

**blinkit**

 Power BI

**Title:**

Blinkit Sales Data Analysis

**Name:**

Debjit Sutar

**College:**

Narula Institute of Technology,  
Agarpara **(6<sup>th</sup> Semester)**

Internship Under:

**Classr**  m

**Phone:**

8981838547

**Website:**

<https://classroomtech.in>

**LinkedIn:**

<https://www.linkedin.com/company/classroom-tech/>

**Duration:**

3 months 1st April 2025 to 30th June 2025



Power BI

## Acknowledgement:

I would like to express my heartfelt gratitude to **Mr. Satyaki Das** of **Classroom Tech** for his invaluable guidance, mentorship, and continuous support throughout this internship.

I also extend my sincere thanks to my **institution, faculty members, family**, and all those who encouraged and supported me during this journey. Their motivation and guidance played a vital role in the successful completion of this project.

Section Title	Slide No.
Introduction	1
Acknowledgement	2
Table of Contents	3
Objective / Problem Statement	4
Dataset Overview	5
Tools Used	6
Methodology	7
Question-Wise Visualizations (Q1–Q40)	8–47
Key Findings	48
Conclusion	49
GitHub Repository Link	50
References	51
Thank You	52



## Objective:

To explore and analyze Blinkit's operational data using Power BI in order to generate meaningful business insights and visual dashboards.

## Key Questions Answered:

- Which products and categories are top-selling?
- What are the peak times for order placement?
- How do campaigns impact conversions and revenue?
- How efficient is the delivery and logistics chain?
- Which regions or customer segments are most active?



## Dataset Overview

**Dataset Name:** Blinkit Sales

**Dataset Source:** Kaggle

**Total Files:** 11 CSV/Excel Files

Dataset Name	Rows × Columns	Key Columns for Analysis
blinkit_orders.csv	5000 × 10	order_id, customer_id, order_date, store_id
blinkit_order_items.csv	5000 × 4	order_id, product_id, quantity, price
blinkit_customers.csv	2500 × 11	customer_id, name, email, segment, pincode
blinkit_customer_feedback.csv	5000 × 8	feedback_id, customer_id, category, sentiment
blinkit_products.csv	268 × 10	product_id, product_name, category_id, price
blinkit_inventory.csv	75172 × 4	product_id, stock_level, stock_received_date
blinkit_inventoryNew.csv	18105 × 4	product_id, stock_received, damaged_stock
blinkit_delivery_performance.csv	5000 × 8	order_id, promised_time, actual_time, distance
blinkit_marketing_performance.csv	5400 × 11	campaign_id, campaign_name, spend, revenue_generated
Category_Icons.xlsx	11 × 2	category_id, icon_url
Rating_Icon.xlsx	5 × 3	rating, emoji



Power BI

## Tools Used:

- **Power BI:**

For importing, cleaning, modeling, and visualizing datasets.

- **GitHub:**

Used for maintaining version control, storing .pbix files, and project documentation.



## Methodology:

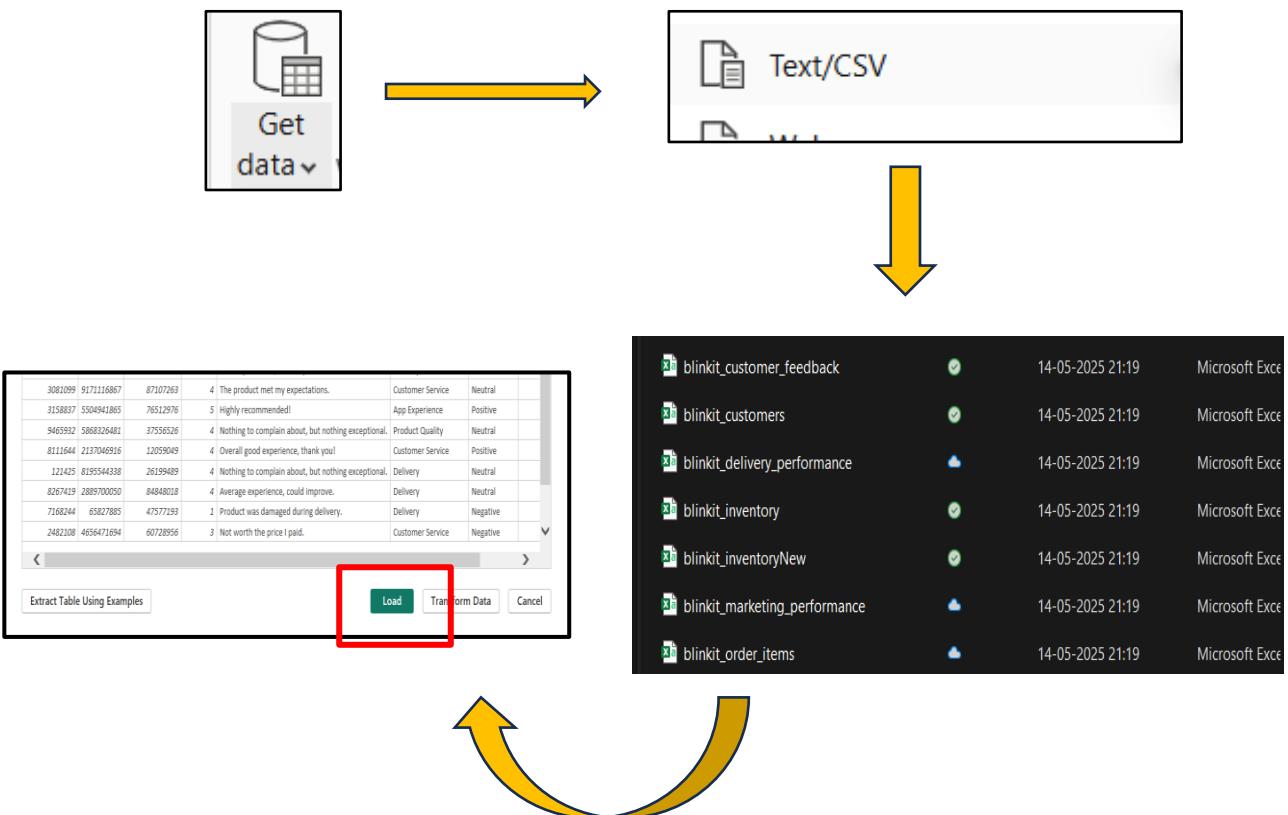
### ➤ Data Cleaning (Power Query):

Removed duplicates from blinkit\_customers  
Handled null values in delivery and feedback tables  
Merged stock\_1 and stock\_2 into inventory\_new  
Standardized column names and data types.

### ➤ Visualization Approach:

Created bar, line, pie, funnel, scatter, KPI, matrix charts  
Used slicers, filters, and bookmarks for interactivity  
Added DAX measures for metrics like ROAS, CLV, Gross Profit

## 1. How can you import data from all 11 tables into Power BI?



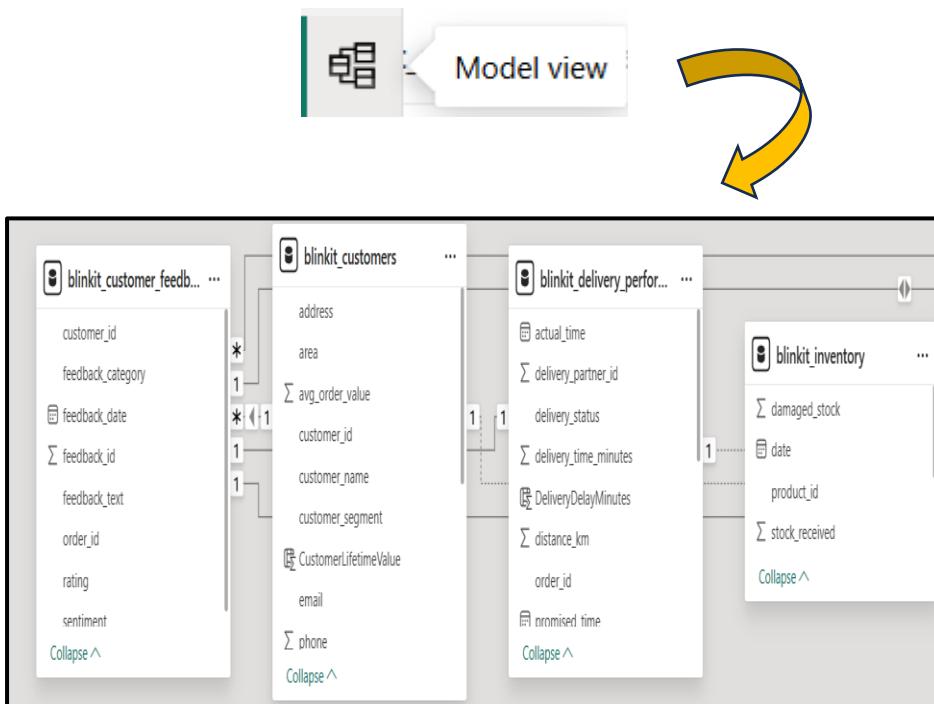
### 1 Use Get Data:

Go to **Home > Get Data**, choose your source (Excel, CSV, or SQL), and select all 11 tables from the Navigator window.

