Detecting Suspicious Activity in Financial Transactions

System Development for Fintech

AML Detection using Machine Learning with SHAP Interpretability

Kees Van Montfort
Raymond Zwaal

Parisa Ghazanfari
500955367

Ghazaleh Ghahremani
500955161

April 2025

Hogeschool van Amsterdam

Table of Contents

Introduction

Anti-money laundering (AML) is a critical aspect of modern financial systems that involves identifying suspicious financial behavior and preventing illicit transactions. This project focuses on building a machine learning system capable of detecting potential money laundering activities using a large-scale transaction dataset, SAML-D.

The dataset comprises over 9.5 million transactions, with only 0.1039% marked as suspicious, presenting a significant class imbalance challenge. Each transaction includes rich metadata such as transaction amount, time, payment type, currency, sender/receiver bank location, and flags for country/currency mismatch. These features offer a comprehensive basis for training robust detection models.

The goal is not only to achieve accurate predictions but also to ensure transparency in decision-making by applying SHAP (SHapley Additive exPlanations) to explain the output of our machine learning models (Lundberg & Lee, 2017).

Dataset Overview

The SAML-D dataset was constructed from a combination of academic literature, real-world banking records, and expert insights. It simulates realistic scenarios in which money laundering may occur. The dataset includes temporal and geographic attributes that allow the system to learn patterns in transactional behavior.

Key features include:

- **Date and Time**: Captures chronological patterns.
- **Sender/Receiver Account and Bank Location**: Useful for identifying behavioral anomalies and high-risk regions.
- **Transaction Amount**: Often a key indicator of suspicious activity.
- **Payment and Currency Type**: Helps detect atypical transactions such as high-value cash deposits or currency mismatches.
- **Is_Laundering**: The binary target variable.
- **Typology**: Categorizes the type of money laundering behavior (e.g., Fan Out, Fan In).

To support deeper analysis and model interpretability, we also incorporated the Laundering_type feature, which classifies transactions into specific typologies such as *Fan-In*, *Structuring*, *Over-Invoicing*, and others. Each of these labels reflects a unique laundering behavior pattern based on domain knowledge and expert definitions. While this column was not used directly for prediction to avoid label leakage, it was essential for exploratory analysis and

visual segmentation of suspicious behavior. A detailed dictionary explaining each laundering type is included in the appendix A for reference.

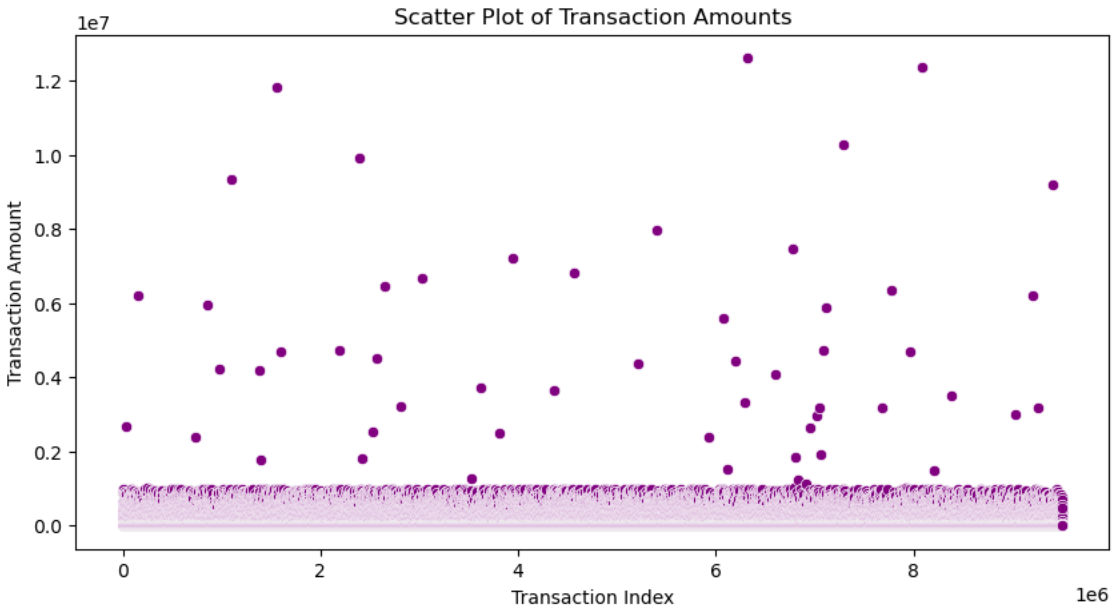
These features provide a comprehensive view of transaction behavior and were crucial in both model training and interpretability.

Exploratory Data Analysis

We began by visualizing the class imbalance, where most transactions were normal. As shown in the bar plot, this imbalance posed a significant challenge for modeling and necessitated the use of resampling strategies and class weight adjustments.

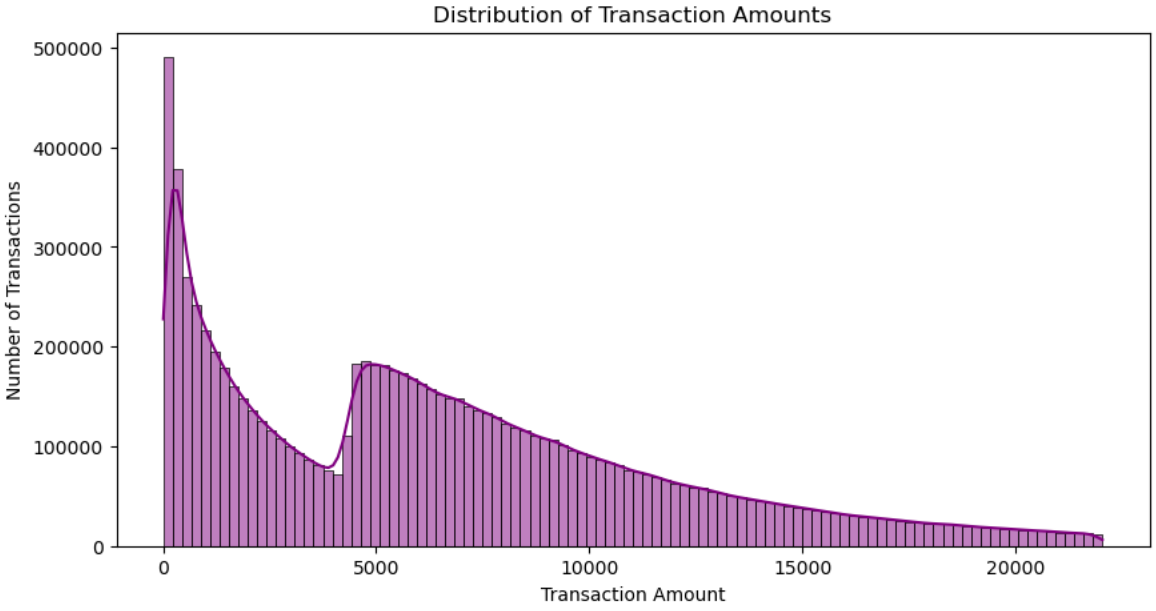
Scatter Plot of Transaction Amounts

This scatter plot illustrates the distribution of transaction amounts across all indexed transactions. The dense cluster near the bottom represents many transactions with relatively low values. However, several significant outliers appear with very high transaction amounts, suggesting potential anomalies. These high-value spikes are characteristic of suspicious behavior and serve as key indicators in the detection of money laundering activities.



