Technocolabs
BUILD WITH AI

If we cannot do great things, we can do small things in a great way.

Delivering smart solutions to smart people | follow us @technocollabs | mail us technocollabs@gmail.com

BIG MART SALES PREDICTION PROJECT REPORT

Submitted by:

Geetika Chawla

Data Science Intern

(Technocolabs)

GitHub : geeet

LinkedIn : geetika-chawla-

E-mail : geetikachawla1010@gmail.com

Project Description:

The data scientists at Big Mart have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim of this data science project is to build a predictive model and find out the sales of each product at a particular store. Using this model, Big Mart will try to understand the properties of products and stores which play a key role in increasing sales.

Dataset:

The dataset is taken from analyticsvidhya.com. There are two different .csv files namely, “Train.csv” having 8523 records and “Test.csv” having 5681 rows. Train data set has both input and output variable(s) We need to predict the sales for test data set.

VARIABLE	DESCRIPTION
Item_Identifier	**Unique ID Product**
Item_Weight	**Weight of Product**
Item_Fat_Content	**Whether the product is low fat or not**
Item_Visibility	**% of total display area in store allocated to this product**
Item_Type	**Category to which product belongs**
Item_MRP	**MRP of the product**
Outlet_Identifier	**Unique store ID**
Outlet_Establishment_Year	**Year in which store was established**
Outlet_Size	**Size of the store**
Outlet_Location_Type	**Type of city in which the store is located**
Outlet_Type	**Grocery store or some sort of supermarket**
Item_Outlet_Sales	**Sales of the product in particular store (Predictor Variable or Response Variable)**

Hypotheses:

1. Store Level Hypotheses:

City type	Stores located in urban or Tier 1 cities should have higher sales because of the higher income levels of people there.
Population Density	Stores located in densely populated areas should have higher sales because of more demand.
Store Capacity	Stores which are very big in size should have higher sales as they act like one-stop-shops and people would prefer getting everything from one place
Competitors	Stores having similar establishments nearby should have less sales because of more competition.
Marketing	Stores which have a good marketing division should have higher sales as it will be able to attract customers through the right offers and advertising.
Location	Stores located within popular marketplaces should have higher sales because of better access to customers.
Customer Behaviour	Stores keeping the right set of products to meet the local needs of customers will have higher sales.
Ambiance	Stores which are well-maintained and managed by polite and humble people are expected to have higher footfall and thus higher sales.

2. Product Level Hypotheses: -

Brand	Branded products should have higher sales because of higher trust in the customer.
Packaging	Products with good packaging can attract customers and sell more.
Utility	Daily use products should have a higher tendency to sell as compared to the specific use products.
Display Area	Products which are given bigger shelves in the store are likely to catch attention first and sell more.
Visibility in Store	The location of product in a store will impact sales. Ones which are right at entrance will catch the eye of customer first rather than the ones in back.
Advertising	Better advertising of products in the store will should higher sales in most cases.
Promotional Offers	Products accompanied with attractive offers and discounts will sell more.

Data Structure and Content:

Train dataset has 8523 rows and 12 features whereas, test dataset has 5681 rows and 11 features. Train has 1 extra column which is the target variable. It is generally a good idea to combine both train and test data sets into one to perform feature engineering and then divide them later again. This saves the trouble of performing the same steps twice on test and train.

```
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            14204 non-null  object
 1   Item_Weight                11765 non-null  float64
 2   Item_Fat_Content           14204 non-null  object
 3   Item_Visibility            14204 non-null  float64
 4   Item_Type                  14204 non-null  object
 5   Item_MRP                   14204 non-null  float64
 6   Outlet_Identifier          14204 non-null  object
 7   Outlet_Establishment_Year  14204 non-null  int64
 8   Outlet_Size                10188 non-null  object
 9   Outlet_Location_Type       14204 non-null  object
 10  Outlet_Type                14204 non-null  object
 11  Item_Outlet_Sales          8523 non-null   float64
 12  source                     14204 non-null  object
dtypes: float64(4), int64(1), object(8)
memory usage: 1.4+ MB
```

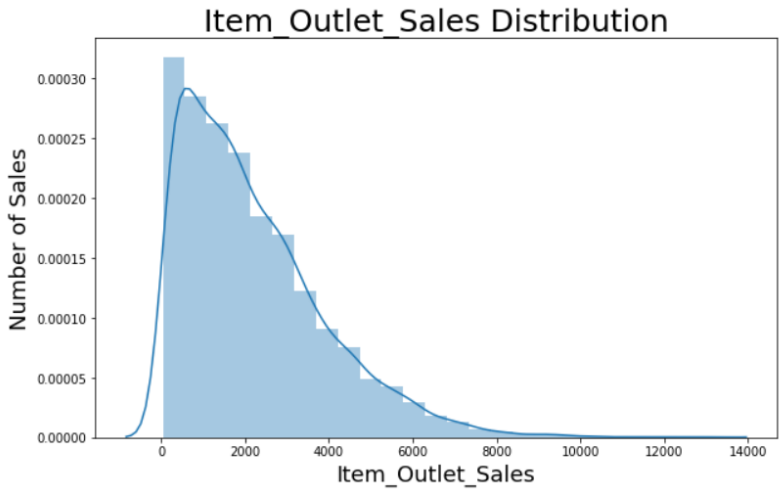
The resultant shape of concatenated data is 14204 rows and 13 columns. Here, we can see missing values invite variables “Item_Weight”, “Outlet_Size” and “Item_Outlet_Sales”.

```
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          5681
source                        0
dtype: int64
```

- ✓ The “Item_Outlet_Sales” is the target variable and 5681 missing values are coming from the test dataset. So, we need not worry about it.
- ✓ “Item_Weight” has 2439 (17.2%) missing values and
- ✓ “Outlet_Size” has 4016 (28.3%) missing values.

