```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import keras

mart=pd.read_csv("bigmart1.csv")

mart
```

```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0              FDA15        9.300          Low Fat         0.016047
1              DRC01        5.920          Regular         0.019278
2              FDN15       17.500          Low Fat         0.016760
3              FDX07       19.200          Regular         0.000000
4              NCD19        8.930          Low Fat         0.000000
...              ...          ...              ...              ...
8518           FDF22        6.865          Low Fat         0.056783
8519           FDS36        8.380          Regular         0.046982
8520           NCJ29       10.600          Low Fat         0.035186
8521           FDN46        7.210          Regular         0.145221
8522           DRG01       14.800          Low Fat         0.044878

                  Item_Type  Item_MRP Outlet_Identifier  \
0                     Dairy  249.8092            OUT049
1               Soft Drinks   48.2692            OUT018
2                      Meat  141.6180            OUT049
3     Fruits and Vegetables  182.0950            OUT010
4                 Household   53.8614            OUT013
...                     ...      ...               ...
8518            Snack Foods  214.5218            OUT013
8519          Baking Goods  108.1570            OUT045
8520    Health and Hygiene   85.1224            OUT035
8521            Snack Foods  103.1332            OUT018
8522            Soft Drinks   75.4670            OUT046

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                          1999      Medium               Tier 1
1                          2009      Medium               Tier 3
2                          1999      Medium               Tier 1
3                          1998         NaN               Tier 3
4                          1987        High               Tier 3
...                         ...         ...                  ...
8518                       1987        High               Tier 3
8519                       2002         NaN               Tier 2
8520                       2004       Small               Tier 2
8521                       2009      Medium               Tier 3
8522                       1997       Small               Tier 1

            Outlet_Type  Item_Outlet_Sales
```

```
0     Supermarket Type1          3735.1380
1     Supermarket Type2           443.4228
2     Supermarket Type1          2097.2700
3         Grocery Store           732.3800
4     Supermarket Type1           994.7052
...                 ...                ...
8518  Supermarket Type1          2778.3834
8519  Supermarket Type1           549.2850
8520  Supermarket Type1          1193.1136
8521  Supermarket Type2          1845.5976
8522  Supermarket Type1           765.6700

[8523 rows x 12 columns]
```

```
mart.head(10)
```

```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDA15        9.300          Low Fat         0.016047
1           DRC01        5.920          Regular         0.019278
2           FDN15       17.500          Low Fat         0.016760
3           FDX07       19.200          Regular         0.000000
4           NCD19        8.930          Low Fat         0.000000
5           FDP36       10.395          Regular         0.000000
6           FDO10       13.650          Regular         0.012741
7           FDP10          NaN          Low Fat         0.127470
8           FDH17       16.200          Regular         0.016687
9           FDU28       19.200          Regular         0.094450

              Item_Type  Item_MRP Outlet_Identifier  \
0                 Dairy  249.8092            OUT049
1           Soft Drinks   48.2692            OUT018
2                  Meat  141.6180            OUT049
3  Fruits and Vegetables  182.0950            OUT010
4             Household   53.8614            OUT013
5          Baking Goods   51.4008            OUT018
6           Snack Foods   57.6588            OUT013
7           Snack Foods  107.7622            OUT027
8          Frozen Foods   96.9726            OUT045
9          Frozen Foods  187.8214            OUT017

   Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                       1999      Medium               Tier 1
1                       2009      Medium               Tier 3
2                       1999      Medium               Tier 1
3                       1998         NaN               Tier 3
4                       1987        High               Tier 3
5                       2009      Medium               Tier 3
6                       1987        High               Tier 3
7                       1985      Medium               Tier 3
8                       2002         NaN               Tier 2
```

```
9                       2007         NaN               Tier 2

         Outlet_Type  Item_Outlet_Sales
0  Supermarket Type1          3735.1380
1  Supermarket Type2           443.4228
2  Supermarket Type1          2097.2700
3      Grocery Store           732.3800
4  Supermarket Type1           994.7052
5  Supermarket Type2           556.6088
6  Supermarket Type1           343.5528
7  Supermarket Type3          4022.7636
8  Supermarket Type1          1076.5986
9  Supermarket Type1          4710.5350
```

checking the datatype of each columns

```
def datainfo():

    temp_ps = pd.DataFrame(index=mart.columns)
    temp_ps['DataType'] = mart.dtypes
    temp_ps["Non-null_Values"] = mart.count()
    temp_ps['Unique_Values'] = mart.nunique()
    temp_ps['NaN_Values'] = mart.isnull().sum()
    temp_ps['NaN_Values_Percentage'] =
(temp_ps['NaN_Values']/len(mart))*100
    return temp_ps

datainfo()
```

```
                           DataType  Non-null_Values  Unique_Values  \
Item_Identifier              object             8523           1559
Item_Weight                 float64             7060            415
Item_Fat_Content             object             8523              5
Item_Visibility             float64             8523           7880
Item_Type                    object             8523             16
Item_MRP                    float64             8523           5938
Outlet_Identifier            object             8523             10
Outlet_Establishment_Year     int64             8523              9
Outlet_Size                  object             6113              3
Outlet_Location_Type         object             8523              3
Outlet_Type                  object             8523              4
Item_Outlet_Sales           float64             8523           3493

                           NaN_Values  NaN_Values_Percentage
Item_Identifier                     0               0.000000
Item_Weight                      1463              17.165317
Item_Fat_Content                    0               0.000000
Item_Visibility                     0               0.000000
Item_Type                           0               0.000000
Item_MRP                            0               0.000000
```

```
Outlet_Identifier                0              0.000000
Outlet_Establishment_Year        0              0.000000
Outlet_Size                   2410             28.276428
Outlet_Location_Type             0              0.000000
Outlet_Type                      0              0.000000
Item_Outlet_Sales                0              0.000000
```

- from above datainfo we find that over data have some null value so we need to drop or filll them with fillna command
- in this datainfo over item_weight does not effect on the dataset
- and the otlet_size matter so we change the data

```
mart["Item_Weight"].fillna(mart["Item_Weight"].mean(),inplace=True)
mart.dropna(inplace=True)
```

```
datainfo()
```

```
                          DataType  Non-null_Values  Unique_Values  \
Item_Identifier             object             6113           1555
Item_Weight                float64             6113            410
Item_Fat_Content            object             6113              5
Item_Visibility            float64             6113           5641
Item_Type                   object             6113             16
Item_MRP                   float64             6113           4694
Outlet_Identifier           object             6113              7
Outlet_Establishment_Year    int64             6113              6
Outlet_Size                 object             6113              3
Outlet_Location_Type        object             6113              3
Outlet_Type                 object             6113              4
Item_Outlet_Sales          float64             6113           3056

                          NaN_Values  NaN_Values_Percentage
Item_Identifier                    0                    0.0
Item_Weight                        0                    0.0
Item_Fat_Content                   0                    0.0
Item_Visibility                    0                    0.0
Item_Type                          0                    0.0
Item_MRP                           0                    0.0
Outlet_Identifier                  0                    0.0
Outlet_Establishment_Year          0                    0.0
Outlet_Size                        0                    0.0
Outlet_Location_Type               0                    0.0
Outlet_Type                        0                    0.0
Item_Outlet_Sales                  0                    0.0
```

checking the shape type of data

```
mart.shape
```

```
(6113, 12)
```

```
mart.describe().T
```

```
                            count         mean          std        min
\
Item_Weight                6113.0    12.888856     4.073798     4.5550

Item_Visibility            6113.0     0.064505     0.050092     0.0000

Item_MRP                   6113.0   141.256859    62.229701    31.2900

Outlet_Establishment_Year  6113.0  1995.794373     8.842615  1985.0000

Item_Outlet_Sales          6113.0  2322.688445  1741.592093    33.9558


                                       25%          50%          75%
max
Item_Weight                       9.800000    12.857645    15.700000
21.350000
Item_Visibility                   0.026681     0.052811     0.092834
0.328391
Item_MRP                         94.012000   143.178600   185.892400
266.888400
Outlet_Establishment_Year  1987.000000  1997.000000  2004.000000
2009.000000
Item_Outlet_Sales           974.731200  1928.156800  3271.075400
13086.964800
```

EDA on that project

```
numerical_feature=mart.select_dtypes("number")
```

```
numerical_feature
```

```
      Item_Weight  Item_Visibility  Item_MRP
Outlet_Establishment_Year  \
0           9.300         0.016047  249.8092
1999
1           5.920         0.019278   48.2692
2009
2          17.500         0.016760  141.6180
1999
4           8.930         0.000000   53.8614
1987
5          10.395         0.000000   51.4008
2009
...           ...              ...       ...                        ..
.
8517       20.750         0.083607  178.8318
1997
```

```
8518        6.865         0.056783  214.5218
1987
8520       10.600         0.035186   85.1224
2004
8521        7.210         0.145221  103.1332
2009
8522       14.800         0.044878   75.4670
1997

      Item_Outlet_Sales
0             3735.1380
1              443.4228
2             2097.2700
4              994.7052
5              556.6088
...                 ...
8517          3608.6360
8518          2778.3834
8520          1193.1136
8521          1845.5976
8522           765.6700

[6113 rows x 5 columns]
```

```
catagorical_feature=mart.select_dtypes("object")
```

```
catagorical_feature
```

