DATA ANALYSIS ON SWIGGY

Efforts by:-
TEAM: **Pandas Of Python**

Appoorva S. Khajuria	23scse1180326
Ojasvi Bakshi	23scse1010260
Bhumika Sahai	23scse1010766
Nimisha Shukla	23scse1011192

Agenda

- Introduction
- Dataset Overview
- Data Cleaning & Preparation
- Exploratory Data Analysis (EDA)
- Price vs Ratings
- Delivery Time Insights & Sales Analysis
- Geographical Analysis
- Key Findings
- Conclusion

INTRODUCTION

Overview of Swiggy and its services
Purpose of the data analysis

Swiggy is one of India's leading food delivery platforms, connecting millions of customers with their favorite local restaurants. With the rise of online food ordering, data-driven insights have become crucial in understanding customer preferences, delivery dynamics, and restaurant performance.

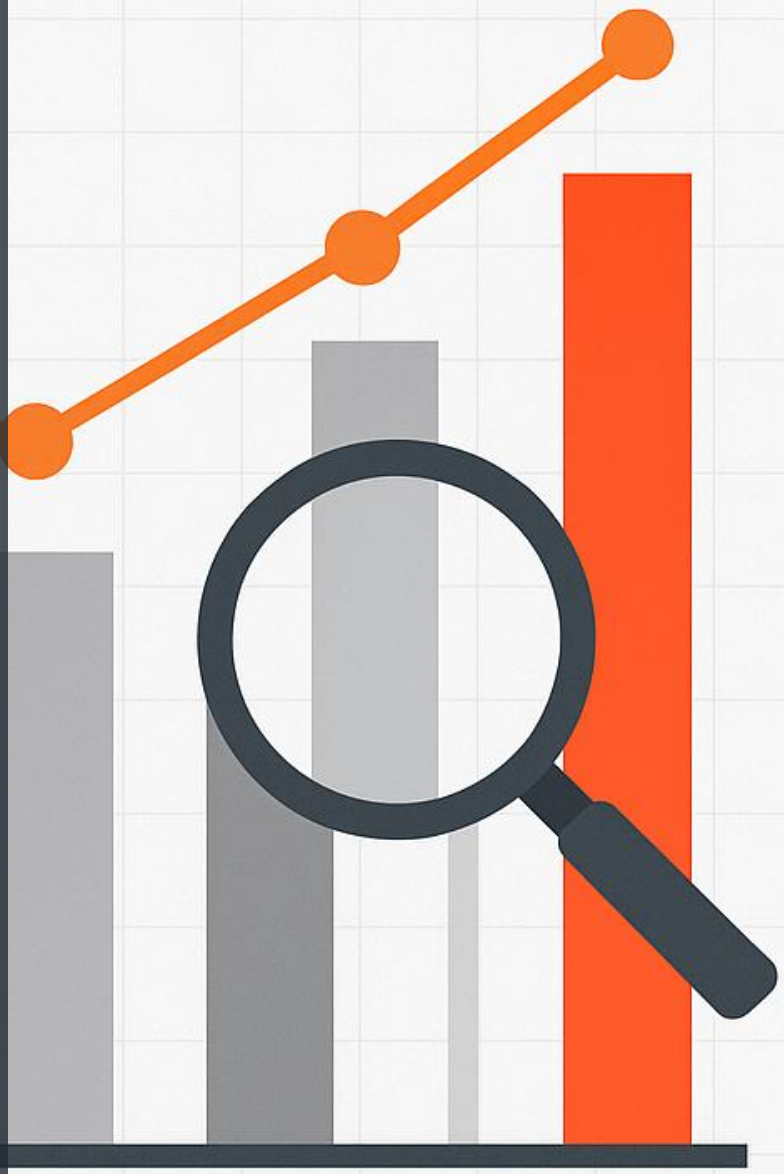
In this analysis, we explore a dataset of over 8,600 food delivery records to uncover patterns, trends, and actionable insights that can help improve service quality, customer satisfaction, and business strategy.

