Author: Evangelos Bikis
Date: November 20, 2021

Google Data Analytics Professional Certification: Capstone Project - Cyclistic

Case Study: How does a Bike-Share Navigate Speedy Success?

Introduction

Cyclistic is a fictional company that runs a bike-share program in Chicago that features more than 5,800 bicycles and 600 docking stations. As a junior data analyst working in the marketing analyst team at Cyclistic, we want to understand how casual riders and annual members use Cyclistic bikes differently. The senior management believes that the company's future success lies in maximizing the number of annual members. Therefore, a thorough analysis of the customer data and identifying possible trends will facilitate the decision-making process upon developing a new marketing strategy to attract more casual riders to purchase an annual membership.

The structure of this analysis follows the six-step data analysis process:

1) Ask
2) Prepare
3) Process
4) Analyze
5) Share
6) Act

For this project, I employ **Microsoft Excel spreadsheets** for the data cleaning process (Prepare, Process), **MySQL** to combine and analyze data (Analyze), and **Tableau** for the data visualization.

Ask

As instructed by management, three questions will guide the future marketing program, and this project's purpose is to answer the first question.

1. How do annual members and casual riders use Cyclistic bikes differently?

2. Why would casual riders buy Cyclistic annual memberships?

3. How can Cyclistic use digital media to influence casual riders to become members?

Business task: Analyze the historical bike trip data of the last twelve (12) months of Cyclistic, discover trends and insights regarding the bike usage of annual and casual riders. The key findings must be presented in visualizations to the stakeholders to facilitate the development of new digital marketing strategies to increase annual members.

Primary stakeholders: Director of marketing, Cyclistic executive team.

Secondary stakeholders: Cyclistic marketing analytics team

Prepare

Dataset: 12 zipped ".csv" files from November 2020 to October 2021.

Data Source: Link (AWS cloud storage)

License: Available by Motivate International Inc.

The data files were downloaded, unzipped, and saved in a subfolder of the project's folder. Then, they opened and saved as ".xls" files in a separate subfolder to keep original and edited data files in different places. Each of the initial data files consisted of 13 columns, and the total rows of the dataset were 5,378,834.

Author: Evangelos Bikis
Date: November 20, 2021

Check the credibility and potential bias in data: ROCCC

- **R**eliability: The dataset contains complete and accurate data regarding bike rides as recorded by Divvy, a bicycle-sharing service of Chicago that collects data from the city of Chicago and makes it available to the public.
- **O**riginal: Data is collected directly from people in Chicago and is available to the public (second-party data).
- **C**omprehensive: The data includes for each ride its id (column A), bike type (B), starting time (C), ending time (D), starting station (E) and id (F), ending station (G), and id (H), starting latitude (I) – longitude (J), ending latitude (K) – longitude (L), and customer's category, casual OR member (M).
- **C**urrent: Dataset dates from November 2020 to October 2021.
- **C**ited: Data is appropriately cited, as we know who the author is, the data source, and the provided data license agreement.

Data Limitations: The data-privacy issues deter using riders' personally identifiable information, so it is impossible to connect bike passes purchased to credit card numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes. Thus, single, or numerous rides cannot be linked to individuals. In addition, the dataset contains some missing values within most of the files, which are located in the starting or ending stations name and id and ending latitude or longitude columns.

Process

Tool: Microsoft Excel Spreadsheet

The following columns were added to each file:

Column	Description	Formula
N: ride_length	Transacts ending and starting time	=D2-C2
O: ride_length_m	Calculated total ride length in minutes	=(T2*24*60)+(HOUR(N2)*60)+MINUTE(N2)+SECOND(N2)/60
T: ride_days	Calculates the days' difference between ending and starting time	=DAYS(D2,C2)
P: day_of_week	Returns 1 for Sunday and 7 for Saturday	=WEEKDAY(C2,1)
Q: day_name	Transforms integers into day names	=IF(P2=1,"Sunday",IF(P2=2,"Monday",IF(P2=3,"Tuesday",IF(P2=4,"Wednesday",IF(P2=5,"Thursday",IF(P2=6,"Friday","Saturday"))))))
R: hour_of_day	Hour in a day from 0 to 23	=HOUR(C2)
S: part_of_day	Based on hour	=IF(AND(R2>=5,R2<=11),"Morning",IF(AND(R2>11,R2<=16),"Afternoon",IF(AND(R2>16,R2<=20),"Evening","Night")))
U: month_num	From 1 to 12	=MONTH(C2)
V: month	Name of month based on integer	=IF(U2=1,"January",IF(U2=2,"February",IF(U2=3,"March",IF(U2=4,"April",IF(U2=5,"May",IF(U2=6,"June",IF(U2=7,"July",IF(U2=8,"August",IF(U2=9,"September",IF(U2=10,"October",IF(U2=11,"November","December")))))))))))
W: season	Based on month	=IF(OR(V2="September",V2="October",V2="November"),"Autumn",IF(OR(V2="December",V2="January",V2="February"),"Winter",IF(OR(V2="March",V2="April",V2="May"),"Spring","Summer")))

