Machine Learning Nanodegree

Capstone Proposal

Himanshu Tripathi

1. Domain Background

Artificial Intelligence (and Machine Learning in particular), has aroused a great amount of interest in the population in the recent years due to its wide range of application and to the large number of successful products currently on the market that embrace machine learning at their core, such as Netflix's accurate recommender system [1], Google's amazing translation platform [2] and even chatbots that provide medical assistance, such as MedWhat [3].

But every coin has two sides. Even though the progress and comfort that technology and automation has brought to mankind is undeniable, with it, new problems such as DDoS attacks [4] and click fraud performed by botnets [5] have arose.

Our goal in this project is, precisely, to develop a classifier capable of detecting suspicious bot activity in a fictional auction website that has been experiencing a concerning exodus of users due to unfair robot traffic in the site. This problem corresponds to the following competition posted by Facebook at Kaggle: Human or Bot?.

2. Problem Statement

Goal

To identify online auction bids that are placed by "robots", helping the site owners easily flag these users for removal from their site to prevent unfair auction activity.

Type of machine learning problem

Given our aim is to categorize a user as a robot or human by its bidding activity, we're clearly working on an instance of supervised learning. Moreover, we will build a classifier that'll help us reach our goal and prevent unfair auction activity in the site.

Target function

$$C: Bids \times Bidders \rightarrow \{0, 1\}$$

Here $Bids$ is the bids dataset (described in the next section), $Bidders$ is the bidders (users) dataset (also described in the next section), and the expected output is a label, where 0 corresponds to a human user and 1 to a robot bidder.

Target function representation

We'll build a classifier to approximate our target function. The algorithms the we will explore are:

- Support Vector Machines.
- K-Nearest Neighbors.
- Naïve Bayes.
- AdaBoost.

3. Datasets and Inputs

To tackle this problem, we will use the data already available as part of the Kaggle Competition. There are two datasets to consider:

- **Bidders dataset:** Includes a list of bidder information, including information such as their id, payment account and mailing address. Several of these fields are obfuscated to protect privacy.
- **Bids dataset:** Includes 7.600.000 bids spread across different auctions. The bids in this dataset are all made from mobile devices.

Given that our problem comes from a Kaggle Competition, we didn’t need to perform any action to obtain the data; it was given as part of the competition’s initial resources.

File descriptions

- **train.csv**: Training set from the bidder dataset.
- **test.csv**: Test set from the bidder dataset.
- **bids.csv**: Bids dataset.
- **sampleSubmission.csv**: Sample submission in the correct format.

Datasets fields

For the **bidder dataset**:

- **bidder_id**: Unique identifier of a bidder.
- **payment_account**: Payment account associated with a bidder. Obfuscated to protect privacy.
- **address**: Mailing address of a bidder. Obfuscated to protect privacy.
- **outcome**: Label of a bidder indicating if it is a robot or not. A value of 1.0 translates into a bidder being a robot, whereas a value of 0.0 means the user’s a human. The outcome was half hand labeled and half stats-based. There are two types of “bots” with distinct levels of proof:
    - Bidders who are identified as bots/fraudulent with strong proof. Their accounts have been already banned from the auction site.
    - Bidders who may have just started their business/clicks or their stats deviate from system wide average. There is no clear proof that they are bots.

For the **bids dataset**:

- **bid_id:** Unique id for the bid.
- **bidder_id:** Unique identifier of a bidder. Corresponds to a bidder in the bidder dataset.
- **auction:** Unique identifier of an auction.
- **merchandise:** The category of the auction site campaign, which means the bidder might has come to this site by way of searching for a product category. For instance, if the bidder arrived at the site by a “home goods” search, but ended up bidding for “sporting goods”, this field will contain “home goods”. This categorical field could be a search term or an online ad.

- **device**: Phone model of a visitor.
- **time:** Time when the bid was made (obfuscated to protect privacy).
- **country:** The country that the IP belongs to.
- **ip:** IP address of a bidder (obfuscated to protect privacy).
- **url:** URL where the bidder was referred from (obfuscated to protect privacy).

4. Solution Statement

To tackle the problem described in Section 2, we will use Supervised Learning. We will build a binary classifier, given that our task is to build a model capable of successfully labeling a user as a robot or human based on their auction activity behavior.

We will make a train-validation split on our training set (train.csv; see Section 3 for more details) so we can hold out a portion of it to measure the performance of the model on unseen data and tune accordingly.

