IGAWorks
CTR Prediction

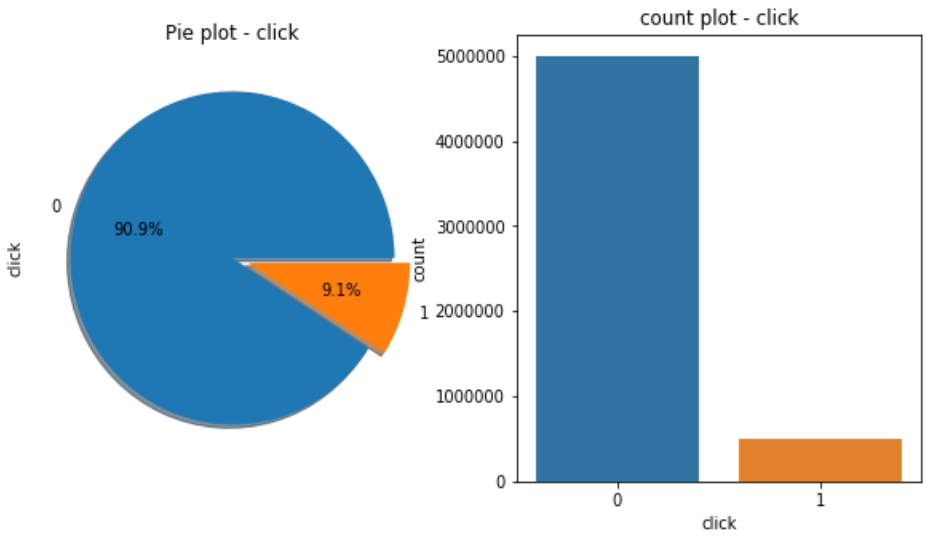
Team 오하이오 _ 윤창원, 이찬주, 김원호

INDEX

1. 탐색적 자료분석

- **LOG DATA_Train.csv / 클릭 비율 및 결측치 확인**

<클릭 비율>



<결측치 확인>

```
column:      click       Percent of NaN value: 0.00%
column: event_datetime   Percent of NaN value: 0.00%
column:     bid_id       Percent of NaN value: 0.00%
column:     ssp_id       Percent of NaN value: 0.00%
column: campaign_id      Percent of NaN value: 0.00%
column:   adset_id       Percent of NaN value: 0.00%
column: placement_type   Percent of NaN value: 0.00%
column:   media_id       Percent of NaN value: 0.00%
column: media_name       Percent of NaN value: 0.00%
column: media_bundle     Percent of NaN value: 0.00%
column: media_domain     Percent of NaN value: 0.00%
column: publisher_id     Percent of NaN value: 0.00%
column: publisher_name   Percent of NaN value: 0.00%
column: device_ifa       Percent of NaN value: 0.00%
column:  device_os       Percent of NaN value: 0.00%
column: device_os_version        Percent of NaN value: 0.00%
column: device_model     Percent of NaN value: 0.00%
column: device_carrier   Percent of NaN value: 0.00%
column: device_make      Percent of NaN value: 0.00%
column: device_connection_type   Percent of NaN value: 0.00%
column: device_language  Percent of NaN value: 0.00%
column: device_country   Percent of NaN value: 0.00%
column: device_region    Percent of NaN value: 0.00%
column: device_city      Percent of NaN value: 0.00%
column: advertisement_id         Percent of NaN value: 0.00%
column: datetime-month   Percent of NaN value: 0.00%
column: datetime-day     Percent of NaN value: 0.00%
column: datetime-hour    Percent of NaN value: 0.00%
column: datetime-minute  Percent of NaN value: 0.00%
column: datetime-second  Percent of NaN value: 0.00%
column: datetime-dayofweek       Percent of NaN value: 0.00%
column:   weekdays       Percent of NaN value: 0.00%
```

- 평균 CTR은 0.091이고 데이터에 결측치는 존재하지 않음을 확인
- Submission파일을 0.091로 예측하고 제출해본 결과 logloss 0.30355
- **목표는 baseline인 0.30355 보다 낮게 예측하는 것**

