PIZZA SALES ANALYSIS REPORT

Author - Dnyanesh Yeole

About the Author

Dnyanesh is a data enthusiast currently pursuing an M.Sc in Mathematics and Scientific Computing from NIT Warangal, India. With a strong foundation in mathematics and computer science, Dnyanesh possesses a particular interest in machine learning and data analysis. He has good knowledge of Python and SQL. He has hands-on experience working with large datasets, cleaning and preprocessing data, and applying statistical and machine learning techniques.

Dnyanesh aims to apply his knowledge and expertise in data analysis to solve real-world problems, leveraging data-driven approaches for informed decision-making. His enthusiasm for the field drives his motivation to make valuable contributions in understanding and utilizing data effectively.

CONTENTS

CHAPTER-1

1. INTRODUCTION

1.1 OBJECTIVES

- The main objective of this project is to identify trends, patterns in the data which can lead to improved sales performance and customer satisfaction.
- Gain insights into customer preferences, sales patterns, and key factors influencing pizza sales.
- Understand customer preferences for pizza toppings.
- Evaluate sales performance across different pizza sizes and crust types.
- Identify popular pizza combinations and variations.
- Assess the impact of promotional activities on sales volume.
- Analyze customer segmentation and preferences for targeted marketing efforts.
- Determine correlations between specific pizza attributes and customer satisfaction.
- Uncover potential seasonal or time-of-day trends in pizza sales.
- Identify repeat customer patterns and loyalty trends.

1.2 LIMITATIONS

- The analysis assumes that the provided data accurately represents real-world pizza sales and customer behavior.
- Limited dataset size, which may not capture the entire scope of pizza sales patterns.
- The analysis is based on a specific dataset, which may not represent the entire population of pizza sales or customer preferences
- The analysis may not account for external factors, such as changes in the competitive landscape, economic conditions, or cultural trends, which could impact pizza sales
- Findings should be interpreted within the context of the specific business or industry and may not be universally applicable.
- Different interpretations or perspectives of the results are possible, and alternative explanations should be considered.

1.3 CHALLENGES

- Difficulty in identifying and targeting specific customer segments with tailored marketing strategies.
- Limited understanding of the impact of promotional activities on sales volume and customer engagement.
- Inconsistent sales performance across different locations and pizza variations.

CHAPTER 2

2. METHODOLOGY

2.1 DATA COLLECTION

The pizza sales data used in this analysis is collected from Kaggle which is the Data Science community. The link to the dataset is given below:
https://www.kaggle.com/datasets/shilongzhuang/pizza-sales

2.2 DATASET DESCRIPTION

The dataset contains information about pizza sales, including details such as pizza ID, order ID, pizza name ID, quantity, order date, order time, unit price, total price, pizza size, pizza category, pizza ingredients, and pizza name. Each column represents a specific attribute related to the pizza sales data.The dataset encompasses 48621 orders of pizza and customer transactions.
This pizza sales dataset make up 12 relevant features:

- **pizza_id**: The unique identifier for each pizza in the dataset.
- **order_id**: The unique identifier for each pizza order.
- **pizza_name_id**: The identifier for each specific pizza name.
- **quantity**: The number of pizzas ordered in each transaction.
- **order_date**: The date when the pizza order was placed.
- **order_time**: The time at which the pizza order was placed.
- **unit_price**: The price of a single unit of pizza.
- **total_price**: The total price of the pizza order, calculated as the unit price multiplied by the quantity.
- **pizza_size**: The size or dimensions of the pizza. (S,M,L,XL,XXL)
- **pizza_category**: The category or classification of the pizza, indicating its type or style. (Classic, Veggie, Supreme, Chicken)
- **pizza_ingredients**: The list of ingredients used in the pizza preparation.
- **pizza_name**: The name or label assigned to each specific pizza.

