**Model Development Document (MDD) / Technical Specification**

**Project Title:** AML Machine Learning Model Implementation for Financial Crime Detection

**Version:** 1.0

**Date:** June 01, 2025

**Prepared By:** Harshal Patil

Contents

[1. Introduction 4](#_Toc199664015)

[1.1 Purpose of the Document 4](#_Toc199664016)

[1.2 Project Overview 4](#_Toc199664017)

[1.3 Scope of the Model 4](#_Toc199664018)

[2. Business Context and Objectives 5](#_Toc199664019)

[2.1 Business Problem 5](#_Toc199664020)

[2.2 Model Objectives 5](#_Toc199664021)

[2.3 Stakeholders 5](#_Toc199664022)

[3. Data Management 6](#_Toc199664023)

[3.1 Data Sources 6](#_Toc199664024)

[3.2 Data Acquisition and Ingestion 6](#_Toc199664025)

[3.3 Data Quality and Preprocessing 6](#_Toc199664026)

[3.4 Data Storage 6](#_Toc199664027)

[4. Model Architecture and Methodology 7](#_Toc199664028)

[4.1 Overall System Architecture 7](#_Toc199664029)

[4.2 Sanctions Screening Model 7](#_Toc199664030)

[4.2.1 Objective 7](#_Toc199664031)

[4.2.2 Algorithm Selection 7](#_Toc199664032)

[4.2.3 Feature Engineering Details 7](#_Toc199664033)

[4.2.4 Training Methodology 8](#_Toc199664034)

[4.2.5 Hyperparameters (Conceptual) 8](#_Toc199664035)

[4.3 AML Transaction Monitoring Model 8](#_Toc199664036)

[4.3.1 Objective 8](#_Toc199664037)

[4.3.2 Algorithm Selection 8](#_Toc199664038)

[4.3.3 Feature Engineering Details 8](#_Toc199664039)

[4.3.4 Training Methodology 9](#_Toc199664040)

[4.3.5 Hyperparameters (Conceptual) 9](#_Toc199664041)

[4.4 Integrated Risk Scoring and Alert Generation 10](#_Toc199664042)

[4.4.1 Objective 10](#_Toc199664043)

[4.4.2 Methodology 10](#_Toc199664044)

[4.4.3 Alert Thresholds 10](#_Toc199664045)

[5. Model Training and Validation 11](#_Toc199664046)

[5.1 Training Environment 11](#_Toc199664047)

[5.2 Data Splitting Strategy 11](#_Toc199664048)

[5.3 Performance Metrics 11](#_Toc199664049)

[5.4 Backtesting Results 12](#_Toc199664050)

[5.5 Stress Testing Results 12](#_Toc199664051)

[6. Model Explainability and Interpretability 14](#_Toc199664052)

[6.1 Global Feature Importance 14](#_Toc199664053)

[6.2 Local Explainability (Conceptual) 14](#_Toc199664054)

[6.3 Transparency Considerations 14](#_Toc199664055)

[7. Model Monitoring and Maintenance 15](#_Toc199664056)

[7.1 Performance Monitoring 15](#_Toc199664057)

[7.2 Data and Concept Drift Detection 15](#_Toc199664058)

[7.3 Retraining Strategy and Frequency 15](#_Toc199664059)

[7.4 Version Control (Conceptual) 15](#_Toc199664060)

[8. Regulatory Compliance Considerations 16](#_Toc199664061)

[8.1 Model Governance Framework 16](#_Toc199664062)

[8.2 Auditability and Reproducibility 16](#_Toc199664063)

[8.3 Bias and Fairness (Conceptual) 16](#_Toc199664064)

[9. Future Enhancements / Roadmap 17](#_Toc199664065)

[10. Appendices (Conceptual) 18](#_Toc199664066)

[10.1 Data Dictionary 18](#_Toc199664067)

[10.2 Code Snippets 18](#_Toc199664068)

# 1. Introduction

## 1.1 Purpose of the Document

This document serves as the Model Development Document (MDD) and Technical Specification for the Anti-Money Laundering (AML) Machine Learning (ML) model implementation. Its primary purpose is to provide a comprehensive record of the model's design, development, training, validation, and operational considerations. It aims to ensure transparency, reproducibility, and compliance with internal governance standards and external regulatory requirements.