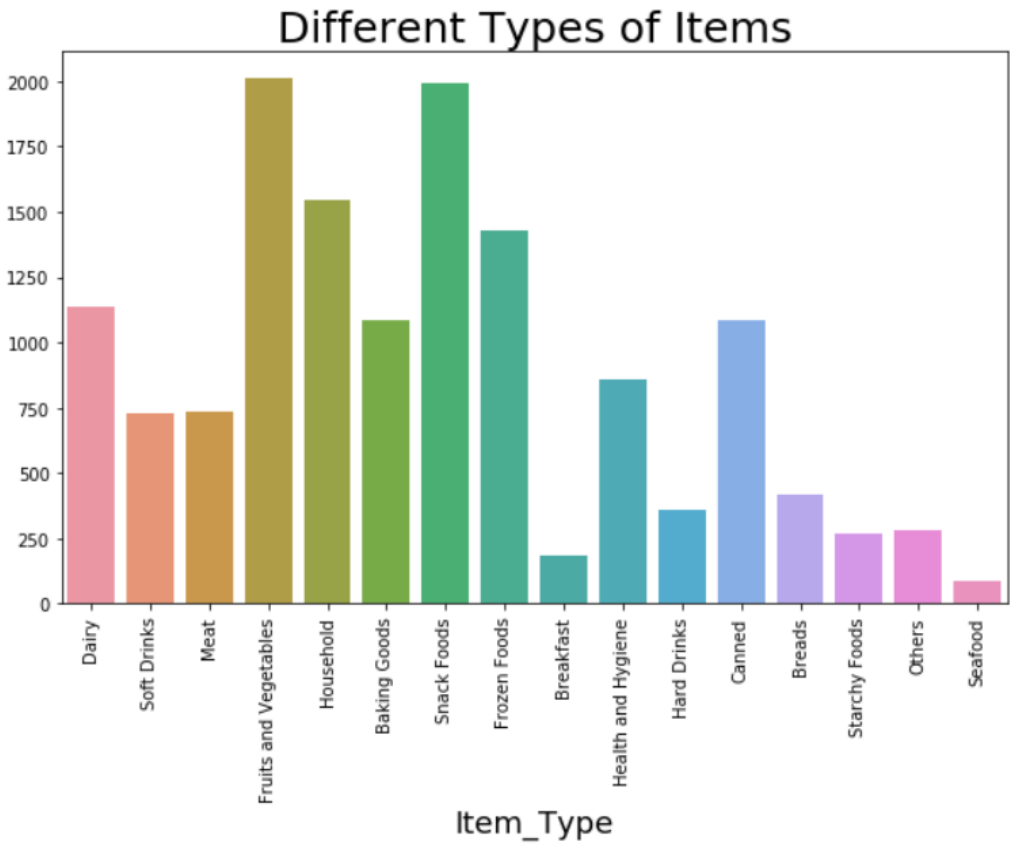
Univariate Analysis:

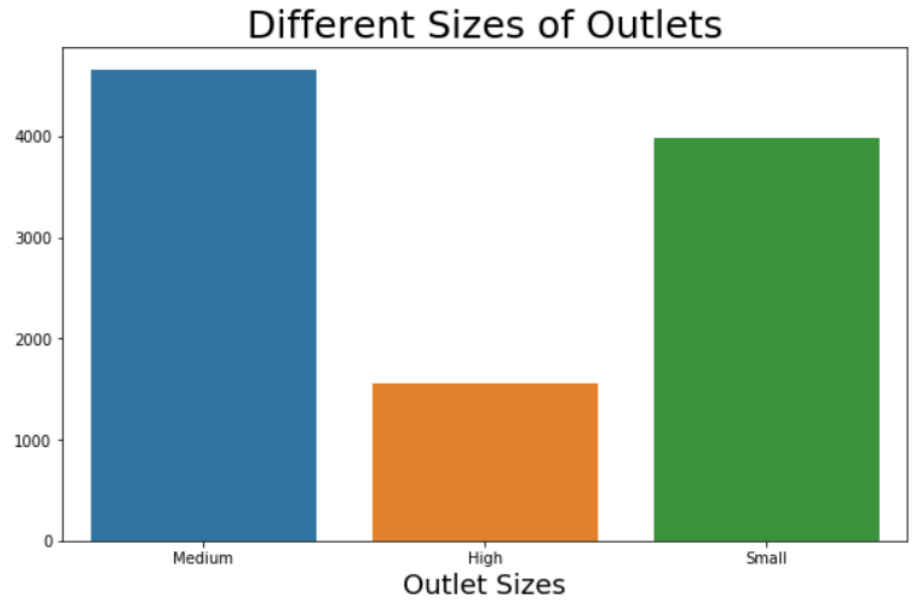
1. Distribution of “Item_Outlet_Sales”