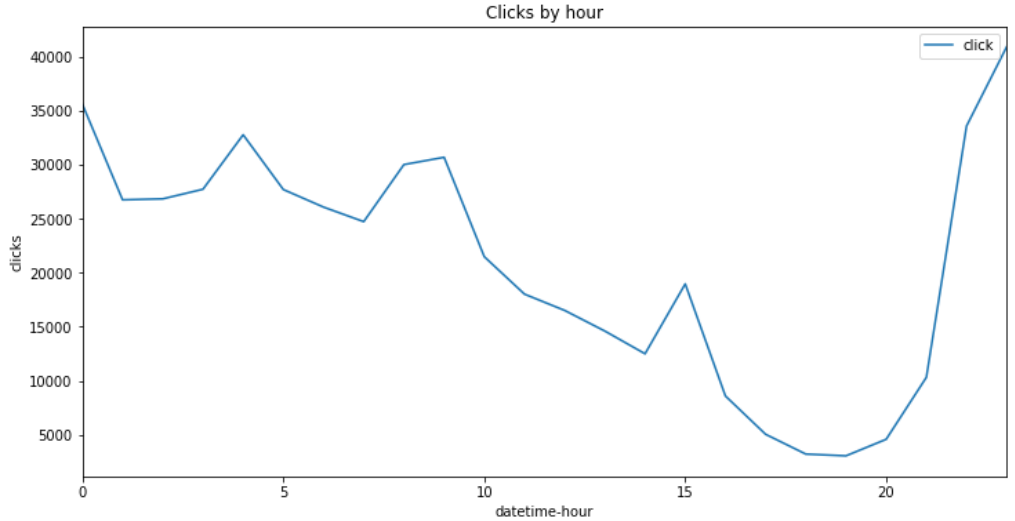
1. 탐색적 자료분석

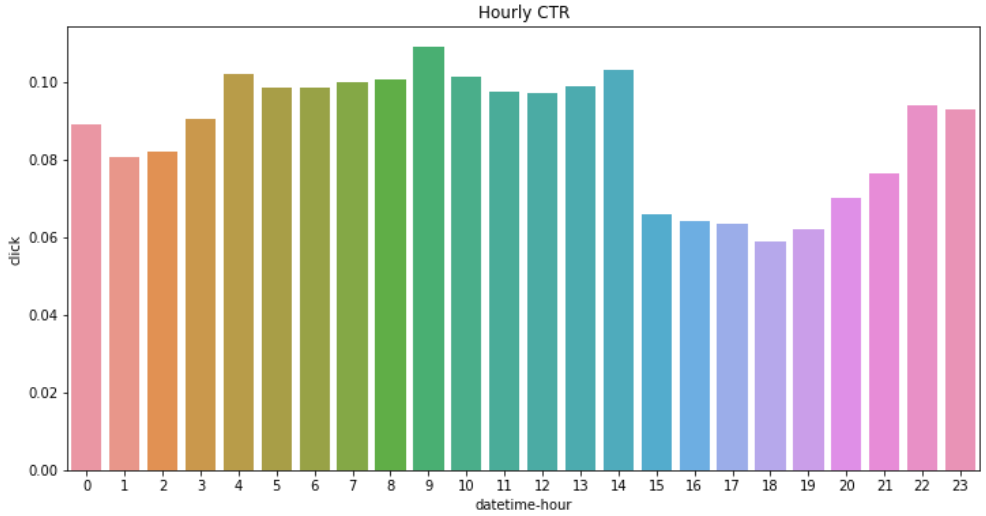
- **LOG DATA_Train.csv / 데이터 구성 및 변수별 카테고리 개수**

1. 미디어 관련 변수	2. 매체 관련 변수	3. 기기 관련 변수
media_domain ..... 162 **media_id** ..... 5815 **media_bundle** ..... 6286 **media_name** 6810	**publisher_name** ..... 1222 **publisher_id** 3939	**device_os** ..... 2 **device_connection** ..... 8 **device_os_version** ..... 125 **device_make** ..... 299 **device_carrier** ..... 536 **device_model** 664
4. 지리/인구통계학적 변수	**5. 광고 관련 변수**	**타겟 변수**
device_country ..... 1 **device_language** ..... 33 **device_region** ..... 148 **device_city** ..... 1326 **device_ifa** 1869137	**placement_type** ..... 4 **ssp_id** ..... 17 **advertisement_id** ... 30 **campaign_id** ..... 186 **adset_id** ..... 872 **event_datetime** ..... (연속형) **bid_id** 5500000	**click** 2

