# End Semester Exam

PSOSM

Gagandeep Singh: 2016037
Suraj Prathik Kumar: 2016101
Vyshakh: 2016120

We used python to complete all the tasks and did it in Jupyter Notebook. We used **Tweepy api** to extract the information from Twitter.

## Question 1

The data was loaded into a dataframe using Pandas

```
In [9]: data.head(10)
```

Out[9]:

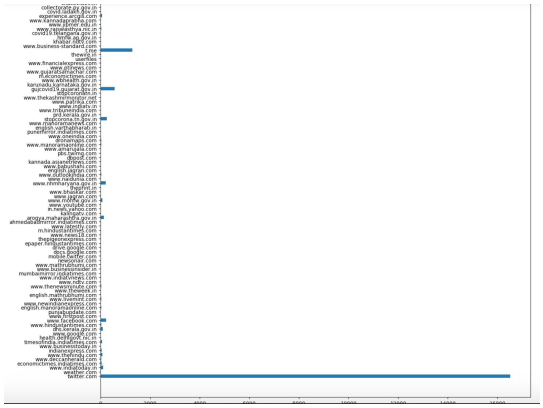
| | Patient Number | State Patient Number | Date Announced | Estimated Onset Date | Age Bracket | Gender | Detected City | Detected District | Detected State | State code | Current Status | Notes | Contracted from which Patient (Suspected) | Nationality | Ty transmis |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | KL-TS-P1 | 30/01/2020 | NaN | 20 | F | Thrissur | Thrissur | Kerala | KL | Recovered | Travelled from Wuhan | NaN | India | Imp |
| 1 | 2 | KL-AL-P1 | 02/02/2020 | NaN | NaN | NaN | Alappuzha | Alappuzha | Kerala | KL | Recovered | Travelled from Wuhan | NaN | India | Imp |
| 2 | 3 | KL-KS-P1 | 03/02/2020 | NaN | NaN | NaN | Kasaragod | Kasaragod | Kerala | KL | Recovered | Travelled from Wuhan | NaN | India | Imp |
| 3 | 4 | DL-P1 | 02/03/2020 | NaN | 45 | M | East Delhi (Mayur Vihar) | East Delhi | Delhi | DL | Recovered | Travelled from Austria, Italy | NaN | India | Imp |
| 4 | 5 | TS-P1 | 02/03/2020 | NaN | 24 | M | Hyderabad | Hyderabad | Telangana | TG | Recovered | Travelled from Dubai to Bangalore on 20th Feb,... | NaN | India | Imp |
| 5 | 6 | NaN | 03/03/2020 | NaN | 69 | M | Jaipur | Italians | Rajasthan | RJ | Recovered | Travelled from Italy | NaN | Italy | Imp |
| 6 | 7 | NaN | 04/03/2020 | NaN | 55 | NaN | Gurugram | Italians | Haryana | HR | Recovered | Travelled from Italy | P6 | Italy | Imp |

All the sites were separated out of the links by splitting the link by **“/”** .

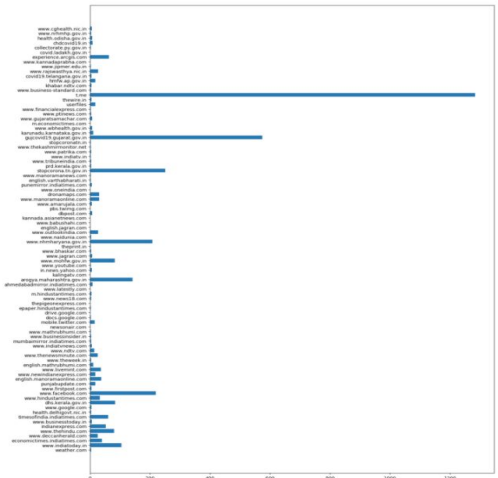
In [72]:
```
def getSite(link):
    l=link.split("/")
    print(l)
    if len(l)>3:
        return l[2]
    return ""
```

In [74]:
```
sources={}
nan=0
for ind,row in data.iterrows():
    print(type(row['Source_1']),type(row['Source_2']),type(row['Source_3']))
    if isinstance(row['Source_1'],str):
        s1=getSite(row['Source_1'])
        if s1!="":
            if s1 in sources.keys():
                sources[s1]+=1
            else:
                sources[s1]=1
    else:
        nan+=1
    if isinstance(row['Source_2'],str):
        s2=getSite(row['Source_2'])
        if s2!="":
            if s2 in sources.keys():
                sources[s2]+=1
            else:
                sources[s2]=1
    else:
        nan+=1
    if isinstance(row['Source_3'],str):
        s3=getSite(row['Source_3'])
        if s3!="":
            if s3 in sources.keys():
                sources[s3]+=1
            else:
                sources[s3]=1
    else:
        nan+=1
```

The major sites on which coronavirus was discussed was Twitter comprising of more than 95% of the links being twitter links.



With Twitter

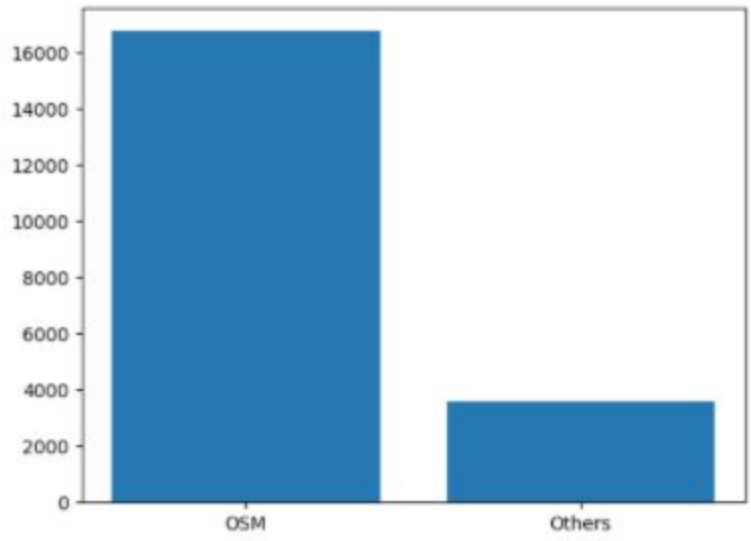


Without Twitter

## Question 2

In comparison to other sources of information regarding COVID-19, online social media has huge numbers of reports.

```
In [92]: print(totalOsm,others)
         16757 3603

In [93]: plt.bar(["OSM","Others"],[totalOsm,others])
Out[93]: <BarContainer object of 2 artists>
```




Social media platforms included Twitter and Facebook. Other platforms were majorly mainstream media sites.

## Question 3

From the twitter links which were collected, twitter id of the tweets were separated through the following code.

```
In [100]: def getTweetID(link):
              l=link.split("/")
              if len(l)<6:
                  return -1
              string=l[5]
              return string.split('?')[0]
```

The following tweet ids were collected.

```
In [103]: for i in tweetID:
              print(i)
```

```
1222819465143832577
1224221485805395968
1240878975846506496
1242088017197559808
1245225191178768386
1238482416936701953
1238882567987662855
1238882567987662855
1238882567987662855
1238882567987662855
1238882567987662855
1239183764946731008
1239183764946731008
1239099655780065282
1239597418640900096
1239454591894163458
1239454591894163458
1239454591894163458
1239454591894163458
1239464597486071808
```

Total tweet links were around 16000, but many tweet ids were repeated.
Therefore, in total there were only 904 different tweet id.
All the tweets were collected using Tweepy.

```
In [172]: for i in range(904,len(tweetID)):
              print(i)
              tw=api.get_status(tweetID[i],tweet_mode='extended')
              tweets.append(tw)
```

There were total of 893 tweets which were valid out of the 904 available

The following are the 10 latest tweets which were cellected.

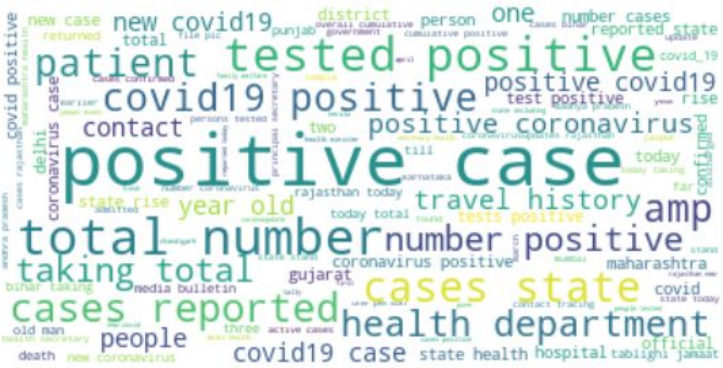
```
In [200]: latest

Out[200]: [(datetime.datetime(2020, 4, 19, 18, 19, 19), 29),
           (datetime.datetime(2020, 4, 19, 17, 46, 32), 449),
           (datetime.datetime(2020, 4, 19, 16, 49, 33), 132),
           (datetime.datetime(2020, 4, 19, 15, 59, 9), 778),
           (datetime.datetime(2020, 4, 19, 15, 18, 1), 90),
           (datetime.datetime(2020, 4, 19, 15, 9, 29), 443),
           (datetime.datetime(2020, 4, 19, 14, 39, 8), 648),
           (datetime.datetime(2020, 4, 19, 14, 28, 56), 179),
           (datetime.datetime(2020, 4, 19, 13, 33, 58), 678),
           (datetime.datetime(2020, 4, 19, 13, 18, 44), 314),
```

## Question 4

The text of all the tweets were collected. All the punctuation marks, links and stop words were removed from the texts using **re** and **nltk** libraries. All the texts were joined into a single document to create a wordcloud.

```
In [254]: wordcloud = WordCloud(background_color="white", max_words=2000)
          wordcloud.generate(allTweets)
          wordcloud.to_image()
```

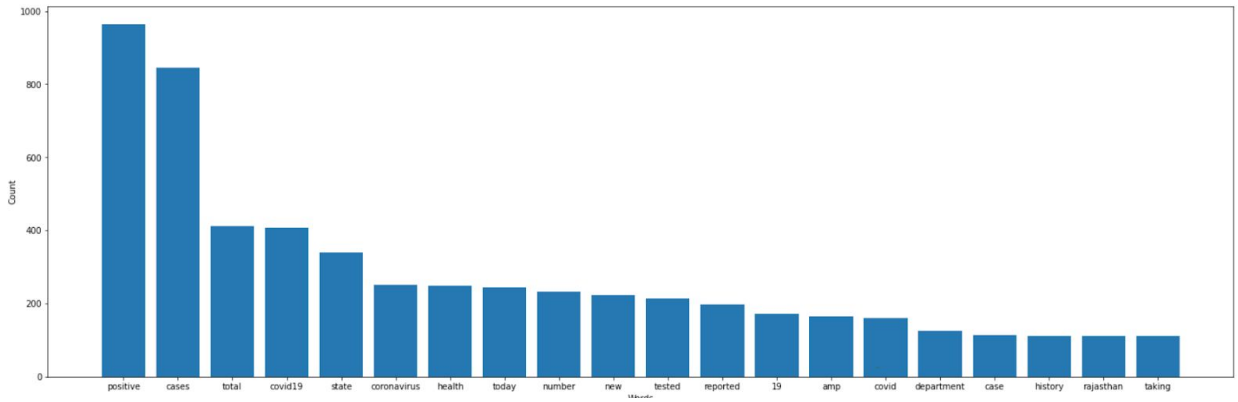
Out[254]:



The following are the top 20 words which appeared in the texts.

In [160]:
```
mostFreqWords={}
for word in topTenWords:
    mostFreqWords[word[0]]=word[1]
mostFreqWords
```

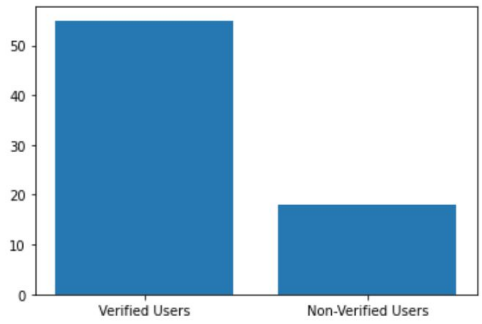
Out[160]:
```
{'positive': 965.0,
 'cases': 845.0,
 'total': 412.0,
 'covid19': 407.0,
 'state': 339.0,
 'coronavirus': 252.0,
 'health': 250.0,
 'today': 244.0,
 'number': 232.0,
 'new': 223.0,
 'tested': 214.0,
 'reported': 197.0,
 '19': 171.0,
 'amp': 165.0,
 'covid': 160.0,
 'department': 126.0,
 'case': 113.0,
 'history': 112.0,
 'rajasthan': 112.0,
 'taking': 111.0}
```



## Question 5

The number of users that tweeted about coronavirus were around 73. Out of these 73 users **55 were verified** and **18 were unverified**.

```
In [71]: ver=[]
         nonver=[]
         for i in users.keys():
             user=api.get_user(i)
             if user.verified:
                 ver.append(user)
             else:
                 nonver.append(user)
```

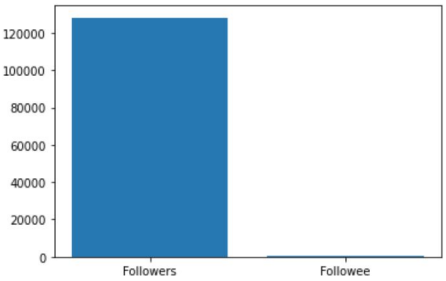



We picked 5 unverified users to analyse their profiles and activities. The following are the quantitative measures used to check if their profiles are credible or not:

1. Followers vs Friends
2. Number of posts in last month
3. Number of retweets vs original tweets
4. Number of tweets with and without media
5. Total number of retweets whose original authors are verified

**User#1**
screen_name: **kansalrohit69**
id: **731117568856142592**

1. Follower vs Followee

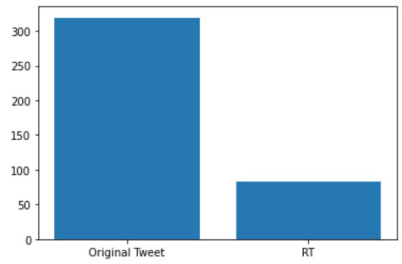



The number of followers are much larger as compared to the number of people this user follows, and more than 120k people follow him. This shows that people have trust in him. This could be an indication that the user is credible.

2. Number of tweets in last month
   The user had only 80 tweets in the last month. This shows that the user does not spam the readers with huge amount of tweets.

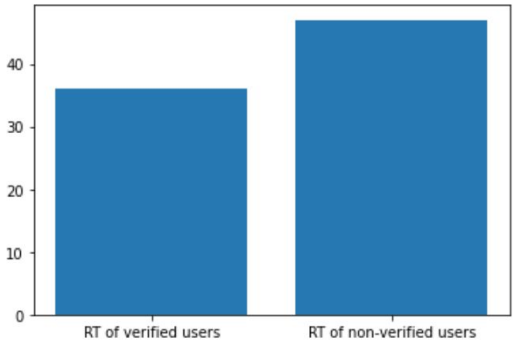
```
In [166]: tweetsInLastMonth=0
          for tweet in tweepy.Cursor(api.user_timeline,id = user.id,tweet_mode="extended").items():
              if tweet.created_at>=startDate and tweet.created_at<=endDate:
                  tweetsInLastMonth+=1
              if tweet.created_at<startDate:
                  break
          tweetsInLastMonth

Out[166]: 80
```

3. Number of retweets vs Original tweets



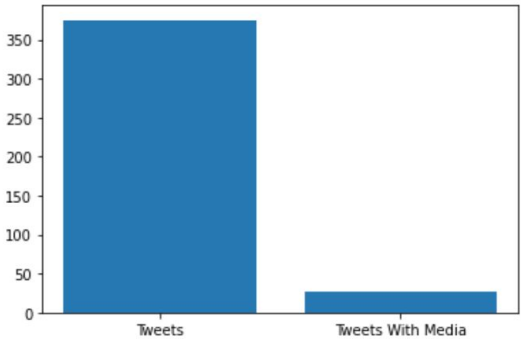

The user had most of his tweets as his own original tweets rather than a retweet. Moreover, analysing his profile manually, it appeared that most tweets and retweets are precise content and based on actual information and numbers.

4. Total number of retweets whose original authors are verified
   The user had almost 50% of his retweets posted by verified users.




   From this we could infer that the information the user is sharing is credible.

5. Number of tweets with and without media



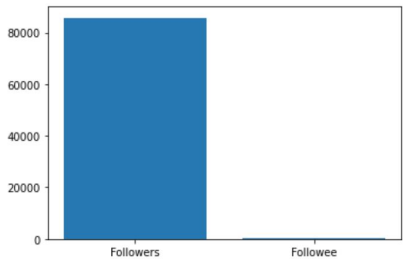

   Most of the tweets the user had were without media as most tweets revolved around precise information based on statistics

Verdict: Analysing the above criterias and manually checking, this user could be considered credible

**User#2**
screen_name: **diprjk**
id: **830669077022531584**


This twitter handle belonged to the Department of Information and Public Relations, Govt. of J&K. So this user is highly likely to be credible.

1. Follower vs Followee

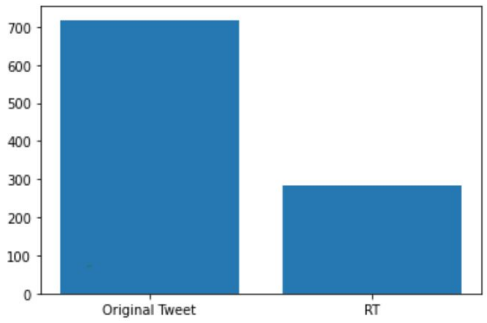



Similar to user#1, this user had quite a huge ratio for followers:followee. This shows that people have trust on the content shared by this profile hence the huge engagement.

2. Number of tweets in last month
   The user had 246 tweets in the last month.

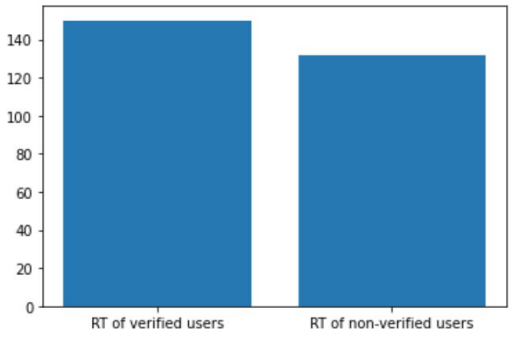
```
In [168]: tweetsInLastMonth=0
          for tweet in tweepy.Cursor(api.user_timeline,id = user.id,tweet_mode="extended").items():
              if tweet.created_at>=startDate and tweet.created_at<=endDate:
                  tweetsInLastMonth+=1
              if tweet.created_at<startDate:
                  break
          tweetsInLastMonth

Out[168]: 246
```

3. Number of retweets vs Original tweets

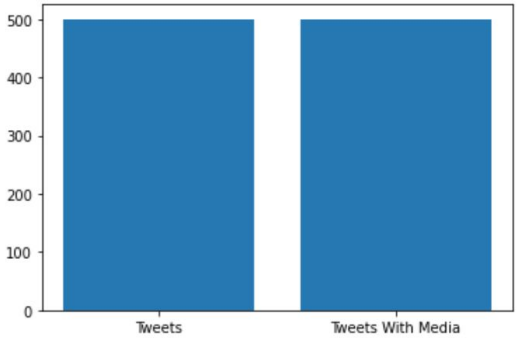



The user has more than 70% its tweets as original tweets. The content posted in tweets are based on information regarding J&K.

