LOW-LEVEL DESIGN

Store Sales Prediction

Document Version Control

Date Issued	Version	Description	Author
30/06/2023	V1.0	Initial HLD- V1.0	Susmitha P
30/06/2023	V1.0	Initial HLD- V1.0	Bharanidharan D

Contents

1.0 Introduction

1.1 What is Low-Level Design Document?

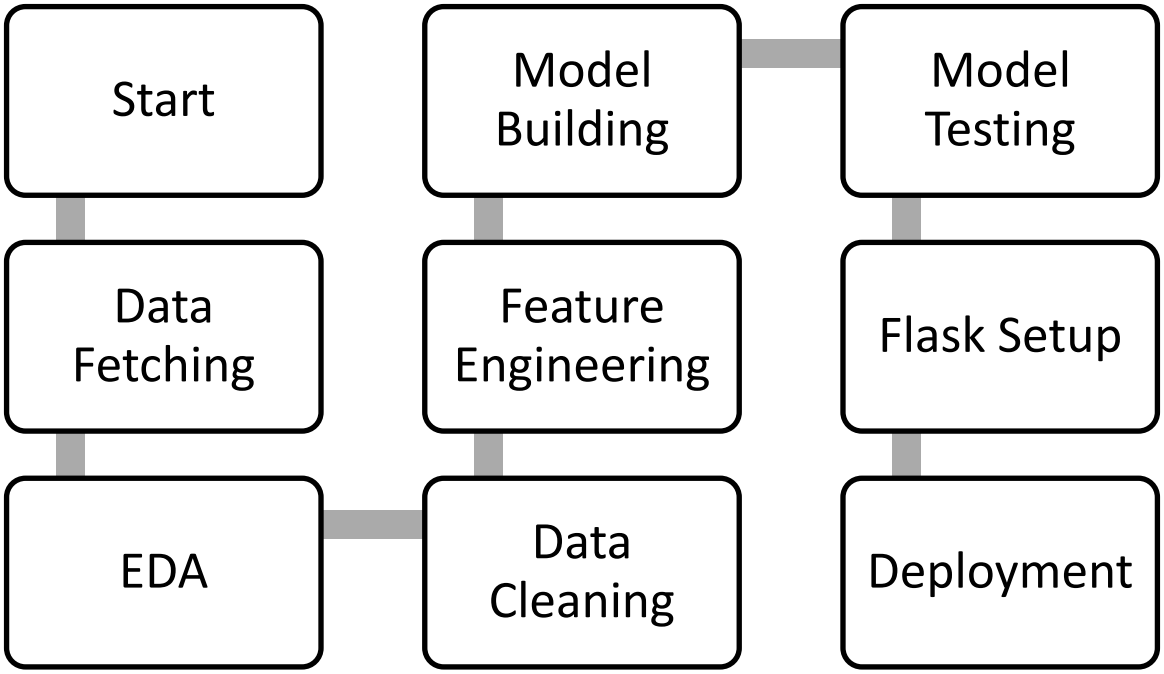
The goal of LLD or Low-Level design document (LDD) is to give the internal logical design of the actual program code. Low-Level design is created based on the High-Level design. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly can directly code the program from the document.

1.2 Scope

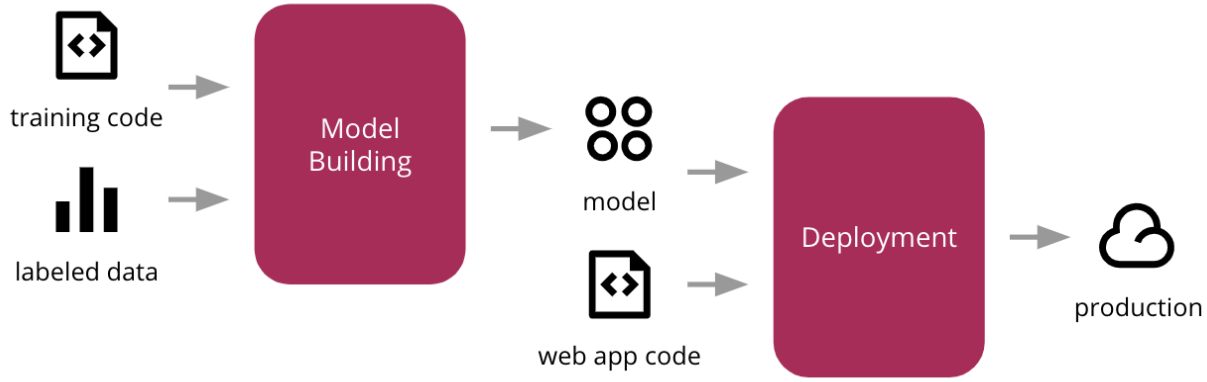
Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2.0 Architecture

2.1 Process Flow



2.2Model Training and Evaluation



3.0 Architecture Description

3.1 Data Collection

We used Big Mart outlet sales data as a dataset for this project where the dataset consists of 12 attributes. There is another dataset for validation purpose. The information each dataset available in Comma-Separated-Values (CSV). Here we use dataset contains 8523 observations and test dataset contains 5681 observations.

3.2 Exploratory Data Analysis

Exploring the data by visualizing the distribution of values in some columns of the dataset, and the relationships between 'Item Outlet Sales' and other columns.

3.3 Data Pre-processing

Data preprocessing is the process of transforming raw data into an understandable format. In data pre-processing all the processes required before sending the data for model building are performed. New attributes were added named ''Outlet years", where the given establishment year is subtracted from the current year. Then mapping of "Fat content" is done based on 'Low', 'Reg' and 'Non-edible'.

3.5 Model Building

After data pre-processing is done, we will split the dataset into training set and validation set. Then processed data is used to give accurate results by applying multiple algorithms. An effective model can predict accurate results by finding exact insights of data. We will calculate RMSE score for each model and select the model with the best score.

3.6 Data Validation

Here Data Validation will be done on the test set.

3.7 Deployment

We will use Heroku platform for the deployment of this project

4.0 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 URL is deployed	Application URL should load completely for the user when URL is accessed
Verify whether user can see input field after opening URL	1. Application is accessible	User should be able to see input fields after opening URL
Verify whether user can edit all the input fields	1. Application is accessible	User should be able to edit all the input fields
Verify whether user has options to filter the inputs fields	1. Application is accessible	User should filter the options of input fields
Verify whether user gets submit button to submit the inputs	1. Application is accessible	User should get submit button to submit the inputs
Verify whether user can see the output after submitting the inputs	1. Application is accessible	User should get outputs after submitting the inputs