Detecting Twitter bots (robots, or automated accounts)

Sanket Nawle
NYU Tandon School of Engineering
New York, U.S.
skn288@nyu.edu

Jayesh Patil
NYU Tandon School of Engineering
New York, U.S.
jpp421@nyu.edu

I. INTRODUCTION

Twitter is an online news and social networking service where users post and interact with messages, "tweets", restricted to 140 characters. Registered users can post tweets, but those who are unregistered can only read them. Users access Twitter through its website interface, SMS or mobile device application [11]. Some people may use Twitter as a newsfeed where they may obtain important news by following particular accounts, whereas some may use it as a means to make people aware about certain things.

In order to understand how popular Twitter is, the following are some of the statistics: There are a 310M monthly active Twitter users and a total of 1.3 billion accounts have been created. There are about 500 million tweets sent per day.

According to the above statistics, it is abundantly clear that Twitter is a very popular application. Since, Twitter can be such an influential platform, people have developed programs known as twitter bots. These bots can be used in a variety of ways such as for increasing number of followers, spamming, retweeting, etc. This practice is becoming so popular that it is estimated that there are about 48 million twitter bots already.

The main aim of this project is to distinguish twitter bots from real accounts. In order to do this, we will be considering certain attributes along with a classification model. We are aiming to obtain high accuracy for the classification. We are planning to use a classification model such as perceptron training model so that the accuracy keeps on increasing as the inputs increase and we retrain the model. Also, since the number of twitter bots as well as real accounts is very large and growing, it will help the model to increase its accuracy over the time. Although it should be noted that classification is always subject to probabilities; we can never be 100% sure that the classification is valid. It should also be noted that as we increase the number of attributes, the accuracy will increase too since the classification model will have a larger distinguishing criteria.

Also, since most of the classification models require threshold values, we are planning to use clustering techniques like KNN clustering to better understand the thresholds. These thresholds can then be used to affect the output of the classification model and hence, we may be able to train the model more efficiently.

II. MOTIVATION

21st century is certainly the era of communication. Social media has been one of the most used form of communication these days. It was revolutionized the way people connect and communicate with each other. Some of the statistics of the social media usage are as follows: 83% of the Americans have some kind of social media account. 86% of Twitter users rely on the site for the latest news. 91% of retail brands use two or more social media services to publicize their product. More than 130,000 advertisers use Twitter actively.

People use social media for communicating a variety of information ranging from their life events to crucial updates. Twitter is certainly a very popular social media service. In order to take advantage of this massive community, people have developed twitter

bots. These bots produce automated tweets. The functions of these bots range from serving as spams to automatically retweeting to a post. According to a study released by University of California, these bots contribute to approximately 24% of the total tweets on Twitter. So, when you think that you are having a conversation with a person on twitter, it may happen that it is actually just a bot program. Also, it is fairly easy to create such bots given the amount of information and services easily available on the internet. These twitter bots are usually benign and silly, but it may not be long enough when we start having bots which are smart enough to fool people. Thus, it would prove to be beneficial to distinguish bots from people on twitter.

III. RELATED WORK

[1] Agarwal, N., Liu, H., Tang, L., and Yu, P. S. Identifying the influential bloggers in a community. In.WSDM, 2008.

[2] Boyd, D., Golder, S., and Lotan, G. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. In HICSS, 2010

[3] Lee, K., Caverlee, J., and Webb, S. Uncovering social spammers: social honeypots + machine learning. In SIGIR 2010.

[4] Lee, K., Eoff, B. D., and Caverlee, J. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. In ICWSM, 2011.

[5] Yarkoni, Tal. Personality in 100,000 words: A large-scale analysis of personality and word usage among bloggers. Journal of Research in Personality, 2010.

[6] Wilkie, A., Michael, M. and Plummer-Fernandez, M. (2015), Speculative method and Twitter: Bots, energy and three conceptual characters. Sociol Rev, 63: 79–101. doi:10.1111/1467-954X.12168

[7] Mohammad Shafahi, Leon Kempers, Hamideh Afsarmanesh, "Phishing through social bots on Twitter", *Big Data (Big Data) 2016 IEEE International Conference on*, pp. 3703-3712, 2016.

[8] Gregory Maus, "Decoding hacking and optimizing societies: Exploring potential applications of human data analytics in sociological engineering both internally and as offensive weapons", *Science and Information Conference (SAI) 2015*, pp. 538-547, 2015.

