ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Môn học: CS116.O21
Giảng viên hướng dẫn: Nguyễn Vũ Anh Khoa

Nhóm sinh viên thực hiên: Nhóm 14

Trần Tuyết Minh	21521144
Nguyễn Thị Mai Trinh	21522718
Nguyễn Thị Huyền Trang	21520488
Lê Thị Như Ý	21522818

TP.HCM, ngày 6 tháng 06 năm 2024

Mục lục

1. Giới thiệu

Cuộc thi "Home Credit - Credit Risk Model Stability" trên Kaggle là một sân chơi trí tuệ thu hút các nhà khoa học dữ liệu, kỹ sư học máy và những người đam mê phân tích dữ liệu từ khắp nơi trên thế giới. Cuộc thi đặt ra thử thách đánh giá độ ổn định của các mô hình dự đoán rủi ro tín dụng, một vấn đề quan trọng trong lĩnh vực tài chính.

Nhóm chúng tôi tham gia cuộc thi với mục tiêu mang đến giải pháp hiệu quả cho việc lựa chọn và huấn luyện mô hình học máy có khả năng dự đoán chính xác khả năng thanh toán khoản vay của khách hàng tiềm năng.. Để đạt được mục tiêu này, chúng tôi đã áp dụng một tập hợp các phương pháp tiên tiến, bao gồm:

- Tiền xử lý dữ liệu một cách kỹ lưỡng:
    - Đọc file parquet và sử dụng polars để tăng tốc độ xử lý.
    - Biến đổi dữ liệu dựa trên định nghĩa của từng loại cột.
    - Xử lý dữ liệu nhiều giá trị và dữ liệu depth 1, 2.
    - Giảm số lượng feature bằng cách nhóm các feature có cùng số giá trị NaN và kiểm tra độ tương quan.
    - Chia dữ liệu bằng cross-validation với StratifiedGroupKFold.
- Sử dụng 2 mô hình CatBoostClassifier và LGBMClassifier:
    - Lựa chọn dựa trên đặc trưng của dữ liệu và hiệu quả của mô hình.
    - Cấu hình các tham số phù hợp cho từng mô hình.
    - So sánh hiệu quả giữa hai mô hình.

Lựa chọn các phương pháp này dựa trên các yếu tố chính: *hiệu quả* và *tính sáng tạo.*

Nhờ sự kết hợp hiệu quả của các phương pháp trên, nhóm chúng tôi đã đạt được thứ hạng 134 trong cuộc thi này, với số điểm là *0.51926*

Báo cáo này sẽ trình bày chi tiết về các phương pháp của chúng tôi, kết quả đạt được và những bài học rút ra từ cuộc thi.

2. Các phương pháp liên quan

2.1. Phương pháp liên quan đến bài toán

Bài toán đánh giá độ ổn định của mô hình dự đoán rủi ro tín dụng nhận được sự quan tâm lớn từ cộng đồng nghiên cứu và thực tiễn. Dưới đây là các phương pháp và kỹ thuật liên quan có thể được áp dụng cho bài toán này:

2.1.1. Các kỹ thuật Validation

- **K-Fold Cross-Validation**: Chia dữ liệu thành K phần và luân phiên sử dụng từng phần làm tập kiểm tra để đảm bảo mô hình được đánh giá trên nhiều tập dữ liệu khác nhau.
- **Stratified Cross-Validation**: Đảm bảo mỗi fold trong quá trình cross-validation có tỷ lệ các lớp mục tiêu giống nhau, đặc biệt quan trọng khi dữ liệu mất cân đối.

2.1.2. Sử dụng đa dạng độ đo để đánh giá khách quan

- **ROC-AUC (Receiver Operating Characteristic - Area Under Curve):** Đo lường khả năng phân biệt của mô hình giữa các lớp.

- **Precision-Recall Curve**: Đặc biệt hữu ích khi dữ liệu mất cân đối, đo lường độ chính xác và khả năng thu hồi.
- **F1-Score**: Kết hợp cả precision và recall, đặc biệt hữu ích khi cần cân bằng giữa chúng.

2.1.3. Các độ đo đánh giá độ ổn định

- **Population Stability Index (PSI):** Đánh giá sự thay đổi của phân phối điểm số mô hình giữa hai tập dữ liệu (ví dụ, giữa các thời điểm khác nhau).
- **Characteristic Stability Index (CSI)**: Tương tự PSI nhưng áp dụng cho các đặc trưng đầu vào của mô hình.

