

Anomaly Detection in Credit Card Transactions

DS200 - Phân tích dữ liệu lớn

GVHD: TS. Đỗ Trọng Hợp

Trang 1

THÀNH VIÊN NHÓM

1.

Tăng Minh Hiển

21520229

2.

Châu Thiên Long

21520331

3.

Nguyễn Thái Thành Long

21520334

NỘI DUNG

01 GIỚI THIỆU BÀI TOÁN

02 DATASET

03 PHƯƠNG PHÁP

04 ĐÁNH GIÁ

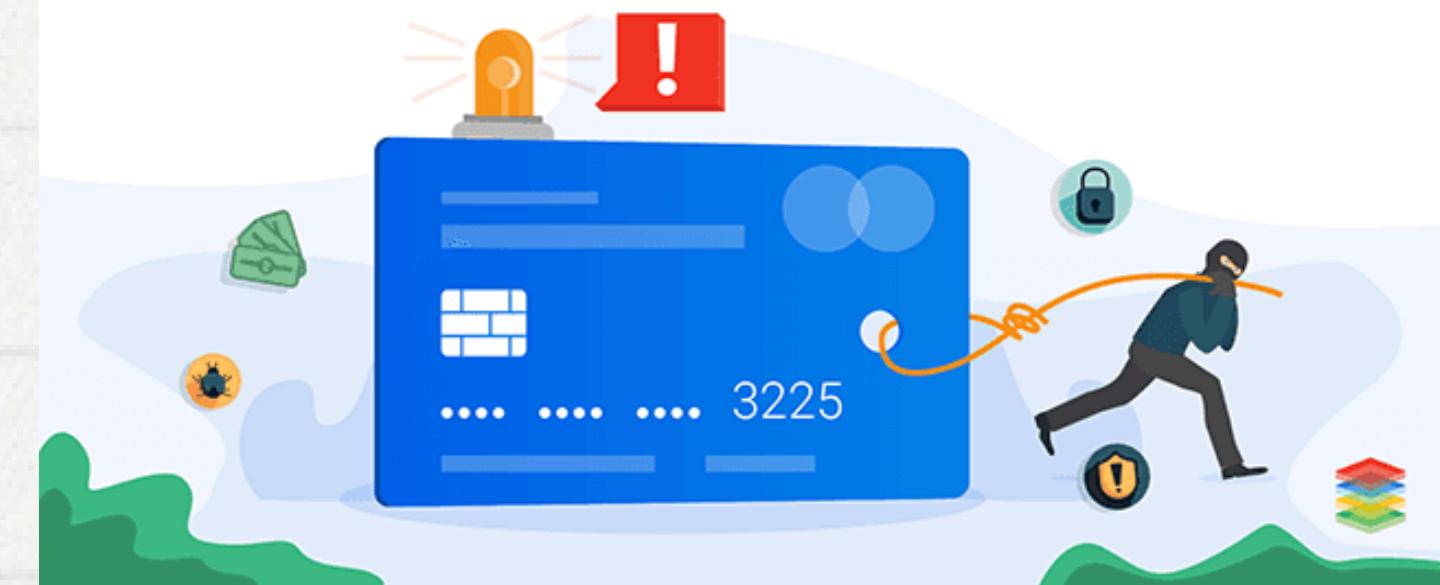
05 KẾT LUẬN

#1 GIỚI THIỆU

Thẻ tín dụng là hình thức giao dịch phổ biến hiện nay. Do đó, đây là một trong những mục tiêu lừa đảo phổ biến nhất nhưng không phải là mục tiêu duy nhất.

Gian lận thẻ tín dụng đã và đang là vấn đề ngày càng gia tăng trong ngành thẻ tín dụng. Với số lượng giao dịch lớn hằng ngày, việc phát hiện gian lận khi giao dịch bằng thẻ tín dụng đem lại nhiều lợi ích nhưng cũng là thách thức lớn.

Credit Card Fraud Detection



#1 GIỚI THIỆU

BÀI TOÁN

Mục tiêu: Xây dựng hệ thống nhận dạng các giao dịch thẻ tín dụng đáng ngờ

Input:

- Tập dữ liệu các giao dịch thẻ tín dụng gồm các giao dịch đã được phân loại đáng ngờ hay không
- Các giao dịch mới chưa được phân loại gian lận hay không

Output:

- Dự báo các giao dịch mới xem có thuộc giao dịch gian lận hay không

#2 DATASET

The screenshot shows a Kaggle dataset page for 'Credit Card Fraud Detection'. At the top, there's a navigation bar with a blue 'KAGGLE' logo, a search bar containing 'credit card', and a user icon. Below the header, the title 'Credit Card Fraud Detection' is displayed in large, bold, dark blue text. Underneath the title, a subtitle reads 'Anonymized credit card transactions labeled as fraudulent or genuine'. To the right of the text is a small image showing several credit cards. At the top right of the page, there are several interactive buttons: a profile icon, a 'MACHINE LEARNING GROUP - ULB AND 1 COLLABORATOR · UPDATED 6 YEARS AGO' link, a '11353' badge with up and down arrows, a 'New Notebook' button, a 'Download (69 MB)' button with a download icon, and a three-dot menu icon.

Credit Card Fraud Detection

Anonymized credit card transactions labeled as fraudulent or genuine



- Tác giả: Machine Learning Group
- Nội dung: Tập dữ liệu gồm các giao dịch thực hiện bằng thẻ tín dụng vào tháng 9 năm 2013 bởi các chủ thẻ ở Châu Âu.

