# Human or Robot?

Robert Recatto
A10741982
University of California, San Diego
9500 Gilman Dr.
La Jolla CA, 92093
rrecatto@ucsd.edu

## INTRODUCTION:

With advancements in technology happening every day and Artificial Intelligence becoming more integrated into everyday society the line between human intelligence and computer intelligence continually becomes slimmer and slimmer. Many argue that this is good for society, as we can automate some menial tasks thus saving people time and money. There are, however, also consequences to these technological advancements as well.

People have begun using these robots to get advantages over others as it becomes harder and harder to distinguish between human and robot behavior. One auction site in particular has been having issues with robots as their human customers feel like the robots have an unfair advantage due to the faster reaction speeds and processing speeds. To combat this they have set up systems in order to try to differentiate their robot from their human customers.

Differentiation has proven a problem for this particular site, though, as robot behavior almost identically mimics the behavior of humans. All of their predictors proved not accurate enough so they went to kaggle with the issue.

They uploaded a good amount of data on kaggle for other programmers to utilize and work on with the intentions that someone would create a more accurate predictor and help solve their issues of Human or Robot.

## 1. THE DATA SET

Luckily for me, the website was able to compile a very large and comprehensive data set in order to make the most accurate predictor possible.

The data set consists of three different subsets:

1. train.csv
This is the training set of bidders, it has four sections:
(i). bidder_id: the unique identifier of the bidder
(ii).payment_account: payment account associated with a bidder.
(iii). address: mailing address of a bidder.
(iv). outcome: 1 indicates Robot, 0 indicates human.

2. test.csv
This is the test set of bidders, it has three sections:
(i). bidder_id: the unique identifier of the bidder
(ii).payment_account: payment account associated with a bidder.
(iii). address: mailing address of a bidder.

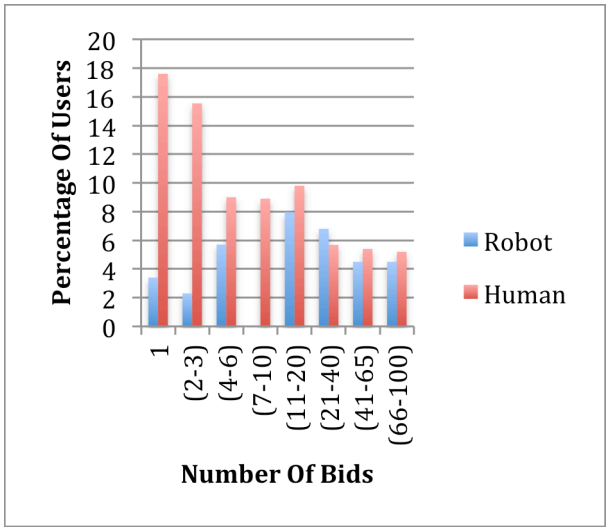
2.bids.csv
This is the set of bids, it has 9 sections:
(i). bid_id: unique id for the bid
(ii). bidder_id: unique identifier of a bidder.
(iii). auction: unique identifier of an auction.
(iv).merchandise: category of auction site campaign
(v). device: phone model of a visitor
(vi). time: time that the bid is made.
(vii). country: the country IP belongs to
(viii). IP: IP address of bidder
(ix). url: url where bidder was referred from.

The length of the training data set and testing data set are only 2014 and 4701 respectively. There are over 7.9 million bids provided, however due to computer strength I was only able to work with a sample size of 2.55 million.

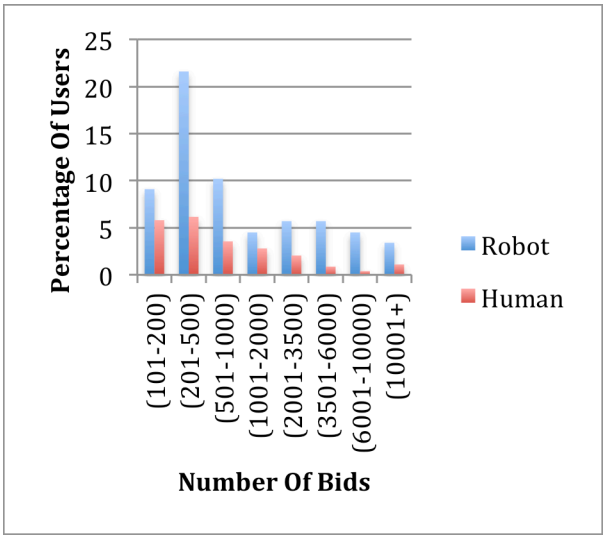
Before making any models I began analyzing the data set to help me decide which features to include in my predictor.