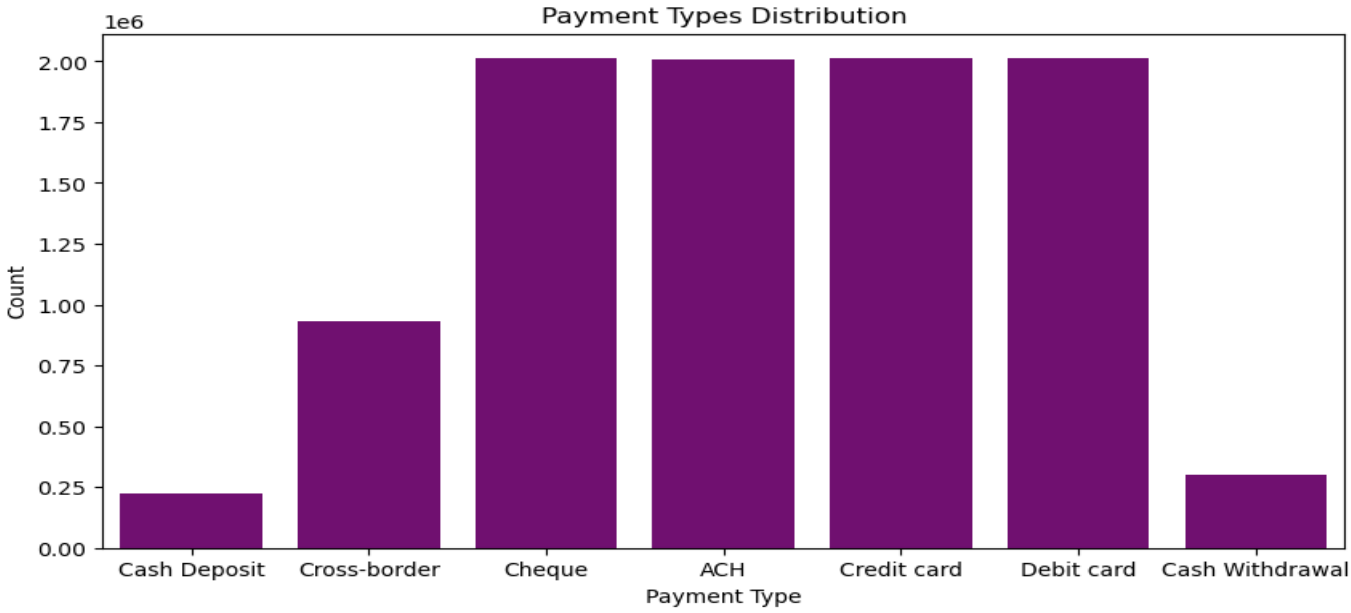
Distribution of Transaction Amounts

To better understand common transactional behavior, this histogram excludes extreme outliers and focuses on typical transaction ranges. Most transactions are concentrated at lower values, especially below 5,000, forming a right-skewed distribution. The sharp drop-off as the amount increases suggests that large transactions are rare and more likely to be anomalous. This refined view helps highlight subtle patterns in regular user behavior while keeping the focus on values where laundering may still occur in disguise



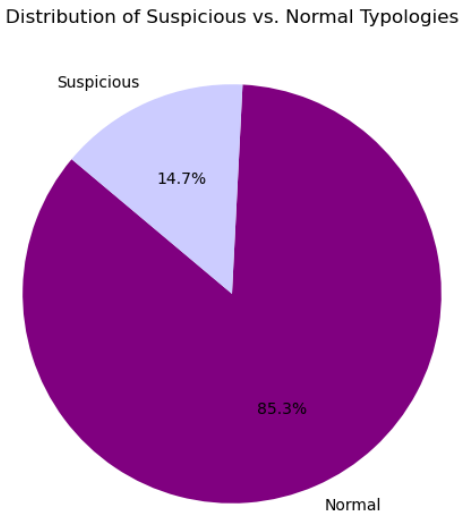
Payment Types Distribution

The distribution of payment methods reveals that cheque, ACH, credit card, and debit card transactions dominate the dataset, each exceeding 2 million occurrences. These common electronic payment types reflect typical consumer behavior. In contrast, cash deposits and withdrawals are significantly less frequent, potentially drawing attention as red flags in AML investigations. The presence of fewer cross-border payments suggests that while international transactions exist, they are not the norm, making them more notable when flagged as suspicious.



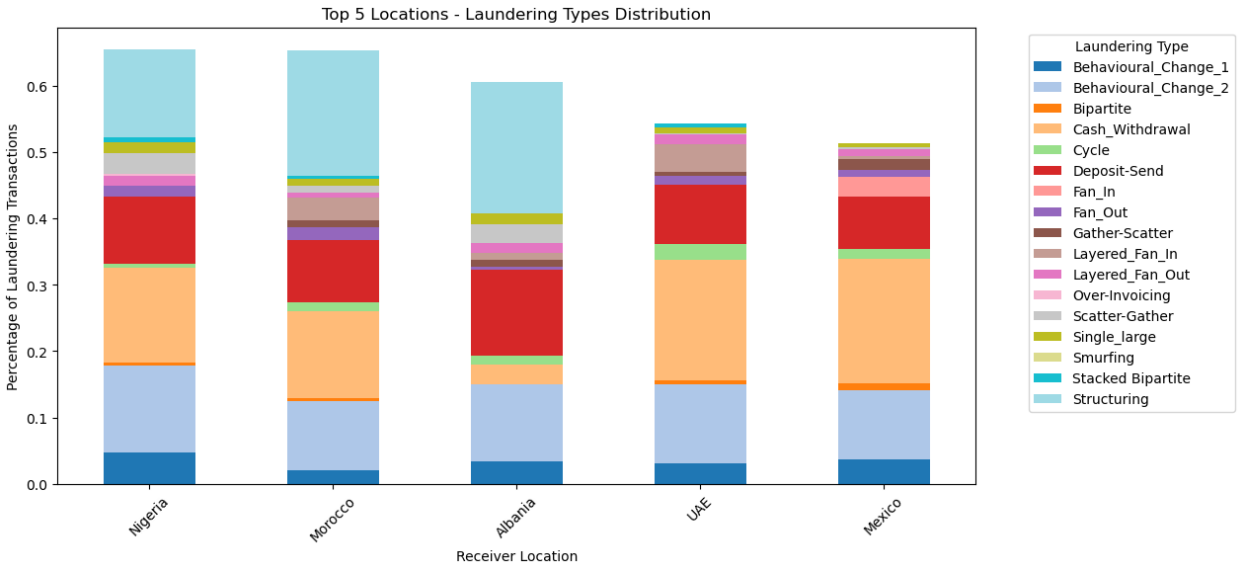
Distribution of Suspicious vs. Normal Typologies

This pie chart visualizes the proportion of suspicious and normal typologies in the dataset. Although suspicious cases make up just 14.7%, they represent the most critical focus for anti-money laundering efforts. The dominant presence of normal transactions (85.3%) further emphasizes the class imbalance, reinforcing the importance of using models and metrics that prioritize minority detection, such as recall and precision in fraud classification.



