고객 이탈 예측 및 상품 추천 모델

고객 구매 데이터에 기반한 예측 모델 개발 및 개인화 마케팅 전략 제안

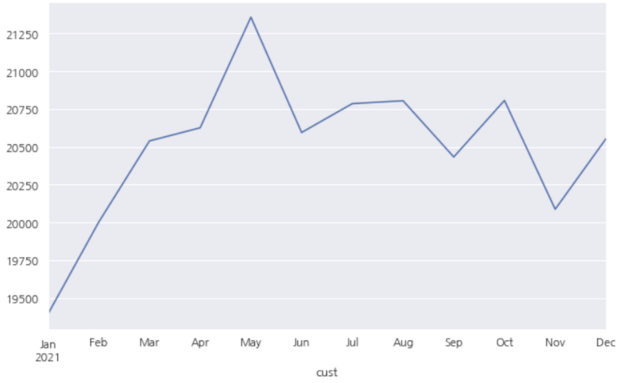
팀명: 크리스탈 012
팀원: 윤서영, 이연주, 허지원

문제 정의 (EDA)

문제정의 (1)

상품구매정보+제휴사이용정보

월별 구매자수

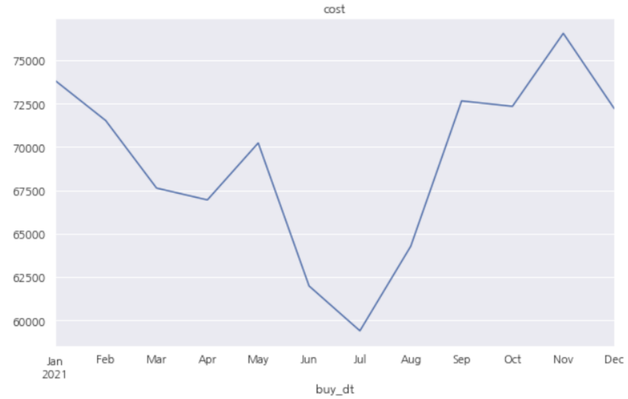


5월: 21,358명
1월: 19,394명

월별 매출액

5월: 10,121,793,088원
7월: 7,732,686,662원

월별 구매당 주문 금액추이



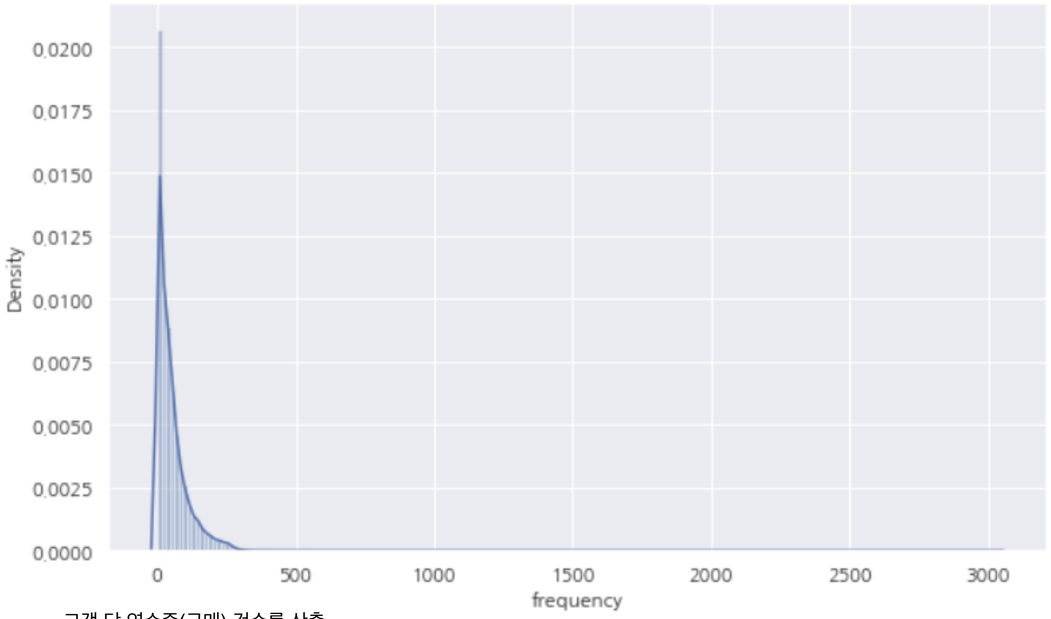
11월: 76,550원
7월: 59,392원

- 상품구매 & 제휴사이용 고객이 하락하는 추세
- 6~7월에 이용 고객 대비 구매 금액이 낮은 성적을 보임
- 이벤트성으로 참여한 고객들이 후에 빠져나간 것으로 보임

문제정의 (1)

상품구매정보+제휴사이용정보

전체 기간동안 1인당 구매 건수



고객 당 영수증(구매) 건수를 산출
* 상품구매정보의 경우, 영수증 번호가 고유하지 않음. 한 행을 구매 건수로 보지 않고, 하나의 영수증 번호를 구매건수로 봄
* 제휴사이용정보의 경우, 영수증 번호는 고유한 식별번호임.

- **전체고객**: 29,756명
- **한번만 구매한 고객**: 1,908명
- 전체고객 29756명중 1번만 구매한 고객은 1908명 으로 전체 데이터 중 가장 많은 비중을 차지
- 고객의 25%의 구매 건수가 11번 이하로 낮은 수치를 기록하며 높은 이탈률을 보임
- 커머스 입장에서는 이탈고객을 잡는 것이 중요하므로 해결책이 필요해보임

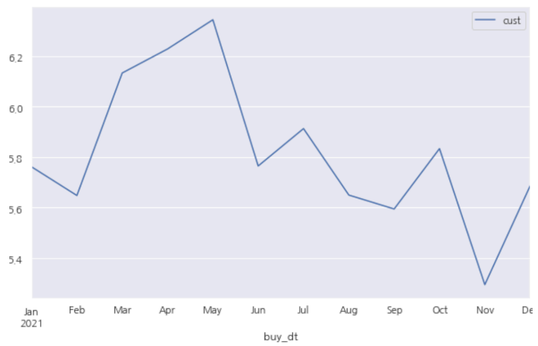
문제: 구매 건수 감소 추세, 높은 이탈률

해결방법: 이탈 고객 예측 모델 구현하기

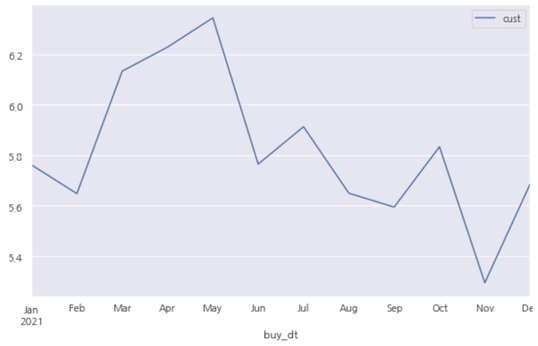
문제정의 (2)