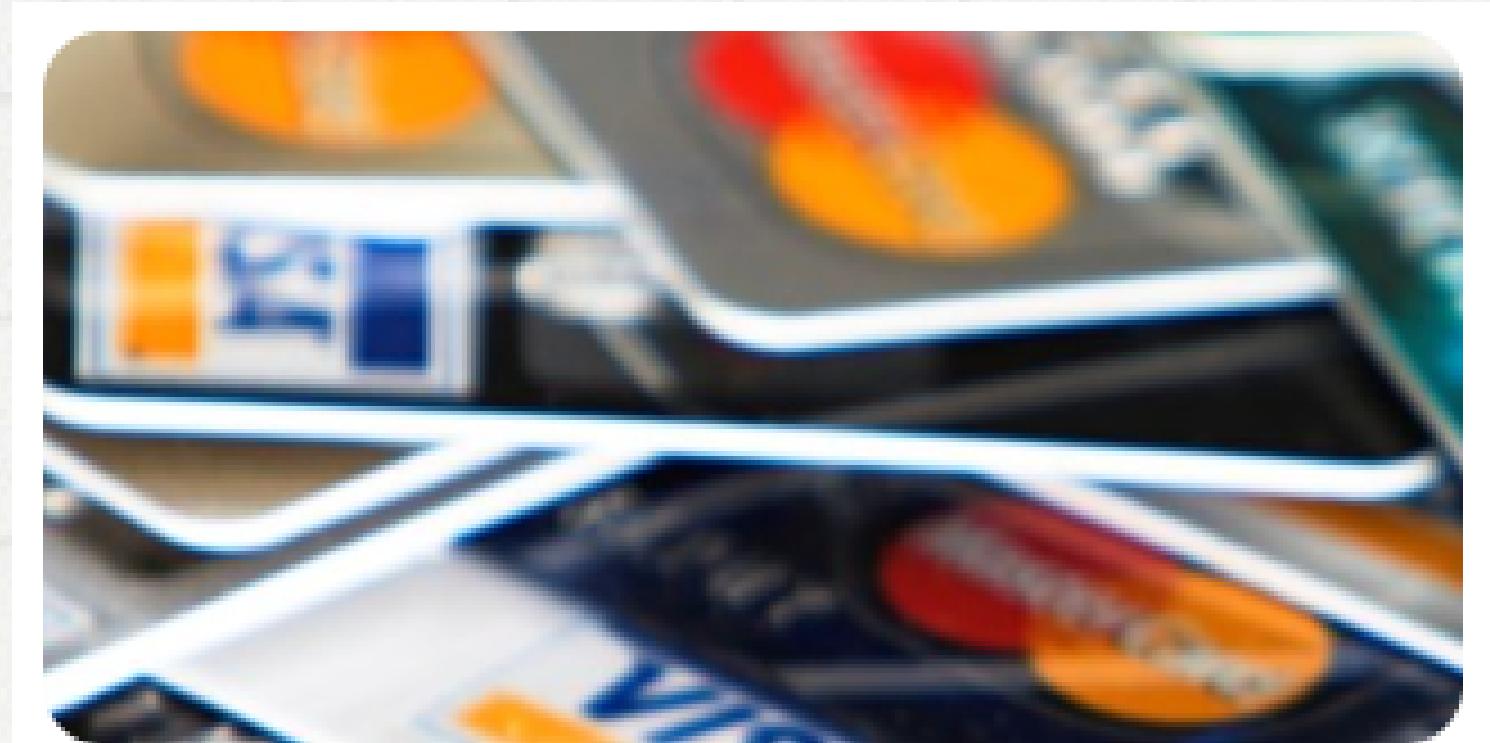
Nguồn dữ liệu: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?select=creditcard.csv>

Trang 6

#2 DATASET

Thông tin về bộ dữ liệu:

- Feature: Gồm 30 cột là các cột dạng numerical:
 - Time: số giây trôi qua giữa mỗi giao dịch với giao dịch đầu tiên trong tập dữ liệu.
 - Amount: giá trị của giao dịch
 - Các cột được mã hóa (V1 - V28): Là kết quả của phép biến đổi PCA từ dữ liệu gốc của tác giả.
- Target: 2 class: (1 - Fraud / 0 - No Fraud)

