

# TalkingData AdTracking Fraud Detection Challenge

# Contents Table

0. Overview

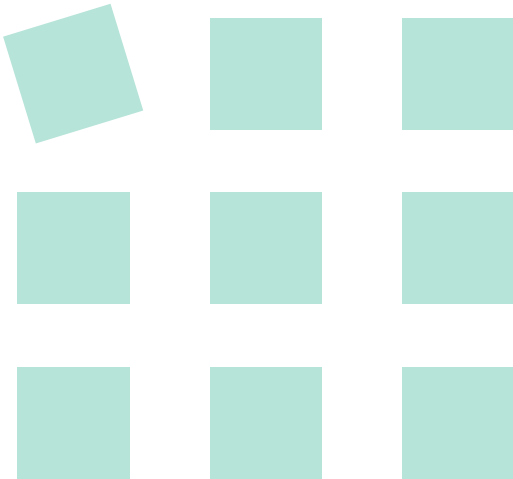
1. Data Exploration

**Method1      Method2      Method3**

2. Data Preprocessing

3. Target Variable Prediction

4. Conclusion



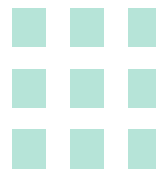
# 0. Overview

## Description

TalkingData, China's largest independent big data service platform, covers over 70% of active mobile devices nationwide. They handle 3 billion clicks per day, of which 90% are potentially fraudulent. The goal of the competition is to create an algorithm that predicts whether a user will download an app after clicking a mobile app ad.

## Evaluation

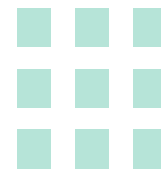
Submissions are evaluated on [area under the ROC curve](#) between the predicted probability and the observed target.



# 0. overview

## variables

- ❖ ip : ip address of click
- ❖ app : app id for marketing
- ❖ device : device type id of user mobile phone
- ❖ 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 download



# 1. Data Exploration

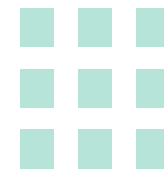
## Explore 100,000 data

- ❖ ip : 100000 non-null int64
- ❖ app : 100000 non-null int64
- ❖ device : 100000 non-null int64
- ❖ os : 100000 non-null int64
- ❖ channel : 100000 non-null int64
- ❖ click\_time : 100000 non-null datetime64
- ❖ attributed\_time : 227 non-null object
- ❖ is\_attributed : 100000 non-null int64

## Check download frequency

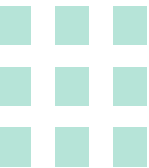
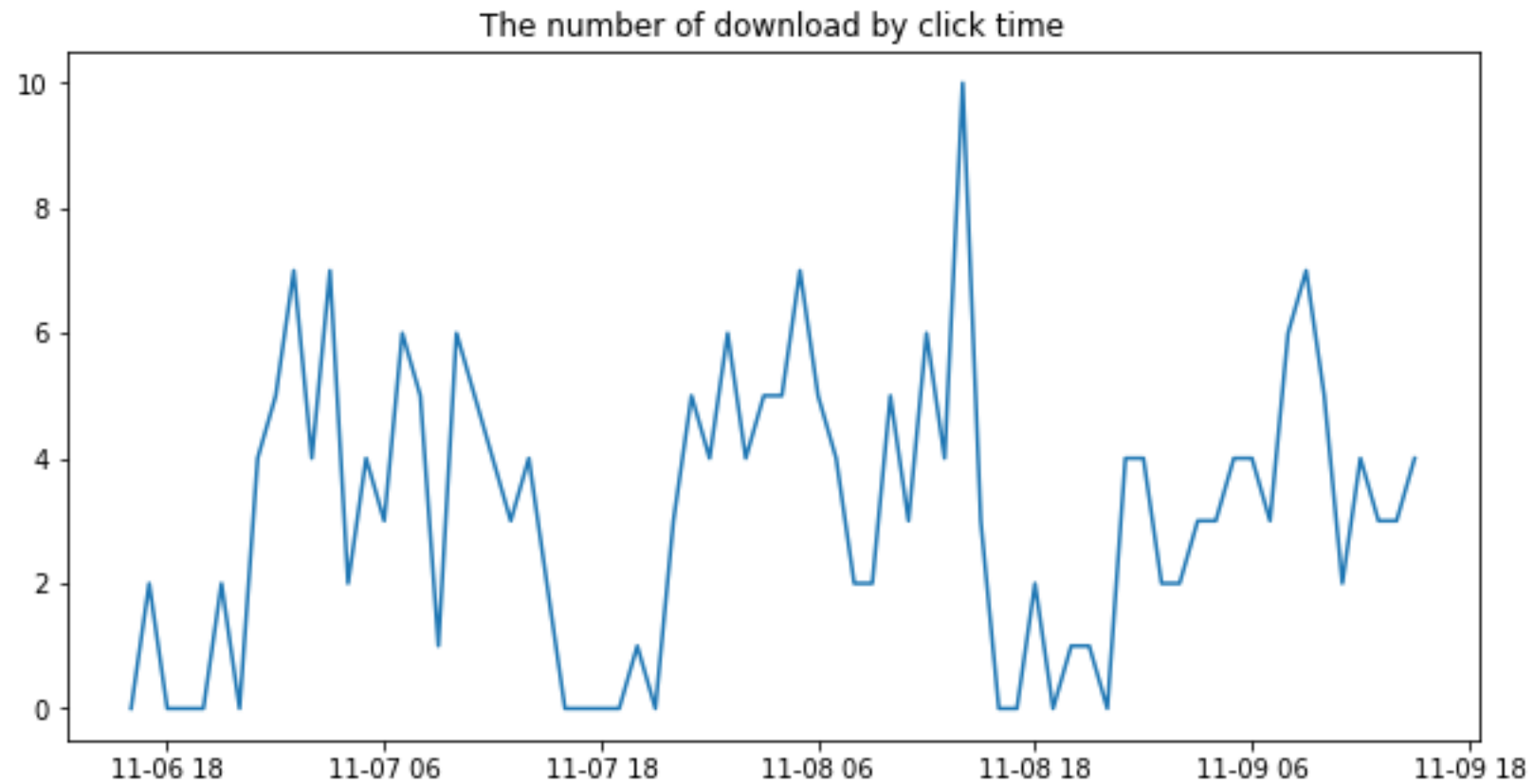
- ❖ 0 : 99773
- ❖ 1 : 227

