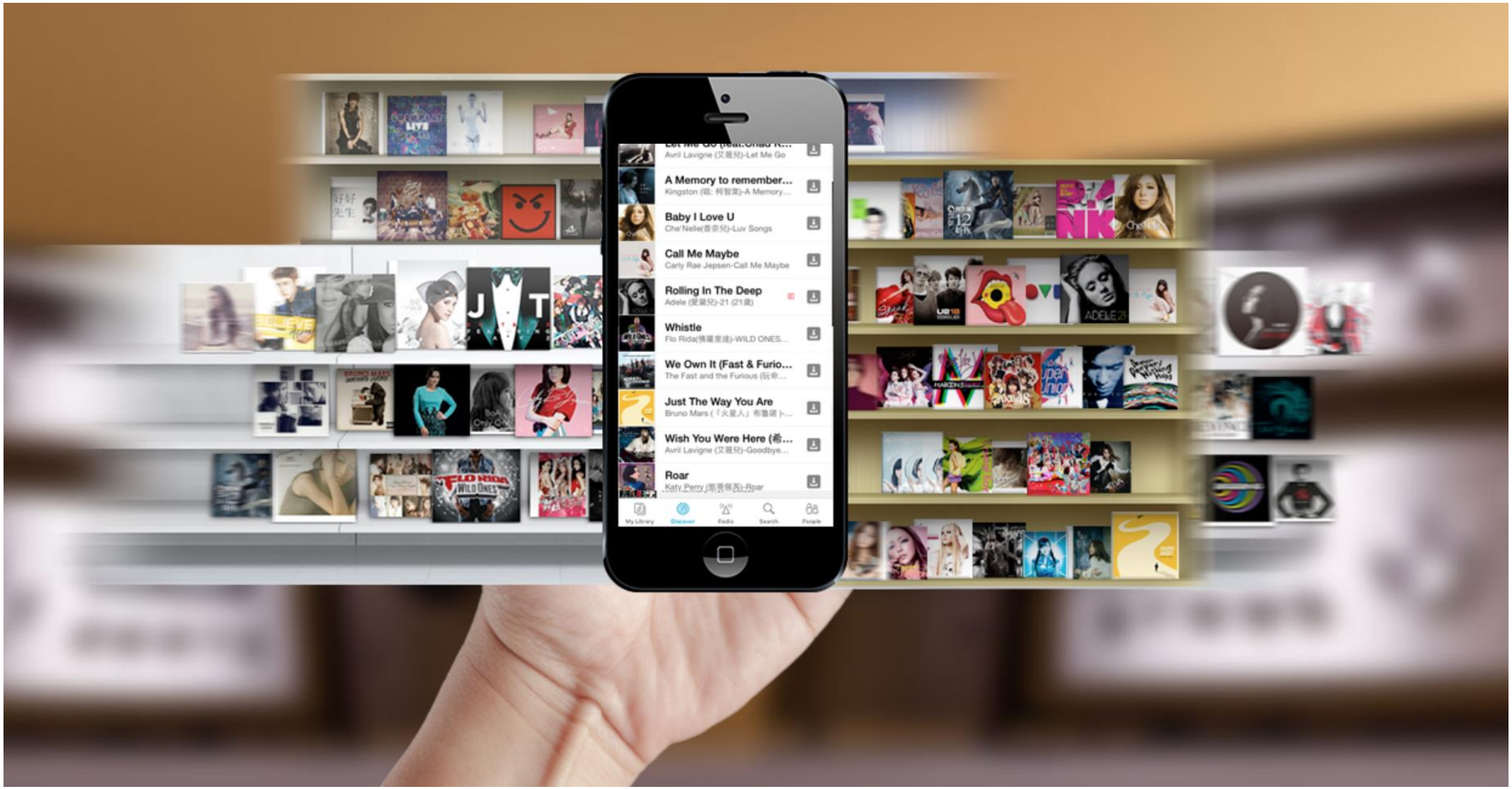
KKBox's Churn Prediction Challenge

Can you predict when subscribers will churn?

