**NÂNG CAO HIỆU QUẢ PHÁT HIỆN RỦI RO TÍN DỤNG VỚI MÔ HÌNH MÁY HỌC**

**Học viện Chính sách và Phát triển**

**1. Giới thiệu bài toán**

Trong bối cảnh nền kinh tế đang phát triển ngày càng sôi động và nhanh chóng, thì các hoạt động cho vay tín dụng tại các ngân hàng cũng sẽ ngày càng mở rộng hơn tới mọi khách hàng cá nhân và tổ chức khác nhau. Đặc biệt là sau giai đoạn chịu ảnh hưởng từ đại dịch Covid19, những nhu cầu về nguồn vốn nhằm phục hồi lại kinh tế sẽ tăng cao. Theo số liệu thống kê từ Ngân hàng Nhà Nước Việt Nam (10/03/2022) cho thấy bắt đầu từ giai đoạn bùng phát dịch bệnh thì tỷ lệ nợ xấu tại các ngân hàng đã có xu hướng tăng trở lại, với mức tăng tỷ lệ nợ xấu nội bảng và nợ xấu gộp lần lượt 1,6% và 4,4% năm 2019 lên mức 1,9% và 7,3% năm 2021 [1].

Tại Việt Nam, tín dụng vẫn chiếm tỷ trọng cao nhất trong tổng tài sản, mang lại nguồn thu nhập lớn nhất song cũng là hoạt động mang lại rủi ro nhất cho ngân hàng. Do vậy, các ngân hàng cần có những biện pháp nhằm hạn chế tối đa những rủi ro tín dụng này để hoạt động của ngân hàng diễn ra tốt nhất. Với những lý do trên, nên hiện nay tại các ngân hàng và các tổ chức tín dụng khác đã quan tâm nhiều hơn và nhận thấy được tầm quan trọng của việc đánh giá ban đầu rủi ro và khả năng trả nợ của khách hàng , từ đó sẽ có cơ sở để quyết định có nên cho khách hàng vay hay không và cho vay bao nhiêu là an toàn.

Việc đánh giá khả năng trả nợ của khách hàng sẽ dựa trên các thông tin mà chúng ta thu thập được của khách hàng có thể dựa trên một số thông tin quan trọng như độ tuổi, thu nhập, lịch sử vay nợ, hôn nhân, tài sản,... Trong bài nghiên cứu này, chúng tôi sẽ đề xuất xây dựng một mô hình học máy để nâng cao khả năng phát hiện rủi ro tín dụng.

**2. Tổng quan nghiên cứu**

Trong quá khứ đã có nhiều bài nghiên cứu đưa ra các đề xuất phương pháp, mô hình nhằm giải quyết bài toán cho rủi ro khi cho vay tín dụng. Một số kết quả nghiên cứu từ các bài nghiên cứu trong quá khứ như sau:

Trong bài nghiên cứu "Phân tích rủi ro tín dụng bằng cách sử dụng công cụ phân loại học máy" ” (*Credit Risk Analysis using Machine Learning Classifiers)* của Trilok Nath Pandey và đồng tác giả (2017), nhóm nghiên cứu đã tiến hành một phân tích sâu sắc về rủi ro tín dụng bằng cách xây dựng và đánh giá các mô hình phân loại trên bộ dữ liệu liên quan. Bộ dữ liệu này cung cấp thông tin về các khía cạnh quan trọng của khách hàng và các yếu tố tiềm ẩn liên quan đến khả năng trả nợ của họ.

Kết quả của nghiên cứu đã cho thấy rằng mô hình Extreme Learning Machine (ELM) đã đạt được hiệu suất cao nhất trong việc dự đoán rủi ro tín dụng. Sự chính xác của mô hình này đã được đo bằng chỉ số accuracy, và những con số đáng chú ý đã được ghi nhận. Đối với hai bộ dữ liệu khác nhau (German và Australian), chỉ số accuracy tương ứng lần lượt là 0.9633 và 0.9632. Điều này làm nổi bật khả năng mạnh mẽ của mô hình ELM trong việc phân loại khách hàng và dự đoán khả năng trả nợ của họ, cung cấp một công cụ quan trọng cho các ngân hàng và tổ chức tài chính trong việc quản lý rủi ro và ra quyết định về tín dụng. [2].

Trong nghiên cứu "Phương pháp học máy để dự đoán mặc định khoản vay của người Trung Quốc trên thị trường P2P" (*Loan default prediction of Chinese P2P market: a machine learning methodology*) của Junhui Xu, Zekai Lu và Ying Xie (2021), nhóm tác giả đã tạo ra một cấu trúc mạnh mẽ để dự đoán khả năng mặc định trong thị trường P2P của Trung Quốc. Bằng cách sử dụng bốn mô hình phân loại phổ biến là Random Forest, XGBoost, Gradient Boost và mạng Neural, họ đã tiến hành một phân tích kỹ lưỡng để đánh giá hiệu suất của mỗi mô hình.

Kết quả của nghiên cứu đã phản ánh sự ấn tượng khi tất cả bốn mô hình đều đạt được độ chính xác và giá trị kappa trên 93%. Điều này chỉ ra rằng các mô hình đã có khả năng dự đoán mặc định với mức độ chính xác cao trên thị trường P2P của Trung Quốc. Đáng chú ý, mô hình Random Forest đã nổi bật với độ chính xác vượt trội hơn cả, khi đạt được mức trên 98%. Sự vượt trội này có thể được hiểu là một bước tiến quan trọng trong việc sử dụng học máy để dự đoán rủi ro tín dụng trong môi trường P2P, đặc biệt là trên thị trường Trung Quốc, nơi mô hình này có thể cung cấp thông tin quý giá cho các nhà đầu tư và các tổ chức tài chính để quản lý rủi ro và ra quyết định về cho vay. [3]

Trong nghiên cứu "Áp dụng mô hình học máy để dự đoán tính đủ điều kiện của khoản vay ngân hàng" (*Machine Learning Models for Predicting Bank Loan Eligibility*) của Ugochukwu .E. Orji và đồng nghiệp (2022), nhóm tác giả đã tiến hành xây dựng và đánh giá sáu mô hình phân loại khác nhau để dự đoán tính đủ điều kiện cho vay của khách hàng ngân hàng. Các mô hình này bao gồm Random Forest, Gradient Boost, Decision Tree, Support Vector Machine, K-Nearest Neighbor và Logistic Regression. Đáng chú ý, trong quá trình xử lý mất cân bằng dữ liệu, nhóm tác giả đã sử dụng thuật toán SMOTE để cải thiện hiệu suất của mô hình.

