



Blade & Soul

게임 속 세상을
상상하다

Team Level_0

Index

1. Introduction  

2. Preprocessing  

3. Clustering  

4. Modeling  

5. Analysis  

6. Conclusion  

유저들이 왜 이탈할까?

블레이드 & 소울에서는 무슨 일이 일어나고 있는가

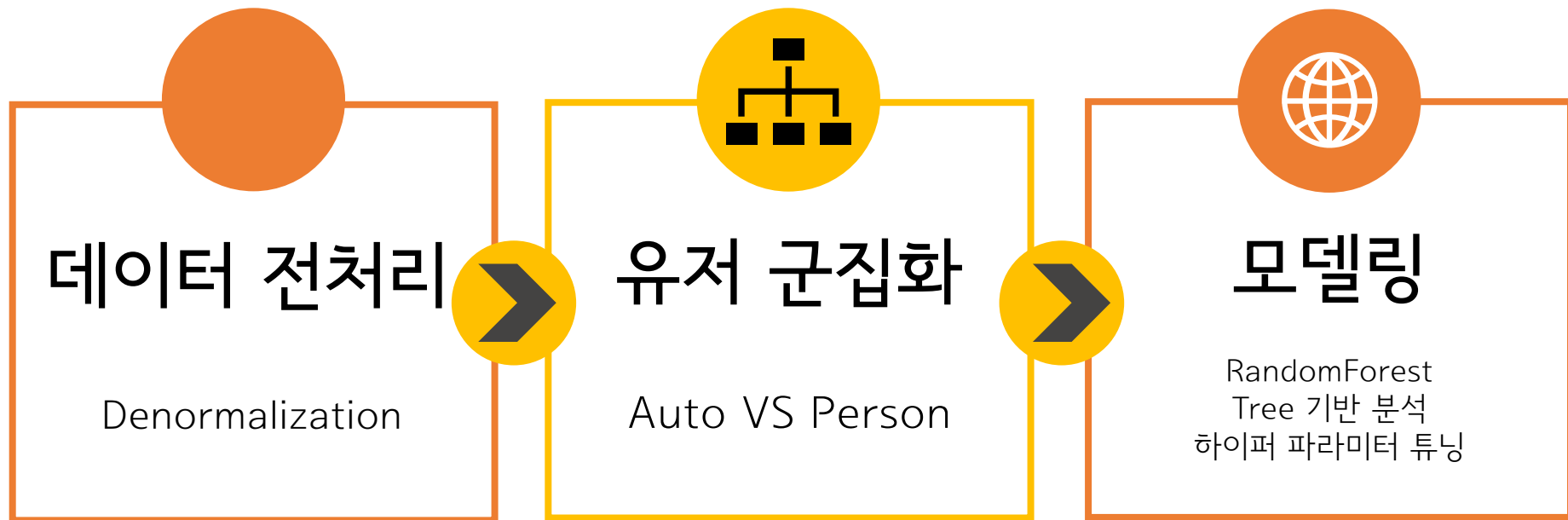
| Introduction



VS



| Introduction



| 데이터 전처리

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	wk	acc_id	cnt_dt	play_time	npc_exp	npc_hong	quest_exp	quest_hon	item_hong	game_con	get_money	duel_cnt	duel_win	partybattle	partybattle	cnt_enter	cnt_enter	cnt_enter	cnt_enter	cnt_enter	cnt_enter
2	7	3dc6f287	4	2.08881	4.40506	-0.2455	4.45406	3.0098	-0.3064	1.18401	-0.0234	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.3256	-0.3681	-0.402
3	8	3dc6f287	5	2.67346	4.76018	-0.2225	6.10748	4.75853	-0.3064	1.70953	-0.0234	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.2592	-0.3681	-0.402
4	3	b8856358	2	-0.6495	-0.231	-0.2908	-0.246	-0.4877	-0.3064	-0.5667	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
5	4	b8856358	2	-0.6582	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
6	5	b8856358	4	-0.6547	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
7	7	b8856358	2	-0.6593	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
8	8	b8856358	5	-0.6555	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
9	8	fa883ca79	6	-0.6582	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
10	8	d094b6b1	3	0.60628	1.37611	-0.143	1.96881	1.84285	-0.2129	0.404	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.2834	-0.0853	-0.1907	-0.06	-0.3681	-0.402
11	1	38e7088c	6	-0.0201	-0.2319	-0.1694	-0.2504	-0.4098	-0.2233	0.0913	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	1.64907	-0.0175	-0.1907	-0.06	-0.3681	-0.402
12	2	38e7088c	7	-0.2134	-0.2319	-0.1726	-0.2504	-0.4109	-0.2149	-0.1012	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	1.21025	-0.1532	-0.1907	-0.1928	-0.3681	-0.402
13	3	38e7088c	6	-0.2467	-0.2319	-0.2303	-0.2504	-0.4465	-0.2648	-0.2913	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	0.45075	-0.221	-0.1907	-0.2592	-0.3681	-0.402
14	4	38e7088c	4	-0.2073	-0.2319	-0.2223	-0.2504	-0.4275	-0.2482	-0.2708	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	0.93177	-0.2437	-0.1907	-0.2813	-0.3681	-0.402
15	6	38e7088c	6	0.35306	-0.2319	-0.0132	-0.2504	-0.3384	-0.1028	0.35646	-0.0234	-0.0362	-0.0413	-0.1675	-0.1599	1.51405	0.32177	-0.1907	0.27191	-0.3681	-0.402
16	7	38e7088c	7	0.33197	-0.2319	0.08303	-0.2504	-0.2981	-0.1215	0.59284	-0.0234	-0.0362	-0.0413	-0.1675	-0.1599	1.66595	0.61579	-0.1907	0.55959	-0.3681	-0.402
17	7	38e7088c	7	0.31884	-0.2319	0.04598	-0.2504	-0.3107	-0.1443	0.49592	-0.0234	-0.0362	-0.0413	-0.1675	-0.1599	1.8938	0.61579	-0.1907	0.55959	-0.3681	-0.402
18	7	38e7088c	7	-0.0732	-0.2319	-0.1442	-0.2504	-0.3706	-0.2378	0.07907	-0.0234	-0.0362	-0.0413	-0.1675	-0.1599	2.12165	0.05037	-0.1907	0.00637	-0.3681	-0.402
19	7	38e7088c	7	0.82469	0.34751	0.48164	-0.104	0.17748	1.18944	0.35446	-0.0215	-0.0362	-0.0413	-0.1675	-0.1599	-0.1231	0.18607	-0.1907	0.16127	0.62687	0.6632
20	7	38e7088c	7	0.51614	0.03198	0.30305	-0.2006	0.24166	0.5558	0.46279	-0.0196	-0.0362	-0.0413	1.22053	1.16852	-0.3087	0.32177	-0.1907	0.27191	2.11936	0.6632
21	7	38e7088c	7	0.74263	-0.137	0.33365	-0.231	0.36105	0.53918	0.48663	-0.0203	0.333	0.15867	2.90595	1.54807	-0.3087	0.29915	2.50628	0.44895	1.62186	0.6632
22	7	38e7088c	7	0.63807	0.09496	-0.0421	-0.1864	0.02974	0.38337	0.32841	-0.0223	-0.0362	-0.0413	1.6171	1.16852	-0.2666	-0.1532	0.70831	-0.1264	1.12436	0.6632
23	7	38e7088c	7	1.13021	0.58033	0.45099	-0.0199	0.03735	1.41796	0.80343	-0.0181	-0.0362	-0.0413	1.22053	0.59921	-0.2328	0.07298	4.30425	0.36043	1.62186	0.6632
24	5	38e7088c	5	0.56503	0.01287	0.52716	-0.1314	0.23972	1.02947	0.65157	0.00557	-0.0362	-0.0413	0.82396	0.21967	-0.2497	0.25392	5.20324	0.60385	2.11936	0.6632
25	5	38e7088c	5	0.44605	-0.2282	0.62342	-0.0642	0.29728	1.1562	0.68202	-0.0117	-0.0362	-0.0413	0.12996	0.0299	-0.3087	0.2313	6.40188	0.67023	1.62186	0.6632
26	7	38e7088c	7	1.33999	-0.1419	2.96872	-0.1954	1.07838	1.12711	1.67422	-0.004	-0.0362	-0.0413	0.32824	-0.1599	-0.0218	1.52047	5.20324	1.93158	1.62186	0.6632
27	4	38e7088c	4	-0.6552	-0.2319	-0.2908	-0.2504	0.46296	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
28	2	38e7088c	2	-0.6589	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
29	3	8d91936f	4	-0.6546	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
30	4	8d91936f	5	-0.6557	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
31	5	8d91936f	4	-0.6569	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
32	6	8d91936f	2	-0.6591	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
33	7	8d91936f	2	-0.6592	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
34	8	8d91936f	4	-0.6552	-0.2319	-0.2908	-0.2504	-0.4877	-0.3064	-0.5698	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.4246	-0.1907	-0.4583	-0.3681	-0.402
35	1	10ec0352	7	-0.4329	-0.2319	-0.1964	-0.2504	-0.4662	-0.2752	-0.1921	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.108	-0.1907	-0.1485	-0.3681	-0.402
36	2	10ec0352	7	-0.462	-0.2319	-0.2024	-0.2504	-0.4499	-0.2233	-0.3434	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.0175	-0.1907	-0.06	-0.3681	-0.402
37	3	10ec0352	6	-0.465	-0.2319	-0.2119	-0.2504	-0.451	-0.244	-0.3709	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.0853	-0.1907	-0.1264	-0.3681	-0.402
38	4	10ec0352	5	-0.5671	-0.2319	-0.2549	-0.2504	-0.4672	-0.3001	-0.4753	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.2663	-0.1907	-0.3034	-0.3681	-0.402
39	5	10ec0352	7	-0.4588	-0.2319	-0.2412	-0.2504	-0.4665	-0.3064	-0.3637	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.221	-0.1907	-0.2592	-0.3681	-0.402
40	6	10ec0352	6	-0.4959	-0.2319	-0.1888	-0.2504	-0.4481	-0.2752	-0.3612	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.1306	-0.1907	-0.1707	-0.3681	-0.402
41	7	10ec0352	7	-0.4547	-0.2319	-0.2025	-0.2504	-0.4666	-0.2731	-0.2497	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.1984	-0.1907	-0.2371	-0.3681	-0.402
42	8	10ec0352	3	-0.3302	-0.2319	-0.261	-0.2504	-0.4837	-0.2939	-0.4116	-0.0235	-0.0362	-0.0413	-0.1675	-0.1599	-0.3087	-0.3567	-0.1907	-0.392	-0.3681	-0.402

| 데이터 전처리

원본 데이터를 볼 때 의미하는 바를 파악하기 힘들다.