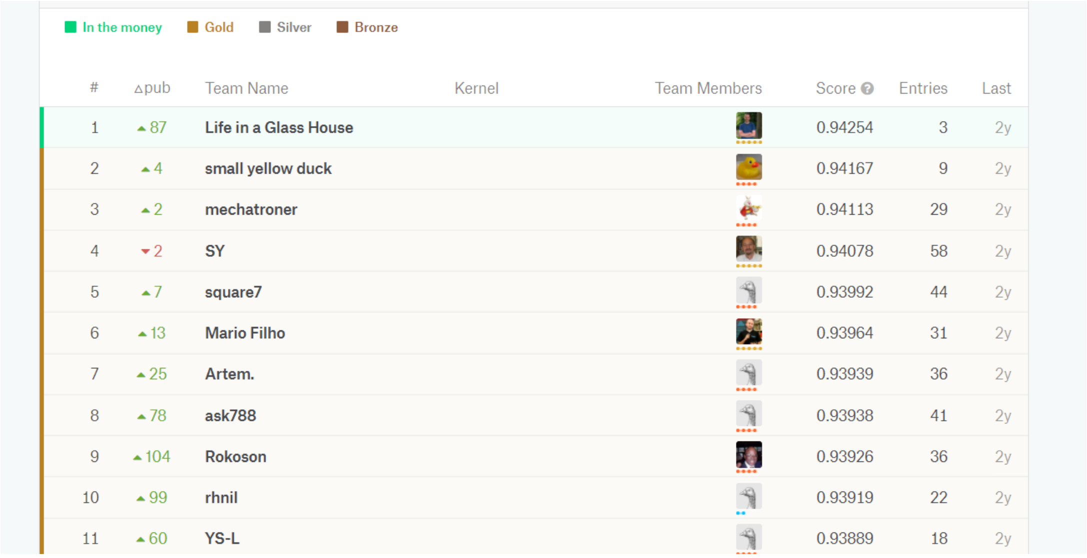
The metric we will use is Area Under Receiver Operating Characteristic Curve [6], which is the metric accepted in the Kaggle Challenge this problem is based on. More details about the metric on Section 6.

5. Benchmark Model

Given the nature of our problem, our benchmark will simply be the score achieved by the winner team of the Kaggle Competition on the Private Leaderboard, which is calculated with approximately 70% of the test data:

$$Benchmark\ Score = Benchmark\ AUC = \ 0.94254$$

The picture below shows the final standings (Winner and Gold – Top 11 teams) for the competition (now closed):



■ In the money ■ Gold ■ Silver ■ Bronze

#	△pub	Team Name	Kernel	Team Members	Score ?	Entries	Last
1	▲87	Life in a Glass House			0.94254	3	2y
2	▲4	small yellow duck			0.94167	9	2y
3	▲2	mechatroner			0.94113	29	2y
4	▼2	SY			0.94078	58	2y
5	▲7	square7			0.93992	44	2y
6	▲13	Mario Filho			0.93964	31	2y
7	▲25	Artem.			0.93939	36	2y
8	▲78	ask788			0.93938	41	2y
9	▲104	Rokoson			0.93926	36	2y
10	▲99	rhnil			0.93919	22	2y
11	▲60	YS-L			0.93889	18	2y

6. Evaluation Metrics

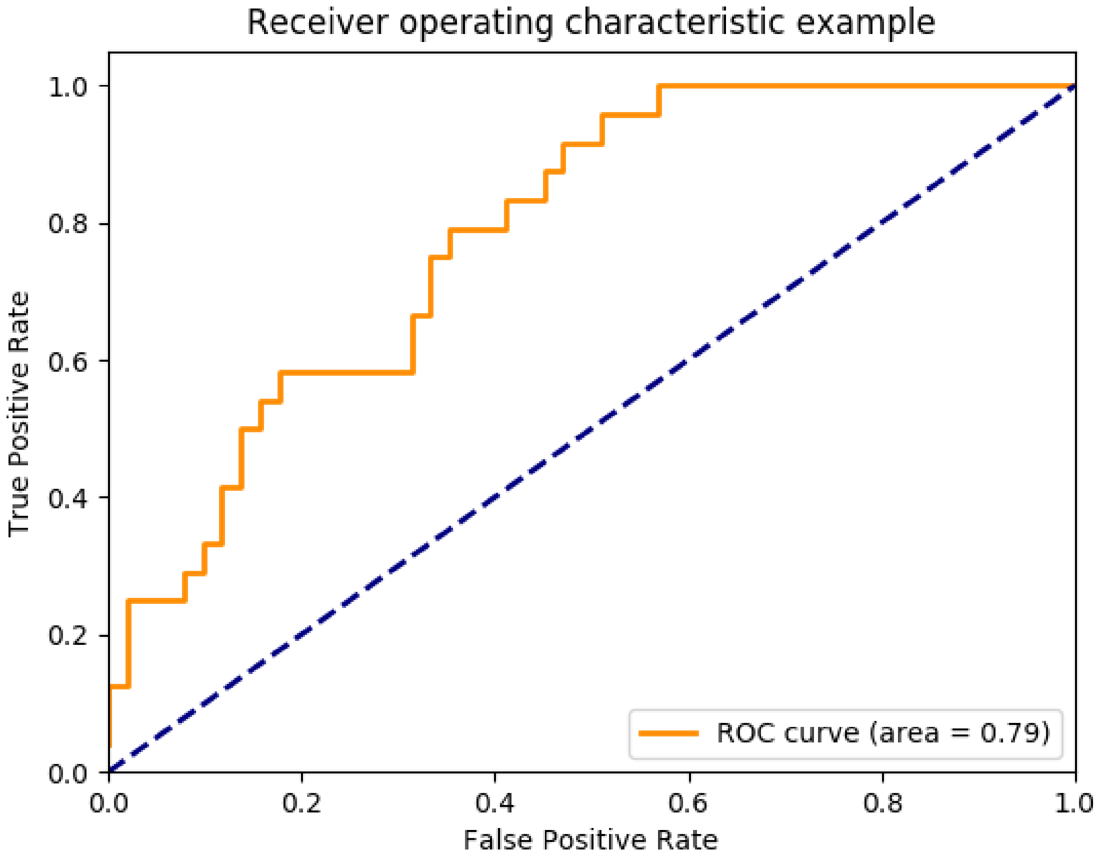
For this project we will use the Area Under Receiver Operating Characteristic Curve as a performance metric, given that's the one used in the Competition and, hence, the benchmark we want to compare our model with.

