Mobile App Ads Click Fraud Detection

Zili Chen, Chen Liang, Jinhao Jiang, Bowen Yang and Jingyang Sui

Summary

Our project solves the problem of detecting fraud clicks on mobile app advertisements which induce clicks but do not end up installing apps and visualizes the top suspicious fraud clicks.

Motivation

The project aims to serve the app developers or companies who pay high advertisement fee for fraudulent clicks.

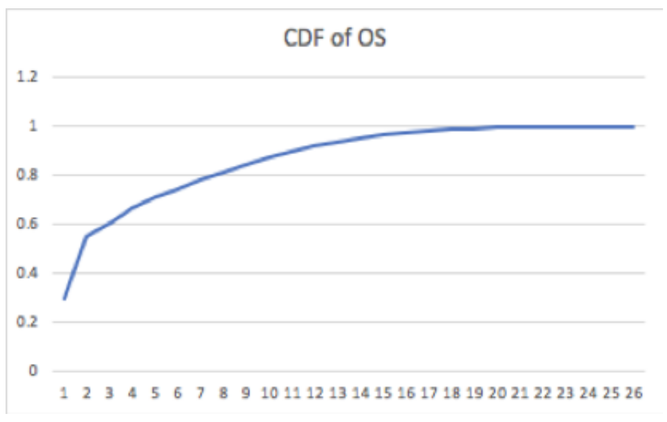
Approaches

Data Preprocessing

- Remove incorrect and abnormal values.
- Data unification.
- Convert data type.

Data Analysis

- Conversion rate doesn't have strong relationship with the IP frequencies.
- Fraud clicks mainly generated by certain small range of app, device, os and channels. Here we display the CDF of OS as an example:



Gradient Boosting Decision Trees (GBDT)

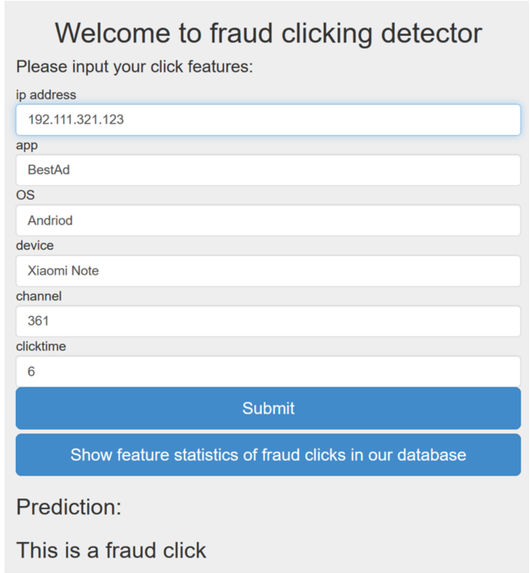
We used GBDT to train model with sigmoid objective, which built an ensemble of decision trees stage-wisely.

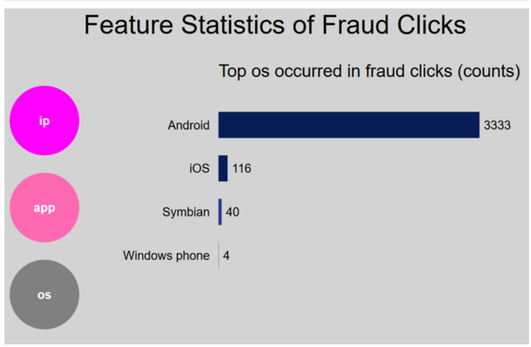
Innovations

- Using negative down-sampling method to deal with imbalanced datasets and improving testing AUC from 0.84 to 0.93.
- Using XGboost built-in functions to explore deeper tree branches and reduce overfitting by applying regularizations

Visualization

Submit features of the click and our model will predict whether it is a fraud click.
Top suspicious fraud clicks for each feature.





Data

Dataset downloaded from Kaggle. There are 100,000 rows (observations) by 8 columns (features).

Experiments & Results

The evaluation metrics is AUC, which measures the two-dimensional area underneath the entire Receiver Operating Characteristic (ROC) curve. ROC is a graph showing True Positive Rate (TPR) vs. False Positive Rate (FPR) of a classification.
We used GBDT to obtain an AUC of 0.94 from training dataset and an average AUC of 0.93 from 5 different test sets.

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{FP + TN}$$

Compared with random forest model, the training AUC and testing AUC respectively decrease to 0.89 and 0.78.