METROCAR FUNNEL ANALYSIS

REPORT CREATED BY: BIJAL PATEL

CREATED ON 19TH JANUARY, 2024

SUMMARY:

Metrocar's business model (a ride-sharing app) is based on a platform that connects riders with drivers through a mobile application. Metrocar acts as an intermediary between riders and drivers, providing a user-friendly platform to connect them and facilitate the ride-hailing process.

This report provides a comprehensive analysis of Metrocar’s customer funnel, aiming to optimize its performance. The analysis covers various aspects, including platform usage, age demographics, surge pricing, Drop-off rates and Conversion rates at different funnel stages.

In our analysis we found out that:

- The “ride completion” and “ride request” stages experience almost a 50% drop-off rate each, making those funnel steps a primary target to research and improve.
- iOS seems to be the most popular platform with around 60% user engagement among the entire customer base.
- Users aged 35-44 outnumbered the others followed by age 25-34. Hence, both age groups could be made a prime target.
- Two peak periods were observed i.e between 8AM - 9AM and 4PM - 5PM. Hence, Surge pricing strategy should be applied to these two peak periods with careful consideration to balance supply and demand while maintaining positive user and driver experiences.
- The stage with the lowest conversion rate of 50.77% in the funnel is the “ride completed” stage, possibly due to long waiting time and especially highly competitive environment during the peak hours.

CONTEXT:

This project aims to analyze the customer funnel of Metrocar, a ride-sharing app (similar to Uber/Lyft), to identify areas for improvement and optimization.

SQL was used to query the data and Tableau for data visualization.

Several business questions have been asked to uncover valuable insights for improving specific areas of the customer funnel. A task has been assigned to conduct a funnel analysis and address the business questions explaining the reason for the recommendations based on insights retrieved from the data.

Below are those business question:

- What steps of the funnel should we research and improve? Are there any specific drop-off points preventing users from completing their first ride?
- Metrocar currently supports 3 different platforms: iOS, android, and web. To recommend where to focus our marketing budget for the upcoming year, what insights can we make based on the platform?
- What age groups perform best at each stage of our funnel? Which age group(s) likely contain our target customers?
- Surge pricing is the practice of increasing the price of goods or services when there is the greatest demand for them. If we want to adopt a price-surging strategy, what does the distribution of ride requests look like throughout the day?
- What part of our funnel has the lowest conversion rate? What can we do to improve this part of the funnel?

Metrocar's Funnel:

The customer funnel for Metrocar typically includes the following stages:

1. App Download: A user downloads the Metrocar app from the App Store or Google Play Store.

2. Signup: The user creates an account in the Metrocar app, including their name, email, phone number, and payment information.

3. Request Ride: The user opens the app and requests a ride by entering their pickup location, destination, and ride capacity (2 to 6 riders).

4. Driver Acceptance: A nearby driver receives the ride request and accepts the ride.

5. Ride: The driver arrives at the pickup location, and the user gets in the car and rides to their destination.

6. Payment: After the ride, the user is charged automatically through the app, and a receipt is sent to their email.

7. Review: The user is prompted to rate their driver and leave a review of their ride experience.

Dataset:

The dataset was provided by Metrocar which was inspired by publicly available datasets for Uber/Lyft. Below is the URL to access the dataset:

postgres://Test:bQNxVzJL4g6u@ep-noisy-flower-846766-pooler.us-east-2.aws.neon.tech/Metrocar

Data Structure:

- app_downloads: contains information about app downloads
  - app_download_key: unique id of an app download
  - platform: ios, android or web
  - download_ts: download timestamp
- signups: contains information about new user signups
  - user_id: primary id for a user
  - session_id: id of app download
  - signup_ts: signup timestamp

    - age_range: the age range the user belongs to

- ride_requests: contains information about rides
    - ride_id: primary id for a ride
    - user_id: foreign key to user (requester)
    - driver_id: foreign key to driver
    - request_ts: ride request timestamp
    - accept_ts: driver accept timestamp
    - pickup_location: pickup coordinates
    - destination_location: destination coordinates
    - pickup_ts: pickup timestamp
    - dropoff_ts: dropoff timestamp
    - cancel_ts: ride cancel timestamp (accept, pickup and dropoff timestamps may be null)
- transactions: contains information about financial transactions based on completed rides:
    - ride_id: foreign key to ride
    - purchase_amount_usd: purchase amount in USD
    - charge_status: approved, canceled
    - transaction_ts: transaction timestamp
- reviews: contains information about driver reviews once rides are completed
    - review_id: primary id of review
    - ride_id: foreign key to ride

- driver_id: foreign key to driver
- user_id: foreign key to user (requester)
- rating: rating from 0 to 5
- free_response: text response given by user/requester

Method:

SQL was used to query the data and Tableau for data visualization.

The Funnel analysis for conversion rate and Drop-off rate has been conducted on two level:

- User Level - the funnel entirely at user- level granularity
- Rides Level - the funnel entirely at rides-level granularity

In Tableau, a parameter was created to toggle between Users Funnel and Rides Funnel.

It's noteworthy that the funnel incorporates two visualized approaches to measure both Conversion Rates and Drop-off Rates at each stage. Those two approaches are as below:

1. Percent of Top - This calculation involves measuring the conversion rate as a percentage of the total number of users who entered the top of the funnel. For our Metrocar example, app downloads represent the top of the funnel.

