KUBIG
DATA SCIENCE & AI

소비자 데이터 기반 마케팅 개선 방안

Team | ML1팀
19기 안태림, 임지우

KUBIG
DATA SCIENCE & AI

CONTENTS

01

프로젝트 소개

프로젝트 주제 및 목표
데이터 소개

02

EDA & 전처리

결측치 & 상관관계
파생변수 생성
Outliers & Transformation

03

예측 모델 선정

H20
Pycaret
Optuna

04

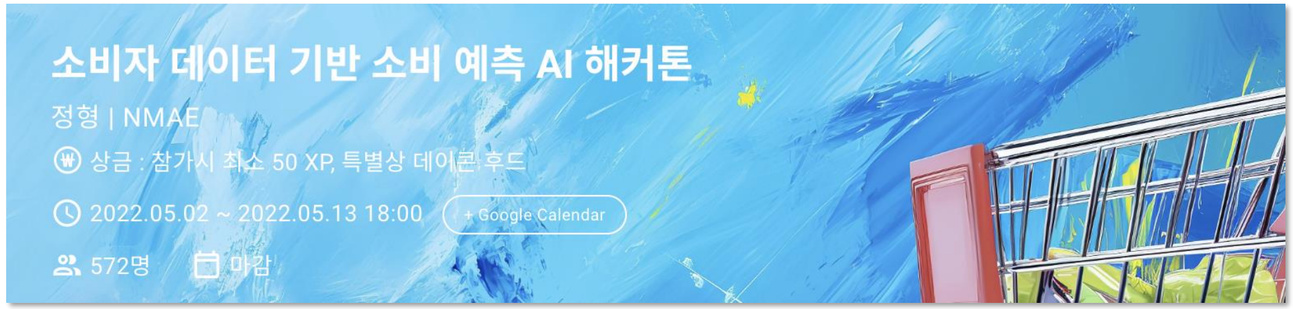
고객 세그먼트 형성

RFM

KUBIG
DATA SCIENCE & AI

01. 프로젝트 소개

01-1. 프로젝트 주제 및 목표



문제 상황

최근 마케팅 캠페인이 기대만큼 효과적이지 않음 => 문제 해결을 위한 데이터 기반 솔루션 제안 및 CRM 강화

분석 목적

1. 소비자 데이터 기반 소비량 예측

소비 예측의 효과

매출 증대: 소비 경향을 이해함으로써 고객의 구매 가능성이 높은 제품에 대한 마케팅 강화
재고 최적화: 예측된 소비 패턴을 기반으로 재고를 조절하여 과잉 재고 비용을 줄이고 재고 부족으로 인한 손실을 방지
마케팅 ROI 개선: 예측 결과를 바탕으로 더 효과적인 마케팅 캠페인을 설계하여 마케팅 투자 대비 수익률을 개선
가격 전략 최적화: 소비 예측을 통해 특정 시기에 대한 수요 증가를 예측하고 가격 전략 조정

2. 고객 세그먼트 형성 및 분석으로 마케팅 개선 방안 제안

고객 타겟팅의 효과

개인화 마케팅 강화: 고객의 행동, 선호도, 구매 이력을 분석하여 개인화된 마케팅 메시지를 제공, 고객 경험을 개선하고 구매 전환율 제고
고객 충성도 향상: 고객의 필요와 관심사에 맞춘 타겟팅은 고객 만족도를 높이고 장기적인 고객 관계를 구축
효율적인 자원 배분: 가장 가치가 높은 고객 그룹에 초점을 맞추어 마케팅 자원을 효율적으로 배분
고객 이탈 방지: 고객 데이터 분석을 통해 이탈 가능성이 높은 고객을 사전에 식별하고 이를 방지하기 위한 전략 수립

01-2. 데이터 소개

Segment	Explanation
id	샘플 아이디
target	고객의 제품 총 소비량
Year_Birth	고객 생년월일
Education	고객 학력
Marital_status	고객 결혼 상태
Income	고객 연간 가구 소득
Kidhome / Teenhome	고객 가구의 자녀 / 청소년 수
Dt_Customer	고객이 회사에 등록한 날짜
Recency	고객의 마지막 구매 이후 일수
NumDealsPurchases	할인된 구매 횟수
NumWebVisitsMonth	지난 달 회사 웹사이트 방문 횟수
AcceptedCmp1~5	고객이 n번째 캠페인에서 제안을 수락한 경우 1, 그렇지 않은 경우 0
Complain	고객이 지난 2년 동안 불만을 제기한 경우 1,그렇지 않은 경우 0
Response	고객이 마지막 캠페인에서 제안을 수락한 경우 1, 그렇지 않은 경우 0
총 22개의 변수	

KUBIG
DATA SCIENCE & AI

02. EDA&데이터 전처리

02-1. EDA

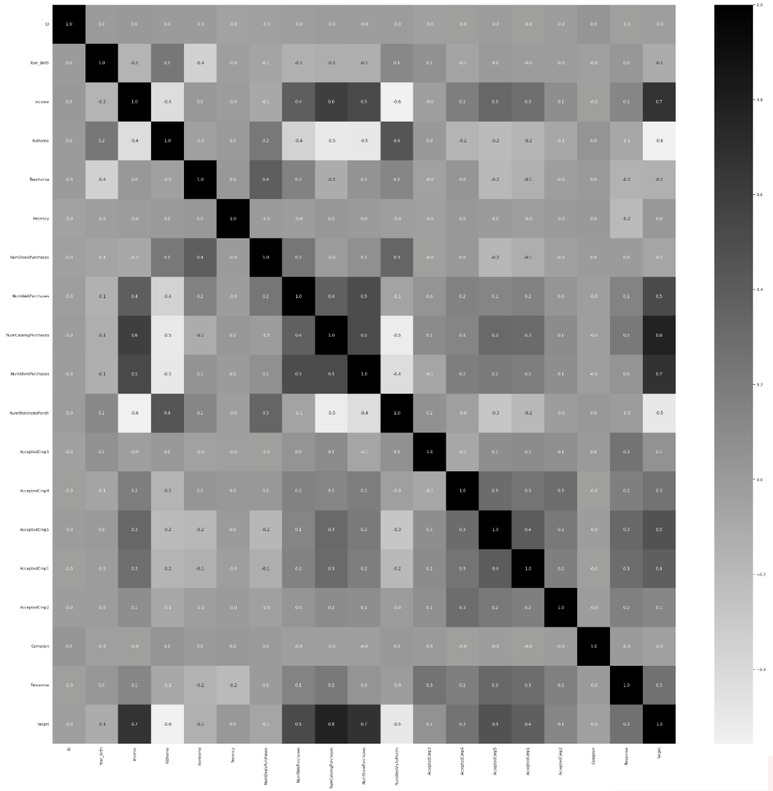
결측치 처리

Income에 존재하는 결측치 확인 후 제거

	ID	Year_Birth	Education	Marital_Status	Income	Ki
10	1994	1983	Graduation	Married	NaN	
27	5255	1986	Graduation	Single	NaN	
43	7281	1959	PhD	Single	NaN	
48	7244	1951	Graduation	Single	NaN	
58	8557	1982	Graduation	Single	NaN	

```
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   2216 non-null   int64
 1   Year_Birth           2216 non-null   int64
 2   Education            2216 non-null   object
 3   Marital_Status       2216 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2216 non-null   int64
 6   Teenhome             2216 non-null   int64
 7   Dt_Customer          2216 non-null   object
 8   Recency              2216 non-null   int64
 9   NumDealsPurchases    2216 non-null   int64
 10  NumWebPurchases      2216 non-null   int64
 11  NumCatalogPurchases  2216 non-null   int64
 12  NumStorePurchases    2216 non-null   int64
 13  NumWebVisitsMonth    2216 non-null   int64
 14  AcceptedCmp3         2216 non-null   int64
 15  AcceptedCmp4         2216 non-null   int64
 16  AcceptedCmp5         2216 non-null   int64
 17  AcceptedCmp1         2216 non-null   int64
 18  AcceptedCmp2         2216 non-null   int64
 19  Complain             2216 non-null   int64
 20  Response             2216 non-null   int64
 21  target               2216 non-null   int64
dtypes: float64(1), int64(18), object(3)
```

상관관계



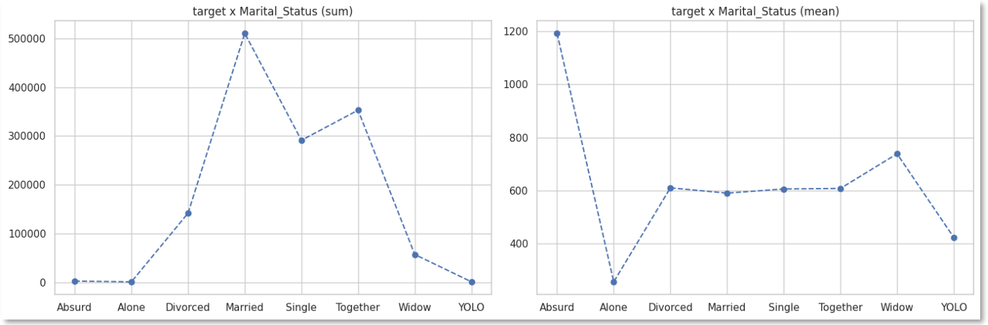
target과 상관도가 높은 변수
- income
- kidhome
- numwebpurchases
- numcatalogpurchases
- numstorepurchases
- numwebvisitsmonth
- acceptedcmp 1, 5

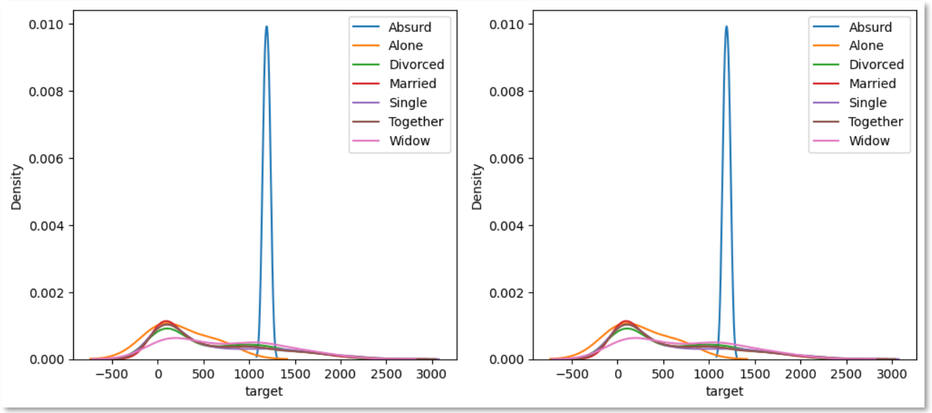
target과 상관도가 낮은 변수
- year_birth
- teen home
- recency
- num deals purchases
- acceptedcmp 2,3,4
- complain
- Response

02-1. EDA

범주형 변수

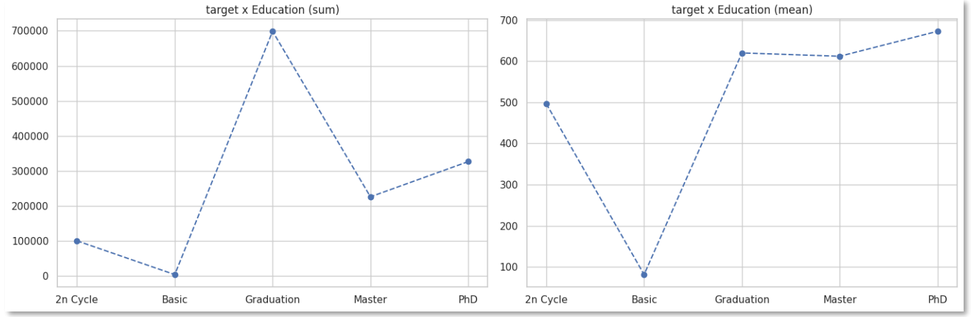
< Marital_Status >

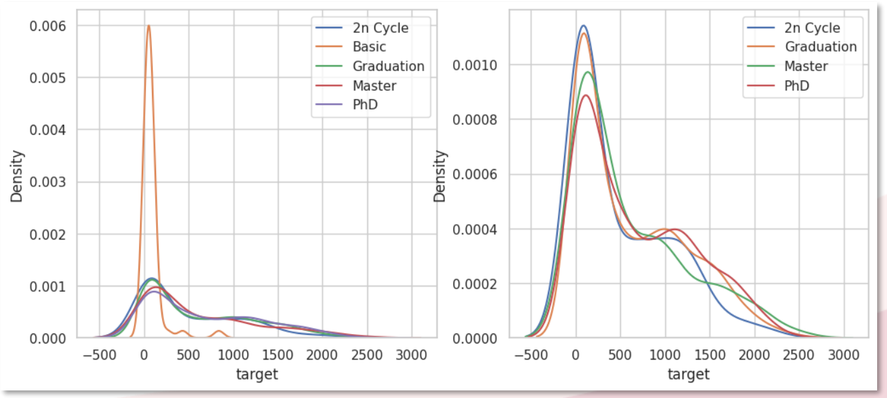




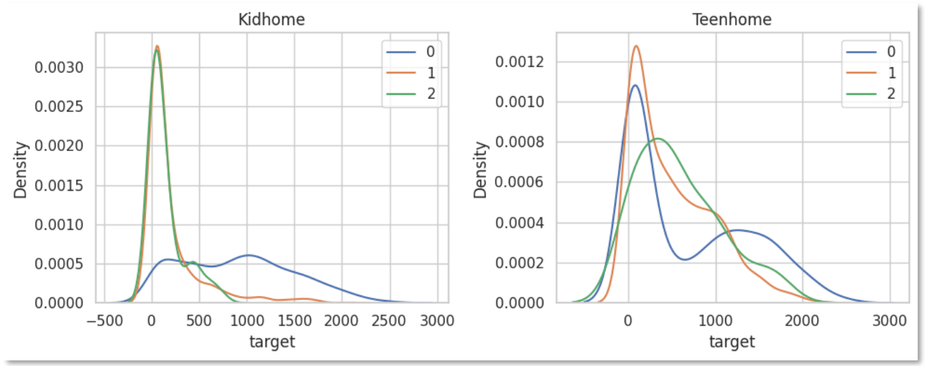
< Education >

- Basic : 고졸, 대학교 재학생
- Graduation : 학사 학위 취득자
- 2nd Cycle : 대학원 재학생
- Master : 석사 학위 취득자
- PhD : 박사 학위 취득자