Kết quả của nghiên cứu đã cho thấy sự ấn tượng với hiệu suất mô hình cao. Mô hình Random Forest đã đạt được độ chính xác lên đến 95%, là mức cao nhất trong số các mô hình được thử nghiệm. Các mô hình còn lại cũng cho thấy hiệu suất tốt, với độ chính xác đều trên 80%. Điều này cho thấy tiềm năng của việc áp dụng học máy trong dự đoán tính đủ điều kiện cho vay của khách hàng ngân hàng và cung cấp một công cụ hữu ích cho các quyết định về cho vay trong lĩnh vực tài chính. [4].

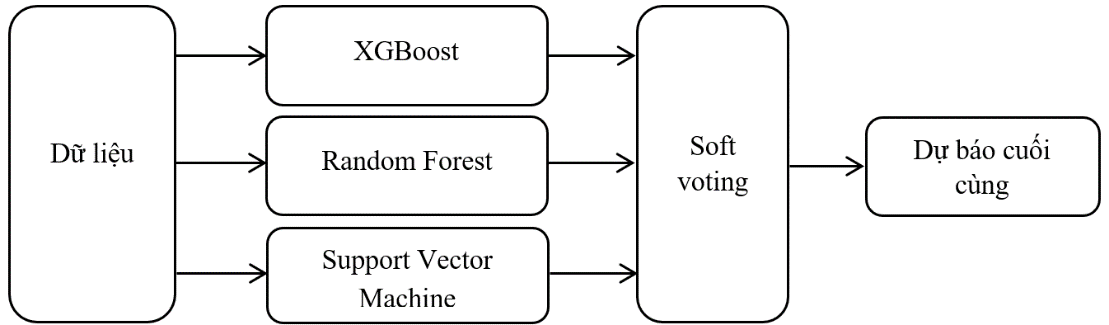
Trong bài nghiên cứu "Ứng dụng một số mô hình học máy để dự đoán khả năng trả nợ của mỗi người khi nộp đơn đi vay tín dụng" của Nguyễn Quốc Dương và Đặng Thế Vinh Hiển (2022), nhóm tác giả đã tiến hành xây dựng và đánh giá một loạt các mô hình phân loại kết hợp với thuật toán Synthetic Minority Over-sampling Technique (SMOTE). Cụ thể, các mô hình được xây dựng bao gồm Logistic Regression, Random Forest, XGBoost và LightGBM. Sử dụng kỹ thuật SMOTE giúp cân bằng dữ liệu và cải thiện hiệu suất dự đoán của các mô hình.

Kết quả của nghiên cứu đã cho thấy sự ấn tượng trong việc dự đoán khả năng trả nợ. Mỗi mô hình đều đạt được độ chính xác đáng kể, với Logistic Regression đạt 70.8%, Random Forest đạt 95.02%, XGBoost đạt 95.5% và LightGBM đạt 95.4%. Điều này chỉ ra rằng các mô hình học máy có khả năng dự đoán khả năng trả nợ của mỗi cá nhân khi nộp đơn vay tín dụng một cách chính xác và hiệu quả, cung cấp một công cụ hữu ích trong quá trình ra quyết định về việc cho vay tín dụng. [5].

**3. Mô hình đề xuất**

Trong bài nghiên cứu chúng tôi sẽ xây dựng mô hình dự đoán tín dụng với 3 mô hình học máy bao gồm Rừng ngẫu nhiên (Random forest), GBTC (Gradient Boosted Trees Classifier) và mô hình Multilayer Perceptron, tận dụng kết quả dự đoán có được từ 3 mô hình này chúng tôi tiếp tục sử dụng mô hình Votting là mô hình sử dụng đánh giá dựa trên số đông để đưa ra kết quả cuối cùng nhằm mục đính xem xét có khả năng tăng được độ chính xác mô hình so với các mô hình cơ bản hay không, nhận xét ưu điểm và nhược điểm của phương pháp này. Cấu trúc của mô hình đề suất như sau:

**Hình 1. Mô hình học máy đề xuất**

******

***3.1 Phương pháp phân loại***

*a. Random Forest (RDF)*

Thuật toán Random Forest hay được gọi là Rừng ngẫu nhiên là ensenmble của cây quyết định, ý tưởng chung của rừng ngẫu nhiên là một cây phân loại có phương sai cao nhưng trung bình của nhiều cây sẽ có phương sai thấp hơn. Điều này là do rừng ngẫu nhiên ước tính nhiều cây để sau đó tổng hợp tất cả các cây và tạo ra một cây phân loại tổng hợp cuối cùng.

Thuật toán Random Forest giải quyết được các vấn đề mà mô hình Cây quyết định gặp phải bao gồm:

* Dễ xảy ra quá khớp nếu số lượng các đặc trưng để hỏi lớn. Khi độ sâu của cây quyết định không bị giới hạn thì có thể tạo ra những node lá chỉ có một vài quan sát. Những kết luận dự báo từ chúng thường chỉ đúng trên tập huấn luyện mà không đúng trên tập kiểm tra.
* Trong tình huống bộ dữ liệu có số lượng biến lớn. Một cây quyết định có độ sâu giới hạn (để giảm thiểu quá khớp) thường bỏ sót những biến quan trọng.
* Cây quyết định chỉ tạo ra một kịch bản dự báo duy nhất cho mỗi một quan sát nên nếu model có hiệu suất kém thì kết quả sẽ bị chệch.

