BigMart Sales Prediction using ML

August 17, 2023

[138]:
```
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
```

Situation: In the present times, shopping malls and large retail stores (Big Marts) employ a strategy of monitoring individual item sales information. This approach assists in predicting future customer demand and making necessary adjustments to inventory management. This valuable data is stored within a data warehouse, encompassing a substantial volume of consumer details and specific item information. Extracting insights from this data repository through data mining allows for the identification of anomalies and regular patterns.

Task: The task at hand involves creating a solution capable of forecasting the sales of various branches of Big Mart. This prediction is to be accomplished using the provided dataset.

[139]:
```
df_train = pd.read_csv(r'C:\Users\aayus\Desktop\Project\train.csv')
df_test = pd.read_csv(r'C:\Users\aayus\Desktop\Project\test.csv')
```

[140]:
```
df_train
```

[140]:

	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]
```

[141]:
```
df_test
```

[141]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0              FDW58       20.750          Low Fat         0.007565
1              FDW14        8.300              reg         0.038428
2              NCN55       14.600          Low Fat         0.099575
3              FDQ58        7.315          Low Fat         0.015388
4              FDY38          NaN          Regular         0.118599
…                  …            …                …                …
5676           FDB58       10.500          Regular         0.013496
5677           FDD47        7.600          Regular         0.142991
```

```
5678           NCO17       10.000          Low Fat         0.073529
5679           FDJ26       15.300          Regular         0.000000
5680           FDU37        9.500          Regular         0.104720

                Item_Type  Item_MRP Outlet_Identifier  \
0            Snack Foods  107.8622            OUT049
1                  Dairy   87.3198            OUT017
2                 Others  241.7538            OUT010
3            Snack Foods  155.0340            OUT017
4                  Dairy  234.2300            OUT027
…                      …       …                …
5676         Snack Foods  141.3154            OUT046
5677       Starchy Foods  169.1448            OUT018
5678  Health and Hygiene  118.7440            OUT045
5679              Canned  214.6218            OUT017
5680              Canned   79.7960            OUT045

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                          1999      Medium               Tier 1
1                          2007         NaN               Tier 2
2                          1998         NaN               Tier 3
3                          2007         NaN               Tier 2
4                          1985      Medium               Tier 3
…                             …          …                    …
5676                       1997       Small               Tier 1
5677                       2009      Medium               Tier 3
5678                       2002         NaN               Tier 2
5679                       2007         NaN               Tier 2
5680                       2002         NaN               Tier 2

            Outlet_Type
0     Supermarket Type1
1     Supermarket Type1
2         Grocery Store
3     Supermarket Type1
4     Supermarket Type3
…                     …
5676  Supermarket Type1
5677  Supermarket Type2
5678  Supermarket Type1
5679  Supermarket Type1
5680  Supermarket Type1

[5681 rows x 11 columns]
```

[142]:
```
df_train.head()
```

[142]:
```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDA15         9.30          Low Fat         0.016047
1           DRC01         5.92          Regular         0.019278
2           FDN15        17.50          Low Fat         0.016760
3           FDX07        19.20          Regular         0.000000
4           NCD19         8.93          Low Fat         0.000000

              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

   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

        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
```

[143]:
```
df_train.isnull().sum()
```

[143]:
```
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
```

[144]:
```
df_train.info()
```

```
<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
```

[145]:
```
df_train.describe()
```

[145]:
```
       Item_Weight  Item_Visibility     Item_MRP  Outlet_Establishment_Year  \
count  7060.000000      8523.000000  8523.000000                8523.000000
mean     12.857645         0.066132   140.992782                1997.831867
std       4.643456         0.051598    62.275067                   8.371760
min       4.555000         0.000000    31.290000                1985.000000
25%       8.773750         0.026989    93.826500                1987.000000
50%      12.600000         0.053931   143.012800                1999.000000
75%      16.850000         0.094585   185.643700                2004.000000
max      21.350000         0.328391   266.888400                2009.000000

       Item_Outlet_Sales
count        8523.000000
mean         2181.288914
std          1706.499616
min            33.290000
25%           834.247400
50%          1794.331000
75%          3101.296400
max         13086.964800
```

[146]:
```
df_train['Item_Weight'].describe()
```

[146]:
```
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
```

[147]:
```
df_train['Item_Weight'].fillna(df_train['Item_Weight'].mean(), inplace=True)
df_test['Item_Weight'].fillna(df_test['Item_Weight'].mean(), inplace=True)
```

[148]:
```
df_train.isnull().sum()
```

[148]:
```
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
dtype: int64
```

[149]:
```
df_train['Outlet_Size']
```

[149]:
```
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
```

[150]:
```
df_train['Outlet_Size'].value_counts()
```

[150]:
```
Medium    2793
Small     2388
High       932
Name: Outlet_Size, dtype: int64
```

[151]:
```
df_train['Outlet_Size'].mode()
```

[151]:
```
0    Medium
Name: Outlet_Size, dtype: object
```

[152]:
```
df_train['Outlet_Size'].fillna(df_train['Outlet_Size'].mode()[0],inplace=True)
df_test['Outlet_Size'].fillna(df_test['Outlet_Size'].mode()[0],inplace=True)
```