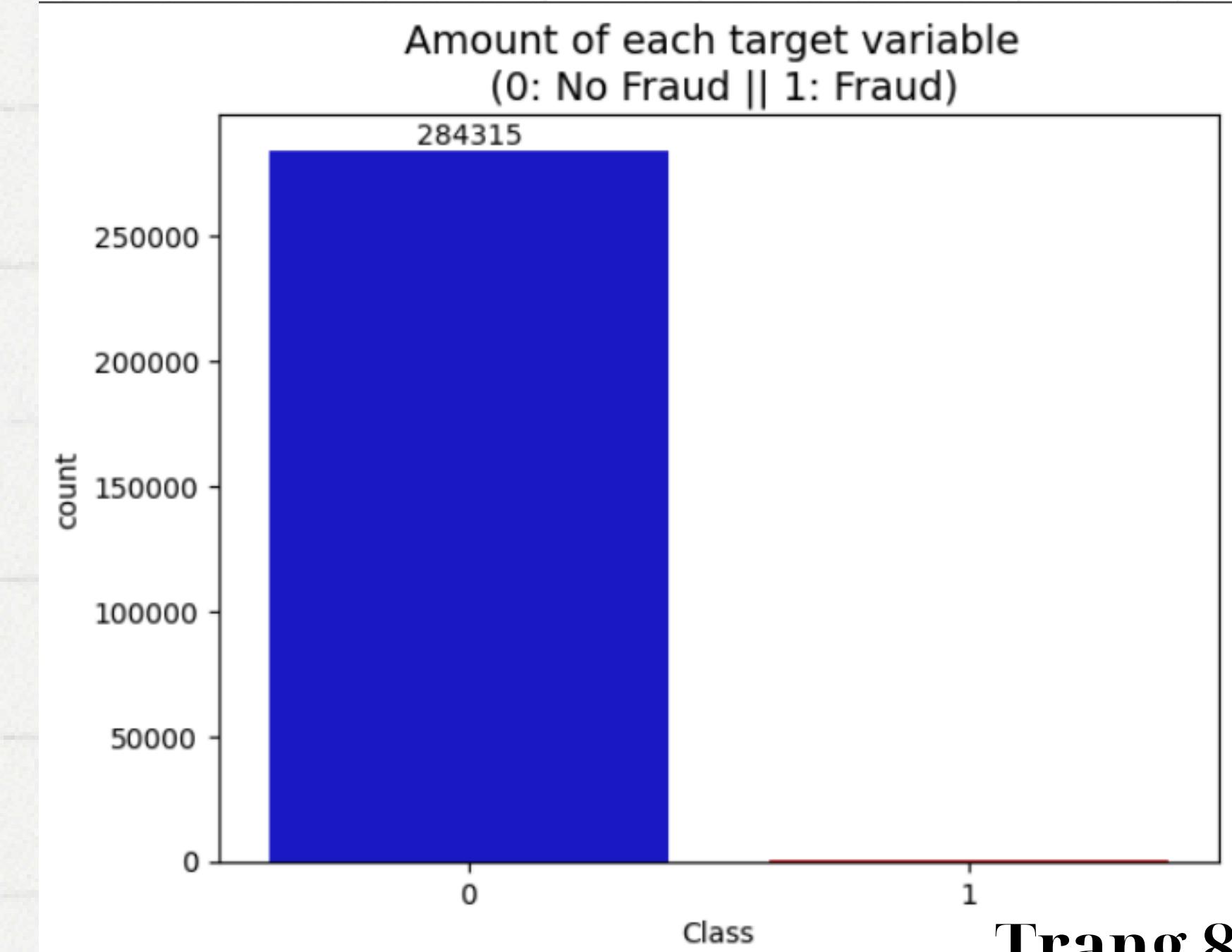


#2 DATASET

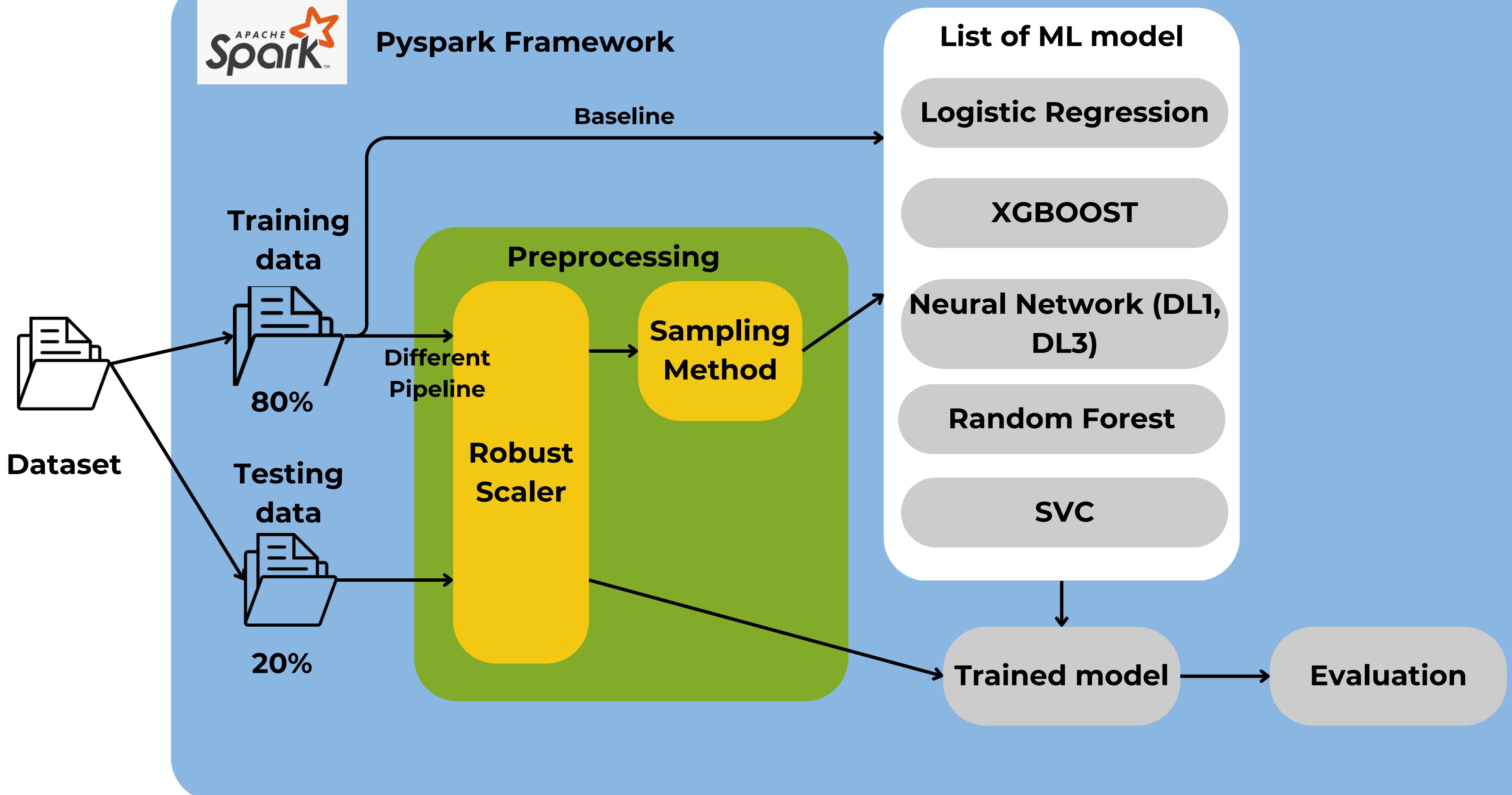
PHÂN PHỐI NHÃN DỮ LIỆU

Phân phối số lượng nhãn dữ liệu:

- 284315 nhãn 0 - No Fraud (99.83%)
- 492 nhãn 1 - Fraud (0.17%)



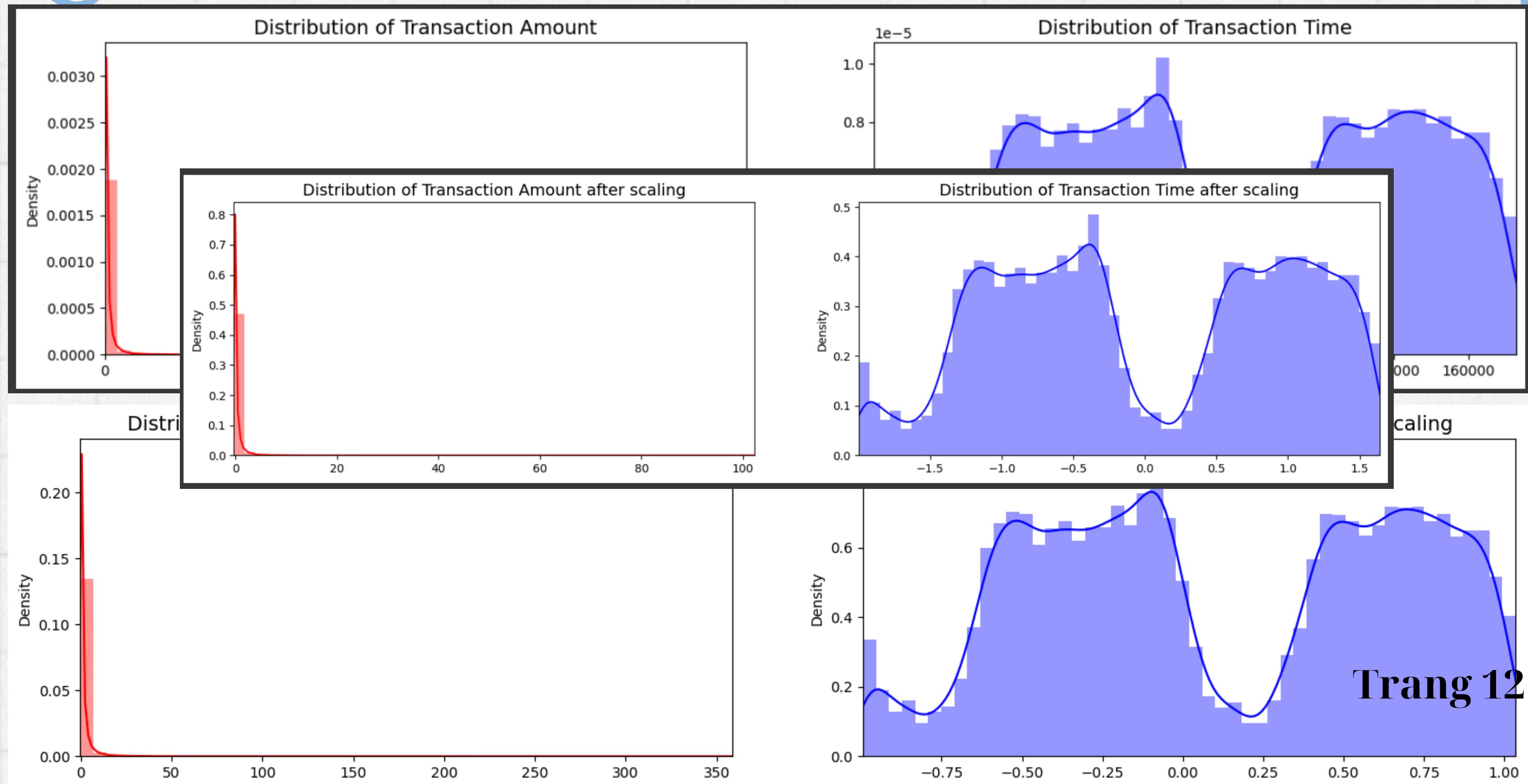
#3 PHƯƠNG PHÁP



#3 PHƯƠNG PHÁP

- Preprocessing:
 - Do dữ liệu bị lệch nặng nên nhóm có sử dụng các phương pháp cân bằng: Undersampling, Oversampling by duplicating minority class và SMOTE.
- Model: Nhóm sử dụng các mô hình phân loại phổ biến như Logistic Regression, Random Forest, XGBoost và mô hình học sâu.

