



2018 BIGCONTEST Champion League
GameSat

INDEX

분석 목적

모델링

결론 및 의의

1

3

5

2

4

탐색 및 전처리

원인 분석

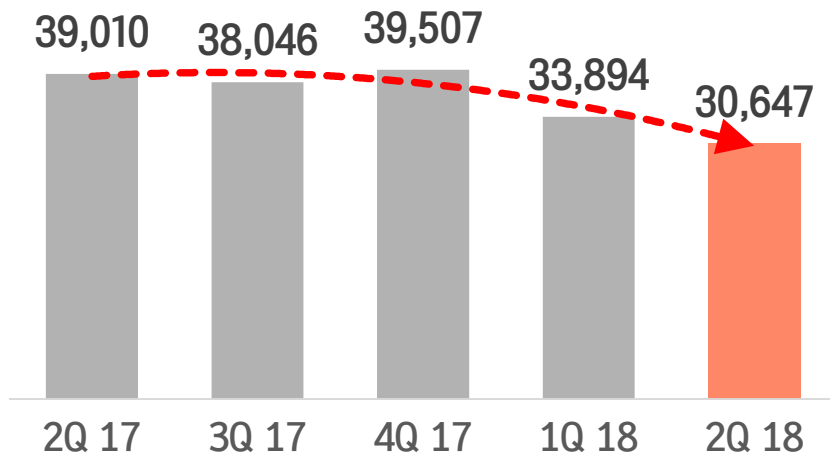


1

분석 목적

분석 배경

블레이드앤소울 매출구성



블레이드앤소울의 매출이
감소 추세

이탈 예측의 필요성

✓
낮은 이탈율은
지속적인 수익과 직결되는 중요한 요소

✓
예측을 통해
유저 이탈 방지를 위한 사전 대응 가능

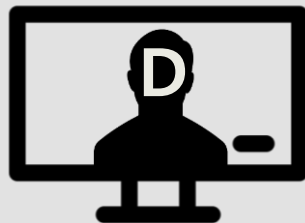
✓
유저 이탈을 막는 것은
신규 유저를 끌어들이는 것보다
비용 면에서 저렴

'이탈' 기준

학습데이터 집계 기간(8주) 이후 4주 연속 미접속 기간 발생 시,
이탈 발생 시점을 기준으로 label 부여

유저	학습데이터 집계 기간								이탈 여부 판단 기간												이탈 여부 (레이블)
	8주								12주												
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	9	10	11	12	
A																					2Month (두달내 이탈)
B																					Week (1주내 이탈)
C																					Month (한달내 이탈)
D																					Month (한달내 이탈)
E																					Retained (비이탈)
F																					Week (1주내 이탈)

예시)



유저 D: 집계 기간 이후 이탈 여부 판단 기간 중
유저 D: 2주차에 4주 연속 미접속 기간 발생

→ 2주차에 이탈한 것으로 판단 (Month)


데이터 소개

Train set 10만 개

Test set 4만 개

Activity
Payment
Guild
Trade
Party
Label

Activity
Payment
Guild
Trade
Party



Week (1주 내 이탈)	Month (한 달 내 이탈)	2month (두 달 내 이탈)	Retained (비이탈)
25000	25000	25000	25000

Week : Month : 2month : Retained = 1 : 1 : 1 : 1

2

탐색 및 전처리

들어가기 전,

데이터 전처리 방향

합리성

현실에서 의미를 가질 수
있는 변수만 사용

일관성

동일한 방식으로
결측치 처리

정보
최대화

주어진 데이터에서
최대한의 정보 활용

들어가기 전,



Wide format

Transaction data를 펼쳐서
Long format → Wide format 변환

동일한 아이디가 가진
첫 주차부터 마지막 주차까지의 정보들을
하나의 row로 합침



NA 처리

각각의 데이터 프레임을
Wide format으로 변환할 때 생기는 NA

NA 발생 주차에

접속함
(관측 가능)

“0”

접속하지 않음
(관측 불가능)

“NA”

1. Activity Data format 변환

Long format

wk	acc_id	cnt_dt
7	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	4
8	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	5
3	b8856358ff62e596fa07e3e40b8e7fd4b7729263c72b44280...	2
4	b8856358ff62e596fa07e3e40b8e7fd4b7729263c72b44280...	2
5	b8856358ff62e596fa07e3e40b8e7fd4b7729263c72b44280...	4
7	b8856358ff62e596fa07e3e40b8e7fd4b7729263c72b44280...	2
8	b8856358ff62e596fa07e3e40b8e7fd4b7729263c72b44280...	5
8	fa883ca7505082114c4024052354f1fb416f6bae26ed06788...	6
8	d094b6b1c5d0a147eaae3e37b256894def52de39c6eca33c...	3
1	38e7088d64485baba2968be8ad56f5b8abeced8ccd95f153...	6
2	38e7088d64485baba2968be8ad56f5b8abeced8ccd95f153...	7
3	38e7088d64485baba2968be8ad56f5b8abeced8ccd95f153...	6
4	38e7088d64485baba2968be8ad56f5b8abeced8ccd95f153...	4
5	38e7088d64485baba2968be8ad56f5b8abeced8ccd95f153...	6

Wide format

week7_cnt_dt	week8_cnt_dt	week1_play_time	week2_play_time
5	7	NA	NA
3	3	-0.6609552	-0.6608563
1	5	NA	NA
7	6	-0.3728307	NA
0	2	NA	-0.6597189
6	4	-0.4499481	-0.4071314
0	5	NA	NA
5	5	NA	NA
0	5	NA	NA
7	7	4.7267895	0.5740608
0	2	NA	NA
2	4	NA	NA
7	1	NA	NA
0	1	NA	NA
4	2	NA	NA
2	1	NA	-0.2086755

유저가 플레이 하지 않은 주차의 값은 “NA”로 처리
(정수값인 cnt_dt 변수는 “0”으로 처리)

1. Activity Feature engineering

↳ play_time

다양한 모델 parameter에 구애받지 않고 중요한 변수로 선택된
“play time”

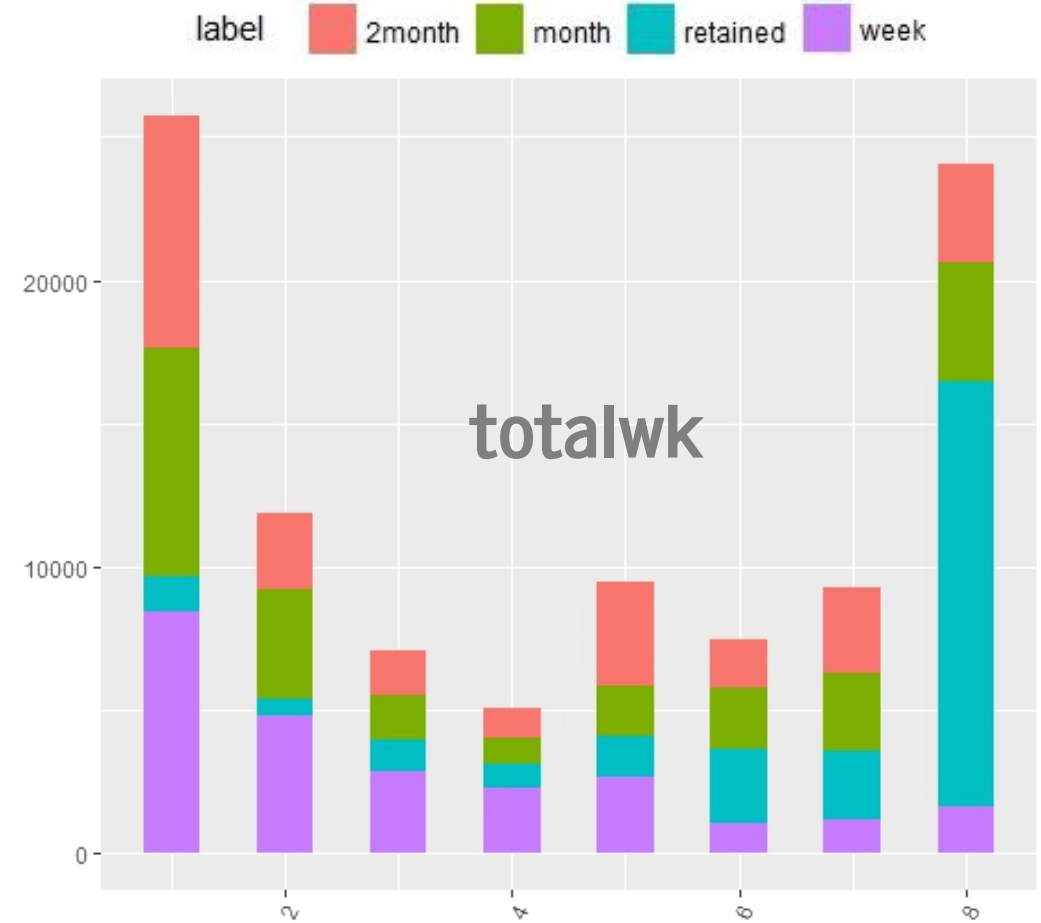
- ◆ **play_time_mean**
8주 간의 평균 play time
- ◆ **play_time_daily**
해당 주의 게임시간을 그 주의 접속 날짜수(cnt_dt)로 나눈 일평균 play time
cnt_dt로 나누기 위해 play time을 (0,100) min-max scaling
- ◆ **sd_play_time**
주차별 play time의 표준편차
- ◆ **lm**
8주 간의 play time 회귀계수(기울기) → 증가/감소 추세 산출

1. Activity Feature engineering

↳ week

유저가 얼마나 꾸준히
게임을 했는지 판단하기 위해
“week” 사용

- ◆ **totalwk**
접속한 주차의 횟수 (1~8)
- ◆ **no_cnt_dt**
첫 접속 주차 이후 미접속한 주차의 횟수
- ◆ **maxleap**
첫 접속 주차 이후 가장 길게 접속하지 않은 기간



