

In [4]:

Out [4]:



## Uber Supply Demand Gap Analysis For Bengaluru,India

### Data of year 2016

Demonstrated by **Biswarup Das**

**Dataset Link:** <https://www.kaggle.com/datasets/anupammajhi/uber-request-data?resource=download>  
(<https://www.kaggle.com/datasets/anupammajhi/uber-request-data?resource=download>)

### Major issues impacting Uber's business are:

- Cancellation of rides going towards the airport were higher than regular trips
- Trips to and from an airport resulted in high consumption of fuel and time. Hence, a trip back to the city without a rider is not economically beneficial for the driver.
- Due to high variance in flight arrivals (higher during evening, late night hours) the driver idle time is higher in morning. As a result, no cars are available during peak ight hours because working hours ends for majority drivers at night.

**Exploratory Data Analysis (EDA)** is done on the raw dataset from Kaggle then to draw some useful insights I used Power BI in later analysis .\*\*

As a part of EDA I have followed below steps:

- **Data Extraction :** Load the raw dataset and inspect different features.
- **Data Cleaning :** Correcting date time formats & filling in missing values.
- **Feature Engineering:** Determine new features.

## Data Description:

- **Request id:** A unique identifier of the request.
- **Pickup point:** The point from which the request was made.
- **Driver id:** The unique identification number of the driver.
- **Status:** The final status of the trip, that can be either completed, cancelled by the driver or no cars available.
- **Request timestamp:** The date and time at which the customer made the trip request.
- **Drop timestamp:** The drop-off date and time, in case the trip was completed.

## Data Extraction

### Importing usefull libraries

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
uber = pd.read_csv("Uber Request Data.csv")
uber.head(5)
```

Out[2]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp
0	619	Airport	1.0	Trip Completed	11/7/2016 11:51	11/7/2016 13:00
1	867	Airport	1.0	Trip Completed	11/7/2016 17:57	11/7/2016 18:47
2	1807	City	1.0	Trip Completed	12/7/2016 9:17	12/7/2016 9:58
3	2532	Airport	1.0	Trip Completed	12/7/2016 21:08	12/7/2016 22:03
4	3112	City	1.0	Trip Completed	13-07-2016 08:33:16	13-07-2016 09:25:47

In [3]:

```
uber.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6745 entries, 0 to 6744
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Request id            6745 non-null   int64
1   Pickup point          6745 non-null   object
2   Driver id             4095 non-null   float64
3   Status                6745 non-null   object
4   Request timestamp     6745 non-null   object
5   Drop timestamp        2831 non-null   object
dtypes: float64(1), int64(1), object(4)
memory usage: 316.3+ KB
```

I can see that two coluns, i.e. **Request timestamp** & **Drop timestamp** are not consistent datetime format

## Data Cleaning

In [4]:

```
sum(uber.duplicated(subset="Request id"))==0 # checking for duplicate rows
```

Out[4]:

True

In [5]:

```
uber.shape # checking dimension of the dataframe
```

Out[5]:

(6745, 6)

In [6]:

```
uber.isnull().sum() # checking null values
```

Out[6]:

```
Request id          0
Pickup point        0
Driver id          2650
Status              0
Request timestamp   0
Drop timestamp     3914
dtype: int64
```

In [9]:

```
# Calculating null value percentage for each column
pd.DataFrame(round(100*(uber.isnull().sum()/len(uber.index)),2))
```

Out[9]:

	0
<b>Request id</b>	0.00
<b>Pickup point</b>	0.00
<b>Driver id</b>	39.29
<b>Status</b>	0.00
<b>Request timestamp</b>	0.00
<b>Drop timestamp</b>	58.03

I can see that 'Driver id' & 'Drop timestamp' have considerable null values, these entries are probably the rides where trip were never assigned to a driver & it was not completed (status- no car available).

In [10]:

```
pd.DataFrame(uber.isnull().sum(axis=1)) # row wise null value count
```

Out[10]:

```

      0
0  0
1  0
2  0
3  0
4  0
... ..
6740 2
6741 2
6742 2
6743 2
6744 2

```

6745 rows × 1 columns

### Converting format of 'Request timestamp' & 'Drop timestamp' to datetime format

In [12]:

```
uber['Request timestamp'] = pd.to_datetime(uber['Request timestamp'])
uber['Drop timestamp'] = pd.to_datetime(uber['Drop timestamp'])
```

In [14]:

```
uber['Request timestamp'].max()
```

Out[14]:

```
Timestamp('2016-12-07 23:54:00')
```

In [16]:

```
uber.info() # rechecking Dtypes
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6745 entries, 0 to 6744
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Request id            6745 non-null   int64
 1   Pickup point          6745 non-null   object
 2   Driver id             4095 non-null   float64
 3   Status                6745 non-null   object
 4   Request timestamp     6745 non-null   datetime64[ns]
 5   Drop timestamp        2831 non-null   datetime64[ns]
dtypes: datetime64[ns](2), float64(1), int64(1), object(2)
memory usage: 316.3+ KB

```

# Feature Engineering

Adding new columns, 'Request Hours' & 'Drop Hours' to the dataset from column 'Request timestamp' & 'Drop timestamp' resp.

In [17]:

```
uber['Request Hours'] = uber['Request timestamp'].apply(lambda x:x.hour)
uber['Drop Hours'] = uber['Drop timestamp'].apply(lambda x:x.hour)
```

In [18]:

```
uber.head(5)
```

Out[18]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp	Request Hours	Drop Hours
0	619	Airport	1.0	Trip Completed	2016-11-07 11:51:00	2016-11-07 13:00:00	11	13.0
1	867	Airport	1.0	Trip Completed	2016-11-07 17:57:00	2016-11-07 18:47:00	17	18.0
2	1807	City	1.0	Trip Completed	2016-12-07 09:17:00	2016-12-07 09:58:00	9	9.0
3	2532	Airport	1.0	Trip Completed	2016-12-07 21:08:00	2016-12-07 22:03:00	21	22.0
4	3112	City	1.0	Trip Completed	2016-07-13 08:33:16	2016-07-13 09:25:47	8	9.0

Adding new column 'Request Time Slot' to each request according to the time range defined in a function below:

- [0hours-8hours] - Early Morning Hours
- [8hours-12hours] - Peak Morning Hours
- [12hours-17hours]- Noon Hours
- [17hours-21hours]- Evening Hours
- [21hours<] - Night Hours

In [21]:

```
def determine_time_slot(x):
    if (x>=0 and x<8):
        return "Early morning hours" #12am - 7am
    elif (x>=8 and x<12):
        return "Peak morning hours" #8am - 11am
    elif (x>=12 and x<17):
        return "Noon hours" #12pm - 4pm
    elif (x>=17 and x<21):
        return "Evening hours" #5pm - 8pm
    elif (x>=21):
        return "Night hours" #9pm onwards
uber['Request Time Slot'] = uber['Request Hours'].apply(determine_time_slot)
```

In [22]:

```
uber.head(5)
```

Out[22]:

	Request id	Pickup point	Driver id	Status	Request timestamp	Drop timestamp	Request Hours	Drop Hours	Request Time Slot
0	619	Airport	1.0	Trip Completed	2016-11-07 11:51:00	2016-11-07 13:00:00	11	13.0	Peak morning hours
1	867	Airport	1.0	Trip Completed	2016-11-07 17:57:00	2016-11-07 18:47:00	17	18.0	Evening hours
2	1807	City	1.0	Trip Completed	2016-12-07 09:17:00	2016-12-07 09:58:00	9	9.0	Peak morning hours
3	2532	Airport	1.0	Trip Completed	2016-12-07 21:08:00	2016-12-07 22:03:00	21	22.0	Night hours
4	3112	City	1.0	Trip Completed	2016-07-13 08:33:16	2016-07-13 09:25:47	8	9.0	Peak morning hours

## Exporting the Updated Uber Dataset

In [23]:

```
uber.to_csv('uber_EDA_update.csv')
```

## Now it's time to move into Power BI for further Analysis...



# Data Visualization with Power BI

Power BI Dashboard Link:

[https://app.powerbi.com/links/qjkc7S8zG6?ctid=b1ca9c39-fb21-4b67-b237-225f1df64e8f&pbi\\_source=linkShare](https://app.powerbi.com/links/qjkc7S8zG6?ctid=b1ca9c39-fb21-4b67-b237-225f1df64e8f&pbi_source=linkShare)

## A. Overview: What the data telling us?

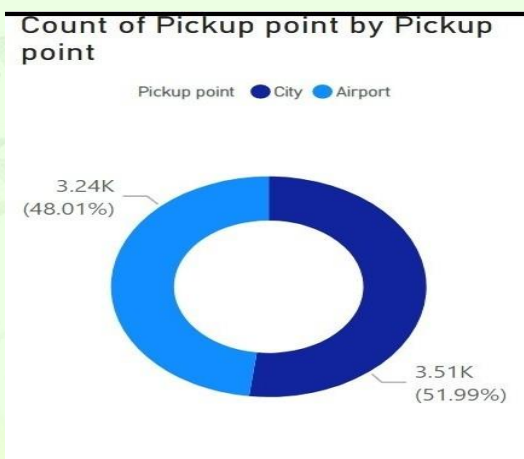


After Exploratory Data Analysis (EDA) and data manipulation this the overall story about Uber cab service in Bangalore city.

**Total Ride Request:** 6745

Now analyse each every visuals thoroughly.

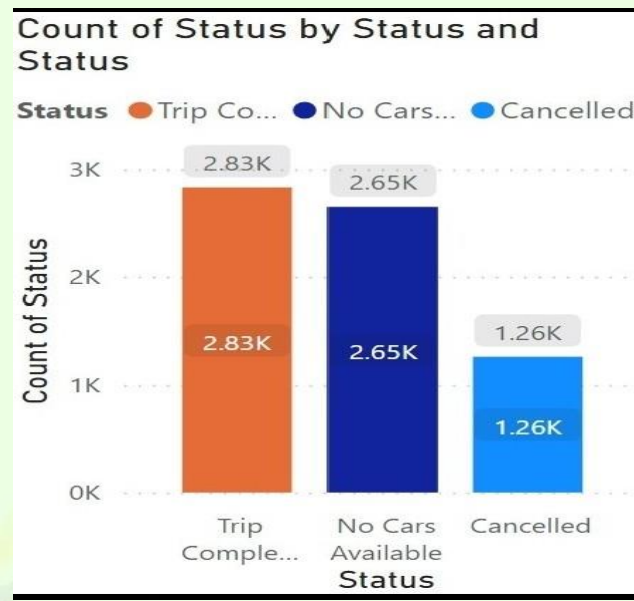
### 1. Pickup points:





City and Airport pick points are almost the same as per data but city has slightly higher in count. Airport have 3.2k i.e. 48.01% of pickup points where as city have 3.5k i.e. 51.99% of pickup points.

## 2. Cab Ride Status:



From the above visual I can see that 2.83k ride requests are completed, 2.65k is the number where cabs are not available which is very alarming and followed by unavailability of the cabs I can also see that cancelled rides is also having 1.26k in count.

