Big Mart Sales Prediction

Dataset Source:

BigMart Sales Prediction

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)

```
Item_Identifier                 0
Item_Weight                  2439
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                  4016
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 17.17% records. Hence, we need to fix these values by taking mean of the products.

```
Item_Weight                  2439
```

2. Outlet_Size:

```
Outlet_Size                  4016
```

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

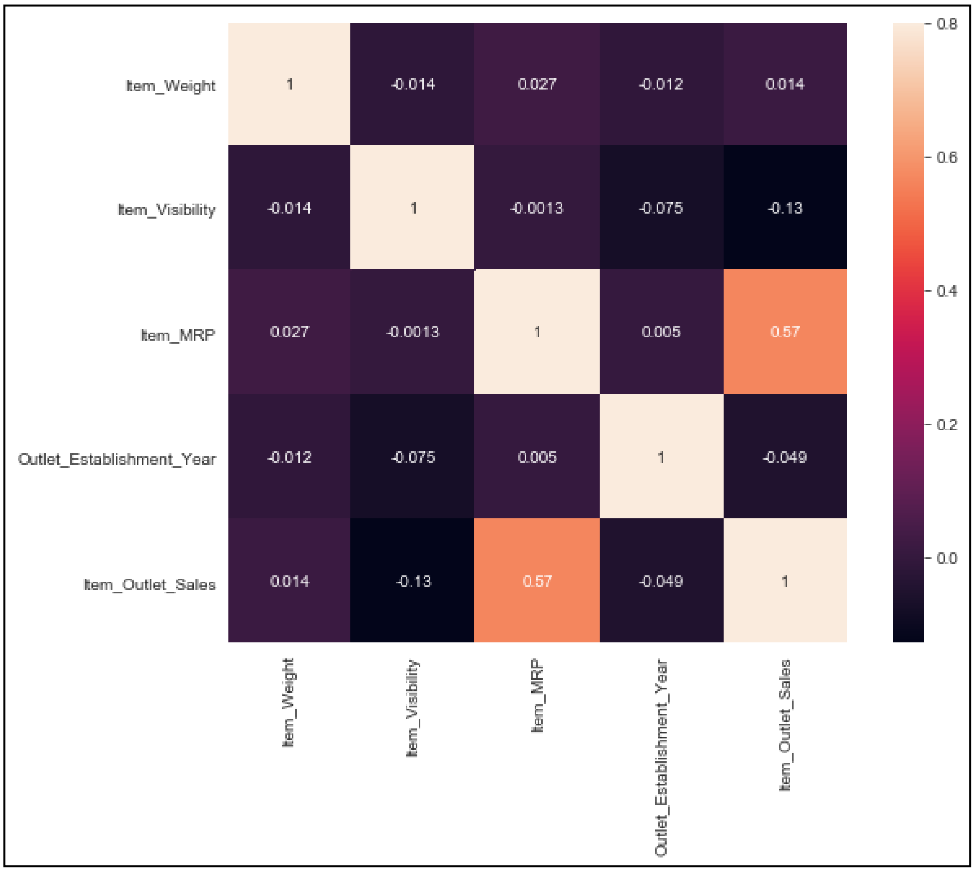

Fig 1: Correlation between features

- 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

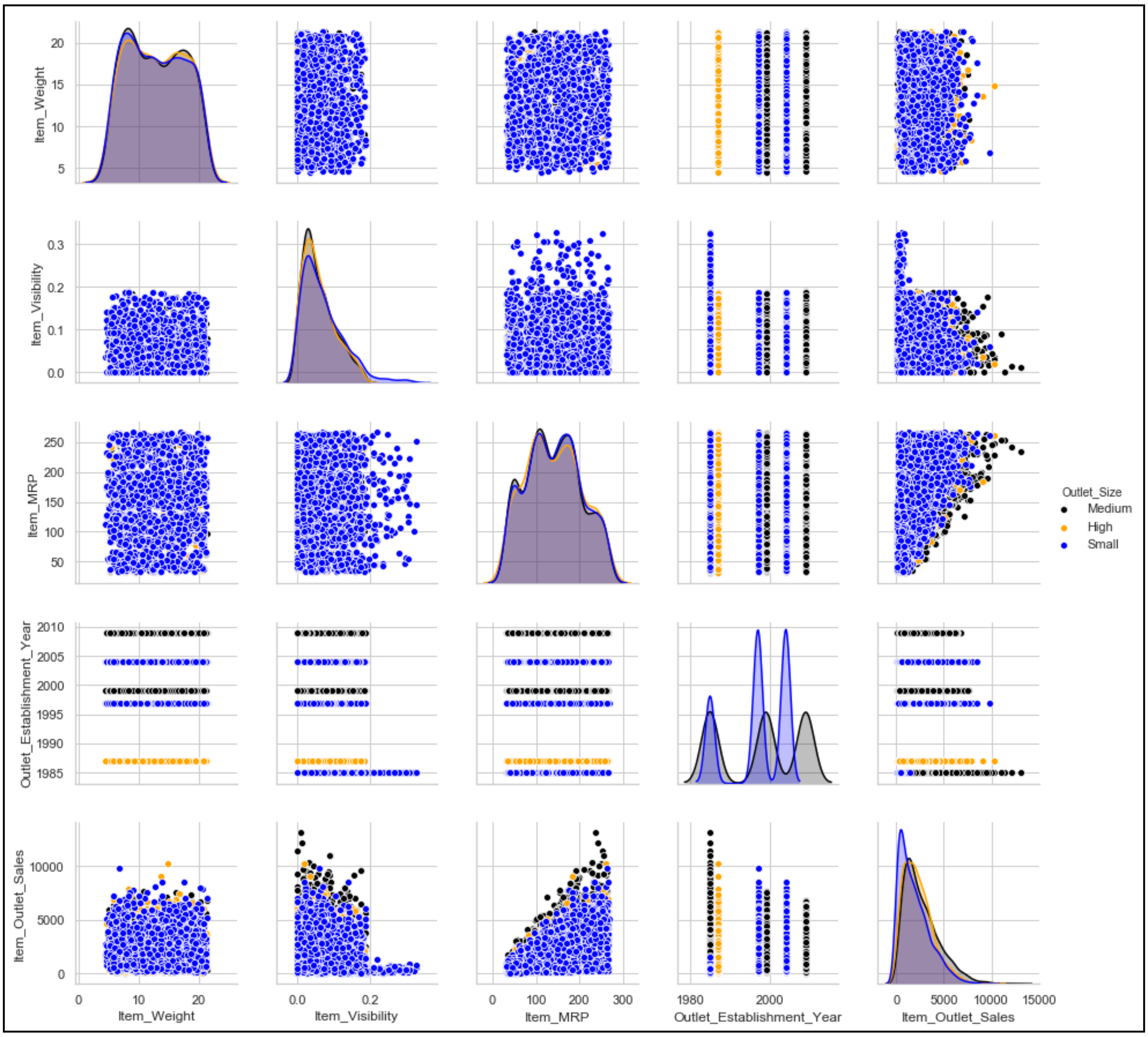

Fig No.2: Scatter Plot Matrix

