기계학습 텀프로젝트

1. 주제선정

최종과제를 위해 Mart 데이터를 골랐다. Mart 데이터는 일정 기간 동안 사람들이 mart 에서 제품을 구매한 것에 대한 여러가지 정보가 담겨져 있는 데이터다. 이 데이터를 활용하여, 고객의 성별을 예측하는 것을 목표로 설정했다. 데이터는 다음과 같이 Kaggle 에서 받았다

https://www.kaggle.com/cherryhillst/korean-markets

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
ID	성별	연령	거주지역	거주지역_	거주지역_	상품대분류	상품중분류	구매지역	구매일자	구매시간	구매수량	구매금액	취소여부	구매지역_	구매지역_	구매월	구매요
4.78E+08	남	84	서울 성동	서울	성동구	가전제품	컴퓨터주변	서울 동대	20141219	13	1	59000	0	서울	동대문구	12	금
4.78E+08	남	84	서울 성동	서울	성동구	가전제품	TV/AV	서울 동대	20141031	14	1	106000	0	서울	동대문구	10	금
4.78E+08	남	84	서울 성동	서울	성동구	가전제품	주방가전	서울 중구	20140815	15	1	37000	0	서울	중구	8	금
4.78E+08	남	84	서울 성동	서울	성동구	의류잡화	여성용의류	서울 동대	20140322	17	1	118000	0	서울	동대문구	3	토
4.8E+08	남	84	서울 서초	서울	서초구	생활잡화	화장품	서울 중구	20140704	12	1	22000	0	서울	중구	7	금
4.8E+08	남	84	서울 서초	서울	서초구	생활잡화	화장품	서울 중구	20140704	12	1	26000	0	서울	중구	7	금
4.8E+08	남	84	서울 서초	서울	서초구	생활잡화	화장품	서울 중구	20140704	12	1	53000	0	서울	중구	7	금
4.8E+08	남	84	서울 서초	서울	서초구	가전제품	생활가전	서울 송파	20141218	13	1	120000	0	서울	송파구	12	목
4.8E+08	남	84	서울 서초	서울	서초구	가전제품	생활가전	서울 중구	20140607	14	1	97000	0	서울	중구	6	토
94790213	남	84	부산 사상	부산	사상구	생활잡화	생활용품	부산 부산	20140714	14	1	12000	0	부산	부산진구	7	월
94790213	남	84	부산 사상	부산	사상구	의류잡화	여성용의류	부산 부산	20140611	14	8	7000	0	부산	부산진구	6	수
94790213	남	84	부산 사상	부산	사상구	의류잡화	여성용의류	부산 부산	20140611	14	12	107000	0	부산	부산진구	6	수
94790213	남	84	부산 사상	부산	사상구	가전제품	생활가전	부산 부산	20141212	16	1	35000	0	부산	부산진구	12	금
94790213	남	84	부산 사상	부산	사상구	가전제품	TV/AV	부산 부산	20140529	17	1	23000	0	부산	부산진구	5	목
94790213	남	84	부산 사상	부산	사상구	식품	가공식품	부산 부산	20140623	17	2	10000	0	부산	부산진구	6	월

데이터는 CSV 로 위와 같이 구성이 되어있는 것을 먼저 확인할 수 있다.

2. 데이터 EDA(피처 탐색 및 시각화)

변수는 총 24 개가 있으며 ID, 성별, 연령, 거주지역, 거주지역_광역, 거주지역_기초, 상품대분류명, 상품중분류명, 구매지역, 구매일자, 구매시간, 구매수량, 구매금액, 취소여부, 구매지역_광역, 구매지역_기초, 구매월,구매요일, 총구매액, 구매계절, 구매시간대, eday, 구매연령대로 구성되어 있다. 또한 101692 개의 행, 즉 제품 구매에 대한 101692 개의 기록들이 있다.

```
Data columns (total 24 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Unnamed: 0   101692 non-null  int64
 1   ID           101692 non-null  int64
 2   성별           101692 non-null  object
 3   연령           101692 non-null  int64
 4   거주지역         101692 non-null  object
 5   거주지역_광역      101692 non-null  object
 6   거주지역_기초      101692 non-null  object
 7   상품대분류명       101692 non-null  object
 8   상품중분류명       101692 non-null  object
 9   구매지역         101692 non-null  object
 10  구매일자         101692 non-null  int64
 11  구매시간         101692 non-null  int64
 12  구매수량         101692 non-null  int64
 13  구매금액         101692 non-null  int64
 14  취소여부         101692 non-null  int64
 15  구매지역_광역      101692 non-null  object
 16  구매지역_기초      101692 non-null  object
 17  구매월          101692 non-null  int64
 18  구매요일         101692 non-null  object
 19  총구매액         101692 non-null  int64
 20  구매계절         101692 non-null  object
 21  구매시간대        101692 non-null  object
 22  eday         101692 non-null  object
 23  구매연령대        101692 non-null  object
```

데이터 전체의 Info 를 불러와 확인해본 결과, 결측치는 없으므로 따로 이에 대한 처리는 필요하지 않다.

예측을 위해 종속변수는 고객의 성별이 됐으며 피처들 사이에 의미가 중복되는 피처들이 많다. 따라서 모델을 통한 feature selection 에 앞서, 임의로 피처들을 선별하여 총 10 개의 피처들을 독립변수로 설정했다. 그 기준은 먼저 연령과 구매연령대처럼 의미가 중복되는 피처는 연령만을 선택하여 사용하기로 결정했다. 또한 성별을 판별하는데 무의미하다고 판단하는 피처는 사용하지 않았다. 선택 결과는 다음과 같다.

*** 의미가 중복되는 피처**

- 연령, 구매연령대 → 연령
- 상품대분류명, 상품 중분류명 → 상품 중분류명
- 구매일자, 구매요일 → 구매 요일
- 구매 계절, 구매 월 → 구매 계절

*** 무의미하다 판단한 피처**

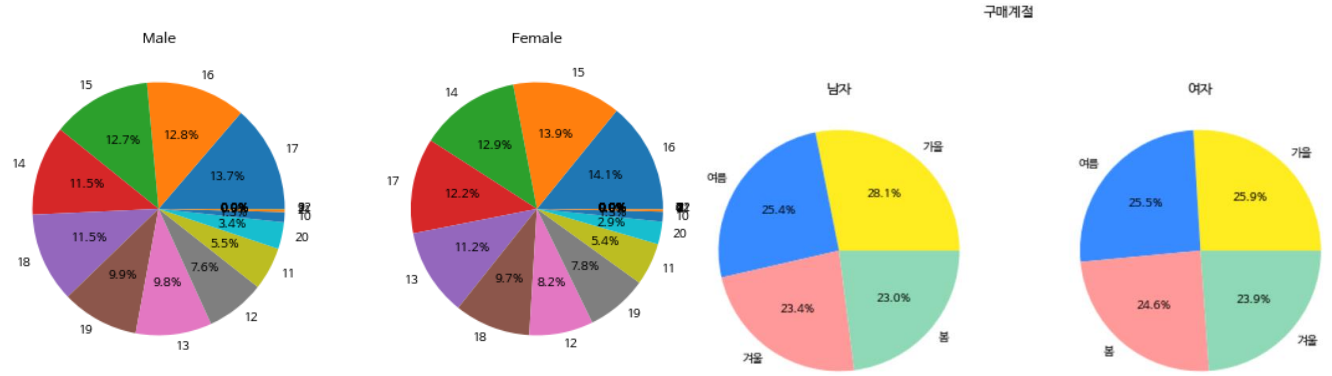
- ID, 구매시간대
- 구매지역, 구매지역_광역, 구매지역_기초
- 거주지역, 거주지역_광역, 거주지역_기초

