WALMART PROJECT ¶

Step 1. Data exploration

In [45]:
```
#importing dependencies

import pandas as pd

#mysql toolkit
import pymysql #this will work as adapter
from sqlalchemy import create_engine
```

In [2]:
```
df= pd.read_csv('Walmart.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	invoice_id	Branch	City	category	unit_price	quantity	date	time	payn
0	1	WALM003	San Antonio	Health and beauty	$74.69	7.0	05/01/19	13:08:00	
1	2	WALM048	Harlingen	Electronic accessories	$15.28	5.0	08/03/19	10:29:00	
2	3	WALM067	Haltom City	Home and lifestyle	$46.33	7.0	03/03/19	13:23:00	
3	4	WALM064	Bedford	Health and beauty	$58.22	8.0	27/01/19	20:33:00	
4	5	WALM013	Irving	Sports and travel	$86.31	7.0	08/02/19	10:37:00	

In [4]:
```
df.tail()
```

Out[4]:

	invoice_id	Branch	City	category	unit_price	quantity	date	time
10046	9996	WALM056	Rowlett	Fashion accessories	$37	3.0	03/08/23	10:10:00
10047	9997	WALM030	Richardson	Home and lifestyle	$58	2.0	22/02/21	14:20:00
10048	9998	WALM050	Victoria	Fashion accessories	$52	3.0	15/06/23	16:00:00
10049	9999	WALM032	Tyler	Home and lifestyle	$79	2.0	25/02/21	12:25:00
10050	10000	WALM069	Rockwall	Fashion accessories	$62	3.0	26/09/20	9:48:00

In [5]:
```
df.columns
```

Out[5]:
```
Index(['invoice_id', 'Branch', 'City', 'category', 'unit_price', 'quantit
y',
       'date', 'time', 'payment_method', 'rating', 'profit_margin'],
      dtype='object')
```

In [6]:
```
df.dtypes
```

Out[6]:
```
invoice_id         int64
Branch            object
City              object
category          object
unit_price        object
quantity         float64
date              object
time              object
payment_method    object
rating           float64
profit_margin    float64
dtype: object
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10051 entries, 0 to 10050
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   invoice_id      10051 non-null  int64
 1   Branch          10051 non-null  object
 2   City            10051 non-null  object
 3   category        10051 non-null  object
 4   unit_price      10020 non-null  object
 5   quantity        10020 non-null  float64
 6   date            10051 non-null  object
 7   time            10051 non-null  object
 8   payment_method  10051 non-null  object
 9   rating          10051 non-null  float64
 10  profit_margin   10051 non-null  float64
dtypes: float64(3), int64(1), object(7)
memory usage: 863.9+ KB
```

In [8]:
```
df.describe()
```

Out[8]:

	invoice_id	quantity	rating	profit_margin
count	10051.000000	10020.000000	10051.000000	10051.000000
mean	5025.741220	2.353493	5.825659	0.393791
std	2901.174372	1.602658	1.763991	0.090669
min	1.000000	1.000000	3.000000	0.180000
25%	2513.500000	1.000000	4.000000	0.330000
50%	5026.000000	2.000000	6.000000	0.330000
75%	7538.500000	3.000000	7.000000	0.480000
max	10000.000000	10.000000	10.000000	0.570000

In [9]:
```
df.shape
```

Out[9]:
```
(10051, 11)
```

In [10]:
```
df["Branch"].unique
```

Out[10]:
```
<bound method Series.unique of 0        WALM003
1        WALM048
2        WALM067
3        WALM064
4        WALM013
          ...
10046    WALM056
10047    WALM030
10048    WALM050
10049    WALM032
10050    WALM069
Name: Branch, Length: 10051, dtype: object>
```

In [11]:
```
df["Branch"].value_counts()
```

Out[11]:
```
WALM058    240
WALM009    238
WALM030    233
WALM069    224
WALM074    212
          ...
WALM013     57
WALM031     56
WALM034     56
WALM014     52
WALM092     51
Name: Branch, Length: 100, dtype: int64
```

In [12]:
```
df["City"].value_counts()
```

Out[12]:
```
Weslaco            399
Waxahachie         381
Port Arthur        240
Plano              238
Richardson         233
                  ...
Irving              57
Lewisville          56
College Station     56
Amarillo            52
Lake Jackson        51
Name: City, Length: 98, dtype: int64
```

In [13]:
```
df["category"].value_counts()
```

Out[13]:
```
Fashion accessories       4579
Home and lifestyle        4561
Electronic accessories     419
Food and beverages         174
Sports and travel          166
Health and beauty          152
Name: category, dtype: int64
```

In [14]:
```
df["date"].max()
```

Out[14]:
```
'31/12/23'
```

In [15]:
```
df["date"].min()
```

Out[15]:
```
'01/01/19'
```

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
invoice_id         0
Branch             0
City               0
category           0
unit_price        31
quantity          31
date               0
time               0
payment_method     0
rating             0
profit_margin      0
dtype: int64
```

