT\_INSTALMENT}$) $-$ Tổng($\mathbf{AMT\_PAYMENT}$)

Các thuộc tính còn lại của tập dữ liệu **installments_payments** sẽ sử dụng các hàm aggregate (min, max, mean, count, sum) để tạo ra các thuộc tính mới.

Sau quá trình trích xuất đặc trưng từ các thuộc tính cơ bản sẽ có bộ dữ liệu mới có thêm các thuộc tính đã trích xuất, các thuộc tính đã được chuyển đổi từ thuộc tính cũ. Với mỗi thuộc tính mới được trích xuất sẽ sử dụng chỉ số đánh giá giá trị dữ liệu (Information value - IV) để đánh giá lượng thông tin mà một thuộc tính cung cấp về biến mục tiêu trong tập dữ liệu nếu cho ra kết quả đủ tốt sẽ đưa vào tạo thành dữ liệu mới và chuyển qua quá trình tổng hợp dữ liệu.

3.3.3. Tổng hợp dữ liệu

Dữ liệu sau quá trình trích xuất đặc trưng từ các tập dữ liệu nhỏ sẽ được tổng hợp thành một bộ dữ liệu lớn cho quá trình đào tạo thông qua phương thức hợp nhất (merge) dựa trên thuộc tính SK_ID_CURR (Mã khách hàng đăng ký khoản vay). Sau khi hợp nhất mỗi SK_ID_CURR sẽ có các thông tin đầy đủ từ dữ liệu cơ bản và các thuộc tính trong quá trình trích xuất đặc trưng.

Dữ liệu này sẽ là nguồn dữ liệu chính cho quá trình đào tạo các mô hình.

CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT QUẢ

4.1. Thực nghiệm

4.1.1. Dữ liệu

Dữ liệu đã được kết hợp từ quá trình chuẩn bị dữ liệu được chia làm 2 phần là dữ liệu huấn luyện (train set) và dữ liệu đánh giá (test set) bao gồm:

- Train set: 356251 mẫu và 781 thuộc tính.
- Test set: 48744 mẫu và 781 thuộc tính.

Train set được sử dụng cho quá trình huấn luyện các mô hình, Test set sẽ sử dụng kết quả dự đoán của mô hình sau khi huấn luyện để dự đoán giá trị xác suất của biến mục tiêu cho các mẫu có trong tập dữ liệu sau đó nộp và đánh giá bằng chỉ số đánh giá của cuộc thi trên nền tảng Kaggle.

4.1.2. Huấn luyện mô hình

4.1.2.1. Mô hình logistic regression

4.1.2.1.1. Tham số huấn luyện

Bảng 4.1 Tham số mô hình Logistic regression

Tham số	**Chú thích**	**Giá trị**
random_state	Vùng xáo trộn dữ liệu	0
class_weight	Chế độ liên kết của các trọng số.	balanced
C	Chỉ số nghịch đảo của cường độ chỉnh hóa	1.0

4.1.2.1.2. Công cụ huấn luyện

Mô hình được tạo bằng ngôn ngữ lập trình python trên trình soạn thảo mã của Kaggle (Kaggle notebook).

Cấu hình phần cứng:

- CPU: Intel Xeon 2-core
- RAM: 30 GB
- ROM: 73 GB

4.1.2.2. Mô hình random forest

4.1.2.2.1. Tham số huấn luyện

Bảng 4.2 Tham số mô hình Random forest

Tham số	**Chú thích**	**Giá trị**
n_estimators	Số cây trong tập hợp	50
class_weight	Chế độ liên kết của các trọng số.	balanced
criterion	Chỉ số đo lường phân tách	gini
max_depth	Độ sâu tối đa của cây	5
min_samples_split	Số lượng mẫu cần thiết để phân tách nốt	2
min_samples_leaf	Số lượng mẫu tối thiểu cần có tại một nút lá.	1
max_features	Phương thức tìm kiếm sự phân chia	auto
bootstrap	Sử dụng các mẫu bootstrap	True

oob_score	Sử dụng các mẫu bên ngoài để ước tính điểm tổng quát	False
n_jobs	Số lượng bộ xử lý chạy song song. -1 có nghĩa là sử dụng tất cả các bộ xử lý	-1
random_state	Vùng xáo trộn dữ liệu	0
verbose	Kiểm soát mức độ chi tiết khi điều chỉnh mô hình và dự đoán	0
warm_start	Sử dụng thêm các công cụ ước tính từ lần điều chỉnh trước đó	False

4.1.2.2.2. Công cụ huấn luyện

Mô hình được tạo bằng ngôn ngữ lập trình python trên trình soạn thảo mã của Kaggle (Kaggle notebook).

Cấu hình phần cứng:

- CPU: Intel Xeon 2-core
- RAM: 30 GB
- ROM: 73 GB

4.1.2.3. Mô hình neural network

4.1.2.3.1. Tham số huấn luyện

Bảng 4.3 Tham số huấn luyện mô hình Neural network

Tham số	**Chú thích**	**Giá trị**
epochs	Số chu kỳ huấn luyện mạng.	300
batch_size	Số lượng mẫu sẽ được truyền qua mạng	4096
verbose	Cách xem tiến trình đào tạo	1

4.1.2.3.2. Công cụ huấn luyện

Mô hình được tạo bằng ngôn ngữ lập trình python trên trình soạn thảo mã của Kaggle (Kaggle notebook) [109].

Cấu hình phần cứng:

- CPU: Intel Xeon 2-core
- RAM: 13 GB
- ROM: 73 GB
- GPU: P100 - 16GB VRAM

4.1.2.4. Mô hình XGBoost

4.1.2.4.1. Tham số huấn luyện

Bảng 4.4 Tham số huấn luyện mô hình XGBoost

Tham số	**Chú thích**	**Giá trị**
learning_rate	Tốc độ học	0.01
n_estimators*	Số cây trong tập hợp	10000, 15000, 20000

max_depth*	Độ sâu tối đa của cây	4, 6, 8
min_child_weight*	Tổng trọng số tối thiểu của các mẫu quan sát	5, 7, 9
subsample	Tỷ lệ mẫu đào tạo	0.8
colsample_bytree	Tỷ lệ thuộc tính xây dựng cây	0.8
objective	Nhiệm vụ học tập và mục tiêu học tập	′binary: logistic′
tree_method	Phương thức huấn luyện	′gpu_hist′
booster	Sử dụng tăng tốc đào tạo	′gbtree′
nthread	Số luồng chạy song song khi huấn luyện	4
scale_pos_weight	Chỉ số kiểm soát cân bằng giữa các lớp	2.5
reg_lambda	Chỉ số chỉnh hóa	1.2
verbose	Chỉ số để mô hình đưa ra đánh giá sau n vòng (n đầu vào)	100
early_stopping_rounds	Mô hình sẽ dừng lại nếu các chỉ số không thay đổi sau n vòng (n đầu vào)	200
eval_metric	Chỉ số đánh giá mô hình	′auc′, ′logloss′

Các chỉ số được gắn * là các chỉ số được gắn nhiều giá trị. Mỗi giá trị sẽ được huấn luyện một lần và tìm ra chỉ số tối ưu.

4.1.2.4.2. Công cụ huấn luyện

Mô hình được tạo bằng ngôn ngữ lập trình python trên trình soạn thảo mã của Kaggle (Kaggle notebook).