[153]:
```
df_train
```

[153]:

	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	Medium	Tier 3	
4	1987	High	Tier 3	
…	…	…	…	
8518	1987	High	Tier 3	
8519	2002	Medium	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]
```

[154]:
```
df_test
```

[154]:
```
     Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0              FDW58    20.750000          Low Fat         0.007565
1              FDW14     8.300000              reg         0.038428
2              NCN55    14.600000          Low Fat         0.099575
3              FDQ58     7.315000          Low Fat         0.015388
4              FDY38    12.695633          Regular         0.118599
…                  …           …                …                …
5676           FDB58    10.500000          Regular         0.013496
5677           FDD47     7.600000          Regular         0.142991
5678           NCO17    10.000000          Low Fat         0.073529
5679           FDJ26    15.300000          Regular         0.000000
5680           FDU37     9.500000          Regular         0.104720

                Item_Type  Item_MRP Outlet_Identifier  \
0             Snack Foods  107.8622            OUT049
1                   Dairy   87.3198            OUT017
2                  Others  241.7538            OUT010
3             Snack Foods  155.0340            OUT017
4                   Dairy  234.2300            OUT027
…                       …       …                 …
5676          Snack Foods  141.3154            OUT046
5677        Starchy Foods  169.1448            OUT018
5678  Health and Hygiene   118.7440            OUT045
5679               Canned  214.6218            OUT017
5680               Canned   79.7960            OUT045

      Outlet_Establishment_Year Outlet_Size Outlet_Location_Type  \
0                          1999      Medium               Tier 1
1                          2007      Medium               Tier 2
2                          1998      Medium               Tier 3
```

```
3                         2007      Medium               Tier 2
4                         1985      Medium               Tier 3
…                          …          …                   …
5676                      1997       Small               Tier 1
5677                      2009      Medium               Tier 3
5678                      2002      Medium               Tier 2
5679                      2007      Medium               Tier 2
5680                      2002      Medium               Tier 2

           Outlet_Type
0     Supermarket Type1
1     Supermarket Type1
2         Grocery Store
3     Supermarket Type1
4     Supermarket Type3
…                  …
5676  Supermarket Type1
5677  Supermarket Type2
5678  Supermarket Type1
5679  Supermarket Type1
5680  Supermarket Type1

[5681 rows x 11 columns]
```

[155]:
```
df_train.head()
```

[155]:
```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDA15         9.30          Low Fat         0.016047
1           DRC01         5.92          Regular         0.019278
2           FDN15        17.50          Low Fat         0.016760
3           FDX07        19.20          Regular         0.000000
4           NCD19         8.93          Low Fat         0.000000

              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

   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      Medium               Tier 3
4                       1987        High               Tier 3
```

```
         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
```

[156]:
```
df_train.isnull().sum()
```

[156]:
```
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
```

[157]:
```
df_test.isnull().sum()
```

[157]:
```
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
dtype: int64
```

[158]:
```
df_train
```

[158]:
```
     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      Medium               Tier 3
4                          1987        High               Tier 3
…                          …         …                  …
8518                       1987        High               Tier 3
8519                       2002      Medium               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]
```

[159]:
```
df_test
```

[159]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	\
0	FDW58	20.750000	Low Fat	0.007565	
1	FDW14	8.300000	reg	0.038428	
2	NCN55	14.600000	Low Fat	0.099575	
3	FDQ58	7.315000	Low Fat	0.015388	
4	FDY38	12.695633	Regular	0.118599	
…	…	…	…	…	
5676	FDB58	10.500000	Regular	0.013496	
5677	FDD47	7.600000	Regular	0.142991	
5678	NCO17	10.000000	Low Fat	0.073529	
5679	FDJ26	15.300000	Regular	0.000000	
5680	FDU37	9.500000	Regular	0.104720	

	Item_Type	Item_MRP	Outlet_Identifier	\
0	Snack Foods	107.8622	OUT049	
1	Dairy	87.3198	OUT017	
2	Others	241.7538	OUT010	
3	Snack Foods	155.0340	OUT017	
4	Dairy	234.2300	OUT027	
…	…	…	…	
5676	Snack Foods	141.3154	OUT046	
5677	Starchy Foods	169.1448	OUT018	
5678	Health and Hygiene	118.7440	OUT045	
5679	Canned	214.6218	OUT017	
5680	Canned	79.7960	OUT045	

	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	\
0	1999	Medium	Tier 1	
1	2007	Medium	Tier 2	
2	1998	Medium	Tier 3	
3	2007	Medium	Tier 2	
4	1985	Medium	Tier 3	
…	…	…	…	
5676	1997	Small	Tier 1	
5677	2009	Medium	Tier 3	
5678	2002	Medium	Tier 2	
5679	2007	Medium	Tier 2	
5680	2002	Medium	Tier 2	

	Outlet_Type
0	Supermarket Type1
1	Supermarket Type1
2	Grocery Store
3	Supermarket Type1
4	Supermarket Type3

```
…                  …
5676  Supermarket Type1
5677  Supermarket Type2
5678  Supermarket Type1
5679  Supermarket Type1
5680  Supermarket Type1

