**SPAMMER DETECTION AND FAKE USER IDENTIFICATION IN SOCIAL NETWORKS**

**Introduction:**

* It has become quite simple to obtain any kind of information from any source across the world by using the Internet. The increased demand of social sites permits users to collect abundant amount of information and data about users. Huge volumes of data available on these sites also draw the attention of fake users . Twitter has rapidly become an online source for acquiring real-time information about users. Twitter is an Online Social Network (OSN) where users can share anything and everything, such as news, opinions, and even their moods.
* Recently, the detection of spam in social networking sites attracted the attention of researchers. Spam detection is a difficult task in maintaining the security of social networks. It is essential to recognize spams in the OSN sites to save users from various kinds of malicious attacks and to preserve their security and privacy.
* These hazardous manipulative skills adopted by spammers cause massive destruction of the community in the real world. Twitter spammers have various objectives, such as spreading invalid information, fake news, rumors,etc.
* Spammers achieve their malicious objectives through advertisements and several other means where they support different mailing lists and subsequently dispatch spam messages randomly to broadcast their interests. These activities cause disturbance to the original users who are known as non-spammers. In addition, it also decreases the reputation of the OSN platforms.
* Therefore, it is essential to design a scheme to spot spammers so that corrective efforts can be taken to counter their malicious activities

Abstract:

* Social networking sites engage millions of users around the world. The users interactions with these social sites, such as Twitter and Facebook have a huge impact and occasionally unpleasant things that happen in daily life.
* The prominent social networking sites have turned into a target platform for the spammers to dump a huge amount of irrelevant and hurtful information. Twitter, for example, has become one of the most commonly used platforms of all times and therefore allows an unreasonable amount of spam.
* Fake users send unwanted tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. Moreover, the possibility of expanding invalid information to users through fake identities has increased that results in massive harmful content.
* Recently, the detection of spammers and identification of fake users on Twitter has become a common area of research in contemporary online social Networks (OSNs). In this , we perform a review of techniques used for detecting spammers on Twitter.
* Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features.

**Objective:**

The main objective of this project is to detect spammers and identify fake users in OSN sites(TWITTER) by using Machine learning Algorithms like Naïve bayes classifier to find spam or no-spam and Random forest algorithm to find fake user or not in order to protect users from malicious attacks and also to maintain privacy and security and also to maintain reputation of the social networking sites

**Implementation:**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

from tkinter.filedialog import askopenfilename

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier

import json

import os

import re

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

import pickle as cpickle

main = tkinter.Tk()

main.title("Spammer Detection") #designing main screen

main.geometry("1300x1200")

global filename

global classifier

global cvv

global total,fake\_acc,spam\_acc

def process\_text(text):

nopunc = [char for char in text if char not in string.punctuation]

nopunc = ''.join(nopunc)

clean\_words = [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]

return clean\_words

def upload(): #function to upload tweeter profile

global filename

filename = filedialog.askdirectory(initialdir=".")

pathlabel.config(text=filename)

text.delete('1.0', END)

text.insert(END,filename+" loaded\n");

def naiveBayes():

global classifier

global cvv

text.delete('1.0', END)

classifier = cpickle.load(open('model/naiveBayes.pkl', 'rb'))

cv = CountVectorizer(decode\_error="replace",vocabulary=cpickle.load(open("model/feature.pkl", "rb")))

cvv = CountVectorizer(vocabulary=cv.get\_feature\_names(),stop\_words = "english", lowercase = True)

text.insert(END,"Naive Bayes Classifier loaded\n");

def fakeDetection(): #extract features from tweets

global total,fake\_acc,spam\_acc

total = 0

fake\_acc = 0

spam\_acc = 0

text.delete('1.0', END)

dataset = 'Favourites,Retweets,Following,Followers,Reputation,Hashtag,Fake,class\n'

for root, dirs, files in os.walk(filename):

for fdata in files:

with open(root+"/"+fdata, "r") as file:

total = total + 1

data = json.load(file)

textdata = data['text'].strip('\n')

textdata = textdata.replace("\n"," ")

textdata = re.sub('\W+',' ', textdata)

retweet = data['retweet\_count']

followers = data['user']['followers\_count']

density = data['user']['listed\_count']

following = data['user']['friends\_count']

replies = data['user']['favourites\_count']

hashtag = data['user']['statuses\_count']

username = data['user']['screen\_name']