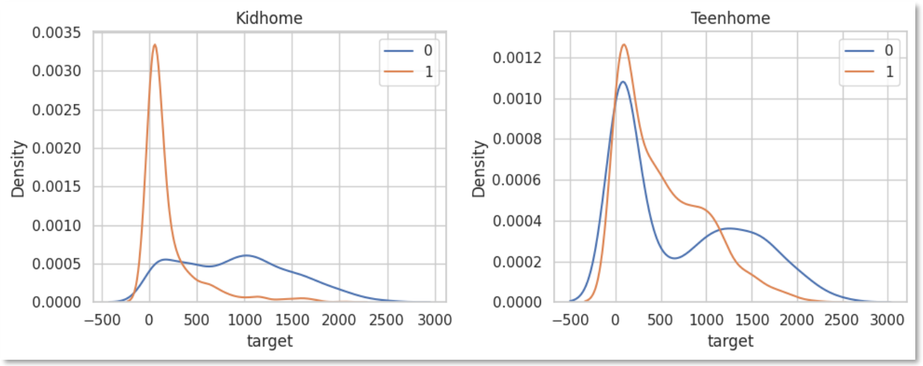




02-2. 파생변수 생성

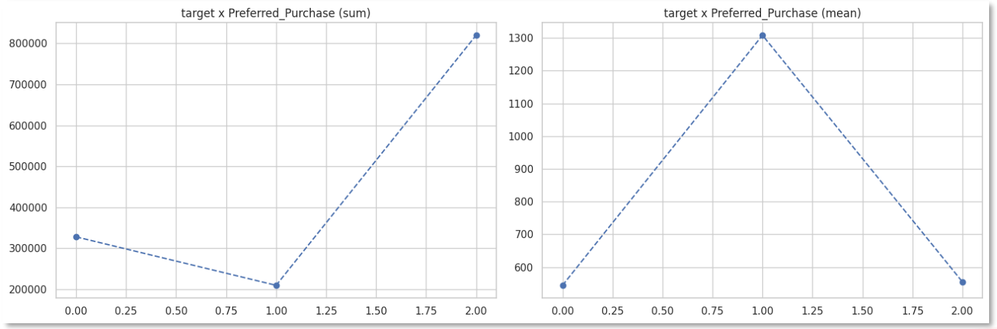


< 자녀 수와 청소년 수에 따른 target 변화 >



< 자녀와 청소년 유무에 따른target 변화 >

binary 변수로 변형: 자녀 수=2를 자녀 수=1에 포함시킨 후, 자녀 없음(0)과 자녀 있음(1)의 binary 변수로 변형
Teenhome 역시 1과 2를 합쳐서, 청소년 없음(0)과 청소년 있음(1)을 나타내는 binary 타입의 변수로 변형



구매 경로 Web(0), Catalog(1), Store(2)에 따른 Target 변화

구매 빈도수가 가장 높은 구매 경로를
Preferred_Purchase 변수로 생성

02-2. 파생변수 생성

```
df['Age'] = 2022-df['Year_Birth']
df['Age_Range'] = (df['Age']//10).replace({2:3,8:7})


df['Dt_Customer'] = df['Dt_Customer'].apply(lambda x: list(map(int,x.split('-'))))
df['Dt_Customer'] = df['Dt_Customer'].apply(lambda x: date(x[2],x[1],x[0]))
df['Days_Customer'] = df['Dt_Customer'].apply(lambda x: (date(2022,1,1)-x).days)


purchase_list = [f'Num{cat}Purchases' for cat in['Web','Catalog','Store']]
df['TotalPurchases'] = [sum(row) for _,row in df[purchase_list].iterrows()]
purchase_dict = {cat: i for i, cat in enumerate(purchase_list)}
df['Preferred_Purchase'] = [purchase_dict[row.index[row.argmax()]] for _,row in df[purchase_list].iterrows()]
```

- 'Age' : 2022년 기준 고객의 나이
- 'Age_Range' : 고객 나이 구간
- 'Days_Customer' : 2022년 1월 1일 기준 고객이 회사에 가입한 일수
- 'TotalPurchases' : 고객의 총 구매 횟수
- 'Preferred_Purchase' : 카탈로그, 매장, 회사 웹사이트 중 가장 큰 값에 해당하는 열

02-2. 파생변수 생성

```
df['RCatalogPurchases'] = np.where(df['TotalPurchases'] == 0, 0, df['NumCatalogPurchases'] / df['TotalPurchases'])
df['RStorePurchases'] = np.where(df['TotalPurchases'] == 0, 0, df['NumStorePurchases'] / df['TotalPurchases'])
df['RWebPurchases'] = np.where(df['TotalPurchases'] == 0, 0, df['NumWebPurchases'] / df['TotalPurchases'])



campaigns = [f'AcceptedCmp{i}' for i in range(1,6)]+['Response']
df['NumAcceptedCmp'] = sum([df[campaign] for campaign in campaigns])


df['Kidhome_has'] = df['Kidhome'].apply(lambda x: 0 if x == 0 else 1)
df['Teenhome_has'] = df['Teenhome'].apply(lambda x: 0 if x == 0 else 1)


test_diff_date = df["Dt_Customer"] - df["Dt_Customer"].min()
df["Pass_Customer"] = [i.days for i in test_diff_date]
```

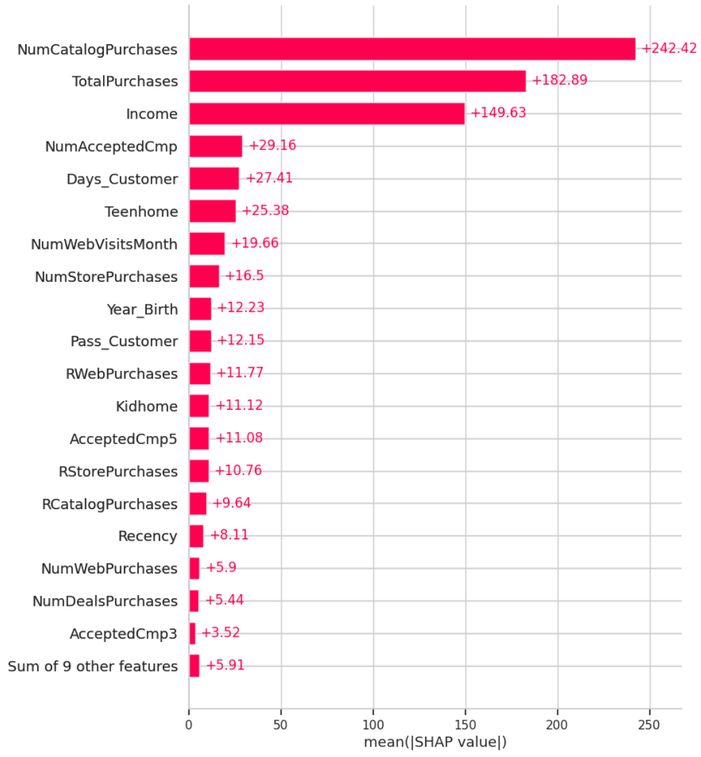
- 'RCatalogPurchases' : 구매 경로별 비율 - 카탈로그
- 'RStorePurchases' : 구매 경로별 비율 - 매장
- 'RWebPurchases' : 구매 경로별 비율 - 회사 웹사이트
- 'NumAcceptedCmp' : 고객이 캠페인에서 제안을 수락한 횟수
- 'Kidhome_has' : 고객 가구에 자녀가 있으면 1, 그렇지 않은 경우 0
- 'Teenhome_has' : 고객 가구에 청소년이 있으면 1, 그렇지 않은 경우 0
- 'Pass_Customer' : 가장 과거 시점의 회사 등록일부터 지난 일수

