Project Report

Group Members:

1. Shubham Chaudhari

2. Amol Shinde

Aim:

The aim is to build a predictive model and find out the sales of each product at a particular store.

Dataset Source:

BigMart Sales Prediction

https://drive.google.com/drive/folders/1DbAB_8M1tNLVI0XQzX29Pet-IkjeyATH

Dataset Information:

Number of Instances: 14204
Number of Attributes: 13
Missing Values: Yes

Introduction:

This dataset consists Big Mart sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store. Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales.

Column Description:

Feature Name	Description
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.
Item_Type	The category to which the product belongs.
Item_MRP	Maximum Retail Price (list price) of the product.
Outlet_Identifier	Unique store ID.
Item_Establishment_Year	The year in which store was established.
Outlet_Size	The size of the store in terms of ground area.
Outlet_Location_Type	The type of city in which the store is located.
Outlet_Type	Whether the outlet is just a grocery store or supermarket.
Item_Outlet_Sales	Sales of the product in the particular store.

Knowing the Dataset:

1. We started our dataset with finding the number of columns and number of rows in train and test datasets.

```
print(train.shape)
(8523, 12)
```

```
print(test.shape)
(5681, 11)
```

2. Now we structured the dataset and find the type of the variables.

```
Item_Identifier               object
Item_Weight                  float64
Item_Fat_Content              object
Item_Visibility              float64
Item_Type                     object
Item_MRP                     float64
Outlet_Identifier             object
Outlet_Establishment_Year      int64
Outlet_Size                   object
Outlet_Location_Type          object
Outlet_Type                   object
Item_Outlet_Sales            float64
dtype: object
```

3. We also concluded the X-Variables and Y-Variable from the dataset.

Pre-processing of Data:

1. Dealing with missing values:

a) First, we have to see how many missing values are (which were left blank for most variables in the data) *(For Train dataset)*

```
Item_Identifier                 0
Item_Weight                  1463
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                  2410
Outlet_Location_Type            0
Outlet_Type                     0
Item_Outlet_Sales               0
dtype: int64
```

2. Exploratory Data Analysis:

a) Fixing of missing values:

1. Item_Weight:

Item_Weight has missing values of about 17.16% records. Hence, we need to fix these values by taking mean of the products.

```
Item_Weight                  1463
```

2. Outlet_Size:

```
Outlet_Size                    2410
```

Outlet_Size has missing values of about 28.27% records. Hence, we need to fix these values by taking mode of the products.

b) Checking of unique values in the dataset

```
Frequency of Categories for varible Item_Fat_Content
Low Fat    5089
Regular    2889
LF          316
reg         117
low fat     112
Name: Item_Fat_Content, dtype: int64


Frequency of Categories for varible Item_Type
Fruits and Vegetables    1232
Snack Foods              1200
Household                 910
Frozen Foods              856
Dairy                     682
Canned                    649
Baking Goods              648
Health and Hygiene        520
Soft Drinks               445
Meat                      425
Breads                    251
Hard Drinks               214
Others                    169
Starchy Foods             148
Breakfast                 110
Seafood                    64
Name: Item_Type, dtype: int64

Frequency of Categories for varible Outlet_Size
Medium    2793
Small     2388
High       932
Name: Outlet_Size, dtype: int64

Frequency of Categories for varible Outlet_Location_Type
Tier 3    3350
Tier 2    2785
Tier 1    2388
Name: Outlet_Location_Type, dtype: int64

Frequency of Categories for varible Outlet_Type
Supermarket Type1    5577
Grocery Store        1083
Supermarket Type3     935
Supermarket Type2     928
Name: Outlet_Type, dtype: int64
```

c) Interferences Drawn

1. Item_Fat_Content has mis-matched factor levels.

```
Low Fat    5089
Regular    2889
LF          316
reg         117
low fat     11
```

2. Minimum value of Item_Visibility is 0. Practically, this is not possible. If an item occupies shelf space in a grocery store, it ought to have some visibility. We'll treat all 0's as missing values.

