

instagram-data-analysis

December 2, 2023

Instagram Reach Analysis

```
[1]: # Importing the necessary python libries
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
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveRegressor
```

```
[2]: df = pd.read_csv('/content/Instagram data.csv',encoding = 'latin1')
df.head()
```

```
[2]: 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...

```

```
[3]: 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

```

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

```

[4]: 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

```

```

[5]: plt.figure(figsize=(10,5))
     plt.title('Distribution of impressions From Home')

```

```
sns.distplot(df['From Home'])
plt.show()
```

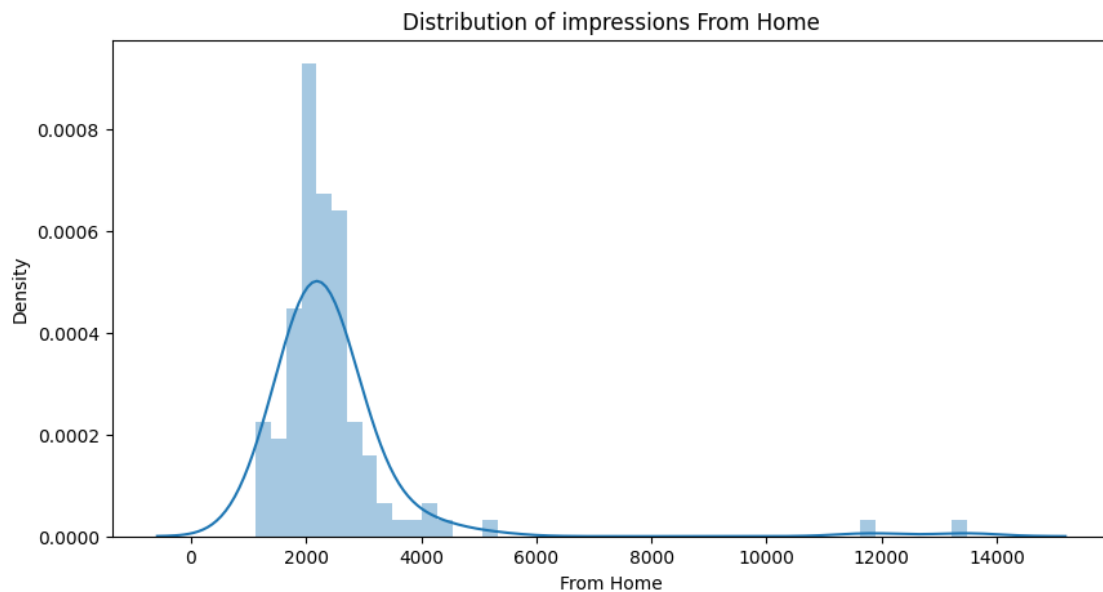
<ipython-input-5-23da7dd62aac>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['From Home'])
```



```
[6]: def column_name():
      plt.figure(figsize=(10,5))
      plt.title('Distribution of impressions ')
      sns.distplot(df['column_name'])
      plt.show()
      return column_name
```

```
[7]: plt.figure(figsize=(10,5))
      plt.title('Distribution of impressions From Hashtags')
      sns.distplot(df['From Hashtags'])
      plt.show()
```

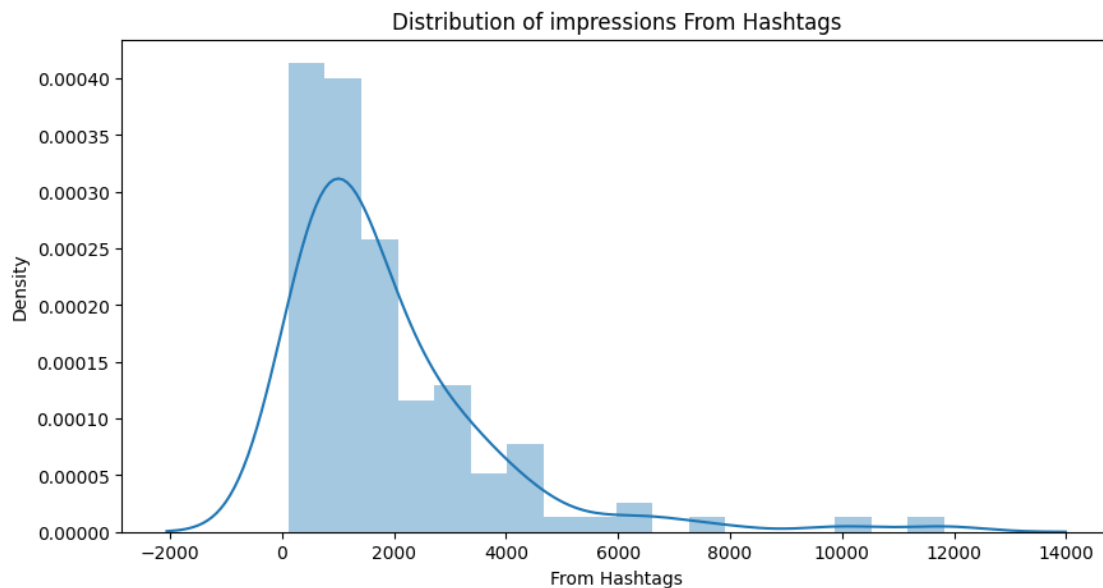
<ipython-input-7-3f113c94b0d6>:3: UserWarning:

``distplot`` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either ``displot`` (a figure-level function with similar flexibility) or ``histplot`` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['From Hashtags'])
```



Hashtags are tool we use to categorize our post on instagram so that we can reach more people based on the kind of content. we are greeting looking at hashtags impression show that not all posts, can be reached using hashtags, but many new users can be reached from hashtags.

```
[8]: plt.figure(figsize=(10,8))
plt.style.use('fivethirtyeight')
plt.title('Distribution of Impression From Explore')
sns.distplot(df['From Explore'])
plt.show()
```

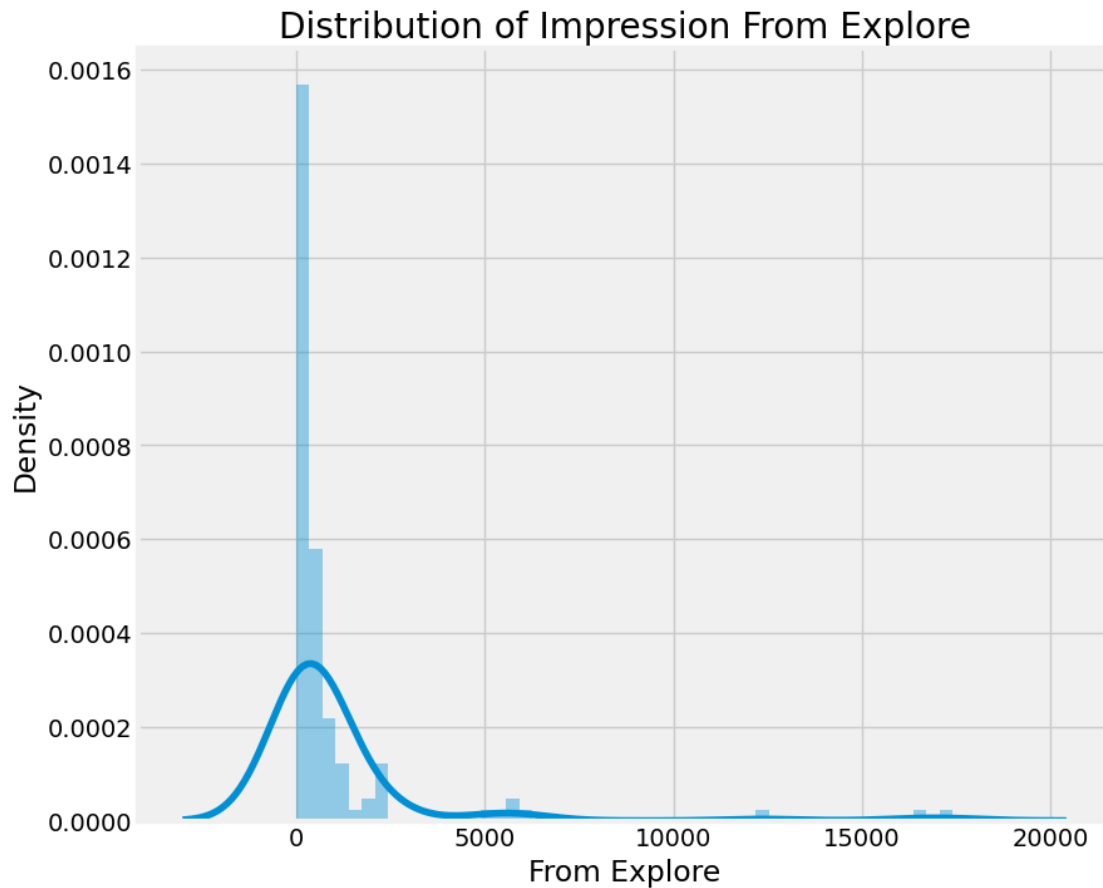
<ipython-input-8-b85446298871>:4: UserWarning:

``distplot`` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either ``displot`` (a figure-level function with similar flexibility) or ``histplot`` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['From Explore'])
```



The explore section of instagram is the recommendation system of instagram. Its recommended posts to the user based on their preferences and interests. By looking at the impression, I have received from the explore selection, I can say that instagram does not recommend our posts much to the users. Some posts have received a good reach from the explore selection.

```
[15]: 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 = 'Impressions on Instagram Posts From Various Sources',
        ↪hole=0.5)
figure.show()

```

```

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

```

```

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

```

```

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

```

```

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

```

```

[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

```

<ipython-input-20-0590b1fd9c8c>:1: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
[22]: # Analyzing Conversion Rate
conversion_rate = (df["Follows"].sum() / df["Profile Visits"].sum()) * 100
print(conversion_rate)
```

41.00265604249668

```
[23]: figure = px.scatter(data_frame = df, x="Profile Visits",
                        y="Follows", size="Follows", trendline="ols",
                        title = "Relationship Between Profile Visits and Followers_
↳Gained")
figure.show()
```

```
[ ]:
```