

End to End Analysis Project Using Walmart Dataset – Complete Analysis Using SQL, Python, and Power BI

Objective

The objective is to ask certain questions based on the dataset and answer them using SQL. For Postgres Supabase will be used. It is available online and the architecture does not have to be maintained in the local system.

After answering the questions using Supabase EDA will be performed using Python. The data will be extracted from the Supabase database and then used in Python. This will be done using the Supabase API.

After processing the entire data and performing analysis the data will be extracted as a CSV file from Python and then a dashboard will be made using Power BI.

Data Dictionary

The dataset contains the following files

- 2019 sales
- 2020 sales
- 2021 sales
- Branch data
- Manager's data

Fields in 2019 sales, 2020 sales, and 2021 sales are as follows:

- Invoice ID
- Branch
- City
- Customer Type
- Gender
- Product Line
- Unit Price
- Quantity
- Tax 5%

- Total
- Time
- Payment
- Cogs
- Gross Margin Percentage
- Gross Income
- Rating
- Day of Transaction
- Month of Transaction
- Year of Transaction

The branch data contains the following fields:

- Branch code
- Branch Name

The manager's data contains the following fields:

- City
- City Manager First Name
- City Manager Last Name

Creating Tables in Supabase

- The tables 2019 sales, 2020 sales, 2021 sales are combined using union and a single table called fact_sales is produced
- The branch and manager tables are joined using 'Left Join' and 'walmart_sales' table is produced
- Finally a combined table with product line codes known as walmart_master_sales is produced

Queries

The following queries are answered in Supabase and I have included the query as well

In how many cities are we doing business?

Nine

```
SELECT DISTINCT city FROM walmart_master_sales;
```

In which city is each branch?

The distinct city and branch name are selected from the data using a query.

```
SELECT DISTINCT city, branch_name FROM walmart_master_sales;
```

What is the most selling product?

Electronic accessories

```
SELECT SUM(quantity) AS sum_quantity, product_line  
FROM walmart_master_sales  
GROUP BY product_line  
ORDER BY sum_quantity DESC;
```

What is the total revenue by month?

The total revenue for each month is found using the following query,

```
SELECT month_of_txn, SUM(total) AS total_revenue  
FROM walmart_master_sales  
GROUP BY month_of_txn  
ORDER BY total_revenue DESC;
```

Which month had the largest cogs?

Month 1

```
SELECT month_of_txn AS month,  
SUM(cogs) AS total_cogs  
FROM walmart_master_sales  
GROUP BY month  
ORDER BY total_cogs DESC;
```

Which product line had the largest revenue?

Food and beverages

```
SELECT product_line,  
SUM(total) AS total_revenue  
FROM walmart_master_sales  
GROUP BY product_line  
ORDER BY total_revenue DESC;
```

Which product had the largest tax?

Home and lifestyle

```
SELECT product_line,  
AVG("tax_5%") AS avg_tax  
FROM walmart_master_sales  
GROUP BY product_line  
ORDER BY avg_tax DESC;
```

Which branch sold more products than average products sold?

Branch A

```
SELECT branch,  
SUM(quantity) AS total_quantity  
FROM walmart_master_sales  
GROUP BY branch  
HAVING SUM(quantity) > (SELECT AVG(quantity) FROM walmart_master_sales);
```

What is the most common product line by gender?

Female, Fashion and Accessories

```
SELECT gender, product_line,  
COUNT(gender) AS total_count  
FROM walmart_master_sales  
GROUP BY gender, product_line  
ORDER BY total_count DESC;
```

How many unique customer types does the data have?

Two (Normal and Member)

```
SELECT DISTINCT customer_type  
FROM walmart_master_sales;
```

How many unique payment methods does the data have?

Three (Credit Card, Ewallet, Cash)

```
SELECT DISTINCT payment  
FROM walmart_master_sales;
```

What is the most common customer type?

Member is the most common customer type

```
SELECT
    customer_type,
    count(*) as count
FROM walmart_master_sales
GROUP BY customer_type
ORDER BY count DESC;
```

What is the gender of most of the customers?

Female

```
SELECT
    gender,
    COUNT(*) as gender_cnt
FROM walmart_master_sales
GROUP BY gender
ORDER BY gender_cnt DESC;
```

Which day of the week has best average rating?

Day 4 with a rating of 7.524

```
SELECT
    day_of_txn,
    AVG(rating) AS avg_rating
FROM walmart_master_sales
GROUP BY day_of_txn
ORDER BY avg_rating DESC
LIMIT 1;
```

What is the month with the highest average rating?