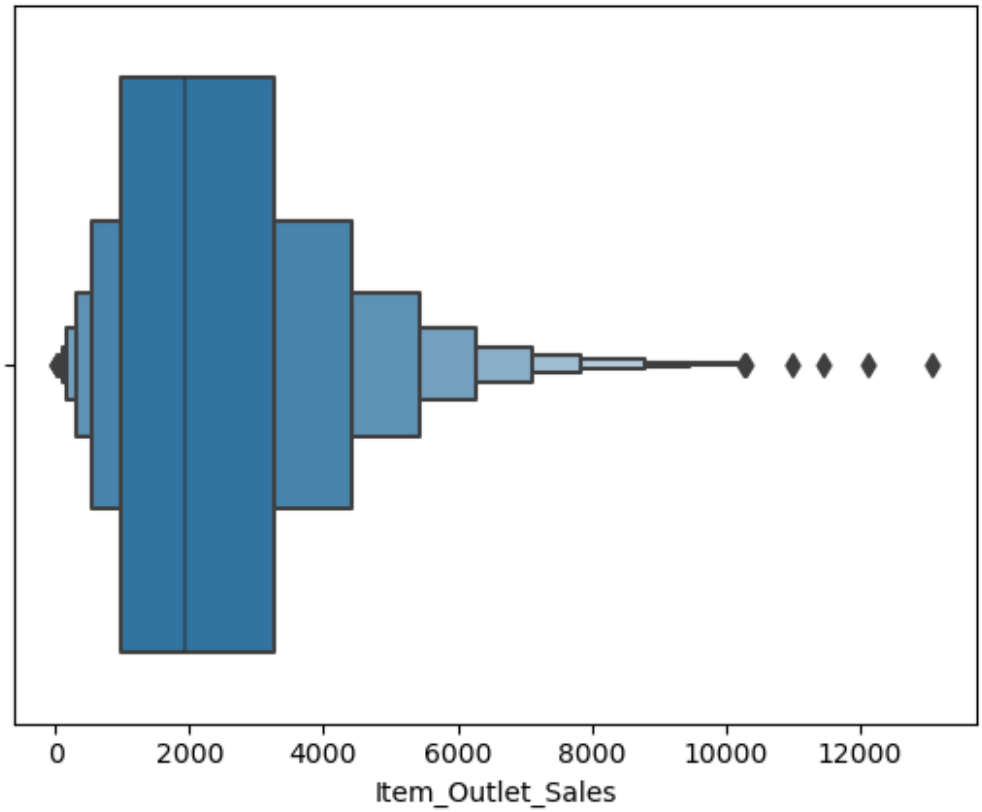
```
     Item_Identifier Item_Fat_Content           Item_Type
Outlet_Identifier  \
0              FDA15          Low Fat               Dairy
OUT049
1              DRC01          Regular         Soft Drinks
OUT018
2              FDN15          Low Fat                Meat
OUT049
4              NCD19          Low Fat           Household
OUT013
5              FDP36          Regular        Baking Goods
OUT018
...              ...              ...                 ...
...
8517           FDF53              reg        Frozen Foods
OUT046
8518           FDF22          Low Fat         Snack Foods
OUT013
8520           NCJ29          Low Fat  Health and Hygiene
OUT035
8521           FDN46          Regular         Snack Foods
OUT018
8522           DRG01          Low Fat         Soft Drinks
```

OUT046

```
     Outlet_Size Outlet_Location_Type        Outlet_Type
0         Medium               Tier 1  Supermarket Type1
1         Medium               Tier 3  Supermarket Type2
2         Medium               Tier 1  Supermarket Type1
4           High               Tier 3  Supermarket Type1
5         Medium               Tier 3  Supermarket Type2
...          ...                  ...                ...
8517       Small               Tier 1  Supermarket Type1
8518        High               Tier 3  Supermarket Type1
8520       Small               Tier 2  Supermarket Type1
8521      Medium               Tier 3  Supermarket Type2
8522       Small               Tier 1  Supermarket Type1

[6113 rows x 7 columns]
```

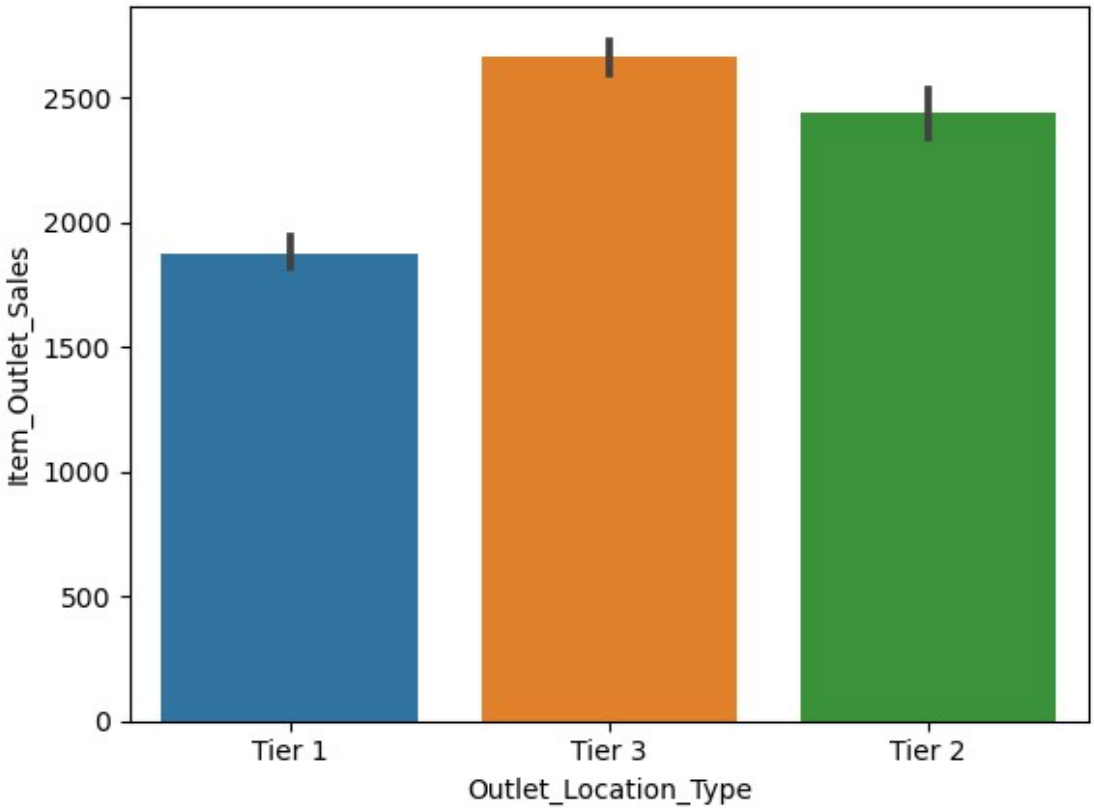
```
sns.boxenplot(data=mart,x="Item_Outlet_Sales")
```

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



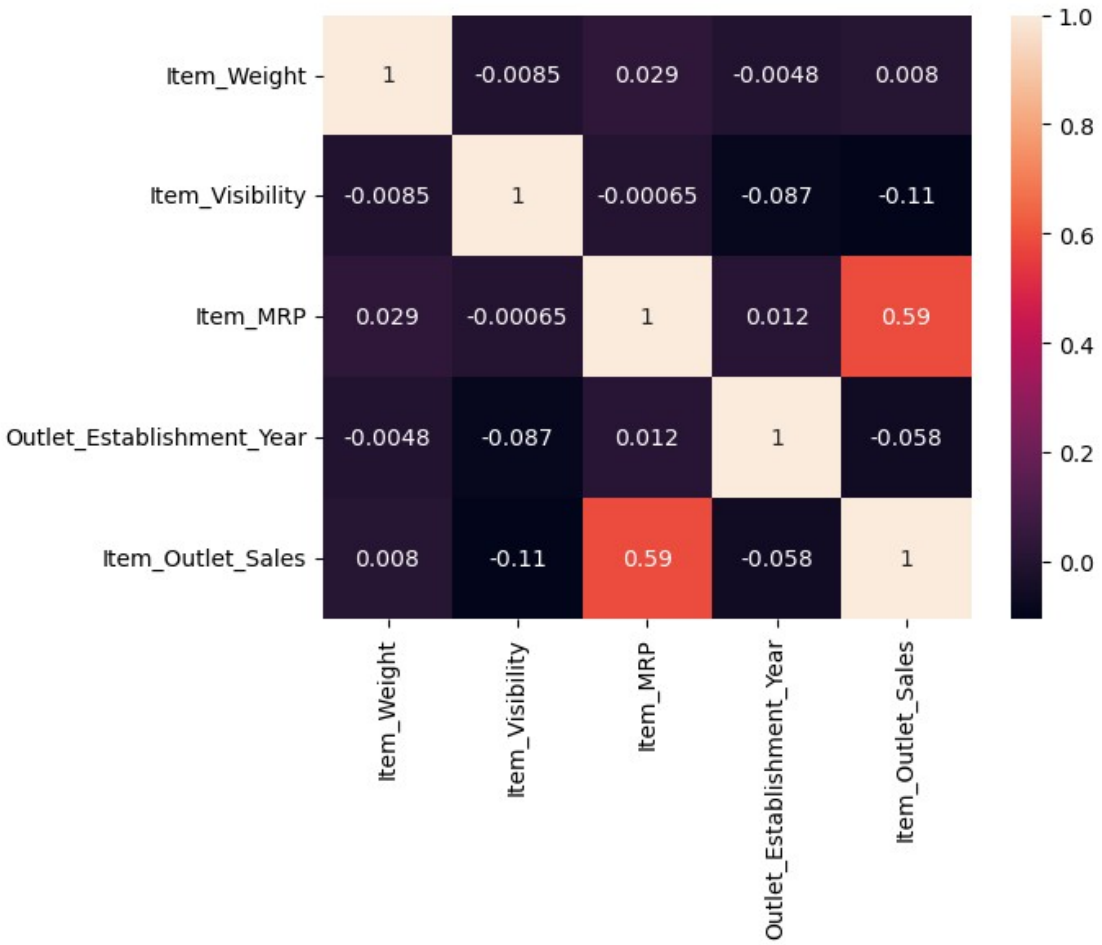

```
sns.barplot(x="Outlet_Location_Type",y='Item_Outlet_Sales',data=mart)
```

```
<AxesSubplot:xlabel='Outlet_Location_Type',
ylabel='Item_Outlet_Sales'>
```