02-2. 파생변수 생성

H20 _ gradient boosting machine feature importance

	variable	relative_importance	scaled_importance	percentage
0	NumCatalogPurchases	1.153105e+09	1.000000	0.353575
1	TotalPurchases	7.607489e+08	0.659740	0.233268
2	Income	4.102312e+08	0.355762	0.125789
3	RCatalogPurchases	2.968261e+08	0.257415	0.091015
4	NumStorePurchases	1.139313e+08	0.098804	0.034935
5	NumWebVisitsMonth	9.030218e+07	0.078312	0.027689
6	NumAcceptedCmp	5.379047e+07	0.046648	0.016494
7	Days_Customer	4.314488e+07	0.037416	0.013229
8	Pass_Customer	3.960998e+07	0.034351	0.012146
9	Teenhome_has	3.508646e+07	0.030428	0.010759
10	AcceptedCmp5	3.135757e+07	0.027194	0.009615
11	NumWebPurchases	3.032011e+07	0.026294	0.009297
12	RStorePurchases	2.868676e+07	0.024878	0.008796
13	RWebPurchases	2.614283e+07	0.022672	0.008016
14	Teenhome	2.068045e+07	0.017935	0.006341
15	Age_Range	2.031694e+07	0.017619	0.006230
16	Recency	1.886130e+07	0.016357	0.005783
17	Age	1.872833e+07	0.016242	0.005743
18	Marital_Status	1.609894e+07	0.013961	0.004936
19	Education	1.565286e+07	0.013575	0.004800
20	NumDealsPurchases	1.247420e+07	0.010818	0.003825
21	AcceptedCmp1	6.458806e+06	0.005601	0.001980
22	Preferred_Purchase	5.678908e+06	0.004925	0.001741
23	Kidhome_has	4.222092e+06	0.003661	0.001295

Pycaret _ xgboost Shap




```
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Education            2216 non-null   object
 1   Income               2216 non-null   float64
 2   Kidhome              2216 non-null   int64
 3   NumWebPurchases      2216 non-null   int64
 4   NumCatalogPurchases  2216 non-null   int64
 5   NumStorePurchases    2216 non-null   int64
 6   NumWebVisitsMonth    2216 non-null   int64
 7   AcceptedCmp5         2216 non-null   int64
 8   AcceptedCmp1         2216 non-null   int64
 9   target               2216 non-null   int64
 10  TotalPurchases       2216 non-null   int64
 11  Preferred_Purchase   2216 non-null   object
 12  RCatalogPurchases    2216 non-null   float64
 13  RStorePurchases      2216 non-null   float64
 14  NumAcceptedCmp       2216 non-null   int64
 15  Kidhome_has          2216 non-null   int64
```

02-3. Outliers & Transformation

Outliers

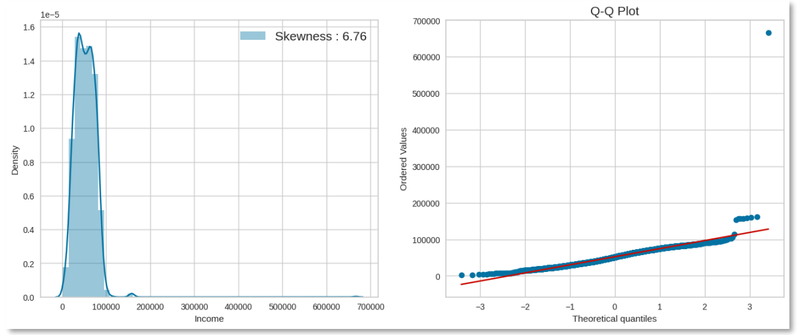
1. 둘 이상의 특성에서 Outlier로 간주되는 데이터 drop
2. 하나 이상의 특성에서 Outlier로 간주되는 데이터 drop

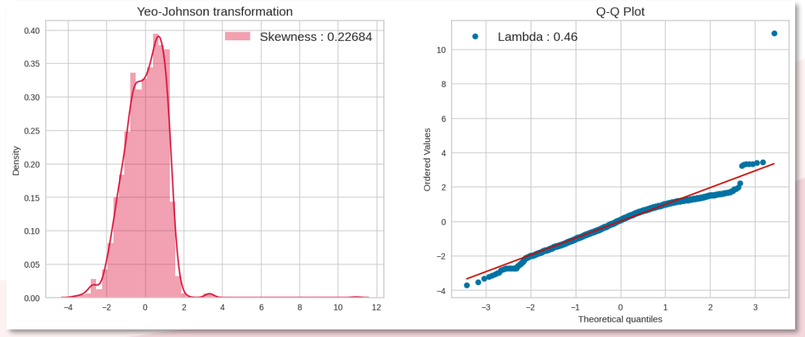
=> 1,2번 보다 Outlier 제거하기 전 모델 성능이 더 좋아서 Outlier를 처리하지 않았습니다.

Transformation

Income	6.763487
Kidhome	0.635610
NumWebPurchases	1.197037
NumCatalogPurchases	1.881075
NumStorePurchases	0.701826
NumWebVisitsMonth	0.218043
TotalPurchases	0.294141
RCatalogPurchases	0.910207
RStorePurchases	0.210789
NumAcceptedCmp	2.440020

<Skewness>





비대칭도가 심한 변수에 대해 transformation을 시도했습니다.

각 변수에 대해 Log, Yeo-Johnson transformation을 진행하고

모델 적합 후 성능 비교를 통해 transformation 적용 여부를 결정했습니다.

최종적으로, income에 Yeo-Johnson transformation을 적용했습니다.

KUBIG
DATA SCIENCE & AI

03. 예측 모델 선정

03-1. H2O

H2O Auto ML을 사용하여 최적의 모델을 탐색했습니다. 변수들의 조합을 다양하게 변경하고 정수형 변수들을 범주형 변수들로 조정하는 과정을 거쳐 결과를 산출했습니다. 최종적으로 Gradient Boosting Machine과 Extra Tree Regressor 모델을 통한 예측에서 가장 좋은 성능을 보여줬습니다.
Gradient Boosting 알고리즘을 통해서 NMAE를 0.177까지 낮추었습니다.