Month 2 with average rating of 7.071 and a rank of 1.

```
SELECT
    month_of_txn,
    AVG(rating) AS avg_rating,
    RANK() OVER (ORDER BY AVG(rating) DESC) AS rating_rank
FROM
    walmart_master_sales
GROUP BY
    month_of_txn;
```

The next step is to perform analysis using Python. For that the data will be fetched from Supabase using API.

How to use the Supabase API for connecting using Python is given in the Supabase documentation itself.

Data Analysis Using Dtale

The following code converts the Supabase table to a DataFrame

```
In [7]: supabase.table('wallmart_master_sales').select("*").execute().data
```

```
2024-03-21 14:26:47,065:INFO - HTTP Request: GET https://bzhpdyytwbztbsxigfvx.supabase.co/rest/v1/wallmart_master_sales?select=* HTTP/1.1 200 OK
```

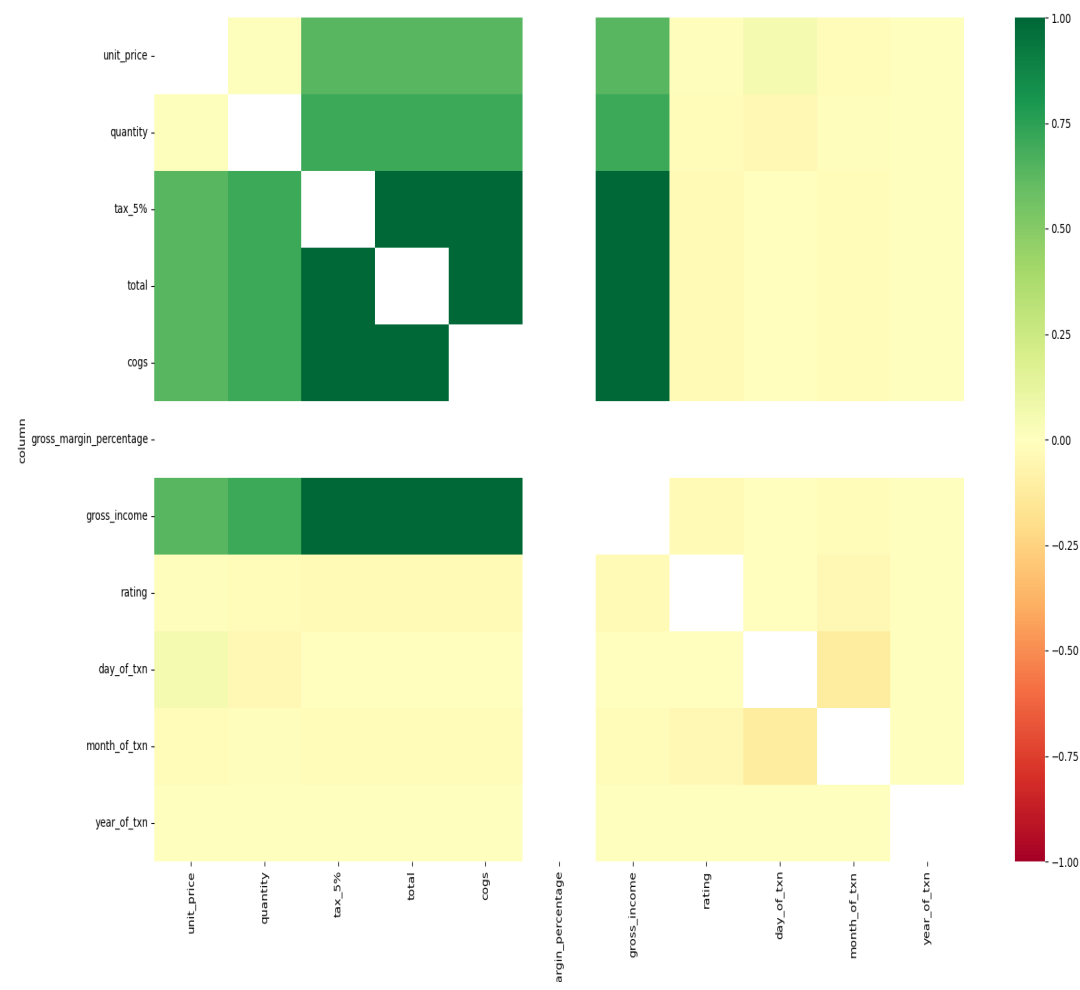
```
Out[7]: [{'invoice_id': '155-45-3814',
          'branch': 'C',
          'city': 'Naypyitaw',
          'customer_type': 'Member',
          'gender': 'Female',
          'product_line': 'Electronic accessories',
          'unit_price': 88.55,
          'quantity': 8,
          'tax_5%': 35.42,
          'total': 743.82,
          'time': '15:29:00',
          'payment': 'Ewallet',
          'cogs': 708.4,
          'gross_margin_percentage': 4.761904762,
          'gross_income': 35.42,
          'rating': 4.7,
          'day_of_txn': 19}
```

```
In [8]: import pandas as pd
```

