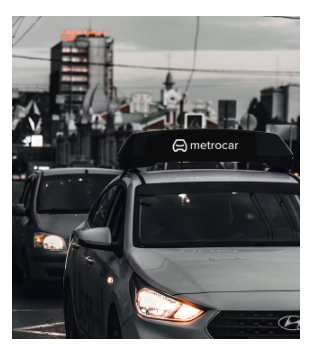
1. **Metrocar - Funnel Analysis**

By Alina Barsan

05/07/2023



1. **Summary**

The objective of this project is to analyze the customer funnel of Metrocar, a ride-sharing app to identify areas for improvement and optimization.

Metrocar's business model is based on a platform that connects riders with drivers through a mobile application.

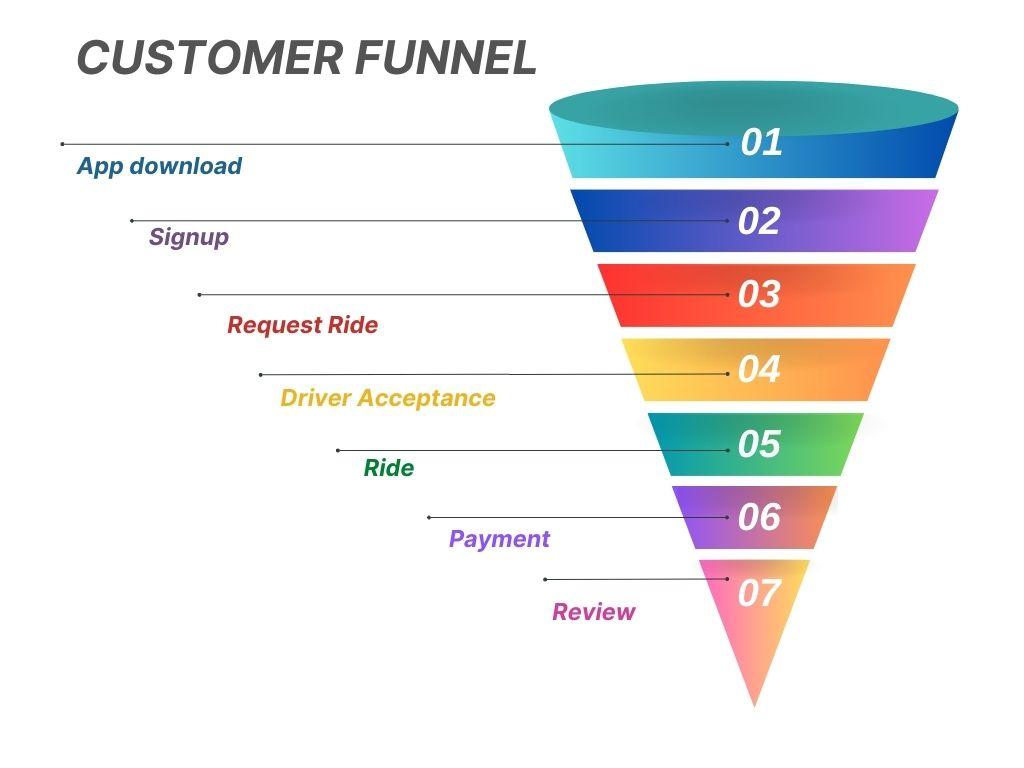
Metrocar acts as an intermediary between riders and drivers, providing a user-friendly platform to connect them and facilitate the ride-hailing process.

1. **Context**

We conducted a funnel analysis, exploring the data with SQL.

