

Summary

In researching the performance of our newest iteration of the Metrocar app, I found several key areas we can focus on to expand our consumer base, and grow profits.

- We have steep drop-off rate of **50.76%** between a ride being accepted, and a ride being completed. I recommend canvassing drivers to discover what is causing the disconnect, and implementing needed changes to lower drop-off.
- Apple/ios has **48%** more users than Android users on the Metrocar app. I recommend marketing equally to both platforms, and even web users, to increase patronage.
- Each stage of our funnel showed the highest participation in the **35-44** year old range. This suggests our **target customers** are in the age range of **35-44** years of age.
- To adopt a price-surfing strategy, our distribution of ride requests are highest during these times of the day:
 - There is a surge in the morning from **8am** to **9am**
 - There is another surge in the afternoon from **4pm** to **7pm**. The hours 4pm to 5pm are the highest.

Context

The customer funnel represents key steps users complete in the metrocar ride experience. The dataset tracks unique users through their entire experience including downloading the app, to leaving a review after a ride is complete.

- **Funnel Step 1 – Downloads – 23,608 users**
 - This step counts of users who downloaded the Metrocar app
- **Funnel Step 2 – Sign Ups – 17,623 users**
 - This step counts users who create an account on the Metrocar app.
- **Funnel Step 3 – Ride Requested – 12,406 users**
 - Counts users who request a ride using the Metrocar app.
- **Funnel Step 4 – Ride Accepted – 12,278 users**
 - Counts users who's requested ride has been accepted.
- **Funnel Step 5 – Ride Completed – 6,233 users**
 - Counts users who's ride has been completed.
- **Funnel Step 6 – Payment – 6,233 users**
 - Counts users who have paid for their ride.
- **Funnel Step 7 – Review – 4,348 users**
 - Counts users who leave a review about their ride.

Results

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?
 - We should research:
 - Research is recommended to figure out why there is a 50% drop off from accepting a ride to completing a ride. Only 50.76% accepted rides are completed is.
 - The steps Download to Sign Up have a 25.36% drop off rate.
 - Sign Up to Ride Request steps have a 29.6% drop off rate.
 - 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?
 - ios/Apple has 48.16% more users of the Metrocar app than Android, the next highest platform.
 - ios/Apple – 7,471 rides accepted and 3,792 completed
 - Android – 3,580 rides accepted and 1,830 completed
 - Web – 1227 accepted and 611 completed
 - Only 10.86 % of total completed rides are web-based

Results

Business Questions Continued

- What age groups perform best at each stage of our funnel? Which age group(s) likely contain our target customers?
 - 35-44 - highest number of users – 29% step 1 – Downloads
 - 35-44 - highest number of users – 29% step 2 - Sign Ups
 - 35-44 – highest number of users – 29% step 3 - Ride Requested
 - 35-44 – highest number of users – 29% step 4 - Ride Accepted
 - 35-44 – highest number of users – 30% step 5 - Ride Completed
 - 35-44 – highest number of users – 29% step 6 - Payment
 - 35-44 – highest number of users – 31% step 7 – Review

****35-44 is most certainly our target group****
- What does the distribution of ride requests look like throughout the day, if we want to adopt a price-surfing strategy?
 - There is a surge in morning ride requests from 8am(275) to 9am(280)
 - There is another surge in afternoon ride requests from 4pm(290-peak) to 7pm(200). The hours of 4pm to 5pm are the highest, drop off is significant starting at 8pm.
- What part of our funnel has the lowest conversion rate? What can we do to improve this part of the funnel?
 - The Ride Accepted to Ride Completed step has the lowest conversion rate (51%) of all our steps.
 - We need to research the cause of this. We could possibly canvass drivers for insights to close this gap.

Appendix

Tableau:

https://public.tableau.com/views/Project2-Metrocar/FunnelSteps?:language=en-US&:display_count=n&:origin=viz_share_link

Code:

Aggregated Data for Tableau:

```
WITH user_ride_status AS (
  SELECT user_id
  FROM ride_requests
  GROUP BY user_id
),
total_users AS (
  SELECT
    a.platform AS platform,
    s.age_range AS age_range,
    a.download_ts::DATE AS download_date,
    COUNT(DISTINCT a.*) AS number_of_users_app_downloaded,
    COUNT(DISTINCT s.user_id) AS total_unique_users_signup,
    COUNT(DISTINCT urs.user_id) AS total_users_ride_requested,
    COUNT(DISTINCT r.ride_id) AS number_of_rides_requested, -- (ride is unique in ride request table)
    COUNT(DISTINCT CASE WHEN r.accept_ts IS NOT NULL THEN r.user_id END) AS
    rides_accepted_by_driver_user_wise,
    COUNT(DISTINCT CASE WHEN accept_ts IS NOT NULL THEN r.ride_id END) AS rides_accepted_by_driver, COUNT(DISTINCT
    CASE WHEN r.dropoff_ts IS NOT NULL THEN r.user_id END) AS
    unique_users_completed_ride,
    COUNT(DISTINCT CASE WHEN dropoff_ts IS NOT NULL THEN r.ride_id END) AS completed_rides, --(at
    unique ride level)
    COUNT(DISTINCT CASE WHEN t.charge_status = 'Approved' THEN r.user_id END) AS
    number_of_users_complete_payments, --(at user level, user who paid)
    COUNT(DISTINCT CASE WHEN t.charge_status = 'Approved' THEN r.ride_id END) AS
    number_of_rides_complete_payments, --(at ride level, rides which paid)
    COUNT(DISTINCT rw.user_id) AS number_of_users_provide_reviews, --(count of user who reviewed)
    COUNT(DISTINCT rw.ride_id) AS number_of_rides_received_reviews --(count of ride which had
    reviews)
  FROM app_downloads AS a
  LEFT JOIN signups AS s ON a.app_download_key = s.session_id
  LEFT JOIN user_ride_status AS urs ON s.user_id = urs.user_id
  LEFT JOIN ride_requests AS r ON s.user_id = r.user_id
  LEFT JOIN transactions AS t ON t.ride_id = r.ride_id
  LEFT JOIN reviews AS rw ON rw.user_id = s.user_id
  GROUP BY platform, age_range, download_date
),
funnel_steps AS (
  SELECT
    1 AS funnel_step,
    'Downloads' AS funnel_name,
    platform,
    age_range,
    download_date,
    number_of_users_app_downloaded AS user_count,
    null::int AS ride_count
  FROM total_users
```



Aggregated Data for Tableau cont.:

```
UNION
SELECT
2 AS funnel_step,
'Sign_UP' AS funnel_name,
platform,
age_range,
download_date,
total_unique_users_signup AS user_count,
null::int AS ride_count
FROM total_users
UNION
SELECT
3 AS funnel_step,
'Ride_Requested' AS funnel_name,platform,
age_range,
download_date,
total_users_ride_requested AS user_count,
number_of_rides_requested AS ride_count
FROM total_users
UNION
SELECT
4 AS funnel_step,
'Ride_Accepted' AS funnel_name,
platform,
age_range,
download_date,
rides_accepted_by_driver_user_wise AS user_count,
rides_accepted_by_driver AS ride_count
FROM total_users
UNION
SELECT
5 AS funnel_step,
'Ride_Completed' AS funnel_name,
platform,
age_range,
download_date,
unique_users_completed_ride AS user_count,
completed_rides AS ride_count
FROM total_users
UNION
SELECT
6 AS funnel_step,
'Payment' AS funnel_name,
platform,
age_range,
download_date,
number_of_users_complete_payments AS user_count,
number_of_rides_complete_payments AS ride_count
FROM total_users
UNION
SELECT
7 AS funnel_step,
'Review' AS funnel_name,platform,
age_range,
download_date,
number_of_users_provide_reviews AS user_count,
number_of_rides_received_reviews AS ride_count
FROM total_users
)
SELECT *
FROM funnel_steps
ORDER BY funnel_steps ASC;
```

Platform Analysis:

```
SELECT
platform,
downloads,
SUM(downloads) OVER () AS total_downloads,
downloads::float / SUM(downloads) OVER () AS pct_of_downloads
FROM (
SELECT
platform,
COUNT(*) AS downloads
FROM app_downloads
GROUP BY platform
) result;
```

Rides Requested per Hour:

```
SELECT
EXTRACT(HOUR FROM request_ts) AS time_hour,
COUNT(*) AS total_ride_request,
SUM(CASE WHEN cancel_ts IS NULL THEN 1 ELSE 0 END) AS completed_ride_at_each_hour,
SUM(CASE WHEN cancel_ts IS NOT NULL THEN 1 ELSE 0 END) AS rides_cancelled_at_each_hour
FROM ride_requests
GROUP BY time_hour
ORDER BY time_hour;
```

Rides Requested per Hour:

```
SELECT
su.age_range AS age_group,
COUNT(rr.ride_id) AS total_rides,
COUNT(DISTINCT rr.user_id) AS total_unique_user
FROM ride_requests rr
LEFT JOIN signups su
ON rr.user_id = su.user_id
GROUP BY age_range
```

