Technical Report: Home Credit - Credit Risk Model Stability
Group 1

Đoàn Nhật Sang, Trương Văn Khải, Lê Ngô Minh Đức, Hoàng Tiến Đạt
University of Information Technology,
Vietnam National University HCMC
Email: {21522542, 21520274, 21520195, 21520696}@*gm.uit.edu.vn*

***Tóm tắt nội dung*—Mục tiêu của cuộc thi Credit Risk Model Stability là dự đoán khách hàng nào có khả năng vỡ nợ cao hơn. Việc đánh giá sẽ ưu tiên những giải pháp ổn định theo thời gian. Trong cuộc thi này, nhóm đạt điểm số 0.599 trên public test và 0.527 trên private test bằng cách sử dụng kết hợp các mô hình LightGBM và CatBoost mà không sử dụng bất kỳ thủ thuật nào với chỉ số đánh giá. Khi nhóm sử dụng các thủ thuật với chỉ số đánh giá, đã đạt điểm số trên bảng xếp hạng public test là 0.662. Kết thúc cuộc thi, nhóm đạt vị trí thứ 66/3883 và giành được huy chương bạc.**

***Index Terms*—Home Credit, Credit Risk Model Stability, Loan Prediction, Stable Model**

1. Introduction

Cuộc thi Credit Risk Model Stability [4] do công ty Home Credit, được thành lập vào năm 1997, tổ chức. Động lực đằng sau dự án này là mở rộng khả năng tiếp cận các dịch vụ tài chính cho những cá nhân có lịch sử tín dụng hạn chế hoặc chưa có. Các phương pháp máy học trước đây thường loại trừ những cá nhân này khỏi các hồ sơ cho vay. Bằng cách tận dụng học máy với cách tiếp cận mới, chúng ta có thể phân tích các đặc điểm đa dạng của khách hàng để đưa ra các quyết định cho vay thông minh, hợp lý hơn, thúc đẩy sự hòa nhập và trao quyền tài chính cho người dùng.

1.1. Data Description

Bộ dữ liệu cho kỳ thi [4] bao gồm nhiều bảng dữ liệu từ các nguồn và mức độ chi tiết khác nhau:

- **base**: Bảng này chứa các thông tin cơ bản của các mẫu dữ liệu.
- **depth=0**: Đặc trưng tĩnh liên quan trực tiếp đến case_id. Gồm 2 bảng.
- **depth=1**: Bản ghi lịch sử liên kết với case_id, lập chỉ mục bằng num_group1. Gồm 10 bảng.
- **depth=2**: Bản ghi lịch sử liên kết với case_id, lập chỉ mục bằng num_group1 và num_group2. Gồm 4 bảng.

Các bảng với depth > 0 yêu cầu áp dụng các hàm tổng hợp để chuyển các bản ghi lịch sử thành đặc trưng duy nhất.

Các cột đặc biệt trong bộ dữ liệu bao gồm:

- **case_id**: Khóa duy nhất cho mỗi trường hợp tín dụng.
- **date_decision**: Ngày ra quyết định cho vay.
- **WEEK_NUM**: Số tuần để tổng hợp dữ liệu.
- **MONTH**: Tháng, dùng để tổng hợp.
- **target**: Nhãn mục tiêu, xác định khách hàng có vỡ nợ hay không.
- **num_group1**: Chỉ mục cho các bản ghi lịch sử trong bảng depth=1 và depth=2.
- **num_group2**: Chỉ mục thứ hai cho các bản ghi lịch sử trong bảng depth=2.

Các biến dự đoán có các ký hiệu biểu thị nhóm biến đổi: **P** - Biến đổi DPD (Số ngày quá hạn); **M** - Mã hóa các danh mục; **A** - Biến đổi số tiền; **D** - Biến đổi ngày; **T** - Biến đổi không xác định; **L** - Biến đổi không xác định. Các biến đổi trong cùng nhóm được đánh dấu bằng chữ cái viết hoa ở cuối tên biến.

1.2. Metrics

Các bài nộp được đánh giá bằng cách sử dụng chỉ số ổn định Gini Stability do ban tổ chức thiết kế. Để hiểu được thang do đánh giá này, trước hết ta cần hiểu về AUC.

1.2.1. AUC

ROC (Receiver Operating Characteristic) là đường cong biểu diễn khả năng phân loại của một mô hình phân loại tại các ngưỡng threshold. Đường cong này dựa trên hai chỉ số:

- **TPR (True Positive Rate)**: Hay còn gọi là recall hoặc sensitivity. Là tỷ lệ các trường hợp phân loại đúng positive trên tổng số các trường hợp thực tế là positive. Chỉ số này sẽ đánh giá mức độ dự báo chính xác của mô hình trên positive.

$$\text{TPR} = \frac{TP}{TP + FN} \tag{1}$$

- **FPR (False Positive Rate)**: Tỷ lệ dự báo sai các trường hợp thực tế là negative thành positive trên tổng số các trường hợp thực tế là negative.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{2}$$

AUC là chỉ số được tính toán dựa trên đường cong ROC nhằm đánh giá khả năng phân loại của mô hình. Phần diện tích nằm dưới đường cong ROC là AUC (area under curve) có giá trị nằm trong khoảng [0, 1]. Khi diện tích này càng lớn thì khả năng phân loại của mô hình càng tốt.

1.2.2. Gini Stability

Để tính được Gini Stability, chúng ta cần phải tính điểm số Gini cho các dự đoán trong từng WEEK_NUM:

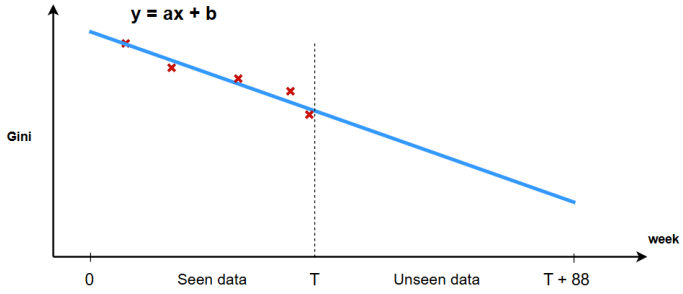
$$gini = 2 * AUC - 1 \tag{3}$$

Sau đó, một hồi quy tuyến tính, $a \cdot x + b$, được khớp với các điểm Gini hàng tuần, và một 'tốc độ giảm' được tính là $min(0, a)$. Điều này được sử dụng để phạt các mô hình bị giảm khả năng dự đoán. Cuối cùng, độ biến thiên của các dự đoán, được tính bằng cách lấy độ lệch chuẩn của các phần dư từ hồi quy tuyến tính trên, áp dụng một hình phạt cho độ biến thiên của mô hình. Chỉ số ổn định cuối cùng được tính như sau:

$$\begin{aligned} \text{stability metric} = \text{mean(gini)} + 88.0 \cdot \min(0, a) \\ - 0.5 \cdot \text{std(residuals)} \end{aligned} \tag{4}$$

Ý nghĩa của từ "ổn định" mà tác giả đề xuất tức là họ muốn tìm được một phương pháp có hiệu suất ổn định theo thời gian và mức gini trung bình trong nhiều tuần càng cao càng tốt. Đó là lý do thang đo này bị ảnh hưởng bởi ba yếu tố, thay vì chỉ một yếu tố duy nhất là AUC. Thực chất, thang đo đánh giá này là một $mean(gini)$ bị phạt bởi hai yếu tố sau:

1) Tỷ lệ giảm điểm gini theo hàng tuần - Falling rate of weekly Gini score.
2) Biến thiên của dự đoán.



Hình 1: Gini Stability.