DATA OVERVIEW

- Dataset Size: 8,680 rows × 10 columns
- Source: Swiggy food delivery data (simulated or collected from listings)

Key Features:

- ID: Unique identifier for each entry
- Area & City: Geographic location of the restaurant
- Restaurant: Name of the restaurant
- Price: Average price per order
- Avg Ratings: Customer rating (out of 5)
- Total Ratings: Number of ratings received
- Food Type: Types of cuisines served
- Address: Local address or landmark
- Delivery Time: Estimated time in minutes



DATA OVERVIEW

- ## INSTALL OPENDATASETS

```
pip install opendatasets
```

```
Requirement already satisfied: opendatasets in /usr/local/lib/python3.11/dist-packages (0.1.22)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from opendatasets) (4.67.1)
Requirement already satisfied: kaggle in /usr/local/lib/python3.11/dist-packages (from opendatasets) (1.7.4.5)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from opendatasets) (8.2.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (6.2.0)
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2025.4.26)
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (3.4.2)
Requirement already satisfied: idna in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (3.10)
Requirement already satisfied: protobuf in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (5.29.4)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2.9.0.post0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2.32.3)
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (75.2.0)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (1.17.0)
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (1.3)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2.4.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (0.5.1)
```

- ## IMPORT OPENDATASETS

```
import opendatasets as od
od.download("https://www.kaggle.com/datasets/abhijitdahatonde/swiggy-restuarant-dataset")
```

```
Skipping, found downloaded files in "./swiggy-restuarant-dataset" (use force=True to force download)
```

- ## DATA PREVIEW

```
import pandas as pd
import numpy as np
df = pd.read_csv("/content/swiggy-restuarant-dataset/swiggy.csv")
df.head()
```

	ID	Area	City	Restaurant	Price	Avg ratings	Total ratings	Food type	Address	Delivery time
0	211	Koramangala	Bangalore	Tandoor Hut	300.0	4.4	100	Biryani,Chinese,North Indian,South Indian	5Th Block	59
1	221	Koramangala	Bangalore	Tunday Kababi	300.0	4.1	100	Mughlai,Lucknowi	5Th Block	56
2	246	Jogupalya	Bangalore	Kim Lee	650.0	4.4	100	Chinese	Double Road	50
3	248	Indiranagar	Bangalore	New Punjabi Hotel	250.0	3.9	500	North Indian,Punjabi,Tandoor,Chinese	80 Feet Road	57
4	249	Indiranagar	Bangalore	Nh8	350.0	4.0	50	Rajasthani,Gujarati,North Indian,Snacks,Desser...	80 Feet Road	63

```
print(df.columns)
```

```
Index(['ID', 'Area', 'City', 'Restaurant', 'Price', 'Avg ratings',
       'Total ratings', 'Food type', 'Address', 'Delivery time'],
      dtype='object')
```

```
print("Shape of dataset:", df.shape)
print("\nColumn names:\n", df.columns)
print("\nFirst 5 rows:\n", df.head())
```

```
Shape of dataset: (8680, 10)

Column names:
 Index(['ID', 'Area', 'City', 'Restaurant', 'Price', 'Avg ratings',
       'Total ratings', 'Food type', 'Address', 'Delivery time'],
      dtype='object')

First 5 rows:
     ID         Area       City         Restaurant  Price  Avg ratings  \
0  211  Koramangala  Bangalore        Tandoor Hut  300.0          4.4
1  221  Koramangala  Bangalore      Tunday Kababi  300.0          4.1
2  246    Jogupalya  Bangalore            Kim Lee  650.0          4.4
3  248   Indiranagar  Bangalore  New Punjabi Hotel  250.0          3.9
4  249   Indiranagar  Bangalore                Nh8  350.0          4.0

   Total ratings                                  Food type  \
0            100         Biryani,Chinese,North Indian,South Indian
1            100                                Mughlai,Lucknowi
2            100                                         Chinese
3            500             North Indian,Punjabi,Tandoor,Chinese
4             50  Rajasthani,Gujarati,North Indian,Snacks,Desser...

        Address  Delivery time
0     5Th Block             59
1     5Th Block             56
2   Double Road             50
3  80 Feet Road             57
4  80 Feet Road             63
```

SWIGGY

DATA CLEANING & PREPARATION

Effective data analysis begins with clean, reliable data. Here's how we prepared the dataset:

DATA CLEANING & PREPARATION

Steps Taken:

- **Missing Values**:
  - Checked for nulls across all columns
  - No significant missing data detected
- **Data Types:**
  - Converted Price, Avg Ratings, and Delivery Time to numerical types
  - Ensured categorical data (e.g., Area, City, Food Type) were properly formatted
- **Duplicates:**
  - Removed any duplicate restaurant entries, if found
- **Outliers:**
  - Examined delivery time and price for unusually high or low values
  - Outliers retained only if contextually valid (e.g., high-end restaurants)
- **Standardization:**
  - Stripped extra whitespace from text fields
  - Unified food type strings for better grouping