접속한 주차의 횟수(totalwk)가 많을수록
retained의 비율이 높아짐

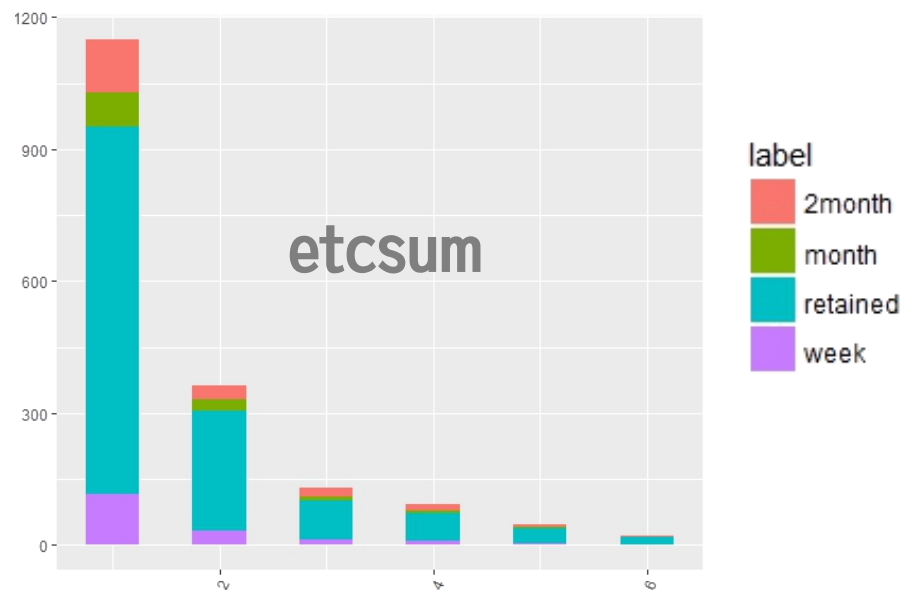
2. Trade Feature engineering

전체 5,543,038개의
거래 데이터

money (돈)	grocery (잡화)	costume (옷)	accessory (액세서리)	gem (보석)	weapon (무기)
2,972,554건	2,567,457건	1,154건	976건	591건	306건

전체 거래량의 0.000546%

“etc”라는 품목으로 통합



etc를 한 번이라도 거래한 사람 중
retained가 차지하는 비율이 높음

◆ **etcsun**
etc를 거래한 횟수를 나타내는 변수 생성

2. Trade Feature engineering

grocery(잡화)와 money(돈)을 판매자와 구매자로 분류
(아래와 같이 네 가지로 분류됨)

Source (판매자)	Target (구매자)
◆ s_money_times 돈을 준 횟수	◆ t_money_times 돈을 받은 횟수
◆ s_grocery_times 잡화를 판매한 횟수	◆ t_grocery_times 잡화를 산 횟수

2. Trade Data format 변환

Long format

trade_week	source_acc_id	target_acc_id	item_type
8	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	96995041e6295f5a5aa86d07f805ec3b38aef10dc08f992cd...	grocery
8	98b97c104e8b943069c3ddcf5f5c367b0406ad25973daa6e...	5dd5ebf6bdd5dad9d6064dc4ec47d0e9c137e674cf909fc8...	accessory
7	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	7d9700471abcd1bc304d7f9b72bb906e7602a0a7565525a...	grocery
1	9c8c6cf69f6b2c48e7c377bb8a0327565973651bc8dbdedd...	0eb20d13ef580d1ccf04d264d69e0cca19e50dc740092681...	money
5	e74804ca43e1def3a26ebde27dae6f68e321d70dfc2726d5...	abadb3ef12ca631a64e31aa8c0210d4f40c7f4d831ba5d11f...	gem
1	8ea2fa432f8e46d499470f2fc8eb540dacdd493f069f07b78...	0eb20d13ef580d1ccf04d264d69e0cca19e50dc740092681...	money

Wide format

acc_id	week1_s_grocery_times	week1_s_money_times	week1_t_grocery_times	week1_t_money_times
0000264b01392acfde44f9d8494f112a701dc5d3e5fda6ad...	NA	NA	NA	NA
0001f130e89288ff0df167b998f6eb7085687f411fcb72440d...	0	0	0	0
00028f0ad71c07f48aef465fd0c98ba6e3f0f3de3d2f7d14c9...	NA	NA	NA	NA
00036e5b6a197c196fa10fc0ad4e2853b22294dff64d2735a...	3	3	0	0
00037fe8e712041a476b8b1b827583cbc34895844057f03d...	NA	NA	NA	NA
0003b127aa1c0c34490db8817574482464aad9f99cffde4d...	6	6	0	0

3. Party Feature engineering, Data format 변환

◆ party_count

해당 주차 파티 플레이 횟수

◆ party_time

해당 주차 파티플레이 시간 (분 단위)
파티 결성 일자 기준

예) 1주차 일요일에 시작해서 2주차 월요일에 파티가 끝났다면
1주차의 파티 플레이 시간으로 기록됨

Long format

party_start_week	party_start_time	acc_id
1	09:14:58.558	11fc85879e5ac9d5c83bfa10d73c4c84c154b9f4d9e1dd5f...
3	11:05:05.176	7176c1516207692857535c30a4650b8e8e586af1fed0fdac...
3	02:18:43.172	8092e194a750aae539862ed4405f67a6dd5b492e7e57e32...
4	09:22:01.936	4ec597c569b92bd0e1bae4e2a06e13b9657fb81795e194c6...
4	06:29:21.182	a4b6aea6cb58e43911e7cb7d6c0497197db7c4ed16e1c9f8...
6	09:12:30.447	4b33f0b6969e591bb19d7ea939af5e45e08c6799ef18e78c...
6	10:58:28.822	a284744f3707f84daf525d5040191fda9a46db4c368fe6b5c...
7	16:30:21.582	0b050fb529044f342c674d6e728dff00c42ded893363feb2...
8	09:13:55.044	603ee11e78413e478b4c6a59362f06f1e918e4207bab2103...
8	01:52:02.512	37d8c7e0212f1af017330a1ac452ed1eee5ee8e7fd0359cf1...

Wide format

acc_id	week1_party_count	week1_party_time
0000264b01392acfd44f9d8494f112a701dc5d3e5fda6ad...	NA	NA
0001f130e89288ff0df167b998f6eb7085687f411fcb72440d...	0	NA
00028f0ad71c07f48aef465fd0c98ba6e3f0f3de3d2f7d14c9...	NA	NA
00036e5b6a197c196fa10fc0ad4e2853b22294dff64d2735a...	14	159.5333
00037fe8e712041a476b8b1b827583cbc34895844057f03d...	NA	NA
0003b127aa1c0c34490db8817574482464aad9f99cffe4d...	7	174.9667
0004733c4175d61e67d1ec9d3602f6c3341180800bc8bf2b...	NA	NA
00047f6584e6bcfed540e1bc53651c0c27f02c5bc5309197a...	NA	NA
0004a2a2f32479b2e0ff35b2a3b9d77f9949a1d2f3f16c38e...	NA	NA
0006502148dc2533ef4ac4b9939ee19f18483a7c3c7256fa8...	143	321.6667

4. Trade & Party Feature engineering

✓ 요일 & 시간 개념 처리

오직 trade data와 party data에만 존재

Day	수요일(1) ~ 화요일(7)
Hour	00:00:00

party_start_week	party_start_day	party_start_time	party_end_week	party_end_day	party_end_time
1	1	09:14:58	1	1	09:41:30
3	3	11:05:05	3	3	13:07:42
3	6	02:18:43	3	6	02:28:58
4	1	09:22:01	4	1	09:47:40
4	5	06:29:21	4	5	06:50:55
6	1	09:12:30	6	1	09:31:51
6	5	10:58:28	6	5	11:01:05
7	5	16:30:21	7	5	16:44:54

▲ Party

trade_week	trade_day	trade_time
8	2	23:32:06
7	6	05:06:31
1	2	20:05:18
1	2	22:18:09
1	3	00:31:10
1	3	23:58:36
1	4	03:05:24
1	4	19:26:44

▲ Trade

→ 유저의 플레이 성향 확인 가능
평일과 주말에 각각 얼마나 접속하는가

→ 시간대별 유저 접속여부(활동여부) 확인
일과시간/저녁시간/새벽에 각각 얼마나 접속하는가

4. Trade & Party Feature engineering

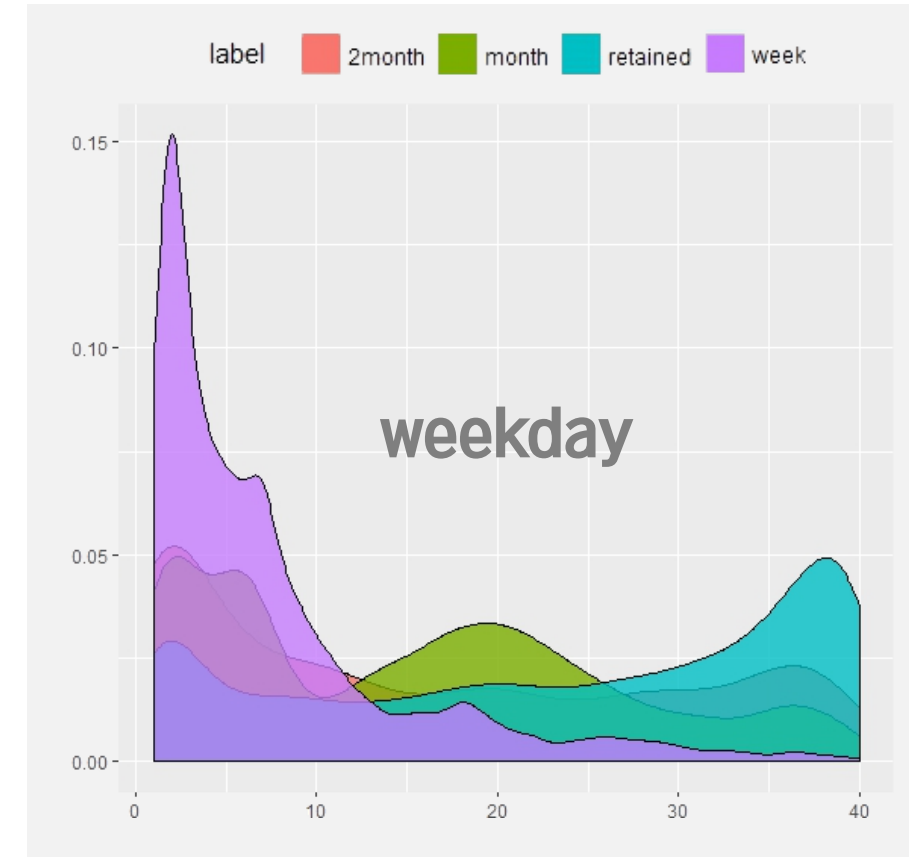
☑ 요일 & 시간 개념 처리

1) trade data 횟수 party data 횟수

- | | | |
|----------------|-----------------|----------------|
| ◆ Tr_daytime | ◆ Pty_daytime | : 08시~16시(일과중) |
| ◆ Tr_nighttime | ◆ Pty_nighttime | : 16시~24시(저녁) |
| ◆ Tr_dawntime | ◆ Pty_dawntime | : 24시~08시(새벽) |