### 2 Load or Transform:

Click **Load** to import directly, or choose **Transform Data** to clean and prepare the tables before loading into Power BI.

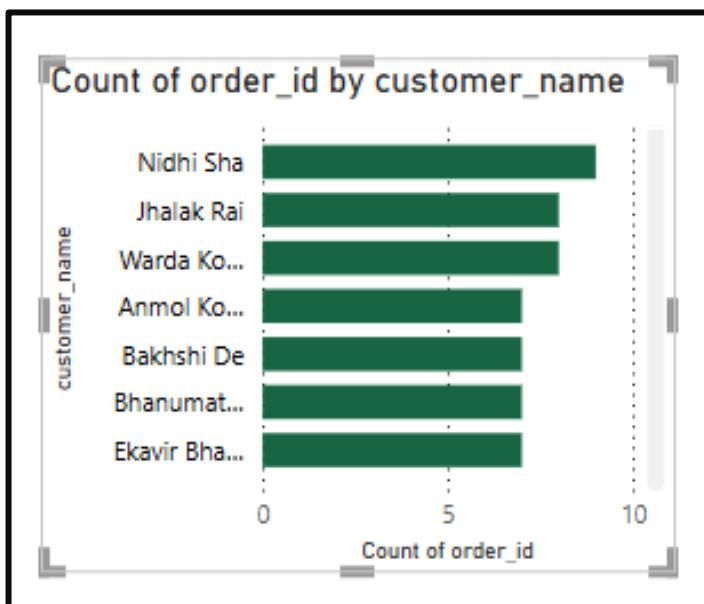
## 2. How do you create relationships between the tables in Power BI?



**1 Open Model View:**  
Go to Model View, where all tables are displayed.

**2 Drag and Connect Fields:**  
Drag a common field (e.g., customer\_id, product\_id) from one table to its match in another to create a relationship.

3. Create a bar chart showing the number of orders placed per customer.



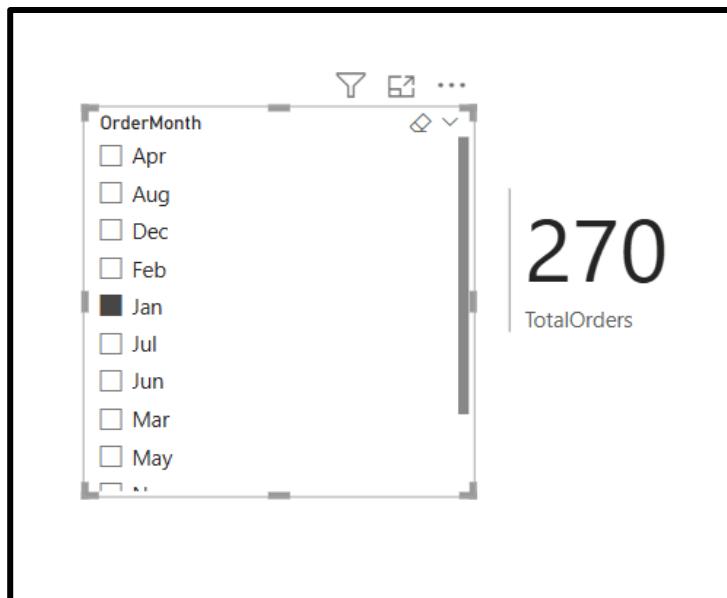
**1 Track Monthly Order Volumes:**

Calculating orders by month reveals overall demand patterns, helping identify trends and seasonal fluctuations.

**2 Spot Sales Peaks and Dips:**

Monthly analysis highlights peak-performing periods, enabling better planning for inventory, staffing, and marketing strategies.

4. How do you calculate the total number of orders placed in a given month?



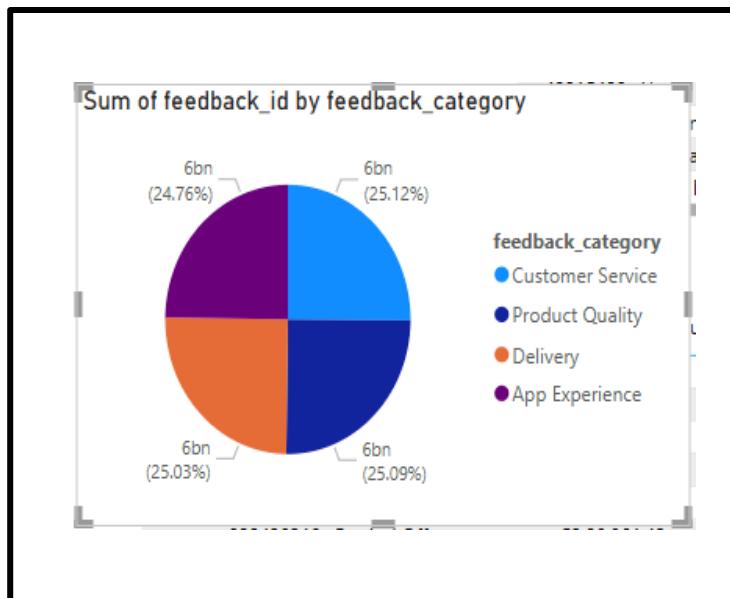
**1 Monitor Monthly Sales Performance:**

Calculating total orders per month helps track business growth and detect seasonal trends in customer demand.

**2 Enable Time-Based Filtering:**

Using date filters or DAX formulas allows dynamic calculation of monthly orders, supporting flexible reporting and analysis.

5. Create a pie chart showing the distribution of feedback categories from Table 1.



## 1 Visualize Feedback Distribution:

A pie chart breaks down feedback into categories (e.g., Positive, Neutral, Negative), offering a clear snapshot of overall customer sentiment.

## 2 Identify Areas for Improvement:

The distribution helps pinpoint dominant feedback types, guiding focus toward improving customer experience where needed.

6. Create a table visualization showing customer details (customer\_id, name, email) from Table 2.

customer_id	customer_name	email
80315623	Varenya Banik	aachalsarma@example.com
70230254	Hamsini Sandhu	aachalsheth@example.net
14161586	Aachal Nazareth	aadhyat1@example.org
61226563	Chameli Kala	aadhyat91@example.com
74735607	Naksh Amble	aadhyabhatta@example.com
71615827	Yash Contractor	aadi81@example.com
27955158	Abeer Kamdar	aaggarwal@example.net
49915498	Harini Mangal	aahana49@example.com
43459807	Charan Brahmbhatt	aahanatailor@example.net
51661209	Vasana Chandran	aaravdave@example.com
67749649	Oviya Mitra	aaravdugar@example.com

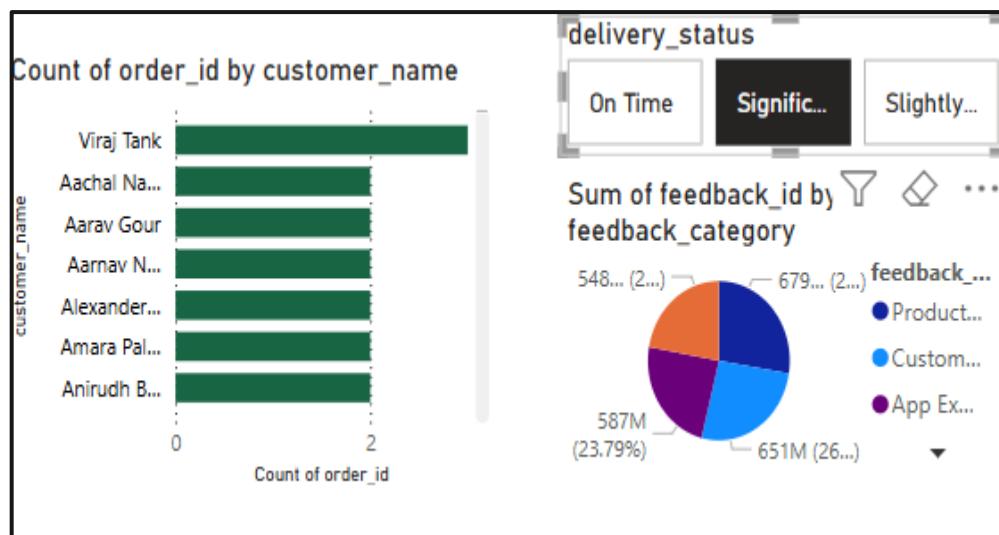
**1 Display Key Customer Information:**

A table visualization neatly presents customer IDs, names, and emails, making it easy to reference and manage customer data.

**2 Enable Quick Lookups and Filtering:**

The structured view allows users to search, sort, or filter specific customers efficiently within Power BI dashboards.

