

Instagram Reach Analysis using Python

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.

INSTAGRAM REACH ANALYSIS

I have been researching Instagram reach for a long time now. Every time I post on my [Instagram account](#), I collect data on how well the post reach after a week. That helps in understanding how Instagram's algorithm is working. If you want to analyze the reach of your Instagram account, you have to collect your data manually as there are some APIs, but they don't work well. So it's better to collect your Instagram data manually.

If you are a data science student and want to learn Instagram reach analysis using Python, you can use the data I have collected from my Instagram account. You can download the dataset I have used for the task of Instagram reach analysis from [here](#). Now in the section below, I will take you through the task of Instagram Reach Analysis and Prediction with Machine Learning using Python.

Reach Analysis Python

Now let's start the task of analyzing the reach of my Instagram account by importing the necessary Python libraries and the [dataset](#):

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import plotly.express as px
6 from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
7 from sklearn.model_selection import train_test_split
8 from sklearn.linear_model import PassiveAggressiveRe
9
10 data
11 print(data.head())
12         = pd.read_csv("Instagram.csv", encoding = latin1)
13
14 data.info()
15
16 data["Reach"] = data["Reach"] / 1000000
17
18 data["Reach"]
19
20 data["Reach"] = data["Reach"].apply(lambda x: int(x))
21
22 data["Reach"]
23
24 data["Reach"] = data["Reach"].apply(lambda x: str(x))
25
26 data["Reach"]
27
28 data["Reach"] = data["Reach"].str.replace(",","")
29
30 data["Reach"]
31
32 data["Reach"] = data["Reach"].str.replace(".","")
33
34 data["Reach"]
35
36 data["Reach"] = data["Reach"].str.replace(",","")
37
38 data["Reach"]
39
40 data["Reach"] = data["Reach"].str.replace(".","")
41
42 data["Reach"]
43
44 data["Reach"] = data["Reach"].str.replace(",","")
45
46 data["Reach"]
47
48 data["Reach"] = data["Reach"].str.replace(".","")
49
50 data["Reach"]
51
52 data["Reach"] = data["Reach"].str.replace(",","")
53
54 data["Reach"]
55
56 data["Reach"] = data["Reach"].str.replace(".","")
57
58 data["Reach"]
59
60 data["Reach"] = data["Reach"].str.replace(",","")
61
62 data["Reach"]
63
64 data["Reach"] = data["Reach"].str.replace(".","")
65
66 data["Reach"]
67
68 data["Reach"] = data["Reach"].str.replace(",","")
69
70 data["Reach"]
71
72 data["Reach"] = data["Reach"].str.replace(".","")
73
74 data["Reach"]
75
76 data["Reach"] = data["Reach"].str.replace(",","")
77
78 data["Reach"]
79
80 data["Reach"] = data["Reach"].str.replace(".","")
81
82 data["Reach"]
83
84 data["Reach"] = data["Reach"].str.replace(",","")
85
86 data["Reach"]
87
88 data["Reach"] = data["Reach"].str.replace(".","")
89
90 data["Reach"]
91
92 data["Reach"] = data["Reach"].str.replace(",","")
93
94 data["Reach"]
95
96 data["Reach"] = data["Reach"].str.replace(".","")
97
98 data["Reach"]
99
100 data["Reach"] = data["Reach"].str.replace(",","")
101
102 data["Reach"]
103
104 data["Reach"] = data["Reach"].str.replace(".","")
105
106 data["Reach"]
107
108 data["Reach"] = data["Reach"].str.replace(",","")
109
110 data["Reach"]
111
112 data["Reach"] = data["Reach"].str.replace(".","")
113
114 data["Reach"]
115
116 data["Reach"] = data["Reach"].str.replace(",","")
117
118 data["Reach"]
119
120 data["Reach"] = data["Reach"].str.replace(".","")
121
122 data["Reach"]
123
124 data["Reach"] = data["Reach"].str.replace(",","")
125
126 data["Reach"]
127
128 data["Reach"] = data["Reach"].str.replace(".","")
129
130 data["Reach"]
131
132 data["Reach"] = data["Reach"].str.replace(",","")
133
134 data["Reach"]
135
136 data["Reach"] = data["Reach"].str.replace(".","")
137
138 data["Reach"]
139
140 data["Reach"] = data["Reach"].str.replace(",","")
141
142 data["Reach"]
143
144 data["Reach"] = data["Reach"].str.replace(".","")
145
146 data["Reach"]
147
148 data["Reach"] = data["Reach"].str.replace(",","")
149
150 data["Reach"]
151
152 data["Reach"] = data["Reach"].str.replace(".","")
153
154 data["Reach"]
155
156 data["Reach"] = data["Reach"].str.replace(",","")
157
158 data["Reach"]
159
160 data["Reach"] = data["Reach"].str.replace(".","")
161
162 data["Reach"]
163
164 data["Reach"] = data["Reach"].str.replace(",","")
165
166 data["Reach"]
167
168 data["Reach"] = data["Reach"].str.replace(".","")
169
170 data["Reach"]
171
172 data["Reach"] = data["Reach"].str.replace(",","")
173
174 data["Reach"]
175
176 data["Reach"] = data["Reach"].str.replace(".","")
177
178 data["Reach"]
179
180 data["Reach"] = data["Reach"].str.replace(",","")
181
182 data["Reach"]
183
184 data["Reach"] = data["Reach"].str.replace(".","")
185
186 data["Reach"]
187
188 data["Reach"] = data["Reach"].str.replace(",","")
189
190 data["Reach"]
191
192 data["Reach"] = data["Reach"].str.replace(".","")
193
194 data["Reach"]
195
196 data["Reach"] = data["Reach"].str.replace(",","")
197
198 data["Reach"]
199
200 data["Reach"] = data["Reach"].str.replace(".","")
201
202 data["Reach"]
203
204 data["Reach"] = data["Reach"].str.replace(",","")
205
206 data["Reach"]
207
208 data["Reach"] = data["Reach"].str.replace(".","")
209
210 data["Reach"]
211
212 data["Reach"] = data["Reach"].str.replace(",","")
213
214 data["Reach"]
215
216 data["Reach"] = data["Reach"].str.replace(".","")
217
218 data["Reach"]
219
220 data["Reach"] = data["Reach"].str.replace(",","")
221
222 data["Reach"]
223
224 data["Reach"] = data["Reach"].str.replace(".","")
225
226 data["Reach"]
227
228 data["Reach"] = data["Reach"].str.replace(",","")
229
230 data["Reach"]
231
232 data["Reach"] = data["Reach"].str.replace(".","")
233
234 data["Reach"]
235
236 data["Reach"] = data["Reach"].str.replace(",","")
237
238 data["Reach"]
239
240 data["Reach"] = data["Reach"].str.replace(".","")
241
242 data["Reach"]
243
244 data["Reach"] = data["Reach"].str.replace(",","")
245
246 data["Reach"]
247
248 data["Reach"] = data["Reach"].str.replace(".","")
249
250 data["Reach"]
251
252 data["Reach"] = data["Reach"].str.replace(",","")
253
254 data["Reach"]
255
256 data["Reach"] = data["Reach"].str.replace(".","")
257
258 data["Reach"]
259
260 data["Reach"] = data["Reach"].str.replace(",","")
261
262 data["Reach"]
263
264 data["Reach"] = data["Reach"].str.replace(".","")
265
266 data["Reach"]
267
268 data["Reach"] = data["Reach"].str.replace(",","")
269
270 data["Reach"]
271
272 data["Reach"] = data["Reach"].str.replace(".","")
273
274 data["Reach"]
275
276 data["Reach"] = data["Reach"].str.replace(",","")
277
278 data["Reach"]
279
280 data["Reach"] = data["Reach"].str.replace(".","")
281
282 data["Reach"]
283
284 data["Reach"] = data["Reach"].str.replace(",","")
285
286 data["Reach"]
287
288 data["Reach"] = data["Reach"].str.replace(".","")
289
290 data["Reach"]
291
292 data["Reach"] = data["Reach"].str.replace(",","")
293
294 data["Reach"]
295
296 data["Reach"] = data["Reach"].str.replace(".","")
297
298 data["Reach"]
299
299 data["Reach"] = data["Reach"].str.replace(",","")
```

```

3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import plotly.express as px
6 from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
7 from sklearn.model_selection import train_test_split
8 from sklearn.linear_model import PassiveAggressiveRe...
9
10 data = pd.read_csv("Instagram.csv", encoding = 'latin1')
11 print(data.head())
12

```

	Impressions	From Home	From Hashtags	From Explore	From Other
Saves \ 98.0	3920.0	2586.0	1028.0	619.0	56.0
194.0	5394.0	2727.0	1838.0	1174.0	78.0
41.0	4021.0	2085.0	1188.0	0.0	533.0
172.0	4528.0	2700.0	621.0	932.0	73.0
96.0	2518.0	1704.0	255.0	279.0	37.0

	Comments	Shares	Likes	Profile Visits	Follows \ 2.0
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:

```
1 data.isnull().sum()
```

```
2
```

```
1 data.isnull().sum()
```

```
2
```

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:

```
1 data = data.dropna()
```

```
2
```

```
1 data = data.dropna()
```

```
2
```

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

```
1 data.info()
```

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

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

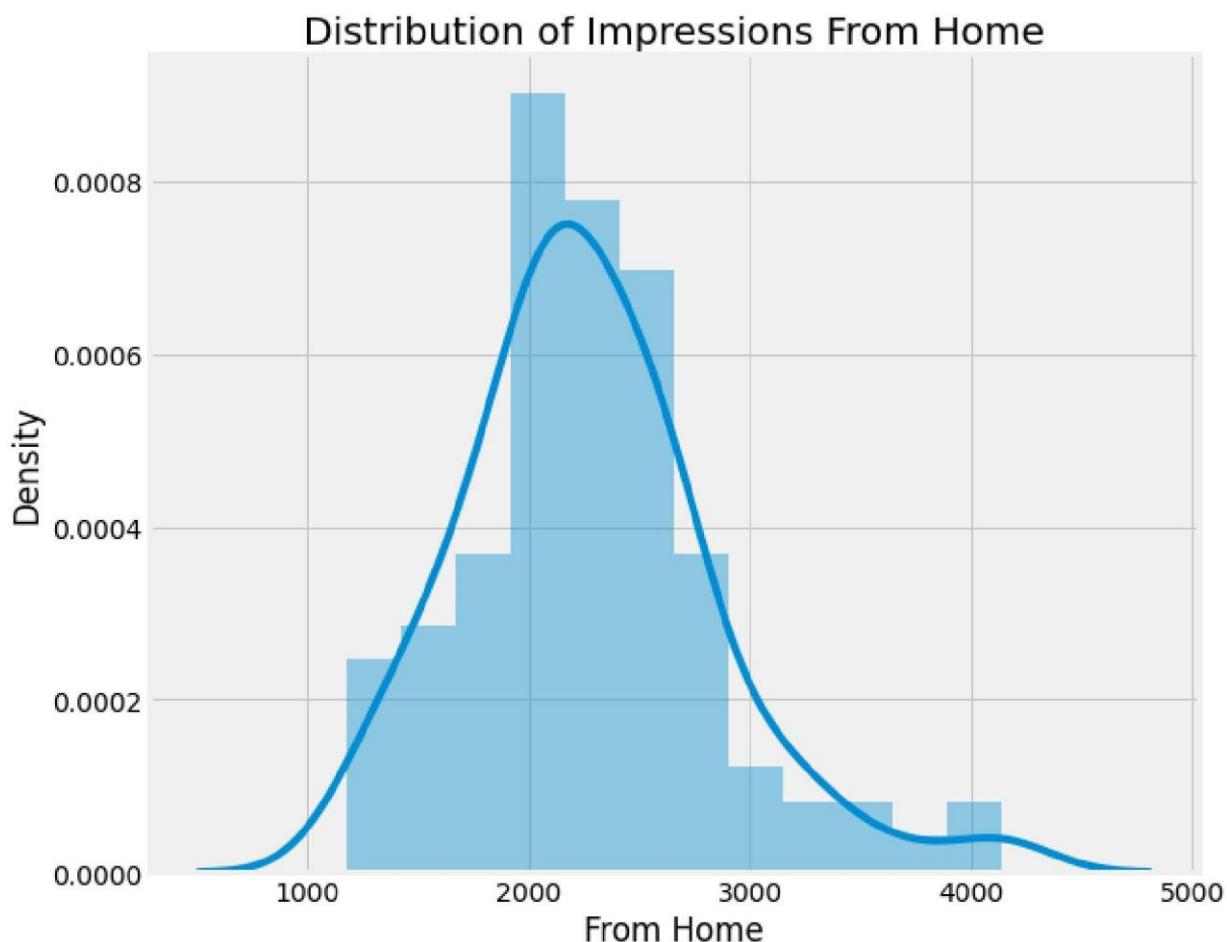
```
Data columns (total 13 columns):
```

#	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
	memory usage:	10.8+ KB	object(2)11

Analyzing Instagram Reach

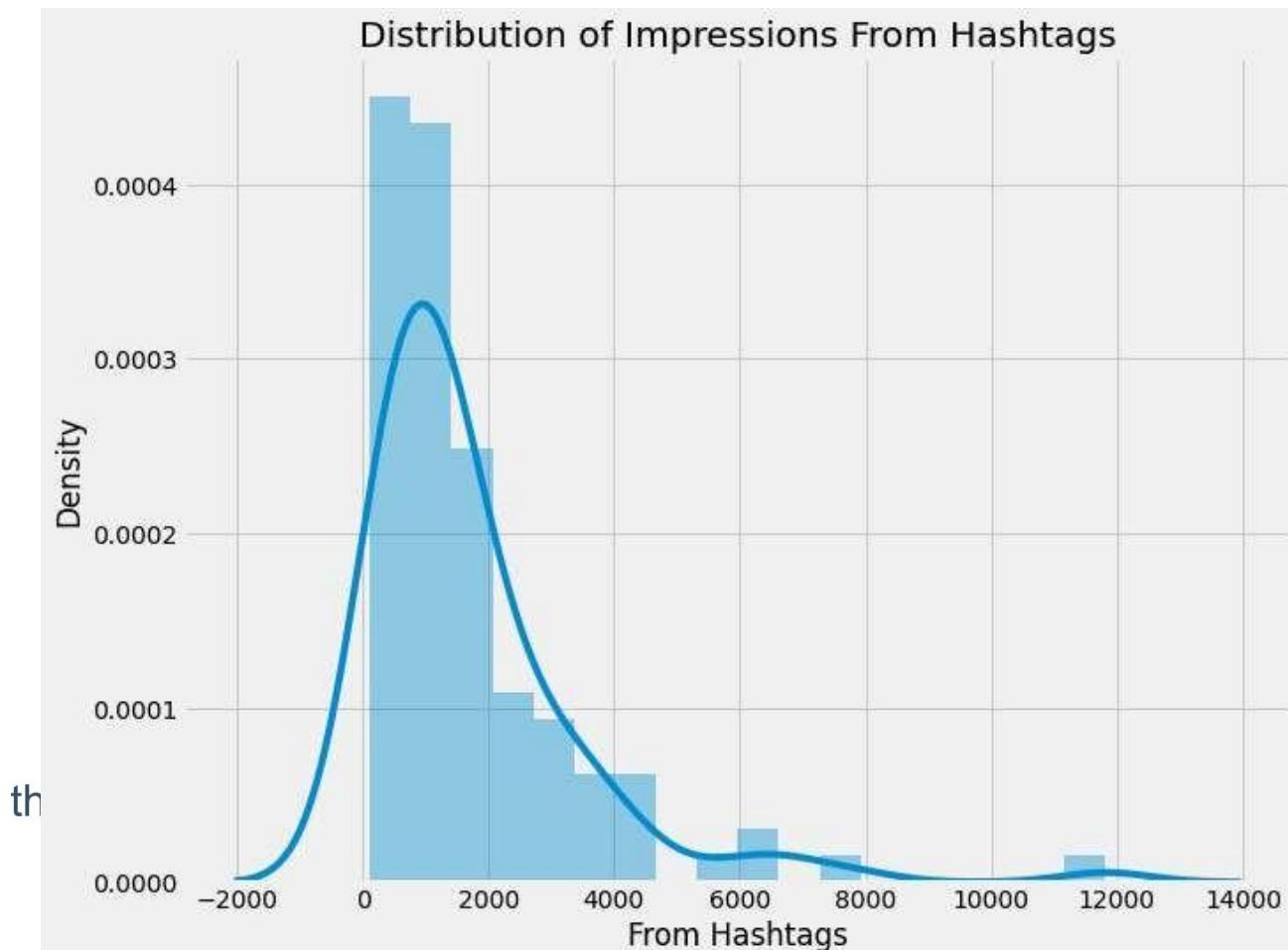
Now let's start with analyzing the reach of my Instagram posts. I will first have a look at the distribution of impressions I have received from home:

```
1 plt.figure(figsize=(10, 8))
2 plt.style.use('fivethirtyeight')
3 plt.title("Distribution of Impressions From Home")
4 sns.distplot(data['From Home'])
5 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:

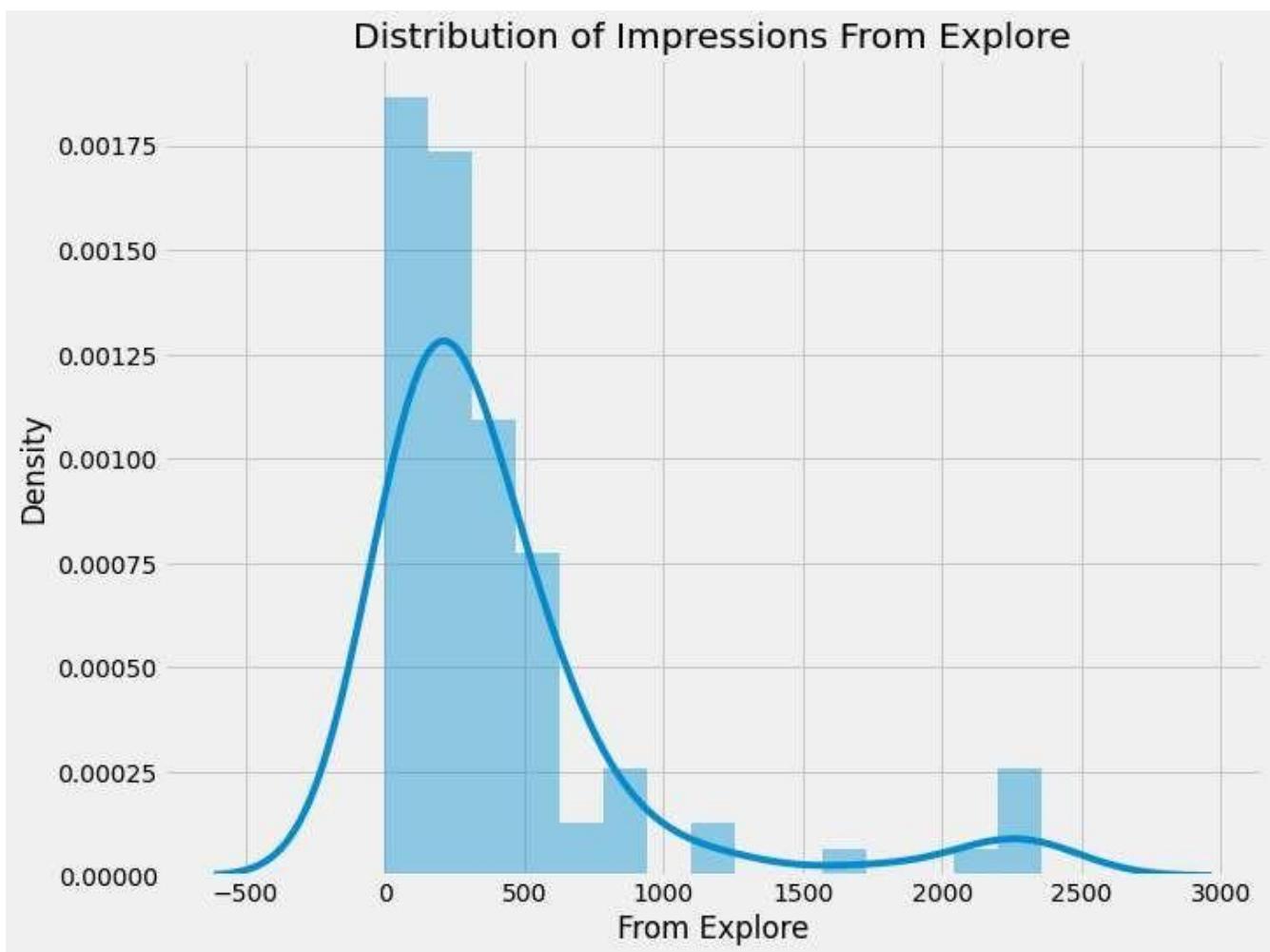
```
1 plt.figure(figsize=(10, 8))  
2 plt.title("Distribution of Impressions From Hashtags")  
3 sns.distplot(data['From Hashtags'])  
4 plt.show()  
5  
1 plt.figure(figsize=(10, 8))  
2 plt.title("Distribution of Impressions From Hashtags")  
3 sns distplot(data['From Hashtags'])  
4 plt.show()
```



Hashtags are tools we use to categorize our posts on Instagram so 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:

```
1 plt.figure(figsize=(10, 8))
2 plt.title("Distribution of Impressions From Explore")
3 sns.distplot(data['From Explore'])
4 plt.show()
5

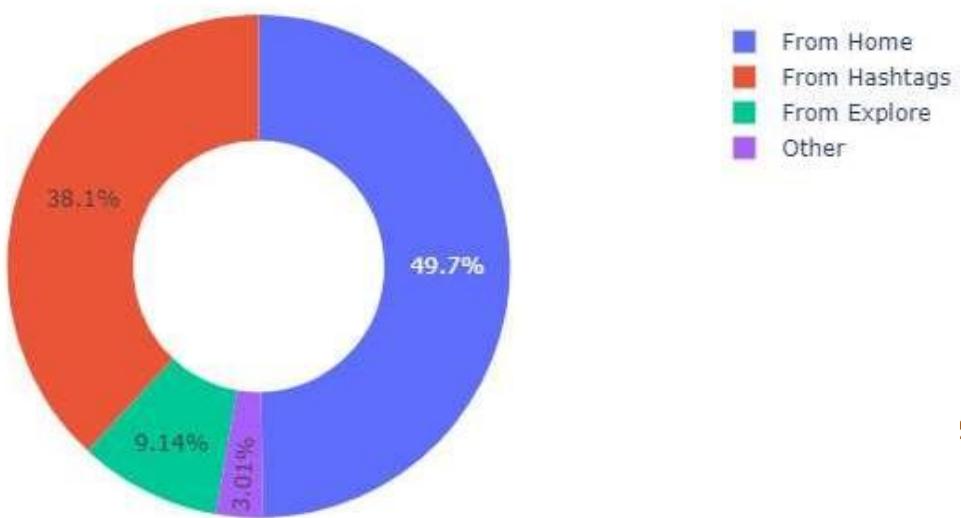
1 plt.figure(figsize=(10, 8))
2 plt.title("Distribution of Impressions From Explore")
3 sns.distplot(data['From Explore'])
4 plt.show()
```



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:

Impressions on Instagram Posts From Various Sources



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.

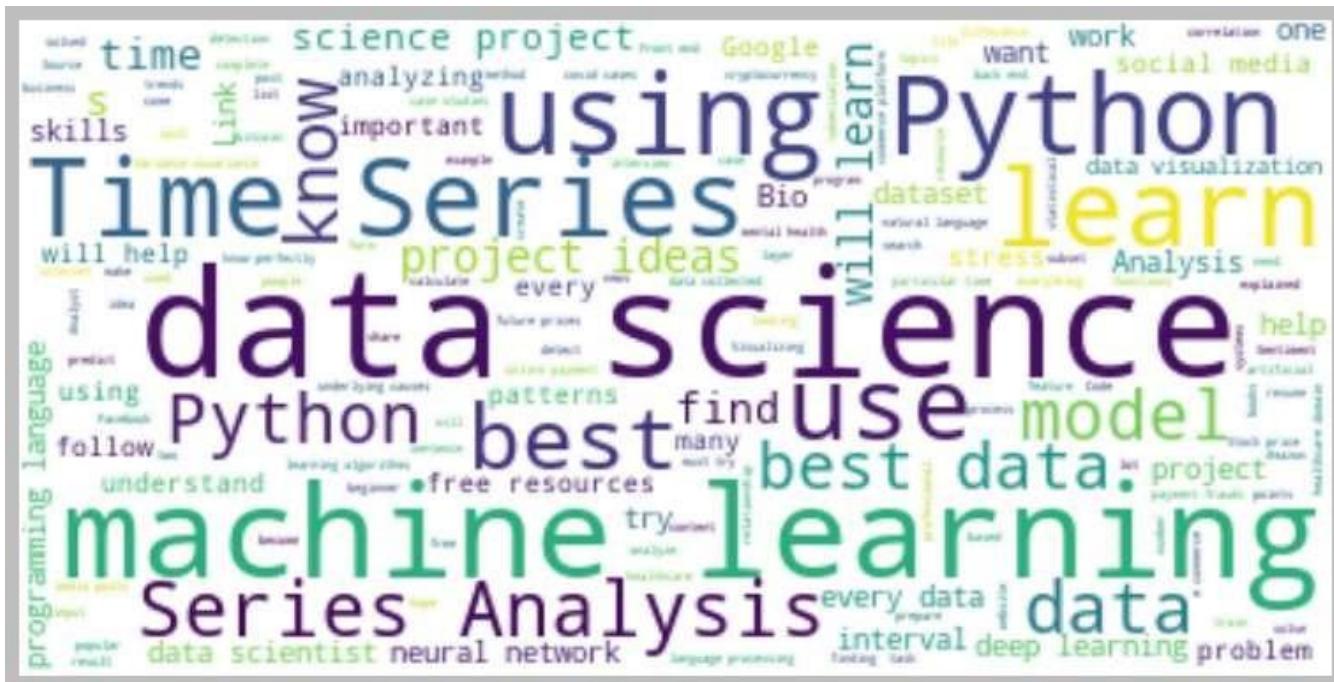
Analyzing Content

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:

```
1 text = " ".join(i for i in data.Caption)
2 stopwords = set(STOPWORDS)
3 wordcloud = WordCloud(stopwords=stopwords, backgroundc
4 plt.style.use('classic')
5 plt.figure( figsize=(12,10))
6 plt.imshow(wordcloud, interpolation='bilinear')
7 plt.axis("off")
8 plt.show()
9
```

```
1 text = " ".join(i for i in data.Caption)
2 stopwords = set(STOPWORDS)
3 wordcloud = WordCloud(stopwords=stopwords, backgroundc
4 plt.style.use('classic')
5 plt.figure( figsize=(12,10))
6 plt.imshow(wordcloud, interpolation='bilinear')
7 plt.axis("off")
8 plt.show()
9
```

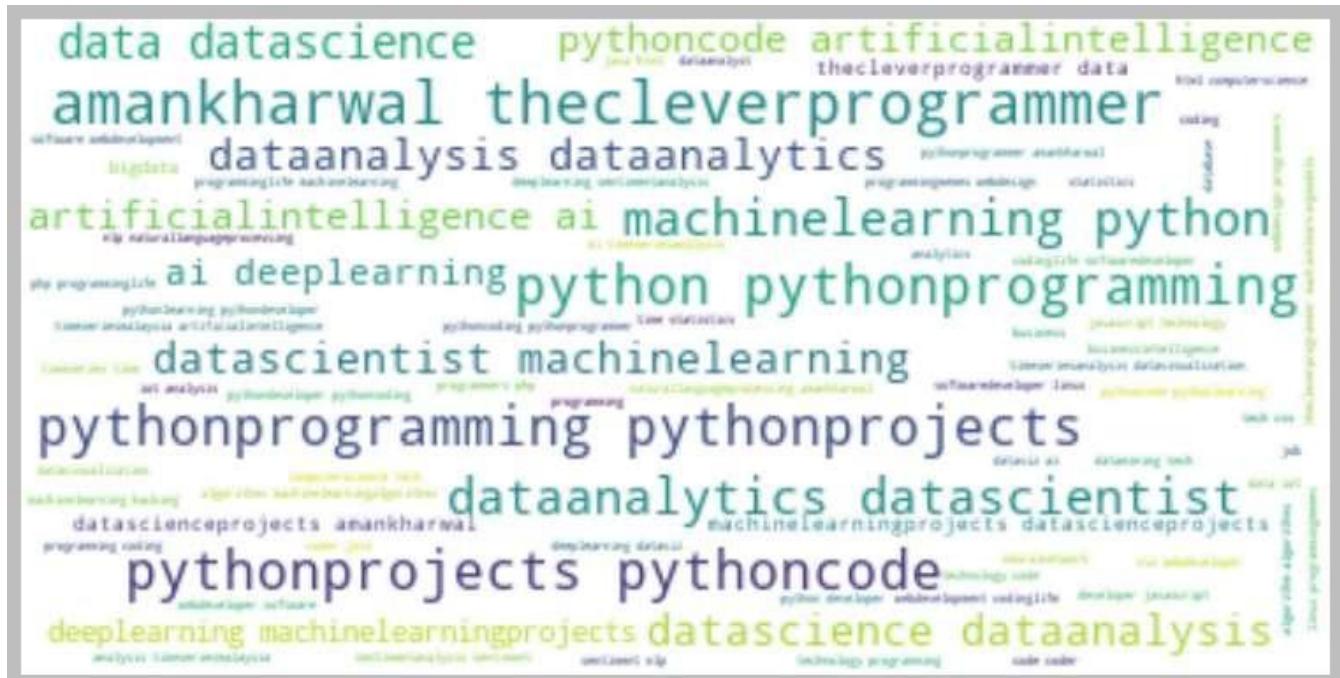


Now let's create a wordcloud of the hashtags column to look at the most used hashtags in my Instagram posts:

```
1 text = " ".join(i for i in data.Hashtags)
2 stopwords = set(STOPWORDS)
3 wordcloud = WordCloud(stopwords=stopwords, background_color='white')
4 plt.figure(figsize=(12,10))
5 plt.imshow(wordcloud, interpolation='bilinear')
6 plt.axis("off")
```

```

7 plt.show( )
8
1 text = " ".join(i for i in data.Hashtags)
2 stopwords = set(STOPWORDS)
3 wordcloud = WordCloud(stopwords=stopwords, backgroundcolor='white',
4 plt.figure(figsize=(12,10))
5 plt.imshow(wordcloud, interpolation='bilinear')
6 plt.axis("off")
7 plt.show()
8
```



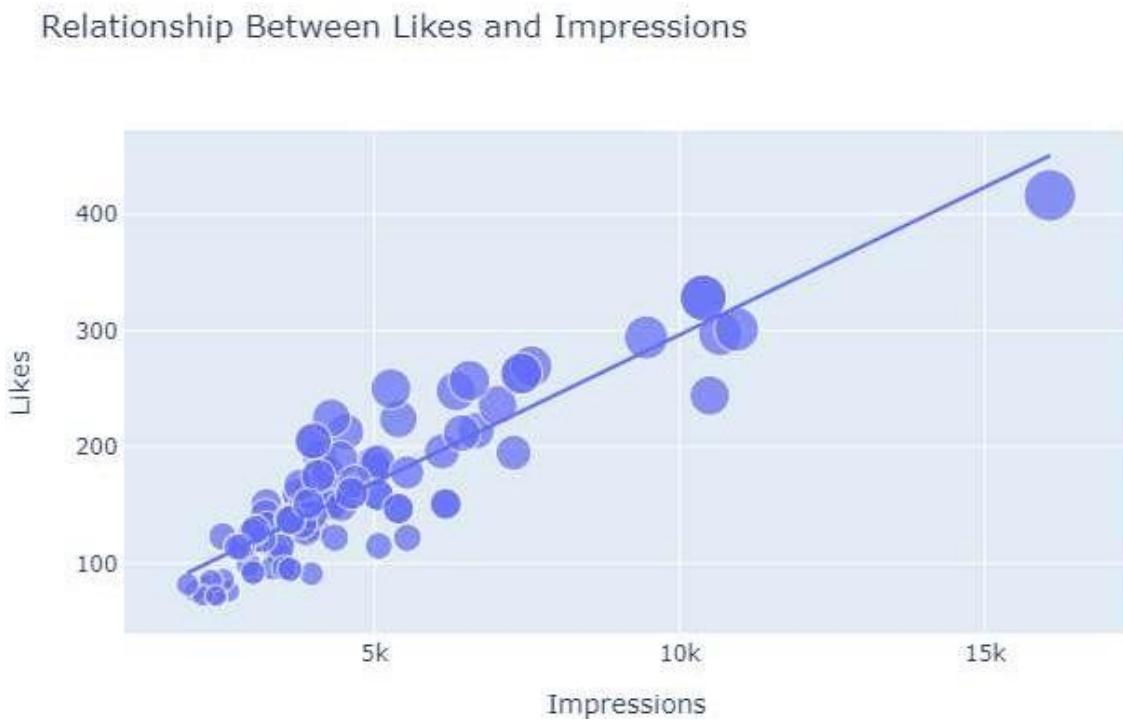
Analyzing Relationships

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:

```
1 figure = px.scatter(data_frame = data, x="Impressions"
2                         y="Likes", size="Likes", trendline="linear",
3                         title = "Relationship Between Likes and Impressions")
4 figure.show()
5

1 figure = px.scatter(data_frame = data, x="Impressions"
2                         y="Likes", size="Likes", trendline="linear",
3                         title = "Relationship Between Likes and Impressions")
4 figure.show()
5
```

