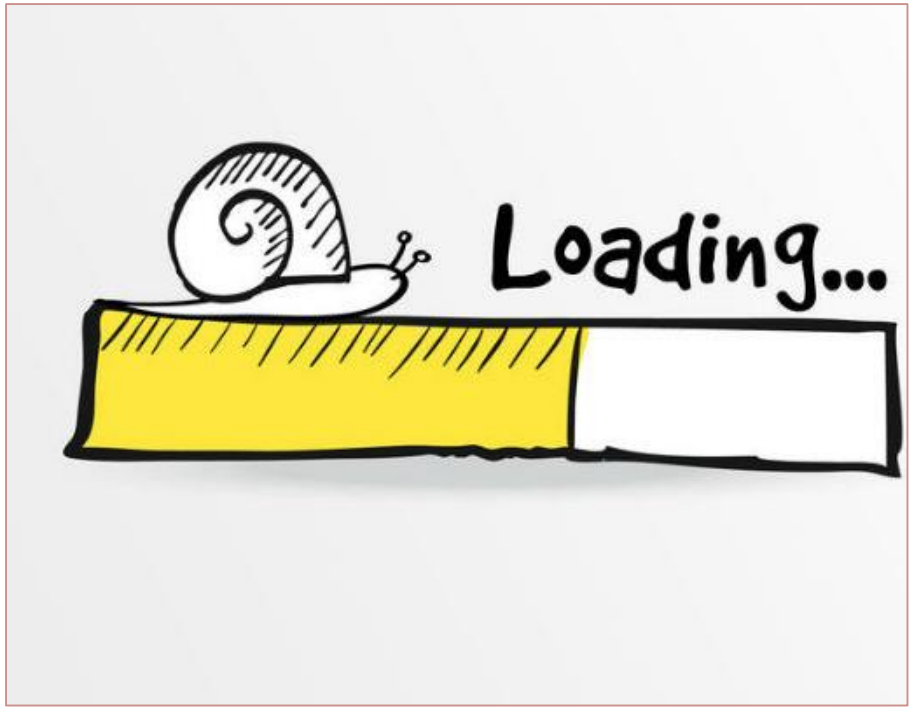
GAME BOT DETECTION

BAO3

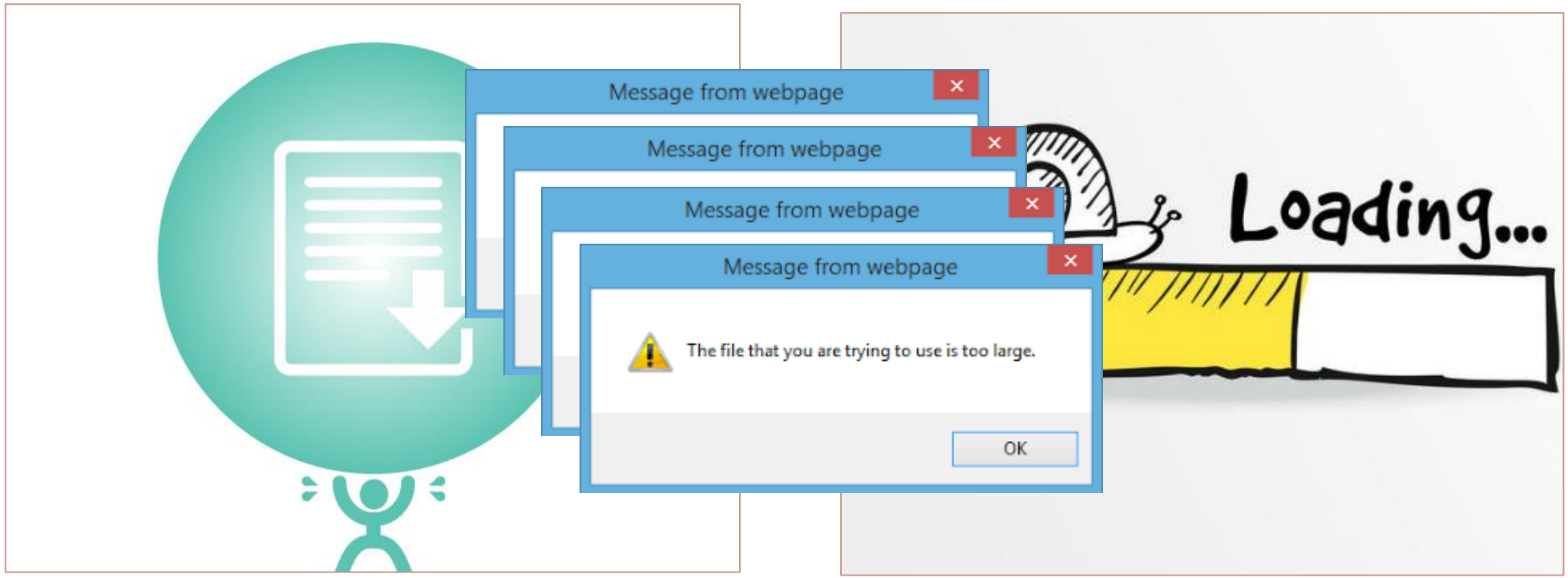
2019 11 22

FIRST APPROACH

FILE TOO BIG



FILE TOO BIG


Message from webpage
Message from webpage
Message from webpage
Message from webpage
The file that you are trying to use is too large.
OK
Loading...

SPLIT

- Computing resource와 Time resource의 전략적 활용 필요
- User account 기반 split **VS** Log number 기반 split
  - User account 기반 split
    - 행동 패턴 분석에 유리
  - Log number 기반 split
    - 관리해야 할 파일 개수가 적어지므로 File IO 속도에 유리
    - 하나의 행동을 기준으로 bot / human을 구분하기 용이함
  - Log number 기반 split 선택

