OLA RIDES ANALYSIS PROJECT

• Project Overview

This project analyses OLA ride data with specific focus on ****ride cancellation patterns****, identifying root causes, and providing data-driven solutions to reduce cancellation rates and improve overall service efficiency.

• Dataset Summary

The dataset contains over 20,000+ rows and 19 columns. It includes data related to ride bookings, vehicle types, payment methods, customer and driver ratings, ride distances, and cancellation reasons.

• Data Cleaning And Standardization Using Python

- **Data Loading** : Imported Dataset Using Pandas
- **Initial Exploration** : Used df.info() to check structure and .describe() for summary statistics.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20407 entries, 0 to 20406
Data columns (total 20 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Date                         20407 non-null  object
 1   Time                         20407 non-null  object
 2   Booking_ID                   20407 non-null  object
 3   Booking_Status               20407 non-null  object
 4   Customer_ID                  20407 non-null  object
 5   Vehicle_Type                 20407 non-null  object
 6   Pickup_Location              20407 non-null  object
 7   Drop_Location                20407 non-null  object
 8   V_TAT                        12652 non-null  float64
 9   C_TAT                        12652 non-null  float64
 10  Canceled_Rides_by_Customer   2081 non-null   object
 11  Canceled_Rides_by_Driver     3654 non-null   object
 12  Incomplete_Rides             12652 non-null  object
...
25%          3.500000         3.500000            NaN
50%          4.000000         4.000000            NaN
75%          4.500000         4.500000            NaN
max          5.000000         5.000000            NaN
```

- **Standardization Of Column Names & Removing ALL Whitespaces** : using pandas string methods (.str.lower() , .str.replace() , .str.strip()) .

```
Index(['date', 'time', 'booking_id', 'booking_status', 'customer_id',
       'vehicle_type', 'pickup_location', 'drop_location', 'v_tat', 'c_tat',
       'canceled_rides_by_customer', 'canceled_rides_by_driver',
       'incomplete_rides', 'incomplete_rides_reason', 'booking_value',
       'payment_method', 'ride_distance', 'driver_ratings', 'customer_rating',
       'vehicle_images'],
      dtype='object')
```

- **Data Consistency Check** : Checked the datatypes of all numerical columns and changed for the required data type .
- **Database Integration**: Connected Python Script to MYSQL Workbench and located the cleaned dataset into the database for further inspection.

```
# Connecting to mysql
!pip install pymysql sqlalchemy
!-m pip install --upgrade pip
```

```
from sqlalchemy import create_engine
import mysql.connector
# Setting up connection with MYSQL
username = "root"
password = "Secure$4u"
host = "localhost"
port = "3306"
database = "OLA_RIDES_ANALYSIS"

engine = create_engine(f"mysql+pymysql://{username}:{password}@{host}:{port}/{database}")

```

```
# Write DataFrame to MySQL ( Loaded this dataframe under table name "TABLE1")
table_name = "TABLE1"   # choose any table name
df.to_sql(table_name, engine , if_exists="replace", index=False)
```

• Data Analysis & Finding Insights Using Python

We performed structured analysis in Python (pandas) to answer key business questions:

- **Retrieve all successful bookings :**

```
## TOTAL SUCCESSFUL or UNSUCCessful BOOKINGS
var=df[df['booking_status']=="Success"]
print(len(var))
```

```
12652
```

- **Find the average ride distance for each vehicle type:**

```
## FINDING AVERAGE DISTANCE RIDE FOR EACH VEHICLE ( let's do it for prime_suv)
var1=df[df['vehicle_type']=="prime_suv"]
var2=var1['ride_distance'].mean()
print(var2)
```

```
15.187096774193549
```

- **Get the total number of cancelled rides by customers:**

```
## CANCELLED RIDES BY THE CUSTOMERS or drivers
var3=df[df['canceled_rides_by_customer'].notna()]
print(len(var3))
```

```
2081
```

- **List the top 5 customers who booked the highest number of rides:**

```
## TOP 5 CUSTOMERS WHO BOOKED HIGHEST NUMBER OF RIDES or rating wise or booking value wise ( just adjust the filter condition & see th
var4 =df.groupby('customer_id').agg(number_of_bookings=('booking_id','count')).sort_values('number_of_bookings',ascending=False).head(
print(var4)
```

Python

customer_id	number_of_bookings
CID940408	3
CID393964	2
CID143850	2
CID190536	2
CID126952	2

- **Get the number of rides cancelled by drivers due to personal and car-related issues:**