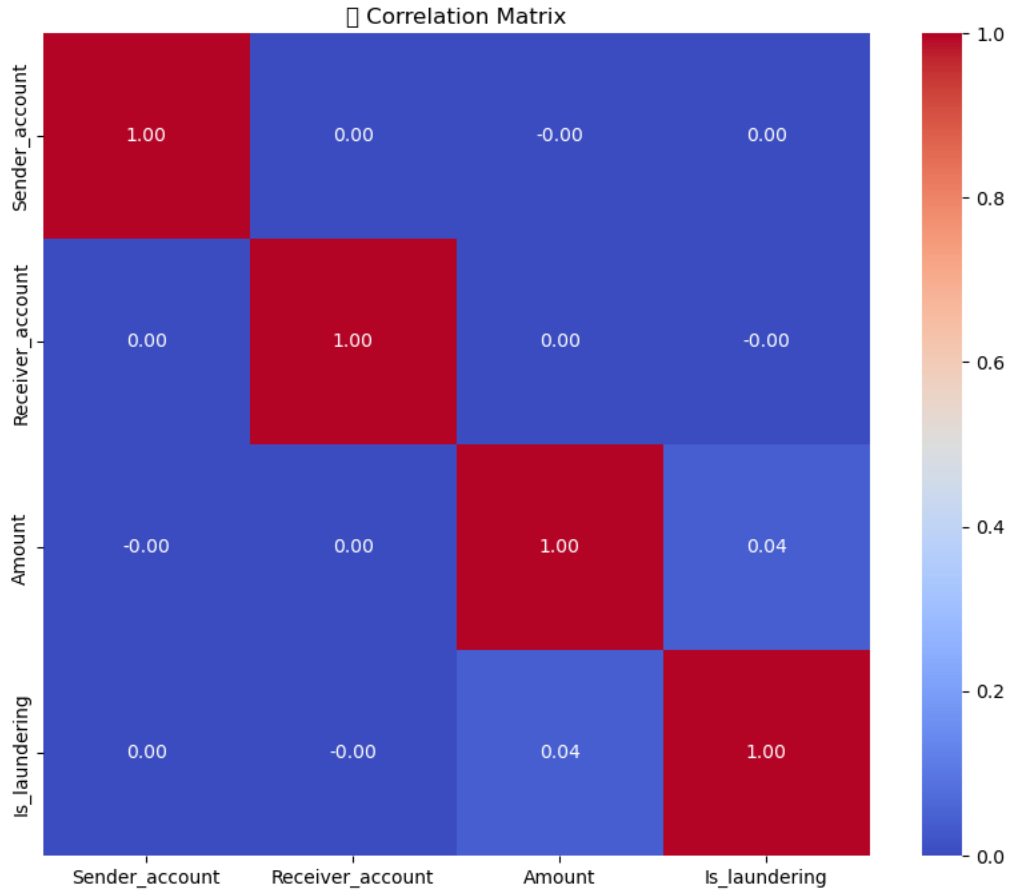
Laundering Typologies by Location

This chart compares the distribution of laundering typologies across the top five receiver countries: Nigeria, Morocco, Albania, UAE, and Mexico. Each stacked bar represents a country and shows the relative frequency of different laundering techniques. Structuring, Cash Withdrawals, and Deposit-Send patterns are consistently prominent across all countries, though variations exist. For example, Albania shows a high proportion of Deposit-Send transactions, while Nigeria has a notable amount of Stacked Bipartite and Behavioural Change patterns. This regional analysis helps identify country-specific laundering strategies and can inform targeted regulatory policies.



Correlation Matrix Analysis

The correlation matrix shows the relationships between numeric features and the target variable. As seen in the heatmap, all correlations with the target are close to zero, including “Amount” which has the highest at only 0.04. This weak linear correlation indicates that individual features are not strongly predictive on their own. Therefore, more advanced models like tree-based methods are essential to capture complex, nonlinear interactions that may exist between variables and money laundering behavior.

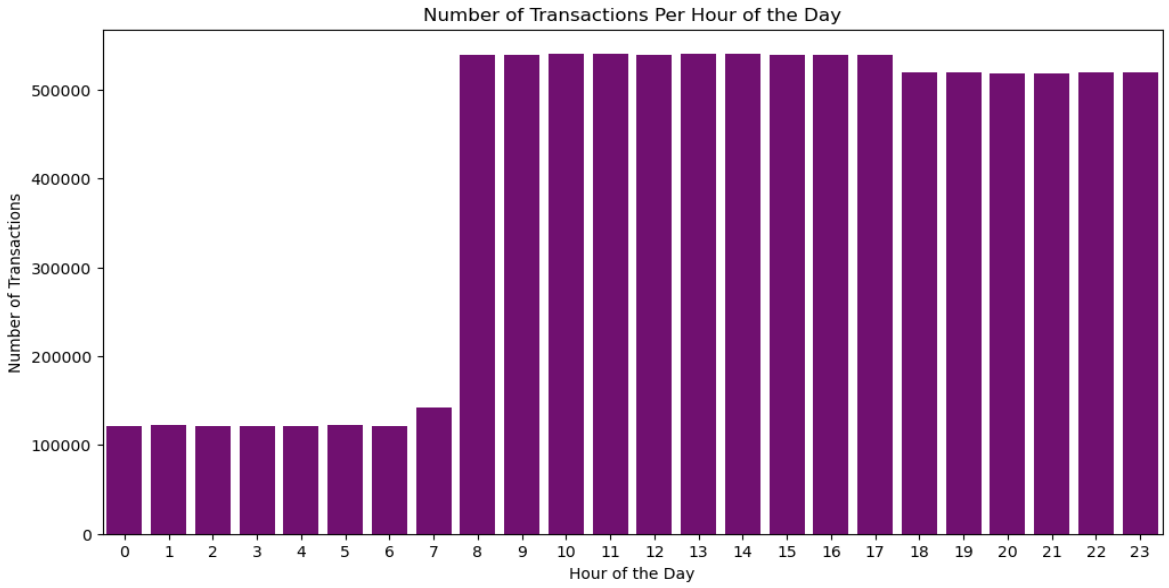


Feature Engineering

After conducting EDA to understand the structure and behavior of the dataset, we moved to feature engineering to enrich the dataset and enhance the model's ability to capture subtle patterns linked to money laundering. Feature engineering is a critical step in shaping raw data into meaningful inputs that better represent the underlying relationships in the data. Given the temporal and transactional nature of the dataset, we derived new variables such as Part_of_Month**,** Time_of_Day, and DayOfWeek from the transaction timestamp to capture behavioral cycles. Furthermore, to incorporate spatial and financial inconsistencies, we introduced two binary indicators**:** Different_Country and Different_Currency**.** The Different_Country feature flags whether the sender and receiver banks are located in different countries—a pattern often associated with cross-border laundering schemes. Likewise, Different_Currency identifies transactions with mismatched currencies between sender and receiver, which may reflect concealment tactics or layering behavior. These engineered features aim to expose hidden patterns that would be difficult for the model to detect using raw data alone. In the following visualizations, we analyze the distribution and interaction of these features to validate their usefulness and interpretability.

