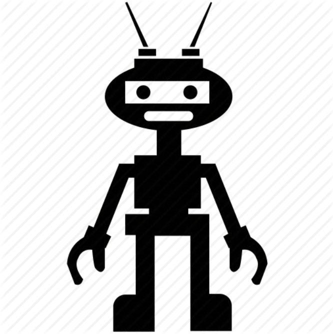
Facebook Recruiting IV: Human or Robot?

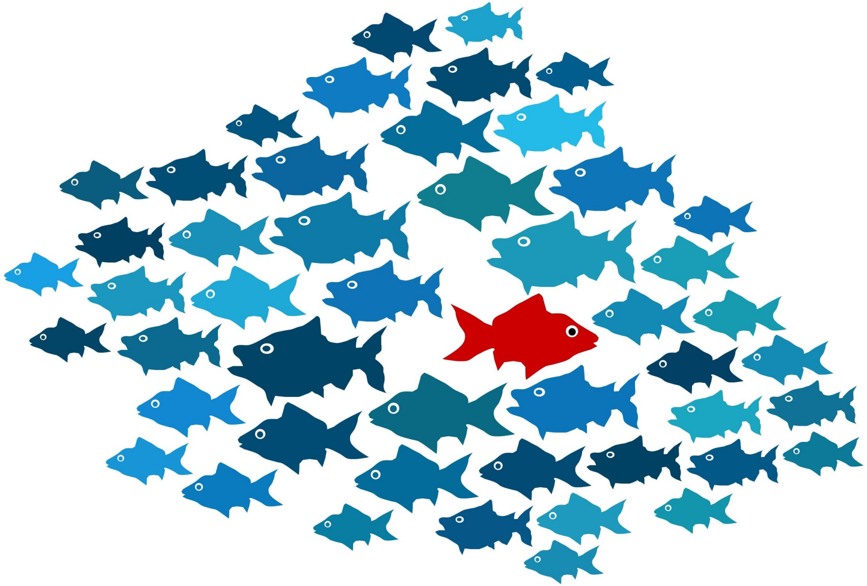


Team R-Clique

Daniel Geng, Meghana Ginjpalli, Sara Melvin, Sonu Mishra, Andrew Wong

Outline

- Background
- Methodology
  - Temporal-based Approach: BIRDNEST
  - Classic Machine Learning Approaches
  - Kaggle Winner: Small Yellow Duck
- Summary



Background: Facebook Ads Bidding

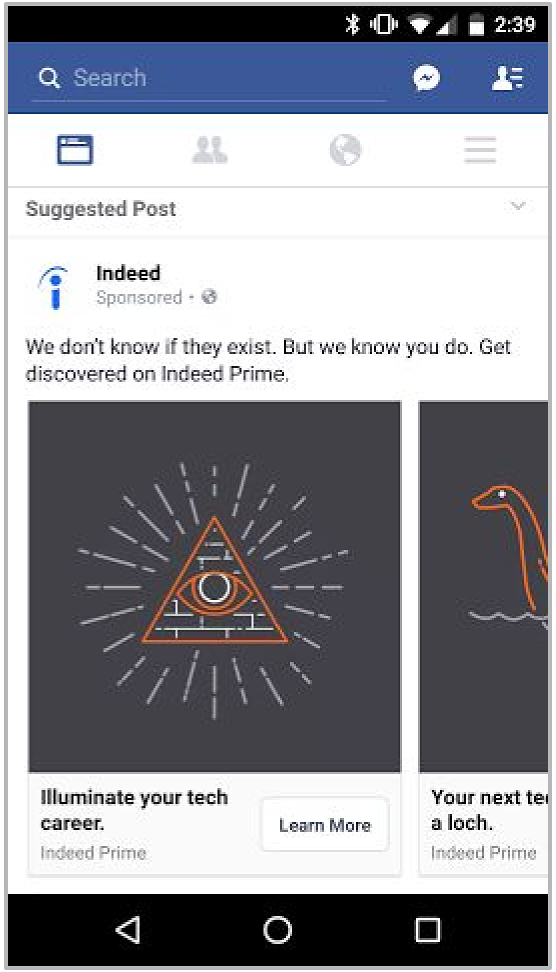
Businesses need to be able to advertise to their target audiences

Facebook auctions off its real estate so that businesses can display their advertisements

Example

Example

indeed
one search. all jobs.



Suggested Post

Indeed
Sponsored ·

We don't know if they exist. But we know you do. Get discovered on Indeed Prime.

Illuminate your tech career.
Indeed Prime

Your next te
a loch.
Indeed Prime

rent.com

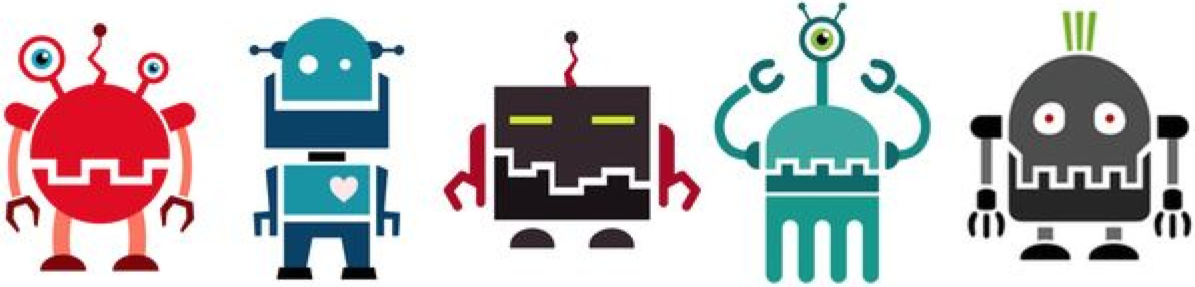
Example

indeed
one search. all jobs.

