

COMPANY BANKRUPTCY PREDICTION

D U C T R A N S P O N S T E R

NỘI DUNG

• • •

- I
- II
- III
- IV
- V
- VI
- VII
- VIII

Business understanding

Analytic approach

Data requirements

Data collection

Data understanding

Data preparation

Modelling & Evaluation

Conclusion



I. Business understanding (Hiểu biết về nghiệp vụ)

Trong kinh doanh, có thể coi phá sản là điều không mong muốn xảy đến nhất với bất kỳ doanh nghiệp nào bởi ảnh hưởng của nó hết sức tiêu cực → Có một nhu cầu mong muốn biết trước được khả năng phá sản của doanh nghiệp.



Câu hỏi đặt ra: "Làm thế nào để dự đoán được khả năng phá sản của doanh nghiệp?"

II. Analytic approach (Hướng tiếp cận giải quyết)

- **Dạng bài toán:** nhãn đầu ra biểu diễn dưới dạng 1 - bankruptcy (phá sản) và 0 - not bankruptcy (không bị phá sản) → Cần sử dụng mô hình phân loại (classification model).
- **Phương pháp:** Stacking ([CatBoost, Light GBM, MLP] | Logistic Regression)
- **Mô tả kỹ thuật:**
 - Ngôn ngữ lập trình: Python
 - Môi trường: Colab Notebook
 - Packages, framework: sklearn, imblearn, catboost, lightgbm, pandas, numpy, matplotlib, seaborn,...
- **Thang đo đánh giá:** F1-Score.
- **Mục tiêu của dự án:** Tìm ra một mô hình dự đoán được khả năng phá sản của công ty tốt nhất có thể, tối thiểu F1-Score Valid ≥ 0.75 .

III. Data requirements (Yêu cầu về dữ liệu)

Vì đây là bài toán dự đoán khả năng phá sản của công ty, do đó dữ liệu cần sử dụng là dữ liệu tài chính của các công ty, cùng kết quả có phá sản hay không của các công ty sau một thời gian hoạt động. Dữ liệu cần được tổ chức dưới dạng bảng, gồm các cột và các dòng.



IV. Data collection (Thu thập dữ liệu)

- **Nguồn dữ liệu:** Dữ liệu về dự đoán phá sản của các công ty Ba Lan được lấy từ cuộc thi Company Bankruptcy Prediction - tổ chức bởi The ISODS trên nền tảng Kaggle.
- **Dữ liệu gồm 2 file:**
 - File train.csv: dùng để xây dựng và huấn luyện mô hình.
 - File test.csv: dùng để đánh giá mô hình, kết quả đánh giá trên tập test là cơ sở để xếp hạng trong cuộc thi.

V. Data understanding (Thấu hiểu dữ liệu)

5.1. Tổng quan về dữ liệu

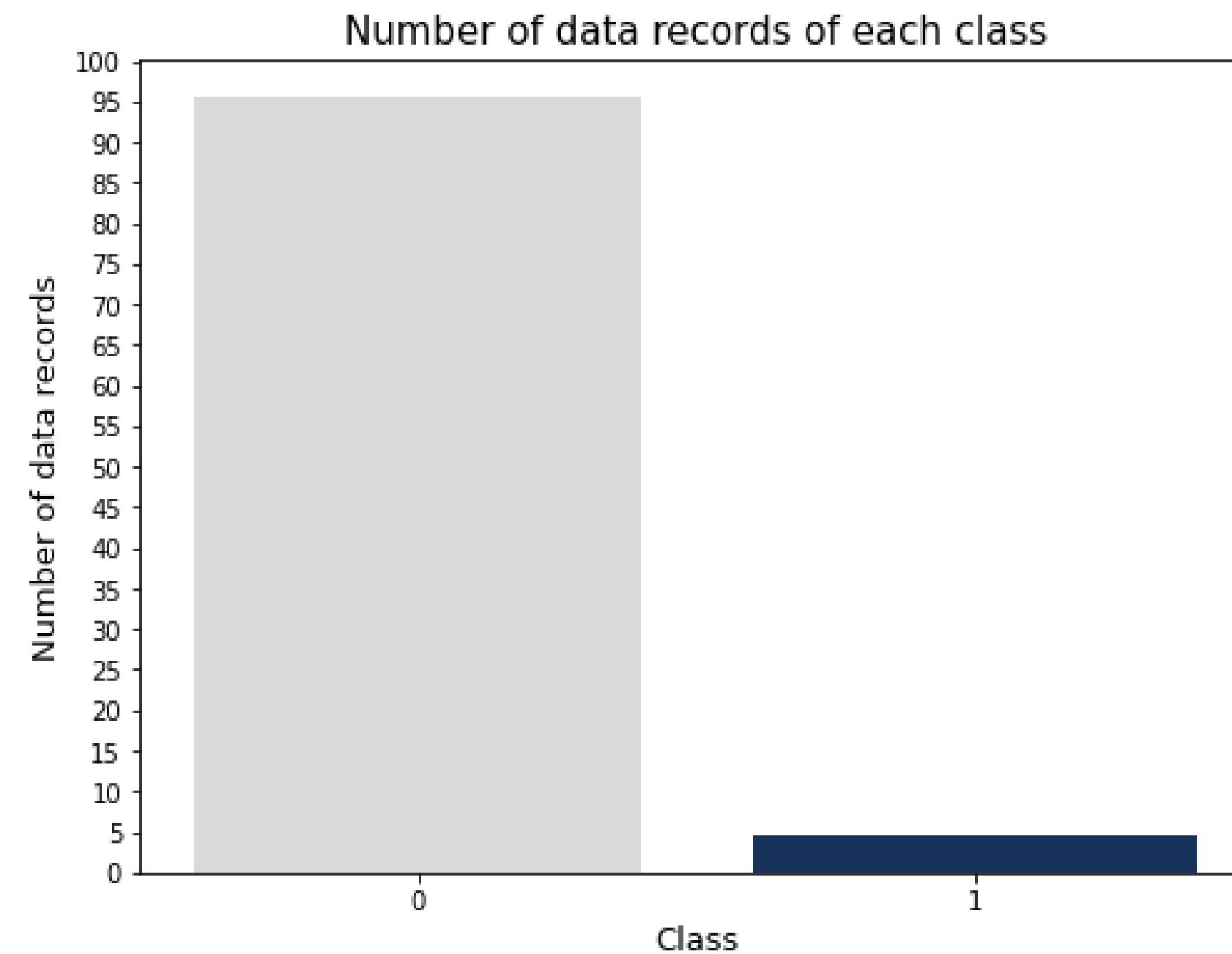
- **Số lượng dữ liệu:**
 - File train.csv: 67 trường (cột) và 25121 bản ghi (dòng).
 - File test.csv: 66 trường (không bao gồm class) và 12374 bản ghi.
- **Giá trị null:** tồn tại ở dạng `?` → Chuyển về `NaN`.
- **Loại dữ liệu:** Chưa đúng định dạng → Chuyển về đúng loại dữ liệu.
- **Tên trường dữ liệu:** 2 trường cần đổi tên, 2 trường cần rút ngắn tên để dễ quan sát.
- **Biến phân loại:** 3 trường (bao gồm id, forecasting_period, class).
- **Biến dạng số:** 64 trường (dữ liệu tài chính).



V. Data understanding (Thấu hiểu dữ liệu)

5.2. EDA - Phân tích dữ liệu khám phá

Biến class - Biến mục tiêu



Nhận xét: Dữ liệu bị mất cân bằng, class=1 chỉ chiếm 4.48 %.

