Information Retrieval and Extraction Major Project

Reverse Auctioning Engine

Team number : 33
Mentor : Anurag Tyagi

Karusala Sri Vaishnavi (201301230)
Sane Sushmitha Reddy (201225012)
Abid Ali (201505595)

Abstract

Online auctions have become an increasingly important aspect of e-commerce. However, for consumers, the chances of landing a winning bid in online auctions has become increasingly more difficult with the abundance of “bidding robots”. Bidding robots such as “BidRobot” and “Auction Sniper” are pieces of software that are configured by the user to follow any number of auctions on different auction sites simultaneously, bidding in place of the user according to predefined settings and preferences. Humans are not able to attend to and monitor auctions with the same capacity as a computer, which can make complex bidding decisions in split-second time and can follow an auction with nonstop, undivided attention. Thus bidding robots gain a significant competitive advantage over their human counterparts, posing an interesting problem for auction sites that wish to level the playing field by identifying and banning bid bots. The project aims to identify auctioning agents that are software robots and thus eliminate them from the online auctions. Such robots have an undue advantage over human agents and their inability to win against robots may lead to plummeting core customer crowd .

Project Scope

On a broad level the project aims to analyse the bids data , extract distinguishable features that help us differentiate between a human and robot, train the classifiers according to these features and finally test the accuracy of our classifier using test data. The scope can be divided into 3 levels –

1.Feature Extraction .
2.Modelling and training.
3.Testing.

Proposed System

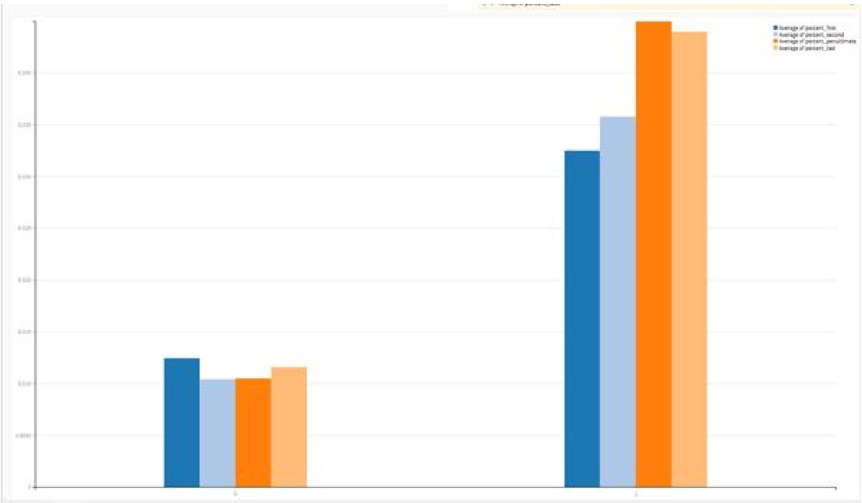
The main challenges in this project are **feature engineering** and **model optimisation**. In many ways, feature design for bot detection is very similar to what needs to be done for fraud detection.

1. Feature Engineering

From a physical point of view :

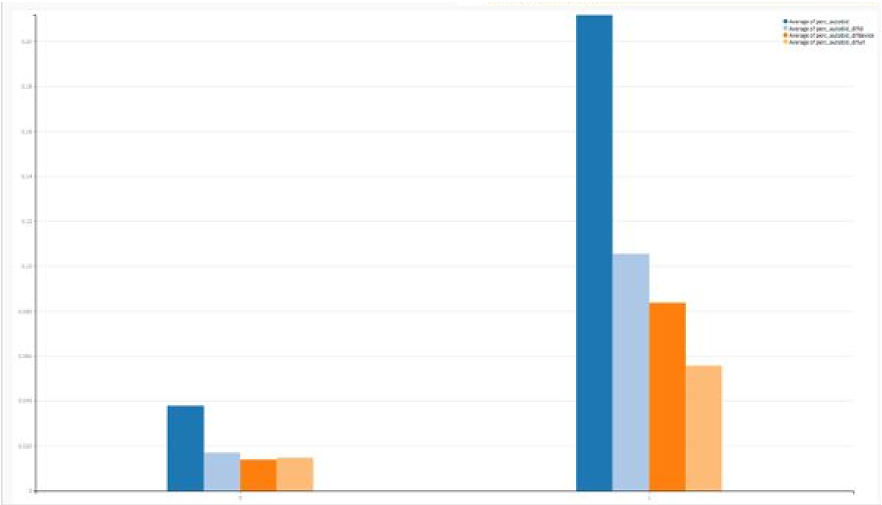
- ➔ A robot is able to take part in more auctions than a human. Hence, the first feature is simply the total number of auctions.
- ➔ A human will use only a small number of devices and IP address whereas a robot can use many more. So creating features on the number of distinct devices, country, IP and URL used by a bidder. Here ,we also used counts on combinations .But,counts increase with global use so we need to rescale them. By counts per auction and then calculate quantiles / aggregates for each bidder.
- ➔ Time obfuscation : The goals of why a robot wants to enter an auction have to be considered while feature engineering. i.e to first win the auction and the second to make the price artificially higher.
- ➔ Also , by intuition a robot would be more likely to be the first to bid on an auction or to bid after the first person has entered. So all this boils down to the following features:
  - % of time first, second, before-last, and last in auction,
  - percentage of bids on one's self
  - and percentage of bids when another bid was done.

