Store Sales Prediction

|  |  |
| --- | --- |
| Written by: | Mahin Prakash |
| Revision Number: | 1.0 |
| Last date of revision: | 8/07/2023 |

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Contents

[Document Version Control. 2](#_TOC_250021)

[Abstract. 4](#_TOC_250020)

1. [Introduction 5](#_TOC_250019)
   1. [Why this High-Level Design Document?. 5](#_TOC_250018)
   2. [Scope.](#_TOC_250017) 6
   3. [Definitions](#_TOC_250016) 6
2. [General Description. 6](#_TOC_250015)
   1. [Product Perspective 6](#_TOC_250014)
   2. [Problem statement 6](#_TOC_250013)
   3. [Proposed Solution](#_TOC_250012) 7
   4. [Data Requirements 7](#_TOC_250009)
   5. [Tools used.](#_TOC_250008) 7
   6. [Constraints](#_TOC_250007) 8
   7. [Assumptions.](#_TOC_250006) 8
3. [Design Details](#_TOC_250005) 8
   1. [Process Flow.](#_TOC_250004) 8
   2. [Deployment Process](#_TOC_250002) 9
   3. [Error Handling](#_TOC_250000) 9
   4. Performance. 9
   5. Reusability. 10
   6. Application Compatibility 10
   7. Resource Utilization 10
   8. Deployment. 10
4. Conclusion 11

Abstract

The project's concept revolves around forecasting the sales of a specific item in stores based on customer demand. This involves studying the historical sales data of the product, which big shopping malls maintain to anticipate future demand. The objective is to examine past records and identify significant correlations between various factors in order to develop a predictive system that can estimate the future demand for a particular product. Such a system would aid in efficiently managing warehouse storage capacity.

1. **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 of 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
    - List and describe the non-functional attributes like: o Security
      * Reliability
      * Maintainability
      * Portability
      * Reusability
      * Application compatibility
      * Resource utilization
      * Serviceability
  1. **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.

* 1. **Definitions**

|  |  |
| --- | --- |
| **Term** | **Description** |
| **DB** | Database, the cloud platform where we will store the data. |
| **ML** | Machine Learning |
| **Streamlit** | Python Framework |

### 

### General Description

#### **Product Perspective**

The Store Sales Prediction is a web application based on machine learning. It can forecast future product demand by examining previous records, providing a numerical estimate of product sales.

#### **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**

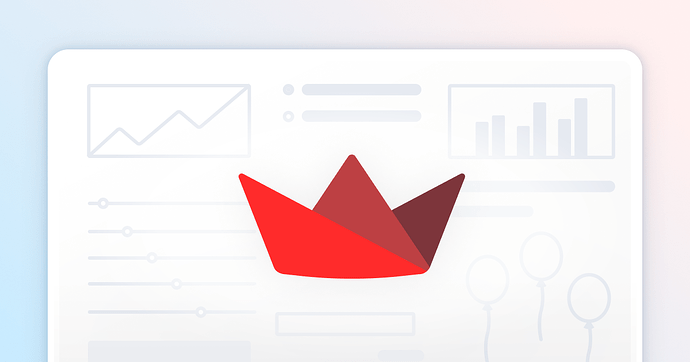
We will perform Exploratory Data Analysis (EDA) to discover significant relationships among various attributes. Additionally, we will employ a machine-learning algorithm to forecast future sales demand. The client will input the necessary features through a web application and obtain the results. The system will receive these features, validate and preprocess them in the backend, and subsequently feed them into a machine learning model that has been fine-tuned with hyperparameters to predict the ultimate outcome.

#### **Data Requirements**

The data required for the building of the project is already available on the dashboard. The Store Sales Prediction data records many product descriptions along with past sales quantity. For building the ml model we will use the dataset that is given. The data is consisting of 8523 rows and various information about products like product id, product category, store id, store location, e.t.c

#### **Tools used**

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Streamlit are used to build the whole model.