Yếu tố phạt đầu tiên là độ dốc của đường màu xanh trên hình vẽ trên. $min(0, a)$ có nghĩa là chúng ta không quan tâm đến trường hợp độ dốc dương (tức là đang đi lên). Tuy nhiên hệ số của falling rate là 88, một số khá cao, thể hiện cho sự trừng phạt rất nặng.

Yếu tố phạt thứ hai liên quan đến độ biến thiên giữa đường màu xanh và các điểm dữ liệu hàng tuần (các điểm x màu đỏ), được thể hiện bởi $std(residuals)$.

2. Related work

2.1. Home Credit Default Risk

Home Credit Default Risk [6] là một cuộc thi được tổ chức bởi Home Credit vào năm 2018 với nỗ lực mở rộng khả năng tiếp cận tài chính cho những người không có tài khoản ngân hàng bằng cách cung cấp trải nghiệm vay tích cực và an toàn. Trong cuộc thi này, các giải pháp top và nhiều cách thức EDA đã được công bố, cung cấp một nguồn tài liệu tham khảo quan trọng cho các cuộc thi liên quan sau đó.

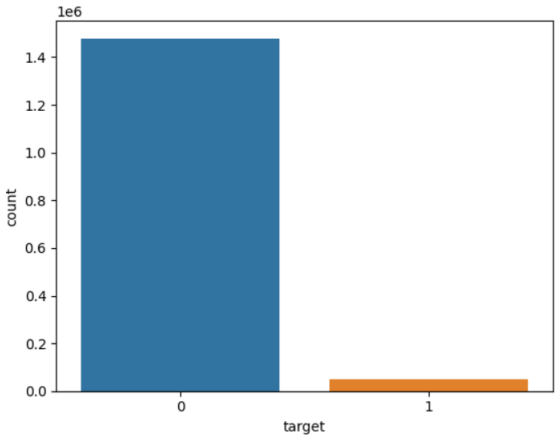
2.2. Credit Risk Model Stability và một số notebook được công bố

Vào thời điểm bắt đầu cuộc thi [4], ban tổ chức đã công bố một phương pháp baseline [3] với hy vọng tạo điều kiện thuận lợi cho các đội chơi. Baseline này chỉ sử dụng một vài bảng và đạt được điểm số tương đối khiêm tốn với 0.361 trên public test. Sau đó, một notebook khác [11] đã được công bố bởi người chơi với việc sử dụng tất cả các bảng, áp dụng các phương pháp tổng hợp các bảng có depth > 0, sử dụng count encoding cho các biến categorical. Phương pháp này đạt được kết quả không quá tệ vào thời điểm được công bố. Không lâu sau, [2] cho thấy phương pháp soft voting giữa các ensemble model cho được kết quả tốt hơn (0.586 trên public test).

Với một số hạn chế tồn tại trong metric của ban tổ chức, vài đội đạt thứ hạng cao đã sử dụng các cách hack metric (các phương pháp này được giữ kín) để cho được kết quả tốt hơn. Mãi cho đến khi [13], một cách hack metric với 0.692 trên public test, được công bố thì bảng xếp hạng mới bị bùng nổ. Tuy nhiên, phương pháp này khi vào private test không cho kết quả cải thiện đáng kể so với phương pháp không dùng hack metric.

3. EDA

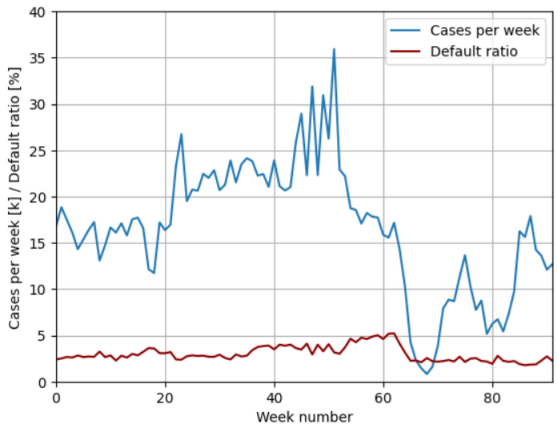
3.1. Imbalanced Dataset Challenges



Hình 2: Count plot của target

Dữ liệu của tập train bị mất cân bằng rất lớn (Hình 2) với 96.85% trường hợp không vỡ nợ (lớp âm) và 3.14% trường hợp vỡ nợ (lớp dương). Điều này phản ánh đúng với thực tế vì những trường hợp vỡ nợ thường chiếm tỷ lệ rất thấp. Ngoài ra, hình 3 cho thấy rằng trong từng giai đoạn tuần riêng biệt, tỷ lệ vỡ nợ không thực sự thay đổi nhiều

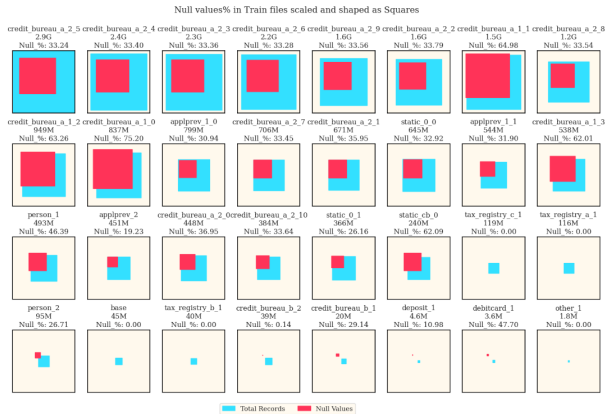
theo thời gian, mặc dù tổng số hợp đồng mỗi tuần có thể thay đổi đáng kể.



Hình 3: Tỷ lệ vỡ nợ và số lượng giao dịch trong các tuần khác nhau.

3.2. Large scale and Null data

Có 32 tệp dữ liệu khác nhau trong thư mục train csv_files, tương tự ở tập test. Các tệp tương tự cũng được cung cấp ở định dạng .parquet trong thư mục tương ứng. Một số tệp csv có kích thước lớn hơn 2 GB. Tệp train_base.csv có 1.526.659 case_id duy nhất, bằng với độ dài của tệp này. Xem chi tiết thông tin ở hình 4.



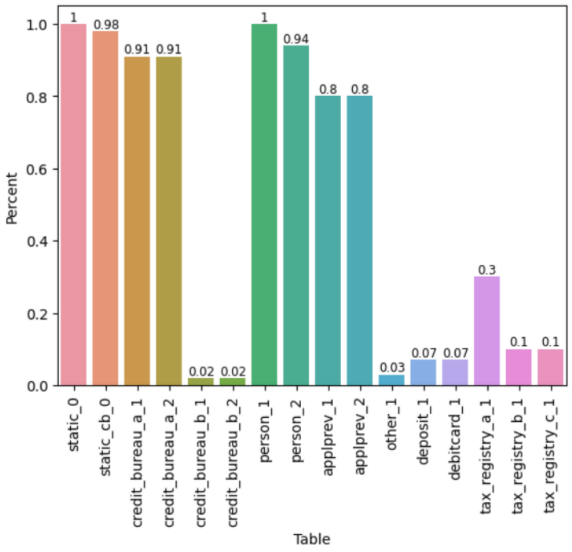
Hình 4: Kích thước các file trong tập huấn luyện và số lượng Nulls.

Kích thước của các hình vuông được điều chỉnh theo tất cả các tệp train, trong đó diện tích lớn nhất là tệp lớn nhất. Các hình vuông màu xanh là tổng số records trong tất cả các cột của tệp train tương ứng. Các hình vuông màu đỏ là tổng số bản records null trong tất cả các cột của tệp train tương ứng.

