HW6 - Analyzing Disinformation Domains

Adeniran Adeniyi

Sunday, April 4, 2021 by 11:59pm

Q1

Datasets D1 and D2 include the number of tweets that each domain was shared in (found in the last column/field of the dataset).

For each of D1 and D2, what were the top 10 domains in terms of tweets? Which ones are no longer live? Load the main web-page in your web browser. How would you classify the domain? Show this information in a table like the one below, sorted by number of tweets. You should have 2 tables, one for D1 and one for D2.

Answer

```
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 31 06:45:41 2021

@author: aadeniran
"""
import pandas as pd
import numpy as np
#read the files in pandas dataframe
dfile1= pd.read_csv("D1.csv")
dfile2 = pd.read_csv("D2.csv")

#sort them in order
dfile1 = dfile1.sort_values(by='# Citations in our Alternative
   Narrative Tweets',ascending=False)
dfile2 = dfile2.sort_values(by='Tweet count', ascending=False)

search = dfile1.copy()

#get only 10 items
dfile1 = dfile1.head(10)
dfile2 = dfile2.head(10)


#drop unwanted columns
dfile1.drop(['Primary Orientation (Determined through Content Analysis)
   ', 'How Cited in Alternative Narrative of Shooting Events'],axis= 1,
```

```
   inplace=True)
dfile2.drop(['URL count'],axis =1, inplace=True)


#rename columns
dfile1.rename(columns={"# Citations in our Alternative Narrative Tweets
   ":"Tweets","Media Type (Determined through Content Analysis)":"
   Website Type"},inplace=True)
dfile2.rename(columns={"Tweet count":"Tweet"},inplace=True)

#swap order for first d1
columns_swap = ["Domain","Tweets","Website Type"]
dfile1 = dfile1.reindex(columns=columns_swap)



#add new columns to the data with NAN values
dfile1['status']= np.nan
dfile2['Website Type']= np.nan
dfile2['status']= np.nan


#change column types to string
dfile2['Website Type'] = dfile2['Website Type'].astype(str)
dfile2['status'] = dfile2['status'].astype(str)
#print(dfile2.dtypes)

numCount = 0
temp =""
 #Match domains in top 10 D2 dataframe with D1 to obtain Website Media
   Type
for index, row in dfile2.iterrows():
    #find a match(es) and store as a dataframe
    temp = search[search['Domain'].str.contains(row['Domain'])]
    #check if data frame is empty
    if(len(temp) == 0):
        #assign NaN value
        final = np.nan
    else:
        #assigne Media Type to final value
        final = temp['Media Type (Determined through Content Analysis)'
   ].iloc[0]
    #insert into dfile2 dataframe
    dfile2.at[index,"Website Type"] = final

dfile1.to_csv("Q1/D1processed.csv", index = False, header=True)
dfile2.to_csv("Q1/D2processed.csv", index = False, header=True)
```

```
68
69
70 #print(type(search[search['Domain'].str.contains("21stcenturywire.com")
       ].iloc[0]))
71 #print(dfile1)
```

Listing 1: question1.py

Table 1: Top 10 High Number of Tweets Domains (Processed D1)

Domain	Tweets	Media	Status
therealstrategy.com	7113	Alternative Media	not live
infowars.com	1741	Alternative Media	live
newsbusters.org	1217	Alternative Media	live
washingtonpost.com	1108	MSM	live
nodisinfo.com	774	Alternative Media	not live
nytimes.com	759	MSM	live
veteranstoday.com	586	Alternative Media	live
beforeitsnews.com	580	Alternative Media	live
rawstory.com	308	Alternative Media	live
hoax.trendolizer.com	299	fact checker	live

Table 2: Top 10 High Number of Tweets Domains (Processed D2)

Domain	Tweets	Media	Status
21stcenturywire.com	3088	Alternative Media	live
clarityofsignal.com	2352	Not Found(Alternative Media)	live
rt.com	1598	Foreign Government Media	live
newsweek.com	1249	Not Found(MSM)	live
alternet.org	1221	Not Found(Alternative Media)	live
sputniknews.com	1076	Foreign Government Media	live
mintpressnews.com	919	Not Found(Alternative Media)	live
cnn.com	756	MSM	live
globalresearch.ca	724	Alternative Media	live
theantimedia.org	682	Alternative Media	live

Discussion

For D1processed I read the D1.csv file in a pandas dataFrame and was able to easily to sort the data according to The number of Tweets in Highest to lowest. As for processed D2 I did the same thing but I read in D2.csv and also compared the data with D1.csv to obtain the Media Values. Those values that are not found shows not found and what i manually classified them. I also manual checked the website on the browser to check if the where active website.

Q2

Compare the amount of overlap between the three datasets. Generate the following datasets:

- a. domains that are present in both D1 and D2
- b. domains that are present in both D2 and D3
- c. domains that are present in both D1 and D3
- d. domains that are present in all three datasets

Answer

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr  4 12:20:50 2021

@author: aadeniran
"""

import pandas as pd

#read the files in pandas dataframe
dfile1= pd.read_csv("D1.csv")
dfile2 = pd.read_csv("D2.csv")
dfile3 = pd.read_csv("D3.csv")

def compareThisB(lowerCase,upperCase):
    #create an empty final dataframe
    number = 0
    column_name = ["Domain"]
    final = pd.DataFrame(columns = column_name)
    for index, row in upperCase.iterrows():
        #find a match(es) and store as a dataframe
```

```
22          #set Uppercases domain to lowercase so that it can propermatch
23          temp = lowerCase[lowerCase['Domain'] == row['Domain'].lower()]
24          #print(dfile1[dfile1['Domain'] == row['Domain']])
25          #check if data frame is empty
26          if(len(temp) == 0):
27              #assign NaN value
28              #final = np.nan
29              pass
30          else:
31          #    print(temp)
32              #assigne Media Type to final value
33              final.at[number,"Domain"] = temp['Domain'].iloc[0]
34              number +=1
35      return final
36
37
38  """
39      I could literally turn it to a function smh
40     Q2 Part A compare D1 and D2
41  """
42  finalA = compareThisB(dfile1,dfile2)
43
44  """
45     Q2 Part B compare D2 and D3
46  """
47  finalB = compareThisB(dfile2,dfile3)
48  """
49     Q2 Part C compare D1 and D3
50  """
51  finalC = compareThisB(dfile1,dfile3)
52
53  """
54     Q2 Part D compare finalA  and D3
55  """
56  finalD = compareThisB(finalA,dfile3)
57
58
59  #convert to csv files
60  finalA.to_csv("Q2/finalA.csv", index = False, header=True)
61  finalB.to_csv("Q2/finalB.csv", index = False, header=True)
62  finalC.to_csv("Q2/finalC.csv", index = False, header=True)
63  finalD.to_csv("Q2/finalD.csv", index = False, header=True)
```

Listing 2: overlap.py for question 2

Table 3: Domains that are present in both D1 and D2 (gotten from finalA)

Domain
rt.com
breitbart.com
theeventchronicle.com
therussophile.org
themillenniumreport.com
beforeitsnews.com
cbsnews.com
thefreethoughtproject.com
veteranstoday.com
theintercept.com
theguardian.com
21stcenturywire.com
infowars.com
thedailybeast.com
heavy.com
blacklistednews.com
presstv.com
dcclothesline.com
theantimedia.org
upi.com
investmentwatchblog.com
dailymail.co.uk
mirror.co.uk
nydailynews.com
fellowshipoftheminds.com
thetruthseeker.co.uk
abovetopsecret.com
cnn.com
worldtruth.tv
sputniknews.com
lewrockwell.com
nytimes.com
intellihub.com
thedailysheeple.com
globalresearch.ca
foxnews.com
thestar.com
activistpost.com
nbcnews.com

Table 4: Domains that are present in both D2 and D3 (gotten from finalB.csv)

Domain
activistpost.com
beforeitsnews.com
breitbart.com
collective-evolution.com
dcclothesline.com
gellerreport.com
humansarefree.com
infowars.com
intellihub.com
ronpaulinstitute.org
sott.net
thewashingtonstandard.com
worldtruth.tv
21stcenturywire.com
davidicke.com
off-guardian.org
presstv.com
ukcolumn.org
rubikon.news
globalresearch.ca
theduran.com

Table 5: Domains that are present in both D1 and D3 (gotten from finalC.csv)

Domain
activistpost.com
beforeitsnews.com
breitbart.com
dcclothesline.com
infowars.com
intellihub.com
wakingtimes.com
worldtruth.tv
zerohedge.com
21stcenturywire.com
presstv.com
globalresearch.ca

Table 6: domains that are present in all three datasets (gotten from finalD.csv)

Domain
activistpost.com
beforeitsnews.com
breitbart.com
dcclothesline.com
infowars.com
intellihub.com
worldtruth.tv
21stcenturywire.com
presstv.com
globalresearch.ca

Discussion

I created a function called compareThisB(lowerCase,upperCase) to handle all the process for comparisons, for a,b,c,and d. This functions takes the two parameters. The second parameters get converted in all lower case before comparisons.

Q3

Collect at least 200 tweets that contain links from domains that appear in both D3 and one of the other datasets (so, dataset b or dataset c from Q2).

Answer

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr  4 14:57:42 2021

@author: aadeniran
"""
import tweepy
import pandas as pd
```

```
import json
#will use finalB
finalB =pd.read_csv("Q2/finalB.csv")
print(finalB)

# OAuth2 procedure
consumer_key = "pnUItdX31QmYpHBFlVcYbocKQ"      # INSERT YOUR KEY HERE
consumer_secret = "gFNX2iztwhfL1tROFCX3UomwRbU8GjUJhHzLQat8DGxvBcyVmw"
     # INSERT YOUR KEY HERE
auth = tweepy.AppAuthHandler(consumer_key, consumer_secret)
api = tweepy.API(auth,wait_on_rate_limit=True)

collection = []

debug = 0
for index, row in finalB.iterrows():

    search_term = row['Domain']
    for page in tweepy.Cursor(api.search, q=search_term, tweet_mode='
   extended', lang='en').pages():

        for tweet in page:
            print("Tweet  \nnnnn")
            #ensure that links are present ran 29 of each
            if(len(tweet.entities["urls"]) > 0 and debug < 30):
                #Get all the required details
                dictP = {"Id":tweet.id_str,"Screen_name":tweet.user.
   screen_name,"Date":tweet.created_at.strftime("%Y%m%d%H%M%S"),"
   Search_Term":search_term,"Full_text":tweet.full_text,"Url":tweet.
   entities["urls"][0]["expanded_url"]}
                collection.append(dictP)
                debug +=1
            else:
                break
        if(debug > 29):
            debug =0
            break
final = pd.DataFrame(collection)
print(final)
final.to_csv("Q3/collection.csv", index = False, header=True)
#re read the file and get into right time format
parse = lambda x: pd.datetime.strptime(x,"%Y%m%d%H%M%S")
process = pd.read_csv("Q3/collectionR.csv",parse_dates=["Date"],
   date_parser=parse)
#convert to the real one collectionR
process.to_csv("Q3/collectionR.csv", index = False, header=True)
#used this website to convert csv to json
```

```
"""
https://csvjson.com/csv2json
"""
```

Listing 3: collecting-tweet.py used for collecting tweet info per domain

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr  4 18:59:18 2021

@author: aadeniran
"""
import pandas as pd
from datetime import datetime
import matplotlib.pyplot as plt
#read the json file
process = pd.read_json("Q3/csvjson.json")



"""
How many Tweet was gathered
"""
#answer 450
print(process.shape[0])
#Answer 450 tweets gathered
"""
Q3
What was the time range of the tweets gathered
"""
maxx= process['Date'].min()
minn = process['Date'].max()

date_range = str(minn) + ' to ' +str(maxx)

#print(date_range)
#range from 2021-04-04 22:21:48 to 2021-03-27 20:25:22
"""
Q3
How many different accounts posted those tweets?
"""
accountNames = []
#put unique user name as a list
accountNames = process["Screen_name"].unique()
#convert to pandas data frame and dump as csv file

t = pd.DataFrame(accountNames, columns=["Unique Screen names"])
t.to_csv("Q3/unqiuer_screen_names.csv",index=False)
```

```
print(len(accountNames))
#answer 306 different  user
"""
Q3
For those domains that had at least one tweet, how many tweets did you discover for each domain?
To answer this question, create a bar chart showing the number of tweets for each domain.
https://stackoverflow.com/questions/17573814/count-occurrences-of-certain-words-in-pandas-dataframe
"""

#get all unique domains
#activistpost = process["Url"].str.contains("www.activistpost.com").sum()
print(process.loc[process["Url"].str.contains("www.activistpost.com")])
#try
#print(accountNames)

# Q3get the list of domains and search and get count
names =[]
totalTweet = []
#Q4 get the unique Screen_name count per domain
uniqunamesCount =[]

d2 = pd.read_csv("Q2/finalB.csv")
#print(d2)
for index, row in d2.iterrows():
    #print(row["Domain"])
    names.append(row["Domain"])
    totalTweet.append(process["Url"].str.contains(row["Domain"]).sum())

    """
    Do Q4 as well gettings the unique names per domain
    """
    #Get the list of unique domains
    getUnique = process.loc[process["Url"].str.contains(row["Domain"])]
    uniqunamesCount.append(getUnique["Screen_name"].unique().size)
    #print(process["Url"].str.contains(row["Domain"]).sum())
    #print("{}  total tweets {}".format(row["Domain"], process["Url"].str.contains(row["Domain"]).sum())
#print(process)
#print(names)
#print(totalTweet)

#plot the graph Q3 bar chart showing the number of tweets for each domain.
```

```
84
85 ax = plt.gca()
86 ax.yaxis.grid()
87 plt.rcParams["figure.figsize"] =(15,8)
88 plt.bar(names,totalTweet)
89 plt.xticks(rotation =90)
90 plt.savefig("Q3/barchar.png",bbox_inches='tight')
91 plt.plot()
92 plt.show()
93
94 """
95 Prepare data for Q4 (Top Ten shared domains in Q3)
96 arrange the values from highest to lowest
97 """
98 d = {"Domain": names,"Total":totalTweet}
99 tweet = pd.DataFrame(d)
100 tweet.sort_values(by="Total",ascending=False,inplace=True)
101 #get the top ten
102 tweet.head(10)
103 tweet.to_csv("Q4/topTentweetPerDomain.csv",index=False)
104
105 #plot the graph Q4 bar chart showing the number of accounts for each
      domain.
106 ax = plt.gca()
107 ax.yaxis.grid()
108 plt.rcParams["figure.figsize"] =(15,8)
109 plt.bar(names,uniqunamesCount)
110 plt.xticks(rotation =90)
111 plt.savefig("Q4/barchar.png",bbox_inches='tight')
112 plt.plot()
113 plt.show()
```

Listing 4: processing.py for question 3 and 4 data processing

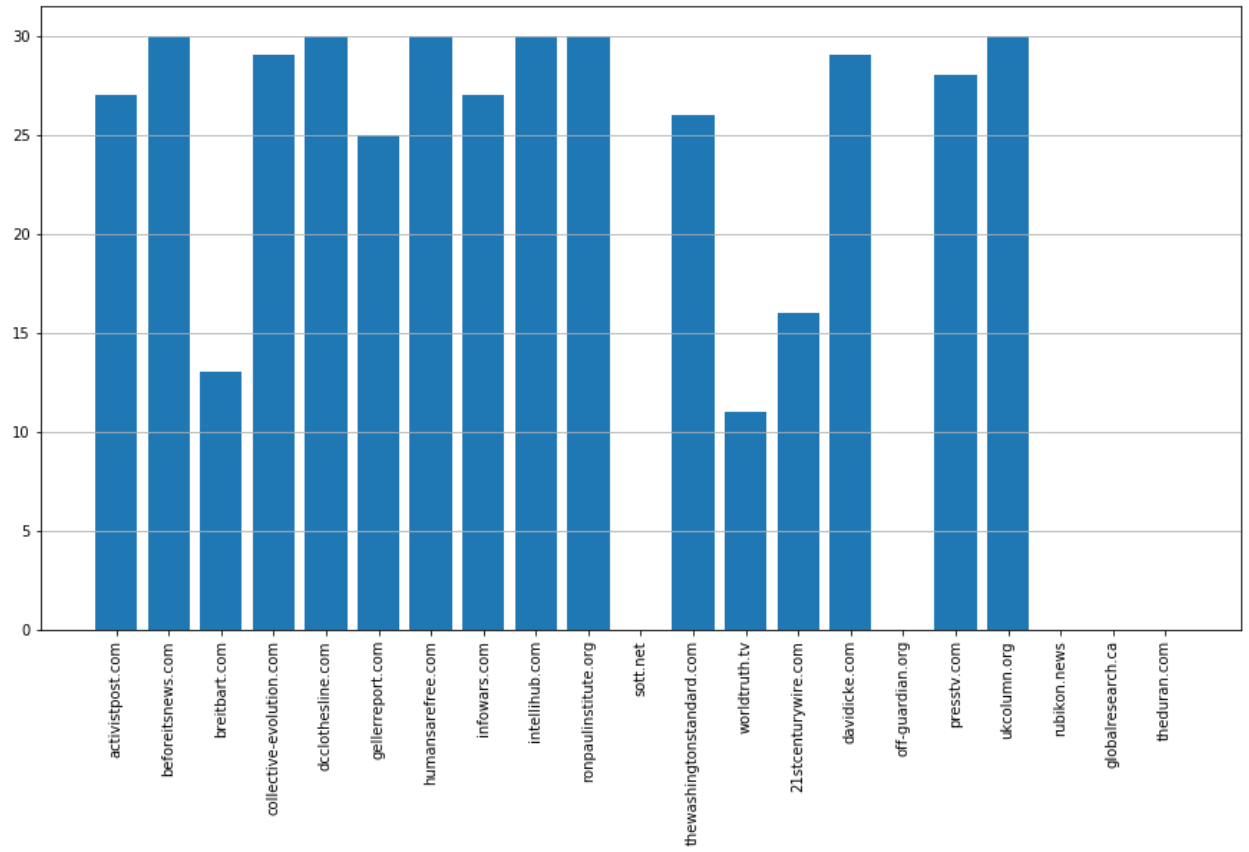

Figure 1: a bar chart showing the number of tweets for each domain

Discussion

Collect-tweet.py does the processing of collecting the tweet information using the lists of domain created in Question 2B(Q2/finalB.csv).

Using the processing.py script, I read the csvjson.json file into a pandas dataFrame, and using the dataFrame I was able to process the information to answer the questions below

- How many tweets did you gather?

  I was able to gather 450 tweets, I read the json converted csvjson.json file into pandas dataFrame. Using the .shape[0] gets the size of the dataFrame

- What was the time range of the tweets you gathered?

  The time range is from 2021-04-04 22:21:48 to 2021-03-27 20:25:22

- How many different accounts posted those tweets?

There are 306 different users, using the unique function to get all unique values in a column. It get the total unique screen names from the data I gathered(450).

- For those domains that had at least one tweet, how many tweets did you discover for each domain? To answer this question, create a bar chart showing the number of tweets for each domain. Answer in Table 1

Q4

Compare the number of tweets per domain from Q3 with the information in D1 and D2. What were the top 10 shared domains from Q3? How does this compare with the top 10 shared domains in D1 and D2 (from Q1)? For those domains that had at least one tweet, how many accounts were posting links for each domain?

To answer this question, create a bar chart showing the number of accounts for each domain.

Answer

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr  4 18:59:18 2021

@author: aadeniran
"""
import pandas as pd
from datetime import datetime
import matplotlib.pyplot as plt
#read the json file
process = pd.read_json("Q3/csvjson.json")



"""
How many Tweet was gathered
"""
#answer 450
print(process.shape[0])
#Answer 450 tweets gathered
"""
Q3
What was the time range of the tweets gathered
"""
maxx= process['Date'].min()
```

```
minn = process['Date'].max()

date_range = str(minn) + ' to ' +str(maxx)

#print(date_range)
#range from 2021-04-04 22:21:48 to 2021-03-27 20:25:22
"""
Q3
How many different accounts posted those tweets?
"""
accountNames = []
#put unique user name as a list
accountNames = process["Screen_name"].unique()
#convert to pandas data frame and dump as csv file

t = pd.DataFrame(accountNames, columns=["Unique Screen names"])
t.to_csv("Q3/unqiuer_screen_names.csv",index=False)
print(len(accountNames))
#answer 306 different  user
"""
Q3
For those domains that had at least one tweet, how many tweets did you
   discover for each domain?
To answer this question, create a bar chart showing the number of
   tweets for each domain.
https://stackoverflow.com/questions/17573814/count-occurrences-of-
   certain-words-in-pandas-dataframe
"""

#get all unique domains
#activistpost = process["Url"].str.contains("www.activistpost.com").sum
   ()
print(process.loc[process["Url"].str.contains("www.activistpost.com")])
#try
#print(accountNames)

# Q3get the list of domains and search and get count
names =[]
totalTweet = []
#Q4 get the unique Screen_name count per domain
uniqunamesCount =[]

d2 = pd.read_csv("Q2/finalB.csv")
#print(d2)
for index, row in d2.iterrows():
    #print(row["Domain"])
    names.append(row["Domain"])
```

```
    totalTweet.append(process["Url"].str.contains(row["Domain"]).sum())

    """
    Do Q4 as well gettings the unique names per domain
    """
    #Get the list of unique domains
    getUnique = process.loc[process["Url"].str.contains(row["Domain"])]
    uniqunamesCount.append(getUnique["Screen_name"].unique().size)
    #print(process["Url"].str.contains(row["Domain"]).sum())
    #print("{}  total tweets {}".format(row["Domain"], process["Url"].
   str.contains(row["Domain"]).sum())
#print(process)
#print(names)
#print(totalTweet)

#plot the graph Q3 bar chart showing the number of tweets for each
   domain.

ax = plt.gca()
ax.yaxis.grid()
plt.rcParams["figure.figsize"] =(15,8)
plt.bar(names,totalTweet)
plt.xticks(rotation =90)
plt.savefig("Q3/barchar.png",bbox_inches='tight')
plt.plot()
plt.show()

"""
Prepare data for Q4 (Top Ten shared domains in Q3)
arrange the values from highest to lowest
"""
d = {"Domain": names,"Total":totalTweet}
tweet = pd.DataFrame(d)
tweet.sort_values(by="Total",ascending=False,inplace=True)
#get the top ten
tweet.head(10)
tweet.to_csv("Q4/topTentweetPerDomain.csv",index=False)

#plot the graph Q4 bar chart showing the number of accounts for each
   domain.
ax = plt.gca()
ax.yaxis.grid()
plt.rcParams["figure.figsize"] =(15,8)
plt.bar(names,uniqunamesCount)
plt.xticks(rotation =90)
plt.savefig("Q4/barchar.png",bbox_inches='tight')
plt.plot()
```

```
113 plt.show()
```

Listing 5: processing.py for question 3 and 4 data processing

Table 7: The top 10 shared domains from Q3

Domain	Number Of Tweets
ronpaulinstitute.org	30
humansarefree.com	30
beforeitsnews.com	30
intellihub.com	30
ukcolumn.org	30
dcclothesline.com	30
collective-evolution.com	29
davidicke.com	29
presstv.com	28
infowars.com	27

Table 8: Top 10 High Number of Tweets Domains (Processed D1)

Domain	Tweets	Media	Status
therealstrategy.com	7113	Alternative Media	not live
infowars.com	1741	Alternative Media	live
newsbusters.org	1217	Alternative Media	live
washingtonpost.com	1108	MSM	live
nodisinfo.com	774	Alternative Media	not live
nytimes.com	759	MSM	live
veteranstoday.com	586	Alternative Media	live
beforeitsnews.com	580	Alternative Media	live
rawstory.com	308	Alternative Media	live
hoax.trendolizer.com	299	fact checker	live

Table 9: Top 10 High Number of Tweets Domains (Processed D2)

Domain	Tweets	Media	Status
21stcenturywire.com	3088	Alternative Media	live
clarityofsignal.com	2352	Not Found(Alternative Media)	live
rt.com	1598	Foreign Government Media	live

Continued on next page

Table 9 – continued from previous page

Domain	Tweets	Media	Status
newsweek.com	1249	Not Found(MSM)	live
alternet.org	1221	Not Found(Alternative Media)	live
sputniknews.com	1076	Foreign Government Media	live
mintpressnews.com	919	Not Found(Alternative Media)	live
cnn.com	756	MSM	live
globalresearch.ca	724	Alternative Media	live
theantimedia.org	682	Alternative Media	live

