Google Data Analytics Case Study

Cyclistic Bike-Share

With

Jupiter-Notebook & POWERBI

CONTENTS

Cyclistic
BIKE-SHARE

00

INTRODUCTION

Company Summary

Cyclistic is a bike-share company located in Chicago. It has a bike share program that features more than 5,800 bicycles and 600 docking stations.

Its provides reclining bikes, hand tricycles, and cargo bikes, making bike-share more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike.

01

ASK

Ask Phase

- **Problem Statement**

Ensure Key growth of Cyclistic by increasing the number of annual members

- **Business Task**

The task here is to show how annual members and casual riders use cyclistic bikes differently.

- **Stakeholders**

The main share holders are the **director of Marketing(Lili moreno)**, **Cyclistic Marketing** and **Cyclistic Analytic** Team

02

PREPARE

Prepare Phase

- **Information on Data Source**

The data is publicly available on Index of bucket "divvy-tripdata“
The data is stored in 12 csv files
The data range from 2021-09 to 2022-08
The data is representative

- **Is Data ROCCC**

➢ Reliable : it has multiple data and for long period
➢ Original: It is third party data
➢ Comprehensive : The data has clear components and informations
➢ Current: The data is quite old(last 1 year data)
➢ Cited : The data source is unknown.

03

PROCESS

Process Phase

❑ Importing library

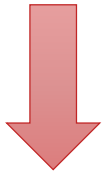
```
#import library

import pandas as pd
from zipfile import ZipFile,Path
import glob
import fnmatch
from io import BytesIO, StringIO
import numpy as np
import datetime
```

❑ Combining all csv files into one

```
#df_master : it is the combined data set

path = r'C:\Users\G84183771\Downloads\Learn\tripdata\*'
#load all zip files in folder
all_files = glob.glob(path)


df_master = pd.DataFrame()
flag = False

for filename in all_files:
    zip_file = ZipFile(filename)
    files = zip_file.namelist()
    with zip_file.open(files[0]) as csvfile:
        df=pd.read_csv(csvfile, encoding='utf8', sep=",")
        df_master=pd.concat([df_master, df])
```

❑ Dataset Columns info

```
''' Having columns of the dataset'''
df_master.columns
```

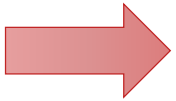


```
Index(['ride_id', 'rideable_type', 'started_at', 'ended_at',
       'start_station_name', 'start_station_id', 'end_station_name',
       'end_station_id', 'start_lat', 'start_lng', 'end_lat', 'end_lng',
       'member_casual'],
      dtype='object')
```

Process Phase

❑ Changing Column data type

```
'''converting to column to good data type'''
df_master['started_at']= pd.to_datetime(df_master['started_at'])
df_master['ended_at']=pd.to_datetime(df_master['ended_at'])
```



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5883043 entries, 0 to 785931
Data columns (total 13 columns):
 #   Column              Dtype
---  ------              -----
 0   ride_id             object
 1   rideable_type       object
 2   started_at          datetime64[ns]
 3   ended_at            datetime64[ns]
 4   start_station_name  object
 5   start_station_id    object
 6   end_station_name    object
 7   end_station_id      object
 8   start_lat           float64
 9   start_lng           float64
 10  end_lat             float64
 11  end_lng             float64
 12  member_casual       object
dtypes: datetime64[ns](2), float64(4), object(7)
memory usage: 628.4+ MB
```

❑ Calculating ride Length

```
''' calculating the ride length'''

## ride_length is in seconds

df_master['ride_length'] = df_master.ended_at-df_master.started_at
df_master['ride_length']=df_master['ride_length'].astype('timedelta64[s]') #converting it to seconds
```

❑ Getting Week Day

```
''' Getting the week day 0=monday and 6=sunday'''

df_master['day_number']=df_master['started_at'].dt.day_of_week
df_master['day_name']=df_master['started_at'].dt.day_name()

df_master['month_name']=df_master['started_at'].dt.to_period('M')
```

Process Phase

❑ Removing inconsistencies

Verify if ride_length is negative