* + - For visualization of the plots, Matplotlib and Seaborn are used.
    - Streamlit community cloud is used for deployment of the model.
    - Front end development and backend development is handled using streamlit
    - GitHub is used as version control system.

#### **Constraints**

The system ought to be easy for users to navigate, ensuring they receive appropriate messages while using the web application. If a user makes a mistake on the web-app page, they should receive a clear error message. All errors and outcomes should be communicated in the simplest manner possible, and the buttons placed on the webpage should be appropriately labeled to prevent user confusion while utilizing the system.

#### **Assumptions**

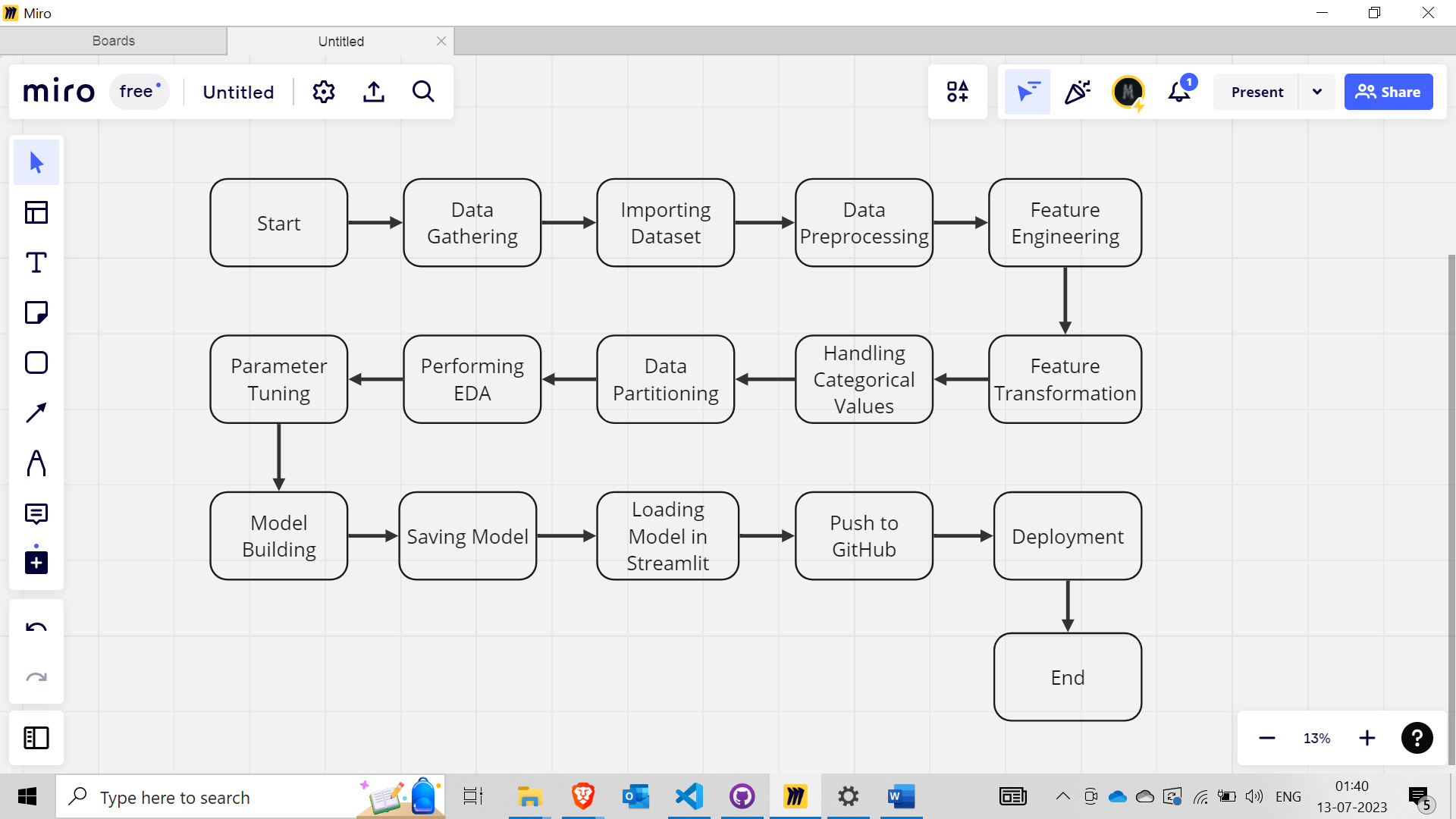
The primary goal is to create a system that generates estimated future demand for a product in stores.

