Credit Card Default Prediction

Low Level Design

Document Version 0.2

Last Revised Date 21 /1/2024

Document Version Control

Date Issued	Version	Version	Author
25/12/2023	1	Initial LLD— V1.0	Aditya Patil
21/1/2024	2	Final LLD __V2.0	Aditya Patil

Contents

1. Introduction

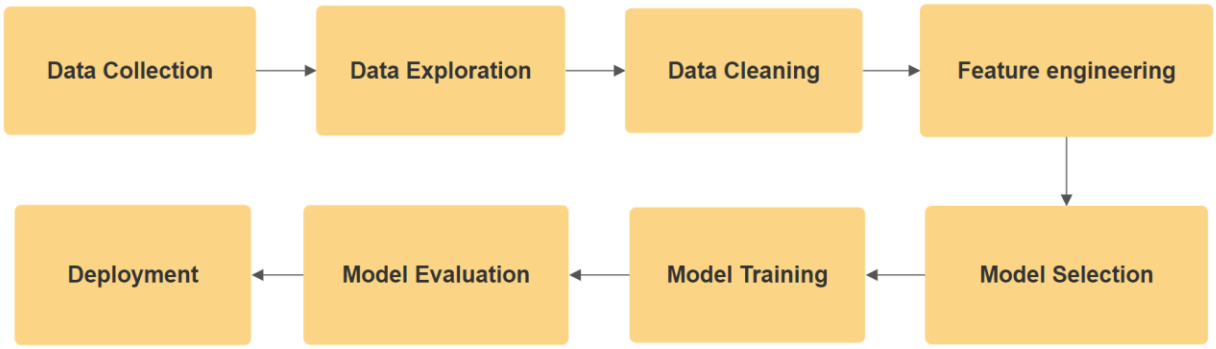
1.1. What is a Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. 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



3. Architecture Description

3.1 Data Collection

Obtained relevant data from commercial banks, ensuring it includes information on credit card owner characteristics and payment history. 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 Exploration

Used descriptive statistics and visualizations to understand the distribution of key features. Identified potential correlations between features and the target variable (credit default).

3.3 Data Cleaning

Imputed missing values using appropriate techniques and instances with incomplete data are removed. Outliers are addressed through methods like trimming or transformation to maintain dataset integrity.

3.4 Feature Engineering

There are no categorical variables for Encoding and Numerical features are standardized to prepare the data for modeling.

3.5 Model Selection

Considering logistic regression, decision trees, random forests, and support vector machines. Suitable machine learning algorithms for classification tasks are selected. Experiments with different algorithms are done to identify the most effective approach.

3.6 Model Training

The dataset is split into a training set and test set for training the data to build the best model, after splitting the data the training data is trained with different classification algorithm to build the model for predicting the result.

3.7 Model Evaluation

Evaluation of different classification model is done through the test dataset, and accuracy score is defined the accuracy score of the logistic regression is more among all the other models therefore the logistic regression is selected for predicting result.

3.8 Deployment

Using the Python Flask the web application API is built, and the user interface is created using HTML/CSS. The form is created for taking the values from user and the result is displayed using render templet through Flask. In such a way deployment is done.

4. Architecture of Model Training