BIG MART SALES PREDICTION

- Big Mart is a big supermarket that sales different products across stores in different cities.
- The dataset was taken from Kaggle website (BigMart Sales Data | Kaggle).
- The dataset contains 8523 entries and 12 features

Detail Compact Column 10 of 12 columns

Item_Ident...	Item_Weight	Item_Fat_...	Item_Visibi...	Item_Type	Item_MRP	Outlet_Ide...	
FDA15	9.3	Low Fat	0.016047301	Dairy	249.8092	OUT049	1
DRC01	5.92	Regular	0.019278216	Soft Drinks	48.2692	OUT018	2
FDN15	17.5	Low Fat	0.016760075	Meat	141.618	OUT049	1
FDX07	19.2	Regular	0	Fruits and	182.095	OUT010	1

- From Big Mart dataset I am going to:
  - analyze customer behavior
  - understand the properties of products and stores which play a key role in increasing sales.
- My goal is to create a model that can predict the sales per product for each store .

- Before Analyzing and exploring the data I do the following:

```
<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
```

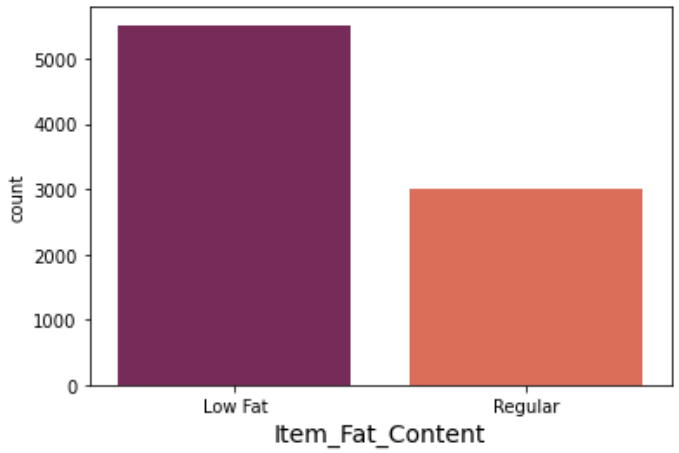
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                8523 non-null   float64
 2   Item_Fat_Content           8523 non-null   int32
 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                8523 non-null   int32
 9   Outlet_Location_Type       8523 non-null   int32
 10  Outlet_Type                8523 non-null   object
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int32(3), int64(1), object(4)
memory usage: 699.3+ KB
```

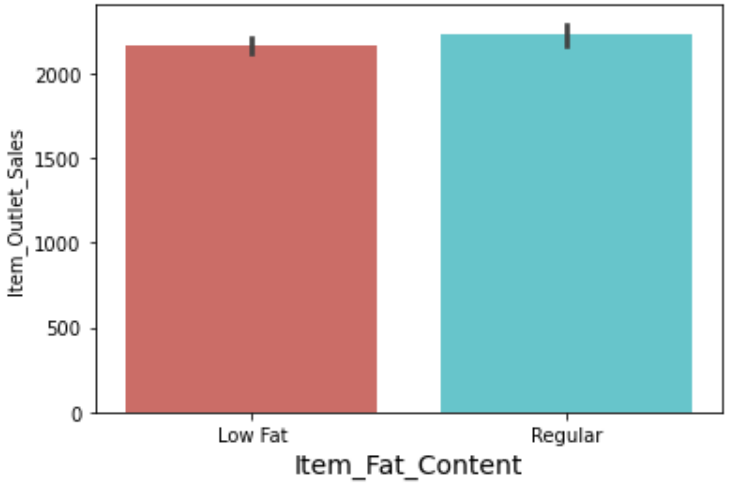
- Before Analyzing and exploring the data I do the following:

```
: Low Fat    5089
  Regular    2889
  LF          316
  reg         117
  low fat     112
```

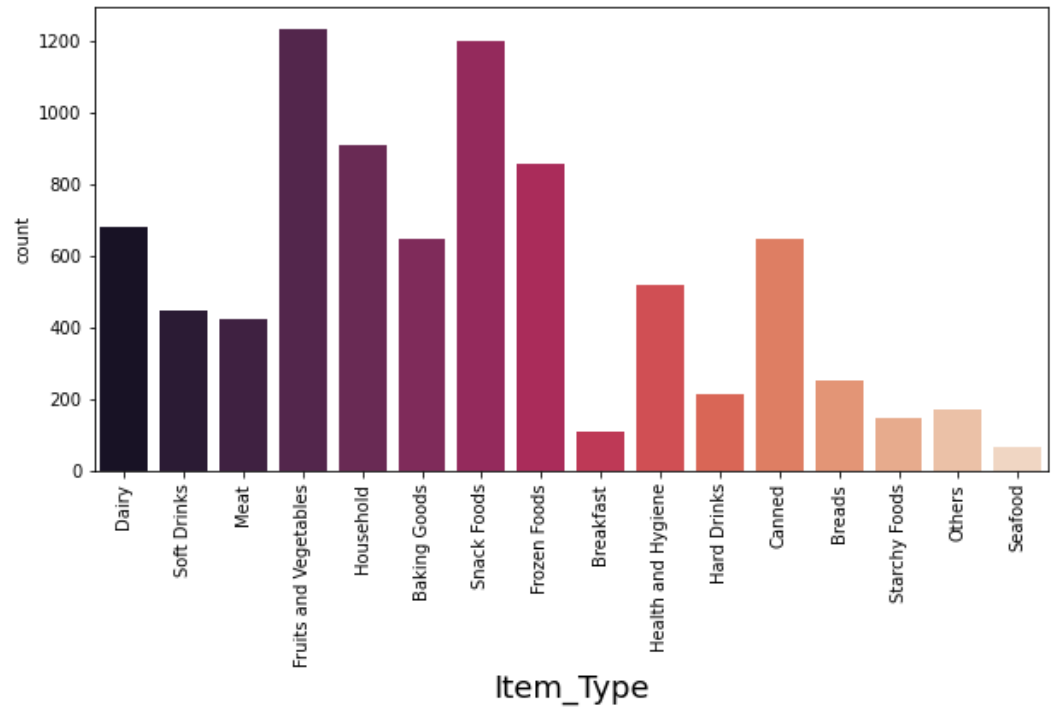
```
Low Fat    5517
Regular    3006
Name: Item_Fat_Content, dtype: int64
```

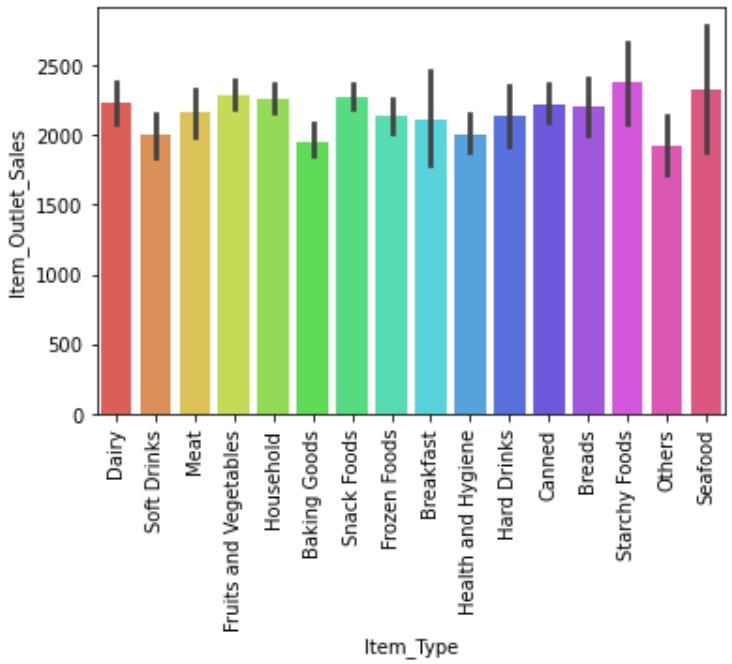
- From analyzing and exploring the data I answer some questions such as:
- Which product people bought more low fat or regular fat?
- Which product have more profitable the low fat or regular fat?



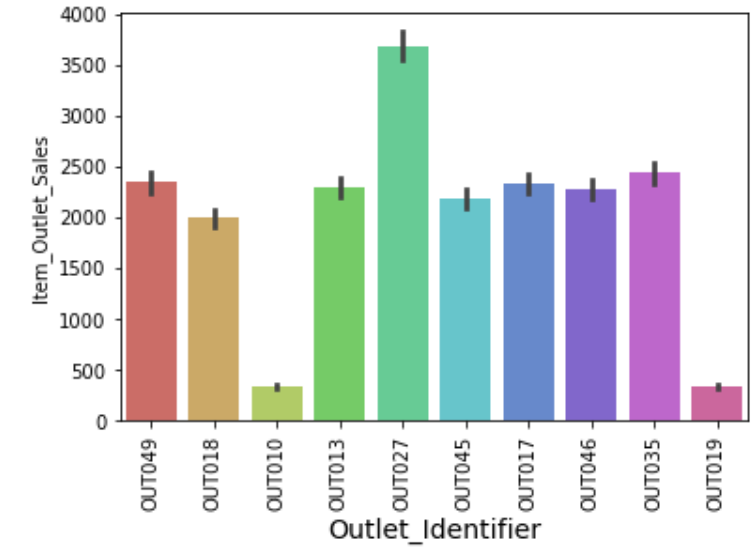


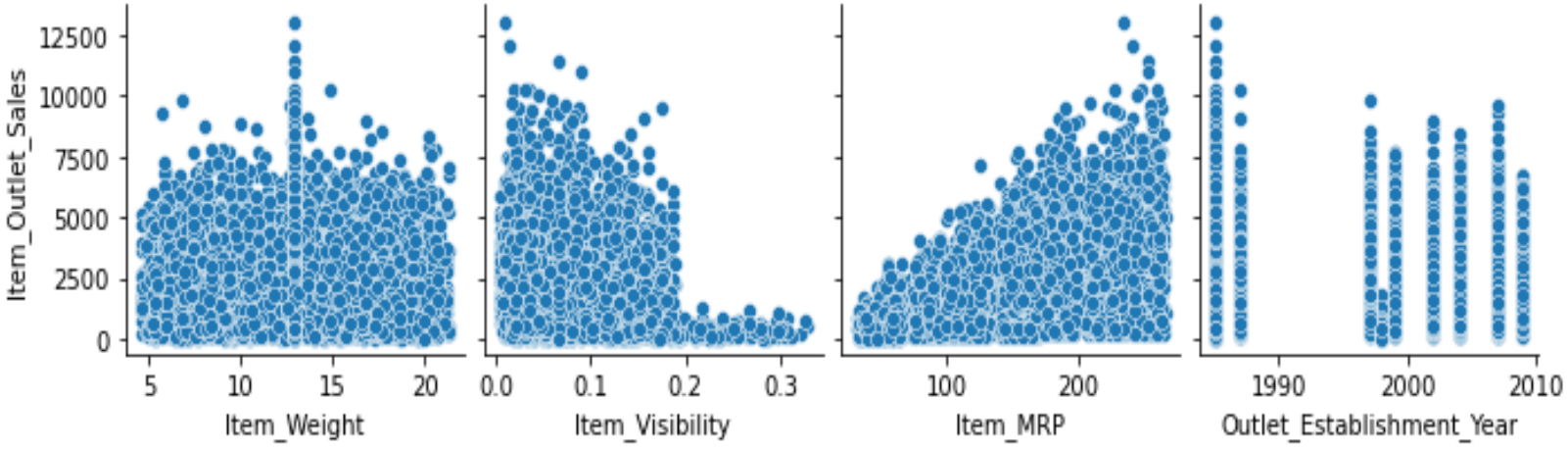
- What kind of product people bought more?
- What product have most profit?



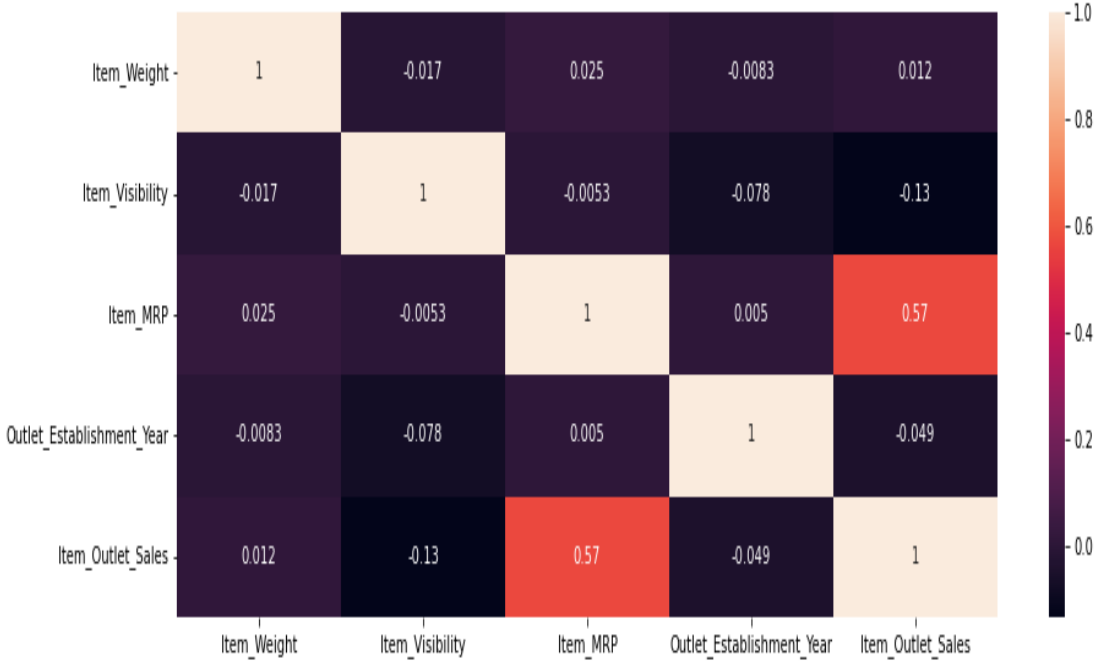


- Which outlet have the top sales?




Item_Outlet_Sales
12500
10000
7500
5000
2500
0
5
10
15
20