[5681 rows x 11 columns]
```

[160]:
```
# EDA USING DTALE LIBRARY
```

[161]:
```
import dtale
```

[162]:
```
dtale.show(df_train)
```

```
<IPython.lib.display.IFrame at 0x2232a69e130>
```

[162]:

[163]:
```
# EDA USING PANDAS PROFILING
```

[164]:
```
!pip install pandas-profiling
```

```
Requirement already satisfied: pandas-profiling in
c:\users\aayus\anaconda3\lib\site-packages (3.6.6)
Requirement already satisfied: ydata-profiling in
c:\users\aayus\anaconda3\lib\site-packages (from pandas-profiling) (4.5.1)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (3.1.2)
Requirement already satisfied: tqdm<5,>=4.48.2 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (4.64.1)
Requirement already satisfied: wordcloud>=1.9.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (1.9.2)
Requirement already satisfied: imagehash==4.3.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (4.3.1)
Requirement already satisfied: pandas!=1.4.0,<2.1,>1.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (1.4.4)
Requirement already satisfied: visions[type_image_path]==0.7.5 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (0.7.5)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (6.0)
Requirement already satisfied: multimethod<2,>=1.4 in
```

```
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (1.9.1)
Requirement already satisfied: scipy<1.12,>=1.4.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (1.9.1)
Requirement already satisfied: htmlmin==0.1.12 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (0.1.12)
Requirement already satisfied: numpy<1.24,>=1.16.0 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (1.21.5)
Requirement already satisfied: requests<3,>=2.24.0 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (2.28.1)
Requirement already satisfied: typeguard<3,>=2.13.2 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (2.13.3)
Requirement already satisfied: seaborn<0.13,>=0.10.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (0.11.2)
Requirement already satisfied: pydantic<2,>=1.8.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (1.10.12)
Requirement already satisfied: matplotlib<4,>=3.2 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (3.5.2)
Requirement already satisfied: phik<0.13,>=0.11.1 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (0.12.3)
Requirement already satisfied: dacite>=1.8 in c:\users\aayus\anaconda3\lib\site-
packages (from ydata-profiling->pandas-profiling) (1.8.1)
Requirement already satisfied: statsmodels<1,>=0.13.2 in
c:\users\aayus\anaconda3\lib\site-packages (from ydata-profiling->pandas-
profiling) (0.13.2)
Requirement already satisfied: pillow in c:\users\aayus\anaconda3\lib\site-
packages (from imagehash==4.3.1->ydata-profiling->pandas-profiling) (9.2.0)
Requirement already satisfied: PyWavelets in c:\users\aayus\anaconda3\lib\site-
packages (from imagehash==4.3.1->ydata-profiling->pandas-profiling) (1.3.0)
Requirement already satisfied: attrs>=19.3.0 in
c:\users\aayus\anaconda3\lib\site-packages (from
visions[type_image_path]==0.7.5->ydata-profiling->pandas-profiling) (21.4.0)
Requirement already satisfied: networkx>=2.4 in
c:\users\aayus\anaconda3\lib\site-packages (from
visions[type_image_path]==0.7.5->ydata-profiling->pandas-profiling) (2.8.4)
Requirement already satisfied: tangled-up-in-unicode>=0.0.4 in
c:\users\aayus\anaconda3\lib\site-packages (from
visions[type_image_path]==0.7.5->ydata-profiling->pandas-profiling) (0.2.0)
Requirement already satisfied: MarkupSafe>=2.0 in
```