4. Total number of retweets whose original authors are verified:
   From all the retweets, more than 50% of the retweets were of verified user. This could be an indication towards the profile being credible.



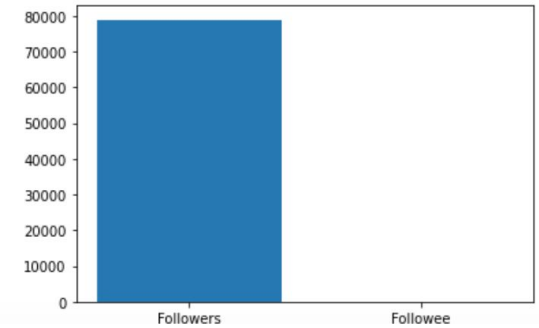
5. Number of tweets with and without media
   50% of the tweets had media in them. This is because they provide information regarding relief from this pandemic, and include images to help support the cause.




**User#3**
screen_name: **TelanganaHealth**
id: **992966051657756672**


This twitter handle belonged to the Ministry of Health Telangana State. So this user is highly likely to be credible.

1. Follower vs Followee



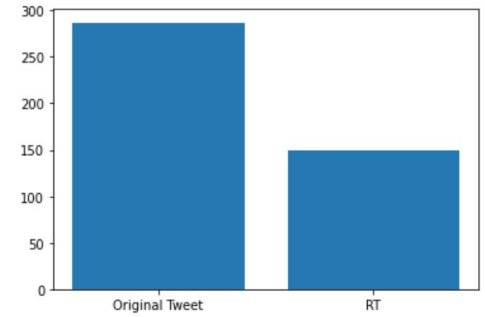

This user had quite a huge ratio for followers:followee. This shows that people have trust on the content shared by this profile hence the huge engagement.

2. Number of tweets in last month

```
In [170]: tweetsInLastMonth=0
          for tweet in tweepy.Cursor(api.user_timeline,id = user.id,tweet_mode="extended").items():
              if tweet.created_at>=startDate and tweet.created_at<=endDate:
                  tweetsInLastMonth+=1
              if tweet.created_at<startDate:
                  break
          tweetsInLastMonth

Out[170]: 61
```

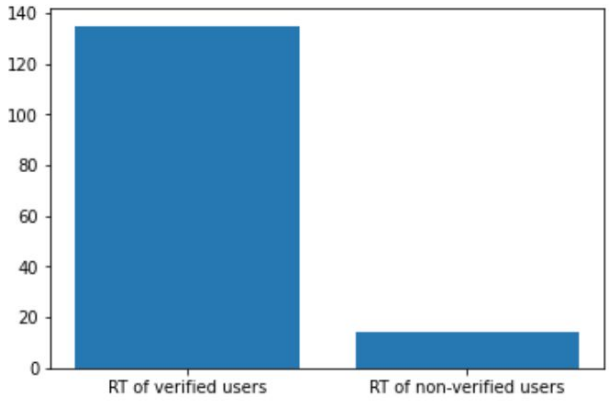
The user had 61 tweets in the last month thus averaging 2 tweets per day, which is very low considering the traffic twitter receives. This shows that only relevant and important information is being shared on the profile.

3. Number of retweets vs Original tweets



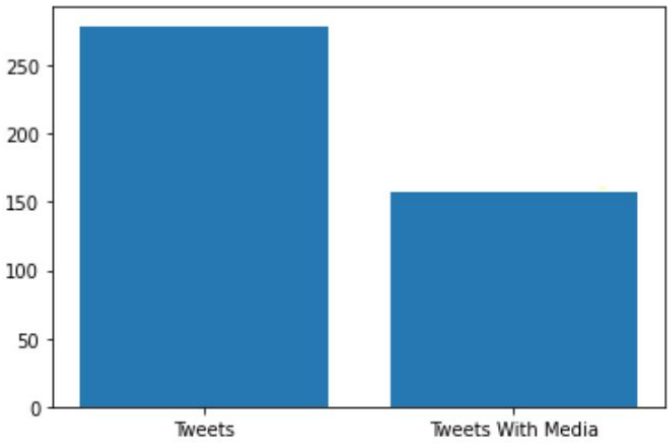

The profile has greater number of original tweets in comparison to retweets

4. Total number of retweets whose original authors are verified:




Most of the tweets were of verified users. This shows the credibility of the tweets posted by this user.