Trong thuật toán Decision Tree, khi xây dựng cây quyết định nếu để độ sâu tùy ý thì cây sẽ phân loại đúng hết các dữ liệu trong tập training dẫn đến mô hình có thể dự đoán tệ trên tập validation/test, khi đó mô hình bị overfitting, hay nói cách khác là mô hình có high variance.

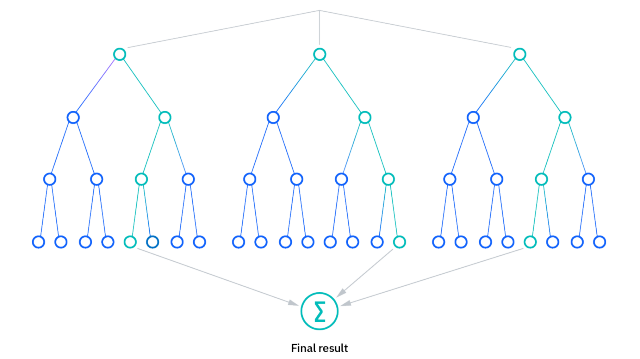
Thuật toán Random Forest gồm nhiều cây quyết định, mỗi cây quyết định đều có những yếu tố ngẫu nhiên:

* Lấy ngẫu nhiên dữ liệu để xây dựng cây quyết định.
* Lấy ngẫu nhiên các thuộc tính để xây dựng cây quyết định.

Do mỗi cây quyết định trong thuật toán Random Forest không dùng tất cả dữ liệu training, cũng như không dùng tất cả các thuộc tính của dữ liệu để xây dựng cây nên mỗi cây có thể sẽ dự đoán không tốt, khi đó mỗi mô hình cây quyết định không bị overfitting mà có thế bị underfitting, hay nói cách khác là mô hình có high bias. Tuy nhiên, kết quả cuối cùng của thuật toán Random Forest lại tổng hợp từ nhiều cây quyết định, thế nên thông tin từ các cây sẽ bổ sung thông tin cho nhau, dẫn đến mô hình có low bias và low variance, hay mô hình có kết quả dự đoán tốt.

Ý tưởng tổng hợp các cây quyết định của thuật toán Random Forest giống với ý tưởng của The Wisdom of Crowds [6] được đề xuất bởi by James Surowiecki vào năm 2004. The Wisdom of Crowds nói rằng thông thường tổng hợp thông tin từ 1 nhóm sẽ tốt hơn từ một cá nhân. Ở thuật toán Random Forest mình cũng tổng hợp thông tin từ 1 nhóm các cây quyết định và kết quả cho ra tốt hơn thuật toán Decision Tree với 1 cây quyết định.

**Ví dụ:** Mọi người muốn mua 1 sản phẩm trên tiki chẳng hạn, khi đọc review sản phẩm, nếu chỉ đọc 1 review thì có thể là ý kiến chủ quan của người đấy, hoặc sản phẩm người ấy mua không may bị lỗi gì; thông thường để có cái nhìn tốt về sản phẩm, mình hay đọc tất cả review rồi cho ra quyết định cuối cùng.



Rừng ngẫu nhiên phân biệt với cây phân loại theo hai cách chủ yếu. Một điểm khác biệt là dữ liệu cây được ước tính và điểm khác biệt thứ hai là cây được ước tính như thế nào. Trong thực tế, nhiều cây được ước tính bằng cách sử dụng các mẫu ngẫu nhiên từ tập dữ liệu huấn luyện, các mẫu được rút ra với sự thay thế và một mẫu duy nhất có thể được sử dụng nhiều lần.

*b. Gradient Boosted Trees (GBTC)*

Mô hình Gradient Boosted Trees (GBTC), là một phương pháp máy học phổ biến trong lĩnh vực học máy và khoa học dữ liệu. Được phát triển từ cây quyết định (Decision Trees), GBTC cải thiện hiệu suất của các mô hình cây quyết định bằng cách kết hợp chúng vào một Ensemble Model thông qua kỹ thuật tăng cường (boosting). GBTC thường được sử dụng cho các vấn đề phân loại (classification) và hồi quy (regression).

Cơ chế hoạt động của GBTC bắt đầu với việc xây dựng một cây quyết định đơn giản, còn gọi là cây cơ sở (base learner). Sau đó, nó liên tục cố gắng cải thiện mô hình bằng cách xây dựng thêm các cây quyết định khác, mỗi cây được huấn luyện để khắc phục những điểm yếu của mô hình hiện tại. Quá trình này được thực hiện bằng cách tạo ra các cây mới dựa trên những sai lệch (residuals) của mô hình trước đó, với mục tiêu là làm giảm sai số dự đoán.

Sau khi xây dựng cây cơ sở ban đầu, GBTC tiếp tục tạo ra các cây mới bằng cách tập trung vào những điểm yếu của mô hình hiện tại. Điều này được thực hiện bằng cách tính toán gradient của hàm mất mát (loss function) đối với mỗi điểm dữ liệu, và sau đó sử dụng các cây quyết định để dự đoán gradient này. Các cây quyết định mới được tạo ra nhằm mục đích làm giảm gradient, từ đó cải thiện mô hình tổng thể.

Quá trình này tiếp tục lặp lại cho đến khi một số lượng cây quyết định đã được xây dựng hoặc khi một tiêu chí dừng được đạt được. Sau đó, kết quả của các cây quyết định này được kết hợp lại để tạo thành một ensemble model mạnh mẽ có khả năng dự đoán tốt hơn so với mô hình ban đầu.

Có một số ưu điểm khi sử dụng GBTC:

* **Hiệu suất cao:** GBTC thường có hiệu suất dự đoán tốt hơn so với các phương pháp khác, đặc biệt là trên các tập dữ liệu có cấu trúc phức tạp.
* **Tính linh hoạt:** GBTC có thể xử lý cả dữ liệu số và dữ liệu phân loại, đồng thời không yêu cầu nhiều tiền xử lý.
* **Khả năng xử lý dữ liệu không đồng nhất:** GBTC có khả năng xử lý dữ liệu không đồng nhất (heterogeneous data), tức là các đặc trưng (features) có thể thuộc các loại khác nhau.