```
c:\users\aayus\anaconda3\lib\site-packages (from jinja2<3.2,>=2.11.1->ydata-
profiling->pandas-profiling) (2.1.3)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\aayus\anaconda3\lib\site-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas-profiling) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in
c:\users\aayus\anaconda3\lib\site-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas-profiling) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\aayus\anaconda3\lib\site-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas-profiling) (1.4.2)
Requirement already satisfied: packaging>=20.0 in
c:\users\aayus\anaconda3\lib\site-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas-profiling) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\aayus\anaconda3\lib\site-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas-profiling) (2.8.2)
Requirement already satisfied: cycler>=0.10 in
c:\users\aayus\anaconda3\lib\site-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas-profiling) (0.11.0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\aayus\anaconda3\lib\site-packages (from pandas!=1.4.0,<2.1,>1.1->ydata-
profiling->pandas-profiling) (2022.1)
Requirement already satisfied: joblib>=0.14.1 in
c:\users\aayus\anaconda3\lib\site-packages (from phik<0.13,>=0.11.1->ydata-
profiling->pandas-profiling) (1.1.0)
Requirement already satisfied: typing-extensions>=4.2.0 in
c:\users\aayus\anaconda3\lib\site-packages (from pydantic<2,>=1.8.1->ydata-
profiling->pandas-profiling) (4.3.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
c:\users\aayus\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-
profiling->pandas-profiling) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\aayus\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-
profiling->pandas-profiling) (3.3)
Requirement already satisfied: charset-normalizer<3,>=2 in
c:\users\aayus\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-
profiling->pandas-profiling) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\aayus\anaconda3\lib\site-packages (from requests<3,>=2.24.0->ydata-
profiling->pandas-profiling) (2022.9.14)
Requirement already satisfied: patsy>=0.5.2 in
c:\users\aayus\anaconda3\lib\site-packages (from statsmodels<1,>=0.13.2->ydata-
profiling->pandas-profiling) (0.5.2)
Requirement already satisfied: colorama in c:\users\aayus\anaconda3\lib\site-
packages (from tqdm<5,>=4.48.2->ydata-profiling->pandas-profiling) (0.4.5)
Requirement already satisfied: six in c:\users\aayus\anaconda3\lib\site-packages
(from patsy>=0.5.2->statsmodels<1,>=0.13.2->ydata-profiling->pandas-profiling)
```

(1.16.0)

[165]:
```
from pandas_profiling import ProfileReport
```

[166]:
```
profile = ProfileReport(df_train, title='Pandas Profilling Report')
```

[167]:
```
profile
```

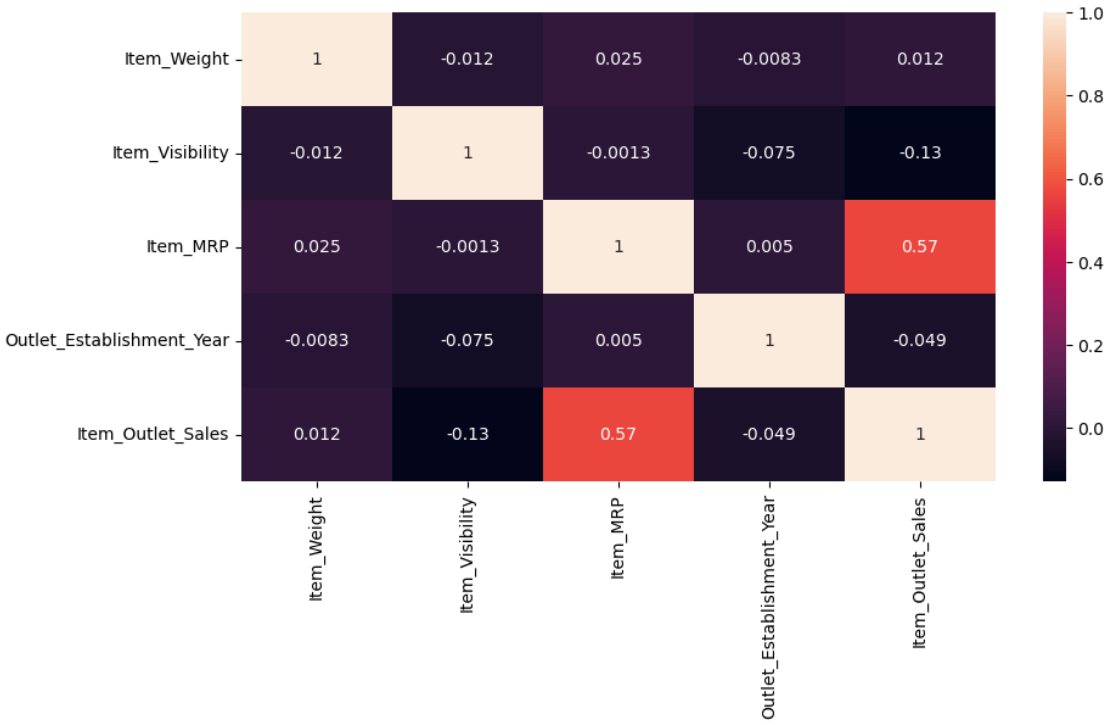
```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
<IPython.core.display.HTML object>
```

[167]:

[168]:
```
# EDA USING SEABORN LIBRARY
```

[169]:
```
plt.figure(figsize=(10,5))
sns.heatmap(df_train.corr(), annot=True)
plt.show
```

[169]: <function matplotlib.pyplot.show(close=None, block=None)>

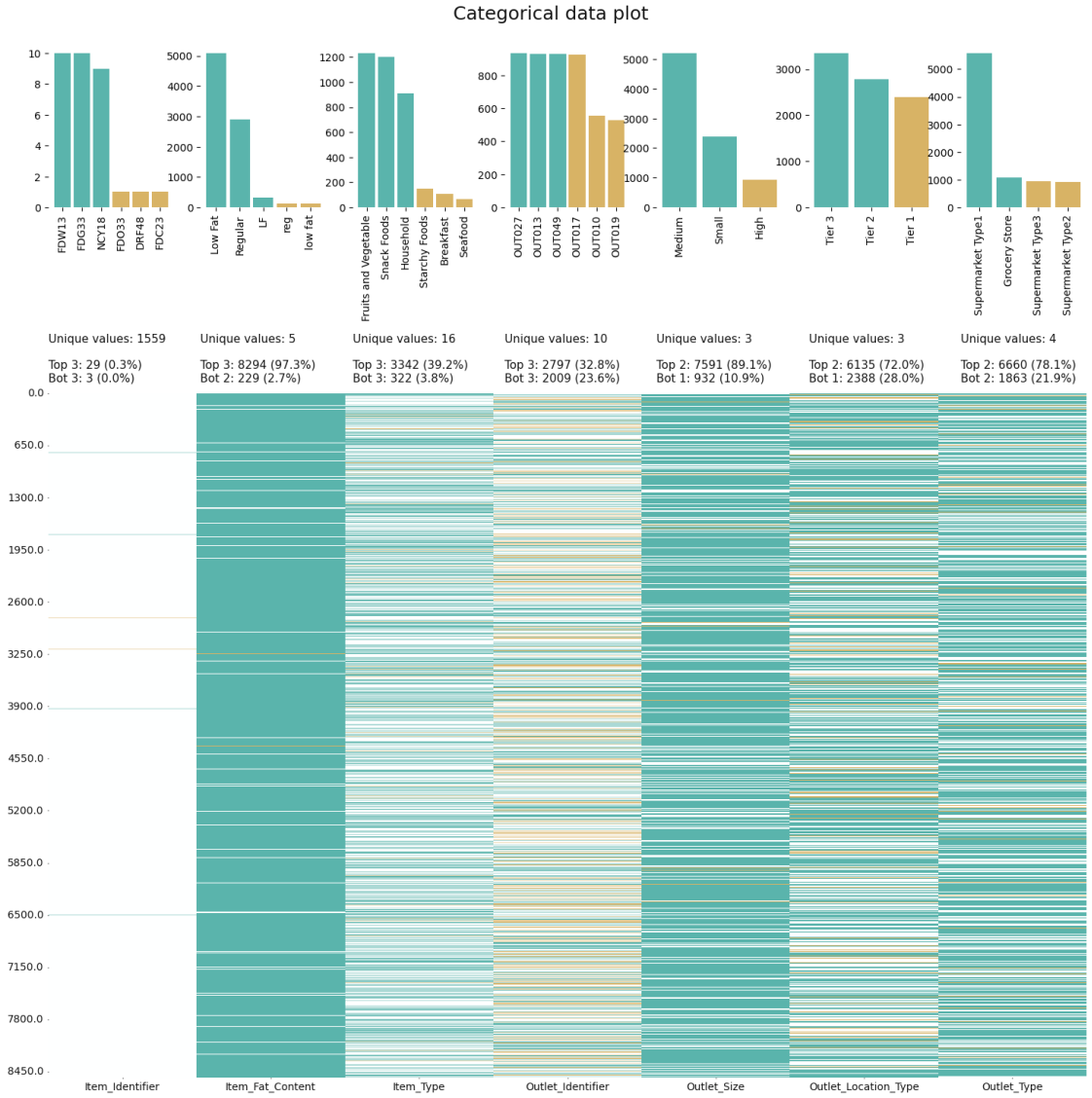


[170]:
```
# EDA USING klib LIBRARY
```

[171]:
```
import klib
```

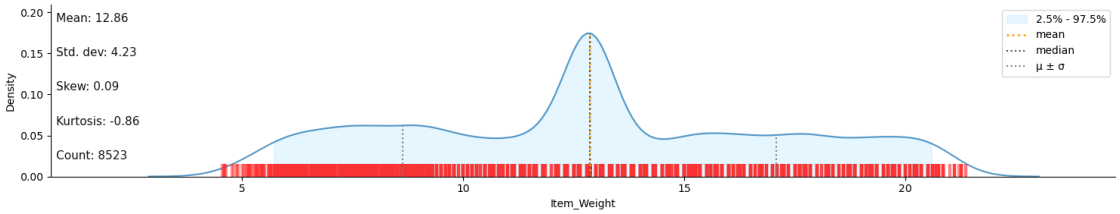
[172]:
```
klib.cat_plot(df_train)
```

[172]:
```
GridSpec(6, 7)
```



[173]:
```
klib.dist_plot(df_train)
```

[173]:
```
<AxesSubplot:xlabel='Item_Weight', ylabel='Density'>
```



1 Data Cleaning

[174]:
```
klib.data_cleaning(df_train)
```

