TalkingData AdTracking Fraud Detection

SeaWIDS Kaggle Team 3
Sonia Arora
Janet Carson
Leila Norouzi
Laura Kehrl - Mentor

Background

- TalkingData

    Chinese data platform covering 75% of mobile devices nationwide

- 3 billion clicks per day

    90% or more are potentially fraudulent

    7.5+ GB containing 184,903,890 rows

- Kaggle Challenge: Look for the legitimate customers

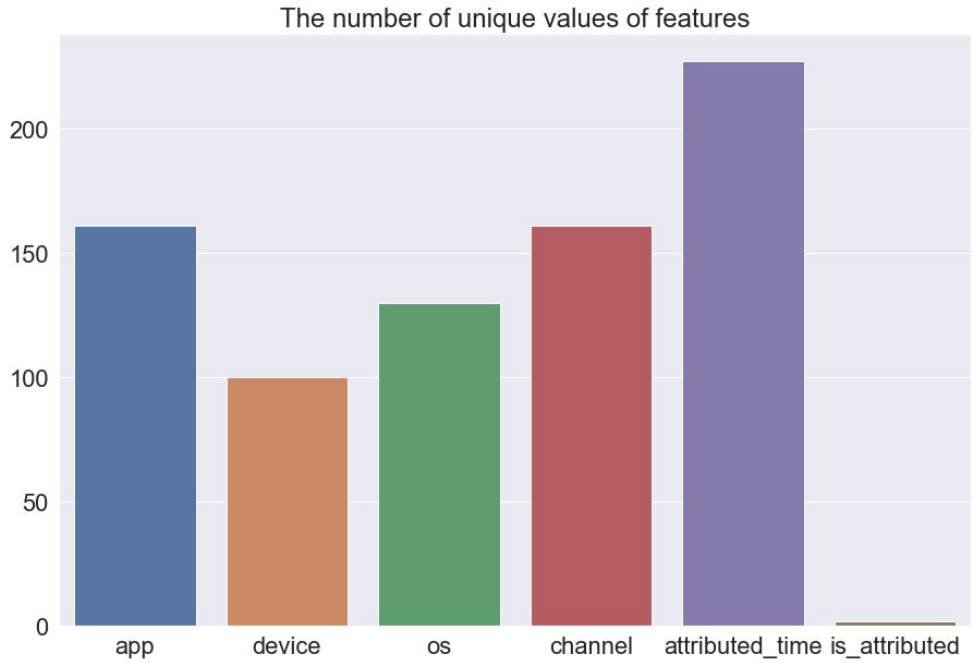
    Predict whether a user will download an app after clicking an ad for it

- IP
- OS
- Device

	ip	app	device	os	channel	click_time	attributed_time	is_attributed
0	87540	12	1	13	497	2017-11-07 09:30:38	NaN	0
1	105560	25	1	17	259	2017-11-07 13:40:27	NaN	0
2	101424	12	1	19	212	2017-11-07 18:05:24	NaN	0