NORMALIZE (Account, Time COLUMN)



#3 PHƯƠNG PHÁP

Logistic Regression

- Là mô hình Máy học được sử dụng rộng rãi với các bài toán phân lớp nhị phân, dùng hàm sigmoid để tạo ra giá trị đầu ra là giá trị xác suất trong khoảng [0,1]
- Cài đặt, sử dụng mô hình: from pyspark.ml.classification import LogisticRegression
- Cài đặt tham số: Điều chỉnh 2 tham số `labelCol='label'`, `featuresCol='features'`

#3 PHƯƠNG PHÁP

SUPPORT VECTOR MACHINE

- Là mô hình Máy học được sử dụng rộng rãi với các bài toán phân lớp nhị phân, dùng hàm sigmoid để tạo ra giá trị đầu ra là giá trị xác suất trong khoảng [0,1]
- Cài đặt, sử dụng mô hình: from pyspark.ml.classification import LinearSVC
- Cài đặt tham số: Điều chỉnh 2 tham số `labelCol='label'`, `featuresCol='features'`

#3 PHƯƠNG PHÁP

Random Forest

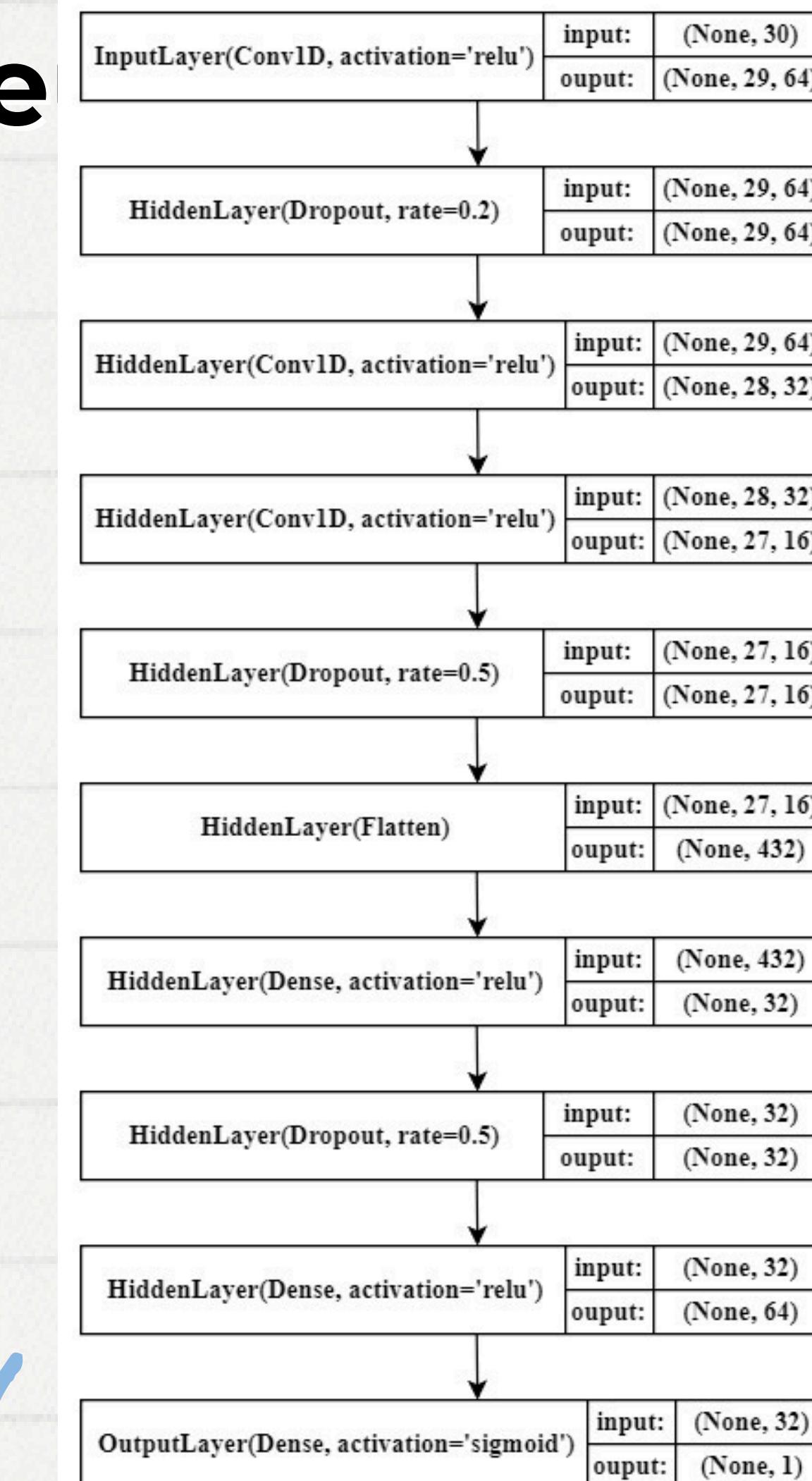
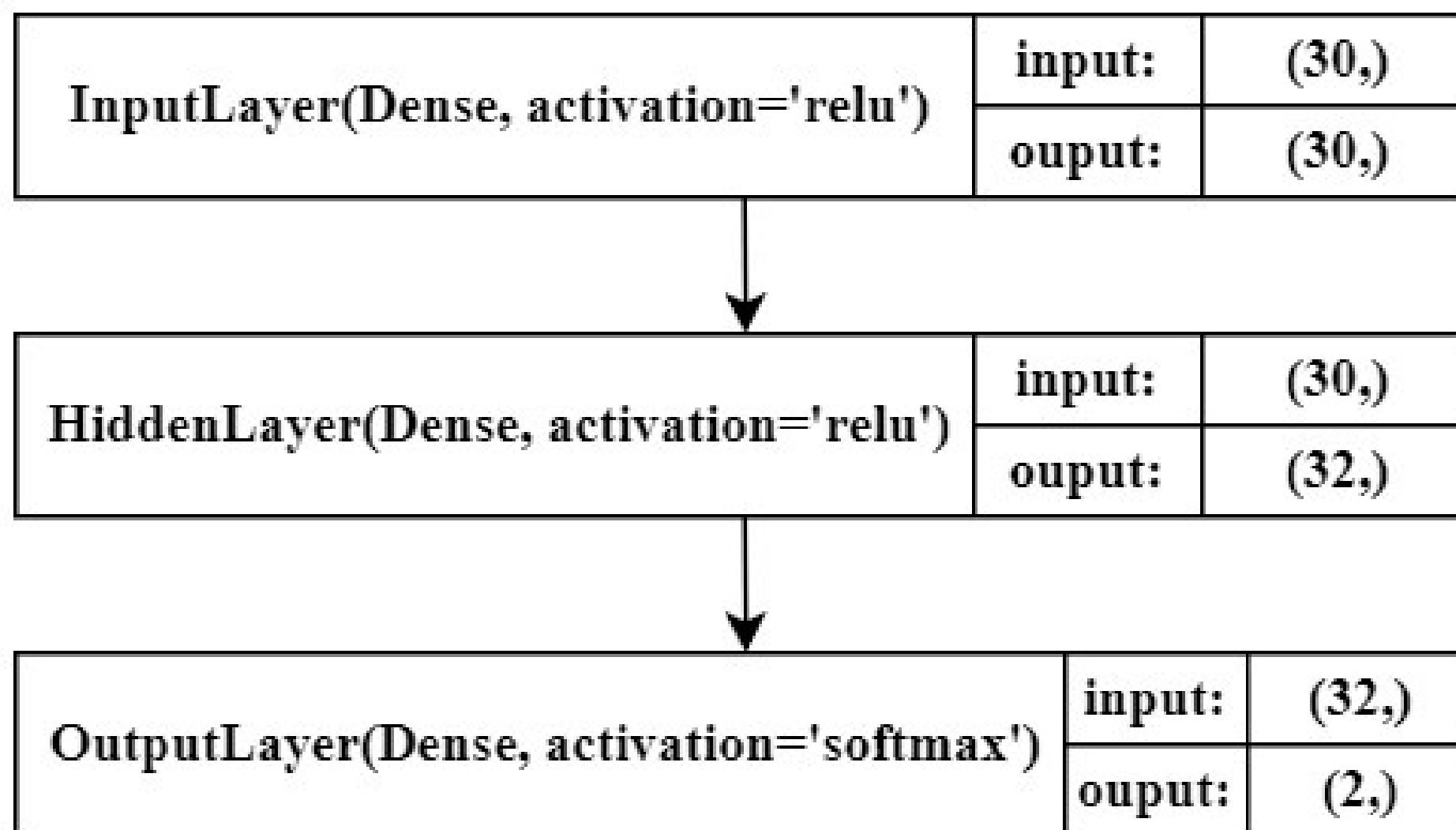
- Là mô hình Máy học sử dụng kỹ thuật Ensemble Learning từ việc tạo nhiều cây quyết định, sau đó sử dụng Majority Voting để đưa ra dự đoán phân loại cuối cùng. Với ưu điểm tạo nhiều cây quyết định, giảm overfitting và tối ưu đánh giá các đặc trưng, nhóm đã lựa chọn mô hình này cho bài toán.
- Cài đặt, sử dụng mô hình: from pyspark.ml.classification import RandomForestClassifier
- Cài đặt tham số: labelCol='label', featuresCol='features'

