Retail Sales Analytics Summary

Presentation by:

Thambidurai Sundaramoorthy

Introduction

In a competitive retail market, businesses need to track sales performance, understand customer behavior, and identify growth opportunities. This project aims to build an end-to-end analytics solution that enables insights generation from raw transactional data using a multi-tool approach.

Business Use Cases

- To analyze sales performance across different stores and staffs.
- To identify high-value customers and buying behavior patterns.
- To perform customer segmentation for targeted marketing.
- To monitor inventory levels and product performance.
- To create an interactive Power BI dashboard for management insights.

Excel – Data Cleaning & Preparation

1. Standardize Column Headers

- All the column headers have been updated in snake_case format for all the datasets across sheets.

2. Remove Duplicates

- No duplicate records found in the entire dataset which has been verified through the conditional formatting.

3. Handle Missing Values

- Null values found in multiple datasets where as manager_id is missing for one staff, order_status is missing for few orders and phone numbers are missing for few customers and replaced all the null values with NA.

Excel – Data Cleaning & Preparation

4. Data Type Conversion

- All the date column datatypes have been converted to the standard excel date format.
- Numerical fields like quantity, price are updated in the number format. Like wise, text formats have been also updated.

5. Data Validation

- Dropdown for order_status has been created using data validation list method to ensure consistency with the options (Delivered, Shipped, Processing, Packed).

Excel – Data Cleaning & Preparation

6. Create New Derived Columns

- Total_price for each items in the order_items have been calculated using the below-mentioned formula.

```
Total_price = (list_price * quantity) / discount
```

7. Merge Lookup Data

- Using VLOOKUP formula, product_names have been merged in to the order_items dataset using the product_id.

8. Create Basic Pivot Table

- Pivot table has been created to summarize the total sales by each category.
- Category_name has been added to the row field and total_price has been added to values by setting the value field settings to sum to get the total_price of each category.

Excel – Data Cleaning & Preparation

9. Sort and Filter for Outliers

- Created filter option and sorted the total_price from high to low.
- Highlighted the products with very high price above 10000/- and low price below 200/- using conditional formatting.

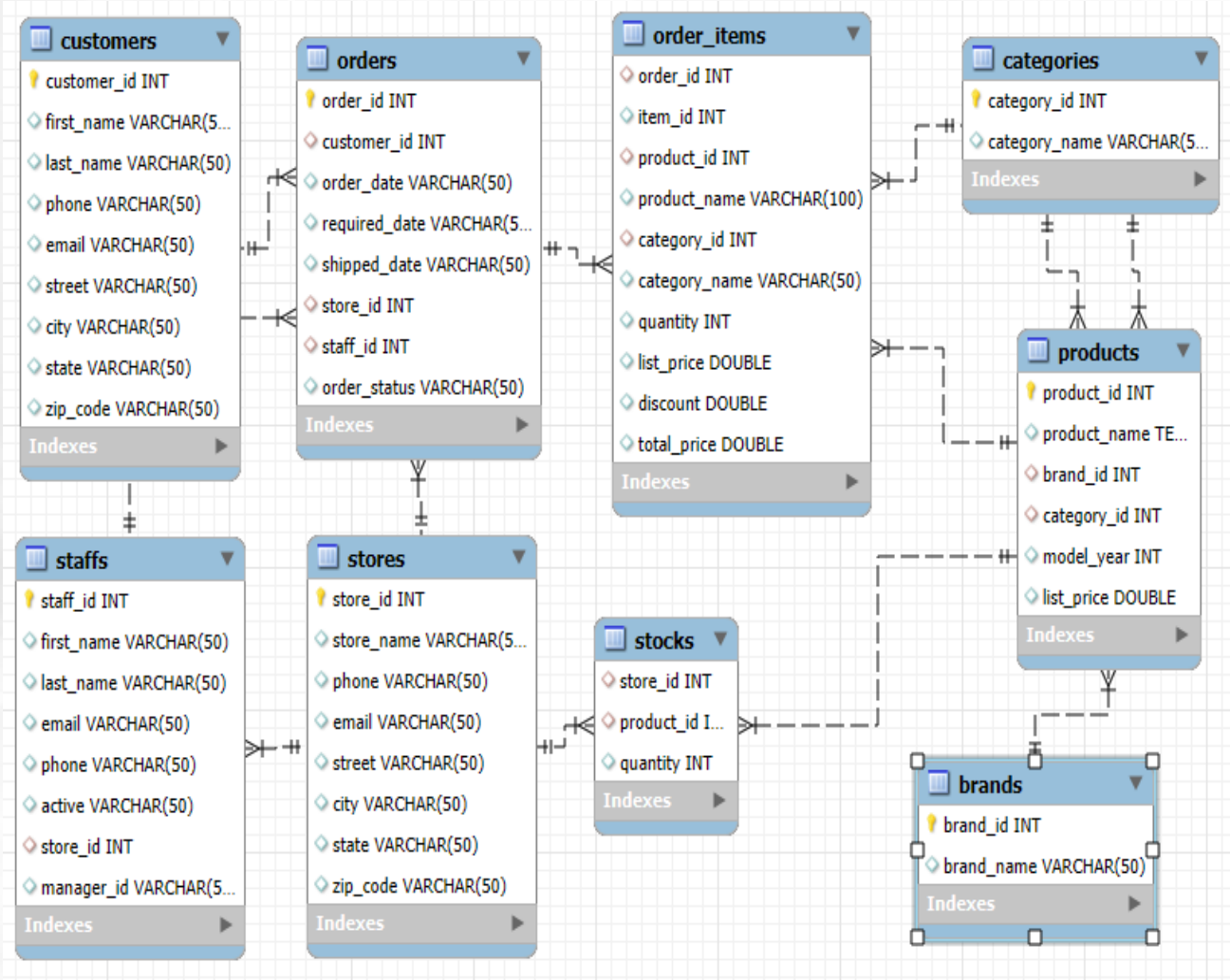
10. Prepare Final CSVs

- All the cleaned datasets have been saved in a separate sheets in CSV format to import in SQL.
- Created one separate excel file and updated all the datasets in the same file in different spread sheets along with the tasks.

SQL – Database Management and Querying

1. Create Tables Based on ERD

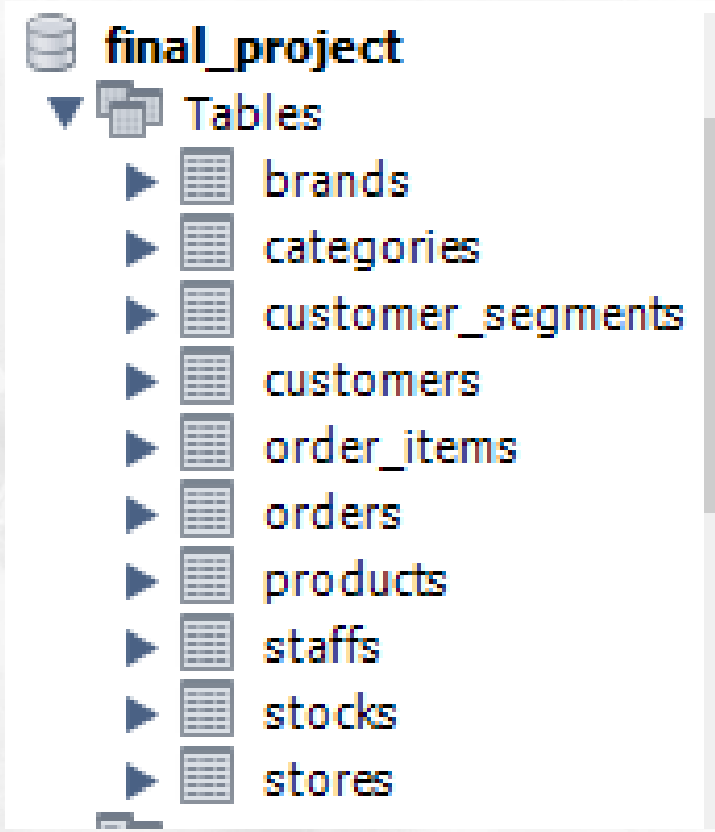
- Created different tables based on the ER diagram in the MySQL workbench by importing the datasets through table data import wizard.
- Unique, primary and foreign keys have been assigned to the columns as required to manage relationships between the tables.
- Datatypes have been updated for each column.



SQL – Database Management and Querying

2. Import CSVs into SQL

- Created the database in the name of final_project.
- Loaded the cleaned dataset CSV files in the MySQL workbench through table data import wizard method in the final_project database.
- Datatypes of each columns in the tables have been updated as per the standard format.



SQL – Database Management and Querying

3. Inner join for Order Details

```
Select orders.order_id,customer_id,order_items.item_id,order_items.product_id,products.product_name,
order_items.category_name,order_items.quantity,order_items.list_price,order_items.discount,
order_items.total_price,orders.order_status from orders inner join order_items on
orders.order_id=order_items.order_idinner join products on order_items.product_id=products.product_id;.
```

- Joined 3 different tables orders, order_items and products to view the detailed line items using inner join method.
- Using order_id, joined orders and order_items tables where as products and order_items tables have been joined using product_id.

SQL – Database Management and Querying

4. Total Sales by Stores

```
select stores.store_id,store_name,round(sum(total_price),2) as Total_sales from stores
inner join orders on stores.store_id=orders.store_id
inner join order_items on orders.order_id=order_items.order_id
group by store_id,store_name;
```

Stores	Total Revenue
Santa Cruz Bikes	1.79M
Baldwin Bikes	5.82M
Rowlett Bikes	0.96M

- Baldwin Bikes has the highest sales of 5.82M which is 68% from the overall revenue followed by Santa Cruz Bikes with 1.79M (21%) and then Rowlett Bikes with 0.96M (11%).

SQL – Database Management and Querying

5. Top 5 Selling Products

```
select product_id,product_name,sum(quantity) as Total_quantities_sold from order_items
group by product_id,product_name
order by Total_quantities_sold desc
limit 5;
```

Products	Quantities Sold
Surly Ice Cream Truck Frameset - 2016	167
Electra Cruiser 1 (24-Inch) - 2016	157
Electra Townie Original 7D EQ - 2016	156
Electra Girl's Hawaii 1 (20-inch) - 2015/2016	154
Trek Slash 8 27.5 - 2016	154

- Surly Ice Cream Truck Frameset – 2016 products is the highest sold in terms of quantity which is of 167.
- Electra brand has 3 products in top 5 in terms of highest quantities sold which is even above 150 quantities for each products.

