Google Data analytics Capstone

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

Created by: Ahmed Marsaoui

Date: June, 2022

Outline

- Introduction
- Methodology
- Defining objectives
- Data sources
- EDA and data manipulation
- Data analysis
- Data visualization
- Results
- Conclusion

Introduction

- The bike share program « Cyclistic » is more and more popular
- Goals differ from one user to another within Chicago city
- Two main users categories with different trends:
    1. Casual riders
    2. Cyclistic members
- The second category generates more profit thanks to their annual membership

➔ **Need of the difference study to help convert casual riders into annual members**

Methodology

- Based on the data analysis process: Ask, prepare, process, analyse, share and act
- Each step is linked into one part from those mentioned in the introduction
- The use of python for Data analyses
- The use of python for the Data visualization
- The use of Github as the sharing medium of the obtained results

ASK: Defining the projects' objectives

- Analyzing the riders' usage patterns to distinguish main differences
- Document All data sources and their EDA
- Results must be clearly visualized
- Recommendation will be delivered to help define the membership conversion program

Key stakeholders include: Cyclistic executive team, Director of Marketing (Lily Moreno), Marketing Analytics team.

PREPARE: Data sources

- Use of a public data source
- A csv file with 13 columns and 551 480 registrations
- Fields' types are: Strings, dates and floats including the rider type (casual/ member)
- Data credibility : Due to the fact that this is a case study using public data, we are going to assume the data is credible.

Process: EDA and data manipulations (1)

- Use of:
  1. spreadsheet for direct data exploration
  2. Python for its advanced EDA operations
- Empty cells occurs in the Start and End stations of each ride

	E	F	G	H	I	J	K	L	M
	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
6	Wells St & Walton St	46			41.89993	-87.63443			member
9	Clark St & Lincoln Ave	141			41.915689	-87.6346			member
2	McClurg Ct & Erie St	142			41.894503	-87.617854			member
1	Dearborn St & Erie St	110			41.893992	-87.629318			casual
3	Sheffield Ave & Willov	93			41.913688	-87.652855			member

	C	D	E	F	G	H	I	J	K	L	M
	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
	31/07/2020 08:30	31/07/2020 08:57			Racine Ave & 35th St	367	41.9	-87.69	41.83070433	-87.6560845	member
	29/07/2020 19:02	29/07/2020 19:22			Western Ave & Walto	374	41.9	-87.69	41.89840433	-87.68659233	member
	30/07/2020 22:02	30/07/2020 22:17			Clarendon Ave & Lelar	251	41.94	-87.65	41.9678415	-87.6499905	member
	31/07/2020 15:54	31/07/2020 16:00					41.92	-87.7	41.91	-87.68	member
	31/07/2020 16:08	31/07/2020 16:15					41.91	-87.68	41.92	-87.7	member
	31/07/2020 12:56	31/07/2020 13:13			Aberdeen St & Randol	621	41.86	-87.63	41.88404066	-87.6543005	member
	30/07/2020 08:32	30/07/2020 08:40			Canal St & Madison St	174	41.87	-87.65	41.8817595	-87.6401626	member

Process: EDA and data manipulations (2)

- Adding a relevant columns: duration of the Trip

K	L	M	N	O	P
end_lat	end_lng	member_casual	duration		
41.83070433	-87.6560845	member	27,3333333		
41.89840433	-87.68659233	member	20,25		
41.9678415	-87.6499905	member	=([@[ended_at]]-[@[started_at]])*144(		
41.91	-87.68	member	5,9		
41.92	-87.7	member	7,05		
41.88404066	-87.6543005	member	16,5666667		
41.8817595	-87.64016266	member	7,83333333		
41.9	-87.62	member	0,28333334		

