DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBJECT :- DATA ANALYTICS
CODE :- UE18CS312

TOPIC :- BIG MARKET SALES PREDICTION

PROJECT BY :-

SINDHURA S – PES2201800374

INTRODUCTION:-

Big markets are the chains of super markets, with stores all around the country.

In this era or nowadays people big markets shopping malls and many things are in demand since they make less or easy for people in their busy life.They kkep track of their sales date of each and every individual item which acts as the evidence for them and even for the customer and update the inventory management as well. And this data also stores or contains large number of customer data and individual attributes in the data warehouse.

What and why is it import?

Better experiences start with better insights.

Since in these days people go for best things and at the less time so even big markets play imporatant role in our daily lives.

It tracks every marketingmsales amd customer success interaction for full prospect and coustomer journey visibility .

About our data set

Here we have taken a dataset which has

8524 rows

12 coloumns

Why is data analysis important?

The process where

- The raw data is collected
- Data is processed dataset has to get cleaned
- Exploratory data analysis
- Models and algorithm
- Communicate visualization report
- Final informative source is ready to use

And also we can understand and get a proper future predictions. It also shows the graphs like:-

- Histogram
- Line graphs
- Bar chart
- Pie chart

How did we approach to the Analysis?

- Data processing
- Checked for missing values

- Checked for null values
- Checked for duplicate values
- Standardization

  Categorical values converted to numerical values and then standardization is done by using appropriate formula.

  Approach

- Normaliztion QQ plot
- Visualization

  Bar chart

  Pie chart

  Line graph

  Histogram

  PCA

  Evaluation of solution or algorithm implemented:-

  Linear regression model which allows to summarize and study the relationship between continious variables, where one variable will be independent and the other one will be dependent. The equation is in the form of y = ax+b, where y is dependent variable and x is independent.

  Association rule

  It use dto find correlations and co-occurrences between data sets. They are ideally use dto explain the patterns in data form seemingly imdependent information repositories, such as relational databases and transactional databases.

  CONFIDENCE

  The confidence of an association rule is a support of (X U Y) divided by the support of X.

  Therefore the rule is in this case support of (2,5,3) divided bt the support of (2,5).

Suppose A^B -> C then confidence = support (A^B->C)
i.e. the number of transactions in which all three items are present/support(A,B),where both A and B are present.

SUPPORT

Here we can see how popular an itemset is, as measured by the proportion of transactions in which item set appears. Example:- if suppose there is a table which has , the support of {apple} is 4 out of 8 or 50% itemsets can also contain multiple items.
Support(X->Y) = support(X U Y) = P(X U Y)

LIFT

We can see how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is.
Lift={X,Y}=support{X,Y}/(support{X} support{Y})

CONVICTION

It measures the implication strength of the rule from statistical independence.

Apripori Algorithm

An algorithm for frequent item set mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the database amd extending them larger and larger item sets as long as those item sets appear sufficiently often in the database.

DATA ANALYSIS

Before applying the apripori algorithm on the data set, we are going to show some visualization to learn more about the transactions.

Example:- we can use the frequency plot () function to create an item frequency bar plot, in order to view the distribution of products.

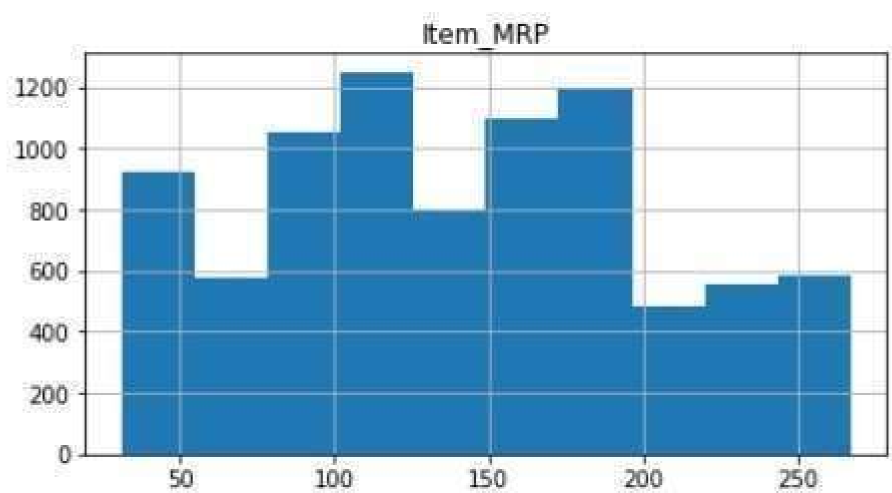
So here you can see some graph visualizations,