Graphical Representation:

1. Correlation between the features

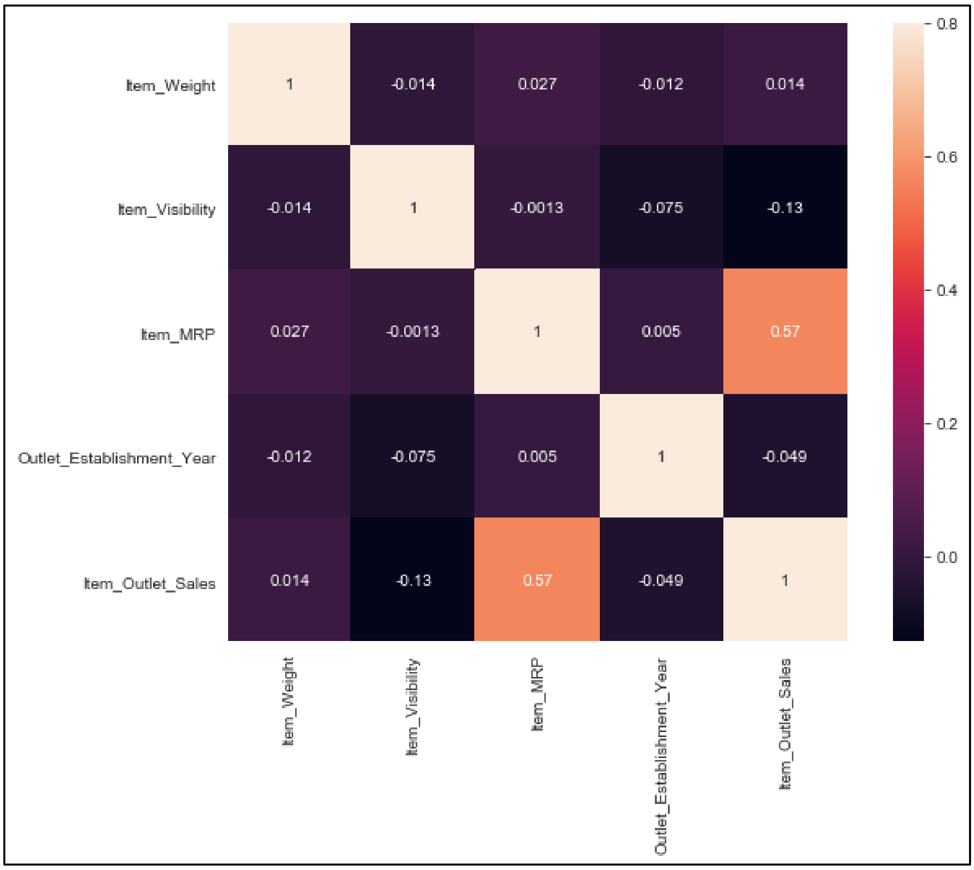


Fig 1: Correlation between features

- There's only one significant correlation is found between the Item_Outlet_Sales and Item_Price
- 0.57 is the correlation value and hence is very useful for our predictions.

