Uber Ride and Demand Report

Introduction

Uber is a ride-hailing platform that connects passengers with drivers in real time. Efficient matching between ride demand and driver availability is crucial for maintaining service quality and customer satisfaction. However, during certain time periods and locations, mismatches between demand and supply can lead to ride cancellations and unfulfilled requests.

This project focuses on analyzing Uber ride request data to understand demand patterns, identify causes of ride failures, and evaluate overall operational performance using data analysis techniques.

Problem Statement

Uber faces operational challenges due to an imbalance between ride demand and driver availability. These challenges result in:

- Ride cancellations by drivers
- Ride failures due to unavailability of cars
- Increased customer dissatisfaction

Identifying when and where these failures occur is essential to improve service efficiency and decision-making.

Objectives

The main objectives of this project are:

- To analyze total Uber ride requests and their outcomes
- To calculate ride completion and failure rates
- To study demand patterns based on time and pickup location
- To identify key reasons for ride failures
- To visualize insights using an Excel dashboard
- To provide data-driven recommendations for improving operations

Dataset Description

The dataset contains information related to Uber ride requests, including request time, pickup location, and ride status.

Key Columns in the Dataset

Column Name	Description
Request_id	Unique identifier for each ride request
Pickup_point	Pickup location (City or Airport)
Status	Ride status (Trip Completed, Cancelled, No Cars Available)
Request_timestamp	Time when the ride was requested
Drop_timestamp	Time when the ride was completed

Tools Used

The following tools were used in this project:

- **Python**: Data cleaning and exploratory data analysis
- **SQL (MySQL)**: Querying data and calculating KPIs
- **MS Excel**: Dashboard creation and data visualization

Python EDA Summary

Exploratory Data Analysis (EDA) was performed using Python to understand, clean, and transform the Uber ride request dataset before analysis.

Data Preprocessing & Feature Engineering

During the EDA process, additional columns were created from existing timestamp fields to enable time-based analysis and performance measurement:

- **request_hour**
  Extracted from the request timestamp to analyze ride demand patterns across different hours of the day.
- **request_day**
  Derived from the request timestamp to study daily ride request trends.
- **ride_duration_min**
  Calculated using the difference between drop timestamp and request timestamp (for completed trips only) to measure ride duration in minutes.

These newly created columns helped in identifying peak hours, daily demand variations, and trip duration patterns.

Handling Missing Values

- The Driver ID column contained missing values for failed rides, indicating driver unavailability.
- The Drop Timestamp and Ride Duration columns contained missing values for rides that were not completed.
- These missing values were retained intentionally, as they represent meaningful business outcomes such as ride cancellations and unfulfilled requests.

EDA Findings Summary

- Ride demand varies significantly across different hours of the day.
- Peak hours show higher ride failure rates due to supply-demand imbalance.
- Airport pickup points experience a higher number of failed rides compared to city pickups.
- Completed rides provide valuable insights into ride duration patterns.

Dataset columns after EDA

Column Name	Description
Request_id	Unique identifier for each ride request
Pickup_point	Pickup location (City or Airport)
Driver_id	Unique identifier of the assigned driver (null for failed rides)
Status	Ride status (Trip Completed, Cancelled, No Cars Available)
Request_timestamp	Date and time when the ride was requested
Drop_timestamp	Date and time when the ride was completed (available for completed trips)
request_hour	Hour of the day extracted from the request timestamp
request_day	Day extracted from the request timestamp to analyze daily trends
ride_duration_min	Duration of the ride in minutes (calculated for completed trips only)

SQL Queries & Analysis

SQL was used to calculate critical performance metrics and validate insights obtained from Python EDA. This section presents the **five most important queries** that directly support the project objectives.

1. Total Ride Requests

Objective:
To determine the overall ride demand.

```
SELECT COUNT(*) AS total_ride_requests

FROM uber_data;
```

Analysis:
This query provides the total number of ride requests and establishes the scale of demand in the dataset.

2. Ride Status Distribution

Objective:
To analyze how ride requests are distributed across different ride outcomes.

```
SELECT status, COUNT(*) AS total_rides

FROM uber_data

GROUP BY status;
```

Analysis:
This helps identify the proportion of completed rides versus failed rides, giving an overview of service efficiency.

