PROJECT DESCRIPTION:

This project is about understanding the User Engagement patterns with the platform using Data Analytics so that the organization can make data-driven decisions. In this regard, we receive various questions by the management team and is asked to provide insights regarding the same. We will use SQL Queries to extract and filter data from the database provided to analyze and get insights.

APPROACH:

We had received a database containing multiple tables related to User engagement like User details, details about photos posted, details about likes, comments, followers and following, hashtags used, with each table related to other tables through Primary Key(s) and Foreign Key(s). We first tried to understand the whole schema of the database by drawing a schema diagram using MySQL software. Then we directly started filtering data as per the questions we had (if required, we would have processed the data like cleaning, transforming etc.). Filtering of data was done using pandas and pymysql libraries in python. We also plotted some graphs for better understanding. Then we directly drew conclusions and insights. Now, since we got all the insights required, we will make a detailed report about it and hand them over to the management team.

TECH-STACK USED:

1. MySQL Workbench 8.0.33 Community Version – A relational database management system used to create the database and its corresponding schema diagram. It's connections are used to connect the database with Python.
2. Python 3.10.9 – Programming language used to extract and filter data as per requirements using SQL Queries.
3. Jupyter Notebook 6.5.2 – Interactive platform to write and execute codes in various programming languages (in this case Python).

INSIGHTS:

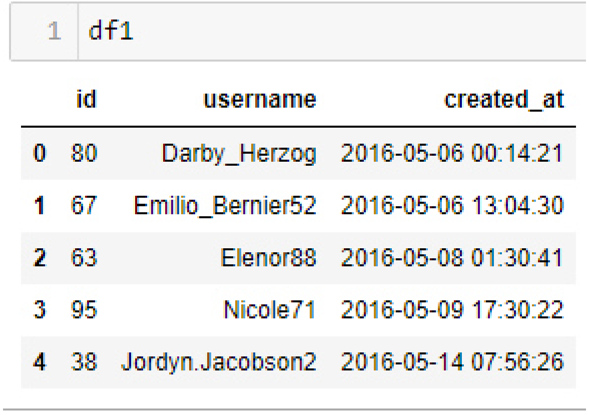
A) Marketing: The marketing team wants to launch some campaigns, and they need your help with the following

1. Rewarding Most Loyal Users: People who have been using the platform for the longest time.
Task: Find the 5 oldest users of the Instagram from the database provided:

Query:-

```
query='SELECT * FROM users ORDER BY created_at LIMIT 5'
df1 = pd.read_sql_query(query, conn)
```

Result:



```
df1
```

	id	username	created_at
0	80	Darby_Herzog	2016-05-06 00:14:21
1	67	Emilio_Bernier52	2016-05-06 13:04:30
2	63	Elenor88	2016-05-08 01:30:41
3	95	Nicole71	2016-05-09 17:30:22
4	38	Jordyn.Jacobson2	2016-05-14 07:56:26

Insights:-

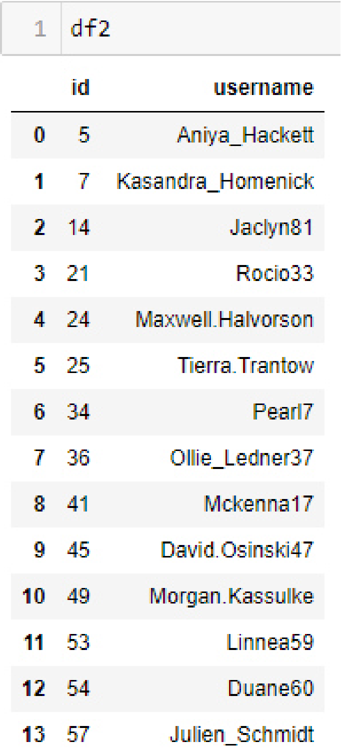
- The 5 oldest users in the current database we have are **Darby Herzog**, **Emilio_bernier52**, **Elenor88**, **Nicole71** and **Jordyn.Jacobson2**.
- We can see that the earliest members of Instagram registered their account in **May, 2016**.

2. Remind Inactive Users to Start Posting: By sending them promotional emails to post their 1st photo.
Task: Find the users who have never posted a single photo on Instagram:

Query:-

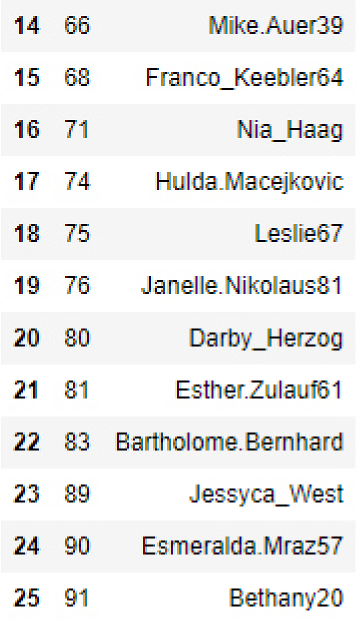
```
query='SELECT u.id, u.username FROM users u LEFT JOIN photos p ON u.id=p.user_id WHERE p.user_id IS NULL'
df2 = pd.read_sql_query(query, conn)
```

Result:



```
df2
```

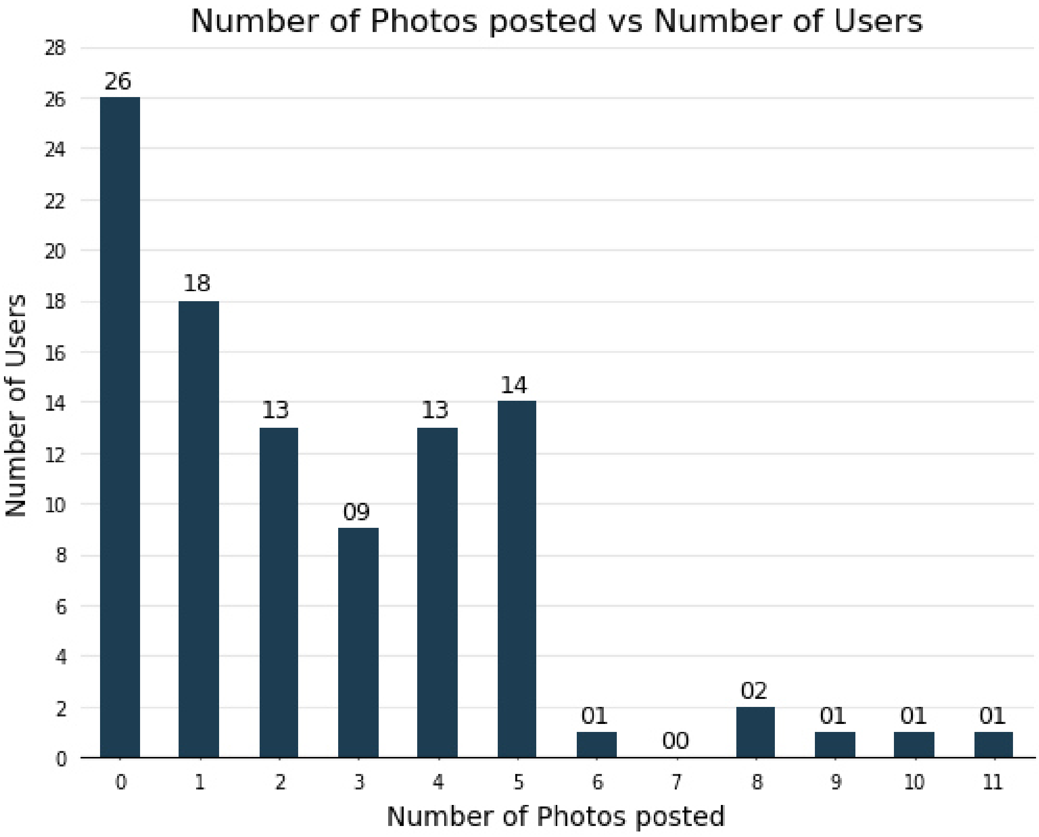
	id	username
0	5	Aniya_Hackett
1	7	Kasandra_Homenick
2	14	Jaclyn81
3	21	Rocio33
4	24	Maxwell.Halvorson
5	25	Tierra.Trantow
6	34	Pearl7
7	36	Ollie_Ledner37
8	41	Mckenna17
9	45	David.Osinski47
10	49	Morgan.Kassulke
11	53	Linnea59
12	54	Duane60
13	57	Julien_Schmidt



14	66	Mike.Auer39
15	68	Franco_Keebler64
16	71	Nia_Haag
17	74	Hulda.Macejkovic
18	75	Leslie67
19	76	Janelle.Nikolaus81
20	80	Darby_Herzog
21	81	Esther.Zulauf61
22	83	Bartholome.Bernhard
23	89	Jessyca_West
24	90	Esmeralda.Mraz57
25	91	Bethany20

Insights:-

- The above column **username** shows all the users who have never posted a single photo in Instagram till the time this dataset was recorded.
- There are a total of **26** such users which accounts to **26%** of the total users.
- The below plot shows how many users posting how many photos.



- From the above plot, we can observe that there are **26** number of users who have never posted in Instagram.
- Also, most users have posted **5 or less than 5** number of photos while very few users have posted **more than 5** number of photos.

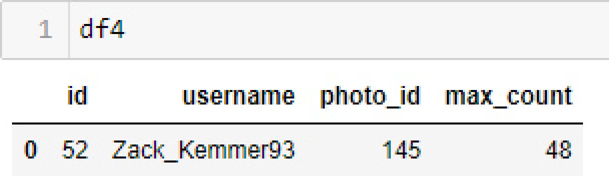
3. Declaring Contest Winner: The team started a contest and the user who gets the most likes on a single photo will win the contest now they wish to declare the winner.

Task: Identify the winner of the contest and provide their details to the team:

Query:-

```
query1='SELECT MAX(totals) FROM (SELECT photo_id AS p, COUNT(photo_id) AS totals FROM likes GROUP BY photo_id) AS r'
query2='SELECT photo_id, count(photo_id) AS c FROM likes GROUP BY photo_id HAVING COUNT(photo_id) = ({})'.format(query1)
query3='SELECT p.user_id AS pui, s.photo_id AS spi, s.c AS cnt FROM photos AS p join ({}) AS s ON p.id=s.photo_id'.format(query2)
query4='SELECT u.id, u.username, t.spi AS photo_id, t.cnt AS max_count FROM users AS u JOIN ({}) AS t ON u.id=t.pui'.format(query3)
df4 = pd.read_sql_query(query4, conn)
```

Result:

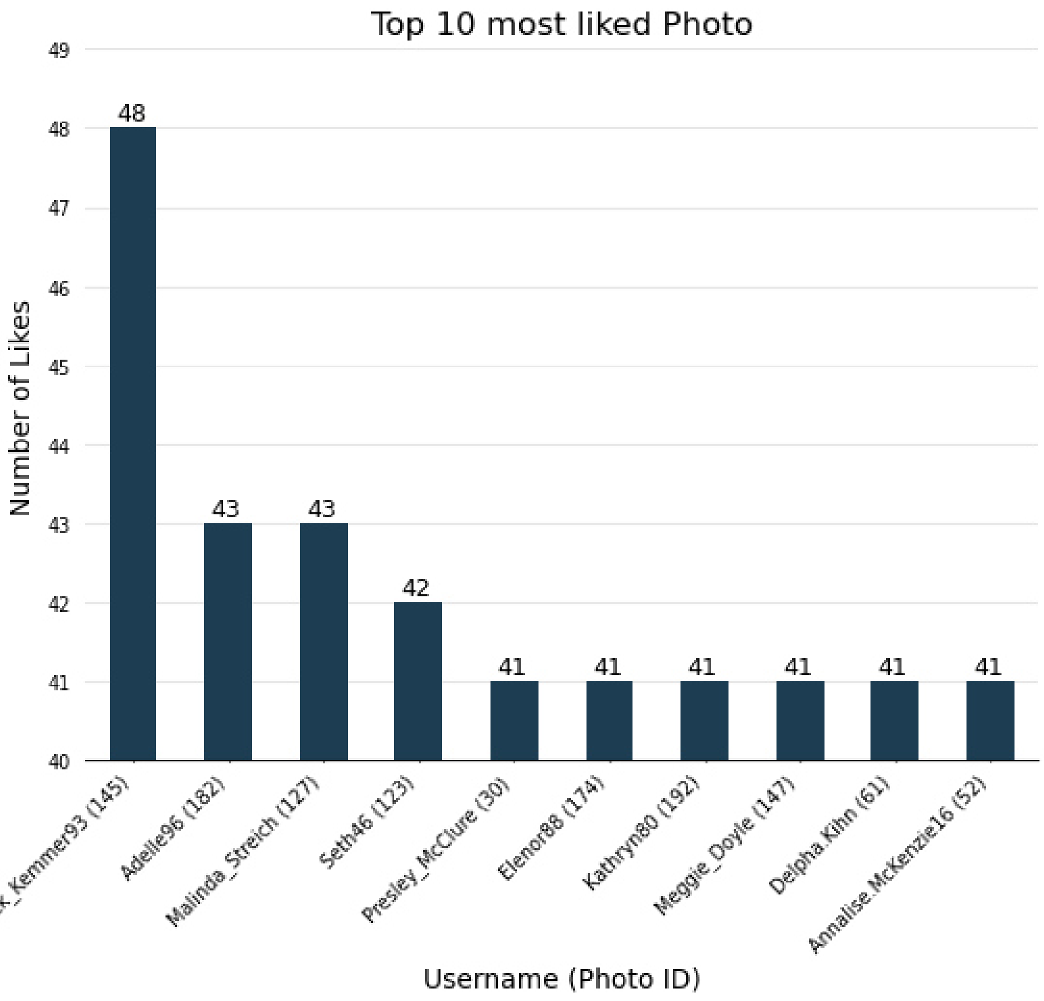


```
df4
```

	id	username	photo_id	max_count
0	52	Zack_Kemmer93	145	48

Insights:-

- The above dataframe shows that User with username **Zack_Kemmer93** has posted a Photo with Photo ID **145** has got the most likes **(48)**.
- The below plot shows the top 10 most liked photos.



- The above plot supports our finding in above observations that User with username **Zack_Kemmer93** who posted a photo with Photo ID **145** has got the most likes.
- The plot also shows that all the **top 10 most liked** Photos have more than **40** likes.

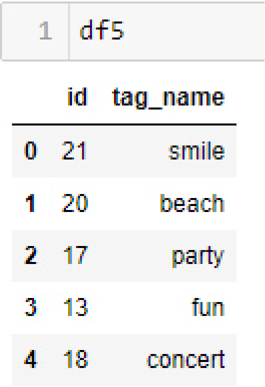
4. Hashtag Researching: A partner brand wants to know, which hashtags to use in the post to reach the most people on the platform.

