BIG MART SALES PREDICTION

By-
Mendu Mahesh
Shreyash Mishra
Vipul Shinkar
Nishi Kumari
Purushottam Agarwal

ABSTRACT

- In today's world big malls and marts record sales data of individual items for predicting future demand and inventory management.
- This data Stores a large number of attributes of the item as well as individual customer data together in a data warehouse.
- This data is mined for detecting frequent patterns as well as anomalies.
- This data can be used for forecasting future sales volume with the help of random forests and multiple linear regression model

PROBLEM STATEMENT

BigMart, a prominent retail company, has embarked on a journey to enhance its sales strategy by leveraging data-driven insights. In 2013, the company collected comprehensive sales data from 10 stores across different cities, encompassing 1559 distinct products.

The overarching goal of this analysis is to construct a predictive model capable of forecasting the sales of individual products within each store. By dissecting the intricate relationship between product characteristics, store attributes, and sales figures, BigMart aims to pinpoint the key factors influencing sales.

These insights will not only unravel the underlying patterns governing sales but will also equip BigMart with actionable intelligence to tailor its marketing strategies, optimize product placement, and ultimately amplify sales and revenue.

DATA EXPLORATION

Column Attributes	Summary
Item Identifier	Unique product ID
Item Weight	Weight of product
Item Fat Content	Whether the product is low fat or not
Item Visibility	The % of total display area of all products in a store
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 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
Outlet Type	Whether the outlet is a grocery store or a supermarket
Item Outlet Sales	Sales of the product in the particular store

SOFTWARE REQUIREMENT SPECIFICATION

1. **Numpy** - It has advanced math functions and a rudimentary scientific computing package.
2. **Pandas** - is a must for data-science. It provides fast, expressive, and flexible data structures to easily (and intuitively) work with structured (tabular, multidimensional, potentially heterogeneous) and time-series data.
3. **Matplotlib, Seaborn and Plotly** - These library helps with data analyzing, and is a numerical plotting library.
4. **Scikit-learn** - is an open-source machine learning library for Python that provides simple and efficient tools for data mining and data analysis, with various classification, regression, clustering algorithms.

QUICK LOOK AT THE DATA STRUCTURE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

	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
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3	Grocery Store	732.3800
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052

Most of the items in the dataset present 8523 non-null values. However, there are some cases such as Item Weight and Outlet Size which seem to present null values. We always have to consider if this absence of values has a significant meaning or not.

Moreover, from the 12 features, 5 are numeric and 7 categorical,

train.describe()

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	7060.000000	8523.000000	8523.000000	8523.000000	8523.000000
mean	12.857645	0.066132	140.992782	1997.831867	2181.288914
std	4.643456	0.051598	62.275067	8.371760	1706.499616
min	4.555000	0.000000	31.290000	1985.000000	33.290000
25%	8.773750	0.026989	93.826500	1987.000000	834.247400
50%	12.600000	0.053931	143.012800	1999.000000	1794.331000
75%	16.850000	0.094585	185.643700	2004.000000	3101.296400
max	21.350000	0.328391	266.888400	2009.000000	13086.964800

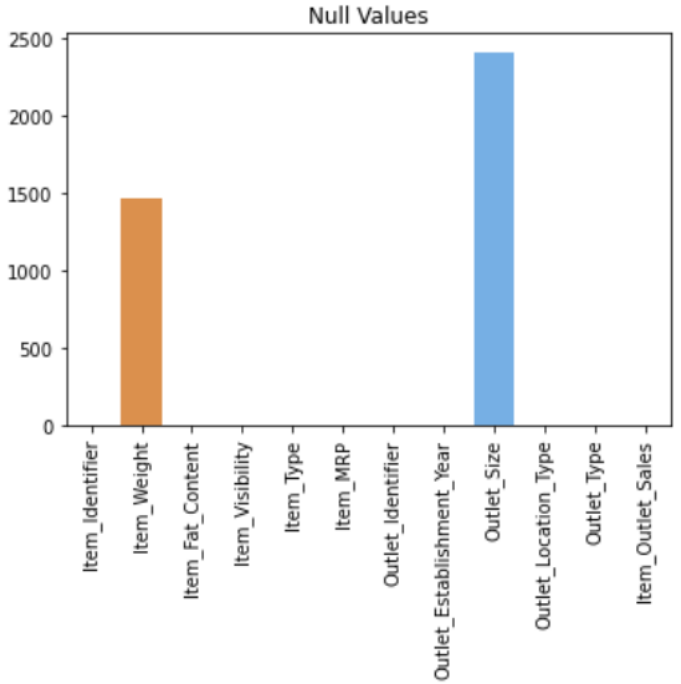
DATA CLEANING

This step typically involves imputing missing values, treat irregularities and treating outliers.

1. We found two variables with missing values - **Item Weight** and **Outlet Size**. Let's impute the former by the average weight of the particular item.

```
# Filling Null Values in Item Weight
item_weight_mean = df.groupby('Item_Identifier')['Item_Weight'].mean()
df['Item_Weight'] = df['Item_Weight'].fillna(df['Item_Identifier'].map(item_weight_mean))

# Filling Null Values in Outlet Size
outlet_size_mode = df.groupby('Outlet_Type')['Outlet_Size'].apply(lambda x: x.mode()[0])
df['Outlet_Size'] = df['Outlet_Size'].fillna(df['Outlet_Type'].map(outlet_size_mode))
```



2. **Item Fat Content** contains some mis code values

```
df.replace({'Item_Fat_Content': {'low fat': 'Low Fat','LF': 'Low Fat','reg': 'Regular'}}, inplace=True)
```

3. In **Item Visibility,** We noticed that the minimum value here is 0, which makes no practical sense. Lets consider it like missing information and impute it with group mean visibility of that product.

```
df['Item_Visibility'].replace(0,np.nan,inplace=True)
Item_Visibility_mean = df.groupby('Item_Identifier')['Item_Visibility'].mean()
df['Item_Visibility'] = df['Item_Visibility'].fillna(df['Item_Identifier'].map(Item_Visibility_mean))
```

4. Using the unique Item Identifier, which starts with "FD," "DR," or "NC," we can categorize items into Food, Drinks, and Non-Consumables, respectively, and create a new column for these categories.

```
df['New_Item_Type'] = df['Item_Identifier'].apply(lambda x: x[:2])
df['New_Item_Type'] = df['New_Item_Type'].replace({'FD':'Food', 'NC':'Non-Consumable', 'DR':'Drinks'})
```

5. Three categories of (Food, Non-Consumables and Drinks).We will use this ‘Non Consumable' category to represent the ‘Fat Content' which will be 'Non-Edible’.