rent.com

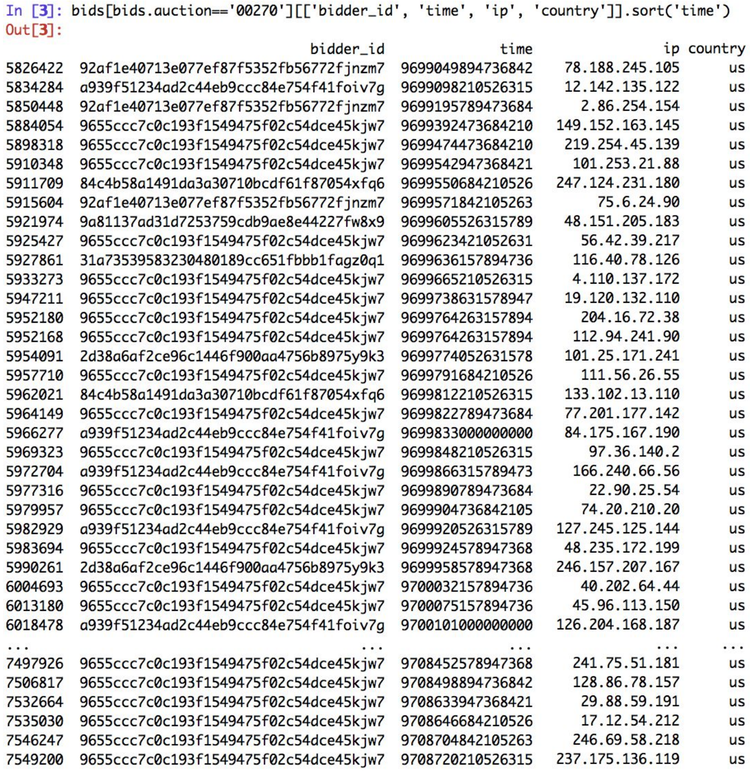
Background: Kaggle Competition

- Human bidders are frustrated as robots keep winning in Facebook auctions
- Result: Customers base is plummeting
- Need to eliminate computer generated bidding from these auctions
- **Goal**: Predict if an online bid is made by a machine or a human



Background: Kaggle Competition

- April 27, 2015 - June 8, 2015
- 985 teams
- Bidder Dataset (~2000 instances):
    - bidder_id, payment_account, address, outcome
- Bid Dataset (~7.6 million instances):
    - bid_id, bidder_id, auction, merchandise, device, time, country, ip, url
- Evaluation is based on area under the ROC curve (AUC)

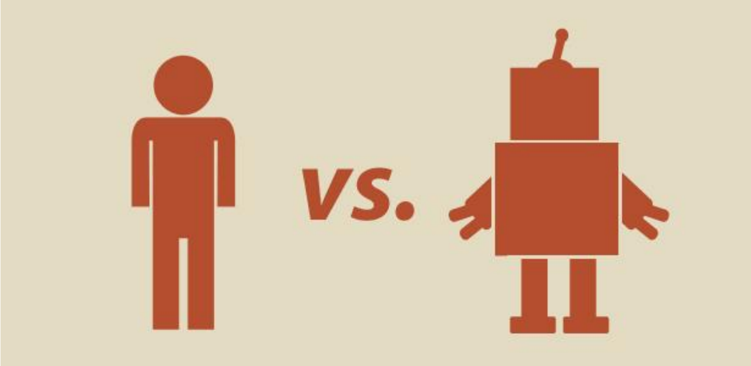


```
In [3]: bids[bids.auction=='00270'][['bidder_id', 'time', 'ip', 'country']].sort('time')
Out[3]:
                                    bidder_id              time               ip country
5826422  92af1e40713e077ef87f5352fb56772fjnzm7  9699049894736842   78.188.245.105      us
5834284  a939f51234ad2c44eb9ccc84e754f41foiv7g  9699098210526315   12.142.135.122      us
5850448  92af1e40713e077ef87f5352fb56772fjnzm7  9699195789473684     2.86.254.154      us
5884054  9655ccc7c0c193f1549475f02c54dce45kjw7  9699392473684210  149.152.163.145      us
5898318  9655ccc7c0c193f1549475f02c54dce45kjw7  9699474473684210   219.254.45.139      us
5910348  9655ccc7c0c193f1549475f02c54dce45kjw7  9699542947368421    101.253.21.88      us
5911709  84c4b58a1491da3a30710bcdf61f87054xfq6  9699550684210526  247.124.231.180      us
5915604  92af1e40713e077ef87f5352fb56772fjnzm7  9699571842105263       75.6.24.90      us
5921974  9a81137ad31d7253759cdb9ae8e44227fw8x9  9699605526315789   48.151.205.183      us
5925427  9655ccc7c0c193f1549475f02c54dce45kjw7  9699623421052631     56.42.39.217      us
5927861  31a73539583230480189cc651fbbb1fagz0q1  9699636157894736    116.40.78.126      us
5933273  9655ccc7c0c193f1549475f02c54dce45kjw7  9699665210526315    4.110.137.172      us
5947211  9655ccc7c0c193f1549475f02c54dce45kjw7  9699738631578947   19.120.132.110      us
5952180  9655ccc7c0c193f1549475f02c54dce45kjw7  9699764263157894     204.16.72.38      us
5952168  9655ccc7c0c193f1549475f02c54dce45kjw7  9699764263157894    112.94.241.90      us
5954091  2d38a6af2ce96c1446f900aa4756b8975y9k3  9699774052631578   101.25.171.241      us
5957710  9655ccc7c0c193f1549475f02c54dce45kjw7  9699791684210526     111.56.26.55      us
5962021  84c4b58a1491da3a30710bcdf61f87054xfq6  9699812210526315   133.102.13.110      us
5964149  9655ccc7c0c193f1549475f02c54dce45kjw7  9699822789473684   77.201.177.142      us
5966277  a939f51234ad2c44eb9ccc84e754f41foiv7g  9699833000000000   84.175.167.190      us
5969323  9655ccc7c0c193f1549475f02c54dce45kjw7  9699848210526315      97.36.140.2      us
5972704  a939f51234ad2c44eb9ccc84e754f41foiv7g  9699866315789473    166.240.66.56      us
5977316  9655ccc7c0c193f1549475f02c54dce45kjw7  9699890789473684      22.90.25.54      us
5979957  9655ccc7c0c193f1549475f02c54dce45kjw7  9699904736842105     74.20.210.20      us
5982929  a939f51234ad2c44eb9ccc84e754f41foiv7g  9699920526315789  127.245.125.144      us
5983694  9655ccc7c0c193f1549475f02c54dce45kjw7  9699924578947368   48.235.172.199      us
5990261  2d38a6af2ce96c1446f900aa4756b8975y9k3  9699958578947368  246.157.207.167      us
6004693  9655ccc7c0c193f1549475f02c54dce45kjw7  9700032157894736     40.202.64.44      us
6013180  9655ccc7c0c193f1549475f02c54dce45kjw7  9700075157894736    45.96.113.150      us
6018478  a939f51234ad2c44eb9ccc84e754f41foiv7g  9700101000000000  126.204.168.187      us
...                                        ...               ...              ...     ...
7497926  9655ccc7c0c193f1549475f02c54dce45kjw7  9708452578947368    241.75.51.181      us
7506817  9655ccc7c0c193f1549475f02c54dce45kjw7  9708498894736842    128.86.78.157      us
7532664  9655ccc7c0c193f1549475f02c54dce45kjw7  9708633947368421     29.88.59.191      us
7535030  9655ccc7c0c193f1549475f02c54dce45kjw7  9708646684210526     17.12.54.212      us
7546247  9655ccc7c0c193f1549475f02c54dce45kjw7  9708704842105263    246.69.58.218      us
7549200  9655ccc7c0c193f1549475f02c54dce45kjw7  9708720210526315  237.175.136.119      us
```