상품구매정보

월별 1인당 구매 건수



월별 1인당 구매 금액



- **월별 인당 구매 건수**
  - **5월:** 6.3건 이상, **11월:** 5.3건 미만
- **월별 인당 구매 금액**
  - **5월:** 50만원 이상, **7월:** 40만원 초반
- 월별 구매 건수, 월별 구매 금액 지표들이 하락하는 추세
- 8월 이후 전략적인 대응으로 일인당 구매금액이 다시 회복되는 것으로 보임
- 전체 총매출도 일정 부분 회복된 것으로 보이나, 사용자의 꾸준한 이용을 이끌어내지 못한 것으로 보임
- 따라서 사용자의 구매 활동 회복보다는 1번 구매당 비용이 커진 것으로 예상

해결방법: 상품 추천 모델 구현하기

세그먼트

RFM 세그먼트

RFM 세그먼트란

CRM(Customer Relationship Management)분야에서 고객의 가치를 분석하는 방법 중 하나

상품구매정보 데이터에서 각 고객 별로 Recency, Frequency, Monetary 점수를 측정

Recency: 기준일-가장 마지막 구매일

Frequency: 고객 당 결제 장바구니 개수

Monetary: 고객 당 총 구매 금액

```
today_date = dt.datetime(2022, 1, 1)
df_rfm = (df_crm.groupby(['cust'])
          .agg({'de_dthr': lambda date: (today_date - date.max()).days,
                'rct_no': lambda num: num.nunique(), 'buy_am':'sum'})
)
df_rfm.columns = ['recency', 'frequency', 'monetary']
```

df_crm: 상품구매정보 데이터를 고객의 장바구니 별로 groupby한 데이터

두 기간 RFM을 이용한 고객세분화

상/하반기 RFM

시간 흐름에 따른 고객 구매 행동 변화를 살펴보기 위해

총 분석 기간인 1년을 상반기(2021.01.01 ~ 2021.06.30)와 하반기(2021.07.01 ~ 2021.12.31)로 나눔

df_crm을 상반기/하반기 데이터로 나누어 RFM 점수를 매김

cust	recency	frequency	monetary	recency_score	frequency_score	monetary_score
M000034966	1	8	397960	5	2	2
M000136117	13	41	18817570	3	4	5
M000225114	28	35	813180	2	4	3
M000261625	19	19	1609000	2	3	4
M000350564	6	15	5861900	4	3	5
...	...	...	...	...	...	...
M999599111	28	5	1457922	2	2	4
M999673157	109	8	168800	1	2	1
M999770689	1	55	506540	5	5	2
M999849895	15	17	592350	3	3	3
M999962961	2	62	8618122	5	5	5

상반기 RFM

cust	recency	frequency	monetary	recency_score	frequency_score	monetary_score
M000034966	8	4	256160	3	1	2
M000136117	1	29	8556060	5	4	5
M000201112	33	5	53120	2	1	1
M000225114	0	43	1124520	5	5	4
M000261625	27	29	4443700	2	4	5
...	...	...	...	...	...	...
M999599111	47	3	26100	2	1	1
M999673157	13	9	2682570	3	2	5
M999770689	0	67	680600	5	5	3
M999849895	30	17	511442	2	3	3
M999962961	0	53	3229126	5	5	5

하반기 RFM

RFM 점수 매기기

각 고객에게 RFM 분석 모형을 이용해 RFM 총 점수를 매김

RFM 분석 모형

RFM = {(w1 x R + w2 x F + w3 x M) x 100} / m

RFM 분석 모형에서 m은 R, F, M이 가질 수 있는 값의 범위를 나눈 구간 수이고 R, F, M ∈ {1, 2, 3, .., , m}

R, F, M을 각각 5구간으로 나누고 각각 0.3, 0.2, 0.5의 가중치를 줌

RFM = {(0.3 x R + 0.2 x F + 0.5 x M) x 100} / 5 (2)

```
[ ]  df_rfm['rfm_score'] = (0.3*df_rfm['recency_score'] + 0.2*df_rfm['frequency_score'] + 0.5*df_rfm['monetary_score']) * 100 / 5
     df_rfm['rfm_score'].astype(int)
```