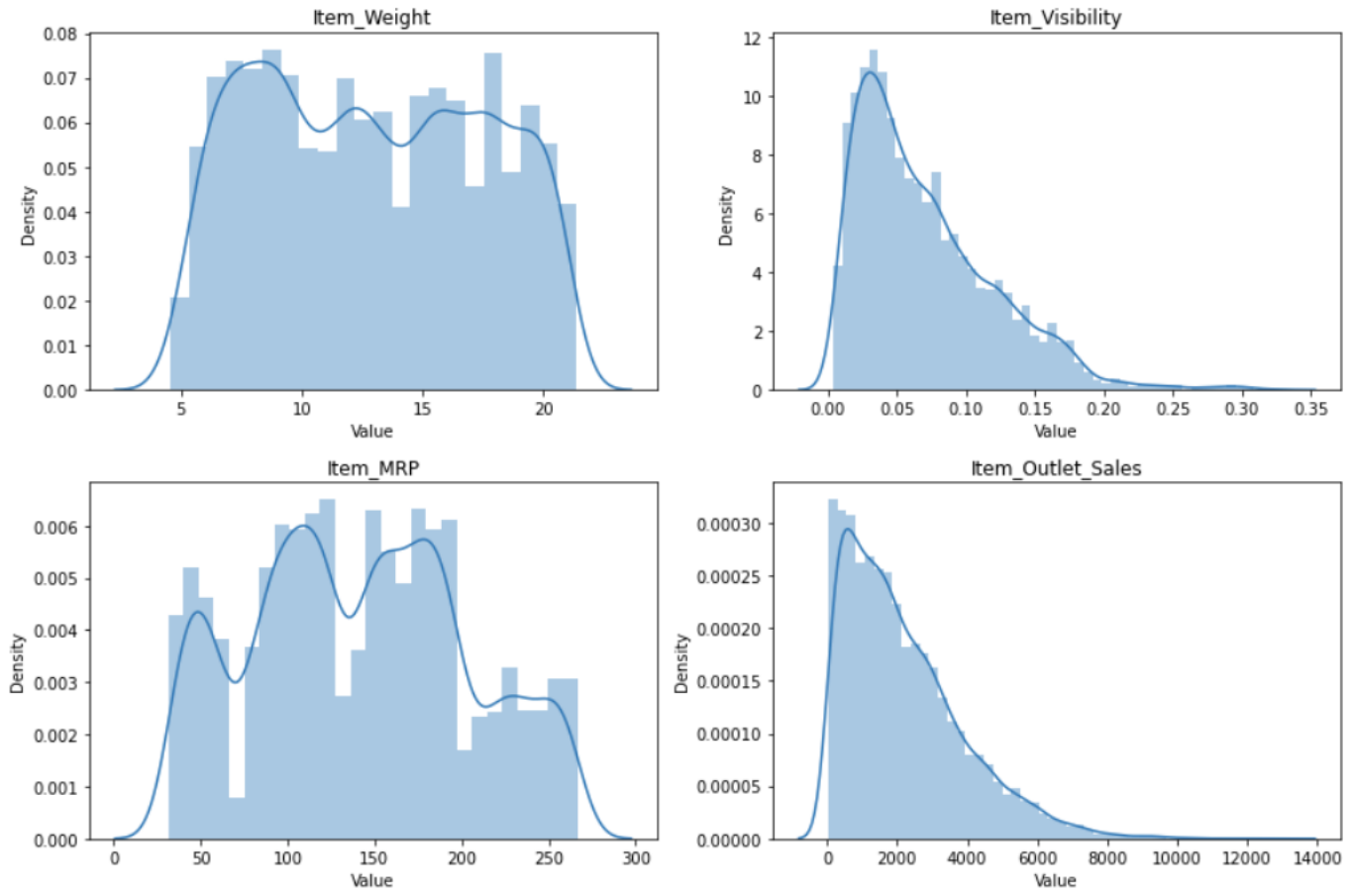
```
# Using this 'Non_Consumable' category to represent the 'Fat_Content' which will be 'Non-Edible'.
df.loc[df['New_Item_Type']=='Non-Consumable', 'Item_Fat_Content'] = 'Non-Edible'
df['Item_Fat_Content'].value_counts()
```

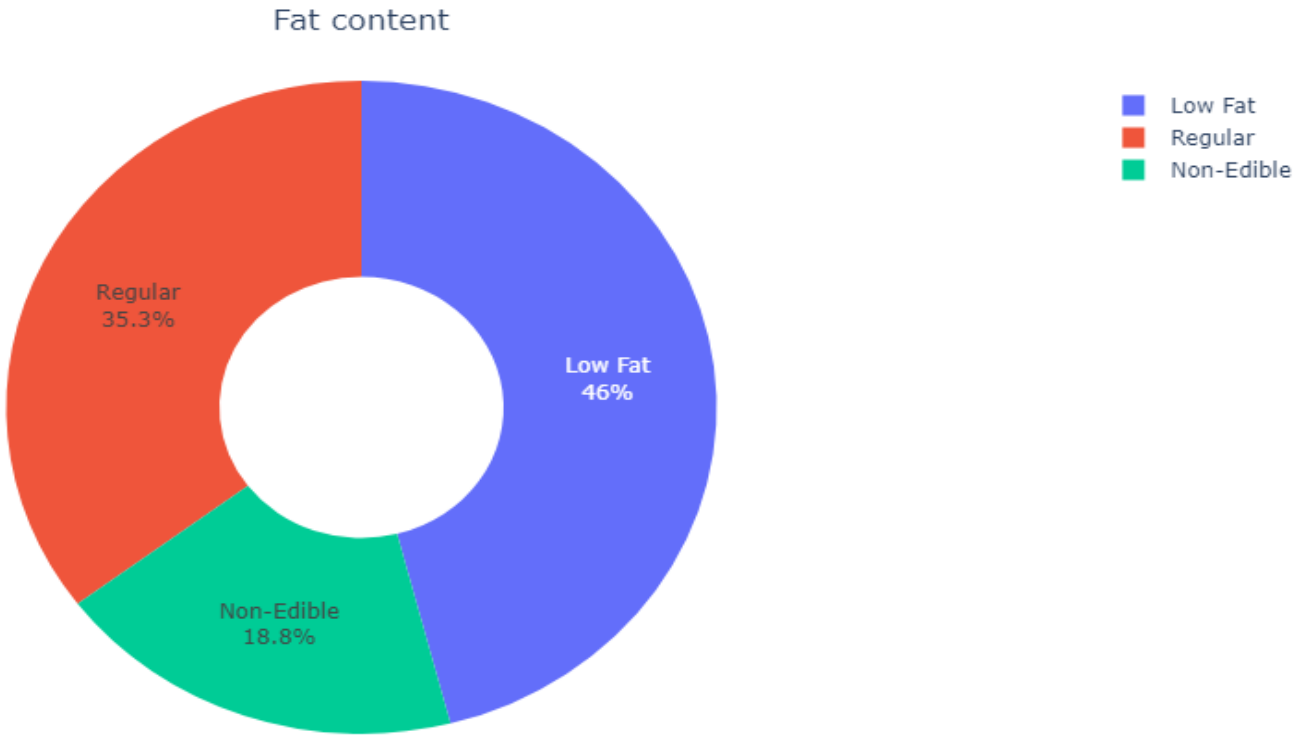
6. As we can see establishment year, let’s calculate the age of the stores. By subtracting the establishment year from the current year (2023), we obtain the number of years that have passed since the outlet was established, giving us an indication of its age. This information might be used to understand patterns related to the outlet's performance

```
# Creating small values for establishment year
df['Outlet_Years'] = 2023 - df['Outlet_Establishment_Year']
```