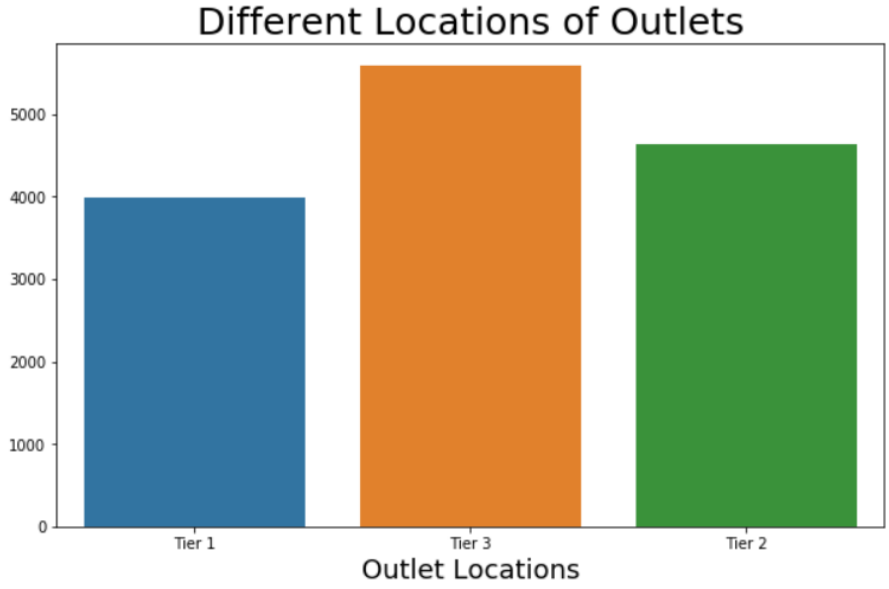


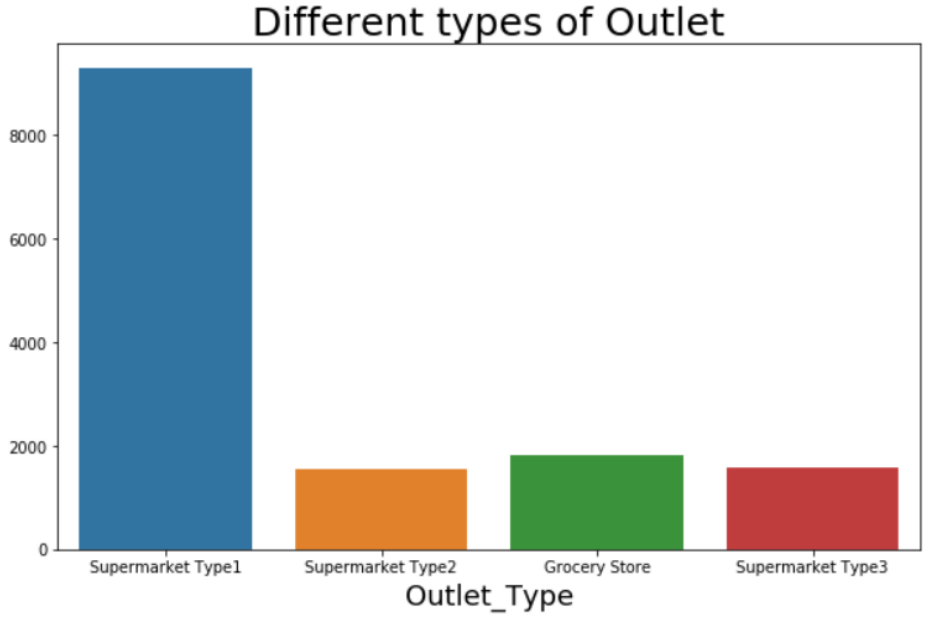
- Our target variable i.e. “Item_Outlet_Sales” follows Normal Distribution.
- It is a right skewed variable and would need some data transformation to treat its skewness.
- Skewness > 1 which indicates that the distribution is highly positively skewed.
- Kurtosis > 1 shows that the distribution is leptokurtic.