Parse progress: |(done) 100%

model_id	rmse	mse	mae	rmsle	mean_residual_deviance
XRT_2_AutoML_2_20240227_153028	200.864	40346.5	108.23	0.307059	40346.5
XRT_1_AutoML_1_20240227_152447	200.864	40346.5	108.23	0.307059	40346.5
GBM_grid_2_AutoML_2_20240227_153028_model_5	202.441	40982.3	107.485	0.301572	40982.3
GBM_grid_1_AutoML_1_20240227_152447_model_5	202.441	40982.3	107.485	0.301572	40982.3
GBM_10_AutoML_2_20240227_153028	202.448	40985.3	109.939	nan	40985.3
GBM_5_AutoML_1_20240227_152447	202.448	40985.3	109.939	nan	40985.3

```
print(f"NMAE Score: {nmae_score}")
```

```
NMAE Score: 0.17796580442332904
```

03-2. Pycaret

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
et	Extra Trees Regressor	108.1323	46193.7668	212.0972	0.8695	0.2725	0.2070
rf	Random Forest Regressor	110.0574	44306.9977	207.9816	0.8750	0.2789	0.2217
xgboost	Extreme Gradient Boosting	113.4111	49463.4865	219.3584	0.8601	0.2876	0.2204
lightgbm	Light Gradient Boosting Machine	114.3731	46317.2898	212.1847	0.8693	0.3009	0.2469
gbr	Gradient Boosting Regressor	121.8044	48237.0638	217.1475	0.8638	0.3372	0.2937
dt	Decision Tree Regressor	133.4780	77285.2346	273.1540	0.7818	0.3359	0.2391
knn	K Neighbors Regressor	152.6768	74038.7096	269.7136	0.7896	0.4340	0.4260
huber	Huber Regressor	159.2808	79528.0087	280.4900	0.7745	0.5713	0.5197
par	Passive Aggressive Regressor	161.2429	83898.9366	288.1965	0.7619	0.5542	0.5504
ada	AdaBoost Regressor	165.4874	60070.6007	243.8657	0.8293	0.6071	0.8025
br	Bayesian Ridge	181.3992	70831.2628	264.8401	0.7990	0.7003	1.0122
lr	Linear Regression	181.6902	70898.5482	265.0310	0.7987	0.7520	1.0420
ridge	Ridge Regression	182.0340	71136.8713	265.4356	0.7981	0.7177	0.9966

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Voting Regressor	100.1774	38274.7142	195.6392	0.9061	0.2442	0.1933

NMAE Score: 0.1597759828367654

Pycaret set up의 train_size, normalize method, compare models의 n_select 등 파라미터를 다양하게 변경하고 앙상블과 튜닝을 활용해서 성능을 높였습니다.

결과적으로 tuning을 적용하지 않은 4개 모델의 앙상블(normalize_method= 'minmax', n_select=4)이 가장 좋은 결과를 보였습니다.

03-3. Optuna

H20와 Pycaret에서 좋은 성능을 보였던 Top 5 모델을 선정하고,
Optuna를 활용해 각 모델에 대해 하이퍼파라미터 최적화를 진행하였습니다.

	NMAE
LGBMRegressor n_estimators=551,max_depth=16,learning_rate=0.010166770478443585,num_leaves=247,min_child_samples=7	0.15894771489271245
XGBRegressor n_estimators=943,max_depth=10,learning_rate=0.022683495743900457, subsammple=0.7374206169292847, colsample_bytree=0.8628427847023177	0.15589243509025516
GradientBoostingRegressor n_estimators=1301,max_depth=9,learning_rate=0.01,subsample=0.5	0.15105868084852017
ExtraTreesRegressor n_estimators=203,max_depth=20	0.16735785028566216
RandomForestRegressor n_estimators=300,max_depth=22	0.16724568767723605

<Top 5 model의 하이퍼파라미터 최적화 결과 >

성능이 가장 좋았던 조합: XGBRegressor + GradientBoostingRegressor => NMAE: 0.15117405295149766

KUBIG
DATA SCIENCE & AI

04. 고객 세분화

04. RFM 분석을 통한 고객 세분화

RFM

RFM (Recency, Frequency, Monetary value)은 고객의 구매 이력을 기반으로 한 마케팅 분석 기법

Recency (최근성)
고객이 마지막으로 구매한 시점. 최근에 구매한 고객일수록 더 높은 가치를 가진다고 간주하여, 최근성은 고객이 여전히 활성 상태인지를 나타내는 지표로 사용

Frequency (빈도)
고객이 특정 기간 동안에 구매한 횟수. 빈도가 높을수록 고객 충성도가 높다고 판단하여, 고객의 브랜드나 제품에 대한 애착을 나타내는 지표로 해석

Monetary value (금전적 가치)
고객이 특정 기간 동안에 지출한 총 금액. 금전적 가치가 높은 고객은 더 높은 수익을 생성하는 것으로, 고객의 구매력을 나타내는 지표.

Customer Segment

Segment	Explanation
Champions	Bought recently, buy often and spend the most.
Loyal Customers	Spend good money with us often. Responsive to promotions.
Potential Loyalist	Recent customers, but spent a good amount and bought more than once.
Recent Customers	Bought most recently, but not often.
Promising	Recent shoppers, but haven't spent much.
Need Attention	Above average recency, frequency and monetary values. May not have bought very recently though.
About To Sleep	Below average recency, frequency and monetary values. Will lose them if not reactivated.
At Risk	Spent big money and purchased often. But long time ago. Need to bring them back.
Cant Lose Them	Made biggest purchases, and often. But haven't returned for a long time.
Hibernating	Last purchase was long back, low spenders and bought seldomly.
Lost	Lowest recency, frequency, and monetary scores.

04. RFM 분석을 통한 고객 세분화

1. Recency, Frequency, MonetaryValue에 해당하는 변수 생성

```
[ ]  # R
     df["InvoiceDate"] = pd.to_datetime(df["Dt_Customer"])
     last_timestamp = df["InvoiceDate"].max() + dt.timedelta(days=1)
     last_timestamp
```

```
Timestamp('2014-12-07 00:00:00')
```

```
[ ]  # F
     purchase_list = [f'Num{cat}Purchases' for cat in['Web','Catalog','Store']]
     df['TotalPurchases'] = [sum(row) for _,row in df[purchase_list].iterrows()]
```

```
[ ]  # M
     df["TotalPrice"] = df["target"]
```

2. Customer segment 분류 – 20%씩 5개 구간으로 분할 및 점수 부여

```
seg_map = {
    'Champions': ['555', '554', '544', '545', '454', '455', '445'],
    'Loyal_Customers': ['543', '444', '435', '355', '354', '345', '344', '335'],
    'Potential_Loyalists': [
        '553', '551', '552', '541', '542', '533', '532', '531', '452', '451',
        '442', '441', '431', '453', '433', '432', '423', '353', '352', '351',
        '342', '341', '333', '323' ],
    'Recent_Customers': ['512', '511', '422', '421', '412', '411', '311'],
    'Promising': [ '525', '524', '523', '522', '521', '515', '514', '513', '425', '424',
                   '413', '414', '415', '315', '314', '313'   ],
    'Customers_Needing_Attention': ['535', '534', '443', '434', '343', '334', '325', '324'],
    'About_To_Sleep': ['331', '321', '312', '221', '213'],
    'At_Risk': [   '255', '254', '245', '244', '253', '252', '243', '242', '235', '234',
                   '225', '224', '153', '152', '145', '143', '142', '135', '134', '133',
                   '125', '124'   ],
    'Cant_Lose_Them': ['155', '154', '144', '214', '215', '115', '114', '113'],
    'Hibernating': [ '332', '322', '231', '241', '251', '233', '232', '223', '222', '132',
                     '123', '122', '212', '211'  ],
    'Lost': ['111', '112', '121', '131', '141', '151']
```