Data Cleaning

In [46]:
```
# drop all the duplicates
```

In [18]:
```
df.drop_duplicates(inplace=True)
```

In [19]:
```
df.duplicated().sum()
```

Out[19]:
```
0
```

In [20]:
```
#dropping all rows with missing records
```

In [21]:
```
df.dropna(inplace= True)
```

In [22]:
```
df.isnull().sum()
```

Out[22]:
```
invoice_id        0
Branch            0
City              0
category          0
unit_price        0
quantity          0
date              0
time              0
payment_method    0
rating            0
profit_margin     0
dtype: int64
```

In [23]:
```
df.shape
```

Out[23]:
```
(9969, 11)
```

In [24]:
```
# converting unit_price to int dtype
```

In [25]:
```
df["unit_price"]=df["unit_price"].str.replace('$','').astype(float)
```

```
C:\Users\Sanjay\AppData\Local\Temp\ipykernel_24448\1308059094.py:1: Futur
eWarning: The default value of regex will change from True to False in a
future version. In addition, single character regular expressions will *n
ot* be treated as literal strings when regex=True.
  df["unit_price"]=df["unit_price"].str.replace('$','').astype(float)
```

In [26]:
```
df.head()
```

Out[26]:

	invoice_id	Branch	City	category	unit_price	quantity	date	time	payn
0	1	WALM003	San Antonio	Health and beauty	74.69	7.0	05/01/19	13:08:00	
1	2	WALM048	Harlingen	Electronic accessories	15.28	5.0	08/03/19	10:29:00	
2	3	WALM067	Haltom City	Home and lifestyle	46.33	7.0	03/03/19	13:23:00	
3	4	WALM064	Bedford	Health and beauty	58.22	8.0	27/01/19	20:33:00	
4	5	WALM013	Irving	Sports and travel	86.31	7.0	08/02/19	10:37:00	

In [27]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9969 entries, 0 to 9999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   invoice_id      9969 non-null   int64  
 1   Branch          9969 non-null   object 
 2   City            9969 non-null   object 
 3   category        9969 non-null   object 
 4   unit_price      9969 non-null   float64
 5   quantity        9969 non-null   float64
 6   date            9969 non-null   object 
 7   time            9969 non-null   object 
 8   payment_method  9969 non-null   object 
 9   rating          9969 non-null   float64
 10  profit_margin   9969 non-null   float64
dtypes: float64(4), int64(1), object(6)
memory usage: 934.6+ KB
```

Feature Engineering

In [28]:
```
df['total']=df['unit_price']*df['quantity']
```

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

Out[29]:

	invoice_id	Branch	City	category	unit_price	quantity	date	time	payn
0	1	WALM003	San Antonio	Health and beauty	74.69	7.0	05/01/19	13:08:00	
1	2	WALM048	Harlingen	Electronic accessories	15.28	5.0	08/03/19	10:29:00	
2	3	WALM067	Haltom City	Home and lifestyle	46.33	7.0	03/03/19	13:23:00	
3	4	WALM064	Bedford	Health and beauty	58.22	8.0	27/01/19	20:33:00	
4	5	WALM013	Irving	Sports and travel	86.31	7.0	08/02/19	10:37:00	

In [30]:
```
# importing dependencies
```

In [31]:
```
pip install pymysql
```

```
Requirement already satisfied: pymysql in c:\users\sanjay\anaconda3\lib\s
ite-packages (1.1.1)Note: you may need to restart the kernel to use updat
ed packages.
```

In [32]:
```
pip install sqlalchemy
```

```
Requirement already satisfied: sqlalchemy in c:\users\sanjay\anaconda3\li
b\site-packages (2.0.38)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\sanja
y\anaconda3\lib\site-packages (from sqlalchemy) (4.12.2)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\sanjay\anacon
da3\lib\site-packages (from sqlalchemy) (1.1.1)
Note: you may need to restart the kernel to use updated packages.
```

In [34]:
```
df.to_csv('walmart_clean_data.csc',index=False)
```

Establishing connection to MYSQL

In [35]:
```
#Mysql connection
```

In [37]:
```
import pandas
import pymysql
from sqlalchemy import create_engine
```

In [42]:
```
engine_mysql = create_engine("mysql+pymysql://root:sanju1198@127.0.0.1:330
try:
    engine_mysql
    print("connection succesfull")
except:
    print("connection failed")
```

```
connection succesfull
```

Exporting Data

In [40]:
```
#exporting data to mysql
```

In [43]:
```
df.to_sql(name="walmart",con=engine_mysql,if_exists="append",index=False)
```

Out[43]:
```
9969
```

In [ ]:
```

```