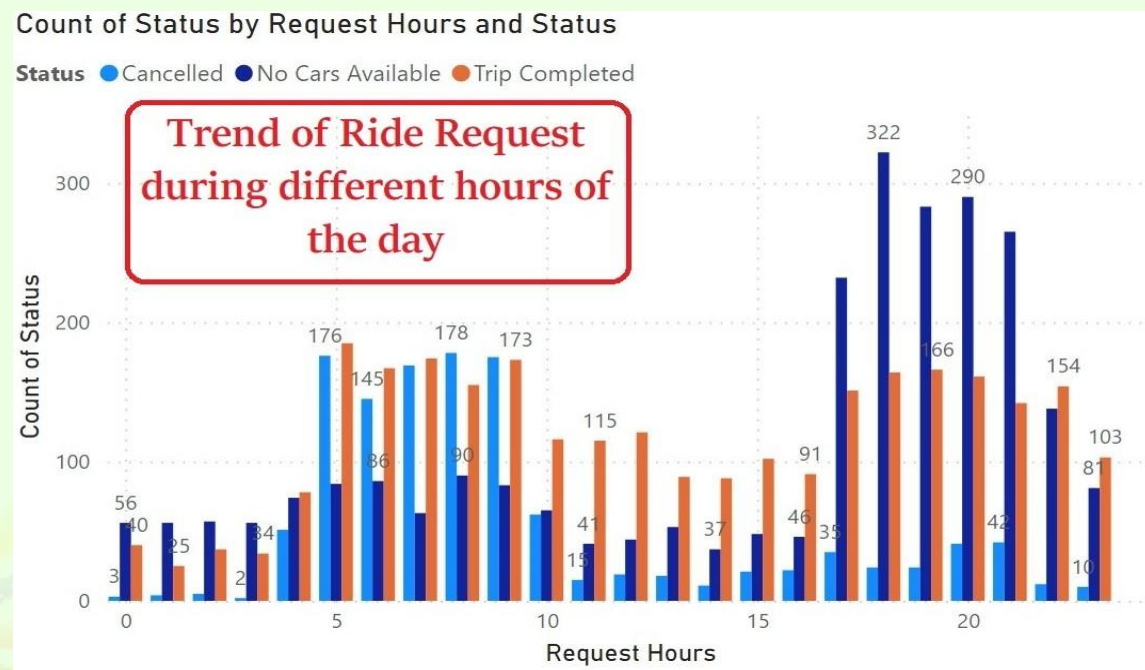
## 3. Cab Request by Time Slot:





The most cab request comes on evening time with 1.89k in count and also in early morning hours with 1.83k in count. Demands goes down in the time of peak morning, night hours and also very low in noon hours.

#### 4. Trend of Request throughout the day:

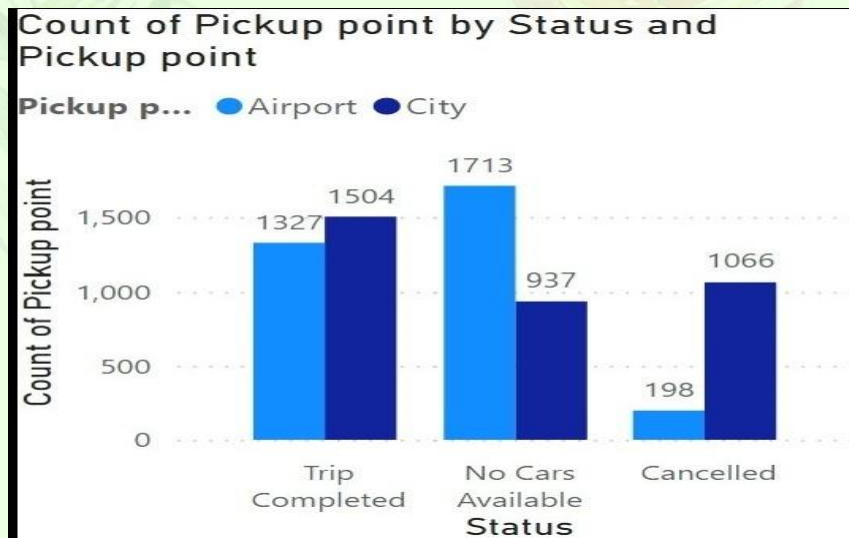


From the visuals I can see that cab request are moderate at 5am to 10am where cab cancellation & ride completed bars are much higher than usual, also the unavailability of the cabs is moderate as there is a office hour rush time.

Now, when I see in the time slot from 5pm to 10pm, the unavailability of cabs are very very high that's why we can see that cancellation of cabs gone drastically down and completion to trip is as high as morning hour office time.

#### 5. Problematic pickup points for all request status:



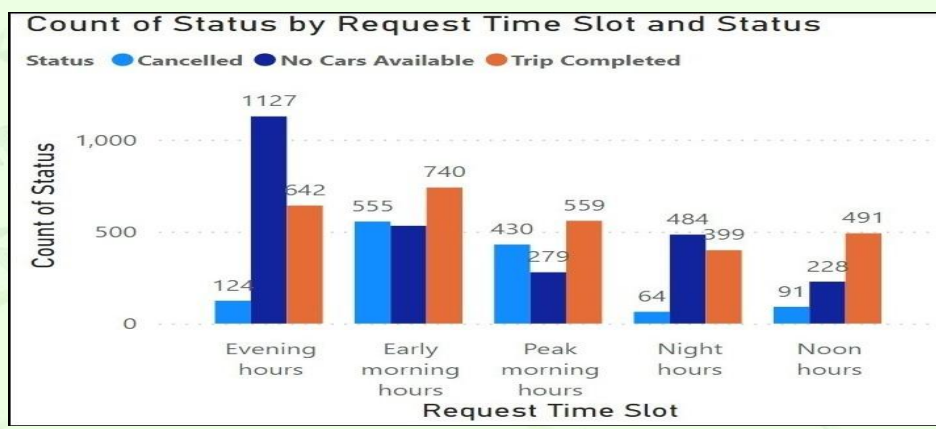


No Cars Available: Between 1600-1700 no of trips where pickup points is 'Airport' uber customers didn't get the car.