2. Exploring Categorical Variables

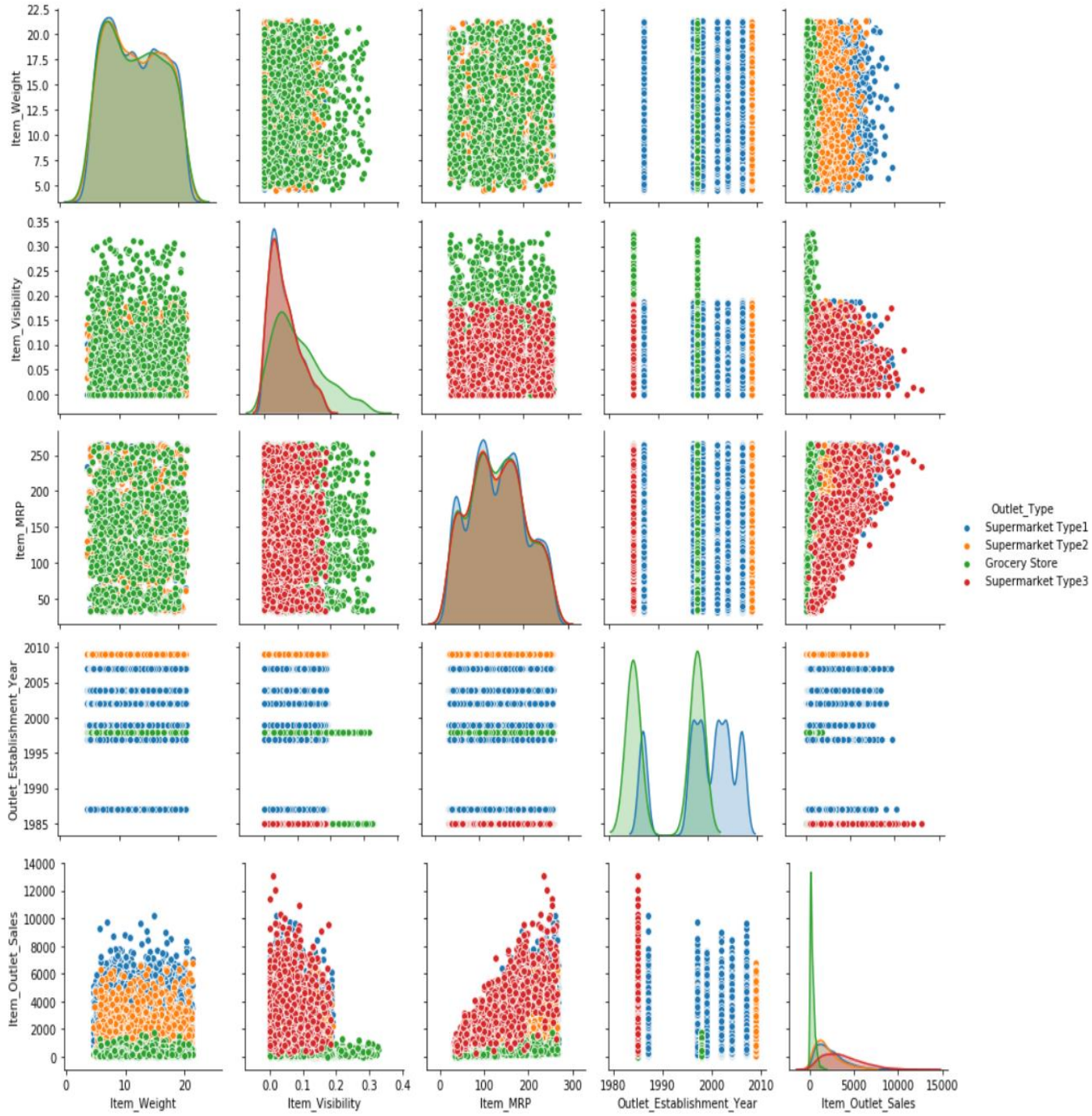



Different Sizes of Outlets
4000
3000
2000
1000
0
Medium
High
Small
Outlet Sizes


Different Locations of Outlets
5000
4000
3000
2000
1000
0
Tier 1
Tier 3
Tier 2
Outlet Locations