1. 탐색적 자료분석

- **LOG DATA_Train.csv / 시간별 클릭수 변화, 클릭률**

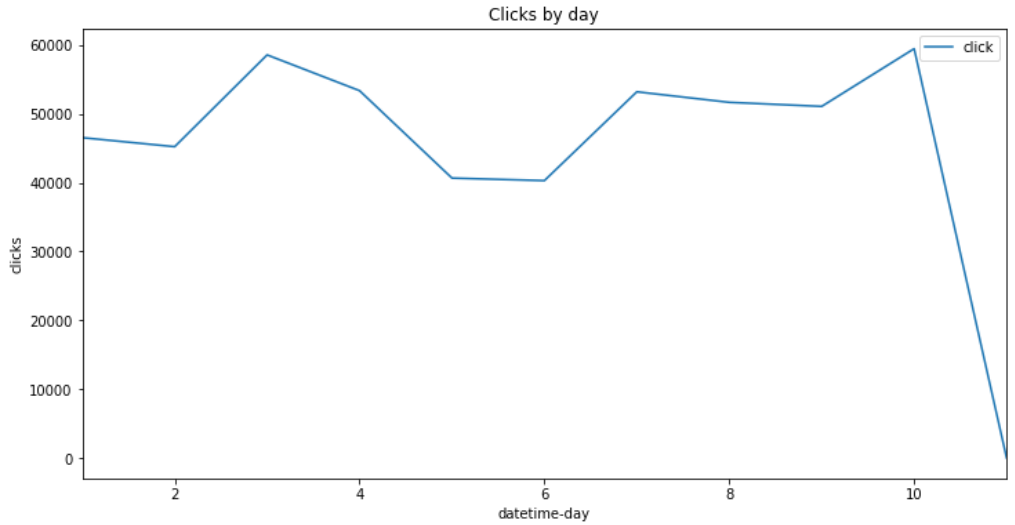


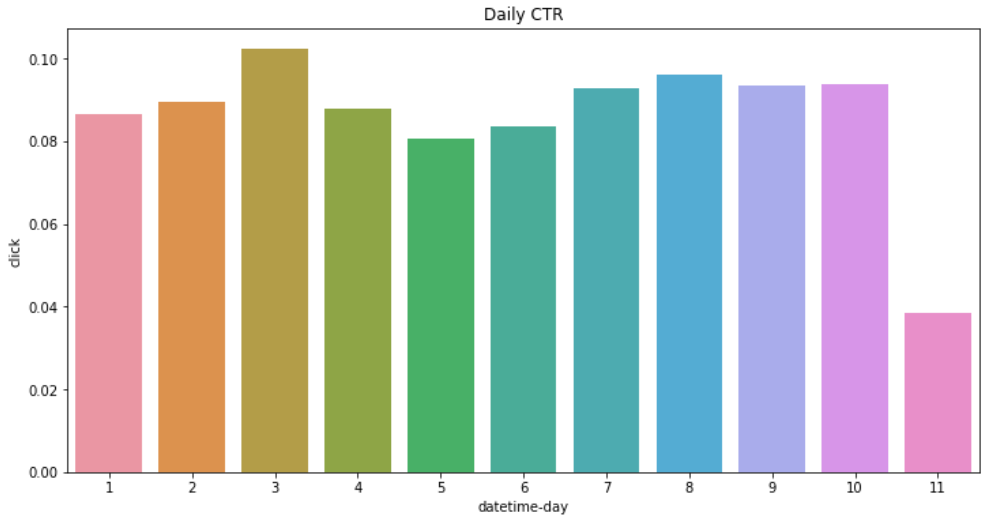


- 클릭수는 오후 6시경에 가장 낮고 9시부터 크게 증가
- 클릭률은 오전대에 주로 높고 오후 3시부터 급격히 낮아지며 오후8시부터 다시 증가
- 시간은 클릭률에 유의미한 영향을 주는 것으로 판단

1. 탐색적 자료분석

- **LOG DATA_Train.csv / 날짜별 클릭수 변화, 클릭률**

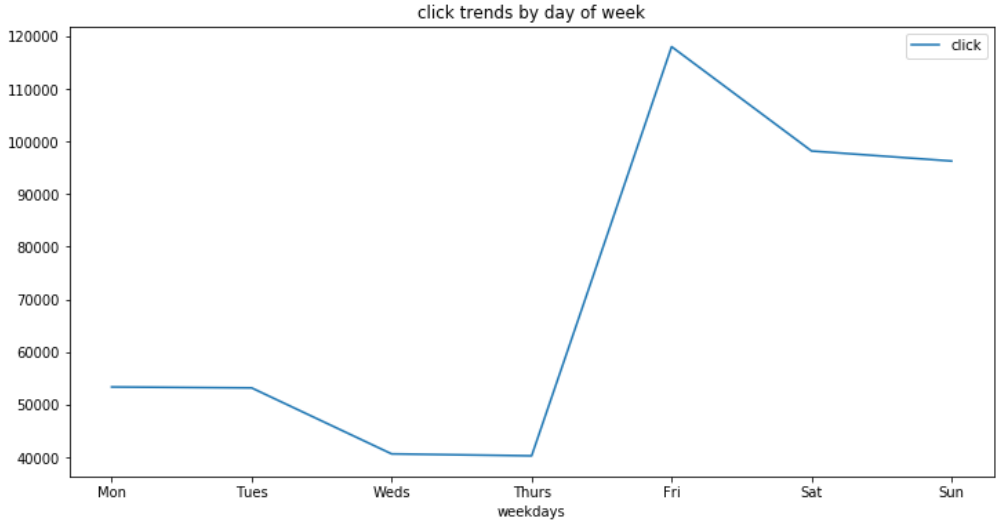


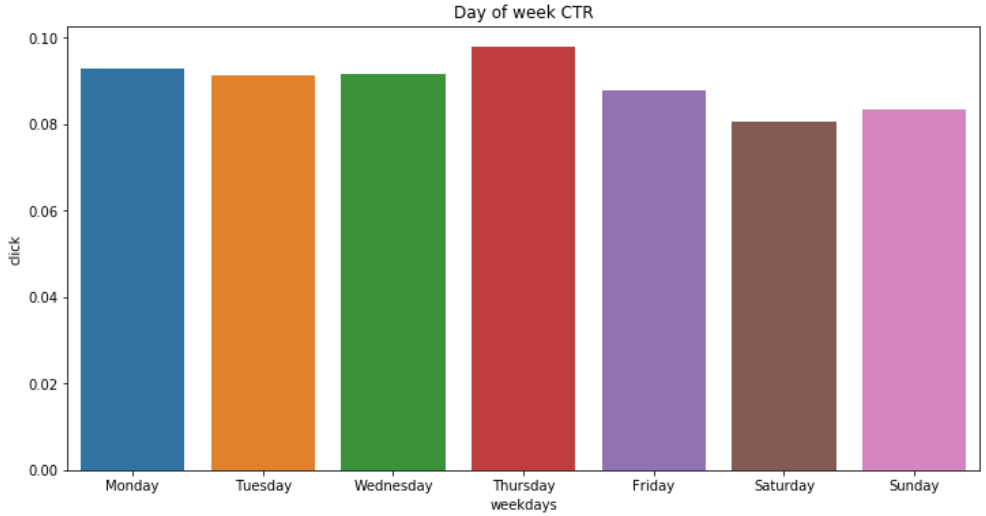


- 클릭수는 3일에 높고 5,6일에 낮으며 10일에 가장 높아지고 11일에 급격히 낮아진다
- 클릭률은 3일에 가장 높고 11일에 가장 낮은 것을 제외하면 날마다 소폭의 변화가 있는 정도
- 11일의 클릭수가 낮은 이유는 로그데이터가 337개밖에 없기 때문이다

