Get Understanding about Data Set

There are 12 variable in dataset

1.item_identifier 2.item_weight 3.item_Fat_Contact 4.item_visibility 5.item_type 6.item_mrp 7.outlet_identifier 8.outlet_establisment_year 9.outlet_size 10.outlet_location_type 11.outlet_type 12.item_outlet_sales

Import Library

```
import pandas as pd
```

```
import numpy as np
```

Import CSV as DataFrame

```
df = pd.read_csv(r'https://raw.githubusercontent.com/YBI-Foundation/Dataset/main/Big%20Sales
```

```
# df = pd.read_csv(r'C:\Users\YBI Foundation\Desktop\Big Sales Data.csv')
```

```
# df = pd.read_csv(r'/content/Big Sales Data.csv')
```

Get the First Five Rows of Dataframe

```
df.head()
```

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP
0	FDT36	12.3	Low Fat	0.111448	Baking Goods	33.4874
1	FDT36	12.3	Low Fat	0.111904	Baking Goods	33.9874
2	FDT36	12.3	LF	0.111728	Baking Goods	33.9874
					Baking	

Get Information of DataFrame

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            14204 non-null  object
 1   Item_Weight                11815 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                14204 non-null  object
 9   Outlet_Location_Type       14204 non-null  object
 10  Outlet_Type                14204 non-null  object
 11  Item_Outlet_Sales          14204 non-null  float64
dtypes: float64(4), int64(1), object(7)
memory usage: 1.3+ MB
```

```
df.columns
```

```
Index(['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'],
      dtype='object')
```

Get the Summary Satistics

```
df.describe()
```

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outl
count	11815.000000	14204.000000	14204.000000	14204.000000	1420
mean	12.788355	0.065953	141.004977	1997.830681	218
std	4.654126	0.051459	62.086938	8.371664	182
min	4.555000	0.000000	31.290000	1985.000000	3
25%	8.710000	0.027036	94.012000	1987.000000	92
50%	12.500000	0.054021	142.247000	1999.000000	176
75%	16.750000	0.094037	185.855600	2004.000000	298
max	30.000000	0.328391	266.888400	2009.000000	3122

```
df['Item_Weight'].fillna(df.groupby(['Item_Type'])['Item_Weight'].transform('mean'), inplace
```

```
df.info()
```

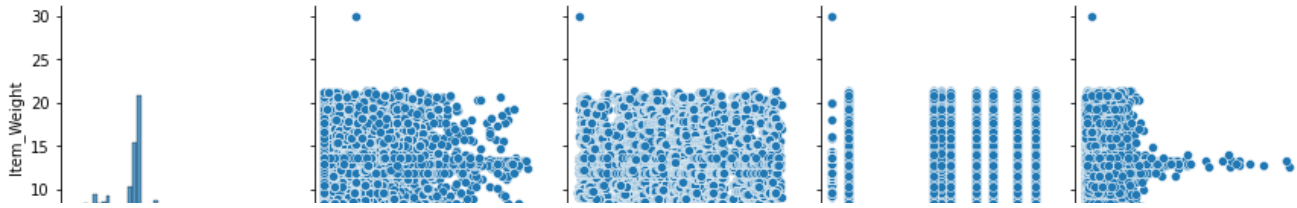
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            14204 non-null  object 
 1   Item_Weight                14204 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                14204 non-null  object 
 9   Outlet_Location_Type       14204 non-null  object 
 10  Outlet_Type                14204 non-null  object 
 11  Item_Outlet_Sales          14204 non-null  float64
dtypes: float64(4), int64(1), object(7)
memory usage: 1.3+ MB
```

```
df.describe()
```

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outl
count	14204.000000	14204.000000	14204.000000	14204.000000	142
mean	12.790642	0.065953	141.004977	1997.830681	21
std	4.251186	0.051459	62.086938	8.371664	18

```
import seaborn as sns
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f4a03890350>
```



