

METROCAR FUNNEL ANALYSIS

BY: Asenoguan Eric

Date: 25/09/2023

1. SUMMARY

Metrocar funnel analysis was done on two levels, the Users funnel and the Ride funnel, with the aim of identifying areas of improvement and optimization.

The Users funnel has seven steps from App downloads to Reviews

While the Ride funnel has five steps from Ride requests to Reviews,

2. CONTEXT

The metrocar 2021 dataset had five tables namely:

App download, Signups, Ride requests, Transactions and reviews,

With SQL code I was able to explore and extract necessary data needed for visualisation and insight in Tableau.

Both Users funnel And Ride funnel were created in Tableau and had Steps from App downloads to Reviews

3. RESULT, BUSINESS QUESTIONS, INSIGHTS AND RECOMMENDATIONS.

3.1 .What steps of the funnel should we research and improve?

Are there any specific drop-off points preventing users from completing their first ride?

Visualisation:

A plot of the funnel using tableau shows a visualisation of both the user and ride levels funnels using the percentage of previous approach and percentage of top approach.

Users Funnel

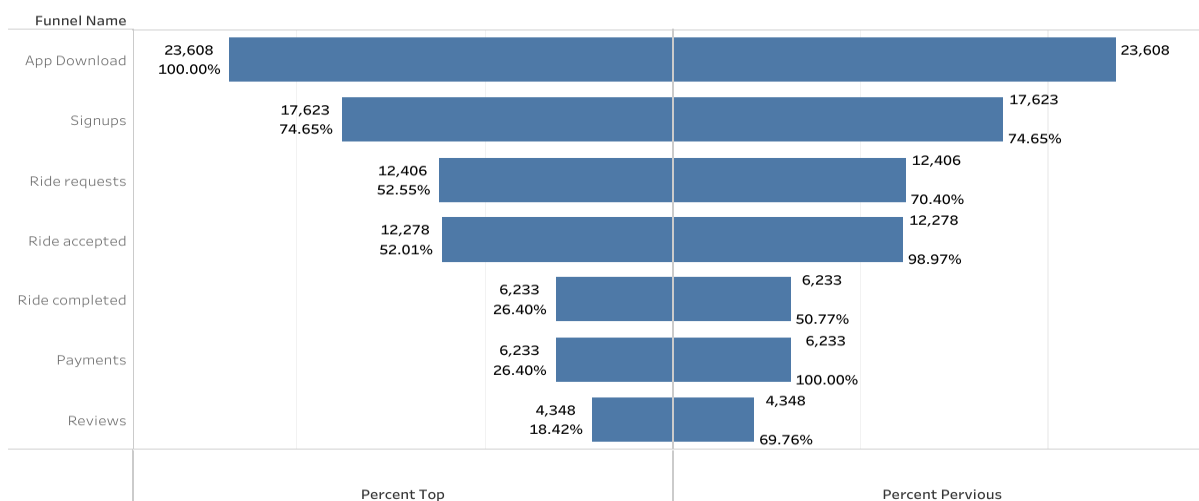


Figure 1: conversion rate of Metrocar Funnel on user level

Ride Funnel

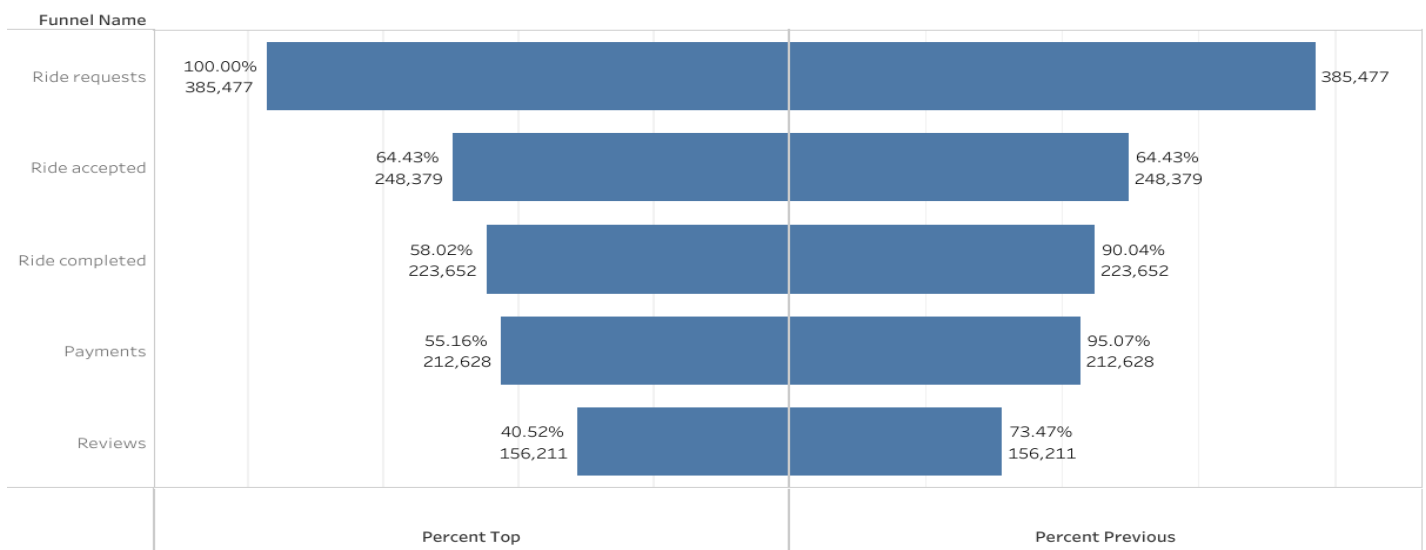


Figure 2: conversion rate of Metrocar Funnel on ride level.

***Note: Drop-off rate = 100 – Conversion rate**

3.1.2 Insights:

On a User funnel using the percentage of previous , the funnel step with the highest drop-off rates was from ride accepted to ride completed with a drop-off rate of about 49.23%, Signups to Ride requests with a drop-off rate of 29.60% and App download to signups with a drop-off rate of 25.35%.

while on a Ride funnel using the percentage of previous which gives us a more precise view of the ride activities, the ride requests to ride accepted had the highest drop-off rate of about 35.57%.

3.1.3 RECOMMENDATION

The steps to research and improve are all these steps with high drop off rate, Reasons for drop off in the Ride funnel could be:

- Insecurity
- No sufficient drivers
- Maybe longer distance.
- Rider becoming Impatient and cancelling request before drivers arrival

3.2 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?

Visualisation:

A bar chart in tableau shows a visualisation of both the user and ride levels granularity for all three platforms.