V. Data understanding (Thấu hiểu dữ liệu)

5.2. EDA - Phân tích dữ liệu khám phá

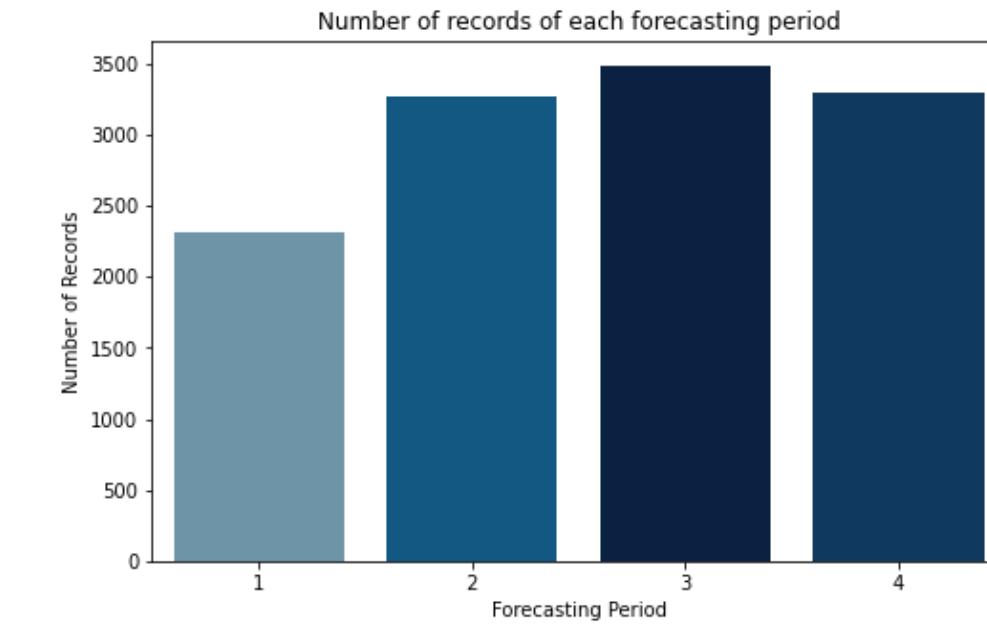
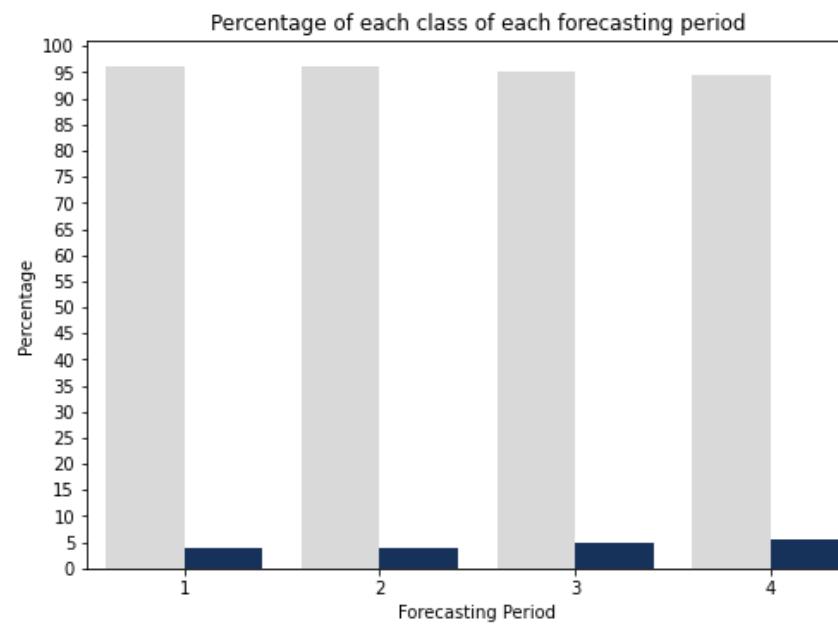
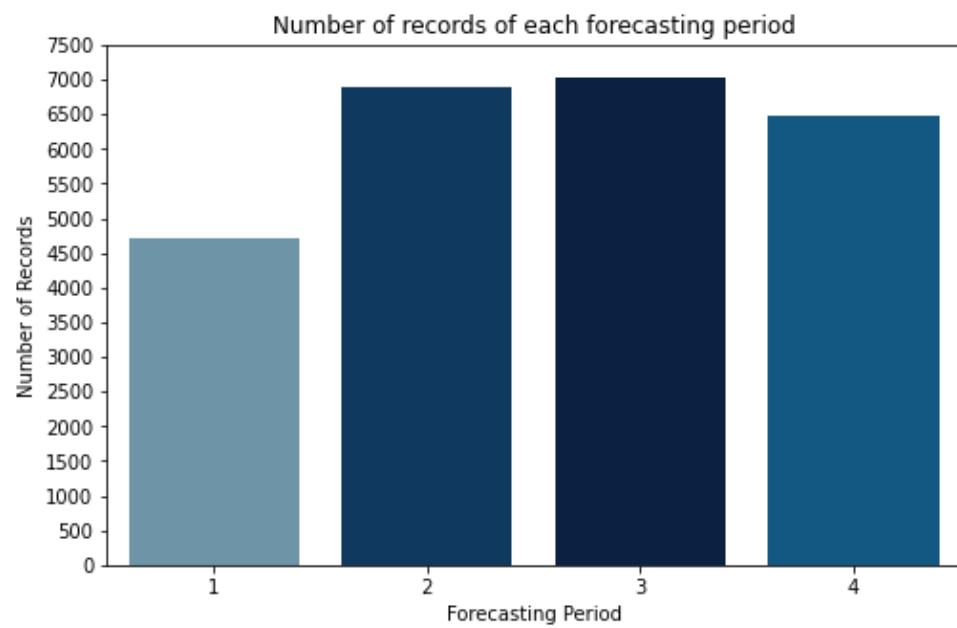
Biến forecasting_period

File train.csv

Forecasting Period	Number of Records	Non bankrupt	Bankrupt	Percentage of Non bankrupt	Percentage of Bankrupt
1	4712	4536	176	96.26486	3.73514
2	6900	6636	264	96.17391	3.82609
3	7018	6683	335	95.22656	4.77344
4	6491	6140	351	94.59251	5.40749

File test.csv

Forecasting Period	Number of Records
1	2315
2	3273
3	3485
4	3301



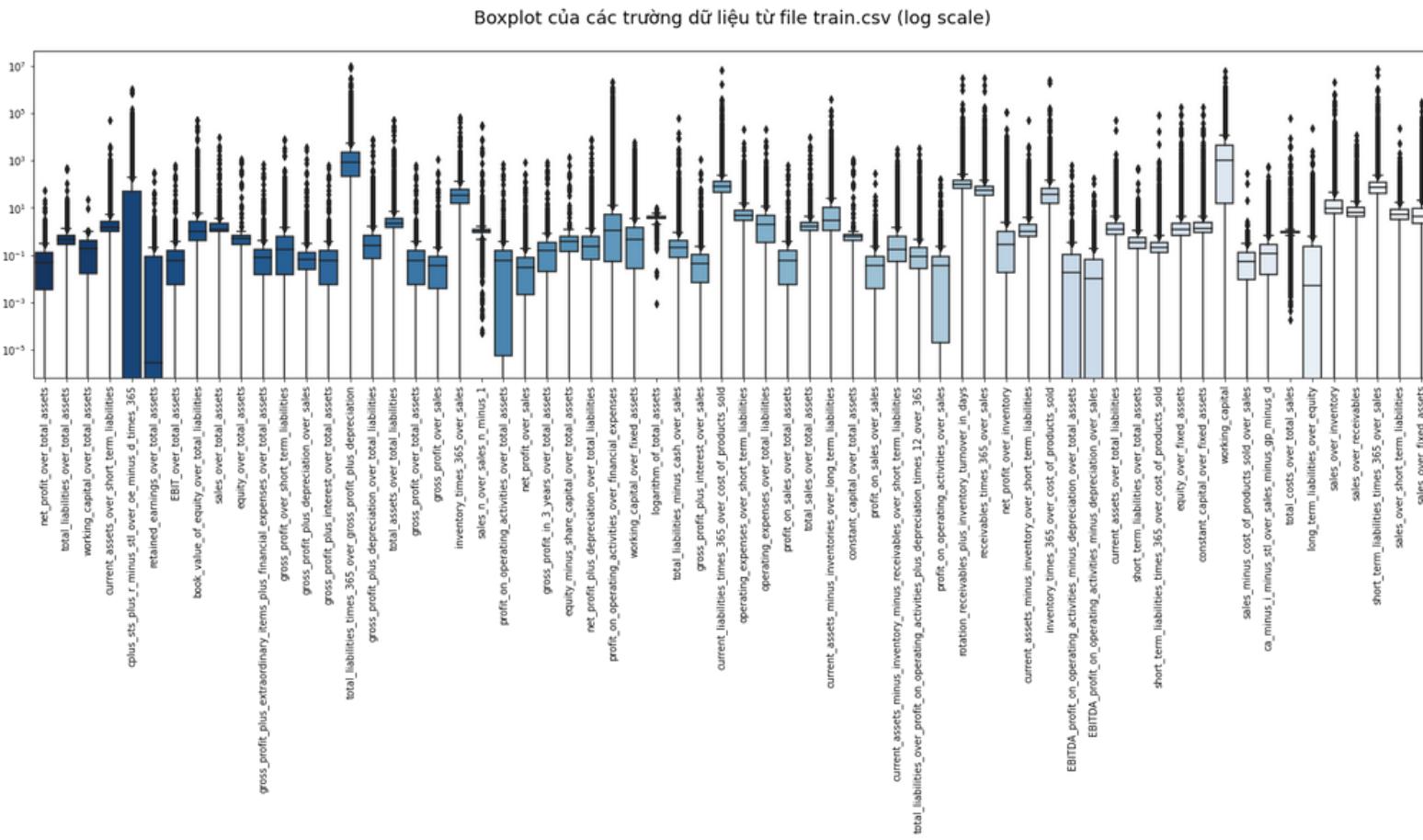
Hướng xử lý về sau: Sử dụng One-hot encoding.

V. Data understanding (Thấu hiểu dữ liệu)

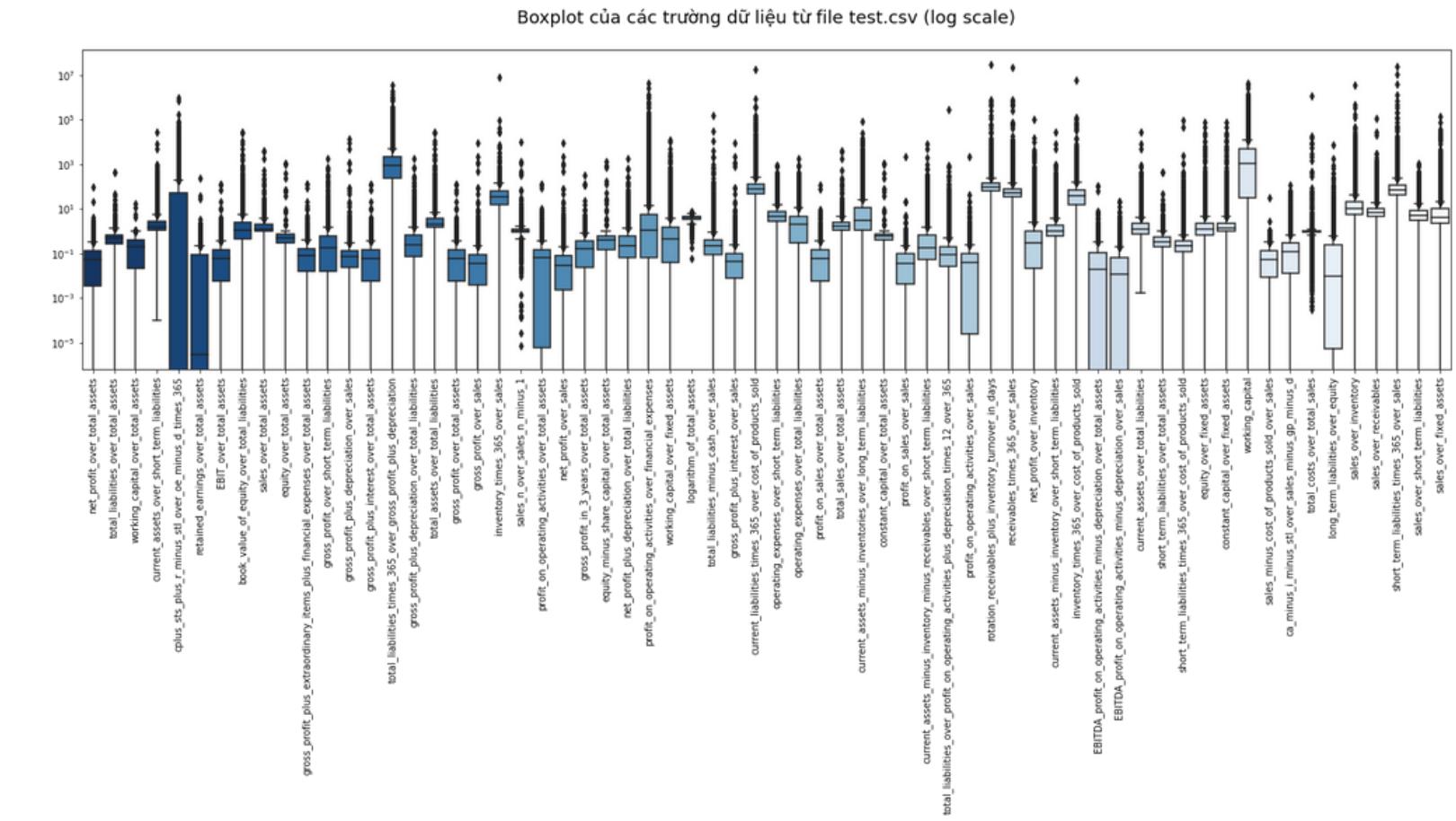
5.2. EDA - Phân tích dữ liệu khám phá

Boxplot

File train.csv



File test.csv



Nhận xét: Khoảng dữ liệu của các trường dữ liệu rất lớn và tồn tại nhiều giá trị ngoại lai (outliers).

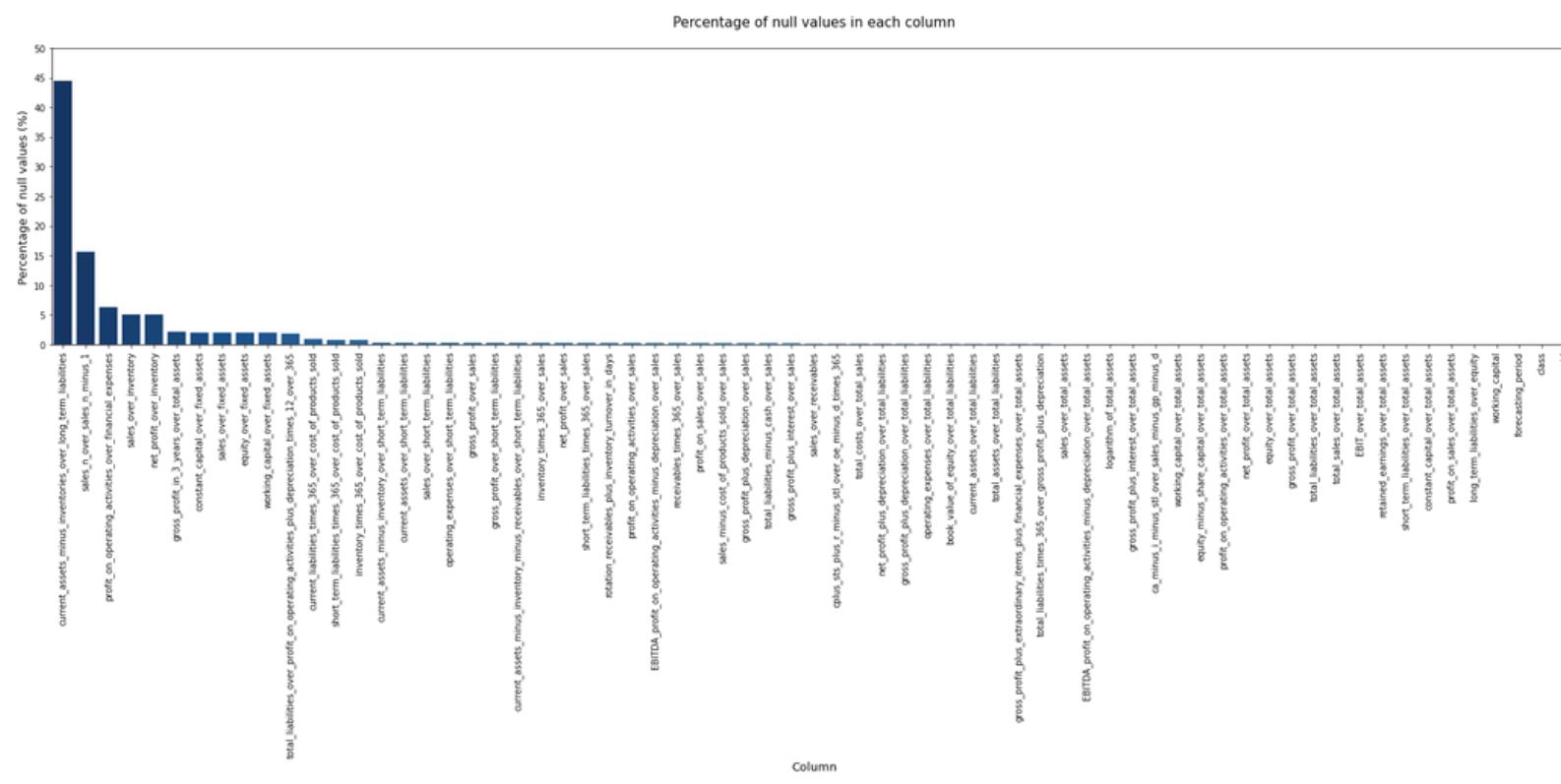
Hướng xử lý về sau: Chuyển giá trị ngoại lai về giới hạn trên và giới hạn dưới.

