Project Report

Department of Computer Science and Engineering

University at Buffalo, Buffalo, NY 14260

Team: Desireddy Sai Sankeerthana — saisanke@buffalo.edu

Vamshi Jamalpur — vamshija@buffalo.edu

Problem Statement :

Each and everyday thousands of products are sold daily in the stores. The retails owners purchase a wide range of products to be sold at their stores and they must be cost-effective so that they can maximize their profits. Purchasing unwanted items or items that are not in demand or buying items in bulk and not being able to sell them profitably would incur a huge loss to their stores. Which pushes the retailers to take wrong decisions and invest in more unwanted things which might lead the store to fall into debt. Hence a retailer must have a better understanding of his customers demands and of the products he is purchasing to increase the sales. To solve this problem, we have come up with a solution that will predict the sales of the products with the help of the factors that are affecting these sales which gives us a better idea on how to increase the profits.

POTENTIAL OF THE PROJECT: This prediction of sales is based on various factors such as the geographical location of the store and the type of products sold in the store etc, which helps the outlet owners to get a good analysis of the products that will increase the sales. This forecasting helps the business owners to know what kind of products are getting more sales based on their location. And helps new markets, outlets find out more about products that would help them earn more profits. It also helps different types of outlet owners to know what kind of products to be purchased based on their outlet size.

Data Source:

In this project, we will be analysing Big Mart sales to find out sales prediction of each product at particular outlets and address the prediction of sales of respective products available in particular outlets

We have collected dataset from Kaggle. Reference for data source https://www.kaggle.com/datasets/shivan118/big-mart-sales-prediction-datasets

DATA CLEANING/PREPROCESSING

- Step-1 Loading the dataset into our environment.

```python
df = pd.read_csv('Sales.csv')

df.head()
```

[299] ✓ 0.8s Python

	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_S
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3	Grocery Store	732.
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.

- Step-2 Now we will be Checking the type of data present in the file like whether it is numerical data or object data etc.

```python
df.info()
```

[300] ✓ 0.3s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                7060 non-null   float64
 2   Item_Fat_Content           8523 non-null   object
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object
 7   Outlet_Establishment_Year  8523 non-null   int64
 8   Outlet_Size                6113 non-null   object
 9   Outlet_Location_Type       8523 non-null   object
 10  Outlet_Type                8523 non-null   object
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

- Step-3 Drop the duplicate values present in our data.

```python
df.drop_duplicates()
```

[429] ✓ 0.2s Python

	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_Outle
0	FDA15	9.30	Low Fat	0.016	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	37
1	DRC01	5.92	Regular	0.019	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	4
2	FDN15	17.50	Low Fat	0.017	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	20
3	FDX07	19.20	Regular	0.000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	7
4	NCD19	8.93	Low Fat	0.000	Household	53.8614	OUT013	1987	Large	Tier 3	Supermarket Type1	9
...	...	...	...	...	...	...	...	...	...	...	...	
8518	FDF22	6.87	Low Fat	0.057	Snack Foods	214.5218	OUT013	1987	Large	Tier 3	Supermarket Type1	27
8519	FDS36	8.38	Regular	0.047	Baking Goods	108.1570	OUT045	2002	Small	Tier 2	Supermarket Type1	5
8520	NCJ29	10.60	Low Fat	0.035	Health and Hygiene	85.1224	OUT035	2004	Small	Tier 2	Supermarket Type1	1
8521	FDN46	7.21	Regular	0.145	Snack Foods	103.1332	OUT018	2009	Medium	Tier 3	Supermarket Type2	18
8522	DRG01	14.80	Low Fat	0.045	Soft Drinks	75.4670	OUT046	1997	Small	Tier 1	Supermarket Type1	7

8523 rows × 12 columns

- Step-4 We have to Check whether there are any missing or null values present in our data to avoid any inconsistencies and loss of data

```python
df.isna().sum()
```

```
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
```

- Step-5 Now we can observe that there are two columns which have null values present.


```python
df.isna().sum()
```

```
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
```


- Step-6: Replacing the null values for the column Item Weight with the mean of that column . This method is useful when we have numeric data.We will Check whether the null values are filled with the mean or not.

```python
df['Item_Weight'].fillna(df['Item_Weight'].mean(),inplace=True)
df.info()
```

[304] ✓ 0.2s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                8523 non-null   float64
 2   Item_Fat_Content           8523 non-null   object
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object
 7   Outlet_Establishment_Year  8523 non-null   int64
 8   Outlet_Size                6113 non-null   object
 9   Outlet_Location_Type       8523 non-null   object
 10  Outlet_Type                8523 non-null   object
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

- Step-7 Now we have the column Outlet_Size as we can observe there is a relation between the Outlet size and the outlet type ,so we cannot directly take the mode of the all outlet size, based on which outlet type has more occurrence we will take the mode.

```python
df.head(10)
```

