

# Super market sales – Gross-income prediction (Linear Regression)

Daffodil International University Main campus ,  
Dhanmondi, Dhaka

Name: Sharif Ahamed

ID: 191-15-12406

Section: E

**Abstract-** A online shop has to store thousands of data every day. Now a days people are getting more involved in online. Mostly in the COVID-19 pandemic situation many things are getting updated offline to online. As example people can't go out for lockdown. But they must have to buy daily essential products. So, they are getting more depended online. As the demand are rising, the management have to store more and more data. To get a healthy revenue they have to predict many things. Like average sell every day, stock refill. We can't stock some product for long time like vegetables, fruits etc. so we have to store them such a amount that can satisfy customers want and also not being rotten. For this problem here we have gross income prediction. That will predict the gross income.

**Keywords—** online, COVID-19, lockdown, stock, prediction

## I. Introduction

An online shop has to store and process a many data directly or indirectly. For wise marketing big companies like Facebook, google they algorithm their data. In what area, in when, who can buy a product they assume their rich database and algorithm. In case a big online shop they need to assume in what area, when, in what weather which product can make them a good revenue. That can improve their sell and income greatly. So, in this program I designed a prediction model that can predict their gross income. So, this program may help them to store product wisely and to make a good daily business model. As this program is based on previous record the performance will increased day by day.

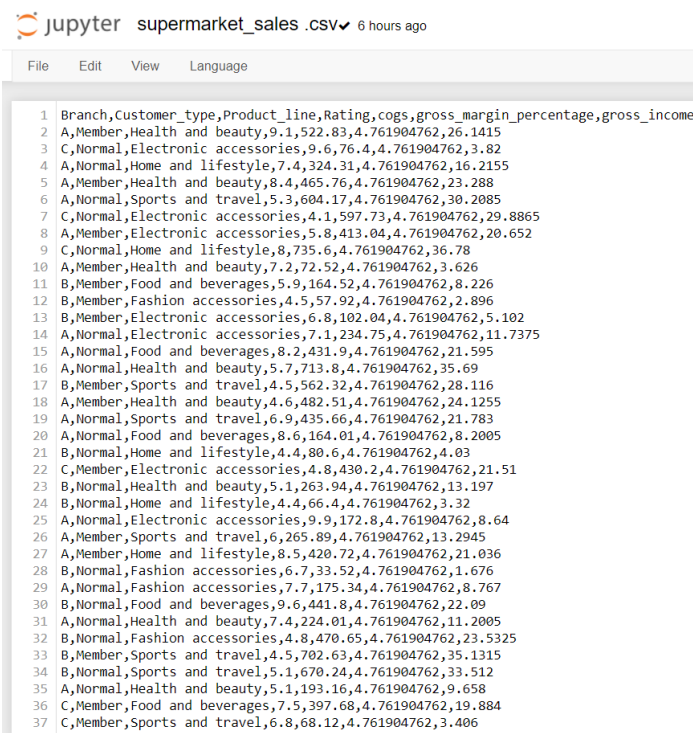
A shops revenue greatly depends on its income. As well as all the performance of that company. So its very important to analyses data and find a acceptable result from which we can assume our next seps.

XXX-X-XXXX-XXXX-X/XX/\$XX.00 ©20XX IEEE

## II. Data Set and Model

### DATA:

The name of my dataset is “Supermarket\_sales” which is in CSV format. In this dataset there are 1000 raw, and 7 columns.



