Machine Learning Methods on Detecting Fraudulent Click Traffic for Mobile App Ads

Cheng Wang

April 26, 2018

SCHOOL OF INFORMATICS AND COMPUTING

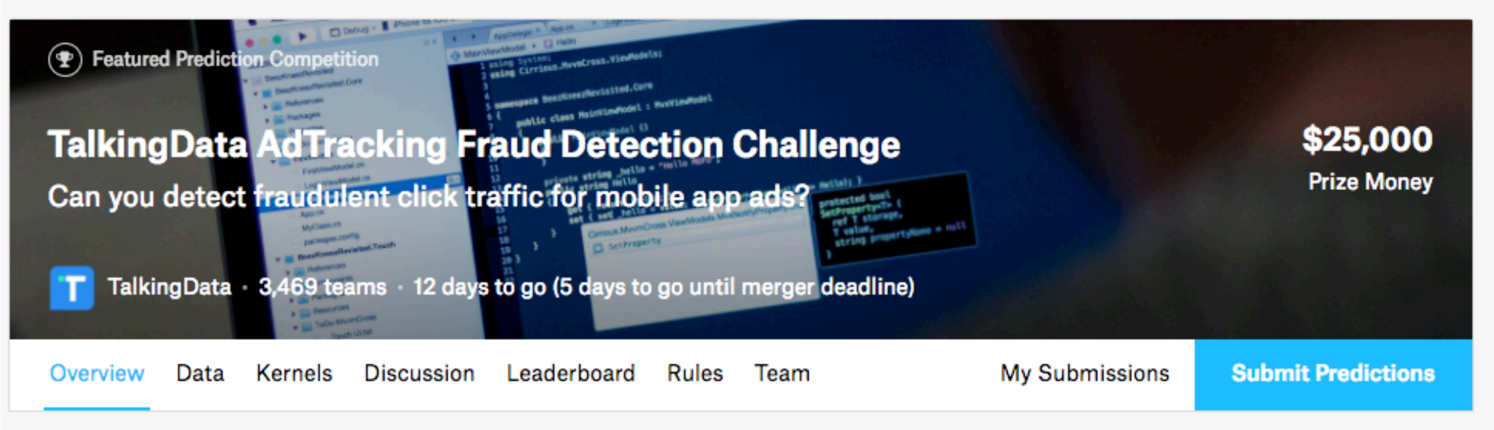
INDIANA UNIVERSITY

Outline

- Background & Introduction
- Exploratory Data Analysis (EDA)
- Machine Learning Model Pipeline
- Results, Discussion, Future Directions

Background & Introduction

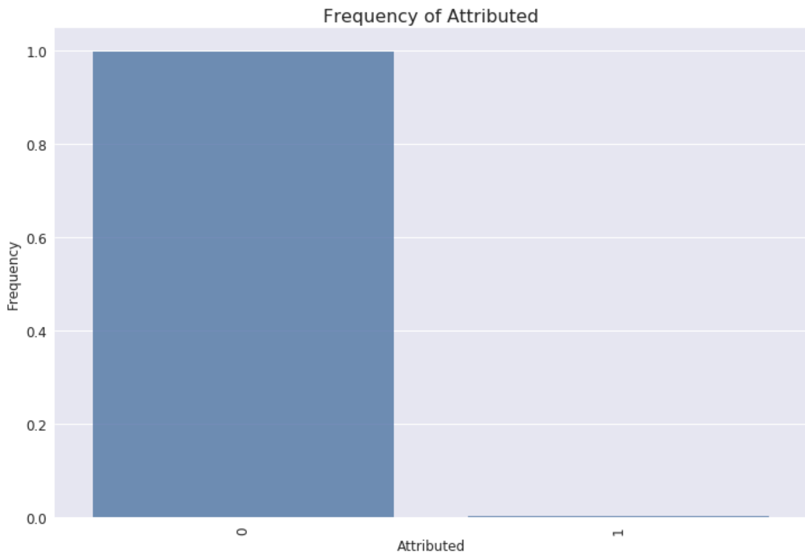
Fraud risk is everywhere, but for companies that advertise online, click fraud can happen at an overwhelming volume, resulting in misleading click data and wasted money.