*** 유의미하다 판단한 피처**

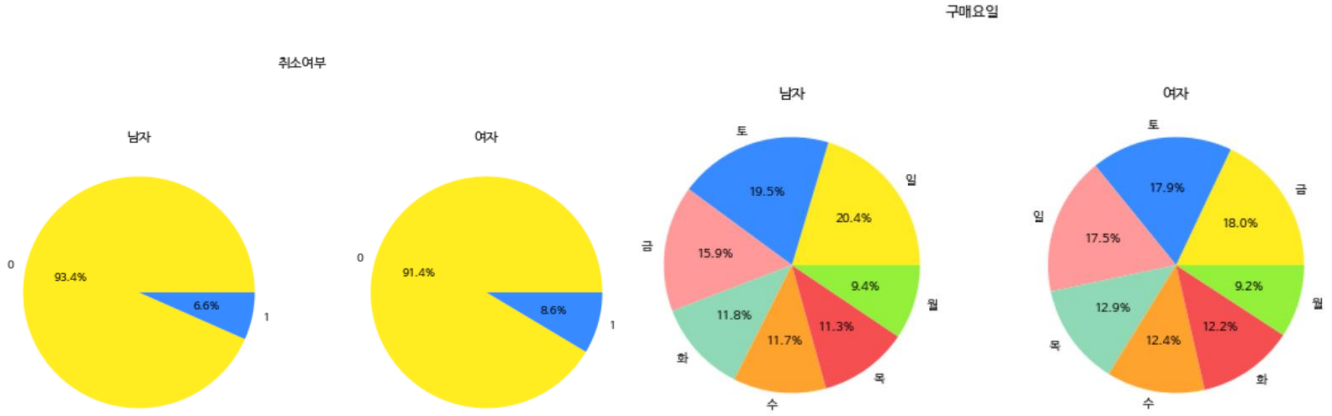
구매시간, 구매수량, 구매금액, 취소여부, 총구매액, eday, 성별

본격적인 분석을 들어가기 전, 위에서 선택한 변수들을 시각화를 통해 간단하게 살펴보며 방향을 잡을 수 있다.

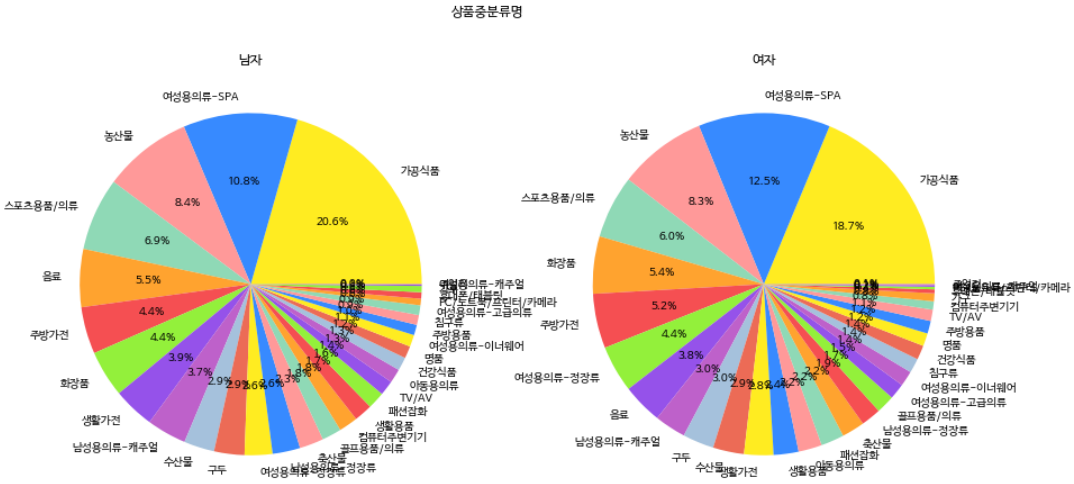
먼저 이산형 변수들을 파이차트로 시각화하여 살펴보았다.



구매시간은 여자와 남자 모두 오후 2 시~5 시에 가장 많다는 것을 관찰할 수 있다. 여자는 구매계절에서 큰 차이를 보이고 있지 않지만 남자는 가을에 28%로 가장 많이 구매하는 것을 알 수 있다.

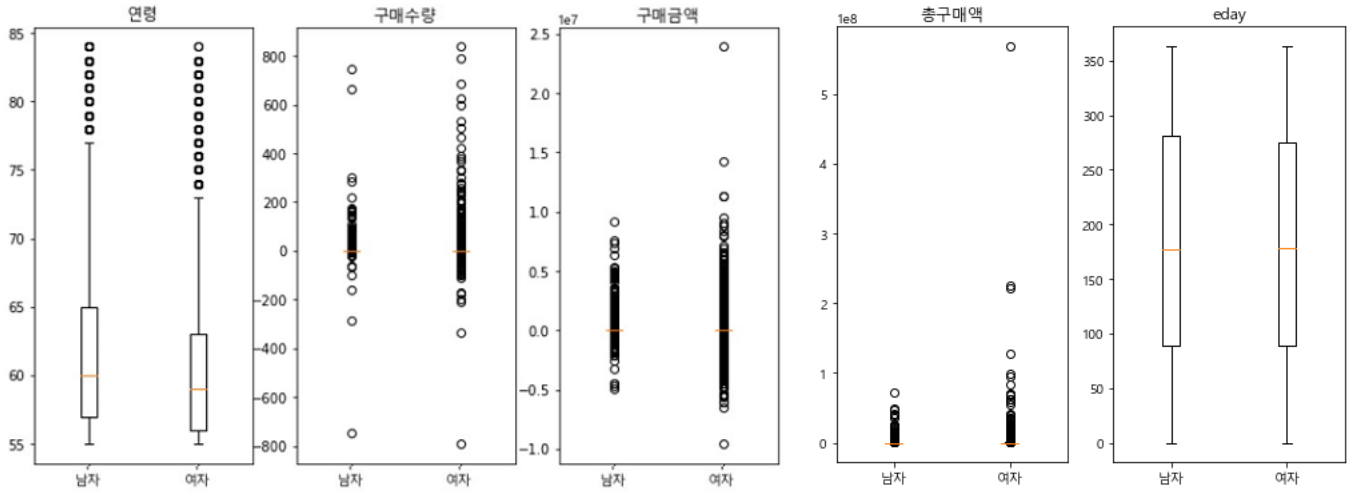


취소여부는 여자가 2% 정도 더 높지만 큰 차이를 보이고 있지는 않다. 구매요일은 남녀 모두 금~일요일에 가장 많이 사는 것을 관찰할 수 있다.



상품중분류명은 남녀별로 왼쪽과 같이 너무 많은 카테고리로 나뉘어져 있어 한눈에 파악하기에 한계가 있다.

다음으로 연속형 변수들을 시각화를 통해 살펴본다.



먼저 연령은 대체로 남자가 더 높은 편이다. 하지만 이 데이터만의 고유 특성이기 때문에 성별과의 연관요인이라고 단정지을 수는 없다.

구매수량은 거의 다 0 과 1 사이라서 boxplot 이 잘 보이지 않는다. 따라서 이상치들이 더욱 눈에 띈다.

구매금액은 범위가 굉장히 넓어서 10000000 로 나눈 값을 시각화한 모습이다. 하지만 이 역시 그래프 하나만으로 단순히 판단하기는 어렵다.

총구매액도 마찬가지로 boxplot 으로 판단하기는 어렵지만 여자에 해당하는 이상치가 많은 것으로 확인됐다.

마지막으로 eday 는 여자와 남자 간의 차이가 거의 없다고 볼 수 있다.

3. 데이터 전처리 및 데이터 불균형 해소

앞서 독립 변수로 '연령', '상품중분류명', '구매시간', '구매수량', '구매금액', '취소여부', '구매요일', '총구매액', '구매계절', 'eday', 총 10 개의 피처들이 선택됐다. 이 중에서 상품중분류명, 구매요일 및 구매계절은 연속형 변수가 아니기 때문에 label-encoding 을 문자열 타입을 숫자로 변환해준다.

