How to detect Twitter bots (robots, or automated accounts)

Ayush Jain,
Masters of Science student
Department of Computer science
New York University
Aj2138@nyu.edu

Saurav Mawandia
Masters of Science student
Department of Computer science
New York University
saurav@nyu.edu

Abstract: This paper proposes algorithm to predict whether a given twitter account is bot or nor? It uses machine learning algorithms such as Random forest, Naïve Bayes and Decision tree classifiers on different attributes of a user

Keywords: machine learning

I. INTRODUCTION

TWITTER bots are computer programs which tweets or follows a user without human intervention. Humans generally use their twitter account through mobile applications or website, while bots either uses Application Programming Interface (API's) or other channel such as coded UI which help them to tweet/ retweet at real time, they can follow a limited number of users (Twitter limits such bots to follow 2000 user, if they don't have enough number of follower's). The bot which are programmed in many languages are hosted on cloud. Thus, they are active at any given point of time. The decreasing cost of cloud and the convenience twitter provides has immensely increased the number of bots in Twitter. A recent study conducted at University of Southern California concluded that there can be as many as 48 million twitter bots [1]

Bots have both pros and cons, on one hand legitimate bots help in tweeting news, blog updates, etc., which is in line with the company's goal to be a news and information channel. On the other hand, malicious bots are used for spamming user's profile, to increase the number of followers of a user, to promote fraudulent activities, phishing or spreading malicious content. For instance, in recent presidential election in U.S. It is alleged bots where used to increase the followers and retweet the tweet of the candidates. This may mislead people, which is not fair in a democratic setup. Thus, there is an immediate need to device a algorithm which can detect malicious bots.

II. MOTIVATION

Recently, Twitter announced it had 319 million monthly active users worldwide, but a study reveals out of those 319 million, as many as 48 million are bot which is over 15% of the active users. These bots are sometimes annoying as they often span other user's profile and possess a threat to the existing user base. A famous personality can be defamed by making bots to follow their profile and then publicly exposing their fake followers. Newt Gingrich, Mitt Romney and the German Conservative Party (CDU) [2] are just some examples of cases where fake-following was proven or assumed to have happened.

We can conclude that the bots are now not only used for marketers looking for fame and sales success but also for malicious purposes. In fact, bots are now big government's business. For example, the US Air Force revealed that it solicited Netrepid, a California based company, to create software that would enable it to mass-produce bots for political purposes. [2]

The aim of bot creators is to make it more humanly, so that no one can predict whether it is a bot or not? On a large scale, they skew the analysis of 'trending topics' on social media networks which is the base for journalists to make their stories. Think about the power if the tweets from your bot come up in google search results. Thus, a bot can overtake media industry by making something look important. So, it is an immediate necessity to come out with an idea to stop such bots from overtaking humans accounts.

III. RELATED WORKS

Many researches have done work on this area to detect a twitter bot. One of the famous works by Emilio Ferrara et al. at Indiana University in Bloomington have created an algorithm to detect a user is bot or not and have hosted it on http://truthy.indiana.edu/botornot . The technique they used was reasonably straightforward. They started gathering a set of social bots from the original group outed in 2011. They picked 15,000 of these and collected their 200 most recent tweets as well as the 100 most recent tweets mentioning them. That produced a dataset of some 2.6 million tweets. The team then collected a related dataset for 16,000 human users consisting of more than 3 million tweets and looked at over 1000 features with these accounts, such as the number of tweets, retweets, mentions and retweets received etc.. [3]. The bots are becoming more smart these days and thus many a times this algorithm fails to predict whether a user is bot or not.

IV. DATA

We have mined 100 accounts (50 bots, 50 human users) and extracted important fields such as tweeter id, screen name, location, description, followers, friends, language, status, name, has profile image etc. The human users were taken from the verified profile from our accounts. The bots were chosen after a lot of analysis. We identified a celebrity and checked for users who retweeted all his tweets. We then identified a pattern, the bots generally don't have profile pics, they don't have a bio, they have a different user name, screen name, either they have many tweets or very less tweets. The bots have less followers and they follow more number of people. They generally don't follow not more than 2000 followers. Then we used a python script to extract all the required fields using the user name. The script and data are uploaded on https://github.com/sauravmawandia/twitter-bot-analysis-ML.

V. ALGORITHMS(S) USED

We have used three ML Algorithms i.e. Random forest, Decision Tree and Naïve Bayes. We loaded all the useful data to a single data frame combining the two CSV file of bots and nonbots. And then cleaned the data from the dataset by taking only values which were useful for analysis. And then converted few attributes which were object to Boolean type. Such as, URL, description, location to whether they are present or not. Few Boolean attributes were objects we also converted them to Boolean.

We trained our model with 70% of data and used 30% for cross validation. We used sklearn library which have algorithms for all the classifiers we have trained and for splitting the train and test data. Now all the data we have is clean and we take all the training data and train the model and then used 5-fold cross validation.

Random forest algorithm uses a decision tree as a predictive model which maps observations to conclusions about the item's target value (represented as branches and the leaves). It is a good predictive modelling approach used in statistics, data mining and machine learning. We will construct a multitude of decision trees to train the algorithm and output the bot class. In the analysis, we have used attributes such as profile image present or not, status and build a decision tree which created random forest to train model.

Entropy is used for splitting the data. Split is decided on other factors as well. An attribute is chosen for split if it has at least 10 samples. We restrict the split by taking at least 50 sample leaves. This makes sure we don't overfit the data. We taring the data using the sklearn.RandomForestClassifier with our modification by supplying the parameters we have changed on the basic classifier. One we train the model, we do cross validation. And find out the required metrics to compare it with other ML algorithms.

Decision tree is acyclic graph that uses branching decision such as entropy and gini to illustrate the outcome of a decision. We use the sklearn decision tree to train the model with 70% data and 30% cross validation. We make sure we don't overfit the data by taking attributes by performing chi square test. We then calculate the metrics for the algorithm.

Naive Bayes classifiers are scalable, requiring several parameters. The maximum likelihood predictor can help us to predict in linear time. Some of the attributes which we have used to predict are number of friends, status, has profile image, number of followers etc. We have used multinomialNB to train our model and performed cross validation and displayed the accuracy, AUC score, precision, recall, f1 score, support.

VI. RESULT

Random Forest:

Accuracy of Random forest is 84.6268656716%.
The AUC Score for Random forest is 0.846295107688.

	precision	recall	f1-score	support
0	0.85	0.85	0.85	342
1	0.84	0.85	0.84	328
avg/total	0.85	0.85	0.85	670

Naïve Bayes:

Accuracy of Naive Bayes is 65.9701492537 %
AUC score for Naive Bayes is 0.666167451148

	precision	recall	f1-score	support
0	0.94	0.36	0.52	342
1	0.59	0.98	0.74	328
avg/total	0.77	0.66	0.62	670

Decision tree:

Accuracy of Decision Tree 82.6865671642 %
AUC score for Decision Tree is 0.826540436457

	precision	recall	f1-score	support
0	0.82	0.82	0.83	342
1	0.83	0.81	0.82	328
avg/total	0.83	0.83	0.83	670

The result clearly shows that random forest has a good accuracy on the given data. The high AUC score also supplements that. Naïve Bayes is not a good classifier for the given data. Random forest also has a good accuracy.

VII. CODE

The code is committed to the git hub repository https://github.com/sauravmawandia/twitter-bot-analysis-ML . The data is in data folder and we have IPython Notebook for the algorithm we have discussed above.

VIII. References

1. http://www.cnbc.com/2017/03/10/nearly-48-million-twitter-accounts-could-be-bots-says-study.html

2. http://www.lutzfinger.com/evil-business-social-media-bots/

3. The rise of Social bots, Emilio Ferrara, Onur varol, Clantin Davis, Flipppo menczer, allessandro flammini, Indian university.