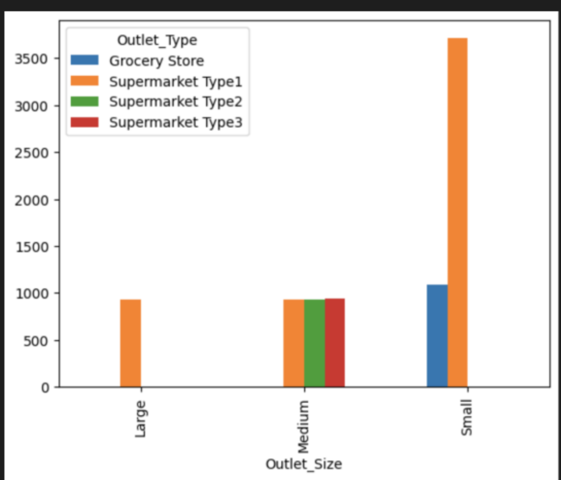
[440] ✓ 0.6s Python

	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_S
0	FDA15	9.300000	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.
1	DRC01	5.920000	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.
2	FDN15	17.500000	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.
3	FDX07	19.200000	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3	Grocery Store	732.
4	NCD19	8.930000	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.
5	FDP36	10.395000	Regular	0.000000	Baking Goods	51.4008	OUT018	2009	Medium	Tier 3	Supermarket Type2	556.
6	FDO10	13.650000	Regular	0.012741	Snack Foods	57.6588	OUT013	1987	High	Tier 3	Supermarket Type1	343.
7	FDP10	12.857645	Low Fat	0.127470	Snack Foods	107.7622	OUT027	1985	Medium	Tier 3	Supermarket Type3	4022.
8	FDH17	16.200000	Regular	0.016687	Frozen Foods	96.9726	OUT045	2002	NaN	Tier 2	Supermarket Type1	1076.
9	FDU28	19.200000	Regular	0.094450	Frozen Foods	187.8214	OUT017	2007	NaN	Tier 2	Supermarket Type1	4710.

```python
ef=df.groupby(['Outlet_Size','Outlet_Type']).size()
ef=ef.unstack()
ef.plot(kind='bar')
```

[431] ✓ 0.3s

```
<AxesSubplot: xlabel='Outlet_Size'>
```



After taking mode and replacing them:

	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_
0	FDA15	9.30	Low Fat	0.016	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735
1	DRC01	5.92	Regular	0.019	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443
2	FDN15	17.50	Low Fat	0.017	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097
3	FDX07	19.20	Regular	0.000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	732
4	NCD19	8.93	Low Fat	0.000	Household	53.8614	OUT013	1987	Large	Tier 3	Supermarket Type1	994
...	...	...	...	...	...	...	...	...	...	...	...	
72	FDH35	18.25	Low Fat	0.000	Starchy Foods	164.7526	OUT045	2002	Small	Tier 2	Supermarket Type1	4604
73	FDG02	7.86	Low Fat	0.011	Canned	189.6188	OUT017	2007	Small	Tier 2	Supermarket Type1	2285
74	NCZ18	7.83	Low Fat	0.186	Household	254.3698	OUT049	1999	Medium	Tier 1	Supermarket Type1	5580
75	FDC29	8.39	Regular	0.024	Frozen Foods	114.0176	OUT046	1997	Small	Tier 1	Supermarket Type1	2290
76	FDQ10	12.85	Low Fat	0.033	Snack Foods	172.3422	OUT049	1999	Medium	Tier 1	Supermarket Type1	1207

- Step-8 : In the Item_Fat_Content attribute we have the four variables with Low Fat, LF, lf, reg,Regular. Replace corresponding names with the Low Fat and Regular

	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_
0	FDA15	9.30	Low Fat	0.016	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735
1	DRC01	5.92	Regular	0.019	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443
2	FDN15	17.50	Low Fat	0.017	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097
3	FDX07	19.20	Regular	0.000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	732
4	NCD19	8.93	Low Fat	0.000	Household	53.8614	OUT013	1987	Large	Tier 3	Supermarket Type1	994
...	...	...	...	...	...	...	...	...	...	...	...	
72	FDH35	18.25	Low Fat	0.000	Starchy Foods	164.7526	OUT045	2002	Small	Tier 2	Supermarket Type1	4604
73	FDG02	7.86	Low Fat	0.011	Canned	189.6188	OUT017	2007	Small	Tier 2	Supermarket Type1	2285
74	NCZ18	7.83	Low Fat	0.186	Household	254.3698	OUT049	1999	Medium	Tier 1	Supermarket Type1	5580
75	FDC29	8.39	Regular	0.024	Frozen Foods	114.0176	OUT046	1997	Small	Tier 1	Supermarket Type1	2290
76	FDQ10	12.85	Low Fat	0.033	Snack Foods	172.3422	OUT049	1999	Medium	Tier 1	Supermarket Type1	1207

- Step-9 : In the Outlet_Size we have high, Medium, small. Changing the size from high To Large

	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_
0	FDA15	9.30	Low Fat	0.016	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735
1	DRC01	5.92	Regular	0.019	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443
2	FDN15	17.50	Low Fat	0.017	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097
3	FDX07	19.20	Regular	0.000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	732
4	NCD19	8.93	Low Fat	0.000	Household	53.8614	OUT013	1987	Large	Tier 3	Supermarket Type1	994
...	...	...	...	...	...	...	...	...	...	...	...	
72	FDH35	18.25	Low Fat	0.000	Starchy Foods	164.7526	OUT045	2002	Small	Tier 2	Supermarket Type1	4604
73	FDG02	7.86	Low Fat	0.011	Canned	189.6188	OUT017	2007	Small	Tier 2	Supermarket Type1	2285
74	NCZ18	7.83	Low Fat	0.186	Household	254.3698	OUT049	1999	Medium	Tier 1	Supermarket Type1	5580
75	FDC29	8.39	Regular	0.024	Frozen Foods	114.0176	OUT046	1997	Small	Tier 1	Supermarket Type1	2290
76	FDQ10	12.85	Low Fat	0.033	Snack Foods	172.3422	OUT049	1999	Medium	Tier 1	Supermarket Type1	1207

- Step-10: Changing the respective object type to Category type

```
df.info()
[310] ✓ 0.2s                                                        Python

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                8523 non-null   float64
 2   Item_Fat_Content           8523 non-null   category
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   category
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   category
 7   Outlet_Establishment_Year  8523 non-null   int64
 8   Outlet_Size                8523 non-null   category
 9   Outlet_Location_Type       8523 non-null   category
 10  Outlet_Type                8523 non-null   category
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: category(6), float64(4), int64(1), object(1)
memory usage: 451.2+ KB
```

- Step-11: Rounding the Item Weight to 2 decimal values

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
FDA15	9.30	Low Fat	0.016	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
DRC01	5.92	Regular	0.019	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
FDN15	17.50	Low Fat	0.017	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
FDX07	19.20	Regular	0.000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	732.3800
NCD19	8.93	Low Fat	0.000	Household	53.8614	OUT013	1987	Large	Tier 3	Supermarket Type1	994.7052

