2023 BDA 데이터분석 활용 공모전

Track2 : 모델링 고도화

MOMA 김지오 김수빈 박기정 임재성

1. 개요

2. 데이터 전처리

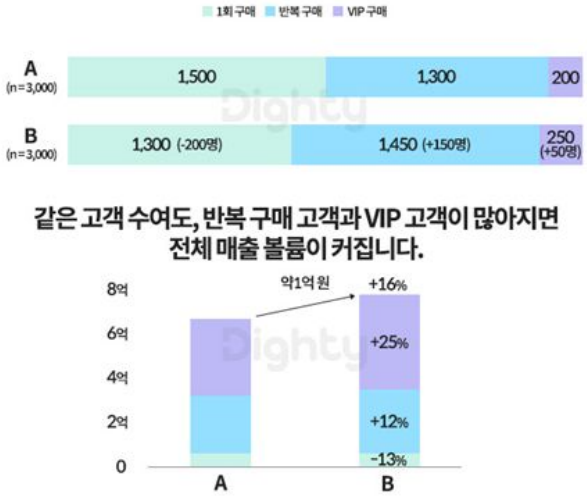
3. 모델링

4. 결과

5. Q&A

1. 개요

1) 분석 목적 및 필요성



출처:https://blog.dighty.com/recipe/?q=YToyOntzOjEyOiJrZXl3b3JkX3R5cGUiO3M6MzoiYWxsIjtzOjQ6InBhZ2UiO2k6MTt9&bmode=view&idx=11207416&t=board

출처 : www.opensurvey.co.kr

CJ더마켓의 ‘the 프라임’ 가입 고객 수를 증가시켜 반복구매자 및 VIP 고객을 확보하고 궁극적으로 매출의 증대를 꾀하고자 한다.

1. 개요

2) 데이터 파악

```
[ ]  df.info() # 데이터 각 컬럼 type 파악
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 45875 entries, 0 to 45874
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   scd            45875 non-null  int64
 1   product_name   45875 non-null  object
 2   net_order_qty  45875 non-null  int64
 3   net_order_amt  45875 non-null  float64
 4   gender         45875 non-null  object
 5   age_grp        45875 non-null  int64
 6   employee_yn    45875 non-null  object
 7   order_date     45875 non-null  int64
 8   prime_yn       45875 non-null  object
dtypes: float64(1), int64(4), object(4)
memory usage: 3.5+ MB
```

```
df.isnull().sum() # 컬럼별 결측치 확인
                  # 없음
```

```
scd             0
product_name    0
net_order_qty   0
net_order_amt   0
gender          0
age_grp         0
employee_yn     0
order_date      0
prime_yn        0
dtype: int64
```

```
[ ]  df.duplicated().sum() # 중복 데이터 확인
                           # 없음
```

```
0
```

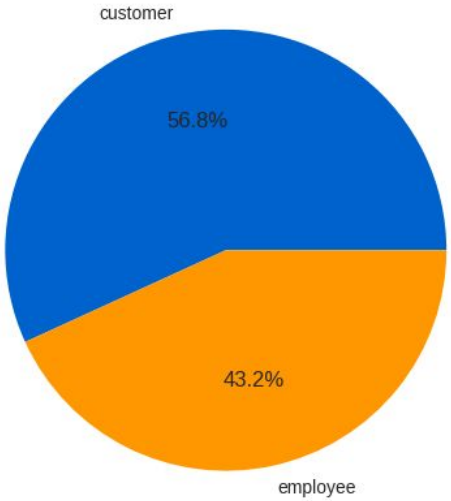
1. 개요

3) 가설설정

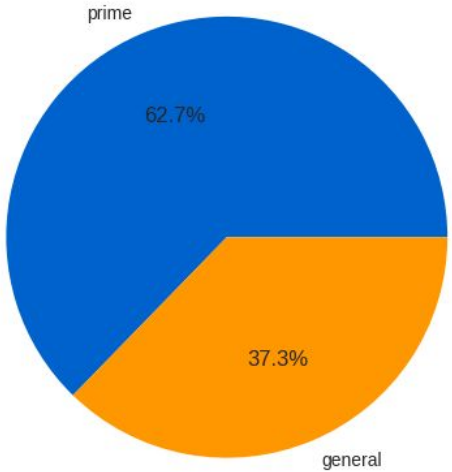
"일반회원과 임직원의 프라임/비프라임 회원 간
구매 특성에 차이가 있을 것이다."

2. 데이터 전처리

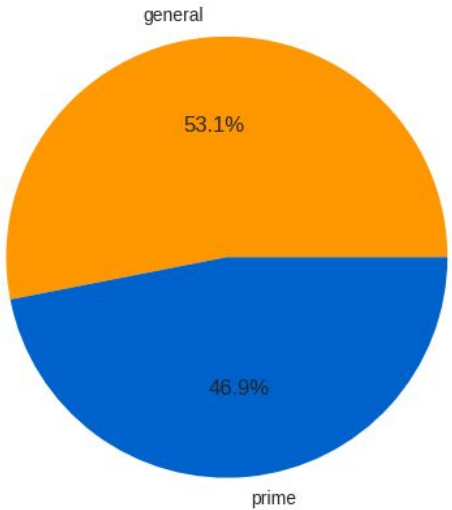
1) EDA - employee_yn, prime_yn 파악



employee_yn 비율



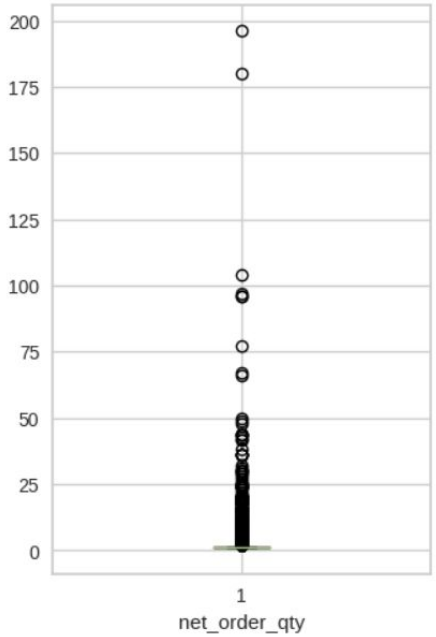
prime_yn 비율 in employee

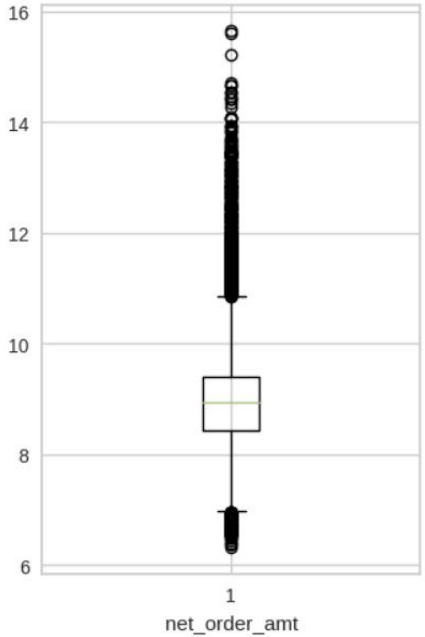


prime_yn 비율 in customer

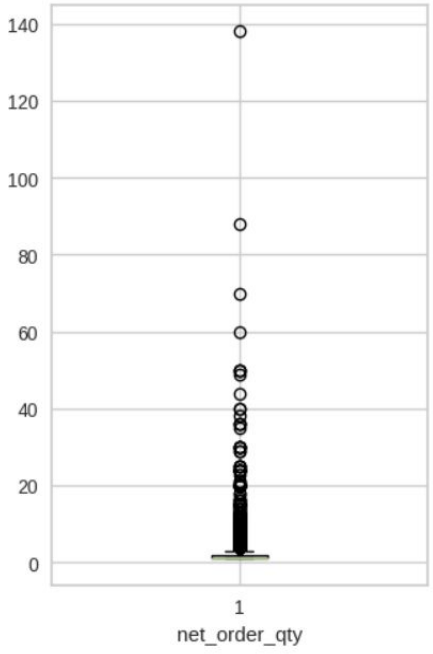
2. 데이터 전처리

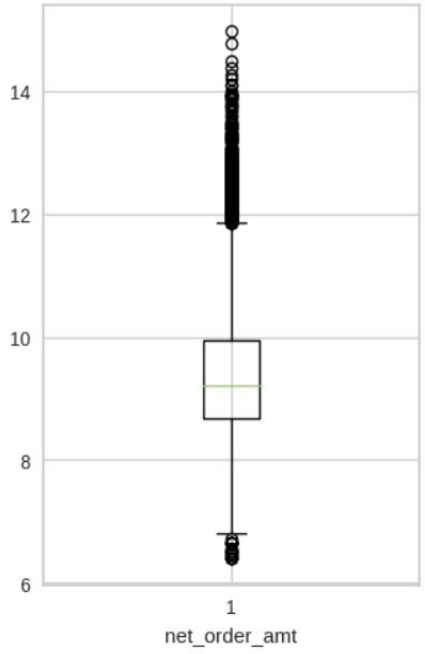
1) EDA - 이상치 확인





customer





employee

2. 데이터 전처리

1) EDA - 이상치 파악

데이터	이상치 후보 판단 기준	이상치 후보 개수
Employee	Z-score 3 이상 & Q1, Q3 기준 1.5*IQR 벗어나는 경우	454
	Z-score 3 이상 & Q1, Q3 기준 3*IQR 벗어나는 경우	412
Customer	Z-score 3 이상 & Q1, Q3 기준 1.5*IQR 벗어나는 경우	449
	Z-score 3 이상 & Q1, Q3 기준 3*IQR 벗어나는 경우	395

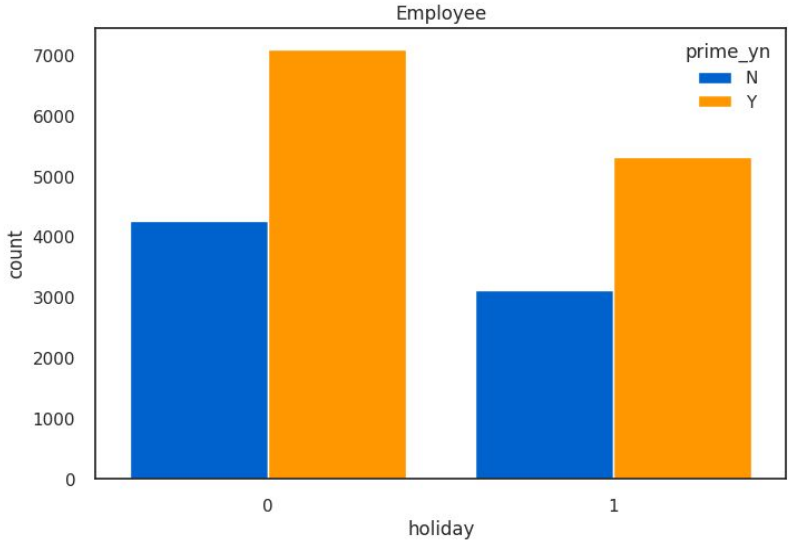
➔ 위에서 나온 이상치 후보들은 상품을 여러 개 구매하였기 때문에 발생했다고 판단

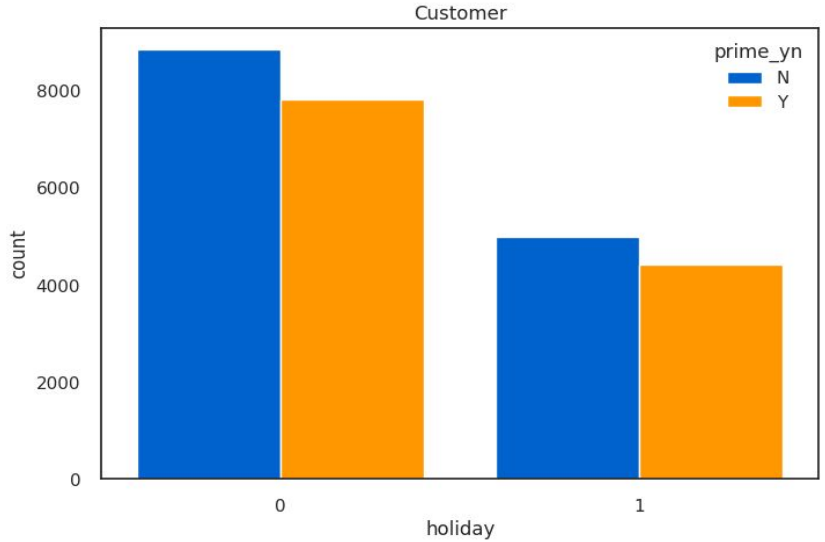
➔ 상품을 여러 개 구매한 사람은 선물세트를 구매한 고객으로, 1월에 명절 선물로 구매하는 주 고객층이라고 판단

➔ **따라서 이상치로 판단하지 않고 분석에 사용함**

2. 데이터 전처리

2) order_date -› holiday





➔ order_date 컬럼에서 해당 날짜가 비영업일이면 1, 영업일이면 0

➔ **Employee와 Customer에서 영업일별로 각각 반대의 결과 나타남**

하지만 같은 데이터 내 프라임과 비프라임 회원과의 차이에서는 같은 경향 보임

2. 데이터 전처리

3-1) product_name -> set_yn

	product_name	set_yn
0	[100개한정] 리턴업 아이시안 루테인 지아잔틴100mgX30캡슐(1개월)X2개	1.0
1	[임직원]이너비 콜렉티브 콜라겐 930mgX84정(4주)X6개	1.0
2	리턴업 아이시안 루테인500mgX30캡슐(1개월)X6개	1.0
3	이너비 콜렉티브 콜라겐 930mgX84정(4주)X2개	1.0
4	이너비 아쿠아뱅크 300mgX56캡슐(4주)X2개	1.0
...	...	...
2878	꼬마 돈까스 400gX4개+꼬마너겟 320gX4개	1.0
2879	[식물성] 비비고 플랜테이블왕교자 420gX2번들X2개	1.0
2880	유자샐러드소스 250gX2개	1.0
2881	[CJ공식몰_23설선물세트특가]한뿌리 맛있는 양배추 80MLX30입X2개	1.0
2882	비비고 특양지곰탕 700g+비비고 왕교자 1.05kgX2개+비비고 남도떡갈비 450...	1.0

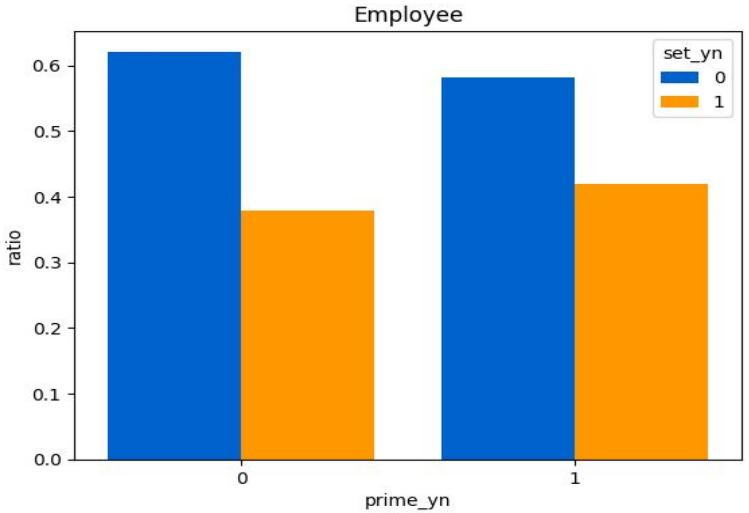
3104 rows × 2 columns

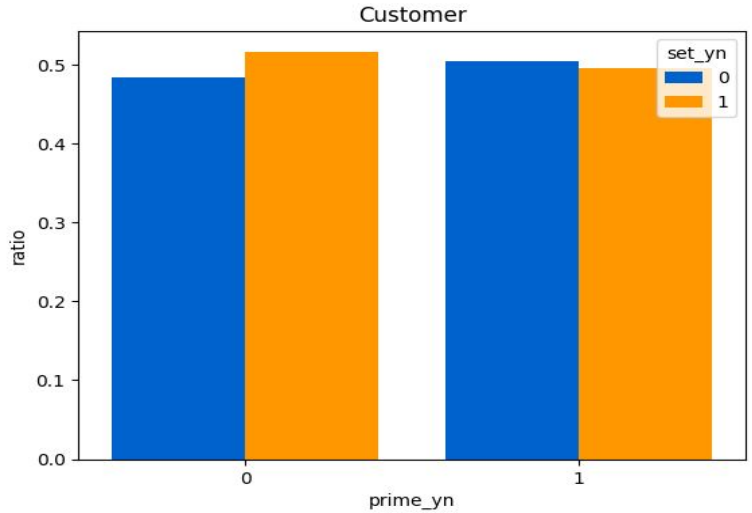
set_yn 변수 생성

➔ 개별 상품인지 여러개의 상품을 묶어서 판매한 상품인지 여부를 파악한 파생변수

➔ 총 3104개의 unique한 product에 대해서 제품이 묶음상품이면 1, 개별상품이면 0으로 설정

직원 / 고객 데이터 각각의 프라임 회원 여부에 따른 묶음상품 여부의 비율





➔ Employee의 경우 프라임 회원의 묶음상품 구매비율이 비프라임 회원에 비해 상대적으로 높음

➔ Customer의 경우 프라임 회원의 개별상품 구매비율이 비프라임 회원에 비해 상대적으로 높음

➔ 하지만 프라임/비프라임 회원 간의 묶음상품 구매비율의 차이가 유의미하게 나타나지 않음

2. 데이터 전처리

3-2) product_name -> category

	product_name	category
0	잔칫집 식혜 240ml 30입	10
1	백설 한입쏙 비엔나 120gX2	6
2	비비고 왕교자 1.05kg	3
3	고메 바삭쫄깃한 탕수육 900g	3
4	햇반 매일잡곡밥210g	1
...	...	...
45870	고메 거멍 모짜체다핫도그 340g	4
45871	[앱전용특가]비비고 차돌된장찌개 460gX4개	2
45872	[앱전용특가]비비고 차돌된장찌개 460gX4개	2
45873	[식물성]고메 플랜테이블 함박스테이크 150g	5
45874	리턴업 전립소 쏘팔메토 골드 1000mgX60캡슐(2개월)	8

13 : 스팸 & 오일 선물세트

14 : 상품이 혼합되어 특정 카테고리로 분류가 불가능한 상품인 경우
ex) 임직원생일선물, 투썸스페셜 기프트세트, 한정set (쯔란갈비/마장반면)

2. 데이터 전처리

3-2) category - Apriori 분석 (임직원)

