07.17~30

최종 모델

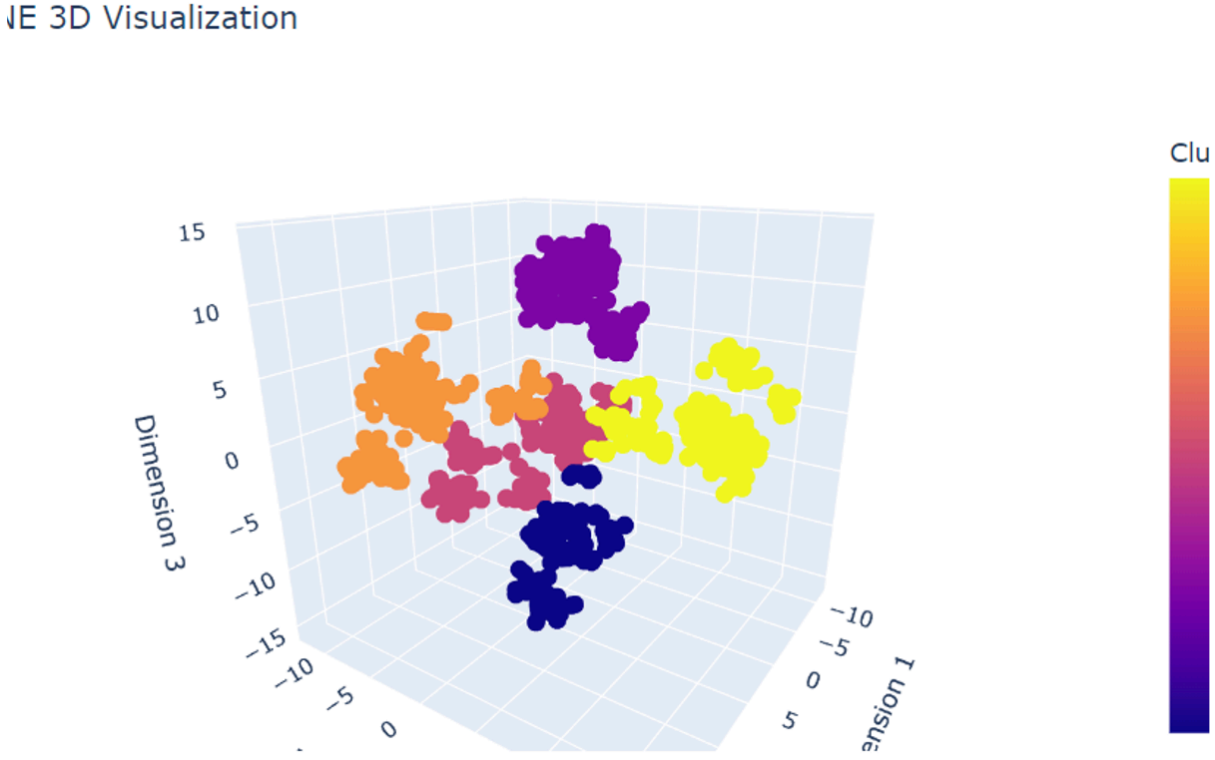
- LDA를 한 이후 데이터: 대략 1억 (행) x 171 (변수) 정도의 크기. 이 크기 그대로 XGB 모델을 실행하는 경우, 계속해서 dead kernel 문제가 발생