Task: Identify and suggest the top 5 most commonly used hashtags on the platform.

Query:-

```
query1='SELECT tag_id, count(tag_id) AS cnt FROM photo_tags GROUP BY tag_id ORDER BY cnt DESC LIMIT 5'
query2='SELECT t.id, t.tag_name FROM tags AS t join({}) AS s ON t.id=s.tag_id'.format(query1)
df5 = pd.read_sql_query(query2, conn)
```

Result:

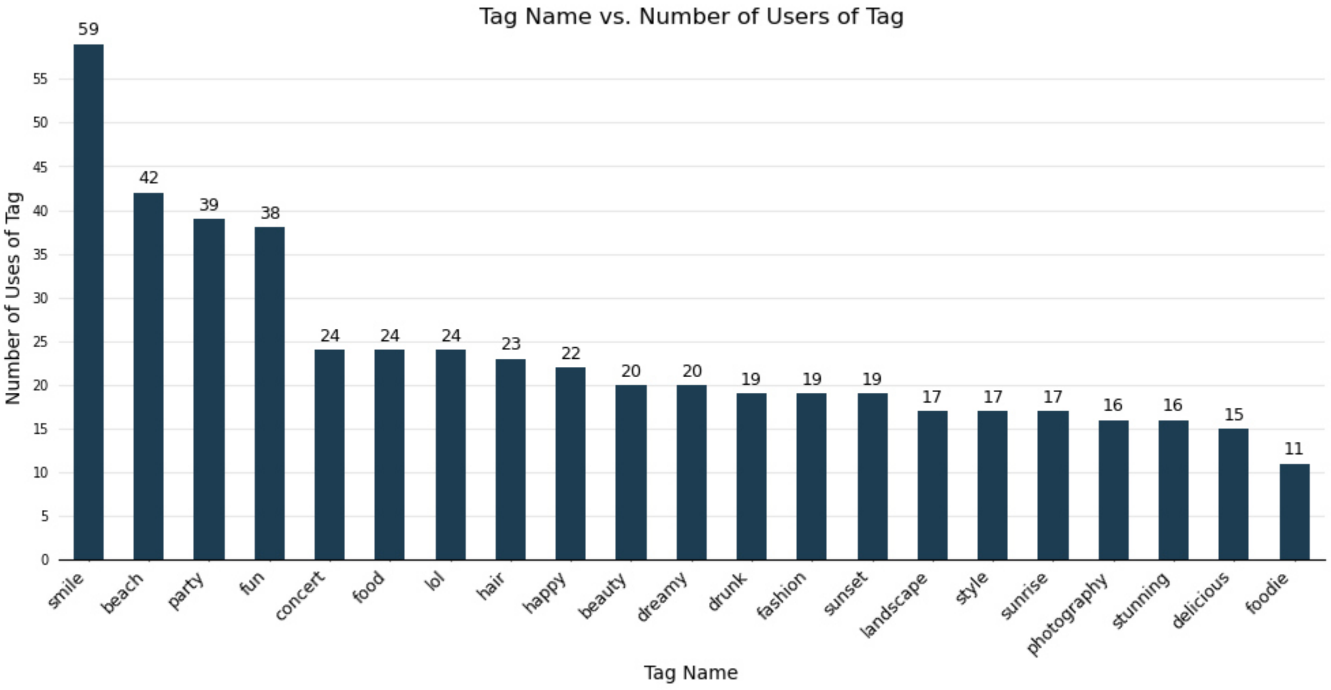


```
df5
```

	id	tag_name
0	21	smile
1	20	beach
2	17	party
3	13	fun
4	18	concert

Insights:-

- From the above dataframe, we can observe that most commonly used hashtags are **smile**, **beach**, **party**, **fun** and **concert**.
- The below plot shows how many users used which tag.



- The above plot supports our finding in above observations that most commonly used hashtags are **smile**, **beach**, **party**, **fun** and **concert**.
- The plot also shows that hashtag **smile** is used considerably more than the rest of the hashtags and hashtag **beach**, **party** and **fun** are used considerably more than the other less used hashtags.

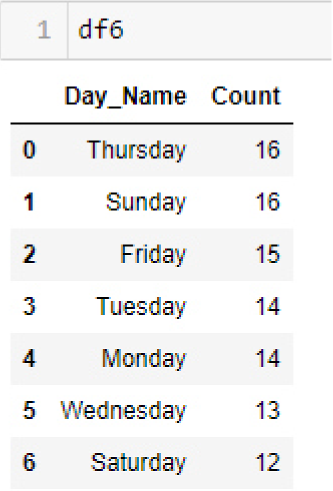
5. Launch AD Campaign: The team wants to know, which day would be the best day to launch ADs.

Task: What day of the week do most users register on? Provide insights on when to schedule an ad campaign.

