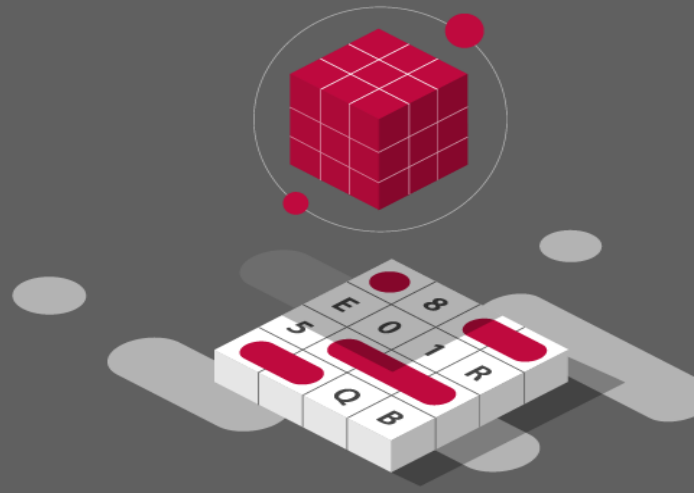
사용자 불편 예지 AI 경진대회



- LearningHB

LG AI Research

DACON
DATA TO VALUE

Team

LearningHB

김학빈
(Hakbin Kim)

YOOYONG

유용준
(Yongjun Yoo)

목차

1. 대회 소개

3. Feature Engineering

5. 결론

2. EDA

4. 모델링

1. 대회 소개

1. 대회 소개

1. 배경

- 비식별화 된 시스템 기록(로그 및 수치 데이터)을 분석하여
  시스템 품질 변화로 사용자에게 불편을 야기하는 요인을 진단

- 다양한 장비/서비스에서 일어나는 시스템 데이터를 통해 사용자의 불편을 예지하기 위해
  '시스템 데이터'와 '사용자 불편 발생 데이터'를 분석하여 불편을 느낀 사용자와 불편 요인들을 탐색

2. 성능 평가 지표 : AUC

3. 대회 일정

- 01.06 대회 시작
- 01.27 팀 병합 마감
- 02.03 대회 종료
- 02.07 상위 20팀 코드, PPT 제출 마감
- 02.15 온라인 평가 대상 발표

1. 대회 소개

4. 데이터 소개

학습 데이터 (user_id : 10000 ~ 24999, 15000명)

train_err_data.csv : 시스템에 발생한 에러 로그
train_quality_data.csv : 시스템 퀄리티 로그
train_problem_data.csv : 사용자 불만 및 불만이 접수된 시간

테스트 데이터(user_id : 30000 ~ 44998, 14999명)

test_err_data.csv : 시스템에 발생한 에러 로그
test_quality_data.csv : 시스템 퀄리티 로그
sample_submission.csv : 사용자 불만 확률(0~1) (제출용)

err_data

Col_name	Type
user_id	int64
time	int64
model_nm	object
fwver	object
errtype	int64
errcode	object

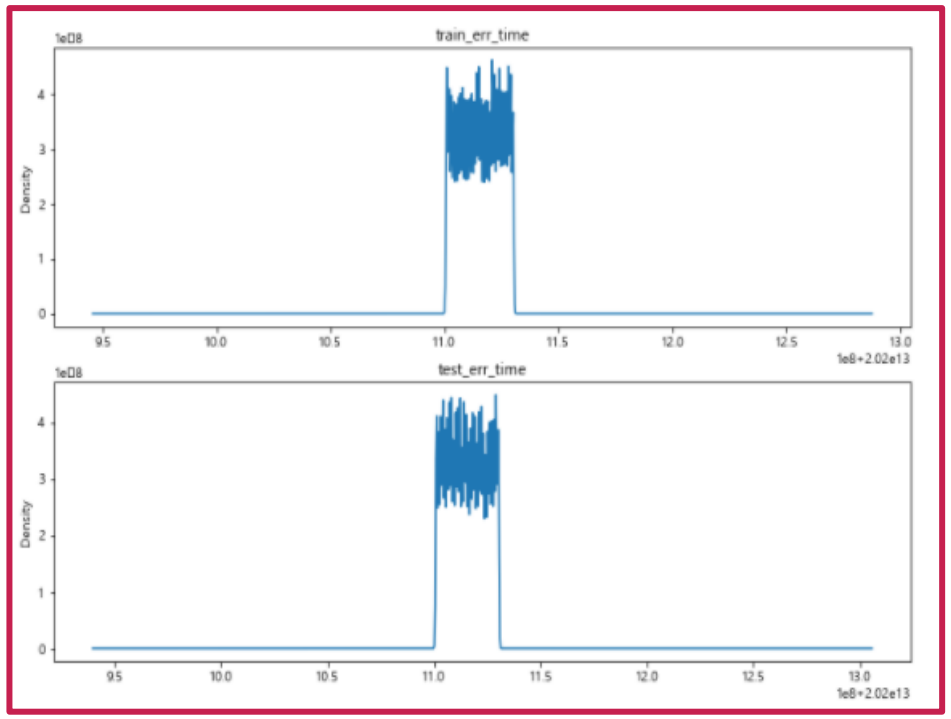
quality_data

Col_name	Type	Col_name	Type	Col_name	Type	Col_name	Type
user_id	int64	**quality_1**	int64	**quality_5**	object	**quality_9**	object
time	int64	**quality_2**	float64	**quality_6**	int64	**quality_10**	object
fwver	object	**quality_3**	int64	**quality_7**	object	**quality_11**	int64
quality_0	float64	**quality_4**	int64	**quality_8**	object	**quality_12**	int64

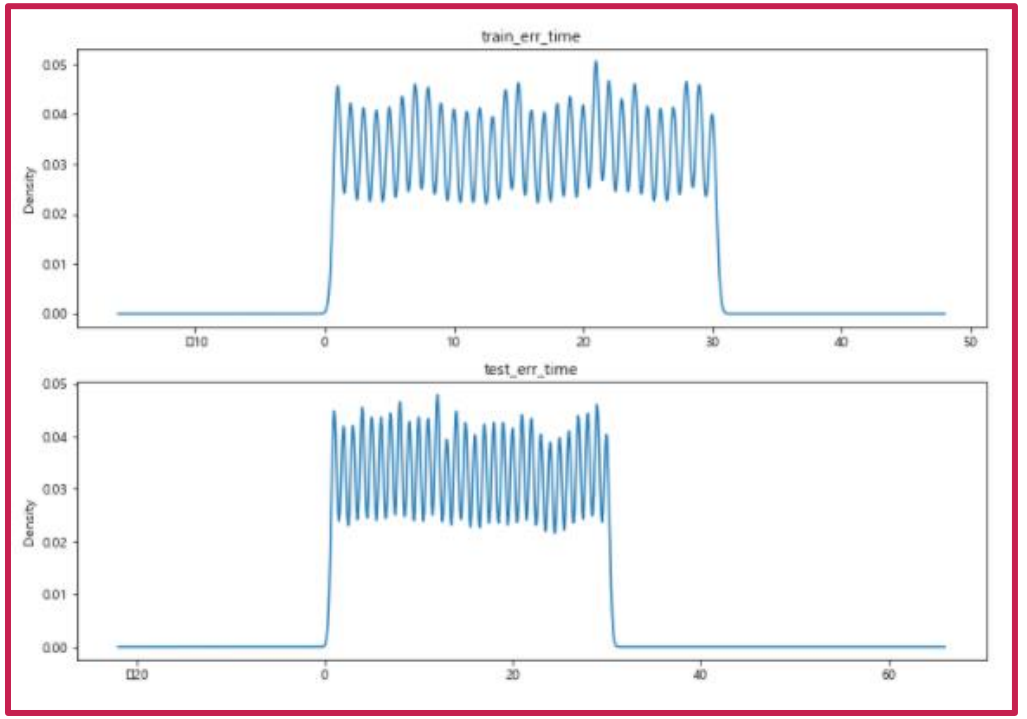
2. EDA

2.1 err data
2.2 quality_data

2. EDA err data - time

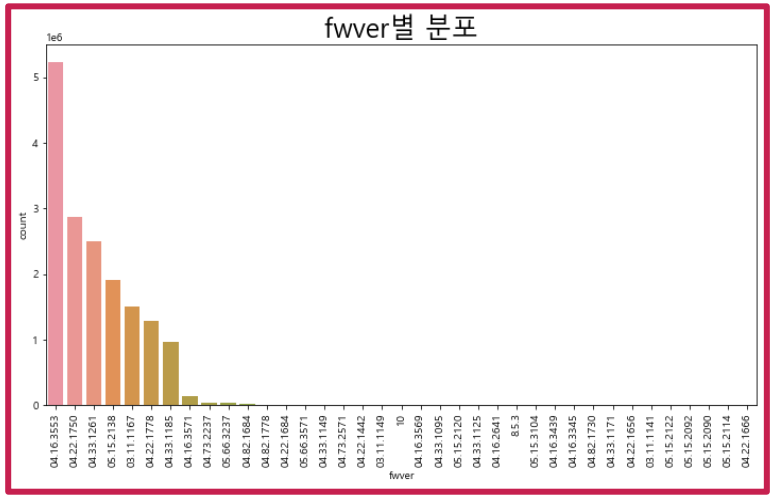


Day

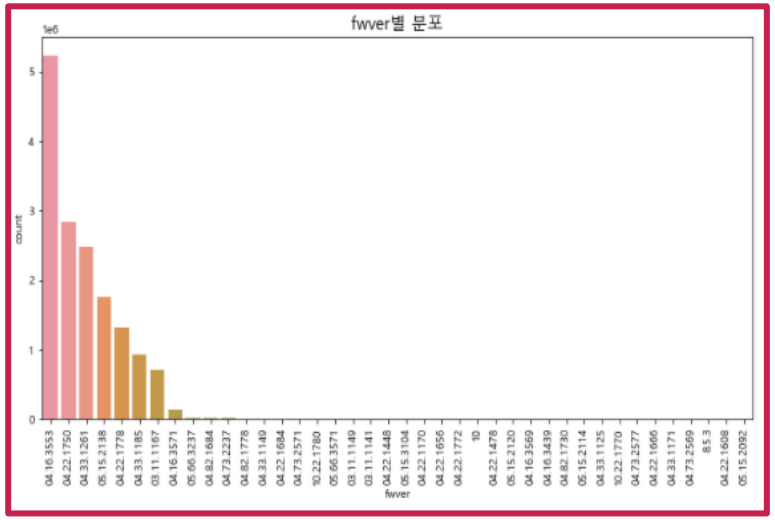


- Err data의 'time' feature 분포 확인
  - ✓ 오전/오후를 주기로 발생 빈도의 변화가 큼을 확인할 수 있음
  - ✓ Error 발생 시간이 문제제기에 의미있는 영향을 끼칠 수 있다고 판단

2. EDA err data - fwver



	model_nm	fwver	count	per
9	model_1	04.16.3553	5145420	33.481388
4	model_0	04.22.1750	2748374	17.883744
17	model_2	04.33.1261	2375757	15.459115
23	model_3	05.15.2138	1841272	11.981207
5	model_0	04.22.1778	1243732	8.092998
16	model_2	04.33.1185	911660	5.932196
27	model_4	03.11.1167	845808	5.503696
11	model_1	04.16.3571	142173	0.925124
35	model_8	04.73.2237	31480	0.204841
33	model_7	05.66.3237	30062	0.195614



	model_nm	fwver	count	per
13	model_1	04.16.3553	5239492	33.743913
7	model_0	04.22.1750	2831471	18.235530
20	model_2	04.33.1261	2476673	15.950523
24	model_3	05.15.2138	1759173	11.329606
9	model_0	04.22.1778	1314265	8.464264
19	model_2	04.33.1185	938117	6.041757
28	model_4	03.11.1167	715429	4.607579
15	model_1	04.16.3571	142617	0.918497
34	model_7	05.66.3237	24761	0.159468
29	model_5	04.82.1684	22667	0.145982

- fwver의 분포 확인
  - ✓ Train_err의 fwver unique() : 37개
  - ✓ Test_err의 fwver unique() : 40개

  => Err_data의 전체 fwver : 46개

```
train에만 있는 fwver: ['04.22.1442', '04.33.1095', '04.16.3345', '05.15.2122', '05.15.2090', '04.16.2641']
test에만 있는 fwver : ['04.22.1478', '10.22.1780', '04.22.1608', '04.73.2577', '10.22.1770', '04.22.1772', '04.22.1170', '04.73.2569', '04.22.1448']
```

- 상위 10개의 fwver이 전체의 약 99.65%를 차지함
- Model_0~4의 fwver이 상위권에 위치함을 알 수 있음
  - ✓ model_1의 '04.16.3553'이 가장 많음
  - ✓ model_0의 '04.22.1750'이 2번째로 많음

2. EDA err data - fwver

Problem user_fwver

	model_nm	fwver	count	per
8	model_1	04.16.3553	2598992	37.595985
3	model_0	04.22.1750	1340414	19.389896
15	model_2	04.33.1261	1036642	14.995651
4	model_0	04.22.1778	561515	8.122653
18	model_3	05.15.2138	519089	7.508935
14	model_2	04.33.1185	383301	5.544680
21	model_4	03.11.1167	327066	4.731207
10	model_1	04.16.3571	93120	1.347037
26	model_7	05.66.3237	16163	0.233808
28	model_8	04.73.2237	10484	0.151657
22	model_5	04.82.1684	10252	0.148301
24	model_5	04.82.1778	4244	0.061392
12	model_2	04.33.1149	2475	0.035802
0	model_0	04.22.1442	2285	0.033054
27	model_7	05.66.3571	1799	0.026024
2	model_0	04.22.1684	1631	0.023593
29	model_8	04.73.2571	1472	0.021293
9	model_1	04.16.3569	977	0.014133
5	model_1	04.16.2641	293	0.004238
17	model_3	05.15.2120	153	0.002213

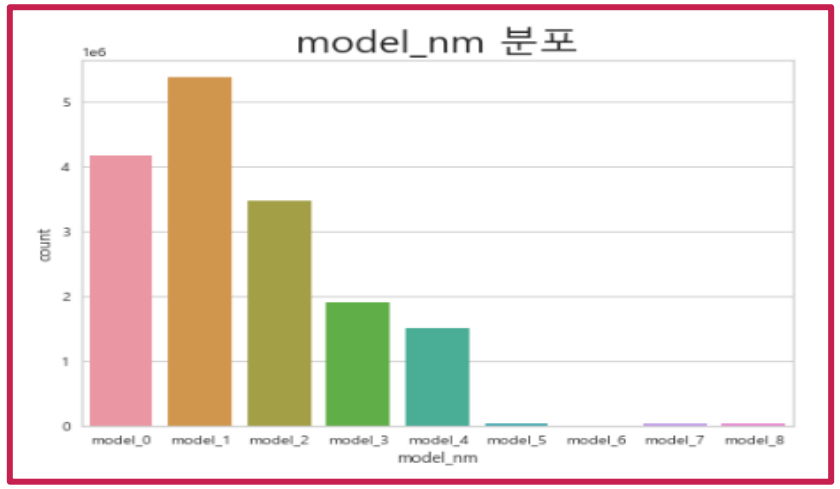
Not problem user fwver

	model_nm	fwver	count	per
7	model_1	04.16.3553	2546428	30.117240
3	model_0	04.22.1750	1407960	16.652295
14	model_2	04.33.1261	1339116	15.838059
19	model_3	05.15.2138	1322183	15.637789
4	model_0	04.22.1778	682217	8.068751
13	model_2	04.33.1185	528359	6.249034
23	model_4	03.11.1167	518742	6.135291
8	model_1	04.16.3571	49053	0.580162
30	model_8	04.73.2237	20996	0.248325
28	model_7	05.66.3237	13899	0.164387
24	model_5	04.82.1684	11247	0.133021
25	model_5	04.82.1778	3732	0.044139
2	model_0	04.22.1684	3442	0.040709
29	model_7	05.66.3571	1380	0.016322
26	model_6	10	1373	0.016239
22	model_4	03.11.1149	1299	0.015364
9	model_2	04.33.1095	858	0.010148
31	model_8	04.73.2571	650	0.007688
17	model_3	05.15.2120	548	0.006481
11	model_2	04.33.1149	513	0.006067

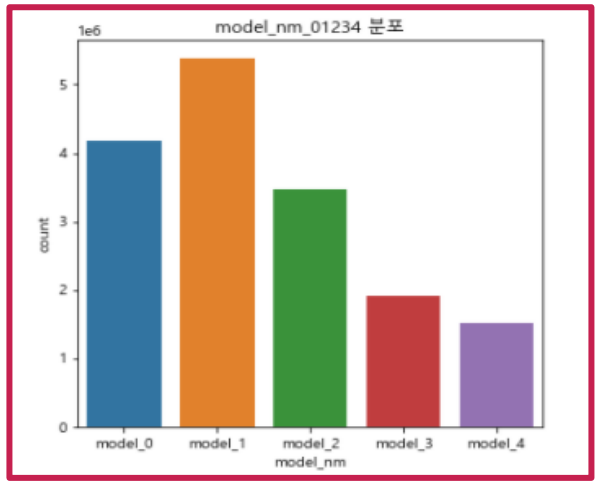
모델 버전	불만 비율	모델 버전	불만 비율
model1 04.16.3571	65.4	model2 04.33.1261	43.6
model1 04.16.3553	50.5	model2 04.33.1185	42.0
model0 04.22.1750	48.7	model4 03.11.1167	38.6
model0 04.22.1778	45.1	model3 05.15.2138	28.1