KKBOX is Asia's leading music streaming service



대회 설명

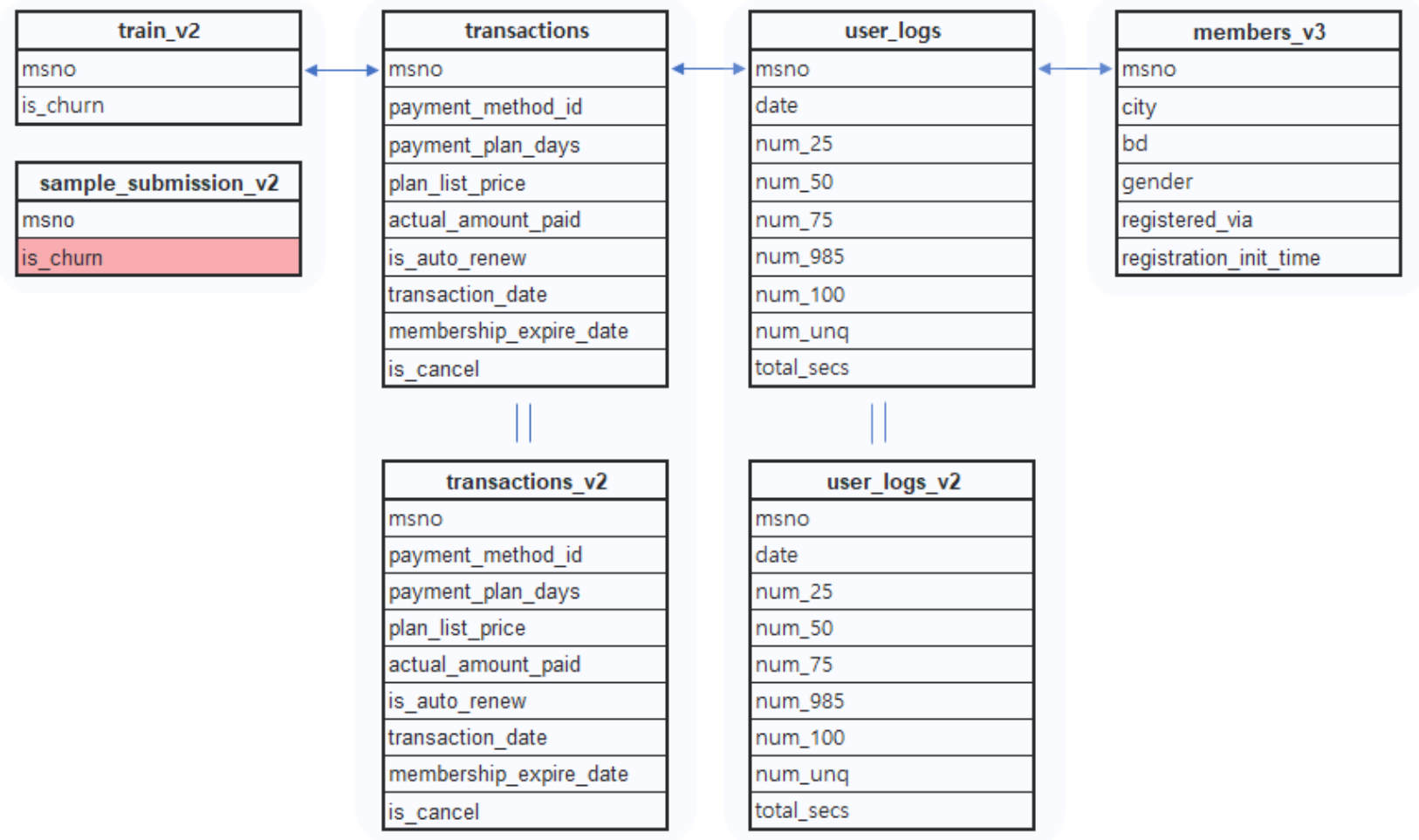
- Kkbox 유료 사용자가 3월에 이탈할 것인가? (Logloss 사용)

- 어머나? 정답셋 유출! 대회 폭파??