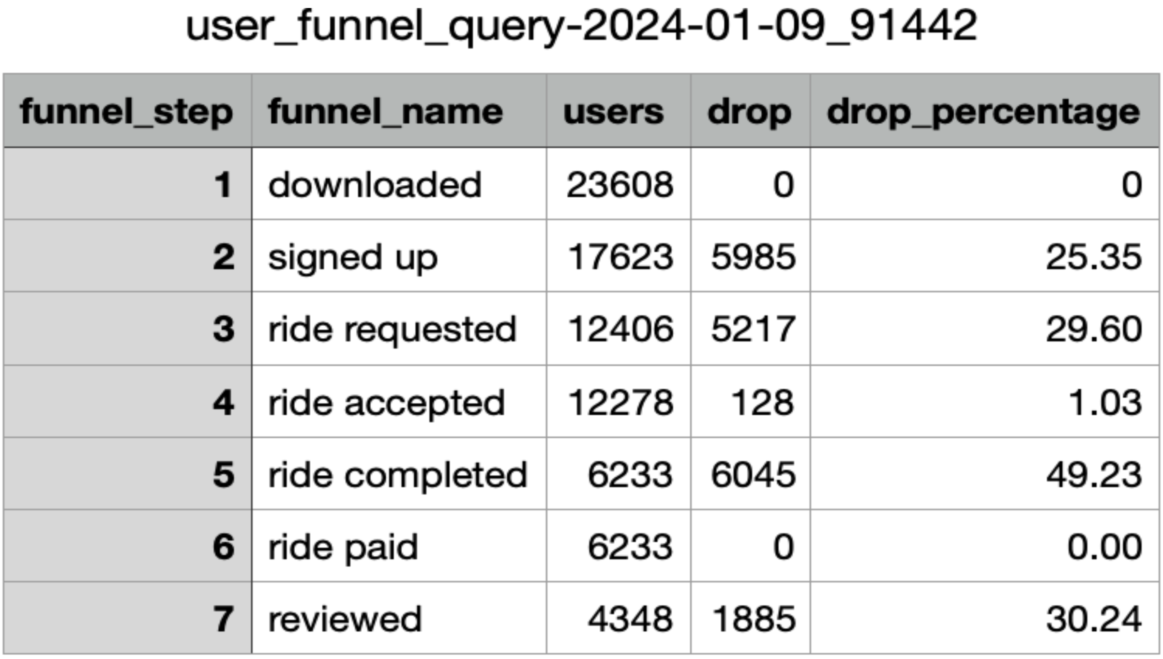
2. Percent of Previous - This calculation involves measuring the conversion rate as a percentage of the users who proceeded to the current stage of the funnel, relative to the number of users who were at the previous stage. In other words, it tracks the progression of users through each stage of the funnel.

Some more findings from the Dataset above using SQL: (SQL query codes included in Appendix)

- Total number of app downloaded: 23,608.
- Total number of users signed up: 17,623.
- Total Rides Requested: 3,85,477
- Total Rides Completed: 2,23,652
- Total Unique Users: 12,406
- Average time from ride pick-up to ride drop-off: 52 minutes: 36.73 seconds
- Total rides accepted: 2,48,379
- Total Payment collected: $42,51,667.61
- Total ride request from each platform:
    a. iOS - 2,34,693
    b. Android - 1,12,317
    c. Web - 38,467
- Drop-off from users signing up to users requesting a ride - 29.60%

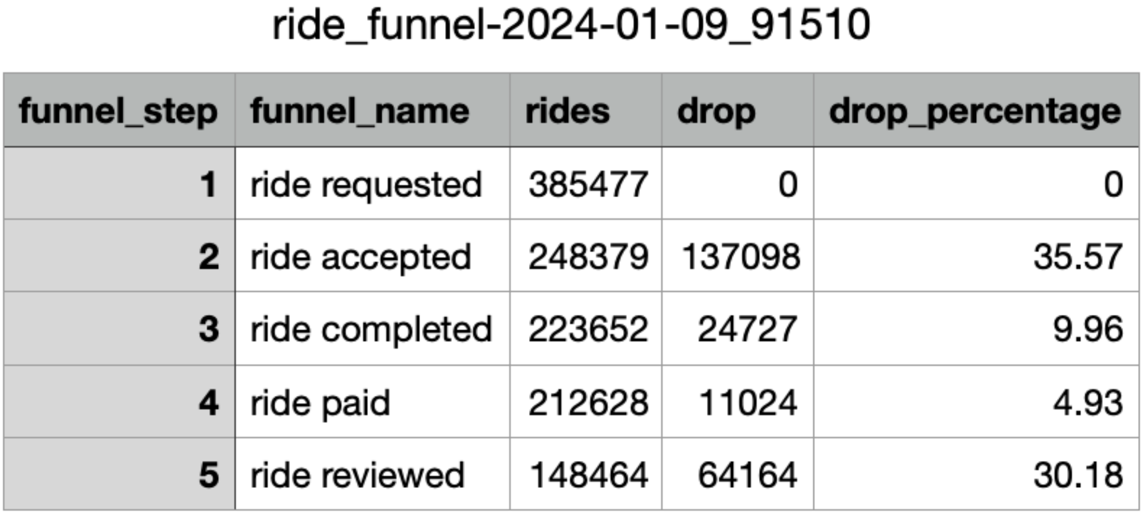
RESULTS:

Firstly, with the dataset given, both funnels were initially built with PostgreSQL. Including: funnel step, funnel stage, total users/rides at each funnel step, Total users/rides drop count from previous funnel stage and drop percentage. (SQL query code for the below table is included in Appendix)



user_funnel_query-2024-01-09_91442

funnel_step	funnel_name	users	drop	drop_percentage
1	downloaded	23608	0	0
2	signed up	17623	5985	25.35
3	ride requested	12406	5217	29.60
4	ride accepted	12278	128	1.03
5	ride completed	6233	6045	49.23
6	ride paid	6233	0	0.00
7	reviewed	4348	1885	30.24



ride_funnel-2024-01-09_91510

funnel_step	funnel_name	rides	drop	drop_percentage
1	ride requested	385477	0	0
2	ride accepted	248379	137098	35.57
3	ride completed	223652	24727	9.96
4	ride paid	212628	11024	4.93
5	ride reviewed	148464	64164	30.18

Upon achieving this, Data Visualization was done using Tableau. This involved collecting and aggregating data once more, utilizing PostgreSQL to construct a suitable extract for Tableau.

Two extracts were created:

- User Extract: included Funnel name, Users, Platform, Age Group, Download Time, Download Date
- Ride Extract: included Funnel name, Rides, Platform, Age Group, Download Time, Download Date

(SQL query code for User Extract and Ride Extract is included in Appendix)

Following Business questions were answered with the above aggregated data through data visualization in Tableau:

1. What steps of the funnel should we research and improve? Are there any specific drop-off points preventing users from completing their first ride?

   ➔ Here we created visualization for both User Funnel as well as Rides Funnel. To answer the above question, I will refer to the User Funnel as below:

   Finding: The User Funnel visualization reveals that the highest drop-off occurred at the "ride requested" stage, with a 49.23% drop-off calculated by the ‘Percent of Previous’ method and a 73.60% drop-off calculated by the ‘Percent of Top’ method. Additionally, there is an overall drop-off of 54.95% at the combined "signed up" + "ride requested" stage, calculated by the ‘Percent of Previous’ method. When using the ‘Percent of Top’ method, the total drop-off at this stage was 72.80%.