The first thing I did was count the number of human bidders among the data set compared to the number of robot bidders in the training data set. There is 1910 human bidders compared to just 104 robot bidders. Also, of these 104 robot bidders and 1910 human bidders, only 88 robots and 1267 humans were actually usable because they are the only ones with bids in my smaller version of the bids dataset. Due to how small the size of the robot bidder dataset was I plan on making any models predicting for robots to be as basic as possible as I do not want to overfit to such a small data set. I then took the average amount of bids executed by robots against average amount of bids executed by humans. Shockingly, human bidders only averaged around 671 bids per user while robot users averaged over 1684 bids per user. Due to the very inconsistent betting patterns of each as well the variance and standard deviation of the bid amount data set was extremely high as well with variances of 17614486.29 and 41925121.89 and standard deviations of 4196.96 and 6474.96.56 for robots and humans, respectively.

Here is the distribution for all users with under 100 bids: (Graph1)




Here is the distribution for all users with over 100 bids: (Graph2)

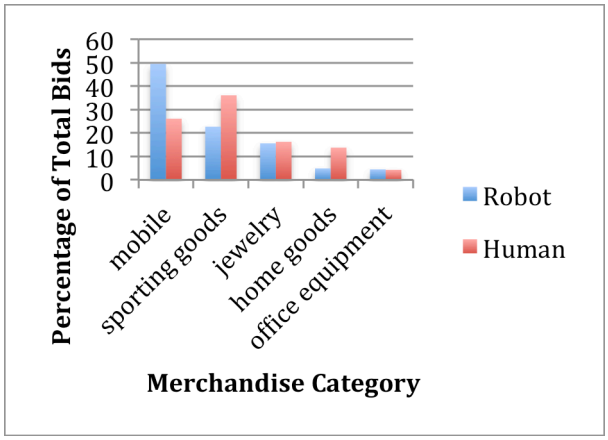



Despite the data being pretty dispersed in terms of bid amounts for robots and humans there is a clear distinction that average robots bid a lot more than their human counterparts. Despite the average human bid amount be 671 bids, the vast majority of human bidders bid under 100 times. Human average is catapulted by the high volume outliers.

While placing robot and human users into their respective tiers I also put stored the "harder" to identify robots and users (users with over 100 bids and robots with under 100 bids) into separate data structures to find more specific correlations for this "harder" data. I did this in hopes of training a PCA style model and testing to see if that improves efficiency versus a model that weighs all data equally.

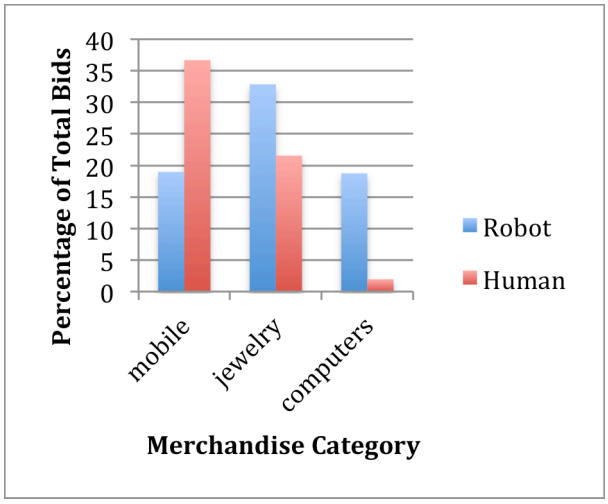
I used this new "harder" data set in conjunction with the full data set in order to not only try to find out about the data set as a whole, but also about specific pieces of the data set. After testing both sets on the fields "country", "device" and "time" I was not able to detect any correlations as robots and countries are spread fairly evenly between both fields. Similar results occurred for both the "harder" and full data sets when I searched if robots tend to bid on the same auctions more often than humans. The data revealed that humans bid on the same auction an average of 18.4 times while robots were 17.1. Since I want to make my model as simple as possible, none of these fields will be features in my model.