DATA CLEANING

```
[ ] # Check how many duplicate rows exist
    duplicates = df.duplicated().sum()
    print(f"Number of duplicate rows: {duplicates}")

    # Drop duplicates if any
    df = df.drop_duplicates()

    # Confirm the new shape
    print("New shape after removing duplicates:", df.shape)
```

```
Number of duplicate rows: 0
New shape after removing duplicates: (8680, 10)
```

```
[ ] # Check for missing/null values
    print("Missing values in each column:")
    print(df.isnull().sum())
```

```
Missing values in each column:
ID              0
Area            0
City            0
Restaurant      0
Price           0
Avg ratings     0
Total ratings   0
Food type       0
Address         0
Delivery time   0
dtype: int64
```

```
# Check again to see missing values
print("Missing values before cleaning:")
print(df.isnull().sum())

# Drop columns with >50% missing values (optional step, based on threshold)
threshold = 0.5 * len(df)
df = df.dropna(thresh=threshold, axis=1)

# Fill missing values
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = df[col].fillna(df[col].mode()[0])  # Most frequent value
    else:
        df[col] = df[col].fillna(df[col].median())  # Median for numeric

print("Missing values after cleaning:")
print(df.isnull().sum())
```

```
Missing values before cleaning:
ID              0
Area            0
City            0
Restaurant      0
Price           0
Avg ratings     0
Total ratings   0
Food type       0
Address         0
Delivery time   0
dtype: int64
Missing values after cleaning:
ID              0
Area            0
City            0
Restaurant      0
Price           0
Avg ratings     0
Total ratings   0
Food type       0
Address         0
Delivery time   0
dtype: int64
```

Exploratory Data Analysis (EDA)

EDA helps us uncover hidden patterns, trends, and relationships within the data.

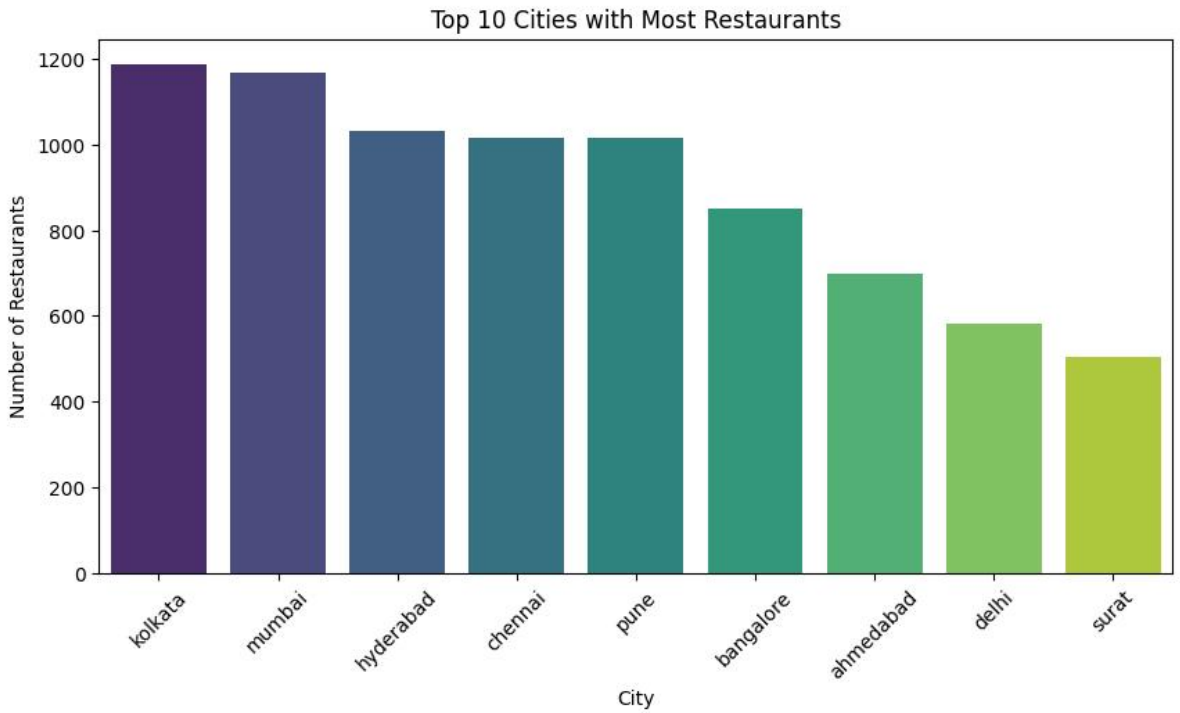
Key Insights Explored:

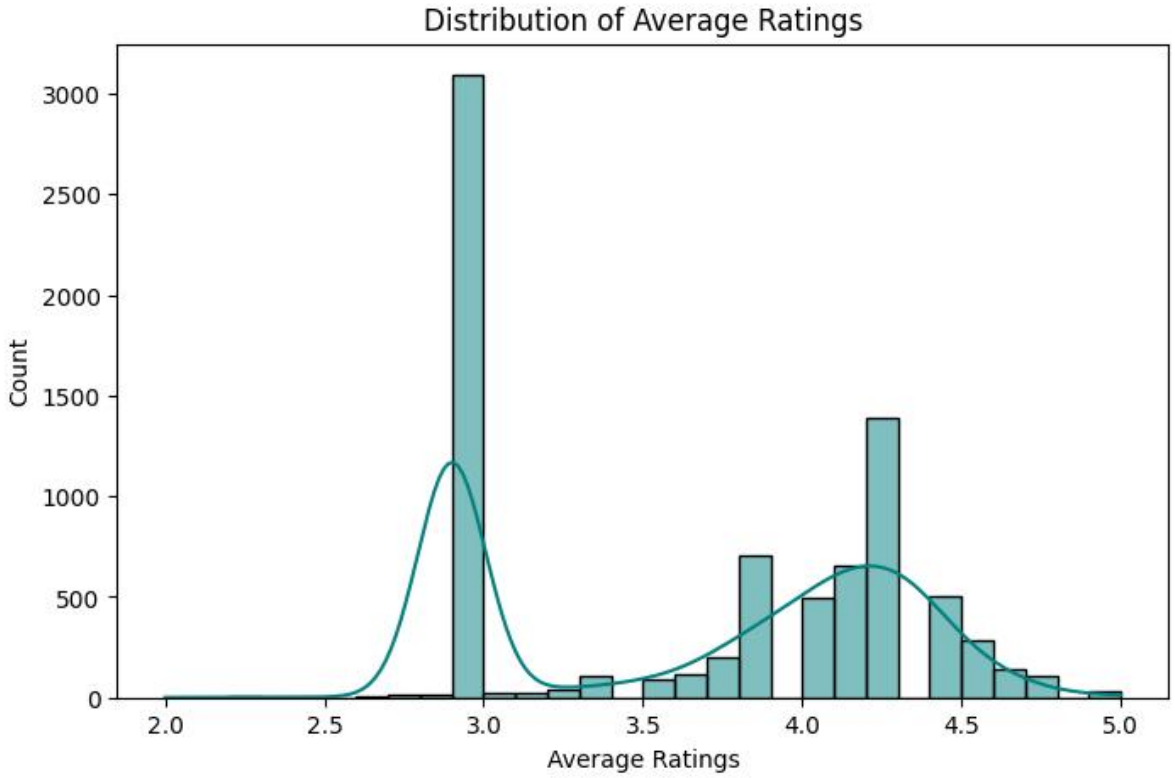
Area & City Distribution

- **Price Distribution**
- **Average Ratings**
- **Total Ratings**
- **Delivery Time**