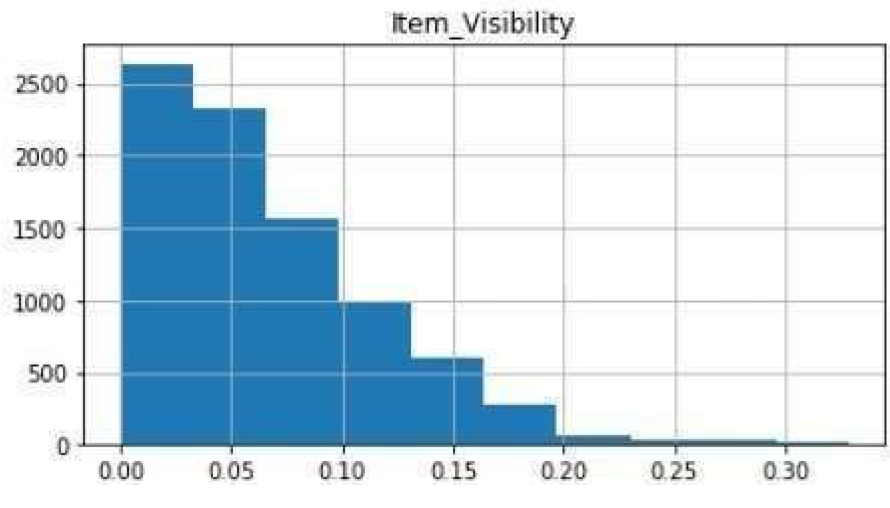
In [5]:

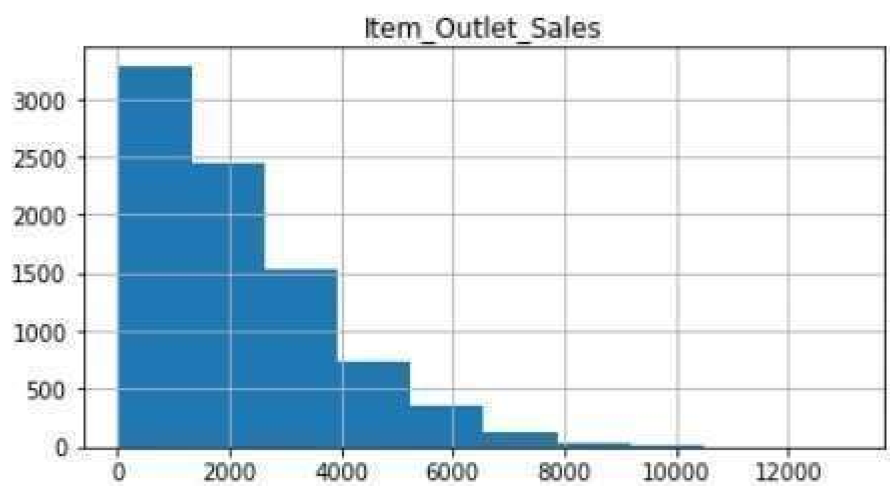
```
df.hist(figsize=(15,12))
```

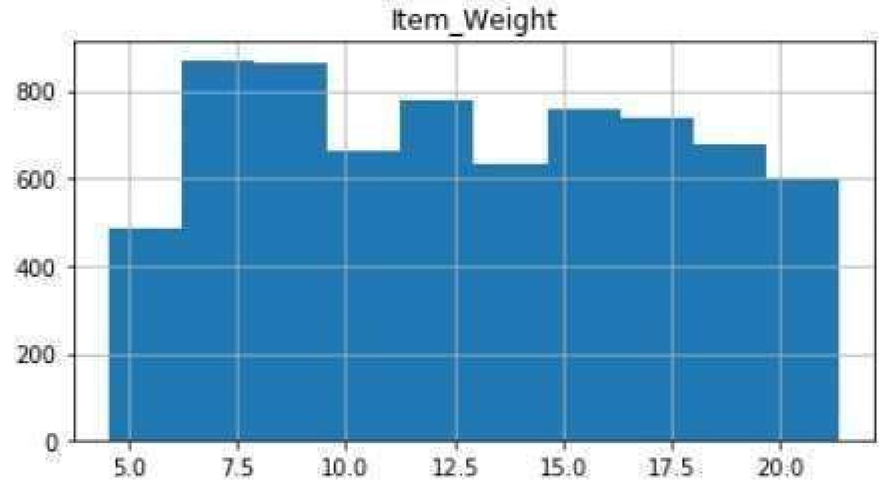
Out[5]:

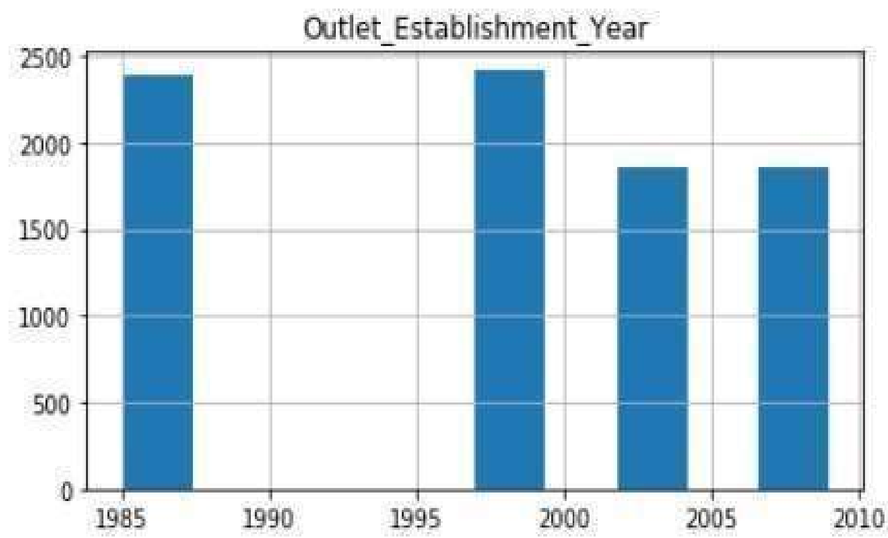
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fb8c03e6048>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb8c032f630>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb8c0359ba8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb8c0309160>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb8c02af6d8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb8c02d6b00>]],
      dtype=object)
```




Item_Visibility
2500
2000
1500
1000
500
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30


Item_Outlet_Sales
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000


Item_Weight
800
600
400
200
0
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Outlet_Establishment_Year
2500
2000
1500
1000
500
0
1985
1990
1995
2000
2005
2010

Column ITEM WEIGHT and OUTLET SIZE can contain missing values so,

Lets see the correlation between target and features

In [7]:

```
corr_matrix=df.corr()
corr_matrix['Item_Outlet_Sales']
```

Out[7]:

```
Item_Weight                    0.014123
Item_Visibility               -0.128625
Item_MRP                       0.567574
Outlet_Establishment_Year     -0.049135
Item_Outlet_Sales              1.000000
Name: Item_Outlet_Sales, dtype: float64
```

Lets start checking the columns relation with target ITEM_OUTLET_SALES price.

First is ITEM_IDENTIFIER

```
df.Item_Identifier.value_counts()
```

Out[8]:

```
FDG33    10
FDW13    10
NCQ06     9
FDX20     9
FDT07     9
FDF56     9
DRN47     9
FDV38     9
FDD38     9
FDU12     9
FDX31     9
NCB18     9
FDV60     9
NCI54     9
FDG09     9
FDO19     9
FDX04     9
FDW49     9
NCF42     9
FDP25     9
NCY18     9
DRE49     9
FDE52     9
```

```
FDM16     2
FDG28     2
FDM38     2
FDR03     2
FDR57     2
FDT33     2
FDE38     2
NCS41     2
FDP15     2
FDE39     2
NCM42     2
FDU43     2
FDD22     2
DRG25     2
FDK57     1
FDT35     1
DRF48     1
FDE52     1
FDY43     1
FDO33     1
FDN52     1
FDQ60     1
FDC23     1
Name: Item_Identifier, Length: 1559, dty
pe: int64
```

ITEM_WAIT column strength is very low so we can drop it and next column is ITEM_FAT_CONTENT

In [9]:

```
df.Item_Fat_Content.value_counts()
```

Out[9]:

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

LOW_FAT and reg belong to same category so replacing LF, Low fat and reg to their category

In [10]:

```
df.Item_Fat_Content=df.Item_Fat_Content.
replace('LF','Low Fat')
```

In [11]:

```
df.Item_Fat_Content=df.Item_Fat_Content.
replace('reg','Regular')
df.Item_Fat_Content=df.Item_Fat_Content.
replace('low fat','Low Fat')
```

Now for ITEM_MRP column

In [14]:

```
fig,axes=plt.subplots(1,1,figsize=(12,
8))
sns.scatterplot(x='Item_MRP',y='Item_Out
let_Sales',hue='Item_Fat_Content',size
='Item_Weight',data=df)
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot a
t 0x7fb8c0242da0>
```

We can use those perpendicular line to divide those

```
df.Item_MRP=pd.cut(df.Item_MRP,bins=[25,
69,137,203,270],labels=
['a','b','c','d'],right=True)
```

In [18]:

```
df.head()
```

Out[18]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_V
0	FDA15	9.30	Low Fat	0.016(
1	DRC01	5.92	Regular	0.019:

```python
df.head()
```

Out[18]:

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_V
0	FDA15	9.30	Low Fat	0.016(
1	DRC01	5.92	Regular	0.019:
2	FDN15	17.50	Low Fat	0.016;
3	FDX07	19.20	Regular	0.000(
4	NCD19	8.93	Low Fat	0.000(

Now lets explore other columns

In [19]:

```
fig,axes=plt.subplots(3,1,figsize=(15,12))
sns.scatterplot(x='Item_Visibility',y='Item_Outlet_Sales',hue='Item_MRP',ax=axes[0],data=df)
sns.boxplot(x='Item_Type',y='Item_Outlet_Sales',ax=axes[1],data=df)
sns.boxplot(x='Outlet_Identifier',y='Item_Outlet_Sales',ax=axes[2],data=df)
```

Out[19]:

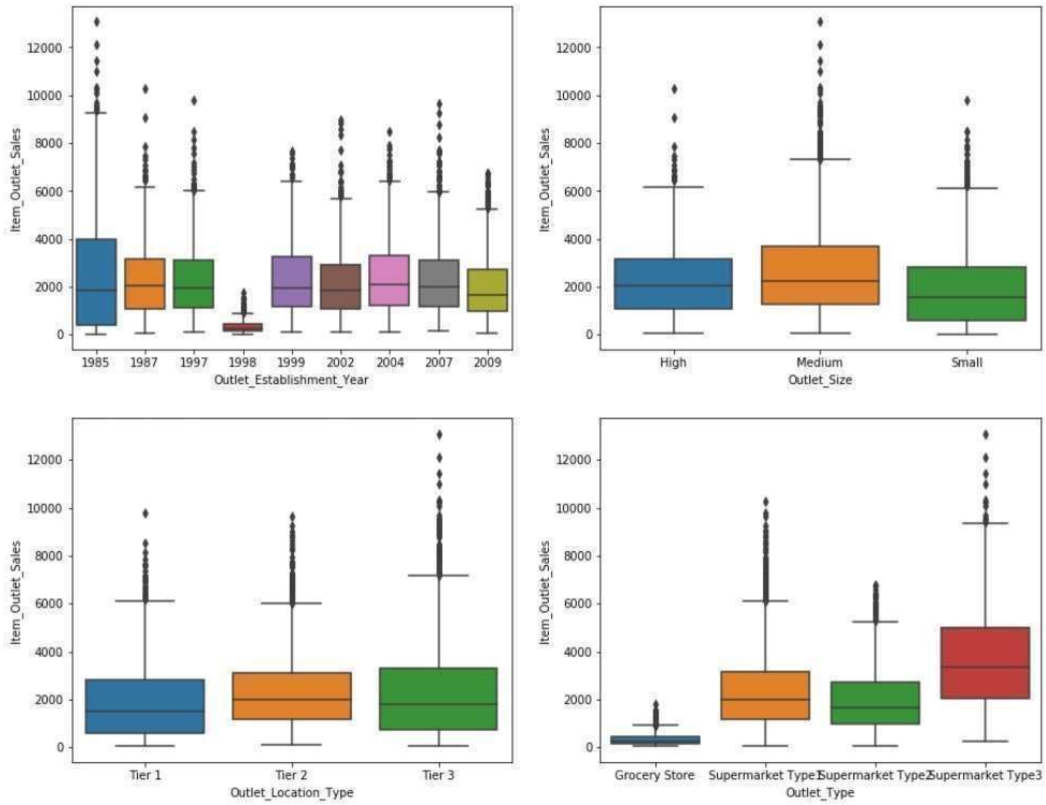
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8c0097780>
```