Cấu hình phần cứng:

- CPU: Intel Xeon 2-core
- RAM: 13 GB
- ROM: 73 GB
- GPU: P100 - 16GB VRAM

4.1.2.5. Mô hình LightBoost

4.1.2.5.1. Tham số huấn luyện

Bảng 4.5 Tham số huấn luyện mô hình LightGBM

Tham số	**Chú thích**	**Giá trị**
learning_rate*	Tốc độ học	0.01, 0.015, 0.02
n_estimators*	Số cây trong tập hợp	5000, 7000, 10000
max_depth*	Độ sâu tối đa của cây	7, 9, 15
min_child_weight	Tổng trọng số tối thiểu của các mẫu quan sát	39
subsample	Tỷ lệ mẫu đào tạo	0.87
colsample_bytree	Tỷ lệ thuộc tính xây dựng cây	0.94

reg_lambda	Chỉ số chỉnh hóa L2	0.041
n_jobs	Sử dụng luồng song song khi huấn luyện	−1
num_leaves	Số lá tối đa trên một cây	100
reg_alpha	Chỉ số chỉnh hóa L1	0.041
min_split_gain	Chỉ số phân chia	0.022
device	Cấu hình sử dụng thiết bị phần cứng	'gpu'
verbose	Chỉ số để mô hình đưa ra đánh giá sau n vòng (n đầu vào)	100
early_stopping_rounds	Mô hình sẽ dừng lại nếu các chỉ số không thay đổi sau n vòng (n đầu vào)	200
eval_metric	Chỉ số đánh giá mô hình	'auc', 'logloss'

Các chỉ số được gắn * là các chỉ số được gắn nhiều giá trị. Mỗi giá trị sẽ được huấn luyện một lần và tìm ra chỉ số tối ưu.

4.1.2.5.2. Công cụ huấn luyện

Mô hình được tạo bằng ngôn ngữ lập trình python trên trình soạn thảo mã của Kaggle (Kaggle notebook).

Cấu hình phần cứng:

- CPU: Intel Xeon 2-core
- RAM: 13 GB

- ROM: 73 GB
- GPU: P100 - 16GB VRAM

4.1.2.6. Mô hình CatBoost

4.1.2.6.1. Tham số huấn luyện

Bảng 4.6 Tham số huấn luyện mô hình CatBoost

Tham số	**Chú thích**	**Giá trị**
learning_rate*	Tốc độ học	0.015, 0.02, 0.05
Iterations*	Số cây trong tập hợp	1000, 3000, 5000
Depth*	Độ sâu tối đa của cây	5, 7, 9
l2_leaf_reg	Chỉ số chỉnh hóa L2	40
bootstrap_type	Phương pháp lấy mẫu trọng số của các thuộc tính	'Bernoulli'
subsample	Số lượng mẫu đóng gói	0.7
scale_pos_weight	Hệ số nhân cho trọng số	5
eval_metric	Chỉ số đánh giá	'AUC'
loss_function	Chỉ số cho hàm mất mát	'Logloss'
metric_period	Tần suất lặp	50
od_type	Chế độ phát hiện overfitting	'Iter'

od_wait	Mô hình sẽ dừng lại nếu các chỉ số không thay đổi sau n vòng (n đầu vào)	45
random_seed	Vùng lấy mẫu	17
allow_writing_files	Ghi lại các chỉ số phân tích	False

Các chỉ số được gắn * là các chỉ số được gắn nhiều giá trị. Mỗi giá trị sẽ được huấn luyện một lần và tìm ra chỉ số tối ưu.

4.1.2.6.2. Công cụ huấn luyện

Mô hình được tạo bằng ngôn ngữ lập trình python trên trình soạn thảo mã của Kaggle (Kaggle notebook).

Cấu hình phần cứng:

- CPU: Intel Xeon 2-core
- RAM: 13 GB
- ROM: 73 GB
- GPU: P100 - 16GB VRAM

4.1.2.7. Huấn luyện

4.1.2.7.1. Phương pháp huấn luyện

Ban đầu huấn luyện mô hình sử dụng kỹ thuật cross-validation với 10-fold. Sau khi huấn luyện đủ số vòng lặp (iteration) sẽ đưa ra đánh giá mô hình nếu kết quả mô hình đạt được kỳ vọng sẽ huấn luyện lại mô hình trên toàn bộ dữ liệu, sau đó đánh giá mô hình được huấn luyện trên toàn bộ dữ liệu.

4.1.2.7.2. Thời gian huấn luyện

Table 4.1 Thời gian huấn luyện của các mô hình

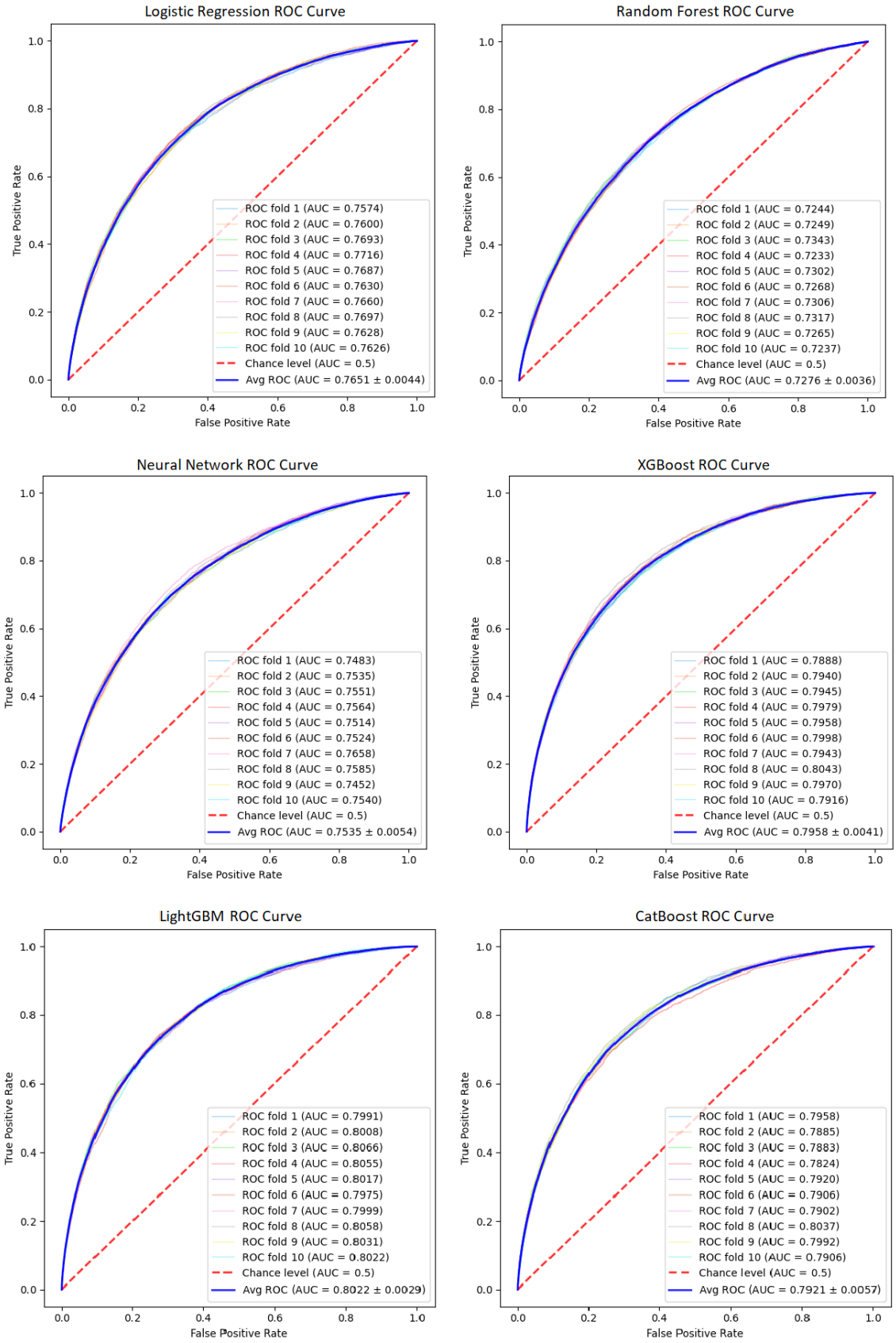
Mô hình	**Thời gian huấn luyện**
Logistic regression	2 tiếng 37 phút
Random forest	3 tiếng 54 phút
Neural network	4 tiếng 19 phút
XGBoost	6 tiếng 22 phút
LightGBM	3 tiếng 48 phút
CatBoost	5 tiếng 35 phút