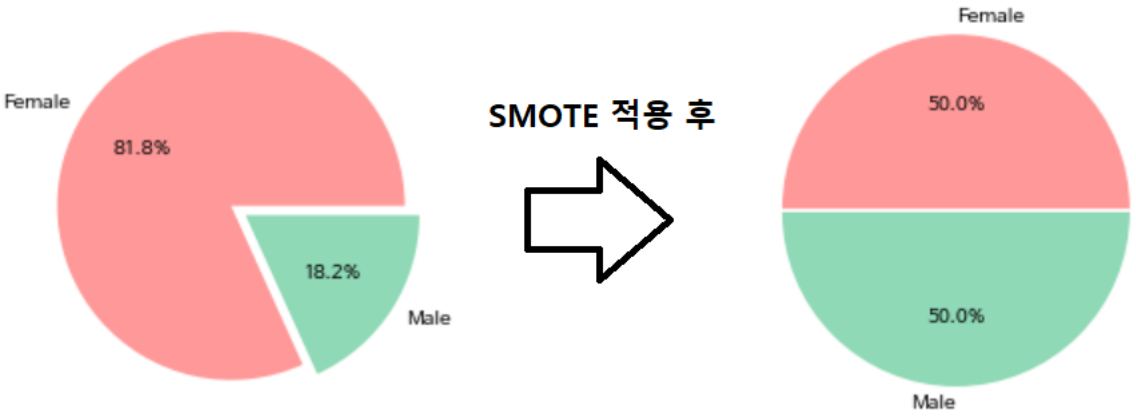
또한 eday 는 12 days 00:00:00.00000000 와 같은 형태로 있기 때문에 분석에 있어서 용이성을 위해 첫 숫자만을 남겨두었다.

데이터 스케일링이 필수는 아니지만 데이터셋의 column 들 간의 숫자의 스케일이 차이가 났기 때문에 선택적으로 표준화를 진행해줬다. StandardScaler 를 사용하면 평균과 표준편차를 사용하여 표준화를 시켜준다.

가장 중요한 성별 데이터의 분포를 확인해보면 여성이 83150, 남성이 18452 의 빈도수로 여성이 압도적으로 높은 빈도수를 보이고 있었다. 물론 decision tree 나 앙상블학습 모델은 불균형 자료에서도 성능을 잘 유지하지만 그 이외 머신러닝 모형들은 클래스 불균형이 심한 경우, 정확도(accuracy)가 높아도 재현율(recall)이 급격히 감소하는 현상을 보이므로 이에 대한 해결이 필요하다.

이러한 클래스 불균형 처리에 대한 해결책으로 데이터 샘플링으로 과소표집 혹은 과대표집을 실행할 수 있는데 우리는 통계적으로 유용한 오버샘플링을 사용하여 소수클래스의 표본을 복제하여 이로부터 train 데이터를 추가한다. 이때 주의해야 할 점으로는, 오버샘플링을 시행하면 무작위 추출, 유의정보, 합성 데이터 생성 등 정보가 손실되지 않는다는 장점이 있으나, 복제된 관측치를 원래 데이터 세트에 추가하면 여러 유형의 관측치를 다수 추가하여 오버피팅을 초래할 수 있다는 단점이 있다. 따라서 이에 대한 비용 민감 학습으로 **SMOTE** 알고리즘을 시행하기로 했다.

SMOTE 은 오버샘플링 기법 중 합성데이터를 생성하는 방식으로 가장 많이 사용되고 있는 모델이다. 합성 소수 샘플링 기술로 다수 클래스를 샘플링하고 기존 소수 샘플을 보간하여 새로운 소수 인스턴스를 합성해내는 원리를 갖고 있다.



SMOTE 를 적용하게 된다면 다음과 같이 최종적으로 남녀 빈도수가 83150 으로 맞춰져서 분포가 동일해진다는 것을 확인할 수 있다.

```
SMOTE 적용 전 학습용 피처/레이블 데이터 세트:  (101692, 49)
0         남
1         남
2         남
3         남
4         남
         ..
101687    여
101688    여
101689    여
101690    여
101691    여
Name: 성별, Length: 101692, dtype: object
SMOTE 적용 후 학습용 피처/레이블 데이터 세트:  (166300, 49) (166300,)
SMOTE 적용 후 레이블 값 분포:
 남    83150
여    83150
```

SMOTE 를 적용하기 전과 후의 결과는 다음 절, 모델 분석에서 확인할 수 있다.

4. 모델링 및 분석

모델링을 위하여, 최적의 모델을 찾기 위해 최대한 많은 모델을 적용해보았다. KNN, SVM, Logistic Regression, Decision Tree, Random Forest, Ensemble, GBM, XGBoost, LightGBM, KMeans 을 포함한 10 개의 모델을 통해서 SMOTE 전화 후의 결과를 비교할 수 있고 최종적으로 성별 예측을 위해 가장 적합한 모델을 분석해보자. 각 모델의 선택 이유 및 과정은 각 분석에서 기술을 한다.

먼저 임의로 선택한 피처들로 SMOTE 실행 전과 후를 비교하고, 이후에 feature selection 을 진행하여 다시 비교를 한다. 모델과 데이터가 너무 큰 관계로 한 번 실행하는데 5 시간이 넘어가므로, 최종적으로 가장 좋은 모델 하나에서 파라미터 조정을 해준다.

여기서, Cross validation 을 통해 train set 에 대해 validation score 를 비교해보고, test data 에 대하여 정확도인 accuracy 를 중점으로 confusion_matrix 와 classification_report 를 활용하여 두 예측 값을 비교하는 방식으로 진행을 하였다.

① KNN

가장 먼저 주변 이웃들의 클래스에 기반하여 데이터의 클래스를 지정하는 KNN 을 적용하였다. 이때, 최적의 k(nearest neighbors)을 찾기 위해 값이 3, 5, 7 인 경우를 적용해보았다.

SMOTE 적용 전

```
validation scores:  [0.79332434 0.7921443  0.79286076]
mean of validation score:  0.7927764666217127
[[ 1681  3860]
 [ 2072 22895]]
```

	precision	recall	f1-score	support
남	0.45	0.30	0.36	5541
여	0.86	0.92	0.89	24967
accuracy			0.81	30508
macro avg	0.65	0.61	0.62	30508
weighted avg	0.78	0.81	0.79	30508

K=3

```
validation scores:  [0.79909811 0.80276467 0.80322825]
mean of validation score:  0.8016970105641716
[[ 1214  4327]
 [ 1426 23541]]
```

	precision	recall	f1-score	support
남	0.46	0.22	0.30	5541
여	0.84	0.94	0.89	24967
accuracy			0.81	30508
macro avg	0.65	0.58	0.59	30508
weighted avg	0.77	0.81	0.78	30508

K=5

```
validation scores:  [0.80723196 0.80870701 0.80963419]
mean of validation score:  0.8085243875028096
[[  879  4662]
 [ 1048 23919]]
```

	precision	recall	f1-score	support
남	0.46	0.16	0.24	5541
여	0.84	0.96	0.89	24967
accuracy			0.81	30508
macro avg	0.65	0.56	0.56	30508
weighted avg	0.77	0.81	0.77	30508

K=7

SMOTE 적용 후

```
validation scores:  [0.78968663 0.78803185 0.79200062]
mean of validation score:  0.7899063673201363
[[22996  1893]
 [ 6852 18149]]
```

	precision	recall	f1-score	support
남	0.77	0.92	0.84	24889
여	0.91	0.73	0.81	25001
accuracy			0.82	49890
macro avg	0.84	0.82	0.82	49890
weighted avg	0.84	0.82	0.82	49890

```
validation scores:  [0.75793733 0.75512203 0.76035358]
mean of validation score:  0.7578043112016593
[[22553  2336]
 [ 7906 17095]]
```

	precision	recall	f1-score	support
남	0.74	0.91	0.81	24889
여	0.88	0.68	0.77	25001
accuracy			0.79	49890
macro avg	0.81	0.79	0.79	49890
weighted avg	0.81	0.79	0.79	49890

```
validation scores:  [0.73778476 0.73587094 0.74009742]
mean of validation score:  0.7379177058065575
[[22104  2785]
 [ 8496 16505]]
```

	precision	recall	f1-score	support
남	0.72	0.89	0.80	24889
여	0.86	0.66	0.75	25001
accuracy			0.77	49890
macro avg	0.79	0.77	0.77	49890
weighted avg	0.79	0.77	0.77	49890

결과를 확인하면 Validation score 는 SMOTE 적용 후, k=3 일 때 가장 높다. SMOTE 을 적용하기 전에는 k=3 일 때 가장 작긴 하지만 0.01 이내의 차이로 큰 변화는 없다. Test 에 대한 정확도는 SMOTE 를 적용하기 전과 후, 모두 k 를 3 으로 지정해주었을 때 성능이 가장 높은 것을 확인할 수 있다.