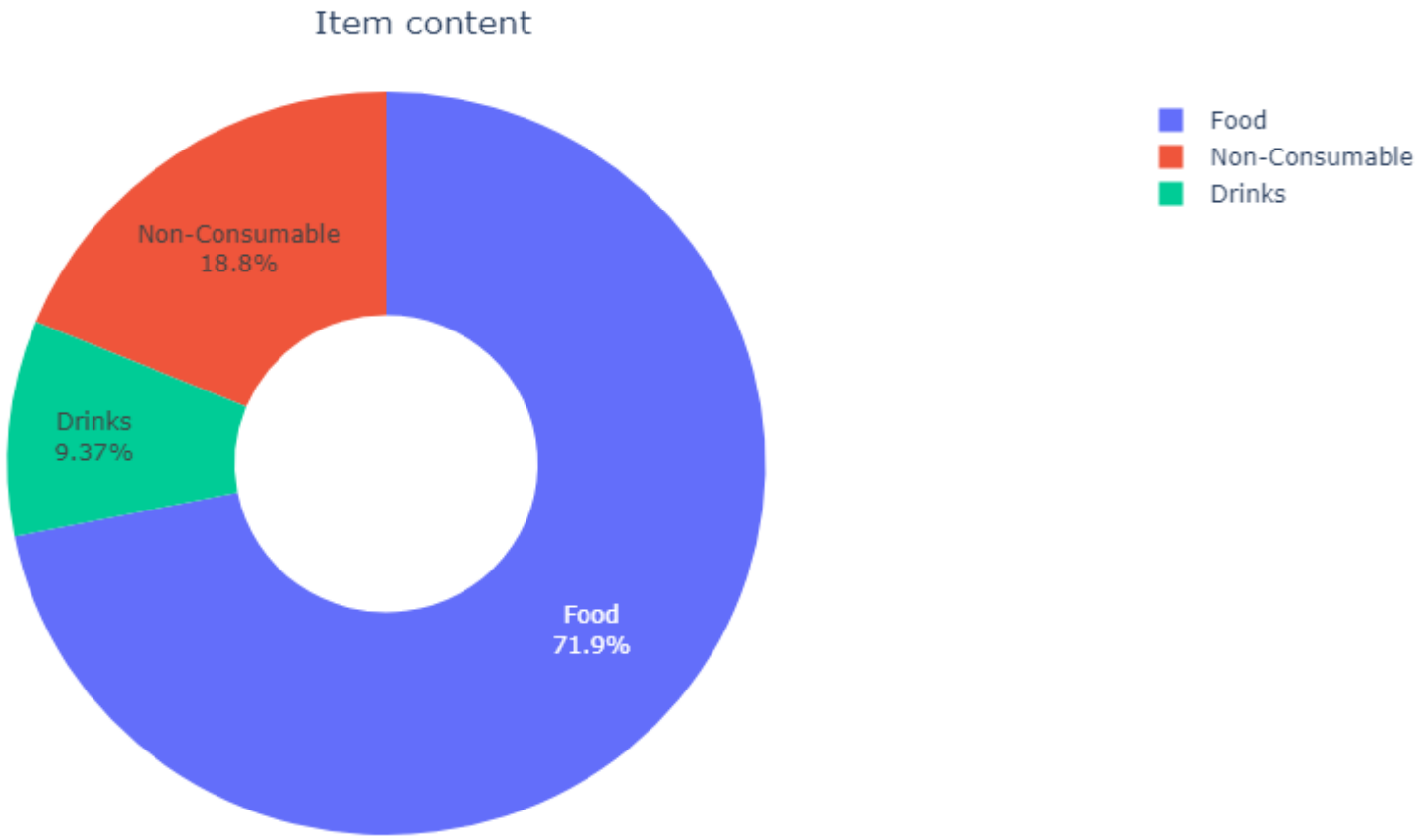
DATA VISUALIZATION

This step typically involves graphical representation of information and data, using visual elements like charts, graphs, and maps, to provide an accessible way to see and understand trends, outliers, and patterns in data.

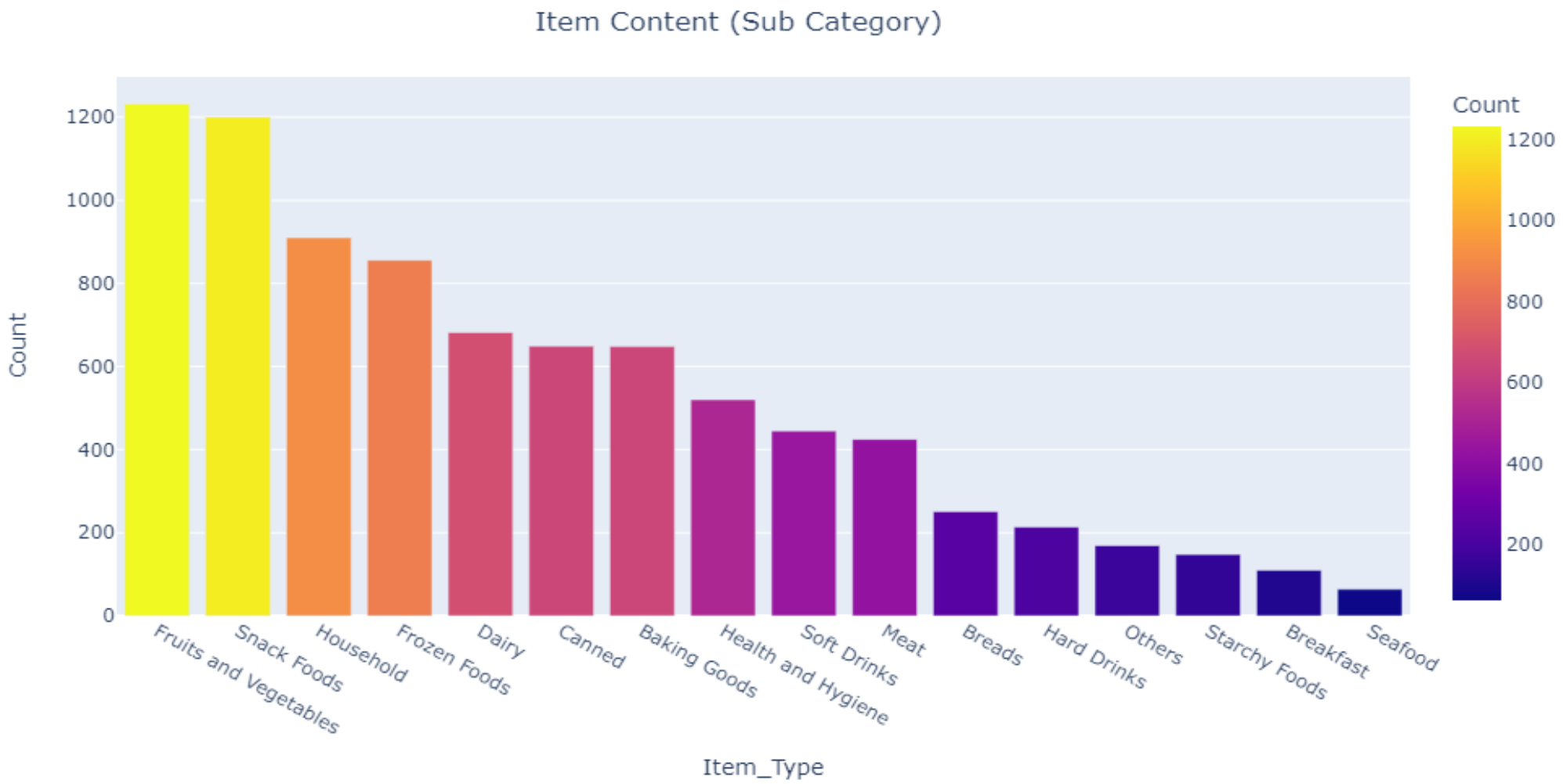




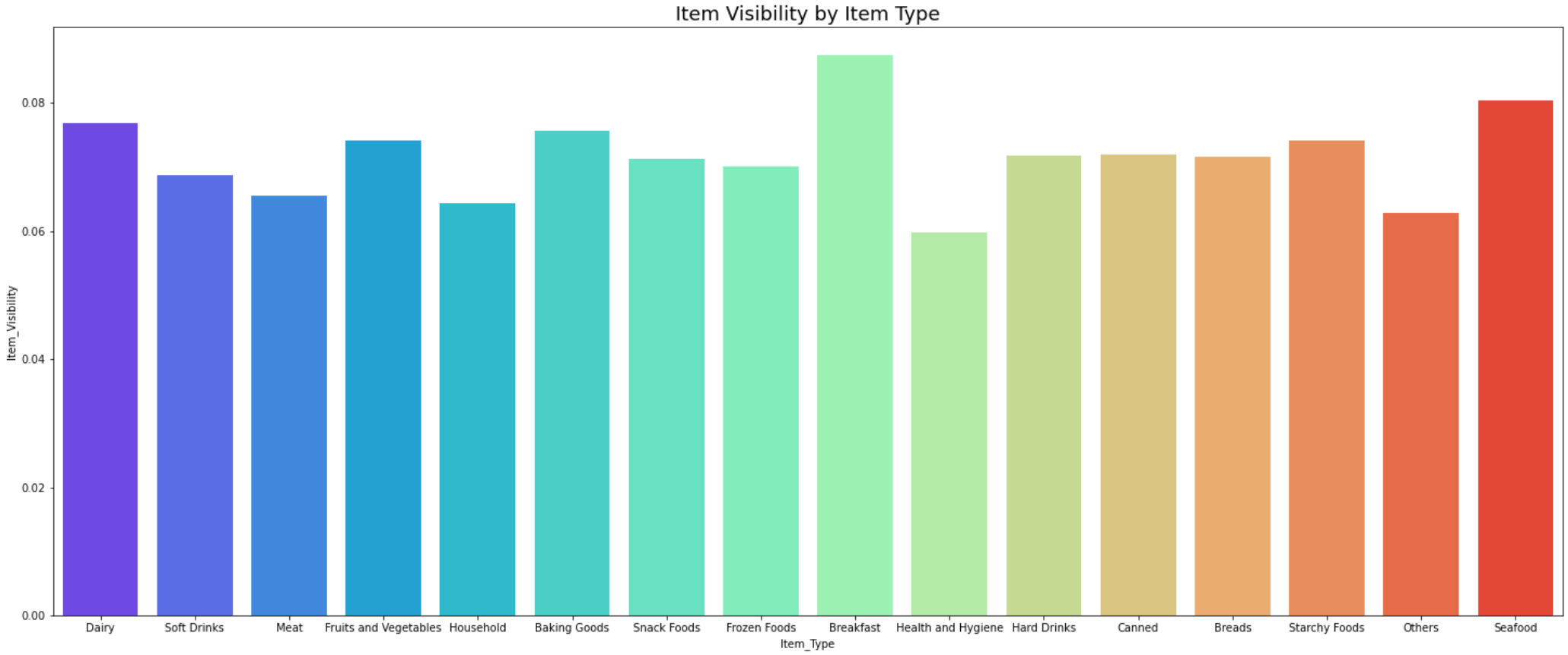
We see that about 46 % items are of low fat. This assumes that nearly half of customers are health conscious and prefer food with lower fat rating.