Trip Completed: Between 1400-1500 no of trips were completed where pickup points is city & around 1300 trips were completed for Airport pickup point.

Cancelled: Between 1000-1050 no of trips were cancelled as well as where pickup point is city which is a little less than trips completed where pickup point is city.

## 6. Trend of Trip throughout the day:

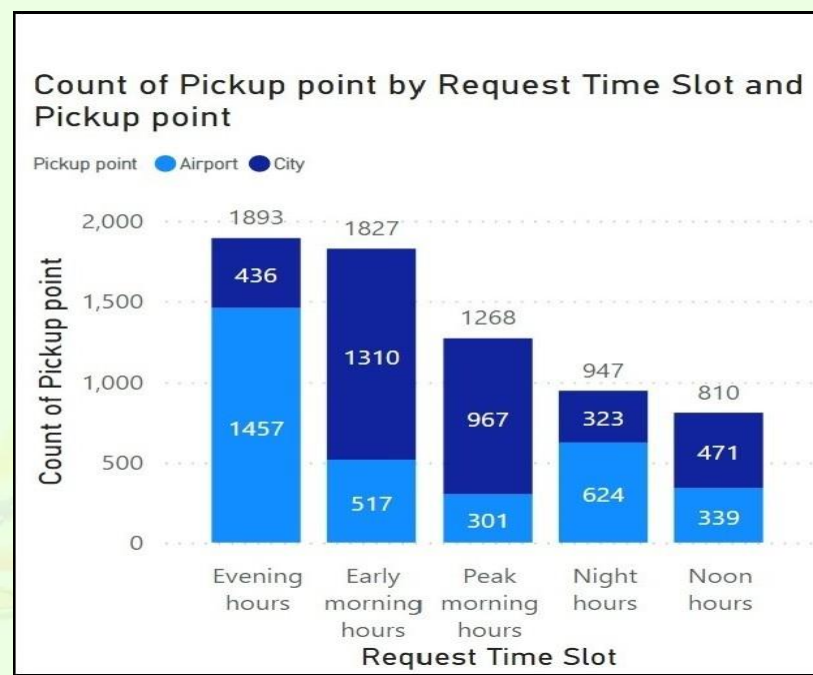


Evening Hours: (5-8pm): No. of request between 1000-1200 ended with

no cabs available.

Early morning hours:(1-7am): No. of request between 500-600 were cancelled and cars were not available. Approx 700-800 no of trips were completed.

## 7. 24 Hours Overall Pickup Point Trend:



Evening hours has the highest pickup points and request with 1893 in count where maximum count is from airport i.e. 1457.

Early morning hours have second highest pickup points with 1827 count but here the story is opposite, the city pickup point is much higher than airport i.e. 1310 in count.



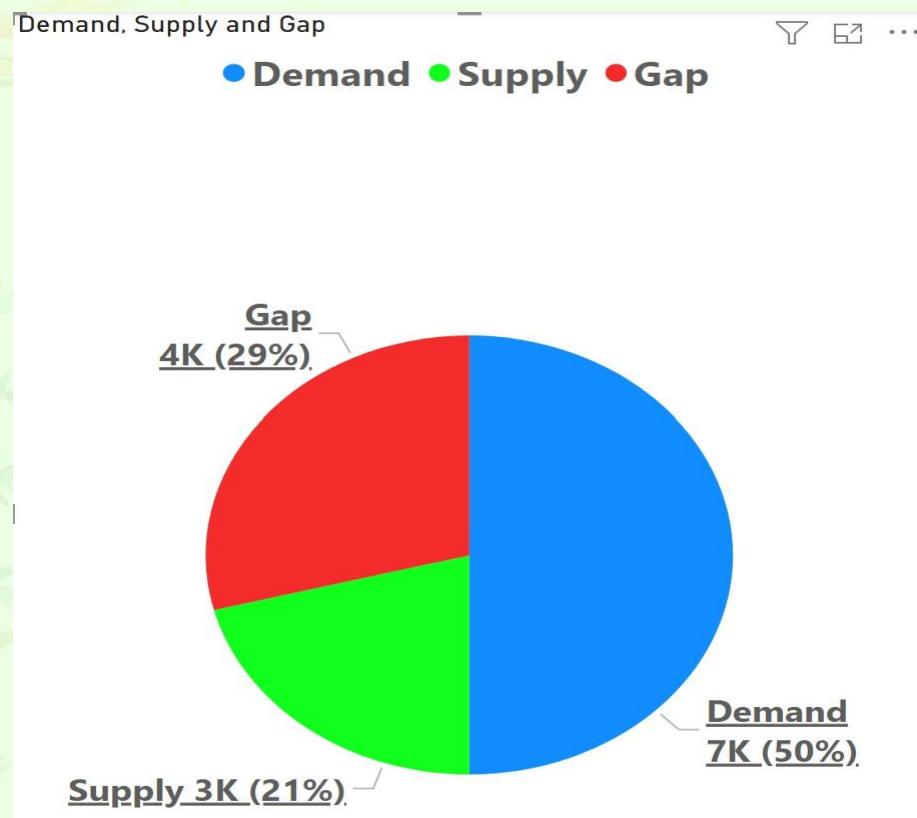
