Funnel Analysis of Metrocar's Customer Journey

Author: Manuk Mikayelyan

July 2023

Summary

This report presents the findings of a funnel analysis conducted on Metrocar's customer journey. The analysis aims to identify areas for improvement and optimization in the customer funnel. Key results include conversion rates at each stage of the funnel, insights on platform performance, age group performance, surge pricing opportunities, and user experience analysis. Based on these results, recommendations are provided to address the identified areas for improvement.

Context

Metrocar is a ride-sharing app that connects riders with drivers through a mobile application. The customer funnel consists of several stages, including app download, signup, ride request, driver acceptance, ride completion, payment, and review. The analysis is based on a dataset containing information on app downloads, user signups, ride requests, transactions, and driver reviews.

We will analyze the data and make recommendations based on the following business questions:

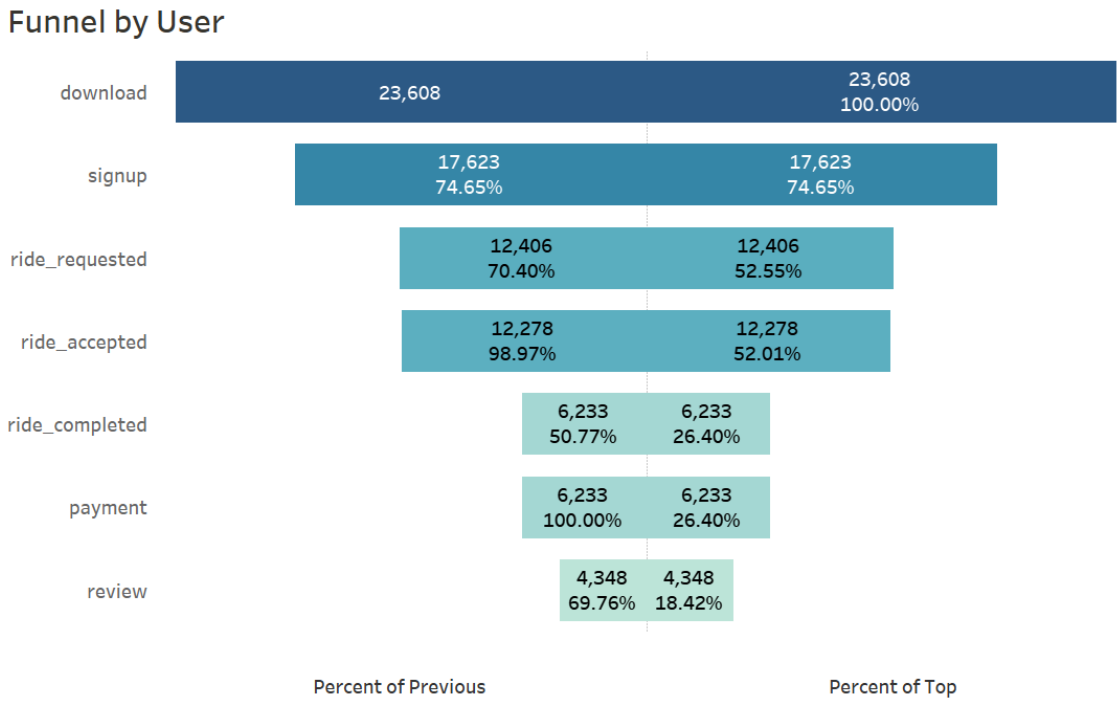
- What steps of the funnel should we research and improve? Are there any specific drop-off points preventing users from completing their first ride?
- Metrocar currently supports 3 different platforms: ios, android, and web. To recommend where to focus our marketing budget for the upcoming year, what 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?

Results

Funnel Conversion Rate Analysis

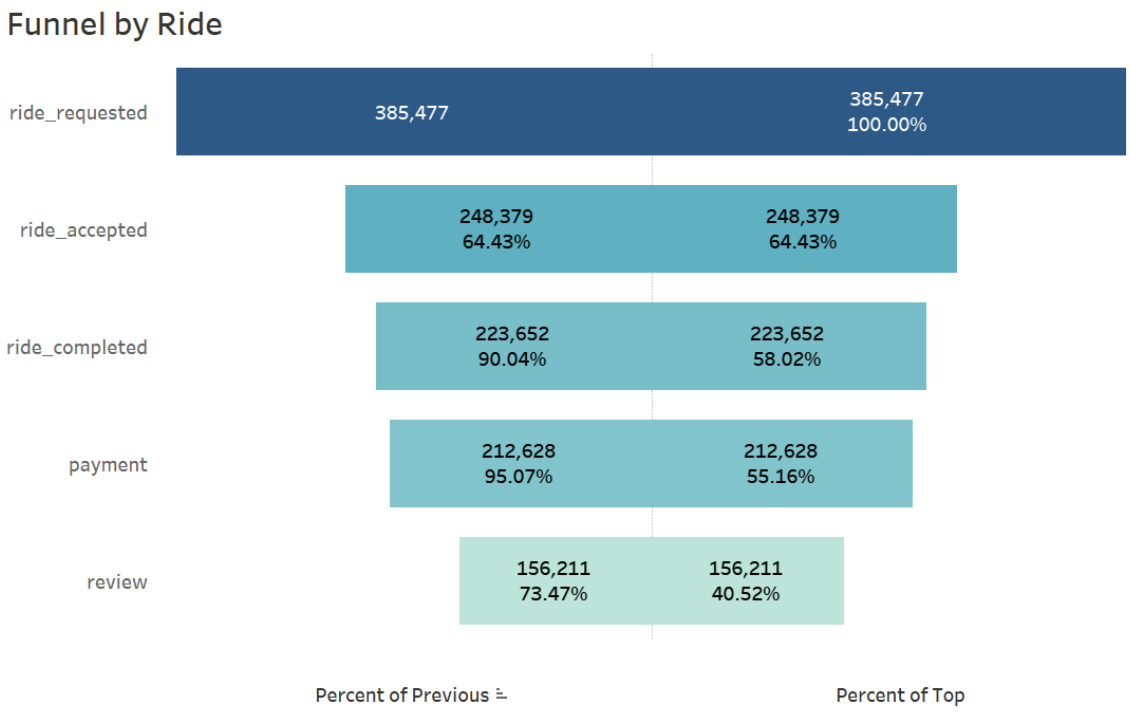
The analysis of Metrocar's customer funnel revealed important insights when considering both the perspective of users and rides. Conversion rates by users provide a holistic understanding of user behavior and engagement throughout the funnel, while conversion rates by rides offer insights into the operational aspects of the service. Here are the key findings (see Figure 1 and Figure 2):

Figure 1.



- App Download to Signup: The conversion rate is approximately 74.65%. This indicates that around 25.35% of users who download the app do not proceed to sign up.
- Signup to Request Ride: The conversion rate is approximately 70.40%. This suggests that about 29.60% of users who sign up do not proceed to request a ride
- Request Ride to Driver Acceptance: The conversion rate is quite high at 98.97%, indicating that a majority of ride requests are accepted by drivers. This step seems to be performing well.
- Driver Acceptance to Ride Completed: The conversion rate drops significantly to approximately 50.77%. This suggests that there may be issues during the ride that cause users to cancel or abandon the ride.
- The conversion rate from Ride Completed to Payment is 100%, indicating that all rides that were completed resulted in a successful payment. This suggests that the payment process is functioning well, and there are no significant issues or barriers in completing the payment after the ride.
- The conversion rate from Payment to Review is approximately 69.76%. This means that around 30.24% of users who made a payment did not leave a review. While this drop-off is not as severe as in some previous stages, it still indicates that there may be room for improvement in encouraging users to provide feedback and rate their ride experience.

Figure 2.

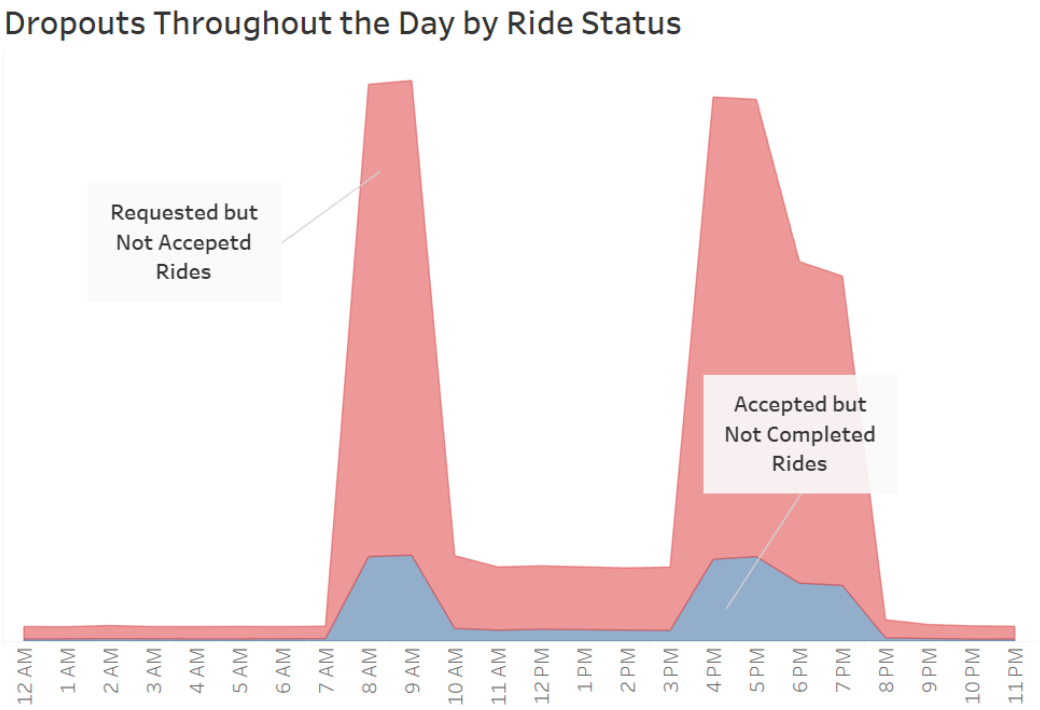


- Request Ride to Driver Acceptance: The number of ride requests is 385,477, and the number of driver acceptances is 248,379. The acceptance rate is approximately 64.43%. This indicates that around 35.57% of ride requests do not result in a driver accepting the ride.
- Driver Acceptance to Ride Completed: The number of driver acceptances is 248,379, and the number of completed rides is 223,652. The completion rate is approximately 90.04%. This suggests that around 9.96% of rides do not reach the completed state after driver acceptance.
- Ride Completed to Payment: The number of completed rides is 223,652, and the number of payments is 212,628. The payment rate is approximately 95.07%. This indicates that around 4.92% of completed rides do not result in a successful payment.
- Payment to Review: The number of payments is 212,628, and the number of reviews is 156,211. The review participation rate is approximately 73.47%. This suggests that around 26.53% of users who made a payment did not leave a review.

The differences between the two calculations indicate variations in the conversion rates for users and rides. Users may exhibit drop-offs at certain stages, but overall, ride-related actions show a high completion rate. Analyzing both conversion rates provides a comprehensive understanding of the user journey and operational aspects, allowing for the prioritization of improvements that address user experience barriers and operational inefficiencies.

In the user funnel, the conversion rate from ride request to ride acceptance is 98.97%, indicating a positive user experience and effective connection with drivers. However, in the ride funnel, the conversion rate from ride request to ride acceptance is 64.43%, suggesting challenges in matching ride requests with available drivers. This lower conversion rate in the ride funnel can be attributed to factors such as driver availability and market conditions (Figure 3). Fluctuations in driver availability may result in unaccepted ride requests when demand exceeds supply. Improving driver availability and optimizing the matching process are essential to enhance the conversion rate in the ride funnel.

Figure 3.



Funnel Drop-offs Analysis by Platform and User Group

The user and ride drop-off analysis by platform and age groups provided insightful findings, particularly regarding the download to signup stage in the user funnel and the transaction approval rate.

Across platforms and age groups, the analysis indicates that drop-off rates are generally consistent (Figures 4, 5, 6, 7). This suggests that users and riders, regardless of platform preference or age group, face similar challenges or encounter similar barriers at various stages of the customer journey. This finding highlights the importance of addressing these common pain points to improve overall conversion rates and user satisfaction.

However, there is one notable exception in the user funnel (Figure 4). For the "Not Defined" age group, the drop-off rate from app download to signup is 53%, whereas for the other age groups, it is 0%. This significant difference suggests that there might be specific factors impacting this particular age group, leading to a higher dropout rate during the signup process. Further investigation is necessary to understand the reasons behind this discrepancy and to implement measures to reduce the dropout rate for the "Not Defined" age group. It is crucial to analyze user behavior, preferences, and potential obstacles specific to this age group to tailor the onboarding experience and improve the conversion rate.

Figure 4.

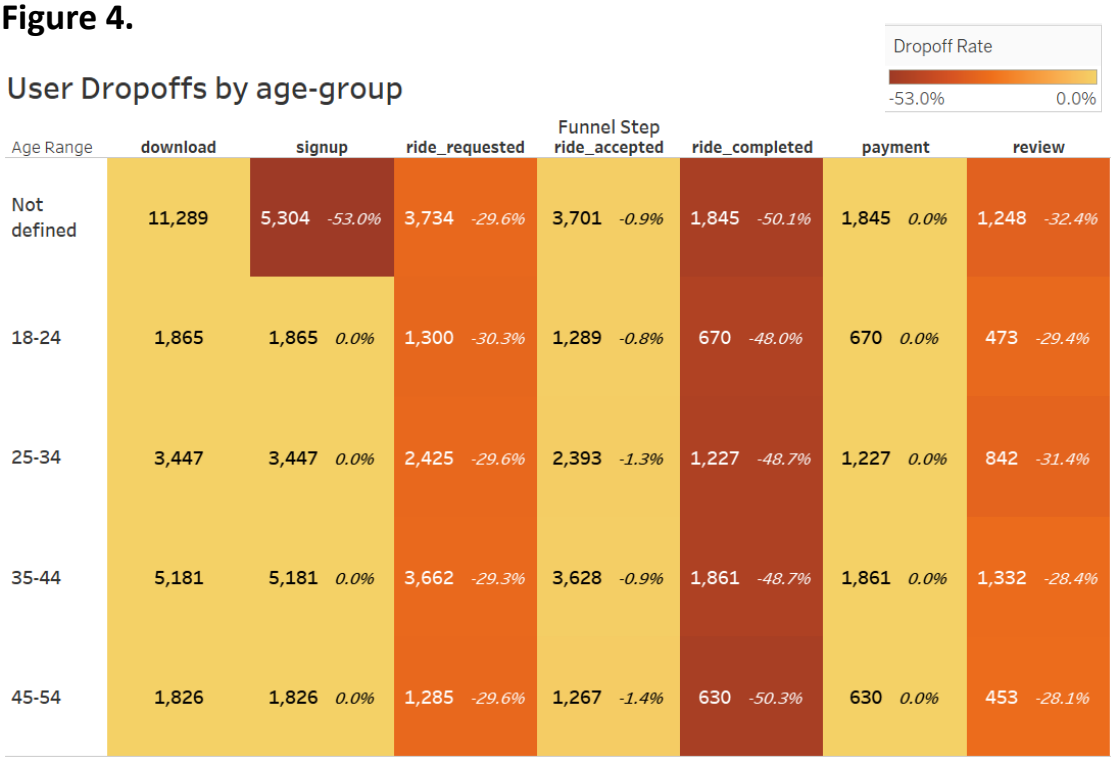


Figure 5.