  → 현재는 500만 * 171 정도의 크기로 학습을 진행, 0.95034

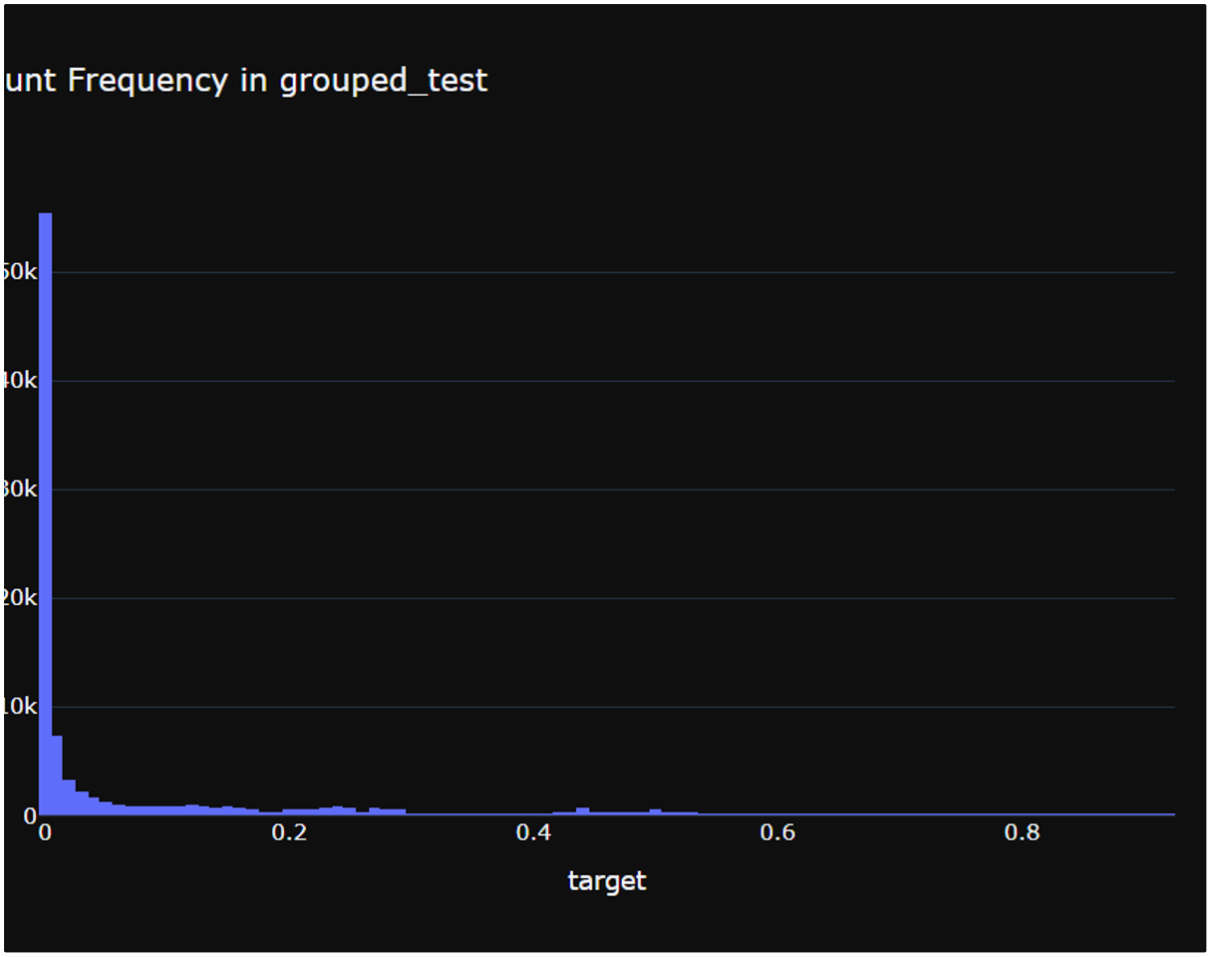
  categorical 변수 ip, device, channel, os 제거 이후 0.94886