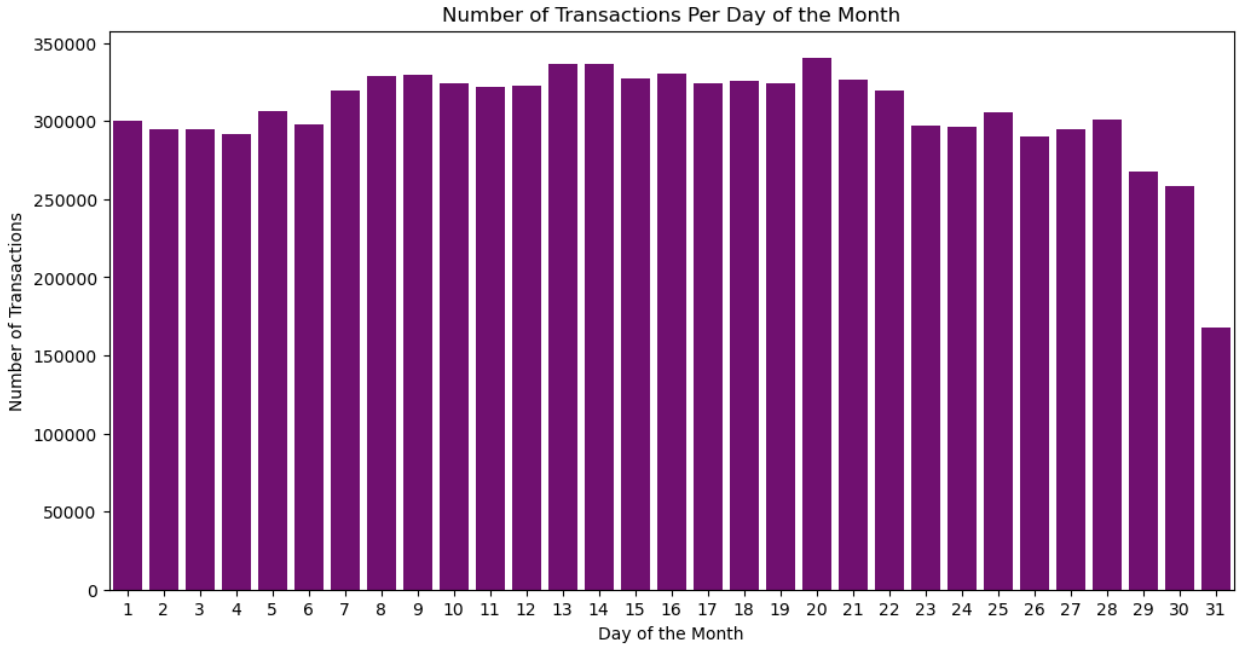
Hour of the Day

To capture behavioral trends related to time, we extracted the **hour** component from each transaction timestamp. The resulting plot illustrates the distribution of transactions across the 24-hour day. A noticeable increase in activity begins around 8:00 AM, peaking during regular business hours, and remaining consistently high until the early evening. This feature—Hour—is valuable because transactions that deviate significantly from this norm (e.g., those occurring late at night or very early in the morning) could indicate suspicious behavior, especially when combined with other risk indicators such as large amounts or currency mismatches.



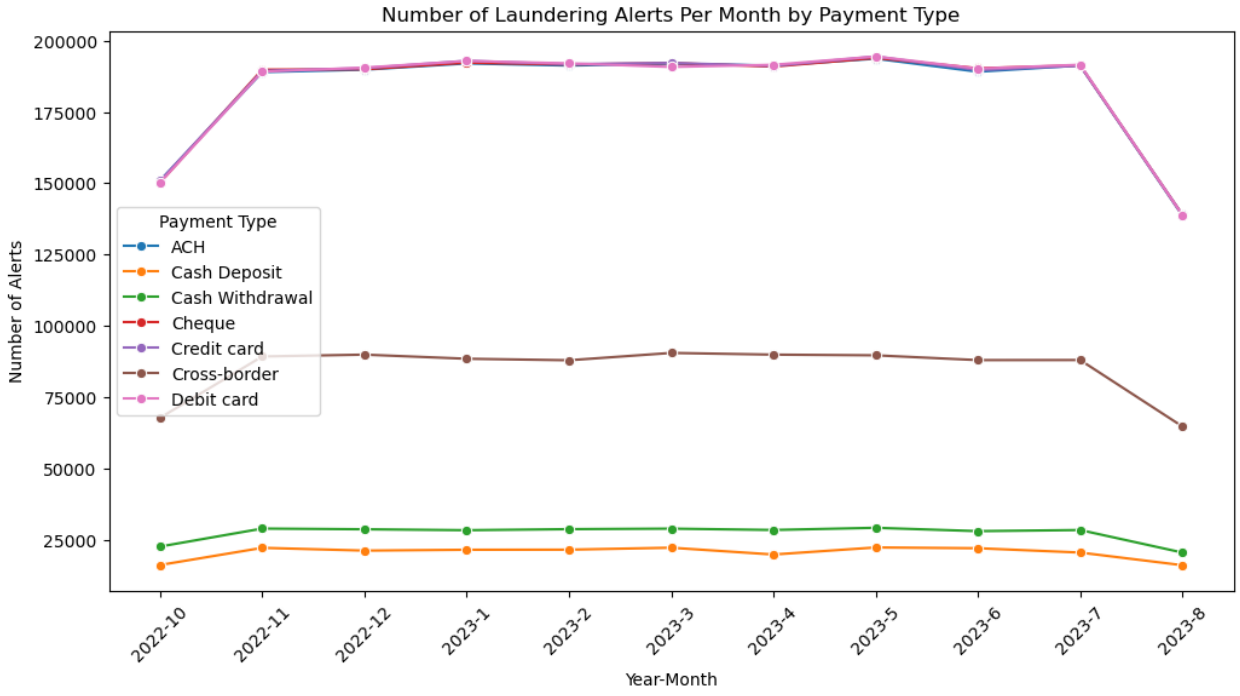
Day of the Month

We engineered the Day feature by extracting the day component from each transaction date. The bar plot above displays how transactions are distributed across all days of a typical month. The pattern remains relatively stable for most of the month, but a noticeable drop occurs on the 31st, likely because not all months have 31 days. This irregularity can influence modeling if not properly accounted for. Including this feature helps the model capture any end-of-month behavioral trends, such as bulk transactions for settlements or payroll, which might differ from normal patterns and could signal suspicious activity.