The graph below depicts that the elements we used for feature extraction that were pretty right.



On average, humans (coded by 0) are way less likely than robots to start an auction or to finish one (rescaled by the number of auctions they entered).

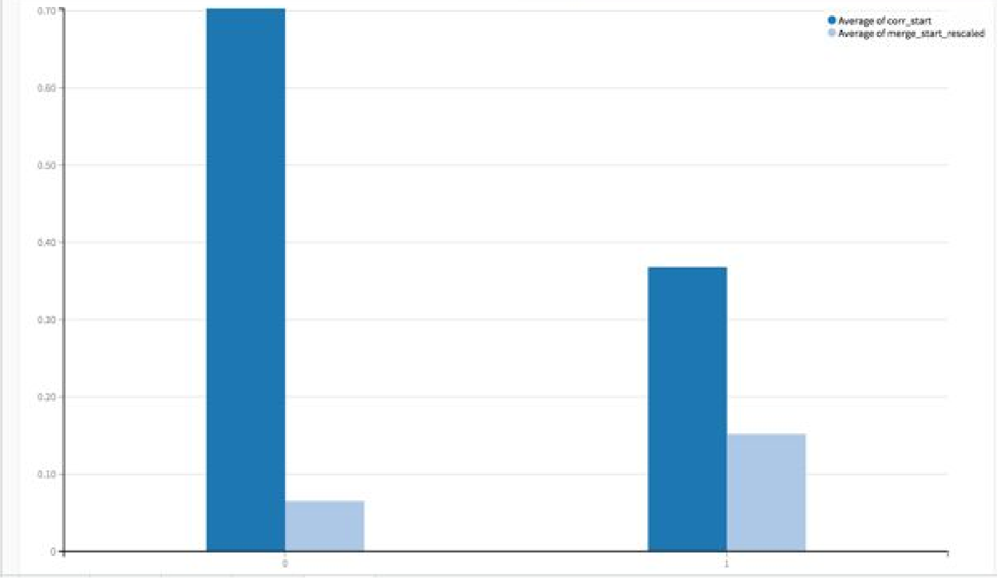
In the same way, if we look at the percentage of bids of a bidder that are bids on him or herself (and the percentage of bids that are auto bids and have a change in IP, URL, or device) we get the following graph:



We can also create features similar to counts of distinct devices, IP, etc. as a human to have several IPs for several auctions and then another one, etc. He will probably not be changing for every bid. So a feature could be the number / percent of times there is a change in the IP.

- ➔ Randomness observed among human and robots are completely different. If a bidder is a robot then it will perform almost perfect randomness as compared to a human bidder.
  - Give ranks to the auction sequence for a user based on the starting times or the finishing time and lets called it as an ordered sequence.
  - get the entire sequence of the auction for that user in the order of bid it make.
  - Compare the above sequence with the ordered sequence as sequence correlation or number of inversions to sort.
  - For humans the number of inversions will be less as compared to a robot and correlation will be high.

The following graph shows the plot for correlation in the sequence(also the number of inversions) for human and robot.



- ➔ Number of auction that are active for a particular bidder for which there is still time left for the auction to end. If the bidder is human then he will not be as active as a robot.

- ➔ Time variable:
  - Number of time a bidder made several bids at the same time. For humans it will measure less as compared to a robot.
  - If a bidder makes a bid every x time units then it is a robot.

Some of the basic features:

- Fraction of auctions won by member (out of total number of auctions member participated in).
- Mean fraction of bids made by member (in auctions member participated in).
- Max, Mean and Standard Deviation of number of auctions member is active in per hour.
- Probability of robot given Country, Merchandise search category, and device.

We give more importance to distinguishable features and less to other features (other features are helpful at boundary conditions and also to increase accuracy) .

2. Classifier Model selection and optimization

→ **Random-forests** : It is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, because they have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance of the final model.

→ **Gradient-boosting :** It is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

→ **AdaBoost** : It is a machine learning meta-algorithm . It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing (e.g., their error rate is smaller than 0.5 for binary classification), the final model can be proven to converge to a strong learner

→ **Bagging-Classifier :** Bagging is a bootstrap ensemble method that creates individuals for its ensemble by training each classifier on a random redistribution of the training set. Each classifier's training set is generated by randomly drawing, with replacement, *N* examples - where *N* is the size of the original training set; many of the original examples may be repeated in the resulting training set while others may be left out. Each individual classifier in the ensemble is generated with a different random sampling of the training set.

→ **Logistic Regression :** The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). As such it is not a classification method. It could be called a qualitative response/discrete choice model **.**

We can combine these similar models and average their output for our final result to prevent overfitting.

Tools expected to be used :

1. Python.
2. sklearn.
3. Pandas.
4. xgboost

Dataset

The dataset is organized into three files: train.csv (bidder data with labels), test.csv (bidder data without the labels meant to be tested), and bids.csv (individual bid data from 7.6 million bids). For the purpose of this project, we will be using the labelled bidder data and bid data to make and test our predictions.
http://www.kaggle.com/c/facebook-recruiting-iv-human-or-bot/data

Related Work

1. Bid-war : Human or Robot .
2. https://www.kaggle.com/c/facebook-recruiting-iv-human-or-bot