2.1.4. Thực hiện kiểm tra độ bền vững

- **Adversarial Testing:** Thử nghiệm mô hình với các mẫu dữ liệu có chứa nhiễu hoặc các thay đổi nhỏ để xem xét độ nhạy của mô hình.
- **Stress Testing**: Kiểm tra mô hình với các kịch bản cực đoan để đánh giá khả năng hoạt động trong điều kiện khắc nghiệt.

2.1.5. Giám sát mô hình

- **Drift Detection**: Sử dụng các kỹ thuật như chi-square test hoặc Kolmogorov-Smirnov test để phát hiện sự thay đổi phân phối dữ liệu đầu vào hoặc kết quả dự đoán theo thời gian.
- **Performance Tracking**: Liên tục theo dõi và ghi nhận các chỉ số hiệu suất của mô hình trên các tập dữ liệu mới để phát hiện sớm các dấu hiệu suy giảm hiệu suất.

2.1.6. Sử dụng các kỹ thuật lựa chọn đặc trưng

- **Recursive Feature Elimination (RFE):** Tự động loại bỏ các đặc trưng không quan trọng để cải thiện hiệu suất và độ ổn định của mô hình.
- **Principal Component Analysis (PCA)**: Giảm chiều dữ liệu bằng cách tìm ra các thành phần chính, giúp giảm nhiễu và cải thiện độ ổn định.

2.1.7. Sử dụng các kỹ thuật Regularization

- **L1 and L2 Regularization**: Sử dụng kỹ thuật điều chỉnh để ngăn chặn việc mô hình quá khớp (overfitting) và cải thiện độ ổn định.
- **Dropout** (trong các mô hình học sâu): Ngẫu nhiên loại bỏ một số đơn vị trong mạng nơ-ron trong quá trình huấn luyện để ngăn chặn quá khớp.

2.1.8. Sử dụng các phương pháp Ensemble

- **Bagging (Bootstrap Aggregating):** Kết hợp nhiều mô hình dự đoán được huấn luyện trên các mẫu dữ liệu ngẫu nhiên khác nhau để giảm thiểu sai số.
- **Boosting**: Kết hợp nhiều mô hình đơn giản thành một mô hình mạnh mẽ bằng cách ưu tiên cải thiện các mẫu khó dự đoán.

Những phương pháp và kỹ thuật này có thể góp phần làm cho mô hình dự đoán rủi ro tín dụng không chỉ chính xác mà còn ổn định và bền vững theo thời gian, cũng như dễ dàng thích ứng với các thay đổi trong dữ liệu đầu vào và điều kiện thị trường.

2.2. Các mô hình khác Các mô hình được nhóm thực hiện

2.2.1. CatBoost

CatBoost là một thuật toán machine learning do Yandex phát triển, chuyên xử lý dữ liệu có chứa các biến phân loại. Đặc điểm nổi bật của CatBoost là khả năng xử lý hiệu quả các biến phân loại mà không cần kỹ thuật mã hóa như One-Hot Encoder. Sử dụng thuật toán symmetric weighted quantile sketch (SWQS), CatBoost tự động xử lý giá trị thiếu, giảm thiểu overfitting và cải thiện hiệu suất tổng thể.
Những điểm nổi bật của CatBoost:

- Xử lý đặc tính phân loại tích hợp, không cần tiền xử lý.
- Cung cấp kết quả xuất sắc mà không cần tinh chỉnh tham số phức tạp.
- Tích hợp sẵn xử lý giá trị khuyết.
- Tự động chuẩn hóa đặc tính.
- Chống overfitting mạnh mẽ.
- Tích hợp kiểm định chéo để chọn siêu tham số tốt nhất.
- Phiên bản GPU nhanh và mở rộng, phù hợp với các tập dữ liệu lớn.

2. LightGBM boosting_type - gbdt

LightGBM khi sử dụng boosting_type được gọi là "gbdt" (Gradient Boosting Decision Tree). Phương pháp gbdt (Gradient Boosted Decision Trees) là phương pháp Gradient Boosting truyền thống và là thuật toán đằng sau một số thư viện nổi tiếng như XGBoost và pGBRT.