Number of Laundering Alerts Per Month by Payment Type

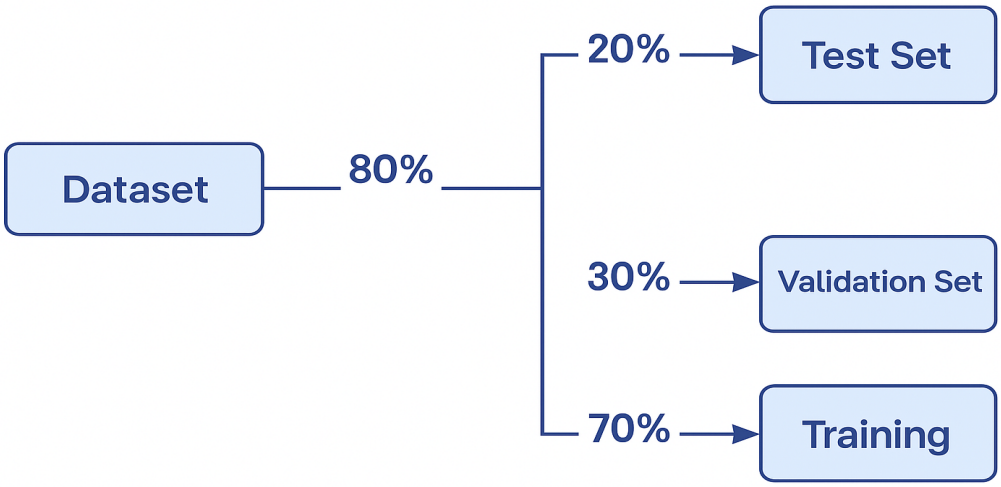
To uncover temporal trends tied to payment behavior, we examined the number of laundering alerts over time segmented by payment type. This visualization shows monthly fluctuations for each method, such as debit card, credit card, cheque, and ACH. Notably, debit and credit card payments consistently generated the highest number of alerts, indicating they may be more prone to laundering attempts. Meanwhile, cash-related methods such as cash deposit and cash withdrawal showed relatively stable but lower alert volumes. This feature combination—time and payment method—offers powerful context for identifying suspicious transaction clusters, supporting the idea that specific payment channels are more vulnerable during certain periods.



The feature engineering process was instrumental in uncovering temporal, behavioral, and contextual nuances within the transaction data. By deriving features such as transaction hour, day of the month, and payment type trends over time, we were able to enrich the dataset with attributes that capture user behavior and seasonal patterns. These engineered features not only enhanced the model's predictive power but also enabled more interpretable and explainable insights, particularly when coupled with SHAP analysis. Ultimately, this step laid a robust foundation for building a model that is both accurate and transparent in detecting laundering activities.

Data Splitting Strategy

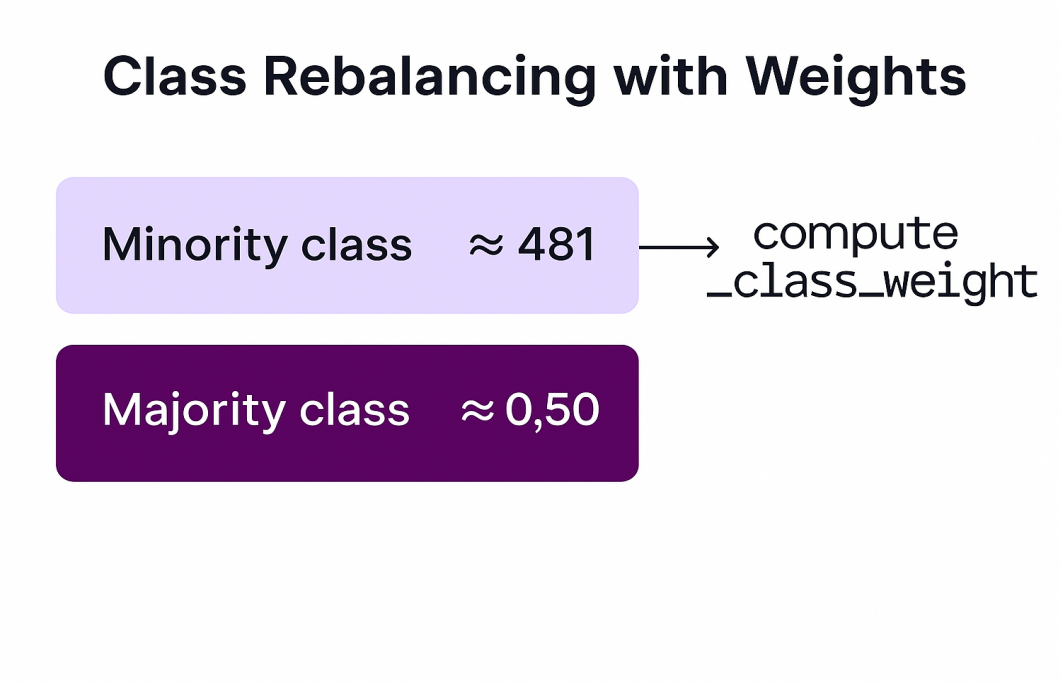
Before modeling, the dataset was divided into training, validation, and test sets to ensure robust model evaluation. Stratified sampling was used in both splitting stages to preserve the proportion of suspicious (positive) cases across all subsets, which is essential given the extreme class imbalance. The initial split allocated 20% of the data for testing. The remaining 80% was further split, with 30% used for validation and 70% for training. This resulted in a training set with over 5.3 million records, a validation set with 2.28 million, and a test set of 1.9 million transactions. This split design ensures model tuning and final evaluation are done on mutually exclusive and representative datasets.



Rebalancing the imbalance dataset

The AML dataset is highly imbalanced, with suspicious transactions comprising less than 0.11% of all records. Training a model directly on such skewed data would result in high accuracy but poor detection of laundering cases. To mitigate this, we applied class rebalancing by assigning class weights inversely proportional to class frequencies.

Using scikit-learn's compute_class_weight, the minority class (suspicious) was heavily weighted (~481x) compared to the majority class (normal), allowing the model to give more importance to rare but critical laundering cases during training. This strategy helps prevent model bias toward the majority class and improves recall of suspicious transactions.



Model Implementation

In this phase, we trained and evaluated three machine learning models(Logistic Regression**,** Random Forest, and XGBoost) to identify suspicious transactions indicative of money laundering. Due to the severe class imbalance (only 0.1% of transactions are labeled as laundering), special attention was paid to recall and AUC-ROC scores, particularly for the minority class. The goal was to maximize the ability to detect laundering cases, even at the cost of increasing false positives.