Image from [1]

Outline

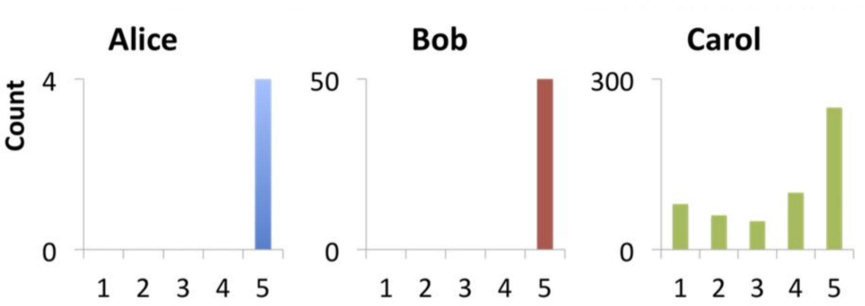
- Background
- Methodology
  - **Temporal-based Approach: BIRDNEST**
  - Classic Machine Learning Approaches
  - Kaggle Winner: Small Yellow Duck
- Summary

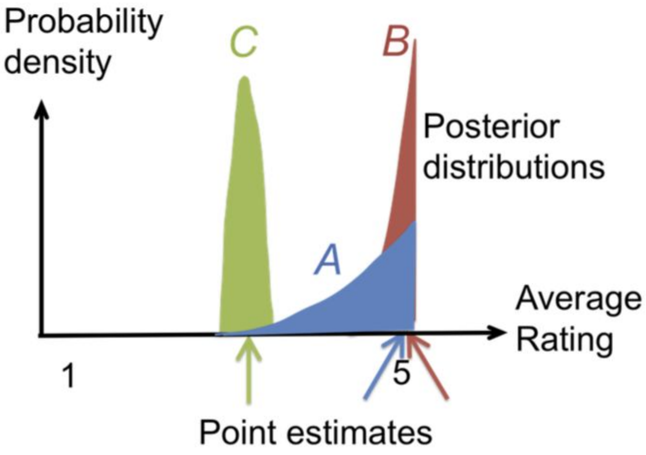


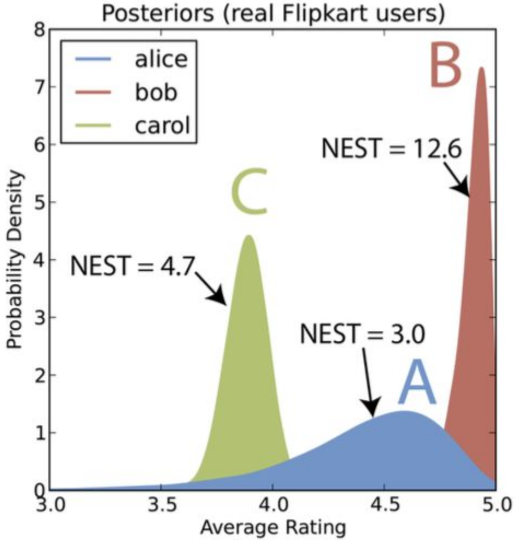
BIRDNEST - Intuition

Two typical questions a person would ask when determining anomalous behavior:

1) What is the distribution of a user? - BIRD
2) How suspicious is that distribution? - NEST







Figures from [2]

BIRDNEST - First Attempt

Extracted only one feature:

- Inter-Arrival Time (IAT) distribution - time between user’s bids

From BIRDNEST paper:

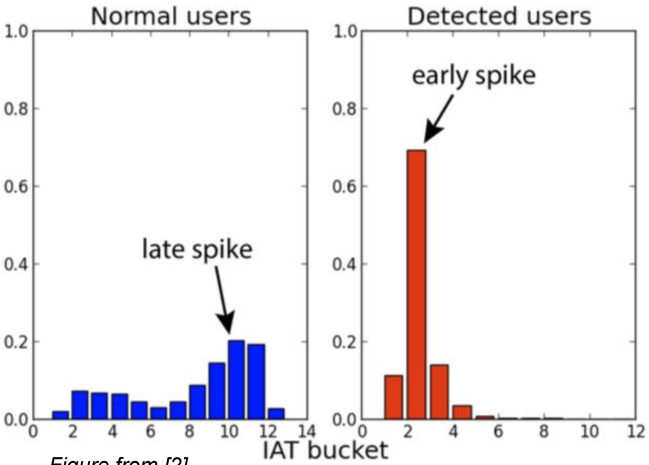
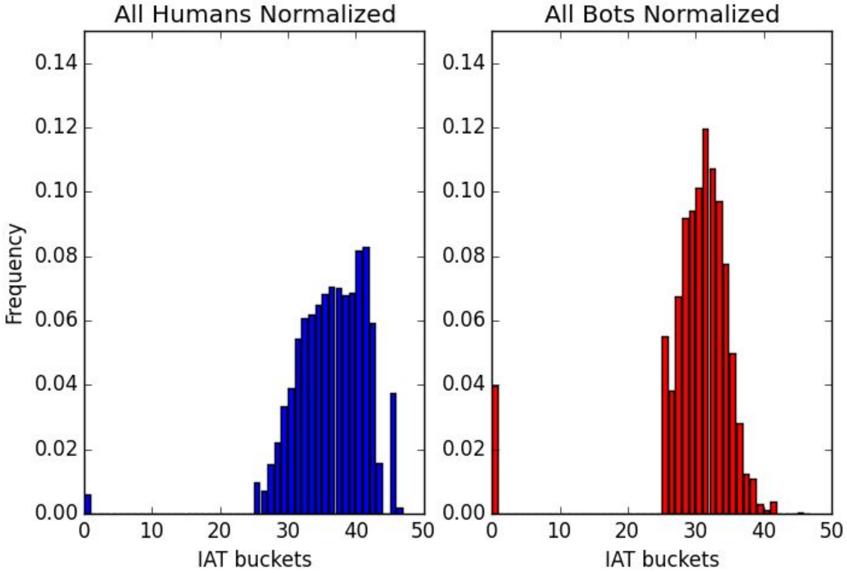


