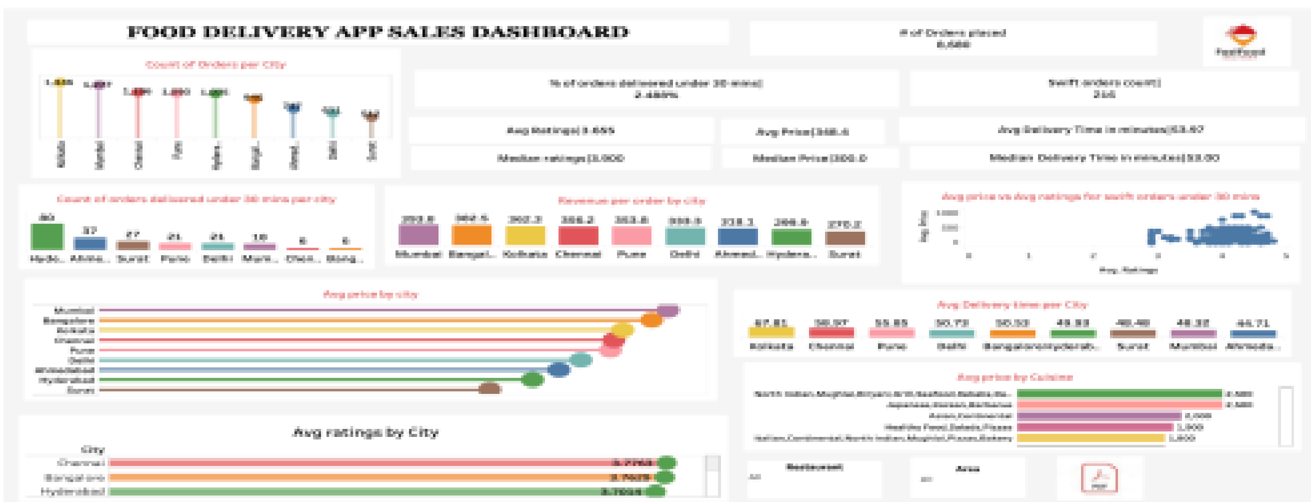
--== Food Delivery app sales dashboard and analysis(A business project)==--

Purpose: Analyzed Food Delivery app sales and I used Tableau to visualize the insights and implement a user friendly dashboard.

Snapshot of the dashboard:



Note: This dashboard is created and tailored to the Indian food delivery market, capturing region-specific trends and consumer behaviors. [Link for the entire dashboard.]()

Insights and Recommendations:

Order count insights:

- Kolkata,Mumbai and Chennai are metropolitan cities and have got the highest number of orders.
- Surat being an upcoming city has got the lowest count of orders.

Delivery time insights:

- The avg delivery time and the median delivery time of all the orders is 53.97 mins and 53.00 respectively. In our case, both mean and the median almost coincide.
- **The avg delivery time per city reveals interesting insights.**
- Kolkata has the highest number of avg delivery time whereas Mumbai has the lowest. ● Studies reveal that Mumbai is generally considered to have better traffic management compared to Kolkata, due to its more developed infrastructure, extensive public transportation network, and better traffic discipline.

