# Sales Forecasting using Big Mart Sales Data

**Team Members:**

Rakesh Maddineni

Rohit Kumar Mupparapu

Sadiq Shaik

Chakri Venkat Katam

GitHub repository:- [big\_mart\_sales\_forecasting](https://github.com/Rohitchow/big_mart_sales_forecasting.git)

<https://github.com/Rohitchow/big_mart_sales_forecasting.git>

Table of Content:

[Sales Forecasting using Big Mart Sales Data 1](#_Toc152681563)

[1. Introduction 3](#_Toc152681564)

[1.1 Brief Overview of the Project 3](#_Toc152681565)

[1.2 Importance of Sales Forecasting in Retail 3](#_Toc152681566)

[2. Background 3](#_Toc152681567)

[2.1 Overview of Big Mart and its Business Context 3](#_Toc152681568)

[2.2 Significance of the Dataset Used 4](#_Toc152681569)

[3. Data Overview 4](#_Toc152681570)

[3.1 Description of the Dataset 4](#_Toc152681571)

[3.1.1 Initial Observations and Data Characteristics 4](#_Toc152681572)

[3.2 Data Preprocessing 4](#_Toc152681573)

[3.2.1 Steps Taken for Data Cleaning, Handling Missing Values, etc. 5](#_Toc152681574)

[3.2.2 Exploratory Data Analysis (EDA) 5](#_Toc152681575)

[3.2.3 Analysis of Key Variables 6](#_Toc152681576)

[3.2.4 Identification of Patterns and Insights 6](#_Toc152681577)

[3.3 Feature Engineering 7](#_Toc152681578)

[4. Model Building and Evaluation 8](#_Toc152681579)

[4.1 Description of the Models Used 8](#_Toc152681580)

[4.2 Model Evaluation and Comparison 8](#_Toc152681581)

[5. Implementation 8](#_Toc152681582)

[5.1 How the Model is Implemented in the Provided Python Script 8](#_Toc152681583)

[5.2 Key Features and Components 8](#_Toc152681584)

[5.3 How to Use the Application 9](#_Toc152681585)

[5.4 Results and Analysis 9](#_Toc152681586)

[Key Features of the Dashboard 9](#_Toc152681587)

[Interactive Graphs and Charts: 9](#_Toc152681588)

[Data Grouping and Analysis: 9](#_Toc152681589)

[Trend Analysis: 10](#_Toc152681590)

[User-Friendly Interface: 10](#_Toc152681591)

[app.py: 10](#_Toc152681592)

[Running of app.py: 11](#_Toc152681593)

[Screenshot of the dashboard page: 11](#_Toc152681594)

[6. Conclusion 13](#_Toc152681595)

[6.1 Summary of Findings 13](#_Toc152681596)

[6.2 Final Thoughts on Project Outcomes 13](#_Toc152681597)

[6.3 Challenges 13](#_Toc152681598)

[6.4 Lessons Learned 13](#_Toc152681599)

[6.5 Scope for Improvement/Extension 13](#_Toc152681600)

## 1. Introduction

### 1.1 Brief Overview of the Project

This project entails the development of a sales forecasting model using historical sales data from Big Mart, a prominent retail chain. The model aims to predict sales based on various factors such as item characteristics, outlet details, and more. Utilizing Python and data mining techniques, the project involves stages of data preprocessing, exploratory data analysis (EDA), feature engineering, model building, and evaluation.

### 1.2 Importance of Sales Forecasting in Retail

Sales forecasting is a critical aspect of retail management. It enables retailers to make informed decisions regarding inventory management, marketing strategies, budget planning, and resource allocation. Effective sales forecasting helps in reducing the risk of stockouts or overstock situations, ensuring optimal use of resources, and maximizing profits. In the competitive retail industry, accurate sales forecasts are essential for strategic planning, maintaining customer satisfaction, and achieving sustainable growth.

## 2. Background

### 2.1 Overview of Big Mart and its Business Context

Big Mart, a well-established name in the retail industry, operates a chain of supermarkets and hypermarkets. With its wide range of products, including groceries, apparel, electronics, and home appliances, Big Mart caters to a diverse customer base. The company's success hinges on its ability to understand market trends, customer preferences, and efficiently manage its vast inventory across different geographical locations.