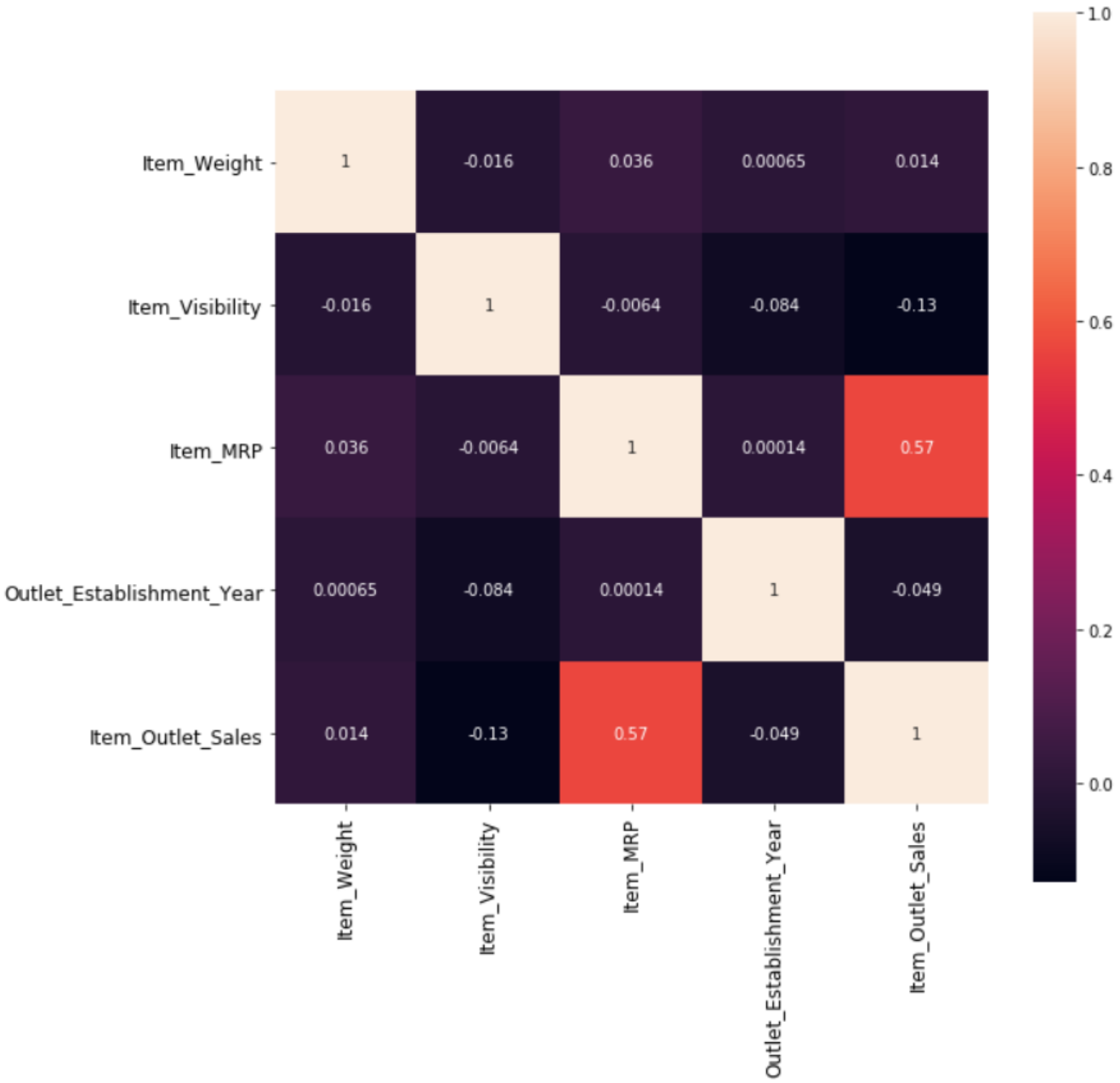

Different types of Outlet
8000
6000
4000
2000
0
Supermarket Type1
Supermarket Type2
Grocery Store
Supermarket Type3
Outlet_Type

Bivariate Analysis:



Pair Plot of all the variables with plot aspect as “Outlet_Type”.

Below is the heatmap showing correlation between all the numerical variables.



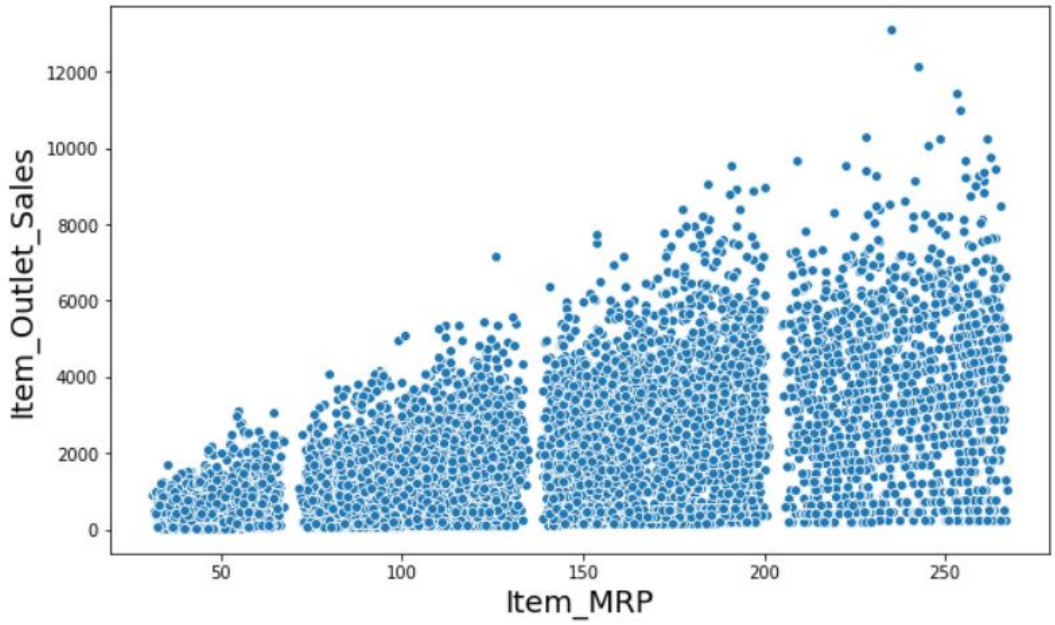
Co-relation Table:

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
Item_Weight	1.000000	-0.015901	0.036236	0.000645	0.014123
Item_Visibility	-0.015901	1.000000	-0.006351	-0.083678	-0.128625
Item_MRP	0.036236	-0.006351	1.000000	0.000141	0.567574
Outlet_Establishment_Year	0.000645	-0.083678	0.000141	1.000000	-0.049135
Item_Outlet_Sales	0.014123	-0.128625	0.567574	-0.049135	1.000000

- Item_Weight has almost negligible correlation (1.4%) with the target variable Item_Outlet_Sales.
- Item_Visibility is having nearly zero correlation (-13%) with the target variable. This means that the sales are not affected by visibility of item which is a contradiction to the general assumption of “more visibility thus, more sales”.
- Item_MRP is positively correlated with sales at an outlet, which indicates that the price quoted by an outlet plays an important factor in sales. Variation in MRP quoted by various outlets depends on their individual sales.
- Outlets situated in location with type tier 2 and medium size are also having high sales, which means that a one-stop-shopping-centre situated in a town or city with populated area can have high sales.

