Store Sales Prediction

Project Report Presentation

Prepared by : Shruti Balan,
Data Science Intern,
Ineuron

Agenda

- Introduction
- Objective
- Data Description
- Architecture
- Model Training and Evaluation Workflow
- Deployment
- Questions

Introduction

- 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.

Objective

- Development of a model for predicting the sales of particular item.
- Benefits:
  - Retail units can understand the properties of products and stores which play key role in increasing sales.
  - Helps to anticipate potential consumer demand and update inventory management.
  - Helps to forecast future sales volume.

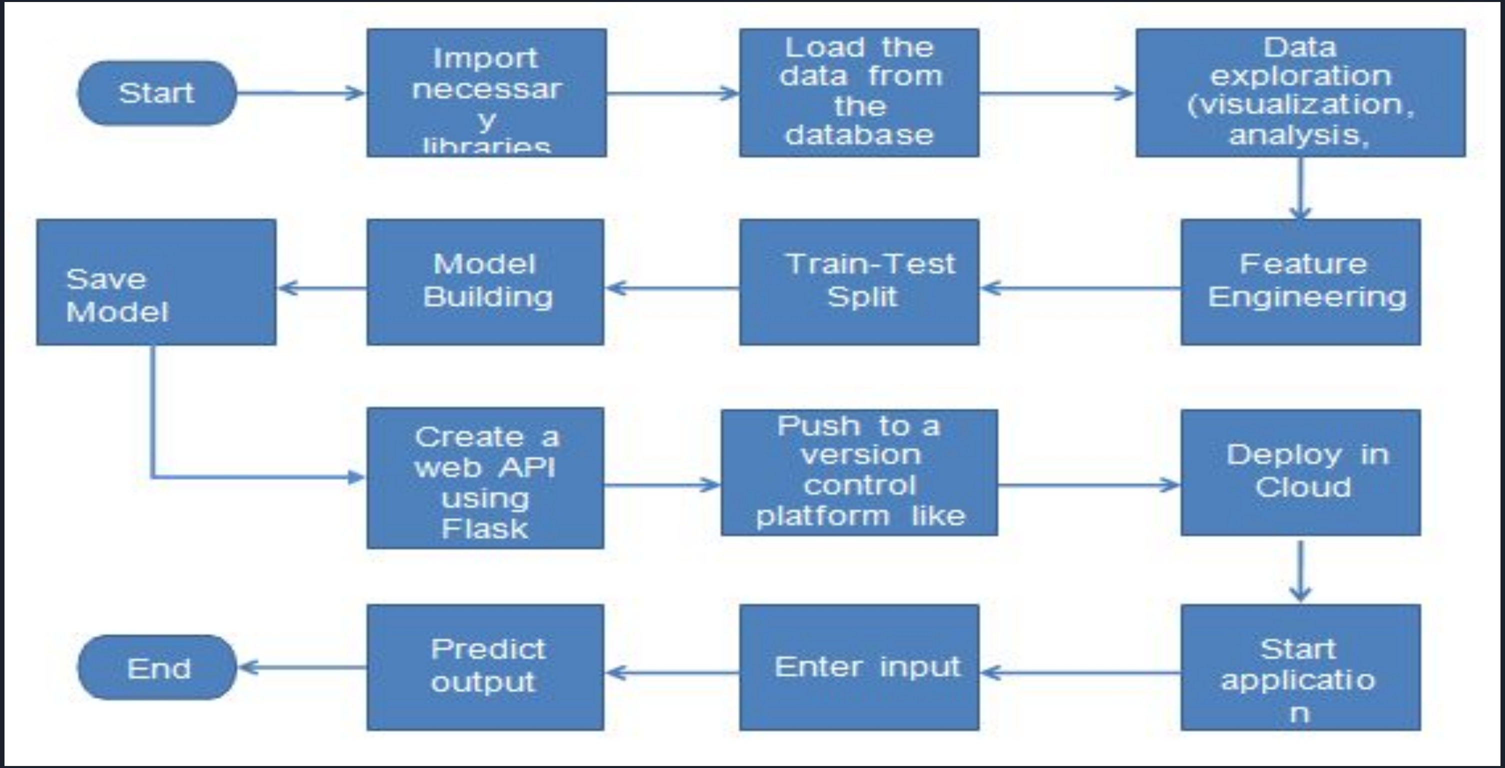
Data Description

- **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.

Data Description

- **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.

Architecture


Start
Import necessar y libraries
Load the data from the database
Data exploration (visualization, analysis,
Feature Engineering
Train-Test Split
Model Building
Save Model
Create a web API using Flask
Push to a version control platform like
Deploy in Cloud
Start applicatio n
Enter input
Predict output
End

Architecture

- **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.