Query:-

```
query1='SELECT DAYNAME(created_at) AS dn FROM users'
query2='SELECT dn AS Day_Name, COUNT(dn) AS Count FROM ({}) AS q1 GROUP BY q1.dn ORDER BY Count DESC'.format(query1)
df6 = pd.read_sql_query(query2, conn)
```

Result:

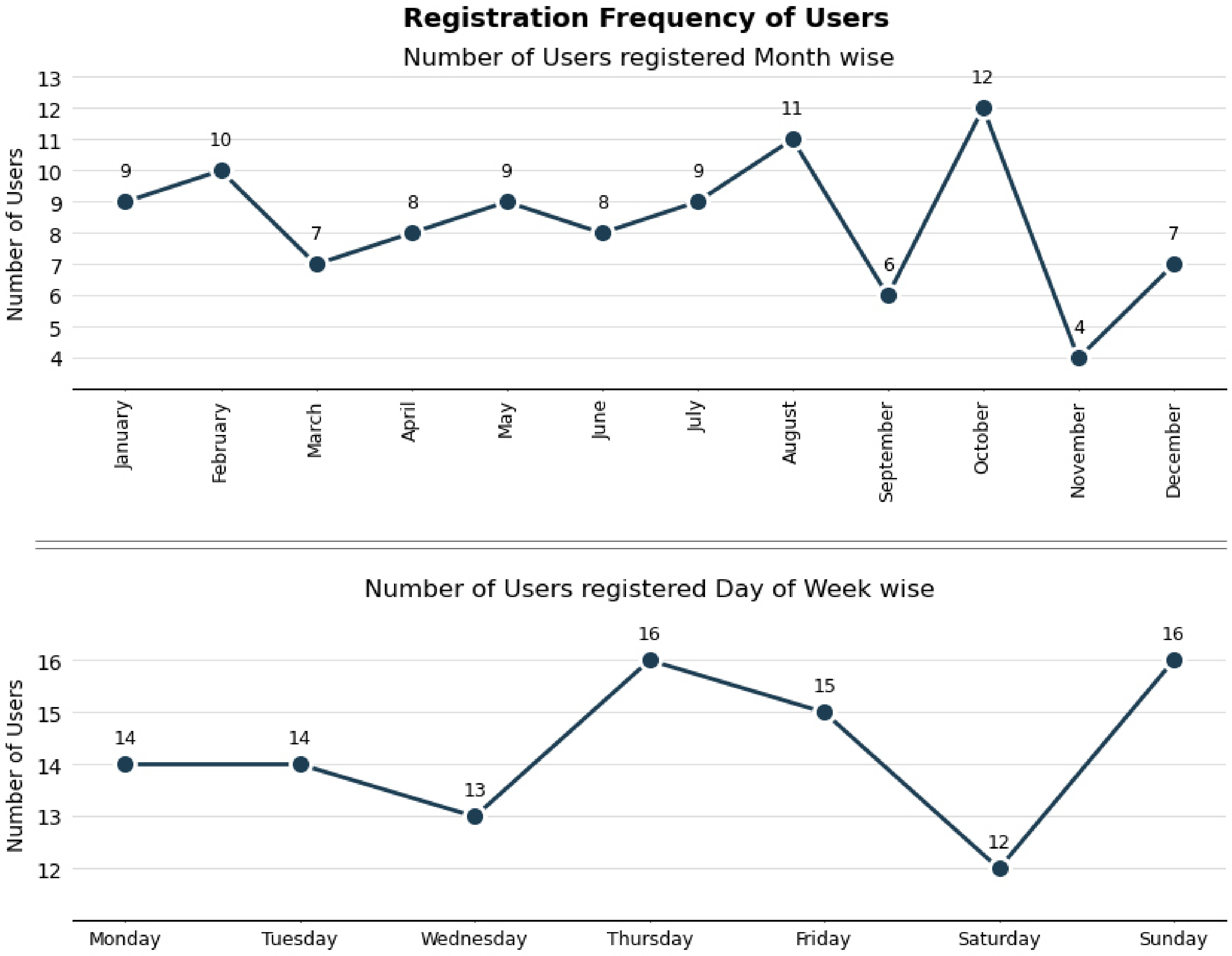


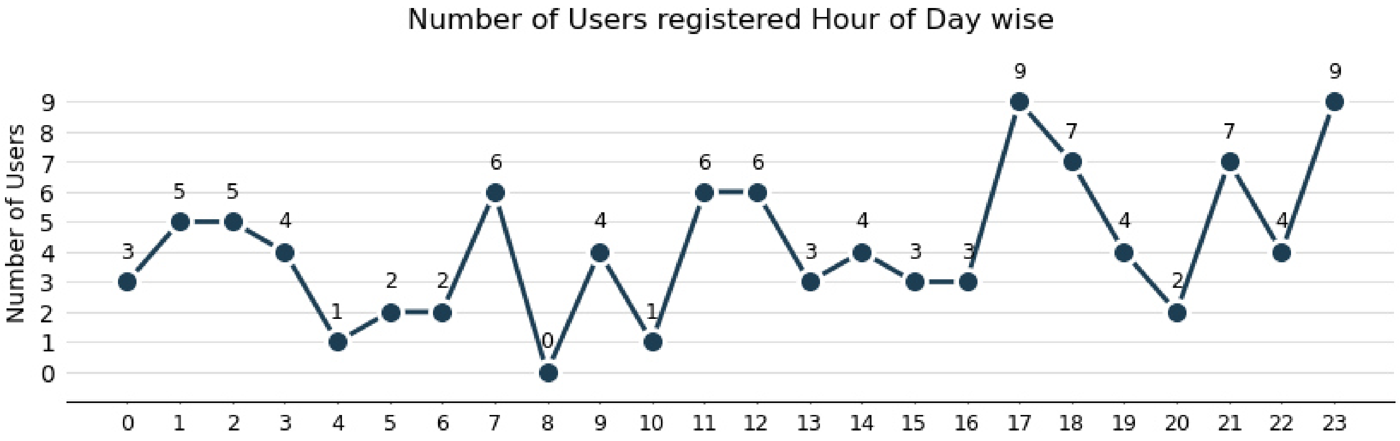
```
df6
```

	Day_Name	Count
