**Fake Profiles Detection**

*A*

*Project Report*

*Submitted in partial fulfilment of the Requirements* *for the award of the Degree of*

**BACHELOR OFENGINEERING**

IN

**INFORMATION TECHNOLOGY**

**By**

**M. Charishma(1602-19-737-008)**

**M. Pranay Kumar Reddy(1602-19-737-027)**

**N. Rachel Reddy(1602-19-737-030)**

A picture containing diagram

Description automatically generated

**Department of Information Technology**

# **Vasavi College of Engineering (Autonomous)**

***ACCREDITED BY NAAC WITH 'A++' GRADE***

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2021-22**

# **Vasavi College of Engineering (Autonomous)**

***ACCREDITED BY NAAC WITH 'A++' GRADE***

# **(Affiliated to Osmania University)**

# **Hyderabad-500 031**

**Department of Information Technology**

A picture containing diagram

Description automatically generated

**DECLARATION BY THE CANDIDATES**

We, **M. Charishma** , **M. Pranay Kumar Reddy**, **N. Rachel Reddy** bearing hall ticket number, **1602-19-737-008**, **1602-18-737-027**, **1602-18-737-030**, hereby declare that the project report entitled “**Fake Profiles Detection**” under the guidance of **Ms. B.Leelavathy**, Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement of Mini project of V semester of **Bachelor of Engineering in Information Technology.**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**M.Charishma**, (1602-19-737-008)

**M.Pranay Kumar Reddy**, (1602-19-737-027)

**N.Rachel Reddy**, (1602-19-737-030)

**Vasavi College of Engineering (Autonomous)**

***ACCREDITED BY NAAC WITH 'A++' GRADE***

**(Affiliated to Osmania University)**

**Hyderabad-500031**

A close-up of a sandwich

Description automatically generated with low confidence**Department of Information Technology**

**BONAFIDE CERTIFICATE**

This is to certify that the project entitled “**Fake Profiles Detection**” being submitted by **M. Charishma , M. Pranay Kumar Reddy, N. Rachel Reddy** bearing **1602-19-737-008**, **1602-19-737-027, 1602-19-737-030**, in partial fulfillment of the requirements for the completion of **Mini Project** of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

**B. Leelavathy Dr. K. Ram MohanRao**

**Assistant Professor Professor & HOD , IT Internal Guide Department of IT**

# **ACKNOWLEDGEMENT**

I extend my sincere thanks to **Dr. S. V. Ramana**, Principal, Vasavi College of Engineering for his encouragement.

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

It is with utmost pleasure that we avail this opportunity to express our heartfelt gratitude to **Ms. B.Leelavathy (Asst. Prof, IT),** our internal guide, under whose guidance we have finished this project successfully. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the work.

Also, we would like to express our sincere gratitude to our project coordinator **Ms.C.Sireesha (Asst Prof, IT)** whose efforts were constant from day one in monitoring our project.

Finally, yet importantly, we would like to express our heartfelt thanks to our respected **Dr. K. Ram Mohan Rao (Prof. & HOD, IT)** who was very helpful in providing resources for the project.

We would like to thank all teaching and non teaching members of the Department of Information Technology, Vasavi College of Engineering for their generous help in various ways for the completion of this project.

**M.Charishma (1602-19-737-008)**

**M.Pranay Kumar Reddy (1602-19-737-027)**

**N.Rachel Reddy (1602-19-737-030)**

**TABLE OF CONTENTS**

1. ABSTRACT 6
2. INTRODUCTION 7
   1. HISTORY 7
   2. SOCIAL IMPACT 7
3. OBJECTIVE 9
   1. LITERACTURE SURVEY 9
   2. [TECHNOLOGY USED](#_bookmark3)  10
4. DESIGN 11
   1. USE CASE DIAGRAM 11
   2. SEQUENCE DIAGRAM 11
5. SOFTWARE REQUIREMENTS 13
6. IMPLEMENTATION AND RESULTS 14
7. CONCLUSION AND FUTURE WORK 29
8. REFERENCES 30

**1.ABSTRACT**

The exponential growth of the Internet interconnections has led to a significant growth of cyber-attack incidents often with disastrous and grievous consequences. Every day several people are creating their profiles on the social network platforms and they are interacting with others independent of the user’s location and time. The social network sites not only provide advantages to the users and also provide security issues to the users as well their information. To analyze who are encouraging threats in social network we need to analyze social networks profiles of the users. Cyber-criminals are creating fake profiles and carrying out defaming/Political & Religious Propaganda on a large scale against individuals or Organizations/Government Bodies and they are creating unrest among the society. It's well known that scammers use fake social media accounts to connect with users and their friends to scrape personal information in an effort to steal identities. Fake account owners also reach out to anyone who's accepted their friend request to try and scam them out of money. Social networks fake profile creation is considered to cause more harm than any other form of cyber-crime. This crime has to be detected even before the user is notified about the fake profile creation. Many algorithms and methods have been proposed for the detection of fake profiles in the literature. This project sheds light on the role of fake identities in advanced persistent threats and covers the many approaches in detecting fake social media profiles. In order to make a relevant prediction of fake or genuine profiles, we will assess the impact of three supervised machine learning algorithms: Random Forest (RF), Decision Tree (DT-J48), and Naïve Bayes (NB).

**2.INTRODUCTION**

Social networking site is a website where each user has a profile and can keep in contact with friends, share their updates, and meet new people who have the same interests. These Online Social Networks (OSN) use web2.0 technology, which allows users to interact with each other. Social networking sites are growing rapidly and changing the way people keep in contact with each other. The online communities bring people with the same interests together which makes it easier for users to make new friends.

**2.1 History**

These social networking sites starting with http://www.sixdegrees.com in 1997 then came http://www.makeoutclub.com in 2000. Sixdegrees.com couldn’t survive much and closed very soon but new sites like myspace, LinkedIn, Bebo became successful and Facebook was launched in 2004 and presently it is the largest social networking site in the world.

**2.2 Social Impact**

In the present generation, the social life of everyone has become associated with online social networks. These sites have made a drastic change in the way we pursue our social life. Adding new friends and keeping in contact with them and their updates has become easier. Online social networks have an impact on science, education, grassroots organizing, employment, business, etc. Researchers have been studying these online social networks to see the impact they make on people. Teachers can teach the students easily through this, making a friendly environment for the students to study. Teachers nowadays are getting themselves familiar with these sites bringing online classroom pages, giving homework, making discussions, etc. which improves education a lot. The employers can use these social networking sites to employ the people who are talented and interested in the work, their background check can be done easily using this. Most of the OSN is free but some charge the membership fee and use this for business purposes and the rest of them raise money by using the advertising. This can be used by the government to get the opinions of the public quickly. The examples of these social networking sites are sixdegrees.com, The Sphere, Nexopia which is used in Canada, Bebo, Hi5, Facebook, MySpace, Twitter, LinkedIn, Google+, Orkut, Tuenti used in Spain, Nasza-Klasa in Poland, Cyworld mostly used in Asia, etc. are some of the popular social networking sites.