7. How can you filter orders based on delivery status (on-time, delayed) from Table 3?



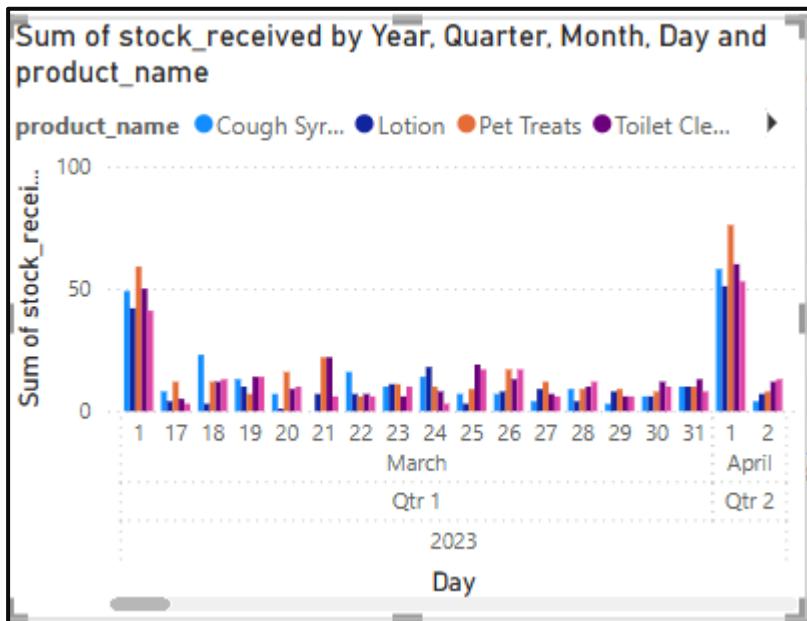
**1 Use Delivery Status as a Filter:**

Apply the `delivery_status` field from Table 3 as a slicer or visual-level filter to view only on-time or delayed orders.

**2 Enable Focused Analysis:**

Filtering by delivery status helps monitor fulfillment performance and identify patterns in delivery delays.

8. Create a column chart showing stock received over time for different products (Table 4 & Table 5).



## 1 Track Stock Inflow

### Trends:

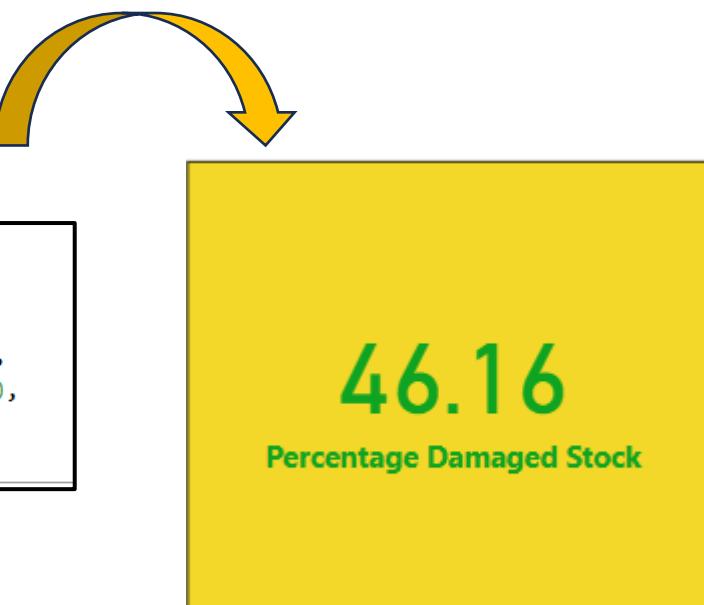
A column chart visualizes how stock quantities change over time, making it easy to monitor restocking patterns.

## 2 Compare Product Stocking Rates:

By combining data from both tables, the chart highlights which products receive stock more frequently or in higher volumes.

9. How do you calculate the percentage of damaged stock per product?

```
1 Percentage Damaged Stock =  
2  
3 DIVIDE(  
4     SUM(blinkit_inv_comb[damaged_stock]),  
5     SUM(blinkit_inv_Comb[stock_received]),  
6     0  
7 ) * 100
```



**1 Apply a DAX Measure for Percentage:**

Use a formula like Damaged % = (Damaged Stock / Total Stock Received) × 100 to calculate damage rate per product.

**2 Identify High-Risk Products:**

This percentage helps flag products with high damage rates, enabling better quality control and supply chain decisions.

10. Show a table with all campaigns (campaign\_id, campaign\_name, spend, revenue\_generated) from Table 6.

Sum of campaign_id	campaign_name	Sum of spend	Sum of revenue_generated
846068832	Weekend Special	53,98,833.87	1,07,14,355.49
934280295	Referral Program	54,54,076.53	1,10,74,147.80
906505539	New User Discount	55,00,364.43	1,08,11,580.51
932946840	Membership Drive	53,70,209.40	1,05,74,853.75
874138266	Flash Sale	54,95,063.46	1,06,68,261.06
932438310	Festival Offer	53,90,061.42	1,05,21,191.73
897157743	Email Campaign	54,32,189.01	1,08,05,355.66
902829894	Category Promotion	55,51,749.00	1,07,47,366.11
894940830	App Push Notification	53,66,967.60	1,06,63,110.00
8121306549		4,89,59,514.72	9,65,80,222.11

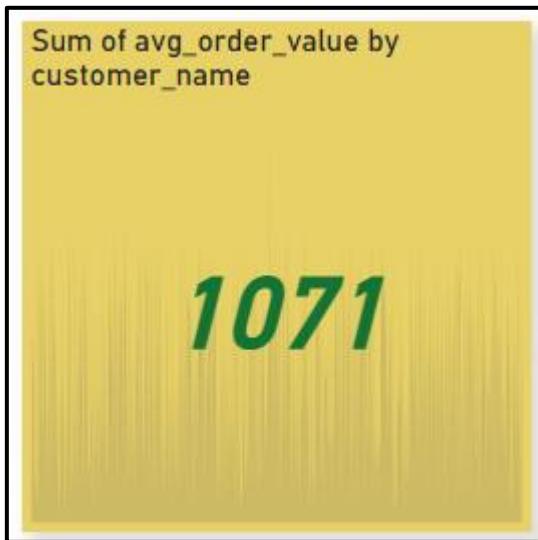
**1 Display Key Campaign Metrics:**

A table view presents each campaign's ID, name, spend, and revenue, making it easy to evaluate marketing performance.

**2 Enable ROI Comparison:**

This structured layout allows quick analysis of campaign effectiveness by comparing spend vs. revenue across campaigns.

11. Create a KPI visual to display the average order value (avg\_order\_value) from Table 2.



**1 Highlight Average Customer Spending:**

A KPI visual shows the average order value at a glance, helping track how much customers typically spend per order.

**2 Monitor Business Health:**

This metric is key for assessing profitability and can signal changes in customer buying behavior over time.

12. How do you calculate the total revenue generated from all campaigns in Table 6?

```
1 TotalCampaignRevenue = SUM(blinkit_marketing_performance[revenue_generated])
```

```
2
```



96.58M

TotalCampaignRevenue

## 1 Use a Simple Aggregation Formula:

Calculate total revenue by summing the revenue\_generated column in Table 6 using `SUM('Table6'[revenue_generated]).`

## 2 Track Overall Campaign Impact:

This total helps assess the combined effectiveness of all campaigns in driving revenue.

13. Create a stacked bar chart to compare order quantity per product (Table 7).



**1 Compare Product Demand Visually:**

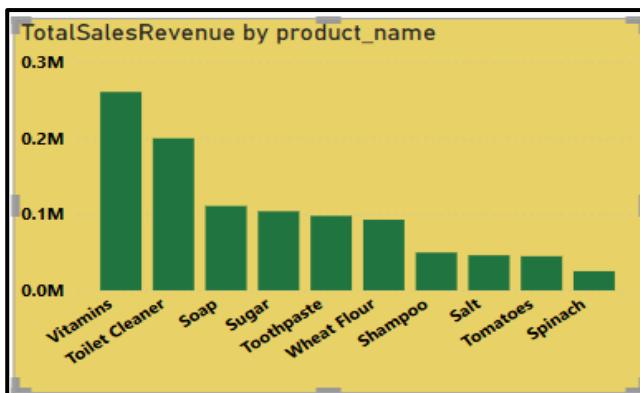
A stacked bar chart shows how many units of each product were ordered, helping identify top-selling items.

**2 Spot Product-Level Sales Patterns:**

The visualization highlights which products contribute most to overall order volumes, supporting inventory and marketing decisions.