0	Thursday	16
1	Sunday	16
2	Friday	15
3	Tuesday	14
4	Monday	14
5	Wednesday	13
6	Saturday	12

Insights:-

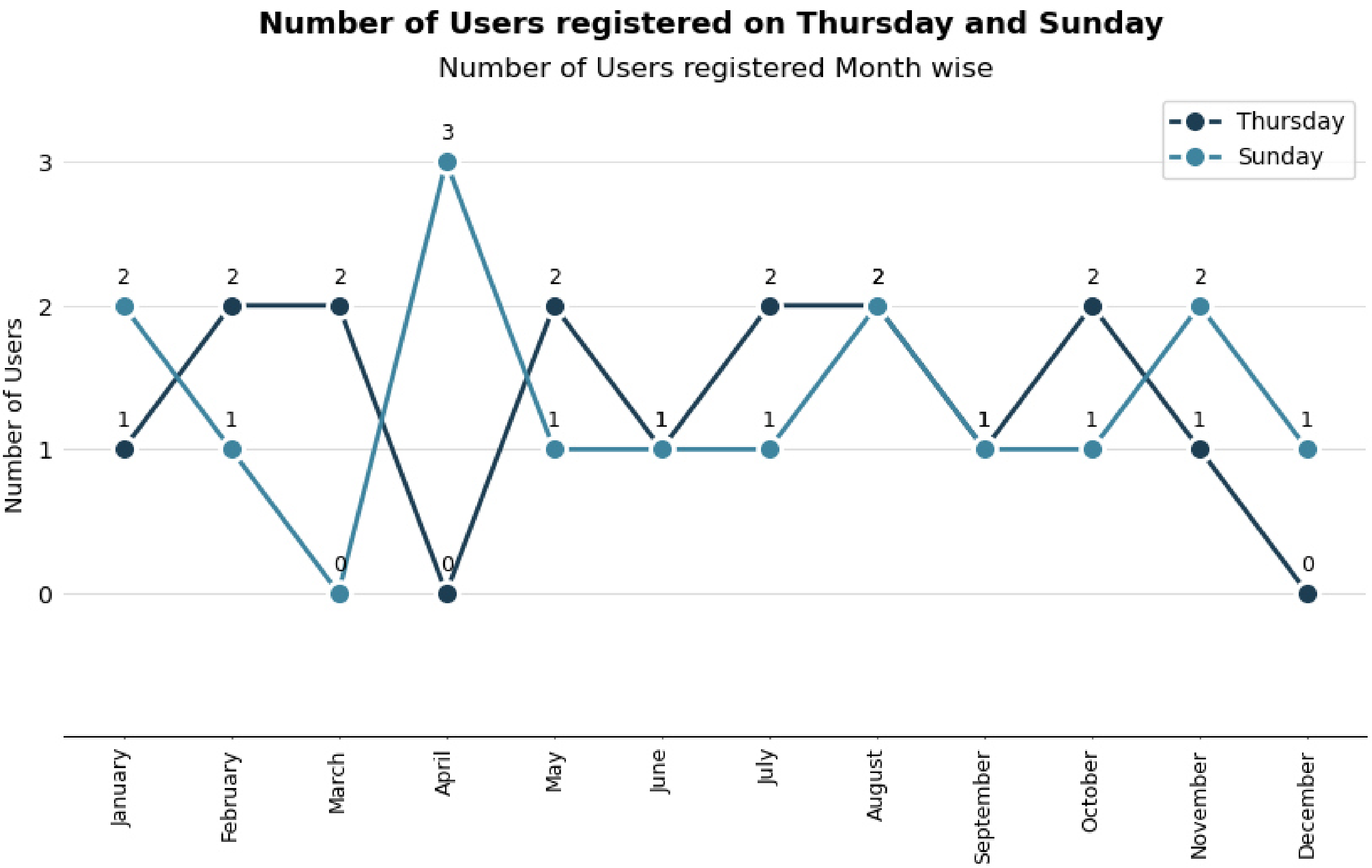
- From the above dataframe, we can observe that most users registered on **Thursday** and **Sunday**.
- The below plot shows number of users registered month wise, day of week wise and hour of day wise.





