**Suspicious transaction detection system**

### **1. Introduction**

Money laundering remains a persistent global challenge, requiring advanced and effective transaction monitoring techniques. Traditional Anti-Money Laundering (AML) methods often struggle to keep pace with evolving fraud tactics, leading to vulnerabilities in financial institutions and the risk of significant financial losses. This project aims to develop a robust machine learning model capable of identifying suspicious transactions, potentially linked to money laundering activities, by analyzing historical transaction data and customer behavior patterns.

### **2. Dataset Overview**

The SAML-D dataset used in this project contains over 9 million transaction records, with only 0.1039% labeled as suspicious, reflecting real-world class imbalance challenges. It includes 12 features and 28 transaction typologies (11 normal and 17 suspicious). These typologies simulate complex transaction flows using 15 graphical network structures to increase detection difficulty.

Key features include:

* **Time and Date**: Tracks transaction chronology.
* **Sender/Receiver Details**: Identifies behavior patterns and networks.
* **Amount and Payment Type**: Highlights unusual values and methods.
* **Location and Currency**: Adds complexity with high-risk regions and mismatches.
* **Is Suspicious**: Binary label for transaction classification.

Data from October 2022 to May 2023 is used for training, while June to August 2023 is reserved for validation.

### **3. Data Preprocessing**

**Feature Engineering:** Account numbers (Sender\_account and Receiver\_account) were deemed non-informative for laundering detection and dropped from the dataset.

Date and time columns were initially explored, but as their significance was minimal, they were not directly included as features.

**Feature Extraction:** Categorical variables such as Payment\_currency, Received\_currency, Sender\_bank\_location, Receiver\_bank\_location, and Payment\_type were identified for encoding.

**Data Transformation:** One-hot encoding was applied to categorical variables, ensuring the transformation did not introduce multicollinearity by dropping the first category for each feature.

**Data Balancing Techniques:** The target variable Is\_laundering exhibited class imbalance, with significantly fewer fraudulent transactions than legitimate ones so we undersample the majority class.

#### **Other Methods Used :** The class\_weight='balanced' parameter in the RandomForestClassifier further addressed class imbalance by adjusting the weight assigned to each class during model training.

### **4. Exploratory Data Analysis (EDA)**

Visit our [Analytics website](https://atlantic-vault-a4e.notion.site/Analytics-17173084af3d804188abc43f9161c2c2) EDA.

### **5. Model Development and Performance**

**Models Considered:**

* **Random Forest Classifier:** Selected for its robustness, ability to handle class imbalance using class\_weight='balanced', and interpretability through feature importance scores.
* **HistGradientBoostingClassifier:** Evaluated for its efficiency on large datasets and inherent handling of categorical data with faster training times.

**Performance Evaluation**:

* The **Random Forest Classifier** achieved an accuracy of **79.0%** on the test set, with balanced precision and recall, especially notable for its improved handling of class imbalance compared to baseline methods.
* The **HistGradientBoostingClassifier**, even with SMOTE applied, yielded a higher overall accuracy of **90.9%**. However, it showed a significant overfitting tendency and poor generalization on minority class (Is\_laundering=1), with a very low precision of **1%** despite high recall.

**Conclusion**:

* **Random Forest Classifier** outperformed in terms of balanced precision, recall, and overall stability. Its confusion matrix demonstrated better fraud detection capabilities without an overwhelming bias towards majority class predictions, making it more suitable for this fraud detection problem.