Tuy nhiên, cũng có một số nhược điểm của GBTC:

* **Chi phí tính toán:** GBTC có thể yêu cầu nhiều tài nguyên tính toán hơn so với các phương pháp khác, đặc biệt là khi số lượng cây quyết định lớn.
* **Dễ bị overfitting:** Trong một số trường hợp, GBTC có thể dễ dàng bị overfitting, đặc biệt là khi không được cấu hình đúng.
* **Đòi hỏi tuning thích hợp:** Để đạt được hiệu suất tốt nhất, GBTC yêu cầu việc tinh chỉnh (tuning) thích hợp của các siêu tham số, điều này có thể là một quá trình tốn thời gian và công sức.

*c. Multilayer Perceptron (MLP)*

Multilayer Perceptron (MLP) là một loại kiến trúc mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) được sử dụng rộng rãi trong lĩnh vực học máy và trí tuệ nhân tạo. MLP là một dạng của mạng nơ-ron feedforward, nghĩa là thông tin chỉ di chuyển theo một hướng, từ đầu vào đến đầu ra, mà không có chu kỳ phản hồi.

Cấu trúc của MLP bao gồm ít nhất ba lớp nơ-ron:

**Lớp đầu vào (input layer):** Là lớp chứa các nơ-ron đại diện cho các đặc trưng hoặc biến đầu vào của mô hình. Mỗi nơ-ron trong lớp này tương ứng với một đặc trưng đầu vào.

**Các lớp ẩn (hidden layers):** Các lớp ẩn chứa các nơ-ron được sắp xếp thành các tầng. Mỗi nơ-ron trong các lớp ẩn kết nối đến tất cả các nơ-ron trong các lớp trước đó và sau đó thông qua các trọng số. Số lượng và kích thước của các lớp ẩn có thể được điều chỉnh tùy thuộc vào độ phức tạp của bài toán và mức độ biểu diễn mà bạn muốn đạt được.

**Lớp đầu ra (output layer):** Là lớp cuối cùng của mạng nơ-ron, chứa các nơ-ron đại diện cho đầu ra dự đoán của mô hình. Số lượng nơ-ron trong lớp này thường phụ thuộc vào số lượng lớp đầu ra mong muốn (ví dụ: một nơ-ron cho mỗi lớp đầu ra trong bài toán phân loại nhị phân, và mỗi nơ-ron đại diện cho một lớp đầu ra trong bài toán phân loại đa lớp).

Quá trình huấn luyện MLP thường bao gồm hai giai đoạn chính: lan truyền tiến (Forward propagation) và lan truyền ngược (Backpropagation). Trong lan truyền tiến, đầu vào được đưa qua mạng và tính toán các giá trị đầu ra. Trong lan truyền ngược, độ lỗi giữa đầu ra dự đoán và đầu ra thực sự được tính toán và lan truyền ngược từ lớp cuối cùng về lớp đầu vào để điều chỉnh các trọng số sao cho giảm thiểu lỗi.

Ưu điểm của MLP bao gồm:

* Khả năng học được các hàm phức tạp và biểu diễn dữ liệu không tuyến tính.
* Linh hoạt trong việc xử lý các loại dữ liệu khác nhau và có thể được sử dụng cho nhiều loại bài toán khác nhau, từ phân loại cho đến dự đoán.
* Có thể được huấn luyện bằng các thuật toán cải tiến như stochastic gradient descent (SGD) và các biến thể của nó.

Tuy nhiên, một số nhược điểm của MLP bao gồm:

* Cần nhiều dữ liệu huấn luyện và thời gian huấn luyện lâu đối với các mô hình phức tạp.
* Dễ bị overfitting đặc biệt là khi có nhiều lớp ẩn và số lượng nơ-ron lớn.
* Cần chú ý tinh chỉnh các siêu tham số như số lượng lớp ẩn, số lượng nơ-ron trong mỗi lớp, và tỷ lệ học.

***3.2 Phương pháp Voting***

Phương pháp Soft voting là phương pháp ước tính xác suất lớp bằng các thuật toán khác nhau có cách tiếp cận tương phản để cải thiện độ chính xác của của dự đoán. Phương pháp này chỉ định một trọng số lớn hơn của bộ phân loại quan trọng là loại cao nhất được chọn bằng cách tổng hợp các xác suất mà các mô hình dự đoán. Soft voting được định nghĩa với công thức (2) như sau:

(2)

Trong đó hàm argmax là hàm cho ra giá trị lớn nhất và biểu thị trọng số liên quan đến dự đoán của bộ phân loại, là xác suất của bộ phân loại trong việc dự đoán một lớp nhất định. Trong quá trình ấn định trọng số của soft voting, các mô hình có độ tin cậy cao được đưa ra nhiều trọng số hơn dựa trên mức độ quan trọng và độ chính xác của bộ phân loại [7].

**IV. Thực nghiệm và Kết quả**

**­4.1. Miêu tả dữ liệu**

Bài nghiên cứu này sẽ thực nghiệm trên bộ dữ liệu *credit\_risk\_dataset.csv* được tải xuống từ trang Kaggle.com [9] vào ngày 01/3/2024. Bộ dữ liệu mang mục đích nghiên cứu, thực nghiệm, trong bộ dữ liệu có 25 trường dữ liệu và 100.000 quan sát.

