CKME136 - Capstone Project
Literature Review and Data Descriptions

TalkingData AdTracking Fraud Detection Challenge
https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection

Phi Huynh - 500777278
https://github.com/PHLHY/Capstone-

Introduction

Click fraud is the act of generating fraudulent clicks on pay-per-click advertisements which results in artificially inflating web traffic and potential loss of revenue to the advertiser.

TalkingData provided a dataset of around 200 million clicks spanning over 4 days for a Kaggle challenge with the aim of building an algorithm for determination of users that will download a mobile application after clicking on a relevant advertisement. This challenge will be the main focus of the research question for this capstone project.

Techniques for classification analysis will be employed on the R platform (RStudio) to tackle this challenge.

Literature Review

A review of research articles and publications on the different areas of click fraud, important features for consideration, and classification algorithms is summarized below to give a comprehensive approach to the research question for this capstone project.

Click Fraud[5]

This article provides a better understanding of what click fraud is and its impact and implications. This is to aid with the terminologies and give background information so that it provides better insight to complete this capstone project. The author explains what click fraud is, the different types of click fraud, ways to detect click fraud, and potential solutions to combat click fraud.

Combating Online Fraud Attacks in Mobile Based Advertisement[2]

The article goes into further details about click fraud in a mobile environment which is relevant to this capstone project. It provided further knowledge on vulnerabilities of click frauds based on advertisement networks and provided suggestions on ways to combat click fraud for the mobile environment.

Classification and Regression Tree[6]

The author explained differences between classification and regression trees along with different algorithms on a car dataset. This would help me select and consider the kinds of algorithms that can help with modeling the data for the capstone project. The author, Loh, found that the GUIDE algorithm has the highest predictive accuracy.

Feature Engineering for Click Fraud Detections[8]

Phua et al. defined feature engineering as “extraction and selection of features” as a key component to click fraud detection to maximize performances of models. While Phua’s team was working on banking data, they found that a relatively large number of clicks, rapid duplicate clicks, or a high percentage of clicks from high-risk countries as important fraud indicators. They used classification techniques such as Gradient Boosted Regression Models (GBM) and Random Forest in R and RIPPER from WEKA as their algorithms for click fraud detection.

Detecting Click Fraud in Online Advertisement: A Data Mining Approach[7]

Oentaryo et al. analyze data mining approaches from different competitors in a competition on real-world fraud data for online advertisements provided by BuzzCity Pte. Ltd. The authors summarize the approach from the top three winners, the runner-up and from the organizer. Those include how each team did their pre-processing/feature extractions, their methods, which classification algorithms they used, and the performances of their models.

Learning From Automatically Labeled Data: Case Study on Click Fraud Prevention[1]

The main focus of classification algorithms is to enhance the model’s performance, however, Berrar advised that when doing supervised learning that we should be wary of automatic class labels influencing our results. If there is a discrepancy between our models and the testing model, then we need to check the original results and consider how the dataset originally decide whether a click was fraudulent as it most likely came from a fraud detection algorithm. What is interesting to note is that Berrar used exclusively an ensemble approach of Random Forests for the algorithm as the author found better model performance based on the author earlier studies.

An Ensemble Learning Based Approach for Impression Fraud Detection in Mobile Advertisement[4]

Haider et al. explained their approach on European datasets based on mobile advertisements in fraud detection. They were able to make their model's performance at 99.32% accuracy, 96.29% precision, and 84.75% for recall. For algorithms, the authors examined Decision Tree Classifier J48, Random Forest, and REPTree along with ensemble learning techniques (bagging and boosting).

An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization[3]

The usage of the ensemble approach for classification algorithms has been a theme in the previous articles. A further understanding of these methods will be helpful for consideration of use for the capstone project. This article explains the differences between bagging, boosting and randomization on the Decision Tree algorithm C4.5. Depending on if there is classification noise (incorrect class labels), the bagging method is more proper to use compared to boosting which usually provides the best result in low noise.