- 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

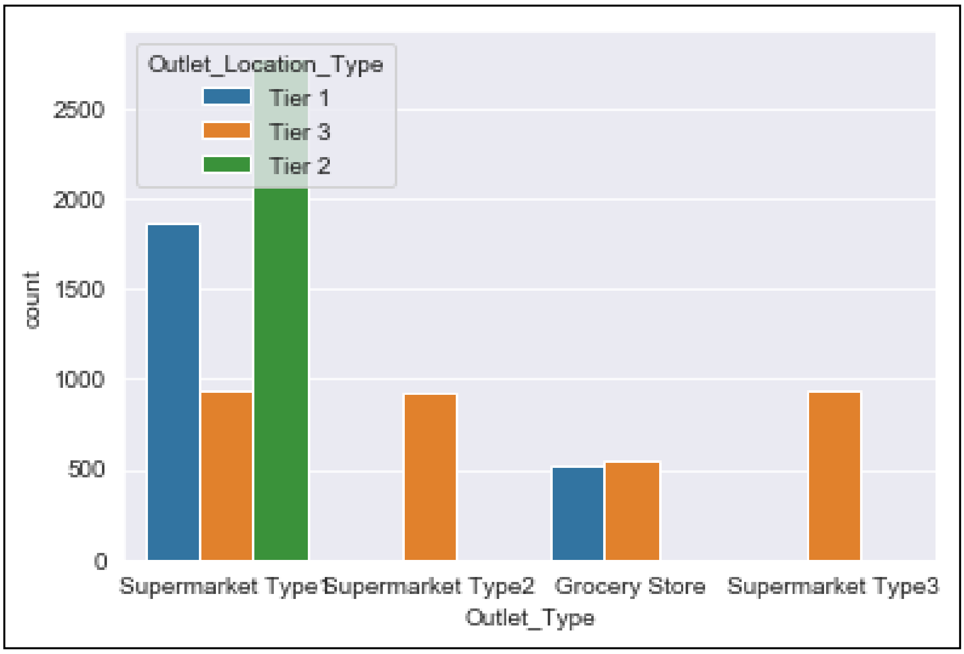

Fig No. 3: Counts of the Outlet Store

- 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

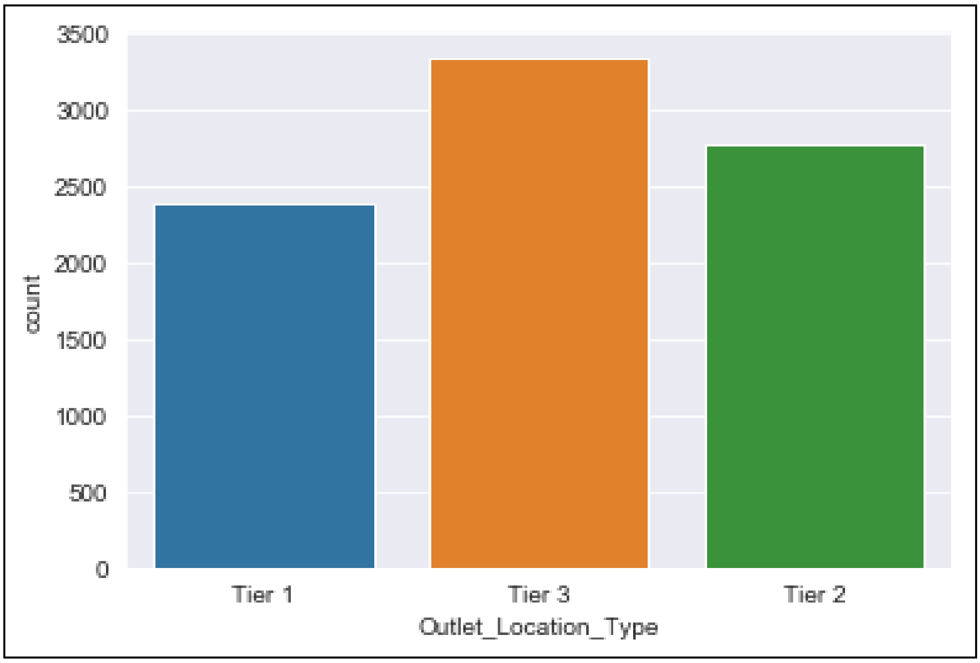

Fig No. 4: Counts of Tier

5. Item Fat Contents

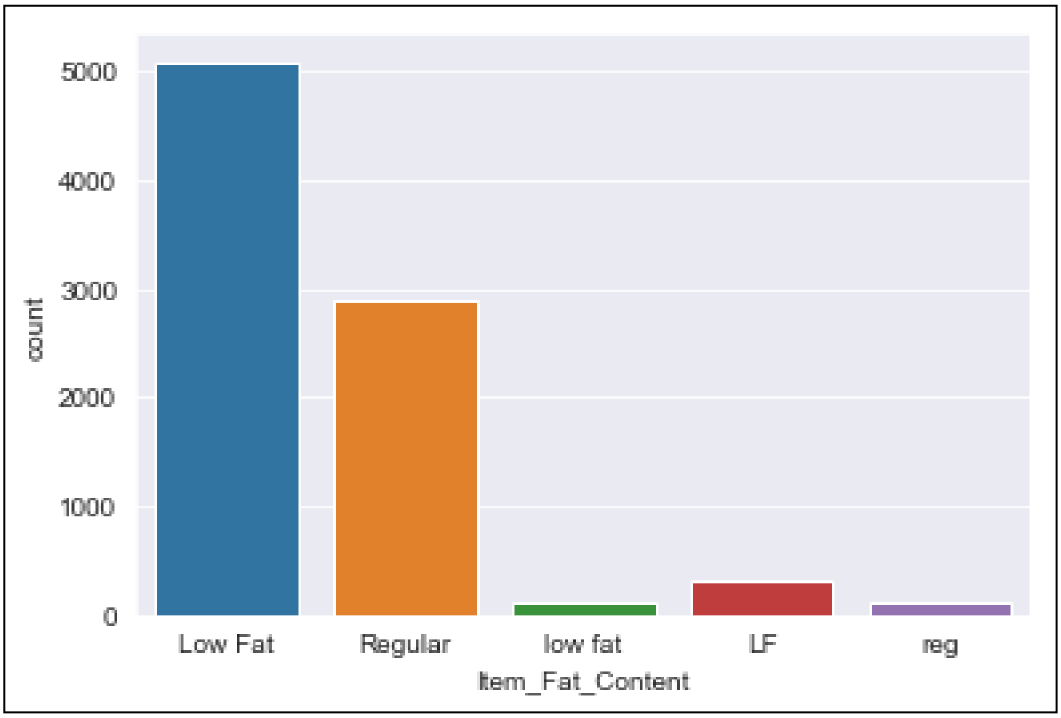

Fig No. 5: Item Fat Content

6. Outlet Size

