High Level Design (HLD)

Store Sales Prediction

Revision Number: 2.0

Last date of revision: 21 January 2023

MUHAMMED SHIBIL C V

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 27/12/2022 | 1.0 | Added Introduction and General description | Muhammed Shibil |
| 27/12/2022 | 1.1 | Added Design details and abstract | Muhammed Shibil |
| 29/12/2022 | 1.2 | Created and organized whole document | Muhammed Shibil |
| 02/01/2023 | 1.3 | Updated Design Details | Muhammed Shibil |
| 02/01/2023 | 1.4 | Checked and added Performance | Muhammed Shibil |
| 17/01/2023 | 1.5 | Added Conclusion and References | Muhammed Shibil |
| 17/01/2023 | 2.0 | Updated diagrams in Design Details | Muhammed Shibil |

**Table of Content**

Document Version Control ……………………………………………………………. 2

Abstract ………………………………………………………………………………………… 4

Introduction ………………………………………………………………………………….. 5

1. Why this High-Level Document? ……………………………………………. 5

2. Scope …………………………….………………………………………………………. 5

3. Definition ………………………………………………………………………………. 5

General Description ………………………………………………………………………. 6

1. Product Perspective ……………………………………………………….………. 6

2. Problem Statement ………………………………………………………..………. 6

3. Problem Solution ……………………………………………………………………. 6

4. Proposed Methodology …………………………………………………….……. 6

5. Further Improvements ……………………………………………………………. 6

6. Data Required ……………………………………………………...…………………. 6

7. Tools Used ………………………………………………………………………………. 7

8. Constraints …………………………………………………………………..…………. 7

9. Assumptions ……………………………………………………………………………. 7

Design Details ……………………………………………………………..…………………. 8

1. Process Workflow ………………………………………………………..…………. 8

2. Error Handling ………………………………………………………………..………. 9

Performance …………………………………………………………………………….……. 10

1. Reusability ……………………………………………………………..………………. 10

2. Application compatibility ………………………………………..………………. 10

3. Resources Utilization ………………………………………………………….……. 10

4. Deployment ……………………………………………………………………..…..…. 10

Conclusion ………………………………………………………………………….……..……. 11

Reference ………………………………………………………………………………..………. 12

**Abstract**

The problem of store sales prediction is a crucial one for retailers as it helps them make data-driven decisions to optimize their inventory, marketing, and other operational decisions. The goal of this project is to use historical sales data and other relevant information to predict future sales for a particular store or group of stores, and provide retailers with insights that can help them improve their business performance. The project scope includes data collection, cleaning and preprocessing, exploratory data analysis, modeling, evaluation, deployment and extract insights. The project aims to use techniques such as time series forecasting, regression analysis, and machine learning, to predict future sales and identify key factors that drive sales.

**Introduction**

**1. Why this High-Level Design Document?**

The purpose of this High-Level Design (HLD) 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 the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project

**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 administrators of the system.

**3. Definition**

The terms used in the projects are:

* MRP : Max. Retail Price

**General Description**

**1. Product Perspective**

The product perspective of the store sales prediction project is focused on providing a solution that can be used by retailers to make data-driven decisions and improve their business performance. The product will take the form of a software application or a platform that can be integrated with a retailer's existing systems. The product will provide retailers with the ability **2. Problem Statement**

To create the machine learning based solution to predict Store Sales Prediction based on the parameters.

**3. Problem Solution**

One problem that a life Store Sales Prediction tool could help solve is the lack of accurate and individualized information on Store Sales. This can make it difficult for individuals, Store Sales, and financial planners to make informed decisions about long-term financial planning..

**4. Further Improvement**

1. Incorporate machine learning and artificial intelligence techniques to improve the accuracy of predictions, by analyzing more data and providing more personalized recommendations.
2. Incorporate more data sources, such as environmental and socio-economic data, to provide a more complete picture of an individual's life expectancy.
3. Develop a user-friendly interface that is accessible to a wide range of users, including individuals, healthcare providers, and financial planners.

