Cyclistic bike-share case study

Terence Chau

18/6/2021

Introduction

This is a case study for Cyclistic, a fictional bike-share company located at Chicago which have more than 5800 bicycles and 600 docking stations. The task is to anaylze the bike riding behaviour between member and casual rider and advice measures to convert casual rider to membership as management believes the company's future success depends on maximizing the number of annual membership.

Data overview

12 months of latest available data in csv format was used and the data has been made available by Motivate International Inc. under this license, and bike rental data from May-2020 Apr-2021 was used for the task. As data was generated directly from the bike sharing system and released by the company, the data is reliable, original, comprehensive and current and therefore suitable to use for the analysis. It includes ride id, bike type, start and end riding time, start and end station name and station ID, start and end station geolocation, and membership or casual rider information.

Column	Description
ride_id	unique identifier (primary key)
rideable_type	bike type
started_at	start riding time
ended_at	end riding time
start_station_name	bike rental station name
start_station_id	bike rental station id
end_station_name	bike return station name
end_station_id	bike return station id
start_lat	bike rental latitude
start_lng	bike rental longitude
end_lat	bike return latitude
end_lng	bike return longitude
member_casual	whether user is member or casual rider

12 csv files were imported into PostgreSQL for further data cleaning and analysis, and at a later stage transferred to Tableau for data visualization.

For the following section, only key scripts are shown, please refer to the script file for more detail.

Data cleaning and processing

As the data for the 12 months are separated in 12 csv file while in the same format, they were merged to be a single table.

However there are duplicated entry in the file for December, it was first imported as a separate table named temp_dec, then delete those duplicated entry and then move to the main table named biketrip.

delete duplicate entry

```
DELETE FROM temp_dec
USING biketrip
WHERE biketrip.ride_id = temp_dec.ride_id;
```

move to main table

```
INSERT INTO biketrip
SELECT * FROM temp_dec;
```

When all 12 csv files were imported into PostgreSQL, some entries with negative riding time were found, i.e. start riding time later than end riding time. Those record were removed.

remove negative riding time

```
WITH remove_neg AS (
    SELECT
      ride_id,
      started_at,
      ended_at,
      EXTRACT(EPOCH FROM (ended_at - started_at)) AS trip_time
    FROM biketrip
    GROUP BY 1, 2, 3
    HAVING EXTRACT(EPOCH FROM (ended_at - started_at)) < '0'
    )
DELETE FROM biketrip
USING remove_neg
WHERE biketrip.ride_id = remove_neg.ride_id;
```

The data is now ready to be analyze.

Analyze and findings

First, let's see the number of member and causal rider in 12 months time.

```
SELECT member_casual, COUNT(ride_id)
FROM biketrip
GROUP BY member_casual
ORDER BY member_casual DESC;
```

Table 2: 2 records

member_casual	count
member	2191584
casual	1540112

Figure 1: **number of ride between 2020-5 to 2021-4**

Member takes 59% of bike rental while casual rider takes the remaining 41%.

Next we check if casual rider and member have their preferred bike type.

```
SELECT rideable_type, member_casual, COUNT(ride_id) AS num_of_ride
FROM biketrip
GROUP BY rideable_type, member_casual
ORDER BY num_of_ride DESC;
```

Table 3: 6 records

rideable_type	member_casual	num_of_ride
docked_bike	member	1373605
docked_bike	casual	1114512
electric_bike	member	425068
classic_bike	member	392911
electric_bike	casual	284024
classic_bike	casual	141576

Both casual rider and member preferred docked bike much more than the other types ot bike, while classic bike is generally the least preferred by casual rider.

Then we would like to see the trend of ride count in 12 months time

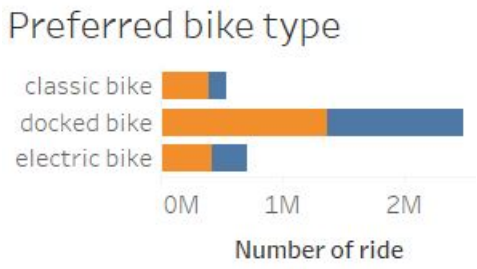

Figure 2: **Preferred bike type**

```sql
SELECT
  CONCAT(EXTRACT(YEAR FROM started_at), '-', EXTRACT(MONTH FROM started_at)) AS month,
  member_casual,
  COUNT(EXTRACT(MONTH FROM started_at)) AS ride_per_month
FROM biketrip
GROUP BY month, member_casual
ORDER BY month;
```

Table 4: Displaying records 1 - 10

month	member_casual	ride_per_month
2020-10	casual	144529
2020-10	member	242213
2020-11	casual	87911
2020-11	member	170940
2020-12	casual	29997
2020-12	member	101142
2020-5	casual	86844
2020-5	member	113258
2020-6	casual	154551
2020-6	member	187985

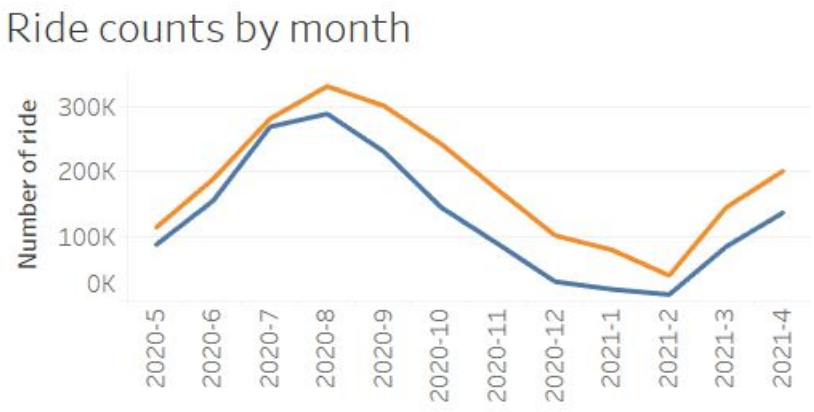

Figure 3: **Number of ride per month**

Here we can see ride numbers are at the peak on August and lowest on February, and the gap of member and casual rider number widen after August.

Bike rental time also behave differently between the two groups.

```sql
WITH long_rent AS (
    SELECT
```

```
    ORDER BY trip_time DESC
)
SELECT member_casual, rideable_type, COUNT(member_casual)
FROM long_rent
WHERE trip_time > '86400' -- 24 hours
GROUP BY member_casual, rideable_type;
```

Table 5: 4 records

member_casual	rideable_type	count
member	docked_bike	232
casual	docked_bike	2409
member	classic_bike	113
casual	classic_bike	260

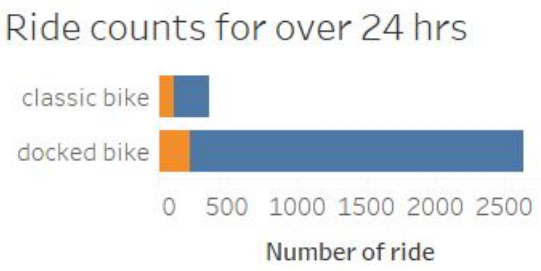

Figure 4: **Rental time for more than 24 hours**

There are more than double of long time classic bike rental for casual riders then members, and more than 10 times of long time classic bike rental for casual riders then members.

