

# PIZZA SALES ANALYSIS

## PROJECT REPORT

**Time Period:** 15 January – 15 December

**Tools Used:** SQL, Python, Power BI

---

### 1. Introduction

The objective of this project is to perform an in-depth analysis of pizza sales to uncover meaningful insights related to revenue, product performance, customer order patterns, and sales trends. Using a combination of **SQL**, **Python**, and **Power BI**, we transform raw transactional data into actionable insights that help optimize menu strategy, improve operational efficiency, and enhance business decision-making.

---

### 2. Dataset Summary

**Dataset Name:** pizza\_sales.csv

**Rows:** ~4,000

**Columns:** ~12

#### Key Columns

Column	Description
<b>order_id</b>	Unique ID per order
<b>pizza_id</b>	Unique pizza ID
<b>pizza_name</b>	Name of the pizza
<b>pizza_category</b>	Category—Classic, Supreme, Veggie, Chicken
<b>pizza_size</b>	S, M, L, XL
<b>quantity</b>	Quantity of pizzas ordered
<b>order_date</b>	Date of order
<b>order_time</b>	Time of order
<b>unit_price</b>	Price of one pizza
<b>total_price</b>	Revenue generated
<b>ingredients</b>	Items used (optional column)

---

### 3. Business Problem Statement

A pizza restaurant chain wants to understand its sales performance over the year to improve profitability and product offerings. The business needs clarity on:

- Which pizzas generate the most revenue

- What categories and sizes drive customer preferences
- When customer demand peaks (daily, monthly, hourly trends)
- Which pizzas underperform and affect profitability

To solve these problems, we define **KPI requirements** and **chart requirements** that will guide our analysis.

---

## 4. KPI Requirements

We calculated key performance indicators (KPIs) to measure business health:

### 1 Total Revenue

The sum of the total price of all pizza orders.

### 2 Average Order Value (AOV)

$$\text{AOV} = \frac{\text{Total Revenue}}{\text{Total Number of Orders}}$$

Represents the average amount spent per order.

### 3 Total Pizzas Sold

The sum of quantities of all pizzas sold.

### 4 Total Orders

The total number of customer orders placed.

### 5 Average Pizzas Per Order

$$\text{Avg Pizzas per Order} = \frac{\text{Total Pizzas Sold}}{\text{Total Orders}}$$

Shows how many pizzas customers typically purchase per order.

---

## 5. Chart Requirements

To visualize and interpret the sales patterns effectively, the following visual requirements were defined:

### A. Trend Analysis

#### 1 Daily Trend for Total Orders

A bar chart showing daily order volume across the week.

Helps identify peak days and low-traffic periods.

#### 2 Hourly Trend for Total Orders

A line chart representing hourly order distribution.

Useful for identifying rush hours and planning staffing levels.

## B. Category Performance Analysis

### 3 Percentage of Sales by Pizza Category

A pie chart showing the revenue contribution of each category (Classic, Supreme, Veggie, Chicken).

### 4 Percentage of Sales by Pizza Size

A pie chart illustrating customer preference across pizza sizes (S, M, L, XL).

## C. Product Sales Analysis

### 5 Total Pizzas Sold by Pizza Category

A funnel chart comparing total pizzas sold within each category.

### 6 Top 5 Best-Selling Pizzas (by Quantity Sold)

A bar chart listing the 5 pizzas with the highest total quantity sold.

### 7 Bottom 5 Worst-Selling Pizzas (by Quantity Sold)

Shows low-performing pizzas that may require recipe changes, promotions, or removal.