**5. Data Required**

1. The data required for a store sales prediction project typically includes:
2. Sales data: Historical sales data for a particular store or group of stores, including information such as the date of the sale, the product or category sold, and the revenue or quantity sold.
3. Economic indicators: Data on economic indicators such as GDP, inflation, and unemployment, which can provide insight into the broader economic context in which the store operates.
4. Promotional events: Information on promotional events such as sales, discounts, and marketing campaigns, which can affect the demand for products.
5. Competitor activity: Information on the activity of competitors, such as their prices, promotions, and store locations, which can affect the demand for products.

**6. Tools Used**

* Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn are used to build the whole model.
* PyCharm and Visual Studio Code is used as IDE.
* For visualization of the plots, Matplotlib and Seaborn are used.
* GitHub is used as version control system.

**7. Constraints**

1. Data availability: A lack of data, particularly in certain demographic or geographic areas, could limit the accuracy of predictions.
2. Data quality: Incorrect or incomplete data could lead to inaccurate predictions.
3. Privacy and ethical concerns: Obtaining and using personal information for life expectancy predictions may raise privacy concerns and ethical considerations, such as how the data will be used and protected.
4. Model's limitations: The Store sales prediction model may have some limitations in its ability to predict Sales, as it may not consider all factors that affect an individual's Store sales.

**8. Assumptions**

1. Linearity: The model may assume that the relationship between the predictors and life expectancy is linear, when in reality it may be more complex.
2. Stationarity: The model may assume that the underlying data is stationary, meaning that the statistical properties of the data do not change over time.
3. Independence: The model may assume that the predictors are independent of one another,

**Design Details**

**1. Error Handling**

When developing a Store sales prediction tool, it is important to consider how to handle errors and outliers in the data to ensure the accuracy of predictions. Here are a few strategies that can be used for error handling:

1. Data validation: Validate the data at the time of input to ensure that it is in the correct format and within acceptable ranges. This can help identify and correct errors before they are used in the prediction model.
2. Data cleaning: Clean and preprocess the data to remove outliers, duplicate data, and missing values that can affect the accuracy of predictions.

**Performance**

**1. Reusability**

Reusability is an important aspect to consider when developing a Store sales prediction tool, as it allows the tool to be used in a variety of different applications. Here are a few strategies to increase reusability:

1. Modularity: Break down the tool into smaller, modular components that can be easily reused in different contexts.
2. API: Create an API (Application Programming Interface) that allows other developers to easily access and use the tool's functionality in their own applications.

**2. Application compatibility**

When developing a Store sales prediction tool, it is important to consider the different applications in which the tool may be used, and to ensure that it is compatible with those applications.

**3. Resource utilization**

At the initial stage, we were using high space to create the model. Once the model is created, our system only needs at least of 2GB RAM and 1 GB of storage to run the application smoothly. Whenever user tries to predict the LE, system uses less than 10% of the processing power.

**4. Deployment**

The code is deployed in GitHub.

**Conclusion**

In conclusion, store sales prediction is a crucial problem for retailers as it helps them make data-driven decisions to optimize their inventory, marketing, and other operational decisions. The goal of this project is to use historical sales data and other relevant information to predict future sales for a particular store or group of stores, and provide retailers with insights that can help them improve their business performance. The project scope includes data collection, cleaning and preprocessing, exploratory data analysis, modeling, evaluation, deployment and extract insights. The project aims to use techniques such as time series forecasting, regression analysis, and machine learning, to predict future sales and identify key factors that drive sales. The ultimate goal is to provide actionable insights that can help retailers make data-driven decisions to improve their business performance.

**References**

1. <https://scikit-learn.org/stable/user_guide.html>
2. <https://numpy.org/doc/>
3. <https://seaborn.pydata.org/examples/regression_marginals.html>
4. <https://seaborn.pydata.org/examples/scatterplot_matrix.html>
5. <https://matplotlib.org/>
6. <https://pandas.pydata.org/docs/>