1. 탐색적 자료분석

- **LOG DATA_Train.csv / 요일별 클릭수 변화, 클릭률**

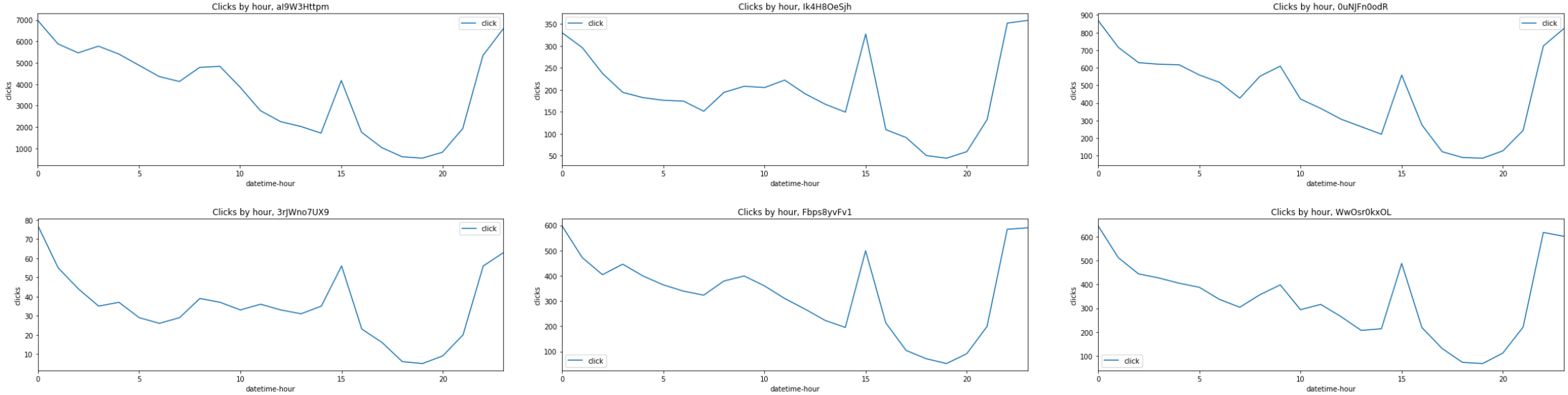




- 클릭수는 금요일과 주말에 높고 주중에는 평균적으로 낮은 양상을 보인다
- 클릭률은 클릭수 추이와 반대로 금요일과 주말이 주중보다 낮은 수치를 보인다

1. 탐색적 자료분석

- **LOG DATA_Train.csv / 지역별 시간에 따른 클릭수 변화**



- Country는 한가지, 시간은 UTC(Coordinated Universal Time) 기준이기 때문에 지역마다 시간이 다르다면 (예를 들어 미국) 시간은 유의미한 변수가 될 수가 없다
- 클릭수 변화를 살펴본 결과 시간대별로 각 지역들은 전부 비슷한 양상을 띄고 있다

1. 탐색적 자료분석

- **LOG DATA_Train.csv / 나머지 범주형 변수들 카테고리별 클릭률**

[device_connection]
category
CTR

```
aEmZFzgDfq,  click through rate: 0.11202867129176879
Hx7e3tE5mu,  click through rate: 0.02499196399871424
WCK2G73H3A,  click through rate: 0.16134515126343332
2xglifxY3C,  click through rate: 0.04713763511711457
6xAYOHll8l,  click through rate: 0.032587978243689274
4sFc3rM27t,  click through rate: 0.07334525939177101
aze5oiXm1V,  click through rate: 0.14285714285714285
55rBbmtMui,  click through rate: 0.0
```

[ssp_id]
category
CTR

```
SrN77Arvqh,  click through rate: 0.018592391146272 65
nwf1A305cO,  click through rate: 0.19391135420991806
M6QaRvdZ8h,  click through rate: 0.1445112199599825
Uox85xVMSC,  click through rate: 0.04661315968804422
VKAHCb2KFB,  click through rate: 0.04684603833790854
ddtzah8tWp,  click through rate: 0.026049019033362986
CD3hRil3bN,  click through rate: 0.036472981116914424
A6EOSZLhXP,  click through rate: 0.08789809843888671
wWGPILy4jH,  click through rate: 0.04554267885001114
```

[media_name]
category
CTR

```
Xbdchs5uK3,  click through rate: 0.024986489784768667
Uk5MGt9vxz,  click through rate: 0.2796980481944098
nOp3mOOieQ,  click through rate: 0.08730136797015338
YJ1AErfOW7,  click through rate: 0.03385265446817974
SjZ9Dgg3gy,  click through rate: 0.07961220374645665
9s3sMlOp5Y,  click through rate: 0.008994066252394524
RmxQM1knLQ,  click through rate: 0.06058413366662281
O2mP8O3Z7Z,  click through rate: 0.02591649767729502
dAWR8DOmzo,  click through rate: 0.0020793735887064 02
```

