



TalkingData AdTracking Fraud Detection Challenge

Shilpa Rajbhandari



Data Source and Goals

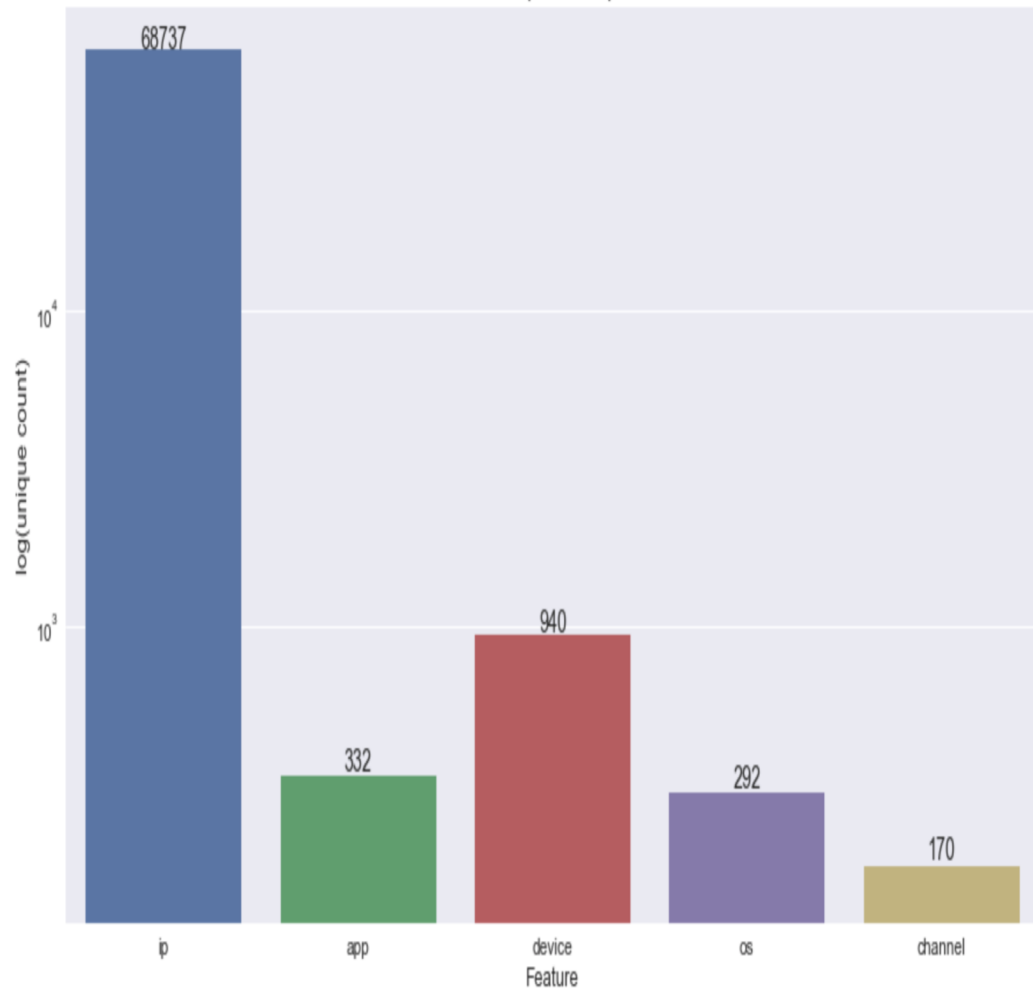
- Kaggle competition
(<https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data>)
- Predicts if the user will download an app after clicking in the mobile apps advertisement.



Project Data

- Train - 7 variables
- Test-5 variables
- Sample
- Target Variable:
 - `Is_attributed`(categorical)
- Predictor Variables:
 - IP
 - App
 - Device
 - OS
 - Channel
 - Click Time
 - Attributed time