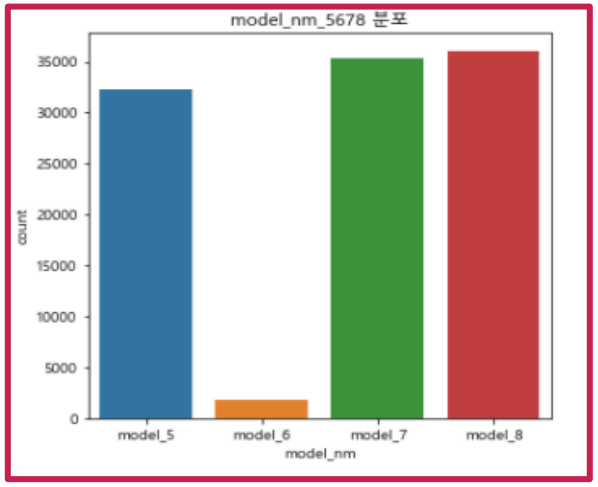
- 문제 제기한 user와 문제 제기하지 않은 user의 fwver 비교
- 두 부류의 fwver별 불만 접수 비율 확인*(두 부류의 발생건수 합이 10만건을 넘는 상위 8개의 모델,버전)*
  - ✓ model1 04.16.3571 사용자들의 불만 비율이 65.4로 가장 높음
  - ✓ model3 05.15.2138 사용자들의 불만 비율이 28.1로 가장 낮음
  - ❑ '04.16.3553' : count 비율도 높고, 불만 비율도 높은 변수이므로 중요 변수로 판단
- fwver에 따라 사용자들의 불만 비율에 유의미한 차이가 있다고 판단

2. EDA err data - model_nm



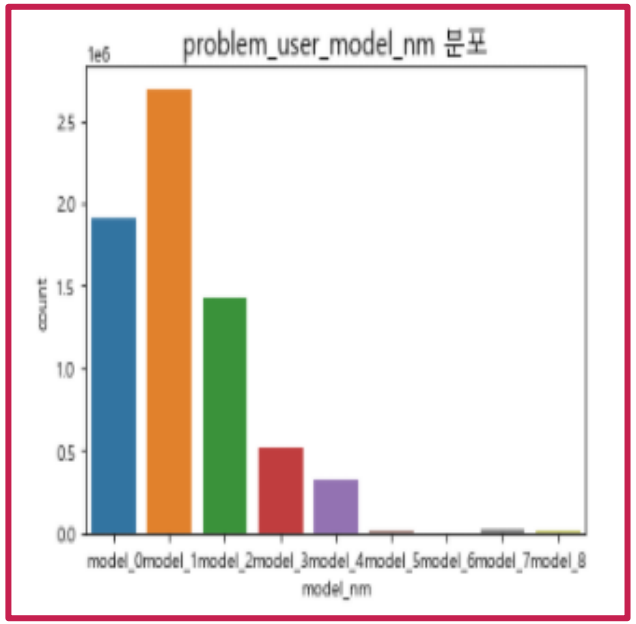
	model_nm	count	per
0	model_0	3999738	26.026404
1	model_1	5289106	34.416356
2	model_2	3291723	21.419331
3	model_3	1842206	11.987284
4	model_4	847124	5.512259
5	model_5	29553	0.192302
6	model_6	1708	0.011114
7	model_7	33241	0.216300
8	model_8	33602	0.218649

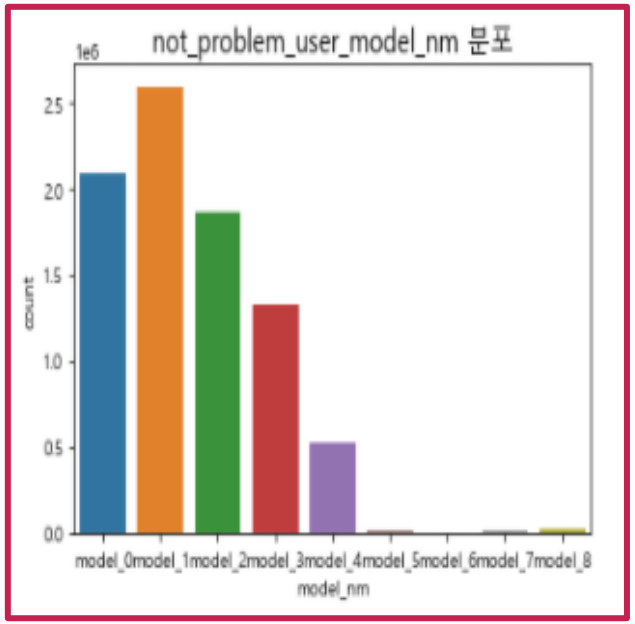




- Model_nm의 분포 확인
  - ✓ model0, 1, 2, 3, 4의 비율이 전체의 약 99.36%를 차지함
  - ✓ Model_0~4 & Model_5~8의 두 부류로 분류한 후 형태 파악
    - ➢ 전체 model 대비 model_0~4의 비율은 거의 동일함
    - ➢ Model_5~8에서 model_6의 비율이 현저하게 떨어짐을 알 수 있음

2. EDA err data - model_nm

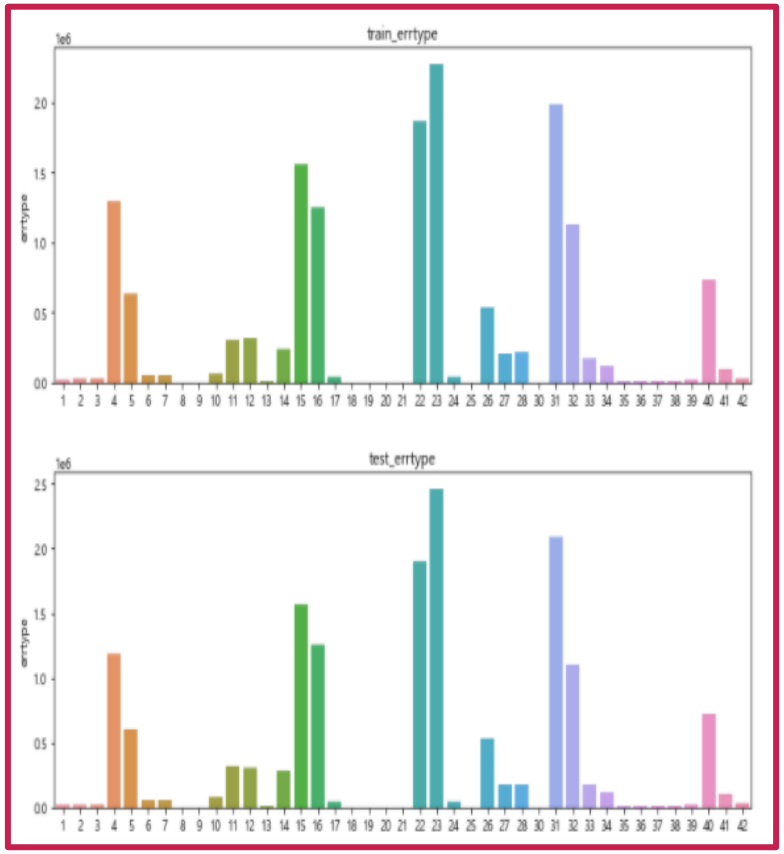




모델	불만 비율	모델	불만 비율
model_7	53.0	model2	43.0
model_1	51.0	Model_8	35.8
Model_4	48.4	Model_3	29.9
Model_0	47.9	Model_6	7.4
Model_5	47.4		

- 문제 제기한 user와 문제 제기하지 않은 user의 model 비교
- 두 부류의 model별 불만 접수 비율 확인
  - ✓ Model_7 사용자들의 불만 비율이 53.0으로 가장 높음
  - ✓ Model_6 사용자들의 불만 비율이 7.4로 가장 낮음
  - ❑ Model_1 : count 비율도 높고, 불만 비율도 높은 변수이므로 중요 변수로 판단
  - ❑ Model_7 : count 비율은 매우 낮으나, 불만 비율은 가장 높게 나왔으므로 의미가 있다고 판단
- model에 따라 사용자들의 불만 비율에 유의미한 차이가 있다고 판단