3. Customer segmentation 진행

```
[ ]  rfm[["segment", "Recency", "Frequency", "MonetaryValue"]].groupby("segment").agg(["mean", "count"])
```

	Recency		Frequency		MonetaryValue	
	mean	count	mean	count	mean	count
segment						
About_To_Sleep	514.548387	31	5.419355	31	61.451613	31
At_Risk	741.552972	387	17.692506	387	1045.739018	387
Cant_Lose_Them	829.023077	130	22.223077	130	1257.038462	130
Champions	282.081818	220	20.840909	220	1320.763636	220
Customers_Needing_Attention	351.715789	95	14.557895	95	910.105263	95
Hibernating	667.088319	351	7.031339	351	144.715100	351
Lost	843.106061	66	3.909091	66	48.106061	66
Loyal_Customers	454.652968	219	19.292237	219	1113.511416	219
Potential_Loyalists	399.767677	198	12.257576	198	345.989899	198
Promising	195.046729	107	6.644860	107	106.915888	107
Recent_Customers	342.864078	412	4.529126	412	49.648058	412

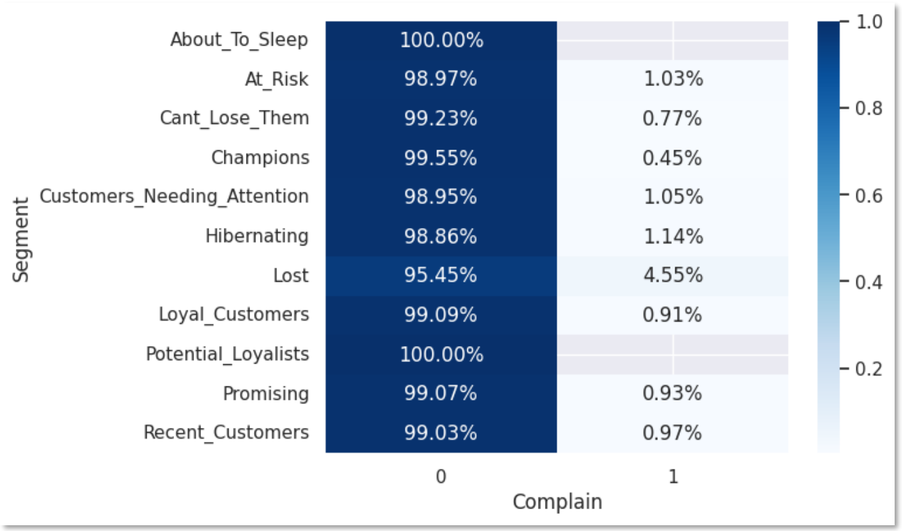
총 11단계의 고객 그룹 설정

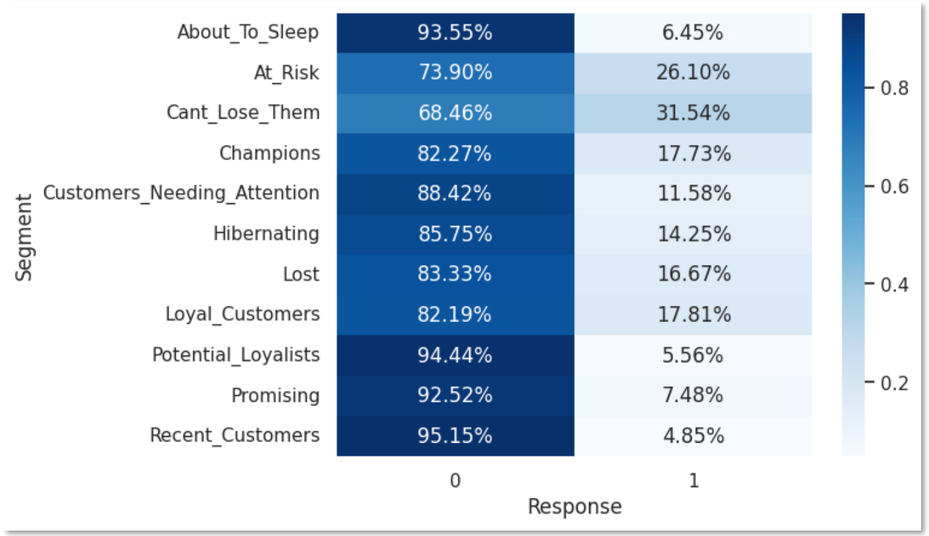
Champions, Loyal Customers, Potential Loyalist

Recent Customers, Promising
Need Attention, About To Sleep
At Risk, Can't Lose Them
Hibernating, Lost

04. RFM 분석을 통한 고객 세분화

4. Segment 별 고객 분석 – Complain, Response





Complain

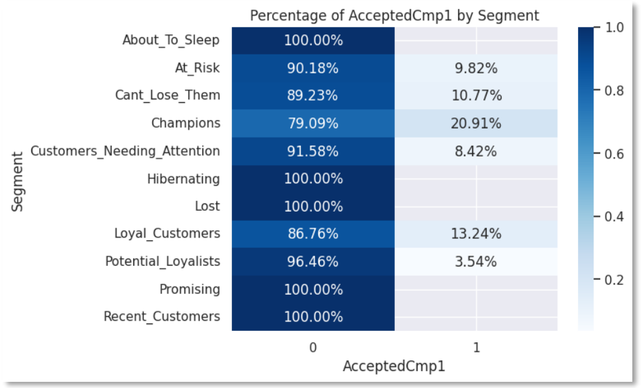
- 지난 2년 동안 불만을 제기한 비율 두드러지게 높은 그룹 : Lost

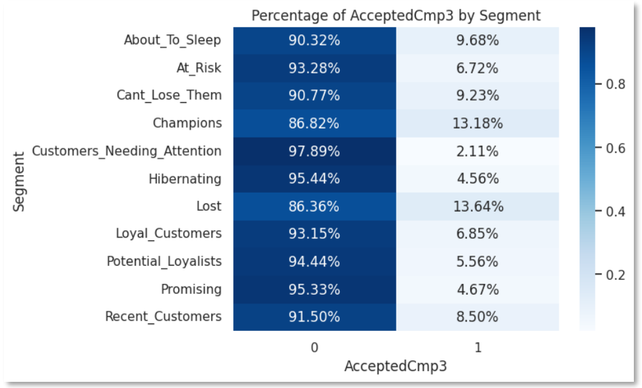
Response : 마지막 캠페인 제안 수락 여부

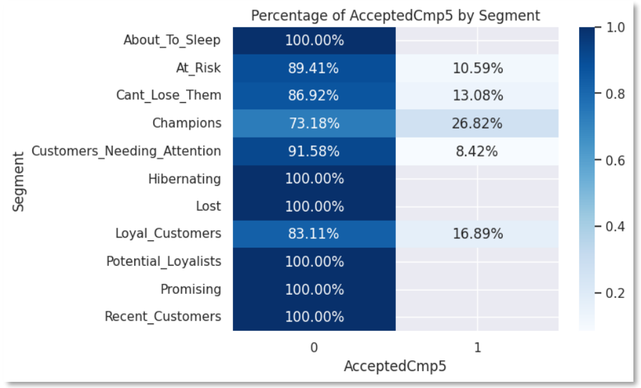
- 제안을 수락한 비율이 높은 그룹 : Can't lose them, At risk
  -> 마지막 캠페인과 유사한 마케팅 캠페인 집행
- 제안 수락한 비율이 낮은 그룹 : Recent customers, Potential Royalists, Promising
  -> 마지막 캠페인과 유사한 마케팅 캠페인 집행X