When using the merchandise category, however, I was able to come up with useful results for both the full set and regular set. For the full set of data, both humans and robots had the same top 5 merchandise categories, however the bidding patterns varied quite significantly: (Graph3)




As you can see, robots bid on mobile goods about twice as often percentage wise as humans.

A different yet equally compelling result can be found by running the same test only bidders with under 100 bids: (Graph4)




This data is an extremely useful set, first of all because it is the "difficult" robots vs. the "easy" humans. Finding how to distinguish robots with lower bid counts is extremely crucial to make a good predictor for this particular problem. So finding these key differences in users with under 100 bids is very important to make sure to add to a good model for this problem.

As for the last fields not yet discussed, "ip" and "url", both were obfuscated as to protect the integrity of the user. Because of this, I did not see how they could be too useful as without a decoder it would be hard to detect relationships between the two fields and a bidders status of human or robot.

# 2. PREDICTIVE TASK

As stated previously, the overall goal of this assignment is to create an accurate predictor that decides whether a bidder is a human or a robot.

I started by gathering all good data from the data set that I deemed useful enough to base my model off of. This includes average bid amount, total bid amount, those classified as "hard" to identify, and common merchandise bid on by both the data set in its entirety and the "harder" data set.

To test the data, I split the training set into two parts, one for training and one for testing as the submissions on kaggle were closed and thus the provided test file was unusable.

## (i). BASELINE

Since the amount of users is very predominantly human users, the obvious first baseline is to just classify all users as human to really good starting percentage accuracy for a baseline predictor. This basic predictor gives an accuracy of 94.2% as only 5.8% of all users in the test set are classified as robots. This, however, does not fully satisfy our goals of successfully identifying users as 0 robots are identified so we venture to improve our predictor using past compiled data.

## (ii). BASIC IMPROVEMENTS

The first obvious basic improvement that comes to mind is by classifying more active bidders as robots. This is likely to improve the ratio only if the right coefficient is used in deciding what makes a bidder "active" enough to be a robot.

A good starting strategy to pick this coefficient is by using basic probability. Since we know that the baseline is already 94.2% accurate, we want to only classify users in the top 5.8% of robot likelihood as robots. To do this, we should classify as robots users that fall within any of the tiers identified previously in which

(# of robots in tier)/(total users in tier) > .942. The closest two tiers to this requirement were tiers 14 and 15 with ratios of .877 and .92, respectively. Upon classifying users in said ranges of bids, the overall accuracy of the classifier decreased for both cases.

After failing at improving the baseline predictor through using clusters I tried to improve it by only classifying the users with the highest amount of bids as robots. I tried this in two different ways, firstly by putting the top 5.8% most frequent bidders into a set and classifying any of those bidders as robots if they come up. Also I tried by putting the users that accounted for the top 5.8% of all votes into a set and classifying any of them as robots if they come up. Unfortunately, though, both resulted in a decrease in accuracy compared to the baseline.

Ultimately, since the baseline is so accurate, finding any basic improvements is extremely difficult as any changes made to the baseline need to be made extremely selectively.

## (iii). BASIC MODEL

At this point I began considering other ways to approach the problem. For this particular task a SVM approach would not make sense as it focuses on the "difficult" to decide options in training the model, but the "easy" robots were even too hard to identify.

Logistic Regression seemed like the best choice for training so I began to construct a very basic model. Of all the possible features explored in the data analysis portion of the report, the largest correlation seemed to exist in the amount of bids per bidder. So I constructed a model with just one feature – number of bids. I used this model and performed logistic regression with a changing boundary line in order to try to find an optimal amount to accept and help improve my model, but to no avail. There was no possible boundary that could possibly improve the baseline accuracy for this particular model.

