Milestone 3: EDA

Acknowledgements

Special thanks to the My Information Bubble (MIB) research group who freely shared the dataset used for their research paper, **The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race**, S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi. *WWW '17 Proceedings of the 26th International Conference on World Wide Web Companion, 963-972, 2017*

Introduction

Twitter is an Online Social Networking (OSN) platform where users can broadcast messages, images, and videos in 280 characters or less. Investigations have uncovered Twitter users who buy followers to give the "appearance of social influence…to bolster political activism, business endeavors or entertainment careers. [1]" In some cases, these types of accounts also engaged in nefarious activity like disinformation and promoting propaganda. Fake followers are just one type of bot account and some researchers [2] estimate bot accounts comprise between 9 and 15% of Twitter users. In response to pressure from Government and advertising customers to reduce this type of social media fraud, Twitter started a campaign in July 2018 that removed suspicious follower accounts, about 6% of all users. Given the massive number of bots and diversity of bots, machine learning techniques can augment human processes to identify malicious bots.

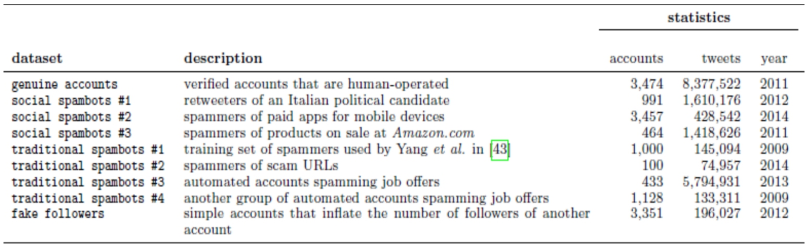
Questions

1. Can machine learning algorithms be applied to classify bot or human activity based on user-level and/or tweet-level data?

2. Can engineered features based on NLP techniques improve on traditional classification Test whether including additional engineering features such as NLP can boost bot prediction accuracy?

Data Cleaning and Extraction

The free Twitter API governs the number of records that can be queried by day per developer, and implies that only a sampled subset of all tweets will be returned for each query. To circumvent these limitations, we decided to use the Twitter dataset from MIB, hosted by Institute of Infomatics and Telmatics of the Italian National Research council. It has the following data collection, where for each dataset there is tweet-level and user-level data.



		statistics		
dataset	**description**	accounts	tweets	year
genuine accounts	verified accounts that are human-operated	3,474	8,377,522	2011
social spambots #1	retweeters of an Italian political candidate	991	1,610,176	2012
social spambots #2	spammers of paid apps for mobile devices	3,457	428,542	2014
social spambots #3	spammers of products on sale at *Amazon.com*	464	1,418,626	2011
traditional spambots #1	training set of spammers used by Yang *et al.* in [43]	1,000	145,094	2009
traditional spambots #2	spammers of scam URLs	100	74,957	2014
traditional spambots #3	automated accounts spamming job offers	433	5,794,931	2013
traditional spambots #4	another group of automated accounts spamming job offers	1,128	133,311	2009
fake followers	simple accounts that inflate the number of followers of another account	3,351	196,027	2012

Figure 1: MIB Data Collection

From this data collection, we randomly selected 200 users: 100 users each from both the genuine accounts and traditional spambots #1. Then for each of the users, we pulled their associated tweets. We made the decision to thin the dataset in this manner to support our ability to train the models in a reasonable time with limited compute resources.

Creating user-level data sample

We selected a smaller sample of 100 users and 100 bots who met the following conditions:
1. English tweeters to support NLP techniques.
2. Tweeted between 100 and 300 times.

We started with the tweets-level data first because we discovered that only a subset of users in the user-level summary data also had tweet-level data. One technical problem we encountered was processing the nearly 1GB tweets.csv file. We encountered out of memory errors when we tried to read the entire file into memory. Our workaround was to filter line by line. Even though this was a relatively clean dataset, we still encountered empty fields and 'NA' in the user_id column which we cleaned.

The processing of traditional bots was slightly different from processing human tweets because the language field in the traditional bot dataset was empty. We employed language detection modules to scan the name, location, and description column to detect if the user was an English Language Twitter users. This step was necessary to prepare the data for NLP technique application. Social spam bots are built to mimic human users, so we were able to use the human sample extraction code on these bots.

Initial Feature Engineering

We added our classification response variable, called 'user_type' to both the user-level and tweet-level datasets. The user_type was set as a Boolean field and indicated a human if 1 and a bot if 0. We planned to also use this field for stratified sampling when generating train/test/validation datasets.

EDA and Feature Selection for User-Level Data

User-level metadata summaries for each of the human and bot datasets were contained in a csv. Our sample dataset was comprised of about 50,000 human and 70,000 bot users. There were 26 different features ranging from geographical data, to metrics about the posting habits of the user. Results of feature analysis indicated that the most useful fields for our classification efforts are: lexical_diversity,followers_count,friends_count,default_profile

A heatmap was generated to visualize the sparsity of the dataset. This highlighted seven empty or all 0 features: 'is_translator', 'follow_request_sent', 'protected', 'verified', 'notifications', 'contributors_enabled'. Other fields appeared to be metadata regarding the data collection and were removed: id, utc_offset,timestamp,crawled_at,created_at.

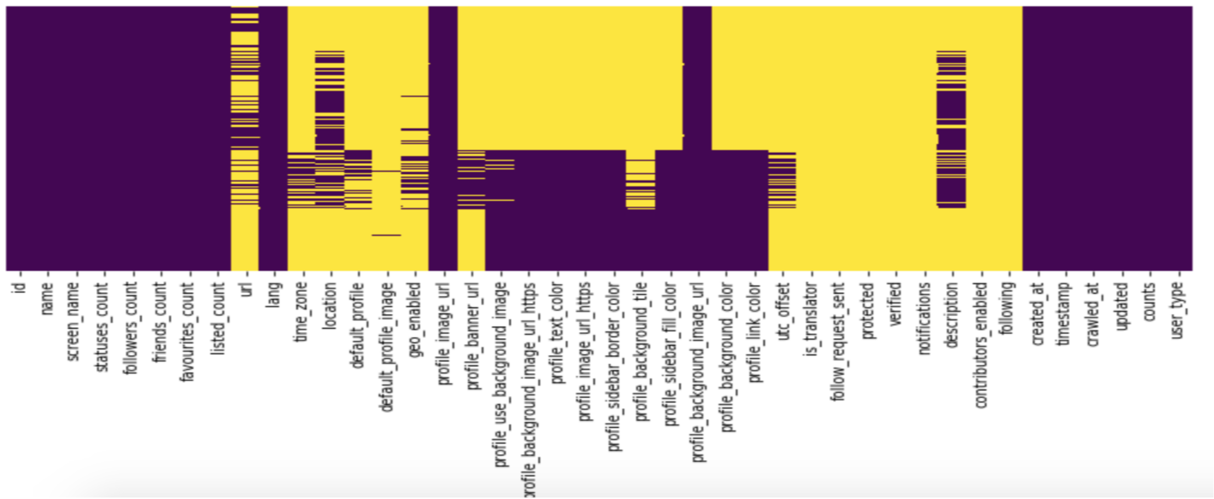
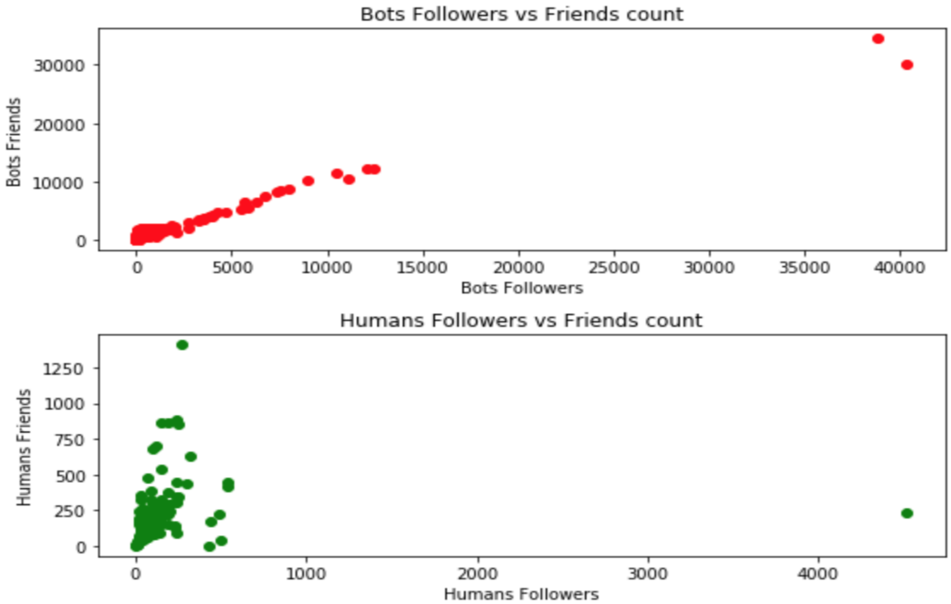
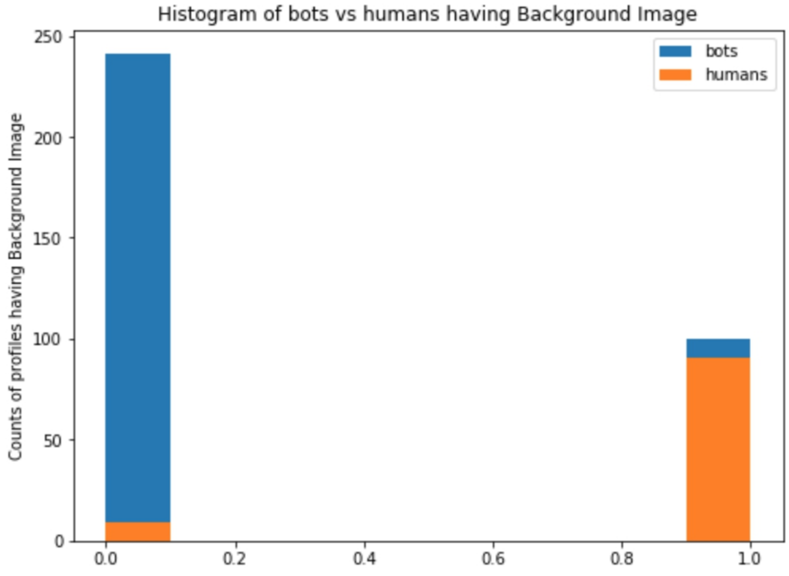


Figure 2: Heatmap of Feature Sparsity

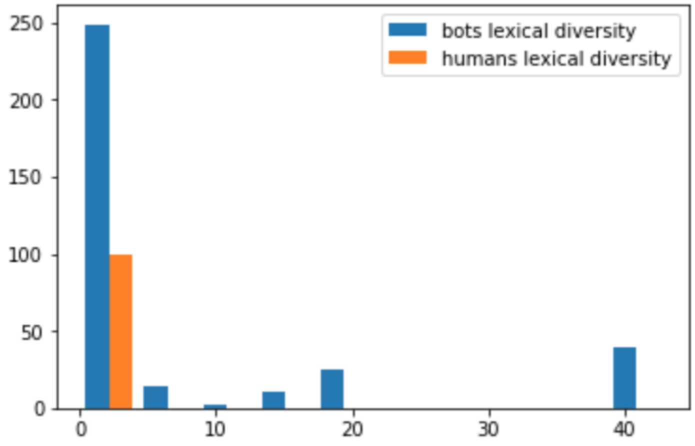
Correlation matrices for both bots and humans were generated to visualize linear correlation. An interesting observation occurred for both the friends_count and followers_count. In bots, there was a strong linear correlation between the two features, whereas this relationship did not exist in the human users.



Another interesting feature was background images. Visualized in the plot below, bots tended to have more background images than humans.



Lexical diversity was computed for each user based on their tweets. Lexical diversity consists of the number of unique tokens in the text divided by the total number of tokens. The histogram plot, below, implies that humans tend to have more lexical diversity within our sample.



EDA and Feature Selection for Tweet-Level Data

Resources

https://www.oreilly.com/library/view/mining-the-social/9781449368180/ch01.html