2) trade data와 party data를 함께 고려

- ◆ Weekday 평일 활동 날짜 수
- ◆ Weekend 주말 활동 날짜 수
- ◆ Playday 평일과 주말을 합친 활동 날짜 수
- ◆ Hour_day 해당 시간에 활동한 날짜의 수
e.g.) Hour_04_day: 04시에 활동한 날짜의 수 (0~56)
- ◆ Week8_hour 8주차에 해당 시간 접속 여부
e.g.) Week8_hour_04: 8주차 04시에 활동한 기록이 있는가



평일 활동 날짜수가 적을수록
week의 비율이 높음

5. Guild Feature engineering

◆ guild_num

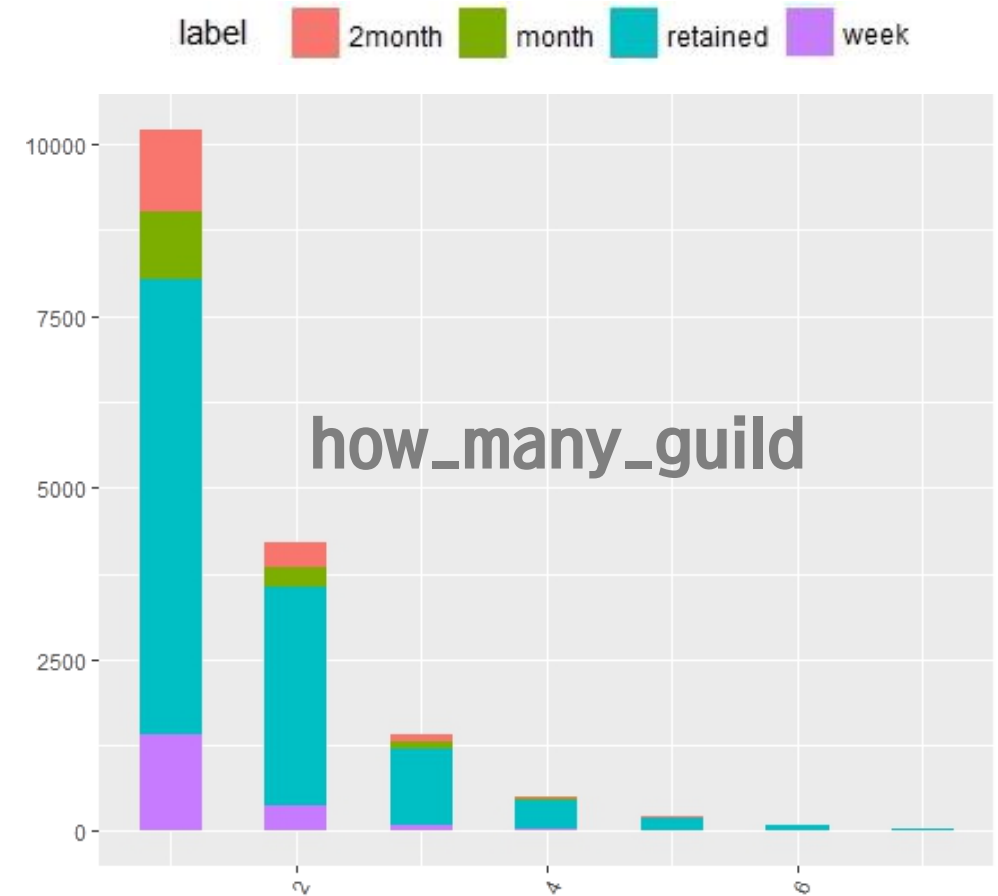
사용자가 보유한 캐릭터 중 길드에 가입한 캐릭터의 수

→ 얼마나 많은 캐릭터를 실질적으로 운용하고 있는가

◆ how_many_guild

사용자가 가입한 길드 중 가장 최대규모 길드의 길드원 수

→ 얼마나 영향력 있는 길드에 속해 있는가



길드를 하나라도 가입한 사람 중
retained가 차지하는 비율이 높음

6.Payment Data format 변환

Long format

payment_week	acc_id	payment_amount
1	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
2	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
3	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
4	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
5	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
6	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
7	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
8	3dc6f2875dc6e6f35b9e2bdb25b391a8003386ff23becd10...	-0.1498985
1	b8856358ff62e596fa07e3e40b8e7fd4b7729263c72b44280...	-0.1498985

Wide format


acc_id	week1_payment_amount	week2_payment_amount	week3_payment_amount
0000264b01392acfd44f9d8494f112a701dc5d3e5fda6ad...	-0.14989847	-0.1498985	-0.1498985
0001f130e89288ff0df167b998f6eb7085687f411fcb72440d...	-0.14989847	-0.1498985	-0.1498985
00028f0ad71c07f48aef465fd0c98ba6e3f0f3de3d2f7d14c9...	-0.14989847	-0.1498985	-0.1498985
00036e5b6a197c196fa10fc0ad4e2853b22294dff64d2735a...	-0.14989847	-0.1498985	-0.1498985
00037fe8e712041a476b8b1b827583cbc34895844057f03d...	-0.14989847	-0.1498985	-0.1498985
0003b127aa1c0c34490db8817574482464aad9f99cffde4d...	-0.14989847	-0.1498985	-0.1498985
0004733c4175d61e67d1ec9d3602f6c3341180800bc8bf2b...	-0.14989847	-0.1498985	-0.1498985
00047f6584e6bcfed540e1bc53651c0c27f02c5bc5309197a...	-0.14989847	-0.1498985	-0.1498985

3

모델링

알고리즘 비교

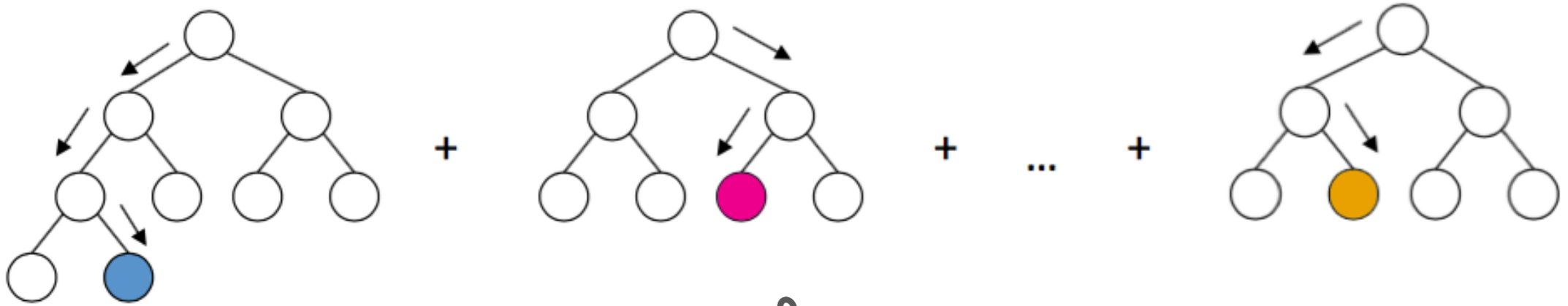
	Regression	Neural Network	Tree-Based
성능	X	O	O
설명력	O	X	△
속도	△	X	O
이상치 영향	큼	큼	적음
결측치 자동처리			
정교함 (parameter)			



Random Forest	Extra Trees	XGBoost
O	△	O
△	△	△
△	O	O
적음	적음	적음
X	X	O
△	△	O

➔ XGBoost를 주 모델로 선정

XGBoost = Extreme Gradient Boosting



- ▶ **정규화 사용**(과적합 방지)으로 높은 정확도
- ▶ **병렬처리**로 빠른 속도
- ▶ Parameter **세부 튜닝** 가능
- ▶ **Cross Validation** 자동수행 함수 내재

Parameter tuning

Max_depth

트리의 최대 깊이

Min_child_weight

노드 분할 시 필요한 최소 Instance weight

Subsample

트리에서 obs 샘플링 비율

Colsample_bytree

트리에서 Feature 샘플링 비율



**Bayesian
Optimization**

&

(나머지 Parameter 고정 후)

Eta

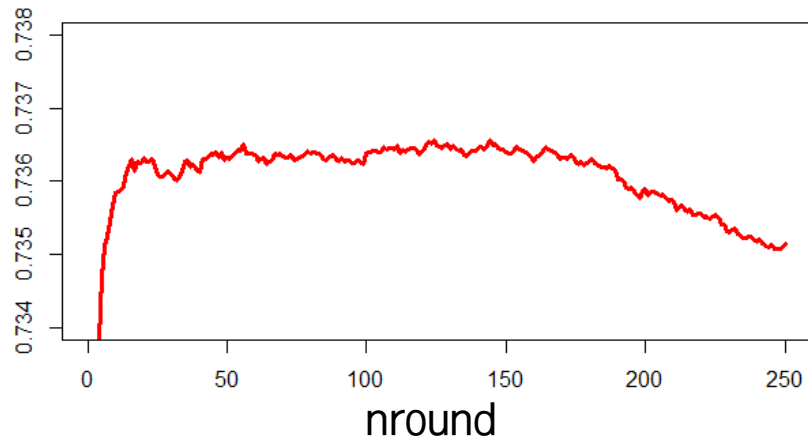
학습 속도



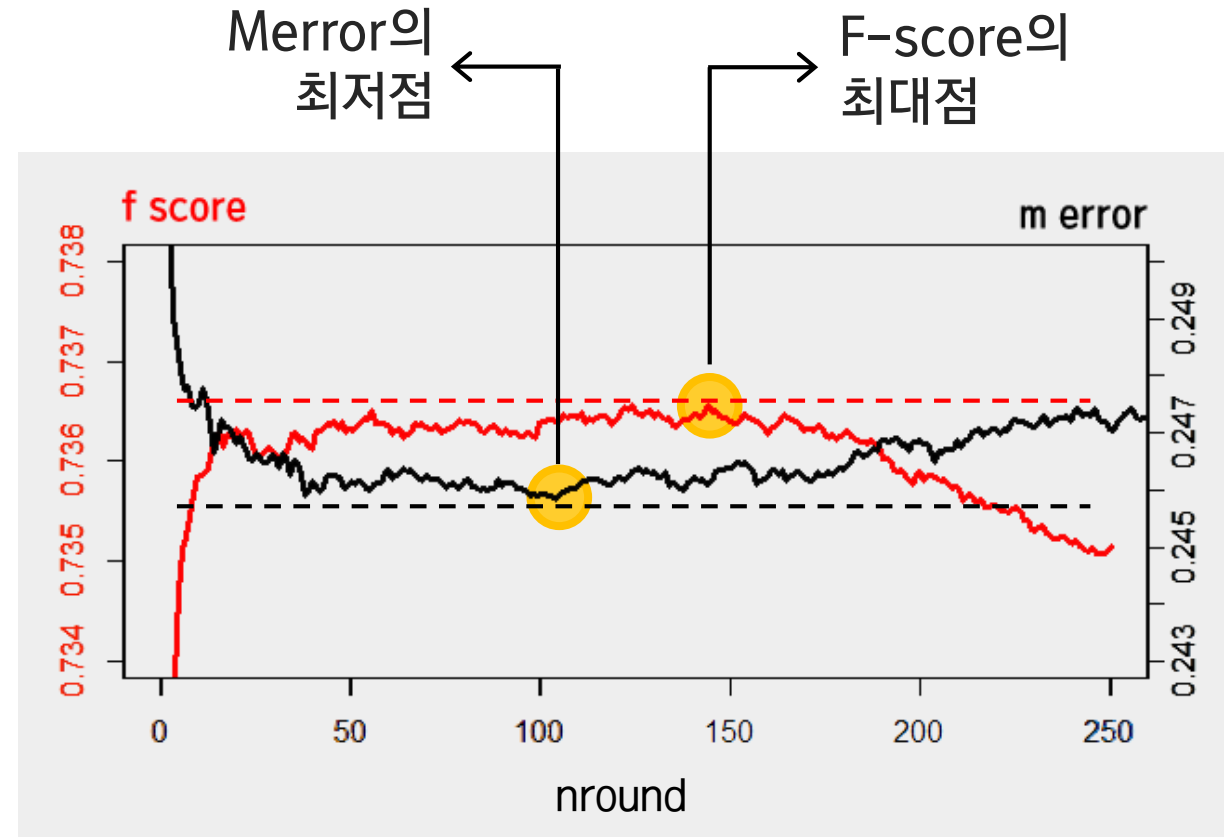
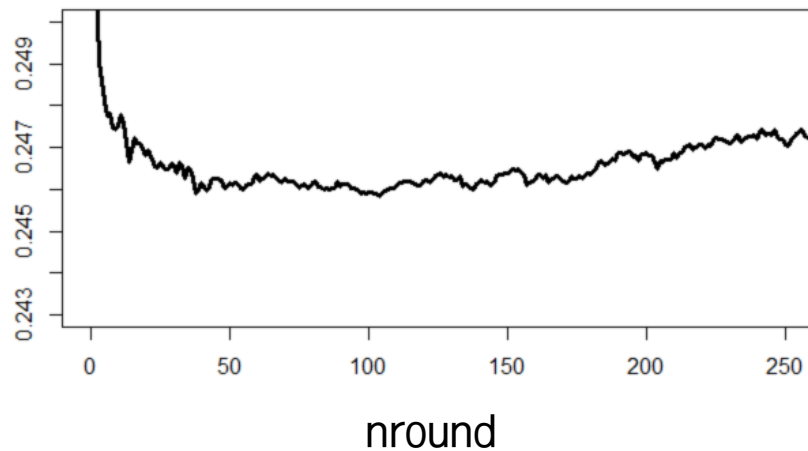
**Grid
Search**

F-score vs merror

F-score



Merror



Merror의 최저점과 F-score의 최대점이 다르기 때문에
Merror는 평가지표로 **부적절**

손실함수 vs 평가지표

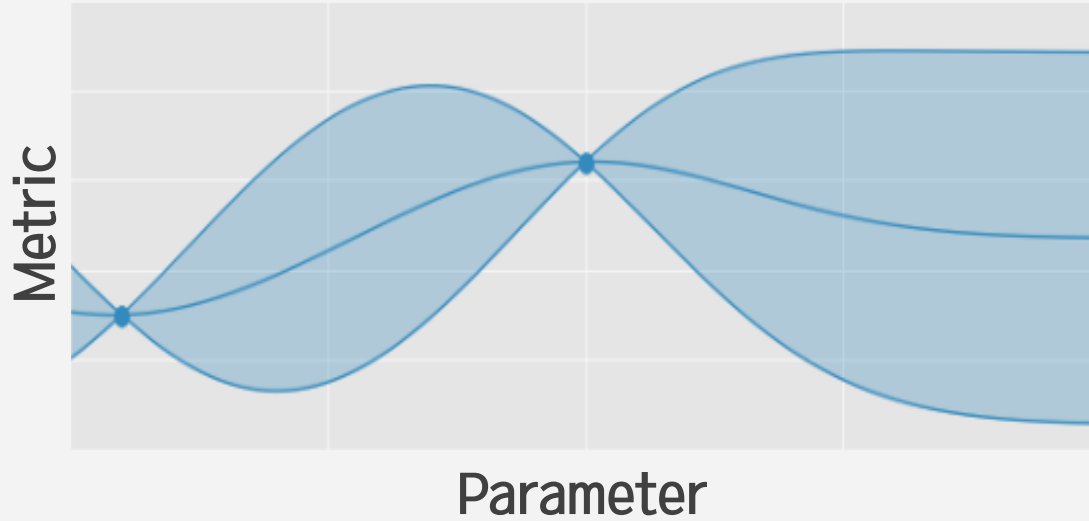
XGBoost의 손실함수	XGBoost의 평가지표
<ul style="list-style-type: none">▪ Gradient Descent를 이용해 손실함수 최적화▪ 다항 분류 시 기본 손실함수는 Cross-Entropy (Softmax)▪ Customized 손실함수 사용 가능▪ 하지만, F-Score는 미분이 불가능해 손실함수로 사용 불가능	<ul style="list-style-type: none">▪ Cross-Validation시 학습 Round 설정에 사용됨▪ Customized 평가지표 사용 가능▪ 최적의 학습 Round를 구해 F-Score를 간접적으로 극대화 가능

평가지표 예시 ►

[10]	train-Customf1:0.767548+0.001617	test-Customf1:0.736612+0.001635
[11]	train-Customf1:0.767602+0.001532	test-Customf1:0.736768+0.001677
[12]	train-Customf1:0.768205+0.001577	test-Customf1:0.736755+0.001640
[13]	train-Customf1:0.768854+0.001174	test-Customf1:0.736399+0.001736
[14]	train-Customf1:0.768916+0.001074	test-Customf1:0.736148+0.001922
[15]	train-Customf1:0.769361+0.001430	test-Customf1:0.736514+0.001453
[16]	train-Customf1:0.769375+0.001452	test-Customf1:0.736317+0.001953
[17]	train-Customf1:0.769696+0.001188	test-Customf1:0.736172+0.001785
[18]	train-Customf1:0.769713+0.001072	test-Customf1:0.736268+0.001789
[19]	train-Customf1:0.769687+0.000971	test-Customf1:0.736138+0.001854
[20]	train-Customf1:0.769970+0.001061	test-Customf1:0.736196+0.002214
[21]	train-Customf1:0.770126+0.000992	test-Customf1:0.736229+0.001928
[22]	train-Customf1:0.770320+0.000931	test-Customf1:0.736352+0.002063
[23]	train-Customf1:0.770443+0.000858	test-Customf1:0.736411+0.002249

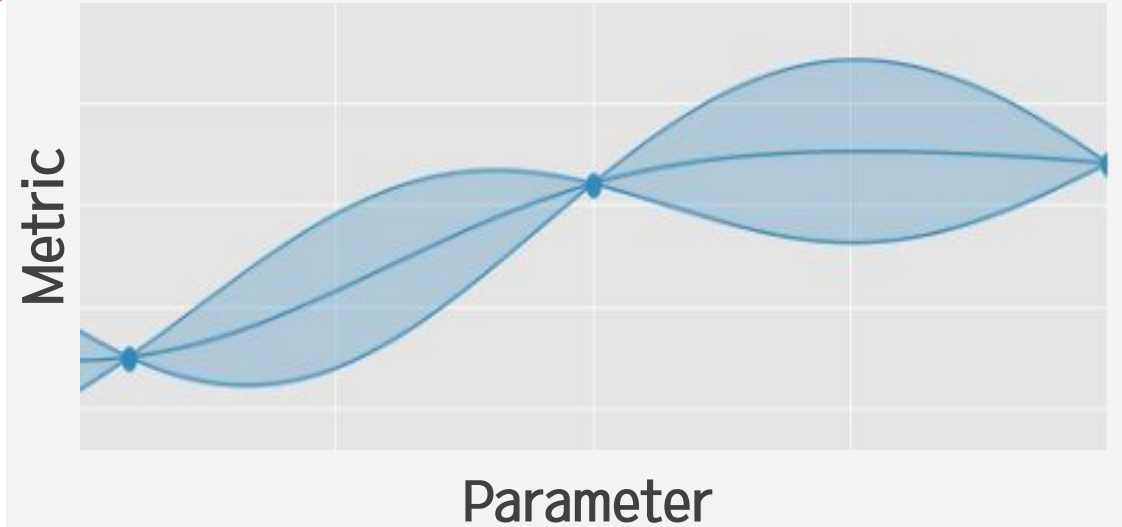
Bayesian Optimization

First Two Iterations



최초에 일정 수의 파라미터를
무작위로 선택한 후
Gaussian Process에 따라 탐색

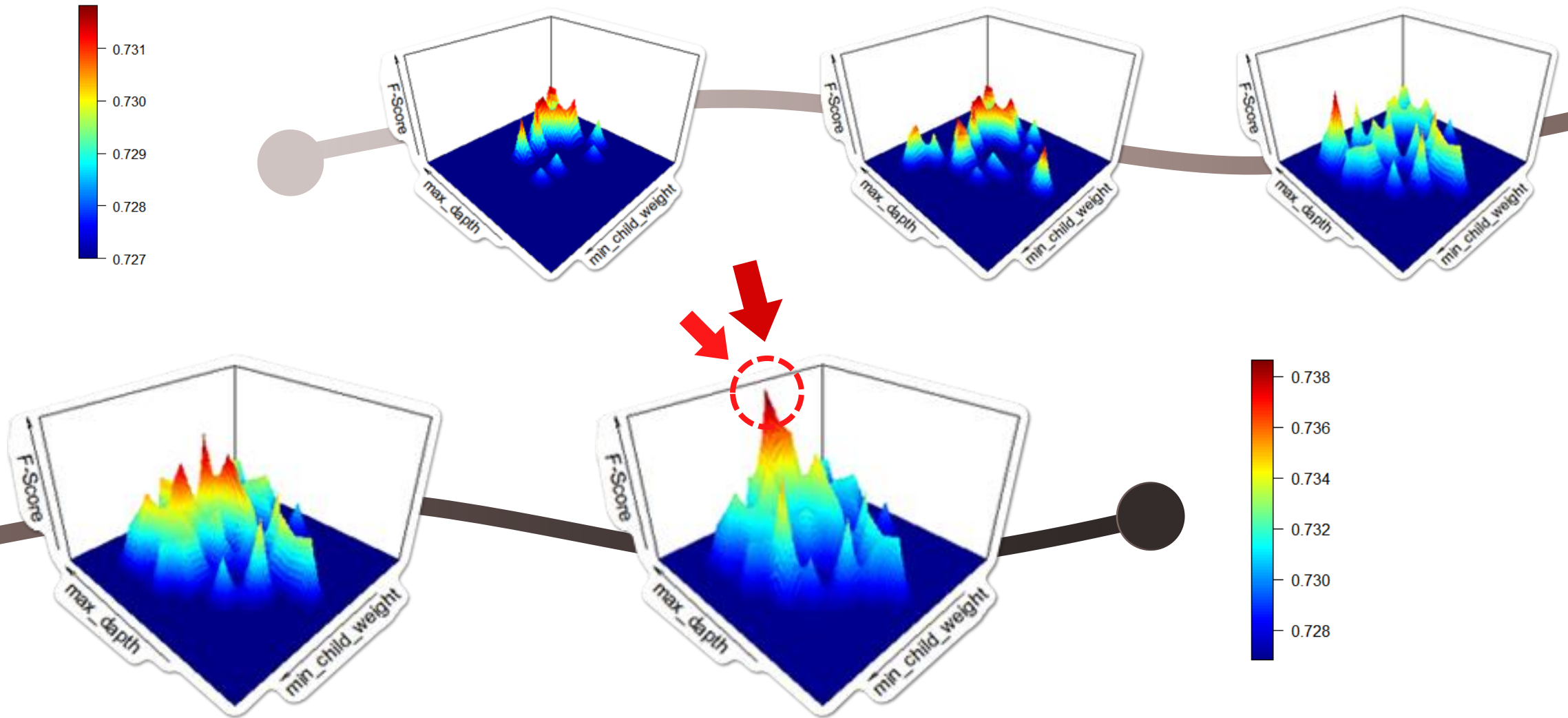
Third Iteration



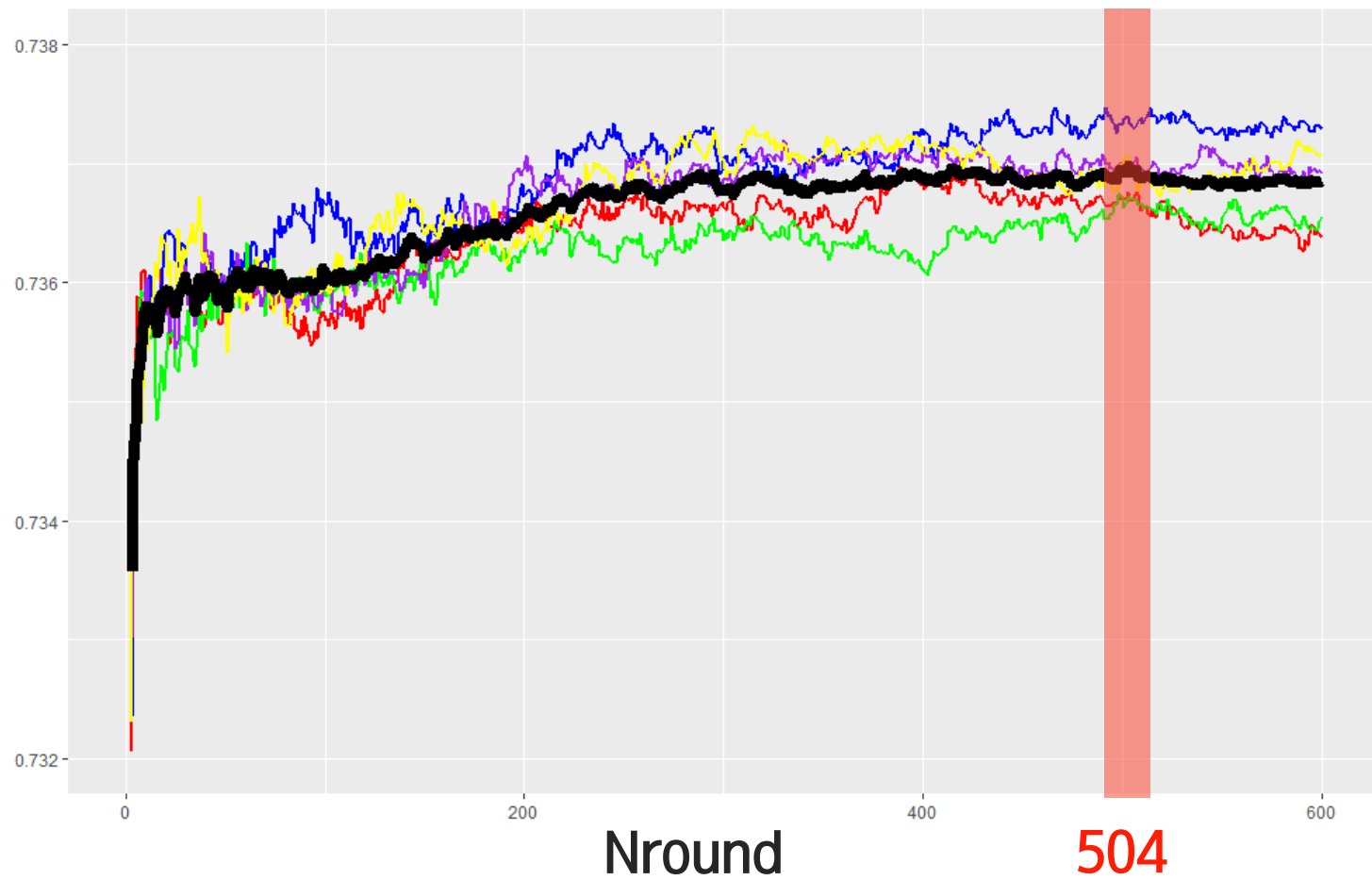
다음 파라미터 탐색지점을
Uncertainty가 높은 부분으로 찾음
→ 동일원리로 다음 지점 탐색

Bayesian Optimization

▼ 최적의 parameter를 찾아가는 과정을 시각화한 plot
(실제로는 5차원의 공간에서 진행)



학습 Round 최적화

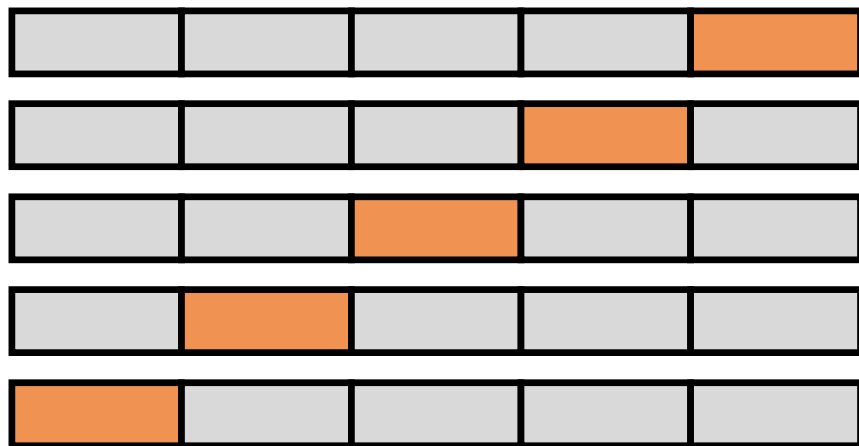


— : average

5 fold CV를 여러 번 실행한 후,
평균값이 높아지고
분산이 적어지는 부분을
Nround로 설정

Stacking

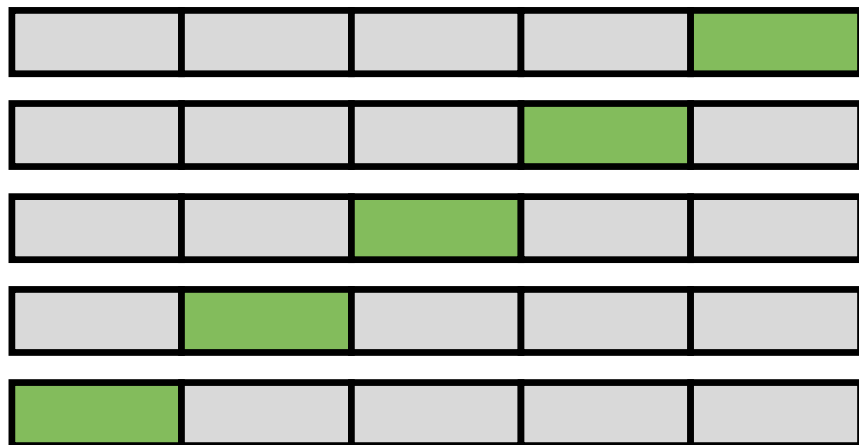
Model 1



<Level 1>

⋮

Model N



모델 앙상블 (Stacking)

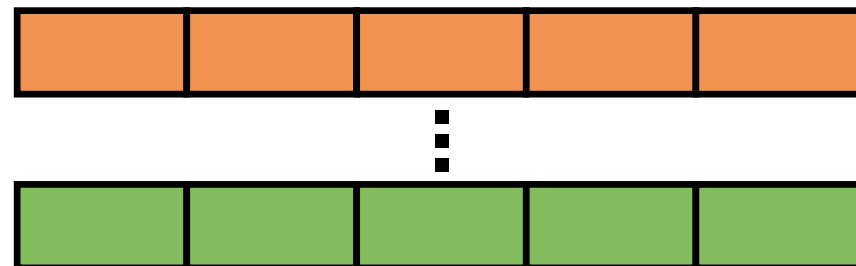
Stacking을 통해 F-Score 극대화 가능

최종 성능 향상을 위해서는
앙상블 하는 모델들의 **다양성**이 중요

→ 세 가지 트리 기반 모델,
한 개의 중요 변수 추가

<Level 2>

Level 2 Model





추가 모델 1: Weighted XGBoost



기존 XGboost 모델에서 Cross Validation을 시행할 때
'Month'를 오분류하는 현상이 지속적으로 발생

→ “Weighted XGBoost”

‘Weight’ 파라미터를 이용하여
'Month' label인 유저에게 **가중치**를 부여
(W : 1.05 ~ 1.2 Search)

Balanced Data임을 고려하여
Month **weight = 1.1**로 설정 후 새로운 모델 생성
(타 label = 1 기준)

Confusion Matrix and Statistics

Reference				
Prediction	2month	month	retained	week
2month	3560	1519	488	43
month	711	2733	157	178
retained	569	331	4163	196
week	160	417	192	4583

2month	month	retained	week
5610	3779	5259	5352

▲ 가중치 주기 전의 matrix



추가 모델 2&3: Tree Ensemble 기반 모형

	Random Forest	Extra Trees
Choosing Variables	Bootstrap Sampling	No Bootstrap (sample from entire train set)
Cut Points	Finding Optimal Cutpoints (오분류 최소화 Cutpoint)	Random Cutpoints (분산 감소효과기대)
Parameters	ntree(=500), <u>* mtry</u>	ntree(=500), <u>* mtry</u> , <u>* numRandomcuts</u>

* Optimize using Grid Search



Level2 변수: Week8_s_grocery_times



Month와 2Month가
기본적으로 구분이 어려움

Month/2Month 5만개로
이항 분류 모델을 생성



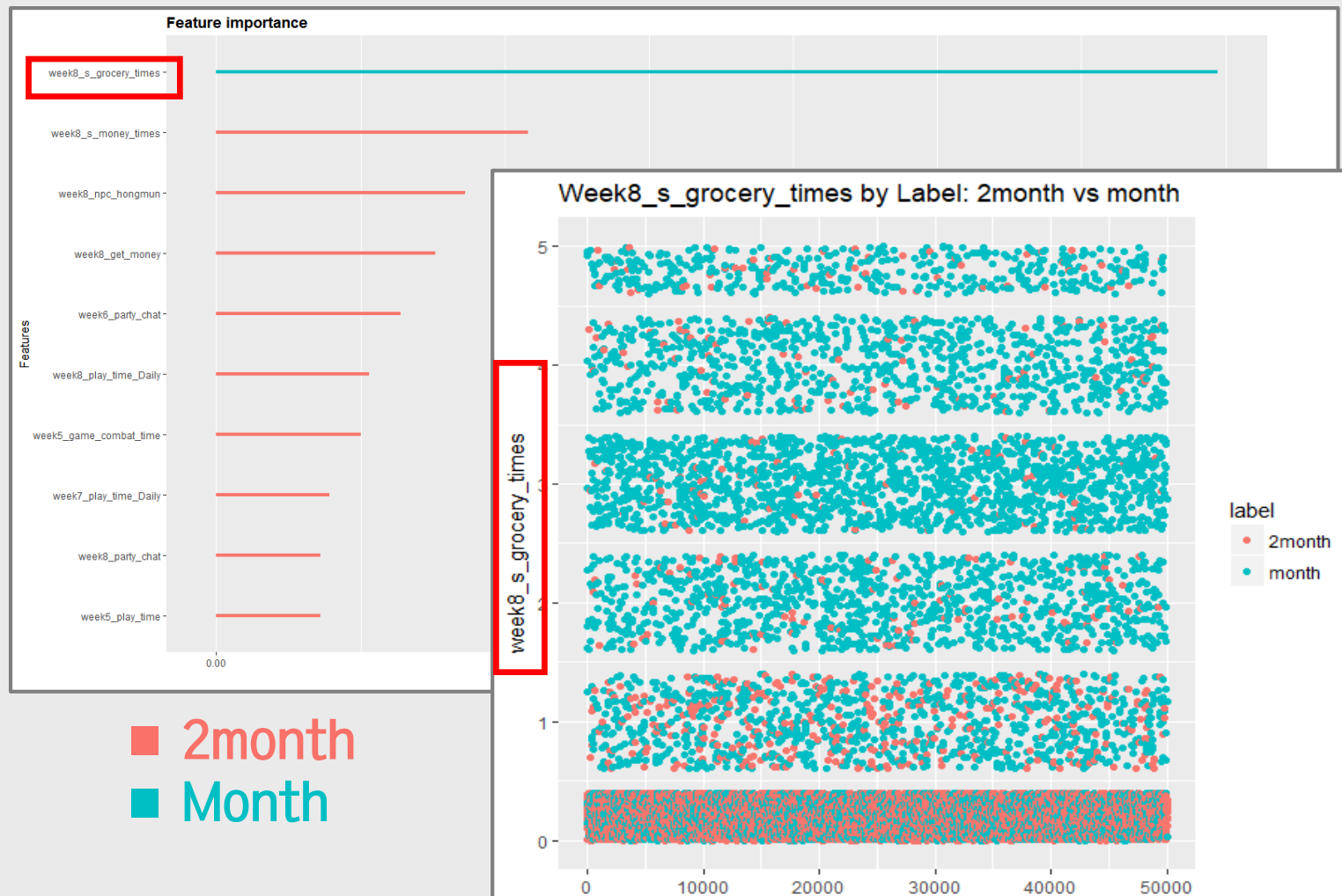
다항분류에서 중요도가 낮았던
'week8_s_grocery_times'가
가장 중요한 설명변수로 등장



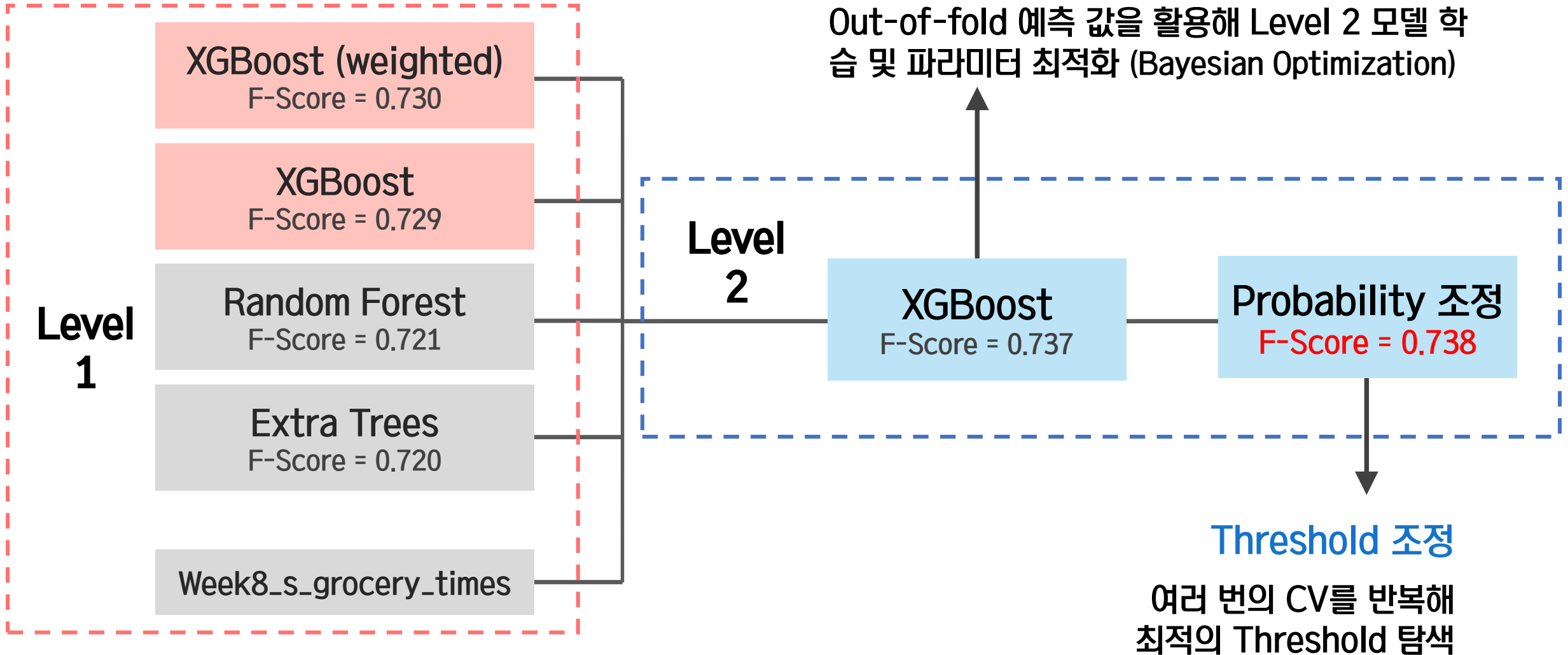
시각화를 통해 month/2month 구분이
이루어지는 변수임을 확인



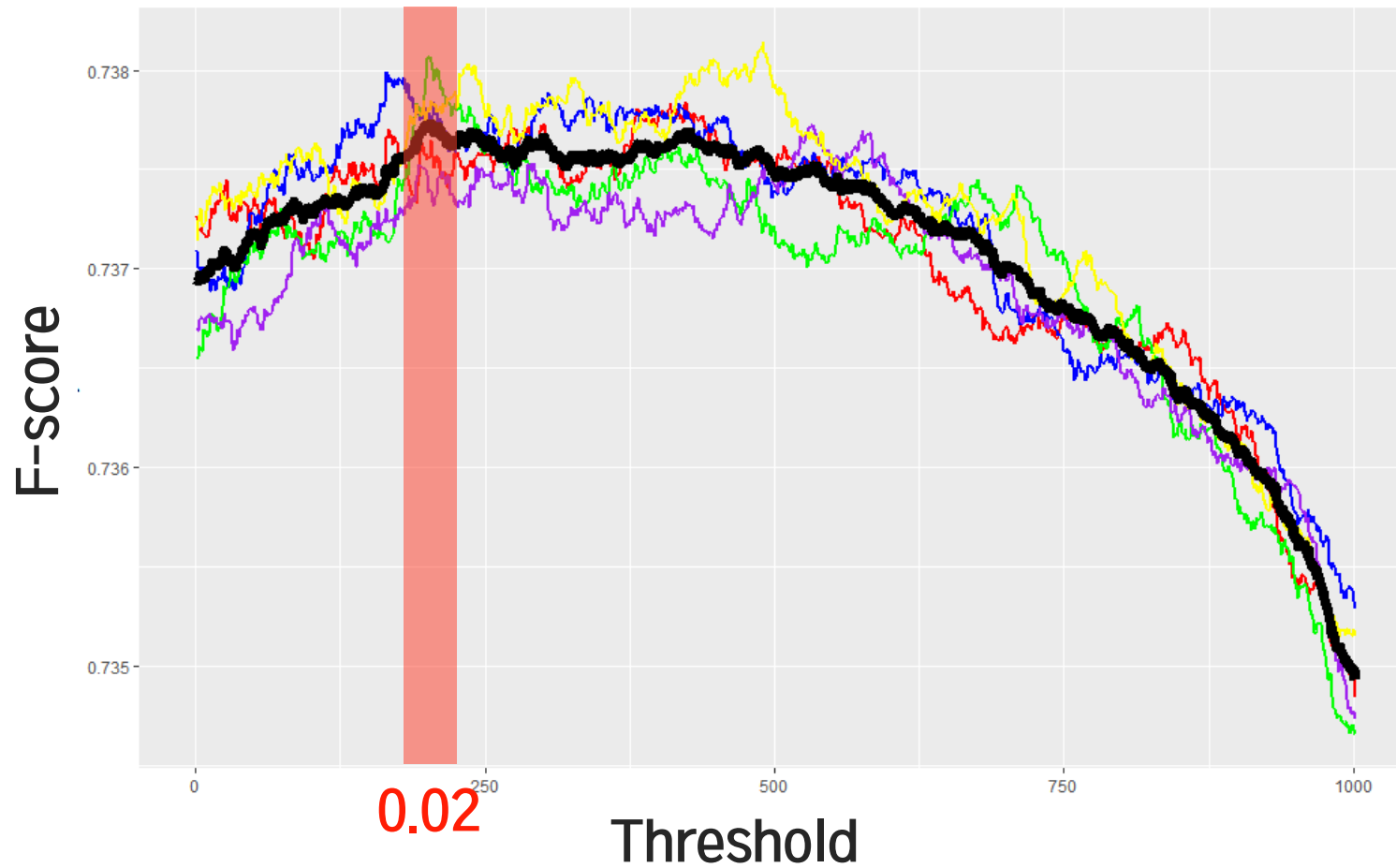
Stacking 과정에서
Level2 모델의 변수로 활용



Stacking



'Month' Threshold 조정



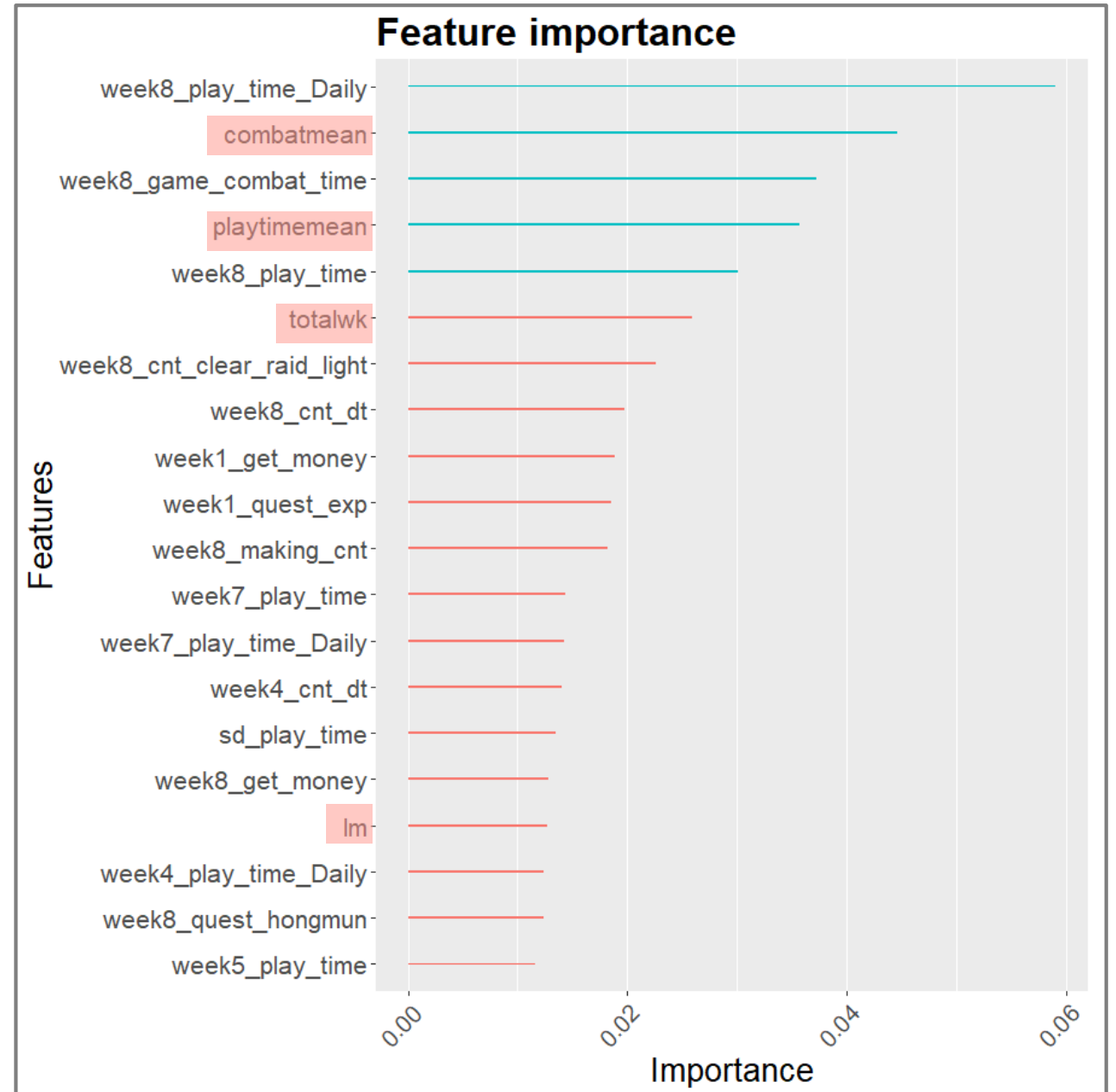
5 fold CV를 여러 번 실행한 후,
'Month'의 확률값에
가중치(+0.02) 부여

4

원인 분석

Feature Importance Plot

각 변수가 트리에서
‘불순도를 얼마나 낮추느냐’에 따라
변수별 중요도 산출
(Information Gain기준)



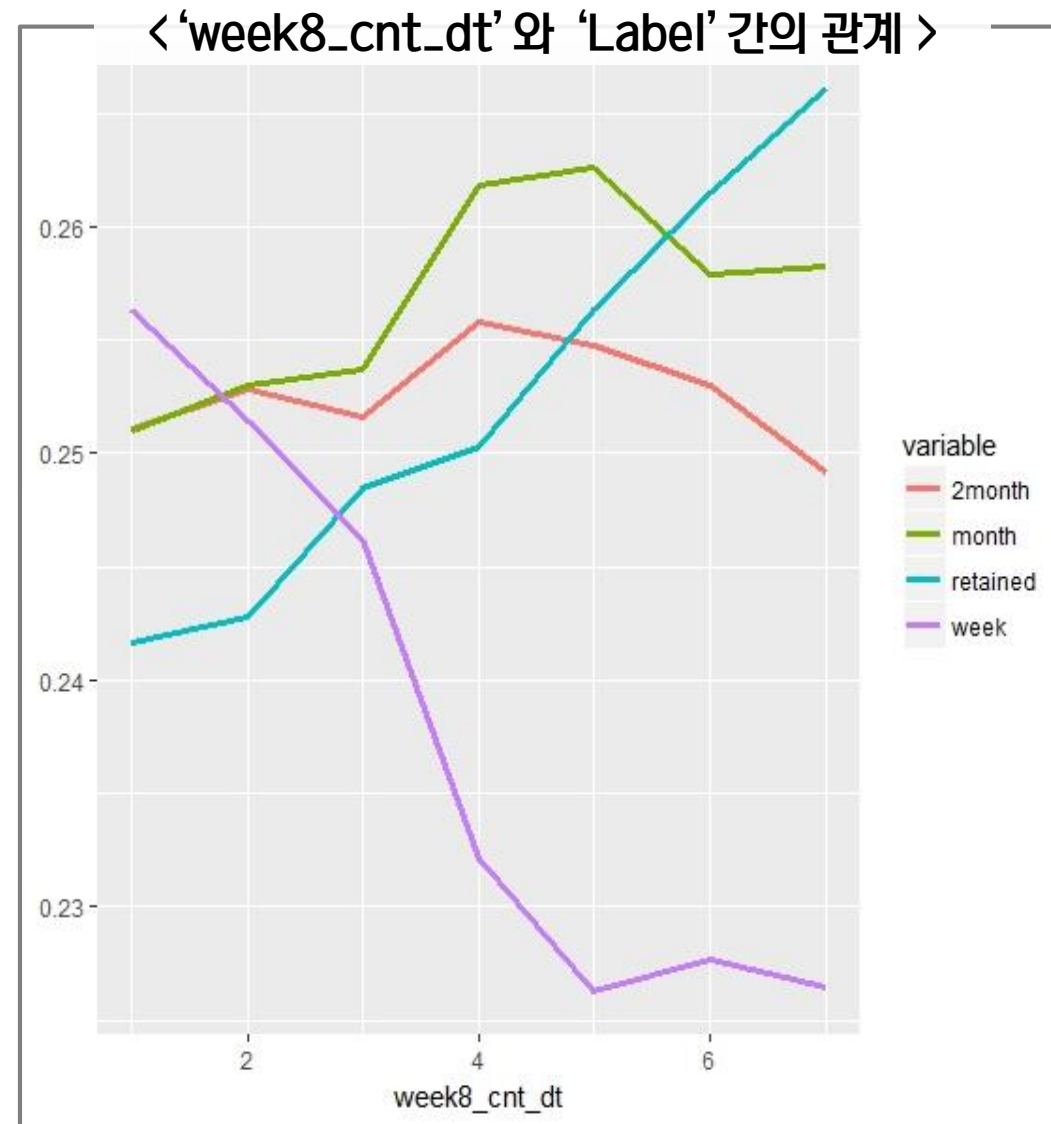
Partial Dependence Plot

다른 모든 변수가 중간값으로 고정되어 있을 때,

특정 변수 변화에 따른
Y(Label)에 속할 확률값을 보여주는 Plot

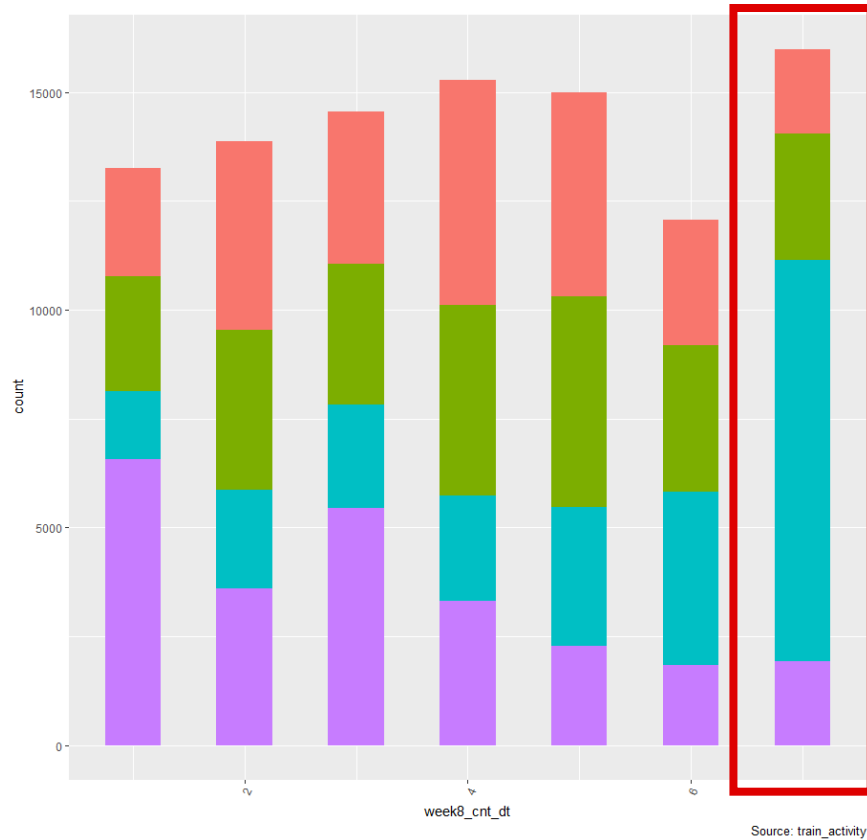


블랙 박스 모델인
XGBoost 모델의 변수 해석 가능

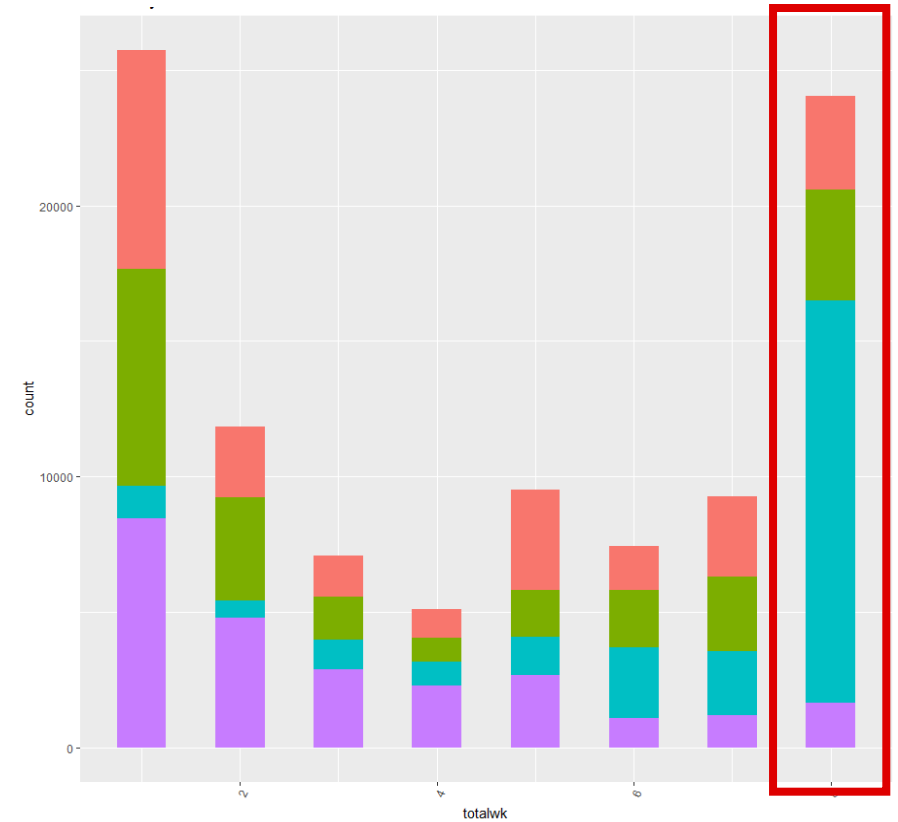


8주차 접속 일수가 많을수록
잔류(retained)일 경향이 높음

8주차 접속 일수
(week8_cnt_dt)



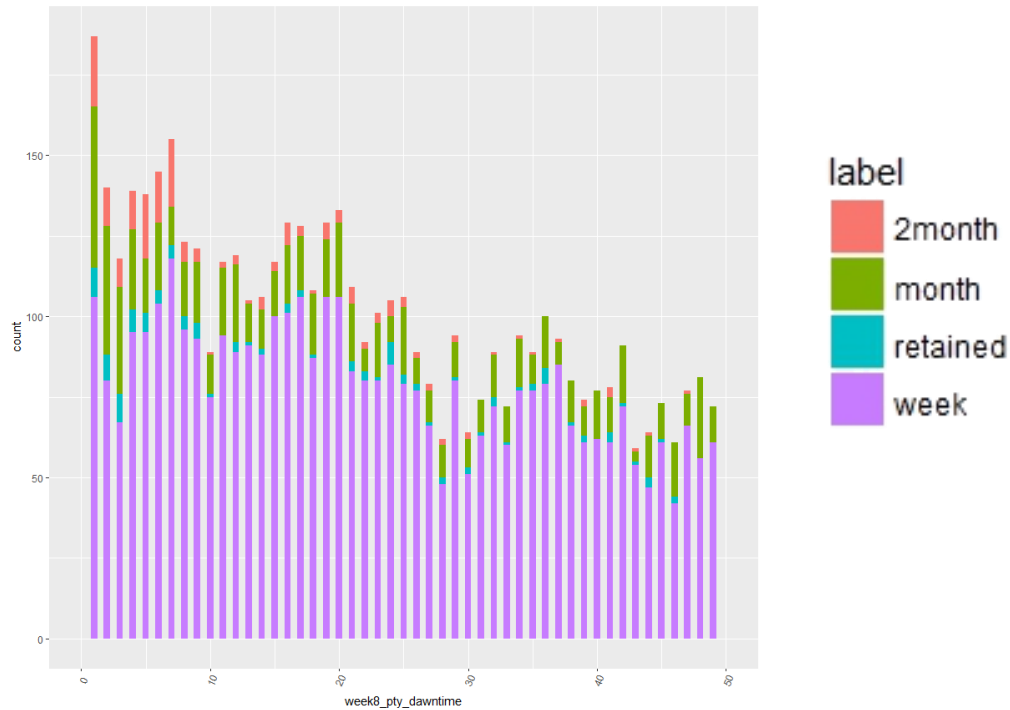
접속한 주차의 횟수
(totalwk)



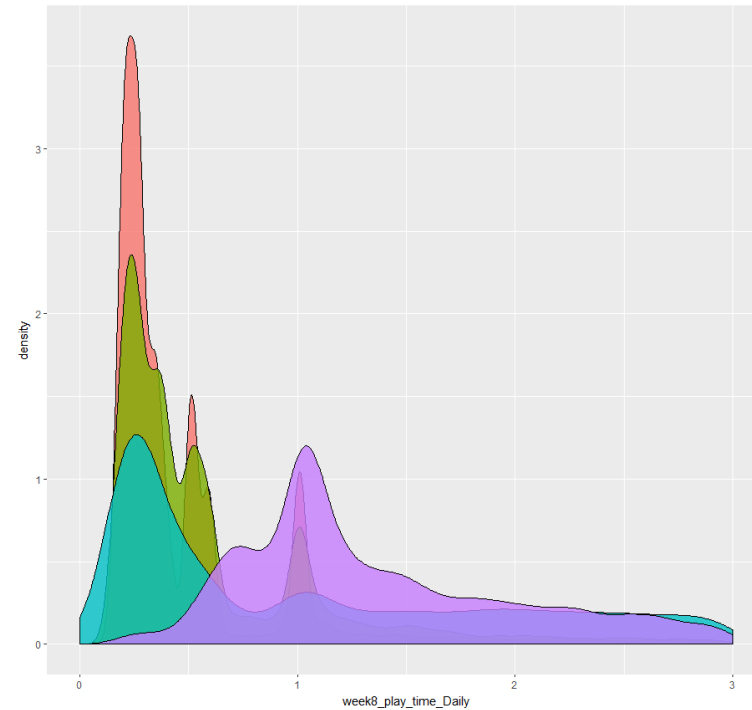
꾸준하게 플레이하는 유저들의 Label은 'Retained' 가 다수

8주차에만 플레이한 신규/복귀 유저들만 추출

8주차 새벽시간 파티 횟수
(week8_pty_dawntime)



8주차 일별 플레이 타임
(week8_play_time_Daily)



8주차 새벽시간 파티 횟수, 일별 플레이 타임 이 많은 대다수가 'Week' label 유저
➔ 신규/복귀 유저들이 단기간 게임을 오래 한 후 쉽게 질려 이탈하는 것으로 추정
이탈 시점 이전 별도의 관리 필요

5

의의 및 한계

의의 및 한계



- 다양한 변수 추출
- Custom F-score 활용
- 효율적인 Parameter 최적화
- 반복적인 CV와 시각화를 통한 과적합 방지
- Month/2month 분류를 위한 다양한 시도



- Guild & Party에 담긴 Social interaction 정보를 활용하지 못 함
- 변수 간 높은 상관관계로 인해 Partial Dependence Plot을 다양한 변수에 적용시키지 못 함

THANK YOU