1	Branch	Customer_type	Product_line	Rating	cogs	gross_margin_percentage	gross_income
2	A,Member	Health and beauty	9.1	522.83	4.761904762	26.1415	
3	C,Normal	Electronic accessories	9.6	76.4	4.761904762	3.82	
4	A,Normal	Home and lifestyle	7.4	324.31	4.761904762	16.2155	
5	A,Member	Health and beauty	8.4	465.76	4.761904762	23.288	
6	A,Normal	Sports and travel	5.3	604.17	4.761904762	30.2085	
7	C,Normal	Electronic accessories	4.1	597.73	4.761904762	29.8865	
8	A,Member	Electronic accessories	5.8	413.04	4.761904762	20.652	
9	C,Normal	Home and lifestyle	8.7	35.6	4.761904762	36.78	
10	A,Member	Health and beauty	7.2	72.52	4.761904762	3.626	
11	B,Member	Food and beverages	5.9	164.52	4.761904762	8.226	
12	B,Member	Fashion accessories	4.5	57.92	4.761904762	2.896	
13	B,Member	Electronic accessories	6.8	102.04	4.761904762	5.102	
14	A,Normal	Electronic accessories	7.1	234.75	4.761904762	11.7375	
15	A,Normal	Food and beverages	8.2	431.9	4.761904762	21.595	
16	A,Normal	Health and beauty	5.7	713.8	4.761904762	35.69	
17	B,Member	Sports and travel	4.5	562.32	4.761904762	28.116	
18	A,Member	Health and beauty	4.6	482.51	4.761904762	24.1255	
19	A,Normal	Sports and travel	6.9	435.66	4.761904762	21.783	
20	A,Normal	Food and beverages	8.6	164.01	4.761904762	8.2005	
21	B,Normal	Home and lifestyle	4.4	80.6	4.761904762	4.03	
22	C,Member	Electronic accessories	4.8	430.2	4.761904762	21.51	
23	B,Normal	Health and beauty	5.1	263.94	4.761904762	13.197	
24	B,Normal	Home and lifestyle	4.4	66.4	4.761904762	3.32	
25	A,Normal	Electronic accessories	9.9	172.8	4.761904762	8.64	
26	A,Member	Sports and travel	6.2	65.89	4.761904762	13.2945	
27	A,Member	Home and lifestyle	8.5	420.72	4.761904762	21.036	
28	B,Normal	Fashion accessories	6.7	33.52	4.761904762	1.676	
29	A,Normal	Fashion accessories	7.7	175.34	4.761904762	8.767	
30	B,Normal	Food and beverages	9.6	441.8	4.761904762	22.09	
31	A,Normal	Health and beauty	7.4	224.01	4.761904762	11.2005	
32	B,Normal	Fashion accessories	4.8	470.65	4.761904762	23.5325	
33	B,Member	Sports and travel	4.5	702.63	4.761904762	35.1315	
34	B,Normal	Sports and travel	5.1	670.24	4.761904762	33.512	
35	A,Normal	Health and beauty	5.1	193.16	4.761904762	9.658	
36	C,Member	Food and beverages	7.5	397.68	4.761904762	19.884	
37	C,Member	Sports and travel	6.8	68.12	4.761904762	3.406	
38	A,Member	Sports and travel	7.7	77.1	4.761904762	15.555	

Figure: Dataset Of “Supermarket\_sales”

:

## MODEL:

```
In [27]: import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import linearRegression
from sklearn import preprocessing
from sklearn import linear_model
```

```
In [28]: data=pd.read_csv("supermarket_sales .csv")
```

```
In [29]: data.head()
```

```
Out[29]:
```

	Branch	Customer_type	Product_line	Rating	cogs	gross_margin_percentage	gross_income
0	A	Member	Health and beauty	9.1	522.83	4.761905	26.1415
1	C	Normal	Electronic accessories	9.6	76.40	4.761905	3.8200
2	A	Normal	Home and lifestyle	7.4	324.31	4.761905	16.2155
3	A	Member	Health and beauty	8.4	465.76	4.761905	23.2880
4	A	Normal	Sports and travel	5.3	604.17	4.761905	30.2085

```
In [18]: plt.figure(figsize=(5,5))
plt.grid()
sns.countplot(x='Customer_type', data= data)
plt.title("Customer_type", fontsize=15)
plt.show()
```

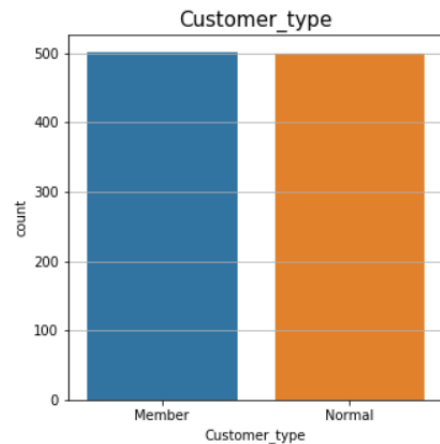


Figure1: Customer type

Import all the libraries and read the dataset using head I called figure function with figure size then grid it. To count function a number of the Member and normal in this dataset used countplot function with a title named 'Customer\_type'. For appearing the figure called the show function. I have done the rest of them following the same way and these are given below.

```
In [30]: data.shape
```

```
Out[30]: (1000, 7)
```

```
In [31]: data.isnull().sum()
```

```
Out[31]: Branch          0
Customer_type          0
Product_line          0
Rating                0
cogs                  0
gross_margin_percentage  0
gross_income          0
dtype: int64
```

```
In [17]: plt.figure(figsize=(5,5))
plt.grid()
sns.countplot(x='Branch', data= data)
plt.title("Branch", fontsize=15)
plt.show()
```