I resaved all ".xls" files into ".csv" files in a different subfolder to keep track of all the steps during the data preparation and cleaning process in case of iterating any step. I checked all files for errors that could lead to problems during the analysis phase. Then, I checked for errors or flaws within all the data files and removed all incomplete or problematic rows as shown below:

- ### value: Filter data in column "ride_length_m", type "#", select only the values that contain "#" from the dropdown list, and delete all problematic rows that appear in each file.
- 0 value: Filter data in column "ride_length_m", select only the values that contain "0" from the dropdown list, and delete all problematic rows that appear in each file.
- Missing values: Filter data in columns "end_lat" or "end_lng", type "" (blank), and delete all problematic rows from the dropdown list that appear in each file.

Then, I removed the following columns to filter out unnecessary data for my analysis:

ride_id, started_at, ended_at, start_station_name, start_station_id, end_station_name, end_station_id, ride_length, day_of_week, ride_days, month_num

(Note: I should have also removed columns with latitude and longitude as they hardly provided any insights for this particular project. Geographical data had no impact on this project to support developing a new digital media strategy, and I also knew all rides are referring to Chicago)

After the cleaning process took place, the data files ended up as follows:

File	Initial Count of rows	Rows with errors	Final Count of rows
202011	259716	1171	258545
202012	131573	551	131022
202101	96834	108	96726
202102	49622	218	49404
202103	228496	179	228317
202104	337230	305	336925
202105	531633	506	531127
202106	729595	783	728812
202107	822410	813	821597
202108	804352	813	803539
202109	756147	702	755445
202110	631226	554	630672
Total	**5378834**	**6703**	**5372131**

Analyze

Tool: MySQL Workbench 8.0.27

I created a database named "cyclistic_project" in MySQL Workbench.

Create database

```
CREATE DATABASE cyclistic_project;

USE cyclistic_project;
```

Create table “trips” to combine all files into one table to increase speed

```
CREATE TABLE trips (

ride_type VARCHAR(255),

start_lat DECIMAL(10,8),

start_lng DECIMAL(11,8),
end_lat DECIMAL(10,8),

end_lng DECIMAL(11,8),

customer_type VARCHAR(255),

ride_minutes DECIMAL(10,2),

ride_day VARCHAR(255),

ride_hour INT,

part_day VARCHAR(255),

ride_month VARCHAR(255),

ride_season VARCHAR(255));
```

Import each file into the table

```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/trips_202011.csv"

INTO TABLE trips

FIELDS TERMINATED BY "," ENCLOSED BY '"'

LINES TERMINATED BY "\n"

IGNORE 1 ROWS;
```

(Same for the rest of the ".csv" files, just replacing the name of the file: trips_202012, trips_202101, trips_202102, trips_202103, trips_202104, trips_202105, trips_202106, trips_202107, trips_202108, trips_202109, trips_202110)

Check if all imported data is correct by comparing COUNT(*) with csv's rows and the sum of ride_length_m with SUM(ride_min). All query's results are exported as a ".csv" file and displayed as a table after each query.

```
SELECT COUNT(*) AS Total_Rows,

SUM(ride_min) AS Total_Ride_Length_Minutes

FROM trips;
```

Imported Data from clean CSV files		
File	**Number of Rows**	**Sum of ride_length_m**
202011	258,545	6,193,115.54
202012	131,022	2,509,246.32
202101	96,726	1,790,979.15
202102	49,404	1,386,560.88
202103	228,317	6,650,809.39
202104	336,925	11,282,440.56
202105	531,127	20,124,525.89
202106	728,812	28,948,667.37
202107	821,597	29,566,472.18
202108	803,539	24,948,362.54
202109	755,445	22,043,442.89
202110	630,672	15,712,215.39
Total	**5,372,131**	**171,156,838.10**

Author: Evangelos Bikis
Date: November 20, 2021

Summary statistics

```
SELECT
customer_type,
COUNT(*) AS total_num,
MAX(ride_minutes) AS Longest_ride,
MIN(ride_minutes) AS Shortest_ride,
AVG(ride_minutes) AS Average_ride
FROM trips
GROUP BY 1
ORDER BY 2 DESC;
```

customer_type	total_num	Longest_ride	Shortest_ride	Average_ride
member	2905532	2873.73	0.02	18.229471
casual	2466599	57131.68	0.02	47.916393

```
SELECT
ride_type,
COUNT(*) AS total_type,
MAX(ride_minutes) AS Longest_ride,
MIN(ride_minutes) AS Shortest_ride,
AVG(ride_minutes) AS Average_ride
FROM trips
GROUP BY 1
ORDER BY 5 DESC;
```

ride_type	total_type	Longest_ride	Shortest_ride	Average_ride
docked_bike	462744	57131.68	0.02	84.075066
classic_bike	3062390	2939.93	0.02	27.633536
electric_bike	1846997	1927.27	0.02	25.78615

Top Days by Customer's category

```
(SELECT
customer_type,
ride_day,
COUNT(ride_day) AS num_days
FROM trips
GROUP BY 1,2
HAVING customer_type = "casual"
ORDER BY 3 DESC LIMIT 1)
UNION
(SELECT
customer_type,
ride_day,
COUNT(ride_day) AS num_days
FROM trips
GROUP BY 1,2
HAVING customer_type = "member"
ORDER BY 3 DESC LIMIT 1);
```

customer_type	ride_day	num_days
casual	Saturday	550971
member	Wednesday	444026

Top bike type per customer’s category

```
(SELECT
customer_type,
ride_type,
COUNT(ride_type) AS top_ride_type
FROM trips
GROUP BY 1,2
HAVING customer_type = "casual"
ORDER BY 3 DESC LIMIT 1)
UNION
(SELECT
customer_type,
ride_type,
COUNT(ride_type) AS top_ride_type
FROM trips
GROUP BY 1,2
HAVING customer_type = "member"
ORDER BY 3 DESC LIMIT 1);
```

Author: Evangelos Bikis
Date: November 20, 2021

customer_type	**ride_type**	**top_ride_type**
casual	classic_bike	1223122
member	classic_bike	1839268

Top ride day per customer type

```
(SELECT
customer_type,
ride_day,
COUNT(ride_day) AS top_ride_day
FROM trips
GROUP BY 1,2
HAVING customer_type = "casual"
ORDER BY 3 DESC LIMIT 1)
UNION
(SELECT
customer_type,
ride_day,
COUNT(ride_day) AS top_ride_day
FROM trips
GROUP BY 1,2
HAVING customer_type = "member"
ORDER BY 3 DESC LIMIT 1);
```

customer_type	**ride_day**	**top_ride_day**
casual	Saturday	550971
member	Wednesday	444026

Top part of day per customer type

```
(SELECT
customer_type,
part_day,
COUNT(part_day) AS top_part_day
FROM trips
GROUP BY 1,2
HAVING customer_type = "casual"
ORDER BY 3 DESC LIMIT 1)
UNION
(SELECT
customer_type,
part_day,
COUNT(part_day) AS top_part_day
FROM trips
GROUP BY 1,2
HAVING customer_type = "member"
ORDER BY 3 DESC LIMIT 1);
```

customer_type	**part_day**	**top_part_day**
casual	Afternoon	895184
member	Afternoon	952018

Top month

```
(SELECT
customer_type,
ride_month,
COUNT(ride_month) AS top_ride_month
FROM trips
GROUP BY 1,2
HAVING customer_type = "casual"
ORDER BY 3 DESC LIMIT 1)
UNION
(SELECT
customer_type,
ride_month,
COUNT(ride_month) AS top_ride_month
FROM trips
GROUP BY 1,2
HAVING customer_type = "member"
ORDER BY 3 DESC LIMIT 1);
```

customer_type	**ride_month**	**top_ride_month**
casual	July	441428
member	September	392028

Author: Evangelos Bikis
Date: November 20, 2021

Top season per customer type

```
(SELECT

customer_type,

ride_season,

COUNT(ride_season) AS top_season

FROM trips

GROUP BY 1,2

HAVING customer_type = "casual"

ORDER BY 3 DESC LIMIT 1)

UNION

(SELECT

customer_type,

ride_season,

COUNT(ride_season) AS top_season

FROM trips

GROUP BY 1,2

HAVING customer_type = "member"

ORDER BY 3 DESC LIMIT 1);
```

customer_type	ride_season	top_season
casual	Summer	1223586
member	Summer	1130362

Export the table with all data into one file

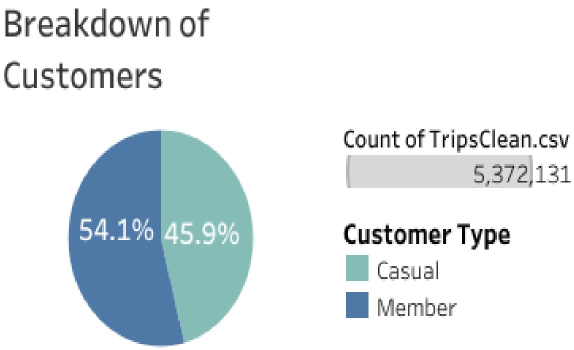
```
SELECT *

INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Tripsclean.csv'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

ESCAPED BY '\\'

LINES TERMINATED BY '\n'

FROM trips;
```

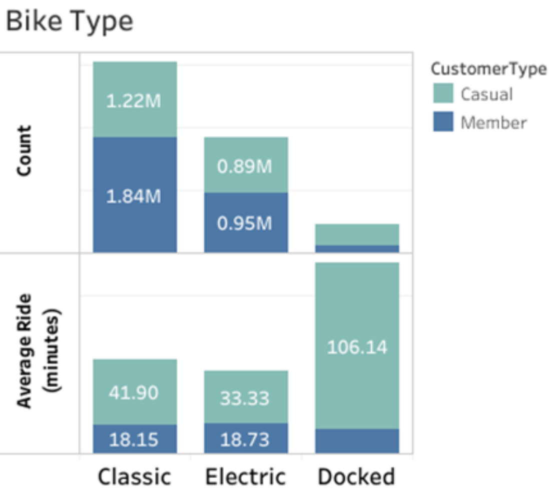
Author: Evangelos Bikis
Date: November 20, 2021

Share

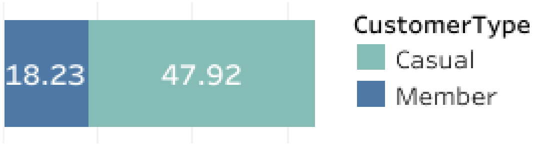
Tool: Tableau Public

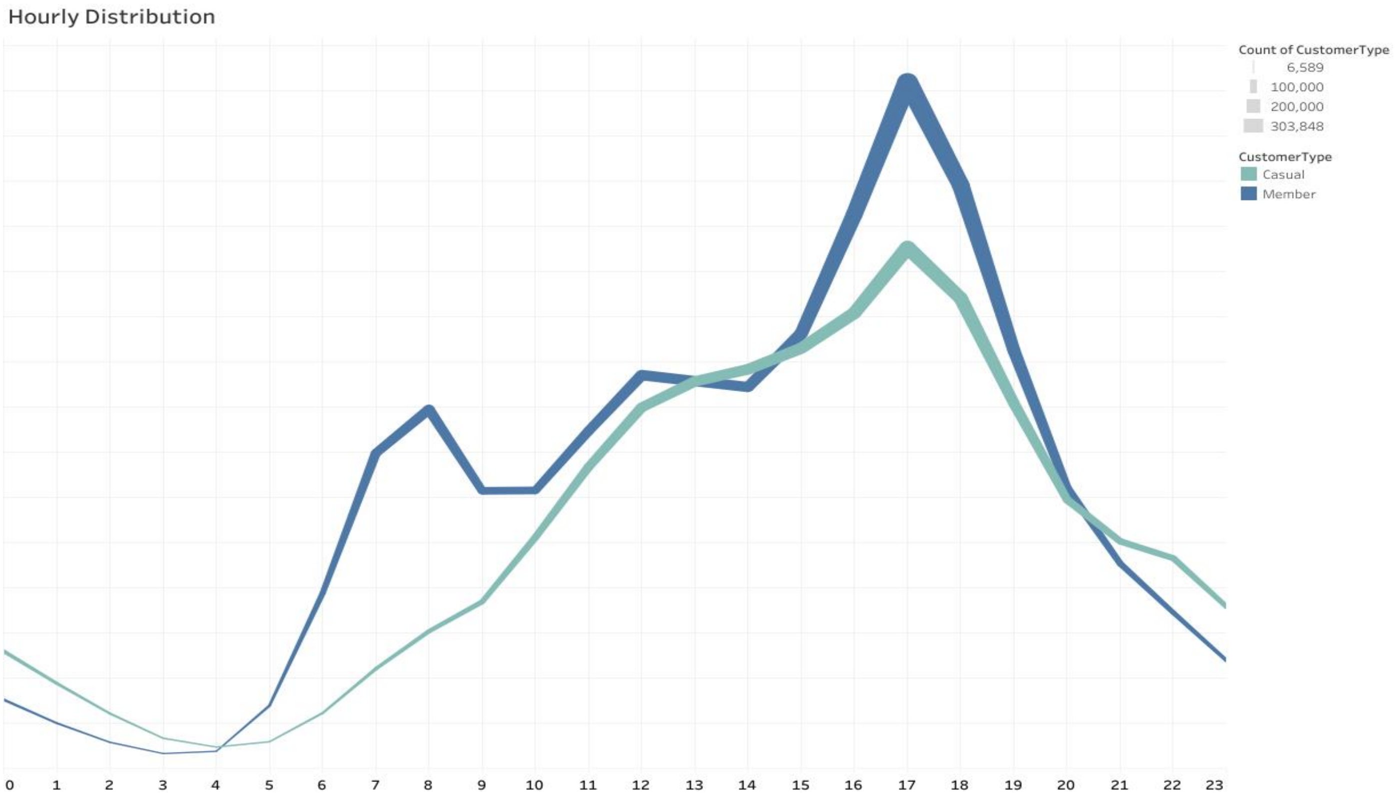
Since I created the ".csv" file with all the cleaned and ready to go for analysis data, I imported it to Tableau and created data visualizations.



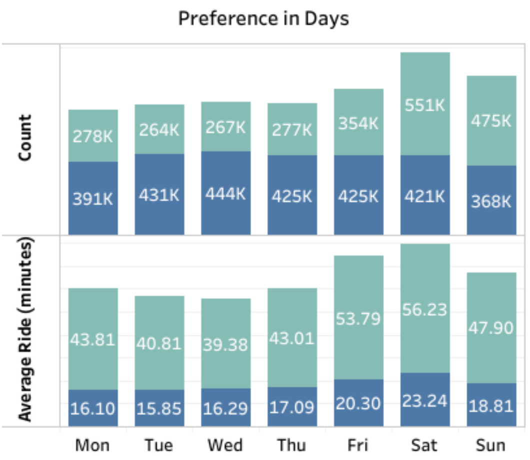


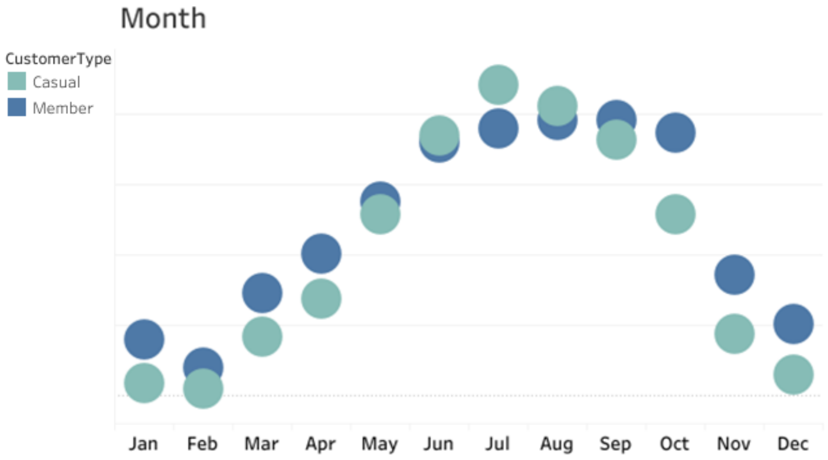
Average ride (in minutes)

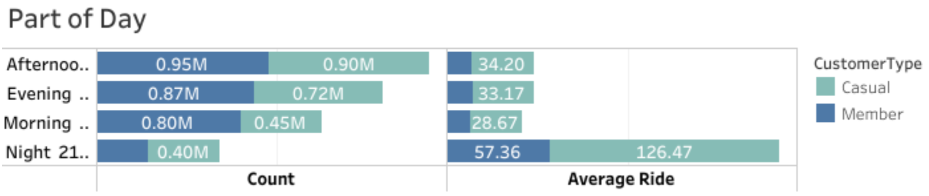


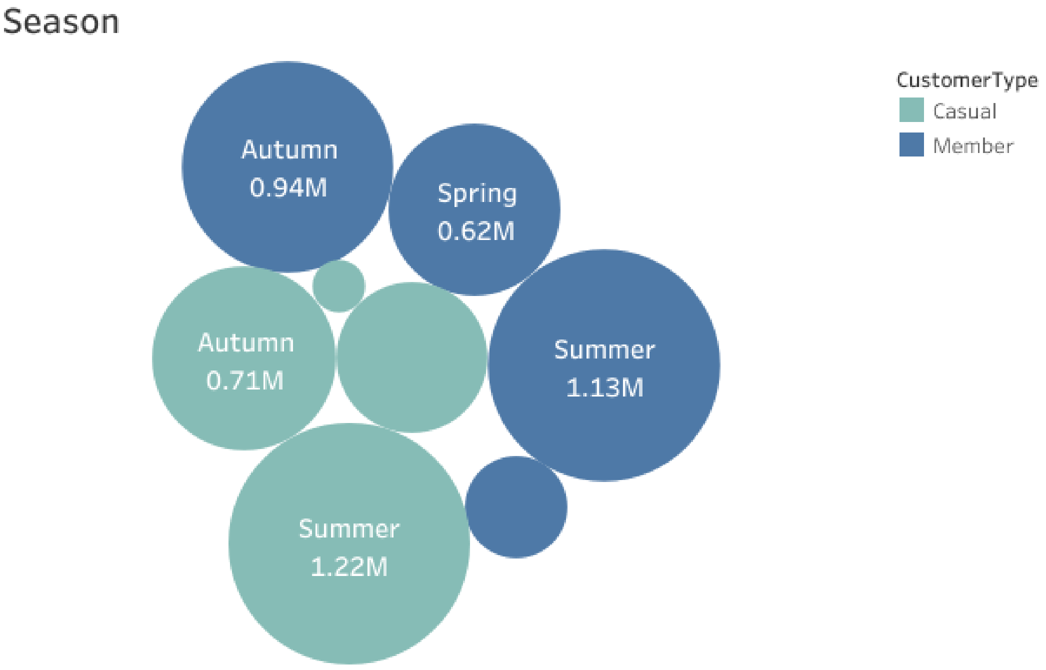


Author: Evangelos Bikis
Date: November 20, 2021



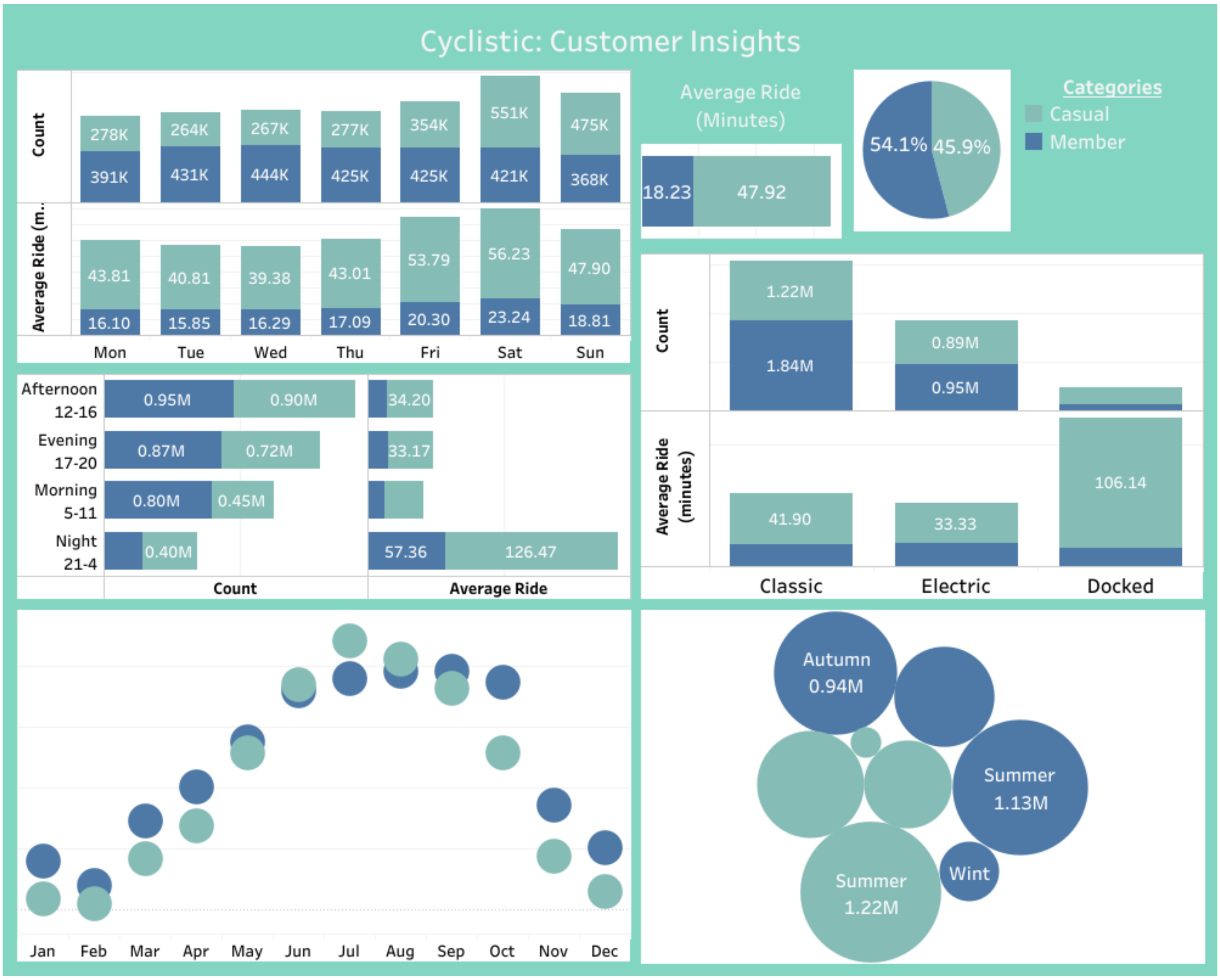




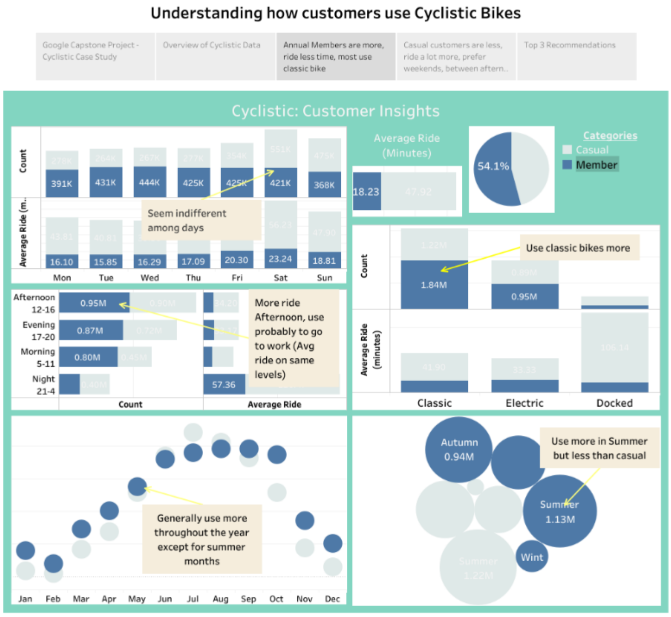


Author: Evangelos Bikis
Date: November 20, 2021

Tableau Dashboard combining relevant data visualization graphs



I created a Tableau story to best present the key findings of my analysis.



Understanding how customers use Cyclistic Bikes

Google Capstone Project - Cyclistic Case Study | Overview of Cyclistic Data | Annual Members are more, ride less time, most use classic bike | Casual customers are less, ride a lot more, prefer weekends, between aftern.. | Top 3 Recommendations