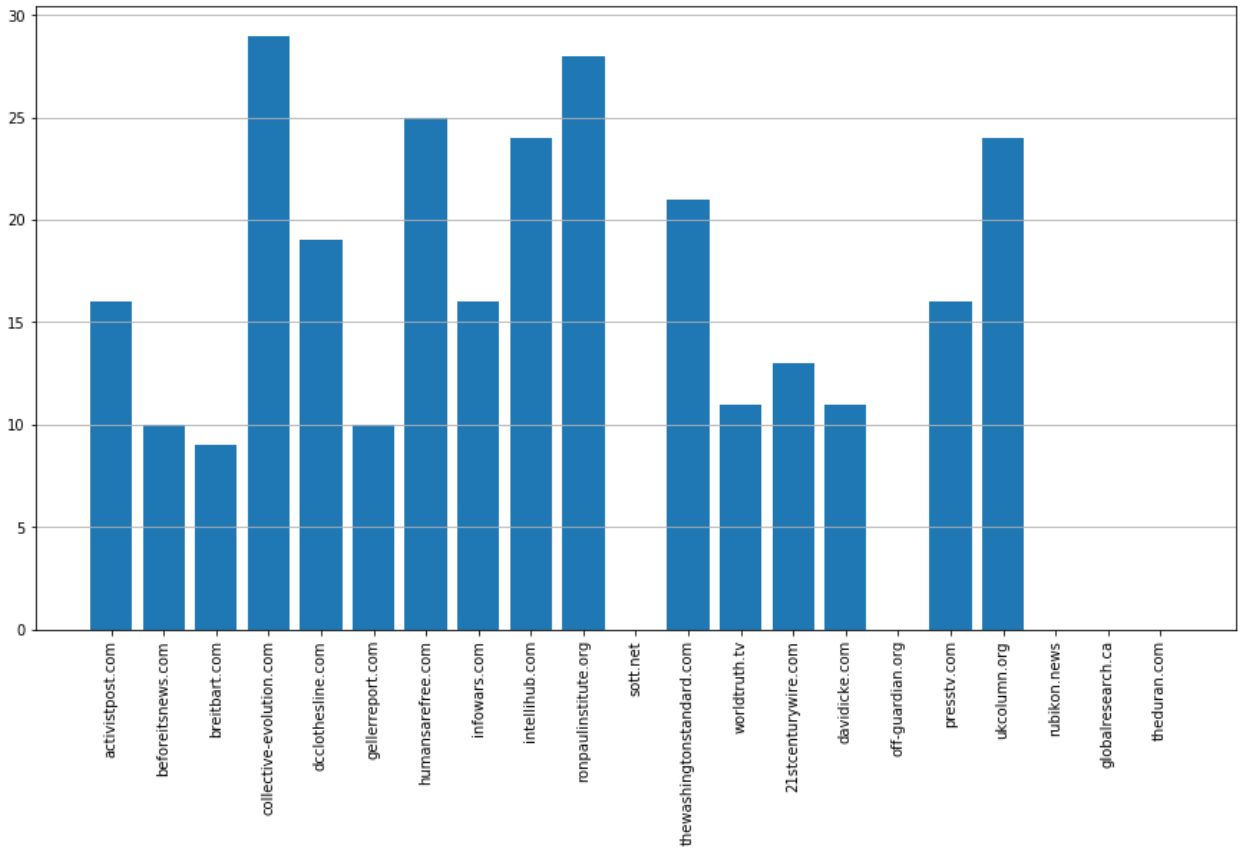

Figure 2: a bar chart showing the number of accounts for each domain

Discussion

I used processing.py to get the questions answered, more details are in processing.py I followed this instructions

- Compare the number of tweets per domain from Q3 with the information in D1 and D2. I used the code in processing I compared the from Q4/tweetPerDomain folders
- What were the top 10 shared domains from Q3?
  Answer in Table 7
- How does this compare with the top 10 shared domains in D1 and D2 (from Q1)?

  They had these domains in common:
  beforeitsnews.com (in D1)
  infowars.com (in D1)
  Nothing at all was common in D2
- For those domains that had at least one tweet, how many accounts were posting links for each domain?To answer this question, create a bar chart showing the number of accounts for each domain.

  Answer in Figure 2

Q5

There have been several online games created to educate people about disinformation and how it spreads on social media. Play one of the games at either https://www.getbadnews.com or https://goviralgame.com. Write a paragraph about your experience and some lessons you learned by playing the game. Take some screenshots as you play to include in your report.

Answer

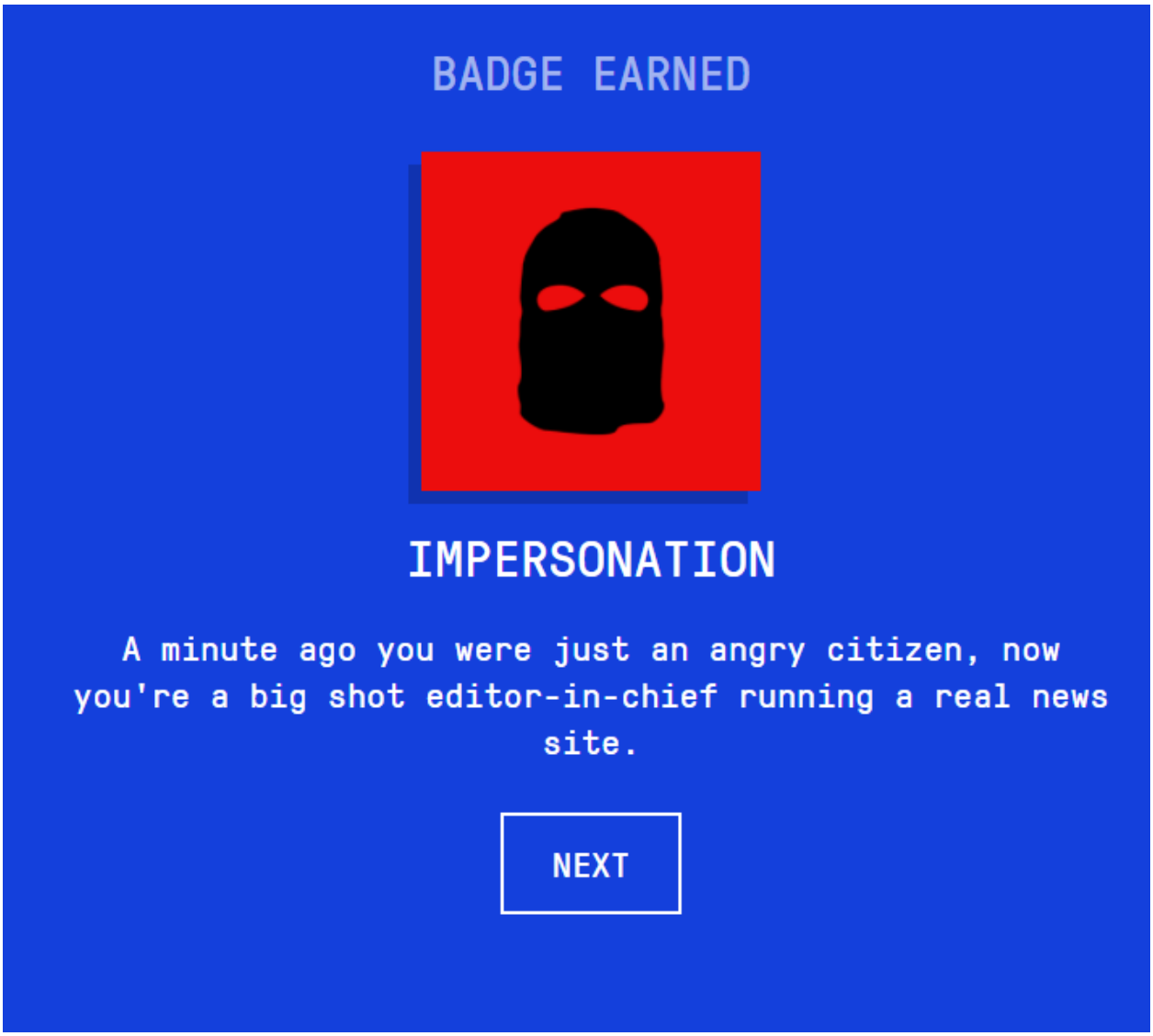

Figure 3: a bar chart showing the number of accounts for each domain

Figure 4: a bar chart showing the number of accounts for each domain

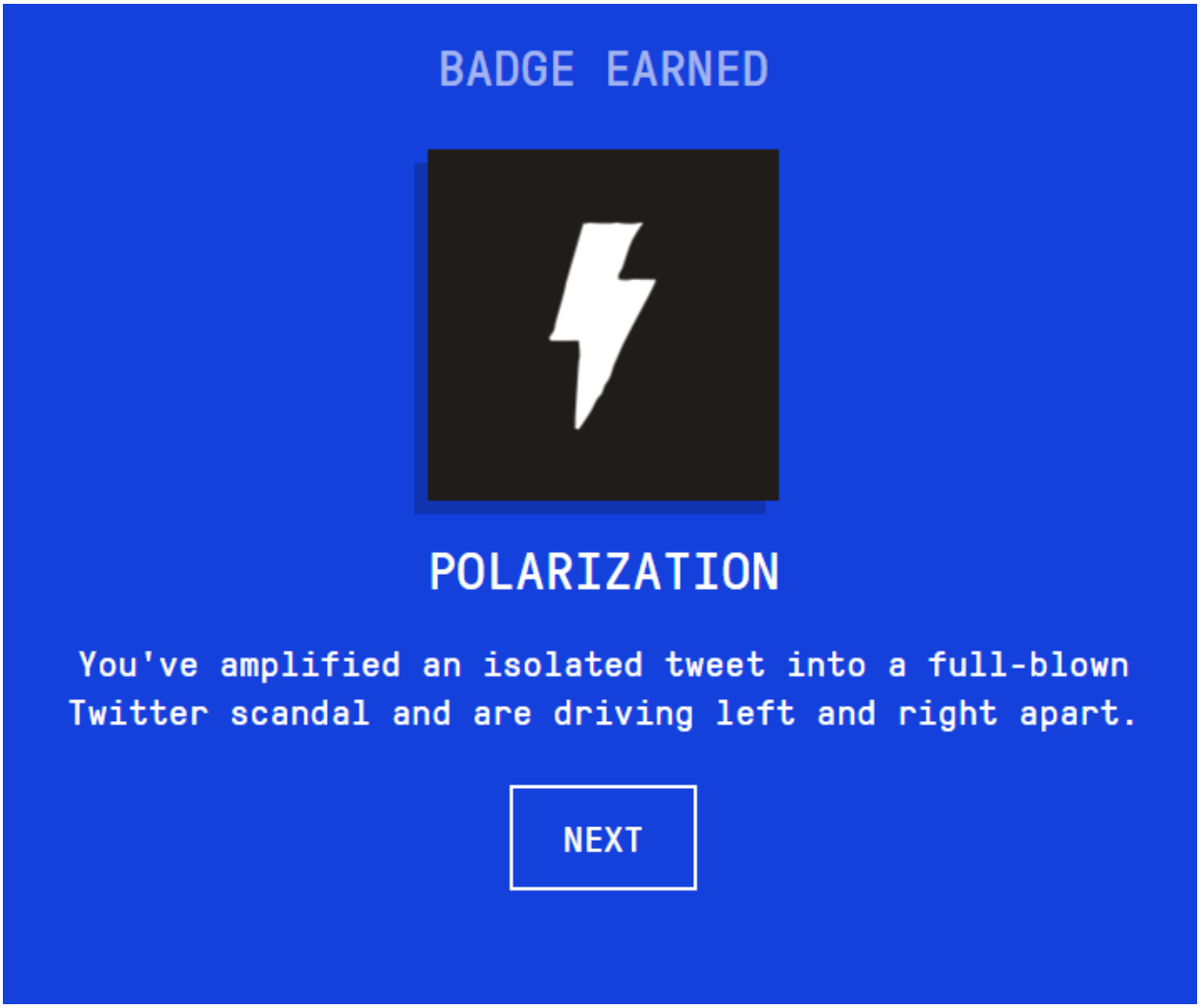

Figure 5: a bar chart showing the number of accounts for each domain

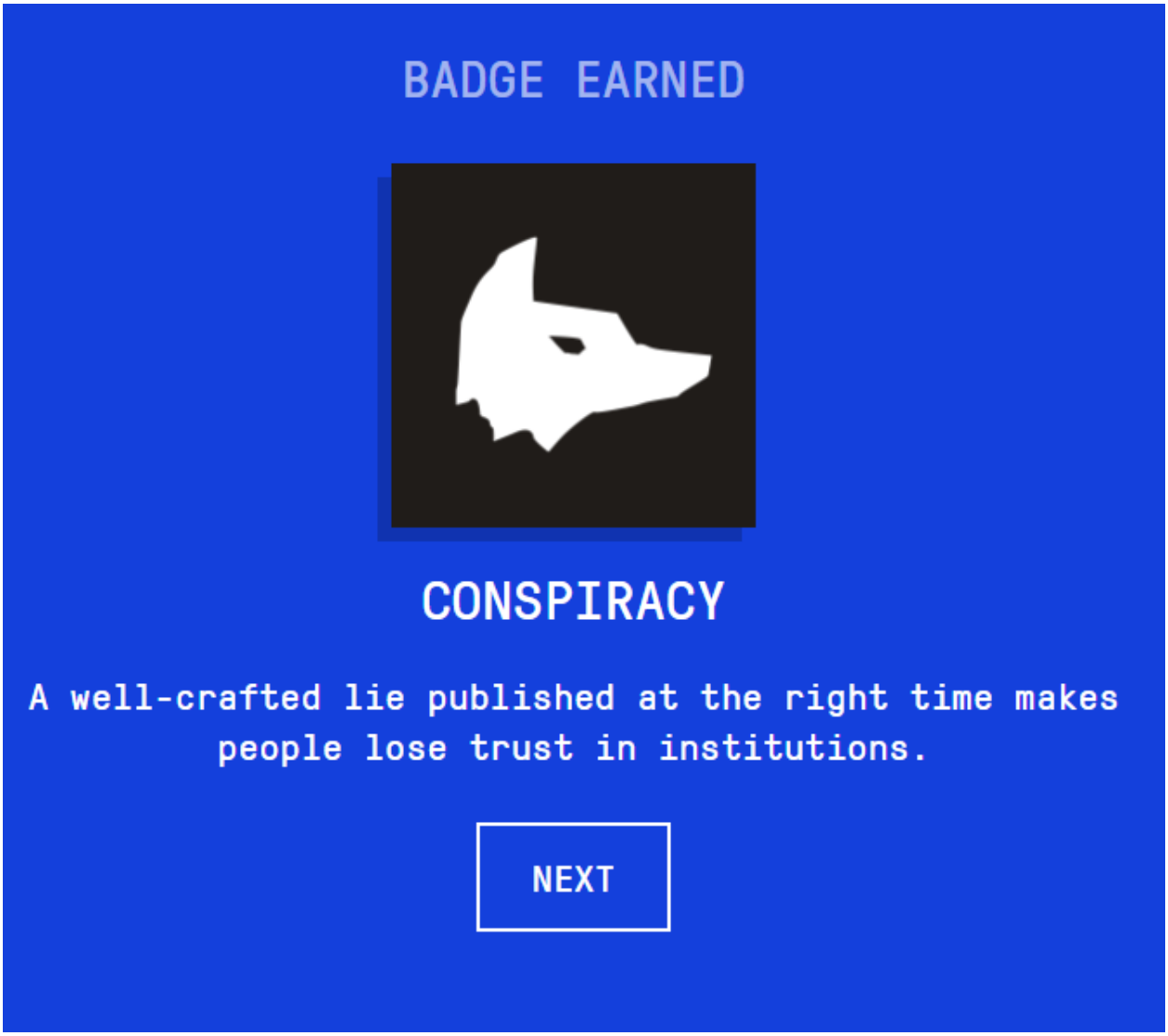

Figure 6: a bar chart showing the number of accounts for each domain

Figure 7: a bar chart showing the number of accounts for each domain

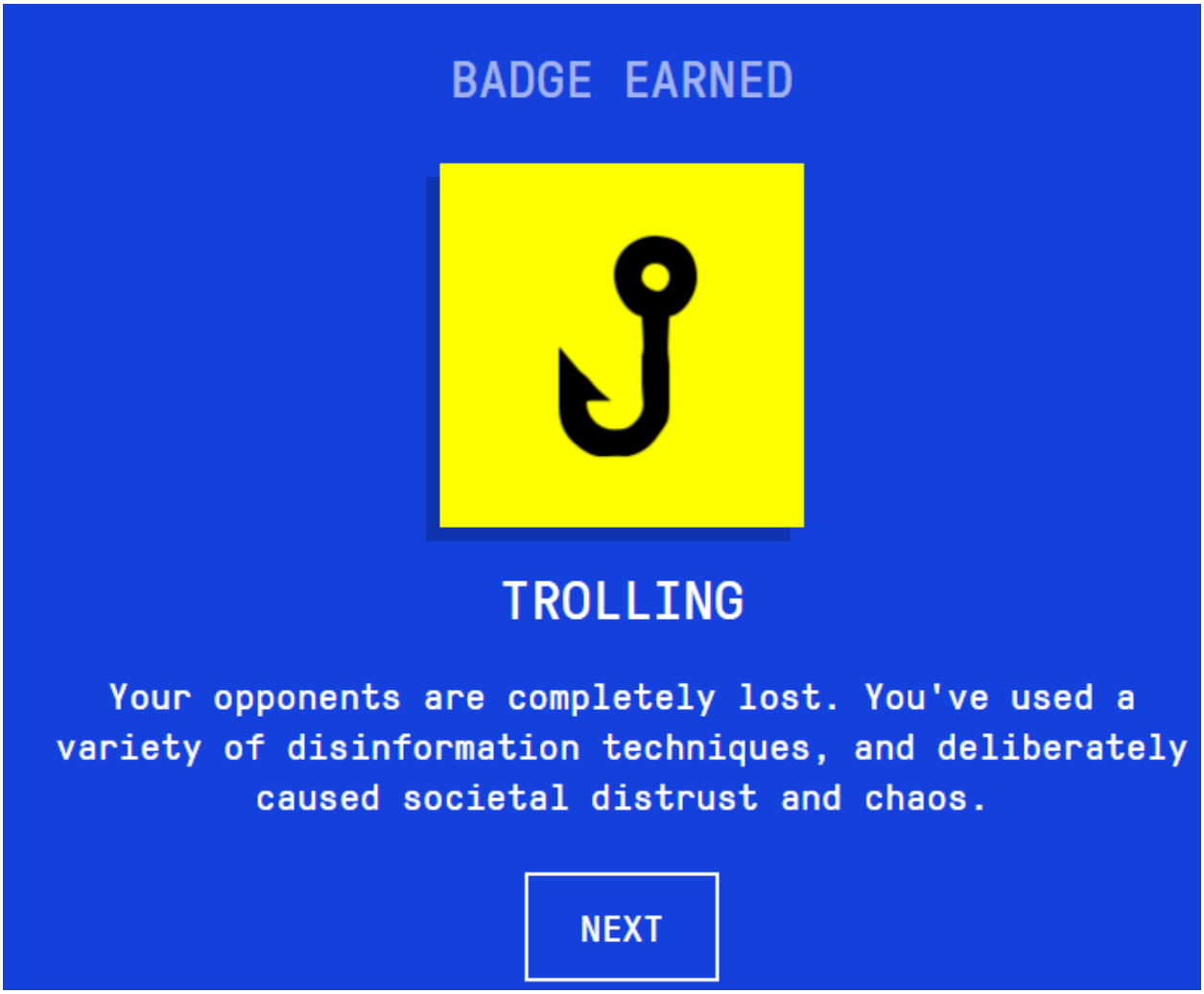

Figure 8: a bar chart showing the number of accounts for each domain

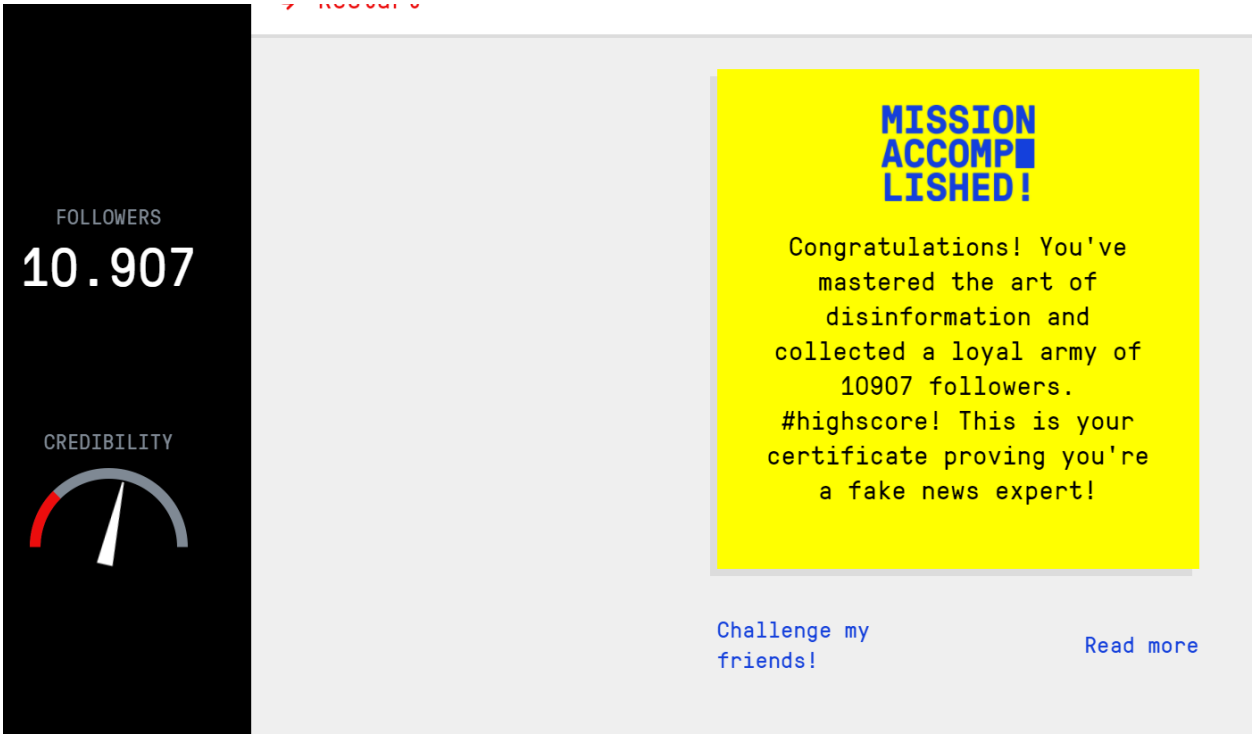

Figure 9: a bar chart showing the number of accounts for each domain

Discussion

I used https://www.getbadnews.com/ to play the came and was able to earn the above badges below:

- Impersonation in Figure 3 (Impersonating someone else and disguising myself as a credible news source which was highly effective in increase my followers)
- Emotion in Figure 4 (Playing to people's emotion out of fear, anger or compassing was a great tool for spreading my messages)
- Polarization in Figure 5 (By finding existing grievance and blowing them out of proportion, drove people apart and made think a story is much more important that it really was.)
- Conspiracy in Figure 6(I can use people's desires for the 'truth' as a tool to lure them into my band of followers)
- Discredit in Figure 7(When someone is attacking my credibility i strike back. I do not apologize nor do I play nice and above all I do not retreat!)
- Trolling in Figure 8 (Is a tool that evokes an emotional response such as anger, irritation or sadness. Dont hold back: your opponent's tears are your followers' mead!)
- The final score was 10907 followers.in Figure 9

References

- https://www.datacamp.com/community/tutorials/pandas-read-csv
- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort_values.html
- https://stackoverflow.com/questions/12021754/how-to-slice-a-pandas-data-frame-by-position
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rename.html
- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html
- https://stackoverflow.com/questions/38288372/unable-to-drop-a-column-from-pandas-dataframe
- https://stackoverflow.com/questions/39092067/pandas-dataframe-convert-column-type-to-string-or-categorical
- https://stackoverflow.com/questions/45164537/filter-pandas-data-frame-based-on-exact-string-match
- https://stackoverflow.com/questions/39092067/pandas-dataframe-convert-column-type-to-string-or-categorical
- https://stackoverflow.com/questions/39092067/pandas-dataframe-convert-column-type-to-string-or-categorical
- https://stackoverflow.com/questions/39092067/pandas-dataframe-convert-column-type-to-string-or-categorical