Instagram stands as one of today's most prominent social media platforms, attracting users from various backgrounds who employ it for diverse purposes such as business promotion, portfolio building, blogging, and content creation. Given its widespread appeal across millions of users, Instagram consistently updates its features to cater to the needs of both content creators and consumers. However, these updates can impact post reach, influencing long-term success. Consequently, content creators must analyze their Instagram reach data to thrive on the platform. This underscores the significance of utilizing data science in social media analytics. For those interested in learning about Instagram reach analysis using Python, this article serves as a valuable resource, providing insights to help content creators adapt to Instagram's evolving landscape effectively.

Top of Form

## Instagram Reach Analysis

## "I've dedicated considerable time to researching Instagram reach, diligently collecting data on post performance after a week. This practice has provided valuable insights into the workings of Instagram's algorithm. Analyzing your Instagram account's reach entails manual data collection, as existing APIs often prove unreliable. Therefore, manual data collection remains the preferred method. If you're a data science student eager to delve into Instagram reach analysis using Python, you can utilize the dataset I've compiled from my Instagram account. You can download the dataset provided in the link below. In the following section, I'll guide you through Instagram Reach Analysis and Prediction with Machine Learning using Python."

## Instagram Reach Analysis using Python

Now let’s start the task of analyzing the reach of my Instagram account by importing the necessary Python libraries and the [**dataset**](https://statso.io/instagram-reach-analysis-case-study/):

## import pandas as pd

## import numpy as np

## import matplotlib.pyplot as plt

## import seaborn as sns

## import plotly.express as px

## from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

## from sklearn.model\_selection import train\_test\_split

## from sklearn.linear\_model import PassiveAggressiveRegressor

## data = pd.read\_csv("Instagram.csv", encoding = 'latin1')

## print(data.head())

**Impressions From Home From Hashtags From Explore From Other Saves \**

**0 3920.0 2586.0 1028.0 619.0 56.0 98.0**

**1 5394.0 2727.0 1838.0 1174.0 78.0 194.0**

**2 4021.0 2085.0 1188.0 0.0 533.0 41.0**

**3 4528.0 2700.0 621.0 932.0 73.0 172.0**

**4 2518.0 1704.0 255.0 279.0 37.0 96.0**

**Comments Shares Likes Profile Visits Follows \**

**0 9.0 5.0 162.0 35.0 2.0**

**1 7.0 14.0 224.0 48.0 10.0**

**2 11.0 1.0 131.0 62.0 12.0**

**3 10.0 7.0 213.0 23.0 8.0**

**4 5.0 4.0 123.0 8.0 0.0**

**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...**

Before starting everything, let’s have a look at whether this dataset contains any null values or not:

data.isnull().sum()

**Impressions 1**

**From Home 1**

**From Hashtags 1**

**From Explore 1**

**From Other 1**

**Saves 1**

**Comments 1**

**Shares 1**

**Likes 1**

**Profile Visits 1**

**Follows 1**

**Caption 1**

**Hashtags 1**

**dtype: int64**

So it has a null value in every column. Let’s drop all these null values and move further:

data = data.dropna()

Let’s have a look at the insights of the columns to understand the data type of all the columns:

data.info()

**<class 'pandas.core.frame.DataFrame'>**

**Int64Index: 99 entries, 0 to 98**

**Data columns (total 13 columns):**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

**0 Impressions 99 non-null float64**

**1 From Home 99 non-null float64**

**2 From Hashtags 99 non-null float64**

**3 From Explore 99 non-null float64**

**4 From Other 99 non-null float64**

**5 Saves 99 non-null float64**

**6 Comments 99 non-null float64**

**7 Shares 99 non-null float64**

**8 Likes 99 non-null float64**

**9 Profile Visits 99 non-null float64**

**10 Follows 99 non-null float64**

**11 Caption 99 non-null object**

**12 Hashtags 99 non-null object**

**dtypes: float64(11), object(2)**

**memory usage: 10.8+ KB**

#### Analyzing Instagram Reach

Let's initiate the analysis by examining the distribution of impressions generated by my Instagram posts. Initially, we'll focus on evaluating the impressions accumulated from the home feed.