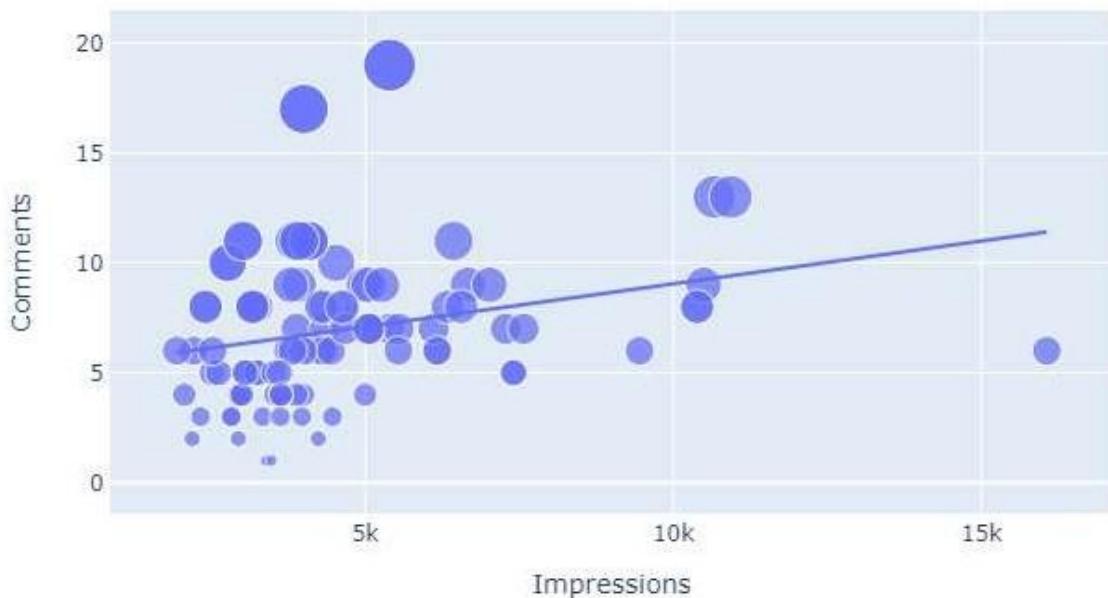


There is a linear relationship between the number of likes and the reach I got on Instagram. Now let's see the relationship between the number of comments and the number of impressions on my Instagram posts:

```
1 figure = px.scatter(data_frame = data, x="Impressions"
2                               y="Comments", size="Comments", tr
3                               title = "Relationship Between Com
4 figure.show()
5
```

```
1 figure = px.scatter(data_frame = data, x="Impressions"
2                               y="Comments", size="Comments", tr
3                               title = "Relationship Between Com
4 figure.show()
5
```

Relationship Between Comments and Total Impressions

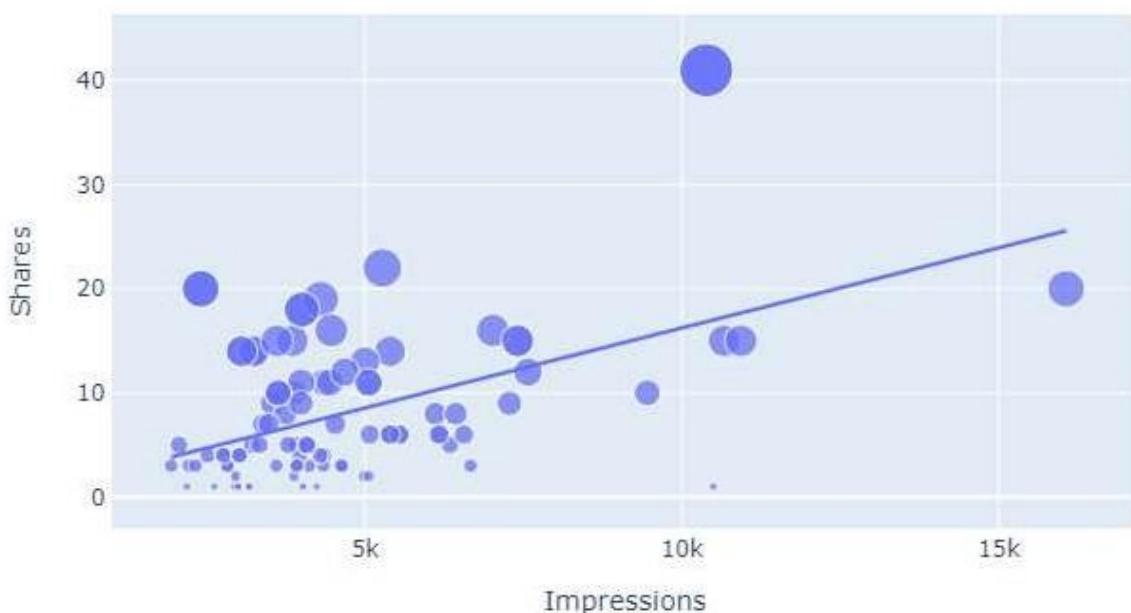


It looks like the number of comments we get on a post doesn't affect its reach. Now let's have a look at the relationship between the number of shares and the number of impressions:

```
1 figure = px.scatter(data_frame = data, x="Impressions"
2                         y="Shares", size="Shares", trend]
3                         title = "Relationship Between Sha
4 figure.show()
5
```

```
1 figure = px.scatter(data_frame = data, x="Impressions"
2                         y="Shares", size="Shares", trend]
3                         title = "Relationship Between Sha
4 figure.show()
5
```