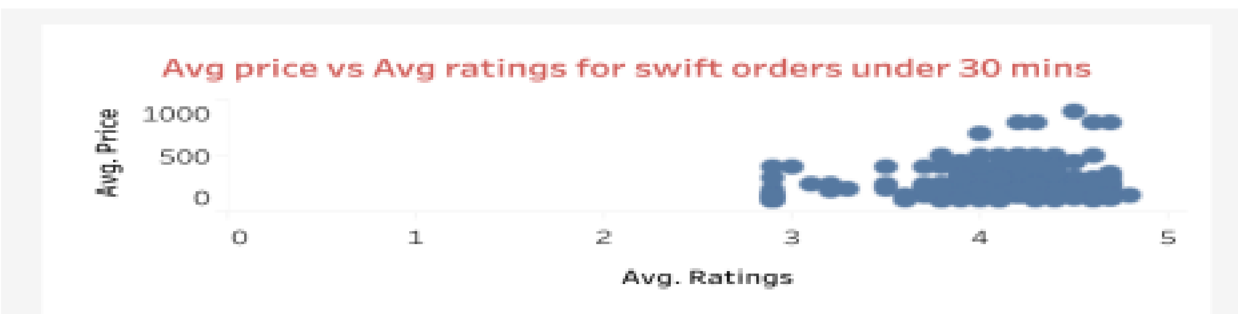
- Only 2.5% of orders have been delivered in less than or equal to 30 mins which can still be improved. **These orders are known as "Swift Orders"**
- When considering the swift orders, Hyderabad has the highest count of swift orders and Bangalore has the lowest.
- Research reveals that while both cities face traffic issues due to rapid growth, Hyderabad's better infrastructure, recent improvements, and relatively more effective traffic management give it an edge over Bangalore, which struggles with severe congestion and infrastructure challenges.
- Better traffic management is recommended for cities with poor road infrastructure.

Revenue/Price insights:

- The avg price per order for Mumbai is the highest and is the lowest for Surat. ● This is attributed to the fact that Mumbai is more developed and has the highest standard of living in India than Surat.
- Other cities like Bangalore,Kolkata and Chennai which have the second,third and fourth highest avg price per order are also more developed with higher cost of living than the rest of the cities in India.

Analysis on swift orders:

Scatter plot between price and ratings for swift orders



- **If we observe the graph, most data points are concentrated between 3.0 and 4.5 in ratings, indicating that orders delivered within 30 minutes generally receive moderate to high ratings indicating that orders delivered within 30 minutes are rarely rated poorly. This might suggest that customers may be willing to pay more for quick deliveries and rate their experience highly.**
- **Premium Orders**: The presence of higher-priced outliers with good ratings might indicate that premium products or services that offer quick delivery are well-received by customers despite their higher price.
- **Low-Price Orders**: Lower-priced orders are spread across a range of ratings, suggesting that price alone does not determine customer satisfaction when delivery time is within 30 minutes. Company can prioritize reducing the delivery time to increase the customer ratings.

NPS Score calculation analysis on customer ratings:

- In evaluating customer satisfaction, a commonly used metric is the Net Promoter Score (NPS).
- This score provides insights into the loyalty and satisfaction of customers based on their likelihood to recommend a product or service. By analyzing the ratings given by customers, we can categorize their responses to better understand their sentiments.
- **I used Python to calculate the NPS score for ratings given by the customers. I took the median ratings to find a midpoint instead of mean since median is unaffected by outliers.**

```
[4]: import pandas as pd

# Assuming 'data' is your DataFrame and 'score' is the column with the NPS scores
data = pd.read_csv('ratings.csv')  # Replace with your actual file or data loading method

# Categorize responses
data['category'] = data['Ratings'].apply(lambda x: 'Promoter' if x >= 3.9 else ('Detractor' if x <= 2 else 'Passive'))

# Calculate the counts of each category
promoters_count = data[data['category'] == 'Promoter'].shape[0]
detractors_count = data[data['category'] == 'Detractor'].shape[0]
total_responses = data.shape[0]

# Calculate percentages
promoters_percentage = (promoters_count / total_responses) * 100
detractors_percentage = (detractors_count / total_responses) * 100

# Calculate NPS
nps = promoters_percentage - detractors_percentage

print(f"NPS Score: {nps:.2f}")
print(f"Promoter%: {promoters_percentage:.2f}")
print(f"Detractor%: {detractors_percentage:.2f}")

NPS Score: 50.98
Promoter%: 50.99
Detractor%: 0.01
```

- The median rating in the dataset is 3.9.
- I categorized the ratings into three groups: **Detractors** (ratings ≤ 2), **Passives** (ratings ≥ 3.9), and **Promoters** (ratings > 4). The Net Promoter Score (NPS) is calculated using the formula: **NPS = % of Promoters - % of Detractors**.
- Based on this calculation, the NPS score is 50.98, which is generally seen as a strong positive indicator of customer satisfaction and loyalty.
- But it is also advised to check with the industry standards for better interpretation to drive further business actions