MSAN 621 Group Project

Human or Robot?

by

Lincent Alberithm

with

Lin Chen, Soi Chong Ma (Albert), and Vincent Rideout

Introduction

Bidding in online auctions can be a fun and exciting way to shop. However, when predatory bots can swarm through every auction it ensures that a human bidder will never stumble upon a great deal, resulting in frustration for the auction site's legitimate users. To stop these robots from bidding they must first be identified, but how do you design a model to distinguish human bidders from bots? This is the challenge posed by a Facebook recruitment Kaggle competition from 2015 and accepted by our team.

Data

The data used for this competition is publically available at Kaggle:

https://www.kaggle.com/c/facebook-recruiting-iv-human-or-bot/data

This data consists of three files: the first two are train.csv and test.csv, which contain bidders information (payment information, address). The third is bid.csv, which has 7.6 million bids made by those bidders (type of merchandise, device type the bid originated from, time, IP address, country, and referring URL). There is a unique bidder_id field which can be used to link bidders to the bids they made. The target for our models was a binary indicator showing whether a bidder was a human or a bot.

The great challenge here was that virtually all of the data had been obfuscated to protect privacy. For example, there were 7351 different device types that people made bids from, but they could not be grouped by manufacturer or any other characteristic because they were simply labeled 'phone0000', 'phone0001', etc. The time of each bid had been transformed into large integers that were not recoverable into dates or times of day. However, they were still useful for comparing the relative magnitudes of time differentials between sets of bids.

We created features for our models based on bid volume, bid timing, and certain categorical variables. Our volume-based metrics were total number of bids for each bidder, along with the average and maximum number of bids each bidder made in a single auction. For timing, we calculated the minimum and average time elapsed between bids, both within a single auction and across all auctions a

bidder participated in. We also calculated the number of unique country and IP address each bidder's activities were traced to. Finally, we used a one-hot-encoding scheme to represent which category of merchandise a bidder was most engaged with.

This dataset was massively unbalanced - only 103 of the 2013 bidders in the training set were bots. While looking through the robot bidders by hand we noticed that a handful of them had very small number of bids. We recognize that it is an unorthodox practice, but we removed these observations from the training set. We believe our model will perform better with these observations removed because a sample with such a small number of bids cannot accurately portray the behavior of these bots. We developed our models with and without these observations, and the training, validation, and test scores were all better with them excluded.

Methods

Before fitting in any models, we modified further with our generated features by performing features selection. There are several feature selection approaches, including univariate selection, recursive feature elimination, principle component analysis (PCA), and feature importance (with coefficients or importance score). For this problem, we used the feature importance selection algorithm: Linear Support Vector Classification, penalized with L1 norm. Seven features were selected out of nineteen and they are:

- total number of bids for each bidder
- the average number of bids each bidder made in a single auction
- the maximum number of bids each bidder made in a single auction
- number of unique country the bidder's activities were traced to
- the number of unique IP address each bidder's activities were traced to
- average time elapsed between bids within a single auction
- average time elapsed between bids across all auctions a bidder participated in

One more thing to consider before fitting any models: because of the unbalance observations between human and robot, the Receiver Operating Characteristics (ROC) area under curve (AUC) score was used to determine model performance instead of accuracy rate. We trained several models with different classifiers: Adaptive Boost, Bagging, Decision Tree, K Nearest Neighbors (KNN), Linear Discriminant Analysis, Logistic Regression, Random Forest, and Support Vector Machines (SVM). We used K-fold (k=10) cross validation to compare the mean AUC score of each model. To achieve better performance, we used python package Grid Search (cross validation), loop, and manually adjustment to do the parameter selection for the models: i.e. criterion, maximum depth of the tree, prior vector, number of trees, learning rate, etc. We then selected the classifier with the best cross validation performance (based on AUC score) as the model.

Results

Cross validation was performed on each algorithm. Every cross validation was run with the same random seed and same number of K-fold to generate comparable models between algorithms. The probability of each bidder being a bot was calculated as our prediction instead of binary (1/0) prediction. As mentioned in the previous section, we used the AUC score as our metric to evaluate model performance. Table 1 next page summarizes the cross validation results on AUC score for each algorithm.

Algorithm	**ROC Area Under Curve**
Adaboost	0.9411
Bagging	0.8565
Decision Tree	0.8501
KNN	0.7816
Linear Discriminative Analysis	0.8041
Logistic Regression	0.7158
Random Forest	0.9437
Support Vector Machines	0.8954

Table 1: Cross Validation Summary

The results suggest that Random Forest and Adaptive Boost perform the best both having scores above 90%. SVM, Decision Tree and Bagging have fairly good scores above 80%. KNN and LDA have scores around 80% and Logistic Regression generates score of around 70%. These are good quantitative measure and in addition, we visualized the performance of some models by generating the ROC curve for those algorithms as shown in Figure 1 below.