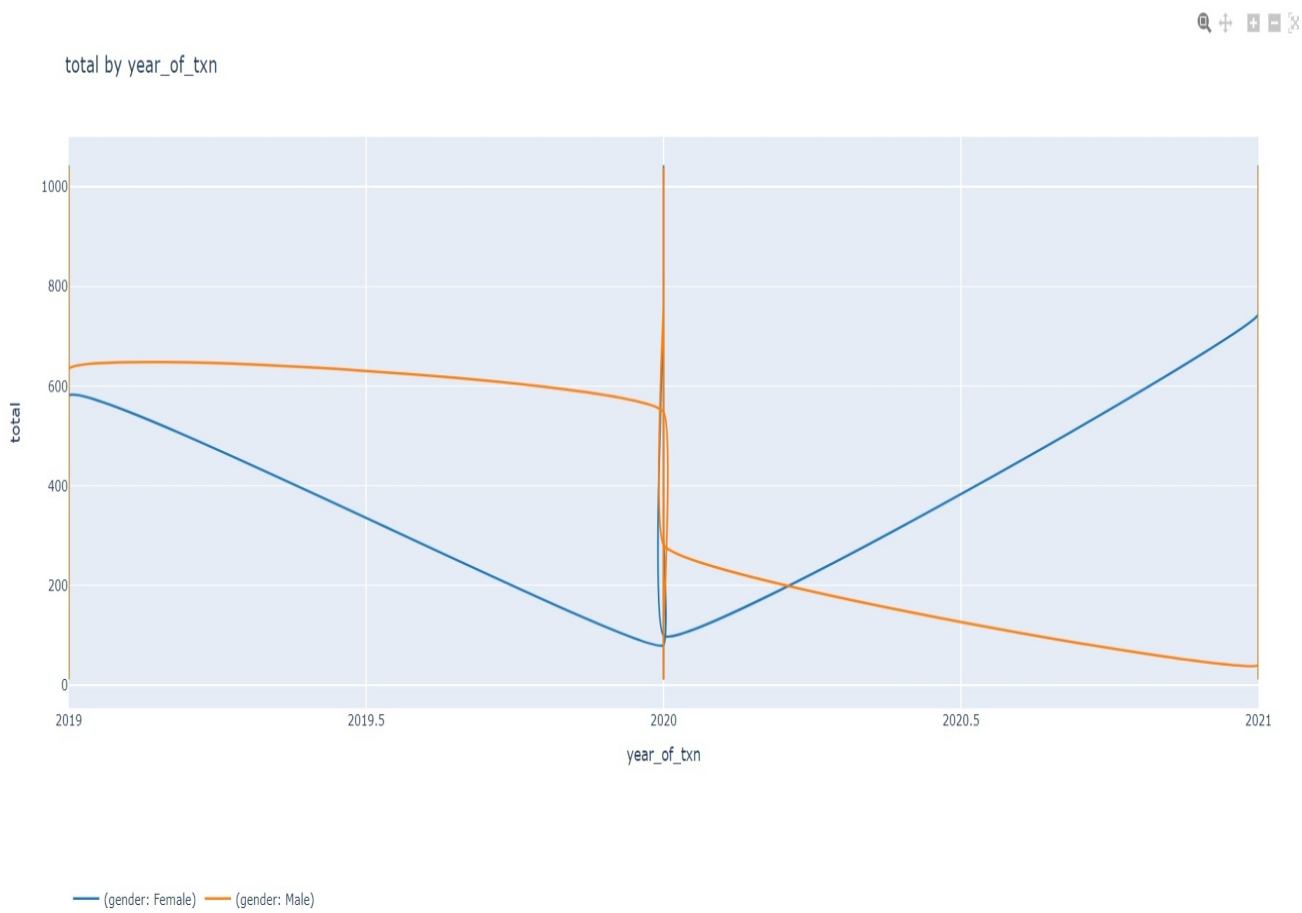
```
2024-03-21 14:26:50,574:INFO - NumExpr defaulting to 8 threads.
```

```
In [9]: api_response = supabase.table('wallmart_master_sales').select("*").execute().data
df = pd.DataFrame(api_response)
```

