Metrocar - Funnel Analysis

Metrocar - Funnel Analysis

By: Data Analyst Team
Date: Oct 20, 2023,

1. Introduction

Metrocar is a ride-sharing app (similar to Uber/Lyft). Metrocar's business model is based on a platform that connects riders with drivers through a mobile application. Metrocar is an intermediary between riders and drivers, providing a user-friendly platform to connect them and facilitate the ride-hailing process.

Stakeholders had asked the data analysis team to analyze Metrocar's data and identify areas for improvement and optimization. For example, Metrocar wants to analyze the percentage of users who download the app but do not complete the registration process, or the percentage of users who request a ride but cancel before the driver arrives.

Funnel analysis was used to identify where users drop off or convert, helping to ultimately increase desired outcomes, such as sales, sign-ups, or conversions. It is widely used in e-commerce, marketing, and product development to drive growth and revenue.

2. Metrocar Funnel Analysis

Funnel analysis is a method in data analysis used to track and understand the sequential steps or stages that users or customers go through when interacting with a product, service, or website. It's called a "funnel" because the shape of the analysis resembles that of a real-world funnel – wide at the top and narrow at the bottom.

Objective

Drive growth and revenue by increasing desired outcomes, such as sales, sign-ups, or conversions.

Dataset structure

All the Metrocar data is available in a dataset stored in a relational database system. This dataset includes 5 tables as below:

- app_downloads: contains information about app downloads
    - app_download_key: unique id of an app download
    - platform: ios, android or web
    - download_ts: download timestamp
- signups: contains information about new user signups
    - user_id: primary id for a user
    - session_id: id of app download
    - signup_ts: signup timestamp
    - age_range: the age range the user belongs to
- ride_requests: contains information about rides
    - ride_id: primary id for a ride
    - user_id: foreign key to user (requester)
    - driver_id: foreign key to driver
    - request_ts: ride request timestamp
    - accept_ts: driver accept timestamp
    - pickup_location: pickup coordinates
    - destination_location: destination coordinates
    - pickup_ts: pickup timestamp
    - dropoff_ts: dropoff timestamp
    - cancel_ts: ride cancel timestamp (accept, pickup and dropoff timestamps may be null)
- transactions: contains information about financial transactions based on completed rides:
    - ride_id: foreign key to ride
    - purchase_amount_usd: purchase amount in USD
    - charge_status: approved, cancelled
    - transaction_ts: transaction timestamp

- reviews: contains information about driver reviews once rides are completed
    - review_id: primary id of review
    - ride_id: foreign key to ride
    - driver_id: foreign key to driver
    - user_id: foreign key to user (requester)
    - rating: rating from 0 to 5
    - free_response: text response given by user/requester

Data extraction

In order to conduct funnel analysis, data were extracted from the relational database system utilizing SQL queries. These queries were specifically designed to retrieve the necessary data for analysis. Furthermore, initial exploratory data analysis was performed using SQL queries, providing preliminary insights into the dataset. Further SQL analysis was done to build customer funnels. An additional SQL query was built to extract aggregated data to be used with other tools. Detailed information regarding those SQL queries can be found in the APPENDIX section of this document.

The extracted data was downloaded in CSV format to facilitate subsequent statistical analysis. Tableau was employed for data visualizations and for drawing various insights. This combination of tools allowed for a comprehensive examination of the dataset, enabling the generation of meaningful statistical insights.

By employing SQL queries for data extraction, conducting funnel analysis, and leveraging the capabilities of Tableau for data manipulation and visualization, a robust and systematic approach was adopted to analyze the dataset and derive valuable conclusions.

Customer Funnel Steps

Metrocar customer funnel includes the following stages:

1. **app_download**: A user downloads the Metrocar app from the App Store or Google Play Store.
2. **sign_up**: The user creates an account in the Metrocar app, including their name, email, phone number, and payment information.
3. **ride_requested**: The user opens the app and requests a ride by entering their pickup location, destination, and ride capacity (2 to 6 riders).
4. **ride_accepted**: A nearby driver receives the ride request and accepts the ride.
5. **ride_completed**: 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.

Metrics

‘Percent of Previous’ and ‘**Percent of Top’** were the two different approaches used to measure the **conversion** and **drop-off rates** at various funnel stages.

- **Percent of Previous**: This calculation involves measuring the conversion rate as a percentage of the users who proceeded to the current stage of the funnel, relative to the number of users at the previous stage. In other words, it tracks the progression of users through each stage of the funnel. This metric is useful when the objective is optimizing the user experience within the service and increasing conversion rates at various stages.