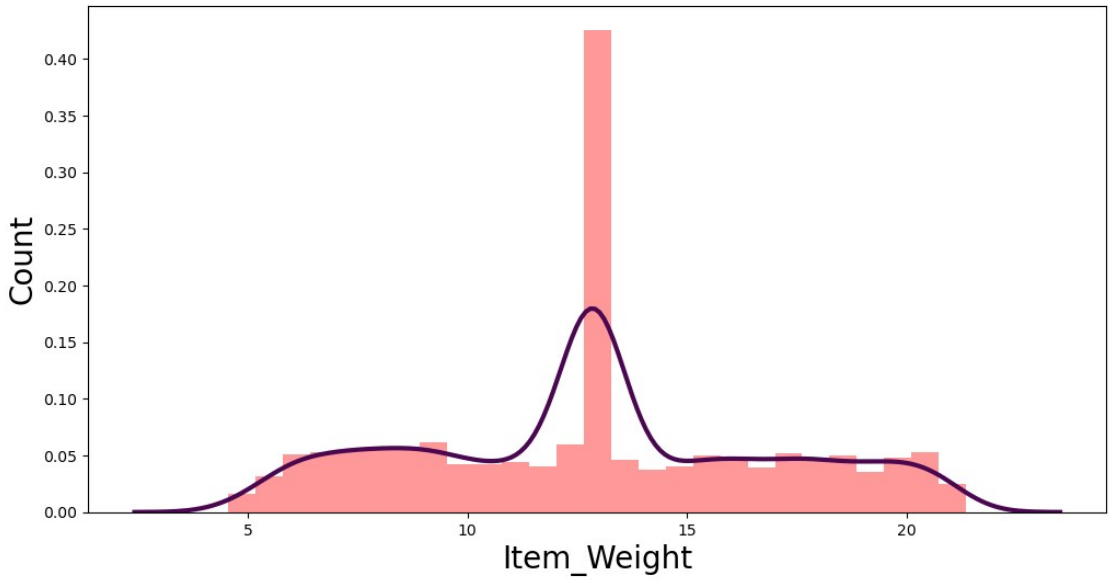


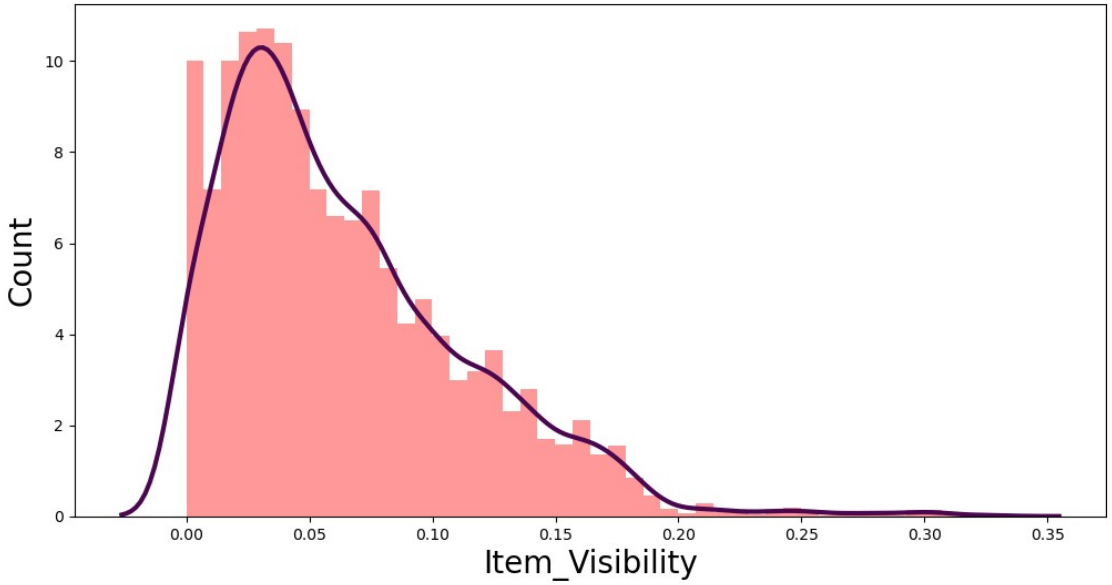

```
sns.heatmap(mart.corr(),annot=True)
```

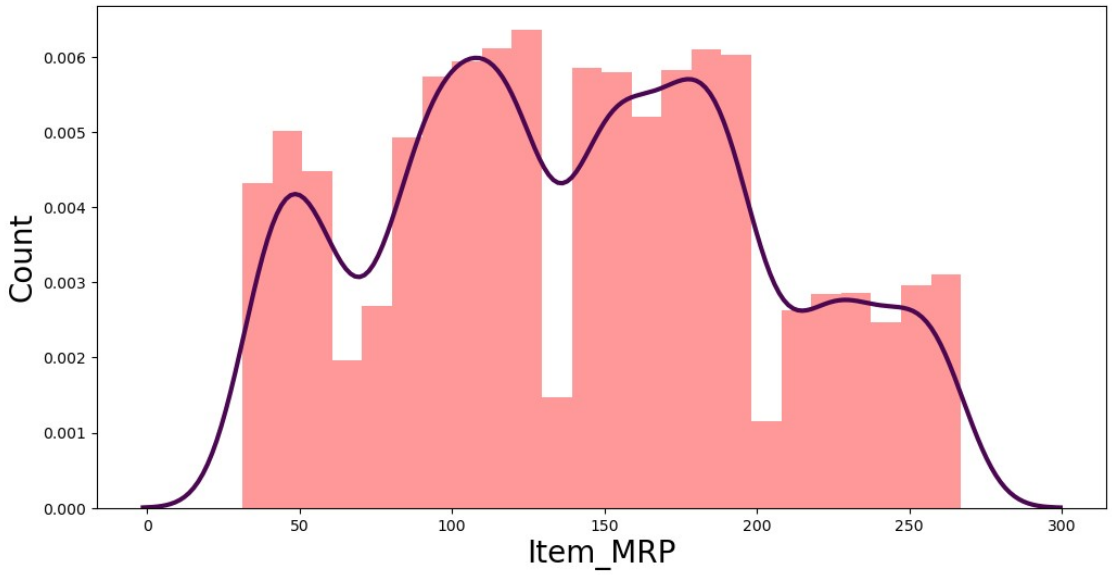
```
<AxesSubplot:>
```

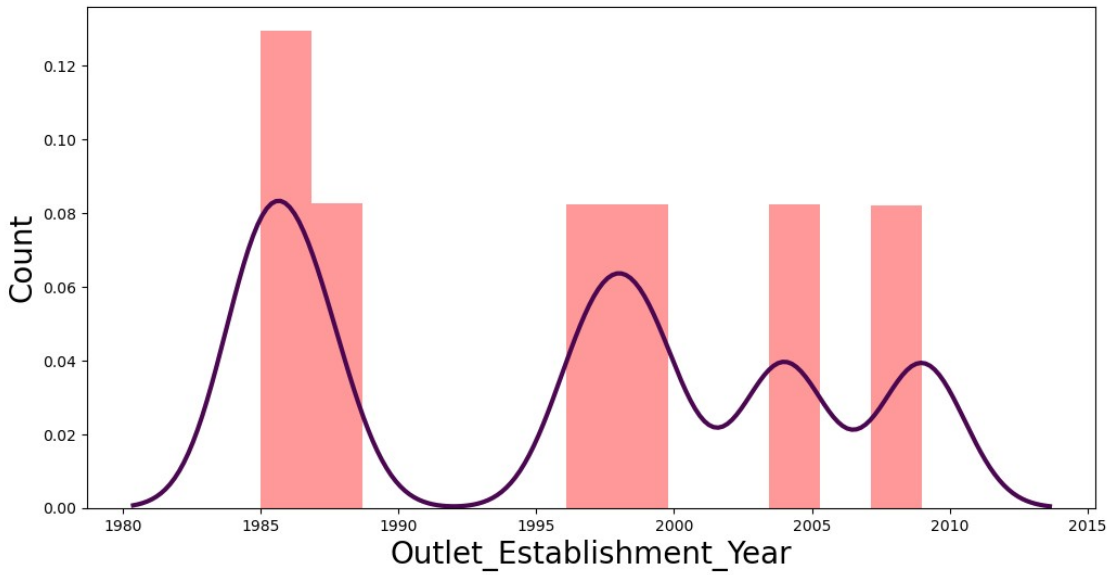


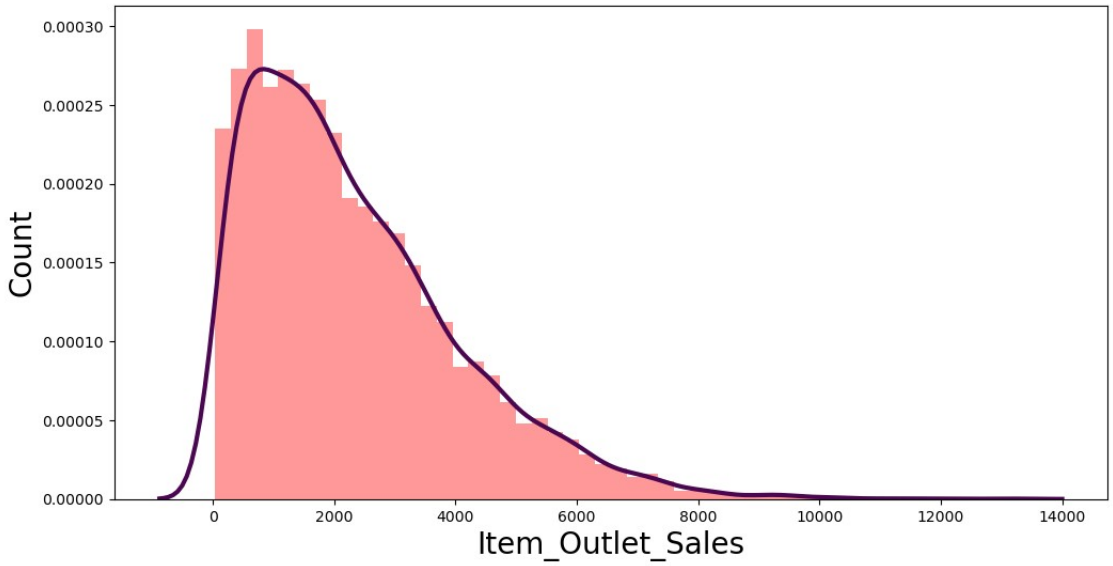

```
import warnings
warnings.filterwarnings("ignore")

for feature in numerical_feature:
    dataset=mart.copy()
    fig, ax = plt.subplots(figsize=(12,6))

sns.distplot(dataset[feature],color='r',kde_kws={'linewidth':3,'color'
:'#4B0751'});
    ax.set_xlabel(feature, fontsize=20)
    ax.set_ylabel("Count", fontsize=20)
```


Count
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
5
10
15
20
Item_Weight


Count
10
8
6
4
2
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Item_Visibility

