High-Level Design Document

Written By:	Vaishali Parashar
Document version:	1.0
Last Revised Date:	16 March 2022

Contents

1. Introduction

1.1. Why this High-Level Design Documents ?

The purpose of this High-Level Design (HLD) Documents is to add necessary details to the current project description to represent a suitable for coding. This document is also intended to help detect contradictions before coding. And can be used as a reference manual for how the modules interact at a high level.

The HLD will be :

- Present all of the design aspects and define them in detail.
- Describe the user interface being implemented.
- Describe the needed Python libraries for the coding.
- Describe the performance requirements.
- Include design features and the architecture of the project.
- List and describe the non-functional attributes like:
    - Security
    - Reliability.
    - Maintainability.
    - Portability.
    - Reusability.
    - Application compatibility.
    - Resource utilisation.
    - Serviceability.

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture(layers), application flow (Navigation), and technology architecture, The HLD uses non-technical and mildly-technical terms which should be understandable to the administrators of the system

1.3 Definition

TERM	DESCRIPTION
MLD	Money Laundering Detection
AML	Anti-Money Laundering
IDE	Integrated Development Environment
Database	Collection of all the information. Monitored by this system.

2. General Description

2.1 Product Perspective

The Money Laundering Detection system is a machine learning based classification model which will help us to detect the money laundering transactions and take the necessary action.

2.2 Problem Statement

To create an AI solution for money laundering which reduces review operation cost by lowering the number of false positive alerts generated by current static rule based AML systems.

2.3 Proposed Solution

The solution proposed her is to create a machine learning model which is able to take the data from Static rule based AML system and further classify them as Fraud or Not Fraud to reduce the False Positives.

This solution can be able to find:-

- Currently invisible transaction behaviour.
- Aberrations in transactions.
- Reduce overall review operation cost and false positives of AML systems.

2.4 Further Improvements

With the help of more data and training MLD systems can be added directly in place of static rule based AML system.

2.5 Technical Requirements

There are not many technical requirements needed for our MLD solution.

2.6 Data Requirements

Data requirement completely depends on our problem statement

- We need transaction history which should be in tabular form or must have at least 10,000 records.
- The transaction data must have features like
    - Type of transaction
    - Timestamp of the transaction
    - Transaction amount
    - Name of the entity who transferred the money
    - Balance before transaction of both transferred and received entity
    - Name of the entity who received the money
    - Number of transactions both transferred and received person made over a particular period of time etc.

– The Transaction data should be imbalanced and should replicate real world data.

2.7 Tools Used

Python Programming Language and frameworks such as Numpy, Pandas, Scikit-Learn and Catboost are used to build the whole model.

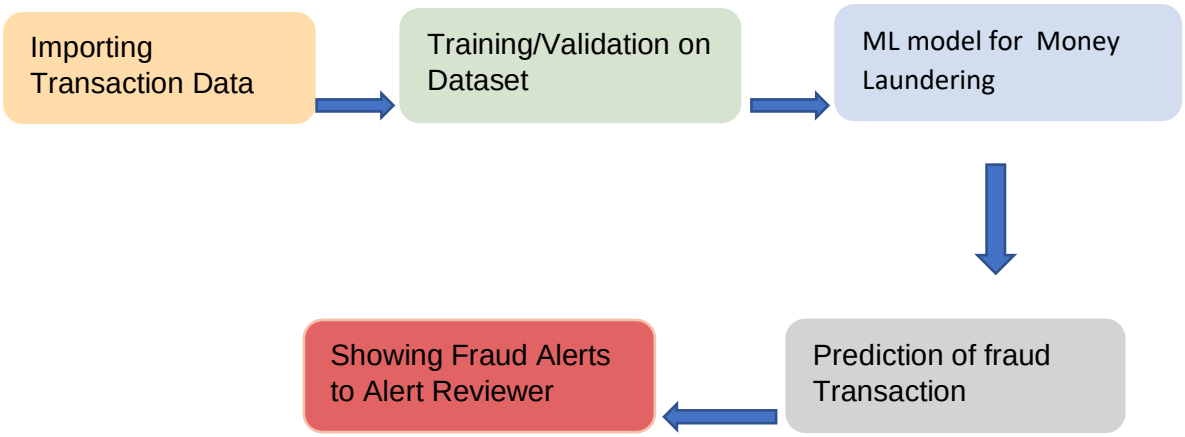
- Visual Studio code is used as an IDE
- For visualisation of the plots, Matplotlib and Seaborn are used.
- Heroku Cloud platform is used for deployment of the model.
- Front End development is done using HTML/CSS.
- Flask framework is used for backend development.
- GitHub is used as a version control system.