- Getting a closer look on the Dataframe o confirm the missing data

```
cyclistic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 551480 entries, 0 to 551479
Data columns (total 14 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   ride_id             551480 non-null  object
 1   rideable_type       551480 non-null  object
 2   started_at          551480 non-null  object
 3   ended_at            551480 non-null  object
 4   start_station_name  551331 non-null  object
 5   start_station_id    551328 non-null  float64
 6   end_station_name    550513 non-null  object
 7   end_station_id      550511 non-null  float64
 8   start_lat           551480 non-null  float64
 9   start_lng           551480 non-null  float64
 10  end_lat             550710 non-null  float64
 11  end_lng             550710 non-null  float64
 12  member_casual       551480 non-null  object
 13  duration            551480 non-null  object
dtypes: float64(6), object(8)
memory usage: 58.9+ MB
```

Process: EDA and data manipulations (3)

- Converting the start and end dates into datetime variable
- Adding a day of the week column for later user

```
for column in ['started_at','ended_at']:
  cyclistic[column]= pd.to_datetime(cyclistic[column])
cyclistic["day_of_week"]=cyclistic['started_at'].dt.day_of_week
cyclistic
```

- Deleting rows with missing data (best approach since the dataset is large enough to analyze without them)
- New samples: 550 425

```
cyclistic=cyclistic.dropna()
cyclistic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 550425 entries, 0 to 551479
Data columns (total 15 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   ride_id             550425 non-null  object
 1   rideable_type       550425 non-null  object
 2   started_at          550425 non-null  datetime64[ns]
 3   ended_at            550425 non-null  datetime64[ns]
 4   start_station_name  550425 non-null  object
 5   start_station_id    550425 non-null  float64
 6   end_station_name    550425 non-null  object
 7   end_station_id      550425 non-null  float64
 8   start_lat           550425 non-null  float64
 9   start_lng           550425 non-null  float64
 10  end_lat             550425 non-null  float64
 11  end_lng             550425 non-null  float64
 12  member_casual       550425 non-null  object
 13  duration            550425 non-null  object
 14  day_of_week         550425 non-null  int64
dtypes: datetime64[ns](2), float64(6), int64(1), object(6)
memory usage: 67.2+ MB
```

Analyse: Calculations (1)

```
cyclistic['duration']=cyclistic['duration'].str.replace(',','.')
cyclistic['duration']=cyclistic['duration'].astype(float)
cyclistic.describe()
```

	start_station_id	end_station_id	start_lat	start_lng	end_lat	end_lng	duration
count	550425.000000	550425.000000	550425.000000	550425.000000	550425.000000	550425.000000	550425.000000
mean	221.579761	222.373920	41.904770	-87.643008	41.905052	-87.643300	37.831988
std	159.906868	160.206544	0.041075	0.024422	0.041223	0.024577	383.691565
min	2.000000	2.000000	41.736646	-87.774704	41.729267	-87.774704	-120.300000
25%	94.000000	94.000000	41.882664	-87.654787	41.882830	-87.655486	9.916670
50%	195.000000	195.000000	41.899643	-87.638973	41.900363	-87.639192	17.883330
75%	308.000000						
max	683.000000						

```
cyclistic= cyclistic[cyclistic['duration'] >= 0]
cyclistic.describe()
```

	start_station_id	end_station_id	start_lat	start_lng	end_lat	end_lng	duration
count	548681.000000	548681.000000	548681.000000	548681.000000	548681.000000	548681.000000	548681.000000
mean	221.587549	222.342086	41.904761	-87.643015	41.905042	-87.643301	37.955599
std	159.925371	160.200394	0.041078	0.024431	0.041227	0.024583	384.294495
min	2.000000	2.000000	41.736646	-87.774704	41.729267	-87.774704	0.000000
25%	94.000000	94.000000	41.882664	-87.654787	41.882830	-87.655486	9.983330
50%	195.000000	195.000000	41.899643	-87.638973	41.900219	-87.639192	17.950000
75%	308.000000	309.000000	41.929546	-87.626217	41.929567	-87.626761	31.900000
max	683.000000	683.000000	42.064854	-87.549386	42.064854	-87.549386	49965.450000

Analyse: Calculations (2)

```
#number of rides per member type
cyclistic[['ride_id','member_casual']].groupby(by=['member_casual']).count()
```

member_casual	ride_id
casual	268125
member	280556

```
#ride_length calculations
#Members vs. Casual Riders
cyclistic[['duration','member_casual']].groupby(by=['member_casual']).mean()
```

member_casual	duration
casual	59.284604
member	17.571648

Analyse: Calculations (3)

```
#number of rides per day of the week per member type
cyclistic[['ride_id','day_of_week','member_casual']].groupby(by=['day_of_week','member_casual']).count()
```

		ride_id
day_of_week	member_casual	
Friday	casual	44872
	member	49369
Monday	casual	39737
	member	45074
Saturday	casual	52568
	member	39823
Sunday	casual	26231
	member	22806
Thursday	casual	34433
	member	37429
Tuesday	casual	45370
	member	52185
Wednesday	casual	24914
	member	33870

```
#number of rides per day of the week
#modifying value of the cells to show the day name instead of the day number
cyclistic['day_of_week']=cyclistic['started_at'].dt.day_name()
cyclistic['day_of_week'].value_counts()
```

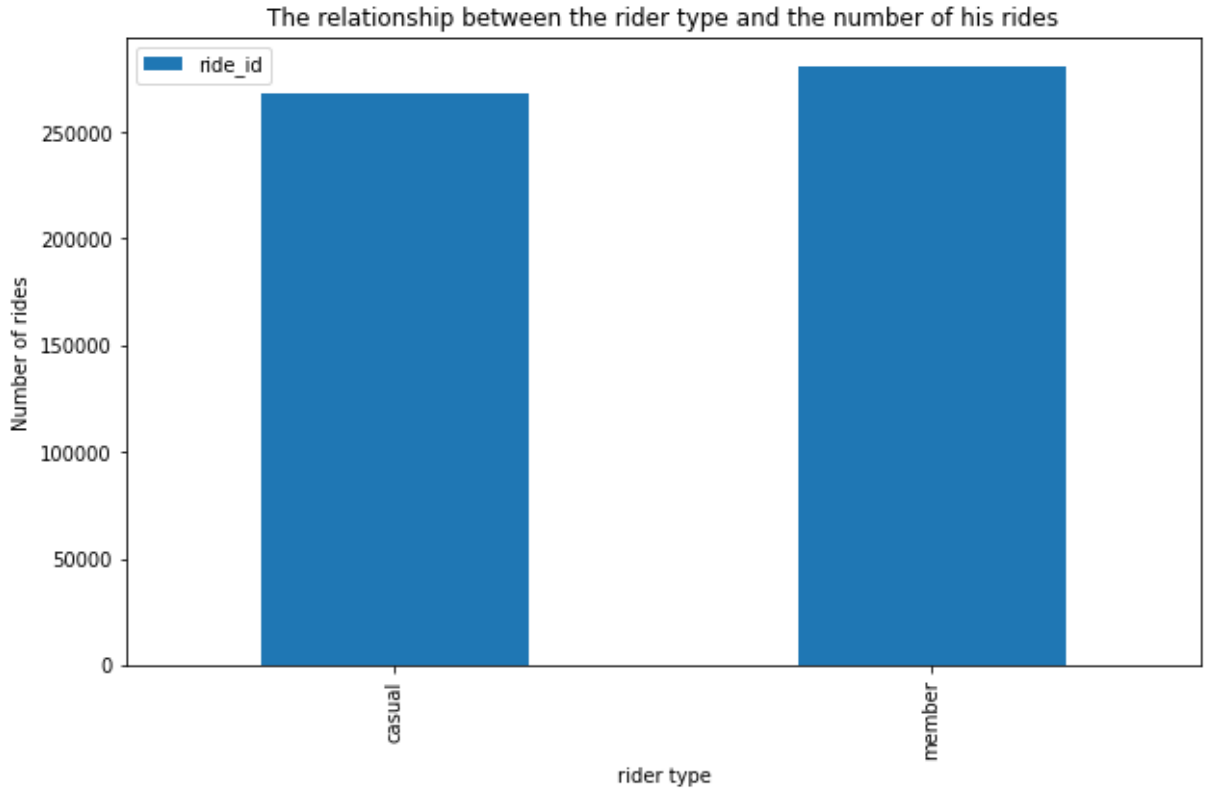
```
Tuesday      97555
Friday       94241
Saturday     92391
Monday       84811
Thursday     71862
Wednesday    58784
Sunday       49037
Name: day_of_week, dtype: int64
```