- **Feature Engineering**

  Encoded categorical variables.

- **Train/Test Split**

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

Architecture

- **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.

- **Save the model**

  Saved the model by converting into a pickle file.

- **Create a Web API**

  The model is used to create a web API using which the users can interact with the application. In this project, Flask has been used for the purpose.

Architecture

- **Cloud Set-up and pushing the app to cloud**
  - Selected Railway for deployment.
  - Used the model to develop a flask application which can predict sales for unseen data.
  - Pushed to Github using and from there deployed the application files from Github to Railway app.

Architecture

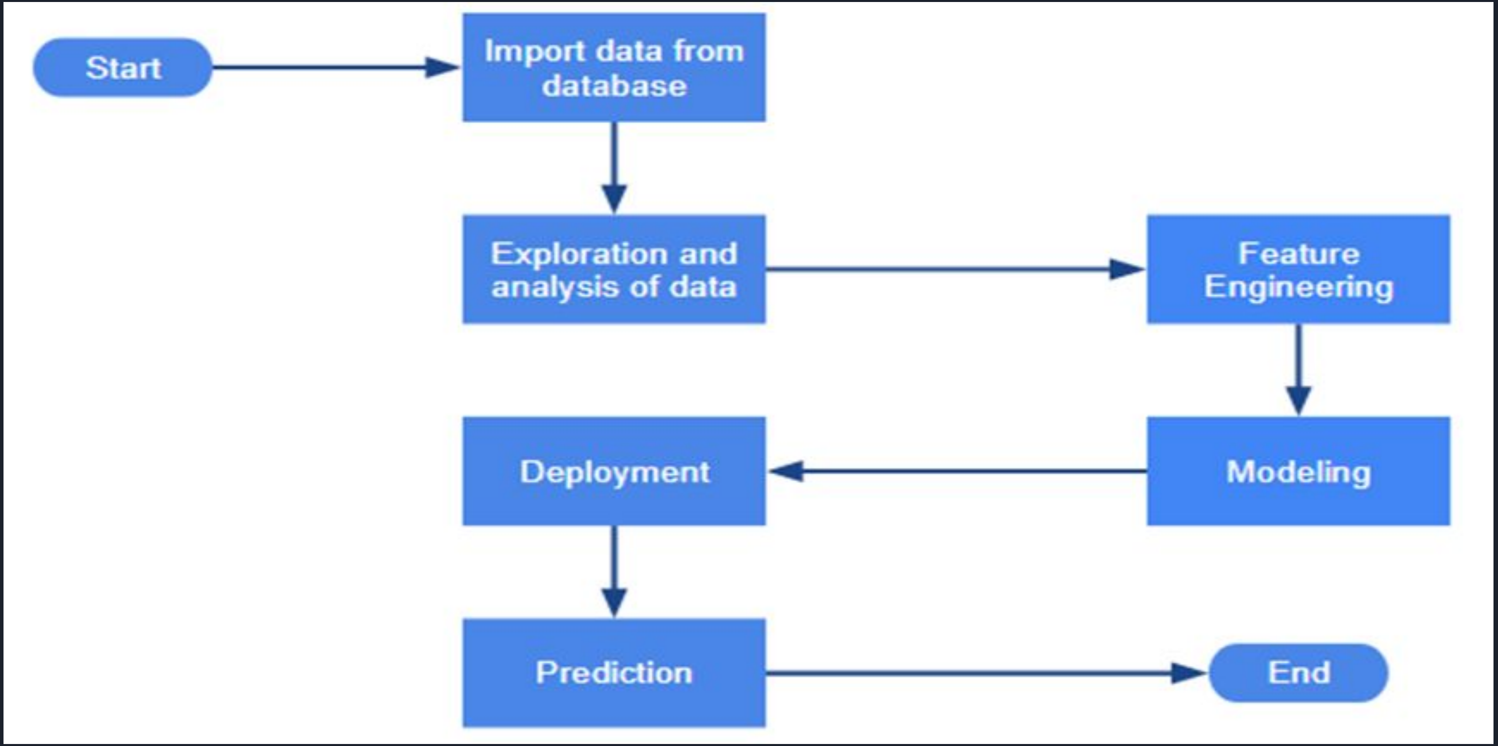
- ➢ **Application Start & Input data by the User**

  Start the application and enter the inputs.

