

Stores Sales Prediction

High Level Design

Domain: Machine Learning

Creator: Harshal S. Patil

Date: 02.06.2023

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date issued | Version | Description | Author |
| Jun 2, 2023 | 1.1 | First Draft | Harshal Patil |

Contents

Abstract ……………………………………………………3

Introduction ………………………………………………..3

What is High-Level Design Document?……………………3

Scope……………………………………………………….4

General Description………………………………………..4

Definitions………………………………………………….4

Product Description…………………………………………4

Problem Statement…………………………………………4

Proposed solution …………………………………………..5

Further improvements………………………………………5

Data requirements…………………………………………..5

Tools used…………………………………………………..6

Hardware Requirements…………………………………….7

Constraints…………………………………………………..7

Assumptions…………………………………………………7

Design Details……………………………………………….7

Process Flow………………………………………………...7

Model deployment…………………………………………..8

Event Log……………………………………………………8

Error Handling……………………………………………….9

Performance………………………………………………….9

Re usability……………………………………………..……9

Application Compatibility……………………………………9

Resource Utilization………………………………………….9

Dashboards………………………………………………….10

KPIs (Key Performance Indicators)………………………...10

Conclusion…………………………………………………..10

Abstract:

This document presents a solution for predicting sales in different stores of Big Mart using data mining techniques. The solution leverage a data warehouse that stores extensive consumer information and item details. By analysing this data, the solution aims to identify anomalies and common patterns to forecast future client demand and optimize inventory management. The proposed solution offers insights into predicting sales and enables the Big Mart management to make informed decisions for effective store operations.

# Introduction

## What is High-Level Design Document?

The goal of this HLD or a high-level design 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 HLD will:

* Present all of design aspects and define them in detail
* Describe all user interfaces being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and architecture of the project
* List and describe the non-functional attributes such as security, reliability, maintainability, portability, reusability, application compatibility. resource utilization, serviceability

## 

## Scope

The HLD documentation presents the structure of the system, such as 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 administrators of the system.

# General Description

## Definitions

|  |  |
| --- | --- |
| Term | Description |
| SSP | Stores Sales Prediction |
| Database | Collection of the Information |
| Cloud | A data center full of services connected to the internet performing service |
| IDE | Integrated Development Environment |
| UI | User Interface |
| Render | A cloud service |

## Product Description

SSP is a Machine Learning based regression model which helps us to do predictive analysis of the Sales using certain features and parameters.

## 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..

## Proposed solution

Using all the standard techniques used in the life cycle of a Data Science project starting from Data Exploration, Data Cleaning, Feature Engineering, Model Selection, Model Building and Model Testing and also building a Front end where a user can fill their information in the form input and get the output instantly.

## Further improvements

This SSP can be easily embedded inside any website or an application and everybody can get quick answer by inputting required data on friendly user interface.

This can be further improved by training more data in the model. Data can be acquired from kaggle.com

## Data requirements

Data requirement completely depend on our problem statement. We need the dataset from Kaggle.com to improve accuracy of the model. Required dataset should contain the following features:-

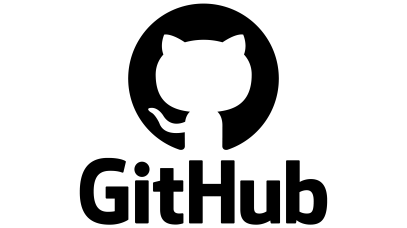
We have train (8523) and test (5681) data set, train data set has both input and output

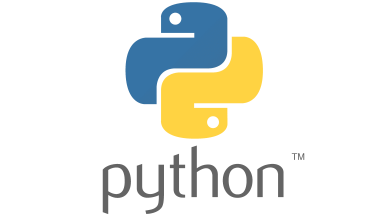
* variable(s). We need to predict the sales for test data set.
* Item\_Identifier: Unique product ID
* Item\_Weight: Weight of product
* Item\_Fat\_Content: Whether the product is low fat or not
* Item\_Visibility: The % of total display area of all products in a store allocated to the
* particular product
* 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.

## 

## Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit - learn, Flask, and a few other libraries were used to build the whole model.





* For visualization tasks, matplotlib, seaborn and plotly were used
* For visualization tasks, matplotlib, seaborn and plotly were used
* Flask were used for building the web application and server to run the code
* GitHub is used as version control system
* NumPy and Pandas were used to clean and interpret data
* Scikit-learn was used to cross validate and compare different models
* XGBoost Regressor was used to build the final model



## 

## Hardware Requirements

* Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
* Minimum 1.10 GHz processor or equivalent.
* Between 1-2 GB of free storage
* Minimum 1 MB of RAM
* 1 GB of hard-disk space

## Constraints

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

## Assumptions

The main objective of this project is to implement the use case as previously mentioned for new dataset that comes through the UI. It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible

# 

# Design Details

## Process Flow

For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below:

**Model Development**

Design UI On

Flask

Model

Evaluation

Cloud

Deployment on

Cloud

Hyper-

Parameter

Tuning

Desinging a

Server

Model

Implementation

Feature

Engineering

Exploratory Data

Analysis

Data

Preprocessing

**Deployment**

Render App to keep Server Running

Loading Pipeline on Github

Integrating Github pipeline code with Flask

Deploying The code on

Render

# Performance

The SSP tool is used to predict Sales is bad or good by providing numeric and Categorical Values. It can be used by various private agencies then it is supposed to be as accurate as possible. So that it doesn’t mislead authorities. Also, model retraining is very important to further enhance its performance.

## Re usability

The code written and the components used should have the ability to be reused with no problems.

## Application Compatibility

The different components for this project will be using Python as an interface between them, each component will have its own task to perform, and it is the job of Python to ensure proper transfer of information.

## Resource Utilization

When any task is performed, it will likely use all the processing power available to it until finished.

# Dashboards

As and when, the system starts to capture the historic/ periodic data for a user, the dashboards will be included display charts over time with progress on various indicators or factors.



## KPI (Key Performance Indicators)

# Sales Performance

# Inventory Turnover

# Customer Conversion Rate

# Store Traffic

# Sales by Location

# Return on Investment (ROI)

# Conclusion

In conclusion, leverage data mining techniques to analyse the vast amount of consumer information and item details stored in data warehouses provides valuable insights for shopping malls and Big Marts. By uncovering anomalies and common patterns, businesses can improve their demand forecasting accuracy and optimize inventory management practices. This optimization leads to enhanced operational efficiency, better client satisfaction, and increased profitability. Implementing a data-driven approach allows retailers to make informed decisions and stay ahead in a competitive market.