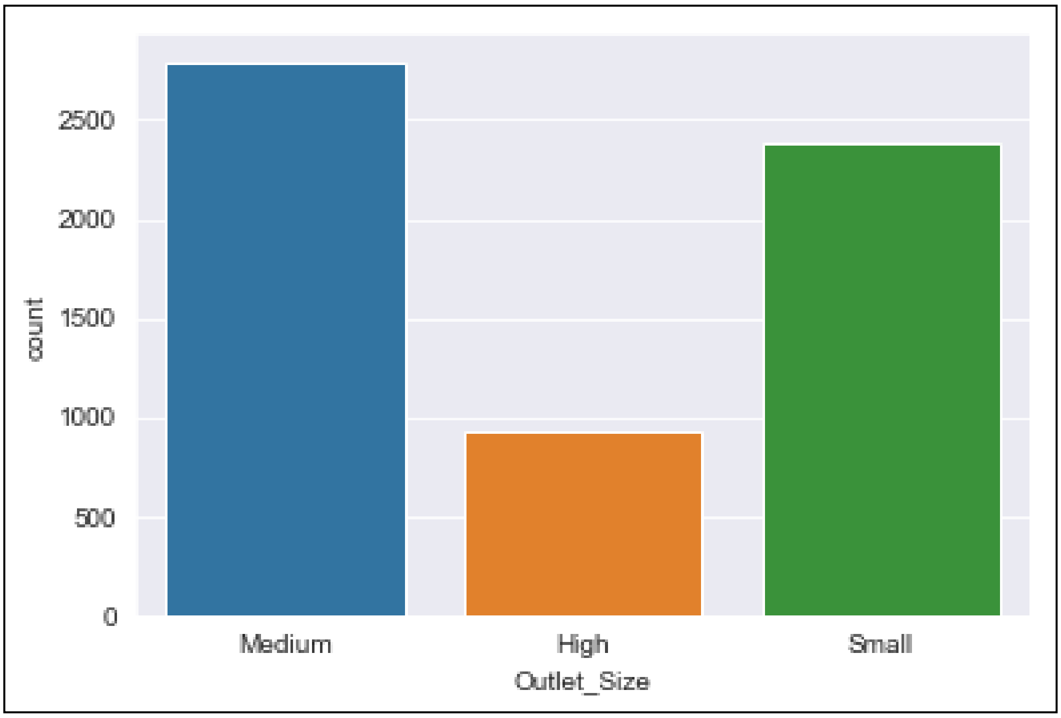

Fig No. 6: Outlet Size

7. Density of Item MRP

Fig No. 7: Density of Item MRP

Looking at the density of the list price of items ***(Item_MRP)***, we clearly see that there are four different price categories. To differentiate between them we introduced a new factor with four price levels: ***Low***, ***Medium***, ***High*** and ***Very High***.