Figure from [2]

In our Data Set -

1984 total users

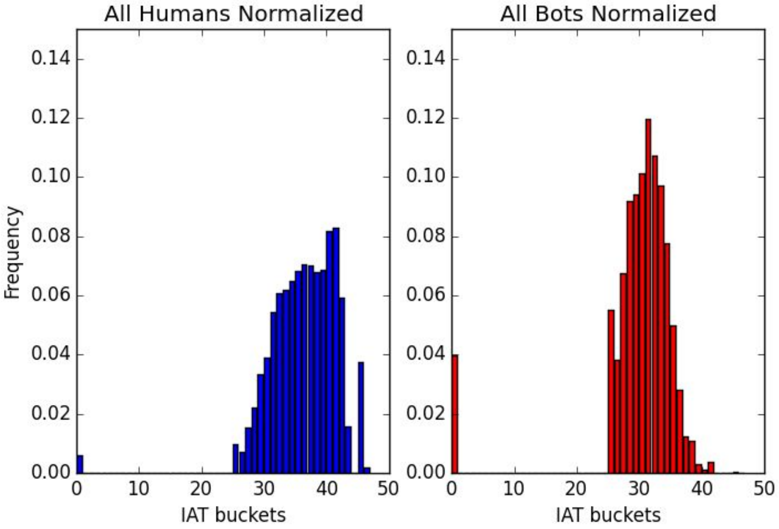
103 bots

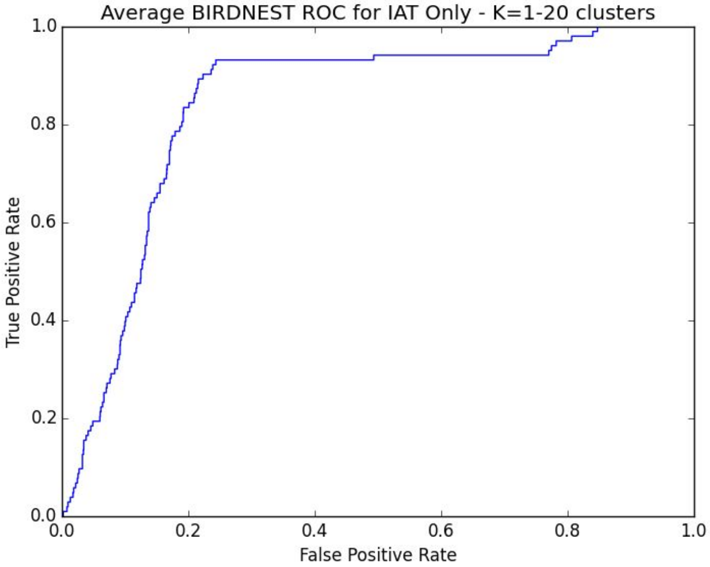


BIRDNEST - First Attempt

Extracted only one feature:

- **Inter-Arrival Time (IAT)** distribution
  - time between user's bids





Average AUC = 0.8415

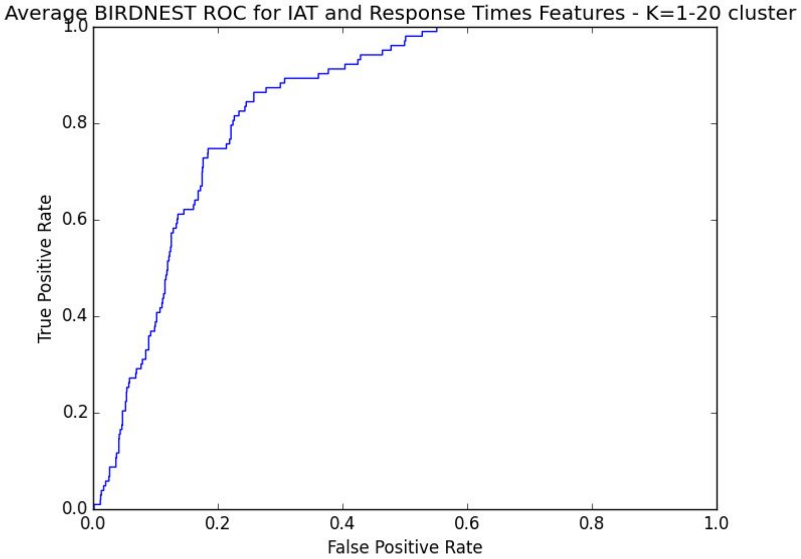
BIRDNEST

Features Extracted:

- Inter-Arrival Time (IAT) distribution - time between user’s bids
- **Response Times distribution** - time between the previous bidder’s bid and the user’s bid

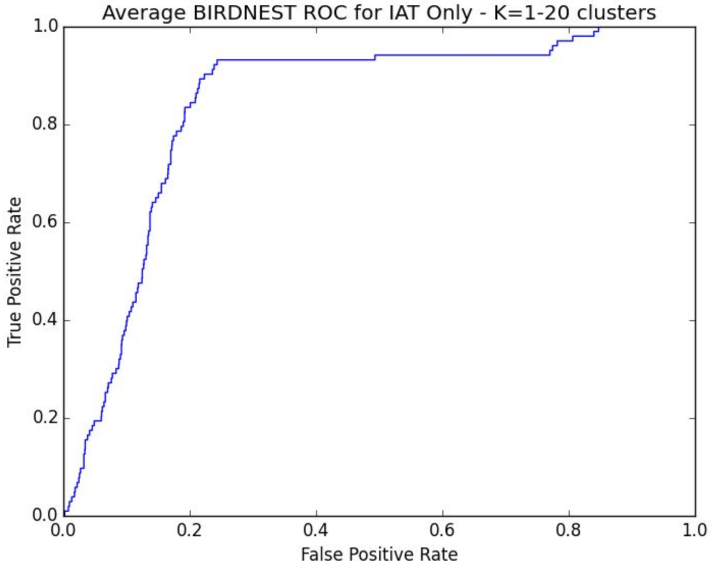
Log-binned Histograms

- IAT distribution = 47 buckets
- Response Time distribution = 44 buckets

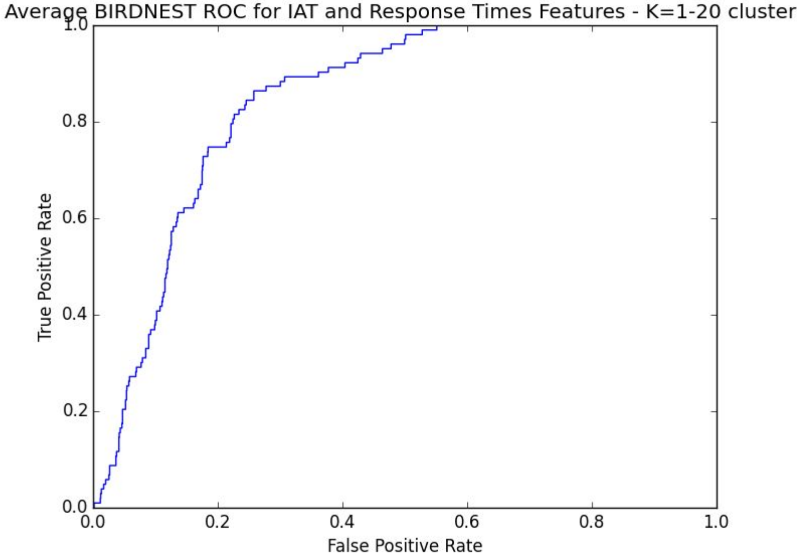


Average AUC = 0.8456

BIRDNEST



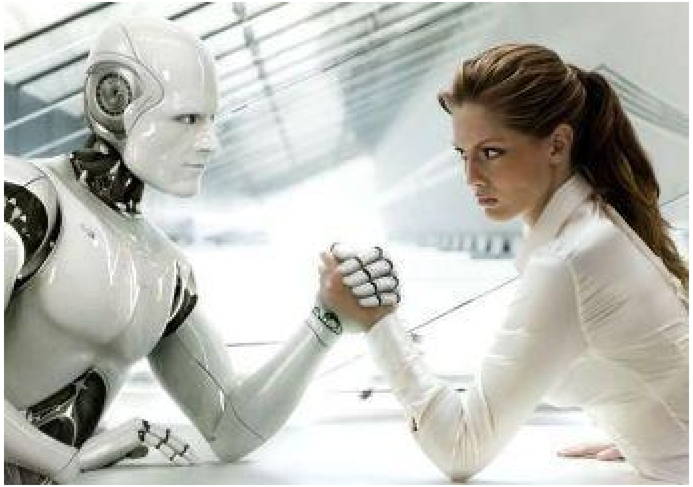
Average AUC = 0.8415



Average AUC = 0.8456

Outline

- Background
- Methodology
  - Temporal-based Approach: BIRDNEST
  - **Classic Machine Learning Approaches**
  - Kaggle Winner: Small Yellow Duck
- Summary



Our Features

Bucketized IAT (base 2, 5)

Bucketized RTs (base 2, 5)

Average IAT

Average Response Time

Number of bids total

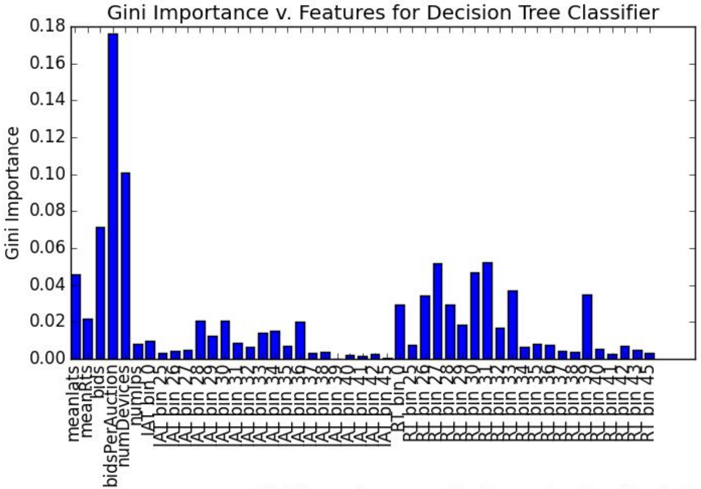
Mean number of bids per auction

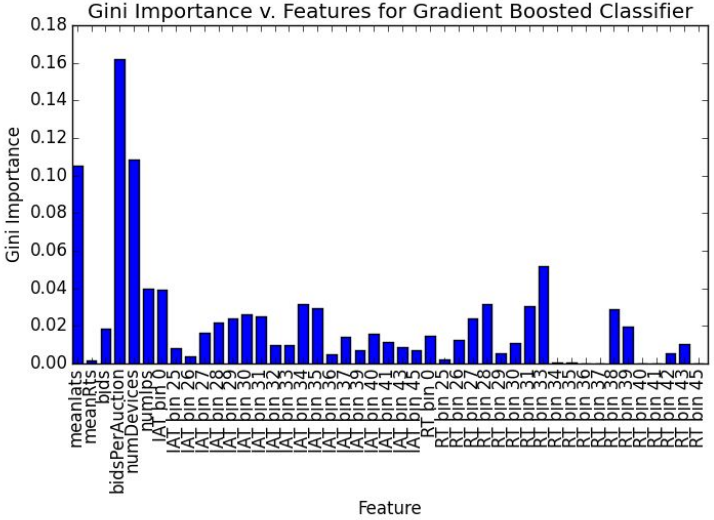
Number of devices

Number of IPs

Entropy of devices

Entropy of IPs





Classification Approaches

FIrst Attempts

SVM (linear, polynomial, radial)

Logistic Regression (primal, dual)

Boosting

Adaboost (weights each classifier based on performance)

Gradient Boosting (fits trees based on negative gradient of loss function)

Random Forests

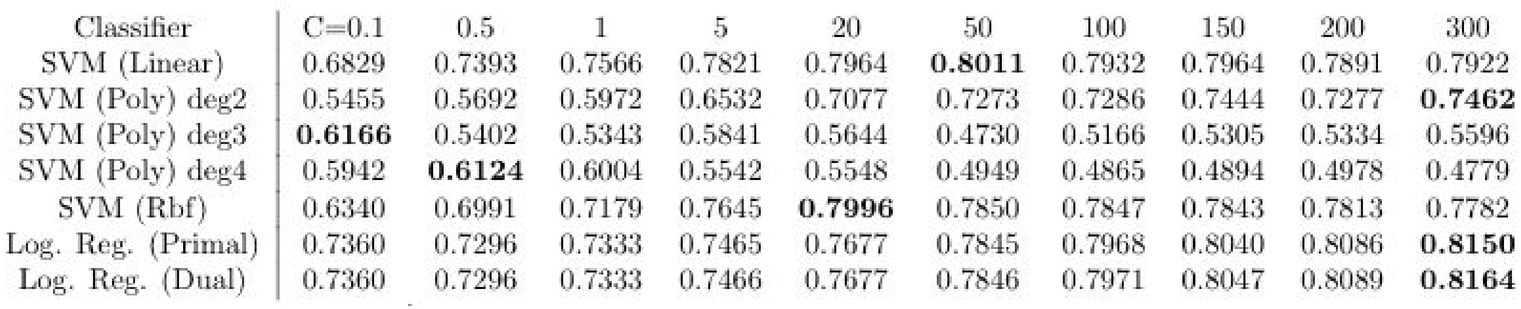
Bagging (trains on subsets of data and uses them to vote)