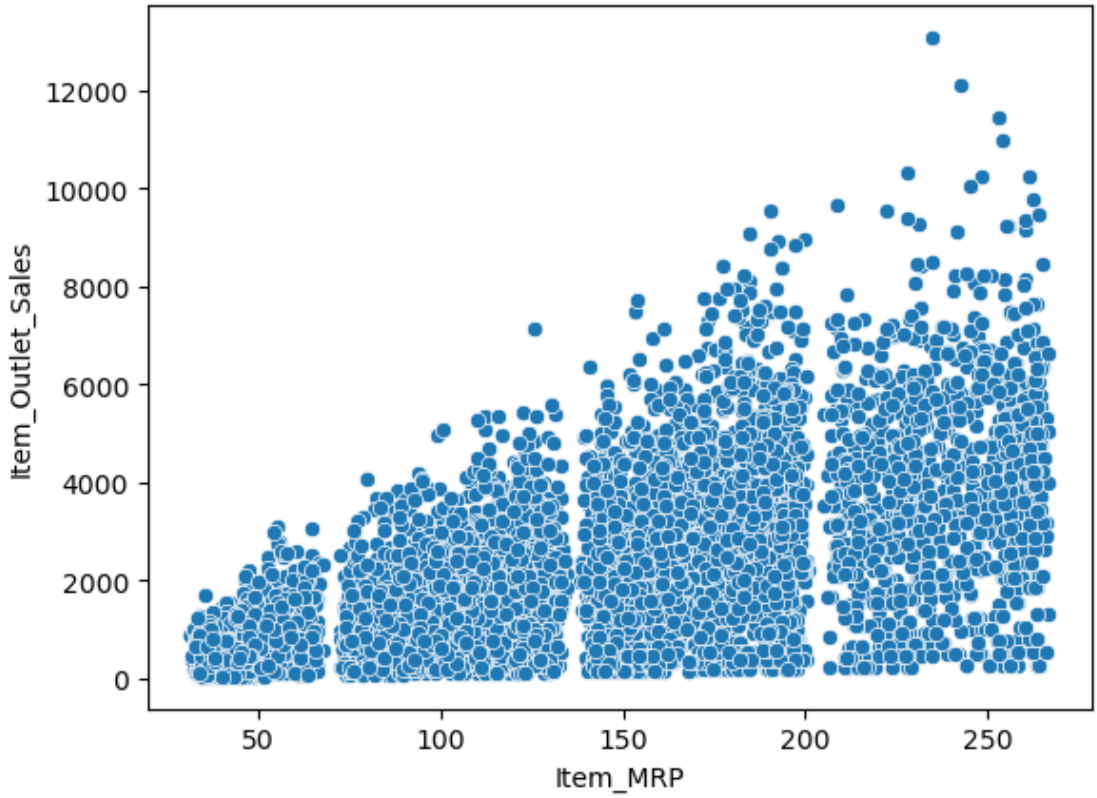

Count
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
50
100
150
200
250
300
Item_MRP


Count
0.12
0.10
0.08
0.06
0.04
0.02
0.00
1980
1985
1990
1995
2000
2005
2010
2015
Outlet_Establishment_Year

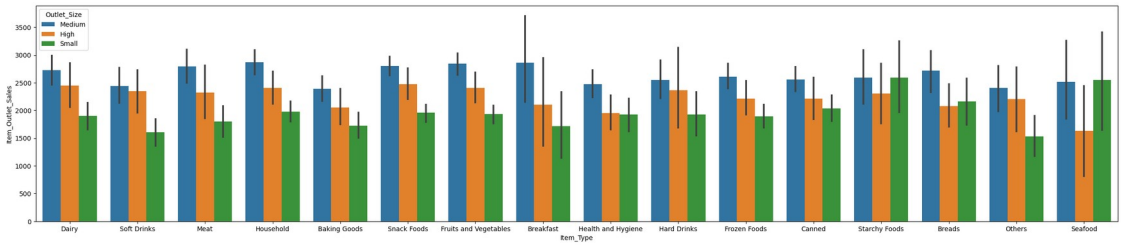



```
sns.scatterplot(x="Item_MRP",y='Item_Outlet_Sales',data=mart)
```

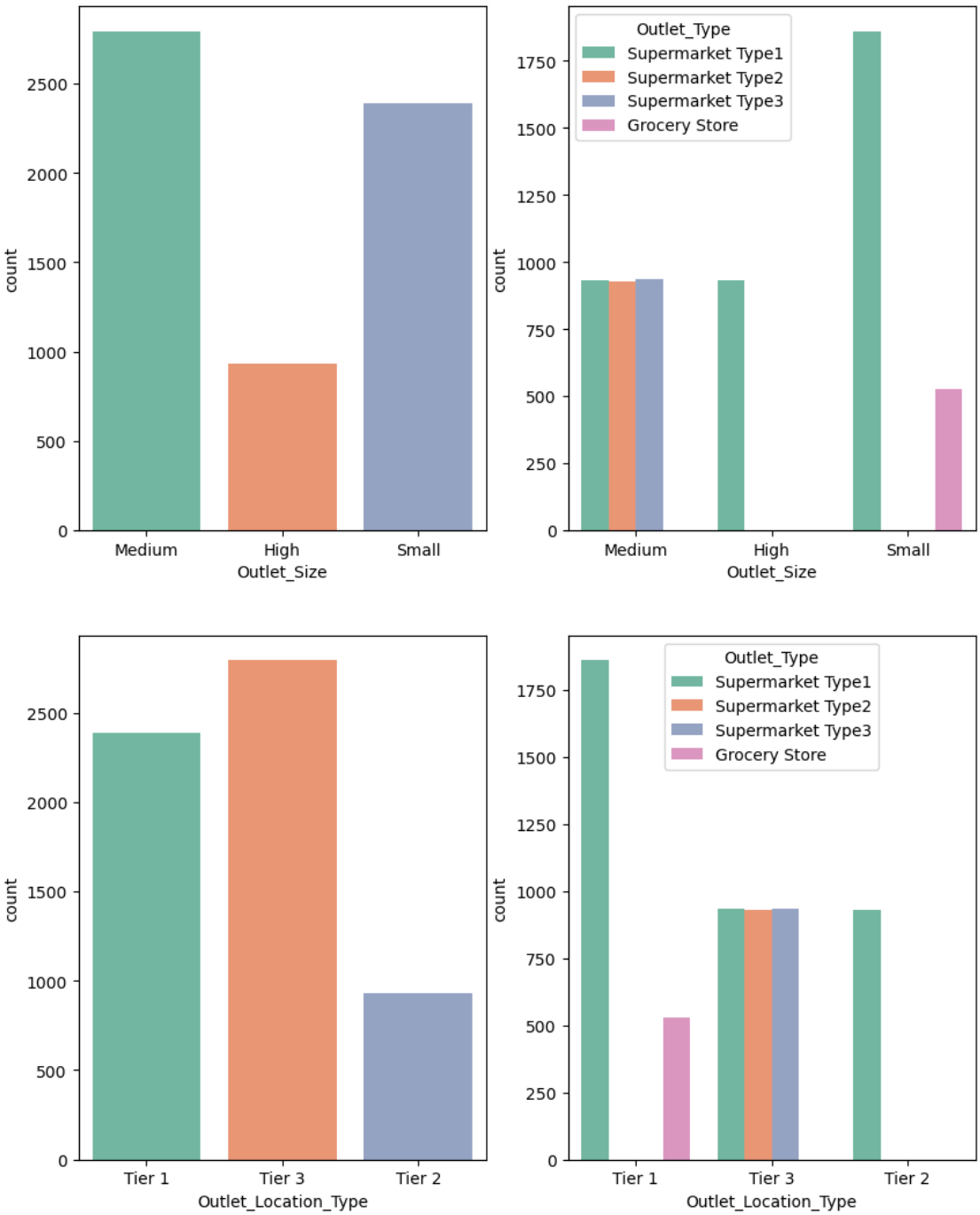
```
<AxesSubplot:xlabel='Item_MRP', ylabel='Item_Outlet_Sales'>
```




```
plt.figure(figsize=(30,6))
sns.barplot(x="Item_Type",y="Item_Outlet_Sales",data=mart,hue="Outlet_
Size")
plt.show()
```

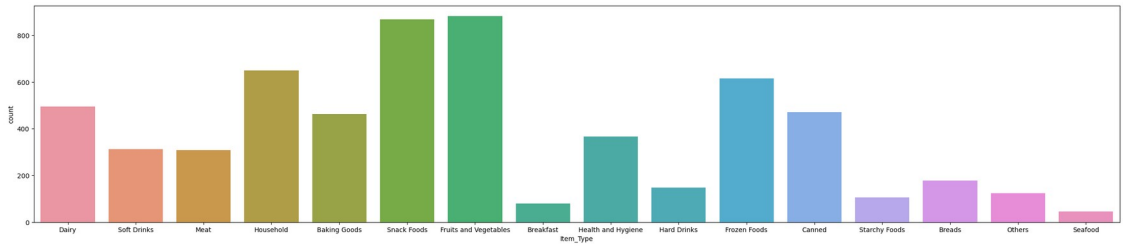



```
plt.figure(figsize=(10,20))
i=1
for col in ['Outlet_Size', 'Outlet_Location_Type']:
    plt.subplot(3,2,i)
    sns.countplot(x=col,data=mart,palette='Set2')
    i+=1
    plt.subplot(3,2,i)
    sns.countplot(x=col,hue='Outlet_Type',data=mart,palette='Set2')
    i+=1
```




```
plt.figure(figsize=(30,6))
sns.countplot(x='Item_Type', data=mart)
```

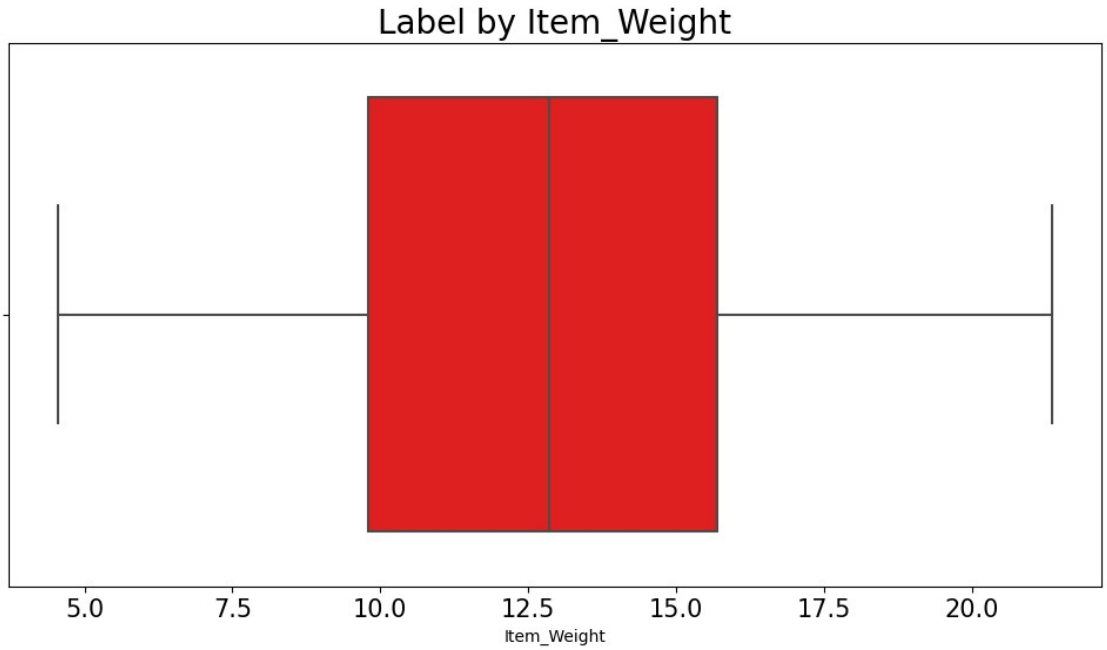
```
<AxesSubplot:xlabel='Item_Type', ylabel='count'>
```

