Low Level Design

Credit Card Default Prediction

Written By	Ankita Ramvir Singh
Document Version	1.0
Last Revised Date	14/03/2023

Contents

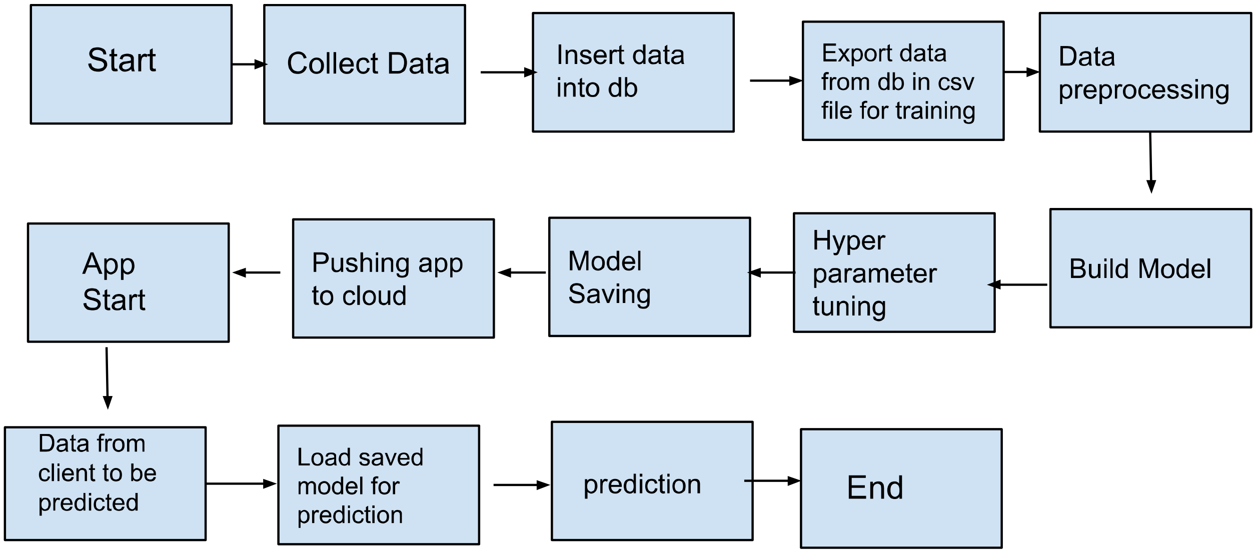
1. Introduction

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 Credit Card Default Prediction System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmers 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 Description/Collect Data

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 Insert Data in Database

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

3.3 Export Data from db in csv file for training

The data in a stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

3.4 Data Preprocessing

Data Pre-processing steps we could use are null value handling,imbalanced data set handling, removing columns, etc.

3.5 Build Model/Hyperparameter tuning

Algorithms will be passed with the best parameters derived from Grid-Search, we will calculate the accuracy and select the best algorithm for model training.

3.6 Model Saving

Model is saved in a pickle file.

3.7 Push app to cloud/App Start

Once the model is trained, create a UI and then pushed the model in AWS cloud.
Then start the app.

3.8 Data from the client to be predicted

Client will input the data from the created UI.

3.9 Load saved model for prediction

Load the trained model pickle file for prediction.

3.10 Prediction

The model will predict the output.