# dataxlab-internship-task4

November 6, 2025

[3]:
```
#Loading the dataset
import pandas as pd
import sqlite3
file_path="C:\\Users\\Jagan\\Downloads\\Train.csv"
df=pd.read_csv(file_path)
print(df.head())
```

```
   ID Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rating  \
0   1               D           Flight                    4                2
1   2               F           Flight                    4                5
2   3               A           Flight                    2                2
3   4               B           Flight                    3                3
4   5               C           Flight                    2                2

   Cost_of_the_Product  Prior_purchases Product_importance Gender  \
0                  177                3                low      F
1                  216                2                low      M
2                  183                4                low      M
3                  176                4             medium      M
4                  184                3             medium      F

   Discount_offered  Weight_in_gms  Reached.on.Time_Y.N
0                44           1233                    1
1                59           3088                    1
2                48           3374                    1
3                10           1177                    1
4                46           2484                    1
```

[11]:
```
print(df.columns)
```

```
Index(['ID', 'Warehouse_block', 'Mode_of_Shipment', 'Customer_care_calls',
       'Customer_rating', 'Cost_of_the_Product', 'Prior_purchases',
       'Product_importance', 'Gender', 'Discount_offered', 'Weight_in_gms',
       'Reached.on.Time_Y.N'],
      dtype='object')
```

[13]:
```
# Connect to SQLite (creates a local DB file)
conn = sqlite3.connect("bigmart.db")
```

```
# Load CSV into SQL table
df.to_sql("train_data", conn, if_exists="replace", index=False)
print("\nData successfully stored in SQLite database as 'train_data'!")

# Verify table
print("\n Preview from SQL table:")
print(pd.read_sql_query("SELECT * FROM train_data LIMIT 5;", conn))
```

```
Data successfully stored in SQLite database as 'train_data'!

 Preview from SQL table:
   ID Warehouse_block Mode_of_Shipment  Customer_care_calls  Customer_rating  \
0   1               D           Flight                    4                2
1   2               F           Flight                    4                5
2   3               A           Flight                    2                2
3   4               B           Flight                    3                3
4   5               C           Flight                    2                2

   Cost_of_the_Product  Prior_purchases Product_importance Gender  \
0                  177                3                low      F
1                  216                2                low      M
2                  183                4                low      M
3                  176                4             medium      M
4                  184                3             medium      F

   Discount_offered  Weight_in_gms  Reached.on.Time_Y.N
0                44           1233                    1
1                59           3088                    1
2                48           3374                    1
3                10           1177                    1
4                46           2484                    1
```

[8]:
```
df.to_sql("train_data", conn, if_exists="replace", index=False)
print("Data loaded into SQLite database as 'train_data'!")
```

```
Data loaded into SQLite database as 'train_data'!
```

[14]:
```
print("\n Column names:", df.columns.tolist())
```

```
 Column names: ['ID', 'Warehouse_block', 'Mode_of_Shipment',
'Customer_care_calls', 'Customer_rating', 'Cost_of_the_Product',
'Prior_purchases', 'Product_importance', 'Gender', 'Discount_offered',
'Weight_in_gms', 'Reached.on.Time_Y.N']
```

[15]:
```
['Item_Identifier', 'Item_Weight', 'Item_Fat_Content',
 'Item_Visibility', 'Item_Type', 'Item_MRP',
 'Outlet_Identifier', 'Outlet_Establishment_Year',
 'Outlet_Size', 'Outlet_Location_Type',
 'Outlet_Type', 'Item_Outlet_Sales']
```

[15]:
```
['Item_Identifier',
 'Item_Weight',
 'Item_Fat_Content',
 'Item_Visibility',
 'Item_Type',
 'Item_MRP',
 'Outlet_Identifier',
 'Outlet_Establishment_Year',
 'Outlet_Size',
 'Outlet_Location_Type',
 'Outlet_Type',
 'Item_Outlet_Sales']
```

[19]:
```
import pandas as pd
import sqlite3

file_path = "C:\\Users\\Jagan\\Downloads\\Train.csv"
df = pd.read_csv(file_path)
conn = sqlite3.connect("delivery_analysis.db")

df.to_sql("train_data", conn, if_exists="replace", index=False)
print(" Data successfully stored in SQLite database!")
```

```
Data successfully stored in SQLite database!
```

[21]:
```
# 1 Average cost of products shipped per warehouse block
query1 = """
SELECT Warehouse_block, ROUND(AVG(Cost_of_the_Product), 2) AS Avg_Cost
FROM train_data
GROUP BY Warehouse_block
ORDER BY Avg_Cost DESC;
"""
print("\n  Average Cost per Warehouse Block:")
print(pd.read_sql_query(query1, conn))

# 2 Shipment mode performance - total orders per shipment type
query2 = """
SELECT Mode_of_Shipment, COUNT(*) AS Total_Orders
FROM train_data
GROUP BY Mode_of_Shipment
ORDER BY Total_Orders DESC;
```

```
"""
print("\n Orders by Mode of Shipment:")
print(pd.read_sql_query(query2, conn))

# 3 Average customer rating by product importance
query3 = """
SELECT Product_importance, ROUND(AVG(Customer_rating), 2) AS Avg_Rating
FROM train_data
GROUP BY Product_importance
ORDER BY Avg_Rating DESC;
"""
print("\n Average Customer Rating by Product Importance:")
print(pd.read_sql_query(query3, conn))

# 4 High-value shipments - products costing above average
query4 = """
SELECT ID, Warehouse_block, Cost_of_the_Product, Discount_offered
FROM train_data
WHERE Cost_of_the_Product > (SELECT AVG(Cost_of_the_Product) FROM train_data)
ORDER BY Cost_of_the_Product DESC
LIMIT 10;
"""
print("\n Top 10 High-Value Shipments:")
print(pd.read_sql_query(query4, conn))

# 5 Create a view for delayed deliveries
query5 = """
CREATE VIEW IF NOT EXISTS delayed_deliveries AS
SELECT * FROM train_data
WHERE [Reached.on.Time_Y.N] = 0;
"""
conn.execute(query5)
print("\n View 'delayed_deliveries' created successfully!")

# 6 Preview from view
print("\n Delayed Deliveries (Sample):")
print(pd.read_sql_query("SELECT * FROM delayed_deliveries LIMIT 5;", conn))
```

```
 Average Cost per Warehouse Block:
  Warehouse_block  Avg_Cost
0               B    212.16
1               C    211.19
2               D    210.91
3               F    209.08
4               A    208.77
```

```
 Orders by Mode of Shipment:
  Mode_of_Shipment  Total_Orders
0             Ship          7462
1           Flight          1777
2             Road          1760

 Average Customer Rating by Product Importance:
  Product_importance  Avg_Rating
0             medium        3.00
1               high        2.99
2                low        2.98

 Top 10 High-Value Shipments:
     ID Warehouse_block  Cost_of_the_Product  Discount_offered
0  4552               B                  310                 5
1  4565               C                  310                 2
2  4820               F                  310                 1
3  5025               A                  310                 7
4  5041               D                  310                 7
5  5220               F                  310                 5
6  5256               F                  310                 4
7  5259               A                  310                 8
8  6317               C                  310                 8
9  6600               F                  310                 5

View 'delayed_deliveries' created successfully!

 Delayed Deliveries (Sample):
     ID Warehouse_block Mode_of_Shipment  Customer_care_calls  \
0  3136               B             Road                    6
1  3137               C             Road                    3
2  3138               F             Road                    4
3  3139               D             Road                    4
4  3141               A             Road                    3

   Customer_rating  Cost_of_the_Product  Prior_purchases Product_importance  \
0                5                  141                3             medium
1                3                  219                3                low
2                3                  161                3               high
3                5                  225                4             medium
4                3                  256                4             medium

  Gender  Discount_offered  Weight_in_gms  Reached.on.Time_Y.N
0      M                 8           5031                    0
1      M                 4           5956                    0
2      M                 7           4245                    0
3      M                 5           4622                    0
4      M                 3           4732                    0
```

[22]:
```
conn.close()
print("\n Database connection closed.")
```

```
 Database connection closed.
```

[23]:
```
SELECT Warehouse_block, ROUND(AVG(Cost_of_the_Product), 2) AS Avg_Cost
FROM train_data
GROUP BY Warehouse_block
ORDER BY Avg_Cost DESC;
```

```
  Cell In[23], line 1
    SELECT Warehouse_block, ROUND(AVG(Cost_of_the_Product), 2) AS Avg_Cost
           ^
SyntaxError: invalid syntax
```

[ ]:
```

```