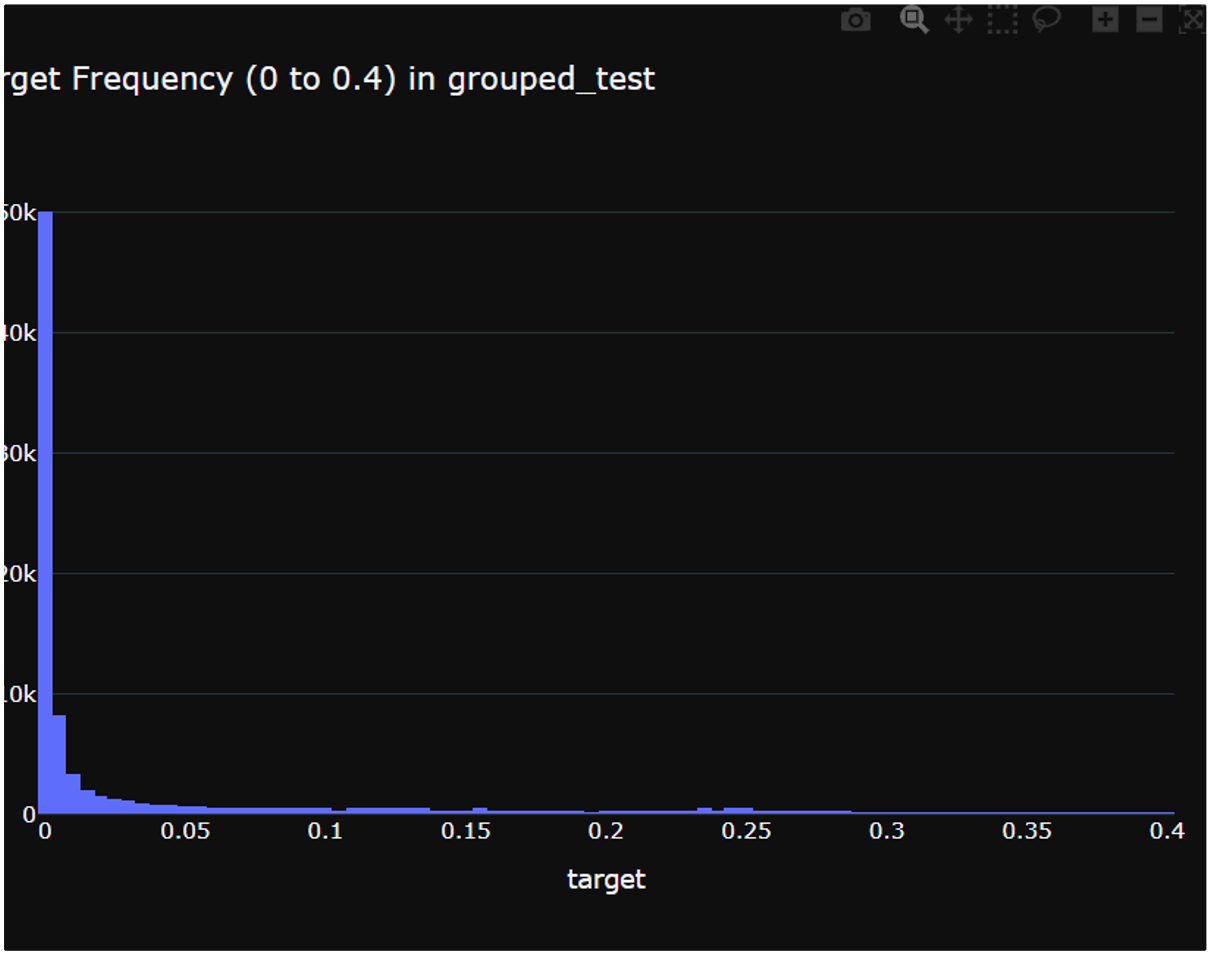
LDA 클러스터

- 클릭 중 랜덤 1000개



평균 예측 target 값 분포


unt Frequency in grouped_test
50k
40k
30k
20k
10k
0
0
0.2
0.4
0.6
0.8
target

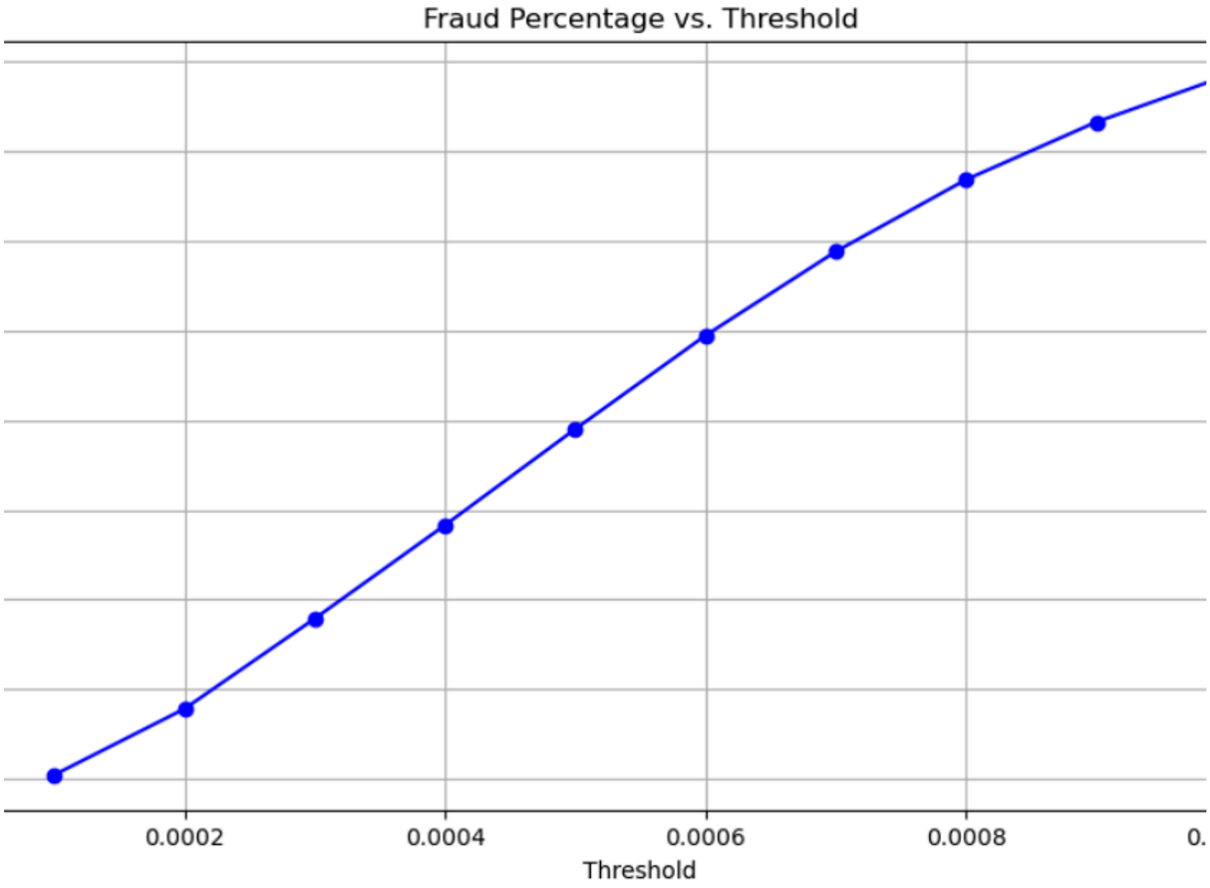

rget Frequency (0 to 0.4) in grouped_test
50k
40k
30k
20k
10k
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
target

유저 비율

- 전체 클릭 수: train(`184,903,890`) + test(`18,790,469`) = `203,694,359`

	ip	count	ratio
0	5348	1421256	0.006977
1	5314	1334383	0.006551
2	73516	839540	0.004122
3	73487	832720	0.004088
4	53454	559689	0.002748
...	...	...	...
95	118367	79588	0.000391
96	118229	79196	0.000389
97	37948	79025	0.000388
98	118252	78745	0.000387
99	118284	78309	0.000384

threshold에 따른 fraud 유저 비율


Fraud Percentage vs. Threshold
0.0002
0.0004
0.0006
0.0008
Threshold

피쳐 중요도


B Feature Importance
topic_app+os+_2
topic_app+os+_4
topic_app+ip+_1
topic_app+ip+_2
topic_app+device+_5
topic_channel+os+_2
topic_os+app+_3
next_click_time_diff_ip
pic_device+channel+_3
topic_channel+app+_4
pic_channel+device+_1
topic_os+channel+_2
app
topic_app+channel+_2