Analyse: Identify trends and relationships

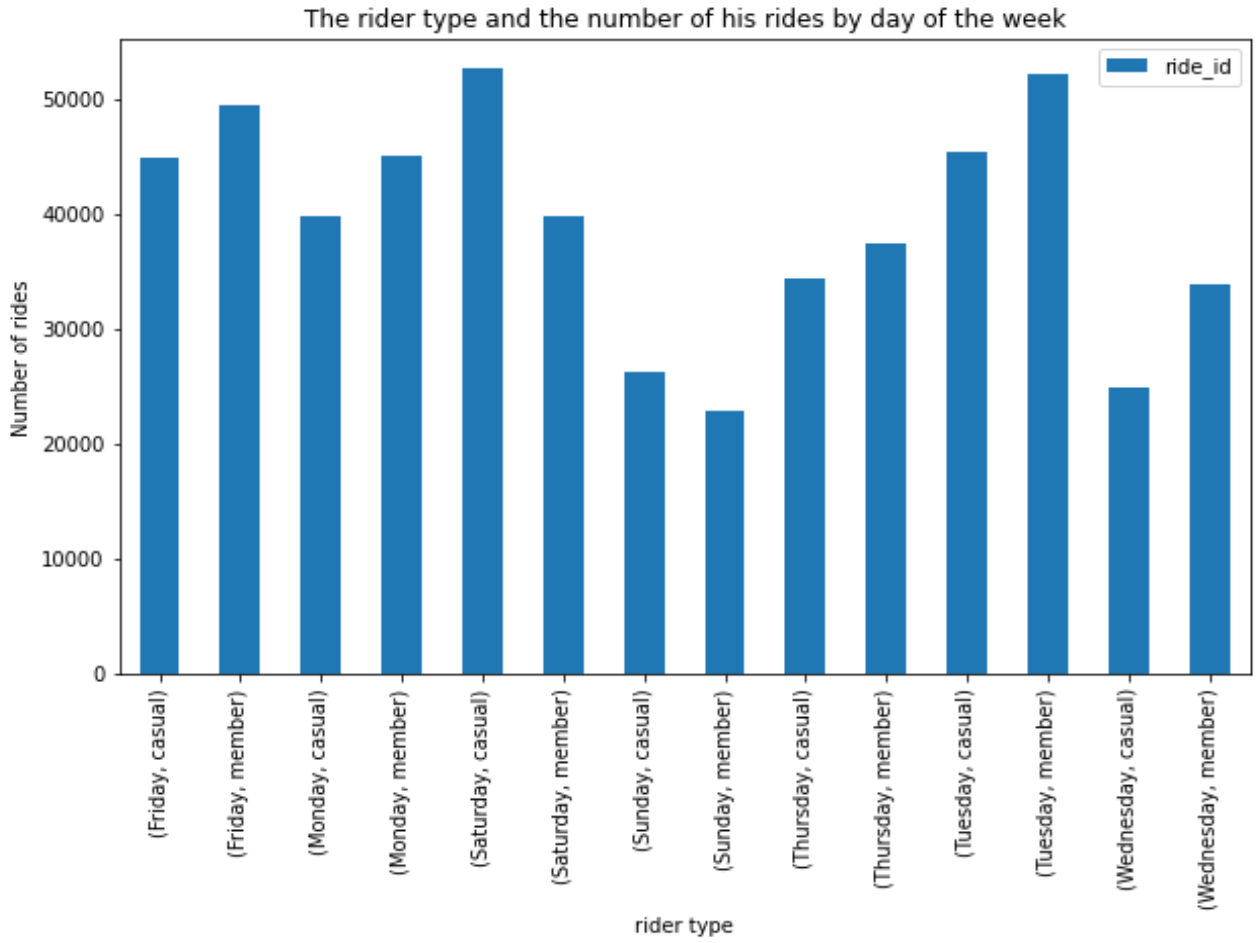
- Although that the number of rides executed by the subscribed members is higher, the rides of casual members is 3,5 longer (sightseeing vs regular trip from an to work/ school)
- Except for Tuesday, the rides number increases on both Friday and Saturday and decline dramatically on Sunday

