Instagram_Reach_Analysis

May 8, 2023

1 Problem Satement :)- Instagram Reach Analysis

1.1 Description:

- Instagram is one of the most popular social media applications today. People using Instagram professionally are using it for promoting their business, building a portfolio, blogging, and creating various kinds of content. As Instagram is a popular application used by millions of people with different niches, Instagram keeps changing to make itself better for the content creators and the users. But as this keeps changing, it affects the reach of our posts that affects us in the long run. So if a content creator wants to do well on Instagram in the long run, they have to look at the data of their Instagram reach. That is where the use of Data Science in social media comes in. If you want to learn how to use our Instagram data for the task of Instagram reach analysis, this article is for you. In this article, I will take you through Instagram Reach Analysis using Python, which will help content creators to understand how to adapt to the changes in Instagram in the long run.

2 1. Importing Libraries

[2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveRegressor
import warnings
warnings.filterwarnings('ignore')
```

3 2. The DataSets

4 2.1. Datasets Information

- Impressions: Number of impressions in a post (Reach)
- From Home: Reach from home

- From Hashtags: Reach from Hashtags
- From Explore: Reach from Explore
- From Other: Reach from other sources
- Saves: Number of saves
- Comments: Number of comments
- Shares: Number of shares
- Likes: Number of Likes
- Profile Visits: Numer of profile visits from the post
- Follows: Number of Follows from the post
- Caption: Caption of the post
- Hashtags: Hashtags used in the post

- Note: Here's the Instagram Data we collected from the account of the founder of Statso.
- DataSets Link (Click Me)

5 2.2. Reading Datsets

[3]:
```
#from google.colab import drive
#drive.mount('/content/drive')
```

[4]:
```
#df=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/DS_PROJECT/
    Instagram_Reach_Analysis/Instagram data.csv",encoding='cp1252')
df=pd.read_csv("Instagram_data.csv",encoding='cp1252')
df.head()
```

[4]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	\
0	3920	2586	1028	619	56	98	
1	5394	2727	1838	1174	78	194	
2	4021	2085	1188	0	533	41	
3	4528	2700	621	932	73	172	
4	2518	1704	255	279	37	96	

	Comments	Shares	Likes	Profile Visits	Follows	\
0	9	5	162	35	2	
1	7	14	224	48	10	
2	11	1	131	62	12	
3	10	7	213	23	8	
4	5	4	123	8	0	

	Caption	\
0	Here are some of the most important data visua...	
1	Here are some of the best data science project...	
2	Learn how to train a machine learning model an...	
3	Here's how you can write a Python program to d...	
4	Plotting annotations while visualizing your da...	

	Hashtags

```
0    #finance #money #business #investing #investme...
1    #healthcare #health #covid #data #datascience ...
2    #data #datascience #dataanalysis #dataanalytic...
3    #python #pythonprogramming #pythonprojects #py...
4    #datavisualization #datascience #data #dataana...
```

6 2.3. Data Exploration

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119 entries, 0 to 118
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Impressions     119 non-null    int64 
 1   From Home       119 non-null    int64 
 2   From Hashtags   119 non-null    int64 
 3   From Explore    119 non-null    int64 
 4   From Other      119 non-null    int64 
 5   Saves           119 non-null    int64 
 6   Comments        119 non-null    int64 
 7   Shares          119 non-null    int64 
 8   Likes           119 non-null    int64 
 9   Profile Visits  119 non-null    int64 
 10  Follows         119 non-null    int64 
 11  Caption         119 non-null    object
 12  Hashtags        119 non-null    object
dtypes: int64(11), object(2)
memory usage: 12.2+ KB
```

- Here All *Feature* is numeric but Caption and Hashtags is Object
- Let's Check how nemeric feature related to each others

[6]:
```
df.describe()
```

[6]:

	Impressions	From Home	From Hashtags	From Explore	From Other	\
count	119.000000	119.000000	119.000000	119.000000	119.000000	
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	

	Saves	Comments	Shares	Likes	Profile Visits	\

count	119.000000	119.000000	119.000000	119.000000	119.000000
mean	153.310924	6.663866	9.361345	173.781513	50.621849
std	156.317731	3.544576	10.089205	82.378947	87.088402
min	22.000000	0.000000	0.000000	72.000000	4.000000
25%	65.000000	4.000000	3.000000	121.500000	15.000000
50%	109.000000	6.000000	6.000000	151.000000	23.000000
75%	169.000000	8.000000	13.500000	204.000000	42.000000
max	1095.000000	19.000000	75.000000	549.000000	611.000000

	Follows
count	119.000000
mean	20.756303
std	40.921580
min	0.000000
25%	4.000000
50%	8.000000
75%	18.000000
max	260.000000

7 3. Handling Null Value

[7]:
```
df.isnull().sum()
```

[7]:
```
Impressions          0
From Home            0
From Hashtags        0
From Explore         0
From Other           0
Saves                0
Comments             0
Shares               0
Likes                0
Profile Visits       0
Follows              0
Caption              0
Hashtags             0
dtype: int64
```

- There is no null vlaue

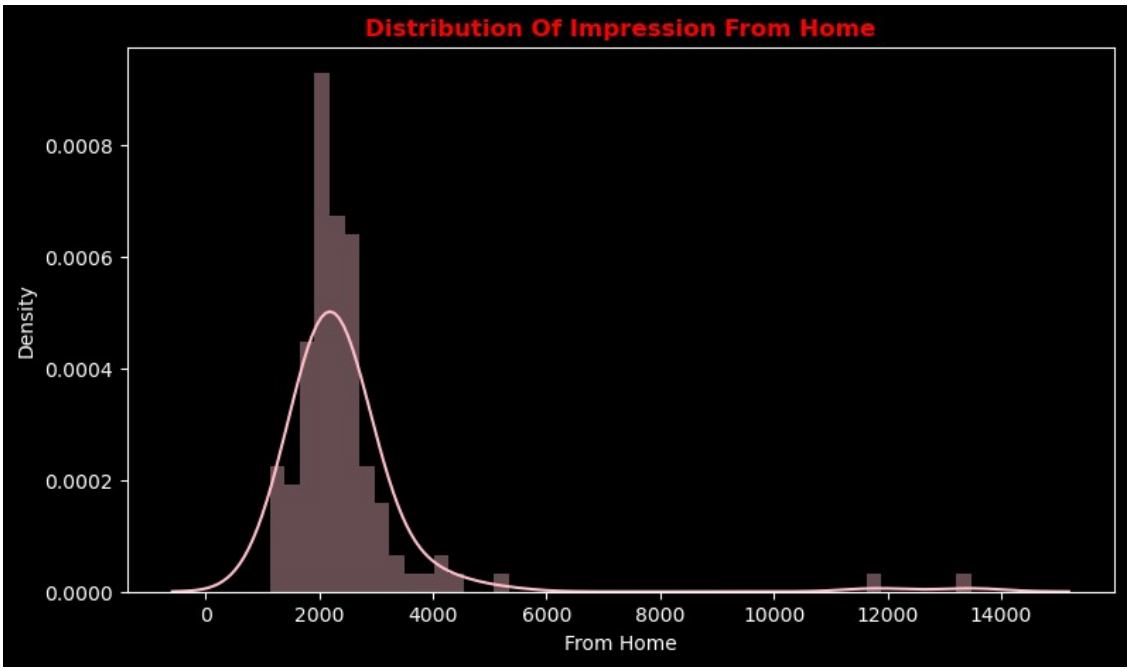
8 4. Data Visualization

[8]:
```
plt.style.use('dark_background')
plt.rcParams.update({'text.color':'white'})
```

[9]:
```
df.columns.unique()
```

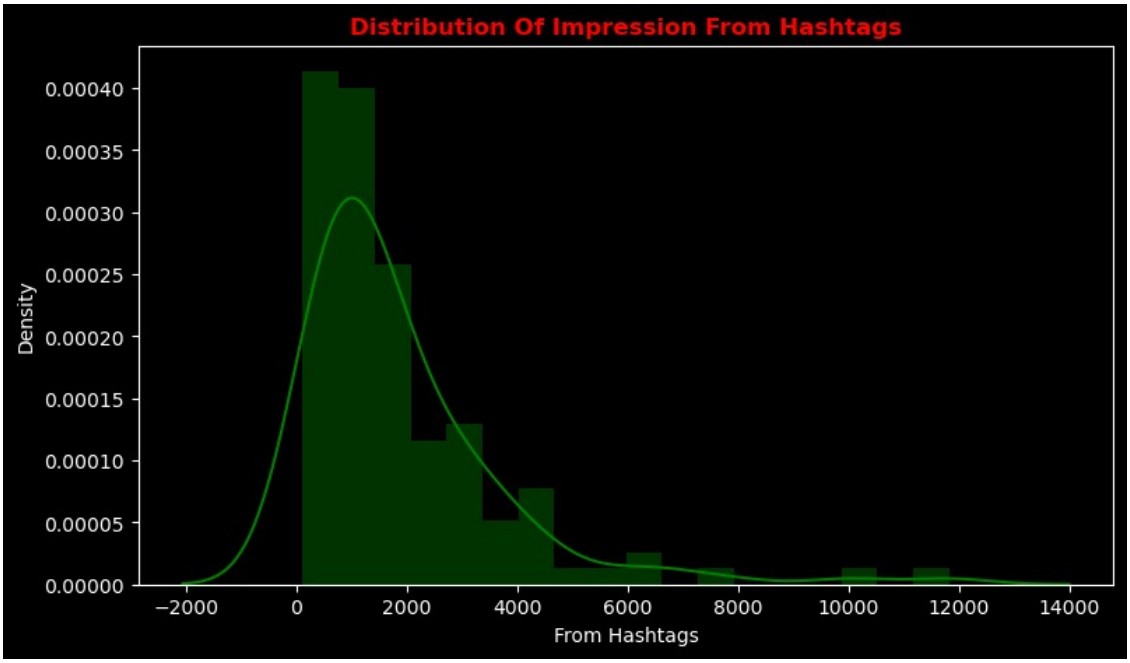
[9]:
```
Index(['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
       'Follows', 'Caption', 'Hashtags'],
      dtype='object')
```

[10]:
```
plt.figure(figsize=(9,5))
plt.title("Distribution Of Impression From Home",weight="bold",color='red')
sns.distplot(df['From Home'],color='pink')
plt.show()
```



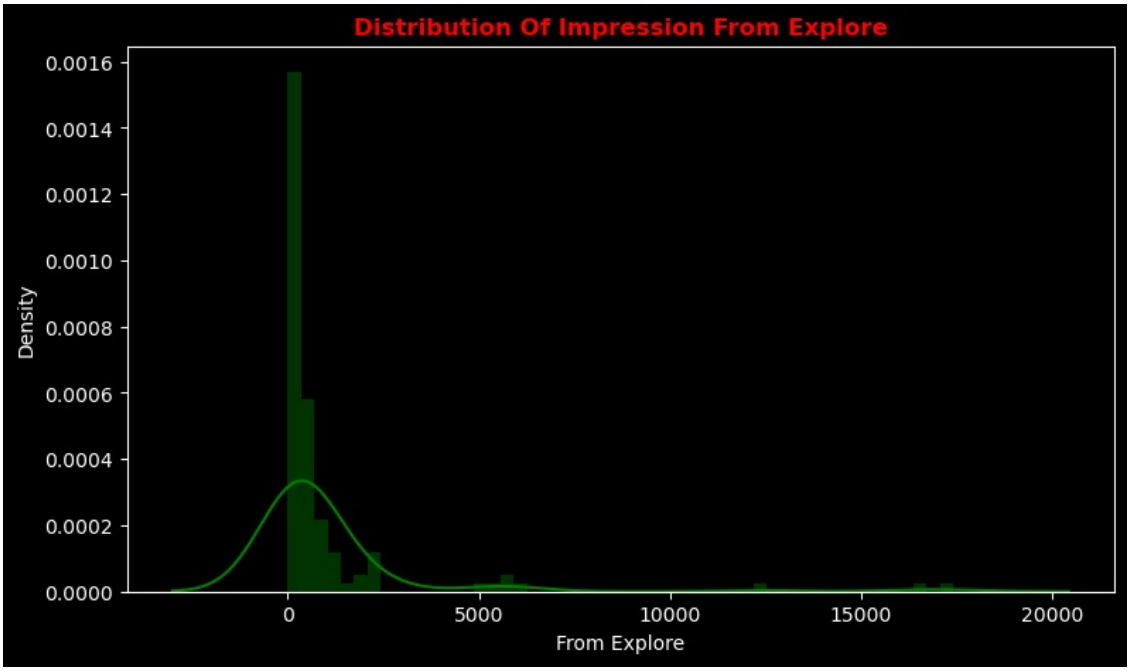
- The impressions I get from the home section on Instagram shows how much my posts reach my followers. Looking at the impressions from home, I can say it's hard to reach all my followers daily
- Now let's have a look at the distribution of the impressions I received from hashtags:

[11]:
```
plt.figure(figsize=(9,5))
plt.title("Distribution Of Impression From Hashtags",weight="bold",color='red')
sns.distplot(df['From Hashtags'],color='green')
plt.show()
```



- Hashtags are tools we use to categorize our posts on Instagram so that we can reach more people based on the kind of content we are creating. Looking at hashtag impressions shows that not all posts can be reached using hashtags, but many new users can be reached from hashtags.
- Now let's have a look at the distribution of impressions I have received from the explore section of Instagram:

[12]:
```
plt.figure(figsize=(9,5))
plt.title("Distribution Of Impression From Explore",weight="bold",color='red')
sns.distplot(df['From Explore'],color='green')
plt.show()
```



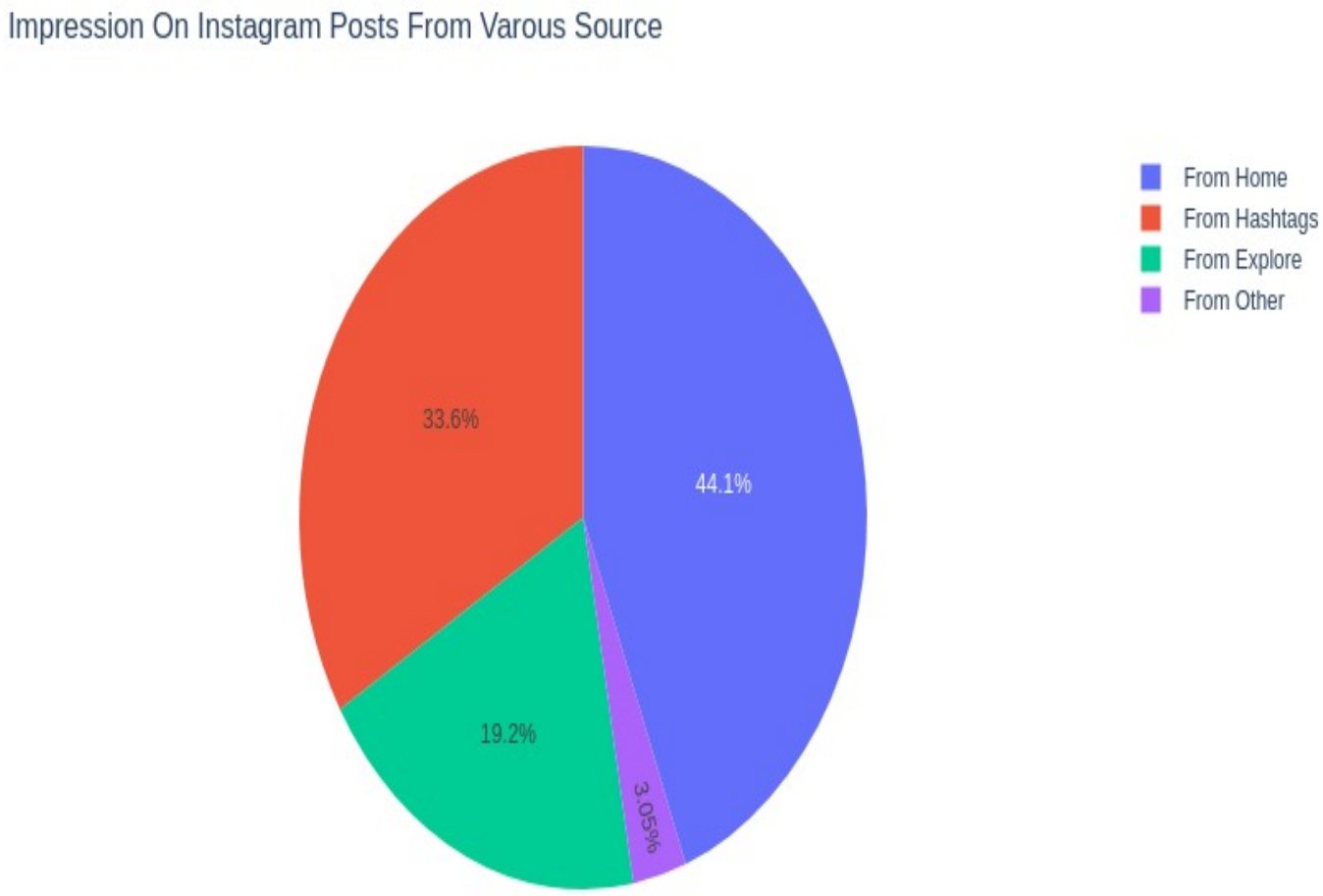
- The explore section of Instagram is the recommendation system of Instagram. It recommends posts to the users based on their preferences and interests. By looking at the impressions I have received from the explore section, I can say that Instagram does not recommend our posts much to the users. Some posts have received a good reach from the explore section, but it's still very low compared to the reach I receive from hashtags.
- Now let's have a look at the percentage of impressions I get from various sources on Instagram:

[13]:
```
plt.figure(figsize=(9,5))
home=df['From Home'].sum()
hashtags=df['From Hashtags'].sum()
explore=df['From Explore'].sum()
other=df['From Other'].sum()

labels=['From Home','From Hashtags','From Explore','From Other']
values=[home,hashtags,explore,other]

fig=px.pie(df,values=values,names=labels,
           title='Impression On Instagram Posts From Varous Source')
fig.show()
```

<Figure size 900x500 with 0 Axes>

- So the above donut plot shows that almost 50 per cent of the reach is from my followers, 38.1 per cent is from hashtags, 9.14 per cent is from the explore section, and 3.01 per cent is from other sources.

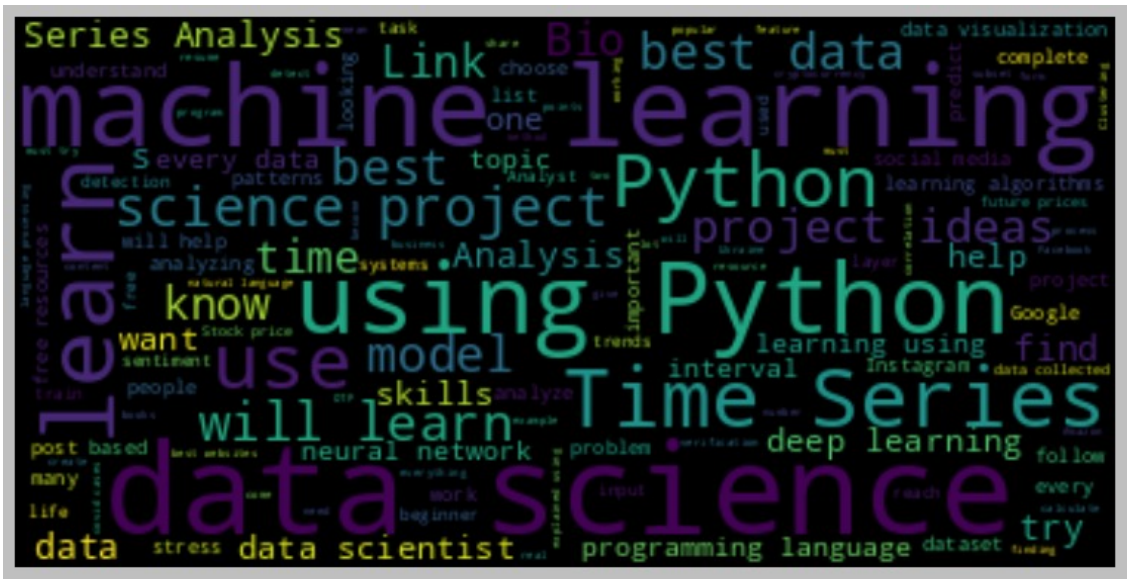
Relationship Between Home,Hashtags,Explore,Others



9 5. Analyzing Content

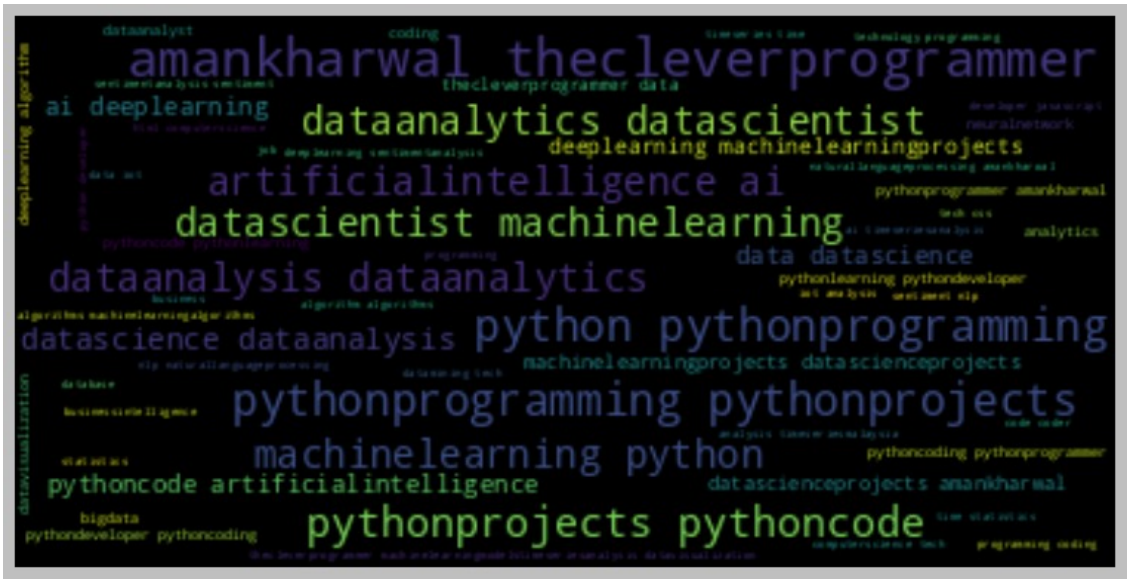
- Now let's analyze the content of my Instagram posts. The dataset has two columns, namely caption and hashtags, which will help us understand the kind of content I post on Instagram.
- Let's create a wordcloud of the caption column to look at the most used words in the caption of my Instagram posts:

[14]:
```
text=' '.join(i for i in df.Caption)
stopwords=set(STOPWORDS)
wordcloud=WordCloud(stopwords=stopwords).generate(text)
plt.style.use('classic')
plt.figure(figsize=(12,10))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis("off")
plt.show()
```



- Now let's create a wordcloud of the hashtags column to look at the most used hashtags in my Instagram posts:

[15]:
```
text=' '.join(i for i in df.Hashtags)
stopwords=set(STOPWORDS)
wordcloud=WordCloud(stopwords=stopwords).generate(text)
plt.style.use('classic')
plt.figure(figsize=(12,10))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis("off")
plt.show()
```



•

10 6. Analyzing Relationships

- Now let's analyze relationships to find the most important factors of our Instagram reach. It will also help us in understanding how the Instagram algorithm works.
- Let's have a look at the relationship between the number of likes and the number of impressions on my Instagram posts:

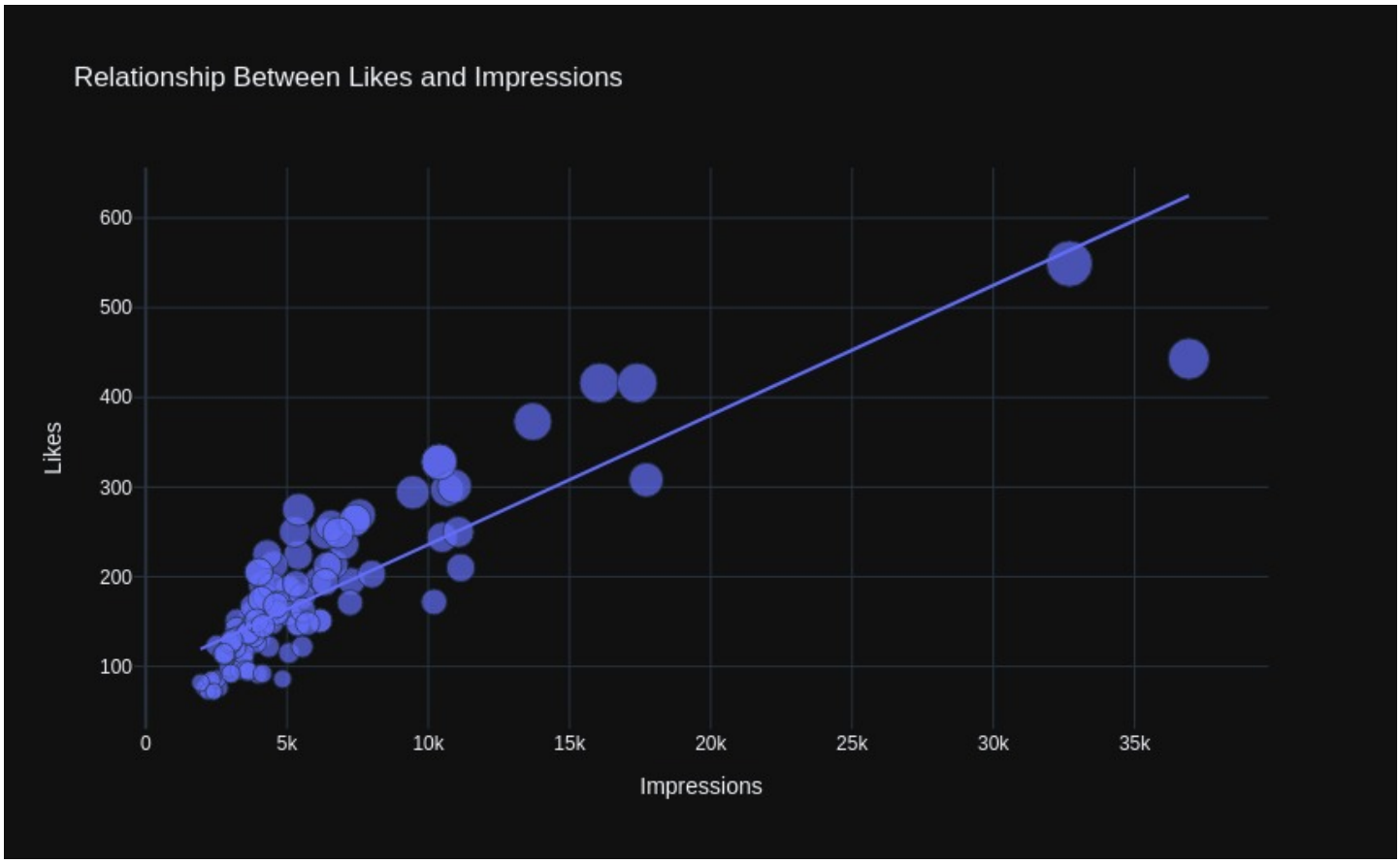
[16]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Likes", size="Likes", trendline="ols",
                    title = "Relationship Between Likes and␣
    Impressions",template="plotly_dark")
figure.show()
```

- There is a linear relationship between the number of likes and the reach I got on Instagram
- Now let's see the relationship between the number of comments and the number of impressions on my Instagram posts:

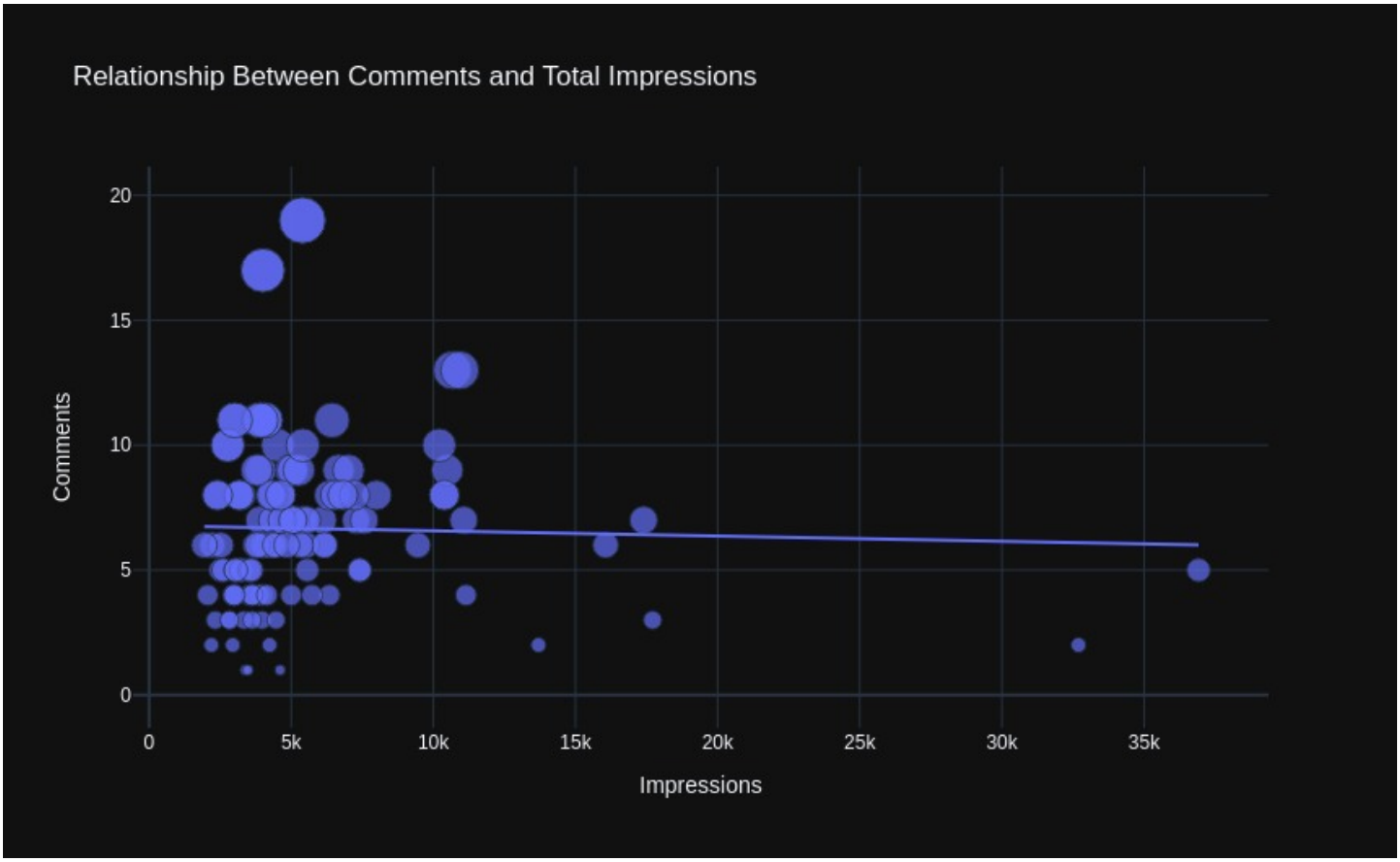
[17]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Comments", size="Comments", trendline="ols",
                    title = "Relationship Between Comments and Total␣
    Impressions",template="plotly_dark")
figure.show()
```

- It looks like the number of comments we get on a post doesn't affect its reach.

Relationship Between Likes and Impression



Relationship Between Comments and Impression



- let's have a look at the relationship between the number of shares and the number of impressions:

[18]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Shares", size="Shares", trendline="ols",
                    title = "Relationship Between Shares and Total␣
    Impressions",template="plotly_dark")
figure.show()
```

- A more number of shares will result in a higher reach, but shares don't affect the reach of a post as much as likes do.
- Now let's have a look at the relationship between the number of saves and the number of impressions:

[19]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Saves", size="Saves", trendline="ols",
                    title = "Relationship Between Post Saves and Total␣
    Impressions",template="plotly_dark")
figure.show()
```

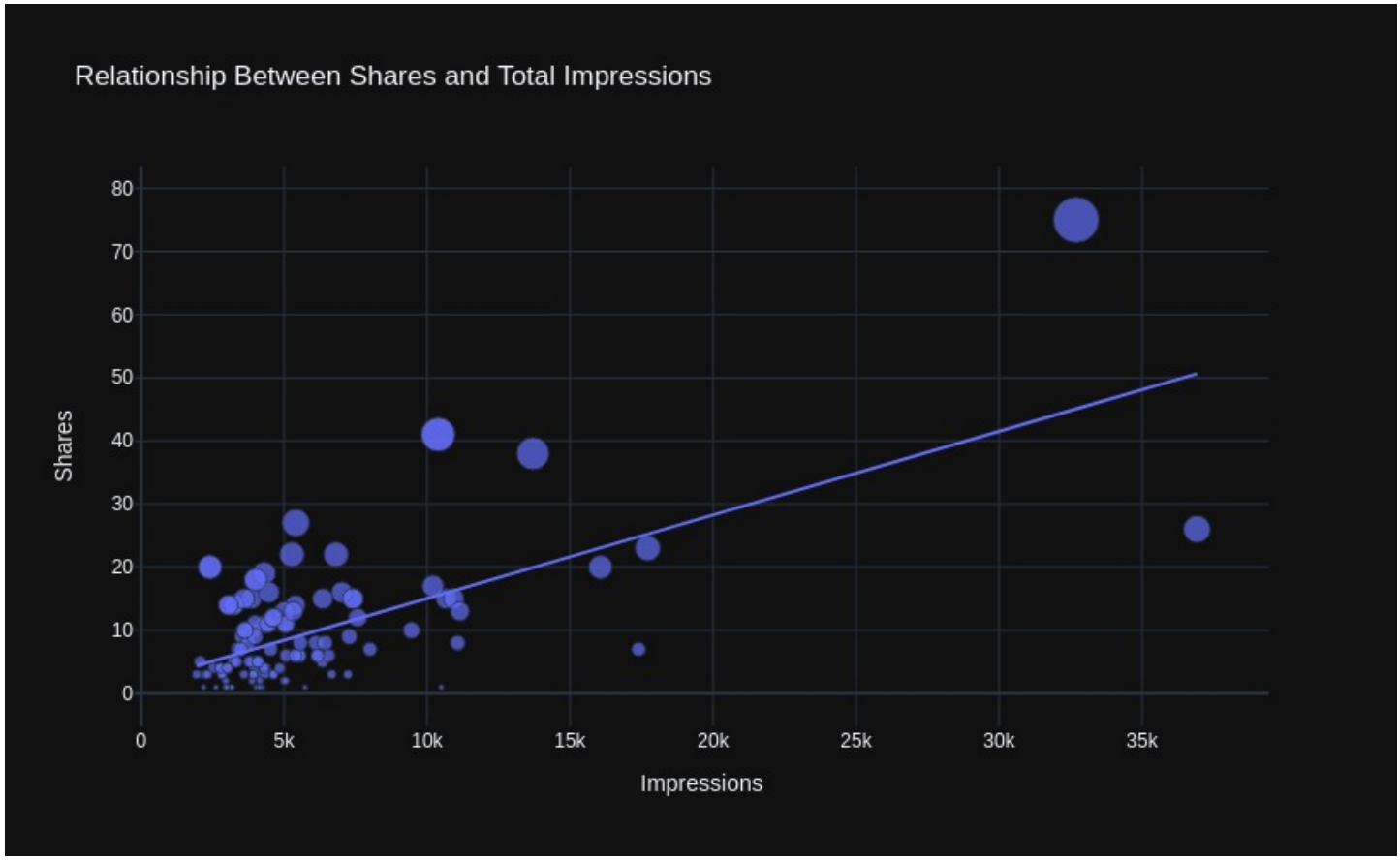
- There is a linear relationship between the number of times my post is saved and the reach of my Instagram post.
- Now let's have a look at the correlation of all the columns with the Impressions column:

[20]:
```
correlation = df.corr()
print(correlation["Impressions"].sort_values(ascending=False))
```

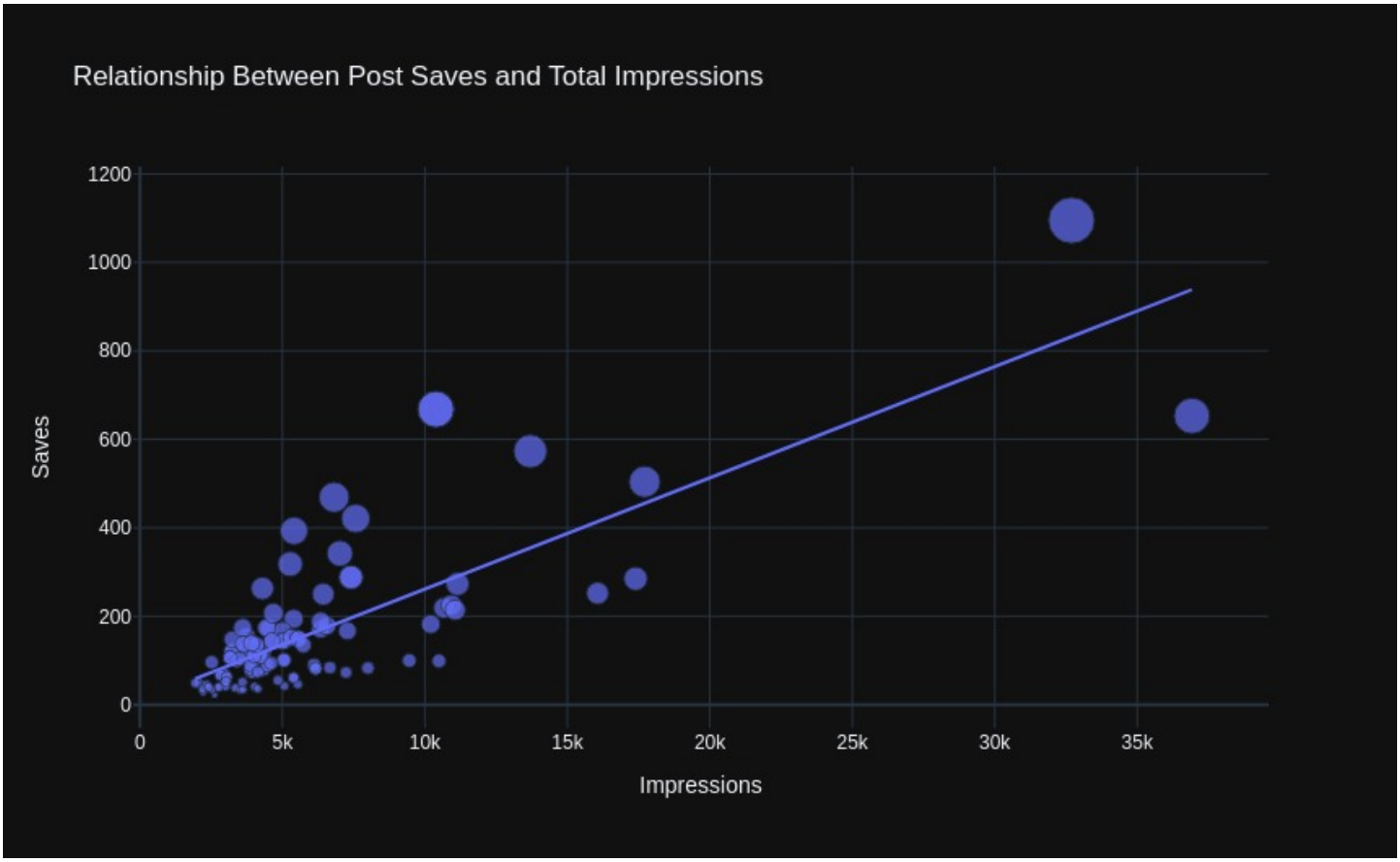
```
Impressions      1.000000
From Explore     0.893607
Follows          0.889363
Likes            0.849835
From Home        0.844698
Saves            0.779231
Profile Visits   0.760981
Shares           0.634675
From Other       0.592960
From Hashtags    0.560760
Comments        -0.028524
Name: Impressions, dtype: float64
```

- So we can say that more likes and saves will help you get more reach on Instagram. The higher number of shares will also help you get more reach, but a low number of shares will not affect your reach either.