========== employee_prime apriori ==========

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
96	(1, 3, 4, 6, 7)	(2)	0.015548	0.344170	0.015194	0.977273	2.839509	0.009843	28.856537
50	(10, 3, 4)	(2)	0.012367	0.344170	0.012014	0.971429	2.822529	0.007758	22.954064
65	(1, 10, 3, 4)	(2)	0.011661	0.344170	0.011307	0.969697	2.817497	0.007294	21.642403
107	(1, 3, 4, 5, 6, 7)	(2)	0.011307	0.344170	0.010954	0.968750	2.814746	0.007062	20.986572
93	(1, 3, 4, 5, 7)	(2)	0.016961	0.344170	0.016254	0.958333	2.784480	0.010417	15.739929
77	(1, 4, 6, 7)	(2)	0.024735	0.344170	0.023675	0.957143	2.781021	0.015162	15.302709
101	(1, 4, 5, 6, 7)	(2)	0.016254	0.344170	0.015548	0.956522	2.779216	0.009953	15.084099
105	(3, 4, 5, 6, 7)	(2)	0.013074	0.344170	0.012367	0.945946	2.748488	0.007868	12.132862
74	(1, 4, 5, 7)	(2)	0.024735	0.344170	0.023322	0.942857	2.739513	0.014809	11.477032
41	(10, 3, 4)	(1)	0.012367	0.335689	0.011661	0.942857	2.808722	0.007509	11.625442
56	(4, 6, 7)	(2)	0.030742	0.344170	0.028975	0.942529	2.738559	0.018395	11.411449
88	(3, 4, 6, 7)	(2)	0.018375	0.344170	0.017314	0.942308	2.737917	0.010991	11.367727
90	(4, 5, 6, 7)	(2)	0.018375	0.344170	0.017314	0.942308	2.737917	0.010991	11.367727
66	(10, 2, 3, 4)	(1)	0.012014	0.335689	0.011307	0.941176	2.803715	0.007274	11.293286
71	(1, 3, 6, 7)	(2)	0.023322	0.344170	0.021908	0.939394	2.729451	0.013882	10.821201
72	(1, 4, 5, 6)	(2)	0.022968	0.344170	0.021555	0.938462	2.726741	0.013650	10.657244
91	(1, 3, 4, 5, 6)	(2)	0.016254	0.344170	0.015194	0.934783	2.716052	0.009600	10.056066
38	(1, 6, 7)	(2)	0.040283	0.344170	0.037456	0.929825	2.701646	0.023592	9.345583
53	(3, 6, 7)	(2)	0.030035	0.344170	0.027915	0.929412	2.700447	0.017578	9.290931
60	(1, 3, 4, 6)	(2)	0.024735	0.344170	0.022968	0.928571	2.698005	0.014455	9.181625

========== employee_general apriori ==========

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
57	(1, 2, 4, 5, 6)	(3)	0.010621	0.229952	0.010621	1.000000	4.348730	0.008179	inf
50	(2, 4, 5, 6)	(3)	0.013808	0.229952	0.013277	0.961538	4.181471	0.010102	20.021243
44	(1, 4, 5, 6)	(3)	0.013277	0.229952	0.012746	0.960000	4.174781	0.009693	19.251195
45	(1, 4, 5, 7)	(3)	0.011683	0.229952	0.011152	0.954545	4.151060	0.008466	16.941052
47	(3, 4, 6, 7)	(1)	0.013808	0.356877	0.012746	0.923077	2.586538	0.007818	8.360595
46	(3, 4, 5, 7)	(1)	0.012215	0.356877	0.011152	0.913043	2.558424	0.006793	7.395911
63	(2, 3, 4, 6, 7)	(1)	0.012215	0.356877	0.011152	0.913043	2.558424	0.006793	7.395911
24	(4, 5, 6)	(3)	0.017525	0.229952	0.015932	0.909091	3.953391	0.011902	8.470526
22	(4, 5, 7)	(2)	0.014870	0.318109	0.013277	0.892857	2.806761	0.008546	6.364312
54	(3, 4, 6, 7)	(2)	0.013808	0.318109	0.012215	0.884615	2.780853	0.007822	5.909719
31	(2, 3, 4, 7)	(1)	0.022305	0.356877	0.019649	0.880952	2.468502	0.011689	5.402230
30	(1, 3, 4, 7)	(2)	0.022305	0.318109	0.019649	0.880952	2.769338	0.012554	5.727881
48	(1, 5, 6, 7)	(3)	0.013277	0.229952	0.011683	0.880000	3.826882	0.008630	6.417065
49	(3, 5, 6, 7)	(1)	0.013277	0.356877	0.011683	0.880000	2.465833	0.006945	5.359356
62	(1, 3, 4, 6, 7)	(2)	0.012746	0.318109	0.011152	0.875000	2.750626	0.007098	5.455125
43	(2, 5, 6, 7)	(1)	0.012746	0.356877	0.011152	0.875000	2.451823	0.006604	5.144981
12	(1, 5, 6)	(3)	0.024960	0.229952	0.021774	0.872340	3.793573	0.016034	6.032041
53	(3, 4, 5, 7)	(2)	0.012215	0.318109	0.010621	0.869565	2.733541	0.006736	5.227828
35	(2, 3, 5, 7)	(1)	0.020181	0.356877	0.017525	0.868421	2.433388	0.010323	4.887732
34	(1, 3, 5, 7)	(2)	0.020181	0.318109	0.017525	0.868421	2.729945	0.011106	5.182369

2. 데이터 전처리

3-2) category - Apriori 분석 (고객)

========== customer_prime apriori ==========

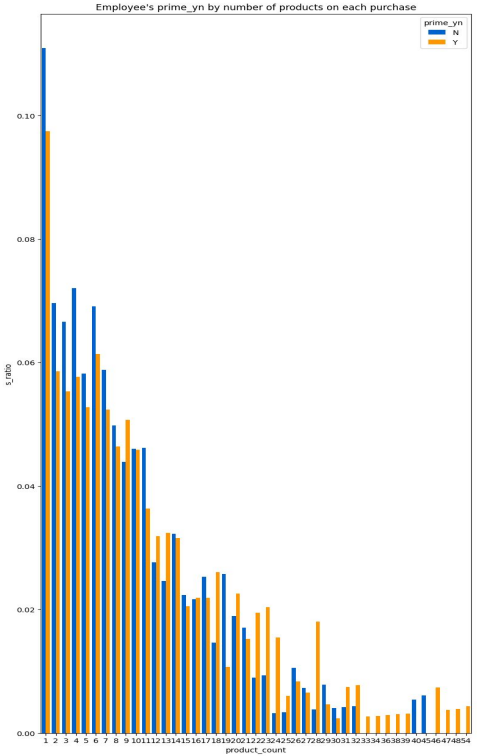
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1	(8, 1)	(2)	0.010421	0.472946	0.010020	0.961538	2.033083	0.005092	13.703407
28	(1, 4, 6, 7)	(2)	0.026453	0.472946	0.024449	0.924242	1.954224	0.011938	6.957114
43	(1, 4, 5, 6, 7)	(2)	0.015631	0.472946	0.014429	0.923077	1.951760	0.007036	6.851703
12	(1, 6, 7)	(2)	0.056914	0.472946	0.052505	0.922535	1.950615	0.025588	6.803789
19	(5, 6, 7)	(2)	0.038477	0.472946	0.035271	0.916667	1.938206	0.017073	6.324649
36	(4, 5, 6, 7)	(2)	0.018838	0.472946	0.017234	0.914894	1.934457	0.008325	6.192886
30	(1, 5, 6, 7)	(2)	0.029659	0.472946	0.026854	0.905405	1.914395	0.012826	5.571715
5	(6, 7)	(2)	0.077756	0.472946	0.069739	0.896907	1.896427	0.032965	5.112425
45	(3, 4, 5, 6, 7)	(2)	0.011222	0.472946	0.010020	0.892857	1.887863	0.004712	4.919172
18	(4, 6, 7)	(2)	0.032866	0.472946	0.029259	0.890244	1.882338	0.013715	4.802049
35	(3, 5, 6, 7)	(2)	0.021643	0.472946	0.019238	0.888889	1.879473	0.009002	4.743487
39	(1, 3, 4, 6, 7)	(2)	0.013226	0.472946	0.011623	0.878788	1.858115	0.005368	4.348196
34	(3, 4, 6, 7)	(2)	0.016032	0.472946	0.014028	0.875000	1.850106	0.006446	4.216433
41	(1, 3, 5, 6, 7)	(2)	0.018036	0.472946	0.015631	0.866667	1.832486	0.007101	3.952906
8	(1, 4, 6)	(2)	0.050902	0.472946	0.044088	0.866142	1.831376	0.020014	3.937404
25	(1, 3, 6, 7)	(2)	0.026453	0.472946	0.022846	0.863636	1.826079	0.010335	3.865063
16	(3, 6, 7)	(2)	0.034068	0.472946	0.029259	0.858824	1.815902	0.013146	3.733300
9	(1, 4, 7)	(2)	0.056112	0.472946	0.048096	0.857143	1.812349	0.021558	3.689379
21	(1, 3, 4, 7)	(2)	0.035671	0.472946	0.030461	0.853933	1.805561	0.013590	3.608294
27	(1, 4, 5, 7)	(2)	0.029259	0.472946	0.024850	0.849315	1.795798	0.011012	3.497723