## 14. How do you calculate the total sales revenue per product?

```
1 TotalSalesRevenue =  
2 SUMX(  
3     blinkit_order_items,  
4     blinkit_order_items[quantity] * RELATED(blinkit_products[price])  
5 )  
6
```



### 1 Calculate Revenue by Linking Tables:

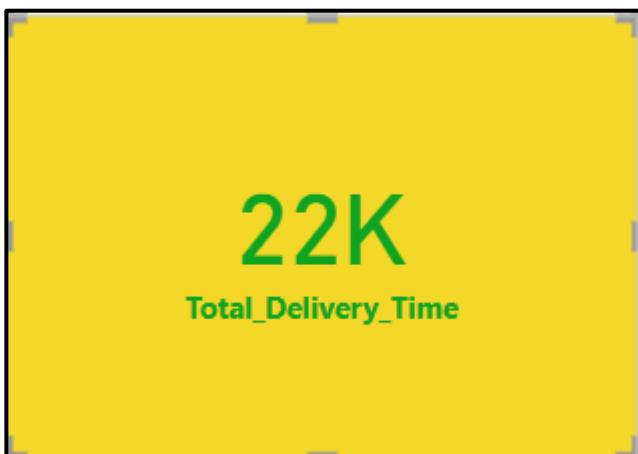
The SUMX function multiplies quantity from `blinkit_order_items` with price from `blinkit_products` using the `RELATED()` function.

### 2 Track Product-Wise Sales Performance:

This measure reveals how much revenue each product generates, helping identify best-sellers and optimize pricing strategies.

15. Create a measure to calculate the total delivery time (actual\_time - promised\_time) in Table 3.

```
1 Total_Delivery_Time = SUM(blinkit_delivery_performance[delivery_time_minutes])
```



## 1 Aggregate Delivery Time Efficiently:

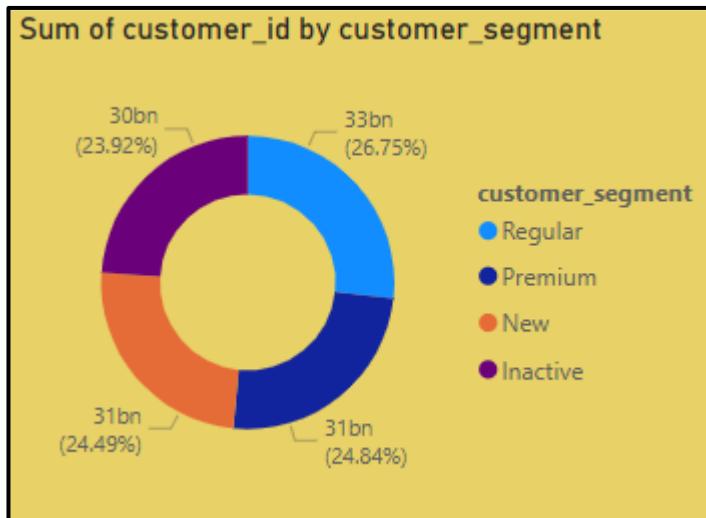
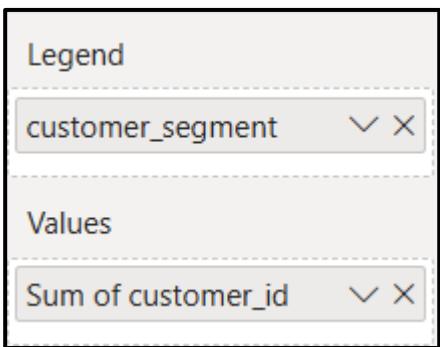
The measure

Total\_Delivery\_Time =  
SUM(blinkit\_delivery\_perfor  
mance[delivery\_time\_minut  
es]) provides the total time  
taken across all deliveries in  
minutes.

## 2 Evaluate Logistics Performance:

This total helps assess  
overall delivery efficiency  
and spot potential delays in  
the fulfillment process.

## 16. How do you visualize customer segments (Table 2) using a pie chart?



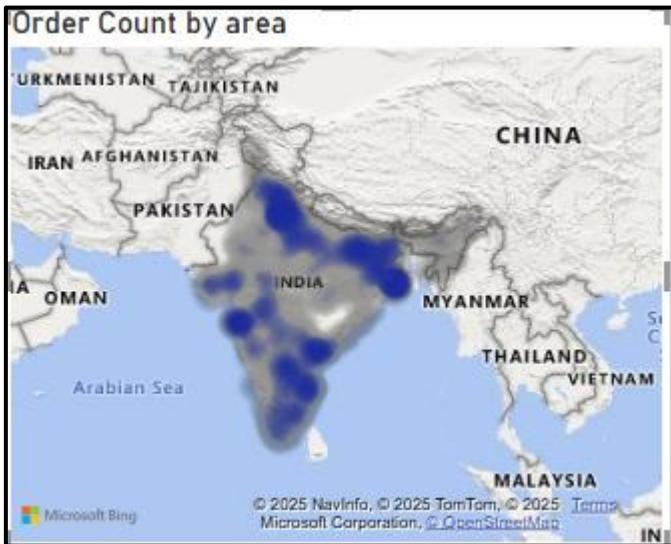
### 1 Use Segment Field as Legend:

Drag the customer\_segment column from Table 2 into the Legend and Values sections of a pie chart to display the distribution.

### 2 Highlight Segment Proportions:

The pie chart visually shows how customers are split across different segments (e.g., Regular, Premium), helping target marketing strategies.

17. Create a heatmap showing the frequency of orders per pincode (Table 2).



**1 Map Order Frequency by Area:**

The heatmap uses the area field to show how frequently orders are placed in each location, making demand patterns easy to spot.

**2 Spot High-Performance Zones:**

By visualizing order volume by area, you can quickly identify top-performing zones and areas with potential for growth.

18. How do you create a calculated column for delivery delays (actual\_delivery\_time - promised\_delivery\_time)?

```
1 DeliveryDelayMinutes =  
2 DATEDIFF(  
3     blinkit_delivery_performance[promised_time],  
4     blinkit_delivery_performance[actual_time],  
5     MINUTE  
6 )
```

**1 Define Delivery Delay per Order:**

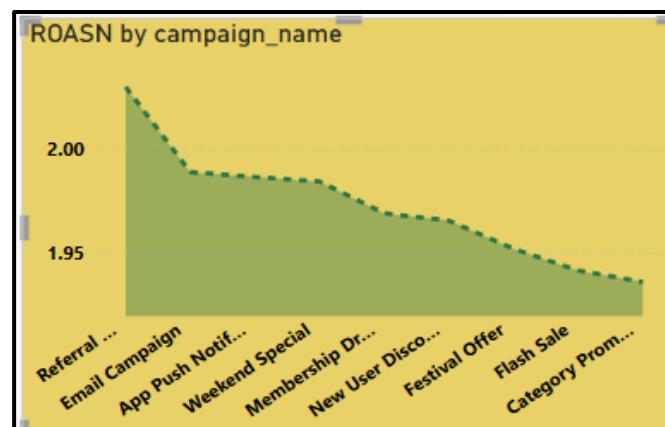
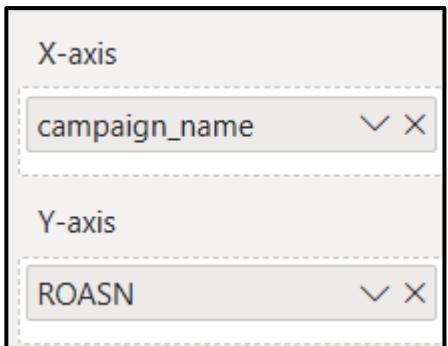
A calculated column helps compute how late each order was delivered by subtracting promised time from actual delivery time.

**2 Enable Delay Pattern Analysis:**

This allows you to track delays on a per-order basis and identify recurring issues across specific areas, dates, or delivery agents.

19. Create a measure to calculate the Return on Ad Spend (ROAS) using (revenue\_generated / spend) from Table 6.

```
1 ROASN =  
2 DIVIDE(  
3     SUM(blinkit_marketing_performance[revenue_generated]),  
4     SUM(blinkit_marketing_performance[spend]),  
5     0  
6 )  
7
```



## 1 Use DAX to Calculate ROASN:

Create a DAX measure that divides total revenue by total spend to compute ROASN across all campaigns.

## 2 Evaluate Marketing Efficiency:

ROAS helps assess how effectively each marketing campaign turns ad spend into revenue, supporting budget optimization.

20. Show a scatter plot of distance vs. delivery time to analyze delivery efficiency (Table 3).



**1** **Visualize Delivery Patterns:**

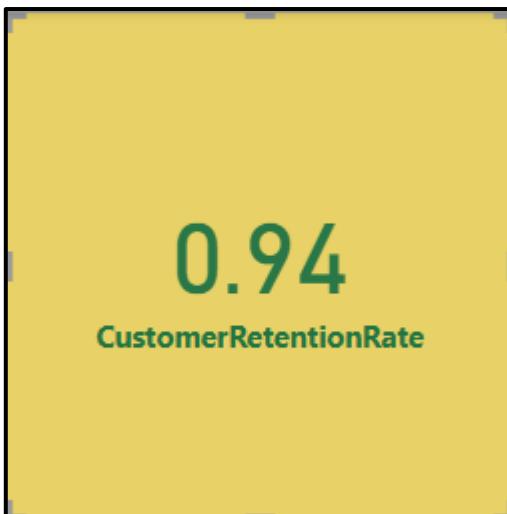
A scatter plot with `distance_km` on the X-axis and `delivery_time_minutes` on the Y-axis helps identify how distance impacts delivery time.