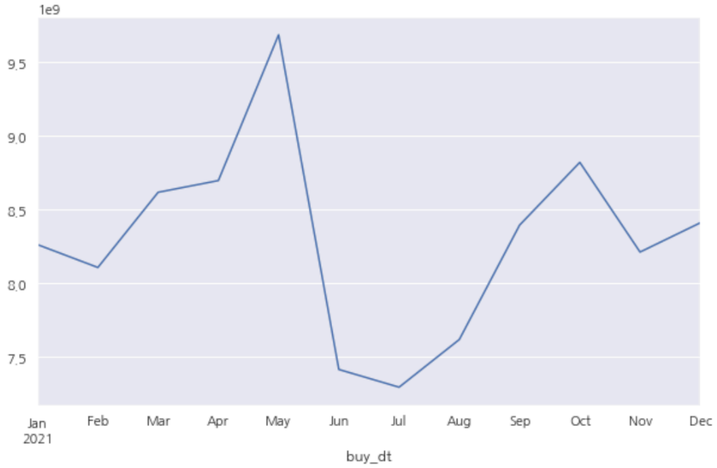
코드 - 가중치합 계산

RFM 가중치 선정이유

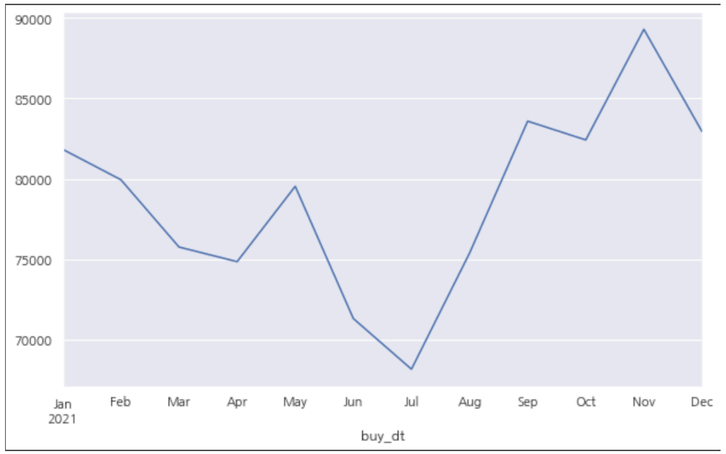
monetary = 0.5로 가중치가 높은 이유

매출액 그래프 감소: 커머스 특성 상 매출액이 중요함. 매출액 성장 저조

두 그래프 추이가 비슷: 구매당 금액은 매출액에 영향을 끼침



월별 매출액



월별 구매당 주문 금액

고객 세그먼트

4개의 세그먼트(최우수고객, 우수고객, 일반고객, 기타고객)

상/하반기로 나 눈 고객별 RFM 점수를 내림차순으로 정렬하여

상위 20%에 해당하는 최우수고객, 그 다음 20%에 해당하는 고객을 우수고객, 그 다음 20%에 해당하는 고객을 일반고객,
나머지를 기타고객으로 분류

cust	recency	frequency	monetary	recency_score	frequency_score	monetary_score	rfm_score	rfm_level
M000034966	8	4	256160	3	1	2	42.0	일반고객
M000136117	1	29	8556060	5	4	5	96.0	최우수고객
M000201112	33	5	53120	2	1	1	26.0	기타고객
M000225114	0	43	1124520	5	5	4	90.0	최우수고객
M000261625	27	29	4443700	2	4	5	78.0	우수고객
...	...	...	...	...	...	...	...	...
M999599111	47	3	26100	2	1	1	26.0	기타고객
M999673157	13	9	2682570	3	2	5	76.0	우수고객
M999770689	0	67	680600	5	5	3	80.0	최우수고객
M999849895	30	17	511442	2	3	3	54.0	일반고객

데이터셋 일부

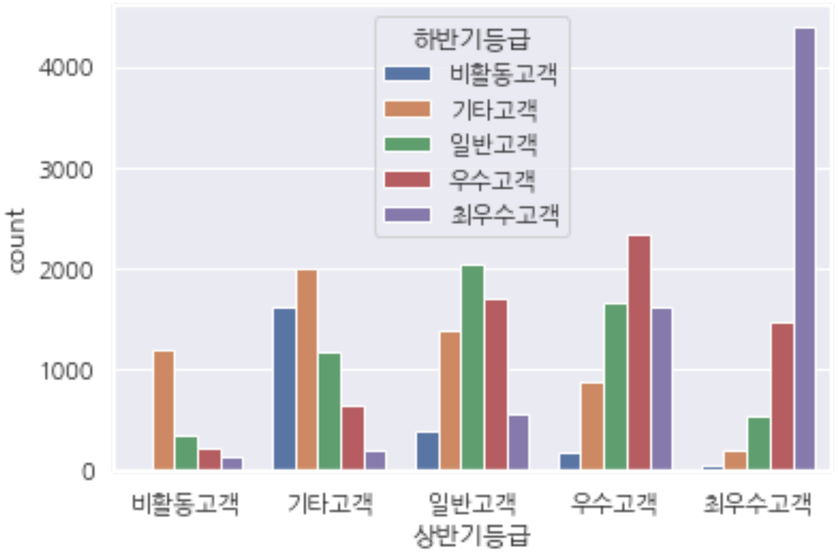
각 기간별 RFM 점수

상/하반기 RFM 점수를 결합하여 상반기에서 하반기의 고객 등급 변화를 살펴봄

결측값은 비활동고객으로 채워 각각 신규/이탈 고객으로 구분함

cust	상반기등급	하반기등급
M000034966	일반고객	일반고객
M000136117	최우수고객	최우수고객
M000201112	비활동고객	기타고객
M000225114	일반고객	최우수고객
M000261625	우수고객	우수고객
...	...	...
M999599111	우수고객	기타고객
M999673157	기타고객	우수고객
M999770689	우수고객	최우수고객
M999849895	우수고객	일반고객
M999962961	최우수고객	최우수고객

