LAB 01

EC 9560 - DATA MINING

THEVARAJAN.R.J

2019/E/146

SEMESTER 7

22 SEP 2023

TITLE:

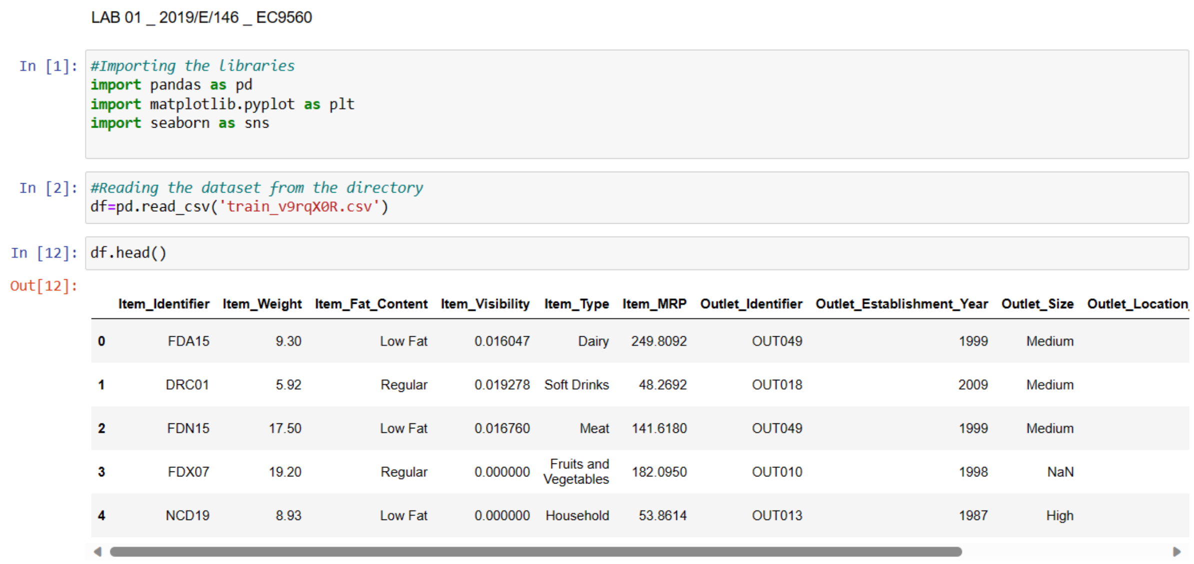
Big Mart Sales Prediction

OBJECTIVE:

Use regression analysis to predict sales based on attributes.

PROGRESS:

DATA PRE-PROCESSING:



LAB 01 _ 2019/E/146 _ EC9560

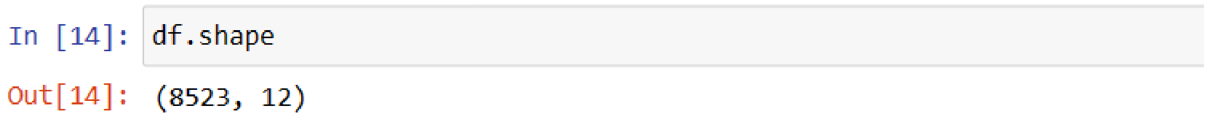
```
In [1]: #Importing the libraries
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

