Twitter Bot Detection

Datta Sainath Dwarampudi
Computer Science,
New York University
New York, USA
dsd298@nyu.edu

Madhu Kiran Gudivada
Computer Science,
New York University
New York, USA
mg5309@nyu.edu

***Abstract*— In recent years, social media accounts are not only controlled by humans but also by bots. Recent literature has focused primarily on detection of bot in social networks. These bots act as a double-edged sword for a common social media user. Few bots generate large amount of data about news and updating feeds, while other bots spread spam or malicious data through tweets, which is of major concern. In this project, we are trying to design, analyze and implement classifiers to detect the probability of a given user account as a bot or not. We shall use 2 classifiers based on Naïve Bayes and Logistic regression for tweet analysis. We shall compare the classifiers and try to improve the better classifier till we get satisfied accuracy.**

Keywords—machine learning, bots, Naïve Bayes, logistic regression, twitter, Natural Language Processing

I. INTRODUCTION

Our project proposal is to determine a user account as a legitimate user account or a bot. There is tremendous work been done in this domain. Bots can be used to generate live scores, weather and news. They can also be used to distribute malicious tweets which have huge consequence on society. There are. According to Emilio Ferrara, a computer scientist and assistant professor at the University of Southern California (USC) has said that online bots which influence the political discourse on social media as never. We shall employ two algorithms based on Naïve Bayes and logistic regression for tweet analysis and then hone one of the algorithm of the above two which gives a better accuracy for bot detection. We shall modify and enhance the code to get better accuracy. Online marketers on twitter are also affected due to huge number of bot in the twitter ecosystem.

II. MOTIVATION

Twitter is one of the most popular social media platforms, it has been plagued by many bots during recent years. This has been a major problem to deal with. A recent publication states that more than half of the twitter accounts are not human. Few other optimistic studies states that 5-9% of the overall population is a bot and these bots generate 24% of all the tweets produced on twitter. In a case, a bot campaign created fake "buzz" about a tech company: automated stock trading algorithms acted on this chatter, resulting in a spurious 200-fold increase in market price. This has motivated us to actively deal with this issue by designing good classifiers using naïve Bayes and logistic regression for tweet analysis.

III. RELATED WORK

We are referring a lot of published papers and websites to complete this project successfully. We tried to refer papers which dealt with how twitter was in the in initial phase [1],[2], to better understand about the community of social media and its usage statistics. Krishnamurthy et al. [1] has studied and segregated twitter into 3 groups: 1) broadcasters, which have a large number of followers; 2) acquaintances, which have about the same number on either followers or following; and 3) miscreants and spammers, which follow a large number of follower but have few followers. Twitter has attracted spammers to post spam content, due to its popularity and openness. Fighting against spam on Twitter has been investigated in recent works [2],[3]&[4]. Yardi et al. [2] dated spam on twitter. His observations, include that spammer send more messages than legitimate users, and more likely to follow other spammers than legitimate users. Thus, a high follower-to-following ratio is a sign of spamming behavior. We are also mainly referring from papers recommended by professors [5],[6] & [7].

IV. DATA

We have gathered 50 bot accounts and 50 user accounts from various sources. It was very tough to gather data for bot accounts. We gathered bot data primarily from major websites which have detailed bot accounts listed in its websites. We even obtained data for bots from news website like CNN and other major websites which increased the number of followers of prominent politicians in United States of America. We tried searching for bots from various other websites, which helped people to build bots and some bot accounts were listed as the work performed by the previously. Some of the statuses contained symbols like emoticons, which were converted to UTF-8 format.

V. ALGORITHM(S)

The major part of any analysis for twitter data would be the tweets. The way a bot tweets and a human tweet plays a major part in the classification of the accounts. So, for the midway report of the project would be highly concentrated on tweets of both users and bots . We gathered the last 200 tweets of all the users and bots. We cleaned the tweets received and then applied the algorithms on them to train the classifiers.

A. Extracting Data

We created new twitter applications for getting the API Keys. We installed twitter libraries to run our code. We used Twitter API: api.user.timeline() to extract the

tweets of every user and bot accounts' tweets.