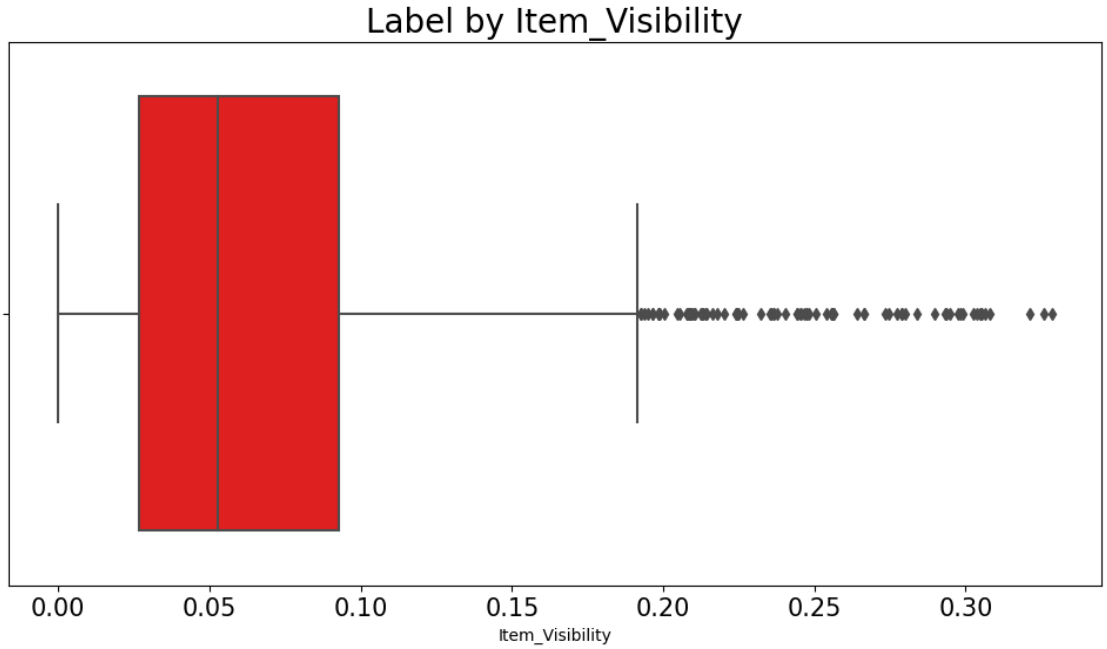


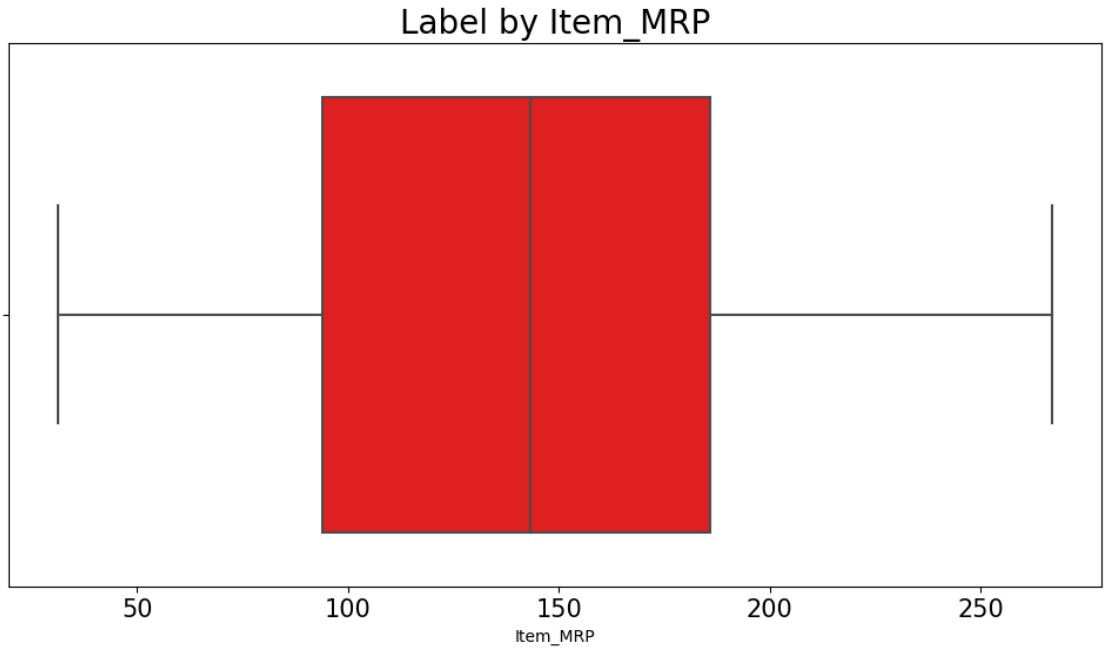
Outliers

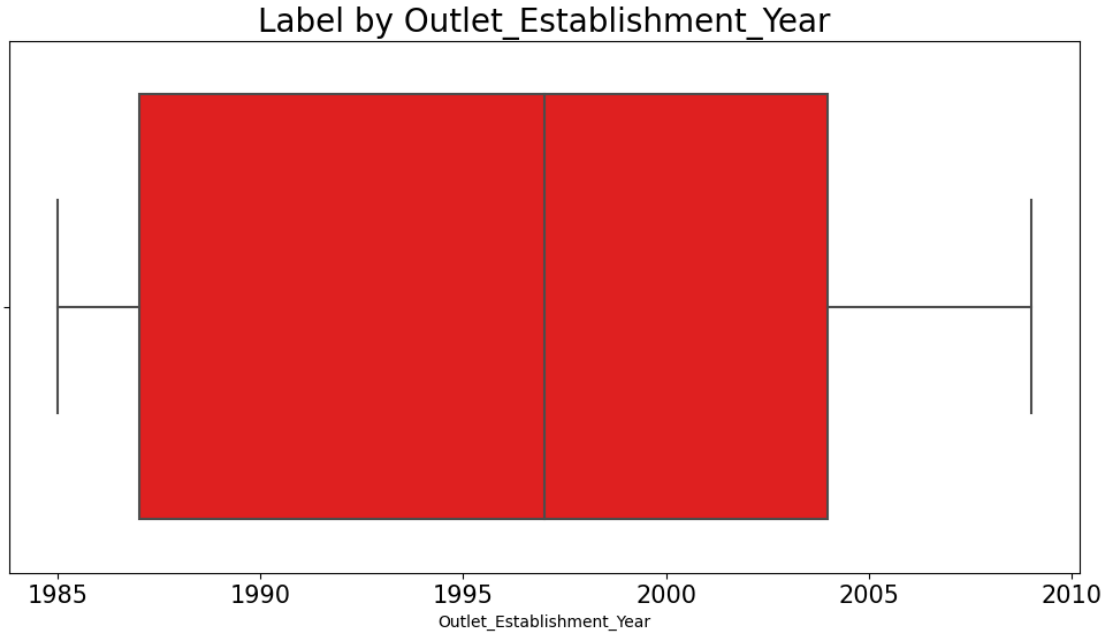
“Outlier Analysis is a process that involves identifying the anomalous observation in the dataset.” Outliers are extreme values that deviates from the other observations in the dataset.

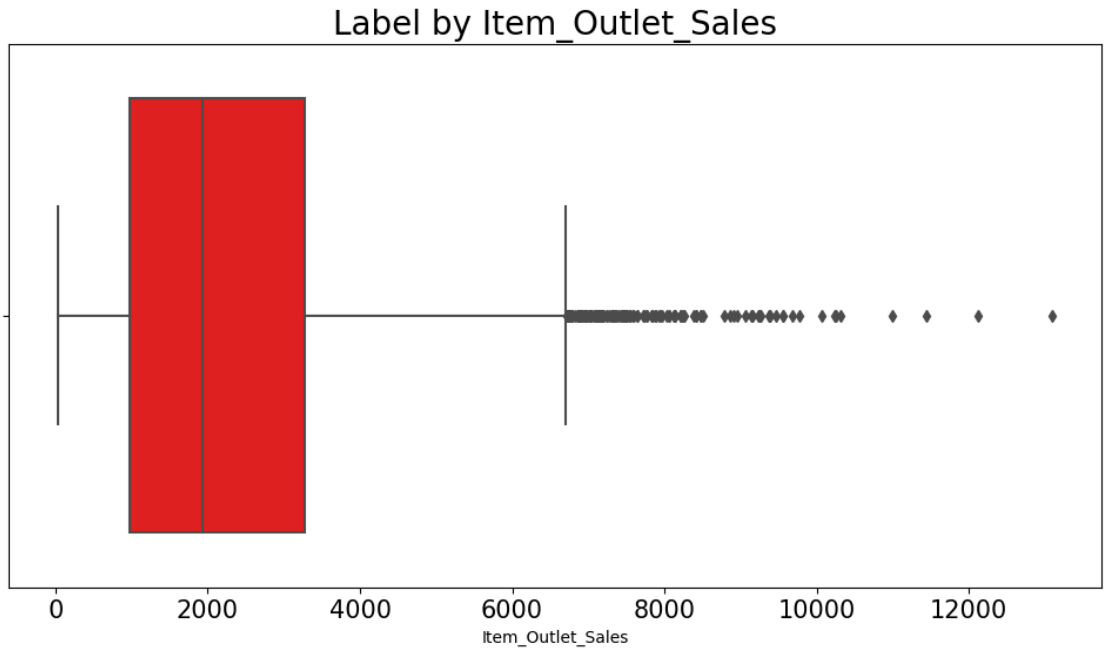
```
for col in dataset.describe().columns:
    fig, ax = plt.subplots(figsize=(12,6))
    sns.boxplot(dataset[col],color='red')
    ax.tick_params(axis='x',labelsize=15)
    ax.tick_params(axis='y',labelsize=15)
    ax.set_title('Label by ' + col, fontsize=20)
plt.show()
```




Label by Item_Visibility
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Item_Visibility