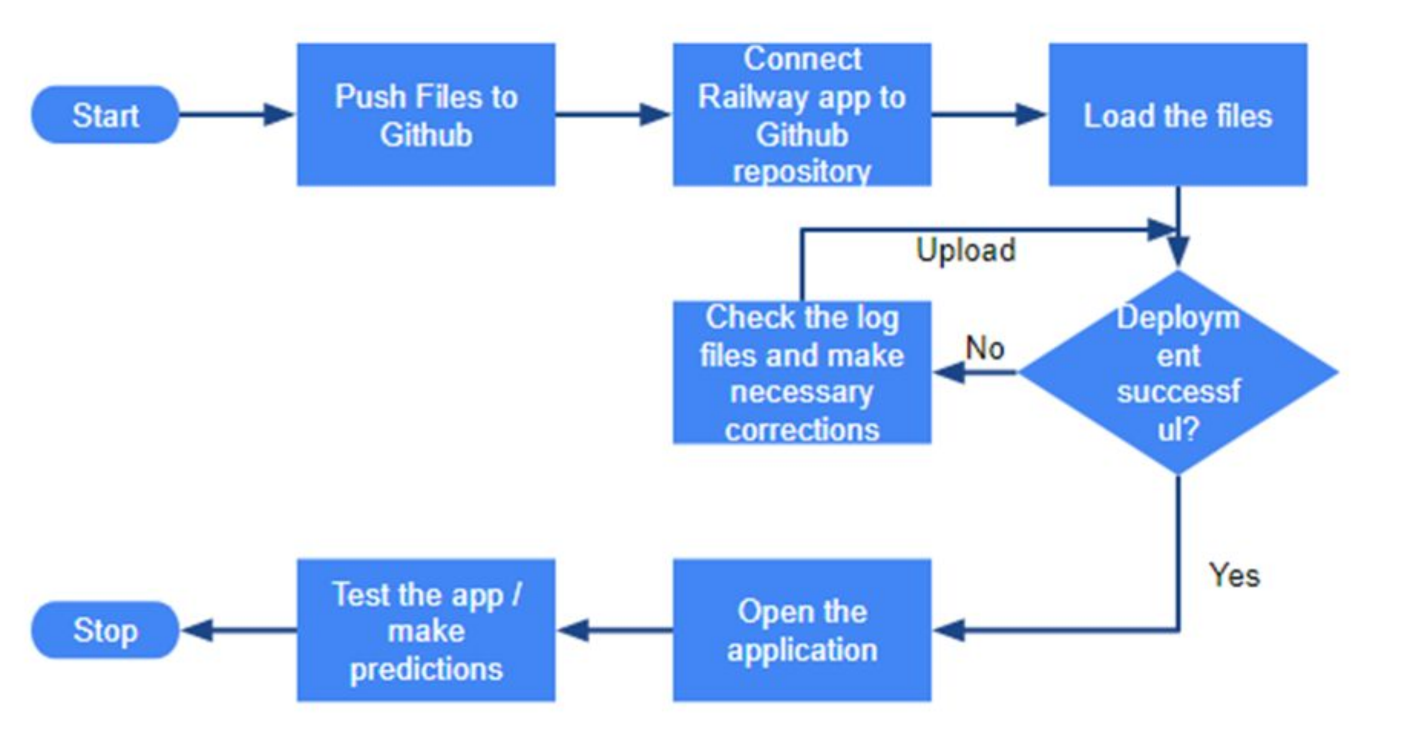
- ➢ **Prediction**

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

Model Training & Evaluation



Deployment


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

FAQ

1)**What is the data source?**

The data is obtained from Kaggle.

Link : https://www.kaggle.com/datasets/brijbhushannanda1979/bigmart-sales-data

2)**What was the type of data?**

The data contained both numerical and continuous type data.

3)**What was the complete flow that you followed in this project?**

Please refer to slides 13 to 15.

4)**How logs are managed?**

We have a separate log files for each stage of the project.

FAQ

5)**What techniques were you using for data pre-processing?**

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Cleaning data and imputing if null values are present.
- Encoding categorical variables

6)**How training was done or what models were used?**

- After loading the dataset, data pre-processing was done.
- For this project, we opted to train the data using the XGBoost Classifier.
- Hyper-parameter tuning, feature selection were performed during the various versions of modeling.
- The best model was selected.

FAQ

7)**How Prediction was done?**

- The test files were provided.
- The test data also underwent preprocessing.
- Then the data was passed through the model and output was predicted.

8)**What are the different stages of deployment?**

- After training the model, we prepared all the necessary files required for deployment and uploaded in a document version control system called Github.
- We then connected to and deployed the model in, Heroku.

THANK YOU