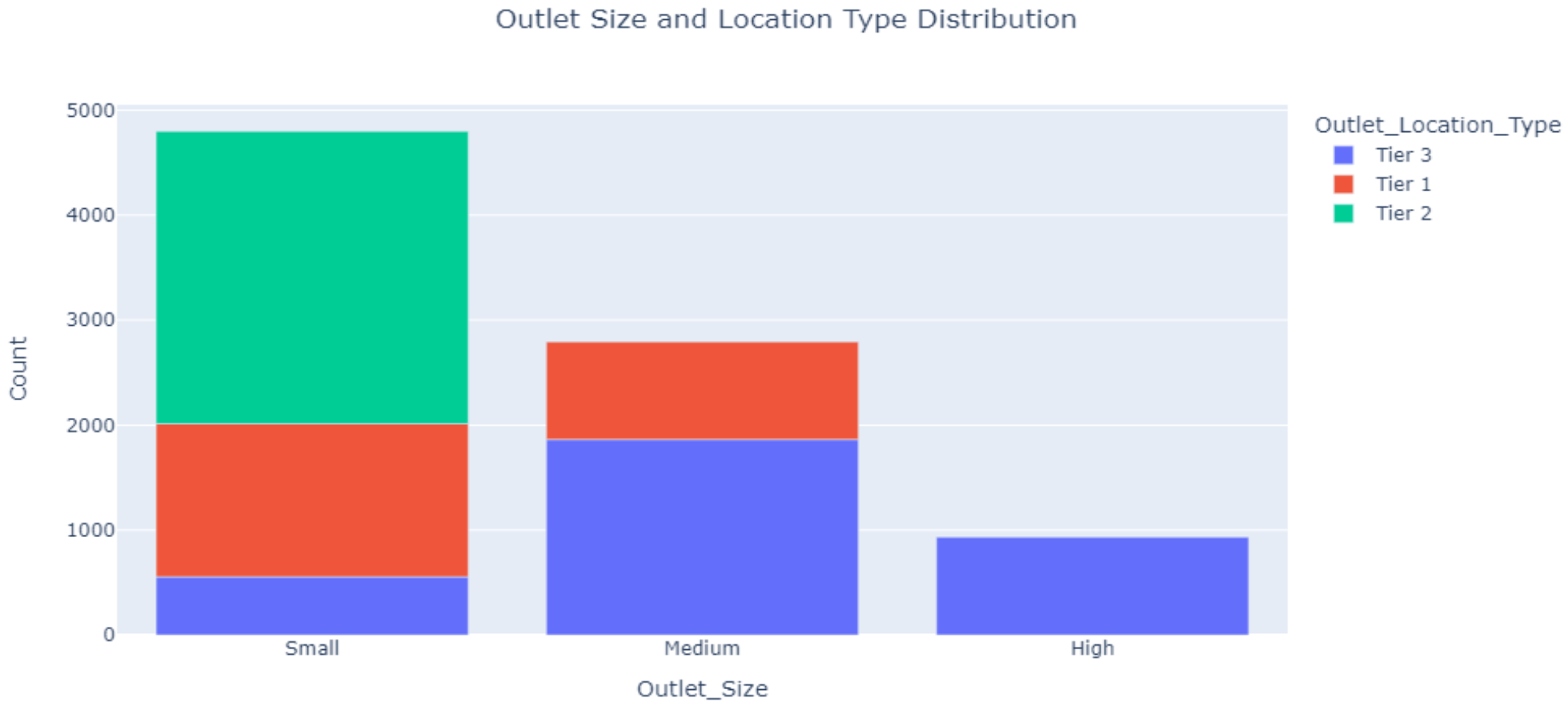
Around 72% items in the store are under food category



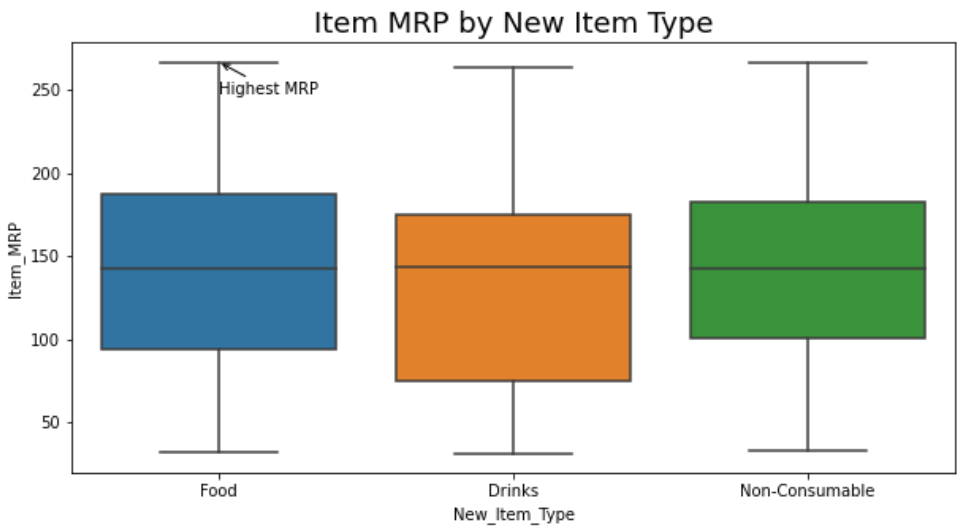
From the above plot, we see that fruits and vegetables were the highest sold item followed closely by Snack foods



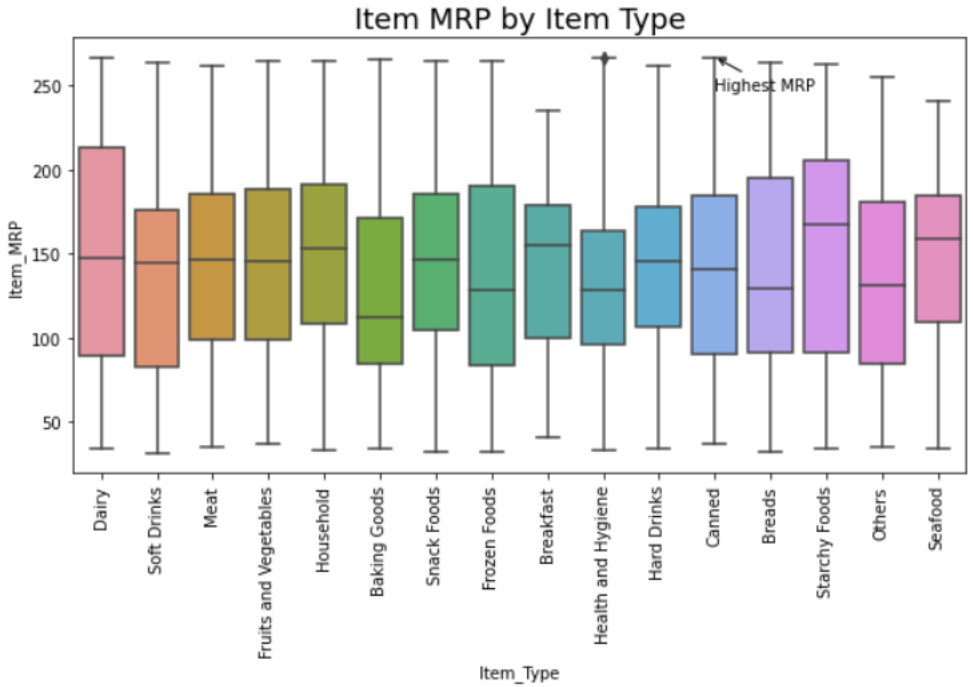
From the above plot, we see that **Breakfast** item type has the most item visibility.



The above count plot shows that the "Small" outlet size is present in all the locations and has the highest store count in comparison to "Medium" and "High."