---

## 6. Data Cleaning & Preparation

Performed using **SQL** and **Python (Pandas)**:

- Converted date/time formats
  - Removed duplicate records
  - Validated numeric fields (unit price, quantity)
  - Created new calculated fields:
    - Sales Amount
    - Day Name
    - Hour of Order
    - Pizza Category
    - Size Groups
  - Built aggregated tables for Power BI import:
    - Daily Sales Summary
    - Monthly Sales Summary
    - Category-wise Sales
    - Pizza-wise Performance Tables
-

## 7. Exploratory Data Analysis (Python + SQL)

### ✓ Python (Jupyter Notebook)

- Data profiling
- Null value handling
- Outlier detection
- Exploratory charts (matplotlib/seaborn)

### ✓ SQL

- Total revenue and order count calculations
  - Aggregating sales by category, size, and pizza type
  - Time-series analysis (Daily, Monthly, Hourly)
  - Top-5 / Bottom-5 product queries
- 

## 8. Dashboard Overview (Power BI)

The interactive Power BI dashboard includes:

### KPI Cards

- Total Revenue
- Total Orders
- Total Pizzas Sold
- Average Order Value
- Average Pizzas per Order

### Trend Visuals

- Daily Order Trend
- Monthly Sales Trend
- Hourly Order Trend

### Category Performance

- Sales by Pizza Category
- Sales by Pizza Size

### Product Performance

- Top 5 Best-Selling Pizzas
- Bottom 5 Worst-Selling Pizzas
- Category-wise pizza sale comparisons

The dashboard enables users to filter by:

- Pizza Category
  - Pizza Size
  - Date Range
- 

## ★ 9. Key Insights

### ↗ Best Performers

- *Classic* pizzas generate the maximum sales.
- *Large* size pizzas contribute the highest revenue.
- Top pizzas have strong demand consistency.

### ↘ Worst Performers

- Certain pizzas contribute very low sales — ideal candidates for promotions or menu removal.

### ↗ Demand Patterns

- Orders peak on weekends, especially **Friday and Saturday**.
  - Evening hours show maximum order activity.
- 

## 💼 10. Business Recommendations

### 1 Strengthen High-Demand Products

- Bundle best-selling pizzas with sides or drinks.

### 2 Fix Low-Performing Pizzas

- Consider recipe improvements
- Introduce limited-time promotions
- Remove items with long-term low demand

### 3 Optimize Staffing & Inventory

- Increase staffing on peak weekend evenings
- Stock more ingredients for top-performing pizza types

### 4 Encourage Upselling

- Promote **Large-size pizzas** (highest revenue contributor)

### 5 Seasonal Promotions

- Introduce discounts in low-performing months
-

## **11. Conclusion**

This Pizza Sales Analysis provides insights that can significantly improve sales, operational efficiency, and customer satisfaction. Using SQL, Python, and Power BI, a complete data workflow has been created—from cleaning to visualization—making this strong end-to-end analytics project.

---

## **12. Appendix**

### **A. Dataset**

[https://github.com/awasthisakshi000-code/pizza\\_sales\\_trend\\_analysis/blob/master/dataset.csv](https://github.com/awasthisakshi000-code/pizza_sales_trend_analysis/blob/master/dataset.csv)

### **B. SQL Files**

[https://github.com/awasthisakshi000-code/pizza\\_sales\\_trend\\_analysis/blob/master/sql\\_query.sql](https://github.com/awasthisakshi000-code/pizza_sales_trend_analysis/blob/master/sql_query.sql)

### **C. Python Notebook**

[https://github.com/awasthisakshi000-code/pizza\\_sales\\_trend\\_analysis/blob/master/notebook.ipynb](https://github.com/awasthisakshi000-code/pizza_sales_trend_analysis/blob/master/notebook.ipynb)

### **D. Power BI Dashboard**

[https://github.com/awasthisakshi000-code/pizza\\_sales\\_trend\\_analysis/tree/master/dashboard.png](https://github.com/awasthisakshi000-code/pizza_sales_trend_analysis/tree/master/dashboard.png)

---

## **Presented by**

Sakshi Awasthi

[awasthi.sakshi000@gmail.com](mailto:awasthi.sakshi000@gmail.com)