Data science project-Pegah Karimi-GH1019718

March 16, 2022

1 M504-AI and Applications

1.0.1 Winter 2022

1.0.2 Assessment Topic:

you are a data science consultant. Your client company has a dataset and a bunch of business questions. Therefore, you are required to build an exploratory data analysis pipeline in a Jupyter Notebook to answer these business questions. Your designed and implemented pipeline will be submitted to your client company.(Canvas)

1.0.3 About the Dataset

In this dataset, historical sales of a grocery corporation have been documented over three months in three different locations. Predictive data analytics approaches are simple to use with this dataset. The growth of supermarkets in most populated cities is increasing and market competition is also high. This dataset is one that contains the historical sales of a supermarket company over a period of 3 months.With this dataset, predictive analytics methods are straightforward to apply.

1.1 1-Importing Necessary Libraries

In this step I imported the necessery libraries that I need like pandas,Numpy,..

[30]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import sklearn.linear_model
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
import sklearn.metrics
```

1.2 2-loading the dataset

I upload the dataset from https://www.kaggle.com/aungpyaeap/supermarket-sales into my jupyter-hub notebook.

[19]:
```
df = pd.read_csv("../../datasets/supermarket_sales - Sheet1.csv")

df.head(5)
```

[19]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet	522.83	4.761905	26.1415	9.1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	76.40	4.761905	3.8200	9.6
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card	324.31	4.761905	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905	23.2880	8.4
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	604.17	4.761905	30.2085	5.3

3- Head function I run df.head function in order to see the first 5 rows of my dataset and make sure this dataset is the right one that I need for my project

[20]:
```
df.head()
```

[20]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet	522.83	4.761905	26.1415	9.1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	76.40	4.761905	3.8200	9.6
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card	324.31	4.761905	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905	23.2880	8.4
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	604.17	4.761905	30.2085	5.3

```
2  13:23  Credit card  324.31                4.761905       16.2155     7.4
3  20:33      Ewallet  465.76                4.761905       23.2880     8.4
4  10:37      Ewallet  604.17                4.761905       30.2085     5.3
```

1.3 4-splitting data into training and testing set

[33]:
```
x = df.drop(["Branch"], axis=1)
y = df["Branch"]
X_train, X_test, y_train, y_test = sklearn.model_selection.train_test_split( x,␣
 ↪y)

print("df:", df.shape)
print("X_train:", X_train.shape)
print("X_test:", X_test.shape)
print("y_train:", y_train.shape)
print("y_test:", y_test.shape)
```

```
df: (1000, 17)
X_train: (750, 16)
X_test: (250, 16)
y_train: (750,)
y_test: (250,)
```

1.4 4-Exploring the data set

In order to clean the dataset i use df.isnull code but there is no null article in the dataset to be cleaned.

[5]:
```
df.isnull().sum()
```

[5]:
```
Invoice ID                0
Branch                    0
City                      0
Customer type             0
Gender                    0
Product line              0
Unit price                0
Quantity                  0
Tax 5%                    0
Total                     0
Date                      0
Time                      0
Payment                   0
cogs                      0
gross margin percentage   0
```

```
gross income                0
Rating                      0
dtype: int64
```

1.5 5-The Describe function

I used describe function in order to have the statistical summery of the dataframe or series. This includes count, mean, min-max, and percentile values of columns.

[6]:
```
df.describe()
```

[6]:
```
         Unit price     Quantity       Tax 5%        Total        cogs  \
count  1000.000000  1000.000000  1000.000000  1000.000000  1000.00000
mean     55.672130     5.510000    15.379369   322.966749   307.58738
std      26.494628     2.923431    11.708825   245.885335   234.17651
min      10.080000     1.000000     0.508500    10.678500    10.17000
25%      32.875000     3.000000     5.924875   124.422375   118.49750
50%      55.230000     5.000000    12.088000   253.848000   241.76000
75%      77.935000     8.000000    22.445250   471.350250   448.90500
max      99.960000    10.000000    49.650000  1042.650000   993.00000

       gross margin percentage  gross income      Rating
count              1000.000000   1000.000000  1000.00000
mean                  4.761905     15.379369     6.97270
std                   0.000000     11.708825     1.71858
min                   4.761905      0.508500     4.00000
25%                   4.761905      5.924875     5.50000
50%                   4.761905     12.088000     7.00000
75%                   4.761905     22.445250     8.50000
max                   4.761905     49.650000    10.00000
```

1.6 Step6: Questions and answers:

Question1-How many costumers of this supermarket are female ? The codes showes that 501 costumers of this supermarket are woman(Female)

[7]:
```
df['Gender'].value_counts()
```

[7]:
```
Female    501
Male      499
Name: Gender, dtype: int64
```

Qestion 2:what type of product sold the most in the supermarket? Food and beverage are sold the most in supermarket

Question 3:what is the most expensive products per unit? The most expensive product is fashion products

[27]:
```
group_prodLine_sum = df.groupby(['Product line']).sum()


group_prodLine_mean = df.groupby(['Product line']).mean()