## 1.2 Project Overview

This project focuses on developing a robust and intelligent system for detecting and preventing financial crime, specifically targeting money laundering and sanctions evasion. It leverages machine learning techniques to automate and enhance the traditional rule-based AML processes, aiming to improve detection rates, reduce false positives, and increase operational efficiency for financial institutions.

## 1.3 Scope of the Model

The scope of this ML model implementation includes:

* **Sanctions Screening:** Identifying potential matches between customer profiles and known sanctions lists.
* **AML Transaction Monitoring:** Detecting anomalous and potentially suspicious transaction patterns.
* **Integrated Risk Scoring:** Combining insights from both screening and monitoring to generate a holistic risk assessment.
* **Foundational MLOps Concepts:** Addressing aspects of model validation (backtesting, stress testing), monitoring, and explainability.

# 2. Business Context and Objectives

## 2.1 Business Problem

Financial institutions face significant challenges in combating money laundering and terrorist financing. Traditional rule-based systems often struggle with high volumes of false positives, leading to inefficient investigations, and may fail to detect sophisticated illicit activities. Non-compliance can result in severe regulatory fines, reputational damage, and enable criminal enterprises.

## 2.2 Model Objectives

The primary objectives of implementing these ML models are to:

* **Enhance Detection Capabilities:** Improve the accuracy and coverage of identifying sanctioned entities and suspicious transaction behaviors.
* **Reduce False Positives:** Minimize the number of benign alerts, thereby optimizing the workload for human AML analysts.
* **Increase Operational Efficiency:** Streamline the alert generation and initial triage process.
* **Provide Explainability:** Offer insights into why specific alerts are generated, aiding analyst investigations and regulatory reporting.
* **Ensure Robustness:** Develop models that perform reliably under various data conditions, including potential degradation or shifts.
* **Support Regulatory Compliance:** Provide a framework for transparent, auditable, and continuously monitored AML processes.

## 2.3 Stakeholders

Key stakeholders for this model include:

* **AML Compliance Officers:** Responsible for overall AML program oversight.
* **Financial Crime Analysts/Investigators:** Users who interact with the model's outputs (alerts).
* **Risk Management:** Overseeing model risk and overall financial crime exposure.
* **Internal Audit:** Ensuring adherence to internal policies and procedures.
* **Regulatory Bodies:** Requiring demonstration of effective AML controls.
* **IT/MLOps Team:** Responsible for model deployment, infrastructure, and maintenance.

# 3. Data Management

## 3.1 Data Sources

The models utilize simulated data representing typical financial institution records:

* **Customer Data:** Contains static information about individual customers (e.g., Customer ID, Name, Address, Date of Birth, Nationality, Country, Industry, Onboarding Date).
* **Sanctions List Data:** A simulated list of sanctioned entities, including names, addresses, dates of birth, nationalities, and sanction types.
* **Transaction Data:** Contains dynamic information about financial transactions (e.g., Transaction ID, Customer ID, Date, Type, Amount, Currency, Sender/Receiver IDs, Sender/Receiver Countries).

## 3.2 Data Acquisition and Ingestion

For this project, data is acquired through simulated generation or loaded from CSV files (customer\_data.csv, UK Sanctions List\_mean.csv). In a production environment, this would involve secure connections to core banking systems, data warehouses, or external sanctions list providers, often via ETL (Extract, Transform, Load) pipelines.

## 3.3 Data Quality and Preprocessing

Data quality is critical for model performance. The following preprocessing steps are applied:

* **Handling Missing Values:** Missing values are typically filled with appropriate defaults (e.g., 0 for numerical features, 'UNKNOWN' or np.nan for categorical).
* **Data Cleaning and Standardization:**
  + Names, addresses, and nationalities are converted to uppercase and stripped of leading/trailing whitespace for consistent matching.
  + Date fields are parsed into a standard datetime format.
  + Customer data columns are standardized to consistent names (e.g., Customer\_ID, Customer\_Name).
* **Feature Engineering (Detailed in Section 4):** Raw data is transformed into meaningful features for each model.
* **Scaling (for AML Model):** Numerical features for the AML model are scaled using StandardScaler to ensure features contribute equally to the model's decision.