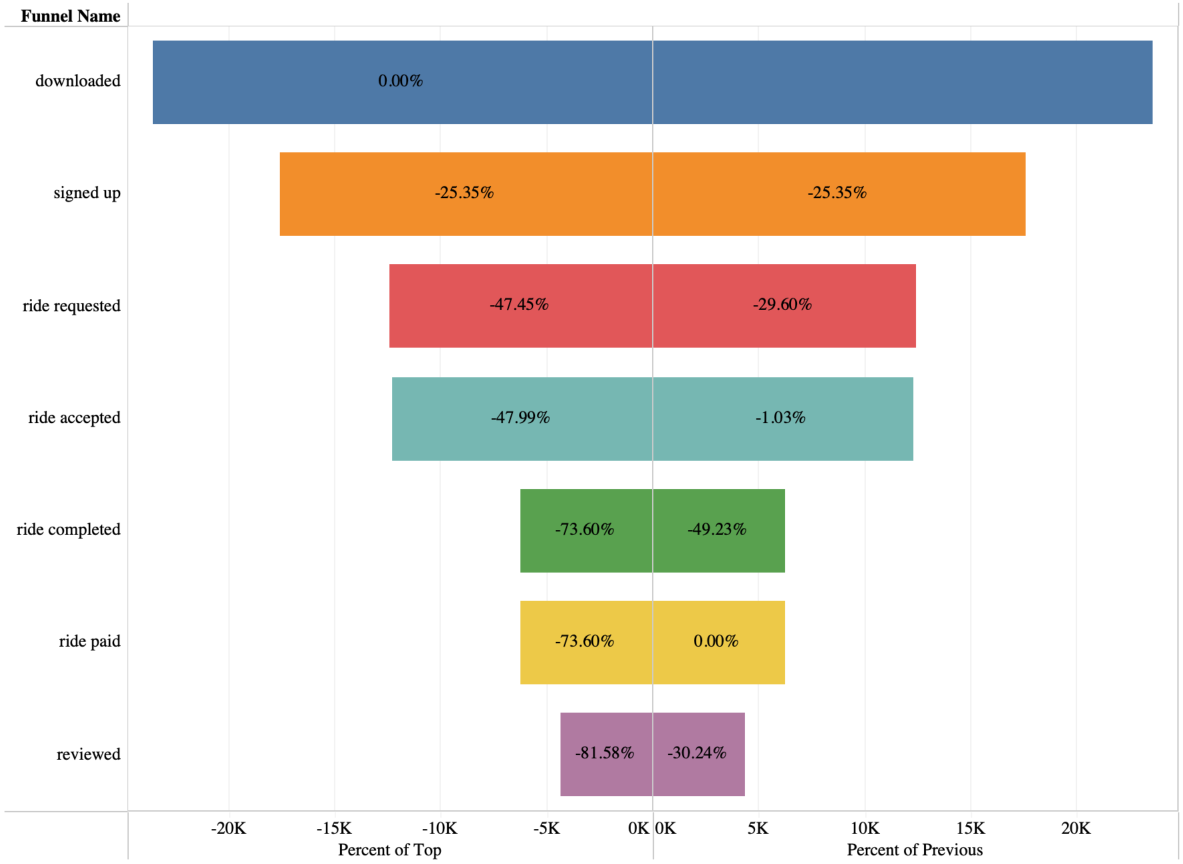


Tableau Interactive Visualization Link:

https://public.tableau.com/views/UserRideFunnelConversionRateforeachstageTopvsPrevious_17055235335770/Sheet5?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Recommendation: To encourage users to request their first ride and ensure the completion of accepted rides, my recommendation would be to implement the following strategies:

- Promotional offers: Introducing first-time user discounts to incentivize users to initiate their first ride. This promotional offer can attract new users and encourage them to request their first ride.
- Referral program: Implementing a referral program offering incentives such as welcome bonuses and ride credits for both the referring user and the new user.

- Immediate Ride Confirmation notification to users and drivers: Prompt communication enhances the user experience, reduces uncertainty, and ensures a seamless process from ride request to ride completion.
- Incentivize prompt departure for drivers: This encourages drivers to depart promptly upon accepting a ride by implementing incentives. This could include bonuses or rewards for timely departures.
- Implementing Ride cancellation fees for user and driver with reason codes: This discourages last-minute cancellations, ensures commitment from both parties, and minimizes disruptions to the service.

2. Metrocar currently supports 3 different platforms: ios, android, and web. To recommend where to focus our marketing budget for the upcoming year, what insights can we make based on the platform?

   ➔ **Findings:** The visualization below indicates that iOS boasts the highest user engagement at 60%, while the web platform exhibits the lowest level of engagement.

   Recommendation: Since IOS platform consistently shows higher user engagement and satisfaction, consider focusing marketing budget on this platform for acquiring and retaining users for upcoming year.

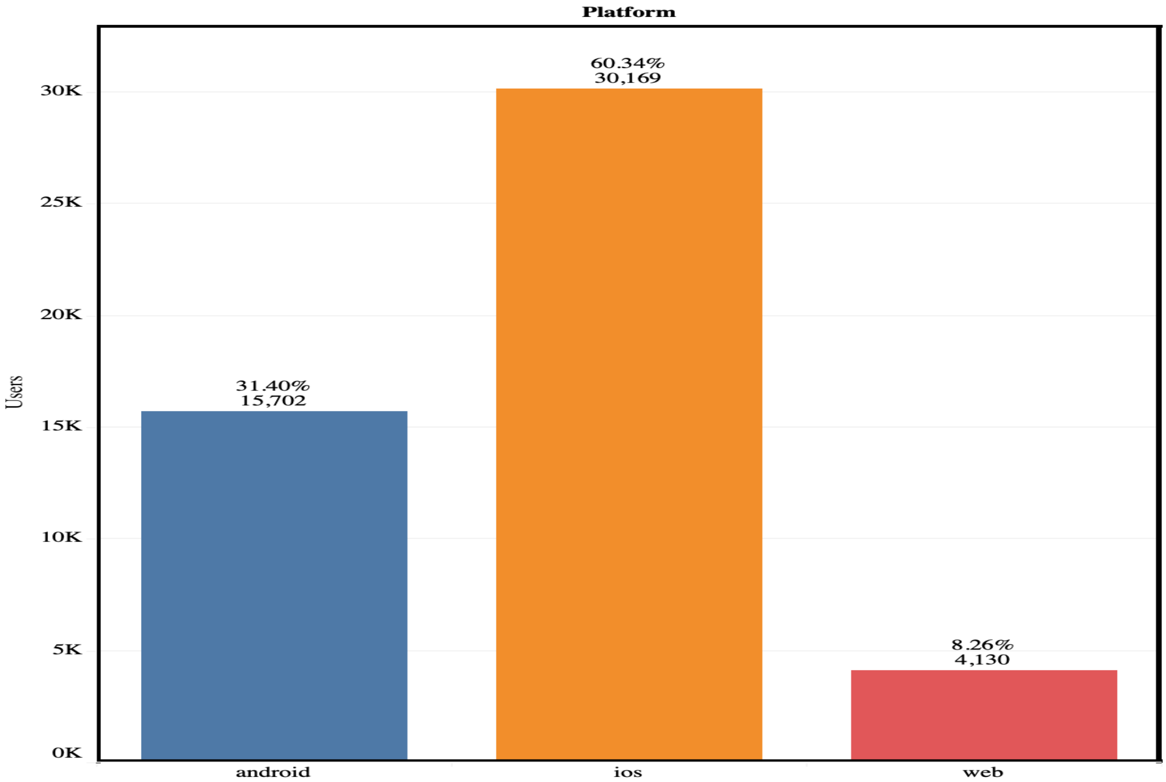


Tableau Interactive Visualization Link:

https://public.tableau.com/views/UserRideFunnelConversionRateforeachstageTopvsPrevious_17055235335770/Sheet2?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

3. What age groups perform best at each stage of our funnel? Which age group(s) likely contain our target customers?

   ➔ **Findings:** The age group of 35-44 exhibits the highest user performance, reaching 27.26%, closely trailed by the 25-34 age group at 18.18%.

   Recommendation: Therefore, our target customers primarily fall within the age groups of 35-44 and 25-34.

It's important to note that 35.37% of users did not share their age. Encouraging users to share age information would lead to more precise findings. Implementing incentives and rewards for completing profiles, along with making Age a 'required field' in profiles, can facilitate this process.

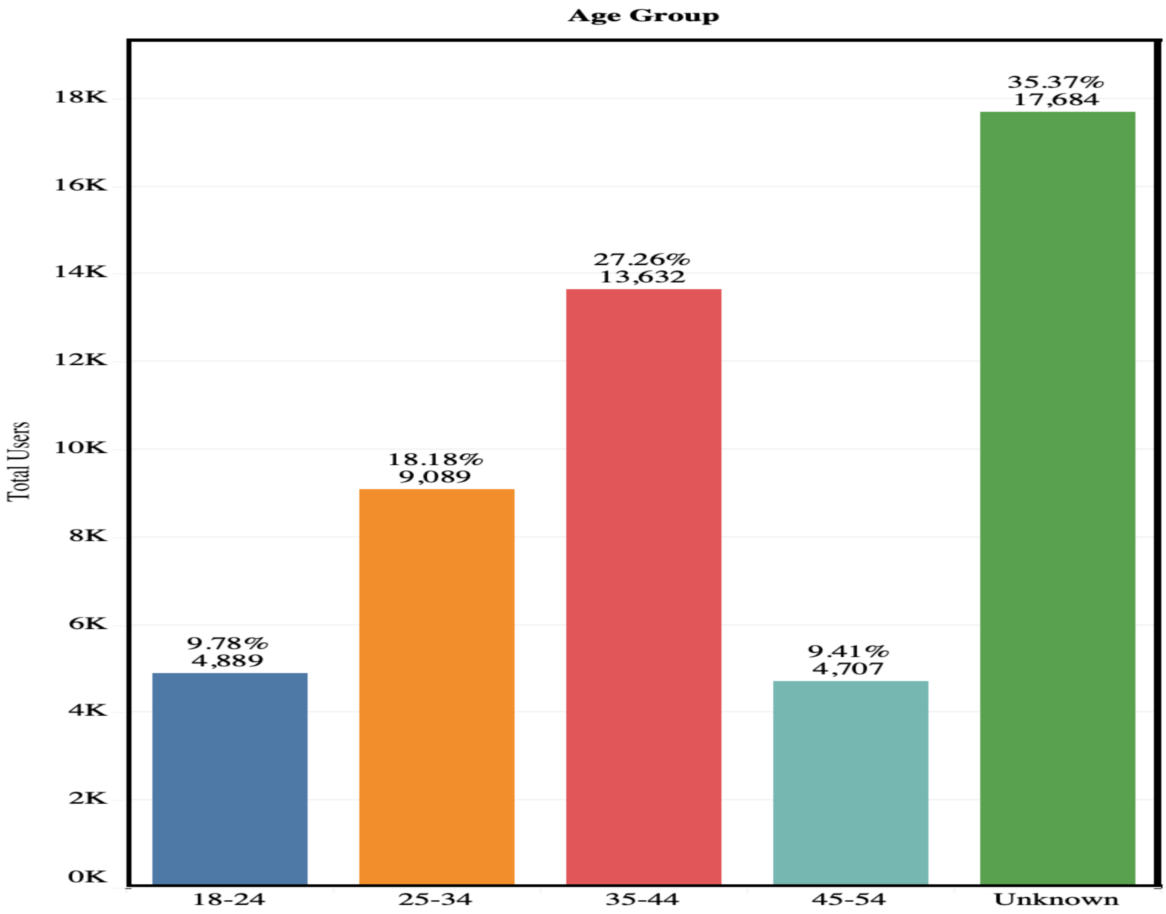


Tableau Interactive Visualization Link:

https://public.tableau.com/views/UserRideFunnelConversionRateforeachstageTopvsPrevious_17055235335770/Sheet3?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

4. Surge pricing is the practice of increasing the price of goods or services when there is the greatest demand for them. If we want to adopt a price-surging strategy, what does the distribution of ride requests look like throughout the day?

➔ **Findings:** Two peak period are seen in the visualization below:

- 1st peak: between 8AM - 9AM
- 2nd peak: between 4PM - 5PM

Recommendation: Therefore, it is recommended to apply surge pricing during these two peak periods, with careful consideration to balance supply and demand, ensuring a positive experience for both users and drivers.