상/하반기 세그먼트



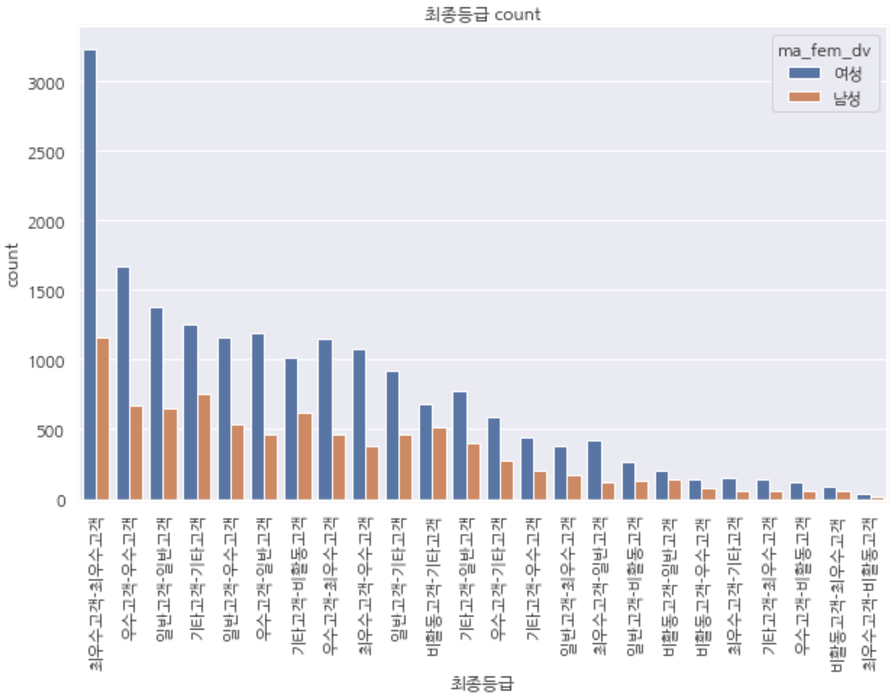
상/하반기 세그먼트 고객 분포

고객 등급 변화 분석

(상반기~하반기)사이에 고객 등급의 변화를 분석

상반기 때 기타고객이었던 고객이 이탈하는 경우는 전체 이탈 중 72%를 차지할 정도로 높았음. 따라서 기타 고객의 이탈을 막는 마케팅 제안

최우수/최우수 같은 충성 고객에 대한 마케팅 제안



고객 세그먼트 변화

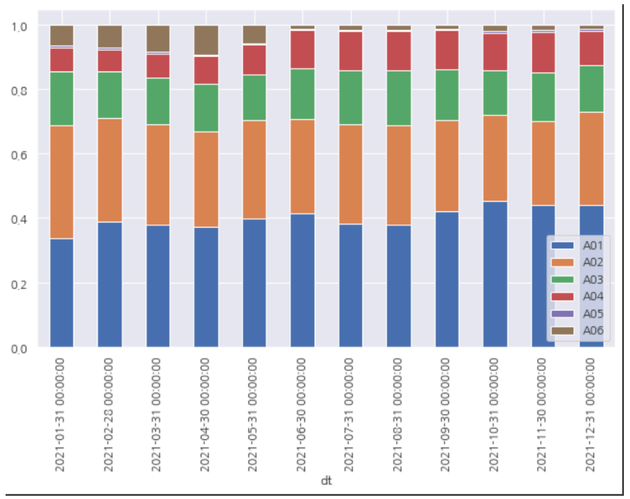
LGBM

LGBM 데이터셋 준비

데이터셋 준비(csv_user_merged_copPlus.csv)

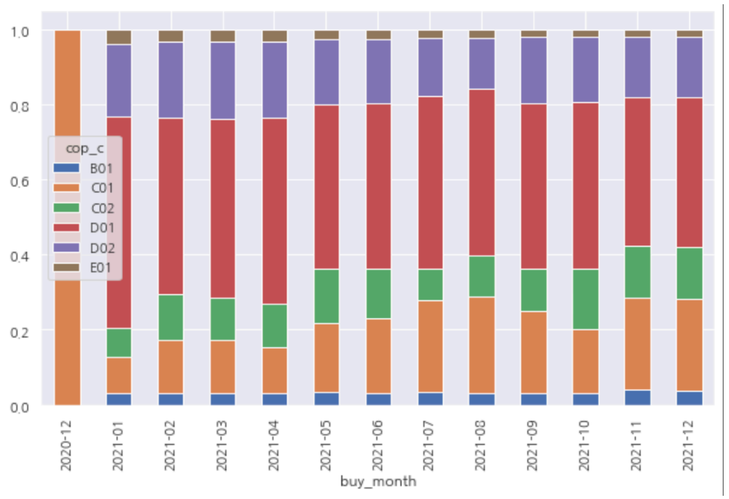
세그먼트 뿐만 아니라 각 제휴사에 대한 고객 순위 정보 추가함

이유: 월별 제휴사별 구매추이를 보면, 제휴사별로 차이가 커서 제휴사 정보를 반영함.



상품구매정보 - 월별 제휴사별 구매건수 추이

월별 제휴사별 구매건수 추이
- A04: 6월부터 비중이 갑자기 확대
- A06: 6월 이후 비중 갑자기 축소



제휴사이용정보 - 월별 제휴사별 이용건수 추이

월별 이용 건수 상대 비교
- F&B 비중 축소 (D01, D02)
- 엔터테인먼트 비중 확대 (C01, C02)
- 숙박 큰 변동 없음
- 렌탈 축소

LGBM 구현

FLAML 학습

automl.fit()를 통해 LGBMClassfier의 가장 좋은 파라미터를 찾음.

```
automl.fit(x_train, y_train, task="classification", time_budget=3600*5, n_jobs=6, n_concurrent_trials=2, log_file_name='segment_automl.log', estimator_list=["lgbm"])
```

```
== Status ==
Current time: 2022-08-10 15:10:37 (running for 05:00:04.65)
Memory usage on this node: 17.1/187.5 GiB
Using FIFO scheduling algorithm.
Resources requested: 0/12 CPUs, 0/3 GPUs, 0.0/133.06 GiB heap, 0.0/30.39 GiB objects (0.0/1.0 accelerator_type:TITAN)
Current best trial: d442f308 with val_loss=1.7156423027740288 and parameters={'n_estimators': 1396, 'num_leaves': 4, 'min_child_samples': 2, 'learning_rate': 0.018951508635514553, 'log_max_bin': 6,
'colsample_bytree': 0.6492182214919888, 'reg_alpha': 0.5607567989480035, 'reg_lambda': 0.0009765625, 'learner': 'lgbm'}
Result logdir: /ray_results/train_2022-08-10_10-10-32
Number of trials: 366/1000000 (366 TERMINATED)
```

베스트 탐색 결과

```
LGBMClassifier(colsample_bytree=0.6492182214919888,
               learning_rate=0.018951508635514553, max_bin=63,
               min_child_samples=2, n_estimators=1396, n_jobs=6, num_leaves=4,
               reg_alpha=0.5607567989480035, reg_lambda=0.0009765625,
               verbose=-1)
```