5. Number of tweets with and without media

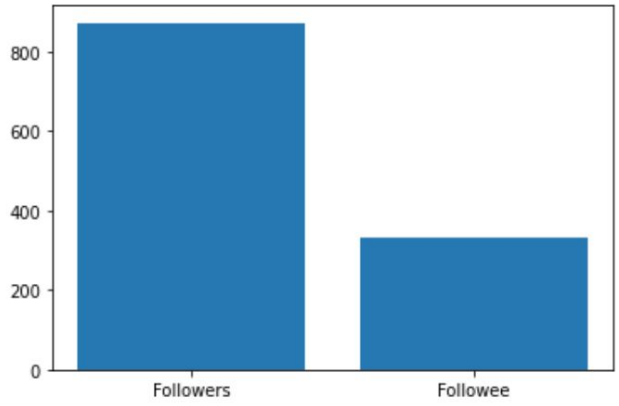



The number of tweets with media are less than tweets without them. Although upon analysing the profile manually, the tweets with media are mostly about some notices and information, which further solidifies its credibility.

**User#4**
screen_name: **wilson_thehindu**
id: **3602209398**


This profile belongs to a journalist who works at The Hindu.

1. Followers vs Friends



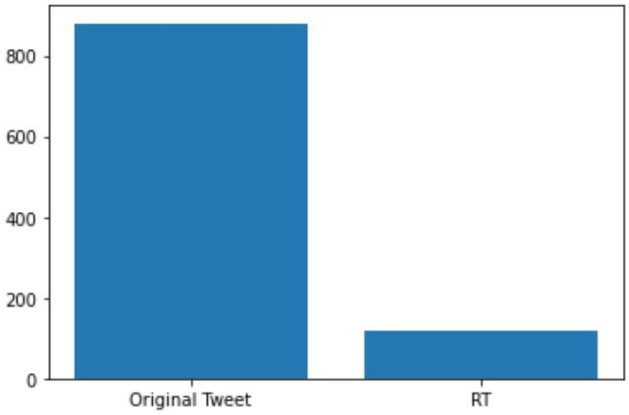

The account has more followers than followee but the number of followers are not many. This might not be a good indication of it being credible.

2. Number of posts in last month

```
In [172]: tweetsInLastMonth=0
          for tweet in tweepy.Cursor(api.user_timeline,id = user.id,tweet_mode="extended").items():
              if tweet.created_at>=startDate and tweet.created_at<=endDate:
                  tweetsInLastMonth+=1
              if tweet.created_at<startDate:
                  break
          tweetsInLastMonth

Out[172]: 105
```

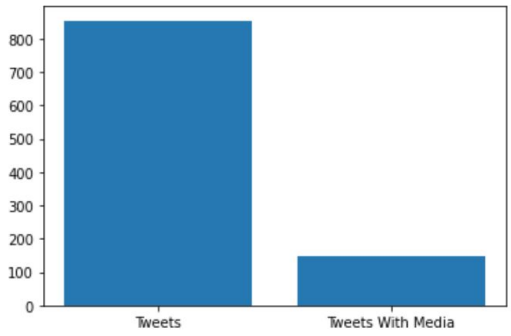
The user had 105 posts in the last month.

3. Number of retweets vs original tweets



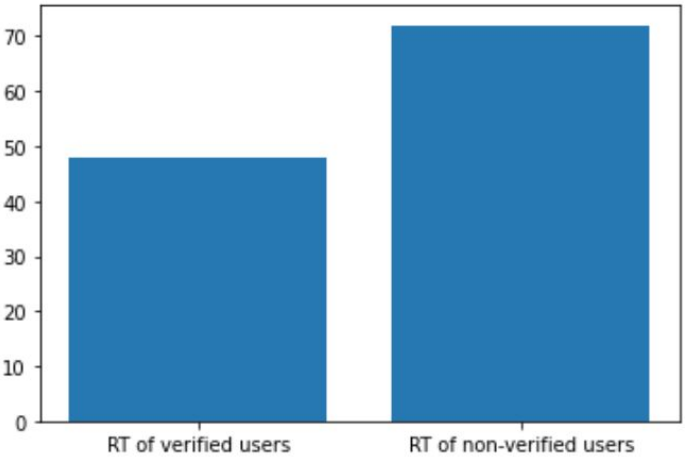

More than 80% of the tweets by the user were original tweets.

4. Number of tweets with and without media




Most tweets did not have media in them. This might be an indication of information not being credible.

5. Total number of retweets whose original authors are verified

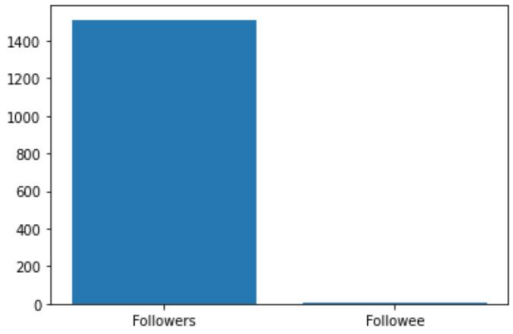



Out of all the retweets, most retweets did not have their original authors as verified.

Verdict: The account seems to be credible, but there is no assurity of it.

**User#5**
screen_name: **CollectorDnh**
id: **1240909753359851520**


1. Followers vs Friends



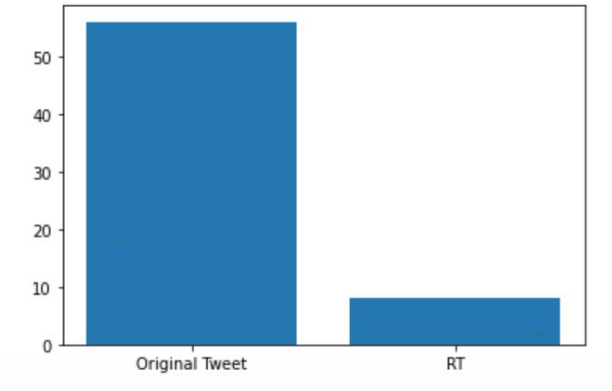

The account has more followers than followee but the number of followers are not many. This might not be a good indication of it being credible.

2. Number of posts in last month

```
In [174]: tweetsInLastMonth=0
          for tweet in tweepy.Cursor(api.user_timeline,id = user.id,tweet_mode="extended").items():
              if tweet.created_at>=startDate and tweet.created_at<=endDate:
                  tweetsInLastMonth+=1
              if tweet.created_at<startDate:
                  break
          tweetsInLastMonth

Out[174]: 41
```

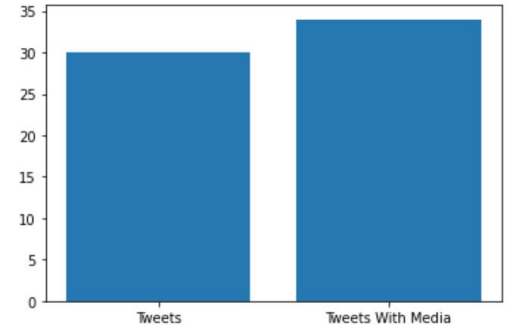
The user has only 41 posts in the last month, which shows that the user does not share much information on the twitter handle.

3. Number of retweets vs original tweets



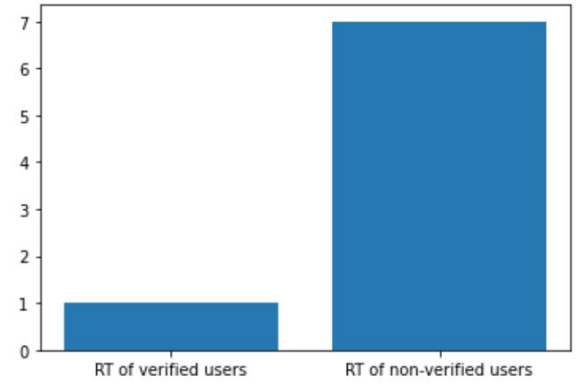

Most of the tweets are original tweets.

4. Number of tweets with and without media




More tweets had media in them compared to tweets without media. This might be a good sign to analyse the credibility of the user. Most media shared were posters and notices regarding coronavirus.

5. Total number of retweets whose original authors are verified




The retweets were mostly of non-verified users.

Verdict: The user may or may not be credible. It has more inclination towards not being credible.

## Question 6

The following PII were collected using **spacy** library and regular expressions.

1. **Human names**

```
In [50]: names={}
         for tweet in tweets:
             doc = nlp(tweet.full_text)
             human=[]
             for ent in doc.ents:
                 if ent.label_=="PERSON":
                     print(ent.text,ent.label_)
                     human.append(ent.text)
             if human:
                 names[tweet.id]=human
```

```
1244518226467467265: ['Edappadi K. Palaniswami'],
1248812492005965824: ['Nitin Madan Kulkarni'],
1250071116405760007: ['Nitin Madan Kulkarni'],
1245377181057835008: ['Tablighi Jamaat'],
1239894891678724097: ['Eatala Rajendra'],
1248021661170970625: ['Nitin Madan Kulkarni'],
1249322012058644480: ['Amit Mohan Prasad'],
1243831255470190592: ['Rohit Kansal'],
```

There were some false positives but most of the names were correctly identified.

2. **Places**

```
places={}
for tweet in tweets:
    doc = nlp(tweet.full_text)
    pl=[]
    for ent in doc.ents:
        if ent.label_=="GPE":
            print(ent.text,ent.label_)
            pl.append(ent.text)
    if pl:
        places[tweet.id]=pl
```

The following code was used to find out the places , cities, and states discussed in the tweet.

```
Rajasthan GPE
Banswara GPE
Bharatpur GPE
Dausa GPE
Jaipur GPE
Jodhpur GPE
Kota GPE
Kapurthala GPE
Bharatpur GPE
Bhilwara GPE
Bikaner GPE
Jaipur GPE
Jaisalmer GPE
Jhunjhunu GPE
Jodhpur GPE
Kota GPE
Sawai Madhopur GPE
```

### 3. Nationality

```
In [53]: groups={}
         for tweet in tweets:
             doc = nlp(tweet.full_text)
             gr=[]
             for ent in doc.ents:
                 if ent.label_=="NORP":
                     print(ent.text,ent.label_)
                     gr.append(ent.text)
             if gr:
                 groups[tweet.id]=gr
```

The following code was used to find the nationality of people that were being reported.
The following nationalities were found.

```
Italian NORP
Indonesian NORP
Indian NORP
Indian NORP
Rajasthan NORP
Malaysian NORP
Swiss NORP
Italian NORP
Italian NORP
Rajasthan NORP
Thai NORP
Italian NORP
Coronovirus NORP
Pimpri-Chinchwad NORP
Shillong NORP
Bomikhal NORP
Shopian NORP
Rajasthan NORP
Pathanamthitta NORP
Thai NORP
```

There were still some false positives that were included in the output.

4. **Mobile numbers**

   Mobile numbers were extracted using the following regex.
   There was only one mobile number which was found, which was a helpline number.

```
In [57]: mobile_numbers={}
         for tweet in tweets:
             number=re.findall('(?:\s+|)((0|(?:(\+|)91))(?:\s|-)*(?:(?:\d(?:\s|-)*\d{9})|(?:\d{2}(?:\s|-)*\d{8})|(?:\d{3}(?:\s|-
             real_num=[]
             for num in number:
                 for val in num:
                     if len(val)>=8:
                         real_num.append(val)
             if len(number)>0:
                 mobile_numbers[tweet.id]=real_num
```

```
{1245265375626842113: ['080-29711171']}
```

5. **Religious and Other Organisations**

```
org={}
for tweet in tweets:
    doc = nlp(tweet.full_text)
    label=[]
    for ent in doc.ents:
        if ent.label_=="ORG":
            print(ent.text,ent.label_)
            label.append(ent.text)
    if label:
        org[tweet.id]=label
```

Different organisations like Public Health and Family Welfare, Directorate of Health Services etc. were collected.

```
Public Health and Family Welfare ORG

Public Health Department ORG

Maharashtra Health Department ORG
```

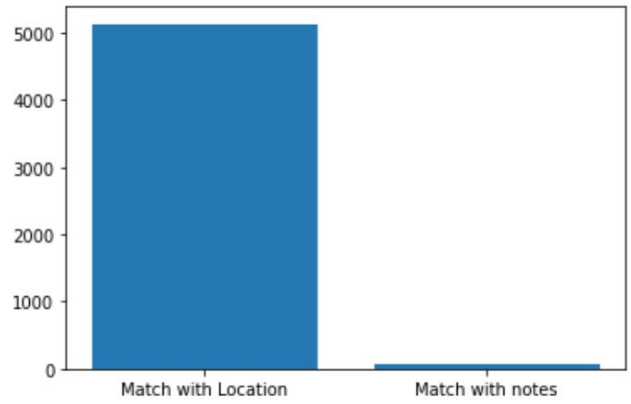
## Question 7

For each row of the data, out of the three sources, the twitter links were collected and the tweet ID were extracted out of these links.
Using **spacy** library, locations were collected from the **Backup Notes** column.

Similarly, locations were taken from the **Detected District, Detected City and Detected State** columns.

For the rows containing twitter links, the full_text of the tweet was taken and using **spacy** library, locations were collected and compared to the locations in **Detected District, Detected City and Detected State** columns and the **Backup Notes** column.
Following was the result:

```
In [236]: print(locationMatch,notesMatch)

          5130 72
```




Most of the notes columns were empty which was quite evident through the results. Only 361 out of 17000 rows had non null values in the **Backup Notes** column.

```
In [243]: count=0
          for ind,row in data.iterrows():
              if isinstance(row['Backup Notes'],str):
                  count+=1
          count

Out[243]: 361
```

Most of the tweets had their locations matched to the detected location columns.

## Question 8

```
users={}
maxTweets=0
userID=0
for tweet in tweets:
    user_id=tweet.user.id
    if user_id in users:
        users[user_id]+=1
    else:
        users[user_id]=1
    if users[user_id]>maxTweets:
      | maxTweets=users[user_id]
        userID=user_id
```

The number of tweets of each distinct userID was calculated and were sorted according to the number of tweets.

The following twitter ID had the maximum amounts of tweets.

Twitter ID: **355989081**
Screen_Name: **ANI**

The user with scree_name **ANI** is **verified**.