게임 유저 이탈 예측

블린이

고동영, 김현우, 김혜주, 손진원, 조현호

Big Contest 2018

index

Chapter 1.

주제 및 데이터의 이해

1. 기본적인 게임 이해

기존에 범람하던 서양식 RPG에서 동양적 판타지 요소를 추가한 퓨전 판타지로,

PVE를 기반으로 PVP, RVR이 진행되는 게임이다. 화려한 그래픽으로 초반부터 많은 기대를 모았다.

2. 실제 플레이를 통해 깨달은 것들

진입장벽의 존재: 인던, 레이드 등 대부분의 컨텐츠들을 즐기기 위해서 요구되는 스펙이 높아 라이트 유저들은 컨텐츠에 요구되는 스펙을 맞추기 위해 상당한 시간을 파밍에 투자하여야 했고, 이에 따라 어느정도 진입장벽이 존재함을 느낄 수 있었다.

사회성의 중요성: 기존 유저층 내에서는 다양한 길드와 세력, 파티시스템 등 유저 간의 활발한 상호작용이 일어났다.

3. 질적조사

대부분의 컨텐츠는 던전과 PVP에 집중되어 있고, 특히 상급 이상의 던전은 각각의 공략법을 알지 못하면 깨기 어려운 난이도들이 많아서 라이트 유저들의 진입장벽이 존재할 것이라고 예상되었다.

또한, 던전이나 생활형 컨텐츠 등의 업데이트가 비교적 늦어 컨텐츠 부족으로 이탈하는 유저들이 있다는 것을 알 수 있었으며, 아이템/장비 획득을 위한 과금 문제, 직업간 밸런스 문제가 대부분의 이탈문제로 언급되었다.

1. Multi Class

week **month** **2month** **retained**

한 명의 유저가 4개 중 하나의 범주를 갖는다.

Binary 분류보다 예측이 더 어렵다.

2. Time dependent data

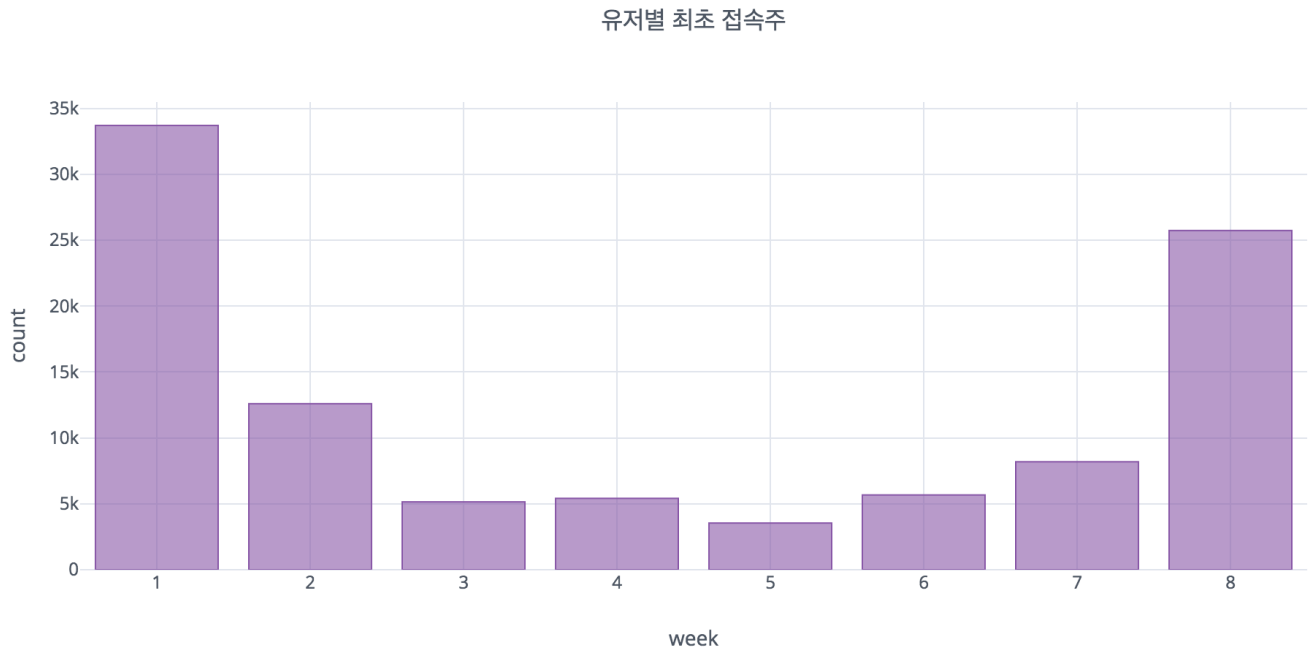
wk	1	2	3	4	5	6	7	8
play_time	50	40	30	50	60	70	80	100

한 명의 유저가 최대 8개의 관찰치를 갖는다.

그 관찰치가 시간에 종속되어있다.

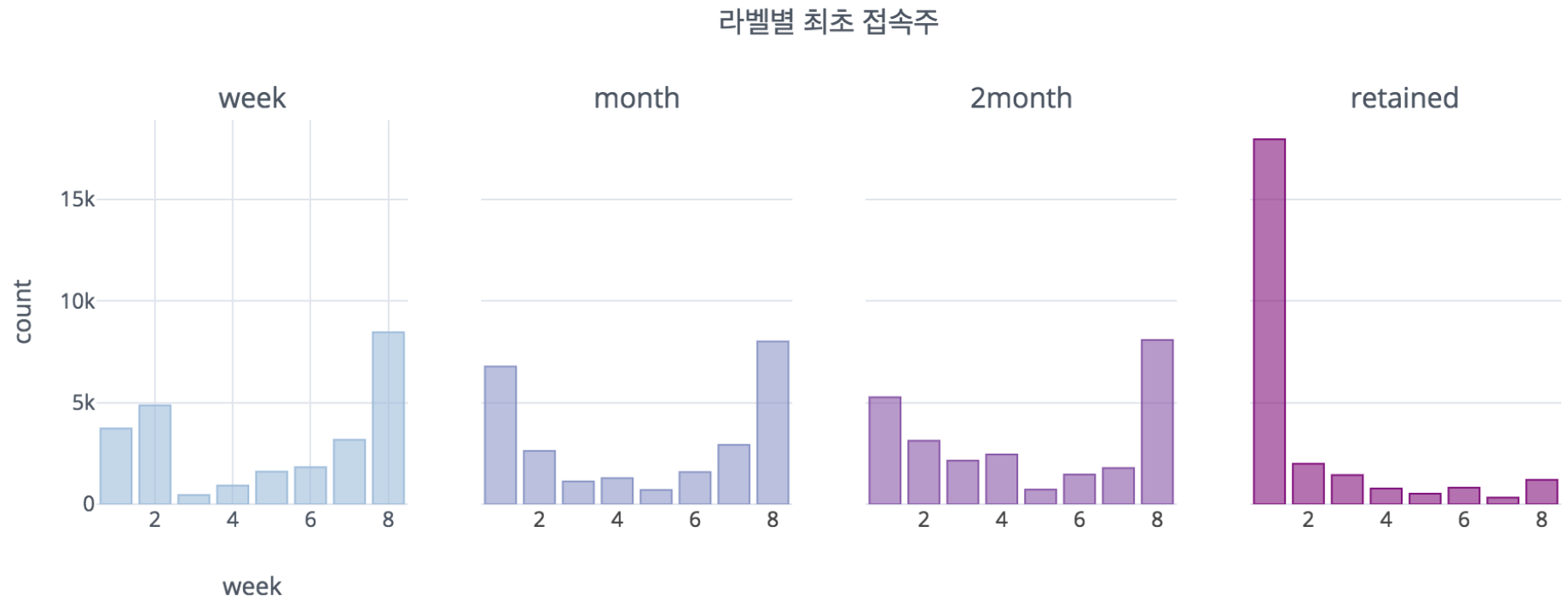
3. 최초 접속 주차의 특징

유저별 최초 접속 주차(first week)를 시각화 했을 때, 1주차와 8주차에 최초 접속한 유저들이 많았다. 이 형태가 의미하는 바가 무엇인지 EDA파트에서 살펴보았다.



Chapter 2.
EDA

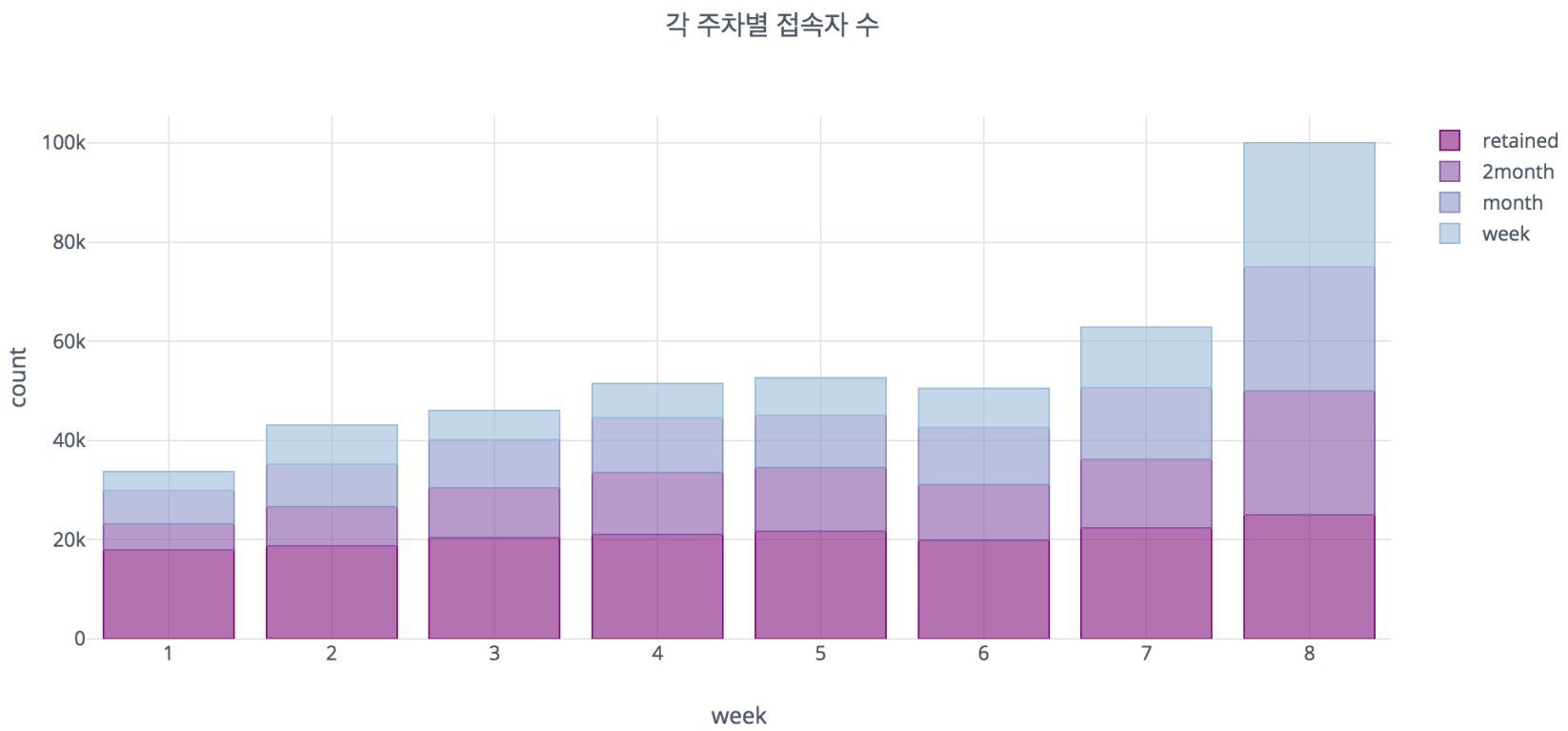
최초 접속 주차



retained 유저는 첫 접속이 1주차인 경우가 대다수이다. 즉, 이들은 거의 매주 접속하는 사람일 것이라고 추정된다.

이탈 유저는 두 종류의 사람일 것이라고 추정된다. 꾸준히 접속하던 사람들과, 데이터를 수집한 시점(8주차)에 가까운 때에 게임에 처음 또는 오랜만에 접속한 사람일 것이다.

주별 접속

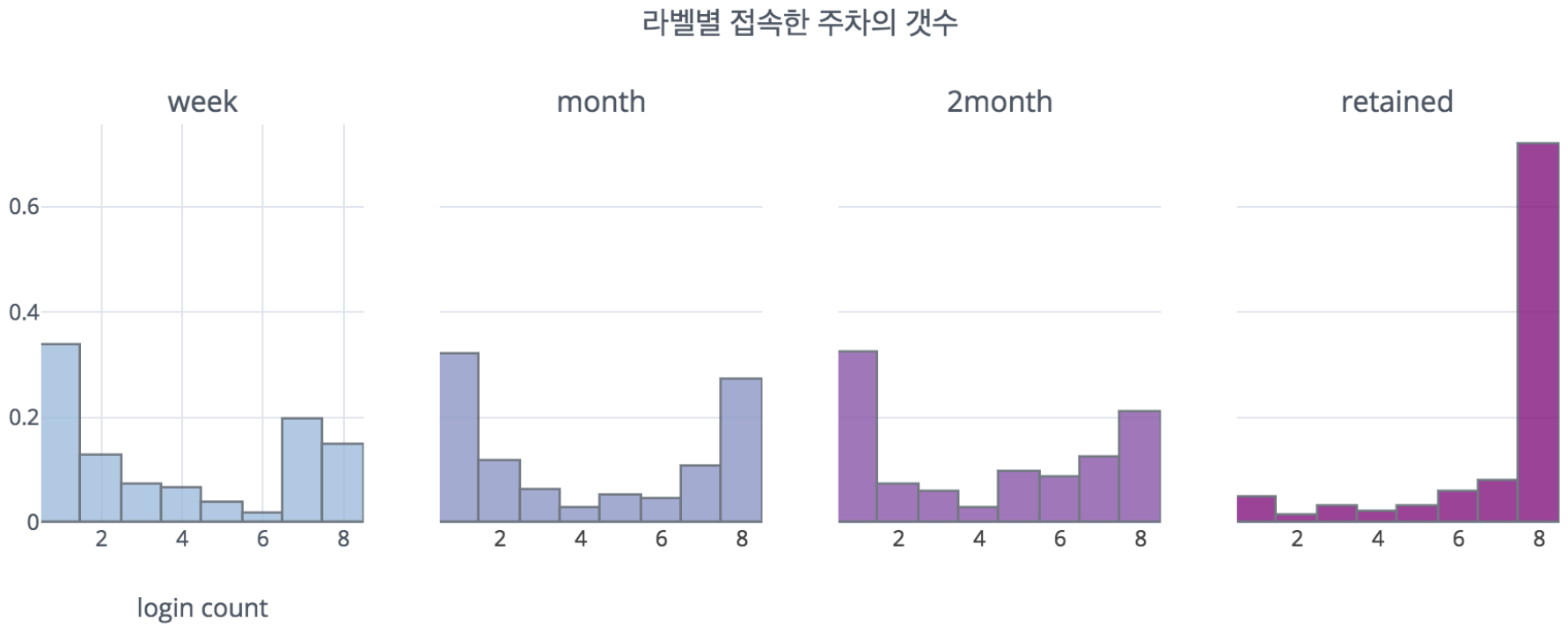


retained 유저의 대부분은 매주 접속했다는 사실을 확인할 수 있다.

이탈 유저는 매우 적은 수만큼 접속하다가, 마지막 주에 많이 접속한 것을 확인할 수 있다.

8주차에 무조건 접속한 인원을 같은 수로 뽑아서 발생하는 문제이다.

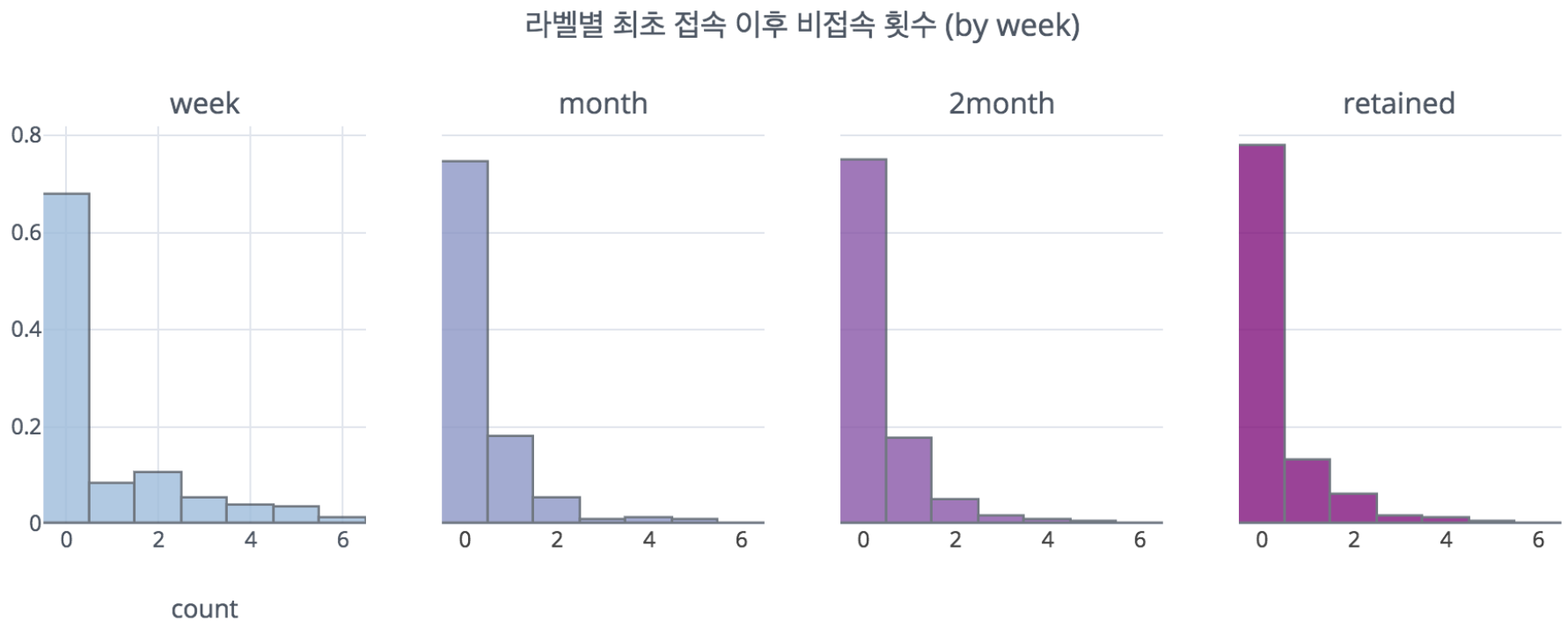
접속한 주차 횟수(1~8)



retained 유저들은 8주 모두 게임에 접속한 사람들이 압도적으로 많았다.

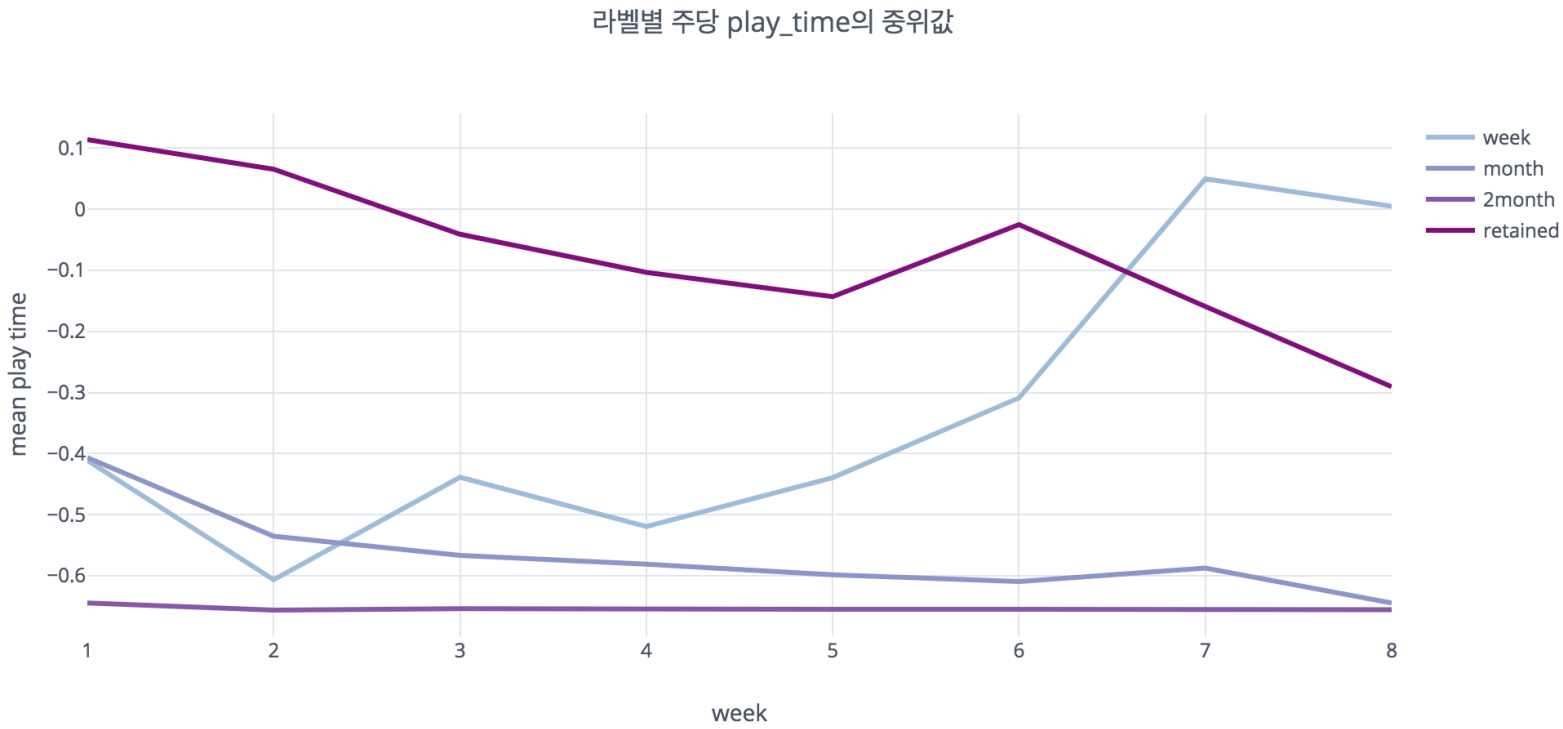
이탈 유저들은 역시 비슷한 게임 접속 분포를 보였다.

비접속 횟수



week 이탈 유저는 다양한 값을 가졌으나, 나머지 유저는 비슷한 형태를 띄었다.

플레이 시간의 차이



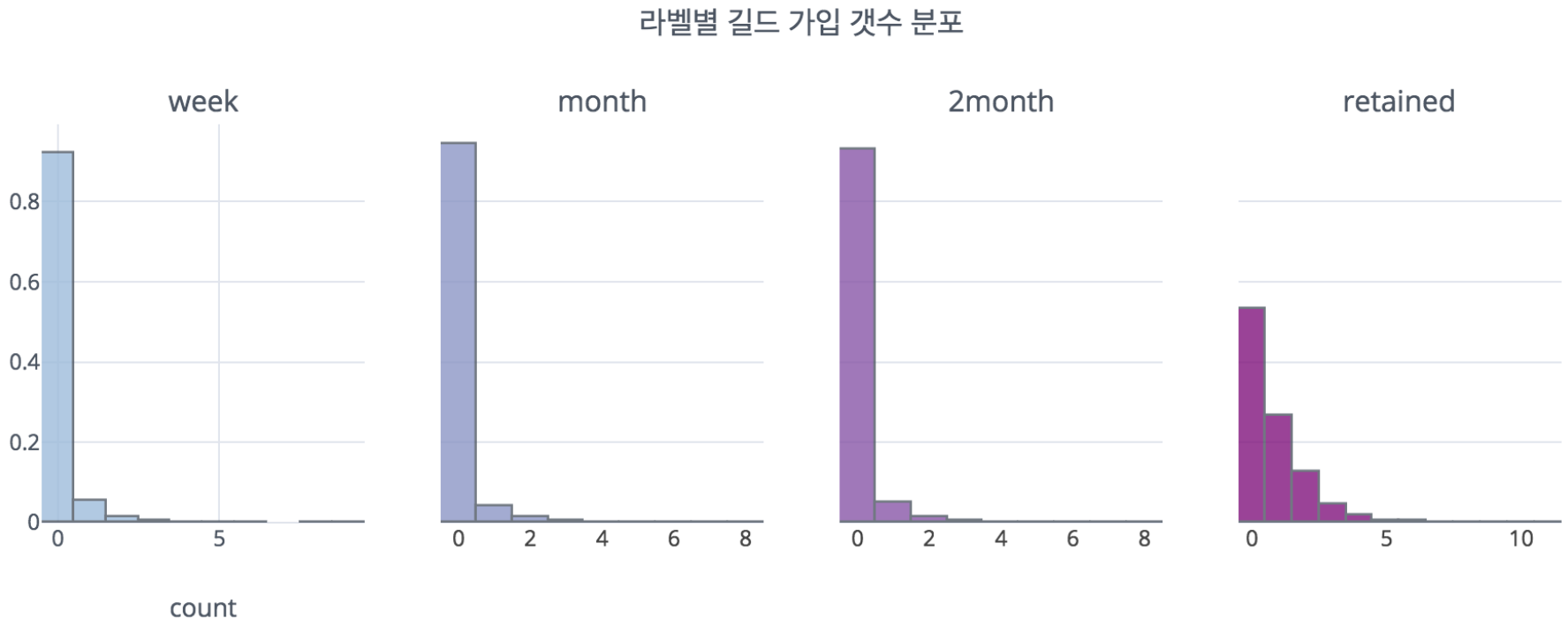
month와 2month(가장 아래 실선 두 개)는 비슷한 트랜드를 보인다.

retained 유저는 꾸준히 높은 시간을 플레이하는 경향성을 보인다.

week 유저는 신규 비율이 높을 것으로 보이므로, 초반 플레이 시간이 반영된 것으로 보인다.

retained와 week가 8주차의 playtime에서는 유사하였으나, 전체 경향성의 측면에서는 큰 차이를 보였다.

길드 활동 정도



이탈 유저들은 길드 활동을 열심히 하지 않는다.

유저의 교류정도가 이탈 여부를 예측하는 데에 중요한 변수로 사용될 수 있음을 확인했다.

EDA 총 결과

retained 유저는 대부분의 경우에 이탈 유저와 다른 데이터 형태를 보였으므로,
분류 모델이 잘 학습할 수 있을 것이라 생각되었다.

이탈 유저는 많은 변수에서 비슷한 데이터 형태를 보였으나,
week 이탈 유저는 독자적인 형태를 보이는 경우가 많았기에, 역시 잘 분류될 것이라 보았다.

month와 2month 유저는 EDA 상으로 잘 분류되는 경우가 없었기에,
이 둘을 잘 분류할 수 있도록 모델을 학습시키는 것이 주요하다고 생각되었다.

Chapter 3.

Feature Engineering

전처리와 변수 생성

제공받은 데이터 파일 6개를 acc_id 기준으로 결합하였다.

데이터/주제 이해, EDA를 바탕으로 추가적으로 변수를 생성하였다.

activity.csv

접속 주차 횟수 nplayweek	8주 중 유저가 접속한 주차의 횟수(1~8)
접속 중단 횟수 stop_n	최초 접속 이후 유저가 접속을 중단한 주차 횟수
접속 패턴 pp_cluster	유저가 접속한 주차를 1, 접속하지 않은 주차를 0으로 설정하여 유저의 8주간 접속/비접속 패턴과 라벨의 비율로 k-means cluster
밤의 바람평야 입장 이후 여부 first_manlab	블레이드앤소울 게시판을 분석한 결과, 만렙을 달성한 후 어떠한 것을 해야하냐는 질문에 '밤의 바람평야에 가서 정보를 얻어라'라는 조언이 많았음. 이에 따라 밤의 바람평야 입장을 만렙 달성 시점이라 가정하고 밤의 바람평야를 입장한 이후를 1로, 그 전을 0으로 표기
인던/던전/레이드 성공 비율 〈XX〉_success_ratio	인던/던전/레이드의 경우 완료 횟수와 참여 횟수 모두 존재함. 상관관계가 높은 두 컬럼을 하나의 컬럼으로 합치기 위해 참여 횟수 대비 완료 횟수를 구 함. XX_success ratio (cnt_clear_XX/ cnt_enter_XX) 변수 생성

추가 생성 변수

trade.csv

거래 횟수 trade_num	해당 주차에 trade한 횟수
gem 거래량 gem_trade_amount	해당 주차의 gem trade amount
아이템 거래 구매 횟수 get_〈item〉	주차 별, item_type 별 거래 중 해당 아이템을 구매한 횟수
아이템 거래 판매 횟수 put_〈item〉	주차 별, item_type 별 거래 중 해당 아이템을 판매한 횟수

추가 생성 변수

payment.csv

지불 여부 payment_if	해당 주차의 payment_amount == MIN 일 경우: payment_if = 0 해당 주차의 payment_amount 〉 MIN 일 경우: payment_if = 1

guild.csv

길드 사람수 guild_count	유저가 속한 길드의 사람 수 총합
가입 길드 수 guild_n	유저가 속한 길드 개수
길드 지불 총액 guild_paysum	유저가 속한 길드의 길드원들의 지불 총액

추가 생성 변수

party.csv

파티 주말 횟수 party_weekend_n	해당 주차의 Party를 시작한 요일이 주말인 횟수
파티 주중 횟수 party_weekday_n	해당 주차의 Party를 시작한 요일이 주중인 횟수
파티 평균 개수 party_count_relative	유저가 속한 파티의 평균 개수. 총 8주 동안의 파티 개수 총합/nplayweek
파티 개수 총합 party_count_abs	총 8주 동안의 파티 개수 총합

추가 생성 변수

party.csv

파티 친구 인원 수 multiply_tot_cnt	friend를 '10분 이상 유지된 파티에 3번 이상 함께 들어간 pair' 라고 정의하고, 한 유저당 friend 인원 수를 구함
파티 친구들의 label 비율 multiply_weight_(label)	파티 친구들의 label 비율. 해당 label에 속한 친구 수/ 친구 수 총합 multiply_weight_retained, multiply_weight_week, multiply_weight_month, multiply_weight_2month

Chapter 4.
Modeling

Deep Learning Approach

딥러닝의 대표적인 연구 분야인 NLP(자연어 처리)는 문장을 구성하는 단어를 word vector 형태로 순서대로 받아들여 문장을 분류한다.
블레이드 앤 소울의 데이터는 8주간의 데이터가 vector 형식으로 순서대로 들어오는 형태이다.

우리는 이러한 데이터와 word vector로 이뤄진 문장이 유사하다고 보았으며, NLP 방법론을 적용하여 데이터가 하는 말을 분류해보고자 하였다.

Machine Learning First Trial

정형 데이터에 대한 분류기를 만듦에 있어 의사결정나무를 토대로 한 모델성능이 우수하다는 것은 잘 알려진 사실이다.
이에 기초하여 의사결정나무 계열의 모델이 시간에 종속하는 반복측정 데이터를 학습할 수 있도록 다양한 변화를 시도하였다.

Flattened Data

1주차의 정보를 고려하면서 동시에 8주차의 정보도 고려하여야 한다.

따라서 한 acc_id당 1~8주의 데이터를 모두 가지고 있도록

flatten된 데이터프레임을 기본 데이터로서 삼는다.

week	X	Y	Z
1			
2			
…			
7			
8			

x1	…	x8	y1	…	y8	z1	…	z8

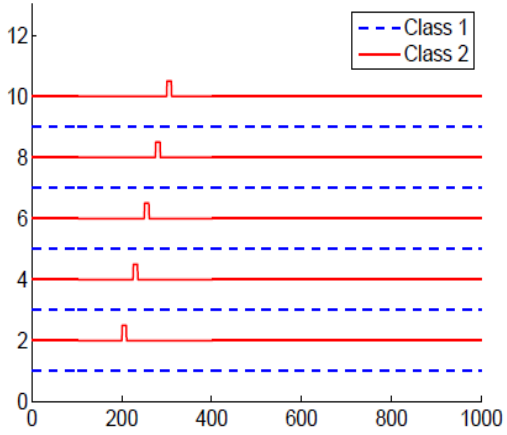
Lag Variable 추가

문제점 : 1~8주를 각각 다른 컬럼으로 설정할 경우, 시계열 데이터의 특성인 time lag를 고려할 수 없게 된다.

Rolling function을 사용하여 변수를 생성

1~8주의 데이터에 대해 window를 sliding하며 대푯값(mean,std..)을 계산한다.

이를 통해 같은 window내에 존재하였다면 lag와 관계없이 같은 대푯값을 취하여, lag를 완화시킬 수 있다. (시도한 대푯값 : mean, std, sum, diff, beta coefficient of simple linear) regression)



https://arxiv.org/abs/1302.2277

Chapter 5.

Final Ensemble Model

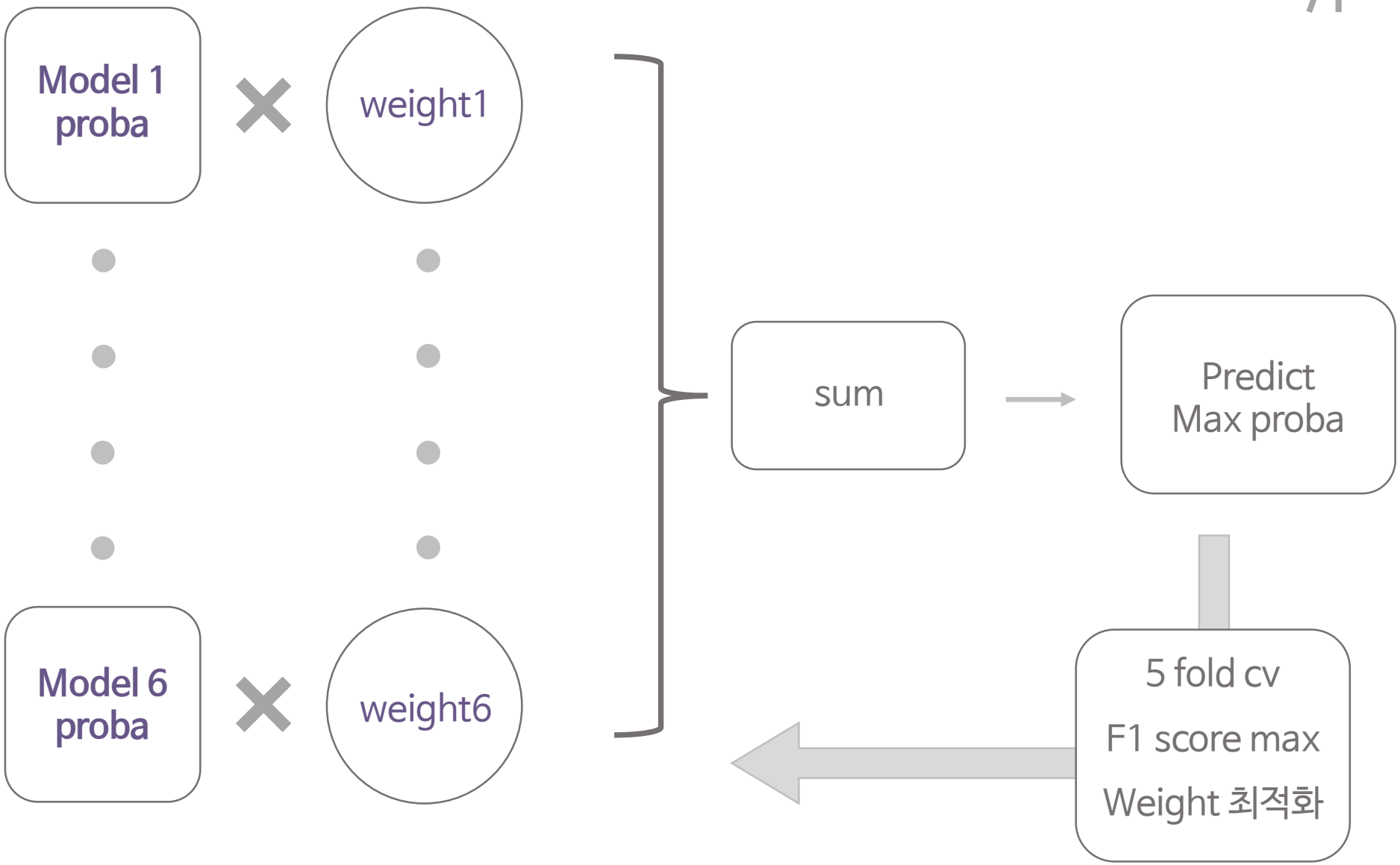
Model Ensemble

완벽한 모델은 있을 수 없다. 모든 모델에는 의도하지 않은 혹은 보완하지 못한 noise가 포함되어 있다. 그러나 각기 다르게 적합한 모델을 모두 동시에 고려하면 각각의 모델이 가지고 있는 unintended noise가 서로 상쇄되고 데이터가 가지고 있는 진짜 관계를 잡아낼 수 있다.

이에 따라 데이터의 전처리 단계부터 모델의 적합방법까지 각 팀원이 각자의 방식으로 모델을 수립하여 **각기 다른 데이터에 각기 다른 모델**을 수립하였다. 각각 다른 강점을 가지고 있고 각각 다른 측면을 바라보고 있는 모델들을 그 성능에 따라 weighted mean하여 최종 모델을 수립하였다.

우리 팀은 이 앙상블 모델을 **'Hyper Bagging'**으로 명명하였다.

Hyper Bagging



Cross Validation의 목적

해당 데이터는 8주차에 접속한 유저에 한하여 샘플링을 하였고, 동일한 라벨 수를 인위적으로 맞추었다는 점에서 특정한 제한 하에서 샘플링 된 데이터다.

또한 평가 기준 척도인 F1score는 4개의 클래스에 대해 동일한 가중치로 계산된 조화평균이다.
조화평균은 그 구조의 특성상 한 class의 성능이 떨어지는 것에 더욱 민감하게 변화한다. 따라서 신뢰할 만한 Validation score를 얻기 위해 Cross validation에 더욱 집중하고자 하였다.

이에, 앙상블 모델을 포함한 모든 모델에 대하여 최소 **5-fold이상의 Cross validation**을 하였고, 이를 통해 가능한 안정적인 Validation error의 추정치를 얻고자 하였다.

Model Part1. Feature selected Data를 사용한 모델

Model Part2. 접속 주차 별로 분리하여 학습시킨 모델

Model Part3. month와 2month를 분류하기 위한 모델

Model Part1. Feature selected Data를 사용한 모델

Feature engineering을 통해 만들어진 변수들은 flatten할 경우 그 수가 8배가 되므로, 데이터의 차원이 지나치게 커져 모델이 학습하는데 장애가 된다.

따라서 만들어진 변수들을 **stepwise방식으로 순차적으로 추가**하고,

이를 Cross Validation을 통해 반복적으로 확인하여 **성능의 향상을 가져 온 변수들만을 이용**한다.

(stepwise에 대한 상세한 설명은 부록에 기재하였다)

이를 통해 모델이 학습을 수월하게 할 수 있도록 유도하고, 학습시간을 줄여 다양한 시도를 가능하게 하였다.

Model Name

RandomForest

Key Idea

가장 기본적인 앙상블 모델인 Random forest를 적합.

각 모델을 학습하며 custom loss function으로써 각 **4개의 class**에 대한 precision& recall,

final f1_score, final accuracy를 반환하도록 하여 **모델이 학습하지 못하는 부분을 파악**.

모델이 학습하기 힘든 class에 대하여 weight를 더 준다.

Model Name

XGboost (with grid search)

Key Idea

모델이 **쉽게 예측하지 못하는 class가 정해져 있다.**

따라서 예측이 힘든 class 분류에 대하여 강점을 갖는 **boosting 계열 모델을 사용하였다.**

Xgboost 는 hyper parameter에 성능이 많이 의존하므로 5-cv grid search를 통해 파라미터를 최적화하였다.

Model Part2. 접속 주차 별로 분리하여 학습시킨 모델

유저가 접속하지 않은 주는 학습을 위해 weekly min 값으로 padding 되어 있다.

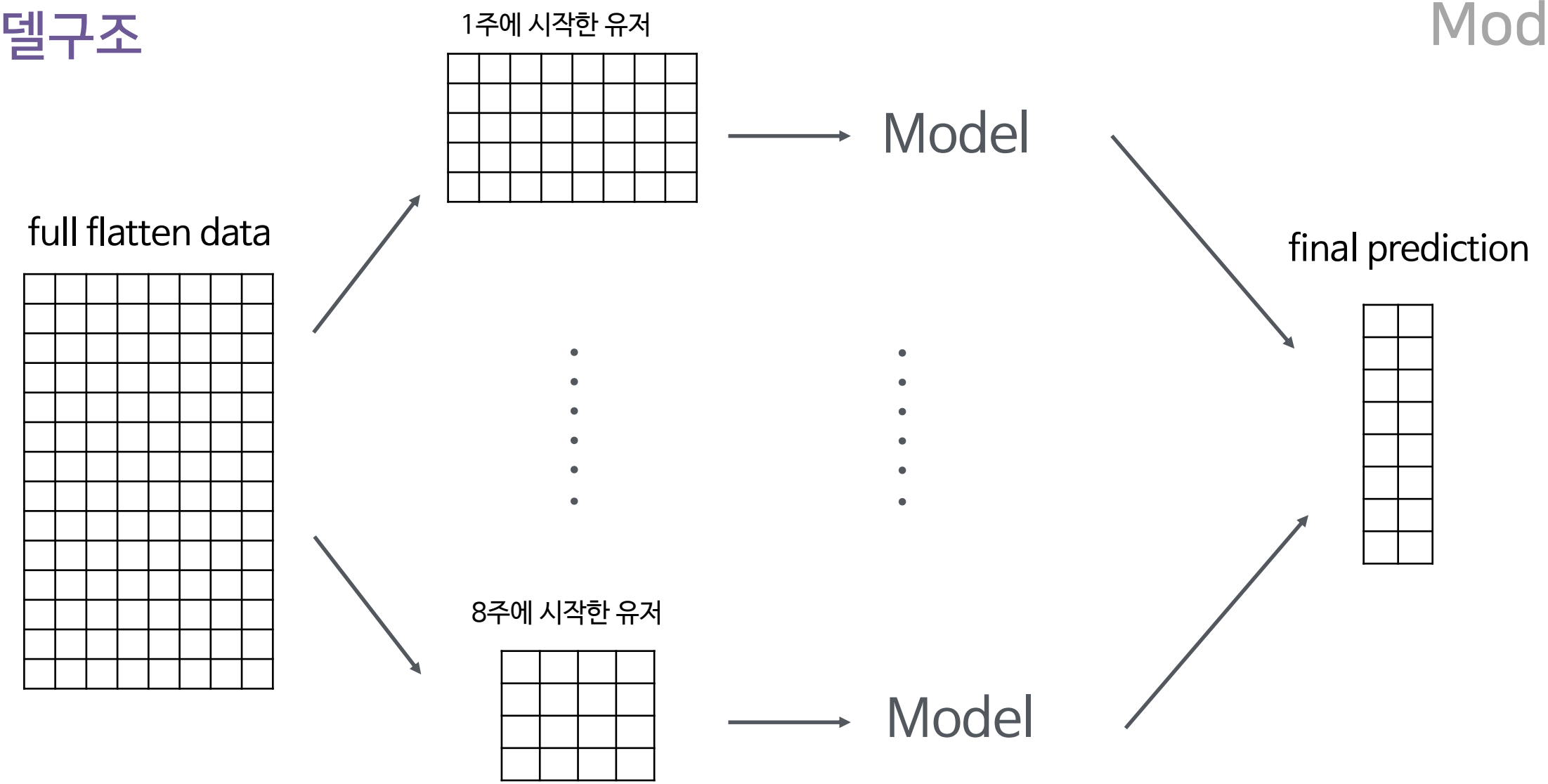
하지만 이 경우, 유저의 첫 접속 이전에 생성된 padding 데이터는 유저와 무관하다.

따라서 첫 접속이 언제였는지에 따라 8 종류의 유저로 분류한 후, 각 유저마다 **첫 접속 이후의 데이터를 활용**한다.

이에 따라 각 유저를 분류할 때 noise로 작용하는 **첫 접속 이전의 padding 값의 영향을 줄일 수 있다**.

모델구조

Model 3


1주에 시작한 유저
Model
full flatten data
final prediction
8주에 시작한 유저
Model

모델명

Beginning Week Considerate Random Forest

key idea

데이터의 특성상 비선형 모델로 적합하는 것이 좋다.

비선형 모델의 대표주자 중 하나인 random forest를 활용한다.

Bagging의 효과를 최대화하기 위해 parameter grid search를 활용한다.

모델명

Beginning Week Considerate XG Boost

key idea

Beginning Week Considerate Random Forest 와 동일하다.

Bagging과 Boosting 각각이 **강점을 가지는 부분이 다르기 때문**에 앙상블 단계에서 **협업**을 통해 전체 성능을 높이고자 하였다.

Model Part3. month와 2month를 분류하기 위한 모델

적합된 모델의 4개의 클래스에 대한 성능을 보았을때, 2month와 month를 구분하는 것에 어려움이 있었다.

이는 분석을 시작하기전 파악한 **multi class**특성과 관련이 있다. 특히, non retained 클래스는 인위적으로 분절된 형태이다. retained와 week의 중간단계인 2month와 month의 경우, 클래스가 **의미적으로 매우 유사하기에,** 모델이 4개의 클래스의 두드러진 특성을 동시에 학습하는데 어려움이 있다.

이에 따라 month와 2month를 잘 구분하여 학습할 수 있도록 모델의 구조를 수정해준다.

Model Name

One Vs Rest classifier using Random Forest Classifier

key idea

클래스간의 연관관계에 관계없이 **각 클래스만의 특성을 예측**할 수 있도록 모델을 유도하기 위해 **One Vs Rest Classifier기법**을 채택했다.

각 클래스를 예측하기에 적절한 **파라미터를 각각 설정**해주고 이를 최종적으로 합쳐 모델의 성능을 향상했다.

Model Name

Classifier Chain using Random Forest Classifier

key idea

Classifier Chain은 multi class 문제를 해결하기 위한 방법으로 'One vs Rest Classifier'와 함께 대표적으로 사용하는 모델이다.

성능 측정

모델명	F1 score(leader board 기준)	weight
Feature Selected Random Forest	0.7296	0.5
Feature Selected XGboost	0.7224	1.55
Beginning Week Considerate Random Forest	0.7258	0.6
Beginning Week Considerate XG Boost	0.7256	1
One Vs Rest classifier using Random Forest Classifier	0.7299	0.6
Classifier Chain using Random Forest Classifier	0.7281	1.3

Hyper Bagging

F1 score = 0.7374

Chapter 6.

유저 이탈 원인 추정

분류 모형 이용

유저를 **이탈/잔류** 두 범주로 구분 후 분류 알고리즘 수행.

잘 학습된 알고리즘을 이용하여, 예측에 사용된 **유의미한 변수 선택.**

선택된 **유의미한 변수와 이탈여부의 관계를 분석**하여 이탈 원인 파악

추정 방법의 문제

이상 속의 모델

모델의 분류 성능(설명력)이 좋아야한다

독립변수와 종속변수의 관계를 볼 수 있어야한다.

현실 속의 모델

관계를 볼 수 있지만 **설명력이 낮거나**

ex)로지스틱 회귀모형

설명력이 높지만 **관계를 볼 수 없다**

ex) Random Forest, XG Boost

LIME

머신 러닝 알고리즘이 **왜 그렇게 판단했는지** 알려준다.

블랙박스 모형을 열어볼 수 있게 도와준다.

문제 해결

성능이 좋은 모델은, 독립변수와 종속변수 간의 깊은 관계를 파악하였을 것이다.

LIME을 이용하여, 모델이 파악한 관계를 열어보자

HOW ??

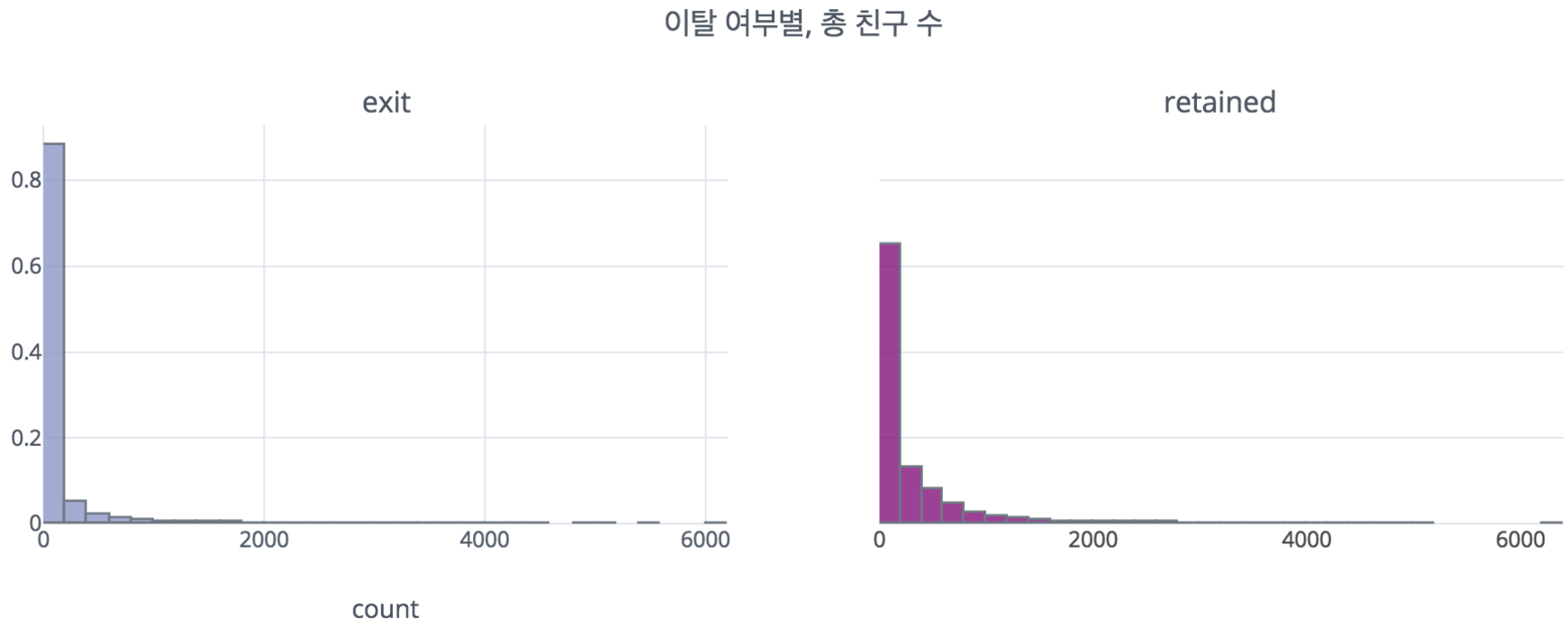
피쳐 엔지니어링이 완료된 두 개의 서로 다른 데이터를 활용하여, 이탈/잔류를 학습한다.

Cross-validation accuracy를 90% 이상으로 올린다.

실제 이탈 유저를 이탈할 것이라고 예측한 경우에, 주요하게 이용된 feature를 모은다.

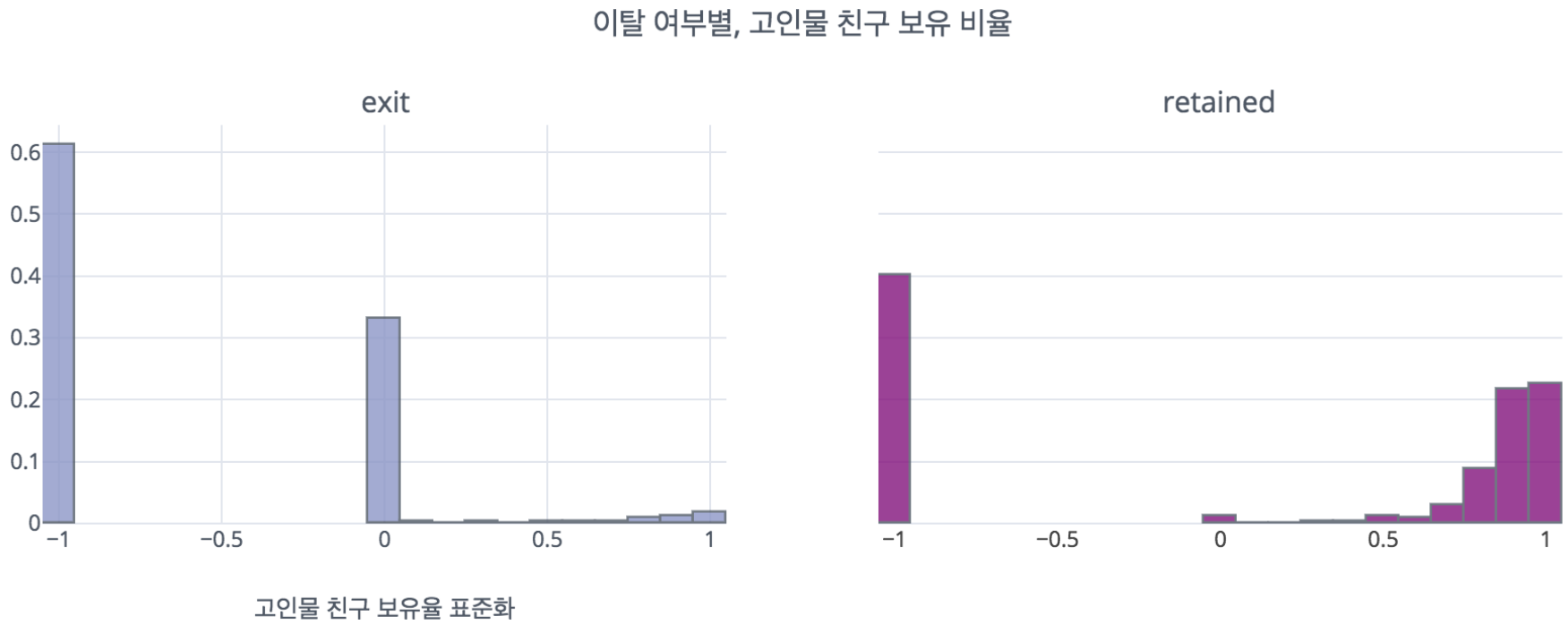
가장 많이 사용된 feature들을 모아 순위를 매긴다.

동료가 있는가



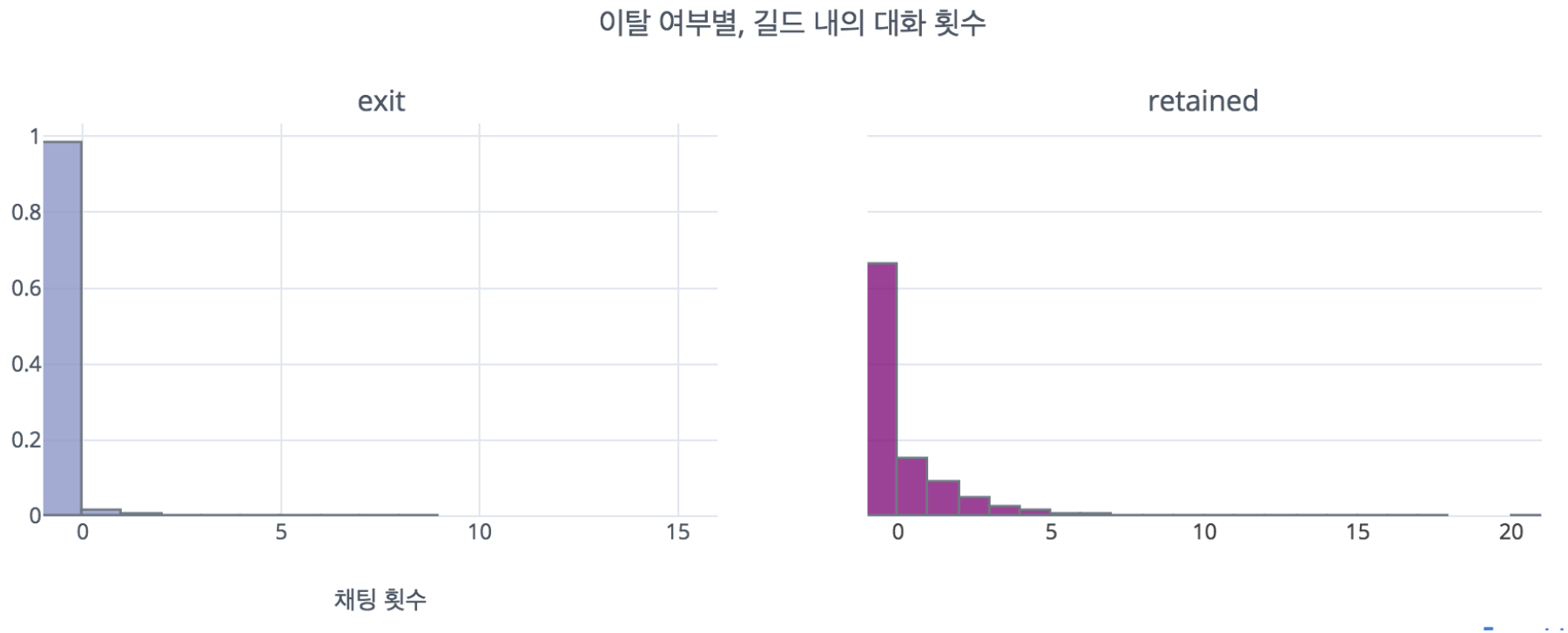
잔여 유저들은 길드, 파티 등으로 관계를 맺는 사람의 숫자가 상대적으로 많다.

동료가 있는가



잔여 유저들은 상대적으로 고여있는 사람들과 관계를 맺고 있는 경우가 많다.

동료가 있는가



잔여 유저들은 길드원들과의 대화가 좀더 활발하다.

동료가 있는가

실제 생활에서도 **친한 사람이 없는 모임**에 나가고 싶지 않듯,

들어가도 **함께할 친구가 없는 게임**은 들어가고 싶지 않다.

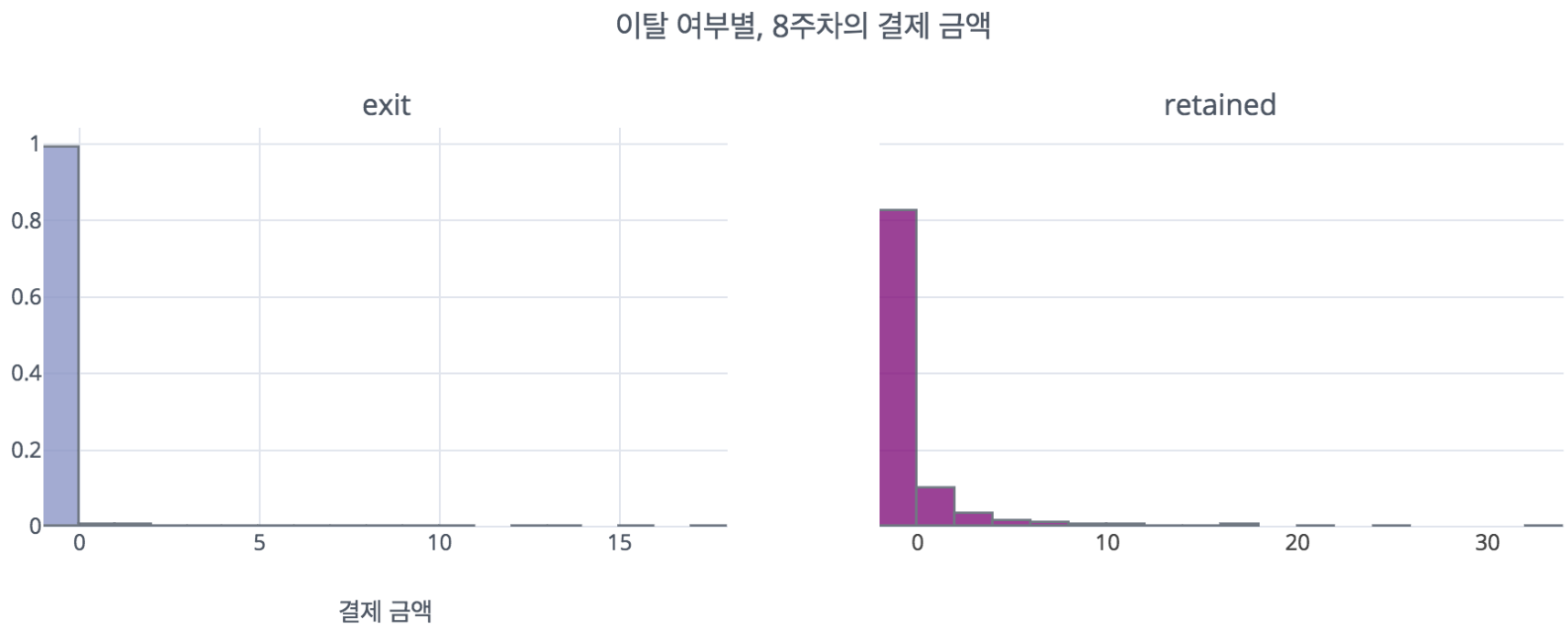
특히, 나에게 도움을 줄 수 있는 사람이 없다면 더더욱 그렇다.

소통은 인간을 더욱 행복하게 한다.

훌륭한 의사소통은 불랙커피처럼 자극적이며,후에 잠들기가 어렵다.

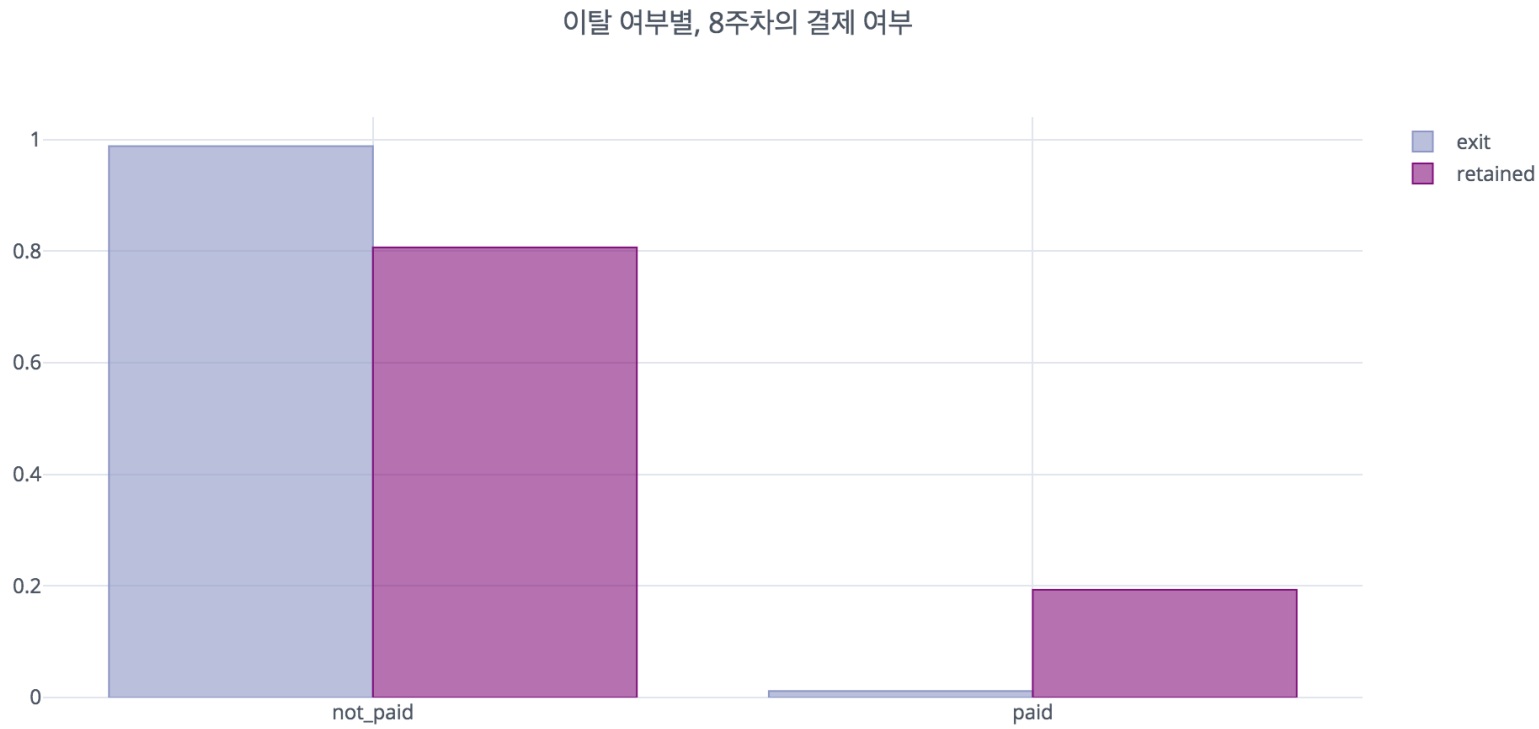
- 린드버그 -

과금 유저인가



잔여 유저들은 이탈유저보다 결제를 많이 했다.

과금 유저인가



단순화해서 살펴보면, 이탈/잔여 그룹의 결제비율이 확연히 다른 것을 알 수 있다.

과금 유저인가

게임머니 뿐만이 아니라, 실제 현금을 투입한 유저들이 더 오래 남았다.

이는 결제를 해서 오래남은 것이 아니라, 결제를 하지 않아서 떠난 것이다.

과도한 과금 유도는 이전부터 해당 게임에 대해 제기된 문제이다.

왜 이탈하는가

게임을 함께 플레이할 동료가 존재하지 않는다.

자신을 도와줄 열성 유저가 존재하지 않는다.

게임에 들어가도 대화할 사람이 없다.

현금 결제를 하고싶지 않다.

이탈 방지 대책

예상 이탈유저에게 함께 플레이할 플레이어를 이어준다.

게임 플레이 과정에서 타유저와 소통할 기회를 만들어준다.

소통의 즐거움을 깨닫게 하여, 블랙커피처럼 잠들지 못하게 한다.

게임 유저 이탈 예측

블린이

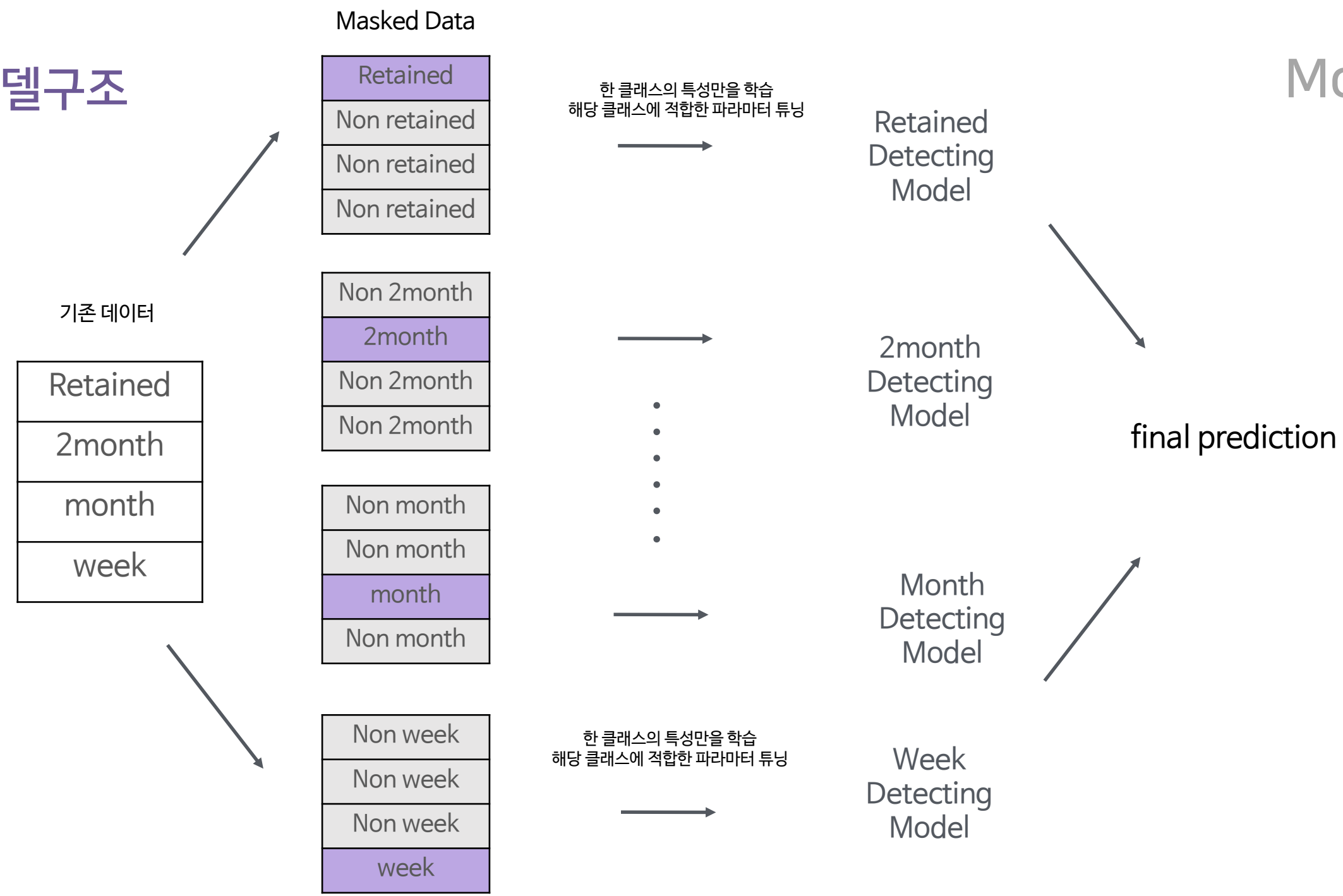
고동영, 김현우, 김혜주, 손진원, 조현호

Big Contest 2018

Chapter 7.
Appendix

모델구조

Model 5


Masked Data
Retained
Non retained
Non retained
Non retained
한 클래스의 특성만을 학습
해당 클래스에 적합한 파라마터 튜닝
Retained Detecting Model
기존 데이터
Retained
2month
month
week
Non 2month
2month
Non 2month
Non 2month
2month Detecting Model
final prediction
Non month
Non month
month
Non month
Month Detecting Model
Non week
Non week
Non week
week
한 클래스의 특성만을 학습
해당 클래스에 적합한 파라마터 튜닝
Week Detecting Model

모델구조

Classifier Chain 특징1.

다수의 label을 동시에 분류하지 않고,
label 하나씩 순서대로 해당 label에 속하는지 아닌지 classification을 진행한다.

step1

X	y1
x1	0
x2	1
x3	0
…	…

step2

X	y1	y2
x1	0	1
x2	1	0
x3	0	1
…	…	…

step3

X	y1	y2	y3
x1	0	1	1
x2	1	0	0
x3	0	1	0
…	…	…	…

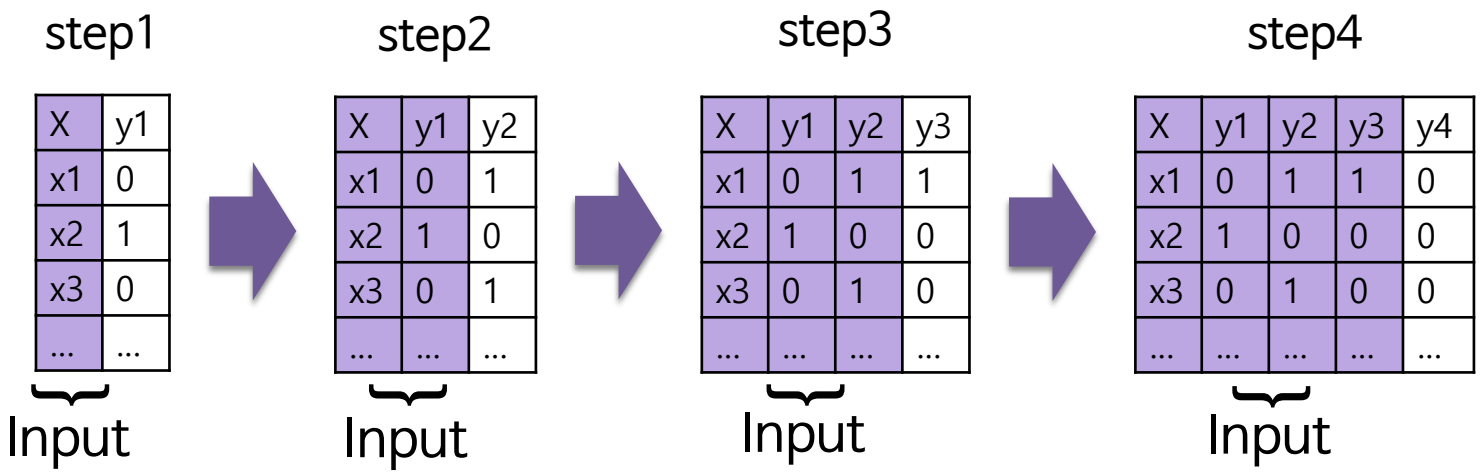
step4

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0
…	…	…	…	…

모델구조

Classifier Chain 특징2.

label correlation을 고려하여 기존의 X변수만 아니라
이전 classifier의 분류 결과를 고려한다.



모델구조

Label Order List			
STEP1	**STEP2**	**STEP3**	**STEP4**
month	week	retained	2month
month	retained	week	2month
retained	month	2month	week
retained	2month	month	week
2month	week	month	retained
2month	retained	month	week

총 6개의 label order list 중
f1 score가 가장 높게 나온
2개의 classifier chain을 앙상블

LIME

https://www.youtube.com/watch?v=hUnRCxnydCc

LIME은 어떻게 판단하는가?

(Local Interpretable Model-agnostic Explanations)

- ✓ 복잡한 모델을 해석하는 것에 대해 이것을 전체적으로는 이해하기 어렵지만, 주변적으로는 이해가 가능하다는 것이 LIME의 주요한 가정 중 하나이다.
- ✓ 해석이 어려운 복잡한 모형을 해석이 가능한 모형으로 근사시킨다.
- ✓ 관찰자가 변수 x에 관심이 있다면, x를 중심으로 임의의 값을 생성한 뒤 x의 값이 변함에 따라 prediction이 얼만큼 바뀌는지를 측정한다. 만약, 바뀌는 정도가 크다면, 이것은 해당 변수가 중요함을 의미한다.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why should i trust you?: Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016.