Insta_reach_analysis

April 30, 2023

0.0.1 Problem Statement:

A business is using an instagram profile/page to reach out its followers and trying to find out the what impacts the most reach to its follower. The business wants to find out the individual impact of various KPIs on the business reach.

[1]:

```
pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\saad\anaconda3\lib\site-
packages (1.8.2.2)
Requirement already satisfied: matplotlib in c:\users\saad\anaconda3\lib\site-
packages (from wordcloud) (3.2.2)
Requirement already satisfied: pillow in c:\users\saad\anaconda3\lib\site-
packages (from wordcloud) (9.0.1)
Requirement already satisfied: numpy>=1.6.1 in c:\users\saad\anaconda3\lib\site-
packages (from wordcloud) (1.19.5)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\saad\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.3.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
c:\users\saad\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: python-dateutil>=2.1 in
c:\users\saad\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\saad\anaconda3\lib\site-
packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: six>=1.5 in c:\users\saad\anaconda3\lib\site-
packages (from python-dateutil>=2.1->matplotlib->wordcloud) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[2]:

```
# Import Libraries
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
%matplotlib inline
```

[3]:
```
df = pd.read_csv('Instagram data.csv', encoding = 'latin1')
df.head()
```

[3]:
```
   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…
```

[4]:
```
#Checking null values
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
```

The dataset contains no null values

```
[5]: #Lets have a look at the culumns and datatype of each column
     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
```

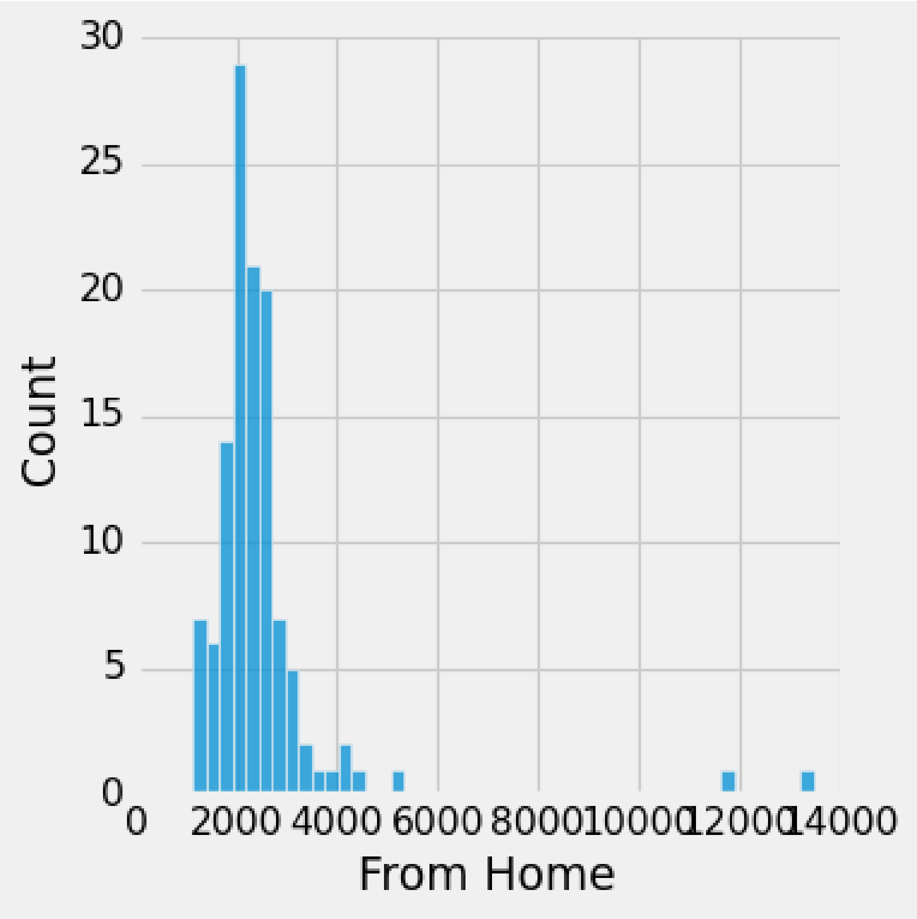
```
[6]: #Here we are checking a descriptive statistical  overview of the dataset
     df.describe()
```

```
[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
```

[22]:
```
# Impressions received from home
plt.figure(figsize=(10, 8))
plt.style.use('fivethirtyeight')
#plt.title("Distribution of Impressions From Home")
sns.displot(df['From Home'])
plt.show()
```

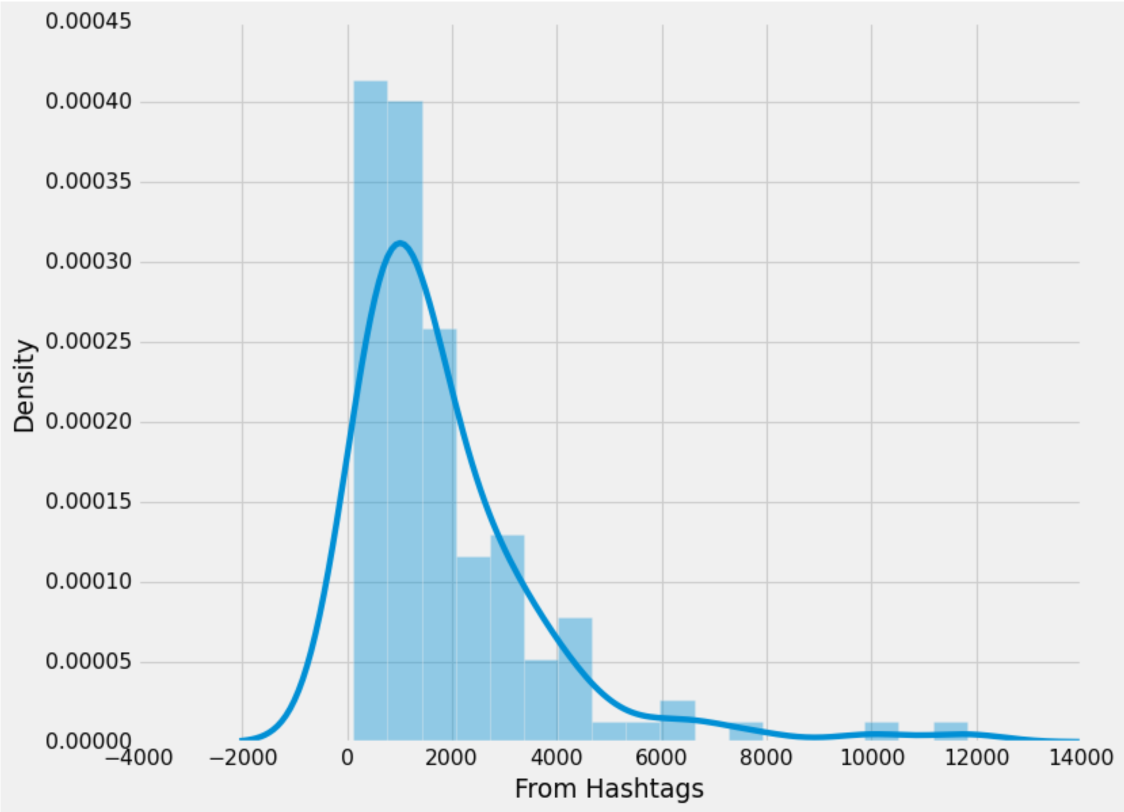
```
<Figure size 800x640 with 0 Axes>
```



It seems that it is hard to reach most of the follower from home.

```
[21]: #From hashtags
plt.figure(figsize=(10, 8))
#plt.title("Distribution of Impressions From Hashtags")
sns.distplot(df['From Hashtags'])
plt.show()
```

```
C:\Users\saad\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning:

`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```

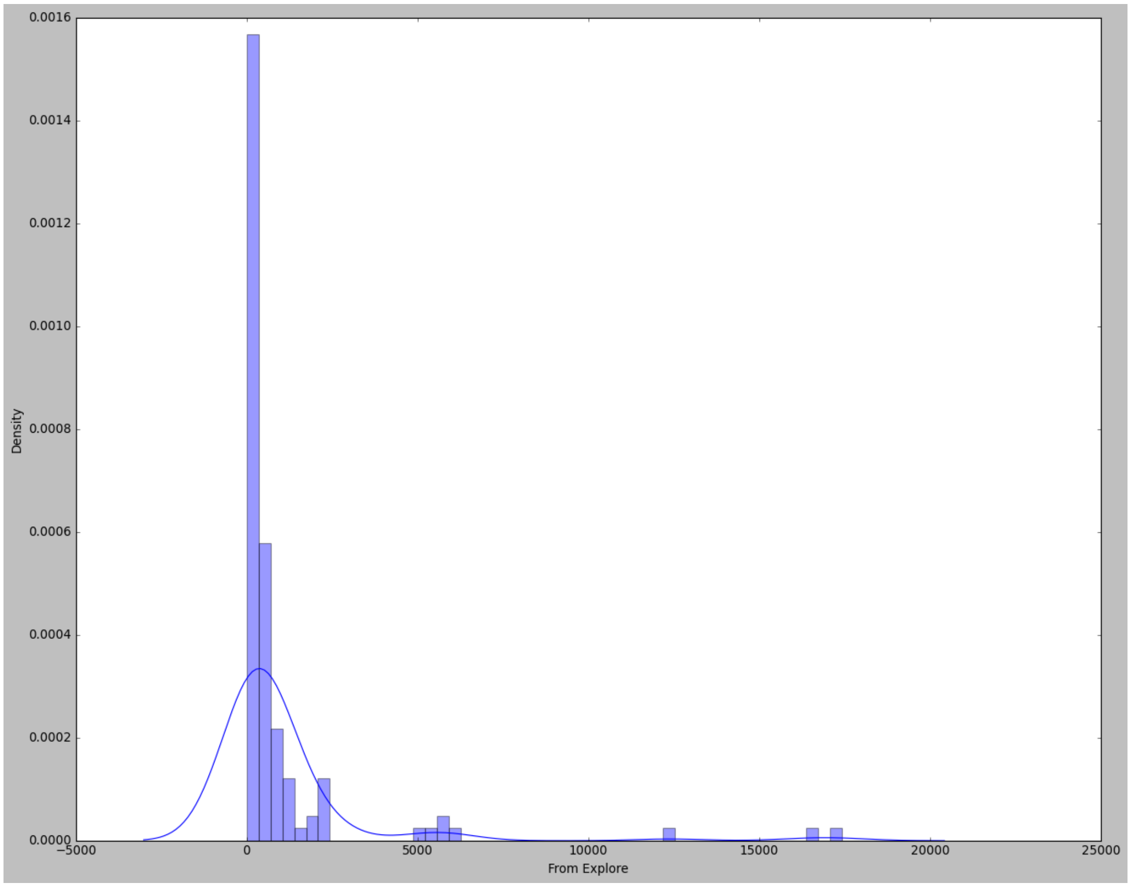


- Hashtags refers to the catagory of posts use to reach target users.

It is clear that hashtags are helpful to reach new users but it is not always possible to reach all the followers using hashtags.

