LAB 02

EC 9560 - DATA MINING

THEVARAJAN.R.J

2019/E/146

SEMESTER 7

06 OCT 2023

TITLE:

Big Mart Sales Prediction

OBJECTIVE:

Use regression analysis to predict sales based on attributes.

PROGRESS:

As Item_Fat_Content feature column contains same datas in different names like Low Fat = LF = low fat and Regular = reg.

```
Item_Fat_Content
Low Fat     5089
Regular     2889
LF           316
reg          117
low fat      112
Name: Item_Fat_Content, dtype: int64
```

Combining those datas into one name:

In [181]:
```
#Combine item_fat content
df['Item_Fat_Content'] = df['Item_Fat_Content'].replace({'LF':'Low Fat','reg':'Regular','low fat':'Low Fat'})
df['Item_Fat_Content'].value_counts()
```

Out[181]:
```
Low Fat    5517
Regular    3006
Name: Item_Fat_Content, dtype: int64
```

Creating new attributes:

```
CREATION OF NEW ATTRIBUTES
```

In [182]:
```
df['New_Item_Type'] = df['Item_Identifier'].apply(lambda x: x[:2])
df['New_Item_Type']
```

Out[182]:
```
0       FD
1       DR
2       FD
3       FD
4       NC
        ..
8518    FD
8519    FD
8520    NC
8521    FD
8522    DR
Name: New_Item_Type, Length: 8523, dtype: object
```

In [183]:
```
df['New_Item_Type'] = df['New_Item_Type'].map({'FD':'Food','NC':'Non-Consumable','DR':'Drinks'})
df['New_Item_Type'].value_counts()
```

Out[183]:
```
Food              6125
Non-Consumable    1599
Drinks             799
Name: New_Item_Type, dtype: int64
```

In [184]:
```
df.loc[df['New_Item_Type']== 'Non-Consumable','Item_Fat_Content'] = 'Non-Edible'
df['Item_Fat_Content'].value_counts()
```

Out[184]:
```
Low Fat       3918
Regular       3006
Non-Edible    1599
Name: Item_Fat_Content, dtype: int64
```

As Outlet_Establishment_Year contain very large number , normalizing it to small number.

NORMALIZING DATA

In [185]:
```
#Create small values for establishment_year
df['Outlet_Years'] = 2013 - df['Outlet_Establishment_Year']
df['Outlet_Years']
```

Out[185]:
```
0       14
1        4
2       14
3       15
4       26
        ..
8518    26
8519    11
8520     9
8521     4
8522    16
Name: Outlet_Years, Length: 8523, dtype: int64
```

Reading test files as well and filling null values.

FILLING NULL VALUES

In [187]:
```
#Reading the test dataset from the directory
df_test=pd.read_csv('test_AbJTz2l.csv')
```

In [188]:
```
df_test
```

Out[188]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Locat
0	FDW58	20.750	Low Fat	0.007565	Snack Foods	107.8622	OUT049	1999	Medium	
1	FDW14	8.300	reg	0.038428	Dairy	87.3198	OUT017	2007	NaN	
2	NCN55	14.600	Low Fat	0.099575	Others	241.7538	OUT010	1998	NaN	
3	FDQ58	7.315	Low Fat	0.015388	Snack Foods	155.0340	OUT017	2007	NaN	
4	FDY38	NaN	Regular	0.118599	Dairy	234.2300	OUT027	1985	Medium	
...	...	...	...	...	...	...	...	...	...	
5676	FDB58	10.500	Regular	0.013496	Snack Foods	141.3154	OUT046	1997	Small	
5677	FDD47	7.600	Regular	0.142991	Starchy Foods	169.1448	OUT018	2009	Medium	
5678	NCO17	10.000	Low Fat	0.073529	Health and Hygiene	118.7440	OUT045	2002	NaN	
5679	FDJ26	15.300	Regular	0.000000	Canned	214.6218	OUT017	2007	NaN	
5680	FDU37	9.500	Regular	0.104720	Canned	79.7960	OUT045	2002	NaN	

5681 rows × 11 columns

In [189]:
```
df_test.head()
```

Out[189]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location
0	FDW58	20.750	Low Fat	0.007565	Snack Foods	107.8622	OUT049	1999	Medium	
1	FDW14	8.300	reg	0.038428	Dairy	87.3198	OUT017	2007	NaN	
2	NCN55	14.600	Low Fat	0.099575	Others	241.7538	OUT010	1998	NaN	
3	FDQ58	7.315	Low Fat	0.015388	Snack Foods	155.0340	OUT017	2007	NaN	
4	FDY38	NaN	Regular	0.118599	Dairy	234.2300	OUT027	1985	Medium	