## 3.4 Data Storage

For this project, processed and intermediate data are stored as CSV files (e.g., sanctions\_list\_cleaned.csv, final\_sanctions\_screening\_results.csv, final\_aml\_screening\_results.csv). In a production setting, this would involve secure databases (SQL, NoSQL) or data lakes.

# 4. Model Architecture and Methodology

## 4.1 Overall System Architecture

The system consists of two primary ML models operating on different data streams, with an overarching integrated risk scoring layer:

1. **Sanctions Screening Model:** Processes customer static data against sanctions lists.
2. **AML Transaction Monitoring Model:** Analyzes transactional data for anomalous patterns.
3. **Integrated Risk Scoring Engine:** Combines outputs from both models and inherent customer risk to generate a unified risk score and alerts.

## 4.2 Sanctions Screening Model

### 4.2.1 Objective

To identify customers who are highly likely to be on a sanctions list, based on fuzzy matching of identifying attributes.

### 4.2.2 Algorithm Selection

**Gradient Boosting Classifier (sklearn.ensemble.GradientBoostingClassifier)** was chosen.

* **Rationale:** Gradient Boosting is an ensemble method known for its high predictive accuracy, ability to handle complex non-linear relationships, and robustness to outliers. It can effectively learn from the nuanced similarities captured by fuzzy matching scores.

### 4.2.3 Feature Engineering Details

Features are engineered by comparing customer attributes against sanctioned entity attributes:

* **name\_fuzz\_ratio**: Fuzzy string matching ratio for names.
* **name\_token\_sort\_ratio**: Token sort ratio for names (handles word order differences).
* **name\_token\_set\_ratio**: Token set ratio for names (handles missing/extra words).
* **name\_match\_score**: Maximum of the above three name-matching scores.
* **address\_match\_score**: Token set ratio for addresses.
* **dob\_match**: Binary indicator (1 if Date of Birth matches, 0 otherwise).
* **nationality\_match**: Binary indicator (1 if Nationality matches, 0 otherwise).
* **customer\_country\_risk\_score**: Numerical score based on the customer's country risk (e.g., HIGH=10, MEDIUM=5, LOW=1).
* **sanction\_type\_severity\_score**: Numerical score based on the type of sanctioned entity (e.g., INDIVIDUAL=10, ENTITY=8).
* **Interaction Features**:
  + **name\_country\_interaction**: name\_match\_score \* customer\_country\_risk\_score
  + **name\_dob\_interaction**: name\_match\_score \* dob\_match

### 4.2.4 Training Methodology

* **Type:** Supervised Learning (Classification).
* **Data Generation:** A synthetic training dataset is generated by pairing random customers with random sanctioned entities. A small percentage of these pairs are intentionally made to be "true matches" (positive labels) by introducing slight variations to sanctioned names and addresses, while the majority are "non-matches" (negative labels). This simulates the imbalanced nature of real-world data.
* **Target Variable:** is\_sanction\_match (1 for a true match, 0 for a non-match).

### 4.2.5 Hyperparameters (Conceptual)

* n\_estimators: Number of boosting stages (e.g., 100-200).
* learning\_rate: Shrinkage factor applied to each tree (e.g., 0.05-0.1).
* max\_depth: Maximum depth of the individual regression estimators (e.g., 3-5).
* Hyperparameters are typically optimized using techniques like GridSearchCV.

## 4.3 AML Transaction Monitoring Model

### 4.3.1 Objective

To detect unusual or anomalous transaction patterns that deviate significantly from a customer's normal behavior or typical transaction profiles, indicative of potential money laundering.

### 4.3.2 Algorithm Selection

**Isolation Forest (sklearn.ensemble.IsolationForest)** was chosen.

* **Rationale:** Isolation Forest is an unsupervised anomaly detection algorithm well-suited for high-dimensional datasets. It works by isolating anomalies as "points that are few and different" using random trees, making it efficient and effective for identifying outliers without requiring labeled data.