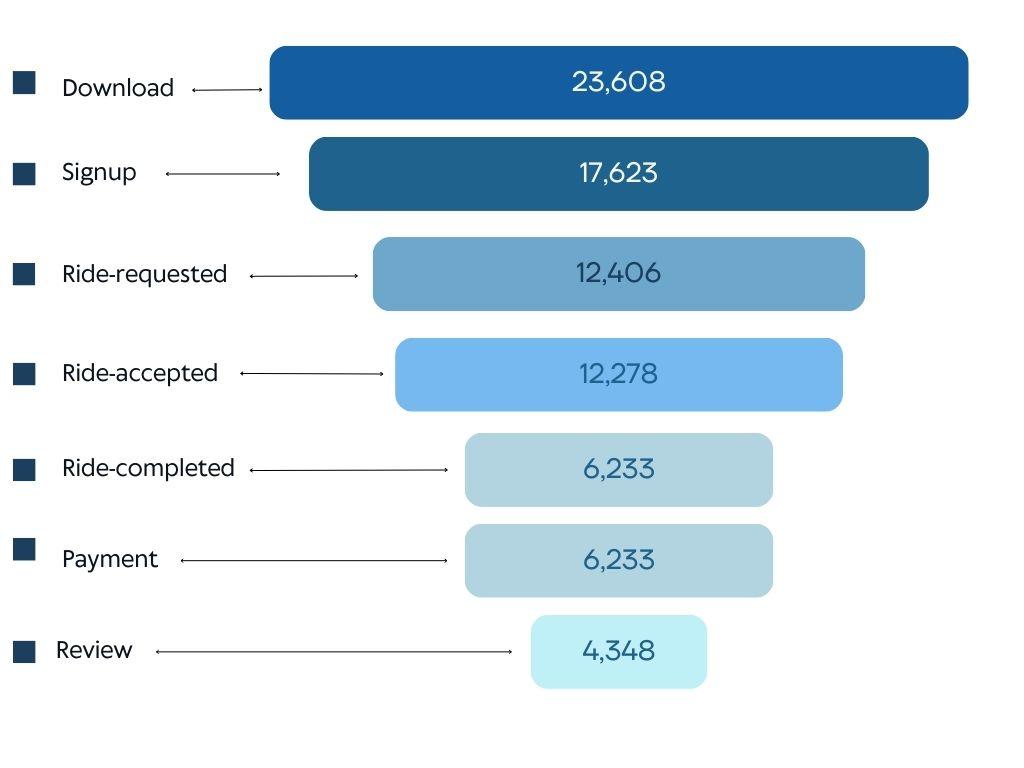
Here are the customer Metrocar’s Funnel Steps:

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.



1. **Key Findings**

We concluded that the app was downloaded 23,608 times; there were 17,623 users that signed up on the app; a total of 385,477 rides requested and 223,652 were rides completed, from which 12,406 were rides requested by unique users and 12,278 were rides accepted by drivers, from which only 6,233 were rides completed and 4,348 users left reviews.



***USER FUNNEL STEPS***

Having a closer look at the age range of the users we found that the age range of 35-44 had the highest count of 5,181, followed by the age range of 25-34 with a count of 3,447, followed by the age range of 18-24 with a count of 1,865 and the lowest count of 1,823 was found in the age range of 45-54.

Focusing on the number of users on each platform, we discovered that the iso platform had the highest number of users 50,274 with a count of 234,693 of rides requested; followed by the android platform having a number of users of 24,215 with a count of 112,317 of rides requested; and on the other hand with the lowest count of 8,240 users for the web platform with a count of 38,467 rides requested.

From further analysis of this data set, we discovered that a higher demand of rides requested was at rush hours 8am - 9am and 4pm - 5pm.

1. **Recommendation**

Through this exploratory data analysis and visualization, we gained several interesting insights into the Metrocar app.

We proceeded by analyzing the funnel steps of the customer experience, where we found a significant drop in numbers from ride requests and rides completed. The data shows that there is demand, especially at rush hours. We could pay the drivers a higher rate during the rush hours, to encourage them to accept rides, when there is heavy traffic and more waiting time, also because of this we should look into optimizing the customer experience and increase the number of drivers to decrease the waiting time for users.

We recommend focusing on the most popular platform, which is ios, to invest the marketing budget for the next coming year, because it has the largest group of users.

We could offer students discounts or special promotions, for example if they refer someone, both parties would get a discount; we could encourage users that completed the ride to leave reviews, by offering them a few $ off for the next ride.

Through this analysis, we have a better idea on the customer experience funnel steps.

All of this can be taken into consideration for Metrocar’s future business plans.

