Low level design Document

Written By	**Arpitha. G**
Document Version	1.0
Last Revised Date	25 – September - 2023

Contents

1. Introduction

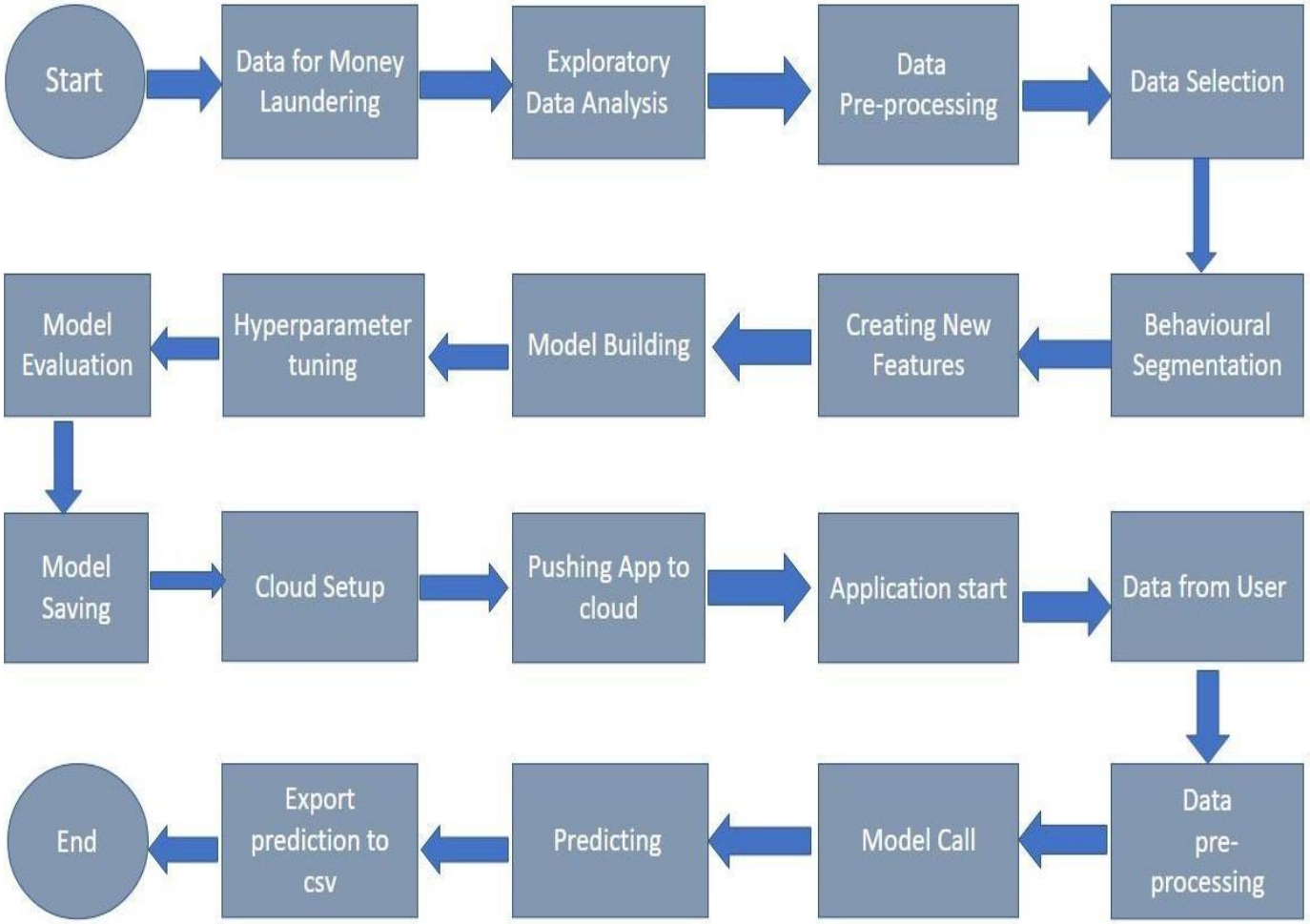
1.1. What is Low-Level Design document ?