Relations between target variable (“Item_Outlet_Sales”) and other numerical variables.

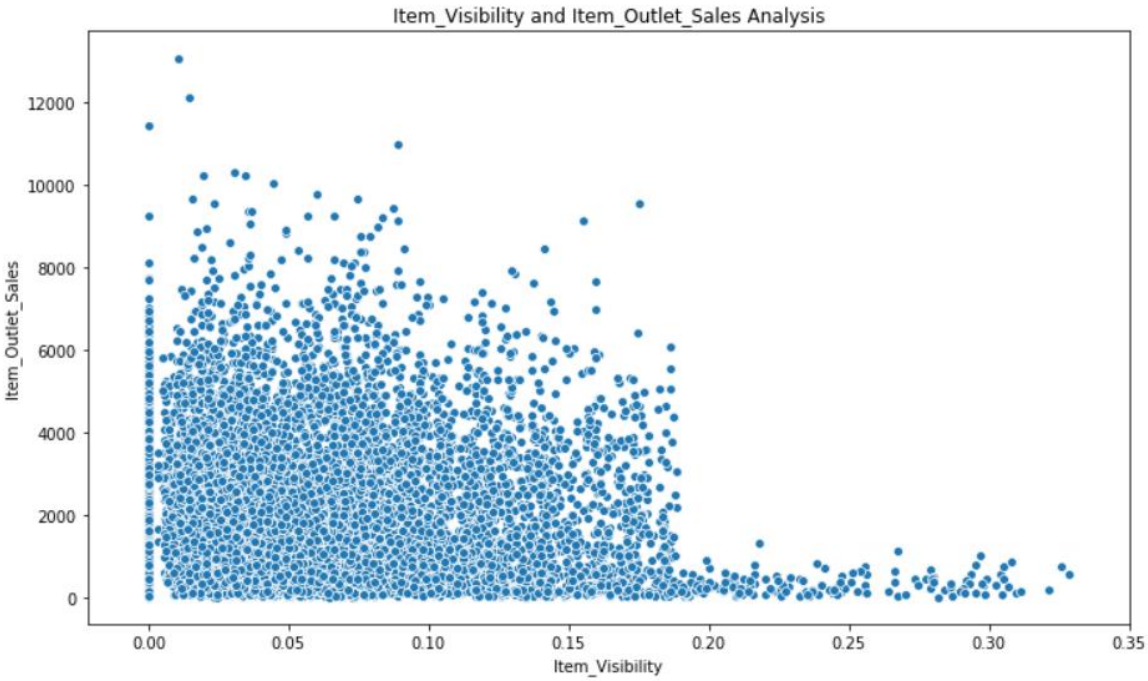
1. Item_MRP and “Item_Outlet_Sales



We can observe that,

- Item_Outlet_Sales increases as the Item_MRP of the item increases.
- There are four segments of prices.
- The price range of MRP 150 to 250 has the highest range of products available.

2. “Item_Visibility and “Item_Outlet_Sales”

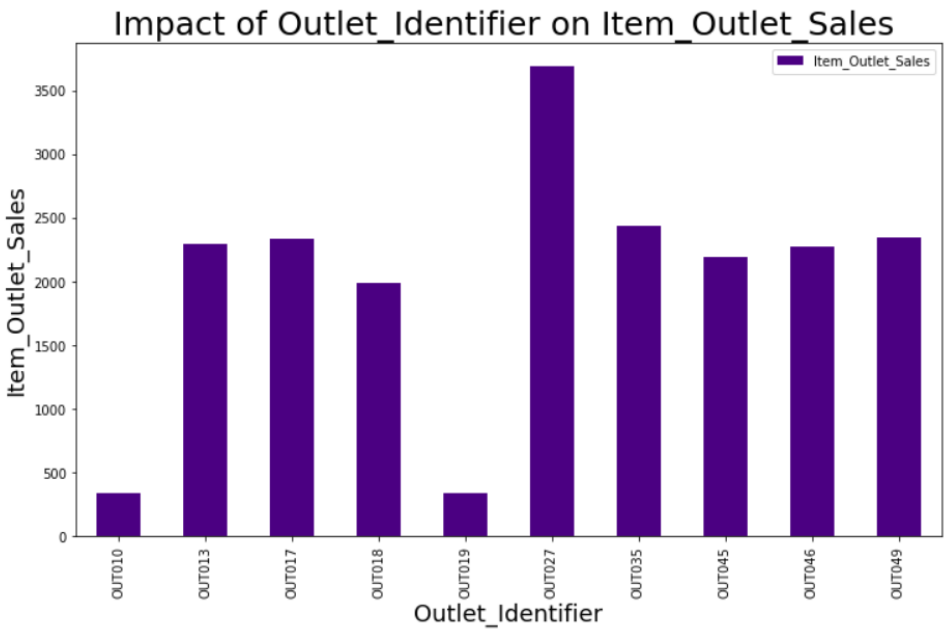


- Here, we can see that there is a sudden decrease in the sales as visibility of the items increases. For “Item_Visibility” > 0.18 (approx.), we see a major decrease in outlet sales.
- Less visible items are sold more compared to more visibility items as outlet contains daily used items which contradicts the null hypothesis.
- There is a string of points at Item_Visibility = 0.0 which seems strange as item visibility cannot be completely zero.