- IP
- OS
- Device

Number of Unique Values per Category

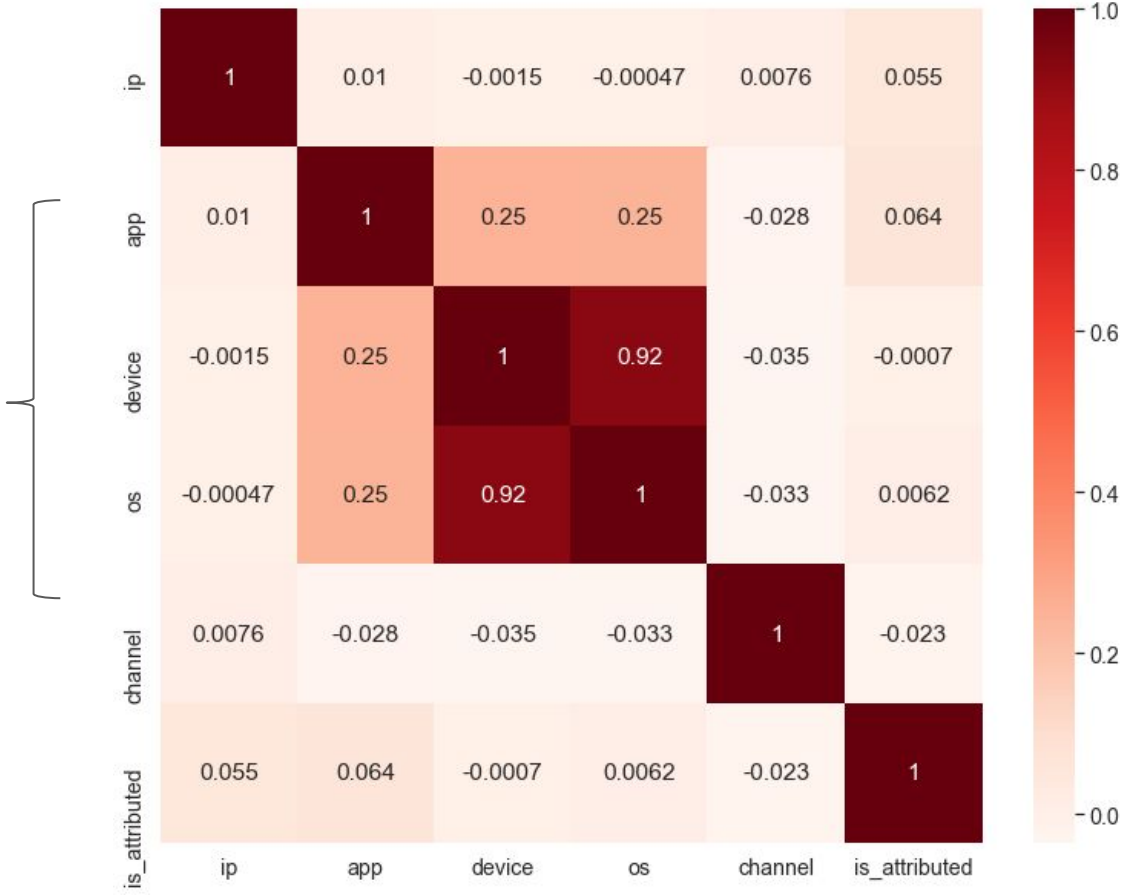


The unique number of values of features in sample data set

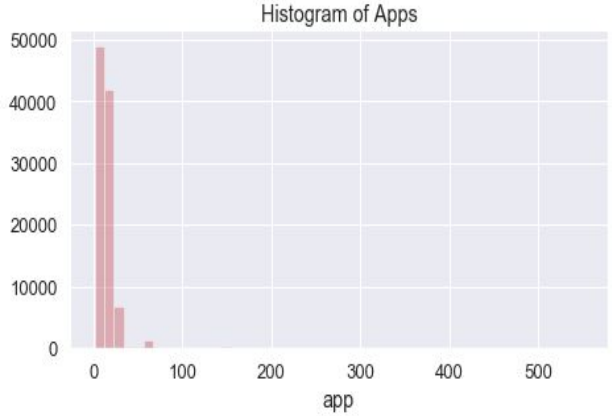
IP unique numbers in sample data set: 34587