```
Shape of cleaned data: (8523, 12) - Remaining NAs: 0


Dropped rows: 0
     of which 0 duplicates. (Rows (first 150 shown): [])

Dropped columns: 0
     of which 0 single valued.     Columns: []
Dropped missing values: 0
Reduced memory by at least: 0.52 MB (-66.67%)
```

[174]:

	item_identifier	item_weight	item_fat_content	item_visibility	\
0	FDA15	9.300000	Low Fat	0.016047	
1	DRC01	5.920000	Regular	0.019278	
2	FDN15	17.500000	Low Fat	0.016760	
3	FDX07	19.200001	Regular	0.000000	
4	NCD19	8.930000	Low Fat	0.000000	
…	…	…	…	…	
8518	FDF22	6.865000	Low Fat	0.056783	
8519	FDS36	8.380000	Regular	0.046982	
8520	NCJ29	10.600000	Low Fat	0.035186	
8521	FDN46	7.210000	Regular	0.145221	
8522	DRG01	14.800000	Low Fat	0.044878	

	item_type	item_mrp	outlet_identifier	\
0	Dairy	249.809204	OUT049	
1	Soft Drinks	48.269199	OUT018	
2	Meat	141.617996	OUT049	
3	Fruits and Vegetables	182.095001	OUT010	
4	Household	53.861401	OUT013	
…	…	…	…	
8518	Snack Foods	214.521805	OUT013	