**2** **Spot Outliers and Delays:**

This chart makes it easy to detect unusual delays for short distances or high efficiency over longer routes, revealing optimization opportunities.

21. Create a measure to calculate customer retention rate using total\_orders from Table 2.

```
1 CustomerRetentionRate =  
2 DIVIDE(  
3     COUNTROWS(  
4         FILTER(  
5             blinkit_customers,  
6             blinkit_customers[total_orders] > 1  
7         )  
8     ),  
9     COUNTROWS(blinkit_customers),  
10    0  
11 )
```



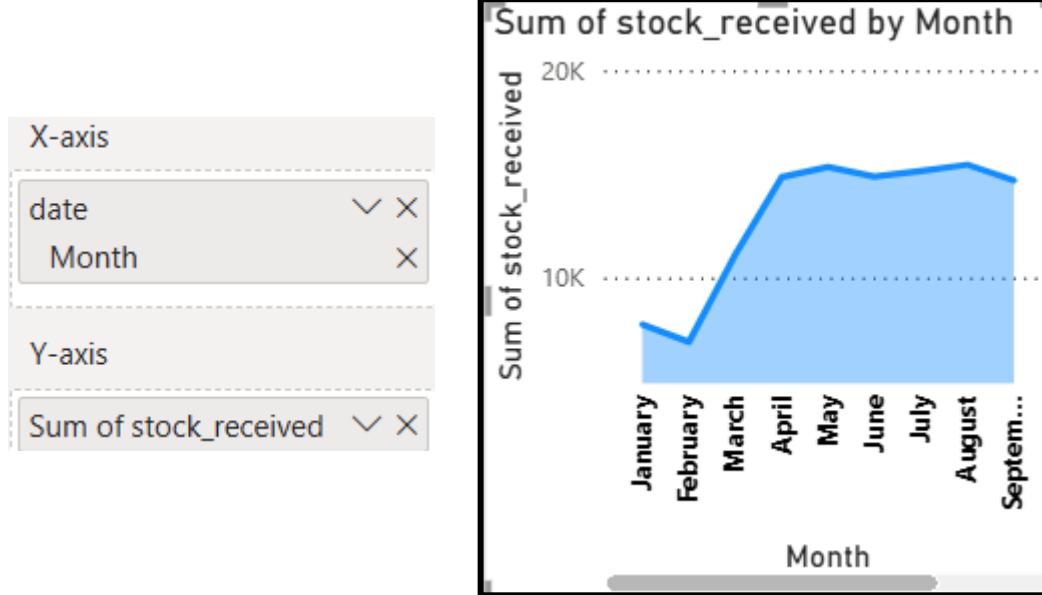
## 1 Define Retention Logic with DAX:

Use a measure to calculate the percentage of customers with more than one order, indicating repeat engagement.

## 2 Track Customer Loyalty Trends:

Retention rate reveals how effectively customers are returning, helping assess satisfaction and long-term value.

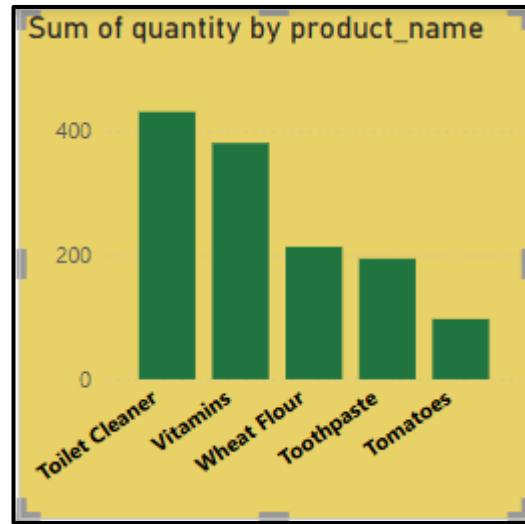
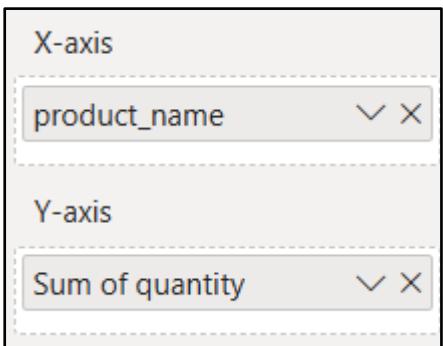
22. How do you create a forecast for future stock levels based on historical stock received data (Table 4 & Table 5)?



**1 Build a Line Chart:**  
Create a line chart with Date on the X-axis and Stock Received on the Y-axis using the merged stock table.

**2 Enable Forecasting:**  
In the Analytics pane, add a Forecast line → set the forecast length and confidence interval.  
Power BI will project future stock levels based on historical trends.

23. Create a report to identify the top 5 best-selling products based on quantity ordered (Table 7).



**1 Rank Products by Quantity Sold:**

Use a bar chart or table to display products sorted by total quantity ordered, highlighting the top 5 best-sellers.

**2 Focus on High-Demand Items:**

This report helps identify which products drive the most sales, supporting inventory planning and promotional strategies.

24. Create a measure to calculate gross profit using margin\_percentage from Table 9.

```
1 Gross_Profit =  
2 SUMX(  
3     blinkit_products,  
4     blinkit_products[price] * (blinkit_products[margin_percentage] / 100)  
5 )
```

**35.99K**  
Gross Profit

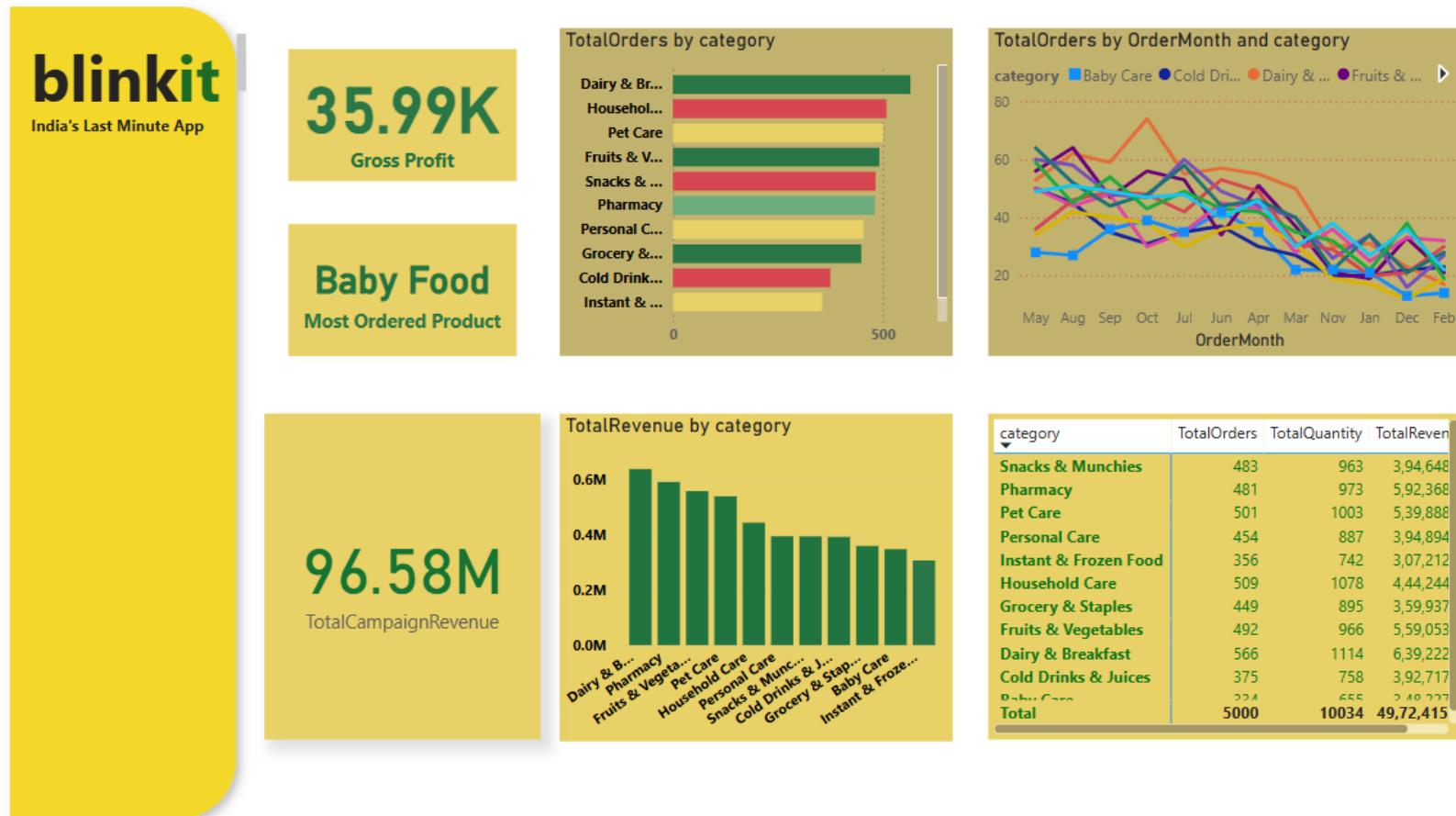
**1 Calculate Profit with DAX Logic:**

Use a measure that multiplies price by margin\_percentage to compute gross profit for each product.