Trong đó, có thể thấy một số bảng gần như đầy đủ thông tin, trong khi một số bảng khác lại có nhiều ô dữ liệu bị thiếu giá trị. credit_bureau_a_1_* là các files có số lượng nhiều nhất trong tập dữ liệu. Ngoài ra, các files trong tập train gần như đều có tỷ lệ null lớn hơn 30%.

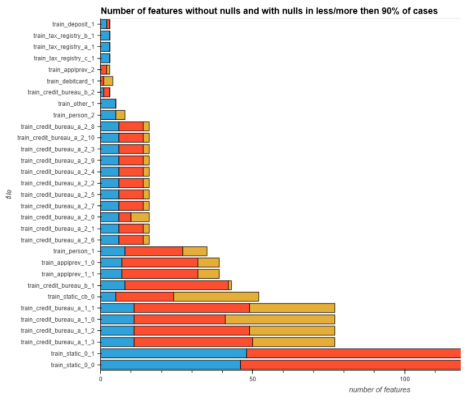
3.3. Case ids in tables

Hình 5 cho thấy phần lớn các bảng đều có case_id chiếm tỷ lệ cao, $>= 80\%$. Có vài bảng chiếm tỷ lệ tương đối trung bình như 3 bảng tax_registry_*. Tuy nhiên, vẫn có các bảng chiếm tỷ lệ rất nhỏ, $< 0.8\%$, bao gồm 5 bảng: credit_bureau_b_1, credit_bureau_b_2, other_1, deposit_1, debitcard_1. Qua đó ta thấy được 5 bảng này có lượng thông tin rất ít để khai thác, nên nhóm sẽ không tập trung feature engineering trên các bảng này và thậm chí có thể không dùng trong quá trình huấn luyện mô hình.



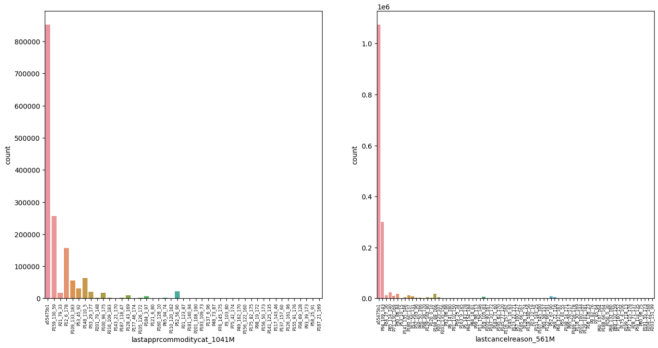
Hình 5: Biểu đồ thể hiện tỷ lệ giữa số lượng case_id duy nhất của từng bảng so với số lượng case_id duy nhất của bảng train_base

3.4. Feature Analysis



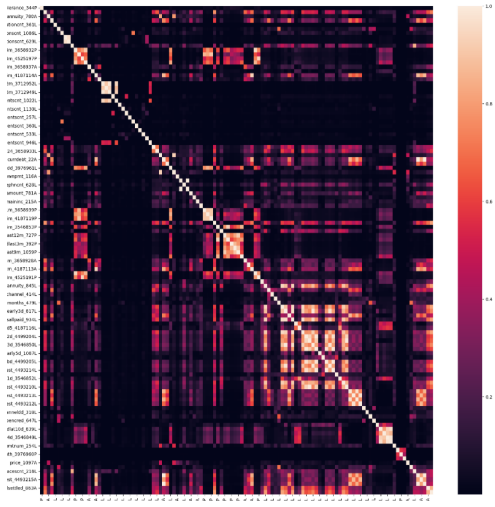
Hình 6: Biểu đồ thể hiện số lượng các đặc trưng không có giá trị null (lam), có giá trị null ít hơn 90% (vàng) và nhiều hơn 90% (đỏ).

Từ hình 6, ta có những nhận xét sau: Phần lớn các đặc trưng trong các bảng có giá trị null nhiều hơn 90%, điều này có thể ảnh hưởng đến quá trình phân tích và cần được xử lý phù hợp; Một số bảng có số lượng đặc trưng không có giá trị null rất ít, cho thấy chất lượng dữ liệu có thể cần được cải thiện; Sự phân bố của các đặc trưng với giá trị null ít hơn 90% cũng cần được chú ý để đảm bảo không bỏ qua những thông tin quan trọng trong quá trình phân tích.



Hình 7: Countplot của lastapprcommoditycat_1041M (trái) và lastcancelreason_561M (phải)

Một số đặc trưng categorical chứa nhiều thể loại với tần số xuất hiện rất thấp như lastapprcommoditycat_1041M, lastcancelreason_561M, v.v. (Xem hình 7). Việc có quá nhiều thể loại, đặc biệt là các thể loại hiếm, có thể dẫn đến nhiễu trong dữ liệu. Nếu sử dụng one-hot encoding cho đặc trưng này cũng sẽ dẫn đến tình trạng có quá nhiều cột.



Hình 8: Correlation heatmap của bảng static_0

Ngoài ra, nhiều đặc trưng số có mối quan hệ gần tuyến tính với nhau. Do dữ liệu có rất nhiều bảng, cũng như nhiều đặc trưng số nên nhóm sẽ chỉ dùng bảng static_0 để báo cáo như hình 8. Các đặc trưng có độ tương quan cao thường gây ra các ảnh hưởng xấu như: đặc trưng bị dư thừa, không những không đóng góp vào việc cải thiện performance của mô hình mà còn làm tăng chi phí tính toán; multicolinearity dẫn đến sự ước lượng hệ số không chính xác và mất ổn định.

4. Data Preprocessing

Các bảng credit_bureau_b_1, credit_bureau_b_2, other_1, deposit_1, debitcard_1 sẽ bị nhóm bỏ qua vì lượng dữ liệu quá ít (Xem hình 5).

4.1. Feature engineering

Được truyền cảm hứng từ top 1 [14] trong cuộc thi Home Credit Default Risk [6], nhóm cố gắng tạo được nhiều đặc trưng nhất có thể để huấn luyện mô hình. Trong giai đoạn này, nhóm sẽ xử lý dữ liệu một cách tuần tự theo các bước được trình bày sau đây. Đầu tiên, nhóm sẽ xử lý dữ liệu, thêm xóa đặc trưng trên các bảng sau:

base: Tạo thêm đặc trưng month_decision, week_decision từ cột date_decision

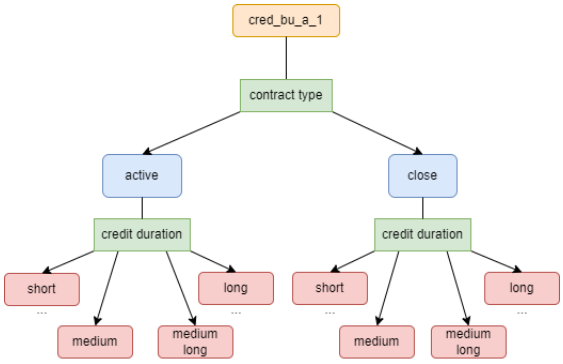
person_1: Trong bảng này, num_group = 0 chính là người làm đơn, các num_group còn lại chính là những người bảo lãnh. Nhóm sẽ tách bảng này thành 2 bảng tương ứng, với bảng người làm đơn sẽ có depth=0, bảng người bảo lãnh sẽ có depth=1. Việc này sẽ giúp ta bảo toàn được thông tin người làm đơn bởi vì chúng sẽ không bị tổng hợp với người bảo lãnh trong giai đoạn aggregation. Ngoài ra, việc này cũng giúp ta tăng thêm số lượng đặc trưng trên person_1 lên 2 lần.