Users Platform Count

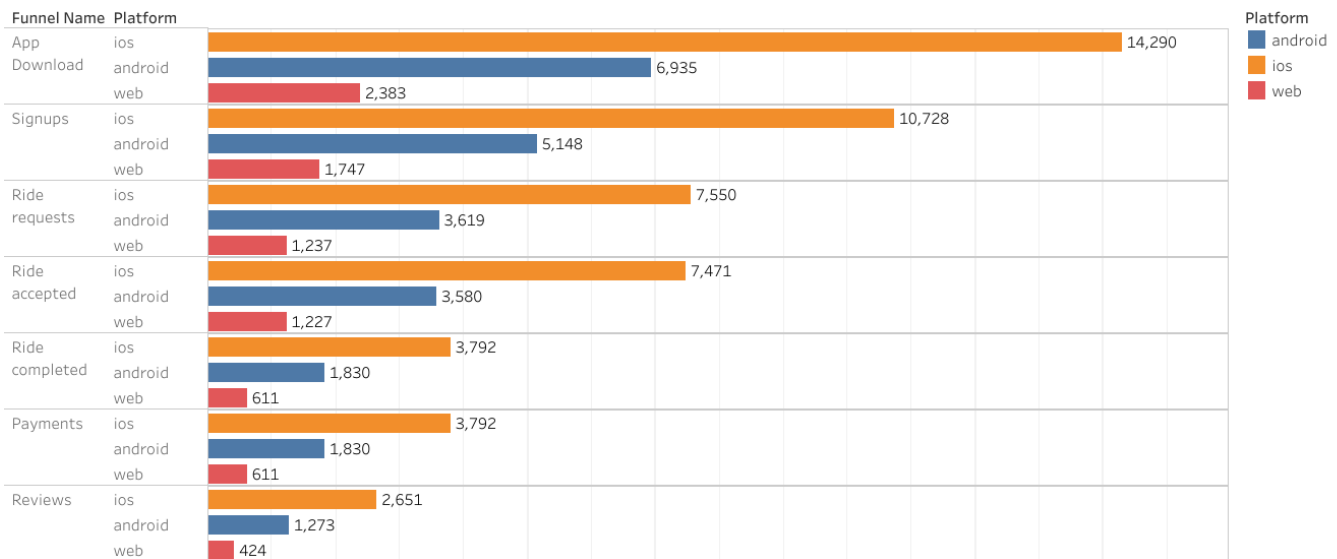


Figure 3: users platform count

Ride Platform Count

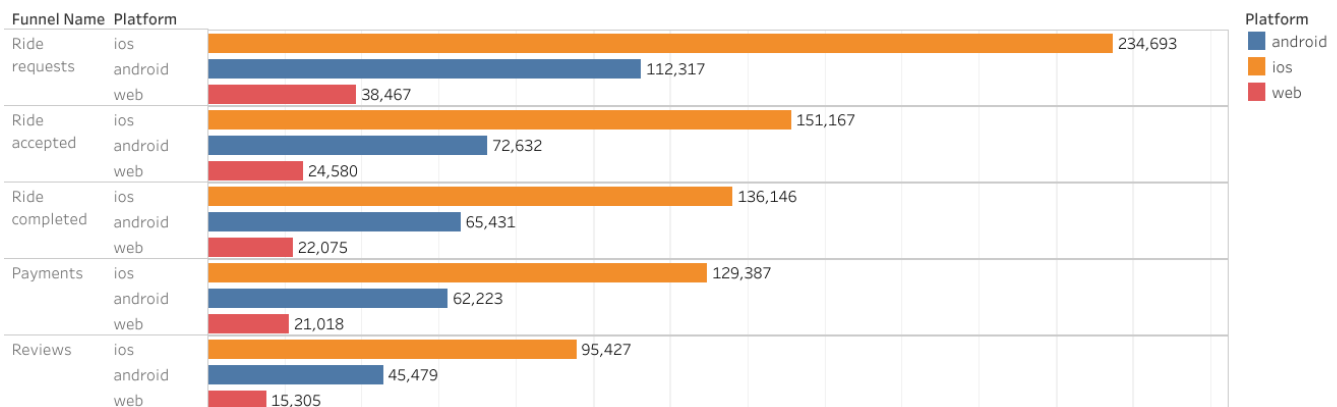


Figure 4: Ride Platform Count

3.2.2 Insights:

From figure 3 and 4 we can see that in both users and Ride level granularity metrocar users are mostly from IOS, which is also the platform with the highest Ride requests,

After IOS we have Android as the second best platform .

3.2.3 RECOMMENDATION:

My Recommendation would be to focus marketing budget in acquiring and retaining users from IOS and Android platform, in other words mobile users

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

Visualisation:

A bar chart in tableau shows a visualisation of both the user and ride levels granularity for all age groups.