Item_Weight
0.0
0.1
0.2
0.3
Item_Visibility
100
200
Item_MRP
1990
2000
2010
Outlet_Establishment_Year

Heatmap Correlations



Before building my model, I do some future engineering.

- Label encoding on: Item_Fat_Content, Outlet_Size, and Outlet_Location_Type.
- One hot encoder on: Item_Type and Outlet_Type.

And I drop some columns:

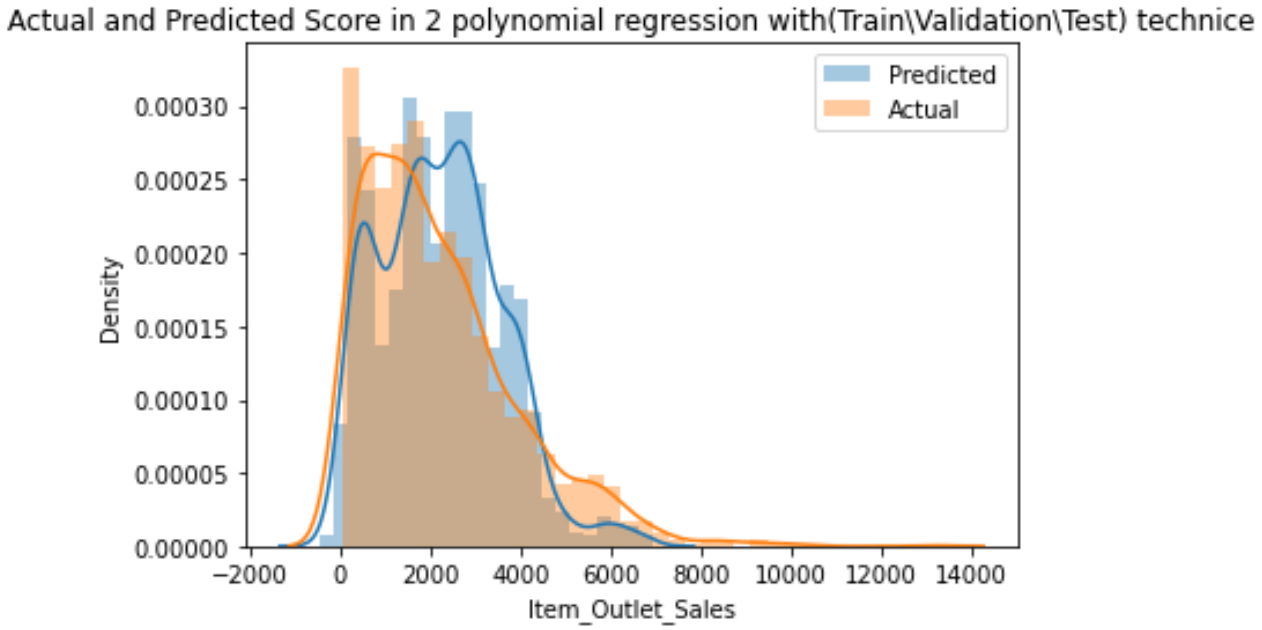
- Item_Identifier, Outlet_Identifier ,Outlet_Establishment_Year, Outlet_Type, Item_Type.

-I split my data to (80% for train / validation) / (20% for test) I fit the models on train set ,and test on validation and test set.

- I used a **Regression Supervised Machine Learning algorithms**

	Models	R^2
0	Degree 2 polynomial regression	0.612
2	Random Forest Regression	0.584
4	Linear Regression	0.574
1	Cross-Validation	0.558
3	Random Forest with Hyper Parameter Tuning	0.556
5	Ridge Regression	-7536.000

Show the difference between the Actual numbers and Predicted numbers



	Actual	Predicted
7186	3649.2498	4661.0
2283	1845.5976	1523.0
2206	2675.1844	3226.0
5446	675.7870	1415.0
6380	3755.1120	3149.0
...	...	...
2879	491.3604	313.0
6094	165.7842	308.0
1598	1225.0720	1020.0
8012	3146.5708	3031.0
7756	2563.3300	2749.0

1705 rows × 2 columns

- I also try to train my model on data split (80% for train) / (20% for test).
- I fit the models on train set, and test on test set.

	Models	R^2
2	Degree 2 polynomial regression	0.595
1	Random Forest Regression	0.589
3	Lasso	0.563
0	Linear Regression	0.562

Thank You

Done by :

Tahani al shedoukhi