Store Sales Prediction

Low Level Design

Shruti Balan
28 February, 2023

Contents

Document Version Control

Date Issued	Version	Description	Author
28 February 2023	1.0	Initial LLD	Shruti Balan

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 Low-Level Design Document?

The purpose of this document is to present a detailed description of the Store Sales Prediction application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate etc. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

1.2. Scope

This software system will be a Web application. This system is designed to predict the sales of a particular product based on certain features pertaining to the product and the outlet in which it is displayed.

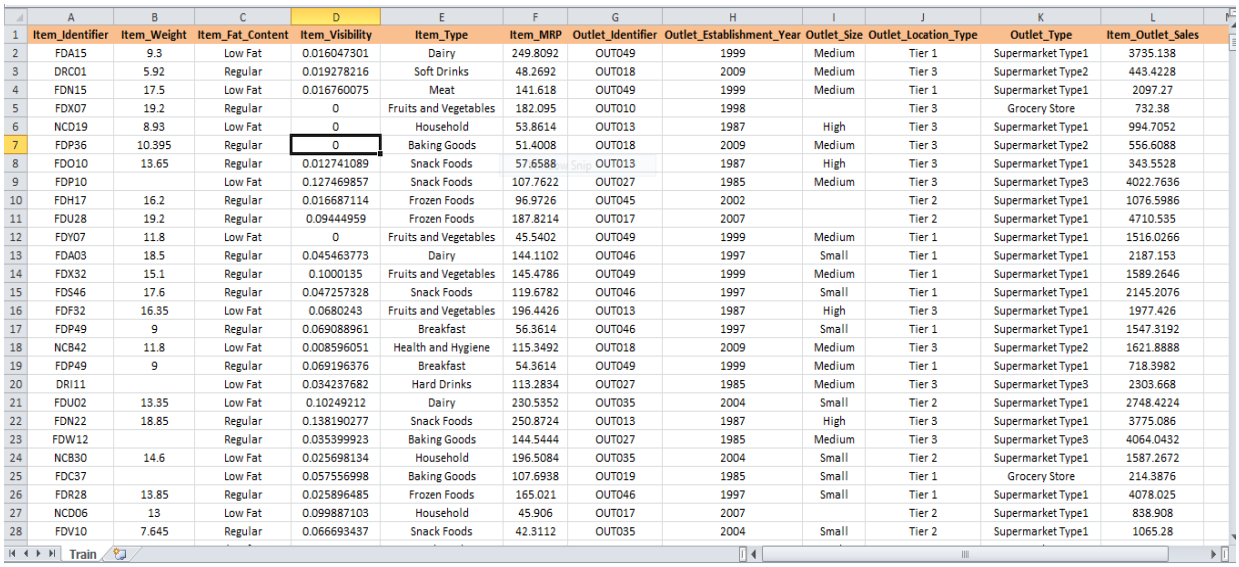
2. Technical Specifications

2.1. Dataset

File Name	Finalized	Source
Train.csv	Yes	https://www.kaggle.com/datasets/brijbhushannanda1979/bigmart-sales-data

2.1.1. Dataset Overview

The data obtained from the repository is in the form of .csv file, from which the data is extracted and stored in the Cassandra database. It contains the data of about 1000 products.



Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
FDA15	9.3	Low Fat	0.016047301	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.138
DRC01	5.92	Regular	0.019278216	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
FDN15	17.5	Low Fat	0.016760075	Meat	141.618	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.27
FDX07	19.2	Regular	0	Fruits and Vegetables	182.095	OUT010	1998		Tier 3	Grocery Store	732.38
NCD19	8.93	Low Fat	0	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052
FDP36	10.395	Regular	0	Baking Goods	51.4008	OUT018	2009	Medium	Tier 3	Supermarket Type2	556.6088
FDO10	13.65	Regular	0.012741089	Snack Foods	57.6588	OUT013	1987	High	Tier 3	Supermarket Type1	343.5528
FDP10		Low Fat	0.127469857	Snack Foods	107.7622	OUT027	1985	Medium	Tier 3	Supermarket Type3	4022.7636
FDH17	16.2	Regular	0.016687114	Frozen Foods	96.9726	OUT045	2002		Tier 2	Supermarket Type1	1076.5986
FDU28	19.2	Regular	0.09444959	Frozen Foods	187.8214	OUT017	2007		Tier 2	Supermarket Type1	4710.535
FDY07	11.8	Low Fat	0	Fruits and Vegetables	45.5402	OUT049	1999	Medium	Tier 1	Supermarket Type1	1516.0266
FDA03	18.5	Regular	0.045463773	Dairy	144.1102	OUT046	1997	Small	Tier 1	Supermarket Type1	2187.153
FDX32	15.1	Regular	0.1000135	Fruits and Vegetables	145.4786	OUT049	1999	Medium	Tier 1	Supermarket Type1	1589.2646
FDS46	17.6	Regular	0.047257328	Snack Foods	119.6782	OUT046	1997	Small	Tier 1	Supermarket Type1	2145.2076
FDF32	16.35	Low Fat	0.0680243	Fruits and Vegetables	196.4426	OUT013	1987	High	Tier 3	Supermarket Type1	1977.426
FDP49	9	Regular	0.069088961	Breakfast	56.3614	OUT046	1997	Small	Tier 1	Supermarket Type1	1547.3192
NCB42	11.8	Low Fat	0.008596051	Health and Hygiene	115.3492	OUT018	2009	Medium	Tier 3	Supermarket Type2	1621.8888
FDP49	9	Regular	0.069196376	Breakfast	54.3614	OUT049	1999	Medium	Tier 1	Supermarket Type1	718.3982
DRI11		Low Fat	0.034237682	Hard Drinks	113.2834	OUT027	1985	Medium	Tier 3	Supermarket Type3	2303.668
FDU02	13.35	Low Fat	0.10249212	Dairy	230.5352	OUT035	2004	Small	Tier 2	Supermarket Type1	2748.4224
FDN22	18.85	Regular	0.138190277	Snack Foods	250.8724	OUT013	1987	High	Tier 3	Supermarket Type1	3775.086
FDW12		Regular	0.035399923	Baking Goods	144.5444	OUT027	1985	Medium	Tier 3	Supermarket Type3	4064.0432
NCB30	14.6	Low Fat	0.025698134	Household	196.5084	OUT035	2004	Small	Tier 2	Supermarket Type1	1587.2672
FDC37		Low Fat	0.057556998	Baking Goods	107.6938	OUT019	1985	Small	Tier 1	Grocery Store	214.3876
FDR28	13.85	Regular	0.025896485	Frozen Foods	165.021	OUT046	1997	Small	Tier 1	Supermarket Type1	4078.025
NCD06	13	Low Fat	0.099887103	Household	45.906	OUT017	2007		Tier 2	Supermarket Type1	838.908
FDV10	7.645	Regular	0.066693437	Snack Foods	42.3112	OUT035	2004	Small	Tier 2	Supermarket Type1	1065.28

2.1.2. Input Schema

Feature Name	Datatype	Null/Required
Item_Identfier	Integer	Required
Item_Weight	Integer	Required
Item_Fat_Content	Integer	Required
Item_Visibility	Integer	Required
Item_Type	Integer	Required
Item_MRP	Integer	Required
Outlet_Identifier	Integer	Required
Outlet_Establishment_Year	Integer	Required
Outlet_Type	Integer	Required
Outlet_Location_Type	Integer	Required
Outlet_Size	Integer	Required
Item_Outlet_Sales	Integer	Required

2.2. Predicting Sales

- The system presents the set of inputs to the user.
- The user gives required information.
- The system should be able to predict the sales for the particular product.