[os_version]
category
CTR

```
V7LhUlY53m,  click through rate: 0.13590113690775338
aU9KSuwh6B,  click through rate: 0.02127378869365828
OD4DvtralJ,  click through rate: 0.15608067900654576
D9UYEWCoxO,  click through rate: 0.03600974189209483
lpSyhOhQCv,  click through rate: 0.16206832538093732
kFcJHyAdyz,  click through rate: 0.0639907490684826
nSsQ2zP1H6,  click through rate: 0.13089777460816 57
EhcFfj14pL,  click through rate: 0.1553800170794193
rsDijTSU35,  click through rate: 0.05785732773209628
```

- 나머지 범주형 변수들의 카테고리별 클릭률을 모두 확인해본 결과 모두 차이가 있음을 발견

1. 탐색적 자료분석

▪ USER DATA _ Audience Profile.csv

변수명	설명	Value count
install_pack	설치된 앱 정보	41,972종류의 앱
cate_code	IGAW 카테고리별 등급	241개 카테고리
age	나이	12종류 나이
gender	성별	2개 성별
marry	결혼여부	2가지 유형
asset	자산 가격	1가지 수치(대부분 결측값)
house_price	자산 지수	3509가지 수치(대부분 결측값)

- 총 1000만 행으로써 로그데이터에 기록이 없는 오디언스들도 포함
- 설치된 앱 정보와 카테고리별 등급은 추정치가 아니므로 유저 특성을 반영하는 기준으로 간주
- 나이, 성별, 결혼여부는 추정치지만 결측치가 없음
- 자산 가격과 자산 지수는 대부분 결측값이므로 모델에 사용할 변수로서 제외

2. 접근방법

- **문제점**

  - **Cardinality가 높은 범주형 자료**
    - 범주형 자료를 처리하는 One-hot-encoding을 사용하는 경우 메모리 문제가 발생
    - 차원을 줄이는 방법이 필요함(grouping, hashing, different encoding method etc.)

  - **11일차 데이터(test data)는 존재하지 않음을 가정한다**
    - 데이터에 새로운 level이 등장할 경우 모델의 예측력이 떨어질 수 있다
    - 새로운 데이터를 결측치로 간주하고 전처리 및 모델링의 진행 필요

2. 접근방법

▪ 접근과정 및 해결방안

● Cardinality가 높은 범주형 자료

- 범주형 자료를 처리하는 One-hot-encoding을 사용하는 경우 메모리 문제가 발생
- 차원을 줄이는 방법이 필요함(grouping, hashing, different encoding method etc.)

➢ Unique값이 높은 변수들에 대해 각각 내부에서 grouping 한다면 정보의 손실 발생

➢ Hashing역시 정보의 손실이 발생할 수 있고, 랜덤하게 level이 묶인다는 단점이 있음

➢ **Binary Encoding 또는 BaseN Encoding은 차원도 줄이고 정보의 손실을 줄임(XGBoost)**

➢ **혹은 범주형 데이터 활용에 최적화된 모델을 활용(Catboost, FFM)**

2. 접근방법

▪ 접근과정 및 해결방안

● 11일차 데이터(test data)는 존재하지 않음을 가정한다

- 데이터에 새로운 level이 등장할 경우 모델의 예측력이 떨어질 수 있다
- 새로운 데이터를 결측치로 간주하고 전처리 및 모델링의 진행 필요

➢ 결측치를 처리하기 위해 train 변수의 level 별로 grouping 한 후 새로운 변수를 만들 수 있다

➢ 하지만 test 데이터에 새로운 변수를 추가하는 전처리 하는 방식은 시간이 오래 걸린다

➢ **결측치를 최적값으로 자동 기입해주는 모델을 활용(XGBoost, CatBoost, FFM)**

3. 데이터 전처리

- 데이터 전처리 과정

A. Log(train, test) 데이터에서 필요 없는 변수 제거 및 정제

B. Audience Profile 의 활용

C. 각 모델별로 필요한 데이터 형식 정리

3. 데이터 전처리

A. Log(train, test) 데이터에서 필요 없는 변수 제거 및 정제

- **Event_datetime**
  - 기존의 '시간'변수는 datetime 형태로 linear format이기 때문에 반복이 존재할 수 있도록 circular format으로 재구성 해야 한다
  - '연','월','일','요일','시','분','초'의 candidates 중에서, 현재 주어진 데이터의 양 (10일분) 과 과적합문제를 고려하여 '요일', '시'의 형태로 재구성함
- **Device_country**
  - 지역별로 시간이 달라지지 않음을 확인했기 때문에 device_region 변수를 변형 없이 유지
  - device country 는 level이 하나이기에 제거
  - 나라는 하나지만 지역과 도시개수는 복수이므로 device_region 변수와 device_city 변수유지
- **Bid_id(5,500,000개의 인스턴스의 id값이므로 제거)**

3. 데이터 전처리

B. Audience Profile 의 활용

목 표 : Audience Profile을 활용해 device_ifa별 CTR을 구한 후 로그데이터(train, test)에 새로운 변수로 추가

기대효과 : 클릭에 영향을 주는 각 Audience별 특징을 모델학습에 반영할 것으로 예상

- **문제점 1: AP는 device_ifa 기준상 train 데이터와 매칭 비율이 너무 낮아 곧바로 변수로 추가하기 어렵다**
  - 즉 AP의 변수들로 audience 개인의 특성에 따른 ctr을 예측한다
  - Audience 별 mean ctr을 구해서 train 데이터와 매칭 되지 않는 경우 평균으로 impute