Let's also check the average riding time

```
WITH instant AS (
    SELECT
      member_casual,
      EXTRACT(EPOCH FROM (ended_at - started_at)) AS trip_time
    FROM biketrip
    )
SELECT
  member_casual,
  AVG(trip_time)/60 AS average_trip_minutes
FROM instant
WHERE trip_time >= '60'
GROUP BY member_casual;
```

Table 6: 2 records

member_casual	average_trip_minutes
casual	44.45016

member_casual	average_trip_minutes
member	16.08702

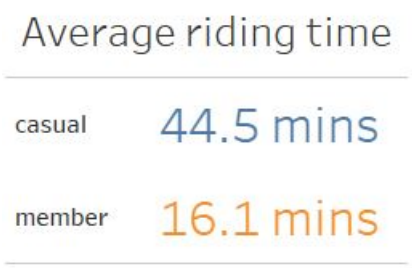

Figure 5: **Average riding time**

Average riding time for casual riders is almost triple of that for members.

Biking behaviour between the 2 groups can also be understand by checking how they use the bikes. Here we check for popular route of popularity over 500.

```
WITH popular_route AS (
    SELECT
      rideable_type,
      member_casual,
      CONCAT(start_station_id, ' to ', end_station_id) AS route
    FROM biketrip
    WHERE
      start_station_id IS NOT NULL AND
      end_station_id IS NOT NULL
)
SELECT rideable_type, member_casual, route, COUNT(route) AS route_count
FROM popular_route
GROUP BY rideable_type, member_casual, route
HAVING COUNT(route) >= '500'
ORDER BY route_count DESC
```

Table 7: Displaying records 1 - 10

rideable_type	member_casual	route	route_count
docked_bike	casual	35 to 35	5352
docked_bike	casual	76 to 76	4836
docked_bike	casual	2 to 2	4428
docked_bike	casual	90 to 90	4274
docked_bike	casual	255 to 255	3346
docked_bike	casual	85 to 85	3064
docked_bike	casual	623 to 623	2979
docked_bike	casual	150 to 150	2920
electric_bike	casual	676 to 676	2809
docked_bike	casual	247 to 247	2693