8. Sales vs Outlet Type

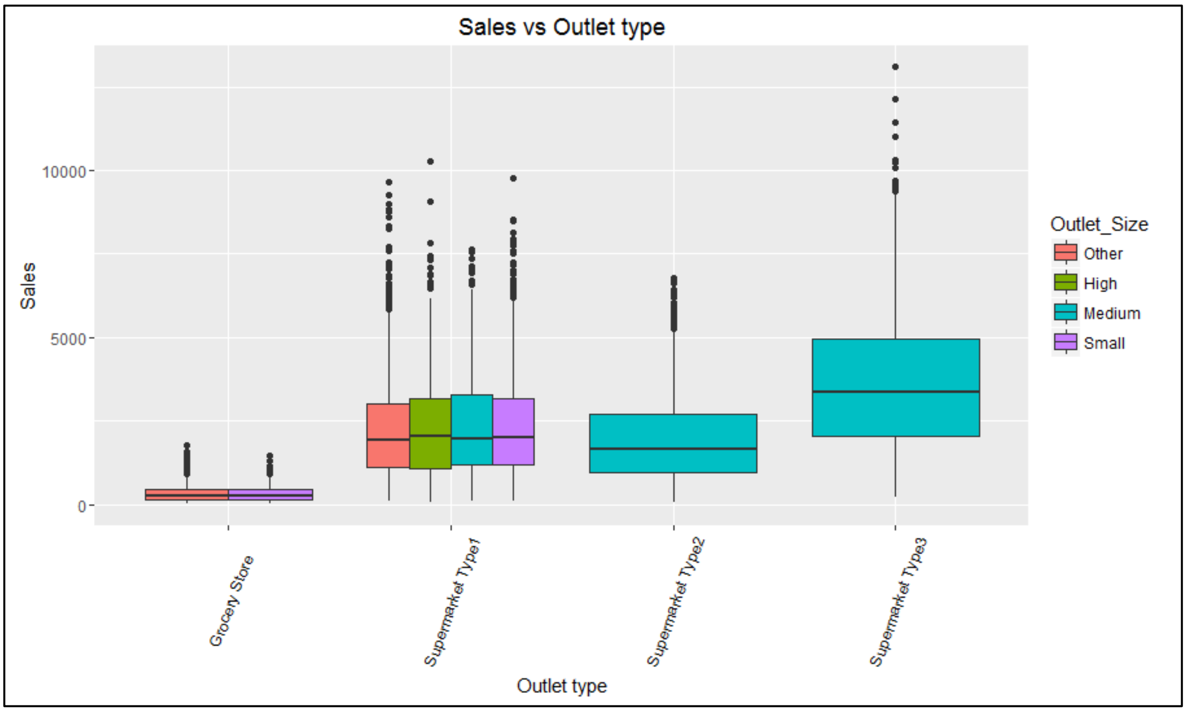

Fig No. 8: Sales vs Outlet Type

We see that there is a clear distinction in sales figures between grocery stores and supermarkets. This is confirmed if we look at sales figures across various item categories:

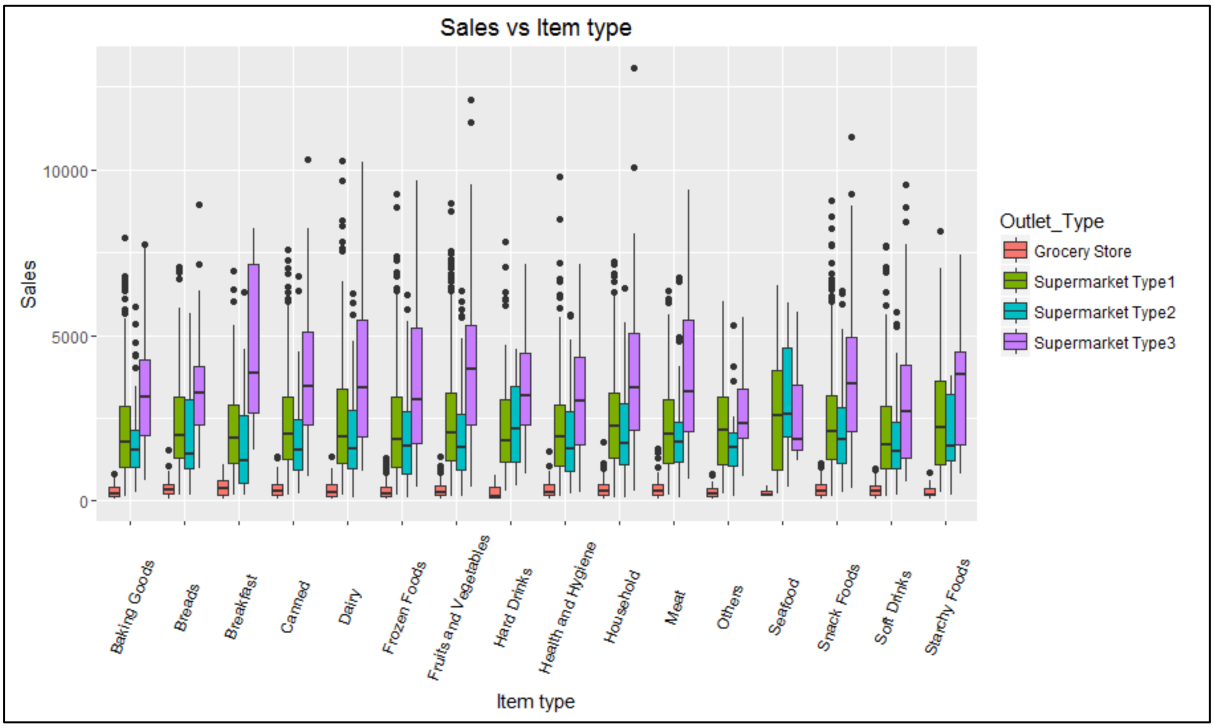

Fig No. 9: Sales vs Item Type

However, the various types of supermarkets cannot be distinguished that easily. This is probably due to other factors, e.g. their location, how long they have been in operation, how well they are managed, etc. In particular, sales in the one Type 2 supermarket in the data are somewhat low. This may be due to the fact that it is still fairly new, having been founded four years ago.