In [20]:

```
fig,axes=plt.subplots(2,2,figsize=(15,12))
sns.boxplot(x='Outlet_Establishment_Year',y='Item_Outlet_Sales',ax=axes[0,0],data=df)
sns.boxplot(x='Outlet_Size',y='Item_Outlet_Sales',ax=axes[0,1],data=df)
sns.boxplot(x='Outlet_Location_Type',y='Item_Outlet_Sales',ax=axes[1,0],data=df)
sns.boxplot(x='Outlet_Type',y='Item_Outlet_Sales',ax=axes[1,1],data=df)
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8bc186c18>
```



From the above graphs we can say that we can drop ITEM_VISIBILITY along with ITEM_WEIGHT.
These columns have very low correlation strength target column

Therefore columns for model training will be like,

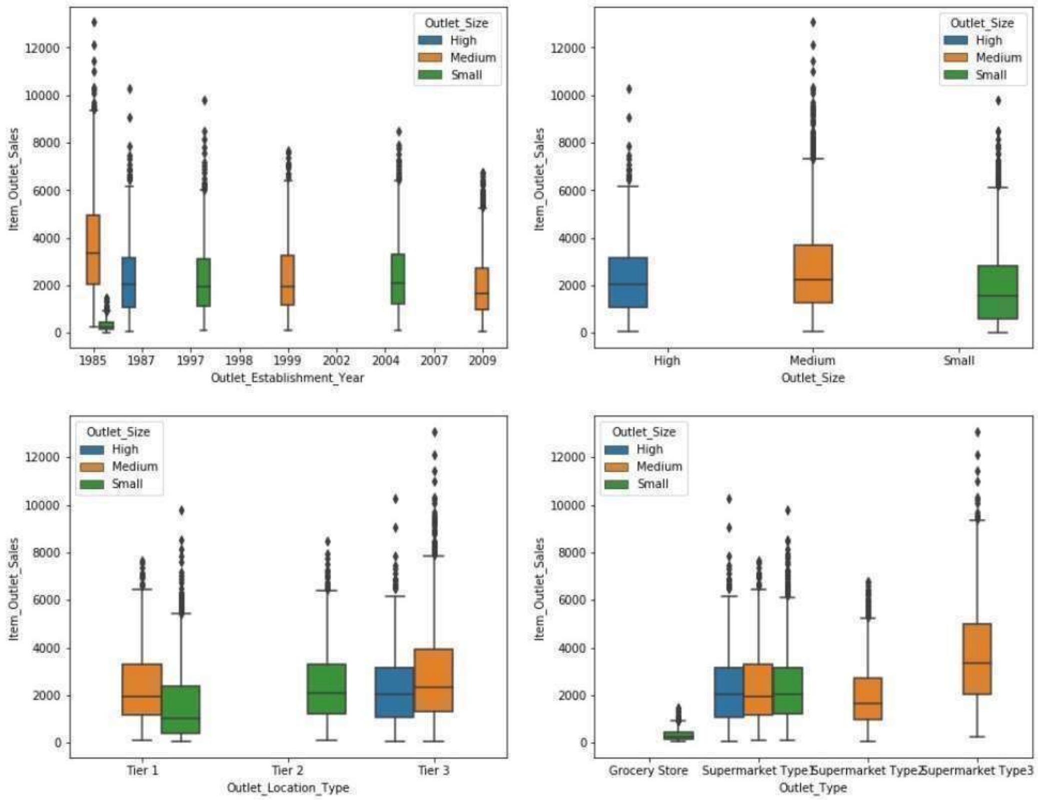
```
attributes=['Item_MRP','Outlet_Type','Outlet_Location_Type','Outlet_Size','Outlet_Establishment_Year','Outlet_Identifier','Item_Type','Item_Outlet_Sales']
```

In [22]:

```
fig,axes=plt.subplots(2,2,figsize=(15,12))
sns.boxplot(x='Outlet_Establishment_Year',y='Item_Outlet_Sales',hue='Outlet_Size',ax=axes[0,0],data=df)
sns.boxplot(x='Outlet_Size',y='Item_Outlet_Sales',hue='Outlet_Size',ax=axes[0,1],data=df)
sns.boxplot(x='Outlet_Location_Type',y='Item_Outlet_Sales',hue='Outlet_Size',ax=axes[1,0],data=df)
sns.boxplot(x='Outlet_Type',y='Item_Outlet_Sales',hue='Outlet_Size',ax=axes[1,1],data=df)
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot a
t 0x7fb8b7734320>
```



