METROCAR ANALYSIS PROJECT

Metrocar is one of the top platforms in the USA that connects riders with drivers through a mobile application. It acts as an intermediary between riders and drivers, providing a user-friendly platform to connect them and facilitate the ride-hailing process.

The central goal of this project is to enhance the customer journey within our ride-hailing app, Metrocar. Utilizing data analysis techniques, our aim is to pinpoint areas for enhancement and refine the overall user experience. In this report, we will discuss the methodology employed for data analysis, focusing on funnel analysis as a key tool to track and understand the sequential steps customers take when interacting with our app. Our analysis is grounded in one year data spanning from March 2021 to April 2022.

Methodology:

For this analysis, we utilized SQL to query relevant data from our database, capturing various stages of the customer journey, including app downloads, signups, ride requests, ride completion, and review . To visualize and interpret the data effectively, we employed tools such as Tableau and Google Sheets.

Objectives and Key Questions:

In this analysis we have gone through some of the questions that are stated below:

- What are the specific drop off points of the users at each stage of the funnel and which steps should be further investigated to address these points effectively ?

- Which platforms should be prioritized for our marketing budget in the upcoming year to maximize users acquisition and engagement ?

- Which stage of our funnel exhibits the lowest conversion rate and what are the underlying reasons behind the lowest conversion rate? How can we optimize this stage to improve conversion rates?

- What factors contribute to the highest ride cancellation rate, and why are drivers receiving more negative ratings?

- What are the primary reasons behind the declined payments for rides, and how can we address these issues to ensure smoother transaction processes?

Funnel Analysis:

We have used a funnel analysis method to calculate the conversion rate for each stage. Funnel analysis serves as the cornerstone of our data analysis approach. It enables us to track and analyze how customers progress through different stages of interaction with our app, akin to a real-world funnel-wide at the top and narrow at the bottom. Our funnel analysis includes 5 stages i.e App Downloaded, Signups, Ride Requests, Ride Taken and Reviews.

Funnel stage	User_count	Conversion rate as per previous stage	Conversion rate as per download
App Downloads	23608	100.00%	100.00%
Signups	17623	74.65%	74.65%
Ride Requests	12406	52.55%	70.40%
Ride Completed	6233	26.40%	50.24%
Reviews	4348	18.42%	69.76%

- App Downloads to Signups: Out of 23608 users who downloaded the app, there is a drop off rate of (100.00% - 74.65%) 25.35% from app downloads to signups. This drop off may be attributed to potential issues during the signup process, including privacy concerns and apprehension about sharing personal information.

- Signups to Ride Requests: The conversion rate from signups to ride requests is 52.55% , indicating the drop off of 22.10% (74.65%-52.55%) from the previous stage. This drop off suggests that users who signed up didn't proceed to ride request rides. Possible reasons for this include encountering issues with the app or hesitancy due to pricing information displayed for rides.

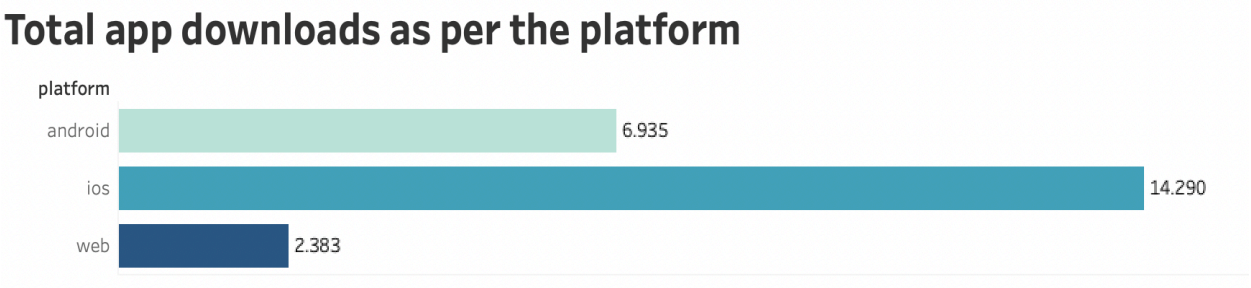
- Ride Requests to Ride Completed: The conversion rate at this stage is 26.15% (52.55%-26.40), indicating the highest drop off. Almost 50% of users drop in this stage. The reason behind this drop off is because the user waited too long for the acceptance of the ride, due to which they cancel the ride and drop off.