- **문제점 2: install_pack(약4만개)와 cate_code(241개)는 바로 독립변수로 쓰기에는 연산, 메모리 문제가 있다**
  - install_pack는 hashing을 통해 256개로 줄이고, 각각 k-means clustering을 통해 정보를 압축

3. 데이터 전처리

B. Audience Profile 의 활용

< install_pack의 hashing 및 벡터화 예시>

device_ifa	10111	10101	01101
A	2	5	4
B	4	2	5
C	1	1	1

< cate_code의 벡터화 예시>

device_ifa	Category1	Category2	Category3
A	2	2	1
B	0	1	0
C	1	1	2

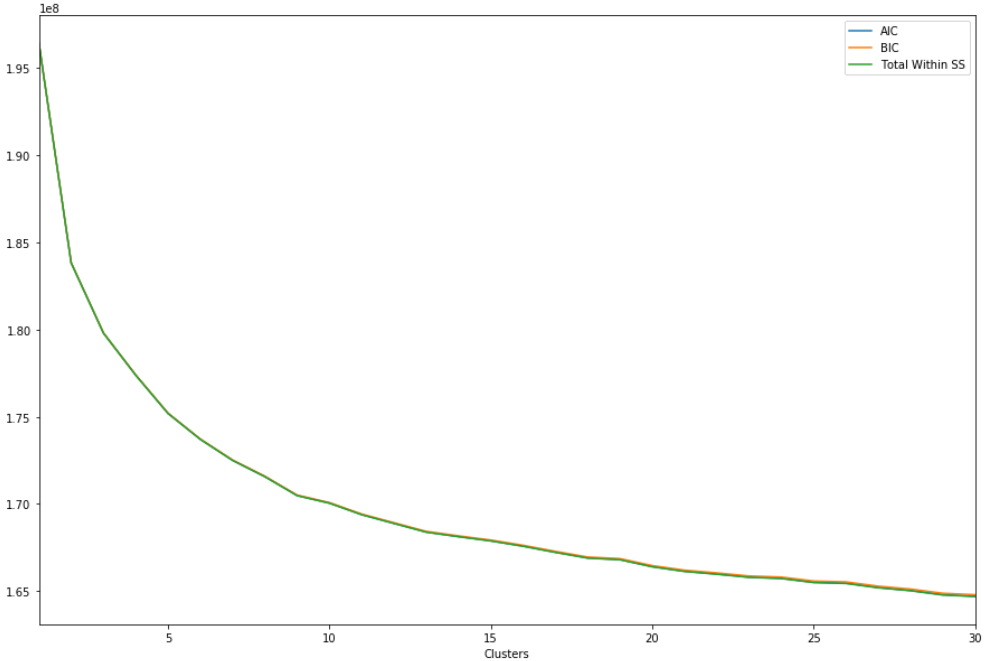
* 해당 카테고리 값을 가지지 않으면 해당 차원의 값은 0

K-means

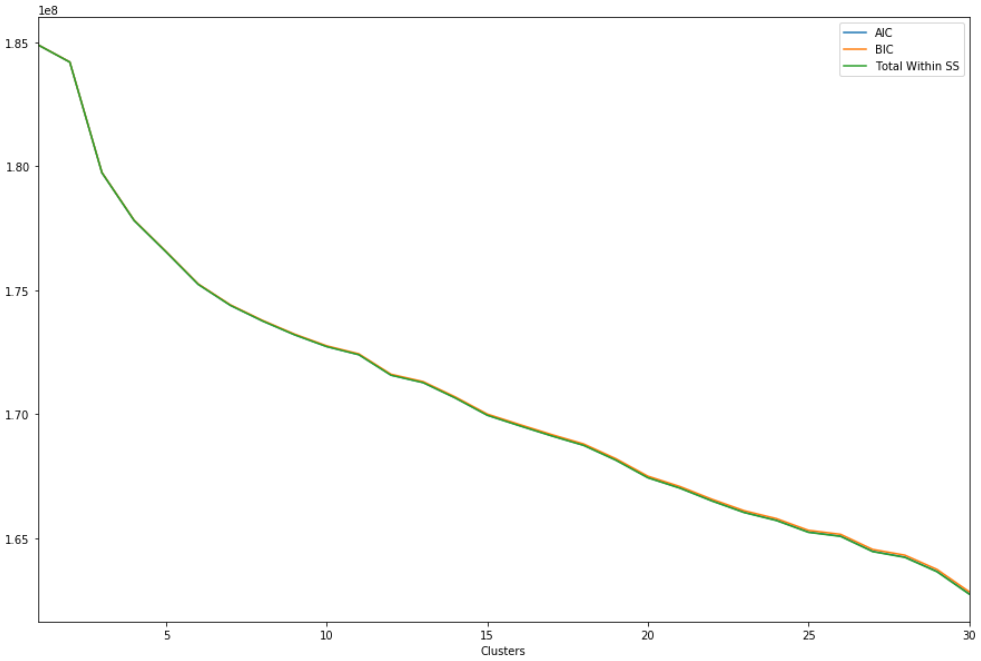
device_ifa	app_clusts	cate_clusts
A	1	5
B	3	2
C	2	6

