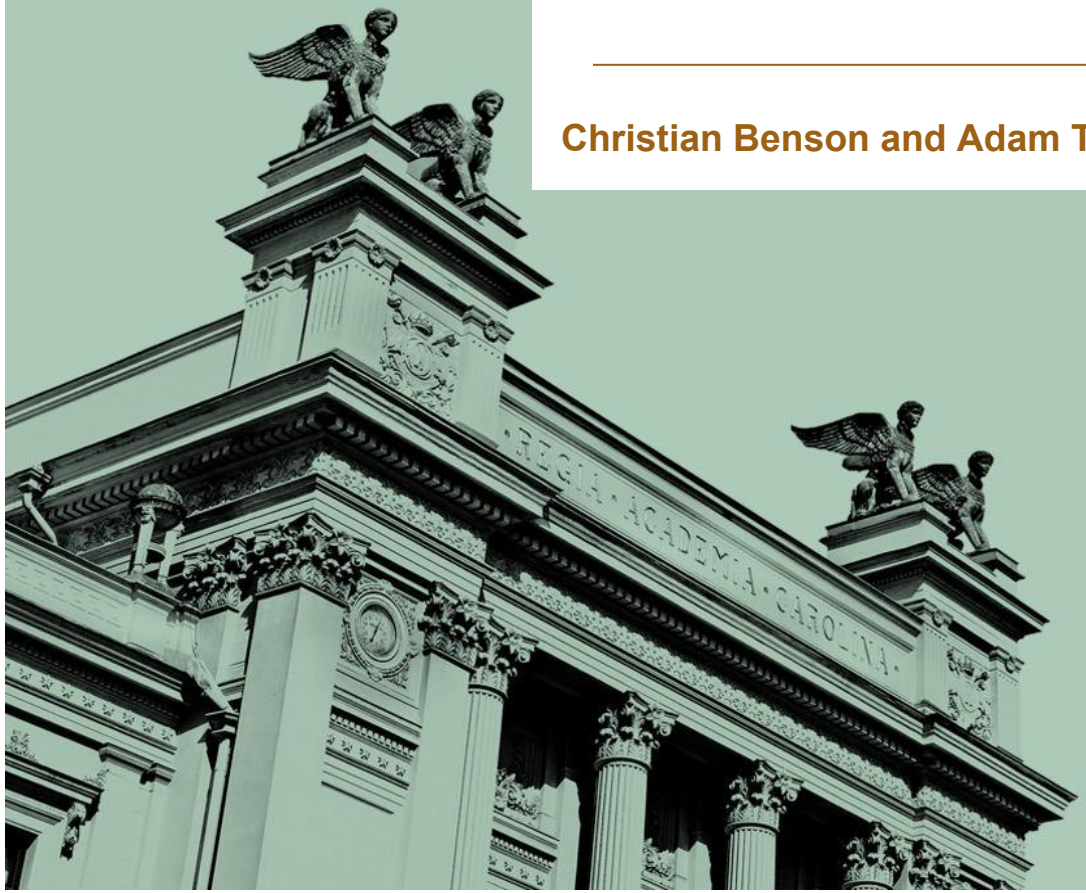
Ad click fraud detection

Christian Benson and Adam Thuvesen

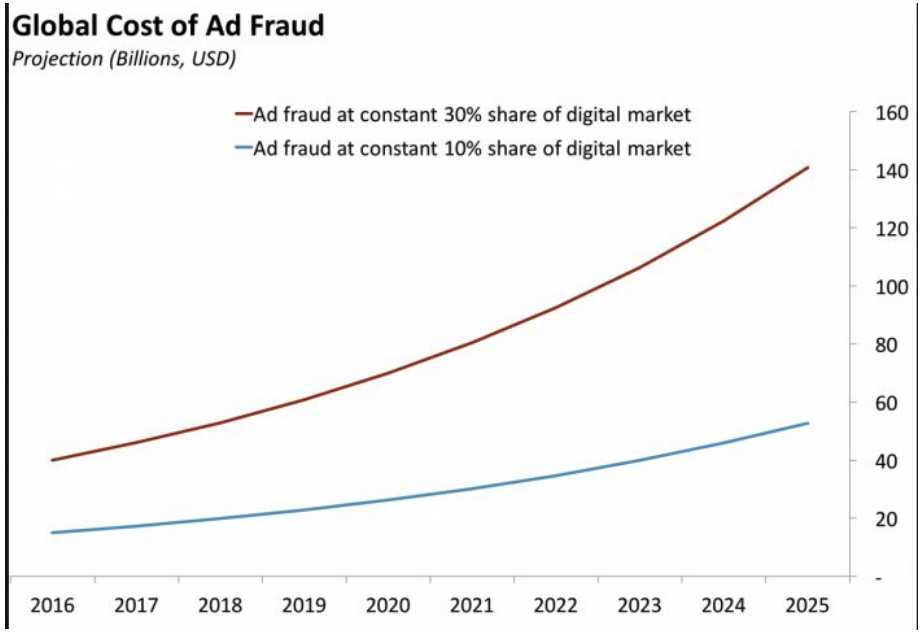


Problem

- Ad click fraud
  - Mobile

- Click fraud is a major issue for advertisers
  - Pay per click ads
    - The app creator (publisher) will profit from more clicks
  - Fraudulent automated clicks
    - The advertiser loses

Problem



- How to detect a fraudulent click in a mobile app?
  - Using data from ad clicks

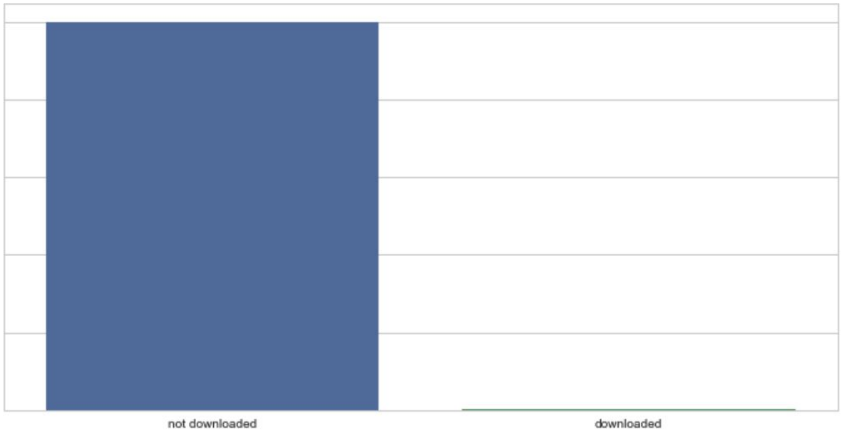
Dataset

- Dataset from Kaggle
- 7 features
  - ip (ip address)
  - app (mobile app)
  - device (type of device)
  - os (operating system)
  - channel (channel id of mobile ad publisher)
  - click time (ad was clicked)
  - attributed time (time of possible download)
  - is attributed (ad led to app download or not)

	ip	app	device	os	channel	click_time	attributed_time	is_attributed
0	87540	12	1	13	497	2017-11-07 09:30:38	NaN	0
1	105560	25	1	17	259	2017-11-07 13:40:27	NaN	0
2	101424	12	1	19	212	2017-11-07 18:05:24	NaN	0
3	94584	13	1	13	477	2017-11-07 04:58:08	NaN	0
4	68413	12	1	1	178	2017-11-09 09:00:09	NaN	0

Dataset

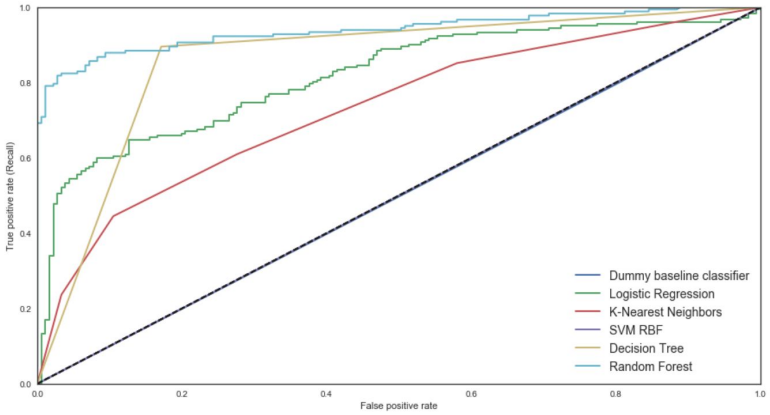
- 187M entries
- Very unbalanced
  - 99.8 % negative samples (not downloaded)