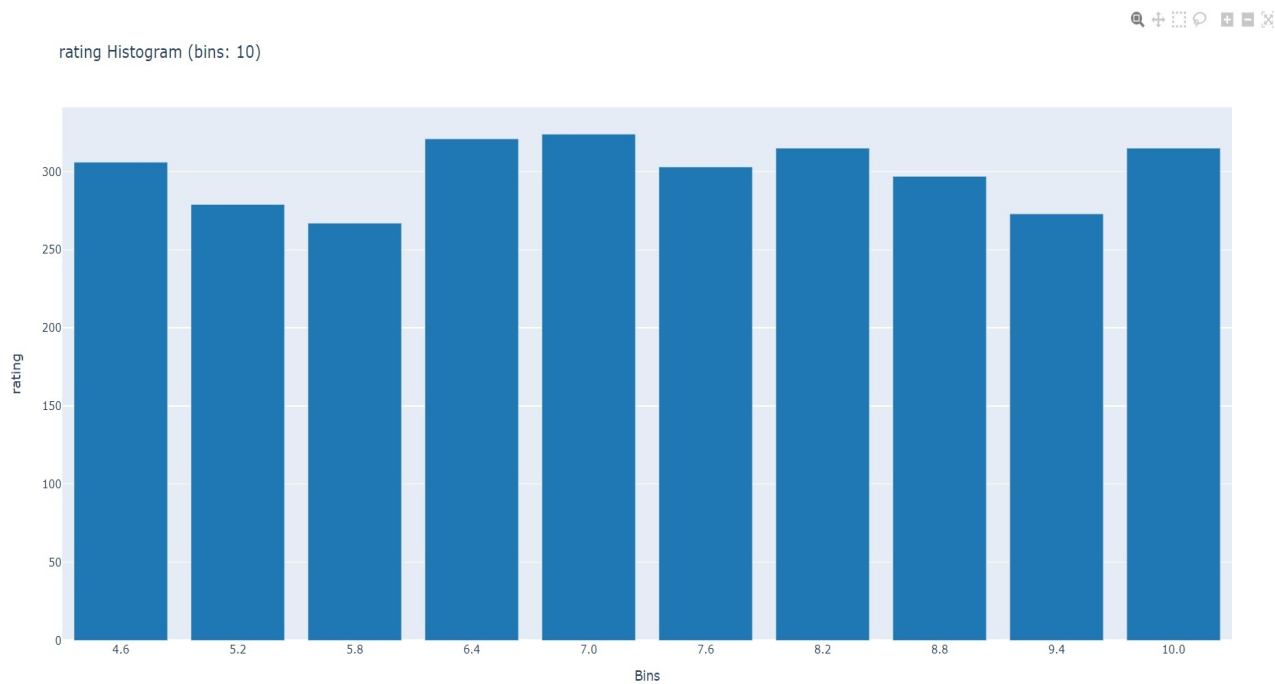
The following is the correlation matrix obtained using Dtale



Line Chart of Total Sales by Year of Transaction Grouped By Gender Generated Using Dtale



Histogram of Rating Generated Using Dtale



Analysis Using Ydata-profiling

Another framework that can be useful for analysis is ydata-profiling. The pandas-profiling has been renamed to ydata-profiling. The advantage of ydata-profiling is that the entire data can be exported as a html file and then visualized offline. It can be visualized within the Jupyter Notebook itself but I have exported the file so that it can be easily visualized and conclusions can be drawn out easily.


The following are the lines of code necessary to export the ydata-profiling report as a html report/file.