B. Audience Profile 의 활용

- K-means AIC Graph (cluster 개수 별 AIC 비교)



<'app_cluster' → 10개 cluster>



<'cate_cluster' → 7개 cluster>

B. Audience Profile 의 활용

- 이후, Gender, Marry, Age, app_cluster, cate_cluster를 이용하여 train set에 나와 있는 device_ifa 별 CTR을 linear regression으로 예측
- 해당 CTR(y hat) 값을 표준화하여 5개의 Category(Very High, High, Average, Low, Very Low)로 분할

- **CTR을 해당 audience의 특성을 통해 예측했을 때 다른 사람들보다 클릭할 확률이 높은 군에 속하는지 낮은 군에 속하는지에 대한 정보로 압축**

C. 각 모델별로 필요한 데이터 형식 정리

XGB용 데이터

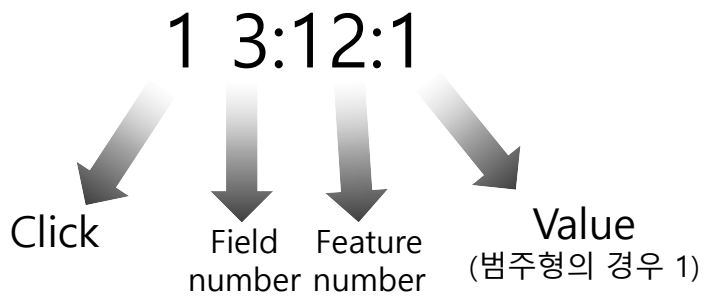
- Binary Encoding

CatBoost용 데이터

- Cat_feature로 변수 변경
- 데이터 타입 'Categorical'로 변환

FFM용 데이터

- 데이터를 LibFFM 형식으로 변환



4. 모델링 및 튜닝

- 모델링 과정

A. Model Selection

B. Feature Selection

C. Subset Selection

D. CatBoost

E. FFM (Field Aware Factorization Machine)

F. Model Ensemble

4. 모델링 및 튜닝

A. Model Selection

	XGBoost	CatBoost	FFM
Default Setting Logloss	0.25023	**0.24540**	**0.24380**
Average Training Time	128 mins	135 mins (CPU) 18 mins (GPU)	**10 mins**

- 디폴트 파라미터로 모델을 학습시켰을 경우 XGBoost가 가장 낮은 성능을 보였다
- CatBoost와 FFM의 성능이 비슷해 이 둘 모델을 선택

4. 모델링 및 튜닝

B. Feature Selection

	Feature Id	Importances
0	media_id	28.553907
1	media_bundle	16.463068
2	advertisement_id	11.964431
3	device_ifa	9.272605
4	adset_id	7.828770
5	media_name	3.695156
6	publisher_name	3.262260
7	device_os_version	3.115455
8	ssp_id	2.948343
9	device_model	2.681418
10	device_connection_type	2.587498
11	device_language	1.476491
12	placement_type	1.231236
13	datetime-hour	0.962574
14	device_make	0.797411
15	weekdays	0.790483
16	campaign_id	0.654754
17	device_carrier	0.602986
18	media_domain	0.550323
19	device_region	0.272668
20	publisher_id	0.204857
21	predicted_cate	0.071881
22	device_city	0.011424
23	device_os	0.000000

- CatBoost의 Feature importance를 출력한 결과 영향이 없는 변수들이 다수 있는 것을 확인
- Importance가 0.5 미만인 변수들을 제거한 결과 디폴트 보다 결과가 좋았지만 추가적으로 변수를 제거한 모델은 그렇지 못했다
- Feature Selection의 효과를 확인

Feature Selection	Test - Logloss
default	0.24558
Drop importance < 0.5	**0.24540**
Drop importance < 0.8	0.24545

C. Subset Selection

- **도메인적(직관적) 접근**

	pair_name	chi2	p	dof	smaller	cramerV
27	(campaign_id, device_os)	5.500000e+06	0.000000e+00	185	device_os	1.000000
19	(campaign_id, adset_id)	1.017500e+09	0.000000e+00	161135	campaign_id	1.000000
36	(campaign_id, advertisement_id)	1.595000e+08	0.000000e+00	5365	advertisement_id	1.000000
37	(adset_id, placement_type)	1.650000e+07	0.000000e+00	2613	placement_type	1.000000
53	(adset_id, advertisement_id)	1.595000e+08	0.000000e+00	25259	advertisement_id	1.000000
...	...	...	...	...	...	...
169	(device_carrier, device_make)	4.009511e+06	0.000000e+00	159430	device_make	0.049460
177	(device_make, device_region)	1.225011e+06	0.000000e+00	43806	device_region	0.038925
118	(media_domain, device_make)	8.529386e+05	0.000000e+00	47978	media_domain	0.031036
178	(device_make, device_city)	1.540327e+06	0.000000e+00	394850	device_make	0.030656
114	(media_domain, device_os)	1.436742e+03	1.751563e-203	161	device_os	0.016162

< 변수들 간의 Cramer V 표>

- OS가 다르면 다른 컬럼들도 그에 맞춰 변화할 가능성이 매우 높음.
- OS로 인해 컬럼들 간의 패턴이 생기게 된다는 것
- OS 외에는? 이를 알아보기 위해 변수 간의 Dependency(Cramer V)를 계산