```
8519           Baking Goods  108.156998            OUT045
8520     Health and Hygiene   85.122398            OUT035
8521            Snack Foods  103.133202            OUT018
8522            Soft Drinks   75.467003            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      Medium               Tier 3
4                          1987        High               Tier 3
…                           …          …                    …
8518                       1987        High               Tier 3
8519                       2002      Medium               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        3735.137939
1     Supermarket Type2         443.422791
2     Supermarket Type1        2097.270020
3         Grocery Store         732.380005
4     Supermarket Type1         994.705200
…                  …               …
8518  Supermarket Type1        2778.383301
8519  Supermarket Type1         549.284973
8520  Supermarket Type1        1193.113647
8521  Supermarket Type2        1845.597656
8522  Supermarket Type1         765.669983

[8523 rows x 12 columns]
```

[175]:
```
klib.clean_column_names(df_train)
```

[175]:
```
     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      Medium               Tier 3
4                          1987        High               Tier 3
…                           …          …                   …
8518                       1987        High               Tier 3
8519                       2002      Medium               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]
```

[176]:
```
df_train.info()
```

```
<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                8523 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
```

[177]:
```
klib.convert_datatypes(df_train)
```

[177]:

	item_identifier	item_weight	item_fat_content	item_visibility	\
0	FDA15	9.300000	Low Fat	0.016047	
1	DRC01	5.920000	Regular	0.019278	
2	FDN15	17.500000	Low Fat	0.016760	
3	FDX07	19.200001	Regular	0.000000	
4	NCD19	8.930000	Low Fat	0.000000	
…	…	…	…	…	
8518	FDF22	6.865000	Low Fat	0.056783	
8519	FDS36	8.380000	Regular	0.046982	
8520	NCJ29	10.600000	Low Fat	0.035186	
8521	FDN46	7.210000	Regular	0.145221	
8522	DRG01	14.800000	Low Fat	0.044878	

	item_type	item_mrp	outlet_identifier	\
0	Dairy	249.809204	OUT049	
1	Soft Drinks	48.269199	OUT018	
2	Meat	141.617996	OUT049	
3	Fruits and Vegetables	182.095001	OUT010	
4	Household	53.861401	OUT013	
…	…	…	…	
8518	Snack Foods	214.521805	OUT013	
8519	Baking Goods	108.156998	OUT045	
8520	Health and Hygiene	85.122398	OUT035	
8521	Snack Foods	103.133202	OUT018	
8522	Soft Drinks	75.467003	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      Medium               Tier 3
4                         1987        High               Tier 3
…                          …          …                    …
8518                      1987        High               Tier 3
8519                      2002      Medium               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        3735.137939
1     Supermarket Type2         443.422791
2     Supermarket Type1        2097.270020
3         Grocery Store         732.380005
4     Supermarket Type1         994.705200
…                  …               …
8518  Supermarket Type1        2778.383301
8519  Supermarket Type1         549.284973
8520  Supermarket Type1        1193.113647
8521  Supermarket Type2        1845.597656
8522  Supermarket Type1         765.669983

[8523 rows x 12 columns]
```

[178]:
```
df_train.info()
```

```
<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                8523 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
```

[179]:
```
klib.pool_duplicate_subsets(df_train)
```

[179]:
```
     item_identifier  item_visibility  item_mrp  item_outlet_sales  \
0              FDA15         0.016047  249.8092          3735.1380
1              DRC01         0.019278   48.2692           443.4228
2              FDN15         0.016760  141.6180          2097.2700
3              FDX07         0.000000  182.0950           732.3800
4              NCD19         0.000000   53.8614           994.7052
…                …               …         …                 …
8518           FDF22         0.056783  214.5218          2778.3834
8519           FDS36         0.046982  108.1570           549.2850
8520           NCJ29         0.035186   85.1224          1193.1136
8521           FDN46         0.145221  103.1332          1845.5976
8522           DRG01         0.044878   75.4670           765.6700

      pooled_vars
0               0
1               1
2               2
3               3
4               4
…             …
8518         8518
8519         8519
8520         8520
8521         8521
8522         8522

[8523 rows x 5 columns]
```

2 Preprocessing task before model building

3 Label Encoding

Label encoding vs hot encoding

[180]:
```
from sklearn.preprocessing import LabelEncoder
le= LabelEncoder()
```

[181]:
```
df_train = df_train.apply(le.fit_transform)
```

[182]:
```
df_train['item_fat_content']= le.fit_transform(df_train['item_fat_content'])
df_train['item_type']= le.fit_transform(df_train['item_type'])
df_train['outlet_size']= le.fit_transform(df_train['outlet_size'])
df_train['outlet_location_type']= le.
 ↪fit_transform(df_train['outlet_location_type'])
df_train['outlet_type']= le.fit_transform(df_train['outlet_type'])
```