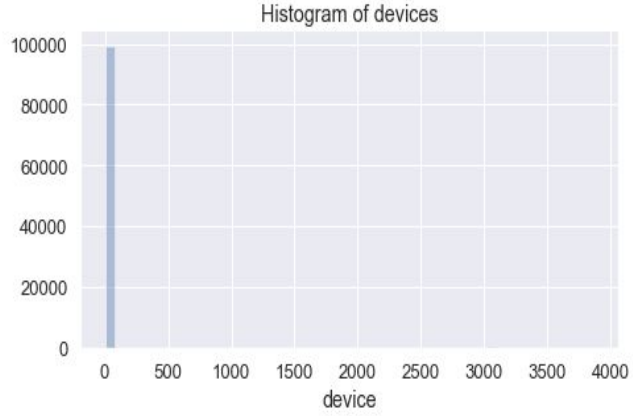
New IP addresses each day

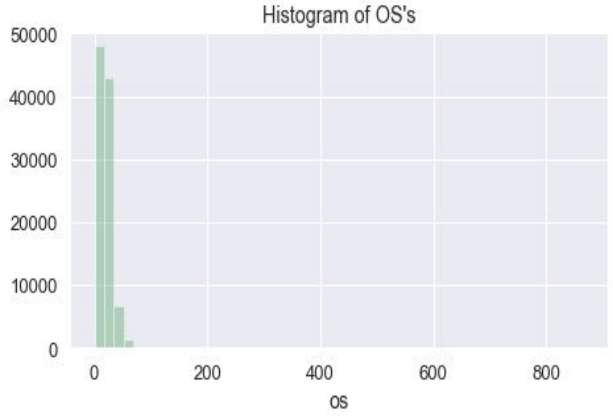
Associations between features

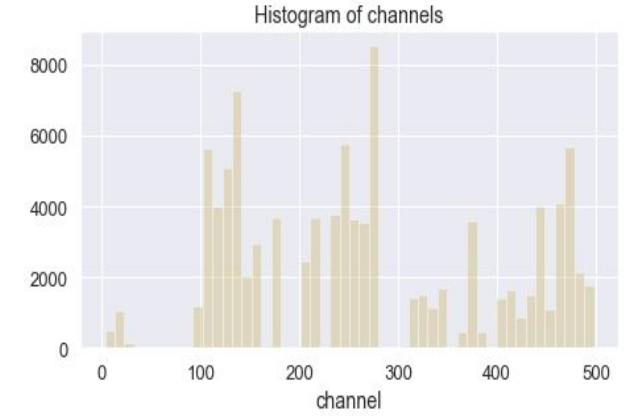


Feature values are unevenly distributed

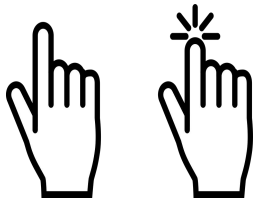






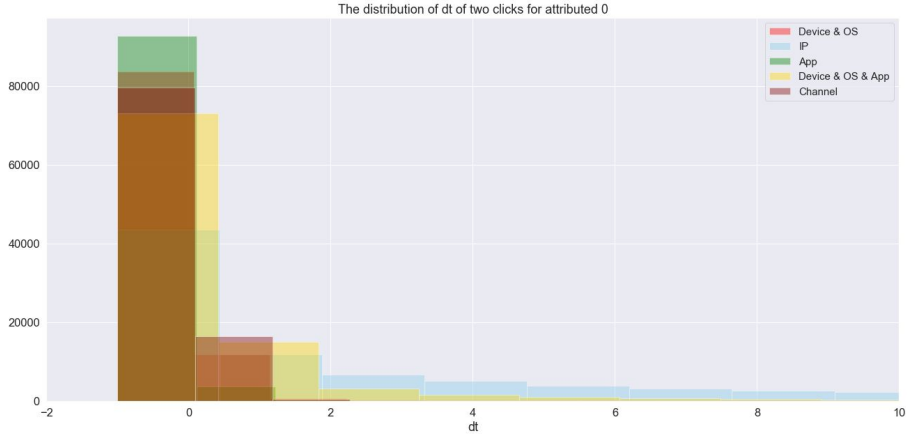


Feature engineering approaches: 1- clickers

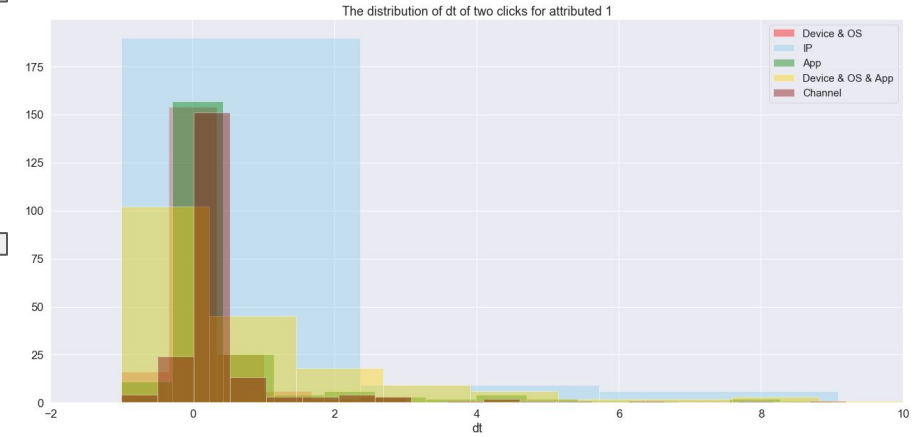


Click time

Defining the time difference between two consecutive click in each group of features