```
df_master[df_master.ride_length < 0]
```

rt_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual	ride_length	day_number	day_name	month_name
NaN	Clybourn Ave & Division St	TA1307000115	41.890000	-87.640000	41.904613	-87.640552	casual	-7802.0	1.0	Tuesday	2022-06
NaN	Cottage Grove Ave & 63rd St	KA1503000054	41.770000	-87.600000	41.780601	-87.605836	casual	-7621.0	1.0	Tuesday	2022-06
13247	Broadway & Cornelia Ave	13278	41.895196	-87.667916	41.945529	-87.646439	member	-71.0	3.0	Thursday	2022-06
A1503000041	Clark St & Schiller St	TA1309000024	41.894503	-87.617854	41.907993	-87.631501	member	-7745.0	1.0	Tuesday	2022-06

After identifying inconsistencies, we create a new dataset with positive ride lengths

```
''' The new data frame with the accurate ride_length'''
df_master=df_master[df_master.ride_length > 0]
```

❑ Check for duplicates

```
df_master[df_master.duplicated()]
```

ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat

❑ Check for missing values

Finding Missing values in every columns

```
df_master.isnull().sum()
```

```
Out[14]: ride_id                 0
         rideable_type           0
         started_at              0
         ended_at                0
         start_station_name    884333
         start_station_id      884331
         end_station_name      946004
         end_station_id        946004
         start_lat               0
         start_lng               0
         end_lat                5727
         end_lng                5727
         member_casual           0
         ride_length             0
         day_number              0
         day_name                0
         month_name              0
         dtype: int64
```

04

ANALYZE

❑ Type of users

We can classify users from the <<member_casual>> column. These users are :

1. **Casual :** These are the target users we want to convert.
2. **Member:** These are annual users of the bike share program

❑ Calculating Average ride length

```
# calculation of the mean of ride_length

mean_value=df_master.ride_length.mean()
print('The mean of ride_length is : {} seconds'.format(mean_value))
```

```
The mean of ride_length is : 1185.366370604564 seconds
```

❑ Calculating max ride length

```
# calculation of the max of ride_length

max_value=df_master.ride_length.max()
print('The max of ride_length is : {} seconds'.format(max_value))
```

```
The max of ride_length is : 2442301.0 seconds
```

Analyze Phase

❑ Calculating Mode of day of the week

```
# calculation of the mode of the day of the week

mode_week_day=df_master.day_name.mode()

print('Mode of the day of the week is : '+str(mode_week_day))
```

```
Mode of the day of the week is : 0    Thursday
Name: day_name, dtype: object
```

❑ Exporting Final dataset for visualization

```
df_master.to_csv('final_trip.csv',index=False)
```

❑ Average ride length per user type

```
df_master.groupby(['member_casual'])['ride_length'].mean()
```

```
member_casual
casual    1758.072523
member     771.358566
Name: ride_length, dtype: float64
```

❑ Average ride length per user type by day of week