- 4월 이탈예측으로 변경

- 데이터 수정

데이터 설명


train_v2
msno
is_churn
sample_submission_v2
msno
is_churn
transactions
msno
payment_method_id
payment_plan_days
plan_list_price
actual_amount_paid
is_auto_renew
transaction_date
membership_expire_date
is_cancel
transactions_v2
msno
payment_method_id
payment_plan_days
plan_list_price
actual_amount_paid
is_auto_renew
transaction_date
membership_expire_date
is_cancel
user_logs
msno
date
num_25
num_50
num_75
num_985
num_100
num_unq
total_secs
user_logs_v2
msno
date
num_25
num_50
num_75
num_985
num_100
num_unq
total_secs
members_v3
msno
city
bd
gender
registered_via
registration_init_time

파생변수 생성

transactions
msno
payment_method_id
payment_plan_days
plan_list_price
actual_amount_paid
is_auto_renew
transaction_date
membership_expire_date
is_cancel

→

파생변수
msno
pay_30days
pay_7days
payday_sum_2015
payday_sum_2016
payday_sum_2017
avg_auto_renew
is_cancel
trans_cnt
fst_exp_date
lst_exp_date
sum_paydays
is_continuous
cancel_0_to_1_cnt
cancel_1_to_0_cnt
last_exp_month
unique_pay_method_cnt
max_pay_method
mean_diff_trans_exp
amt_per_day

user_logs
msno
date
num_25
num_50
num_75
num_985
num_100
num_unq
total_secs

→

파생변수
msno
sum_weight_logs_day
lst_day
log_count
mean_num_25
mean_num_50
mean_num_75
mean_num_985
mean_num_100
mean_total_secs
mean_rate_25
mean_rate_50
mean_rate_75
mean_rate_985
mean_rate_100
median_visit_days

최종 데이터셋

- X 변수 39개
- Y 변수 is_churn

고객ID

target

msno
is_churn
sum_weight_logs_day
lst_day
log_count
mean_num_25
mean_num_50
mean_num_75
mean_num_985
mean_num_100
mean_total_secs
mean_rate_25
mean_rate_50
mean_rate_75
mean_rate_985
mean_rate_100
median_visit_days
pay_30days
pay_7days
payday_sum_2015
payday_sum_2016
payday_sum_2017
avg_auto_renew
is_cancel
trans_cnt
fst_exp_date
lst_exp_date
sum_paydays
is_continuous
cancel_0_to_1_cnt
cancel_1_to_0_cnt
last_exp_month
unique_pay_method_cnt
max_pay_method
mean_diff_trans_exp
amt_per_day
city
bd
gender
registered_via
registration_init_time

User_logs
파생변수 15개

transaction
파생변수 19개

members
변수 5개

고려해야 할 점

- Unbalanced Data

Y변수 is_churn의 비율이 0:1 => 93:7
해결책 : upsampling, downsampling, threshold 조정
=> 0,1을 나누는 것 때문에 시행한다고 생각. Logloss score에서는 필요 X

- Feature Selection

해결책 : Stepwise, forward, backward deletion, lasso.
⇒Tree based Model은 변수를 줄였을 때 성능향상이 일어나기보다는
성능이 조금이라도 떨어진다.

최종 모델

Gbm, Randomforest 두 모델의 조화평균
=>0.10246 (public) / 0.10389 (private) 13등!

- 왜 선택했나?

Deeplearning => 0.12212
Xgboost => 0.11408
Gbm => 0.10376
Randomforest => 0.10322

최종 모델

Gbm, Randomforest 두 모델의 조화평균
=>0.10246 (public) / 0.10389 (private) 13등!