Trường dữ liệu *“Credit\_Score”* thể hiện khách hàng có khả năng rủi ro, trong đó có 3 nhãn dữ liệu bao gồm **Poor**, **Standard**, **Good**. Tới số lượng lần lượt bằng:

* Poor với **28998** quan sát (**28,99%**)
* Standard với **53174** quan sát (**53,17%**)
* Good với **17828** quan sát (**17,82%)**

Hình 4.1. Phân phối của Biến mục tiêu

**A colorful bar chart with black background

Description automatically generated**

*Nguồn: Kết quả quá trình thực nghiệm*

|  |  |  |
| --- | --- | --- |
| Bảng 4.1 Mô tả về bộ dữ liệu thực nghiệm | | |
| Trường dữ liệu | **Mô tả** |
| ID | Đại diện cho một định danh duy nhất |
| Customer\_ID | Đại diện cho một định danh duy nhất của một người |
| Age | Đại diện cho tuổi |
| SSN | Đại diện cho số an sinh xã hội của một người |
| Occupation | Đại diện cho nghề nghiệp của người đó |
| Annual\_Income | Đại diện cho thu nhập hàng năm của người đó |
| Monthly\_Inhand\_Salary | Đại diện cho lương cơ bản hàng tháng của một người |
| Num\_Bank\_Accounts | Đại diện cho số tài khoản ngân hàng mà một người sở hữu |
| Num\_Credit\_Card | Đại diện cho số thẻ tín dụng khác mà một người sở hữu |
| Interest\_Rate | Đại diện cho lãi suất trên thẻ tín dụng |
| Num\_of\_Loan | Đại diện cho số khoản vay mà một người đã vay từ ngân hàng |
| Type\_of\_Loan | Đại diện cho loại hình vay mà một người đã vay |
| Delay\_from\_due\_date | Đại diện cho số ngày trung bình bị trễ so với ngày thanh toán |
| Num\_of\_Delayed\_Payment | Đại diện cho số lần trung bình bị trễ thanh toán của một người |
| Changed\_Credit\_Limit | Đại diện cho tỷ lệ thay đổi trong hạn mức thẻ tín dụng |
| Num\_Credit\_Inquiries | Đại diện cho số lần kiểm tra thông tin tín dụng |
| Credit\_Mix | Đại diện cho phân loại về sự kết hợp của các khoản tín dụng |
| Outstanding\_Debt | Đại diện cho số nợ còn lại cần thanh toán |
| Credit\_Utilization\_Ratio | Đại diện cho tỷ lệ sử dụng của thẻ tín dụng |
| Credit\_History\_Age | Đại diện cho tuổi của lịch sử tín dụng của người đó |
| Payment\_of\_Min\_Amount | Đại diện cho việc thanh toán chỉ mức tối thiểu bởi người đó |
| Total\_EMI\_per\_month | Đại diện cho số tiền EMI hàng tháng |
| Amount\_invested\_monthly | Đại diện cho số tiền hàng tháng được đầu tư bởi khách hàng |
| Payment\_Behaviour | Đại diện cho hành vi thanh toán của khách hàng |
| Monthly\_Balance | Đại diện cho số dư hàng tháng của khách hàng |
| Credit\_Score | Đại diện cho phân hạng điểm tín dụng (Poor, Standard, Good) (Mục tiêu) |

*Nguồn: Từ nguồn dữ liệu*

**­4.2. Tiền xử lý dữ liệu**

1. Xử lý thiếu dữ liệu

Trong bộ dữ liệu của chúng tôi, có một phần không nhỏ (35.18%) các dòng chứa dữ liệu thiếu, phân bố tương đối đều trong các trường sau (Hình 4.2):

* Monthly\_Inhand\_Salary: 15.0% (15,002 dòng thiếu)
* Type\_of\_Loan: 11.41% (11,408 dòng thiếu)
* Num\_of\_Delayed\_Payment: 7.0% (7,002 dòng thiếu)
* Amount\_invested\_monthly: 4.48% (4,479 dòng thiếu)
* Num\_Credit\_Inquiries: 1.97% (1,965 dòng thiếu)
* Monthly\_Balance: 1.2% (1,200 dòng thiếu)

**Hình 4.2.1. Số lượng cột bị thiếu dữ liệu**

A black and white bar code

Description automatically generated

*Nguồn: Kết quả quá trình thực nghiệm*

Để xử lý dữ liệu thiếu, chúng tôi sử dụng phương pháp điền giá trị trung bình (mean) cho các cột dạng số (numeric), và sử dụng giá trị 'NAH' (Not Available Here) để điền cho các cột dạng phân loại (categorical) nhằm đại diện cho việc dữ liệu bị thiếu. Sự lựa chọn 'NAH' ở đây không chỉ là để điền vào khoảng trống mà còn thể hiện rằng trong quá trình thu thập dữ liệu, có khả năng các khách hàng đã chọn cố ý để trống thông tin này.

**Ví dụ,** một khách hàng có công việc không ổn định (như làm tự do), có thể chọn để không cung cấp thông tin về nghề nghiệp để hạn chế ngân hàng biết được chi tiết về tình hình tài chính của họ. Từ điều này, chúng tôi nhận thấy rằng dữ liệu vẫn mang thông tin ẩn.

**Hình 4.2.2. Dữ liệu sau khi xử lý**

A black and white striped background

Description automatically generated

*Nguồn: Kết quả quá trình thực nghiệm*

Ngoài ra chúng tôi còn tiến hành xử lý dữ liệu ở một số trường dữ liệu ghi sai ví dụ, trường dữ liệu “Occupation” có các dòng ghi là “\_\_\_\_\_\_\_” các dòng dữ liệu này sẽ được ghi thành “None”, việc ghi “None” cũng tương tự ở trên. Chúng tôi cũng làm tương tự với các trường dữ liệu ghi sai.

