Low Level Design

Credit Card Default Prediction

|	
Written By	Alansha M
Last Revised Date	24 – 03 -2024

Contents

1. Introduction

1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Credit card default prediction. LLD describes the class diagrams with the methods and relations between classes and program specs. 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. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture

Import Library

Save Model

Prediction
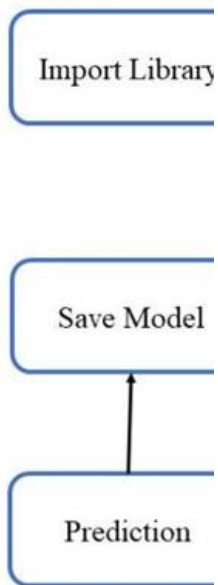
3. Architecture Description

3.1. Data Description

The dataset was taken from Kaggle (URL: https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset), This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

3.2. Data Insertion into Database

i   Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
ii   Table creation in the database.
iii Insertion of files in the table

3.3. Data Pre-processing

This included importing of important libraries such as seaborn, matplotlib, pandas etc. We imported the same dataset mentioned above from Kaggle.

3.4. Model Building

We tried and tested multiple models such as XGBoost, RandomForest,Decision Tree, for the model and came up with the model with the best performance, i.e the Random Forest Classifier.

3.5. Data from User

Here we will collect data from user such as Sex and Bill AMT, Pay AMT, Limit balance, marital status, Age; as well as information directly provided by the user such as Bill AMT and Pay AMT of each month

3.6. Data Validation

Here Data Validation will be done, given by the user

3.7. User Data Inserting into Database

Collecting the data from the user and storing it into the database. Here the database used is Astra DB Cassandra database.

3.8. Data Clustering

The model created during training will be loaded, and clusters for the user data will be predicted.

3.9. Model Call for Specific Cluster

Based on the cluster number, the respective model will be loaded and will be used to predict/Recommend the data for that cluster.

3.10. Recipe Recommendation & Saving Output in Database

After calling model Recipe/Output will be recommended, this output will be saved in Database and it will beused to show the same Output if other users provide the same data.

3.11. Deployment

We will be deploying the model to AWS.

Here is the image of the same: