PROJECT PROGRESS REPORT

CS795 - Introduction to Data Science

Team Members

1. **Chandrasekhar Reddy Muthyala**
   UIN: 01088628

   Email : *cmuth001@odu.edu*

2. **Puneeth Shankar Bikkasandra**
   UIN: 01101060

   Email : *pbikk001@odu.edu*

Table of Contents

1. Abstract:

The data scientists have collected sales data for around 1500 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim of this project is to build a predictive model and find out the sales of each product at a particular store. Using this machine learning model, the shopping chain will try to understand the properties of products and stores which play a key role in increasing sales.

2. Objective:

Build a predictive model to find out the future sales of each product at a particular store.

3. Target Audience:

Shop Owners and Shopping Chain Management.

4. Tools & Technologies:

Below are the technologies we used and where we used in our project

- **Coding Platform:** Python 3 Jupyter Notebook
- **Pandas Library:** We used this library to load the CSV data and used for data munging and preparation.[1]
- **scikit-learn:** It's a machine learning library, we used this library for regression analysis of sales data.
- **Matplotlib:** It's a 2D plotting library which produces a good quality of images to download and visualize data. In this project, we used this library to plot graphs in the data exploration.[2]
- **Seaborn:** it is a data visualization library and build on top of matplotlib, will give a high-level and high-quality interface for plotting attractive statistical graphs. We used this library in data exploration section. [3]
- **Numpy:** it's a core library for scientific computations in python and also provides a high-performance of multi-dimensional array objects, we used this library for finding absolute , mean, standard deviation, etc.,[4]

5. System Architecture & Design:

Train Data: Raw sales data of different stores and is used to train the prediction model.

Test Data: This Data is used to predict the future sales of the products at different stores.

Pre-processing: To achieve better results from the prediction model, the format of the data must be in proper manner. This step includes cleaning, removing un-wanted attributes, replacing missing values with meaning full data and transformations.

Feature Extraction: After prep-processing the data, analyze the data and finding the core feature responsible for prediction model. In this stage includes combining multiple attributes and extracting the important feature for the prediction model.

Removing Outliers: After finding core features, remove outliers of the important features. In this process we will identify the threshold of the features and remove data while one falls outside of the fence.

Train Prediction Model: In this stage we have a meaning full and clean data. In this stage we apply a machine algorithm to our training data.

Verify Model: In this stage we will validate and identify the best fit Prediction model. In this process check the model error rate by changing the feature selection and choose the best prediction model.

Prediction Process: After identifying the best fit model, pass the test data to the model. The output of the process is the future sales of each store of each product.

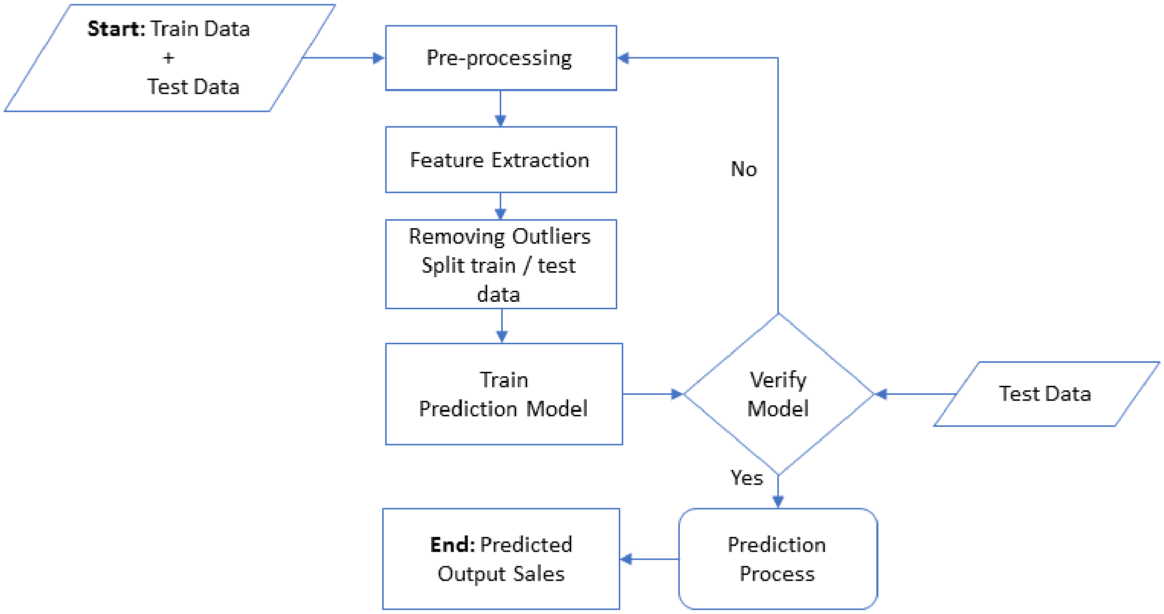


Figure 1.System Architecture

6. Data Exploration:

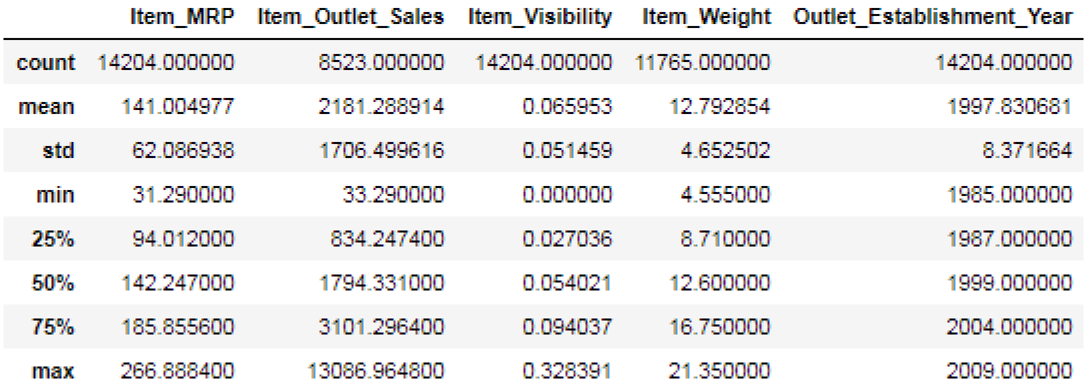
There are no short cuts for the data exploration we have to dive deep in understanding the data to build an accurate modal. First step need to identify predictor variables, target variable, data type of variables and category of variables. [5]

S.No	Variable	Description
1	Item_Identifier	Unique product ID
2	Item_Weight	Weight of product
3	Item_Fat_Content	Whether the product is low fat or not
4	Item_Visibility	The % of total display area of all products in a store allocated to the particular product
5	Item_Type	The category to which the product belongs
6	Item_MRP	Maximum Retail Price (list price) of the product
7	Outlet_Identifier	Unique store ID
8	Outlet_Establishment_Year	The year in which store was established
9	Outlet_Size	The size of the store in terms of ground area covered
10	Outlet_Location_Type	The type of city in which the store is located
11	Outlet_Type	Whether the outlet is just a grocery store or some sort of supermarket
12	Item_Outlet_Sales	Sales of the product in the particular store. This is the outcome variable to be predicted.

Figure 2. Dataset description

7. Dataset Description:

It will help in identifying the importance of each variable and how much percentage of data is distributed across dataset and helps in feature selection for prediction modal.



	Item_MRP	Item_Outlet_Sales	Item_Visibility	Item_Weight	Outlet_Establishment_Year
count	14204.000000	8523.000000	14204.000000	11765.000000	14204.000000
mean	141.004977	2181.288914	0.065953	12.792854	1997.830681
std	62.086938	1706.499616	0.051459	4.652502	8.371664
min	31.290000	33.290000	0.000000	4.555000	1985.000000
25%	94.012000	834.247400	0.027036	8.710000	1987.000000
50%	142.247000	1794.331000	0.054021	12.600000	1999.000000
75%	185.855600	3101.296400	0.094037	16.750000	2004.000000
max	266.888400	13086.964800	0.328391	21.350000	2009.000000

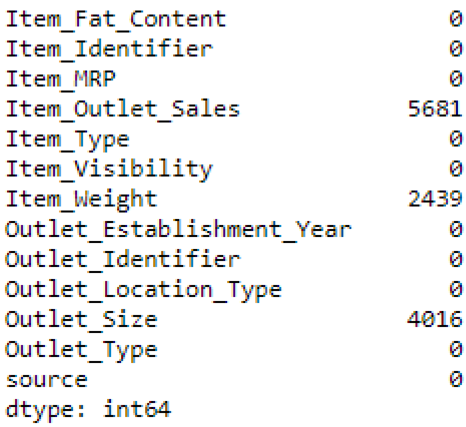
Figure 3. Data description

Some Observations:

1. Item Visibility: It has a minimum value ZERO value in practical scenarios it does not make any sense because every product will occupy at least some amount of space in the store.
2. Outlet Establishment Years: It ranges from 1985 to 2009, find the age of the store with this column. It should have a reasonable impact on the sales of the items.
3. Item Outlet Sales: Considering the maximum difference between minimum value and maximum value, this infers that we have some outliers in this column.

7.1 Identifying Missing Values Count:

Identified which features have the missing values and importance of the feature using pandas describe.

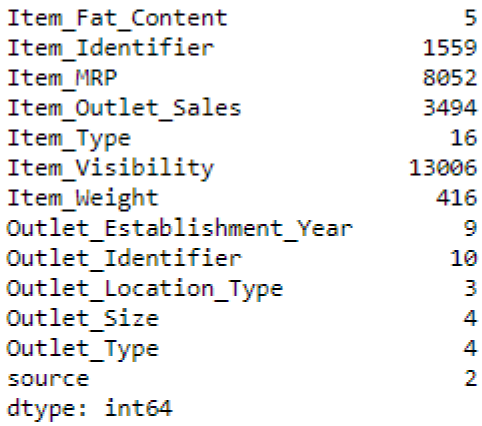


```
Item_Fat_Content                0
Item_Identifier                 0
Item_MRP                        0
Item_Outlet_Sales            5681
Item_Type                       0
Item_Visibility                 0
Item_Weight                  2439
Outlet_Establishment_Year       0
Outlet_Identifier               0
Outlet_Location_Type            0
Outlet_Size                  4016
Outlet_Type                     0
source                          0
dtype: int64
```

Figure 4. Each feature missing values count

7.2 Identifying Unique Values of Each Feature:

Each feature has its own importance in our predictive modal, so Identify each feature unique value in the dataset. In our dataset, below image will give information about unique values. This will help us in identifying what are the unique values and of each feature.



```
Item_Fat_Content                5
Item_Identifier              1559
Item_MRP                     8052
Item_Outlet_Sales            3494
Item_Type                      16
Item_Visibility             13006
Item_Weight                   416
Outlet_Establishment_Year       9
Outlet_Identifier              10
Outlet_Location_Type            3
Outlet_Size                     4
Outlet_Type                     4
source                          2
dtype: int64
```

Figure 5. Unique Values of Each Feature

This will tell us how many stores, unique items in the whole dataset. Item visibility and Item MRP have more variation in dataset.

7.3 Categorical Objects:

To find out categorical objects and numerical feature which will helps in data cleaning. Below are the categorical feature in our dataset.