SQL – Database Management and Querying

6. Customer Purchase Summary

```
select customers.customer_id,first_name,last_name,count(distinct orders.order_id) as Total_orders,sum(item_id)
as Total_items,round(sum(total_price),2) as Total_revenue from customers
inner join orders on customers.customer_id=orders.customer_id
inner join order_items on orders.order_id=order_items.order_id
group by customer_id,first_name,last_name;
```

- For each customer, retrieved the total orders placed, total items purchased and total revenue.
- There are 1445 customers overall where as maximum orders placed by single customer is 3 where as maximum items purchased is 36 and the highest revenue provided is 37800/-.
- This has been performed using inner join and group by methods.

SQL – Database Management and Querying

7. Segment Customers by Total Spend

```
select customers.customer_id,first_name,last_name,
round(sum(total_price),2) as Total_spent,
case
when sum(total_price) >=10000 then 'High'
when sum(total_price) >=1000 then 'Medium'
else 'Low’
end as Customer_segment from customers
inner join orders on customers.customer_id=orders.customer_id
inner join order_items on orders.order_id=order_items.order_id
group by customer_id,first_name,last_name;
```

- Customers have been classified in to 3 different categories based on the amount spent.
- The customers spent above 10000 has been classified as high where as customers spent between 1000 to 9999 has been classified as medium.
- Remaining customers spent below 1000 are classified as low.

SQL – Database Management and Querying

8. Staff Performance Analysis

```
select staffs.staff_id,first_name,last_name,count(orders.order_id) as
Total_handled_orders,round(sum(total_price),2) as Total_revenue_generated from staffs
inner join orders on staffs.staff_id=orders.staff_id
inner join order_items on orders.order_id=order_items.order_id
group by staff_id,first_name,last_name;
```

Staffs	Handled Orders	Revenue Generated
2	462	837375.78
3	544	952665.27
6	1615	2938714.12
7	1580	2887187.73
8	269	516667.66
9	252	445880.75

- Staff Id 6 has handled highest orders of 1615 and generated the maximum revenue of 2.93M where as Staff Id 9 has handled only 252 orders and generated revenue of 0.44M being the lowest.

SQL – Database Management and Querying

9. Stock Alert Query

```
select products.product_id,product_name,sum(quantity) as Total_stocks from stores
inner join stocks on stores.store_id=stocks.store_id
inner join products on stocks.product_id=products.product_id
group by product_id,product_name
having sum(quantity) < 10;
```

Products	Total Stocks
Trek Domane SLR Frameset - 2018	5
Electra Superbolt 1 20" - 2018"	9

- Out of 321 products, only 2 products have less than 10 quantity of stocks overall.

SQL – Database Management and Querying

10. Create Final Segmentation Table

- The customer segments table was exported directly from Python ML to MySQL and stored with the respective values instead of creating it manually in the MySQL workbench.
- The segment numbers have been assigned with type of segments like mentioned in the below table.

Segments	Type of Segments
0	At_risk
1	Loyal
2	New
3	Need_attention

Python and ML Tasks

1. Load Data from SQL

Task	Query
Pandas and SQL Libraries Installation	➢ !pip install pandas ➢ !pip install pymysql
Importing the required Libraries	➢ import pandas as pd ➢ import numpy as np ➢ import matplotlib.pyplot as plt ➢ import seaborn as sns ➢ import pymysql
SQL Database Connection Creation	pymysql.connect(host='127.0.0.1',user='root',passwd='password2924')
Customers dataset Retrieval	pd.read_sql_query("select * from final_project.customers",myconnection)
Orders dataset Retrieval	pd.read_sql_query("select * from final_project.orders",myconnection)
Order_items dataset Retrieval	pd.read_sql_query("select * from final_project.order_items",myconnection)

Python and ML Tasks

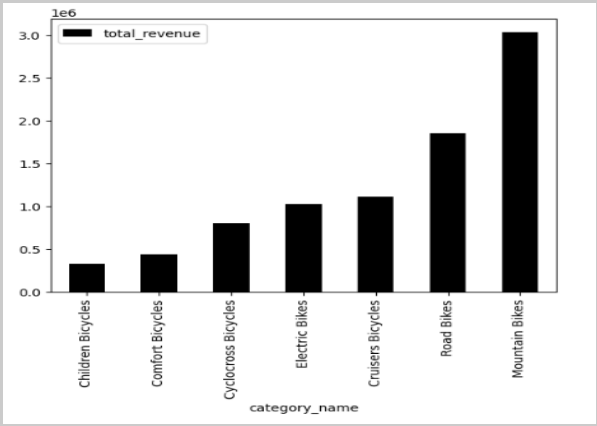
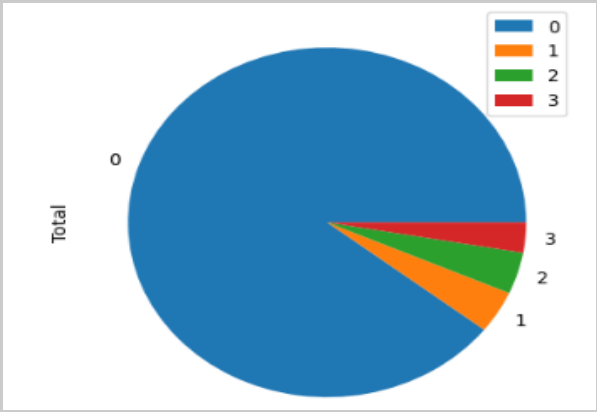
2. Basic EDA (Exploratory Data Analysis)

(a) Summarizing the dataset

Dataset	Task	Query
Customers	df.describe()	print(customers_df.describe())
	df.info()	print(customers_df.info())
	df.value_counts()	print(customers_df.value_counts())
Orders	df.describe()	print(orders_df.describe())
	df.info()	print(orders_df.info())
	df.value_counts()	print(orders_df.value_counts())
Order_items	df.describe()	print(order_items_df.describe())
	df.info()	print(order_items_df.info())
	df.value_counts()	print(order_items_df.value_counts())

Python and ML Tasks

(b) Basic Charts Plotting

Chart Types	Task	Query	Chart
Bar Chart	Grouping category name and total revenue calculation	order_items_df.groupby(["category_name"]).agg(total_revenue = ("total_price","sum")).sort_values("total_revenue",ascending = [True]).reset_index()	
	Plotting	category_revenue.plot(kind="bar",x="category_name",y="total_revenue",color="black")	
Pie Chart	Grouping order status and total orders calculation	orders_df.groupby(["order_status"]).agg(Total = ("order_status","count")).sort_values("Total",ascending =[False]).reset_index()	
	Plotting	order_status_count.plot(kind="pie",x="order_status",y="Total",color="black")	

Python and ML Tasks

(b) Basic Charts Plotting – Line Chart

Chart Types	Task	Query	Chart
Line Chart	Merging customers and orders dataset	pd.merge(customers_df,orders_df,on="customer_id")	
	Grouping state and total orders calculation	customer_orders.groupby(["state"]).agg(total_orders = ("order_id","count")).reset_index()	
	Plotting	plt.plot(state_orders["state"],state_orders["total_orders"]) plt.xlabel("state") plt.ylabel("total_orders") plt.title("Total Orders Per State") plt.show()	

Python and ML Tasks

3. Calculate RFM Features for Customers

Task	Query
Merging orders and order_items dataset	pd.merge(orders_df,order_items_df,on='order_id')
Merging merged_orders and customers dataset	pd.merge(merged_orders,customers_df,on='customer_id')
Freezing the date to identify the date difference	merged_dataset['order_date'].max() + pd.Timedelta(days=1)
RFM Calculation	merged_dataset.groupby('customer_id').agg({ 'order_date':lambda x: (freezed_date - x.max()).days, 'order_id':'nunique', 'total_price':'sum' }).reset_index()
Renaming RFM Columns	rfm.columns = ['customer_id','recency','frequency','monetary']

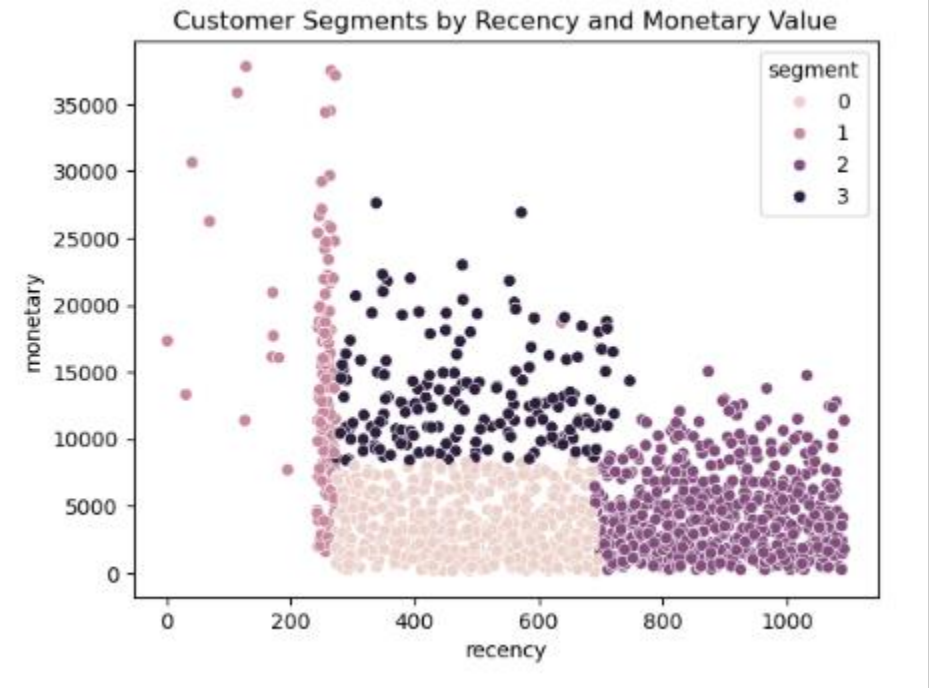
Python and ML Tasks

4. Customer Segmentation using KMeans

Task	Query
Importing the required Libraries	➢ from sklearn.preprocessing import MinMaxScaler ➢ from sklearn.cluster import KMeans
Normalize RFM using MinMaxScaler	MinMaxScaler().fit_transform(rfm[['recency','frequency','monetary']])
Applying KMeans to segment customers into 4 groups	KMeans(n_clusters=4,random_state=40)
	kmeans = KMeans(n_clusters=4,random_state=40)
Train (.fit) and label together (.predict) to cluster	rfm['segment'] = kmeans.fit_predict(rfm_normalized)
Retrieving rfm[segment] data value counts for each segment	print(rfm['segment'].value_counts())

Python and ML Tasks

Scatter Plotting for RFM

Query	Chart
sns.scatterplot(x='recency',y='monetary',hue='segment',data=rfm) plt.title('Customer Segments by Recency and Monetary Value') plt.show()	

Python and ML Tasks

5. Export Segmentation Results to SQL

Task	Query
SQL Library Installation	!pip install sqlalchemy pymysql
Importing the SQL Libraries	from sqlalchemy import create_engine
SQL Database Connection Creation	create_engine('mysql+pymysql://root:password2924@127.0.0.1/final_project')
Exporting segment datas from Python to SQL	rfm.to_sql('customer_segments',engine,if_exists='replace',index=False)

- Customer segments table has been exported from Python to SQL and table has been created in the database.