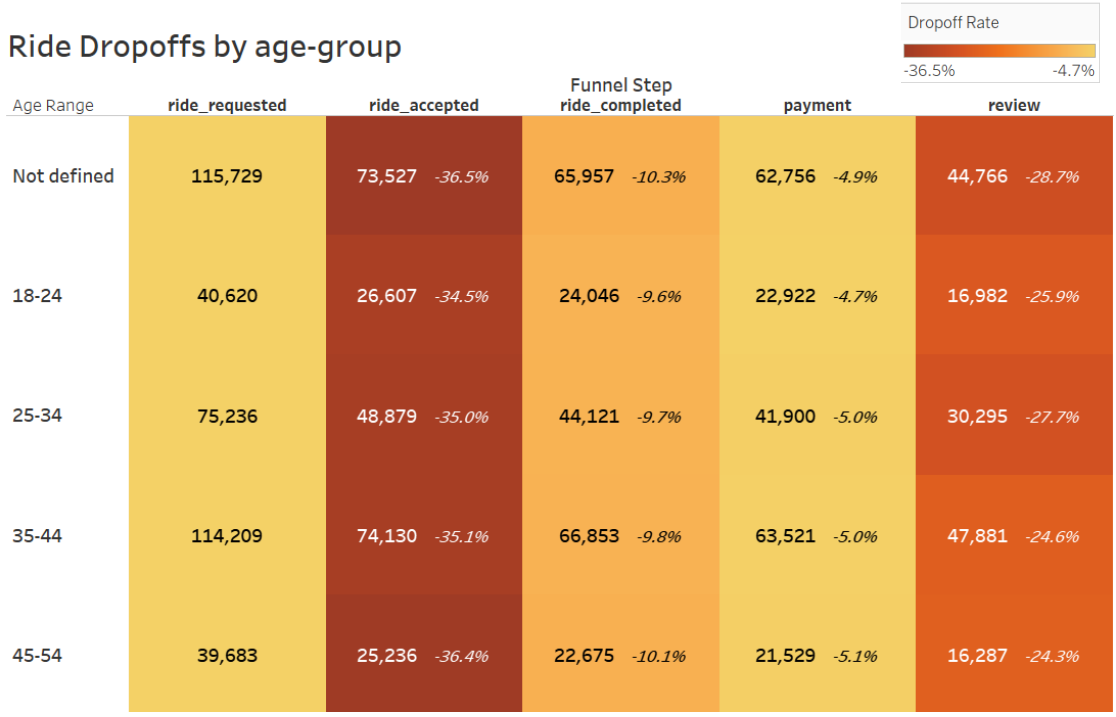


Figure 6.

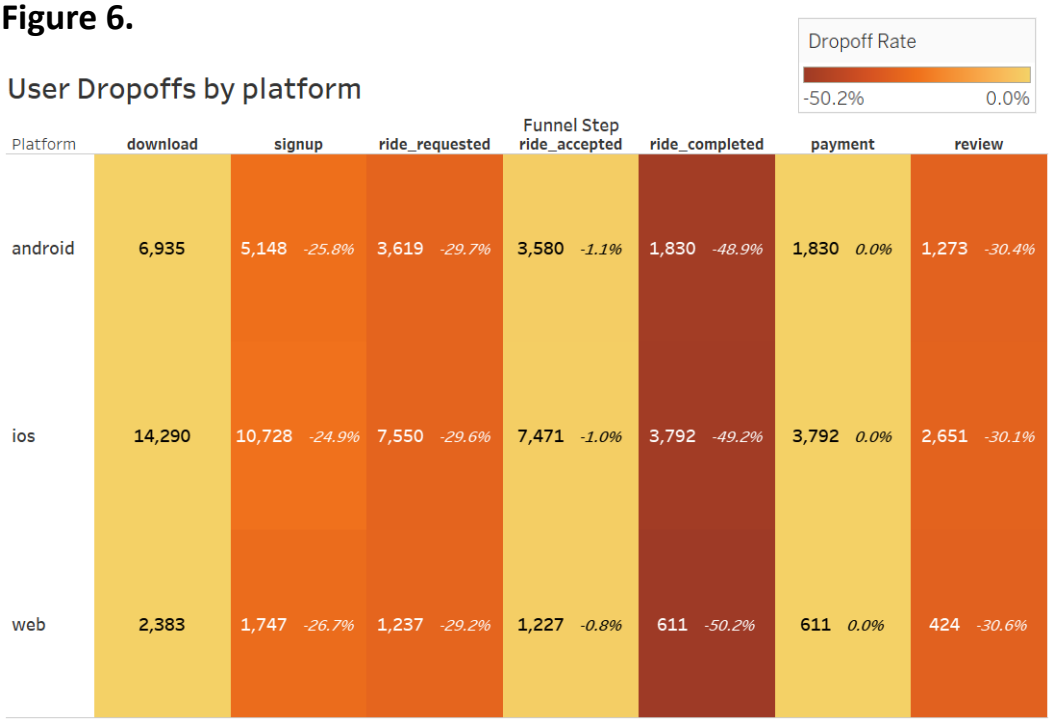
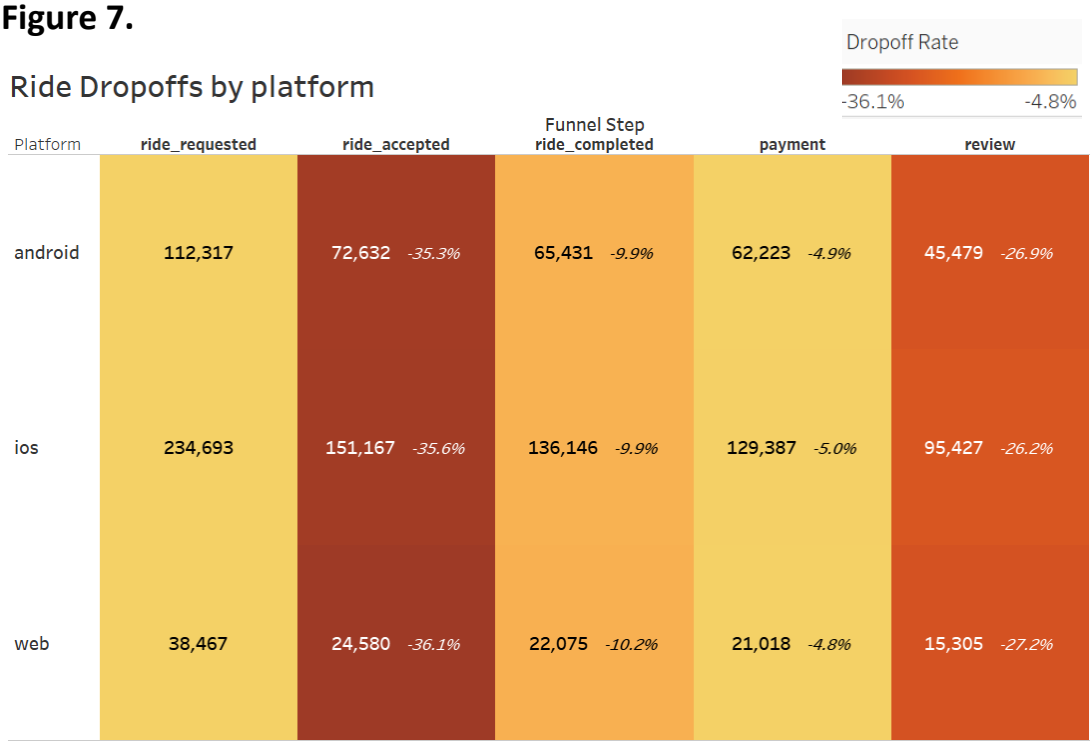
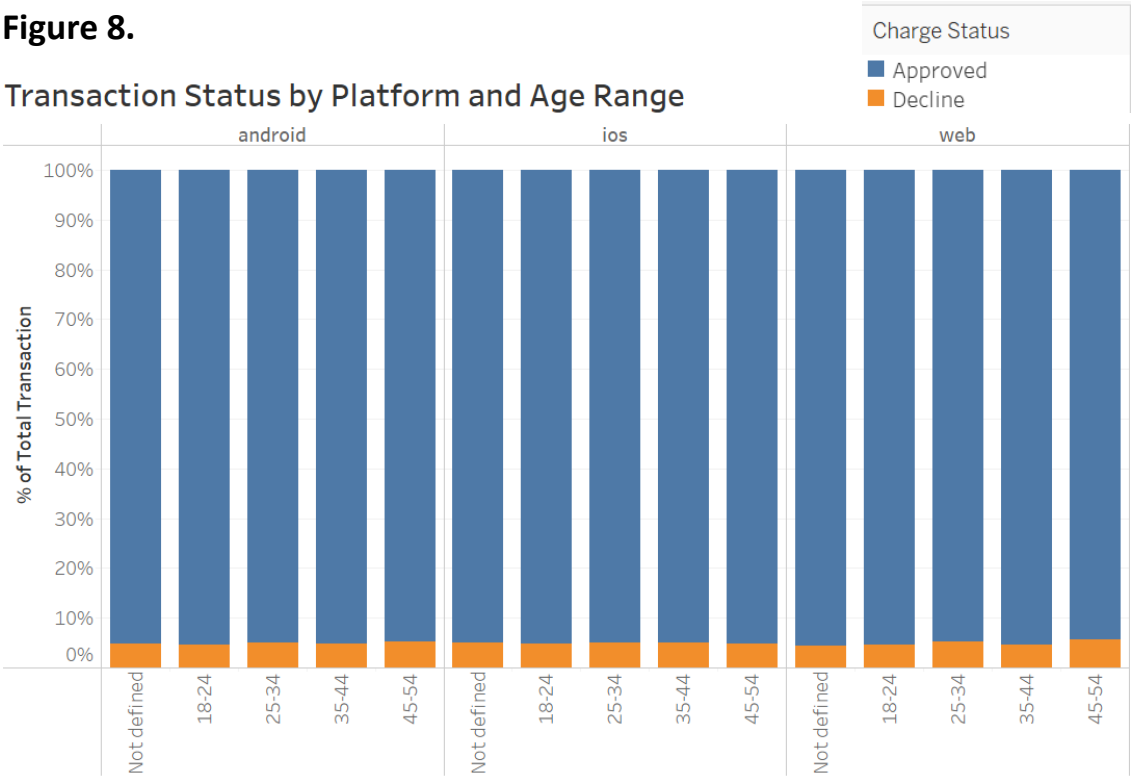