### 4.3.3 Feature Engineering Details

Features are derived from individual transactions and aggregated customer transaction history:

* **Individual Transaction Features:**
  + **Amount\_USD**: Transaction amount converted to USD.
  + **Transaction\_Hour**: Hour of the day the transaction occurred.
  + **Transaction\_DayOfWeek**: Day of the week the transaction occurred.
  + **Sender\_Country\_Risk\_Score**: Risk score of the sender's country.
  + **Receiver\_Country\_Risk\_Score**: Risk score of the receiver's country.
  + **Geographic\_Risk\_Score**: Maximum of sender and receiver country risk scores.
  + **TxType\_DEPOSIT, TxType\_WITHDRAWAL, etc.**: One-hot encoded features for transaction types.
* **Aggregated Customer Features (derived per customer over a period):**
  + **Total\_Amount**: Sum of all transaction amounts for the customer.
  + **Avg\_Amount**: Average transaction amount for the customer.
  + **Num\_Transactions**: Total number of transactions for the customer.
  + **Max\_Amount**: Maximum transaction amount for the customer.
  + **Min\_Amount**: Minimum transaction amount for the customer.
  + **Unique\_Counterparties**: Number of unique counterparties for the customer.
* **Interaction Feature:**
  + **Amount\_Geo\_Risk\_Interaction**: Amount\_USD \* Geographic\_Risk\_Score.

### 4.3.4 Training Methodology

* **Type:** Unsupervised Learning (Anomaly Detection).
* **Data:** Trained on a dataset of predominantly normal transactions.
* **Anomaly Score:** The model outputs a decision\_function score, where lower scores indicate higher likelihood of being an anomaly.
* **Alert Flag:** Transactions with anomaly scores below a certain threshold (determined by the contamination parameter or a post-training analysis) are flagged as 'ALERT'.

### 4.3.5 Hyperparameters

* contamination: The expected proportion of outliers in the dataset (e.g., 0.01). This implicitly sets the threshold for anomaly detection.
* n\_estimators: Number of base estimators (trees) in the ensemble (e.g., 100-200).
* max\_features: Number of features to draw from X to train each base estimator (e.g., 1.0 for all features).

## 4.4 Integrated Risk Scoring and Alert Generation

### 4.4.1 Objective

To provide a single, unified risk score for each customer, combining insights from sanctions screening, transaction monitoring, and inherent customer risk, enabling prioritized alert management.

### 4.4.2 Methodology

A simple **weighted sum** approach is used to calculate the Integrated\_Risk\_Score:

* **Components:**
  + Customer\_Country\_Risk\_Score (from initial customer data).
  + Max\_Sanction\_Match\_Probability (output from Sanctions Screening Model).
  + Mapped\_AML\_Risk (transformed from AML anomaly score, higher for more anomalous transactions).
* **Weights:** Configurable weights are applied to each component (e.g., 0.3 for country risk, 0.4 for sanctions, 0.3 for AML). These weights can be tuned based on business risk appetite and regulatory guidance.
* **Normalization:** The final integrated score is normalized to a 0-100 scale for easier interpretation.
* **Overall Risk Level:** Categorized into 'LOW', 'MEDIUM', 'HIGH' based on score thresholds (e.g., <30 LOW, 30-60 MEDIUM, >60 HIGH).

### 4.4.3 Alert Thresholds

Alerts are generated for customers whose Integrated\_Risk\_Score exceeds a predefined alert\_threshold\_score (e.g., 50). An Alert\_Reason is also generated based on the contributing factors (sanctions match, transaction anomaly, high country risk).

# 5. Model Training and Validation

## 5.1 Training Environment

* **Programming Language:** Python
* **Key Libraries:**
  + pandas: For data manipulation and analysis.
  + numpy: For numerical operations.
  + scikit-learn: For machine learning algorithms (GradientBoostingClassifier, IsolationForest, StandardScaler, GridSearchCV).
  + fuzzywuzzy: For fuzzy string matching.
  + matplotlib, seaborn: For data visualization.
  + joblib: For model persistence (saving/loading models).
  + scipy.stats: For statistical tests (e.g., KS test for drift detection).

## 5.2 Data Splitting Strategy

* **Sanctions Model:** A synthetic dataset is generated and conceptually split into training and test sets. The model is trained on the training set and evaluated on the test set.
* **AML Model:** Trained on unlabeled transaction data (unsupervised). Evaluation involves comparing its anomaly predictions against a small set of "true suspicious" labels (simulated for evaluation purposes).