1. Item_Fat_Content
2. Item_Identifier
3. Outlet_Identifier
4. Item_Type
5. Outlet_Location_Type
6. Outlet_Size
7. Outlet_Type
8. Source

8. Data Cleaning:

This step is more important in building an accurate modal. “Dirty data is pervasive and prevents people from doing useful things,” said Eugene Wu [6]. Data analyzed until now, has shown how the data is spread across with unique and null values. This step mainly involves imputing missing values and treating with outliers. Missing values and Outliers will play a major role in misleading the regression prediction modal.

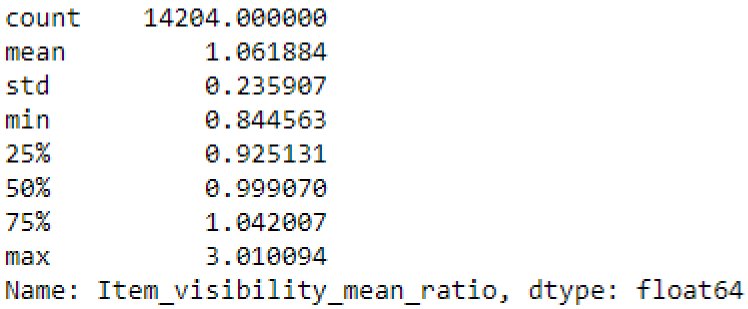
In the data exploration section, we found two features have missing values *Item_Weight* and *Outlet_Size*. To make the right prediction, we have filled *Item_Weight* values with average weight of the particular item and *outlet_size* by mode of the *outlet_size* for the particular object. There are no missing values in the dataset post this step.

9. Feature Engineering:

We have observed some feature with abnormalities in the variation of data in the previous section. In this section we will remove that kind of data and create a new feature with the help of existing features.

9.1. Creating Item Visibility Mean Ratio:

We have observed some of the records in the *Item_Visibility* column being ZERO, which is not making sense in real time situations. Therefore, we will create a new feature "*Item_visibility_mean_ratio*", and fill missing values with average value of that particular item and calculate the mean ratio of each item across all stores.



```
count    14204.000000
mean         1.061884
std          0.235907
min          0.844563
25%          0.925131
50%          0.999070
75%          1.042007
max          3.010094
Name: Item_visibility_mean_ratio, dtype: float64
```

Figure 6. Item_visibility_mean_ratio description

9.2. Create Generalize Item Type:

In the data exploration section we saw there were 16 unique item types present in the dataset which will help in analyzing data. The first two letters of the item, mentions a pattern, such as FD or DR or NC, which corresponds to Food, Drink, Non-consumable. We have created a new category name as "*item_type_generalize*"

• Food	10201
• Non-Consumable	2686
• Drinks	1317

9.3. Creating Outlet Store Age:

Sales of the items is also depending on the outlet store age, because there may be a good chance of people going to new store when compared old store, if they are nearby, so we are creating a new feature "*Outlet_year*".

9.4. Modifying Item Fat Content:

In the earlier section if we observe we have five unique categories, we can observe that there are only two categories in five ways. Therefore, we will correct them by mapping everything into “Low Fat” and “Regular Fat”

Before modification:

Low Fat	8485
Regular	4824
LF	522
reg	195
low fat	178

After modification:

Low Fat	9185
Regular	5019

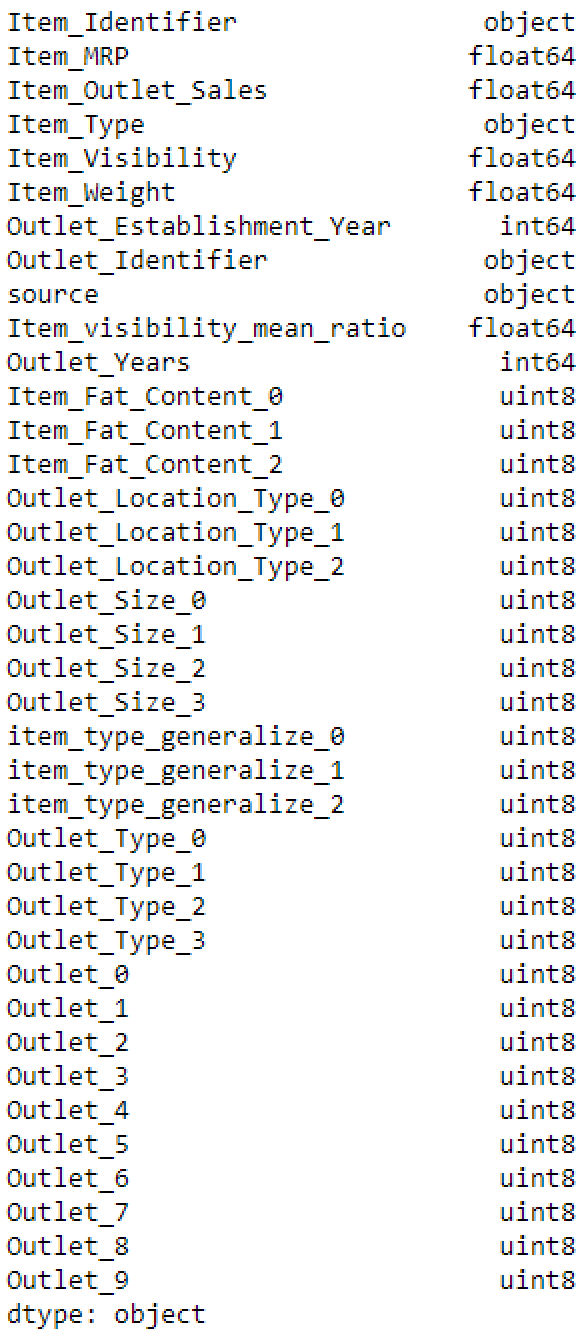
In this section above, we created a Non-consumable product from item identifiers, and by combining the above change we can create a one more category all non-consumable products of low fat are non-edible in item fat content. Basically we are splitting low fat category into another category “Non-Edible”

After splitting:

Low Fat	6499
Regular	5019
Non-Edible	2686

9.5. Converting Categorical Features In To Numerical:

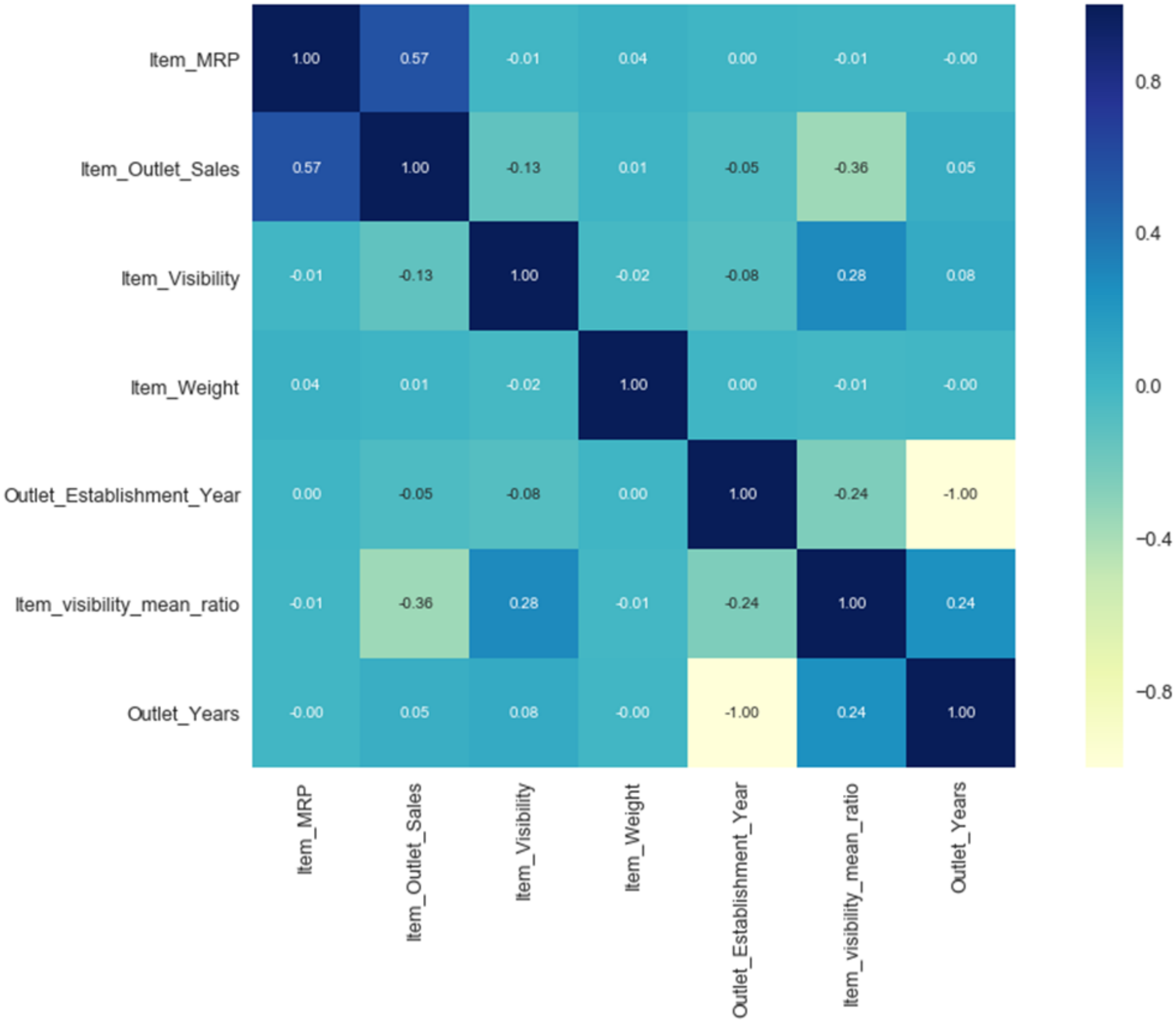
As *sklearn* library accepts only numerical values for prediction model, so in this stage it will convert all the categorical into numerical values and assign some values to it. Above “*Item_Fat_Content*”, we obtained three categories “Low Fat”, “Regular” and “Non-Edible” and will convert it into “Item_Fat_Content_0”, “Item_Fat_Content_1” and “Item_Fat_Content_2” respectively.



```
Item_Identifier                 object
Item_MRP                       float64
Item_Outlet_Sales              float64
Item_Type                       object
Item_Visibility                float64
Item_Weight                    float64
Outlet_Establishment_Year        int64
Outlet_Identifier               object
source                          object
Item_visibility_mean_ratio     float64
Outlet_Years                     int64
Item_Fat_Content_0               uint8
Item_Fat_Content_1               uint8
Item_Fat_Content_2               uint8
Outlet_Location_Type_0           uint8
Outlet_Location_Type_1           uint8
Outlet_Location_Type_2           uint8
Outlet_Size_0                    uint8
Outlet_Size_1                    uint8
Outlet_Size_2                    uint8
Outlet_Size_3                    uint8
item_type_generalize_0           uint8
item_type_generalize_1           uint8
item_type_generalize_2           uint8
Outlet_Type_0                    uint8
Outlet_Type_1                    uint8
Outlet_Type_2                    uint8
Outlet_Type_3                    uint8
Outlet_0                         uint8
Outlet_1                         uint8
Outlet_2                         uint8
Outlet_3                         uint8
Outlet_4                         uint8
Outlet_5                         uint8
Outlet_6                         uint8
Outlet_7                         uint8
Outlet_8                         uint8
Outlet_9                         uint8
dtype: object
```