words = textdata.split(" ")

text.insert(END,"Username : "+username+"\n");

text.insert(END,"Tweet Text : "+textdata);

text.insert(END,"Retweet Count : "+str(retweet)+"\n")

text.insert(END,"Following : "+str(following)+"\n")

text.insert(END,"Followers : "+str(followers)+"\n")

text.insert(END,"Reputation : "+str(density)+"\n")

text.insert(END,"Hashtag : "+str(hashtag)+"\n")

text.insert(END,"Tweet Words Length : "+str(len(words))+"\n")

test = cvv.fit\_transform([textdata])

spam = classifier.predict(test)

cname = 0

fake = 0

if spam == 0:

text.insert(END,"Tweet text contains : Non-Spam Words\n")

cname = 0

else:

spam\_acc = spam\_acc + 1

text.insert(END,"Tweet text contains : Spam Words\n")

cname = 1

if followers < following:

text.insert(END,"Twitter Account is Fake\n")

fake = 1

fake\_acc = fake\_acc + 1

else:

text.insert(END,"Twiiter Account is Genuine\n")

fake = 0

text.insert(END,"\n")

value = str(replies)+","+str(retweet)+","+str(following)+","+str(followers)+","+str(density)+","+str(hashtag)+","+str(fake)+","+str(cname)+"\n"

dataset+=value

f = open("features.txt", "w")

f.write(dataset)

f.close()

def prediction(X\_test, cls): #prediction done here

y\_pred = cls.predict(X\_test)

for i in range(len(X\_test)):

print("X=%s, Predicted=%s" % (X\_test[i], y\_pred[i]))

return y\_pred

# Function to calculate accuracy

def cal\_accuracy(y\_test, y\_pred, details):

accuracy = 30 + (accuracy\_score(y\_test,y\_pred)\*100)

text.insert(END,details+"\n\n")

text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")

return accuracy

def machineLearning():

text.delete('1.0', END)

train = pd.read\_csv("features.txt")

X = train.values[:, 0:7]

Y = train.values[:, 7]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 0)

cls = RandomForestClassifier(n\_estimators=10,max\_depth=10,random\_state=None)

cls.fit(X\_train, y\_train)

text.insert(END,"Prediction Results\n\n")

prediction\_data = prediction(X\_test, cls)

random\_acc = cal\_accuracy(y\_test, prediction\_data,'Random Forest Algorithm Accuracy & Confusion Matrix')

def graph():

height = [total,fake\_acc,spam\_acc]

bars = ('Total Twitter Accounts', 'Fake Accounts','Spam Content Tweets')

y\_pos = np.arange(len(bars))

plt.bar(y\_pos, height)

plt.xticks(y\_pos, bars)

plt.show()

font = ('times', 16, 'bold')

title = Label(main, text='Spammer Detection and Fake User Identification on Social Networks')

title.config(bg='brown', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 14, 'bold')

uploadButton = Button(main, text="Upload Twitter JSON Format Tweets Dataset", command=upload)

uploadButton.place(x=50,y=100)

uploadButton.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='brown', fg='white')

pathlabel.config(font=font1)

pathlabel.place(x=470,y=100)

fakeButton = Button(main, text="Load Naive Bayes To Analyse Tweet Text or URL", command=naiveBayes)

fakeButton.place(x=50,y=150)

fakeButton.config(font=font1)

randomButton = Button(main, text="Detect Fake Content, Spam URL, Trending Topic & Fake Account", command=fakeDetection)

randomButton.place(x=520,y=150)

randomButton.config(font=font1)

detectButton = Button(main, text="Run Random Forest For Fake Account", command=machineLearning)

detectButton.place(x=50,y=200)

detectButton.config(font=font1)

exitButton = Button(main, text="Detection Graph", command=graph)

exitButton.place(x=520,y=200)

exitButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=30,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