- Step-12 :Rounding the Item visibility to 3 decimal values

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
FDA15	9.30	Low Fat	0.016	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
DRC01	5.92	Regular	0.019	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
FDN15	17.50	Low Fat	0.017	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
FDX07	19.20	Regular	0.000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	732.3800
NCD19	8.93	Low Fat	0.000	Household	53.8614	OUT013	1987	Large	Tier 3	Supermarket Type1	994.7052

- Step- 13: Changing the categorical values to numerical values

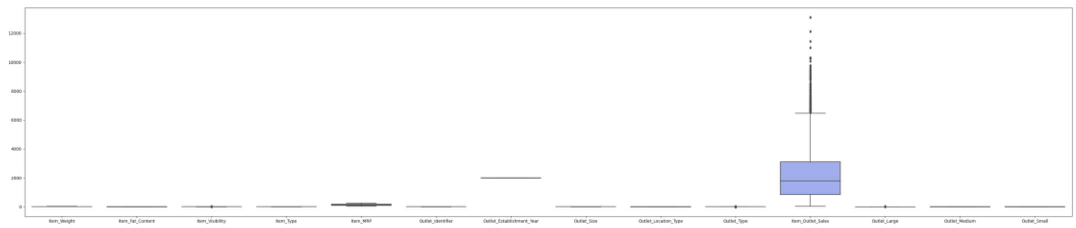
	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Ye
0	FDA15	9.30	0	0.016	4	249.8092	9	199
1	DRC01	5.92	1	0.019	14	48.2692	3	200
2	FDN15	17.50	0	0.017	10	141.6180	9	199
3	FDX07	19.20	1	0.000	6	182.0950	0	199
4	NCD19	8.93	0	0.000	9	53.8614	1	198
...	...	...	...	...	...	...	...	
8518	FDF22	6.87	0	0.057	13	214.5218	1	198
8519	FDS36	8.38	1	0.047	0	108.1570	7	200
8520	NCJ29	10.60	0	0.035	8	85.1224	6	200
8521	FDN46	7.21	1	0.145	13	103.1332	3	200
8522	DRG01	14.80	0	0.045	14	75.4670	8	199

8523 rows × 12 columns

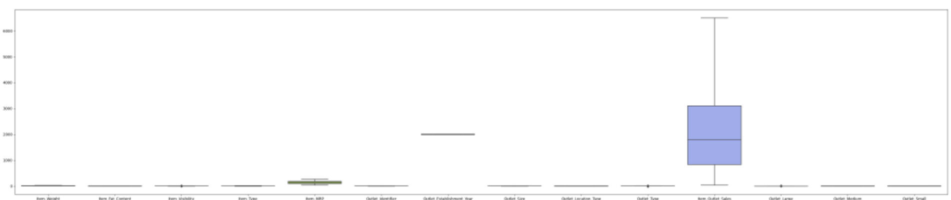
- Step-14: One hot encoding to Outlet Size as outlet size depends on the outlet_type so we are encoding the outlet size.

utlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales	Outlet_Large	Outlet_Medium	Outlet_Small
1999	1	0	1	3735.1380	0.0	1.0	0.0
2009	1	2	2	443.4228	0.0	1.0	0.0
1999	1	0	1	2097.2700	0.0	1.0	0.0
1998	2	2	0	732.3800	0.0	0.0	1.0
1987	0	2	1	994.7052	1.0	0.0	0.0
...	...	...	...	...	...	...	...
1987	0	2	1	2778.3834	1.0	0.0	0.0
2002	2	1	1	549.2850	0.0	0.0	1.0
2004	2	1	1	1193.1136	0.0	0.0	1.0
2009	1	2	2	1845.5976	0.0	1.0	0.0
1997	2	0	1	765.6700	0.0	0.0	1.0

- Step-15 :Check if there are any outliers ,if found using the IQR change it to the range.



After removing outliers



- Step-16: Normalising the non numerical variables

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Ye
0	FDA15	0.282738	0	0.048780	4	0.927507	9	0.58333
1	DRC01	0.081548	1	0.057927	14	0.072068	3	1.00000
2	FDN15	0.770833	0	0.051829	10	0.468288	9	0.58333
3	FDX07	0.872024	1	0.000000	6	0.640093	0	0.54166
4	NCD19	0.260714	0	0.000000	9	0.095805	1	0.08333
...	...	...	...	...	...	...	...	
8518	FDF22	0.138095	0	0.173780	13	0.777729	1	0.08333
8519	FDS36	0.227976	1	0.143293	0	0.326263	7	0.70833
8520	NCJ29	0.360119	0	0.106707	8	0.228492	6	0.79166
8521	FDN46	0.158333	1	0.442073	13	0.304939	3	1.00000
8522	DRG01	0.610119	0	0.137195	14	0.187510	8	0.50000

8523 rows × 15 columns

EDA(EXPLORATORY DATA ANALYSIS):

- Step-1: Describe the data

```python
df.describe()
```

✓ 0.3s

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	8523.000000	8523.000000	8523.000000	8523.000000	8523.000000
mean	12.858153	0.066133	140.992782	1997.831867	2181.288914
std	4.225989	0.051588	62.275067	8.371760	1706.499616
min	4.550000	0.000000	31.290000	1985.000000	33.290000
25%	9.310000	0.027000	93.826500	1987.000000	834.247400
50%	12.860000	0.054000	143.012800	1999.000000	1794.331000
75%	16.000000	0.095000	185.643700	2004.000000	3101.296400
max	21.350000	0.328000	266.888400	2009.000000	13086.964800