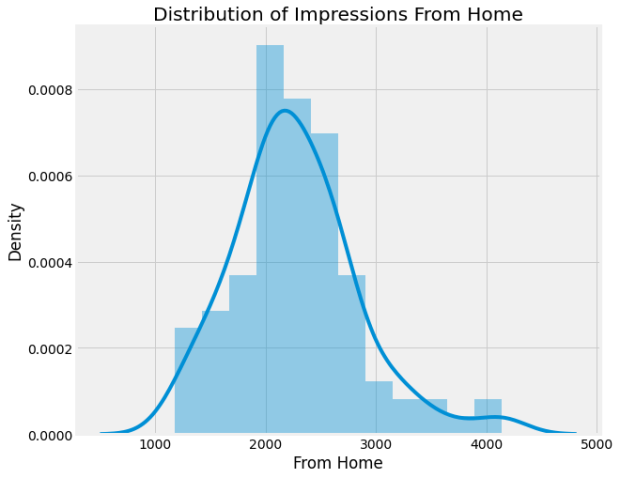
plt.figure(figsize=(10, 8))

plt.style.use('fivethirtyeight')

plt.title("Distribution of Impressions From Home")

sns.distplot(data['From Home'])

plt.show()



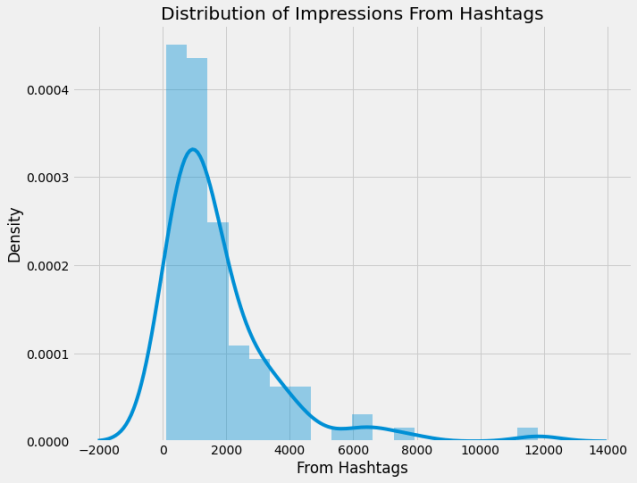
The impressions I receive from the home section on Instagram reflect the extent to which my posts reach my followers. Analyzing impressions from the home feed reveals the challenge of reaching all my followers on a daily basis. Now, let's shift our focus to examining the distribution of impressions garnered from hashtags.

plt.figure(figsize=(10, 8))

plt.title("Distribution of Impressions From Hashtags")

sns.distplot(data['From Hashtags'])

plt.show()



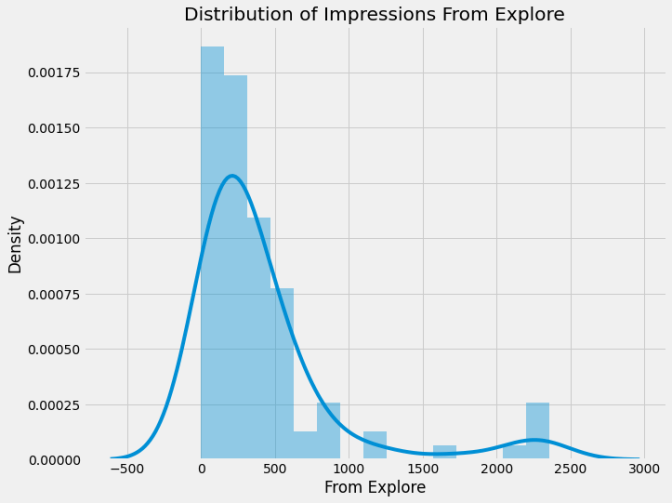
Hashtags are tools we use to categorize our posts on Instagram so that we can reach more people based on the kind of content we are creating. Looking at hashtag impressions shows that not all posts can be reached using hashtags, but many new users can be reached from hashtags. Now let’s have a look at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

plt.figure(figsize=(10, 8))

plt.title("Distribution of Impressions From Explore")

sns.distplot(data['From Explore'])

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:

home = data["From Home"].sum()

hashtags = data["From Hashtags"].sum()

explore = data["From Explore"].sum()

other = data["From Other"].sum()

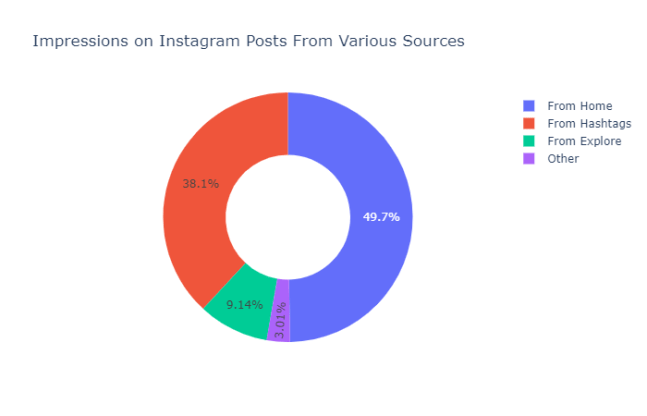
labels = ['From Home','From Hashtags','From Explore','Other']

values = [home, hashtags, explore, other]

fig = px.pie(data, values=values, names=labels,

title='Impressions on Instagram Posts From Various Sources', hole=0.5)

fig.show()



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.