Figure 7.



Another observation is that approximately 5% of transactions across platforms and age groups did not approve (Figure 8). This consistency suggests that the transaction approval issue is not heavily influenced by platform or age. Further analysis can be conducted to identify the reasons behind these unapproved transactions, such as payment failures, technical issues, or user errors. Addressing these transaction approval issues can enhance the overall payment process, minimize disruptions, and improve the user experience.

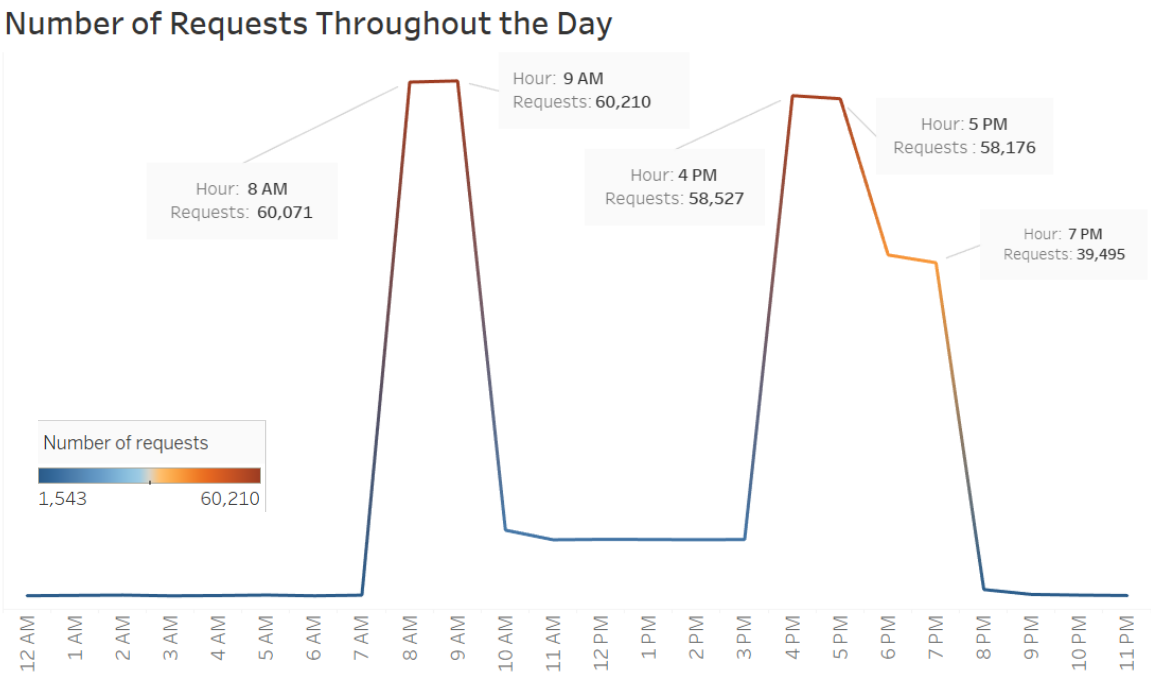
Figure 8.



Surge pricing

As showed in Figure 9 the distribution of ride requests throughout the day indicates high demand from users during the following time periods: 8 to 9 am and 4 pm to 7 pm. This information can be valuable for considering the adoption of a price-surging strategy.

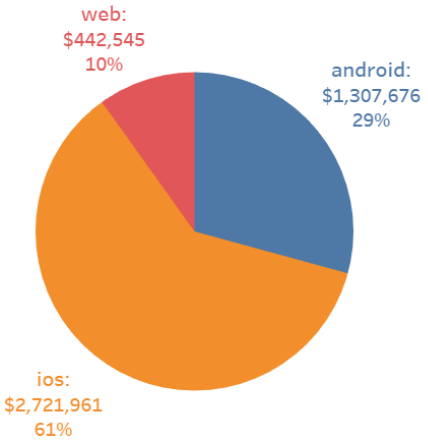
Figure 9.