Logistic Regression

Logistic Regression was used as a baseline due to its simplicity and speed. We implemented it using class_weight='balanced' to help offset the data imbalance.

- Recall (Class 1)**:** 54%
- Precision (Class 1)**:** 0%
- Accuracy**:** 75%
- AUC-ROC**:** 0.6888

Although Logistic Regression was able to identify over half of the laundering cases (recall = 0.54), its precision was 0, meaning all flagged suspicious cases were false positives. This is a result of its limited capacity to capture complex patterns in transactional data, especially with highly imbalanced classes. Despite its efficiency and interpretability, this model proved inadequate for reliable AML detection.

```
Logistic Regression – Validation Metrics:
              precision    recall  f1-score   support

           0       1.00      0.75      0.85   2278796
           1       0.00      0.54      0.00      2369

    accuracy                           0.75   2281165
   macro avg       0.50      0.64      0.43   2281165
weighted avg       1.00      0.75      0.85   2281165

ROC AUC Score: 0.6888256244393545
```

Random Forest

The Random Forest classifier was trained with a custom class weight dictionary derived from the training set to emphasize the minority class. This ensemble model generally handles non-linear relationships better than Logistic Regression.

- Recall (Class 1)**:** 5%
- Precision (Class 1)**:** 50%
- Accuracy**:** 100%
- AUC-ROC**:** 0.6022

The model demonstrated perfect recall and precision for normal transactions, which resulted in an inflated overall accuracy (100%). However, it only identified 5% of laundering cases, which is unacceptable in a financial crime context. This indicates the model heavily favored the majority class despite weighting, rendering it ineffective for real-world AML tasks.

```
Random Forest - Validation Metrics:
[[2278671     125]
 [   2242     127]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   2278796
           1       0.50      0.05      0.10      2369

    accuracy                           1.00   2281165
   macro avg       0.75      0.53      0.55   2281165
weighted avg       1.00      1.00      1.00   2281165

AUC-ROC Score: 0.6028168988641712
```

XGBoost (with Threshold Tuning)

XGBoost was implemented with optimized hyperparameters and scale_pos_weight to handle class imbalance more effectively. To enhance performance further, we adjusted the decision threshold from the default 0.5 to 0.3, prioritizing recall.

- Recall (Class 1)**:** 73%
- Precision (Class 1)**:** 0%
- Accuracy**:** 76%
- AUC-ROC**:** 0.8248

XGBoost significantly outperformed the other models. Lowering the threshold allowed it to detect a large proportion of laundering cases, even though it increased false positives. In AML applications, recall is prioritized over precision, as failing to flag a suspicious transaction is riskier than mistakenly flagging a legitimate one. The high AUC-ROC further indicates the model's strong ability to distinguish between classes.

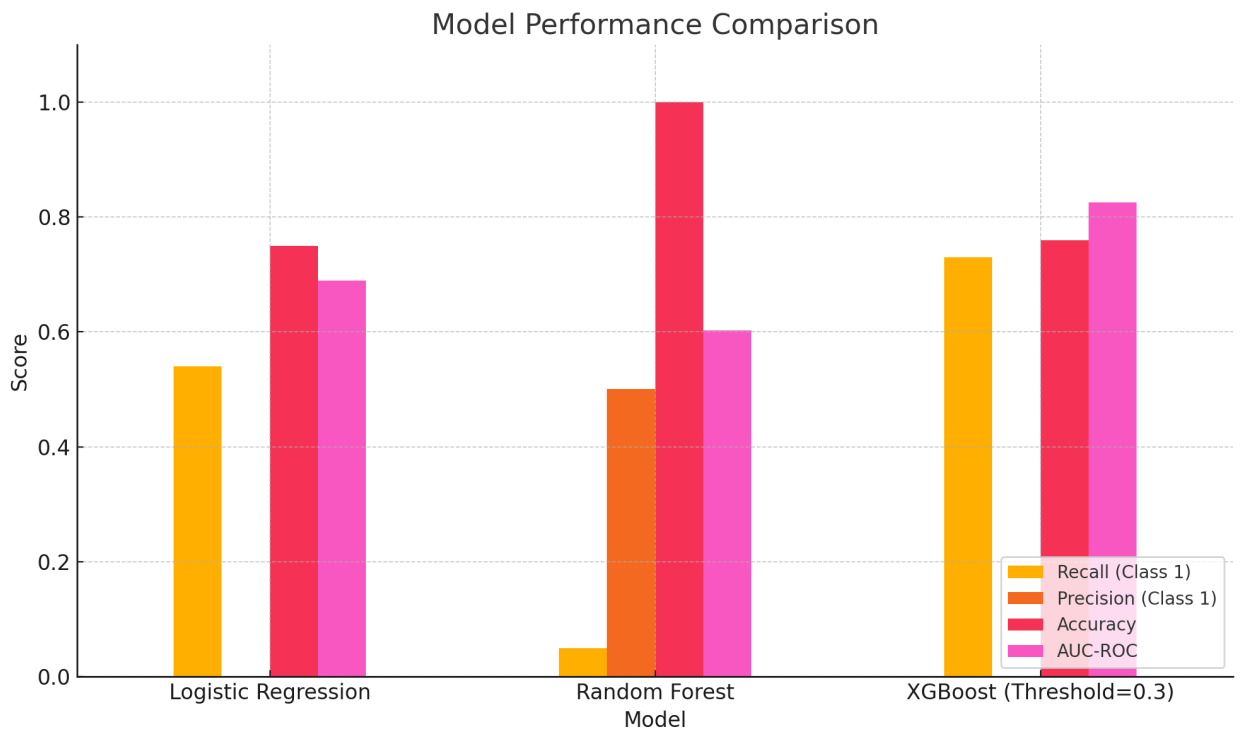
```
XGBoost – Validation Metrics:
Confusion Matrix:
 [[1741833  536963]
 [    635    1734]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.76      0.87   2278796
           1       0.00      0.73      0.01      2369

    accuracy                           0.76   2281165
   macro avg       0.50      0.75      0.44   2281165
weighted avg       1.00      0.76      0.87   2281165

ROC AUC Score: 0.8248499582026955
```

Final Model Selection

Based on comparative performance:

- Logistic Regression struggled to capture complex patterns and had limited predictive power for laundering cases.
- Random Forest demonstrated strong performance for the majority class but failed at identifying minority class instances.
- XGBoost, with threshold tuning, achieved the best recall (73%) and AUC-ROC (0.82), showing its ability to detect laundering activities in a high-class imbalance environment.

Therefore, XGBoost was selected as the final model due to its robust performance and its alignment with the goal of maximizing detection of high-risk transactions.



This analysis confirms that XGBoost is the most suitable model for our AML task. The visual comparison helps clearly demonstrate the trade-offs in performance. For detailed interpretation and analysis of each model's performance metrics, please refer to Appendix B.