3. Item_Weight and Item_Outlet_Sales Analysis

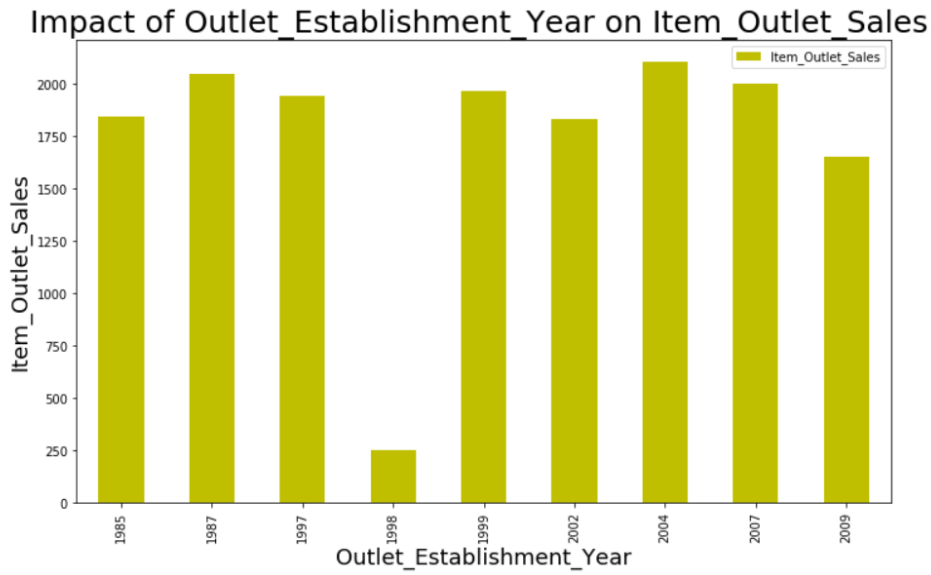
- Item_Outlet_Sales is spread well across the entire range of the Item_Weight without any obvious pattern.
- Item_Weight is shown to have a low correlation with the target variable.

4. Impact of Outlet_Identifier on Item_Outlet_Sales Analysis



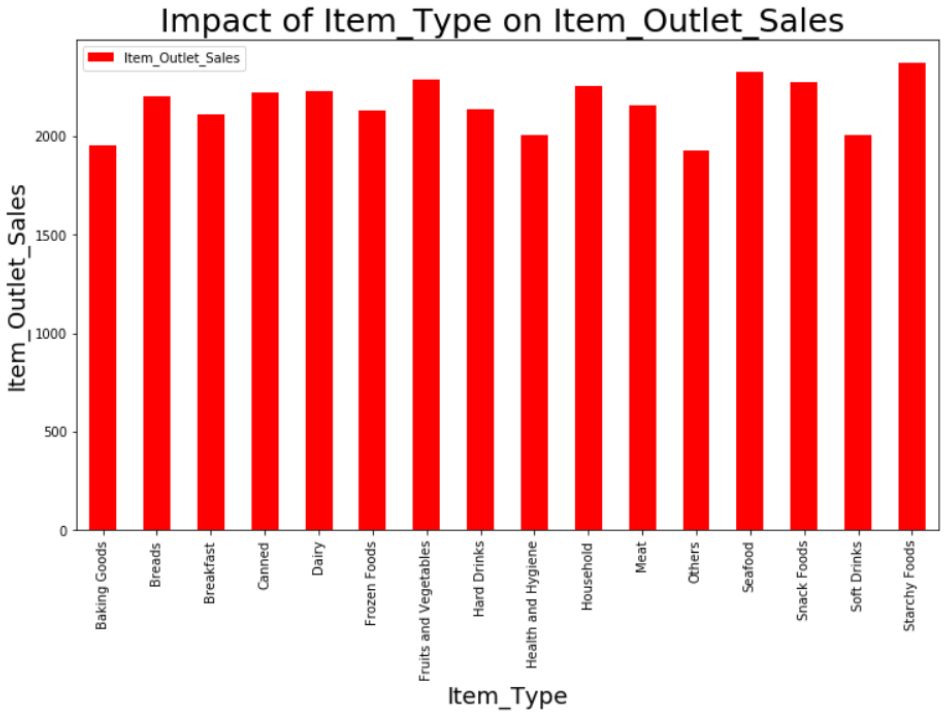
- The average sales are around 2000.
- 'OUT027' has the highest sales.
- 'OUT010' and 'OUT019' has a quite similar distribution depicting very smaller number of sales.

5. Outlet_Establishment_Year and Item_Outlet_Sales Analysis



All the outlets established had great sales in all the years from 1985 to 2009 except in the year 1998, which experienced less sales.