### 2.2 Significance of the Dataset Used

The dataset used in this project comprises sales records from various Big Mart outlets. It includes details such as item attributes, outlet information, and historical sales figures. This dataset provides a comprehensive view of sales dynamics, making it a valuable resource for developing a sales forecasting model. The insights gained from this dataset are instrumental in helping Big Mart optimize its operations and sales strategies.

## 3. Data Overview

### 3.1 Description of the Dataset

The dataset used for this project is sourced from Big Mart's historical sales records. It comprises various features, including item attributes (like item identifier, weight, fat content, type, and visibility), outlet characteristics (such as outlet identifier, size, location type, and type), and the target variable, item outlet sales. This comprehensive dataset enables a detailed analysis of factors influencing sales, essential for building an accurate forecasting model.

### 3.1.1 Initial Observations and Data Characteristics

Initial observations reveal a mix of numerical and categorical variables. The dataset contains missing values in certain columns, such as 'Item\_Weight' and 'Outlet\_Size', necessitating careful preprocessing. The diversity of features, from product specifications to outlet details, offers a multi-dimensional view of the sales ecosystem.

### 3.2 Data Preprocessing

Data preprocessing involved handling missing values, normalizing categorical data, and ensuring data consistency. Missing values in 'Item\_Weight' were filled with the mean weight, while 'Outlet\_Size' missing values were replaced with the mode. Additionally, inconsistencies in the 'Item\_Fat\_Content' category were rectified by standardizing various representations into uniform categories.

### 3.2.1 Steps Taken for Data Cleaning, Handling Missing Values, etc.

The steps taken for data cleaning included analyzing each feature for missing or inconsistent data. After identifying missing values, appropriate imputation strategies were employed: mean imputation for continuous variables and mode imputation for categorical variables. For inconsistent categorical data, standardization techniques were applied to maintain data uniformity.

A close-up of a computer code

Description automatically generated

### 3.2.2 Exploratory Data Analysis (EDA)

The EDA phase involved a thorough investigation of the dataset. Numerical features like 'Item\_Weight' and 'Item\_Visibility' were analyzed using histograms to understand their distribution. Categorical features, such as 'Outlet\_Size', were explored through count plots, providing insights into the frequency distribution of different categories.

A screenshot of a graph

Description automatically generated

### 3.2.3 Analysis of Key Variables

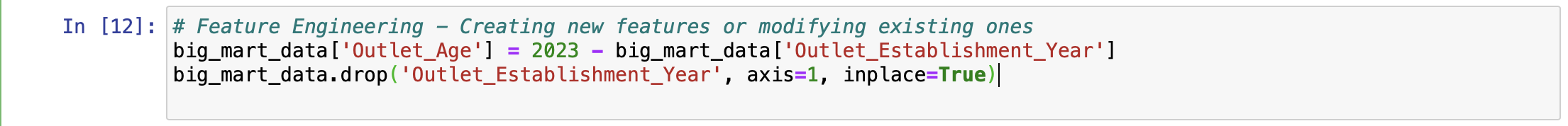
Key variables were analyzed to uncover trends and patterns. This involved examining the distribution of variables like 'Item\_Fat\_Content' and 'Item\_Type', and exploring relationships between features and the target variable 'Item\_Outlet\_Sales'. Such analysis helped in identifying potential predictors for the sales forecasting model.

A screenshot of a computer

Description automatically generated

### 3.2.4 Identification of Patterns and Insights

The identification of patterns and insights was a crucial outcome of the EDA. It revealed, for instance, the impact of item visibility on sales, and the influence of outlet characteristics on item demand. These insights were pivotal in guiding the subsequent feature engineering and model building phases.



In summary, this feature engineering step transforms the dataset by replacing the 'Outlet\_Establishment\_Year' column with a new 'Outlet\_Age' column, providing a potentially more informative feature for analysis and model building.

### 3.3 Feature Engineering

Feature engineering involved creating new features and transforming existing ones to enhance the model's predictive power. A significant step was the creation of 'Outlet\_Age', derived from the 'Outlet\_Establishment\_Year', providing a more direct measure of the outlet's age. Categorical variables were encoded into numerical format using label encoding, making them suitable for algorithmic processing.