LGBM 학습 결과

피처 중요도

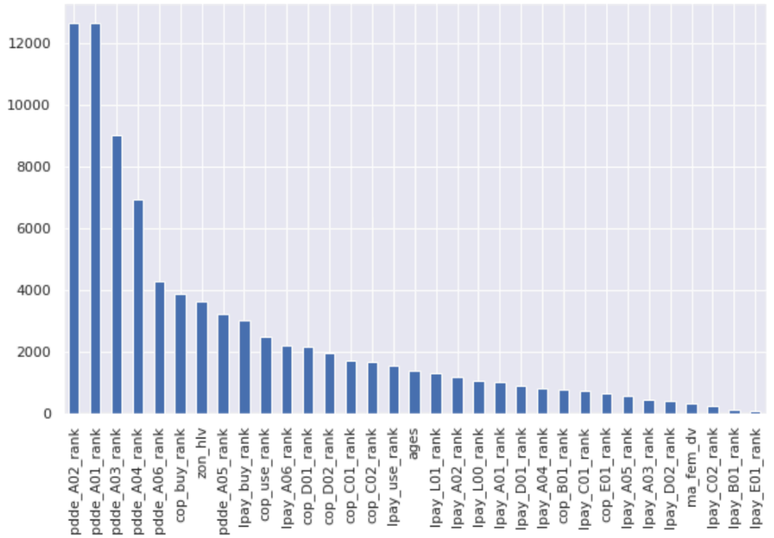
유통사 구매 순위의 피처 중요도가 크다. 특히 A01, A02

f1 score Top-2

기타고객/비활동고객의 f1-score: 0.73

최우수고객/최우수고객의 f1-score: 0.74

이탈고객인 기타고개/비활동고객의 예측 신뢰성이 높음



피처 중요도

	precision	recall	f1-score	support
기타고객/기타고객	0.56	0.80	0.66	556
기타고객/비활동고객	0.70	0.77	0.73	299
기타고객/우수고객	0.00	0.00	0.00	130
기타고객/일반고객	0.23	0.19	0.21	245
기타고객/최우수고객	0.00	0.00	0.00	37
우수고객/기타고객	0.10	0.01	0.01	132
우수고객/비활동고객	0.00	0.00	0.00	16
우수고객/우수고객	0.28	0.57	0.38	587
우수고객/일반고객	0.17	0.06	0.08	354
우수고객/최우수고객	0.18	0.08	0.11	383
일반고객/기타고객	0.23	0.14	0.18	274
일반고객/비활동고객	0.16	0.04	0.07	71
일반고객/우수고객	0.14	0.08	0.11	355
일반고객/일반고객	0.22	0.44	0.30	411
일반고객/최우수고객	0.00	0.00	0.00	133
최우수고객/기타고객	0.00	0.00	0.00	30
최우수고객/비활동고객	0.00	0.00	0.00	3
최우수고객/우수고객	0.25	0.08	0.13	347
최우수고객/일반고객	0.21	0.03	0.05	110
최우수고객/최우수고객	0.65	0.86	0.74	1016
accuracy			0.41	5489
macro avg	0.20	0.21	0.19	5489
weighted avg	0.34	0.41	0.35	5489

f1-score report

LGBM 결과보고

LGBM에서 나온 정확도 높은 두 세그먼트의 고객 분석 (EDA)

최우수-최우수

기타-비활동 (이탈고객)

lgbm 결과보고 내용 상세

1. 온오프 통계분석

2. 상품구매정보분석

3. 제휴이용정보분석

4. 엘페이 이용률

	precision	recall	f1-score	support
기타고객/기타고객	0.56	0.80	0.66	556
기타고객/비활동고객	0.70	0.77	0.73	299
기타고객/우수고객	0.00	0.00	0.00	130
기타고객/일반고객	0.23	0.19	0.21	245
기타고객/최우수고객	0.00	0.00	0.00	37
우수고객/기타고객	0.10	0.01	0.01	132
우수고객/비활동고객	0.00	0.00	0.00	16
우수고객/우수고객	0.28	0.57	0.38	587
우수고객/일반고객	0.17	0.06	0.08	354
우수고객/최우수고객	0.18	0.08	0.11	383
일반고객/기타고객	0.23	0.14	0.18	274
일반고객/비활동고객	0.16	0.04	0.07	71
일반고객/우수고객	0.14	0.08	0.11	355
일반고객/일반고객	0.22	0.44	0.30	411
일반고객/최우수고객	0.00	0.00	0.00	133
최우수고객/기타고객	0.00	0.00	0.00	30
최우수고객/비활동고객	0.00	0.00	0.00	3
최우수고객/우수고객	0.25	0.08	0.13	347
최우수고객/일반고객	0.21	0.03	0.05	110
최우수고객/최우수고객	0.65	0.86	0.74	1016
accuracy			0.41	5489
macro avg	0.20	0.21	0.19	5489
weighted avg	0.34	0.41	0.35	5489

온오프 통계 분석

KL-divergence: 분포 간 거리 비교를 위해 활용

채널, 제휴사 별 카운트 분포화 정렬 (P=기타/비활성고객, Q=전체고객)

결과

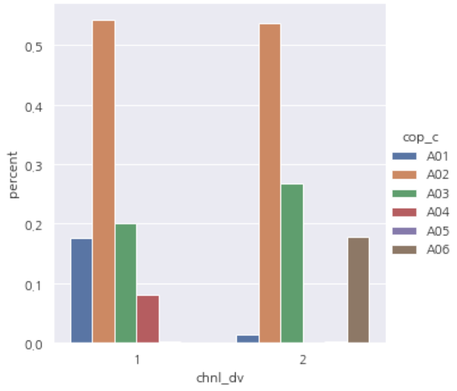
기타-비활동 고객의 온라인 A06 분포가 가장 도드라지게 나타남

온라인 마케팅 시 A06의 비중을 확대하고 엘페이 사용을 제안함

효과적인 엘페이 활성화가 기대됨

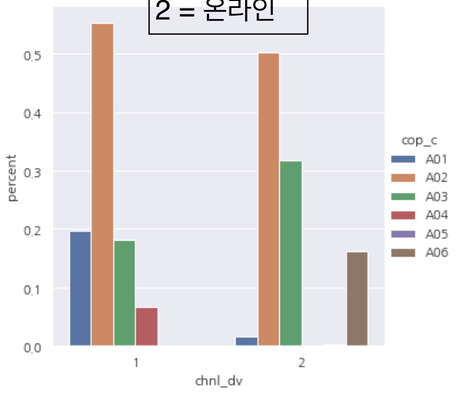
KL-divergence 결과

chnl_dv	cop_c	
1	A02	-0.095287
	A01	-0.037293
2	A03	-0.002205
1	A03	-0.001247
2	A05	0.000882
	A02	0.028749
	A01	0.039198
1	A04	0.039708
	A05	0.049149
2	A06	0.129672

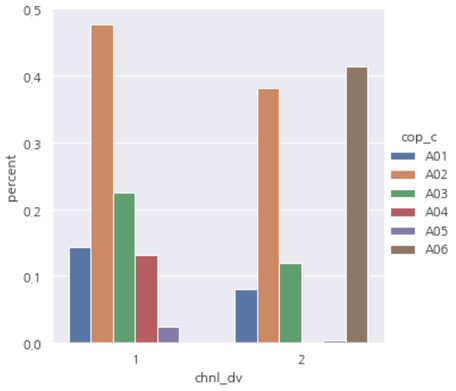


전체 고객

1 = 오프라인
2 = 온라인



최우수/최우수 고객



기타-비활동고객

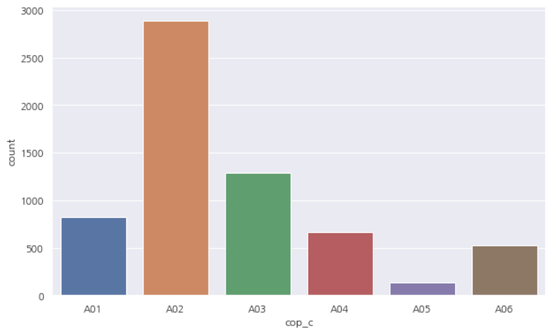
상품 구매 정보 분석

상품 구매 정보 분석

상품 구매 정보 데이터의 cop_c.count()를 분포화하여 **KL-divergence** 이용해서 정렬 **(P=기타/비활성, Q=전체고객)**

결과

기타/비활동 고객의 A06 분포가 가장 도드라지게 나타남



기타/비활동고객 유통사 카운트

A02	-0.077366
A01	-0.026490
A03	-0.003531
A04	0.039708
A05	0.048880
A06	0.129672

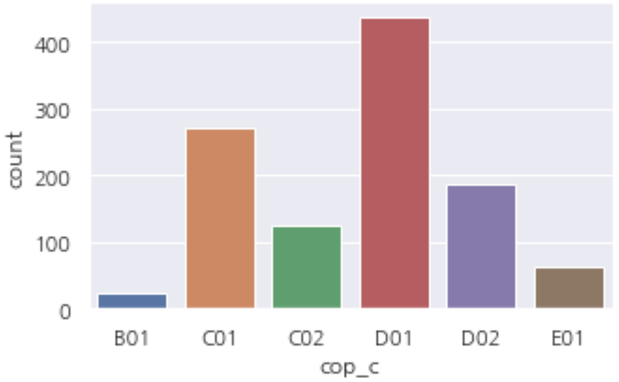
KL-divergence 결과

제휴 이용 정보 분석

제휴 이용 정보 데이터의 cop_c.count()를 분포화하여 **KL-divergence** 이용해서 정렬 **(P=기타/비활성, Q=전체고객)**

결과

기타/비활동 고객의 C01(엔터테인먼트) 분포가 가장 도드라지게 나타남



기타/비활동고객 제휴사 카운트

D01	-0.052354
C02	-0.009343
B01	-0.008854
D02	-0.005452
E01	0.046484
C01	0.056444

KL-divergence 결과

엘페이 이용률

새로운 데이터 셋 설명

한 행의 정보: 고객 한 명의 정보

lpay_prob = lpay_rct_no(엘페이 구매 건수) / rct_no(유통사/제휴사 구매 건수)

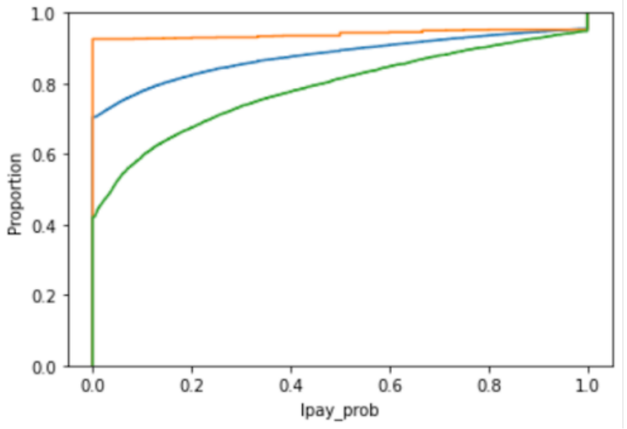
그래프 해석 (lpay_prob의 누적분포(ecdf) 그래프)

파란색 - 전체고객, **주황색** - 기타고객/비활동고객, **초록색** - 최우수고객/최우수고객

x축이 0인 경우, lpay 한번도 결제 안 한 경우

lpay를 아예 결제 안 한 경우(lpay_prob=0), 최우수/최우수 고객 < 전체 고객 < 기타/비활동고객

결론: (비활동 => 최우수)로 갈수록 엘페이 이용률 상승하므로, 엘페이 추가 할인 등의 마케팅 전략을 통해 **엘페이 활성화**를 기대할 수 있음



그래프 결과

cust_no	rct_no	lpay_rct_no	cust	label
0	13	0.0	M000034966	일반고객/일반고객
1	1	0.0	M000059535	기타고객/비활동고객
2	85	4.0	M000136117	최우수고객/최우수고객
3	5	0.0	M000201112	신규고객
4	88	0.0	M000225114	일반고객/최우수고객
...	...	...	...	...
29751	2	0.0	M999708287	신규고객
29752	137	79.0	M999770689	최우수고객/최우수고객
29753	35	0.0	M999849895	일반고객/일반고객
29754	7	0.0	M999926092	일반고객/기타고객
29755	138	16.0	M999962961	최우수고객/최우수고객