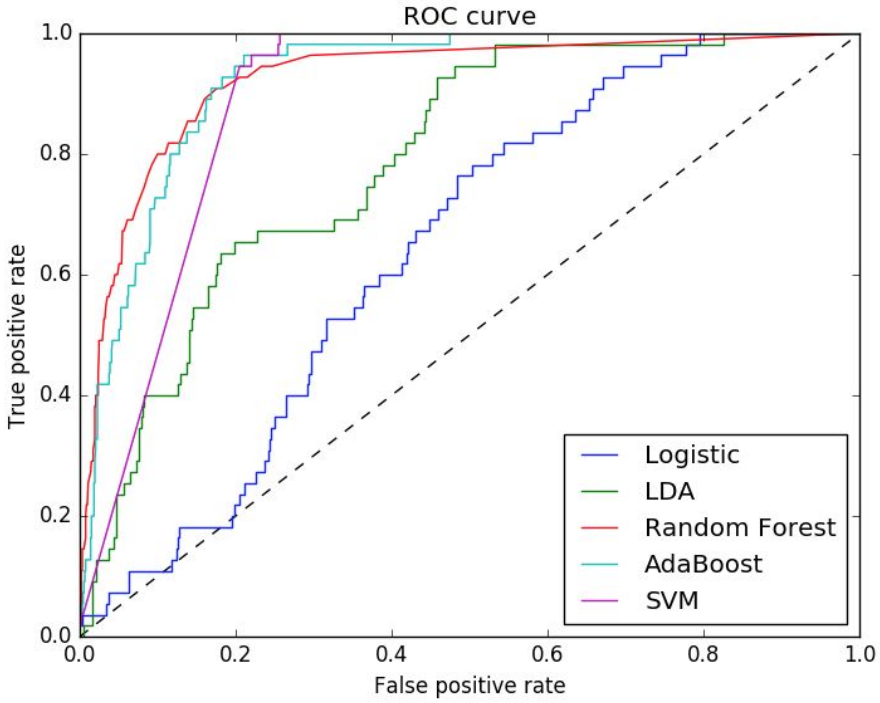

Figure 1. ROC Curve of Algorithms

The ROC plot shows that Random Forest and Adaptive Boost have the largest area under curve (closest to 1) indicating these two models perform the best on identify potential bots. SVM came in having the second largest area followed by LDA and Logistic Regression. The plot gives us a visual alternative to evaluate model performance. Based on both numerical and graphical results, we decided to employ Random Forest to be our final model.

Random forest provides many advantages (and thus performed the best) to this problem such as high dimensionality and having a potential non linear decision boundary between human and bot. Multiple cross validations were performed on Random Forest by using different number of trees ranging from 10 to 600. Different AUC scores were calculated and plotted against number of trees. Figure 2 below shows that the optimal score appeared around 280.

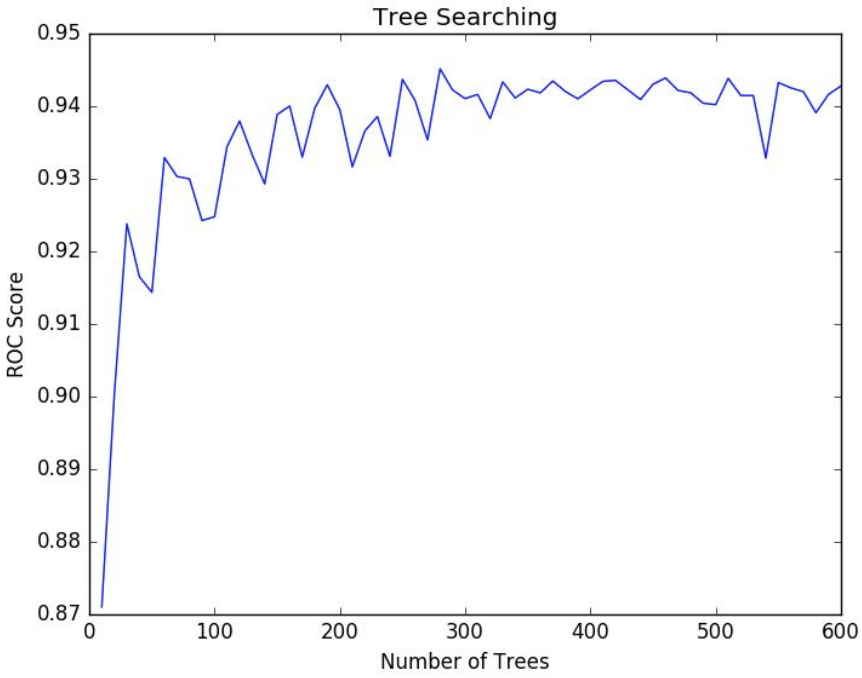

Figure 2. ROC Score vs. Number of Trees

With the optimized model, we generated the prediction of probability that a bidder would be a robot using the test set provided by Kaggle. The test set does not contain outcome so the only way to evaluate our model performance is through the Kaggle challenge website. We uploaded our prediction and received an AUC score of 0.9207. We were ranked the 228th place among 985 submission. Compared to the first place's AUC score of 0.9425, we are only off by around 2% accuracy. The Kaggle test result suggests that our model performed fairly well.

Conclusion

We started with raw data and limited features which provided us a good chance to tackle the problem from scratch. We had the opportunity to brainstorm on how to utilize the limited features and we were able to build useful features for our model. We also learned the mechanisms of different classifiers, which helped us understand the advantages and disadvantages of different algorithms when solving different problems.

Furthermore, we have a better understanding of different metrics on classification problems. For instance, with Logistic Regression, the model has an accuracy rate over 90% but it could only identify two bots out of around one hundred bots from the train set. But with the metric of ROC, it could only achieve 70% AUC score, which reveals the true capability of Logistic Regression in this problem.

Lastly, more possible features can be developed and added to the model. Potential works include assigning cluster groups to bidders prior of classification or finding maximum number of bids in a time frame. Reasonable features addition can substantially improve model accuracy and in the future.