- Review stage:. The conversion rate for reviews based on completed rides is 18.42%, with a drop off rate of 7.98% (26.40%-18.42) from the previous stage. This indicates that not all the users who completed rides provided reviews. The reason behind less reviews might be because most of the people thought that it is the lengthy method to write the review.

The suggestion from this analysis is that the company should prioritize addressing the stage with the lowest conversion rate, focusing on improving the ride request to ride taken process by simplifying booking, monitoring driver acceptance rates, and offering incentives for the driver with the high performance.

Platforms to be focused:

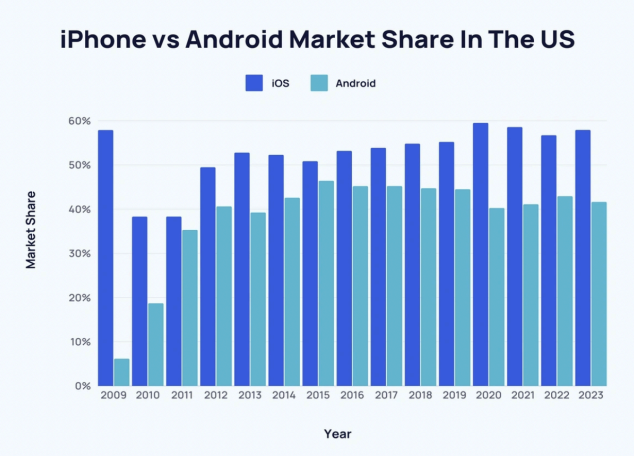
Metrocar is one of the mobile based apps which includes only 3 Platforms namely; IOS, android and web. The total app downloaded by the users as per IOS is 14290 which is the highest among the three and the lowest is the web which is 2383.



Platform	Total downloaded	Total signups	Total ride taken	Total reviews
android	6935	5148	3619	1273
ios	14290	10728	7550	2651
web	2383	1747	1237	424

We have gone through the funnel analysis of this platform during our analysis. And we found out the following information:

- Funnel analysis of the platform reveals user drop-off every stage on each platform.
- No significant platform-based variations observed; IOS is the most utilized, followed by Android and then web.
- IOS and Android are preferred for ride bookings due to their convenience, likely because users often book rides while on the way.



Real data from the USA indicates a higher preference for IOS over Android, this is why the number of signups is more from IOS.

Funnel with the lowest conversion rate:

The ride completed stage of the funnel has the lowest conversion rate i.e 50.24% of ride requests being completed. Out of 12406 ride requests, only 6233 were completed, resulting in a nearly 50% drop-off. Analysis reveals that prolonged wait times for ride acceptance contribute to this drop-off.

Avg_diff_request_accept	Avg_diff_accept_pickup	Avg_diff_request_cancel
0:06:53	0:08:30	0:12:33