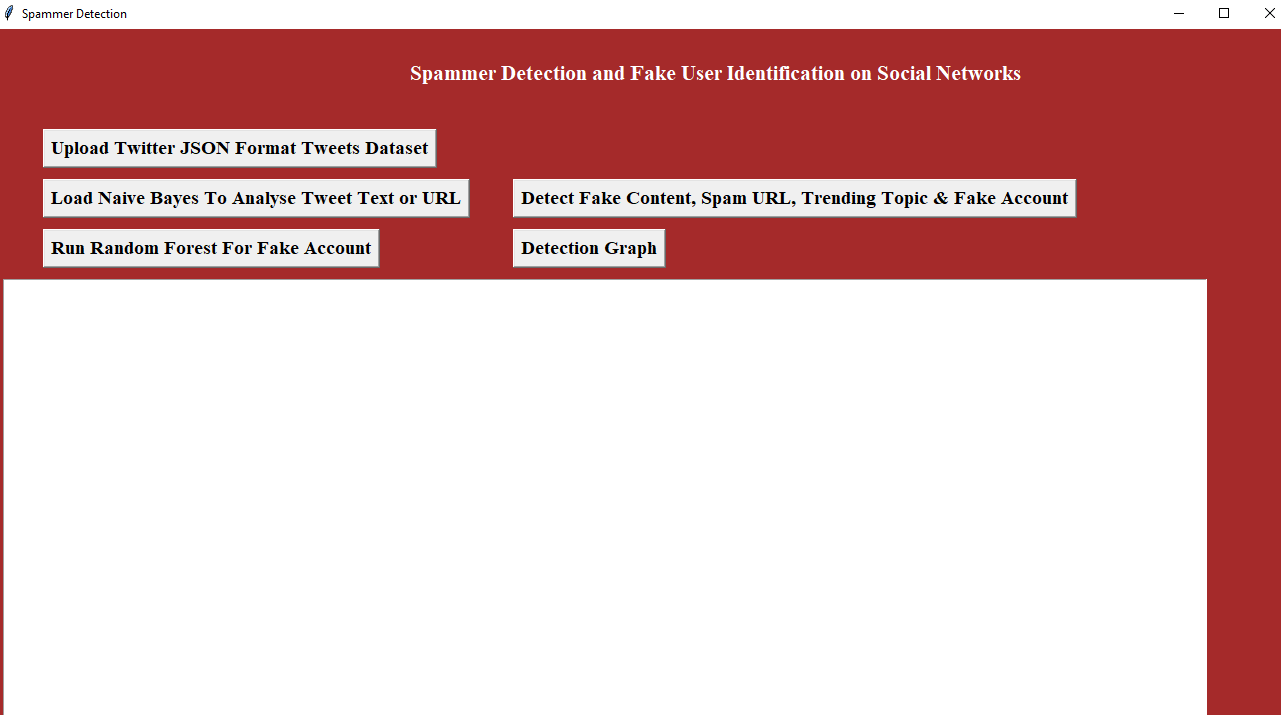
text.place(x=10,y=250)

text.config(font=font1)

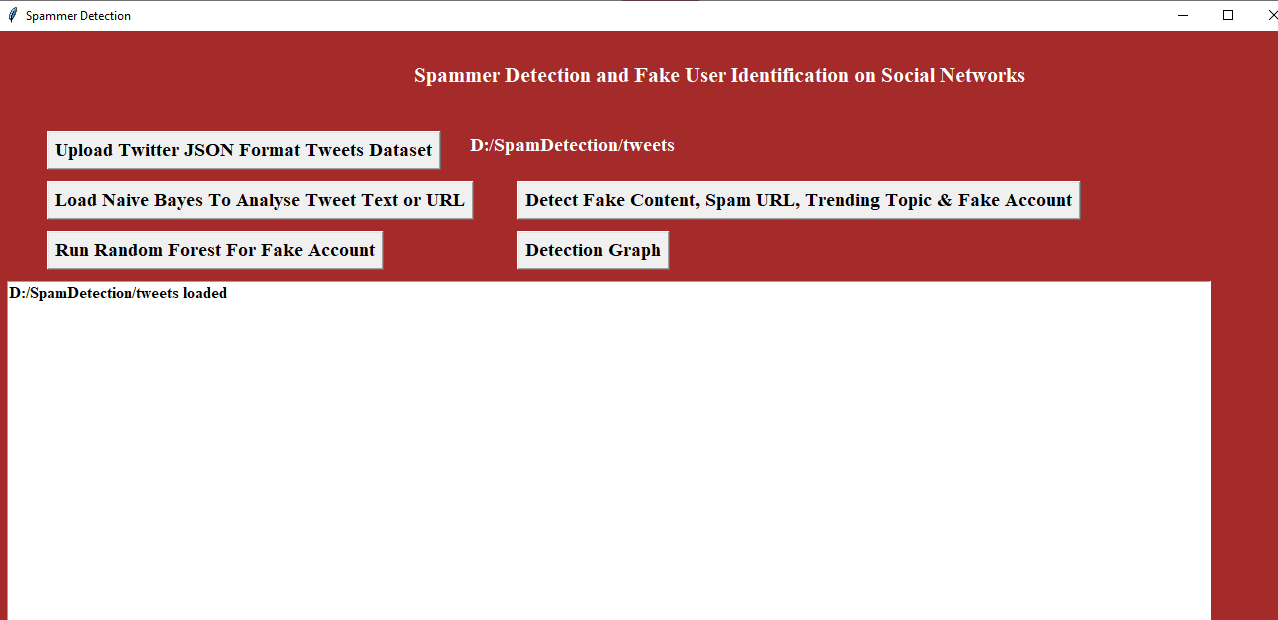
main.config(bg='brown')

main.mainloop()

**Experimental Results**:



In above screen click on ‘Upload Twitter JSON Format Tweets Dataset’ button and upload tweets folder



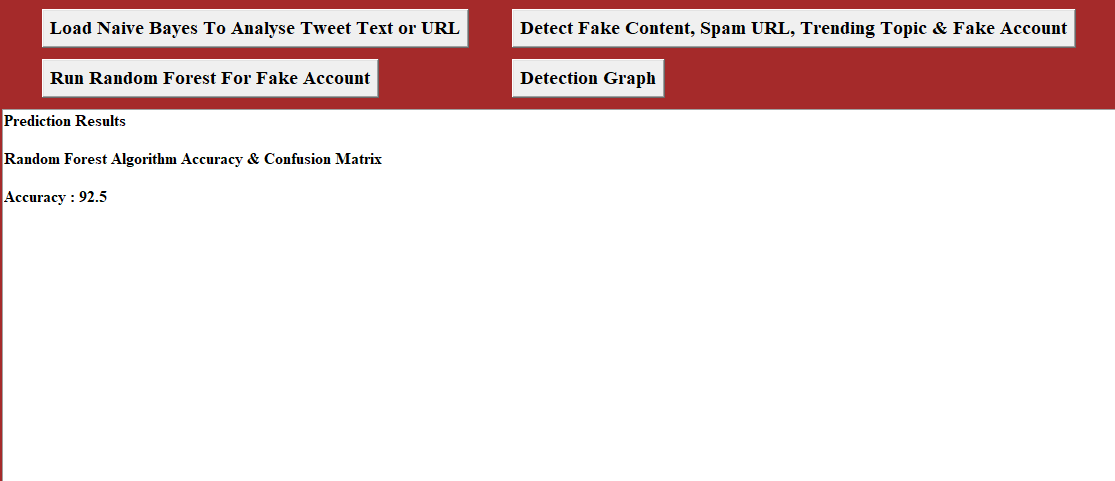
In above screen we can see all tweets from all users loaded



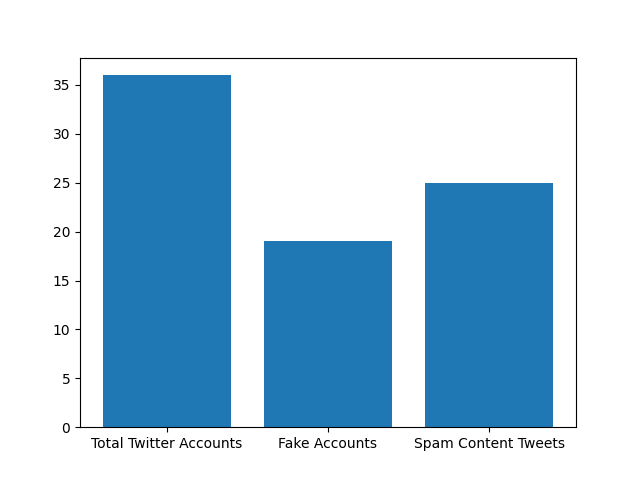
In above screen naïve bayes classifier loaded



In above screen all features extracted from tweets dataset and then analyse those features to identify tweets is no spam or spam. In above text area each records values are separated with empty line and each tweet record display values as TWEET TEXT, FOLLOWERS, FOLLOWING etc with account is fake or genuine and tweet text contains spam or non-spam words.



In above screen we got random forest prediction accuracy as 92.5%.



In above graph x-axis represents total tweets, fake account and spam words content tweets and y-axis represents count of them