Gradient Boosting Model

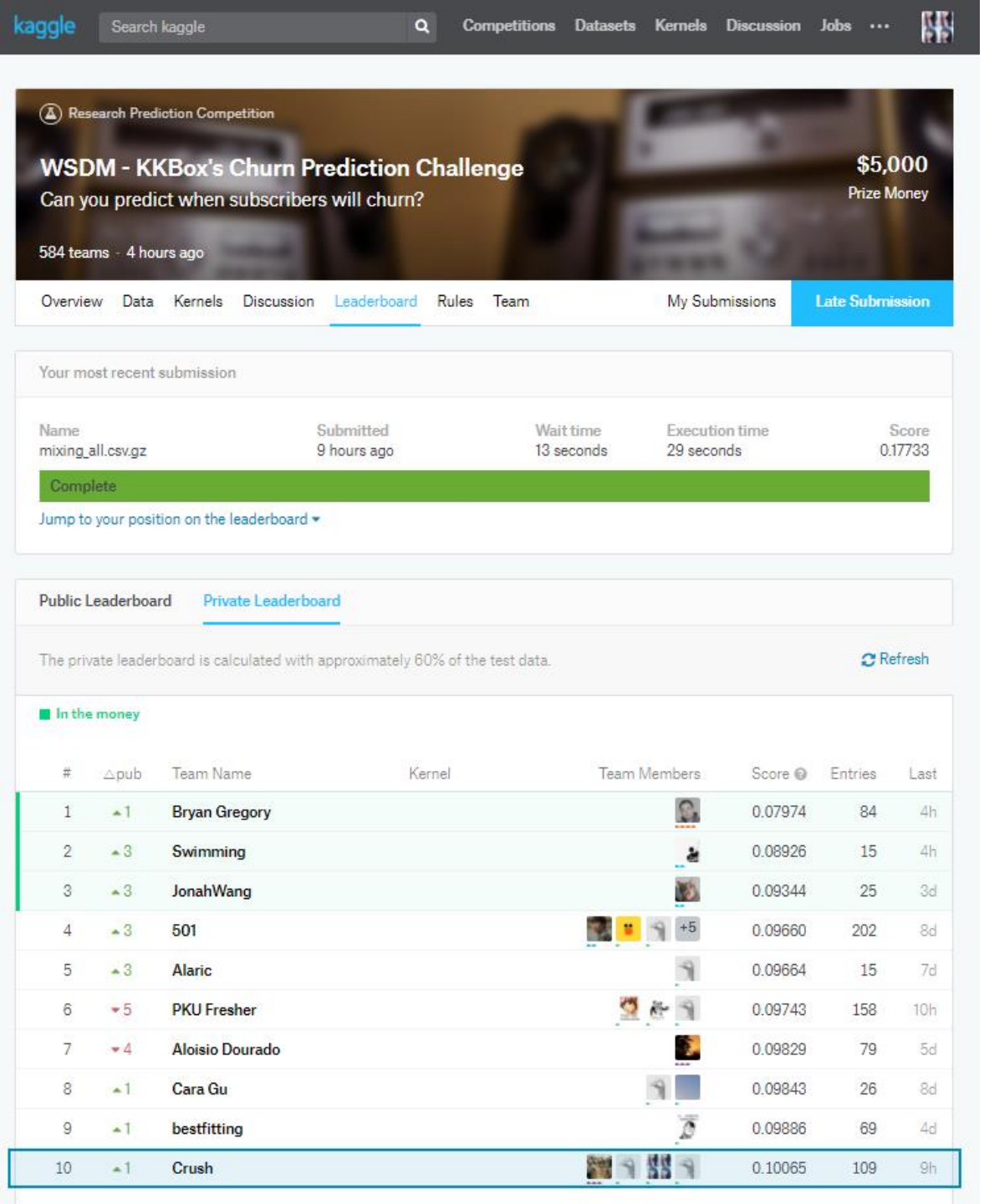
Variable Importances:

	variable	relative_importance	scaled_importance	percentage
1	amt_per_day	155194.046875	1.000000	0.355432
2	max_pay_method	88679.320312	0.571409	0.203097
3	avg_auto_renew	60690.804688	0.391064	0.138997
4	payday_sum_2017	24875.697266	0.160288	0.056971
5	log_count	22305.449219	0.143726	0.051085
6	pay_30days	9502.431641	0.061229	0.021763
7	is_continuous	9329.525391	0.060115	0.021367
8	lst_day	9122.298828	0.058780	0.020892
9	is_cancel	8959.684570	0.057732	0.020520
10	sum_weight_logs_day	8016.595703	0.051655	0.018360
11	payday_sum_2016	7302.453613	0.047054	0.016724
12	fst_exp_date	6260.383301	0.040339	0.014338
13	unique_pay_method_cnt	5730.938477	0.036928	0.013125
14	sum_paydays	3878.261475	0.024990	0.008882
15	mean_diff_trans_exp	3149.848877	0.020296	0.007214
16	median_visit_days	2239.236816	0.014429	0.005128
17	lst_exp_date	1807.467285	0.011646	0.004140
18	pay_7days	1271.361450	0.008192	0.002912
19	trans_cnt	1238.784058	0.007982	0.002837
20	mean_rate_100	1101.769653	0.007099	0.002523

Random Forest

Variable Importances:

	variable	relative_importance	scaled_importance	percentage
1	amt_per_day	2183857.250000	1.000000	0.133806
2	payday_sum_2017	1859294.625000	0.851381	0.113920
3	max_pay_method	1443778.625000	0.661114	0.088461
4	sum_paydays	1426693.125000	0.653290	0.087414
5	avg_auto_renew	1128250.125000	0.516632	0.069128
6	pay_30days	1080356.250000	0.494701	0.066194
7	trans_cnt	719900.625000	0.329646	0.044109
8	payday_sum_2016	655629.187500	0.300216	0.040171
9	log_count	478663.656250	0.219183	0.029328
10	sum_weight_logs_day	457376.312500	0.209435	0.028024
11	lst_day	367648.468750	0.168348	0.022526
12	pay_7days	314521.781250	0.144021	0.019271
13	is_continuous	305252.093750	0.139777	0.018703
14	mean_diff_trans_exp	298999.812500	0.136914	0.018320
15	fst_exp_date	274157.718750	0.125538	0.016798
16	lst_exp_date	237266.640625	0.108646	0.014537
17	mean_rate_100	236352.453125	0.108227	0.014481
18	registered_via	231842.625000	0.106162	0.014205
19	city	213360.703125	0.097699	0.013073
20	unique_pay_method_cnt	211144.453125	0.096684	0.012937



- 0.10065 -> 10등 -> 다함께 한것
- 0.10389 -> 13등 -> 발표는 나 혼자 한 걸로

- 단, 10등에서 말소 !!! ㅠㅠ

- 응??? 이유는?
- 계정 3개 사용
- 룰을 제대로 안 읽은 잘못.
- 그리고 10등까지 할 줄 몰랐음.

- 그래서 열심히 했지만 기록으로 남는게 없어요!
- 경험은 했지만~ 코드도 있지만~ 기록이 안남네요 ㅠㅠ

작업환경

<실제 작업환경>
- RAM 8G
- CPU i5
- Macbook pro
- Rstudio
- H2O packages

<대회 후 작업환경>
- RAM 64G
- CPU i7-8700K 12 cores
- OS Windows10
- Rstudio
- H2O packages
- docker

한계

Testset : 907,471
Trainset : 10,394,394
(2016년 2월 – 2017년 2월)
User_logs : 410,502,906
Members : 6,769,473

- 데이터가 너무 커서 시간이 너무 오래 걸린다.
- 캐글 대회 1등은 데이터 핸들링을 MS SQL로 했다고 한다.
- 데이터 핸들링 코드 짜는 거 제외 모두 돌리는데 최소 12시간은 걸렸음. 맥북으로는 transaction만 처리하는데 12시간 걸렸음.
- 모델을 한번 fitting 하는 시간도 한번에 40분 정도 걸려서 nfolds=30이 한계치. 맥북으로는 2시간 정도?!(3배 차이)
- 그래서 모델 튜닝을 많이 못해본 것이 가장 아쉽다.
- 모델 튜닝하지 않고 기본 ntree=200, nfolds=3을 기본으로 피팅.
- 파생변수 만드는 것에 더 집중.

최종 결과 및 느낀점

- 어떤 것이든 룰을 먼저 확인.
- 모델 튜닝 부분이 부족.
- Docker를 활용하여 rstudio작업환경 구성법을 배움.
- 파생변수를 만들면서 dplyr을 아주 잘 활용할 수 있게 되었음.
- 코드를 짤 때부터 주석을 잘 달아둬야 한다.

- 코드
https://github.com/RyuJiseung/WSDM_2018/tree/master/code

Thank you