## 5.3 Performance Metrics

* **Sanctions Screening Model (Classification):**
  + **Accuracy:** Overall correctness of predictions.
  + **Precision:** Proportion of positive predictions that were actually correct (minimizes false positives).
  + **Recall:** Proportion of actual positives that were correctly identified (minimizes false negatives/missed detections).
  + **F1-Score:** Harmonic mean of precision and recall.
  + **ROC AUC:** Measures the model's ability to distinguish between classes across various thresholds.
  + **Confusion Matrix:** Visualizes true positives, true negatives, false positives, and false negatives.
* **AML Transaction Monitoring Model (Anomaly Detection):**
  + **Classification Report:** Provides precision, recall, F1-score (when evaluated against simulated true labels).
  + **ROC AUC:** Measures the model's ability to rank anomalies higher than normal instances.
  + **Anomaly Score Distribution:** Visual inspection of scores.
  + **Confusion Matrix:** Visualizes true positives, true negatives, false positives, and false negatives.

## 5.4 Backtesting Results

* **Scenario:** The models were evaluated on a simulated historical dataset, representing data from a previous period (e.g., one year prior to current data). This assesses the model's performance on unseen, time-shifted data.
* **Key Findings:**
  + (Expected Outcome): Performance metrics (Precision, Recall, F1, ROC AUC) for both Sanctions and AML models might show slight variations compared to initial training validation, indicating how well the model generalizes to historical data.
  + (Interpretation): A significant drop in performance would suggest potential concept drift or overfitting to the original training period, necessitating model retraining or re-calibration.

## 5.5 Stress Testing Results

* **Scenario 1: Data Quality Degradation (Sanctions Screening Model)**
  + **Methodology:** Introduced simulated typos, missing characters, or slight variations in customer names and addresses within the test data.
  + **Expected Impact:** Degradation in name\_match\_score and address\_match\_score features.
  + **Key Findings:**
    - (Expected Outcome): A decrease in the Sanctions model's recall (missing more true sanctioned entities) and potentially an increase in false positives (flagging more near-misses that aren't true matches due to ambiguity).
    - **Robustness Assessment:** This highlights the model's sensitivity to data quality, emphasizing the need for robust data ingestion and cleaning pipelines in production.
* **Scenario 2: Subtle Anomalies (AML Transaction Monitoring Model)**
  + **Methodology:** True suspicious transactions in the test data were made to mimic normal transactions more closely (e.g., reduced amounts for large transactions, high-risk countries mixed with low-risk ones).
  + **Expected Impact:** Reduced distinctiveness of anomalous patterns.
  + **Key Findings:**
    - (Expected Outcome): A decrease in the AML model's ability to detect true anomalies, leading to lower recall for suspicious transactions. Precision might also be affected if the model struggles to differentiate subtle anomalies from normal noise.
    - **Robustness Assessment:** This evaluates the model's resilience against sophisticated money laundering techniques designed to evade detection. It underscores the need for continuous model updates as typologies evolve.

# 6. Model Explainability and Interpretability

## 6.1 Global Feature Importance

For the **Sanctions Screening Model** (Gradient Boosting Classifier), global feature importance is extracted using model.feature\_importances\_. This provides a high-level understanding of which features contribute most to the model's overall decision-making process.

* **Key Insights:** Typically, name\_match\_score and related fuzzy matching features, along with customer\_country\_risk\_score, are expected to be highly important, reflecting their direct relevance to sanctions screening.

## 6.2 Local Explainability (Conceptual)

While not fully implemented in code due to complexity, the concept of local explainability is critical. Techniques like **SHAP (SHapley Additive exPlanations)** or **LIME (Local Interpretable Model-agnostic Explanations)** would be used to explain *individual* model predictions.

* **Purpose:** For any given alert, an analyst needs to understand *why* that specific customer or transaction was flagged.
* **Mechanism:** These tools attribute the model's prediction to specific feature values for that instance, showing which features pushed the prediction towards "suspicious" and by how much, and which features had a mitigating effect.
* **Benefit:** Enhances trust in the model, empowers analysts to conduct more efficient investigations, and provides crucial evidence for regulatory reporting.

## 6.3 Transparency Considerations

The use of interpretable models (like Gradient Boosting, which allows feature importance) and the application of post-hoc explainability techniques (like SHAP/LIME) are vital to ensure the model's decision-making process is transparent to analysts, auditors, and regulators. This is a core requirement for compliance.