#3 PHƯƠNG PHÁP

XGBOOST (Extreme Gradient Boosting)

- Là kết hợp nhiều mô hình cây quyết định (decision tree) yếu thành một mô hình mạnh mẽ hơn. Mỗi mô hình cây quyết định được tạo ra nhằm giảm thiểu lỗi dự đoán của mô hình tổng thể. Quá trình này được lặp đi lặp lại cho đến khi đạt được độ chính xác mong muốn hoặc khi đáp ứng một số tiêu chí dừng.
- Cài đặt, sử dụng mô hình: from xgboost.spark import SparkXGBClassifier
- Cài đặt tham số: labelCol='label', featuresCol='features'

#3 PHƯƠNG PHÁP - Ne



#4 ĐÁNH GIÁ

ĐỘ ĐO ĐÁNH GIÁ

- Macro Precision: trung bình cộng Precision trên từng class
- Macro Recall: trung bình cộng Recall trên từng class
- Macro F1 score: trung bình cộng F1 score trên từng class

$$Macro\ Precision = \frac{\sum_{i=0}^n Precision_i}{n} \text{ với } n \text{ là số lớp}$$

$$Macro\ Recall = \frac{\sum_{i=0}^n Recall_i}{n} \text{ với } n \text{ là số lớp}$$