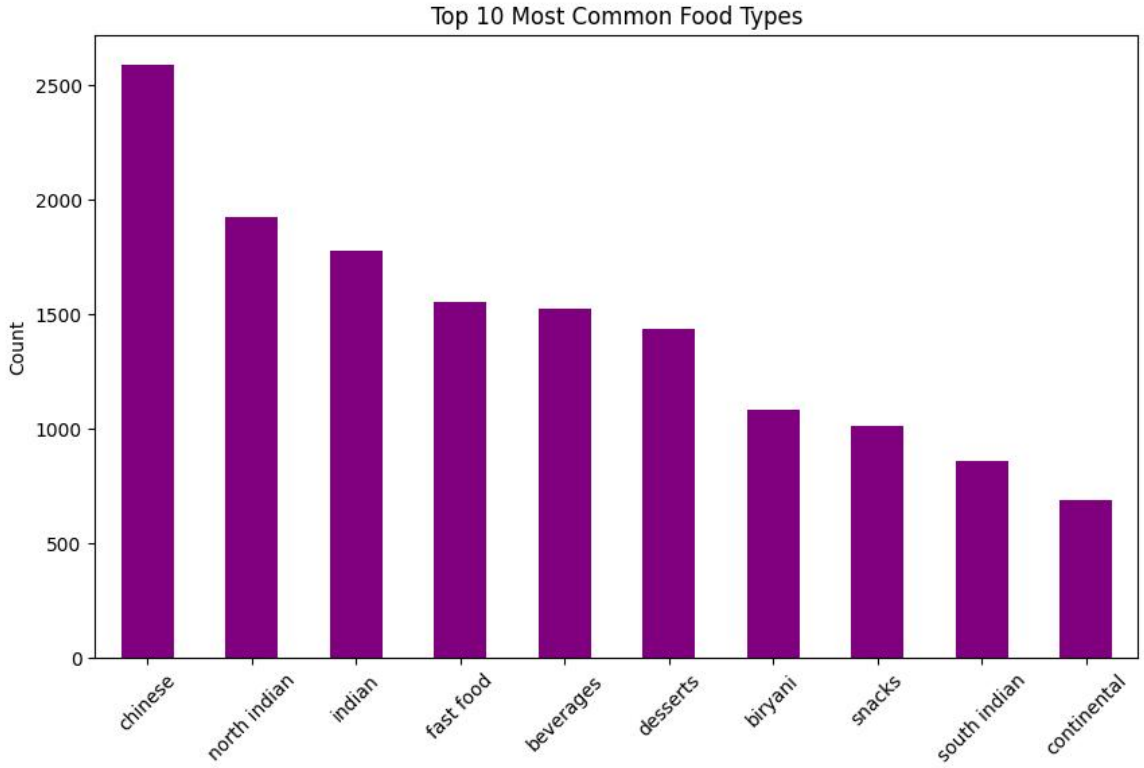
Food Type Frequency

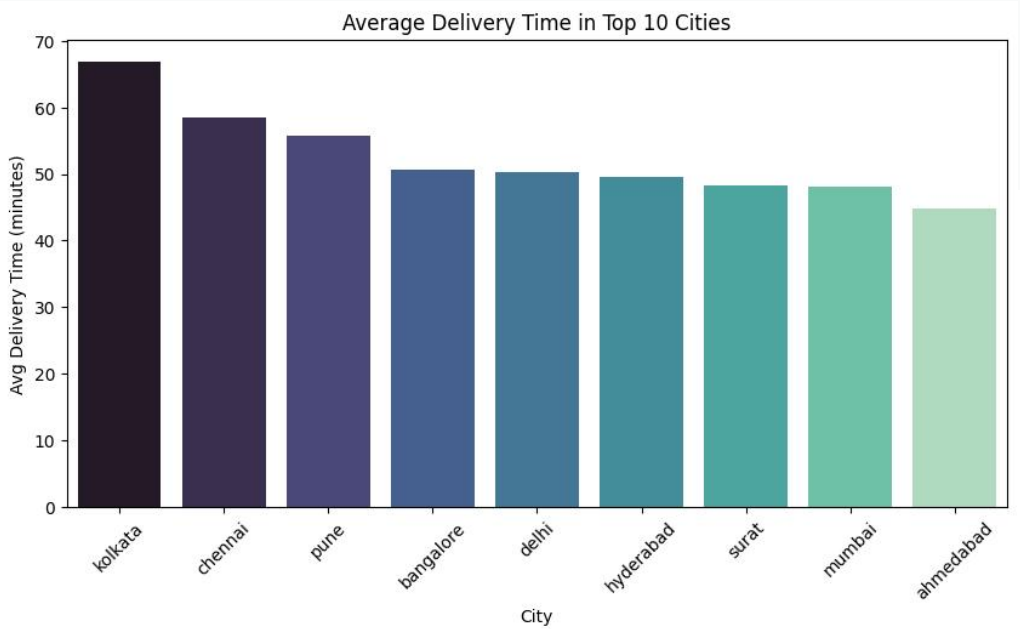
Exploratory Data Analysis (EDA)