```
## Number of rides cancelled by drivers due to personal and car-related issues or any other reasons as
var5 =df[df['canceled_rides_by_driver']=="Personal & Car related issue"]
print(len(var5))
```

Python

```
1263
```

- **Find the maximum and minimum driver ratings for Prime Sedan bookings:**

```
## MAX & MIN DRIVER RATING ON PRIME SUV or in any vehicle
#print(df['driver_ratings'].dtype) # firstly checking the datatype of this column so that numerical op
var6= df[df['vehicle_type']=="prime_suv"]
#print(var6[['vehicle_type','driver_ratings']])
print(var6['driver_ratings'].min())
print(var6['driver_ratings'].max())
```

✓ 0.0s Python

```
3.0
5.0
```

- **Retrieve all rides where payment was made using UPI:**

```python
## RIDES WHERE PAYMENT METHOD IS UPI or any other
#First cleaing the strings of payment methoid column
df['payment_method']=df['payment_method'].str.lower()
df['payment_method']=df['payment_method'].str.replace(' ','_')
df['payment_method']=df['payment_method'].str.strip()
print(df['payment_method'])
var7= df[df['payment_method']=="upi"]
print(len(var7))
```

Python

```
0              NaN
1             cash
2              upi
3              NaN
4      credit_card
          ...
20402          NaN
20403         cash
20404         cash
20405          upi
20406          NaN
Name: payment_method, Length: 20407, dtype: object
5113
```

- **Find the average customer rating per vehicle type:**

```python
## Average customer rating or driver ratings per vehicle type ( let's take mini vehicle here)
#print(df['customer_rating'].dtype) # Checking the Data type of cyustomer RATING COLUMN
var8= df[df['vehicle_type']=="mini"]
var9=var8['customer_rating'].mean()
print(var9)
```

✓ 0.0s Python

```
4.019822320932816
```

- **Calculate the total booking value of rides completed successfully:**

```python
## TOTAL value OF rides COMPLETED successfully or unsuccessfully
print(df['incomplete_rides'].unique()) # checking the different values inside the incomplete_rides column
var10= df[df['incomplete_rides']=="No"]
var11= var10['booking_value'].sum()
print(var11)
```

Python

```
[nan 'No' 'Yes']
6429897
```

- **List all incomplete rides along with the reason:**