2. Scatter Plot Matrix

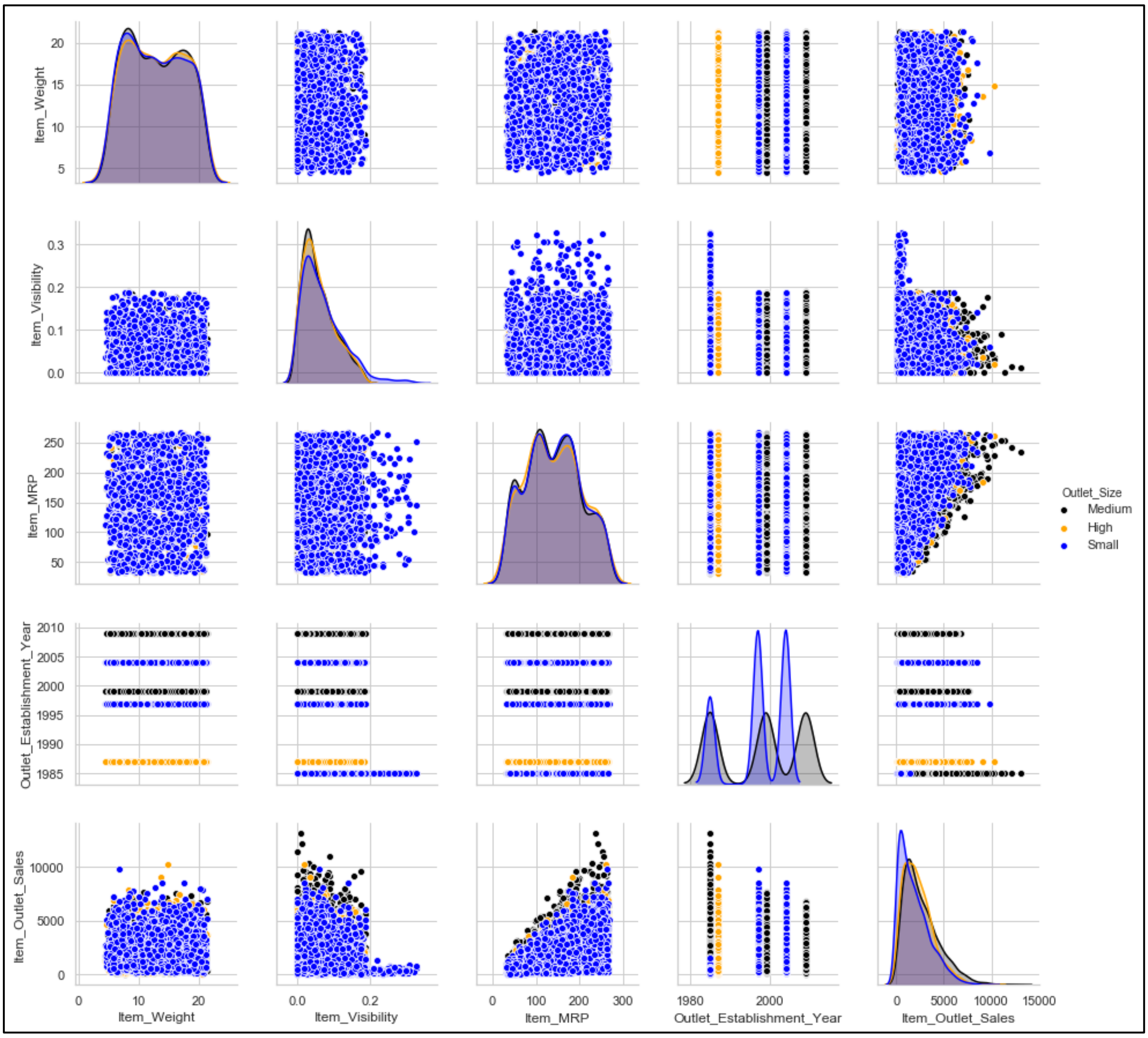


Fig No.2 Scatter Plot Matrix

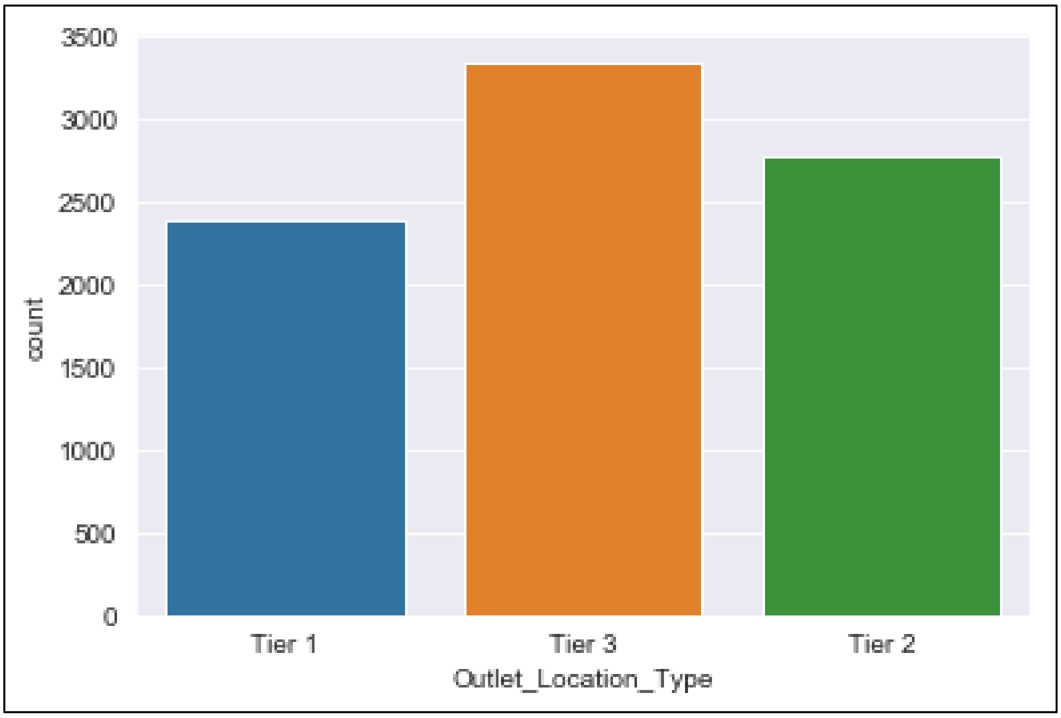
- Item Weight for the Grocery store accounts for less weighted products and also have less sales
- Sales increase with the type of market, the product is sold from