The missing values in the outlet size category concern one grocery store and two type 1 supermarkets. From what we have seen above, the grocery store clearly falls in the category ***Small.*** From the sales figures the type 1 supermarkets could be either ***Small*** or ***Medium***. Since type 1 supermarkets are most often classified as small, we replace those missing size levels by ***Small***.

9. Item Visibility Distribution

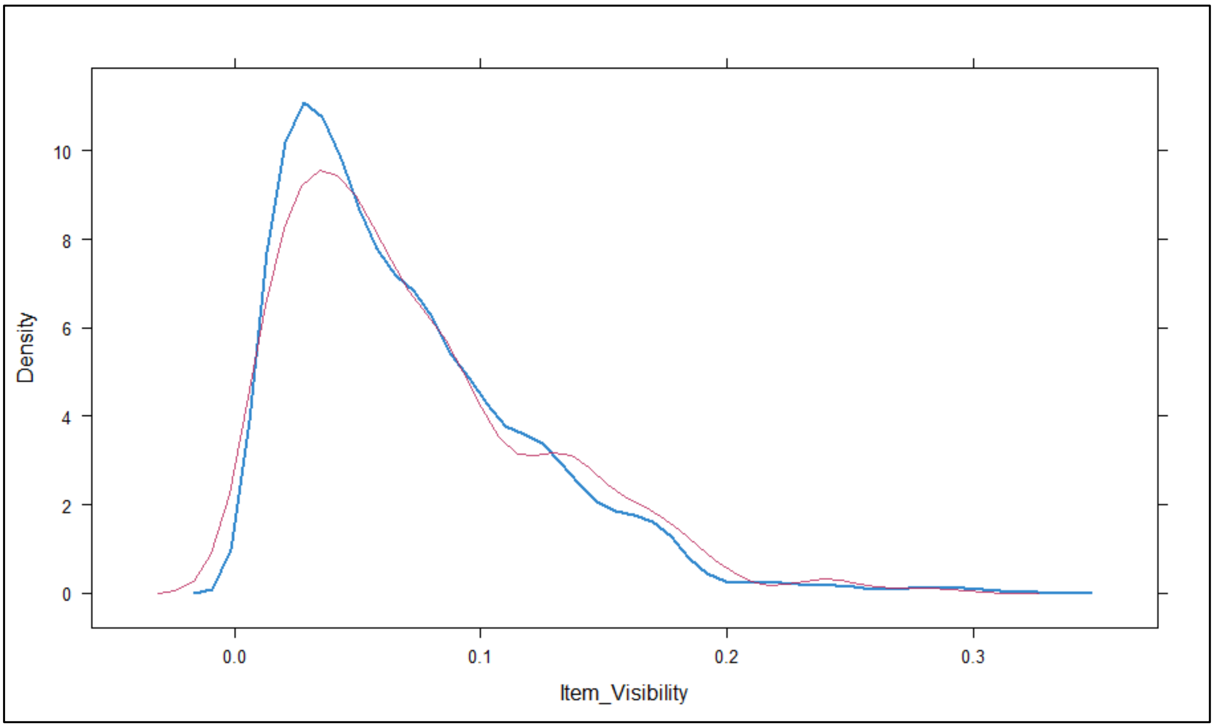

Fig No. 10: Item Visibility Distribution

The percentage of display space in a store devoted to that particular item. Looking at the average visibility of items in each shop, neatly confirms our earlier suspicion that grocery stores have a smaller selection of wares on offer, i.e. the average visibility per item is higher than in supermarkets. Also, we again see that the median visibilities in supermarkets on the one hand and grocery stores on the other are suspiciously similar.