download proportion : 0.00227



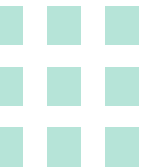
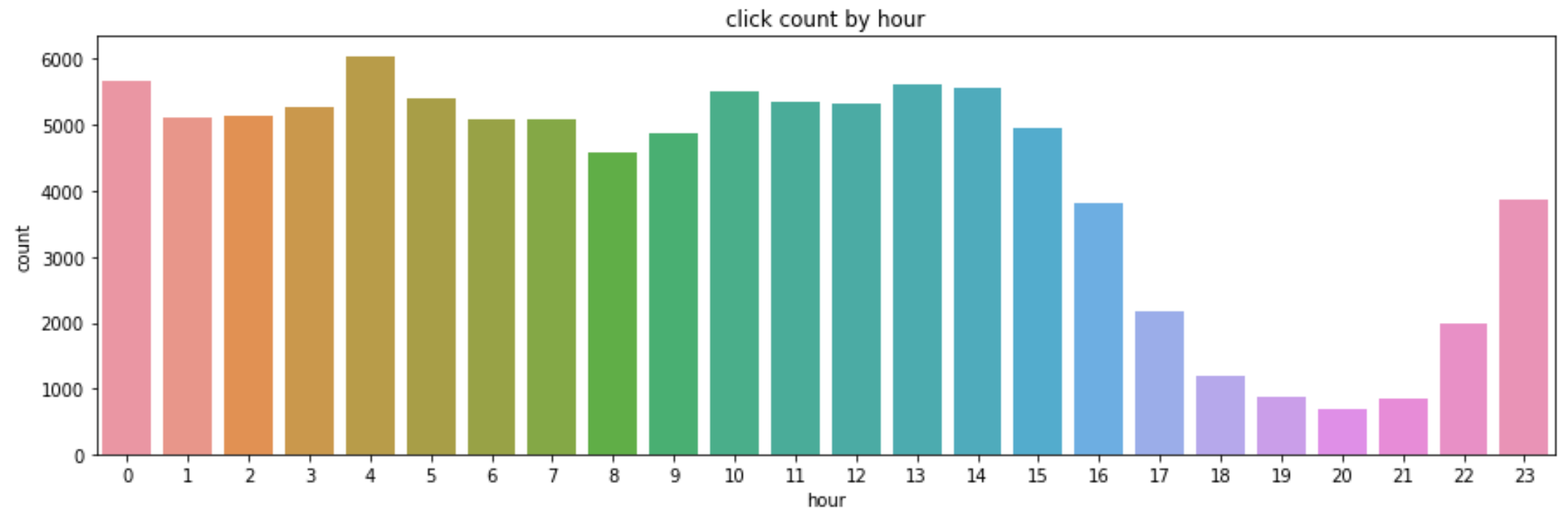
# 1. Data Exploration

Check the number of download by click time



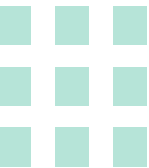
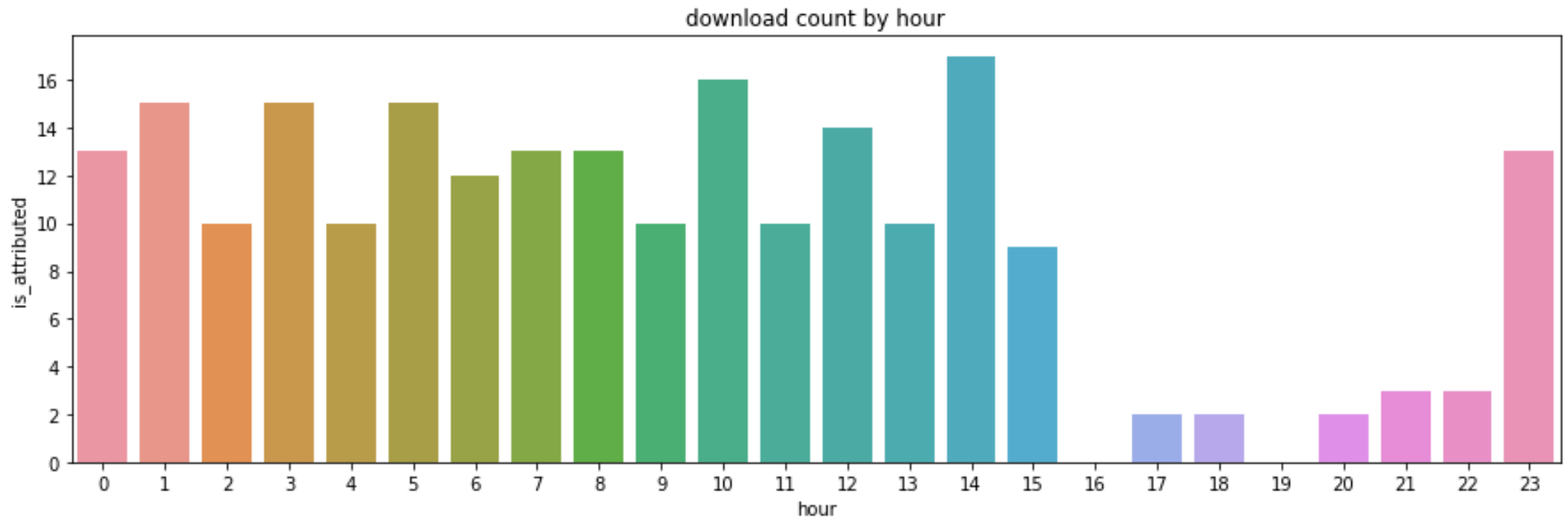
# 1. Data Exploration