SQL – Database Management and Querying

Customer Segments Table

```
❖ alter table customer_segments add column
  type_of_segments varchar(50);

❖ update customer_segments
      set type_of_segments = case
                  when segment = 0 then 'at_risk'
                  when segment = 1 then 'loyal'
                  when segment = 2 then 'new'
                  when segment = 3 then 'need_attention'
                  else type_of_segments
            end
      where segment in (3,2,1,0);
```

- Created new column type_of_segments in SQL and updated it based on the segment numbers.
- Below shared is the sample data for reference.

Sample Data from the dataset

Cx Id	R	F	M	Segment	Segment Type
1	41	3	30644.78	1	loyal
95	640	1	13169.63	3	need_attention
96	749	1	9097.71	2	new
4	255	3	24197.81	1	loyal
100	307	1	5999.88	0	at_risk

Power BI – Visualization & Dashboarding

1. Connect Power BI to SQL

- All the dataset tables have been imported from SQL to Power BI by providing the server host number and the database name.
- Data types have been verified and updated accordingly.

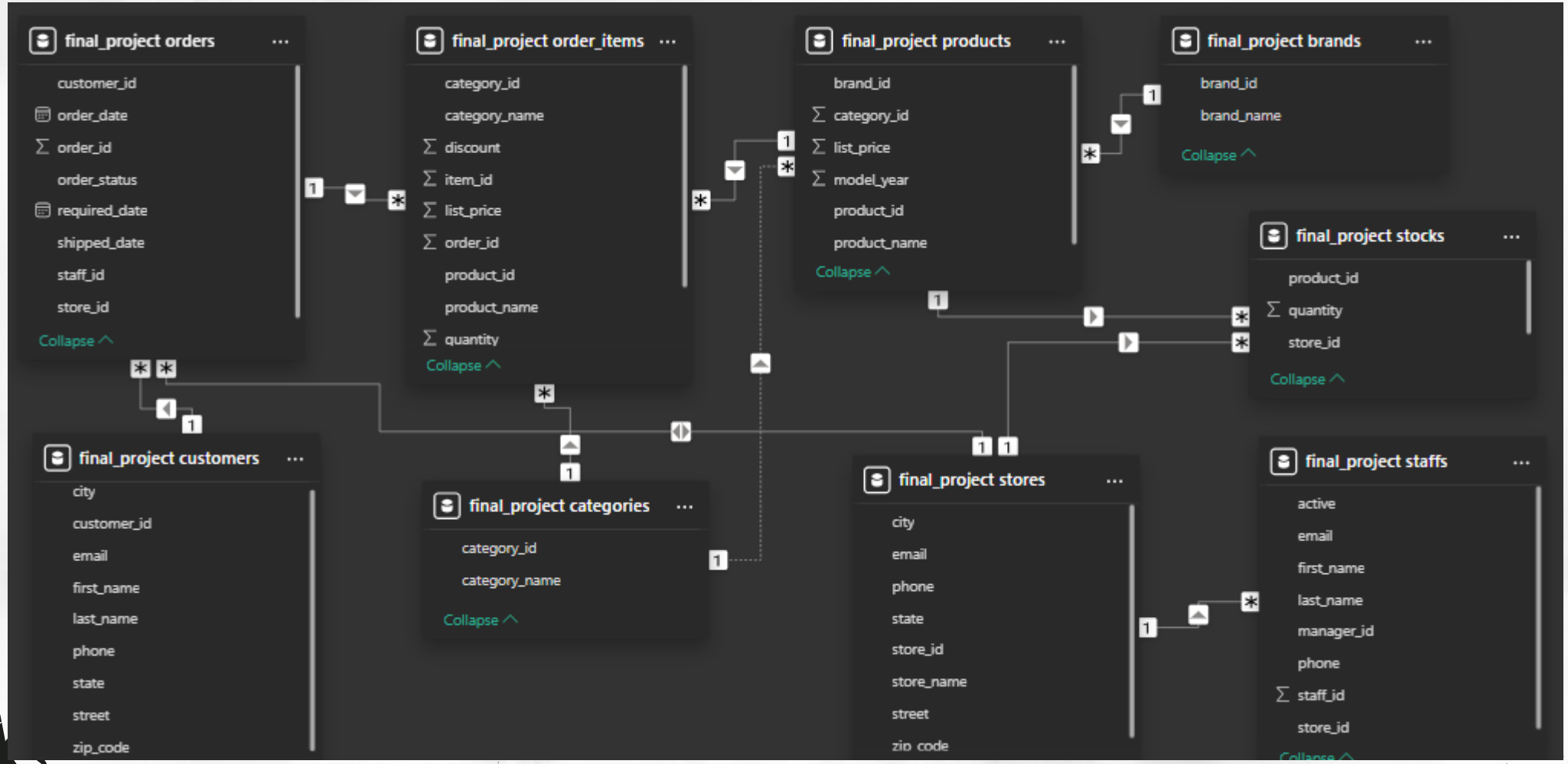
2. Create Relationship Between Tables

Model View	Relationship
1	One
*	Many

- The foreign key relationship between the dataset tables have been defined through the connecting common columns in tables.
- The model view snapshot with one/many relationship cardinality and cross-filter direction with single/both has been mapped accordingly.

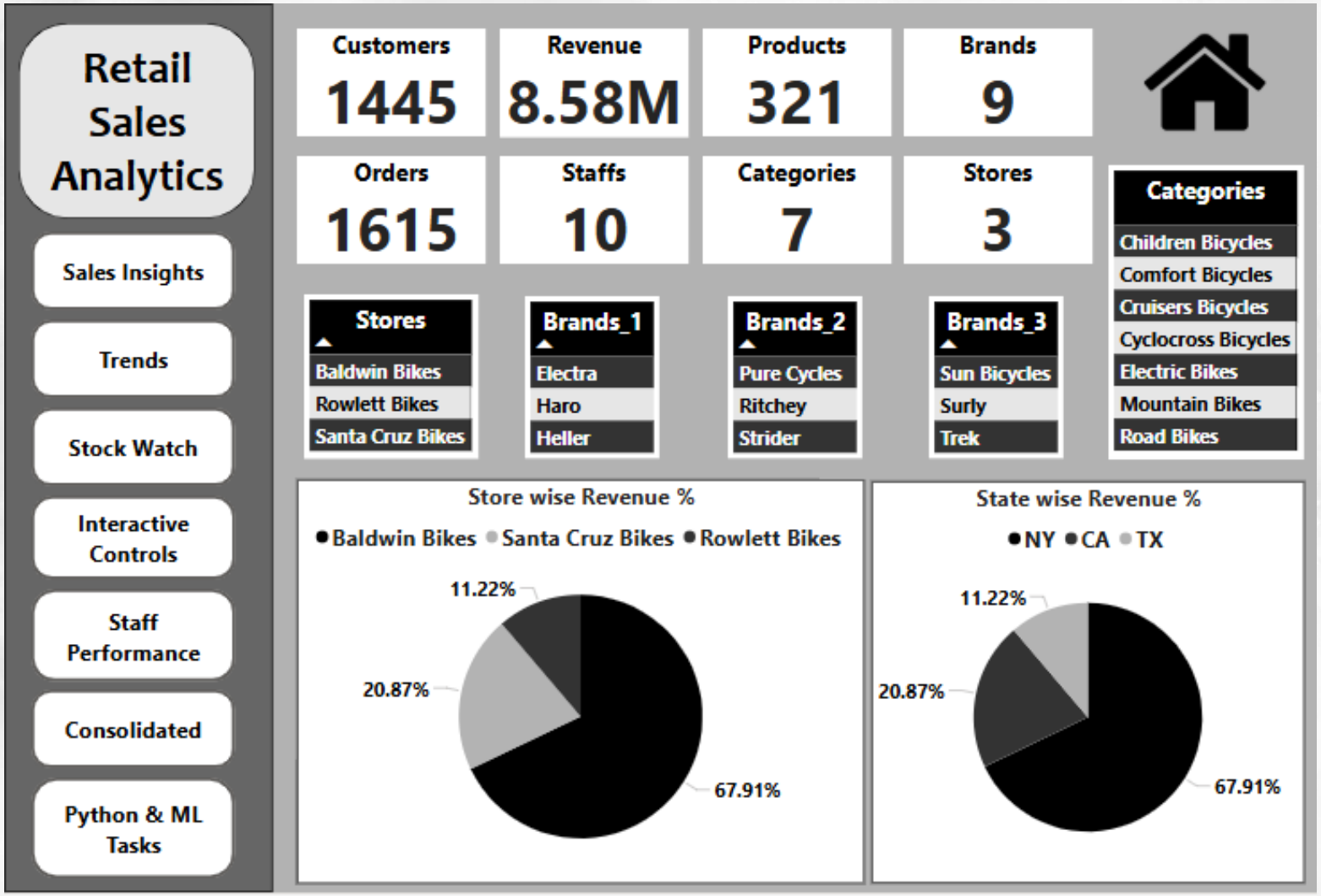
Power BI – Visualization & Dashboarding

Relationship Model View



Power BI – Visualization & Dashboarding

Home Page

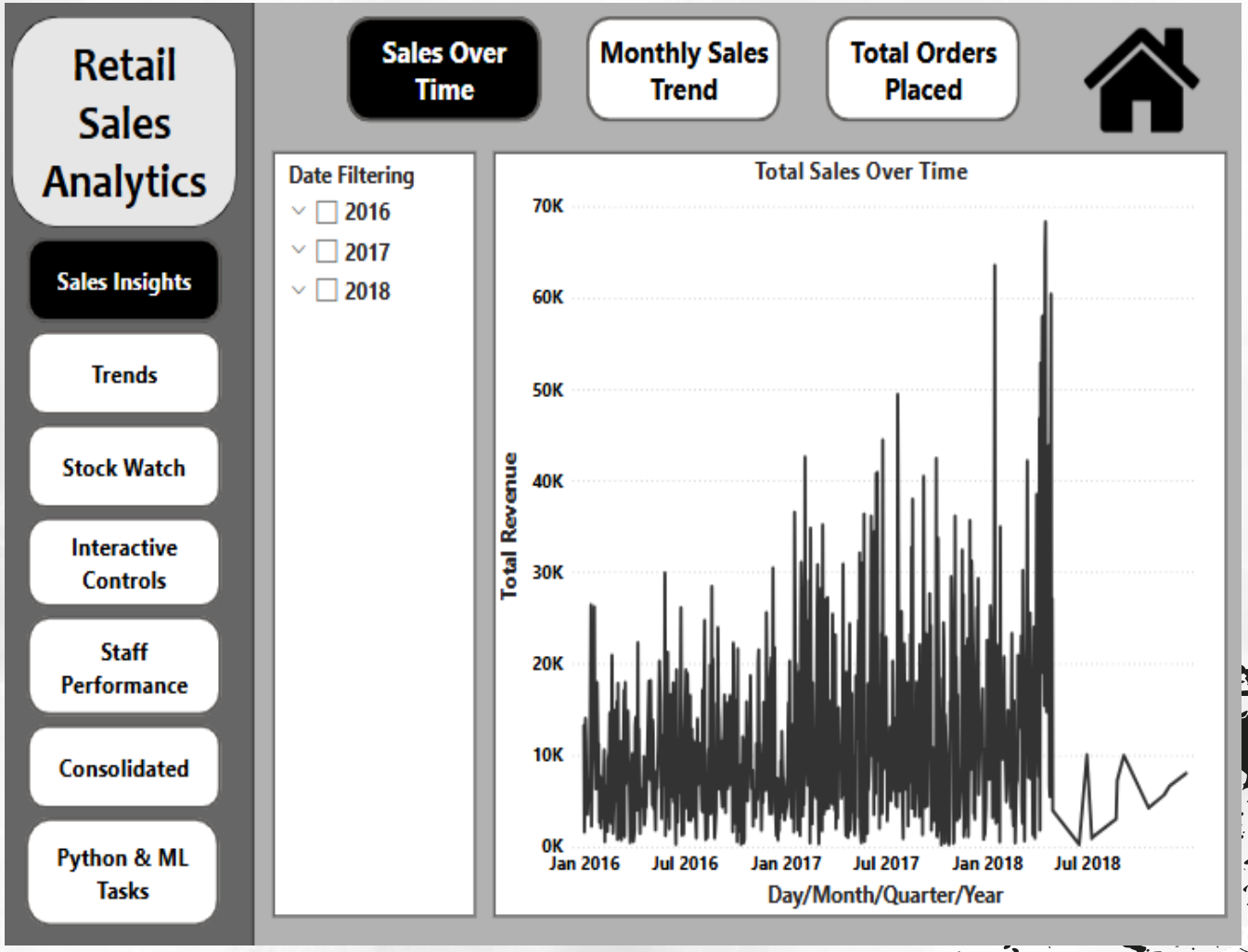


Power BI – Visualization & Dashboarding

3. Sales Overview Report

3(a). Sales Over Time

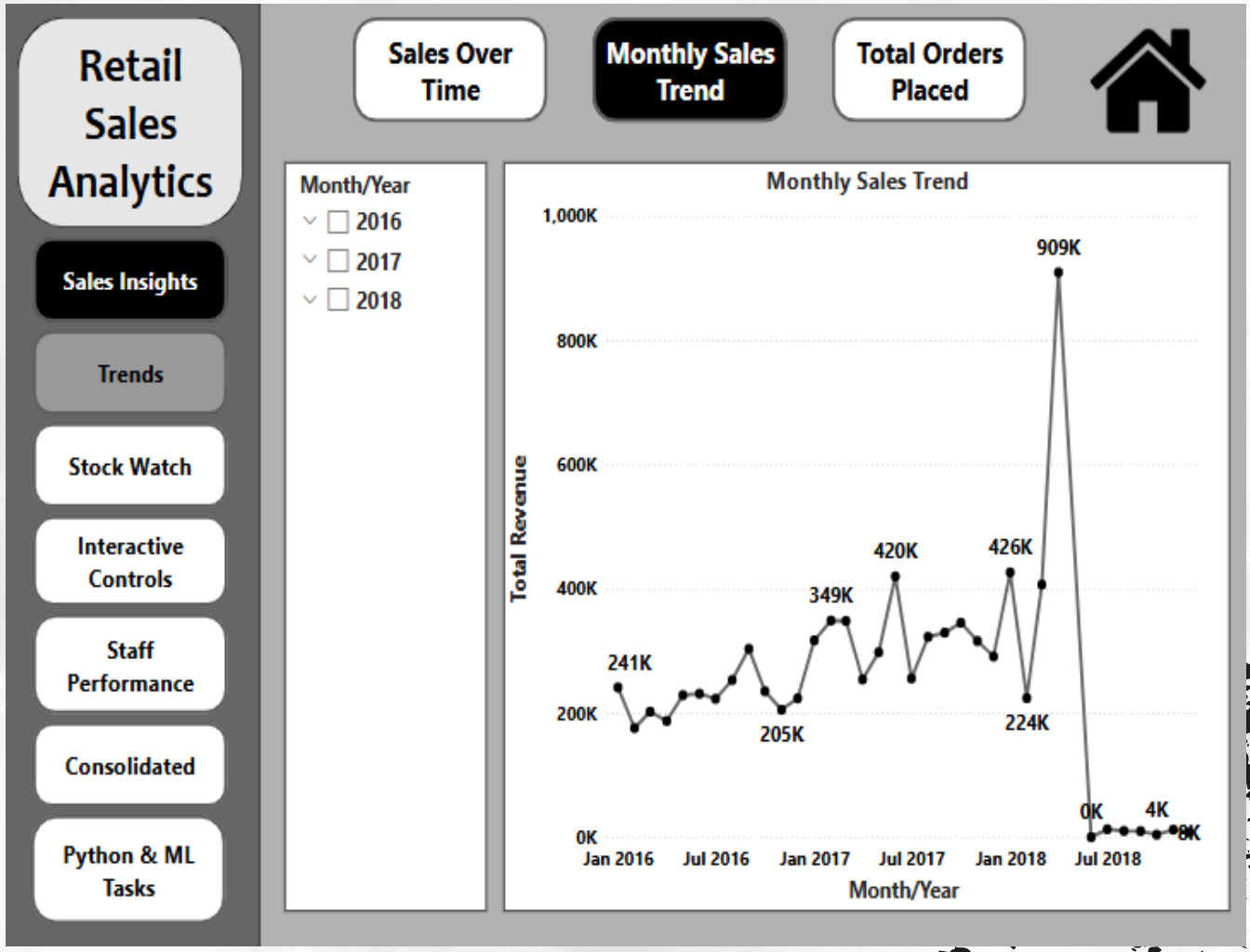
- The highest sales happened during the particular date was 2 consecutive days which is 15th & 16th April in 2018 which consists of 65K and 68.2K respectively being the top in terms of daily sales.
- The lowest sale happened during the particular date was on 10th October 2017 which is of 190/-.



Power BI – Visualization & Dashboarding

3(b). Monthly Sales Trend

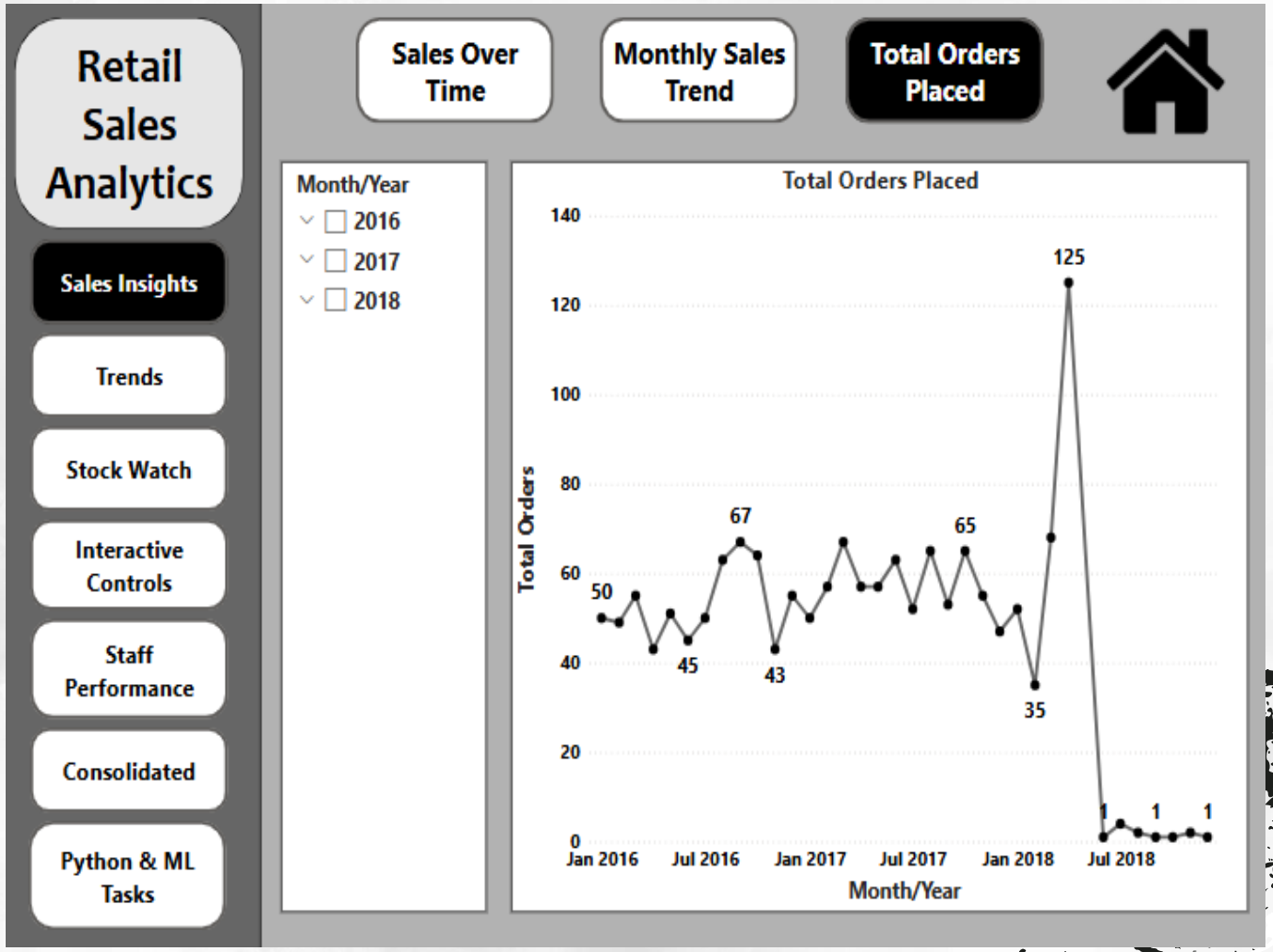
- The highest sales in terms of revenue happened during the month of April in the year 2018 which is of 0.90M being the top where as the lowest sale happened during the month of June in the year 2018 which is of 210/-.
- In year wise data, the highest sales happened during the year 2017 of 3.84M being the top.



