High-Level Design (HLD)

Stores Sales Prediction

Written By	Shaik Imran Fazil
	Bandi Lokesh
	O Dinesh Kumar
	Shaik Chetansha
Version	1.0
Date	07-02-2023

Version	Date	Author	Comments

Document Change Control Record

Review

Version	Date	Reviewer	Comments

Approval

Version	Review date	Reviewed By	Approved By	Comments

Abstract

The goal of the project is to create a system that can forecast future consumer demand for specific products in retail settings. by looking into a product's previous sales figures. Large shopping centres keep track of the products they offer in order to predict future demand. Manufacturing and product warehouses are useful for keeping many different products. The major objective of this study is to examine historical data, identify significant relationships among various features, and develop a system that can make predictions about how much a certain product will be in demand. The management of warehouse storage capacity will be aided by this method.

1. Introduction

1.1 Why these High-Level Design Documents?

The goal of these High-Level Design(HLD) Documents is to supplement the current project description with the information required to depict an appropriate for coding. The purpose of this text is to aid in the early detection of discrepancies. Additionally, it serves as a reference guide for the high-level interactions between the modules.

The HLD will consist of:
- 1. Presenting and thoroughly defining all design elements.
- 2. Describe the implemented user interface.
- 3. Describe the Python libraries that are required for the coding.
- 4. Describe the necessary performance standards.
- 5. Include design elements and the project's architecture.
- List and describe the non-functional attributes like:
  - Security
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application Compatibility
  - Resource Utilization
  - Serviceability

1.2 Scope

The HLD documentation outlines the system's organisational components, including the database architecture, application architecture (layers), application flow (navigation), and technology architecture. The HLD employs mildly technical and non-technical language that system administrators should be able to grasp.

1.3 Definition

TERM	Description
DB	Database, the cloud platform where the data will be stored. Can be considered cloud storage.
ML	Machine Learning
API or APIs	Application Programming Interface can be considered a website link from there can extract information.

2. General Description

2.1 Product Perspective

The Retail Salse Prediction is a machine learning (ML)-based web application that can forecast product demand by looking at historical data. It will provide the figure used to calculate product sales.

2.2 Problem Statement

Tu creates a system that can analyse information about a product and forecast how much demand there will be for it in the future. The application we create must be able to deliver outcomes.

2.3 Proposed Solution

The significant relationships between various parameters will be discovered using performe EDA, and a machine-learning algorithm will be used to forecast future sales demand. The client will input the necessary feature and receive results via the online application. The system will collect features, which will then be sent to the backend for validation and preprocessing before being fed to a machine learning model with hyperparameter tuning to forecast the result.

2.4 Data Requirements

The data needed for the project's construction is already there on the dashboard. Numerous product descriptions and historical sales figures are included in the Store Sales Prediction data. We will utilise the provided dataset to build the ML model. The data set includes 8523 rows and various product information, including product identifiers, product categories, store identifiers, retail locations, etc.

2.5 Tool Used

Python is the programming language utilised here, and we will also use various other python-based libraries, such as Scikit-Learn for machine learning, Pandas for data manipulation, Numpy for numerical computing, and Flask for building custom APIs. For all modular coding and the development of bespoke APIs, Visual Studio Code is utilised as the Python IDE. We'll use GitHub to store all code files that should be accessible to the public.
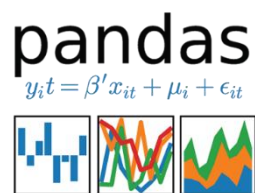
2.6 Constraints

The system should be simple to use, and when utilising the web app, the user should receive all necessary messages. If the user makes a mistake on the web-app page, he or she should receive a suitable error message as well. The user should not be confused when using the system since

all mistakes and results should be presented in the simplest way possible, and all buttons that will be added to the webpage should have accurate labels.
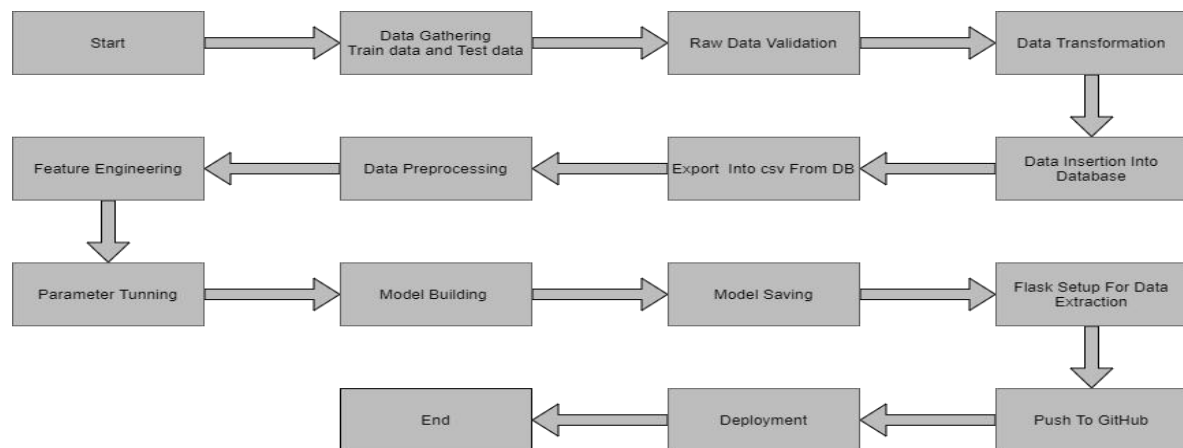
2.7 Assumptions