Label by Item_MRP
50
100
150
200
250
Item_MRP





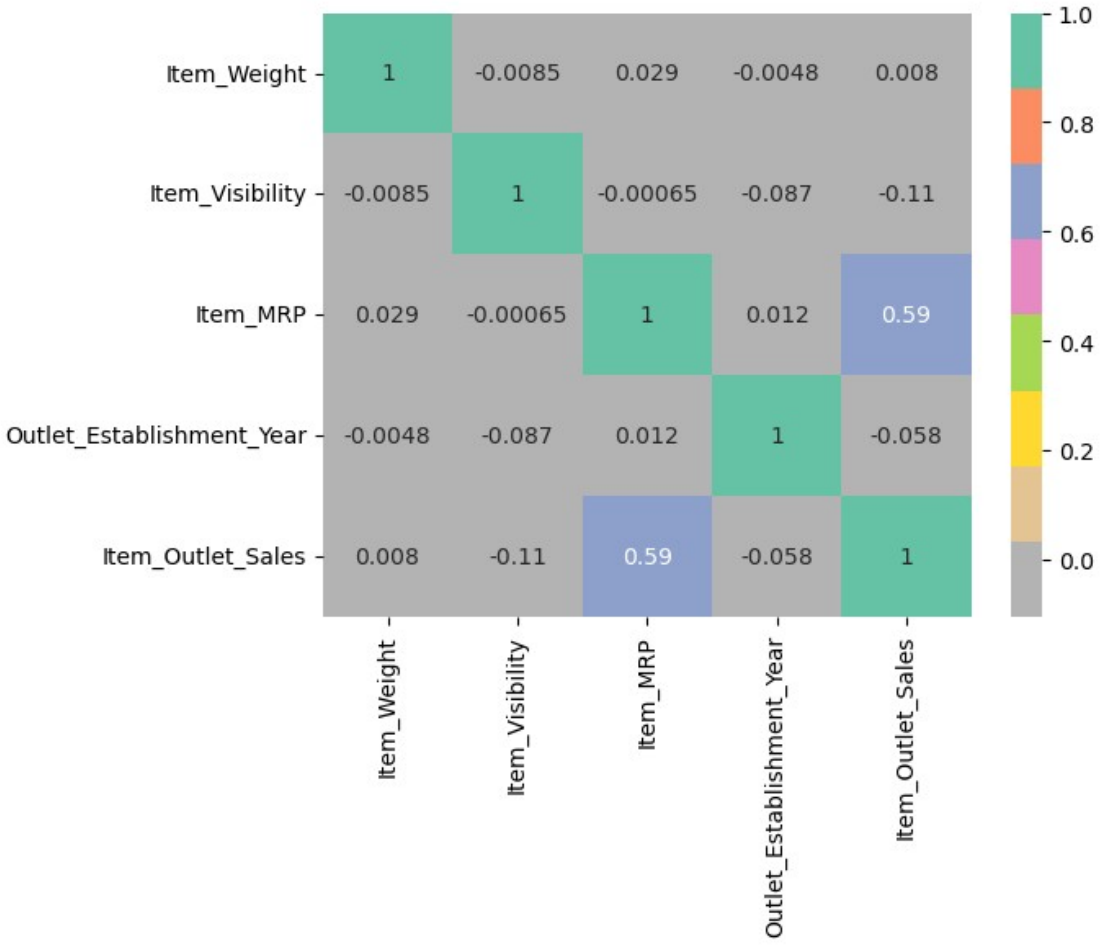

```
mart.isnull().sum()
```

```
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
```

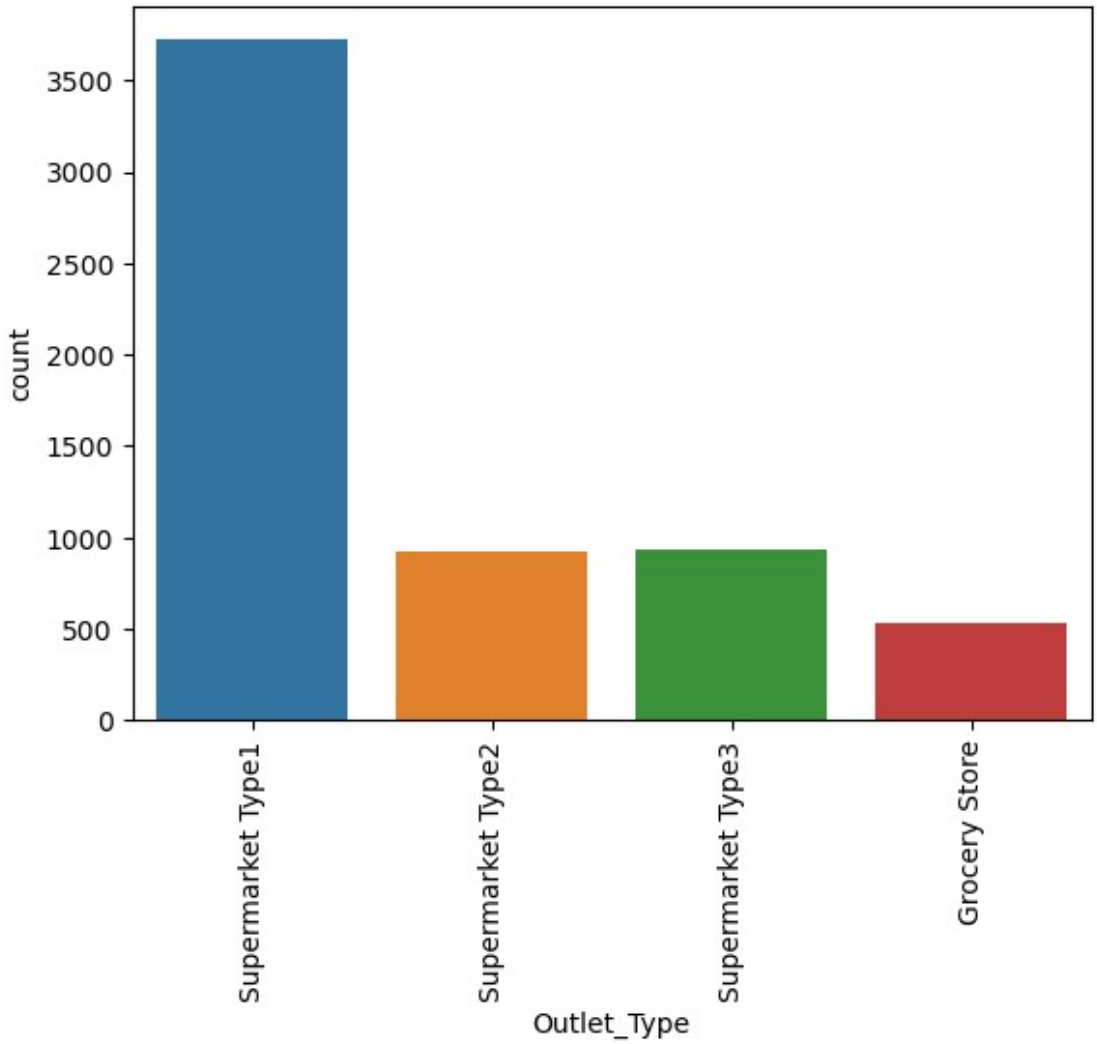
```
Outlet_Location_Type          0
Outlet_Type                   0
Item_Outlet_Sales             0
dtype: int64
```

```
sns.heatmap(numerical_feature.corr(),annot=True,data=mart,cbar=True,xt
icklabels='auto',yticklabels='auto',linecolor='white',linewidths=0,cma
p="Set2_r")
```

```
<AxesSubplot:>
```




```
sns.countplot(data=mart,x='Outlet_Type')
plt.xticks(rotation = 90)
plt.show()
```



grouping the data

```
mart[["Item_Weight","Item_Outlet_Sales"]].groupby(["Item_Weight"]).agg
("median")
```

```
             Item_Outlet_Sales
Item_Weight
4.555                1565.9616
4.590                1358.2320
4.610                1831.6158
4.615                4893.6300
4.635                1927.4910
...                        ...
21.000               1356.2346
21.100               2015.3766
21.200               2954.1546
21.250               2470.1180
21.350               5206.5560
```

[410 rows x 1 columns]

```
mart[["Item_Fat_Content","Item_Outlet_Sales"]].groupby(["Item_Fat_Cont
ent"]).agg("median")
```

Item_Fat_Content	Item_Outlet_Sales
LF	2028.6926
Low Fat	1907.5170
Regular	1974.4299
low fat	1722.4246
reg	1598.5858

```
mart[["Item_Visibility","Item_Outlet_Sales"]].groupby(["Item_Visibilit
y"]).agg("median")
```

Item_Visibility	Item_Outlet_Sales
0.000000	1844.2660
0.003575	3229.7958
0.003589	1691.7978
0.003598	2922.1962
0.003607	1384.1982
...	...
0.306543	291.6204
0.308145	889.5088
0.321115	199.7400
0.325781	761.0094
0.328391	588.5672

[5641 rows x 1 columns]

```
mart[["Item_Type","Item_Outlet_Sales"]].groupby(["Item_Type"]).agg("me
dian")
```