The data was still too split when training with the dataset in its fullest so I decided to split the dataset in order to better find correlations. Doing so can help better understand particular subsets of the data. When data is as seemingly random as this, it can prove very effective in raising accuracy.

## (iv). REVERSE PCA

Normally, Principal Component Analysis is used to maximize randomness of a particular data set and to make sure that components that give most information are kept. However, for this particular data set, the "randomness" seems to overwhelm any possible correlations thus making even easy to identify robots unrecognizable.

To combat this problem I began training a model based solely off of the "easier" to identify robots. Like before, I split the training set into two pieces, the so called "easy" piece of users with over 6,000 bids; and the "hard" piece of users with under that. Focusing on the "easy" piece I started with a model with one numeric feature (number of bids) and four Boolean features (bid on mobile, bid on computer, bid on sporting goods, bid on jewelry). These categories in particular were chosen because they demonstrated the largest discrepancies between human and robot users in the training set. Getting the information necessary for the features was simple as number of bids was just the count of how many times a username came up in the bid dataset and the four boolean features are one if any of said bids were on an item of the corresponding category.

I also changed my strategy for classifying as robot or human as well. For this particular problem, falsely classifying any user is extremely detrimental due to the high accuracy, as you must correctly identify nearly 19 users for every 1 misclassification. To combat this problem, I fit my predictor to classify as robot the greatest theta values of the "easy" test set in order until a misclassification occurs. That way I minimize misclassification error and maximize percentage correctly identified.

Using the above strategies, the five dimensional model described earlier was able to correctly identify two consecutive robots, increasing the percentage accuracy from baseline to 94.4%. Although just two in a row seems slightly insignificant, be reminded that if you were to pick any two users at random, the chance of picking two robots is just over .3%.

Overall, because of the limitation of the dataset, this was about as complex as a model should get without too much over-fitting. With only 1000 bidders to fit to, over-fitting can very easily occur.

# 3. LITERATURE

## (i). KAGGLE

The data as a whole comes from a competition on kaggle in which there are over 1000 people are competing to produce the most accurate robot classifier. It should be noted, however that for the competition on kaggle, users are allowed to post their predicted probability that a user is a robot, contrary to the binary approach of predicting that I took.

The kaggle competition was judged based off of area under the ROC curve. This setup, differs slightly from my judging criterion though as area under the ROC curve is equal to the probability will rank a randomly chosen robot greater than a randomly chosen human.

There are some differences between my data and the data used by users in the competition as well: I was only able to use 2.5 million of the 7.9 million bids in the database and I was not able to use the provided testing data set so I split my training set in half and tested on the second half of the set.

In terms of comparative performance between models it is hard to accurately compare performance relative to tasks as goals of my predictor compared to that of predictors on kaggle are very different. I was only able to correctly identify 2 robots however I was 100% accurate in identifying these robots. The leader on the leaderboard of kaggle has an area under the ROC curve of ~.94. My particular model would only have an area under the ROC curve of 2/(# of robots), which would surely be much lower than .94. This goes to show that predictors used on the same data set are not very portable in completing tasks other than the task it is specifically designed to predict.

## (ii). HISTORY

This idea of identifying robot users compared to human users is an idea that reaches far beyond just this kaggle competition, though.

This theme extends all the way to the 1950's originating with the Turing Test. The Turing Test describes a test in which a panel interacts with two separate entities, one human and one robot. If the panel is not able to determine with over 50% which entity is human and which entity is robot, then the robot is deemed "intelligent." This test was letter adapted for modern use into the Loebner Prize with the same goal but with only one entity being judged at a time.

Whereas I identified robots by correlations in bidding patterns, though, the Turing Test and Loebner Prize focus on a Natural Language Processing, NLP, approach to identifying. Natural Language Processing is the idea of computers understanding language and giving responses that make sense to a human counterpart. This process is extremely difficult to do effectively, however, as there are only a few programs in existence that are able to do so with any success including Joseph Weizenbaum's ELIZA and the most recent Loebner Prize winner, A.L.I.C.E. However, due to the difficulty of Natural Language Processing, not robot in history has been deemed "intelligent" by Turing's standards by fooling over 50% of judges.[1]

# 4. CONCLUSIONS

Unfortunately, I was not able to create a model as successful as either of my counterparts described above.

