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.

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 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 a data collection, consisting of genuine accounts, social spambots and traditonal spambots where for each dataset there is tweet-level and user-level data. 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 Data Sample

We crafted a smaller sample of 100 users and 100 bots who were English tweeters (to support NLP techniques) and 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**, and **default_profile.**

A heatmap was generated to visualize the sparsity of the dataset. This highlighted seven empty or all 0 features (all yellow): '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.

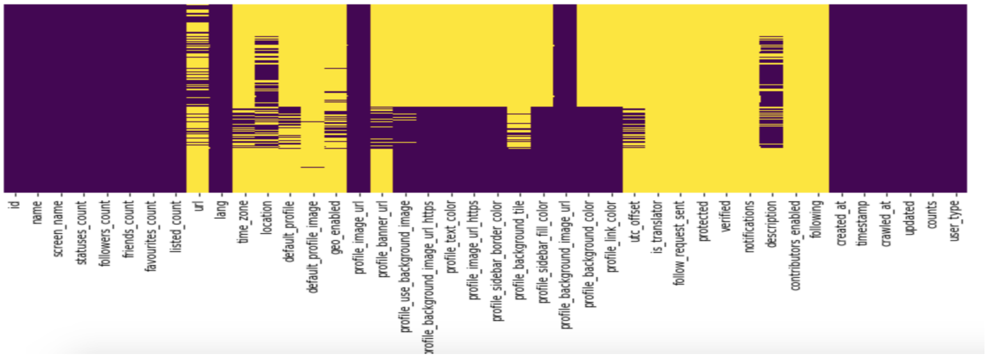

Figure 1: Heatmap of Feature Sparsity

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.

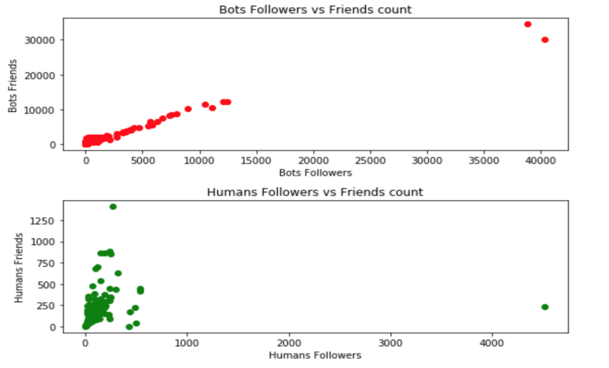

Figure 2: Friends versus Followers Plot

Another interesting feature was background images. Visualized in Figure 3, bots tended to have more background images than humans. Lexical diversity was an engineered field computed for each user based on their tweet text. 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.

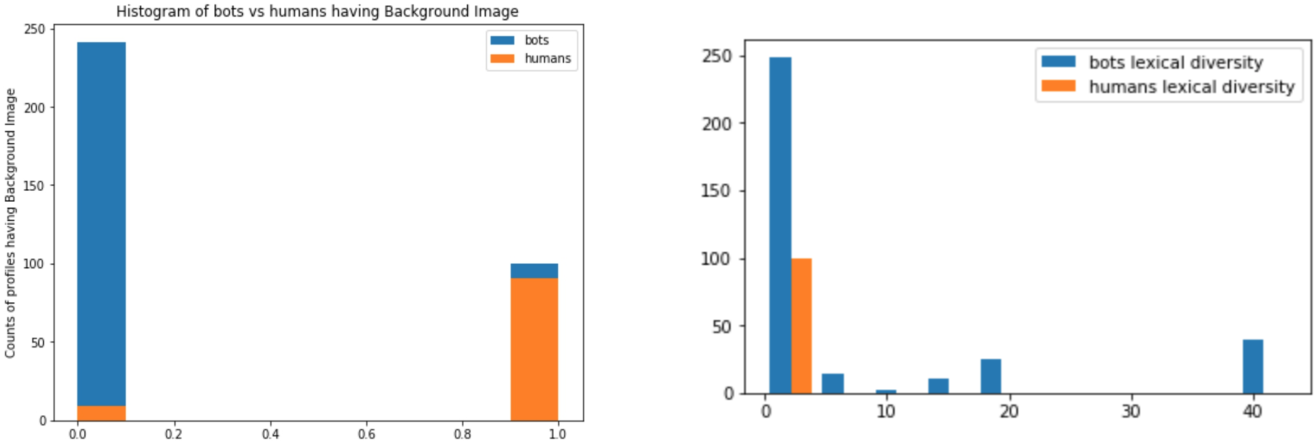

Figure 3: Counts of Background Image and Lexical Diversity Per User Type

EDA and Feature Selection for Tweet-Level Data

Results of analyzing features indicated that the most useful fields for our classification efforts are: **retweet count, favorite count, num hashtags, num urls, and num mentions and sentiment_negative**, an engineered field derived from sentiment analysis.This dataset was comprised of 50,000 human tweets and 70,000 bot tweets, with 26 features and one response. The tweet content is in a column called 'text' within the tweets.csv files. The 'user_type' is our response variable and contains a 1 if the account is a bot. Analyzing the .info() output shows us that five of the features are empty or have 0 for every value and we will drop these from the dataset: 'geo', 'contributors', 'favorited', 'possibly_sensitive', 'retweeted', 'reply_count'.

Since we are not performing any network analysis, we will also drop the 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', which are fields that can be used to reconstruct communications connectivity of an individual tweet. Other fields that we removed were id source, created_at, crawled_at, and updated. Since we are not performing time series analysis, we will also remove timestamp.

Correlation matrices were generated to identify linear relationships between features. Pairplots were generated for the remaining 14 features to identify non-linear correlation. Strong linear correlation was identified between the positive and neutral sentiment features. These sentiment features were generated by applying the sentiment analysis methods built in the textblob library to the tweet texts. This model, trained on labeled movie review data, classifies text as either neutral, positive or negative in sentiment.

Next, overlayed feature histograms were plotted to identify features that would help the models discriminate between the two classes. The following showed the strongest predictive value. The “retweet_count“ was almost entirely a human behavior. Notice any bot contributions are so small that they are not indicated in the plot. “Num_urls” showed strong profiles for bots in having 1 url whereas humans had a nearly even frequency of values.

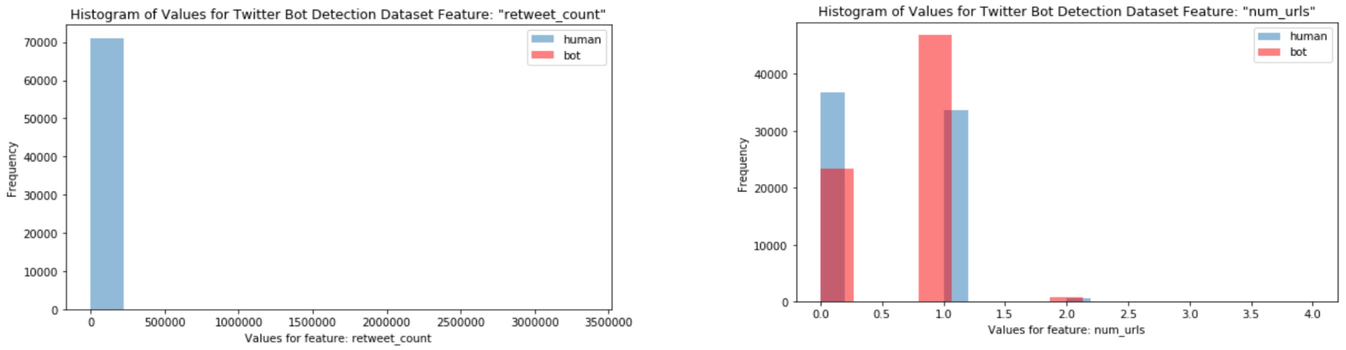

Figure 4: Retweet Count and Num Urls Histograms

Num_mentions and favorite_count had opposite predictive strengths, with a greater than zero favorite count being strongly human, and 0 num_mentions being strongly bot related.

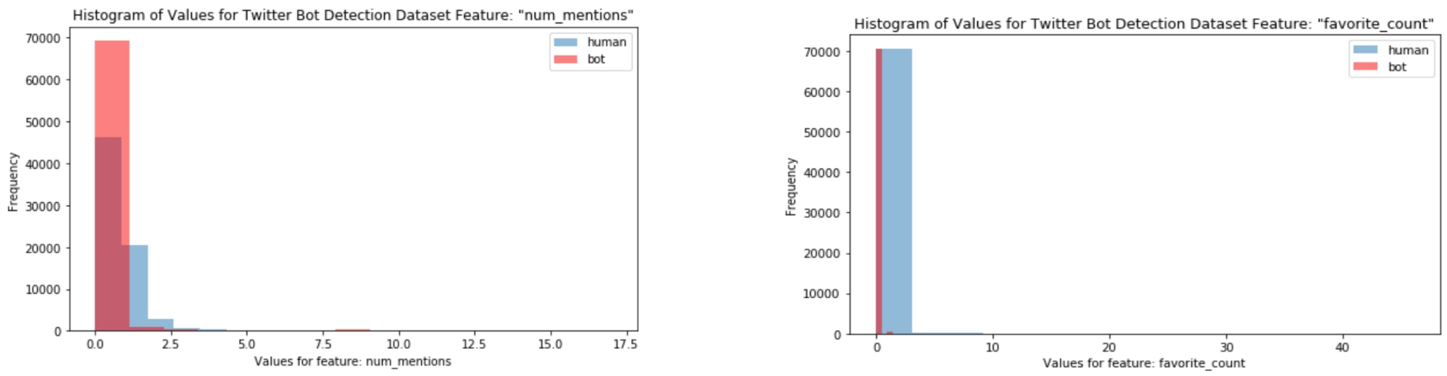

Figure 5: Num_mentions and Favorite Count Histograms

We also performed filtering based on low variance features by using sklearn's variance threshold method which resulted in the following features: retweet_count, favorite_count, num_hashtags, num_urls, num_mentions, sentiment_negative, sentiment_neutral, sentiment_positive. However, based on our feature analysis, we will leave out the collinear features (sentiment_positive and sentiment_neutral).