```
In [2]: #Reading the dataset from the directory
        df=pd.read_csv('train_v9rqX0R.csv')
```

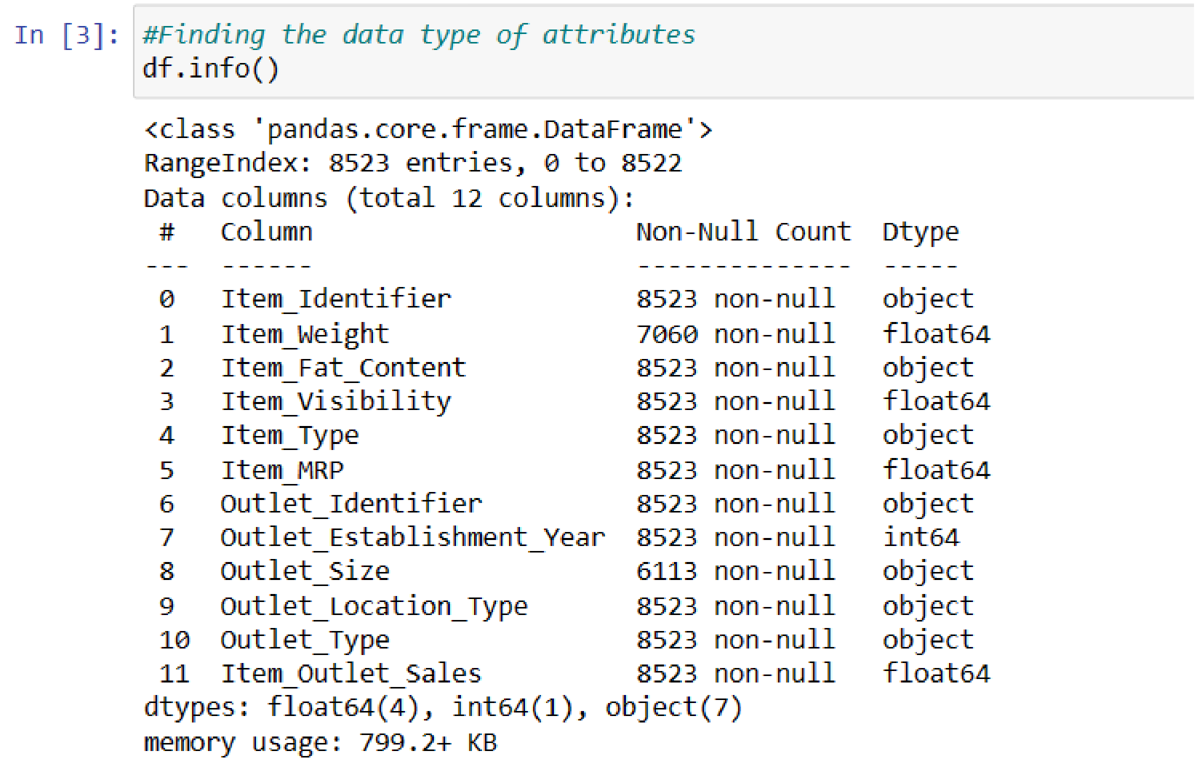
```
In [12]: df.head()
```

Out[12]:

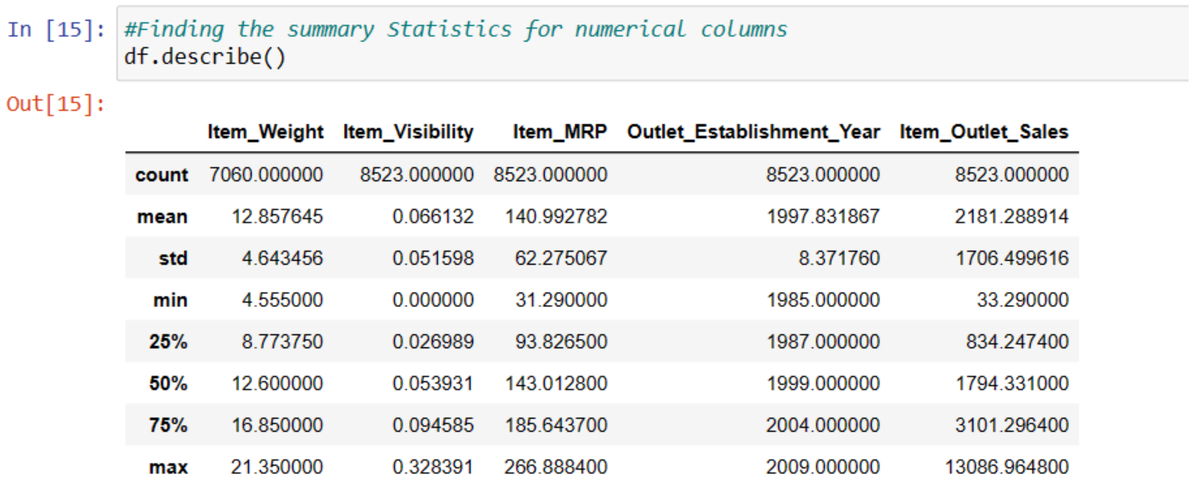
	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	