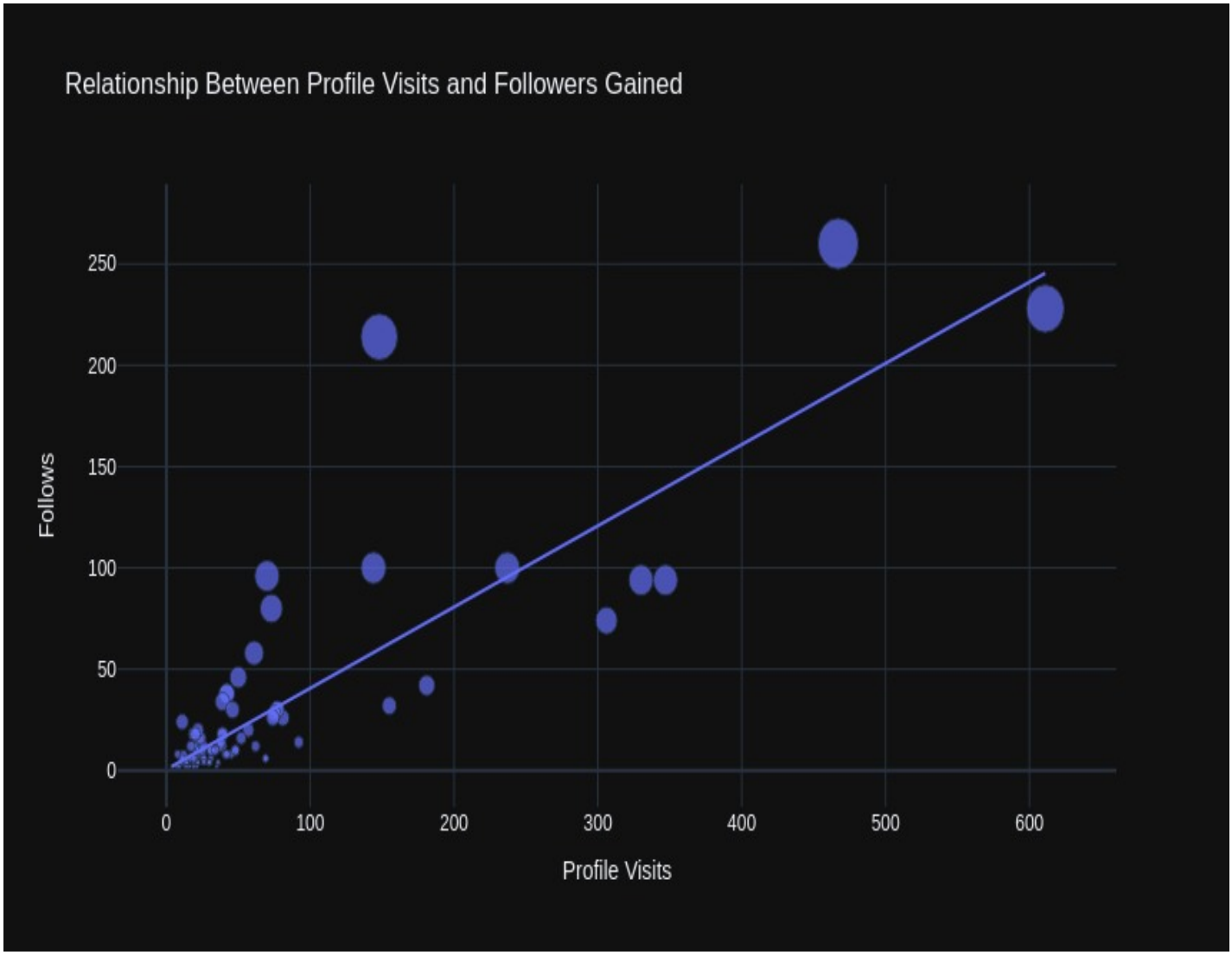
Relationship Between Shares and Total Impression



Relationship Between Post Saves and Total Impression



Relationship Between Profile Visits and Followers Gained



11 7. Analyzing Conversion Rate

In Instagram, conversation rate means how many followers you are getting from the number of profile visits from a post. The formula that you can use to calculate conversion rate is (Follows/Profile Visits) * 100. Now let's have a look at the conversation rate of my Instagram account:

[21]:
```
conversion_rate = (df["Follows"].sum() / df["Profile Visits"].sum()) * 100
print(conversion_rate)
```

41.00265604249668

- So the conversation rate of my Instagram account is 41% which sounds like a very good conversation rate.

[22]:
```
figure = px.scatter(data_frame = df, x="Profile Visits",
                    y="Follows", size="Follows", trendline="ols",
                    title = "Relationship Between Profile Visits and Followers␣
    Gained",template="plotly_dark")
figure.show()
```

- The relationship between profile visits and followers gained is also linear.

12 8. Model

- Now in this section, I will train a machine learning model to predict the reach of an Instagram post.
- Let's split the data into training and test sets before training the model:

[23]:
```
x = np.array(df[['Likes', 'Saves', 'Comments', 'Shares',
                 'Profile Visits', 'Follows']])
y = np.array(df["Impressions"])
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.2,
                                                    random_state=42)
```

- Now here's is how we can train a machine learning model to predict the reach of an Instagram post using Python:

[24]:
```
model = PassiveAggressiveRegressor()
model.fit(x_train, y_train)
model.score(x_test, y_test)
```

[24]: 0.7461051129678506

13 9. Testing

- Now let's predict the reach of an Instagram post by giving inputs to the machine learning model:

[25]:
```
# Features = [['Likes','Saves', 'Comments', 'Shares', 'Profile Visits',␣
    'Follows']]
features = np.array([[282.0, 233.0, 4.0, 9.0, 165.0, 54.0]])
model.predict(features)
```

[25]: array([8847.82112033])

14 Reference

- Aman Kahrwal (medium.com)
- Google

15 THANK YOU