Low Level Design

Credit Card Default Prediction

|  |  |
| --- | --- |
| Written By | Md. Asif Mredha |
| Document Version | 0.1 |
| Last Revised Date | 16-Jan-2023 |

**Document Control**

**Reviews**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
|  |  |  |  |

**Approval Status**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review Date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |

**Contents**

1. **Introduction**
2. **Architecture**
3. **Architecture Description**
4. **Unit Test Cases**

**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 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**

End

Create Web Framework

Model Building

Data Preprocessing

Export data from database

Data for Recommendation

Start

3.1. Data Description

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.

3.3. 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.

3.3. Data Pre-processing

Data Pre-processing steps we could use are Null value handling, stop words removal, punctuation removal, Tokenization, Lemmatization, TFIDF, Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

3.4. Model Building

After clusters are created, we will find the best model for each cluster. For each cluster, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the AUC scores for models and select the model with the best score. Similarly, the models will be selected for each cluster. All the models for every cluster will be saved for use in Recommendation.

3.5. Deployment

We will be deploying the model to Flask. This is a workflow diagram for the Credit Card Default Prediction.

**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 |