Users Age Range Count

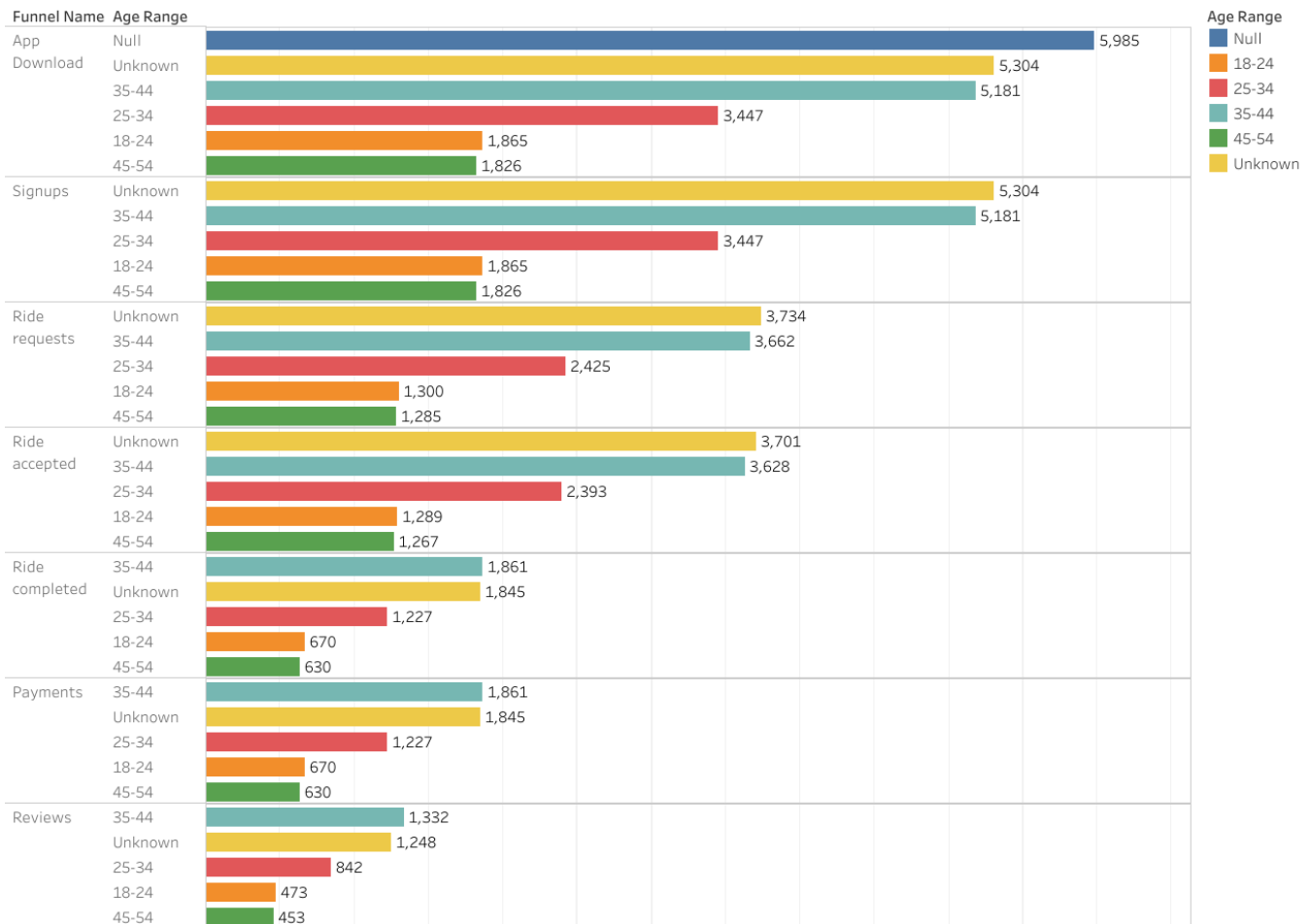


Figure 5: Users age-groups

Ride Age Range Count

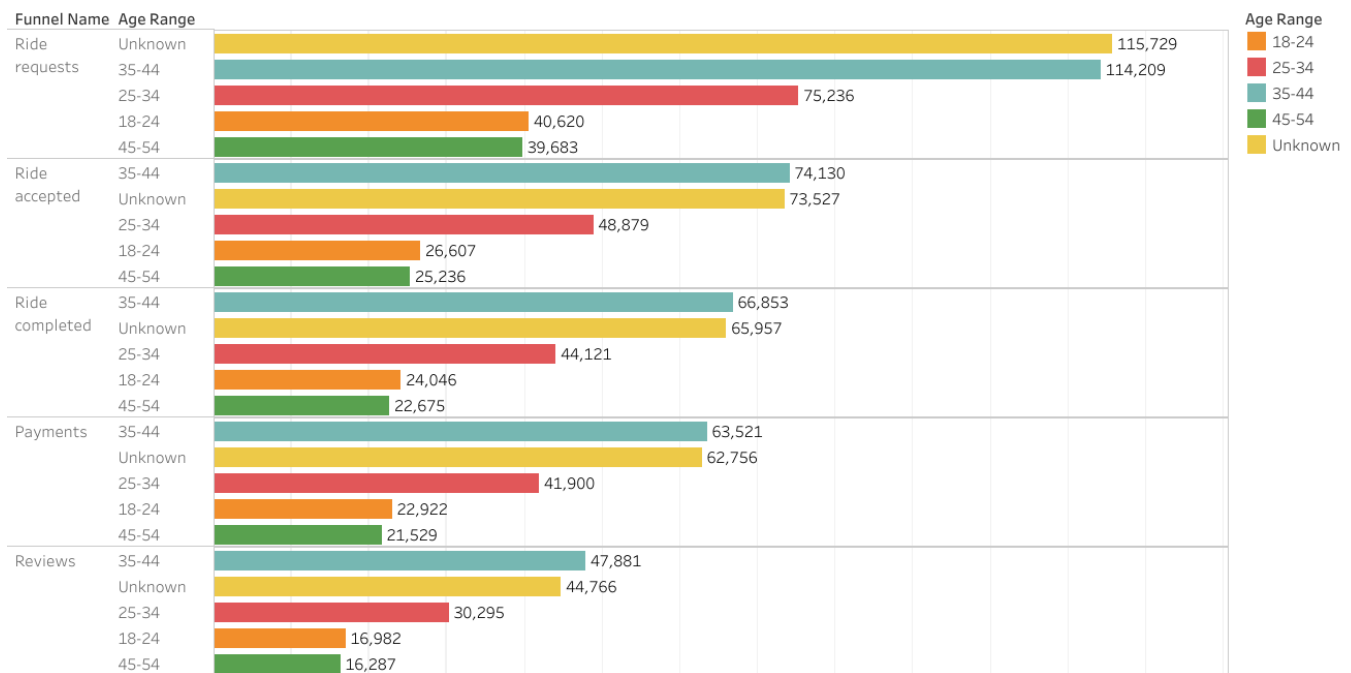


Figure 6: Ride age-group

3.3.2 Insights:

From figure 5 and 6 we can see that in both users and Ride level granularity metrocar users are between age-group 35-44 for non-null age-group,

It is also good to note that the age groups with the target customers are 35-44 and 25-34 for the non-null age groups

3.3.3 RECOMMENDATIONS

To be sure that metrocar target customers are between the age group of 35-44 and 25-34, I would like to have age-groups of the null and unknown age group if possible.

3.4: 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-surfing strategy, what does the distribution of ride requests look like throughout the day?

Visualisation:

A trend line chart in tableau shows the distribution of ride requests throughout the day.