2. Xử lý mất cân bằng

Trong biến mục tiêu “Credit\_Score” thể hiện khách hàng có khả năng rủi ro, trong đó có 3 nhãn dữ liệu bao gồm **Poor, Standard, Good**. Bài nghiên cứu tiến hành gộp 2 nhãn dữ liệu Standard, Good thành 1 nhãn duy nhất là Good để có thể thuận thiện xác định nhãn Poor. Trong nhãn dữ liệu Good chiếm 71.002 quan sát (tỷ lệ là 71% toàn dữ liệu) và nhãn Poor chiếm 28.998 quan sát (tỷ lệ là 28,99% toàn dữ liệu) (Hình 4.2)

Hình 4.2. Phân phối của Biến mục tiêu

**A blue and orange rectangles

Description automatically generated**

*Nguồn: Kết quả quá trình thực nghiệm*

Phát hiện ra sự mất cân bằng trong dữ liệu là một vấn đề quan trọng, vì nó có thể dẫn đến việc mô hình học không chính xác hoặc không công bằng đối với các nhãn thiểu số. Để đối phó với tình trạng này, chúng tôi quyết định thực hiện phương pháp under sampling. Bằng cách loại bỏ một phần của các mẫu được gán nhãn 'Good', chúng tôi nhằm mục đích tạo ra một tập dữ liệu cân bằng hơn, trong đó các nhãn 'Good' và 'Poor' có trọng số tương đương. Quyết định này không chỉ giúp cải thiện hiệu suất của mô hình, mà còn đảm bảo rằng các dự đoán được thực hiện dựa trên một tập dữ liệu biểu diễn chính xác hơn về phân phối của các lớp trong thực tế. Điều này sẽ đảm bảo tính công bằng và chính xác của mô hình, đồng thời nâng cao khả năng áp dụng của nó trong các tình huống thực tế.

3. Encoder

Sau bước trên, chúng tôi tiến hành encoder các trường dữ liệu sau: 'Month', 'Type\_of\_Loan', 'Credit\_Mix', 'Credit\_History\_Age', 'Payment\_of\_Min\_Amount', và 'Payment\_Behaviour'. Việc encoder này nhằm chuyển đổi các biến phân loại thành các biến dạng số, giúp cho các thuật toán máy học có thể hiểu được và xử lý dữ liệu dễ dàng hơn.

Việc encoder đảm bảo rằng các biến phân loại được biểu diễn dưới dạng số, mà không làm mất đi ý nghĩa của chúng. Điều này giúp cho các mô hình máy học có thể sử dụng được dữ liệu, đồng thời không gây ra sự hiểu lầm trong việc đánh giá dữ liệu đầu vào. Chẳng hạn, encoder sẽ chuyển đổi các giá trị trong trường 'Month' từ các chuỗi tên tháng thành các giá trị số tương ứng (ví dụ: 'Jan' -> 1, 'Feb' -> 2), giữ nguyên thông tin về thứ tự và khoảng cách giữa các tháng. Điều này giúp cho thuật toán máy học hiểu được rằng có sự liên kết giữa các giá trị tháng và có thể sử dụng thông tin này để tạo ra dự đoán chính xác.

4. Vector hóa

Tiếp theo, chúng tôi tiến hành Vector hóa các biến với **VectorAssembler** đã được tiền xử lý dữ liệu trước đó. **VectorAssembler** là một công cụ quan trọng trong quá trình chuẩn bị dữ liệu cho mô hình học máy trong Apache Spark. Nó giúp kết hợp các cột dữ liệu thành một vectơ đặc trưng lớn, là đầu vào cho các thuật toán học máy trong Spark.

Việc sử dụng VectorAssembler giúp chúng tôi tổ chức và chuẩn bị dữ liệu một cách hiệu quả, đồng thời đảm bảo rằng mô hình học máy có thể sử dụng dữ liệu một cách dễ dàng và hiệu quả. Bằng cách này, chúng tôi có thể kết hợp tất cả các biến đặc trưng đã được xử lý trước đó thành một vectơ đặc trưng duy nhất, mà mô hình học máy có thể sử dụng để học và tạo dự đoán.

Quá trình này không chỉ giúp giảm bớt công đoạn tiền xử lý dữ liệu mà còn tạo ra một quy trình tự động hóa hóa dữ liệu mà có thể tái sử dụng cho các dự án và mô hình học máy khác. Điều này giúp tăng tốc quá trình phát triển mô hình và giảm thiểu các lỗi có thể xảy ra do xử lý dữ liệu thủ công.

5. Chuẩn hóa dữ liệu

Cuối cùng, chúng tôi thực hiện việc chuẩn hóa dữ liệu bằng phương pháp z-score (z-score normalization). Phương pháp này là một trong những kỹ thuật chuẩn hóa dữ liệu phổ biến nhất trong học máy. Phương pháp z-score normalization được thực hiện bằng cách áp dụng công thức sau cho mỗi giá trị của biến đặc trưng

* z là giá trị chuẩn hóa của biến đặc trưng.
* x là giá trị ban đầu của biến đặc trưng.
* μ là giá trị trung bình của biến đặc trưng trong tập dữ liệu.
* σ là độ lệch chuẩn của biến đặc trưng trong tập dữ liệu.

Bằng cách này, mỗi biến đặc trưng sẽ được chuyển đổi sao cho giá trị trung bình của biến là 0 và độ lệch chuẩn là 1. Quá trình này giúp đồng bộ hóa và định lượng biến đặc trưng theo cùng một tỷ lệ, giúp cho mô hình học máy có thể học được hiệu quả hơn và dự đoán chính xác hơn trên các biến đa dạng.

Việc chuẩn hóa dữ liệu bằng phương pháp z-score cũng giúp tăng tốc quá trình học của mô hình và giảm thiểu các vấn đề như độ lệch và ảnh hưởng của các giá trị ngoại lai. Đồng thời, nó cũng tạo điều kiện thuận lợi cho việc so sánh giữa các biến đặc trưng khác nhau một cách công bằng và chính xác.

6. Chia dữ liệu Huấn luyện và kiểm định

Chúng tôi tiến hành chia tập dữ liệu thành hai phần, một phần dành cho huấn luyện và một phần dành cho kiểm định mô hình, với tỷ lệ là:

* 80% cho tập huấn luyện (43868 quan sát).
* 20% cho tập kiểm định mô hình (10786 quan sát).

