Metro Car Funnel Analysis

By Nicholas Giddings

To get a feel of the data, I first did an initial analysis to answer some basic questions in SQL.

How many times was the app downloaded?

SELECT COUNT(*)
FROM app_downloads
	23,608 Downloads

How many users signed up on the app?

SELECT COUNT(*)
FROM signups
	17,623 Signups

How many rides were requested through the app?

SELECT COUNT(*)
FROM ride_requests
	385,477 Total Ride Requests

How many rides were requested and completed through the app?

SELECT COUNT(*) AS total_rides
FROM ride_requests
WHERE pickup_ts IS NOT NULL AND dropoff_ts IS NOT NULL
	223,652 Total Completed Rides

How many rides were requested and how many unique users requested a ride?

SELECT COUNT(*) AS total_rides,
COUNT (DISTINCT user_id) AS unique_user_req
FROM ride_requests
	385,477 Rides Requested
	12,406 Unique User Ride Requests

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

SELECT ROUND(AVG(strftime('%s', dropoff_ts) - strftime('%s', pickup_ts)) / 60, 2) AS avg_time_minutes
FROM ride_requests
WHERE pickup_ts IS NOT NULL AND dropoff_ts IS NOT NULL
	Average length of ride: 52.61 Minutes

How many rides were accepted by a driver?

SELECT COUNT(accept_ts) AS total_accepted
FROM ride_requests
WHERE accept_ts IS NOT NULL
	248,379 Accepted Rides

How many rides did we successfully collect payments and how much was collected?

SELECT COUNT(*) AS total_successful,
ROUND(SUM(purchase_amount_usd), 2) AS total_revenue
FROM transactions
WHERE charge_status = 'Approved'
	212,628 Total Successful Payments
	4,251,667.61 Total Revenue

How many ride requests happened on each platform?

SELECT SUM(CASE WHEN platform = 'android' THEN 1 ELSE 0 END) AS android_requests, SUM(CASE WHEN platform = 'ios' THEN 1 ELSE 0 END) AS ios_requests, SUM(CASE WHEN platform = 'web' THEN 1 ELSE 0 END) AS web_requests FROM app_downloads JOIN signups ON app_download_key = session_id JOIN ride_requests USING (user_id)	112,317 Android 23,4693 Ios 38,467 Web

What is the drop-off from users signing up to users requesting a ride?

SELECT COUNT(DISTINCT user_id) AS total_signups, COUNT(DISTINCT CASE WHEN rr.user_id IS NULL THEN s.user_id END) AS dropped_users, ROUND((COUNT(DISTINCT CASE WHEN rr.user_id IS NULL THEN s.user_id END) * 1.0 / COUNT(DISTINCT user_id)), 2) * 100 AS drop_rate FROM signups s LEFT JOIN ride_requests rr USING (user_id)	17,623 signups 5,217 Distinct User Ride Requests 30.0 % Drop Off Rate

---

To create visualisations for the presentation, I wanted to use Tableau. Due to the size and complexity of the data, it was beneficial to aggregate the data. I made two aggregations. First aggregating by platform and age range. I used resulting aggregated data to create these visualisations.

You will note that I created two funnels. The first shows a user level analysis that does not include unique users that were accepted for a ride, completed a ride and paid for a ride. This is because that information would create a misleading representation of the data as it does not count users who used the app more than once to request a ride. That is where the second funnel comes in, that uses a count of total rides.

The second aggregated dataset I created was to aggregate the data by day of week and hour of day, so that I could create this visualisation.

Below you can find the 2 queries I used to create these aggregates.

Aggregated by platform and age

```sql
WITH user_details as
(SELECT app_download_key, user_id, platform, age_range
FROM app_downloads
LEFT JOIN signups
ON app_downloads.app_download_key = signups.session_id
WHERE user_id IS NOT NULL),

downloads as
(SELECT 0 as step , 'downloads' as name, user_details.platform, user_details.age_range,
COUNT(distinct app_downloads.app_download_key) as total_distinct_users,
0 as total_rides
FROM app_downloads
LEFT JOIN signups ON app_downloads.app_download_key = signups.session_id
LEFT JOIN user_details USING (user_id)
GROUP BY user_details.platform, user_details.age_range),

signup as
(SELECT 1, 'signups', user_details.platform,  user_details.age_range,
COUNT(distinct user_id) as users_signups,0
FROM signups JOIN user_details USING (user_id)
WHERE signup_ts IS NOT NULL
GROUP BY user_details.platform, user_details.age_range),

requested as
(SELECT 2, 'ride_requested', user_details.platform,  user_details.age_range,
COUNT(distinct user_id) as users_requested,
COUNT(distinct ride_id) as rides_requested
FROM ride_requests
JOIN user_details USING (user_id)
WHERE request_ts IS NOT NULL
GROUP BY user_details.platform, user_details.age_range),

accepted as
(SELECT 3, 'ride_accepted', user_details.platform, user_details.age_range,
COUNT(distinct user_id) as users_accpeted,
COUNT(distinct ride_id) as rides_accepted
FROM ride_requests
JOIN user_details USING (user_id)
WHERE accept_ts IS NOT NULL
GROUP BY user_details.platform, user_details.age_range),

completed as
(SELECT 4, 'ride_completed', user_details.platform, user_details.age_range,
COUNT(distinct user_id),
COUNT(distinct ride_id) as rides_completed
FROM ride_requests
JOIN user_details USING (user_id)
WHERE dropoff_ts IS NOT NULL
GROUP BY user_details.platform, user_details.age_range),
payment as
(SELECT 5, 'payment', user_details.platform, user_details.age_range,
COUNT(distinct user_id) as user_count,
COUNT(ride_id) as count_rides
FROM transactions
JOIN ride_requests USING (ride_id)
JOIN user_details USING (user_id)
WHERE charge_status = 'Approved'
GROUP BY user_details.platform, user_details.age_range),

review as
(SELECT 6, 'reviews', user_details.platform, user_details.age_range,
COUNT(distinct user_id),
COUNT(distinct ride_id) as count_rides
FROM reviews
JOIN user_details USING (user_id)
GROUP BY user_details.platform, user_details.age_range)

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 review
ORDER BY step
```

Aggregate by time and day

```
SELECT
CASE CAST (
STRFTIME('%w', request_ts) as integer)
WHEN 0 THEN 'Sunday'
WHEN 1 THEN 'Monday'
WHEN 2 THEN 'Tuesday'
WHEN 3 THEN 'Wednesday'
WHEN 4 THEN 'Thursday'
WHEN 5 THEN 'Friday'
else 'Saturday' END as request_day,
STRFTIME('%H', request_ts) as request_hour,
COUNT(distinct ride_id) as total_rides
FROM ride_requests
WHERE request_ts IS NOT NULL AND
        accept_ts IS NOT NULL
GROUP BY 1, 2
ORDER BY 3 DESC
```