Power BI – Visualization & Dashboarding

3(c). Total Orders Placed

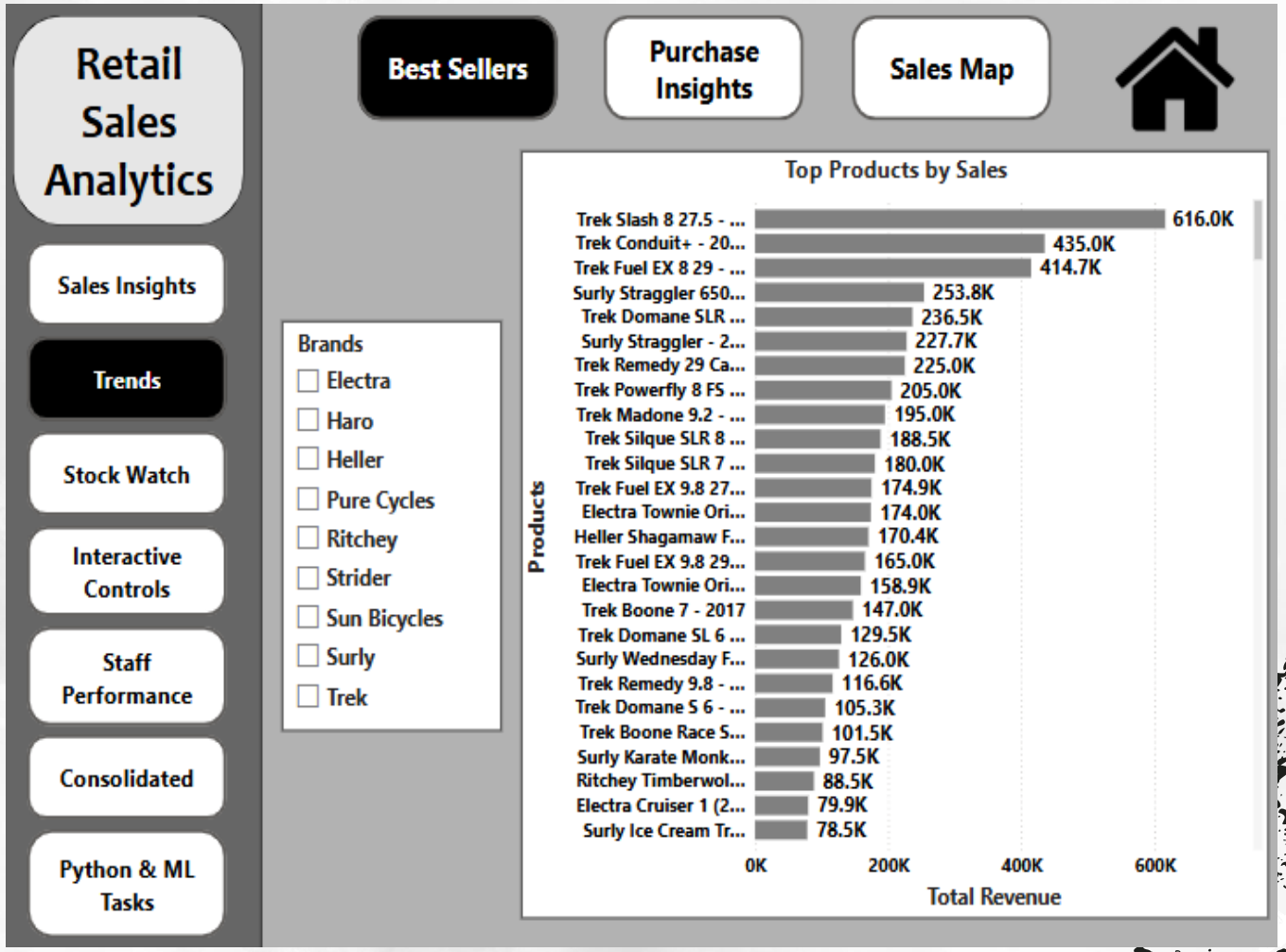
- The highest number of orders were placed during the month of April in the year 2018 which is of 125 orders. This is being the top and only month that crossed above 100 orders.
- The lowest number of orders also placed in the same year 2018 which consists of 1 during the months June, October, September, December.



Power BI – Visualization & Dashboarding

4. Top Product by Sales

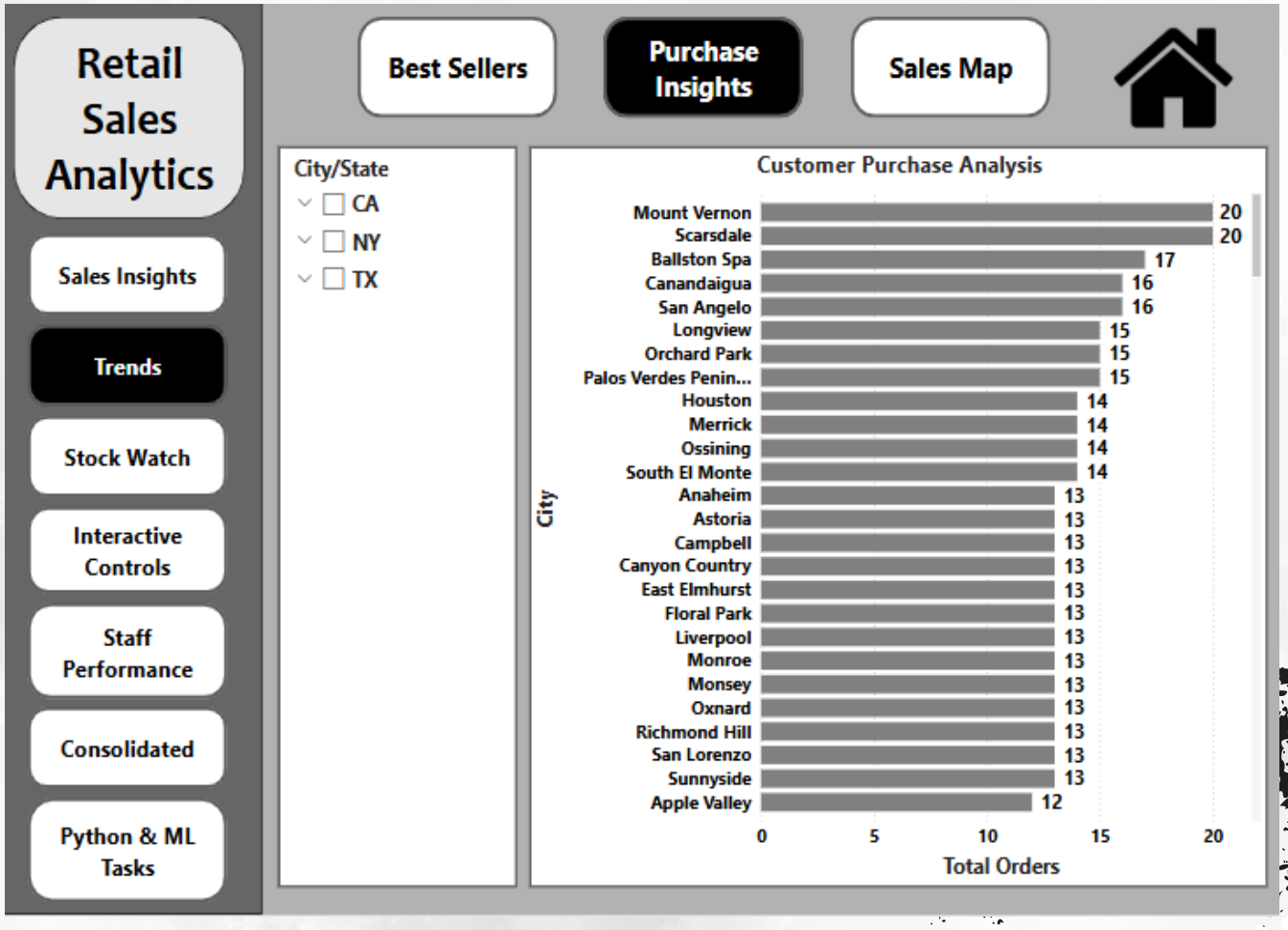
- The top selling product in terms of revenue is Trek Slash 8 27.5-2016 which is of 616K.
- The top 3 highest selling products in terms of revenue is occupied by the Brand Trek.
- Also, Trek brand products occupied 8 positions in top 10 where as remaining 2 positions were occupied by Surly brand.



Power BI – Visualization & Dashboarding

5. Customer Purchase Analysis

- The highest number of orders placed by the customers in the City named Mount Vernon and Scarsdale of 20 orders being the most.
- These both cities belong to the state NY.
- The least number of orders also placed from the same state NY with 3 different cities which is 1 order from each city. Middle Village, Tonawanda and Westbury are the cities.