② SVM

다음은 서포트벡터머신(SVM)을 적용해보았다. 커널은 rbf(방사형기저함수)을 사용하여 정규분포모형의 비선형 적합을 이용하였다. SVM 은 결측 자료 및 이상치에 굉장히 민감하게 반응하는 특성이 있다.

```
validation scores:  [0.81751517 0.81747303 0.81726231]
mean of validation score:  0.8174168352438751
[[    2  5539]
 [    2 24965]]
              precision    recall  f1-score   support

           남       0.50      0.00      0.00      5541
           여       0.82      1.00      0.90     24967

    accuracy                           0.82     30508
   macro avg       0.66      0.50      0.45     30508
weighted avg       0.76      0.82      0.74     30508
```

SMOTE 적용 후

```
validation scores:  [0.59535099 0.59075329 0.58969667]
mean of validation score:  0.5919336533172285
[[14518 10371]
 [ 9604 15397]]
              precision    recall  f1-score   support

           남       0.60      0.58      0.59     24889
           여       0.60      0.62      0.61     25001

    accuracy                           0.60     49890
   macro avg       0.60      0.60      0.60     49890
weighted avg       0.60      0.60      0.60     49890
```

결과를 확인해보면 자료불균형을 해소하며 성능을 높여줄거라는 예상과 달리 SMOTE 후에는 validation 및 test 에 대한 정확도가 감소하였다. 이는 SMOTE 는 minor class 인 남성의 가상 데이터를 생성하는 동안 인접해 있는 major class 인 여성의 데이터들의 위치는 고려하지 않기에 class 가 겹치거나 노이즈를 만들어졌을 수 있기 때문이라고 예측할 수 있다. 따라서 SVM 에 적용할 경우, 밴드를 사이에 두고 일정한 간격을 유지하려고 하는데 SMOTE 과정에서 class 가 겹치거나 노이즈가 생성될 경우 앞서 기술한 특성처럼 서프토 벡터들이 오분류되어 성능이 저하될 수 있다.

③ Logistic 회귀 분류

로지스틱 회귀는 sigmoid 함수 및 threshold 를 사용하여 회귀를 통해 분류를 하는 기법이다. 이 역시 결측 자료 및 이상치에 굉장히 민감하게 반응하는 특성이 있다. 로지스틱 회귀를 적용하기 위해 먼저 성별 column 을 문자열이 아닌 숫자형으로 바꿔주었다.

```
validation scores:  [0.81738874 0.81734659 0.81734659]
mean of validation score:  0.8173606428410878
[[    0  5541]
 [    0 24967]]
              precision    recall  f1-score   support

           1       0.00      0.00      0.00      5541
           2       0.82      1.00      0.90     24967

    accuracy                           0.82     30508
   macro avg       0.41      0.50      0.45     30508
weighted avg       0.67      0.82      0.74     30508
```

SMOTE 적용 후

```
validation scores:  [0.56239048 0.56047213 0.5550344 ]
mean of validation score:  0.5592990027362558
[[12775 12114]
 [ 9719 15282]]
              precision    recall  f1-score   support

           1       0.57      0.51      0.54     24889
           2       0.56      0.61      0.58     25001

    accuracy                           0.56     49890
   macro avg       0.56      0.56      0.56     49890
weighted avg       0.56      0.56      0.56     49890
```

Logistic Regression 의 경우, SVM 과 유사하게 SMOTE 과정을 거치면서 validation 과 test 모두에 대한 정확도가 감소하였다. 그 이유를 분석해보기 위해 정확도와 AUC-ROC 커브를 비교해보면 다음과 같은 결과가 나온다.

SMOTE 전

```
accuracy:0.818
ROC accuracy:0.501
최적 하이퍼 파라미터:{'C': 0.01, 'penalty': 'l2'}, 최적 평균 정확도:0.817
```

SMOTE 후

```
accuracy:0.578
ROC accuracy:0.578
최적 하이퍼 파라미터:{'C': 5, 'penalty': 'l2'}, 최적 평균 정확도:0.576
```

이때 SMOTE 를 통해 정확도는 감소하지만 오히려 ROC 정확도는 증가하는 것을 알 수 있다. 따라서 우리는 SMOTE 를 거치면서 노이즈가 커졌기 때문에 위에 설명한 로지스틱 회귀의 특성처럼 이에 민감하게 반응하여 accuracy 가 감소한 것임을 유추할 수 있다.

④ Decision Tree

의사결정나무 중 하나인 Decision Tree 를 사용하여 분류를 할 수 있다. 이는 Gini Index 를 통하여 불순도 측도를 측정하였다.

```
validation scores:  [0.7607468  0.75939818 0.76192684]
mean of validation score:  0.7606906046302541
[[ 2471  3070]
 [ 3696 21271]]
```

	precision	recall	f1-score	support
남	0.40	0.45	0.42	5541
여	0.87	0.85	0.86	24967
accuracy			0.78	30508
macro avg	0.64	0.65	0.64	30508
weighted avg	0.79	0.78	0.78	30508

SMOTE 적용 후

```
validation scores:  [0.84212968 0.83697137 0.8413267 ]
mean of validation score:  0.8401425823632421
[[21768  3121]
 [ 3884 21117]]
```

	precision	recall	f1-score	support
남	0.85	0.87	0.86	24889
여	0.87	0.84	0.86	25001
accuracy			0.86	49890
macro avg	0.86	0.86	0.86	49890
weighted avg	0.86	0.86	0.86	49890

의사결정나무의 결과를 확인해보면 SMOTE 후, validation 및 test 에 대한 전체적인 정확도가 상승했음을 알 수 있다. 이는 2 가지 모형 모두에 우수한 성능을 보이며 불균형 자료 분류에도 높은 우수성을 보이는 의사결정나무의 특성을 보여준다. 남자의 정밀도, recall 과 f1-score 는 증가한 반면 여자의 정밀도, recall 과 f1-score 는 감소했다는 점에서 한계를 가짐을 보인다.