Check click count per hour



# 1. Data Exploration

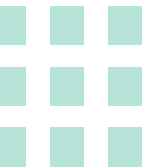
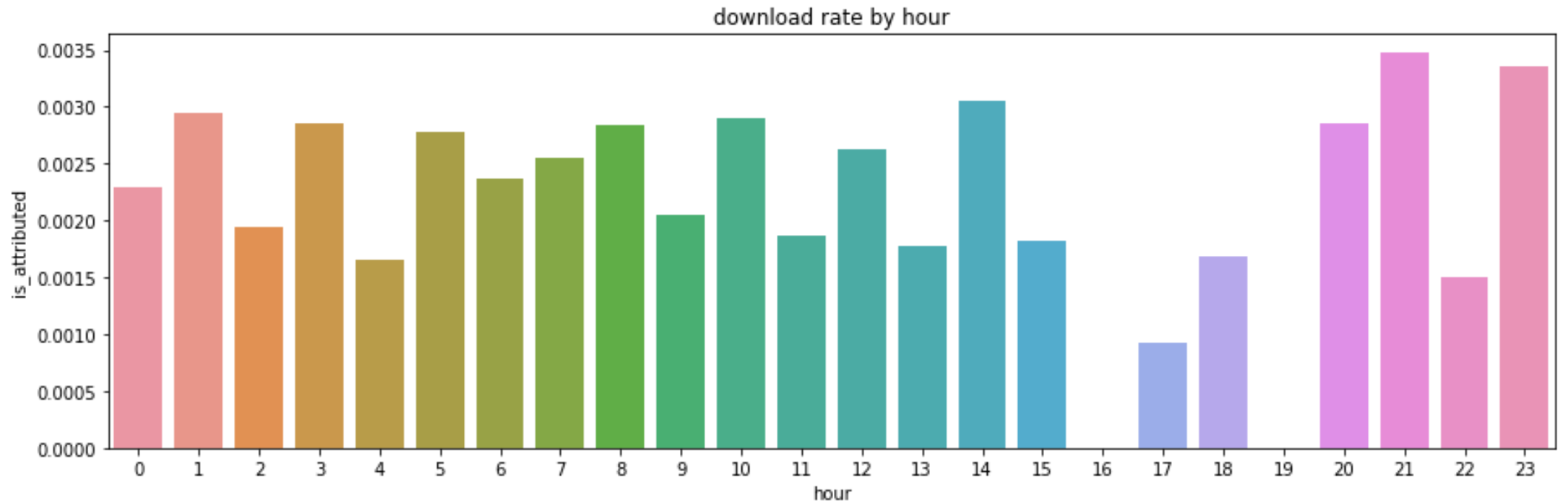
Check download count per hour





# 1. Data Exploration

Check download rate per hour



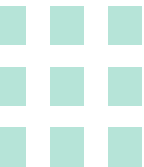
# 1. Data Exploration

Check click count, download count, download rate (by app, device, os, channel)

Don't put the graph here because it is so large.

Please refer to the address below to view it.

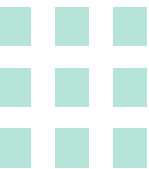
[https://github.com/MinPinSunHwa/Ad\\_Tracking\\_Project](https://github.com/MinPinSunHwa/Ad_Tracking_Project)



# 1. Data Exploration

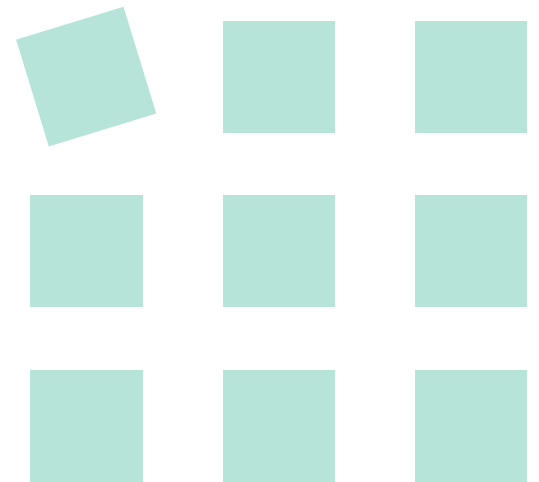
## Check correlation

❖ ip	: 0.053833
❖ app	: 0.059722
❖ device	: 0.001630
❖ os	: 0.001630
❖ channel	: -0.024133
❖ hour	: -0.005629





# Method 1



## 2. Data Preprocessing

Train all data

preprocessing

Train sample  
data

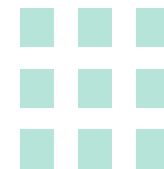
preprocessing

### Make derived variables

Create derived variables in each train all dataset and train sample dataset.

A total of **14 derived variables** are created.

❖ hour : hour from click time



## 2. Data Preprocessing

Train all data

preprocessing

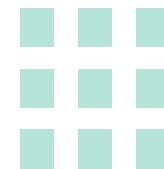
Train sample  
data

preprocessing

### Make derived variables

# : download proportion

- ❖ ip\_attr\_prop : # by ip
- ❖ app\_attr\_prop : # by app
- ❖ device\_attr\_prop : # by device
- ❖ os\_attr\_prop : # by os
- ❖ channel\_attr\_prop : # by channel
- ❖ hour\_attr\_prop : # by hour
- ❖ tot\_attr\_ptop : the sum of the above 6 variables



## 2. Data Preprocessing

Train all data

preprocessing

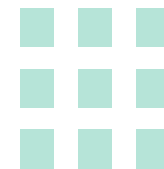
Train sample  
data

preprocessing

### Make derived variables

# : download proportion

- ❖ ip\_hour\_prop : # by ip and hour
- ❖ ip\_app\_prop : # by ip and app
- ❖ ip\_channel\_prop : # by ip and channel
- ❖ hour\_app\_prop : # by hour and app
- ❖ hour\_channel\_prop : # by hour and channel
- ❖ tot\_vv\_prop : the sum of the above 5 variables