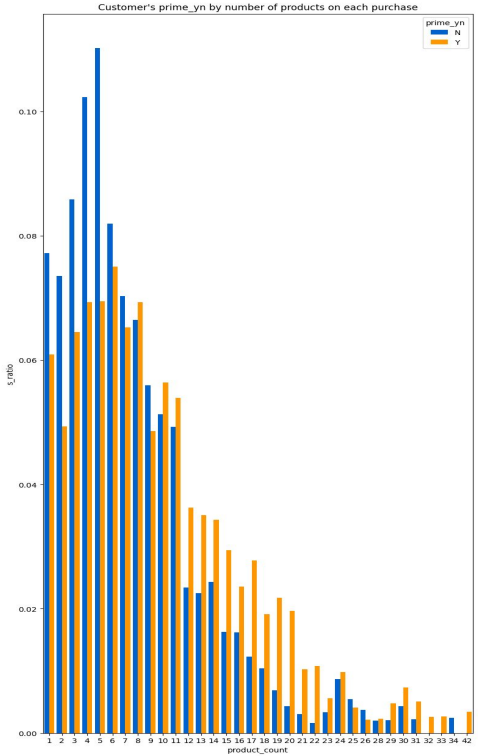
========== customer_general apriori ==========

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
3	(1, 5, 6, 7)	(2)	0.014804	0.386938	0.012482	0.843137	2.179001	0.006754	3.908273
1	(5, 6, 7)	(2)	0.021480	0.386938	0.017997	0.837838	2.165305	0.009686	3.780552
6	(4, 5, 6, 7)	(2)	0.012192	0.386938	0.010160	0.833333	2.153663	0.005442	3.678374
7	(1, 2, 4, 5, 7)	(3)	0.015385	0.431930	0.012772	0.830189	1.922043	0.006127	3.345299
0	(4, 6, 7)	(2)	0.018287	0.386938	0.015094	0.825397	2.133152	0.008018	3.511176
2	(1, 4, 6, 7)	(2)	0.012482	0.386938	0.010160	0.813953	2.103578	0.005330	3.295210
4	(1, 4, 5, 7)	(3)	0.020319	0.431930	0.016255	0.800000	1.852151	0.007479	2.840348
5	(3, 5, 6, 7)	(2)	0.013062	0.386938	0.010450	0.800000	2.067517	0.005396	3.065312