The screenshot of the first 5 rows of table is given below:

	A	B	C	D	E	F	G	H	I	J	K	L
1	pizza_id	order_id	pizza_nam	quantity	order_date	order_tim	unit_price	total_pric	pizza_size	pizza_cate	pizza_ingredients	pizza_name
2	1	1	hawaiian_	1	01-01-2015	11:38:36	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	The Hawaiian Pizza
3	2	2	classic_dl	1	01-01-2015	11:57:40	16	16	M	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, Bacon	The Classic Deluxe Pizza
4	3	2	five_chee	1	01-01-2015	11:57:40	18.5	18.5	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Gouda Cheese, Romano Cheese,	The Five Cheese Pizza
5	4	2	ital_supr_	1	01-01-2015	11:57:40	20.75	20.75	L	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, Green Olives, Garlic	The Italian Supreme Pizza
6	5	2	mexicana_	1	01-01-2015	11:57:40	16	16	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red Onions, Cilantro, Corn, Chipotle	The Mexicana Pizza

2.3 DATA PREPROCESSING

- **Handling Missing Values:** No missing value present in the dataset
- **Removing Duplicates:** No duplicated present in the dataset
- **Checking Data Type of attributes:**
    - orders_date column datatype is incorrect.
    - orders_time column datatype is incorrect

```
mysql> show columns from pizza_sales;
+-------------------+--------+------+-----+---------+-------+
| Field             | Type   | Null | Key | Default | Extra |
+-------------------+--------+------+-----+---------+-------+
| pizza_id          | int    | YES  |     | NULL    |       |
| order_id          | int    | YES  |     | NULL    |       |
| pizza_name_id     | text   | YES  |     | NULL    |       |
| quantity          | int    | YES  |     | NULL    |       |
| order_date        | text   | YES  |     | NULL    |       |
| order_time        | text   | YES  |     | NULL    |       |
| unit_price        | double | YES  |     | NULL    |       |
| total_price       | double | YES  |     | NULL    |       |
| pizza_size        | text   | YES  |     | NULL    |       |
| pizza_category    | text   | YES  |     | NULL    |       |
| pizza_ingredients | text   | YES  |     | NULL    |       |
| pizza_name        | text   | YES  |     | NULL    |       |
+-------------------+--------+------+-----+---------+-------+
```

Changing the datatype of order_date and order_time column:

```
ALTER TABLE pizza_sales
MODIFY COLUMN order_time TIME;

ALTER TABLE pizza_sales
MODIFY COLUMN order_date DATE;
```

Table: pizza_sales

Columns:

pizza_id	int
order_id	int
pizza_name_id	text
quantity	int
order_date	date
order_time	time
unit_price	double
total_price	double
pizza_size	text
pizza_category	text
pizza_ingredients	text
pizza_name	text

- **Removing Data Redundancy:** The given dataset is in 1NF i.e. first normal form. To remove data redundancy, we need to apply normalization techniques. We can apply following steps in order to remove data redundancy
    - **Identify the the functional dependencies between attributes**
    - **Establish relationships between attributes**
    - **Divide the tables**
    - **Define Primary Keys and Foreign Keys**
    - **Normalize the tables**

2.4 NORMALIZATION PROCESS

The normalization process aims to organize the dataset into multiple tables, minimize redundancy, and establish relationships between entities. The brief Overview of Normal Forms is given below:

- **First Normal Form (1NF)**:
    - Each attribute in a table must contain only atomic values
    - There should not be any duplicate row
- **Second Normal Form (2NF)**:
    - The table must be in 1NF
    - There should not be any **Partial dependency** in the relation i.e. all non-key attributes should be dependent on the entire primary key.
- **Third Normal Form (3NF)**:
    - The table must be in 2NF
    - There should not be any **Transitive dependency** in the relation i.e. all non-key attributes should be dependent only on the primary key and not on other non-key attributes.
- **Boyce-Codd Normal Form (BCNF)**:
    - All non-key attributes must be functionally dependent on the entire primary key
    - There should be no non-trivial dependencies between the candidate keys

To determine the normal form of the pizza_sales table:

"pizza_ingredients" attribute contains a list of ingredients; it violates 1NF because it would be considered a multi-valued attribute. To conform to 1NF, we need to remove the multi-valued attribute and create a separate table to represent the relationship between pizzas and ingredients. Since, "pizza_ingredients" is a multi-valued attribute, the original "pizza_sales" table is **not in 1NF**. We need to obtain highest normal form i.e BCNF

To identify Functional Dependencies in the pizza_sales table:

To begin the normalization process, we analyzed the relationships and dependencies between attributes in the dataset. Based on this analysis, the following functional dependencies were identified:

1. pizza_id -> pizza_name_id, pizza_size, pizza_category, pizza_ingredients, pizza_name
2. order_id -> quantity, order_date, order_time
3. pizza_name_id -> pizza_name
4. pizza_name -> pizza_ingredients, pizza_category

Now, we have a relation R i.e. pizza_sales with 12 attributes and above functional dependencies. Splitting the multivalued attribute `pizza_ingredients` is not possible or practical, we are not going to obtain 1NF.