Share: Data visualisation

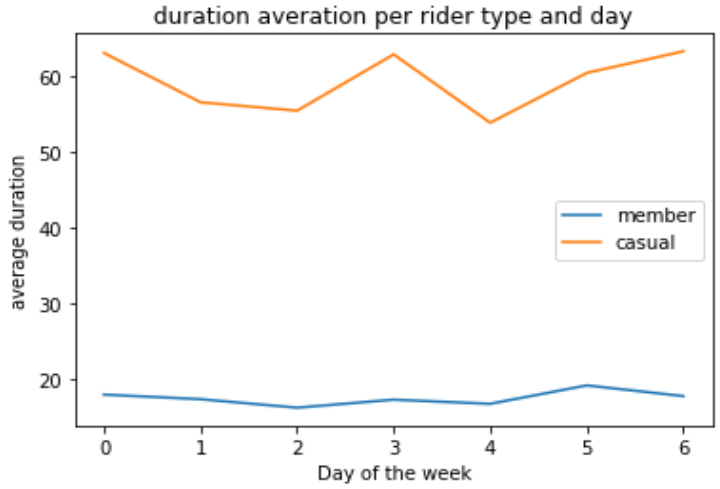
```
#Number of rides by rider type
import matplotlib.pyplot as plt
df=cyclistic[['ride_id','member_casual']].groupby(by=['member_casual']).count()
df.plot(kind='bar',figsize=(10,6))
plt.xlabel('rider type') # add to x-label to the plot
plt.ylabel('Number of rides') # add y-label to the plot
plt.title('The relationship between the rider type and the number of his rides') #
plt.show()
```



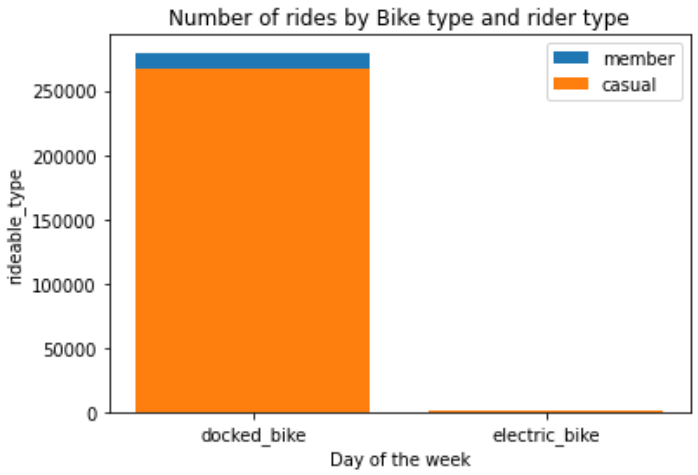
```
df=cyclistic[['ride_id','day_of_week','member_casual']].groupby(by=['day_of_week','member_casual']).count()
df.plot(kind='bar',figsize=(10,6))
plt.xlabel('rider type') # add to x-label to the plot
plt.ylabel('Number of rides') # add y-label to the plot
plt.title('The rider type and the number of his rides by day of the week') # add title to the plot
plt.show()
```