Number of unique values per feature

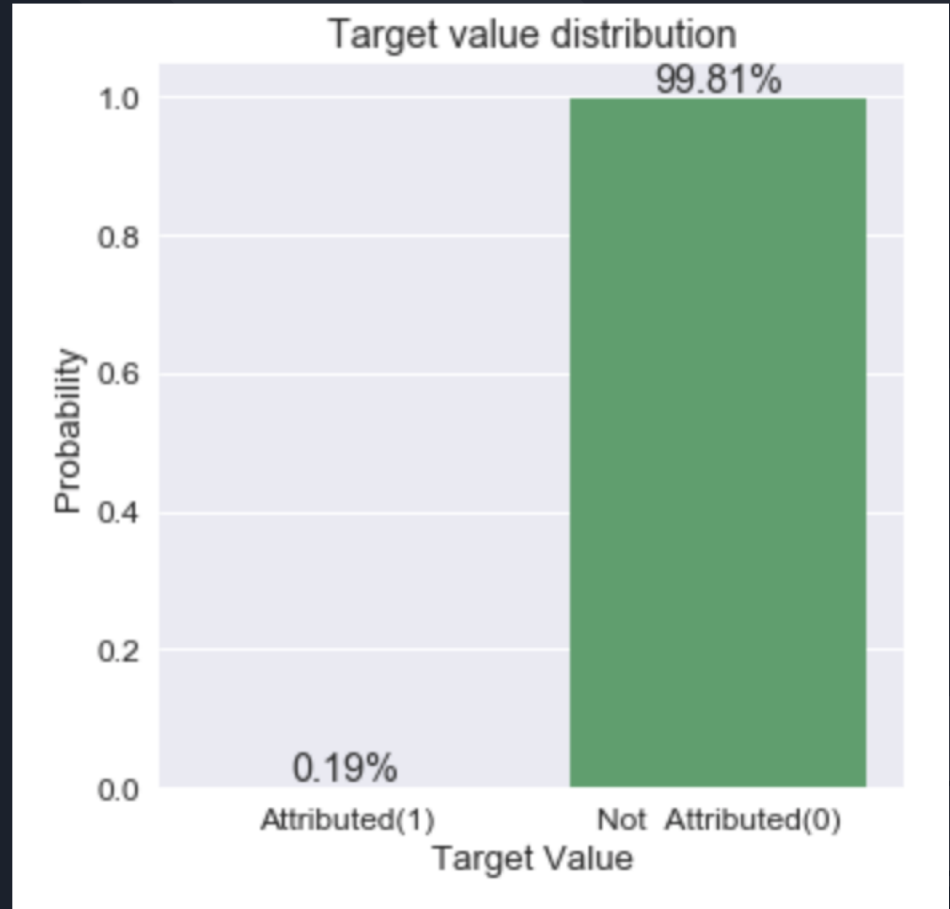


Exploratory Data Analysis

10mill rows, 250 skip rows

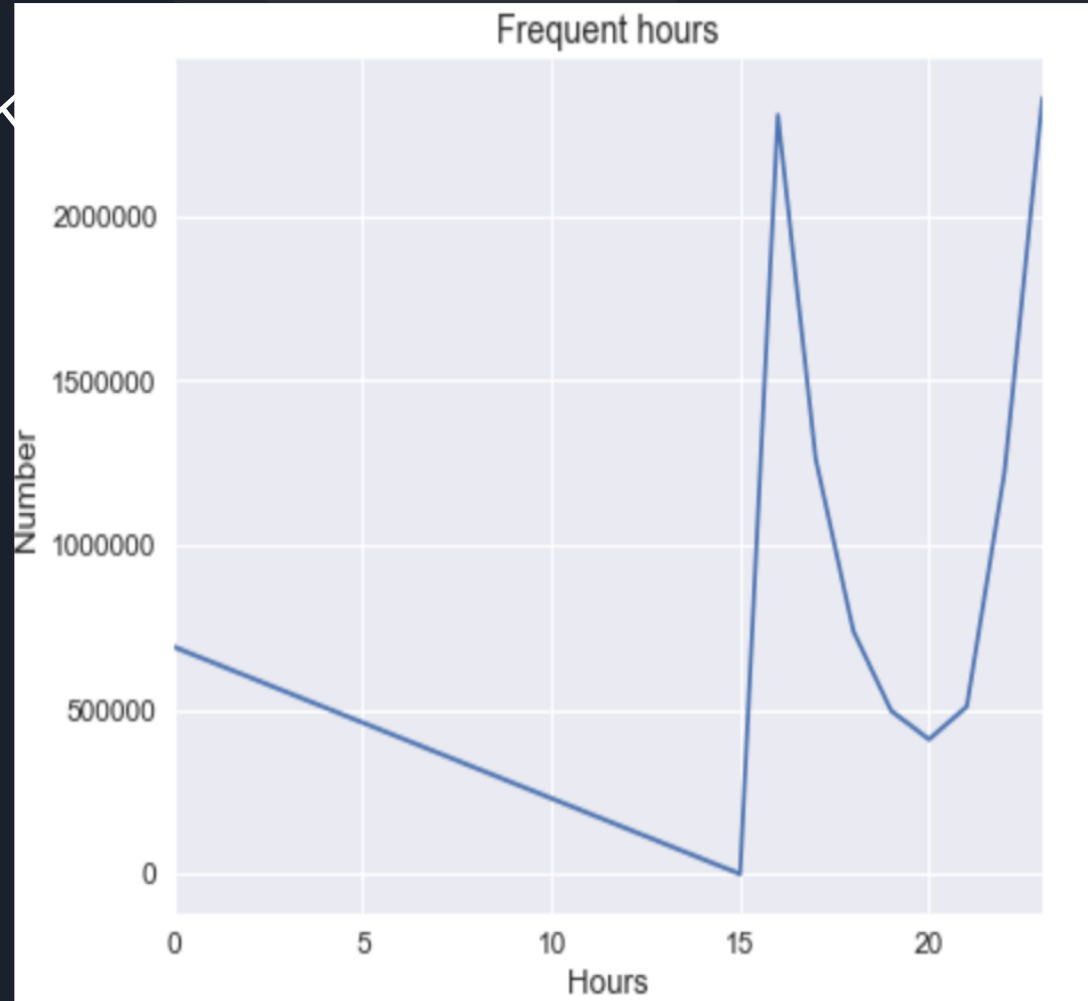
Target value distribution

- Only 0.19% has downloaded an app after clicking the mobile app
- We have a unbalanced dataset



Hourly distribution of click time

- From 12am-3pm, there is a decreasing trend.
- Highest number of click is at 4 pm





Feature Description

- ATTRIBUTION:

```
Calculating confidence-weighted rate for: ['os', 'device'].
```

```
Saving to: os_device_confRate. Group Max /Mean / Median / Min: 2348023 / 6472.49 / 3.0 / 1
```

- AGGREGATION:

- Average clicks on app by distinct users; is it an app they return to?
- How popular is the app or channel?