Mô hình logistic regression cho thời gian huấn luyện nhanh nhất. Ngược lại, XGBoost là mô hình có thời gian huấn luyện lâu nhất nhất trong các mô hình được huấn luyện. Trong 3 mô hình sử dụng kỹ thuật gradient boosting gồm XGBoost, LightGBM, CatBoost mô hình LightGBM có tốc độ huấn luyện nhanh nhất, kết quả này minh chứng cho ưu điểm về tốc độ huấn luyện mô hình của LightGBM đã được đưa ra ở phần tổng quan thuật toán (*2.4.5.1*) khi có tốc độ đào tạo nhanh hơn nhiều so với các thuật toán sử dụng kỹ thuật gradient boosting. LightGBM có được điều này nhờ sử dụng kỹ thuật xây dựng mô hình GBDT và kỹ thuật lấy mẫu GOSS (*2.4.5.2*).

4.2. Kết quả

4.2.1. Kết quả huấn luyện cross-validation

Quá trình huấn luyện các mô hình được thực hiện dựa trên kỹ thuật cross-validation với 10-fold. Kết quả của các mô hình được trình bày trong hình 4.1 và bảng 4.7.



Hình 4.1 Đồ thị biểu diễn đường cong ROC qua các vòng lặp của các mô hình huấn luyện bằng kỹ thuật cross-validation

Bảng 4.7 Kết quả các mô hình huấn luyện bằng kỹ thuật cross-validation

Mô hình	$\mathbf{mean_{AUC}}$	$\mathbf{std_{AUC}}$
Logistic regression	0.7651	0.0044
Random forest	0.7276	0.0036
Neural network	0.7535	0.0054
XGBoost	0.7958	0.0041
LightGBM	0.8022	0.0029
CatBoost	0.7921	0.0057

Từ các đồ thị biểu diễn đường cong ROC qua từng vòng lặp huấn luyện của các mô hình cho thấy hiệu suất các mô hình trên các vòng lặp huấn luyện (iteration) có độ nhất quán cao, điều này cũng được thể hiện ở giá trị độ lệch chuẩn AUC ($\mathrm{std_{AUC}}$) của các mô hình khi mà các mô hình có độ lệch chuẩn AUC rất thấp. Bước đầu, điều này cho thấy mô hình có khả năng khái quát hóa cao trên toàn bộ dữ liệu huấn luyện.

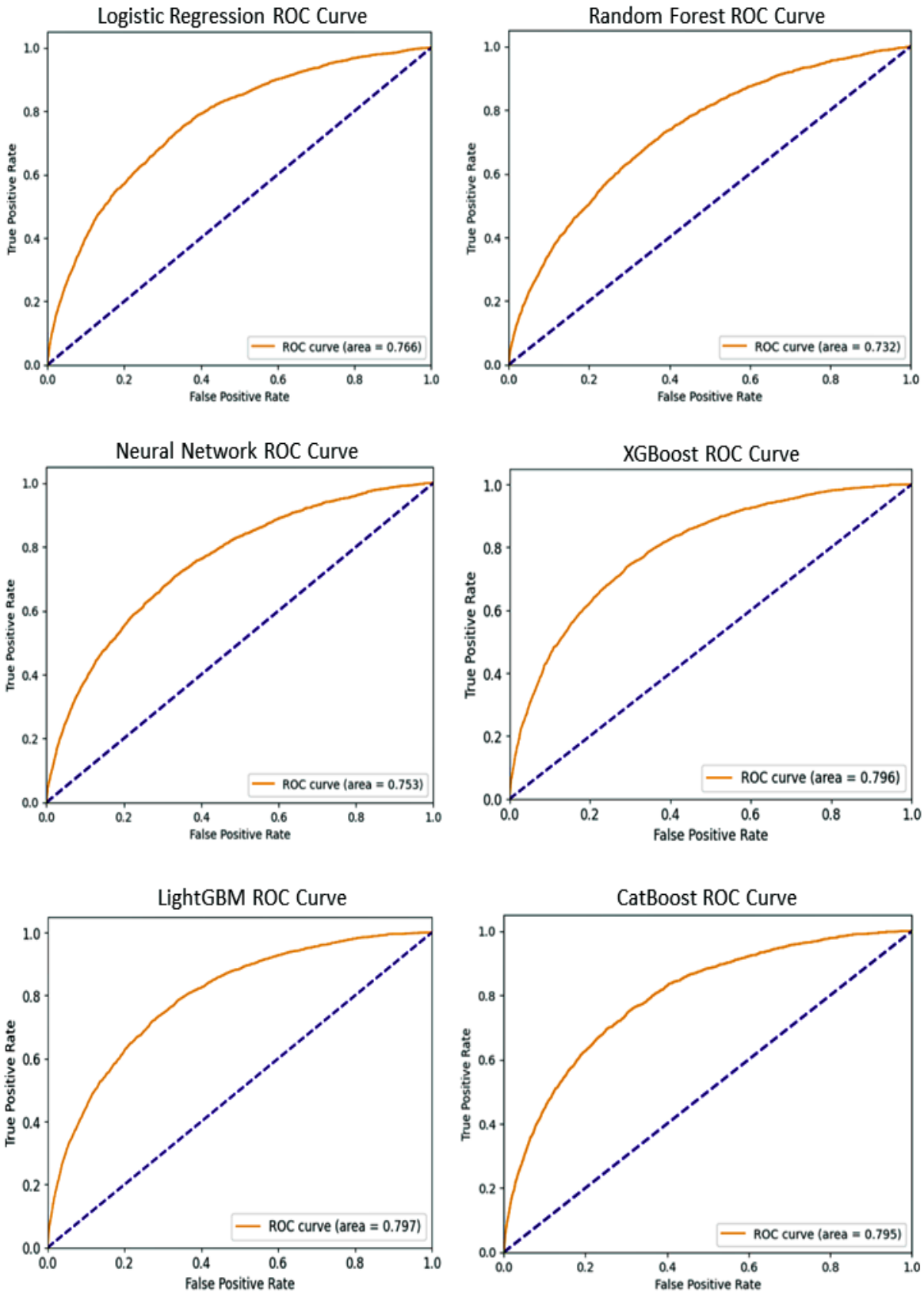
Giá trị trung bình ($\mathrm{mean_{AUC}}$) cao nhất đạt được ở mô hình LightGBM, mô hình này cũng có giá trị độ lệch chuẩn AUC thấp nhất trong các mô hình đã huấn luyện. Mô hình random forest có giá trị trung bình của chỉ số AUC thấp nhất. Độ lệch chuẩn của chỉ số AUC cao nhất ở mô hình CatBoost.