```
In [14]: df.shape

Out[14]: (8523, 12)
```



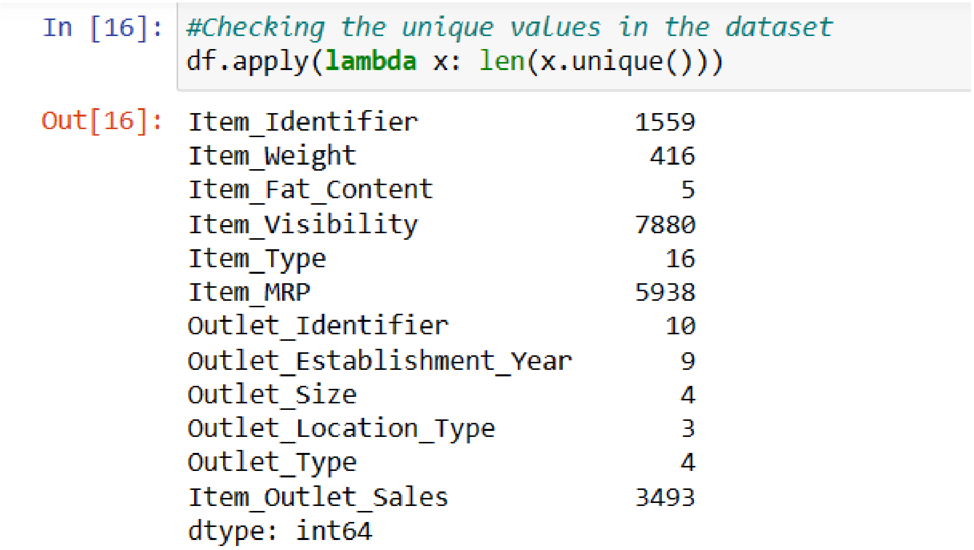
```
In [3]: #Finding the data type of attributes
        df.info()