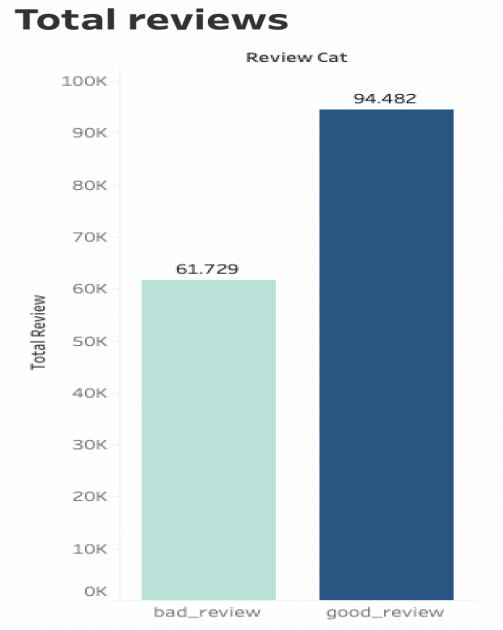
- The average wait time from request to acceptance of the ride is around 7 minutes, average wait time from acceptance of ride to pickup is 8.5 minutes and average wait time from request time to cancellation time is more than 12.5 minutes, which means most of the users who canceled the ride waited for more than 12 minutes.

- Users often waited longer than average before canceling the ride, indicating dissatisfaction with acceptance times.

- To improve conversion rates, we suggest conducting driver satisfaction surveys to gauge job satisfaction, motivation, and pay scale satisfaction. Incentives can be offered based on ride performance to motivate existing drivers, reducing the need for additional hiring and expenses.

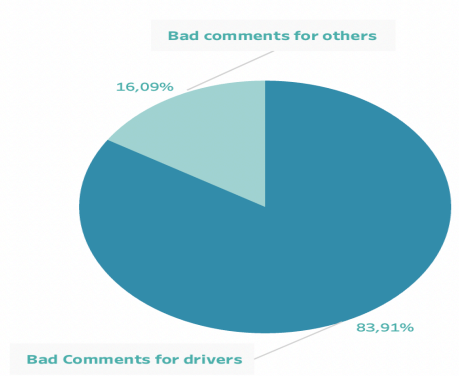
Reasons behind the highest cancellation rate of the ride and drivers getting the bad comments:

We have already discussed earlier the reason behind the highest cancellation rate is attributed to drivers' delays in accepting rides



The analysis shows the total number of bad and good reviews as per the ratings that are given by the users. From this we can see that the good reviews are more than the bad ones. But most of the bad reviews are the bad comments which are related to the drivers.

Analysis indicates that a significant portion of negative reviews are related to driver behavior and service quality. Approximately 84% of all negative comments are directed towards drivers, Based on user ratings and feedback.

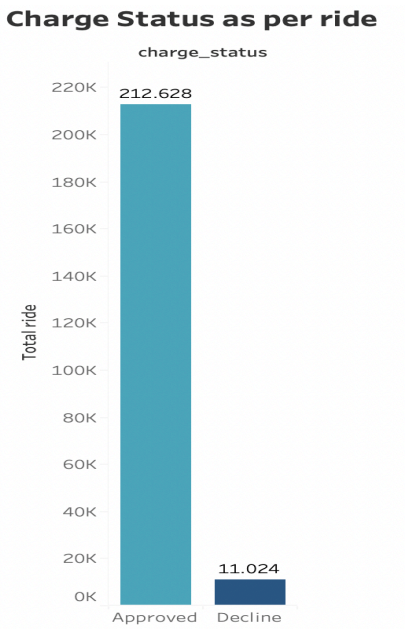


Reviews
A complete disaster. The driver canceled the ride last minute, leaving me stranded in an unfamiliar location.
Extremely disappointed. The driver was rude and unprofessional, and the ride was uncomfortable.
Horrible service. The driver was reckless and drove well above the speed limit.
A nightmare experience. The driver was rude, and the car was in terrible condition.
Unreliable and unprofessional. The driver was late, and the app crashed repeatedly.

The company should go through the comments that are provided by the users and the company should focus on driver training and feedback mechanism, driver incentives and recognition and improved communication and transparency.

Reason behind the declined payment of the ride:

Payment is the last part to complete the ride. There are some charges that were declined during the ride. The ride was taken and completed but charge status was declined.



From this we can see that the number of rides that declined the payment is 11024. The reason behind the decline of the payment might be because of the mode of the payment and might be because of the expired card or insufficient funds. Since there is no further information regarding the mode of payment we can also say that it might be because of the technical glitches . Another issue that might be is there might be a problem with the app.