Transactions by user demographics and platform type

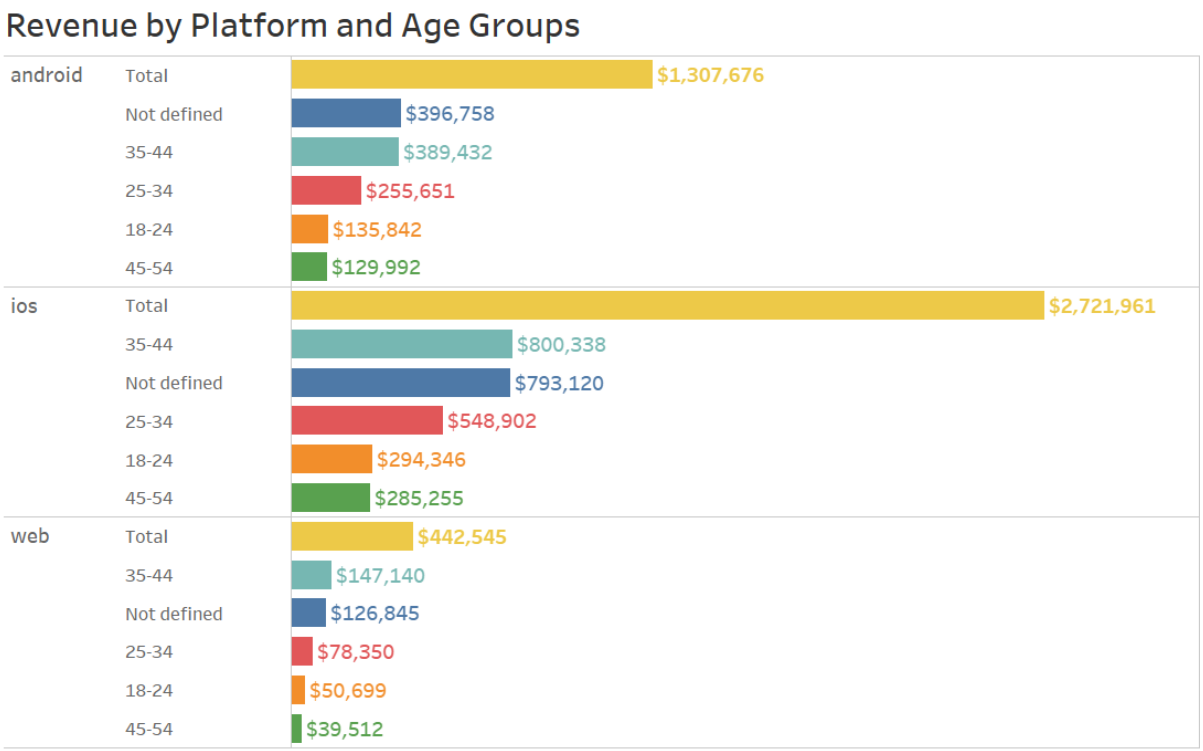
As showed in the Figure 10 below the performance of Metrocar's platforms and their contribution to revenue can help guide the allocation of the marketing budget for the upcoming year. The revenue breakdown by platform indicates that iOS contributes the highest share at 61% (2.7 million), followed by Android at 29% (1.3 million), and the web platform at 10% (442K). This suggests that iOS users generate the most revenue for Metrocar, while Android and the web platform also play significant roles.

Figure 10.



The clustered bar chart below (Figure 11) demonstrates that the 25-34 and 35-44 age groups contribute the most across all platforms in terms of revenue. This information highlights the importance of targeting and tailoring marketing efforts to these age segments for all platforms.

Figure 11.

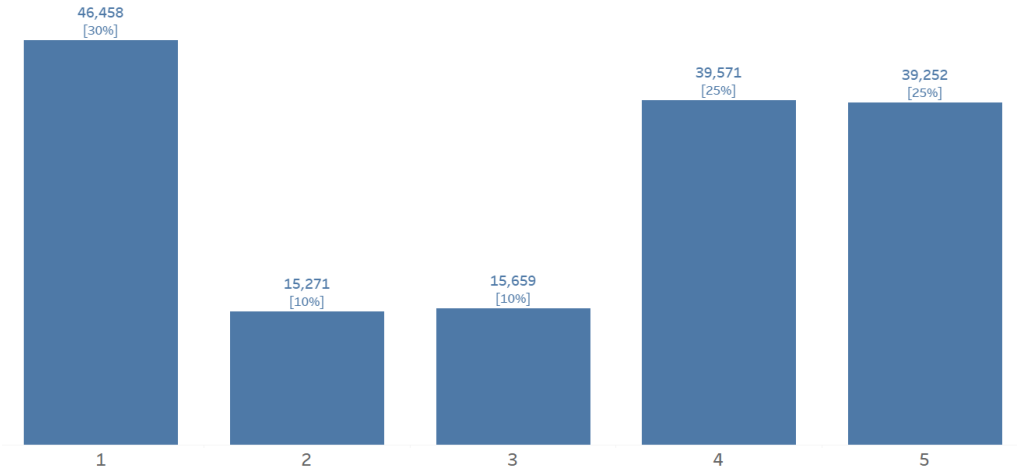


It's worth noting that there is a significant contribution to revenue from an age group labeled as 'Not Defined.' However, since the age group is unknown, it becomes challenging to specifically target this segment. Further analysis could be conducted to identify the reasons behind this unidentified age group and find ways to gather accurate age data for these users. It is important to determine the actual age distribution within this group to effectively incorporate them into marketing strategies.

User Experience Analysis

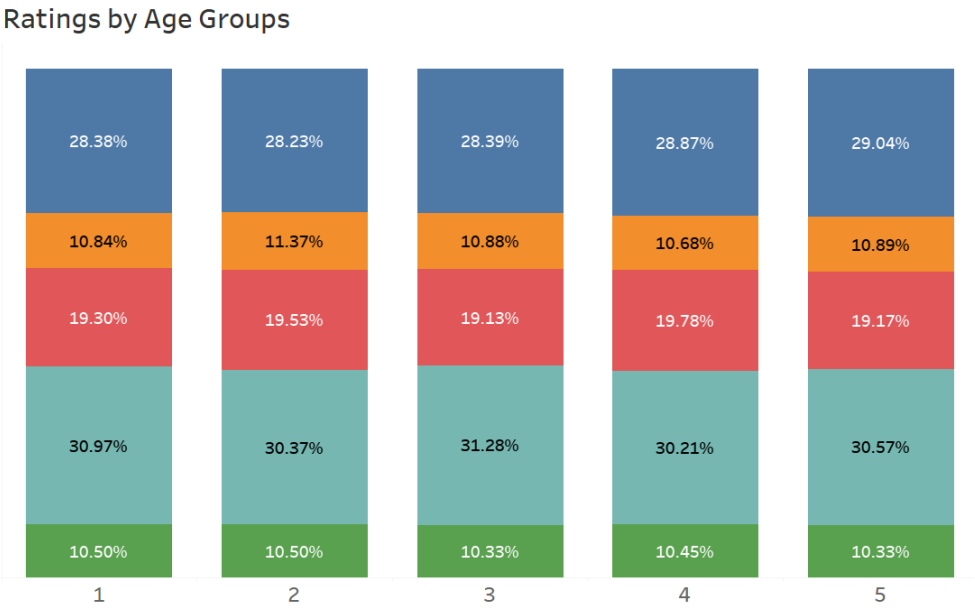
A user experience analysis was conducted to gain insights into user reviews and ratings. The analysis reveals the distribution of user ratings on a scale of 1 to 5 (Figure 12), with the following distribution: 30% of users gave a rating of 1, 10% rated 2, 10% rated 3, 25% rated 4, and 25% rated 5. This distribution indicates a significant portion of users expressing lower satisfaction with their ride experiences, as 30% of users gave the lowest rating of 1.