The goal of the LLD or Low Level design document (LLDD) is to give the internal logic design of the actual programme code for the Money Laundering Prevention. LLD Describes the class diagrams with the methods and relations between classes and programmes specifications . It describes the modules so that the programmer can directly code the program from the document .

1.2. Scope

Low level design (LLD) is a component level design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organisation may be defined during requirement analysis and then refined during data design work.

2. Architecture



3. Architecture Description

3.1. Data Description

The synthetic data set generated using the simulator called PaySim. PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. This data set contains 6362620 transaction records with 11 features.

3.2. Exploratory Data Analysis :

Exploratory data analysis or EDA is an important step in any data analysis or data science project . EDA is the process of investigating the dataset to discover patterns and anomalies (outlier) and form hypothesis based on our understanding of the dataset.

3.3. Data Pre-processing

Data Pre-processing is the process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data.

3.4. Feature Engineering :

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. If feature engineering is done correctly, it increases the predictive power of the machine learning algorithms by creating features from raw data that facilitate the machine learning process.

3.5. Feature Selection :

A feature selection algorithm can be seen as the combination of a search technique for proposing new features subset, along with an evaluation measure which scores the different feature subsets. The simplest algorithm is to test each possible subset of features finding the one which minimises the error rate . This is an exhaustive search of the space and is computationally intractable for all. But the smallest of feature sets.

3.6. Model Building :

A machine learning model is built by learning and generalising from training data, then applying that acquired knowledge to new data it has never seen before to make predictions and fulfil its purpose. Lack of data will prevent you from building the model, and access to data isn't enough.

3.7. Decision Tree Classifier :

Decision tree is the most powerful and popular tool for classification and prediction . A decision tree is a flow chart like tree structure, where each internal load denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal load) holds a class label.

3.8. XGBoost Classifier :

XGBoost Model for Classification XGBoost is short for Extreme Gradient Boosting and is an efficient implementation of the scholastic gradient boosting machine learning algorithm. The stochastic gradient boosting algorithm, also called gradient boosting machines or tree boosting , is a powerful machine learning technique that performs well or even best on a wide range of challenging machine learning problems.

3.9. Hyperparameter Tuning:

While model parameters are learning during training – such as the slope and intercept in a linear regression – hyperparameters must be set by the data Scientist before training . In the case of a random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node.

3.10. Model Validation :

For machine learning systems, we should be running model evaluation and model tests in parallel. Model evaluation covers metrics and plots which summarize performance on a validation or test dataset . Model testing involves explicit cheeks for behaviours that we expect our model to follow.

3.10.01. Flask:

Flask is a micro web framework written in python. Here we use flask to create an API that allows to send data, and receive a prediction as a response. Flask supports extensions that can add application features as if they were implemented in flask itself .

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether the User is able to sign up in the application	1. Application is accessible	The User should be able to sign up in the application
Verify whether user is able to successfully login to the application	1. Application is accessible 2. User is signed up to the application	User should be able to successfully login to the application
Verify whether user is able to see input fields on logging in	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to see input fields on logging in

Verify whether user is able to edit all input fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance with the selections user made	1. The application is accessible 2. User is signed up to the application 3. User is logged in to the application	The recommended results should be in accordance with the selections user made
Verify whether the user has options to filter the recommended results as well	1. The application is accessible 2. User is signed up to the application 3. The user is logged in to the application	Users should have options to filter the recommended results as well
Verify whether KPIs modify as per the user inputs for the user's health	1. The application is accessible 2. User is signed up to the application 3. User is logged in to the application	KPIs should be modified as per the user inputs for the user's health

Verify whether the KPIs indicate details of the suggested recipe	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The KPIs should indicate details of the suggested recipe