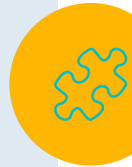

UX
About
AdTracking~
전 형 주

접근에 관한 사용자 경험

하기 싫다!!!

1. fraud 용어에 대한 불편함

2. Big Data



분석을 위한 계획 일정표 설계의 사용자 경험

1주	2주	3주
데이터 탐색	알고리즘 설계	실제 Train/Test에 알고리즘 적용
파생변수 설계	샘플 데이터로 파악	결과 분석을 통한 알고리즘 개선
사용할 알고리즘 선택	알고리즘 개선	

파생 변수 생성에 관한 사용자 경험

EDA

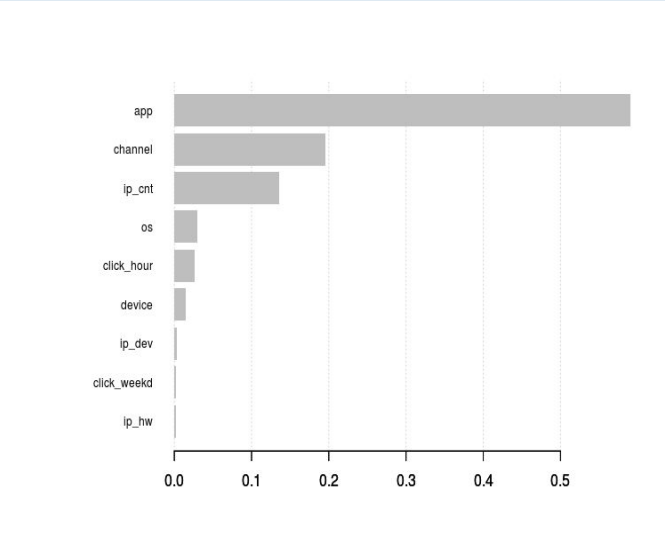
시각화

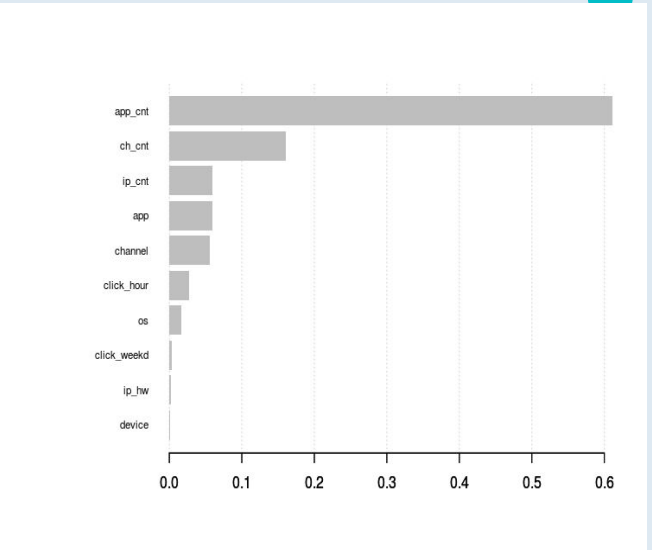
과연 필수 일까?

VS

데이터 외부에서 생각

알고리즘을 통해 찾는 법





xgboost importance 결과를 활용

파생변수 중요도 확인 및 추가

알고리즘에 관한 사용자 경험

Algorithm	Size	Valid.AUC	Time	Submisiion.AUC
GLM	10e6	0.920524	17.8	0.8986
GLM	20e6	0.921052		0.8924
DecisionTree	10e6	0.921312	73.73	0.6558
RandomForest	10e6	memory.limit(52G)		
XGB	10e6	0.974007	39.68	0.9199
	20e6	0.974759	76.07	0.9014
	30e6	0.974503	105.12	0.9421
	50e6	0.9755		0.9485
	ALL	0.977902	900	0.9071
LGBM	10e6	0.973704	26.93	0.8978
	20e6	0.975455	60.63	0.9353

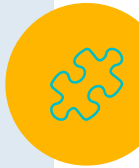
```
> model.rf <- randomForest(as.factor(is_attributed)~., adtr)
Error: cannot allocate vector of size 52.2 Gb
```

UX about

Server for

Big Data

KBig Server	Kaggle Kernels
130G Memory	17G Memory
No DataSet	Dataset
Basic Server	Docker Image for D.S.