Extra Trees (trains on subsets of data and averages them)

Parameter Sweep

Finding an optimal classifier using our features:

- swept over the inverse regularization term C
- tried different kernels (SVM) and optimization methods (log. regression)
- Best AUC came from logistic regression*



Classifier	C=0.1	0.5	1	5	20	50	100	150	200	300
SVM (Linear)	0.6829	0.7393	0.7566	0.7821	0.7964	**0.8011**	0.7932	0.7964	0.7891	0.7922
SVM (Poly) deg2	0.5455	0.5692	0.5972	0.6532	0.7077	0.7273	0.7286	0.7444	0.7277	**0.7462**
SVM (Poly) deg3	**0.6166**	0.5402	0.5343	0.5841	0.5644	0.4730	0.5166	0.5305	0.5334	0.5596
SVM (Poly) deg4	0.5942	**0.6124**	0.6004	0.5542	0.5548	0.4949	0.4865	0.4894	0.4978	0.4779
SVM (Rbf)	0.6340	0.6991	0.7179	0.7645	**0.7996**	0.7850	0.7847	0.7843	0.7813	0.7782
Log. Reg. (Primal)	0.7360	0.7296	0.7333	0.7465	0.7677	0.7845	0.7968	0.8040	0.8086	**0.8150**
Log. Reg. (Dual)	0.7360	0.7296	0.7333	0.7466	0.7677	0.7846	0.7971	0.8047	0.8089	**0.8164**

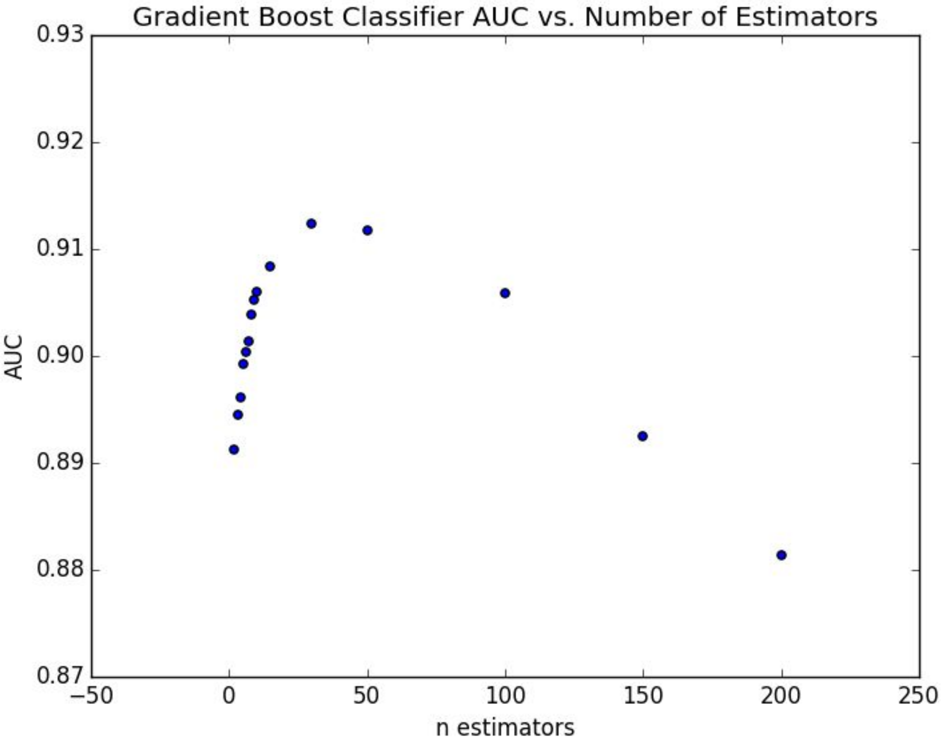
Parameter Sweep

Finding an optimal classifier using our features:

- swept over the number of estimators in the ensemble
- tried different loss functions

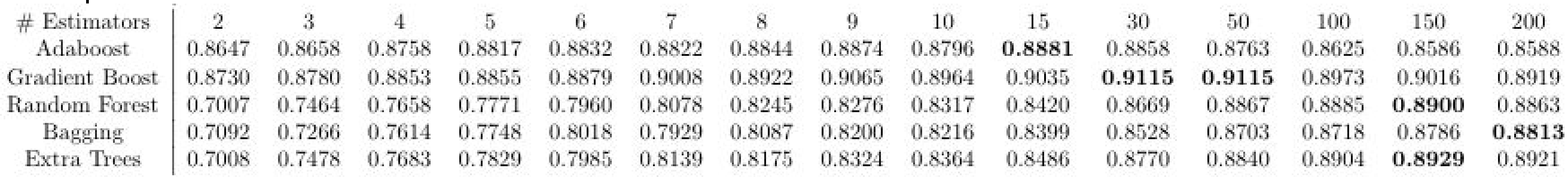
Cross-validation to prevent overfitting

- split up the *users* with 80/20 partition, trained/tested each classifier with partitions