To find the Candidate keys for the given relation:

To determine the candidate keys in a relation, we need to identify the minimal set of attributes that can uniquely identify each tuple in the relation. From the given attributes of the "pizza_sales" relation we get following candidate keys:

1. “pizza_id”
2. (“order_id”, “pizza_name_id”)

To find the decomposed tables:

1. **Pizza Table:**
    a. **Attributes: pizza_name_id (Primary Key), pizza_name, pizza_size, pizza_category, pizza_ingredients**
    b. This table represents different types of pizzas. The primary key is pizza_name_id, which uniquely identifies each pizza. It includes attributes such as pizza_name, pizza_size, pizza_category, and pizza_ingredients.
2. **Orders Table:**
    a. **Attributes: order_id (Primary Key), quantity, order_date, order_time**
    b. This table stores order-related information. The primary key is order_id.
3. **OrdersPizza Table:**
    a. **Attributes: order_id (Foreign Key), pizza_name_id (Foreign Key)**
    b. This table represents the relationship between orders and pizzas. It captures the information about which pizzas were included in each order. The primary key can be a composite key consisting of the foreign keys (order_id, pizza_name_id).

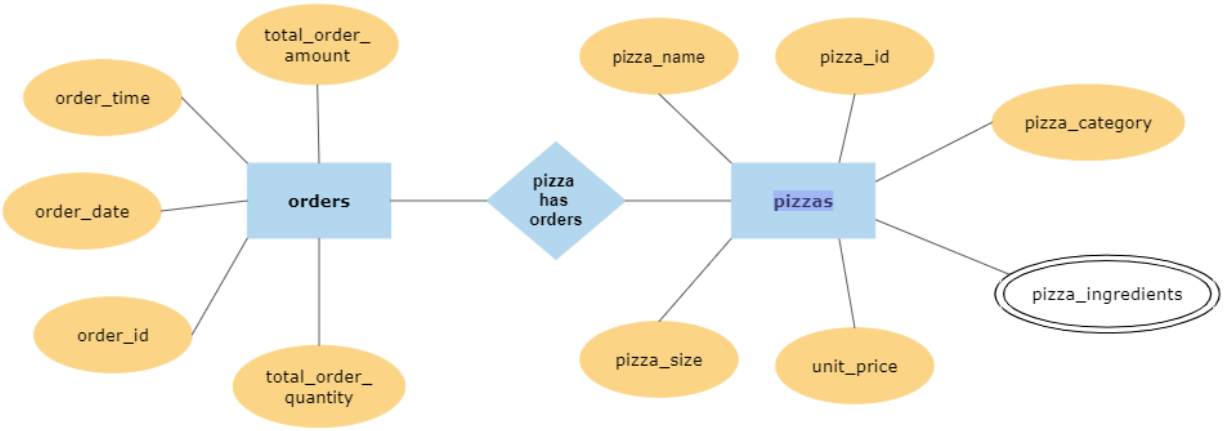
2.5 FEATURE ENGINEERING

In the given pizza_sales dataset, pizza_id is the primary key. Since we have decomposed it into 2 tables, we have omitted two attributes: pizza_id and total_price (which can be obtained by multiplying quantity and unit_price). Also we have renamed pizza_name_id as pizza_id.