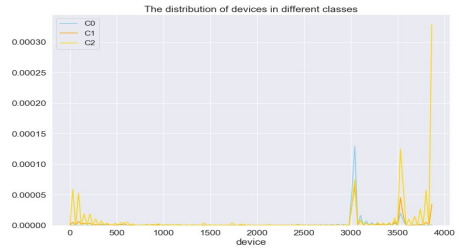
Clustering based on the time difference of two clicks Using GMM

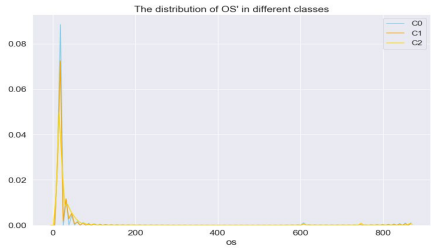


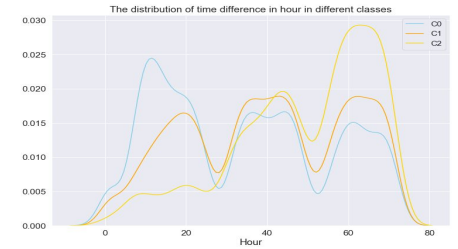
Defining min time as a indication of fraudulent click:
dt_{min}(app): 1.08 sec

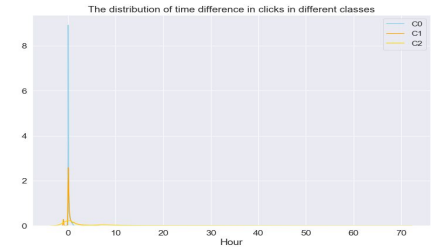
Fraud: $dt_{min}(app) < 1.08$

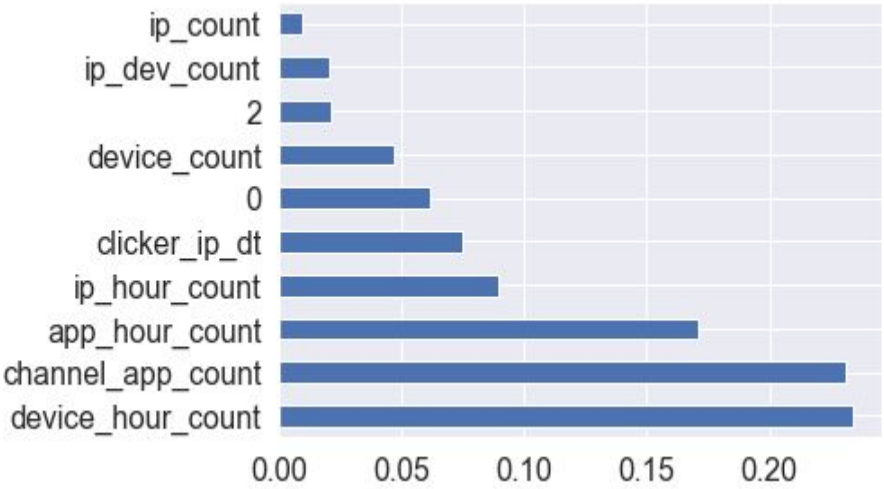
Feature selection











Normalized Value Counts

1. Count up appearances of each category

   2 orange, 2 gray, 1 yellow, 1 blue, 1 green...