tax_registry_{a;b;c}: Nhóm đã lọc bỏ các cột "tên của nhà tuyển dụng" name_4527232M, name_4917606M, employername_160M tương ứng với từng bảng a, b, c. Lý do là vì các cột này có số lượng giá trị duy nhất rất lớn (name_4527232M: 147037, name_4917606M: 55857, employername_160M: 152835) và cũng không có quá nhiều ý nghĩa khi sử dụng.

credit_bureau_a_1: Đây chính là bảng mà nhóm có thể tạo ra được nhiều feature nhất. Quy trình thực hiện được minh họa trong hình 9. Cụ thể, nhiều đặc trưng trong bảng sẽ có thể được gôm nhóm thành active và close. Vì vậy, nhóm sẽ tách bảng này thành 2 bảng active_credit_bureau_a_1 và close_credit_bureau_a_1. Nếu để ý, trong mỗi bảng này sẽ có 2 đặc trưng về ngày bắt đầu, kết thúc của hợp đồng (dateofcredstart_739D, dateofcredend_289D cho active và dateofcredstart_181D, dateofcredend_353D cho close). Như vậy, ta có thể tạo thêm một đặc trưng là credit_duration. Dựa trên đặc trưng mới tạo này, mỗi bảng con sẽ được tách thành 4 bảng con nữa: short (Nếu $credit\_duration \in [0, 120)$), medium (Nếu $credit\_duration \in [120, 240)$), medium long (Nếu $credit\_duration \in [240, 480)$), long (Nếu $credit\_duration \in [480, +\infty)$). Ở đây, chúng tôi chọn 4 tháng = 120 ngày để làm cột mốc tách. Cuối cùng, dựa trên đặc trưng "công ty tài chính" (financialinstitution_591M cho active, financialinstitution_382M cho close), một lần nữa, ta sẽ tạo ra các bảng nhỏ hơn.

Sau khi thêm và loại bỏ một vài đặc trưng trên từng bảng, ta sẽ tiếp tục thực hiện các thao tác sau:

Aggregation: Do dữ liệu có các bảng depth > 0, nghĩa là ứng với một case_id, chúng ta sẽ có nhiều hàng tương ứng trong bảng. Vì vậy, chúng ta cần tổng hợp các hàng này



Hình 9: Quy trình chia và tạo thêm đặc trưng của credit_bureau_a_1

lại thành 1 hàng duy nhất. Nhóm sẽ sử dụng các cách tổng hợp như sau:

- Đối với đặc trưng categorical: max, min, mode, n_unique, count (không xét giá trị null).
- Đối với đặc trưng không phải categorical (bao gồm số, ngày và không bao gồm các cột WEEK_NUM, case_id, MONTH, num_group_1, num_group_2): max, min, mean.
- Đối với num_group: min, max, count

Join: Sau khi chuẩn bị xong các bảng, ta sẽ tiến hành join các bảng lại với nhau để tạo thành bảng hoàn chỉnh.

Loại bỏ cột có null quá nhiều: Loại bỏ các cột có tỷ lệ null vượt quá threshold = 0.999. Trong thực nghiệm, nhóm sẽ thử một số ngưỡng để tạo ra các tập data khác nhau.

Tạo các đặc trưng khoảng thời gian: Ứng với một đặc trưng có kiểu date, ta sẽ tính số ngày từ nó đến date_decision

Xử lý các đặc trưng categorical: Đầu tiên, nhóm sẽ thay thế giá trị null bằng token đặc biệt __null__. Ngoài ra, trong một đặc trưng categorical, nếu tần suất của một thể loại nào đó $<= 0.001$ thì sẽ bị loại bỏ và được thay thế bằng một thể loại đặc biệt __other__. Việc này giúp loại bỏ nhiễu giúp tăng tính tổng quát hóa của mô hình.

Tạo các đặc trưng tỷ lệ tiền: Một trong những dấu hiệu nhận biết khả năng vỡ nợ của khách hàng là chỉ số credit utilization [12], chỉ số này được tính như sau: $\frac{total\_balances}{total\_credit\_limit}$. Dựa trên khái niệm này, nhóm đã thực nghiệm trên bảng credit_bureau_a_1 và cho kết quả khả quan trên local CV. Vì vậy nhóm quyết định thêm các đặc trưng tỷ lệ giữa các đặc trưng tiền và đặc trưng credamount_770A. Ngoài ra, nhóm cũng sẽ sử dụng thêm mainoccupationinc_384A làm mẫu số để tăng thêm số lượng đặc trưng. Đối với trường hợp tỷ lệ là null hoặc nan, nhóm sẽ thay thế bằng giá trị lớn như 10^{10}

Loại bỏ đặc trưng dựa trên số lượng thể loại: Nhóm sẽ loại bỏ các đặc trưng chỉ có một giá trị và các đặc trưng categorical có nhiều hơn 200 thể loại.

4.2. Memory usage optimization

Vì đây là bài toán liên quan đến dữ liệu lớn, nên việc sử dụng bộ nhớ hiệu quả rất quan trọng. Nhóm sẽ sử polar để tiền xử lý dữ liệu thay vì dùng pandas. Bên cạnh đó, nhóm cũng sẽ ép kiểu dữ liệu số nguyên, số thực về kiểu sử dụng ít dung lượng hơn nhưng vẫn đảm bảo miền giá trị của chúng. Còn với kiểu categorical, nhóm sẽ ép thành kiểu pl.Enum.

4.3. Feature selection

Correlation: Việc xóa đi những đặc trưng có độ tương quan cao sẽ giúp tăng tính tổng quát của mô hình, cải thiện hiệu suất, tăng cường tính ổn định trong việc xếp hạng độ quan trọng của các đặc trưng. Vì vậy nhóm quyết định thực hiện loại bỏ các đặc trưng này, các bước thực hiện như sau:

Bước 1: Nhóm các đặc trưng thành các nhóm (null_group) dựa trên số lượng null của chúng. Mỗi null_group sẽ chứa các đặc trưng có số lượng null bằng nhau.

Bước 2: Ứng với mỗi null_group, ta sẽ tiếp tục gom các đặc trưng dựa trên độ tương quan của chúng (gọi là correlation_group). Đầu tiên, mỗi correlation_group đều có 1 đặc trưng làm gốc, sau đó, ta sẽ lặp qua các đặc trưng khác để kiểm tra xem nó có thuộc cùng nhóm với đặc trưng gốc để đưa vào nhóm hay không. Hai đặc trưng x, y được xem là cùng nhóm nếu $f(x, y) >= threshold$. Với:

$$f(x,y) = \begin{cases} cont\_coef(x,y), & \text{Nếu x, y là categorical} \\ |pearson\_corr(x,y)|, & \text{Nếu x, y là số thực} \\ 0, & \text{Còn lại} \end{cases} \quad (5)$$

Trong đó, $cont\_coef$ là Pearson's contingency coefficient dùng để đo mức độ liên kết giữa 2 biến categorical. Nó được bắt nguồn từ thống kê chi bình và cung cấp một thang đo chuẩn hóa, có miền giá trị thuộc $[0; 1)$ với 0 cho việc không có liên kết và 1 cho việc liên kết hoàn hảo giữa 2 biến. Còn $pearson\_corr$ là độ tương quan Pearson, dùng để đo mức độ tương quan giữa 2 biến numerical. $pearson\_corr \in [-1, 1]$, nếu giá trị càng gần 1 hoặc -1 thì tương quan càng cao, còn nếu gần 0 thì tương quan thấp. Lưu ý, các đặc trưng đã được phân nhóm sẽ không được xét trong lần lặp tiếp theo và giá trị threshold trong trường hợp 2 biến categorical hay 2 biến numerical có thể khác nhau.