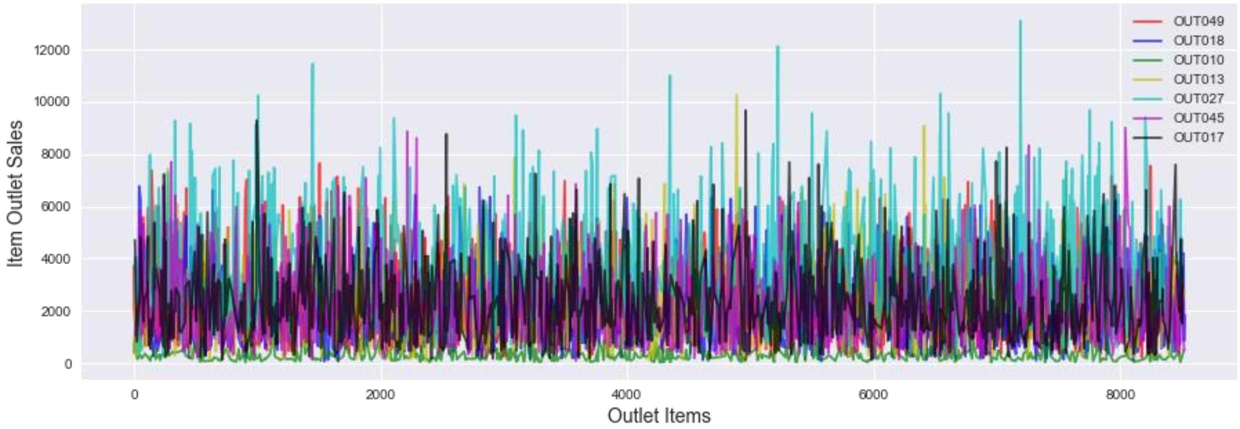
Figure 7. Converted Categorical items into Numerical Items

10. Data Visualization:



10.1. All Stores Sales Data Visualization:

The above diagram will give gist which outlet exactly have more sales on each item. In the below plot unique color differentiate different Outlet. By seeing the plot “OUT027”, “OUT013”, “OUT045” and “OUT017” have more sales in comparison to other Outlets.



10.2. Analysis Of Sales vs Item Type Using Violin Plot:

In this we are finding the distribution of sales data of every item type on the store. This will give the broad information how sales are distributed across the item type. Each violin bar displays the five-number (minimum, first quartile, median, third quartile, and maximum), which is the summary of the data. Below are the five-number distribution of each item type for each outlet.

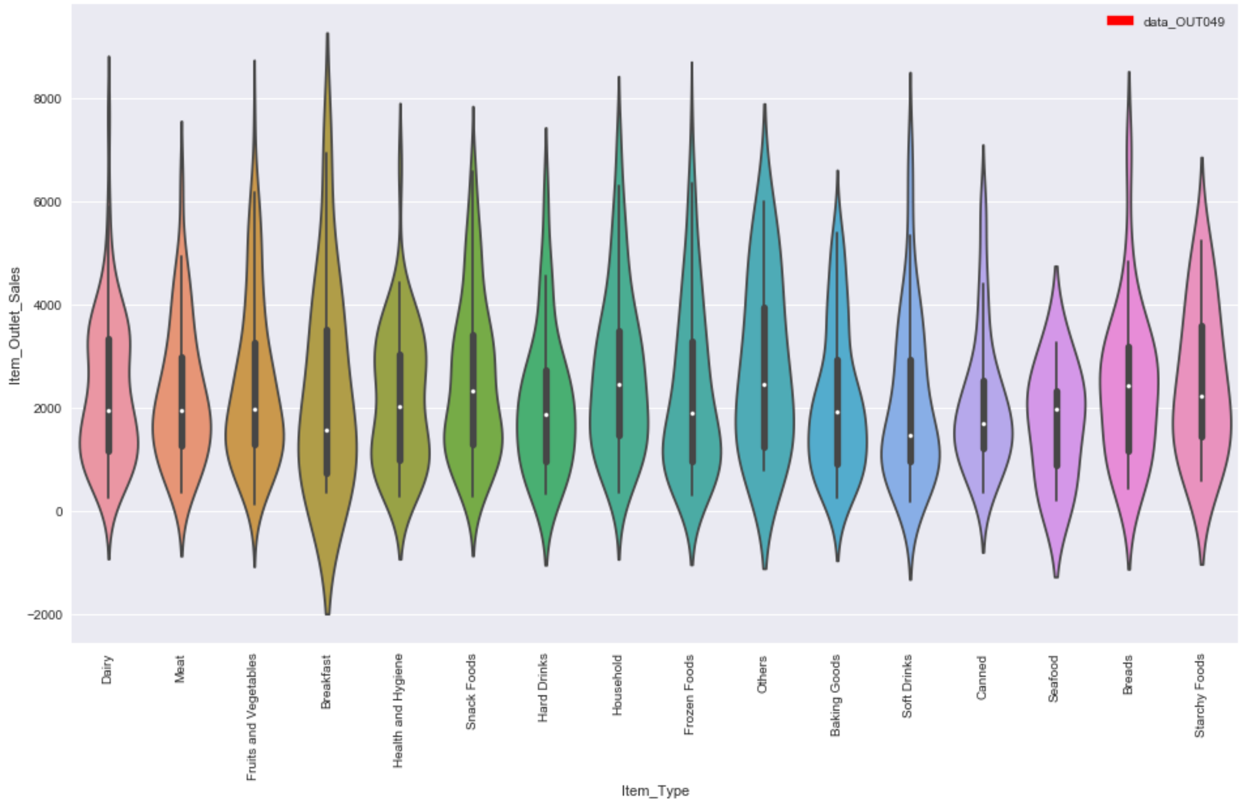


Figure 8. Violin plot of Item type of OUT049 store

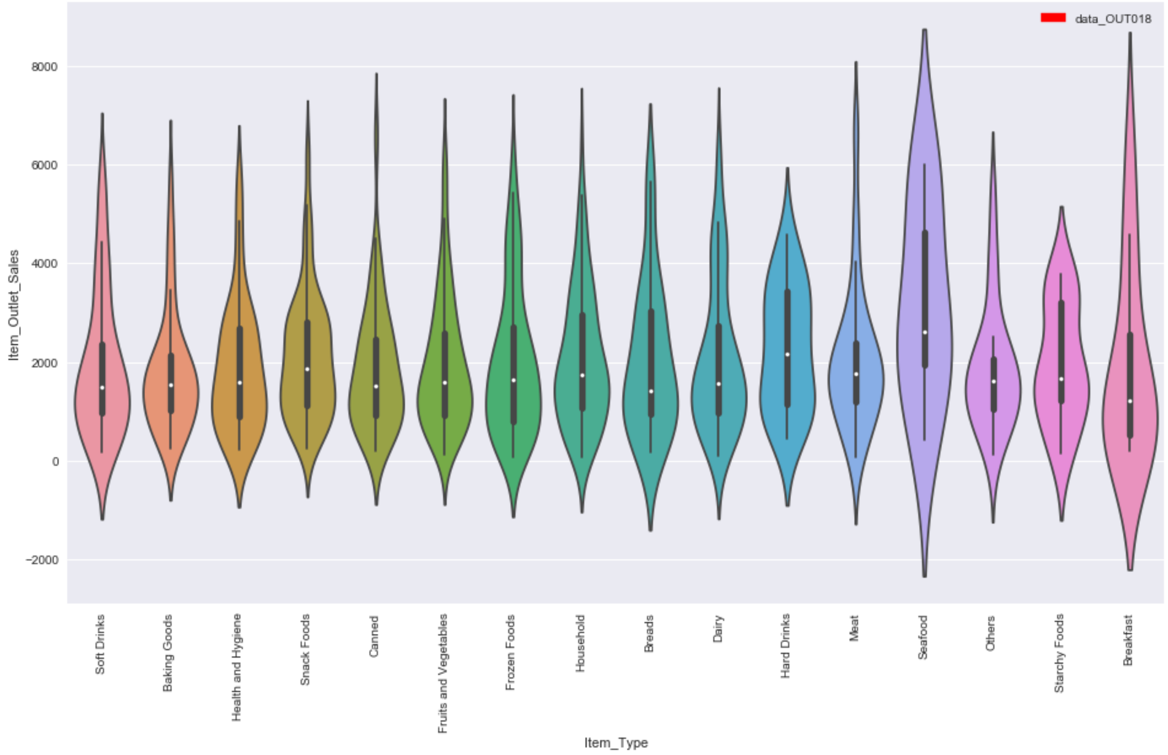


Figure 9.Violin plot of Item type of OUT018 store.

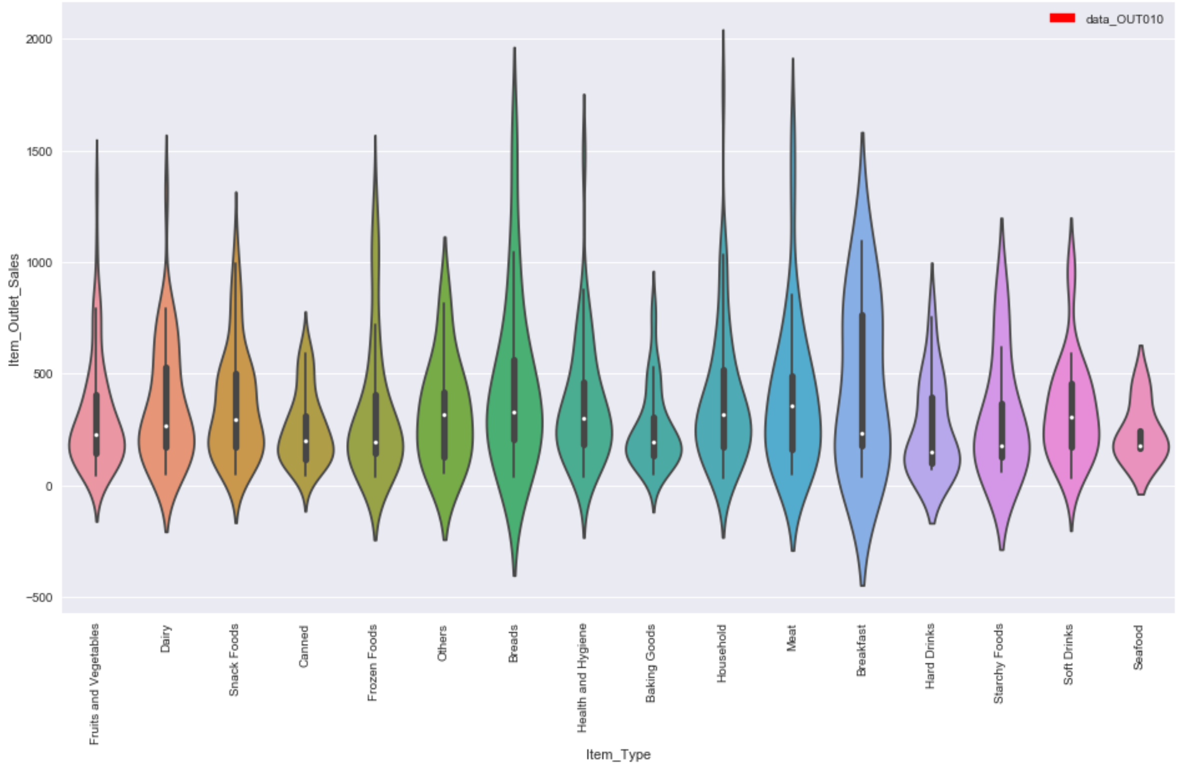


Figure 10. Violin plot of Item type of OUT010 store.