group_prodLine_sum
```

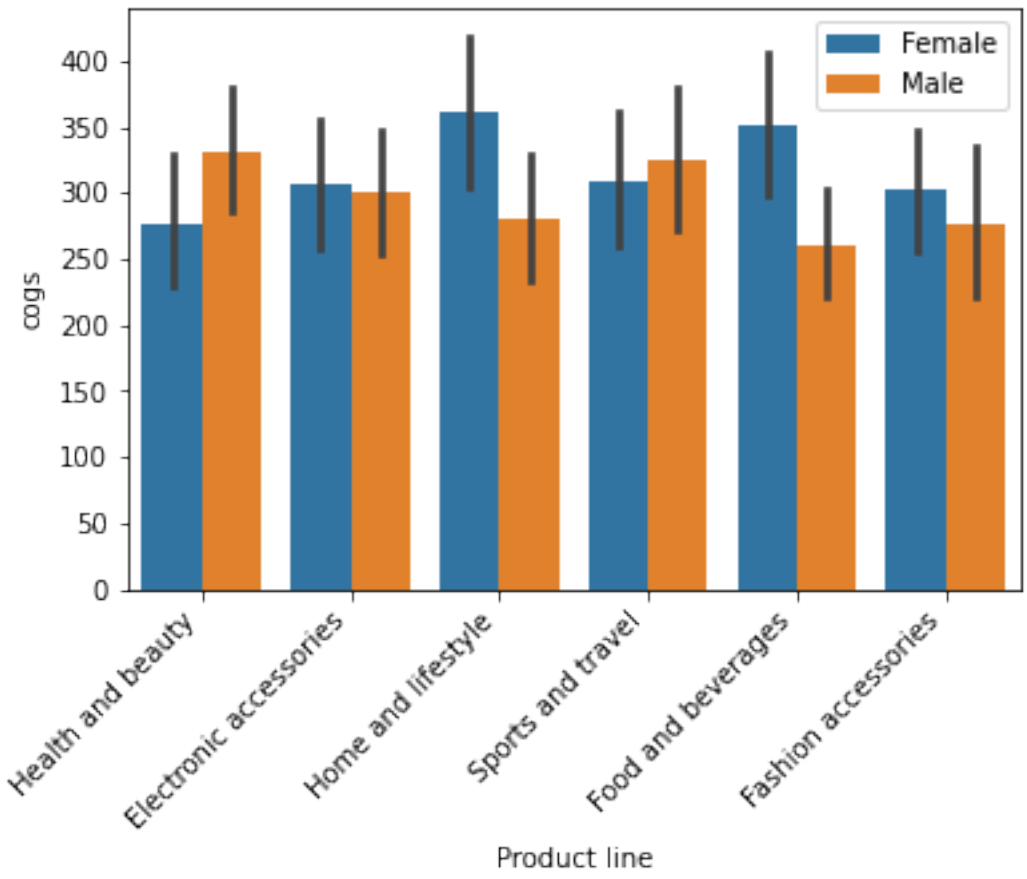
[27]:

Product line	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income	Rating
Electronic accessories	9103.77	971	2587.5015	54337.5315	51750.03	809.523810	2587.5015	1177.2
Fashion accessories	10173.35	902	2585.9950	54305.8950	51719.90	847.619048	2585.9950	1251.2
Food and beverages	9745.54	952	2673.5640	56144.8440	53471.28	828.571429	2673.5640	1237.7
Health and beauty	8337.88	854	2342.5590	49193.7390	46851.18	723.809524	2342.5590	1064.5
Home and lifestyle	8850.71	911	2564.8530	53861.9130	51297.06	761.904762	2564.8530	1094.0
Sports and travel	9460.88	920	2624.8965	55122.8265	52497.93	790.476190	2624.8965	1148.1

Question4:Which products sold the most based on gender? As the chart shows,women are more intrested in fashion ,food and beverage,home and lifestyle products while men are more intrested in sports and travel,health and beauty.

[32]:
```
sns.barplot(x = 'Product line', y = 'cogs', hue = 'Gender', data = df)
plt.legend(loc = 'upper right')
plt.xticks(rotation = 45, ha = 'right')
```

[32]:
```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Health and beauty'),
  Text(1, 0, 'Electronic accessories'),
  Text(2, 0, 'Home and lifestyle'),
  Text(3, 0, 'Sports and travel'),
  Text(4, 0, 'Food and beverages'),
  Text(5, 0, 'Fashion accessories')])
```



Question5:What is the best payment term based on each city? The results showes that: In Mandalay and Yaygon cities, Ewallet is the most popular one,cash and credit card comes after that. In naypyitaw cash Has the highest amount and credit card has the lowest

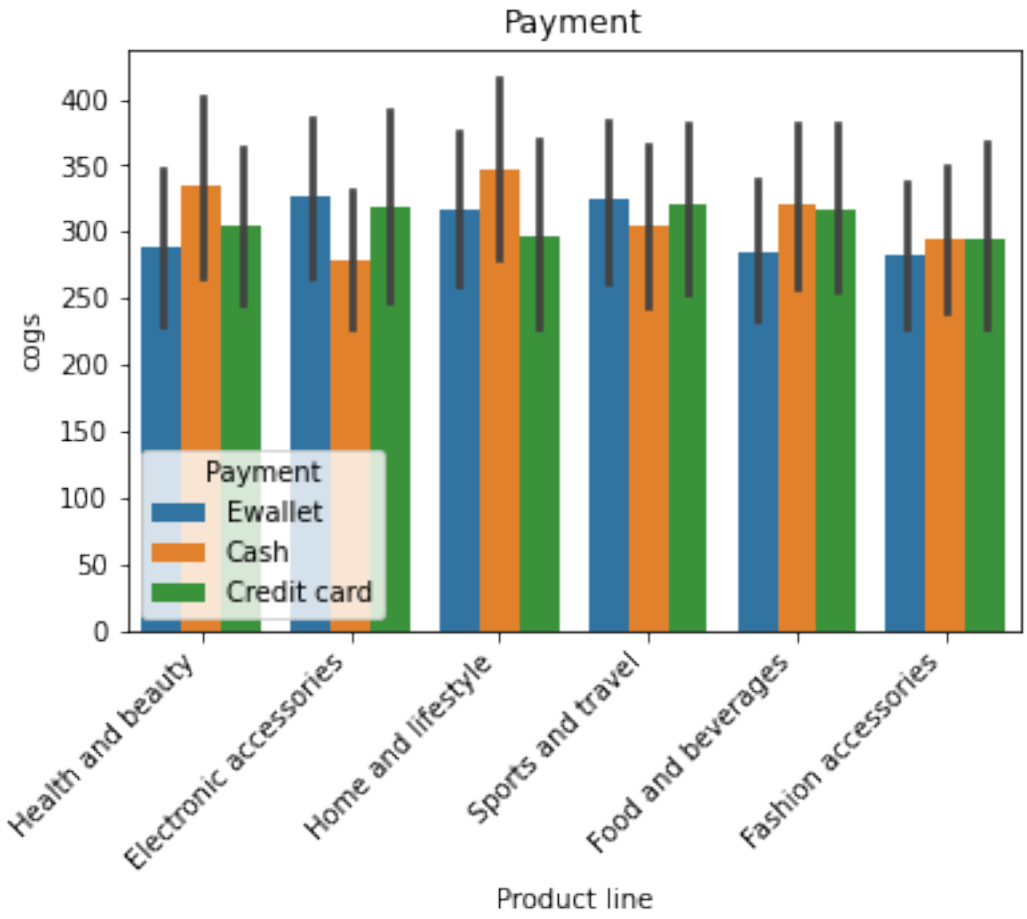
[14]:
```
Type_of_costumer = df.groupby('City')['Payment'].value_counts()
Type_of_costumer
```

[14]:
```
City       Payment
Mandalay   Ewallet        113
           Cash           110
           Credit card    109
Naypyitaw  Cash           124
           Ewallet        106
           Credit card     98
Yangon     Ewallet        126
           Cash           110
           Credit card    104
Name: Payment, dtype: int64
```