- The visibility of grocery products (Grocery store) is higher as compared to other supermarkets

3. Checking the counts of the outlets store with respect to their location

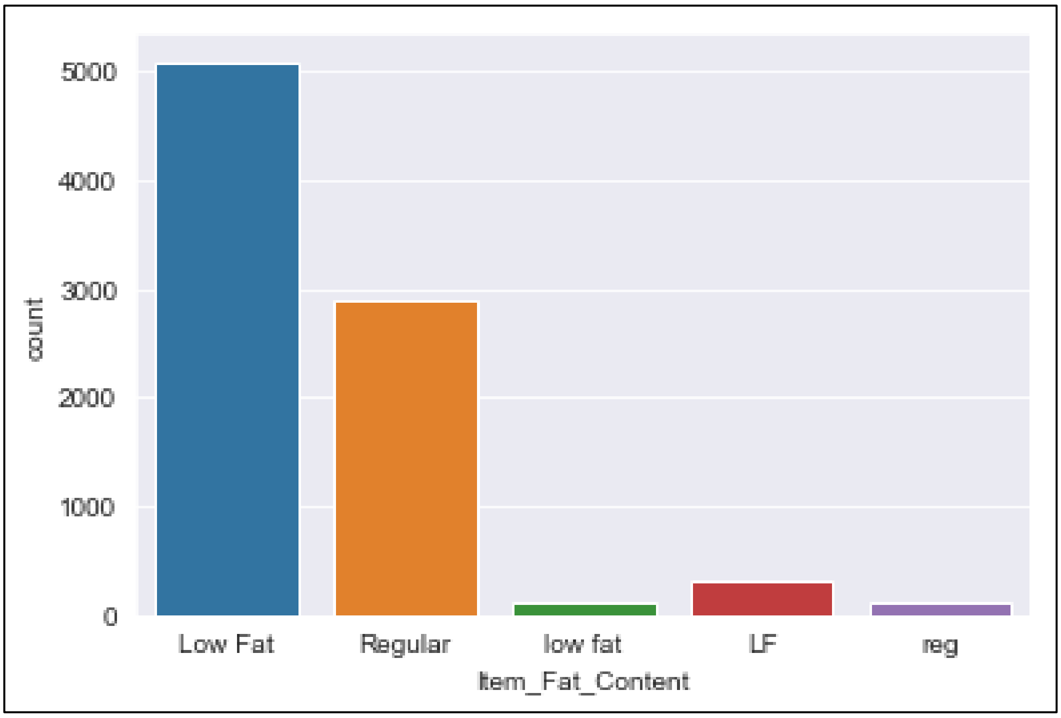
Fig No. 3 Counts of the Outlet Store

- Clearly, Supermarket type 1 dominates the other ones
- Supermarket 1 comprises all the tier2 location and is majorly present at tier1 location