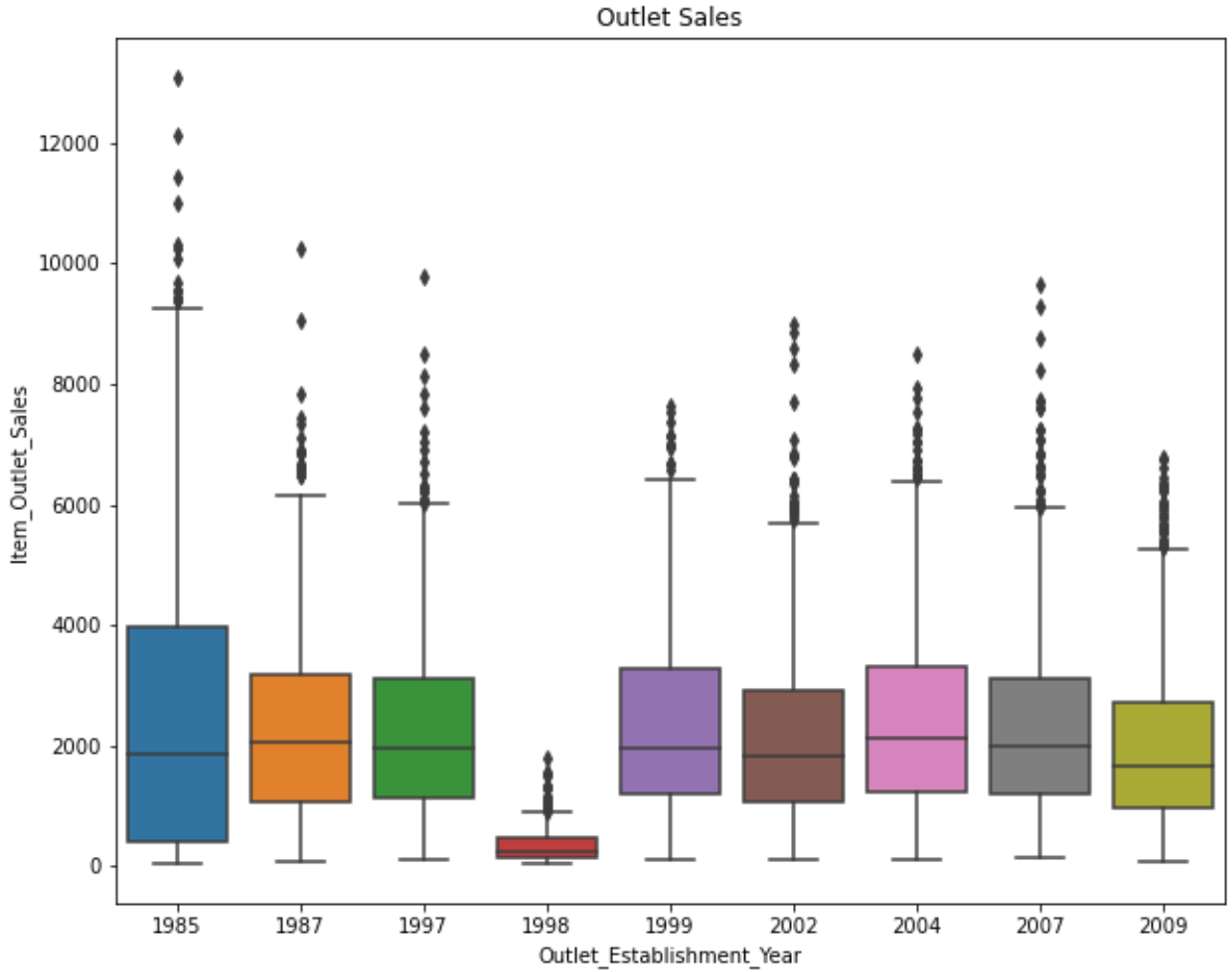


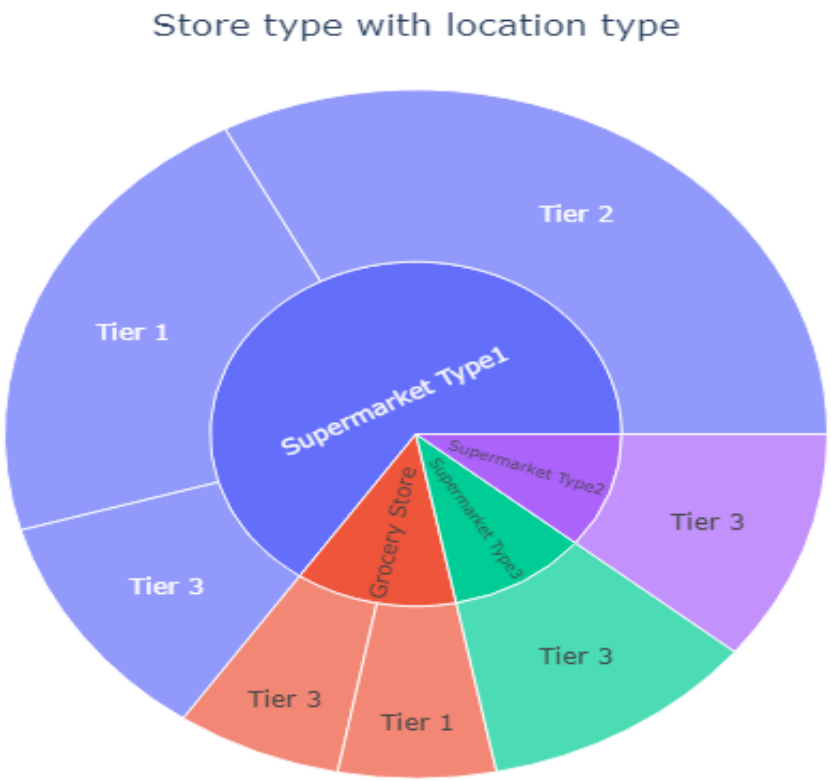
In this boxplot comparison shows that Food category has the highest MRP



From the above boxplot, shows that Canned Item Type has the highest MRP

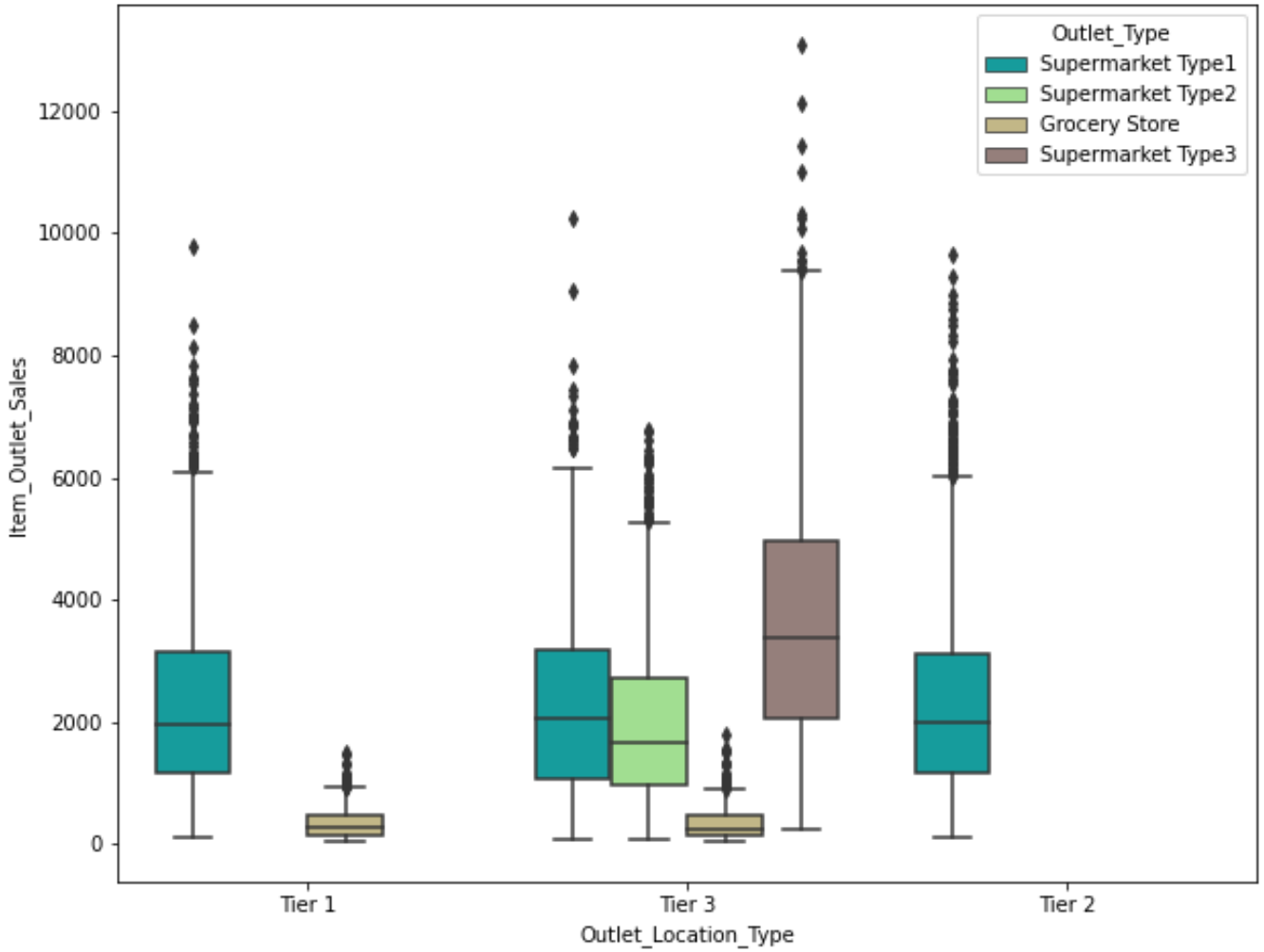
As it can be seen, the sales reported by the older stores is higher than the relatively newer stores (except for the 1998 established store).