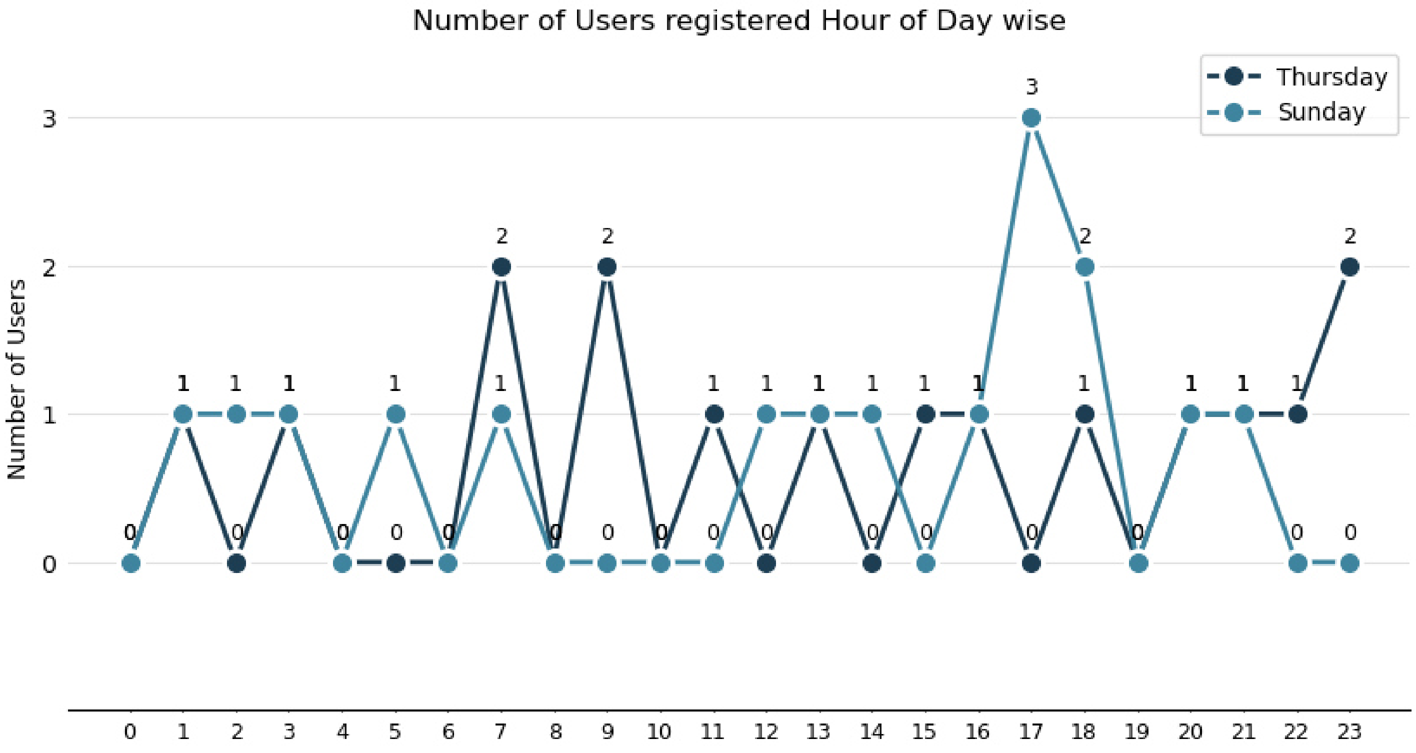
Number of Users registered Hour of Day wise

- If the Marketing team is looking at **Months**, then the best Month to schedule an Ad Campaign is the month of **October** followed by **August** and the **February**.
- If the Marketing team is looking at **Days of Week**, then the best Days of a Week to schedule an Ad Campaign are **Thursday** and **Sunday**.
- If the Marketing team is looking at **Hours of Day**, then the best Hours of a Day to schedule an Ad Campaign are **17th Hour (5 P.M)** and **23rd Hour (11 P.M)**. We didn't find any particular trend while observing Number of Users registered Hour of Day wise.
- The below plot shows number of users registered on Thursday and Sunday month wise and hour of day wise.



Number of Users registered on Thursday and Sunday

Number of Users registered Month wise



- For **Thursday**, we could not find any trend while observing "**Number of Users registered Month wise**" plot. We also could not find any trend while observing "**Number of Users registered Hour of Day wise**" plot.
- For **Sunday**, there was no particular trend but the month of **April** showed highest number of User registrations. Also, there was no particular trend but the **17th Hour(5 P.M)** showed highest number of User registrations.
- More data could have helped us in observing any important trends from the plots.
- Thus, from the limited data we have, we can say that the best Month to schedule an Ad Campaign are **Sundays** of **April** and the best Hours to schedule an Ad Campaign are **17th Hour(5 P.M)** of **Sundays**.