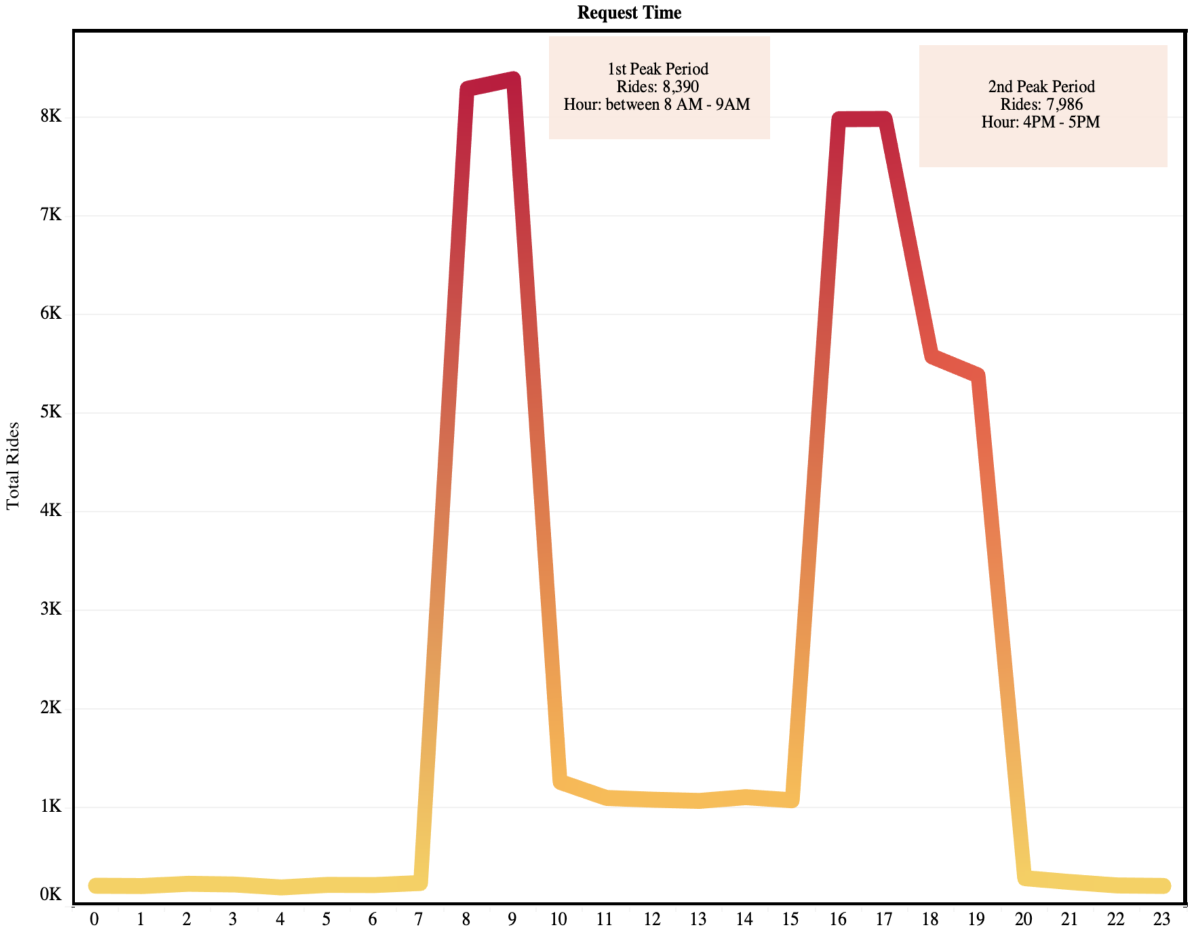


Tableau Interactive Visualization Link:

https://public.tableau.com/views/UserRideFunnelConversionRateforeachstageTopvsPrevious_17055235335770/Sheet4?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

5. What part of our funnel has the lowest conversion rate? What can we do to improve this part of the funnel?

➔ **Findings:** From the below visualization, the "ride completed" stage has the lowest conversion rate of 50.77% calculated by Percent of Previous method and 26.40% Conversion rate calculated by Percent of Top method.

Recommendation: To address the challenge of the lowest conversion rate at the "ride completed" stage, consider implementing the following strategies:

- Offering Incentives for Ride Completion: Provide users with incentives, discounts, or loyalty rewards for completing their rides. This can encourage users to finalize their journey and contribute to higher conversion rates.
- Addressing Wait Time Concerns: Mitigate wait time concerns by improving ride ETA accuracy and implementing proactive communication. Keeping users informed about their ride's progress can enhance their experience and reduce uncertainties.
- Regularly Assessing Competitive Offerings: Stay competitive by regularly assessing and adapting to competitor offerings. Understanding market trends and user preferences allows Metrocar to enhance its services, making it a more attractive option for users.