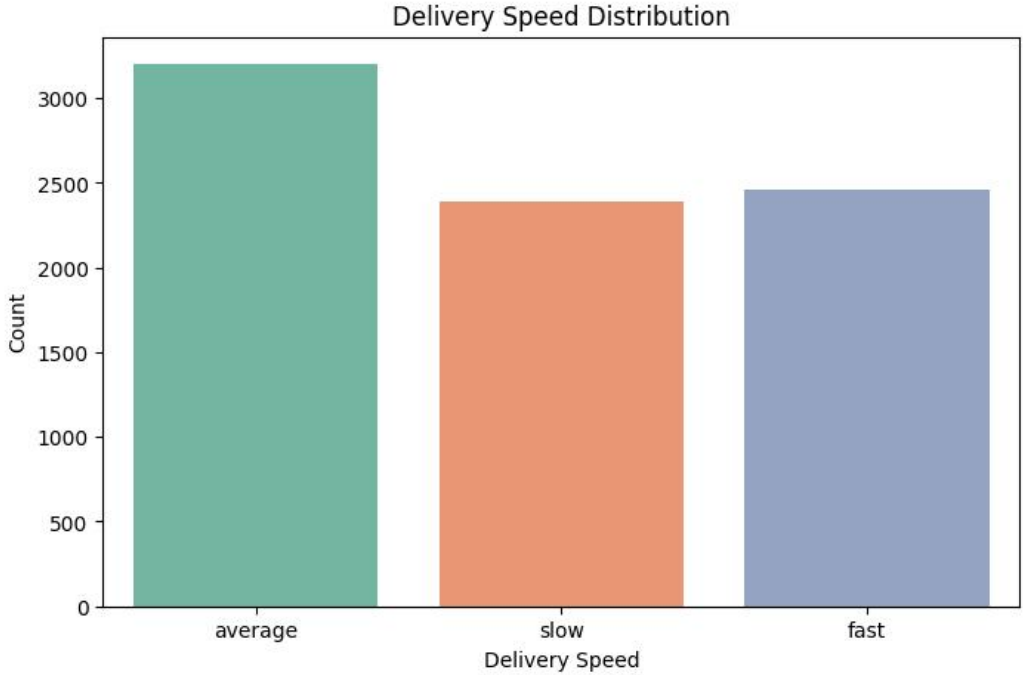




Exploratory Data Analysis (EDA)