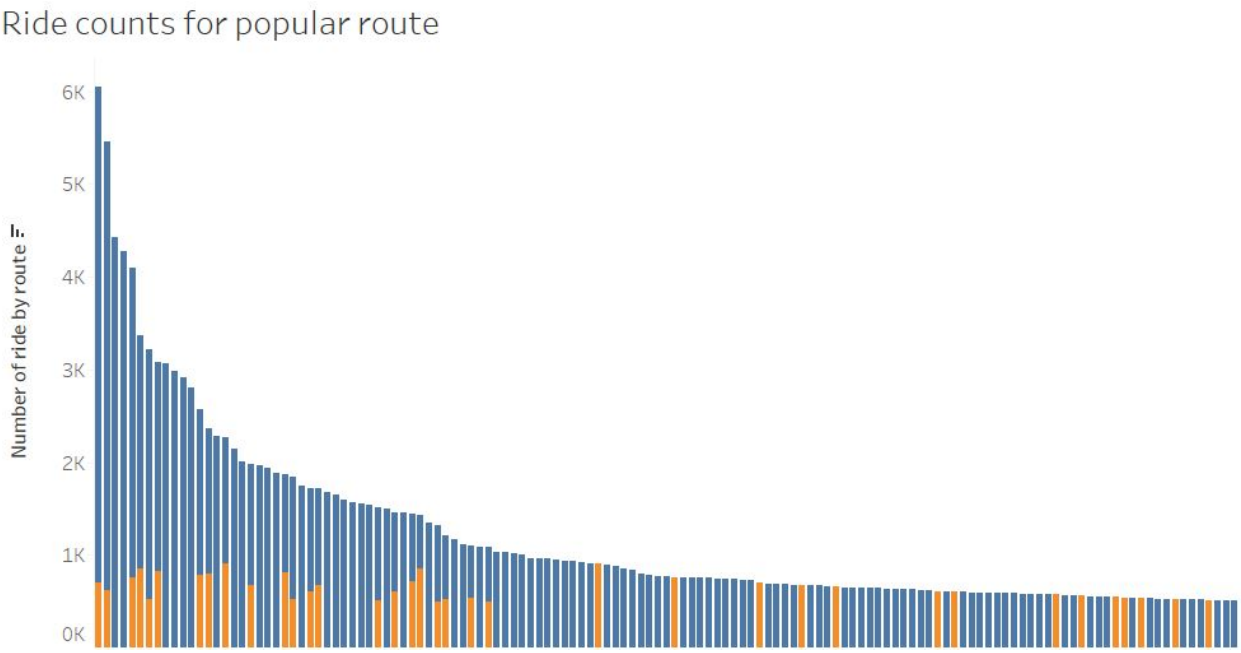

Figure 6: **Ride counts for popular route**

The most popular routes are dominated by casual rider.

Let's see how users use bike throughout the weeks

```sql
WITH dateofweek AS(
  SELECT member_casual, EXTRACT(DOW FROM started_at) AS dow
  FROM biketrip
)
SELECT member_casual,
  CASE
    WHEN dow = 0 THEN 'Sunday'
    WHEN dow = 1 THEN 'Monday'
    WHEN dow = 2 THEN 'Tuesday'
    WHEN dow = 3 THEN 'Wednesday'
    WHEN dow = 4 THEN 'Thursday'
    WHEN dow = 5 THEN 'Friday'
    WHEN dow = 6 THEN 'Saturday'
    ELSE 'others'
  END AS date_of_week,
  COUNT(dow)
FROM dateofweek
GROUP BY member_casual, dow
ORDER BY COUNT(dow) DESC;
```

Table 8: Displaying records 1 - 10

member_casual	date_of_week	count
casual	Saturday	358721
member	Saturday	340937
member	Friday	334732
member	Wednesday	323538
member	Thursday	319275
member	Tuesday	307072
member	Monday	287008
casual	Sunday	280994
member	Sunday	279022
casual	Friday	228972

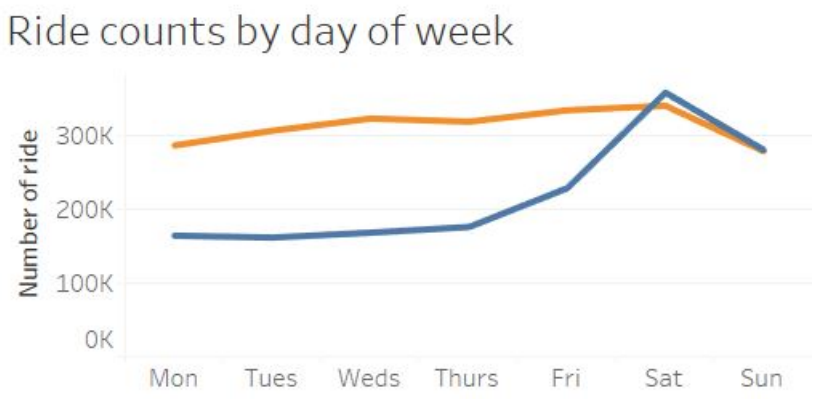

Figure 7: **Bike usage throughout the weeks**

Members have significant higher bike usage on weekdays but drops on Sunday, while casual rider's number peak on weekends.

Check also users biking time

```
WITH hours AS(
  SELECT member_casual, EXTRACT(HOUR FROM started_at) AS bike_hour
  FROM biketrip
)
SELECT member_casual, bike_hour, COUNT(bike_hour)
FROM hours
GROUP BY member_casual, bike_hour
ORDER BY COUNT(bike_hour) DESC;
```

Table 9: Displaying records 1 - 10

member_casual	bike_hour	count
member	17	231538
member	18	202167
member	16	191097

member_casual	bike_hour	count
member	15	159015
casual	17	152494
member	12	143804
member	14	142910
member	13	142755
member	19	140667
casual	18	137718