[19]:
```
#From Explore
plt.figure(figsize=(18, 14))
#plt.title("Distribution of Impressions From Explore")
sns.distplot(df['From Explore'])
plt.show()
```

```
C:\Users\saad\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning:

`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```



- Explore is basically a recomendation algorithm on instagram that helps post to rach meaningful users based on their catagory of interests.

From the the graph we found out that instagram doesnot recomend/show our post to expected amount of users. Although some of the posts receives good reach but comperatively lower than the reach we received from hashtags. It might also implies that there might be some issues regarding post production/making. For instance, lower ranking keywords are being used or posting schedule etc.

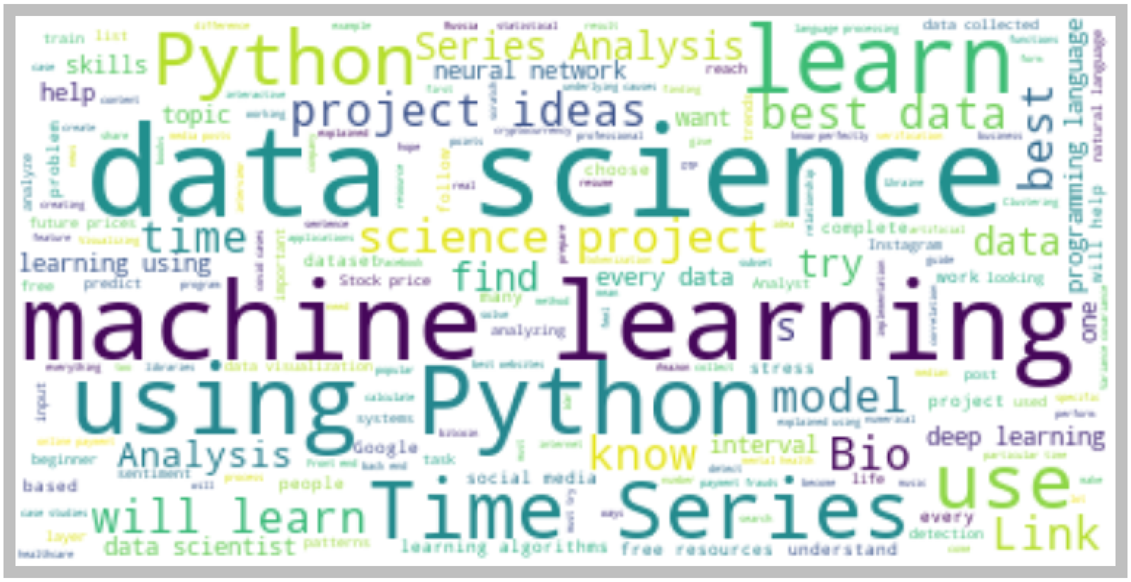
[10]:

```
# Let us explore the impression from all the sources.
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','Other']
values = [home, hashtags, explore, other]

fig = px.pie(df, values=values, names=labels,
             title='Impressions on Instagram Posts From Various Sources')
fig.show()
```

We receive most (around 45%) of our reach from the followers whereas lowest reach comes from other sources. Except followers, it seems that hashtags can be an effective way to reach other users also.

[11]:

```
#Lets have a look at the most used word in the captions.
text = " ".join(i for i in df.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()
```



As we can see that the most used word in the caption of our posts are machine learning, time series, data sciences etc.

0.0.2 Let us analyse the relationship among different variables.

[12]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Likes", size="Likes",
                    title = "Relationship Between Likes and Impressions")
figure.show()
```

[13]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Comments", size="Comments",
                    title = "Relationship Between Comments and Total␣
 ↪Impressions")
figure.show()
```

[14]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Shares", size="Shares",
                    title = "Relationship Between Shares and Total Impressions")
figure.show()
```

[15]:
```
figure = px.scatter(data_frame = df, x="Impressions",
                    y="Saves", size="Saves",
                    title = "Relationship Between Post Saves and Total␣
 ↪Impressions")
figure.show()
```

From the above analysis we found out the following:

```
- There is strong correlation between 'likes' and 'Impressions'. The more the likes are, the mo
- Fromt the second analysis, it is clear that the relationship between 'comments' and 'impressi
- Shares and impressions are also correlated. However, the impact of shares wont be so signific
- Almost same in the case of relationship between 'saves' and 'Impressions'. Number of impressi
```

[16]:
```
# Just having a look at the correlation with impression of all columns.
corr = df.corr()
print(corr["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
```

[17]:

```
# How about the conversion rate?
conversion_rate = (df["Follows"].sum() / df["Profile Visits"].sum()) * 100
print('Rate of Conversion:',conversion_rate)
```

```
Rate of Conversion: 41.00265604249668
```

[18]:

```
figure = px.scatter(data_frame = df, x="Profile Visits",
                    y="Follows", size="Follows",
                    title = "Relationship Between Profile Visits and Followers␣
 ↪Gained")
figure.show()
```

Now it is clear that the number of follower increases with the increase number of profile visit, meaning that both the variables have linear relationship.

Conclusion: From the above analysis we can come to the decision that the company should focus on the using proper hashtags to reach more users outside of its followers. This will help the business gaining more reach and impression as well.