## 2. Data Preprocessing

Train all data

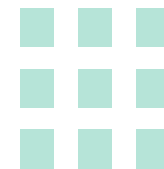
preprocessing

Train sample  
data

preprocessing

### Check correlation

❖ ip_attr_prop	: 0.438892
❖ app_attr_prop	: 0.444209
❖ device_attr_prop	: 0.201987
❖ os_attr_prop	: 0.226293
❖ channel_attr_prop	: 0.389942
❖ hour_attr_prop	: 0.008851
❖ tot_attr_ptop	: 0.532482





## 2. Data Preprocessing

Train all data

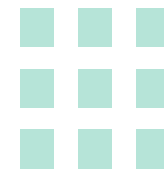
preprocessing

Train sample  
data

preprocessing

### Check correlation

❖ ip_hour_prop	: 0.582208
❖ ip_app_prop	: 0.755585
❖ ip_channel_prop	: 0.715354
❖ hour_app_prop	: 0.457047
❖ hour_channel_prop	: 0.416602
❖ tot_vv_prop	: 0.739013



## 2. Data Preprocessing

Train all data

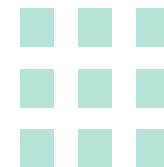


Test data  
preprocessing

### Preprocess test data

Based on train all dataset except 'hour' variable, 13 derived variables are created in the test dataset.

Because train all dataset is the most data, the value of the test dataset can be filled without as many blanks as possible, thus creating derived variables in the test dataset using train all dataset.

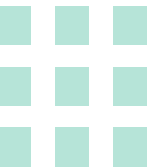


### 3. Target Variable Prediction

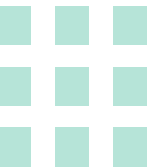
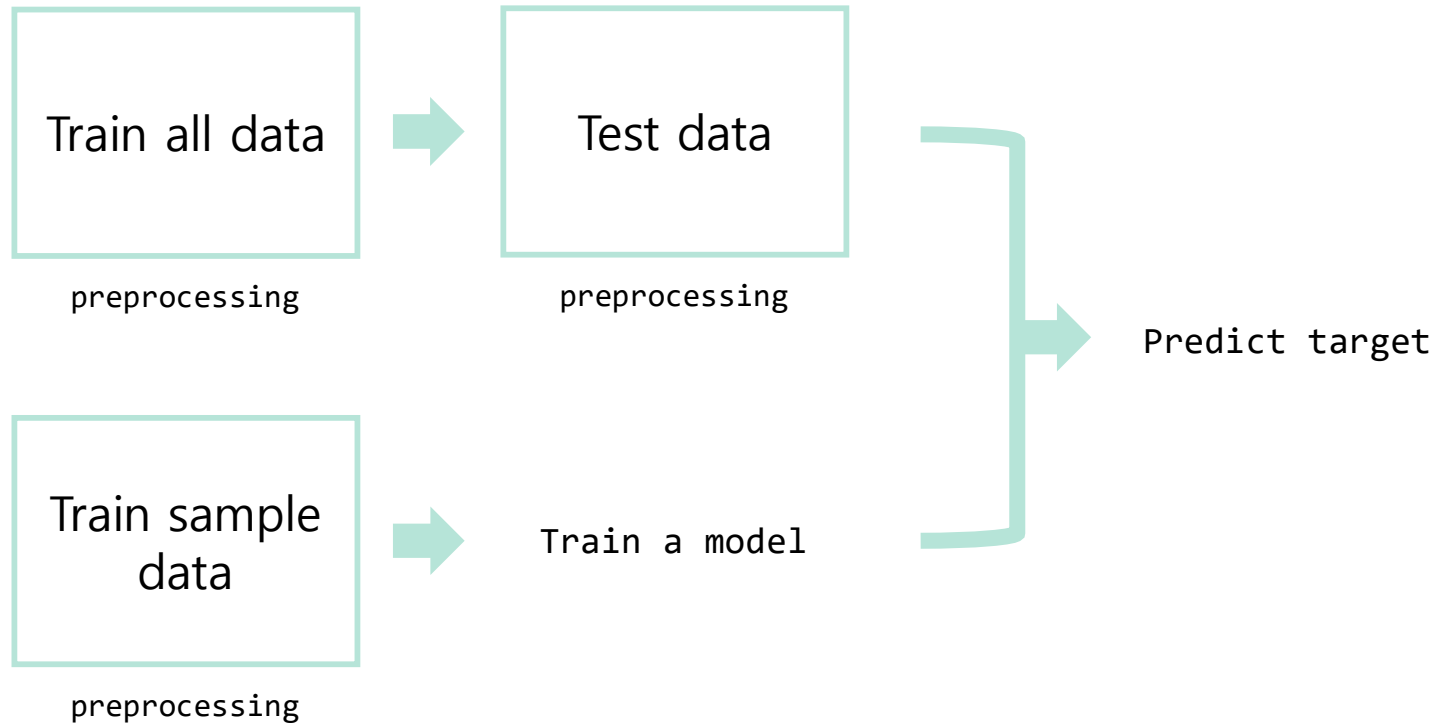
#### Create functions

Create functions prior to prediction of the target variable.

- ❖ `check_data` : To check data distribution
- ❖ `examine_outlier` : To check for values other than 0 and 1



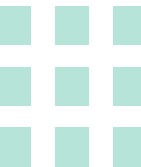
### 3. Target Variable Prediction



### 3. Target Variable Prediction

Create features to use a model

- ❖  $\text{feat1} = \text{ip\_attr\_prop}, \text{app\_attr\_prop}, \text{device\_attr\_prop}, \text{os\_attr\_prop}, \text{channel\_attr\_prop},$   
 $\text{hour\_attr\_prop}, \text{tot\_attr\_prop}$
- ❖  $\text{feat2} = \text{ip\_hour\_prop}, \text{ip\_app\_prop}, \text{ip\_channel\_prop}, \text{hour\_app\_prop}, \text{hour\_channel\_prop},$   
 $\text{tot\_vv\_prop}$
- ❖  $\text{feat3} = \text{feat1} + \text{feat2}$
- ❖  $\text{feat4} = \text{ip\_attr\_prop}, \text{app\_attr\_prop}, \text{channel\_attr\_prop}, \text{tot\_attr\_prop}$
- ❖  $\text{feat5} = \text{feat4} + \text{feat2}$
- ❖  $\text{feat6} = \text{app\_attr\_prop}, \text{channel\_attr\_prop}, \text{hour\_app\_prop}, \text{hour\_channel\_prop}$

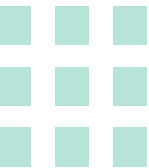


### 3. Target Variable Prediction

#### Predict target variable

- ❖ Linear Regression
- ❖ Ridge
- ❖ Logistic Regression
- ❖ Decision Tree
- ❖ Random Forest
- ❖ Gradient Boosting
- ❖ K-Nearest Neighbors
- ❖ Support Vector machines
- ❖ LightGBM

Skip because it takes too long



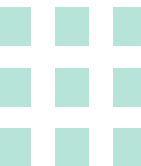
### 3. Target Variable Prediction

Predict target variable

❖ Linear Regression

	10m	20m	30m
feat1	0.9336475	0.3937085	0.9396936
feat2	0.7903207	0.7990348	0.8090254
feat3	0.6832881	0.6891693	0.6870306
feat4	0.9394377	0.9393066	0.9394337
feat5	0.6786381	0.6730954	0.6829231
feat6	0.9467690	0.9468087	0.9466697

- ✓ 10m, 20m, 30m : 10, 20, 30 million train data
- ✓ The value in table : kaggle score (AUC)



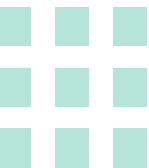
### 3. Target Variable Prediction

Predict target variable

❖ Logistic Regression

C	10m	20m	30m
0.01	0.9518560	0.9518226	0.9518260
0.1	0.9517896	0.9518113	0.9517822
1	0.9517904	0.9517846	0.9517540
10	0.9517882	0.9517830	0.9517553

✓ feature : feat6





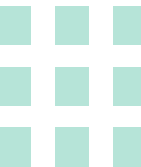
### 3. Target Variable Prediction

Predict target variable

❖ Decision Tree

max_depth	10m	20m	30m
3	0.9039194	0.9039806	0.9040380
4	0.9068583	0.9065484	0.9067215
5	0.9379549	0.9245333	0.9310434

✓ feature : feat6



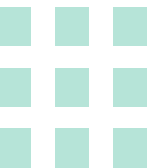
### 3. Target Variable Prediction

Predict target variable

❖ Random Forest

<div>n_estimators</div> <div>max_depth</div>	30	50	70
3	0.9117286	0.9325352	0.9325768
4	0.9446114	0.9444698	0.9481182
5	0.9511519	0.9506940	0.9506489

- ✓ feature : feat6
- ✓ sample : 10m
- ✓ max\_features : 1



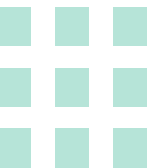
### 3. Target Variable Prediction

Predict target variable

❖ Gradient Boosting

n_estimators max_depth	30	50
3	0.9058254	0.9069254
4	0.9426463	0.9432340
5	0.9477711	0.9486383

- ✓ feature : feat6
- ✓ sample : 10m
- ✓ learning\_rate : 0.01

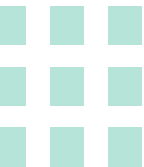


### 3. Target Variable Prediction

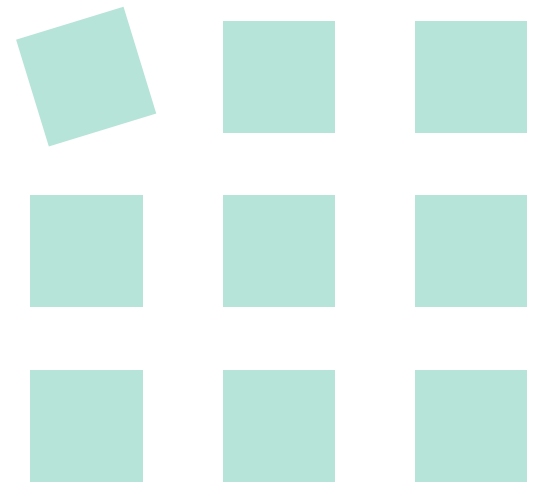
Predict target variable

❖ LightGBM

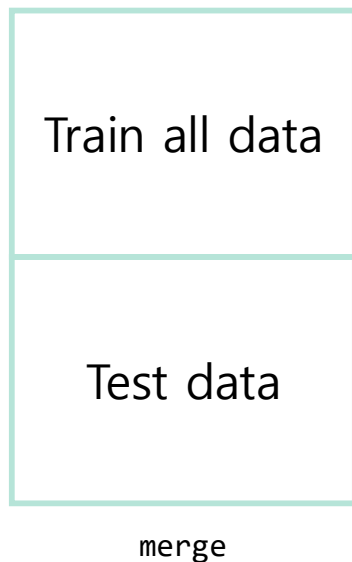
	10m	20m	30m
feat1	0.9426481	0.9411704	0.9398357
feat2	0.8694790	0.8232350	0.8775217
feat3	0.8694790	0.8467034	0.8577380
feat4	0.9410401	0.9413678	0.9411245
feat5	0.8921562	0.8471011	0.8415991
feat6	0.9514271	0.9528658	0.9526517



# Method2



## 2. Data Preprocessing

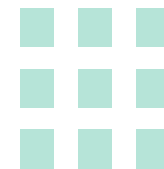


Make and fill a variable 'is\_attributed' in test data

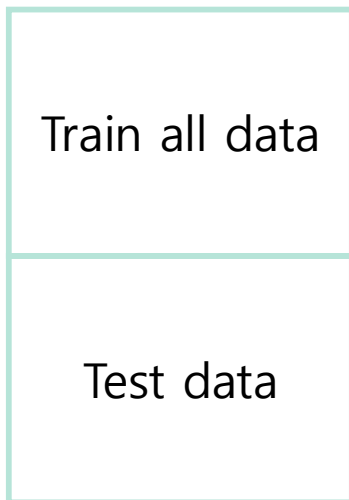
Make a variable 'is\_attributed' in test data, then fill the variable with the proportion of download in train data

Merge train data and test data

Combine train data and test data to make derived variables together.