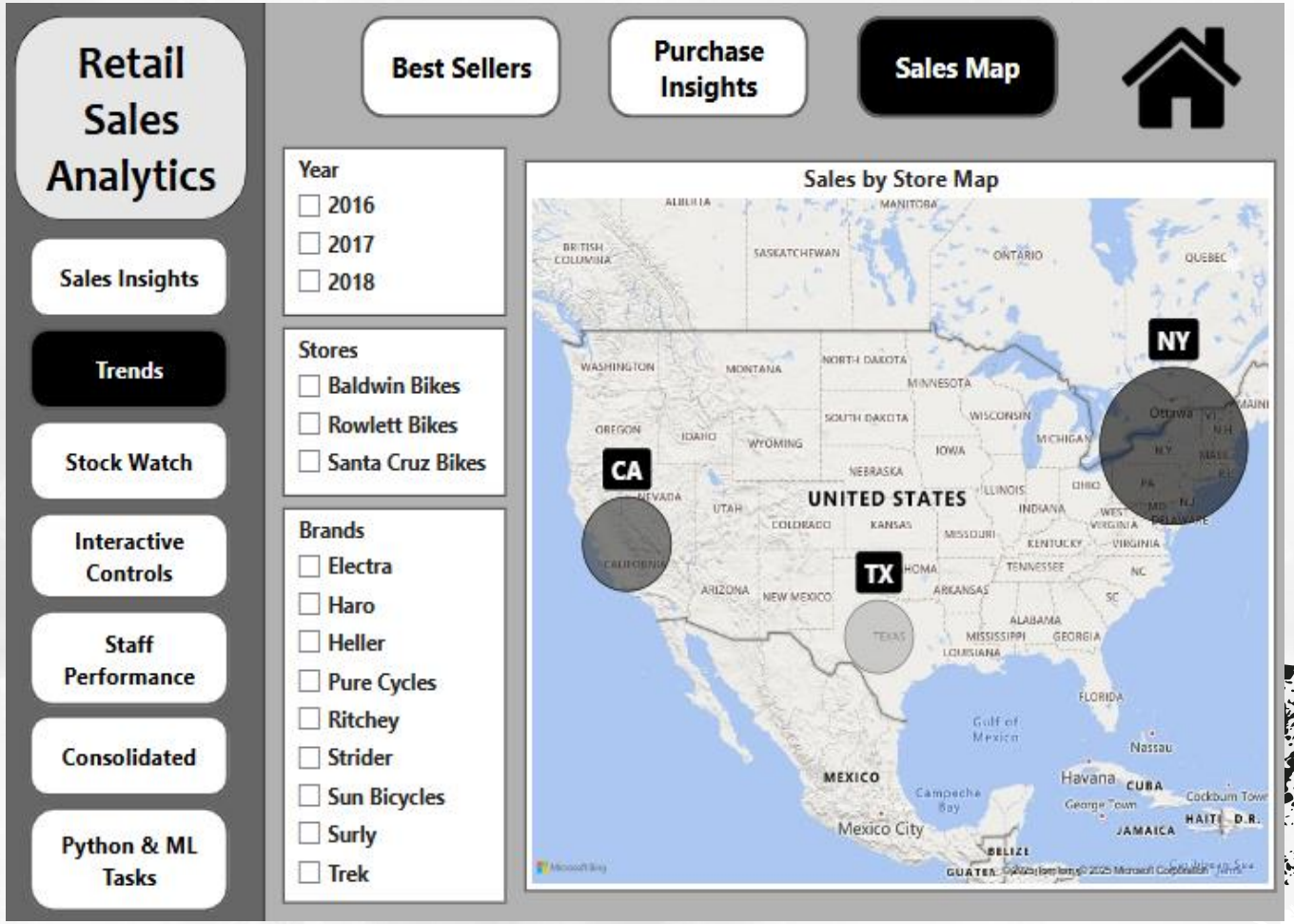


Power BI – Visualization & Dashboarding

6. Sales by Store Map

- The highest revenue generated store is Baldwin Bikes which is of 5.82M being the top from NY state followed by Santa Cruz Bikes with 1.79M from CA state.
- The lowest revenue generated was from Rowlett Bikes of 0.96M from TX state respectively.

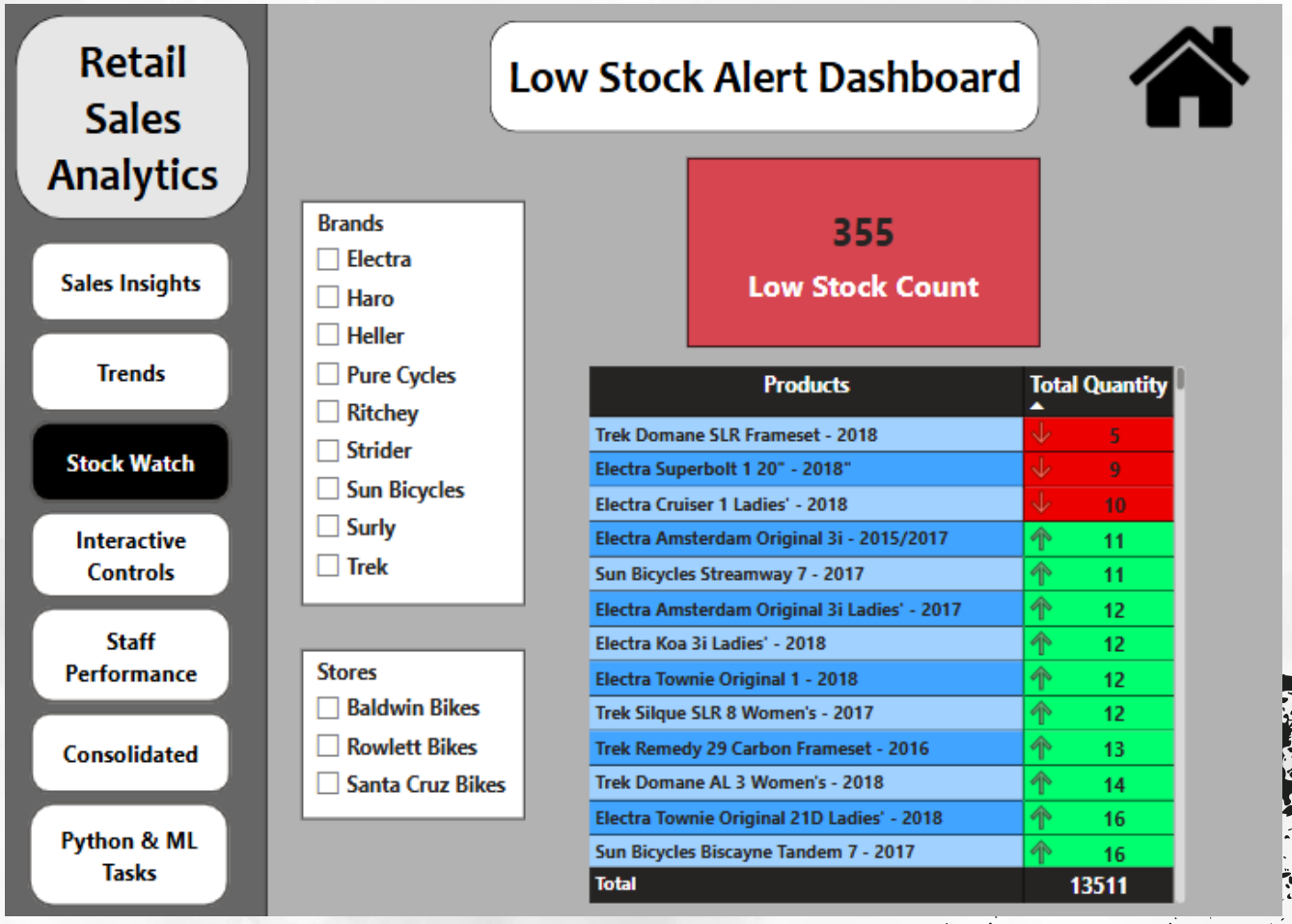
Stores	Total Revenue
Santa Cruz Bikes	1.79M
Baldwin Bikes	5.82M
Rowlett Bikes	0.96M



Power BI – Visualization & Dashboarding

7. Low Stock Alert Dashboard

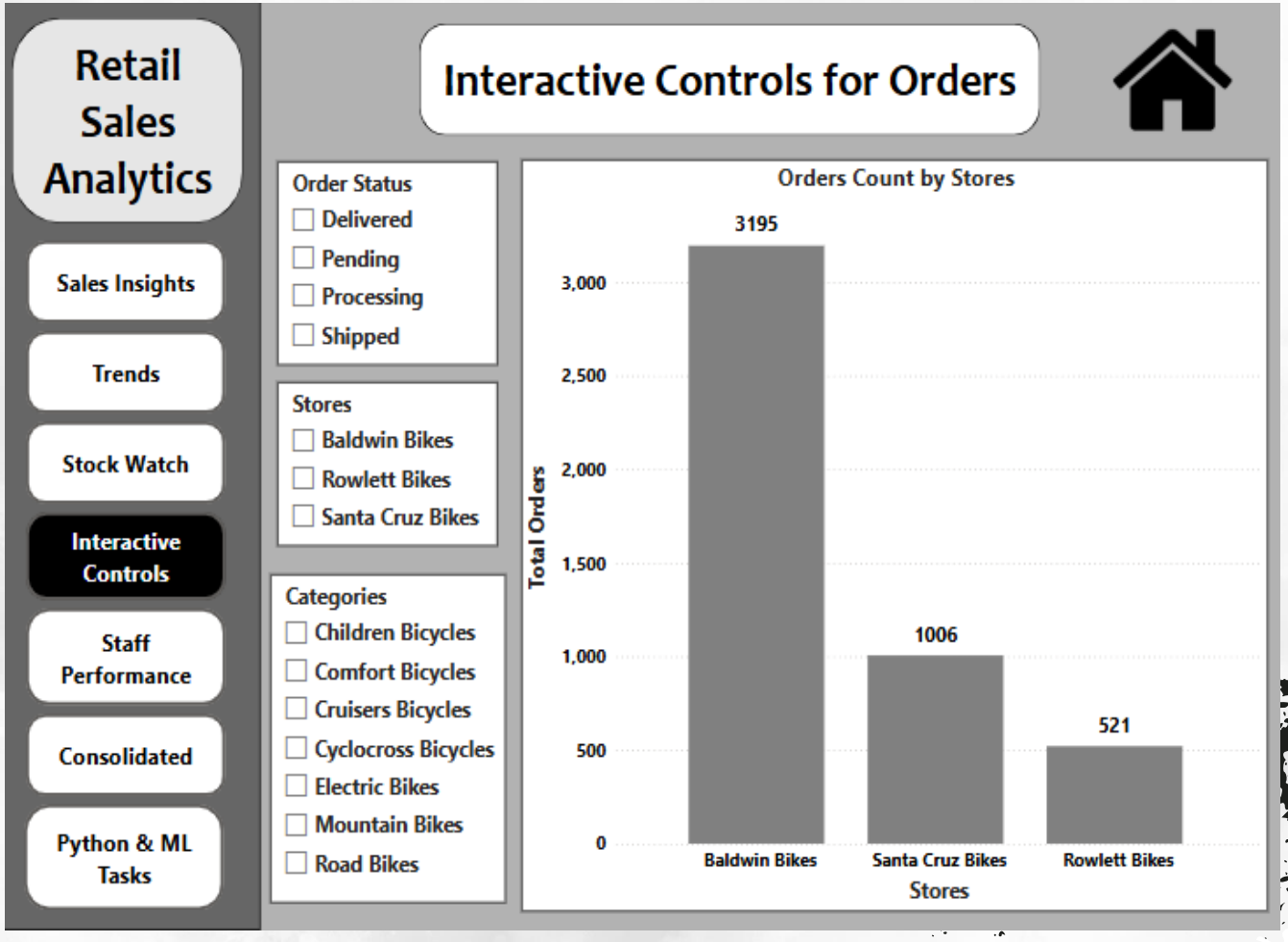
- There are 355 products overall has a stock of <= 10 units. This is the count of each products in each stores overall. The same products would have been repeated in this count in different stores.
- By merging the stocks of each products in all the 3 stores, it has been found that there are 3 products which has a stock of <= 10 units.