```python
## ALL incomplete rides along with reason :
#print(df[['incomplete_rides_reason','incomplete_rides']].tail(30)) # just doing basic inspection
var12=df[df['incomplete_rides']=="Yes"]
#print(var12[['booking_id','customer_id','incomplete_rides','incomplete_rides_reason']])
print(var12[['booking_id','customer_id','incomplete_rides','incomplete_rides_reason']])
```

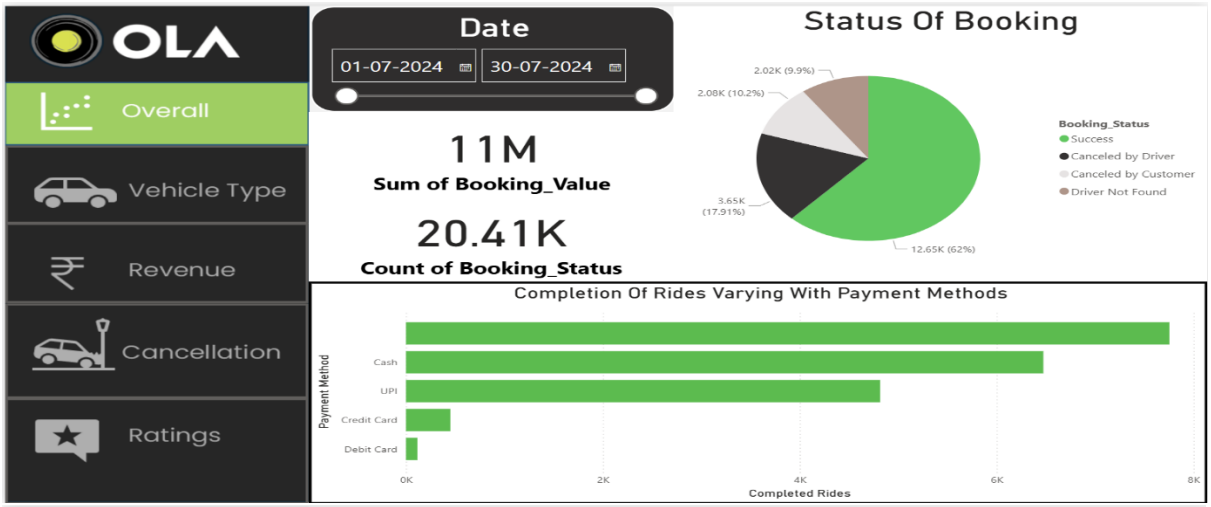
Python

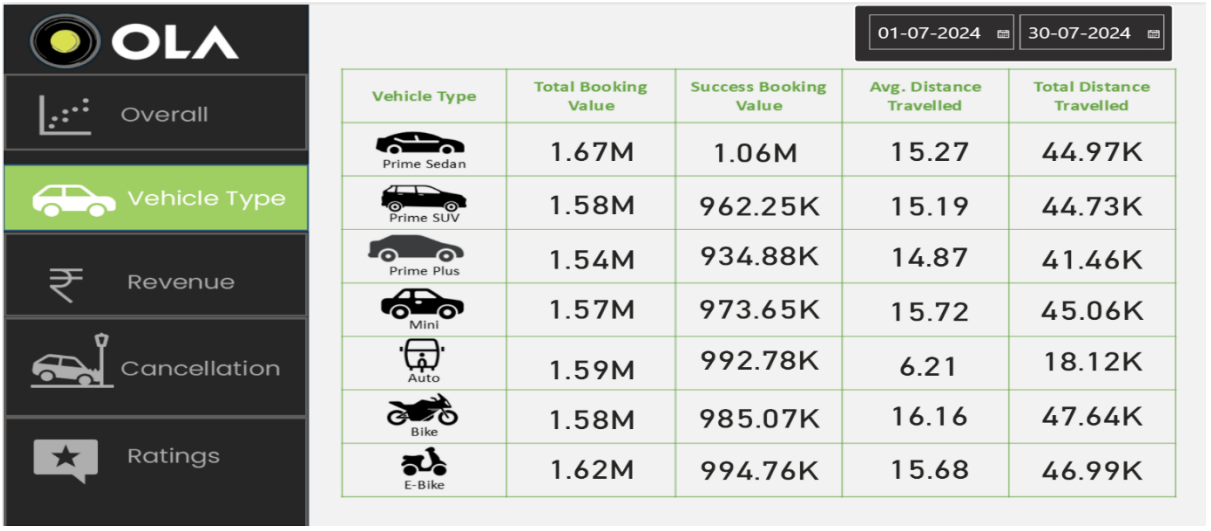
```
         booking_id customer_id incomplete_rides incomplete_rides_reason
38     CNR5176704322   CID296026             Yes         Customer Demand
49     CNR9312632867   CID649563             Yes       Vehicle Breakdown
70     CNR7924302885   CID517661             Yes         Customer Demand
93     CNR1640228587   CID190281             Yes            Other Issue
101    CNR7623690602   CID526261             Yes            Other Issue
...              ...         ...             ...                    ...
20197  CNR3500429121   CID108258             Yes            Other Issue
20275  CNR9654855291   CID923733             Yes            Other Issue
20289  CNR6348178557   CID948352             Yes       Vehicle Breakdown
20378  CNR7902243107   CID514958             Yes         Customer Demand
20390  CNR7459607546   CID845808             Yes         Customer Demand