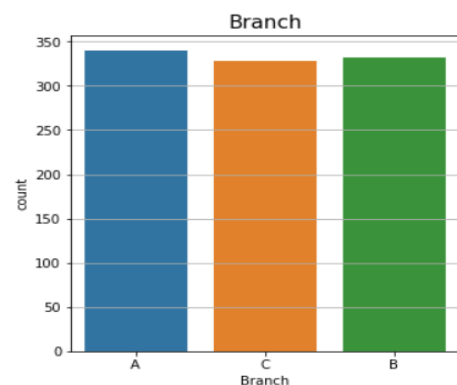
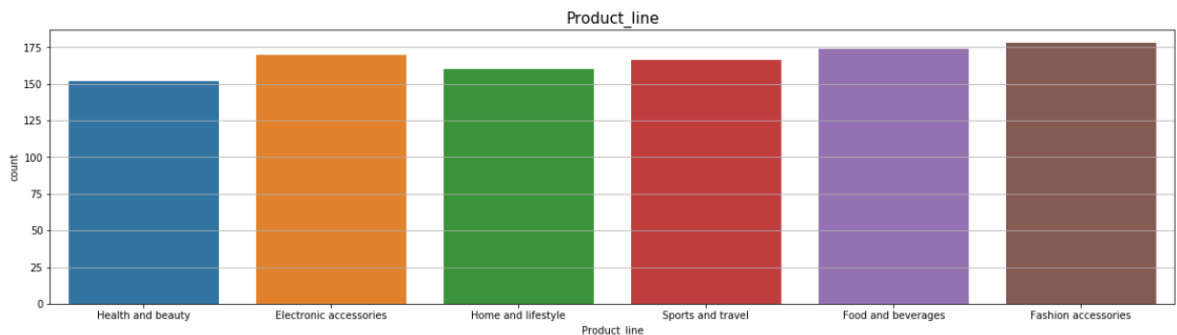


Figure2: Branch

Checked the shape and the number of null values in this dataset by using .shape function and .isnull().sum() function respectively.

```
In [19]: plt.figure(figsize=(20,5))
plt.grid()
sns.countplot(x='Product_line', data= data)
plt.title("Product_line", fontsize=15)
plt.show()
```



**Figure3: Product categories**

According to figure 1 we can see there are two types of customer. One is Member another normal. And from the plot we can see that their number are almost the same.

From figure 2 we can see there are three branches. Branch A, branch B, branch C. B and C has an almost same data. And A has slightly more data which is negatable.

Figure 3 shows the product line. Which gives us a visual presentation of the categories of the dataset. From figure 3 we can see that there are six categories. They are Health and beauty, Electronic accessories, home and lifestyle, sports and travel, food and beverages, fashion accessories.

```
In [33]: slr = LinearRegression()
slr.fit(X_train, y_train)
y_pred = slr.predict(X_test)
y_pred
```

```
Out[33]: array([[ 1.7405],
 [14.14 ],
 [23.0725],
 [19.128 ],
 [ 3.408 ],
 [21.966 ],
 [ 7.288 ],
 [ 5.622 ],
 [10.752 ],
 [24.81 ],
 [14.296 ],
 [16.8175],
 [ 4.876 ],
 [23.097 ],
 [43.866 ],
 [30.368 ],
 [11.2005],
 [15.228 ],
 [13.0025],
 [ 4.336 ],
 [24.665 ],
 [13.715 ],
 [ 2.531 ],
 [34.986 ],
 [ 7.9 ],
 [ 6.685 ],
 [11.0115],
 [ 8.377 ],
 [33.1065],
 [ 9.076 ],
 [44.982 ],
 [ 1.5205],
```

### III. RESULT

Here I have applied the single linear regression method. The single linear regression model has two types of variables, one individual and the other dependent. The dependent value is calculated to depend on the independent value. Before that, the data set has to be split and fitted. It is then called the prediction function to predict the data. The cogs of my project are independent but gross-income is a dependent value. To predict gross-income, the age value must be passed as a parameter in the prediction function. So here I have passed the X test value and by calling y the output shows the gross-income value of different cogs.

#### Predicted values of gross-income based on cogs

There are some non-numeric values. As example customer\_type, product\_line etc. For linear regression we can't use non-linear regression. We have to convert them into numeric values. For that we have to encode them into numeric values.

After encoding them we can use them to predict our final outcome gross-income. For predict I had to input some random values into predict function. Then I got my prediction result.

```
In [37]: reg.predict([[0,1,3,9.0,500.5555,3.79]])
```

```
Out[37]: array([25.027775])
```

**Predicted value of gross-income**

#### IV.CONCLUSION

People are adapting technologies fast. One day almost every work will be depended on online. AI is a great part of this evaluation. Maybe one day online shops will completely depend on AI. For that Data processing is a part and parcel thing. So, my little approach can be developed into next level. But whatever we say Ai has some limitation compare to human. Ai haven't reached that intelligence yet to beat human brain. Real life dataset is huge and full of new experiences. In case of new data human may perform well than AI.