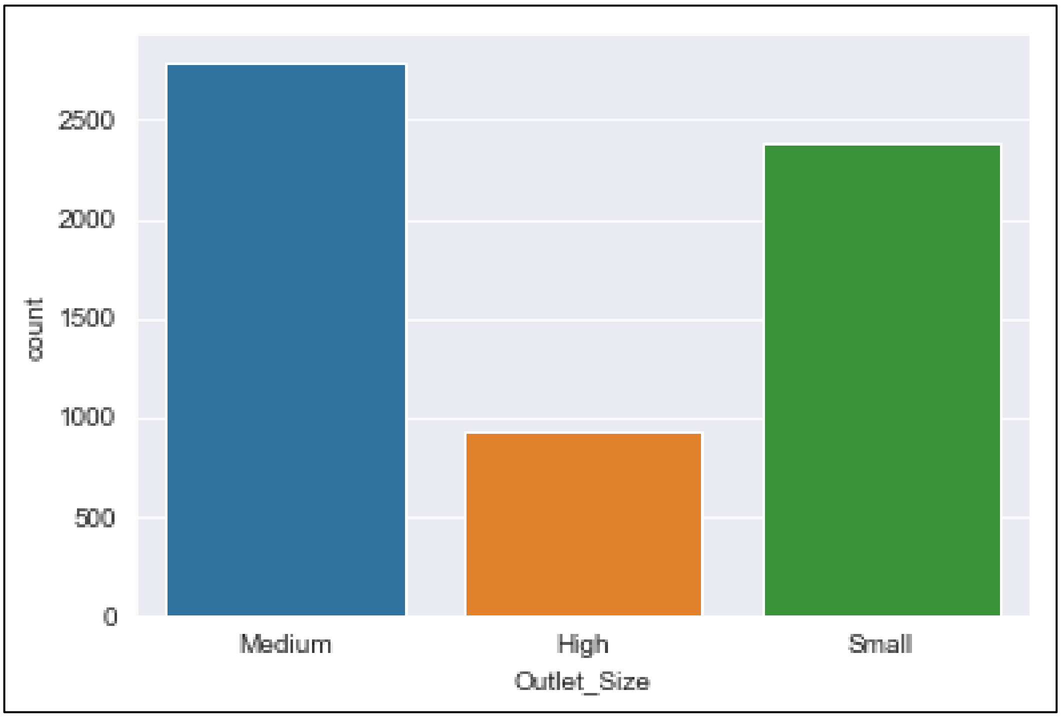
4. Counts of Tier



5. Item Fat Contents



6. Outlet Size



Feature Extraction:

Selecting models

1. Decision Trees: By iteratively and hierarchically observing the level of certainty of predicting whether someone would be readmitted or not, we find the relative importance of different factors using a more human-like decision making strategy in establishing this determination.

2. **Random Forests**: By considering more than one decision tree and then doing a majority voting, random forests helped in being more robust predictive representations than trees as in the previous case. For both Decision Trees and Random Forests, we removed the interaction terms from the feature set since these are already accounted for in tree-based models.

3. Support Vector Machines: Support Vector Machines can help model linearly inseparable data, thus allowing us to explain complex non-linear relationships. However, because of high-dimensional structure and complexity, they are limited by their interpretability to gain insights on how different features are weighted/assigned importance.

4. K-nearest Neighbors: While K-nearest neighbors provide decent predictions, they cannot help in deciding the features that contribute to this decision the most, since features are weighted equally (assuming we normalize them) based on simply their contribution to the proximity/distance function