- Step-2: Check if there are any null values

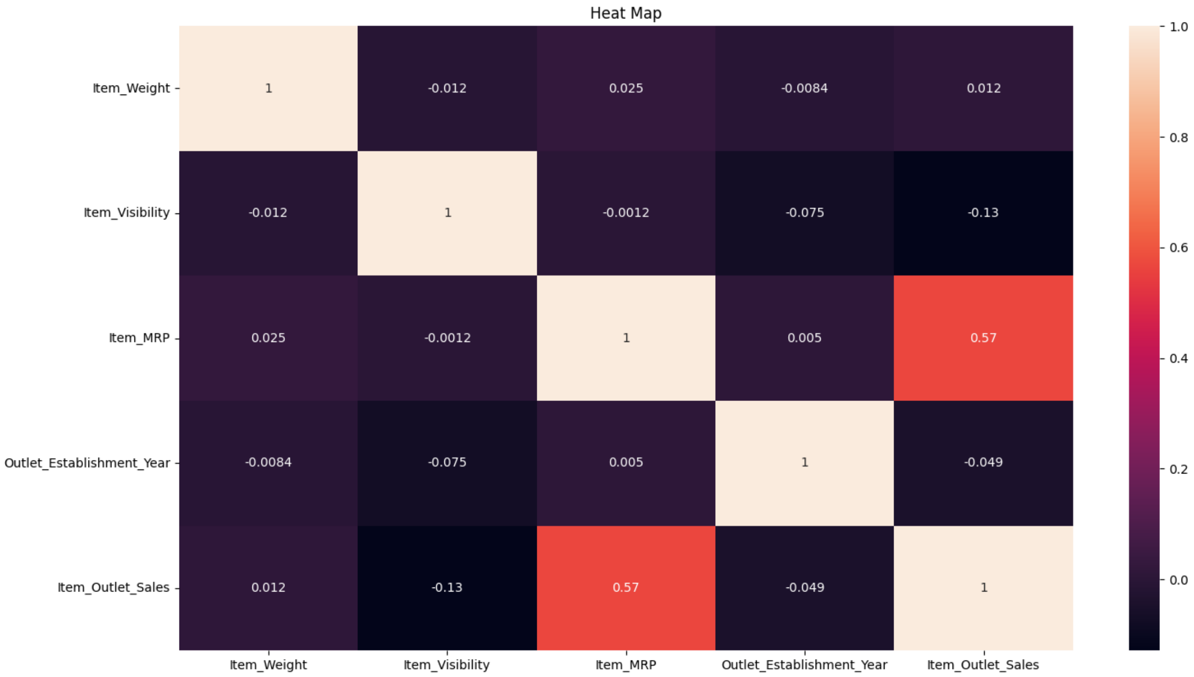
```python
df.isna()
```

✓ 0.3s Python

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Ye
0	False	False	False	False	False	False	False	Fal
1	False	False	False	False	False	False	False	Fal
2	False	False	False	False	False	False	False	Fal
3	False	False	False	False	False	False	False	Fal
4	False	False	False	False	False	False	False	Fal
...	...	...	...	...	...	...	...	
8518	False	False	False	False	False	False	False	Fal
8519	False	False	False	False	False	False	False	Fal
8520	False	False	False	False	False	False	False	Fal
8521	False	False	False	False	False	False	False	Fal
8522	False	False	False	False	False	False	False	Fal