Figure 12.



Furthermore, a stacked bar chart was created to examine the distribution of age groups within each rating group. Surprisingly, the analysis demonstrates that the age group distribution appears to be equal across all rating groups. This suggests that user satisfaction levels do not significantly vary based on age.

Figure 13.



The word cloud analysis of user reviews with ratings of 1 or 2 reveals several common themes and issues.

Figure 14.

Word Cloud Analysis of User Reviews (Ratings 1-2)

The top five problems highlighted include:

- Reckless driving: Users expressed concerns about drivers exceeding the speed limit and driving in a reckless manner, leading to feelings of discomfort and safety issues.
- Late arrivals and cancellations: Many users reported instances of drivers arriving late or canceling rides at the last minute, resulting in inconvenience and being left stranded in unfamiliar locations.
- Service quality and dissatisfaction: Users frequently mentioned experiences described as "horrible," "terrible," and "worst," indicating overall dissatisfaction with the service provided by Metrocar.
- Issues with communication and responsiveness: Users cited difficulties in communicating with drivers and encountering unresponsive or unprofessional behavior, leading to frustration and poor customer service experiences.
- Vehicle conditions and cleanliness: Some users reported encountering unpleasant smells, dirty conditions, and unsatisfactory cleanliness standards in the vehicles, contributing to a negative perception of the service.

These top-rated problems highlight the need for Metrocar to address driver behavior, improve punctuality and reliability, prioritize customer satisfaction, enhance communication channels, and ensure vehicle cleanliness. By focusing on resolving these key issues, Metrocar can enhance the user experience, increase customer loyalty, and improve the overall reputation of their service.

Recommendation

Based on the funnel analysis, the following recommendations are provided:

1. Improve Ride Completion Rate: Identify and address any issues during the ride that may lead to users canceling or abandoning their rides. Enhancing the ride experience can increase the conversion rate from Driver Acceptance to Ride Completed.
2. Enhance Review Participation: Encourage users to provide feedback and rate their ride experience to improve the conversion rate from Payment to Review. Implement strategies such as in-app prompts or incentives to motivate users to leave reviews.
3. Investigate "Not Defined" Age Group Drop-off Rate: Further investigate the factors influencing the high drop-off rate (53%) from App Download to Signup for the "Not Defined" age group. Analyze user behavior, preferences, and potential obstacles specific to this age group to tailor the onboarding experience and improve the conversion rate.

4. Address Transaction Approval Issues: Conduct a detailed analysis to understand the reasons behind unapproved transactions. Identify and resolve payment failures, technical issues, or user errors to improve the transaction approval rate.
5. Consider Surge Pricing Strategy: Utilize the insights on high demand periods (8-9 am and 4-7 pm) to adopt a price-surging strategy. Implement surge pricing during peak hours to optimize driver availability and increase service availability, thereby maximizing revenue potential.
6. Request Data on "Not Defined" Age Group: As the "Not Defined" age group contributes significantly to revenue, it is crucial to gather accurate age data for these users. Implement measures to collect age information from users within this group, either through account settings or optional surveys. This data will provide valuable insights for targeted marketing efforts and personalized user experiences.
7. Collect Data on Driver Experience: In addition to analyzing the user experience, it is equally important to gather feedback and insights on the driver experience. Implement a mechanism for drivers to provide feedback on their interactions with riders, ride conditions, and overall satisfaction. This data can help identify areas of improvement in driver training, support, and operational processes, leading to a better overall service for both drivers and riders.
8. Enhance Service Quality and Customer Satisfaction: Based on the word cloud analysis, focus on improving overall service quality to address common complaints such as late arrivals, communication issues, and vehicle cleanliness. Implement measures to enhance driver professionalism, improve punctuality, and ensure high standards of vehicle cleanliness to meet user expectations and enhance customer satisfaction

Appendix

1. Link to Tableau dashboard – https://public.tableau.com/views/MetrocarFunnel_1_MM/Story1?:language=en-US&:display_count=n&:origin=viz_share_link

2. Link to video presentation - https://drive.google.com/file/d/1lnpC-A7YjceNLGgMglM1ZQEhPOTSTnpo/view?usp=sharing

3. Dataset 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, cancelled
- 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

