@Pulley Twitter Analysis

By Raghul Easwar

raghuleaswar@gmail.com

Introduction

This is a mini project that aims to gather some insights about the Twitter followers/following accounts of @Pulley's official handle.

This helps us assess the online presence of the company, and I was mainly interested with a few fundamental questions:-

1.What are the demographics of Pulley's followers, i.e where are they based?

2.What is the popularity of these followers/following users, and how many have a verified status?

3.What are these followers interested about, what kind of background do they have?

To answer these questions, I scrape through @Pulley's profile using Twitter API.

```
#Data Gathering
import requests
#Placing a request to get @Pulley's user id using the username
url = "https://api.twitter.com/2/users/by/username/pulley?user.fields=description,location,name,public_metrics,username,verified"

payload={}
headers = {
  'Authorization': 'Bearer XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'Cookie': 'guest_id=v1%3A167104791372375046'
}

response = requests.request("GET", url, headers=headers, data=payload)

pulley_user_details=response.json()
```

```
pulley_user_details
```

```
{'data': {'description': 'Cap table platform for hyper-growth startups. Our mission is to help more founders by making it easy to
understand fundraising, cap tables, and compensation.',
  'username': 'pulley',
  'id': '1311944193938341889',
  'public_metrics': {'followers_count': 2270,
   'following_count': 45,
   'tweet_count': 38,
   'listed_count': 30},
  'location': 'San Francisco, CA',
  'name': 'Pulley',
  'verified': False}}
```

```
#User ID is stored

pulleyuserid=pulley_user_details['data']['id']
```

```
#Placing requests to get list of followers of the user @Pulley
#Parameters like location,name,bio,verified status are follower count are also requested
#The following scripts iterates the requests until all data is scraped. This has to be done because of the API limit per request
i=0
while i < pulley_user_details['data']['public_metrics']['followers_count']:
  if i==0:

    url = f"https://api.twitter.com/2/users/{pulleyuserid}/followers?user.fields=description,id,location,name,public_metrics,username,ver
    payload={}
    headers = {
      'Authorization': 'Bearer XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXtwJ',
      'Cookie': 'guest_id=v1%3A167104791372375046'
     }
    response = requests.request("GET", url, headers=headers, data=payload)
    followerdetails=response.json()
    followers_df=pd.DataFrame(followerdetails['data'])
    print(response.text)
    next_token=followerdetails['meta']['next_token']
    i=i+1000
    print(i,'a')
  else:
    url = f"https://api.twitter.com/2/users/{pulleyuserid}/followers?user.fields=description,id,location,name,public_metrics,username,ver
    payload={}
    headers = {
      'Authorization': 'Bearer XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXwJ',
```

```
      'Cookie': 'guest_id=v1%3A167104791372375046'
    }

    response = requests.request("GET", url, headers=headers, data=payload)

    followerdetails=response.json()
    followers_df=followers_df.append(followerdetails['data'],ignore_index=True)
    if followerdetails['meta']['result_count']>=1000:
      next_token=followerdetails['meta']['next_token']
      i=i+1000
      print(i,'b')

    else:
      break
```

```
##Data Cleaning

#All the data is stored in jsn files, cleaned structurally and stored in pandas dataframes
#Null values are taken care of
followers_df['location'].fillna("Not Available",inplace=True)
followers_df['Followers']=1
```

```
for i in range(len(followers_df)):
  followers_df.loc[[i+1],['Followers']]=followers_df['public_metrics'][i+1]['followers_count']
```

```
followers_df=followers_df.drop(['public_metrics'],axis=1)
```

```
followers_df
```