[183]:
```
df_train
```

[183]:
```
      item_identifier  item_weight  item_fat_content  item_visibility  \
0                 156          284                 1              664
1                   8           57                 2              880
2                 662          376                 1              715
3                1121          393                 2                0
4                1297          265                 1                0
…                 …            …                   …                …
8518              370          125                 1             3912
8519              897          233                 2             3278
8520             1357          299                 1             2302
8521              681          149                 2             7175
8522               50          347                 1             3108

      item_type  item_mrp  outlet_identifier  outlet_establishment_year  \
0             4      5592                  9                          4
1            14       473                  3                          8
2            10      2901                  9                          4
3             6      4227                  0                          3
4             9       627                  1                          1
…         …       …                    …                          …
8518         13      4955                  1                          1
8519          0      2023                  7                          5
8520          8      1263                  6                          6
8521         13      1857                  3                          8
8522         14      1011                  8                          2

      outlet_size  outlet_location_type  outlet_type  item_outlet_sales
0               1                     0            1               2540
1               1                     2            2                422
2               1                     0            1               1639
3               1                     2            0                670
4               0                     2            1                865
…           …                     …            …                …
8518            0                     2            1               2047
8519            1                     1            1                516
8520            2                     1            1               1018
8521            1                     2            2               1466
8522            2                     0            1                697

[8523 rows x 12 columns]
```

4 Splitting Data into Train and Test

[184]:
```
X=df_train.drop('item_outlet_sales',axis=1)
Y=df_train['item_outlet_sales']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, random_state=101,␣
↪test_size=0.2)
```

5 Standarization

[185]:
```
X.describe()
```

[185]:

	item_identifier	item_weight	item_fat_content	item_visibility	\
count	8523.000000	8523.000000	8523.000000	8523.000000	
mean	779.714889	298.756776	1.369354	3709.198639	
std	449.222377	95.451067	0.644810	2396.606857	
min	0.000000	0.000000	0.000000	0.000000	
25%	395.500000	285.000000	1.000000	1595.500000	
50%	783.000000	326.000000	1.000000	3708.000000	
75%	1167.000000	361.000000	2.000000	5789.500000	
max	1558.000000	415.000000	4.000000	7879.000000	

	item_type	item_mrp	outlet_identifier	outlet_establishment_year	\
count	8523.000000	8523.000000	8523.000000	8523.000000	
mean	7.226681	2941.096562	4.722281	3.790684	
std	4.209990	1675.483234	2.837201	2.730322	
min	0.000000	0.000000	0.000000	0.000000	
25%	4.000000	1535.500000	2.000000	1.000000	
50%	6.000000	2949.000000	5.000000	4.000000	
75%	10.000000	4351.500000	7.000000	6.000000	
max	15.000000	5937.000000	9.000000	8.000000	

	outlet_size	outlet_location_type	outlet_type
count	8523.000000	8523.000000	8523.000000
mean	1.170832	1.112871	1.201220
std	0.600327	0.812757	0.796459
min	0.000000	0.000000	0.000000
25%	1.000000	0.000000	1.000000
50%	1.000000	1.000000	1.000000
75%	2.000000	2.000000	1.000000
max	2.000000	2.000000	3.000000

[187]:
```
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
```

[188]:
```
X_train_std= sc.fit_transform(X_train)
```

[189]:
```
X_test_std= sc.transform(X_test)
```

[190]:
```
X_train_std
```

[190]:
```
array([[ 0.59264645,  1.00439018, -0.57382672, …, -1.95699503,
         1.08786619, -0.25964107],
       [-1.50626006, -1.23760194, -0.57382672, …, -0.28872895,
        -0.13870429, -0.25964107],
       [-0.23049765,  1.01496562,  0.97378032, …, -0.28872895,
        -0.13870429, -0.25964107],
       …,
       [ 1.44241516,  0.23238346, -0.57382672, …,  1.37953713,
        -1.36527477, -0.25964107],
       [ 0.81008074, -0.09545501,  0.97378032, …, -0.28872895,
        -0.13870429, -0.25964107],
       [ 1.114045  ,  0.62367454, -0.57382672, …, -0.28872895,
         1.08786619,  0.98524841]])
```

[191]:
```
X_test_std
```

[191]:
```
array([[ 1.39360338,  0.06317651, -0.57382672, …, -0.28872895,
         1.08786619,  0.98524841],
       [ 0.7146759 ,  0.86690953, -0.57382672, …, -1.95699503,
         1.08786619, -0.25964107],
       [ 0.3419314 , -1.15299846,  0.97378032, …,  1.37953713,
        -1.36527477, -0.25964107],
       …,
       [ 1.38250979,  0.61309911, -0.57382672, …, -0.28872895,
         1.08786619, -1.50453056],
       [ 1.23385574,  0.77173062, -0.57382672, …, -0.28872895,
         1.08786619,  0.98524841],
       [ 0.37299344, -2.33744713,  0.97378032, …, -0.28872895,
        -0.13870429, -0.25964107]])
```

[192]:
```
Y_test
```

[192]:
```
8179     801
8355    2056
3411    1529
7089     779
6954    1858
        …
1317    1386
4996     808
531      356
3891    1144
6629    1839
Name: item_outlet_sales, Length: 1705, dtype: int64
```