실제값 최소변화 단위를 알고 있다면,

정규화 된 값의 최소값과 그 다음 작은값을 이용하여
Denormalize 할 수 있다.

$$\min0 = \frac{x_0 - m}{s}$$

$$\min1 = \frac{x_1 - m}{s}$$

$$s = \frac{x_1 - x_0}{\min1 - \min0}$$

$$m = x_0 - s \times \min0$$

	A	B	C	D	E	F	G	H	I	J	K
1	wk	acc_id	cnt_dt	play_time	npc_exp	npc_hong	quest_exp	quest_hor	item_hong	game_cor	get_mone
14		5 00028f0ac	2	4	0	0	0	0	0	0	0
15		7 00028f0ac	1	2	0	0	0	0	0	0	0
16		8 00028f0ac	5	8.6	0	0	0	0	0	0	0
17		1 00036e5b	3	486.7	58548	23515	2588780	8848160	80000	161.7	0.6909
18		3 00036e5b	6	141.2	0	0	0	162000	0	34.3	1.1588
19		4 00036e5b	7	656.9	0	0	0	189000	0	39.6	1.4231
20		6 00036e5b	2	27.4	0	0	0	27000	0	5.3	0.1743
21		7 00036e5b	7	149.1	0	0	0	148500	0	34.2	1.1255
22		8 00036e5b	6	140.8	0	0	0	162000	0	32.1	1.128
23		2 00037fe8e	1	3.3	0	0	0	0	0	0	0
24		8 00037fe8e	2	18.5	168	0	8610	0	0	2	0
25		1 0003b127	7	356.8	0	50640	0	198100	200000	159.2	1.4074
26		2 0003b127	7	428.9	0	66469	0	222000	220000	184.8	1.789
27		3 0003b127	6	1144.3	0	91060	0	306760	400000	622.2	2.0243
28		4 0003b127	3	61	0	0	0	81000	0	20	0.5304
29		5 0003b127	2	56.8	0	0	0	54000	0	17	0.5396
30		6 0003b127	4	82.1	0	0	0	108000	0	26.2	0.7906
31		7 0003b127	6	140.1	0	0	0	162000	0	45	1.2963
32		8 0003b127	4	92.1	0	636	0	119200	0	27.3	0.725
33		8 0004733c	5	4.8	0	0	0	0	0	0	0
34		6 00047f65f	2	20.3	168	0	8610	0	0	1.6	0
35		7 00047f65f	5	10	0	0	0	0	0	0	0

복원 결과



| 데이터 전처리

파티 데이터 전처리

1. 파티의 인원수를 구한다.
2. 파티의 유지시간을 분 단위로 구한다.
3. 파티 유지 시간이 지나치게 짧은 경우 실수로 파티를 만들었거나, 무의미한 정보로 판단.
4. 각 파티의 hash를 풀어서 각각의 acc_id에 대한 정보로 만든다.
5. 각 acc_id 에 대해서 파티 생성 주, 주간 누적 파티시간,
주간 누적 파티원 수, 주간 누적 파티횟수, 평균 파티원 수를 구함.

| 데이터 전처리

party_start	party_start	party_start	party_end	party_end	party_end	hashed
1	1	42:01.1	1	1	11:28.5	633e2b44f
3	3	05:05.2	3	3	07:42.5	7176c151f
3	6	18:43.2	3	6	28:58.2	8092e194a
6	1	12:30.4	6	1	31:51.9	4b33f0b69
6	5	58:28.8	6	5	01:05.3	a284744f3
8	1	14:37.6	8	1	14:37.6	6015d440c
8	5	52:02.5	8	5	28:25.3	37d8c7e0c
4	1	26:03.4	4	1	37:05.3	bb8e63d2f
4	3	17:17.8	4	3	17:52.0	86fdd61ek
6	7	32:23.4	6	7	32:28.6	e7c5009f3
6	7	25:13.9	6	7	40:34.9	8a126a151
7	2	19:33.9	7	2	21:08.5	b1e14188a
7	6	13:18.3	7	6	17:28.0	2c0d64cec
8	1	26:42.2	8	1	42:46.6	99e31d50f
2	1	34:31.0	2	1	03:15.6	7a9f5ee03
4	3	17:39.6	4	3	30:32.5	28822df10
6	6	24:54.2	6	6	47:32.3	22f366abb
3	3	27:24.9	3	3	04:54.7	eb58b5d8f
5	1	39:55.9	5	1	05:29.2	0a4efa277
6	1	13:20.9	6	1	05:03.7	286fbcee7
6	1	23:31.4	6	1	41:40.5	32779c0dl



acc_id	time	member_c	party_cour	mean_mer	wk
000020b4f	4099.1	282	47	6	1
000020b4f	11748.9	818	129	6.341085	2
000020b4f	5860.5	672	110	6.109091	3
000020b4f	2763.4	744	124	6	4
000020b4f	11656.9	720	118	6.101695	5
000020b4f	5990.5	720	120	6	6
000020b4f	5798.6	714	118	6.050847	7
000020b4f	7246.9	516	85	6.070588	8
0000264b0	371.2	83	12	6.916667	7
0000264b0	708.2	184	24	7.666667	8
000069166	13.1	9	1	9	3
0000988c5	600.9	192	32	6	8
0000a2ff9	99.6	22	4	5.5	7
0000a2ff9	296.8	92	14	6.571429	8
0000d357f	6431.7	737	122	6.040984	5
0000d357f	6401.8	912	152	6	6
0000d357f	11328.4	750	122	6.147541	7
0000d357f	3874.1	306	51	6	8
0000d571f	242.2	43	6	7.166667	8

| 데이터 전처리

거래 데이터 전처리

각 acc_id 별로

주간 누적 건넨 재화량, 받은 재화량을 구한다.

기존 거래 데이터와 누적 거래량을 근거로

이상치를 발견하여 군집화 할 수 있음.

acc_id	give_money	rec_money	give_grocery	rec_grocery	wk
0000264b	200.457	0	521	10	7
0000264b	400.0039	0	996	20	8
0000548b	75.9711	0	154	0	6
00006916	306.8733	0	876	3	2
0000d571	99.9681	0	264	5	8
0001f130e	30100	0	0	0	7
00027152	54.9217	0	114	0	3
00027152	77.8348	0	61	0	4
000291b3	0	0	0	384	5
0002cb16	500	1780	0	0	1
0002cb16	88600	8919	0	15	2
0002cb16	5944	22890	0	0	3
0002cb16	1035000	111500	2	10	4
0002cb16	37000	0	0	0	5
00036e5b	61.0854	0	87	0	1
00036e5b	44.8744	0	142	0	3
00036e5b	52.0625	0	135	0	4
00036e5b	8.0167	0	21	0	6
00036e5b	39.1184	0	125	0	7
00036e5b	37.82	0	102	0	8
00038d64	39.2363	0	32	0	1

| 데이터
통합



| 데이터 통합

데이터 전처리 결과 얻은 변수 목록

Party_time : 누적 파티 유지시간

Member_count : 누적 파티원 수

Party_count : 누적 파티 횟수

Mean_mem : 파티 1회당 평균 파티원 수

Give_money : 누적 건넨 금화량

Rec_money : 누적 받은 금화량

Give_grocery : 누적 건넨 잡화량

Rec_grocery : 누적 받은 잡화량

Is_guild : 길드 가입 유무

Payment : 누적 캐쉬 구매

모든 설명변수들을 wk를 기준으로
더미화한다.

wk값이 1이면 '설명변수_1'

2이면 '설명변수_2'

| 데이터 통합

그저 더미화만 했을 때는 성능이 좋지 않았다.

원본 데이터의 경우 마지막 주인 8주를 기준으로 정렬

왜 모두가 다른 정보를 가진 마지막 주일까?

게임을 시작한 주가 기준이 되어야 한다!

각 계정별로 시작한 주를 1주로 보고 시작 주 기준 정렬한다.

시작 주로 매긴 번호가 같은 것들을

하나의 설명변수 클래스로 묶는다. (time_wk=시작 주 번호)

wk	acc_id	time_wk
1	0001f130e	1
2	0001f130e	2
3	0001f130e	3
4	0001f130e	4
5	0001f130e	5
6	0001f130e	6
7	0001f130e	7
8	0001f130e	8
3	00028f0ad	1
4	00028f0ad	2
5	00028f0ad	3
7	00028f0ad	4
8	00028f0ad	5
1	00036e5b0	1
3	00036e5b0	2
4	00036e5b0	3
6	00036e5b0	4
7	00036e5b0	5
8	00036e5b0	6
2	00037fe8e	1
8	00037fe8e	2

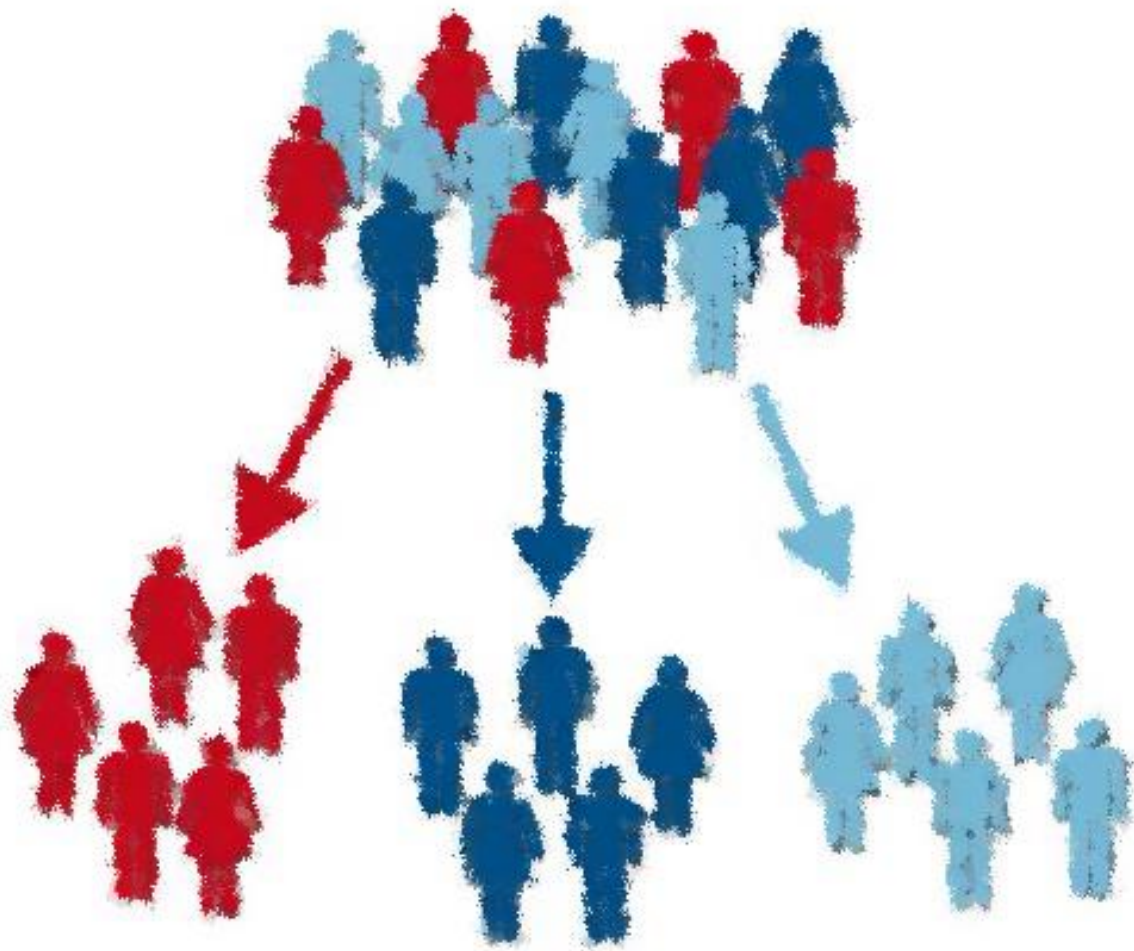
| 데이터 통합

wk	acc_id	time_wk
1	0001f130e	1
2	0001f130e	2
3	0001f130e	3
4	0001f130e	4
5	0001f130e	5
6	0001f130e	6
7	0001f130e	7
8	0001f130e	8
3	00028f0ad	1
4	00028f0ad	2
5	00028f0ad	3
7	00028f0ad	4
8	00028f0ad	5
1	00036e5b0	1
3	00036e5b0	2
4	00036e5b0	3
6	00036e5b0	4
7	00036e5b0	5
8	00036e5b0	6
2	00037fe8e	1
8	00037fe8e	2



acc_id	변수_wk							
0001f130e	1	2	3	4	5	6	7	8
00028f0ad	1	2	3	4	5			
00036e5b0	1	2	3	4	5	6		
00037fe8e	1	2						

| 데이터
군집화



| 데이터 군집화

trade_wee	trade_day	trade_time	source_acc	target_acc	item_type	item_amou
1	1	0:00:00	3c3baeef1	f9fd4dae9	grocery	2
1	1	0:00:00	3c3baeef1	f9fd4dae9	money	28.5122
1	1	0:00:00	a4785c1f4	1bccfbc74	money	220
1	1	0:00:02	918c5389e	065a4d4a8	money	200
1	1	0:00:02	9c8c6cf69	0eb20d13	grocery	1
1	1	0:00:02	9c8c6cf69	0eb20d13	money	28.5134
1	1	0:00:02	adf064d97	d19edcf09	grocery	3
1	1	0:00:02	adf064d97	d19edcf09	money	41.4474
1	1	0:00:03	462a3599c	9cf88c33a	money	991.76
1	1	0:00:03	eebbe9df8	9689eb21	grocery	29
1	1	0:00:03	eebbe9df8	9689eb21	money	58.9492
1	1	0:00:04	0fed9cb7b	107a6731	grocery	2
1	1	0:00:04	0fed9cb7b	107a6731	money	28.5183
1	1	0:00:04	2787d157f	da9d8976	money	220
1	1	0:00:05	ea5327e54	f9fd4dae9	grocery	3
1	1	0:00:05	ea5327e54	f9fd4dae9	money	25.5104

복원된 거래 데이터를 보면 특이점이 있다.

일반적 거래 - 물건을 주고 돈을 받는다.

이상한 거래 - 물건도 주고 돈도 준다.

Train, Test 통합

총 6,848,159 건 중

2,871,734 건(41.93%)이 이상한 거래

| 데이터 군집화

전체 41.93%의 대규모의 이상한 거래에 주목.

이 거래는 다수의 Source_acc_id가 소수의 Target_acc_id 에게 재화를 공급하고 있으며, 일종의 조직적인 행동으로 보였다.

실제 블레이드&소울 내에는 매크로를 사용해 재화를 수집하는 상당히 많은 Auto 계정이 존재하기 때문에, 우리는 이들을 **Auto 계정으로 추측**.

이상한 거래의 Source_acc_id 수 : 106,926개, Target_acc_id 수 : 9,897개

(교집합 제거 시 101,491개, 9,897개)



| 데이터 군집화

전체 Activity 데이터 대조 결과

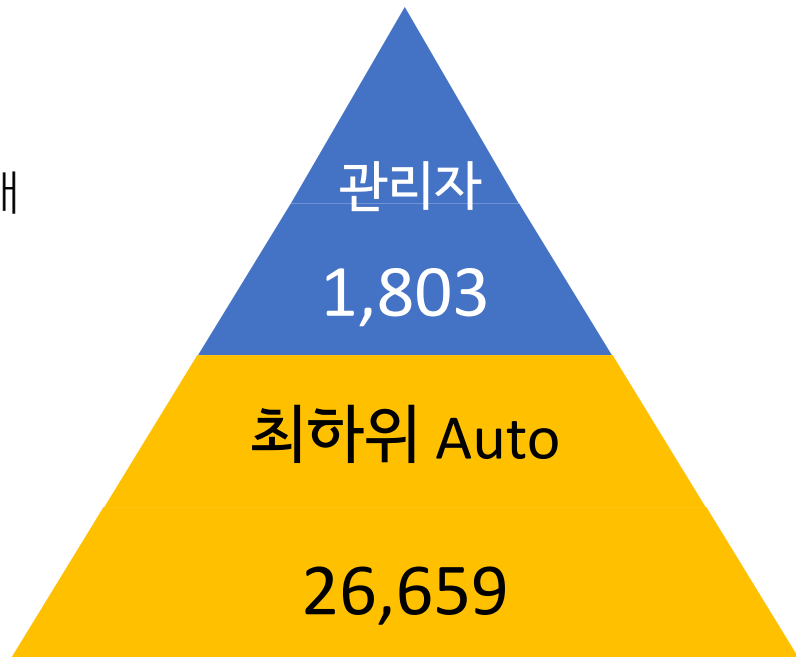
Source_acc_id : 32,710개, Target_acc_id : 6,292개

Auto가 절대 하지 않는 행동

- 레이드, 숙련 인던, 문파 가입, 제작 활동, 캐쉬 결제

Auto의 목적

- 매크로를 통한 Auto관리자 계층의 재화 수집



해당 조건으로 걸러낸 결과 - 최하층 Auto : 26,659개, Auto관리자 : 1,803개

| 데이터 군집화

최하층 Auto : 26,659개, Auto관리자 : 1,803개

Train, Test 분할 시 Auto 계정 개수

Train : 최하위 18,957개, 관리자 1,278개

Test : 최하위 7,702개, 관리자 525개



일반인 유저와 다른 행동 구조를 가지고 있음 -> 군집화 결정

| 데이터 군집화

3,083 X 6 = 18,498 !!!

거래 데이터에서 Auto가 아닌 또 다른 이상한 군집을 발견!

1	1	00:01:48	204e781dbbc1a0eba5ff0b20c7a0599c0b2e03dca9231d...	28cc223796fb957882941742506a35af7007d47ecb63cc...	money	<u>18498.0000</u>
1	1	00:01:48	66adab08748877c55993041a97f0c5353c10d8e568455e...	e93fbd0f6ec46650396bb3312eed395e37885ca29ba03c...	money	3000.0000
1	1	00:01:51	19b88d7a5c3d4fb2707bd04c024609b7bcf50015b79188...	c29a3cce691ded220e2c069aa32011d0cb8a30914ba6a5...	money	44000.0001
1	1	00:02:02	28cc223796fb957882941742506a35af7007d47ecb63cc...	fa0750ca411485255897c81e55c2a6964fa8ff7e30ef07...	money	3083.0000
1	1	00:02:15	204e781dbbc1a0eba5ff0b20c7a0599c0b2e03dca9231d...	7d993cd5c5d9b4500f6154aba29c6ce27b8308b729549f...	money	3083.0000
1	1	00:02:15	28cc223796fb957882941742506a35af7007d47ecb63cc...	46efb0f8b0439b7822d5184243e4e30a69314a8af52181...	money	3083.0000
1	1	00:02:23	204e781dbbc1a0eba5ff0b20c7a0599c0b2e03dca9231d...	b67e95e7108bd312eb6ab40ba6c7c8002571fefb5d6d78...	money	3083.0000
1	1	00:02:27	28cc223796fb957882941742506a35af7007d47ecb63cc...	1e92c4c9b2dfaa0a97ef38633603db748f009e06236569...	money	3083.0000

금화를 같은 금액으로 나눠가지는 그룹이 있다.

- 값비싼 재화를 획득 한 이익금 분배
- 버스기사의 이익금 분배
- 레이드 이익금의 분배

| 데이터 군집화

이 그룹의 의미는?

1. 값비싼 재화를 얻을 능력이 있다.
2. 레이드를 본인 능력으로 클리어 할 수 있다.
3. 버스기사 활동을 할 수 있다.

게임을 열심히 하는
충성 고객일 것이다!



| 데이터 군집화

반복되는 168과 8,610

다른 데이터는 비어있다!

이들의 의미는 무엇인가?

acc_id	label	cnt_dt_1	play_time	npc_exp_1	npc_hongr	quest_exp	quest_hon	item_hong	game_con	get_mone
0004733c	month	5	4.8	0	0	0	0	0	0	0
00069cf4	d2month	2	23	168	0	8610	0	0	2.4	0
0008086e	cweek	1	617.4	180396	0	713460	0	0	121.4	0.3066
00107e6b	d2month	2	20	168	0	8610	0	0	1.8	0
00141e39	d2month	4	4911.7	1127304	28382	9991610	16556000	0	1396.7	4.1087
001f47b2	d4week	2	26.8	0	0	0	0	0	0	0
001fd0fb	d4month	2	18.6	168	0	8610	0	0	1.8	0
0024f591	cmonth	2	0.5	0	0	0	0	0	0	0
0027bf07	amonth	1	2.7	0	0	0	0	0	0	0
002acc8a	e2month	2	20.2	168	0	8610	0	0	1.8	0
002c2826	cmonth	5	28.7	168	0	8610	0	0	1.9	0
0030e33d	d2month	1	87	11784	0	38550	0	0	14.1	0.0085
00352b2b	bweek	7	5894.2	343122	637372	4298120	11091990	1160000	2759.8	6.2047
00388013	d2month	1	18.3	168	0	8610	0	0	1.4	0
003a284a	cweek	3	3688.7	697764	20053	6625000	16464630	0	833.6	2.3167
003b24ef	cweek	5	4123.2	337668	516077	4298120	11106070	940000	1640.7	5.3156
003e814f	e2month	6	6	0	0	0	0	0	0	0
00458006	e2month	3	25.2	168	0	8610	0	0	1.8	0
0047c869	cmonth	2	18.3	168	0	8610	0	0	1.9	0
004888a0	d7week	4	3847.2	348486	257345	4298120	11064630	750000	1258	169.9031
004970fe	e2week	1	10.2	0	0	0	1700	0	0.3	0
004ba8c1	fmonth	3	732.2	240690	0	829340	0	0	136.6	0.4802

| 데이터 군집화

튜토리얼 경험치 총량 = 8,610

대부분이 이후 아무런 활동이 없다.

튜토리얼만 완료한

Dummy 계정으로 판단

학습에 있어 심각한 노이즈가 됨



| 데이터 군집화

acc_id	label	cnt_dt_1	play_time	npc_exp_1	npc_hong	quest_exp	quest_hon
076144fbc	month	2	2.2	0	0	0	0
07634510f	month	2	106.2	0	44871	0	12900
0764d7a3e	week	3	3221.3	759660	12112	5982420	8779030
07686c8ae	month	4	23.4	168	0	8610	0
076b3c79f	2month	4	3.5	0	0	0	4500000
076e27b0f	2month	4	3.5	0	0	0	0
076f06153	week	3	3657.8	667914	22369	5982420	15441910
0770cbc2f	2month	4	3.5	0	0	0	0
0775ebb6f	month	5	31.7	168	0	8610	0
07761509f	2month	6	48.2	336	0	17220	0
0776a2abf	2month	5	23.8	168	0	8610	0
077aa340c	retained	2	20.5	168	0	8610	0
077c42e7c	month	6	6.1	0	0	0	0
0781023af	month	4	6.5	0	0	0	0
0782f5a40	2month	4	24.4	168	0	8610	0
07830d49f	week	1	128.8	26136	0	62270	0
0785ed69f	week	2	36.8	0	0	0	4500000
0786a059f	2month	6	36.7	168	0	8610	0

퀘스트 홍문경험치 획득량 = 4,500,000



-> 고속성장캐릭터

| 데이터 군집화

크게 3개의 군집으로 군집화 하였다.

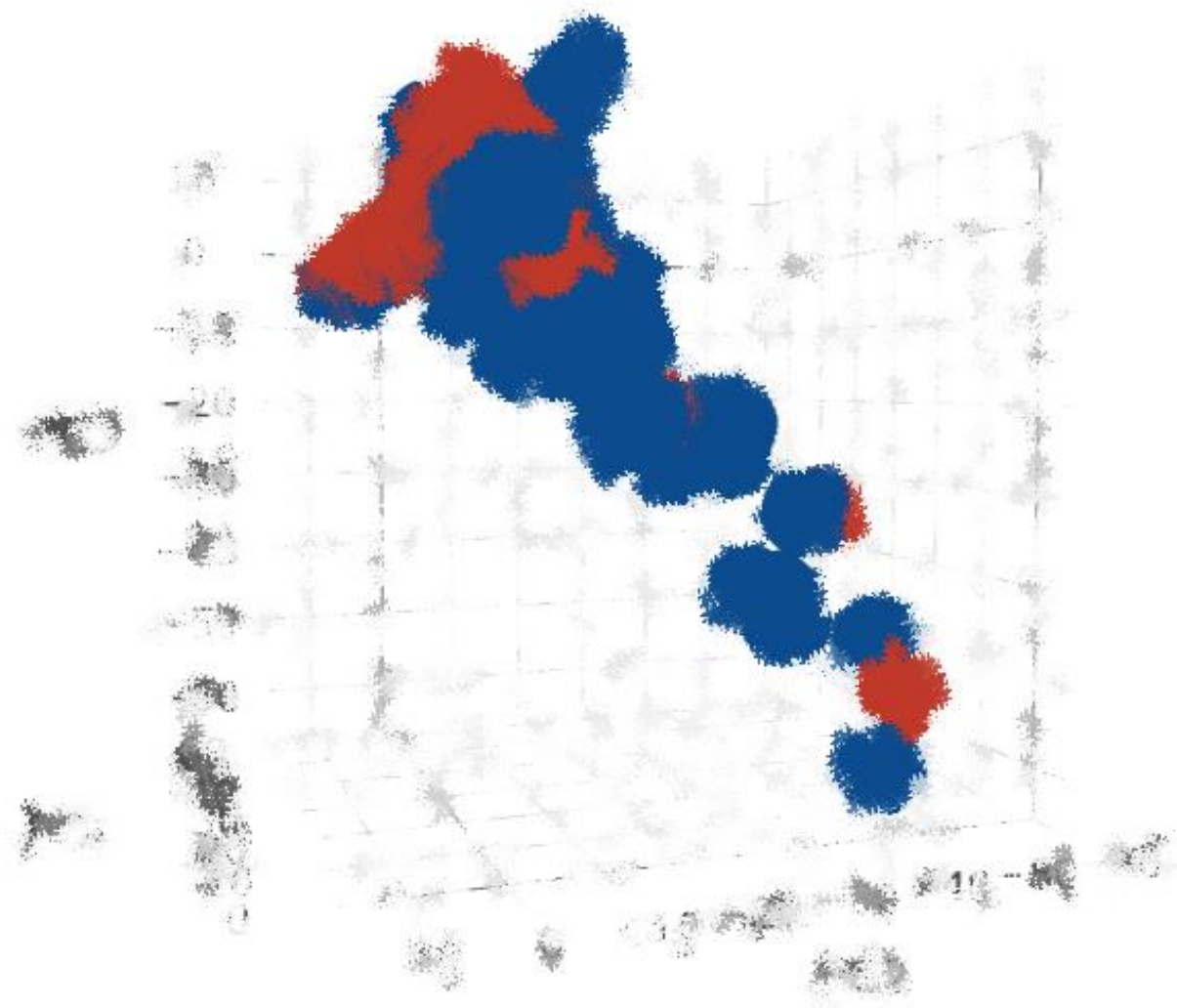
- Auto, Person, Dummy_8610

총 8개의 군집

- ant_id : 최하위 Auto
- master_id : Auto 관리자
- Auto_id : 통합 Auto
- Normal_id : 일반인
- Hard_id : 게임에 적응한
충성고객
- Person_id : 모든 사람 계정
- 8610_id : 튜토리얼 클리어 계정
- Jump_id : 고속성장캐릭터 계정

acc_id	ant_id	master_id	auto_id	normal_id	hard_id	person_id	8610_id	jump_id
0000264b0	1	0	1	0	0	0	0	0
0001f130e	0	0	0	1	0	1	0	0
00028f0ad	0	0	0	0	0	0	1	0
0002a56a0	0	0	0	1	0	1	0	1
00036e5b0	1	0	1	0	0	0	0	0
00037fe8e	0	0	0	0	0	0	1	0
0003b127a	1	0	1	0	0	0	0	0
0003bfdek	0	0	0	1	0	1	0	0
0004733c4	0	0	0	1	0	1	0	0
00047f658	0	0	0	0	0	0	1	0
0004a2a2f	0	0	0	0	0	0	1	0
000650214	1	0	1	0	0	0	0	0
00069cf4d	0	0	0	0	0	0	1	0
0007838a3	0	0	0	1	0	1	0	0
0007fbe4b	0	0	0	1	0	1	0	0
0008086e0	0	0	0	1	0	1	0	0
0008c8c21	0	0	0	1	0	1	0	1
0008f5dce	0	0	0	0	0	0	1	0
000a3b32c	0	0	0	1	0	1	0	0
000a6998e	0	0	0	1	0	1	0	0
000c248e3	0	0	0	1	0	1	0	0

| Modeling



| Modeling - Process

통합데이터

통합 - (10000, 227)

Person/
Auto
분할데이터

Person - (79834,
227)
Auto - (20166, 227)

Person/
Auto
분할데이터
+
Feature
추가

Person - (79834,
585)
Auto - (20166, 585)

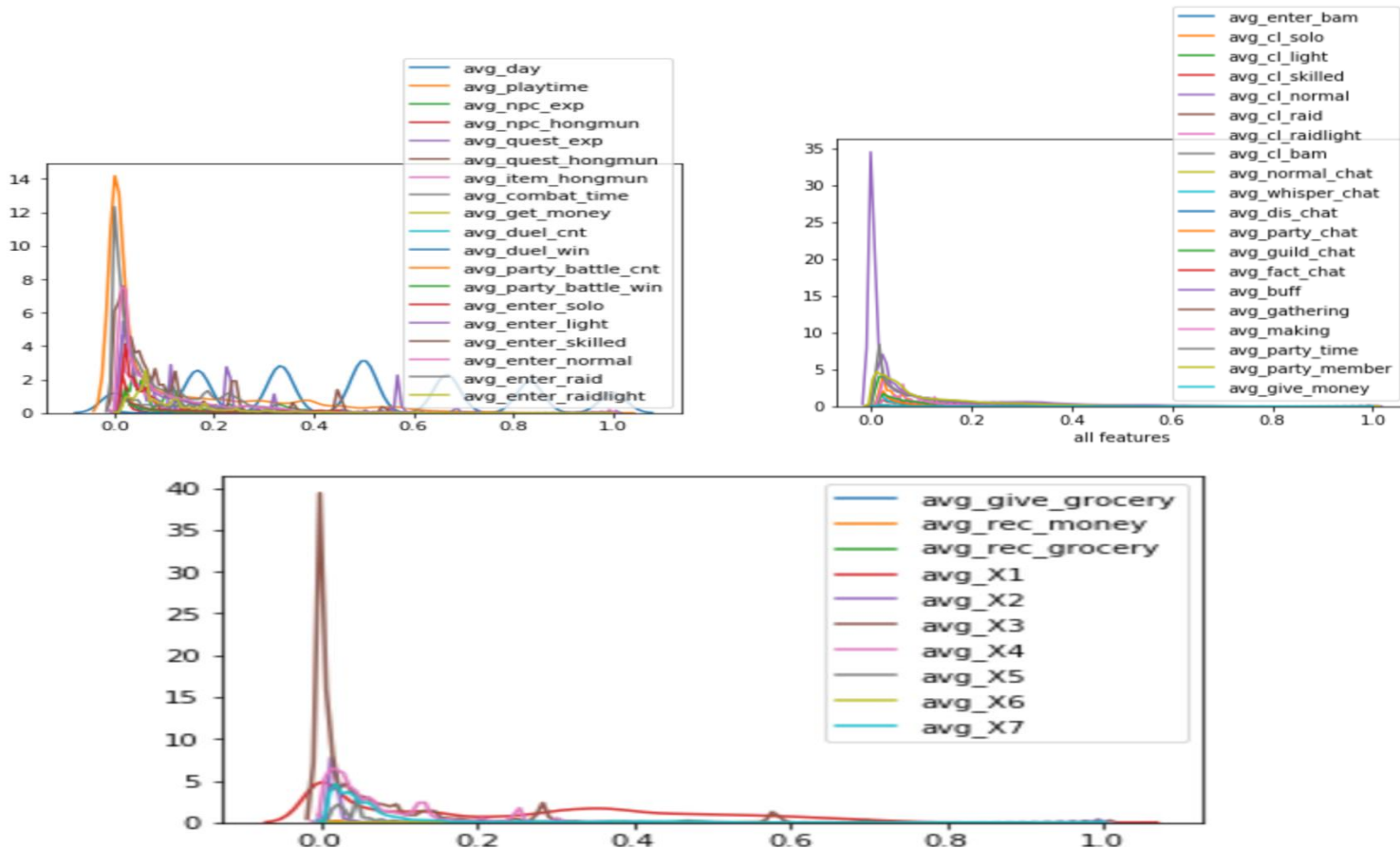
Person/
Auto
분할데이터
+
Feature
추가
-
8610

Person - (62665,
585)
Auto - (20166, 585)

하이퍼
파라미터
튜닝

| Modeling - Feature의 분포

게임 유저의 특성상 다음 그림과 같이 매우 Skewed한 분포를 보임.



| Modeling - Model Sketch

모든 classifier 모형에 대한 성능 평가

모델명	parameter	성능	순위
Random Forest	Default	0.715043730412898	1
Bagging	Default	0.710735657695165	2
XGBoost	Default	0.6884774133054054	3
KNN	Default	0.6758639327415544	4
Support Vector Machine	Gaussian Kernel	0.67545532415544	5
Multiple Layer Perception	3 Hidden Layer Adam Optimaizer Dropout Early Stopping Batch size = 10000 Epoch = 500	0.65966666	6
LDA	Default	0.6056211552482658	7
QDA	Default	0.47673299167763794	8
Logistic Regression	Default	0.47617197756021656	9
Naive Bayes	Default	0.4273648571416887	10

변수는 더미화 변수를 포함하여 585개
(100000, 585)

Train : Validation = 7 : 3

Decision Tree 기반으로 한 앙상블 모형의 성능이 대체로
좋다.

변수의 분포가 대체로 Skewed 한 것이 원인이라고 예상

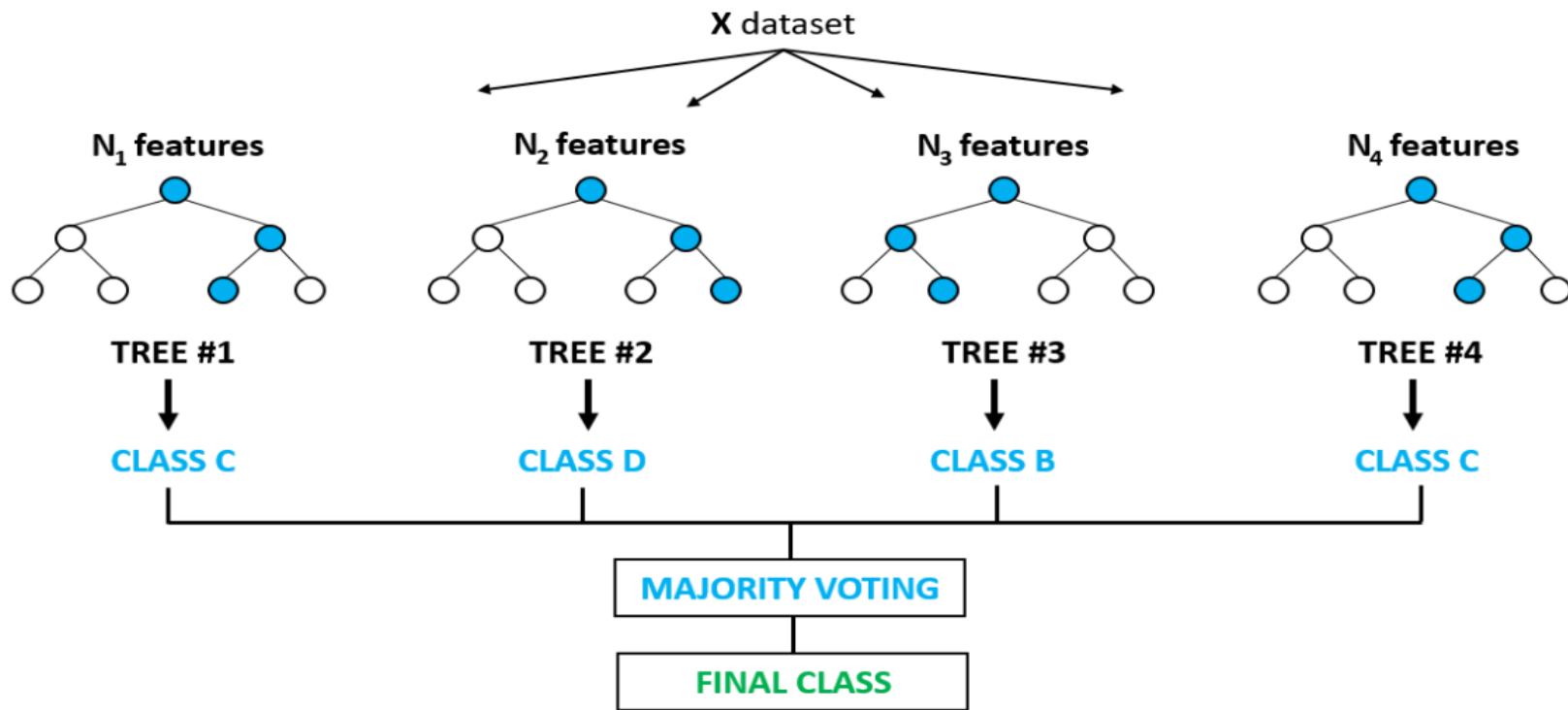
KNN의 성능이 좋은 것도 흥미롭다.

Random Forest를 기본 모델로 선정

| Modeling - Random Forest

Random Forest :

여러 개의 랜덤으로 선택된 의사결정트리가 모여 하나의 숲을 이룬다는 개념



| Modeling - Random Forest

Random Forest의 알고리즘

1. 주어진 train data에 대하여 복원 추출법으로 n 개를 선택
2. 선택한 n 개의 데이터 샘플에서 Feature를 비복원 추출법으로 d 개를 선택
(d 는 Sklearn에서 `max_features`)
3. 이렇게 구성된 데이터 셋을 통해 Decision Tree를 학습하고 생성함.
(버피팅을 방지하기 위해 `max_depth`를 통해 깊이가 깊어지는 것을 방지)
4. 1 ~ 3번 단계를 k 번 반복함. (k 는 `n_estimator`)
5. 최종적으로 k 개의 Decision Tree를 이용하여 예측하고 예측의 결과에 대해 Voting을 진행하여 가장 많이 뽑힌 것을 최종 예측결과로 할당.

| Modeling - 하이퍼 파라미터 튜닝

튜닝 규칙

1. Seed값을 고정한다. 동일한 규칙을 통해 성능이 좋아지는 것을 확인하기 위해서이다.
2. 성능에 가장 큰 영향을 미치는 `n_estimator`(2에서 설명한)의 가장 좋은 값을 찾는다.
3. 가장 좋은 `n_estimator`에 대하여 과대적합 방지를 위한 `tree`의 깊이인 `max_depth`의 적절한 값을 찾아 오버피팅을 방지한다.
4. 2,3번을 통해 결정한 `n_estimator`와 `max_depth`를 fix시켜 몇 개의 Feature를 선택할 것인지에 대한 `max_features`(2에서 설명한)의 Best tune을 찾는다.
5. 마지막으로 가지를 분할하는 데 필요한 최소 샘플 수인 `min_samples_split`, 각 가지에서 필요한 최소 샘플 수인 `min_samples_leaf`를 결정하여 최종 파라미터를 결정한다.

| Modeling - 하이퍼 파라미터 튜닝

1777	10	auto	0.7346516370076865	
1833	10	auto	0.7347312583359263	
1888	10	auto	0.7346781520200154	
1944	10	auto	0.7348209427485223	Best
2000	10	auto	0.7347995950679171	

1. n_estimator 튜닝

1950	10	75	10	0.7375401205558921	
1950	10	75	15	0.7383601364669425	Best
1950	10	75	20	0.7368171421337554	
1950	10	75	30	0.7350016200050617	

3. min_samples_split 튜닝

1950	10	58	0.7356065428985181	
1950	10	63	0.7358525987530192	
1950	10	69	0.7361901560522929	
1950	10	75	0.7362587985536095	Best
1950	10	80	0.7358568939804893	
1950	10	86	0.735798295436545	
1950	10	92	0.7357331045459541	

2. max_features 튜닝

1950	10	75	15	1	0.7383601364669425	
1950	10	75	15	2	0.7391481800750069	Best
1950	10	75	15	3	0.7373818247414129	
1950	10	75	15	4	0.7360107050000017	

4. min_samples_leaf 튜닝

최종적으로 선정된 Hyper Parameter(bootstrap samples를 하지 않았을 때)

Parameter 이름						
n_estimator	max_depth	max_features	min_samples_split	min_samples_leaf	성능	default와 성능 비교
1950	10	75	15	2	0.7430061331827581	0.0279624027698601

| Modeling - Result_1

데이터	통합	2분할		2분할 + feature		2분할 + feature - 8610	
구분	Origin	Person	Auto	Person	Auto	Person	Auto
F1 Score (#데이터)	0.7225 (100000)	0.6899 (79834)	0.8584 (20166)	0.694063 (79834)	0.863746 (20166)	0.737460 (62665)	0.866259 (20166)
총	0.7225	0.726211		0.730066		0.772825	

다음은 각 데이터에 대한

Random Forest(n_estimator = 300) F1 Score

서로 다른 분포를 가지는 Person과 Auto를 나눌 경우, 새로운 변수를 추가할 경우 성능이 향상되는 것을 볼 수 있었다.

레이블에 대한 랜덤 분포를 가지는 8610데이터를 없앨 경우 성능이 크게 향상 되는 것을 볼 수 있었다.

성능을 최대한으로 올리기 위해 8610을 제외한 데이터 모델에 대한 하이퍼 파라미터 튜닝을 진행하였다.

| Modeling - Result_2 after Tuning

Person데이터에 대한 최종 Hyper Parameter

Parameter 이름						
n_estimator	max_depth	max_features	min_samples_split	min_samples_leaf	성능	n_estimator = 300과 비교
1269	30	24	2	1	0.761228	0.023767

8610데이터는 우리가 예측할 대상이 아니라고 판단하였다.

Auto 데이터에 대한 최종 HyperParameter(False)

Parameter 이름						
n_estimator	max_depth	max_features	min_samples_split	min_samples_leaf	성능	n_estimator = 300과 비교
1421	50	24	2	2	0.8706608	0.004401

8610을 제외한 Person, Auto 데이터에 대해 각각 하이퍼파라미터 튜닝을 진행하였다.

튜닝 결과, 각각 모두 F스코어가 향상된 모습을 보였다.

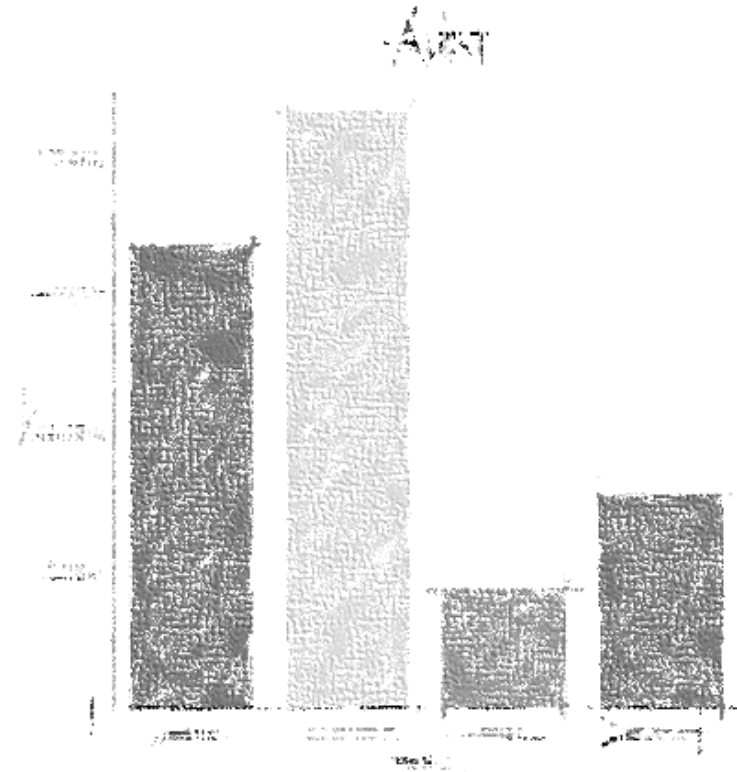
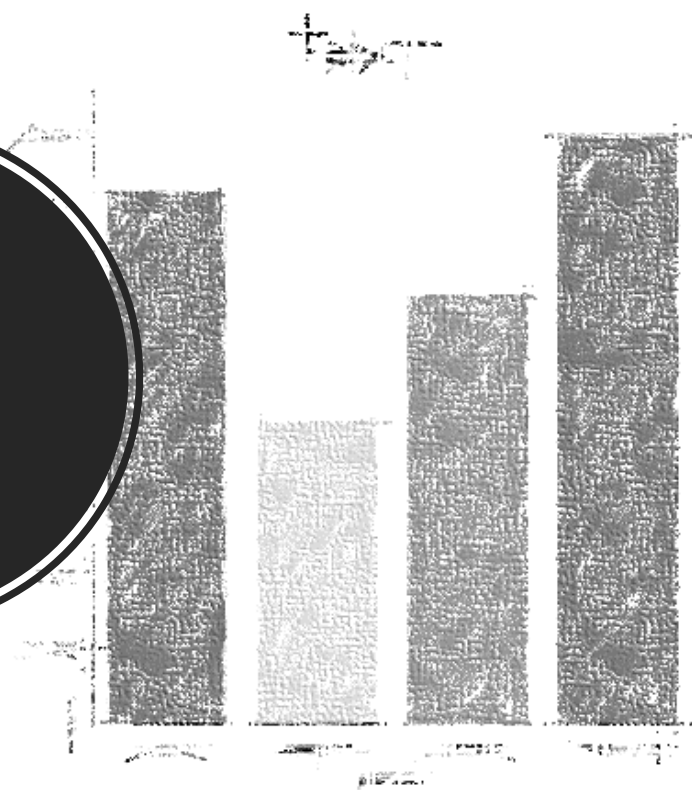
2분할 + feature - 8610	
Person	Auto
0.737460 (62665)	0.866259 (20166)
0.772825	

하이퍼 파라미터 튜닝 후	
2분할 + feature - 8610	
Person	Auto
0.76123 (62665)	0.87066 (20166)
0.77152	

다만 두개를 합친 결과 튜닝 전보다 F스코어가 낮았다.

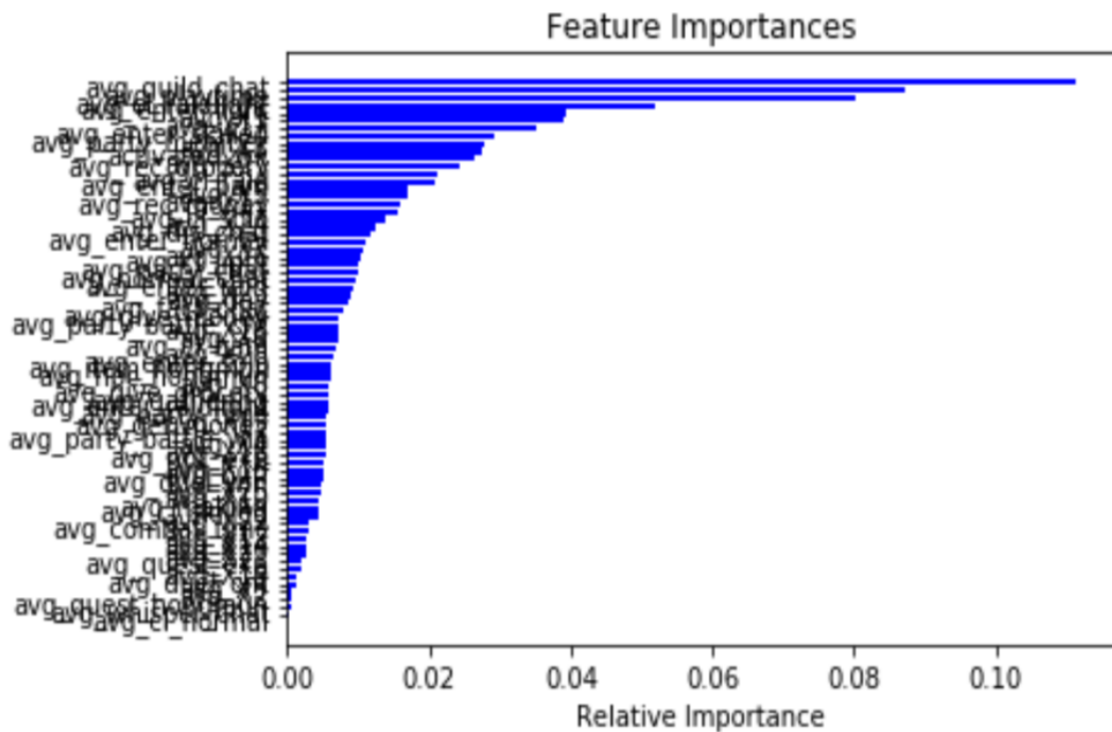
제대로 데이터를 분할하고, 각각에 대하여 모델을 적합시키고 해당하는 데이터를 예측하는 것이 좋을 것으로 판단된다.

| 분석



| Analysis - Retained&Non-retained

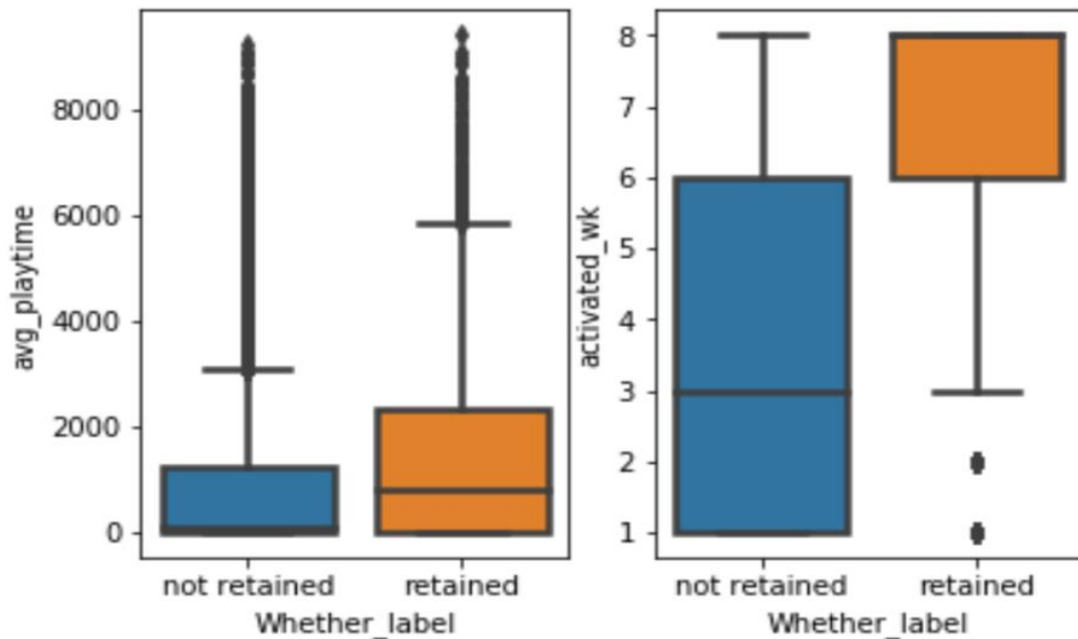
RandomForest의 Feature Importance 그래프



변수명	중요도
avg_playtime	0.1113369430490359
activated_wk	0.08707159929740305
avg_guild_chat	0.0803576674567623
avg_X5	0.05197654680508853
avg_item_hongmun	0.039308144815832
avg_give_money	0.038970995850017745
avg_X23	0.03505011279980037
avg_X1	0.029100588584096688
avg_cl_raid	0.02766373301629841
avg_day	0.02757783464320575
avg_rec_money	0.026568658354712126
avg_X19	0.02449535394498176

| Analysis - Retained&Non-retained

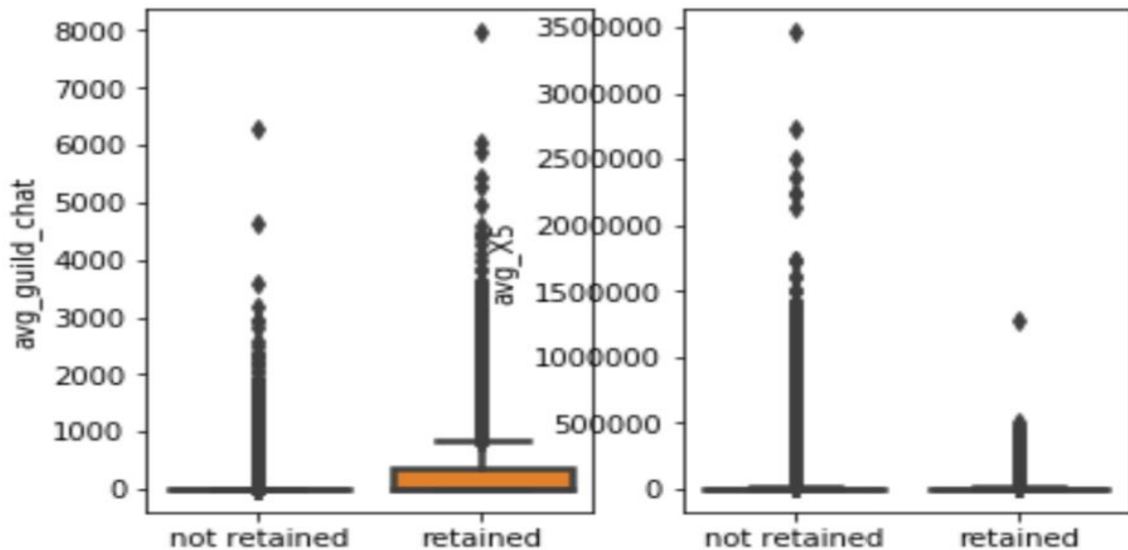
Playtime & activated week



- ▶ 비이탈자의 경우 확연하게 이탈자보다 접속시간이 길다.
- ▶ 마찬가지로 비이탈자의 경우 확연하게 이탈자보다 접속주가 많음을 볼 수 있다.

| Analysis - Retained&Non-retained

Guild_chat & 전체플레이시간 대비 경험치 합산

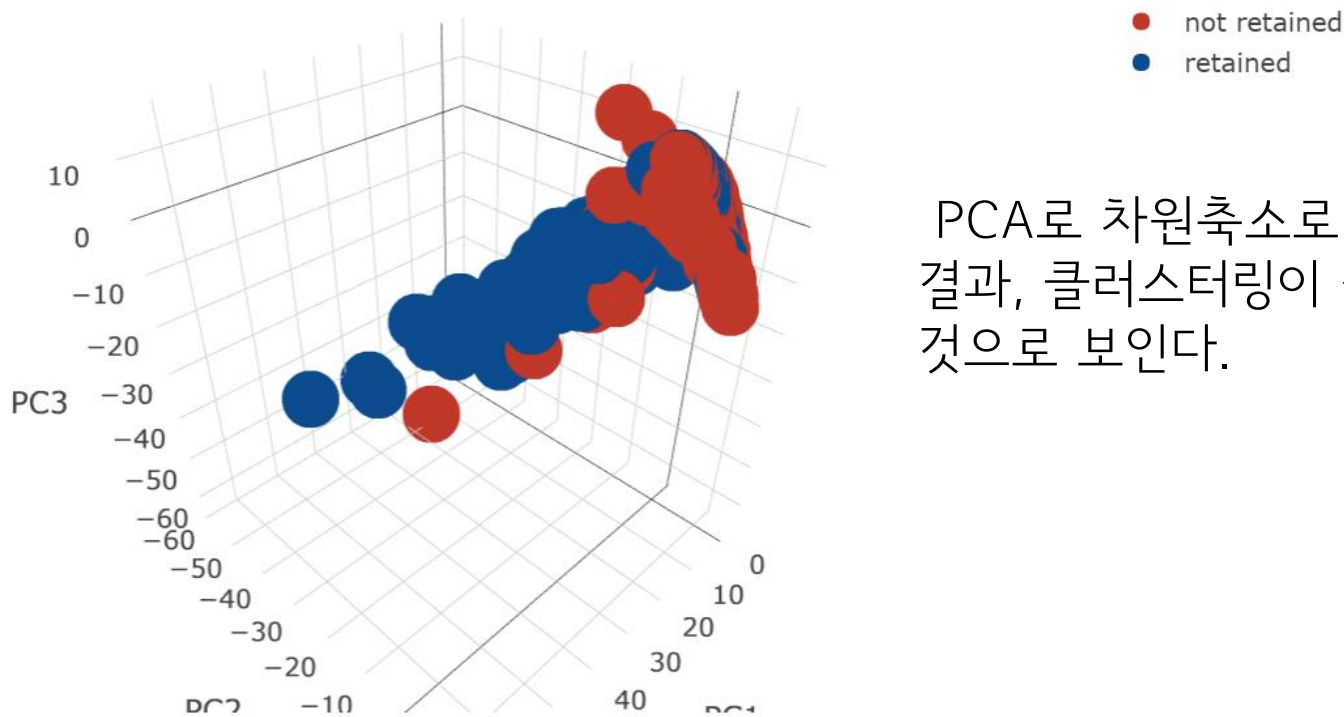


▶ 비이탈자의 경우 이탈자에 비해 길드 채팅이 많다. 따라서 길드의 가입여부 또한 둘을 나누는 중요 변수로 작용한다.

▶ 전체플레이시간 대비 모든 경험치의 경우 비이탈자들이 이상치가 더 많았는데, 이는 작업장 계정의 메크로 계정들로 보인다.

| Analysis - Retained&Non-retained

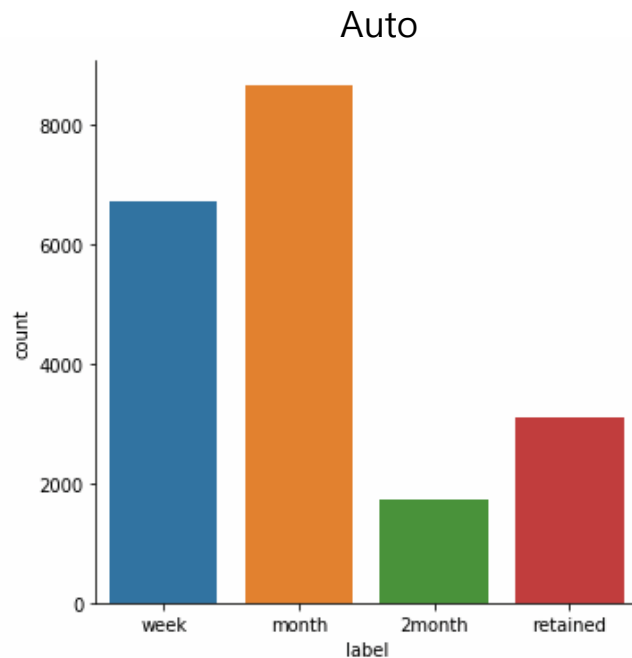
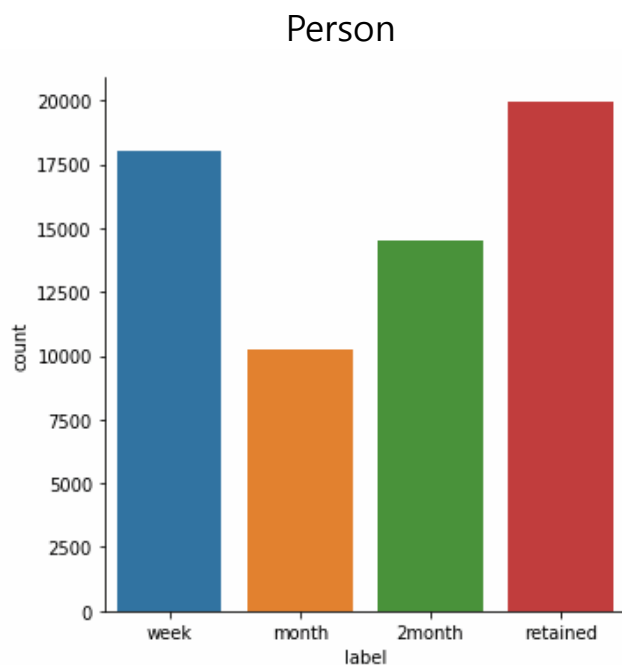
이탈자 & 비이탈자 PCA Clustering



PCA로 차원축소로 시각화를 한 결과, 클러스터링이 잘 이루어진 것으로 보인다.

**더 나아가,
군집화에 기반해
이탈자/비이탈자 특성을
살펴보자**

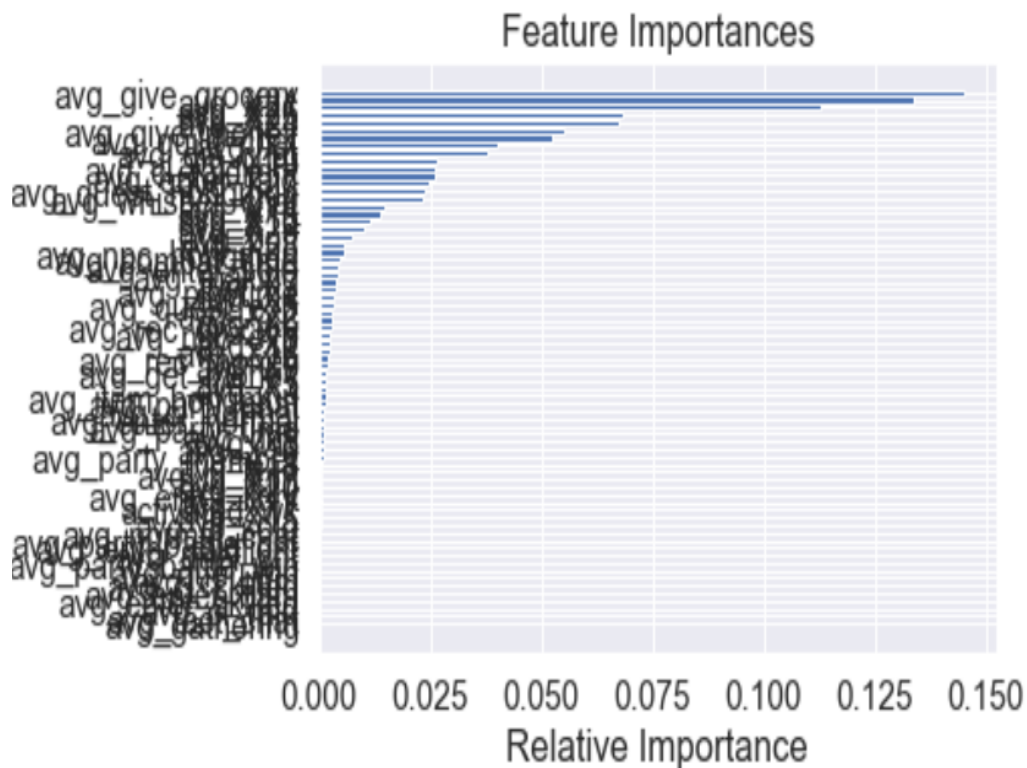
| Analysis - Person&Auto



- ▶ 일반유저의 경우 : 겉보기에는 label의 분포가 고르게 보인다. -> 과연 그럴까?
- ▶ 작업장의 경우 : week와 month의 비율이 높았는데 이는 자발적 이탈이 아닌 강제이탈(계정 정지)로 보인다.

| Analysis - Person&Auto

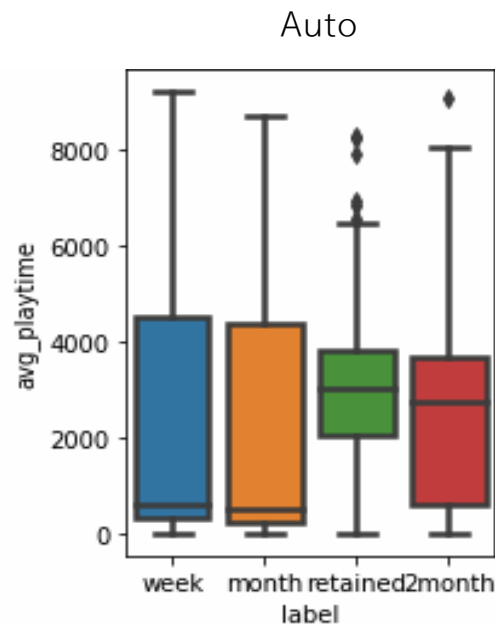
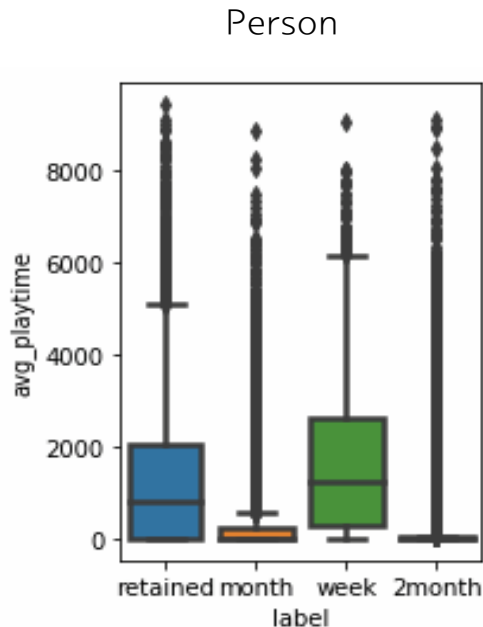
RandomForest의 Feature Importance 그래프



변수명	중요도
avg_give_grocery	0.14012217867177118
avg_X24	0.06766431423663191
avg_X26	0.06627846846462597
avg_enter_rai	0.06108786492190912
avg_X11	0.05734912086667064
avg_give_money	0.05638244938393592
avg_X25	0.045006769188937076
avg_X27	0.03483442534551179
avg_X1	0.027157299480545433
avg_cl_raidligh	0.023820408396156473
avg_whisper_chat	0.02341998439617976

| Analysis - Person&Auto

플레이시간

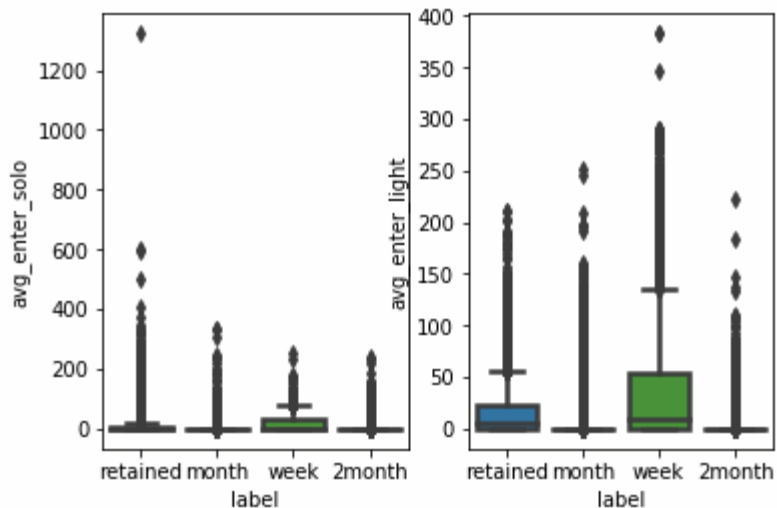


- ▶ 일반유저 : 개인 사정에 따라 오래 할 수도 있고, 짧게 할 수도 있음.
- ▶ 작업장 : 매크로이므로 라벨별 플레이타임이 균일함.

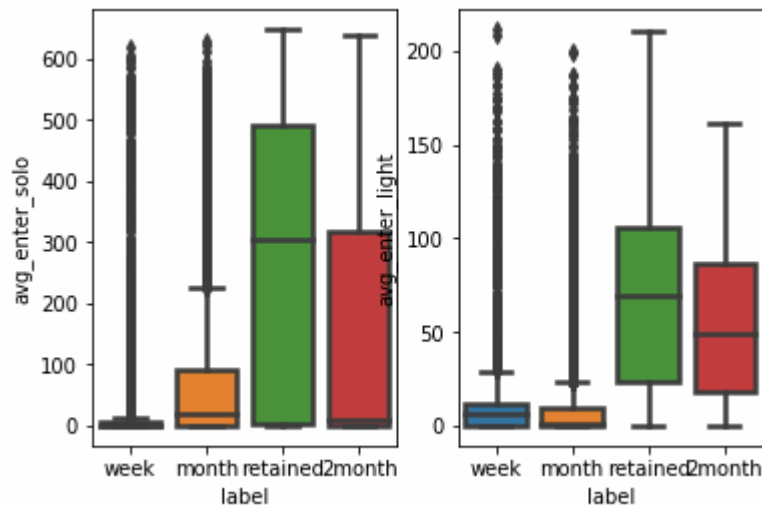
| Analysis - Person&Auto

Solo/light 인던 입장횟수

Person



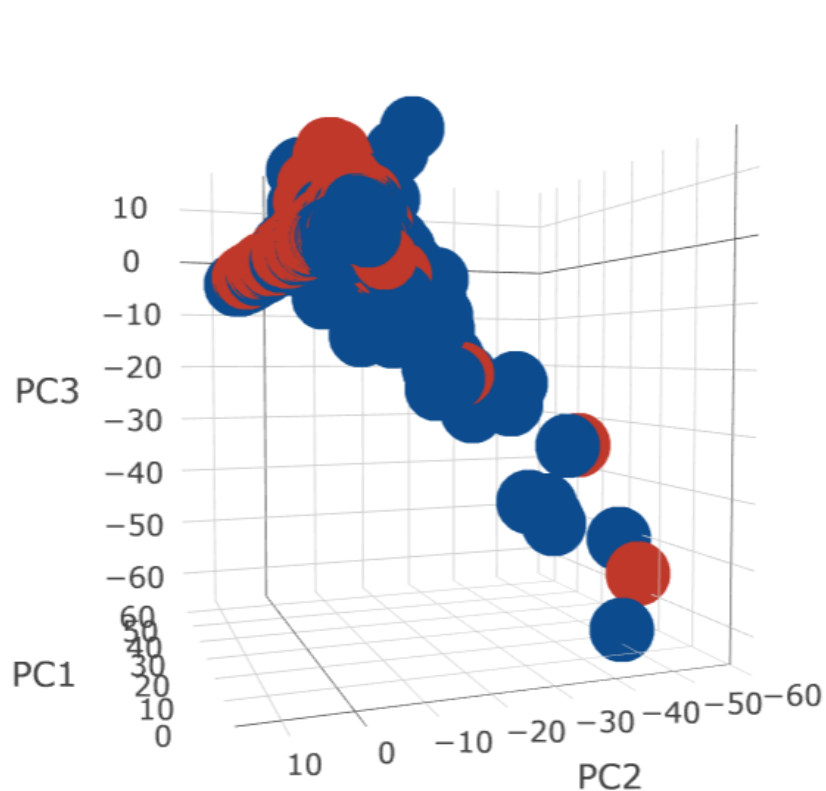
Auto



▶ 작업장의 경우 압도적인 Solo, Light 인던 수치를 보이는데 매크로를 사용하기 때문에 이러한 분포가 나타나는 것으로 생각된다.

| Analysis - Person&Auto

일반유저 & 작업장 PCA Clustering



● Auto
● Person

PCA로 차원축소로 시각화를 한 결과,
클러스터링이 잘 이루어진 것으로 보인다.

| Analysis - Person&Auto

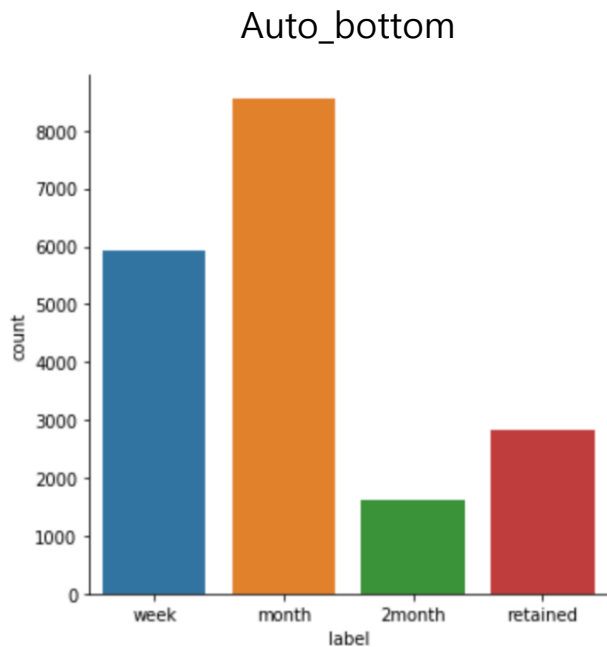
군집화에 기반해,
Auto와 Person을
각 각 분리해서
이탈 원인을 분석해보자

| Analysis - Auto

Auto는 ?



| Analysis - Auto



- ▶ 최하위 오토의 경우 : week, month 비율이 높다. 그리고 retained보다 non-retained가 월등히 높다. 이는 자발적 이탈이 아닌 강제이탈(계정 정지)로 보인다.
- ▶ 관리자의 경우 : week 비율이 굉장히 높다. 이 또한 자발적 이탈이 아닌 강제이탈(계정 정지)로 보인다.
- ▶ 작업장의 경우 이탈비율(강제이탈)이 굉장히 높은데, 이는 불소 내부 시스템이 작업장 계정들을 잘 잡아내고 있는 것으로 생각된다.

Auto 유저의 이탈은 “강제이탈”

| Analysis - Person



Person? ?

THANK YOU