It has been estimated that up to 50% of the activity on Twitter comes from “bots” [1]: algorithmically-automated accounts created to advertise products, distribute spam, or sway public opinion. Detecting bots is necessary in order to identify bad actors in the “Twitter-verse” and protect genuine users from misinformation and malicious intents. Given a Twitter user’s profile and tweet history, our project is to build a binary classifier that identifies a given user as “bot” or “human.”

**Data**

For our training and dev datasets, we selected a subset of labeled datasets from the “Bot repository” hosted by Indiana University, consisting of user profile information and tweets for 5,424 accounts collected between 2009 and 2014 [2,3].

For our dev and test datasets, we utilized an annotated dataset generated by Varol et. al in 2016, consisting of 2,245 randomly sampled accounts hand-labeled [4]. Given these annotated user IDs, we downloaded the raw tweet and user content ourselves from the Twitter API.

**Features**

We have used 12 features, either immediately available from the Twitter API or from statistics that we derived:

- **User statistics:** Verification status, Followers count, Favorites count, Friends count, Friend / follower ratio
- **Tweet statistics:** No. mentions per tweet, No. hashtags per tweet, No. of urls per tweet, Retweet count per tweet, Favorite count per tweet, Unique tweet places count, Variance in tweet rate

**Models**

For all of our data-fitting procedures, we made use of the open-source ‘sci-kit-learn’ python library. We began with a simple logistic regression model that approximates the binary classification as a sigmoidal shape (Eq. 1), and fitted the parameters to our data via an L2-regularized coordinate descent algorithm. Next we used a gradient boosting tree classifier, producing an estimator (Eq. 2) from an ensemble of 100 sigmoidal decision stumps. Finally, we employed a neural network with two hidden layers of 3 and 4 nodes, respectively. (Eq. 3)

**Results**

For our application, the severity of type-1 error is greater than that of type-2; It is more offensive for a genuine user to be incorrectly labelled as a “bot” than the reverse. Therefore we have chosen to use an F$_P$=0.5 score (weighted harmonic mean of precision and recall) as our evaluation metric.

**Discussion**

While the NN had the best F-score at a threshold of 0.5, the GBM model had a greater AUPRC and better accuracy for its most confident positive predictions. For this reason, we would select the gradient boosting classifier as our best model.

We found that all of our models generalized well on our train-dev dataset, though their performance on the test distribution was not as good. There are several possible explanations for this discrepancy; the most likely is a distribution mismatch. We tested this hypothesis by adding gradually larger partitions from the dev dataset into our training set. As expected, model performance improved as more dev data was added (with an F-score > 0.95 with the GBM).

In addition, the task of identifying a given account as a human or a bot is hard even for humans, meaning that the “ground truth” for labeled datasets is questionable. We asked our own group of 3 volunteers to label 25 randomly sampled examples from our dev dataset and their average agreement with that “ground truth” was 79%.

**Future Work**

If we had another 6 months to work on this, we would:

- Explore using text-based features (e.g. tweet sentiment, topic extraction, words used) using an English-only dataset and utilizing NLP techniques
- Gather our own dataset, including friendship and follower relationships to identify bot communities (Bots tend to work in groups)
- Build a production-quality web plugin, providing real-time classification of accounts to Twitter users

**References**