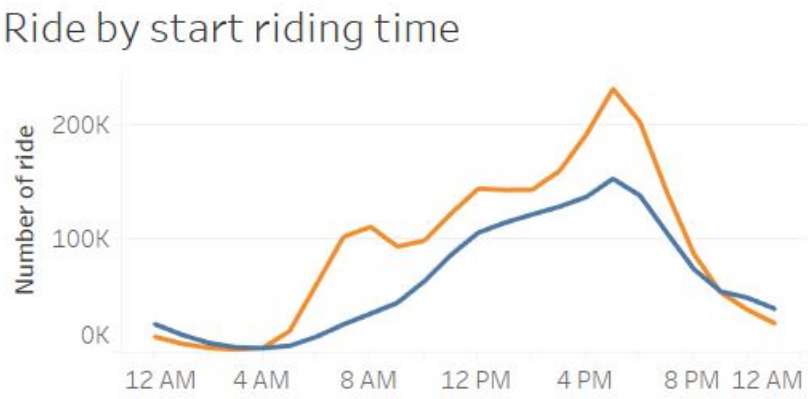

Figure 8: **Bike usage throughout the days**

The number of rides for both casual rider and member gradually increase from the morning until 6pm and then the number of ride drops till the end of the day, while ride number for member is even higher then casual rider at morning and evening rush hour. Also number of ride for casual rider is higher than member in the evening and mid-night.

Last, let's check their average riding distance

```
WITH bd AS (
SELECT member_casual, (point(start_lng, start_lat)<@>point(end_lng, end_lat)) * 1.609344 AS distance
FROM biketrip
WHERE
  start_lng IS NOT NULL AND
  start_lat IS NOT NULL AND
  end_lng IS NOT NULL AND
  end_lat IS NOT NULL AND
  (point(start_lng, start_lat)<@>point(end_lng, end_lat)) * 1.609344 <> '0'  -- convert miles to KM
)
SELECT member_casual, AVG(distance) AS avg_ride_km
FROM bd
GROUP BY member_casual;
```

Table 10: 2 records

member_casual	avg_ride_km
casual	2.543380
member	2.366714

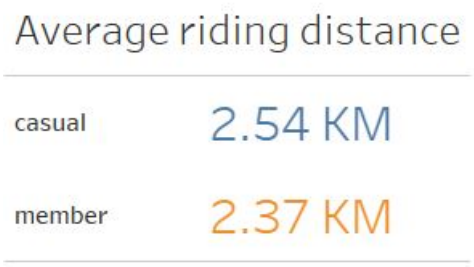

Figure 9: **Average riding distanve**

Although casual rider have a slightly longer average riding distance then member, their average distance are similar.

Insight Summary

From the anaylsis above, we have understand some key facts and know some behavioral difference between member and casual rider.

- Our current user ratio of member to casual rider is around 6:4.
- Majority of user for both group prefer docked bike over the others, while classic bike are least preferred by casual rider.
- Ride numbers are higher in summer and lower in winter, with member more willing to ride in autumn and winter than casual rider.
- There are significant more casual rider rent a docked bike for over 24 hours. Perhaps they take the bike to vacation, or they simply ride back home where no docking station nearby?
- Average riding time for casual riders is almost triple of that for members.
- Since members account for around 60% of bike rental yet the most popular route are dominated by casual rider, casual rider tends to ride between popular route, while member's riding route are more spread around.
- Members have significant higher bike usage on weekdays while casual rider rides more on weekends, most likely member mainly bike for commuting while casual rider bike for leisure.
- Data on riding hours further support that member's usage for commuting.

A Visualized dashboard was made and can be view at the following link:

https://public.tableau.com/app/profile/terence.chau/viz/Cyclisticbike-sharecasestudy/Dashboard1

Recommendation

To maximize the number of membership riders, we can target to convert the current casual rider to membership rider as they are already aware us and have tried our service.

We can set up a type of membership that tailor made for the current casual rider, i.e. optimize for leisure usage, for example some type of weekend pass.

We can also introduce membership promotion during winter time to boost the low riding number in winter.