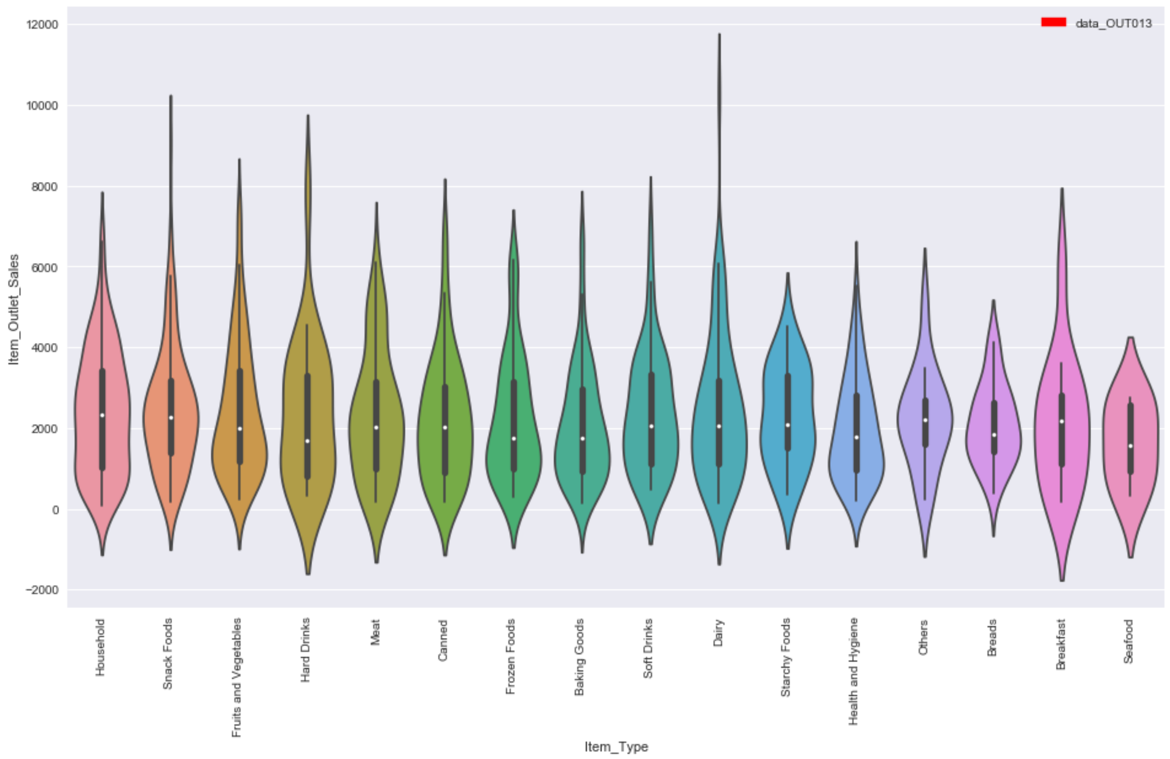


Figure 11. Violin plot of Item type of OUT013store

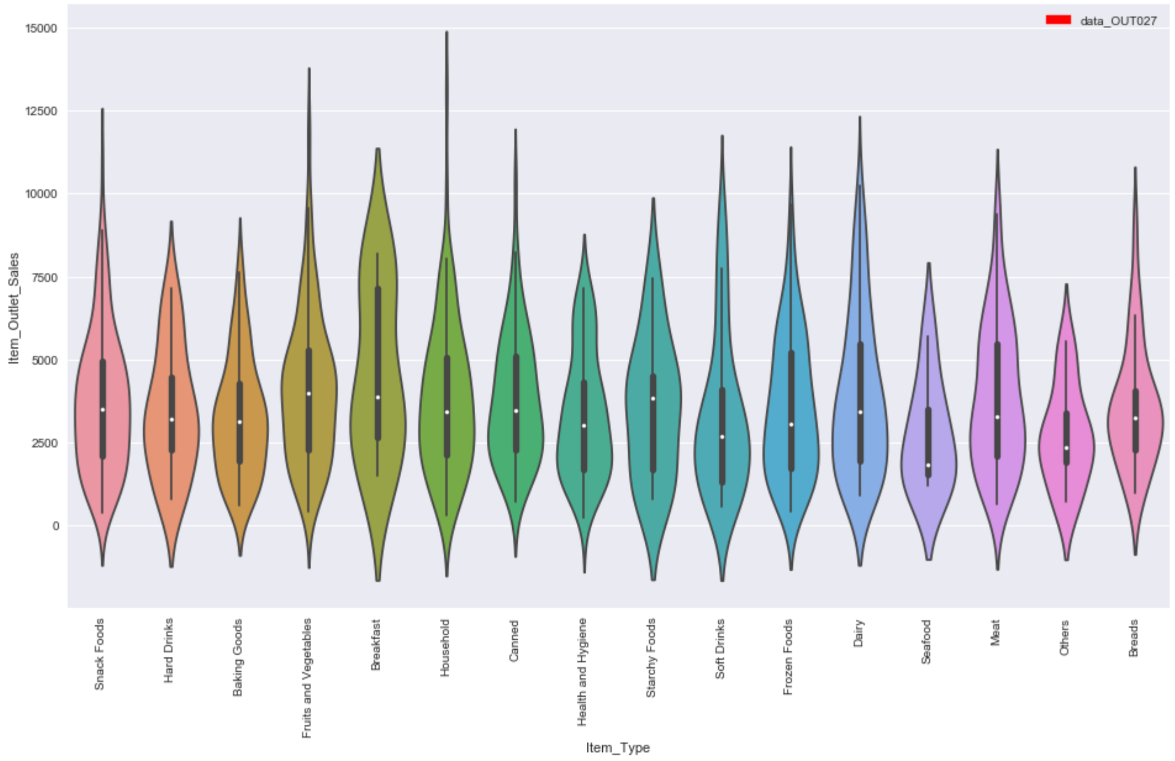


Figure 12.Violin plot of Item type of OUT027 store

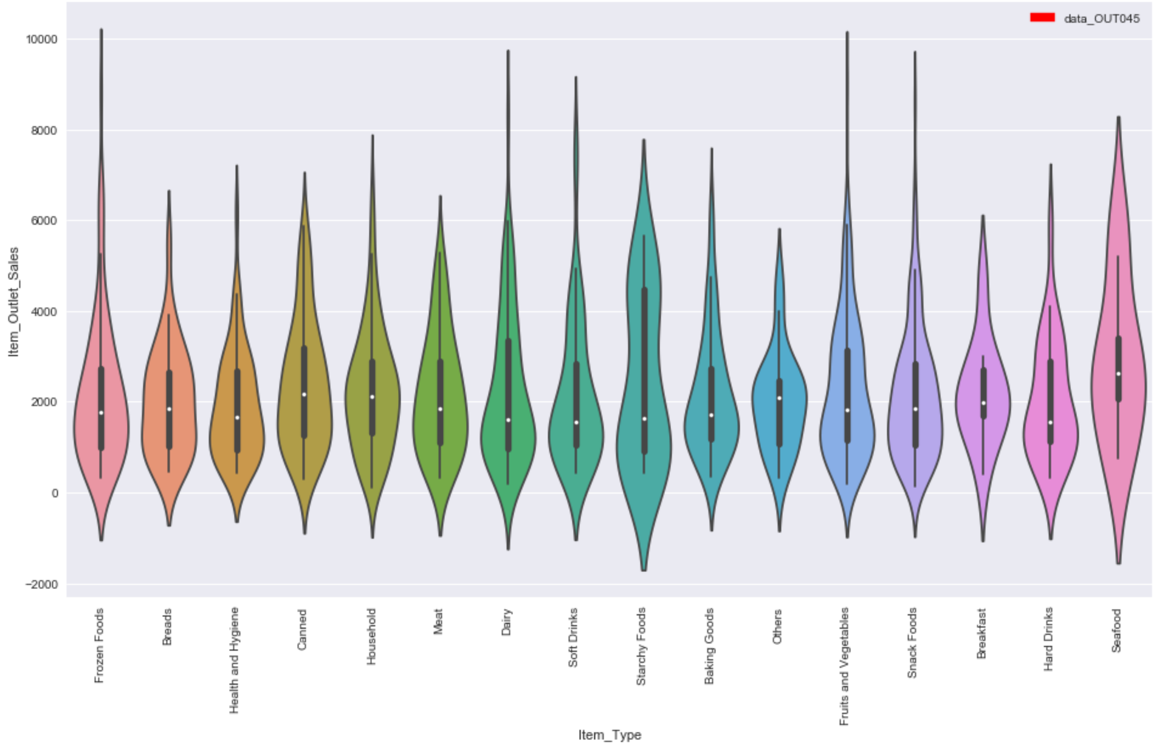


Figure 13. Violin plot of Item type of OUT045 store

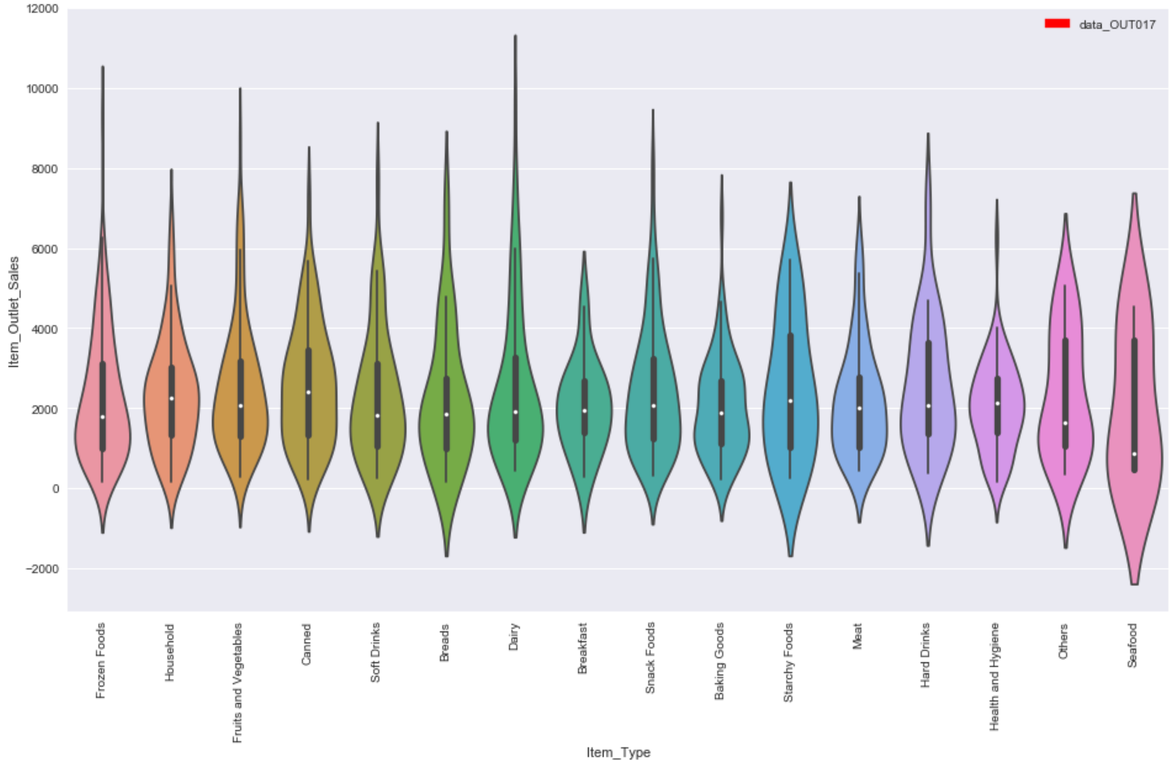


Figure 14. Violin plot of Item type of OUT017 store

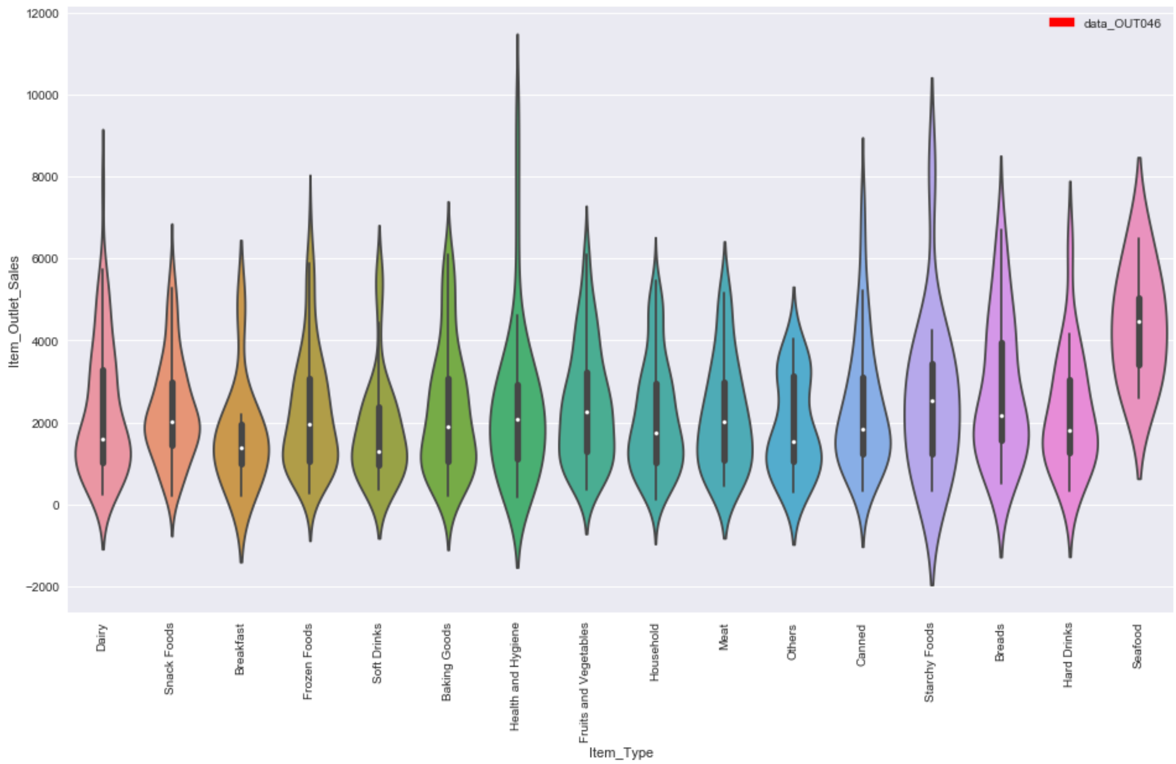


Figure 15. Violin plot of Item type of OUT046 store

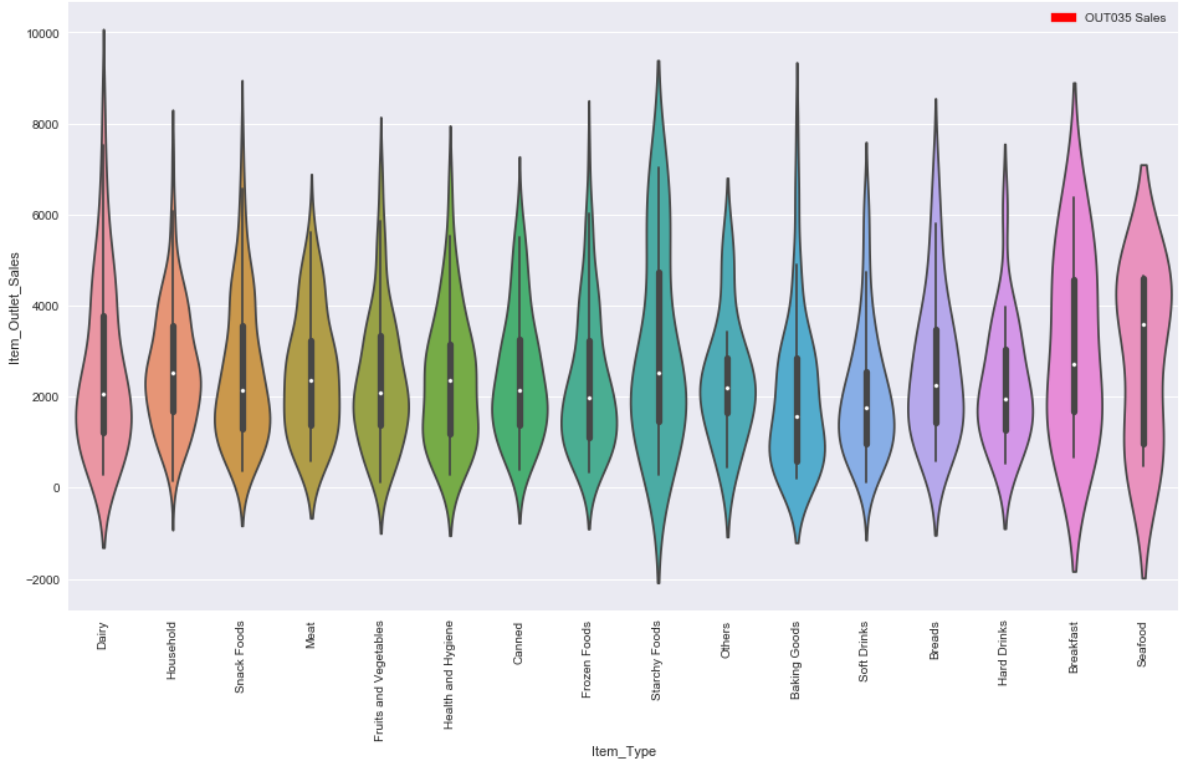


Figure 16. Violin plot of Item type of OUT035 store