6. Impact of Item_Type on Item_Outlet_Sales Analysis



Distribution of Item_Outlet_Sales across the categories of Item_Type is not very distinct.

Treating Irregularities in the Data:

Item_Fat_Content had some irregularities i.e. typing errors. We have, therefore, replaced “LF” as “Low Fat”, “reg” as “Regular” and “low fat” as “Low Fat”.

```
Original Categories:
Low Fat    8485
Regular    4824
LF          522
reg         195
low fat     178
Name: Item_Fat_Content, dtype: int64

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

Missing Value Treatment:

Filling all the missing values in the numerical column “Item_Weight” with average of all the values in it.

```
item_avg_weight = df.groupby(["Item_Identifier"])["Item_Weight"].mean()
item_avg_weight
miss_bool = df['Item_Weight'].isnull()
print('Orignal #missing: %d'% sum(miss_bool))
df.loc[miss_bool,'Item_Weight'] = df.loc[miss_bool,'Item_Identifier'].apply(lambda x: item_avg_weight.loc[x])
print('Final #missing: %d'% sum(df['Item_Weight'].isnull()))
```

```
Orignal #missing: 2439
Final #missing: 0
```

In categorical column “Outlet_Size”, filled all the missing values with mode of that column.

```
from scipy.stats import mode
outlet_size_mode = df.pivot_table(values='Outlet_Size', columns='Outlet_Type',aggfunc=(lambda x:mode(x).mode[0]))
print('Mode for each Outlet_Type:')
print(outlet_size_mode)
miss_bool = df['Outlet_Size'].isnull()
print('\nOrignal #missing: %d'% sum(miss_bool))
df.loc[miss_bool,'Outlet_Size'] = df.loc[miss_bool,'Outlet_Type'].apply(lambda x: outlet_size_mode[x])
print('\nFinal #missing: %d'%sum(df['Outlet_Size'].isnull()))
```

```
Mode for each Outlet_Type:
Outlet_Type Grocery Store Supermarket Type1 Supermarket Type2  \
Outlet_Size         Small             Small            Medium

Outlet_Type Supermarket Type3
Outlet_Size            Medium

Orignal #missing: 4016

Final #missing: 0
```

Feature Engineering:

The Item_Type variable has 16 unique categories which might prove to be very useful in analysis. So, it is a good idea to combine them. If we look at the Item_Identifier, i.e. the unique ID of each item, it starts with either FD, DR or NC. If we see the categories, these look like being Food, Drinks and Non-Consumables.

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

Marking non-consumables as separate category named as “Non-Edible” in low_fat

```
Low Fat       6499
Regular       5019
Non-Edible    2686
Name: Item_Fat_Content, dtype: int64
```

Modelling:

The following models have been used:

- Linear Regression
- Regularized Linear Regression
  - Ridge Regression
  - Lasso Regression
- Decision Tree
- Random Forest
- XGBoost

• Since, scikit-learn accepts only numerical variables, we need to convert all categories of nominal variables into numeric type variables. I have created a new variable Outlet same as Outlet_Identifier but encoded that.

• One-Hot-Coding refers to creating dummy variables, one for each category.

• For example, Item_Fat_Content has 3 categories - 'Low Fat', 'Regular' and 'Non-Edible'. One hot coding will remove this variable and will generate 3 new variables instead of 1. Each will have binary numbers – 0 (if the category is not present) and 1(if category is present).

• Similarly, One-Hot Encoding has been performed on the variables, namely, 'Item_Fat_Content', 'Outlet_Location_Type', 'Outlet_Size', 'Outlet_Type', 'Item_Type_Combined' and 'Outlet'.

• Mean Absolute Error (MAE) represents the difference between the original and predicted values extracted by averaging the absolute difference over the data. • Root Mean Squared Error (RMSE) is the square root of mean squared error (MSE) which is the average of the square of the difference between the original and predicted values of the data. RMSE is basically the square root of the variance of the residuals. • The RMSE indicates the absolute fit of the model i.e. how close the observed data points are to the model's predicted values.

• RMSE is a good measure of how accurately the model predicts the response and is the most important criterion for fit if the main purpose of the model is prediction. Lower values of RMSE indicate better fit.

Summary:

Comparing the Cross-Validation Scores of different models.

Model	CV score (mean)	CV score (std)
Linear Regression	1129	48.52
Ridge Regression	1130	49.7
Lasso Regression	1128	48.64
Decision Tree	1089	47.80
Random Forest	1081	45.74
XGBoost	1258	58.10

Comparison of MAE, RMSE and accuracy of different models.

Model	Mean Squared Error	Root Mean Square Error
Linear Regression	836.11	1127
Ridge Regression	836.02	1129
Lasso Regression	835.44	1128
Decision Tree	741.63	1058
Random Forest	748.60	1068
XGBoost	248.67	351

Conclusion:

As the profit made by the Big Mart is directly proportional to the accurate predictions of sales, they are desiring more accurate prediction algorithm so that the company will not suffer any losses.

XgBoost has produced more accurate predictions as compared to the other available techniques like linear regression, regularized linear regression, random forest, etc.

It is also concluded that XGBoost with lowest MAE & RMSE and also with the highest accuracy of 95.76%, among all the other models, performs better as compared to the other existing models.