5. Data Exploration in SQL

```
-- Explore the Metrocar Data with SQL
--1. How many times was the app downloaded?
-- 23608
SELECT
  COUNT(DISTINCT app_download_key)
FROM
  app_downloads;
  --2. How many users signed up on the app?
-- 17623
SELECT
  COUNT(DISTINCT user_id)
FROM
  signups;
--3. How many rides were requested through the app?
-- 385477
SELECT
  COUNT(DISTINCT ride_id)
FROM
  ride_requests;
 --4. How many rides were requested and completed through the app?
 -- requested - 385477, completed - 223652
 SELECT
  COUNT(DISTINCT ride_id) ride_requested,
  (
    SELECT
      COUNT(DISTINCT ride_id) rides_completed
    FROM
      ride_requests
    WHERE
      dropoff_ts IS NOT NULL
  )
FROM
  ride_requests;
--5. How many rides were requested and how many unique users requested a ride?
```

```
-- rides - 385477, uses - 12406
SELECT
  COUNT(DISTINCT ride_id) rides_requested,
  COUNT(DISTINCT user_id) unique_users
FROM
  ride_requests;
--6. What is the average time of a ride from pick up to drop off?
-- 52 min and 36.73 sec
SELECT
  AVG(dropoff_ts - pickup_ts) AS average_ride_duration
FROM
  ride_requests;
--7. How many rides were accepted by a driver?
-- 248379
SELECT
  COUNT(ride_id)
FROM
  ride_requests
WHERE
  accept_ts IS NOT NULL;
--8. How many rides did we successfully collect payments and how much was collected?
-- 212628 and 4251667.61
SELECT DISTINCT
  charge_status,
  COUNT(*),
  SUM(purchase_amount_usd)
FROM
  transactions
GROUP BY
  1;
--9. How many ride requests happened on each platform?
-- android - 112317, ios - 234693, web - 38467
SELECT DISTINCT
  a.platform,
  COUNT(ride_id)
FROM
  ride_requests r
  LEFT JOIN
    signups s
    ON r.user_id = s.user_id
  LEFT JOIN
    app_downloads a
    ON s.session_id = a.app_download_key
GROUP BY
  1;
--10. What is the drop-off from users signing up to users requesting a ride?
-- 29.60%
WITH cte_signups AS
(
  SELECT DISTINCT
    user_id
  FROM
    signups
)
,
```

```
cte_ride_request AS
(
  SELECT DISTINCT
    s.user_id
  FROM
    cte_signups s
    INNER JOIN
      ride_requests r
      ON s.user_id = r.user_id
)
,
cte_funnel AS
(
  SELECT
    '1' as n,
                    'signups' AS step,
    COUNT(user_id) AS users
  FROM
    cte_signups
  UNION
  SELECT
    '2' as n,
                    'ride_request' AS step,
    COUNT(user_id) AS step
  FROM
    cte_ride_request
  ORDER BY
    n
)

SELECT
  *,
  lag(users, 1) OVER() AS lag,
  CAST((lag(users, 1) OVER()::FLOAT - users)*100 / lag(users, 1) OVER() AS NUMERIC(5, 2)) dropp_off
FROM
  cte_funnel;

-- Developing Metrocar Funnel Metrics
--1. How many unique users requested a ride through the Metrocar app?
-- 12406
SELECT
  COUNT( DISTINCT
  CASE
    WHEN
      request_ts IS NOT NULL
    THEN
      user_id
  END
)
FROM
  ride_requests;

--2. How many unique users completed a ride through the Metrocar app?
-- 6233
SELECT
```

```
  COUNT( DISTINCT
  CASE
    WHEN
      dropoff_ts IS NOT NULL
    THEN
      user_id
  END
)
FROM
  ride_requests;

--3. Of the users that signed up on the app, what percentage these users requested a ride?
-- 70.4%
WITH ride_requested AS
(
  SELECT DISTINCT
    user_id
  FROM
    ride_requests
  WHERE
    request_ts IS NOT NULL
)
,
total AS
(
  SELECT
    s.user_id total_users,
    r.user_id ride_request_users
  FROM
    signups s
    LEFT JOIN
      ride_requested r
      ON s.user_id = r.user_id
)
,
funnel AS
(
  SELECT
    '1' as n,
                    'signups' AS step,
    COUNT(*) AS users
  FROM
    total
  UNION
  SELECT
    '2' as n,
                    'ride_request' AS step,
    COUNT(ride_request_users) AS users
  FROM
    total
  ORDER BY
    n
)
SELECT
  *,
```

```
  lag(users, 1) OVER(),
  CAST(users::FLOAT*100 / lag(users, 1) OVER() AS NUMERIC(5, 1)) converstion_rate
FROM
  funnel;
--4. Of the users that signed up on the app, what percentage these users completed a ride?
-- 35.4%
WITH cte_signups AS
(
  SELECT DISTINCT
    user_id AS signup
  FROM
    signups
)
,
cte_request AS
(
  SELECT DISTINCT
    signup AS requested
  FROM
    cte_signups AS s
    LEFT JOIN
      ride_requests r
      ON s.signup = r.user_id
  WHERE
    request_ts IS NOT NULL
)
,
cte_completed AS
(
  SELECT DISTINCT
    requested AS completed
  FROM
    cte_request AS cr
    LEFT JOIN
      ride_requests AS rr
      ON cr.requested = rr.user_id
  WHERE
    rr.dropoff_ts IS NOT NULL
)
,
funnel AS
(
  SELECT
    '1' AS n,
    'signup' AS step,
    COUNT(*) AS users
  FROM
    cte_signups
  UNION
  SELECT
    '2' AS n,
    'request' AS step,
    COUNT(*) AS users
  FROM
    cte_request
```

```
  UNION
  SELECT
    '3' AS n,
    'completed' AS step,
    COUNT(*) AS users
  FROM
    cte_completed
  ORDER BY
    n
)
SELECT
  *,
  lag(users, 1) OVER(),
  FIRST_VALUE(users) OVER(),
  CAST(users::FLOAT*100 / FIRST_VALUE(users) OVER() AS NUMERIC(5, 1)) AS conversion_rate
FROM
  funnel;
--5. Using the percent of previous approach, what are the user-level conversion rates for the
-- first 3 stages of the funnel (app download to signup and signup to ride requested)?
-- 74.6% and 70.4%
WITH cte_downloads AS
(
  SELECT DISTINCT
    app_download_key AS download
  FROM
    app_downloads
)
,
cte_signups AS
(
  SELECT DISTINCT
    s.user_id AS signups
  FROM
    signups AS s
    LEFT JOIN
      cte_downloads AS d
      ON d.download = s.session_id
)
,
cte_request AS
(
  SELECT DISTINCT
    signups AS request
  FROM
    ride_requests rr
    LEFT JOIN
      cte_signups s
      ON rr.user_id = s.signups
  WHERE
    rr.request_ts IS NOT NULL
)
,
cte_completed AS
(
  SELECT DISTINCT
```

```
    request AS completed
  FROM
    ride_requests rr
    LEFT JOIN
      cte_request r
      ON rr.user_id = r.request
  WHERE
    rr.dropoff_ts IS NOT NULL
)
,
funnel AS
(
  SELECT
    '1' AS n,
    'download' AS step,
    COUNT(*) AS users
  FROM
    cte_downloads
  UNION
  SELECT
    '2' AS n,
    'signup' AS step,
    COUNT(*) AS users
  FROM
    cte_signups
  UNION
  SELECT
    '3' AS n,
    'request' AS step,
    COUNT(*) AS users
  FROM
    cte_request
  UNION
  SELECT
    '4' AS n,
    'completed' AS step,
    COUNT(*) AS users
  FROM
    cte_completed
  ORDER BY
    n
)
SELECT
  *,
  CAST(users::FLOAT*100 / lag(users, 1) OVER() AS NUMERIC(5, 1)) AS conversion_rate
FROM
  funnel;
--6. Using the percent of top approach, what are the user-level conversion rates for the first 3
-- stages of the funnel (app download to signup and signup to ride requested)?
-- 74.6% and 52.5%
WITH cte_downloads AS
(
  SELECT DISTINCT
    app_download_key AS download
  FROM
```

```
    app_downloads
)
,
cte_signups AS
(
  SELECT DISTINCT
    s.user_id AS signups
  FROM
    signups AS s
    LEFT JOIN
      cte_downloads AS d
      ON d.download = s.session_id
)
,
cte_request AS
(
  SELECT DISTINCT
    signups AS request
  FROM
    ride_requests rr
    LEFT JOIN
      cte_signups s
      ON rr.user_id = s.signups
  WHERE
    rr.request_ts IS NOT NULL
)
,
cte_completed AS
(
  SELECT DISTINCT
    request AS completed
  FROM
    ride_requests rr
    LEFT JOIN
      cte_request r
      ON rr.user_id = r.request
  WHERE
    rr.dropoff_ts IS NOT NULL
)
,
funnel AS
(
  SELECT
    '1' AS n,
    'download' AS step,
    COUNT(*) AS users
  FROM
    cte_downloads
  UNION
  SELECT
    '2' AS n,
    'signup' AS step,
    COUNT(*) AS users
  FROM
    cte_signups
```

```
  UNION
  SELECT
    '3' AS n,
    'request' AS step,
    COUNT(*) AS users
  FROM
    cte_request
  UNION
  SELECT
    '4' AS n,
    'completed' AS step,
    COUNT(*) AS users
  FROM
    cte_completed
  ORDER BY
    n
)
SELECT
  *,
  CAST(users::FLOAT*100 / lag(users, 1) OVER() AS NUMERIC(5, 1)) AS conversion_rate_pop,
  CAST(users::FLOAT*100 / FIRST_VALUE(users) OVER() AS NUMERIC(5, 1)) AS conversion_rate_pot
FROM
  funnel;
--7. Using the percent of previous approach, what are the user-level conversion rates for the
-- following 3 stages of the funnel? 1. signup, 2. ride requested, 3. ride completed
-- 70.4% and 50.2%
WITH cte_downloads AS
(
  SELECT DISTINCT
    app_download_key AS download
  FROM
    app_downloads
)
,
cte_signups AS
(
  SELECT DISTINCT
    s.user_id AS signups
  FROM
    signups AS s
    LEFT JOIN
      cte_downloads AS d
      ON d.download = s.session_id
)
,
cte_request AS
(
  SELECT DISTINCT
    signups AS request
  FROM
    ride_requests rr
    LEFT JOIN
      cte_signups s
      ON rr.user_id = s.signups
  WHERE
```

```
    rr.request_ts IS NOT NULL
)
,
cte_completed AS
(
  SELECT DISTINCT
    request AS completed
  FROM
    ride_requests rr
    LEFT JOIN
      cte_request r
      ON rr.user_id = r.request
  WHERE
    rr.dropoff_ts IS NOT NULL
)
,
funnel AS
(
  SELECT
    '1' AS n,
    'download' AS step,
    COUNT(*) AS users
  FROM
    cte_downloads
  UNION
  SELECT
    '2' AS n,
    'signup' AS step,
    COUNT(*) AS users
  FROM
    cte_signups
  UNION
  SELECT
    '3' AS n,
    'request' AS step,
    COUNT(*) AS users
  FROM
    cte_request
  UNION
  SELECT
    '4' AS n,
    'completed' AS step,
    COUNT(*) AS users
  FROM
    cte_completed
  ORDER BY
    n
)
SELECT
  *,
  CAST(users::FLOAT*100 / lag(users, 1) OVER() AS NUMERIC(5, 1)) AS conversion_rate_pop,
  CAST(users::FLOAT*100 / FIRST_VALUE(users) OVER() AS NUMERIC(5, 1)) AS conversion_rate_pot
FROM
  funnel;
```

```
--8. Using the percent of top approach, what are the user-level conversion rates for the following
-- 3 stages of the funnel? 1. signup, 2. ride requested, 3. ride completed (hint: signup is the top
--  of this funnel)
WITH cte_downloads AS
(
  SELECT DISTINCT
    app_download_key AS download
  FROM
    app_downloads
)
,
cte_signups AS
(
  SELECT DISTINCT
    s.user_id AS signups
  FROM
    signups AS s
    LEFT JOIN
      cte_downloads AS d
      ON d.download = s.session_id
)
,
cte_request AS
(
  SELECT DISTINCT
    signups AS request
  FROM
    ride_requests rr
    LEFT JOIN
      cte_signups s
      ON rr.user_id = s.signups
  WHERE
    rr.request_ts IS NOT NULL
)
,
cte_completed AS
(
  SELECT DISTINCT
    request AS completed
  FROM
    ride_requests rr
    LEFT JOIN
      cte_request r
      ON rr.user_id = r.request
  WHERE
    rr.dropoff_ts IS NOT NULL
)
,
funnel AS
(
  --  select '1' as n, 'download' as step, count(*) as users from cte_downloads
  --         union
  SELECT
    '1' AS n,
    'signup' AS step,
```

```
    COUNT(*) AS users
  FROM
    cte_signups
  UNION
  SELECT
    '2' AS n,
    'request' AS step,
    COUNT(*) AS users
  FROM
    cte_request
  UNION
  SELECT
    '3' AS n,
    'completed' AS step,
    COUNT(*) AS users
  FROM
    cte_completed
  ORDER BY
    n
)
SELECT
  *,
  CAST(users::FLOAT*100 / lag(users, 1) OVER() AS NUMERIC(5, 1)) AS conversion_rate_pop,
  CAST(users::FLOAT*100 / FIRST_VALUE(users) OVER() AS NUMERIC(5, 1)) AS conversion_rate_pot
FROM
  funnel;

--  Explore the reviews by creating a list of used words to analyze in Tableau

SELECT REPLACE(REPLACE(TRIM(name_part), '.', ''), ',', '') AS cleaned_name_part
FROM (
 SELECT LOWER(unnest(string_to_array(review, ' '))) AS name_part
 FROM reviews
 WHERE rating = 1 OR rating = 2
) subquery
WHERE REPLACE(REPLACE(TRIM(name_part), '.', ''), ',', '') NOT IN ('the', 'a', 'was', 'and', 'to', 'or', 'ride', 'with',
'driver', 'in', 'metrocar', 'took', 'up', 'me', 'ever');
```