The major goal is to have a system in place that will generate an estimate of future demand for a product in retailers.
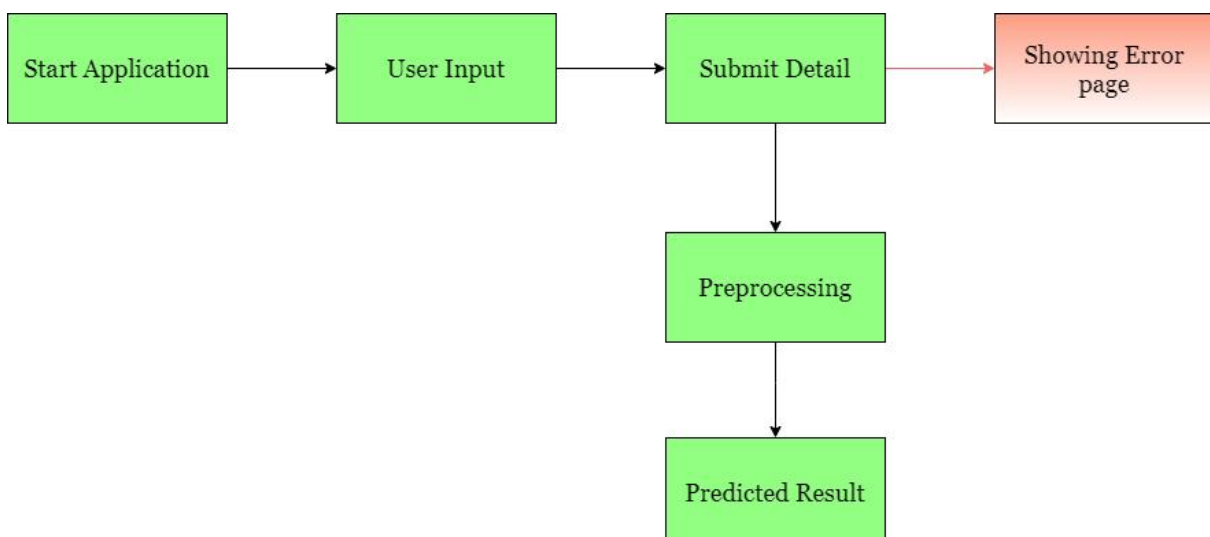
3. Design Details

3.1 Process Flow

For this project, we'll follow the workflow flow below. Modular coding, or using oops notions to build the complete project from beginning to end, will be the foundation of the method.

3.2 Deployment Process

3.3 Error Handling

If a processing issue occurred, the user should be presented with an error message that is entirely understandable to non-technical people. A clear error notice should be displayed so the user can correct his error and retry the process. Errors will happen, and they should all be handled correctly. And we need to manage and log any error that occurs with our application.

4. Performance

Machine-learning methods are necessary for the salse price prediction. We will test a variety of ML algorithms and select the one that best predicts the target. The data we give the algorithms will determine how well our system performs. And how well the model performs will rely on its completion. and the deployment server, as well as the web application. All of these parts should work together to make our programme work effectively.

4.1 Reusability

The project should adhere to all coding standards, and the code and modules developed during the building process should be written in a modular style. Any place should be able to use our technology without any problems. Additionally, it must handle any incorrect input values supplied by the user and issue a clear error notice so that the user can fix their error and supply accurate information in order to receive the desired result. Additionally, the system need to be reusable under all circumstances with various inputs and values that are all consistent with how it was trained.

4.2 Application Compatibility

The system was created using various libraries and the Python programming language. Every library has its own features, and they should all operate well under our unstable system. The web application will be created using HTML/CSS and the web APIs will be created using Flask. The application's components must all function correctly for a result to be generated that cannot be interpreted.

4.3 Resource Utilization

Our programme should make the best use of the provided resource and should only require a little amount of internet to function and use the Web page's APIs. Our system shouldn't utilise a lot of processing power because that will slow down the application. Our programme will be installed on a cloud platform, and it must properly operate and make use of the cloud's resources.

5. Deployment

We will host our application on the Heroku cloud platforms for the deployment process. The system will be run on a cloud platform, which will also provide us the freedom to use our application anywhere in the world.

6. Conclusion

Business owners and manufacturers will soon be able to estimate future product demand thanks to the Sales Store Prediction. It will benefit the product supply chain as well as their ability to expand their business. We keep track of product sales history, prior performance, and store data. In order to discover the internal pattern and forecast the target price or the sales demand of the product in the future, we will analyse the historical data and create an ML model.

7. Reference

For a collection of logos and images, use Google Image.

To draw the diagrams, create a rough sketch.