⑤ Random Forest

Random Forest 는 여러 개의 결정트리(Decision Tree)들이 모여 구성하는 앙상블 학습 기법의 한 종류이다. Tree 들을 통한 Bagging 을 이용하여 결정트리에서 흔히 발생하는 과대적합 문제를 해결할 수 있다.

```
validation scores:  [0.83715442 0.83787087 0.83458361]
mean of validation score:  0.8365363002922005
[[ 1223  4318]
 [  305 24662]]
```

	precision	recall	f1-score	support
남	0.80	0.22	0.35	5541
여	0.85	0.99	0.91	24967
accuracy			0.85	30508
macro avg	0.83	0.60	0.63	30508
weighted avg	0.84	0.85	0.81	30508

SMOTE 적용 후

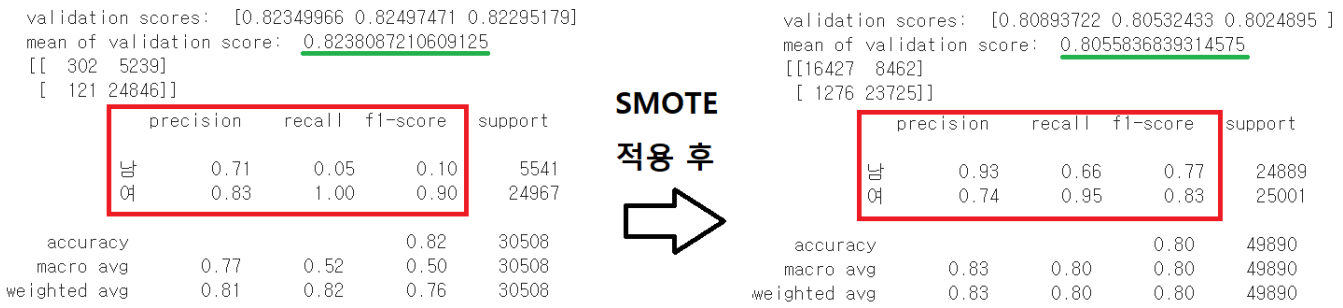
```
validation scores:  [0.89879909 0.89472463 0.89423498]
mean of validation score:  0.8959195698005061
[[21346  3543]
 [ 1061 23940]]
```

	precision	recall	f1-score	support
남	0.95	0.86	0.90	24889
여	0.87	0.96	0.91	25001
accuracy			0.91	49890
macro avg	0.91	0.91	0.91	49890
weighted avg	0.91	0.91	0.91	49890

랜덤포레스트의 결과를 보면 SMOTE 이후, validation 과 test 에 대한 전체적인 정확도 대폭 증가하며 Decision Tree 의 결과와 비교를 해도 우수한 성능을 냄을 알 수 있다. 여자의 정밀도가 상승했지만 눈여겨볼 점은 남자의 정밀도에서 눈에 띄는 상승이 있었다는 점이다. 특히 recall 과 f1-score 는 0.6 점 정도의 상승이 있었다는 점에서 Random Forest 모델에 SMOTE 를 통하여 자료 불균형 해소를 적절히 이뤘음을 알 수 있다.

⑥ GBM

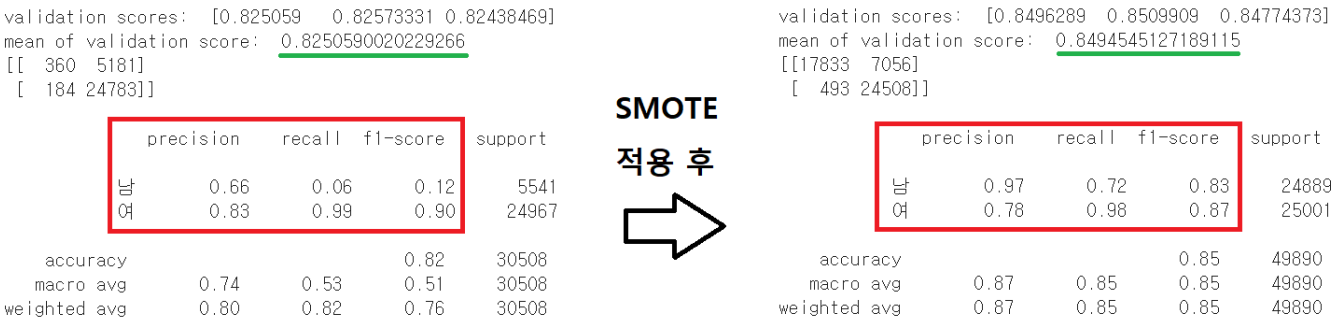
기울기부스팅은 머신러닝 알고리즘 중에서도 가장 예측 성능이 높은 모델 중 하나이다. 앙상블 학습 모델이지만 랜덤포레스트와 달리 Boosting 을 이용하여 weak learner 들을 통해 모델을 개선해 나가는 학습방법이라는 특성을 지닌다.



GBM 의 결과를 살펴보면 SMOTE 를 적용한 후, validation 과 test 정확도가 둘 다오히려 미세하게 감소함을 알 수 있다. 이를 자세히 살펴보면 자료 불균형 해소를 통해 남자의 정밀도, recall 과 f1-score 는 증가했지만 여자의 recall 과 f1-score 는 감소했다는 한계를 가짐을 알 수 있다.

⑦ XGBoost

XGBoost 는 GBM 과 유사한 학습방식을 갖고 있지만 과대적합을 방지하기 위한 정규화를 실행한다. 또한 피처들을 임의로 일부를 선택하여 사용하기도 해서 성능을 높인다는 장점을 갖고 있다.



XGBoost 의 결과를 확인해보면 SMOTE 후, validation score, test 셋에 대한 전체 정확도, 남자와 여자 모두의 정밀도, recall 과 f1-score 가 증가했다는 것을 알 수 있다. XGBoost 에서는 GBM 과 비교하여 SMOTE 후의 성능이 더 우수하다는 것을 알 수 있는데 이는 위에서 설명한 XGBoost 의 장점이 작용했다는 것으로 유추해볼 수 있다.

⑧ LightGBM

LightGBM 은 다시 한번 XGBoost 와 유사한 알고리즘이지만 leaf 에 대해 트리를 확장해나가며 좌우로 불균형한 나무로 학습을 하게 된다. 따라서 이는 같은 leaf 의 수로 XGBoost 보다 높은 성능을 보일 수 있다는 장점이 있다.

LightGBM 의 결과를 확인해보면 SMOTE 후, validation 과 test 에 대한 정확도가 증가했으며 남자의 정밀도, recall 및 f1-score 이 모두 증가했지만 여자의 정밀도, recall 및 f1-score 가 모두 감소했음을 알 수 있다.

⑨ Kmeans

마지막으로 분류 기법은 아니지만 비슷한 유형의 점들을 군집화함으로써 데이터를 K 개의 클러스터로 나누는 Kmeans 를 사용했다. 이는 X 의 피처들을 기반으로 2 개의 군집, 즉 여성과 남성을 구분하도록 모델을 형성했다.

```
validation scores:  [0.78898348 0.57307822 0.21122724]
mean of validation score:  0.5244296471117106
[[ 3249  2292]
 [14975  9992]]
```

	precision	recall	f1-score	support
0	0.18	0.59	0.27	5541
1	0.81	0.40	0.54	24967
accuracy			0.43	30508
macro avg	0.50	0.49	0.40	30508
weighted avg	0.70	0.43	0.49	30508

SMOTE 적용 후 ⇨

```
validation scores:  [0.49819606 0.50248692 0.49985826]
mean of validation score:  0.5001804139193511
[[ 1131 23758]
 [ 1093 23908]]
```

	precision	recall	f1-score	support
0	0.51	0.05	0.08	24889
1	0.50	0.96	0.66	25001
accuracy			0.50	49890
macro avg	0.51	0.50	0.37	49890
weighted avg	0.51	0.50	0.37	49890

Kmeans 의 결과를 확인해보면 SMOTE 적용 후, validation score 는 감소, test set 의 전체적인 정확도가 증가했으며 남성은 대체적으로 감소를 했지만 여성의 recall 과 f1-score 가 많이 증가했음을 알 수 있다. 하지만 SMOTE 전과 후 모두 정확도가 0.43 과 0.50 으로 모두 우수한 성능을 보이지 못한다. 이의 이유로는 이상치에 민감하기 때문에 노이즈가 발생할 시, 중심점을 잘 찾지 못하며 원형(spherical)의 cluster 가 아니면 군집화가 제대로 작동할 수 없기 때문이라고 추측할 수 있다.

⑩ Ensemble Learning

앞서 사용한 9 개의 모델을 활용하여 각각 SMOTE 를 적용하기 전과 후를 비교해보았다. 이 중에서 다음과 같이 validation 에서 가장 우수한 성능을 보인 모형들을 앙상블 학습에 적용하기로 한다.