3. LightGBM boosting_type - goss

Một biến thể khác của LightGBM khi sử dụng boosting_type là goss (Gradient-based One-Side Sampling). GOSS là một triển khai mới và nhẹ hơn của gbdt.
gbdt tiêu chuẩn đáng tin cậy nhưng không đủ nhanh trên các tập dữ liệu lớn. Do đó, GOSS đề xuất một phương pháp lấy mẫu dựa trên gradient để tránh phải tìm kiếm toàn bộ không gian tìm kiếm. Phương pháp này giữ lại các điểm dữ liệu có gradient lớn và thực hiện lấy mẫu ngẫu nhiên trên các điểm có gradient nhỏ, làm giảm không gian tìm kiếm và giúp hội tụ nhanh hơn. GOSS tập trung vào việc chọn các mẫu có gradient lớn để huấn luyện, cải thiện hiệu suất mà không làm giảm độ chính xác của mô hình, giúp LightGBM trở nên hiệu quả hơn trong việc xử lý các tập dữ liệu lớn.

4. Kết hợp hai mô hình “ggbd” và “goss”

Nhóm mong muốn việc kết hợp hai mô hình LightGBM boosting_type là "gbdt" và "goss" có thể mang lại lợi ích từ cả hai phương pháp. Sự kết hợp này tận dụng được

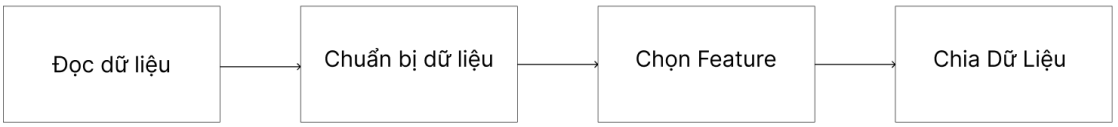
khả năng học tập từ các mẫu quan trọng của GOSS và sự ổn định của gbdt, giúp cải thiện hiệu suất và độ chính xác của mô hình dự đoán .

5. Mô hình kết hợp CatBoost với 2 mô hình "ggbd" và "goss"

Mô hình kết hợp CatBoost với hai biến thể của LightGBM (gbdt và GOSS) tạo ra một hệ thống ensemble mạnh mẽ. Việc kết hợp các mô hình này có thể tận dụng được điểm mạnh của từng mô hình riêng lẻ, từ khả năng xử lý biến phân loại hiệu quả của CatBoost đến tốc độ và hiệu quả của LightGBM. Kỹ thuật ensemble này giúp nâng cao độ chính xác và tính ổn định của các dự đoán .

3. Phương pháp của nhóm

3.1 Tiền xử lý dữ liệu:



Bộ dữ liệu Home Credit trên Kaggle chứa thông tin về hồ sơ vay vốn của khách hàng, được sử dụng cho cuộc thi phân loại rủi ro tín dụng. Tuy nhiên, dữ liệu thô này cần được tiền xử lý trước khi sử dụng cho mô hình học máy từ đó có thể nâng cao hiệu suất mô hình và khả năng tổng quát hoá.

3.1.1 Đọc dữ liệu

- Sử dụng polars để đọc file parquet thay vì pandas vì hiệu suất cao hơn.
- Chuyển đổi dữ liệu từ polars sang pandas để sử dụng các thư viện học máy trong Python.

3.1.2 Chuẩn bị dữ liệu:

Loại bỏ cột:

- Xác định loại biến cho mỗi cột dựa trên định nghĩa (P, M, A, D, T, L).
- Các biến được biến đổi theo các nhóm tương tự như sau:

● **P - Transform DPD (Days past due):** Biến đổi liên quan đến số ngày quá hạn thanh toán.

● **M - Masking categories:** Biến đổi liên quan đến việc ẩn các danh mục hoặc nhóm.

● **A - Transform amount:** Biến đổi liên quan đến các số tiền.

● **D - Transform date:** Biến đổi liên quan đến ngày tháng.

● **T - Unspecified Transform:** Biến đổi không được xác định rõ ràng.

- **L - Unspecified Transform:** Biến đổi không được xác định rõ ràng.
- Loại bỏ các cột có hơn 200 giá trị khác nhau để tránh overfitting.

Nối dữ liệu:

- **Đối với các cột thuộc loại depth 1, 2:**
  - Nối dữ liệu depth 1, 2 với dữ liệu gốc (depth 0) theo case_id.
  - Cập nhật tên cột cho phù hợp với độ sâu (ví dụ: max_P_FLAG_1, max_P_FLAG_2)

3.1.3 Chọn features:

Xác định giá trị thiếu (NaN) trong các đặc trưng số:

- Sử dụng hàm isna() của Pandas để tạo một DataFrame nans_df nhằm xác định vị trí các giá trị thiếu (NaN) trong các đặc trưng số (nums) của tập dữ liệu huấn luyện (df_train).