Relationship Between Shares and Total Impressions

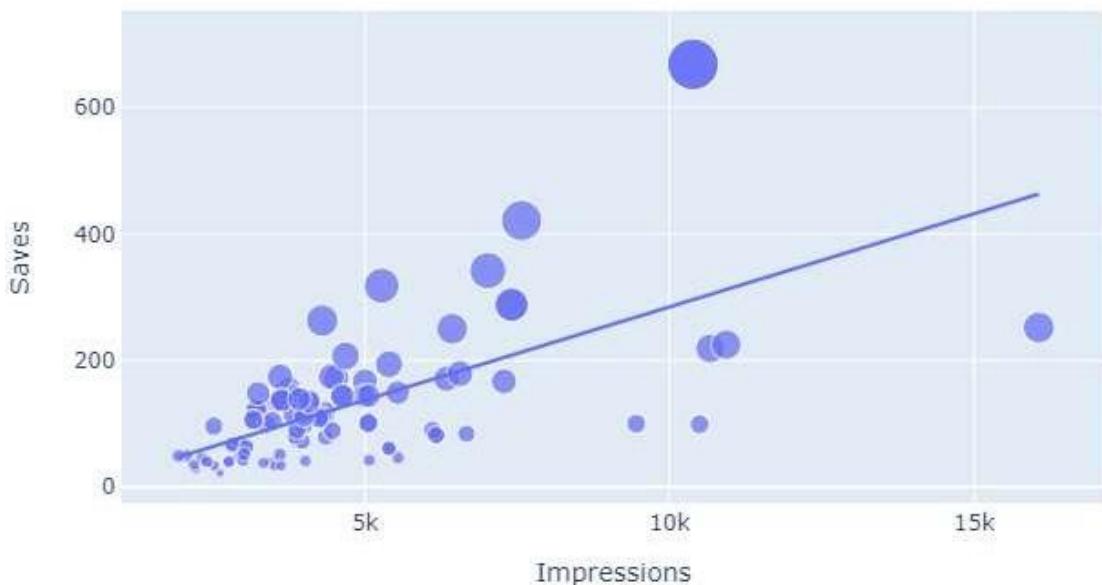


number of impressions between the number of saves and the ave

```
1 figure = px.scatter(data_frame = data, x="Impressions"
2                               y="Saves", size="Saves", trendline="linear",
3                               title = "Relationship Between Post Impressions and Saves")
4 figure.show()
```

```
1 figure = px.scatter(data_frame = data, x="Impressions"
2                               y="Saves", size="Saves", trendline="linear",
3                               title = "Relationship Between Post Impressions and Saves")
4 figure.show()
```

Relationship Between Post Saves and Total Impressions



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:

```
1 correlation = data.corr()  
2 print(correlation["Impressions"].sort_values(ascending=True))  
3
```

```
1 correlation = data.corr()  
2 print(correlation["Impressions"].sort_values(ascending=True))  
3
```

```
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, conversion 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 conversion rate of my

```
1 conversion_rate = (data["Follows"].sum() / data["Profile Visits"])
2 print(conversion_rate)
3
4
5
```

```
1 conversion_rate = (data["Follows"].sum() / data["Profile Visits"])
2 print(conversion_rate)
3
```

31.17770767613039

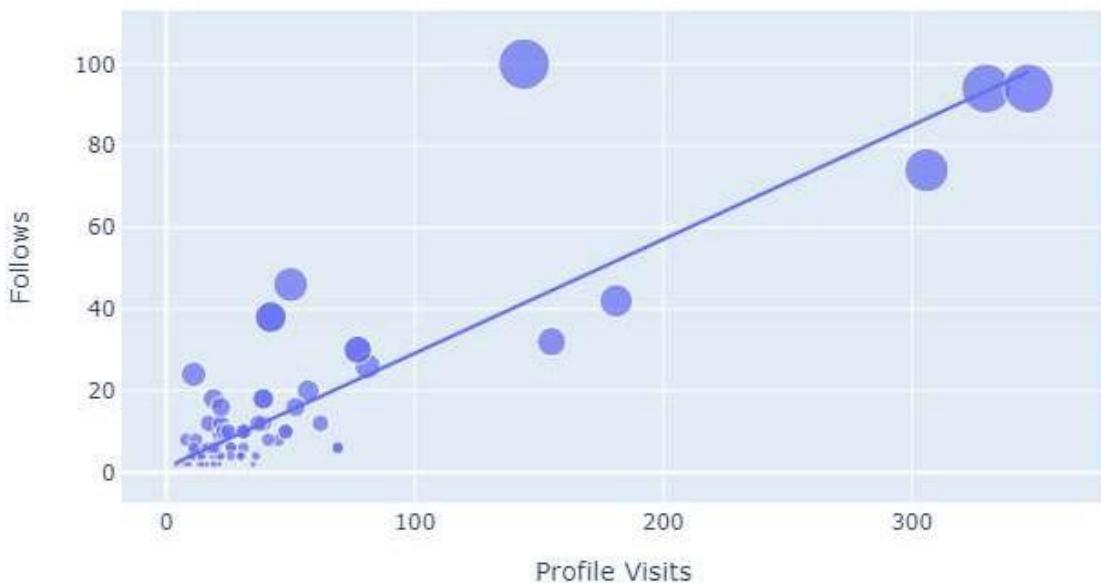
So the conversation rate of my Instagram account is 31% which sounds like a very good conversation rate. Let's have a look at the relationship between the total profile visits and the number of followers gained from all profile visits:

```
1 figure = px.scatter(data_frame = data, x="Profile Visits", y="Follows", size="Follows", trendline="linear", title = "Relationship Between Profile Visits and Followers")
2
3
4 figure.show()
```

5

```
1 figure = px.scatter(data_frame = data, x="Profile Visits", y="Follows", size="Follows", trendline="linear", title = "Relationship Between Profile Visits and Followers Gained")
2
3
4 figure.show()
5
```

Relationship Between Profile Visits and Followers Gained



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:

```
1 x = np.array(data[['Likes', 'Saves', 'Comments', 'Shares',
2                               'Profile Visits', 'Follows']])
3 y = np.array(data["Impressions"])
4 xtrain, xtest, ytrain, ytest = train_test_split(x, y,
5                                                 test_size=0.2, random_state=42)
6
7
```



```
1 x = np.array(data[['Likes', 'Saves', 'Comments', 'Shares',
2                               'Profile Visits', 'Follows']])
3 y = np.array(data["Impressions"])
4 xtrain, xtest, ytrain, ytest = train_test_split(x, y,
```

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

```
1 model      = PassiveAggressiveRegressor()
2 model.fit(xtrain, ytrain)
3 model.score(xtest, ytest)
4

1 model = PassiveAggressiveRegressor()
2 model.fit(xtrain, ytrain)
3 model.score(xtest, ytest)
4
```

0.9428392959517574

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

```
1 # Features = [['Likes','Saves', 'Comments', 'Shares',
2 features = np.array([[282.0, 233.0, 4.0, 9.0, 165.0,
3 model.predict(features)
4

1 # Features = [['Likes','Saves', 'Comments', 'Shares',
2 features = np.array([[282.0, 233.0, 4.0, 9.0, 165.0,
3 model.predict(features)
4

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.