[9] http://truthy.indiana.edu/botornot/

[10] ttp://www.erinshellman.com/bot-or-not/

[11] https://en.wikipedia.org/wiki/Twitter

IV. DATA

The data has been fetched from Twitter using their API (tweepy). The data dictionary consists of: id, Id_str, Screen_name, Location, Description, Url, Followers_count, Friends_count, Listed_count, Created_at, Favourites_count, Verified, Statuses_count, Lang, Status, Default_profile, Default_profile_image, Has_extended_profile, name and Bot. It consists of 2232 observations, out of which 1056 are bots and rest are human.

id	id_str	screen_na	location	descriptio	url	followers_	friends_cc
3.98E+09	3.98E+09	mcgucket_bot		A bot that tweets ev		1129	7
8.41E+17	8.41E+17	BowieK66				0	22
2.77E+09	2.77E+09	ducknoteprice				3	0
3.3E+09	3.3E+09	robotrecip	robot kitcl	tasty recip	http://t.cc	505	13
3.22E+09	3.22E+09	everyumlaut		bot by @dbaker_h		15	0
7.30E+17	7.30E+17	glossatory	Australah	SOCIAL OI	https://t.c	16	1
8.20E+17	8.20E+17	Fancypant	vancouver			41	394
2.6E+09	2.6E+09	Hedgehog	Moebius	@Hedgeh	http://t.cc	549	370
8.33E+17	8.33E+17	jamieph93986621		I wasn't bor	yesterday	0	60
8.25E+17	8.25E+17	NothemDonella		Your diac	No	0	43
2.22E+09	2.22E+09	pony_strategies		Best Strategy Guides		102	3

Image 1: A snippet of data set

For analysis of this data, we have narrowed our choice of information to limited attributes, viz., Followers_count, Friends_count, Listed_count, Favourites_count, Verified, Statuses_count, Default_profile, Default_profile_image, Has_extended_profile. Since we intend to use trees as our algorithms, removing attributes having distinct values allows us to eradicate the notable problem of decision tree (Higher Information gain for attributes having distinct values). Once the columns were dropped, we analysed the remaining for missing values. We found the presence of missing or abnormal values in a particular column. We cleaned the data by dealing with missing values. In our code, we substituted ‘false’ in place of missing values of “Has_extended_profile”

attribute.

Also by our initial inspections, we came to realise that many of the twitter bots have the word ‘bot’ in either their name, screen name or their description. Hence, we derived another attribute called name_bot which has the value ‘1’ if the twitter account has the substring ‘bot’ in its name, screen name or description and value ‘0’ otherwise. Also, we derived another attribute called ‘age’ using the original attribute ‘created_at’ since twitter bots came into existence after a while twitter became such a popular social media site. But the accounts which were created before the rise of these twitter bots are naturally, real accounts.

V. ALGORITHM(S) USED

1. **Decision Trees**: A commonly used model which predicts the value of result class, given information in the form of attributes. It could be realized as an “if - else” loop where the model checks for a condition and moves to the next level until result class. Each leaf represents a value of the result class. The essence of Decision Tree lies in generating the condition to be checked through the data. This condition is formed by calculating Entropy and Information gain.

Entropy is calculated as follows:

$$H(T) = I_E(p_1, p_2, \ldots, p_n) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

Information Gain is calculated as follows:

$$IG(T, a) = H(T) - H(T|a)$$

The information gain decides which attribute to split on each step to build the tree. At each step, we choose the nodes which leads to purely result class. Otherwise, the one with higher information gain is chosen for the split. Consider the example of decision tree below. Consider a result label of new observation is to be predicted, ‘Outlook - Sunny, Humidity - Normal, Wind - Strong’. In such case, the value of the root node is considered first. It follows the path based on the values of attributes finally leading to Result class named “Yes”.

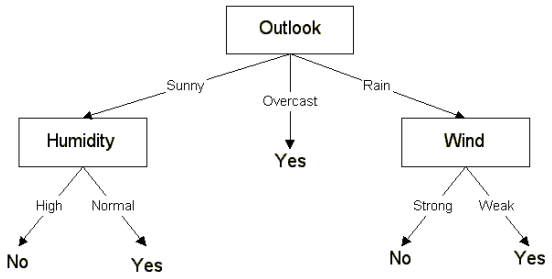

Image 2: Example of Decision Tree

Decision trees can be easily overfitted. Hence we use pruning to prevent it.

2. **Random Forest**: As an advancement over Decision trees, we used Random Forest after we calculated the accuracy of our decision tree model. The simple idea behind Random forest is to create N decision trees, each having T nodes, and randomly choosing S training samples. For creating condition at each node, we randomly choose a set of F features to calculate the decision at that node. Finally, the best decision tree amongst all is chosen (The tree is chosen by testing the N trees, and taking majority vote amongst all trees.)

3. **K-Nearest Neighbors**: K - nearest neighbor is used in pattern recognition. It is a non - parametric method used for classification and regression. In this algorithm, the training examples are multidimensional feature space with each result label. The principle behind the algorithm is to find K training samples closest in distance to the new point. The distance can be any metric measure, like Euclidean distance.

4. **AdaBoost:** Boosting is a general ensemble method that creates a strong classifier using different weak classifiers. **AdaBoost** was one such successful boosting algorithm which was developed for binary classification. It is used to boost the performance of **any** machine learning algorithm. The most suited and common algorithm used with AdaBoost is **decision trees** algorithm with one level. Since these trees are

short and only contain one decision for classification, they are often called decision stumps. Predictions are made by calculating the **weighted average** of the weak classifiers.

5. **Gradient Boost**: The basic principle of gradient boost classifier is same as that of adaboost, i.e., using the idea of using different weak classifiers and building a strong one Gradient boosting involves three elements: a) A loss function to be optimized (depends on the type of problem being solved) b) A weak learner to make predictions: (Decision trees are used as the weak learner in gradient boosting) c) An additive model to add weak learners to minimize the loss function (Trees are added one at a time, and existing trees in the model are not changed)
It can benefit from regularization methods that penalize various parts of the algorithm and generally improve the performance of the algorithm by reducing overfitting.

VI: CODE

An iPython notebooks has been uploaded to GitHub (Links below), which consists of the code build for using the above algorithm. We have used algorithms defined in scikit learn library for training our model. For calculating metrics, we have used cross validation, also called as rotational estimation. It breaks down a sample of data into subsets, performs analysis on one subset, and validating the analysis on other subsets.

Link to the entire code:
https://github.com/jayeshpatil1995/CS-6923-Machine-Learning

VII: RESULT

We used metric function defined in scikit learn for calculating the following metrics for each classifiers. The scale of the score range from 0 to 1.

	Decision Tree	Random Forest	K Nearest Neighbors	Gradient Boost
Accuracy	0.983	0.904	0.873	0.937
Precision	0.99	0.896	0.856	0.934
Recall	0.979	0.903	0.880	0.932
F1 Score	0.983	0.899	0.95	0.933
AUC Score	0.997	0.967	0.95	0.983

Table 1: A tabular form containing results obtained.

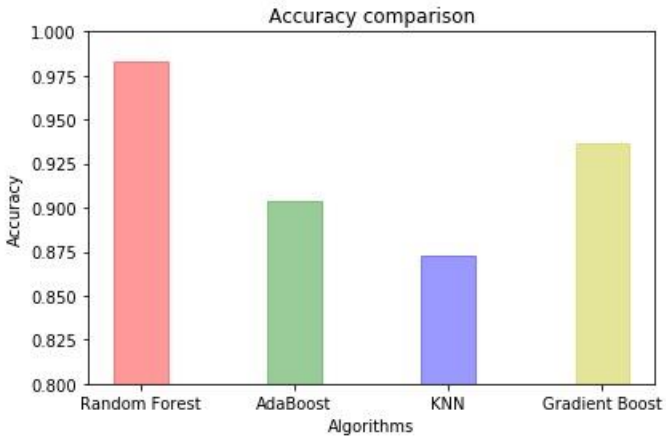

Image 3: Accuracy comparison between different algorithms

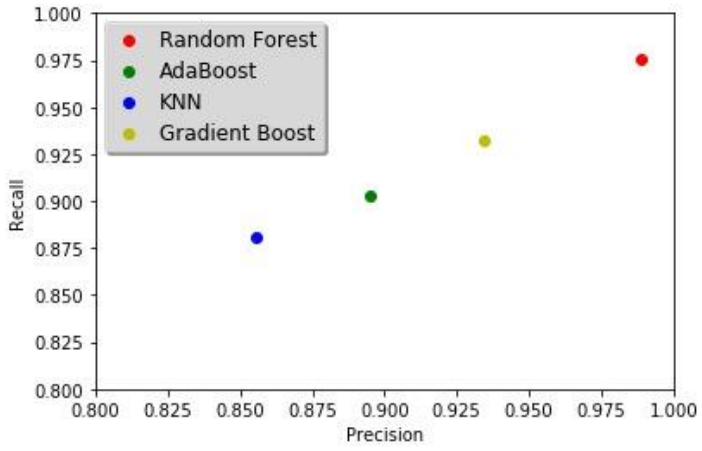

Image 4: Plotting Precision vs Recall

VIII: VIDEO LINK

https://youtu.be/1K8x9P672HY

IX: EVALUATION

Initial inspection of the code helped us to observe patterns such as ‘if the ‘name’, ‘screen_name’ or ‘description’ attributes of a twitter account contain the substring ‘bot’, then almost all of the times that twitter account is a bot’. Also, the attribute ‘verified’ was obviously trustworthy to know that the account is not a bot.

What we could have done more to increase the accuracy of our models would be the use of clustering algorithms to find interesting patterns so that we could have imposed more constraints and adjusted the threshold and feature importance while classification.

Also, we could have used a combination of different algorithms and based on their accuracy, we could have used appropriate weights and performed the classification for better prediction.

Also, we could have done natural language processing in order to understand the sentiment of the tweets of a particular twitter account and include that information in our classification.

X: CONCLUSION

As seen in the above diagram, Random Forest gives the least training error. It is evident that Random Forest is the best classifier to be used for the data set we had been provided. On the other hand, KNN proved to be the worst of the lot. AdaBoost and Gradient Boost performed well but not as good as Random Forest.