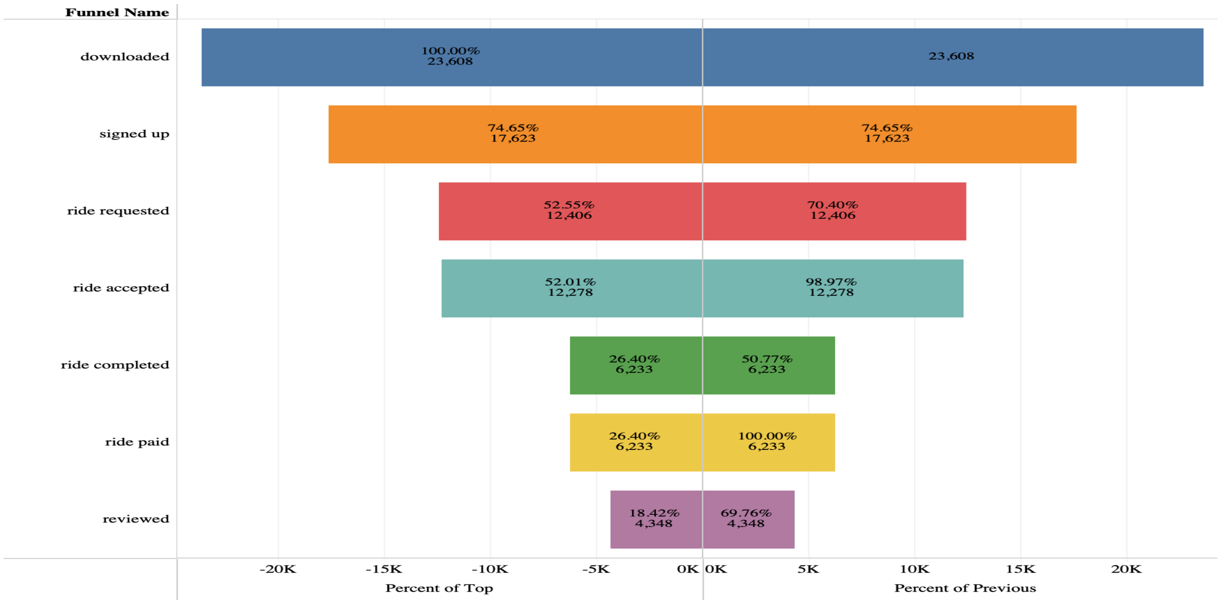
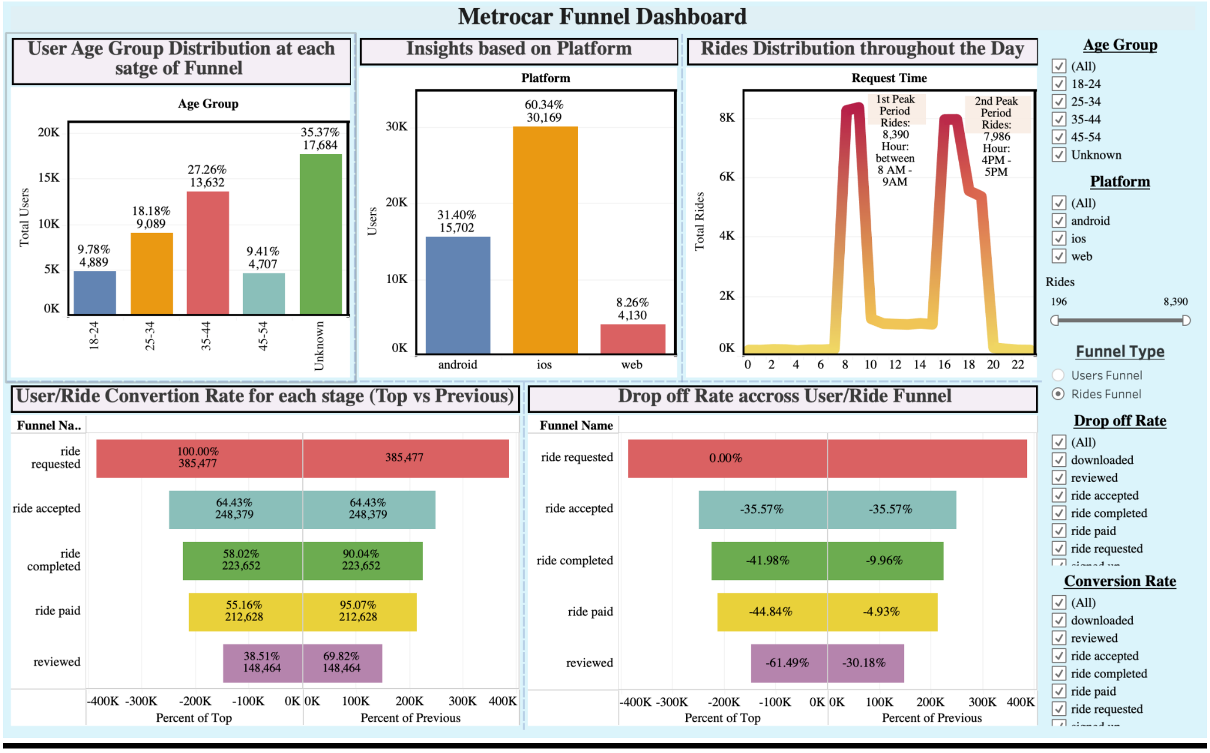


Tableau Interactive visualization link:

https://public.tableau.com/views/UserRideFunnelConversionRateforeachstageTopvsPrevious/Sheet1?:language=en-GB&:display_count=n&:origin=viz_share_link

Metrocar Funnel Interactive Dashboard:



Age Group
☑ (All)
☑ 18-24
☑ 25-34
☑ 35-44
☑ 45-54
☑ Unknown

Platform
☑ (All)
☑ android
☑ ios
☑ web

Rides
196 8,390

Funnel Type
○ Users Funnel
● Rides Funnel

Drop off Rate
☑ (All)
☑ downloaded
☑ reviewed
☑ ride accepted
☑ ride completed
☑ ride paid
☑ ride requested

Conversion Rate
☑ (All)
☑ downloaded
☑ reviewed
☑ ride accepted
☑ ride completed
☑ ride paid
☑ ride requested

Tableau Interactive Dashboard visualization link:

https://public.tableau.com/views/UserRideFunnelConversionRateforeachstageTopvsPrevious_17055235335770/Dashboard2?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

APPENDIX:

SQL Queries:

- **User Funnel Query**: includes total users at each funnel step, Total users drop count from previous funnel stage and drop percentage.

```
 WITH

downloads AS (

  SELECT 1 AS step, 'downloaded' AS funnel_step,
COUNT(app_download_key) AS users

  FROM app_downloads),


signups AS (

  SELECT 2 AS step, 'signed up' AS funnel_step, COUNT(DISTINCT
user_id) AS users

  FROM signups),


ride_requesters AS (
  SELECT 3 AS step, 'ride requested' AS funnel_step, COUNT(DISTINCT user_id) AS users
  FROM ride_requests),

ride_accepted AS (
  SELECT 4 AS step, 'ride accepted' AS funnel_step, COUNT(DISTINCT user_id) AS users
  FROM ride_requests
  WHERE accept_ts IS NOT NULL),

completed_ride AS (
  SELECT 5 AS step, 'ride completed' AS funnel_step, COUNT(DISTINCT user_id) AS users
  FROM ride_requests AS rr
  WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL),

successful_payment AS (
  SELECT 6 AS step, 'ride paid' AS funnel_step, COUNT(DISTINCT user_id) AS users
  FROM ride_requests AS rr
  INNER JOIN transactions AS ts
  ON rr.ride_id = ts.ride_id
  WHERE ts.charge_status = 'Approved'),

reviewed_ride AS (
  SELECT 7 AS step, 'reviewed' AS funnel_step, COUNT(DISTINCT user_id) AS users
  FROM reviews),

funnel AS (
  SELECT *
  FROM downloads
  UNION ALL
  SELECT *
  FROM signups
  UNION ALL
  SELECT *
  FROM ride_requesters
  UNION ALL
  SELECT *
  FROM ride_accepted
  UNION ALL
  SELECT *
  FROM completed_ride
  UNION ALL
  SELECT *
  FROM successful_payment
  UNION ALL
  SELECT *
  FROM reviewed_ride
```

```
  ORDER BY step, users DESC)

SELECT
  step AS funnel_step,
  funnel_step AS funnel_name,
  users AS Users,
  COALESCE((LAG(users, 1) OVER(ORDER BY step, users DESC)-users), 0) AS Drop,
  COALESCE(ROUND(((LAG(users, 1) OVER(ORDER BY step, users DESC)-users)::numeric/LAG(users, 1) OVER(ORDER BY step, users DESC)::numeric)*100, 2), 0) AS Drop_Percentage
FROM funnel;
```