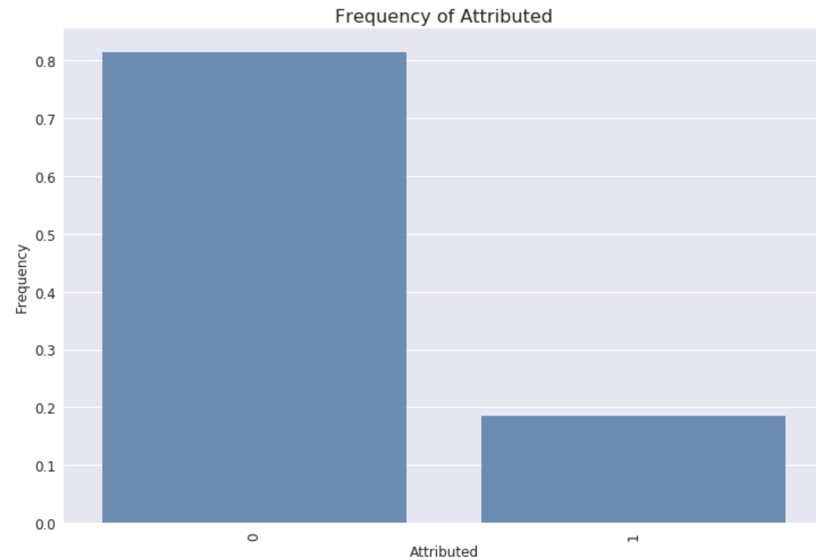


In this project, we develop the solution one step further by building an efficient machine learning algorithm that predicts whether a user will download an app after clicking a mobile app ad.

Exploratory Data Analysis (EDA)

Dealing with imbalanced data by random over sampling.





0.99773 : 0.00227 0.81432 : 0.18568

Variable correlation coefficient

"ip", "app" and "channel" have highest correlation with "is_attributed", i.e, the target prediction class.

Machine Learning Model Pipeline

Goal : predict class "is_attributed" based on a series of features:

Feature list: numerical variables
"ip": ip address of click, "app": app id for marketing, "device": device type id of user mobile phone , "os": os version id of user mobile phone, "channel": channel id of mobile ad publisher, "click_time": timestamp of click (UTC).

Data Preprocessing

- Raw data:
  80% training
  20% test

After balancing data

- Split 80% training
  80% training
  20% test
- Balancing data
- Scaling data

Feature selection

- Univariate Selection
- Recursive Feature Elimination
- Principal Component Analysis

ML Classifier Model

- Baseline model (SGDClassifier)
- SVM
- Logistic Regression
- Random Forest
- KNN
- Ensemble Learning

Performance Evaluation

- Prediction accuracy
- Precision score
- Recall score
- F1- score
- Confusion matrix