Phân nhóm đặc trưng theo số lượng giá trị thiếu:

- Mục đích: Nhóm các đặc trưng có cùng số lượng giá trị thiếu để giảm thiểu độ trùng lặp.
- Cách thực hiện:
  - Duyệt qua từng đặc trưng số (col) trong nums.
  - Tính toán tổng số lượng giá trị thiếu (cur_group) cho đặc trưng hiện tại (col) bằng cách sử dụng cột tương ứng trong nans_df.
  - Sử dụng từ điển nans_groups để lưu trữ các nhóm này.
    - Nếu một nhóm với cùng số lượng giá trị thiếu (cur_group) đã tồn tại, đặc trưng hiện tại (col) được thêm vào nhóm đó.
    - Ngược lại, tạo một nhóm mới với đặc trưng hiện tại (col) là phần tử đầu tiên.

Chọn đặc trưng trong nhóm :

- Mục đích: Chọn một đặc trưng duy nhất từ mỗi nhóm được xác định dựa trên số lượng giá trị thiếu.
- Cách thực hiện:
  - Định nghĩa hàm reduce_group với mục đích chọn một đặc trưng duy nhất từ mỗi nhóm.
  - Hàm này nhận danh sách các nhóm (grps) làm đối số (giả sử là kết quả của logic nhóm được bình luận ở bước 2).
  - Duyệt qua từng nhóm (g) trong grps.
    - Theo dõi đặc trưng có số lượng giá trị duy nhất tối đa (mx) và tên đặc trưng tương ứng (vx).
    - So sánh số lượng giá trị duy nhất (n) của mỗi đặc trưng (gg) trong nhóm với giá trị tối đa hiện tại (mx).

            - Nếu một đặc trưng có nhiều giá trị duy nhất hơn giá trị tối đa hiện tại, cập nhật mx và vx.
        - Hàm trả về danh sách use chứa các đặc trưng được chọn, mỗi nhóm được đại diện bởi đặc trưng có nhiều giá trị duy nhất nhất.

Chọn đặc trưng theo độ tương quan :

- Mục đích: Nhóm các đặc trưng có độ tương quan cao để loại bỏ sự trùng lặp.
- Cách thực hiện (đã bình luận):
    - Định nghĩa hàm group_columns_by_correlation với mục đích nhóm các đặc trưng có độ tương quan cao.
    - Hàm này nhận ma trận tương quan (matrix) và ngưỡng tương quan (threshold) làm đối số.
    - Khởi tạo danh sách rỗng groups để lưu trữ các nhóm đặc trưng được xác định.
    - Khởi tạo danh sách remaining_cols để theo dõi các đặc trưng chưa được gán vào nhóm nào (ban đầu chứa tất cả các đặc trưng).
    - Lặp lại cho đến khi không còn đặc trưng nào trong remaining_cols:
        - Lấy đặc trưng đầu tiên (col) từ remaining_cols.
        - Tạo một nhóm (group) chứa đặc trưng hiện tại (col).
        - Tạo danh sách correlated_cols để lưu trữ các đặc trưng tương quan.
        - Duyệt qua các đặc trưng còn lại (c) trong remaining_cols.
            - Tính toán hệ số tương quan giữa đặc trưng hiện tại (col) và đặc trưng còn lại (c) sử dụng ma trận tương quan.
            - Nếu hệ số tương quan lớn hơn hoặc bằng ngưỡng (threshold), thêm cả đặc trưng hiện tại (col) và đặc trưng còn lại (c) vào danh sách tương ứng (group và correlated_cols).
        - Sau khi duyệt qua tất cả các đặc trưng còn lại, thêm nhóm hiện tại (group) vào danh sách groups.
        - Cập nhật remaining_cols bằng cách xóa các đặc trưng trong correlated_cols (đã được gán vào nhóm).
- Hàm này về cơ bản xác định các nhóm đặc trưng có độ tương quan cao với nhau.

Chọn đặc trưng cuối cùng:

Khởi tạo danh sách rỗng uses để lưu trữ tập các đặc trưng được chọn cuối cùng.

Duyệt qua từng cặp khóa-giá trị (k, v) trong từ điển nans_groups.