PARALLELIZE

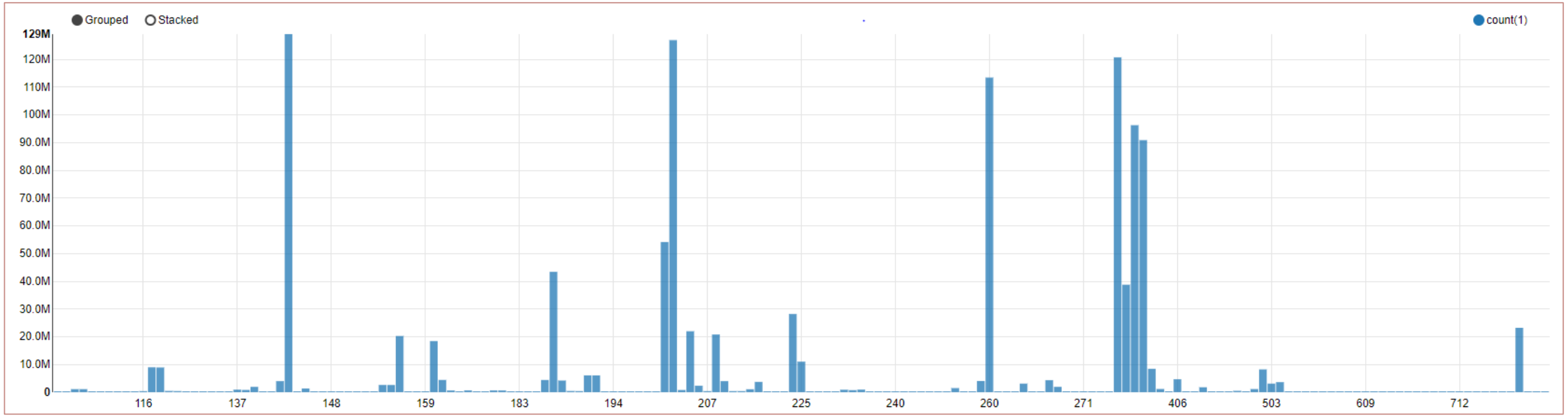
IDEATION

- 행동 기반의 분석
  - 단독 행동 타입 – 아이템 수집, 경험치 획득, 비행, …
  - 상호 작용 타입 – 거래, 파티플레이, …

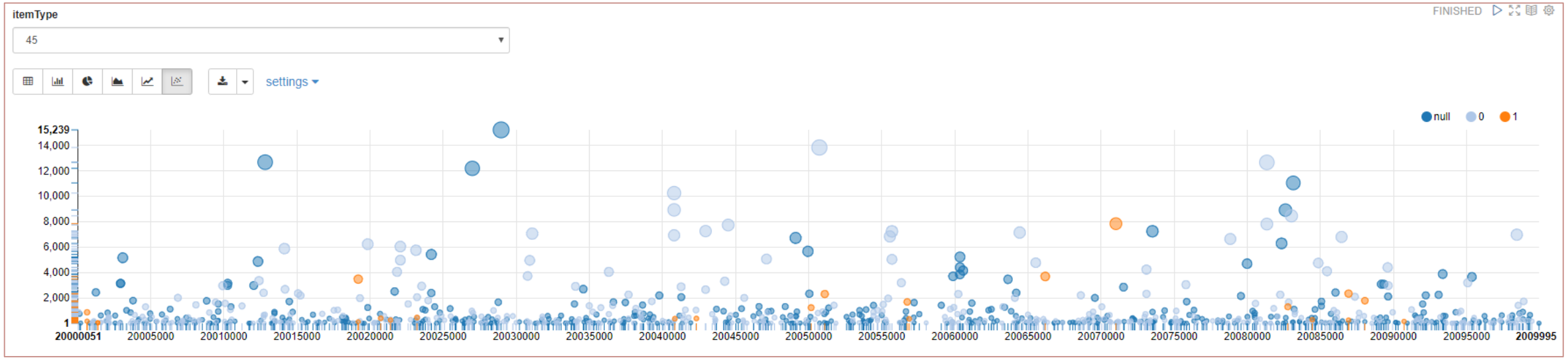
- Bot의 목적
  - 돈 – 게임머니 / 현실머니
  - 경험치

- Bot이 하기 어려운 행동
  - 비행
  - 대화

IDEATION


Grouped
Stacked
count(1)
129M
120M
110M
100M
90.0M
80.0M
70.0M
60.0M
50.0M
40.0M
30.0M
20.0M
10.0M
0
116
137
148
159
183
194
207
225
240
260
271
406
503
609
712

