Low Level Design

Stores Sales Prediction

|  |  |
| --- | --- |
| Written By | Namitha Kolur |
| Document Version | 0.1 |
| Last Revised Date | 21 – July -2023 |

**Document Control**

### Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 0.1 | 21 –July - 2023 | Namitha Kolur | Introduction & Architecture |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Reviews:

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

### Approval Status:

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

Contents

1. [Introduction 1](#_bookmark0)
   1. [What is Low-Level design document? 1](#_bookmark1)
   2. [Scope 1](#_bookmark2)
2. [Architecture 2](#_bookmark3)
3. [Architecture Description 3](#_bookmark4)
   1. [Data Description 3](#_bookmark5)
   2. [Data Insertion into Database 3](#_bookmark8)
   3. [Data Pre-processing 3](#_bookmark10)
   4. [Feature Selection 3](#_bookmark11)

3.5  [Maching Learning Techniques 4](#_bookmark12)

3.6  [Model Building 4](#_bookmark13)

3.7 [Evaluation 4](#_bookmark14)

3.8 [Cloud Setup 4](#_bookmark15)

3.9 [Pushing to Cloud 4](#_bookmark16)

* 1. [Application Start 4](#_bookmark17)
  2. [Data from User 4](#_bookmark18)
  3. [Data Validation 4](#_bookmark19)

3.13 Result Prediction………………………………………………………………………………………………………………4

4. Unit Test Cases……………………………………………………………………………………………………………………...5

# **Introduction**

## **What is Low-Level design document?**

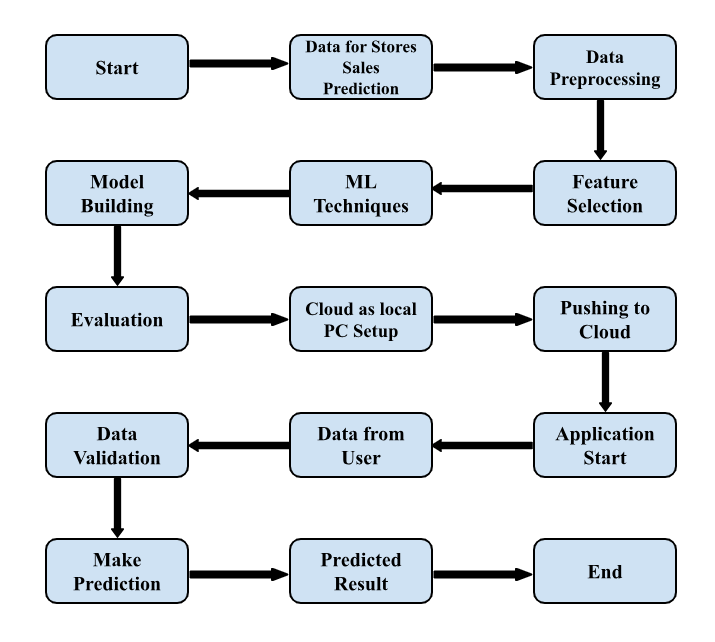
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can 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 used in this project is collected from Kaggle. The dataset is divided into two sets of data. One is the test (5681) data and the other is the train (8523) data. The train dataset has both input and output variables.

## **Data Insertion into Database**

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

## **Data Pre-processing**

## Data Pre-processing steps we could use are data cleaning, data integration, data reduction and data transformation.

## **Feature Selection**

Feature Selection helps us to find the best set of features that allows us to build the necessary model for the project. This helps in selecting a subset of features from an initially large volume of features.

## **Machine Learning Techniques**

Based on the problem statement and requirements we can use supervised or unsupervised technique which fits the project.

## **Model Building**

Depending on the data type of the target variable we are either going to be building a classification or regression model. The main aspect of machine learning model building is to obtain actionable insights and in order to achieve that it is important to be able to select a subset of important features from the vast number.

## **Evaluation**

The Evaluation of accuracy can be done using the test data. Mean Absolute error can be found using test data and prediction data.

## **Cloud as local PC Setup**

Using local PC as the cloud deployment platform, the platform is setup for deploying the virtual app.

## **Pushing to Cloud**

Once the local PC is setup, the virtual app created will be pushed to the cloud or run on the local PC and will finally be deployed into the cloud or local PC.

## **Application Start**

Once the virtual app is run on the local PC we can open the web application using any web browser.

## **Data from user**

Using a web browser we open the web application and provide the necessary information as the input for prediction.

## **Data Validation**

Once the input is provided and we click on the submit button, the system will provide the output based on its requirements.

## **Result Prediction**

Once the data validation is completed the prediction will be done for the type of product in Stores and Big Marts provided in the input.

# **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 edit all input fields | 1. Application is accessible 2. User can open the application. | 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 can edit the inputs. | User should get Submit button to submit the inputs |
| Verify whether user is presented with suitable results on clicking  submit | 1. Application is accessible | User should be presented with suitable results on clicking  submit |
| Verify whether the suitable results are in accordance to the selections user made | 1. Application is accessible | The suitable results should be in accordance to the selections user made |
| Verify whether KPIs modify as per the user inputs for the prediction | 1. Application is accessible | KPIs should modify as per the user inputs for the prediction |