<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
```



In [15]:
```
#Finding the summary Statistics for numerical columns
df.describe()
```

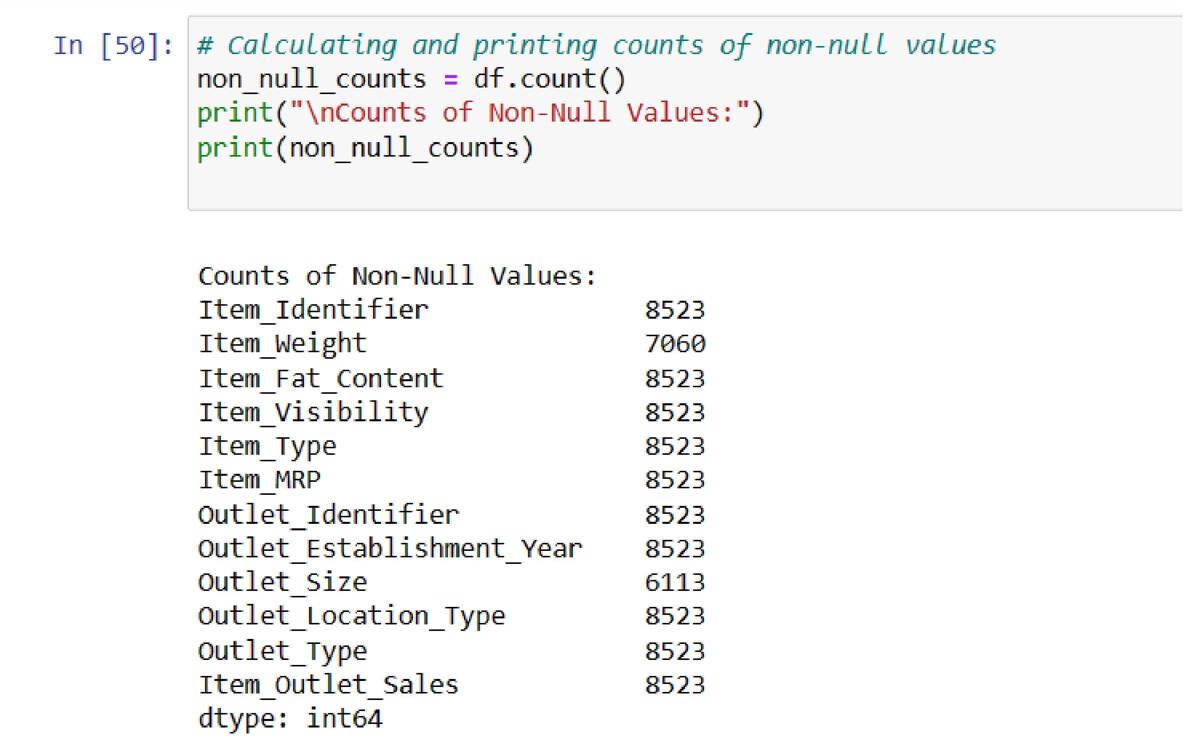
Out[15]:

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	7060.000000	8523.000000	8523.000000	8523.000000	8523.000000
mean	12.857645	0.066132	140.992782	1997.831867	2181.288914
std	4.643456	0.051598	62.275067	8.371760	1706.499616
min	4.555000	0.000000	31.290000	1985.000000	33.290000
25%	8.773750	0.026989	93.826500	1987.000000	834.247400
50%	12.600000	0.053931	143.012800	1999.000000	1794.331000
75%	16.850000	0.094585	185.643700	2004.000000	3101.296400
max	21.350000	0.328391	266.888400	2009.000000	13086.964800



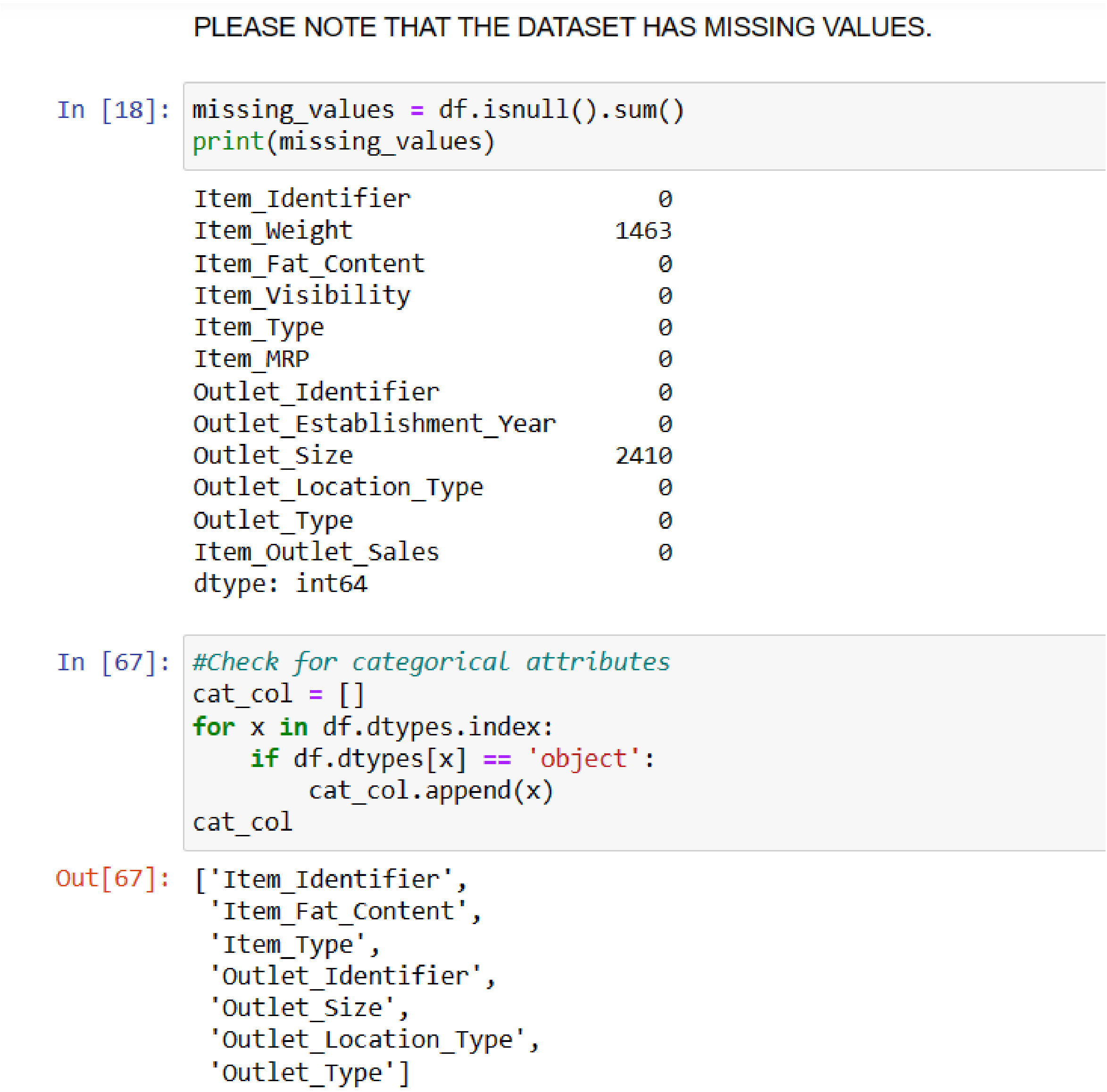
In [16]:
```
#Checking the unique values in the dataset
df.apply(lambda x: len(x.unique()))
```

Out[16]:
```
Item_Identifier              1559
Item_Weight                   416
Item_Fat_Content                5
Item_Visibility              7880
Item_Type                      16
Item_MRP                     5938
Outlet_Identifier              10
Outlet_Establishment_Year       9
Outlet_Size                     4
Outlet_Location_Type            3
Outlet_Type                     4
Item_Outlet_Sales            3493
dtype: int64
```



```
In [50]: # Calculating and printing counts of non-null values
         non_null_counts = df.count()
         print("\nCounts of Non-Null Values:")
         print(non_null_counts)
```

```
Counts of Non-Null Values:
Item_Identifier              8523
Item_Weight                  7060
Item_Fat_Content             8523
Item_Visibility              8523
Item_Type                    8523
Item_MRP                     8523
Outlet_Identifier            8523
Outlet_Establishment_Year    8523
Outlet_Size                  6113
Outlet_Location_Type         8523
Outlet_Type                  8523
Item_Outlet_Sales            8523
dtype: int64
```



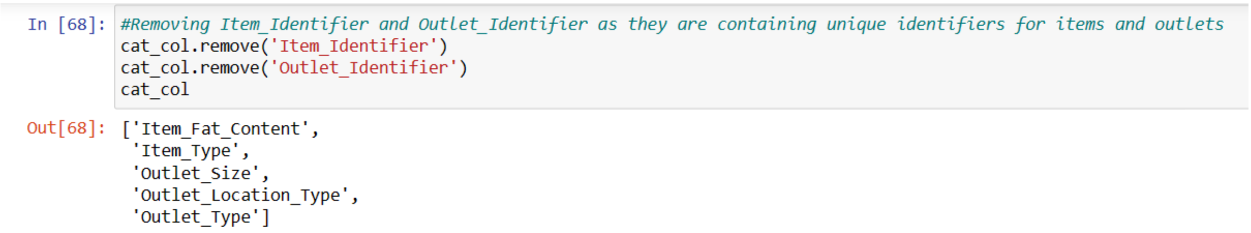
PLEASE NOTE THAT THE DATASET HAS MISSING VALUES.

```
In [18]: missing_values = df.isnull().sum()
         print(missing_values)
```

```
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
```

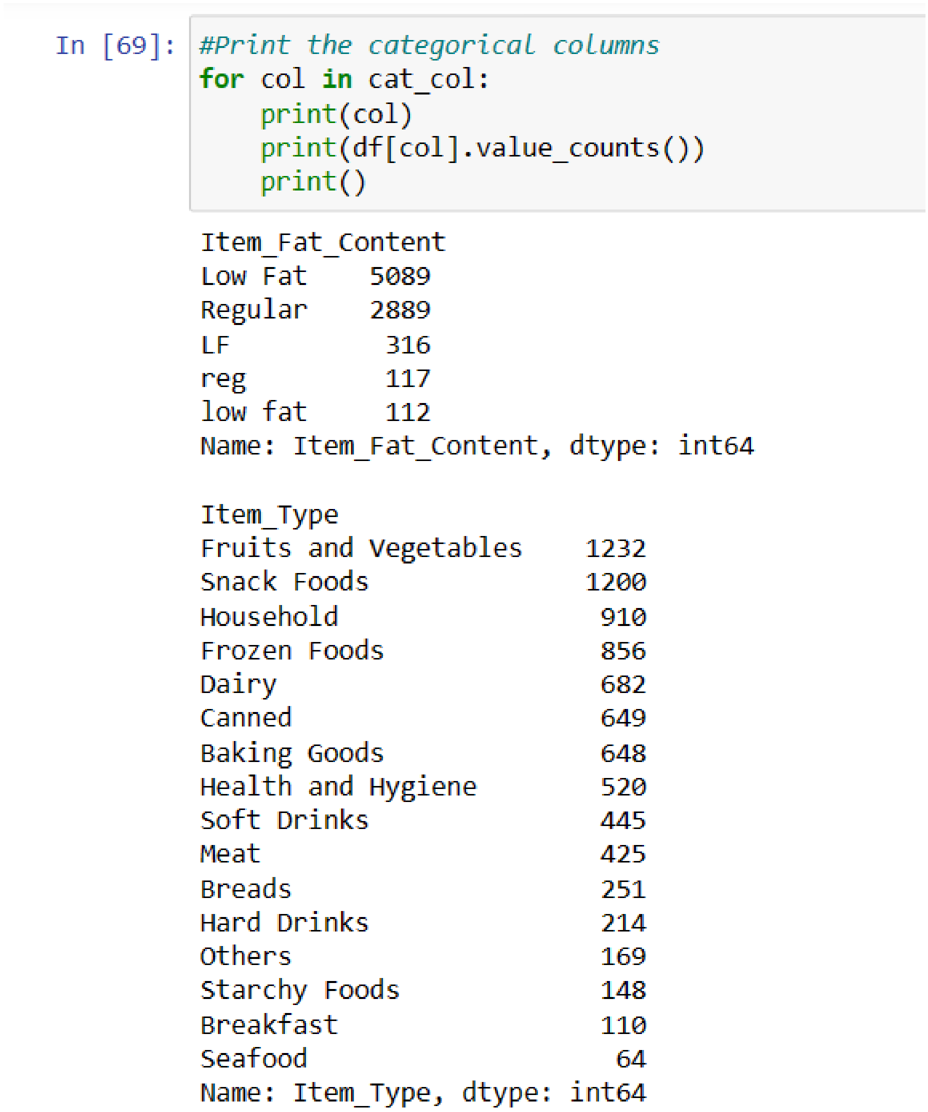
```
In [67]: #Check for categorical attributes
         cat_col = []
         for x in df.dtypes.index:
             if df.dtypes[x] == 'object':
                 cat_col.append(x)
         cat_col
```

```
Out[67]: ['Item_Identifier',
          'Item_Fat_Content',
          'Item_Type',
          'Outlet_Identifier',
          'Outlet_Size',
          'Outlet_Location_Type',
          'Outlet_Type']
```



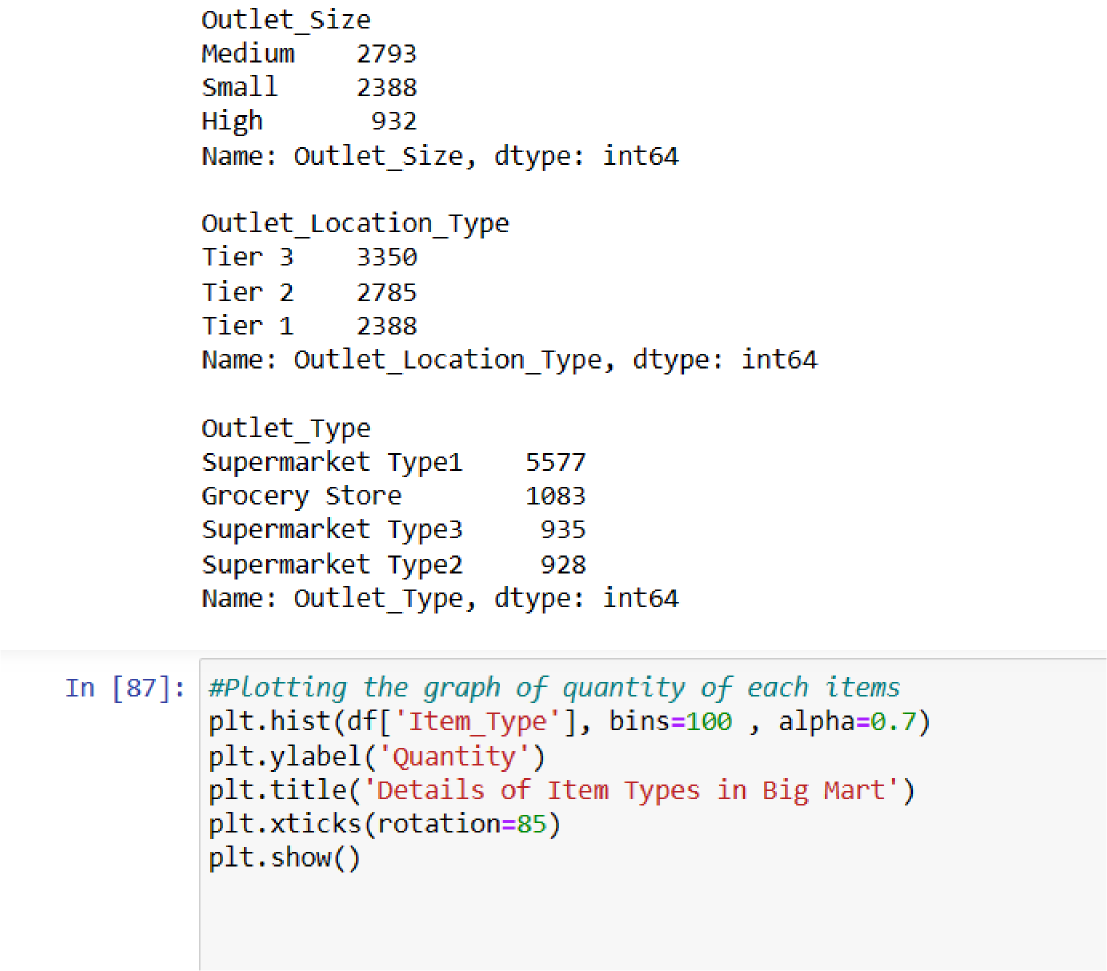
```
In [68]: #Removing Item_Identifier and Outlet_Identifier as they are containing unique identifiers for items and outlets
         cat_col.remove('Item_Identifier')
         cat_col.remove('Outlet_Identifier')
         cat_col
```

```
Out[68]: ['Item_Fat_Content',
          'Item_Type',
          'Outlet_Size',
          'Outlet_Location_Type',
          'Outlet_Type']
```