- **Rides Funnel Query :** includes total rides at each funnel step, Total rides drop count from previous funnel stage and drop percentage.

```
WITH
ride_requesters AS (
  SELECT 1 AS funnel_step, 'ride requested' AS funnel, COUNT(ride_id) AS rides
  FROM ride_requests),

ride_accepted AS (
  SELECT 2 AS funnel_step, 'ride accepted' AS funnel, COUNT(ride_id) AS rides
  FROM ride_requests
  WHERE accept_ts IS NOT NULL),

completed_ride AS (
  SELECT 3 AS funnel_step, 'ride completed' AS funnel, COUNT(rr.ride_id) AS rides
  FROM ride_requests AS rr
  INNER JOIN transactions AS ts
  ON rr.ride_id = ts.ride_id
  WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL),

successful_payment AS (
  SELECT 4 AS funnel_step, 'ride paid' AS funnel, COUNT(rr.ride_id) AS rides
  FROM ride_requests AS rr
  INNER JOIN transactions AS ts
  ON rr.ride_id = ts.ride_id
  WHERE ts.charge_status = 'Approved'),

reviewed_ride AS (
  SELECT 5 AS funnel_step, 'ride reviewed' AS funnel, COUNT(rv.ride_id) AS rides
  FROM ride_requests AS rr
  INNER JOIN transactions AS ts ON rr.ride_id = ts.ride_id
  INNER JOIN reviews AS rv ON ts.ride_id = rv.ride_id
  WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL AND ts.charge_status = 'Approved'),
funnel AS (
  SELECT *
  FROM ride_requesters
  UNION ALL
  SELECT *
  FROM ride_accepted
  UNION ALL
  SELECT *
  FROM completed_ride
  UNION ALL
  SELECT *
  FROM successful_payment
  UNION ALL
  SELECT *
  FROM reviewed_ride
  ORDER BY rides DESC)

SELECT
  funnel_step AS funnel_step,
  funnel AS funnel_name,
  rides AS Rides,
  COALESCE((LAG(rides, 1) OVER(ORDER BY rides DESC)-rides), 0) AS Drop,
 COALESCE(ROUND(((LAG(rides, 1) OVER(ORDER BY rides DESC)-rides)::numeric/LAG(rides, 1) OVER(ORDER BY rides DESC)::numeric)*100, 2), 0) AS Drop_Percentage

FROM funnel;
```

- **User aggregated Query for Tableau Visualization:**

```
WITH

downloads AS (

 SELECT 'downloaded' AS funnel,

   COUNT(app_download_key) AS users, ad.platform,

   COALESCE(su.age_range) AS age_group,

   CAST(download_ts AS TIME) AS download_time,

   CAST(download_ts AS DATE) AS download_date

 FROM app_downloads AS ad

 RIGHT JOIN signups AS su

  ON ad.app_download_key = su.session_id

 GROUP BY ad.platform, su.age_range, download_date, download_time),


 sign_ups AS (

 SELECT 'signed up' AS funnel,

   COUNT(user_id) AS users, ad.platform,

   COALESCE(su.age_range) AS age_group,

   CAST(download_ts AS TIME) AS download_time,

   CAST(download_ts AS DATE) AS download_date
 FROM signups AS su
   INNER JOIN app_downloads AS ad
   ON su.session_id = ad.app_download_key
 GROUP BY ad.platform, su.age_range, download_date, download_time),

ride_requested AS (
SELECT 'ride requested' AS funnel,
     COUNT(DISTINCT rr.user_id) AS users, ad.platform,
     COALESCE(su.age_range) AS age_group,
     CAST(download_ts AS TIME) AS download_time,
     CAST(download_ts AS DATE) AS download_date
  FROM ride_requests AS rr
     INNER JOIN signups AS su
     ON rr.user_id = su.user_id
     INNER JOIN app_downloads AS ad
     ON su.session_id = ad.app_download_key
  GROUP BY ad.platform, su.age_range, download_date, download_time),

ride_accepted AS (
  SELECT 'ride accepted' AS funnel,
     COUNT(DISTINCT rr.user_id) AS users, ad.platform,
     COALESCE(su.age_range) AS age_group,
     CAST(download_ts AS TIME) AS download_time,
     CAST(download_ts AS DATE) AS download_date
  FROM ride_requests AS rr
    INNER JOIN signups AS su
    ON su.user_id = rr.user_id
    INNER JOIN app_downloads AS ad
    ON su.session_id = ad.app_download_key
  WHERE accept_ts IS NOT NULL
  GROUP BY ad.platform, su.age_range, download_date, download_time ),

ride_completed AS (
  SELECT 'ride completed' AS funnel,
    COUNT(DISTINCT rr.user_id) AS users, ad.platform,
    COALESCE(su.age_range, 'Not specified') AS age_group,
    CAST(download_ts AS TIME) AS download_time,
    CAST(download_ts AS DATE) AS download_date
  FROM ride_requests AS rr
    INNER JOIN signups AS su
    ON rr.user_id = su.user_id
    INNER JOIN app_downloads AS ad
    ON su.session_id = ad.app_download_key
    INNER JOIN transactions AS ts
    ON rr.ride_id = ts.ride_id
  WHERE accept_ts IS NOT NULL
  GROUP BY ad.platform, su.age_range, download_date, download_time),
ride_paid AS (
  SELECT 'ride paid' AS funnel,
    COUNT(DISTINCT rr.user_id) AS users, ad.platform,
    COALESCE(su.age_range) AS age_group,
    CAST(download_ts AS TIME) AS download_time,
    CAST(ad.download_ts AS DATE) AS download_date
  FROM ride_requests AS rr
    INNER JOIN signups AS su
    ON rr.user_id = su.user_id
    INNER JOIN app_downloads AS ad
    ON su.session_id = ad.app_download_key
    INNER JOIN transactions AS ts
    ON rr.ride_id = ts.ride_id
  WHERE accept_ts IS NOT NULL AND ts.charge_status = 'Approved'
  GROUP BY ad.platform, su.age_range, download_date, download_time),

ride_reviewed AS (
  SELECT 'ride reviewed' AS funnel,
    COUNT(DISTINCT rv.user_id) AS users, ad.platform,
    COALESCE(su.age_range) AS age_group,
    CAST(download_ts AS TIME) AS download_time,
    CAST(download_ts AS DATE) AS download_date
  FROM ride_requests AS rr
    INNER JOIN signups AS su
    ON rr.user_id = su.user_id
    INNER JOIN app_downloads AS ad
    ON su.session_id = ad.app_download_key
    INNER JOIN transactions AS ts
    ON rr.ride_id = ts.ride_id
    INNER JOIN reviews AS rv
    ON ts.ride_id = rv.ride_id
  WHERE accept_ts IS NOT NULL AND ts.charge_status = 'Approved'
  GROUP BY ad.platform, su.age_range, download_date, download_time)
```

```
SELECT *
FROM downloads
UNION ALL
SELECT *
FROM sign_ups
UNION ALL
SELECT *
FROM ride_requested
UNION ALL
SELECT *
FROM ride_accepted
UNION ALL
```

```
SELECT *
FROM ride_completed
UNION ALL
SELECT *
FROM ride_paid
UNION ALL
SELECT *
FROM ride_reviewed
```