10.3. Analysis Of Sales vs Item Visibility:

Until now, we have a visualization of outlet vs item sales, but now we are analyzing a sales data with respect to item visibility. This tells how the sales happening with respect to size of the items on the store. If we see “OUT_27” is having the highest sales compare to other stores, by plot we can observe few outliers, above 10,000 sales, and should replace them with mean of that particular item store value.

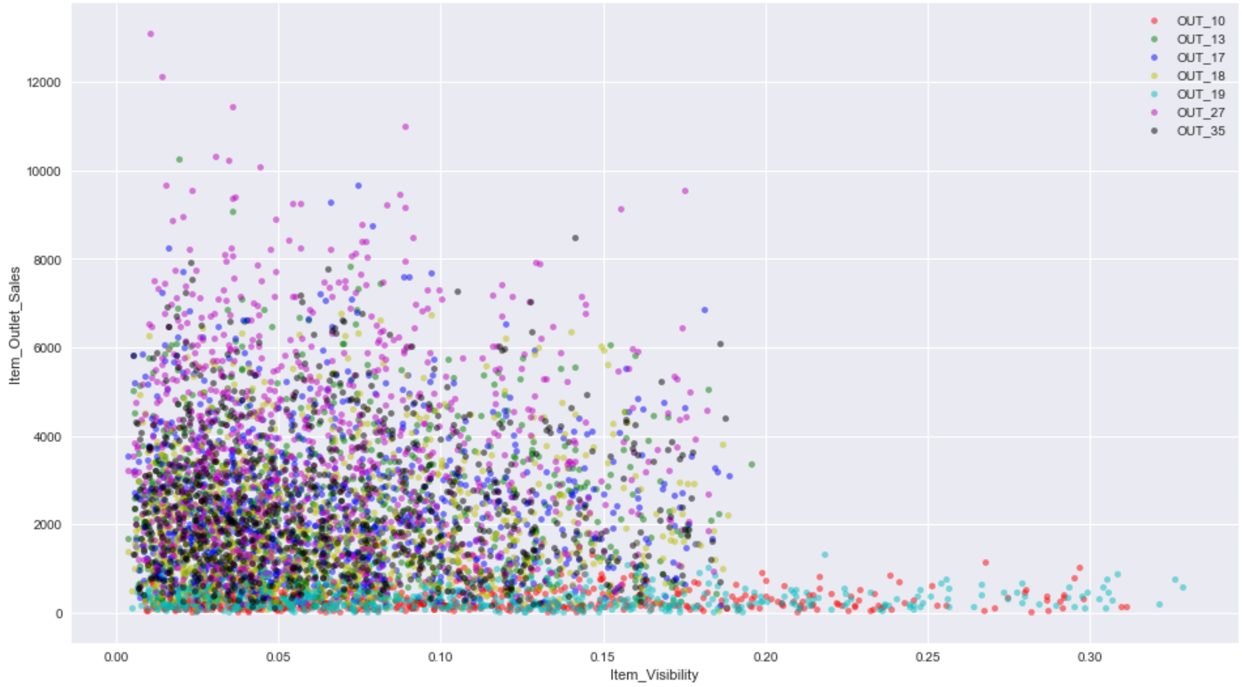
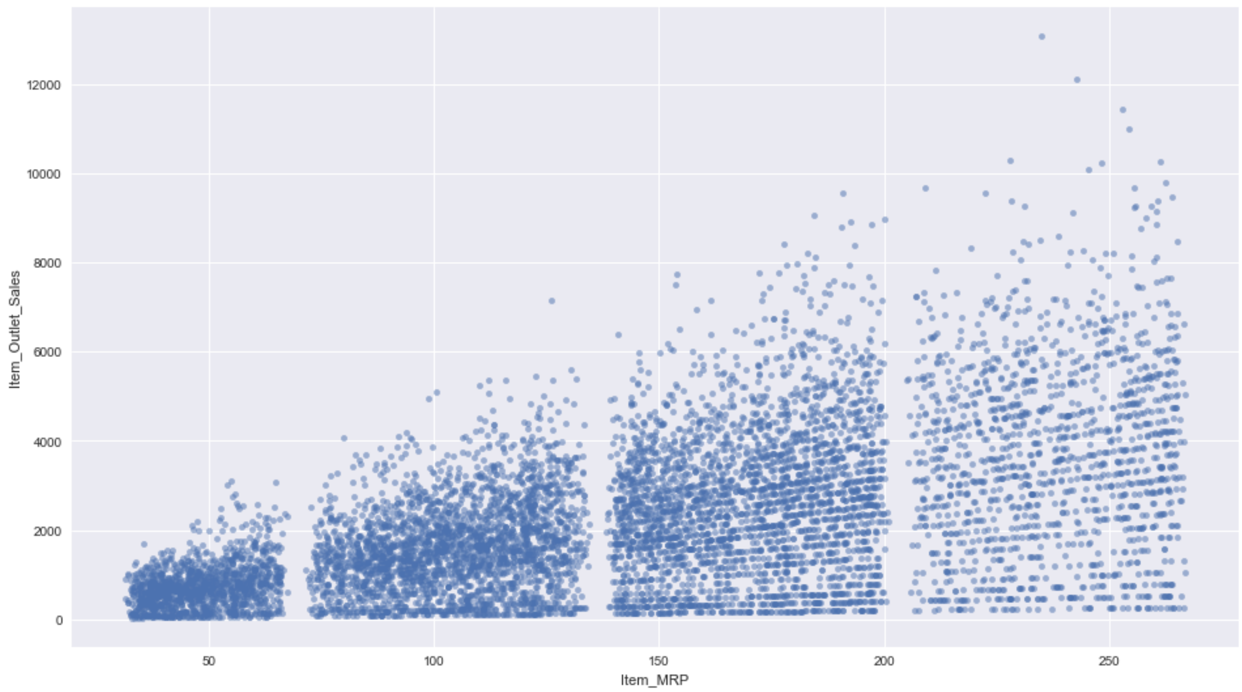