Ride Request Per Hours

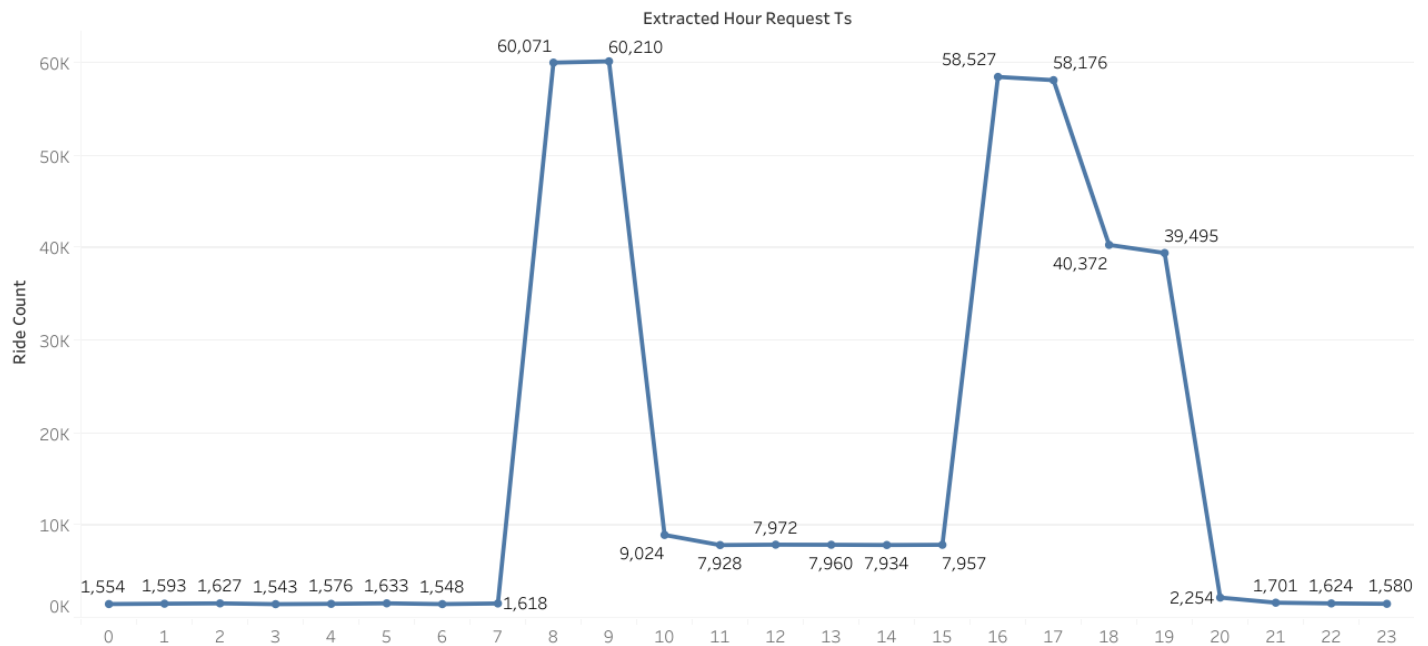


Figure 7: Hourly distribution of ride requests.

3.4.2 Insights:

My distribution show two period of increase ride request, these are between 8AM-9AM and 4PM-19AM,

60,071 ride requests were made between 07:00 - 08:00 hours while 60,210 ride requests were made between 08:00- 09:00 hours of the day. Also, 58,527 ride requests were made between 15:00-16:00 hours while 58,176 ride requests were made between 16:00-17:00 hours, also 40,372 ride requests between 17:00-18:00 and 39,495 ride requests between 19:00-20:00 hours.

3.4.3 RECOMMENDATION:

Surge pricing can be implemented for rides requests between 07:00 – 10:00 hours and between 15:00 – 20:00 hours of the day and also drivers should be paid more during these hours.

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

Visualisation:

I created a kind of funnel chart in tableau showing percent previous and drop off rate for both users and ride level granularity to understand better the conversion rate and drop off rate, Clearly steps with a low conversion rate has high drop off rate.