- Nếu một nhóm (v) chứa nhiều hơn một đặc trưng (chỉ ra sự trùng lặp tiềm ẩn do thiếu giá trị), tiến hành chọn lọc thêm:
    - Lấy các đặc trưng của nhóm (Vs) từ nans_groups[k].
    - Tính toán ma trận tương quan cho các đặc trưng nhóm này bằng cách sử dụng một tập con DataFrame (df_train[Vs]).
    - Gọi hàm group_columns_by_correlation (giả sử hàm này được bỏ chú thích) để xác định các nhóm đặc trưng tương quan bên trong nhóm cụ thể này (Vs).

    - Gọi hàm reduce_group (giả sử hàm này được bỏ chú thích) để chọn một đặc trưng duy nhất từ mỗi nhóm tương quan được xác định.
    - Các đặc trưng được chọn từ các nhóm này được thêm vào danh sách uses.
- Nếu một nhóm (v) chỉ chứa một đặc trưng, đơn giản là thêm đặc trưng đó vào danh sách uses, coi như nó có liên quan.

Cuối cùng, đoạn mã chọn lọc lại dữ liệu huấn luyện (df_train), chỉ giữ lại các cột (đặc trưng) có tên xuất hiện trong danh sách uses.

3.1.4 Chia dữ liệu

StratifiedGroupKFold:

- Chia dữ liệu thành n_splits phần (folds) theo cách bảo toàn tỷ lệ lớp (stratification) trong mỗi fold.
- Giữ nguyên thứ tự dữ liệu trong mỗi nhóm được xác định bởi một cột được chỉ định (ví dụ: weeks trong trường hợp này).
- Đảm bảo rằng mỗi fold có sự phân bố lớp mục tiêu (target) tương tự như tập dữ liệu gốc.

Cross-validation:

- Sử dụng bộ chia dữ liệu cv để lặp lại quá trình chia dữ liệu thành tập huấn luyện và tập kiểm tra.
- Trong mỗi lần lặp:
    - Huấn luyện mô hình học máy trên tập huấn luyện.
    - Đánh giá hiệu suất mô hình trên tập kiểm tra.
    - Ghi lại kết quả đánh giá (ví dụ: độ chính xác, độ f1, v.v.).
- Tính toán hiệu suất trung bình của mô hình trên tất cả các lần lặp.

3.2 Phương pháp huấn luyện mô hình

3.2.1 Cấu hình mô hình:

- **CatBoostClassifier:**
    - Chọn CatBoost vì khả năng xử lý tốt với dữ liệu dạng bảng và tự động xử lý các giá trị thiếu.
- **LGBMClassifier:**
    - Cấu hình hai phiên bản của LightGBM:
        - params_lgb sử dụng boosting type "gbdt".
        - params_lgb2 sử dụng boosting type "goss".
    - Các tham số như: objective, metric, max_depth, learning_rate, n_estimators, colsample_bytree, colsample_bynode, reg_alpha, reg_lambda, extra_trees, num_leaves, device.

3.2.2. Quy trình huấn luyện:

- Sử dụng StratifiedGroupKFold để chia dữ liệu thành các fold.
- Trong mỗi fold:
  - **CatBoostClassifier:**
    - Tạo và huấn luyện mô hình trên tập huấn luyện.
    - Đánh giá mô hình trên tập kiểm tra và ghi lại điểm AUC.
    - Lưu trữ mô hình và điểm số.
  - **LGBMClassifier:**
    - Tương tự như CatBoost, huấn luyện và đánh giá hai phiên bản của LightGBM.
    - Lưu trữ mô hình và điểm số.
  - **VotingClassifier:**
    - Kết hợp các mô hình đã huấn luyện bằng cách sử dụng Voting Classifier với phương pháp "soft" voting.
    - Đánh giá hiệu suất của mô hình tổng hợp và ghi lại điểm số.

4. Phân tích và thảo luận các kết quả

4.1. Phương pháp chính và phương pháp baseline

Mô hình	Private score	Accuracy	Run
Mô hình chính	0.519	0.98	1997.3s
Baseline	0.244	0.75	156.1s