- **Percent of Top**: This calculation involves measuring the conversion rate as a percentage of the total number of users who entered the top of the funnel. The top of the funnel represents the initial stage where users enter or show interest in a particular process or journey. This metric is useful when the priority is to evaluate the effectiveness of marketing and user acquisition efforts.

Note: For Metrocar funnel analysis, since customer interaction with the service is sequential in nature and our objective is optimizing the user experience within the app and increasing conversion rates at various stages, we focused on ‘the **‘Percent of Previous’** metric by user_count. Additionally, we further enhanced that analysis with additional attributes e.g. Platform, Age_range, and download_dt. We also briefly explored the ‘**Percent of Top’** metric but for reference only.

Funnel Analysis

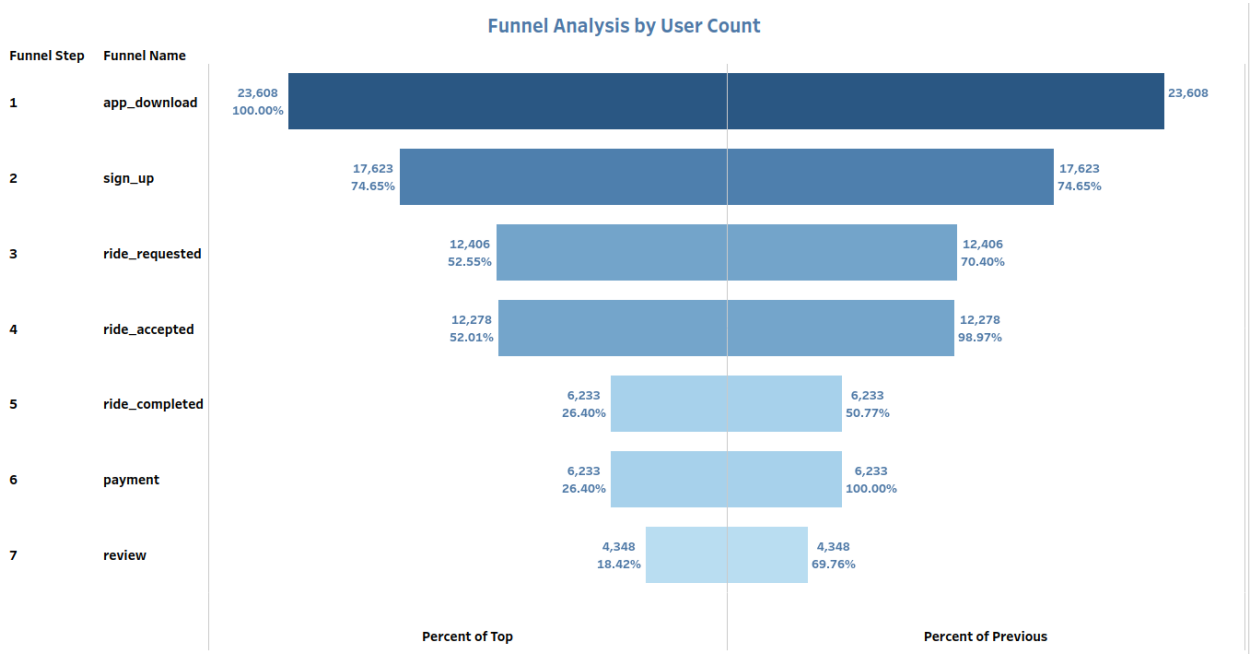
Metrocar customer funnel analysis was completed using the above metrics for **user_count** and **ride_count**.

A. By user_count (User Funnel)

Funnel Analysis by **user_count** using the **'Percent of Previous'** metric**:**

funnel_step	funnel_name	user_count	lag	diff	conversion_rate	dropoff_percent
1	app_download	23608				
2	sign_up	17623	23608	5985	0.7465	0.2535
3	ride_requested	12406	17623	5217	0.7040	0.2960
4	ride_accepted	12278	12406	128	0.9897	0.0103
5	ride_completed	6233	12278	6045	0.5077	0.4923
6	payment	6233	6233	0	1.0000	0.0000

| 7 | review | 4348 | 6233 | 1885 | 0.6976 | 0.3024 |



The above table tracks user counts at different steps of the customer journey using the 'Percent of Previous' metric. Here's a breakdown of the information in the table:

- **funnel_step**: This column represents the step in the customer journey.
- **funnel_name**: The name of the step in the customer journey.
- **user_count**: The number of users at each step of the funnel.
- **lag**: The difference between the current step's user count and the previous step's user count. It represents how many users progressed from the previous step to the current step.
- **diff**: The absolute difference between the current step's user count and the initial step (app_download) user count. It shows how many users were lost or gained at each step.
- **conversion_rate**: This is the conversion rate from one step to the next. It is calculated as the current step's user count divided by the previous step's user count. For example, for "sign_up," the conversion rate is 0.7465, indicating that 74.65% of users who downloaded the app signed up.
- **dropoff_percent**: This is the drop-off rate from one step to the next. It is calculated as 1 minus the conversion rate. For example, for "sign_up," the drop-off percent is 0.2535, indicating that 25.35% of users who downloaded the app did not sign up.

Key insights:

- The initial step is "app_download," with **23,608** users.
- The "sign_up" step has a conversion rate of **74.65**%, meaning that 74.65% of app downloaders sign up.

- The "ride_requested" step has a conversion rate of **70.40**% from "sign_up."
- "ride_accepted" has a high conversion rate of **98.97**% from "ride_requested," indicating a low drop-off rate.
- However, there's a significant drop in "ride_completed" with only **50.77**% of users who accepted a ride proceeding to complete it.
- The "payment" step retains all users from the previous step, indicating a **100**% conversion rate.
- "review" has a **69.76**% conversion rate from "payment," with a 30.24% drop-off rate.

Business questions:

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

- **Drop-off points where improvements are needed:**
    - "app_download" to "sign_up," with a drop-off of **25.35**%.
    - "ride_requested" to "ride_accepted," with a drop-off of **10.03**%.
    - "ride_accepted" to "ride_completed," with a significant drop-off of **49.23**%.
    - The part of the funnel with the **lowest conversion rate** is "ride_accepted" to "ride_completed" with a conversion rate of **50.77**%.

- **Recommendations:**
    - For the drop-off from "app_download" to "sign_up," consider **optimizing the sign-up process and possibly reducing friction** to encourage more users to sign up.
    - Perform **supply-side customer funnel analysis**: Currently, only demand-side customer data is available for analysis. If available, request access to supply-side data such as driver onboarding, app, activation, ride acceptance, ride completion, payment, tip, review, etc., and perform supply-side funnel analysis. That would provide additional insights to understand the reasons for customer drop-offs, quality of service, and potential revenue improvements.
        - Investigate why there's a drop-off between "ride_requested" and "ride_accepted" and work on **improving the user experience** in this transition.
        - Address the substantial drop-off between "ride_accepted" and "ride_completed" by understanding the reasons behind it. It could be related to **the quality of service, user experience, or other factors**.
            - Encourage customers to provide reviews/feedback for each ride
            - Perform Sentiment Analysis on review/feedback data using Machine learning/NLP techniques etc.

- Perform detailed analysis by combining results of sentiment analysis, demand-side funnel, and supply-side funnel.

Platform

Platforms used by customers: **android, ios, web**

Segment contribution:

platform	downloads	total_downloads	pct_of_downloads
ios	14290	23608	0.6053
web	2383	23608	0.1009
android	6935	23608	0.2938

Business question: Metrocar currently supports 3 different platforms (iOS, Android, and Web). What insights can we make based on the platform?

Key insights:

- Metrocar currently supports 3 different platforms (iOS, Android, and Web). iOS has a **60.53**% share followed by Android for **29.38**%.
- Applying platform level filter in Tableau funnel visualization shows **NO** significant difference in conversion and dropoff rates per platform.

Recommendations:

- **Equal Treatment Across Platforms**: Since there is no significant difference in conversion and dropoff rates per platform, it suggests that Metrocar's app is performing consistently across iOS, Android, and Web. This is a positive sign, as it indicates that users on all platforms are experiencing similar journeys. Therefore, Metrocar should continue to provide equal attention and support to all three platforms.
- **Invest in Cross-Platform Consistency**: Given that the user experiences across platforms appear similar, ensure that the app's features, user interface, and functionality are consistent across iOS, Android, and Web. Consistency can improve user satisfaction and trust in the app, regardless of the platform they are using.
- **Feature and Performance Enhancements**: Since there's no significant difference in conversion rates, consider allocating resources for feature enhancements and improving the overall performance of the app rather than focusing on a single platform. This approach can benefit users across all platforms.
- **Monitoring and Feedback**: Continue to monitor user behavior and feedback on each platform to identify any platform-specific issues or preferences. Regularly analyzing user data and feedback can help catch any emerging platform-specific trends.

- **Marketing Allocation**: When allocating a marketing budget, consider the relative share of users on each platform. While the conversion rates are similar, since there is a significant difference in the number of users between iOS and Android, allocate marketing resources accordingly i. e. more to iOS-based customers.
- **User Experience Testing**: Periodically conduct user experience testing on different platforms to ensure that any updates or changes do not negatively impact one platform over another. This will help maintain a consistent and high-quality user experience.

In summary, while iOS has the largest share of users, the funnel analysis suggests that Metrocar is performing consistently across platforms. The focus should be on ensuring a consistent and high-quality experience on all platforms, along with performance enhancements and monitoring of platform-specific trends to adapt as needed.

Age_Range

Age ranges of customers: **18-24, 25-34, 35-44, 45-54, Unknown**

Segment contribution:

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

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

Insights:

- **Age Distribution**: The age range with the highest number of signups is "**35-44**," accounting for **29.40**% of total signups, followed by "**25-34**" with **19.56**%. The "**Unknown**" category is also significant, representing **30.10**% of total signups, which likely includes users who did not provide their age.
- **Age Groups with Lower Signups**: The age ranges "18-24" and "45-54" have lower percentages of signups, accounting for 10.58% and 10.36%, respectively.
- **User Age Data**: The "Unknown" category, representing 30.10% of signups, might contain valuable data if more users could provide their age information.
- Applying the Age_Range filter in Tableau funnel visualization shows **NO** significant difference in conversion and dropoff rates per age range once the customer has signed up.

Recommendations:

- **Targeted Marketing and User Experience**:
    - Focus marketing efforts and user experience improvements on the age ranges "35-44" and "25-34" since they have the highest percentages of signups. Consider tailoring promotions or features that specifically appeal to these age groups.
- **Improving Signup Rates for Other Age Groups**:
    - To increase signups among the age ranges "18-24" and "45-54," consider targeted marketing campaigns that address the unique preferences and needs of these age groups. This might involve promotions, incentives, or features that resonate with these users.
- **Encourage Age Data Entry**:
    - Encourage users to provide their age information during the signup process. Offering incentives or explaining the benefits of sharing this data can help reduce the "Unknown" category and provide more accurate demographic insights.
- **User Segmentation and Personalization**:
    - Leverage the age data to segment users for personalized experiences. Tailor notifications, promotions, and user interfaces to better suit the preferences and behaviors of different age groups.
- **Data Analysis and Iteration**:
    - Continuously analyze user data, feedback, and conversion rates across different age groups. Use this information to improve the app's features and marketing strategies.
- **Retention Strategies**:
    - Once users are onboarded, focus on retention strategies tailored to different age groups. Understanding the preferences and needs of each age group can help in retaining and engaging users effectively.

In summary, the age-range data provides valuable insights into the distribution of signups among different age groups. Metrocar should use this information to target marketing efforts, improve the signup process, and provide personalized experiences to enhance user acquisition and retention across all age ranges.

download_dt

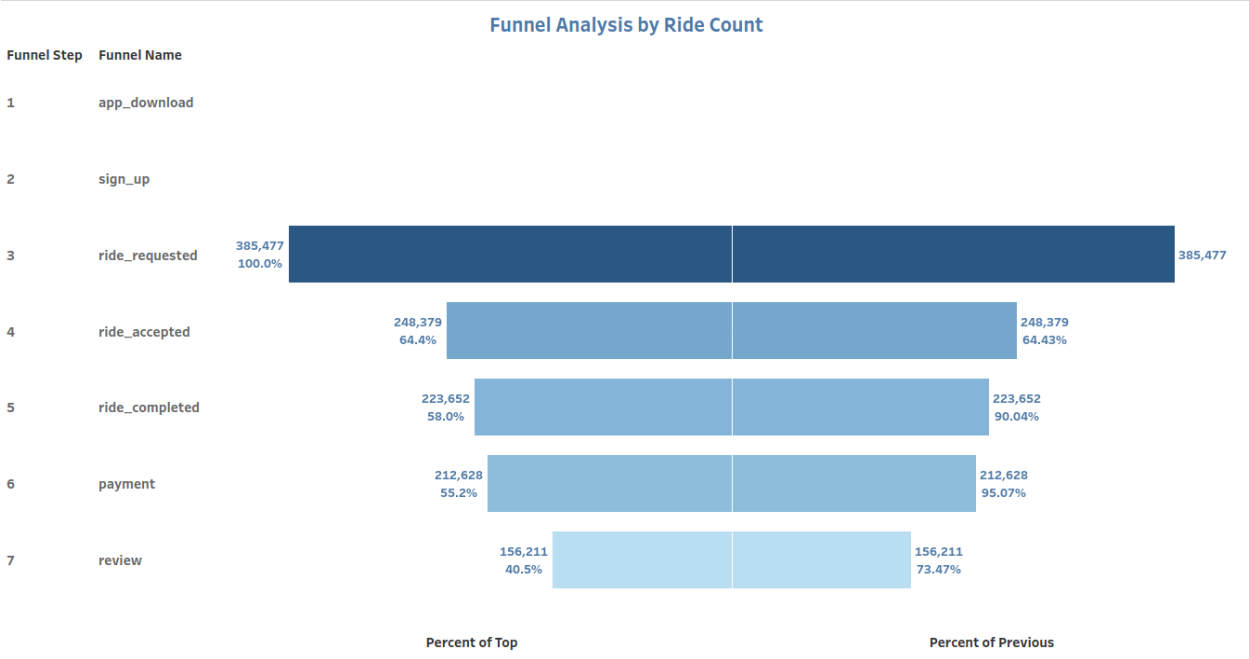
Business question: Surge pricing is the practice of increasing the price of goods or services when there is the greatest demand for them. If we want to adopt a price-surging strategy, what does the distribution of ride requests look like throughout the day?

- As the ride trip distribution visualization in Tableau shows, there are two activity peaks during the day. One between 8 AM and 9 AM the other one between 4 PM and 7 PM. These peaks could be due to working hours and are suitable for applying surge pricing.

Funnel Analysis by user_count using the 'Percent of Top' metric:

funnel_step	funnel_name	user_count	first_value	diff	conversion_rate	dropoff_percent
1	app_download	23608	23608	0	1.0000	0.0000
2	sign_up	17623	23608	5985	0.7465	0.2535
3	ride_requested	12406	23608	11202	0.5255	0.4745
4	ride_accepted	12278	23608	11330	0.5201	0.4799
5	ride_completed	6233	23608	17375	0.2640	0.7360
6	payment	6233	23608	17375	0.2640	0.7360
7	review	4348	23608	19260	0.1842	0.8158

B. By ride_count (Rides Funnel)



Funnel Analysis by **ride_count** using the **'Percent of Previous'** metric:

Note: The first 2 steps that are (not ride_count related) are excluded

| funnel_step | funnel_name | ride_count | lag | diff | conversion_rate | dropoff_percent |

| ---------- | ---------- | ---------- | ------ | ------ | ---------- | ---------- |
3	ride_requested	385477				
4	ride_accepted	248379	385477	137098	0.6443	0.3557
5	ride_completed	223652	248379	24727	0.9004	0.0996
6	payment	212628	223652	11024	0.9507	0.0493
7	review	156211	212628	56417	0.7347	0.2653

Key Insights:

- Ride Requested to Ride Accepted:
  - The 'ride_requested' step has 385,477 ride requests.
  - The 'ride_accepted' step has a conversion rate of **64.43**%, with 137,098 ride requests being accepted.
  - 35.57% of users dropped off at this step.
- Ride Accepted to Ride Completed:
  - The 'ride_accepted' step is followed by 'ride_completed' with 223,652 ride completions.
  - The conversion rate from 'ride_accepted' to 'ride_completed' is **90.04**%.
  - Approximately 9.96% of users dropped off at this stage.
- Ride Completed to Payment:
  - The 'ride_completed' step is followed by 'payment' with 212,628 payments.
  - The conversion rate from 'ride_completed' to 'payment' is **95.07**%.
  - Only 4.93% of users did not complete the payment process.
- Payment to Review:
  - The 'payment' step is followed by 'review' with 156,211 reviews.
  - The conversion rate from 'payment' to 'review' is **73.47**%.
  - Approximately 26.53% of users dropped off at this step.

Recommendations:

- There is a significant drop-off from 'ride_requested' to 'ride_accepted,' with **35.57**% of users not proceeding. Investigate the reasons for this drop and consider optimizing the ride acceptance process.
- Although the conversion rates from 'ride_accepted' to 'ride_completed' and 'payment' are good, optimizing these steps to reduce drop-offs further is advisable.
- The drop-off from 'payment' to 'review' is substantial (26.53%). It's important to analyze the payment and review processes to identify any issues that might be causing users to abandon the whole ride experience including feedback.

Funnel Analysis by **ride_count** using the **'Percent of Top'** metric:

Note: The first 2 steps that are (not ride_count related) are excluded

funnel_step	funnel_name	ride_count	first_value	diff	conversion_rate	dropoff_percent
3	ride_requested	385477	385477	0	1.0000	0.0000
4	ride_accepted	248379	385477	137098	0.6443	0.3557
5	ride_completed	223652	385477	161825	0.5802	0.4198
6	payment	212628	385477	172849	0.5516	0.4484
7	review	156211	385477	229266	0.4052	0.5948

3. Recommendations

Summarized recommendations to drive revenue growth and repeat business:

1. **Optimize Sign-Up Process**: Improve the "app_download" to "sign_up" transition to reduce the drop-off rate of 25.35%. Reduce sign-up friction.
2. **Improve Ride Acceptance**: Investigate and enhance the user experience from "ride_requested" to "ride_accepted" to reduce the 10.03% drop-off.
3. **Address "ride_accepted" to "ride_completed" Drop-Off**: Further analyze the reasons for the significant 49.23% drop-off rate and work on improving this transition.
    a. **Supply-side analysis**: Request access to supply-side data for insights into driver onboarding and other factors affecting customer drop-offs.
    b. **Encourage Reviews**: Prompt users to provide ride reviews and feedback, as it has a 69.76% conversion rate from "payment."
    c. **Sentiment Analysis**: Consider sentiment analysis using machine learning/NLP on review/feedback data to gain deeper insights and enhance customer satisfaction.
    d. **Combine Sentiment and Funnel Analysis**: Combine results of sentiment analysis, demand-side funnel, and supply-side funnel for comprehensive insights.
4. **Equal Treatment Across Platforms**: Maintain equal attention and support for iOS, Android, and Web due to similar conversion and drop-off rates.
5. **Marketing Allocation**: Allocate marketing resources according to the relative share of users on each platform, considering the significant difference in user numbers between iOS and Android.
6. **Targeted Marketing and User Experience**: Focus marketing efforts and user experience improvements on the "35-44" and "25-34" age groups, which have the highest sign-up percentages.
7. **User Segmentation and Personalization**: Utilize age data for segmenting users and offering personalized experiences. Encourage users to provide their age information during signup, offering incentives or explaining the benefits.
8. **Surge Pricing Strategy**: Implement surge pricing during peak hours between 8 AM - 9 AM and 4 PM - 7 PM for effective revenue management.

4. Conclusion

In conclusion, the funnel analysis provided valuable insights into Metrocar's customer journey. The analysis highlights opportunities for optimizing the sign-up process, improving ride acceptance, and addressing significant drop-offs in ride completion and review stages. Additionally, the analysis emphasizes the importance of equal treatment across platforms and surge pricing during peak hours. These recommendations aim to enhance user acquisition, retention, and overall customer experience.

5. APPENDIX

A. SQL code

```
/*
Funnel Analysis - MetroCar

postgres://Test:bQNxVzJL4g6u@ep-noisy-flower-846766-pooler.us-east-2.aws.neon.tech/Metrocar
*/

-- Metrocar customer funnel analysis by user_count using the 'Percent of Previous' metric

-- app_download : (DEFINES THE GROUP WE FOLLOW THROUGH THE FUNNEL)
WITH app_download AS (
SELECT
COUNT(DISTINCT app_download_key) AS total_users_app_downloaded
FROM app_downloads
),

-- sign_ups (FROM THE app_download ABOVE)
sign_ups AS (
SELECT
COUNT(DISTINCT user_id) total_users_signed_up
FROM signups
),

-- user_ride_status (FROM THE sign_ups ABOVE)
user_ride_status AS (
SELECT
user_id,
MAX(
CASE
WHEN accept_ts IS NOT NULL
THEN 1
ELSE 0
END
```

```sql
) AS ride_accepted,
MAX(
CASE
WHEN dropoff_ts IS NOT NULL
THEN 1
ELSE 0
END
) AS ride_completed
FROM ride_requests
GROUP BY user_id
),

-- payment_status
payment_status AS (
SELECT
r.user_id,
COUNT(*) AS total_rides_with_payment
FROM transactions AS t
LEFT JOIN ride_requests AS r
ON t.ride_id = r.ride_id
WHERE charge_status = 'Approved'
GROUP BY r.user_id
),

-- review_status
review_status AS (
SELECT
user_id,
COUNT(*) AS total_reviews_per_user
FROM reviews
GROUP BY user_id
),

-- steps
steps AS (
SELECT
1 AS funnel_step,
'app_download' AS funnel_name,
total_users_app_downloaded AS user_count
FROM app_download
UNION
SELECT
2 AS funnel_step,
'sign_up' AS funnel_name,
total_users_signed_up AS user_count
FROM sign_ups
UNION
SELECT
3 AS funnel_step,
'ride_requested' AS funnel_name,
COUNT(*) AS user_count --total_users_ride_requested
FROM user_ride_status
```

```sql
UNION
SELECT
4 AS funnel_step,
'ride_accepted' AS funnel_name,
SUM(ride_accepted) AS user_count --total_users_ride_accepted
FROM user_ride_status
UNION
SELECT
5 AS funnel_step,
'ride_completed' AS funnel_name,
SUM(ride_completed) AS user_count --total_users_ride_completed
FROM user_ride_status
UNION
SELECT
6 AS funnel_step,
'payment' AS funnel_name,
COUNT(*) AS user_count
FROM payment_status
UNION
SELECT
7 AS funnel_step,
'review' AS funnel_name,
COUNT(*) AS user_count
FROM review_status
)


SELECT
funnel_step,
funnel_name,
user_count,
lag(user_count, 1) OVER (ORDER BY funnel_step),
(lag(user_count, 1) OVER (ORDER BY funnel_step)) - user_count AS diff,
ROUND(user_count::numeric / lag(user_count, 1) OVER (ORDER BY funnel_step), 4) AS
conversion_rate,
ROUND((1.0 - user_count::numeric / lag(user_count, 1) OVER (ORDER BY funnel_step)), 4) AS
dropoff_percent
FROM steps
ORDER BY funnel_step ASC
;
/*
| funnel_step | funnel_name | user_count | lag | diff | conversion_rate | dropoff_percent |
| ----------- | ------------- | ---------- | ----- | ---- | --------------- | ----------------
|
| 1 | app_download | 23608 | | | | |
| 2 | sign_up | 17623 | 23608 | 5985 | 0.7465 | 0.2535 |
| 3 | ride_requested | 12406 | 17623 | 5217 | 0.7040 | 0.2960 |
| 4 | ride_accepted | 12278 | 12406 | 128 | 0.9897 | 0.0103 |
| 5 | ride_completed | 6233 | 12278 | 6045 | 0.5077 | 0.4923 |
| 6 | payment | 6233 | 6233 | 0 | 1.0000 | 0.0000 |
| 7 | review | 4348 | 6233 | 1885 | 0.6976 | 0.3024 |
*/
```

```
---------------------------------
---------------------------------

/*
Metrocar funnel data extraction for analysis in other tools e. g. Tableau
*/
WITH
app_download AS (
SELECT
COUNT(ad.app_download_key) AS total_users_app_downloaded,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM app_downloads AS ad
LEFT JOIN signups AS s ON ad.app_download_key = s.session_id
GROUP BY ad.platform, s.age_range, ad.download_ts::DATE
),
sign_ups AS (
SELECT
COUNT(s.user_id) AS total_users_signed_up,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM signups AS s
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
GROUP BY ad.platform, s.age_range, ad.download_ts::DATE
),
user_ride_status AS (
SELECT
rr.user_id,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt,
MAX(CASE WHEN rr.accept_ts IS NOT NULL THEN 1 ELSE 0 END) AS ride_accepted,
MAX(CASE WHEN rr.dropoff_ts IS NOT NULL THEN 1 ELSE 0 END) AS ride_completed
FROM ride_requests AS rr
LEFT JOIN signups AS s ON rr.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
GROUP BY rr.user_id, ad.platform, s.age_range, ad.download_ts::DATE
),
payment_status AS (
SELECT
rr.user_id,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt,
COUNT(*) AS total_rides_with_payment
FROM transactions AS t
LEFT JOIN ride_requests AS rr ON t.ride_id = rr.ride_id
LEFT JOIN signups AS s ON rr.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
```

```
WHERE charge_status = 'Approved'
GROUP BY rr.user_id, ad.platform, s.age_range, ad.download_ts::DATE
),
review_status AS (
SELECT
rv.user_id,
COUNT(*) AS total_reviews_per_user,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM reviews AS rv
LEFT JOIN signups AS s ON rv.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
GROUP BY rv.user_id, ad.platform, s.age_range, ad.download_ts::DATE
),
steps AS (
SELECT
1 AS funnel_step,
'app_download' AS funnel_name,
total_users_app_downloaded AS user_count,
platform,
age_range,
download_dt
FROM app_download
UNION
SELECT
2 AS funnel_step,
'sign_up' AS funnel_name,
total_users_signed_up AS user_count,
platform,
age_range,
download_dt
FROM sign_ups
UNION
SELECT
3 AS funnel_step,
'ride_requested' AS funnel_name,
COUNT(*) AS user_count,
platform,
age_range,
download_dt
FROM user_ride_status
GROUP BY platform, age_range, download_dt
UNION
SELECT
4 AS funnel_step,
'ride_accepted' AS funnel_name,
SUM(ride_accepted) AS user_count,
platform,
age_range,
download_dt
FROM user_ride_status
```

```sql
GROUP BY platform, age_range, download_dt
UNION
SELECT
5 AS funnel_step,
'ride_completed' AS funnel_name,
SUM(ride_completed) AS user_count,
platform,
age_range,
download_dt
FROM user_ride_status
GROUP BY platform, age_range, download_dt
UNION
SELECT
6 AS funnel_step,
'payment' AS funnel_name,
COUNT(*) AS user_count,
platform,
age_range,
download_dt
FROM payment_status
GROUP BY platform, age_range, download_dt
UNION
SELECT
7 AS funnel_step,
'review' AS funnel_name,
COUNT(*) AS user_count,
platform,
age_range,
download_dt
FROM review_status
GROUP BY platform, age_range, download_dt
),

requested_rides AS (
SELECT COUNT(*) AS total_rides_requested,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM ride_requests AS rr
LEFT JOIN signups AS s ON rr.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
GROUP BY ad.platform, s.age_range, ad.download_ts::DATE
),
accepted_rides AS (
SELECT COUNT(*) AS total_rides_accepted,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM ride_requests AS rr
LEFT JOIN signups AS s ON rr.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
WHERE rr.accept_ts IS NOT NULL
```

```
GROUP BY ad.platform, s.age_range, ad.download_ts::DATE
),
completed_rides AS (
SELECT COUNT(*) AS total_rides_completed,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM ride_requests AS rr
LEFT JOIN signups AS s ON rr.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
WHERE rr.dropoff_ts IS NOT NULL
GROUP BY ad.platform, s.age_range, ad.download_ts::DATE
),
payment_rides AS (
SELECT COUNT(*) AS total_rides_with_payment,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM transactions AS t
LEFT JOIN ride_requests AS rr ON t.ride_id = rr.ride_id
LEFT JOIN signups AS s ON rr.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
WHERE charge_status = 'Approved'
GROUP BY ad.platform, s.age_range, ad.download_ts::DATE
),
review_rides AS (
SELECT COUNT(*) AS total_rides_with_review,
ad.platform,
COALESCE(s.age_range, 'Not Specified') AS age_range,
ad.download_ts::DATE AS download_dt
FROM reviews AS rv
LEFT JOIN signups AS s ON rv.user_id = s.user_id
LEFT JOIN app_downloads AS ad ON s.session_id = ad.app_download_key
GROUP BY ad.platform, s.age_range, ad.download_ts::DATE
)

-- Main Query
SELECT
funnel_step,
funnel_name,
platform,
age_range,
download_dt,
user_count,
CASE
WHEN funnel_name = 'ride_requested' THEN (SELECT total_rides_requested FROM requested_rides
WHERE requested_rides.platform = steps.platform AND requested_rides.age_range =
steps.age_range AND requested_rides.download_dt = steps.download_dt)
WHEN funnel_name = 'ride_accepted' THEN (SELECT total_rides_accepted FROM accepted_rides WHERE
accepted_rides.platform = steps.platform AND accepted_rides.age_range = steps.age_range AND
accepted_rides.download_dt = steps.download_dt)
```

```
WHEN funnel_name = 'ride_completed' THEN (SELECT total_rides_completed FROM completed_rides
WHERE completed_rides.platform = steps.platform AND completed_rides.age_range =
steps.age_range AND completed_rides.download_dt = steps.download_dt)
WHEN funnel_name = 'payment' THEN (SELECT total_rides_with_payment FROM payment_rides WHERE
payment_rides.platform = steps.platform AND payment_rides.age_range = steps.age_range AND
payment_rides.download_dt = steps.download_dt)
WHEN funnel_name = 'review' THEN (SELECT total_rides_with_review FROM review_rides WHERE
review_rides.platform = steps.platform AND review_rides.age_range = steps.age_range AND
review_rides.download_dt = steps.download_dt)
ELSE NULL
END AS ride_count
FROM steps
ORDER BY funnel_step, platform, age_range, download_dt ASC
;

-- results in the aggregated dataset with 27886 rows
--------------------------------------
--------------------------------------
```

Note: Additional SQL code files are available in the links provided in the next section.

B. Links

1. **Dashboard on Tableau Public**: https://public.tableau.com/app/profile/r.h1008/viz/Metrocar-FunnelAnalysis_16978466864670/Metrocar-FunnelAnalysis

2. **Github**: https://github.com/ranga4all1/metrocar-funnel-analysis