[193]:
```
import joblib
```

[194]:
```
joblib.dump(sc,r'C:\Users\aayus\Desktop\Project\Models\sc.sav')
```

[194]:
```
['C:\\Users\\aayus\\Desktop\\Project\\Models\\sc.sav']
```

6 Model Building

[195]:
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
```

[196]:
```
lr.fit(X_train_std,Y_train)
```

[196]:
```
LinearRegression()
```

[197]:
```
X_test.head()
```

[197]:

	item_identifier	item_weight	item_fat_content	item_visibility	\
8179	1409	305	1	3815	
8355	1103	381	1	2651	
3411	935	190	2	4881	
7089	336	408	1	3427	
6954	365	175	1	1624	

	item_type	item_mrp	outlet_identifier	outlet_establishment_year	\
8179	8	1753	3	8	
8355	13	3152	1	1	
3411	1	1062	8	2	
7089	6	207	2	7	
6954	3	3291	7	5	

	outlet_size	outlet_location_type	outlet_type
8179	1	2	2
8355	0	2	1
3411	2	0	1
7089	1	1	1
6954	1	1	1

[198]:
```
Y_pred_lr=lr.predict(X_test_std)
```

[199]:
```
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

[200]:
```
print(r2_score(Y_test,Y_pred_lr))
print(mean_absolute_error(Y_test,Y_pred_lr))
print(np.sqrt(mean_squared_error(Y_test,Y_pred_lr)))
```

```
0.5511512103720639
486.8525564523135
609.6711243347048
```

[201]:
```
joblib.dump(lr,r'C:\Users\aayus\Desktop\Project\Models\random_forest_grid.sav')
```

[201]: ['C:\\Users\\aayus\\Desktop\\Project\\Models\\random_forest_grid.sav']

7 Random Forest Regressor

[202]:
```
from sklearn.ensemble import RandomForestRegressor
rf= RandomForestRegressor(n_estimators=1000)
```

[207]:
```
rf.fit(X_train_std,Y_train)
```

[207]: RandomForestRegressor(n_estimators=1000)

[204]:
```
Y_pred_rf= rf.predict(X_test_std)
```

[205]:
```
print(r2_score(Y_test,Y_pred_rf))
print(mean_absolute_error(Y_test,Y_pred_rf))
print(np.sqrt(mean_squared_error(Y_test,Y_pred_rf)))
```

```
0.6122963521706927
430.52423929618766
566.6247501477168
```

[208]:
```
joblib.dump(rf,r'C:\Users\aayus\Desktop\Project\Models\random_forest_grid.sav')
```

[208]: ['C:\\Users\\aayus\\Desktop\\Project\\Models\\random_forest_grid.sav']

8 Hyper Parameter Tuning

[209]:
```
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import GridSearchCV

# define models and parameters
model = RandomForestRegressor()
n_estimators = [10, 100, 1000]
max_depth=range(1,31)
min_samples_leaf=np.linspace(0.1, 1.0)
max_features=["auto", "sqrt", "log2"]
min_samples_split=np.linspace(0.1, 1.0, 10)

# define grid search
grid = dict(n_estimators=n_estimators)

#cv = RepeatedStratifiedKFold(n_splits=5, n_repeats=3, random_state=101)

grid_search_forest = GridSearchCV(estimator=model, param_grid=grid, n_jobs=-1,
                           scoring='r2',error_score=0,verbose=2,cv=2)

grid_search_forest.fit(X_train_std, Y_train)
```

```
# summarize results
print(f"Best: {grid_search_forest.best_score_:.3f} using {grid_search_forest.
 ↪best_params_}")
means = grid_search_forest.cv_results_['mean_test_score']
stds = grid_search_forest.cv_results_['std_test_score']
params = grid_search_forest.cv_results_['params']

for mean, stdev, param in zip(means, stds, params):
    print(f"{mean:.3f} ({stdev:.3f}) with: {param}")
```

```
Fitting 2 folds for each of 3 candidates, totalling 6 fits
Best: 0.622 using {'n_estimators': 1000}
0.589 (0.002) with: {'n_estimators': 10}
0.619 (0.001) with: {'n_estimators': 100}
0.622 (0.000) with: {'n_estimators': 1000}
```

[210]:
```
grid_search_forest.best_params_
```

[210]:
```
{'n_estimators': 1000}
```

[211]:
```
grid_search_forest.best_score_
```

[211]:
```
0.6222578887562102
```

[212]:
```
Y_pred_rf_grid=grid_search_forest.predict(X_test_std)
```

[213]:
```
r2_score(Y_test,Y_pred_rf_grid)
```

[213]:
```
0.6124515742947404
```

9 Save your model

[214]:
```
import joblib
```

[215]:
```
joblib.dump(grid_search_forest,r'C:
 ↪\Users\aayus\Desktop\Project\Models\random_forest_grid.sav')
```

[215]:
```
['C:\\Users\\aayus\\Desktop\\Project\\Models\\random_forest_grid.sav']
```

[216]:
```
model=joblib.load(r'C:\Users\aayus\Desktop\Project\Models\random_forest_grid.
 ↪sav')
```

[217]:
```
model.predict(X_test_std)
```

[217]:
```
array([1404.825, 2262.748, 1104.438, …,  424.837, 1316.068, 1704.556])
```