# High Level Design (HLD)

# Store Sales Prediction

Revision Number: 1.0

Last date of revision: 31-03-2022

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 31-03-2022 | 1 | Initial HLD -V1.0 | Madhav  Pavitra  Neet |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Content

Document Version Control

Abstract

1. Introduction………………………………………………………………………………….5
   1. Why this High-Level Design Document?.......................................................................5
   2. Scope…………………………………………………………………………………...6
2. General Description…………………………………………………………………………7
   1. Product Perspective…………………………………………………………………….7
   2. Problem Statement……………………………………………………………………...7
   3. Proposed Solution………………………………………………………………………7
   4. Further Improvements…………………………………………………………………..7
   5. Technical Requirements………………………………………………………………...8
   6. Data Requirements………………………………………………………………………8
   7. Tools Used………………………………………………………………………………8
   8. Constraints………………………………………………………………………………9
3. Design Details………………………………………………………………………………..10
   1. Process Flow……………………………………………………………………………10
      1. Model Training and Evaluation…………………………………………………10
      2. Deployment Process…………………………………………………………….10
4. Performance………………………………………………………………………………….11
   1. Reusability………………………………………………………………………………11
   2. Application Compatibility………………………………………………………………12
   3. Resource Utilization…………………………………………………………………….12
   4. Deployment……………………………………………………………………………..12

5 Conclusion……………………………………………………………………………………13

# ABSTRACT

# In this era of modernisation and e-learning, everything is based onto internet which generate a huge amount of data, which can be used by humans in various ways to create more advancement in this humongous world of data. This whole process is completed with various machine learning algorithms to solve the purpose and get desired data, but particularly in this case we are trying to predict the sales of store using different machine learning techniques and trying to determine the best algorithm suited to our particular problem statement.

# 1 Introduction

# Why this High-Level Design Document?

The HLD describes the primary features it will deliver, both functional and non-functional. It is best to list all the elements for clarity without going near the implementation details, as these aspects are probably not for this particular audience or in simpler terms it also means that A high-level design document (HLD) describes the architecture used in the development of a particular software product. It usually includes a diagram that depicts the envisioned structure of the software system.

The HLD will:

* + HLD presents all of the design aspects (taken from business requirements and expected outcome) and defines them in form of a diagram.
  + It describes the user interface being implemented and description of hardware and software interfaces.
  + It describes the performance requirements and flow of user’s daily process.
  + HLD includes design features and the architecture of the project.
  + Port numbering, VLAN, physical specifications etc are not part of High-Level Design
  + List and describe the non-functional attributes like:
    - Security
    - Relatability
    - Maintainability
    - Portability
    - Reusability
    - Application Compatibility
    - Resource utilization
    - Serviceability

# Scope

The High-Level Design documentation presents the structure of the system as the application/database architecture, application flow and technology architecture. High-Level Design documentation may use some non-technical terms unlike Low Level design which should be strictly technical jargon. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system

# General Description

# 2.1 Product Perspective

Store Sales Prediction is a solution which is able to predict the sales of the different stores of Big Mart according to the provided dataset and helps understand the product sales in any particular store of any locality.

# 2.2 Problem Statement

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.

# 2.3 Proposed Solution

We start by performing data cleaning followed by EDA to find useful relation between different attributes and will use a machine-learning algorithm such as liner regression to predict the future sales demand. The client will have to fill the required input and will get result through the web application and then the input attributes will get features and it will be passed into the backend where the features will be validated and pre-processed and then it will be passed to a hyperparameter tuned machine learning model to predict the final outcome.

# 2.4 Further Improvements

We can save the data for future use and can take help in future for prediction way more precisely and help the store to generate greater profit by buying the right stuff for particular store and to attract more customers and identify the item's future demand. It will automatically be assigned for production.

# 2.5 Technical Requirements

Any Device with Internet Access and runs in any browser.

# 2.6 Data requirements

The data required for the building of the project is already available on the dashboard. The Store Sales Prediction data recorded many product descriptions along with past sales quantity. For building the ml model we will use the dataset that is given. The data is consisted of 8523 rows and various information about products like product id, product category, store id, store location, etc. We need to predict the sales for the test data set.

* Item\_Identifier: Unique product ID
* Item\_Weight: Weight of product
* Item\_Fat\_Content: Whether the product is low fat or not
* Item\_Visibility: The % of total display area of all products in a store allocated to the particular product
* Item\_Type: The category to which the product belongs
* Item\_MRP: Maximum Retail Price (list price) of the product
* Outlet\_Identifier: Unique store ID
* Outlet\_Establishment\_Year: The year in which store was established
* Outlet\_Size: The size of the store in terms of ground area covered
* Outlet\_Location\_Type: The type of city in which the store is located
* Outlet\_Type: Whether the outlet is just a grocery store or some sort of supermarket
* Item\_Outlet\_Sales: Sales of the product in the particular store. This is the outcome

variable to be predicted.

# 2.7 Tools Used

Programming language and frameworks such as Numpy, Pandas, Scikit-learn, pandas-profiling, etc are used to build the whole model.



* VS code and Google colab can be used as IDE.
* For visualization of the plots Matplotlib, Seaborn is used.
* Flask is used for deployment of the model.
* Front end Development is done using HTML/CSS, Bootstrap 4.
* Python is used for backend development.
* GitHub is used as a version control system.

# 2.8 Constraints

The webpage is user friendly and is automated and users should have basic knowledge of the web page they will be using prepared.

# Design Details

# Process Flow

For Identifying the different types of anomalies, we will use a machine learning model. Below is the process flow diagram as shown below.

Proposed Methodology



# Model Training and Evaluation

# 

# Deployment Process

Start the application

Enter the details

Submit the data

Execute by model

Predicted result

# 4 Performance

The Performance of the model depends on the dataset as well as the data cleaning which is performed on the data set which help in creating dataset for greater accuracy and much closer prediction with less error.

# Reusability

The code and the module are created during the time of building the project should maintain all coding guidelines and full project code is written in a Modular fashion. Our system should have the flexibility to work properly from any location. And it should handle any improper input value from the user and should give a meaningful error message so the user can correct his/her mistake and enter valid input to get the result. And the system should be reusable in every manner with different types of inputs values that are all are it has been trained.

# Application Compatibility

In an ideal environment, a developer would be able to access stable and trusted code libraries. They’d then be able to reuse code from those libraries as building blocks within their application.

So, you can reuse code when it can be:

* Easily extended and adapted for the new application.
* Ported to different hardware if needed.
* Shown to be free from defects or problems that affect the reliability, safety, or security of the new application.

# Resource Utilization

When any task is performed, it will likely use all the processing power available, internet until that function is finished

# Deployment

# 

# 5 Conclusion

During this article, we have developed a [predictive model](https://www.neuraldesigner.com/learning/tutorials/model-deployment) that can help retailers determine the number of sales they are going to make in the future. Using this model, retailers can plan the number of products they will need. As a consequence, the system will allow them to increase their profits. You can use [Neural Designer](https://www.neuraldesigner.com/) to build predictive models from your data and forecast your own company's sales or test it using the [data set](https://www.neuraldesigner.com/learning/tutorials/data-set) you can find below.