We see that the two distributions are reasonably close to each other.

10. Correlation between Numerical Variables

Looking at correlations between numerical variables one notices a strong positive correlation of 0.57 between *Item_MRP* and *Item_Outlet_Sales* and a somewhat weaker negative correlation of -0.13 between *Item_Visibility* and *Item_Outlet_Sales*. This is confirmed by a principle component analysis:

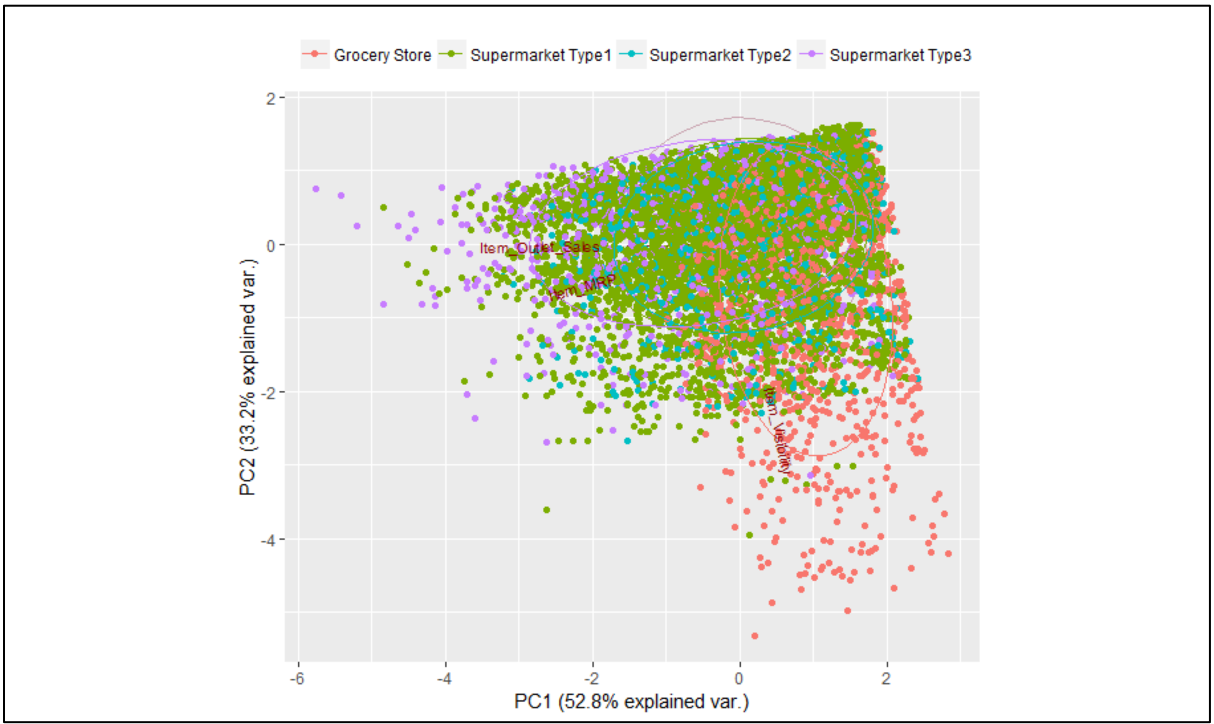

Fig No. 11: Correlation between Numerical Variables

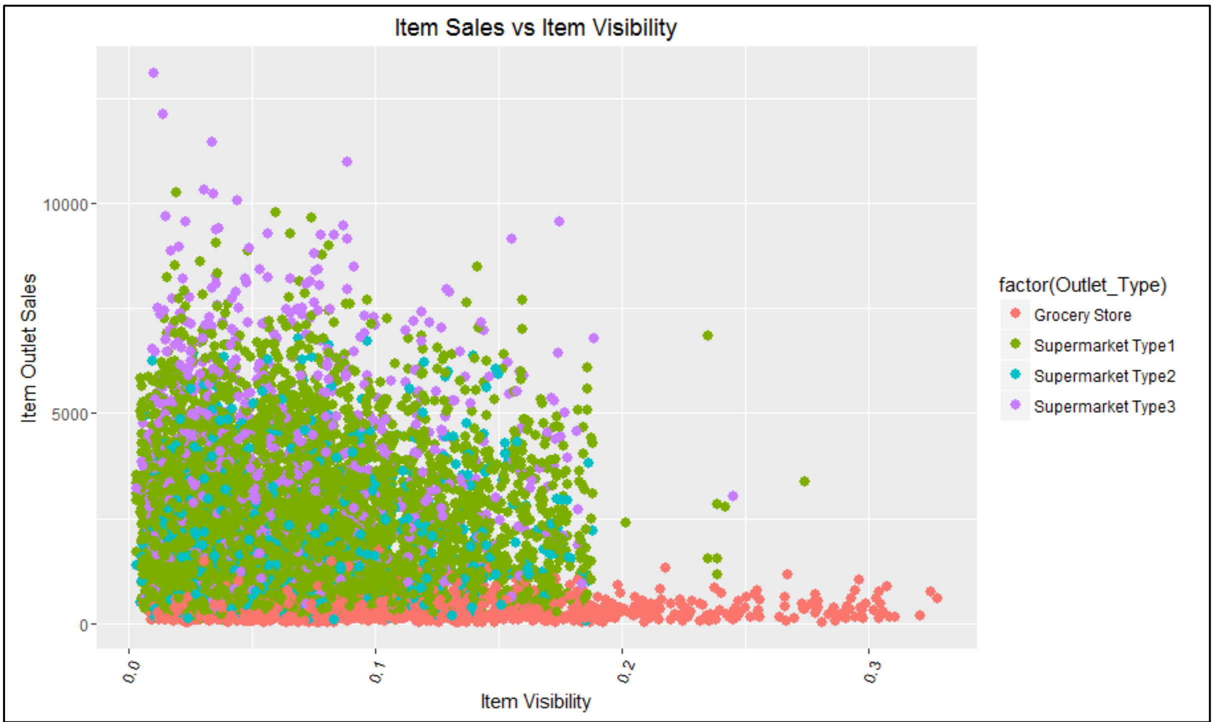

Fig No. 12: Item Sales vs Item Visibility

Again, we notice differences between grocery stores and supermarkets. This is clearly seen in a scatter plot of sales vs. visibilities.

3. Feature Extraction

1. Grabbing 1st characters of Item Identifier

'FD' – Food
'NC' – Non-Consumable
'DR' - Drinks

```
Food              10201
Non-Consumable     2686
Drinks             1317
Name: Item_Type_Combined, dtype: int64
```

2. Combining Item_Fat_Content

Mismatch type variables like 'low-fat', 'LF', 'reg' can be combined to 'Low Fat' and 'Regular'

```
Low Fat    9185
Regular    5019
Name: Item_Fat_Content, dtype: int64
```

3. Importing Label Encoder from sklearn

Approach to encoding categorical values is to use a technique called label encoding. Label encoding is simply converting each value in a column to a number.

We have done Label Encoding for 'Item_Fat_Content', 'Outlet_Location_Type', 'Outlet_Size', 'Item_Type_Combined', 'Outlet_Type', 'Outlet'

4. Getting Dummy variables

A dummy variable is a numerical variable used in regression analysis to represent subgroups of the sample in the study. The dummy variables act like 'switches' that turn various parameters on and off in an equation.

Also, particularly in regression analysis, a dummy variable (also known as an indicator variable, design variable, Boolean indicator, binary variable, or qualitative variable) is one that takes the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected

We have created dummy variables of 'Item_Fat_Content', 'Outlet_Location_Type', 'Outlet_Size', 'Outlet_Type', 'Item_Type_Combined', 'Outlet'