Bước 3: Trong mỗi correlation_group, ta sẽ giữ lại đặc trưng có số giá trị duy nhất cao nhất.

Feature Importance của LightGBM: Do gặp phải vấn đề tràn bộ nhớ khi huấn luyện CatBoost [9], XGBoost [1] trên dữ liệu được feature selection với correlation, nên nhóm quyết định giảm số lượng đặc trưng xuống. Những đặc trưng được giữ lại sẽ có $lgbm\_importance >= threshold$, trong đó, $lgbm\_importance$ là độ quan trọng trung bình của 1 đặc trưng trong 5-fold cross-validation của LightGBM [5].

5. Machine Learning Models

Trong cuộc thi, nhóm chủ yếu sử dụng các thuật toán Gradient Boosting. Gradient Boosting hoạt động bằng cách thêm tuần tự các dự đoán vào một nhóm, mỗi dự đoán sẽ sửa lỗi trước đó, phương pháp này cố gắng điều chỉnh bộ dự đoán mới phù hợp với các lỗi còn lại do bộ dự đoán trước đó tạo ra. Các bước của thuật toán này như sau:

- Bước 1: Thuật toán cơ sở đọc dữ liệu và gán trọng số bằng nhau cho mỗi mẫu dữ liệu.
- Bước 2: Những dự đoán sai do thuật toán cơ sở đưa ra sẽ được xác định. Trong lần lặp tiếp theo, tiến hành cập nhật model chính.
- Bước 3: Lặp lại bước 2 cho đến khi thuật toán đạt được điều kiện dừng.

5.1. XGBoost

Mô hình XGBoost [1] (regularizing gradient boosting) được sử dụng nhiều ở các cuộc thi Kaggle. XGBoost sử dụng cả L1 và L2 để trừng phạt mô hình rất phức tạp, đồng thời, mô hình này có khả năng xử lý dữ liệu thưa thớt có thể được tạo ra từ các bước tiền xử lý hoặc giá trị thiếu. Tiếp cận theo chiều sâu này cải thiện đáng kể hiệu suất tính toán. Cải tiến quan trọng của XGBoost là chỉ xem xét các mục không bị thiếu, xem việc không xuất hiện như một giá trị thiếu và học cách xử lý giá trị thiếu một cách tốt nhất.

Tuy nhiên, XGBoost vẫn có 1 số điểm hạn chế như sử dụng thuật toán dựa trên sắp xếp trước để học cây quyết định nên mô hình vẫn có thể mất thời gian và đòi hỏi tài nguyên tính toán đáng kể đối với các tập dữ liệu lớn. Ngoài ra, nếu không được điều chỉnh một cách thích hợp hoặc nếu dữ liệu có nhiễu, tập dữ liệu huấn luyện nhỏ nhưng sử dụng mô hình phức tạp có thể dẫn đến hiện tượng overfitting.

Siêu tham số của XGBoost: XGBoost bao gồm nhiều siêu tham số cần được tinh chỉnh để đạt hiệu suất tối ưu. Một số siêu tham số quan trọng bao gồm:

- Boosting type (booster): Loại boosting được sử dụng.
- Objective function (objective): Mục tiêu của mô hình.
- Evaluation Metric (eval_metric): Chỉ số đánh giá hiệu quả của mô hình.
- Maximum Depth (max_depth): Cung cấp về độ sâu của cây, nó cũng là tham số để kiểm soát được vấn đề Overfitting của mô hình. Nếu cảm thấy mô hình bị Overfitting, hãy giảm tham số này xuống.
- Learning Rate (learning_rate): Tốc độ học của mô hình. Giá trị nhỏ hơn thường dẫn đến quá trình huấn luyện chậm hơn nhưng có thể mang lại mô hình tốt hơn.
- Number of Estimators (n_estimators): Số lượng cây (iterations) mà mô hình sẽ huấn luyện.
- Feature fraction (colsample_bytree): Tỷ lệ các cột (features) được chọn để xây dựng mỗi cây.
- Feature fraction by Node (colsample_bynode): Tỷ lệ các cột được chọn mỗi khi một nút được chia tách.
- L1 regularization (alpha): Hệ số của L1 regularization term, giúp giảm overfitting.
- L2 regularization (lambda): Hệ số của L2 regularization term, cũng giúp giảm overfitting.

5.2. LightGBM

Tương tự như XGBoost, LightGBM [5] của Microsoft là một framework phân tán hiệu năng cao sử dụng cây quyết định cho nhiệm vụ xếp hạng, phân loại và hồi quy. LightGBM sử dụng Leaf-wise (Best first) Tree growth, nó sẽ chọn lá có độ lỗi delta tối đa để phát triển.

LightGBM sử dụng thuật toán dựa trên biểu đồ histogram để nhóm các giá trị đặc trưng (thuộc tính) liên tục vào các thùng riêng biệt từ đó giúp giảm thời gian huấn luyện và bộ nhớ sử dụng. LightGBM là mô hình có tốc độ xử lý nhanh hơn XGBoost, điều này có thể được lý giải do LightGBM có sử dụng GOSS (Gradient Based One Side Sampling) và EFB (Exclusive Feature Bundling). Bên cạnh đó, LightGBM cũng được khuyến khích sử dụng để giải quyết các tập dữ liệu lớn có cấu trúc.

Siêu tham số của LightGBM: LightGBM cho phép người dùng có thể thiết lập hơn 100 siêu tham số. Trong đó, có những siêu tham số quan trọng như sau:

- Boosting type (boosting_type): Loại boosting được sử dụng.
- Objective function (objective): Mục tiêu của mô hình.
- Evaluation metric (metric): Chỉ số đánh giá hiệu quả của mô hình.
- Maximum depth (max_depth): Cung cấp về độ sâu của cây, nó cũng là tham số để kiểm soát được vấn đề Overfitting của mô hình. Nếu cảm thấy mô hình bị Overfitting, hãy giảm tham số này xuống.
- Learning rate (learning_rate): Tốc độ học của mô hình. Giá trị nhỏ hơn thường dẫn đến quá trình huấn luyện chậm hơn nhưng có thể mang lại mô hình tốt hơn.
- Number of estimators (n_estimators): Số lượng cây (iterations) mà mô hình sẽ huấn luyện.
- Feature fraction (colsample_bytree): Tỷ lệ các cột (features) được chọn để xây dựng mỗi cây.
- Feature fraction by node (colsample_bynode): Tỷ lệ các cột được chọn mỗi khi một nút được chia tách.
- L1 regularization (reg_alpha): Hệ số của L1 regularization term, giúp giảm overfitting.
- L2 regularization (reg_lambda): Hệ số của L2 regularization term, giúp giảm overfitting.
- Extra trees (extra_trees): Cây sẽ lựa chọn ngẫu nhiên một số điểm chia tách, tăng sự ngẫu nhiên và giúp giảm overfitting.
- Number of leaves (num_leaves): Số lượng lá tối đa cho mỗi cây. Số lượng lá càng nhiều thì mô hình càng phức tạp, nhưng cũng dễ bị overfitting.

5.3. CatBoost

CatBoost [9] là một mô hình được kết hợp từ hai nhánh "Category" và "Boosting". Được phát triển bởi các nhà nghiên cứu và kỹ sư của Yandex, nó là sự kế thừa của thuật toán MatrixNet được sử dụng rộng rãi trong công ty để xếp hạng các nhiệm vụ, dự báo và đưa ra đề xuất. CatBoost cho phép tự động xử lý các đặc trưng hạng mục và cho phép Fast Gradient Boosting trên Cây quyết định bằng cách sử dụng GPU.