- Phương pháp baseline áp dụng mô hình LGB với boosting_type là "gbdt" truyền thống và một phần nhỏ dữ liệu, cho nên không thể học được đa dạng đặc trưng.
- Mô hình chính của nhóm là mô hình ensemble kết hợp LGB boosting_type là "gbdt" và LGB boosting_type là "goss" giúp giảm kích thước dữ liệu và tăng tốc độ huấn luyện mà không làm giảm đáng kể độ chính xác của mô hình và đồng thời áp dụng CatBoost. Do đó, có hiệu suất tốt hơn nhờ khả năng tận dụng những ưu điểm riêng biệt của từng thuật toán. Sự kết hợp này giúp giảm thiểu các lỗi do từng mô hình đơn lẻ gây ra, dẫn đến độ chính xác cao hơn và tính tổng quát tốt hơn.
- So sánh file submission.csv
  Lấy chuẩn là kết quả của mô hình chính có áp dụng Metric Hack (*vì theo như kết quả EDA ta thấy target = 0 chiếm đa số thế nên dư đoán nhiều kết quả là 0 chính xác sẽ cho ra hiệu suất tốt hơn*)

case_id	score
57543	0.0
57549	0.0
57551	0.0
57552	0.0
57569	0.0
57630	0.0
57631	0.0
57632	0.0
57633	0.0
57634	0.007544061574921726

Mô hình chính	Baseline
<table><tr><th>case_id</th><th>score</th></tr><tr><td>57543</td><td>0.011412127116090074</td></tr><tr><td>57549</td><td>0.01818347648440663</td></tr><tr><td>57551</td><td>0.011251555338995968</td></tr><tr><td>57552</td><td>0.03700805663896306</td></tr><tr><td>57569</td><td>0.03131222058807231</td></tr><tr><td>57630</td><td>0.01947298998674398</td></tr><tr><td>57631</td><td>0.04310656110665041</td></tr><tr><td>57632</td><td>0.04796632967918856</td></tr><tr><td>57633</td><td>0.019051258786158452</td></tr><tr><td>57634</td><td>0.05777516222381707</td></tr></table>	<table><tr><th>case_id</th><th>score</th></tr><tr><td>57543</td><td>0.05358334772222262</td></tr><tr><td>57549</td><td>0.05431428699580822</td></tr><tr><td>57551</td><td>0.041261482652024326</td></tr><tr><td>57552</td><td>0.02999266959719471</td></tr><tr><td>57569</td><td>0.01195287660966685</td></tr><tr><td>57630</td><td>0.02673307945480869</td></tr><tr><td>57631</td><td>0.0113786495846606516</td></tr><tr><td>57632</td><td>0.004467965516136927</td></tr><tr><td>57633</td><td>0.0437225135894588</td></tr><tr><td>57634</td><td>0.03798860884815231</td></tr></table>
- Biên độ của score dự đoán dao động từ 0.01 đến trên 0.05 - So với chuẩn, phân bố score cao thấp khá tương đồng theo case_id	- Biên độ của score dự đoán dao động từ 0.01 đến trên 0.05 - So với chuẩn, phân bố score cao thấp không tương đồng theo case_id => dự đoán sai nhiều trường hợp, do không được tiếp xúc với nhiều đặc trưng

4.2. Kết quả một số phương pháp khác

4.2.1. Phương pháp Balanced RF Classifier và TargetEncoding

Mô hình	Private score	Accuracy	Run
Balanced RF Classifier và TargetEncoding	0.51507	0.735	755.3s

- Target Encoding: thay thế mỗi danh mục bằng giá trị trung bình của biến mục tiêu cho danh mục đó, hữu ích trong các trường hợp mà tập dữ liệu có các đặc trưng phân loại với số lượng danh mục cao.
- Balanced Random Forest khác biệt so với Random Forest cổ điển ở chỗ nó sẽ rút một mẫu bootstrap từ lớp thiểu số và lấy mẫu có hoàn lại cùng số lượng mẫu từ lớp đa số.

Nó vẫn giữ các ưu điểm vốn có của Random Forest, chẳng hạn như tính bền vững trước overfitting, khả năng xử lý dữ liệu có số chiều cao, và cung cấp các điểm quan trọng của đặc trưng.

- File submission.csv

case_id	score
57543	0.4693333333333333
57549	0.48266666666666663
57551	0.4103333333333333
57552	0.487
57569	0.47400000000000003
57630	0.38966666666666666
57631	0.526
57632	0.45933333333333337
57633	0.471
57634	0.5446666666666666

- Ta thấy, biên độ của score khá nhỏ từ khoảng 0.038 đến hơn 0.05
- Score không thiên hướng phân loại theo một nhãn nào [0, 1]

=> Kém hiệu quả hơn so với thuật toán boosting khi sử dụng trên các bộ dữ liệu lớn hoặc có nhiều đặc điểm phức tạp.