Figure 17. All store sales data vs Visibility of item

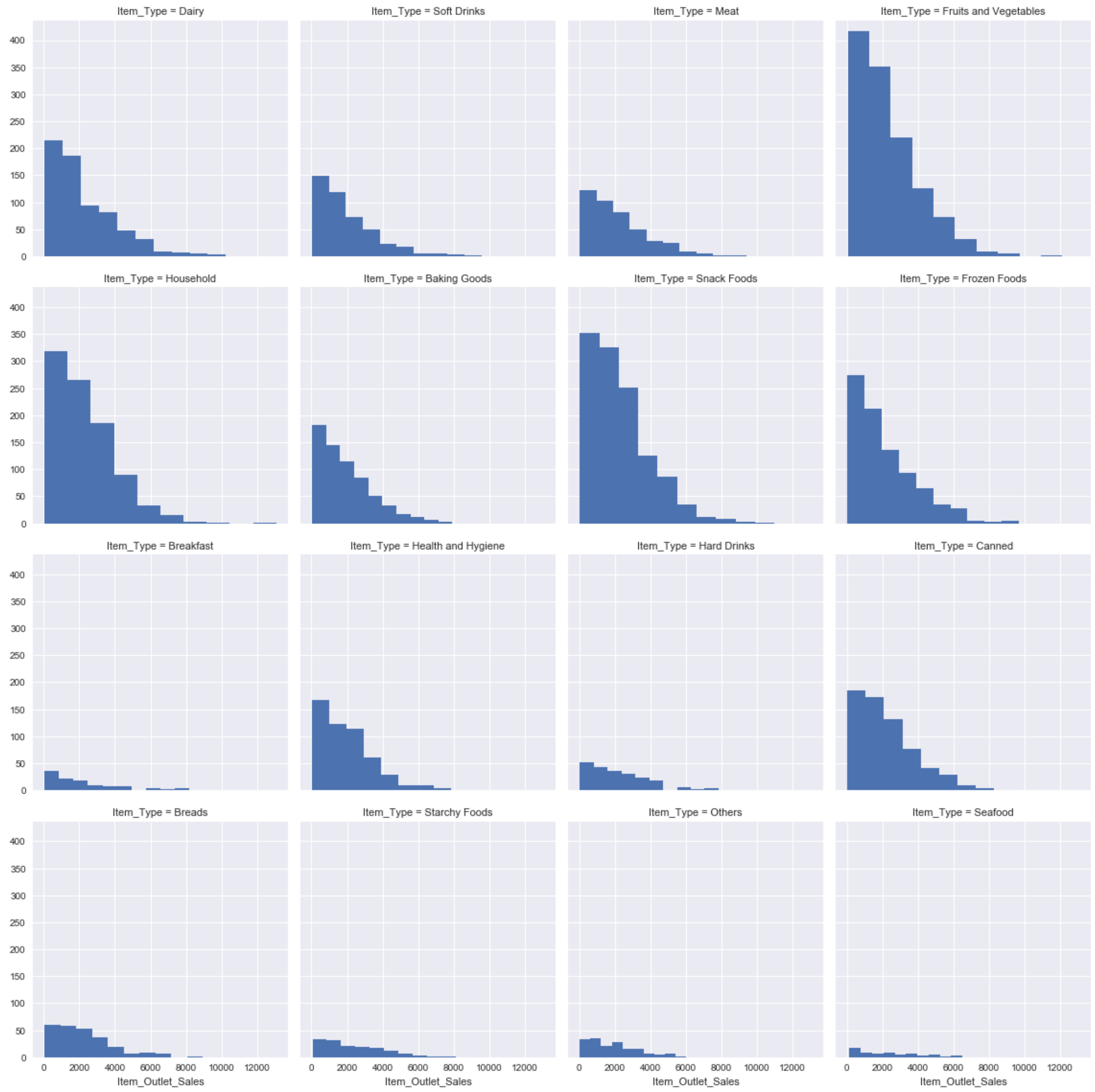
10.4. Analysis Of All Store Sales vs MRP:

This is an analysis plot of sales vs. price; here we can observe how sales are happening with respect to price. We can infer from this that sales of items are happening from $150 - $200 range. Here are we can see few outliers only for certain type of items on certain day and we can observe more sales of specific item, which we consider as outliers/noise. We can replace those values with mean to get the accurate model.



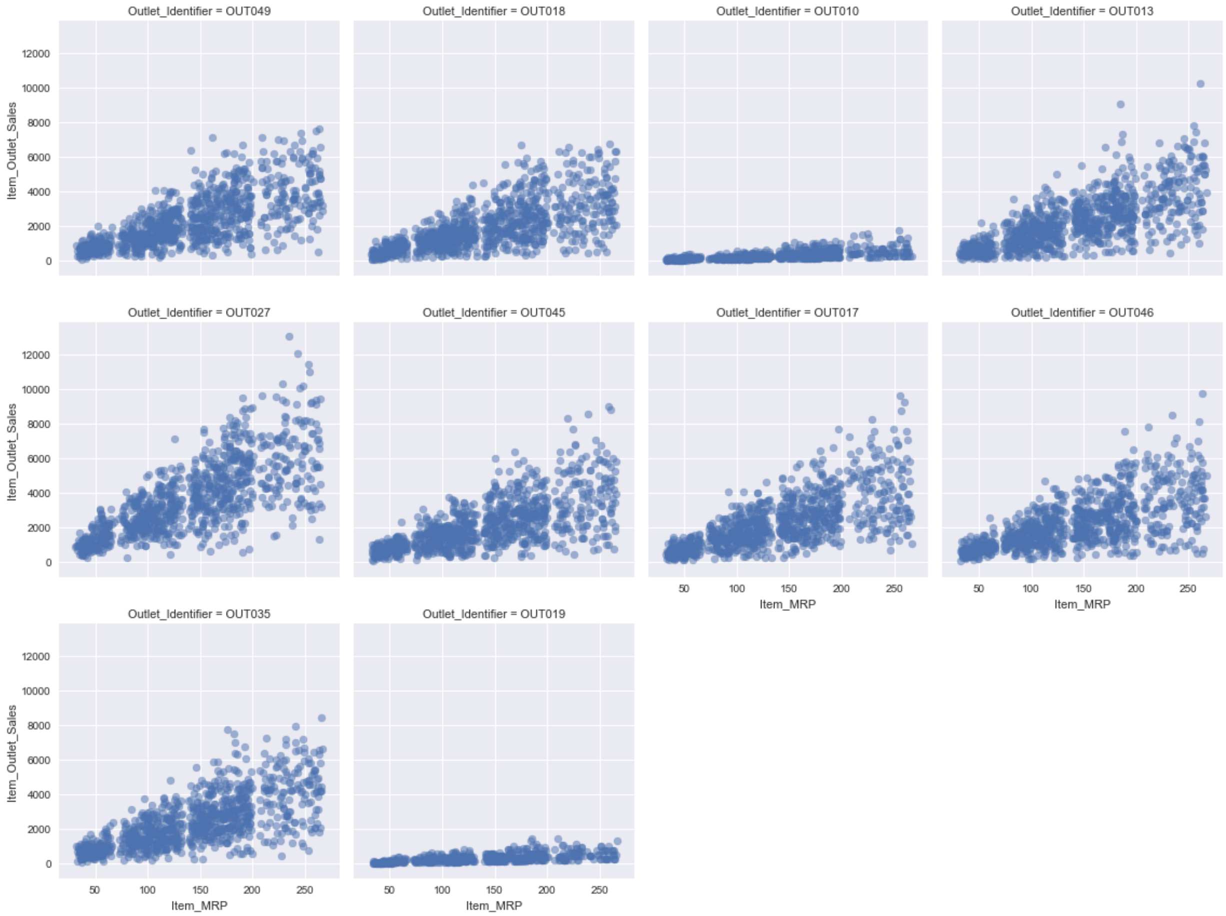
10.5. Analysis Of Sales vs Item Type:

In this we are analyzing the sales of a item from all the store with respect to Item type, which give an information about how sales are happening and which item type have more sales. Let say Item Type Dairy have sales up to 200 whereas Fruits and vegetables have more than 400. You can see least sales happening for sea foods with the visualization. This tells that the sea food items have less demand on the stores. This will give a good understanding of which item have more sales on the store.