### **6. Fraud Detection System**

### **System Overview**

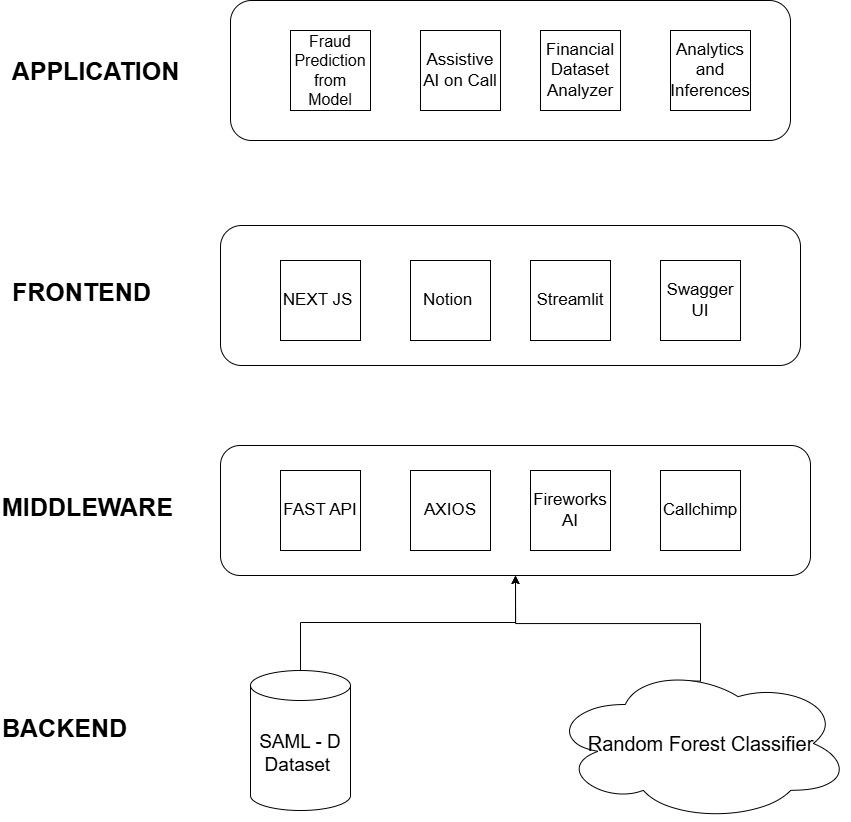
### The Fraud Detection System is an AI-driven application providing advanced tools for identifying and preventing fraudulent transactions. Key features include:

1. **Fraud Detection and Prediction:** Uses a Random Forest Classifier to predict potentially fraudulent transactions by analyzing various financial attributes.
2. **AI Assistive Calling:** A voice-based AI alerts users about suspicious transactions, allowing them to confirm or deny their authenticity in real time.
3. **Financial Data Analyzer:** Analyzes transaction data to detect anomalies and trends, providing insights for improved risk management.
4. **Analytics Dashboard:** Displays key fraud trends, patterns, and system performance metrics with interactive graphs and detailed reports.

This comprehensive solution enhances financial security and user engagement with intuitive and actionable fraud prevention tools.

### **8. Deployment**

* **Architecture**



The application follows a microservice architecture, integrating multiple components for seamless performance. The frontend is built using Next.js for dynamic rendering, while Streamlit deploys the chatbot interface. FastAPI handles backend API services, and Axios manages API requests from the frontend.

### **API Integration**

* **FastAPI**: Provides efficient endpoints for fraud detection predictions.
* **Axios**: Facilitates communication between the frontend and backend services.
* **Fireworks AI**: Powers the chatbot for customer engagement and fraud notifications.
* **Callchimp**: Manages AI-driven assistive voice calls for user alerts.

### **10. Conclusion**

### The fraud detection system effectively identifies fraudulent transactions, offers AI-assisted calling for customer alerts, and provides financial data analytics. Using technologies like FastAPI, Fireworks AI, and Streamlit, the solution addresses key challenges, though further improvements in scalability and model adaptability are needed.

### **Future Development**

1. Real-time learning for dynamic fraud pattern adaptation.
2. Enhanced scalability for high-volume transactions.
3. Advanced analytics for deeper insights.
4. Stronger security measures for better data privacy.
5. Multi-currency and multi-language support for global reach.

### **11. References**

[**Dataset :www.kaggle.com/datasets/berkanoztas/synthetic-transaction-monitoring-dataset-aml/data**](https://www.kaggle.com/datasets/berkanoztas/synthetic-transaction-monitoring-dataset-aml/data)