2.3. Logging

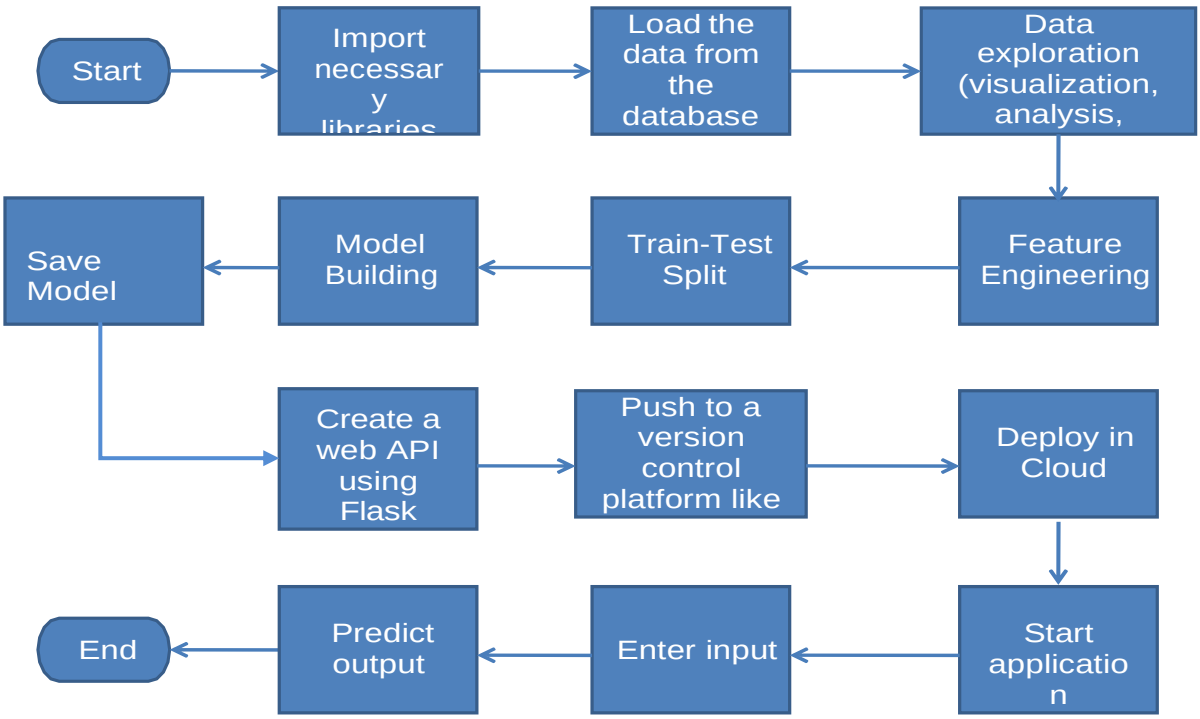
We should be able to log every activity done by the user.

- The System identifies at what step logging required.
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

2.4. Deployment

The application is deployed in Railway app.

3. Architecture



4. Architecture Description

4.1. Data Description

This dataset is taken from the Kaggle (url: https://www.kaggle.com/datasets/brijbhushannanda1979/bigmart-sales-data). It contains information on the items and the outlet in which the item is sold/displayed. There are 12 variables:

Content There are 12 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.

4.2. Data Exploration

We divide the data into two types: numerical and categorical. We explore through each type one by one. Within each type, we explore, visualize and analyze each variable one by one and note down our observations.

4.3. Feature Engineering

Encoded categorical variables.

4.4. Train/Test Split

Split the data into 70% train set and 30% test set.

4.5. Model Building

Built models and trained and tested the data on the models.
Compared the performance of each model and selected the best one.
Feature importance and/or hyper-parameter tuning performed to improve the performance of the selected model.

4.6. Save the model

Saved the model by converting into a pickle file.

4.7. Cloud Setup & Pushing the App to the Cloud

Selected Railway for deployment.

Used the model to develop a flask application which can predict sales for unseen data, pushed to Github and from there, deployed the application files to Railway app.

4.8. Application Start and Input Data by the User

Start the application and enter the inputs.

4.9. Prediction

After the inputs are submitted the application runs the model and makes predictions. The out is displayed as a message indicating the sale price at which the product will be sold.

5. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to see input fields on logging in	1. Application URL is accessible 2. Application is deployed	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application URL is accessible 2. Application is deployed	User should be able to edit all input fields
Verify whether user gets submit button to submit the inputs	1. Application URL is accessible 2. Application is deployed	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application URL is accessible 2. Application is deployed	User should be presented with recommended results on clicking submit