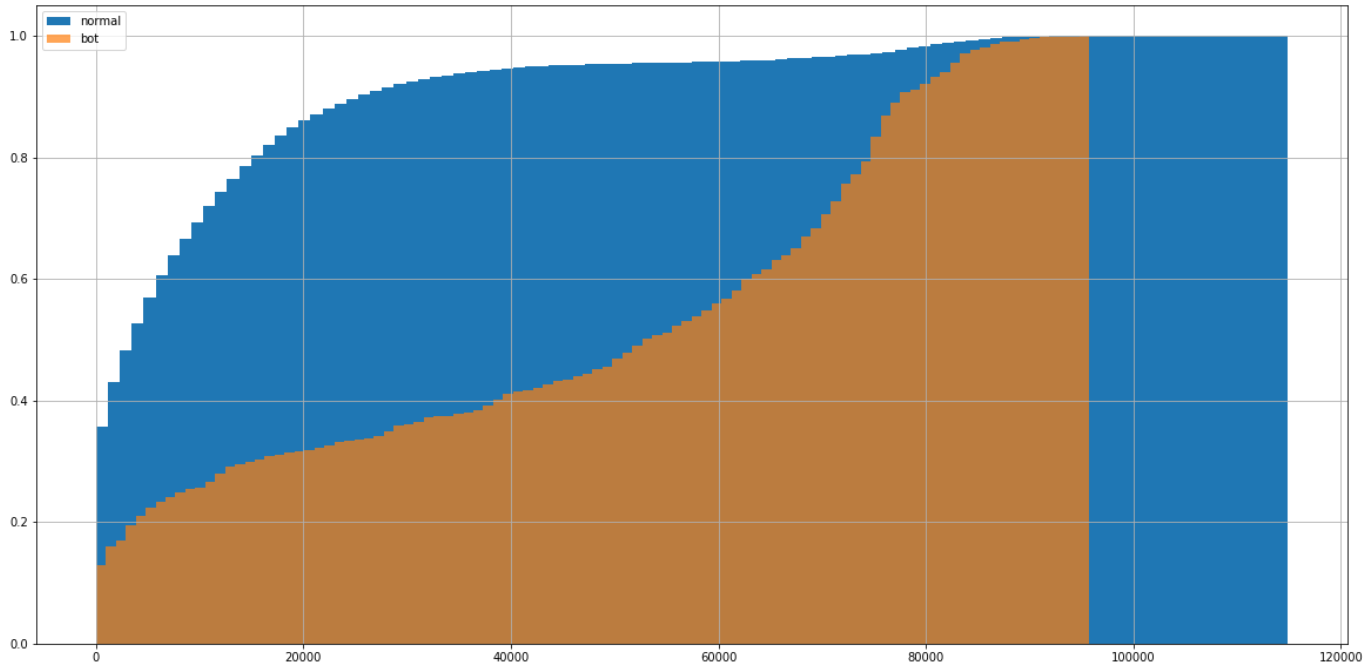
IDEATION


itemType
45
settings
FINISHED
null
0
1
15,239
14,000
12,000
10,000
8,000
6,000
4,000
2,000
1
20000051
20005000
20010000
20015000
20020000
20025000
20030000
20035000
20040000
20045000
20050000
20055000
20060000
20065000
20070000
20075000
20080000
20085000
20090000
20095000
2009995

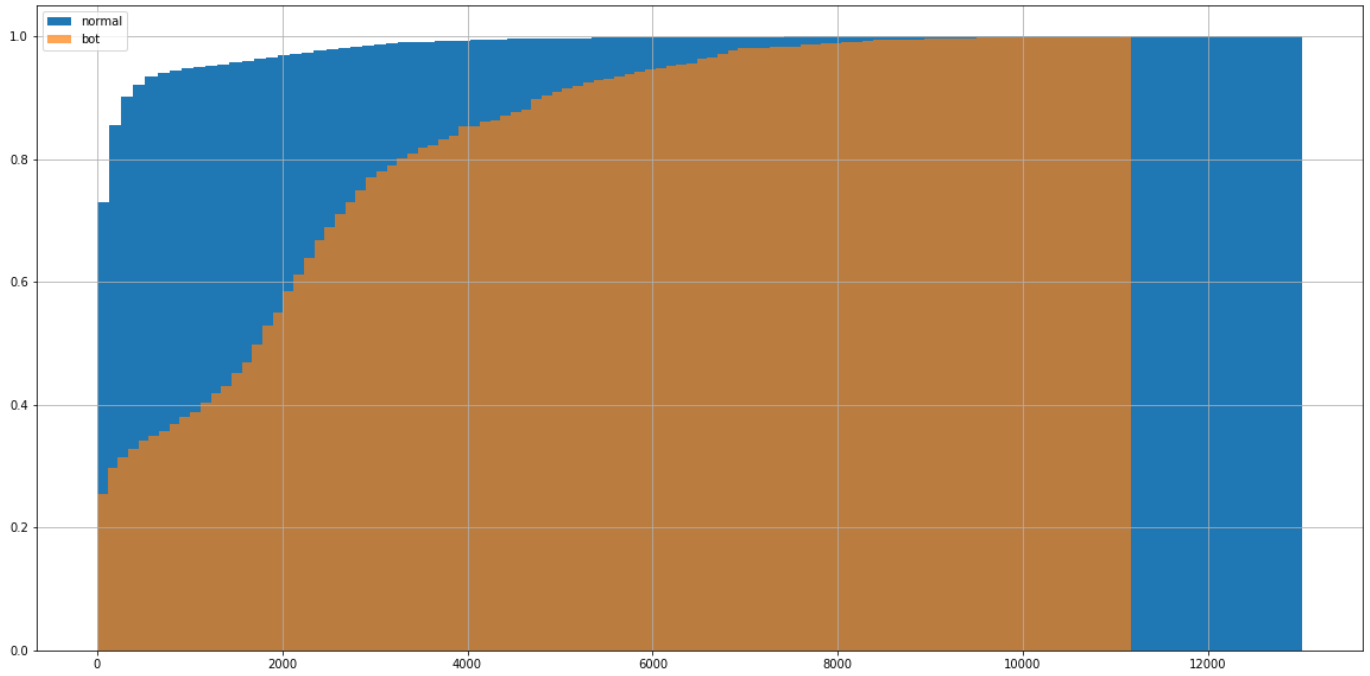
FEATURE SELECTION

FEATURE SELECTION

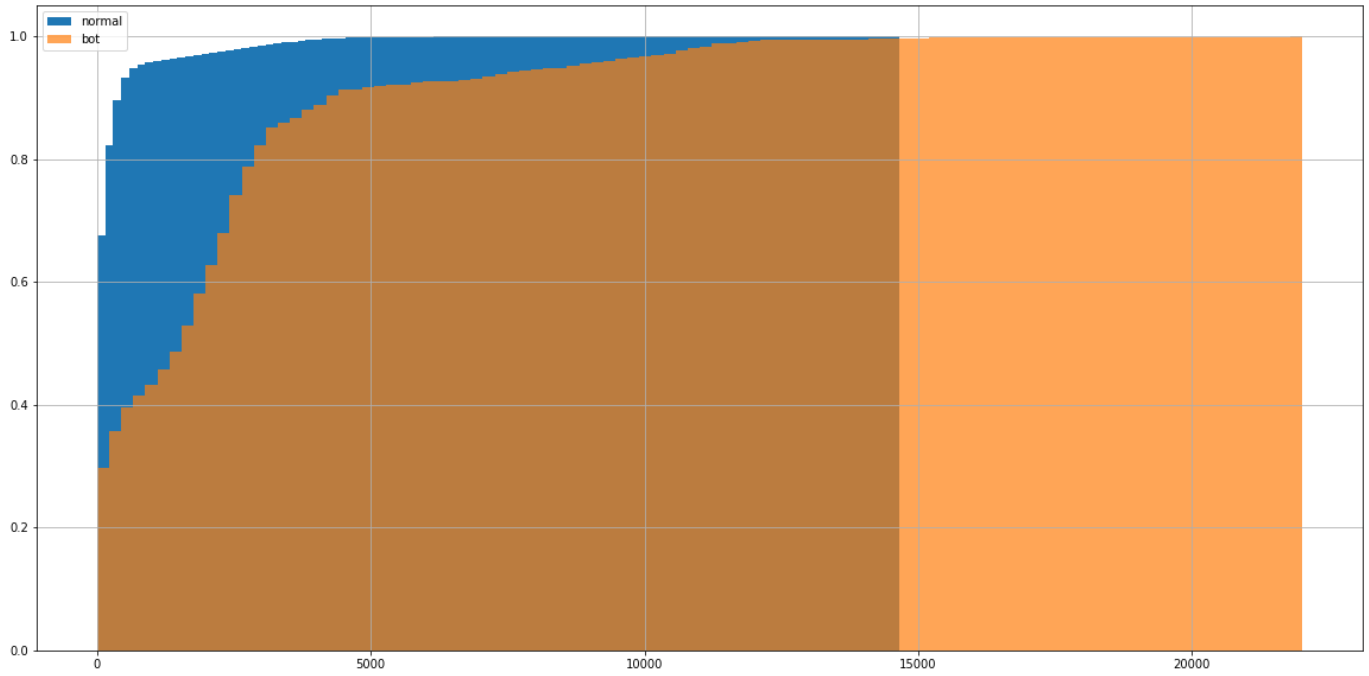
- Single Action
  - play_time – 캐릭터의 전체 플레이 시간
  - playtime_per_day – 캐릭터의 전체 플레이 시간 / 접속 로그가 있는 날의 수
  - all_count – 로그 타입에 관계없이 수집한 캐릭터 별 로그의 총 숫자
  - max_level – 로그에 기록된 캐릭터의 레벨 중 가장 높은 값
  - exp_ratio / exp_per_day – 경험치 획득 로그의 수를 총 로그 개수로 나눈 것, 접속 로그가 있는 날의 수로 나눈 것
  - money_ratio / money_per_day – 키나 획득 로그의 수를 총 로그 개수로 나눈 것, 접속 로그가 있는 날의 수로 나눈 것
  - item_ratio / item_per_day – 아이템 획득 로그의 수를 총 로그 개수로 나눈 것, 접속 로그가 있는 날의 수로 나눈 것
  - total_login – 로그인 로그의 수



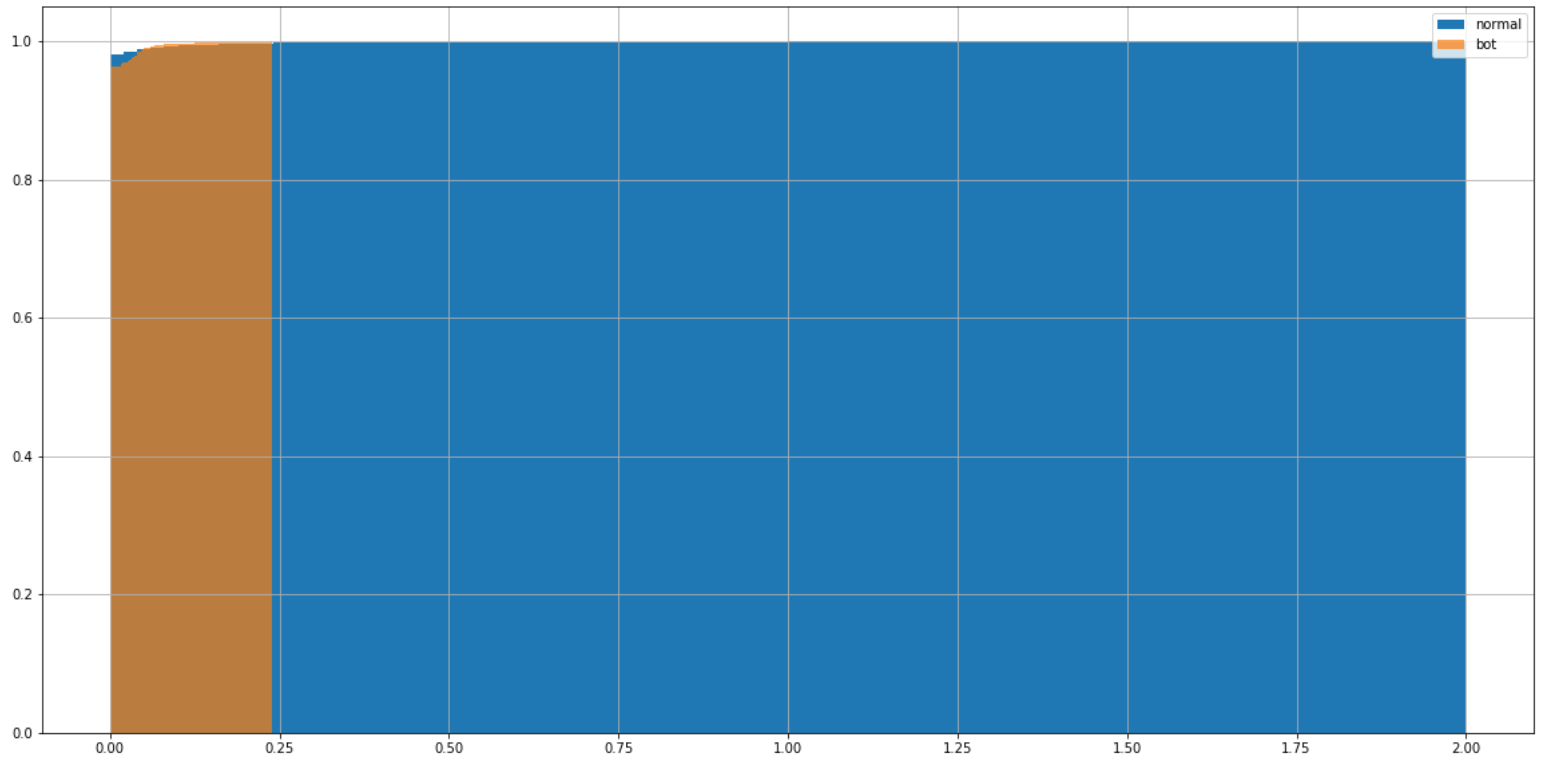
PLAYTIME PER DAY



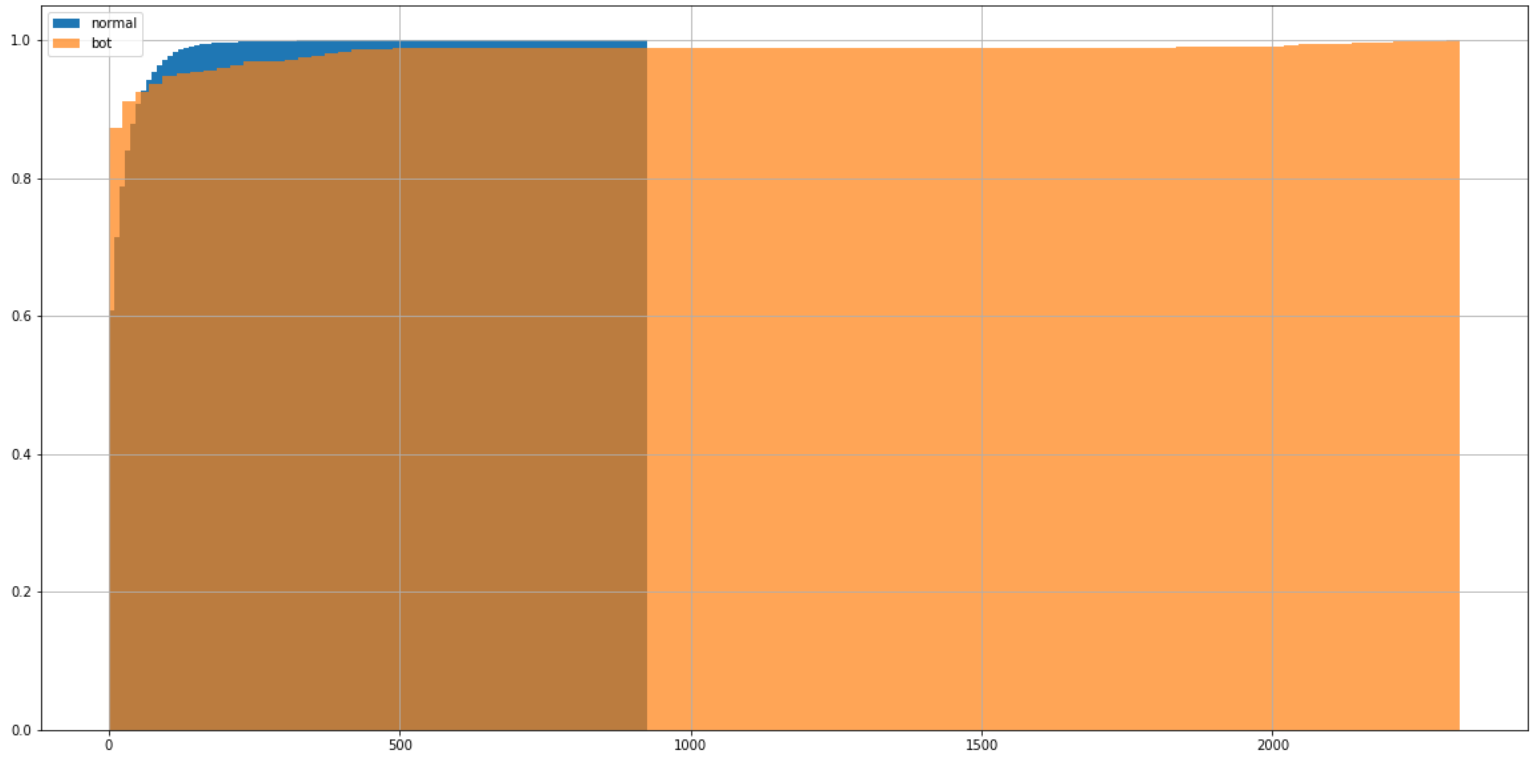
ITEMS PER DAY



EXP PER DAY



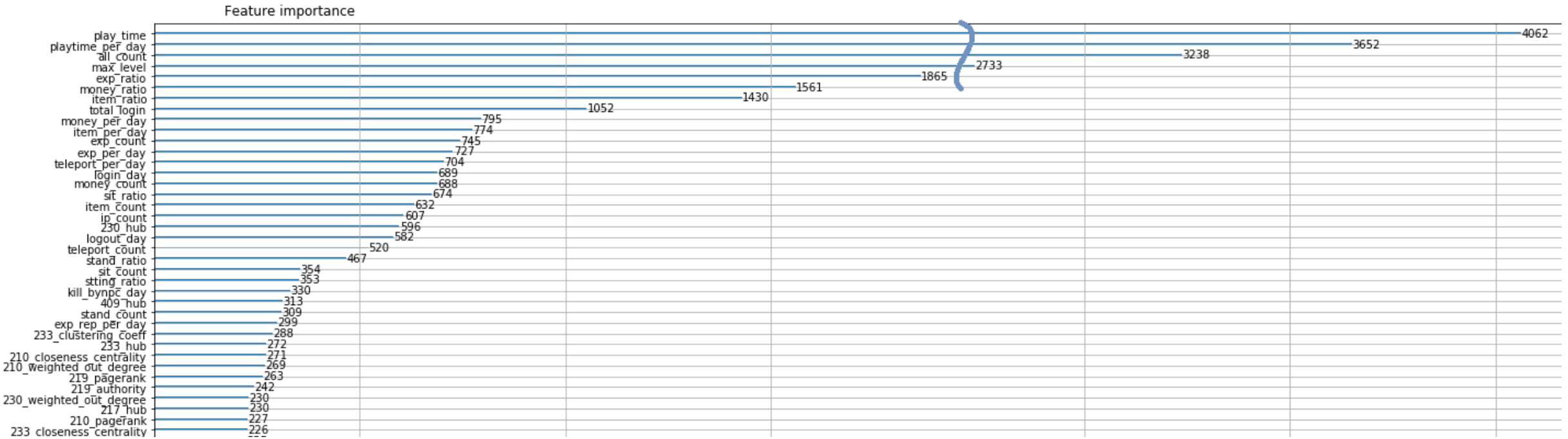
CAPTCHA INCORRECT RATIO



FLIGHT PER DAY

FEATURE SELECTION

- Linked Action
  - 우편 받음
  - 스킬 취소 (외부 영향)
  - 판매대행 구입
  - 거래 주기
  - 거래 완료
  - 거래 시작


Feature importance
play_time
playtime_per_day
all_count
max_level
exp_ratio
money_ratio
item_ratio
total_login
money_per_day
item_per_day
exp_count
exp_per_day
teleport_per_day
login_day
money_count
sit_ratio
item_count
ip_count
230_hub
logout_day
teleport_count
stand_ratio
sit_count
sting_ratio
kill_bynpc_day
409_hub
stand_count
exp_rep_per_day
233_clustering_coeff
233_hub
210_closeness_centrality
210_weighted_out_degree
219_pagerank
219_authority
230_weighted_out_degree
217_hub
210_pagerank
233_closeness_centrality
4062
3652
3238
2733
1865
1561
1430
1052
795
774
745
727
704
689
688
674
632
607
596
582
520
467
354
353
330
313
309
299
288
272
271
269
263
242
230
230
227
226

TRAIN & TEST

ALGORITHMS SELECTION

Classifier	Overall accuracy
Logistic regression	0.942
Decision tree	0.930
Random forest	0.969
XGBoost	0.981

TEST

- 8000여 건의 데이터
- 단순 데이터 분리(7:3)는 어렵다고 판단, K-Fold 사용

```
%python
res_xgb = []
f1_xgb = []
for train_index, test_index in gkf.split(training, groups=training.actor_account):
    X_train = training.iloc[train_index].drop(columns=['class', 'actor', 'actor_account'])
    X_test = training.iloc[test_index].drop(columns=['class', 'actor', 'actor_account'])
    account_test = pd.DataFrame(training.iloc[test_index]['actor_account'])

    y_train = training.iloc[train_index]['class']
    y_test = training.iloc[test_index][['actor_account', 'class']].groupby('actor_account').max()
    xgbc.fit(X_train, y_train)
    account_test['pred'] = xgbc.predict(X_test)
    y_pred = account_test.groupby('actor_account').max()
    res_xgb.append(accuracy_score(y_test, y_pred))
    f1_xgb.append(f1_score(y_test, y_pred))
print(res_xgb)
print(np.mean(res_xgb))
print(f1_xgb)
print(np.mean(f1_xgb))
```

FINISHED

```
[0.9888475836431226, 0.9826732673267327, 0.980246913580247, 0.9765721331689272, 0.9790382244143033, 0.9790382244143033, 0.9802712700369913, 0.9876543209876543, 0.9777777777777777, 0.9802712700369913]
0.9812390985387051
[0.9243697478991597, 0.8409090909090908, 0.8769230769230769, 0.819047619047619, 0.8349514563106796, 0.8571428571428572, 0.8596491228070176, 0.9, 0.8363636363636363, 0.8518518518518519]
0.8601208459254988
```

Metric	Result
Accuracy	0.979
Precision	0.951
Recall	0.777
F1	0.855

1 ROUND RESULT

FUTURE WORK

FUTURE WORK

- 패턴 분석
- 시간차 분석

- 다른 field에 적용

THANK YOU
Q&A