29756 rows × 4 columns

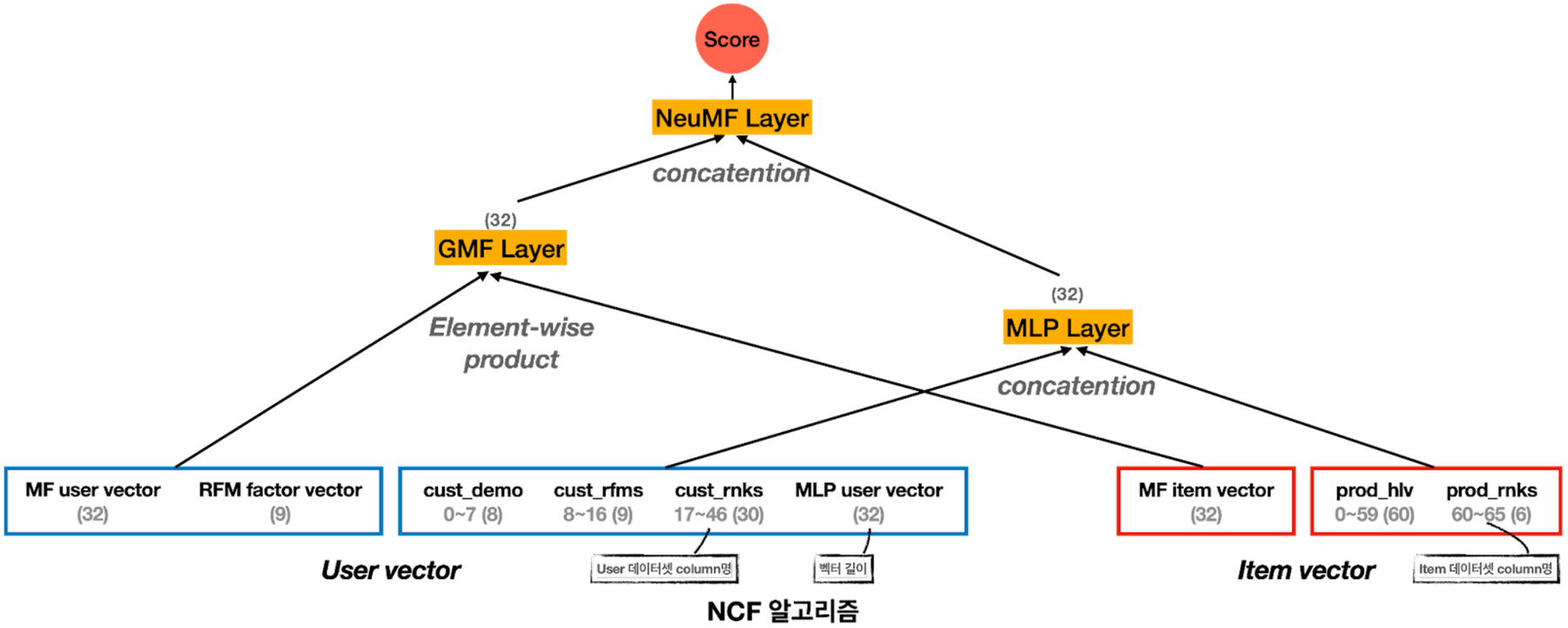
데이터셋(df_nlpay) 설명

NCF

상품 추천 모델 – Neural Collaborative Filtering

NCF 모델 알고리즘 summary

상품구매정보, 제휴이용, 엘페이이용, 데모 결합한 추천 모델



데이터 전처리 for NCF

전처리 for User 벡터: 속성 cust 기준으로 1~4까지 차례로 LEFT JOIN

1. 고객 데모 정보

    고객 데모 정보와 라벨링한 고객 세그먼트 정보를 Merge

    이때, 세그먼트 정보는 LGBM과 달리, 유통사 정보만으로 라벨링한 세그먼트 정보임

2. 상품 구매 정보

    전체 구매 정보의 경우는 RFM 에 총 구매 빈도와, 구매 금액 랭크가 반영되어 있으므로 유통사 코드(A01~A06)에 대한 빈도만 추가함

3. 제휴사 이용 정보

    제휴사 전체 이용 빈도/제휴사(B01~E01) 개별 이용 빈도 등수, 금액 등수

4. 엘페이 이용 정보

    엘페이 전체 이용 빈도/제휴사(A01~L01) 개별 이용 빈도 등수, 금액 등수

5. cust 인덱스값

전처리 for Item 벡터: 속성 pd_c 기준으로 1, 2를 Merge

1. 상품 분류 정보

    상품 코드, 대분류명

2. 상품 구매 정보

    상품별 유통사 코드(A01~A06) 빈도 등수

3. pd_c 인덱스값

전처리 for label

1. 사용자-아이템 인덱스 사전: 상품구매 정보 파일로부터 요약및 가공한 데이터프레임

상품 구매 정보(cust, pd_c) + Item 벡터(pd_c_no) + User 벡터(cust_no): Merge

2. label 속성 추가

추천 알고리즘을 적용하기 위해 고객-상품 구매가 여러 건 이라도 한 건으로 처리함

implicit feedback '1'로 처리

구매 횟수 표시하지 않고, 구매/미구매만 표시

네거티브 피드백(미구매 정보)는 없음

- 포지티브 피드백 있음: 구매 상품(1)

- 네거티브 피드백 없음: 미구매 상품(없음)

네거티브 샘플링을 통한 데이터셋 구성

NCFData()에서 수행

방법: 구매하지 않은 상품 목록에서 학습 에포크 마다 랜덤 생성

데이터셋 분할

df_data_train 학습: 1,162,936

df_data_validation 검증: 290,735

df_data_test 테스트: 161,52

* 범주형 데이터 처리 one-hot encoding
  : 범주형 변수를 수치형 변수로 인코딩 처리함
* 상/하반기 데이터 구분하지 않음

NCF 데이터 셋 상세설명 - 입력데이터

user 벡터(X) - csv_np_user.csv

0 ~ 16: one-hot encoding vector 모음

0 ~ 1: 성별 정보, 2 ~ 7: 연령 정보, 8 ~ 11: 상반기 고객 세그먼트 구분값, 12 ~ 16: 하반기 고객 세그먼트 구분값

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
남성	여성	20대	30대	40대	50대	60대	70대	상_기타고객	상_우수고객	상_일반고객	상_최우수고객	하_기타고객	하_비활동고객	하_우수고객	하_일반고객	하_최우수고객

17 ~ 46: continuous vector 모음

17 ~ 22: “상품 구매 정보”의 **각** 유통사 이용 랭크

23 ~ 24: “제휴사 이용 정보”의 **전체** 이용 횟수 랭크, 이용 금액 랭크, 25 ~ 30: “제휴사 이용 정보”의 **각** 제휴사 이용 랭크,

31 ~ 32: “엘페이 이용 정보”의 **전체** 이용 횟수 랭크, 이용 금액 랭크, 33 ~ 46: “엘페이 이용 정보”의 **각** 유통사/제휴사 이용 랭크

47: 고객 인덱스값 (숫자로 된 간접 번호)

17	18	19	20	21	22	23	24	25	26	...	38	39	40	41	42	43	44	45	46	47
pdde_A01_rank	pdde_A02_rank	pdde_A03_rank	pdde_A04_rank	pdde_A05_rank	pdde_A06_rank	cop_use_rank	cop_buy_rank	cop_B01_rank	cop_C01_rank	...	lpay_A06_rank	lpay_B01_rank	lpay_C01_rank	lpay_C02_rank	lpay_D01_rank	lpay_D02_rank	lpay_E01_rank	lpay_L00_rank	lpay_L01_rank	cust_no

item 벡터(X) - csv_np_item.csv

0 ~ 59: one-hot encoding vector: clac_hlv_nm 대분류

0	1	2	3	4	5	6	7	8	9	...	50	51	52	53	54	55	56	57	58	59
가구	건강식품	건강용품	건해산물	계절가전	공구/안전용품	과일	과자	구기/필드스포츠	금융/보험서비스	...	축산물	출산/육아용품	침구/수예	커피/차	컴퓨터	테넌트/음식점	패션잡화	퍼스널케어	헬스/피트니스	화장품/뷰티케어

60 ~ 65: continuous vector: 상품이 각 유통사를 통해 구매된 횟수를 유통사별로 랭크화

66: 상품 인덱스값: (‘숫자로 된 간접 번호’)

60	61	62	63	64	65	66
A01_rank	A02_rank	A03_rank	A04_rank	A05_rank	A06_rank	pd_c_no

NCF 모델 아키텍처

---

1) Classification Loss

BCEwithlogitloss

2) Target(label)

값: 0 (비구매) 또는 1 (구매)

미구매 정보는 없으므로 네거티브 샘플링 필요

- 구매한 상품 = 포지티브 피드백
- 네거티브 샘플링한 상품 = 네거티브 피드백

3) 네거티브 샘플링을 통한 데이터셋 구성

네거티브 샘플링 수행 함수

Trainer에서 매 학습 에포크 시작시 콜백을 통해 호출됨.

절차

1. 고객의 구매 상품 한 건당 네거티브 샘플 self.num_ng를 샘플링하여 추가
2. 고객마다 구매한 적이 없는 후보 셋을 모아둔 후, 이로부터 self.num_ng 만큼 네거티브 샘플링
3. 이를 포지티브 샘플들과 연결하여 self.samples 구성

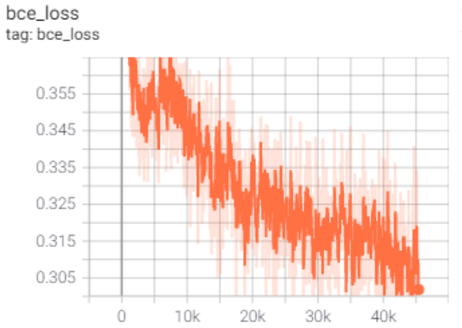
4) pytorch_lightning으로 NCF 구현

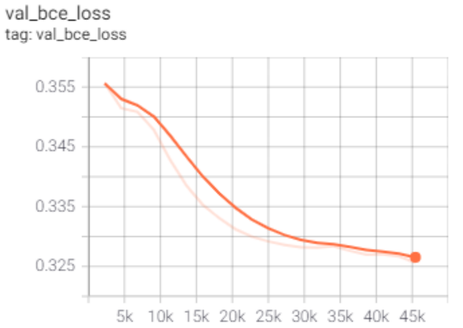
- 20 에포크
- Binary cross entropy 측정

NCF 학습 결과

loss 그래프 : 검증셋에 대한 로스(val_bce_loss)가 꾸준히 감소

결과 : val_bce_loss에 의해 학습이 잘 되고 있다는 것을 알 수 있음





테스트셋에 대한 HitRatio 조사

- 네거티브샘플링
- 포지티브 샘플 1개 당 49개의 네거티브 샘플 생성

즉, 총 50개의 데이터를 묶음으로 순위 결정

- 예측값으로 내림차순 정렬
- 상위 K 등 내에 포지티브 샘플이 위치하면 HIT
- K를 1부터 9까지 변화하며 조사

	HR
k	
1	0.045123
2	0.087498
3	0.132164
4	0.171194
5	0.212378
6	0.252920
7	0.289706
8	0.327638
9	0.367584

해석: 고객에게 9개 추천하면 그중 한 개는 구매할 가능성이 36.7%

결과 : 학습시, 테스트셋의 포지티브 샘플이 전혀 사용되지 않고도 약 30%의 히트율이 얻어지므로 상품 추천으로 활용할 수 있을 것으로 기대

마케팅전략

마케팅전략

LGBM & NCF

“정교한 예측 세그먼트에 구매전환이 높을 상품 추천 "

inactive로 전환 예상되는 고객에게 선제 대응

추천 상품을 제안 (앱푸시, 이메일, SMS 등 매체 이용)

해당 상품 구매시 엘페이 추가 할인

LGBM - 세그먼트 분석을 통한 마케팅 전략

이탈 고객(Non Active) (중요)

전략 - 구매빈도확대

방법 - 관심 상품이나 제휴사와 관련된 구매 활동 시 할인이나 관련 쿠폰 발급

제휴사 A06, C01: 관련 온라인 쿠폰 발급, 엘페이로 결제 시 추가 할인

충성 고객(Loyal)

전략 - 충성도 유지

방법 - 차별적인 부가 서비스 제공, 모바일 쿠폰북 제공

마케팅전략

LGBM & NCF

“정교한 예측 세그먼트에 구매전환이 높을 상품 추천 "

inactive로 전환 예상되는 고객에게 선제 대응

추천 상품을 제안 (앱푸시, 이메일, SMS 등 매체 이용)

해당 상품 구매시 엘페이 추가 할인

LGBM - 세그먼트 분석을 통한 마케팅 전략

이탈 고객(Non Active) (중요)

전략 - 구매빈도확대

방법 - 관심 상품이나 제휴사와 관련된 구매 활동 시 할인이나 관련 쿠폰 발급

제휴사 A06, C01: 관련 온라인 쿠폰 발급, 엘페이로 결제 시 추가 할인

충성 고객(Loyal)

전략 - 충성도 유지

방법 - 차별적인 부가 서비스 제공, 모바일 쿠폰북 제공

사용한 라이브러리

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from glob import glob
import os
import tqdm
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.utils.data as data
import pytorch_lightning as pl
from typing import List, Optional, Tuple, Union
import argparse
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder, MinMaxScaler
from datetime import datetime
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error, classification_report
import lightgbm as lgb
import ray
from flaml import AutoML
from flaml.data import get_output_from_log
from IPython.display import display, HTML
```