Power BI – Visualization & Dashboarding

8. Interactive Filters and Slicers

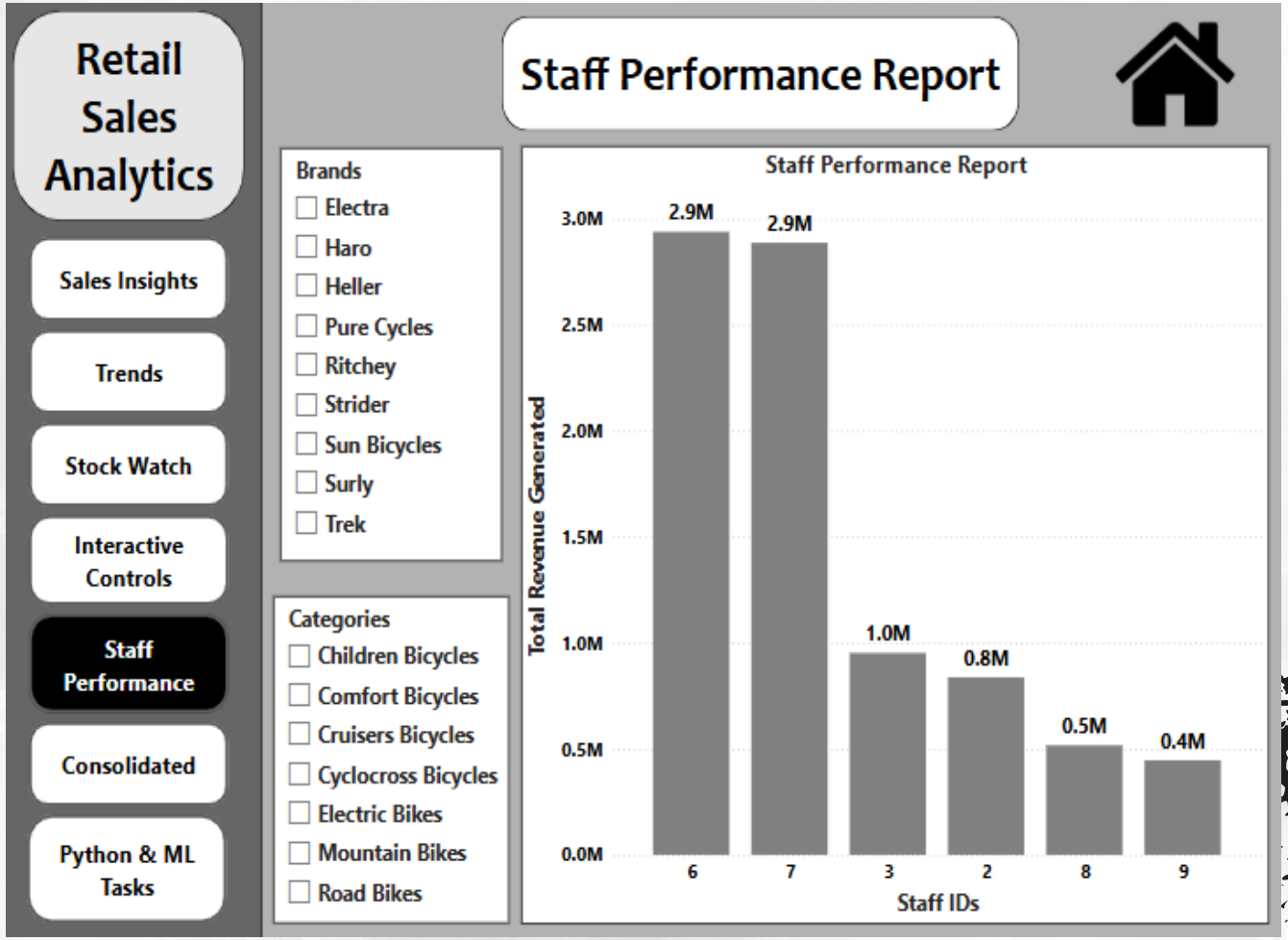
- This data shows the number of orders placed by each store where we could found that Baldwin Bikes has highest orders placed of 3.19K Followed by Santa Cruz Bikes with 1.0K and Rowlett Bikes with 521 orders.
- The interactive slicers has been created to filter through the order status, stores and categories in the Power BI.



Power BI – Visualization & Dashboarding

9. Staff Performance Report

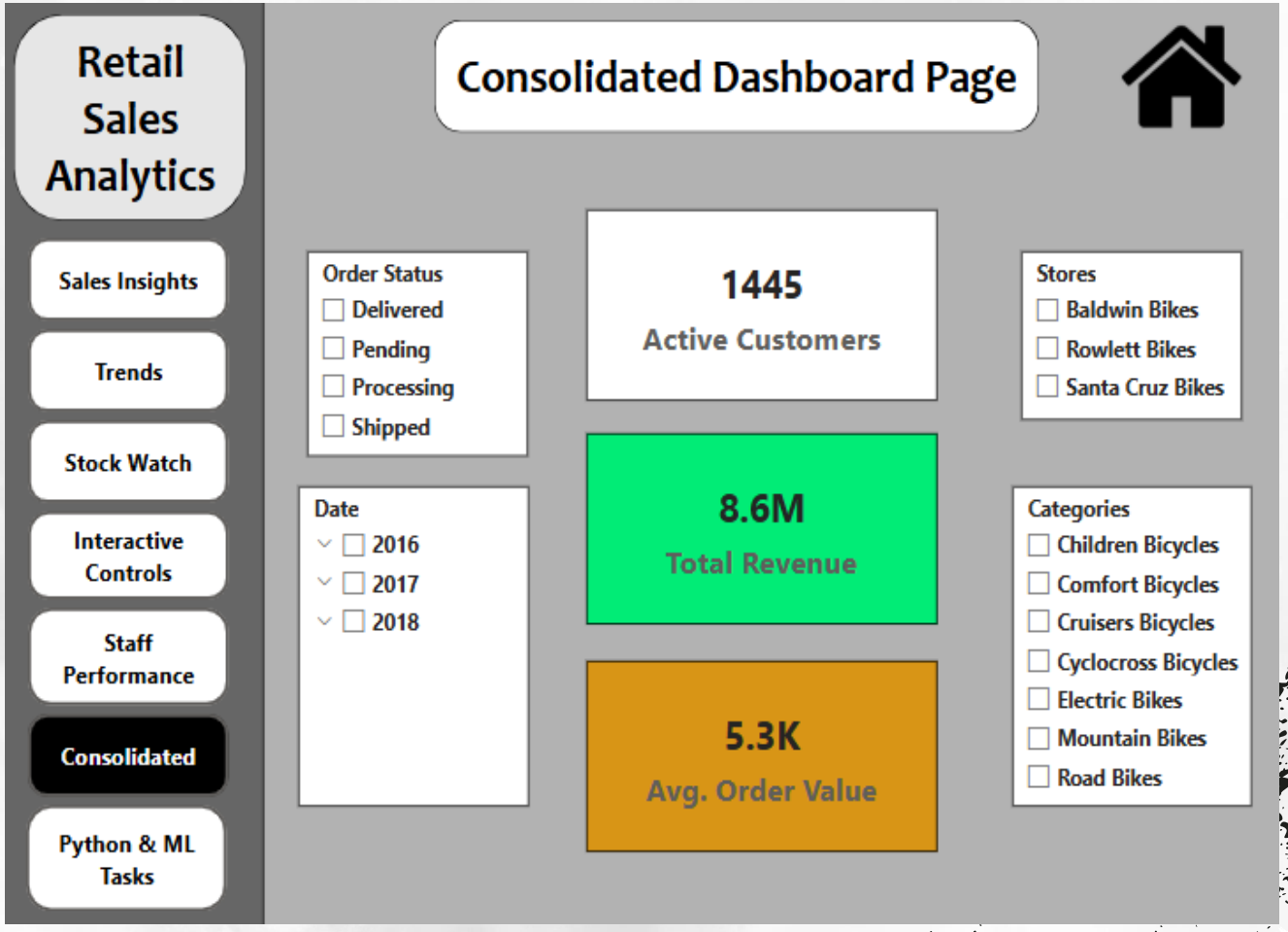
- The highest revenue generated by the Staffs is Staff Id 6 with 2.93M followed by Staff Id 7 with 2.88M.
- The least revenue generated by the Staff Id 9 with 0.4M.
- Staff IDs 1, 4, 5 and 10 have not handled any orders or generated any revenue.



Power BI – Visualization & Dashboarding

10. Consolidated Dashboard Page

- The final consolidated dashboard page with KPI fields have been created.
- There are 1445 active customers with the total revenue generated of 8.6M where their overall average order value is 5.3K.
- The slicers created with order status, date, stores and categories to filter the data and to apply the KPI field values based on the requirement.



Power BI – Visualization & Dashboarding

11. Import customer_segments Table

- Customer_segments table has been imported from SQL to Power BI.
- The relationships with other tables have been created using customer_id.

14. Use Segments as Report Filters

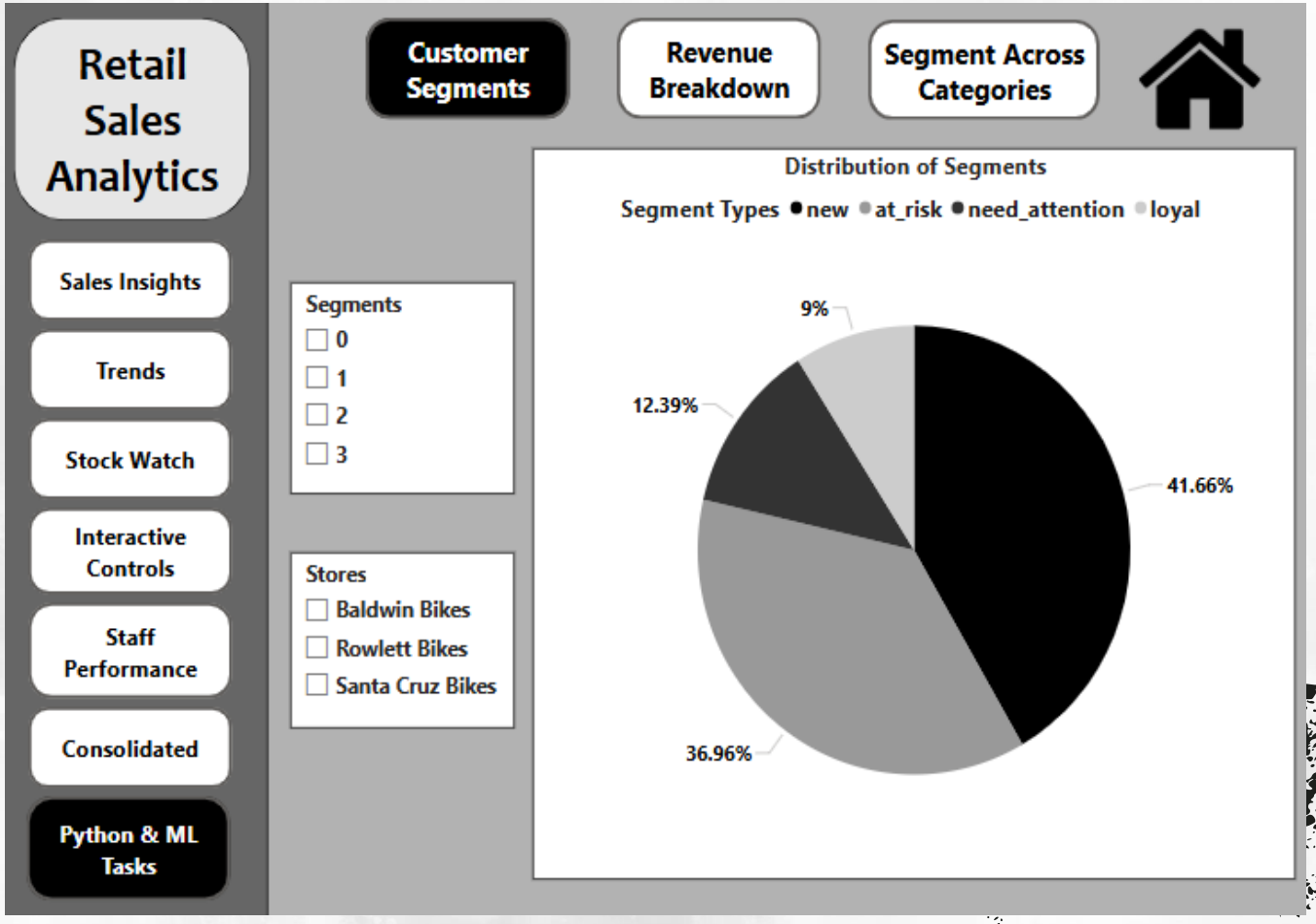
- An interactive slicers to filter across dashboards have been created for the customer_segments.