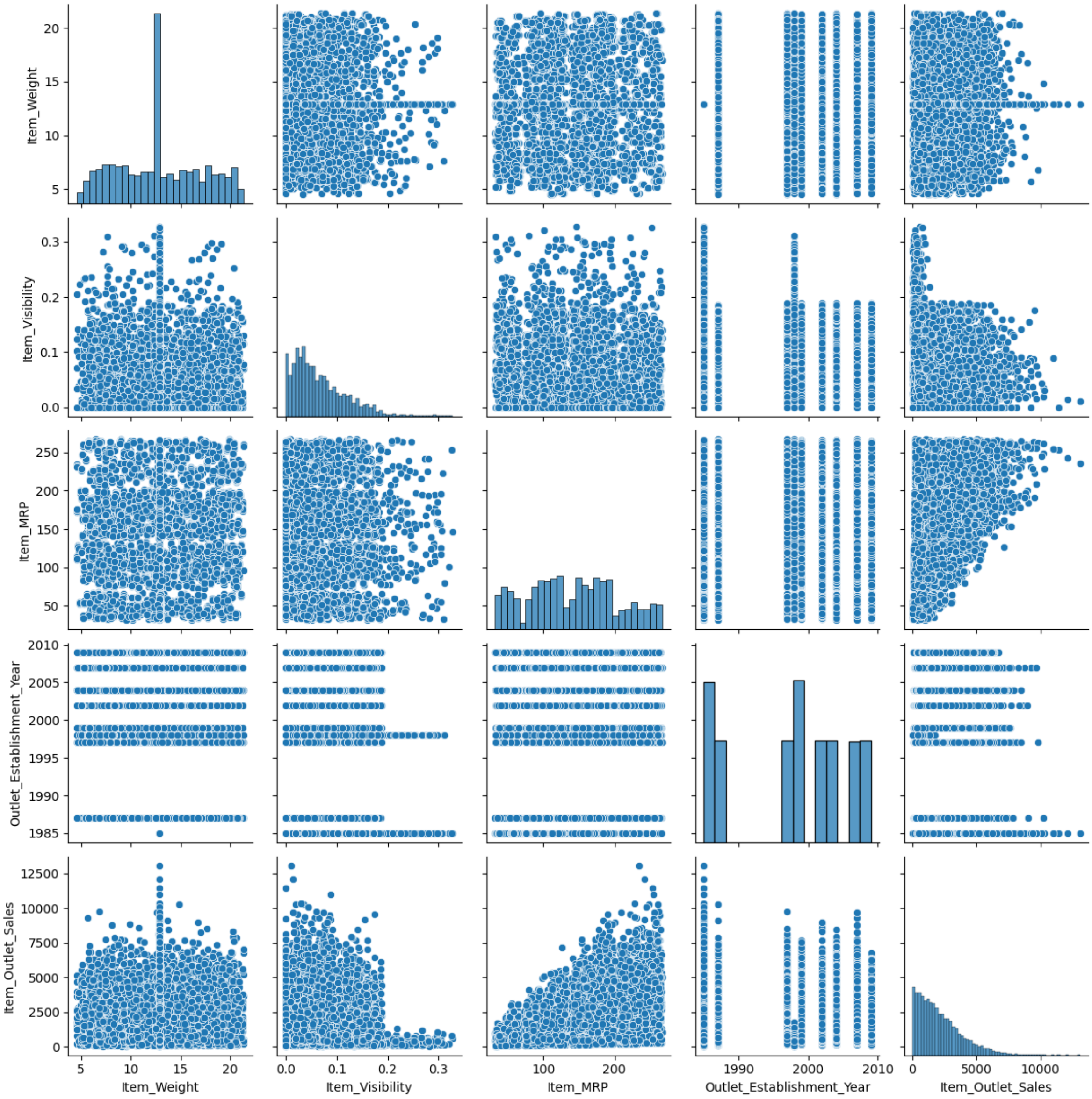
8523 rows × 12 columns

- Step-3: Check the co-relation between the numerical data



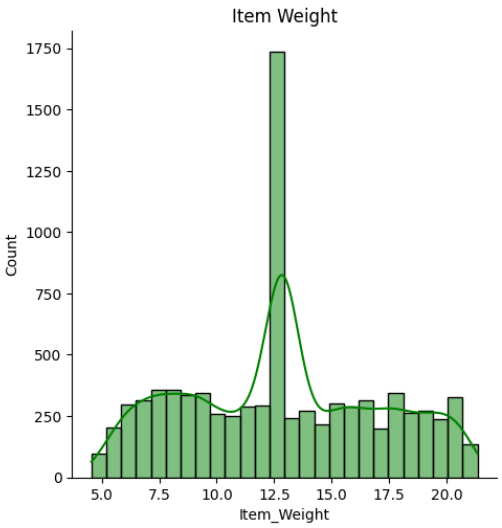
As we can see the co-relation between the numerical data.the negative integers represents there is less/no relation between the features, positive integers represents there is good/high relation between the features that can impact the other one.As we can see there is strong correlation between the mrp and the item sales.

Step:4 Plot all the numerical attributes.(pair plot)



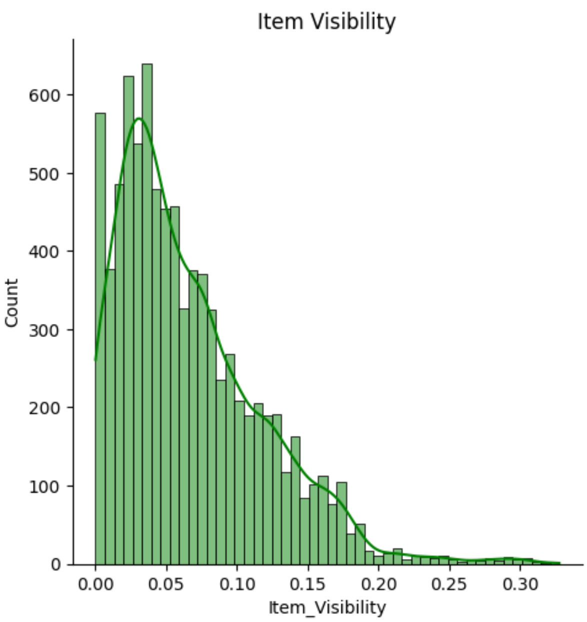
From the given graph we can see Item_outlet goes on increasing with Item_mrp. And Item_weight has similar correlation with Item_outlet_sales

UNIVARIATE ANALYSIS:



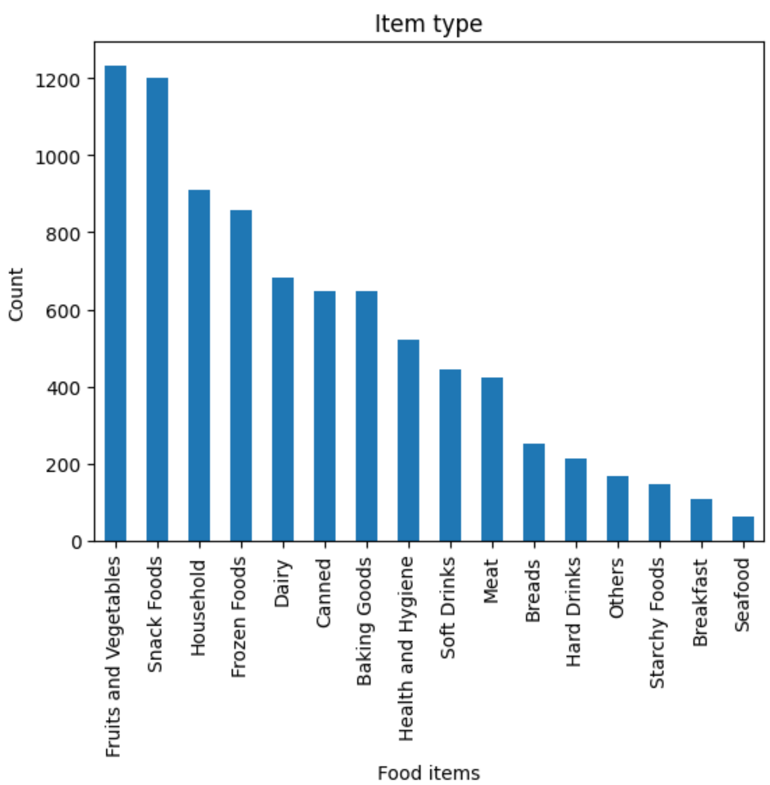
Observation:

In the given bar graph, all the weights lie in between the 5 and 20 and most of them are lying in the range of 12.5