04. RFM 분석을 통한 고객 세분화

4. Segment 별 고객 분석 – AcceptedCmp





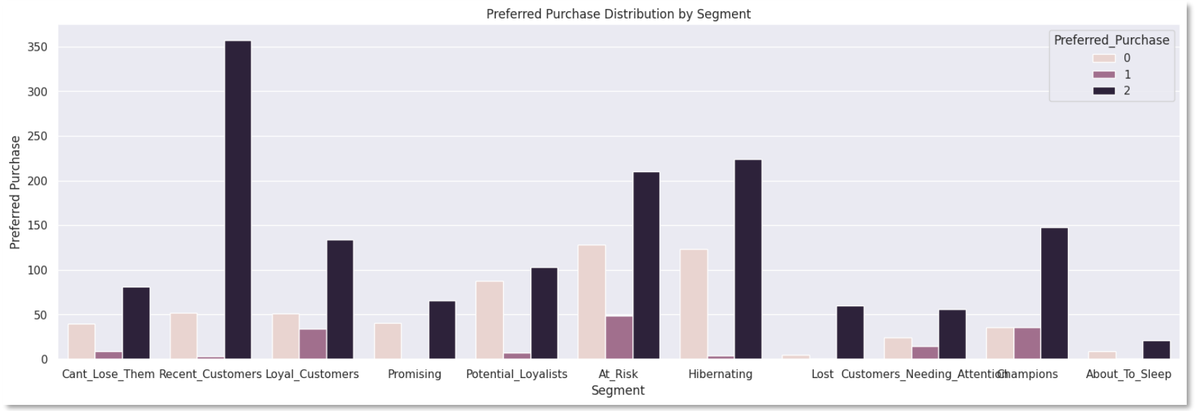


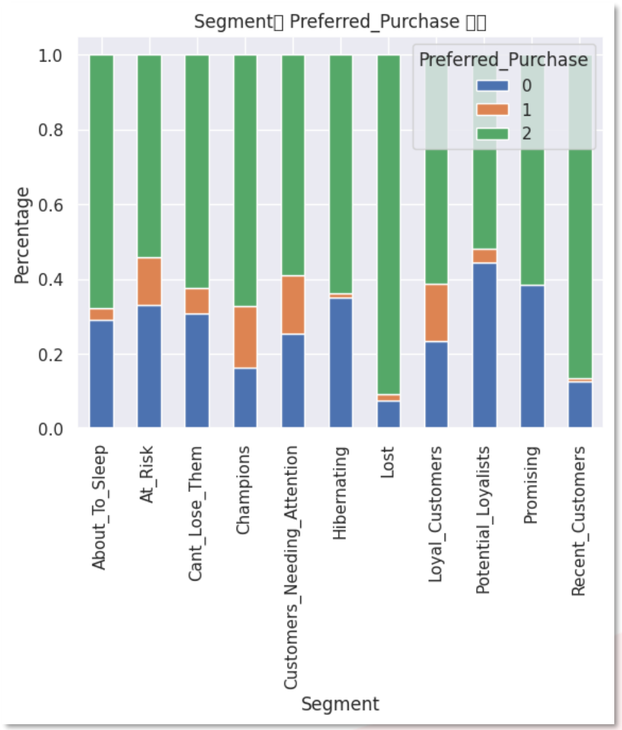
AcceptedCmp

- 대부분의 캠페인에 대해 수락한 비율 높은 그룹 :
  At_Risk, Cant_Lose_Them, Champions, Loyal_Customers
- 의외로 Hibernating, Lost의 특정 캠페인 수락율은 높았음
  나머지 캠페인은 0인데 3번째, 마지막 캠페인은 높음
  => 이들을 깨우려면 마지막 캠페인과 유사하게 공략
- Response에서 Promising의 수락 비율이 높지 않지만 다른 캠페인 대비 Promising의 수락율 높았던 편에 속함

04. RFM 분석을 통한 고객 세분화

4. Segment 별 고객 분석 – Preferred_Purchase



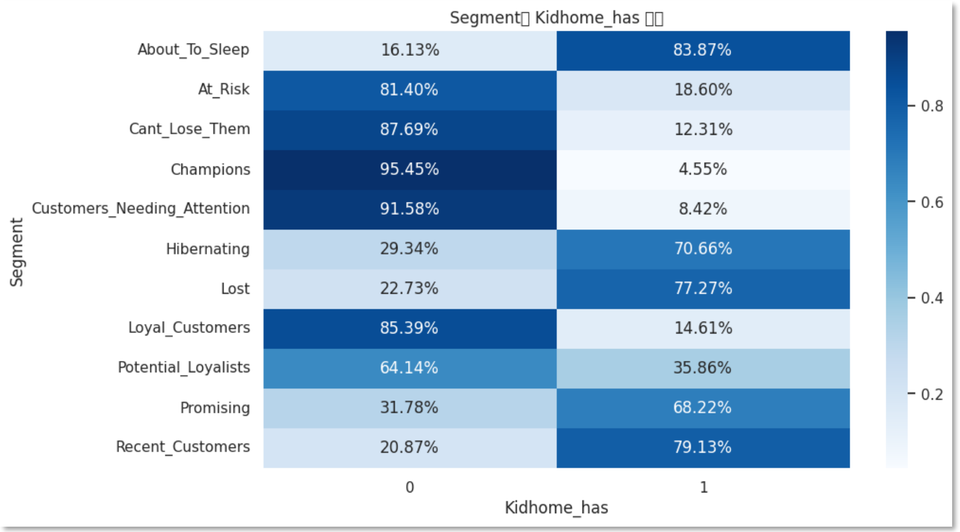


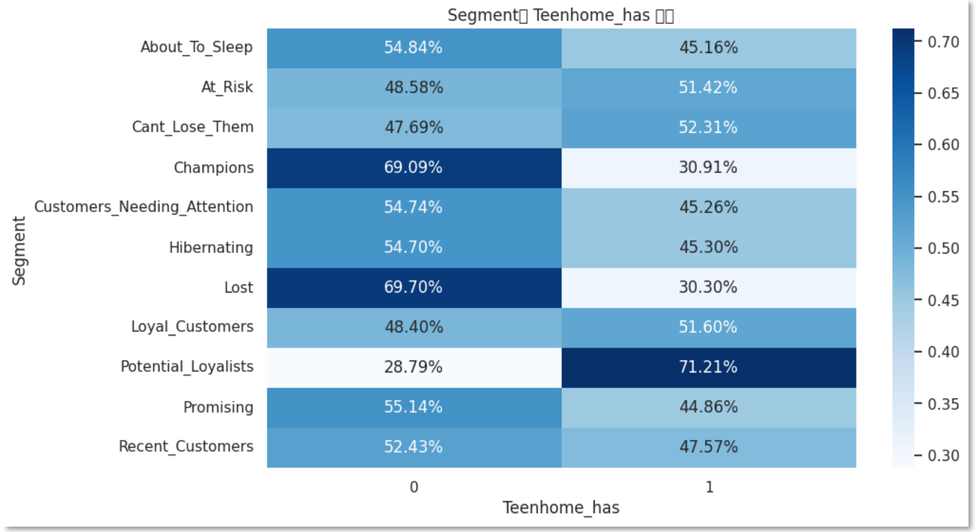
Preferred_Purchase : 0: 'Web', 1: 'Catalog', 2: 'Store'