V. Data understanding (Thấu hiểu dữ liệu)

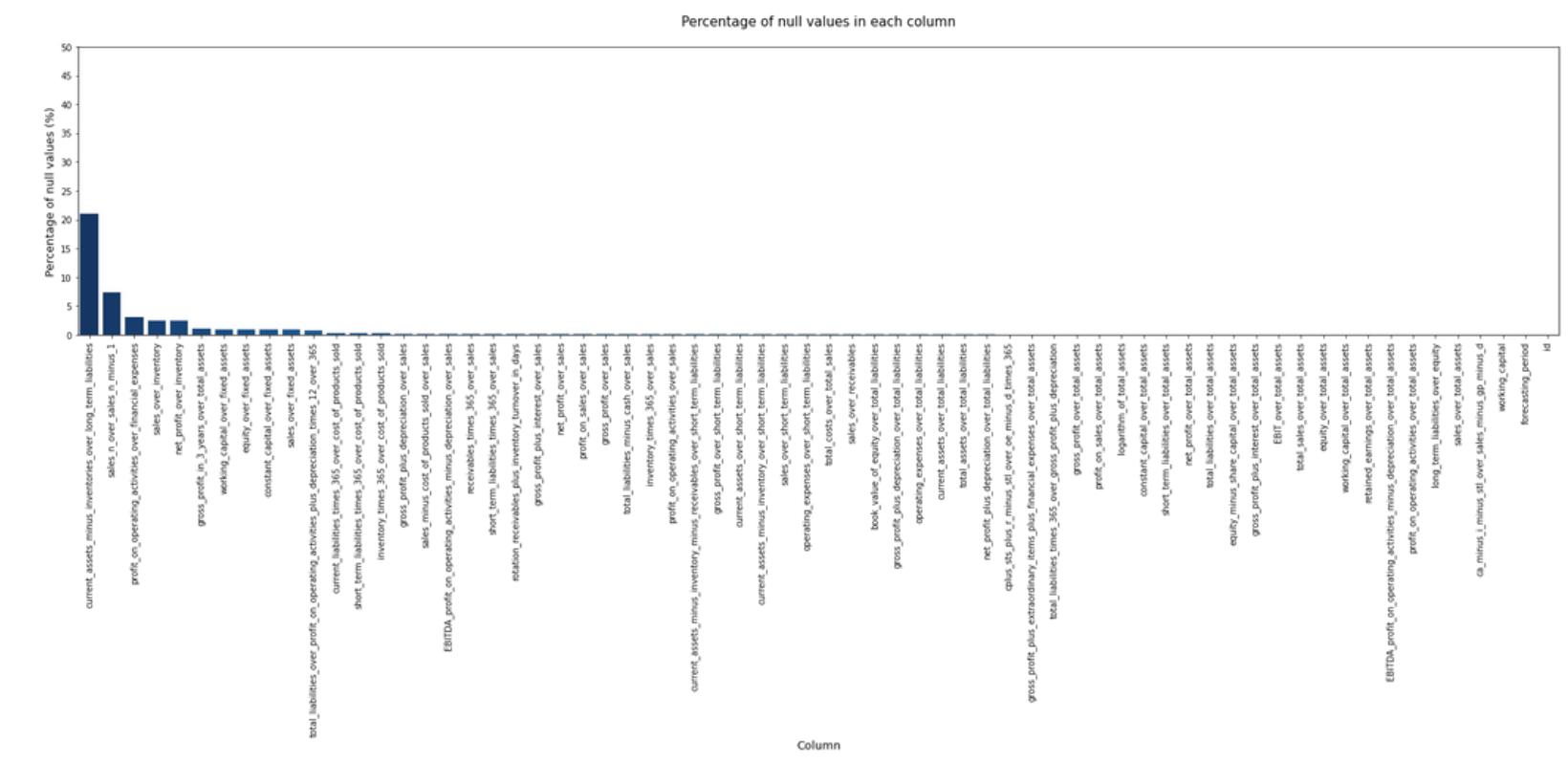
5.2. EDA - Phân tích dữ liệu khám phá

Kiểm tra giá trị null

File train.csv



File test.csv



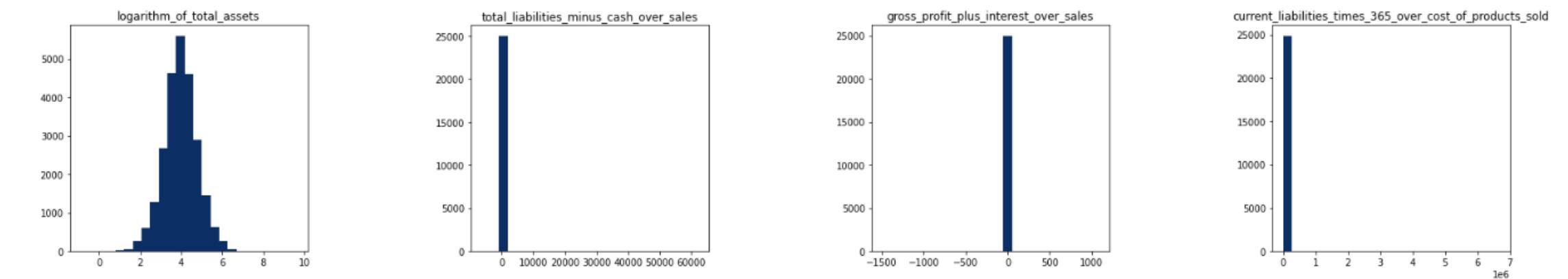
Nhận xét: Cột `current_assets_minus_inventories_over_long_term_liabilities` có giá trị null chiếm nhiều nhất lên tới 44.397% (ở file train.csv), 21.03% (ở file test.csv).

Hướng xử lý về sau: Loại bỏ cột nhiều null nhất, điền giá trị null bằng -9.

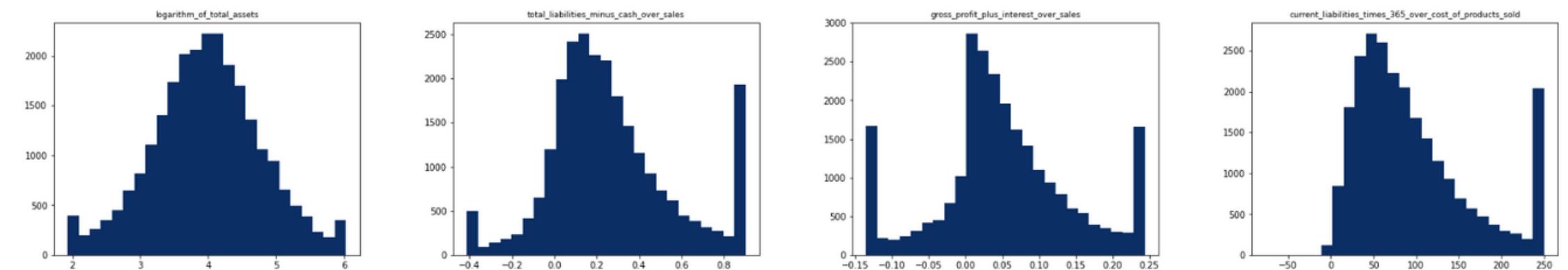
VI. Data preparation (Chuẩn bị dữ liệu)

6.1. Xử lý giá trị ngoại lai

Trước khi xử lý outliers



Sau khi xử lý outliers



Hướng xử lý về sau: Sử dụng StandardScaler để chuẩn hóa dữ liệu.