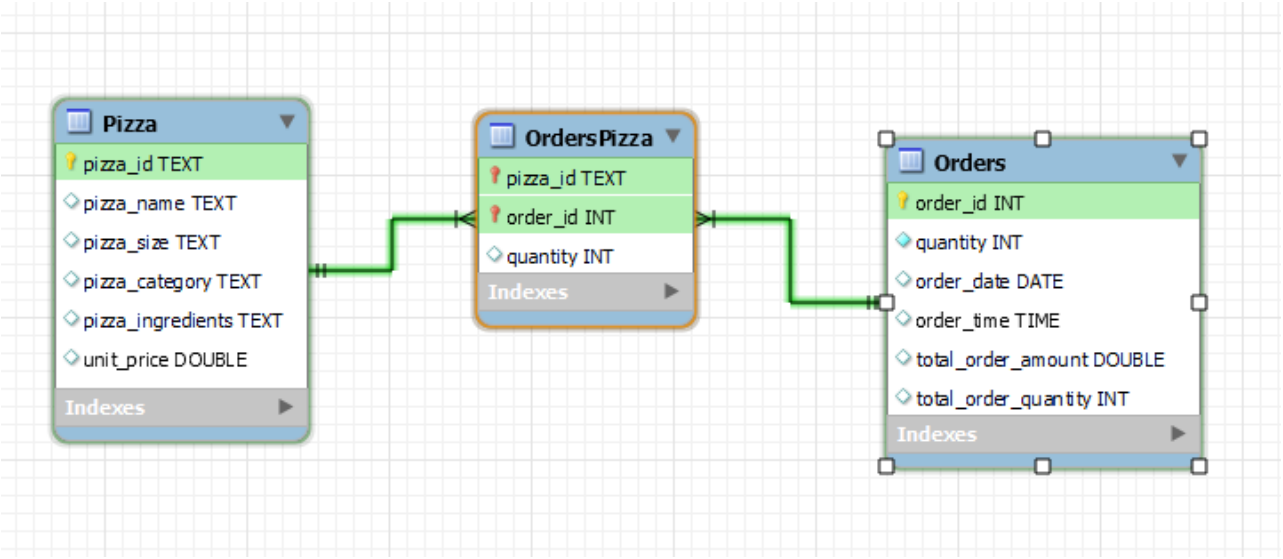
We have created some new features which may be helpful. We added total_order_amount and total_order_quantity to the Orders table. The new schema of Orders Table are as follows

Orders Table -> order_id (Primary Key), quantity, order_date, order_time, **total_order_amount**, **total_order_quantity**

2.5 E-R DIAGRAM



2.6 SCHEMA DIAGRAM



CHAPTER 3

3. DATA ANALYSIS USING SQL

3.1 PROBLEM STATEMENTS

KPI's REQUIREMENT

We need to analyze key performance indicators for our pizza sales data to gain insights into our business performance. Specifically, we want to calculate following metrics;

1. Total Revenue
2. Average Order Value
3. Total Pizza Sold
4. Total Orders
5. Average Pizzas Per Order

SECTOR WISE ANALYSIS

1. **Sales Performance Analysis**

    a. What is the average unit price and revenue of pizza across different categories?
    b. What is the average unit price and revenue of pizza across different sizes?
    c. What is the average unit price and revenue of most sold 3 pizzas?

2. **Seasonal Analysis**

    a. Which days of the week have the highest number of orders?
    b. At what time do most orders occur?
    c. Which month has the highest revenue?
    d. Which season has the highest revenue?

3. **Customer Behavior Analysis**

    a. Which pizza is the favorite of customers (most ordered pizza)?
    b. Which pizza is ordered the most number of times?
    c. Which pizza size is preferred by customers?
    d. Which pizza category is preferred by customers?

4. **Pizza Analysis**

    a. The pizza with the least price and highest price
    b. Number of pizzas per category
    c. Number of pizzas per size
    d. Pizzas with more than one category

3.2 SQL QUERIES USED IN ANALYSIS

KPI's REQUIREMENT

- **Total Revenue:**

```
-- Total Revenue

SELECT SUM(total_order_amount) AS Total_Revenue from orders;
```

Total_Revenue
817860.0499999926

The Total Revenue of the pizza sales is **$ 817,860.**

- **Average Order Value:**

```
-- Average Order value

SELECT AVG(total_order_amount) AS avg_order_value FROM orders;
```

avg_order_value
38.30726229508162

The Average Order Value i.e. the average amount spent per order is **$ 38.30**

- **Total Pizzas Sold**

```
-- Total Pizzas Sold

SELECT SUM(quantity) AS total_pizzas_sold FROM orderspizza;
```

total_pizzas_sold
49574

Total Pizzas Sold are **49,574**

- **Total Orders**

```
-- Total orders

SELECT COUNT(*) AS total_orders FROM orders;
```