#### 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:

text = " ".join(i for i in data.Caption)

stopwords = set(STOPWORDS)

wordcloud = WordCloud(stopwords=stopwords, background\_color="white").generate(text)

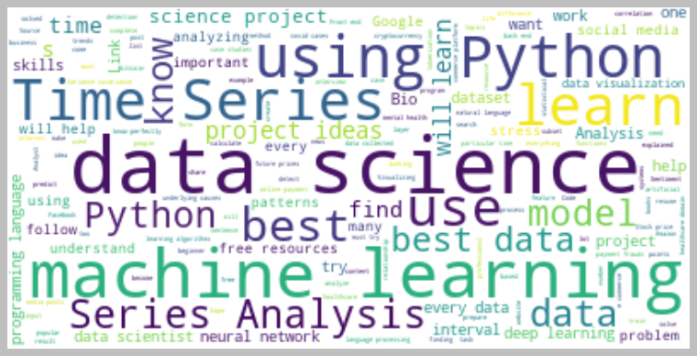
plt.style.use('classic')

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:

text = " ".join(i for i in data.Hashtags)

stopwords = set(STOPWORDS)

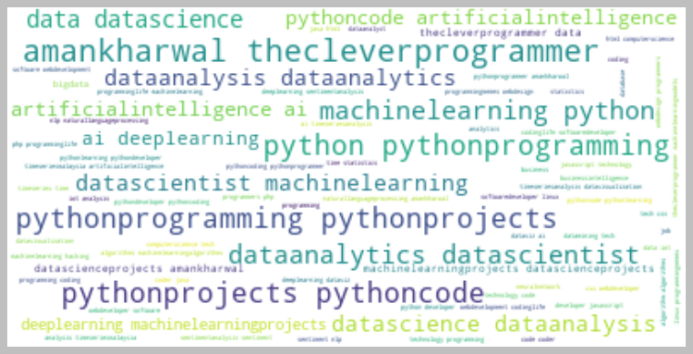
wordcloud = WordCloud(stopwords=stopwords, background\_color="white").generate(text)

plt.figure( figsize=(12,10))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis("off")

plt.show()



#### 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.

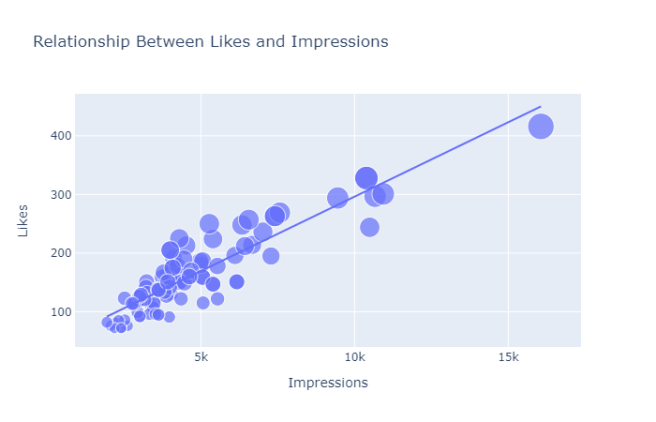
Let’s have a look at the relationship between the number of likes and the number of impressions on my Instagram posts:

figure = px.scatter(data\_frame = data, x="Impressions",

y="Likes", size="Likes", trendline="ols",

title = "Relationship Between Likes and Impressions")

figure.show()



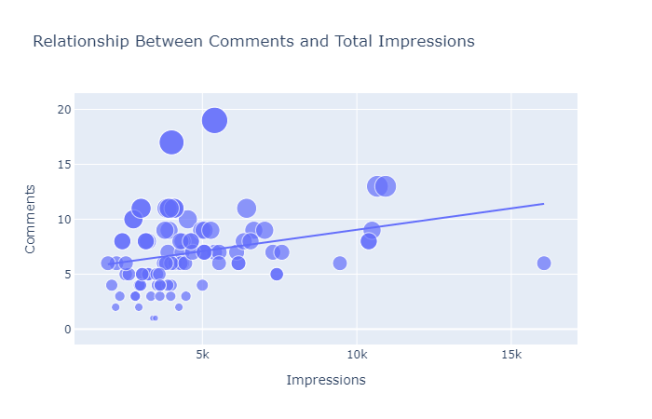
There is a linear relationship between the number of likes and the reach I got on Instagram. Now let’s see the relationship between the number of comments and the number of impressions on my Instagram posts:

figure = px.scatter(data\_frame = data, x="Impressions",

y="Comments", size="Comments", trendline="ols",

title = "Relationship Between Comments and Total Impressions")

figure.show()