3. Ride Completion Rate

Objective:
To calculate the percentage of rides successfully completed.

```
SELECT

  ROUND(

    SUM(CASE WHEN status = 'Trip Completed' THEN 1 ELSE 0 END)

    / COUNT(*) * 100, 2

  ) AS completion_rate_percent

FROM uber_data;
```

Analysis:
The completion rate is a key KPI that reflects Uber’s ability to fulfill ride requests.

4. Hourly Ride Demand

Objective:
To identify peak demand hours during the day.

```
SELECT
    request_hour,
    COUNT(*) AS total_requests
FROM uber_data
GROUP BY request_hour
ORDER BY request_hour;
```

Analysis:
This query highlights time periods with high ride demand, which are critical for driver allocation planning.

5. Requests by Pickup Point

Objective:
To compare ride demand between different pickup locations.

```
SELECT
    pickup_point,
    COUNT(*) AS total_requests
FROM uber_data
GROUP BY pickup_point;
```

Analysis:
This helps identify location-based demand differences, particularly between city and airport pickups.

Excel Dashboard Overview

An interactive Excel dashboard titled **"Uber Ride and Demand Dashboard"** was created to visually represent key insights derived from Python EDA and SQL analysis. The dashboard provides a consolidated view of ride demand, completion performance, cancellations, and supply gaps across time and pickup locations.

1. Dashboard KPIs

The top section of the dashboard displays key performance indicators (KPIs) that summarize overall platform performance:

- **Total Requests:** Total number of ride requests received
- **Completed Trips:** Total number of successfully completed rides
- **Cancellation Rate:** Percentage of ride requests that were not completed

These KPIs provide a quick snapshot of Uber's operational efficiency.

2. Hourly Ride Demand Analysis

The **Hourly Ride Demand** line chart shows how ride requests vary across different hours of the day.

- Demand peaks during **morning and evening hours**
- Mid-day hours show relatively lower request volumes
- Peak demand hours align with higher failure and cancellation trends

This visualization helps identify critical time periods requiring better driver availability.

3. Completed vs Failed Trips

The **Completed vs Failed Trips** bar chart compares successful and unsuccessful rides on an hourly basis.

- Failed trips increase during peak demand hours
- Completion rates are lower when demand exceeds supply
- Indicates a clear **supply–demand imbalance** during high-traffic periods

4. Ride Status Breakdown

The **Ride Status Breakdown** chart shows the distribution of:

- Trip Completed
- Cancelled
- No Cars Available

The analysis highlights that **"No Cars Available"** is a major contributor to ride failures, indicating driver shortages rather than customer-driven cancellations.

5. Supply Gap Analysis

The **Supply Gap Analysis** chart compares ride outcomes between **City** and **Airport** pickup points.

- Airport locations show a higher number of failed rides
- City locations have better ride completion performance
- Suggests the need for improved driver allocation near airports

6. Requests by Pickup Location

The **Requests by Pickup Location** chart compares total ride demand between City and Airport pickups.

- City pickups generate higher overall demand
- Airport pickups, despite lower demand, experience higher failure rates