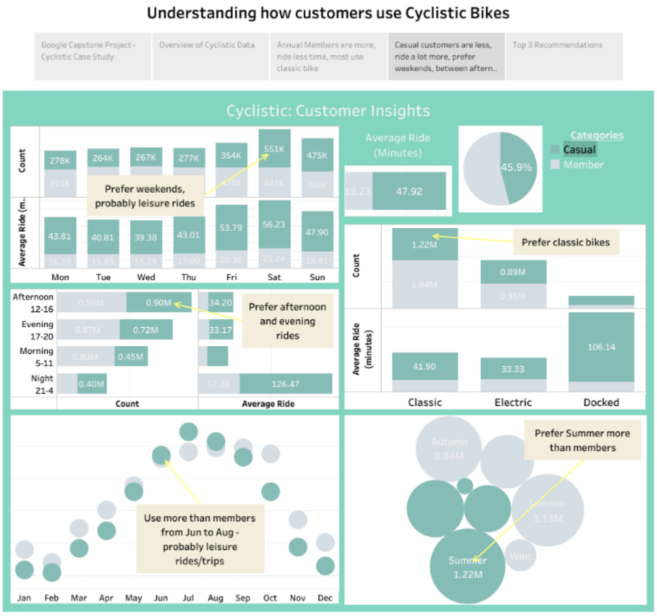
Seem indifferent among days

More ride Afternoon, use probably to go to work (Avg ride on same levels)

Use classic bikes more

Generally use more throughout the year except for summer months

Use more in Summer but less than casual



Understanding how customers use Cyclistic Bikes

Google Capstone Project - Cyclistic Case Study | Overview of Cyclistic Data | Annual Members are more, ride less time, most use classic bike | Casual customers are less, ride a lot more, prefer weekends, between aftern.. | Top 3 Recommendations