Results & Discussion

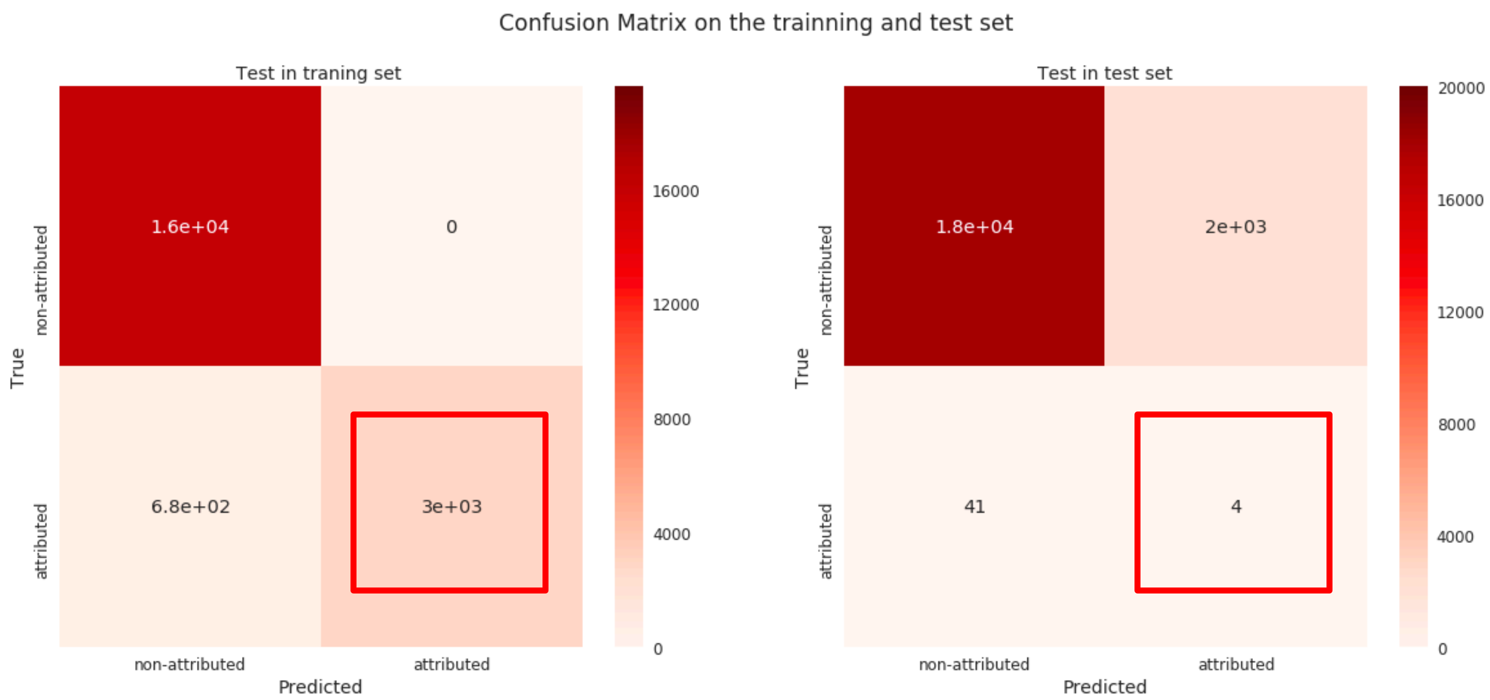
Table 1. Prediction performance of ML models

	ExpID	Data Description	Accuracy	Precision	Recall	F1 score
0	Logistic Regression	Test set for training	0.998000	0.996000	0.996000	0.996000
1	Logistic Regression	Real Test set	0.753000	0.002000	0.244000	0.004000
2	Random Forest	Test set for training	0.971000	1.000000	0.843000	0.915000
3	Random Forest	Real Test set	0.880000	0.002000	0.089000	0.003000
4	Support Vector Machine	Test set for training	0.998000	0.996000	0.993000	0.995000
5	Support Vector Machine	Real Test set	0.798000	0.002000	0.178000	0.004000
6	K Nearest Neighbor	Test set for training	0.998000	1.000000	0.991000	0.995000
7	K Nearest Neighbor	Real Test set	0.855000	0.002000	0.156000	0.005000
8	Voting Ensemble	Test set for training	0.998521	0.998344	0.993681	0.996007
9	Voting Ensemble	Real Test set	0.795350	0.001969	0.177778	0.003894
10	Boosting Ensemble	Test set for training	0.965364	1.000000	0.813462	0.897137
11	Boosting Ensemble	Real Test set	0.898800	0.002013	0.088889	0.003937
12	Bagging Ensemble	Test set for training	0.972710	0.999357	0.853571	0.920729
13	Bagging Ensemble	Real Test set	0.867700	0.002296	0.133333	0.004515

1. High prediction accuracy on both balanced test set and real test set
2. High F1-score on balanced test set, low F1-score on real test set

Results & Discussion

Result from Boosting Algorithms model (Stochastic Gradient Boosting)



1. Large number of records are predicted accurately on balanced test set.
2. Low number of records are predicted accurately on real test set.

Future Directions

- Implement other sampling techniques

  E.g., synthetic minority oversampling technique (SMOTE)

- Improve the feature selection method

  E.g., LASSO regression

- Hyperparameter tuning on the ensemble learning models

Thank you for your attention !