VI. Data preparation (Chuẩn bị dữ liệu)

6.2. Lựa chọn đặc trưng

Phương pháp lựa chọn đặc trưng:

- **Bước 1:** Loại bỏ cột nhiều null nhất và cột id.
- **Bước 2:** Xác định mức độ quan trọng của đặc trưng theo CatBoost.
- **Bước 3:** Đánh giá kết quả của các models CatBoost, Light GBM, MLP khi loại bỏ thêm lần lượt từng đặc trưng có mức độ ít quan trọng nhất sau mỗi lần lặp.
- **Bước 4:** Chọn ra số lượng đặc trưng tối ưu đem lại kết quả tốt nhất.

Kết quả: Sử dụng 32 đặc trưng có mức độ quan trọng tốt nhất để xây dựng models (giữ lại 1/2 số đặc trưng so với ban đầu).



VI. Data preparation (Chuẩn bị dữ liệu)

6.3. Chia tách dữ liệu

- Chia tách dữ liệu thành 2 tập dữ liệu train/validation với tỉ lệ 8/2.
- Tập dữ liệu train dùng để xây dựng model và tập validation dùng để kiểm định model.
- Thiết lập tỉ lệ nhãn class=1/class=0 của tập dữ liệu train và tập dữ liệu valid tương tự nhau.



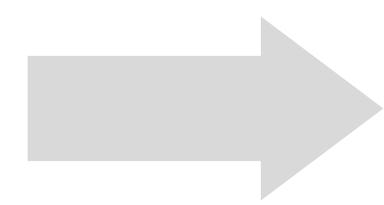
VI. Data preparation (Chuẩn bị dữ liệu)

6.4. Pipeline

Biến dạng số:

- Giá trị không null
- Giá trị có null

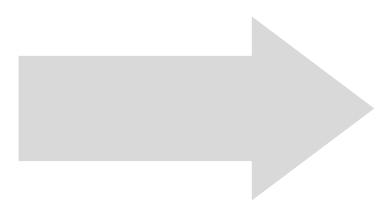
Biến phân loại



Biến dạng số:

- StandardScaler
- Đullen giá trị -9

Biến phân loại: One-hot encoding



X_train_transformed (20096, 34)

X_valid_transformed (5025, 34)

X_test_transformed (12374, 34)



VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.1. Mô hình ban đầu

Tạo các models với tham số thiết lập mặc định

```
CB_model = CatBoostClassifier(verbose=False, eval_metric='F1', random_state=0)
LGBM_model = lgb.LGBMClassifier(random_state=0)
MLP_model = MLPClassifier(random_state=0)
```

Kết quả

	model	f1_train	f1_valid	recall_train	recall_valid	precision_train	precision_valid	accuracy_train	accuracy_valid
0	MLPClassifier	0.949309	0.737113	0.914539	0.635556	0.986826	0.877301	0.995621	0.979701
1	CatBoostClassifier	0.936246	0.697143	0.880133	0.542222	1.000000	0.976000	0.994626	0.978905
2	LGBMClassifier	0.959584	0.658892	0.922309	0.502222	1.000000	0.957627	0.996517	0.976716
3	XGBClassifier	0.630468	0.549521	0.463929	0.382222	0.983529	0.977273	0.975617	0.971940
4	GradientBoostingClassifier	0.644825	0.527331	0.480577	0.364444	0.979638	0.953488	0.976264	0.970746
5	DecisionTreeClassifier	1.000000	0.506667	1.000000	0.506667	1.000000	0.506667	1.000000	0.955821
6	RandomForestClassifier	0.999445	0.416667	0.998890	0.266667	1.000000	0.952381	0.999950	0.966567
7	SVM	0.469992	0.412587	0.308546	0.262222	0.985816	0.967213	0.968800	0.966567
8	KNeighborsClassifier	0.515873	0.404040	0.360710	0.266667	0.905292	0.833333	0.969646	0.964776
9	AdaBoostClassifier	0.224779	0.226148	0.140954	0.142222	0.554585	0.551724	0.956409	0.956418
10	LogisticRegression	0.176633	0.159420	0.106548	0.097778	0.516129	0.431373	0.955464	0.953831

Nhận xét: 3 models tốt nhất lần lượt là MLP, CatBoost, Light GBM dựa trên F1-Score Valid.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Sử dụng SMOTE và Kết quả

CatBoost

	model	f1_train	f1_valid	recall_train	recall_valid	precision_train	precision_valid	accuracy_train	accuracy_valid
0	CatBoost - 0.6	0.979178	0.743455	0.965594	0.631111	0.993151	0.904459	0.998159	0.980498
1	CatBoost - 0.9	0.984975	0.730570	0.982242	0.626667	0.987723	0.875776	0.998656	0.979303
2	CatBoost - 0.3	0.971559	0.728261	0.947836	0.595556	0.996499	0.937063	0.997512	0.980100
3	CatBoost - 0.7	0.983221	0.727749	0.975583	0.617778	0.990981	0.885350	0.998507	0.979303
4	CatBoost - 0.8	0.985523	0.723514	0.982242	0.622222	0.988827	0.864198	0.998706	0.978706
5	CatBoost - 0.5	0.976325	0.720000	0.961154	0.600000	0.991982	0.900000	0.997910	0.979104
6	CatBoost - 1.0	0.983808	0.717557	0.977802	0.626667	0.989888	0.839286	0.998557	0.977910
7	CatBoost - 0.4	0.975085	0.717391	0.955605	0.586667	0.995376	0.923077	0.997811	0.979303
8	CatBoost - 0.1	0.951220	0.713483	0.908990	0.564444	0.997564	0.969466	0.995820	0.979701
9	CatBoost - 0.2	0.964490	0.700000	0.934517	0.560000	0.996450	0.933333	0.996915	0.978507

MLP

	model	f1_train	f1_valid	recall_train	recall_valid	precision_train	precision_valid	accuracy_train	accuracy_valid
0	MLP - 0.4	0.994444	0.709524	0.993341	0.662222	0.995551	0.764103	0.999502	0.975721
1	MLP - 0.6	0.996674	0.700696	0.997780	0.671111	0.995570	0.733010	0.999701	0.974328
2	MLP - 0.7	0.973499	0.700637	0.998890	0.733333	0.949367	0.670732	0.997562	0.971940
3	MLP - 0.3	0.975477	0.695260	0.993341	0.684444	0.958244	0.706422	0.997761	0.973134
4	MLP - 0.1	0.972527	0.692124	0.982242	0.644444	0.963003	0.747423	0.997512	0.974328
5	MLP - 0.8	0.994475	0.687225	0.998890	0.693333	0.990099	0.681223	0.999502	0.971741
6	MLP - 1.0	0.996678	0.683371	0.998890	0.666667	0.994475	0.700935	0.999701	0.972338
7	MLP - 0.5	0.991160	0.682578	0.995560	0.635556	0.986799	0.737113	0.999204	0.973532
8	MLP - 0.2	0.983916	0.679157	0.984462	0.644444	0.983370	0.717822	0.998557	0.972736
9	MLP - 0.9	0.992837	0.670968	1.000000	0.693333	0.985777	0.650000	0.999353	0.969552

Light GBM

	model	f1_train	f1_valid	recall_train	recall_valid	precision_train	precision_valid	accuracy_train	accuracy_valid
0	Light GBM - 0.5	0.879018	0.690909	0.834628	0.591111	0.928395	0.831250	0.989699	0.976318
1	Light GBM - 0.2	0.913947	0.683473	0.854606	0.542222	0.982143	0.924242	0.992785	0.977512
2	Light GBM - 0.3	0.886905	0.682927	0.826859	0.560000	0.956354	0.875000	0.990545	0.976716
3	Light GBM - 0.4	0.889941	0.680739	0.834628	0.573333	0.953105	0.837662	0.990744	0.975920
4	Light GBM - 0.7	0.877153	0.680101	0.847947	0.600000	0.908442	0.784884	0.989351	0.974726
5	Light GBM - 0.1	0.935294	0.674286	0.882353	0.524444	0.994994	0.944000	0.994526	0.977313
6	Light GBM - 0.8	0.870548	0.673367	0.854606	0.595556	0.887097	0.774566	0.988605	0.974129
7	Light GBM - 0.6	0.875651	0.664975	0.840178	0.582222	0.914251	0.775148	0.989301	0.973731
8	Light GBM - 0.9	0.871022	0.660241	0.865705	0.608889	0.876404	0.721053	0.988505	0.971940
9	Light GBM - 1.0	0.865137	0.650943	0.857936	0.613333	0.872460	0.693467	0.988008	0.970547



VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Sử dụng SMOTE và Kết quả

	model	f1_train	f1_valid	recall_train	recall_valid	precision_train	precision_valid	accuracy_train	accuracy_valid
CatBoost	0 CatBoost - 0.6	0.979178	0.743455	0.965594	0.631111	0.993151	0.904459	0.998159	0.980498

	model	f1_train	f1_valid	recall_train	recall_valid	precision_train	precision_valid	accuracy_train	accuracy_valid
Light GBM	0 Light GBM - 0.5	0.879018	0.690909	0.834628	0.591111	0.928395	0.831250	0.989699	0.976318

MLP Không cải thiện được

Kết quả:

- Sử dụng SMOTE đã giúp CatBoost và Light GBM cải thiện được kết quả, còn MLP thì không.
- Tuy vậy, F1-Score Valid vẫn chưa đạt được mục tiêu đề ra.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Tinh chỉnh siêu tham số và Kết quả

(1) Grid Search

CatBoost	<pre>parameters_CB = {'cb_depth': [3, 4, 5, 6, 7], 'cb_learning_rate': [0.01, 0.03, 0.1, 0.3, 0.5], 'cb_l2_leaf_reg': [1, 3, 5, 10, 15]}</pre>	<ul style="list-style-type: none"> F1-Score Valid = 0.71899. Không cải thiện được.
Light GBM	<pre>parameters_LGBM = { # 'lgbm_max_bin': [200, 230, 255, 270, 300], # 'lgbm_learning_rate': [0.01, 0.03, 0.1, 0.3, 0.5], # 'lgbm_num_leaves': [10, 20, 31, 40, 50], 'lgbm_max_depth': [-1, 4, 5, 6, 7], 'lgbm_lambda_l1': [1, 3, 5, 7, 9], 'lgbm_min_data_in_leaf': [15, 20, 25, 30, 35]}</pre>	<ul style="list-style-type: none"> Lần 1: F1-Score Valid = 0.70652. Lần 2: F1-Score Valid = 0.7037 LGBMClassifier(learning_rate=0.3, max_bin=300, random_state=0).
MLP	<pre>parameters_MLP = { 'mlp_learning_rate_init': [0.001, 0.005, 0.01, 0.03, 0.05], 'mlp_activation': ['identity', 'logistic', 'tanh', 'relu'], 'mlp_solver': ['lbfgs', 'sgd', 'adam'], 'mlp_learning_rate': ['constant', 'invscaling', 'adaptive']}</pre>	<ul style="list-style-type: none"> F1-Score Valid = 0.71078. Không cải thiện được.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Tinh chỉnh siêu tham số và Kết quả

(2) Tuning thủ công

CatBoost	<ul style="list-style-type: none">• CatBoostClassifier(verbose=False, eval_metric='F1', random_state=0).• Giữ nguyên.
Light GBM	<ul style="list-style-type: none">• LGBMClassifier(learning_rate=0.3, max_depth=6, n_estimators=1000, random_state=0).• F1-Score Valid = 0.736, đã cải thiện nhưng chưa đạt mục tiêu đề ra.
MLP	<ul style="list-style-type: none">• MLPClassifier(random_state=0).• Giữ nguyên.



VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

```
CB_best_model = CatBoostClassifier(  
    verbose=False, eval_metric='F1', random_state=0  
)  
  
LGBM_best_model = lgb.LGBMClassifier(  
    learning_rate=0.3, max_depth=6, n_estimators=1000, random_state=0  
)  
  
MLP_best_model = MLPClassifier(random_state=0)  
  
estimators = [  
    ('CB', CB_best_model), ('LGBM', LGBM_best_model), ('MLP', MLP_best_model)  
]  
  
stacking_model = StackingClassifier(  
    estimators=estimators,  
    final_estimator=LogisticRegression(random_state=0),  
    n_jobs=2  
)
```



VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(1) Không sử dụng SMOTE: chưa tunning threshold

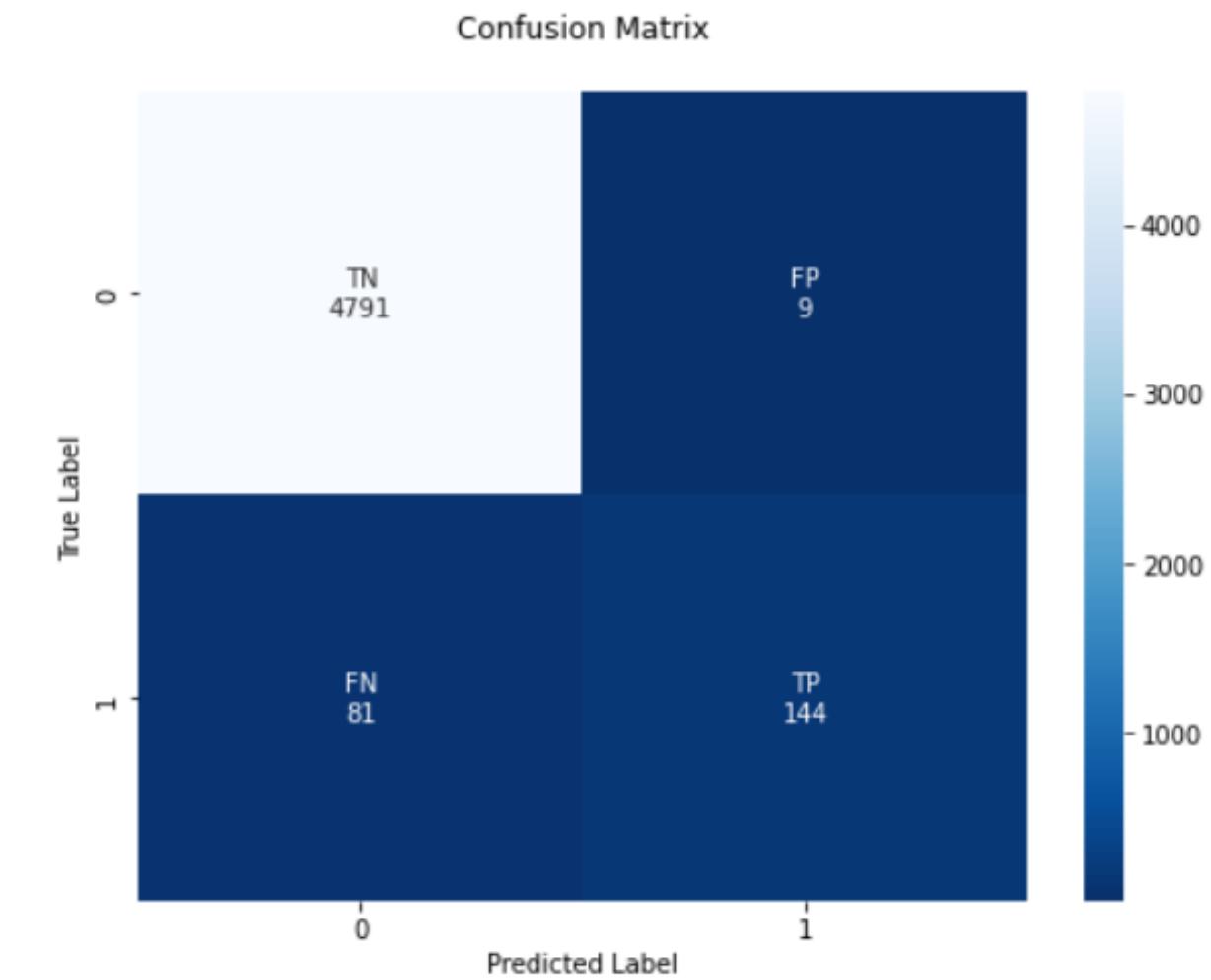
TRAINING DATA

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19195
1	1.00	0.96	0.98	901
accuracy				20096
macro avg	1.00	0.98	0.99	20096
weighted avg	1.00	1.00	1.00	20096

VALIDATION DATA

	precision	recall	f1-score	support
0	0.98	1.00	0.99	4800
1	0.94	0.64	0.76	225
accuracy				5025
macro avg	0.96	0.82	0.88	5025
weighted avg	0.98	0.98	0.98	5025