Nhóm lựa chọn sử dụng CatBoost trong cuộc thi này để giải quyết các trường dữ liệu dạng phân loại (categorical)

vì CatBoost được cho là sẽ đem lại hiệu suất tốt hơn các thuật toán khác trong việc xử lý các dạng dữ liệu kiểu này.

Siêu tham số của CatBoost: Vì nhóm tập trung vào việc điều chỉnh tham số LightGBM nên có khả năng mô hình LightGBM sẽ bị overfit trên tập dữ liệu test. Nên ở CatBoost, nhóm sẽ coi model này như một model hỗ trợ, kết hợp (essemble với LightGBM) để làm giảm sự overfitting của toàn bộ pipeline của nhóm. Chính vì vậy CatBoost sẽ được huấn luyện hầu như trên các tham số default của mô hình, và nhóm chỉ thay đổi một số tham số chính sau:

- Evaluation Metric (eval_metric): Chỉ số đánh giá hiệu quả của mô hình.
- Traning Type (task_type): Lựa chọn huấn luyện model trên GPU hay CPU.
- Learning Rate (learning_rate): Chỉ số đánh giá hiệu quả của mô hình.
- Number of Iterations (iterations): Xác định số lượng cây mà CatBoost sẽ xây dựng trong quá trình huấn luyện.

5.4. Blending

Kĩ thuật Blending mà nhóm sử dụng là Soft Voting, một kĩ thuật kết hợp các dự đoán xác suất của các mô hình. Phương pháp này thường đem lại hiệu suất tốt hơn với so với các dự đoán riêng lẻ của từng mô hình. Soft Voting có thể được mô tả như sau:

1) Giả sử nhóm có một danh sách các mô hình $M = \{M_1, M_2, ..., M_k\}$, trong đó k là số lượng mô hình trong danh sách.
2) Mỗi mô hình M_i có thể được mô tả bằng một hàm dự đoán f_i, trong đó $f_i(X)$ là dự đoán của mô hình M_i trên dữ liệu đầu vào X.
3) Phương pháp kết hợp các dự đoán từ các mô hình này được thực hiện bằng cách tính trung bình của các dự đoán:

$$\hat{y}(X) = \frac{1}{k} \sum_{i=1}^{k} f_i(X) \tag{6}$$

6. Experiment

6.1. Our pipeline

Sau nhiều lần thảo luận, trao đổi và thực nghiệm, nhóm xác định được một quy trình để thực hiện bài toán được chia thành từng gian đoạn, bao gồm: Data Processing, Feature Selection, Training Model, Manual Hyperparameter Tuning và cuối cùng là Inference. Việc xác định được một quy trình thực hiện rõ ràng đã giúp nhóm có cải thiện về hiệu suất trên Public LeaderBoard đáng kể khi dễ dàng quản lý code, phân chia nhân lực vào tập trung ở từng giai đoạn khác nhau để cải thiện hiệu quả. Quy trình thực hiện của nhóm như hình 10.

6.1.1. Data Processing

Sau khi thực hiện các bước tiền xử lý như ở mục 4.1, nhóm thu được một bộ dữ liệu gồm 1,526,659 hàng, 1,838 đặc trưng mà trước đó đã loại bỏ 63 cột hầu như là null.

6.1.2. Feature Selection

Từ bộ dữ liệu mới thu được, nhóm thực hiện kĩ thuật Feature Selection mục 4.3 dựa trên độ tương quan giữa các đặc trưng. Tiếp đến, nhóm kết hợp với các bước sau để tìm được các bộ lựa chọn đặc trưng mang lại hiệu suất tốt. Kết qua thu được như sau:

1) Processed Data 1 (PD1): Sử dụng numerical_threshold = 0.8, categorical_threashold = 0.9. Lúc này bộ dữ liệu giảm từ 1,838 đặc trưng xuống còn 858. Trong đó có 703 Numerical Features và 155 Categorical Features.
2) Processed Data 2 (PD2): Sử dụng numerical_threshold = 0.85, categorical_threashold = 0.85. Lúc này bộ dữ liệu giảm từ 1,838 đặc trưng xuống còn 874. Trong đó có 748 Numerical Features và 126 Categorical Features.

Ngoài ra, nhóm còn thực hiện kĩ thuật Feature Selection dựa trên LightGBM Model. Sau khi LightGBM được huấn luyện, nhóm truy xuất tầm quan trọng của các đặc trưng từ mô hình này. Từ đó lại lọc ra một bộ features mới với các có tầm quan trọng thấp đã bị loại bỏ.

6.1.3. Model Training

A. Cross Validation

Nhóm dùng Stratified Group 5 Fold để chia dữ liệu thành các tập huấn luyện và kiểm tra dựa trên WEEK_NUM. Nhóm muốn chia dữ liệu đầu vào với đặc trưng 'X', nhãn 'Y', và chia chúng theo nhóm các tuần trong cột "WEEK_NUM". Sau đó, dữ liệu tuần tự được chia thành các tập huấn luyện và kiểm tra với 5 folds, đảm bảo tỉ lệ các lớp mục tiêu được giữ nguyên trong mỗi fold và các nhóm tuần không bị chia nhỏ. Trong huấn luyện và dự đoán, nhóm sẽ không sử dụng trường case_id, WEEK_NUM.

B. Model Training Phase 1

Từ hai bộ đặc trưng PD1 và PD2, nhóm tiến hành huấn luyện mô hình LightGBM với Cross Valdidation 5 Folds. Quá trình huấn luyện như sau:

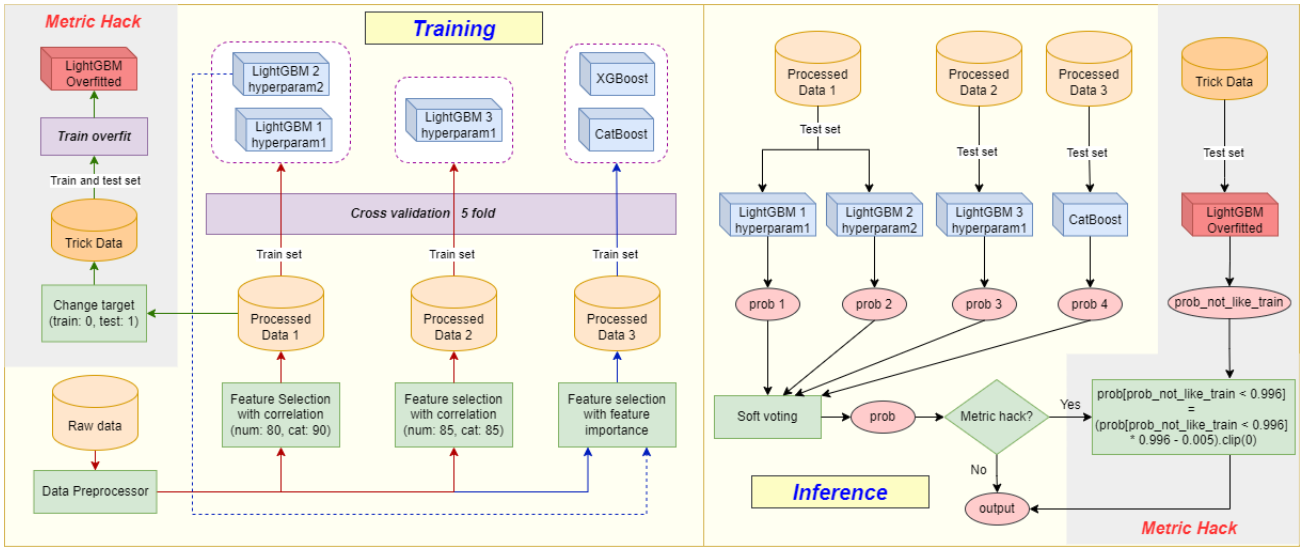
1) Nhóm 1: 5 mô hình LightGBM (hypterparamter 1) trên PD1 (Bảng 4 - TN3).
2) Nhóm 2: 5 mô hình LightGBM (hypterparameter 1) trên PD2 (Bảng 4 - TN7).
3) Nhóm 3: 5 mô hình LightGBM (hypterparameter 2) trên PD1. Do PD1 cho kết quả tốt hơn PD2, nên nhóm chỉ sử dụng hyperparameter 2 cho PD1 (Bảng 4 - TN9).