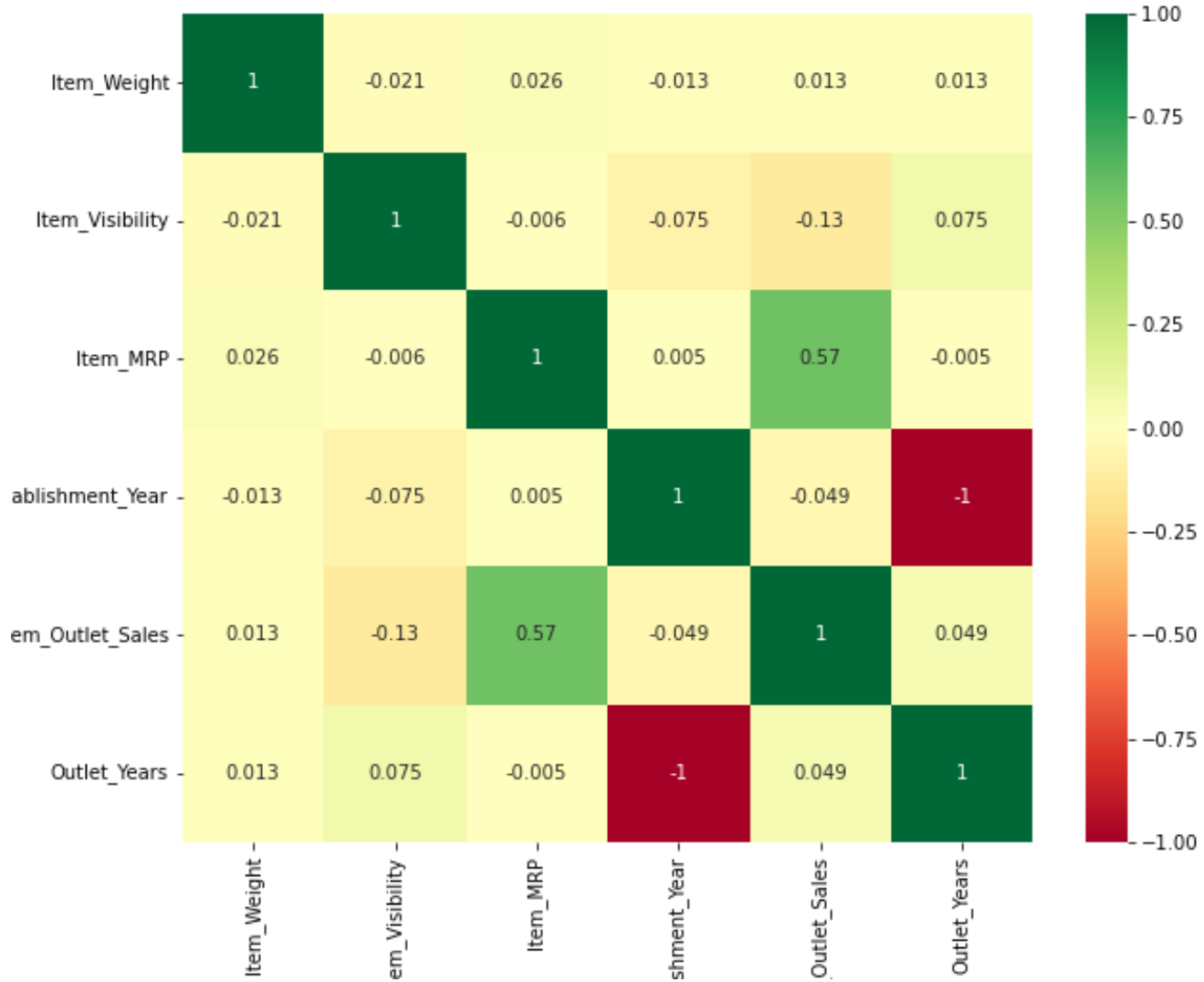
As we can see, majority of the stores are of type 1 supermarket distributed over various location tiers.

Supermarket type 2 and 3 are confined to only tier 3 locations. Very small section of the stores are actually grocery stores.



As we can see, tier 3 locations seem to be selling better than both tier 2 and tier 1. It is also to be noted that tier 3 has more number of stores in it. Hence, the sales are better too.

'Item MRP' has a high positive correlation with **'Item Outlet Sales'**, this indicates that items with a higher maximum retail price tend to have higher sales.



CHANGING CATEGORICAL TO NUMERICAL

Converting categorical data into numerical format is a common preprocessing step in machine learning. Two popular techniques for doing this are label encoding and one-hot encoding

1. **Label Encoding:**
   Label encoding assigns a unique integer to each category of a given feature. It is particularly useful for ordinal data where the order matters.

```
from sklearn.preprocessing import LabelEncoder
label = LabelEncoder()
lst = ['Outlet_Size', 'Outlet_Location_Type']
for i in lst:
    df[i] = label.fit_transform(df[i])
```

2. **One-Hot Encoding:**
   One-hot encoding creates binary columns for each category and returns a matrix with binary values representing the presence of each category. It is suitable for nominal data where no ordinal relationship exists.

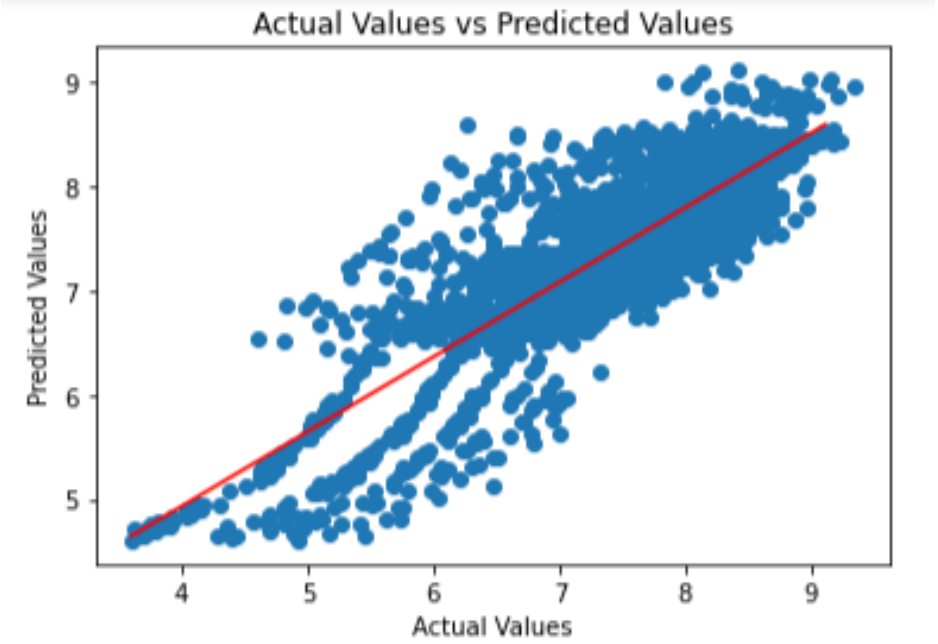
```
df = pd.get_dummies(df, columns=['Item_Fat_Content', 'Outlet_Size','Outlet_Type','Item_Type'])
df.head()
```