**2 Assess Product Profitability:**

This measure highlights which items generate the most profit, guiding pricing and product strategy decisions.

25. How do you build a dashboard to track order trends across different categories (Table 9)?



## 1 Combine Multiple Visuals for Insights:

Use bar charts, line charts, KPIs, and tables to display total orders, revenue, and quantity by category, giving a complete view of category-wise performance.

## 2 Enable Trend Monitoring Over Time:

Use a line chart with OrderMonth and Category to visualize how order volumes fluctuate monthly across categories, helping spot seasonality and patterns.

26. Build a time-series analysis of daily order counts using order\_date from Table 8.

order_id	customer_id	order_date	store_id
9998297778	7728406	22-09-2023 17:07:28	5907
9997994965	13796199	27-12-2023 14:02:18	4117
9997121955	46623192	21-04-2024 00:03:15	6589
9996182478	39976965	18-04-2023 13:13:06	2861
9993542681	42589311	23-05-2023 04:12:41	4596
9992859795	4356173	12-11-2023 20:07:41	805
9991338690	98444183	29-07-2024 11:34:33	7390
9989636336	191616	24-05-2024 07:26:24	7821
9982178143	89968482	08-12-2023 14:07:42	2963
9981950902	39153585	28-07-2024 05:46:17	7848
9979010790	13465850	13-02-2024 01:25:09	6979
9977205005	22221200	11-09-2023 15:04:12	5020

store_id
1
2
9
12
14
15
16
17
18

**1** **Highlight Top Performing Stores:**

Stores **5907**, **6589**, and **7848** record the highest order volumes, indicating strong customer engagement in these locations.

**2** **Showcase Wide Operational Spread:**

Orders are placed across a diverse range of stores, reflecting Blinkit's extensive geographic footprint and customer reach.

27. How do you use DAX to find the most frequently ordered product?

Baby Food  
Most Ordered Product



```
1 Most Ordered Product =
2 VAR TopProduct =
3   TOPN(
4     1,
5     SUMMARIZE(
6       blinkit_order_items,
7       blinkit_order_items[product_id],
8       "TotalQty", SUM(blinkit_order_items[quantity])
9     ),
10    [TotalQty],
11    DESC
12  )
13 RETURN
14 CALCULATE(
15   SELECTEDVALUE(blinkit_products[product_name]),
16   FILTER(
17     blinkit_products,
18     blinkit_products[product_id] = MAXX(TopProduct, blinkit_order_items[product_id])
19   )
20 )
```

**1 Use DAX to Identify Top Product:**

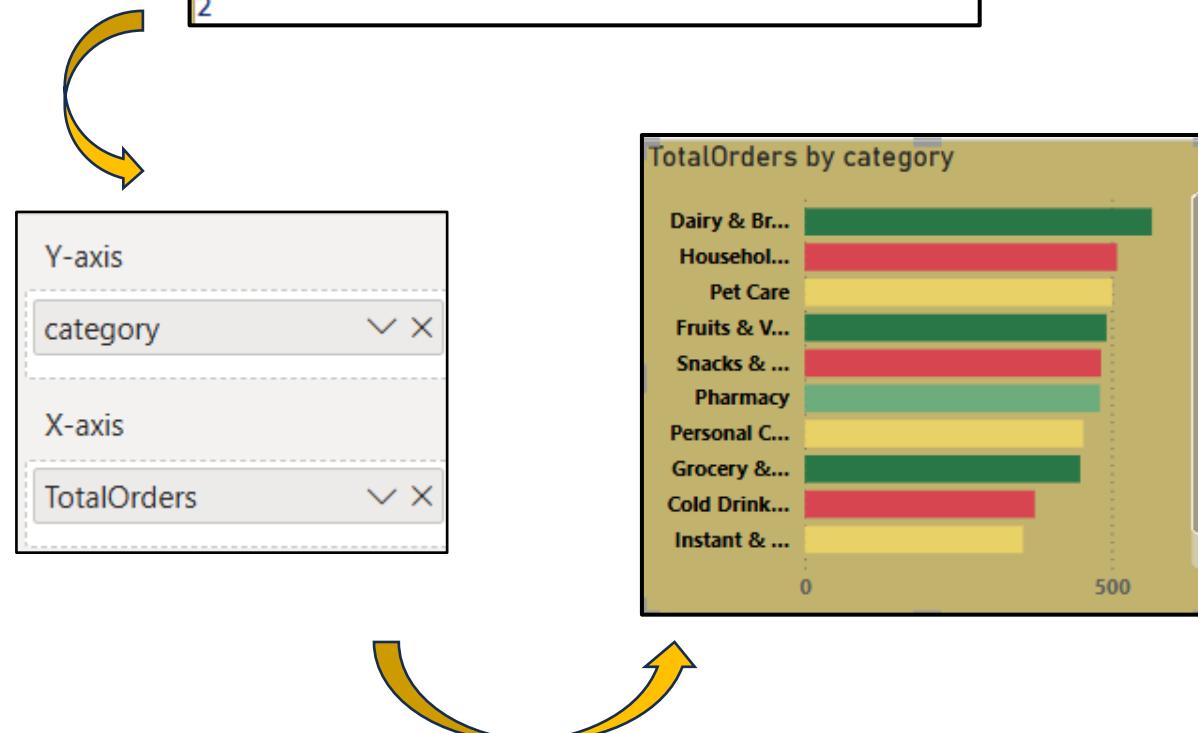
This measure dynamically finds the most frequently ordered product by ranking based on total quantity across all orders.

**2 Display in a Card Visual:**

The result is shown as a single product name (e.g., "Baby Food") in a card, giving an instant view of top-performing inventory.

28. Create a matrix visualization to show order quantity per product category (Table 9).

```
1 TotalOrders = COUNTROWS(blinkit_order_items)  
2
```



## 1 Summarize Data in a Matrix Format:

Use a matrix visual with category in Rows and TotalOrders in Columns, showing total quantity as the Values field.

## 2 Enable Detailed Comparison:

The matrix allows easy comparison of how different products perform within each category, supporting better inventory and sales decisions.

29. How do you calculate customer lifetime value using avg\_order\_value and total\_orders (Table 2)?

```
1 CustomerLifetimeValue =  
2 blinkit_customers[avg_order_value] * blinkit_customers[total_orders]  
3
```



Columns	
customer_id	✓ X
customer_name	✓ X
Sum of CustomerLifeti...	✓ X

customer_id	customer_name	Sum of CustomerLifetimeValue
45383958	Aachal Mangat	9,161.61
14161586	Aachal Nazareth	3,893.67
15487049	Aadhyaa Cherian	28,920.00
87222820	Aadhyaa Padmanabhan	6,469.58
65618148	Aadhyaa Palla	14,800.59
35885052	Aadhyaa Ravi	1,838.87
79206969	Aadi Bains	721.13
10608845	Aadi Gole	1,470.72
65692224	Aahana Buch	3,159.99
44426129	Aahana Gopal	5,915.50
Total		2,89,08,720.95



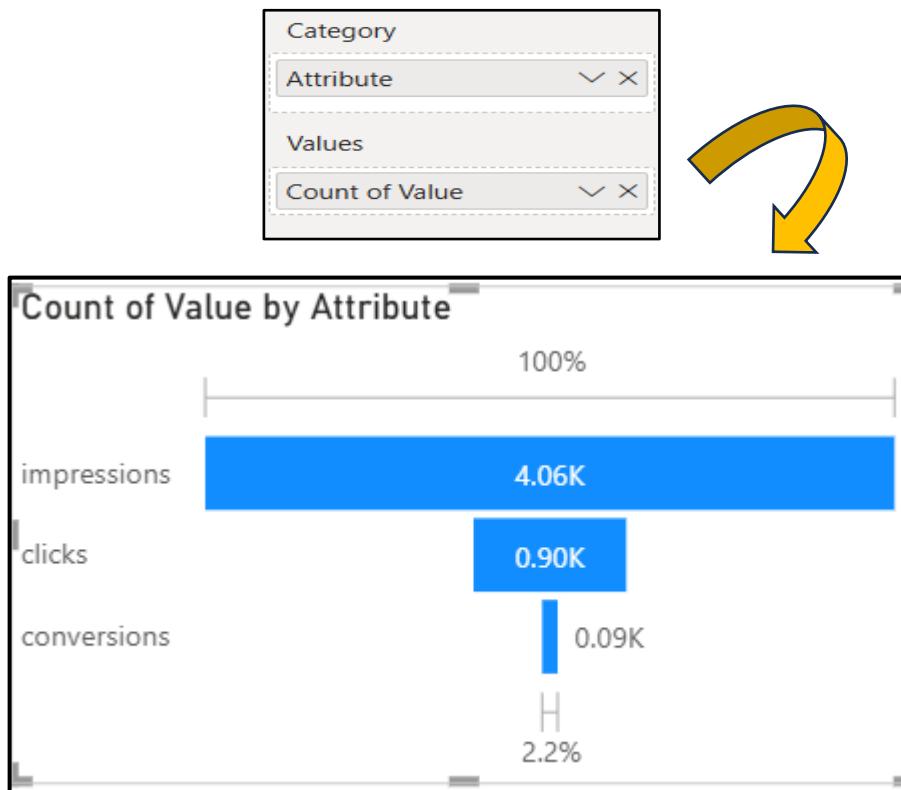
## 1 Calculate CLV Using Key Metrics:

Multiply avg\_order\_value by total\_orders to compute each customer's lifetime value, reflecting their overall contribution.