2. 데이터 전처리

4) scd -> scd_count





scd_count 변수

하나의 주문번호에 따른 구매 품목의 개수

주문할 때 한번에 몇 개의 상품을 구매하는지 확인

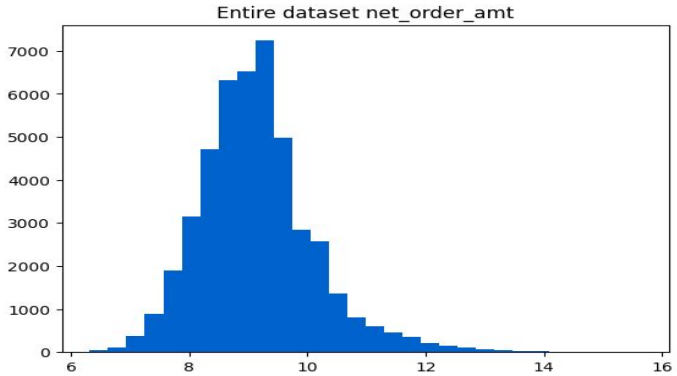
시각화 결과

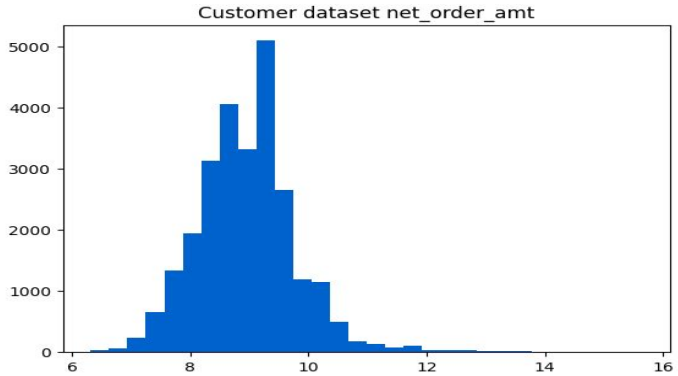
➔ product의 수가 적을 때 : 비프라임 회원의 비율 높음

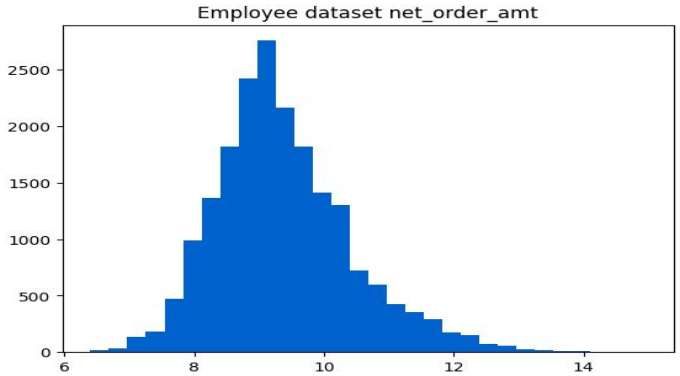
➔ product의 수가 많을 때 : 프라임 회원의 비율 높음

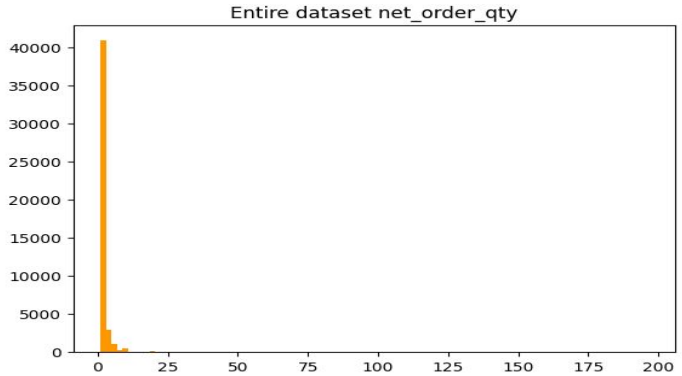
2. 데이터 전처리

5) net_order_amt

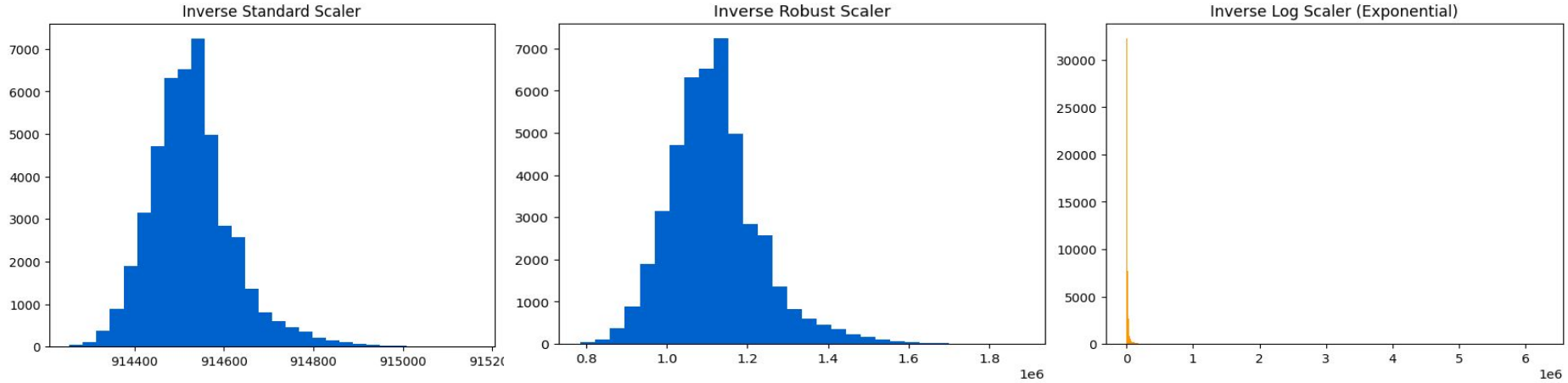








< net_order_amt 변수에 사용된 스케일러 유추 >



➔ Standard와 Robust Scaling의 경우 변환 전 데이터의 통계량을 정확히 알 수 없음

➔ 임의의 가중치를 각 통계량에 랜덤으로 부여하고 반복적으로 수행하면서 역변환

➔ 그 결과, 분포의 형태 자체는 변하지 않음

➔ 반면, 로그변환의 역변환인 exponential을 적용한 그래프는 net_order_qty와 유사하게 왜도가 매우 큰 분포로 나타남

➔ **따라서 net_order_amt 변수에 로그변환이 적용됐을 가능성 높음**

< 로그 변환 추가 증명 : 실제 가격 데이터 사용 >

	product_name	net_order_qty	price_per_unit	true_price	dif_price	discount_rate
66	The더건강한그릴후랑크300g*2	1	9981.00000	9980.0	-1.00000	-0.0001
945	밀당의고수 바삭한 김말이 400g	1	5481.00000	5480.0	-1.00000	-0.0002
1073	백설 카놀라유 900ml	1	5701.00000	5700.0	-1.00000	-0.0002
1080	백설 토마토라구 파스타소스 375g	1	6781.00000	6780.0	-1.00000	-0.0001
1869	행복한콩 몽글몽글순두부350gx2개	1	2201.00000	2200.0	-1.00000	-0.0005
...	...	...	...	...	...	...
234	[2023설임직원캠페인]특별한선택 O호	11	30937.09091	49900.0	18962.90909	0.3800
198	[2023설선물세트]특별한선택 1호	4	31140.25000	51900.0	20759.75000	0.4000
154	[2023설선물세트] 특별한선택 THE호	5	31740.20000	52900.0	21159.80000	0.4000
142	[2023설선물세트] CJ명가 초사리 곱창돌김3호	2	32640.50000	56900.0	24259.50000	0.4264
235	[2023설임직원캠페인대량구매] 특별한선택 Y호	196	31864.00510	58000.0	26135.99490	0.4506

dif_price : 실제 가격과 로그 역변환 후 얻은 상품의 한 단위 당 가격의 차이

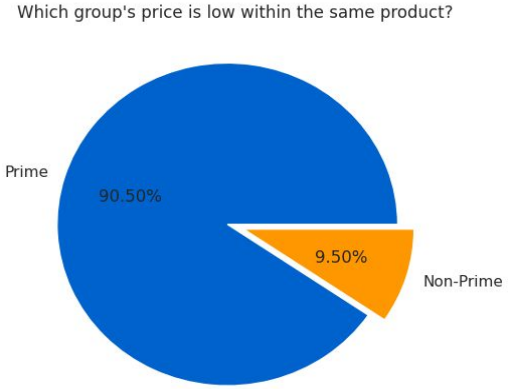
➔ 로그 변환한 값이 실제 가격과 거의 일치하는 데이터가 상당 부분 존재

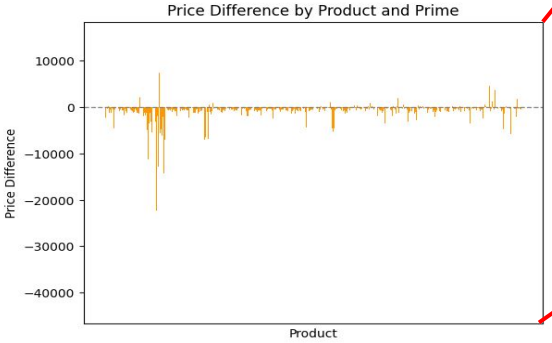
➔ 할인율(discount_rate)의 경우 최대가 약 45%인 것으로 보아 1월 행사가 적용된 것으로 보임