topic_app+ip+_4
t_click_time_diff_ip_app
pic_channel+device+_5
topic_app+channel+_4
topic_app+os+_5
topic_app+ip+_3
41
10.550000190734863
3.640000104904175
3.0899999141693115
2.190000057220459
1.5
1.25
1.2100000381469727
1.1399999856948853
1.1299999952316284
0.7300000190734863
0.6899999976158142
0.6600000262260437
0.6399999856948853
0.6100000143051147
0.5699999928474426
0.5600000023841858
0.5
0.46000000834465027
0.4399999976158142
0
0.1
0.2
0.3
0.4
Importance
Feature
topic_app+os+_2
topic_app+os+_4
topic_app+ip+_1
topic_app+ip+_2
topic_app+device+
topic_channel+os+
topic_os+app+_3
next_click_time_dif
topic_device+chann
topic_channel+app
topic_channel+devi
topic_os+channel+
app
topic_app+channel
topic_app+ip+_4
next_click_time_dif
topic_channel+devi

threshold

- train데이터에서 0만 나오는 유저(ip) 수 : `32,358` (전체 유저 수 : `277,396` 비율 11.66%)

	ip	count	target
1	5	24	0.0
11	33	451	0.0
16	55	396	0.0
22	81	519	0.0
29	104	1373	0.0
...	...	...	...
277330	364713	45	0.0
277361	364744	394	0.0
277366	364749	31	0.0
277378	364761	226	0.0
277392	364775	24	0.0