For the case of creating a test as accurate as the Turing Test or Loebner Prize, it is not fair to compare because of the vast difference in tasks of the robots. For the case of the Turing Test and Loebner Prize, robots must be very good at NLP, a process much more advanced than simply mimicking human bidding patterns. Therefore any difference in accuracies are more likely due to the complexities of the robots in question, not the accuracy of the test.

More can be learned by comparing my model to that of those on the leaderboards of kaggle, to which my model fell short. These shortcomings can be attributed to multiple factors. Firstly, I had much less data to work with than my kaggle competitors as my trainings set was ½ the size of theirs and my testing set was ¼ the size of theirs. Also, I was only able to use about ½ of the total bids from the bid database. This restricts me not only from an information standpoint but also by limiting the complexity of my model due to high possibilities of overfitting. Had I had more data to

work with I would have been able to make a more complex model without worrying about overfitting and thus would have been able to make more accurate predictions. It is naïve, however, to assume that this is the only contributor to my shortcomings though as there was also some clear flaws in my model which likely kept me from being able to compete with my kaggle counterparts.

Let me first start by thoroughly describing my model in its final state. Theta had 6 dimensions overall, 5 of which were features. Theta[0] represented the base if all features are 0. Theta[1] describes how robot rating increases per bid cast by the bidder. Theta[2]-[5] were all binary features, meaning if a bidder bid on any item of the categories represented by Theta[2]-[5] then the value of theta at that index was added to the baseline. Due to high amounts of variance among the train set in its entirety I was forced to train my model on a subset of the data with a lower variance. (Doing this limited possible complexity of my model even more so but it had to be done to find any relevant correlations). After doing this I got the theta values of all the data in the test set that would also qualify to be in the particular subset that my model was trained on, and ran it through a sigmoid function in order to get probability. Finally, I took the bidders with the highest "robot probabilities" to classify as robots.

Some things I did right about my model were finding a good subset of the training set in order to limit variance as much as possible. This helped me get clear cut correlations and data that was not as muddled as when I ran my model on the data set in its entirety. When the standard deviation of a data set is over twice the size of the average as was the case for this particular dataset, it is going to be very difficult meaningful connections in the data. I think I also chose my features, although few, well. Just bid amount alone was not enough as when I tried that model it was a lot more inaccurate. However, adding category does not really double count because a bid doesn't necessarily mean that a user bid on a specific item, and it increased the accuracy to reflect the fact that robots that bid in bulk bid on 'mobile' goods about twice as frequently percentage-wise as humans.

There is also a good amount I could have improved about my model as well though. Firstly, I could have found more subsets of the data with low variance other than just for that of bid count. Although finding these subsets is somewhat difficult it helps a lot to find correlations in subsets because then you can use multiple models on the data in it's entirety and send entries to certain model's depending on if it fits the qualifications of the subset. This helps add accuracy without overfitting on any one particular subset of data. Also, I could have utilized some of the fields such as country, phone and time a little bit better in my model. I initially chose not to use them at all because there were no obvious correlations between and of the fields and whether or not a user is a robot. However, had I dug deeper and tried to find more specified correlations or possibly even combining them to make binary constrains, (for example, probability is a robot given from x country AND using y device), then I'm sure I could have eventually found a way to scrape for information from these fields as well. Finally, if I had more data I could have also used the "harder" robot to identify correlations I discovered in the data analysis portion. Unfortunately though, there was less than 20 robots in the training set to fit to any model, so it was impossible to make any accurate model off of such little data.

All in all, despite being able to improve the baseline slightly, I still believe my model was slightly successful. The structure of the data helped me understand a lot about each individual bidder due to the high volume of bids recorded, but not necessarily anything about common behaviors among robot bidders or human bidders due to the low volume of bidders. Working with such an accurate baseline and such little, varied data, I had the odds against me from the start. However, I was still able to find real correlations among the small test data subset without overfitting to improve accuracy to 94.6% and only incorrectly identify about 1 user for every 20.

# REFERENCES

[1]. http://www.psych.utoronto.ca/users/reingold/courses/ai/turing.html (1999)