Baseline

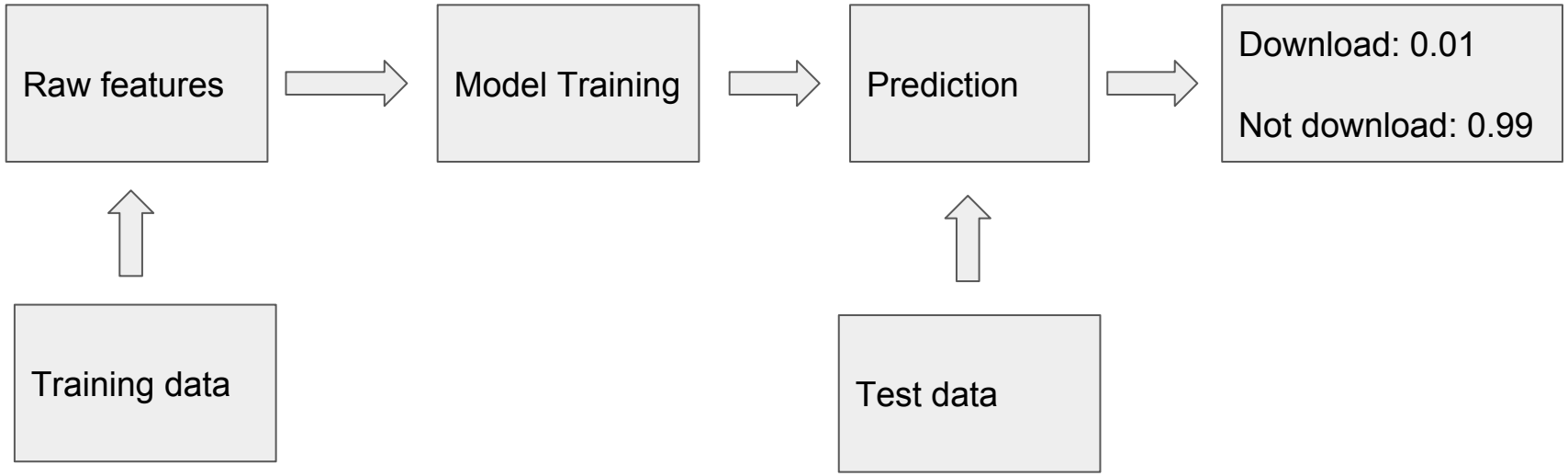
- Dummy
- k-NN
- SVM
- Logistic Regression
- Decision Trees
- Random Forest

- Metric
  - ROC-AUC



Model	ROC AUC
Random Forest	0.938999
Decision Tree	0.862197
Logistic Regression	0.817376
K-Nearest Neighbors	0.730056
SVM RBF	0.500000
Dummy baseline classifier	0.497799

Architecture



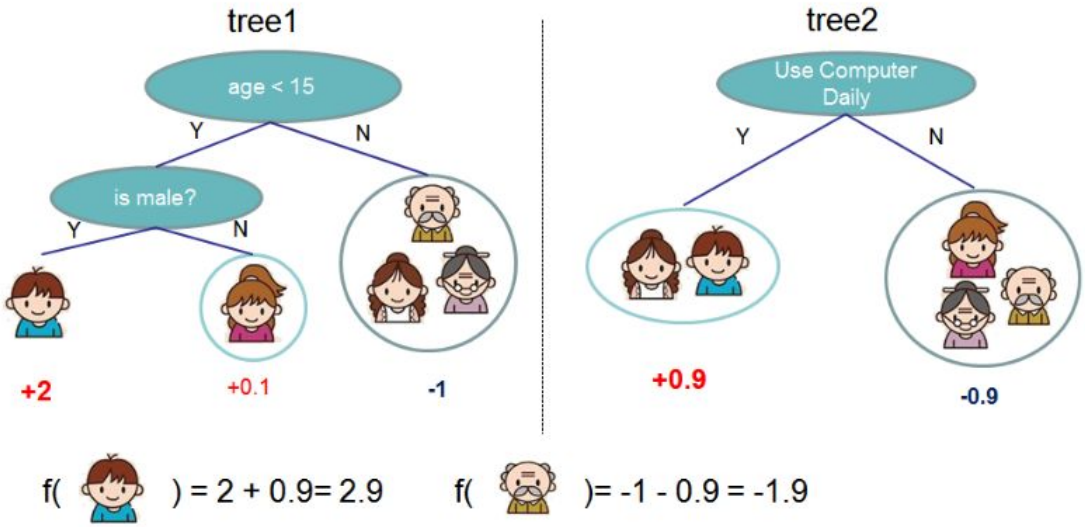
- Raw data is used to train model
- Using trained model to predict on test set