- **Rides aggregated query for Tableau visualization:**

```
WITH
rides_requested AS (
SELECT
  'ride requested' AS funnel,
  COUNT(DISTINCT rr.ride_id) AS rides,
  ad.platform,
  COALESCE(su.age_range, 'Not specified') AS age_group,
  CAST(rr.request_ts AS TIME) AS request_time,
  CAST(rr.request_ts AS DATE) AS request_date,
  0 AS revenue
FROM
  ride_requests AS rr
  INNER JOIN signups AS su ON rr.user_id = su.user_id
  INNER JOIN app_downloads AS ad ON su.session_id = ad.app_download_key
GROUP BY
  ad.platform,
  su.age_range,
  request_date, request_time),

rides_accepted AS (
SELECT
  'ride accepted' AS funnel,
  COUNT(DISTINCT rr.ride_id) AS rides,
  ad.platform,
  COALESCE(su.age_range, 'Not specified') AS age_group,
  CAST(rr.request_ts AS TIME) AS request_time,
  CAST(rr.request_ts AS DATE) AS request_date,
  0 AS revenue
FROM
  ride_requests AS rr
  INNER JOIN signups AS su ON rr.user_id = su.user_id
  INNER JOIN app_downloads AS ad ON su.session_id = ad.app_download_key
WHERE rr.accept_ts IS NOT NULL
GROUP BY
  ad.platform,
  su.age_range,
  request_date, request_time),

rides_completed AS (
SELECT
  'ride completed' AS funnel,
  COUNT(DISTINCT rr.ride_id) AS rides,
  ad.platform,
  COALESCE(su.age_range, 'Not specified') AS age_group,
  CAST(rr.request_ts AS TIME) AS request_time,
  CAST(rr.request_ts AS DATE) AS request_date,
  0 AS revenue
FROM
  ride_requests AS rr
  INNER JOIN signups AS su ON rr.user_id = su.user_id
  INNER JOIN app_downloads AS ad ON su.session_id =
ad.app_download_key
WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL
GROUP BY
  ad.platform,
  su.age_range,
  request_date, request_time),

rides_paid AS (
SELECT
```

```
  'ride paid' AS funnel,
  COUNT(DISTINCT rr.ride_id) AS rides,
  ad.platform,
  COALESCE(su.age_range, 'Not specified') AS age_group,
  CAST(rr.request_ts AS TIME) AS request_time,
  CAST(rr.request_ts AS DATE) AS request_date,
  SUM(ts.purchase_amount_usd) AS revenue
FROM
  ride_requests AS rr
  INNER JOIN signups AS su ON rr.user_id = su.user_id
  INNER JOIN app_downloads AS ad ON su.session_id =
ad.app_download_key
  INNER JOIN transactions AS ts ON rr.ride_id = ts.ride_id
WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL AND
ts.charge_status = 'Approved'
GROUP BY
  ad.platform,
  su.age_range,
  request_date, request_time),


rides_reviewed AS (
SELECT
  'ride reviewed' AS funnel,
  COUNT(DISTINCT rr.ride_id) AS rides,
  ad.platform,
  COALESCE(su.age_range, 'Not specified') AS age_group,

  CAST(rr.request_ts AS TIME) AS request_time,

  CAST(rr.request_ts AS DATE) AS request_date,

  SUM(ts.purchase_amount_usd) AS revenue

FROM

  ride_requests AS rr

  INNER JOIN signups AS su ON rr.user_id = su.user_id

  INNER JOIN app_downloads AS ad ON su.session_id =
ad.app_download_key

  INNER JOIN transactions AS ts ON rr.ride_id = ts.ride_id

  INNER JOIN reviews AS rv ON ts.ride_id = rv.ride_id

WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL AND
ts.charge_status = 'Approved'

GROUP BY

  ad.platform,

  su.age_range,

  request_date, request_time)


SELECT *

FROM rides_requested

UNION ALL

SELECT *

FROM rides_accepted

UNION ALL

SELECT *
```

```
FROM rides_completed
UNION ALL
SELECT *
FROM rides_paid
UNION ALL
SELECT *
FROM rides_reviewed
```

- **Query for Total number of app downloaded :**

  ```
  SELECT COUNT(app_download_key)
  FROM app_downloads;
  ```

- **Query for Total number of users signed up:**

  ```
  SELECT COUNT(DISTINCT session_id)
  FROM signups;
  ```

- **Query for Total Rides Requested :**

  ```
  SELECT COUNT(ride_id)
  FROM ride_requests;
  ```

- **Query for Total Rides Completed :**

```
SELECT COUNT(dropoff_ts)

FROM ride_requests;
```

- **Query for Total Unique Users:**

```
SELECT COUNT(ride_id), COUNT(DISTINCT user_id)

FROM ride_requests;
```

- **Query for Average time from ride pick-up to ride drop-off :**

```
SELECT AVG(pickup_ts-dropoff_ts) as avg_time

FROM ride_requests

WHERE pickup_ts is not null;
```

- **Query for Total rides accepted**:

```
SELECT COUNT(accept_ts) as accepted_rides

FROM ride_requests;
```

- **Query for Total Payment collected :**

```
SELECT

COUNT(charge_status) as success_payments,
SUM(purchase_amount_usd) as payments

FROM transactions

WHERE charge_status = ‘Approved’;
```

- **Query for Total ride request from each platform:**

```
SELECT platform, COUNT(request_ts)

FROM app_downloads ad

LEFT JOIN signups s ON ad.app_download_key=s.session_id

LEFT JOIN ride_requests rr on s.user_id=rr.user_id

GROUP BY platform;
```

- **Query for Drop-off from users signing up to users requesting a ride** -

```
    WITH user_dropoff AS (

SELECT

  COUNT(DISTINCT s.user_id) AS user_signup,

  COUNT(DISTINCT rr.user_id) AS user_requesting_ride

FROM app_downloads ad
```

```
  LEFT JOIN signups s ON ad.app_download_key = s.session_id
  LEFT JOIN ride_requests rr ON s.user_id = rr.user_id
)
SELECT
  user_signup - user_requesting_ride AS drop_off_count,
  ((user_signup - user_requesting_ride) / user_signup::float) * 100 AS drop_off_percentage
FROM user_dropoff;
```