We then stored all of the bots tweets in a csv file named "Bots_tweets.csv" and all of the user tweets in a csv file named "Users_tweets.csv"

B. Cleaning the Data

This is the most important and crucial step of the whole project. This can be treated as a very trival step in this project but it is very important in cleaning the data. This step involves correction, detection and removal of unwanted text/data of errors and inconsistences present in the database due to inaccurate data retrieval or entry. If data is not cleaned properly, it will lead to number of problems like false conclusion and inappropriate fit to the classifier. However, data cleaning also has a negative effect on data. We may have a risk of loss important or valid data. But, in this project, we have taken enough care not to lose vital data.

The data received from the twitter API is crude and contains a lot of unwanted symbols(emoticons) and hex values. We used the NLTK package to remove these symbols. On removing these unnecessary symbols. We tried tokenizing the tweets from which stop words were removed easily. The remaining words were stemmed to ensure that the count vectorizer identifies the similar words efficiently.

C. Data Labelling & Conversion

We needed to include this step before starting to implement the algorithm/classifiers because till this stage we have always written induvial scripts for bots and user accounts for extracting tweets and cleaning tweets. We wanted to maintain separate csv files and python scripts for bot and user accounts for better documentation and better readability of code.

On extracting cleaned tweets of bots and users, we labeled the tweets to distinguish between them. The labelled data was converted to numerical data using CountVectorizer() function from sklearn package. CountVectorizer() supports counts of N-grams of words or consecutive characters. After fitting, the vectorizer must have a dictionary full of feature indices. The index value of a word in the vocabulary is linked to its frequency in the whole training. Generally, occurrences can give a good estimate of frequency of words occurring . If we dealt with larger documents, we must divide the number of occurrences of each word in a document by the total number of words in the document, this is termed as term frequencies. We work on small documents called tweets, therefore it is not a compulsion to implement term frequency times inverse document frequency. It is not going to be a big difference even if implemented. The data was converted from text data to numerical to feed it into the classifier.

D. Training Data

All the cleaned data which was converted to numerical data is split into Training Data and Test Data using a 10-fold cross validation using the Stratified K-Fold technique.

K-Fold Cross-validation is used in this project instead of manually splitting the sets to training and testing data sets because Cross-validation approach can estimate how accurately a predictive model will perform in real time. The former approach is much better than the later for prediction in the future twitter accounts. In case of stratified K-Fold Cross-validation , folds are selected so that the mean response value is approximately equal in all the fold.

E. Algorithms implemented

One the best methods to classify text data is Naïve Bayes. As Wikipedia states that "Naïve Bayes is a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features". For discrete features like the ones encountered in document classification (include spam filtering), Multinomial and Bernoulli distributions are very popular.

1. Multinomial Naïve Bayes:

Wikipedia defines Multinomial Naïve Bayes as "A multinomial event model, samples (feature vectors) that represent the frequencies with which certain events have been generated by a multinomial where the probability that event occurs". A feature vector is represented as a histogram, by counting the number of times event was observed for an instance. This is the event model typically is used for document classification, with events representing the occurrence of a word in a single document.

2. Bernoulli Naïve Bayes.

In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks where binary term occurrence features are used rather than term frequencies. This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms.
A Naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

3. Logistic Regression

We have used the Naïve Bayes in the previous 2 algorithm very well. In the third algorithm, we have used another popular method to classify the data i.e. by regression. Logistic regression can be defined as "a regression model where the dependent variable (DV) is categorical". Logistic regression can be used for

categorical outputs like pass/fail, win/lose, alive/dead, healthy/sick and in our case bot/nonbot. This can be best suited for our twitter bot detection as the output could be only two values, "0" in case of users and "1" in case of bots. Therefore, we can definitely use this algorithm for our project and no need to use multinomial logistic regression. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. Thus, it treats the same set of problems as probity regression using similar techniques, with the latter using a cumulative normal distribution curve instead

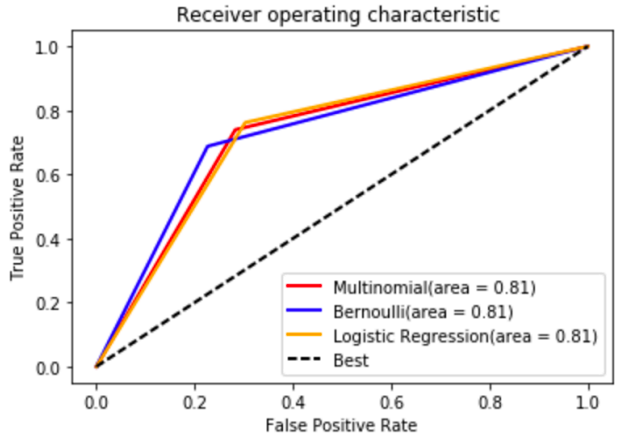
VI. PERFORMANCE AND RESULTS

We have used sklearn package to calculate metrics of our models. With the help of sklearn.model_selection package and matlplotlib.pyplot to calculate and plot the results.

We calculated the average accuracy, recall, precision, f1 and roc_auc(area under curve) scores for the 10-fold values and plotted an ROC curve by using the TPR (True Positive Rates) and FPR (False Positive Rates)

	MultinoialNB	BernoulliNB	LogisticReg
accuracy	0.7305	0.7227	0.7357
precision	0.7929	0.8159	0.7855
recall	0.7396	0.6879	0.7621
f1 score	0.7634	0.7436	0.7724
roc_auc	0.8121	0.8146	0.8108

Table 1: Average values of different parameters of all the implemented algorithms



Graph 1: ROC graph of implemented algorithms

We used Mean Squared Error to estimate the error rates of the 3 classifiers implemented.

	MultinoialNB	BernoulliNB	LogisticReg
MSE	0.2871	0.2772	0.2704

Table 2: Mean Squared Error for implemented algorithms

VII. CODE

Code Link: https://github.com/dattasainathd/TwitterBot3

VIII. FUTURE WORK

The performance of the classifiers based purely on tweet data has yield average performance which can be observed from the evaluation parameters. Also, the low mean squared error on the cross-validation data is very low which shows that the classifier is having high bias which leads to under fitting of the model.

This problem has raised because of lack of complexity of the classifier. Since we are using only one parameter i.e. the tweet data for estimation the model has become biased.

So, we have decided to increase the number of parameters used for classification by add the following,

1) **Sentimental Analysis**: We would add Twitter specific sentiment analysis algorithms, including happiness, arousal-dominance-valence, and emoticon scores.

2) **Friends Hierarchy** that include an account's social contacts, such as the median, moments, and entropy of the distributions of their number of followers, posts, and so on

3)**User metadata** which include the language, time of creation, locations etc.

4)**Behavior Features** like when a user posts a tweet and time between tweets and length of tweets etc.

We are planning to add the feature scores to Random Forests classifier which will be trained to classify the test data.

VIII. REFERENCES

[1] B. Krishnamurthy, P. Gill, and M. Arlitt, "A Few Chirps about Twitter," Proc. First Workshop Online Social Networks, 2008.

[2] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd, "Detecting Spam in a Twitter Network," First Monday, vol. 15, no. 1, Jan. 2010.

[3] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: The Underground on 140 Characters or Less," Proc. 17th ACM Conf. Computer and Comm. Security, pp. 27-37, 2010.

[4] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended Accounts in Retrospect: An Analysis of Twitter Spam," Proc. ACM SIGCOMM Conf. Internet Measurement Conf., pp. 243-258, 2011.

[5] Emilio Ferrara, Onur Varol, Clayton Davis, Flippo Menczer, Allessandro Flammini, "The Rise of Social Bots", Communications of ACM 59 (70, 96-104, 2016

[6] Lee, et al. Who Will Retweet This? Automatically Identifying and Engaging Strangers on Twitter to Spread Information.

[7] Erin Shellman, http://www.erinshellman.com/bot-or-not/.