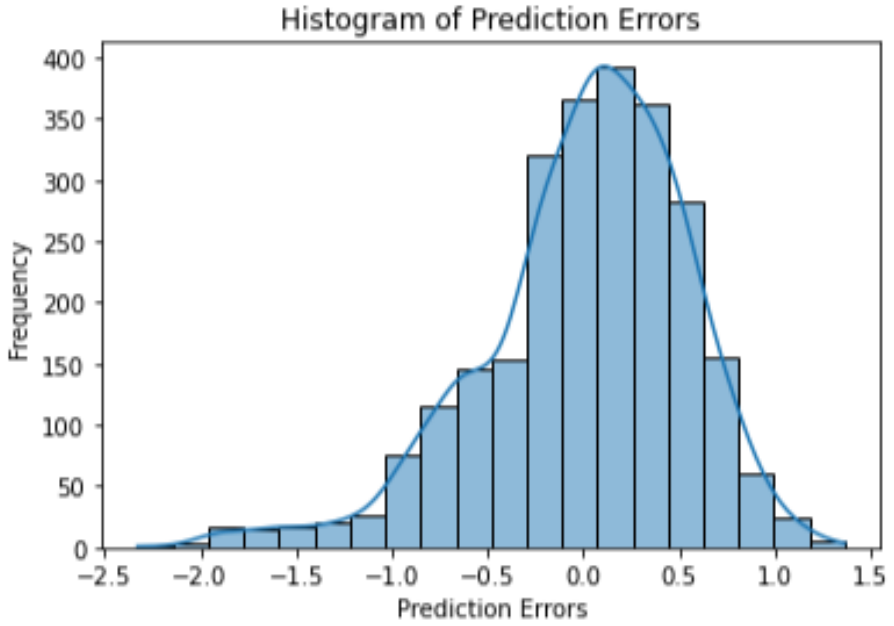
LINEAR REGRESSION

Linear regression is a supervised machine learning algorithm that models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data.

We applied Linear Regression Model in our dataset:

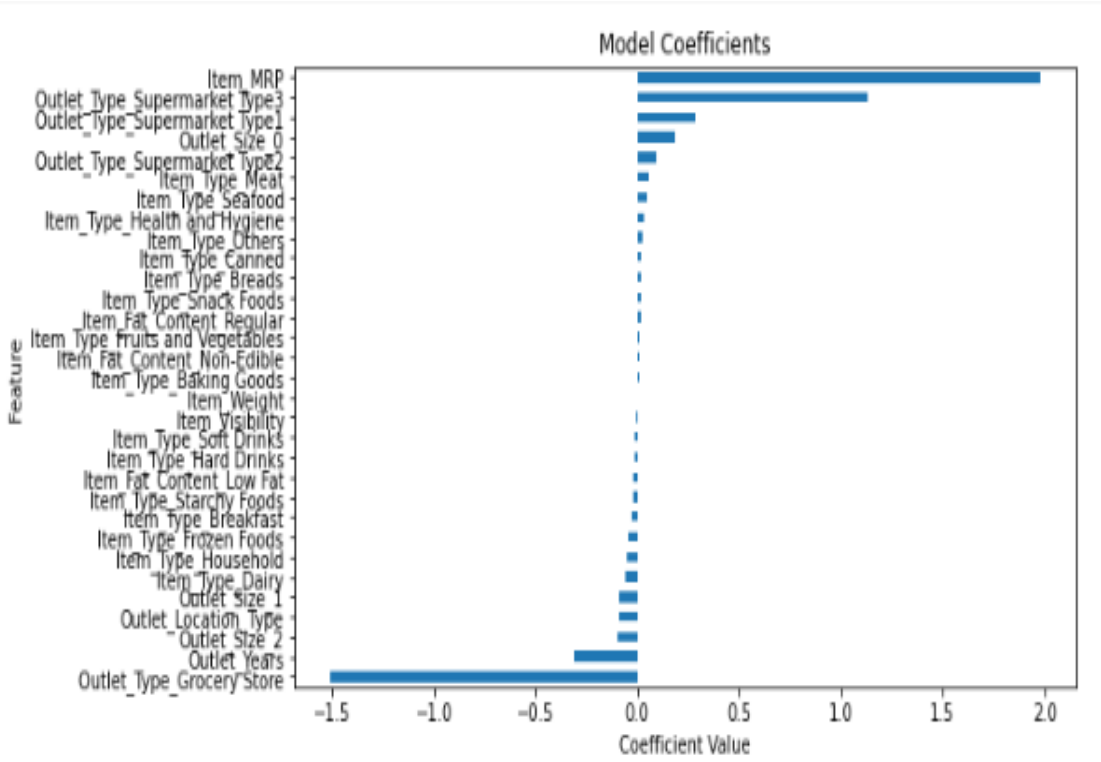
	R2 Score	Adjusted R2 Score
Metrics	0.711607	0.708067





The positive values are attributes with positive coefficients and negative values are attributes with negative coefficients.

There are minor values between positive and negative coefficients. This indicates that the center attributes do not provide significant information.

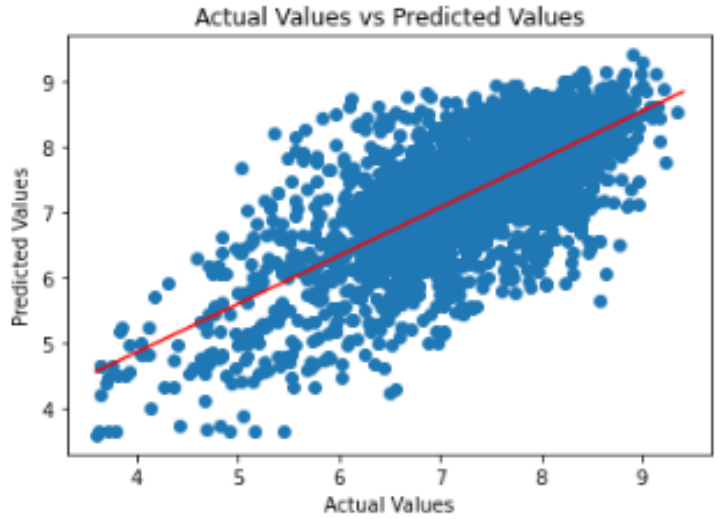


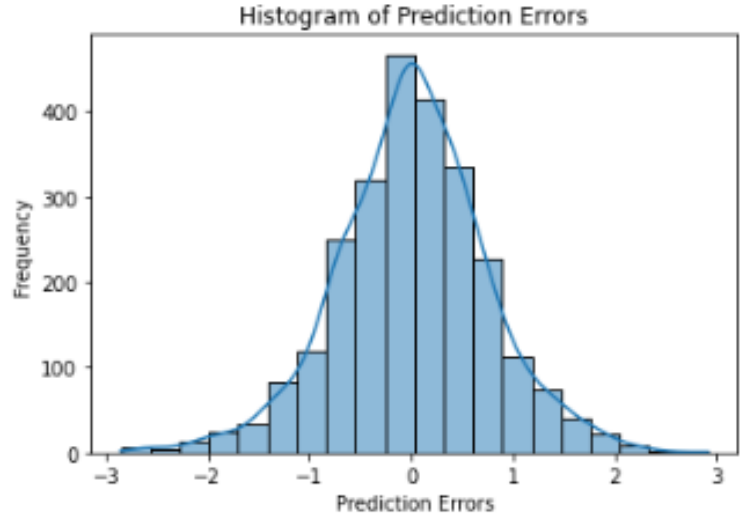
DECISION TREE

A decision tree is a supervised machine learning algorithm that uses a tree-like graph of decisions and their possible consequences to make predictions or classify data.