F1-Score Valid = 0.7619



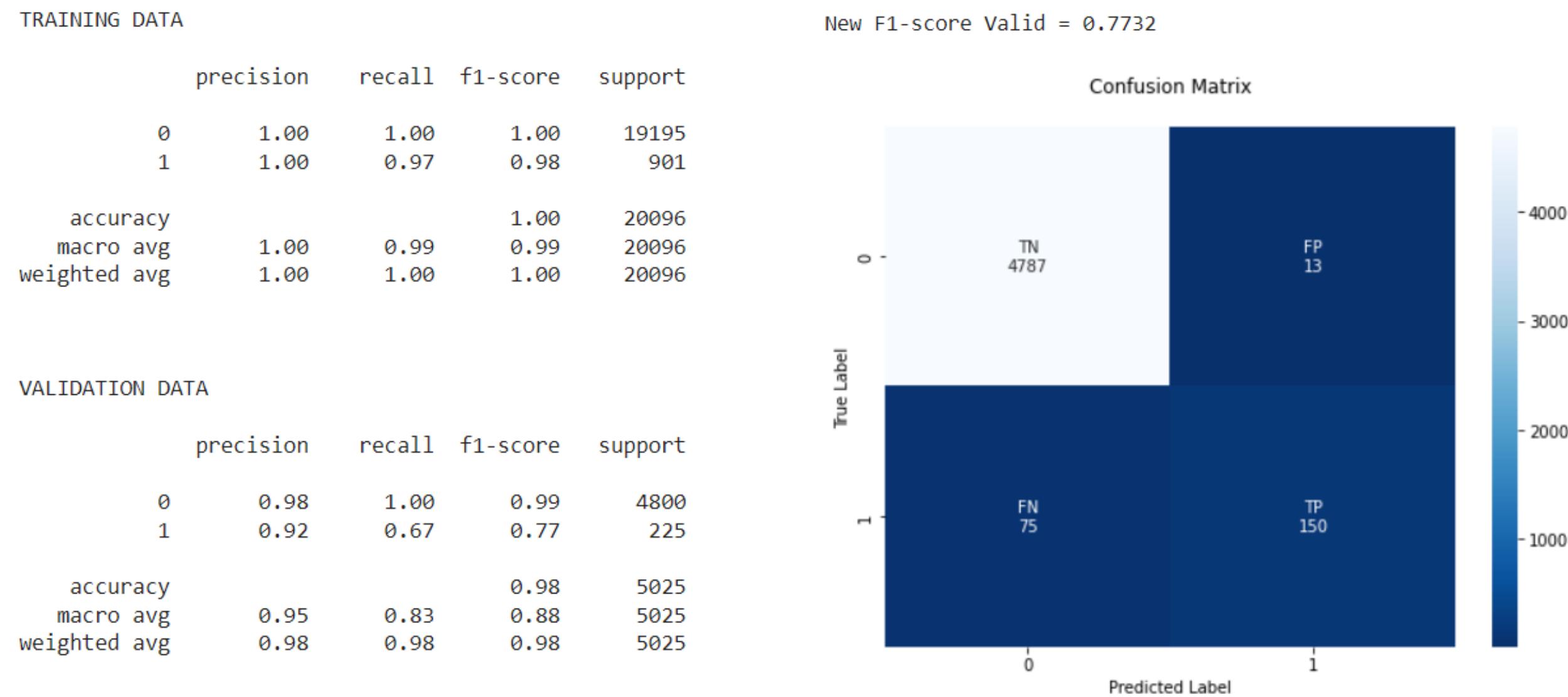
Kết quả: F1-Score Valid = 0.7619, đã đạt mục tiêu đề ra.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(1) Không sử dụng SMOTE: threshold = 0.38095 (1/2 F1-score ban đầu)



Kết quả: F1-Score Valid = 0.7732, đã đạt mục tiêu đề ra, tốt hơn so với model Stacking ban đầu.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(1) Không sử dụng SMOTE: threshold = 0.38095 (1/2 F1-score ban đầu)

Kết quả theo forecasting_period

	forecasting_period	f1_valid	recall_valid	precision_valid	accuracy_valid
0	1	0.835821	0.777778	0.903226	0.988518
1	2	0.734694	0.610169	0.923077	0.980966
2	3	0.725490	0.596774	0.925000	0.980583
3	4	0.809917	0.720588	0.924528	0.981732



VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(1) Không sử dụng SMOTE: threshold = 0.089 (optimal)

TRAINING DATA

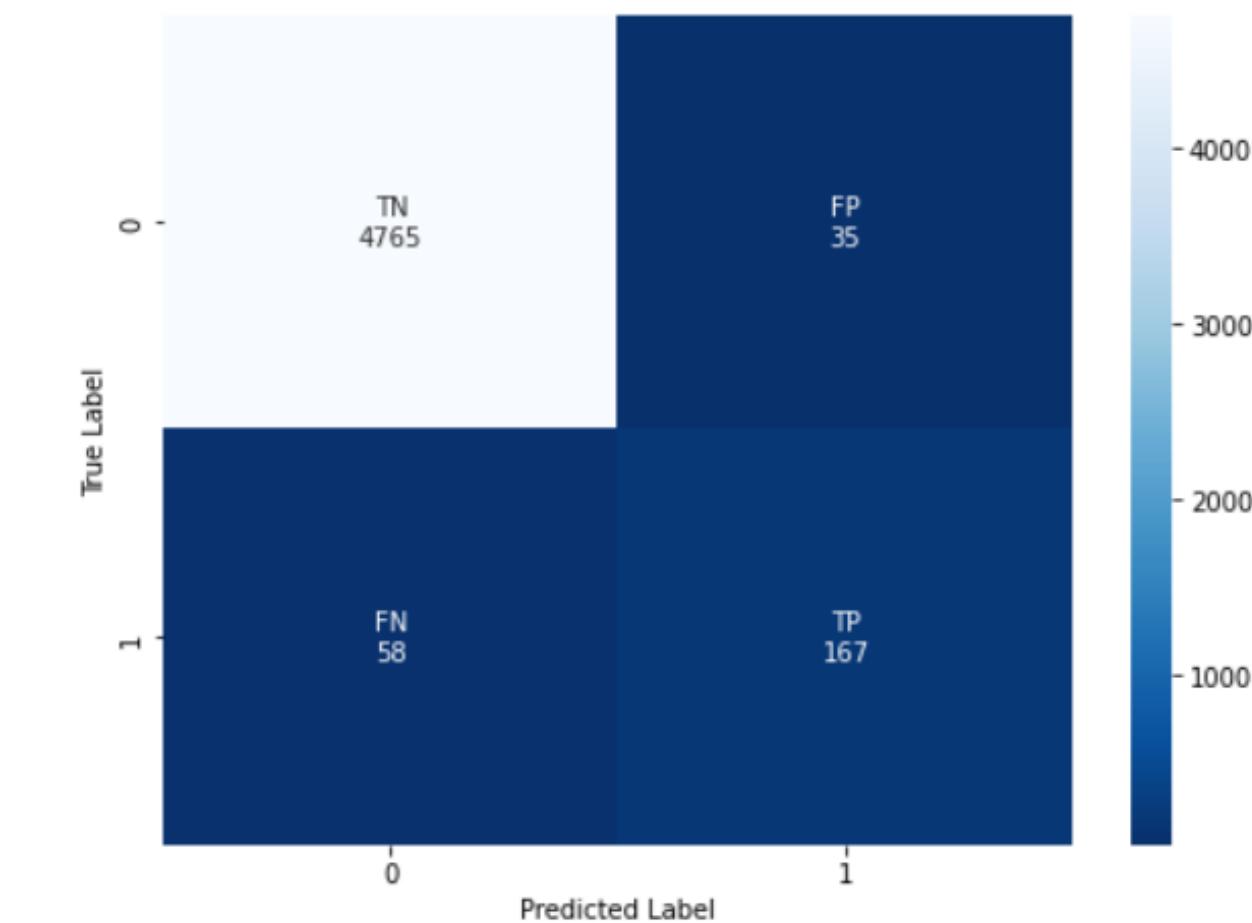
	precision	recall	f1-score	support
0	1.00	1.00	1.00	19195
1	0.96	0.99	0.98	901
accuracy			1.00	20096
macro avg	0.98	1.00	0.99	20096
weighted avg	1.00	1.00	1.00	20096