1. **Appendix**

**SQL - codes -**

*1.How many times was the app downloaded?*

SELECT COUNT(app\_download\_key) as total\_downloads

FROM app\_downloads;

23608

*2.How many users signed up for the app?*

SELECT COUNT(user\_id) as total\_signups

FROM signups;

17623

*3.How many rides were requested through the app?*

SELECT COUNT(ride\_id) as total\_rides

FROM ride\_requests;

385477

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

SELECT COUNT(ride\_id) as rides\_completed

FROM transactions;

223652

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

SELECT COUNT(DISTINCT user\_id) as unique\_users

FROM ride\_requests;

12406

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

SELECT AVG(dropoff\_ts - pickup\_ts) AS average\_time\_diff

FROM ride\_requests;

52 minutes 36.738 seconds

*7.How many rides were accepted by a driver?*

SELECT COUNT(accept\_ts) AS total\_accepted

FROM ride\_requests;

248379

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

SELECT COUNT(charge\_status) AS total\_rides,

SUM(purchase\_amount\_usd) AS collected

FROM transactions

WHERE charge\_status = 'Approved';

212628, 4251667.610

*9.How many ride requests happened on each platform?*

SELECT COUNT(rr.ride\_id) AS ride\_requests,

platform

FROM app\_downloads a

JOIN signups s

ON a.app\_download\_key = s.session\_id

JOIN ride\_requests rr

ON s.user\_id = rr.user\_id

GROUP BY platform;

112317 - android

234693 - ios

38467 - web

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

SELECT

COUNT(DISTINCT s.user\_id) AS signup\_count,

COUNT(DISTINCT rr.user\_id) AS ride\_request\_count,

(COUNT(DISTINCT s.user\_id) - COUNT(DISTINCT rr.user\_id)) AS drop\_off\_count,

((COUNT(DISTINCT s.user\_id) - COUNT(DISTINCT rr.user\_id)) / COUNT(DISTINCT s.user\_id)::float) AS drop\_off\_rate

FROM signups s

LEFT JOIN ride\_requests rr

ON s.user\_id = rr.user\_id;

29.6%

*11.How many unique users completed a ride through the Metrocar app?*

SELECT COUNT(DISTINCT user\_id) as unique\_users

FROM ride\_requests rr

JOIN transactions t

ON rr.ride\_id = t.ride\_id

WHERE charge\_status ='Approved';

6233

*12.Of the users that signed up on the app,*

*What percentage of these users requested a ride?*

*requested / total signup and \* 100*

SELECT

(COUNT(DISTINCT rr.user\_id) \* 100.0) / COUNT(DISTINCT s.user\_id) AS percentage\_requested

FROM signups s

LEFT JOIN

ride\_requests rr

ON s.user\_id = rr.user\_id;

70.39

*13.Of the users that signed up on the app,*

*What percentage of these users completed a ride?*

SELECT

(COUNT(DISTINCT rr.user\_id) \* 100.0) / COUNT(DISTINCT s.user\_id) AS percentage\_completed

FROM signups s

LEFT JOIN ride\_requests rr

ON s.user\_id = r.user\_id

AND rr.dropoff\_ts IS NOT NULL;

35.36

*14.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 sign up to ride requested)?*

SELECT

(COUNT(DISTINCT s.user\_id) \* 100.0) / COUNT(DISTINCT a.app\_download\_key ) AS conversion\_rate\_signup,

(COUNT(DISTINCT rr.user\_id) \* 100.0) / COUNT(DISTINCT s.user\_id) AS conversion\_rate\_ride\_requested

FROM app\_downloads a

LEFT JOIN signups s

ON a.app\_download\_key = s.session\_id

LEFT JOIN ride\_requests rr

ON s.user\_id = rr.user\_id;

74.6%, 70.4%

*15.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 sign up to ride requested)?*

SELECT

(COUNT(DISTINCT s.user\_id) \* 100.0) / COUNT(DISTINCT a.app\_download\_key ) AS conversion\_rate\_signup,

(COUNT(DISTINCT rr.user\_id) \* 100.0) / COUNT(DISTINCT a.app\_download\_key ) AS conversion\_rate\_ride\_requested