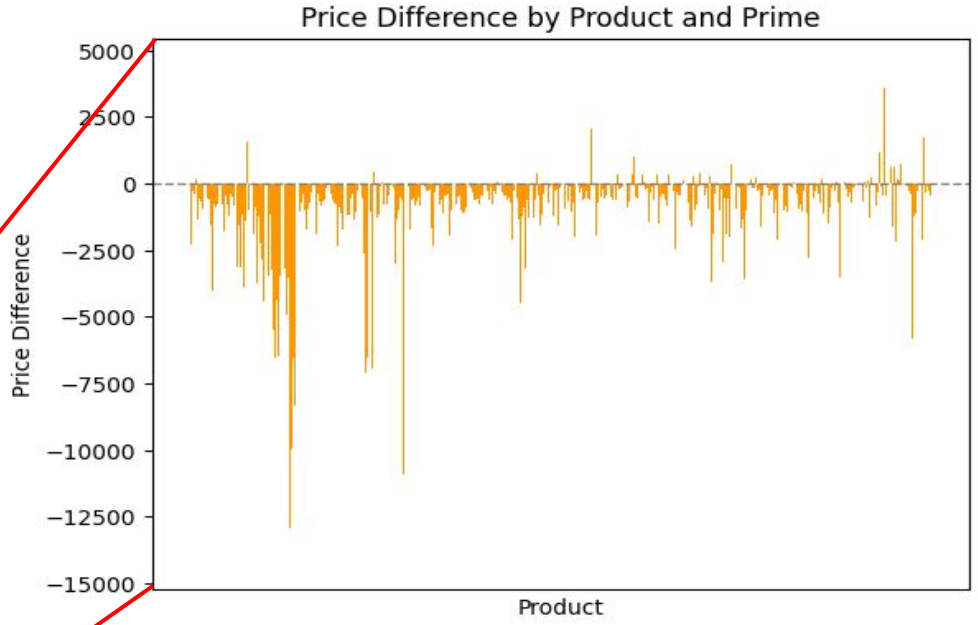
요약 : 로그 변환이 적용되었을 것이라는 가정에 대한 근거가 충분하므로 net_order_amt를 exponential 변환 후 net_order_qty로 나누어 price_per_unit을 얻고, 이를 상품 한 단위 당 가격으로 간주

2. 데이터 전처리

5) net_order_amt -> price_per_unit





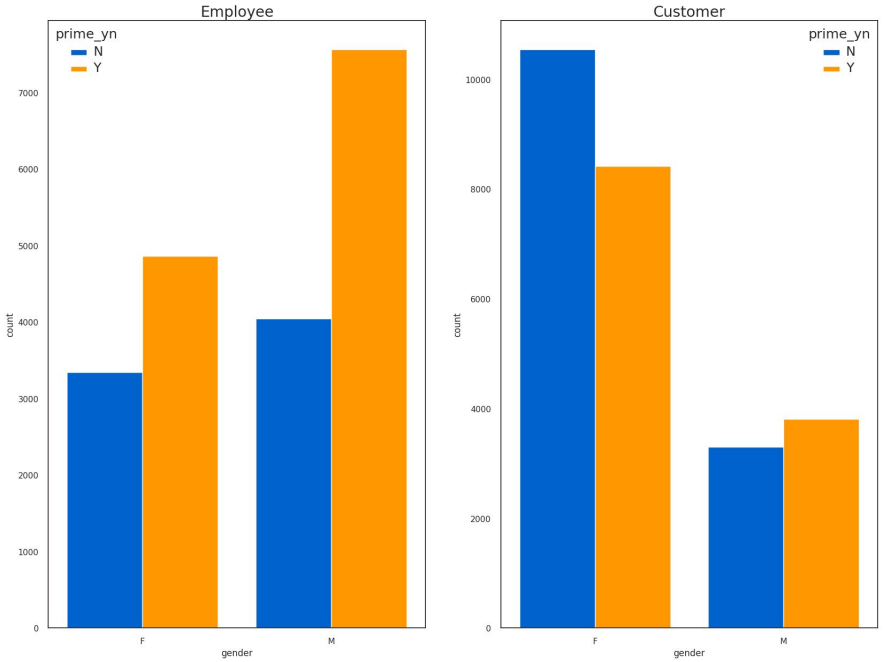


같은 상품 내 프라임 회원의 구매 금액 평균이

비프라임 회원에 비해 전체적으로 더 낮으므로 모델에 사용

2. 데이터 전처리

6) gender & age_group



gender

3	2	1949	0.156912
	1	1673	0.134691
	6	1234	0.099348
	7	948	0.076322
	3	612	0.049271
	4	598	0.048144
	8	565	0.045487
	13	392	0.031559
	10	299	0.024072
	5	248	0.019966
	9	169	0.013606
	12	97	0.007809
	11	51	0.004106
	14	30	0.002415

< 임직원-프라임 - Age Group 3 >

3	1	1118	0.151408
	2	1087	0.147210
	6	608	0.082340
	7	543	0.073537
	3	487	0.065953
	4	322	0.043608
	8	298	0.040358
	13	249	0.033722
	10	186	0.025190
	5	184	0.024919
	9	95	0.012866
	12	59	0.007990
	14	26	0.003521
	11	25	0.003386

< 임직원-비프라임-Age Group 3 >

5	2	865	0.069640
	1	659	0.053055
	6	546	0.043958
	7	404	0.032526
	3	337	0.027131
	8	222	0.017873
	13	169	0.013606
	4	158	0.012720
	5	153	0.012318
	10	67	0.005394
	9	50	0.004025
	12	43	0.003462
	11	35	0.002818
	14	4	0.000322

< 임직원-프라임-Age Group 5 >

5	1	277	0.037514
	2	266	0.036024
	7	187	0.025325
	3	177	0.023971
	6	175	0.023700
	4	173	0.023429
	8	93	0.012595
	13	58	0.007855
	5	42	0.005688
	10	29	0.003927
	9	15	0.002031
	11	6	0.000813
	12	4	0.000542
	14	2	0.000271

< 임직원-비프라임-Age Group 5 >

3	1	1289	0.105371
	2	967	0.079048
	3	696	0.056895
	6	294	0.024033
	5	286	0.023379
	7	278	0.022725
	4	274	0.022398
	13	120	0.009810
	8	72	0.005886
	11	43	0.003515
	12	25	0.002044
	10	7	0.000572
	14	4	0.000327

< 고객-프라임-Age Group 3 >

3	1	1592	0.115054
	2	1111	0.080292
	3	1103	0.079714
	5	517	0.037364
	13	408	0.029486
	4	339	0.024500
	6	293	0.021175
	7	281	0.020308
	9	197	0.014237
	12	41	0.002963
	11	31	0.002240
	8	30	0.002168
	10	8	0.000578
	14	7	0.000506