Item_Type	Item_Outlet_Sales
Baking Goods	1701.7848
Breads	2055.9904
Breakfast	1561.9668
Canned	1901.5248
Dairy	1851.5898
Frozen Foods	1808.3128
Fruits and Vegetables	2022.7004
Hard Drinks	1940.1412
Health and Hygiene	1884.2140
Household	2125.2336
Meat	1964.7758
Others	1837.6080
Seafood	2154.1959

```
Snack Foods                    2084.9527
Soft Drinks                    1617.2282
Starchy Foods                  2181.1608
```

```
mart[["Item_MRP","Item_Outlet_Sales"]].groupby(["Item_MRP"]).agg("medi
an")
```

```
          Item_Outlet_Sales
Item_MRP
31.2900            898.8300
31.4900            466.0600
31.8900            366.1900
31.9558            679.1160
32.0558           1018.6740
...                     ...
266.1884          4239.8144
266.2884          2914.8724
266.5884          3974.8260
266.6884          3974.8260
266.8884          5034.7796

[4694 rows x 1 columns]
```

```
mart[["Outlet_Identifier","Item_Outlet_Sales"]].groupby(["Outlet_Ident
ifier"]).agg("median")
```

```
                   Item_Outlet_Sales
Outlet_Identifier
OUT013                       2050.6640
OUT018                       1655.1788
OUT019                        265.3213
OUT027                       3364.9532
OUT035                       2109.2544
OUT046                       1945.8005
OUT049                       1966.1074
```

```
mart[["Outlet_Establishment_Year","Item_Outlet_Sales"]].groupby(["Outl
et_Establishment_Year"]).agg("median")
```

```
                           Item_Outlet_Sales
Outlet_Establishment_Year
1985                                1845.5976
1987                                2050.6640
1997                                1945.8005
1999                                1966.1074
2004                                2109.2544
2009                                1655.1788
```

```
mart[["Outlet_Size","Item_Outlet_Sales"]].groupby(["Outlet_Size"]).agg
("median")
```

```
             Item_Outlet_Sales
Outlet_Size
High                 2050.6640
Medium               2251.0698
Small                1544.6560
```

```
mart[["Outlet_Type","Item_Outlet_Sales"]].groupby(["Outlet_Type"]).agg
("median")
```

```
                   Item_Outlet_Sales
Outlet_Type
Grocery Store               265.3213
Supermarket Type1          2024.0320
Supermarket Type2          1655.1788
Supermarket Type3          3364.9532
```

```
mart.head()
```

```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDA15        9.300          Low Fat         0.016047
1           DRC01        5.920          Regular         0.019278
2           FDN15       17.500          Low Fat         0.016760
4           NCD19        8.930          Low Fat         0.000000
5           FDP36       10.395          Regular         0.000000

      Item_Type  Item_MRP Outlet_Identifier  Outlet_Establishment_Year
\
0         Dairy  249.8092            OUT049                       1999

1   Soft Drinks   48.2692            OUT018                       2009

2          Meat  141.6180            OUT049                       1999

4     Household   53.8614            OUT013                       1987

5  Baking Goods   51.4008            OUT018                       2009


  Outlet_Size Outlet_Location_Type        Outlet_Type
Item_Outlet_Sales
0      Medium               Tier 1  Supermarket Type1
3735.1380
1      Medium               Tier 3  Supermarket Type2
443.4228
2      Medium               Tier 1  Supermarket Type1
2097.2700
4        High               Tier 3  Supermarket Type1
994.7052
5      Medium               Tier 3  Supermarket Type2
556.6088
```

```
sns.heatmap
```

```
<function seaborn.matrix.heatmap(data, *, vmin=None, vmax=None,
cmap=None, center=None, robust=False, annot=None, fmt='.2g',
annot_kws=None, linewidths=0, linecolor='white', cbar=True,
cbar_kws=None, cbar_ax=None, square=False, xticklabels='auto',
yticklabels='auto', mask=None, ax=None, **kwargs)>
```

labelencoding

```
from sklearn.preprocessing import LabelEncoder

Le=LabelEncoder()

mart["Item_Identifier"]=Le.fit_transform(mart["Item_Identifier"])
mart["Item_Fat_Content"]=mart["Item_Fat_Content"].replace({'LF':0,
'Low Fat':1, 'Regular':2, 'low fat':3, 'reg':4})

mart["Outlet_Identifier"]=mart["Outlet_Identifier"].replace({"OUT013":
0,"OUT018":1,"OUT019":2,"OUT027":3,"OUT035":4,"OUT046":5,"OUT049":6})

mart["Item_Type"]=mart["Item_Type"].replace({"Fruits and
Vegetables":0,"Snack Foods":1,"Household":2,"Frozen
Foods":3,"Dairy":4,"Canned":5,"Baking Goods":6,"Health and
Hygiene":7,"Soft Drinks":8,"Meat":8,"Breads":10,"Hard
Drinks":11,"Others":12,"Starchy
Foods":13,"Breakfast":14,"Seafood":15})

mart["Outlet_Size"]=mart["Outlet_Size"].replace({"Small":0,"Medium":1,
"High":2})
mart["Outlet_Location_Type"]=mart["Outlet_Location_Type"].replace({"Ti
er 1":0,"Tier 2":1,"Tier 3":2})
mart["Outlet_Type"]=mart["Outlet_Type"].replace({"Grocery
Store":0,"Supermarket Type1":1,"Supermarket Type2":2,"Supermarket
Type3":3})

mart
```

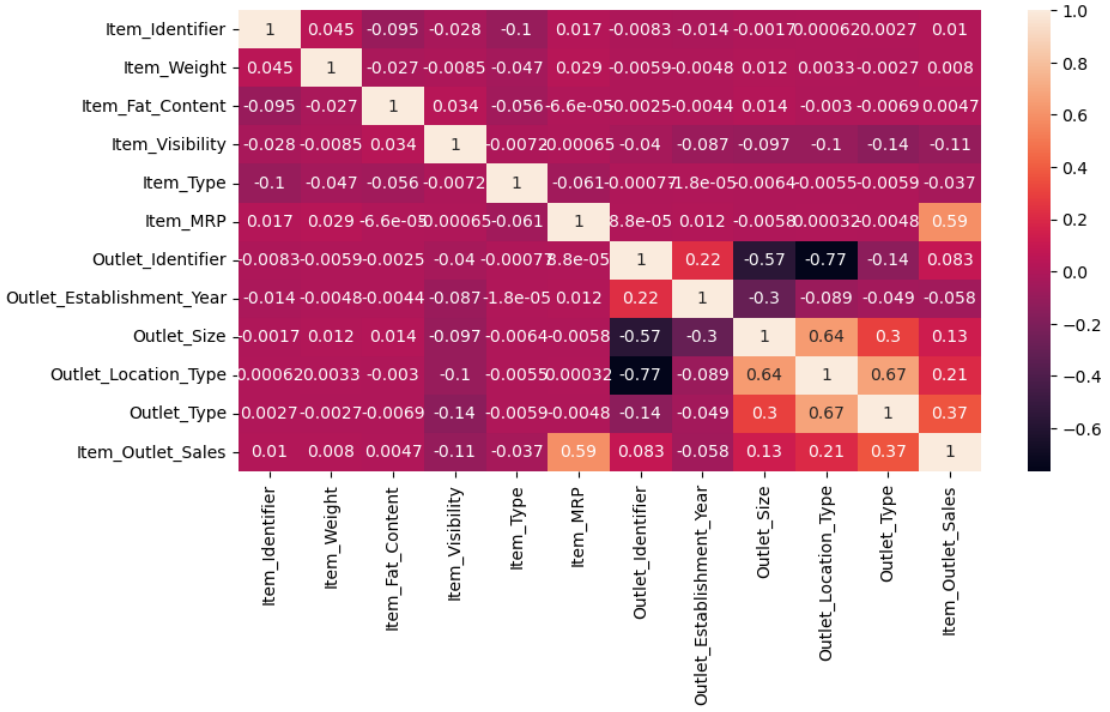
```
      Item_Identifier  Item_Weight  Item_Fat_Content  Item_Visibility
\
0                 155        9.300                 1         0.016047

1                   8        5.920                 2         0.019278

2                 661       17.500                 1         0.016760

4                1294        8.930                 1         0.000000

5                 757       10.395                 2         0.000000

...               ...          ...               ...              ...
```

```
8517              388       20.750                 4         0.083607

8518              369        6.865                 1         0.056783

8520             1354       10.600                 1         0.035186

8521              680        7.210                 2         0.145221

8522               49       14.800                 1         0.044878


      Item_Type  Item_MRP  Outlet_Identifier
Outlet_Establishment_Year  \
0             4  249.8092                  6
1999
1             8   48.2692                  1
2009
2             8  141.6180                  6
1999
4             2   53.8614                  0
1987
5             6   51.4008                  1
2009
...         ...       ...                ...                         ..
.
8517          3  178.8318                  5
1997
8518          1  214.5218                  0
1987
8520          7   85.1224                  4
2004
8521          1  103.1332                  1
2009
8522          8   75.4670                  5
1997

      Outlet_Size  Outlet_Location_Type  Outlet_Type
Item_Outlet_Sales
0               1                     0            1
3735.1380
1               1                     2            2
443.4228
2               1                     0            1
2097.2700
4               2                     2            1
994.7052
5               1                     2            2
556.6088
```

```
...           ...                   ...           ...                  ..
.
8517             0                     0             1
3608.6360
8518             2                     2             1
2778.3834
8520             0                     1             1
1193.1136
8521             1                     2             2
1845.5976
8522             0                     0             1
765.6700

[6113 rows x 12 columns]
```

```
plt.figure(figsize=(10,5))
sns.heatmap(mart.corr(),annot=True)
plt.show
```

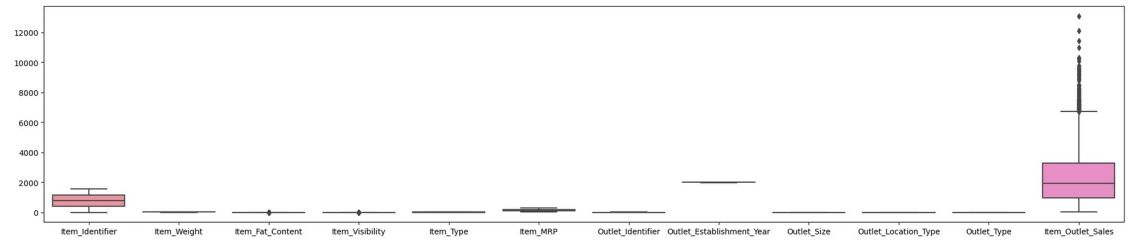
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Checking Outlier !

```
plt.figure(figsize=(25,5))
sns.boxplot(data=mart)
```

```
<AxesSubplot:>
```



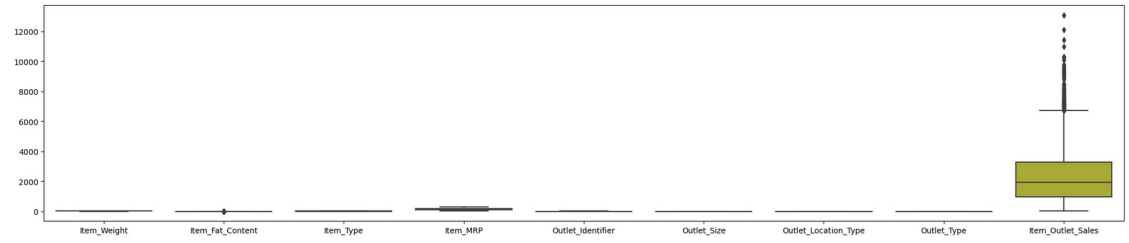

```
mart.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6113 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            6113 non-null   int32  
 1   Item_Weight                6113 non-null   float64
 2   Item_Fat_Content           6113 non-null   int64  
 3   Item_Visibility            6113 non-null   float64
 4   Item_Type                  6113 non-null   int64  
 5   Item_MRP                   6113 non-null   float64
 6   Outlet_Identifier          6113 non-null   int64  
 7   Outlet_Establishment_Year  6113 non-null   int64  
 8   Outlet_Size                6113 non-null   int64  
 9   Outlet_Location_Type       6113 non-null   int64  
 10  Outlet_Type                6113 non-null   int64  
 11  Item_Outlet_Sales          6113 non-null   float64
dtypes: float64(4), int32(1), int64(7)
memory usage: 597.0 KB
```

```
mart.drop(["Item_Identifier","Item_Visibility","Outlet_Establishment_Y
ear"],axis=1,inplace=True)
```