So, the company should make a secure and safe medium of payment for the users in that way where there are not any issues related with the money.

Recommendations and Insights:

Some of the insights and recommendation we have found out during our analysis are mentioned below:

1. Optimizing ride acceptance and pickup times:
    - Implement advanced driver tracking functionality in the app to provide transparency and accountability during each stage of the ride.
    - Provide incentives for drivers based on their performance and customer ratings to encourage better service and reduce ride cancellations.
2. Driver Training and Incentives:
    - Offer training programs for drivers to improve their behavior and customer service skills.
    - Introduce incentives for drivers with higher ratings and positive reviews to motivate them to maintain good performance.
3. Platform Optimization:
    - Focus on enhancing the mobile app experience for IOS and Android platforms as they are preferred over web platforms for ride booking.
    - Allocate resources to improve the functionality, user interface, performance of the mobile app to cater to the majority of users and remove web platforms from the app.
4. Payment Optimization:
    - Update the payment method to include multiple payment modes and ensure a seamless payment process with minimal technical glitches.
    - Integrate features to verify payments directly with the bank to reduce declined payments and increase user confidence in the payment process.
5. Promotional Offers:
    - Introduce attractive offers such as discounts on the first ride to attract new users.
    - Provide targeted discounts based on age groups or other demographics to encourage more users to use the app for their rides.
6. Streamlined Review Process:
    - Simplify the review process by implementing a one- click review system or automatically promoting users to leave a review after completing a ride.
    - Make the review process more user-friendly and less-time consuming to increase the likelihood of users leaving feedback.

By implementing these strategies, the ride booking platform can address the identified issues, improve user satisfaction, and ultimately increase user retention and conversion rates.

