2025

Bright Sales Analysis

ANNA NTO MPAHLENI

BRIGHTLIGHT TUTORIALS

Contents

Nto Anna M

Bright Sales Analysis

Objectives according to the metrics required.

1. Daily Sales Price Per Unit

- Formula: Sales / Quantity Sold
- This gives the price at which each unit was sold daily.

2. Average Unit Sales Price

- Formula: AVG(Sales / Quantity Sold)
- This gives the overall average selling price of a unit over the dataset.

3. Daily % Gross Profit

- Formula: ((Sales - Cost of Sales) / Sales) * 100
- This metric shows what percentage of the sales value was profit.

4. Daily % Gross Profit Per Unit

- Formula: ((Sales - Cost of Sales) / Quantity Sold)
- This tells you the profit margin per unit sold daily.

5. Price Elasticity of Demand

- Formula: (% Change in Quantity Sold) / (% Change in Price)
- Pick 3 periods where promotions occurred, calculate the changes, and analyze elasticity.

6. Additional Insights

- Generate **KPIs**, such as:
    - Highest and lowest daily sales.
    - Trends in sales price over time.
    - Seasonal patterns in sales.

1. Data Collection

The sales data was sourced from the Brightlight internal database, covering transactions from **January 2024 to April 2025**. This dataset includes:

- **Daily sales transactions**, categorized by product type and region.
- **Customer purchase histories** for trend analysis.
- **Pricing fluctuations** to evaluate price elasticity of demand.

To ensure accuracy, missing values were handled using SQL's COALESCE function and verified against reference datasets.

2. Data Processing & Cleaning

The dataset was cleaned and pre-processed using **Snowflake SQL** and **Python (pandas)**:

- **Duplicate records** were removed by using SQL's DISTINCT clause.
- **NULL values** in revenue and quantity fields were replaced with appropriate estimates (COALESCE in SQL).
- Outliers in pricing and sales volume were identified using **interquartile range (IQR) analysis** in Python.

3. Analytical Techniques

This analysis focused on key performance indicators (KPIs) for **sales monitoring**:

- **Gross Profit Percentage** = (Total Revenue - Cost of Goods Sold) / Total Revenue * 100
- **Price Elasticity of Demand** = (Percentage Change in Quantity Sold) / (Percentage Change in Price)
- **Sales Growth Rate** = (Current Sales - Previous Sales) / Previous Sales * 100
- **Customer Retention Rate** = (Returning Customers / Total Customers) * 100

SQL queries in **Snowflake** aggregated sales data using GROUP BY and HAVING, while Python (pandas) helped visualize sales trends.

```sql
SELECT
    Date,
    ((Sales - Cost_Of_Sales) / Sales) * 100 AS Gross_Profit_Percentage
FROM sales_analysis;

WITH price_changes AS (
    SELECT
        Date,
        LAG(Sales / Quantity_Sold) OVER (ORDER BY Date) AS Prev_Daily_Unit_Price,
        LAG(Quantity_Sold) OVER (ORDER BY Date) AS Prev_Quantity_Sold,
        (Sales / Quantity_Sold) AS Current_Daily_Unit_Price,
        Quantity_Sold AS Current_Quantity_Sold
    FROM sales_analysis
)
SELECT
    Date,
    ((Current_Quantity_Sold - Prev_Quantity_Sold) / Prev_Quantity_Sold) AS Quantity_Change_Percentage,
    ((Current_Daily_Unit_Price - Prev_Daily_Unit_Price) / Prev_Daily_Unit_Price) AS Price_Change_Percentage,
    ((Current_Quantity_Sold - Prev_Quantity_Sold) / Prev_Quantity_Sold) /
    ((Current_Daily_Unit_Price - Prev_Daily_Unit_Price) / Prev_Daily_Unit_Price) AS Price_Elasticity
FROM price_changes
WHERE Prev_Quantity_Sold IS NOT NULL AND Prev_Daily_Unit_Price IS NOT NULL;
```

4. Visualization & Interpretation

Data visualization was conducted using:

- **Matplotlib & Seaborn** for trend analysis and correlation plots.
- **Bar charts** to track regional sales performance.
- **Scatter plots** to analyze price elasticity and customer behavior.

These visualizations identified seasonal trends, high-performing products, and price sensitivity across customer segments.

```python
[ ] df["Price_Per_Unit"] = df["Sales"] / df["Quantity Sold"]

    plt.figure(figsize=(10,5))
    plt.scatter(df["Price_Per_Unit"], df["Quantity Sold"], alpha=0.7, color="purple")
    plt.xlabel("Price Per Unit")
    plt.ylabel("Quantity Sold")
    plt.title("Price Elasticity Scatter Plot")
    plt.show()
```

```python
df = pd.read_csv('/content/drive/MyDrive/bright sales analysis/Sales Case 3 Study.csv')
df["Date"] = pd.to_datetime(df["Date"], dayfirst=True)
df.sort_values("Date", inplace=True)

plt.figure(figsize=(12, 6))
plt.plot(df["Date"], df["Sales"], marker="o", linestyle="-", label="Daily Sales")
plt.xlabel("Date")
plt.ylabel("Sales (Rand)")
plt.title("Daily Sales Trend")
plt.xticks(rotation=45)
```

Conclusion

- The analysis assumes **consistent pricing models** across regions.
- Data gaps were minimized, but potential discrepancies in **discounted sales** were acknowledged.
- External economic factors affecting demand were not incorporated.