32358 rows × 3 columns

```
zero_target_groups['count'].describe()
```

```
count    32358.000000
mean       189.876136
std        385.280619
min          1.000000
25%         13.000000
50%         50.000000
75%        206.000000
max      12489.000000
Name: count, dtype: float64
```

- train에만 등장하는 유저: `239,232`
- test에만 등장하는 유저: `55,772`
- train & test 둘 다 등장하는 유저: `38,164`
- 총합하면 전체 유저: `333,168`
- 전체 유저의 5% :~ `16,650` 명

→ fruad 규칙:

type1 (16,308명): train에서 50번 이상 클릭 되고, 단 한번도 is_attributed가 1이 나오지 않은 ip

type2 (342명) : train에서 걸러지지 않은 ip중, test에서 50번 이상 클릭되고 평균 예측 `target` 하위 342명 유저

threshold = `0.000117` 이하인 342명 fraud 처리

- output: fraud_set.pkl → fraud로 분류한 ip set

main

- 목표

Our goal is to maximize the **expected total number of infections caught** at the border, i.e.,

$$\mathbb{E}\left[\sum_{t=1}^{\mathcal{T}}\sum_{x\in\mathcal{X}}\sum_{e=1}^{\mathcal{E}}T_{xe}(t)R_x(t)\right].$$

→ 클릭 하나 당 검사 여부 결정: fraud 검사 시도 횟수 * 적중률(= 검사로 걸러내는 횟수) 최대화

- 그룹 나누는 방식
  - 알고리즘으로는 ip device 등을 기반으로 20-30개 그룹으로 나누고..
  - 6시간마다 해당 20-30개 그룹에 검사 리소스를 분배하는 방식

    → train, test 를 다 합쳐도 day 6~10 총 5일. 6시간 주기이므로 총 20번의 사이클. 충분할까?

<참고> 논문에서 타입을 클러스터링하는 룰

- 각 승객의 나라를 좀 더 세분화하여 '지역'의 개념 부여
  - 총 C개의 나라, U개의 지역으로 분리 (C = 170, U = 17,000)
- 모든 지역을 타입으로 분류하는 것은 아니고, LASSO를 통해 변수의 계수가 양수인 지역만 채택

> 💡 '나라 id (1~C)' + '지역 id (1 ~ U) 중 LASSO를 통해 채택된 변수'

- 이렇게 정의된 유형은 매주 업데이트
- 실제 LASSO 분석에서는

  1) 나라(1~C) x 성별(3가지),

  2) 나라(1~C) x 나이(0대, 10대, .., 90대 → 10개의 그룹으로 분리)

  이 두 가지 조합에 대해서도 변수로 추가하여 LASSO를 했으나 모든 변수의 계수가 0이 되어 실제 모델에서는 제외.

- 각 그룹마다 beta distribution을 이용해 fraud 일 확률을 추정
  - 베타 분포 파라미터 초기값 세팅, 이 세팅은 매일 리셋이 되는지
  - 베타 분포 파라미터를 어떻게 업데이트를 해줄 것인지
  - 눈문 P.4 ~ P.7 참고

$$P_k = \sum_{t'=t-16}^{t-3} P_k(t'),$$

the total number of type k passengers that tested positive over the past 14 days of test

$$N_k = \sum_{t'=t-16}^{t-3} N_k(t'),$$

the total number of type k passengers that tested negative over the past 14 days of test
iased, and natural estimate of the prevalence for type k is

$$\hat{r}_k^{naive} = \frac{P_k}{P_k + N_k}.$$

$$\sqrt{\text{Excess MSE of } \hat{r}_k^{naive} \text{ over Baseline}} = \sqrt{.000334} = 0.018,$$

is larger than the typical prevalence of most countries. In other words, any potential s
y washed out by noise.