4.2.2. Kết quả huấn luyện trên toàn bộ dữ liệu

Sau khi huấn luyện mô hình bằng kỹ thuật cross-validation, nhận thấy chất lượng mô hình đồng đều trên toàn bộ iteration. Tiếp tục tiến hành huấn luyện lại mô hình trên toàn bộ dữ liệu. Các chỉ số đánh giá của quá trình huấn luyện này được đề cập dưới đây:

Bảng 4.8 Chỉ số đánh giá trên biến phân loại

Mô hình	Biến phân loại	Precision	Recall	F1-score
Logistic regression	0	0.95	0.82	0.88
	1	0.21	0.35	0.31
Random forest	0	0.93	0.96	0.94
	1	0.30	0.20	0.24
Neural network	0	0.92	1.00	0.96
	1	0.28	0.06	0.08
XGBoost	0	0.94	0.99	0.96
	1	0.45	0.09	0.16
LightGBM	0	0.93	1.00	0.96
	1	0.54	0.02	0.03
CatBoost	0	0.94	0.97	0.96
	1	0.57	0.32	0.27



Hình 4.2 Đồ thị biểu diễn đường cong ROC của các mô hình huấn luyện trên toàn bộ dữ liệu

Bảng 4.9 Kết quả tổng hợp chỉ số đánh giá mô hình

Mô hình	**Accuracy rate (%)**	**Recall rate (%)**	**Precision Rate (%)**	**Specificity Rate (%)**	**F1 Score**	**ROC AUC**
Logistic regression	79.4673	22.0183	21.3050	81.6223	0.3078	0.7662
Random forest	89.5841	20.0632	30.1126	95.8220	0.2408	0.7323
Neural network	91.9287	13.795	27.9842	92.3761	0.2015	0.7528
XGBoost	92.8227	9.3738	45.1923	99.1395	0.1553	0.7959
LightGBM	92.983	33.6278	54.217	99.885	0.3475	0.7969
CatBoost	91.8122	55.4502	36.4597	97.0953	0.2746	0.7951

Vì dữ liệu có sự mất cân bằng lớn khi tỷ lệ giá trị phân loại là 0 (khách hàng không có nguy cơ vỡ nợ) chiếm tỷ lệ lớn trong bộ dữ liệu gây khó khăn cho việc mô hình có thể dự đoán tốt cho giá trị phân loại là 1 (khách hàng có nguy cơ vỡ nợ) nên với các chỉ số đánh giá trên từng biến phân loại (*Bảng 4.8*) sẽ quan tâm nhiều hơn đến các chỉ số đánh giá trên giá trị phân loại là 1. Từ các chỉ số đánh giá mô hình dự đoán trên 2 biến phân loại có thể thấy mô hình CatBoost cho chất lượng phân loại tốt nhất trên các giá trị phân loại là 1 đồng nghĩa với việc mô hình này bao quát tốt được thông tin từ các thuộc tính của biến mục tiêu phân loại là 1 trong bộ dữ liệu. Ngoài ra, mô hình XGBoost và LightGBM cũng đạt chất lượng gần bằng với mô hình CatBoost. Các mô hình còn lại có có chỉ số đánh giá chất lượng

dự đoán giá trị phân loại là 1 chưa tốt, các mô hình này có thể phải cần nhiều hơn dữ liệu có giá trị phân loại là 1.

Khi đánh giá chung các mô hình dựa trên tất cả các giá trị dự đoán trong *bảng 4.9* cũng cho thấy mô hình CatBoost có chất lượng các kết quả dự đoán là tốt nhất khi phần lớn các chỉ số đánh giá chất lượng của mô hình này cao hơn so với các mô hình còn lại. Tương tự như kết quả đánh giá trên giá trị phân loại là 1, mô hình XGBoost và mô hình LightGBM cũng có giá trị các chỉ số gần bằng mô hình CatBoost và cao hơn các mô hình còn lại. Các chỉ số đánh giá trên mô hình logistic regression là khá thấp, các chỉ số đánh giá của mô hình Random forest và Neural network gần tương đương nhau.

Chỉ số ROC AUC cho mức cao nhất ở mô hình LightGBM tuy nhiên giá trị này không quá chênh lệch ở cả 3 mô hình XGBoost, LightGBM và CatBoost lần lượt là $0.7959, 0.7969, 0.7951$. Chỉ số này thấp hơn ở các mô hình Logistic regression, Random forest và Neural network với giá trị thấp nhất ở mô hình Random forest (0.7662).

Từ các chỉ số đánh giá chất lượng của các mô hình được huấn luyện trên toàn bộ dữ liệu, nhìn chung, XGBoost, LightGBM và CatBoost có kết quả tốt và đạt được kỳ vọng đề ra trong đó mô hình CatBoost tốt nhất trên tổng quan các chỉ số tuy nhiên không quá vượt trội so với 2 mô hình còn lại là XGBoost, LightGBM. Các mô hình, Logistic regression, Random forest và Neural network cho kết quả tốt trên một số chỉ số như accuracy rate, specificity rate tuy nhiên các chỉ số còn lại khá thấp, có thể các mô hình đã khái quát tốt các thông tin về dữ liệu của các giá trị phân lớp là 0 nhưng chưa tốt trên các giá trị phân lớp là 1 điều này đến từ việc mất cân bằng của dữ liệu đầu vào.

4.2.3. Kết quả điểm LB

Các mô hình sau khi huấn luyện sẽ dự đoán các dữ liệu trên tập kiểm thử được đính kèm trong bộ dữ liệu với các biến mục tiêu bị thiếu. Sau khi dự đoán giá trị cho các biến mục tiêu sẽ nộp kết quả lên hệ thống cuộc thi đã đưa ra bài toán và bộ dữ liệu.

Các chỉ số đánh giá của cuộc thi dựa trên kết quả dự đoán của mô hình được đưa ra dưới đây:

Bảng 4.10 Chỉ số đánh giá LB Kaggle của các mô hình

Mô hình	**Private score**	**Public score**
Logistic regression	0.75805	0.76044
Random forest	0.71982	0.71719
Neural network	0.75303	0.75098
XGBoost	0.77623	0.77708
LightGBM	0.76562	0.77291
CatBoost	0.79095	0.79598

Mô hình CatBoost vẫn là mô hình đạt kết quả tốt nhất và có kết quả khá vượt trội so với các mô hình còn lại. XGBoost và LightGBM vẫn là 2 mô hình có kết quả tốt. Mô hình Logistic regression tuy được đánh giá khá thấp trên các chỉ số chất lượng ở phần *4.2.2* tuy nhiên lại cho kết quả khả quan ở điểm đánh giá LB. Trong đó mô hình Random forest có điểm đánh giá LB thấp nhất và thấp hơn hẳn so với các mô hình còn lại.

4.2.4. Kết quả mô hình học đồng bộ

Sau quá trình đánh giá chất lượng của các mô hình huấn luyện, từ 3 mô hình có hiệu suất tốt nhất bao gồm XGBoost, LightGBM và CatBoost áp dụng phương pháp học đồng bộ (ensemble learning) với phương thức voting để tạo ra kết quả dự đoán mới.

Với thời gian huấn luyện 8 tiếng 13 phút, kết quả của phương thức học đồng bộ được đưa ra dưới đây:

Bảng 4.11 Chỉ số đánh giá trên biến phân loại của mô hình kết hợp sử dụng phương pháp ensemble learning

Biến phân loại	Precision	Recall	F1-score
0	0.96	0.94	0.93
1	0.59	0.33	0.29

Bảng 4.12 Kết quả tổng hợp chỉ số đánh giá mô hình kết hợp sử dụng phương pháp ensemble learning

Accuracy rate (%)	Recall rate (%)	Precision Rate (%)	Specificity Rate (%)	F1 Score	ROC AUC	Private score	Public score
93.7244	59.1632	28.0749	94.1562	0.3016	0.8293	0.7991	0.8095

Từ các chỉ số đánh giá chất lượng mô hình học đồng bộ với phương pháp voting (*bảng 1.10, bảng 4.11*) có thể thấy mô hình có cải thiện được chất lượng so với các mô hình đã huấn luyện. Các chỉ số đánh giá chất lượng dự đoán giá trị phân loại là 1 và các chỉ số đánh giá trên toàn bộ mô hình tốt hơn so với mô hình dự đoán tốt nhất đã được huấn luyện là CatBoost. Private score và public score đánh giá trên mô hình lần lượt là 0.7996, 0.8052 cũng tốt hơn so với mô hình CatBoost với private score 0.7909 và public score 0.7959.