Receiver Operating Characteristic Curve is a commonly used tool for measuring the performance of a binary classifier. It is a plot that shows the relation between the True Positive Rate and False Positive Rate for every possible decision threshold. These rates are defined as follows:

$$True\ Positive\ Rate = Sensitivity = \frac{True\ Positives}{All\ Positives}$$

$$False\ Positive\ Rate = \ 1 - Specificity = 1 - \frac{True\ Negatives}{All\ Negatives} = \frac{False\ Positives}{All\ Negatives}$$

A typical ROC plot could look like this:



The Area Under Receiver Operating Characteristic Curve is the proportion of the square that falls below and left of the ROC curve (shown in orange). A score of 0.5 means our classifier is no better than random guessing, and 1.0 is the score of a perfect classifier.

One very appealing advantage of Receiver Operating Characteristic Curve is that they are insensitive to poorly predicted "probabilities". This means that if what our classifier outputs is an actual probability between 0.0 and 1.0, or simply a number between, say, 0.9 and 1.0 it won't make any difference in the resulting curve, because it focuses on how well the model separates the two classes [7]. Thus, we can think of AUC as the probability of the classifier ranking higher a randomly chosen positive observation than a randomly chosen negative observation, which makes it a great metric for datasets with highly unbalanced classes.

7. Project Design

Programming Language and Libraries

- Python 3.5.
- Scikit-learn 0.19.0
- Pandas 0.20.3
- NumPy 1.13.3
- Seaborn 0.8.1
- Matplotlib 2.1.0

Datasets and files

- **train.csv**: Training set corresponding to the bidders' dataset.
- **test.csv:** Test set corresponding to the bidders' dataset.
- **bids.csv:** Auctions' bids information.

For more details on datasets and files, refer to Section 3.

Machine Learning Design

- **Problem category:** Supervised learning.
- **Output:** Discrete. Categorical. Classification task.
- **Target function:** $C: Bids \times Bidders \rightarrow \{0, 1\}$ **,** where $Bids$ is the bids dataset, $Bidders$ is the bidders (users) dataset and the expected output is a label, where 0 corresponds to a human user and 1 to a robot bidder.
- **Representation of learned function:** Binary classifier. Candidates are:
  - AdaBoost.
  - Naïve Bayes.
  - K-Nearest Neighbors.
  - Support Vector Machines.

Workflow

Here's the workflow we will follow in this project:

1. Exploratory Data Analysis of bidders’ (train) dataset:
2. Exploratory Data Analysis of bids dataset.
3. Data preprocessing.
4. Feature engineering from data in bids dataset.
5. Feature selection.
6. Feature transformation (Using Component Analysis algorithms such as PCA, ICA or RCA).
7. Comparison between the classifiers described in previous sections.
    a. Model performance analysis with learning curves.
    b. Model performance evaluation.
8. Selected model tuning using GridSearch.
9. Comparison between obtained results and benchmark score.

References

[1] https://dl.acm.org/citation.cfm?id=2843948

[2] https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

[3] https://medwhat.com/

[4] https://en.wikipedia.org/wiki/Denial-of-service_attack

[5] https://dl.acm.org/citation.cfm?id=1323139

[6] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

[7] https://www.youtube.com/watch?v=OAl6eAyP-yo