Twitter_Spammer_Detection

April 23, 2019

```
In [1]: #................................First Section........................................
        #Collection data from twitter for legitimate users
        #For collecting the data from twitter I am using Tweepy module
        #For that I need Counsumer_KEY, Counsumer_secret_KEY, Access_token, Access_Token
        #That all I can get from twitter app.devloper where I need to sign in and make an acco
        #After that a simple program in python can extract data from twitter in given limit by
```

```
In [105]: 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
```

```
In [106]: #Connection Authentication
```

```
In [107]: consumer_key  = 'd9Ksoz6Wb1jD0mqbW8rjaSNb7'
          consumer_secret = 'pHXnVSJeLb0xaYlbOR7BWFdDNhZSF6IzegZV87qUSUqy6Qe8qG'
          access_token = '3648603434-dGRu1nHet22tdoYeqaAGoN8MyZrNw9oXZQvGZUD'
          access_token_secret = 'PZ8pcQBCb5zVPLRQNVQZc3Yzi0rz1wPef607R07gzcv0f'

          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 [6]: #Collecting Data list of username of a given screen_name
        #Save data in txt file
```

```
In [7]: # printing all the friends names of the user
        print('Name of the Friends of user')
        friends = []
        for friend in tweepy.Cursor(api.friends, screen_name = 'PoliceRajasthan').items(20):
            try:
                        friends.append(friend.screen_name)
                        print(friend.screen_name)
                        time.sleep()
            except Exception as e:
                        pass

        with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend1.txt", "w") as f:
             for item in friends:
                f.write("%s\n" % item)
```

```
Name of the Friends of user
boxervijender
IndiaSports
unwomenindia
DainikBhaskar
MinistryWCD
BoomFactsHindi
PoliceJodhpur
PcrRural
AjmerPcr
pcrjaipurrural
PCRRajsamand
pcrnagaur
BharatpurPolice
AhmedabadPolice
dtptraffic
JprRuralPolice
Gulab_kataria
IgpJaipur
ChghPolice
PCR_Hanumangarh
```

```
In [8]: # printing all the friends names of the user
        print('Name of the Friends of user')
        friends = []
        for friend in tweepy.Cursor(api.friends, screen_name = 'Uppolice').items(20):
            try:
                        friends.append(friend.screen_name)
                        print(friend.screen_name)
                        time.sleep()
            except Exception as e:
                        pass

        with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend2.txt", "w") as f:
             for item in friends:
                f.write("%s\n" % item)
```

```
Name of the Friends of user
Dilipdubey03
upcopsachin
```

```
AnjanaPed
SkochSameer
mobobistudios
TAHLKANEWS
rakeshbjpup
CyberDost
spgrpjhansi
kumbhMelaPolUP
NBTMumbai
AtulGargBJP
sdrf_up
UD197
SantoshMahiLko
devmuraribapu65
wpl1090
927BIGFM
BPRDIndia
fireserviceup
```

```
In [9]: # printing all the friends names of the user
        print('Name of the Friends of user')
        friends = []
        for friend in tweepy.Cursor(api.friends, screen_name = 'MumbaiPolice').items(20):
            try:
                        friends.append(friend.screen_name)
                        print(friend.screen_name)
                        time.sleep()
            except Exception as e:
                        pass

        with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend3.txt", "w") as f:
             for item in friends:
                f.write("%s\n" % item)
```

```
Name of the Friends of user
PoliceWaliPblic
TawdeVinod
assampolice
cyberabadpolice
rpomumbai
PoliceRajasthan
Uppolice
TwitterIndia
KirenRijiju
ajaydevgn
Thane_R_Police
MahaDGIPR
```

```
BSF_India
AdlCPCrimeMum
narendramodi
DCPSangramsinh
DattaCP
ThaneCityPolice
Navimumpolice
IPS_Association
```

```
In [10]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'PunjabPoliceInd').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend4.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
trafficchd
KhannaPolice
CyberDost
RajaBrar_INC
faridkotpolice1
PPSM_SASNAGAR
PPASRR2
sspofficefazil1
MuktsarSsp
TarnTaranPolice
pp_sangrur
PpSbsn
PP_Patiala
pp_pathankot
moga_pp
pp_mansa
PP_Ldhrural
PPkhanna3
PP_kapurthala
SMCelljal_Rural
```

```
In [11]: # printing all the friends names of the user
```

```
print('Name of the Friends of user')
friends = []
for friend in tweepy.Cursor(api.friends, screen_name = 'KolkataPolice').items(20):
    try:
                friends.append(friend.screen_name)
                print(friend.screen_name)
                time.sleep()
    except Exception as e:
                pass

with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend5.txt", "w") as f:
     for item in friends:
        f.write("%s\n" % item)
```

```
Name of the Friends of user
RajeshKumarIPS
Neelsher
CyberDost
DcpEast
AltNews
CPKolkata
KPSouthsubnDiv
NICFS_India
KPCentralDiv
KPSouthwestDiv
KPSouthDiv
KPPortDiv
KPDetectiveDept
KPNorthDiv
KPSoutheastDiv
KPEastsubnDiv
BlrCityPolice
MumbaiPolice
DelhiPolice
KPTrafficDept
```

```
In [12]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'DelhiPolice').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass
         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend6.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
hgsdhaliwalips
EOWDelhi
LifeCoachSharat
DCP_DelhiMetro
nihar15aug
rashtrapatibhvn
ahir_hansraj
MOSHomeIndia
rajnathsingh
DCP_CCC_Delhi
NavbharatTimes
Outlookindia
ians_india
htTweets
KhabarNwi
indiatvnews
adcp1South
DCP_Shd
BaniwalDP
Ravindra_IPS
```

```
In [13]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'BlrCityPolice').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend7.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
BngWeather
UdhampurPolice
DHFWKA
digilocker_ind
BaramullaPolice
```

```
KashmirPolice
DistrictPolice1
Tripura_Police
JmuKmrPolice
PoliceRajasthan
assampolice
hydcitypolice
KolkataPolice
dtptraffic
AhmedabadPolice
GujaratPolice
CPDelhi
DDNewsLive
IAF_MCC
CISFHQrs
```

```
In [14]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'noidapolice').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend8.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
SiManojThakur1
VikramA79117869
assampolice
GuwahatiPol
GujaratPolice
AhmedabadPolice
fireserviceup
venkatashok
bareillytraffic
ECISVEEP
DmHapur
DEHRA_CHOKI
ceoup
uttarakhandcops
ProDixit
```

```
airnewsalerts
PIB_India
adgpi
NIA_India
BharatKeVeer
```

```
In [15]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'igrangemeerut').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend9.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
RajatSharmaLive
skochgroup
SkochSameer
kumbhMelaPolUP
Etahpolice
digbasti
digdevipatan
digmirzapur
ADGZonPrayagraj
adgzonevaranasi
adgzonekanpur
digmoradabad
igrangeagra
shravastipolice
gorakhpurpolice
kaushambipolice
hathraspolice
IgRangeVaranasi
sonbhadrapolice
jaunpurpolice
```

```
In [16]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'noidatraffic').items(20):
             try:
                             friends.append(friend.screen_name)
                             print(friend.screen_name)
                             time.sleep()
             except Exception as e:
                             pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend10.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
TrafficIg
sspnoida
ajay_sharmaips
ParivahanUP
UPPolNRI
ajay85ldh
NoidaUP100
SidharthNSingh
DainikBhaskar
arunjaitley
JagranNews
ptshrikant
rajnathsingh
drdineshbjp
HMOIndia
News18UP
kpmaurya1
myogiadityanath
HomeDepttUP
SspGhaziabad
```

```
In [17]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'adgzonemeerut').items(20):
             try:
                             friends.append(friend.screen_name)
                             print(friend.screen_name)
                             time.sleep()
             except Exception as e:
                             pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend11.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
chandanmedia
SspGhaziabad
shivpal_rana
policemedianews
uppstf
indiatvnews
EconomicTimes
BBCHindi
airnewsalerts
News18India
abpnewstv
ndtv
ZeeNewsHindi
THexplains
TOIIndiaNews
NavbharatTimes
TheOfficialSBI
ndtvindia
DDNewsHindi
News18_UK
```

```
In [18]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'meerutpolice').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend12.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
bijnorpolice
HsyTimes
CyberDost
Ariffaizylawar
Uppolice
UPPViralCheck
UPPolNRI
ASTITV17
dgpup
```

```
myogiadityanath
rashtrapatibhvn
DainikBhaskar
AmarUjalaNews
_NationalVoice
TwitterIndia
shravastipolice
ANI
jhansipolice
IASassociation
ZeeNewsHindi
```

```
In [19]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'bulandshahrpol').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend13.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
ghazipurpolice
ambedkarnagrpol
faizabadpolice
News18UP
Barabankipolice
sitapurpolice
bahraichpolice
bhadohipolice
gondapolice
balrampurpolice
bastipolice
gorakhpurpolice
kushinagarpol
santkabirnagpol
fatehgarhpolice
auraiyapolice
etawahpolice
chitrakootpol
jalaunpolice
```

```
jhansipolice
```

```
In [20]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'saharanpurpol').items(20):
             try:
                             friends.append(friend.screen_name)
                             print(friend.screen_name)
                             time.sleep()
             except Exception as e:
                             pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend14.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
skochgroup
SkochSameer
samayupuk
Dineshdcop
DeepakKumarIPS2
smittal_ips
dm_ghaziabad
Anubhav26266011
upcopvishal
UPPViralCheck
RubyTomar14
LalitPayal
UPPolNRI
ASTITV17
SHO_JEWAR
sundersaini1
YASMinistry
ndmaindia
eShineNews
MinOfPower
```

```
In [21]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'shamlipolice').items(20):
             try:
                             friends.append(friend.screen_name)
                             print(friend.screen_name)
```

```
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend15.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
GaonConnection
bstvlive
ajay85ldh
skochgroup
SkochSameer
varanasitraffic
ShamliTraffic
OP_Singh83
CyberDost
kumbhMelaPolUP
abpnewstv
ZeeNews
aajtak
ndtvindia
samachartoday4u
mediaamantra
DainikBhaskar
ZeeNewsHindi
News18India
allahabdtraffic
```

```
In [22]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'hapurpolice').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend16.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
CyberDost
```

```
upgrp
skochgroup
SkochSameer
sangamchaudha20
deeepak34093
Aalam__Ansari
HNN24X7
kumbhMelaPolUP
Rahulsiupp
DmHapur
NewsStateHindi
_ShivamBhatt
rjraunac
PMOIndia
DelhiTimesTweet
AjayendraR
UPGovt
narendramodi
MinistryWCD
```

```
In [23]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'baghpatpolice').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend17.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
CyberDost
UPPolNRI
dgpup
dtptraffic
PMOIndia
HMOIndia
DelhiPolice
IPS_Association
up100
igrangealld
igrangeagra
```

```
igrangemeerut
adgzoneagra
digrangealigarh
SspGhaziabad
upcoprahul
uptrafficpolice
noidapolice
ChiefSecyUP
CMOfficeUP
```

```
In [24]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'CCTPolice_Alert').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend18.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
THChennai
news18dotcom
polimernews
MalaimurasuTv
newsglitzcom
tangedconews
Suyaatchi
Arappor
IndiaTodayFLASH
NatarajIPS
vikatan
fx16pix
deccanchennai
DeccanChronicle
PTI_News
BBCIndia
tamil_murasu
TamilTheHindu
maalaimalar
timesofindia
```

```
In [25]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'chennaipolice_').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass

         with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend19.txt", "w") as f:
              for item in friends:
                 f.write("%s\n" % item)
```

```
Name of the Friends of user
Andrew_Sesuraj
KarthiAk57
anilachankunju
itisaprashanth
anilkunju
Vishnuaiadmk1
admk_surya
AdmkSivaranjan
rajivgandhi_n
prabhaayyappan
admk_satheesh
Sai72100878
AdmkArun
SelvaMugavai
Veerasa23144200
SelvamAdmk
VHh0ryw5wTWZQgV
Jaganat39464129
maalaitamizhaga
vijayadmk3
```

```
In [26]: # printing all the friends names of the user
         print('Name of the Friends of user')
         friends = []
         for friend in tweepy.Cursor(api.friends, screen_name = 'hydcitypolice').items(20):
             try:
                         friends.append(friend.screen_name)
                         print(friend.screen_name)
                         time.sleep()
             except Exception as e:
                         pass
```

```
with open("/home/radhey/Final_Project/Data/Leg_User_txt/friend20.txt", "w") as f:
     for item in friends:
        f.write("%s\n" % item)
```

```
Name of the Friends of user
NameisNani
lrvr1974
skochgroup
MLA54327644
WomenCid
cpkarimnagar
cpwrlc
CyberProtectUK
spsangareddy
spsuryapet
cpramagundam
sp_kamareddy
cp_nizamabad
Vikarabadpolice
spsiricilla
CPRODGPTS
ndmaindia
InsptrJbh
NICMeity
CyberDost
```

```
In [27]: #Now collect 30 tweet from each user that I extracted from twitter
```

```
In [129]: Total_Data = []
          fo = open("/home/radhey/Final_Project/Data/Leg_User_txt/friend20.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))
```

```
535
```

```
In [130]: #Now from tweet extract useful atributes
```

```
In [131]: 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 Tota
    Data["UserFollowersCount"] = [tweet.author.followers_count for tweet in Total_Da
    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_Da
    # 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
```

```
Out[131]: (535, 19)
```

```
In [132]: Data.tail(4)
```

```
Out[132]:                  TwittID                                          TextData  \
          531  1080812656922046465  RT @russi109: Ministry Of Home Affairs (Govt o...
          532  1080812530992173061  RT @IamHiteshB: For awareness of cyber crimes,...
          533  1080811435293237249  RT @JagdishDewasi07: For awareness of cyber cr...
          534  1080811372202446848  @the_ajitsingh Dear Sir/Ma'am, you can report ...

                    TweetCreatedAt  RetweetCount  TweetFavouriteCount  \
          531 2019-01-03 13:06:32            17                    0
          532 2019-01-03 13:06:02            22                    0
          533 2019-01-03 13:01:41            11                    0
          534 2019-01-03 13:01:26             0                    2

                       TweetSource              UserID UserScreenName    UserName  \
          531  Twitter Web Client  970591741131804672      CyberDost  Cyber Dost
          532  Twitter Web Client  970591741131804672      CyberDost  Cyber Dost
          533  Twitter Web Client  970591741131804672      CyberDost  Cyber Dost
          534  Twitter Web Client  970591741131804672      CyberDost  Cyber Dost

                     UserCreatedAt                                     UserDescription  \
          531 2018-03-05 09:27:58  https://t.co/CS0TpWjXGS                         ...
          532 2018-03-05 09:27:58  https://t.co/CS0TpWjXGS                         ...
          533 2018-03-05 09:27:58  https://t.co/CS0TpWjXGS                         ...
          534 2018-03-05 09:27:58  https://t.co/CS0TpWjXGS                         ...
```

```
     UserDescriptionLength  UserFollowersCount  UserFriendsCount UserLocation  \
531                    156               54216                76        India
532                    156               54216                76        India
533                    156               54216                76        India
534                    156               54216                76        India

     HttpCount  HashtagCount  MentionCount  TweetCount
531          0             0             2         432
532          0             2             2         432
533          0             2             2         432
534          1             0             1         432
```

```
In [133]: # Save data in csv_files
```

```
In [134]: import sys
          # Saving data with item space separating
          Data.to_csv('/home/radhey/Final_Project/Data/Leg_User_csv/friend20.csv', sep=',' , en
```

```
In [135]: # extracting Spam data from twitter by searching @spam and find out the user for repo
          #hypothesis is that there is highly chances is that that user be fake
          #We can later analyse by text any volgor word and find our later first like legitima
```

```
In [155]: # printing all the friends names of the user

          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(1,10):
                  twitterStream.filter(track=["cougar"])
          except KeyboardInterrupt:
              print("Key board interuption")
          with open("/home/radhey/Final_Project/Data/Spam_User_text/spam12.txt", "w") as f:
               for item in friends:
                  f.write("%s\n" % item)
```

```
kolot_50
WorshipAdmin
```

```
CougarSora
Ruin2day
JamesALogan1
AmericanGoldSPP
BSherSB
Key board interuption
cat: stream.txt: No such file or directory
```

```
In [156]: #Now collect 30 tweet from each spam user that I extracted from twitter
```

```
In [206]: Total_Data = []
          fo = open("/home/radhey/Final_Project/Data/Spam_User_text/spam12.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))
```

```
210
```

```
In [207]: #Now from tweet extract useful atributes
```

```
In [208]: 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 Tota
    Data["UserFollowersCount"] = [tweet.author.followers_count for tweet in Total_Da
    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_Da
    # 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

Out[208]: (210, 19)

In [209]: # Save data in csv_files

In [210]: # Saving data with item space separating
          Data.to_csv('/home/radhey/Final_Project/Data/Spam_User_csv/spam10.csv', sep=',' , en

In [211]: #First Merge all Data csv files both legitimate or Spammer

In [30]: import csv
         import glob
         import os
         # get data file names
         path = '/home/radhey/Final_Project/Data/Leg_User_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)

Out[30]:        Unnamed: 0              TwittID  \
         11114         569  1057136047337943041
         11115         570  1055763859154370562
         11116         571  1055699969884217344
         11117         572  1055063501109104640


                                                           TextData       TweetCreatedAt  \
         11114  @ravijansaamna @uptrafficpolice @adgzonealld @...  2018-10-30 05:04:08
         11115  RT @igrangealld:  25.10.2018   ...  2018-10-26 10:11:33
         11116  @RahulBhasin17 @AllahabadAdmin1 @allahabadpoli...  2018-10-26 05:57:41
         11117  @ToRahulKapoor @parvaiz_alam   ...  2018-10-24 11:48:35

                RetweetCount  TweetFavouriteCount         TweetSource      UserID  \
         11114             0                    0  Twitter Web Client  3266889528
         11115            21                    0  Twitter Web Client  3266889528
         11116             1                    3  Twitter Web Client  3266889528
         11117             0                    1  Twitter Web Client  3266889528

                 UserScreenName                  UserName         UserCreatedAt  \
         11114  allahabdtraffic  Traffic Police Prayagraj  2015-07-03 09:06:39
         11115  allahabdtraffic  Traffic Police Prayagraj  2015-07-03 09:06:39
         11116  allahabdtraffic  Traffic Police Prayagraj  2015-07-03 09:06:39
         11117  allahabdtraffic  Traffic Police Prayagraj  2015-07-03 09:06:39

                                                  UserDescription  \
11114  Official Twitter account of Allahabad #Traffic...
11115  Official Twitter account of Allahabad #Traffic...
11116  Official Twitter account of Allahabad #Traffic...
11117  Official Twitter account of Allahabad #Traffic...

       UserDescriptionLength  UserFollowersCount  UserFriendsCount  \
11114                    138                7608               146
11115                    138                7608               146
11116                    138                7608               146
11117                    138                7608               146

           UserLocation  HttpCount  HashtagCount  MentionCount  TweetCount
11114  Allahabad, India          0             0             4        4937
11115  Allahabad, India          0             0             1        4937
11116  Allahabad, India          1             0             5        4937
11117  Allahabad, India          1             0             2        4937
```

```
In [31]: Total_leg.to_csv('/home/radhey/Final_Project/Leg_data.csv', sep=',' , encoding='utf8'
```

```
In [32]: # Merging Spammer Data
         import csv
         import glob
         import os
         # get data file names
         path = '/home/radhey/Final_Project/Data/Spam_User_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)
```

```
Out[32]:       Unnamed: 0              TwittID  \
         5390         746  1120300621578551296
         5391         747  1120300607309524992
         5392         748  1120300592046444545
         5393         749  1120300537314979840


                                                 TextData        TweetCreatedAt  \
         5390  RT @s___fire: your sex life is going bad ? you...  2019-04-22 12:17:37
         5391  RT @s___fire: Find your fantasy here and make ...  2019-04-22 12:17:33
         5392  RT @sexole: ONLINE EN https://t.co/wkT9BMovtL ...  2019-04-22 12:17:29
         5393  RT @DomUrch: @irinagomez60\n@HQPornHQ\n@Erotik...  2019-04-22 12:17:16

               RetweetCount  TweetFavouriteCount          TweetSource      UserID  \
         5390            22                    0  Twitter for Android  1055696622
         5391            18                    0  Twitter for Android  1055696622
         5392             1                    0  Twitter for Android  1055696622
         5393           121                    0  Twitter for Android  1055696622

              UserScreenName   UserName        UserCreatedAt UserDescription  \
         5390    Giovannini8  giancarlo  2013-01-02 17:56:31             NaN
         5391    Giovannini8  giancarlo  2013-01-02 17:56:31             NaN
         5392    Giovannini8  giancarlo  2013-01-02 17:56:31             NaN
         5393    Giovannini8  giancarlo  2013-01-02 17:56:31             NaN

               UserDescriptionLength  UserFollowersCount  UserFriendsCount  \
         5390                      0                1755              2130
         5391                      0                1755              2130
         5392                      0                1755              2130
         5393                      0                1755              2130

              UserLocation  HttpCount  HashtagCount  MentionCount  TweetCount
         5390          NaN          1             0             1      150737
         5391          NaN          1             0             1      150737
         5392          NaN          2             2             1      150737
         5393          NaN          0             0            11      150737
```

```
In [33]: Total_leg.to_csv('/home/radhey/Final_Project/Spam_data.csv', sep=',' , encoding='utf8
```

```
In [301]: concatenate()
```

```
friend18.csv
friend19.csv
friend11.csv
friend2.csv
friend9.csv
friend7.csv
friend8.csv
friend20.csv
friend10.csv
friend12.csv
friend14.csv
friend3.csv
friend6.csv
friend5.csv
friend13.csv
friend4.csv
friend16.csv
friend1.csv
friend17.csv
friend15.csv
```

```
In [212]: #.....................Section Seconfd........................
          # lodading legitimate User Data
```

```
In [34]: import pandas as pd
         Total_leg_data = pd.read_csv('Leg_data.csv')
         Total_leg_data.fillna(0, inplace=True)
         Total_leg_data.shape

Out[34]: (11118, 21)

In [35]: Total_leg_data.head(2)

Out[35]:    Unnamed: 0  Unnamed: 0.1              TwittID  \
         0           0             0  1120183242387120128
         1           1             1  1119860017400664065

                                                      TextData       TweetCreatedAt  \
         0  RT @Rama_Krishnan: Candidates of @ammkofficial...  2019-04-22 04:31:11
         1  RT @TheHinduCinema: Even though hed prefer to...  2019-04-21 07:06:48

            RetweetCount  TweetFavouriteCount TweetSource     UserID UserScreenName  \
         0             6                    0   TweetDeck  613357772      THChennai
         1            10                    0   TweetDeck  613357772      THChennai

            ...        UserCreatedAt  \
         0  ...  2012-06-20 11:24:09
         1  ...  2012-06-20 11:24:09

                                             UserDescription UserDescriptionLength  \
         0  The official twitter account of The Hindu's re...                   145
         1  The official twitter account of The Hindu's re...                   145

            UserFollowersCount  UserFriendsCount    UserLocation HttpCount  \
         0               62144               297  Chennai, India         0
         1               62144               297  Chennai, India         0

            HashtagCount  MentionCount  TweetCount
         0             0             2       21157
         1             1             1       21157

         [2 rows x 21 columns]

In [8]: colname=['Unnamed: 0','Unnamed: 1','TwittID', 'TextData', 'TweetCreatedAt','RetweetCou
        Total_leg_data.columns=colname
        Total_leg_data.head(2)

Out[8]:    Unnamed: 0  Unnamed: 1       TwittID  \
        0           0         NaN  0.000000e+00
        1           1         0.0  1.120183e+18

                                                     TextData       TweetCreatedAt  \
        0                                                   0                    0
```

```
1  RT @Rama_Krishnan: Candidates of @ammkofficial...  2019-04-22 04:31:11

  RetweetCount TweetFavouriteCount TweetSource     UserID UserScreenName  ...  \
0            0                   0           0          0              0  ...
1            6                   0   TweetDeck  613357772       THChennai  ...

  UserFollowersCount UserFriendsCount    UserLocation HttpCount HashtagCount  \
0                  0                0               0         0            0
1              62144              297  Chennai, India         0            0

  MentionCount TweetCount Unnamed: 21 Unnamed: 22 Unnamed: 23
0            0          0         NaN         NaN         NaN
1            2      21157         NaN         NaN         NaN

[2 rows x 24 columns]
```

```
In [26]: #drop Unused columns
         #Total_leg_data.drop("Unnamed: 23", axis=1, inplace=True)
         Total_leg_data = Total_leg_data.drop([0], axis=0)
```

```
In [27]: Total_leg_data
```

```
Out[27]:             TwittID                                           TextData  \
         1      1.120183e+18  RT @Rama_Krishnan: Candidates of @ammkofficial...
         2      1.119860e+18  RT @TheHinduCinema: Even though hed prefer to...
         3      1.119853e+18  RT @rsujatha_30: Schedule released for Tamilna...
         4      1.119833e+18  RT @rsujatha_30: DOTE to conduct online counse...
         5      1.119568e+18  Here's one of the earliest of his column Madra...
         6      1.119567e+18  Bishwanath Ghosh writes on S. Muthiah on the o...
         7      1.119566e+18  Just in | S. Muthiah, chronicler of Chennai's ...
         8      1.119487e+18  RT @dipakragav: Just in : Maggie Amritraj, mot...
         9      1.119140e+18  @the_hindu @dsureshkumar Read The Hindu's repo...
         10     1.119139e+18  The Election Commission of India has sought fo...
         11     1.119130e+18  TN Higher Secondary Certificate examination re...
         12     1.119093e+18  RT @_poorvaja: Last year, the pass percentage ...
         13     1.119090e+18  RT @_poorvaja: Tiruppur tops the districts wit...
         14     1.119090e+18  RT @_poorvaja: Plus 2 board exam results annou...
         15     1.118795e+18  RT @the_hindu: #LokSabhaElections2019: Newlywe...
         16     1.118794e+18  RT @SunithaSekar: Did anyone in #Chennai cast ...
         17     1.118730e+18  RT @Teekkayy: 13.48% polling in #Tamilnadu til...
         18     1.118730e+18  Makal Needhi Maiyam president  @ikamalhaasan w...
         19     1.118724e+18  RT @SunithaSekar: DMK leader M.K. Stalin and h...
         20     1.118711e+18  RT @the_hindu: #LokSabhaElections2019: Enthusi...
         21     1.118710e+18  RT @the_hindu: #LokSabhaElections2019: Selvi R...
         22     1.118708e+18  RT @the_hindu: Makal Needhi Maiyam president  ...
         23     1.118693e+18  RT @_poorvaja: Actors Ajith, Shalini and Rajin...
         24     1.117805e+18  RT @sang1983: The Income Tax department  Inve...
         25     1.117735e+18  RT @imranhindu: Madras HC directs TN Govt to v...
```

```
26     1.117285e+18  #LokSabhaElection2019 | The road map titled J...
27     1.116581e+18  RT @imranhindu: An astrologer moves Madras HC ...
28     1.116217e+18  RT @imranhindu: Madras HC refuses to grant int...
29     1.115482e+18  RT @imranhindu: Madras HC directs TN Govt to p...
30     1.115159e+18  RT @the_hindu: A Division Bench quashed the pr...
...             ...                                               ...
11108  1.065543e+18   RT @Uppolice: #UPPInNews https://t.co/PaV9BnPLEx
11109  1.065203e+18   RT @Uppolice: #UPPInNews https://t.co/needcWeWUj
11110  1.065203e+18  RT @Uppolice: Know road safety, No injury. No ...
11111  1.065168e+18  @utkarsh2993 @uptrafficpolice @adgzonealld @ig...
11112  1.064826e+18     :   #trafficm...
11113  1.064819e+18  @SheikhAjmalAhm2 @allahabadpolice @up100 @dgpu...
11114  1.064503e+18  @drnkagrawal @CMOfficeUP @uptrafficpolice ...
11115  1.064499e+18  RT @allahabadpolice: facebook    14 ...
11116  1.064499e+18  RT @Uppolice:   @faizabadpolice  ...
11117  1.064499e+18  RT @adgzonealld:   19/11/2018  ...
11118  1.064489e+18     :   19.11.2018  ...
11119  1.063662e+18  @pankajvermacs @uptrafficpolice @adgzonealld @...
11120  1.063406e+18    : ....    17.11...
11121  1.063405e+18     :    ...
11122  1.063127e+18    :-   /     ...
11123  1.063039e+18  RT @Uppolice: #UPPInNews #uppolice https://t.c...
11124  1.062657e+18  RT @Uppolice: #DGPUP addressed school children...
11125  1.062632e+18  RT @dharmveerinfo: ADG  .., ...
11126  1.062607e+18  @drnkagrawal     ...
11127  1.062607e+18  @amitkiransingh @dharmveerinfo @DM_PRAYAGRAJ @...
11128  1.062315e+18  RT @Uppolice:        ...
11129  1.059746e+18  RT @Uppolice:       ...
11130  1.059413e+18  @Uppolice @uptrafficpolice @allahabadpolice @a...
11131  1.058585e+18  RT @Uppolice:      ...
11132  1.058310e+18    ,     ...
11133  1.058302e+18  @utkarsh2993 @allahabadpolice   ...
11134  1.057136e+18  @ravijansaamna @uptrafficpolice @adgzonealld @...
11135  1.055764e+18  RT @igrangealld:  25.10.2018   ...
11136  1.055700e+18  @RahulBhasin17 @AllahabadAdmin1 @allahabadpoli...
11137  1.055064e+18  @ToRahulKapoor @parvaiz_alam   ...

                          TweetCreatedAt     RetweetCount  \
1                    2019-04-22 04:31:11  <class 'float'>
2                    2019-04-21 07:06:48  <class 'float'>
3                    2019-04-21 06:38:53  <class 'float'>
4                    2019-04-21 05:17:33  <class 'float'>
5                    2019-04-20 11:45:34  <class 'float'>
6                    2019-04-20 11:43:57  <class 'float'>
7                    2019-04-20 11:40:23  <class 'float'>
8                    2019-04-20 06:26:12  <class 'float'>
9                    2019-04-19 07:24:57  <class 'float'>
10                   2019-04-19 07:21:30  <class 'float'>
```

```
11                    2019-04-19 06:45:00  <class 'float'>
12                    2019-04-19 04:17:17  <class 'float'>
13                    2019-04-19 04:07:01  <class 'float'>
14                    2019-04-19 04:06:47  <class 'float'>
15                    2019-04-18 08:36:43  <class 'float'>
16                    2019-04-18 08:30:18  <class 'float'>
17                    2019-04-18 04:14:57  <class 'float'>
18                    2019-04-18 04:14:53  <class 'float'>
19                    2019-04-18 03:50:45  <class 'float'>
20                    2019-04-18 02:59:23  <class 'float'>
21                    2019-04-18 02:57:20  <class 'float'>
22                    2019-04-18 02:50:34  <class 'float'>
23                    2019-04-18 01:49:10  <class 'float'>
24                    2019-04-15 14:59:48  <class 'float'>
25      immovable assets of teaching &amp  <class 'float'>
26                    2019-04-14 04:34:28  <class 'float'>
27                    2019-04-12 05:56:31  <class 'float'>
28                    2019-04-11 05:49:48  <class 'float'>
29                    2019-04-09 05:10:04  <class 'float'>
30                    2019-04-08 07:47:19  <class 'float'>
...                                   ...              ...
11108                 2018-11-22 09:50:28  <class 'float'>
11109                 2018-11-21 11:18:38  <class 'float'>
11110                 2018-11-21 11:18:21  <class 'float'>
11111                 2018-11-21 09:00:36  <class 'float'>
11112                 2018-11-20 10:19:34  <class 'float'>
11113                 2018-11-20 09:53:06  <class 'float'>
11114                 2018-11-19 12:58:38  <class 'float'>
11115                 2018-11-19 12:41:10  <class 'float'>
11116                 2018-11-19 12:40:58  <class 'float'>
11117                 2018-11-19 12:40:50  <class 'float'>
11118                 2018-11-19 12:01:29  <class 'float'>
11119                 2018-11-17 05:14:46  <class 'float'>
11120                 2018-11-16 12:18:26  <class 'float'>
11121                 2018-11-16 12:15:00  <class 'float'>
11122                 2018-11-15 17:51:10  <class 'float'>
11123                 2018-11-15 12:02:03  <class 'float'>
11124                 2018-11-14 10:41:58  <class 'float'>
11125                 2018-11-14 09:01:32  <class 'float'>
11126                 2018-11-14 07:23:40  <class 'float'>
11127                 2018-11-14 07:21:58  <class 'float'>
11128                 2018-11-13 12:02:30  <class 'float'>
11129                 2018-11-06 09:55:47  <class 'float'>
11130                 2018-11-05 11:52:44  <class 'float'>
11131                 2018-11-03 05:02:24  <class 'float'>
11132                 2018-11-02 10:47:33  <class 'float'>
11133                 2018-11-02 10:16:25  <class 'float'>
11134                 2018-10-30 05:04:08  <class 'float'>
```

```
11135                 2018-10-26 10:11:33  <class 'float'>
11136                 2018-10-26 05:57:41  <class 'float'>
11137                 2018-10-24 11:48:35  <class 'float'>

      TweetFavouriteCount          TweetSource      UserID   UserScreenName  \
1                       0            TweetDeck   613357772        THChennai
2                       0            TweetDeck   613357772        THChennai
3                       0            TweetDeck   613357772        THChennai
4                       0            TweetDeck   613357772        THChennai
5                       5            TweetDeck   613357772        THChennai
6                       7            TweetDeck   613357772        THChennai
7                      68            TweetDeck   613357772        THChennai
8                       0            TweetDeck   613357772        THChennai
9                       9            TweetDeck   613357772        THChennai
10                     24            TweetDeck   613357772        THChennai
11                     34            TweetDeck   613357772        THChennai
12                      0            TweetDeck   613357772        THChennai
13                      0            TweetDeck   613357772        THChennai
14                      0            TweetDeck   613357772        THChennai
15                      0            TweetDeck   613357772        THChennai
16                      0            TweetDeck   613357772        THChennai
17                      0            TweetDeck   613357772        THChennai
18                    173            TweetDeck   613357772        THChennai
19                      0            TweetDeck   613357772        THChennai
20                      0            TweetDeck   613357772        THChennai
21                      0            TweetDeck   613357772        THChennai
22                      0            TweetDeck   613357772        THChennai
23                      0            TweetDeck   613357772        THChennai
24                      0            TweetDeck   613357772        THChennai
25                 aided  2019-04-15 10:23:45          19                0
26                     31            TweetDeck   613357772        THChennai
27                      0            TweetDeck   613357772        THChennai
28                      0            TweetDeck   613357772        THChennai
29                      0            TweetDeck   613357772        THChennai
30                      0            TweetDeck   613357772        THChennai
...                   ...                  ...         ...              ...
11108                   0   Twitter Web Client  3266889528  allahabdtraffic
11109                   0   Twitter Web Client  3266889528  allahabdtraffic
11110                   0   Twitter Web Client  3266889528  allahabdtraffic
11111                   2   Twitter Web Client  3266889528  allahabdtraffic
11112                   6   Twitter Web Client  3266889528  allahabdtraffic
11113                   2   Twitter Web Client  3266889528  allahabdtraffic
11114                   2   Twitter Web Client  3266889528  allahabdtraffic
11115                   0   Twitter Web Client  3266889528  allahabdtraffic
11116                   0   Twitter Web Client  3266889528  allahabdtraffic
11117                   0   Twitter Web Client  3266889528  allahabdtraffic
11118                   3   Twitter Web Client  3266889528  allahabdtraffic
11119                   1   Twitter Web Client  3266889528  allahabdtraffic
```

```
11120                   6   Twitter Web Client  3266889528  allahabdtraffic
11121                   5   Twitter Web Client  3266889528  allahabdtraffic
11122                  11      Twitter Web App  3266889528  allahabdtraffic
11123                   0   Twitter Web Client  3266889528  allahabdtraffic
11124                   0   Twitter Web Client  3266889528  allahabdtraffic
11125                   0   Twitter Web Client  3266889528  allahabdtraffic
11126                   0   Twitter Web Client  3266889528  allahabdtraffic
11127                   0   Twitter Web Client  3266889528  allahabdtraffic
11128                   0   Twitter Web Client  3266889528  allahabdtraffic
11129                   0   Twitter Web Client  3266889528  allahabdtraffic
11130                   3   Twitter Web Client  3266889528  allahabdtraffic
11131                   0   Twitter Web Client  3266889528  allahabdtraffic
11132                  34   Twitter Web Client  3266889528  allahabdtraffic
11133                   0   Twitter Web Client  3266889528  allahabdtraffic
11134                   0   Twitter Web Client  3266889528  allahabdtraffic
11135                   0   Twitter Web Client  3266889528  allahabdtraffic
11136                   3   Twitter Web Client  3266889528  allahabdtraffic
11137                   1   Twitter Web Client  3266889528  allahabdtraffic

                       UserName        UserCreatedAt  \
1           The Hindu - Chennai  2012-06-20 11:24:09
2           The Hindu - Chennai  2012-06-20 11:24:09
3           The Hindu - Chennai  2012-06-20 11:24:09
4           The Hindu - Chennai  2012-06-20 11:24:09
5           The Hindu - Chennai  2012-06-20 11:24:09
6           The Hindu - Chennai  2012-06-20 11:24:09
7           The Hindu - Chennai  2012-06-20 11:24:09
8           The Hindu - Chennai  2012-06-20 11:24:09
9           The Hindu - Chennai  2012-06-20 11:24:09
10          The Hindu - Chennai  2012-06-20 11:24:09
11          The Hindu - Chennai  2012-06-20 11:24:09
12          The Hindu - Chennai  2012-06-20 11:24:09
13          The Hindu - Chennai  2012-06-20 11:24:09
14          The Hindu - Chennai  2012-06-20 11:24:09
15          The Hindu - Chennai  2012-06-20 11:24:09
16          The Hindu - Chennai  2012-06-20 11:24:09
17          The Hindu - Chennai  2012-06-20 11:24:09
18          The Hindu - Chennai  2012-06-20 11:24:09
19          The Hindu - Chennai  2012-06-20 11:24:09
20          The Hindu - Chennai  2012-06-20 11:24:09
21          The Hindu - Chennai  2012-06-20 11:24:09
22          The Hindu - Chennai  2012-06-20 11:24:09
23          The Hindu - Chennai  2012-06-20 11:24:09
24          The Hindu - Chennai  2012-06-20 11:24:09
25                    TweetDeck            613357772
26          The Hindu - Chennai  2012-06-20 11:24:09
27          The Hindu - Chennai  2012-06-20 11:24:09
28          The Hindu - Chennai  2012-06-20 11:24:09
```

```
29          The Hindu - Chennai  2012-06-20 11:24:09
30          The Hindu - Chennai  2012-06-20 11:24:09
...                         ...                  ...
11108  Traffic Police Prayagraj  2015-07-03 09:06:39
11109  Traffic Police Prayagraj  2015-07-03 09:06:39
11110  Traffic Police Prayagraj  2015-07-03 09:06:39
11111  Traffic Police Prayagraj  2015-07-03 09:06:39
11112  Traffic Police Prayagraj  2015-07-03 09:06:39
11113  Traffic Police Prayagraj  2015-07-03 09:06:39
11114  Traffic Police Prayagraj  2015-07-03 09:06:39
11115  Traffic Police Prayagraj  2015-07-03 09:06:39
11116  Traffic Police Prayagraj  2015-07-03 09:06:39
11117  Traffic Police Prayagraj  2015-07-03 09:06:39
11118  Traffic Police Prayagraj  2015-07-03 09:06:39
11119  Traffic Police Prayagraj  2015-07-03 09:06:39
11120  Traffic Police Prayagraj  2015-07-03 09:06:39
11121  Traffic Police Prayagraj  2015-07-03 09:06:39
11122  Traffic Police Prayagraj  2015-07-03 09:06:39
11123  Traffic Police Prayagraj  2015-07-03 09:06:39
11124  Traffic Police Prayagraj  2015-07-03 09:06:39
11125  Traffic Police Prayagraj  2015-07-03 09:06:39
11126  Traffic Police Prayagraj  2015-07-03 09:06:39
11127  Traffic Police Prayagraj  2015-07-03 09:06:39
11128  Traffic Police Prayagraj  2015-07-03 09:06:39
11129  Traffic Police Prayagraj  2015-07-03 09:06:39
11130  Traffic Police Prayagraj  2015-07-03 09:06:39
11131  Traffic Police Prayagraj  2015-07-03 09:06:39
11132  Traffic Police Prayagraj  2015-07-03 09:06:39
11133  Traffic Police Prayagraj  2015-07-03 09:06:39
11134  Traffic Police Prayagraj  2015-07-03 09:06:39
11135  Traffic Police Prayagraj  2015-07-03 09:06:39
11136  Traffic Police Prayagraj  2015-07-03 09:06:39
11137  Traffic Police Prayagraj  2015-07-03 09:06:39

                                         UserDescription  \
1      The official twitter account of The Hindu's re...
2      The official twitter account of The Hindu's re...
3      The official twitter account of The Hindu's re...
4      The official twitter account of The Hindu's re...
5      The official twitter account of The Hindu's re...
6      The official twitter account of The Hindu's re...
7      The official twitter account of The Hindu's re...
8      The official twitter account of The Hindu's re...
9      The official twitter account of The Hindu's re...
10     The official twitter account of The Hindu's re...
11     The official twitter account of The Hindu's re...
12     The official twitter account of The Hindu's re...
13     The official twitter account of The Hindu's re...
```

```
14     The official twitter account of The Hindu's re...
15     The official twitter account of The Hindu's re...
16     The official twitter account of The Hindu's re...
17     The official twitter account of The Hindu's re...
18     The official twitter account of The Hindu's re...
19     The official twitter account of The Hindu's re...
20     The official twitter account of The Hindu's re...
21     The official twitter account of The Hindu's re...
22     The official twitter account of The Hindu's re...
23     The official twitter account of The Hindu's re...
24     The official twitter account of The Hindu's re...
25                                             THChennai
26     The official twitter account of The Hindu's re...
27     The official twitter account of The Hindu's re...
28     The official twitter account of The Hindu's re...
29     The official twitter account of The Hindu's re...
30     The official twitter account of The Hindu's re...
...                                                  ...
11108  Official Twitter account of Allahabad #Traffic...
11109  Official Twitter account of Allahabad #Traffic...
11110  Official Twitter account of Allahabad #Traffic...
11111  Official Twitter account of Allahabad #Traffic...
11112  Official Twitter account of Allahabad #Traffic...
11113  Official Twitter account of Allahabad #Traffic...
11114  Official Twitter account of Allahabad #Traffic...
11115  Official Twitter account of Allahabad #Traffic...
11116  Official Twitter account of Allahabad #Traffic...
11117  Official Twitter account of Allahabad #Traffic...
11118  Official Twitter account of Allahabad #Traffic...
11119  Official Twitter account of Allahabad #Traffic...
11120  Official Twitter account of Allahabad #Traffic...
11121  Official Twitter account of Allahabad #Traffic...
11122  Official Twitter account of Allahabad #Traffic...
11123  Official Twitter account of Allahabad #Traffic...
11124  Official Twitter account of Allahabad #Traffic...
11125  Official Twitter account of Allahabad #Traffic...
11126  Official Twitter account of Allahabad #Traffic...
11127  Official Twitter account of Allahabad #Traffic...
11128  Official Twitter account of Allahabad #Traffic...
11129  Official Twitter account of Allahabad #Traffic...
11130  Official Twitter account of Allahabad #Traffic...
11131  Official Twitter account of Allahabad #Traffic...
11132  Official Twitter account of Allahabad #Traffic...
11133  Official Twitter account of Allahabad #Traffic...
11134  Official Twitter account of Allahabad #Traffic...
11135  Official Twitter account of Allahabad #Traffic...
11136  Official Twitter account of Allahabad #Traffic...
11137  Official Twitter account of Allahabad #Traffic...
```

```
      UserDescriptionLength   UserFollowersCount  \
1                       145                62144
2                       145                62144
3                       145                62144
4                       145                62144
5                       145                62144
6                       145                62144
7                       145                62144
8                       145                62144
9                       145                62144
10                      145                62144
11                      145                62144
12                      145                62144
13                      145                62144
14                      145                62144
15                      145                62144
16                      145                62144
17                      145                62144
18                      145                62144
19                      145                62144
20                      145                62144
21                      145                62144
22                      145                62144
23                      145                62144
24                      145                62144
25      The Hindu - Chennai  2012-06-20 11:24:09
26                      145                62144
27                      145                62144
28                      145                62144
29                      145                62144
30                      145                62144
...                     ...                  ...
11108                   138                 7608
11109                   138                 7608
11110                   138                 7608
11111                   138                 7608
11112                   138                 7608
11113                   138                 7608
11114                   138                 7608
11115                   138                 7608
11116                   138                 7608
11117                   138                 7608
11118                   138                 7608
11119                   138                 7608
11120                   138                 7608
11121                   138                 7608
11122                   138                 7608
```

```
11123                   138                 7608
11124                   138                 7608
11125                   138                 7608
11126                   138                 7608
11127                   138                 7608
11128                   138                 7608
11129                   138                 7608
11130                   138                 7608
11131                   138                 7608
11132                   138                 7608
11133                   138                 7608
11134                   138                 7608
11135                   138                 7608
11136                   138                 7608
11137                   138                 7608

                                        UserFriendsCount      UserLocation  \
1                                                    297    Chennai, India
2                                                    297    Chennai, India
3                                                    297    Chennai, India
4                                                    297    Chennai, India
5                                                    297    Chennai, India
6                                                    297    Chennai, India
7                                                    297    Chennai, India
8                                                    297    Chennai, India
9                                                    297    Chennai, India
10                                                   297    Chennai, India
11                                                   297    Chennai, India
12                                                   297    Chennai, India
13                                                   297    Chennai, India
14                                                   297    Chennai, India
15                                                   297    Chennai, India
16                                                   297    Chennai, India
17                                                   297    Chennai, India
18                                                   297    Chennai, India
19                                                   297    Chennai, India
20                                                   297    Chennai, India
21                                                   297    Chennai, India
22                                                   297    Chennai, India
23                                                   297    Chennai, India
24                                                   297    Chennai, India
25     The official twitter account of The Hindu's re...               145
26                                                   297    Chennai, India
27                                                   297    Chennai, India
28                                                   297    Chennai, India
29                                                   297    Chennai, India
30                                                   297    Chennai, India
...                                                  ...               ...
```

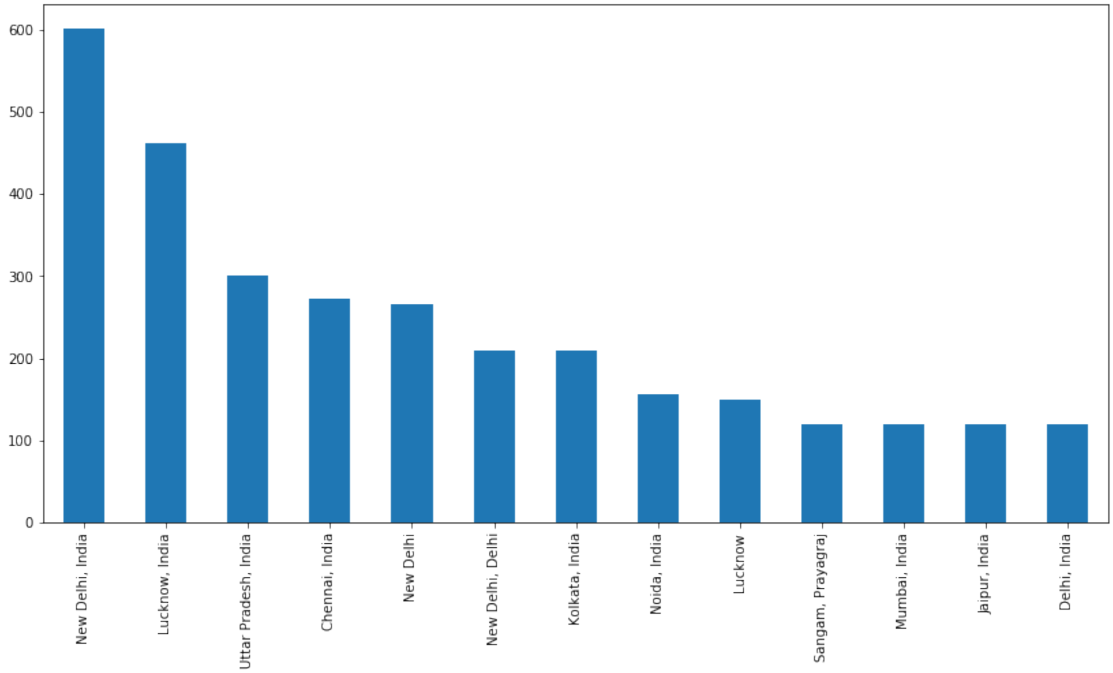
```
11108                                                146  Allahabad, India
11109                                                146  Allahabad, India
11110                                                146  Allahabad, India
11111                                                146  Allahabad, India
11112                                                146  Allahabad, India
11113                                                146  Allahabad, India
11114                                                146  Allahabad, India
11115                                                146  Allahabad, India
11116                                                146  Allahabad, India
11117                                                146  Allahabad, India
11118                                                146  Allahabad, India
11119                                                146  Allahabad, India
11120                                                146  Allahabad, India
11121                                                146  Allahabad, India
11122                                                146  Allahabad, India
11123                                                146  Allahabad, India
11124                                                146  Allahabad, India
11125                                                146  Allahabad, India
11126                                                146  Allahabad, India
11127                                                146  Allahabad, India
11128                                                146  Allahabad, India
11129                                                146  Allahabad, India
11130                                                146  Allahabad, India
11131                                                146  Allahabad, India
11132                                                146  Allahabad, India
11133                                                146  Allahabad, India
11134                                                146  Allahabad, India
11135                                                146  Allahabad, India
11136                                                146  Allahabad, India
11137                                                146  Allahabad, India

      HttpCount HashtagCount    MentionCount  TweetCount
1             0            0               2       21157
2             0            1               1       21157
3             0            2               2       21157
4             0            0               1       21157
5             1            0               1       21157
6             1            0               0       21157
7             0            0               0       21157
8             0            0               1       21157
9             1            0               2       21157
10            1            0               0       21157
11            1            0               0       21157
12            0            0               1       21157
13            0            0               1       21157
14            0            0               1       21157
15            1            1               1       21157
16            0            2               2       21157
```

```
17            0            2               2       21157
18            1            0               2       21157
19            0            2               2       21157
20            0            1               1       21157
21            0            1               2       21157
22            0            0               3       21157
23            0            0               1       21157
24            0            0               1       21157
25        62144          297  Chennai, India           0
26            1            1               0       21157
27            0            0               1       21157
28            0            0               1       21157
29            0            0               1       21157
30            0            1               1       21157
...         ...          ...             ...         ...
11108         1            1               1        4937
11109         1            1               1        4937
11110         1            2               1        4937
11111         1            0               5        4937
11112         1            2               3        4937
11113         0            0               4        4937
11114         1            0               3        4937
11115         0            0               1        4937
11116         1            2               2        4937
11117         0            0               1        4937
11118         1            0               0        4937
11119         1            0               5        4937
11120         1            0               0        4937
11121         1            2               0        4937
11122         1            2               0        4937
11123         1            2               1        4937
11124         0            2               1        4937
11125         0            0               1        4937
11126         0            0               1        4937
11127         1            0               6        4937
11128         1            2               1        4937
11129         1            2               1        4937
11130         1            0               5        4937
11131         0            0               1        4937
11132         1            2               0        4937
11133         1            0               2        4937
11134         0            0               4        4937
11135         0            0               1        4937
11136         1            0               5        4937
11137         1            0               2        4937

[11137 rows x 19 columns]
```

```
In [214]: #drow bar plot to see tweet come from the locations
```

```
In [36]: location_data = Total_leg_data['UserLocation'].value_counts()
         location_data[2:15].plot(kind='bar', figsize=(14,7))
```

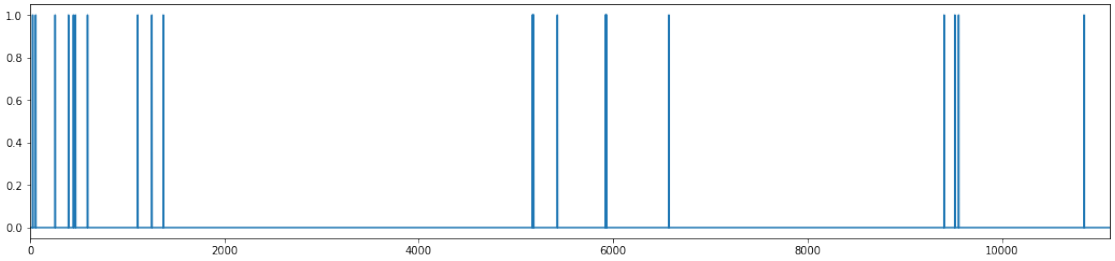
```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe60993df60>
```



```
In [216]: # draw pie chart for a word how many times it used in tweets
          # Hypothesis is Legitimate users user very less compare to spammer
```

```
In [37]: 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()
```

```
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe6097b8a90>
```



```
In [38]: Total_leg_data=Total_leg_data.fillna(0)
         Total_leg_data.shape
```

```
Out[38]: (11118, 21)
```

```
In [218]: # Save Followers count
```

```
In [39]: temp1 = Total_leg_data[["UserFollowersCount"]]
         temp1.to_csv('temp1.csv', sep=',',encoding='utf8')
```

```
In [243]: #Retweet ratio also will be higher compare to spammer user
```

```
In [40]: Total_leg_data[['RetweetCount']] = Total_leg_data[['RetweetCount']].astype(float)
         Total_leg_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11118 entries, 0 to 11117
Data columns (total 21 columns):
Unnamed: 0                11118 non-null int64
Unnamed: 0.1              11118 non-null int64
TwittID                   11118 non-null int64
TextData                  11118 non-null object
TweetCreatedAt            11118 non-null object
RetweetCount              11118 non-null float64
TweetFavouriteCount       11118 non-null int64
TweetSource               11118 non-null object
UserID                    11118 non-null int64
UserScreenName            11118 non-null object
UserName                  11118 non-null object
UserCreatedAt             11118 non-null object
UserDescription           11118 non-null object
UserDescriptionLength     11118 non-null int64
UserFollowersCount        11118 non-null int64
UserFriendsCount          11118 non-null int64
UserLocation              11118 non-null object
HttpCount                 11118 non-null int64
HashtagCount              11118 non-null int64
MentionCount              11118 non-null int64
TweetCount                11118 non-null int64
dtypes: float64(1), int64(12), object(8)
memory usage: 1.8+ MB
```

```
In [269]: Total_leg_data.drop("Unnamed: 24", axis=1, inplace=True)
```

```
In [41]: # to see how many people have zero tweet
         Total_leg_data = Total_leg_data[Total_leg_data.TweetCount!=0]
         len(Total_leg_data[Total_leg_data.TweetCount<30])
```

```
Out[41]: 378
```

```
In [42]: Total_leg_data[["RetweetCount"]] = Total_leg_data[["RetweetCount"]].astype(float)
         Total_leg_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11118 entries, 0 to 11117
Data columns (total 21 columns):
Unnamed: 0                11118 non-null int64
Unnamed: 0.1              11118 non-null int64
TwittID                   11118 non-null int64
TextData                  11118 non-null object
TweetCreatedAt            11118 non-null object
RetweetCount              11118 non-null float64
TweetFavouriteCount       11118 non-null int64
TweetSource               11118 non-null object
UserID                    11118 non-null int64
UserScreenName            11118 non-null object
UserName                  11118 non-null object
UserCreatedAt             11118 non-null object
UserDescription           11118 non-null object
UserDescriptionLength     11118 non-null int64
UserFollowersCount        11118 non-null int64
UserFriendsCount          11118 non-null int64
UserLocation              11118 non-null object
HttpCount                 11118 non-null int64
HashtagCount              11118 non-null int64
MentionCount              11118 non-null int64
TweetCount                11118 non-null int64
dtypes: float64(1), int64(12), object(8)
memory usage: 1.9+ MB
```

```
In [43]: Total_leg_data.loc[:,"AvgHashtag"] = (Total_leg_data.groupby('UserID')["HashtagCount"]
         Total_leg_data.loc[:,"AvgURLCount"] = (Total_leg_data.groupby('UserID')["HttpCount"].
         Total_leg_data.loc[:,"AvgMention"] = (Total_leg_data.groupby('UserID')["MentionCount"]
         Total_leg_data.loc[:,"AvgRetweet"] = (Total_leg_data.groupby('UserID')["RetweetCount"]
         Total_leg_data.loc[:,"AvgFavCount"] = (Total_leg_data.groupby('UserID')["TweetFavouri
```

```
In [44]: # Selecting Repeted columns only and droping the repeted rows

         unique_leg_row = Total_leg_data[["UserID", "UserScreenName", "UserCreatedAt", "UserDe
         leg_data = unique_leg_row.drop_duplicates()
         leg_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 375 entries, 0 to 11088
Data columns (total 13 columns):
UserID                    375 non-null int64
```

```
UserScreenName          375 non-null object
UserCreatedAt           375 non-null object
UserDescriptionLength   375 non-null int64
UserFollowersCount      375 non-null int64
UserFriendsCount        375 non-null int64
UserLocation            375 non-null object
AvgHashtag              375 non-null float64
AvgURLCount             375 non-null float64
AvgMention              375 non-null float64
AvgRetweet              375 non-null float64
AvgFavCount             375 non-null float64
TweetCount              375 non-null int64
dtypes: float64(5), int64(5), object(3)
memory usage: 41.0+ KB
```

```
In [45]: # Saving the reduced legitimate data
         fre = leg_data["UserFriendsCount"]
         fre.to_csv("Temp_leg.csv", sep=',',encoding='utf8')
```

```
/home/radhey/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: FutureWarning: The
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
In [46]: # Datatype conversion from object to float
         leg_data[['UserFriendsCount']] = leg_data[['UserFriendsCount']].astype(float)
         leg_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 375 entries, 0 to 11088
Data columns (total 13 columns):
UserID                  375 non-null int64
UserScreenName          375 non-null object
UserCreatedAt           375 non-null object
UserDescriptionLength   375 non-null int64
UserFollowersCount      375 non-null int64
UserFriendsCount        375 non-null float64
UserLocation            375 non-null object
AvgHashtag              375 non-null float64
AvgURLCount             375 non-null float64
AvgMention              375 non-null float64
AvgRetweet              375 non-null float64
AvgFavCount             375 non-null float64
TweetCount              375 non-null int64
dtypes: float64(6), int64(4), object(3)
memory usage: 41.0+ KB
```

```
/home/radhey/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py:3391: SettingWithCopyWa
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  self[k1] = value[k2]
```

```
In [47]: # Add a Column to LEgitimate Data that this is not Spam =0
         leg_data.loc[:, "SpammerOrNot"]=0
         leg_data.tail()
```

```
/home/radhey/anaconda3/lib/python3.6/site-packages/pandas/core/indexing.py:362: SettingWithCopy
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  self.obj[key] = _infer_fill_value(value)
/home/radhey/anaconda3/lib/python3.6/site-packages/pandas/core/indexing.py:543: SettingWithCopy
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  self.obj[item] = s
```

```
Out[47]:                    UserID   UserScreenName        UserCreatedAt  \
         10968  767677235805511680      mediaamantra  2016-08-22 10:58:10
         10998            67378160      DainikBhaskar  2009-08-20 18:04:36
         11028           461841349       ZeeNewsHindi  2012-01-12 07:52:31
         11058            98362607        News18India  2009-12-21 12:11:21
         11088          3266889528  allahabdtraffic  2015-07-03 09:06:39

                UserDescriptionLength  UserFollowersCount  UserFriendsCount  \
         10968                    128                1566             958.0
         10998                     76              634524              46.0
         11028                    110             1868923              22.0
         11058                     47             1035839              89.0
         11088                    138                7608             146.0

                    UserLocation  AvgHashtag  AvgURLCount  AvgMention  AvgRetweet  \
         10968    Lucknow, India    0.400000          1.0    0.600000    3.400000
         10998             India    8.933333          4.0    1.333333   12.533333
         11028             India    2.033333          2.9    1.500000  244.966667
         11058             India    2.033333          2.0    1.566667   20.733333
         11088  Allahabad, India    0.733333          0.7    1.966667   35.100000

                AvgFavCount  TweetCount  SpammerOrNot
         10968     4.533333       14896             0
         10998    87.900000      119712             0
```

```
11028   992.166667      181029               0
11058   100.533333      285844               0
11088     2.633333        4937               0
```

```
In [48]: leg_data["TweetCount"].describe()
```

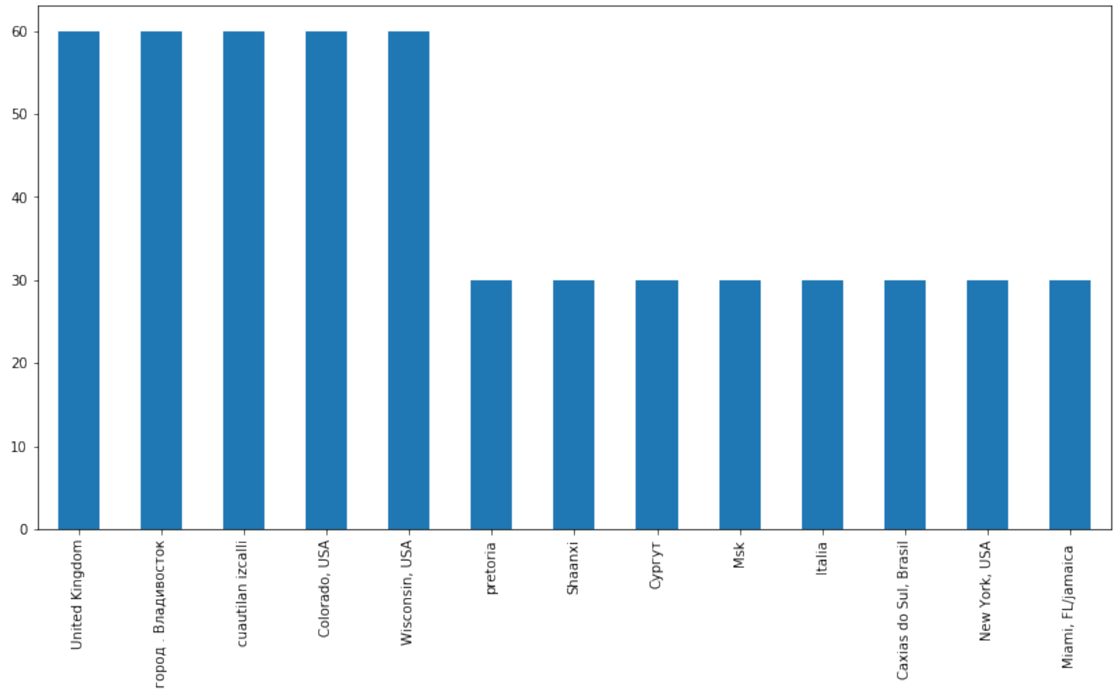
```
Out[48]: count       375.000000
         mean      41288.162667
         std       93281.144477
         min           1.000000
         25%         324.000000
         50%        3883.000000
         75%       20650.000000
         max      596778.000000
         Name: TweetCount, dtype: float64
```

```
In [50]: # Now Loading Spammer Data
         Total_spam_data = pd.read_csv("Spam_data.csv")
         Total_spam_data.fillna(0, inplace=True)
         Total_spam_data.shape
```

```
Out[50]: (5394, 21)
```

```
In [51]: %matplotlib inline
         location_data = Total_spam_data['UserLocation'].value_counts()
         location_data[2:15].plot(kind='bar', figsize=(14,7))
```

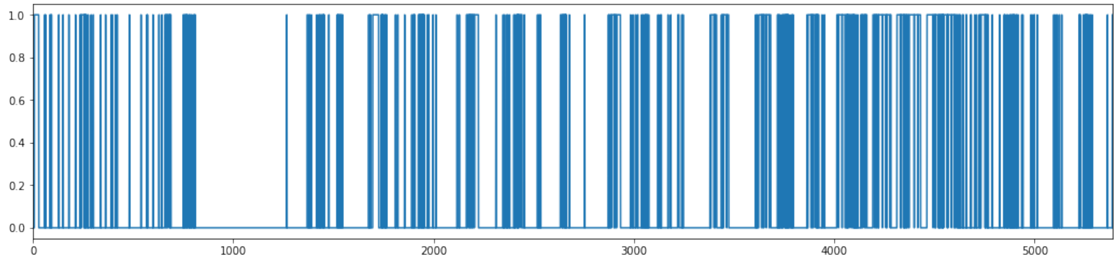
```
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe6089f1198>
```



```
In [ ]: #By Analyize Tweet I find that there is a lot of volgor word used by spam user compare
```

```
In [52]: import matplotlib.pyplot as plt
         import string as str
         %matplotlib inline
         plt.rcParams['figure.figsize'] = (18,4)
         plt.rcParams['font.family'] = 'sans-serif'
         text = Total_spam_data['TextData']
         is_sex = text.str.contains('sex')
         is_sex=is_sex.astype(float)
         is_sex.plot()
```

```
Out[52]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe6089bb550>
```



```
In [53]: Total_spam_data=Total_spam_data.fillna(0)
         Total_spam_data.shape
```

```
Out[53]: (5394, 21)
```

```
In [54]: temp2 = Total_spam_data[["UserFollowersCount"]]
         temp2.to_csv('temp2.csv', sep=',',encoding='utf8')
```

```
In [55]: Total_spam_data[['RetweetCount']] = Total_spam_data[['RetweetCount']].astype(float)
         Total_spam_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5394 entries, 0 to 5393
Data columns (total 21 columns):
Unnamed: 0                5394 non-null int64
Unnamed: 0.1              5394 non-null int64
TwittID                   5394 non-null int64
TextData                  5394 non-null object
TweetCreatedAt            5394 non-null object
RetweetCount              5394 non-null float64
TweetFavouriteCount       5394 non-null int64
```

```
TweetSource               5394 non-null object
UserID                    5394 non-null int64
UserScreenName            5394 non-null object
UserName                  5394 non-null object
UserCreatedAt             5394 non-null object
UserDescription           5394 non-null object
UserDescriptionLength     5394 non-null int64
UserFollowersCount        5394 non-null int64
UserFriendsCount          5394 non-null int64
UserLocation              5394 non-null object
HttpCount                 5394 non-null int64
HashtagCount              5394 non-null int64
MentionCount              5394 non-null int64
TweetCount                5394 non-null int64
dtypes: float64(1), int64(12), object(8)
memory usage: 885.0+ KB
```

```
In [56]: Total_spam_data = Total_spam_data[Total_spam_data.TweetCount!=0]
         len(Total_spam_data[Total_spam_data.TweetCount<30])

Out[56]: 54

In [57]: Total_spam_data.loc[:,'AvgHashtag'] = (Total_spam_data.groupby('UserID')["HashtagCoun
         Total_spam_data.loc[:,'AvgURLCount'] = (Total_spam_data.groupby('UserID')["HttpCount"]
         Total_spam_data.loc[:,'AvgMention'] = (Total_spam_data.groupby('UserID')["MentionCoun
         Total_spam_data.loc[:,'AvgRetweet'] = (Total_spam_data.groupby('UserID')["RetweetCoun
         Total_spam_data.loc[:,'AvgFavCount'] = (Total_spam_data.groupby('UserID')["TweetFavou

In [58]: Total_spam_data.tail(4)

Out[58]:       Unnamed: 0  Unnamed: 0.1              TwittID  \
         5390        5390           746  1120300621578551296
         5391        5391           747  1120300607309524992
         5392        5392           748  1120300592046444545
         5393        5393           749  1120300537314979840


                                                    TextData       TweetCreatedAt  \
         5390  RT @s___fire: your sex life is going bad ? you...  2019-04-22 12:17:37
         5391  RT @s___fire: Find your fantasy here and make ...  2019-04-22 12:17:33
         5392  RT @sexole: ONLINE EN https://t.co/wkT9BMovtL ...  2019-04-22 12:17:29
         5393  RT @DomUrch: @irinagomez60\n@HQPornHQ\n@Erotik...  2019-04-22 12:17:16

               RetweetCount  TweetFavouriteCount          TweetSource      UserID  \
         5390          22.0                    0  Twitter for Android  1055696622
         5391          18.0                    0  Twitter for Android  1055696622
         5392           1.0                    0  Twitter for Android  1055696622
         5393         121.0                    0  Twitter for Android  1055696622
```

```
     UserScreenName  ... UserLocation HttpCount HashtagCount  MentionCount  \
5390    Giovannini8  ...            0         1            0             1
5391    Giovannini8  ...            0         1            0             1
5392    Giovannini8  ...            0         2            2             1
5393    Giovannini8  ...            0         0            0            11

      TweetCount  AvgHashtag AvgURLCount  AvgMention  AvgRetweet  AvgFavCount
5390      150737    0.833333    1.766667         3.6  138.733333          0.0
5391      150737    0.833333    1.766667         3.6  138.733333          0.0
5392      150737    0.833333    1.766667         3.6  138.733333          0.0
5393      150737    0.833333    1.766667         3.6  138.733333          0.0

[4 rows x 26 columns]
```

```
In [59]: # Selecting Repeted columns only and droping the repeted rows

         unique_spam_row = Total_spam_data[["UserID", "UserScreenName", "UserCreatedAt", "User
         spam_data = unique_spam_row.drop_duplicates()
         spam_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 177 entries, 0 to 5364
Data columns (total 13 columns):
UserID                   177 non-null int64
UserScreenName           177 non-null object
UserCreatedAt            177 non-null object
UserDescriptionLength    177 non-null int64
UserFollowersCount       177 non-null int64
UserFriendsCount         177 non-null int64
UserLocation             177 non-null object
AvgHashtag               177 non-null float64
AvgURLCount              177 non-null float64
AvgMention               177 non-null float64
AvgRetweet               177 non-null float64
AvgFavCount              177 non-null float64
TweetCount               177 non-null int64
dtypes: float64(5), int64(5), object(3)
memory usage: 19.4+ KB
```

```
In [60]: # Saving the reduced Spammer data
         fre = spam_data["UserFriendsCount"]
         fre.to_csv("Temp_spam.csv", sep=',',encoding='utf8')
```

```
/home/radhey/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: FutureWarning: The
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
In [61]: # Datatype conversion from object to float
         spam_data[['UserFriendsCount']] = spam_data[['UserFriendsCount']].astype(float)
         spam_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 177 entries, 0 to 5364
Data columns (total 13 columns):
UserID                   177 non-null int64
UserScreenName           177 non-null object
UserCreatedAt            177 non-null object
UserDescriptionLength    177 non-null int64
UserFollowersCount       177 non-null int64
UserFriendsCount         177 non-null float64
UserLocation             177 non-null object
AvgHashtag               177 non-null float64
AvgURLCount              177 non-null float64
AvgMention               177 non-null float64
AvgRetweet               177 non-null float64
AvgFavCount              177 non-null float64
TweetCount               177 non-null int64
dtypes: float64(6), int64(4), object(3)
memory usage: 19.4+ KB
```

```
In [62]: # Add a Column to LEgitimate Data that this is not Spam =0
         spam_data.loc[:, "SpammerOrNot"]=1
         spam_data.tail()
```

```
Out[62]:                    UserID   UserScreenName        UserCreatedAt  \
         5214   956015377888305152  jcroldanroldan1  2018-01-24 04:06:42
         5244  1103478268919980035    Sariw56676073  2019-03-07 02:11:35
         5274  1036466998446710786   Cris9666450351  2018-09-03 04:12:43
         5304            125706019      Grinder0420  2010-03-23 16:13:23
         5364           1055696622      Giovannini8  2013-01-02 17:56:31

               UserDescriptionLength  UserFollowersCount  UserFriendsCount  \
         5214                      0                 393            3734.0
         5244                      0                  12               0.0
         5274                      0                  19              44.0
         5304                     96                2305            2587.0
         5364                      0                1755            2130.0

              UserLocation  AvgHashtag  AvgURLCount  AvgMention  AvgRetweet  \
         5214            0    0.766667     1.066667         1.1  158.700000
         5244            0    8.233333     1.033333         0.0    0.000000
         5274            0    0.766667     0.800000         1.4   97.533333
         5304            0    0.100000     0.700000         1.1   22.433333
         5364            0    0.833333     1.766667         3.6  138.733333
```

```
      AvgFavCount  TweetCount  SpammerOrNot
5214     0.000000        9382             1
5244     0.333333         114             1
5274     0.000000        1845             1
5304     0.066667      143508             1
5364     0.000000      150737             1
```

```
In [63]: spam_data["TweetCount"].describe()
```

```
Out[63]: count    1.770000e+02
         mean     2.532717e+04
         std      9.549593e+04
         min      1.000000e+00
         25%      6.410000e+02
         50%      4.744000e+03
         75%      1.185200e+04
         max      1.150378e+06
         Name: TweetCount, dtype: float64
```

```
In [64]: leg_data["TweetCount"].describe()
```

```
Out[64]: count       375.000000
         mean      41288.162667
         std       93281.144477
         min           1.000000
         25%         324.000000
         50%        3883.000000
         75%       20650.000000
         max      596778.000000
         Name: TweetCount, dtype: float64
```

```
In [65]: # Merging the legitimate and spammer data
         import pandas as pd
         frames = [leg_data, spam_data]
         Total_data = pd.concat(frames, axis=0, sort=False)
         Total_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 552 entries, 0 to 5364
Data columns (total 14 columns):
UserID                       552 non-null int64
UserScreenName               552 non-null object
UserCreatedAt                552 non-null object
UserDescriptionLength        552 non-null int64
UserFollowersCount           552 non-null int64
UserFriendsCount             552 non-null float64
UserLocation                 552 non-null object
AvgHashtag                   552 non-null float64
```

```
AvgURLCount              552 non-null float64
AvgMention               552 non-null float64
AvgRetweet               552 non-null float64
AvgFavCount              552 non-null float64
TweetCount               552 non-null int64
SpammerOrNot             552 non-null int64
dtypes: float64(6), int64(5), object(3)
memory usage: 64.7+ KB
```

```
In [66]: Total_data.reset_index()
         Total_data.to_csv('Total_data.csv', sep=',', encoding='utf8')
```

```
In [67]: #.........................Section Third.....................
         # loading total Data
         # from here machine learning will start
         import pandas as pd
         import datetime
         Total_data = pd.read_csv('Total_data.csv')
         Total_data.fillna(0, inplace=True)
         Current_Time = datetime.datetime.strftime(datetime.datetime.now(), '%Y-%m-%d %H:%M:%S
         Total_data.loc[:, "Current_Time"]=Current_Time
         Total_data.to_csv('Total_data.csv', sep=',', encoding='utf8')
         Total_data = pd.read_csv('Total_data.csv')
         Total_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 552 entries, 0 to 551
Data columns (total 17 columns):
Unnamed: 0               552 non-null int64
Unnamed: 0.1             552 non-null int64
UserID                   552 non-null int64
UserScreenName           552 non-null object
UserCreatedAt            552 non-null object
UserDescriptionLength    552 non-null int64
UserFollowersCount       552 non-null int64
UserFriendsCount         552 non-null float64
UserLocation             552 non-null object
AvgHashtag               552 non-null float64
AvgURLCount              552 non-null float64
AvgMention               552 non-null float64
AvgRetweet               552 non-null float64
AvgFavCount              552 non-null float64
TweetCount               552 non-null int64
SpammerOrNot             552 non-null int64
Current_Time             552 non-null object
dtypes: float64(6), int64(7), object(4)
memory usage: 73.4+ KB
```

```
In [68]: #debugging purpose if some data type do not appear as the should be
         temp1=Total_data[["UserCreatedAt"]]
         Total_data.tail(3)
```

```
Out[68]:      Unnamed: 0  Unnamed: 0.1               UserID  UserScreenName  \
         549         549          5274  1036466998446710786  Cris9666450351
         550         550          5304            125706019     Grinder0420
         551         551          5364           1055696622     Giovannini8

                    UserCreatedAt  UserDescriptionLength  UserFollowersCount  \
         549  2018-09-03 04:12:43                      0                  19
         550  2010-03-23 16:13:23                     96                2305
         551  2013-01-02 17:56:31                      0                1755

              UserFriendsCount UserLocation  AvgHashtag  AvgURLCount  AvgMention  \
         549              44.0            0    0.766667     0.800000         1.4
         550            2587.0            0    0.100000     0.700000         1.1
         551            2130.0            0    0.833333     1.766667         3.6

              AvgRetweet  AvgFavCount  TweetCount  SpammerOrNot         Current_Time
         549   97.533333     0.000000        1845             1  2019-04-23 01:00:09
         550   22.433333     0.066667      143508             1  2019-04-23 01:00:09
         551  138.733333     0.000000      150737             1  2019-04-23 01:00:09
```

```
In [69]: # converting string to float
         Total_data["UserFriendsCount"] = Total_data["UserFriendsCount"].convert_objects(conve
```

```
/home/radhey/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: FutureWarning: conv
For all other conversions use the data-type specific converters pd.to_datetime, pd.to_timedelta
```

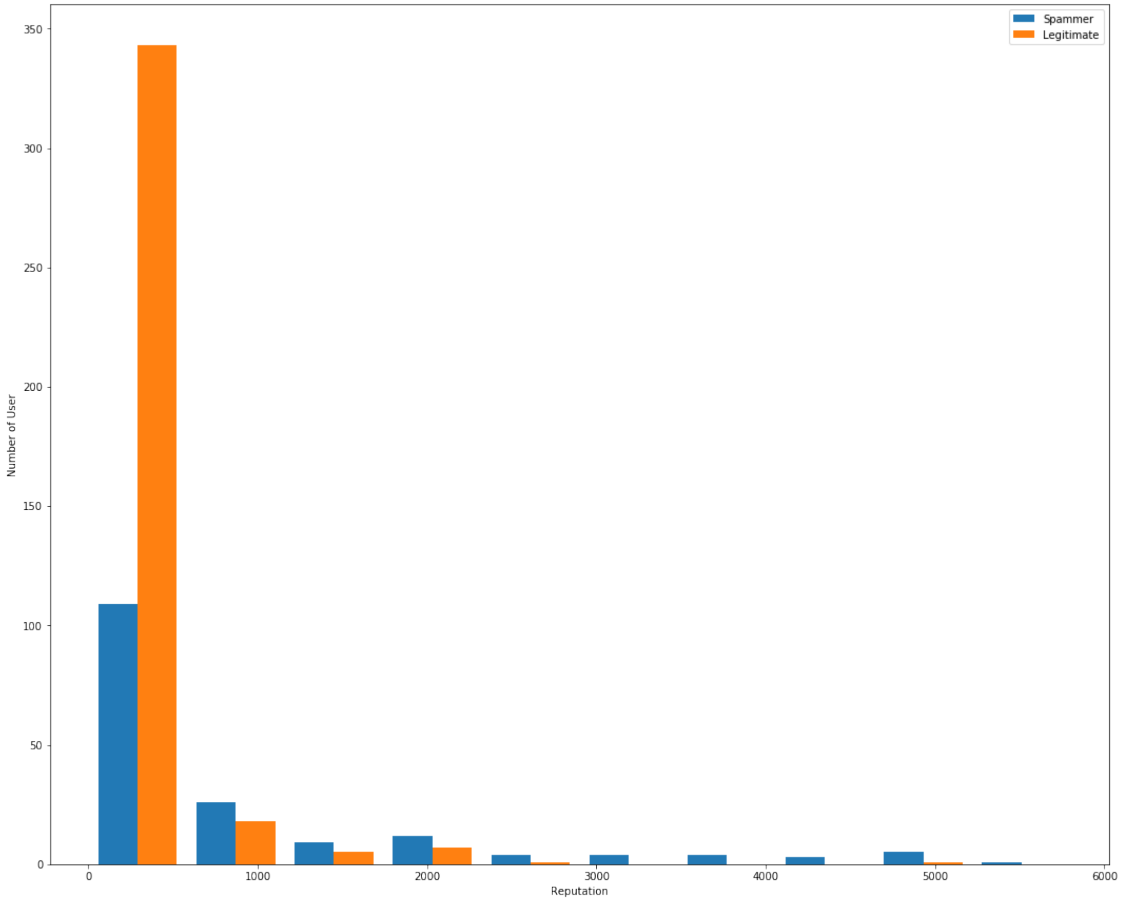
```
In [70]: Total_data["UserFriendsCount"].describe()
```

```
Out[70]: count     552.000000
         mean      436.949275
         std       875.788595
         min         0.000000
         25%        26.750000
         50%       106.500000
         75%       366.500000
         max      5799.000000
         Name: UserFriendsCount, dtype: float64
```

```
In [71]: #Adding Reputaion features
         Total_data.loc[:,"Reputation"]=Total_data["UserFollowersCount"]/(Total_data["UserFoll
         Total_data["Reputation"].describe()
         Total_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 552 entries, 0 to 551
Data columns (total 18 columns):
Unnamed: 0                552 non-null int64
Unnamed: 0.1              552 non-null int64
UserID                    552 non-null int64
UserScreenName            552 non-null object
UserCreatedAt             552 non-null object
UserDescriptionLength     552 non-null int64
UserFollowersCount        552 non-null int64
UserFriendsCount          552 non-null float64
UserLocation              552 non-null object
AvgHashtag                552 non-null float64
AvgURLCount               552 non-null float64
AvgMention                552 non-null float64
AvgRetweet                552 non-null float64
AvgFavCount               552 non-null float64
TweetCount                552 non-null int64
SpammerOrNot              552 non-null int64
Current_Time              552 non-null object
Reputation                552 non-null float64
dtypes: float64(7), int64(7), object(4)
memory usage: 77.7+ KB
```

```
In [74]: import pandas as pd
         import time
         import matplotlib.pyplot as plt
         %matplotlib inline
         plt.rcParams['figure.figsize']=(18,15)
         plt.rcParams['font.family']='sans-serif'

         data0 = Total_data[Total_data.Reputation > .1]
         plt.hist([data0[data0.SpammerOrNot==1].Reputation.values,
                  data0[data0.SpammerOrNot==0].Reputation.values],label=["Spammer", "Legitimat
         plt.legend()
         plt.xlabel("Reputation")
         plt.ylabel("Number of User")
         # to save fig
         plt.savefig('repuation.png')
```



```
In [75]: #1. Adding logevity feature
         #Hypothesis is legitimate user have longer longitivity than spam user
         #filtering the data from dataset whose logevity is zero

In [76]: data = Total_data
         data["Current_Time"] = pd.to_datetime(data["Current_Time"])
         data["UserCreatedAt"] = pd.to_datetime(data["UserCreatedAt"])
         data['AgeOfAccount'] = (data['Current_Time'] - data['UserCreatedAt'])/np.timedelta64(
         cols = ['AgeOfAccount']
         data[cols] = data[cols].mask(data[cols]<0)
         data.AgeOfAccount.describe()
         #data["AgeOfAccount"]=((data["Current_Time"] - data["UserCreatedAt"]).astype('timedel
         #data.AgeOfAccount.describe()

Out[76]: count     552.000000
         mean     1477.410593
         std      1085.302359
         min         1.253218
         25%       518.489207
         50%      1202.138142
         75%      2207.585249
         max      3895.484734
         Name: AgeOfAccount, dtype: float64
```

```
In [77]: #2. Adding tweet per day feature
         data1 = data
         data1.loc[:, "TweetPerDay"] = data1["TweetCount"]/data1["AgeOfAccount"]
         data1["TweetPerDay"].describe()
```

```
Out[77]: count    552.000000
         mean      18.617462
         std       42.748508
         min        0.001138
         25%        0.823242
         50%        3.344650
         75%       15.522447
         max      484.918012
         Name: TweetPerDay, dtype: float64
```

```
In [78]: #3 Adding the feature Number of Tweet
         data1.loc[:,"TweetPerFollower"] = data1["TweetCount"]/data1["UserFollowersCount"]
```

```
In [79]: #4 Dropping the infinte values from pandas for followerCount

         import numpy as np
         #to remove unwanted data
         data1.TweetPerFollower=data1.TweetPerFollower.round(2).fillna(0)
         data1 = data1[np.isfinite(data1['TweetPerFollower'])]
         data1["TweetPerFollower"].tail(3)
```

```
Out[79]: 549    97.11
         550    62.26
         551    85.89
         Name: TweetPerFollower, dtype: float64
```

```
In [80]: # Adding the feature Age of Account/Number of Following
         #Hypothesis is that it is very low for spammer and very high for legitimate user
```

```
In [81]: data1.loc[:,"AgeByFollowing"] = data1["AgeOfAccount"]/data1["UserFriendsCount"]
         data1 = data1[np.isfinite(data1['AgeByFollowing'])]
         data1[['AgeByFollowing']] = data1[['AgeByFollowing']].astype(float)
         data1["AgeByFollowing"].describe()
```

```
Out[81]: count     540.000000
         mean       59.585938
         std       234.072793
         min         0.002277
         25%         2.324010
         50%           8.220824
         75%          35.757579
         max        3002.728368
         Name: AgeByFollowing, dtype: float64
```

```
In [82]: #Separating Spammer and legitimate user
```

```
In [83]: #Spammer_dataframe
         spam_data = data1[data1.SpammerOrNot==1]
         len(spam_data)

Out[83]: 171
```

```
In [84]: #legitimate_dataframe
         leg_data = data1[data1.SpammerOrNot==0]
         len(leg_data)

Out[84]: 369
```

```
In [85]: # Exploring the AgeByFollowing feature
         # for Spammer, Hypothesis is: Age is low and following number is high, so reuslt is v
         # for Legitimate user, Hypothesis is: Age is high and following number is low, so res
```

```
In [86]: leg_data["AgeByFollowing"].describe()

Out[86]: count     369.000000
         mean       57.710487
         std       204.375940
         min         0.095925
         25%         5.633549
         50%        12.972459
         75%        43.172882
         max      3002.728368
         Name: AgeByFollowing, dtype: float64
```

```
In [87]: spam_data["AgeByFollowing"].describe()

Out[87]: count     171.000000
         mean       63.632963
         std       288.572198
         min         0.002277
         25%         0.555243
         50%         1.872940
         75%         7.009605
         max      2909.880324
         Name: AgeByFollowing, dtype: float64
```

```
In [88]: #Selecting the Additional features
In [89]: M = data1[['Reputation', 'AvgHashtag', 'AvgRetweet', 'UserFollowersCount','UserFriends
         y = data1["SpammerOrNot"]
         data1.columns
         M.shape

Out[89]: (540, 13)
```

```
In [90]: # Save these training data
         data1.reset_index()
         data1.to_csv('Total_training_data.csv', sep=',', encoding='utf8')
```

```
In [91]: # Splitting the data
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(M, y, test_size=0.2, random_state=
         print(X_train.shape)
         print(X_test.shape)

(432, 13)
(108, 13)
```

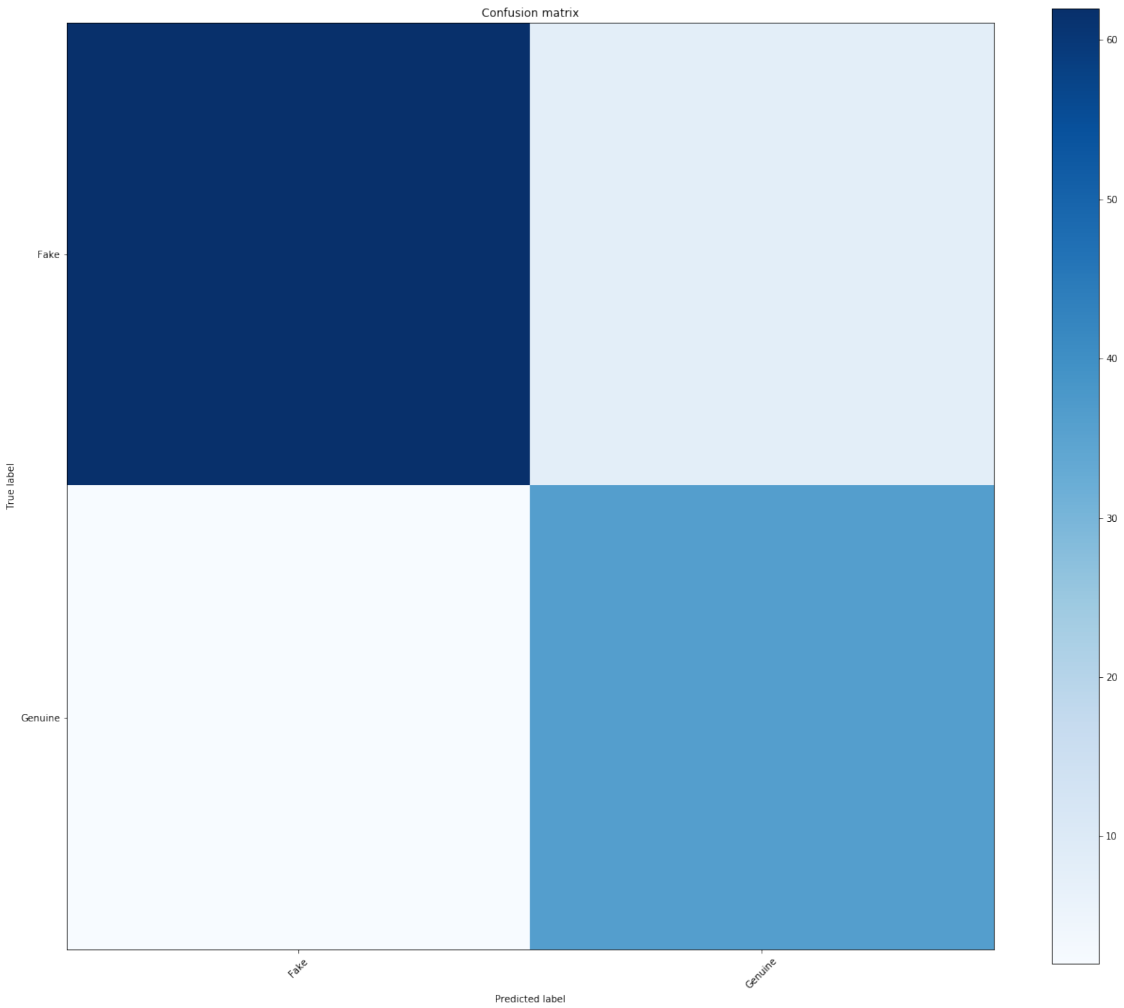
```
In [92]: # Evaluating classifier
```

```
In [101]: # for total X
          from sklearn.metrics import accuracy_score
          from sklearn.metrics import classification_report
          from sklearn.metrics import confusion_matrix
          from sklearn.neighbors import KNeighborsClassifier
          knn = KNeighborsClassifier(n_neighbors=5)
          knn.fit(X_train, y_train)
          y_pred = knn.predict(X_test)
          print(accuracy_score(y_test,y_pred))

          from sklearn.model_selection import cross_val_score
          scores = cross_val_score(knn, M, y, cv=10, scoring='accuracy')
          print("Tenfol cross validation score")
          print(scores)
          print(scores.mean())
          print("\n")
          print("Classifier performance report: ")
          print(classification_report(y_test, y_pred))
          print("Confusion Matrix: ")
          print(confusion_matrix(y_test, y_pred))

0.9074074074074074
Tenfol cross validation score
[0.81818182 0.88888889 0.77777778 0.74074074 0.88888889 0.81481481
 0.96296296 0.96296296 0.96296296 0.90566038]
0.8723842195540309
```

```
Classifier performance report:
              precision    recall  f1-score   support

           0       0.97      0.89      0.93        70
           1       0.82      0.95      0.88        38

   micro avg       0.91      0.91      0.91       108
   macro avg       0.89      0.92      0.90       108
weighted avg       0.92      0.91      0.91       108

Confusion Matrix:
[[62  8]
 [ 2 36]]
```

```
In [100]: def plot_confusion_matrix(cm, title='Confusion matrix', cmap=plt.cm.Blues):
              target_names = ['Fake', 'Genuine']
              plt.imshow(cm, interpolation='nearest', cmap=cmap)
              plt.title(title)
              plt.colorbar()
              tick_marks = np.arange(len(target_names))
              plt.xticks(tick_marks, target_names, rotation=45)
              plt.yticks(tick_marks, target_names)
              plt.tight_layout()
              plt.ylabel('True label')
              plt.xlabel('Predicted label')
              plt.show()
```

```
In [102]: cm = confusion_matrix(y_test, y_pred)
          plot_confusion_matrix(cm)
```



```
In [94]: #support is sum of TP+FN, second FP+TN which gives actual 0(Non_Spammer) and actual 1

In [95]: from sklearn import metrics
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.model_selection import cross_val_score
         est = DecisionTreeClassifier()
         est.fit(X_train, y_train)
         y_pred = est.predict(X_test)
         scores = cross_val_score(knn, M, y, cv=10, scoring='accuracy')
         print(accuracy_score(y_test,y_pred))
         print("Tenfol cross validation score")
         print(scores)
         print(scores.mean())
         print("\n")
         print("Classifier performance report: ")
         print(classification_report(y_test, y_pred))
         print("Confusion Matrix: ")
         print(confusion_matrix(y_test, y_pred))
```

```
0.9351851851851852
Tenfol cross validation score
[0.81818182 0.88888889 0.77777778 0.74074074 0.88888889 0.81481481
 0.96296296 0.96296296 0.96296296 0.90566038]
0.8723842195540309


Classifier performance report:
              precision    recall  f1-score   support

           0       0.97      0.93      0.95        70
           1       0.88      0.95      0.91        38

   micro avg       0.94      0.94      0.94       108
   macro avg       0.92      0.94      0.93       108
weighted avg       0.94      0.94      0.94       108

Confusion Matrix:
[[65  5]
 [ 2 36]]
```

```
In [96]: # attempt to find out most deciding feature
```

```
In [97]: est = DecisionTreeClassifier()
         est.fit(M,y)
         print(est.feature_importances_)
```

```
[0.0244507  0.03046077 0.01392124 0.04205354 0.         0.03272953
 0.054944   0.03541686 0.01806989 0.04341183 0.02763518 0.55280871
 0.12409775]
```

```
In [98]: ## Evaluation of Accuracy of classifier with Naive Bayes G is less accurate than M
```

```
In [99]: from sklearn.naive_bayes import BernoulliNB
         est = BernoulliNB()
         est.fit(X_train, y_train)
         y_pred = est.predict(X_test)
         scores = cross_val_score(knn, M, y, cv=10, scoring='accuracy')
         print(accuracy_score(y_test,y_pred))
         print("Tenfol cross validation score")
         print(scores)
         print(scores.mean())
         print("\n")
         print("Classifier performance report: ")
         print(classification_report(y_test, y_pred))
         print("Confusion Matrix: ")
         print(confusion_matrix(y_test, y_pred))
```

```
0.75
Tenfol cross validation score
[0.81818182 0.88888889 0.77777778 0.74074074 0.88888889 0.81481481
 0.96296296 0.96296296 0.96296296 0.90566038]
0.8723842195540309


Classifier performance report:
              precision    recall  f1-score   support

           0       0.77      0.87      0.82        70
           1       0.69      0.53      0.60        38

   micro avg       0.75      0.75      0.75       108
   macro avg       0.73      0.70      0.71       108
weighted avg       0.74      0.75      0.74       108

Confusion Matrix:
[[61  9]
 [18 20]]
```

```
In [103]: from sklearn.ensemble import RandomForestClassifier
          est = RandomForestClassifier(n_estimators=7, max_depth=7, min_samples_split=5)
          est.fit(X_train, y_train)
          y_pred = est.predict(X_test)
          scores = cross_val_score(knn, M, y, cv=10, scoring='accuracy')
          print(accuracy_score(y_test,y_pred))
          print("Tenfol cross validation score")
          print(scores)
          print(scores.mean())
          print("\n")
          print("Classifier performance report: ")
          print(classification_report(y_test, y_pred))
          print("Confusion Matrix: ")
          print(confusion_matrix(y_test, y_pred))
```

```
0.9537037037037037
Tenfol cross validation score
[0.81818182 0.88888889 0.77777778 0.74074074 0.88888889 0.81481481
 0.96296296 0.96296296 0.96296296 0.90566038]
0.8723842195540309
```

```
Classifier performance report:
              precision    recall  f1-score   support

           0       0.97      0.96      0.96        70
           1       0.92      0.95      0.94        38

   micro avg       0.95      0.95      0.95       108
   macro avg       0.95      0.95      0.95       108
weighted avg       0.95      0.95      0.95       108

Confusion Matrix:
[[67  3]
 [ 2 36]]
```

```
In [104]: # Random Sample Data Collection

In [108]: 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(1,10):
                  twitterStream.filter(track=["car"])
          except KeyboardInterrupt:
              print("Key board interuption")
          with open("stream.txt", "w") as f:
               for item in friends:
                  f.write("%s\n" % item)
          !cat stream.txt
```

```
Dady330
donynyn1
CurrentSocials
haramlaflame
_JamesShu
basiljh
```

```
Raima_Ouattara
arroba_551
samanthaaajae
RioNextDoor
eudoguinha
cxnhoto777
FreebandFlav4
oluwamisegun
CallMeKi__
CLeonard46
ONEeJuice
KatlyGold
mia_sansone
thcxns
slctiio
mariesimspon95
vanitascrimes
Jip8659
insimricky
XiggyMatsu
apk_share
HogardJacques
Mark_Kawada
raina_kinser
ehiludido
DriftersPsyche
gracexreec
akhilgupta_me
vascogsb
nomis6259
nenetteemk
Gorgioussdf
Rich65k
braykxo
kayansub
Brianketer5
blease_no
Key board interuption
Dady330
donynyn1
CurrentSocials
haramlaflame
_JamesShu
basiljh
Raima_Ouattara
arroba_551
samanthaaajae
RioNextDoor
```

```
eudoguinha
cxnhoto777
FreebandFlav4
oluwamisegun
CallMeKi__
CLeonard46
ONEeJuice
KatlyGold
mia_sansone
thcxns
slctiio
mariesimspon95
vanitascrimes
Jip8659
insimricky
XiggyMatsu
apk_share
HogardJacques
Mark_Kawada
raina_kinser
ehiludido
DriftersPsyche
gracexreec
akhilgupta_me
vascogsb
nomis6259
nenetteemk
Gorgioussdf
Rich65k
braykxo
kayansub
Brianketer5
blease_no
```

```
In [109]: Total_Data = []
          fo = open("stream.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))
```

```
1258
```

```
In [110]: 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 Tota
            Data["UserFollowersCount"] = [tweet.author.followers_count for tweet in Total_Da
            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_Da
            # 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

Out[110]: (1258, 19)

In [111]: Data.tail()

Out[111]:                  TwittID                                           TextData  \
          1253  1120030627615715328  RT @capribot: A golden prince was easy to lov...
          1254  1120030564990496770  RT @thiriumcupcakes: Jewish headcanons, anyone...
          1255  1120029794832396288  RT @harryhateskale: happy easter. welcome back...
          1256  1120029719779512331  RT @xor: using this caption for every one of m...
          1257  1120025386073636866  RT @skwrnf: #Hankcon Easter bunny!Connor\nsoft...

                     TweetCreatedAt  RetweetCount  TweetFavouriteCount  \
          1253 2019-04-21 18:24:45             9                    0
          1254 2019-04-21 18:24:30             9                    0
          1255 2019-04-21 18:21:26          2719                    0
          1256 2019-04-21 18:21:08          7397                    0
          1257 2019-04-21 18:03:55           127                    0

                        TweetSource               UserID UserScreenName UserName  \
          1253  Twitter for iPhone  1097059909231878145      blease_no     bich
          1254  Twitter for iPhone  1097059909231878145      blease_no     bich
          1255  Twitter for iPhone  1097059909231878145      blease_no     bich
          1256  Twitter for iPhone  1097059909231878145      blease_no     bich
          1257  Twitter for iPhone  1097059909231878145      blease_no     bich

                     UserCreatedAt                                    UserDescription  \
          1253 2019-02-17 09:07:19  Tester shame account while I figure this awful...
          1254 2019-02-17 09:07:19  Tester shame account while I figure this awful...
          1255 2019-02-17 09:07:19  Tester shame account while I figure this awful...
          1256 2019-02-17 09:07:19  Tester shame account while I figure this awful...
          1257 2019-02-17 09:07:19  Tester shame account while I figure this awful...

                UserDescriptionLength  UserFollowersCount  UserFriendsCount  \
          1253                    136                   4                51
          1254                    136                   4                51
          1255                    136                   4                51
          1256                    136                   4                51
          1257                    136                   4                51

               UserLocation  HttpCount  HashtagCount  MentionCount  TweetCount
          1253                       0             0             1        1668
          1254                       0             0             1        1668
          1255                       1             0             1        1668
          1256                       1             0             1        1668
          1257                       1             1             1        1668
```

```
In [112]: # Saving data with item space separating
          Data.to_csv('Leg_data9.csv', sep=',' , header = True )
```

```
In [113]: # saving data with item space separating
          Leg_Data = pd.read_csv('Leg_data9.csv')
          Total_leg = Leg_Data.drop('Unnamed: 0', 1)
          Total_leg.to_csv('Total_leg.csv', sep=',',encoding='utf8')
```

```
In [114]: # DAta Loading .............
```

```
In [115]: leg_data = pd.read_csv('Total_leg.csv')
          leg_data.fillna(0, inplace=True)
          leg_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1258 entries, 0 to 1257
Data columns (total 20 columns):
Unnamed: 0               1258 non-null int64
TwittID                  1258 non-null int64
TextData                 1258 non-null object
TweetCreatedAt           1258 non-null object
RetweetCount             1258 non-null int64
TweetFavouriteCount      1258 non-null int64
TweetSource              1258 non-null object
UserID                   1258 non-null int64
UserScreenName           1258 non-null object
```

```
UserName                  1258 non-null object
UserCreatedAt             1258 non-null object
UserDescription           1258 non-null object
UserDescriptionLength     1258 non-null int64
UserFollowersCount        1258 non-null int64
UserFriendsCount          1258 non-null int64
UserLocation              1258 non-null object
HttpCount                 1258 non-null int64
HashtagCount              1258 non-null int64
MentionCount              1258 non-null int64
TweetCount                1258 non-null int64
dtypes: int64(12), object(8)
memory usage: 196.6+ KB
```

```
In [116]: temp1 = leg_data
          temp1 = temp1[["RetweetCount"]]
          temp1.to_csv('temp11.csv',sep=',', encoding='utf8')
```

```
In [117]: leg_data.loc[:,'AvgHashtag'] = (leg_data.groupby('UserID')["HashtagCount"].transform
          leg_data.loc[:,'AvgURLCount'] = (leg_data.groupby('UserID')["HttpCount"].transform('s
          leg_data.loc[:,'AvgMention'] = (leg_data.groupby('UserID')["MentionCount"].transform
          leg_data.loc[:,'AvgRetweet'] = (leg_data.groupby('UserID')["RetweetCount"].transform
          leg_data.loc[:,'AvgFavCount'] = (leg_data.groupby('UserID')["TweetFavouriteCount"].t
```

```
In [118]: unique_leg_row = leg_data[["UserID", "UserScreenName", "UserCreatedAt", "UserDescrip
          leg_data1 = unique_leg_row.drop_duplicates()
          leg_data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43 entries, 0 to 1228
Data columns (total 13 columns):
UserID                    43 non-null int64
UserScreenName            43 non-null object
UserCreatedAt             43 non-null object
UserDescriptionLength     43 non-null int64
UserFollowersCount        43 non-null int64
UserFriendsCount          43 non-null int64
UserLocation              43 non-null object
AvgHashtag                43 non-null float64
AvgURLCount               43 non-null float64
AvgMention                43 non-null float64
AvgRetweet                43 non-null float64
AvgFavCount               43 non-null float64
TweetCount                43 non-null int64
dtypes: float64(5), int64(5), object(3)
memory usage: 4.7+ KB
```

```
In [119]: Total_spam_data = pd.read_csv("Spam_data.csv")
          Total_spam_data.fillna(0, inplace=True)
          Total_spam_data.shape

Out[119]: (5394, 21)
```

```
In [120]: Total_spam_data.loc[:,'AvgHashtag'] = (Total_spam_data.groupby('UserID')["HashtagCoun
          Total_spam_data.loc[:,'AvgURLCount'] = (Total_spam_data.groupby('UserID')["HttpCount'
          Total_spam_data.loc[:,'AvgMention'] = (Total_spam_data.groupby('UserID')["MentionCoun
          Total_spam_data.loc[:,'AvgRetweet'] = (Total_spam_data.groupby('UserID')["RetweetCoun
          Total_spam_data.loc[:,'AvgFavCount'] = (Total_spam_data.groupby('UserID')["TweetFavou
```

```
In [121]: unique_spam_row = Total_spam_data[["UserID", "UserScreenName", "UserCreatedAt", "Use
          spam_data1 = unique_spam_row.drop_duplicates()
          spam_data1.loc[:,"SpammerOrNot"]=1
          spam_data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 177 entries, 0 to 5364
Data columns (total 14 columns):
UserID                   177 non-null int64
UserScreenName           177 non-null object
UserCreatedAt            177 non-null object
UserDescriptionLength    177 non-null int64
UserFollowersCount       177 non-null int64
UserFriendsCount         177 non-null int64
UserLocation             177 non-null object
AvgHashtag               177 non-null float64
AvgURLCount              177 non-null float64
AvgMention               177 non-null float64
AvgRetweet               177 non-null float64
AvgFavCount              177 non-null float64
TweetCount               177 non-null int64
SpammerOrNot             177 non-null int64
dtypes: float64(5), int64(6), object(3)
memory usage: 20.7+ KB
```

```
/home/radhey/anaconda3/lib/python3.6/site-packages/pandas/core/indexing.py:362: SettingWithCopy
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  self.obj[key] = _infer_fill_value(value)
/home/radhey/anaconda3/lib/python3.6/site-packages/pandas/core/indexing.py:543: SettingWithCopy
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
```

```
  self.obj[item] = s
```

```
In [122]: frames = [leg_data1, spam_data1]
          Total_random_data = pd.concat(frames, axis=0, sort=False)
          Total_random_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 220 entries, 0 to 5364
Data columns (total 14 columns):
UserID                   220 non-null int64
UserScreenName           220 non-null object
UserCreatedAt            220 non-null object
UserDescriptionLength    220 non-null int64
UserFollowersCount       220 non-null int64
UserFriendsCount         220 non-null int64
UserLocation             220 non-null object
AvgHashtag               220 non-null float64
AvgURLCount              220 non-null float64
AvgMention               220 non-null float64
AvgRetweet               220 non-null float64
AvgFavCount              220 non-null float64
TweetCount               220 non-null int64
SpammerOrNot             177 non-null float64
dtypes: float64(6), int64(5), object(3)
memory usage: 25.8+ KB
```

```
In [123]: Total_random_data.reset_index()
          Total_random_data.to_csv('Total_random_data.csv',sep=',', encoding='utf8')
```

```
In [124]: Total_random_data = pd.read_csv('Total_random_data.csv')
          Total_random_data.fillna(0, inplace=True)
          Current_Time = datetime.datetime.strftime(datetime.datetime.now(), '%Y-%m-%d %H:%M:%
          Total_random_data.loc[:, "Current_Time"]=Current_Time
          Total_random_data.to_csv('Total_random1_data.csv', sep=',', encoding='utf8')
          Total_random_data = pd.read_csv('Total_random1_data.csv')
          Total_random_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 220 entries, 0 to 219
Data columns (total 17 columns):
Unnamed: 0               220 non-null int64
Unnamed: 0.1             220 non-null int64
UserID                   220 non-null int64
UserScreenName           220 non-null object
UserCreatedAt            220 non-null object
UserDescriptionLength    220 non-null int64
UserFollowersCount       220 non-null int64
```

```
UserFriendsCount         220 non-null int64
UserLocation             220 non-null object
AvgHashtag               220 non-null float64
AvgURLCount              220 non-null float64
AvgMention               220 non-null float64
AvgRetweet               220 non-null float64
AvgFavCount              220 non-null float64
TweetCount               220 non-null int64
SpammerOrNot             220 non-null float64
Current_Time             220 non-null object
dtypes: float64(6), int64(7), object(4)
memory usage: 29.3+ KB
```

```
In [126]: #debugging
          temp1 = Total_data[["UserCreatedAt"]]
          temp1.to_csv('temp111.csv', sep=',', encoding='utf8')
```

```
In [127]: Total_data =  Total_random_data
```

```
In [128]: #Adding features
          Total_data.loc[:,"Reputation"]=Total_data["UserFollowersCount"]/(Total_data["UserFol
          Total_data["Reputation"].describe()
          Total_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 220 entries, 0 to 219
Data columns (total 18 columns):
Unnamed: 0               220 non-null int64
Unnamed: 0.1             220 non-null int64
UserID                   220 non-null int64
UserScreenName           220 non-null object
UserCreatedAt            220 non-null object
UserDescriptionLength    220 non-null int64
UserFollowersCount       220 non-null int64
UserFriendsCount         220 non-null int64
UserLocation             220 non-null object
AvgHashtag               220 non-null float64
AvgURLCount              220 non-null float64
AvgMention               220 non-null float64
AvgRetweet               220 non-null float64
AvgFavCount              220 non-null float64
TweetCount               220 non-null int64
SpammerOrNot             220 non-null float64
Current_Time             220 non-null object
Reputation               220 non-null float64
dtypes: float64(7), int64(7), object(4)
memory usage: 31.0+ KB
```

```
In [129]: Total_data.SpammerOrNot.value_counts()

Out[129]: 1.0    177
          0.0     43
          Name: SpammerOrNot, dtype: int64

In [130]: # logitivity features
          data = Total_data
          data["Current_Time"] = pd.to_datetime(data["Current_Time"])
          data["UserCreatedAt"] = pd.to_datetime(data["UserCreatedAt"])
          data['AgeOfAccount'] = (data['Current_Time'] - data['UserCreatedAt'])/np.timedelta64
          cols = ['AgeOfAccount']
          data[cols] = data[cols].mask(data[cols]<0)
          data.AgeOfAccount.describe()

Out[130]: count     220.000000
          mean     1373.349460
          std      1163.271559
          min         1.270613
          25%       263.913782
          50%      1192.280353
          75%      2383.831085
          max      3895.502130
          Name: AgeOfAccount, dtype: float64

In [131]: data1 = data
          data1.loc[:, "TweetPerDay"] = data1["TweetCount"]/data1["AgeOfAccount"]
          data1["TweetPerDay"].describe()

Out[131]: count    220.000000
          mean      24.355166
          std       61.186687
          min        0.001594
          25%        1.890692
          50%        6.026452
          75%       21.099913
          max      484.883181
          Name: TweetPerDay, dtype: float64

In [132]: data1.loc[:,"TweetPerFollower"] = data1["TweetCount"]/data1["UserFollowersCount"]

In [133]: data1.TweetPerFollower=data1.TweetPerFollower.round(2).fillna(0)
          data1 = data1[np.isfinite(data1['TweetPerFollower'])]
          data1["TweetPerFollower"].tail(3)

Out[133]: 217    97.11
          218    62.26
          219    85.89
          Name: TweetPerFollower, dtype: float64
```

```
In [134]: Test_data = data1
```

```
In [135]: #Saving Total test data
          Test_data.reset_index()
          Test_data.to_csv('Total_test_data.csv', sep=',', encoding='utf8')
```

```
In [136]: # Final state of loading training and testing data............
```

```
In [137]: # loading training data
          Train_data = pd.read_csv('Total_training_data.csv')
          Train_data.fillna(0, inplace=True)
```

```
In [138]: # loadind test data
          Test_data = pd.read_csv('Total_test_data.csv')
          Test_data.fillna(0, inplace=True)
```

```
In [139]: # selecting the features for training and testing data
          Train = Train_data[['Reputation', 'AvgHashtag', 'AvgRetweet', 'AvgFavCount','AvgMenti
          y_train =Train_data["SpammerOrNot"]
          print("Training set value counts:\n")
          print(y_train.value_counts())
          Test = Test_data[['Reputation', 'AvgHashtag', 'AvgRetweet', 'AvgFavCount','AvgMentio
          y_test = Test_data["SpammerOrNot"]
          print("Testing set value counts:\n")
          print(y_test.value_counts())
```

```
Training set value counts:

0    369
1    171
Name: SpammerOrNot, dtype: int64
Testing set value counts:

1.0    177
0.0     43
Name: SpammerOrNot, dtype: int64
```

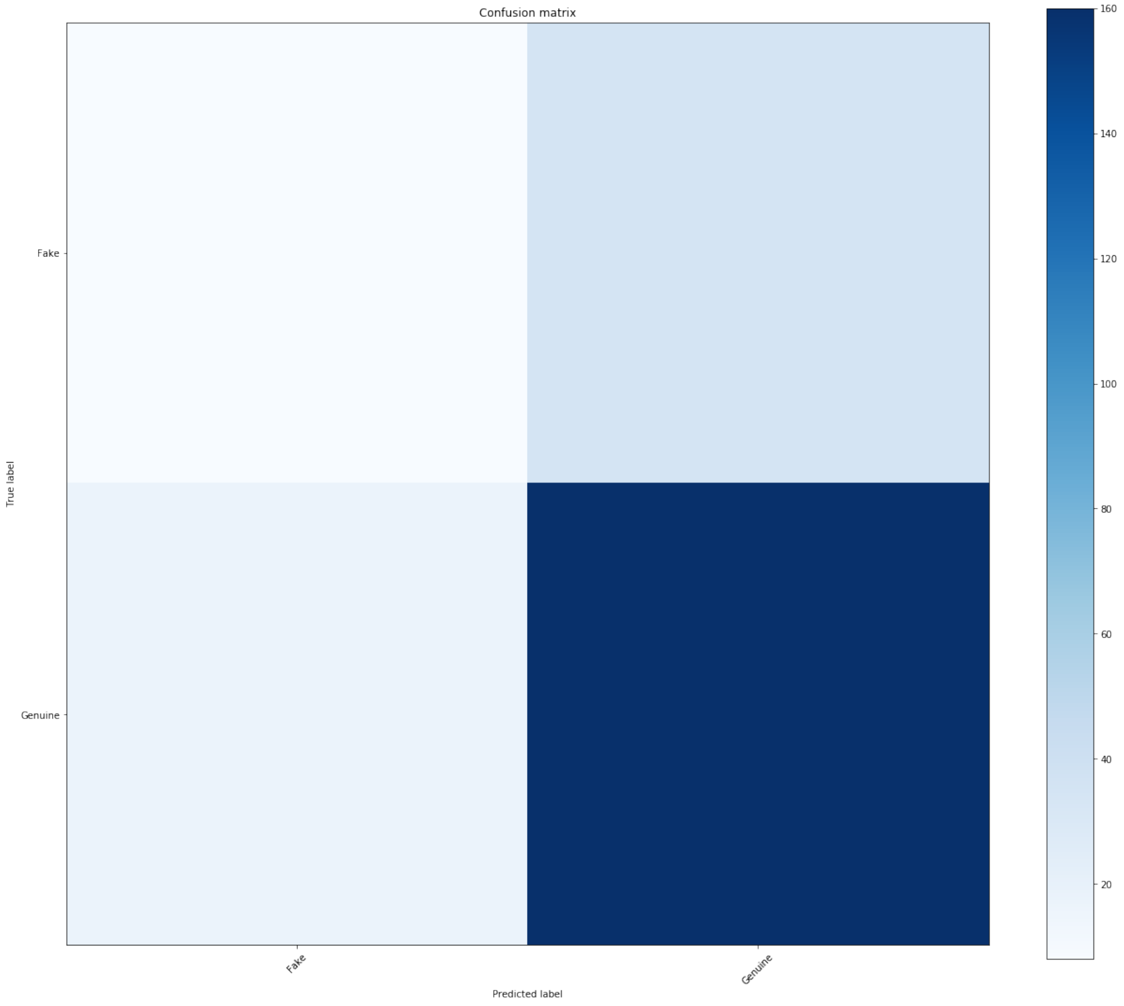
```
In [140]: from sklearn.ensemble import RandomForestClassifier
          est = RandomForestClassifier(n_estimators=11, max_depth=11, min_samples_split=8)
          est.fit(Train, y_train)
          y_pred = est.predict(Test)
          scores = cross_val_score(knn, M, y, cv=10, scoring='accuracy')
          print(accuracy_score(y_test,y_pred))
          print("Tenfol cross validation score")
          print(scores)
          print(scores.mean())
          print("\n")
          print("Classifier performance report: ")
```

```
        print(classification_report(y_test, y_pred))
        print("Confusion Matrix: ")
        print(confusion_matrix(y_test, y_pred))
```

```
0.7636363636363637
Tenfol cross validation score
[0.81818182 0.88888889 0.77777778 0.74074074 0.88888889 0.81481481
 0.96296296 0.96296296 0.96296296 0.90566038]
0.8723842195540309


Classifier performance report:
              precision    recall  f1-score   support

         0.0       0.32      0.19      0.24        43
         1.0       0.82      0.90      0.86       177

   micro avg       0.76      0.76      0.76       220
   macro avg       0.57      0.55      0.55       220
weighted avg       0.72      0.76      0.74       220

Confusion Matrix:
[[  8  35]
 [ 17 160]]
```

```
In [144]: def plot_confusion_matrix(cm, title='Confusion matrix', cmap=plt.cm.Blues):
              target_names = ['Fake', 'Genuine']
              plt.imshow(cm, interpolation='nearest', cmap=cmap)
              plt.title(title)
              plt.colorbar()
              tick_marks = np.arange(len(target_names))
              plt.xticks(tick_marks, target_names, rotation=45)
              plt.yticks(tick_marks, target_names)
              plt.tight_layout()
              plt.ylabel('True label')
              plt.xlabel('Predicted label')
              plt.show()
```

```
In [145]: cm = confusion_matrix(y_test, y_pred)
          plot_confusion_matrix(cm)
```


Confusion matrix
True label
Fake
Genuine
Predicted label
Fake
Genuine
160
140
120
100
80
60
40
20

```
In [ ]:
```