SQL Queries

```
SELECT COUNT(*) FROM app_downloads;

WITH total_download_signups AS(SELECT COUNT(app_downloads.app_download_key) AS
total_app_downloaded,
                               COUNT(signups.user_id) AS total_user_signups
FROM app_downloads
LEFT JOIN signups ON
app_downloads.app_download_key=signups.session_id)

SELECT (total_user_signups::float* 100.0/total_app_downloaded )AS constumer_appdownloads_signups,
total_user_signups
FROM total_download_signups;

WITH total_signups_ride AS(SELECT COUNT(DISTINCT signups.user_id) AS total_user_signups,
                           COUNT(DISTINCT ride_requests.user_id) AS total_ride_taken
FROM signups
LEFT JOIN app_downloads ON
signups.session_id=app_downloads.app_download_key
LEFT JOIN ride_requests ON
signups.user_id=ride_requests.user_id)

SELECT total_ride_taken::float * 100.0 / total_user_signups AS customer_signups_ride, total_ride_taken
FROM total_signups_ride
;

WITH ride_taken AS(SELECT COUNT(DISTINCT request_ts),
     (SELECT COUNT(DISTINCT user_id)FROM ride_requests) AS total_ride_requested,
                               COUNT(DISTINCT user_id) AS user_id_ridetaken

FROM ride_requests
WHERE dropoff_ts IS NOT NULL AND pickup_ts IS NOT NULL)

SELECT  user_id_ridetaken,
                                    user_id_ridetaken::float * 100.0 /total_ride_requested AS
customer_ride_taken
FROM ride_taken
;

WITH total_review_user AS (SELECT (SELECT COUNT(DISTINCT user_id)FROM ride_requests WHERE
dropoff_ts IS NOT NULL AND pickup_ts IS NOT NULL ) AS total_ride_requested,
                                                    COUNT(DISTINCT reviews.user_id) AS
total_review
FROM ride_requests
LEFT JOIN reviews ON
ride_requests.ride_id=reviews.ride_id)

SELECT
   total_review,
   total_review::float * 100.0 / total_ride_requested AS customer_review_ride
FROM total_review_user;


SELECT COUNT(DISTINCT app_download_key) AS total_download,
platform
FROM app_downloads
GROUP BY platform
```

```
WITH funnel_data AS (
    SELECT
        platform,
        COUNT(DISTINCT app_download_key) AS total_downloaded,
          COUNT(DISTINCT CASE WHEN signups.user_id IS NOT NULL THEN app_download_key END) AS
total_signed_up,
          COUNT(DISTINCT CASE WHEN ride_requests.request_ts IS NOT NULL THEN app_download_key
END) AS total_requested,
          COUNT(DISTINCT CASE WHEN ride_requests.dropoff_ts IS NOT NULL THEN app_download_key
END) AS total_taken,
        COUNT(DISTINCT CASE WHEN reviews.review_id IS NOT NULL THEN app_download_key END) AS
total_reviewed
    FROM
        app_downloads
    LEFT JOIN
        signups ON app_downloads.app_download_key = signups.session_id
    LEFT JOIN
         ride_requests ON signups.user_id = ride_requests.user_id
    LEFT JOIN
        reviews ON ride_requests.user_id = reviews.user_id
    GROUP BY
        platform
)
SELECT
    platform,
    total_downloaded,
    total_signed_up,
    total_requested,
    total_taken,
    total_reviewed
FROM
    funnel_data;


WITH funnel_data AS (
    SELECT
        age_range,
        COUNT(DISTINCT app_download_key) AS total_downloaded,
          COUNT(DISTINCT CASE WHEN signups.user_id IS NOT NULL THEN app_download_key END) AS
total_signed_up,
          COUNT(DISTINCT CASE WHEN ride_requests.request_ts IS NOT NULL THEN app_download_key
END) AS total_requested,
          COUNT(DISTINCT CASE WHEN ride_requests.dropoff_ts IS NOT NULL THEN app_download_key
END) AS total_taken,
        COUNT(DISTINCT CASE WHEN reviews.review_id IS NOT NULL THEN app_download_key END) AS
total_reviewed
    FROM
        app_downloads
    LEFT JOIN
        signups ON app_downloads.app_download_key = signups.session_id
    LEFT JOIN
         ride_requests ON signups.user_id = ride_requests.user_id
    LEFT JOIN
        reviews ON ride_requests.user_id = reviews.user_id
    GROUP BY
        age_range
)
SELECT
    age_range,
    total_downloaded,
    total_signed_up,
    total_requested,
```

```
   total_taken,
   total_reviewed
FROM
   funnel_data;


WITH review_category AS (SELECT
      CASE WHEN rating <= 2 THEN 'bad_review'
                           ELSE 'good_review'
      END AS review_cat


FROM reviews)

SELECT COUNT(*) AS total_review, review_cat FROM review_category
GROUP BY review_cat
;


WITH review_category AS (SELECT review_id, review,
      CASE WHEN rating <= 2 THEN 'bad_review'
                           ELSE 'good_review'
      END AS review_category

FROM reviews),

comment_category AS (SELECT *, CASE WHEN review LIKE'%driver%' THEN 'bad_comments_for_drivers'
                                                                ELSE 'bad_comments_not_for_drivers'
        END AS comment_category
FROM review_category )

SELECT COUNT(*) AS total_comments,
                            comment_category
FROM comment_category
GROUP BY comment_category;


WITH review_category AS (SELECT review_id, review,
      CASE WHEN rating <= 2 THEN 'bad_review'
                           ELSE 'good_review'
      END AS review_category

FROM reviews)
SELECT review,
                            COUNT(review_id) AS total_review
FROM review_category
WHERE review_category = 'bad_review'
GROUP BY review
ORDER BY 2 DESC
;

SELECT
      AVG(accept_ts - request_ts) AS avg_diff_request_accept,
      AVG(pickup_ts - accept_ts) AS avg_diff_accept_pickup,
      AVG(cancel_ts - request_ts) AS avg_diff_request_cancel
FROM ride_requests
;
```