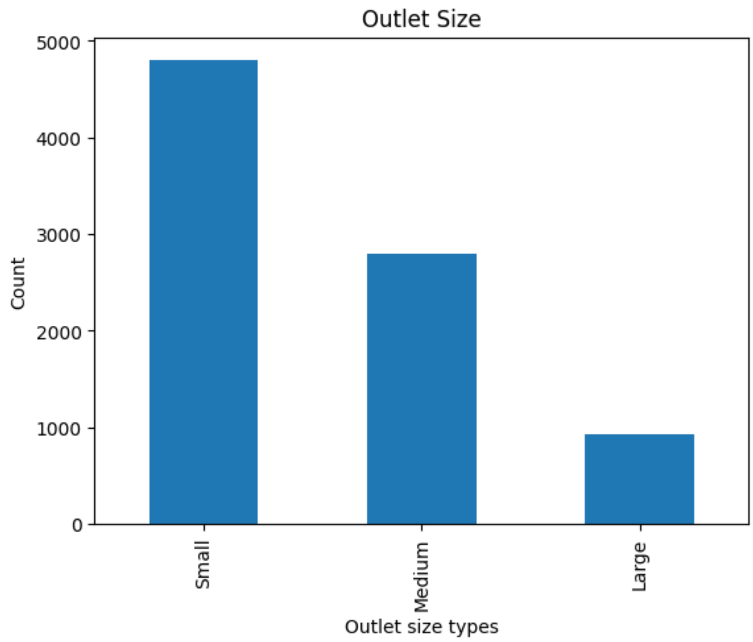
Observation:

In the given bar graph, all the items visibility lie in between the 0 and 0.30 and most of them are lying in the range 0.20-0.50 .It states us that most of the products are less visible in a big mart.



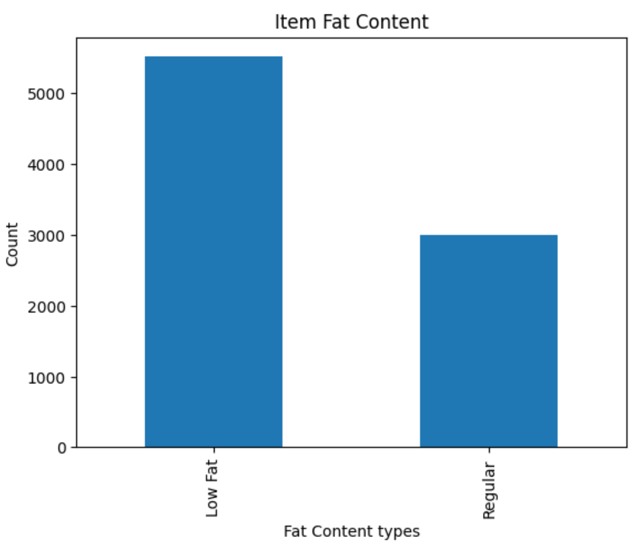
Observation:

By analysing the graph ,most of products are fruits and vegetables which consists of near to 14 % of whole data and sea food is the least selling in the count.

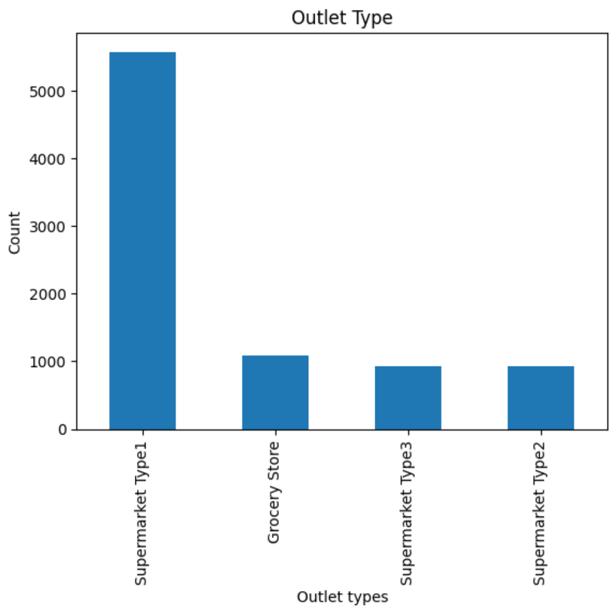


Observation:

By analysing the graph, most of them are the small stores which is nearly (4800), 56% . And the least from the large stores which is (<900) or 9%.

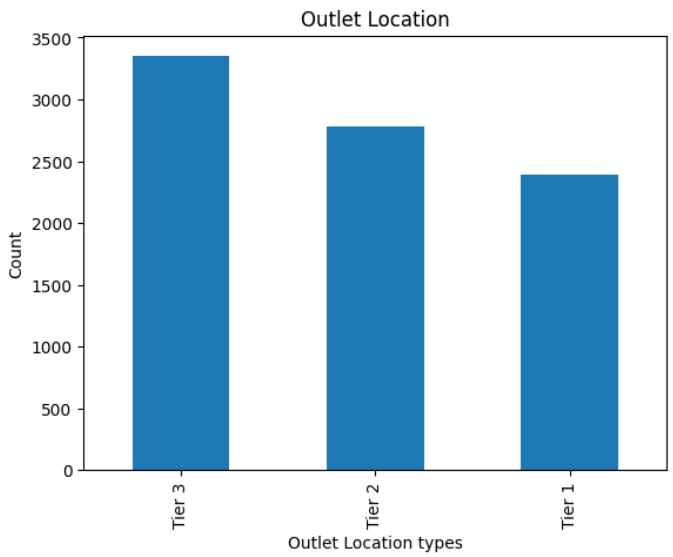


Observation:By analysing the graph, most of the products are in Low fat which is nearly (>5000) 58% and 35% of the products are low fat.



