Machine Learning Engineer Nanodegree

Capstone Proposal

Gary Yang
Oct 15, 2018

TalkingData AdTracking Fraud Detection Challenge

https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection

Proposal:

With over 1 billion smart mobile devices in active use every month, it's no wonder fraud risk is now everywhere. Fraud risk is one of the most impactful risk that needs to be tackle with the ever growing mobile industry. Symantec recently has found 68 fraudulent apps by five different developers that contain aggressive advertisements. Users who download these apps end up wasting time watching advertisement. We can see that with how many mobile devices are out there that mobile frauds are having a serious impact on the mobile industry.

In a way this is a personal interest and motivation as I want to learn for about fraud detection. It's always going to be an ongoing issue that needs to be tackle. This is also a topic that I tend to have work with a little more at my workplace but would like to expand more upon to improve myself.

Source: https://www.symantec.com/blogs/threat-intelligence/apps-containing-aggressive-adware-found-google-play, https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection

Problem Statement:

TalkingData, China's largest independent big data service platform, covers over 70% of active mobile devices nationwide. They handle 3 billion clicks per day, of which 90% are potentially fraudulent. Their current approach to prevent click fraud for app developers is to measure the journey of a user's click across their portfolio, and flag IP addresses who produce lots of clicks, but never end up installing apps. With this information, they've built an IP blacklist and device blacklist. (https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection)

Datasets and Inputs:

Kaggle has provided a generous dataset covering approximately 200 million clicks over 4 days!

The training data contains millions of row of data consisting of (184,903,890):

- *ip: ip address of click.*
- *app: app id for marketing.*
- *device: device type id of user mobile phone (e.g., iphone 6 plus, iphone 7, huawei mate 7, etc.)*
- *os: os version id of user mobile phone*
- *channel: channel id of mobile ad publisher*
- *click_time: timestamp of click (UTC)*
- *attributed_time: if user download the app for after clicking an ad, this is the time of the app download*
- *is_attributed: the target that is to be predicted, indicating the app was downloaded (broken down to 2-class: app downloaded(1) and app not downloaded(0)) – Currently the data is heavily imbalance as the distribution leans towards an app not downloaded. Taking a quick look at the sample training data we can see that out of 100,000 samples only 227 shows that an app was downloaded vs 99,773 non-downloaded.*

The testing data is similar but contains two more rows (18,790,469):

- *click_id: reference for making predictions*
- *is_attributed: not included*

The test data is not labelled, grading is done by uploading the file containing the download probability of each click in the test data to Kaggle.

This data is well-prepared and processed. All features are labeled, missing data had been filter out or dropped.

Solution Statement:

To start of I would have to use pandas to do some exploratory data analysis. Afterwards, I believe the best approach would be using a classification model to determine if a click would in terms download an app into said mobile device. In terms of choosing a model, on the discussion board on kaggle(https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/discussion), I will attempt to use XGBoost which is an open-source software library which provides a Gradient Boosting framework. (https://xgboost.readthedocs.io/en/latest/) Its designed to be highly efficient, flexible, and portable that seems be used quite often in the Kaggle community.

Benchmark Model:

The benchmark a classifying model that determines whether a click on an ad would download an app. This would be a binary classification task as we would like to see if an app is downloaded or not. We also want to see if the results yields a Receiver operating characteristic score of atleast 0.3892528 (https://en.wikipedia.org/wiki/Receiver_operating_characteristic).

Evaluation Metrics:

Submissions to kaggle are evaluated on area under the ROC curve between the predicted probability and the observed target. (https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection#evaluation)

Project Design:

This data is well-prepared and processed. All features are labeled, missing data had been filter out or dropped.

1. EDA(exploratory data analysis) – I will first have to take data and look at how it is distributed.
2. Clean Data – I will then determine which features is key and how they relate to my goal.
3. Prepare Data – The data is pretty well-prepared, and since test and training data are already provided not much is needed to be done at this stage besides any processing if needed. Looking at the data we can see that click_time column is given to us in a yyyy-mm-dd hh-mm-ss format. I think that best approach would be to break this down even further to day, hour, and minute to better definite new features.
4. Create new features such as how often clicks by a user happens on certain time(holidays, different day of the week, etc) base on given features.
5. Fine tune hyperparameters.
6. Perform training
7. Report results.