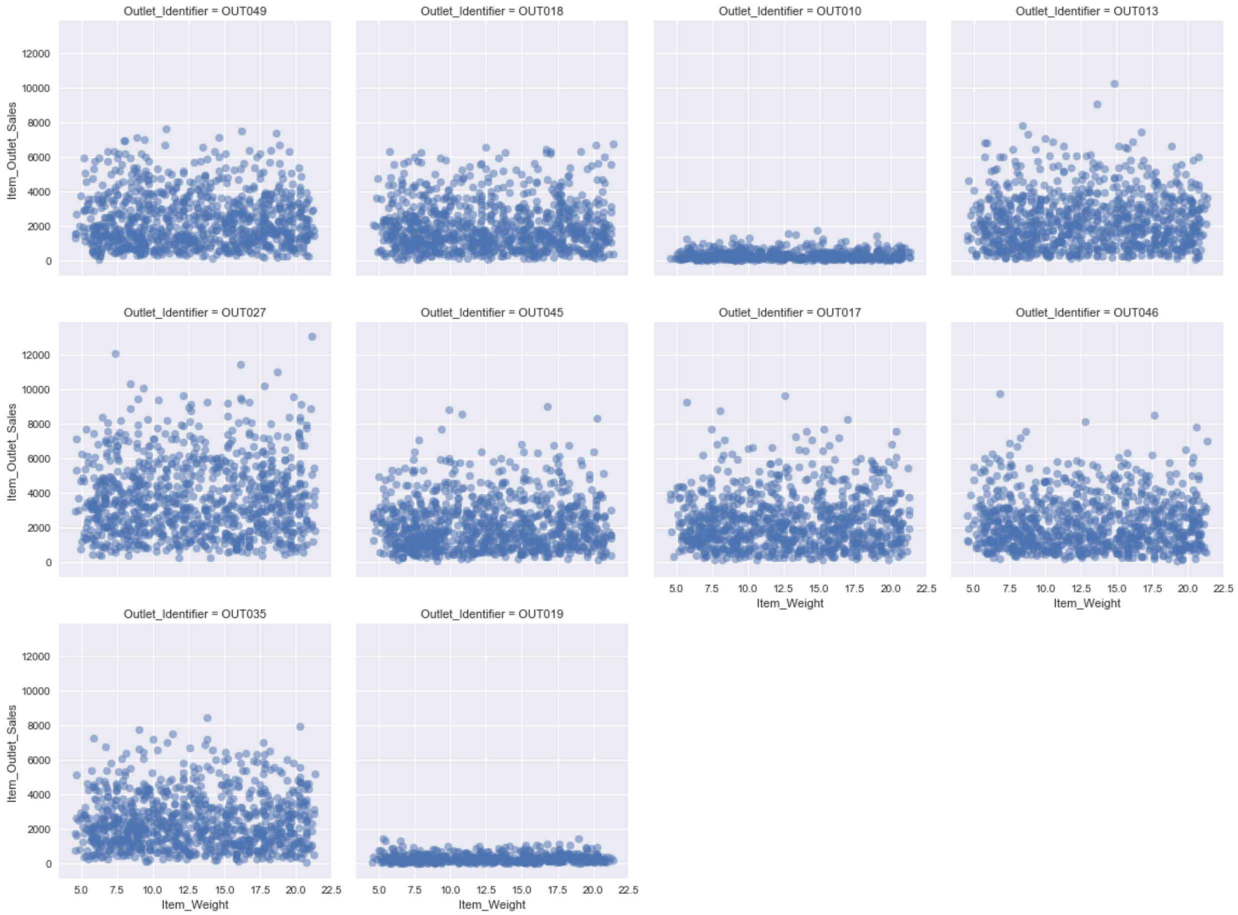
10.6. Analysis Of Outlet Sales vs MRP:

In this we can visualize the sales with respect to the MRP, we can infer some valuable information from the below plot. As we discussed above sales of all the stores with different colors, now we can see all the sales of the different outlet in single plot. Now we can clearly identify which store is performing more sales and which store is performing less sales in detailed in the below visualization plot. Outlet27 is performing more sales, Outlet10 and Outlet19 have vey low sales.



10.7. Analysis Of Outlet Sales vs Item Weight:

In this plot, we are analyzing the sales of the store with respect to the Item weight in that particular store. Based on these visualizations, we will clearly help in interpreting how sales are happening in the store and which range of weight have more sales.


Outlet_Identifier = OUT049
Outlet_Identifier = OUT018
Outlet_Identifier = OUT010
Outlet_Identifier = OUT013
Outlet_Identifier = OUT027
Outlet_Identifier = OUT045
Outlet_Identifier = OUT017
Outlet_Identifier = OUT046
Outlet_Identifier = OUT035
Outlet_Identifier = OUT019
Item_Outlet_Sales
0
2000
4000
6000
8000
10000
12000
Item_Weight
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5

11. Model Building:

11.1. Linear Regression

It is used for predictive analysis. It is a technique which explains the degree of relationship between two or more variables (multiple regression, in that case) using a best fit line / plane. Simple Linear Regression is used when we have, one independent variable and one dependent variable

Regression technique tries to fit a single line through a scatter plot (see below). The simplest form of regression with one dependent and one independent variable is defined by the formula:

$$Y = aX + b$$

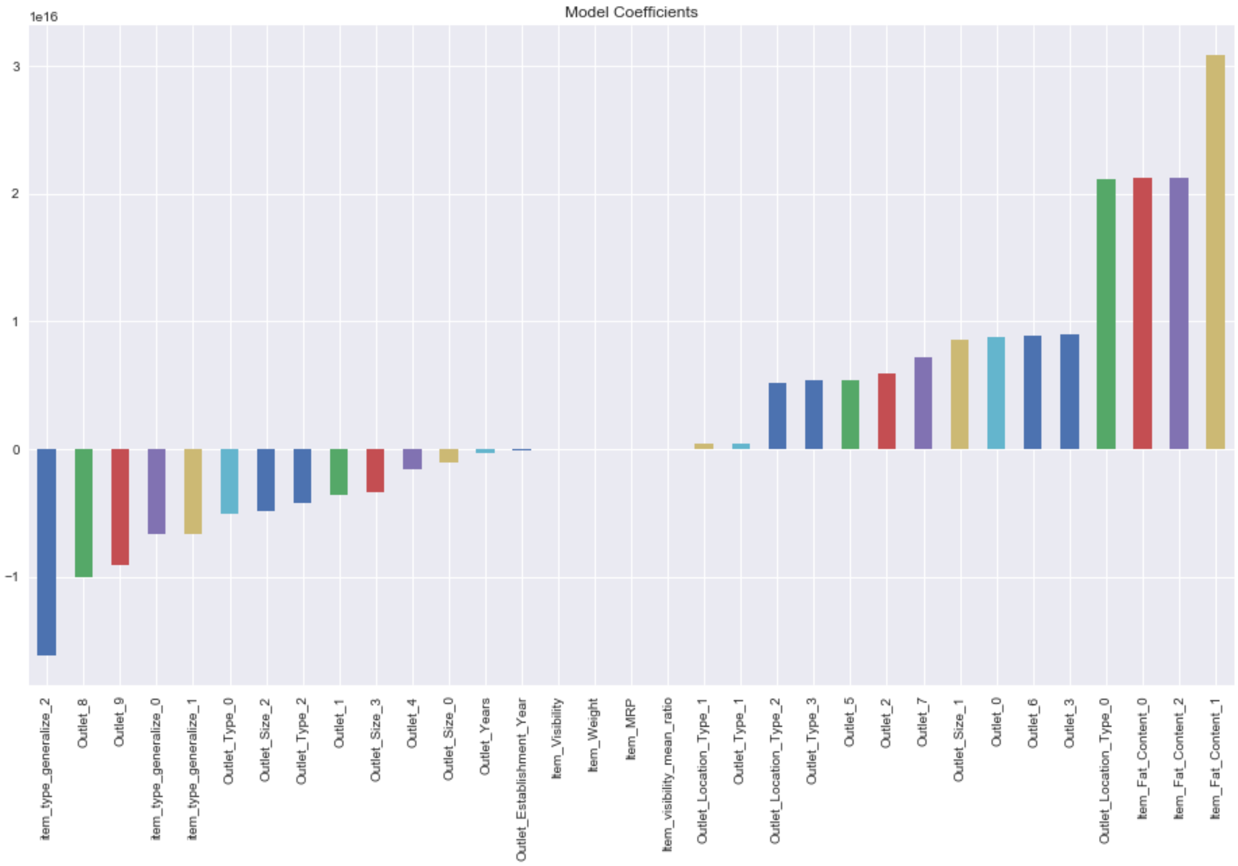
Model Report:

RMSE: 1128

CV Score:

Mean - 1129|STD - 43.88 | Min - 1075| Max - 1210

GRAPH:



11.2. Decision Tree Model

It allows us to develop classification systems that predict or classify future observations based on a set of decision rules. If you have data divided into classes that interest you (for example, high- versus low-risk loans, subscribers versus nonsubscribers, voters versus nonvoters, or types of bacteria), you can use your data to build rules that you can use to classify old or new cases with maximum accuracy. For example, you might build a tree that classifies credit risk or purchase intent based on age and other factors.

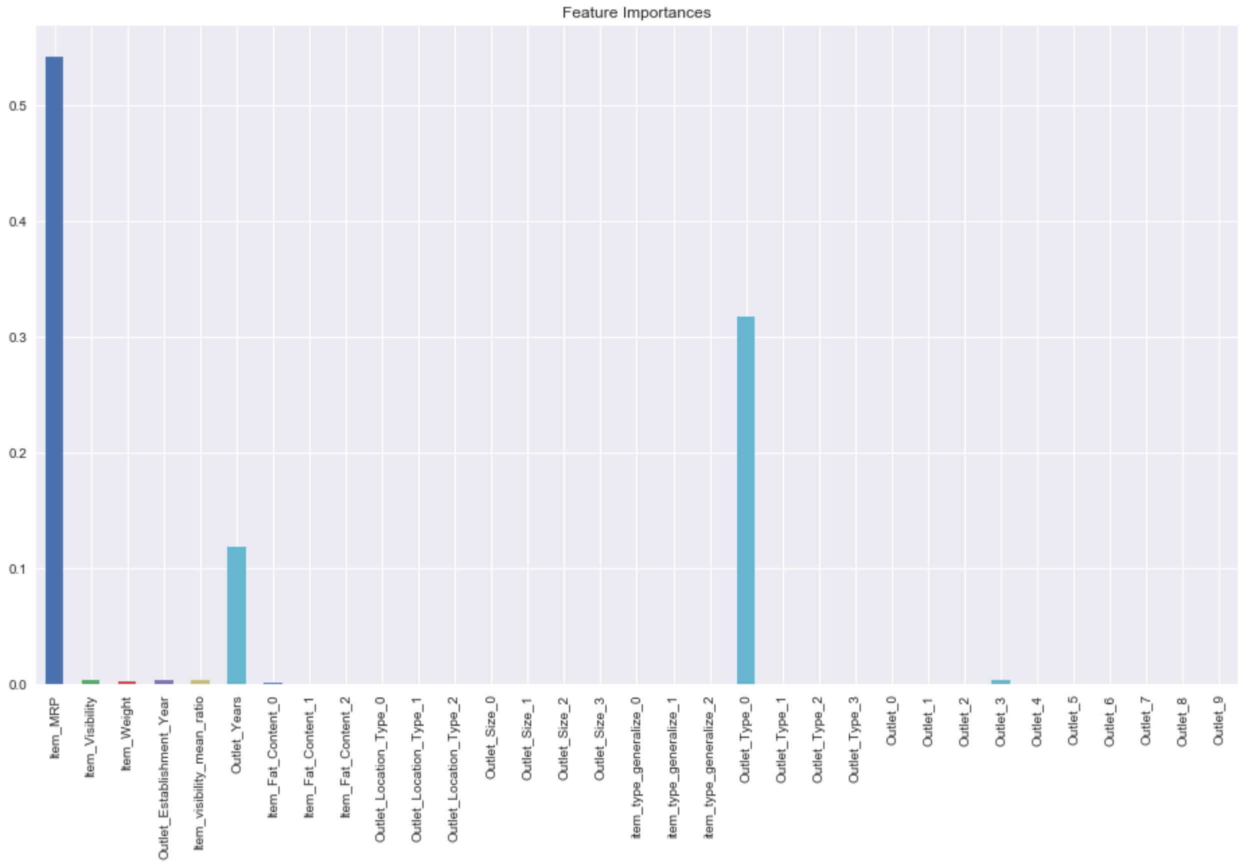
Decision trees is a machine learning technique that are used for classification and regression problems. The idea behind this algorithm goes in a top-down approach where you all the train cases at the node and then you split the tree in to branches until you the reach the leaf node. Decision tree uses Gini Index/Entropy to split the nodes. Gini Index measures the impurity of the attributes and choses the attributes which are the purest. The attribute with Gini score 0 is the purest.