Tỷ lệ này được chọn để đảm bảo rằng mô hình có đủ dữ liệu để học và hiểu các mẫu dữ liệu, đồng thời cũng đảm bảo rằng chúng ta có một tập dữ liệu đủ lớn để kiểm định hiệu suất của mô hình một cách đáng tin cậy. Điều này giúp cân bằng giữa việc huấn luyện mô hình một cách hiệu quả và đánh giá hiệu suất của mô hình trên dữ liệu mới.

**4.3. Phương pháp đánh giá**

Đối với phương pháp đánh giá trong bài toán phân loại, trong bài nghiên cứu sẽ sử dụng các phương pháp đánh giá hiệu suất của mô hình dựa trên ma trận nhầm lẫn (*Confusion Matrix)* bằng các chỉ đó đánh giá như sau accuracy, precision, recall và F1 scose. Ma trận nhầm lẫn có dạng như sau:

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | TP | FN |
| Negative | FP | TN |

*Nguồn: Nhóm tác giả*

* TP (*True Positive*) là tổng số trường hợp dự báo khớp mẫu đúng,
* TN (*True Negative*) là tổng số trường hợp dự báo khớp với mẫu sai.
* FP (*False Positive*) là tổng số trường hợp dự báo các quan sát thuộc mẫu đúng tính thành sai.
* FN (*False Negative*) là tổng số trường hợp dự báo các quan sát thuộc mẫu sai tính thành đúng.

Accuracy là độ chính xác của mô hình, nó là tỷ lệ giữa số lượng mẫu được phân loại chính xác trên cho tổng số mẫu của tập dữ liệu thử nghiệm. Accuracy được xác định với công thức (3):

(3)

Precision là tỷ lệ giữa số lượng mẫu được dự đoán là tích cực trên tổng mẫu là tíchcực của tập dữ liệu thử nghiệm. Precision được xác định với công thức (4):

(4)

Recall là tỷ lệ giữa số lượng mẫu được dự đoán là tiêu cực trên tổng mẫu là tiêu cực của tập dữ liệu thử nghiệm và được xác định với công thức (5):

(5)

F1 scose được hiểu là trung bình trọng số của Precision và Recall, nó thể hiện sự cân bằng giữa chúng, trong đó F1 scose đạt giá trị tốt nhất là 1 và kém nhất là 0, chỉ số F1 - scose được xác định với công thức (6):

(6)

***4.4. Các tham số và môi trường cài đặt***

Dưới đây là giới thiệu về 4 mô hình học máy cùng các tham số quan trọng:

***Random Forest (RDF) với Entropy:***

* numTrees = 101 (Số cây)
* maxDepth = 25 (số tầng tối đa)
* seed = 42 (Mã hạt giống cho quá trình ngẫu nhiên)
* impurity = 'entropy' (Phương pháp đo lường độ)

***Random Forest (RDF) với Gini:***

* numTrees = 101 (Số cây)
* maxDepth = 25 (số tầng tối đa)
* seed = 42 (Mã hạt giống cho quá trình ngẫu nhiên)
* impurity = 'gini' (Phương pháp đo lường độ)

***Gradient Boosted Trees (GBT):***

* maxDepth = 5 (số tầng tối đa của cây)
* maxBins = 32 (số lượng bins tối đa cho các tính năng)
* minInstancesPerNode = 1 (số lượng mẫu tối thiểu cần phải có trong mỗi nút của cây)
* lossType = 'logistic' (Loại hàm mất mát được sử dụng trong quá trình tối ưu hóa)

***Multilayer Perceptron (MLP):***

* solver = 'l-bfgs' (Thuật toán tối ưu hóa)
* maxIter = 100 (Số lượng vòng lặp tối đa)
* stepSize = 0.03 (Kích thước bước cho quá trình tối ưu hóa)
* layers = [22, 128, 64, 32, 16, 8, 2] (Cấu trúc của các lớp trong mạng nơ-ron)

***4.5. Kết quả thực nghiệm***

Mô hình đề xuất và các mô hình đơn lẻ khi huấn luyện với tập dữ liệu huấn luyện và được thử nghiệm với tập dữ liệu thực nghiệm đã thu được các chỉ số đánh giá mô hình phân loại được trình bày trong (Bảng 3) như sau:

**Bảng : Kết quả đánh giá mô hình đề xuất**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô hình** | **Accuracy** | **Recall** | **Precision** | **F1-score** |
| **RDF (1)** | 0.7904 | 0.7544 | 0.8096 | 0.7810 |
| **RDF (2)** | 0.7878 | 0.7453 | 0.8112 | 0.7768 |
| **GBT** | **0.8090** | **0.8055** | 0.8082 | **0.8069** |
| **MLP** | 0.7760 | 0.7612 | 0.7811 | 0.7710 |
| **Voting** | 0.7936 | 0.7642 | **0.8087** | 0.7858 |

*Nguồn: Kết quả từ quá trình thực nghiệm*

Dựa trên bảng đánh giá hiệu suất của các mô hình học máy, chúng ta có thể đưa ra nhận xét sau:

**Random Forest (RDF):**

* RDF (1) với **Entropy** đạt được accuracy đáng kể là 0.7904, cùng với recall và precision là 0.7544 và 0.8096 tương ứng. Tuy nhiên, F1-score của RDF (1) là 0.7810, cho thấy sự cân bằng giữa recall và precision.
* RDF (2) với **Gini** cũng đạt được kết quả tương tự, với accuracy là 0.7878 và F1-score là 0.7768. Cả hai mô hình đều cho thấy sự ổn định và khả năng dự đoán tốt trên dữ liệu.
* Từ 2 mô hình trên chúng tôi quyết định lựa chọn mô hình RDF (1) để đưa vào mô hình Voting bởi mô hình này cho thấy F1-score tốt hơn.

**Gradient Boosted Trees (GBT):**

* GBT đạt được accuracy cao nhất trong các mô hình, đạt 0.8090. Điều này cho thấy GBT có khả năng dự đoán mạnh mẽ trên tập dữ liệu, và F1-score ở mức cao, là 0.8069.