## 2 Identify High-Value Customers:

This metric helps segment and prioritize customers based on their long-term revenue potential.

30. How do you create a funnel chart to track the campaign conversion process (Table 6)?



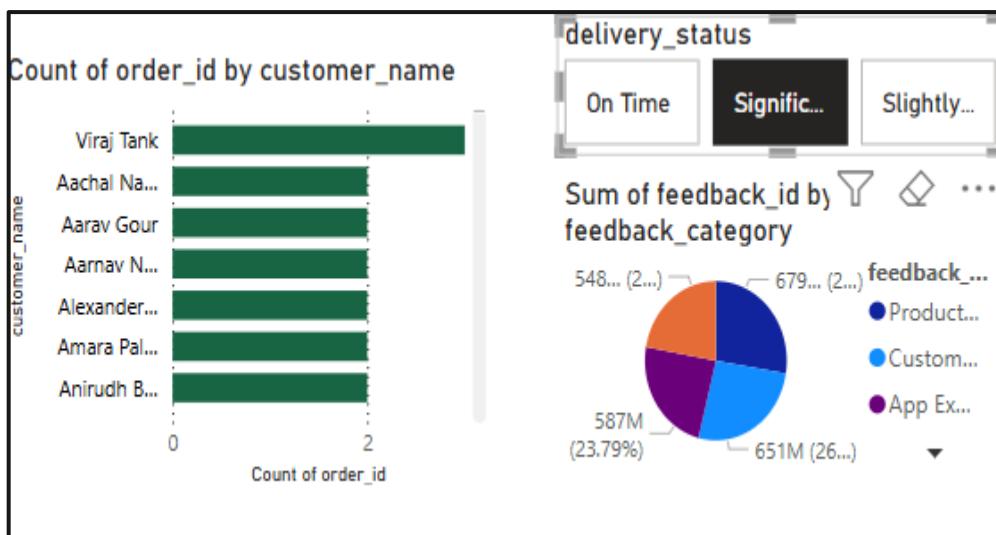
**1 Use Funnel Visual in Power BI:**

Select the Funnel chart from the Visualizations pane.

**2 Add Campaign Stages:**

Drag fields like Impressions → Clicks → Conversions (in order) to the Values section to visualize how users move through the campaign funnel.

31. Create a dynamic slicer for filtering orders by delivery status (on-time vs delayed).



**1 Use Delivery Status as a Filter:**

Apply the `delivery_status` field from Table 3 as a slicer or visual-level filter to view only on-time or delayed orders.

**2 Enable Focused Analysis:**

Filtering by delivery status helps monitor fulfillment performance and identify patterns in delivery delays.

32. Use Power BI parameters to filter data dynamically for a selected date range.

The screenshot shows a Power BI interface. At the top, there is a parameter field labeled "Field" with "feedback\_date" selected. Below it is a data table with columns "customer\_id" and "feedback\_category". The data includes rows for various customer IDs and their feedback categories such as "App Experience", "Delivery", "Product Quality", and "Customer Service". To the right of the table is a date slicer titled "feedback\_date" with two date inputs: "16-03-2023" and "04-11-2024". A horizontal line connects the date slicer to the data table, indicating its dynamic filtering effect.

customer_id	feedback_category
31813	App Experience
119099	App Experience
119099	Delivery
119099	Product Quality
188838	App Experience
191616	App Experience
191616	Product Quality
408590	App Experience
767523	Customer Service

**1 Enable Dynamic Date Filtering with Slicers:**

A slicer on `feedback_date` allows users to interactively filter feedback records within a specific date range.

**2 Analyze Time-Specific Feedback Trends:**

This setup helps track how customer feedback categories change over time, supporting timely response and service improvement.

### 33. Create a drill-through report to analyze feedback details (Table 1).

The diagram illustrates a drill-through process. On the left, a screenshot of a Power BI report shows a table with columns: customer\_id, customer\_name, and customer\_segment. The data includes rows for Indrajit Pau (Regular), Ekya Pau (New), Onkar Suresh (New), Jeet Gandhi (Pending), Zansi Parsa, Idika Basu, Brijesh Tata, Hiral Baral, Sudiksha Nagar, Ira Chhabra, Januja Zachariah, and a total row. A right-click context menu is open over the last row, with 'Drill through' selected. This leads to a second screenshot on the right, which is a detailed report titled 'customer\_id Sum of total\_orders address'. It shows a single row for customer\_id 75482 with a sum of total\_orders of 13 and the address 95/861, Jaggi, Sikar-343285. Below this is a 'Total' row with a value of 13.

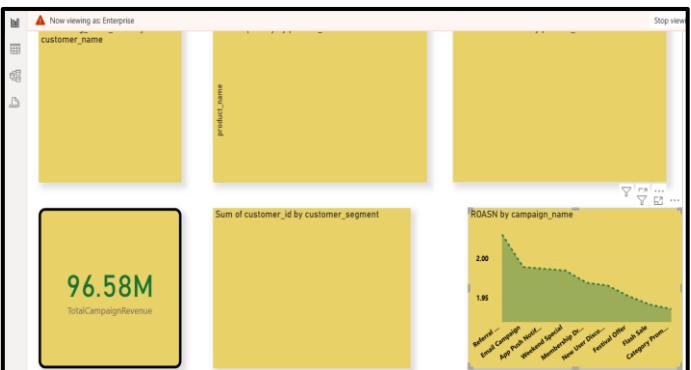
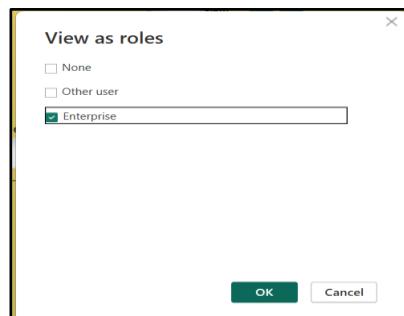
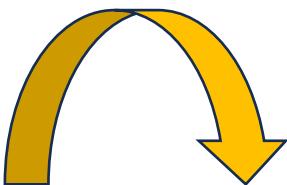
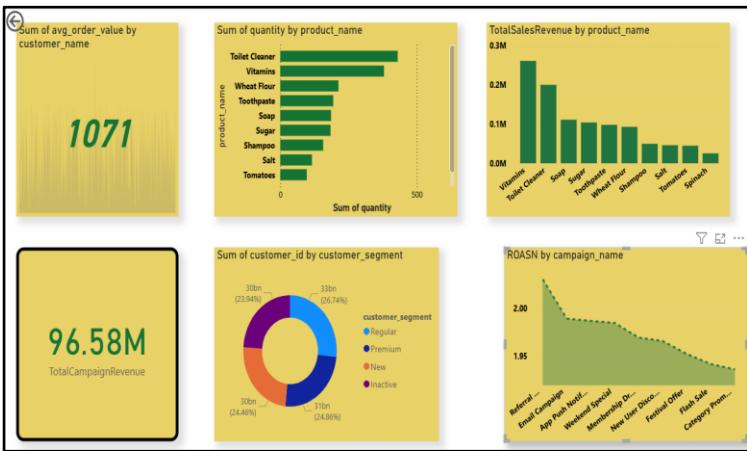
#### 1 Enable Drill-Through on customer\_id or feedback\_id:

A dedicated drill-through page is created where customer\_id or feedback\_id is added to the Drill-through field. This allows users to right-click on summarized visuals and explore detailed feedback records.

#### 2 Enable Dynamic Date Filtering with Feedback Date Slicer:

A slicer using feedback\_date is placed on the drill-through page. This lets users interactively filter and analyze feedback trends across specific time windows for more actionable insights.

34. How do you set up row-level security to restrict access by customer segment (Table 2)?



**1 Create a Role in Power BI Desktop:**  
 Use Modeling → Manage Roles and apply the filter:  
`[customer_segment] = "Enterprise"`

**2 Assign Users in Power BI Service:**  
 After publishing, go to Dataset → Security and assign users to the EnterpriseAccess role to restrict their view to Enterprise data only.