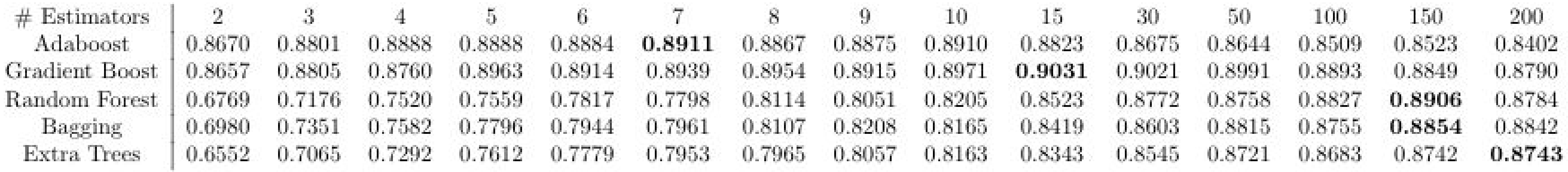
Parameter Sweep

Sweep for Five Features



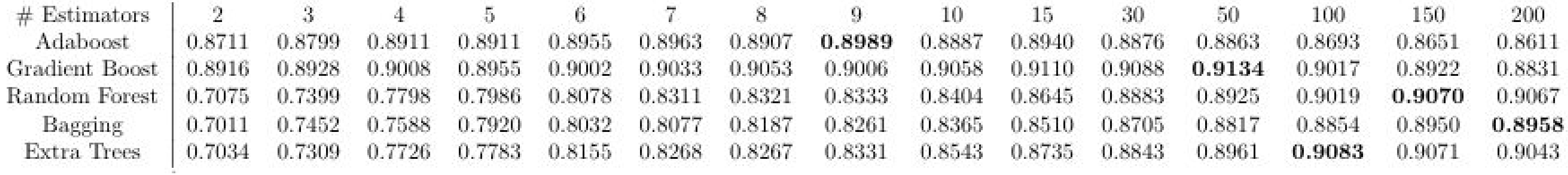
# Estimators	2	3	4	5	6	7	8	9	10	15	30	50	100	150	200
Adaboost	0.8647	0.8658	0.8758	0.8817	0.8832	0.8822	0.8844	0.8874	0.8796	**0.8881**	0.8858	0.8763	0.8625	0.8586	0.8588
Gradient Boost	0.8730	0.8780	0.8853	0.8855	0.8879	0.9008	0.8922	0.9065	0.8964	0.9035	**0.9115**	**0.9115**	0.8973	0.9016	0.8919
Random Forest	0.7007	0.7464	0.7658	0.7771	0.7960	0.8078	0.8245	0.8276	0.8317	0.8420	0.8669	0.8867	0.8885	**0.8900**	0.8863
Bagging	0.7092	0.7266	0.7614	0.7748	0.8018	0.7929	0.8087	0.8200	0.8216	0.8399	0.8528	0.8703	0.8718	0.8786	**0.8813**
Extra Trees	0.7008	0.7478	0.7683	0.7829	0.7985	0.8139	0.8175	0.8324	0.8364	0.8486	0.8770	0.8840	0.8904	**0.8929**	0.8921

Sweep for All Features



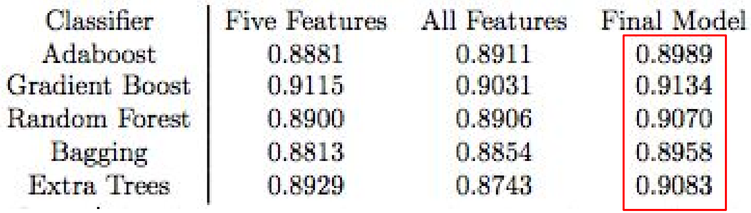
# Estimators	2	3	4	5	6	7	8	9	10	15	30	50	100	150	200
Adaboost	0.8670	0.8801	0.8888	0.8888	0.8884	**0.8911**	0.8867	0.8875	0.8910	0.8823	0.8675	0.8644	0.8509	0.8523	0.8402
Gradient Boost	0.8657	0.8805	0.8760	0.8963	0.8914	0.8939	0.8954	0.8915	0.8971	**0.9031**	0.9021	0.8991	0.8893	0.8849	0.8790
Random Forest	0.6769	0.7176	0.7520	0.7559	0.7817	0.7798	0.8114	0.8051	0.8205	0.8523	0.8772	0.8758	0.8827	**0.8906**	0.8784
Bagging	0.6980	0.7351	0.7582	0.7796	0.7944	0.7961	0.8107	0.8208	0.8165	0.8419	0.8603	0.8815	0.8755	**0.8854**	0.8842
Extra Trees	0.6552	0.7065	0.7292	0.7612	0.7779	0.7953	0.7965	0.8057	0.8163	0.8343	0.8545	0.8721	0.8683	0.8742	**0.8743**

Sweep for Four Features + Entropies



# Estimators	2	3	4	5	6	7	8	9	10	15	30	50	100	150	200
Adaboost	0.8711	0.8799	0.8911	0.8911	0.8955	0.8963	0.8907	**0.8989**	0.8887	0.8940	0.8876	0.8863	0.8693	0.8651	0.8611
Gradient Boost	0.8916	0.8928	0.9008	0.8955	0.9002	0.9033	0.9053	0.9006	0.9058	0.9110	0.9088	**0.9134**	0.9017	0.8922	0.8831
Random Forest	0.7075	0.7399	0.7798	0.7986	0.8078	0.8311	0.8321	0.8333	0.8404	0.8645	0.8883	0.8925	0.9019	**0.9070**	0.9067
Bagging	0.7011	0.7452	0.7588	0.7920	0.8032	0.8077	0.8187	0.8261	0.8365	0.8510	0.8705	0.8817	0.8854	0.8950	**0.8958**
Extra Trees	0.7034	0.7309	0.7726	0.7783	0.8155	0.8268	0.8267	0.8331	0.8543	0.8735	0.8843	0.8961	**0.9083**	0.9071	0.9043

Best Estimators



Classifier	Five Features	All Features	Final Model
Adaboost	0.8881	0.8911	0.8989
Gradient Boost	0.9115	0.9031	0.9134
Random Forest	0.8900	0.8906	0.9070
Bagging	0.8813	0.8854	0.8958
Extra Trees	0.8929	0.8743	0.9083