4.2.2. Phương pháp Linear Regression with VIF predictor reduction and outlier reduction

- Linear Regression là một mô hình đơn giản và tuyến tính, thích hợp cho các bộ dữ liệu có mối quan hệ tuyến tính rõ ràng giữa các biến độc lập và biến phụ thuộc.
- Việc sử dụng VIF để giảm số lượng predictor giúp loại bỏ các biến đồng biến cao, cải thiện độ chính xác và độ ổn định của mô hình tuyến tính.
- Outlier reduction giúp loại bỏ các điểm dữ liệu bất thường, cải thiện hiệu suất của mô hình tuyến tính. Tuy nhiên, Linear Regression vẫn có thể không xử lý tốt các dữ liệu phức tạp và phi tuyến tính.

Mô hình	Private score	Accuracy	Run
Linear Regression with VIF predictor reduction and outlier reduction	0.30977	0.354	7228.8s

- File submission.csv

case_id	score
57543	0.037612898647785185
57549	0.09399235248565674
57551	0.021912098862230778
57552	0.04265385568141937
57569	0.10466689467430115
57630	0.026651515066623686
57631	0.049517746269702914
57632	0.038853201642632486
57633	0.03161150738596916
57634	0.05010153278708458

- Biên độ của score lớn, từ 0.02 đến 0.1
- So với chuẩn, ở những case_id cần score thấp hoặc bằng 0 thì lại có dự đoán quá cao (Ví dụ: case_id 57569 có score hơn 0.1)

=> Nhiều dự đoán sai dẫn đến hiệu suất thấp.

- Linear Regression thích hợp với dữ liệu ít phức tạp, do đó không hoạt động tốt với bộ dữ liệu của cuộc thi.

4.3. Thực nghiệm liên quan đến mô hình chính

- Ablation study

Mô hình	Accuracy	Nhận xét
CatBoost	0.762	Hiệu suất tốt, xử lý tốt các biến phân loại và dữ liệu mất cân đối.
LGB boosting_type là "gbdt"	0.765	Hiệu suất cao, sử dụng Gradient Boosting Decision Tree.
LGB boosting_type là "goss"	0.766	Hiệu suất cao, sử dụng Gradient-based One-Side Sampling, cải thiện hơn so với gbdt.
Kết hợp hai mô hình LGB trên	0.725	Hiệu suất giảm khi kết hợp, có thể do sự không đồng nhất trong các boosting types.
Mô hình CatBoost + hai mô hình LGB	0.98	Hiệu suất rất cao, tận dụng được ưu điểm của cả CatBoost và LGB, mô hình ensemble mạnh mẽ vì có tính tổng quát cao và học được nhiều khía cạnh khác nhau từ dữ liệu.

- Thay đổi learning rate

Learning rate	Accuracy	Private score	Nhận xét
0.5	0.99	0.45	Hiệu suất cao trên tập train, nhưng overfitting cao, hiệu suất trên tập test thấp.
0.05	0.98	0.519	Hiệu suất tốt cả trên tập train và test, learning rate hợp lý, cân bằng giữa huấn luyện và tổng quát.
0.005	0.9	Timeout Error	Learning rate quá thấp, dẫn đến thời gian huấn luyện quá lâu, không khả thi trong thực tế.

5. Kết luận, hướng phát triển tương lai và bảng phân công

Nhóm đã tham gia cuộc thi "Home Credit - Credit Risk Model Stability" trên Kaggle và đạt điểm số khá ổn (0.51926), xếp hạng 134/3858 đội tham gia. Dự án này giúp nhóm nắm rõ được quy trình xây dựng một dự án thực tế, từ tiền xử lý, thăm dò dữ liệu, xây dựng mô hình rồi tinh chỉnh tham số. Thực nghiệm và kết quả được trình bày trong phần 4 của báo cáo cũng giúp nhóm hiểu rõ hơn về ưu nhược điểm các mô hình (CatBoost, gbdt, goss, …), phương pháp (VIF, TargetEncoding,...), tham số (Learning

rate,) đã sử dụng. Việc còn hơn 100 nhóm xếp hạng cao hơn cho thấy bài toán này vẫn còn hướng giải quyết tốt hơn, cần không ngừng nghiên cứu và tìm tòi.