**3.OBJECTIVE**

In today’s online social networks there have been a lot of problems like fake profiles, online impersonation, etc. To date, no one has come up with a feasible solution to these problems. In this project, I intend to give a framework with which the automatic detection of fake profiles can be done so that the social life of people become secured and by using this automatic detection technique we can make it easier for the sites to manage the huge number of profiles, which can’t be done manually.

**3.1 Literature Survey**

Various fake record recognition methodologies depend on the investigation of individual interpersonal organization profiles, with the point of distinguishing the qualities or a combination thereof that help in recognizing the legitimate and the fake records. In particular, various features are extracted from the profiles and posts, and after that Machine learning algorithms are used so as to construct a classifier equipped for recognizing fake records. For instance, Nazir et al. (2010) describes recognizing and describing phantom profiles in online social gaming applications. The article analyses a Facebook application, the online game “Fighters club”, known to provide incentives and gaming advantage to those users who invite their peers into the game. The authors contend that by giving such impetuses the game motivates its players to make fake profiles. By presenting those fake profiles into the game, the user would increase a motivating force of an incentive for him/herself. Adikari and Dutta (2014) depict recognizable proof of fake profiles on LinkedIn. The paper demonstrates that fake profiles can be recognized with 84% exactness and 2.44% false negative, utilizing constrained profile information as input. Techniques, for example, neural networks, SVMs, and Principal component analysis are applied. Among others, highlights, for example, the number of languages spoken, training, abilities, suggestions, interests, and awards are utilized. Qualities of profiles, known to be fake, posted on uncommon sites are utilized as a ground truth. Chu et al. (2010) go for separating Twitter accounts operated by humans, bots, or cyborgs (i.e., bots and people working in concert). As a part of the detection problem formulation, the Identification of spamming records is acknowledged with the assistance of an Orthogonal Sparse Bigram (OSB) text classifier that uses pairs of words as features. Stringhini et al. (2013) analyze Twitter supporter markets. They describe the qualities of Twitter devotee advertises and group the clients of the business sectors. The authors argue that there are two major kinds of accounts who pursue the “client”: fake accounts(“sybils”), and compromised accounts, proprietors of which don’t presume that their followers' rundown is expanding. Clients of adherent markets might be famous people or legislators, meaning to give the appearance of having a bigger fan base, or might be cybercriminals, going for making their record look progressively authentic, so they can rapidly spread malware. What's more, spam. Thomas et al. (2013) examine black market accounts utilized for distributing Twitter spam.

This survey is done to comprehend the need and prerequisite of the general population, and to do as such, we went through different sites and applications and looked for the fundamental data. Based on these data, we made an audit that helped us get new thoughts and make different arrangements for our task. We reached the decision that there is a need for such an application and felt that there is a decent extent of progress in this field too.

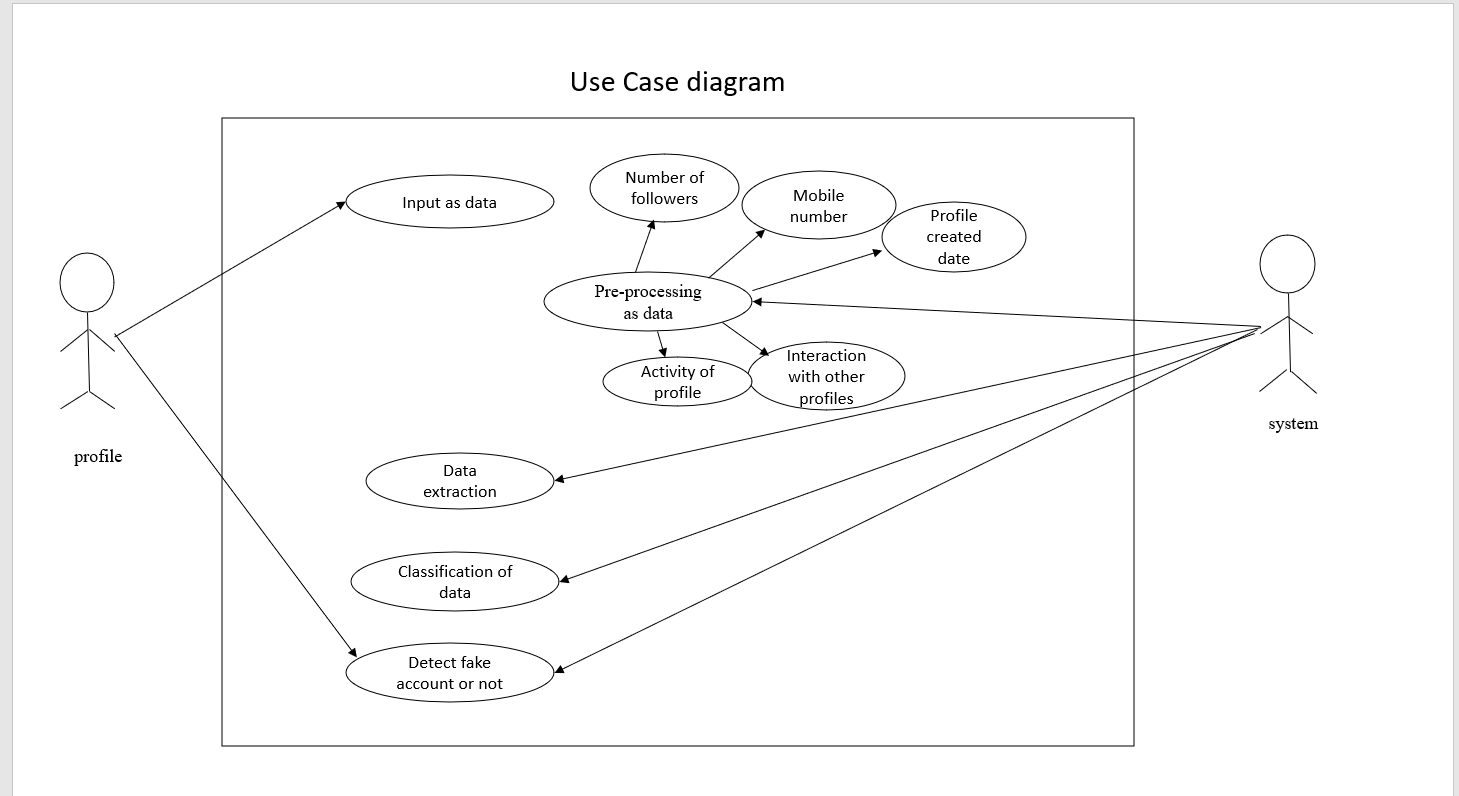
**3.2 Technology Used**

**PYTHON** - Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed AND supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

**JUPYTER NOTEBOOK** - Project Jupyter is a nonprofit organization created to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

**4.DESIGN**

**4.1 Use Case Diagram**



**4.2 Sequence Diagram**

1. The detection process starts with the selection of the profile that needs to be tested.

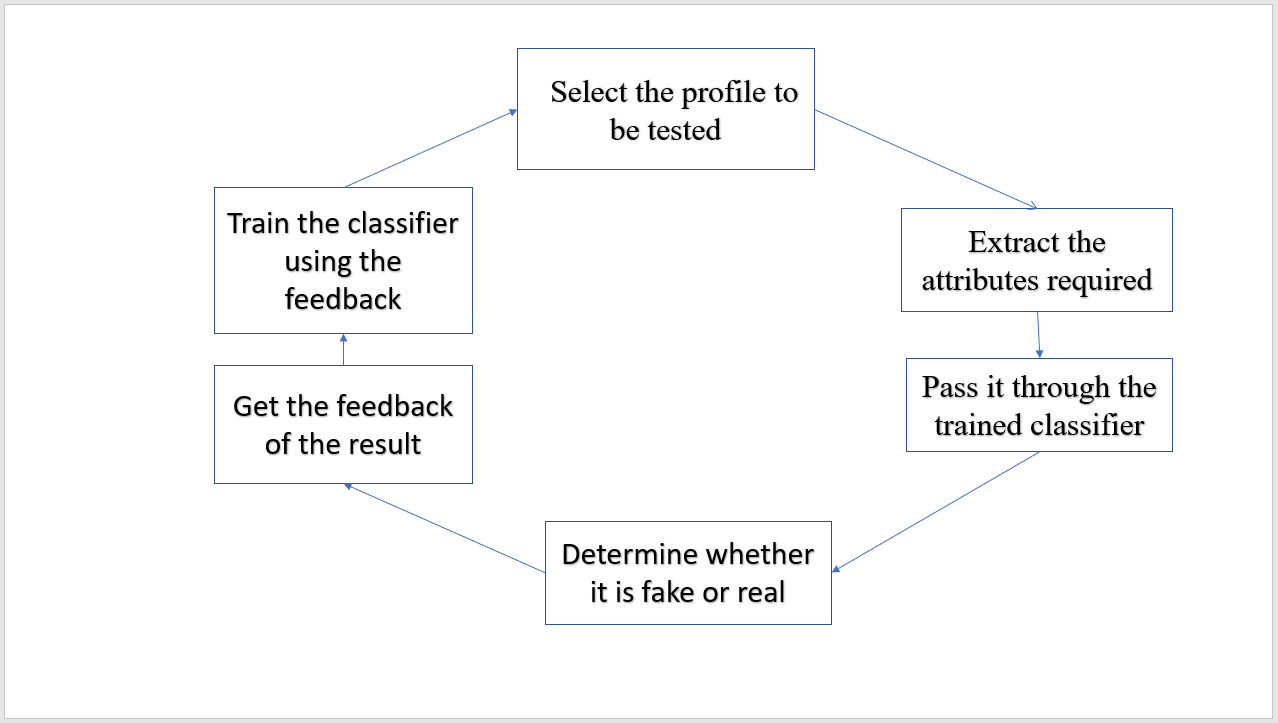
2. After the selection of the profile, the suitable attributes (i.e. features) are selected on which the classification algorithm is implemented.

3. The attributes extracted are passed to the trained classifier. The classifier gets trained regularly as new training data is fed into the classifier.

4. The classifier determines whether the profile is fake or genuine.

5. The classifier may not be 100% accurate in classifying the profile so; the feedback of the result is given back to the classifier.

6. This process repeats and as the time proceeds, the no. of training data increases and the classifier becomes more and more accurate in predicting the fake profiles.



**5.SOFTWARE REQUIREMENTS SPECIFICATION**

**5.1 Python**

**•** Numpy

**•** Tweepy

**•** Pandas

**•** matplotlib

**•** sklearn

**5.2 Operating System**

**•** Windows or Linux

**REQUIREMENTS ANALYSIS**

1. **PYTHON:** Python is the programming language in which our project is developed.
2. **Libraries Used:**

1. Numpy: Used to handle multi-dimensional arrays

2. Tweepy: Used for connecting Twitter API.

3. Pandas: Used for data analysis.

4. Matplotlib: Used for plotting.

5. Sklearn: Used to implement Machine Learning Algorithms

1. **Operating System:** Project is tested on Windows 10.

**6.IMPLEMENTATION:**

#In[1]:

import pandas as pd

import tweepy

import time

import numpy as np

import matplotlib.pyplot as plt

from tweepy import Stream

from tweepy.streaming import StreamListener

import numpy as np

import matplotlib.pyplot as plt

#In[2]:

consumer\_key = 'd9Ksoz6Wb1jD0mqbW8rjaSNb7'

consumer\_secret = 'pHXnVSJeLbOxaYlbOR7BWFdDNhZSF6IzegZV87qUSUqy6Qe8qG'

access\_token = '3648603434-dGRu1nHet22tdoYeqaAGoN8MyZrNw9oXZQvGZUD'

access\_token\_secret = 'PZ8pcQBCb5zVPLRQNVQZc3Yzi0rz1wPef6O7RO7gzcvOf'

auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret)

auth.set\_access\_token(access\_token, access\_token\_secret)

api = tweepy.API(auth, wait\_on\_rate\_limit=True)

#In[3]:

print('Friends Names: ')

friends = []

for friend in tweepy.Cursor(api.friends, screen\_name = 'rajshamani').items(20):

try:

friends.append(friend.screen\_name)

print(friend.screen\_name)

time.sleep()

except Exception as e:

pass

with open(r"C:\Users\maram\OneDrive\Documents\dataset0.txt", "w") as f:

for item in friends:

f.write("%s\n" % item)

#In[4]

print('Friends Names:')

friends = []

for friend in tweepy.Cursor(api.friends, screen\_name = 'JeffBezos').items(20):

try:

friends.append(friend.screen\_name)

print(friend.screen\_name)

time.sleep()

except Exception as e:

pass

with open(r"C:\Users\maram\OneDrive\Documents\dataset1.txt", "w") as f:

for item in friends:

f.write("%s\n" % item)

#In[]

#extracting tweets from users

Total\_Data = []

fo = open(r"C:\Users\maram\OneDrive\Documents\dataset0.txt", "r")

f = fo.readlines()

fo.close()

dataset = map(lambda s: s.strip(),f)

try:

for datavar in dataset:

data = api.get\_user(datavar)

counter = 0

for status in tweepy.Cursor(api.user\_timeline, id = datavar).items(30):

try:

counter= counter+1

Total\_Data.append(status)

time.sleep()

except Exception as e:

pass

except Exception as e:

pass

print(len(Total\_Data))

#In[]

import urllib.parse

import pandas as pd

def process\_http(string):

url\_count = 0

for i in string.split():

s, n, p, pa, q, f = urllib.parse.urlparse(i)

if s and n:

url\_count += 1

return url\_count

def process\_hashtag(string):

hashtag\_count = 0

for i in string.split():

s, n, p, pa, q, f = urllib.parse.urlparse(i)

if i[:1] == '#':

hashtag\_count += 1

return hashtag\_count

def process\_mention(string):

mention\_count=0

for i in string.split():

s, n, p, pa, q, f = urllib.parse.urlparse(i)

if i[:1] == '@':

mention\_count += 1

return mention\_count

def process\_data(Total\_Data):

TwittID = [tweet.id for tweet in Total\_Data]

# Making the dataset in pandas frame

Data = pd.DataFrame(TwittID, columns = ['TwittID'])

# processing the data in Tweet level

Data["TextData"] = [tweet.text for tweet in Total\_Data]

Data["TweetCreatedAt"] = [tweet.created\_at for tweet in Total\_Data]

Data["RetweetCount"] = [tweet.retweet\_count for tweet in Total\_Data]

Data["TweetFavouriteCount"] = [tweet.favorite\_count for tweet in Total\_Data]

Data["TweetSource"] = [tweet.source for tweet in Total\_Data]

# processing the data in User Graph level

Data["UserID"] = [tweet.author.id for tweet in Total\_Data]

Data["UserScreenName"] = [tweet.author.screen\_name for tweet in Total\_Data]

Data["UserName"] = [tweet.author.name for tweet in Total\_Data]

Data["UserCreatedAt"] = [tweet.author.created\_at for tweet in Total\_Data]

Data["UserDescription"] = [tweet.author.description for tweet in Total\_Data]

Data["UserDescriptionLength"] = [len(tweet.author.description) for tweet in Total\_Data]

Data["UserFollowersCount"] = [tweet.author.followers\_count for tweet in Total\_Data]

Data["UserFriendsCount"] = [tweet.author.friends\_count for tweet in Total\_Data]

Data["UserLocation"] = [tweet.author.location for tweet in Total\_Data]

# Data["url"] = [tweet.author.url for in Total\_Data]

# Data["User\_mention"] = [user\_mentions.author.screen\_name for tweet in Total\_Data]

# Data["HashTag"] = [hashtag.text for tweet in Total\_Data]

Data["HttpCount"] = [process\_http(tweet.text) for tweet in Total\_Data]

Data["HashtagCount"] = [process\_hashtag(tweet.text) for tweet in Total\_Data]

Data["MentionCount"] = [process\_mention(tweet.text) for tweet in Total\_Data]

Data["TweetCount"] = [tweet.author.statuses\_count for tweet in Total\_Data]

return Data

Data = process\_data(Total\_Data)

Data.shape

#In[]

Data.tail(9)

#In[]

#saving data in csv file

import sys

Data.to\_csv(r'C:\Users\maram\OneDrive\Documents\Fdataset.csv',sep=',' , encoding='utf8')

#In[]

friends = []

class listener(StreamListener):

def on\_data(self, data):

try:

tweet = data.split(',"screen\_name":"')[1].split('","location')[0]

print(tweet)

friends.append(tweet)

return True

except BaseException as e:

print('failed on data' + str(e))

time.sleep(5)

def on\_error(self, status):

print(status)

twitterStream = Stream(auth, listener())

try:

for x in range(1,2):

twitterStream.filter(track=["cougar"])

except KeyboardInterrupt:

print("Keyboard interrupt")

with open(r"C:\Users\maram\OneDrive\Documents\dataset0.txt", "w") as f:

for item in friends:

f.write("%s\n" % item)

#In[]

Total\_Data = []

fo = open(r"C:\Users\maram\OneDrive\Documents\spam0.txt", "r")

f = fo.readlines()

fo.close()

dataset = map(lambda s: s.strip(),f)

try:

for datavar in dataset:

data = api.get\_user(datavar)

counter = 0

for status in tweepy.Cursor(api.user\_timeline, id = datavar).items(30):

try:

counter= counter+1

Total\_Data.append(status)

time.sleep()

except Exception as e:

pass

except Exception as e:

pass

print(len(Total\_Data))

#In[]

import urllib.parse

import pandas as pd

def process\_http(string):

url\_count = 0

for i in string.split():

s, n, p, pa, q, f = urllib.parse.urlparse(i)

if s and n:

url\_count += 1

return url\_count

def process\_hashtag(string):

hashtag\_count = 0

for i in string.split():

s, n, p, pa, q, f = urllib.parse.urlparse(i)

if i[:1] == '#':

hashtag\_count += 1

return hashtag\_count

def process\_mention(string):

mention\_count=0

for i in string.split():

s, n, p, pa, q, f = urllib.parse.urlparse(i)

if i[:1] == '@':

mention\_count += 1

return mention\_count

def process\_data(Total\_Data):

TwittID = [tweet.id for tweet in Total\_Data]

# Making the dataset in pandas frame

Data = pd.DataFrame(TwittID, columns = ['TwittID'])

# processing the data in Tweet level

Data["TextData"] = [tweet.text for tweet in Total\_Data]

Data["TweetCreatedAt"] = [tweet.created\_at for tweet in Total\_Data]

Data["RetweetCount"] = [tweet.retweet\_count for tweet in Total\_Data]

Data["TweetFavouriteCount"] = [tweet.favorite\_count for tweet in Total\_Data]

Data["TweetSource"] = [tweet.source for tweet in Total\_Data]

# processing the data in User Graph level

Data["UserID"] = [tweet.author.id for tweet in Total\_Data]

Data["UserScreenName"] = [tweet.author.screen\_name for tweet in Total\_Data]

Data["UserName"] = [tweet.author.name for tweet in Total\_Data]

Data["UserCreatedAt"] = [tweet.author.created\_at for tweet in Total\_Data]

Data["UserDescription"] = [tweet.author.description for tweet in Total\_Data]

Data["UserDescriptionLength"] = [len(tweet.author.description) for tweet in Total\_Data]

Data["UserFollowersCount"] = [tweet.author.followers\_count for tweet in Total\_Data]

# processing the data in User Graph level

Data["UserID"] = [tweet.author.id for tweet in Total\_Data]

Data["UserScreenName"] = [tweet.author.screen\_name for tweet in Total\_Data]

Data["UserName"] = [tweet.author.name for tweet in Total\_Data]

Data["UserCreatedAt"] = [tweet.author.created\_at for tweet in Total\_Data]

Data["UserDescription"] = [tweet.author.description for tweet in Total\_Data]

Data["UserDescriptionLength"] = [len(tweet.author.description) for tweet in Total\_Data]

Data["UserFollowersCount"] = [tweet.author.followers\_count for tweet in Total\_Data]

Data["UserFriendsCount"] = [tweet.author.friends\_count for tweet in Total\_Data]

Data["UserLocation"] = [tweet.author.location for tweet in Total\_Data]

# Data["url"] = [tweet.author.url for in Total\_Data]

# Data["User\_mention"] = [user\_mentions.author.screen\_name for tweet in Total\_Data]

# Data["HashTag"] = [hashtag.text for tweet in Total\_Data]

Data["HttpCount"] = [process\_http(tweet.text) for tweet in Total\_Data]

Data["HashtagCount"] = [process\_hashtag(tweet.text) for tweet in Total\_Data]

Data["MentionCount"] = [process\_mention(tweet.text) for tweet in Total\_Data]

Data["TweetCount"] = [tweet.author.statuses\_count for tweet in Total\_Data]

return Data

Data = process\_data(Total\_Data)

Data.shape

#In[]

#merging all data

import csv

import glob

import os

# get data file names

path = 'C:\Users\maram\OneDrive\Documents\CompleteDataSet.csv'

filenames = glob.glob(path + "/\*.csv")

content = []

for filename in filenames:

content.append(pd.read\_csv(filename, error\_bad\_lines=False))

Total\_leg = pd.concat(content, ignore\_index=True)

Total\_leg.tail(4)

#In[]

Total\_leg.to\_csv('C:\Users\maram\OneDrive\Documents\LegitimateDataSet.csv', sep=',' , encoding='utf8')

#In[]

# Merging Spammer Data

import csv

import glob

import os

# get data file names

path = 'C:\Users\maram\OneDrive\Documents\SpammerDataSet.csv'

filenames = glob.glob(path + "/\*.csv")

content = []

for filename in filenames:

content.append(pd.read\_csv(filename, error\_bad\_lines=False))

Total\_leg = pd.concat(content, ignore\_index=True)

Total\_leg.tail(4)

#In[]

Total\_leg.to\_csv('C:\Users\maram\OneDrive\Documents\SpammerDataSet.csv', sep=',' , encoding='utf8')

#In[]

import pandas as pd

Total\_leg\_data = pd.read\_csv('LegitimateDataSet.csv')

Total\_leg\_data.fillna(0, inplace=True)

Total\_leg\_data.shape

#In[]

location\_data = Total\_leg\_data['UserLocation'].value\_counts()

location\_data[2:15].plot(kind='bar', figsize=(14,7))

#In[]

plt.rcParams['figure.figsize'] = (18,4)

plt.rcParams['font.family'] = 'sans-serif'

text = Total\_leg\_data['TextData']

is\_sex = text.str.contains('sex')

is\_sex=is\_sex.astype(float)

is\_sex.plot()

#In[]

Total\_leg\_data=Total\_leg\_data.fillna(0)

Total\_leg\_data.shape

#In[]

temp1 = Total\_leg\_data[["UserFollowersCount"]]

temp1.to\_csv('temp1.csv', sep=',',encoding='utf8')

#In[]

Total\_leg\_data[['RetweetCount']] = Total\_leg\_data[['RetweetCount']].astype(float)

Total\_leg\_data.info()

#In[]

Total\_leg\_data = Total\_leg\_data[Total\_leg\_data.TweetCount!=0]

len(Total\_leg\_data[Total\_leg\_data.TweetCount<30])

#In[]

Total\_leg\_data.loc[:,"AvgHashtag"] = (Total\_leg\_data.groupby('UserID')["HashtagCount"].transform('sum'))/30

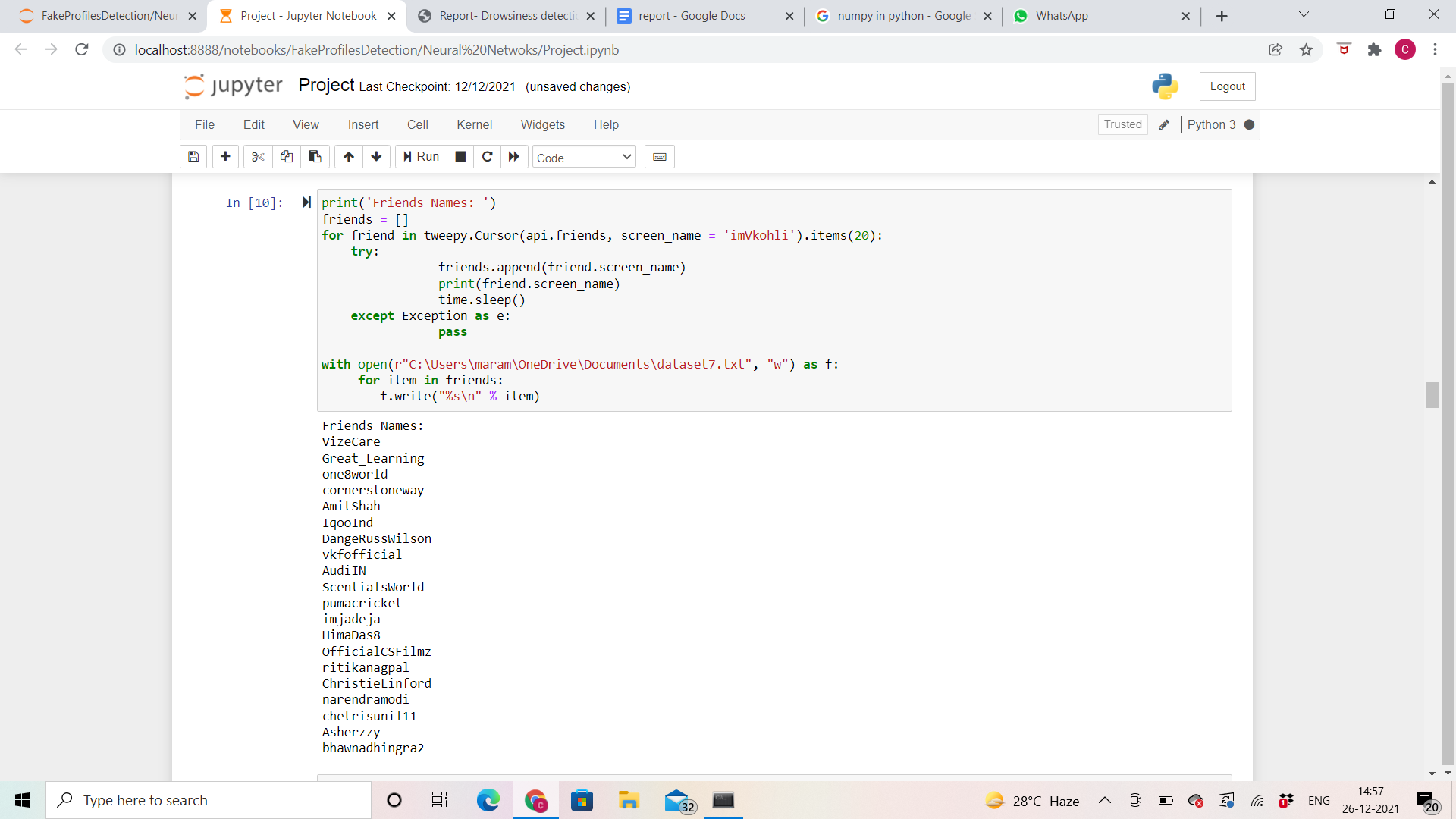
Total\_leg\_data.loc[:,"AvgURLCount"] = (Total\_leg\_data.groupby('UserID')["HttpCount"].transform('sum'))/30

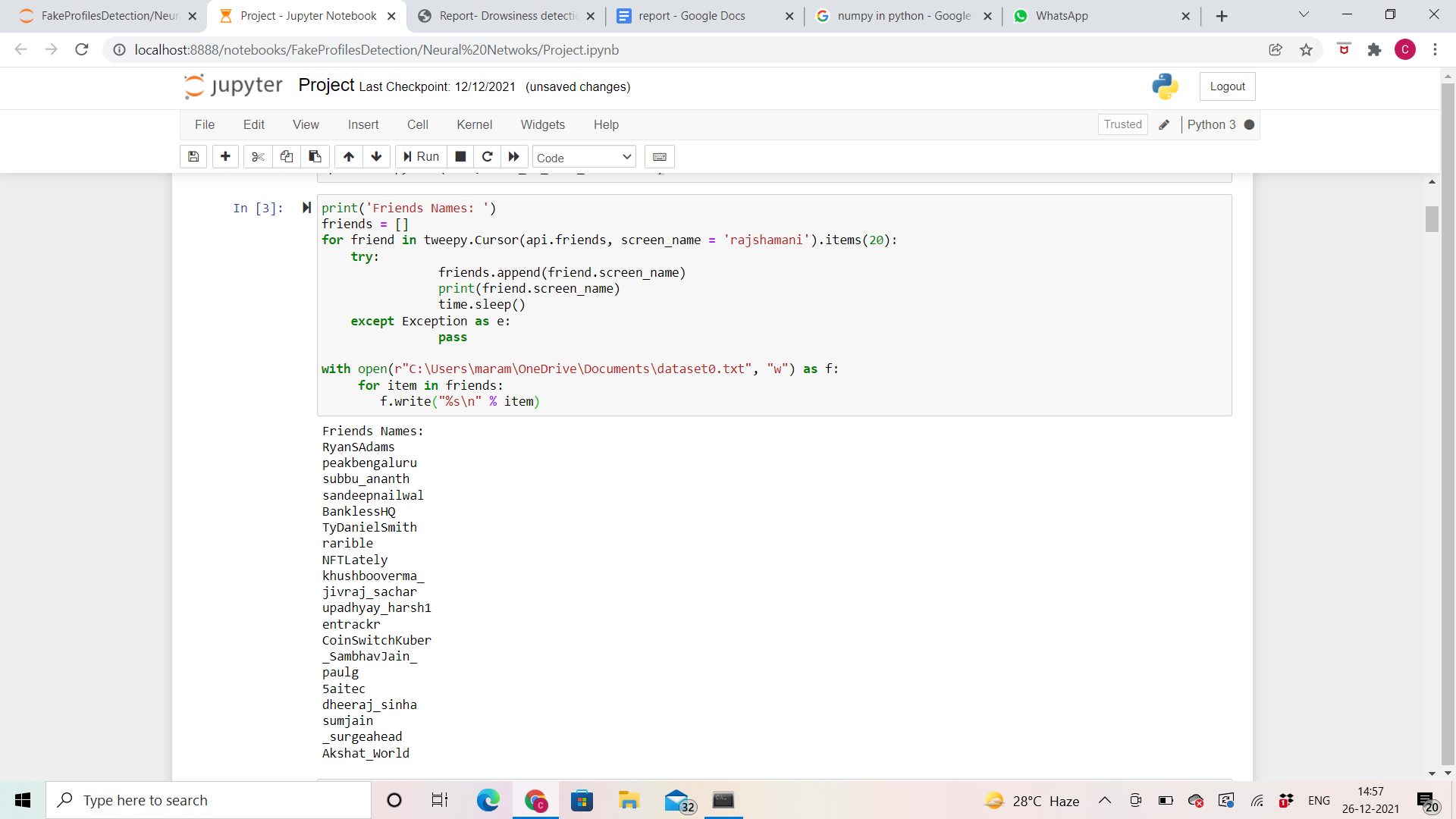
Total\_leg\_data.loc[:,"AvgMention"] = (Total\_leg\_data.groupby('UserID')["MentionCount"].transform('sum'))/30

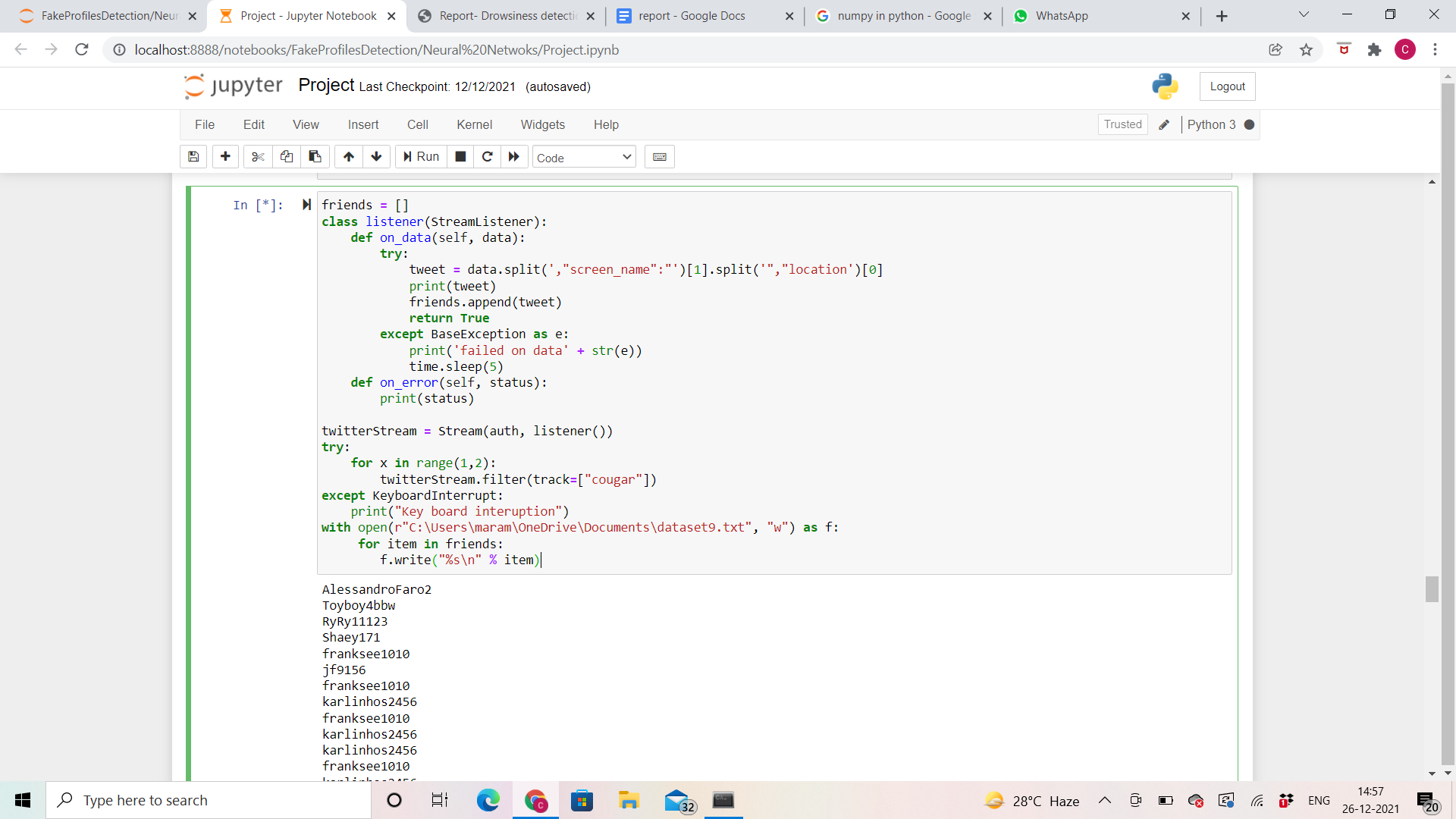
Total\_leg\_data.loc[:,"AvgRetweet"] = (Total\_leg\_data.groupby('UserID')["RetweetCount"].transform('sum'))/30

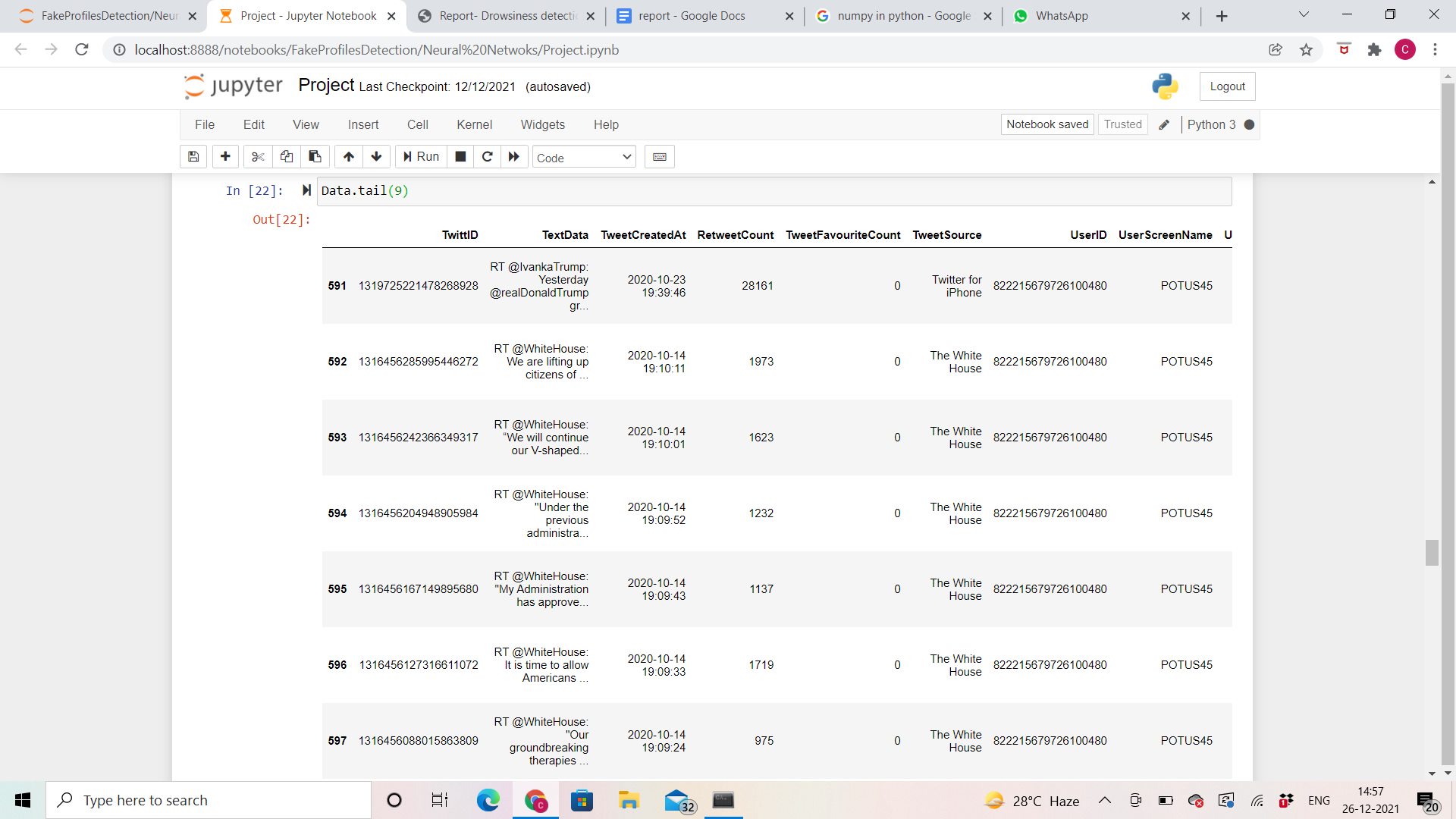
Total\_leg\_data.loc[:,"AvgFavCount"] = (Total\_leg\_data.groupby('UserID')["TweetFavouriteCount"].transform('sum'))/30

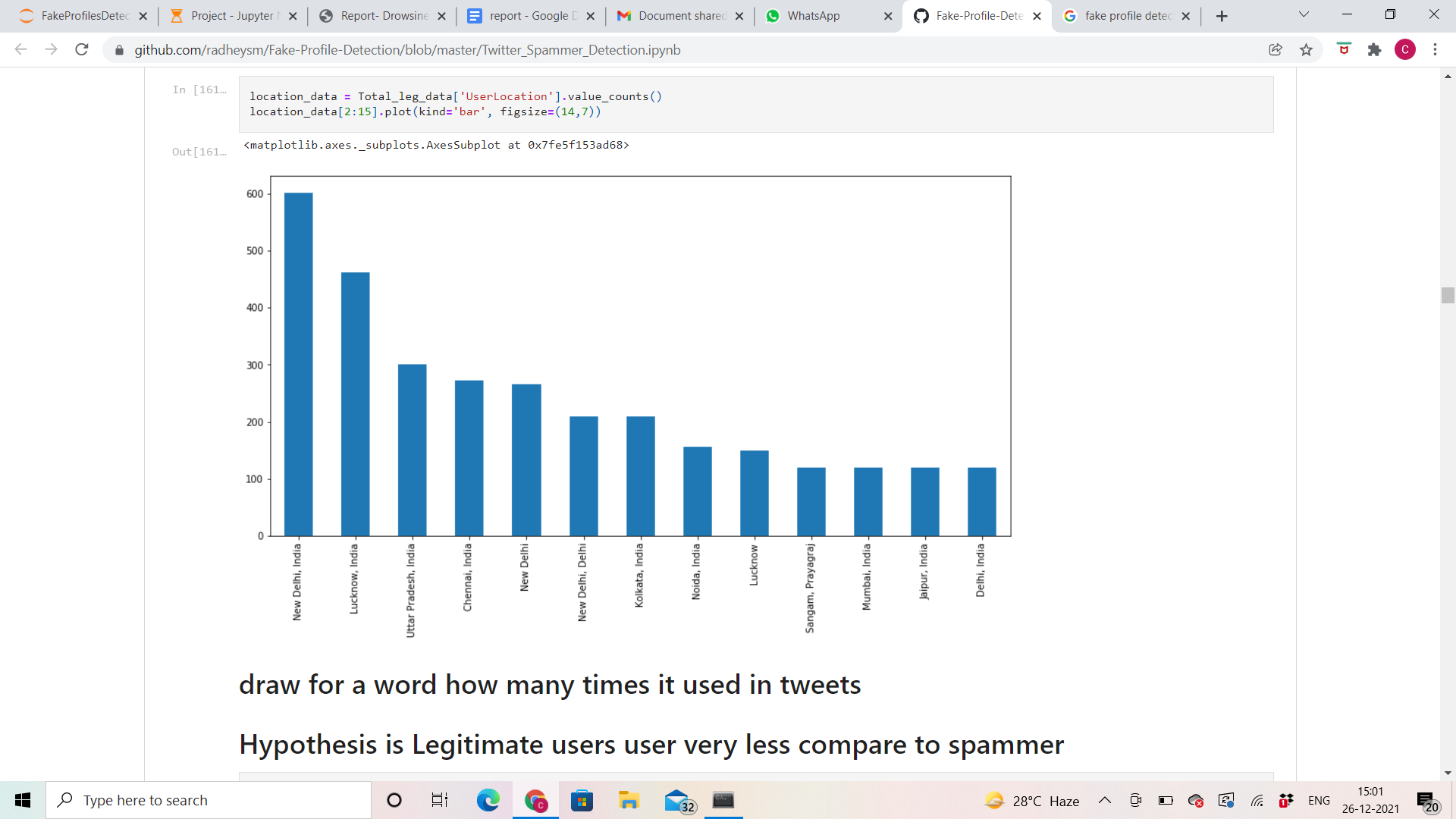
**Results:**

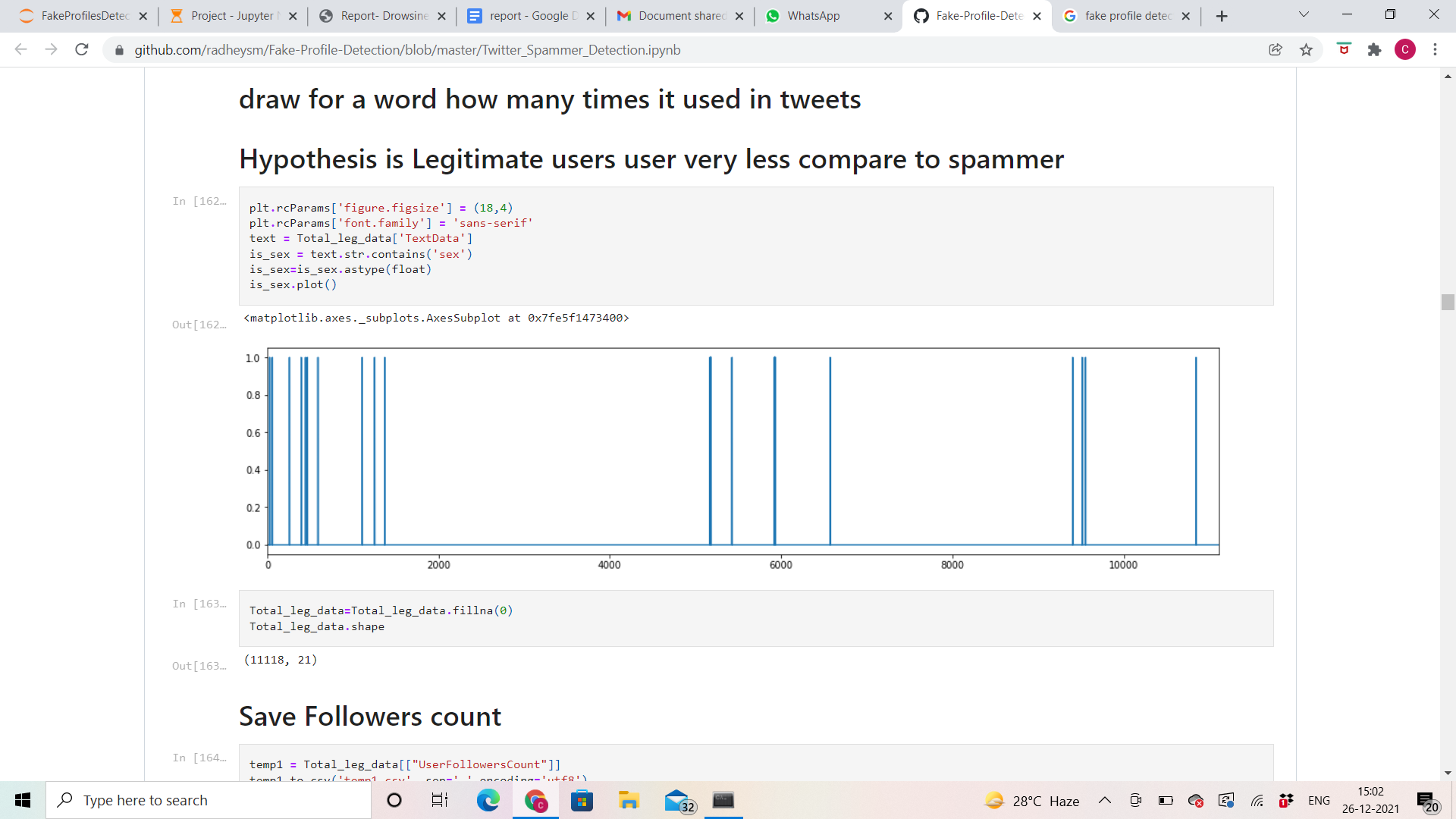












**7.Conclusion and Future Work**

In this paper, we proposed an approach to identify the fake profile in social-network using limited profile data, about 2816 users. As we concluded in our paper, we demonstrate that with limited profile data our approach can identify the fake profile with 99.64% correctly classified instances and only 0.35% incorrectly classified instances, which is comparable to the results obtained by other existing approaches based on the larger data set and more profile information. Our research can be a motivation to work on limited social network information and find solutions to make better decisions through authentic data. Additionally, we can attempt similar approaches in other domains to find successful solutions to the problem where the least amount of information is available. In future work we expect to run our model using more sophisticated concepts such as ontology engineering, in order to semantically analyze user posts, and comportments. This later concept can improve the quality of prediction of fake or genuine profiles.

**8.REFERENCES**

[**https://www.pantechsolutions.net/fake-profile-identification-using-machine-learning**](https://www.pantechsolutions.net/fake-profile-identification-using-machine-learning)

[**https://www.researchgate.net/publication/330629456\_Detecting\_Fake\_Accounts\_on\_Social\_Media**](https://www.researchgate.net/publication/330629456_Detecting_Fake_Accounts_on_Social_Media)

**GITHUB LINK:**

[**https://github.com/maramcharishma/FakeProfilesDetection**](https://github.com/maramcharishma/FakeProfilesDetection)