Note conversion rate + drop off rate = 100

Users Dropoff

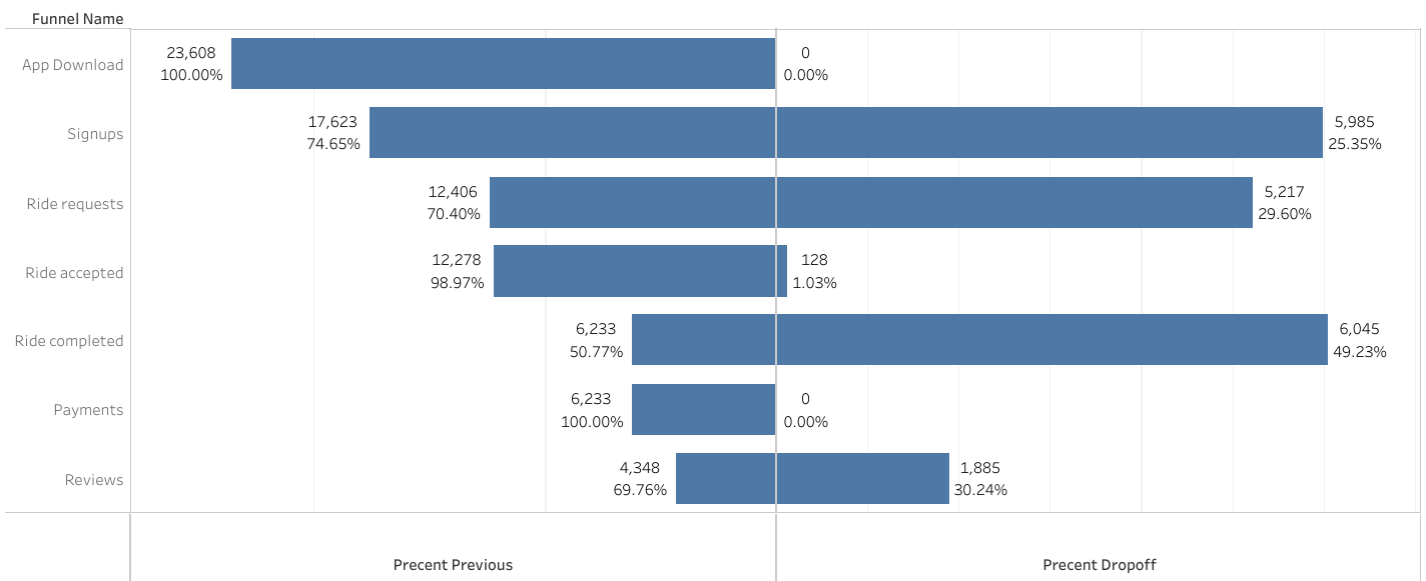


Figure 8: Users level drop off rate

Ride Dropoff

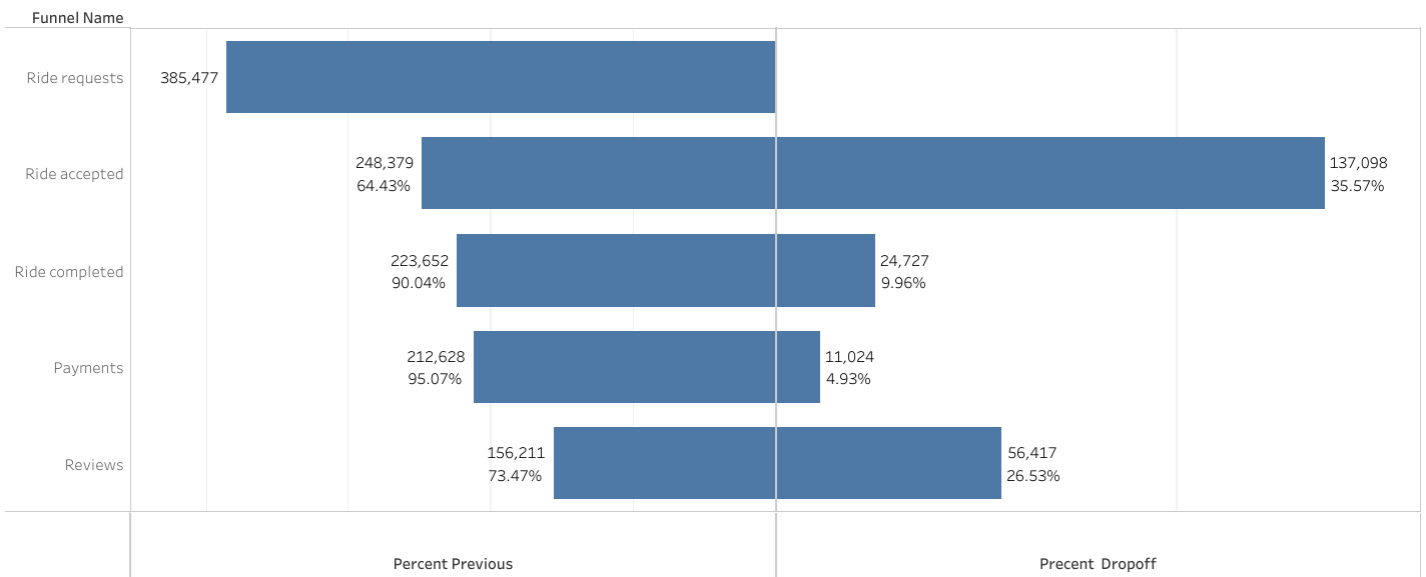


Figure 9: Ride level drop off rate

3.5.2 Insights

The users level granularity has the ride accepted to ride completed as the part of the users funnel with the lowest conversion rate (50.77%) and highest drop off rate (49.23%),

While ride level granularity has the ride requests to ride accepted as part of the ride funnel with the lowest conversion rate(64.43%) and highest drop off rate(35.57)

3.5.3: RECOMMENDATION

What can be done to improve these steps are not what I can answer with metrocar dataset, Stakeholders should look at my report and make necessary decisions on improving the parts of the funnel with low conversion rate.