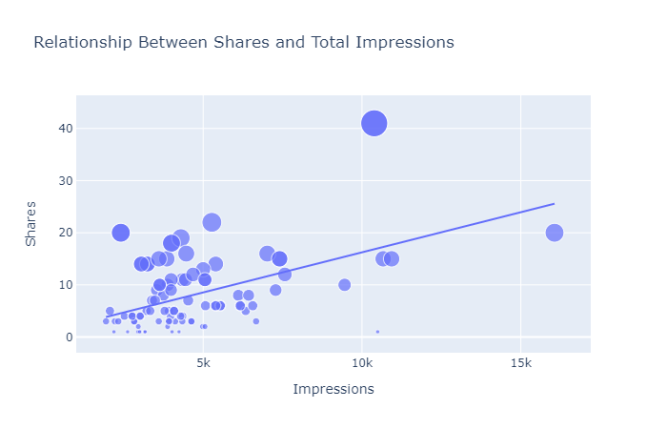
It looks like the number of comments we get on a post doesn’t affect its reach. Now let’s have a look at the relationship between the number of shares and the number of impressions:

figure = px.scatter(data\_frame = data, x="Impressions",

y="Shares", size="Shares", trendline="ols",

title = "Relationship Between Shares and Total Impressions")

figure.show()



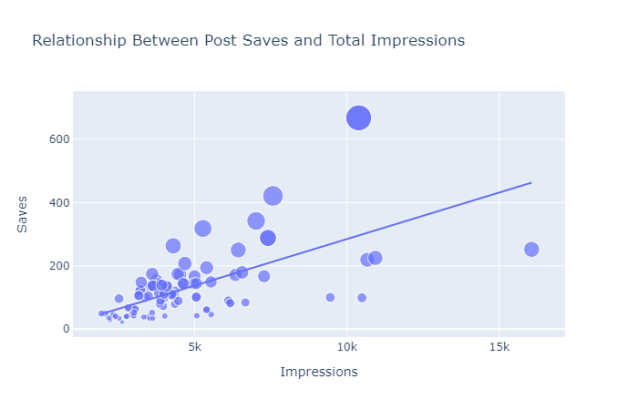
A more number of shares will result in a higher reach, but shares don’t affect the reach of a post as much as likes do. Now let’s have a look at the relationship between the number of saves and the number of impressions:

figure = px.scatter(data\_frame = data, x="Impressions",

y="Saves", size="Saves", trendline="ols",

title = "Relationship Between Post Saves and Total Impressions")

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:

correlation = data.corr()

print(correlation["Impressions"].sort\_values(ascending=False))

**Impressions 1.000000**

**Likes 0.896277**

**From Hashtags 0.892682**

**Follows 0.804064**

**Profile Visits 0.774393**

**Saves 0.625600**

**From Home 0.603378**

**From Explore 0.498389**

**Shares 0.476617**

**From Other 0.429227**

**Comments 0.247201**

**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.

#### 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:

conversion\_rate = (data["Follows"].sum() / data["Profile Visits"].sum()) \* 100

print(conversion\_rate)

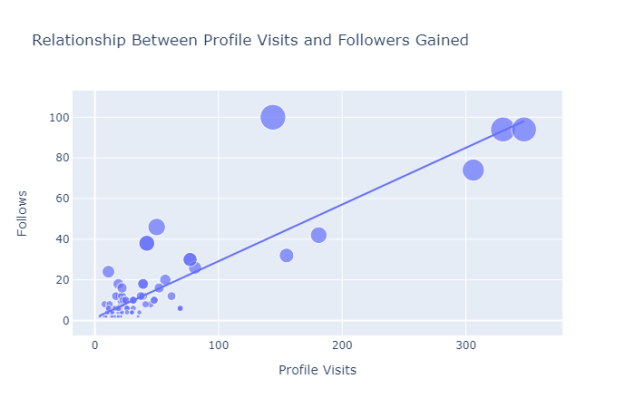
So the conversation rate of my Instagram account is 31% which sounds like a very good conversation rate. Let’s have a look at the relationship between the total profile visits and the number of followers gained from all profile visits:

figure = px.scatter(data\_frame = data, x="Profile Visits",

y="Follows", size="Follows", trendline="ols",

title = "Relationship Between Profile Visits and Followers Gained")

figure.show()



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

### Instagram Reach Prediction 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:

x = np.array(data[['Likes', 'Saves', 'Comments', 'Shares',

'Profile Visits', 'Follows']])

y = np.array(data["Impressions"])

xtrain, xtest, ytrain, ytest = 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:

model = PassiveAggressiveRegressor()

model.fit(xtrain, ytrain)

model.score(xtest, ytest)

**0.9428392959517574**

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

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)

**array([10319.5922441])**

### Summary

So this is how you can analyze and predict the reach of Instagram posts with machine learning using Python. If a content creator wants to do well on Instagram in a 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. I hope you liked this article on the task of Instagram Reach Analysis using Python. Feel free to ask valuable questions in the comments section below.