## 35. Use Power Query to clean and transform customer data before loading it into Power BI.

The screenshot shows the Microsoft Power Query Editor interface. The 'Transform' ribbon tab is active. In the main area, there is a table with columns: customer\_id, customer\_name, email, address, and phone. The 'Applied Steps' pane on the right shows a step named 'Removed Duplicates'. The status bar at the bottom indicates '11 COLUMNS, 999+ ROWS'.

### 1 Remove Duplicates from blinkit\_customers:

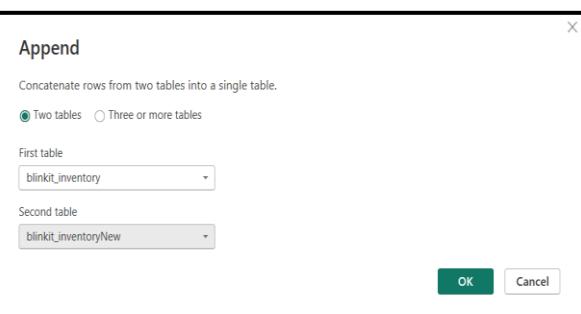
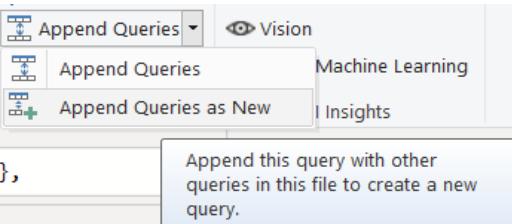
In Power Query, select relevant columns in the blinkit\_customers table → right-click → choose Remove Duplicates to ensure each customer record is unique.

### 2 Load Cleaned Data into Power BI Model:

After transformation, click Close & Apply to load the cleaned blinkit\_customers table into the Power BI data model for accurate reporting.

36. How do you merge stock data from Table 4 & Table 5 to create a unified stock report?

> blinkit\_inventory  
> blinkit\_inventory\_new



> blinkit\_inv\_comb  
> blinkit\_inventory  
> blinkit\_inventory\_new

## 1 Merge Tables in Power Query:

In Power Query, use Append Queries to combine Table 4 and Table 5 into one. Make sure both have matching columns (e.g., product\_id, stock\_in, date). Rename the result as inventory\_new.

## 2 Load into Power BI Model:

Click Close & Apply to load inventory\_new into Power BI. You can now use it to build unified stock-level visuals and reports.

37. Build a Power BI alert that triggers when stock levels go below the minimum threshold (Table 9).

```
1 LowStockCount =  
2 CALCULATE(  
3     COUNTROWS(blinkit_products),  
4     blinkit_products[max_stock_level] < blinkit_products[min_stock_level]  
5 )
```

### 1 Create a Measure:

Use the pre-defined measure LowStockCount to calculate products where `max_stock_level < min_stock_level`.

### 2 Set Alert in Power BI Service:

Add the measure to a card visual, publish the report, then go to ... > Manage Alerts on the card to create an alert that triggers when the value crosses a defined limit.

38. Create a report to track customer orders by store\_id (Table 8).

order_id	customer_id	order_date	store_id
9998297778	7728406	22-09-2023 17:07:28	5907
9997994965	13796199	27-12-2023 14:02:18	4117
9997121955	46623192	21-04-2024 00:03:15	6589
9996182478	39976965	18-04-2023 13:13:06	2861
9993542681	42589311	23-05-2023 04:12:41	4596
9992859795	4356173	12-11-2023 20:07:41	805
9991338690	98444183	29-07-2024 11:34:33	7390
9989636336	191616	24-05-2024 07:26:24	7821
9982178143	89968482	08-12-2023 14:07:42	2963
9981950902	39153585	28-07-2024 05:46:17	7848
9979010790	13465850	13-02-2024 01:25:09	6979
9977205005	22221200	11-09-2023 15:04:12	5020

store_id
1
2
9
12
14
15
16
17
18

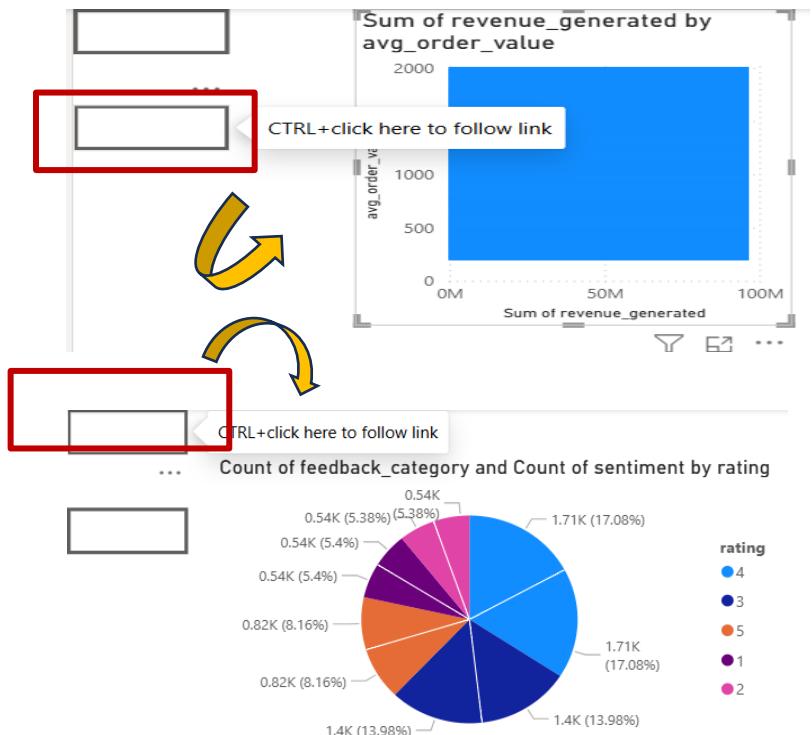
**1 Highlight Top Performing Stores:**

Stores **5907**, **6589**, and **7848** record the highest order volumes, indicating strong customer engagement in these locations.

**2 Showcase Wide Operational Spread:**

Orders are placed across a diverse range of stores, reflecting Blinkit's extensive geographic footprint and customer reach.

## 39. How do you use bookmarks to create different views of the dashboard?



### 1 Create Bookmarks in Power BI:

Set up visuals the way you want, then go to View > Bookmarks Pane → click Add to save the current view as a bookmark.

### 2 Link with Buttons:

Insert a button or shape, then assign the bookmark to it via Action > Type: Bookmark.

Now, clicking the button switches the view as shown in your image.

40. Build a map visualization showing order density per area (Table 2).



**1 Use Area or Pincode Field in Map Visual:**

Drag the area or pincode column from Table 2 into the Map visual's Location field, and use order\_id in the Values field set to Count.

**2 Visualize Order Density Geographically:**

The map now displays bubble sizes or heat points representing the number of orders per area, helping identify high-demand locations.



## Key Findings:

- **Top Customers:**

Customers like Viraj Tank and Aachal Nazareth placed the most orders.

- **Feedback Focus:**

Most feedback relates to Product Quality and App Experience.

- **Stock Trends:**

Cough Syrup and Pet Treats had steady stock received in March.

- **Campaign ROI:**

Weekend Special and Referral Program earned the highest revenue.

- **Delivery Status:**

Mix of on-time and delayed deliveries shows scope for improvement.

## Conclusion:

- Gained hands-on experience with Power BI dashboards, DAX, and data modeling
- Learned to clean, transform, and visualize real-world datasets
- Analysis supports better inventory planning, marketing strategy, and customer retention efforts

**blinkit**

 Power BI

## GitHub Repository Link:

<https://github.com/Debjitsutar/blinkit-powerbi-project>

 This repository contains the complete Power BI project, including datasets, .pbix file, visualizations, and report documentation.



## References:

**Kaggle** – Blinkit Sales Dataset Source of raw data for customer, orders, products, inventory, delivery, and marketing performance.

**Power BI Documentation** –

<https://learn.microsoft.com/en-us/power-bi/> For DAX functions, data modeling, and visualization best practices.

**Microsoft Power Query Docs** –

<https://learn.microsoft.com/en-us/power-query/> used for data cleaning, merging, and transformation workflows.

**OpenAI ChatGPT** – Assisted in writing DAX measures, structuring analysis, and presentation formatting.

**GitHub** – Version control and storage of project files and .pbix report.

**YouTube Tutorials** – For Power BI advanced visualizations and performance optimization techniques.



# Thank You

Debjit Sutar

✉️ [debjitsutar65@gmail.com](mailto:debjitsutar65@gmail.com)

🔗 [linkedin.com/in/debjit-sutar-65a97a31b](https://linkedin.com/in/debjit-sutar-65a97a31b)

👉 Special thanks to **Classroom Tech** for their constant guidance and support throughout the internship journey.

This experience helped me apply classroom knowledge to real-world data using Power BI. Grateful for the opportunity to learn, build, and grow professionally.