Average IAT

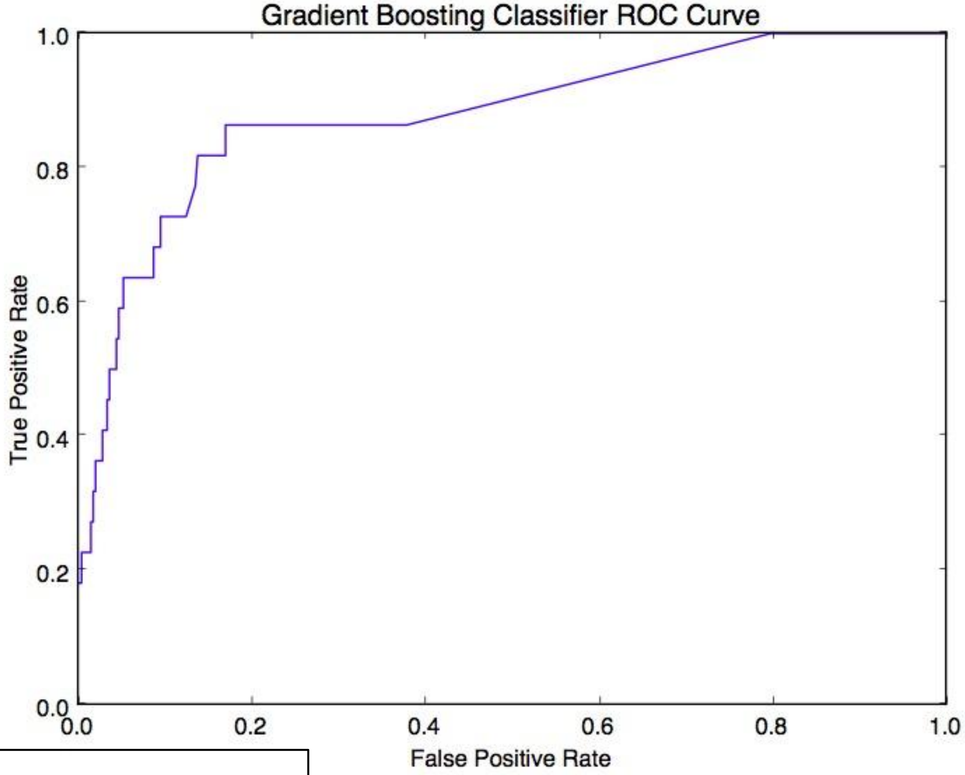
Number of bids total

Mean number of bids per auction

Number of devices

Entropy of devices

Entropy of IPs



Average AUC = 0.9134

Combined Model

Train five different estimators and vote

- Used optimal classifiers from previous slides to train model
- Tried both mean and median to threshold
- Better than everything but Gradient Boosting Classifier

Train the same estimator with different start states and vote

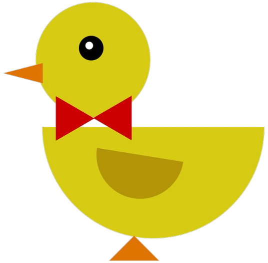
- Swept over a range of number of estimators with no clear trend
- Sometimes outperformed Gradient Boosting Classifier

Used different base estimators in ensemble methods

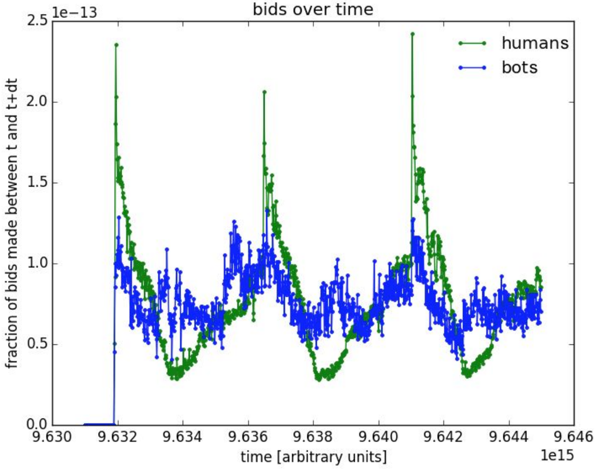
Outline

- Background
- Methodology
  - Temporal-based Approach: BIRDNEST
  - Classic Machine Learning Approaches
  - **Kaggle Winner: Small Yellow Duck**
- Summary

Kaggle Winner: Small Yellow Duck



- Observations:
    - Human bidding activity peaks daily due to auctions ending at the same time everyday
    - Auctions tend to last for more than two weeks
    - Robots do not place any bids between 11 to 14 days before the auction ends



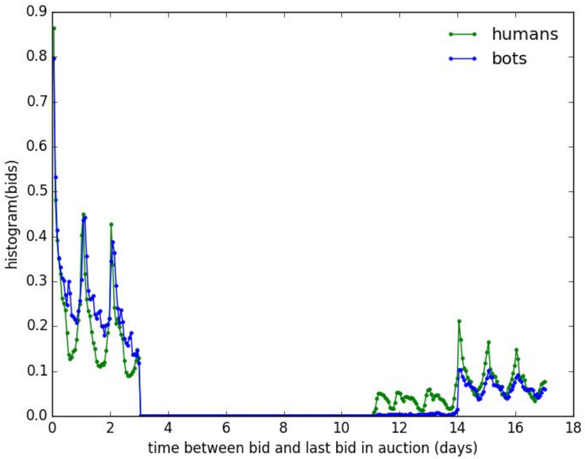


Image from [1]

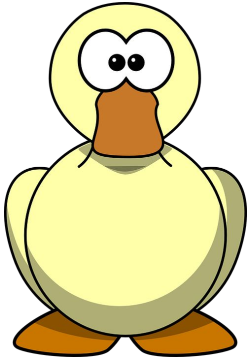
Kaggle Winner: Small Yellow Duck

- Feature Extraction:
    - entropy for how many bids a user placed on each day of the week
    - max number of bids in a 20 min span
    - total number of bids placed by user
    - average number of bids a user placed per URL
    - number of bids placed by the user on each of three weekdays in the data
    - median time between user’s bid and user’s previous bid
    - mean number of bids a user made per auction
    - min and median times between a user’s bid and previous bid by another user in the same auction

Kaggle Winner: Small Yellow Duck

- Classification Model: average of the probabilities predicted by five instances of the **RandomForestClassifier**
- Random Forests = ensemble learning method that constructs multiple decision trees in order to create a stronger classification model
- Runtime
  - Training and Predicting ~ 3 min
  - Cross validation with 100+ different train/valid splits ~ 20 min
    - 80% train, 20% validation

AUC Score = 0.94167



Summary

Algorithm	AUC
Small Yellow Duck: Random Forests	0.9417
Gradient Boost	0.9134
Extra Trees	0.9083
Random Forests	0.9070
Adaboost	0.8989
Bagging	0.8958
BIRDNEST	0.8456
Logistic Regression	0.8164
SVM (linear kernel)	0.8011

References

[1] Kaggle Team. "Facebook IV Winner's Interview: 2nd Place, Kiri Nichol(aka Small Yellow Duck)." *No Free Hunch*. N.p., 19 June 2015. Web. 20 Feb. 2016.

[2] Hooi, B., N. et. al. “BIRDNEST: Bayesian Inference for Ratings-Fraud Detection.” arXiv:1511.06030. Nov 2015.

Thank You!

Team R-Clique