- NEXTCLICK:

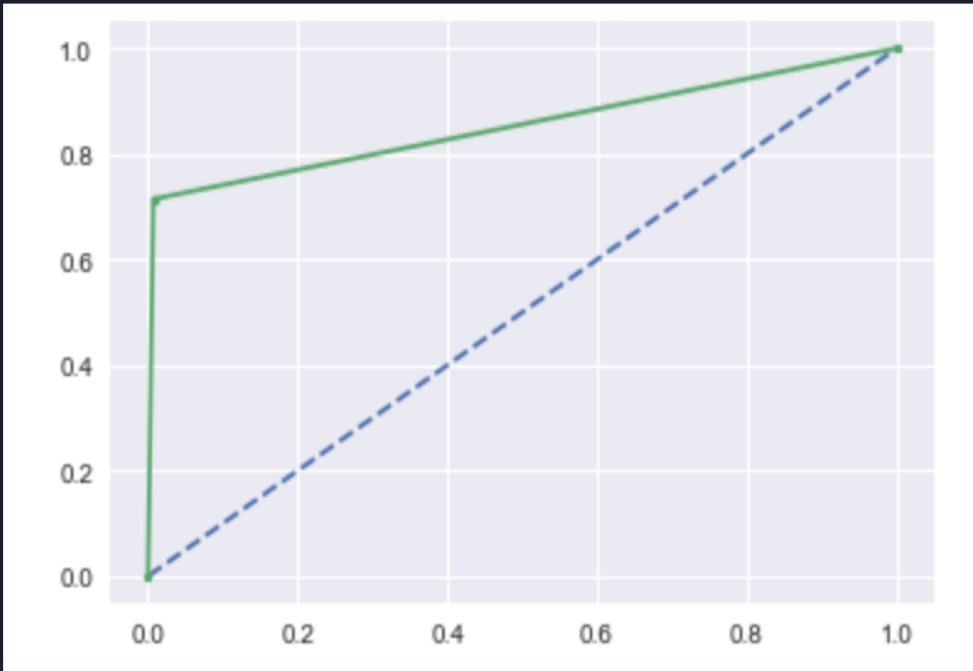
- Frequency of past and future in same ip, app and channel



XG Boost

- MODEL SPECIFICATION:
 - Gradient boosting decision algorithm
 - Training and Valid Accuracy increases in each iteration

```
[0]      train-auc:0.90349      valid-auc:0.879048
[10]     train-auc:0.938977     valid-auc:0.939491
[20]     train-auc:0.941291     valid-auc:0.940179
[30]     train-auc:0.943421     valid-auc:0.942232
[40]     train-auc:0.949386     valid-auc:0.942464
[50]     train-auc:0.954674     valid-auc:0.941487
[60]     train-auc:0.958978     valid-auc:0.940165
[70]     train-auc:0.961994     valid-auc:0.941022
[80]     train-auc:0.964812     valid-auc:0.942077
[90]     train-auc:0.967064     valid-auc:0.942405
[99]     train-auc:0.968163     valid-auc:0.942163
[102.00958800315857] Finish XGBoost Training
```

ROC Curve:

- Usefulness of true positive against false positive rate



Confusion Matrix

Accuracy Result/Confusion Matrix/:

- Accuracy: Overall, how often is the classifier correct? 99.275%
- Misclassification Rate: Overall, how often is it wrong? 0.746%.
- True Positive Rate: When it's actually yes, how often does it predict yes? 0.115%
- False Positive Rate: When it's actually no, how often does it predict yes? 0.70%
- True Negative Rate: When it's actually no, how often does it predict no? 99.299%

```
array([[99139, 700],  
       [ 46, 115]])
```




Random Forest

Standardization:

- Not necessary
 - Robust to numerical instabilities due to partitioning rules that wouldn't change with scaling

Model Specifications:

- Number of trees
 - Ideally select largest amount of trees your computer can handle

- 
- The model predicts 0.99808 accuracy.
 - Random_state = 99, n_estimator=100

Accuracy: 0.99808



Best Classifier

XGBoost and Random Forest model predicts download app best

Accuracy Score:

- XGBoost= 0.968
- RF=0.99808



Conclusions

- If non encoded features were provided, the model would explain better result and helps in tracking fraud click.
- Hour 16, 17, 22, 23 has the highest click time of the day.
- Need more computation power to run the whole dataset.