We applied Decision Tree Regressor in our dataset:

	R2 Score	Adjusted R2 Score
Metrics	0.451173	0.444435



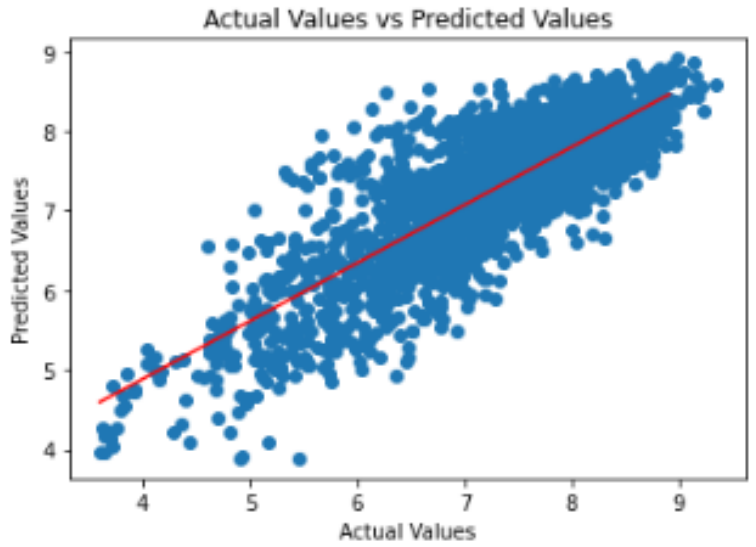


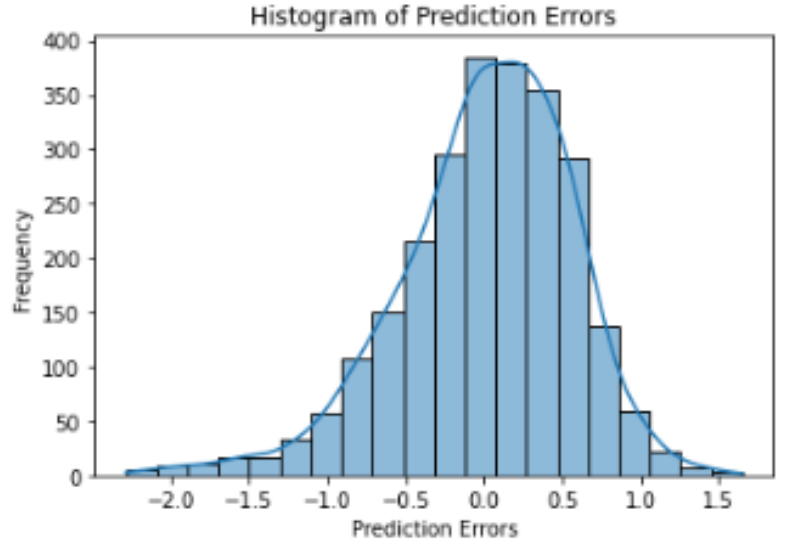
RANDOM FOREST

Random Forest is an ensemble learning method that combines multiple decision trees to create a more accurate and robust prediction model.

We applied Random Forest Regressor in our dataset:

	R2 Score	Adjusted R2 Score
Metrics	0.696008	0.692276



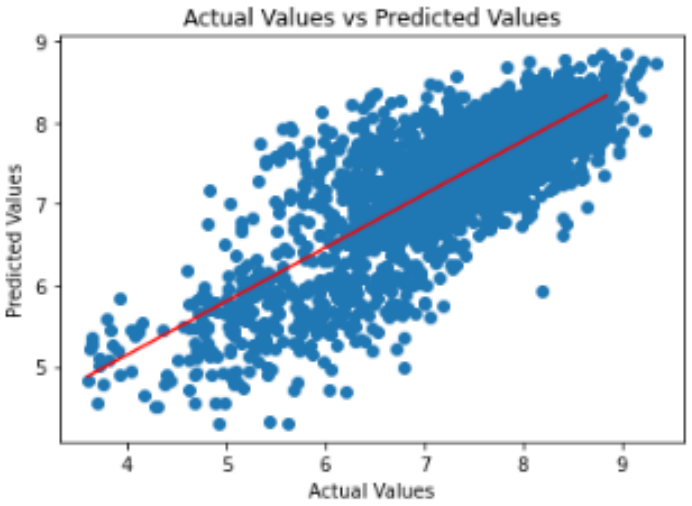


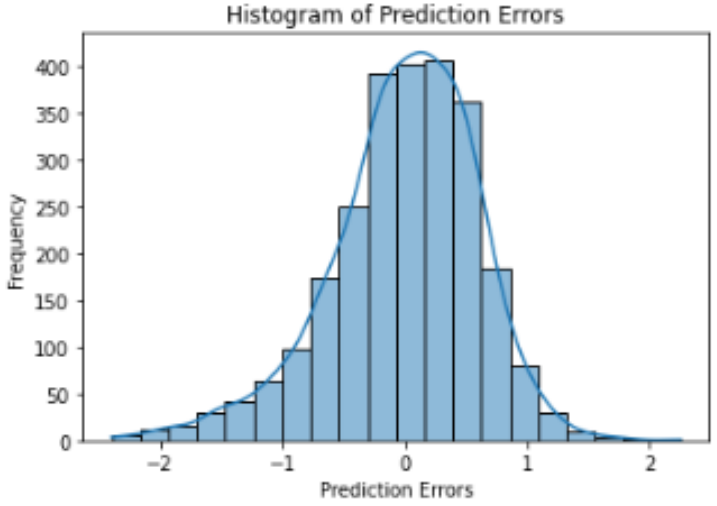
K-NEAREST NEIGHBORS (K-NN)

The k-Nearest Neighbors (k-NN) algorithm is a type of instance-based learning that classifies a data point based on how its neighbors are classified, with "k" representing the number of neighbor data points considered.

We applied KNeighborsRegressor in our dataset:

	R2 Score	Adjusted R2 Score
Metrics	0.635054	0.630574



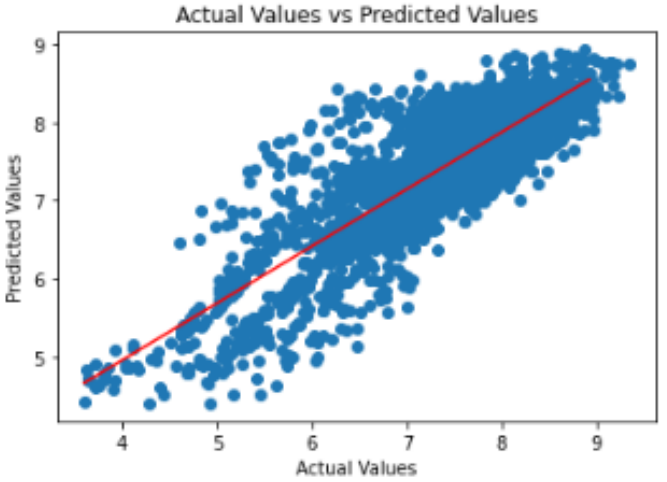


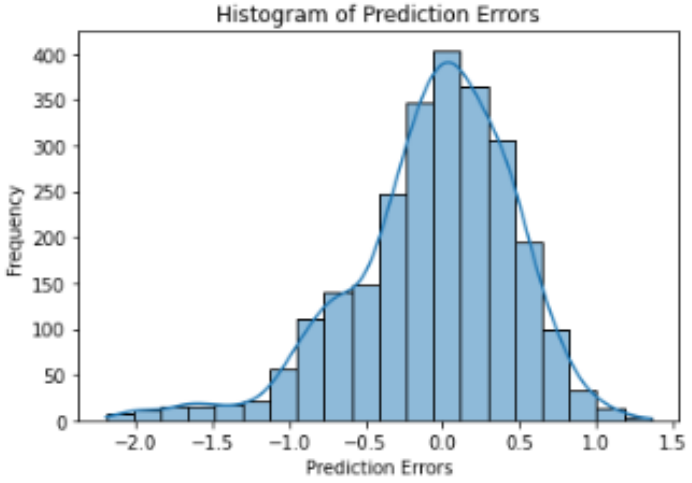
SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm that finds the optimal hyperplane to separate different classes by maximizing the margin between the closest points of the classes, known as support vectors.

We applied SVM Regressor in our dataset:

	R2 Score	Adjusted R2 Score
Metrics	0.716137	0.712652





FINDINGS & CONCLUSION

- Out of all the machine learning algorithms, SVM performed well, followed by linear regression.
- This research proposed a framework that predicts mart's sales using a machine learning model and different techniques.
- Further, in the future, a retailer checks the score of a specific product by entering product attributes and its store's information, like location, and culture.
- Also, we consider an online App for the customer's review regarding the stores and specific products for future work. This App works as a ranking App. Customers rank the stores by giving feedback; this helps the other customers to move on towards the stores. Sort of forum allows checking the demand for a specific store's specific product.

ML Algorithms	R2 Square	Adjusted R2 Square
Linear Regression	71.12%	**70.80%**
Decision Tree	45.12%	44.44%
Random Forest	69.60%	69.22%
K-NN	63.50%	63.05%
SVM	71.61%	**71.26%**


THANK
YOU!