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

Cheng Wang

**Abstract**

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. Ad channels can drive up costs by simply clicking on the ad at a large scale. TalkingData, China’s largest independent big data service platform, 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. To help develop their solution one step further, I join the Kaggle competition to build an efficient machine learning algorithm that predicts whether a user will download an app after clicking a mobile app ad. To support the modeling, TalkingData has provided a generous dataset covering approximately 200 million clicks.

**Description of the Dataset**

TalkingData provides the dataset including both the training set and test set, which is accessible from Kaggle. The dataset is stored as “.csv” file. Three files are included: “train.csv” refers to the full training set, “train\_sample.csv” refers to 100,000 randomly-selected rows of training data in order to inspect data before training on the full set, “test.csv” refers to the test set.

In the dataset, 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.

**Team**

The project can be teamed up with 2~3 teammates.

**Machine Learning Methods**

The task is to predict whether the app was downloaded based on user’s portfolio. Basically, this is a binary classification problem of supervised learning. I consider using several ML algorithms such as KNN, SVM, Logistic regression, Naïve Bayesian, etc. to find the best algorithm with optimal parameters.

**Result Evaluation**

The accuracy score and confusion matrix will be used to measure the success of the prediction.

**talking_data/gantt_diagram.pdfGantt Diagram**

**Machine Learning Pipeline**

1. Exploratory data analysis, visualize data to gain thoughts, including histogram plots, variable correlation plots.

2. Prepare the data, e.g., handling text and categorical input attributes, transform numerical data (standardizing, dealing with missing values).

3. Split the training data set into k folds for cross-validation and start building scikit-learn pipelines.

4. Apply ML algorithm (classifier, e.g., SGDClassifier, SVM, etc) to training data set, hyperparameter tuning (use grid search, random search CV), and feature selection to find the optimal classifier set for the training data set.

5. Apply the optimized ML algorithm (classifier) to the test data set, report the accuracy score and confusion matrix.

6. Summarize the report including the findings of investigating the data, the prediction accuracy of the test set, and the comparison of different models by performing significance test to test which ML pipeline is better.