Power BI – Visualization & Dashboarding

12. Visualize Customer Segments

- The pie chart has been created to visualize the distribution of segments for customers.
- The segment 2 have more customers with 42% followed by segment 0 with 37%.

Segments	Type of Segments	Total Customers (%)
0	At_risk	36.96%
1	Loyal	9%
2	New	41.66%
3	Need_attention	12.39%

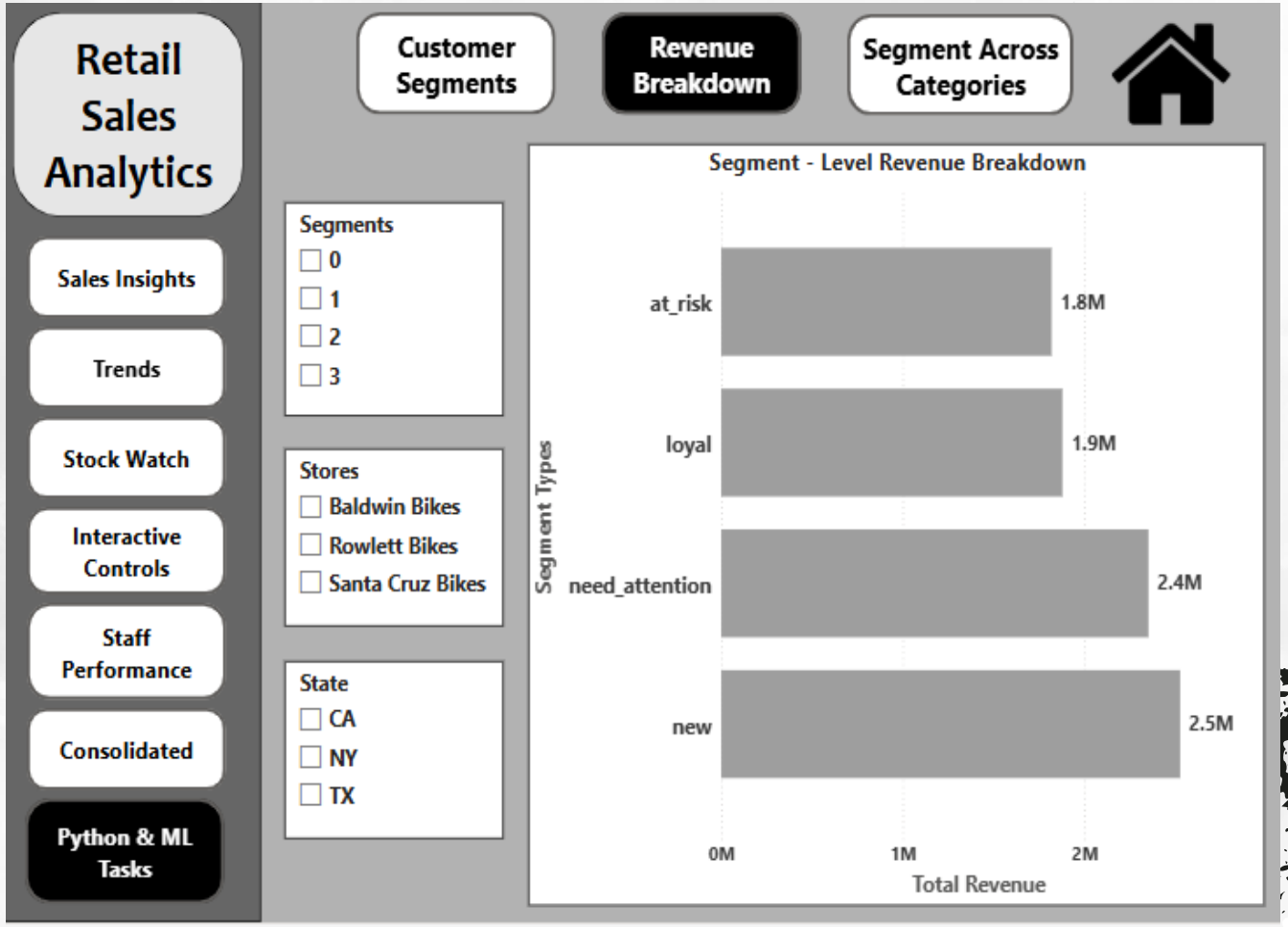


Power BI – Visualization & Dashboarding

13. Segment-Level Revenue Breakdown

- The bar chart has been created to visualize the segment-level revenue breakdown.
- The highest revenue generated by the customer segment 2 which is 2.5M.

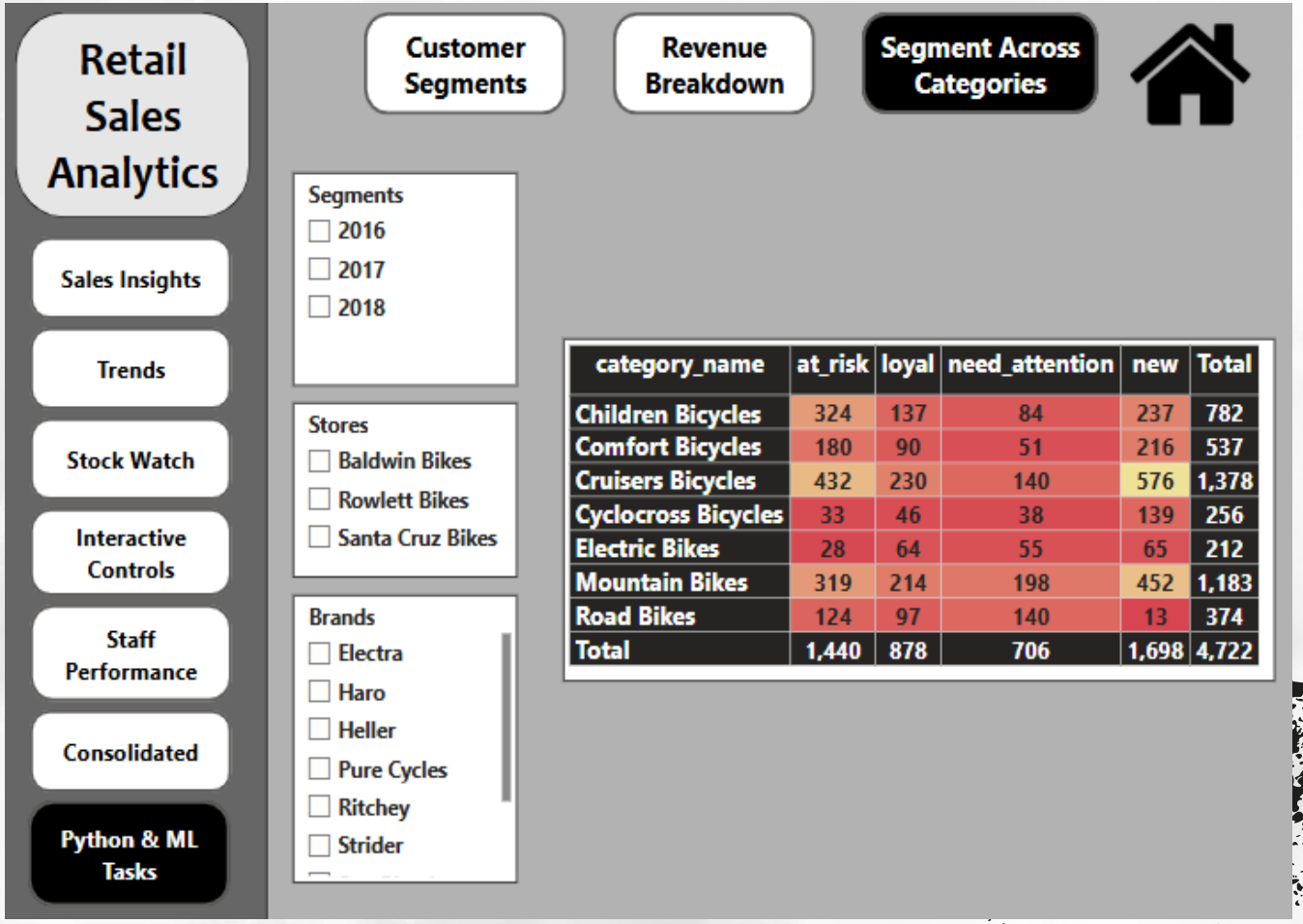
Segments	Type of Segments	Total Revenue
0	At_risk	1.8M
1	Loyal	1.9M
2	New	2.5M
3	Need_attention	2.4M



Power BI – Visualization & Dashboarding

15. Segment Heatmap

- The matrix visual has been created to compare the segments across categories.
- The customers fall under segment 2 has placed most orders in Cruisers Bicycles category of 576 followed by Mountain Bikes of 452 orders
- The segment 2 also has the least orders of 13 in Road Bikes category.



category_name	at_risk	loyal	need_attention	new	Total
Children Bicycles	324	137	84	237	782
Comfort Bicycles	180	90	51	216	537
Cruisers Bicycles	432	230	140	576	1,378
Cyclocross Bicycles	33	46	38	139	256
Electric Bikes	28	64	55	65	212
Mountain Bikes	319	214	198	452	1,183
Road Bikes	124	97	140	13	374
Total	1,440	878	706	1,698	4,722

THANK YOU

Thambidurai Sundaramoorthy

stdurai95@gmail.com