**Multilayer Perceptron (MLP):**

* MLP đạt được accuracy là 0.7760, thấp hơn so với các mô hình khác. Tuy nhiên, nó vẫn có một hiệu suất tương đối tốt với F1-score là 0.7710.

**Voting (RDF (1), GBT, MLP):**

* Mô hình Voting kết hợp kết quả từ RDF (1), GBT và MLP, cho thấy sự cải thiện về accuracy so với các mô hình cơ bản. Accuracy đạt 0.7936, cao hơn so với mô hình RDF (1) và MLP, nhưng thấp hơn so với GBT. F1-score của Voting cũng cao (đạt 0.7858), về Precision có thể thấy mô hình Voting đã đạt được hiệu suất cao hơn.

Từ kết quả đánh giá trên có thể thấy được rằng mô hình nên được chọn là mô hình **Gradient Boosted Trees** bởi tất cả các chỉ số đánh giá của mô hình rất tốt với bộ dữ liệu sau và nên được các ngân hàng ứng dụng trong thực tế để giải quyết bài toán.

Đối với mô hình đề xuất của chúng tôi tuy vẫn chưa đạt được chỉ số đánh giá tôi, điều này chúng tôi đưa ra các giải thích sau:

***Khả năng hiệu chỉnh tham số chưa tối ưu:***

* Mô hình đề xuất có thể cần được điều chỉnh tham số một cách tối ưu hơn để đạt được hiệu suất cao nhất. Việc hiệu chỉnh tham số không chỉ bao gồm việc chọn lựa các giá trị tối ưu cho các tham số mô hình, mà còn bao gồm việc tối ưu hóa các kỹ thuật huấn luyện và kiểm định mô hình.

***Sự phức tạp của bài toán:***

* Một số bài toán có tính phức tạp cao, đặc biệt là trong lĩnh vực tài chính như dự đoán tín dụng. Các mô hình có thể gặp khó khăn trong việc học và tổng hợp các mẫu phức tạp từ dữ liệu, dẫn đến hiệu suất không tối ưu.

***Thiếu thông tin hoặc nhiễu trong dữ liệu:***

* Dữ liệu thực tế thường không hoàn hảo và có thể chứa các giá trị thiếu hoặc nhiễu. Các giá trị này có thể ảnh hưởng đến quá trình huấn luyện và dự đoán của mô hình, gây ra hiện tượng giảm hiệu suất.

***Sự lựa chọn mô hình:***

* Có thể mô hình đề xuất không phù hợp với đặc điểm của dữ liệu hoặc không tốt như các mô hình cơ bản khác hoặc có thể khi kết hợp với nhau không tốt.

**5. Kết luận**

Nghiên cứu này cung cấp cho các ngân hàng thương mại một công cụ giúp có thể dự đoán được khả năng trả nợ củ khách, với mô hình đề xuất khi sử dụng các mô hình phân loại bao gồm Random Forest, GBT và MLP, sau đó tận dụng kết quả dự báo của các mô hình chúng tôi đã kết hợp kết quả dự đoán đó sử dụng với phương pháp Voting nhằm mục đích cải thiện được khả năng dự đoán của mô hình. Với việc tiến hành thực nghiệm với bộ dữ liệu “Credit Risk Dataset”, các kết quả thực nghiệm qua các mô hình cho thấy mô hình GBT đạt cao nhất về các chỉ số Accuracy : 0.8090, Recall: 0.8055 và F1: 0.8069 và mô hình Voting đạt cao nhất ở Precision với 0.8087. Từ các kết quả trên cho thấy có thể ứng dụng mô hình GBT nếu cần hiệu suất dự đoán bằng chỉ số Recall và có thể ứng dụng mô hình Voting nếu cần hiệu suất ở Precision.

Trong thời gian tới, chúng tôi sẽ tiếp tục cải thiện mô hình khi thay đổi các mô hình phân loại khác vào K láng giềng gần nhất, hồi quy logistic, Light GBM… và cải thiện hơn trong quá trình chuẩn bị dữ liệu để mô hình có thể cải thiện được khả năng dự đoán tốt hơn, tiếp sau đó bài cũng sẽ ứng dụng mô hình với các tập dữ liệu tương tự và dữ liệu thực tế để có được cái nhìn tổng quan hơn về mô hình đề xuất.

**TÀI LIỆU THAM KHẢO**

[1]. “*Nợ xấu 'phình to': 'Cây đũa thần 42' chưa thể hoàn tất sứ mệnh lịch sử”* (2022), truy cập ngày 1 tháng 10 năm 2022, từ https://www.vietnamplus.vn/.

[2]. Pandey, Trilok Nath et al. “Credit risk analysis using machine learning classifiers.” 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (2017): 1850-1854.

[3]. Xu, Junhui et al. “Loan default prediction of Chinese P2P market: a machine learning methodology.” Scientific Reports 11 (2021): n. pag.

[4]. Orji, Ugochukwu. E. et al. “Machine Learning Models for Predicting Bank Loan Eligibility.” 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON) (2022): 1-5.

[5]. Nguyễn Quốc Dương và Đặng Thế Vinh Hiển. “Ứng dụng một số mô hình học máy để dự đoán khả năng trả nợ của mỗi người khi nộp đơn đi vay tín dụng.” Hội Thảo Khoa Học Quốc Gia – Phân tích định lượng và các vấn đề kinh tế và xã hội trong môi trường số, 04/2022, pp. 269-278.

[6]. The Wisdom ofCrowds https://en.wikipedia.org/wiki/The\_Wisdom\_of\_Crowds

[7]. Zhou, Z. Ensemble Methods: Foundation Sand Algorithms; CRC Press: Boca Raton, FL, USA, 2012; ISBN 978-1-439-830031.

[9]. Credit Risk Dataset: <https://www.kaggle.com/datasets/parisrohan/credit-score-classification>.

[11] Nguyen Duc Hieu: P2P Lending (Cho Vay Ngang Hàng) Là Gì? Lợi Ích Và Rủi Ro, 05/09/2022