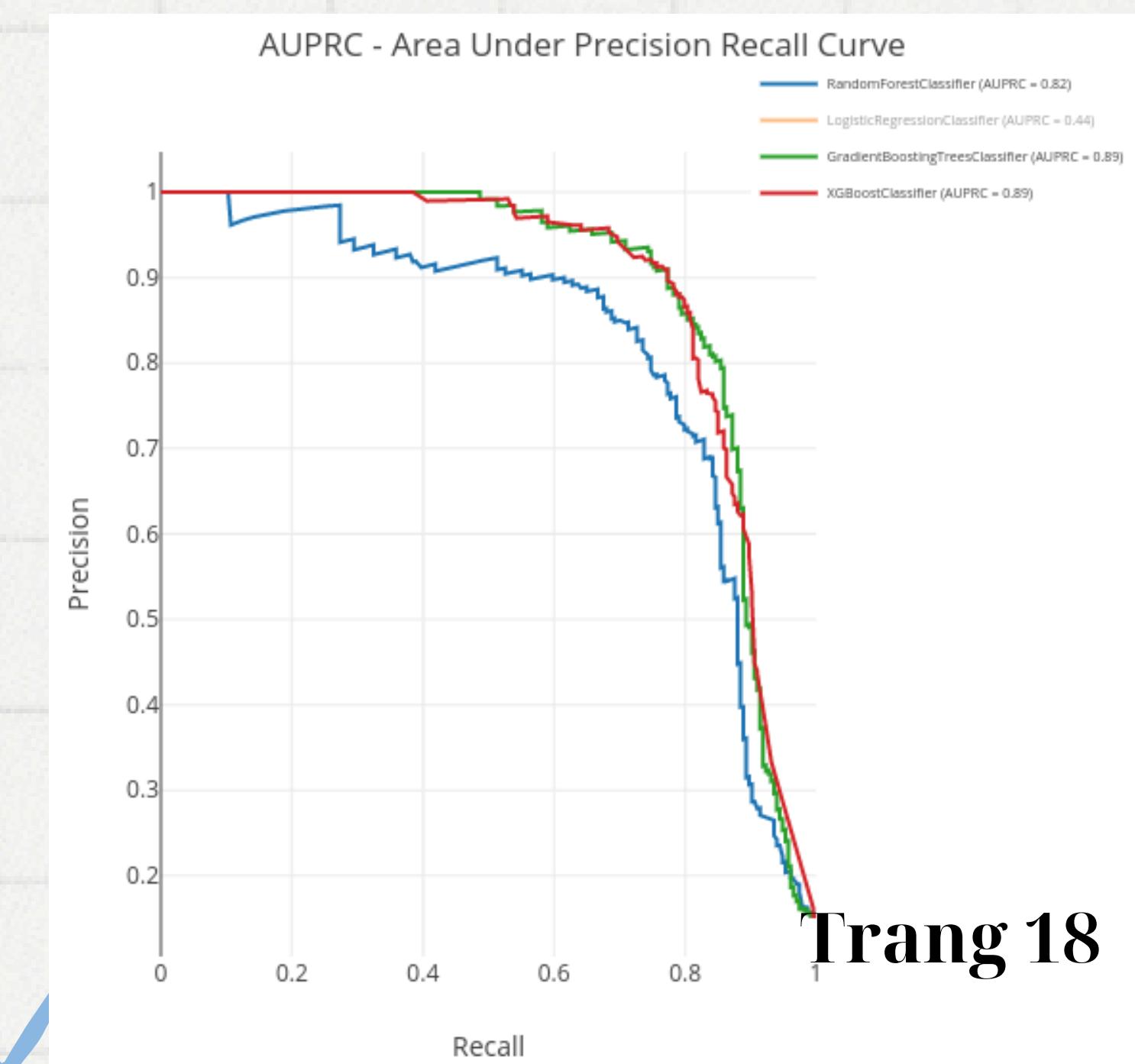
$$Macro\ F1 = \frac{\sum_{i=0}^n F1_i}{n} \text{ với } n \text{ là số lớp}$$

#4 ĐÁNH GIÁ

ĐỘ ĐO ĐÁNH GIÁ

AUPRC (Area under the precision-recall curve):

- Diện tích dưới đường cong Precision-Recall.
Nó tính toán mức độ hiệu quả của mô hình trong việc cân bằng giữa Precision và Recall trên toàn bộ các ngưỡng dự đoán.
- Đây là chỉ số hữu ích khi đối mặt với dữ liệu mất cân bằng.