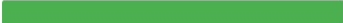
```
In [27]: from ydata_profiling import ProfileReport


2024-03-21 15:21:49,057:INFO - Pandas backend loaded 2.0.3
2024-03-21 15:21:49,088:INFO - Numpy backend loaded 1.24.3
2024-03-21 15:21:49,091:INFO - Pyspark backend NOT loaded
2024-03-21 15:21:49,093:INFO - Python backend loaded


In [28]: profile = ProfileReport(df, title="Profiling Report")

In [29]: profile.to_file('report.html')

Summarize dataset: 100%  97/97 [00:14<00:00, 3.89it/s, Completed]

Generate report structure: 100%  1/1 [00:10<00:00, 10.23s/it]

Render HTML: 100%  1/1 [00:02<00:00, 2.45s/it]

Export report to file: 100%  1/1 [00:00<00:00, 49.88it/s]
```

Understanding the Report

- The data does not contain any missing values.
- There is skewness in the data.
- The total, cogs, tax values have right skewed distribution. The reason for the right skewed distribution is that the total, cogs, tax values occur more times in the lesser threshold compared to the higher values.
- The visualization of the report is available in the report.html file.

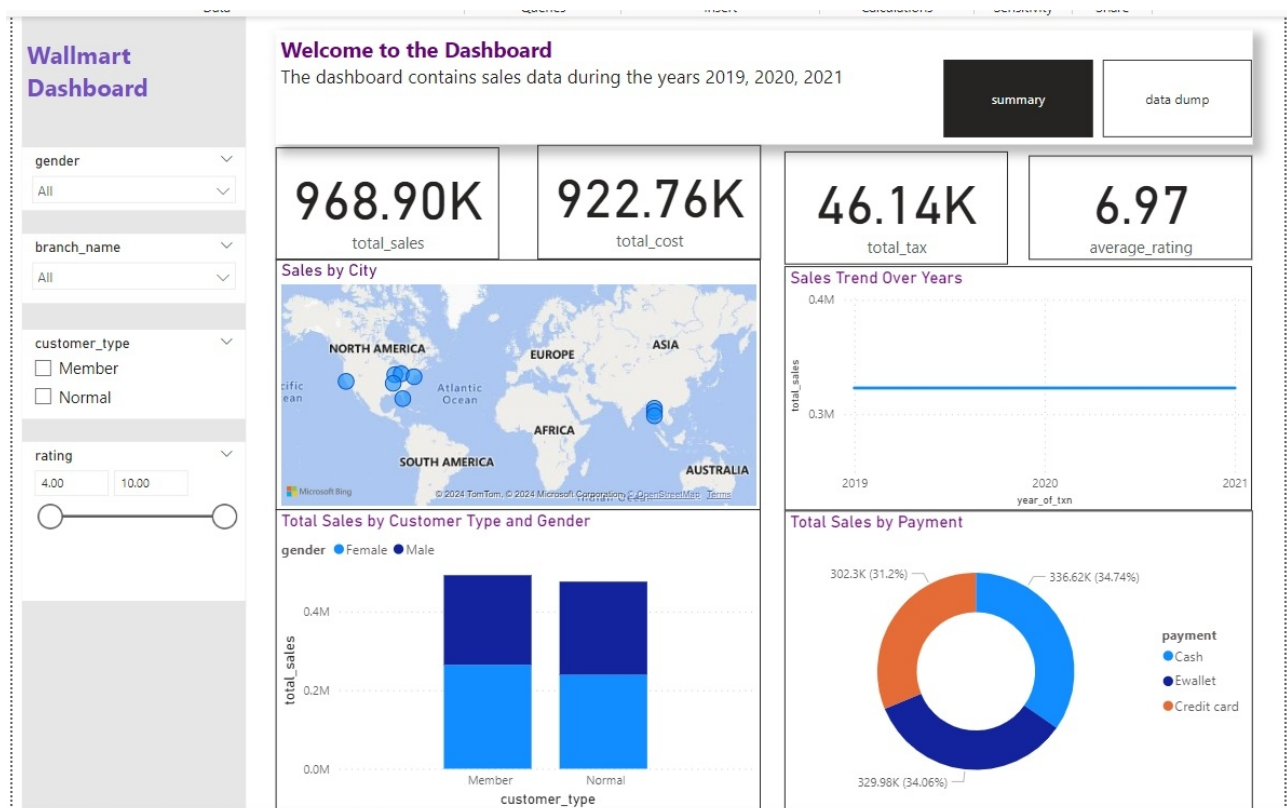
Converting the DataFrame to CSV File

The DataFrame is converted to a CSV file for using it in Power BI. I did not fetch the data from Supabase to Power BI using an API. Since the data is completely processed and then extracted in Python, the same processed data can be extracted as a simple CSV file and then used in Power BI. This allows not to repeat the same process of extracting the data again and again from Supabase.