Chi tiết hyperparameters 1 và 2 có thể xem trong bảng 1

C. Model Training Phase 2

Dựa trên một nhóm mô hình LightGBM, nhóm có thể thu được bộ dữ liệu mới sau khi thực hiện feature selection dựa trên độ quan trọng trung bình của đặc trưng trong nhóm mô hình đó (Mục 4.3). Như vậy, đáng lẽ nhóm sẽ phải thu được 3 bộ dữ liệu mới, nhưng do trước đó, hyperparamter 2 của LightGBM vẫn chưa được dùng, nên nhóm chỉ thu được 2 bộ dữ liệu mới sau:

1) PD3: Dựa trên 5 mô hình LightGBM với hypterparameter 1 trên PD1, thu được 441 đặc trưng trong



Hình 10: Quy trình thực hiện cho kết quả tốt nhất của nhóm. Trong quá trình training (Bên trái), đường màu đỏ là luồng của phase 1, màu lam là luồng của phase 2, màu lục là luồng của metric hacking

Hyperparameter	Hyperparameters 1	Hyperparameters 2
boosting_type	gbdt	gbdt
objective	binary	binary
metric	auc	auc
max_depth	10	20
max_bin	250	255
learning_rate	0.05	0.05
n_estimators	2000	2000
colsample_bytree	0.8	0.8
colsample_bynode	0.8	0.8
reg_alpha	0.1	0.1
reg_lambda	10	10
extra_trees	True	True
num_leaves	64	64
device	cpu	cpu

Bảng 1: Bộ siêu tham số thứ 1 và 2 của LightGBM

đó có 402 Numerical Features và 39 Categorical Features.

2) PD4: Dựa trên 5 mô hình LightGBM với hypter-paramter 1 trên PD2, thu được 433 đặc trưng trong đó có 390 Numerical Features và 43 Categorical Features.

Sau đó, nhóm tiến hành huấn luyện các mô hình CatBoost, XGBoost với Cross Valdidation 5 Folds trên bộ features này. Cuối cùng thu được hai nhóm CatBoost (Bảng 4 - TN4 và TN11) và một nhóm XGBoost (Bảng 4 - TN5), mỗi nhóm có 5 mô hình.

Các siêu tham số nhóm chọn cho việc huấn luyện các mô hình CatBoost, XGBoost lần lượt ở bảng 2, 3. Có thể thấy, siêu tham số cho CatBoost được chọn khá đơn giản.

6.1.4. Post-process

Thang đo Gini Stability 4 có một hạn chế là nếu biết thời điểm giao dịch (date_decision), hay cụ thể hơn là WEEK_NUM, chúng ta hoàn toàn có thể chơi gian lận (hack

Tham số	Giá trị
eval_metric	AUC
task_type	GPU
learning_rate	0.05
iterations	6000

Bảng 2: Các siêu tham số của mô hình

Tham số	Giá trị
booster	gbtree
objective	binary:logistic
eval_metric	auc
max_depth	10
learning_rate	0.05
n_estimators	600
colsample_bytree	0.8
colsample_bynode	0.8
alpha	0.1
lambda	10
tree_method	auto

Bảng 3: Các siêu tham số của mô hình XGBoost

metric) bằng cách giảm gini score trong vài tuần đầu để khiến đường hồi quy có hệ số gần 0 hoặc dương, từ đó không bị phạt nặng bởi falling rate. Tuy nhiên, cột date_decision và WEEK_NUM đã bị chuyển đổi, nên ta cần một cách gián tiếp để biết được các giao dịch thuộc các tuần đầu tiên.

Theo [13], do tập test ngoài chứa các WEEK_NUM tương lai, còn có chứa WEEK_NUM trong tập train nên có thể xem những mẫu trong tập train sẽ ứng với các WEEK_NUM đầu tiên. Cộng với xu hướng các mẫu dữ liệu thường có chung đặc điểm tại những thời điểm gần nhau nên chúng ta sẽ sử dụng một mô hình để dự đoán mẫu nào trong tập test sẽ có WEEK_NUM gần với tập train. Để làm được điều này, nhóm thực hiện như phần Metric Hack ở hình 10. Cụ thể, nhóm lấy một bộ dữ liệu đã có sẵn (PD1), sau đó sửa target của tập train thành 0 và tập test thành 1

rồi nối 2 tập này lại với nhau tạo thành Trick Data. Tiếp đến train overfit 1 mô hình LightGBM (Max depth = -1) trên bộ dữ liệu này. Sau đó, mô hình sẽ dự đoán xác suất không giống với tập train của một mẫu trong tập test. Cuối cùng, nhóm kết hợp chúng với những dự đoán từ pipeline không sử dụng hack (prob) để giảm Gini Score trong các tuần đầu tiên như sau:

$$\text{prob}[\text{prob\_not\_like\_train} < 0.996]$$

$$= (\text{prob}[\text{prob\_not\_like\_train} < 0.996] \times 0.996 - 0.005)^{+} \tag{7}$$

Trong đó: $(\cdot)^{+}$ biểu thị việc cắt các giá trị nhỏ hơn 0 để chúng bằng 0.

6.2. Comparison

Trong quá trình tham gia cuộc thi, nhóm thực hiện việc tạo ra rất nhiều pretrained model cũng như các tập feature để đạt được điểm số tốt nhất. Nhưng nhóm sẽ chỉ trình bày một số pipeline mà nhóm cho là có ảnh hưởng tích cực trong việc thiết kế và tìm ra các phương pháp tốt nhất. Quá trình thực nghiệm của nhóm được trình bày trong bảng 4.

Từ bảng ta thấy được, mô hình LightGBM có ảnh hưởng tích cực nhất đến kết quả của nhóm. Bởi vì, mô hình này được huấn luyện trên một tập dữ liệu có nhiều đặc trưng hơn nhiều so với XGBoost, CatBoost và một phần cũng vì nhóm tập trung nhiều hơn vào việc tinh chỉnh hyperparameters của LightGBM so với các mô hình còn lại.

Mặc dù mô hình CatBoost cho kết quả đơn không quá tốt so với LightGBM nhưng khi ensemble lại cải thiện kết quả, việc này cho thấy mô hình CatBoost giúp tăng tính tổng quát hóa của toàn bộ pipeline ensemble. Mô hình XGBoost cho kết quả là 0.573 trên publib test, một kết quả không quá tốt, và khi ensemble thì kết quả (không được đề cập trong bảng) cũng không được cao như các phương pháp ensemble được đề cập trong bảng. Ngoài ra ta có thể thấy được suy giảm điểm số đáng kể khi từ public test chuyển sang private test.

Phương pháp metric hack cho kết quả cải thiện đáng ngạc nhiên ở public test, nhưng sang private test lại chỉ cải thiện được một phần nhỏ so với pipeline gốc. Việc này có thể được lý giải bởi public test thực chất là 30% khoảng thời gian đầu của toàn bộ tập test, chứa nhiều hợp đồng có thời điểm gần với tập train, còn private test là 70% tiếp theo. Có thể thấy, các điểm dữ liệu ở mốc thời điểm quá xa có phân phối rất khác so với những điểm dữ liệu trong tập train nên mô hình LightGBM overfit sẽ khó phát hiện được nhưng tuần đầu tiên trong private test.

7. Conclusion

Qua cuộc thi này, nhóm đã học hỏi được nhiều kiến thức về xử lý liệu lớn, mô hình và kể cả các kiến thức có liên quan đến ngân hàng. Kết thúc cuộc thi, với pipeline không có sử dụng hack, nhóm kết hợp sử dụng 15 mô hình LightGBM và 5 mô hình CatBoost. Kết quả nhóm đạt được điểm số 0.59905 trên public test, và 0.52769 trên private test. Với pipeline có sử dụng hack, khá tương tự với pipeline có không sử dụng hack tuy nhiên lại có thêm một quá trình post-processing để bổ sung vào, kết quả nhóm đạt được 0.66287 trên public test và 0.52782 trên private test, đạt vị trí 66/3883 toàn bộ cuộc thi.

TN	Pipeline	Public Test	Private Test
1	Baseline ban tổ chức	0.36	0.244
2	5 LGBM hyperparam1, 5 CatBoost (Không dùng LGBM feature) trước khi thêm đặc trưng	0.588	0.511
3	5 LGBM hyperparam1 trên PD1	0.594	0.522
4	5 CatBoost trên PD3	0.588	0.514
5	5 XGB trên PD3	0.573	0.495
6	TN3 + TN4	0.596	0.523
7	5 LGBM hyperparam1 trên PD2	0.593	0.526
8	TN7 + TN4	0.595	0.523
9	5 LGBM hyperparam2 trên PD1	0.597	0.524
10	TN3 + TN4 + TN9	0.598	0.52765
11	TN3 + TN7 + TN9 + 5 CatBoost trên PD4	0.598	0.525
12	TN3 + TN7 + TN9 + TN4	0.599	0.52769
13	TN2 + Metric Hack	0.652	0.515
14	TN12 + Metric Hack	**0.662**	**0.52782**

Bảng 4: Kết quả của các pipeline trên Public Test

Sau cuộc thi này, nhóm sẽ cố gắng học hỏi các pipeline của những người sở hữu thứ hạng cao, xem những thảo luận hay những góc nhìn của những người tham gia cuộc thi để có thể cải thiện điểm số, đồng thời loại bỏ được các bước xử lý dữ liệu không cần thiết, tinh gọn toàn bộ quy trình của nhóm. Ngoài ra, nhóm cũng sẽ chạy thử các phương pháp hyperparameter tuning như grid search, random search, bayesian search (Optuna) để tìm được các bộ siêu tham số tối ưu, thứ mà nhóm vẫn chưa thực hiện được trên toàn bộ dữ liệu do hạn chế về thời gian chạy thực nghiệm trên Kaggle (tối đa khoảng 12 tiếng cho một lần chạy).

8. Acknowledgement

Chúng tôi muốn gửi lời cảm chân thành đến ThS. Nguyễn Vũ Anh Khoa và thầy Trương Quốc Trường đến từ Đại học Công nghệ Thông tin vì đã hướng dẫn và chỉ bảo tận tình trong quá trình tiến hành đồ án của nhóm. Sự chuyên nghiệp và tầm nhìn sâu sắc của hai thầy đã có đóng góp to lớn đến sự thành công của đồ án này. Ngoài ra, nhóm cũng xin cảm ơn đến những người chơi đã công bố các công trình như [7] [8] [10] [13] [3] [11],... cũng như toàn bộ cộng đồng Kagglers đã giúp nhóm hiểu rõ hơn về bài toán. Cuối cùng, bảng phân công của nhóm như bảng 5.

Tài liệu

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, August 2016.

[2] harrychan123. (lgb + cat ensemble) +stacking, April 2024. URL https://www.kaggle.com/code/harrychan123/lgb-cat-ensemble-stacking.

MSSV	Họ và tên	Công việc	Mức độ hoàn thành
21522542	Đoàn Nhật Sang	• Phân công, quản lý công việc • EDA trên static_0, static_cb_0, credit_bureau_*_* • Feature engineering trên bảng static_0, static_cb_0, credit_bureau_a_* • Tìm hiểu metric hack • Chạy Optuna để tìm kiếm hyperparamter tối ưu nhưng thời gian chạy quá lâu, thường xuyên bị ngắt trên kaggle. Sau đó, tìm hiểu một số notebooks và phát hiện hyperparamters tốt cho LightGBM, XGBoost • Viết report phần Related work, Data preprocessing, Acknowledgement, References. • Thuyết trình	100%
21520274	Trương Văn Khải	• Phân công, quản lý công việc • EDA trên other_1, tax_registry_a_1, tax_registry_b_1, tax_registry_c_1. • Chạy baseline cũ với các cách mã hóa categorical khác nhau như: Count Encoding, Target Encoding, Label Encoding, WOE Encoding. • Huấn luyện mô hình, kiểm tra mô hình trên nhiều cách feature selection. • Ensemble các mô hình lại với các trọng số khác nhau (tune bằng tay), tune các hyperameter trong metric hack • Viết report phần Introduction, Machine Learning Models, Experiments. • Thuyết trình	100%
21520195	Lê Ngô Minh Đức	• EDA trên applprev_1, applprev_2, deposit_1. • Feature engineering trên các bảng applprev_*. • Tìm hiểu, cài đặt Deep Feature Synthesis, XGBoost • Viết report phần EDA.	100%
21520696	Hoàng Tiến Đạt	• EDA trên debitcard_1, person_*. • Feature engineering trên bảng person_*. • Cài đặt feature selection • Làm slide.	100%

Bảng 5: Bảng phân công công việc

[3] Daniel Herman. Home credit 2024 starter notebook, February 2024. URL https://www.kaggle.com/code/jetakow/home-credit-2024-starter-notebook.

[4] Daniel Herman, Tomas Jelinek, Walter Reade, Maggie Demkin, and Addison Howard. Home credit - credit risk model stability. https://kaggle.com/competitions/home-credit-credit-risk-model-stability, 2024.

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[6] Anna Montoya, KirillOdintsov, and Martin Kotek. Home credit default risk, 2018. URL https://kaggle.com/competitions/home-credit-default-risk.

[7] IGOR PI. [home credit 2024] eda part i, 2024. URL https://www.kaggle.com/code/pib73nl/home-credit-2024-eda-part-i.

[8] IGOR PI. [home credit 2024] eda part ii, 2024. URL https://www.kaggle.com/code/pib73nl/home-credit-2024-eda-part-ii.

[9] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features, 2019.

[10] skrydg. kaggle-home-credit-credit-risk-model-stability, 2024. URL https://github.com/skrydg/kaggle_home_credit_risk_model_stability.

[11] takumimukaiyama. Lightgbm_countencoding, April 2024. URL https://www.kaggle.com/code/takumimukaiyama/lightgbm-countencoding.

[12] The Investopia Team and Thomas J. Brock. Credit utilization ratio: Definition, calculation, and how to improve, 2018. URL https://www.investopedia.com/terms/c/credit-utilization-rate.asp.

[13] tritionalval. This is the way, May 2024. URL https://www.kaggle.com/code/tritionalval/this-is-the-way.

[14] Bojan Tunguz. 1st place solution, 2018. URL https://www.kaggle.com/competitions/home-credit-default-risk/discussion/64821.
=2