# 7. Model Monitoring and Maintenance

## 7.1 Performance Monitoring

Continuous monitoring of model performance is essential to ensure models remain effective in production.

* **Key Metrics Monitored:** Precision, Recall, F1-Score, ROC AUC (for both models), Alert Volume, False Positive Rate, True Positive Rate (if ground truth is available from analyst dispositions).
* **Thresholds:** Predefined thresholds for each KPI are set. If performance drops below these thresholds, it triggers an alert for review.
* **Frequency:** Monitoring is typically performed daily, weekly, or monthly, depending on data velocity and risk appetite.

## 7.2 Data and Concept Drift Detection

* **Data Drift:** Monitoring for changes in the statistical properties of the input features over time (e.g., mean, variance, distribution shifts). The **Kolmogorov-Smirnov (KS) test** is used to compare feature distributions between a reference dataset (e.g., training data) and new incoming data.
* **Concept Drift:** Monitoring for changes in the underlying relationship between input features and the target variable (e.g., new money laundering typologies emerging). This is harder to detect directly without new labeled data but can be inferred from performance drops.

## 7.3 Retraining Strategy and Frequency

* **Scheduled Retraining:** Models are retrained periodically (e.g., quarterly, semi-annually) using the most recent available data.
* **Event-Driven Retraining:** Retraining is triggered if significant data drift is detected, model performance drops below predefined thresholds, or major regulatory changes occur.
* **Data for Retraining:** New, validated historical data, potentially including analyst-dispositioned alerts (true positives/negatives), is used for retraining.

## 7.4 Version Control (Conceptual)

All model artifacts (code, trained models, scalers, feature lists) are managed under a robust version control system (e.g., Git). A Model Registry stores metadata for each version, including training details, performance, and deployment status.

# 8. Regulatory Compliance Considerations

The development and deployment of AML ML models are subject to stringent regulatory oversight. Key considerations include:

## 8.1 Model Governance Framework

* **Policy and Procedures:** Establishing clear internal policies and procedures for the entire model lifecycle, from development to retirement.
* **Roles and Responsibilities:** Defining clear roles and responsibilities for model owners, developers, validators, and users.
* **Risk Assessment:** Ongoing assessment of model risk (e.g., financial, reputational, regulatory risks).

## 8.2 Auditability and Reproducibility

* All model decisions, data transformations, and code changes must be fully auditable and reproducible. This includes logging every step of the data pipeline, model training, and inference.
* The ability to recreate model outputs from a specific point in time is crucial for regulatory examinations.

## 8.3 Bias and Fairness (Conceptual)

* Models must be developed and monitored to ensure they do not introduce or perpetuate unfair bias against specific customer groups (e.g., based on nationality, ethnicity, or socioeconomic status).
* Regular fairness audits are conducted, and mitigation strategies are implemented if bias is detected.

# 9. Future Enhancements / Roadmap

Potential future enhancements for this AML ML model implementation include:

* **Real-time Processing:** Adapting models for low-latency inference to screen transactions in real-time.
* **Advanced Network Analysis:** Implementing Graph Neural Networks (GNNs) to detect complex money laundering networks and relationships beyond individual transactions.
* **Enhanced User Interface (UI):** Developing a dedicated, interactive case management system for analysts with rich visualizations and integrated investigation tools.
* **Automated Feedback Loops:** More sophisticated integration of analyst dispositions directly into model retraining pipelines (e.g., active learning).
* **Explainable AI Integration:** Full integration of SHAP/LIME outputs directly into the analyst's alert review interface.
* **Cloud-Native Deployment:** Leveraging cloud services (e.g., AWS SageMaker, GCP AI Platform, Azure Machine Learning) for scalable and managed MLOps.

# 10. Appendices (Conceptual)

## 10.1 Data Dictionary

(Placeholder for detailed definitions of all input and output features used by the models)

## 10.2 Code Snippets

(References to key functions or modules within the codebase, e.g., calculate\_sanctions\_features(), feature\_engineer\_transactions(), train\_sanctions\_model(), train\_aml\_model(), monitor\_model\_performance(), detect\_data\_drift())