total_orders
21350

There are **21,350** orders placed by customers

- **Average Pizzas per order**

```
-- Average Pizzas per order

SELECT SUM(total_order_quantity)/COUNT(*) AS avg_pizzas_per_order FROM orders;
```

	avg_pizzas_per_order
▶	2.3220

The average pizzas per order are **2.32**

SECTOR WISE ANALYSIS

We have created a view for better analysis:

```
CREATE VIEW pizza_sales_view AS
SELECT t1.order_id, t3.pizza_id, order_date, order_time, unit_price,
pizza_size, pizza_category, pizza_ingredients, pizza_name,
t1.quantity
FROM orderspizza t1
JOIN orders t2 ON t1.order_id = t2.order_id
JOIN pizza t3 ON t1.pizza_id = t3.pizza_id;
```

- **SALES PERFORMANCE ANALYSIS**

1. What is the average unit price and revenue of pizza across different categories?

```
-- What is the average unit price and revenue of pizza across different categories?

SELECT pizza_category, AVG(unit_price) AS average_unit_price,
SUM(quantity*unit_price) AS revenue_per_category
FROM pizza_sales_view
GROUP BY pizza_category
ORDER BY revenue_per_category DESC;
```

	pizza_category	average_unit_price	revenue_per_category
▶	Classic	14.797489539748964	220053.10000000015
	Supreme	17.36322917551144	208196.99999999822
	Chicken	17.70940822931114	195919.5
	Veggie	16.61263865839836	193690.45000000283

2. What is the average unit price and revenue of pizza across different sizes?

```
-- What is the average unit price and revenue of pizza across different sizes?

SELECT pizza_size, AVG(unit_price) AS average_unit_price,
SUM(quantity*unit_price) AS revenue_per_size
FROM pizza_sales_view
GROUP BY pizza_size
ORDER BY revenue_per_size DESC;
```

pizza_size	average_unit_price	revenue_per_size
L	19.802553168520383	375318.70000000845
M	15.951218719532012	249382.25
S	12.364327650845178	178076.49999999828
XL	25.5	14076
XXL	35.95000000000002	1006.6000000000005

3. What is the average unit price and revenue of most sold 3 pizzas?

```
-- What is the average unit price and revenue of most sold 3 pizzas?

SELECT pizza_name, AVG(unit_price) AS average_unit_price,
SUM(quantity*unit_price) AS revenue_per_pizza
FROM pizza_sales_view
GROUP BY pizza_name
ORDER BY revenue_per_pizza DESC
LIMIT 3;
```

pizza_name	average_unit_price	revenue_per_pizza
The Thai Chicken Pizza	18.28606911447084	43434.25
The Barbecue Chicken Pizza	17.57293423271501	42768
The California Chicken Pizza	17.448523023457863	41409.5

- **SEASONAL ANALYSIS**

1. Which days of the week have the highest number of orders?

```
-- Which days of the week have the highest number of orders?

SELECT DATE_FORMAT(order_date, '%W') AS day_of_week, COUNT(*) AS order_count
FROM Orders
GROUP BY day_of_week
ORDER BY order_count DESC;
```

	day_of_week	order_count
▶	Friday	3538
	Thursday	3239
	Saturday	3158
	Wednesday	3024
	Tuesday	2973
	Monday	2794
	Sunday	2624

2. At what time do most orders occur?

```
--  At what time most of the orders have occurred?

SELECT DATE_FORMAT(order_time, '%H:%i') AS delivery_time, COUNT(*) AS delivery_count
FROM Orders
GROUP BY delivery_time
ORDER BY delivery_count DESC
LIMIT 5;
```

	delivery_time	delivery_count
▶	13:04	68
	13:18	58
	13:05	56
	12:30	56
	12:22	55

3. Which month has the highest revenue?

```
-- Which month has highest revenue?

SELECT monthname(order_date) AS month, SUM(total_order_amount) AS revenue
FROM Orders
GROUP BY month
ORDER BY revenue DESC
LIMIT 5;
```

	month	revenue
▶	July	72557.89999999986
	May	71402.74999999988
	March	70397.09999999989
	November	70395.3499999999
	January	69793.2999999999

