



# 게임 유저 이탈 예측 모형 : 블레이드 & 소울

유저 이탈 분석 보고서  
: 유저 행동 양식을 중심으로

2018 빅 콘테스트 (Analysis 분야)

팀명 : 도토리

2018

September



# 목차

## Intro

대규모 다중 사용자 온라인 롤플레이팅 게임(MMORPG)은 전세계적으로 많은 사용자들이 플레이하는 게임의 장르이다. 게임의 장르가 많은 사용자를 보유한 만큼, 사용자들의 이탈 또한 지속적으로 일어난다. 사용자들의 이탈은 기업의 수익과 밀접한 관계가 있으므로 사용자들의 이탈 예측은 기업들에게 중요한 숙제이다.

이번 공모전에서 우리는 제공받은 “블레이드 앤 소울”의 데이터를 이용하여 사용자들의 유형을 분류하였다. 또한 정제된 데이터의 중요도를 비교하여 어떠한 변수들이 유저 이탈에 작용하였는지 예상하고, 분류된 변수들을 사용하여 이탈 예측 모델을 구축하였다. 이탈 예측 모델로 저희는 XG-Boost를 사용하였고, 74.2의 accuracy와 73.23의 F-score(사전 평가)의 성능을 보였다.

## Table of contents

### 1. 분석결과 및 제언

- 1-1. 데이터 맵핑
- 1-2. 집단 별 특징
- 1-3. 이탈유형에 따른 원인
- 1-4. 이탈유형에 따른 방지 방안
- 1-5. 실패보고서

### 2. 데이터 분석 과정

- 2-1. 변수 선택 과정
- 2-2. 모델링

### 3. 부록: 데이터 탐색

- 3-1. Activity
- 3-2. Guild
- 3-3. Trade
- 3-4. Payment
- 3-5. Party



## Category 01

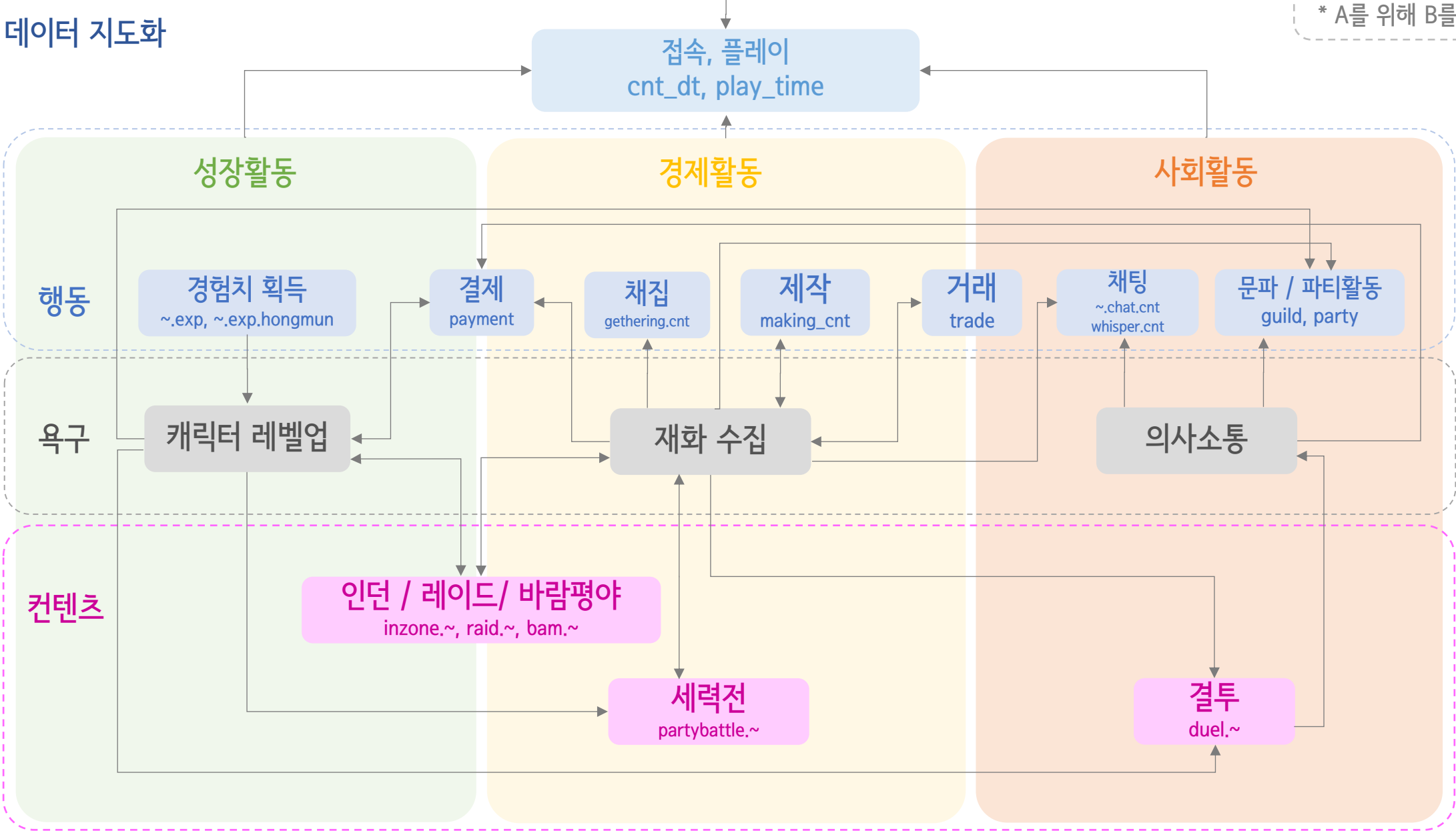
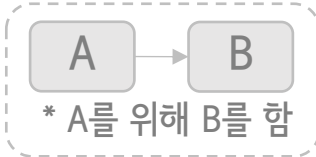
### 분석결과 및 제언

- 1-1 \_데이터 맵핑
- 1-2 \_집단 별 특징
- 1-3 \_이탈유형에 따른 원인
- 1-4 \_이탈유형에 따른 방지 방안
- 1-5 \_실패보고서

1. 분석결과 및 제언 \_ 1-1. 데이터 맵핑

전체 데이터 지도화

유저이탈에 영향을 미치는 요소 분석



# 1. 분석 결과 및 제언 \_ 1-2 집단 별 특징(Activity)

앞선 데이터 탐색결과와 이를 바탕으로 설정한 각 집단 별 특징

	Week	Month	2month	Retained
성장활동	캐릭터 성장활동이 활발함	Week와 유사한 양상을 보이거나 수치가 더 낮음	캐릭터 성장 활동을 거의 하지 않음	아이템을 이용한 경험치를 주로 획득
레이드, 인던	라이트 인던, 레이드 등 진입장벽이 낮은 콘텐츠를 즐김	레이드와 인던을 즐기는 유저가 많지 않음	레이드와 인던을 즐기는 유저가 많지 않음	숙련인던, 레이드 등 진입장벽이 높은 콘텐츠를 즐김
전투	NPC, 몹들과 주로 전투함	전투를 자주 하지 않음	전투를 자주 하지 않음	PvP 콘텐츠를 즐김
접속, 플레이 시간	접속횟수는 적으나 접속 시 오랫동안 접속해있음	-	상대적으로 접속 횟수와 시간이 낮음	가장 꾸준히 접속함
캐릭터 수	주로 한 캐릭터만 키우는 것으로 예상됨	-	-	부 캐릭터를 다수 갖고 있는 것으로 예상됨
사회활동	인던, 레이드, 퀘스트 클리어를 위한 파티활동			커뮤니티 활동 활발함

# 1. 분석 결과 및 제언 \_ 1-2 집단 별 특징(Guild)

앞선 데이터 탐색결과와 이를 바탕으로 설정한 각 집단 별 특징

	Week	Month	2month	Retained
문파 가입여부	문파 가입비율이 각 집단별 약 10~15%밖에 되지 않음			과반수 이상이 문파에 가입
부 케릭 문파 가입 여부	약 8~10%의 소수에 인원만 부케릭을 문파에 가입 하고 있음			약 20%(5,000명)가 부 케릭을 문파에 가입을 하고 있음
문파 규모	주로 1인 문파 혹은 16인 이상 대규모 문파에 가입하고 있음	개인, 소, 중규모 문파보다는 대규모 문파에 약 2배 이상 가입하고 있어 대규모 문파를 선호하고 있는것으로 보여짐 Month -> 2Month -> Retained 이동하면서 대규모 문파 가입여부의 비율이 커짐		과반수 이상의 유저들이 대규모 문파에 가입하고 있음

# 1. 분석 결과 및 제언 \_ 1-2 집단 별 특징(Trade)

앞선 데이터 탐색결과와 이를 바탕으로 설정한 각 집단 별 특징

	Week	Month	2month	Retained
평균 판매 횟수	가장 적은 판매 횟수를 가짐	-	가장 많은 판매 횟수를 가짐	-
평균 구매 횟수	가장 적은 구매 횟수를 가짐	-	가장 많은 구매 횟수를 가짐	-
평균 재화 판매횟수	가장 적은 판매 횟수를 가짐	-	가장 많은 판매 횟수를 가짐	-
평균 재화 구매횟수	-	가장 많은 구매 횟수를 가짐	-	가장 적은 구매 횟수를 가짐
잡화 판매횟수	가장 적은 판매 횟수를 가짐	-	가장 많은 판매 횟수를 가짐	-
잡화 구매횟수	가장 적은 구매 횟수를 가짐	-	가장 많은 판매 횟수를 가짐	-

# 1. 분석 결과 및 제언 \_ 1-2 집단 별 특징(Payment)

Payment 데이터 탐색결과와 이를 바탕으로 설정한 각 집단 별 특징

	Week	Month	2month	Retained
과금여부	과금 유저가 가장 적게 위치	과금 유저가 두번째로 적게 위치	과금 유저가 세번째로 적게 위치	대부분의 유저가 위치
과금수준	평균 과금액 가장 낮음	평균 과금액 두번째로 낮음	평균 과금액 세번째로 낮음	헤비 과금러의 80%이상을 차지 평균 과금액 높음
데이터 특징	과금 유저가 많은 집단일수록 이탈시기가 늦음 과금의 양이 많은 집단일수록 이탈시기가 늦음			



## 1. 분석 결과 및 제언 \_ 1-2 집단 별 특징(Party)

### 앞선 데이터 탐색결과와 이를 바탕으로 설정한 각 집단 별 특징

	Week		Month		2month		Retained
전반적 파티 활동	저조	◀	미비	◀	보통	◀	활발
평일 파티 활동	저조	◀	미비	◀	보통	◀	활발
주말 파티 활동	활발	▬	활발	▬	활발	▬	활발
결론	Week에서 Retained로 갈수록 파티 활동이 활발하게 일어나는 것을 확인할 수 있음. 주말에만 파티 활동이 활발해지는 유저들을 평일에도 파티 활동을 왕성하게 할 수 있도록 유치하는 전략이 필요함.						

# 1. 분석 결과 및 제언 \_ 1-3. 이탈 유형에 따른 원인

## 이탈 유형에 따른 집단별 데이터 특성을 살펴보고 유저들이 이탈 원인을 추정

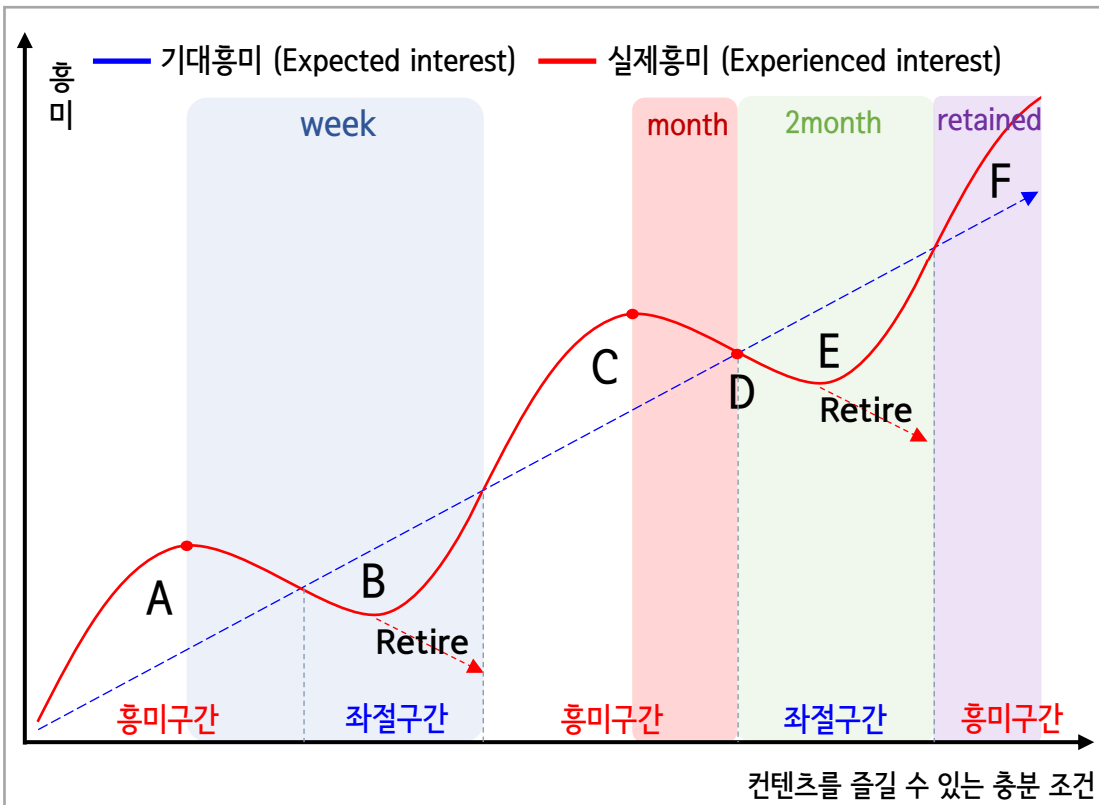
유저들의 주된 이탈 원인은 결국 본인이 기대했던 게임에 대한 흥미(재미)의 기대치와 실제로 게임을 플레이 하면서 체감한 흥미(재미)의 격차에서 발생하는 좌절구간이 주 원인으로 생각됐다.

컨텐츠를 즐길 수 있는 충분 조건

캐릭터 측면의 요소 (캐릭터 레벨, 직업, 장비수준, 재력 등)

+

플레이어 측면의 요소 (게임에 대한 이해, 컨트롤 실력, 상황대처 능력, 인맥 등)



A → B : 컨텐츠 진입장벽 ↑↑, 게임에 대한 흥미 ↓

C → D : 컨텐츠 진입장벽 ↑, 재화의 필요성 ↑, 게임에 대한 흥미 ↓

D → E : 성장에 대한 한계 체감 ↓, 게임에 대한 흥미 ↓

E → F : 많은 컨텐츠 플레이 ↑, 과금 ↑, 문파 활동 ↑

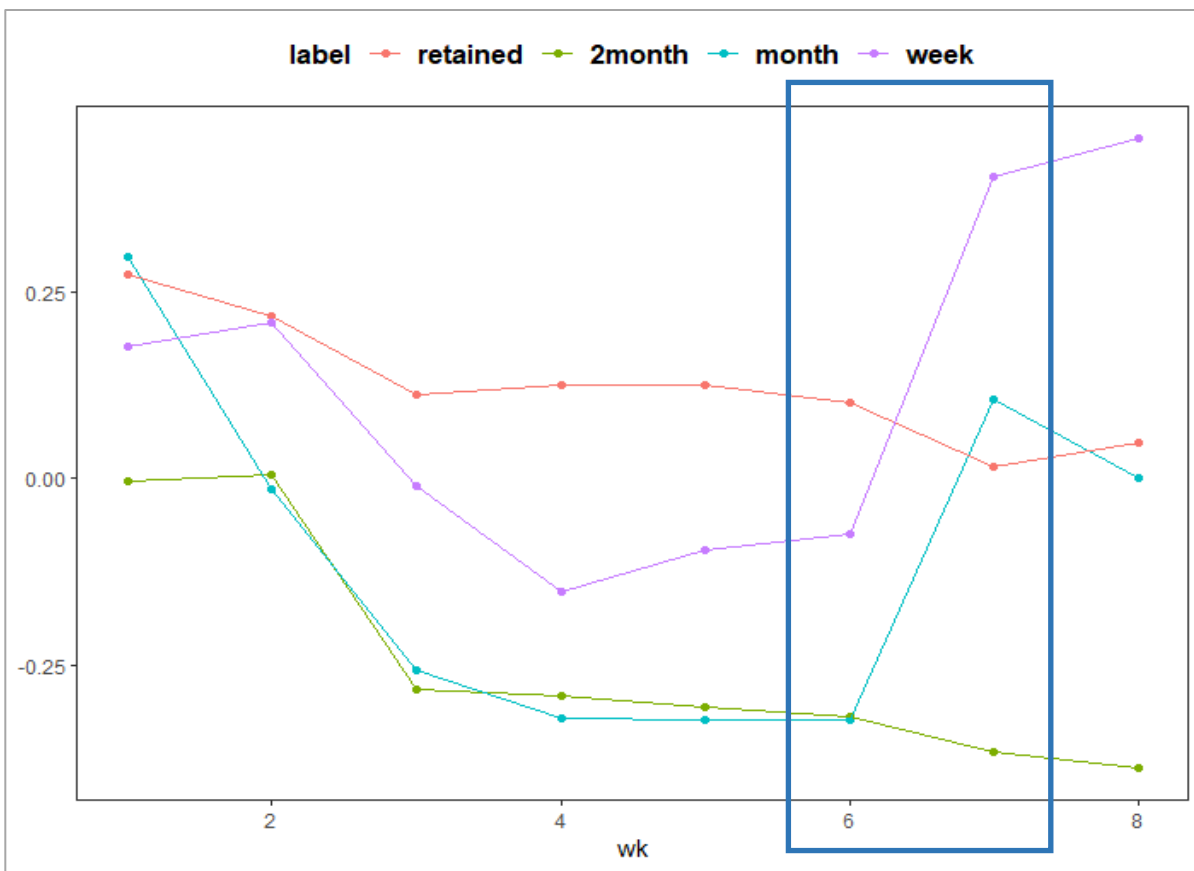
	Week	Month	2month	Retained
Activity	접속 횟수는 적으나 접속 시 플레이 시간이 길며 주로 한 캐릭터를 키우고 있음. NPC 퀘스트와 사냥을 위한 전투 플레이 캐릭터 성장 활동을 활발히 하며, 라이트 인던, 레이드 등 진입장벽이 낮은 컨텐츠를 주로 즐기며 클리어를 위한 파티활동을 하고 있음.	Week집단과 유사하나 성장 활동은 줄어들고 레이드와 인던 등 컨텐츠를 즐기는 부분이 많아짐	성장활동이 거의 멈췄을 뿐만 아니라 레이드와 인던을 즐기는 부분이 적어짐. 상대적으로 접속 횟수와 플레이 시간이 적음.	아이템을 이용한 경험치를 주로 획득하며 숙련인던, 레이드, PvP 등 진입장벽이 높은 컨텐츠를 즐김. 집단 내 가장 꾸준히 접속하며 커뮤니티 활동이 활발한것으로 보이며 부 캐릭터를 다수 갖고 있는 것으로 추측됨.
Guild	문파 가입비율이 매우 낮으며, 주로 1인 문파로 혼자 플레이하거나 대규모 문파에 가입하는 양상을 보임	문파 가입비율이 매우 낮으며, 1인문파, 소, 중 규모 문파보다는 대규모 문파를 선호함	문파 가입비율이 매우 낮으며, 1인문파, 소, 중 규모 문파보다는 대규모 문파를 선호함	과반수 이상이 문파에 가입하고 있으며 20%이상이 부캐릭을 문파에 가입시켜 놓음
Trade	-가장 적은 재화, 잡화 판매 횟수 -가장 적은 잡화 구매 횟수	-가장 많은 재화 구매 횟수	-가장 많은 재화, 잡화 판매 횟수 -가장 많은 잡화 구매 횟수	-가장 적은 재화 구매 횟수
Payment	-과금 유저가 가장 적음 -평균 과금액 가장 낮음	-과금 유저가 두번째로 적음 -평균 과금액 두번째로 낮음	-과금 유저가 세번째로 적음 -평균 과금액 세번째로 낮음	-대부분의 유저가 위치 -헤비 과금러의 80% 이상을 차지하며, 평균 과금액 높음
Party	-전반적 파티활동 저조	-전반적 파티활동 미비	-전반적 파티활동 보통	-전반적 파티활동 활발
이탈 원인	A → B	C → D	D → E	E → F

# 1. 분석 결과 및 제언 \_ 1-4. 이탈 유형에 따른 이탈 방지 방안

## Week 이탈에 따른 처치 방안

1. 이벤트 활성화 : 게임의 흥미를 높이기 위해서, 경험치의 축적이 쉽지 않아 게임을 지루할 수 있게 하는 홍문 경험치를 위해 이벤트를 진행하여, 홍문 경험치 버프를 받을 수 있도록 돕는다.
2. 낮은 수준의 콘텐츠 추가 : 라이트 인던과 숙련 인던 사이의 적절한 수준의 인던과 같은 콘텐츠를 추가하여, 갑작스런 진입장벽을 없애 week 유저로의 이탈을 예방

week의 경우 갑작스럽게 6주차부터 퀘스트 홍문 경험치가 급증하는 것을 볼 수 있다.  
이는 6주차부터 신입 유저보다 장시간 미접속 유저(복귀유저)의 유입이 큰 것으로 확인해 볼 수있음

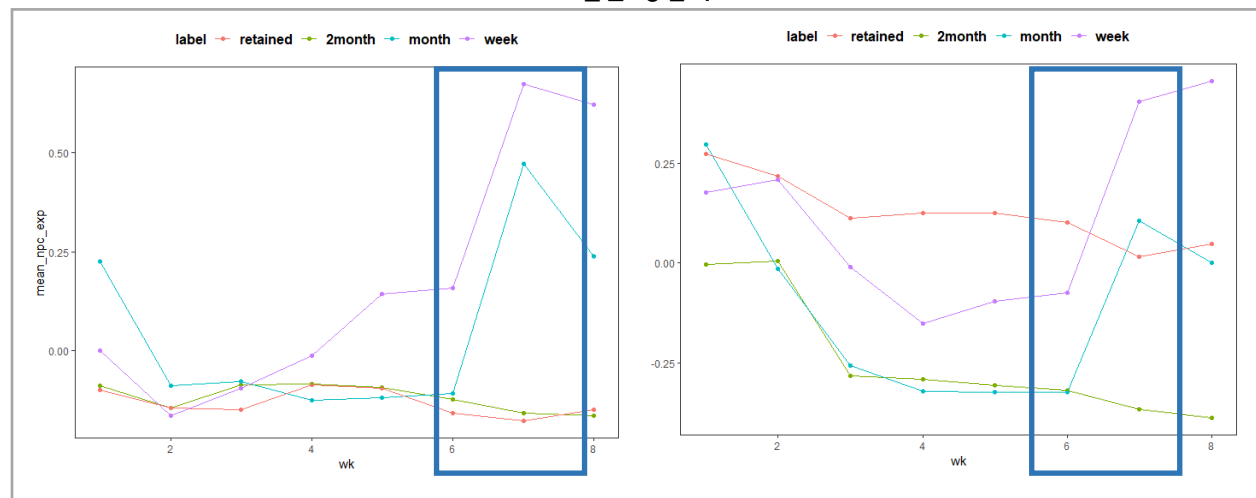


하지만 일반경험치의 양도 week의 경우가 6주차부터 크게 증가하는 것으로 보아 게임 진행 초반 유저 (신입 유저)의 유입도 많은 것으로 보여진다.

week유저는 무과금 유저가 대부분이고, 대체로 게임 진행 도중 새로운 콘텐츠의 높은 진입장벽으로 인한 게임의 흥미를 잃어간 유저들이다. 이 때 week의 54%의 유저들이 6주차부터 데이터가 존재한 유저들이며, 이 중 신입유저일 경우 처음 게임 시작의 의욕은 강했지만, 게임 스타일이 자기 자신과 맞지 않거나. 과도한 의욕등으로 시작했으나, 길드나, 파티의 횃수가 적은 것으로 보아 지인의 도움이 없었던 것으로 파악하여 진행에 어려움을 겪었을 것으로 본다.

복귀유저의 경우 '다시 게임을 시작해 볼까?'라는 의욕으로 진행하였지만, 게임을 장시간 미접속으로 이끈 콘텐츠 진입장벽에 다시 가로막혀 게임에 금방 흥미를 잃은 것으로 볼 수 있다.

### <일반 경험치>



# 1. 분석 결과 및 제언 \_ 1-4. 이탈 유형에 따른 이탈 방지 방안

## Month, 2Month 이탈에 따른 처치 방안

1. 과금 아이템 구매 가능한 보상 콘텐츠 : 과금 아이템을 무과금 유저도 게임 플레이 시간과 노력을 들여 보상을 받아 구매할 수 있도록 하는 콘텐츠 추가
2. 컨트롤 위주의 콘텐츠 : 배틀로얄 / AOS 처럼 모든 유저가 동등한 조건에서 플레이하여 게이머자체의 실력 향상을 도모해주고 흥미를 올려주는 콘텐츠 추가

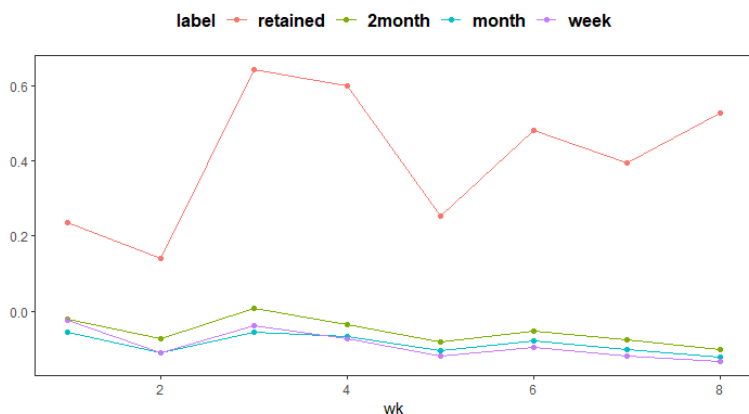
label	평균 재화 구매 횟수
retained	52.9
2month	389.5
month	488.2
Week	141.2

과금의 양과 홍문경험치, 일반경험치의 그래프를 보면, 대체로 week와 유사한 특징을 보인다.

하지만 Month의 경우 week와는 달리 높은 진입장벽에 바로 무너지지 않고 진입장벽에 맞서 조금 더 노력을 해본 유저들의 집단이다. week, 2month, retained들과 비교하여 상대적으로 재화의 구매 횟수가 많았던 것으로 보아 재화가 많이 필요한 무기성장을 위해 재화를 얻으려 했으며, week에 비해 진입장벽에 맞서 성장을 해보려 노력을 한 것으로 보인다.

2Mounth의 경우 Month와 매우 유사한 성질을 지닌 집단이며, Month의 비해 좀 더 인내심이 있는 집단이라 생각된다.

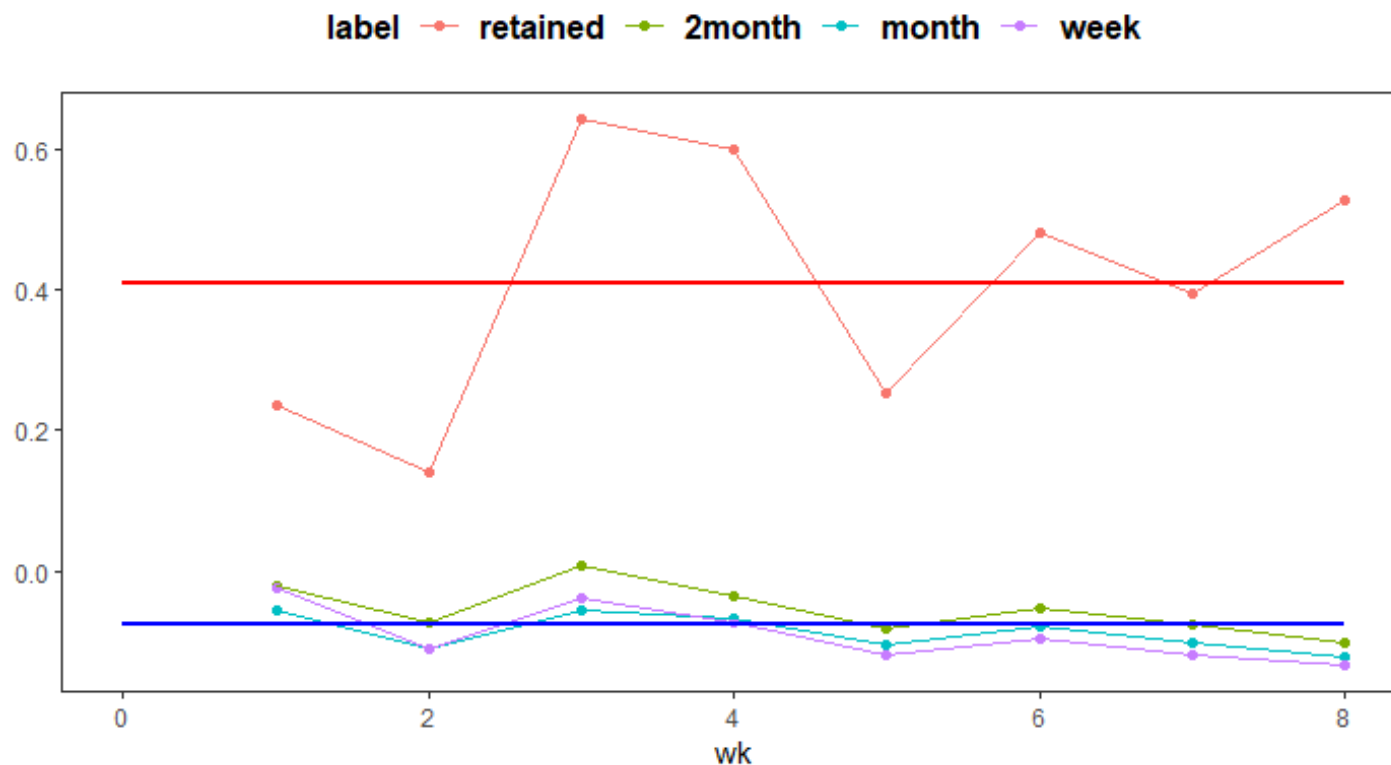
평균  
과금의 양



# 1. 분석 결과 및 제언 \_ 1-4. 이탈 유형에 따른 이탈 방지 방안

## 이탈 여부에 따른 방지 방안

이탈자와 비이탈자의 차이가 가장 큰 것은 평균 과금 금액이다. 이는 유저들이 게임을 유지하기 위해서는 과금의 필요성이 높다고 느끼기 때문이라고 본다. 때문에 과금이 적은 유저들의 경우 게임의 진행에 한계를 가지며 이탈자로의 방향을 잡는 것으로 생각된다. 이를 막기 위해서 과금의 부담감을 줄이는 노력성 이벤트(특정 목표 달성을 할 때 혜택을 주는 이벤트)나, 기존 콘텐츠의 진입장벽을 낮춰 조금 더 원활하게 게임을 유지해 나갈 수 있도록 하는 것이 게임 이탈을 방지하는 중요한 요인이 될 것이다.



# 1. 분석 결과 및 제언 \_ 1-5 실패 보고서

## 1. 시도했지만... 실패한 사례들

Trial	모델학습에 Party Data이용	유저를 데이터 보유여부로 분류 후 각각 모델링	Bayesian Optimization을 통한 Hyper-parameter최적화	Bagging	접속주차에 따른 유저분류 및 변수 추가
시도 배경	가장 용량이 큰 Party Data가 제공됨	유저 별로 Data보유 여부가 달랐으므로, 모든 데이터로 학습시킨 모델은 데이터가 적은 유저를 설명하지 못할 것이라고 예상	XGBoost는 하이퍼 파라미터 조정을 통해 모델 성능개선을 기대할 수 있지만, 너무 많은 경우의 수가 있으므로 효율성을 위해 Bayesian Opt사용	높은 bias로 인한 Underfitting , 높은 분산으로 인한 Overfitting을 중화하기 위해 사용되며, Keggel 수상 팀들 추천하는 기법이라는 정보를 얻음	유저별 접속 주차에 따른 128개의 유형이 존재했고 이 유형들에 대해서 구분자 변수 추가
내용	데이터 정제에 10일 이상 소요, 총 200여 개의 파생변수 생성	유저들을 데이터 보유 여부에 따라 Segmentation	각 하이퍼 파라미터 조합을 Bayesian Opt를 이용하여 Cross Validation	모델링하여 도출되는 결과(label)들을 모아 각 Feature들로 만들어 모델학습, 또는 그 결과값들의 평균을 구함	유저별로 128개 유형 중 속하는 값을 원핫 인코딩을 통해 부여 후 모델링
예상 결과	정확도 향상이 예상됨	정확도 향상이 예상됨	모델 성능이 개선될 것이라 예상	모델의 과도한 Overfitting 감소가 예상	세분화된 자료로 인해 정확도 향상을 예측함
실제 결과	Train데이터에서는 73%에서 75%로 예측률이 증가했으나, Test데이터에서는 67%로 급락	Train 데이터에서는 90% 이상의 예측률을 보였으나, Test 데이터에서는 65%의 낮은 예측률을 보임	예측률의 미미한 수준의 증감 (0.1~0.05 % 이내)이 이루어짐. 투입되는 시간대비 예측률 증가가 적고, 감소한 경우도 있었음	결과값들의 수를 늘릴수록 모델의 성능이 조금씩 개선됐지만, 결과값들이 충분히 많지 않았음	기존 activity 데이터의 평균값을 이용한 모델의 정확도 보다 3.0%p 이상 떨어짐
실패 원인	Train과 test의 party데이터가 서로 비슷한 특징을 갖지 않는 것으로 보임	각 Segment별로 유저의 수 편차가 크며, 때문에 모델학습에 충분한 데이터를 갖지 못한 것으로 예상	학습시킨 모델의 성능자체가 뛰어나지 않은 경우, 하이퍼 파라미터 설정은 미미한 수준의 변화만을 가져온다는 정보를 찾아볼 수 있었음.	프로젝트 진행 동안 수 없이 실시한 모델링들의 조건과 결과를 기록저장 해 놓아야 한다는 점을 간과	128개 유형 중 8개 유형 외에, 나머지 유형은 볼륨이 적기 때문에 오류가 커져 정확도가 떨어진 것으로 예상

# 1. 분석 결과 및 제언 \_ 1-5 실패 보고서

## 2. 구상했으나... 실패한 사례들

구상	Guild별 점수화	Classification을 통한 봇 구분	Guild데이터 외 Guild 가입 이력이 있는 유저	아이템을 판매량 변화에 따른 이탈 예상
구상 배경	문파에 가입하는 사람들과 문파 설립의 목적은 저마다 다르므로, 문파에 가입한 사람끼리는 같은 목적을 가지고 있으며 이 목적들이 이탈여부에 영향을 미친다고 가설을 세움	채집, 채팅 등 단순 반복행동의 횟수가 비정상적으로 높은 데이터들을 발견. 특정행동을 반복하는 봇 혹은 매크로라고 판단	Guild 데이터가 없지만, Activity 데이터에 guild_chat 값이 최소가 아닌 유저들이 존재 따라서 이 유저들은 Guild 데이터가 입력되기 전 문파에 가입되어 있었지만 탈퇴한 이력이 있는 유저라 판단	보통 게임유저들은 이탈하기 직전 아이템을 지인들에게 뿌리거나 판매하여 현금화하는 경우가 많음
내용	문파마다 성장활동, 경제활동, 사회활동을 점수화	K-means 클러스터링을 통해 봇과 일반유저를 분류	Guild데이터와 해당 유저 label 확인	주차별/일별 Trade data 변동량 조사
예상 결과	문파별 성장, 경제, 사회활동 점수를 매기고, 이를 바탕으로 유저 특징을 유추	봇 유저는 이익을 극대화하기 위해 계속 봇을 가동시키므로 retained 일 것이라고 예상	탈퇴한 이력이 있고 Guild 데이터에는 없는 유저라 이탈자일 가능성이 높다 추측	갑자기 아이템 판매량이 급증한 유저는 곧 이탈할 것이라고 예상
실패 원인	문파당 가입된 유저들의 데이터를 모두 가지고 있는 경우가 드물기 때문에, 문파 내 일부 유저들로 전체 문파의 특성을 대표하는 것은 위험하다고 판단	봇은 단순행동만 반복하는 것 외에, 전투 같은 복잡한 행동을 반복하는 종류도 존재. 또한, 봇을 사용하다가 이탈하거나 제재를 받는 경우도 존재하기에 retained라고 확정지을수 없었음	Guild 데이터에는 없지만 guild_chat이 있는 유저들이 839명으로 소수 유저였으며, retained의 비율이 55%로 높은 비율이었으며, 이탈자일 것이라고 성급히 판단할 수 없었음	아이템 변동량이 급증하는 경우는 1. 부 캐릭터에서 본 캐릭터로 아이템을 옮길 때, 2. 값비싼 특정 장비를 구매하기 위해 단시간에 모든 아이템을 판매할 때, 3. 갑자기



## Category 02

### 데이터 분석 과정

2-1 \_ 변수 선택 과정

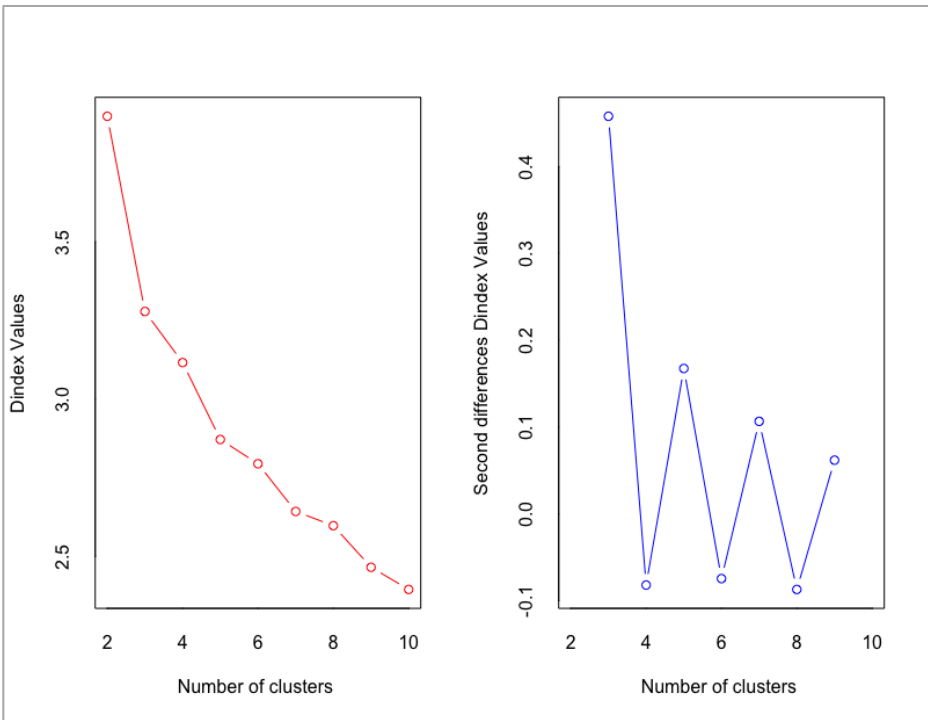
2-2 \_ 모델링



## 2. 데이터 분석 과정 \_2-1 K-means

### K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

10만명의 다양한 성향을 지닌 유저들을 하나의 모형으로 예측 시 모형에 대한 설명력 뿐만 아니라 정확도가 낮아질 수 있기때문에 K-means 이용하여 군집분석을 진행했다.



\*\*\*\*\* Conclusion \*\*\*\*\*

According to the majority rule,  
the best number of clusters is 3

군집분석을 시도한 결과 가장 최고의 군집 개수는 3개로 나왔다.  
하지만 다양한 특성의 유저들을 더 잘 설명하기 위해  
군집의 개수를 4개로 선정하였다.

	그룹 1	그룹2	그룹3	그룹4
cnt_dt	3.455331	6.365712	4.667238	5.148702
wk_n	4.236131	7.871101	1.81319	4.853933

	그룹 1	그룹2	그룹3	그룹4
retained	0.20684	0.900869	0.03532	0.257882
2month	0.300586	0.053838	0.042685	0.105814
month	0.27457	0.031619	0.398728	0.111705
week	0.218004	0.013673	0.523268	0.524599

chat	그룹 1	그룹2	그룹3	그룹4
normal_chat	-0.0209	0.125958	-0.00442	0.02692
guild_chat	-0.28963	1.827559	-0.34197	-0.32702
party_chat	-0.23017	1.442721	-0.2112	-0.15174
faction_chat	-0.02178	0.131822	-0.02394	-0.02365
whisper_chat	-0.19675	1.057199	-0.22324	-0.05118
district_chat	-0.00828	0.040264	-0.01654	-0.01642

4개의 군집으로 나눈 결과 그룹 2는 모든 채팅에서 다른 그룹보다 상대적으로 높은 값을 보였다. 반대로 그룹 3의 경우 다른 그룹보다 상대적으로 낮은 채팅 횟수를 보였다.

그룹마다 label의 비율을 확인해 본 결과 그룹2는 내부에서 retained의 비율이 가장 높았으며, group3의 경우 retained의 비율이 다른 그룹에 비해 상대적으로 그룹 내부에서 가장 낮은 비율을 가졌다.

이는 데이터 탐색 과정에서 채팅의 값이 높은 retained의 특징과 채팅의 값이 낮은 이탈자 집단의 특징을 잘 반영하는 것을 확인해 볼 수 있다.

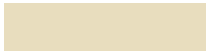

또한, 그룹 2에서는 평균 활동 주 횟수가 8에 가깝고, 그룹 3에서는 평균 활동 주 횟수가 1~2 사이에 있다. 즉, 그룹 2에는 8주 모두 데이터가 있는 유저가, 그룹 3에는 8주에만 데이터가 있는 유저가 많은 비중을 차지하고 있다는 것을 알 수 있다.

이를 바탕으로 4개의 군집의 특징을 반영하여 유저들을 4개의 그룹으로 나누었다. 4개의 그룹은 각각 개별의 특성이 강하므로, 전체를 함께 모델링할 경우 정확한 이탈 예측을 하기가 어렵다. 그래서 각 그룹 특성에 따른 변수를 추가하고 모델링을 진행하였다.

## 2. 데이터 분석 과정 \_ 2-1 변수 선택 과정

### K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

K-means의 결과로 이루어진 4개의 집단의 평균 wk를 배경으로 유저들을 4개의 그룹으로 분류한다.

 : 그룹에서 가장 큰 비중  
 : 그룹에서 가장 작은 비중

Group1	24,045명(24.05%)	retained	14823명	0.616	1~8 주 모든 주차의 데이터가 존재하는 유저 집단
		2month	3,453명	0.144	
		month	4,112명	0.171	
		week	1,657명	0.0689	
Group2	33,925명(33.93%)	retained	1,523명	0.0449	7,8주 & 8주 데이터만 존재하는 유저 집단
		2month	9,855명	0.29	
		month	10,924명	0.322	
		week	11,623명	0.343	
Group3	15,805명(15.81%)	retained	3,098명	0.196	group1의 유저 외에 결석한 주(wk)가 없는 유저 집단
		2month	5,418명	0.343	
		month	3,591명	0.227	
		week	3,698명	0.234	
Group4	26,225명(26.23%)	retained	5,556명	0.212	결석한 주(wk)가 있는 유저 집단
		2month	6,274명	0.239	
		month	6,373명	0.243	
		week	8,022명	0.306	

## 2. 데이터 분석 과정 \_ 1-1 변수 선택 과정

### K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

각각의 group 마다의 Activity 데이터의 특징을 반영해서 새로운 변수를 추가한다.

#### group1

새로운 18개의 변수를 생성

od_play_time	플레이 시간의 표준편차
per_cnt_dt	출석일수/전제일수(56일)
cv_play_time	플레이 시간의 변동계수
max_rate_of_change_npc_exp	npc 사냥경험치의 최대 경험치 변화율
min_rate_of_change_npc_exp	npc 사냥경험치의 최소 경험치 변화율
max_rate_of_change_npc_hongmun	npc 홍문사냥경험치의 최대 경험치 변화율
min_rate_of_change_npc_hongmun	npc 홍문사냥경험치의 최소 경험치 변화율
max_rate_of_change_quest_exp	퀘스트 경험치의 최대 경험치 변화율
min_rate_of_change_quest_exp	퀘스트 경험치의 최소 경험치 변화율
max_rate_of_change_quest_hongmun	퀘스트 홍문 경험치의 최대 경험치 변화율
min_rate_of_change_quest_hongmun	퀘스트 홍문 경험치의 최소 경험치 변화율
max_rate_of_change_item_hongmun	아이템 홍문 경험치의 최대 경험치 변화율
min_rate_of_change_item_hongmun	아이템 홍문 경험치의 최소 경험치 변화율

maxmin_rate_of_change_npc_exp	max_rate_of_change_npc_exp - min_rate_of_change_npc_exp
maxmin_rate_of_change_npc_hongmun	max_rate_of_change_npc_hongmun - min_rate_of_change_npc_hongmun
maxmin_rate_of_change_quest_exp	max_rate_of_change_quest_exp - min_rate_of_change_quest_exp
maxmin_rate_of_change_quest_hongmun	max_rate_of_change_quest_hongmun - min_rate_of_change_quest_hongmun
maxmin_rate_of_change_item_hongmun	max_rate_of_change_item_hongmun - min_rate_of_change_item_hongmun

#### group2

8주에만 데이터가 있는 유저들이 속한  
group2에는 한 개의 변수만 추가.

변수	설명
start_game	데이터가 처음 시작된 주를 표시

#### group3

group1, group2의 새로운 변수들과 함께  
2개의 새로운 변수를 추가로 생성.

변수	설명
Diff_cnt_dt	7~8주 cnt_dt의 기울기
Diff_play_time	7~8주의 play_time의 기울기

#### group4

group1, group2의 새로운 변수들과 함께  
3개의 새로운 변수를 추가로 생성.

변수	설명
absence_Game	8주 내 데이터 없는 주의 합계
max_conti_Absence	최대 연속 결석 주
Danger_retained	최대 연속 결속 주를 이용한 위험도

## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

RandomForest 모델을 사용하여 각 집단별 이탈여부를 예측하고 전체 종합도를 살펴본 결과 72.28%로 기존 모델보다 약 1.61%p 상승했다.

전체 정확도 : 71.17% -> 72.28% (1.61% **상승**)

기  
존



신  
규

group1

OOB estimate of error rate: 20.3%  
Confusion matrix:  
2month month retained week class.error  
2month 1720 178 1537 18 0.50188242  
month 452 2662 857 141 0.35262646  
retained 833 76 13891 23 0.06287526  
week 29 195 541 892 0.46167773

정확도 변화 :  
79.70% -> 80.87%  
1.17%p 상승

OOB estimate of error rate: 19.13%  
Confusion matrix:  
2month month retained week class.error  
2month 1546 170 1722 15 0.55227339  
month 380 2746 889 97 0.33219844  
retained 522 60 14217 24 0.04088241  
week 21 174 527 935 0.43572722

group2

OOB estimate of error rate: 32.55%  
Confusion matrix:  
2month month retained week class.error  
2month 5947 3321 52 535 0.39654997  
month 3681 5726 48 1469 0.47583303  
retained 632 304 236 351 0.84504268  
week 52 557 39 10975 0.05575153

정확도 변화 :  
67.45% -> 67.52%  
0.07%p 상승

OOB estimate of error rate: 32.48%  
Confusion matrix:  
2month month retained week class.error  
2month 5958 3308 58 531 0.39543379  
month 3693 5731 51 1449 0.47537532  
retained 640 298 242 343 0.84110309  
week 59 551 38 10975 0.05575153

group3

OOB estimate of error rate: 27.48%  
Confusion matrix:  
2month month retained week class.error  
2month 4905 38 358 117 0.09468439  
month 1867 1224 234 266 0.65914787  
retained 758 52 2133 155 0.31149128  
week 70 220 209 3199 0.13493780

정확도 변화 :  
72.52% -> 73.89%  
1.37%p 상승

OOB estimate of error rate: 26.11%  
Confusion matrix:  
2month month retained week class.error  
2month 4824 117 363 114 0.1096346  
month 1727 1369 246 249 0.6187691  
retained 651 77 2229 141 0.2805036  
week 49 213 179 3257 0.1192537

group4

OOB estimate of error rate: 32.67%  
Confusion matrix:  
2month month retained week class.error  
2month 4481 768 692 333 0.28578259  
month 1881 3478 508 506 0.45426016  
retained 2422 289 2462 383 0.55687545  
week 105 236 446 7235 0.09810521

정확도 변화 :  
67.33% -> 71.12%  
0.27%p 상승

OOB estimate of error rate: 28.88%  
Confusion matrix:  
2month month retained week class.error  
2month 4527 489 944 314 0.27845075  
month 1853 3422 622 476 0.46304723  
retained 1514 306 3390 346 0.38984881  
week 87 209 415 7311 0.08863126

## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

### K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

각각의 group 마다의 Payment 데이터의 특징을 반영해서 새로운 변수를 추가한다.

#### group1

8주 모두 활동 데이터가 있어서 다른 그룹과는 달리 활동 주 외에 결제와 관련된 변수는 넣지 않음. 새로운 변수 5개 추가

변수	설명
Max_in_pay	활동주차에 포함된 주에 결제한 최대 금액
Min_in_pay	활동주차에 포함된 주에 결제한 최소 금액
mean_in_pay	활동주차에 포함된 주에 결제한 평균 금액
Maxmin_in_pay	$\text{max\_in\_pay} - \text{min\_in\_pay}$
Pay_in	활동주차에 포함된 주에 결제한 횟수

#### group2 & group3 & group4

활동 주 외에도 결제한 유저가 있어 그것을 확인할 수 있는 활동 주 외에 결제와 관련된 변수를 추가했다. group1의 새로운 변수들과 함께 5개의 새로운 변수를 추가로 생성.

변수	설명
Max_out_pay	활동주차 외에 포함된 주에 결제한 최대 금액
Min_out_pay	활동주차 외에 포함된 주에 결제한 최소 금액
mean_out_pay	활동주차 외에 포함된 주에 결제한 평균 금액
Maxmin_out_pay	$\text{max\_in\_pay} - \text{min\_in\_pay}$
Pay_out	활동주차 외에 포함된 주에 결제한 횟수

## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

RandomForest 모델을 사용하여 각 집단별 이탈여부를 예측하고 전체 종합도를 살펴본 결과 72.65%로 기존 모델보다 약 0.37%p 상승했다.

전체 정확도 : 72.28% -> 72.56% (0.37%p 상승)

기  
존



신  
규

group1

OOB estimate of error rate: 19.13%  
Confusion matrix:  
2month month retained week class.error  
2month 1546 170 1722 15 0.55227339  
month 380 2746 889 97 0.33219844  
retained 522 60 14217 24 0.04088241  
week 21 174 527 935 0.43572722

정확도 변화 :  
80.87% -> 80.91%  
0.04%p 상승

OOB estimate of error rate: 19.09%  
Confusion matrix:  
2month month retained week class.error  
2month 1570 176 1696 11 0.54532291  
month 386 2753 877 96 0.33049611  
retained 544 55 14196 28 0.04229913  
week 22 171 528 936 0.43512372

group2

OOB estimate of error rate: 32.48%  
Confusion matrix:  
2month month retained week class.error  
2month 5958 3308 58 531 0.39543379  
month 3693 5731 51 1449 0.47537532  
retained 640 298 242 343 0.84110309  
week 59 551 38 10975 0.05575153

정확도 변화 :  
67.52% -> 67.36%  
0.16%p 하락

OOB estimate of error rate: 32.64%  
Confusion matrix:  
2month month retained week class.error  
2month 5959 3303 49 544 0.39533232  
month 3675 5619 46 1584 0.48562798  
retained 630 295 244 354 0.83978989  
week 54 502 37 11030 0.05101953

group3

OOB estimate of error rate: 26.11%  
Confusion matrix:  
2month month retained week class.error  
2month 4824 117 363 114 0.1096346  
month 1727 1369 246 249 0.6187691  
retained 651 77 2229 141 0.2805036  
week 49 213 179 3257 0.1192537

정확도 변화 :  
73.89% -> 74.00%  
0.11%p 상승

OOB estimate of error rate: 26%  
Confusion matrix:  
2month month retained week class.error  
2month 4790 123 381 124 0.1159099  
month 1701 1402 233 255 0.6095795  
retained 624 84 2250 140 0.2737250  
week 50 208 186 3254 0.1200649

group4

OOB estimate of error rate: 28.88%  
Confusion matrix:  
2month month retained week class.error  
2month 4527 489 944 314 0.27845075  
month 1853 3422 622 476 0.46304723  
retained 1514 306 3390 346 0.38984881  
week 87 209 415 7311 0.08863126

정확도 변화 :  
71.12% -> 71.10%  
0.02%p 하락

OOB estimate of error rate: 28.9%  
Confusion matrix:  
2month month retained week class.error  
2month 4528 495 950 301 0.27829136  
month 1859 3424 608 482 0.46273341  
retained 1519 308 3393 336 0.38930886  
week 87 208 426 7301 0.08987784

## 2. 데이터 분석 과정 \_ 2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

각각의 group 마다의 Guild 데이터의 특징을 반영해서 새로운 변수를 추가한다.

group1 & group2 & group3 & group4

전체 그룹에 4개의 데이터를 추가

변수	설명
Character_n	유저당 길드에 가입된 캐릭터 수
Max_guild_n	유저당 가입된 길드 중 가장 큰 인원 수
Min_guild_n	유저당 가입된 길드 중 가장 작은 인원 수
Guild_chat_n	유저의 평균 길드채팅횟수/ 길드에 가입된 캐릭터 수



## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

RandomForest 모델을 사용하여 각 집단별 이탈여부를 예측하고 전체 종합도를 살펴본 결과 72.67%로 기존 모델보다 약 0.02%p 상승했다.

전체 정확도 : 72.65% -> 72.67% (0.02%p 상승)

기  
존



신  
규

group1

OOB estimate of error rate: 19.09%  
Confusion matrix:  
2month month retained week class.error  
2month 1570 176 1696 11 0.54532291  
month 386 2753 877 96 0.33049611  
retained 544 55 14196 28 0.04229913  
week 22 171 528 936 0.43512372

정확도 변화 :  
80.91% -> 80.88%  
0.03%p 하락

OOB estimate of error rate: 19.12%  
Confusion matrix:  
2month month retained week class.error  
2month 1563 176 1697 17 0.54735013  
month 377 2752 882 101 0.33073930  
retained 539 61 14199 24 0.04209674  
week 20 174 530 933 0.43693422

group2

OOB estimate of error rate: 32.64%  
Confusion matrix:  
2month month retained week class.error  
2month 5959 3303 49 544 0.39533232  
month 3675 5619 46 1584 0.48562798  
retained 630 295 244 354 0.83978989  
week 54 502 37 11030 0.05101953

정확도 변화 :  
67.36% -> 67.48%  
0.12%p 상승

OOB estimate of error rate: 32.52%  
Confusion matrix:  
2month month retained week class.error  
2month 5959 3305 46 545 0.3953323  
month 3680 5645 52 1547 0.4832479  
retained 631 287 248 357 0.8371635  
week 57 487 40 11039 0.0502452

group3

OOB estimate of error rate: 26%  
Confusion matrix:  
2month month retained week class.error  
2month 4790 123 381 124 0.1159099  
month 1701 1402 233 255 0.6095795  
retained 624 84 2250 140 0.2737250  
week 50 208 186 3254 0.1200649

정확도 변화 :  
74.00% -> 73.99%  
0.01%p 하락

OOB estimate of error rate: 26.01%  
Confusion matrix:  
2month month retained week class.error  
2month 4817 100 381 120 0.1109265  
month 1740 1371 231 249 0.6182122  
retained 623 92 2252 131 0.2730794  
week 44 222 178 3254 0.1200649

group4

OOB estimate of error rate: 28.9%  
Confusion matrix:  
2month month retained week class.error  
2month 4528 495 950 301 0.27829136  
month 1859 3424 608 482 0.46273341  
retained 1519 308 3393 336 0.38930886  
week 87 208 426 7301 0.08987784

정확도 변화 :  
71.10% -> 71.05%  
0.02%p 하락

OOB estimate of error rate: 28.95%  
Confusion matrix:  
2month month retained week class.error  
2month 4508 502 955 309 0.28147912  
month 1836 3431 628 478 0.46163502  
retained 1516 306 3394 340 0.38912887  
week 89 213 419 7301 0.08987784



## 2. 데이터 분석 과정 \_ 2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

각각의 group 마다의 Trade 데이터의 특징을 반영해서 새로운 변수를 추가한다.

group1 & group2 & group3 & group4

새로운 변수 54개 추가

주체	변수	설명
판매자 (source)	Week1~week8	해당 주 판매 횟수/전체 판매 횟수
	Day1~day7	해당 요일 판매 횟수/전체 판매 횟수
	Earlymorning~night	해당 시간 판매 횟수/전체 판매 횟수
	묶음 변수6개(품목)	해당 품목 판매 횟수/전체 판매 횟수
구매자 (target)	Week1~week8	해당 주 판매 횟수/전체 판매 횟수
	Day1~day7	해당 요일 판매 횟수/전체 판매 횟수
	Earlymorning~night	해당 시간 판매 횟수/전체 판매 횟수
	묶음 변수6개(품목)	해당 품목 판매 횟수/전체 판매 횟수

## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

RandomForest 모델을 사용하여 각 집단별 이탈여부를 예측하고 전체 종합도를 살펴본 결과 72.84%로 기존 모델보다 약 0.17%p 상승했다.

전체 정확도 : 72.67% -> 72.84% (0.17%p 상승)

기  
존



신  
규

group1

OOB estimate of error rate: 19.12%  
Confusion matrix:  
2month month retained week class.error  
2month 1563 176 1697 17 0.54735013  
month 377 2752 882 101 0.33073930  
retained 539 61 14199 24 0.04209674  
week 20 174 530 933 0.43693422

정확도 변화 :  
80.88% -> 81.24%  
0.36%p 상승

OOB estimate of error rate: 18.76%  
Confusion matrix:  
2month month retained week class.error  
2month 1584 170 1691 8 0.54126846  
month 380 2787 880 65 0.32222763  
retained 545 54 14204 20 0.04175943  
week 16 150 532 959 0.42124321

group2

OOB estimate of error rate: 32.52%  
Confusion matrix:  
2month month retained week class.error  
2month 5959 3305 46 545 0.3953323  
month 3680 5645 52 1547 0.4832479  
retained 631 287 248 357 0.8371635  
week 57 487 40 11039 0.0502452

정확도 변화 :  
67.48% -> 67.56%  
0.08%p 상승

OOB estimate of error rate: 32.44%  
Confusion matrix:  
2month month retained week class.error  
2month 5961 3296 48 550 0.39512938  
month 3694 5707 45 1478 0.47757232  
retained 628 279 236 380 0.84504268  
week 53 519 36 11015 0.05231007

group3

OOB estimate of error rate: 26.01%  
Confusion matrix:  
2month month retained week class.error  
2month 4817 100 381 120 0.1109265  
month 1740 1371 231 249 0.6182122  
retained 623 92 2252 131 0.2730794  
week 44 222 178 3254 0.1200649

정확도 변화 :  
73.99% -> 73.93%  
0.06%p 하락

OOB estimate of error rate: 26.07%  
Confusion matrix:  
2month month retained week class.error  
2month 4840 97 377 104 0.1066814  
month 1765 1341 249 236 0.6265664  
retained 638 77 2240 143 0.2769529  
week 42 191 201 3264 0.1173607

group4

OOB estimate of error rate: 28.95%  
Confusion matrix:  
2month month retained week class.error  
2month 4508 502 955 309 0.28147912  
month 1836 3431 628 478 0.46163502  
retained 1516 306 3394 340 0.38912887  
week 89 213 419 7301 0.08987784

정확도 변화 :  
71.05% -> 71.32%  
0.27%p 상승

OOB estimate of error rate: 28.68%  
Confusion matrix:  
2month month retained week class.error  
2month 4513 497 957 307 0.28068218  
month 1870 3427 609 467 0.46226267  
retained 1491 294 3434 337 0.38192945  
week 77 183 433 7329 0.08638743

## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

### K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

RandomForest 모델의 메소드 중 importance 함수를 이용해 중요도를 떨어뜨리는 변수들을 제거했다.

그룹	제거변수
Group1	source_accessory, source_costume, source_gem, source_weapon, target_accessory, target_costume, target_gem, target_weapon, mean_cnt_enter_bam, mean_cnt_clear_bam, mean_faction_chat, min_in_pay, mean_gathering_cnt
Group2	mean_cnt_enter_bam, mean_faction_chat, min_out_pay, target_weapon, source_accessory, target_accessory, source_weapon, source_gem
Group3	mean_faction_chat, mean_cnt_clear_bam, mean_gathering_cnt, min_in_pay, max_out_pay, min_out_pay, mean_output_pay, maxmin_out_pay, source_accessory, source_costume, source_gem, source_weapon, target_accessory, target_costume, target_weapon, target_gem
Group4	source_gem, min_in_pay, source_costume, target_costume, pay_out_n, mean_faction_chat, mean_gathering_cnt, source_weapon, target_accessory, max_out_pay, mean_output_pay, maxmin_out_pay

## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

중요도를 떨어뜨리는 변수들을 제거하고 모델링을 통해 전체 종합도를 살펴본 결과 73.60%로 기존 모델보다 약 0.76%p 상승했다.

전체 정확도 : 72.84% -> 73.60% (0.76%p 상승)

기  
존



신  
규

group1

OOB estimate of error rate: 18.76%  
Confusion matrix:  
2month month retained week class.error  
2month 1584 170 1691 8 0.54126846  
month 380 2787 880 65 0.32222763  
retained 545 54 14204 20 0.04175943  
week 16 150 532 959 0.42124321

정확도 변화 :  
81.24% -> 82.00%  
0.76%p 상승

OOB estimate of error rate: 18%  
Confusion matrix:  
2month month retained week class.error  
2month 1617 181 1641 14 0.53171156  
month 357 2815 867 73 0.31541829  
retained 421 58 14325 19 0.03359644  
week 20 142 534 961 0.42003621

group2

OOB estimate of error rate: 32.44%  
Confusion matrix:  
2month month retained week class.error  
2month 5961 3296 48 550 0.39512938  
month 3694 5707 45 1478 0.47757232  
retained 628 279 236 380 0.84504268  
week 53 519 36 11015 0.05231007

정확도 변화 :  
67.56% -> 68.05%  
0.49%p 상승

OOB estimate of error rate: 31.95%  
Confusion matrix:  
2month month retained week class.error  
2month 5908 3385 58 504 0.40050736  
month 3549 6049 75 1251 0.44626510  
retained 569 367 261 326 0.82862771  
week 77 624 53 10869 0.06487138

group3

OOB estimate of error rate: 26.07%  
Confusion matrix:  
2month month retained week class.error  
2month 4840 97 377 104 0.1066814  
month 1765 1341 249 236 0.6265664  
retained 638 77 2240 143 0.2769529  
week 42 191 201 3264 0.1173607

정확도 변화 :  
73.93% -> 75.06%  
1.13%p 하락

OOB estimate of error rate: 24.94%  
Confusion matrix:  
2month month retained week class.error  
2month 4426 418 468 106 0.1830934  
month 1265 1808 286 232 0.4965191  
retained 479 137 2345 137 0.2430600  
week 38 188 188 3284 0.1119524

group4

OOB estimate of error rate: 28.68%  
Confusion matrix:  
2month month retained week class.error  
2month 4513 497 957 307 0.28068218  
month 1870 3427 609 467 0.46226267  
retained 1491 294 3434 337 0.38192945  
week 77 183 433 7329 0.08638743

정확도 변화 :  
71.32% -> 72.19%  
0.87%p 상승

OOB estimate of error rate: 27.81%  
Confusion matrix:  
2month month retained week class.error  
2month 4229 704 1053 288 0.32594836  
month 1485 3719 711 458 0.41644437  
retained 1170 387 3688 311 0.33621310  
week 74 201 450 7297 0.09037646



## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

K-means : Activity 데이터를 통한 10만개 유저들의 데이터 군집화

RandomForest모델링을 사용하여 전체 종합도를 살펴본 결과 74.89%로 기존 모델보다 약 1.29%p 상승했다.

전체 정확도 : 73.60% -> 74.89% (1.29%p 상승)

기  
존



신  
규

group1

OOB estimate of error rate: 18%  
Confusion matrix:  
2month month retained week class.error  
2month 1617 181 1641 14 0.53171156  
month 357 2815 867 73 0.31541829  
retained 421 58 14325 19 0.03359644  
week 20 142 534 961 0.42003621

정확도 변화 :  
82.00% -> 82.15%  
0.15%p 상승

OOB estimate of error rate: 17.85%  
Confusion matrix:  
2month month retained week class.error  
2month 1627 156 1656 14 0.52881552  
month 386 2797 856 73 0.31979572  
retained 475 41 14297 10 0.03548539  
week 24 130 472 1031 0.37779119

group2

OOB estimate of error rate: 31.95%  
Confusion matrix:  
2month month retained week class.error  
2month 5908 3385 58 504 0.40050736  
month 3549 6049 75 1251 0.44626510  
retained 569 367 261 326 0.82862771  
week 77 624 53 10869 0.06487138

정확도 변화 :  
68.05% -> 71.38%  
3.33%p 상승

OOB estimate of error rate: 28.62%  
Confusion matrix:  
2month month retained week class.error  
2month 6052 3421 64 318 0.38589548  
month 3639 6547 35 703 0.40067741  
retained 650 322 316 235 0.79251477  
week 43 267 11 11302 0.02761765

group3

OOB estimate of error rate: 24.94%  
Confusion matrix:  
2month month retained week class.error  
2month 4426 418 468 106 0.1830934  
month 1265 1808 286 232 0.4965191  
retained 479 137 2345 137 0.2430600  
week 38 188 188 3284 0.1119524

정확도 변화 :  
75.06% -> 75.62%  
0.56%p 하락

OOB estimate of error rate: 24.38%  
Confusion matrix:  
2month month retained week class.error  
2month 4798 164 356 100 0.1144334  
month 1696 1443 247 205 0.5981621  
retained 584 95 2322 97 0.2504842  
week 26 131 153 3388 0.0838291

group4

OOB estimate of error rate: 27.81%  
Confusion matrix:  
2month month retained week class.error  
2month 4229 704 1053 288 0.32594836  
month 1485 3719 711 458 0.41644437  
retained 1170 387 3688 311 0.33621310  
week 74 201 450 7297 0.09037646

정확도 변화 :  
72.19% -> 72.33%  
0.14%p 상승

OOB estimate of error rate: 27.67%  
Confusion matrix:  
2month month retained week class.error  
2month 4511 481 998 284 0.28100096  
month 1864 3458 602 449 0.45739840  
retained 1424 290 3558 284 0.35961123  
week 60 154 367 7441 0.07242583

## 2. 데이터 분석 과정 \_2-1 변수 선택 과정

Activity 데이터 중에서 변수별 Max, Min, Diff(Max-Min), (Max-mean)/mean 값들을 전진 제거법을 통해 중요 변수 생성

Activity만 사용하여 XGBoost 모델로 정확도를 측정한 결과 71.155%로 나왔다. Activity 변수 내에서 모델의 정확도를 올려줄 수 있는 추가적인 변수가 있을지 알아보기 위해서 변수의 max, min, diff, (max-mean)/mean 값을 전진 제거법을 통해 알아보았다. 총 106번의 분석결과 아래 표에 나와있는 변수들이 모형의 정확도를 올려줄 수 있다는 것을 확인할 수 있었다.

model	dataset	ratio	feature	accuracy
XGBClassifier	train_activity	8:2(train:test)	37(mean)	<b>0.71155</b>

n\_estimators=500, max\_depth=6, seed=37

base\_score=0.5, booster='gbtree', colsample\_bylevel=1, colsample\_bytree=1, gamma=0, learning\_rate=0.1, max\_delta\_step=0, max\_depth=6, min\_child\_weight=1, missing=None, n\_estimators=500, n\_jobs=1, nthread=None, objective='multi:softprob', random\_state=0, reg\_alpha=0, reg\_lambda=1, scale\_pos\_weight=1, seed=37, silent=True, subsample=1

Activity	+0.415 %	+0.3 %	+0.095 %	+0.04 %	+0.04 %	+0.04 %
<b>play_time</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> <li>Diff(Max-Min)</li> <li>(Max-mean)/mean</li> </ul>	<b>cnt_enter_inzone_normal</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> <li>Diff(Max-Min)</li> </ul>	<b>quest_hongmun</b> <ul style="list-style-type: none"> <li>(Max-mean)/mean</li> </ul>	<b>cnt_dt</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> <li>Diff(max-min)</li> </ul>	<b>gathering_cnt</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> <li>Diff(max-min)</li> </ul>	<b>cnt_enter_inzone_skilled</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> </ul>	
	<b>party_battle_cnt</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> </ul>	<b>get_money</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> </ul>	<b>factions_chat</b> <ul style="list-style-type: none"> <li>Max</li> <li>Min</li> <li>diff(max-min)</li> </ul>	<b>ETC : +0.005% ~ 0.035 %</b>		
				<ul style="list-style-type: none"> <li>party_chat</li> <li>npc_hongmun</li> <li>cnt_enter_inzone_light</li> <li>get_money</li> </ul>	<ul style="list-style-type: none"> <li>guild_chat</li> <li>cnt_clear_bam</li> <li>whisper_chat</li> </ul>	

## 2. 데이터 분석 과정 \_2-2. 모델링

최종 모델에 Activity로 만든 변수를 추가하여 모델링을 한 결과 0.737로 약 0.005 감소하여 변수를 사용하지 않았다.

앞서 activity 데이터 변수들의 Max, Min, diff, (max-mean)/mean 변수들의 전진 제거법을 사용하여 추출한 변수들을 추가하여 기존 activity 모델에 적용한 결과 기존 보다 0.58%p가 향상됐다.

model	Dataset	ratio	feature	accuracy
XGBClassifier	Activity + 유의미한 변수별 Max, Min, Diff(Max-Min), (Max-mean)/mean	8:2(train:test)	75(mean, max, min, diff, (max-mean)/mean)	0.7173

n\_estimators=500, max\_depth=6, seed=37  
base\_score=0.5, booster='gbtree', colsample\_bylevel=1, colsample\_bytree=1, gamma=0, learning\_rate=0.1, max\_delta\_step=0, max\_depth=6,  
min\_child\_weight=1, missing=None, n\_estimators=500, n\_jobs=1, nthread=None, objective='multi:softprob', random\_state=0, reg\_alpha=0, reg\_lambda=1,  
scale\_pos\_weight=1, seed=37, silent=True, subsample=1

Activitiy 데이터를 통해 추가한 변수 38개를 최종 모델에 추가하여 XGBoost 모델을 생성했다. 측정 결과 기존 정확도보다 약 -0.005 감소했다. 전체를 예측하는데 있어 유의하지 않는것 같다.

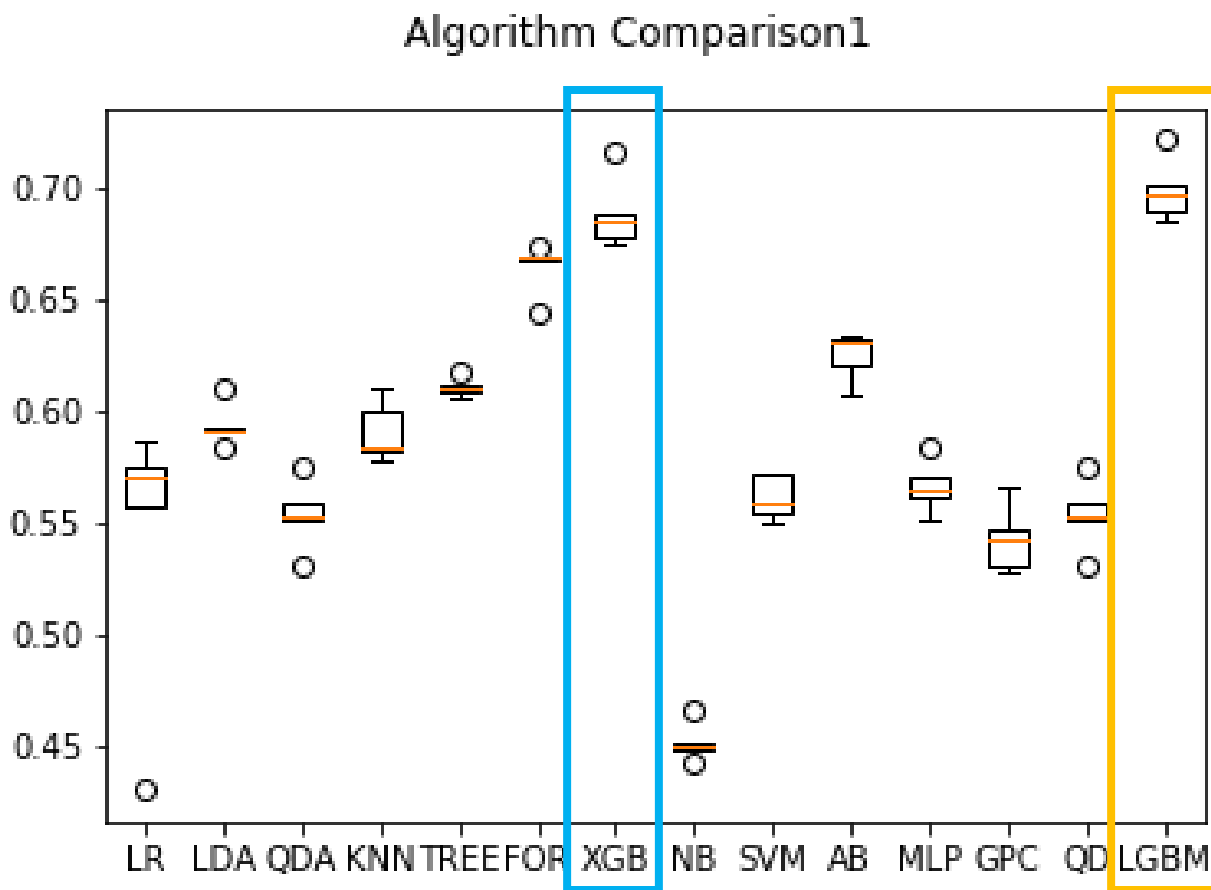
model	Dataset	Ratio	Feature	Accuracy
XGBClassifier	최종(Activity + Guild + Payment + Trade) + 추가(Activity 중 전진 제거법 생성 변수)	8:2(train:test)	544 = 506(최종) + 38(추가)	- 0.005 (0.742 -0.737)

## 2. 데이터 분석 과정 \_2-2. 모델링

선정된 모델 중 가장 F-Score가 높았던 모델 = XGBoost : Classifier

다양한 알고리즘을 적용해보기 위해서 Null 값이 있는 컬럼을 삭제하고, 알고리즘을 적용해 k-Fold (k = 5) Cross Validation을 수행하여 F1-Score 및 score의 std를 계산.

N= 10000, seed = 37 이외의 Parameter는 Default 값으로 설정.



	model	f1_score	std
1	LightGBM	0.699066	0.012458
2	XGBoost	0.688405	0.014499
3	RandomForest	0.664790	0.010277
4	AdaBoost	0.624761	0.010002
5	DecisionTree	0.610866	0.004106
6	LDA	0.593985	0.008802
7	KNN	0.590635	0.012406
8	MLP	0.566195	0.010570
9	SVM	0.561542	0.008900
10	QDA	0.553450	0.014405
11	Logistic	0.543709	0.057415
12	GaussianProcess	0.542672	0.013170
13	GaussianNB	0.451463	0.007699



## 2. 데이터 분석 과정 \_2-2. 모델링

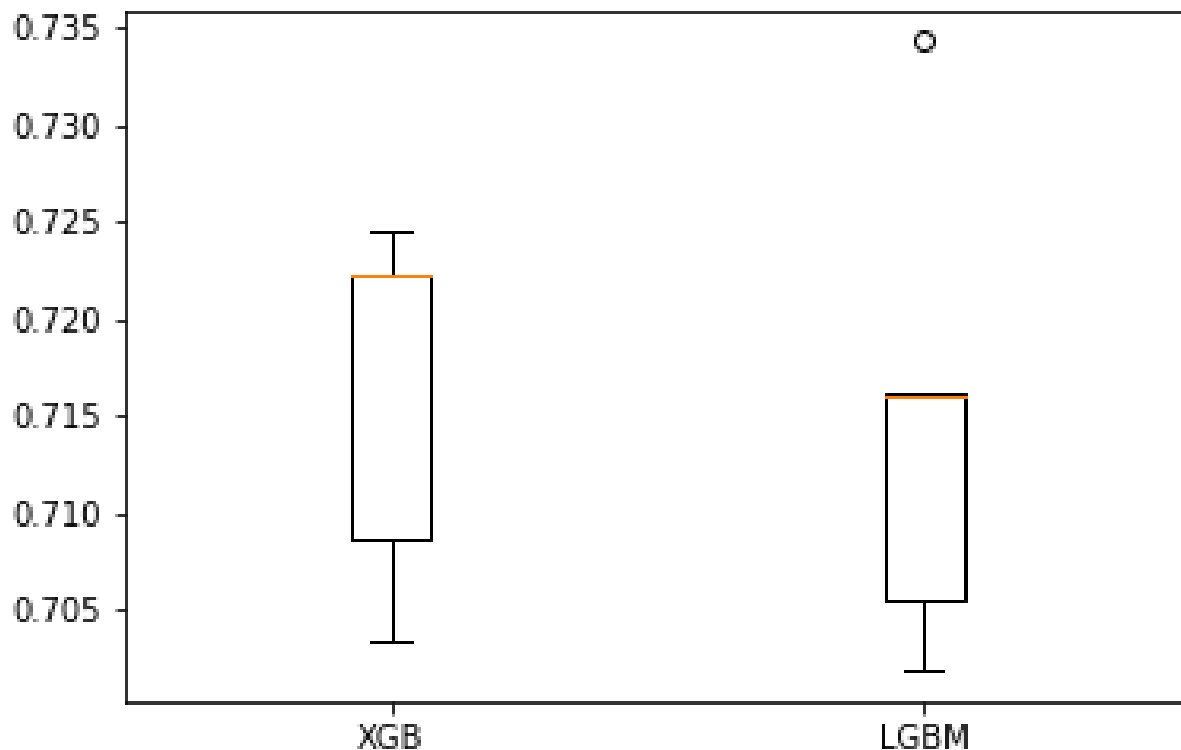
선정된 모델 중 가장 F-Score가 높았던 모델 = XGBoost : Classifier

F1-score가 가장 높았던 XGB와 LGBM 두 모델을 선택해 재비교.

N= 10000, n\_estimators = 100, max\_depth = 10

XGB 모델을 선택

Algorithm Comparison



	model	f1_score	std
1	XGB	0.716215	0.008528
2	LGBM	0.714794	0.011287

## 2. 데이터 분석 과정 \_2-2. 모델링

선정된 모델 중 가장 F-Score가 높았던 모델 = XGBoost : Classifier

데이터를 세분화하여 각각의 모델을 만들어 예측 해보고, 다양한 변수를 생성 혹은 축약하여 만든 모델들 보다, 기존의 변수들(party 제외)을 가지고 만든 모델의 F-Score 이 가장 성능이 우수했다.



Train Data  
(8:2)

### XGBoost : Classifier

- 투입된 변수: 총 506개 (Party 제외)
  - activity : 326개(64.2%)
  - guild : 2개(0.4%)
  - payment : 22개(4.3%)
  - trade : 156개(30.8%)
- 파라미터 설정
  - n\_estimators = 1,000
  - max\_depth = 10
  - seed = 37

## 2. 데이터 분석 과정 \_2-2. 모델링

### 선정된 모델 중 가장 F-Score가 높았던 모델 = XGBoost : Classifier

데이터를 세분화하여 각각의 모델을 만들어 예측 해보고, 다양한 변수를 생성 혹은 축약하여 만든 모델들 보다, 기존의 변수들(party 제외)을 가지고 만든 모델의 F-Score 이 가장 성능이 우수했다.

model	Dataset	ratio	feature	accuracy
XGBClassifier	Activity + Guild + Payment + Trade	8:2(train:test)	506(mean)	0.742

n\_estimators=1,000, max\_depth=10, seed=37

base\_score=0.5, booster='gbtree', colsample\_bylevel=1, colsample\_bytree=1, gamma=0, learning\_rate=0.1, max\_delta\_step=0, max\_depth=10, min\_child\_weight=1, missing=None, n\_estimators=1,000, n\_jobs=1, nthread=None, objective='multi:softprob', random\_state=0, reg\_alpha=0, reg\_lambda=1, scale\_pos\_weight=1, seed=37, silent=True, subsample=1

위 모델을 통해 Test data를 예측한 결과 홈페이지 내에서 73.23% 결과를 얻을 수 있었다.

2018 빅콘테스트 2018 BIG CONTEST				
순위표				
순위	ID	마지막 제출 일자	최고 점수	
	hang1***@naver.com	2018-08-13 01:15:50 AM	0.7388	
	soso0***@naver.com	2018-08-13 11:48:46 PM	0.7323	



## Category 03

### 부록:데이터 탐색

3-1 \_ Activity

3-2 \_ Guild

3-3 \_ Trade

3-4 \_ Payment

3-5 \_ Party

## Category 03

부록: 데이터 탐색

### 3-1 \_ Activity Data

3-2 \_ Guild Data

3-3 \_ Trade Data

3-4 \_ Payment Data

3-5 \_ Party Data





### 3. 부록: 데이터 탐색 \_ 3-1. Activity

Activity : 유저의 인 게임 활동 정보를 일주일 단위로 집계한 자료

게임 활동 정보를 집계한 자료는 총 38개의 변수와 440,323개의 인스턴스(row)로 구성되어 있었다. 서로 연관성이 있는 변수들을 묶어 이탈여부와 세부적으로 살펴보고자 한다.

#### A-1. EXP : 경험치

홍문과 일반 경험치의 대조와  
더불어 유저의 성장 활동을 볼 수  
있는 변수, 버프템과 관련이 크다



[EXP:경험치]

npc\_exp, npc\_hongmun  
quest\_exp, quest\_hongmun  
item\_hongmun, cnt\_use\_buffitem

#### A-2. INZONE : 인던

던전의 난이도에 따른  
유저들의 활동을  
비교해볼 수 있다.



[INZONE:인던]

cnt\_enter\_inzone\_solo, light, skilled, normal  
cnt\_cleaer\_inzone\_solo, light, skilled, normal

#### A-3. CHAT : 채팅

유저들의 사회활동을  
볼 수 있는 변수



[CHAT:채팅]

nomal\_chat, whisper\_chat  
district\_chat, party\_chat  
guild\_chat, faction\_chat

#### A-4. ITEM : 물품

물품과 돈에  
관련된 변수



[ ITEM:물품 ]

get\_money  
gathering\_cnt  
making\_cnt

### 3. 부록: 데이터 탐색 \_ 3-1. Activity

Activity : 유저의 인 게임 활동 정보를 일주일 단위로 집계한 자료

#### A-5. ACT : 활동

유저의 활동량을  
확인해볼 수 있는 변수



[ACT:활동]

cnt\_dt  
play\_time

#### A-6. FIGHT : 싸움

전투와 관련된  
변수들



[FIGHT:싸움]

game\_combat\_time  
duel\_cnt, duel\_win  
partybattle\_cnt, partybattle\_win

#### A-7. RAID : 레이드

다수의 유저 파티인원을  
필요로 하는 전투 콘텐츠  
변수들



[RAID:레이드]

cnt\_enter\_raid, cnt\_enter\_raid\_light  
cnt\_clear\_raid, cnt\_clear\_raid\_light

#### A-8. BAM : 바람평야(인던)

인던의 한 종류로 과거에는  
플레이 유저가 있었으나 현재는 누  
구도 이용하지 않는 콘텐츠



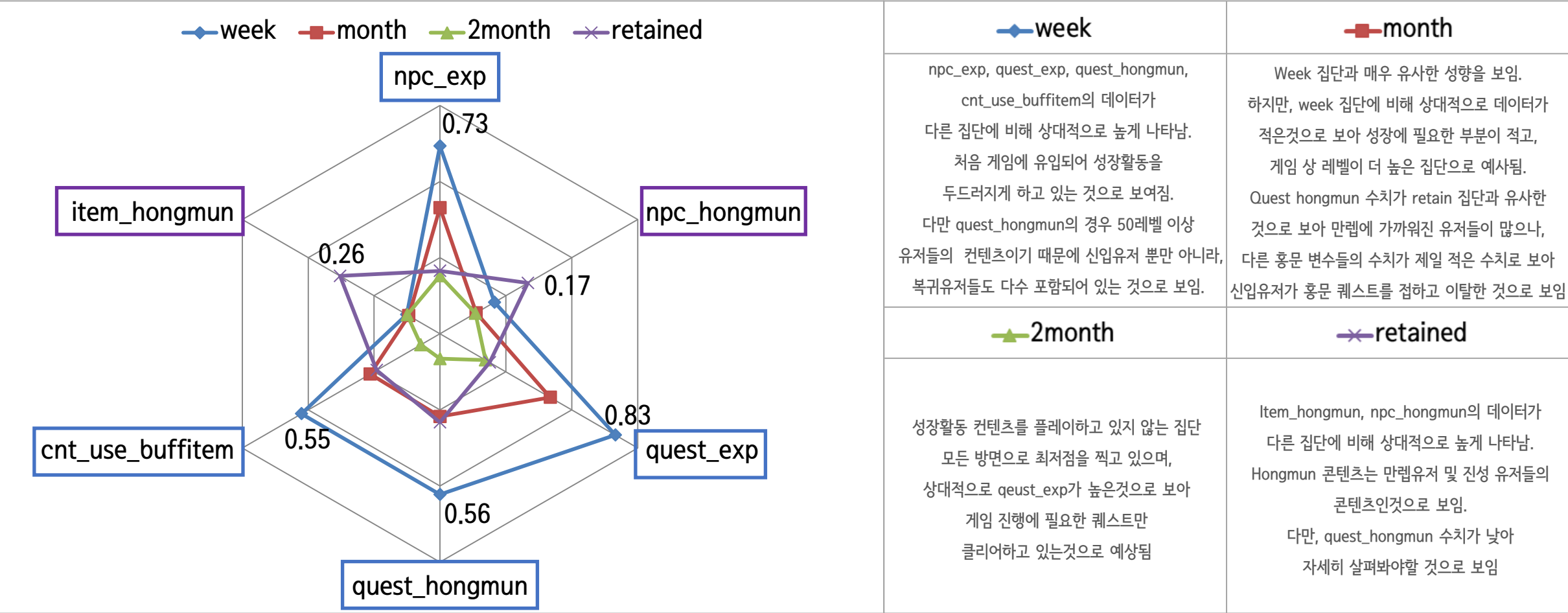
[BAM:바람평야]

cnt\_enter\_bam  
cnt\_clear\_bam

### 3. 부록: 데이터 탐색 \_ 3-1. Activity

#### A-1. EXP : 경험치(집단 간 유저들의 수치를 평균으로 집계)

이탈 여부에 따라 EXP(경험치)를 살펴보면, week집단은 npc경험치, 퀘스트 경험치, 퀘스트 홍문경험치, 버프아이템 사용 수 변수에서(npc\_exp, quest\_exp, quest\_hongmun, cnt\_use\_buffite) 매우 높게 나타났으며, retain집단은 아이템 홍문경험치, npc 홍문경험치에서(item\_hongmun, npc\_hongmun) 두드러지게 높은 것을 살펴볼 수 있음. hongmun의 경우 retain집단의 특성으로 보였으나 quest\_hongmun은 week에서 2배 이상 상대적으로 높은 것을 보아 세부 집단 별 살펴봐야 함

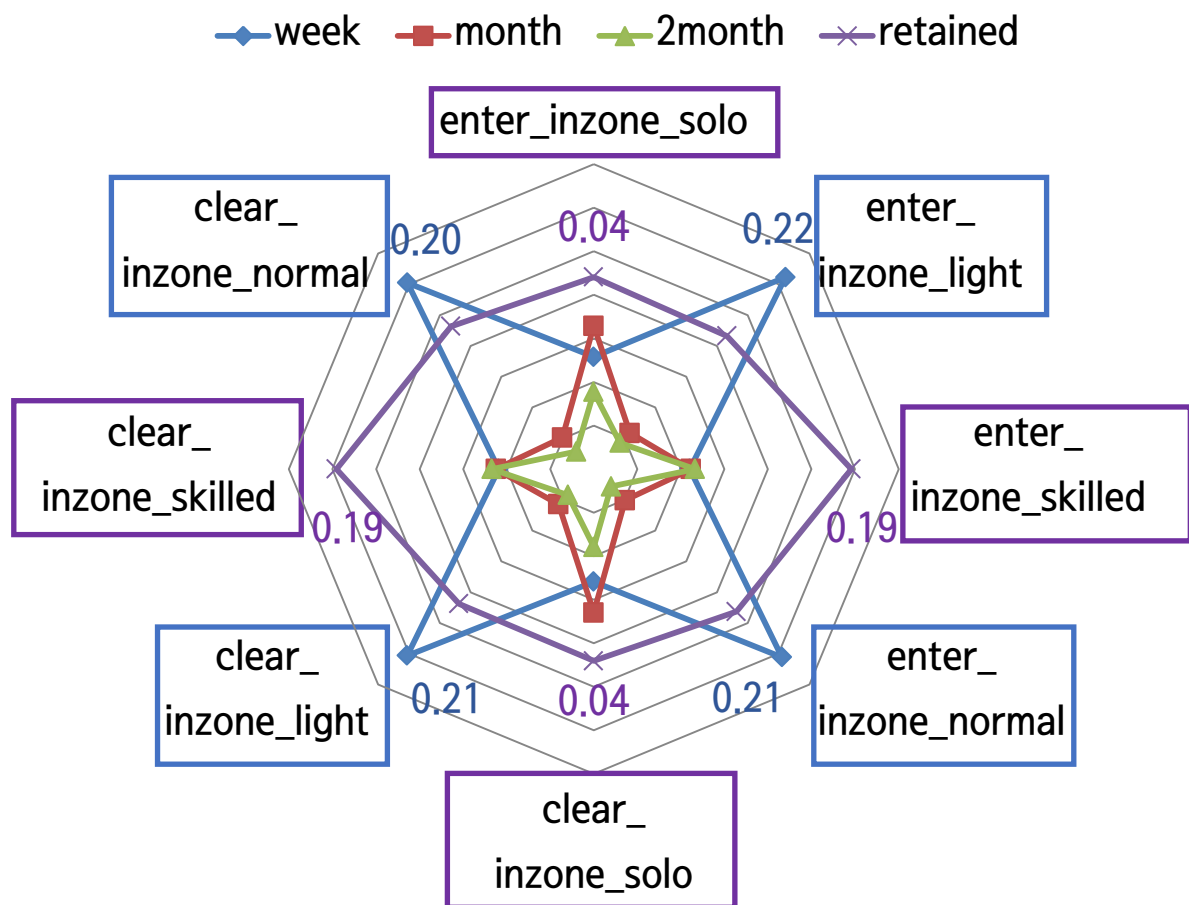




### 3. 부록: 데이터 탐색 \_3-1. Activity

## A-2. INZONE : 인던(집단 간 유저들의 수치를 평균으로 집계)

인스턴트 던전 플레이 데이터를 살펴보면, week집단은 노말인던, 라이트인던(inzone\_normal, inzone\_light) 변수가 높게 나타났으며, 숙련인던, 솔로인던(inzone\_skilled, inzone\_solo) 변수는 상대적으로 낮게 나타남. retain집단은 이와 반대로 숙련인던과 솔로인던 수치가 높은 반면, 노말과 라이트 인던 수치는 상대적으로 낮게 나타남.

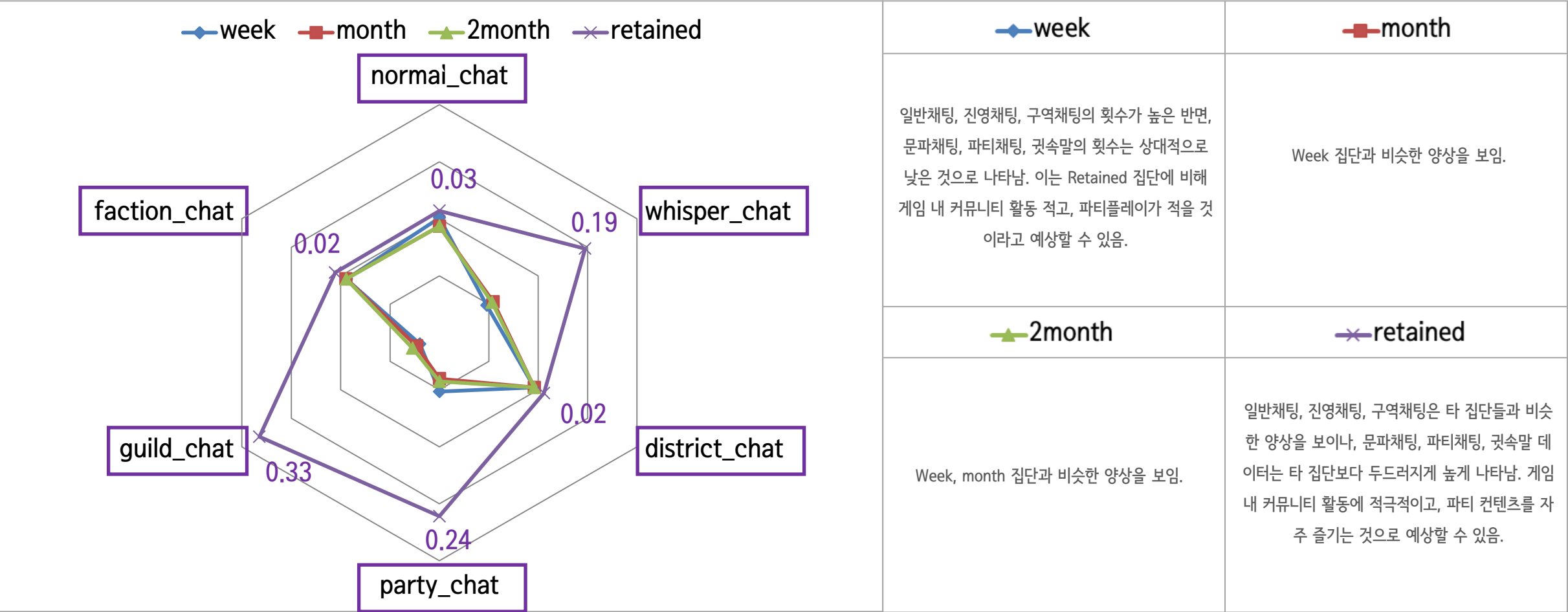


 week	 month
<p>노말인던, 라이트인던(inzone_normal, inzone_light )의 데이터가 다른 집단에 비해 상대적으로 높게 나타난 반면, 숙련인던(inzone_skilled)은 낮게 나타남.</p> <p>이를 통해 상대적으로 난이도가 쉬운 콘텐츠를 주로 즐기며, 솔로인던(inzone_solo)이 낮은것으로 보아, 주로 파티플레이를 즐기는 것으로 예상됨.</p>	<p>모든 인스턴트 던전 플레이 데이터가 week 또는 retained에 비해 낮음. 인스턴트 던전 플레이 자체를 잘 하지 않는것으로 보임. 특히 난이도가 쉬운 콘텐츠인 노말, 라이트 인스턴트 던전은 거의 즐기지 않는 것으로 보임</p>
 2month	 retained
<p>Month와 비슷한 양상을 보임. 4개의 집단 중예가 가장 인스턴트 던전을 즐기지 않는 것으로 나타남.</p>	<p>인스턴트 던전 플레이를 적극적으로 즐김.</p> <p>솔로인던(inzone_solo )데이터가 높게 나타난 것으로 보아, 혼자서 던전을 클리어 할 수 있는 숙련 유저가 많은 것으로 예상됨. inzone_skilled 데이터가 높은 것으로 보아 난이도가 높은 콘텐츠를 즐기는 유저가 많은 집단으로 예상됨.</p>

### 3. 부록: 데이터 탐색 \_ 3-1. Activity

#### A-3. CHAT : 채팅(집단 간 유저들의 수치를 평균으로 집계)

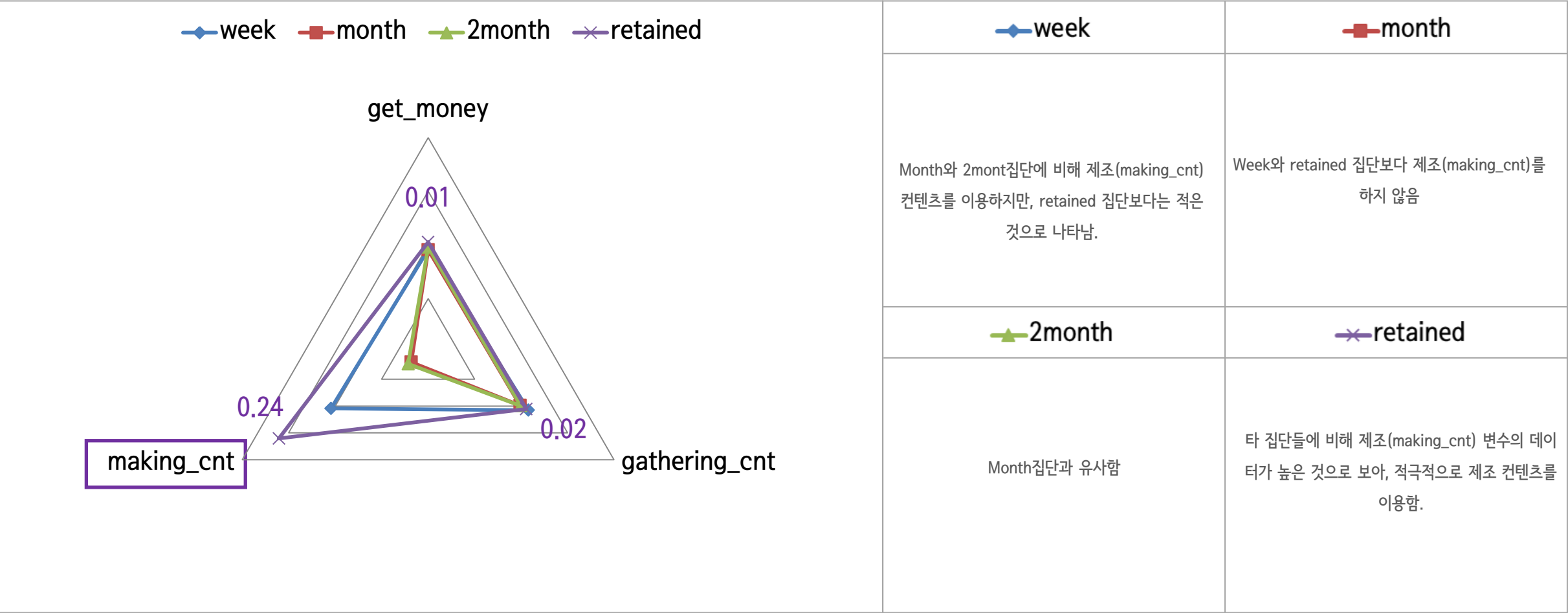
채팅 데이터에서는 Week, month, 2month 집단에서는 집단간 유의미한 차이를 발견하기 어려움. 세 집단 모두 일반채팅(normal\_chat)과 진영채팅(faction\_chat), 구역채팅(district\_chat)은 데이터는 높은 반면, 문파(guild\_chat)와 파티(party\_chat), 귓속말(whisper\_chat) 데이터는 낮게 나타남. Retained 집단은 다른 집단들보다 채팅을 많이 하는 것으로 보이며 특히 문파 채팅, 파티채팅, 귓속말 횟수가 타 집단보다 두드러지게 높은것으로 나타남.



### 3. 부록: 데이터 탐색 \_ 3-1. Activity

#### A-4. ITEM : 물품(집단 간 유저들의 수치를 평균으로 집계)

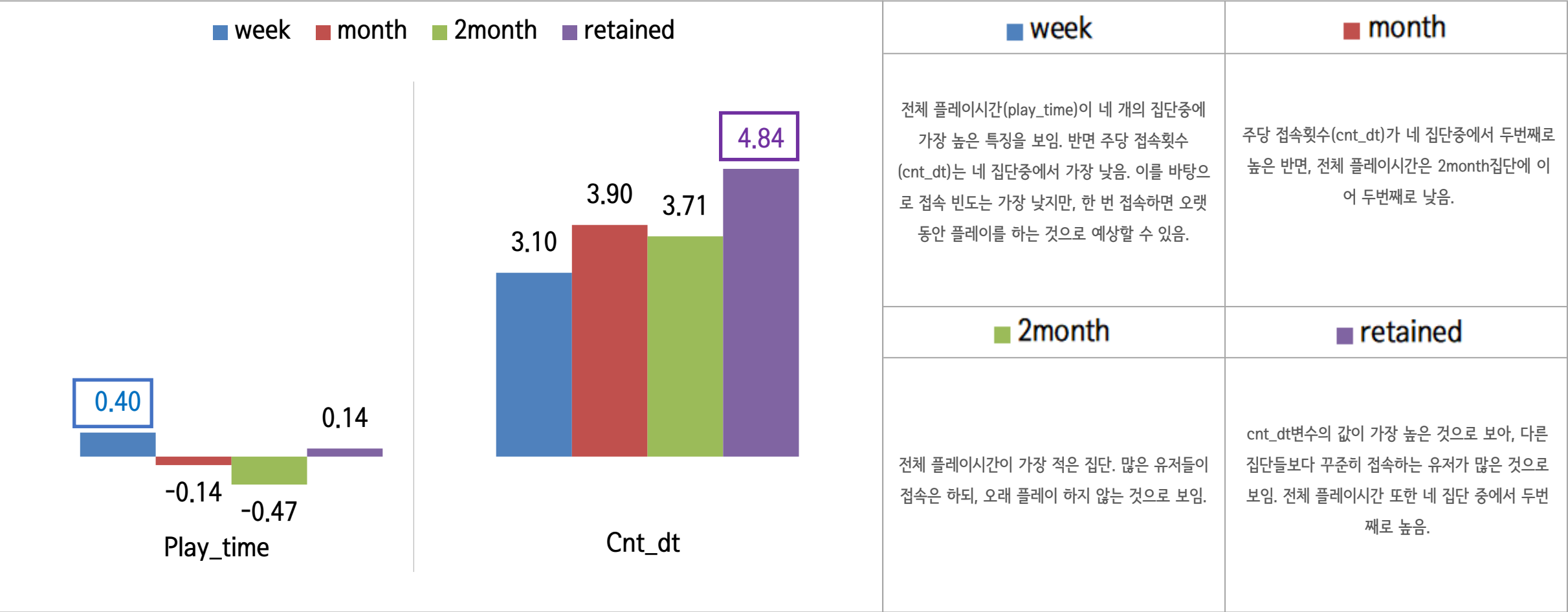
재화획득(get\_money), 채집횟수(gathering\_cnt) 변수에서는 집단간의 유의미한 차이를 발견하기 힘들. 제조(making\_cnt)횟수는 집단간 차이를 발견할 수 있으며, retained 집단이 가장 제조를 많이 하는 것으로 나타났고, week집단은 그 다음으로 제조를 많이 하는 것으로 나타남. Month와 2month집단은 거의 제조를 하지 않는 것으로 보임.



### 3. 부록: 데이터 탐색 \_ 3-1. Activity

#### A-5. ACT : 활동(집단 간 유저들의 수치를 평균으로 집계)

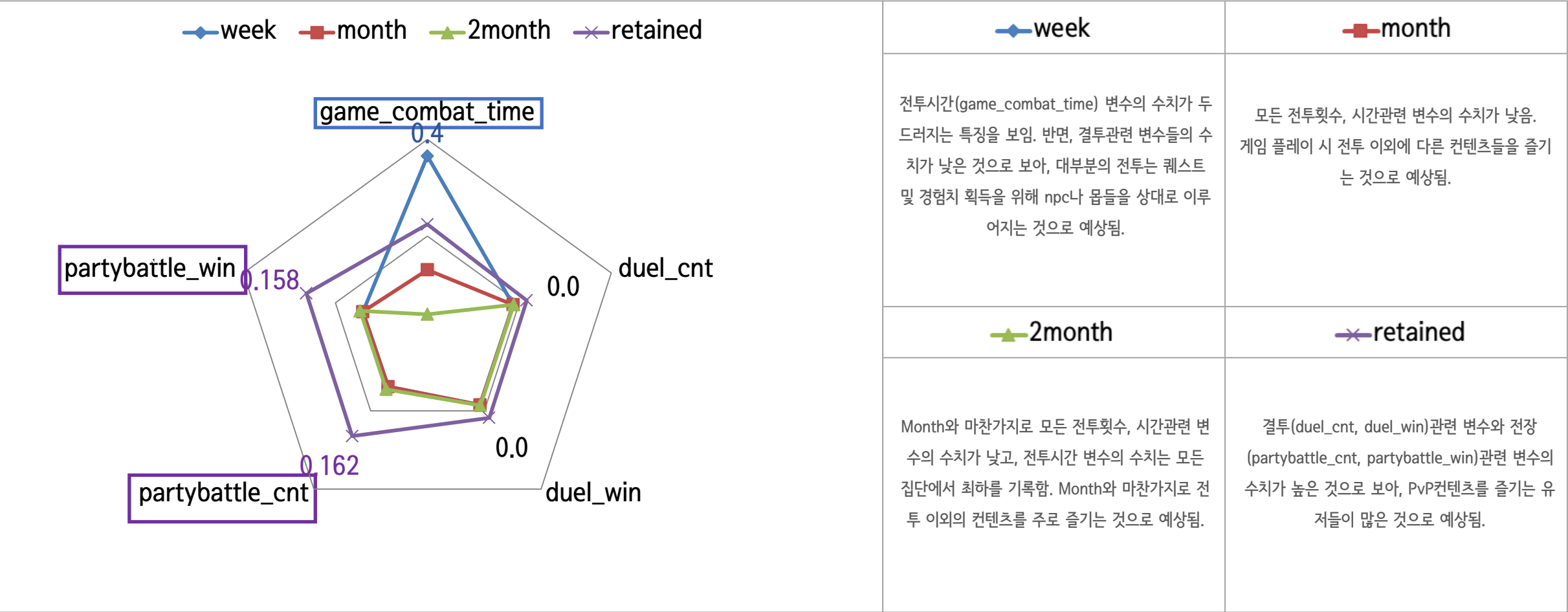
Week집단은 전체 플레이 시간(play\_time)은 가장 높은 반면, 주당 평균 접속횟수(Cnt\_dt)는 가장 낮은 특징을 보임. Retained 집단은 주당 평균 접속횟수가 가장 많고, 플레이시간 또한 전체집단의 평균보다 높음. Month와 2month 집단은 플레이시간이 전체집단의 평균보다 낮은 반면, 주당 접속횟수는 week집단보다 높은 특징을 보임.



### 3. 부록: 데이터 탐색 \_ 3-1. Activity

#### A-6. FIGHT : 싸움(집단 간 유저들의 수치를 평균으로 집계)

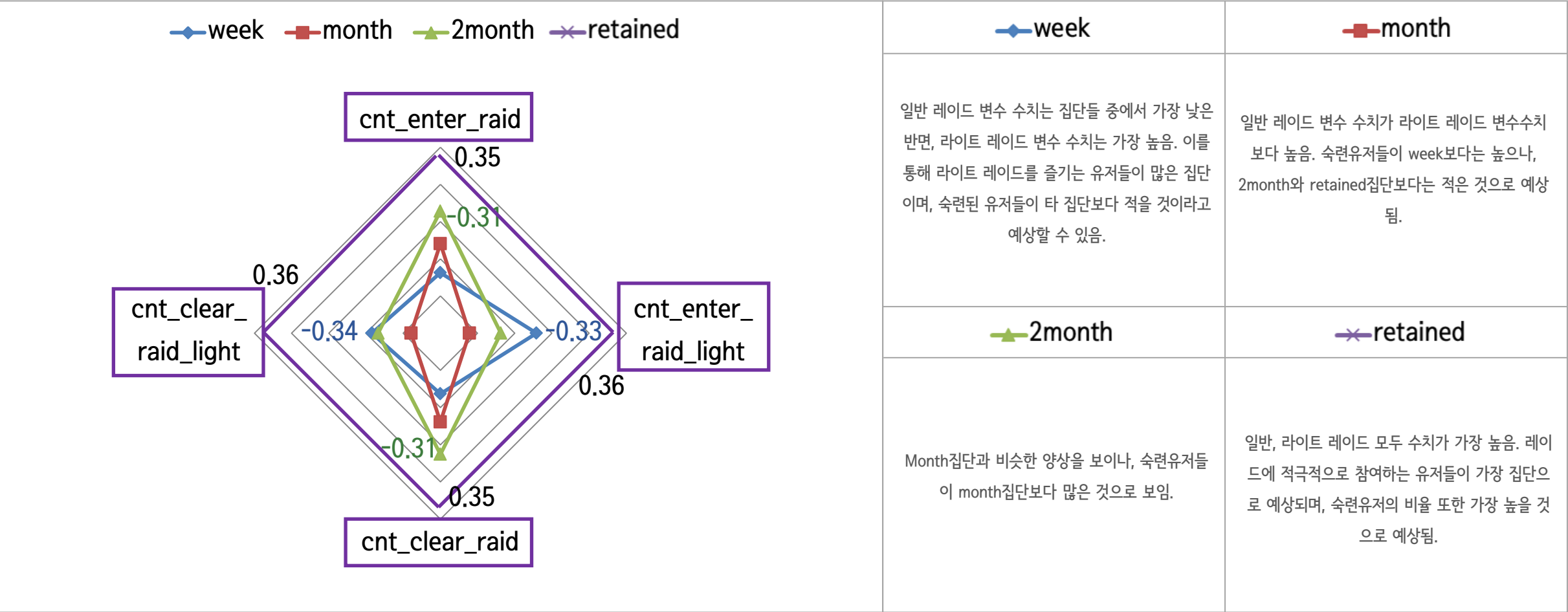
Week집단은 전투시간(game\_combat\_time)은 수치가 두드러지는 특징을 보임. retained 집단은 결투 참여횟수와 승리횟수(duel\_cnt, duel\_win), 전장참여 및 승리(partybattle\_cnt, partybattle\_win)가 높은 특징을 보임. Month와 2month집단은 전투시간과 횟수 변수 모두에서 낮은 수치를 보이며, 2month집단은 가장 전투를 하지 않는 것으로 보임.



### 3. 부록: 데이터 탐색 \_ 3-1. Activity

#### A-7. RAID : 레이드(집단 간 유저들의 수치를 평균으로 집계)

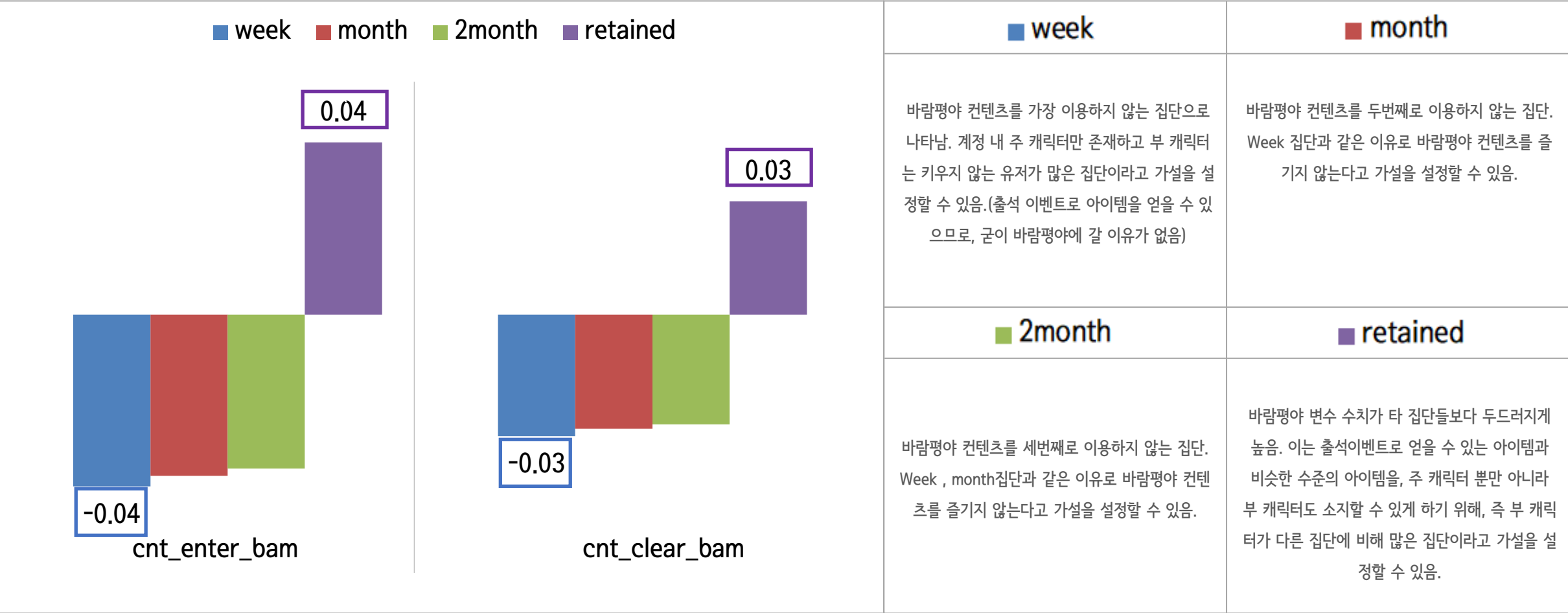
Week집단은 라이트 레이드(cnt\_enter\_raid\_light, cnt\_clear\_raid\_light) 변수의 수치가 높은 반면, 일반 레이드(cnt\_enter\_raid, cnt\_clear\_raid)의 변수 수치는 타 집단들보다 낮음. Month집단과 2month집단은 라이트보다 일반레이드의 수치가 높고, retained집단은 모든 레이드에서 수치가 가장 높음. 일반레이드 입장 횟수와 클리어 횟수는 게임 이탈시기와 비례하는 것으로 보임.



### 3. 부록: 데이터 탐색 \_ 3-1. Activity

#### A-8. BAM : 바람평야(인던)(집단 간 유저들의 수치를 평균으로 집계)

바람평야 관련 변수(cnt\_enter\_bam : 바람평야 입장횟수, cnt\_clear\_bam : 바람평야 클리어횟수)에서는 retained 집단이 다른 세 집단들보다 높은 수치를 나타냄. 이탈시기가 빠른 집단일수록 바람평야 관련 변수의 데이터 수치가 낮은 것을 확인할 수 있음. 현재 바람평야는 많은 유저들이 찾지 않는 콘텐츠(출석이벤트로 동급의 아이টে을 얻을 수 있기 때문)이므로, 해당 데이터를 통해 부 캐릭터 보유여부와 관련된 가설을 세울 수 있음.





## Category 03

부록: 데이터 탐색

### 3-2 \_ Guild Data

3-3 \_ Trade Data

3-4 \_ Payment Data

3-5 \_ Party Data



### 3. 부록: 데이터 탐색 \_ 3-2. Guild

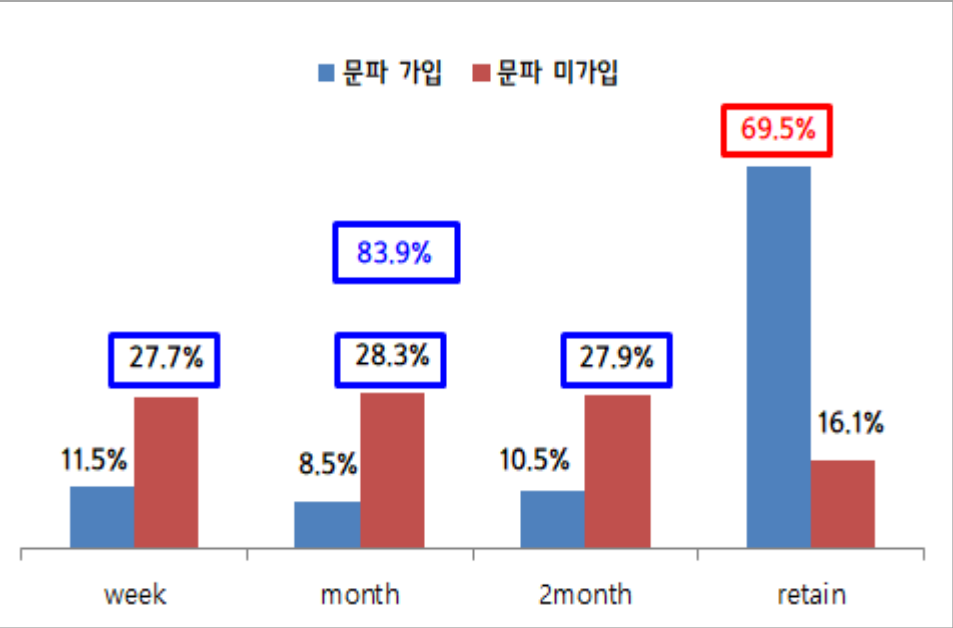
#### guild : 8주차 기준으로 문파별 문파원 목록을 집계한 데이터

문파별 문파원 목록을 집계한 자료는 총 2개의 변수와 9,963개의 인스턴스(row)로 구성되었다.

리스트를 정제해 10만명 유저들중에서 문파에 가입 여부와 가입한 유저들 데이터를 살펴본 결과, 전체 10만명 유저들 중에서 약 1만 6천 7백명(16.7%)이 문파에 가입한 데이터가 존재했다.

가입한 경험이 있는 유저들을 대상으로 이탈여부와 문파에 대해서 살펴보기로 했다.

	전체	가입	미가입
빈도	100,000	16,736	83,264
비율	(100%)	(16.7%)	(83.3%)



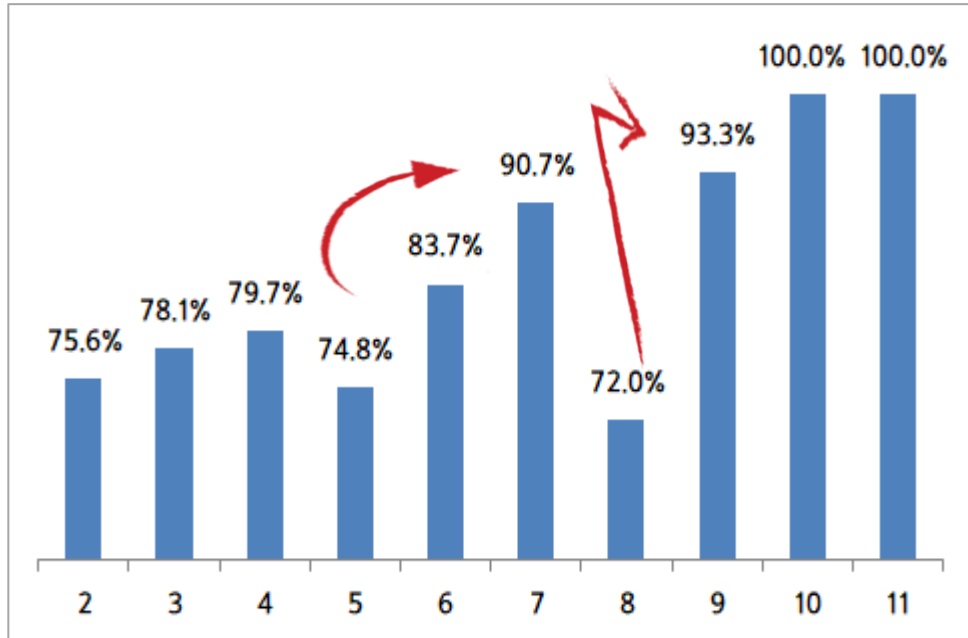
문파에 가입한 경험이 있는 유저 16,736명 중 약 70%는 retained인 반면에 문파 가입 경험이 없는 유저 83,264명의 83.9%는 이탈(week, month, 2month)을 하고 있는 것으로 나타남. 문파를 가입한 경험이 있는 유저들은 그렇지 않은 유저들에 비해서 게임에 대해 재미를 더 즐기고 있으며 더 오랫동안 게임을 플레이하는 것으로 생각됨.

이탈 여부		가입	미가입
유저 수 (10만 명 기준)		16,736(16.7%)	83,264(83.3%)
	week	11.5%	27.7%
	month	8.5%	28.3%
	2month	10.5%	27.9%
	retained	69.5%	16.1%

### 3. 부록: 데이터 탐색 \_ 3-2. Guild

#### guild : 8주차 기준으로 문파별 문파원 목록을 집계한 데이터

문파 가입 유저 16,736명 중 6,525명이 2개 이상의 캐릭터를 문파에 가입 했었다. 문파 중복에 대해서 2회 이상과 1회를 전체로 살펴봤을때 2개 이상 중복되어 있는 집단의 경우 retain의 비율이 10%p이상 차이가 나며 중복 횟수가 많을수록 retain의 비율이 감소하는 부분도 있지만 증가하는 추세를 보여주고 있다. 많은 캐릭터를 키우는 유저일 수록 retain의 비율이 높아지며 모델에 매우 유의한 변수라고 생각됨.



	2month	month	retained	week
2	372 (8.8%)	293 (7%)	3,181 (75.6%)	364 (8.6%)
3	111 (7.9%)	100 (7.1%)	1,097 (78.1%)	96 (6.8%)
4	37 (7.3%)	31 (6.1%)	404 (79.7%)	35 (6.9%)
5	24 (11%)	11 (5%)	163 (74.8%)	20 (9.2%)
6	6 (6.1%)	4 (4.1%)	82 (83.7%)	6 (6.1%)
7	2 (4.7%)	2 (4.7%)	39 (90.7%)	0 (0%)
8	2 (8%)	2 (8%)	18 (72%)	3 (12%)
9	0 (0%)	0 (0%)	14 (93.3%)	1 (6.7%)
10	0 (0%)	0 (0%)	2 (100%)	0 (0%)
11	0 (0%)	0 (0%)	3 (100%)	0 (0%)
2회 이상	554 (8.5%)	443 (6.8%)	5,003 (76.7%)	525 (8.0%)
1회	1,205 (11.8%)	974 (9.5%)	6,629 (64.9%)	1,403 (13.7%)

### 3. 부록: 데이터 탐색 \_ 3-2. Guild

#### guild : 8주차 기준으로 문파별 문파원 목록을 집계한 데이터

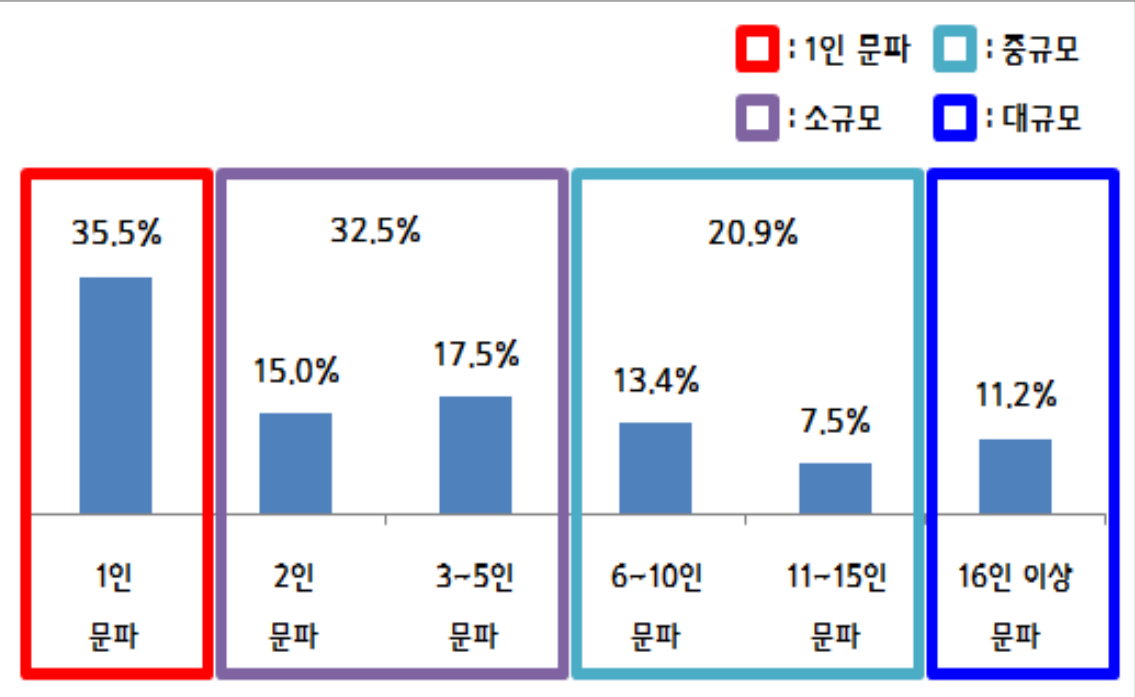
9,963개의 문파들에 규모에 대해서도 전체적으로 살펴보기로 했다. 문파 규모에 따른 빈도를 살펴보면 문파 인원이 1명인(1인 문파) 문파들이 총 3,536개로 전체의 35.5%를 차지하며 그 다음으로는 ‘2명’(1,491명, 15.0%), ‘3명’(783명, 7.9%), ‘4명’(395명, 5.7%) 등등 나타났다. 문파들 중 가장 인원수가 많았던 문파는 총 77명이 가입한 문파로써 1개가 존재했다.

문파 인원	문파 수	비율	문파 인원	문파 수	비율	문파 인원	문파 수	비율	문파 인원	문파 수	비율	문파 인원	문파 수	비율
1	3,536	35.5	16	85	0.9	31	24	0.2	46	1	0.0	61	1	0.0
2	1,491	15.0	17	91	0.9	32	21	0.2	47	6	0.1	63	2	0.0
3	783	7.9	18	75	0.8	33	15	0.2	48	7	0.1	64	1	0.0
4	568	5.7	19	76	0.8	34	25	0.3	49	7	0.1	65	1	0.0
5	395	4.0	20	71	0.7	35	18	0.2	50	4	0.0	66	2	0.0
6	334	3.4	21	58	0.6	36	15	0.2	51	5	0.1	67	1	0.0
7	307	3.1	22	49	0.5	37	19	0.2	52	5	0.1	68	1	0.0
8	259	2.6	23	49	0.5	38	20	0.2	53	3	0.0	69	1	0.0
9	225	2.3	24	44	0.4	39	10	0.1	54	6	0.1	70	2	0.0
10	207	2.1	25	40	0.4	40	7	0.1	55	2	0.0	71	1	0.0
11	186	1.9	26	33	0.3	41	12	0.1	56	3	0.0	73	1	0.0
12	174	1.7	27	37	0.4	42	10	0.1	57	1	0.0	77	1	0.0
13	144	1.4	28	45	0.5	43	5	0.1	58	2	0.0			
14	128	1.3	29	42	0.4	44	6	0.1	59	2	0.0			
15	111	1.1	30	37	0.4	45	5	0.1	60	2	0.0			

### 3. 부록: 데이터 탐색 \_ 3-2. Guild

#### guild : 8주차 기준으로 문파별 문파원 목록을 집계한 데이터

문파의 규모를 전체적으로 보기보다는 일정 부분별로 집산화하여 살펴보기로 했다. 전체 분포를 살펴보면 2개의 구분으로 나누었다. 구분1은 '1인 문파', '2인 문파', '3~5인 문파', '6~10인 문파', '11~15인 문파', '16인 이상 문파'로 총 6개 집단 구분2는 '개인', '소규모', '중규모', '대규모'로 살펴봤다. 구분1로 전체 비율을 살펴보면 1인 문파가 35.5%로 가장 높았으며, 그 다음으로는 '3~5인 문파'(17.5%), '2인 문파'(15.0%), '6~10인 문파'(13.4%), '16인 문파'(11.2%), '11~15인 문파'(7.5%) 순으로 나타났다. 구분2로 전체 비율을 살펴보면 구분 1과 같이 1인 문파가 35.5%로 가장 높으며, 그 다음으로는 '소규모'(32.5%), '중규모'(20.9%), '대규모'(11.2%)로 나타났다.

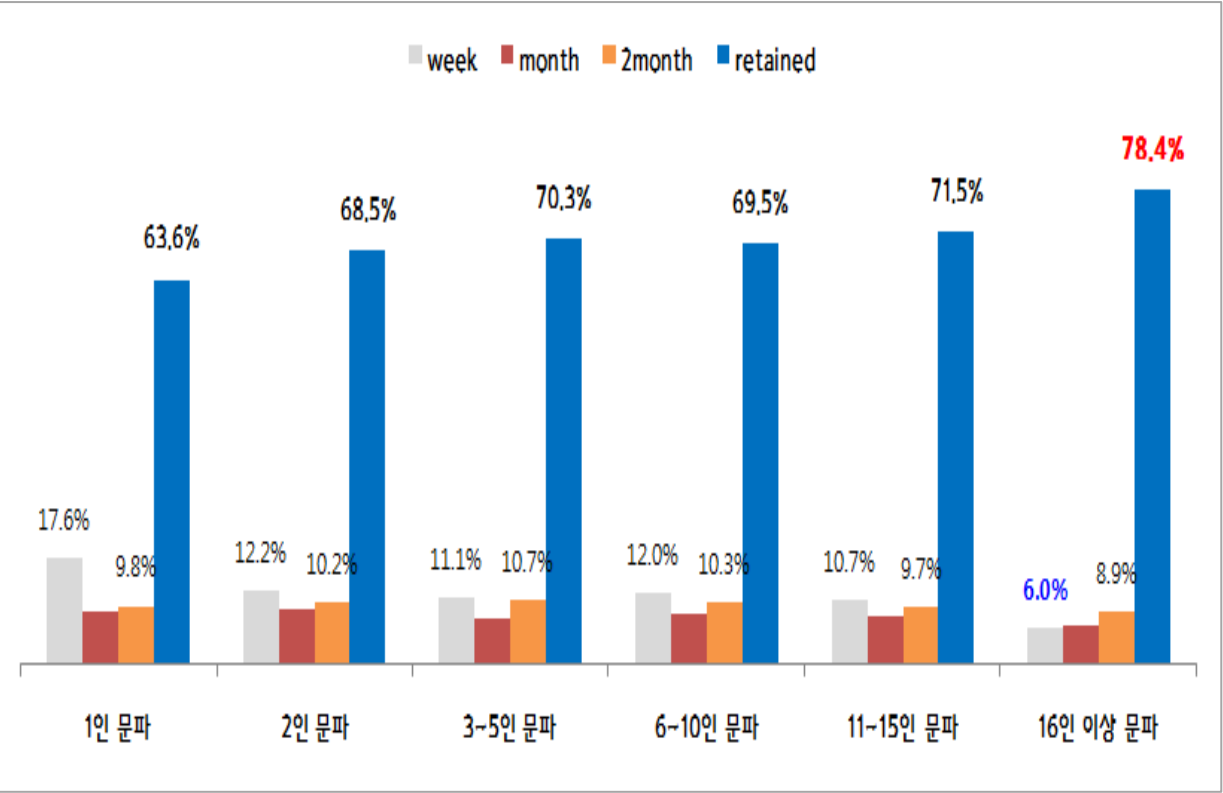


구분1	구분2	빈도(개)	비율(%)
1인 문파	개인	3,536	35.5
2인 문파	소규모	1,491	15.0
3~5인 문파		1,746	17.5
6~10인 문파	중규모	1,332	13.4
11~15인 문파		743	7.5
16인 이상 문파	대규모	1,115	11.2

### 3. 부록: 데이터 탐색 \_ 3-2. Guild

#### guild : 8주차 기준으로 문파별 문파원 목록을 집계한 데이터

문파를 가입 유저 16,736명이 8주간 포함되어 있었던 문파 총 27,238개의 문파들이 구분별 어떻게 분포되었는지 살펴보기로 함. 구분1로 살펴보면 16인 이상 문파가 11,245개(41.3%)로 가장 많았으며, 2인 문파가 1,851개(6.8%)로 가장 적었음. 구분2로 살펴보면 대규모 문파가 11,245개(41.3%)로 가장 많았으며, 개인 문파가 3,536개(13.0%)로 가장 적었음. 규모가 큰 문파에 가입될수록 retain의 비율이 증가하는 것을 볼 수 있으며 중규모(~15인 문파)까지 week의 비율이 높지만 16인 이상 문파에 가입한 인원일수록 week의 비율도 낮아지는 것을 볼 수 있음.



구분1	구분2	전체				
			week	month	2month	retained
1인 문파	개인 (13.0%)	3,536 (13.0%)	624 (17.6%)	317 (9%)	345 (9.8%)	2,250 (63.6%)
2인 문파	소규모 (17.6%)	1,851 (6.8%)	226 (12.2%)	169 (9.1%)	188 (10.2%)	1,268 (68.5%)
3~5인 문파		2,949 (10.8%)	327 (11.1%)	231 (7.8%)	317 (10.7%)	2,074 (70.3%)
6~10인 문파	중규모 (28.1%)	4,066 (14.9%)	487 (12%)	338 (8.3%)	417 (10.3%)	2,824 (69.5%)
11~15인 문파		3,591 (13.2%)	384 (10.7%)	292 (8.1%)	347 (9.7%)	2,568 (71.5%)
16인 이상 문파	대규모 (41.3%)	11,245 (41.3%)	680 (6%)	746 (6.6%)	1,002 (8.9%)	8,817 (78.4%)



## Category 03

부록: 데이터 탐색

### 3-3 \_ Trade Data

3-4 \_ Payment Data

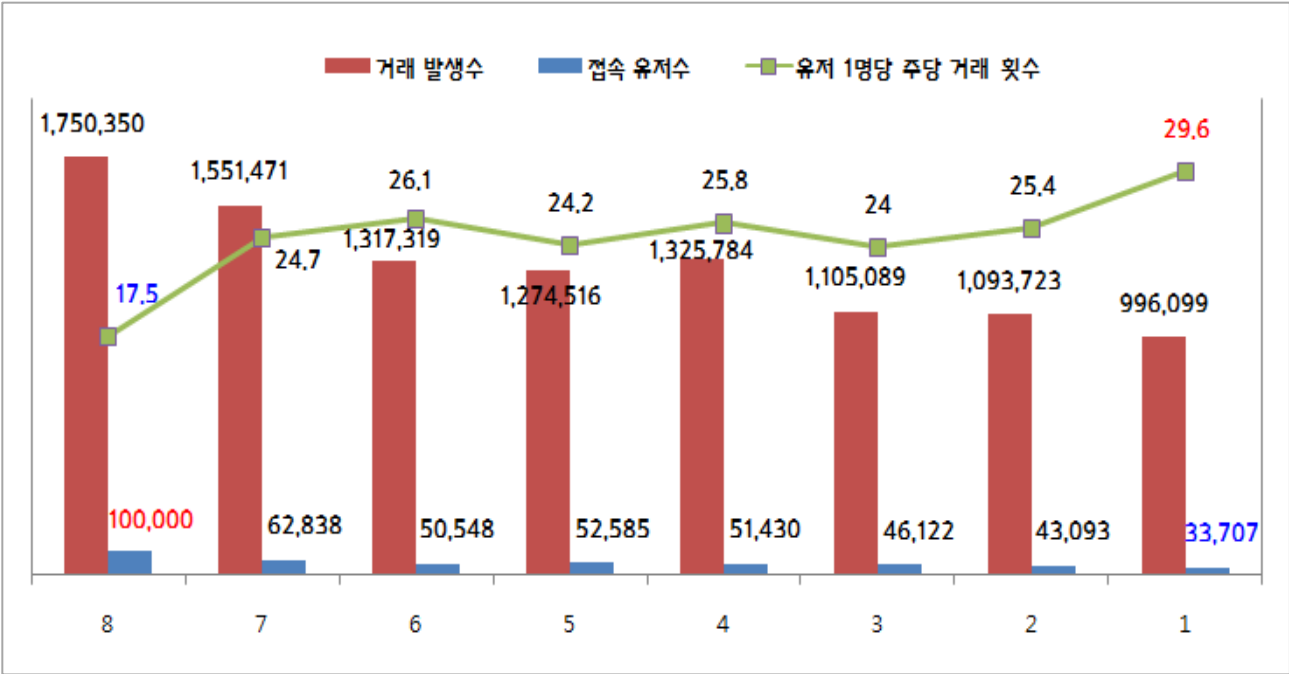
3-5 \_ Party Data



### 3. 부록: 데이터 탐색 \_ 3-3. Trade

#### Trade : 유저간 1:1 거래 내역을 집계한 데이터

유저간 1:1 거래 내역을 집계한 자료는 총 7개의 변수와 10,414,351개의 인스턴스(row)로 구성되어 있었다. 거래 발생 주차(1~8주)를 살펴보면 8주차가 16.8%로 가장 높았고, 그 다음으로는 ‘7주’(14.9%), ‘4주’(12.7%), ‘6주’(12.6%), ‘5주’(12.2%), ‘3주’(10.6%), ‘2주’(10.5%), ‘1주’(9.6%)순으로 나타났다. 5주차를 제외하고는 시간이 흐름에 따라 거래량이 증가하는 것을 볼 수 있었다. 유저 1명당 주당 거래 횟수는 평균 23.7회이며 가장 횟수가 많았던 주는 1주차로 가장 접속유저수도 적고 거래 횟수도 적지만 유저 1명당 거래횟수가 29.6회로 가장 적었던 8주차 17.5회보다 12.1회 이상 차이가 나는 것을 볼 수 있었다. 이를 통해 8주차에 신입 유저가 대량으로 유입됨에 따라 유저 수 대비 거래 횟수가 낮아졌다는 것을 예상할 수 있다.



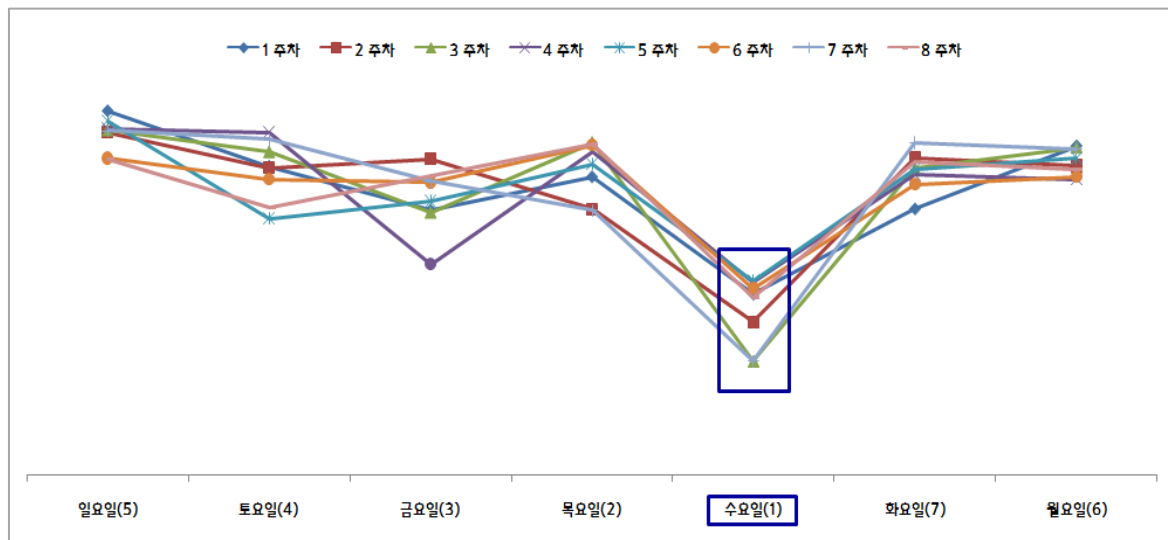
trade week	거래 발생수	비율	차이 (전주와 차이)	접속 유저수	접속 유저수 대비 거래 발생 비율
8	1,750,350	16.8%	1.9%	100,000	17.5
7	1,551,471	14.9%	2.2%	62,838	24.7
6	1,317,319	12.6%	0.4%	50,548	26.1
5	1,274,516	12.2%	-0.5%	52,585	24.2
4	1,325,784	12.7%	2.1%	51,430	25.8
3	1,105,089	10.6%	0.1%	46,122	24.0
2	1,093,723	10.5%	0.9%	43,093	25.4
1	996,099	9.6%	-	33,707	29.6
총계	10,414,351	100.0%	-	440,323	23.7

### 3. 부록: 데이터 탐색 \_ 3-3. Trade

#### Trade : 유저간 1:1 거래 내역을 집계한 데이터

거래 발생 일(trade\_day)을 살펴보면 일요일이 16.9%로 가장 높았고, 그 다음으로는 '월요일'(15.5%), '금요일'(15.3%), '화요일'(15.2%), '토요일'(15.1%). '금요일'(13.8%), '수요일'(8.2%)순으로 나타남.

주차별로 trade\_day 거래 발생량을 살펴본 결과, 모든 주차에서 수요일에 발생한 거래량이 제일 적은 것을 발견할 수 있었고 대부분 일요일에 가장 많은 거래량이 발생했다.



trade_day	거래 발생수	비율	차이(전주와 차이)
일요일 (5)	1,756,846	16.9%	1.8%
토요일 (4)	1,571,756	15.1%	1.3%
금요일 (3)	1,438,356	13.8%	-1.5%
목요일 (2)	1,593,521	15.3%	7.1%
수요일 (1)	851,196	8.2%	-7.0%
화요일 (7)	1,583,822	15.2%	-0.3%
월요일 (6)	1,618,854	15.5%	-1.3%
총 계	10,414,351	100.0%	-



1주차			2주차			3주차		
요일	거래발생량	비율	요일	거래발생량	비율	요일	거래발생량	비율
일요일(5)	180,175	18.1%	일요일(5)	186,184	17.0%	일요일(5)	189,532	17.2%
토요일(4)	152,795	15.3%	토요일(4)	166,942	15.3%	토요일(4)	177,520	16.1%
금요일(3)	131,166	13.2%	금요일(3)	171,992	15.7%	금요일(3)	144,388	13.1%
목요일(2)	147,619	14.8%	목요일(2)	144,474	13.2%	목요일(2)	182,615	16.5%
수요일(1)	89,692	9.0%	수요일(1)	83,361	7.6%	수요일(1)	62,665	5.7%
화요일(7)	131,765	13.2%	화요일(7)	172,617	15.8%	화요일(7)	168,439	15.2%
월요일(6)	162,887	16.4%	월요일(6)	168,153	15.4%	월요일(6)	179,930	16.3%
총계	996,099	1	총계	1,093,723	100%	총계	1,105,089	100.0%

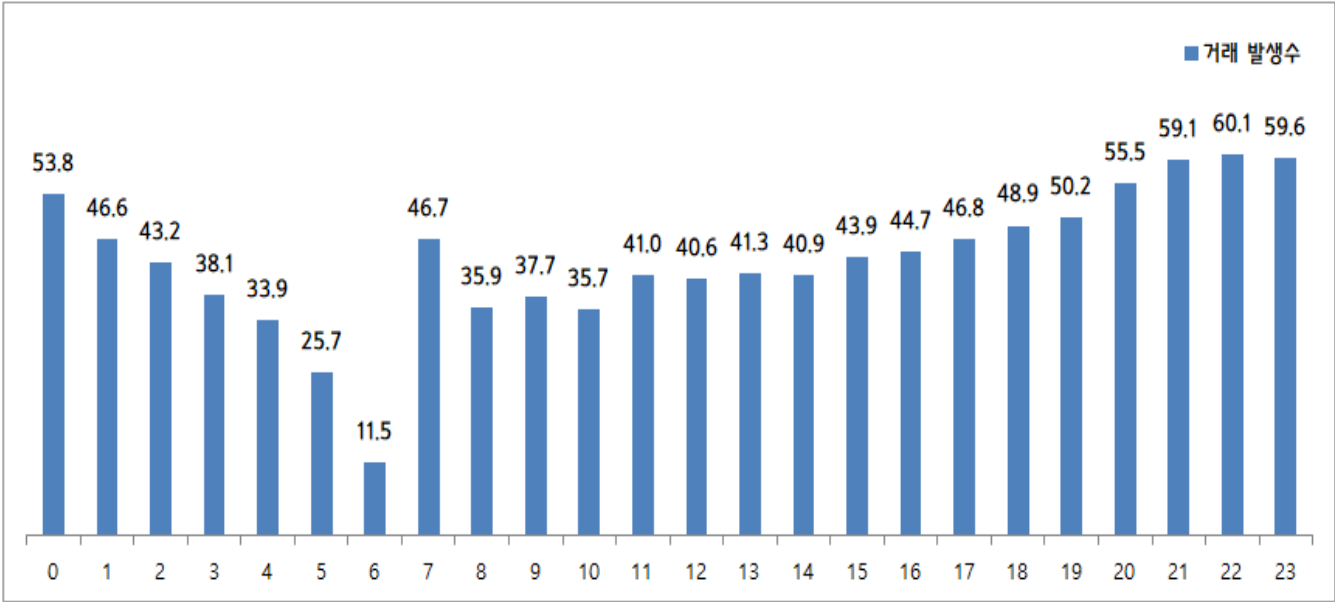
4주차			5주차			6주차		
요일	거래발생량	비율	요일	거래발생량	비율	요일	거래발생량	비율
일요일(5)	228,130	17.2%	일요일(5)	224,183	17.6%	일요일(5)	207,455	15.7%
토요일(4)	225,883	17.0%	토요일(4)	162,325	12.7%	토요일(4)	193,650	14.7%
금요일(3)	138,900	10.5%	금요일(3)	173,315	13.6%	금요일(3)	191,737	14.6%
목요일(2)	213,012	16.1%	목요일(2)	197,125	15.5%	목요일(2)	216,103	16.4%
수요일(1)	127,359	9.6%	수요일(1)	123,214	9.7%	수요일(1)	122,342	9.3%
화요일(7)	197,829	14.9%	화요일(7)	193,746	15.2%	화요일(7)	190,621	14.5%
월요일(6)	194,671	14.7%	월요일(6)	200,608	15.7%	월요일(6)	195,411	14.8%
총계	1,325,784	100.0%	총계	1,274,516	100.0%	총계	1,317,319	100.0%

7주차			8주차		
요일	거래발생량	비율	요일	거래발생량	비율
일요일(5)	265,829	17.1%	일요일(5)	275,358	15.7%
토요일(4)	259,309	16.7%	토요일(4)	233,332	13.3%
금요일(3)	226,270	14.6%	금요일(3)	260,588	14.9%
목요일(2)	204,295	13.2%	목요일(2)	288,278	16.5%
수요일(1)	87,897	5.7%	수요일(1)	154,666	8.8%
화요일(7)	256,527	16.5%	화요일(7)	272,278	15.6%
월요일(6)	251,344	16.2%	월요일(6)	265,850	15.2%
총계	1,551,471	100.0%	총계	1,750,350	100.0%

### 3. 부록: 데이터 탐색 \_ 3-3. Trade

#### Trade : 유저간 1:1 거래 내역을 집계한 데이터

거래 발생 일(trade\_day)을 살펴보면 일요일이 16.9%로 가장 높았고, 그 다음으로는 ‘월요일’(15.5%), ‘금요일’(15.3%), ‘화요일’(15.2%), ‘토요일’(15.1%). ‘금요일’(13.8%), ‘수요일’(8.2%)순으로 나타남.  
주차별로 trade\_day 거래 발생량을 살펴본 결과, 모든 주차에서 수요일에 발생한 거래량이 제일 적은 것을 발견할 수 있었고 대부분 일요일에 가장 많은 거래량이 발생했다.



시간	빈도	비율	시간	빈도	비율		
0	538,428	5.2%	12	405,830	3.9%		
1	466,483	4.5%	13	413,472	4.0%		
새벽 (23.2%)	2	431,776	4.1%	오후 (24.8%)	14	409,393	3.9%
	3	380,523	3.7%		15	438,509	4.2%
	4	338,779	3.3%		16	447,397	4.3%
	5	256,736	2.5%		17	468,086	4.5%
	6	114,602	1.1%		18	488,769	4.7%
7	466,833	4.5%	19	502,410	4.8%		
아침 (20.0%)	8	359,334	3.5%	저녁 (32.0%)	20	554,967	5.3%
	9	376,808	3.6%		21	591,152	5.7%
	10	356,843	3.4%		22	600,532	5.8%
	11	410,317	3.9%		23	596,372	5.7%

### 3. 부록: 데이터 탐색 \_ 3-3. Trade

#### Trade : 유저간 1:1 거래 내역을 집계한 데이터

거래 아이템 종류는 money가 53.84%로 가장 높았고, 그 다음으로는 'grocery'(46.10%), 'costume'(0.02%), 'accessory'(0.02%), 'gem'(0.01%), 'weapon'(0.01%)순으로 나타남. 유저들은 보통 잡화와 돈을 거래 하는게 대부분의 거래활동으로 보여짐. 주차별로 거래 아이템 종류를 살펴보면 'money'와 'grocery'의 거래량이 8주차에 가장 많고 1주차에 가장 적은 이유는 접속 유저수의 차이로 인해 발생한 것으로 보여짐. 8주차에 신규유저가 많이 유입된 것으로 예상되어 'money'와 'grocery'의 거래량이 많은 것으로 보임. 아이템 종류별로 살펴봤을때 money의 경우 다른 종류들에 비해 상대적으로 max값이 두드러진다. money는 상대적으로 큰 양을 주고 받았으며 편차가 큰 반면 두 번째로 거래량이 많은 grocery는 다른 아이템종류와 같은 분포를 보여주고 있다.

주차	money		grocery		costume	
	빈도	비율	빈도	비율	빈도	비율
1	555,107	9.9%	440,392	9.2%	207	9.8%
2	601,835	10.7%	491,188	10.2%	299	14.2%
3	613,430	10.9%	490,867	10.2%	349	16.6%
4	709,299	12.6%	615,859	12.8%	236	11.2%
5	687,473	12.3%	586,350	12.2%	249	11.8%
6	715,067	12.8%	601,615	12.5%	212	10.1%
7	816,065	14.6%	734,673	15.3%	267	12.7%
8	909,107	16.2%	840,465	17.5%	283	13.5%

주차	accessory		gem		weapon	
	빈도	비율	빈도	비율	빈도	비율
1	229	12.7%	122	11.1%	42	7.5%
2	199	11.0%	131	11.9%	71	12.7%
3	251	13.9%	125	11.4%	67	12.0%
4	203	11.3%	138	12.6%	49	8.8%
5	230	12.8%	140	12.8%	74	13.3%
6	206	11.4%	140	12.8%	79	14.2%
7	243	13.5%	134	12.2%	89	16.0%
8	241	13.4%	168	15.3%	86	15.4%

item_	count	mean	std	min	25%	50%	75%	max
amount								
money	5,607,383	0.05	1.38	-0.06	-0.06	-0.05	-0.05	173.68
grocery	4,801,409	-0.06	0.00	-0.06	-0.06	-0.06	-0.06	-0.06
costume	2,102	-0.06	0.00	-0.06	-0.06	-0.06	-0.06	-0.06
accessory	1,802	-0.06	0.00	-0.06	-0.06	-0.06	-0.06	-0.06
gem	1,098	-0.06	0.00	-0.06	-0.06	-0.06	-0.06	-0.06
weapon	557	-0.06	0.00	-0.06	-0.06	-0.06	-0.06	-0.06

Category 03

데이터 탐색

## 3-4 \_ Payment Data

3-5 \_ Party Data

1

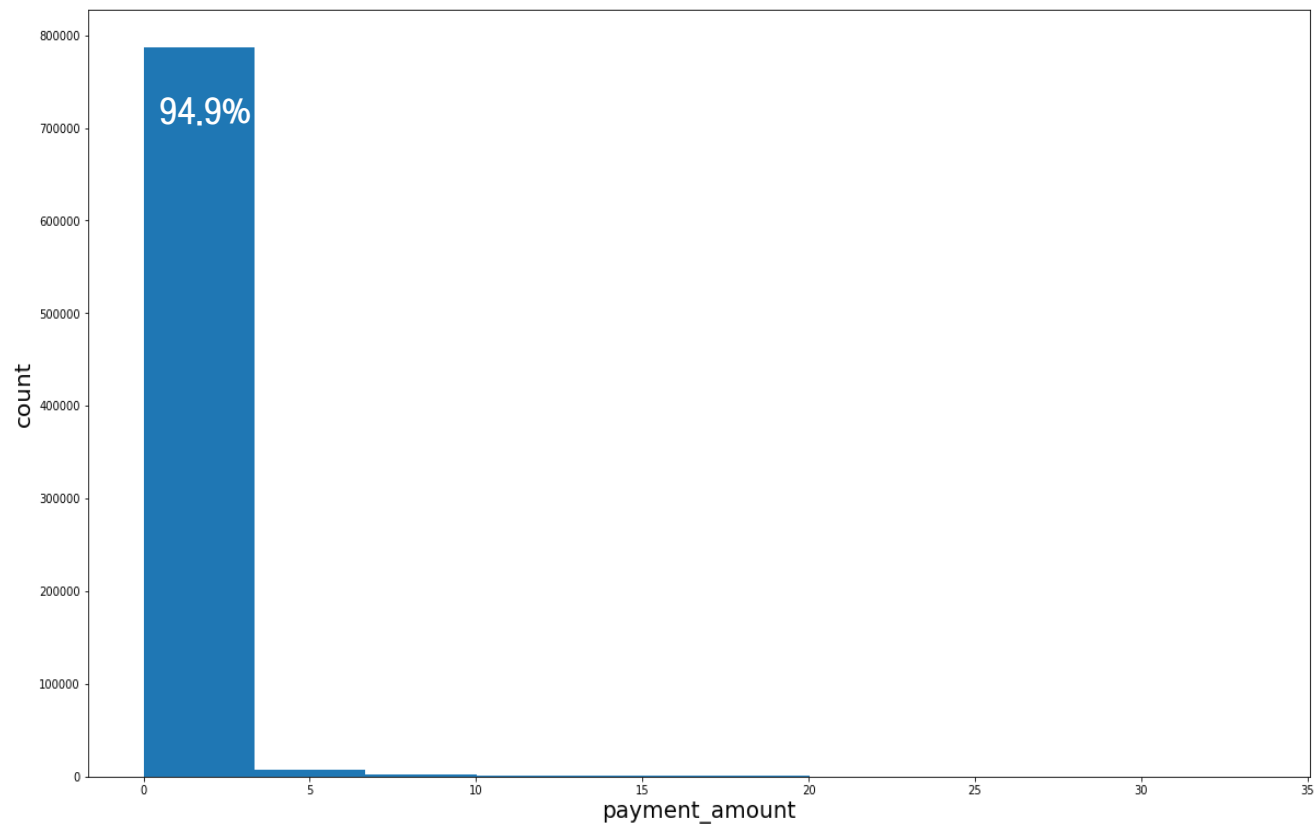




### 3. 부록: 데이터 탐색 \_ 3-4 Payment

#### Payment : 유저별 주간 결제 금액을 집계한 데이터

유저간 주간 결제 금액을 집계한 자료는 총 3개의 변수와 800,000개의 인스턴스(row)로 구성되어 있었다. 결제 금액(payment\_amount)을 카테고리화 해서 빈도분석 실시 결과 -0.1499(min)의 빈도가 전체의 94.91%로 대부분의 유저들이 같은 amount를 가지고 있었다. 이들은 게임에서 돈을 쓰지 않는 무과금 플레이어라고 생각하고 이들을 제외한 나머지 payment\_amount를 40,706개 데이터를 살펴보기로 했다.



payment_amount	빈도	비율	payment_amount	빈도	비율
-0.1499	759,294	94.91	:	:	:
-0.11474	454	0.06	24.10792	1	0.00
-0.07959	206	0.03	25.16261	1	0.00
-0.04443	133	0.02	26.42823	1	0.00
-0.00927	95	0.01	32.51027	1	0.00
:	:	:	33.24855	1	0.00

### 3. 부록: 데이터 탐색 \_ 3-4 Payment

#### Payment : 유저별 주간 결제 금액을 집계한 데이터

40,706개의 데이터를 가지고 빈도분석한 결과는 아래와 같다.

전체의 5%를 넘는 상위 6개는 전체의 약 55%를 차지하고 있었다.

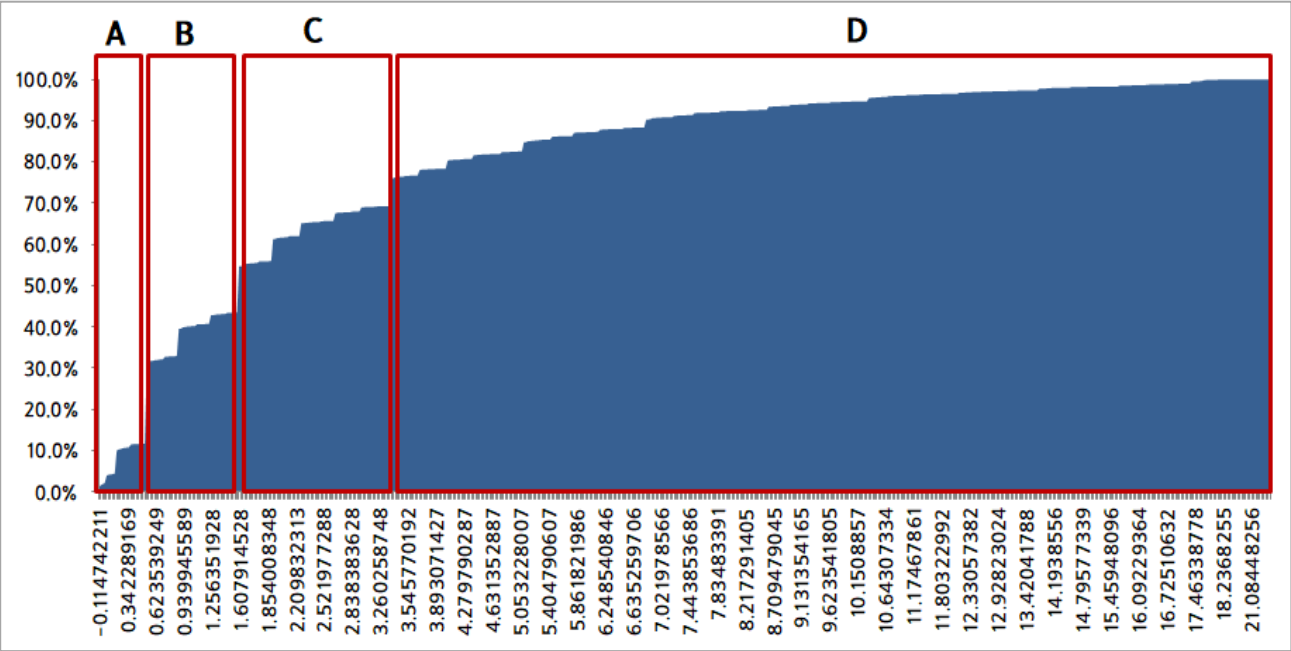
40,706개의 데이터를 가지고 전체 히스토그램을 살펴본 결과 일정 구간별 편차가 심한 구간이 발견됐고, 이런 구간별로 과금 유저들을 구분 짓는 기준으로 보았다.

전체 비율을 살펴보고 무 과금자를 제외한 총 4개 집단으로 나눌 수 있었다.

가장 비율이 높았던 집단은 B로 전체의 31.8%를 차지하고 있었고, 그 다음으로는 'D' (30.7%), 'C' (25.7%), 'A' (11.7%) 순으로 나타남.

집단별 이름을 A= 소 과금러, B=중 과금러, C=대 과금러, D=헤비 과금러로 명명하고 각 집단별로 이탈여부를 살펴보기로 함.

집단	빈도	비율	누적 비율	범위
A(소 과금러)	4,767	11.7%	11.7%	<= 0.5223
B(중 과금러)	12,959	31.8%	43.5%	<= 1.5728
C(대 과금러)	10,474	25.7%	69.3%	<= 3.3348
D(헤비 과금러)	12,506	30.7%	100.0%	<= 33.2485
-	40,706	100.0%	100.0%	-

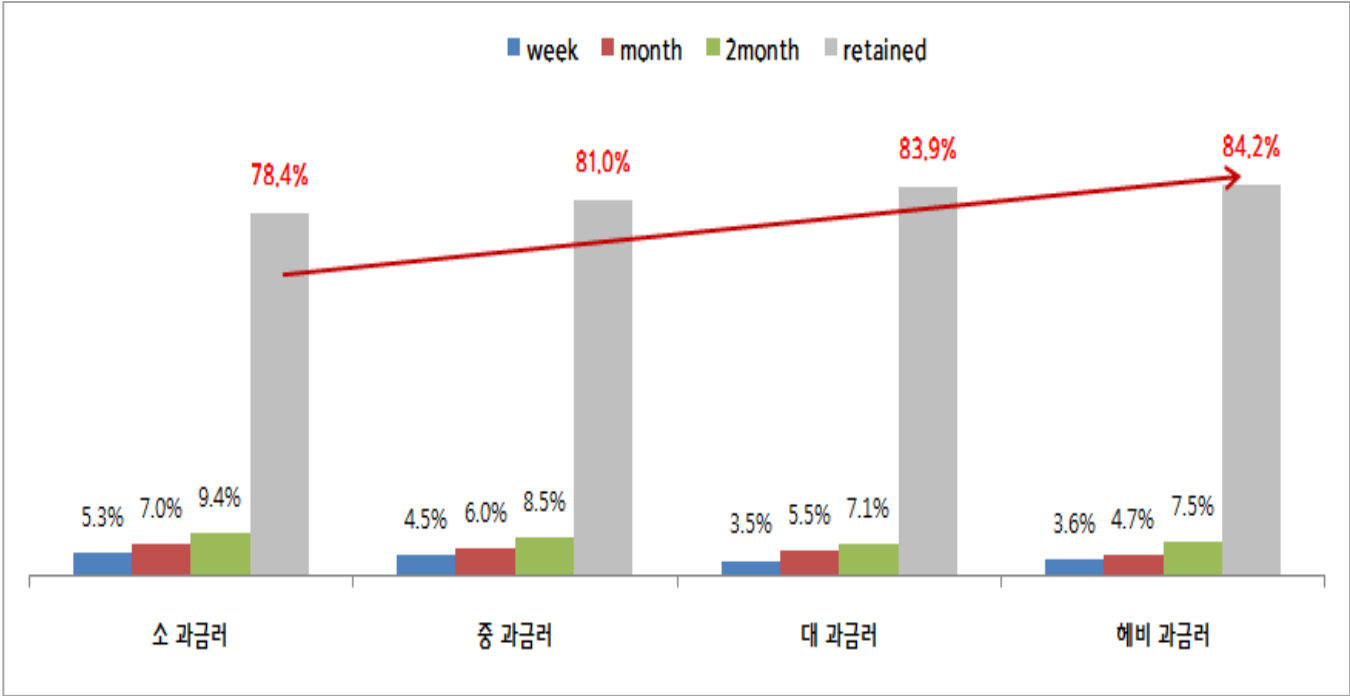


payment_amount	빈도	비율
0.553227	8,013	19.7%
1.607915	4,527	11.1%
0.904789	2,681	6.6%
3.365728	2,576	6.3%
0.201664	2,282	5.6%
1.959477	2,180	5.4%
:	:	:
총계	40,706	100%

### 3. 부록: 데이터 탐색 \_ 3-4 Payment

#### Payment : 유저별 주간 결제 금액을 집계한 데이터

payment 여부에 따른 집단별 이탈여부를 살펴보면 과금유저 집단중에서 **헤비 과금러**가 **retain**의 비율이 **84.2%**로 가장 높았고 그 다음으로는 ‘대 과금러’(83.9%), ‘중 과금러’(81.0%), ‘소 과금러’(78.4%) 순으로 나타남. 과금의 정도가 높아질수록 retain의 비율이 높아지는 것으로 보임.

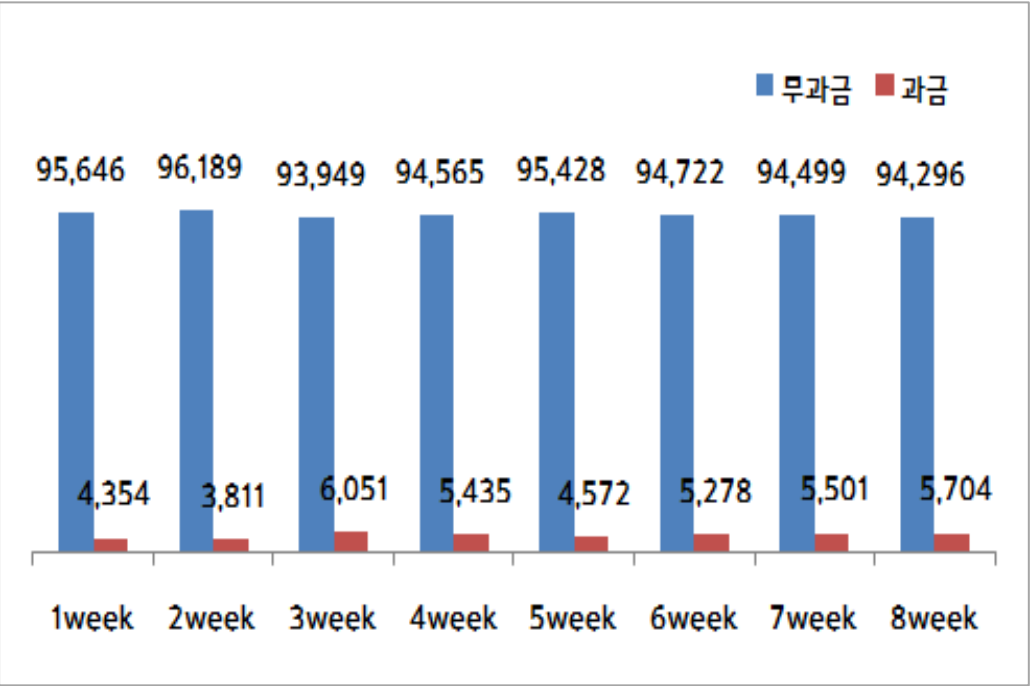


구분	week	month	2month	retained	전체	
무과금	198,351	197,714	196,775	166,454	759,294	
	(26.1%)	(26%)	(25.9%)	(21.9%)	(100%)	
과금	소	253	333	446	3,735	4,767
	과금러	(5.3%)	(7%)	(9.4%)	(78.4%)	(100%)
	중	581	783	1,097	10,498	12,959
	과금러	(4.5%)	(6%)	(8.5%)	(81%)	(100%)
	대	365	577	744	8,788	10,474
	과금러	(3.5%)	(5.5%)	(7.1%)	(83.9%)	(100%)
	헤비	450	593	938	10,525	12,506
	과금러	(3.6%)	(4.7%)	(7.5%)	(84.2%)	(100%)

### 3. 부록: 데이터 탐색 \_ 3-4 Payment

Payment : 유저별 주간 결제 금액을 집계한 데이터

주차별로 실제 과금을 한 유저들을 살펴보면 3주차때 과금을 한 유저가 6,051명(6.4%)으로 가장 높았고, 2주차의 경우 과금을 한 유저가 3,811명(4.0%)로 가장 낮았다.

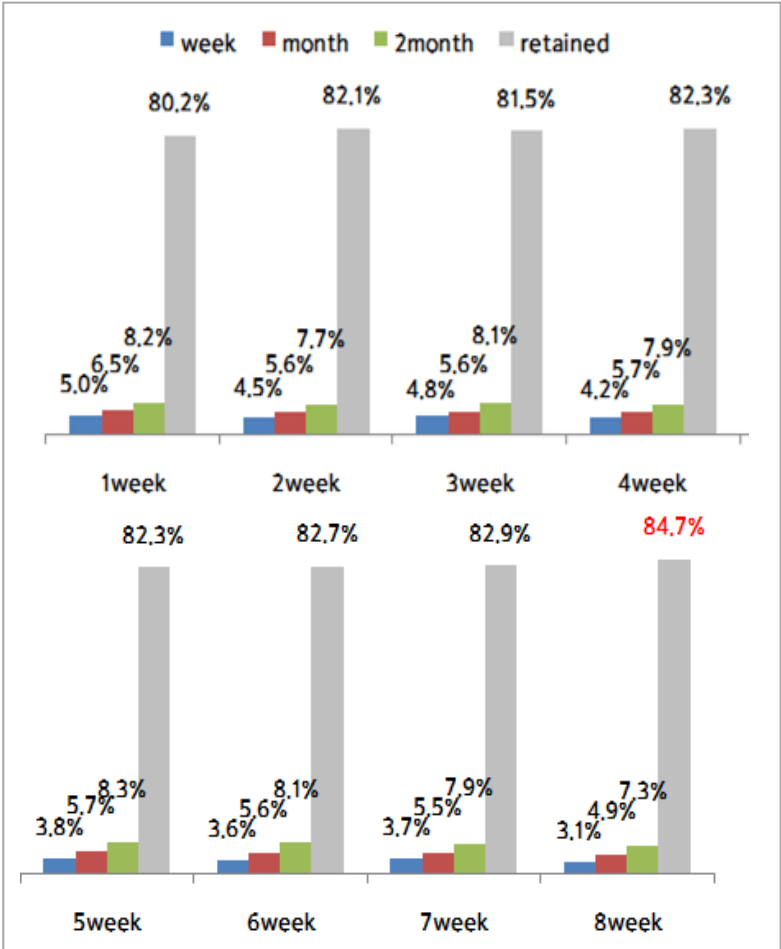


전체	1week	2week	3week	4week	5week	6week	7week	8week
800,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000
무과금	95,646	96,189	93,949	94,565	95,428	94,722	94,499	94,296
과금	4,354	3,811	6,051	5,435	4,572	5,278	5,501	5,704
비율	4.6%	4.0%	6.4%	5.7%	4.8%	5.6%	5.8%	6.0%

### 3. 부록: 데이터 탐색 \_ 1-4 Payment

#### Payment : 유저별 주간 결제 금액을 집계한 데이터

주차별로 유저들의 이탈여부를 과금여부로 살펴보면 retain의 경우 무과금일때는 비율이 가장 낮았으며, 과금일때는 비율이 전체의 80%를 넘어 대부분의 과금유저들은 retain이라고 볼 수 있었다. 과금여부의 변수는 retain의 특징을 잘 나타나는 변수라고 볼 수 있음.



구분	무과금					과금				
	week	month	2month	retain	합계	week	month	2month	retain	합계
1week	24,782	24,715	24,643	21,506	95,646	218	285	357	3,494	4,354
	(25.9%)	(25.8%)	(25.8%)	(22.5%)	(100%)	(5%)	(6.5%)	(8.2%)	(80.2%)	(100%)
2week	24,829	24,785	24,705	21,870	96,189	171	215	295	3,130	3,811
	(25.8%)	(25.8%)	(25.7%)	(22.7%)	(100%)	(4.5%)	(5.6%)	(7.7%)	(82.1%)	(100%)
3week	24,712	24,659	24,512	20,066	93,949	288	341	488	4,934	6,051
	(26.3%)	(26.2%)	(26.1%)	(21.4%)	(100%)	(4.8%)	(5.6%)	(8.1%)	(81.5%)	(100%)
4week	24,774	24,692	24,570	20,529	94,565	226	308	430	4,471	5,435
	(26.2%)	(26.1%)	(26%)	(21.7%)	(100%)	(4.2%)	(5.7%)	(7.9%)	(82.3%)	(100%)
5week	24,828	24,741	24,622	21,237	95,428	172	259	378	3,763	4,572
	(26%)	(25.9%)	(25.8%)	(22.3%)	(100%)	(3.8%)	(5.7%)	(8.3%)	(82.3%)	(100%)
6week	24,808	24,707	24,573	20,634	94,722	192	293	427	4,366	5,278
	(26.2%)	(26.1%)	(25.9%)	(21.8%)	(100%)	(3.6%)	(5.6%)	(8.1%)	(82.7%)	(100%)
7week	24,797	24,696	24,565	20,441	94,499	203	304	435	4,559	5,501
	(26.2%)	(26.1%)	(26%)	(21.6%)	(100%)	(3.7%)	(5.5%)	(7.9%)	(82.9%)	(100%)
8week	24,821	24,719	24,585	20,171	94,296	179	281	415	4,829	5,704
	(26.3%)	(26.2%)	(26.1%)	(21.4%)	(100%)	(3.1%)	(4.9%)	(7.3%)	(84.7%)	(100%)

Category 03

부록: 데이터 탐색

## 3-5 \_ Party Data





# 3. 부록: 데이터 탐색 \_ 3-5. Party

## Party-1. 전처리 후 260개의 Feature 도출

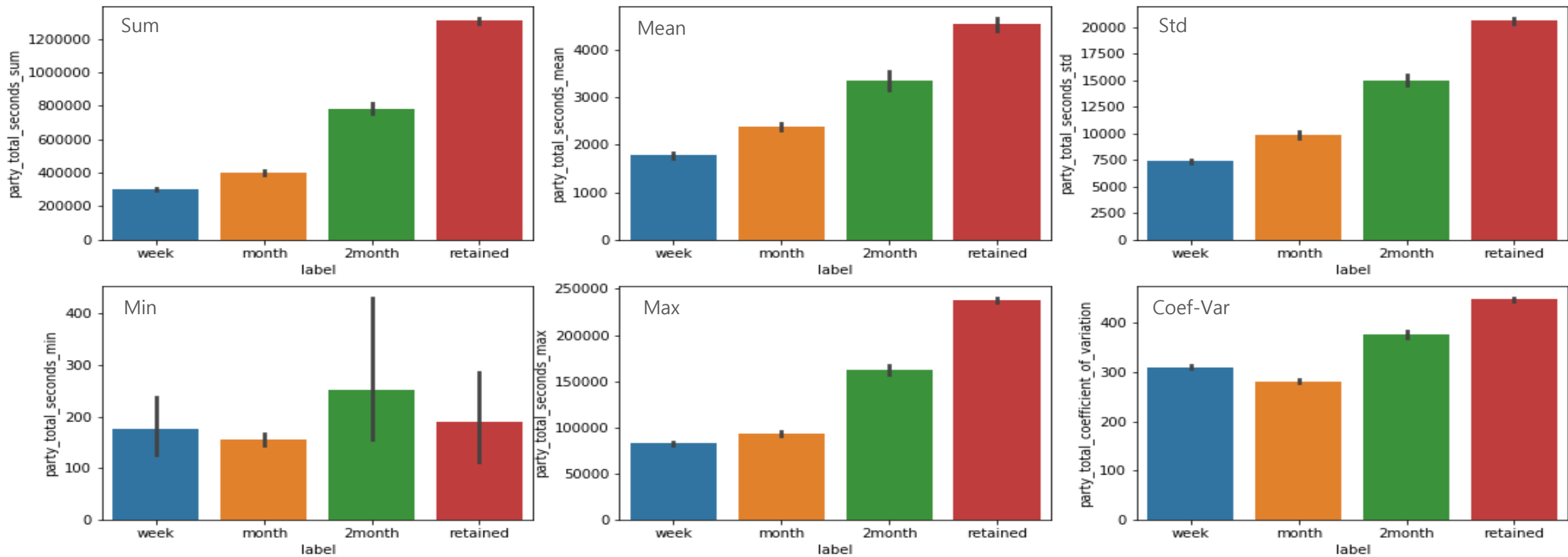
변수명	설명	변수의 개수
Party_total_seconds	<ul style="list-style-type: none"><li>1~8 주간 파티에 소속되어있던 시간의 합, 평균, 표준편차, 최댓값, 최솟값, 변동계수</li></ul>	6개
Party_by_re_day	<p>1주차 1일을 기준으로 8주차 7일까지의 상대 일수를 생성</p> <ul style="list-style-type: none"><li>일별 파티에 소속되어 있던 시간의 합, 평균</li><li>(8주차 총 시간 ) / (총 시간), (7&amp;8 주차 총 시간) / (총 시간), (6 &amp; 7 &amp; 8주차 총 시간) / (총 시간)</li><li>(8주차 평균 시간 )/ (평균 시간), (7&amp;8 주차 평균 시간) / (평균 시간), (6 &amp; 7 &amp; 8주차 평균 시간) / (평균 시간)</li><li>8주차의 변동 계수, 7&amp;8주차의 변동 계수, 6&amp;7&amp;8주차의 변동 계수</li></ul>	121개
Party_re_day_diff	<ul style="list-style-type: none"><li>(마지막 파티 종료 일) - (처음 파티 시작 일)</li><li>(마지막 파티 소속 시간)과 (처음 파티 소속 시간) 사이의 변화율</li></ul>	3개
Party_by_day_of_week	<ul style="list-style-type: none"><li>요일별 파티 소속 총 시간, 요일별 파티 소속 평균 시간</li></ul>	14개
Party_weekday_weekend	<ul style="list-style-type: none"><li>평일 파티 소속 시간의 합, 평균, 표준편차, 최댓값, 최솟값, 변동계수</li><li>주말 파티 소속 시간의 합, 평균, 표준편차, 최댓값, 최솟값, 변동계수</li><li>(평일 총 시간) / (총 시간) , (평일 평균 시간) / (총 평균 시간) , (평일 시간의 최댓값) / (시간의 최댓값)</li></ul>	15개
Party_by_time_class	<p>1~8주간 파티에 소속되어 있던 총 시간을 10개의 class로 나눔</p> <ul style="list-style-type: none"><li>class 별 해당 파티 소속 횟수</li><li>class 별 해당 파티 소속 총 시간</li><li>class 별 해당 파티 소속 평균 시간</li></ul>	30개
Party_seconds_by_hour	<ul style="list-style-type: none"><li>1~24시 별 파티 소속 총 시간</li><li>1~24시 별 파티 소속 평균 시간</li></ul>	48개
Party_seconds_by_hour_class	<p>1~24시를 "새벽", "아침", "오전", "오후", "저녁", "밤" 6개의 class로 나눔</p> <ul style="list-style-type: none"><li>class별 파티 소속 총 시간</li><li>class별 파티 소속 평균 시간</li></ul>	12개
Party_by_member_class	<p>파티 멤버 수를 "1인", "6인", "12인", "24인 이상", "그외 비정형" 5개의 class로 나눔</p> <ul style="list-style-type: none"><li>class별 파티 소속 총 시간</li><li>(class별 파티 소속 총 시간) / (총 파티 소속 시간)</li><li>평균 파티 멤버 수</li></ul>	11개

### 3. 부록: 데이터 탐색 \_ 3-5. Party

#### Party-2. 파티 소속 시간의 합, 평균, 표준편차, 최솟값, 최댓값, 변동계수

week에서 retained으로 갈수록 총 시간 및 평균 시간, 최댓값이 증가함을 알 수 있으며, 표준편차 또한 증가함을 알 수 있음.

파티 소속 시간의 총량이 적을 수록 이탈할 가능성이 있다고 볼 수 있음.



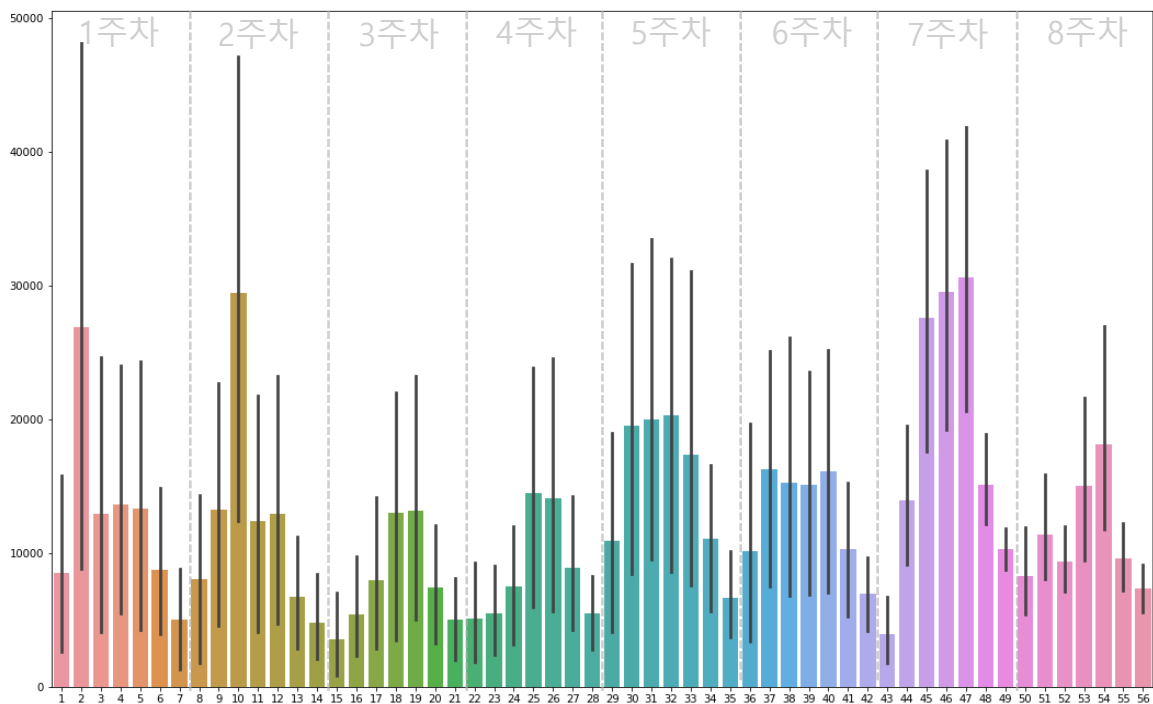
### 3. 부록: 데이터 탐색 \_ 3-5. Party

#### Party-3. 일별 party 소속 시간

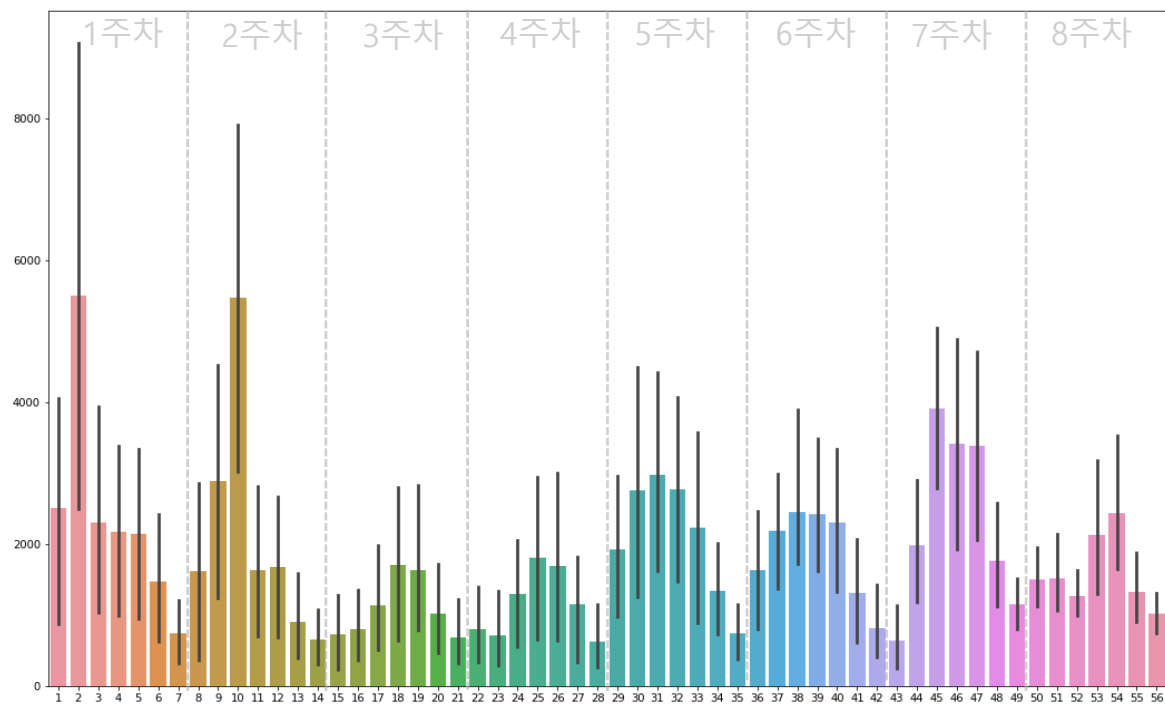
1주차 1일부터 8주차 7일까지 총 56일간의 변화를 살펴봄

1주차와 2주차에서는 수요일, 목, 금요일에 높은 수치를 확인할 수 있으며, 나머지 주차에서는 주말에 총 시간 및 평균 시간이 높은 것을 확인할 수 있음

#### 일별 party 소속 총 시간



#### 일별 party 소속 평균 시간



# 1. 데이터 탐색 \_ 3-5. Party

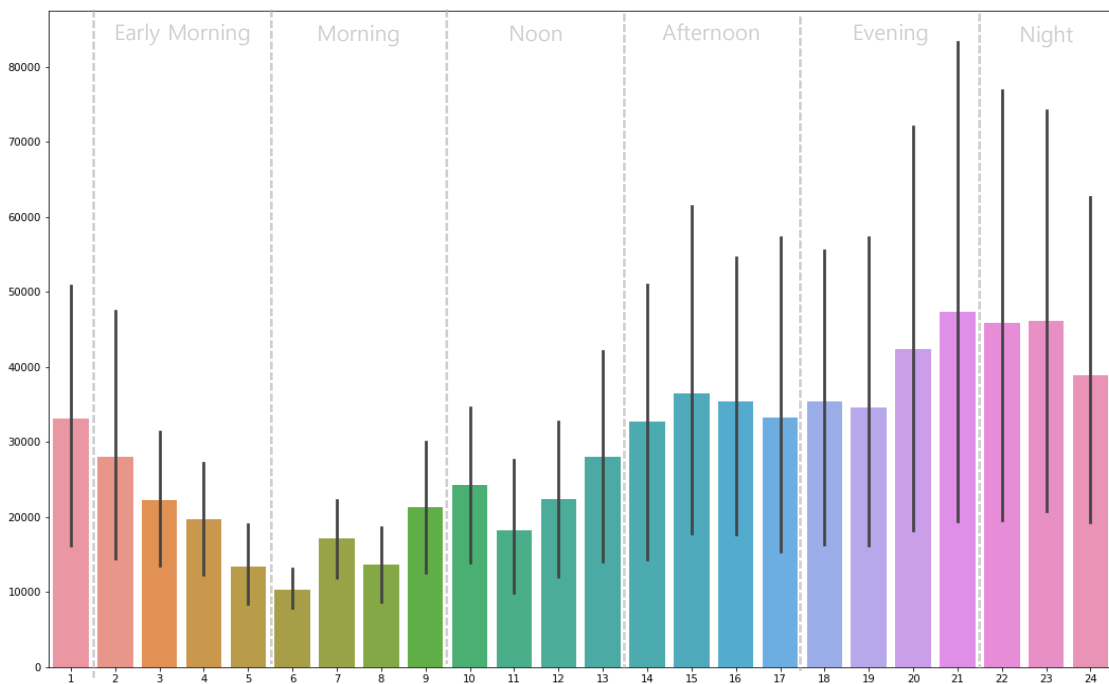
## Party-3. Party 시작 시간별 party 소속 시간

시간대 : Early Morning (02~06시), Morning (06~10시), Noon (10~14시), Afternoon (14~18시), Evening (18~22시), Night (22~02시)

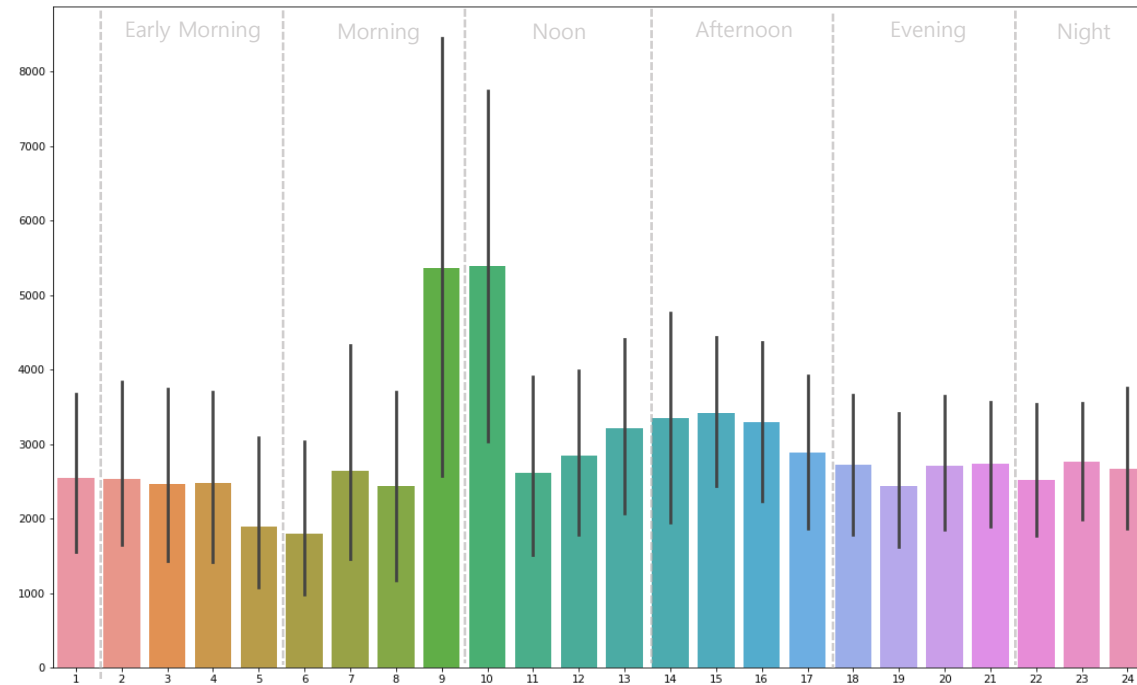
Party 시작 시간 별 총 시간을 보면, 새벽/오전 시간대보다는 오후/밤 시간대에 높은 수치를 확인할 수 있으나, 평균 시간을 보면 9시와 10시에 높은 수치를 확인할 수 있음

이는 게임 내 전체적으로는 오후/밤 시간대에 파티를 많이 시작 하나, 개인에 따라서 9시와 10시에 파티를 시작하는 경우가 많다는 것을 추론할 수 있음

### 시작 시간별 party 소속 총 시간



### 시작 시간별 party 소속 평균 시간

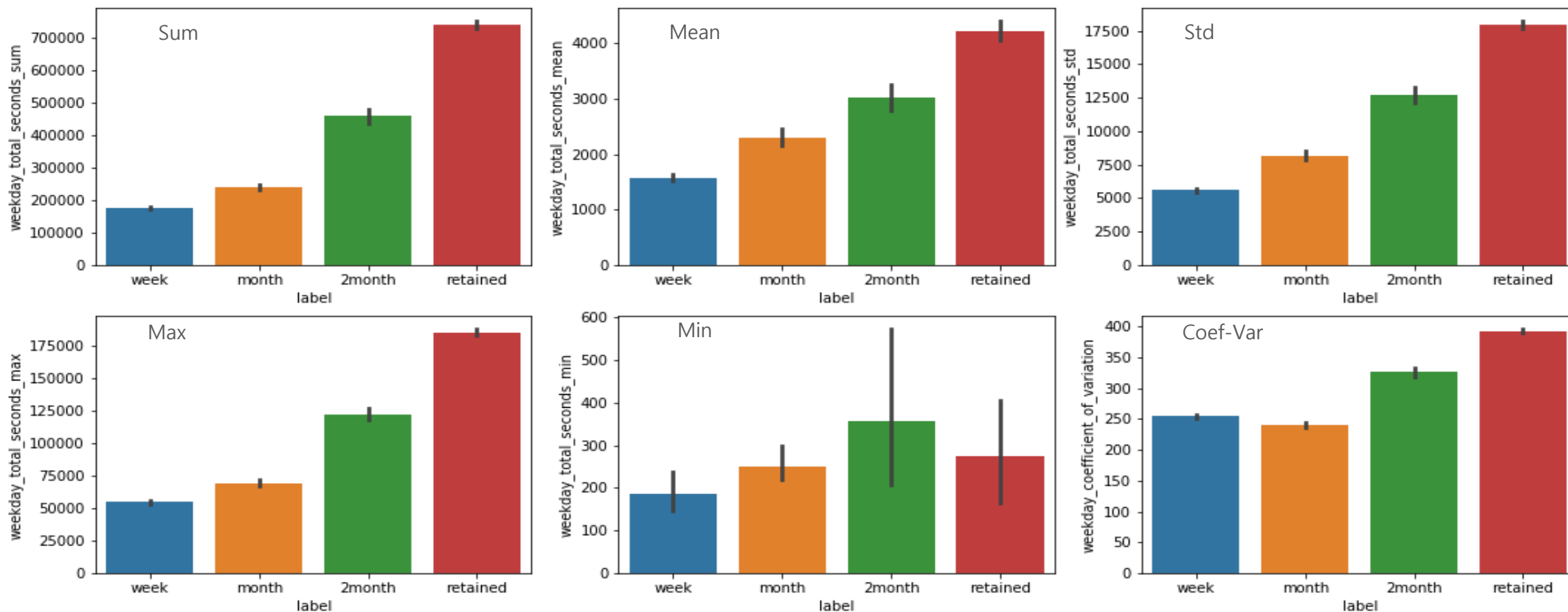


### 3. 부록: 데이터 탐색 \_ 3-5. Party

#### Party-4. 평일 파티 소속 시간의 합, 평균, 표준편차, 최솟값, 최댓값, 변동계수

week에서 retained으로 갈수록 평일 파티 총 시간 및 평균 시간, 최댓값이 증가함을 알 수 있으며, 표준편차 또한 증가함을 알 수 있음

평일 파티 소속 시간의 총량이 적을 수록 이탈할 가능성이 있다고 볼 수 있음.

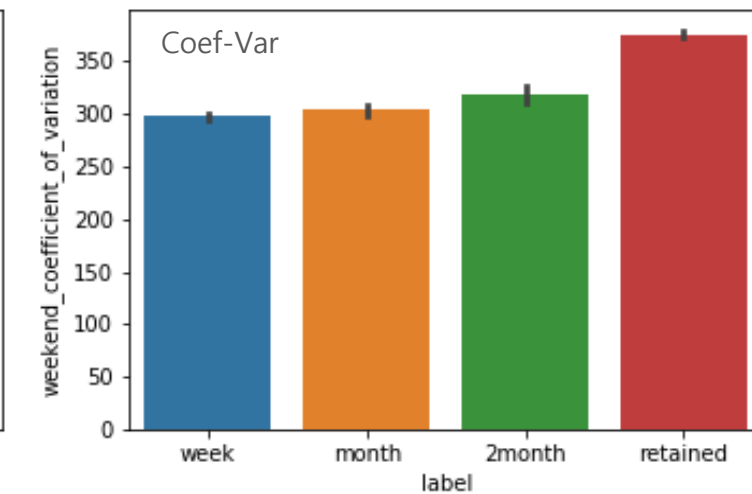
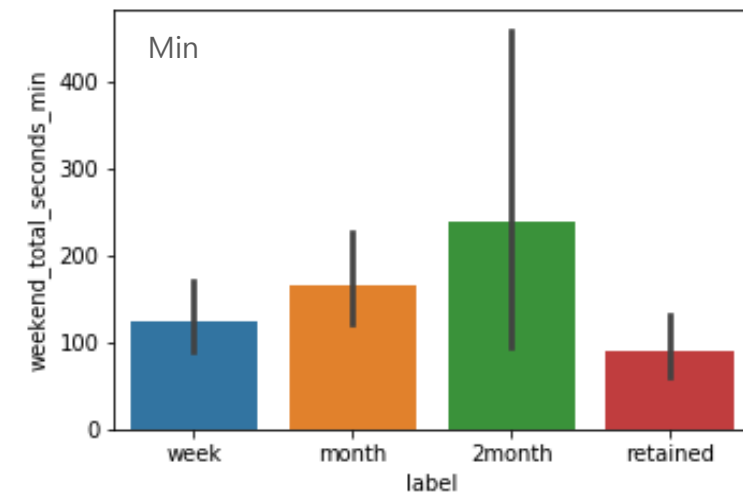
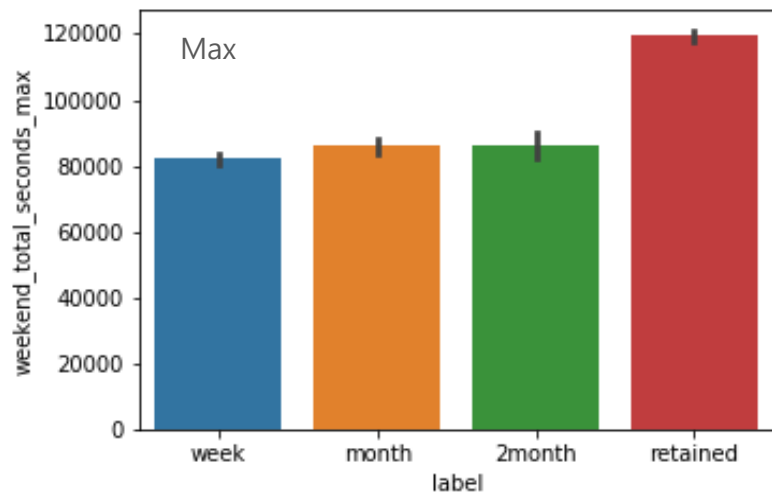
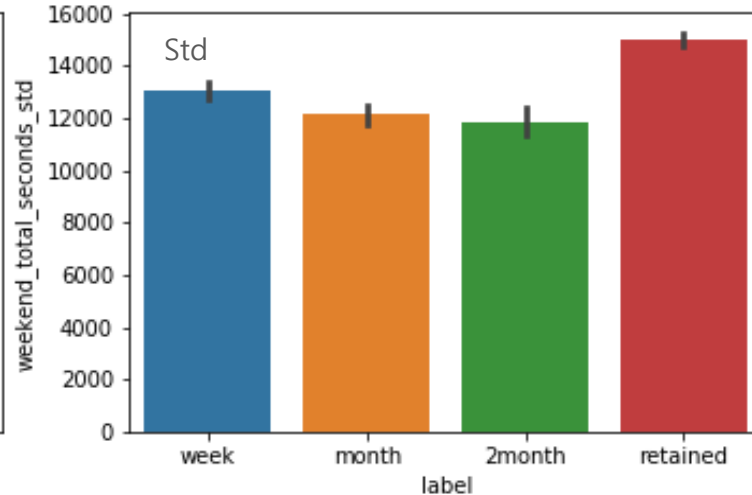
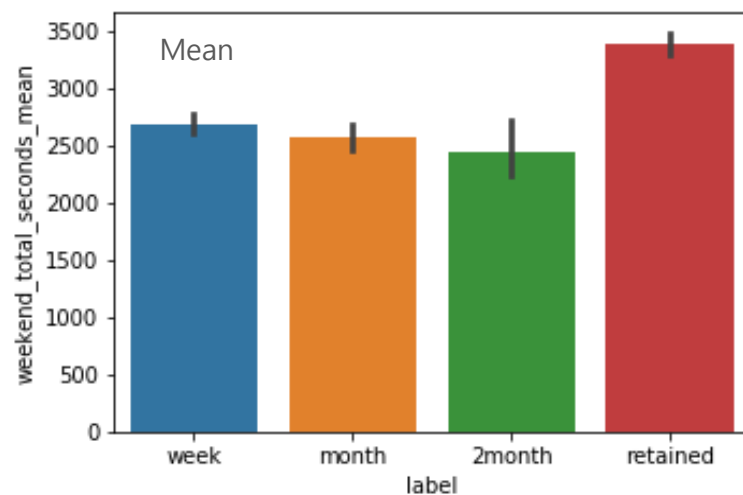
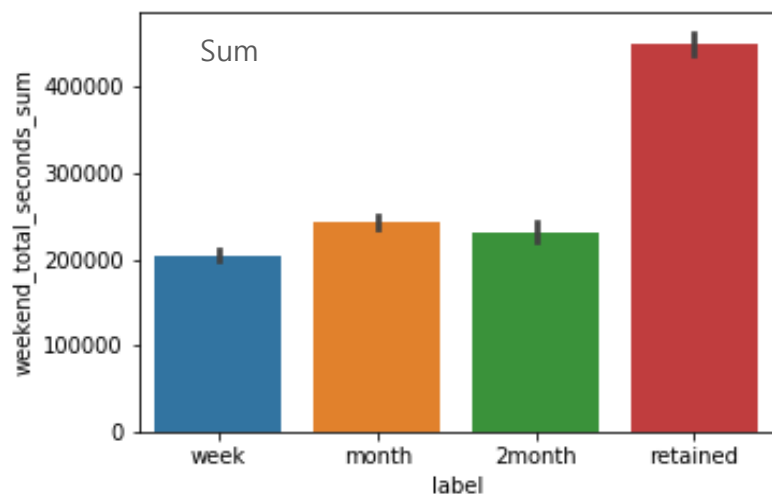


### 3. 부록: 데이터 탐색 \_ 3-5. Party

#### Party-4. 주말 파티 소속 시간의 합, 평균, 표준편차, 최솟값, 최댓값, 변동계수

반면, 주말의 경우 4개의 label 모두 비슷한 분포를 보임.

이를 통해 알 수 있는 것은 주말 뿐만 아니라 평일에도 지속적으로 게임을 즐기는 유저가 이탈을 적게 한다는 점임.





# 게임 유저 이탈 예측 모형 : 블레이드 & 소울

Thank for watching |

2018 빅 콘테스트 (Analysis 분야)

팀명 : 도토리

| 2018

| September

