CREDIT CARD DEFAULT PREDICTION

: Low Level Design (LLD) :

CREATED BY

Rakesh

Document Version Control

Date	Version	Description	Author
26-July-2022	1.0	Initial LLD	Rakesh

Table of Contents

1. Introduction:

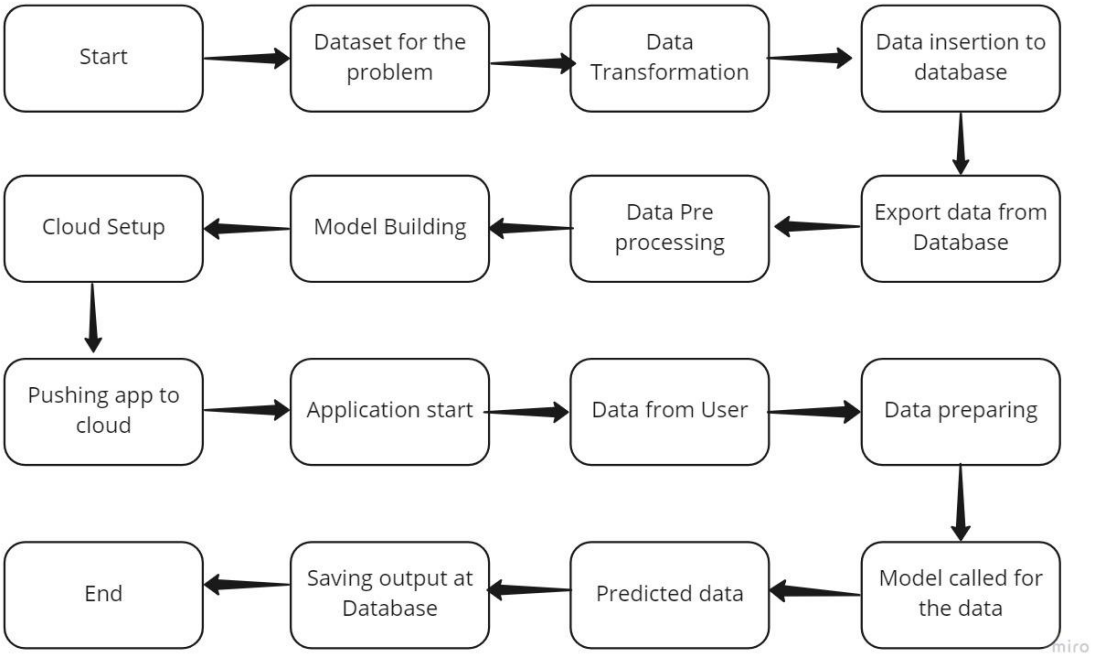
1.1 Why this Low-Level Design Document?

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 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 for the problem

UCI_Credit_Card dataset from UCI repository which has 30000 datapoints. This dataset contains last six months credit card usage history of different customers with customer details.

3.2 Data Transformation

In the Transformation Process, we will convert our original dataset which is in excel format to CSV Format.

3.3 Data Insertion into Database

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

3.4 Export Data from Database

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

3.5 Data Pre-processing

The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Feature Selection, Feature scaling, Data Balancing using sampling techniques etc.

3.6 Model Building

After all the data pre-processing we will find the best model data. Each try, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the Accuracy scores for models and select the model with the best score. Then the best model will be saved for the prediction purpose.

3.7 Data from User

Here we will collect data of user such as age, limit-balance, educational status , marital status, sex, payment statuses, bill amounts, paid amount against each bills of last six months.

3.8 Data preparing

Here given data will be undergone all the pre-processing techniques (3.5) which we done on the early available dataset.

3.9 Model calling for the data

The saved model will be called for the prediction on the given data.

3.10 Predicted data

On the given data the loaded model will perform prediction.

3.11 Saving output at Database

The given data with the newly predicted data will be inserted to the pre-defined table available in the database for future usages.

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 user is able to see input fields on logging in	1. Application is accessible 2. User is able to see input fields.	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 able to see input fields. 3. User is able to edit input fields.	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 able to see input fields. 3. User is able to edit input fields. 4.User is able to see submit button.	User should get Submit button to submit the inputs
Verify whether user is presented with prediction results on clicking submit	1. Application is accessible 2. User is able to see input fields. 3. User is able to edit input fields. 4.User is able to see submit button.	User should be presented with Predicted with results on clicking submit

THE END