POWER BI DASHBOARD

POWER BI DASHBOARD

SWIGGY

OVERVIEW

Amount | Quantity

149K
ORDERS COUNT

725M
TOP_10

100K
USER COUNT

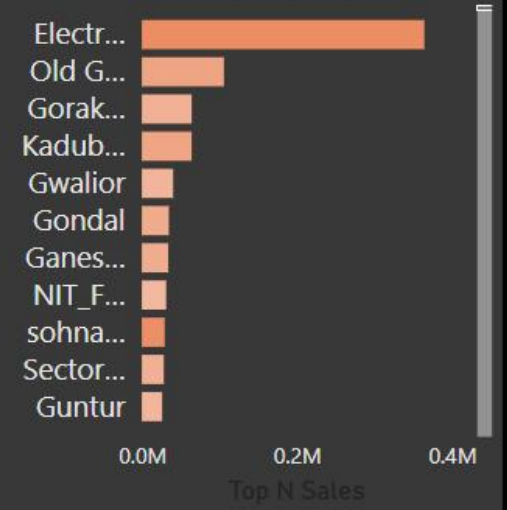
VEG
156K

NON-VEG
140K

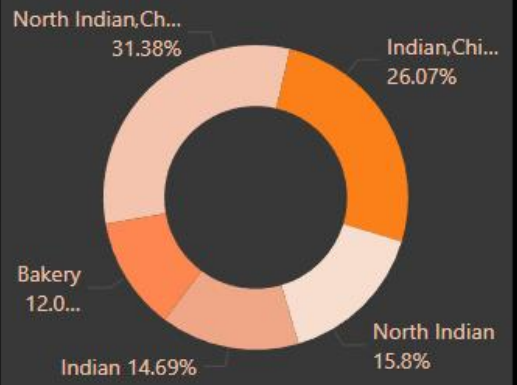
BOTH
14K

Default
Top 10
Top 100
Top 20
Top 30
Top 5

TOP CITIES



TOP SELLING CUISINES



SWIGGY

DASHBOARD

YEAR

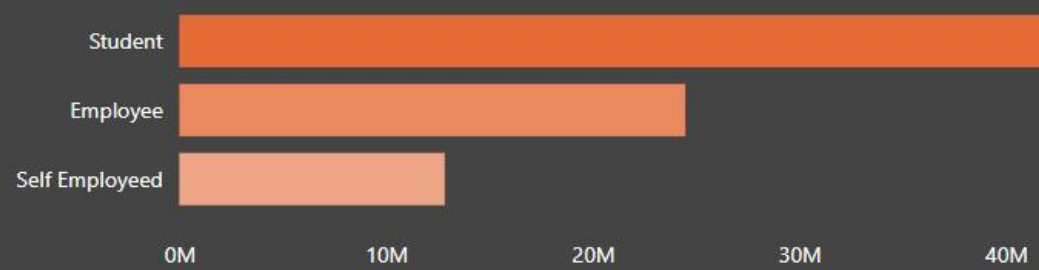
2017 | 2018 | 2019 | 2020

USER PERFORMANCE

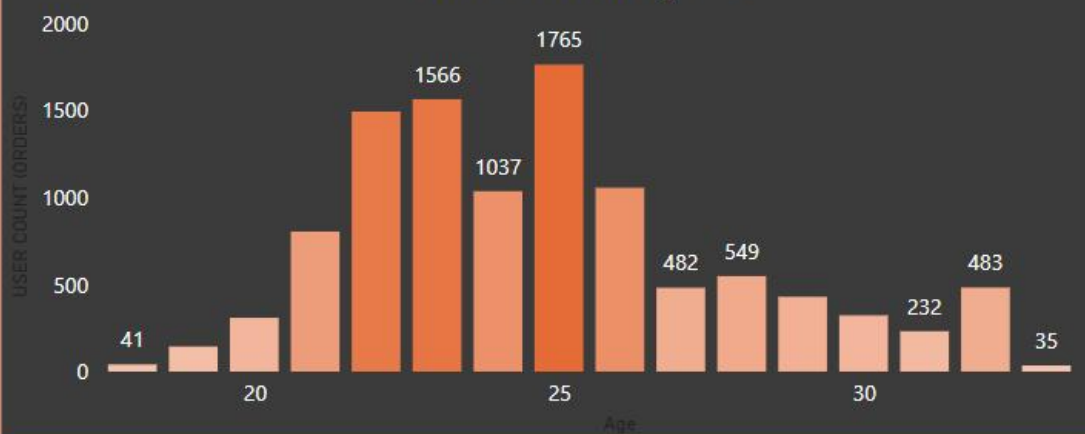
467M	358M
CY_SALES	PY_SALES
408M	11K
TOP_10	USER COUNT