Model Report:

RMSE: 1151

CV Score: Mean - 1250| STD - 43.05| Min - 1184| Max - 1344

GRAPH



11.3. Random Forest Model

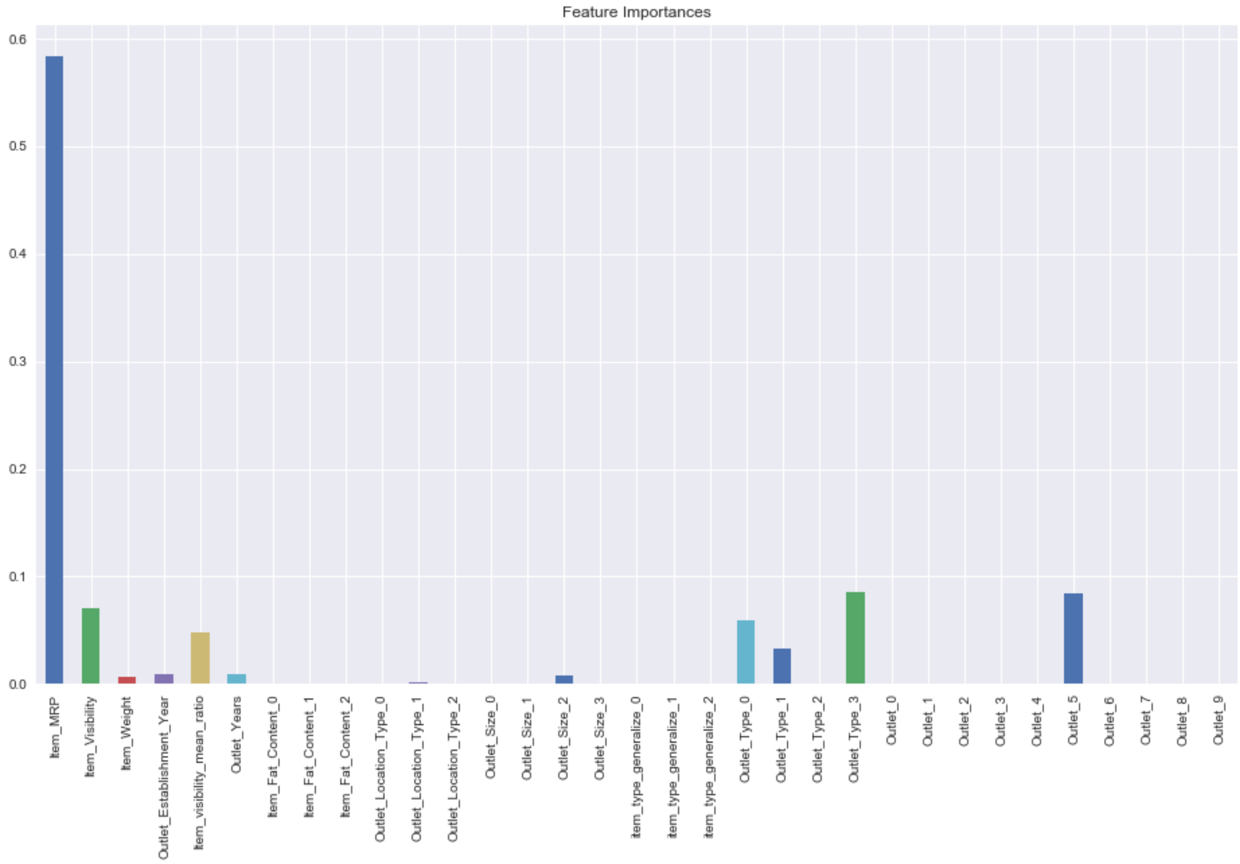
It is a popular ensemble learning method. As the name suggests it creates a forest of decision trees and out of those trees the one which has the highest majority is chosen as a final model which will be used for prediction. Random forest takes N attributes form the dataset and then it splits the data into edges, just like decision trees which uses Gini Index or Entropy to determine the split points Random Forest also considers those metrics to choose the best split point. It will create N number of trees with each tree is made on subset of data and in the end it calculates the votes that each tree has and chooses the one which has the majority votes.

Model Report:

RMSE: 1147

CV Score: Mean - 1247| STD - 45.92 | Min - 1164| Max - 1346

GRAPH



11.4. Ridge Regression

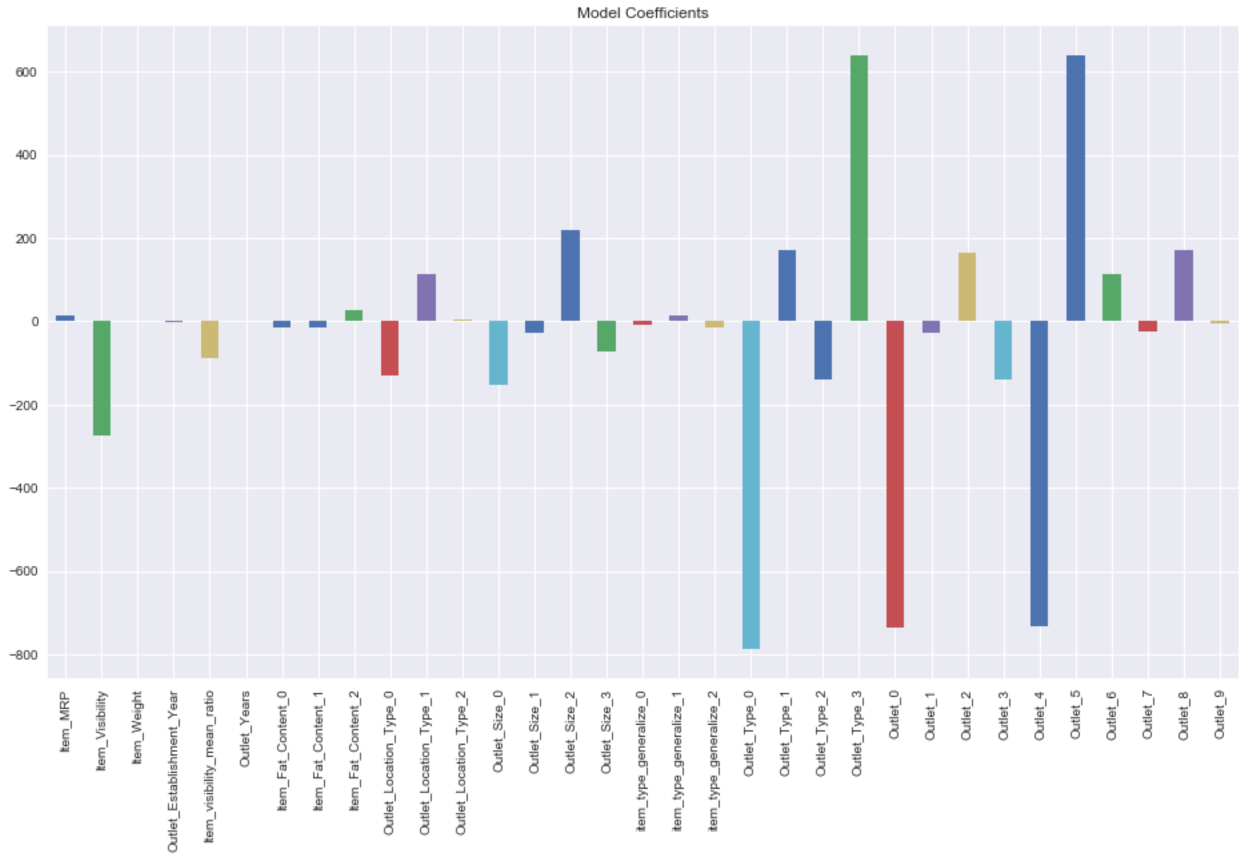
It is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable. Another biased regression technique, principal components regression, is also available in NCSS. Ridge regression is the more popular of the two methods.

Model Report :

RMSE : 1129

CV Score : Mean - 1130| STD - 44.63| Min - 1076| Max - 1217

GRAPH



11.5. AdaBoost Model

It is used with short decision trees. Further, the first tree is created, the performance of the tree on each training instance is used. Also, we use it to weight how much attention the next tree. Thus, it is created should pay attention to each training instance. Hence, training data that is hard to predict is given more weight. Although, whereas easy to predict instances are given less weight.

Each instance in the training dataset is weighted. The initial weight is set to:

Weight(xi) = 1/n

Where xi is the i'th training instance and n is the number of training instances

Model Report :

RMSE : 1147

CV Score : Mean - 1159| STD - 40.96| Min - 1085| Max - 1230

12. Evaluating Metrics:

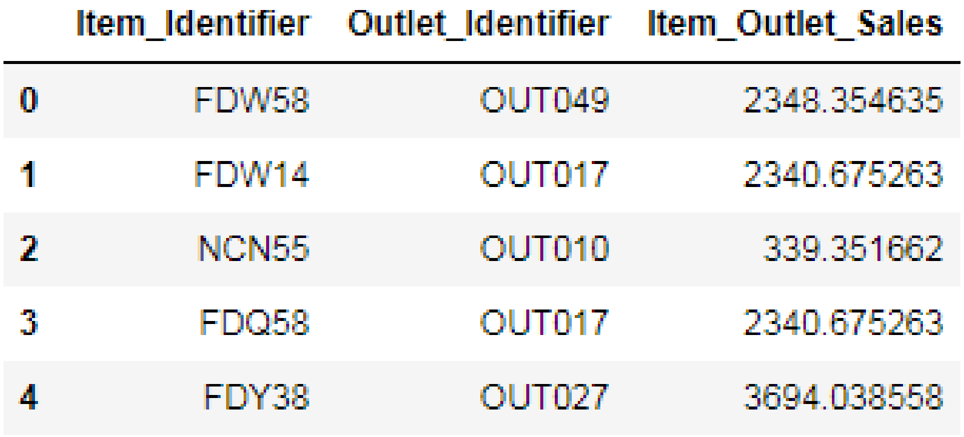
Model Name	RMSE	Cross Validation Score			
		Mean	Std	Min	Max
Linear Regression	1128	1129	43.88	1075	1210
Ridge Regression	1129	1130	44.63	1076	1217
AdaBoost Regressor	1147	1159	40.96	1085	1230
Decision Tree Regressor	1151	1250	43.05	1184	1344
Random Forest Regressor	1147	1247	45.92	1164	1346

13. Conclusion:

We observed that the Linear Regression algorithm performs better where the RMSE value is at the lowest compared to other Models.

Model Result:

RMSE: 1128



	Item_Identifier	Outlet_Identifier	Item_Outlet_Sales
0	FDW58	OUT049	2348.354635
1	FDW14	OUT017	2340.675263
2	NCN55	OUT010	339.351662
3	FDQ58	OUT017	2340.675263
4	FDY38	OUT027	3694.038558

14. References:

[1] Pandas (2018) Pandas Library. [Online]. URL: https://pandas.pydata.org/

[2] Matplot (2018) Matplotlib Library. [Online]. URL: https://matplotlib.org/

[3] Seaborn. (July 2018) statistical data visualization. [Online].
URL: https://seaborn.pydata.org/

[4] Numpy, Python Numpy Tutorial. [Online].
URL: http://cs231n.github.io/python-numpy-tutorial/

[5] Kaggle. (2018) Data Exploration and Price Prediction, House Sales. [Online].
URL: https://www.kaggle.com/fg1983/data-exploration-and-price-prediction-house-sales

[6] Columbia University, A Data-Cleaning Tool for Building Better Prediction Models. [Online].
URL: https://datascience.columbia.edu/data-cleaning-tool-building-better-prediction-models