4. Which season has the highest revenue?

```
-- which season has highest revenue
SELECT
  CASE
    WHEN MONTH(order_date) IN (3, 4, 5) THEN 'Spring'
    WHEN MONTH(order_date) IN (6, 7, 8) THEN 'Summer'
    WHEN MONTH(order_date) IN (9, 10, 11) THEN 'Fall'
    ELSE 'Winter'
  END AS season,
  SUM(total_order_amount) AS season_sales
FROM orders
GROUP BY season
ORDER BY season_sales DESC;
```

season	season_sales
Spring	210536.65000000055
Summer	209066.35000000036
Winter	199654.05000000048
Fall	198603.0000000004

- **CUSTOMER BEHAVIOR ANALYSIS**

1. Which is the favorite pizza of customers (most ordered Pizza) ?

```
-- Which is the favourite pizza of customers (most ordered pizza)?

SELECT pizza_name, pizza_size, COUNT(order_id) AS count
FROM pizza_sales_view
GROUP BY pizza_name, pizza_size
ORDER BY count DESC
LIMIT 1;
```

pizza_name	pizza_size	count
The Big Meat Pizza	S	1811

2. Which pizza is ordered the most number of times?

```
-- Which pizza is ordered most number of times?

SELECT pizza_name, COUNT(order_id)  AS count, avg(unit_price)
FROM pizza_sales_view
GROUP BY pizza_name
ORDER BY count DESC
LIMIT 5;
```

pizza_name	count	avg(unit_price)
The Classic Deluxe Pizza	2416	15.575951986754967
The Barbecue Chicken Pizza	2372	17.57293423271501
The Hawaiian Pizza	2370	13.317194092827004
The Pepperoni Pizza	2369	12.468657661460531
The Thai Chicken Pizza	2315	18.28606911447084

3. Which Pizza size is preferred by customers ?

```
-- Which pizza size is preferred by cutomers?

SELECT pizza_size, COUNT(order_id) AS count, avg(unit_price)
FROM pizza_sales_view
GROUP BY pizza_size
ORDER BY count DESC;
```

pizza_size	count	avg(unit_price)
L	18526	19.802553168520383
M	15385	15.951218719532012
S	14137	12.364327650845178
XL	544	25.5
XXL	28	35.95000000000002

4. Which Pizza category is preferred by customers?

```
-- which pizza category is preferred by customers?

SELECT pizza_category, COUNT(order_id) AS count, avg(unit_price)
FROM pizza_sales_view
GROUP BY pizza_category
ORDER BY count DESC;
```

pizza_category	count	avg(unit_price)
Classic	14579	14.797489539748964
Supreme	11777	17.36322917551144
Veggie	11449	16.61263865839836
Chicken	10815	17.70940822931114

- **PIZZA ANALYSIS**

1. The pizza with the least price and highest price

```
-- pizza with least price
SELECT pizza_name AS lowest_priced_pizza, unit_price
FROM pizza
ORDER BY unit_price
LIMIT 1;

-- pizza with highest price
SELECT pizza_name  AS highest_priced_pizza, unit_price
FROM pizza
ORDER BY unit_price DESC
LIMIT 1;
```

lowest_priced_pizza	unit_price
The Pepperoni Pizza	9.75

highest_priced_pizza	unit_price
The Greek Pizza	35.95

2. Number of pizzas per category

```
-- number of pizzas per category

SELECT pizza_category, count(pizza_name) AS count
FROM pizza
GROUP BY pizza_category
ORDER BY count DESC;
```

pizza_category	count
Supreme	25
Classic	24
Veggie	24
Chicken	18

3. Number of pizzas per size

```
-- Number of pizzas per pizza_size

SELECT pizza_size,count(pizza_name) AS count
FROM pizza
GROUP BY pizza_size
ORDER BY count DESC;
```

pizza_size	count
L	30
S	30
M	29
XL	1
XXL	1

4. Pizzas with more than one category

```
-- PIZZAS WITH MORE THAN ONE CATEGORY

SELECT pizza_name, COUNT(DISTINCT pizza_category) AS category_count
FROM pizza
GROUP BY pizza_name
HAVING category_count > 1;
```

pizza_name	category_count

CHAPTER 4

4. DISCUSSION OF RESULTS

4.1 SUMMARY STATISTICS FROM ANALYSIS

- **The KPI's Requirement is tabular format:**

METRIC	VALUE
Total Revenue	$ 817,860
Average Amount Spent per Order	$ 38.30
Total Pizzas Sold	49,574
Total orders	21,350
Average Pizzas per Order	2.32

- **Highest and Lowest Revenue Generator Categories:**

CATEGORY	HIGHEST	LOWEST
Pizza	**The Thai Chicken Pizza** (Revenue - $ 43,434 Unit price - $ 18.28)	**The Brie Carre Pizza** (Revenue - $ 11,588 Unit price - $ 23.65)
Pizza Size	**Large (L)** (Revenue - $ 375,318 Unit price - $ 19.80)	**XXL** (Revenue - $ 1006 Unit price - $ 35.95)
Pizza Category	**Classic** (Revenue - $ 220,053 Unit price - $ 14.80)	**Veggie** (Revenue - $ 193,690 Unit price - $ 16.61)

- **Most and Least Ordered Categories (i.e most preferable categories by customers):**

CATEGORY	MOST ORDERED	LEAST ORDERED
Pizza	**The Big Meat Pizza** (Orders - 1811 Unit price - $ 18.28)	**The Brie Carre Pizza** (Orders - 28 Unit price - $ 23.65)
Pizza Size	**Large (L)** (Orders - 18526 Unit price - $ 19.80)	**XXL** (Orders - 28 Unit price - $ 35.95)
Pizza Category	**Classic** (Orders - 14579 Unit price - $ 14.80)	**Chicken** (Orders - 10815 Unit price - $ 17.70)

- **Seasonal Analysis Statistics:**

TIME PERIOD	HIGHEST REVENUE	LOWEST REVENUE	MOST ORDERED	LEAST ORDERED
Day of Week	**Friday** (Revenue - $ 136073)	**Sunday** (Revenue - $ 99203)	**Friday** (Orders - 3538)	**Sunday** (Orders - 2624)
Month	**July** (Revenue - $ 72557)	**October** (Revenue - $ 64027)	**July** (Orders - 1935)	**October** (Orders - 1646)
Season	**Spring** (Revenue - $ 210536)	**Fall** (Revenue - $ 198603)	**Summer** (Orders - 5549)	**Fall** (Orders - 5099)

- **Pizza Analysis Statistics:**

  - **Pizza with Highest and Lowest Price**

CATEGORY	HIGHEST PRICE	LOWEST PRICE
Pizza	**The Pepperoni Pizza** ($ 9.75)	**The Greek Pizza** ($ 35.95)

  - **The availability of pizzas per size and category**

CATEGORY	MOST AVAILABILITY	LEAST AVAILABILITY
PIZZA SIZE	**Large (L)** (30 Pizzas available)	**XXL** (1 Pizzas available)
PIZZA CATEGORY	**Supreme** (25 Pizzas available)	**Chicken** (18 Pizzas available)

4.2 TALKING ABOUT RESULTS

- This dataset contains information about pizza sales in 2015.
- Large size pizzas are preferred over other sizes.
- Classic category pizzas are preferred over other categories.
- Each pizza belongs to only one category out of the four available.
- The Classic category is the most popular and best-selling category.
- The Classic Deluxe pizza is the top-selling item.
- Large size pizzas have the highest demand and popularity.
- The Brie Carre Pizza is the least selling pizza.
- The pizza shop is usually busy during lunch and dinner hours.
- The number of orders is high on Fridays.
- The Big Meat Pizza is the most frequently ordered item.
- The most busy hours are from 12 pm to 2 pm in the lunch time.
- Large size pizzas are the most commonly ordered.
- Classic category pizzas are the preferred choice.

CHAPTER 5

5. CONCLUSION

5.1 OVERALL SUMMARY

5.1.1 Overall Sales Performance

During the analyzed period of 2015, the pizza sales performance demonstrated strong results. The total revenue generated amounted to $817,860, with a significant number of pizzas sold, reaching 49,574. This indicates a healthy demand for pizzas during this period.

5.1.2 Size and Category Preferences

Customers displayed a strong preference for large-sized pizzas, indicating a preference for sharing or group occasions. The Classic category emerged as the most preferred among customers, solidifying its popularity and demonstrating its ability to cater to a wide customer base.