In [23]:

```
data=df[attributes]
```

In [24]:

```
data.info()
```

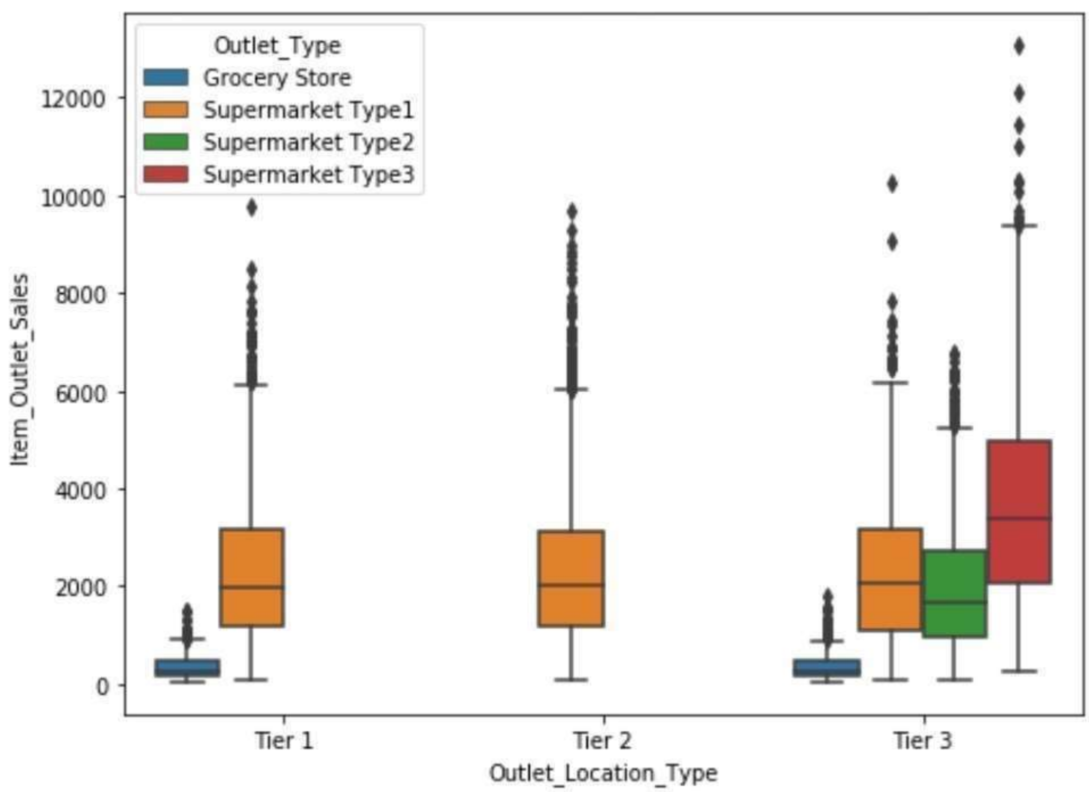
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 8 columns):
Item_MRP                     8523 non-nu
ll category
Outlet_Type                  8523 non-nu
ll category
Outlet_Location_Type         8523 non-nu
ll category
Outlet_Size                  6113 non-nu
ll category
Outlet_Establishment_Year    8523 non-nu
ll int64
Outlet_Identifier            8523 non-nu
ll category
Item_Type                    8523 non-nu
ll category
Item_Outlet_Sales            8523 non-nu
ll float64
dtypes: category(6), float64(1), int64
```

In [25]:

```
fig,axes=plt.subplots(1,1,figsize=(8,6))
sns.boxplot(y='Item_Outlet_Sales',hue='Outlet_Type',x='Outlet_Location_Type',data=data)
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8b74d8080>
```



In [26]:

```
data[data.Outlet_Size.isnull()]
```

Out[26]:

	Item_MRP	Outlet_Type	Outlet_Location_Type	Ou
3	c	Grocery Store	Tier 3	Na
8	b	Supermarket Type1	Tier 2	Na
9	c	Supermarket Type1	Tier 2	Na
25	a	Supermarket Type1	Tier 2	Na
28	a	Grocery Store	Tier 3	Na
30	a	Grocery Store	Tier 3	Na
33	b	Supermarket Type1	Tier 2	Na
45	c	Grocery Store	Tier 3	Na
46	c	Supermarket Type1	Tier 2	Na

We should observe one thing here that is when OUTLET_TYPE = supermarket type 1 and OUTLET_LOCATION_TYPE is tier 2 then the outlet size is null and when OUTLET_TYPE =GROCERY store and OUTPUT_LOCATION _TYPE is tier 3 then the outlet size is always null,so

In [27]:

```
data.groupby('Outlet_Type').get_group('G
rocery Store')['Outlet_Location_Type'].v
alue_counts()
```

Out[27]:

```
Tier 3    555
Tier 1    528
Tier 2      0
Name: Outlet_Location_Type, dtype: int64
```

In [28]:

```
data.groupby('Outlet_Type').get_group('Grocery Store')
```

Out[28]:

	Item_MRP	Outlet_Type	Outlet_Location_Type	Out
3	c	Grocery Store	Tier 3	Nal
23	b	Grocery Store	Tier 1	Sm
28	a	Grocery Store	Tier 3	Nal
29	a	Grocery Store	Tier 1	Sm
30	a	Grocery Store	Tier 3	Nal
45	c	Grocery Store	Tier 3	Nal
49	c	Grocery Store	Tier 1	Sm
59	c	Grocery Store	Tier 1	Sm
		Grocery		

In [29]:

```
data.groupby(['Outlet_Location_Type','Outlet_Type'])['Outlet_Size'].value_counts()
```

Out[29]:

```
Outlet_Location_Type  Outlet_Type
Outlet_Size
Tier 1                Grocery Store
Small                 528
                      Supermarket Type1
Medium                930

Small                 930
Tier 2                Supermarket Type1
Small                 930
Tier 3                Supermarket Type1
High                  932
                      Supermarket Type2
Medium                928
                      Supermarket Type3
Medium                935
Name: Outlet_Size, dtype: int64
```

In [30]:

```
(data.Outlet_Identifier=='OUT010').value
_counts()
```

Out[30]:

```
False    7968
True      555
Name: Outlet_Identifier, dtype: int64
```

In [31]:

```
data.groupby('Outlet_Size').Outlet_Ident
ifier.value_counts()
```

Out[31]:

```
Outlet_Size  Outlet_Identifier
High         OUT013               932
Medium       OUT027               935
             OUT049               930
             OUT018               928
Small        OUT035               930
             OUT046               930
             OUT019               528
Name: Outlet_Identifier, dtype: int64
```

We will check for the outliers now

In [34]:

```
data.head()
```

Out[34]:

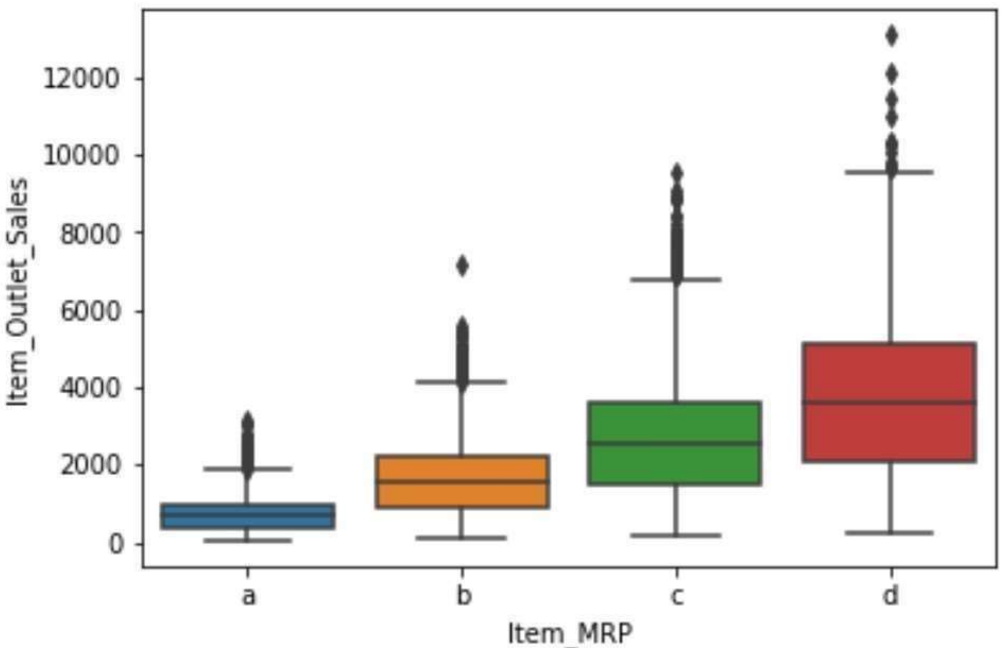
	Item_MRP	Outlet_Type	Outlet_Location_Type	Outlet_
0	d	Supermarket Type1	Tier 1	d
1	a	Supermarket Type2	Tier 3	a
2	c	Supermarket Type1	Tier 1	c
3	c	Grocery Store	Tier 3	c
4	a	Supermarket Type1	Tier 3	a

In [35]:

```
sns.boxplot(x='Item_MRP',y='Item_Outlet_Sales',data=data)
```

Out[35]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8b73f17f0>
```



In [36]:

```
data[data.Item_MRP=='b'].Item_Outlet_Sal
es.max()
```

Out[36]:

```
7158.6816
```

In [37]:

```
data[data.Item_Outlet_Sales==7158.6816]
```

Out[37]:

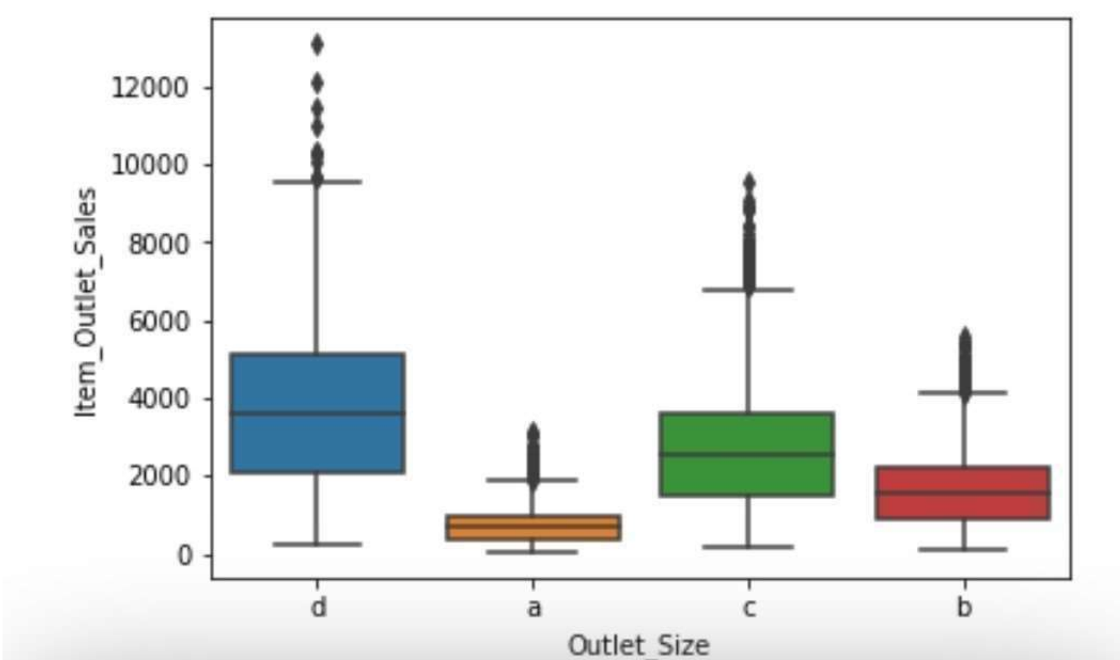
	Item_MRP	Outlet_Type	Outlet_Location_Type	Ou
7737	d	Supermarket Type3	Tier 3	d
7796	b	Supermarket Type3	Tier 3	b

```
data=data.drop(index=4289)
```

```
sns.boxplot(x='Outlet_Size',y='Item_Outlet_Sales',data=data)
```

Out[44]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8b72006a0>
```



And many more…

SUMMARY

In this kernel we have learned about the Apriori algorithm, one of the most frequently used algorithms in data mining. We have reviewed some statistical concepts (support, confidence, lift and conviction) to select interesting rules, we have chosen the appropriate values to execute the algorithm and finally we have visualized the resulting association rules.By the way, if you want to view more kernels about other machine learning algorithms or statistical techniques, you can check the following links:

- Image Compression using PCA
- k-Nearest Neighbors algorithm (k-NN) in the Iris data set
- Clustering wines with k-means

THANK YOU