

Funnel Analysis

By: Satakshi Salaria

Summary

Metrocar functions as a ride-hailing service akin to industry giants such as Uber and Lyft. The core concept of Metrocar revolves around a mobile application that serves as a linkage between passengers and drivers. Acting as an intermediary, Metrocar furnishes a user-friendly platform that streamlines the interaction between riders and drivers, thereby enhancing the ease and efficiency of the ride-hailing process.

Context

The typical customer journey for Metrocar encompasses the following stages:

- **App Download:** A user downloads the Metrocar app from either the App Store or Google Play Store.
- **Signup:** The user creates an account within the Metrocar app, providing their name, email, phone number, and payment details.
- **Request Ride:** The user opens the app and requests a ride by specifying their pickup location, destination, and the number of riders (2 to 6).
- **Driver Acceptance:** A nearby driver receives and confirms the ride request.
- **Ride:** The driver arrives at the pickup location, and the user enters the vehicle for the trip to their destination.
- **Payment:** Following the ride, the app automatically charges the user, and a receipt is sent to their email.
- **Review:** The user is encouraged to rate their driver and offer a review of their ride experience.

Key Findings

Funnel Analysis Metrics:

- **Funnel Step:** Represents distinct stages within the customer journey.
- **Funnel Name:** Identifies each corresponding step in the customer journey.
- **User Count:** Total count of users at every stage in the funnel.
- **Lag:** Indicates the variance in user count between the current and prior steps, reflecting user progression.
- **Difference (Diff):** Depicts the absolute change in user count relative to the initial step (download), signifying user gain or loss at each stage.
- **Conversion Rate:** Calculated as the ratio of users transitioning from one step to the subsequent one. For instance, a 0.7465 conversion rate for "sign-up" signifies that 74.65% of app downloaders proceeded to sign up.
- **Drop-off Percent:** Complementary to the conversion rate, showing the proportion of users who do not advance to the following step. For example, a 0.2535 drop-off percent for "sign-up" indicates that 25.35% of app downloaders did not complete the sign-up process.

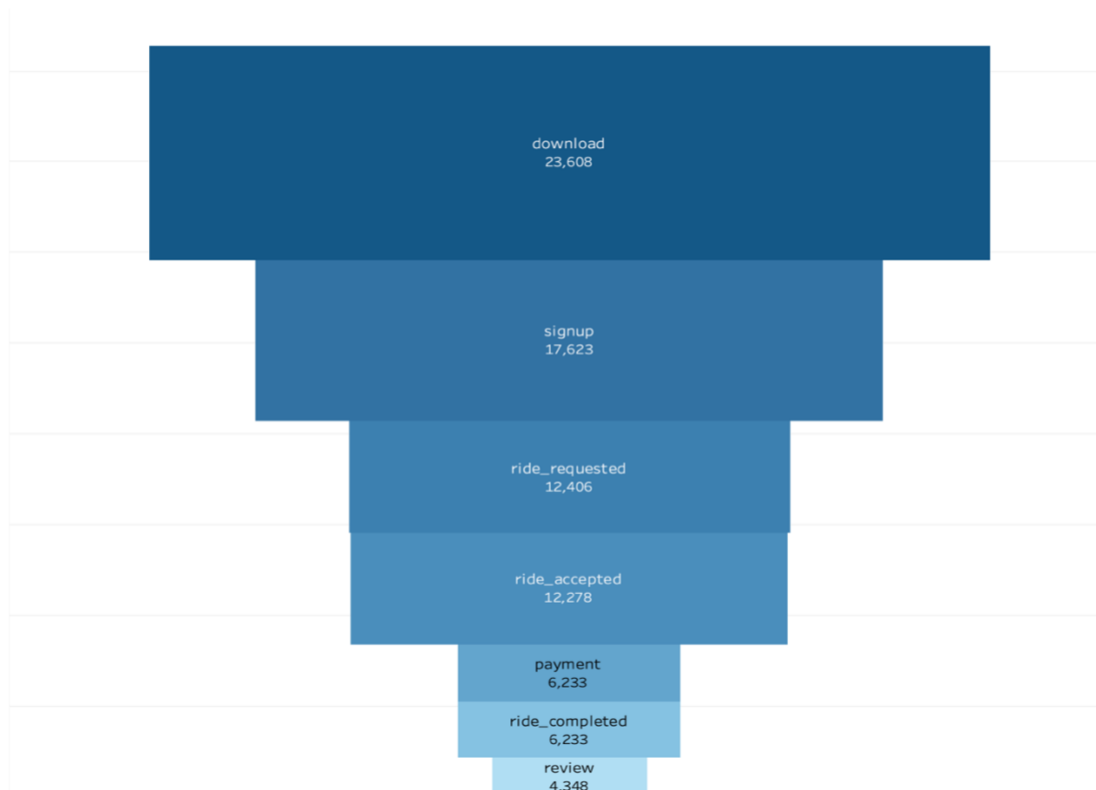
Out[30]:

	funnel_step	funnel_name	user_count	lag	diff	conversion_rate	dropoff_percent
0	1	app_download	23608	NaN	NaN	NaN	NaN
1	2	sign_up	17623	23608.0	5985.0	0.7465	0.2535
2	3	ride_requested	12406	17623.0	5217.0	0.7040	0.2960
3	4	ride_accepted	12278	12406.0	128.0	0.9897	0.0103
4	5	ride_completed	6233	12278.0	6045.0	0.5077	0.4923
5	6	payment	6233	6233.0	0.0	1.0000	0.0000
6	7	review	4348	6233.0	1885.0	0.6976	0.3024

Funnel Analysis:

- **Download:** Initiated with 23,608 users downloading the app, signifying the beginning of the funnel.
- **Sign-Up:** Demonstrated a strong 74.65% conversion rate as app downloaders progressed to sign up.
- **Ride Requested:** Displayed consistent engagement with a 70.40% conversion rate from sign-up to ride request.
- **Ride Accepted:** Showcased an impressive 98.97% conversion rate from ride request to ride acceptance, indicating minimal drop-off.
- **Payment:** Encountered a significant decline with only 50.77% of users who accepted rides proceeding to complete them, suggesting an area requiring improvement.
- **Ride completed:** Achieved a flawless 100% conversion rate from ride completion to payment, indicating a smooth transactional experience.
- **Review:** Demonstrated a 69.76% conversion rate from payment to review yet highlighted an area for improvement with a notable 30.24% drop-off rate.

Funnel Analysis

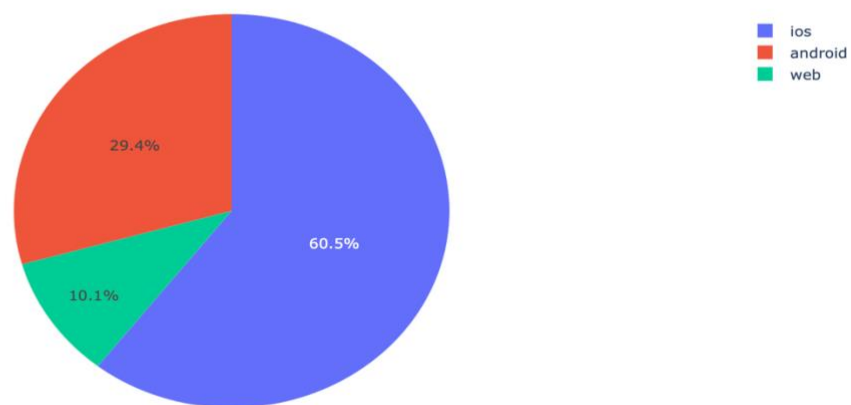


Segmentation of customers:

Platform:

Metrocar currently serves three platforms: iOS, Android, and Web. iOS maintains the majority share, accounting for 60.53%, followed by Android at 29.38%.

Percent_of_downloads



Age:

Insights on Age Group Performance within the Funnel:

- Notably, the age bracket of "35-44" emerges prominently, constituting 29.40% of signups, closely followed by "25-34" at 19.56%. Furthermore, the "Unknown" category, encompassing 30.10% of signups, likely comprises individuals who opted not to disclose their age.
- Significantly, the age groups "18-24" and "45-54" demonstrate comparatively lower signup rates, accounting for 10.58% and 10.36%, respectively.
- The "Unknown" category, representing 30.10% of signups, holds potential insights if users choose to divulge their age information, which could offer valuable analytical data.

Out [60]:

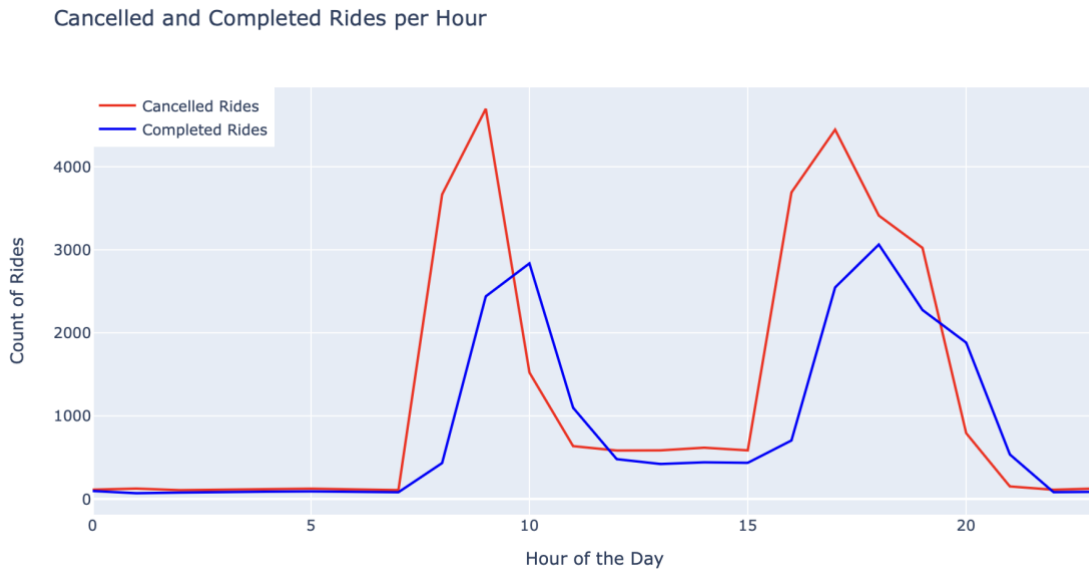
	age_range	signups	total_signups	pct_of_signups
0	18-24	1865	17623	0.1058
1	25-34	3447	17623	0.1956
2	35-44	5181	17623	0.2940
3	45-54	1826	17623	0.1036
4	Unknown	5304	17623	0.3010

Cancelled and Completed Rides

From the provided data on cancelled and completed rides per hour throughout the day, some patterns and insights can be observed:

- **Morning Peak in Cancellations:** There's a noticeable surge in cancelled rides during the morning rush hours between 8 AM and 10 AM. The cancellation rate significantly escalates during this period, peaking at 4697 cancellations at 9 AM.
- **Evening Rush Hour Peak:** Similar to the morning peak, there's another substantial increase in cancelled rides during the evening rush hours, specifically between 4 PM and 6 PM. The highest spike in cancellations occurs at 5 PM, reaching 4446 cancellations.
- **Lower Cancellations during Late Night and Early Morning:** In contrast, late-night hours from 11 PM to 5 AM exhibit comparatively lower cancellation rates. These hours generally experience fewer cancellations compared to the daytime peaks.
- **Variation in Completed Rides:** Completed rides show fluctuations throughout the day. There's a significant increase in completed rides during peak commute hours in the morning and evening, with a peak at 9 AM and another significant peak at 5 PM.
- **Lowest Completed Rides late at Night:** Completed rides tend to decrease during late-night hours from 11 PM to 5 AM, following a pattern similar to the lower cancellation rates observed during these hours.

These patterns suggest that specific times of the day, particularly during morning and evening rush hours, experience heightened activity with a surge in both completed and cancelled rides. Understanding these patterns can help in optimizing resource allocation and service provision to meet user demands during peak hours and potentially minimize cancellations during these periods.



Recommendation

Upon scrutinizing user demographics, we noticed the following:

- **Refine Onboarding Process:** Streamline the transition from "app_download" to "sign_up" to alleviate the 25.35% drop-off. Simplify the sign-up procedure to enhance user retention by minimizing complexities.
- **Improve Ride Acceptance Journey:** Investigate and refine the user journey from "ride_requested" to "ride_accepted" to mitigate the 10.03% drop-off. Prioritize improvements that facilitate seamless ride acceptance, focusing on enhancing user experience during this phase.
- **Tackle Ride Completion Challenges:** Analyze the significant 49.23% drop-off between these stages. Conduct comprehensive supply-side analysis, encourage comprehensive user reviews, leverage sentiment analysis for deeper insights, and align demand-side and supply-side funnel analysis for holistic improvements.

- **Ensure Platform Consistency:** Ensure consistent attention and support across iOS, Android, and Web platforms due to observed similar conversion and drop-off rates. Aim to maintain uniform user experiences across different platforms.
- **Strategic Marketing Resource Allocation:** Allocate marketing resources proportionate to user platform shares, paying particular attention to the noticeable difference in user volumes between iOS and Android.
- **Targeted Marketing and Enhanced Experience:** Tailor marketing strategies and experience enhancements toward the "35-44" and "25-34" age brackets, optimizing user journeys based on their higher sign-up percentages and engagement.
- **Data-Driven Personalization Strategies:** Utilize user data, including canceled rides, and encourage the provision of age information during sign-up to tailor experiences. Create incentives or benefits to prompt data sharing and offer more personalized journeys.
- **Strategic Surge Pricing:** Implement surge pricing during peak hours (8 AM - 10 AM and 4 PM - 8 PM) to effectively manage revenue streams during periods of heightened demand while maintaining a balanced user experience.

These recommendations aim to target specific areas identified in the funnel analysis, aiming to enhance user experiences, mitigate dropout rates, and strategically foster revenue growth for Metrocar.

Conclusion

In summary, the comprehensive funnel analysis has provided crucial insights into Metrocar's customer journey. The recommendations presented aim to optimize the sign-up process, enhance ride acceptance, and address significant drop-off points during ride completion and review stages. Additionally, these suggestions emphasize the importance of equitable treatment across platforms and strategic surge pricing to elevate user acquisition, retention, and overall customer satisfaction. Implementing these recommendations can significantly contribute to refining the user experience, enhancing service quality, and fostering sustainable growth for Metrocar.

Appendix

SQL Query

- How many times was the app downloaded?

```
SELECT COUNT(app_download_key) as total_downloads
FROM app_downloads;
```
- How many users signed up on the app?

```
SELECT COUNT(user_id) as total_signups
FROM signups;
```
- How many rides were requested through the app?

```
SELECT COUNT(ride_id) as total_rides
FROM ride_requests;
```
- How many rides were requested and completed through the app?

```
SELECT COUNT(ride_id) as rides_completed
FROM transactions;
```
- How many rides were requested and how many unique users requested a ride?

```
SELECT COUNT(DISTINCT user_id) as unique_users
FROM ride_requests;
```
- 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;
```
- How many rides were accepted by a driver?

```
SELECT COUNT(accept_ts) AS total_accepted
FROM ride_requests;
```
- 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';
```
- 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;

```

- 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;

```

- 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';

```

- what percentage these users requested a ride?

```

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;

```

- what percentage 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 = rr.user_id
AND rr.dropoff_ts IS NOT NULL;

```

- what are the user-level conversion rates for the first 3 stages of the funnel ?

```

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;
```

- what are the user-level conversion rates for the first 3 stages of the funnel ?
 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;

- 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;
```

- 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;
```

- Total_users_ride_requested and Total_users_ride_completed

```
WITH user_ride_status AS (
SELECT
    user_id,
    MAX(
        CASE
            WHEN dropoff_ts IS NOT NULL
            THEN 1
            ELSE 0
        END
    ) AS ride_completed
FROM ride_requests
GROUP BY user_id
)
SELECT
    COUNT(*) AS total_users_ride_requested,
    SUM(ride_completed) AS total_users_ride_completed
FROM user_ride_status;
```

- Waiting_time
SELECT accept_ts - request_ts as waiting_time, rr.*
FROM ride_requests rr;

- Funnel code:
SELECT COUNT(*), 'app_downloads' AS funnel_steps
FROM app_downloads

UNION

```
SELECT COUNT(*), 'signups' AS funnel_steps
FROM signups
```

UNION

```
SELECT COUNT(DISTINCT user_id), 'rides_requested' AS funnel_steps
FROM ride_requests
```

UNION

```
SELECT COUNT(DISTINCT user_id), 'rides_completed' AS funnel_steps
FROM ride_requests
WHERE cancel_ts IS NULL;
```

- or with lag function

```
WITH TEMP1 AS (
  SELECT COUNT(*) AS counter, 'app_downloads' AS funnel_steps
  FROM app_downloads
```

```
UNION
```

```
SELECT COUNT(*) AS counter, 'signups' AS funnel_steps
FROM signups
```

```
UNION
```

```
SELECT COUNT(DISTINCT user_id) AS counter, 'rides_requested' AS funnel_steps
FROM ride_requests
```

```
UNION
```

```
SELECT COUNT(DISTINCT user_id) AS counter, 'rides_completed' AS funnel_steps
FROM ride_requests
WHERE cancel_ts IS NULL
)
SELECT *,
lag(counter,1) OVER(ORDER BY counter DESC) AS lag_num,
counter - lag(counter,1) OVER(ORDER BY counter DESC) AS difference
FROM temp1;
```

- 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;

Jupyter notebook link

- file:///Users/satakshisalaria/Documents/Mastery_Project3/Untitled3.html
- <file:///Users/satakshisalaria/Downloads/Untitled2.html>

Tableau Link

- https://public.tableau.com/app/profile/satakshi.salaria/viz/Book1_16999744624240/FunnelAnalysis?publish=yes



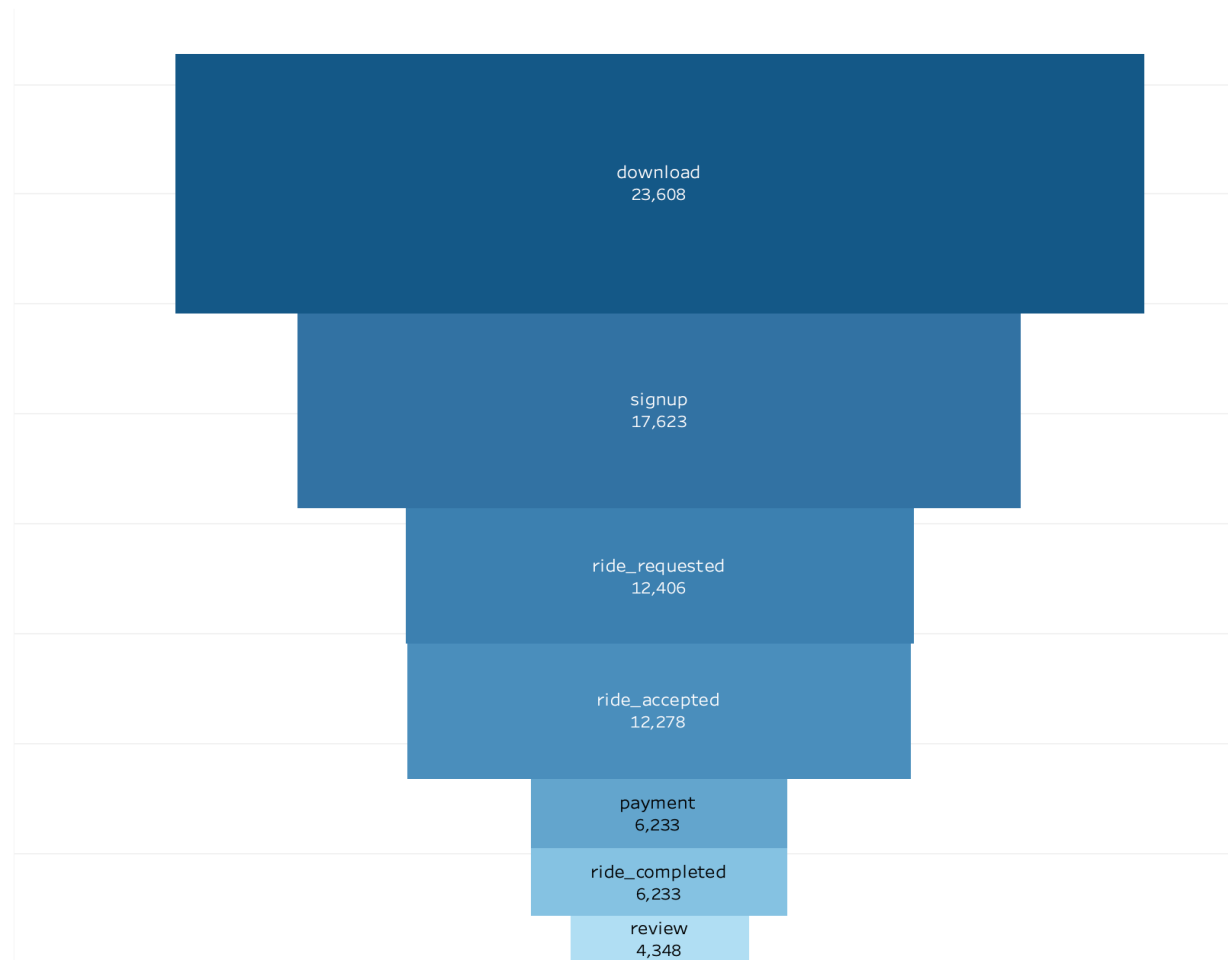
Funnel Analysis

BY Satakshi Salaria

Steps for Funnel Analysis

- App Download
- Signup
- Request Ride
- Driver Acceptance
- Ride
- Payment
- Review

Funnel Analysis

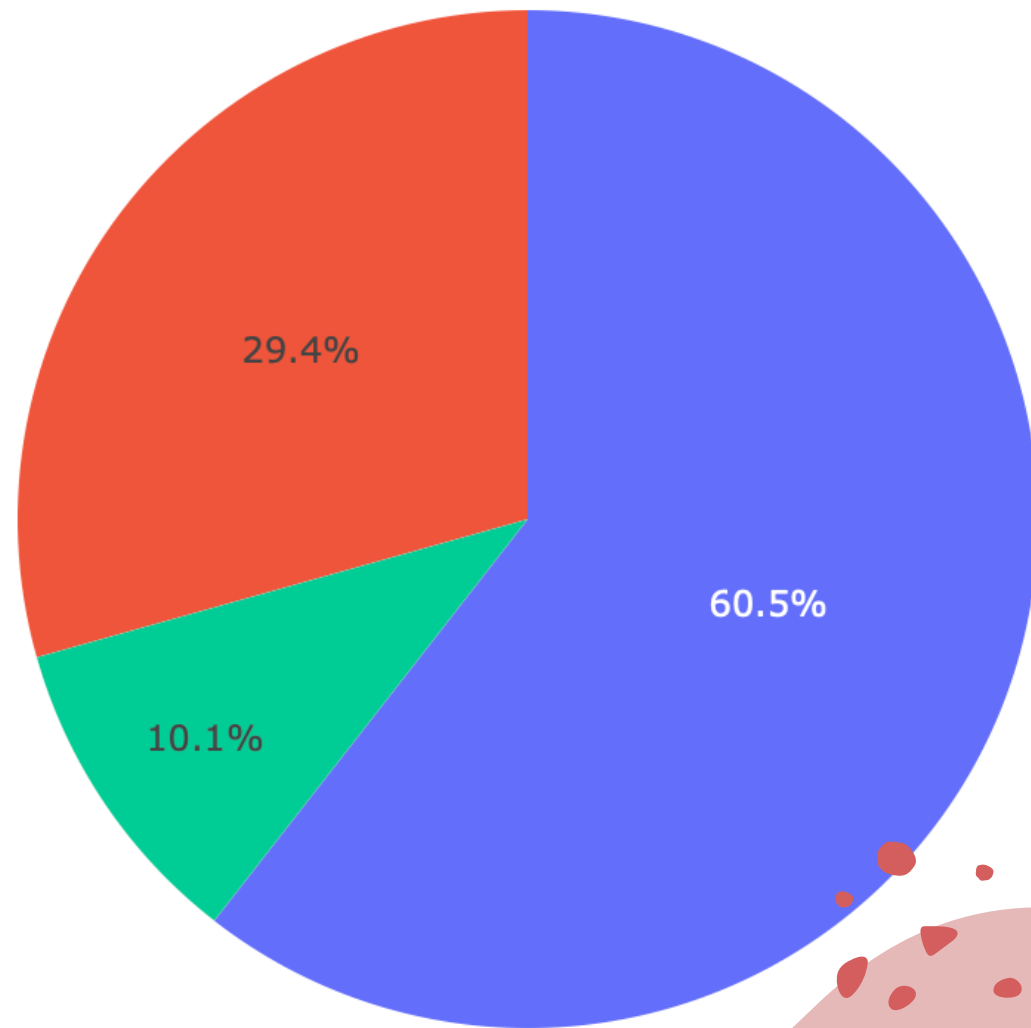


Platform

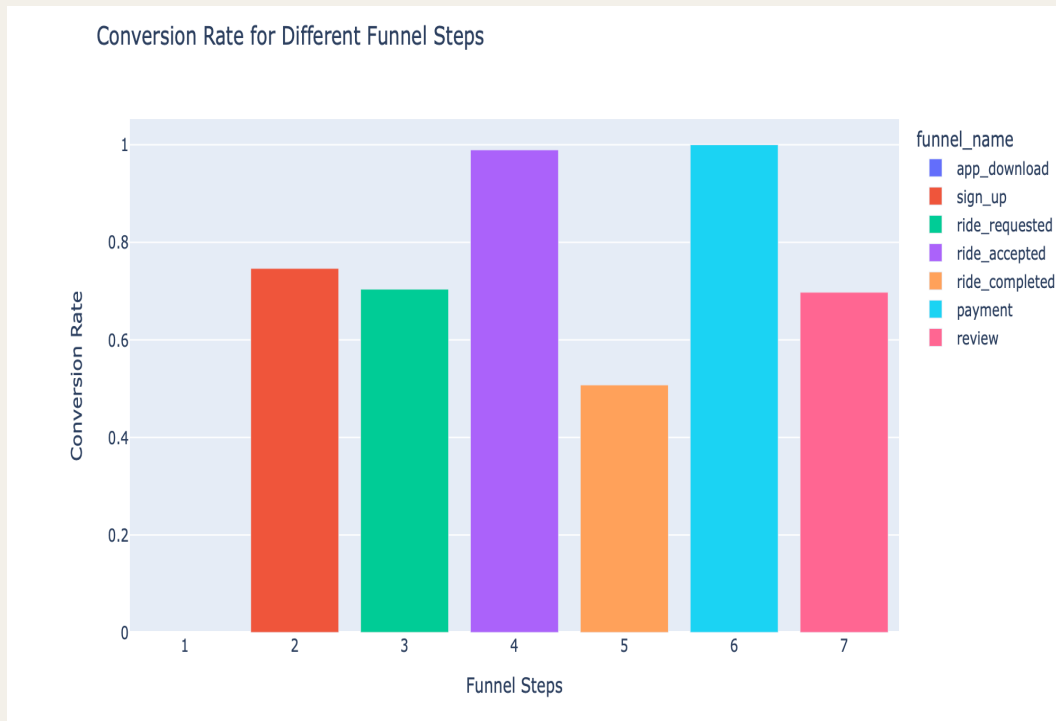
Ios - 60.5%

Android - 29.4%

Web - 10.1%



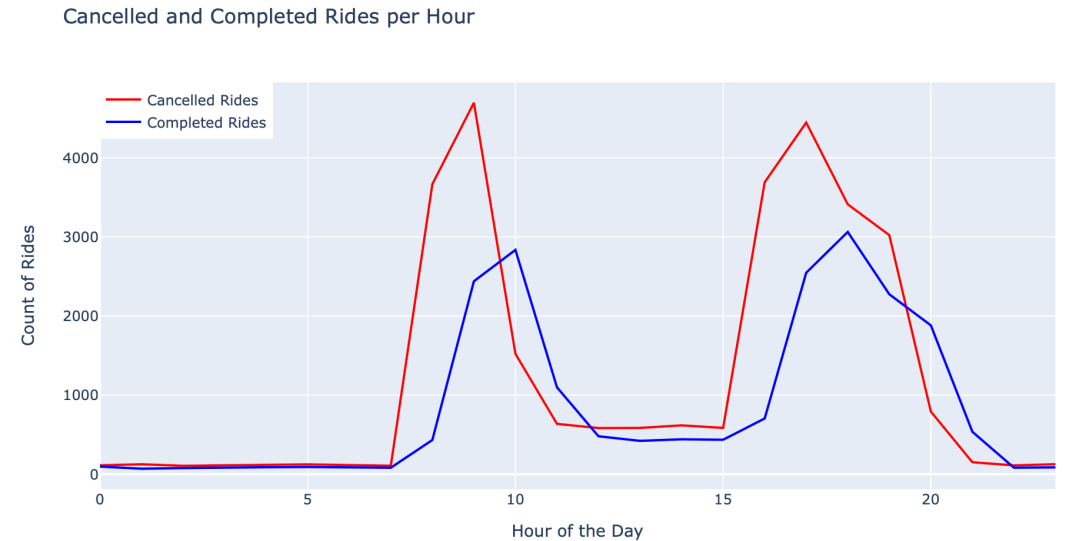
Conversion Rate for different funnel steps



- App Download: this stage serves as the starting point, and there's no conversion rate because it's the base step.
- Signup :This stage signifies the transition from app download to signing up for the service. The conversion rate of 0.7465 means that 74.65% of users who downloaded the app proceeded to sign up.
- Ride Requested : After signing up, this stage involves users making a ride request. The conversion rate of 0.704 indicates that 70.4% of users who signed up went on to request a ride.
- Ride completed : This stage represents the completion of the ride by users. The conversion rate of 0.5077 means that about 50.77% of users who had their ride accepted went on to complete the ride.
- Payment : This stage indicates successful payment after completing the ride. The conversion rate of 1.00 signifies that 100% of users who completed the ride also made the payment.
- Review : The final stage involves users providing a review after making the payment. The conversion rate of 0.6976 suggests that approximately 69.76% of users who made the payment proceeded to leave a review.

Cancelled and Completed Rides per Hour

Implement surge pricing during peak hours (8 AM - 10 AM and 4 PM - 8 PM) to effectively manage revenue streams during periods of heightened demand while maintaining a balanced user experience.



Recommendation

- **Refine Onboarding Process:** Streamline the transition from "app download" to "sign up" to alleviate the 25.35% drop-off. Simplify the sign-up procedure to enhance user retention by minimizing complexities.
- **Improve Ride Acceptance Journey:** Investigate and refine the user journey from "ride requested" to "ride accepted" to mitigate the 10.03% drop-off. Prioritize improvements that facilitate seamless ride acceptance, focusing on enhancing user experience during this phase.
- **Tackle Ride Completion Challenges:** Analyze the significant 49.23% drop-off between these stages. Conduct comprehensive supply-side analysis, encourage comprehensive user reviews, leverage sentiment analysis for deeper insights, and align demand-side and supply-side funnel analysis for holistic improvements.
- **Ensure Platform Consistency:** Ensure consistent attention and support across iOS, Android, and Web platforms due to observed similar conversion and drop-off rates. Aim to maintain uniform user experiences across different platforms.
- **Strategic Marketing Resource Allocation:** Allocate marketing resources proportionate to user platform shares, paying particular attention to the noticeable difference in user volumes between iOS and Android.
- **Targeted Marketing and Enhanced Experience:** Tailor marketing strategies and experience enhancements toward the "35-44" and "25-34" age brackets, optimizing user journeys based on their higher sign-up percentages and engagement.
- **Data-Driven Personalization Strategies:** Utilize user data, including canceled rides, and encourage the provision of age information during sign-up to tailor experiences. Create incentives or benefits to prompt data sharing and offer more personalized journeys.
- **Strategic Surge Pricing:** Implement surge pricing during peak hours (8 AM - 10 AM and 4 PM - 8 PM) to effectively manage revenue streams during periods of heightened demand while maintaining a balanced user experience.

Conclusion

In summary, the comprehensive funnel analysis has provided crucial insights into Metrocar's customer journey. The recommendations presented aim to optimize the sign-up process, enhance ride acceptance, and address significant drop-off points during ride completion and review stages.

Additionally, these suggestions emphasize the importance of equitable treatment across platforms and strategic surge pricing to elevate user acquisition, retention, and overall customer satisfaction. Implementing these recommendations can significantly contribute to refining the user experience, enhancing service quality, and fostering sustainable growth for Metrocar.