6. Data Aggregation in SQL

```
--Final data aggregation
WITH user_details AS
(
  SELECT
    app_download_key,
    user_id,
    platform,
    age_range,
    DATE(download_ts) AS download_dt
  FROM
    app_downloads
    LEFT JOIN
      signups
      ON app_downloads.app_download_key = signups.session_id
)
,
```

```
ride_details AS
(
  SELECT
    ride_id,
    user_details.*
  FROM
    ride_requests
    LEFT JOIN
      user_details USING(user_id)
)
,
downloads AS
(
  SELECT
    0 AS step,
    'download' AS name,
    platform,
    age_range,
    download_dt,
    COUNT(DISTINCT app_download_key) AS users_count,
    0 AS count_rides
  FROM
    user_details
  GROUP BY
    platform,
    age_range,
    download_dt
)
,
signup AS
(
  SELECT
    1 AS step,
    'signup' AS name,
    user_details.platform,
    user_details.age_range,
    user_details.download_dt,
    COUNT(DISTINCT user_id) AS users_count,
    0 AS count_rides
  FROM
    signups
    JOIN
      user_details USING(user_id)
  WHERE
    signup_ts IS NOT NULL
  GROUP BY
    user_details.platform,
    user_details.age_range,
    user_details.download_dt
)
,
requested AS
(
  SELECT
    2,
```

```
    'ride_requested',
    user_details.platform,
    user_details.age_range,
    user_details.download_dt,
    COUNT(DISTINCT user_id) AS users_count,
    COUNT (DISTINCT ride_id) AS count_rides
  FROM
    ride_requests
    JOIN
      user_details USING(user_id)
  WHERE
    request_ts IS NOT NULL
  GROUP BY
    user_details.platform,
    user_details.age_range,
    user_details.download_dt
)
,
accepted AS
(
  SELECT
    3,
    'ride_accepted',
    user_details.platform,
    user_details.age_range,
    user_details.download_dt,
    COUNT(DISTINCT user_id) AS users_count,
    COUNT (DISTINCT ride_id) AS count_rides
  FROM
    ride_requests
    JOIN
      user_details USING(user_id)
  WHERE
    accept_ts IS NOT NULL
  GROUP BY
    user_details.platform,
    user_details.age_range,
    user_details.download_dt
)
,
completed AS
(
  SELECT
    4,
    'ride_completed',
    user_details.platform,
    user_details.age_range,
    user_details.download_dt,
    COUNT(DISTINCT user_id) AS users_count,
    COUNT (DISTINCT ride_id) AS count_rides
  FROM
    ride_requests
    JOIN
      user_details USING(user_id)
  WHERE
```

```
    dropoff_ts IS NOT NULL
  GROUP BY
    user_details.platform,
    user_details.age_range,
    user_details.download_dt
)
,
payment AS
(
  SELECT
    5,
    'payment',
    ride_details.platform,
    ride_details.age_range,
    ride_details.download_dt,
    COUNT(DISTINCT user_id) AS users_count,
    COUNT (DISTINCT ride_id) AS count_rides
  FROM
    transactions
    LEFT JOIN
      ride_details USING(ride_id)
  WHERE
    charge_status = 'Approved'
  GROUP BY
    ride_details.platform,
    ride_details.age_range,
    ride_details.download_dt
)
,
review AS
(
  SELECT
    6,
    'review',
    user_details.platform,
    user_details.age_range,
    user_details.download_dt,
    COUNT(DISTINCT user_id) AS users_count,
    COUNT (DISTINCT ride_id) AS count_rides
  FROM
    reviews
    JOIN
      user_details USING(user_id)
  GROUP BY
    user_details.platform,
    user_details.age_range,
    user_details.download_dt
)
SELECT
  *
FROM
  downloads
UNION
SELECT
  *
```

```
FROM
  signup
UNION
SELECT
  *
FROM
  requested
UNION
SELECT
  *
FROM
  accepted
UNION
SELECT
  *
FROM
  completed
UNION
SELECT
  *
FROM
  payment
UNION
SELECT
  *
FROM
  review
ORDER BY
  1,
  2,
  3,
  4,
  5;
```