Female | Male

SALES VS OCCUPATION



USER COUNT by AGE



OVERVIEW | USER PERFORMANCE | CITY ANALYSIS | +

DASHBOARD

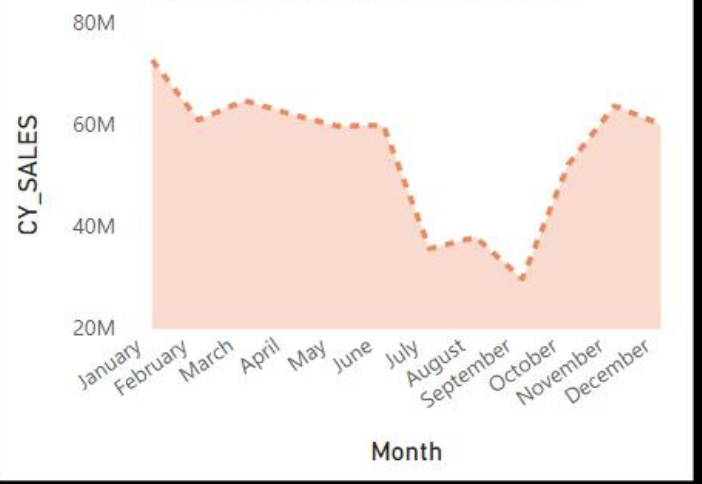
SWIGGY

CITY PERFORMANCE

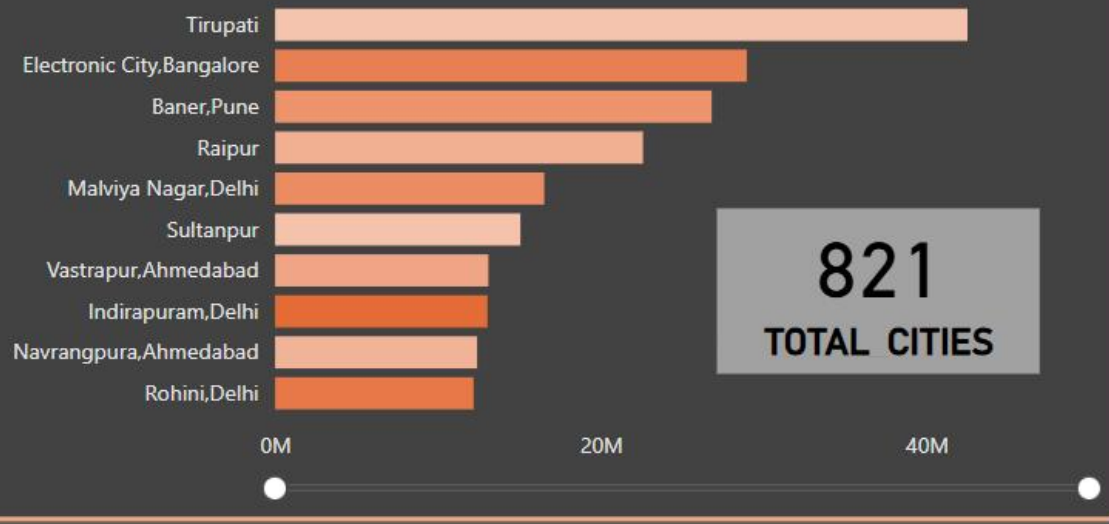
CITY

All

CURRENT YEAR SALES BY MONTH

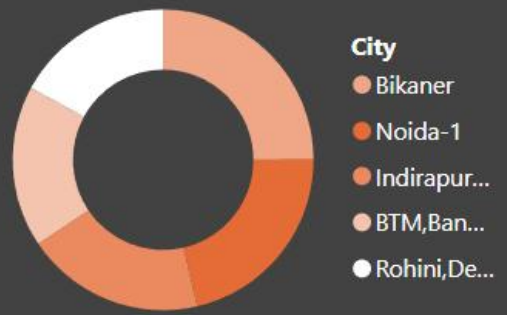


SALES BY CITY

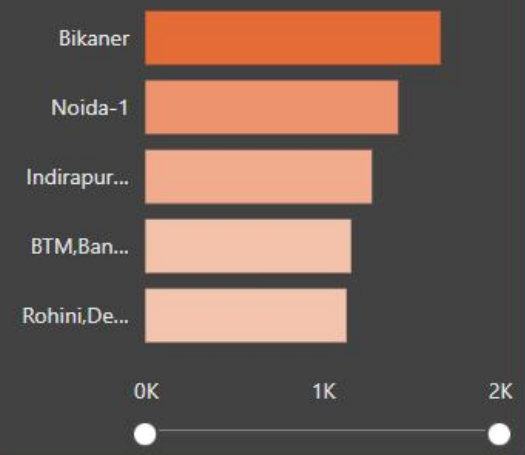


821
TOTAL CITIES

CITIES WITH MAX USERS



TOP RATINGS



Prices vs Sales $

Insights:

No Direct Correlation:
High-priced restaurants don't always have the highest ratings.

Mid-Range Winners:
Restaurants priced between **₹250–₹450** tend to receive the best ratings (4.0+), suggesting a balance of value and quality.

Low Price ≠ Low Rating:
Several budget-friendly places also maintain ratings above **4.2**, indicating great service and food at affordable prices.

Customer Perception:
Users value taste, consistency, and service more than premium pricing.

Delivery Time Insights & Sales Analysis

Delivery Time Insights

Average Delivery Time: Around **55–60 minutes**

Fastest Areas:

- **Koramangala** and **Jogupalya** show faster deliveries (under 55 mins)

Slower Zones:

- Delays observed in **Indiranagar**, especially during peak hours

Ratings vs Delivery:

- Longer delivery times often correlate with slightly lower ratings.

Geographical Analysis

Analyzing restaurant distribution and performance by location.

City & Area Focus
City: All data points are from **Bangalore**

Top Areas:

- **Koramangala** – Highest restaurant density, fastest delivery
- **Indiranagar** – Diverse cuisine, slightly higher prices
- **Jogupalya** – Fewer options, but high-rated listings

Insights by Area

- **Koramangala**:
  - Best mix of affordable food, fast delivery, and high ratings
- **Indiranagar**:
  - Offers premium-priced restaurants with slightly slower delivery
- **Rating Trends**:
  - Higher density areas show more variation in ratings and delivery

Possible Map Visuals:

•Heatmap of restaurant count by area
•Bubble map showing Avg Rating vs Price per location

Key Findings ✓

Impact factor	Measurement
Peak Ordering Hours Identified:	Highest orders occur between 7PM – 10PM
High Demand Zones Mapped:	Tier-1 city neighborhoods and college areas show consistently high order density.
Customer Preferences Detected:	Fast food and North Indian cuisine are top categories.
Delivery Time Optimization Needed:	Delays are common in rainy weather and during flash sales, especially in metro cities.
Loyalty Indicators:	Repeat customers are more likely to tip and leave positive feedback
Data Quality Improvements Noted:	~6% of raw data had missing delivery timestamps or inconsistent location entries i.e, need for real-time validation.

CONCLUSION

Data-Driven Decisions for Smarter Operations

- **Clean, structured data** is the foundation of Swiggy's analytics journey, ensuring trust and accuracy.
- **EDA uncovered valuable insights** into customer behavior, delivery patterns, and restaurant performance.
- **Key findings** like peak order times, demand zones, and loyalty trends can guide operational strategies.
- Identifying **data quality gaps and anomalies** helps improve real-time systems and predictive models.
- These insights empower Swiggy to deliver a more **personalized, efficient, and seamless customer experience.**