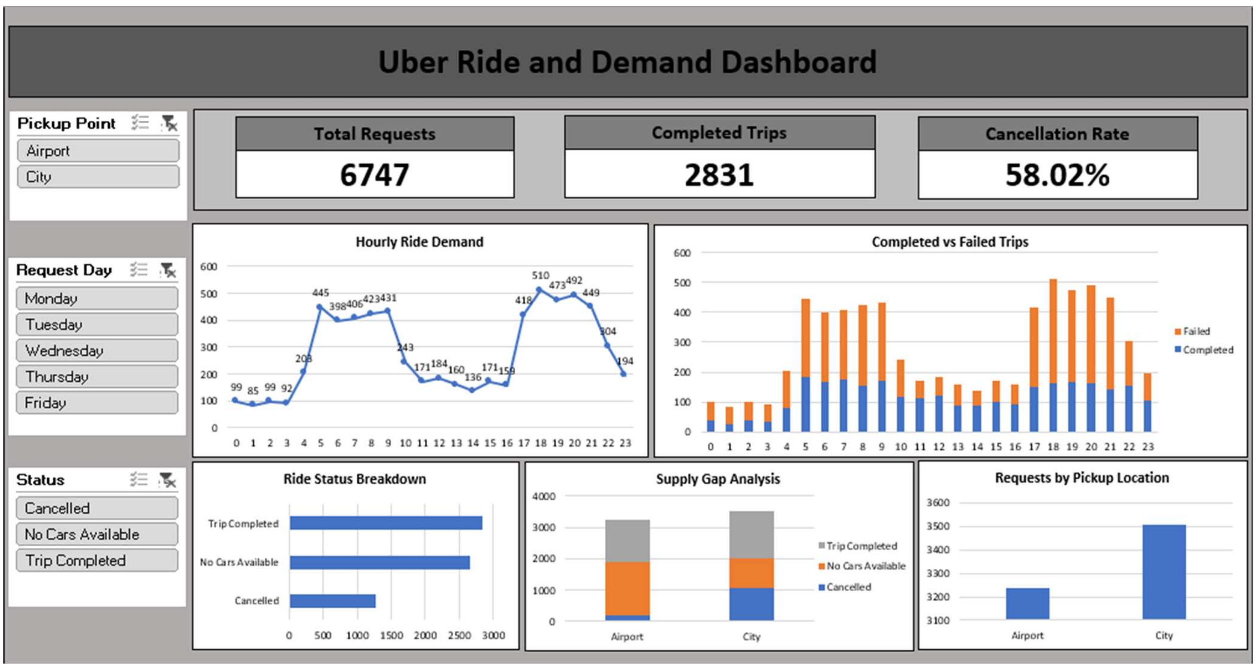
7. Interactive Filters

The dashboard includes slicers for:

- **Pickup Point**
- **Request Day**
- **Ride Status**

These filters allow users to dynamically explore trends based on location, day, and ride outcome.

8. Dashboard Visualization



Key Insights

- Ride demand shows clear **time-based patterns**, with peak requests occurring during **morning and evening hours**.
- **Ride failure rates increase during peak demand periods**, indicating a mismatch between ride demand and driver availability.
- **Airport pickup locations experience higher ride failure rates** compared to city pickups, despite having lower overall demand.
- A significant portion of failed rides is due to **"No Cars Available"**, highlighting supply-side constraints rather than customer cancellations.

- **City pickup locations demonstrate better ride completion performance**, suggesting more stable driver availability in urban areas.

- Completed trips provide consistent ride duration patterns, which can be used to estimate driver turnover and availability during peak hours.

Recommendations

Based on the analysis and insights derived from Python EDA, SQL queries, and the Excel dashboard, the following recommendations are proposed to improve Uber's operational efficiency and service quality:

- **Increase driver availability during peak hours**
  Deploy more drivers during morning and evening peak demand periods to reduce ride failures and cancellations.
- **Improve driver allocation near airport locations**
  Introduce location-based driver incentives to ensure sufficient driver presence at airports, where higher failure rates are observed.
- **Use demand forecasting for proactive planning**
  Historical demand patterns can be used to predict peak hours and plan driver supply in advance.
- **Introduce targeted driver incentives**
  Offer bonuses or surge-based incentives during high-demand time slots to encourage driver participation.
- **Monitor cancellation and failure metrics regularly**
  Continuous tracking of ride failures and cancellation rates can help identify emerging issues and take timely corrective action.

Conclusion

This project successfully analyzed Uber ride request data using Python, SQL, and Microsoft Excel to understand demand patterns, service performance, and operational challenges. Through exploratory data analysis and structured querying, key trends related to peak demand hours, ride completion rates, and pickup location performance were identified.

The analysis revealed that ride failures increase during high-demand periods and are more prominent at airport pickup locations, primarily due to driver unavailability. The Excel dashboard effectively visualized these insights, enabling quick interpretation of key metrics and trends.

Overall, this project demonstrates how data-driven analysis can help identify inefficiencies in ride-hailing operations and support informed decision-making. The insights and recommendations derived from this study can assist in improving driver allocation, reducing ride failures, and enhancing customer satisfaction.