Low Level Design

Credit Default Prediction System

|  |  |
| --- | --- |
| Written By | Vignesh R |
| Document Version | 0.1 |
| Last Revised Date | 14 – Aug -2022 |

**Document Control**

### Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 0.1 | 14 – Aug -  2022 | Vignesh | Initial Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Reviews:

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

### Approval Status:

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

Contents

[1. Introduction 1](#_Toc112436993)

[1.1. What is Low-Level design document? 1](#_Toc112436994)

[1.2. Scope 1](#_Toc112436995)

[2. Architecture 2](#_Toc112436996)

[3. Architecture Description 4](#_Toc112436997)

[3.1. Data Description 4](#_Toc112436998)

[3.2. Web Download 4](#_Toc112436999)

[3.3. Data Ingestion 4](#_Toc112437000)

[The downloaded data will be stratified split into train and test sets for further processing 4](#_Toc112437001)

[3.4. Data Validation 4](#_Toc112437002)

[The data validation step will check whether the train test files exist. if it exists then it will schema of the files as per the requirement and only it found the valid schema it will perform the data drift check. The details about the data drift along with other validations will be stored as an ratification for further reference. 4](#_Toc112437003)

[3.5. Data Transformation 4](#_Toc112437004)

[3.6. Model Building 4](#_Toc112437005)

[3.7. Model Evaluation 4](#_Toc112437006)

[3.8. Model Push 5](#_Toc112437007)

[3.9. Deployment 5](#_Toc112437008)

[4. Unit Test Cases 5](#_Toc112437009)

# Introduction

## What is Low-Level design document?

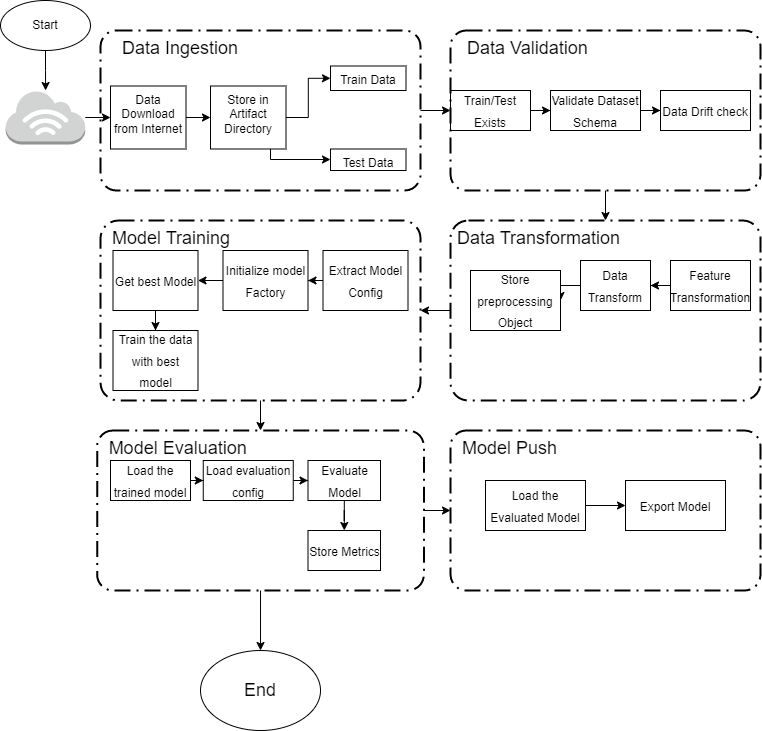
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-

step [refinement](https://en.wikipedia.org/wiki/Refinement_(computing)) 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

# Architecture



# Architecture Description

## 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. It has 25 features

## Web Download

This data will be fetched and downloaded from the web

## Data Ingestion

## The downloaded data will be stratified split into train and test sets for further processing

## Data Validation

## The data validation step will check whether the train test files exist. if it exists then it will schema of the files as per the requirement and only it found the valid schema it will perform the data drift check. The details about the data drift along with other validations will be stored as an ratification for further reference.

## Data Transformation

In the Transformation Process, we will convert our original dataset by filling the missing values and apply feature selection, feature transformations and store the preprocessing object.

## Model Building

After the transformation process, the model related details like, model name to be used, model parameters will be picked from the model config and then initialize the model factory.

The model factory is responsible for training the model with the default parameters given and also to perform the grid search cv operations based on the parameters given and then get the best score and best param out from the grid search cv operations. The best model will be selected and stored.

## Model Evaluation

Here we will load the best trained model which we stored in our repository in the previous step. Then the evaluation metric related details are stored in the separate configuration file. We need to fetch the model evaluation metrics from the config then pass the test csv file to perform predict operation. Then the predicted results will be compared with the original result based on the evaluation metrics in the config and store the final results in the artifact.

## Model Push

Load the evaluated model from the previous step and then store the model in the corresponding repository.

## Deployment

We will be deploying the model to Heroku as a containerized approach.

# 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 the menus in the application | 1. Application is accessible | User should be able to successfully see the menus in the application |
| Verify whether user is able to see input fields on logging in | 1. Application is accessible | User should be able to see input fields |
| Verify whether user can see the logs under the “View logs” menu | 1. Application is accessible 2. Model should be trained once in web | User should be able to see logs under the “View logs” menu |
| Verify whether user can see the logs under the “View Artifacts” menu | 1. Application is accessible 2. Model should be trained once in web | User should be able to see logs under the “View Artifacts” menu |
| Verify whether user can see the logs under the “View Trained Model” menu | 1. Application is accessible  2. Model should be trained once in web | User should be able to see logs under the “View Trained Model” menu |
| Verify whether user is able to edit all input fields | 1. Application is accessible  2. Model should be trained once in web | User should be able to edit all input fields |
| Verify whether user gets Submit button to submit the inputs | 1. Application is accessible  2. Model should be trained once in web | User should get Submit button to submit the inputs |
| Verify whether user is presented with results on clicking  submit | 1. Application is accessible  2. Model should be trained once in web | User should be presented with results on clicking  submit |
| Verify whether the results are in accordance to the selections user made | 1. Application is accessible  2. Model should be trained once in web | The results should be in accordance to the selections user made |