4. 모델링 및 튜닝

C. Subset Selection

< 변수들의 Dependency Score 표>

Feature	Dependency score based on Cramer V
device_os	10.85084533333838
ssp_id	10.631603090253387
placement_type	10.764284055678383
media_id	10.013154910527854
…	…

- 다른 변수들이 해당 변수에 Dependent하는 경향이 크다는 건 위와 마찬가지로 그 컬럼에 따라 패턴이 생길 수 있다는 것
- 그럼 이 패턴을 더욱 정교하게 모델에 학습시키기 위해 해당 컬럼에 맞춰 Subset을 나눈 뒤, '전문' 모델을 생성 이후 각 모델의 예측 값을 logistic function을 이용해 평균을 구하는 앙상블 모델이 가능해진다
- Subset은 dependency 점수가 10점 이상인 변수들 중 상대적으로 적은 level을 갖고 있는

device_os, ssp_id, placement_type 을 각각 3~4개로 쪼개서 학습시키기로 결정

4. 모델링 및 튜닝

D. CatBoost

- Cardinality가 높은 범주형 데이터를 처리하는 용으로 만들어진 GBM기반 모델
- Cat_features 기능으로 효율적으로 차원을 줄이고 정보의 손실을 최소화
- Hyper-parameter tuning은 randomized search 활용

OS_A	OS_B	Ssp_1	Ssp_2	Ssp_3	Ssp_4	Place_1	Place_2	Place_3
0.245487	0.368897	0.0721962	0.4274828	0.300912	0.1506318	0.387319	0.080409	0.267619

	Full_ssp_id	Full_Placement_tpye
Submission score	**0.24550**	**0.24463**

* 성능 및 계산 속도의 한계로 최종 모델로는 사용하지 않음

E. FFM (Field Aware Factorization Machine)

$$\phi_{\mathrm{FFM}}(\boldsymbol{w}, \boldsymbol{x}) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\boldsymbol{w}_{j_1, f_2} \cdot \boldsymbol{w}_{j_2, f_1}) x_{j_1} x_{j_2},$$

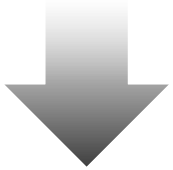
Juan, Y., Zhuang, Y., Chin, W. S., & Lin, C. J. (2016, September). Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 43-50).

- FFM(Field-aware Factorization Machines)는 선형 모델의 일종으로 단순 Poly2 선형 회귀와 다르게, Field(Colum) 및 Feature(Column 안의 category)의 **Latent Vector들 간의 곱으로 변수 간 상호작용을 설명**
- 이는 빠르고 효과적인 알고리즘으로 특히 CTR prediction에 좋은 성능을 보인다

E. FFM (Field Aware Factorization Machine)

< Validation - Logloss of each subsets >

OS_A	OS_B	Ssp_1	Ssp_2	Ssp_3	Ssp_4	Place_1	Place_2	Place_3
0.242271	0.368753	0.072354	0.430035	0.301177	0.149595	0.367319	0.079419	0.268616

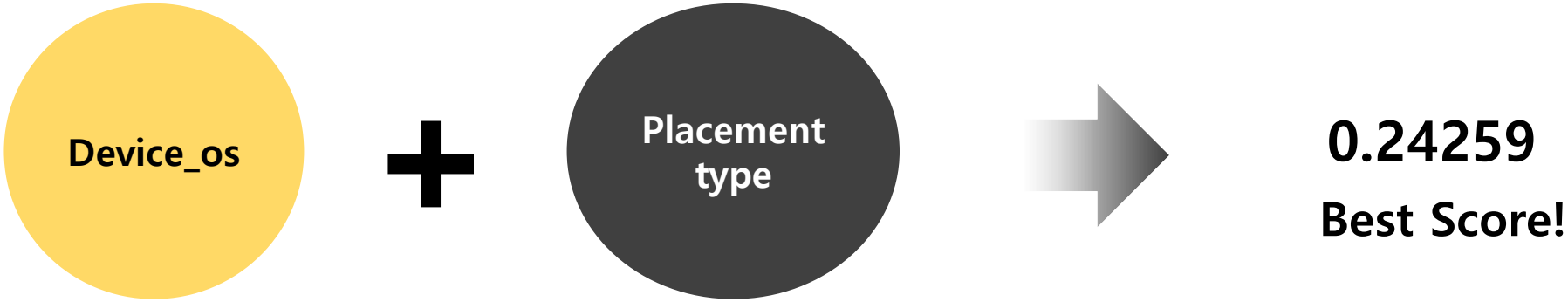


	Full_OS	Full_ssp_id	Full_Placement_tpye
Submission score	0.24352	0.24505	0.24327

4. 모델링 및 튜닝

F. Model Ensemble (Using only FFM model)

- **Ensemble of multiple subsets**



- Final Probability = $f\left(\frac{f^{-1}(\text{device\_os 모델의 prob}) + f^{-1}(\text{placement\_type 모델의 prob})}{2}\right)$ per each instance

  (f : logistic function)
- 시도해본 다양한 조합 중 가장 나은 성능을 보인 조합(그 결과 ssp_id subset 모델은 제외)

IGAWorks
Thank You

- **Team 오하이오** _ 윤창원, 이찬주, 김원호