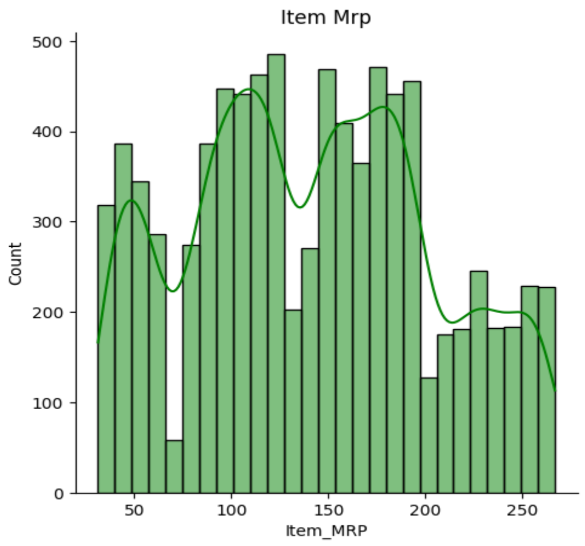
Observation:

By analysing the graph, most of the outlets are from the supermarket type 1 with more than 62%(>5000) , and the least from the supermarket type 2 with 9%.



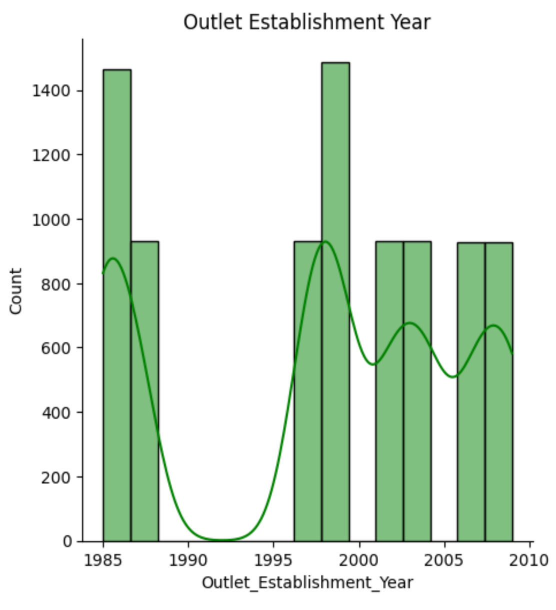
Observation:

Most of them come from the tier-3 location which consists of 39% of them and less number of sells came from the tier-1 which is 25%.



Observation:

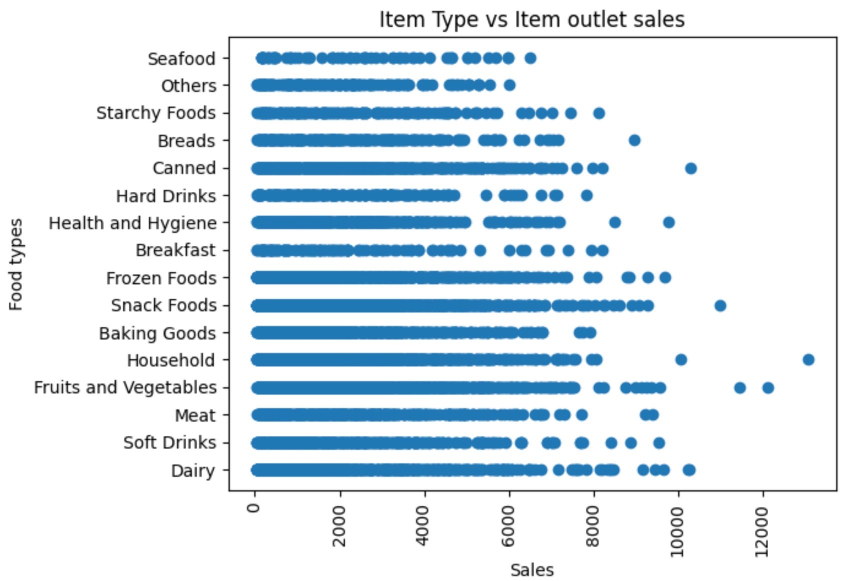
Most of them are in the range of 100-200, and less number of products in the range of 70.and most of the products have the price in range of 100-20



Observation:

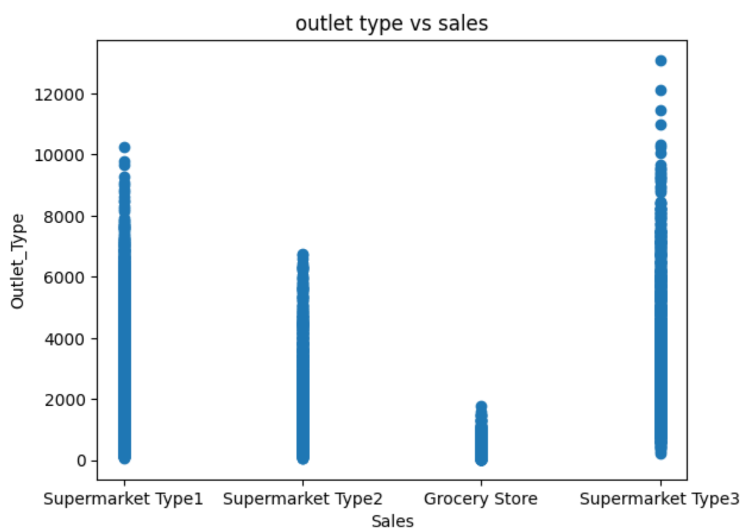
Most of outlets are established in the year of 1995-200 and 1985-1990.and no outlets are established between 1990-1995.