## 2. Data Preprocessing



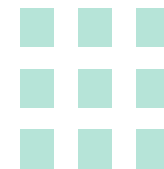
preprocessing

### Make derived variables

Create **21 derived variables** in merged dataset.

After preprocessing separate dataset, then extract a sample.

❖ 14 derived variables made in method1



## 2. Data Preprocessing

Train all data

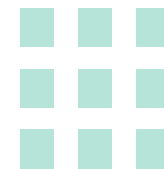
Test data

preprocessing

### Make derived variables

# : download proportion among download

- ❖ ip\_attr\_tot\_prop : # by ip
- ❖ app\_attr\_tot\_prop : # by app
- ❖ device\_attr\_tot\_prop : # by device
- ❖ os\_attr\_tot\_prop : # by os
- ❖ channel\_attr\_tot\_prop : # by channel
- ❖ hour\_attr\_tot\_prop : # by hour
- ❖ tot\_attr\_tot\_ptop : the sum of the above 6 variables





## 2. Data Preprocessing

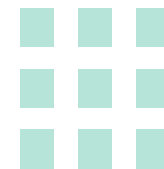
Train all data

Test data

preprocessing

### Check correlation

❖ ip_attr_prop	: 0.438472
❖ app_attr_prop	: 0.442714
❖ device_attr_prop	: 0.235278
❖ os_attr_prop	: 0.226075
❖ channel_attr_prop	: 0.389457
❖ hour_attr_prop	: 0.007377
❖ tot_attr_ptop	: 0.547662



## 2. Data Preprocessing

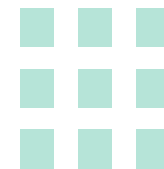
Train all data

Test data

preprocessing

### Check correlation

❖ ip_attr_tot_prop	: -0.003495
❖ app_attr_tot_prop	: 0.235278
❖ device_attr_tot_prop	: -0.044279
❖ os_attr_tot_prop	: -0.001541
❖ channel_attr_tot_prop	: 0.264980
❖ hour_attr_tot_prop	: 0.007057
❖ tot_attr_tot_ptop	: 0.026574



## 2. Data Preprocessing

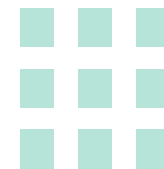
Train all data

Test data

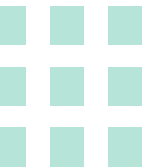
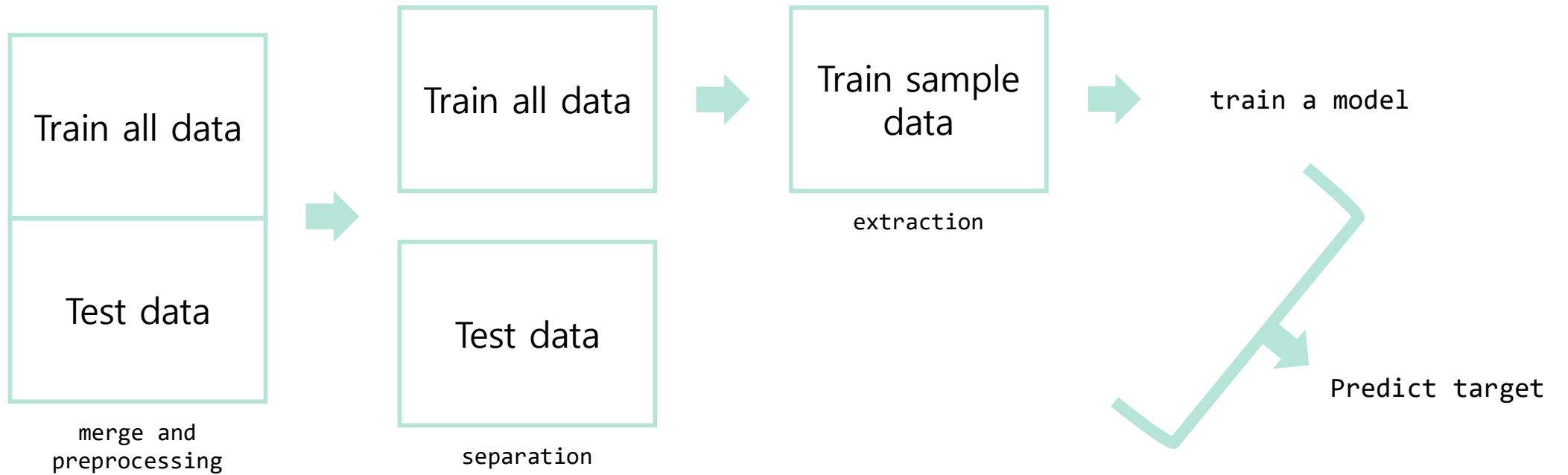
preprocessing

### Check correlation

❖ ip_hour_prop	: 0.581782
❖ ip_app_prop	: 0.753387
❖ ip_channel_prop	: 0.713664
❖ hour_app_prop	: 0.452420
❖ hour_channel_prop	: 0.413714
❖ tot_vv_ptop	: 0.739452



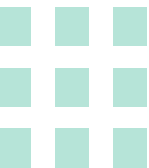
### 3. Target Variable Prediction



### 3. Target Variable Prediction

Create features to use a model

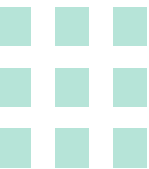
- ❖  $\text{feat1} = \text{ip\_attr\_prop}, \text{app\_attr\_prop}, \text{device\_attr\_prop}, \text{os\_attr\_prop}, \text{channel\_attr\_prop},$   
 $\text{hour\_attr\_prop}, \text{tot\_attr\_prop}$
- ❖  $\text{feat2} = \text{ip\_hour\_prop}, \text{ip\_app\_prop}, \text{ip\_channel\_prop}, \text{hour\_app\_prop}, \text{hour\_channel\_prop},$   
 $\text{tot\_vv\_prop}$
- ❖  $\text{feat3} = \text{feat1} + \text{feat2}$
- ❖  $\text{feat4} = \text{ip\_attr\_prop}, \text{app\_attr\_prop}, \text{channel\_attr\_prop}, \text{tot\_attr\_prop}$
- ❖  $\text{feat5} = \text{feat4} + \text{feat2}$
- ❖  $\text{feat6} = \text{feat5} + \text{app\_attr\_tot\_prop}, \text{channel\_attr\_tot\_prop}$