Idea

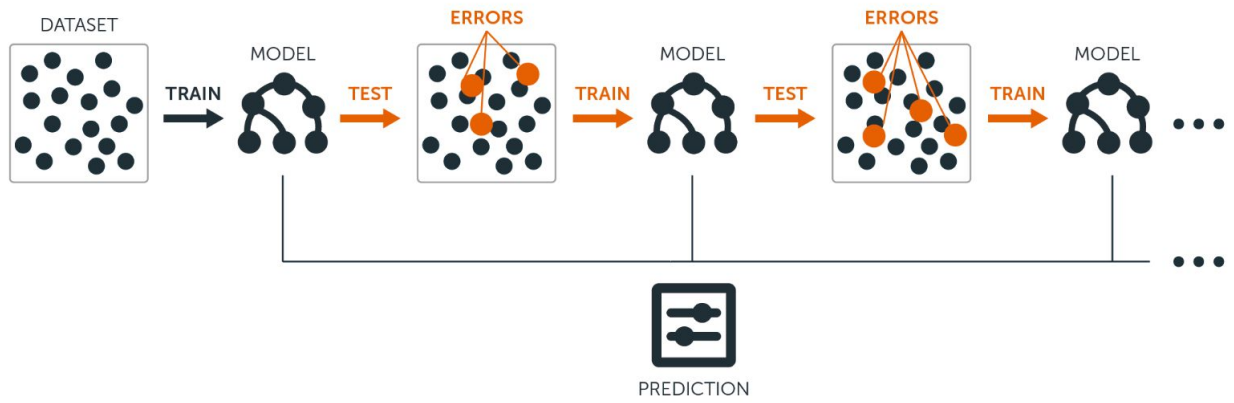
- Decision trees performed well
- Research in the area supported various ensemble of decision trees to be successful in similar problems
- Data preprocessing - extract new features
- Gradient boosted trees
  - Frameworks
    - XGB popular
    - Microsofts LGBM newly gaining attention
- Neural net

How it works - Decision Trees

Ensemble of Decision Trees



How it works - Gradient Boosted Trees



- Error = bias + variance

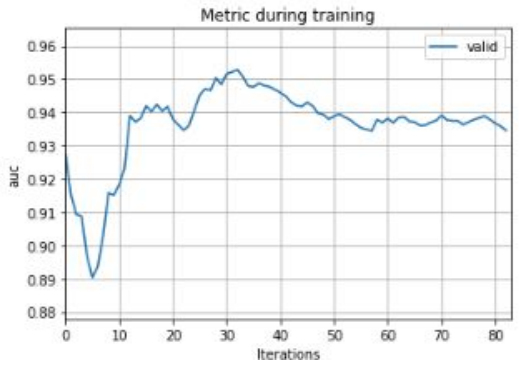
Data preprocessing

- Data preprocessing - extract new features
  - Unique occurrences
  - Total count
  - Cumulative count
  - Variance
  - Mean
  - Aggregation
  - Previous/next click
  - Time
- 23-30 features in total

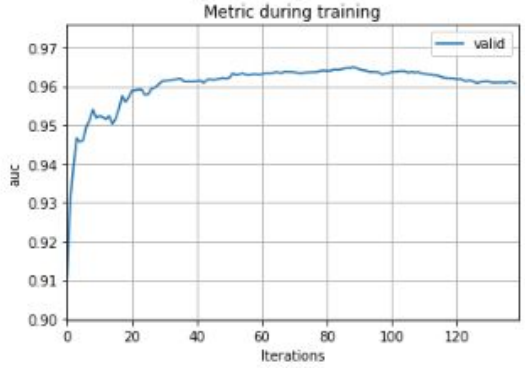
Training

- Trained on 10M entries
- Models
  - Neural net with embedding layer
  - LGBM
  - XGB

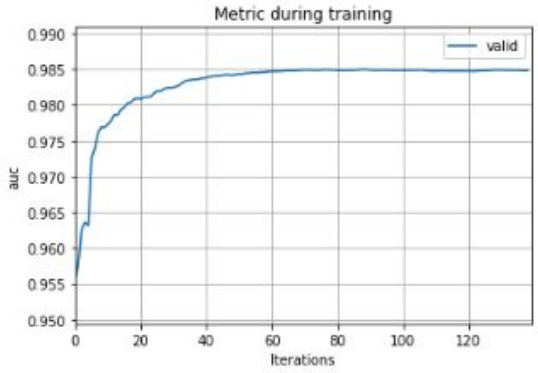
```
Model Report
bst1.best_iteration:  33
auc: 0.952805066468
Plot metrics during training...
```




```
Model Report
bst1.best_iteration:  89
auc: 0.964936424171
Plot metrics during training...
```




```
Model Report
bst1.best_iteration:  89
auc: 0.984961614709
Plot metrics during training...
```




```
[79.83014464378357]: model training time
```