Power BI Dashboard

The dashboard contains the summary and the data dump. Overall the dashboard has the following,

- A filter pane on the left side containing filters for gender, branch name, customer type and rating
- A navigation button at the top to switch between the summary and data dump
- Cards summarizing total sales, total cost, tax, and rating
- A map of sales by city
- A line chart of sales trend over the years
- A bar chart of total sales by customer type and gender
- A doughnut chart of total sales by payment
- The next page contains the data dump





invoice_id	branch	branch_name	city	Sum of cogs	customer_type	Sum of quantity	Sum of tax_5%	total_cost	total_sales	total_tax	average_rating
898-04-2717	A	Canada	Chicago	687.60	Normal	9	34.38	687.60	721.98	34.38	7.50
898-04-2717	A	Canada	Yangon	1,375.20	Normal	18	68.76	1,375.20	1,443.96	68.76	7.50
896-34-0956	A	Canada	Yangon	63.96	Normal	3	3.20	63.96	67.16	3.20	5.90
895-66-0685	B	Chile	Mandalay	162.72	Member	9	8.14	162.72	170.86	8.14	8.00
895-03-6665	B	Chile	Mandalay	985.77	Normal	27	49.29	985.77	1,035.06	49.29	4.20
894-41-5205	C	US	Naypyitaw	1,036.32	Normal	24	51.82	1,036.32	1,088.14	51.82	8.30
892-05-6689	A	Canada	Yangon	424.80	Normal	15	21.24	424.80	446.04	21.24	6.20
891-58-8335	B	Chile	Mandalay	621.81	Member	21	31.09	621.81	652.90	31.09	6.50
891-01-7034	B	Chile	Mandalay	1,344.78	Normal	18	67.24	1,344.78	1,412.02	67.24	6.70
889-04-9723	B	Chile	Mandalay	1,069.68	Member	12	53.48	1,069.68	1,123.16	53.48	7.80
888-02-0338	A	Canada	Yangon	708.21	Normal	27	35.41	708.21	743.62	35.41	5.90
887-42-0517	C	US	Naypyitaw	1,745.94	Normal	21	87.30	1,745.94	1,833.24	87.30	6.60
886-77-9084	C	US	Naypyitaw	1,725.36	Normal	24	86.27	1,725.36	1,811.63	86.27	5.50
886-54-6089	A	Canada	Yangon	205.74	Normal	18	10.29	205.74	216.03	10.29	7.70
886-18-2897	A	Canada	Yangon	848.40	Normal	15	42.42	848.40	890.82	42.42	4.50
885-56-0389	C	US	Naypyitaw	157.05	Member	3	7.85	157.05	164.90	7.85	4.00
885-17-6250	A	Canada	Yangon	239.22	Normal	3	11.96	239.22	251.18	11.96	7.30
884-80-6021	A	Canada	Yangon	2,204.10	Member	30	110.21	2,204.10	2,314.31	110.21	9.50
883-69-1285	B	Chile	Mandalay	299.52	Member	6	14.98	299.52	314.50	14.98	7.00
883-17-4236	C	US	Naypyitaw	86.34	Normal	6	4.32	86.34	90.66	4.32	7.20
882-40-4577	A	Canada	Yangon	807.12	Member	12	40.36	807.12	847.48	40.36	8.00
881-41-7302	C	US	Naypyitaw	194.97	Normal	3	9.75	194.97	204.72	9.75	4.50
880-46-5796	A	Canada	Yangon	2,307.60	Member	30	115.38	2,307.60	2,422.98	115.38	5.60
880-35-0356	A	Canada	Yangon	676.80	Member	9	33.84	676.80	710.64	33.84	4.80
878-30-2331	C	US	Naypyitaw	1,636.50	Member	30	81.83	1,636.50	1,718.33	81.83	7.10
877-22-3308	A	Canada	Yangon	476.10	Member	30	23.81	476.10	499.91	23.81	5.80
875-46-5808	B	Chile	Mandalay	777.00	Member	30	38.85	777.00	815.85	38.85	8.70
875-31-8302	B	Chile	Mandalay	280.14	Normal	3	14.01	280.14	294.15	14.01	9.60
Total				9,22,762.14		16530	46,138.11	9,22,762.14	9,68,900.25	46,138.11	6.97

Summary

The analytical questions are answered using Supabase. The exploratory data analysis is performed using frameworks Dtale and ydata-profiling and the report of the ydata-profiling is generated. A dashboard is made using Power BI.