2. EDA err data - errtype

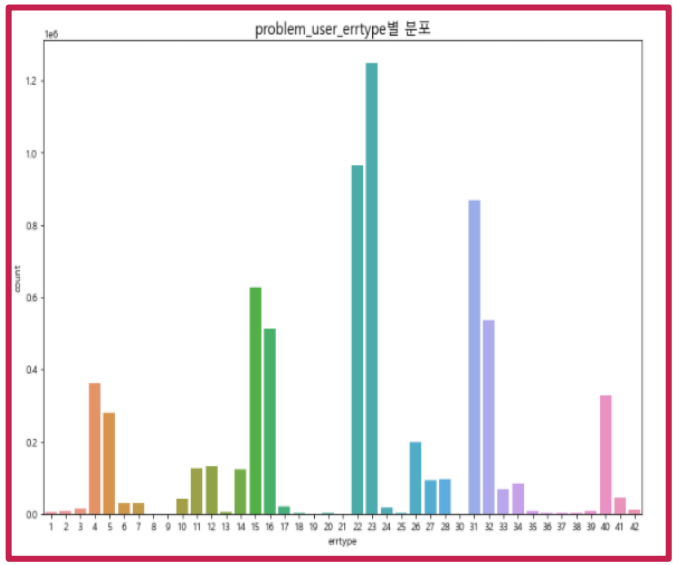


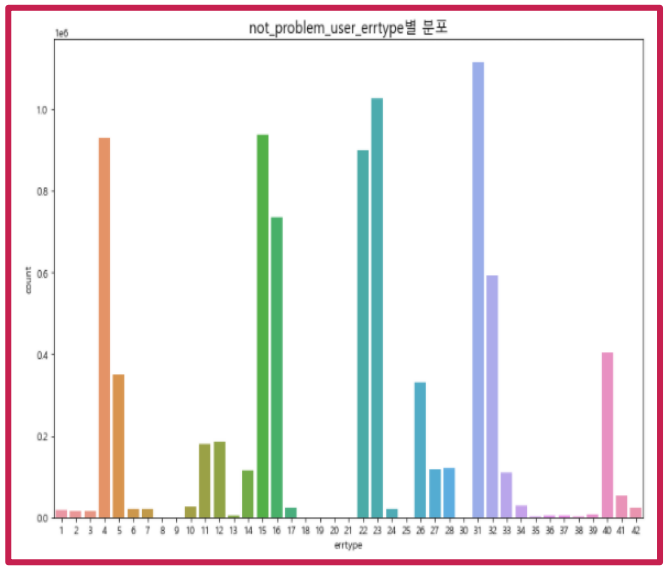
	errtype	count	per
0	23	2276011	14.810065
1	31	1985037	12.916690
2	22	1863495	12.125813
3	15	1562517	10.167341
4	4	1291986	8.406988
5	16	1248142	8.121694
6	32	1130247	7.354548
7	40	730283	4.751971
8	5	630279	4.101243
9	26	530998	3.455218
10	12	318847	2.074746
11	11	305205	1.985977
12	14	239212	1.556559
13	28	215338	1.401210
14	27	210131	1.367328
15	33	178271	1.160014
16	34	114833	0.747221
17	41	99368	0.646590
18	10	68832	0.447892
19	7	51972	0.338183

	errtype	count	per
0	23	2460031	15.843344
1	31	2093495	13.482741
2	22	1899965	12.236349
3	15	1570320	10.113336
4	16	1251366	8.059176
5	4	1180677	7.603917
6	32	1105475	7.119593
7	40	727903	4.687916
8	5	601554	3.874190
9	26	528395	3.403024
10	11	321325	2.069430
11	12	302658	1.949209
12	14	280446	1.806157
13	33	178077	1.146870
14	28	177303	1.141885
15	27	172564	1.111364
16	34	114658	0.738432
17	41	103529	0.666758
18	10	78109	0.503046
19	7	56518	0.363993

- errtype의 분포 확인
  - ✓ train과 test의 형태가 거의 동일함
  - ✓ errtype 상위 20개가 전체의 약 98%를 차지함
  - ✓ errtype 발생 빈도는 Errtype_23, 31, 22, 15, 4, 16, 32 …순으로 발생함
  - ✓ Errcode에 비해 비교적 여러 type에 분포함

2. EDA err data - errtype

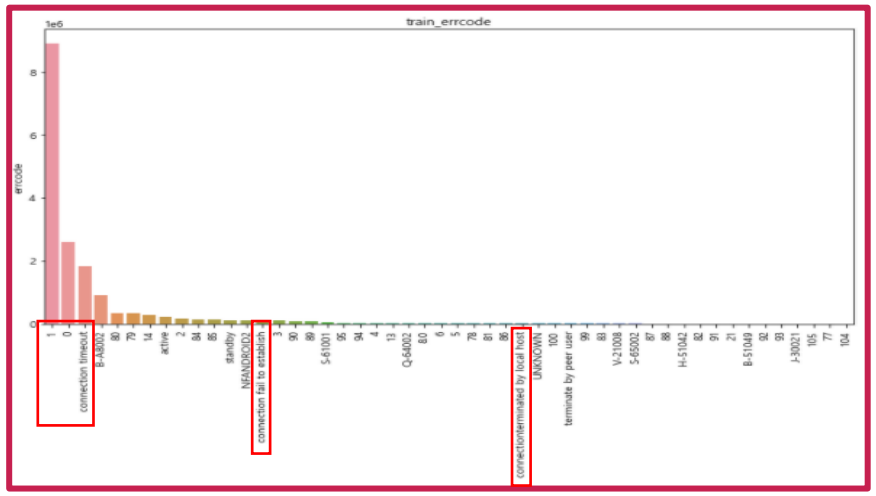




errtype	불만 비율	errtype	불만 비율
25	88.5	12	41.5
18	87.6	42	41.4
20	87.0	11	41.2
19	81.7	16	41.1
21	80.7	4	40.7
30	71.9	15	40.7
10	63.7	32	37.4
33	61.8	2	35.7
36	58.1	9	34.8
41	57.3	8	33.7

- 문제 제기한 user와 문제 제기하지 않은 user의 errtype 비교
- 두 부류의 errtype별 불만 접수 비율 확인(**상위 10개, 하위 10개** 추출)
  - ✓ Errtype_25 발생 시 사용자들의 불만 비율이 88.5로 가장 높음
  - ✓ Errtype_8 발생 시 사용자들의 불만 비율이 33.7로 가장 낮음
  - ❑ Errtype 발생 count 상위 8개 type인 23, 31, 22, 15, 4, 16, 32, 40이 불만 비율의 상위권에 위치하지 않음을 확인
  - ❑ Errtype 25 18 20 19 21은 발생 횟수는 낮지만 불만 제기 비율이 80% 이상임
- errtype에 따라 사용자들의 불만 비율에 유의미한 차이는 존재하지만 단순히 count와는 다른 양상을 띔을 알 수 있음

2. EDA err data - errcode

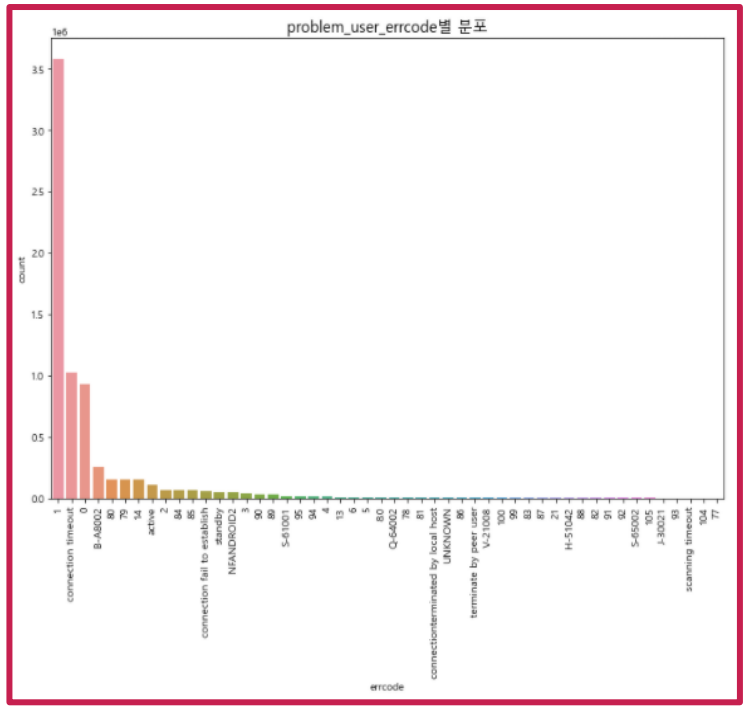


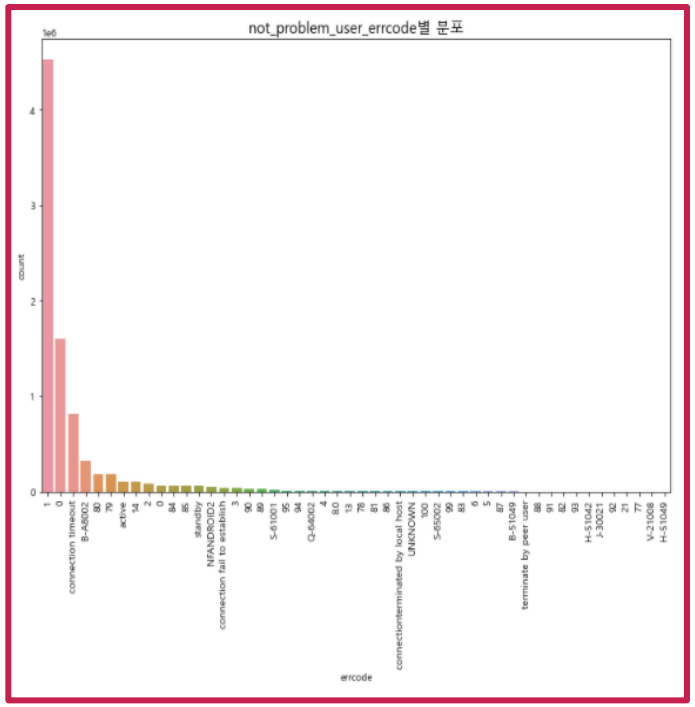
	errcode	count	per
0	1	8097696	52.691928
1	0	2594264	16.880946
2	connection timeout	1835262	11.942100
3	B-A8002	575827	3.746922
4	80	333929	2.172885
5	79	332356	2.162650
6	14	250982	1.633147
7	active	219195	1.426308
8	2	150920	0.982041
9	84	129829	0.844801

	errcode	count	per
0	1	8102249	52.180934
1	0	2559697	16.485223
2	connection timeout	1890320	12.174233
3	B-A8002	547219	3.524256
4	80	326075	2.100022
5	79	324096	2.087276
6	14	302905	1.950800
7	active	225493	1.452243
8	2	145311	0.935847
9	84	140608	0.905558

- Errcode의 분포 확인
  - ✓ train과 test의 형태가 거의 동일함
  - ✓ '1', '0', 'connection timeout'이 대부분을 차지함
  - ✓ 특정 code에 분포가 치중되어있음
  - ✓ 'connection'이 자주 발생하는 것으로 보아 중요한 변수로 판단
  - ✓ errcode 상위 20개가 전체의 약 99.2%를 차지함

2. EDA err data - errcode

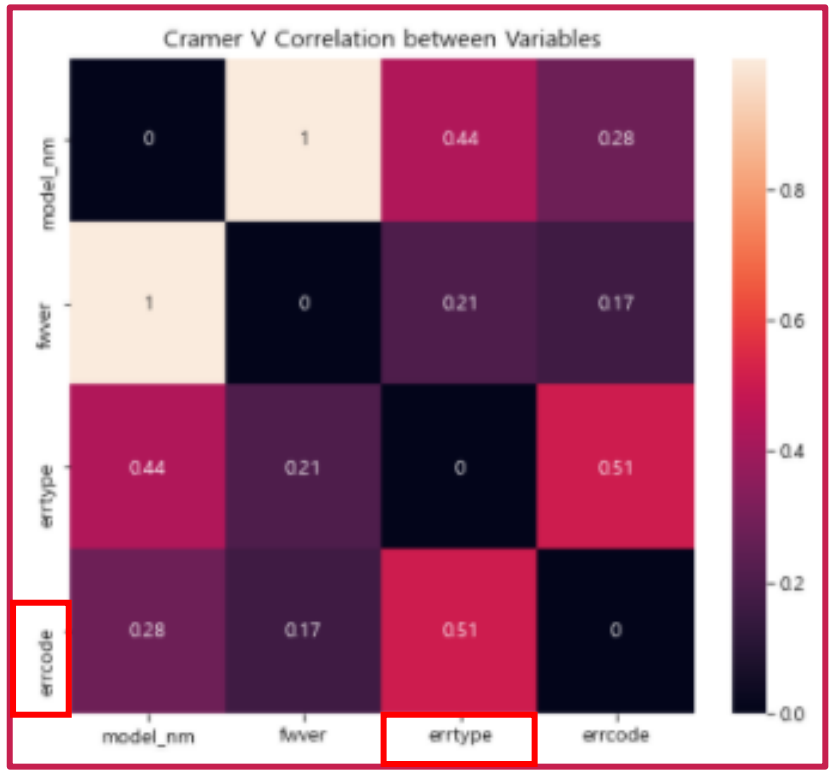




errcode	불만 비율	errcode	불만 비율
79	56.9	84	50.4
standby	54.7	connection fail to establish	48.8
NFANDROID2	53.9	2	48.5
4	53.8	3	48.1
0	53.4	S-61001	46.0
95	53.1	1	45.7
89	52.1	80	45.0
90	52.1	14	45.0
active	50.7	B-A8002	44.6
85	50.5	connection timeout	38.0

- 문제 제기한 user와 문제 제기하지 않은 user의 errcode 비교
- 두 부류의 errcode별 불만 접수 비율 확인(**상위 20개** 추출 : 전체의 약 99.2%)
  - ✓ Errcode '79' 발생 시 사용자들의 불만 비율이 56.9로 가장 높음
  - ✓ Errcode 'connection timeout' 발생 시 사용자들의 불만 비율이 38.0로 가장 낮음
  - ❑ Errcode '79' : count 비율도 높고, 불만 비율도 높은 변수이므로 중요 변수로 판단
  - ❑ 'connection', '0', '1' 역시 불만 비율도 높게 나왔으므로 중요 변수로 판단
- errcode에 따라 사용자들의 불만 비율에 유의미한 차이가 있다고 판단

2. EDA err data - 상관관계



Effect size (ES)	Translate
ES ≤ 0.2	The result is weak. Although the results are statistically significant, the fields are only weakly related.
0.2 <ES ≤ 0.6	The result is reasonable. The fields are properly related.
ES> 0.6	The result is strong. The fields are strongly related.

- **Feature별 상관관계 파악**

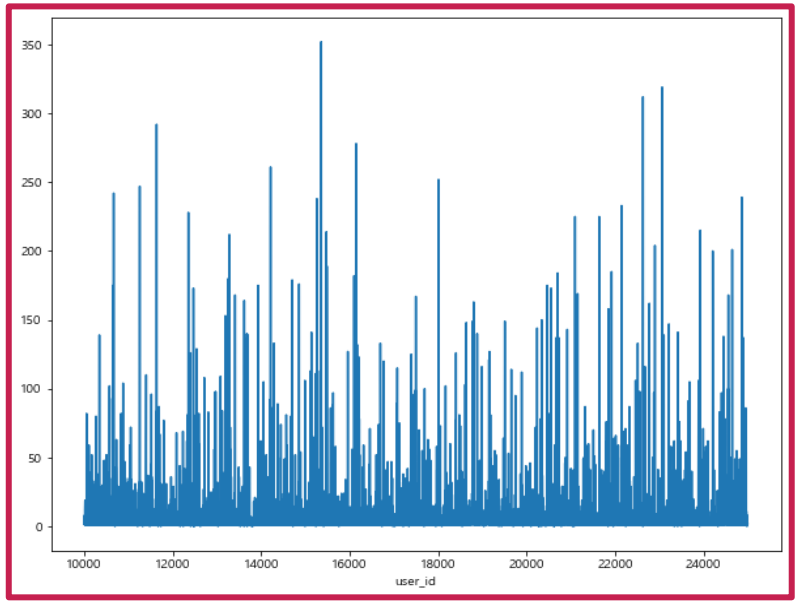
✓ Cramér's V is a measure of the effect size of the chi-square test of independence. It measures how strongly the two categorical fields are related.

✓ **이산형 변수이므로 Cramers V 사용**

✓ **0 < cramers's V < 1**

✓ **errtype & errcode의 cramer's V : 0.51로 err의 종류라는 예상에 맞게 어느정도 관련이 있음을 알 수 있음**

2. EDA quality data - time



user_id	time	
10000	20201129090000	12
	20201130210000	12
10002	20201104110000	12
	20201106010000	12
	20201111010000	12
		..
24993	20201130203000	12
24995	20201128202000	12
	20201129002000	12
24997	20201108230000	12
	20201124033000	12

- quality_data에서 user별 quality 측정 횟수
  - ✓ Time 1번은 2시간동안 10분 간격으로 찍힌 것이므로 quality_0~12의 값들이 12회 측정됨
  - ✓ 그 속에서의 변화들이 user의 문제제기에 영향을 미친다고 판단

```
train_quality[train_quality.user_id==10002]
```

	time	user_id	fwver	quality_0	quality_1	quality_2	quality_3	quality_4	quality_5	quality_6	quality_7	quality_8	quality_9	quality_10
24	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
25	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
26	20201104110000	10002	05.15.2138	2.0	0	1.0	0	0	0	0	0	0	1	0
27	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
28	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
29	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
30	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
31	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
32	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
33	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
34	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0
35	20201104110000	10002	05.15.2138	0.0	0	0.0	0	0	0	0	0	0	1	0

2. EDA quality data - fwver

```
train_quality의 fwver :  27

array(['05.15.2138', '04.22.1750', '04.16.3553', '04.33.1261',
       '04.22.1778', '04.33.1185', '04.16.3571', '05.66.3571',
       '03.11.1149', '03.11.1167', '04.82.1684', '04.82.1778',
       '04.33.1149', '05.66.3237', '04.73.2237', '09.17.1431',
       '04.22.1684', '05.15.2120', '04.33.1125', '05.15.2122',
       '04.22.1666', '04.22.1656', '04.16.3439', '04.73.2571',
       '05.15.2114', '04.16.3345', '04.22.1442'], dtype=object)

test_quality의 fwver :  22

array(['04.33.1261', '05.15.2138', '04.22.1750', '04.22.1778',
       '04.16.3553', '09.17.1431', '03.11.1167', '04.33.1149',
       '04.33.1185', '04.22.1684', '04.82.1684', '04.16.3571',
       '04.73.2571', '04.82.1778', '04.73.2237', '05.66.3237',
       '05.66.3571', '03.11.1149', '05.15.2120', '04.33.1125',
       '04.16.3439', '05.15.2114'], dtype=object)

train에만 있는 fwver: ['04.22.1666', '04.22.1442', '04.16.3345', '05.15.2122',
'04.22.1656']
test에만 있는 fwver : []
```

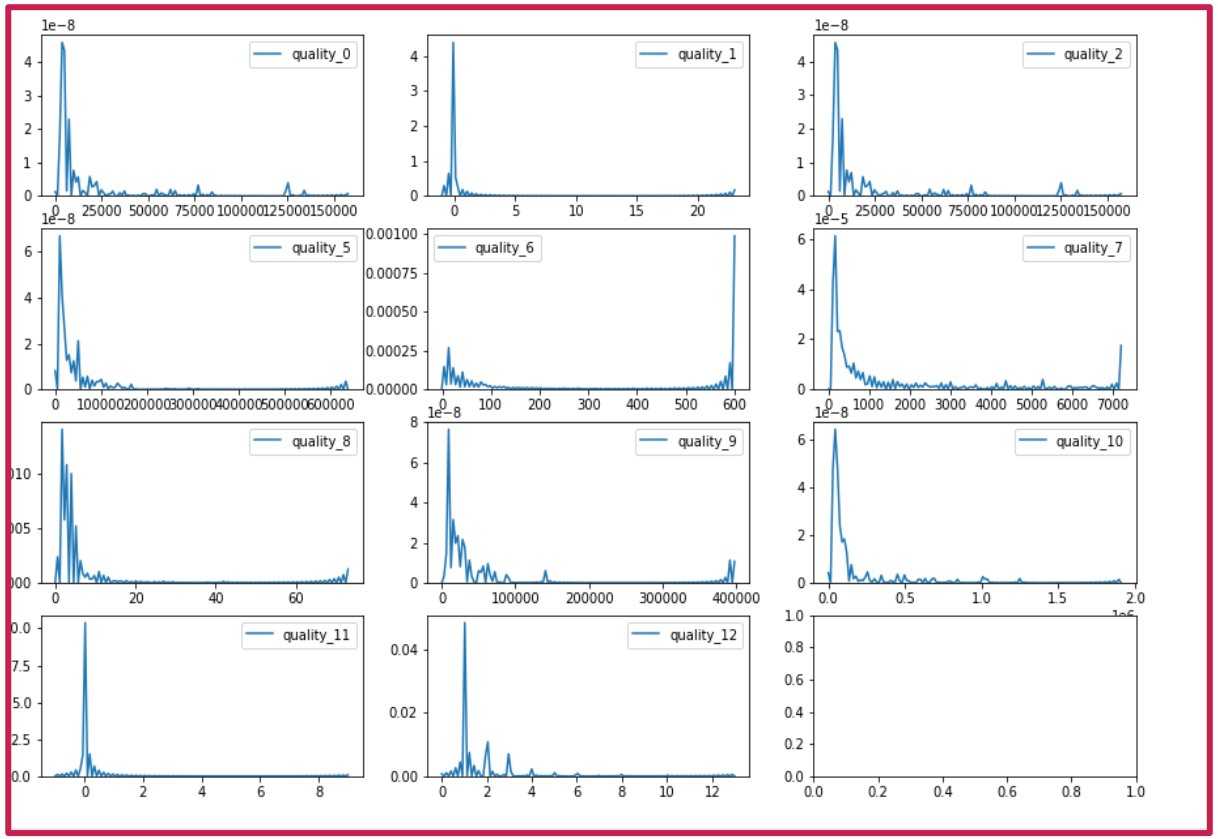
- Quality_data의 fwver 확인
  - ✓ Train_quality의 fwver unique() : 27개
  - ✓ Test_quality의 fwver unique() : 22개
    - ➢ train_quality의 fwver에 test의 fwver 모두 포함됨
  - ✓ quality_data에만 있는 fwver : ['09.17.1431']

```
quality_data의 fwver :  27

array(['05.15.2138', '04.22.1750', '04.16.3553', '04.33.1261',
       '04.22.1778', '04.33.1185', '04.16.3571', '05.66.3571',
       '03.11.1149', '03.11.1167', '04.82.1684', '04.82.1778',
       '04.33.1149', '05.66.3237', '04.73.2237', '09.17.1431',
       '04.22.1684', '05.15.2120', '04.33.1125', '05.15.2122',
       '04.22.1666', '04.22.1656', '04.16.3439', '04.73.2571',
       '05.15.2114', '04.16.3345', '04.22.1442'], dtype=object)

err_data에만 있는 fwver: ['04.22.1478', '04.33.1095', '04.82.1730', '05.15.209
0', '05.15.3104', '04.22.1608', '03.11.1141', '04.22.1170', '05.15.2092', '10.2
2.1780', '04.16.3569', '10.22.1770', '10', '8.5.3', '04.73.2577', '04.22.1772',
'04.33.1171', '04.73.2569', '04.22.1448', '04.16.2641']
```

- data 전체 fwver unique() : 47개

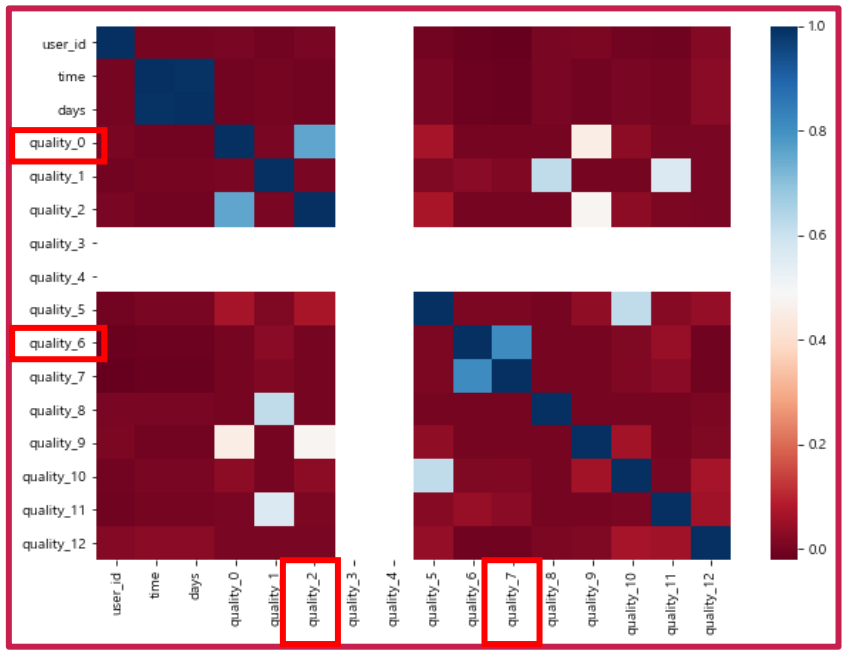
```
전체 fwver : 47
{'04.22.1478', '05.66.3237', '04.22.1666', '04.33.1095', '04.82.1730', '04.16.3553', '04.33.1149', '03.11.1167', '05.15.2090', '0
5.15.3104', '04.33.1185', '05.15.2120', '04.22.1608', '04.22.1448', '04.82.1684', '04.22.1778', '04.22.1442', '03.11.1141', '04.7
3.2571', '04.22.1170', '04.73.2237', '04.33.1261', '09.17.1431', '05.15.2092', '10.22.1780', '04.16.3569', '10.22.1770', '04.16.34
39', '10', '05.66.3571', '04.22.1750', '04.16.3345', '04.22.1684', '05.15.2122', '04.33.1125', '8.5.3', '05.15.2138', '04.22.165
6', '04.73.2577', '05.15.2114', '04.22.1772', '04.33.1171', '04.82.1778', '04.73.2569', '03.11.1149', '04.16.3571', '04.16.2641'}
```

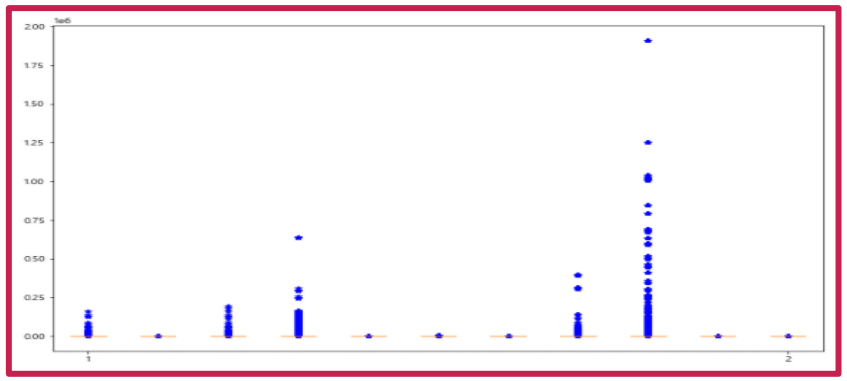
2. EDA quality data – quality_0~12



- quality_0~12의 분포 확인
  - ✓ Quality_3, 4는 모든 값이 0이므로 의미없는 변수로 판단하여 제거하기로 결정
  - ✓ 모든 quality는 대부분이 0, 1, -1로 이루어져 있음
  - ✓ Quality_10의 분포 : 3이 99828회로 가장 많고 다음 2, 0 ,1, 4 순으로 발생

2. EDA quality data - 상관관계





- Feature별 상관관계 파악
  - ✓ quality_0~12 : 연속형 변수
  - ✓ quality_3, 4는 모든 값이 0이므로 상관관계 x
  - ✓ quality_0 & 2의 상관관계는 0.759533로 높음
  - ✓ quality_6 & 7의 상관관계는 0.810524로 높음

- Quality_0~12의 boxplot
  - ✓ Quality_10의 분포가 가장 넓게 퍼져 있음을 알 수 있음

3. Feature Engineering

3. Feature Engineering - 파생변수

1. 에러 코드 발생횟수
(errtcode_1 ~ errtype_code_4351)

- 유저별로 에러코드의 발생 횟수

2. 주사용 fwver
('04.22.1448' ~ '09.17.1431')

- err_data에서 각 유저별로 사용중인 fwver

3. quality 발생 비율
('q0_value*cnt*per' ~ 'q12_value*cnt*per')

유저별 quality의 총 발생 횟수 대비 quality_{}의 발생 비율

4. 에러 코드 누적값
(day_0 ~ day_44)

- 45일간 각 유저의 errcode count를 일단위 누적한 값

5. 에러 타입 비율
(errtype_rate_1 ~ errtype_rate_42)

- 유저별로 발생한 총 에러 횟수와 각 에러타입 비율을 계산한 값

6. 에러 발생간격
(dt_time_step, q_dt_time_step)

- 반복적인 에러가 불편 접수와 연관이 있음을 가정하고 'time' feature를 datetime 형식으로 변환한 뒤 각 유저들의에러발생 최대간격 , 최소간격, 평균간격을 계산한 값

7. quality 통계 & 발생간격
(quality_0_mean ~ quality_12_max)

- quality_0~12의 mean, min, max

8. 주사용모델
(model_0 ~ model_8)

- err_data에서 각 유저별로 가장 많이 기록된 모델 onehot

4. 모델링

4. 모델링

- **Input data shape**

  ✓ **Train_set shape: (15000, 4989)**
  ✓ **Test_set shape: (14999, 4989)**

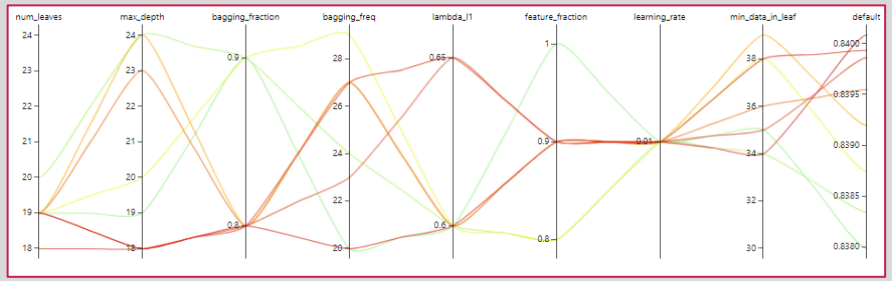
✓ **Lightgbm , xgboost, catboost로 각각 훈련 시켜보니 Lightgbm이 가장 높은 성능을 보임**
세 모델로 test를 예측하고 soft voting 하여 평가한 결과 성능향상은 없었음.

✓ **위의 개별 모델들의 파라미터를 튜닝하여 Logistic regreesion으로 스태킹 시도**
초기에는 더 나은 성능을 보였지만 최종 데이터에서는 Lightgbm이 더 높은 성능을 보임.

✓ **LogisticRegression, RandomForest, KNeighborsClassifier 각각의 모델을 GridSearch하여 앙상블을 시도하였으나 성능이 Lightgbm에 비해 현저하게 떨어졌음.**

4. 모델링

- 최종 모델
  - ➢ LightGBM / out of fold ensemble
  - ➢ Microsoft nni로 최적 파라미터 탐색
- 5 겹 k-fold 교차검증
- 각 fold에서 훈련된 모델로 test set을 예측
- 5개의 test set 예측값을 앙상블하여 최종 예측결과 생성



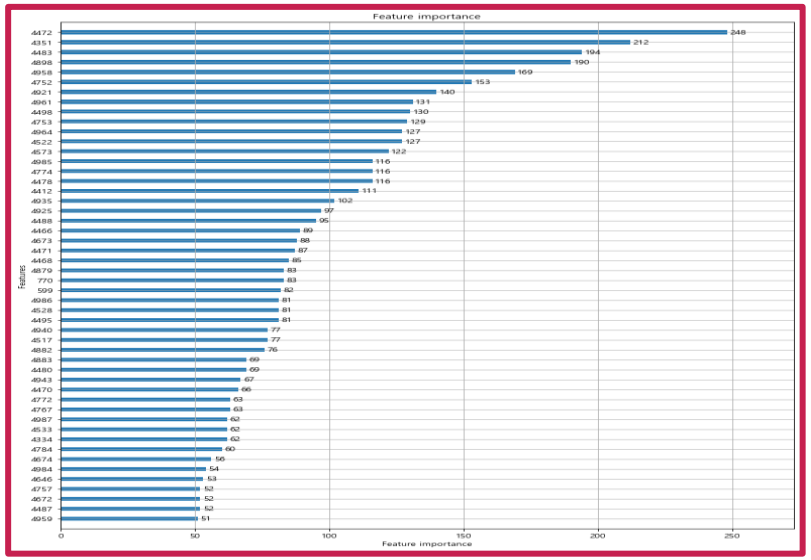

```
params =      { 'boosting_type': 'gbdt',
                'objective': 'binary',
                'metric': 'auc',
                'seed': 1010,
                'max_depth': 18,
                'num_leaves': 19,
                'learning_rate': 0.01,
                'bagging_fraction': 0.8,
                'bagging_freq': 20,
                'lambda_l1': 0.6,
                'feature_fraction' : 0.9,
                'min_data_in_leaf' : 34,
                'scale_pos_weight' : 1,  #defailt : 1
                'boost_from_average' : True  #default:True
               }
```

```
model = lgb.train(
                    params,
                    train_set       = d_train,
                    num_boost_round = 10000,
                    valid_sets      = [d_train, d_val],
                    feval           = f_pr_auc,
                    verbose_eval    = 100,
                    early_stopping_rounds = 800
                   )
```

교차검증 점수	Public 점수	private 점수
0. 8404646	0.84447	0.84165

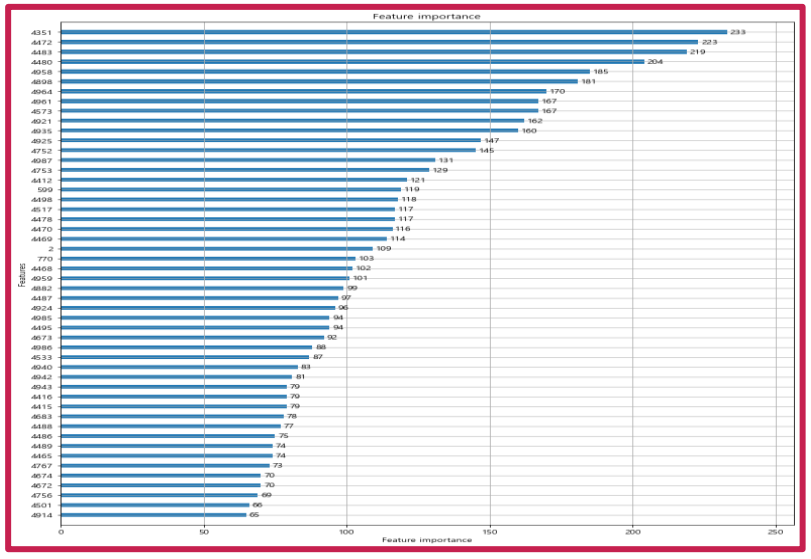
4. 모델링 – feature importance(fold별 상위 7개)

Model_0



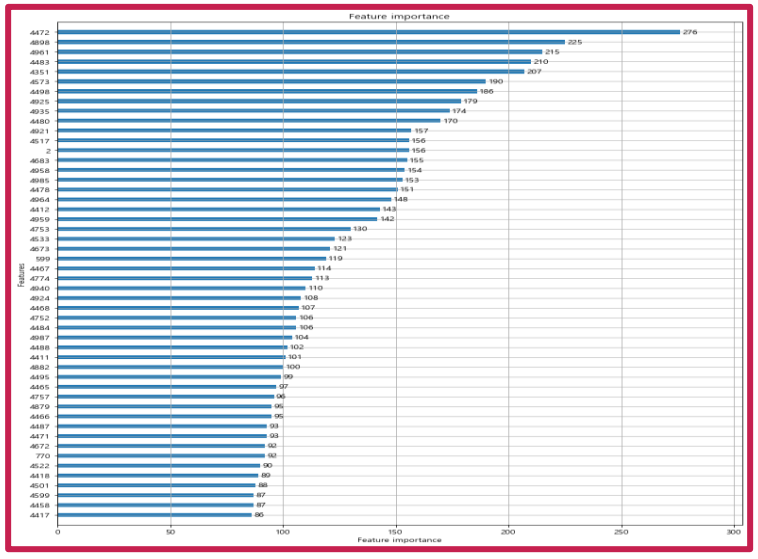
feature_num	feature_name	importance
4472	errtype_rate_18	248
4351	errcode_4351 : '5'	212
4483	errtype_rate_30	194
4898	Errcounts_day_std	190
4958	fw_model 변화	169
4752	Errcounts hour std	153
4921	Errcounts_day_max	140

Model_1



feature_num	feature_name	importance
4351	errcode_4351 : '5'	293
4472	errtype_rate_18	223
4483	errtype_rate_30	219
4480	errtype_rate_26	204
4958	fw_model 변화	185
4954	fw_model 변화	181
4961	Time_term	170

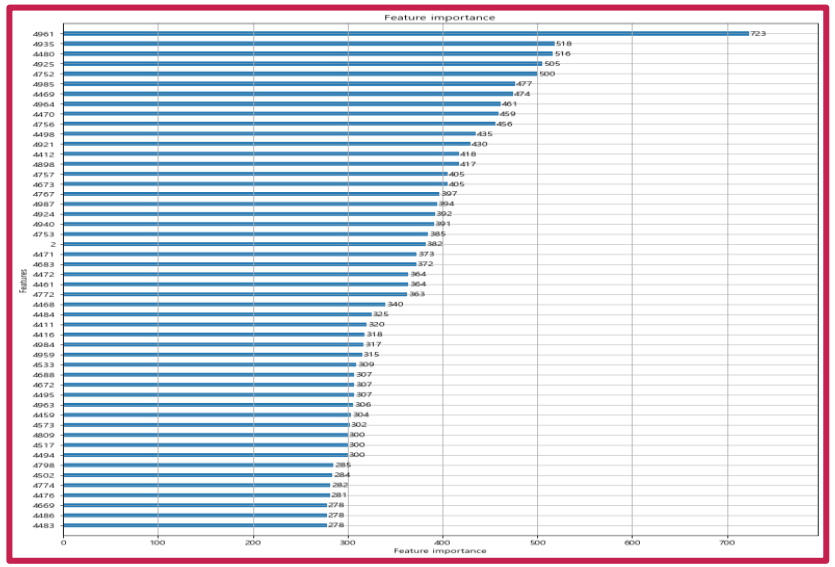
Model_2



feature_num	feature_name	importance
4472	errtype_rate_18	276
4898	Errcount_day_std	225
4961	Time_term	215
4483	errtype_rate_30	210
4351	errcode_4351	207
4573	Errtype_7	190
4498	dt_step_max	186

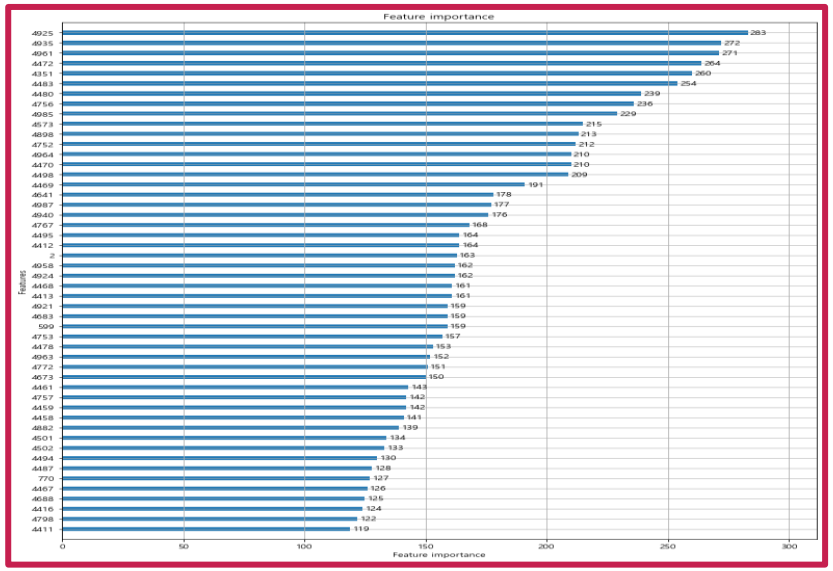
4. 모델링 – feature importance(fold별 상위 7개)

Model_3



feature_num	feature_name	importance
4961	Time_term	723
4935	Errcounts day max	518
4480	errtype_rate_26	516
4925	Errcounts day max	505
4752	Errcounts hour std	500
4985	가장 오랫동안 사용한 fw의 파생변수	477
4459	errtype_rate_5	474

Model_4



feature_num	feature_name	importance
4925	Errcounts day max	283
4935	Errcounts day max	272
4961	Time_term	271
4472	errtype_rate_18	264
4351	errcode_4351 : '5'	260
4483	errtype_rate_30	254
4480	errtype_rate_26	239

- Feature importance 확인 결과
- errtype_18 & errtype_30
  - 불만 비율에서 높은 순위에 있던 errtype18, 30이 대부분의 모델에서 공통적으로 중요변수로 선정됨
- Error 발생 count 관련 변수들이 중요함
  - Errcount가 클수록 불만제기율 증가
  - 편차가 작을수록 불만제기율 증가
- 시간 관련 변수가 중요함
  - Err 발생 간격이 불만 제기율에 영향을 미침
- Fwver의 변화가 중요함
  - User별로 fwver_model의 변화가 초기 에러의 발생을 야기할 수 있고, 이는 불만 제기율의 상승에 영향을 미칠 수 있다고 판단
- Errcode : '5'의 중요도가 높게 측정됨
  - '5'의 발생 빈도나 불만 제기율은 낮은 반면에 높은 중요도로 선택된 것으로 보아 주요한 변수라고 판단

5. 결론

5. 결론 – 결과 및 비즈니스 활용 방안

<Err_data>

- Errtype & errcode
  - ✓ Err_data의 errtype & errcode는 어느정도 비슷한 성격을 띄고 있음을 파악 하였고, 불만 제기에 있어 주요 변수로 선정된 type 및 code에 대한 추가적인 관리 필요
  - ✓ 발생한 Errtype, errcode와 불만을 제기한 유저의 비율로 어떤 에러가 사용자에게 가장 큰 불편을 주는지 분석하여 에러 재발 최소화
- Fwver & model_nm
  - ✓ User의 Model 또는 fwver의 변경 시에 err 발생 빈도가 높으므로 초기 조치를 통한 불만 제기율 감소 기대
  - ✓ 사용자 불만 제기 비율이 낮았던 모델, 버전과 높은 모델을 비교하여 개선사항 도출

<Quality_data>

- Fwver
  - ✓ Err_data와 마찬가지로 fwver의 초기 변경에 대한 조치 요망
- Quality_value & time
  - ✓ 시간의 변동에 따른 quality_value의 변화에 주목할 필요가 있음
  - ✓ Quality_value의 발생 비율에 따라 불만 제기율에 영향을 끼치므로 측정된 quality_value에 대한 적절한 관리 요망

Time 관련 변수의 중요성을 인지하고,
사용자 불만이 접수되기 이전에 이를 예측하고 조치함으로써
더 나은 UX 제공할 것으로 기대

Thank You

Appendix – 최종 features

errcode_{}	유저별 에러코드 count 값	**q_dt_time_step_max**	쿼리티 로그 발생 최대 간격
day_{}	관측기간 내 errcode 일 누적치	**q_dt_time_step_mean**	쿼리티 로그 발생 평균 간격
Fwver	fwver 원핫 feature	**quality_{}_min**	유저별 쿼리티값 최소
q{}_value*cnt*per	유저별 Quality 발생 비율	**quality_{}_max**	유저별 쿼리티값 최대
errtype_rate_{}	유저별 에러타입 발생 비율	**quality_{}_median**	유저별 쿼리티값 중간값
dt_time_step_min	에러 발생 최소간격	**quality_{}_sum**	유저별 쿼리티값 합
dt_time_step_max	에러 발생 최대간격	**quality_{}_mean**	유저별 쿼리티값 평균
dt_time_step_mean	에러 발생 평균간격	**model_{}**	주 사용 모델
q_dt_time_step_min	쿼리티 로그 발생 최소간격	**others**	Dacon 코드공유 feature

참고 코드 출처 :
https://dacon.io/competitions/official/235687/codeshare/2356?page=1&dtype=recent&ptype=%20