Prefer weekends, probably leisure rides

Prefer afternoon and evening rides

Prefer classic bikes

Use more than members from Jun to Aug - probably leisure rides/trips

Prefer Summer more than members

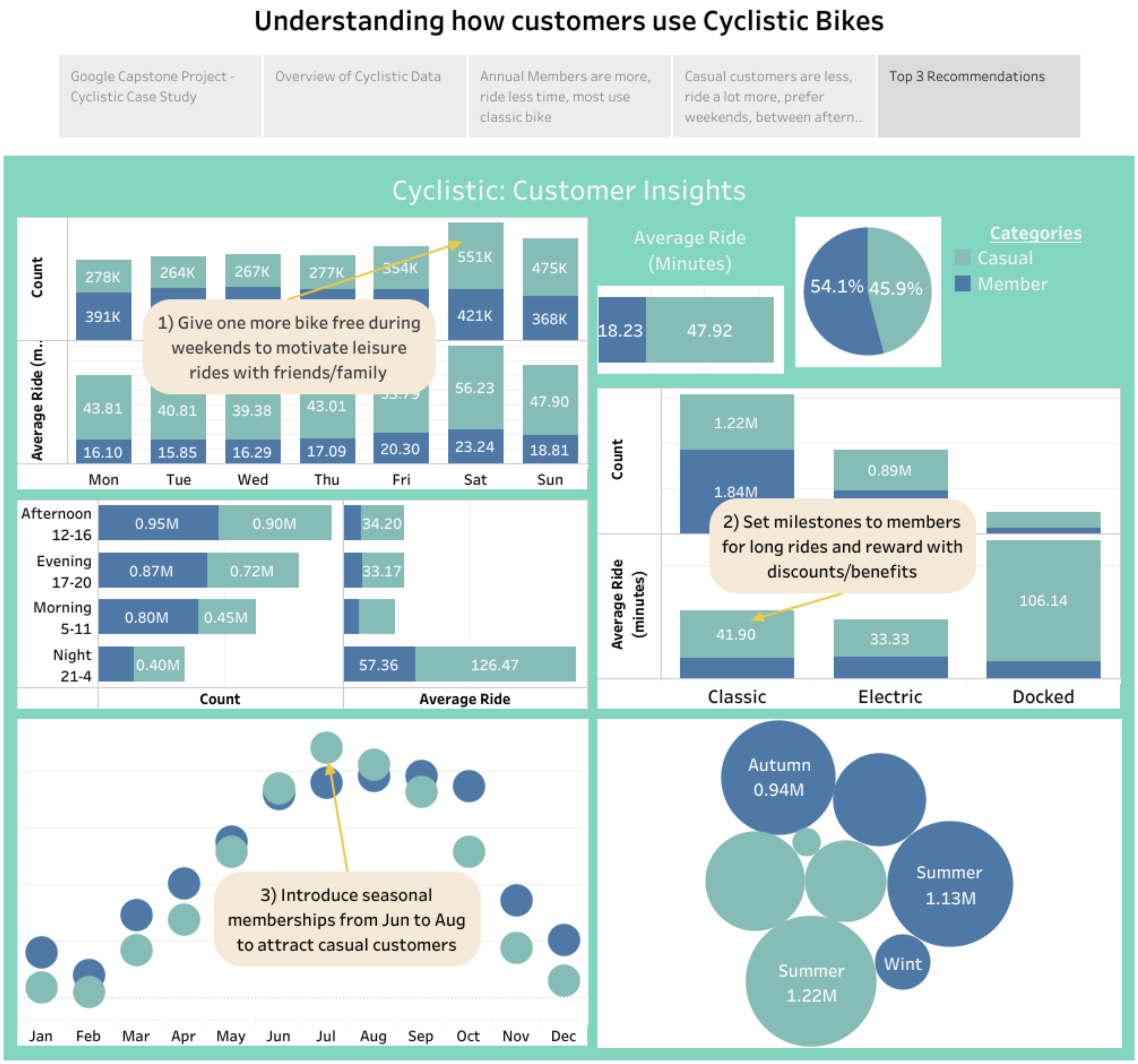
Author: Evangelos Bikis
Date: November 20, 2021

Act

Based on the key findings of the analysis, the decision-makers are to form a new digital media marketing strategy to convert casual riders to annual members.

My top three recommendations are:

1) Give one more bike free during weekends to motivate leisure rides with friends or family.
2) Set milestones to members for long rides and reward with discounts or benefits.
3) Introduce seasonal memberships only for summer months to attract casual riders that don't want an annual membership.



Understanding how customers use Cyclistic Bikes

1) Give one more bike free during weekends to motivate leisure rides with friends/family

2) Set milestones to members for long rides and reward with discounts/benefits

3) Introduce seasonal memberships from Jun to Aug to attract casual customers