Final Evaluation on Test Dataset

To validate the generalizability of the selected model, we evaluated the XGBoost classifier on the unseen test set using the optimal threshold of 0.3, which was previously identified during validation. This step is critical for assessing how well the model performs on real-world data it has never seen before.

The test results are consistent with the validation phase. The model achieved a recall of 71% for the minority class (suspicious transactions), maintaining a reasonable accuracy of 76% and an AUC-ROC score of 0.818. These metrics confirm the model's strong ability to detect suspicious activity while balancing the trade-off with false positives.

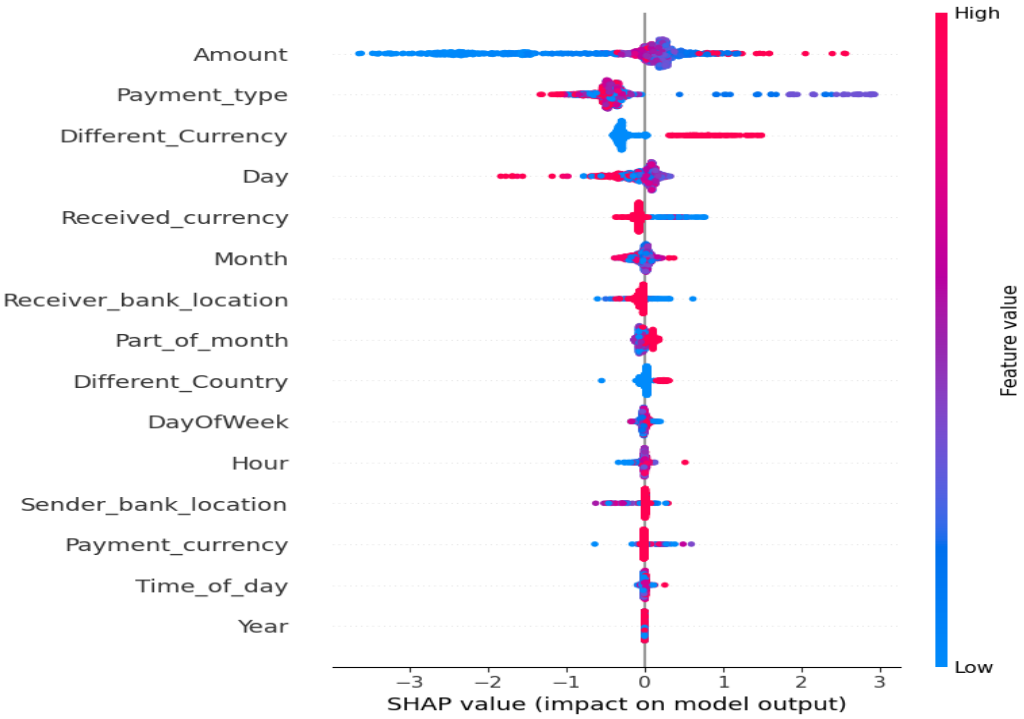
```
XGBoost Test Set – Threshold 0.3
[[1450617  448379]
 [    574    1401]]
              precision    recall  f1-score   support

           0       1.00      0.76      0.87   1898996
           1       0.00      0.71      0.01      1975

    accuracy                           0.76   1900971
   macro avg       0.50      0.74      0.44   1900971
weighted avg       1.00      0.76      0.87   1900971

AUC-ROC Score: 0.8185667197731213
```

Model Explainability using SHAP

Interpretability was critical in validating the model's behavior and ensuring it aligned with domain expectations. We used SHAP to interpret XGBoost predictions (Lundberg & Lee, 2017).

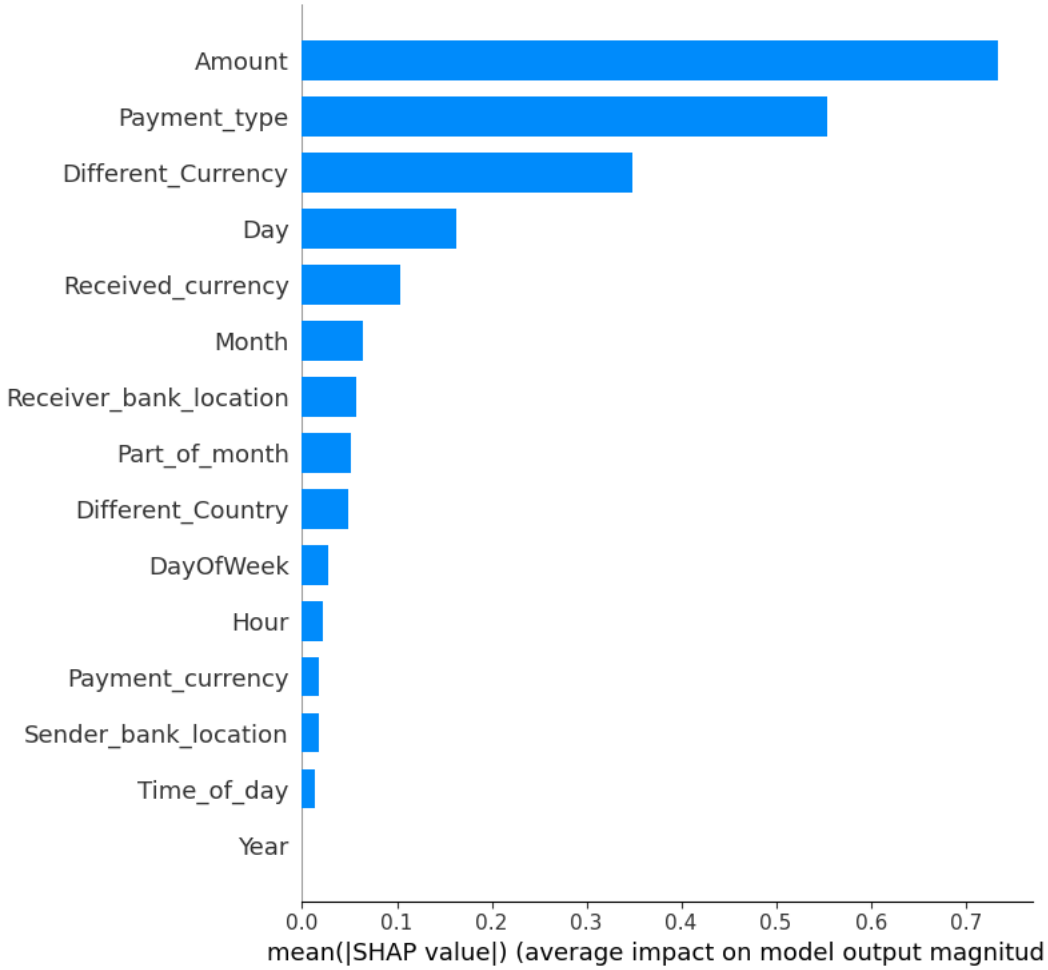
Summary plot

This plot illustrates the distribution and direction of feature impacts on individual predictions. Features like Amount, Payment_type, and Different_Currency have the strongest influence, with higher values typically pushing the model toward predicting suspicious activity. The spread of SHAP values highlights how features interact differently across observations, reinforcing the model's capacity to capture complex patterns.