```
#Print the categorical columns
for col in cat_col:
    print(col)
    print(df[col].value_counts())
    print()
```

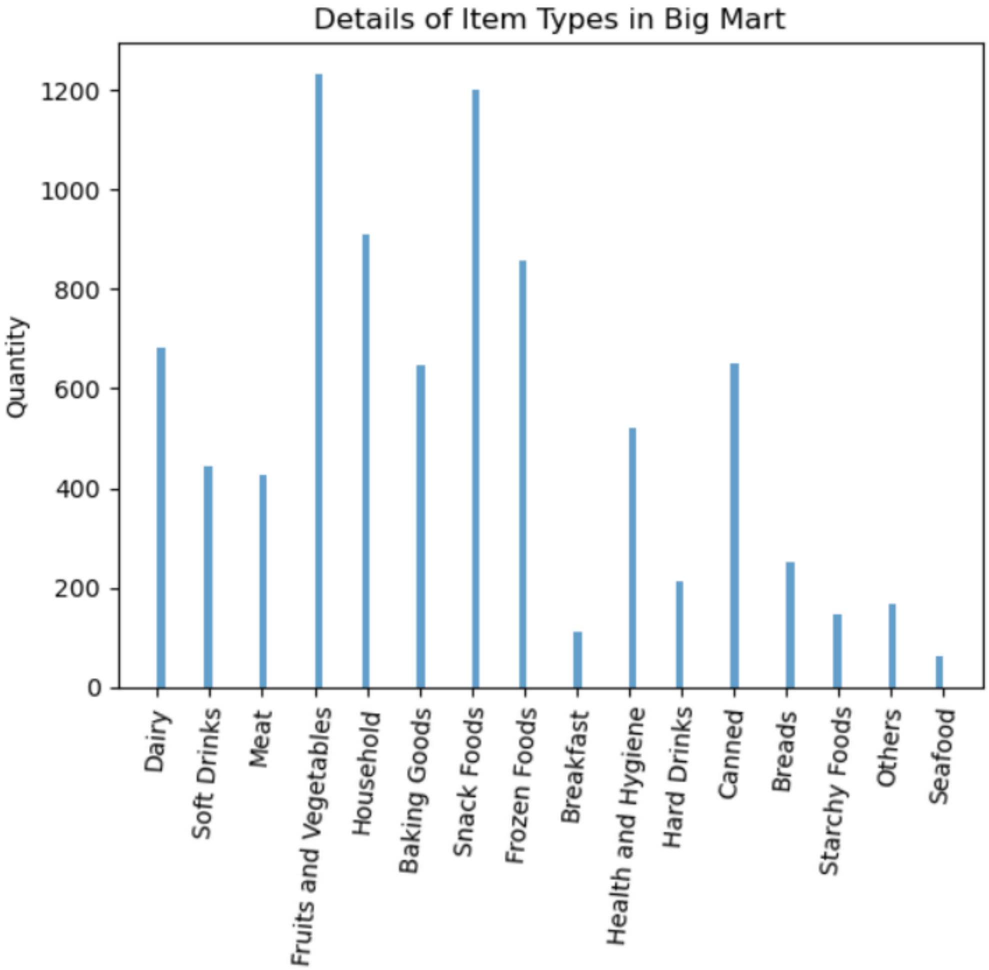
```
Item_Fat_Content
Low Fat     5089
Regular     2889
LF           316
reg          117
low fat      112
Name: Item_Fat_Content, dtype: int64

Item_Type
Fruits and Vegetables    1232
Snack Foods              1200
Household                 910
Frozen Foods              856
Dairy                     682
Canned                    649
Baking Goods              648
Health and Hygiene        520
Soft Drinks               445
Meat                      425
Breads                    251
Hard Drinks               214
Others                    169
Starchy Foods             148
Breakfast                 110
Seafood                    64
Name: Item_Type, dtype: int64
```



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

Outlet_Location_Type
Tier 3    3350
Tier 2    2785
Tier 1    2388
Name: Outlet_Location_Type, dtype: int64

Outlet_Type
Supermarket Type1    5577
Grocery Store        1083
Supermarket Type3     935
Supermarket Type2     928
Name: Outlet_Type, dtype: int64
```

In [87]:

```
#Plotting the graph of quantity of each items
plt.hist(df['Item_Type'], bins=100 , alpha=0.7)
plt.ylabel('Quantity')
plt.title('Details of Item Types in Big Mart')
plt.xticks(rotation=85)
plt.show()
```


Details of Item Types in Big Mart
Quantity
0
200
400
600
800
1000
1200
Dairy
Soft Drinks
Meat
Fruits and Vegetables
Household
Baking Goods
Snack Foods
Frozen Foods
Breakfast
Health and Hygiene
Hard Drinks
Canned
Breads
Starchy Foods
Others
Seafood