Từ những chỉ số và đánh giá trên cho thấy việc xây dựng mô hình sử dụng phương pháp học đồng bộ giúp cải thiện chất lượng dự đoán của các mô hình độc lập. Tuy nhiên, chất lượng mô hình học đồng bộ chưa thực sự vượt trội so với các mô hình độc lập khi so với mô hình tốt nhất thì các chỉ số đánh giá chất lượng và điểm đánh giá LB chỉ tăng nhẹ. Điều này có thể đến từ việc các mô hình độc lập có sự chênh lệch

trong chất lượng đào tạo hoặc việc sử dụng phương pháp voting trong học đồng bộ chưa kết hợp tốt các dự đoán của những mô hình độc lập.

4.2.5. Điểm tín dụng

Khi đã xây dựng được các mô hình và đánh giá sẽ chọn ra mô hình có kết quả tốt nhất để dự đoán, các giá trị này sẽ được chuyển đổi thành điểm tín dụng dựa trên phương pháp chuyển đổi điểm tín dụng (*2.5.7*). Trong phần này cũng đưa ra các dự đoán lớp phân loại của mô hình với ngưỡng phân loại là 0.6 được tính toán dựa trên phương pháp tìm ngưỡng phân loại (*2.5.6*) và các giá trị phân loại thực tế.

Kết quả của một số dữ liệu được dự đoán và chuyển đổi thành điểm tín dụng được trình bày dưới đây:

Bảng 4.13 Giá trị dự đoán của mô hình CatBoost được chuyển đổi thành điểm tín dụng của 10 mẫu dữ liệu với ngưỡng phân loại 0.6

STT	ID	Xác suất dự đoán	Nhãn thực tế	Nhãn dự đoán	Điểm tín dụng
1	439073	0.6089	1	1	515
2	415124	0.5912	1	0	524
3	347952	0.7674	1	1	427
4	158208	0.6917	1	1	469
5	453230	0.5842	1	0	528
6	183658	0.5405	0	0	552
7	144091	0.5102	0	0	569
8	139861	0.4907	0	0	580
9	199727	0.2175	0	0	730
10	236737	0.0251	0	0	836

Trong 10 mẫu dữ liệu đã đưa ra có 5 mẫu mang nhãn phân lớp là 0 và 5 mẫu mang nhãn phân lớp là 1. Mô hình đã phân lớp đúng hầu hết các mẫu dữ liệu với ngưỡng phân loại tìm được dựa trên phương pháp tìm ngưỡng phân loại.

4.2.6. Thuộc tính quan trọng

Ngoài kết quả dự đoán, các mô hình sẽ đưa các thông tin về các thuộc tính quan trọng ảnh hưởng đến kết quả phân loại. Các thuộc tính dưới đây được tổng hợp từ 3 mô hình tốt nhất là XGBoost, LightGBM và CatBoost.

Bảng 4.14 Các thuộc tính quan trọng của mô hình

Thuộc tính	**Chú thích**
APP_NAME_EDUCATION_TYPE_Higher_education	Khách hàng cấp giáo dục bậc cao
APP_EXT_SOURCE_2	Phân nhóm theo ext_source_1
APP_EXT_SOURCE_3	Phân nhóm theo ext_source_2
APP_NAME_INCOME_TYPE_Working	Khách hàng có nguồn thu nhập từ lao động
APP_NAME_INCOME_TYPE_Pensioner	Khách hàng có nguồn thu nhập từ hưu trí
APP_CODE_GENDER	Giới tính khách hàng
PREV_INS_NEW_NUM_PAID_LATER_SUM	Số lần trả muộn khoản trả góp
CCB_CNT_DRAWINGS_ATM_CURRENT_MEAN	Trung bình số tiền rút từ ATM

PREV_NAME_CONTRACT_STATUS_Refused_MEAN	Trung bình số lượng hợp đồng xoay vòng
APP_NAME_EDUCATION_TYPE_Secondary	Khách hàng có cấp độ giáo dục cao nhất là giáo dục trung học
APP_FLAG_DOCUMENT_3	Khách hàng cung cấp tài liệu loại 3
PREV_INS_AMT_PAYMENT_MIN_SUM	Tổng số tiền khách hàng thanh toán trả góp mức tối thiểu
APP_EXT_SOURCE_1	Phân nhóm theo ext_source_3
APP_FLAG_OWN_CAR	Khách hàng sở hữu ô tô
BB_MONTHS_BALANCE_COUNT	Số tháng có số dư tín dụng
BB_NEW_CLOSED_AMT_ANNUITY_MEAN	Số tiền trung bình hàng năm của các hợp đồng có trạng thái đã đóng
APP_AMT_REQ_CREDIT_BUREAU_QRT	Số lượng thắc mắc gửi về phong tín dụng của khách hàng trước 3 tháng nộp đơn
APP_FLAG_EMP_PHONE	Khách hàng cung cấp số điện thoại cơ quan

APP_REGION_RATING_CLIENT_W_CITY	Xếp hạng khu vực sinh sống tính đến thành phố
BB_AMT_CREDIT_SUM_OVERDUE_SUM	Tổng số tiền khách hàng trả muộn

4.3. Kết luận

Sau quá trình thực nghiệm các mô hình trên dữ liệu Home Credit Default Risk đã qua xử lý và trích xuất đặc trưng. Một số kết luận được rút ra như sau:

- Quá trình trích xuất đặc trưng từ dữ liệu gốc là quá trình rất quan trọng. Đây là nền tảng chất lượng cho quá trình huấn luyện mô hình.
- Mô hình CatBoost có chất lượng tốt nhất trong các mô hình dự đoán. Bên cạnh đó, mô hình LightGBM và XGBoost cũng có kết quả tốt.
- Mô hình LightGBM nên được sử dụng cho các bài toán cần tối ưu thời gian huấn luyện nhưng vẫn mang lại hiệu suất mô hình đủ tốt trên các loại dữ liệu tương tự.
- Do dữ liệu mất cân bằng nên mô hình Logistic regression và Random forest không đạt kết quả cao.
- Deep learning không phù hợp với loại dữ liệu này khi mô hình Neural network không đạt kết quả tốt.
- Ensemble learning giúp cải thiện chất lượng dự đoán của mô hình, tuy nhiên chất lượng cải thiện chưa thực sự vượt trội.
- Kết quả huấn luyện mô hình từ kỹ thuật cross-validation và LB score không đồng nhất khi mà kết quả trên tập huấn luyện nhận thấy AUC tăng nhưng khi nộp lên hệ thống của cuộc thi thì public score và private score có thể giảm xuống.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả

5.1.1. Kết quả đạt được

5.1.1.1. Kết quả bài toán

Sau quá trình nghiên cứu và thực nghiệm đã xây dựng thành công các mô hình chấm điểm tín dụng dựa trên các thuật toán nổi bật trong học máy và đưa ra được mô hình tốt nhất với chỉ số đánh giá độ chính xác xấp xỉ 80% và có thứ hạng cao trên bảng xếp hạng hiệu suất mô hình của cuộc thi được tổ chức. Mô hình chấm điểm tín dụng đưa ra có kết quả khá tốt và cho thấy mức độ khả quan cao về việc ứng dụng mô hình vào bối cảnh thực tế mà đề tài đã đưa ra từ phần tổng quan nghiên cứu. Ngoài ra, mô hình còn cho thấy tiềm năng cao khi ứng dụng được với các mô hình chấm điểm tín dụng truyền thống, điều này có ý nghĩa thực tiễn rất lớn giúp cho các ngân hàng, các định chế tài chính, các tổ chức tín dụng có thể đánh giá triệt để được các rủi ro tín dụng có liên quan đến hoạt động cho vay.

5.1.1.2. Kiến thức và kỹ năng

Trong suốt quá trình nghiên cứu và thực hiện đề tài này, bản thân đã đạt được một số kết quả:

- Hiểu được tầm quan trọng của bài toán chấm điểm tín dụng và các kiến thức chuyên ngành tài chính tín dụng liên quan.
- Nắm được các lý thuyết về các phương pháp, kỹ thuật của các thuật toán Logistic regression, Random forest, XGBoost, LightGBM, CatBoost, Neural Networks cùng với đó là các kiến thức liên quan đến việc tìm hiểu bài toán, chuẩn bị dữ liệu, huấn luyện, kết hợp mô hình, đánh giá mô hình và tạo điểm tín dụng từ kết quả đầu ra của mô hình.
- Thực nghiệm xây dựng và so sánh được các mô hình dữ liệu từ các thuật toán trên.
- Trong quá trình thực hiện luận văn đã giúp bản thân nâng cao được khả năng đọc hiểu và tham khảo các bài nghiên cứu khoa học từ đó giúp cung cấp được những nền tảng kiến thức trong quá trình thực hiện đề tài.

- Bản thân cũng nâng cao được khả năng trình bày, soạn thảo và báo cáo các nội dung nghiên cứu và thực nghiệm.

5.2. Hướng phát triển

Việc áp dụng các thuật toán học máy vào mô hình chấm điểm tín dụng cho thấy kết quả khả thi và mang tính ứng dụng cao. Trong tương lai, ngoài những kết quả đã nghiên cứu và thực nghiệm sẽ định hướng phát triển đề tài với các mục tiêu:

- Nghiên cứu thêm về các phương pháp khác có liên quan đến xây dựng mô hình chấm điểm tín dụng và các thuật toán học máy. Bên cạnh đó tiếp tục cập nhật các phương pháp thuật toán mới nhằm đưa ra được mô hình chất lượng hơn.
- Thực nghiệm các thuật toán với một số bộ dữ liệu khác có liên quan đến cho vay tín dụng như *Lending Club Loan Data* [110], *default of credit card clients* [111], … từ đó nghiên cứu về mô hình chấm điểm tín dụng với quy mô rộng hơn và khái quát hơn về nhiều mẫu dữ liệu khác nhau.
- Triển khai mô hình đã huấn luyện lên nền tảng web để phù hợp hơn với định hướng phát triển mô hình cho người dùng sử dụng.

TÀI LIỆU THAM KHẢO

[1] "Tạp chí tài chính," 21 11 2021. [Online]. Available: https://tapchitaichinh.vn/giai-phap-tin-dung-va-xu-ly-no-xau-trong-dieu-kien-nen-kinh-te-bi-tac-dong-boi-dai-dich-covid-19.html.

[2] G. T. T. Huyền, "Một số kĩ thuật học máy cho chấm điểm tín dụng," *Tạp chí Khoa học & Đào tạo Ngân hàng,* no. 227, pp. 34-40, 2021.

[3] M. A. Qureshi, "Credit Scoring and Its Applications in Banking: A Comprehensive Review," *Journal of Risk and Financial Management,* 2020.

[4] https://www.investopedia.com/terms/f/ficoscore.asp, "Investopedia," Fair Isaac Corporation, 18 February 2023. [Online]. Available: https://www.investopedia.com/terms/f/ficoscore.asp.

[5] H. Sargeant, "Algorithmic decision-making in financial services: economic and normative outcomes in consumer credit," *AI and Ethics,* 2022.

[6] "The Council on Foreign Relations," 19 February 2015. [Online]. Available: https://www.cfr.org/backgrounder/credit-rating-controversy.

[7] S. Khemakhem and Y. Boujelbene, "Predicting credit risk on the basis of financial and non-financial variables and data mining," *Review of Accounting and Finance,* 2018.

[8] "Home Credit Default Risk," Group Home Credit, [Online]. Available: https://www.kaggle.com/competitions/home-credit-default-risk/data.

[9] T. Tamplin, "Finance Strategists," Finance Strategists, March 2023. [Online]. Available:

https://www.financestrategists.com/banking/credit-score/?gclid=Cj0KCQjw0tKiBhC6ARIsAAOXutmxWG5jpfc1eJ3dfaBte6PeMwYtsbilgOatm1KD_EPoUc91_yytbq4aAsvgEALw_wcB.

[10] P. G. M. K. Brevoort Kenneth, "Data Point: Credit Invisibles," *Consumer Financial Protection Bureau,* 2015.

[11] F. R. B. o. Philadelphia, "The Long-Term Effects of Credit Scores on Credit Market Outcomes," 2015.

[12] S. J. Hand David, Statistics in Finance (Arnold Applications of Statistics Series), London: Wiley, 1998.

[13] "FICO," Fair Isaac Corporation, [Online]. Available: https://www.fico.com/en/products/fico-score.

[14] A. Hayes, "investopedia," February 2023. [Online]. Available: https://www.investopedia.com/terms/f/ficoscore.asp.

[15] B. Fav, "What is a Credit Score & How is it Calculated?," America's Debt Help, [Online]. Available: https://www.debt.org/credit/report/scoring-models/.

[16] "The VantageScore Model - Empowering lenders to make confident and inclusive lending decisions.," VantageScore, 2023. [Online]. Available: https://www.vantagescore.com/lenders/why-vantagescore/our-models/.

[17] G. Z. J. L. Y Zhu, "The impact of alternative data sources on credit scoring models," *Journal of Business Research,* pp. 205-215, 2019.

[18] Y. L. X. L. H Lin, "Incorporating social media data into credit scoring models," *Decision Support Systems,* pp. 78-88, 2019.

[19] B. Bhattacharya, "Credit risk assessment: The impact of macroeconomic factors," *Journal of Risk Research,* pp. 1098-1119, 2019.

[20] "IBM," [Online]. Available: https://www.ibm.com/topics/machine-learning.

[21] R. W. Picard, E. Vyzas and J. Healey, "Toward machine emotional intelligence: Analysis of affective physiological state," *IEEE transactions on pattern analysis and machine intelligence,* vol. 23, pp. 1175-1191, 2001.

[22] A. Saxe, S. Nelli and C. Summerfield, "If deep learning is the answer, what is the question?," *Nature Reviews Neuroscience,* vol. 22, pp. 55-67, 2021.

[23] G. Xiao, J. Li, Y. Chen and K. Li, "MalFCS: An effective malware classification framework with automated feature extraction based on deep convolutional neural networks," *Journal of Parallel and Distributed Computing,* vol. 141, pp. 49-58, 2020.

[24] W. D. W. D. Luo Cuicui, "A deep learning approach for credit scoring using credit default swaps," *Engineering Applications of Artificial Intelligence,* vol. 65, pp. 465-470, 2017.

[25] H. Z. W. S. J Jiang, "Credit risk assessment based on neural networks," *International Journal of Simulation: Systems, Science and Technology,* vol. 19(14), pp. 1-8, 2018.

[26] A. T. Bouzouita, "A machine learning approach to credit scoring: Evidence from SMEs," *Journal of Business Research,* vol. 117, pp. 807-816, 2020.

[27] D. Zhang, X. Zhou, S. C. Leung, Zheng and Jiemin, "Vertical bagging decision trees model for credit scoring," *Expert Systems with Applications,* vol. 37, pp. 7838-7843, 2010.

[28] D. Lachos-Perez, S. Dussán-Sarria and L. Giraldo-Gómez, "Bayesian variable selection and classification model for credit scoring," *Expert Systems with Applications,* 2021.

[29] LaValley and M. P, "Logistic regression," *Circulation,* vol. 117, pp. 2395-2399, 2008.

[30] C. Gavin, T. Nicola and G. Mark, "Sparse multinomial logistic regression via bayesian l1 regularisation," *Advances in neural information processing systems,* vol. 19, 2006.

[31] L. Rokach, "Ensemble methods for classifiers," *Data mining and knowledge discovery handbook,* pp. 957-980, 2005.

[32] S. Omer and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 8, p. 1249, 2018.

[33] F. Khaled , M. M. Gaber and E. Elyan, "Random forests: from early developments to recent advancements," *Systems Science & Control Engineering: An Open Access Journal,* vol. 2, pp. 602-609, 2014.

[34] D. G. Thomas, "Ensemble learning," *The handbook of brain theory and neural networks,* vol. 2, pp. 110-125, 2002.

[35] R. Polikar, "Ensemble learning," *Ensemble machine learning: Methods and applications,* pp. 1-34, 2012.

[36] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," *Advances in neural information processing systems,* vol. 8, 1995.

[37] N. Hu and R. B. Dannenberg, "Bootstrap learning for accurate onset detection," *Machine Learning,* vol. 65, pp. 457-471, 2006.

[38] B. Boehmke and B. M. Greenwell, Hands-on machine learning with R, CRC press, 2019.

[39] K. M. Mendez, S. N. Reinke and D. I. Broadhurst, "A comparative evaluation of the generalised predictive ability of eight machine learning algorithms across ten clinical metabolomics data sets for binary classification," *Metabolomics,* vol. 15, pp. 1-15, 2019.

[40] T.-H. Lee, A. Ullah and R. Wang, "Bootstrap aggregating and random forest," *Macroeconomic forecasting in the era of big data: Theory and practice,* pp. 389-429, 2020.

[41] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE transactions on pattern analysis and machine intelligence,* vol. 20, pp. 832-844, 1998.

[42] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR),* vol. 9, pp. 381-386, 2020.

[43] E. Angelini, G. Di Tollo and A. Roli, "A neural network approach for credit risk evaluation," *The quarterly review of economics and finance,* vol. 48, pp. 733-755, 2008.

[44] N. D. Sim, "Viblo," 4 2019. [Online]. Available: https://viblo.asia/p/nn-mang-no-ron-nhan-tao-neural-networks-bWrZn6dwZxw.

[45] M. Diganta , "Mish: A Self Regularized Non-Monotonic Activation Function," *arXiv,* 2020.

[46] D. Liu, "medium," November 2017. [Online]. Available: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7.

[47] S. Shaw, "WanDB," 5 2020. [Online]. Available:

https://wandb.ai/shweta/Activation%20Functions/reports/Activation-Functions-Compared-with-Experiments--VmlldzoxMDQwOTQ.

[48] B. Jason, "A Gentle Introduction to the Rectified Linear Unit (ReLU)," Machine Learning Mastery, 2019. [Online]. Available:

https://en.wikipedia.org/wiki/Rectifier_(neural_networks).

[49] D. Liu, " A Practical Guide to ReLU," Medium, 30 November 2017. [Online].

[50] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *Ieee Potentials*, vol. 13, pp. 27-31, 1994.

[51] S. P. Fard and Z. Zainuddin, "The universal approximation capabilities of 2pi-periodic approximate identity neural networks," in *2013 International Conference on Information Science and Cloud Computing Companion*, IEEE, 2013, pp. 793-798.

[52] Z. Xu, B. Liu, B. Wang, C.-J. Sun, X. Wang, Z. Wang and C. Qi, "Neural response generation via gan with an approximate embedding layer," *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 617-626, 2017.

[53] X. Liu, D. Yang and A. El Gamal, "Deep neural network architectures for modulation classification," *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 915-919, 2017.

[54] "keras.io," [Online]. Available: https://keras.io/api/layers/core_layers/dense/.

[55] "EpyNN," [Online]. Available: https://epynn.net/Dense.html.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, pp. 1929-1958, 2014.

[57] "keras.io," [Online]. Available:

https://keras.io/api/layers/regularization_layers/dropout/.

[58] "viblo," [Online]. Available:

https://viblo.asia/p/dropout-trong-neural-network-E375zevdlGW.

[59] "github," 6 2022. [Online]. Available: https://github.com/dmlc/xgboost.

[60] G. Kalipe, V. Gautham and R. K. Behera, "Predicting malarial outbreak using machine learning and deep learning approach: a review and analysis," *2018 International Conference on Information Technology (ICIT),* pp. 33-38, 2018.

[61] M. Piryonesi and T. El-Diraby, "Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index," *Journal of Infrastructure Systems,* vol. 26, 2020.

[62] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani and J. Friedman, "Boosting and additive trees," *The elements of statistical learning: data mining, inference, and prediction,* pp. 337-387, 2009.

[63] P. Madeh and T. El-Diraby, "Using machine learning to examine impact of type of performance indicator on flexible pavement deterioration modeling," *Journal of Infrastructure Systems,* vol. 27, 2021.

[64] J. Fan, X. Wang, L. Wu, H. Zhou, F. Zhang, X. Yu, X. Lu and Y. Xiang, "Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in China," *Energy conversion and management,* vol. 164, pp. 102-111, 2018.

[65] "LightGBM," Microsoft Corporation, 2022. [Online]. Available:

https://lightgbm.readthedocs.io/en/v3.3.2/.

[66] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems,* vol. 30, 2017.

[67] A. Sharma, "Medium," 2018. [Online]. Available:

https://towardsdatascience.com/what-makes-lightgbm-lightning-fast-a27cf0d9785e.

[68] A. Kumar, "Linkedin," May 2022. [Online]. Available:

https://www.linkedin.com/pulse/xgboost-vs-lightgbm-ashik-kumar.

[69] J. Mahmood, G.-E. Mustafa and M. Ali, "Accurate estimation of tool wear levels during milling, drilling and turning operations by designing novel hyperparameter tuned models based on LightGBM and stacking," *Measurement,* vol. 190, 2022.

[70] "CatBoost," [Online]. Available: https://catboost.ai/.

[71] "Yandex," [Online]. Available: https://yandex.com/company/.

[72] "math.uwaterloo.ca," [Online]. Available:

https://wiki.math.uwaterloo.ca/statwiki/index.php?title=CatBoost:_unbiased_boosting_with_categorical_features#Ordered_Boosting.

[73] A. Dorogush, V. Ershov and A. Gulin, "CatBoost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363,* 2018.

[74] "nvidia," Dec 2018. [Online]. Available:

https://developer.nvidia.com/blog/catboost-fast-gradient-boosting-decision-trees/.

[75] B. Dhananjay and J. Sivaraman, "Analysis and classification of heart rate using CatBoost feature ranking model," *Biomedical Signal Processing and Control*, vol. 68, 2021.

[76] "leetcode," [Online]. Available:

https://assets.leetcode.com/uploads/2021/02/19/symtree1.jpg.

[77] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, "CatBoost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2008.

[78] J. Hancock and T. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *Journal of big data*, vol. 7, pp. 1-45, 2020.

[79] R. C. Turner, A. Fuggetta, L. Lavazza and A. L. Wolf, "A conceptual basis for feature engineering," *Journal of Systems and Software*, vol. 49, pp. 3-15, 1999.

[80] Z. Guoping, "A Necessary Condition for a Good Binning Algorithm in Credit Scoring," *Applied Mathematical Sciences*, vol. 8, pp. 3229-3242, 2014.

[81] W. Wang, C. Lesner, A. Ran, M. Rukonic, J. Xue and E. Shiu, "Using small business banking data for explainable credit risk scoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 13396-13401.

[82] H. A. Abdou, "Genetic programming for credit scoring: The case of Egyptian public sector banks," *Expert systems with applications*, vol. 36, pp. 11402-11417, 2009.

[83] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv*, 2016.

[84] J. Daniel and M. H. James, "Logistic Regression," in *Speech and Language Processing*, California, Stanford University Press, 2023.

[85] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu and J. Hu, "Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation," *Computational Materials Science*, vol. 171, 2020.

[86] G. James, D. Witten, T. Hastie and R. Tibshirani, in *An Introduction to Statistical Learning*, Springer, 2013, p. 181.

[87] M. Kuhn and K. Johnson, in *Applied Predictive Modeling*, 2013, p. 70.

[88] G. James, D. Witten, T. Hastie and R. Tibshirani, in *An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)*, Springer, 2013, p. 184.

[89] J. Cai, X. Chu, K. Xu, H. Li and J. Wei, "Machine learning-driven new material discovery," *Nanoscale Advances*, vol. 2, pp. 3115-3130, 2020.

[90] L. Rokach, "Ensemble-based classifiers," *Artificial intelligence review*, vol. 33, pp. 1-39, 2010.

[91] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of artificial intelligence research*, vol. 11, pp. 169-198, 1999.

[92] P. Sollich and A. Krogh, "Learning with ensembles: How overfitting can be useful," *Advances in neural information processing systems*, vol. 8, 1995.

[93] J. Xu, Y. Zhang and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Information sciences*, vol. 507, pp. 772-794, 2020.

[94] Y. Li, T. Bellotti and N. Adams, "Issues using logistic regression with class imbalance, with a case study from credit risk modelling," *Foundations of Data Science,* vol. 1, pp. 389-417, 2019.

[95] A. Luque, A. Carrasco, A. Mart and A. de Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition,* vol. 91, pp. 216-231, 2019.

[96] J. Brownlee, "Machine Learning Mastery," August 2020. [Online]. Available: https://machinelearningmastery.com/confusion-matrix-machine-learning/.

[97] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters,* vol. 27, pp. 861-874, 2006.

[98] M. Vuk and T. Curk, "ROC curve, lift chart and calibration plot," *Advances in methodology and Statistics,* vol. 3\, pp. 89-108, 2006.

[99] S. Leteurtre, A. Martinot, A. Duhamel, F. Proulx, B. Grandbastien, J. Cotting, R. Gottesman, A. Joffe, J. Pfenninger and P. Hubert, "Validation of the paediatric logistic organ dysfunction (PELOD) score: prospective, observational, multicentre study," *The Lancet,* vol. 362, pp. 192-197, 2003.

[100] "Kaggle," 2019. [Online]. Available: https://www.kaggle.com/c/home-credit-default-risk.

[101] J. Youden's, "infogalactic," [Online]. Available:

https://infogalactic.com/info/Youden%27s_J_statistic.

[102] K. M. Beyene and A. El Ghouch, "Time-dependent ROC curve estimation for interval-censored data," *Biometrical Journal,* vol. 64, pp. 1056-1074, 2022.

[103] M. J. Miller, "Credit Reporting Systems and the International Economy," The MIT Press, Cambridge, 2003.

[104] G. G. Grinstein and M. O. Ward, "Introduction to data visualization," *Information visualization in data mining and knowledge discovery,* vol. 1, pp. 21-45, 2002.

[105] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *International Journal of Emerging Technology and Advanced Engineering,* vol. 2, pp. 42-47, 2012.

[106] D. Marghescu, "Multidimensional data visualization techniques for financial performance data: A review," *Turku Centre for Computer Science,* 2007.

[107] C. Jiang, Z. Wang, R. Wang and Y. Ding, "Loan default prediction by combining soft information extracted from descriptive text in online peer-to-peer lending," *Annals of Operations Research,* vol. 266, pp. 511-529, 2018.

[108] M. Zhang, W. Li, Q. Du, L. Gao and B. Zhang, "Feature extraction for classification of hyperspectral and LiDAR data using patch-to-patch CNN," *IEEE transactions on cybernetics,* vol. 50, pp. 100-111, 2018.

[109] "Kaggle," [Online]. Available: https://www.kaggle.com/docs/notebooks.

[110] "Kaggle," [Online]. Available:

https://www.kaggle.com/datasets/wordsforthewise/lending-club.

[111] "UCI Machine Learning Repository," [Online]. Available:

https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients.

NHẬT KÝ LÀM VIỆC

Tuần	Từ ngày	Đến ngày	Nội dung
1	16/01/2023	22/01/2023	Đăng ký đề tài nghiên cứu. Làm các hồ sơ thủ tục liên quan. Họp bàn đề tài với giáo viên hướng dẫn.
2	23/01/2023	29/01/2023	Tìm hiểu về đề tài. Đưa ra các nội dung liên quan đến đề tài, hướng nghiên cứu.
3	30/01/2023	05/02/2023	Nghiên cứu các kiến thức liên quan đến hoạt động tín dụng cho vay. Học thêm các kỹ thuật liên quan đến tài chính ngân hàng nhằm hiểu thêm về đề tài.
4	06/02/2023	12/02/2023	Tiếp tục tìm hiểu đề tài thông qua các nghiên cứu có liên quan.
5	13/02/2023	19/02/2023	Tìm hiểu nghiên cứu sâu vào các thuật toán học máy và kỹ thuật liên quan để tiếp cận đến ứng dụng của thuật toán vào chấm điểm tín dụng.
6	20/02/2023	26/02/2023	Tìm kiếm dữ liệu phù hợp, đọc các tài liệu, các cuộc thảo luận liên quan nhằm tận dụng triệt để các thông tin dữ liệu mang lại.
7	27/02/2023	05/03/2023	Tiến hành các bước xử lý dữ liệu, phân tích và khám phá dữ liệu.
8	06/03/2023	12/03/2023	Xây dựng mô hình Logistic regression và Random forest trên bộ dữ liệu đã xử lý. Đánh giá các mô hình.

9	13/03/2023	19/03/2023	Kết quả huấn luyện mô hình Logistic regression và Random forest không tốt và cần tìm hướng xử lý mới. Trình bày vấn đề với giáo viên hướng dẫn, họp bàn tìm hướng giải quyết vấn đề
10	20/03/2023	26/03/2023	Sau khi trình bày vấn đề với giáo viên hướng dẫn được thông tin về việc cần có quá trình trích xuất đặc trưng từ dữ liệu. Tiến hành quá trình trích xuất đặc trưng từ các kiến thức liên quan đến cho vay tín dụng xây dựng được các thuộc tính mới sát hơn với dữ liệu tín dụng thực tế.
11	27/03/2023	02/04/2023	Xây dựng lại mô hình Logistic regression và Random forest. Kết quả đánh giá cho thấy sự khả quan hơn.
12	03/04/2023	09/04/2023	Tiếp tục xây dựng mô hình Neural network với kỳ vọng mô hình có thể học sâu hơn các thông tin từ dữ liệu.
14	10/04/2023	16/04/2023	Từ thông tin của GVHD, tìm hiểu và xây dựng mô XGBoost.
15	17/04/2023	23/04/2023	Các mô hình đã có kết quả nhưng chưa đạt được kỳ vọng đặt ra nên tiếp tục tìm hiểu thêm các thông tin liên quan có thể cải tiến chất lượng. Được biết thông tin về 2 mô hình LightGBM và CatBoost có thể cho kết quả tốt hơn. Tìm hiểu thông tin, cách thức hoạt động, xây dựng của 2 mô hình LightGBM và CatBoost.

16	24/04/2023	30/04/2023	Xây dựng 2 mô hình LightGBM và CatBoost. Kết quả cho thấy sự cái thiện chất lượng đáng kể. Tổng hợp và đánh giá chi tiết các mô hình đã đào tạo. Đưa ra mô hình tốt nhất. Tạo phương pháp chấm điểm từ kết quả đầu ra của mô hình.
17	01/05/2023	07/05/2023	Tìm hiểu phương pháp học đồng bộ. Thực nghiệm phương pháp dựa trên 3 mô hình tốt nhất là XGBoost, LightGBM và CatBoost. Đánh giá quá trình thực nghiệm phương pháp.
18	08/05/2023	14/05/2023	Đánh giá tổng quan quá trình nghiên cứu và thực nghiệm. Tổng hợp các thông tin có trong quá trình nghiên cứu và thực nghiệm. Hoàn thiện báo cáo và bài thuyết trình.