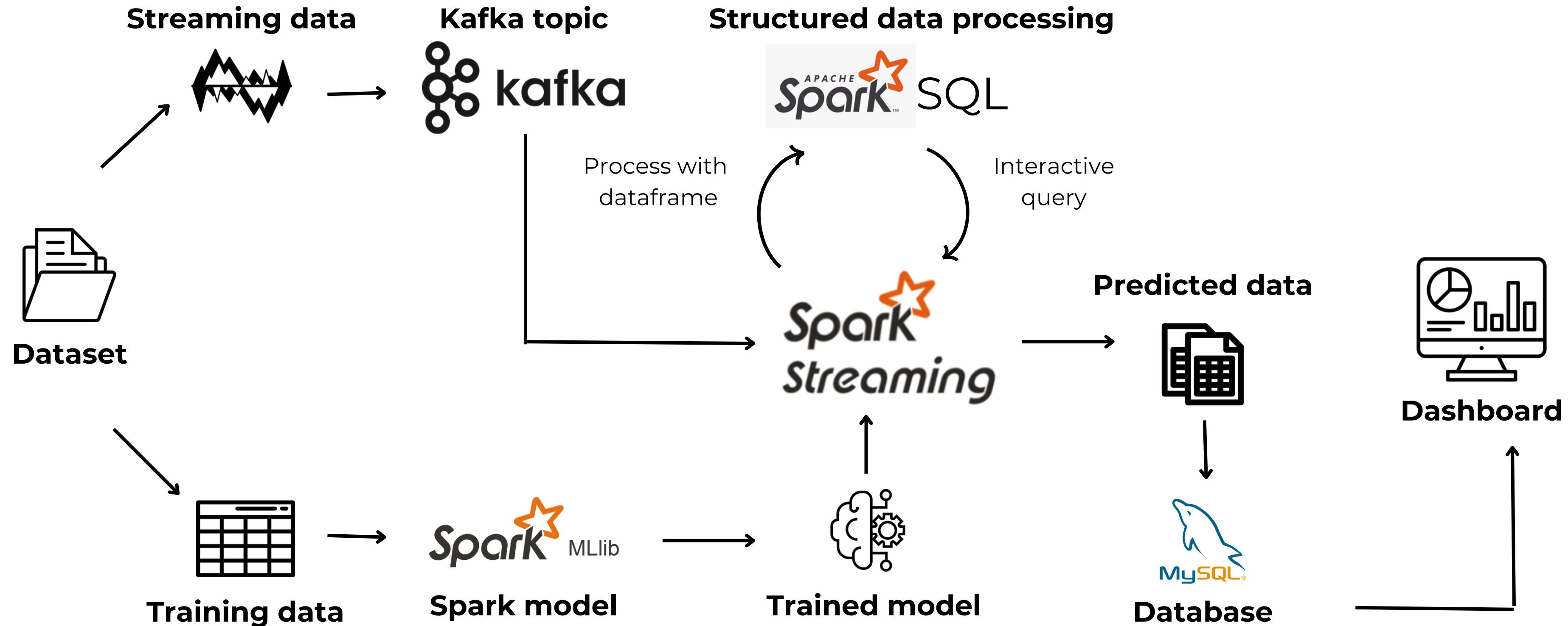
Pipeline		Macro Precision	Macro Recall	Macro F1 Score	AUPRC	AUC-ROC
(Baseline)	Logistic Regression	0.95	0.83	0.88	0.72	0.96
RobustScaler + Undersampling	Logistic Regression	0.76	0.90	0.81	0.72	0.98
	Random Forest	0.84	0.91	0.87	0.74	0.98
RobustScaler + Oversampling duplicating minority class	Logistic Regression	0.63	0.91	0.70	0.73	0.99
	Random Forest	0.83	0.91	0.86	0.72	0.98
RobustScaler	DL1	0.93	0.92	0.93	0.91	0.99
	DL3	0.93	0.92	0.93	0.78	0.98
	XGBoost	0.97	0.89	0.93	0.73	0.98
	SVC	0.89	0.87	0.88	0.71	0.91
RobustScaler + OverSampling (SMOTE library)	Logistic Regression	0.53	0.94	0.56	0.73	0.99
	Random Forest	0.61	0.92	0.68	0.73	0.98

Pipeline		Precision_Class1	Recall_Class1	F1 Score_Class1
(Baseline)	Logistic Regression	0.90	0.65	0.76
RobustScaler + Undersampling	Logistic Regression	0.52	0.80	0.63
	Random Forest	0.68	0.81	0.74
RobustScaler + Oversampling duplicating minority class	Logistic Regression	0.26	0.82	0.39
	Random Forest	0.65	0.81	0.72
RobustScaler	DL1	0.86	0.85	0.85
	DL3	0.85	0.78	0.81
	XGBoost	0.96	0.77	0.85
	SVC	0.79	0.74	0.76
RobustScaler + OverSampling (SMOTE library)	Logistic Regression	0.07	0.91	0.12
	Random Forest	0.23	0.84	0.36

#4 ĐÁNH GIÁ

- So sánh về hiệu năng với mô hình Baseline, có thể thấy phương pháp RobustScaler + OverSampling (SMOTE library), có kết quả thấp nhất và cho kết quả kém nhất. Ngược lại, mô hình Neural Network (DL1) và XGBoost cho kết quả tốt nhất dựa trên các thông số
- Phương pháp Resampling có hiệu quả tốt với mô hình Random Forest khi cho kết quả tốt hơn nhiều mô hình khác nhưng với Logistic Regression lại cho kết quả thấp hơn
- Robust Scaler đem lại cải thiện hiệu suất ở hầu hết các mô hình so với baseline
- Với dữ liệu thuộc class 1, XGBoost và các mô hình Neural Network cho kết quả về chỉ số tốt.

System architecture



#5 KẾT LUẬN

Ưu điểm

- Nhóm thực hiện thử nghiệm với nhiều mô hình khác nhau và đã thu được các thử nghiệm tích cực.
- Các mô hình Máy học áp dụng là các mô hình dễ cài đặt và dễ nắm bắt về lý thuyết
- Áp dụng các công nghệ dữ liệu lớn để hiện thực hệ thống

Nhược điểm

- Dữ liệu đã bị biến đổi, gây hạn chế trong việc tìm hiểu mối quan hệ giữa thuộc tính và xây dựng mô hình
- Chưa áp dụng các mô hình tân tiến nhất của bài toán này
- Vẫn còn hạn chế và chưa đa dạng ở các bước tiền xử lý dữ liệu.
- Mô hình chưa được thử nghiệm để tối ưu tham số.
- Demo chưa tiếp cận gần thực tế

**CẢM ƠN THẦY VÌ
ĐÃ THEO DÕI**

DS200 - Phân tích dữ liệu lớn

Trang 25