- 전반적으로 모든 세그먼트에서 Store Purchase가 가장 많음
- Recent_Customer : 매장 직접 구매 압도적
  => 매장으로 유인하는 마케팅 방법(캠페인 등) 필요
- Loyal_Customer, At_risk: 카탈로그 사용해서 구매 많이 함
  => 카탈로그 구매 유도하는 캠페인 effective할 것
- Promising: 카탈로그를 통한 구매가 없음
- Potential_loyalists: 카탈로그를 통한 구매가 거의 존재하지 않고 웹과 매장 구매 많음

04. RFM 분석을 통한 고객 세분화

4. Segment 별 고객 분석 – Kidhome_has, Teemhome_has





Kidhome_has

- 자녀 비율이 가장 높은 그룹 :
  About_To_Sleep, Recent_Customers, Lost 순
- 80% 이상이 가정에 자녀가 없는 그룹 :
  At risk, Can't lose them, Champions,
  Customers needing attention, Loyal

Teenhome_has

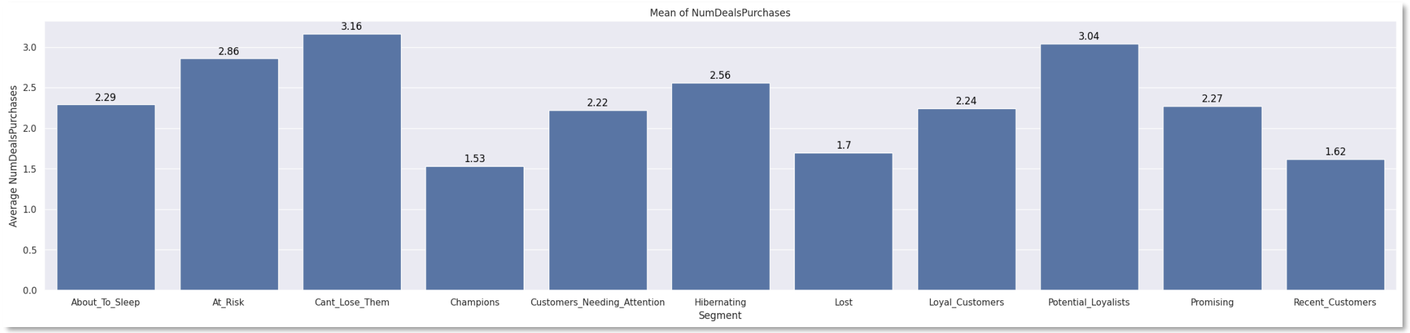
- Potential Loyalists :
  자녀 있는 비율 낮았는데 청소년 비율은 70%
  => grandchildren 많은 세그먼트로 추측

04. RFM 분석을 통한 고객 세분화

4. Segment 별 고객 분석 – NumDealsPurchases, Income

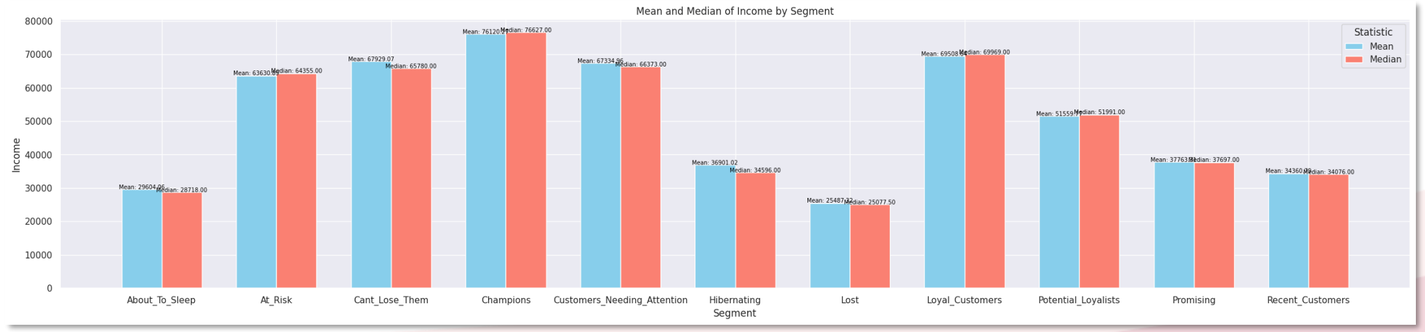
NumDealPurchases

- 할인 구매 횟수가 높은 그룹 : Cant lose them, At risk, Potential_Loyalists
- 할인 구매 횟수가 낮은 그룹 : Recent Customers, Champions, Lost



Income

- Cant lose them, Loyal, At risk, Champions 소득이 높음



04. RFM 분석을 통한 고객 세분화

5. 고객 Segment별 마케팅 캠페인 제안

	insight	마케팅 캠페인 제안
At risk	마지막 캠페인에서 제안을 수락한 비율이 높음	1. 마지막 캠페인 성공 요소 분석 후 유사 캠페인 기획 2. 페이퍼/디지털 카탈로그 배포. 할인 프로모션으로 어필 3. 청소년 있는 가정 위한 제품, 소득 높은 고객 대상으로 하는 제품 제안
	카탈로그를 통한 구매가 비교적 많이 존재함	
	80% 이상이 자녀 없지만 50% 정도는 청소년 있음 -> grandchildren	
	Together & Married 높음, 가구 소득이 높음, 할인 구매 횟수가 높음	
Recent Customers	3번째 캠페인 응답률 가장 높았으나 전반적으로 마케팅 캠페인 응답률이 낮음	1. 비교적 최근에 구매했지만 반복적인 구매 행동을 보이지는 않음 -> 이유 추측: 제품이나 서비스를 처음 시도해보거나, 특정 이유로 구매를 결정한 경우 아직 브랜드나 제품에 대한 충성도가 형성되지 않아 반복 구매로 이어지지 않는 것으로 추정됨. 따라서 구매를 반복적, 지속적으로 전환시키기 위해 고객 경험 개선 및 마케팅을 통해 브랜드에 대한 관심을 유지하여 재구매를 유도하는 것을 제안 EX. 매장 방문 유도 프로모션: 가족 단위 고객 대상으로 한 매장 내 이벤트, 매장 방문 고객 대상 프로모션 진행
	매장 직접 구매 압도적 -> 매장으로 유인하는 마케팅 방법(캠페인 등) 필요	
	카탈로그 효과 없음, 할인 구매 횟수 낮음, 자녀 비율 높음	

KUBIG
DATA SCIENCE & AI

Thank You