A screen shot of a graph

Description automatically generated

## 4. Model Building and Evaluation

### 4.1 Description of the Models Used

The project utilized the Random Forest Regressor for building the sales forecasting model. Random Forest, an ensemble learning method, operates by constructing multiple decision trees during training and outputting the average prediction of the individual trees. This model was chosen for its robustness to overfitting and its ability to handle large datasets with numerous features effectively.

### 4.2 Model Evaluation and Comparison

The model's performance was evaluated using the R2 score, a statistical measure that represents the proportion of the variance for the dependent variable that's explained by the independent variables in a regression model. The R2 score provides an insight into the accuracy of the model's predictions. An extensive evaluation process was conducted to ensure the model's reliability and effectiveness in forecasting sales.

## 5. Implementation

### 5.1 How the Model is Implemented in the Provided Python Script

The implementation of the model in Python involved several key steps. After data preprocessing and feature engineering, the dataset was split into training and testing sets. The Random Forest Regressor was then trained on the training set. The model's prediction capabilities were harnessed in a Python script, which could be used to make sales forecasts based on new input data.

### 5.2 Key Features and Components

The project's key components include data preprocessing modules, exploratory data analysis techniques, feature engineering methods, and the Random Forest regression model. These components collectively contribute to the project's capability to accurately predict sales.

### 5.3 How to Use the Application

A step-by-step guide on using the application is provided within the Python script. Users can input the relevant data pertaining to item and outlet characteristics, and the script will output a sales forecast. This user-friendly approach ensures that the model can be effectively utilized for practical sales forecasting needs.

# 5.4 Results and Analysis

The 'app.py' file contains code for a web-based application, built using a Python framework called Dash. This application is essentially a dashboard - a tool that displays important information in an easy-to-understand way. The purpose of this dashboard is to show data from the 'Train.csv' file, which has lots of information about sales in different stores.

## Key Features of the Dashboard

## Interactive Graphs and Charts:

What It Does: The dashboard uses a library called Plotly to make graphs and charts that users can interact with. For example, you can click on parts of the graph to see more details.

Why It's Important: These interactive elements make it easier to understand complex data. They help in spotting trends and patterns in sales, like which products are selling the best.

## Data Grouping and Analysis:

What It Does: The application organizes the sales data in various ways. It can group sales by the type of item, the type of store, and other categories.

Why It's Important: This helps in understanding how different products or stores are performing. For instance, it can show which type of store sells the most of a certain product.

## Trend Analysis:

What It Does: The dashboard can show how sales have changed over time. This includes looking at sales trends over different years and comparing different kinds of sales data.

Why It's Important: This is useful for predicting future sales and understanding how different factors affect sales.

## User-Friendly Interface:

What It Does: The design of the dashboard is simple and easy to use, even for people who are not experts in data analysis.

Why It's Important: This makes the data accessible to everyone, not just experts. It helps different team members make informed decisions based on the data.

# app.py:

A screenshot of a computer program

Description automatically generated

## Running of app.py:

A screenshot of a computer program

Description automatically generated

## Screenshot of the dashboard page:

A screenshot of a computer

Description automatically generated

Here in this page, we can see the graphs and trends using the dashboard.

A screenshot of a computer

Description automatically generated

This is a drop-down menu in which we can select which type of store you want to visualize the data.

A screen shot of a graph

Description automatically generated

Various metrics such as Item\_Outlet\_Sales and Outlet\_Establishment\_year.

## 6. Conclusion

### 6.1 Summary of Findings

The project successfully developed a sales forecasting model using Big Mart's historical sales data. The findings revealed significant insights into the factors influencing sales, and the developed model demonstrated reliable predictive performance.

### 6.2 Final Thoughts on Project Outcomes

The project outcomes are promising, indicating the potential of machine learning models in transforming retail sales forecasting. The model provides Big Mart with a tool to anticipate sales trends, thereby aiding in strategic decision-making.

### 6.3 Challenges

Challenges faced during the project included handling missing data, ensuring data quality, and selecting appropriate features for the model. Overcoming these challenges was crucial for the success of the project.

### 6.4 Lessons Learned

Key learnings from the project include the importance of thorough data preprocessing, the effectiveness of exploratory data analysis in uncovering insights, and the advantages of ensemble methods in predictive modeling.

### 6.5 Scope for Improvement/Extension

Future enhancements could include integrating more sophisticated machine learning algorithms, incorporating additional features, and expanding the dataset to include more recent sales data. These improvements could further refine the model's accuracy and applicability.