Question 7 : How customers use Payment methods on different products? As the chart showes,people pay more in cash for health and beauty,home and lifestyle,food and beverage

[16]:
```
plt.title('Payment')
sns.barplot(x = 'Product line', y = 'cogs', hue = 'Payment', data = df)
plt.xticks(rotation = 45, ha = 'right')
```

[16]:
```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Health and beauty'),
  Text(1, 0, 'Electronic accessories'),
  Text(2, 0, 'Home and lifestyle'),
  Text(3, 0, 'Sports and travel'),
  Text(4, 0, 'Food and beverages'),
  Text(5, 0, 'Fashion accessories')])
```



Question8: which day of the week has the most purchased items happend? Most purchased frequences happens on Saturday

[17]:
```
df['Date'] = pd.to_datetime(df['Date'], format='%m/%d/%Y') # convert column␣
↪'Date' into datetime type
df["week_days"] = df["Date"].dt.day_name()
df['week_days'].value_counts()
```

[17]:
```
Saturday     164
Tuesday      158
Wednesday    143
Friday       139
Thursday     138
Sunday       133
Monday       125
Name: week_days, dtype: int64
```

Question9 :In which hour of the day this supermarket has the best selling ? The most often purchased hour is between 19h and 20h.

[18]:
```
df['Hours_only'] = pd.to_datetime(df['Time'], format='%H:%M') # convert column␣
↪'Time' into datetime type
df['Hours_only'] = df['Hours_only'].dt.hour # Keeping only hours from datetime
df['Hours_only'].value_counts() # Count frequence per appearance
```

[18]:
```
19    113
13    103
15    102
10    101
18     93
11     90
12     89
14     83
16     77
20     75
17     74
Name: Hours_only, dtype: int64
```

Question 10:what is the number of normal and membership costumer that each city has?

[24]:
```
customertype_per_city = df.groupby('City')['Customer type'].value_counts()
customertype_per_city
```

[24]:
```
City       Customer type
Mandalay   Normal           167
           Member           165
Naypyitaw  Member           169
           Normal           159
```

```
Yangon     Normal           173
           Member           167
Name: Customer type, dtype: int64
```

[ ]:

[ ]: