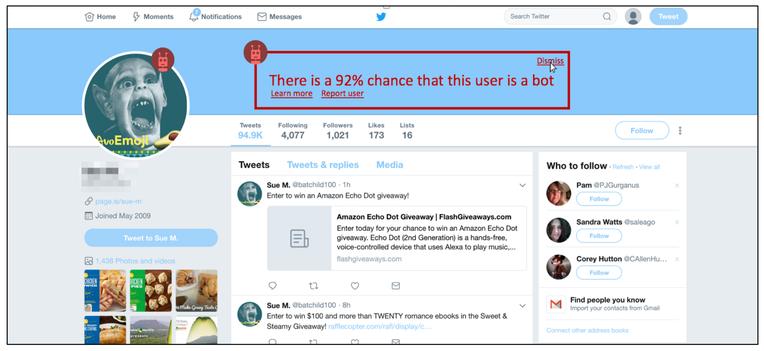
I spot a bot.

Sahil Nayyar (srnayyar@stanford.edu), Jessica Wetstone (wetstone@stanford.edu)

Goal and Motivation

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."

Figure 1: Mockup of a bot-classifier web plugin



Data

For our training and train-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 ID's, 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)

Eqn. 1: Log. Reg.

$$\hat{y} = \text{logistic}(w^T x + b)$$

Eqn. 2: Gradient Boost

$$\hat{y} = \text{logistic}\left(\sum_{k=1}^{K} w_k \phi_k(x)\right)$$

Eqn. 3: Neural Network

$$\begin{aligned} \hat{y} &= \text{sign}\left(W^{[3]} z^{[2]} + b^{[3]}\right) \\ a^{[2]} &= \text{relu}\left(W^{[2]} a^{[1]} + b^{[2]}\right) \\ a^{[1]} &= \text{relu}\left(W^{[1]} x + b^{[1]}\right) \end{aligned}$$

Results

For our application, the severity of type-I error is greater than that of type-2; It is more offensive for a genuine user to be incorrectly labeled as a "bot" than the reverse. Therefore we have chosen to use an $F_{\beta=0.5}$ score (weighted harmonic mean of precision and recall) as our evaluation metric.

Table 1: F scores

	Train		Cross Validation		Test	
	Acc.	F-score	Acc.	F-score	Acc.	F-score
LR	95.4%	0.960	96.0%	0.962	72.5%	0.547
GBM	99.9%	1.000	98.4%	0.989	75.1%	0.602
NN	97.6%	0.982	97.3%	0.977	77.7%	0.661

Fig. 2: Precision vs. Recall

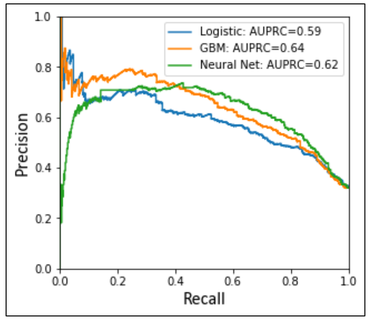
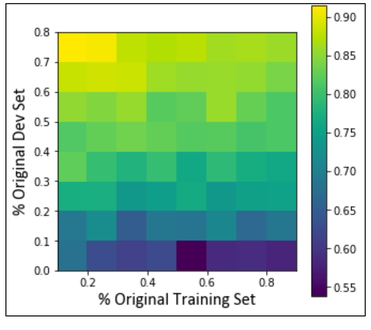


Fig. 3: Repartitioning Our Data



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

[1] Gorwa, Robert. "Twitter has a bot problem and Wikipedia might be the solution". Quartz Media, 2017. https://qz.com/1108092/twitter-has-a-serious-bot-problem-and-wikipedia-might-have-the-solution/

[2] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M.. Fame for sale: efficient detection of fake Twitter followers. *Decision Support Systems*, pp. 80, 56-71, 2015.

[3] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 963-972, ACM, Apr. 2017.

[4] Varol, Onur, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. "Online Human-Bot Interactions: Detection, Estimation, and Characterization." *ICWSM*, 2017.