Qua cuộc thi, nhóm đã học hỏi được nhiều kinh nghiệm quý báu về việc tiến hành 1 dự án trên dữ liệu thực tế. Dự án giúp nhóm nâng cao kiến thức và kỹ năng chuyên môn, tiếp xúc với nguồn dữ liệu thực tế, trau dồi thêm kỹ năng làm việc nhóm và tư duy giải quyết vấn đề và mở rộng network.

Dựa trên những kinh nghiệm và bài học rút ra từ cuộc thi, nhóm chúng tôi đề xuất một số hướng phát triển tương lai cho dự án này:

- Tối ưu hóa và mở rộng mô hình hiện tại:
  + Tối ưu hóa hiệu suất bằng kỹ thuật hyperparameter tuning.
  + Phát triển hệ thống học tự động và liên tục (autoML).
- Nghiên cứu và phát triển phương pháp tiên tiến: Tích hợp trí tuệ nhân tạo tiên tiến như học sâu và Explainable AI.
- Thu thập thêm dữ liệu đa dạng, chất lượng cao và hữu ích cho dự đoán từ nhiều nguồn.
- Áp dụng kiến thức vào thực tế và lĩnh vực khác:
  + Đa dạng hóa ứng dụng trong bảo hiểm, chăm sóc sức khỏe, và thị trường chứng khoán.
  + Phân tích hành vi khách hàng để tối ưu hóa chiến lược kinh doanh.
- Phát triển hệ thống và công cụ hỗ trợ: Tích hợp hệ thống giám sát hiệu suất mô hình và tự động hóa quy trình xử lý dữ liệu.
- Phát triển chiến lược dài hạn và nghiên cứu thị trường để điều chỉnh giải pháp dự đoán rủi ro phù hợp.

Tham gia cuộc thi "Home Credit - Credit Risk Model Stability" là một trải nghiệm vô cùng quý giá, giúp các thành viên trong nhóm trưởng thành và phát triển bản thân một cách vượt bậc. Với tinh thần ham học hỏi, không ngừng sáng tạo và cống hiến, chúng em tin tưởng bản thân sẽ đạt được những thành công to lớn hơn nữa trong tương lai.

Thành viên	MSSV	Công việc thực hiện	Đánh giá
Trần Tuyết Minh	21521144	- Nghiên cứu và thực hiện phương pháp giải quyết bài toán - Viết báo cáo phần 3	10/10
Lê Thị Như Ý	21522818	- Nghiên cứu và thực hiện phương pháp giải quyết bài toán - Viết báo cáo phần 4 - Làm slide	10/10
Nguyễn Thị Mai Trinh	21522718	- Nghiên cứu, xây dựng, thực thi quy trình dự đoán - Viết báo cáo phần 2 và 5 - Thuyết trình	10/10
Nguyễn Thị Huyền Trang	21520488	- Nghiên cứu, xây dựng, thực thi quy trình dự đoán	10/10

		- Viết báo cáo phần 1 - Thuyết trình	

Bảng phân công công việc

6. Tài liệu liên quan

[1] Feature Selection Using Principal Component Analysis | IEEE Conference Publication | IEEE Xplore

[2] 1706.09516 (arxiv.org)

[3] LightGBM: A Highly Efficient Gradient Boosting Decision Tree (neurips.cc)

[4] Keras: The Python Deep Learning library - NASA/ADS (harvard.edu)

[5] Metalearning: a survey of trends and technologies | Artificial Intelligence Review (springer.com)

[6] sách: Bishop, C. M. (2006). "Pattern Recognition and Machine Learning."

[7]【Machine Learning】Vietnamese - Framework LightGBM (Updating) #Python3 - Qiita

[8] The structure of LightGBM model (LGB). LGB is a kind of implementation... | Download Scientific Diagram (researchgate.net)

[9] CatBoost in Machine Learning - GeeksforGeeks

[10] A Complete Guide to Credit Risk Modelling (listendata.com)

[11] Prokhorenkova, L., et al. (2018). "CatBoost: unbiased boosting with categorical features." Advances in neural information processing systems.

[12] Ke, G., et al. (2017). "LightGBM: A highly efficient gradient boosting decision tree." Advances in neural information processing systems.

[13] Ke, G., et al. (2017). "A scalable tree boosting system." Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[14] "Gradient-based One-Side Sampling (GOSS) for LightGBM." Microsoft Documentation.

[15] Ensemble Learning with CatBoost and LightGBM. (https://towardsdatascience.com/ensemble-learning-using-catboost-and-lightgbm-27a062fbb11d)

[16] https://neptune.ai/blog/lightgbm-parameters-guide