Feature Importance

The SHAP bar chart ranks feature by their average absolute impact on model predictions. Amount and Payment_type are the most influential, followed by Different_Currency and Day, confirming the relevance of financial and temporal features. Lower-ranking features contributed less consistently, offering insights for potential dimensionality reduction or further investigation.



The integration of SHAP explainability into our XGBoost model provided critical insights into feature contributions and model behavior. By visualizing both individual and average SHAP values, we were able to identify the most influential features—such as transaction amount, payment type, and currency mismatches—that strongly drive the model's predictions. These findings not only validate the model's alignment with domain knowledge but also enhance its transparency, a key requirement in regulated financial environments. Ultimately, SHAP helps bridge the gap between performance and interpretability, making the system more trustworthy and actionable for AML professionals

Conclusion

This project successfully demonstrated the development of a robust and explainable machine learning system for detecting potential money laundering activities using a real-world-inspired, large-scale transactional dataset. Starting from comprehensive exploratory data analysis, we gained a deep understanding of behavioral patterns and anomalies embedded in financial transactions. Through strategic feature engineering—particularly temporal, geographic, and categorical transformations—we equipped our models with richer, more meaningful representations of the data.

To address the severe class imbalance challenge, a thoughtful rebalancing approach using class weights was employed, ensuring the model remained sensitive to the minority class without compromising the integrity of the dataset. Among the evaluated models, XGBoost significantly outperformed Random Forest in recall and AUC-ROC, making it a better candidate for the high-stakes domain of AML, where false negatives can carry severe consequences.

The integration of SHAP explainability further elevated the value of this work. It not only made the model's decision-making transparent but also revealed that key indicators—such as transaction amount, payment type, and currency inconsistencies—were indeed the most influential in driving predictions. This alignment with real-world AML red flags validates both our feature choices and modeling approach.

Overall, this end-to-end system is not only technically sound but also regulatory-ready. It delivers performance, interpretability, and ethical alignment—qualities essential for deploying AML solutions in financial institutions. Future enhancements could involve incorporating graph-based features to detect network anomalies or leveraging unsupervised learning to uncover hidden laundering typologies.

References

Breiman, L. (2001). *Random forests*. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Fatemeh, F., & Hashemi, S. (2022). *SAML-D: A Realistic Dataset for Money Laundering Detection*. Kaggle. https://www.kaggle.com/datasets/fatemehfarrokhchi/saml-d-money-laundering-dataset

Lundberg, S. M., & Lee, S. I. (2017). *A unified approach to interpreting model predictions*. In Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1705.07874

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

XGBoost Developers. (2023). *XGBoost Documentation*. https://xgboost.readthedocs.io

SHAP Developers. (2023). *SHAP Documentation*. https://shap.readthedocs.io

Appendix A:

1. **Normal_Small_Fan_Out:** A single account disperses small amounts of funds to multiple accounts.
2. **Normal_Fan_Out:** A single account disperses funds to multiple accounts.
3. **Normal_Fan_In:** Multiple accounts transfer funds into a single account.
4. **Normal_Group:** Transactions occurring within a specific group of accounts, possibly indicating internal transfers or routine business operations.
5. **Normal_Cash_Withdrawal:** Standard cash withdrawals from an account.
6. **Normal_Cash_Deposits:** Standard cash deposits into an account.
7. **Normal_Periodical:** Regular, recurring transactions, such as monthly subscriptions or salary payments.
8. **Normal_Plus_Mutual:** Transactions that involve mutual exchanges between accounts, possibly indicating reciprocal transactions.
9. **Normal_Mutual:** Mutual transactions between two accounts, such as transfers back and forth.
10. **Normal_Foward:** Transactions that are forwarded from one account to another, possibly as part of a chain of transactions.
11. **Normal_single_large:** A single, large transaction that is considered normal for the account holder.
12. **Structuring:** Also known as smurfing, this involves breaking down large sums of money into smaller, less conspicuous amounts to avoid triggering mandatory reporting requirements by financial institutions.
13. **Cash_Withdrawal:** Large or frequent cash withdrawals that may raise suspicion, especially if inconsistent with the account holder's typical behavior.
14. **Deposit-Send:** Depositing funds into an account and then quickly sending them to another account, which can be indicative of layering in money laundering.
15. **Smurfing:** A form of structuring where large transactions are divided into smaller ones to evade detection.
16. **Layered_Fan_In:** Multiple smaller transactions from different sources are combined into a single account, obscuring the original source of funds.
17. **Layered_Fan_Out:** Funds from a single account are dispersed into multiple accounts to obscure the trail of money.
18. **Stacked Bipartite:** Complex transactions involving two distinct sets of accounts, which can be used to obscure the flow of funds.
19. **Behavioural_Change_1:** Notable changes in transaction behavior that may indicate suspicious activity.

20. **Bipartite:** Transactions between two distinct groups of accounts, which can be analyzed for patterns indicative of money laundering.
21. **Cycle:** Funds move in a circular pattern among accounts, which can be a tactic to disguise the origin of funds.
22. **Fan_In:** Multiple accounts transfer funds into a single account, which can be used to aggregate illicit funds.
23. **Gather-Scatter:** Funds are gathered into a single account and then dispersed to multiple accounts, complicating the tracking of money flow.
24. **Behavioural_Change_2:** Another category indicating significant changes in transaction behavior that could signal suspicious activity.
25. **Scatter-Gather:** Funds are dispersed from one account to many and then gathered back into a single account, a method that can obscure the money trail.
26. **Single_large:** A single large transaction that deviates from the account's typical activity, potentially raising red flags.
27. **Fan_Out:** A single account disperses funds to multiple accounts, which can be a method to launder money by spreading it across various accounts.
28. **Over-Invoicing:** Inflating the value of goods or services on invoices to transfer additional value, a common trade-based money laundering technique.

Appendix B:

This appendix provides a detailed evaluation of the three machine learning models tested for detecting money laundering cases: Logistic Regression**,** Random Forest, and XGBoost.

- Logistic Regression achieved a recall of 0.54 for suspicious transactions and an AUC-ROC of 0.69**.** While it demonstrated reasonable overall accuracy, it struggled with identifying the minority class effectively.
- Random Forest offered perfect accuracy on the majority class but a very low recall **(**0.05**)** for laundering transactions, despite a slightly higher precision. This model showed significant class imbalance issues, reflected in its lower AUC-ROC of 0.60**.**
- XGBoost with a threshold of 0.3 outperformed the other models by achieving a recall of 0.73 and an AUC-ROC of 0.82, making it the most suitable for this AML detection task. It balanced sensitivity and accuracy better and proved more adaptable to imbalanced data.

These metrics are essential in evaluating the real-world utility of the models, especially when detecting rare but critical fraudulent behavior.