Classification Task:
Click-Through-Rate prediction

Paul Pereira

Background : Real-Time Bidding

- The ad industry is worth billions of dollars. Google AdSense provides the majority of Google's revenue.
- The marketplace is organized as a real-time auction.
- When a user visits a webpage that displays ads, a bid requests is sent to advertisers participating in the market.
- The auctions are usually run as Generalized Second Price Auctions
- In order to determine whether the player should bid, or what amount to bid on the ad placement, we need to know the CTR of the ad.
- The expected utility of the player given bid b is defined as:

  -U(b, x, p) $= P(win|b) * CTR(x) * R_p - b$

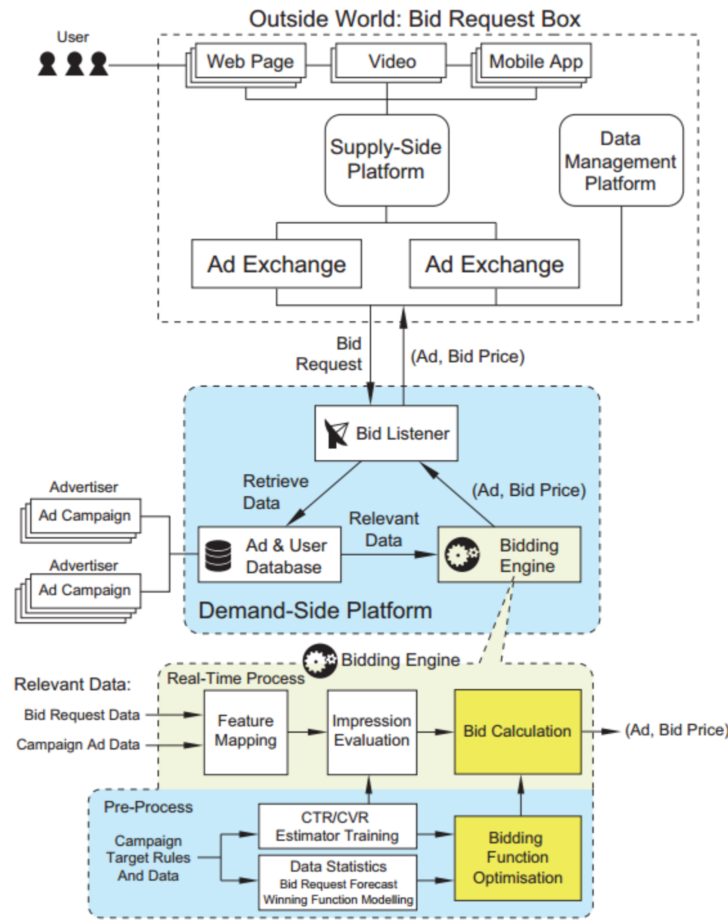


Figure 1: An illustration of a demand-side platform and its bidding engine in RTB display advertising.

[2]

iPinYou dataset

- The dataset used is from the iPinYou global RTB bidding algorithm competition
- Provides over 400 million examples of successful auction bids
- Data manipulations:
  - Remove fields that have no predictive value
  - Use one hot encoding for every feature except floor price (used for logistic regression and SVM)
  - Use frequency of feature in cases where the ad was clicked on. (used for boosting)

Col #	Description	Example
*1	Bid ID	015300008...3f5a4f5121
2	Timestamp	20130218001203638
†3	Log type	1
*4	iPinYou ID	35605620124122340227135
5	User-Agent	Mozilla/5.0 (compatible; \ MSIE 9.0; Windows NT \ 6.1; WOW64; Trident/5.0)
*6	IP	118.81.189.*
7	Region	15
8	City	16
*9	Ad exchange	2
*10	Domain	e80f4ec7...c01cd1a049
*11	URL	hz55b00000...3d6f275121
12	Anonymous URL ID	Null
13	Ad slot ID	2147689_8764813
14	Ad slot width	300
15	Ad slot height	250
16	Ad slot visibility	SecondView
17	Ad slot format	Fixed
*18	Ad slot floor price	0
19	Creative ID	e39e178ffd...1ee56bcd
*20	Bidding price	753
*†21	Paying price	15
*†22	Key page URL	a8be178ffd...1ee56bcd
*23	Advertiser ID	2345
*24	User Tags	123,5678,3456

[1]

Classification Task

- Given a feature vector representing a bidding request, predict the probability that the user will click on the ad (other KPIs available as well).
- Usual classifiers to try are logistic regression and SVM with a linear kernel.
- Can improve results by training one classifier by bidder.
- Need to deal with class imbalance(about 1 in 10000 ads gets clicked on).
  - Over-sampling or over-penalize miss-classification of positive examples.

Tree Boosting

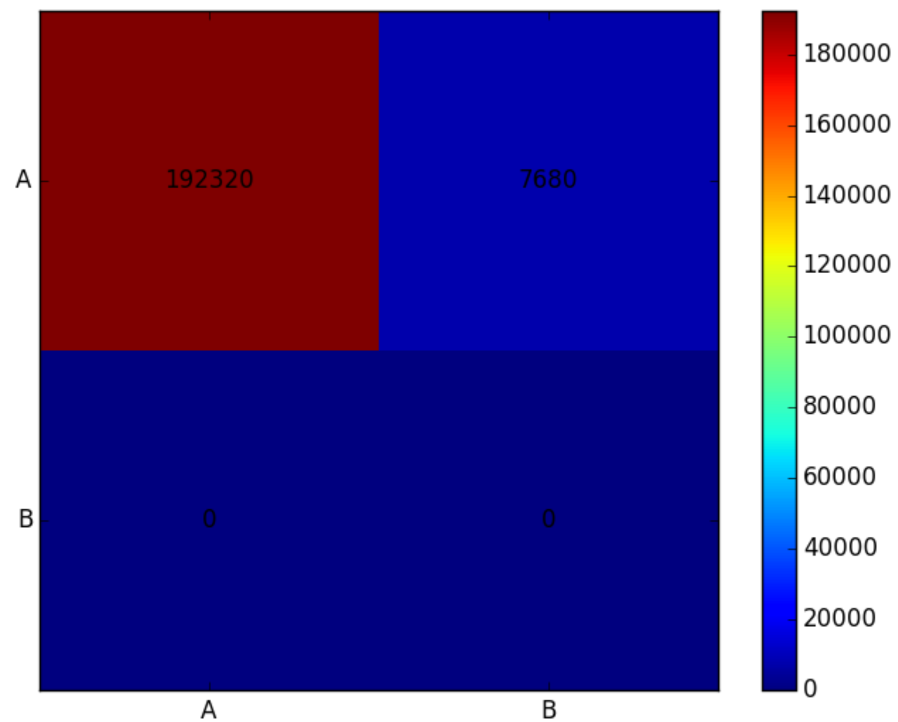
- Motivation : Performs well in classification competitions and deals with class imbalance.
- The idea is to train many simple classifiers, in this case decisions trees and combine the information from those classifiers to make a better prediction.
  - $F(x) = \sum \lambda_m * h_m(x)$
- Basic Idea behind learning a decision tree:
  - Pick a feature (in our case all the features are real values)
  - Find t that maximizes the information gain:
    - $IG(Y|X:t) = H(Y|X < t) * P(X < t) + H(Y|X > t) * P(X > t)$
- Basic Idea behind updating the ensemble:
  - Pick a tree that minimizes the loss function when trained on the dataset where the label is the residual r_i
    - $r_i = -\frac{\nabla L(y_i,F(x_i))}{\nabla F(x_i)}$

Results (so far)

- Methodology : Used 1M examples, split 80/20 into training and testing.
- Used 10-fold cross validation to pick best features (tree depth, l2 regularization).

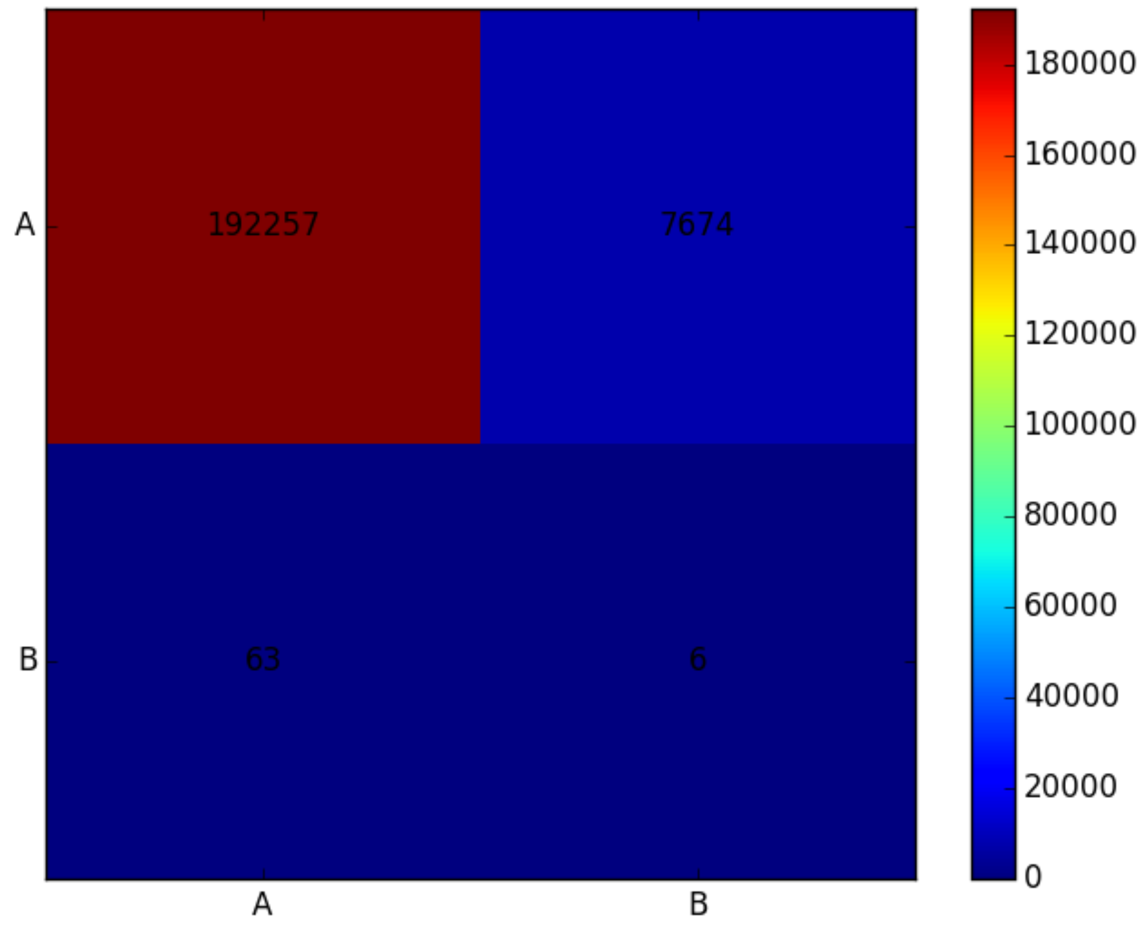
Confusion Matrix for LR and SVM

$A\_00 = 192320$
$A\_01 = 7680$
$A\_10 = 0$
$A\_11 = 0$



Confusion Matrix for Boosted Decision Tree

A_00 = 192257
A_01 = 7674
A_10 = 63
A_11 = 6



References

- [1] Real-Time Bidding Benchmarking with iPinYou, Weinan Zhang, Jun Wang
- [2] Optimal Real-Time Bidding for Display Advertising, Weinan Zhang
- [3] Practical Lessons from predicting Clicks on Ads at Facebook, Xinran He