✧ Random Forest

첫번째로 Random Forest 같은 경우에는 (SMOTE 전과 후를 포함하여) 총 18 개의 모델 중 validation score 가 0.89 로 가장 높게 나왔다. 특히 SMOTE 적용 후에 전반적인 성능이 증가하였으며 남녀 모두에서 test 에 대한 precision, recall, f1-score 이 0.9 에 수렴하는 우수함을 보인다.

✧ LightGBM

LightGBM 역시 SMOTE 후, validation score 가 0.88 에 육박하는 우수한 성능을 보였다. 또한 자료불균형을 해소하기 위해 남성의 데이터를 추가함에 따라 test data 에 대해 여자의 f1-score 가 거의 모든 모형에서 하락했지만 LightGBM 에서는 0.01 감소폭으로 남자뿐만 아니라 여자 분류에서도 좋은 성능을 보였다.

따라서 이 validation score 가 가장 높은 두 모델을 이용하여 SMOTE 전과 후를 비교하여 앙상블 학습을 시행하면 다음 같은 결과가 나왔다.

```
validation scores:  [0.83336143 0.83230782 0.83192852]
mean of validation score:  0.8325325915936165
[[  810  4731]
 [  161 24806]]
```

	precision	recall	f1-score	support
남	0.83	0.15	0.25	5541
여	0.84	0.99	0.91	24967
accuracy			0.84	30508
macro avg	0.84	0.57	0.58	30508
weighted avg	0.84	0.84	0.79	30508

SMOTE 적용 후 ⇨

```
validation scores:  [0.90132461 0.90302296 0.90178594]
mean of validation score:  0.9020445040795324
[[20679  4210]
 [  386 24615]]
```

	precision	recall	f1-score	support
남	0.98	0.83	0.90	24889
여	0.85	0.98	0.91	25001
accuracy			0.91	49890
macro avg	0.92	0.91	0.91	49890
weighted avg	0.92	0.91	0.91	49890

앙상블 학습(Random Forest+LightGBM)의 결과를 확인해보면 SMOTE 후, validation score 가 0.90, test 정확도가 0.91 로 높은 숫자를 기록했다. 또한 여성의 재현율을 제외하면 남성과 여성 모두의 정밀도, 재현율, f1-score 가 증가하였음을 알 수 있다. 단순히 정확도로만 따지면 랜덤포레스트와 함께 가장 우수한 면을 보였지만 정밀도, 재현율과 f1-score 모두를 총체적으로 살펴보면 9 개의 모델 중 가장 성능이 좋았던 Random Forest 및 LightGBM 에 비해서도 향상된 결과를 증명했다.

5. 임의 피처를 통한 모델별 결과 분석 결론

SMOTE 를 적용했을 때 가장 성능이 좋은 모델은 LightGBM 과 Random Forest 를 이용한 Ensemble 모델이라는 결론을 도출할 수 있다.

자료불균형 문제를 해소하기 위해 SMOTE 를 적용했지만, 이를 적용한다고 해서 무조건 모든 모델의 성능이 개선되는 것은 아니다. 또한 SMOTE 를 적용하기 전, Minor Class 의 Recall(재현율)이 심각하게 낮은 것은 해당 Class 에 속한 데이터의 수가 현저하게 작기 때문에 일어난 현상이다. (특히, 데이터들의 분포 및 이상치에 영향을 받는 KNN 과 SVM 의 경우 해당 문제가 도드라지게 보인다)

6. Feature Selection 적용 후 결과와 비교

위의 방법의 한계점으로는 바로 전처리 과정에서 감으로 피처들을 선택하며 배제할 데이터를 선정했다는 것이다. 알고리즘을 선택하여 피처 선택에 있어서 근거를 제시하면 더 강력한 모델을 세울 수 있을 것이다.

이로 분류에 있어서 feature selection 으로 사용되는 SelectPercentile, f_classif 를 이용할 수 있다. F_classif 은 일변량 통계에서는 개개의 특성과 타깃 사이에 중요한 통계적 관계가 있는지를 계산하고 관련되어 있다고 판단되는 특성을 선택한다. 계산한 p-value 에 기초하여 특성을 제외하는 방식을 선택한다.

이 역시 자료불균형의 문제를 해소하기 위해 SMOTE 를 사용한다.

피처들 24 개 중, 성별은 종속변수이기 때문에 제외를 해야한다. 또한 ID 과 Unnamed 는 분류에 있어서 의미를 갖지 않는 변수들이기 때문에 이 역시 드랍해주면 총 21 개의 변수들이 남는다. 우리는 앞서 임의로 선택했던 변수들과 비교해보기 위해 동일하게 개수가 10 개가 나올 수 있도록 설정해준다.

```
X_train.shape (116410, 21)
X_train_selected.shape (116410, 10)
```

다음과 같이 10 개의 피처들이 선택되었다.

```
Index(['연령', '상품대분류명', '상품중분류명', '구매일자', '구매시간', '취소여부', '구매월', '구매요일',
       '구매시간대', '구매연령대'],
      dtype='object')
```

10 개 중 절반 이상이 앞서 임의로 선택한 피처들과 겹치는 것을 알 수 있다.

새롭게 선택된 피처들에 대해서도 10 개의 모델을 적용하면 다음과 같다.

① KNN

K=3

```
validation scores:  [0.8526956  0.85297529 0.85593897]
mean of validation score:  0.8538699525328936
[[24130   759]
 [ 4794 20207]]
```

	precision	recall	f1-score	support
남	0.83	0.97	0.90	24889
여	0.96	0.81	0.88	25001
accuracy			0.89	49890
macro avg	0.90	0.89	0.89	49890
weighted avg	0.90	0.89	0.89	49890

K=5

```
validation scores:  [0.82561076 0.82612169 0.82545164]
mean of validation score:  0.8257280312453351
[[23757  1132]
 [ 5989 19012]]
```

	precision	recall	f1-score	support
남	0.80	0.95	0.87	24889
여	0.94	0.76	0.84	25001
accuracy			0.86	49890
macro avg	0.87	0.86	0.86	49890
weighted avg	0.87	0.86	0.86	49890

K=7

```
validation scores:  [0.80509741 0.80370074 0.80362343]
mean of validation score:  0.8041405295345494
[[23412  1477]
 [ 6616 18385]]
```

	precision	recall	f1-score	support
남	0.78	0.94	0.85	24889
여	0.93	0.74	0.82	25001
accuracy			0.84	49890
macro avg	0.85	0.84	0.84	49890
weighted avg	0.85	0.84	0.84	49890

Feature selection 과 KNN 을 모두 적용한 결과, 마찬가지로 k=3 일 때 가장 우수한 성능을 보였다. validation score 는 0.85 로 0.06 정도를 증가했고 test 에 대한 accuracy 도 0.81 에서 0.89 로 증가하며 좋은 결과를 냈다.

② SVM

```
validation scores:  [0.677224   0.67592712 0.67430353]
mean of validation score:  0.6758182165983943
[[16915  7974]
 [ 7555 17446]]
```

	precision	recall	f1-score	support
남	0.69	0.68	0.69	24889
여	0.69	0.70	0.69	25001
accuracy			0.69	49890
macro avg	0.69	0.69	0.69	49890
weighted avg	0.69	0.69	0.69	49890

Feature selection 과 SVM 을 모두 적용한 결과 validation score 는 0.68 로 0.08 정도를 증가했고 test 에 대한 accuracy 도 0.60 에서 0.69 로 증가하며 좋은 결과를 냈다. 하지만 여전히 다른 모델에 비해 성능이 떨어지는데 SMOTE 과정에서 class 가 겹치거나 노이즈가 생성될 경우 앞서 기술한 특성처럼 서프토 벡터들이 오분류되어 성능이 저하될 가능성 때문이다.

③ LOGISTIC 회귀 분류

```
[[13075 11814]
 [ 9849 15152]]
```

	precision	recall	f1-score	support
1	0.57	0.53	0.55	24889
2	0.56	0.61	0.58	25001
accuracy			0.57	49890
macro avg	0.57	0.57	0.57	49890
weighted avg	0.57	0.57	0.57	49890

```
validation scores:  [0.56620451 0.56402855 0.55779192]
mean of validation score:  0.5626749975988541
accuracy:0.566
ROC accuracy:0.566
최적 하이퍼 파라미터:{'C': 10, 'penalty': 'l2'}, 최적 평균 정확도:0.563
```

Feature selection 과 로지스틱 회귀 분류를 모두 적용한 결과 validation score 는 0.57 로 0.01 정도를 증가했고 test 에 대한 accuracy 도 0.56 에서 0.57 로 증가하였다. 하지만 feature selection 을 수행한 것에 비하여 증가폭이 크지 않았고 여전히 노이즈에 민감한 모델의 특성 때문에 좋은 성능을 보이지 못했다.

④ Decision Tree

```
validation scores:  [0.90346356 0.90438884 0.90652785]
mean of validation score:  0.9047934140525178
[[23154  1735]
 [ 2036 22965]]

              precision    recall  f1-score   support

          남       0.92      0.93      0.92     24889
          여       0.93      0.92      0.92     25001

    accuracy                           0.92     49890
   macro avg       0.92      0.92      0.92     49890
weighted avg       0.92      0.92      0.92     49890
```

Feature selection 과 의사결정나무를 모두 적용한 결과 validation score 는 0.90 로 0.06 정도를 증가했고 test 에 대한 accuracy 도 0.86 에서 0.92 로 증가하며 좋은 결과를 냈다. 이는 앞서 임의로 feature 들을 선택하여 적용했던 의사결정나무에 비해 눈에 띄는 정확도의 개석이 있었다. 모든 모델을 통틀어서 0.92 로 가장 좋은 성능을 보인다. 이는 불균형 자료 분류에도 높은 우수성을 보이는 의사결정나무의 특성을 다시 한번 보여준다.

⑤ Random Forest

```
validation scores:  [0.92467374 0.92338299 0.92400149]
mean of validation score:  0.9240194085187144
[[21884  3005]
 [  777 24224]]

              precision    recall  f1-score   support

          남       0.97      0.88      0.92     24889
          여       0.89      0.97      0.93     25001

    accuracy                           0.92     49890
   macro avg       0.93      0.92      0.92     49890
weighted avg       0.93      0.92      0.92     49890
```

Feature selection 과 Random Forest 를 모두 적용한 결과 validation score 는 0.92 로 0.02 정도를 증가했고 test 에 대한 accuracy 도 0.91 에서 0.92 로 증가하며 좋은 결과를 냈다. 이는 앞서 진행했던 의사결정나무보다 더 좋은 validation score 를 보인다. Random forest 는 여러 개의 decision tree 들을 함께 학습시키는 앙상블 학습으로, 의사결정나무보다 더 좋은 성능을 낸 것임을 확인할 수 있다.

⑥ GBM

```
validation scores:  [0.82888362 0.82813185 0.82756488]
mean of validation score:  0.8281934482418004
[[17474  7415]
 [ 1097 23904]]

              precision    recall  f1-score   support

          남       0.94      0.70      0.80     24889
          여       0.76      0.96      0.85     25001

    accuracy                           0.83     49890
   macro avg       0.85      0.83      0.83     49890
weighted avg       0.85      0.83      0.83     49890
```

Feature selection 과 Gradient Boosting Classifer 를 모두 적용한 결과 validation score 는 0.83 로 0.005 정도를 증가했고 test 에 대한 accuracy 도 0.82 에서 0.83 로 증가했다. 하지만 앞서 feature selection 을 통해 증가폭이 컸던 모델들과는 달리 정확도에 대한 별다른 개선은 없었음을 확인할 수 있다.

⑦ XGBoost

```
validation scores:  [0.87689413 0.87776718 0.87477772]
mean of validation score:  0.8764796803156575
[[19346  5543]
 [  665 24336]]

              precision    recall  f1-score   support

           남       0.97      0.78      0.86     24889
           여       0.81      0.97      0.89     25001

    accuracy                           0.88     49890
   macro avg       0.89      0.88      0.87     49890
weighted avg       0.89      0.88      0.87     49890
```

Feature selection 과 XGBoost 를 모두 적용한 결과 validation score 는 0.88 로 0.03 정도를 증가했고 test 에 대한 accuracy 도 0.85 에서 0.88 로 증가하며 좋은 결과를 보였다. XGBoost 는 GBM 과 유사한 학습 알고리즘을 갖고 있지만 정규화를 통해 차별화를 하며 정확도를 개선하였음을 결과로 증명한다.

⑧ LightGBM

```
validation scores:  [0.91671065 0.91183749 0.91255908]
mean of validation score:  0.9137024052208302
[[22615  2274]
 [ 1816 23185]]

              precision    recall  f1-score   support

           남       0.93      0.91      0.92     24889
           여       0.91      0.93      0.92     25001

    accuracy                           0.92     49890
   macro avg       0.92      0.92      0.92     49890
weighted avg       0.92      0.92      0.92     49890
```

Feature selection 과 LightGBM 를 모두 적용한 결과 validation score 는 0.91 로 0.03 정도를 증가했고 test 에 대한 accuracy 도 0.88 에서 0.92 로 증가하며 좋은 결과를 보였다. 또한 랜덤포레스트와 의사결정나무와 함께 test 정확도가 0.92 로 가장 높다는 점에 주목을 할 수 있다.

⑨ Kmeans

```
validation scores:  [0.49984538 0.5059918  0.49369894]
mean of validation score:  0.49984537410873076
[[ 9928 14961]
 [10264 14737]]

              precision    recall  f1-score   support

           0       0.49      0.40      0.44     24889
           1       0.50      0.59      0.54     25001

    accuracy                           0.49     49890
   macro avg       0.49      0.49      0.49     49890
weighted avg       0.49      0.49      0.49     49890
```

Feature selection 과 Kmeans 를 모두 적용한 결과 validation score 는 0.50 으로 feature selection 을 적용하기 전과 비슷한 숫자이다. 또한 test 에 대한 accuracy 는 다른 모델들과 다르게 0.50 에서 0.49 로 오히려 감소를 하였다. 비록 소폭감소이지만 이는 feature selection 이 인접 이웃과의 관계를 통해 학습해나가는 kmeans 의 군집화와는 큰 시너지를 내지 않는다는 것을 알 수 있다.

⑩ Ensemble Learning

위에서 validation 에 대한 결과가 가장 좋았던 모델들로 앙상블 학습을 시행한다. 그 모델들은 다음과 같다.

✧ Decision Tree

첫번째로 Decision Tree 는 앞서 임의로 feature 들을 선택했을 때와는 달리 0.9 를 넘는 test 정확도를 가졌다는 점이 눈에 띈다. 남녀 모두의 precsion, recall, f1-score 이 증가하여 0.9 를 넘어 높은 성능을 보여준다.

✧ Random Forest

Random Forest 같은 경우에는 총 9 개의 모델 중 validation score 가 0.924 로 가장 높게 나왔다. Feature selection 적용 후에 전반적인 성능이 증가하였으며 특히 남자의 precision 과 여자의 recall 이 0.97 에 육박하는 우수함을 보인다.

✧ LightGBM

LightGBM 역시 feature selection 후, validation score 가 0.914 에 도달하는 우수한 성능을 보였다. Feature selection 이전에는 특히 남성의 recall 과 f1-score 이 0.7 점대로 낮은 점수를 보였지만 anova 를 통해서 남녀 모두의 precision, recall, f1-score 가 0.9 를 넘는 개선을 불러왔다.

따라서 이 세가지 모델을 이용하여 앙상블 학습을 시행하면 다음 같은 결과가 나왔다.

```
validation scores:  [0.93871766 0.93887071 0.94031389]
mean of validation score:  0.939300752367446
[[22435  2454]
 [  388 24613]]

              precision    recall  f1-score   support

           남       0.98      0.90      0.94     24889
           여       0.91      0.98      0.95     25001

    accuracy                           0.94     49890
   macro avg       0.95      0.94      0.94     49890
weighted avg       0.95      0.94      0.94     49890
```

이는 앞서 임의로 피처들을 선택했을 때보다 validation 정확도는 0.94 로 증가를 하였다. 또한 test 정확도가 0.03 높은 0.94 를 기록하는 우수성을 보인다. 특히 0.8 점대였던 남성의 recall 과 여성의 precision 을 개선했다는 점이 눈에 띈다.

Kmeans 에서의 prediction label 과 앙상블에서의 prediction 을 비교하는 confusion matrix 을 불러오면 다음과 같다.

```
array([[14649, 15049],
       [ 9786, 10406]], dtype=int64)
```

kmeans 과 앙상블이 모두 남자라고 예상한 데이터는 14649 개, 모두 여자라고 예상한 데이터는 10406 개이며, kmeans 는 여자인데 앙상블은 남자로, 혹은 kmeans 은 남자인데 앙상블은 여자로, 예측이 엇갈린 데이터는 9786 과 15049 개로 나뉘었다.

이렇게 feature selection 을 실행해준 후, validation score 를 기반으로 가장 좋은 좋은 모델은 앙상블 학습에서 decision tree, random forest, lightGBM 을 사용했을 때임을 알 수 있다. 따라서 정확도를 더 높이기 위해 gridSearchCV 를 통하여 가장 좋은 파라미터들의 조합을 찾아준다.

후보들은 다음과 같이 모두 6 개의 속성을 조정해주었다.

{'LGBM__num_leaves': [31, 127],'LGBM__reg_alpha': [0.1, 0.5], 'RF__max_features': ['auto', 'sqrt'],'RF__min_samples_leaf': [2, 4], 'DT__criterion':['gini','entropy'],'DT__max_depth': [3,5]}

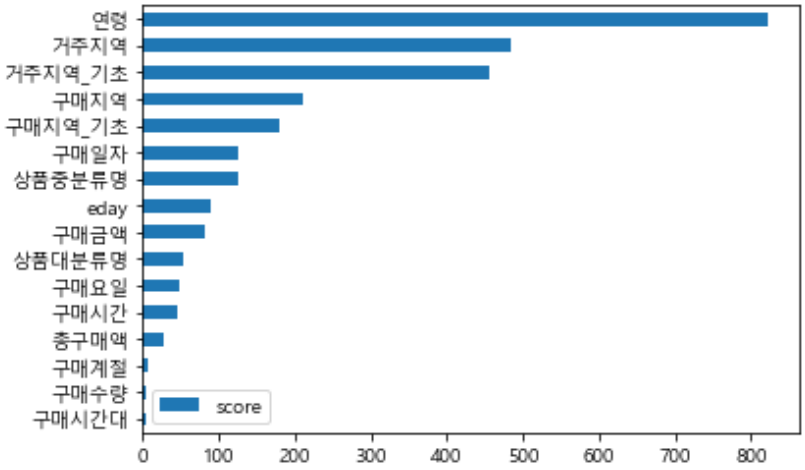
```
validation scores:  [0.95760746 0.9560601  0.95763214]
mean of validation score:  0.9570999011462762
[[23536  1353]
 [  339 24662]]

              precision    recall  f1-score   support

           남       0.99      0.95      0.97     24889
           여       0.95      0.99      0.97     25001

    accuracy                           0.97     49890
   macro avg       0.97      0.97      0.97     49890
weighted avg       0.97      0.97      0.97     49890
```

Best Parameter 는 다음과 같이 나왔다.

```
{'DT__criterion': 'gini', 'DT__max_depth': 3, 'LGBM__num_leaves': 127,
'LGBM__reg_alpha': 0.1, 'RF__max_features': 'auto', 'RF__min_samples_leaf': 2}
```

이때, 앞서 gridSearch 를 시행해주지 않았을 때보다 validation score 가 0.02 상승했음을 알 수 있다. Test 에 대한 accuracy 는 0.97 로 앞서 진행한 모든 모델들과 확연한 차이를 띈다는 점에서 분류를 성공적으로 마쳤다는 것을 알 수 있다.

또한 XGBoost 를 사용하여 피처들을 따로 선택할 수 있다. XGB 의 feature_importance 속성을 사용하면 다음과 같이 중요한 피처 순서대로 정렬을 할 수 있다.



XGBoost 에 따르면, 연령, 거주지역, 거주지역_기초, 구매지역, 구매지역_기초, 구매일자, 상품중분류명, eday, 구매금액, 상품대분류명 순으로 10 개를 선택할 수 있다. 이는 우리가 처음에 선택한 피처들과 몇 개 차이는 있지만 전반적인 맥락에서는 일치함으로 보아 적절한 모델을 선택하여 성별을 예측했음을 알 수 있다.

7. 최종 결론 및 한계점

우리는 대한민국 마트 소비에 대한 데이터를 갖고 데이터의 다양한 측면에 대해 EDA 를 통해 탐구할 수 있었고 성별 예측이라는 주제로 총 10 개의 모델을 수행해보았다. 이때, 한 모델당 세가지 경우를 모두 수행해보았는데 1) 원데이터에서 임의로 피처를 골라낸 자료, 2) SMOTE 를 적용하여 임의로 피처를 골라낸 자료, 3) SMOTE 를 적용하여 feature selection(ANOVA 사용)으로 피처를 골라낸 자료로 모두 수행을 하여 최종 모델에 대한 gridSearch 까지 총 31 가지의 모델을 적용하였다. 이때 전반적으로 1) 방법에 비해 SMOTE 를 적용한 2) 방법은 성능 개선에 도움이 되었지만 SMOTE 를 적용한다고 무조건 정확도가 높아지는 것이 아니라는 것을 알았다.

반면 feature selection 을 적용한 3) 방법은 2)방법에 비해 Kmeans 를 제외한 8 개의 모델에서 모두 개선된 성능을 보였다. 이로 통해 우리는 데이터 분석을 시행 할 시, 분석자 임의로 피처를 뽑아내기보다 여러가지 feature selection 기법을 사용하여 먼저 가장 좋은 방법을 찾아내야한다는 근거를 얻을 수 있다.

또한 validation 과 test set 에 대한 accuracy 들을 비교한 결과 두 수치 사이에 큰 차이를 보인 모델은 없었다. 따라서 우리는 31 개의 모델 중, 과적합(overfitting)이 일어난 모델은 없다고 결론지을 수 있다.

이렇게 31 개의 모델을 통해 최종적으로 성능이 가장 좋았던 모델은 바로 SMOTE 를 적용하여 자료불균형 문제를 해결하고, feature selection 과 gridSearch 를 적용하여 decision tree, random forest 와 lightGBM 을 합친 앙상블 기법임을 알 수 있다. 이는 최종 성별 예측에 있어 97%라는 높은 정확도를 기록하였지만 추가적인 모델이나 gridSearch 에서 후보 파라미터의 개수나 종류를 다양화하여 성능을 더 높일 수 있다.

해당 데이터 분석 프로젝트를 통하여, 그동안 배운 내용을 실습하며 실제 raw 데이터에 다양한 기법들을 적용하고 데이터 처리하는 방법을 익힐 수 있다는 데 의의가 있다.

반면, 해당 프로젝트가 갖는 한계점은 주제에 있어서 목적이 뚜렷하지 않다는 것이다. 실제 분석에 있어서, 단순히 성별을 예측하는 것만으로 큰 의미를 갖기 힘들다. 하지만 이는 성별 예측과 함께 상품중분류명이나 구매요일을 연관시킨다면 마트 측에서 이를 마케팅 전략으로 사용함으로써 더 많은 수익을 창출하고 한계점을 극복할 수 있다.