In [190]:
```
df_test.isnull().sum()
```

Out[190]:
```
Item_Identifier                 0
Item_Weight                   976
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                  1606
Outlet_Location_Type            0
Outlet_Type                     0
dtype: int64
```

In [190]:
```
df_test.isnull().sum()
```

Out[190]:
```
Item_Identifier                 0
Item_Weight                   976
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                  1606
Outlet_Location_Type            0
Outlet_Type                     0
dtype: int64
```

In [191]:
```
#For train dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 14 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
 12  New_Item_Type              8523 non-null   object
 13  Outlet_Years               8523 non-null   int64
dtypes: float64(4), int64(2), object(8)
memory usage: 932.3+ KB
```

In [192]:
```
df.isnull().sum()
```

Out[192]:
```
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
```

In [240]:
```
df['Item_Weight']
```

Out[240]:
```
0        9.300
1        5.920
2       17.500
3       19.200
4        8.930
         ...
8518     6.865
8519     8.380
8520    10.600
8521     7.210
8522    14.800
Name: Item_Weight, Length: 8523, dtype: float64
```

In [241]:
```
df['Item_Weight'].describe()
```

Out[241]:
```
count    7060.000000
mean       12.857645
std         4.643456
min         4.555000
25%         8.773750
50%        12.600000
75%        16.850000
max        21.350000
Name: Item_Weight, dtype: float64
```

Item_Weight is a numerical column and its mean used to fill its null values.

AS ITEM_WEIGHT IS A NUMERICAL COLUMN, FILLING THE NULL VALUES USING ITS CORRESPONDING COLUMN'S MEAN VALUE. MEAN = 12.857645

In [242]:
```
#Filling the missing value directly in the original  train Dataframe
df['Item_Weight'].fillna(df['Item_Weight'].mean(),inplace=True)
#Filling the missing value directly in the original  test Dataframe
df_test['Item_Weight'].fillna(df_test['Item_Weight'].mean(),inplace=True)
```

In [243]:
```
df.isnull().sum()
```

Out[243]:
```
Item_Identifier               0
Item_Weight                   0
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
New_Item_Type                 0
Outlet_Years                  0
dtype: int64
```

In [244]:
```
df['Outlet_Size']
```

Out[244]:
```
0       Medium
1       Medium
2       Medium
3          NaN
4         High
          ...
8518      High
8519       NaN
8520     Small
8521    Medium
8522     Small
Name: Outlet_Size, Length: 8523, dtype: object
```

In [245]:
```
df['Outlet_Size'].value_counts()
```

Out[245]:
```
Medium    2793
Small     2388
High       932
Name: Outlet_Size, dtype: int64
```

In [246]:
```
df['Outlet_Size'].mode()
```

Out[246]:
```
0    Medium
Name: Outlet_Size, dtype: object
```

Outlet_Size is a categorical column and its mode is used to fill its null values.

AS OUTLET_SIZE IS A CATEGORICAL COLUMN, FILLING THE NULL VALUES USING ITS CORRESPONDING COLUMN'S MODE VALUE

In [247]:
```
#Filling null values directly in the original train Dataframe
df['Outlet_Size'].fillna(df['Outlet_Size'].mode()[0],inplace=True)
#Filling null values directly in the original test Dataframe
df_test['Outlet_Size'].fillna(df_test['Outlet_Size'].mode()[0],inplace=True)
```

In [248]:
```
df.isnull().sum()
```

Out[248]:
```
Item_Identifier              0
Item_Weight                  0
Item_Fat_Content             0
Item_Visibility              0
Item_Type                    0
Item_MRP                     0
Outlet_Identifier            0
Outlet_Establishment_Year    0
Outlet_Size                  0
Outlet_Location_Type         0
Outlet_Type                  0
Item_Outlet_Sales            0
New_Item_Type                0
Outlet_Years                 0
dtype: int64
```

SELECTING FEATURES BASED ON GENERAL REQUIREMENTS.