Dataset

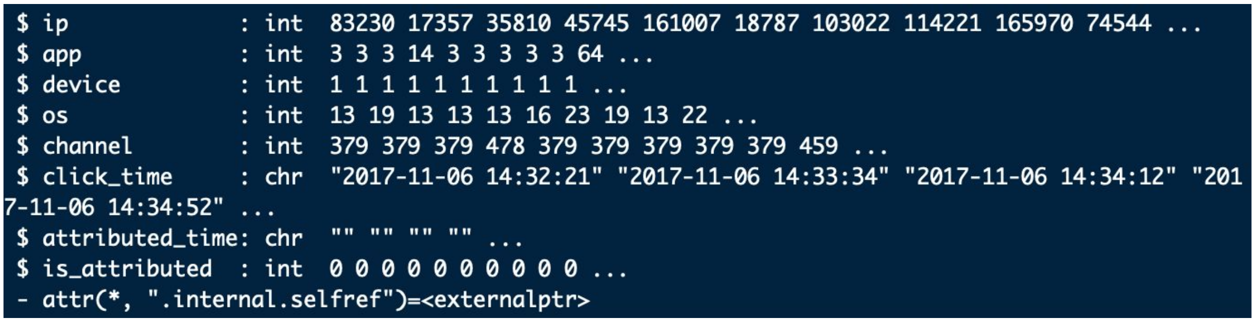
The datasets used for this capstone project are from a Kaggle competition “TalkingData AdTracking Fraud Detection Challenge” found from this link: “*https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection”.*

Training and testing datasets are provided by TalkingData for this competition. In total, there are over 200 million clicks that were captured over a 4 days span. The training set includes 184903890 data points with 8 attributes and the testing set includes 18790469 data points with 7 attributes.

Attributes of these datasets includes:
ip, app, device, os, channel, click_time, attributed_time, and is_attributed (target attribute)

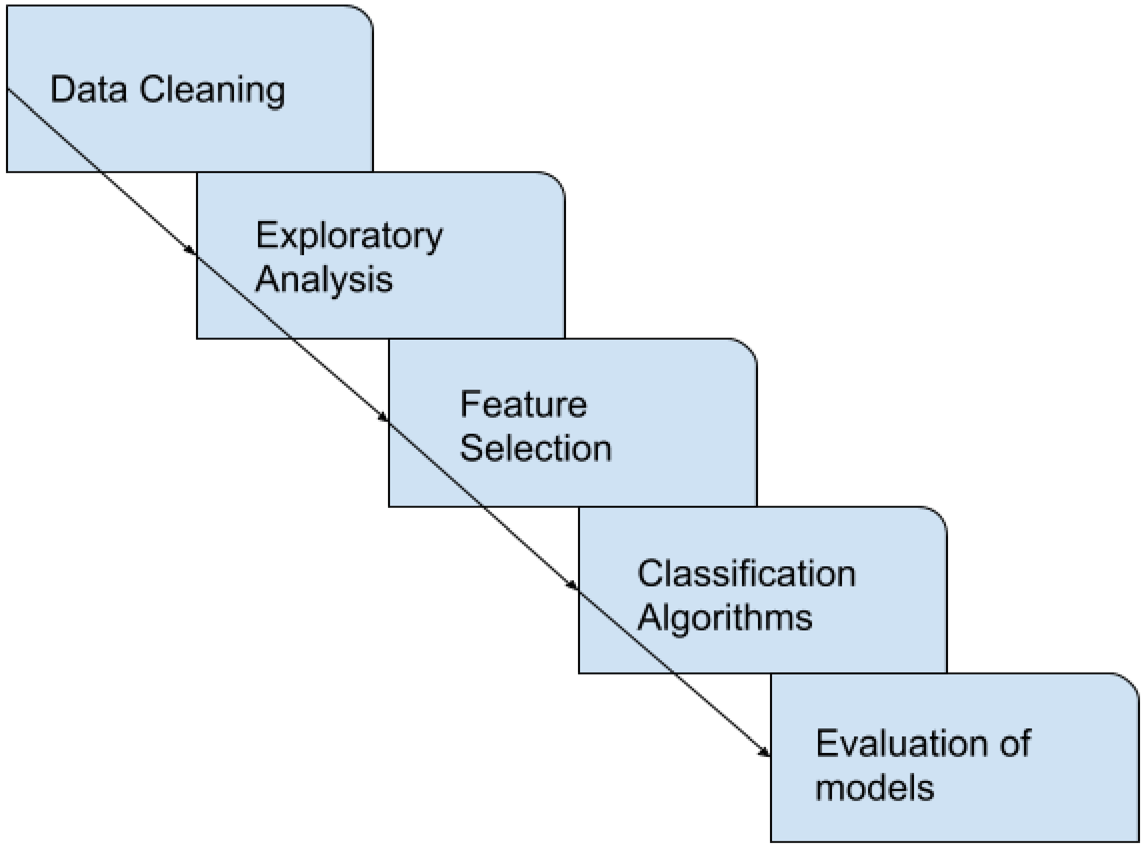
Description of the attributes were provided by TalkingData which is found on this link: “*https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data*”