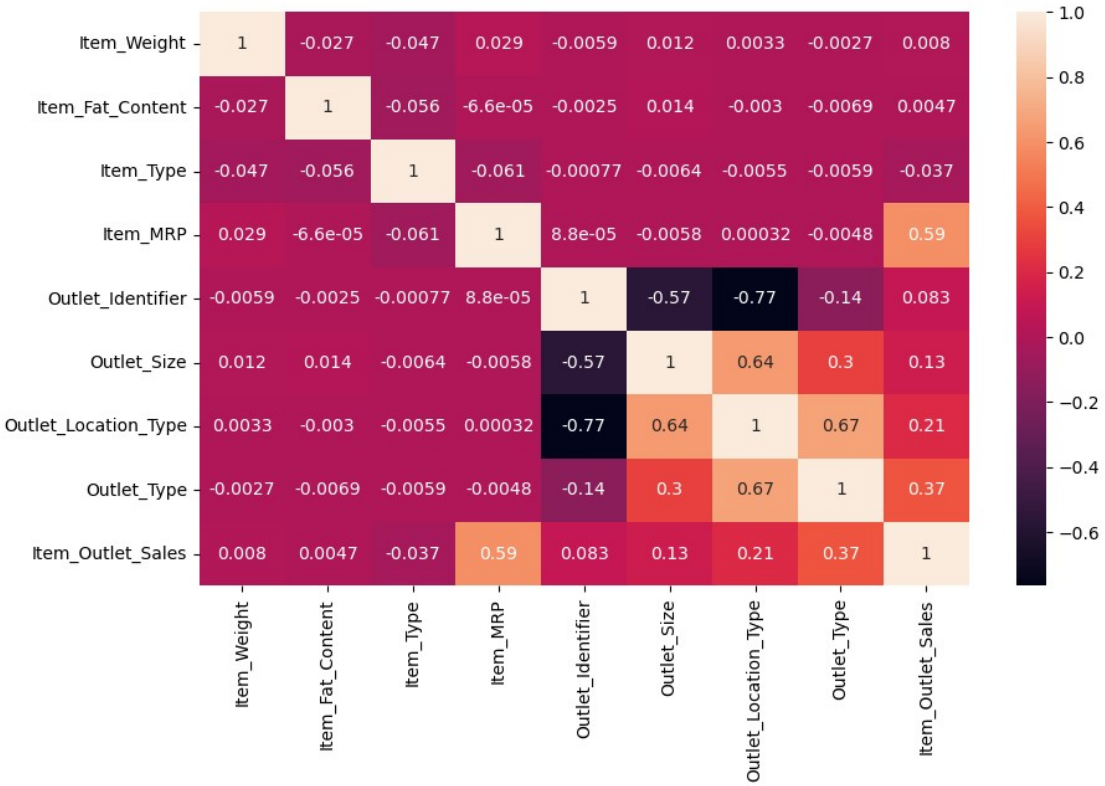
```
plt.figure(figsize=(25,5))
sns.boxplot(data=mart)
```

```
<AxesSubplot:>
```




```
plt.figure(figsize=(10,6))
sns.heatmap(mart.corr(),annot=True)
```

```
<AxesSubplot:>
```




```
X=mart.drop("Item_Outlet_Sales",axis=1)
Y=mart["Item_Outlet_Sales"]

test_X=X.iloc[0:500]
test_Y=Y.iloc[0:500]
#testing only first 500

from sklearn.metrics import r2_score

score={}
```

RANDOM FOREST REGRESSOR

```
from sklearn.ensemble import RandomForestRegressor

rf=RandomForestRegressor(n_estimators=500)

rf.fit(X,Y)

RandomForestRegressor(n_estimators=500)

rf_pred=rf.predict(test_X)

r2_score(test_Y,rf_pred)

0.939023238479339
```

XGBOOST

```
# pip install xgboost

import xgboost as xg

xgb=xg.XGBRegressor(n_estimators=500)

xgb.fit(X,Y)

XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None,
feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None,
importance_type=None,
             interaction_constraints=None, learning_rate=None,
max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan,
monotone_constraints=None,
             n_estimators=500, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...)

xgb_pred=xgb.predict(test_X)

r2_score(test_Y,xgb_pred)

0.9774015892942607

import pickle

 pickle.dump(xgb,open(r"model.pkl","wb"))

import sklearn

import streamlit as st

# pip install streamlit

Requirement already satisfied: streamlit in c:\programdata\anaconda3\
lib\site-packages (1.22.0)Note: you may need to restart the kernel to
use updated packages.

WARNING: Ignoring invalid distribution -rotobuf (c:\programdata\
anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\programdata\
anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\programdata\
anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\programdata\
anaconda3\lib\site-packages)
```

```
WARNING: Ignoring invalid distribution -rotobuf (c:\programdata\
anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -rotobuf (c:\programdata\
anaconda3\lib\site-packages)


Requirement already satisfied: pyarrow>=4.0 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (12.0.0)
Requirement already satisfied: gitpython!=3.1.19 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (3.1.31)
Requirement already satisfied: pympler>=0.9 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (1.0.1)
Requirement already satisfied: rich>=10.11.0 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (13.3.5)
Requirement already satisfied: validators>=0.2 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (0.20.0)
Requirement already satisfied: importlib-metadata>=1.4 in c:\
programdata\anaconda3\lib\site-packages (from streamlit) (4.11.3)
Requirement already satisfied: tzlocal>=1.1 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (5.0.1)
Requirement already satisfied: typing-extensions>=3.10.0.0 in c:\
programdata\anaconda3\lib\site-packages (from streamlit) (4.3.0)
Requirement already satisfied: tenacity<9,>=8.0.0 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (8.0.1)
Requirement already satisfied: pillow>=6.2.0 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (9.2.0)
Requirement already satisfied: protobuf<4,>=3.12 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (3.20.3)
Requirement already satisfied: cachetools>=4.0 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (5.3.0)
Requirement already satisfied: toml in c:\programdata\anaconda3\lib\
site-packages (from streamlit) (0.10.2)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\
site-packages (from streamlit) (1.23.5)
Requirement already satisfied: blinker>=1.0.0 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (1.6.2)
Requirement already satisfied: tornado>=6.0.3 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (6.1)
Requirement already satisfied: pandas<3,>=0.25 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (1.4.4)
Requirement already satisfied: click>=7.0 in c:\programdata\anaconda3\
lib\site-packages (from streamlit) (8.0.4)
Requirement already satisfied: python-dateutil in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (2.8.2)
Requirement already satisfied: requests>=2.4 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (2.28.1)
Requirement already satisfied: watchdog in c:\programdata\anaconda3\
lib\site-packages (from streamlit) (2.1.6)
Requirement already satisfied: altair<5,>=3.2.0 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (4.2.2)
```

```
Requirement already satisfied: packaging>=14.1 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (21.3)
Requirement already satisfied: pydeck>=0.1.dev5 in c:\programdata\
anaconda3\lib\site-packages (from streamlit) (0.8.1b0)
Requirement already satisfied: jinja2 in c:\programdata\anaconda3\lib\
site-packages (from altair<5,>=3.2.0->streamlit) (2.11.3)
Requirement already satisfied: entrypoints in c:\programdata\
anaconda3\lib\site-packages (from altair<5,>=3.2.0->streamlit) (0.4)
Requirement already satisfied: toolz in c:\programdata\anaconda3\lib\
site-packages (from altair<5,>=3.2.0->streamlit) (0.11.2)
Requirement already satisfied: jsonschema>=3.0 in c:\programdata\
anaconda3\lib\site-packages (from altair<5,>=3.2.0->streamlit)
(4.16.0)
Requirement already satisfied: colorama in c:\programdata\anaconda3\
lib\site-packages (from click>=7.0->streamlit) (0.4.5)
Requirement already satisfied: gitdb<5,>=4.0.1 in c:\programdata\
anaconda3\lib\site-packages (from gitpython!=3.1.19->streamlit)
(4.0.10)
Requirement already satisfied: zipp>=0.5 in c:\programdata\anaconda3\
lib\site-packages (from importlib-metadata>=1.4->streamlit) (3.8.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\
programdata\anaconda3\lib\site-packages (from packaging>=14.1-
>streamlit) (3.0.9)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\
anaconda3\lib\site-packages (from pandas<3,>=0.25->streamlit) (2022.1)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\
lib\site-packages (from python-dateutil->streamlit) (1.16.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\
programdata\anaconda3\lib\site-packages (from requests>=2.4-
>streamlit) (1.26.11)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\
programdata\anaconda3\lib\site-packages (from requests>=2.4-
>streamlit) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\
anaconda3\lib\site-packages (from requests>=2.4->streamlit) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\
anaconda3\lib\site-packages (from requests>=2.4->streamlit)
(2022.9.14)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\
programdata\anaconda3\lib\site-packages (from rich>=10.11.0-
>streamlit) (2.15.1)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in c:\
programdata\anaconda3\lib\site-packages (from rich>=10.11.0-
>streamlit) (2.2.0)
Requirement already satisfied: tzdata in c:\programdata\anaconda3\lib\
site-packages (from tzlocal>=1.1->streamlit) (2023.3)
Requirement already satisfied: decorator>=3.4.0 in c:\programdata\
anaconda3\lib\site-packages (from validators>=0.2->streamlit) (5.1.1)
Requirement already satisfied: smmap<6,>=3.0.1 in c:\programdata\
anaconda3\lib\site-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19-
```

```
>streamlit) (5.0.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\programdata\
anaconda3\lib\site-packages (from jinja2->altair<5,>=3.2.0->streamlit)
(2.0.1)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!
=0.17.2,>=0.14.0 in c:\programdata\anaconda3\lib\site-packages (from
jsonschema>=3.0->altair<5,>=3.2.0->streamlit) (0.18.0)
Requirement already satisfied: attrs>=17.4.0 in c:\programdata\
anaconda3\lib\site-packages (from jsonschema>=3.0->altair<5,>=3.2.0-
>streamlit) (21.4.0)
Requirement already satisfied: mdurl~=0.1 in c:\programdata\anaconda3\
lib\site-packages (from markdown-it-py<3.0.0,>=2.2.0->rich>=10.11.0-
>streamlit) (0.1.2)
```