Machine Learning Engineer Nanodegree

Capstone Proposal

Keyun Shang

March 11th, 2018

Proposal

Domain Background

For companies that advertise online, an overwhelming volume of click fraud can lead to misleading click data and wasted money.

Supervised learning is a method of machine learning that can learn or build a pattern from training data and use the pattern to predict the output for new data. Companies can use supervised learning to detect and identify click fraud patterns hidden in the big data.

Problem Statement

TalkingData is china's biggest independent data service platform and suffers from huge volumes of fraudulent traffic. Currently , they have built a blacklist for ip addresses and devices which could be the source of click fraud. Now their problem is to build an algorithm that predicts whether a user download an app after clicking a mobile app ad.

The link to the data source is below.
https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection

Datasets and Inputs

The dataset consists of the following parts.

- train.csv - the training set
- train_sample.csv - 100,000 randomly-selected rows of training data, to inspect data before downloading full set
- test.csv - the test set

The fields description for each file is below.

Each row of the training data contains a click record, with the following features.

- 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

Note that ip, app, device, os, and channel are encoded.

The test data is similar, with the following differences:

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

As this is a Kaggle competition project, the data is provided by Kaggle and TalkingData, they can be obtained here.

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

Note: We will not use test.csv because it doesn't include the label data. Instead, we will split the train.csv into train set and test set. Test.csv is used only for Kaggle evaluation, but as stated later in Evaluation Metrics section, in this study, we do not use Kaggle score for evaluation.

Solution Statement

First, we need to understand the relationship between the features as input variables and the target labels as output variables. Second, we will build a benchmark model as the start point for evaluation. Third, we will build and train several new models with several commonly used algorithms such as decision tree, Gassic NB, SVM etc. We will evaluate and choose the best model from them. Finally, We will optimize the best model with grid search.

Benchmark Model

As there is no available benchmark model for this project at present, we will use the naive predictor as the benchmark model. Naive predictor is a simplest predictor that predicts the data with current values. Any new model should be better than or at least the same with it.

Evaluation Metrics

As this is a Kaggle competition project, the best way for evaluation would be the Kaggle score for the test set. However, it is not appropriate in this study due to the following reasons:

1) The number of submissions per day is limited by Kaggle.
2) If the Kaggle project expires before we complete the study, we can no longer obtain Kaggle score.

For academic purpose, we will use F-beta score as the metric instead.

$$F_\beta = (1+\beta^2) \cdot ((precision \cdot recall) \div (\beta^2 \cdot precision + recall))$$

Project Design

The project will consist of the following parts.

1) Analyze the feature of the training data
2) Evaluate the data with the benchmark model to get an accuracy score
3) Build models with several commonly used algorithms such as decision tree, Guassic NB.etc.
4) Choose the best model through the evaluation
5) Optimize the best model with grid search
6) Draw a conclusion