A summary of the attributes is shown below:



```
 $ ip            : int  83230 17357 35810 45745 161007 18787 103022 114221 165970 74544 ...
 $ app           : int  3 3 3 14 3 3 3 3 3 64 ...
 $ device        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ os            : int  13 19 13 13 13 16 23 19 13 22 ...
 $ channel       : int  379 379 379 478 379 379 379 379 379 459 ...
 $ click_time    : chr  "2017-11-06 14:32:21" "2017-11-06 14:33:34" "2017-11-06 14:34:12" "201
7-11-06 14:34:52" ...
 $ attributed_time: chr  "" "" "" "" ...
 $ is_attributed  : int  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

Approach



Step 1: Data cleaning

- Datasets needs to be check for any missing values and for any potential outliers. It will have to be examined if those values can be imputed or removed without affecting the quality of the datasets.
- Taking a look at the click_time attribute and separating so that we can have it in a date and time format. Specifically, with the date component, since the data was collected over 4 days, likely the year and month can be removed from it

Step 2: Exploratory analysis

- Explore patterns within the data and answer questions set in the abstract to get a better appreciation of the data
    - Are there particular times in the days where a user will download an application?
    - Do fraudulent clicks have any relations to the types of mobile devices used?
    - Are there particular types of an application that a user will be more likely to download?
- At this stage, attributes should be tested against another in terms of correlations
- Checking if the dataset is balanced and any necessary adjustment needs to be made to order to balance the dataset
- Visualizing of the data in graphs

Step 3: Feature selection

- Examining which attributes should be included for the classification algorithms for optimizing the performance of the models. This can be looked at by techniques such as Feature Importance by Gain or Principle Component Analysis as indicated.

Step 4: Classification algorithms

- Classification algorithms such as Random Forest and others as necessary will be done to create models for our target variable “is_attributed”.
- Ensure that validation sets are considered

Step 5: Evaluation of models

- Confusion matrix and other evaluative measures are done to test the performance of the models

References

1. Berrar, Daniel. "Learning from automatically labeled data: case study on click fraud prediction." *Knowledge and Information Systems* 46.2 (2016): 477-490.
2. Cho, Geumhwan, et al. "Combating online fraud attacks in mobile-based advertising." *EURASIP Journal on Information Security* 2016.1 (2016): 2.
3. Dietterich, Thomas G. "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization." *Machine learning* 40.2 (2000): 139-157.
4. Haider, Ch Md Rakin, et al. "An ensemble learning based approach for impression fraud detection in mobile advertising." *Journal of Network and Computer Applications* 112 (2018): 126-141.
5. Jansen, Bernard J. "Click fraud." *Computer* 40.7 (2007): 85-86.
6. Loh, Wei‐Yin. "Classification and regression trees." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.1 (2011): 14-23.
7. Oentaryo, Richard, et al. "Detecting click fraud in online advertising: a data mining approach." *The Journal of Machine Learning Research* 15.1 (2014): 99-140.
8. Phua, Clifton, et al. "Feature engineering for click fraud detection." *ACML Workshop on Fraud Detection in Mobile Advertising*. 2012