	id	verified	username	name	description	location	Followers
1	778658843303772160	False	JohnSnow012	Rain	Chimbinha não para mané	Not Available	92
2	1208037875402575873	False	AndreLu53404656	Andre Luiz Silveira	Front End Developer\nReact/React Native	São José, Brasil	12
3	626916502	False	rodrigomsnet	Rodrigo Moreno	Trabalho com projetos de desenvolvimento educa...	Goiânia - State of Goiás, Braz	54
4	2389240458	False	iMatheeeus	FullMatt Alchemist	Oi vamos jogar um valorants?	Meu país Pernambuco	11
5	3087846207	False	brielackerman	xoxa capenga frágil e inconsistente	falo de livros e de publicidade e de escrita e...	Rio de Janeiro, Brasil	162
...	...	...	...	...	...	...	...
2267	1885152368	False	VLC2015	Adam Langley	Learning and operating @anduriltech |\nDirecto...	Newport Beach, CA	237
2268	570825874	False	simonwhite87	Simon White	Co-founder & CTO @rebanknow (YC W19) - buildin...	London, England	609
2269	749427855617060864	False	4gnle	agnle 🦇🔊	I love the internet / gym bro / dev	Not Available	462

```
#A request is placed to get the details of accounts followed by Pulley
#Same procedure of wrangling and cleaning takes place
url = f"https://api.twitter.com/2/users/{pulleyuserid}/following?user.fields=description,id,location,name,public_metrics,url,username,ver

payload={}
headers = {
  'Authorization': 'Bearer XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXtwJ',
  'Cookie': 'guest_id=v1%3A167104791372375046'
}

response = requests.request("GET", url, headers=headers, data=payload)

followingdetails=response.json()
following_df=pd.DataFrame(followingdetails['data'])
```

```
following_df.index+=1
```

```
following_df
```