UX about

R packages

```
> system.time(adt[, clikcer_n2 := 1:.N, by=list(ip, device,os)])
 사용자  시스템 elapsed
  0.047   0.000   0.047
> system.time(adt <- adt %>% group_by(ip, device, os) %>% mutate(clicker_N = 1:n()))
 사용자  시스템 elapsed
  1.555   0.011   1.567
>
```

```
> a <- system.time(source("test_dplyr.R"))
Class 'difftime'  atomic [1:100000] -120638 -135627 -151524 -104288 -291609 ...
  ..- attr(*, "units")= chr "secs"
> b <- system.time(source("test_datatable.R"))
>
> cat(a,"₩n",b)
11.2 0.14 11.36 NA NA
 5.49 0.1 5.59 NA NA
>
```

UX about

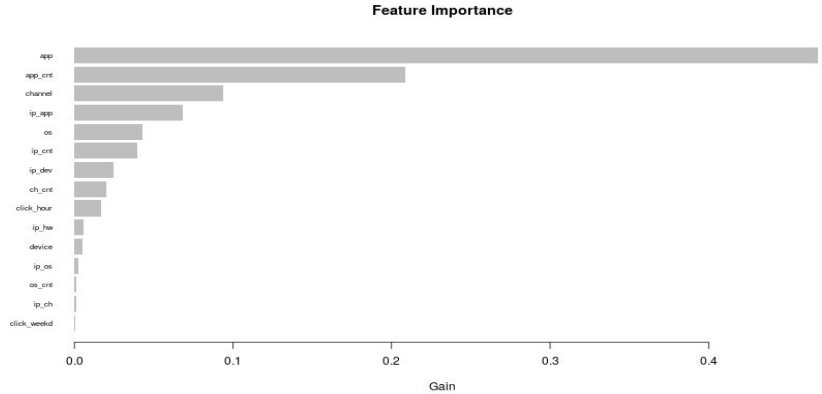
LightGBM Categorical Features

Optimal Split for Categorical Features

We often convert the categorical features into one-hot coding. However, it is not a good solution in tree learner. The reason is, for the high cardinality categorical features, it will grow the very unbalance tree, and needs to grow very deep to achieve the good accuracy.

Actually, the optimal solution is partitioning the categorical feature into 2 subsets, and there are `2^(k-1) - 1` possible partitions. But there is a efficient solution for regression tree[8]. It needs about `k * log(k)` to find the optimal partition.

The basic idea is reordering the categories according to the relevance of training target. More specifically, reordering the histogram (of categorical feature) according to it's accumulate values (`sum_gradient / sum_hessian`), then find the best split on the sorted histogram.



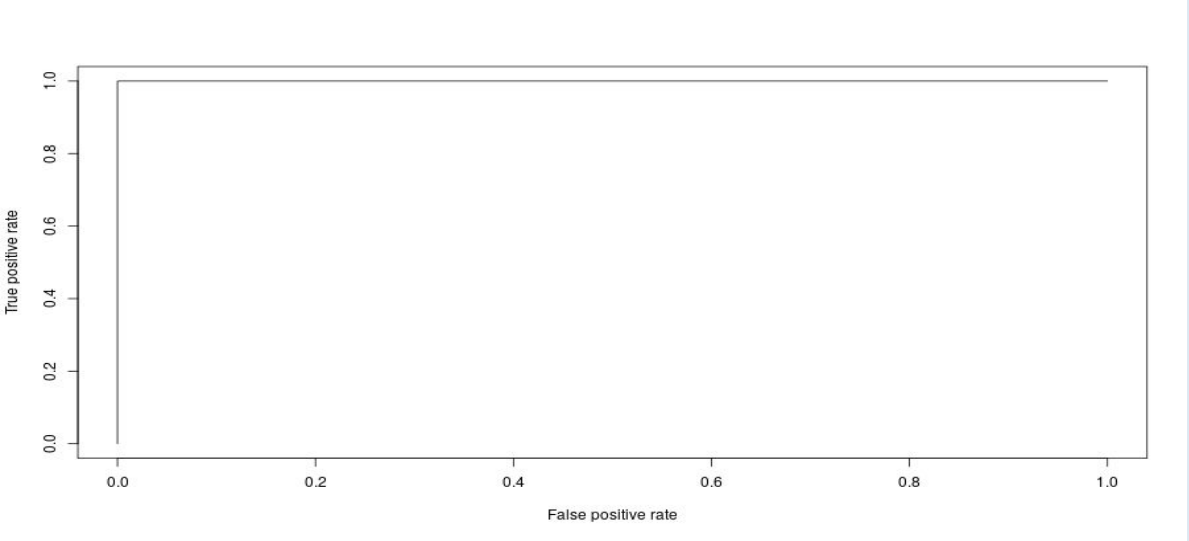
LGBM	10e6	0.973704	26.93	0.8978
	20e6	0.975455	60.63	0.9353
LGBM	10e6		29.52	0.9649
+ Clicker_N	50e6		90	0.9678
+ Catergoricla Feature	10e7			0.9612

Light GBM 기본에

Categorical Features 옵션 추가로

알고리즘 성능 개선

Thinking !!!


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0