2. Divide by the largest count found

   Orange -> 2/2, Yellow -> 1/2 etc.

3. Calculated over training set, database merge to test set

Except : IP addresses, because they come and go over time, were binned and recalculated each day

Normalized Value Counts

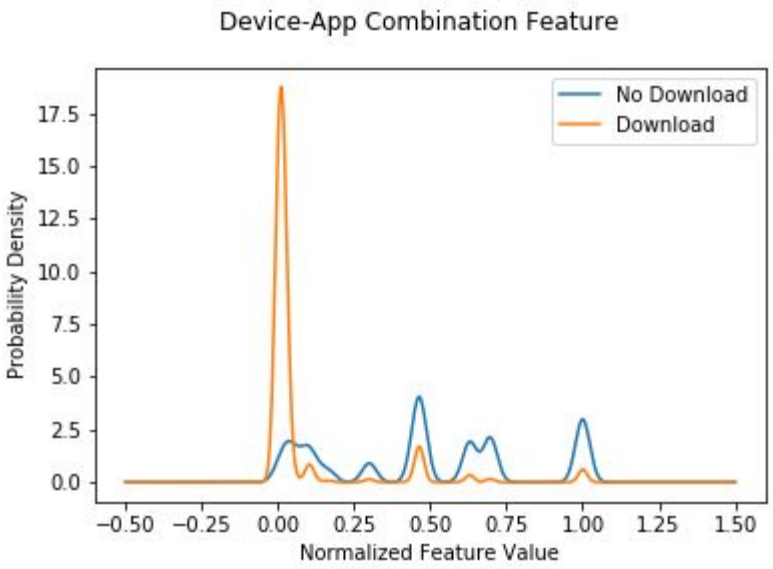
We’re training on whether it’s a popular or long-tail app/channel/etc, not on a specific value

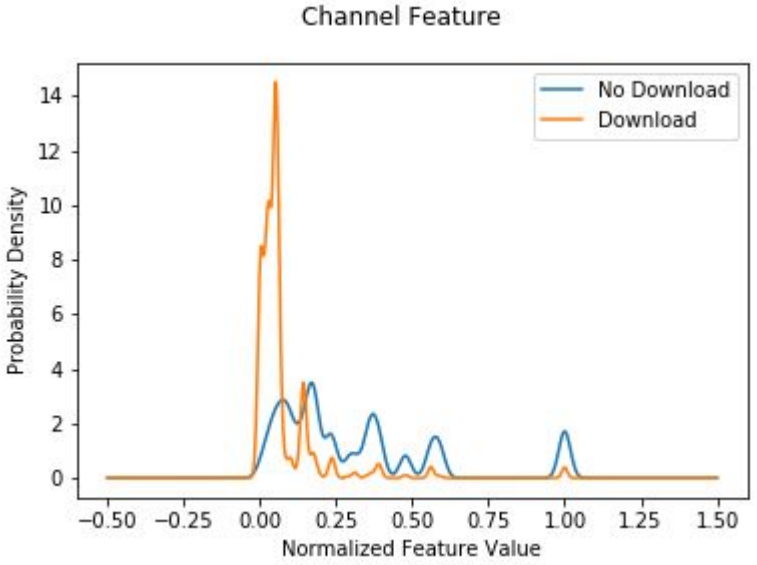
Normalized to adjust to different development stages

We found two useful recurring pairs of features:

Device + App
Channel + App

Feature Engineering - Two Examples





Model Building

PySpark - Spark.ML library

GBTree model

Class imbalance

- Downsampled majority class (2.7 percent of class “no download”)
- Upsampled minority by 4 times (bootstrap with replacement)
- 7 million rows per model - three different random seeds

Train Validation Split on each of the three sets

Results

Train Set:

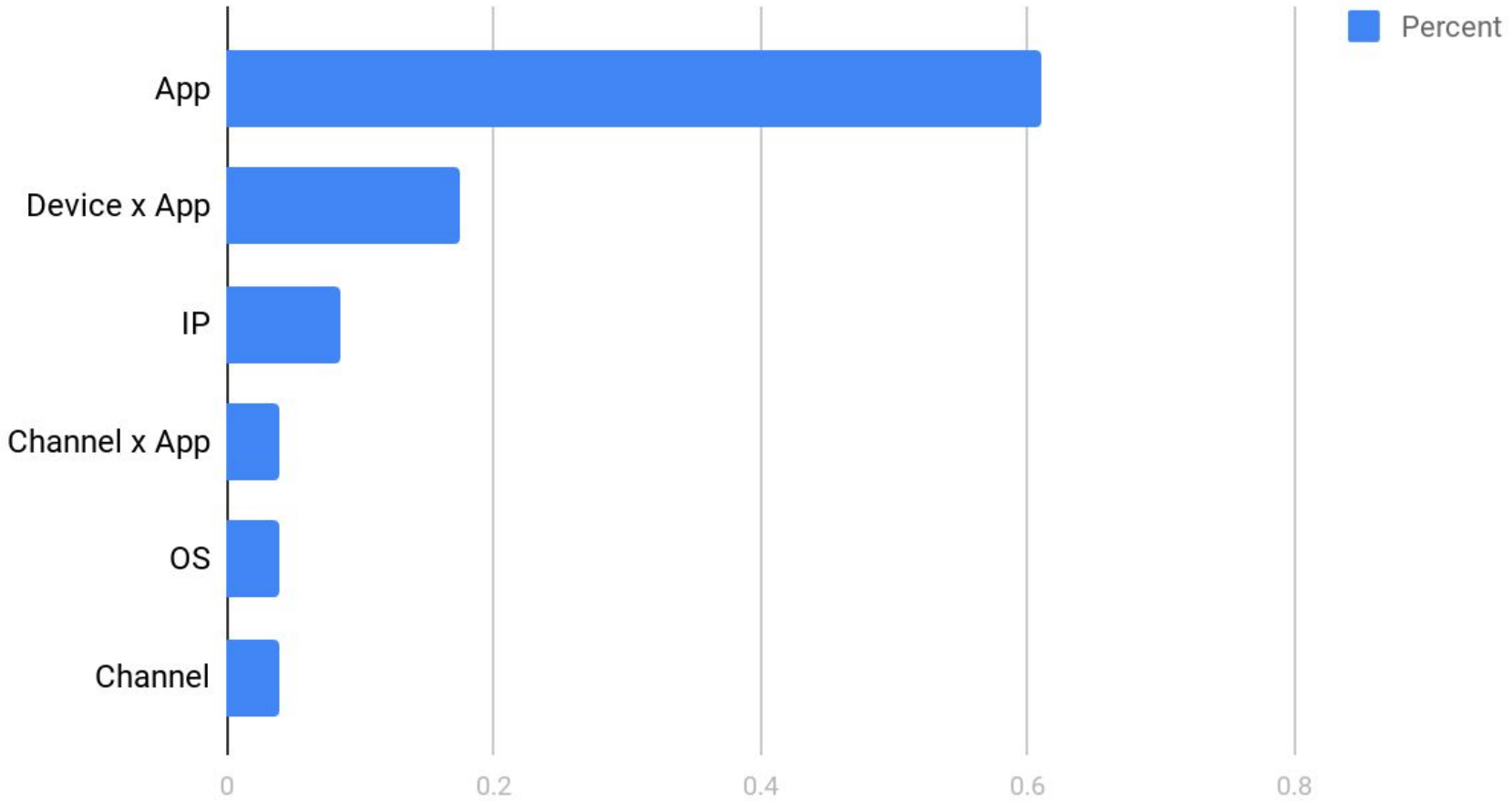
Area under ROC:	.97104	Precision:	11.9%
Accuracy:	98.4%	Recall:	85.7%

Test Set:

Area under ROC:

Our best model (private score)	0.96586
Kaggle winner:	0.98349

Feature Importance



Discussion

SEAWIDS Team 3 Model

Six features

Downsampled training sets

Average of three random sets

15 submissions

Kaggle Competition Winner

“Hundreds” of features

Even more highly downsampled

Average of five random sets

106 submissions

Thank You!