4. CONCLUSION

The Metrocar funnel analysis shows that most users and rides were lost between the ride request and ride completion stage. Metrocar App should be further optimised and more drivers should be employed and trained, incentives for good performance to help boost user conversion at these stages of the funnel. More marketing budget should be channelled towards users on iOS platform and Android . Finally, surge pricing should be introduced for ride requests between 07:00 – 09:00 hours and between 15:00 – 17:00 hours of the day.

5. Appendix

Link to video presentation

<https://www.loom.com/share/7ea6f2f46fd64496a6557545d423dbca>

Link to tableau funnel chart

<https://public.tableau.com/authoring/MetrocarCompleteFunnelAnalysis/Sheet1#1>

Link to tableau Surge pricing

<https://public.tableau.com/authoring/MetrocarRideRequestHours/RideRequestPerHours#1>

SQL code used to query metrocar dataset

WITH funnel AS (

```
SELECT app_download_key,platform, download_ts,s.user_id,s.session_id,s.signup_ts,age_range,
       ri.ride_id,ri.driver_id,ri.request_ts,ri.accept_ts,ri.pickup_location,ri.dropoff_location,
       ri.pickup_ts,ri.dropoff_ts,ri.cancel_ts,t.transaction_id,t.purchase_amount_usd,
       t.charge_status,t.transaction_ts,re.review_id,re.rating,re.review
```

```
FROM app_downloads AS a
```

```
Left JOIN signups AS s
```

```
ON a.app_download_key = s.session_id
```

```
LEFT JOIN ride_requests AS ri
```

```
ON s.user_id = ri.user_id
```

```
LEFT JOIN transactions AS t
```

```
ON ri.ride_id = t.ride_id
```

```
LEFT JOIN reviews AS re
```

```
ON t.ride_id = re.ride_id),
```

```
funnel_table AS (
```

```
SELECT 1 AS funnel_step,
```

```
       'App Download' AS funnel_name,
```

```
       COUNT(DISTINCT app_download_key) AS user_count,platform, age_range,
```

```
       EXTRACT(MONTH FROM download_ts) AS extracted_month,
```

```
       NULL::INTEGER AS ride_count
```

```
FROM funnel
```

```
GROUP BY platform,age_range,6
```

```
union
```

```
SELECT 2 AS funnel_step,
```

```
       'Signups' AS funnel_name,
```

```
       COUNT(DISTINCT session_id) AS user_count,platform, age_range,
```

```
       EXTRACT(MONTH FROM download_ts) AS extracted_month,
```

```
       NULL::INTEGER AS ride_count
```

```
FROM funnel
```

```
WHERE age_range IS NOT NULL
```

```
GROUP BY platform,age_range,6
```

```
union
```

```
SELECT 3 AS funnel_step,
```

```
       'Ride requests' AS funnel_name,
```

```
       COUNT(DISTINCT rr.user_id) AS user_count,f.platform, f.age_range,
```

```
       EXTRACT(MONTH FROM f.download_ts) AS extracted_month,
```

```
       COUNT(DISTINCT f.ride_id) AS ride_count
```

```
FROM funnel f
```

```
INNER JOIN ride_requests rr ON f.user_id = rr.user_id
```

```
GROUP BY f.platform,f.age_range,6
```

```
union
```

```
SELECT 4 AS funnel_step,
```

```
       'Ride accepted' AS funnel_name,
```

```
       COUNT(DISTINCT user_id) AS user_count,platform, age_range,
```

```
       EXTRACT(MONTH FROM download_ts) AS extracted_month,
```

```

COUNT(ride_id) AS ride_count
FROM funnel
WHERE accept_ts IS NOT NULL
GROUP BY platform,age_range,6
union
SELECT 5 AS funnel_step,
      'Ride completed' AS funnel_name,
      COUNT(DISTINCT user_id) AS user_count,platform, age_range,
      EXTRACT(MONTH FROM download_ts) AS extracted_month,
      COUNT(ride_id) AS ride_count
FROM funnel
WHERE dropoff_ts IS NOT NULL
GROUP BY platform,age_range,6
union
SELECT 6 AS funnel_step,
      'Payments' AS funnel_name,
      COUNT(DISTINCT user_id) AS user_count,platform, age_range,
      EXTRACT(MONTH FROM download_ts) AS extracted_month,
      COUNT(ride_id) AS ride_count
FROM funnel
WHERE charge_status = 'Approved'
GROUP BY platform,age_range,6
union
SELECT 7 AS funnel_step,
      'Reviews' AS funnel_name,
      COUNT(DISTINCT user_id) AS user_count,platform, age_range,
      EXTRACT(MONTH FROM download_ts) AS extracted_month,
      COUNT(ride_id) AS ride_count
FROM funnel
WHERE review IS NOT NULL
GROUP BY platform,age_range,6
),
funnel_sum AS (
SELECT funnel_step, funnel_name, user_count, platform, age_range,extracted_month,ride_count,
      SUM(user_count)OVER(PARTITION BY funnel_name ORDER BY funnel_step) AS total_value,
      SUM(ride_count)OVER(PARTITION BY funnel_name ORDER BY funnel_step) AS total_ride
FROM funnel_table
ORDER BY funnel_step),

funnel_lag AS (
SELECT
      funnel_step, funnel_name,user_count,platform, age_range,extracted_month,ride_count,
      total_value,total_ride,
      ROUND((total_value::NUMERIC / FIRST_VALUE(total_value)
      OVER (ORDER BY funnel_step)) * 100, 2) AS percent_top,
      LAG(total_value,1,23608)OVER(ORDER BY funnel_step) AS lag_total_value,
      ROUND(((total_ride::NUMERIC / 385477)) * 100, 2) AS ride_percent_top,
      LAG(total_ride)OVER(ORDER BY funnel_step) AS lag_total_ride