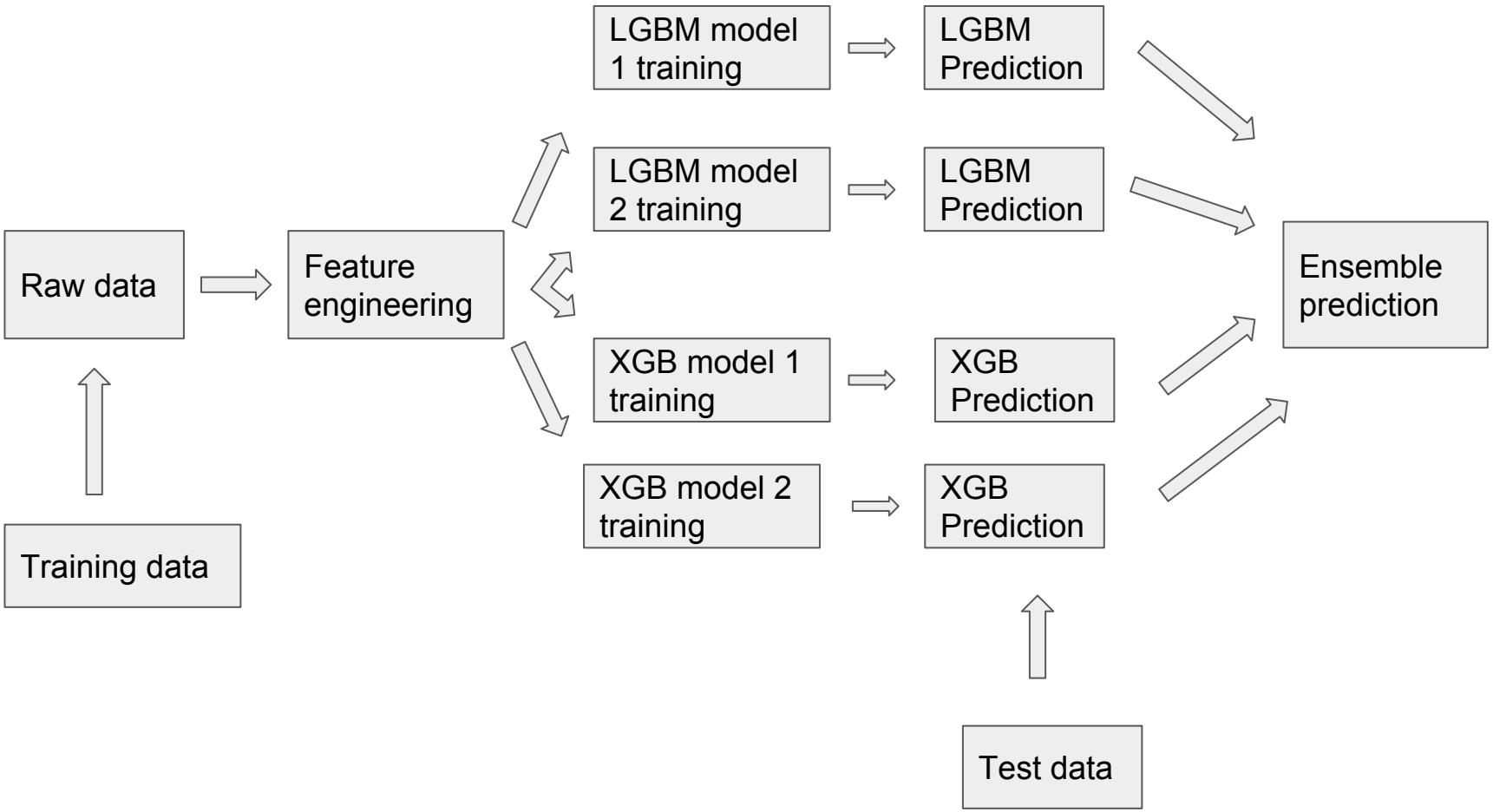
Solution

- Feature Engineering
  - Create new features from existing ones
- Gradient Boosted Trees
  - XGB
  - LGBM
- Ensemble of LGBM and XGB models
- Neural net not performing quite as well

Ensemble

- Combining two or more models for better results
- Can be done in several ways
- Logarithmic average

Solution architecture



Results

- LGBM best model: 0.9784
- XGB best model: 0.9733
- Neural net best model: 0.9508

- Logarithmic ensemble mix including the two best LGBM and the two best XGB: **0.9787**

Thank you for listening!