<u>BIVARIATE ANALYSIS:</u>

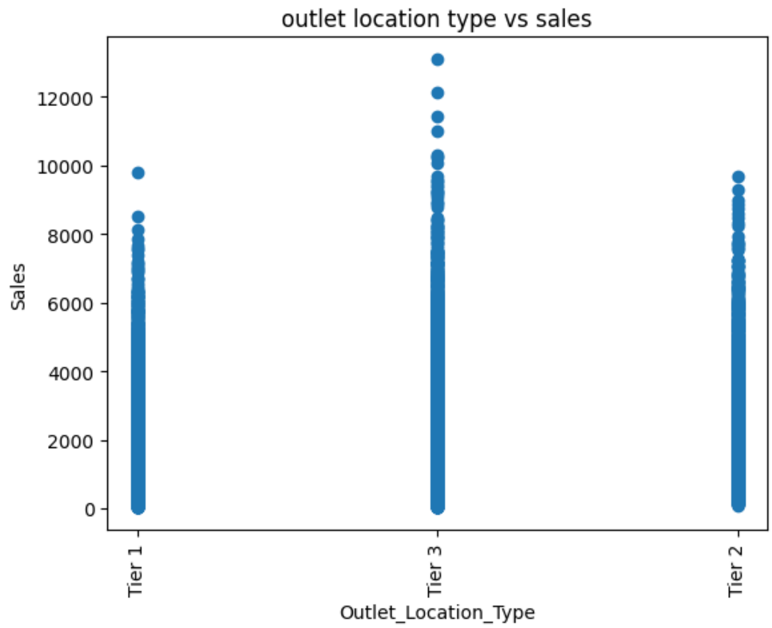


<u>Observation:</u>

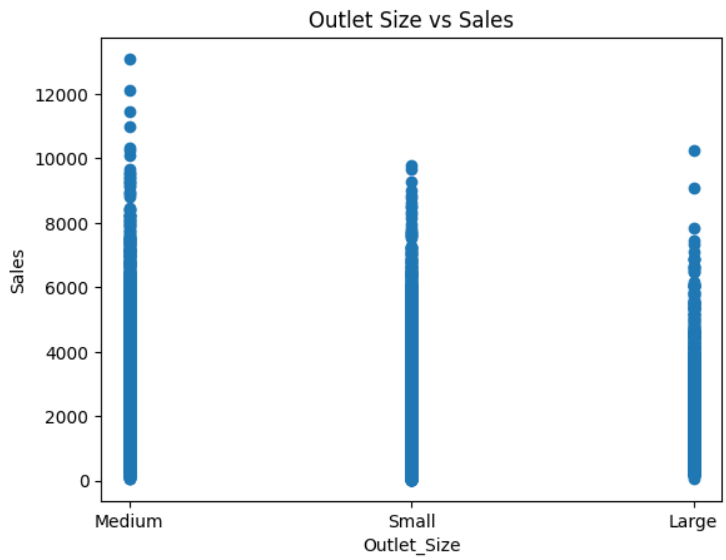
By analysing the graph, most of the sales came from the fruits and vegetables, dairy and household and least with the others and hard drinks.



Observation: By analysing the graph, we can see most of the sales came from the supermarket type 3 and less sales from the grocery store.

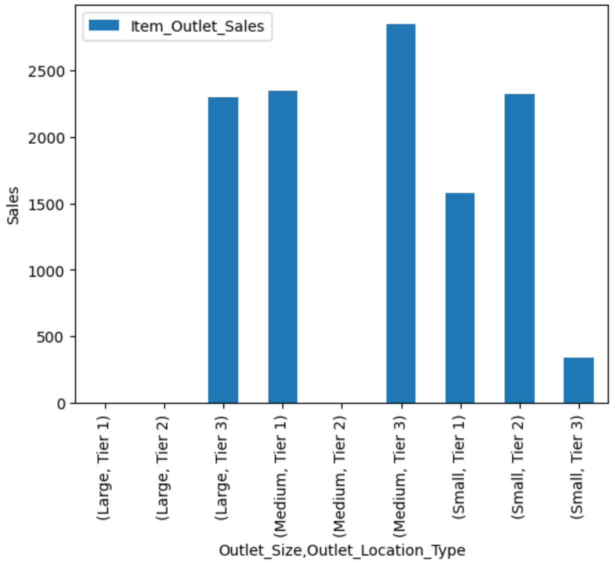


Observation: By analyzing the graph, we can see most of the sales came from the tier 3 location and less sales from the tier 1.



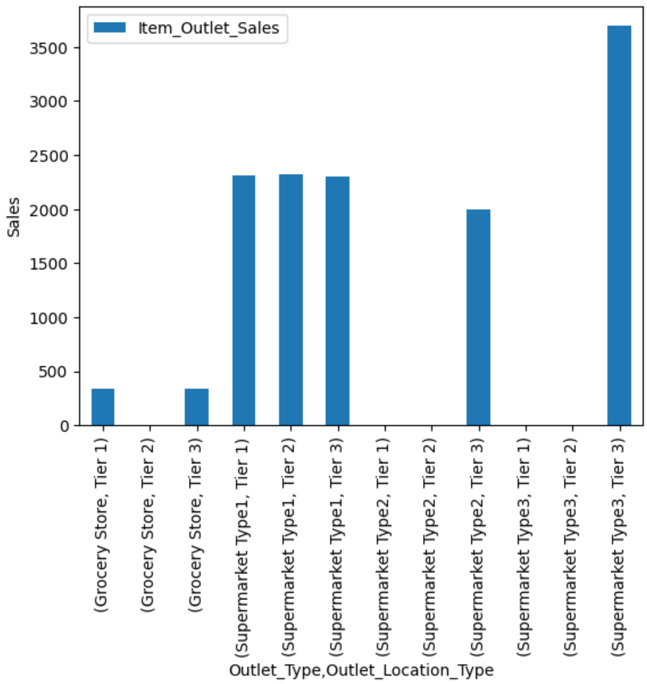
Observation:By analysing the graph , we can see most of the sales came from the medium size store. And less sales from the large store.

MULTIVARIATE ANALYSIS:



Observation:

By analyzing the graph we can see most of the sales came from the medium size store which was located in the tier-3 location, and less number of sales came from the Small sized store from the tier-3 location.



Observation:

By analyzing the graph we can see that most of the sales came from the tier-3 and the type of the supermarket is supermarket type-3.From grocery store tier-3 There are less number of sales.