< 고객-비프라임-Age Group 3 >

5	2	1121	0.091637
	1	1044	0.085343
	3	515	0.042099
	7	395	0.032290
	6	379	0.030982
	4	319	0.026077
	5	264	0.021581
	13	185	0.015123
	8	46	0.003760
	12	38	0.003106
	11	26	0.002125
	10	9	0.000736
	9	3	0.000245
	14	1	0.000082

< 고객-프라임-Age Group 5 >

5	2	905	0.065404
	1	736	0.053191
	3	512	0.037002
	6	386	0.027896
	7	286	0.020669
	5	210	0.015177
	4	198	0.014309
	13	160	0.011563
	8	38	0.002746
	12	29	0.002096
	11	11	0.000795
	10	10	0.000723
	9	5	0.000361
	14	1	0.000072

< 고객-비프라임-Age Group 5 >

3. 모델링

1) 모델 선택

각 모델의 f1-score

[변수 선택]
- 제거할 변수 : scd, product_name, order_date, holiday, set_yn,
- 데이터 분리 기준 : employee_yn(model), prime_yn(target)
- **모델 변수(feature) : net_order_qty, net_order_amt, gender, age_grp, category, price_per_unit, scd_count**

Employee	
Decision Tree	0.7133
Random Forest	0.7398
SVM	0.7703
SGD	0.4630
Logistic Regression	0.7709
Naive Bayes	0.7037
XGBoost	**0.8465**
CatBoost	0.8415

Customer	
Decision Tree	0.7009
Random Forest	0.6447
SVM	0.3890
SGD	0.4287
Logistic Regression	0.4683
Naive Bayes	0.5894
XGBoost	**0.7860**
CatBoost	0.7416

3. 모델링

2) 하이퍼 파라미터 튜닝

1. learning_rate와 estimator 고정

① learning_rate = 0.1
➢ 학습률

② n_estimators = 1000
➢ 반복 수행 횟수

2. Tree-specific parameter 수정

① max_depth = 5
➢ 트리의 최대 깊이를 설정

② min_child_weight = 1
➢ leaf node 에 포함되는 최소 관측치의 수

③ gamma = 0
➢ leaf node의 추가분할을 결정할 최소손실 감소값

④ subsample = 0.8
➢ 학습시 데이터 샘플링 비율

⑤ colsample_bytruee = 0.8
➢ 트리생성에 필요한 feature의 샘플링 비율

3. Regularization parameter 수정

① reg_alpha = 0 (default)
➢ L1 정규화 규제 파라미터

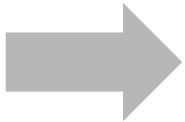
4. learning_rate 낮추고 반복

3. 모델링

2) 하이퍼 파라미터 튜닝

[employee]

```
xgb1 = XGBClassifier(
    learning_rate =0.1,
    n_estimators=1000,
    max_depth=5,
    min_child_weight=1,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    objective= 'binary:logistic',
    nthread=-1,
    scale_pos_weight=1,
    seed=seed
)
modelfit_e(xgb1, train_emp, predictors)
```



```
xgb_e = XGBClassifier(
    learning_rate =0.015,
    n_estimators=1000,
    max_depth=6,
    min_child_weight=1,
    gamma=0.4,
    reg_alpha=1e-05,
    subsample=0.96,
    colsample_bytree=0.9,
    objective= 'binary:logistic',
    nthread=-1,
    scale_pos_weight=1,
    seed=seed
)
modelfit_e(xgb_e, train_emp, predictors)
```

3. 모델링

2) 하이퍼 파라미터 튜닝

[customer]

```
xgb1 = XGBClassifier(
    learning_rate =0.1,
    n_estimators=1000,
    max_depth=5,
    min_child_weight=1,
    gamma=0,
    subsample=0.8,
    colsample_bytree=0.8,
    objective= 'binary:logistic',
    nthread=-1,
    scale_pos_weight=1,
    seed=seed
)
modelfit_c(xgb1, train_cus, predictors)
```

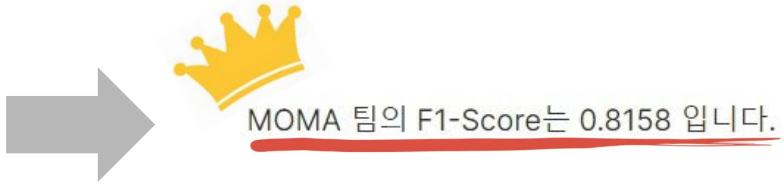
```
xgb_c = XGBClassifier(
    learning_rate =0.01,
    n_estimators=1000,
    max_depth=9,
    min_child_weight=1,
    gamma=0.2,
    reg_alpha=0.01,
    subsample=0.93,
    colsample_bytree=0.8,
    objective= 'binary:logistic',
    nthread=-1,
    scale_pos_weight=1,
    seed=seed
)
modelfit_c(xgb_c, train_cus, predictors)
```

4. 결과

1) f1 score 기반 모델 결과 설명

[employee] Model Report
Training F1_score : 0.8912
accuracy : 0.8506437768240344
recall : 0.9752032847596812
precision: 0.8204971889182415

[customer] Model Report
Training F1_score : 0.9027
accuracy : 0.9119294207901802
recall : 0.8708411673342598
precision: 0.9370217257454482



4. 결과

2) 예상 기대효과

❏ **category**

- 카테고리별 구매 특성 파악(ex. 장바구니 분석 등)을 통해 프라임 회원 예측 가능
- 해당 카테고리 상품들을 묶어서 판매하는 등의 마케팅적 인사이트도 얻을 수 있을 것으로 기대

❏ **price_per_unit & scd_count**

- 프라임 회원 여부 예측에 큰 도움이 될 것으로 기대

❏ **gender & age_grp**

- 상대적으로 프라임 회원의 비율이 적은 그룹에 특화된 마케팅 전략을 수립한다면 프라임 회원 수를 늘리는데 도움이 될 것으로 보임

❏ **set_yn, holiday**

- 프라임 회원 여부 예측에 크게 작용하지 않을 것으로 보임
- 프라임 회원에 대한 마케팅 전략 수립 시 이를 참고하여 비용 및 시간을 절감할 수 있을 것이라 예상

전략적 마케팅을 통해 기존 프라임 고객도 유지하면서 새로운 프라임 회원 또한 확보할 수 있을 것이다. 이는 위에서 언급했던 반복구매 고객의 수의 증가로 이어질 것이고, 궁극적으로 매출 증가로도 이어질 수 있을 것으로 기대된다.

Q & A

THANK YOU