	name	verified	username	id	location	public_metrics	url	de
1	Mark Erdmann	False	markerdmann	18205337	Calgary, Alberta	{'followers_count': 457, 'following_count': 90...	https://t.co/M8KzprRLUI	Co-
2	Shreyas Doshi	True	shreyas	2048741	Oakland, CA	{'followers_count': 205423, 'following_count':...	https://t.co/TPtxxvBISI	star built
3	suryasays	False	suryasays	7438022	les, nyc, usa, eth.	{'followers_count': 1925, 'following_count': 1...	https://t.co/9qXBEASEIA	|no ceo @ (acc
4	Brandon Camhi	False	bcamhi	371638870	NaN	{'followers_count': 301, 'following_count': 62...	NaN	Growt
5	Varun Sharma	False	varunconfirms	120726288	Our Planet	{'followers_count': 334, 'following_count': 95...	https://t.co/mV9dkqoMX0	@e @sc
6	Visible	False	VisibleVC	913181174	Customers on every continent.	{'followers_count': 4318, 'following_count': 2...	https://t.co/jB6TQthnD5	resou fou
7	Rasty Turek	False	synopsi	14710416	San Diego, CA	{'followers_count': 13997, 'following_count': ...	https://t.co/j4p8Ewhg2J	Build Di
8	Elizabeth Yin 💛	True	dunkhippo33	14756283	San Francisco Bay Area, CA	{'followers_count': 106015, 'following_count':...	https://t.co/HTK9QmFYJA	@hu De
9	Shin Kim	False	_shinkim	816345787084906497	San Francisco	{'followers_count': 491, 'following_count': 38...	https://t.co/8utyB7oOkW	Ma c
						{'followers_count': 765.		He
					Homes across	{'followers_count':		C
13		False	vael_gilboa	2992492374	Brooklyn, NY		https://t.co/b0VvcEIkF4	

```
for i in range(len(following_df)):
  following_df.loc[[i+1],['Followers']]=following_df['public_metrics'][i+1]['followers_count']
```

```
following_df['location'].fillna("Not Available",inplace=True)
```

```
following_df['Is_Following']=1
following_df['Followers']=following_df['Followers'].astype(int)
```

```
following_df=following_df.drop(['public_metrics'],axis=1)
```

```
#Both the followers dataframe and following dataframe are merged together with the addition of two columns-Is_Follower and Is_Following
#Since a follower can also be followed, these columns are created with a binary 1 or 0 for true or false conditions. This makes it easier
#all the data in a consolidated table
followerandfollowing=pd.merge(followers_df, following_df, how='outer', on=['id','name','username','verified','location','description','Fo
```

```
followerandfollowing['Is_Following'].fillna(0,inplace=True)
followerandfollowing['Is_Follower'].fillna(0,inplace=True)

followerandfollowing['Is_Follower']=followerandfollowing['Is_Follower'].astype(int)
followerandfollowing['Is_Following']=followerandfollowing['Is_Following'].astype(int)
```

```
followerandfollowing.index+=1
```

```
#This is a list of followers of Pulley being followed back
followerandfollowing[(followerandfollowing['Is_Follower']==1)&(followerandfollowing['Is_Following']==1)]
```

```
#Since location field does not give coordinates we use the following library to get the coordinates. For locations with no values or unre
#values, the coordinates are assigned null values
locator = Nominatim(user_agent="myGeocoder")
followerandfollowingdata['Lat']=1.0
followerandfollowingdata['Lon']=1.0
for i in range(len(followerandfollowingdata)):
```

```
  if followerandfollowingdata['location'][i+1]!='Not Available':
    location = locator.geocode(followerandfollowingdata['location'][i+1],timeout=10000)
    print(location)
    if location is not None:
      followerandfollowingdata['Lat'][i+1]=location[1][0]
      followerandfollowingdata['Lon'][i+1]=location[1][1]
    else:
      followerandfollowingdata['Lat'][i+1]=None
      followerandfollowingdata['Lon'][i+1]=None
  else:
    followerandfollowingdata['Lat'][i+1]=None
    followerandfollowingdata['Lon'][i+1]=None
```

```
#To understand the interests of these users, we want to look into their user bios.
#To standardize data formats, remove unnecessary words and to tokenize them, we use the following libraries
import regex as re
!pip install transformers
from transformers import AutoTokenizer
from transformers import AutoModelForSequenceClassification
from scipy.special import softmax
```

```
import nltk
#Data in each user's description is brought to the same format and stopwords are also removed

followerandfollowingdata['description']=followerandfollowingdata['description'].astype(str)
def cleandescription(desc):
  desc=desc.lower()
  desc = ' '.join(re.sub("([@#][A-Za-z0-9_]+)|(\w+:\/\/\S,.&+)"," ", desc).split())
  desc = re.sub(r"http\S+|www\S+|https\S+", '', desc, flags = re.MULTILINE)
  text_tokens = word_tokenize(desc)
  desc = [word for word in text_tokens if not word in stopwords.words()]
  desc = (" ").join(desc)
  desc = re.sub(r'[^\w\s]', '',desc)

  return desc

followerandfollowingdata['description']=followerandfollowingdata['description'].apply(cleandescription)
```

```
# All words are stored in a dictionary for each user
dictofwordsbyuser={}

for i in range(len(followerandfollowingdata)):
  dictofwordsbyuser[i+1]=followerandfollowingdata['description'][i+1].split()

dictofwordsbyuser
```

```
#All the words of all users are restructured and inserted into one list
fullstringofwords=''

for i in range(len(followerandfollowingdata['description'])):
  b=followerandfollowingdata['description'][i+1]
  fullstringofwords=fullstringofwords+""+b
```

```
fullstringofwords
```

```
'chimbinha manéfront developer reactreact nativetrabalho projetos desenvolvimento educação digital  programação  edição produção ví
deo  modelagem animação 3dvamos jogar valorants falo livros publicidade escrita marketing falo falo falo  nan bahia  brasil  doado
r órgãos    bios  development full stacknan\u200d product design lead  community \u200d building  musician  hang wife   burkland o
utsourced cfos  accountants  tax experts hr services growing startups guaranteed hookups area lost cryptoverse ceo  founder gigano
mics  mission make insurance loveable  pit bull named petey s made iceweb69 enthusiastreborn bron water spirit  cheer  thy faith m
ade thee  holyfire\u200d swe intern venture   cs student   winner 2021 founder  ceo  inventory autopilot  exdirector engineering
product builder20 yrs healthcarelong broken   worse south  buildingbuilding  nanstrategy  investment  ir china  korea investment b
```

```
#Total words in the list is 12638
fullstringofwords_split=fullstringofwords.split()
len(fullstringofwords_split)
```

```
12638
```

```
#Unique words are 5455
#To fetch unique words, we use the function to parse through the entire list and store only unique values in another list

stringwoduplicates = list(set(fullstringofwords_split))
```

```
len(stringwoduplicates)
stringwoduplicates.sort()
```

```
5455
```

```
#Both strings are compared and we take the count of every unique word
def countOccurrences(str, word):

    # split the string by spaces in a
    a = str.split(" ")

    # search for pattern in a
    count = 0
    for i in range(0, len(a)):

        # if match found increase count
        if (word == a[i]):
           count = count + 1

    return count


worddict={}
for i in range(len(stringwoduplicates)):
  worddict[stringwoduplicates[i]]=countOccurrences(fullstringofwords,stringwoduplicates[i])
```

```
1
```

```
#The dictionary is stored in a Series
wordseries  = pd.Series(worddict,index=worddict.keys())
```

```
# Data is sorted in the dictionary to get a clear idea
import operator

sorted_d = dict( sorted(worddict.items(), key=operator.itemgetter(1),reverse=True))
sortedwords=pd.Series(sorted_d,index=worddict.keys())
```

```
#Both the datasets - followers data as well as words data are extracted as excel
# files and loaded on a PowerBI dashboard to create an interactive dashboard
sortedwords.to_excel("\content\wordsdata.xlsx")
```

```
a
#Location data shows a very high number of users from San Franciso
```

```
Not Available                605
San Francisco, CA            110
New York, NY                  40
Los Angeles, CA               37
London, England               35
                             ...
Lancaster, PA                  1
Online                         1
in a sweater polo 👚             1
Awka                           1
Mountains and Great Lakes      1
Name: location, Length: 880, dtype: int64
```

Dashboard

The dashboard tries to answer the initial questions raised, at a high-level.

Features:

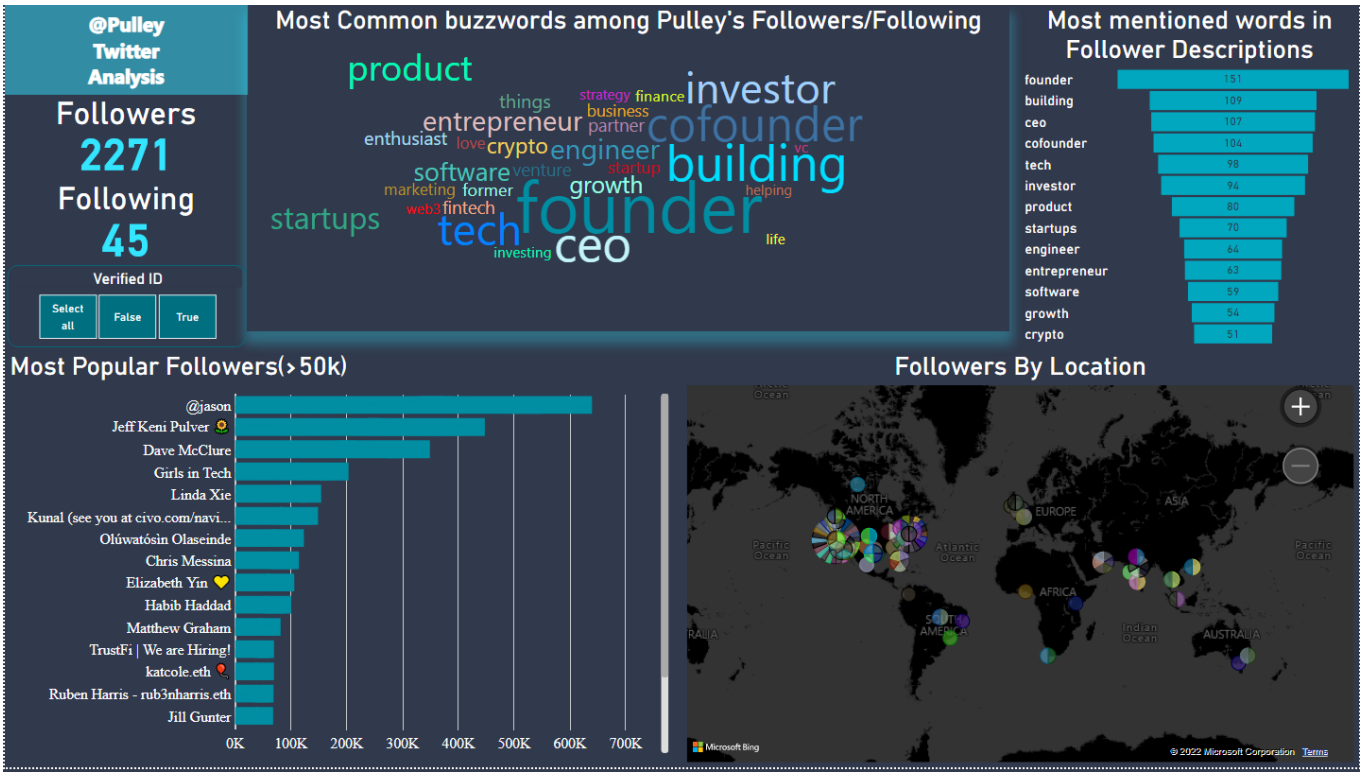
-Interactive

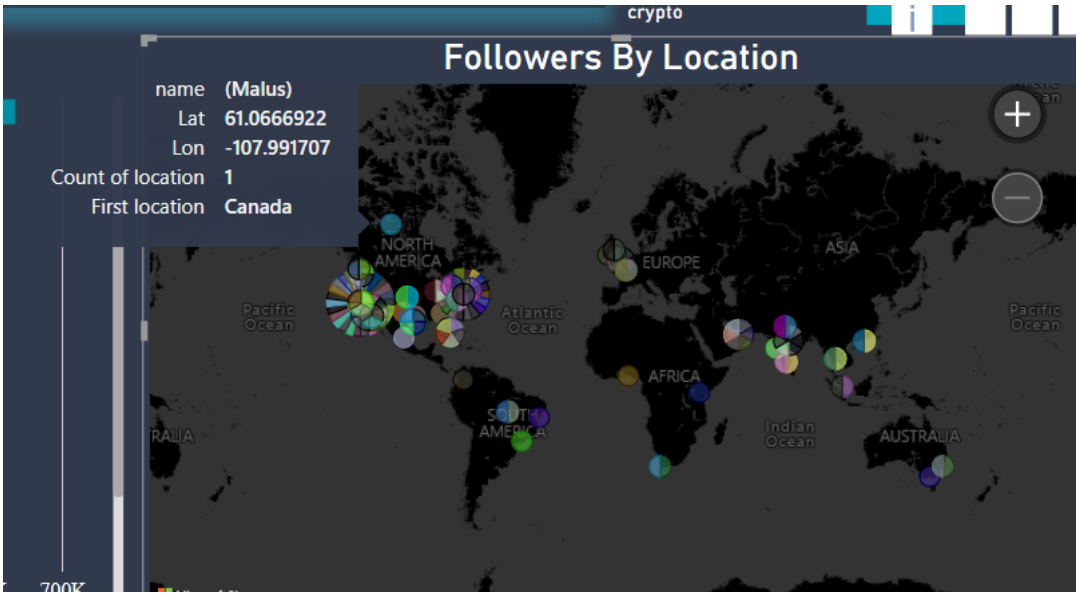
-Has Universal filter to filter between verified and/or non-verified accounts

-Visualizes buzzwords surrounding Pulley followers/following as mentioned

-Uses coordinates to plot on a map the locations of users

-Follower/Following filter can also be added ofr future views

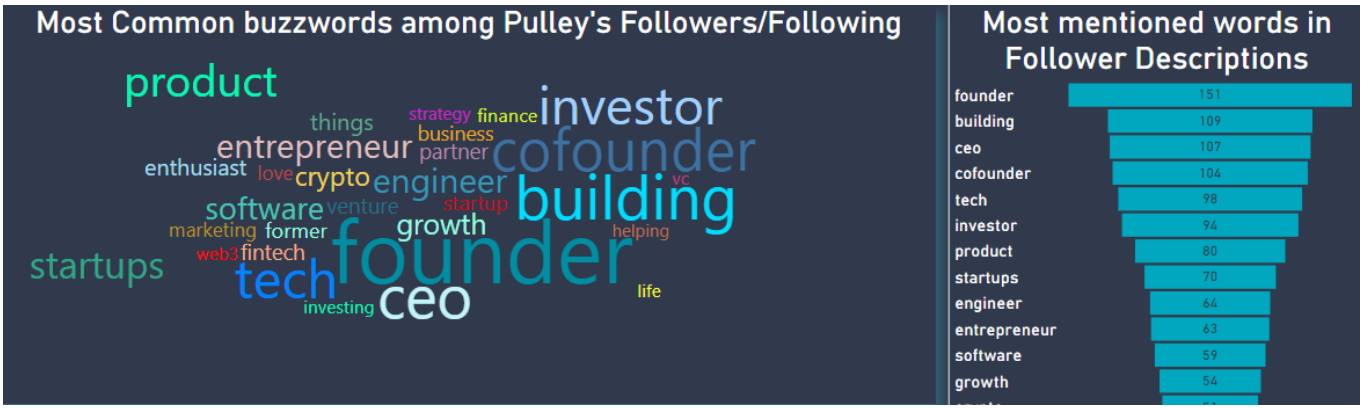


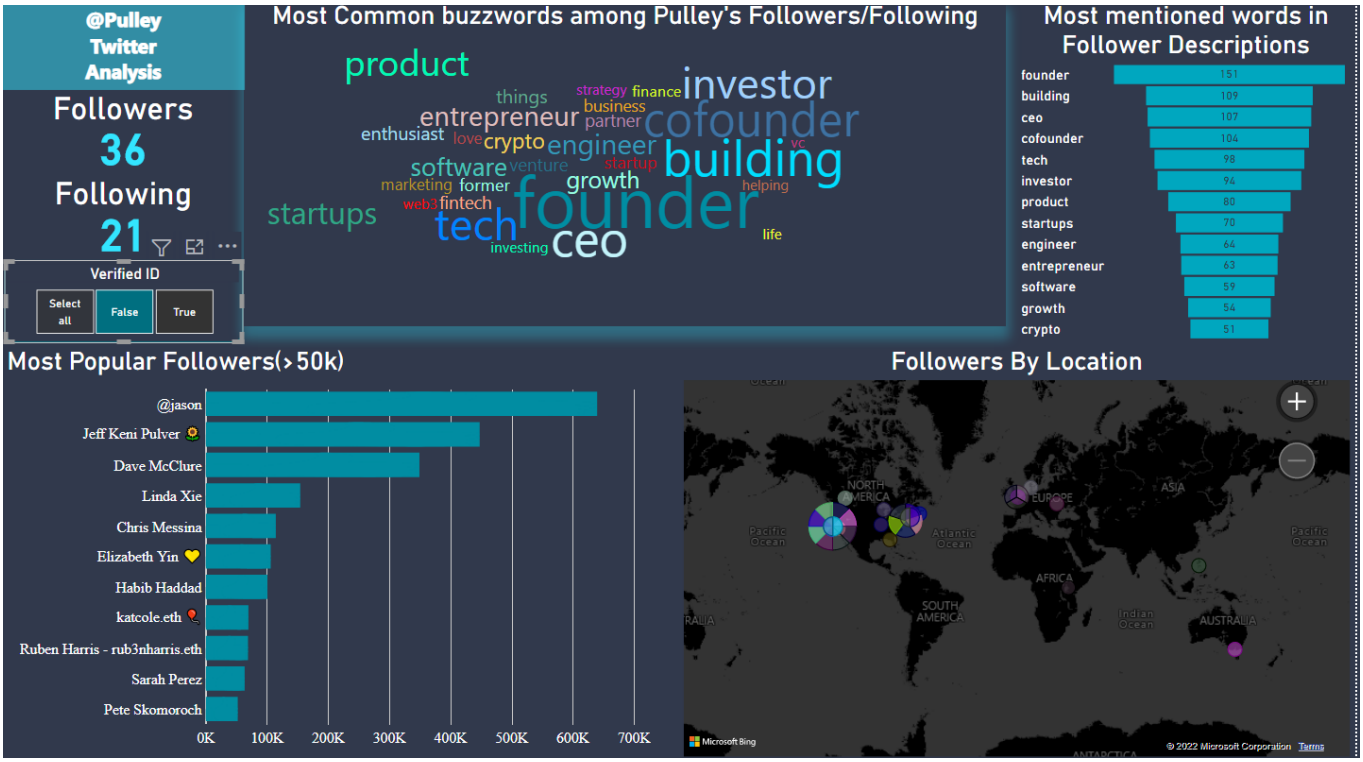


We can see the username, location plotted on the graph. A high concentration of followers evidently come from the US

-It is interesting to see what all the followers talk about/ are passionate about. It is not the most surprising to see words like founder, ceo, startups considering Pulley's domain.

-But it is quite amusing to find out that the buzzwords are all up and coming tech related jargon, indicating a crowd that is tech-savvy and up to date with updates in the tech world. web3, crypto, engineer are all good examples.





-With the verified filter we can notice that Pulley is being followed by 36 verified accounts, and Pulley follows 21 verifies accounts.

-We can also see the most popular verified IDs following Pulley.

Conclusion

Interesting insights could be drawn from scraping through @Pulley's Twitter account.

The dashboard could potentially help Pulley:

1.Understand how the brand's online presence is. It could give insights on how to navigate further and target the right people.

2.Understand where Pulley's followers come from. Even though the US dominates the follower count, it could be observed how diverse the followers are in terms of geography.

3.Track influencial people who have taken interest on Pulley and seek out for partnerships and collaborations.

4.Measure growth through follower count.

5.Stay on top of what the followers are interested in. Helps explore leads in the industry, and helps understand the theme and culture of the followers.

Scope for improvement

This is a high-level basic analysis giving fundamental insights. To further disect followers:

1.Scrape through tweets of followers talking about startups,funding,Pulley etc to be more up to date with new developments

2.Conduct more analysis on other metrics apart from followers, popularity, location. Dig deeper into location based interests etc.

3.Build more visualizations and readily available tables to instantly see the insight in a detailed form to disect it.

4.Validate and iterate with more questions.