Low Level Design (LLD)

Store Sales Prediction

Revision Number: 1.0
Last Date of Revision: 04/06/2022

Contents

Document Version Control

Date Issued	Version	Description	Author
04/06/2022	1	Initial LLD-V1.0	Amit Ranjan Sahoo

Abstract:

Nowadays, shopping malls and Big Marts keep track of individual item sales data in order to forecast future client demand and adjust inventory management. In a data warehouse, these data stores hold a significant amount of consumer information and particular item details. By mining the data store from the data warehouse, more anomalies and common patterns can be discovered.

1 Introduction:

1.1 Why this Low Level Design Document:

The purpose of this Low-Level Design (LLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The LLD will:

- Present all of the design aspects and define them in a detail
- Describe the user interface being implemented
- Includes design features and architecture of the project
- List and describe the nonfunctional attribute like:
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Serviceability

1.2 Scope:

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrator of the system.

1.3 Constraints:

The System must be user-friendly, as automated as possible, and the user should not be required to know any other work.

2. Technical specifications:

2.1 Logging:

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description

1. The system identifies at what step logging required.
2. The system should be able to log each and every system flow.

3. Developers can choose a logging method. You can choose database logging or file logging as well.
4. System should not hang even after so many loggings. Logging just because we can debug issues, so logging is mandatory.

2.2 Database:

System needs to store each and every record in the database, dataset as well as logging. And it has to send data to the user on request. Here we have used a MongoDB database.

3. Deployment:

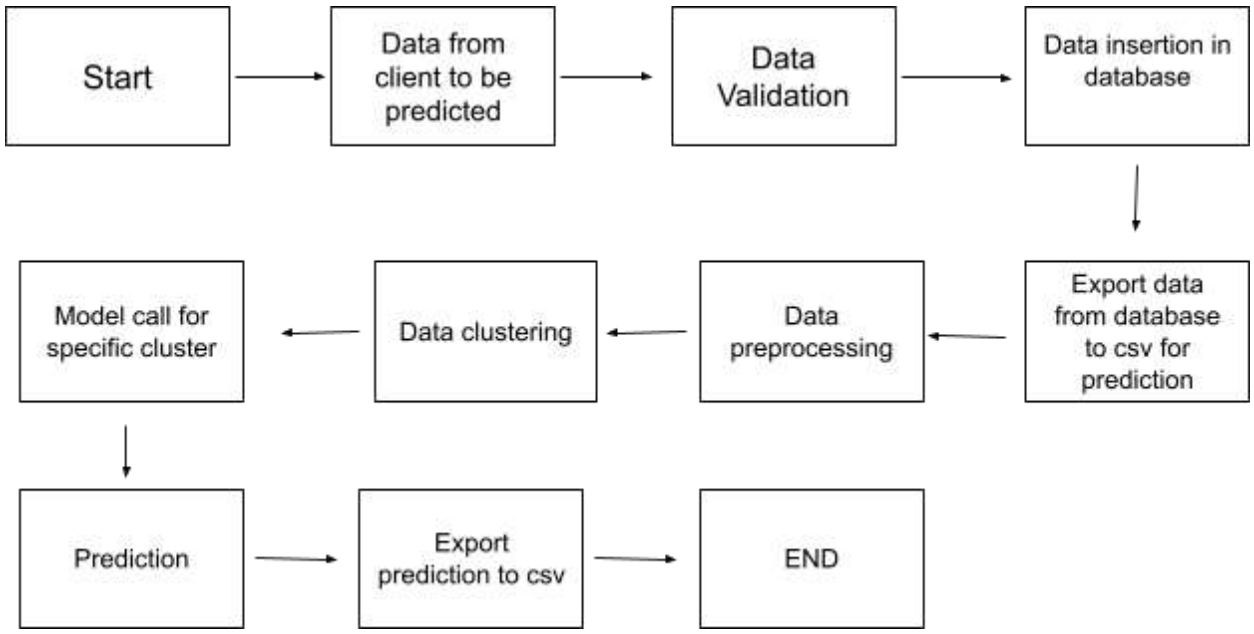
4. Technology stack:

Front End	HTML/CSS/JS/Bootstrap
Backend	Python/ Flask
Database	MongoDB
Deployment	Heroku
version control	GitHub

5. Proposed Solution:

Here a system has been created which predicts the sales of an item in a particular outlet.

6. Architecture:



7. Error Handling:

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

8. KPIS (Key Performance Indicators):

1. Organization can predict the sales easily
2. Rectify the sales of the each product which is maximum sales.

9. Conclusion:

Nowadays, shopping malls and Big Marts keep track of individual item sales data inorder to forecast future client demand and adjust inventory management. In a data Warehouse, these data stores hold a significant amount of consumer information and particular item details. By mining the data store from the data warehouse, more anomalies and common patterns can be discovered.