Store Sales Prediction

High Level Design

Shruti Balan

17 February, 2023

Document Version Control

Date Issued	Version	Description	Author
17 February 2023	1.0	Initial HLD	Shruti Balan

Contents

Abstract

Nowadays, shopping malls and supermarkets 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. This project discusses the implementation of a model which predicts the sales of a given product based on factors such as the fat content, weight, type of outlet the item is sold and other outlet characteristics.

1. Introduction

1.1. 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.

1.2. The HLD will:

- Present all of the design aspects and define them in detail
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes

  like:
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilization
  - Serviceability

1.3. 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.

2. General Description

2.1.Product Perspective

The Store Sales Prediction system is a machine learning-based regression model which will help us to predict the sales of a particular product based on certain features pertaining to the item and outlet in which the item is displayed.

2.2. Problem statement

3. Nowadays, shopping malls and supermarkets 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. The goal is to predict the sales of a particular product based on certain features pertaining to the item and outlet in which the item is displayed.

3.1. Proposed Solution

The solution proposed here is a web application, which predict the sales of a particular product based on certain features pertaining to the item and outlet in which the item is displayed.

3.2. Data Requirements

This dataset is taken from the UCI Machine Learning Repository (url: https://www.kaggle.com/datasets/brijbhushannanda1979/bigmart-sales-data).
There are 21 variables:

- **Item_Identifier :** Unique product ID.
- **Item_Weight** weight of the product (quantitative)
- **Item_Fat_Content** : Whether the fat is low fat or not (categorical)
  - **Regular**
  - **Low Fat**

- **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 the store was established.
- **Outlet_Type :** Whether the outlet is just a grocery store or some sort of supermarket.
- **Oulet_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.
- **Item_Outlet_Sales :** Sales of a product in a particular store. This is the target variable.

3.3. Tools used

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

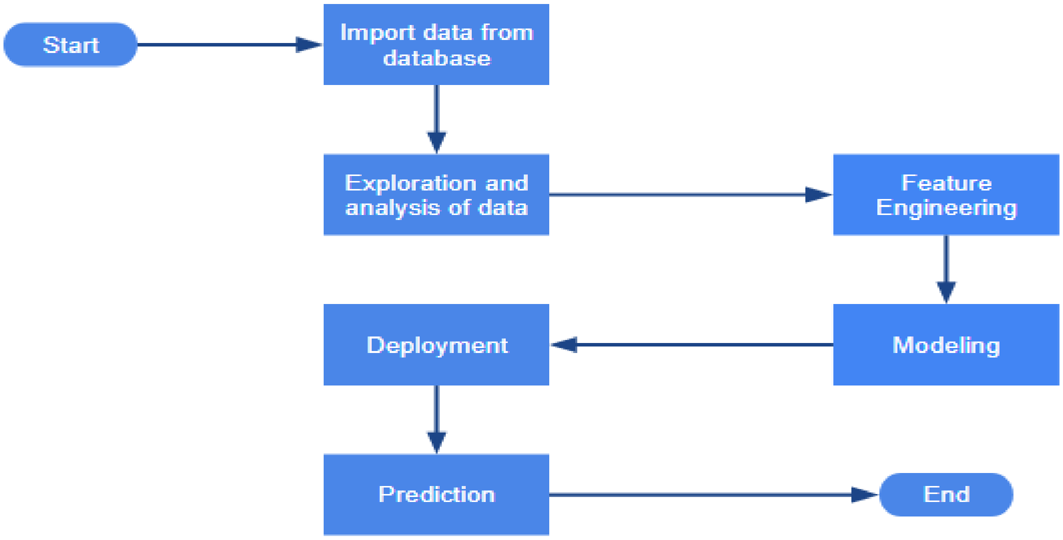
- Jupyter Lab is used as IDE.
- For visualization of the plots, Matplotlib and Seaborn are used.
- RailwayApp is used for deployment of the model.
- Front end development is done using HTML/CSS
- Python is used for backend development.
- Cassandra is used as database
- GitHub is used as version control system.

4. Design Details

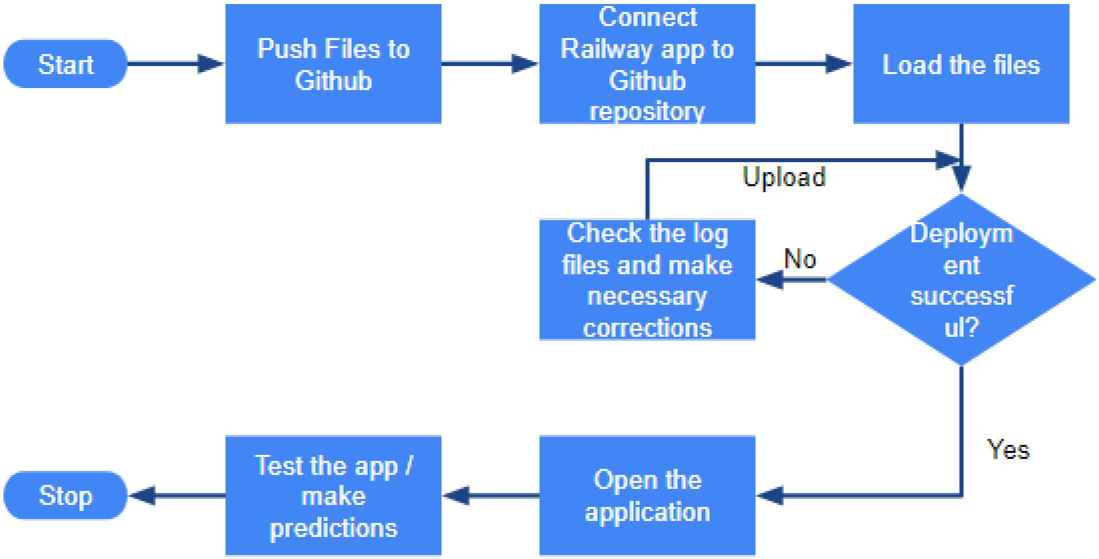
4.1. Process Flow

For identifying the class of each profile, we will use a machine learning model.
Below is the process flow diagram is as shown below.

4.2. Proposed methodology



4.3. Deployment Process


Start
Push Files to Github
Connect Railway app to Github repository
Load the files
Upload
Deployment successful?
No
Check the log files and make necessary corrections
Yes
Open the application
Test the app / make predictions
Stop

4.4. Event Log

The event logs are stored in log files.

4.5. Performance

The Sales Prediction app is used to predict the sales of a particular item in a particular store based on features pertaining to the item and outlets.

4.6. Reusability

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

4.7. 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 the Python to ensure proper transfer of information.

4.8. Resource Utilization

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

4.9. Deployment

The model can be deployed in any cloud services such as Microsoft Azure, AWS, Google, Heroku etc.

5. Conclusion

This application will classify profiles/applicants for credit into Good Risk and Bad Risk categories, and can help financial institutions in taking necessary actions to prevent further loss.