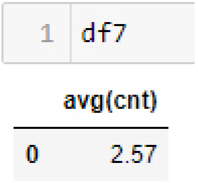
B) Investor Metrics: Our investors want to know if Instagram is performing well and is not becoming redundant like Facebook, they want to assess the app on the following grounds

1. User Engagement: Are users still as active and post on Instagram or they are making fewer posts**.**
Task: Provide how many times does average user posts on Instagram. Also, provide the total number of photos on Instagram/total number of users

Query:-

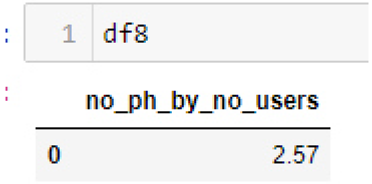
```
query1='SELECT u.id AS uid_in_ig, p.user_id AS uid_ph_pst FROM users u LEFT JOIN photos p ON u.id=p.user_id'
query2='SELECT count(q1.uid_ph_pst) AS cnt FROM ({}) AS q1 GROUP BY q1.uid_in_ig'.format(query1)
query3='SELECT avg(cnt) FROM ({}) AS q2'.format(query2)
df7 = pd.read_sql_query(query3, conn)

query4='SELECT COUNT(q1.uid_ph_pst)/COUNT(distinct q1.uid_in_ig) AS no_ph_by_no_users FROM ({}) AS
q1'.format(query1)
df8 = pd.read_sql_query(query4, conn)
```

Result:



```
df7
```

	avg(cnt)
0	2.57



```
df8
```

	no_ph_by_no_users
0	2.57

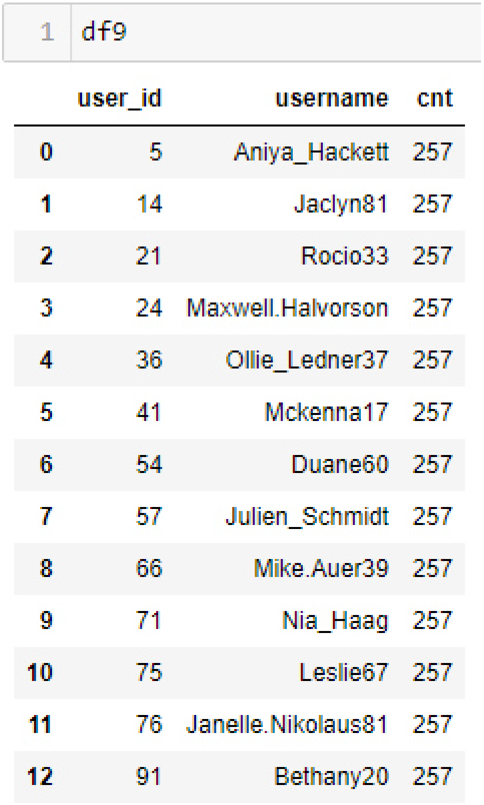
Insights:-

- Thus an **average user posts 2.57** times in Instagram which is the same as the **ratio between number of photos posted in Instagram and total number of users in Instagram**.
- An average user posting **2.57** times is a low number. The team needs to work towards more User engagement.

2. Bots & Fake Accounts: The investors want to know if the platform is crowded with fake and dummy accounts.
Task: Provide data on users (bots) who have liked every single photo on the site (since any normal user would not be able to do this).

Query:-
query1=**'SELECT user_id, COUNT(user_id) AS cnt FROM likes GROUP BY user_id'**
query2=**'SELECT q1.user_id, u.username, q1.cnt FROM ({}) AS q1 JOIN users AS u ON q1.user_id=u.id WHERE q1.cnt=ALL(SELECT COUNT(id) FROM photos)'**.format(query1)
df9 = pd.read_sql_query(query2, conn)

Result:



```
df9
```

	user_id	username	cnt
0	5	Aniya_Hackett	257
1	14	Jaclyn81	257
2	21	Rocio33	257
3	24	Maxwell.Halvorson	257
4	36	Ollie_Ledner37	257
5	41	Mckenna17	257
6	54	Duane60	257
7	57	Julien_Schmidt	257
8	66	Mike.Auer39	257
9	71	Nia_Haag	257
10	75	Leslie67	257
11	76	Janelle.Nikolaus81	257
12	91	Bethany20	257

Insights:-

- The above dataframe shows the list of Users who have liked all 257 photos in the database. These Users can be identified as **Bots**.
- There are **13** number of such Users which account for **13%** of total Users.

RESULT:

Through this project, I was able to understand how data analysis helps organizations make data-driven decisions. In this project based on data from Instagram, I was able to get insights about various questions like which users have been using the platform for the longest, which users are inactive in the platform, which hashtags can be used for ad/promotional contents for maximum reach, how many fake/bot accounts are present, whether the platform is growing or became stagnant in its growth. I can communicate these insights to the management team as per the requirements using which they can make proper decisions.

This Project has also helped me in understanding the basics of SQL and its working. Also, since I used python environment to run the SQL queries, I was also able to understand how to link the two platforms.