In [249]:
```
#Removing Item_Identifier and Outlet_Identifier as they are containing unique identifiers for items and outlets
df.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True)
df_test.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True)
```

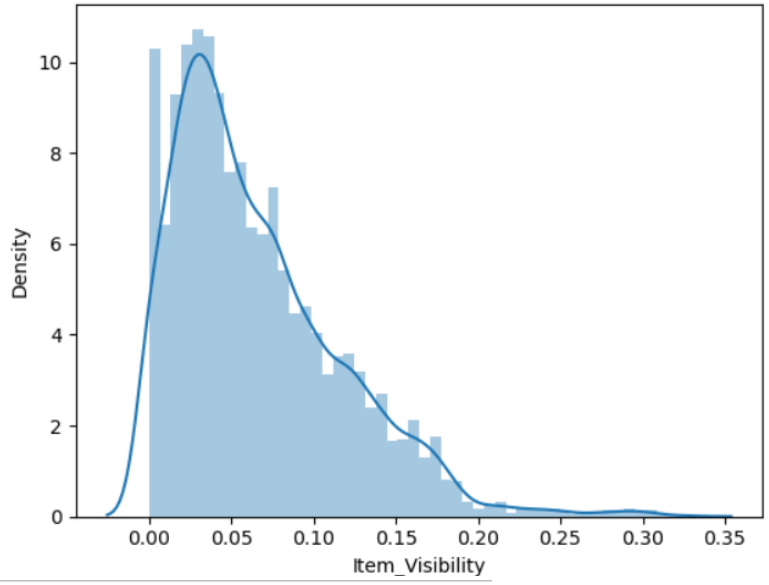
In [250]:
```
df.head()
```

Out[250]:

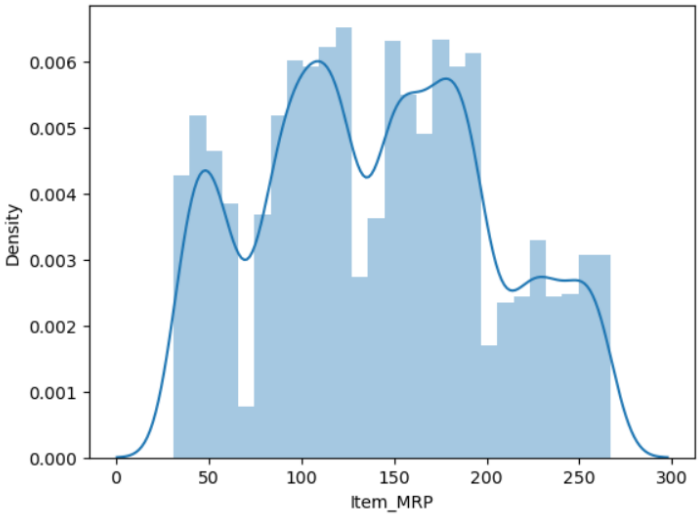
	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_
0	9.30	Low Fat	0.016047	Dairy	249.8092	1999	Medium	Tier 1	Supermarket Type1	3735
1	5.92	Regular	0.019278	Soft Drinks	48.2692	2009	Medium	Tier 3	Supermarket Type2	443
2	17.50	Low Fat	0.016760	Meat	141.6180	1999	Medium	Tier 1	Supermarket Type1	2097
3	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	1998	Medium	Tier 3	Grocery Store	732
4	8.93	Non-Edible	0.000000	Household	53.8614	1987	High	Tier 3	Supermarket Type1	994

VISUALIZATION:

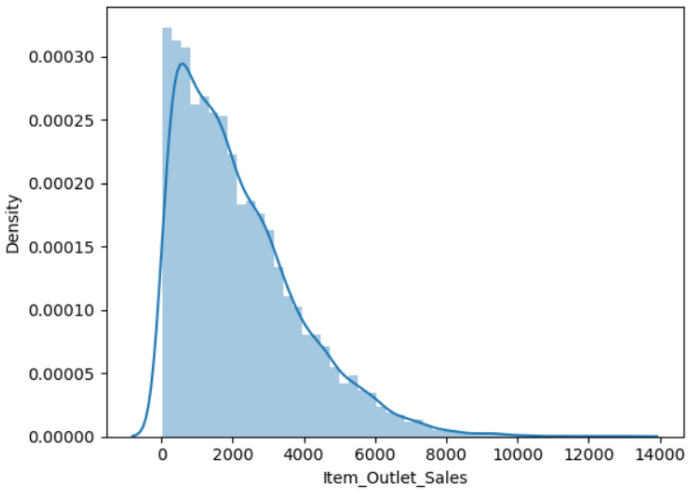
```
In [252]: sns.distplot(df['Item_Visibility'])
```




```
: sns.distplot(df['Item_MRP'])
```




```
: sns.distplot(df['Item_Outlet_Sales'])
```


0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
Density
0
2000
4000
6000
8000
10000
12000
14000
Item_Outlet_Sales

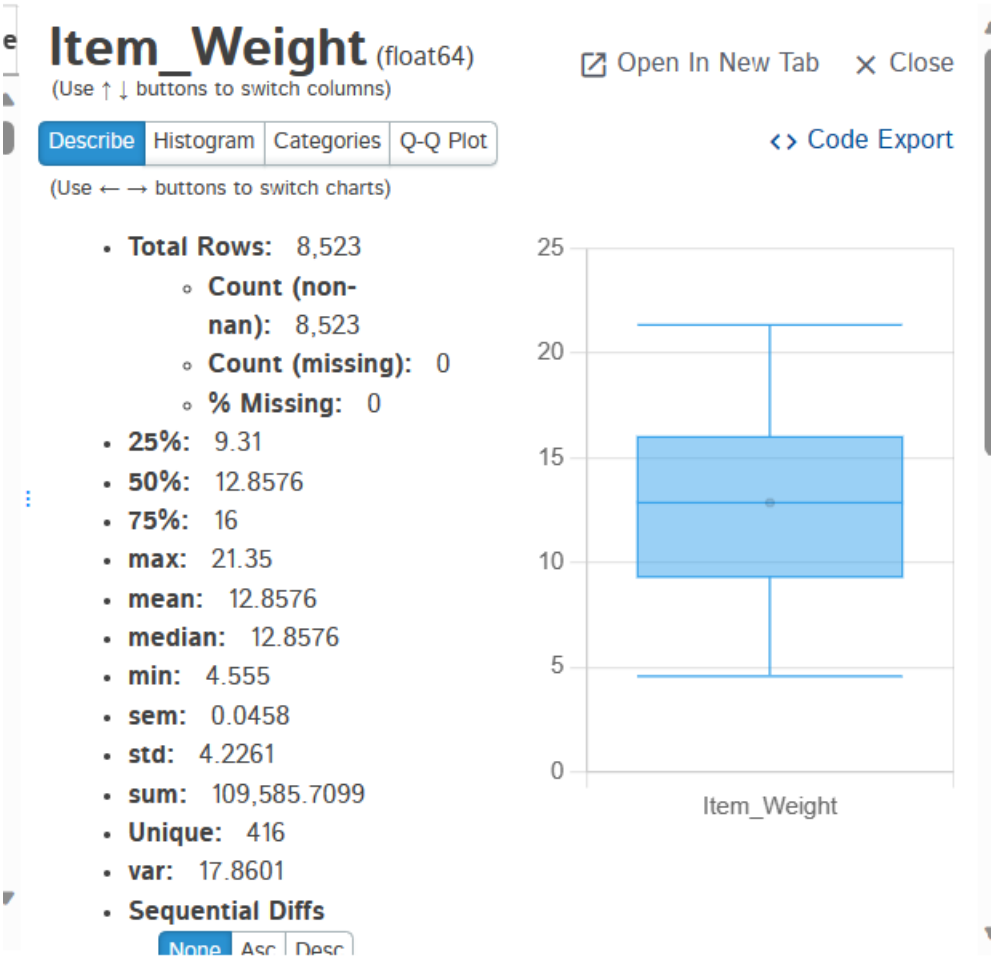
For Exploratory Data Analysis (EDA), tried dtale to analysis.

```
In [255]: import dtale

In [256]: dtale.show(df)
```

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size
0	9.30	Low Fat	0.02	Dairy	249.81	1999	Medium
1	5.92	Regular	0.02	Soft Drinks	48.27	2009	Medium
2	17.50	Low Fat	0.02	Meat	141.62	1999	Medium
3	19.20	Regular	0.00	Fruits and Vegetables	182.10	1998	Medium
4	8.93	Non-Edible	0.00	Household	53.86	1987	High
5	10.40	Regular	0.00	Baking Goods	51.40	2009	Medium
6	13.65	Regular	0.01	Snack Foods	57.66	1987	High
7	12.86	Low Fat	0.13	Snack Foods	107.76	1985	Medium
8	16.20	Regular	0.02	Frozen Foods	96.97	2002	Medium
9	19.20	Regular	0.09	Frozen Foods	187.82	2007	Medium
10	11.80	Low Fat	0.00	Fruits and Vegetables	45.54	1999	Medium
11	18.50	Regular	0.05	Dairy	144.11	1997	Small
12	15.10	Regular	0.10	Fruits and Vegetables	145.48	1999	Medium
13	17.60	Regular	0.05	Snack Foods	119.68	1997	Small
14	16.35	Low Fat	0.07	Fruits and Vegetables	196.44	1987	High
15	9.00	Regular	0.07	Breakfast	56.36	1997	Small
16	11.80	Non-Edible	0.01	Health and Hygiene	115.35	2009	Medium

Using dtale library, description of Item_Weight was analyzed:



- There is no outliers for Item_Weightaccording to this graph.

- For Outlet_Size: