METROCAR - FUNNEL ANALYSIS

Olalekan Alebiosu
November 2023

SUMMARY

This cutting-edge business model harnesses the power of technology for optimal leverage, typically through a smartphone app, to facilitate seamless and convenient ride sharing experience. Customers request rides, and track their drivers in real-time, pay electronically, while drivers have flexibility to work on their own schedules. This model has disrupted the traditional taxi industry and has become a prominent player in the transportation industry.

CONTEXT

The stakeholders have requested that the data analysis team conduct a thorough examination of Metrocar's data to pinpoint opportunities for enhancement and optimization. For instance, a key area of focus is analyzing the proportion of users who initiate an app download but fail to complete the registration process. Additionally, Metrocar is interested in understanding the percentage of users who place a ride request but then cancel it prior to the driver's arrival.

Funnel analysis is strategically implemented to accurately identify specific stages where users disengage or successfully convert, thereby significantly enhancing crucial outcomes such as increased sales, sign-ups, and conversion rates. This method is extensively leveraged in various domains, including marketing and product development, and is considered essential in propelling business growth and amplifying revenue.

Funnel analysis stands as a crucial technique in data analytics, employed to meticulously track and analyze the series of steps or stages customers navigate through while interacting with a product, service, or website. The term "funnel" aptly describes this analysis due to its resemblance to a real-world funnel – starting broad at the top and progressively narrowing down.

Objective:
Our primary aim is to foster business growth and enhance revenue by amplifying key outcomes such as bolstering sales, increasing sign-ups, and boosting overall conversions. Furthermore, this analysis will provide insights to address pertinent business questions related to these objectives.

Dataset Structure:
The comprehensive dataset for Metrocar, housed within a relational database system, comprises five detailed tables:

- app_download: This table chronicles app download details, including a unique identifier
  - (app_download_key) for each download,
  - the platform used (iOS, Android, or Web)
  - the timestamp of the download (download_ts).
- signups: Focused on new user sign-ups, this table includes crucial information like
  - the primary user ID (user_id),
  - the session ID linked to the app download (session_id),
  - the timestamp of the signup (signup_ts), and
  - the user's age range (age_range).
- 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 null
- transactions: contains information about financial transactions
  - ride id: foreign key to ride
  - purchase_amount_usd: purchase amount in USD
  - charge_status: approved, canceled
  - Transcsation_ts :transaction timestamps

- ❖ Reviews : contains information about driver review once the ride is 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
  - ➢ Rating :rating drivers for a completed ride

DATA EXTRACTION

To initiate the funnel analysis, data was meticulously extracted from the relational database system using tailored SQL queries. These queries were carefully crafted to gather the specific data needed for the analysis. An initial exploration of the dataset was conducted through SQL queries, offering early insights. Further SQL-based analysis was performed to construct customer funnels, and an additional query was developed to pull aggregated data for use in conjunction with other analytical tools. The details of these SQL queries are thoroughly documented in the APPENDIX section of this report.

The extracted data was then formatted into a CSV file to streamline the subsequent stages of statistical analysis. For the visualization and deeper exploration of the data, Tableau was utilized, enabling the extraction of diverse insights. This synergy of SQL for data extraction, combined with the advanced analytical and visual capabilities of Tableau, established a comprehensive and methodical framework for examining the dataset and uncovering significant statistical insights. This approach ensured a thorough and insightful analysis, leading to valuable and actionable conclusions.

Metrocar Customer Funnel Stages:

The Metrocar customer journey is structured into several distinct stages:

App Download: Users initiate their journey by downloading the Metrocar app from either the App Store or Google Play Store.
Sign-Up: Next, users register an account within the app, providing essential details such as their name, email, phone number, and payment information.
Ride Request: Users then request a ride by specifying their pickup location, destination.
Ride Acceptance: A nearby driver receives and accepts the ride request.

Ride Completion: The driver reaches the pickup location, the user boards the car, and is transported to their destination.
Payment: Post-ride, the payment is processed automatically through the app, and a receipt is emailed to the user.
Review: Finally, users are encouraged to rate their driver and share feedback on their ride experience.

Metrics for Funnel Analysis:

Two primary metrics were employed to assess the conversion and drop-off rates at each stage of the funnel:

Metrics for Funnel Analysis:

Percent of Previous: This metric calculates the conversion rate as a percentage of users who advanced to a given stage, relative to the number in the preceding stage. It is instrumental in monitoring user progression and is vital for optimizing user experience and enhancing conversion rates at different points of the funnel.
Percent of Top: This approach calculates the conversion rate based on the total number of users at the funnel's initial stage. It is particularly valuable for evaluating the effectiveness of marketing strategies and user acquisition efforts.

Conversion rate: the rate at which users move from one step to another
Drop-off rate: the rate at which users drop from one step to another without finishing.

Funnel Metrics Description:

funnel_step: Indicates the specific stage within the customer's journey.
funnel_name: Names each step in the customer journey for clarity.
user_count: Tracks the quantity of users present at each funnel stage.
lag: Reflects the user progression, calculated as the difference in user count between the current and previous funnel steps.
diff: Measures the net change in user count from the initial step (app download) to the current step, highlighting user retention or loss at each stage.
conversion_rate: Represents the proportion of users moving from one stage to the next, calculated by dividing the current step's user count by the previous step's count. For

instance, a 74.65% conversion rate at the "sign up" stage implies that 74.65% of those who downloaded the app proceeded to sign up.
dropoff_percent: Calculates the rate at which users exit the funnel between stages, computed as 1 minus the conversion rate. For example, a 25.35% drop-off at the "sign up" stage signifies that 25.35% of users who downloaded the app did not sign up.

Key Insights:

The journey begins with "app_download," attracting 23,608 users.
A notable drop-off of 25.35% is observed from "app_download" to "sign_up."
The "ride_requested" stage shows a 70.40% conversion from "sign_up."
"ride_accepted" demonstrates a high conversion of 98.97% from "ride_requested," indicating minimal drop-off.
However, "ride_completed" experiences a significant drop, with only 50.77% completing the ride after acceptance.
"payment" maintains a 100% conversion, suggesting all users who complete their ride proceed to payment.
The "review" stage sees a 69.76% conversion rate from "payment," with a 30.24% drop-off.

Recommendations:

Key Drop-Off Points for Improvement:
Transition from "app_download" to "sign_up" (25.35% drop-off).
Stage from "ride_requested" to "ride_accepted" (10.03% drop-off).
Crucial drop from "ride_accepted" to "ride_completed" (49.23% drop-off).
Strategies:
Optimizing Sign-Up: Enhance the sign-up process to decrease friction and boost user conversion from download to registration.
Supply-Side Analysis: Expand the analysis to include supply-side data (driver-related metrics) to gain comprehensive insights into service quality and customer drop-offs.
Enhancing Transition from Ride Request to Acceptance: Investigate and improve the user experience during this critical transition.
Addressing Ride Completion Drop-Offs: Explore the causes behind the substantial drop-off between ride acceptance and completion. This could stem from service quality, user experience, or other factors.

Encouraging Reviews and Feedback: Motivate customers to provide feedback after each ride.
Sentiment Analysis: provides a qualitative dimension to the quantitative data, offering a more comprehensive understanding of the customer journey. It helps in pinpointing emotional drivers and barriers within the funnel, guiding more effective strategies to enhance customer experience and conversion rates.
Integrated Analysis: Combine insights from sentiment analysis, demand-side, and supply-side funnel data for a holistic understanding and improvement strategy.

Enhanced Metrocar Funnel Analysis Insights and Recommendations:

Key Insights:

Metrocar operates across three platforms: iOS, Android, and Web. iOS leads with a 60.53% user share, followed by Android at 29.38%.
Tableau funnel analysis, incorporating a platform-level filter, reveals no significant variances in conversion and drop-off rates across these platforms.
Recommendations:

Uniform Platform Support: Given the consistent performance across all platforms, Metrocar should continue to equally prioritize and maintain iOS, Android, and Web platforms, ensuring a uniform user experience.
Cross-Platform Consistency: To ensure uniformity in user experience, Metrocar should maintain consistent features, user interfaces, and functionality across all platforms.
Feature and Performance Enhancement: Rather than focusing on a single platform, Metrocar should allocate resources towards enhancing features and overall app performance, benefiting users across all platforms.
Ongoing Monitoring and Feedback: Continuously monitor user behavior and feedback on each platform to identify and address any emerging platform-specific trends or issues.
Strategic Marketing Allocation: Allocate marketing budgets in proportion to the user base on each platform, with a more significant allocation towards iOS due to its larger user share.
Regular User Experience Testing: Conduct periodic user experience tests across different platforms to ensure that updates or changes maintain a consistent quality across all platforms.

Age Distribution Analysis:
Key Insights:

The predominant age group for sign-ups is "35-44," accounting for 29.40%, followed by "25-34" at 19.56%. The “Unknown” category is notably large, comprising 30.10% of total signups.
Lower signup rates are observed in the "18-24" and "45-54" age groups, representing 10.58% and 10.36%, respectively.

Recommendations:

Targeted Marketing for Dominant Age Groups: Focus marketing efforts and user experience enhancements on the "35-44" and "25-34" age groups.
Increasing Signup Rates in Other Age Groups: Develop targeted marketing initiatives tailored to the unique preferences of the "18-24" and "45-54" age groups.
Encouraging Age Data Entry: Motivate users to provide their age information during signup, possibly through incentives like getting discounted rides on their birthday, to minimize the "Unknown" category.
User Segmentation and Personalization: Utilize age data for user segmentation, offering personalized notifications and user interfaces.
Continual Data Analysis and Iteration: Regularly analyze user data, feedback, and conversion rates across age groups to refine app features and marketing strategies.
Focused Retention Strategies: Implement retention strategies tailored to the preferences of different age groups to maintain engagement.

Ride Trip Distribution and Conversion Rates:
Key Insights:

Peak activity periods are observed between 8 AM - 10AM and 4 PM - 10 PM, likely due to standard workday commutes.
There is a significant drop-off of 35.57% from "ride_requested" to "ride_accepted."
The conversion rate from "ride_accepted" to "ride_completed" is 90.04%, and from "ride_completed" to "payment" is 95.07%.
A notable 26.53% drop-off occurs from "payment" to "review."

Recommendations:

Optimize Ride Acceptance Process: Investigate and address the high drop-off rate from ride requests to acceptance.
Enhance Ride Completion and Payment Steps: Although conversion rates are high, further optimization of these stages could reduce drop-offs.
Review and Payment Process Analysis: Examine the payment and review stages to identify and resolve issues potentially deterring users from completing the feedback process.
Overall Strategy:
The analyses suggest Metrocar is achieving consistent performance across platforms and shows potential areas for improvement in user experience and platform-specific enhancements. The age-range data should be leveraged for targeted marketing and user experience strategies, focusing on retention and conversion optimizations at critical funnel stages.

Conclusion

In conclusion, the funnel analysis offered key insights into the customer journey at Metrocar, pinpointing areas for improvement. It underscored the need to refine the signup process, enhance ride acceptance rates, and tackle notable drop-offs at the ride completion and review phases. The analysis also stressed the necessity of maintaining consistency across all platforms and implementing dynamic pricing during peak hours. These strategies are geared towards boosting user acquisition, retention, and overall customer satisfaction and experience.

APPENDIX

SQL CODE:

```
--1 total ride requested
SELECT COUNT(*) AS total_rides
FROM ride_requests;
```

```sql
--1 How many times was the app downloaded?
SELECT COUNT(DISTINCT app_download_key)
FROM app_downloads;

-- 2 How many users signed up on the app?
SELECT COUNT(DISTINCT user_id)
FROM signups;

-- 3 How many rides were requested through the app?
SELECT COUNT(DISTINCT ride_id) AS Total_ride_request
FROM ride_requests;

-- 4 How many rides were requested and completed through the app?
SELECT
   COUNT(DISTINCT ride_id) AS Total_ride_request ,
   COUNT(
      CASE
         WHEN dropoff_ts IS NOT NULL AND accept_ts IS NOT NULL AND
pickup_ts IS NOT NULL
         THEN ride_id
      END
   ) AS completed_rides
FROM ride_requests;

select count(distinct ride_id)
from ride_requests
where cancel_ts is null;


SELECT COUNT(DISTINCT ride_id)
FROM transactions
WHERE charge_status = 'Approved';
```

```sql
-- 5 How many rides were requested and how many unique users requested a ride?
SELECT
   COUNT(DISTINCT user_id) AS Total_user_request,
   COUNT(DISTINCT ride_id) AS total_rides
FROM ride_requests;

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

SELECT *,dropoff_ts - pickup_ts AS time_taken
FROM ride_requests;

SELECT AVG(dropoff_ts - pickup_ts) AS avg_time_taken
FROM ride_requests;

-- 7 How many rides were accepted by a driver?
SELECT COUNT(accept_ts) AS accepted_rides
FROM ride_requests;


SELECT COUNT (DISTINCT ride_id)
FROM ride_requests
WHERE accept_ts IS NOT NULL;


-- 8 How many rides did we successfully collect payments and how much was collected
SELECT
   SUM(purchase_amount_usd) AS total_amount,
   COUNT(DISTINCT transaction_id) AS total_transaction
FROM transactions
WHERE charge_status = 'Approved' AND purchase_amount_usd IS NOT NULL;


-- 9 How many ride requests happened on each platform?
SELECT platform, COUNT(*) AS total_platform_rides
FROM app_downloads
GROUP BY platform;
```

```
SELECT a.platform,b.user_id
FROM app_downloads AS a
INNER JOIN signups AS b
ON a.app_download_key = b.session_id;

SELECT p.*,c.ride_id
FROM
(SELECT a.platform,b.user_id
FROM app_downloads AS a
INNER JOIN signups AS b
ON a.app_download_key = b.session_id) AS P
INNER JOIN ride_requests AS C
ON P.user_id = C.user_id;



SELECT platform, COUNT(DISTINCT ride_id)
FROM
(SELECT p.*,c.ride_id
FROM
(SELECT a.platform,b.user_id
FROM app_downloads AS a
INNER JOIN signups AS b
ON a.app_download_key = b.session_id) AS P
INNER JOIN ride_requests AS C
ON P.user_id = C.user_id) AS T
GROUP BY platform;


--10 What is the drop-off from users signing up to users requesting a ride?
WITH user_ride_status  AS (
  SELECT user_id
  FROM ride_requests
  GROUP BY user_id
)
SELECT
```

```
COUNT (*) AS total_signup,
COUNT(DISTINCT user_ride_status.user_id) AS total_user_ride
FROM signups s
FULL JOIN user_ride_status
ON s.user_id = user_ride_status.user_id;


--What is the drop-off from users signing up to users requesting a ride?
SELECT COUNT(DISTINCT a.user_id),COUNT(DISTINCT b.user_id)
FROM ride_requests AS a
RIGHT JOIN signups AS b
ON a.user_id = b.user_id;

TABLEAU EXTRACT
WITH
-- user related CTE's
  app_download AS (
    SELECT
      COUNT(DISTINCT 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(DISTINCT 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 related CTE's
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
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
),
```

```
-- ride related CTE's
  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
;
```

LINKS

Dashboard Tableau Public
:https://public.tableau.com/views/MetrocarRideshareCompany/MetrocarAnalysis?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Tableau links for Visualization

Funnel Analysis by user count
https://public.tableau.com/views/MetrocarRideshareCompany/FunnelAnalysisbyusercount?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Funnel Analysis by ride count
https://public.tableau.com/views/MetrocarRideshareCompany/FunnelAnalysisbyridecount?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

App Download by Quarter

https://public.tableau.com/views/MetrocarRideshareCompany/AppDownloadbyQuarter?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Age Range of User
https://public.tableau.com/views/MetrocarRideshareCompany/AgeRangeofUser?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Surge Price Analysis
https://public.tableau.com/views/MetrocarRideshareCompany/SurgePriceAnalysis?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

Video Presentation Link:
https://www.loom.com/share/8d6259c658f54211b79611921403e149?sid=e3430c1f-7fda-489f-99e6-de6aa20b38f3