```
draw2=df_master.groupby(['member_casual','day_name'])['ride_length'].mean().unstack()
draw2
```

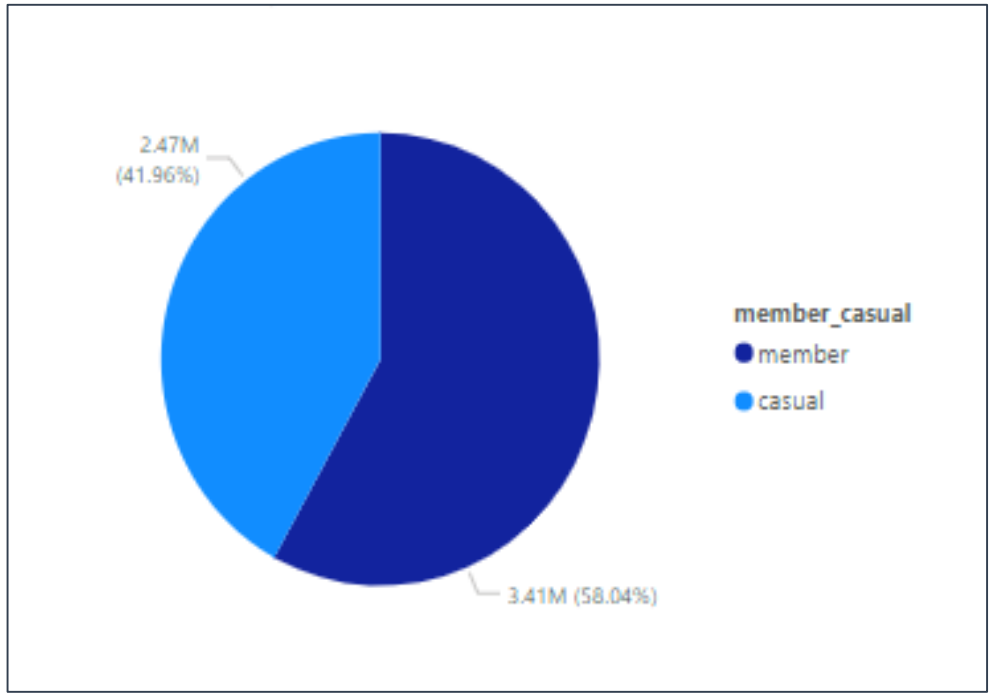
day_name	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
member_casual							
casual	1673.518862	1790.722222	1922.618228	2051.404179	1561.735976	1555.067563	1502.584880
member	755.722793	747.054532	858.159594	885.520164	742.521129	730.728589	731.830212

05

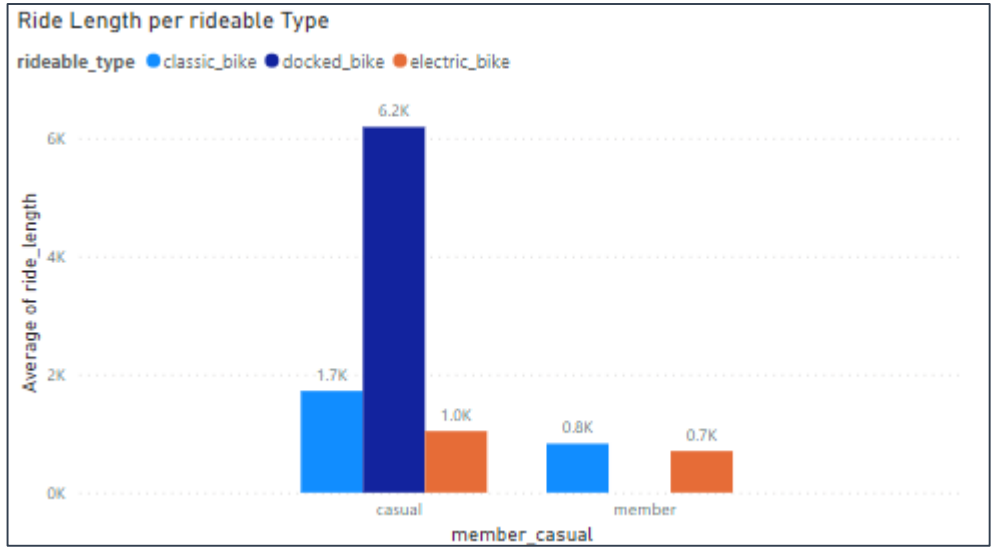
SHARE

Share Phase

- **58.04%(3.41M)** of the riders are annual members while **41.96%(2.47M)** are casual riders.
- Annual members form the majority of total riders. So increasing them will help to reach the objective

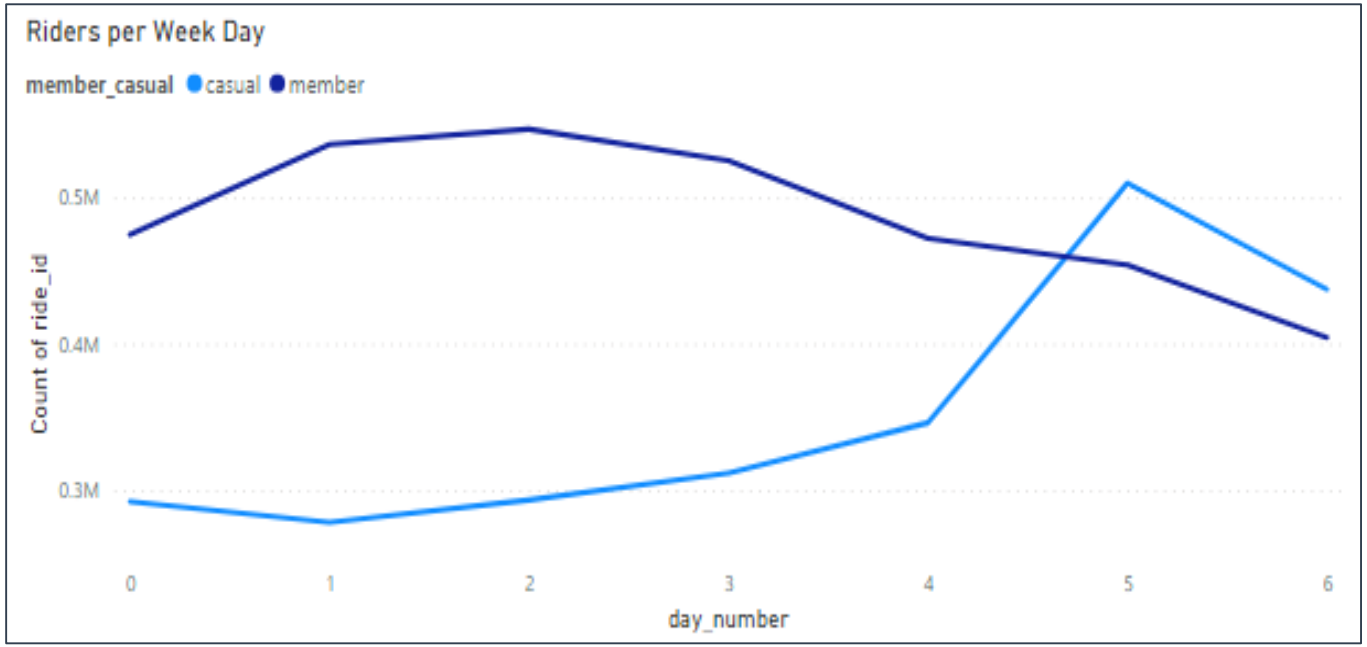


- Casual users use all bikes types
- Annual members prefer classic bikes and electric bikes
- Casual users mainly rides using docked bike



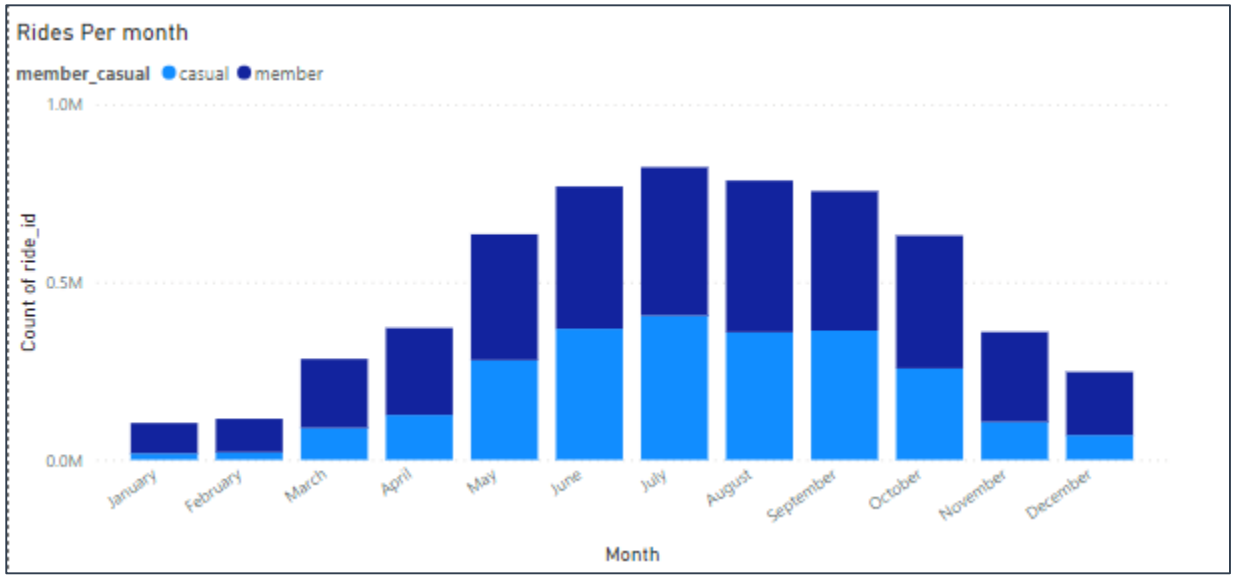
Share Phase

- Annual members rides decrease during the weekend. Annual members mainly rides during the week.
- Casual members rides are small during during the week but increase during the weekend.
- This graph shows that the casual members mainly rides for leisure.



Share Phase

- Both rider types number obey the same trends on monthly overview.
- Number of Rides increase in the year between the month of April to October.
- Number of rides decrease between November February. This correspond to Winter season.



06

ACT

Recommendations

- Special marketing campaign must be done during the weekend as rides numbers are higher during the weekend.
- Create a special campaign for docked bike as casual users prefers to ride on these bikes
- Discount can be done during the week to help casual riders to increase their rides times.
- Company should increase campaigns activities during summer seasons as riders are increasing.

THANK YOU!

Presentation by **GNINGHAYE Malcolmx Hassler**

https://www.linkedin.com/in/malcolmx-hassler-gninghaye-guemandeu-77a5b11b0/