5.1.3 Best-selling Items

The Classic Deluxe pizza stood out as the best-selling item, consistently outperforming other menu options. This pizza's unique flavor and combination of ingredients resonated well with customers, contributing to its popularity.

5.1.4 Customer Demand and Order Patterns

The demand for large-sized pizzas highlights customer preferences for larger portions, potentially for social gatherings or shared meals. The Brie Carre Pizza had the least sales, indicating a lower appeal among customers. Lunch and dinner hours were found to be the busiest periods, with higher customer activity during these times.

5.1.5 Weekly and Daily Trends

Fridays experienced a significant surge in the number of orders, suggesting increased sales activity at the end of the workweek.

5.2 SURPRISING (NOTABLE) CONCLUSIONS

- **Limited Orders at Opening Time**: Despite opening at 9 AM, the number of orders during this early period is almost non-existent. This could be attributed to breakfast being a less popular time for pizza consumption, with customers typically opting for other food choices during this part of the day.

- **Lunchtime Rush**: The most bustling period for the pizza shop is between 12 PM and 2 PM, indicating a significant influx of customers during lunch hours. This can be attributed to people taking their lunch breaks from work or school and seeking a quick and convenient meal option.

- **Sunday as the Least Busy Weekday**: Sundays exhibit lower customer activity compared to other weekdays, suggesting that the shop's main customer base consists primarily of working individuals. Many people may prefer to dine out or order pizza during the weekdays due to their busy schedules.

- **Seasonal Sales Patterns**: Spring season experiences higher sales compared to other seasons, while sales during the fall season are relatively lower. This could be influenced by various factors such as weather, seasonal events, or cultural preferences during different times of the year.

- **Preference for Large Size**: The most preferred pizza size by customers is large. This indicates a tendency for customers to order pizzas in groups, potentially for sharing among family, friends, or colleagues.

- **Single Category for Each Pizza**: Each pizza is classified into only one category out of the four available. This helps maintain clarity in categorization and ensures that customers can easily identify the type of pizza they desire.

- **The Big Meat Pizza**: This pizza stands out as the most ordered item, indicating a high demand for its combination of meat toppings. Despite not belonging to the highest revenue-generating category, its popularity suggests a strong preference for this specific flavor profile.

- **The Thai Chicken Pizza**: Although the Thai Chicken Pizza generates the highest revenue per unit price, it falls under the least available category. This indicates that there is a niche market or specific customer segment that highly values this unique pizza flavor, leading to a higher willingness to pay for it.

- **Classic Category Dominance**: The Classic category emerges as the most ordered, highest revenue-generating, and most available pizza category. This popularity could be due to the timeless appeal and widespread recognition of classic pizza flavors, making them a go-to choice for customers.

- **Average Pizza Per Order**: With an average of 2.32 pizzas per order, it is evident that customers usually come in groups when ordering. This suggests that the pizza shop attracts customers looking to share a meal or cater to gatherings or social events.

5.3 FINAL WORDS AND SUGGESTIONS

The analysis of the pizza sales data has provided valuable insights into customer preferences, order patterns, and revenue generation. The findings indicate that the pizza shop attracts a diverse customer base, including working individuals and school students. The popularity of large-sized pizzas suggests that customers often come in groups or seek convenient meal options for gatherings. The Classic category stands out as the preferred choice among customers, showcasing the timeless appeal of classic pizza flavors. Additionally,The Thai Chicken Pizza and the Big Meat Pizza, featuring the popular chicken topping, have emerged as top sellers, showcasing the importance of incorporating well-balanced flavor profiles that include chicken.

Considering the analysis results, the pizza shop can consider the following suggestions to further improve its business:

- **Menu Optimization**: Based on the popularity of large-sized pizzas and the Classic category, the pizza shop can focus on expanding its offerings in these areas.
- **Marketing Strategies:** Leveraging the insights gained from peak order times, such as the lunchtime and dinnertime rush, the pizza shop can implement targeted marketing campaigns to attract customers during these busy periods.
- **Menu Recommendations**: Based on the analysis, highlighting the best-selling and unique pizzas like the Classic Deluxe and the Thai Chicken Pizza can attract customer attention. Displaying these recommendations prominently on menus and promotional materials can increase their visibility and encourage customers to try these popular options.