```

```
FROM funnel_sum  
ORDER BY funnel_step)
```

```
SELECT funnel_step,funnel_name,user_count,platform,age_range,extracted_month,total_value,  
       percent_top,lag_total_value,ride_count,total_ride,ride_percent_top,lag_total_ride,  
       ROUND((total_value::NUMERIC / FIRST_VALUE(lag_total_value) OVER (PARTITION BY funnel_step  
       ORDER BY funnel_step )) * 100, 2) AS percent_previous,  
       ROUND((total_ride::NUMERIC / FIRST_VALUE(lag_total_ride) OVER (PARTITION BY funnel_step  
       ORDER BY funnel_step )) * 100, 2) AS ride_percent_previous,  
       ROUND((1 - (total_value::NUMERIC / FIRST_VALUE(lag_total_value) OVER (PARTITION BY  
funnel_step  
       ORDER BY funnel_step ))) * 100, 2) AS percent_user_dropoff,  
       ROUND((1 - (total_ride::NUMERIC / FIRST_VALUE(lag_total_ride) OVER (PARTITION BY funnel_step  
       ORDER BY funnel_step ))) * 100, 2) AS percent_ride_dropoff,  
       FIRST_VALUE(lag_total_value) OVER (PARTITION BY funnel_step  
       ORDER BY funnel_step )-total_value AS user_dropoff_num,  
       FIRST_VALUE(lag_total_ride) OVER (PARTITION BY funnel_step  
       ORDER BY funnel_step ) - total_ride AS ride_dropoff_num  
FROM funnel_lag  
ORDER BY funnel_step
```

SQL CODE FOR SURGE PRICING

```
SELECT COUNT(ride_id) AS ride_count,COUNT(DISTINCT user_id) AS users_count,  
       EXTRACT(HOUR FROM request_ts) AS extracted_hour_request_ts,  
       COUNT(driver_id) AS drivers_count  
FROM ride_requests  
GROUP BY 3
```

Understanding the Metrocar Database

1. How many times was the app downloaded?

```
SELECT COUNT(*)  
FROM app_downloads
```

2. How many users signed up on the app?

```
SELECT COUNT(*)  
FROM signups
```

3. How many rides were requested through the app

```
SELECT COUNT(*)  
FROM ride_requests
```

4. How many rides were requested and completed through the app?

```
SELECT COUNT(*)  
FROM ride_requests  
WHERE dropoff_ts IS NOT NULL
```

5. How many rides were requested and how many unique users requested a ride

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

6. What is the average time of a ride from pick up to drop off?

```
SELECT AVG(DATE_TRUNC('minute',((dropoff_ts - pickup_ts)* 60)))  
FROM ride_requests
```

7. How many rides were accepted by a driver?

```
SELECT COUNT(*)  
FROM ride_requests  
WHERE accept_ts IS NOT NULL
```

8. How many rides did we successfully collect payments and how much was collected?

```
SELECT charge_status, COUNT(*) AS ride_count, SUM(purchase_amount_usd) AS amt  
FROM transactions  
WHERE charge_status = 'Approved'  
GROUP BY 1
```

9. How many ride requests happened on each platform?

```
SELECT platform, COUNT(*) AS ride_request  
FROM ride_requests AS r  
JOIN signups AS s  
ON r.user_id = s.user_id  
JOIN app_downloads AS a  
ON a.app_download_key = s.session_id  
GROUP BY 1
```

10. What is the drop-off from users signing up to users requesting a ride?

```
WITH T1 AS (SELECT 'Sign up' AS step, COUNT(*) AS users_count
             FROM signups
             union
             SELECT 'Request' AS step, COUNT(DISTINCT user_id) As users_count
             FROM ride_requests),
T2 AS (
SELECT step, users_count, lag(users_count, 1) OVER ()
FROM T1)
SELECT
ROUND((1 - (lag::NUMERIC /users_count )) * 100, 2) AS percent_dropoff
FROM T2
```