3. Design Details

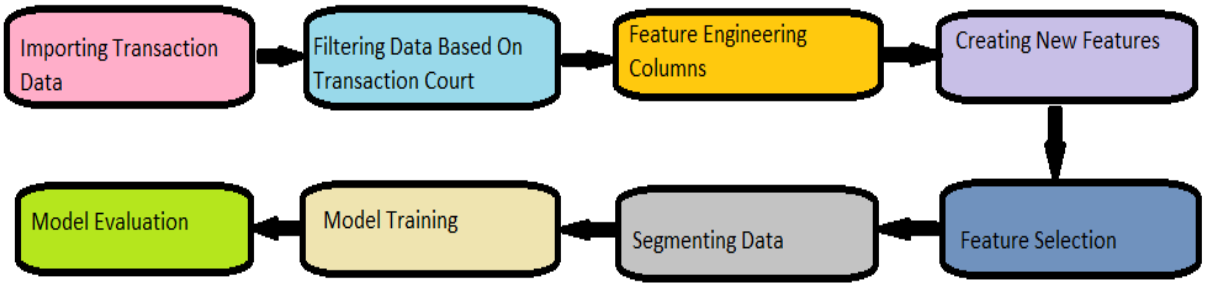
3.1 Process Flow

For identifying the different types of Money Laundering Fraud Transactions , we will use a machine learning based model. Below is the process flow diagram shown.

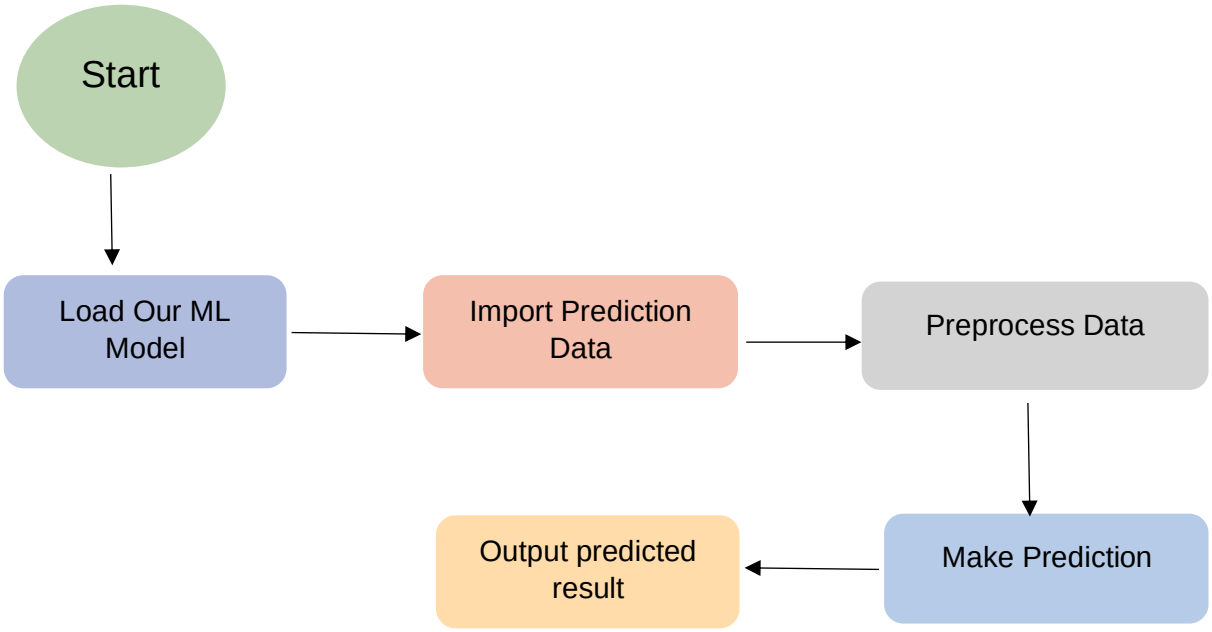
Proposed Methodology



3.2 Model Training Flow



3.3 Deployment Flow



4. Performance

Our machine learning model performed well compared to static AML systems. It reduced the false positive significantly. We used f1 score as our performance metrics for our model Deployment process. Our ML model achieved an f1 score of 0.74.

4.1 Reusability

The code written and the components used to have the ability to be reused with no problems.

4.2 Application Compatibility

The different components of this project will be using Python as an interface between them. Each component will have its own tasks to perform, and it is the job of the path and to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed , it will likely use all the necessary processing power until that task is finished.

4.4 Deployment

This MLD model is deployed in Heroku Cloud Platform and the predicted results are stored in the folder in the form of .csv .

5 . Conclusion

The designed MLD solution uses machine learning to detect money laundering transactions and it is reducing the reiew operation cost by lowering the number of false positive alerts generated by current static rule based AML systems**.**