[795 rows x 4 columns]
```

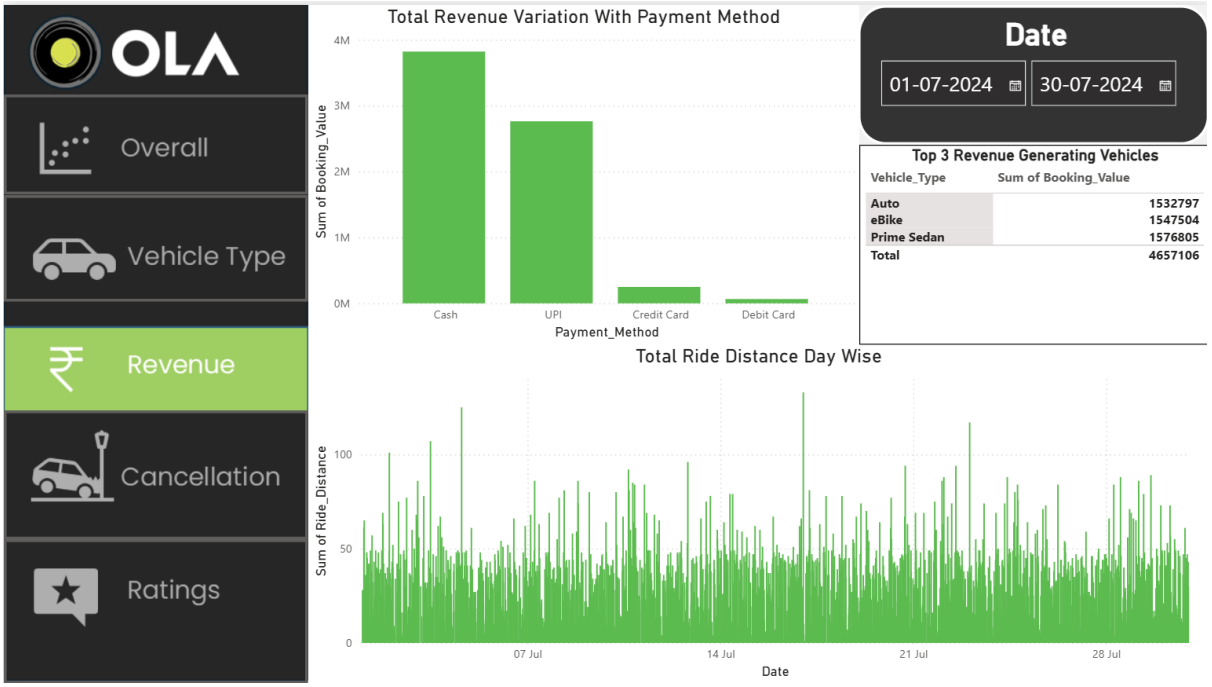
- # Dashboard In Power BI

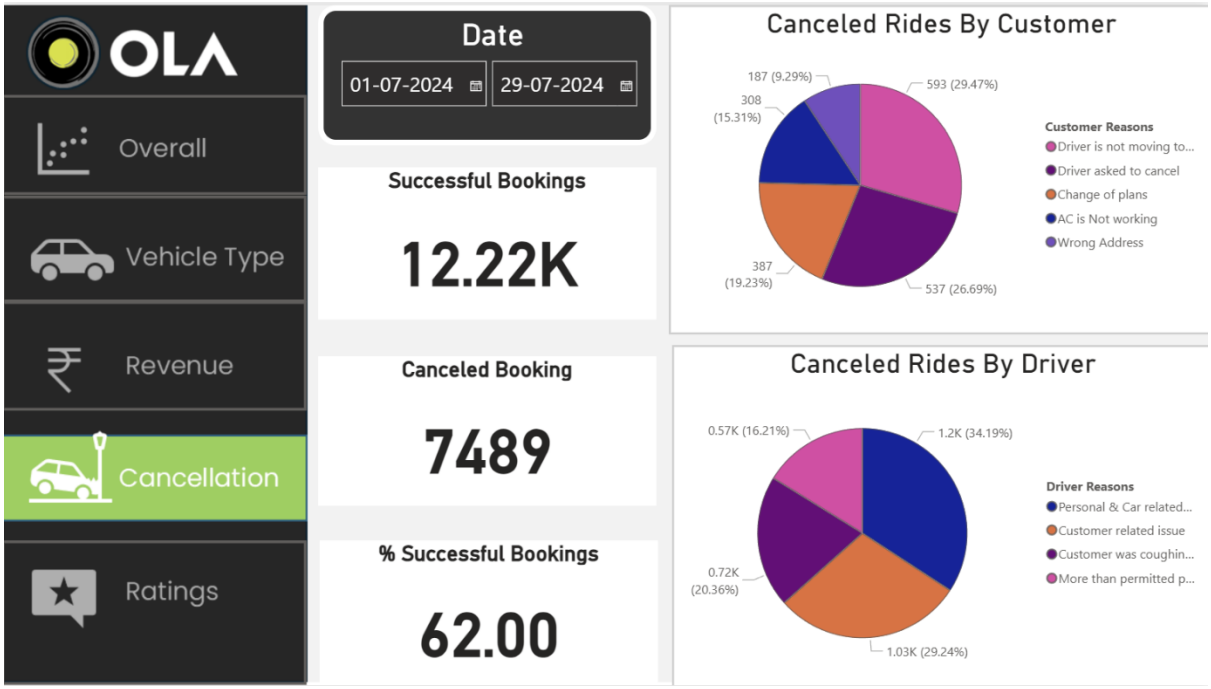
Finally, we built an interactive dashboard in Power BI to present insights visually.

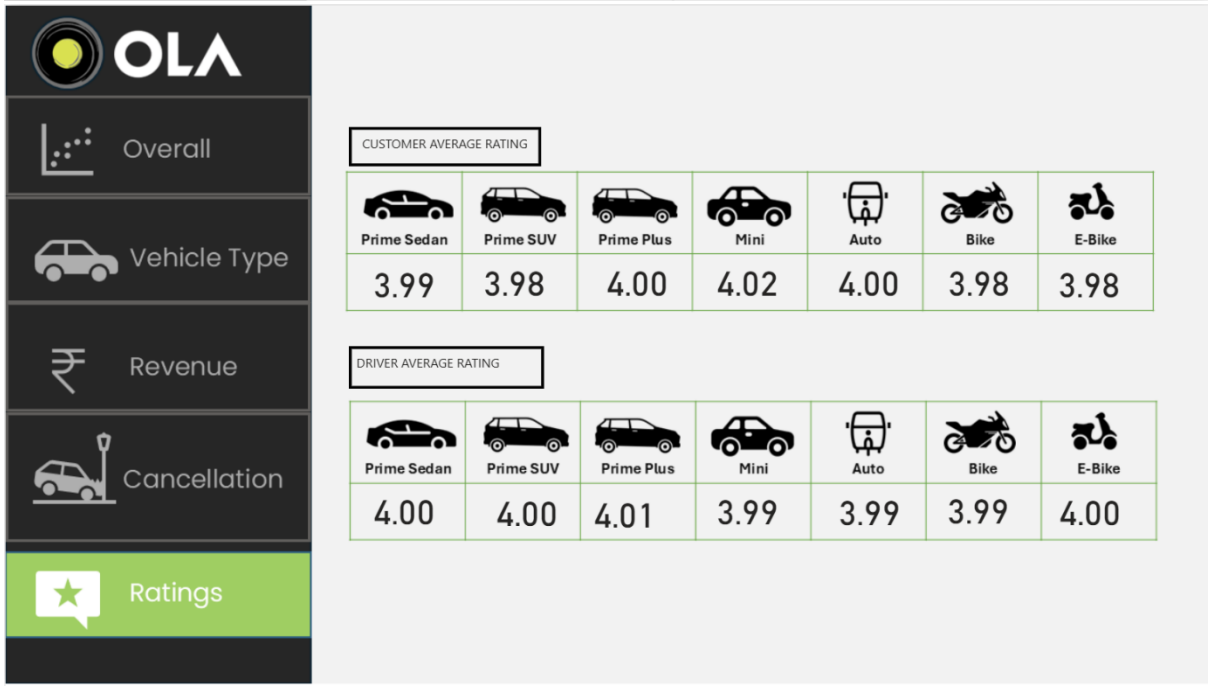




Vehicle Type	Total Booking Value	Success Booking Value	Avg. Distance Travelled	Total Distance Travelled
Prime Sedan	1.67M	1.06M	15.27	44.97K
Prime SUV	1.58M	962.25K	15.19	44.73K
Prime Plus	1.54M	934.88K	14.87	41.46K
Mini	1.57M	973.65K	15.72	45.06K
Auto	1.59M	992.78K	6.21	18.12K
Bike	1.58M	985.07K	16.16	47.64K
E-Bike	1.62M	994.76K	15.68	46.99K







• Business Insights

We found several high impactful insights which upon implementation can significantly reduce the cancellation rate and will increase overall efficiency .

1. **Payment Method Impact on Cancellations :** UPI payments show significantly lower cancellation rates compared to cash payments
2. **Vehicle Wise Cancellation Analysis :** Premium vehicles have higher cancellation rates due to availability issues

3. **Cancellation Reason Analysis :**
   Customer Side : Driver is not moving (29.47%) & Change of plans (26.69%)
   Driver Side : Personal & Car related (34.19%) & Customer related issue (29.24%)

4. **Rating Analysis :**
   Customer Side : Customers are happy with mini vehicles
   Driver Side : Drivers are happy with premium vehicles (like prime SUV)

• Business Recommendations

1. **Charge a fee for last-minute "change of plans " cancellations : this reason makes up over a quarter (26.69 %) of customer cancellations.**
2. **Fix the "Driver not moving issue " :** it's the biggest reason (29.47 %) of customer canceled rides.
3. **Help drivers to keep their cars in good condition :** "personal & car related issue " is the major reason (34.19 &) for driver canceled rides.
4. **Make a strict rule against drivers asking customers to cancel :** this bad practice is responsibe for (15.31%) of customer canceled rides.
5. **Give better cars and training to prime drivers :** these have genrally the lower ratings in comparison to the mini type vehicles.