Applied ML Final Project Proposal - Team 21

Jace Yang, Kevia Qu, Sarosh Sopariwalla, Yu-Chieh Chen, Yunzhe Zhang

Background and Context

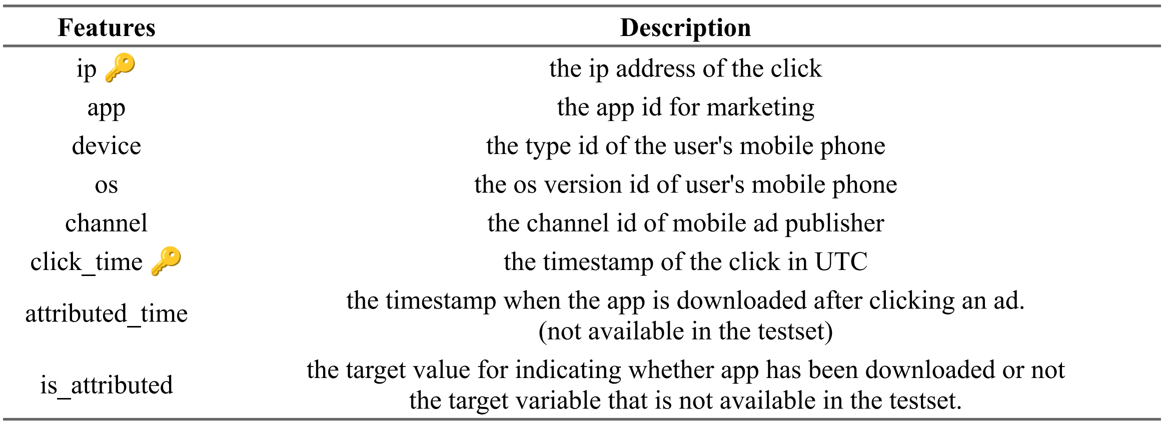
Many app-driven companies advertise their products and services through mobile ads. To pick an advertisement agency to develop and deploy their ads, companies ask about the proportion of users that download the app after seeing an ad. Hoping to impress businesses, many advertising agencies commit "click fraud" to pad their statistics.

In particular, these agencies may have a bot click the ad hundreds of times without ever downloading anything. Even worse, sometimes advertisers make the exit button for the ad so small that users accidentally click the ad when they did not mean to. (This marketing may even dissuade users from downloading an app with such a bothersome ad.) To prevent click fraud, TalkingData, a big data company, has been using brute force to ban users with suspicious click patterns. In particular, TalkingData measures the journey of each user's clicks; if they find an IP address with many clicks but very few installs, they will blacklist these IP addresses and devices.

While this method has worked so far, it would be much more efficient if we could predict the likelihood of a user downloading an app after clicking on the ad. If this likelihood is very low, we can blacklist these users and have advertising agencies focus on those who are more likely to download apps in response to our advertisements.

Description of the Dataset

The dataset was obtained from the TalkingData AdTracking Fraud Detection Challenge host by kaggle, and it covers approximately 200 million clicks over 4 days with 8 variables:



Features	Description
ip 🔑	the ip address of the click
app	the app id for marketing
device	the type id of the user's mobile phone
os	the os version id of user's mobile phone
channel	the channel id of mobile ad publisher
click_time 🔑	the timestamp of the click in UTC
attributed_time	the timestamp when the app is downloaded after clicking an ad. (not available in the testset)
is_attributed	the target value for indicating whether app has been downloaded or not the target variable that is not available in the testset.

Thus, we can use the timestamps and ips of a specific user to track their download (or not) activity among channels in a specific time slot. However, this data is extremely unbalanced with only 18,717 out of the 10,000,000 clicks that resulted in a download (less than 0.2%). Other artifacts include certain ip addresses appearing multiple times in the data, emulating the bot-like behavior.

ML Techniques

Determining whether a click is "fraudulent" or not can be framed as a binary classification problem. Tree-based ensemble modelling techniques, such as Random Forest, can be used, as well as boosting methods, such as XGBoost and LightGBM, to handle the highly biased data. Similarly, variations of SVMs will be attempted, such as using soft-margin classifiers and kernelized SVMs. Given that the data is highly imbalanced, other considerations during model training will include methods to resample the data (such as bagging) and possibly expand dimensions to improve model performance on the ROC-AUC evaluation metric.

Application of these ML techniques on this large and unbalanced dataset may help advertisers stay one step ahead of ad fraudsters and by helping the advertising agencies identify and focus on those who are more likely to download apps in response to our advertisements.