### 3. Target Variable Prediction

Create features to use a model

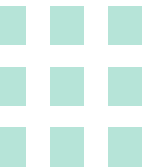
- ❖  $\text{feat7} = \text{app\_attr\_prop}, \text{channel\_attr\_prop}, \text{hour\_app\_prop}, \text{hour\_channel\_prop}$
- ❖  $\text{feat8} = \text{feat7} + \text{app\_attr\_tot\_prop}, \text{channel\_attr\_tot\_prop}$
- ❖  $\text{feat9} = \text{app\_attr\_prop}, \text{device\_attr\_prop}, \text{os\_attr\_prop}, \text{channel\_attr\_prop}, \text{hour\_attr\_prop}$
- ❖  $\text{feat10} = \text{feat9} + \text{hour\_app\_prop}, \text{hour\_channel\_prop}$
- ❖  $\text{feat11} = \text{feat10} + \text{app\_attr\_tot\_prop}, \text{channel\_attr\_tot\_prop}$



### 3. Target Variable Prediction

Predict target variable

- ❖ LightGBM
- ❖ LightGBM : add categorical\_feature (app, channel)
- ❖ Mean of the highest 3 scores

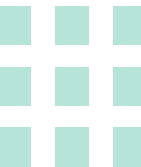


### 3. Target Variable Prediction

Predict target variable

❖ LightGBM

	10m	20m	30m	40m	50m
feat1				0.5688519	
feat2				0.7514380	
feat3				0.5293284	
feat4				0.5320984	
feat5				0.2968826	
feat6				0.6316038	





### 3. Target Variable Prediction

Predict target variable

❖ LightGBM

	10m	20m	30m	40m	50m
feat7	0.9509782	0.9519082	0.9505800	0.9509782	0.9520227
feat8	0.9538612	0.9527098	0.9525610	0.9532771	0.9525889
feat9	0.9572276	0.9550265	0.9568368	0.9532595	0.9556014
feat10	0.9501722	0.9504824	0.9508289	0.9524248	0.9516097
feat11	0.9544192	0.9564199	0.9538744	0.9536148	0.9525215

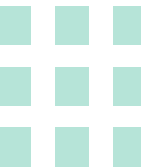


### 3. Target Variable Prediction

Predict target variable

❖ LightGBM : add categorical\_feature

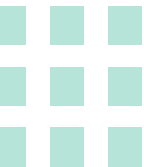
	10m	20m	30m	40m	50m
feat7	0.9536391	0.9541964	0.9543481		
feat8	0.9544556	0.9544834	0.9539668		
feat9	0.9592092	0.9591456	0.9594930	0.9569738	0.9585332
feat10	0.9579120	0.9571316	0.9572331	0.9565375	0.9558988
feat11	0.9576898	0.9583076	0.9570422	0.9567561	0.9542069



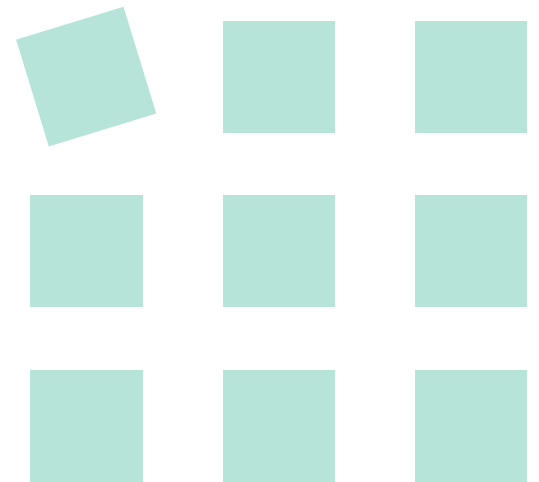
### 3. Target Variable Prediction

Predict target variable

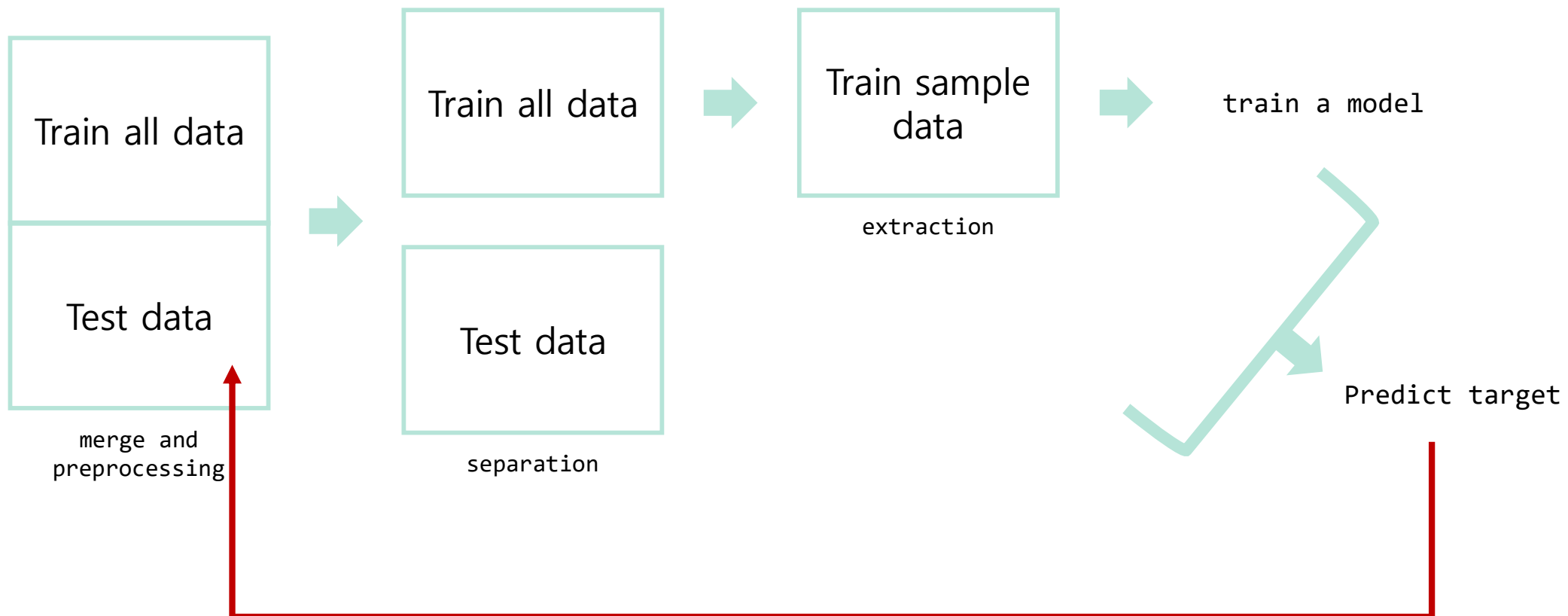
❖ Mean of the highest 3 scores : 0.9601829