**B. Demand-Supply-Gap Analysis:** Is Uber performing well in Bengaluru as per 2016 data? Is there any gap in their cab service?

Let's find out.

**Calculating demand-supply-gap metrics:**

1. **Total Demand:** Total trips made with all three status. For doing this I made a quick measure in Power BI.
2. **Total Supply:** Total number of trips completed. For doing this I made a quick measure in Power BI.
3. **Gap:** Difference between total demand and supply. For doing this I made a quick measure in Power BI.

After creating the measures when I plot in dashboard I got this:





**Total Demand : 6745 (100%)**

**Total Supply : 2831 (41%)**

**Total Gap : 3914 (58%)**

**Below is the overall Demand-Supply-Gap details:**



### **A. Demand:**

#### **Demand by Request Time Slot:**



Evening Hours has the highest demand time slot with 1893 in count & second highest is the Early Morning hours with 1827 in count. Lowest demand in Noon hours with 810 in count.

#### Demand by Pickup Point:



In the case of Pickup point city has slightly higher demand (i.e. 3.5k in count) in comparison with airport, who has 3.2k demand in count.

#### **B. Supply:**

##### Supply by Request Time Slot:



Early morning hours has the highest supply time slot with 740 in count &



second highest is the Evening hours with 642 in count. Lowest supply by time slot in Night hours with 399 in count.

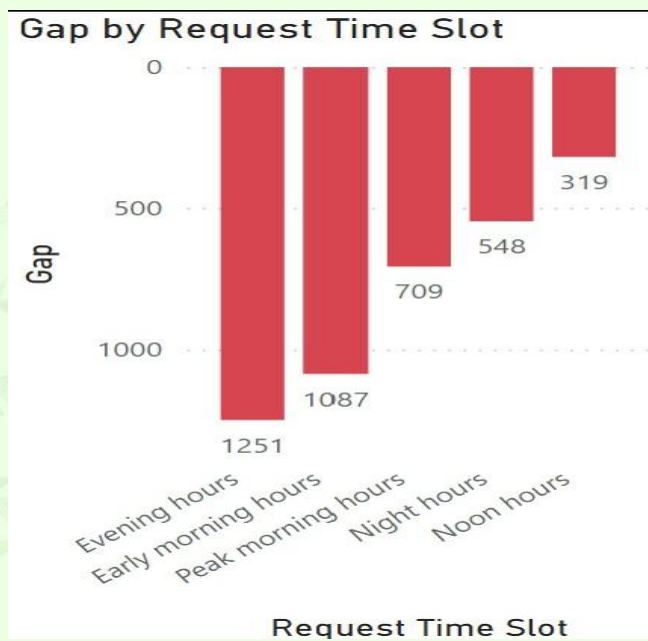
#### Supply by Pickup Point:



In the case of Pickup point city has slightly higher supply (i.e.1504 in count) in comparsion with airport, who has 1327 supply in count.

#### C. Gap:

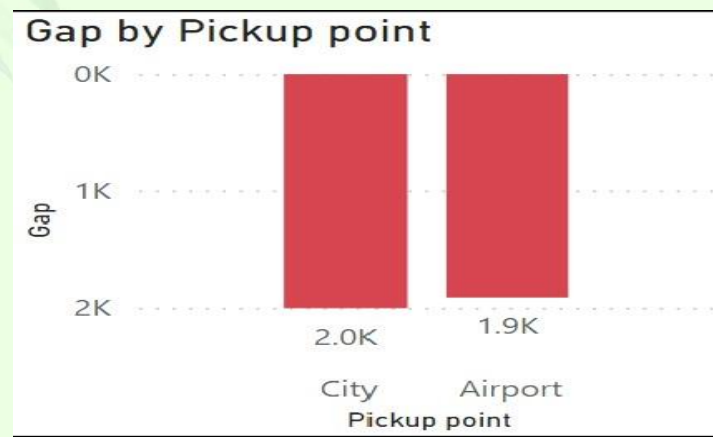
##### Gap by Request Time Slot:



Evening hours has the highest time slot gap with a count of 1251 and Early morning hours has second highest gap with a count of 1087. Noon

hours has the lowest gap i.e. 319.

Gap by Pickup point:



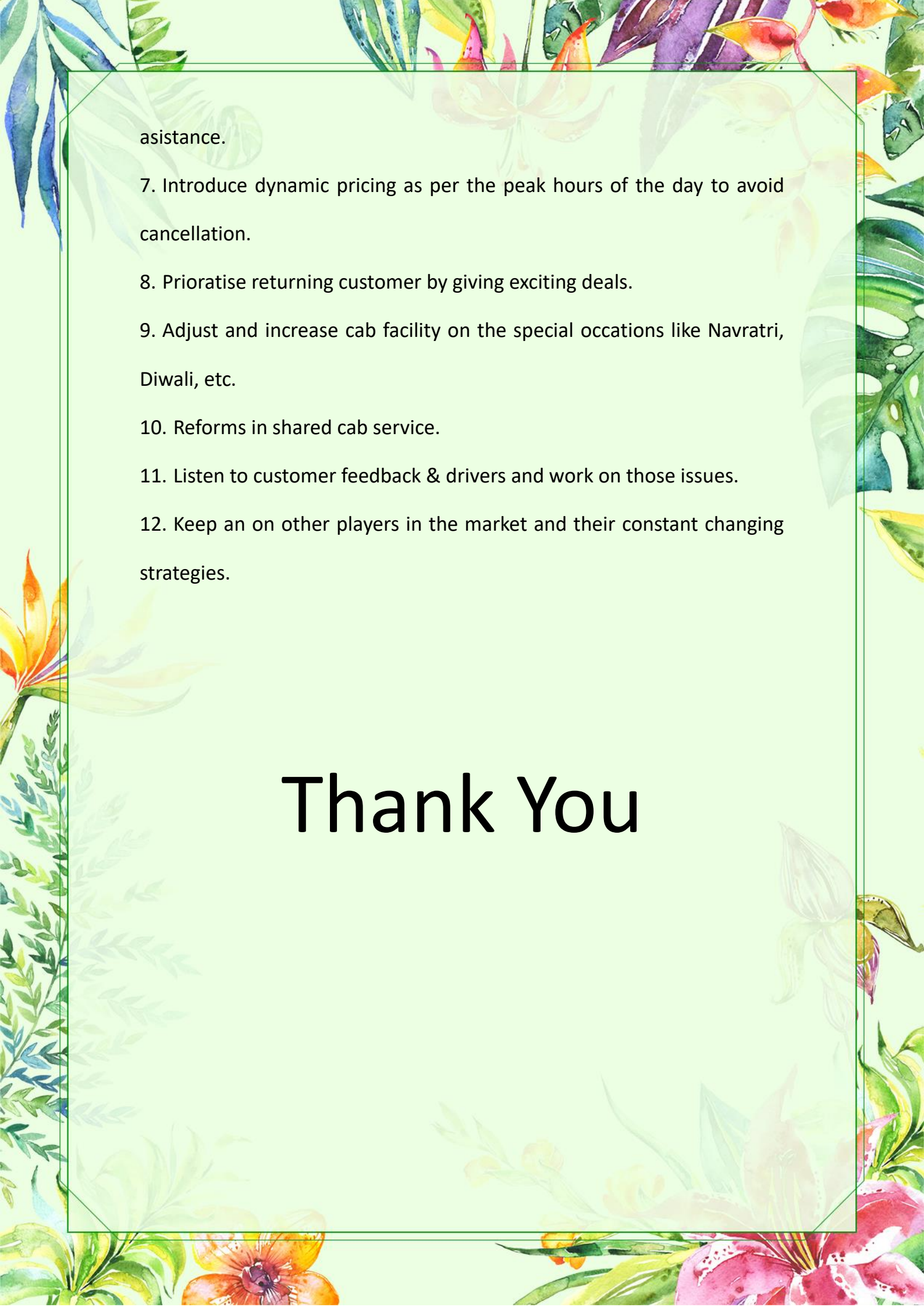
In the case of pick up point I can see there are similar gap of both city and airport. City has slightly higher gap i.e. 2.0k and airport has 1.9k gap.

**Verdict:**

Overall analysis says that uber has a huge gap in demand supply chain. Total demand is 6745 but there is a supply of only 2831 i.e. 41% and huge gap of 3914 which is 58%, so this is very alarming situation. They need to improve their service to reduce this huge gap & I have some suggestions given below:

1. Increase cab count in city.
2. Focus on Peak hours like office hours.
3. Also focus on non peak hours.
4. Increase pick up point availability.
5. Increase cab drivers & motivate them by providing lucrative incentives.
6. Look for investors to support their business by pumping financial



A decorative border made of watercolor-style illustrations of various tropical plants and flowers, including blue and green leaves, orange and yellow flowers, and pink and purple blooms, framing the central text area.

assistance.

7. Introduce dynamic pricing as per the peak hours of the day to avoid cancellation.

8. Prioratise returning customer by giving exciting deals.

9. Adjust and increase cab facility on the special occasions like Navratri, Diwali, etc.

10. Reforms in shared cab service.

11. Listen to customer feedback & drivers and work on those issues.

12. Keep an on other players in the market and their constant changing strategies.

# Thank You