VALIDATION DATA

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4800
1	0.83	0.74	0.78	225
accuracy			0.98	5025
macro avg	0.91	0.87	0.89	5025
weighted avg	0.98	0.98	0.98	5025

Optimal F1-score Valid = 0.7822

Confusion Matrix



Kết quả: F1-Score Valid = 0.7822, đã đạt mục tiêu đề ra và là kết quả tốt nhất hiện tại.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(1) Không sử dụng SMOTE: threshold = 0.089 (optimal)

Kết quả theo forecasting_period

forecasting_period	f1_valid	recall_valid	precision_valid	accuracy_valid
0	1	0.833333	0.833333	0.833333
1	2	0.728972	0.661017	0.812500
2	3	0.761062	0.693548	0.843137
3	4	0.814815	0.808824	0.820896

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(2) Sử dụng SMOTE

	model	f1_train	f1_valid	recall_train	recall_valid	precision_train	precision_valid	accuracy_train	accuracy_valid
0	Stacking - 0.1	0.996663	0.764103	0.994451	0.662222	0.998885	0.903030	0.999701	0.981692
1	Stacking - 0.8	0.999445	0.754522	0.998890	0.648889	1.000000	0.901235	0.999950	0.981095
2	Stacking - 0.9	1.000000	0.751295	1.000000	0.644444	1.000000	0.900621	1.000000	0.980896
3	Stacking - 0.3	0.996126	0.746341	0.998890	0.680000	0.993377	0.827027	0.999652	0.979303
4	Stacking - 0.4	0.999445	0.744063	0.998890	0.626667	1.000000	0.915584	0.999950	0.980697
5	Stacking - 0.6	0.999445	0.741514	0.998890	0.631111	1.000000	0.898734	0.999950	0.980299
6	Stacking - 0.2	0.998332	0.740554	0.996670	0.653333	1.000000	0.854651	0.999851	0.979502
7	Stacking - 0.7	1.000000	0.740360	1.000000	0.640000	1.000000	0.878049	1.000000	0.979900
8	Stacking - 1.0	0.998891	0.737113	1.000000	0.635556	0.997785	0.877301	0.999900	0.979701
9	Stacking - 0.5	0.998334	0.734908	0.997780	0.622222	0.998889	0.897436	0.999851	0.979900

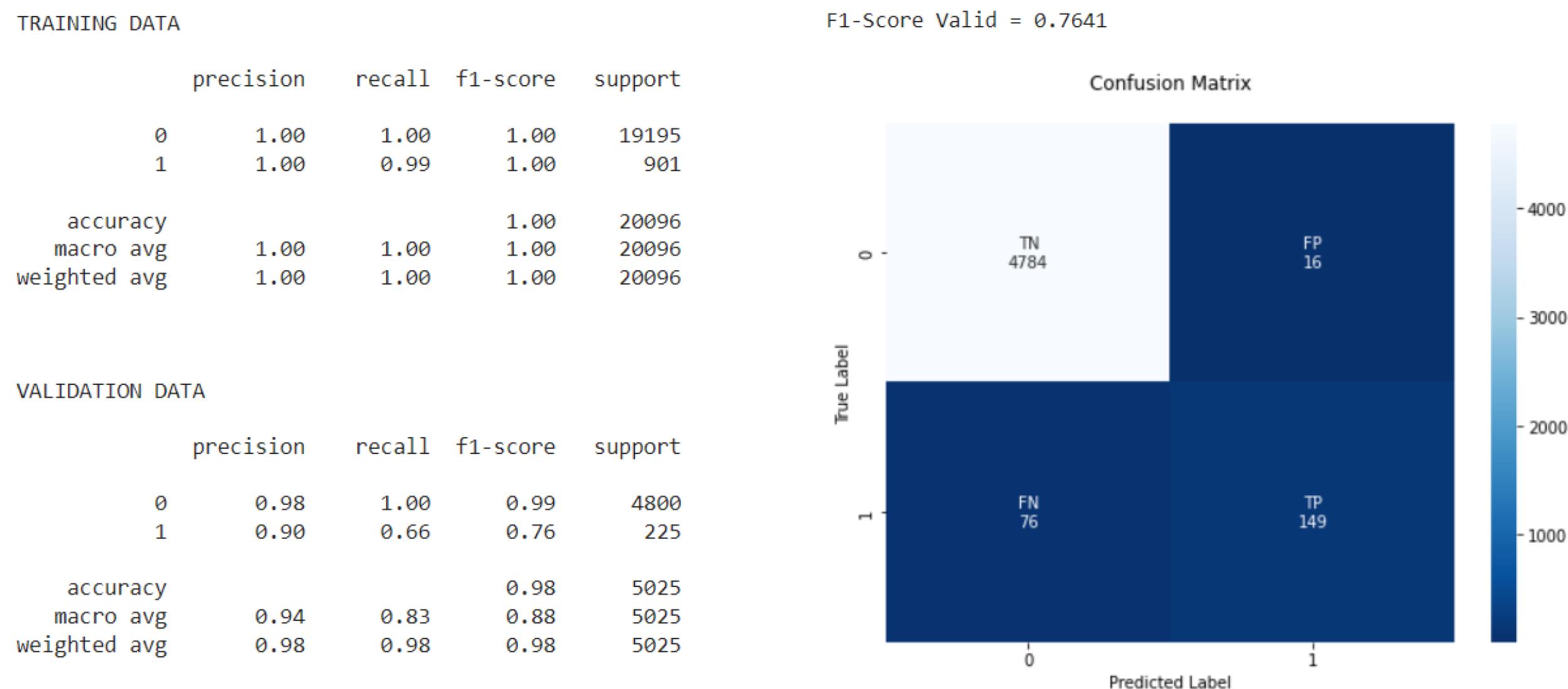
Kết quả: Sử dụng SMOTE để sinh mẫu với tỉ lệ class=1/class=0 = 0.1 giúp model Stacking đạt được F1-Score Valid tốt nhất bằng 0.764103, đạt được mục tiêu đề ra và tốt hơn so với model Stacking không sử dụng SMOTE + trước khi tuning threshold.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(2) Sử dụng SMOTE: threshold = 0.5 (optimal)



Kết quả: F1-Score Valid = 0.7641, đã đạt mục tiêu đề ra nhưng không tốt bằng kết quả của model Stacking khi không sử dụng SMOTE + sau khi đã tuning threshold.

VII. Modelling & Evaluation (Lập mô hình & Đánh giá)

7.2. Cải thiện mô hình

Kết hợp các mô hình sử dụng Stacking và Kết quả

(2) Sử dụng SMOTE: threshold = 0.5 (optimal)

Kết quả theo forecasting_period

forecasting_period	f1_valid	recall_valid	precision_valid	accuracy_valid
0	1 0.818182	0.750000	0.900000	0.987474
1	2 0.740000	0.627119	0.902439	0.980966
2	3 0.693069	0.564516	0.897436	0.978502
3	4 0.813008	0.735294	0.909091	0.981732

VIII. Conclusion (Kết luận)

Tổng hợp kết quả tốt nhất trên tập validation và tập test (Kaggle)

STT	Model	F1-Score Valid	Accuracy Score Valid	Public Score	Private Score
1	Stacking + Không SMOTE + Threshold=0.089	0.7822	0.98149	0.97413	0.97952
2	Stacking + Không SMOTE + Threshold=0.38095	0.7732	0.98249	0.97833	0.98297
3	Stacking + SMOTE 0.1	0.7641	0.98169	0.98060	0.98135

Nhận xét :

- Trên tập valid, model tốt nhất là Stacking không sử dụng SMOTE + threshold = 0.089.
- Trên tập test, model tốt nhất là Stacking không sử dụng SMOTE + threshold = 0.38095.
- Chênh lệch điểm số giữa tập validation và tập test nhiều khả năng là do cuộc thi đang sử dụng thang đo Accuracy.
- Ngoài ra, vẫn còn tồn tại tình trạng overfitting. Tuy vậy, hiện tại vẫn chấp nhận tình trạng overfitting này để đạt được F1-Score tốt nhất hiện tại trên tập validation.

Kết luận: Dựa trên mục tiêu đặt ra ban đầu của dự án là tìm ra model tốt nhất có thể dựa trên F1-Score Valid Lựa chọn model **Stacking không sử dụng SMOTE + threshold = 0.089**.



THANK YOU !!!