FROM app\_downloads a

LEFT JOIN signups s

ON a.app\_download\_key = s.session\_id

LEFT JOIN ride\_requests rr

ON s.user\_id = rr.user\_id;

74.6%, 52.5%

*16.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*

SELECT

(COUNT(DISTINCT rr.user\_id) \* 100.0) / COUNT(DISTINCT s.user\_id) AS conversion\_rate\_ride\_requested,

(COUNT(DISTINCT

CASE

WHEN rr.dropoff\_ts IS NOT NULL

THEN rr.user\_id

END

) \* 100.0) / COUNT(DISTINCT rr.user\_id) AS conversion\_rate\_ride\_completed

FROM signups s

LEFT JOIN ride\_requests rr

ON s.user\_id = rr.user\_id;

70.4%, 50.2%

*17.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*

SELECT

(COUNT(DISTINCT rr.user\_id) \* 100.0) / COUNT(DISTINCT s.user\_id) AS conversion\_rate\_ride\_requested,

(COUNT(DISTINCT

CASE

WHEN rr.dropoff\_ts IS NOT NULL

THEN rr.user\_id

END

) \* 100.0) / COUNT(DISTINCT s.user\_id) AS conversion\_rate\_ride\_completed

FROM signups s

LEFT JOIN ride\_requests rr

ON s.user\_id = rr.user\_id;

70.4%, 35.4%

*waiting time*

SELECT accept\_ts - request\_ts as waiting\_time, rr.\*

FROM ride\_requests rr;

*Funnel code:*

with user\_details AS (

SELECT app\_download\_key, signups.user\_id, platform, age\_range, ride\_id, date(download\_ts) AS download\_dt

FROM app\_downloads

LEFT JOIN signups

ON app\_downloads.app\_download\_key = signups.session\_id

LEFT JOIN ride\_requests

ON ride\_requests.user\_id = signups.user\_id),

downloads AS (

SELECT 0 as funnel\_step,

'download' as funnel\_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 funnel\_step,

'signup' as funnel\_name,

user\_details.platform,

user\_details.age\_range,

user\_details.download\_dt,

COUNT (DISTINCT signups.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 as funnel\_step,

'ride\_requested' as funnel\_name,

user\_details.platform,

user\_details.age\_range,

user\_details.download\_dt,

COUNT (DISTINCT user\_id) as users\_count,

COUNT (DISTINCT ride\_requests.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 as funnel\_step,

'ride\_accepted' as funnel\_name,

user\_details.platform,

user\_details.age\_range,

user\_details.download\_dt,

COUNT (DISTINCT user\_id) as users\_count,

COUNT (DISTINCT ride\_requests.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 as funnel\_step,

'ride\_completed' as funnel\_name,

user\_details.platform,

user\_details.age\_range,

user\_details.download\_dt,

COUNT (DISTINCT user\_id) as users\_count,

COUNT (DISTINCT ride\_requests.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 as funnel\_step,

'payment' as funnel\_name,

user\_details.platform,

user\_details.age\_range,

user\_details.download\_dt,

COUNT (DISTINCT user\_id) AS users\_count,

COUNT (DISTINCT transactions.ride\_id) as count\_rides

FROM transactions

JOIN user\_details

USING (ride\_id)

WHERE charge\_status = 'Approved'

GROUP BY user\_details.platform, user\_details.age\_range, user\_details.download\_dt),

reviews AS (

SELECT 6 as funnel\_step,

'review' as funnel\_name,

user\_details.platform,

user\_details.age\_range,

user\_details.download\_dt,

COUNT (DISTINCT reviews.user\_id) as users\_count,

COUNT (DISTINCT reviews.ride\_id) as count\_rides

FROM reviews

JOIN user\_details

USING (ride\_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 reviews

ORDER BY funnel\_step, platform, age\_range, download\_dt;

**Tableau link**

[https://public.tableau.com/views/FSteps/Funnel?:language=en-GB&:display\_count=n&:origin=viz\_share\_link](https://public.tableau.com/views/Steps2/PotvsPop?:language=en-GB&:display_count=n&:origin=viz_share_link)