Low Level Design

Credit Card Default Prediction

|  |  |
| --- | --- |
| Written By | Mohd Salman |
| Document Version | 0.1 |
| Last Revised Date | 12 -Dec -2022 |

**Document Control**

### Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 0.1 | 12 -Dec -  2022 | Mohd Salman | Introduction & Architecture defined |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Reviews:

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

### Approval Status:

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

Contents

[1. Introduction 1](#_Toc119335002)

[1.1. What is Low-Level design document? 1](#_Toc119335003)

[1.2. Scope 1](#_Toc119335004)

[2. Architecture 2](#_Toc119335005)

[3. Architecture Description 3](#_Toc119335006)

[3.1. Data Description 3](#_Toc119335007)

[3.2. Data Pre-processing 3](#_Toc119335008)

[3.3. Data Analysis 3](#_Toc119335009)

[3.4. Feature Engineering 3](#_Toc119335010)

[3.5. Train/Test Split 3](#_Toc119335011)

[3.6. Selecting Model 3](#_Toc119335013)

[3.7. Prediction 3](#_Toc119335014)

[3.8. Save Model 3](#_Toc119335015)

[3.9. Data from User 4](#_Toc119335017)

[3.10. Data Validation 4](#_Toc119335018)

[3.11. User Data Inserting into Database 4](#_Toc119335019)

[3.12. Data Clustering 4](#_Toc119335020)

[3.13. Model Call for Specific Cluster 4](#_Toc119335021)

[3.14. Saving Output in Database 4](#_Toc119335022)

[3.15. Deployment 4](#_Toc119335023)

[1. Unit Test Cases 5](#_Toc119335024)

# Introduction

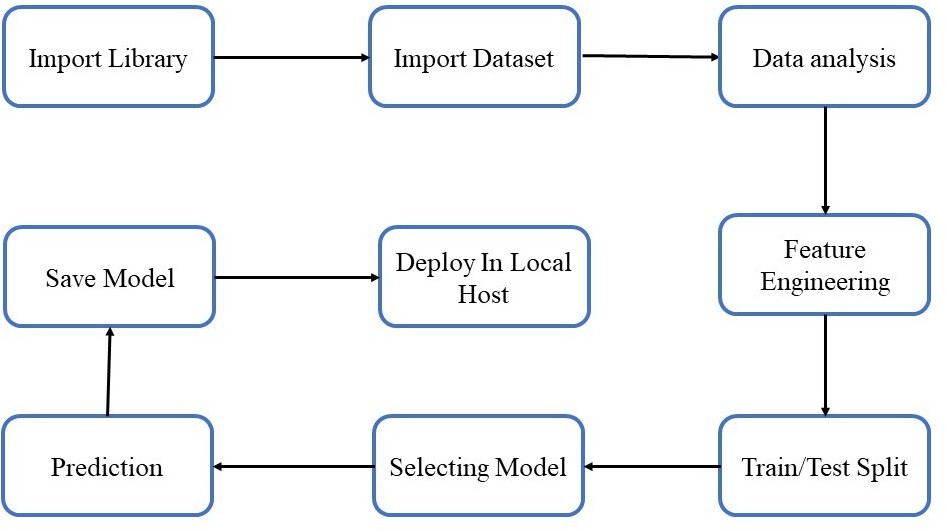
## What is Low-Level design document?

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

The dataset was taken from Kaggle (URL: [https://www.kaggle.com/uciml/default-](https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset) [of-credit-card-clients-dataset](https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset)), 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.

## Data Pre-processing

This included importing of important libraries such as numpy, seaborn, matplotlib, pandas etc. We imported the same dataset mentioned above from Kaggle.

## Data Analysis

Here we handled the null values, changed the column names, plotted multiple graphs in seaborn, matplotlib and other visualization library for proper understanding of the data and the distribution of information in the same. As there were no null values in the data, we proceeded with the visualization and analysis. For each specific feature we analyzed the data using visualization, and jotted down the important key points which can impact the final predictions.

## Feature Engineering

Merging 2 or mode columns to get in depth knowledge and information regarding the data.

## Train/Test Split

## This library was imported from sklearn to divide the final dataset into the ratio of 80-20%, where 80% of the data was used to train the model and the latter 20% was used to predict the same.

## Selecting Model

We tried and tested multiple models such as XG Boost, Random Forest, Decision Tree, ADA Boost for the model and came up with the model with the best performance, i.e. the Random Forest Classifier.

## Prediction

The Accuracy of Random Forest was 82.12% and the F1 score was 47.8%.

## Save Model

## Model was saved using the pickle library which saves the file in a binary mode.

## Data from User

Here we will collect from user such as gender, education, limit balance, repayment status etc.

## Data Validation

Here Data Validation will be done, given by the user

## User Data Inserting into Database

Collecting the data from the user and storing it into the database. The database can be either MySQL, Mongo DB or Cassandra database.

## Data Clustering

The model created during training will be loaded, and clusters for the user data will be predicted.

## Model Call for Specific Cluster

Based on the cluster number, the respective model will be loaded and will be used to predict/Recommend the data for that cluster.

## Saving Output in Database

After calling model Output will be predicted, this output will be saved in Database and it will be used to show the same Output if other users provide the same data.

## Deployment

We will be deploying the model to AWS.

# 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 the recommended results are in accordance to the selections user made | 1. Application URL is accessible  2. Application is deployed | The recommended results should be in accordance to the selections user made |

This is the workflow diagram :