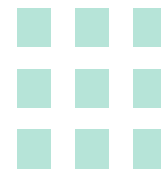
# Method3



## 2. Data Preprocessing



Inserts the target variable predicted by the best score in method2 into the variable 'is\_attributed' in test dataset.



## 2. Data Preprocessing

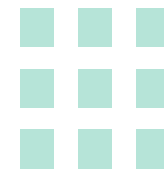
Train all data

Test data

preprocessing

### Check correlation

❖ ip_attr_prop	: 0.433923
❖ app_attr_prop	: 0.415598
❖ device_attr_prop	: 0.195802
❖ os_attr_prop	: 0.217134
❖ channel_attr_prop	: 0.361186
❖ hour_attr_prop	: 0.001310
❖ tot_attr_ptop	: 0.437112



## 2. Data Preprocessing

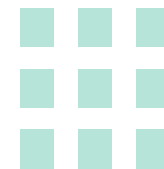
Train all data

Test data

preprocessing

### Check correlation

❖ ip_attr_tot_prop	: -0.004292
❖ app_attr_tot_prop	: 0.058505
❖ device_attr_tot_prop	: -0.047266
❖ os_attr_tot_prop	: -0.007671
❖ channel_attr_tot_prop	: 0.175313
❖ hour_attr_tot_prop	: 0.001170
❖ tot_attr_tot_ptop	: -0.013554



## 2. Data Preprocessing

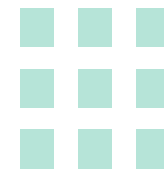
Train all data

Test data

preprocessing

### Check correlation

❖ ip_hour_prop	: 0.565648
❖ ip_app_prop	: 0.706752
❖ ip_channel_prop	: 0.672597
❖ hour_app_prop	: 0.394257
❖ hour_channel_prop	: 0.352294
❖ tot_vv_ptop	: 0.676771

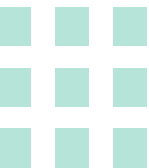




### 3. Target Variable Prediction

Create features to use a model

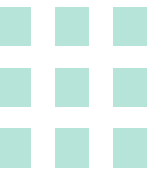
- ❖  $\text{feat1} = \text{ip\_attr\_prop}, \text{app\_attr\_prop}, \text{device\_attr\_prop}, \text{os\_attr\_prop}, \text{channel\_attr\_prop},$   
 $\text{hour\_attr\_prop}, \text{tot\_attr\_prop}$
- ❖  $\text{feat2} = \text{ip\_hour\_prop}, \text{ip\_app\_prop}, \text{ip\_channel\_prop}, \text{hour\_app\_prop}, \text{hour\_channel\_prop},$   
 $\text{tot\_vv\_prop}$
- ❖  $\text{feat3} = \text{feat1} + \text{feat2}$
- ❖  $\text{feat4} = \text{ip\_attr\_prop}, \text{app\_attr\_prop}, \text{channel\_attr\_prop}, \text{tot\_attr\_prop}$
- ❖  $\text{feat5} = \text{feat4} + \text{feat2}$
- ❖  $\text{feat6} = \text{feat5} + \text{app\_attr\_tot\_prop}, \text{channel\_attr\_tot\_prop}$



### 3. Target Variable Prediction

Create features to use a model

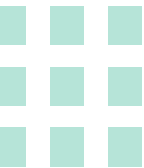
- ❖  $\text{feat7} = \text{app\_attr\_prop}, \text{channel\_attr\_prop}, \text{hour\_app\_prop}, \text{hour\_channel\_prop}$
- ❖  $\text{feat8} = \text{feat7} + \text{app\_attr\_tot\_prop}, \text{channel\_attr\_tot\_prop}$
- ❖  $\text{feat9} = \text{app\_attr\_prop}, \text{device\_attr\_prop}, \text{os\_attr\_prop}, \text{channel\_attr\_prop}, \text{hour\_attr\_prop}$
- ❖  $\text{feat10} = \text{feat9} + \text{hour\_app\_prop}, \text{hour\_channel\_prop}$
- ❖  $\text{feat11} = \text{feat10} + \text{app\_attr\_tot\_prop}, \text{channel\_attr\_tot\_prop}$



### 3. Target Variable Prediction

Predict target variable

- ❖ LightGBM : add categorical\_feature (app, channel)
- ❖ Mean of the highest 3 scores



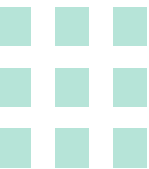
## 4. Conclusion

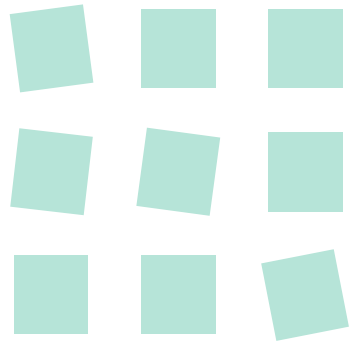
### Result

- ❖ Variables related to app and channel were important.
- ❖ The best score : 0.960

### Realization

- ❖ It was more important to know which variables to use than which model to use.





# Thank you.