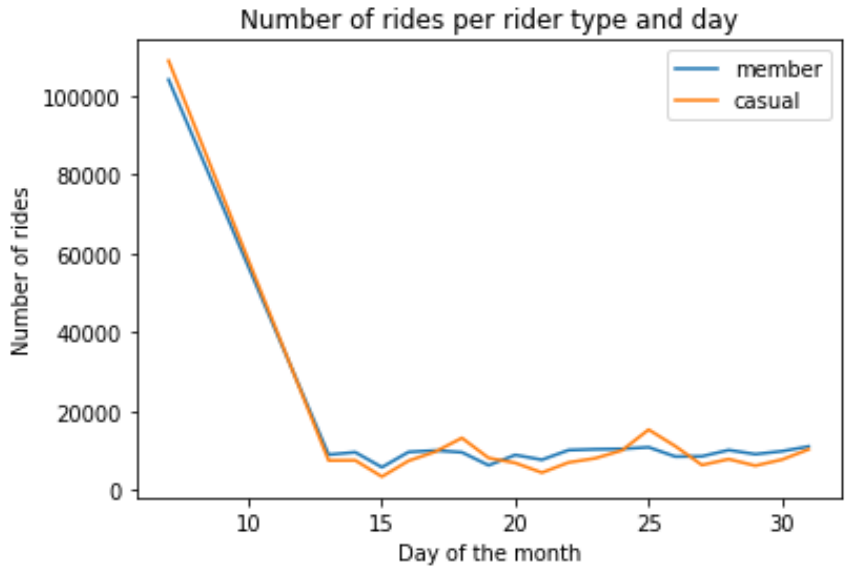
```
#cyclistic['day_of_week']=cyclistic['started_at'].dt.day_of_week
df=cyclistic[cyclistic['member_casual']=='member'][['duration','day_of_week']].groupby(by=['day_of_week']).mean()
df2=cyclistic[cyclistic['member_casual']=='casual'][['duration','day_of_week']].groupby(by=['day_of_week']).mean()
plt.plot(df.index,df['duration'],label='member')
plt.plot(df2.index,df2['duration'],label='casual')
plt.xlabel('Day of the week') # add to x-label to the plot
plt.ylabel('average duration') # add y-label to the plot
plt.title('duration averation per rider type and day') # add title to the plot
plt.legend()
plt.show()
```



```
#cyclistic['day_of_week']=cyclistic['started_at'].dt.day_of_week
df=cyclistic[cyclistic['member_casual']=='member'][['rideable_type','day_of_week']].groupby(by=['rideable_type']).count()
df2=cyclistic[cyclistic['member_casual']=='casual'][['rideable_type','day_of_week']].groupby(by=['rideable_type']).count()
plt.bar(df.index,df['day_of_week'],label='member')
plt.bar(df2.index,df2['day_of_week'],label='casual')
plt.xticks(df.index, df2.index)
plt.xlabel('Day of the week') # add to x-label to the plot
plt.ylabel('rideable_type') # add y-label to the plot
plt.title('Number of rides by Bike type and rider type') # add title to the plot
plt.legend()
plt.show()
```



```
df=cyclistic[cyclistic['member_casual']=='member'][['ride_id','day']].groupby(by=['day']).count()
df2=cyclistic[cyclistic['member_casual']=='casual'][['ride_id','day']].groupby(by=['day']).count()
plt.plot(df.index,df['ride_id'],label='member')
plt.plot(df2.index,df2['ride_id'],label='casual')
plt.xlabel('Day of the month') # add to x-label to the plot
plt.ylabel('Number of rides') # add y-label to the plot
plt.title('Number of rides per rider type and day') # add title to the plot
plt.legend()
plt.show()
```



Results: Present your findings

- A ride lasts on average about 37 min (18 min for members and 60 min for casuals)
- The most popular days are Tuesday and Saturdays
- Sunday is the least day for rides numbers
- A casual ride last about 3,5 times than a member ride all week
- The docked Bike is still far more popular than the electric one
- Both riders rent bikes with same cadency (Max is on the first 10 days of the month) : This can be explained by the weather conditions relative to the month of the study (snow begins on the third week of April, 2020)

Act: conclusion (1)

- Main objective: Discover behavioral difference between members and casuals riders to gain the latters loyalty
- Members uses bikes either for work or school since there is no variation in the number of rides over the days contrary to tha casuals
- The pick of use for casuals is on the weekends (Saturdays)

Act: conclusion (2)

My three recommendations are:

1. Launch a specific weekend only membership with a lower price than the annual memberships
2. Launch campaigns on sunny months (Spring, summer) to avoid severe weather conditions such as Wind, Snow and rain to maximize interaction
3. As casual riders rent as they go, offering them sightseeing pragmas with the city monuments and sights could be a great type of casuals memberships.