#### **Design Details**

#### **Process Flow**

We will be using following process flow for this project. The process will be based on modular coding i.e. use of oops concepts to build the entire project from start to end.

Proposed methodology



#### **Deployment Process**

A picture containing diagram, text, plan, screenshot

Description automatically generated

#### **Error Handling**

If an error occurs during the processing, it is important to display a user-friendly error message that can be understood by anyone. The error message should provide meaningful information so that the user can identify their mistake and retry the process with improvements. It is crucial to handle all the errors properly, and we must log every error for our application and ensure proper management of them.

#### **Performance**

The prediction of the sales price relies on machine learning algorithms. We will train different algorithms and determine the most suitable one for predicting the desired outcome. The performance of our system will be determined by the data we provide to the algorithms, as well as the chosen model, web application, and deployment server. By ensuring the proper functioning of all these components, our program should operate smoothly.

#### **Reusability**

The code and module developed during the project construction phase must adhere to coding guidelines, ensuring the entire project code is written in a modular manner. Our system needs to be flexible enough to function effectively from any location. It should be capable of handling erroneous user input and providing clear error messages, allowing users to correct their mistakes and enter valid input to obtain accurate results. Additionally, the system should be reusable in various scenarios, accommodating different types of input values for which it has been trained.

#### **Application Compatibility**

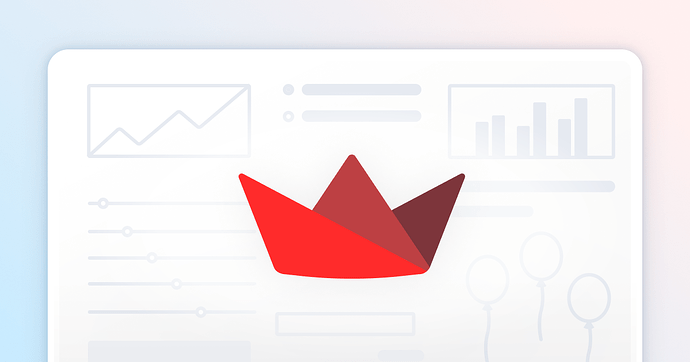
Various libraries and Python programming languages are employed to construct the system. Each library possesses its unique functionality, and it must function effectively within our dynamic system. Django will be utilized for creating the web APIs, while HTML/CSS will be employed for crafting the web application. Every element of the application needs to operate correctly and yield results without any ambiguity.

#### **Resource Utilization**

Our application needs to efficiently use the available resource and minimize internet usage for accessing APIs on the web page. It should avoid consuming excessive computational resources to prevent slowing down the application. Additionally, when deployed on a cloud platform, the application should effectively utilize the provided cloud resources to ensure proper functionality.

#### **Deployment**

For the deployment process, we will using Streamlit Community Cloud for hosting our application. The cloud platform will run the system and it will give the flexibility to use our application globally.



#### **Conclusion**

The Sales Store Prediction aims to assist business owners and manufacturing companies in forecasting future product demand. This prediction tool not only aids in business growth but also supports an efficient supply chain for products. By utilizing our extensive collection of product sales records and store information, we analyze historical data to construct a machine learning model capable of identifying internal patterns. This model enables us to predict the future target value or sales demand for the product.