Get Categories and Counts of Categorical Variables

```
df[['Item_Identifier']].value_counts()
```

```
Item_Identifier
FDQ08              10
FDO24              10
FDQ19              10
FDQ28              10
FDQ31              10
                   ..
FDM52               7
FDM50               7
FDL50               7
FDM10               7
FDR51               7
Length: 1559, dtype: int64
```

```
df[['Item_Fat_Content']].value_counts()
```

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

Item_Weight Item_Visibility Item_MRP Outlet_Establishment_Year Item_Outlet_Sales

```
df.replace({'Item_Fat_Content': {'LF':'Low Fat','reg':'Regular','low fat':'Low Fat'}}, inpla
```

```
df[['Item_Fat_Content']].value_counts()
```

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

```
df.replace({'Item_Fat_Content': {'Low Fat': 0,'Regular' : 1}}, inplace=True)
```

```
df[['Item_Type']].value_counts()
```

```
Item_Type
Fruits and Vegetables    2013
Snack Foods              1989
Household                1548
Frozen Foods             1426
Dairy                    1136
Baking Goods             1086
Canned                   1084
Health and Hygiene        858
Meat                      736
Soft Drinks               726
Breads                    416
Hard Drinks               362
Others                    280
Starchy Foods             269
Breakfast                 186
Seafood                    89
dtype: int64
```

```
df.replace({'Item_Tupe':{'Fruite and Vegetables':0,'Snack Foods':0,'Household':1,
                         'Frozen Foods': 0, 'Dairy': 0, 'Baking Goods': 0,
                         'Canned' : 0, 'Health and Hygiene' : 1,
                         'Meat' : 0,'Soft Drinks' : 0,'Breads' : 0,'Head Drinks' : 0,
                         'Others' : 2,'Starchy Foods' : 0, 'Breakfast' : 0, 'Seafood' :0
                         }},inplace=True)
```

```
df[['Item_Type']].value_counts()
```

```
Item_Type
Fruits and Vegetables    2013
Snack Foods              1989
Household                1548
Frozen Foods             1426
Dairy                    1136
Baking Goods             1086
Canned                   1084
Health and Hygiene        858
Meat                      736
Soft Drinks               726
Breads                    416
Hard Drinks               362
Others                    280
Starchy Foods             269
Breakfast                 186
Seafood                    89
dtype: int64
```

```
df[['Outlet_Identifier']].value_counts()
```

```
Outlet_Identifier
OUT027               1559
OUT013               1553
```

```
OUT035               1550
OUT046               1550
OUT049               1550
OUT045               1548
OUT018               1546
OUT017               1543
OUT010                925
OUT019                880
dtype: int64
```

```
df.replace({'Outlet_Identifier':{'OUT027' : 0,'OUT013': 1,
                                 'OUT049' : 2, 'OUT046' : 3, 'OUT035' : 4,
                                 'OUT045' : 7, 'OUT010' : 8, 'OUT019' : 9,
                                 }},inplace=True)
```

```
df[['Outlet_Size']].value_counts()
```

```
Outlet_Size
Medium         7122
Small          5529
High           1553
dtype: int64
```

```
df.replace({'Outlet_Size': {'Small': 0,'Medium' : 1, 'High' : 2}},inplace=True)
```

```
df[['Outlet_Size']].value_counts()
```

```
Outlet_Size
1              7122
0              5529
2              1553
dtype: int64
```

```
df[['Outlet_Location_Type']].value_counts()
```

```
Outlet_Location_Type
Tier 3                  5583
Tier 2                  4641
Tier 1                  3980
dtype: int64
```

```
df.replace({'Outlet_Location_Type': {'Tier 1': 0,'Tier 2' :1,'Tier 3' : 2}},inplace=True)
```

```
df[['Outlet_Type']].value_counts()
```

```
Outlet_Type
Supermarket Type1    9294
Grocery Store        1805
Supermarket Type3    1559
```

```
Supermarket Type2    1546
dtype: int64
```

```
df.replace({'Outlet_Type': {'Grocery Store': 0,'Supermarket Type1' : 1, 'Suprtmarket Type2'
```

```
df[['Outlet_Type']].value_counts()
```

```
Outlet_Type
1                    9294
0                    1805
3                    1559
Supermarket Type2    1546
dtype: int64
```

```
df.head()
```

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP
0	FDT36	12.3	Low Fat	0.111448	Baking Goods	33.4874
1	FDT36	12.3	Low Fat	0.111904	Baking Goods	33.9874
2	FDT36	12.3	LF	0.111728	Baking Goods	33.9874
3	FDT36	12.3	Low Fat	0.000000	Baking Goods	34.3874
4	FDP12	9.8	Regular	0.045523	Baking Goods	35.0874

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            14204 non-null  object
 1   Item_Weight                14204 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                14204 non-null  int64
 9   Outlet_Location_Type       14204 non-null  int64
 10  Outlet_Type                14204 non-null  object
 11  Item_Outlet_Sales          14204 non-null  float64
dtypes: float64(4), int64(3), object(5)
memory usage: 1.3+ MB
```

Get Shape of DataFrame

```
df.shape
```

```
(14204, 12)
```

```
y = df['Item_Outlet_Sales']
```

```
y.shape
```

```
(14204,)
```

```
y
```

```
0         436.608721
1         443.127721
2         564.598400
3        1719.370000
4         352.874000
             ...
14199    4984.178800
14200    2885.577200
14201    2885.577200
14202    3803.676434
14203    3644.354765
Name: Item_Outlet_Sales, Length: 14204, dtype: float64
```

```
X = df[['Item_Weight', 'Item_Fat_Content', 'Item_Visibility',
        'Item_Type', 'Item_MRP', 'Outlet_Identifier',
        'Outlet_Establishment_Year', 'Outlet_Size', 'Outlet_Size', 'Outlet_Size', 'Outlet_Lo
        'Outlet_Type']]
```

```
X = df.drop(['Item_Identifier', 'Item_Outlet_Sales'], axis=1)
```

```
X.shape
```

```
(14204, 10)
```

```
X
```

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Iden
0	12.300000	Low Fat	0.111448	Baking Goods	33.4874	
1	12.300000	Low Fat	0.111904	Baking Goods	33.9874	C
2	12.300000	LF	0.111728	Baking Goods	33.9874	C
3	12.300000	Low Fat	0.000000	Baking Goods	34.3874	
4	9.800000	Regular	0.045523	Baking Goods	35.0874	C
...	...	...	...	...	...	
14199	12.800000	Low Fat	0.069606	Starchy Foods	261.9252	
14200	12.800000	Low Fat	0.070013	Starchy Foods	262.8252	C

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_std = df[['Item_Weight', 'Item_Visibility', 'Item_MRP', 'Outlet_Establishment_Year' ]]

X_std = sc.fit_transform(X_std)

X_std
```

```
array([[-0.11541705,  0.88413635, -1.73178716,  0.13968068],
       [-0.11541705,  0.89300616, -1.72373366,  1.09531886],
       [-0.11541705,  0.88958331, -1.72373366,  1.3342284 ],
       ...,
       [ 0.00220132,  0.07011952,  1.96538148, -1.29377659],
       [ 0.20444792,  0.06469366,  1.97343499, -1.53268614],
       [ 0.00220132,  0.07334891,  1.97504569,  0.13968068]])
```

```
X
```

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Iden
0	12.300000	Low Fat	0.111448	Baking Goods	33.4874	
1	12.300000	Low Fat	0.111904	Baking Goods	33.9874	C
2	12.300000	LF	0.111728	Baking Goods	33.9874	C
3	12.300000	Low Fat	0.000000	Baking Goods	34.3874	
4	9.800000	Regular	0.045523	Baking Goods	35.0874	C
...	...	...	...	...	...	
14199	12.800000	Low Fat	0.069606	Starchy	261.9252	
14200	12.800000	Low Fat	0.070013	Starchy	262.8252	C

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.1, random_state=2529)
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((12783, 10), (1421, 10), (12783,), (1421,))
```

Get Model Train

```
from sklearn.ensemble import RandomForestRegressor
```

```
rfr = RandomForestRegressor(random_state=2529)
```

0s completed at 3:17 PM