Alchemist 3기 아이디어톤

df_pig 조

1. 데이터 확인

1 pd.read_csv로 데이터를 불러오고 head, info, describe()를 통해 데이터의 형태를 확인

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC
0	Female	21.000000	1.620000	64.000000	yes	no	2.0	3.0	Sometimes
1	Female	21.000000	1.520000	56.000000	yes	no	3.0	3.0	Sometimes
2	Male	23.000000	1.800000	77.000000	yes	no	2.0	3.0	Sometimes
3	Male	27.000000	1.800000	87.000000	no	no	3.0	3.0	Sometimes
4	Male	22.000000	1.780000	89.800000	no	no	2.0	1.0	Sometimes
...	...	...	...	...	...	...	...	...	...
2106	Female	20.976842	1.710730	131.408528	yes	yes	3.0	3.0	Sometimes
2107	Female	21.982942	1.748584	133.742943	yes	yes	3.0	3.0	Sometimes
2108	Female	22.524036	1.752206	133.689352	yes	yes	3.0	3.0	Sometimes
2109	Female	24.361936	1.739450	133.346641	yes	yes	3.0	3.0	Sometimes
2110	Female	23.664709	1.738836	133.472641	yes	yes	3.0	3.0	Sometimes

2111 rows × 17 columns

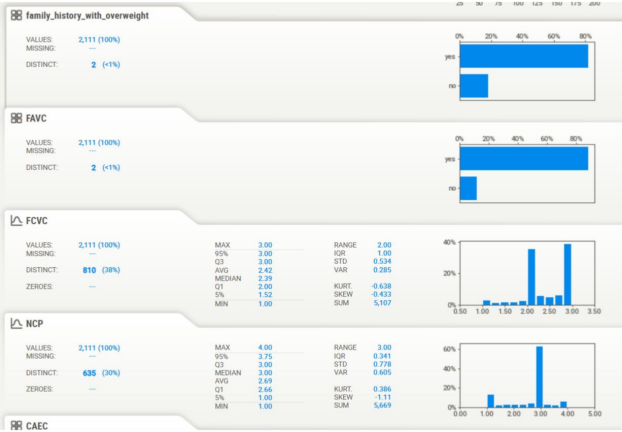
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Gender                          2111 non-null   object 
 1   Age                             2111 non-null   float64
 2   Height                          2111 non-null   float64
 3   Weight                          2111 non-null   float64
 4   family_history_with_overweight  2111 non-null   object 
 5   FAVC                            2111 non-null   object 
 6   FCVC                            2111 non-null   float64
 7   NCP                             2111 non-null   float64
 8   CAEC                            2111 non-null   object 
 9   SMOKE                           2111 non-null   object 
 10  CH2O                            2111 non-null   float64
 11  SCC                             2111 non-null   object 
 12  FAF                             2111 non-null   float64
 13  TUE                             2111 non-null   float64
 14  CALC                            2111 non-null   object 
 15  MTRANS                          2111 non-null   object 
 16  NObeyesdad                      2111 non-null   object 
dtypes: float64(8), object(9)
```

```
pig.describe()
```

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
count	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000
mean	24.312600	1.701677	86.586058	2.419043	2.685628	2.008011	1.010298	0.657866
std	6.345968	0.093305	26.191172	0.533927	0.778039	0.612953	0.850592	0.608927
min	14.000000	1.450000	39.000000	1.000000	1.000000	1.000000	0.000000	0.000000
25%	19.947192	1.630000	65.473343	2.000000	2.658738	1.584812	0.124505	0.000000
50%	22.777890	1.700499	83.000000	2.385502	3.000000	2.000000	1.000000	0.625350
75%	26.000000	1.768464	107.430682	3.000000	3.000000	2.477420	1.666678	1.000000
max	61.000000	1.980000	173.000000	3.000000	4.000000	3.000000	3.000000	2.000000

2 데이터에서 null값은 존재하지 않고 일부 칼럼이 비정형 데이터임을 확인

3 sweetviz를 통한 데이터 시각화



4 smote로 인한 문제 확인

```
pig[(['Age', 'FCVC', 'NCP', 'TUE'])].value_counts()
```

```
Age        FCVC  NCP       TUE
21.000000  2.0   1.000000  0.000000    18
18.000000  2.0   3.000000  0.000000    16
23.000000  2.0   3.000000  1.000000    15
21.000000  2.0   3.000000  0.000000    13
                           1.000000    12
                                       ..
21.282238  3.0   3.000000  0.849236     1
21.274628  3.0   3.489918  0.128394     1
21.243142  3.0   1.726260  0.000000     1
21.238416  3.0   3.000000  0.890527     1
61.000000  3.0   3.000000  1.000000     1
Name: count, Length: 1784, dtype: int64
```

2. 데이터 전처리

1. 실수형 데이터 스케일링 (StandardScaler)

```
#실수형 데이터의 피처 세개 정규화
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
float_list=['Height','Weight','CH2O']
#타겟값 인코딩
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
pig['NObeyesdad'] = label_encoder.fit_transform(pig['NObeyesdad'])
#실수형 데이터만 선택한 데이터프레임
pig_float=pig[float_list]
```

데이터 전처리

2. smote로 인해 변환된 함수 round() 적용

```
#smote로 인해 변환된 함수 반올림 적용.
pig.Age = pig.Age.round(); pig.FCVC = pig.FCVC.round(); pig.NCP = pig.NCP.round(); pig.TUE = pig.TUE.round()
```

3. 피처와 타겟 구분

```
#피처와 타겟 분리
X_features = pig.drop(['NObeyesdad'], axis=1, inplace=False)
y_target = pig['NObeyesdad']
```

4. 비정형 데이터에 원핫인코딩 적용

```
X_features_ohe = pd.get_dummies(X_features, columns=['Gender','family_history_with_overweight','FAVC','CAEC','SMOKE','SCC','CALC','MTRANS'], dtype=int)
X_features_ohe
```

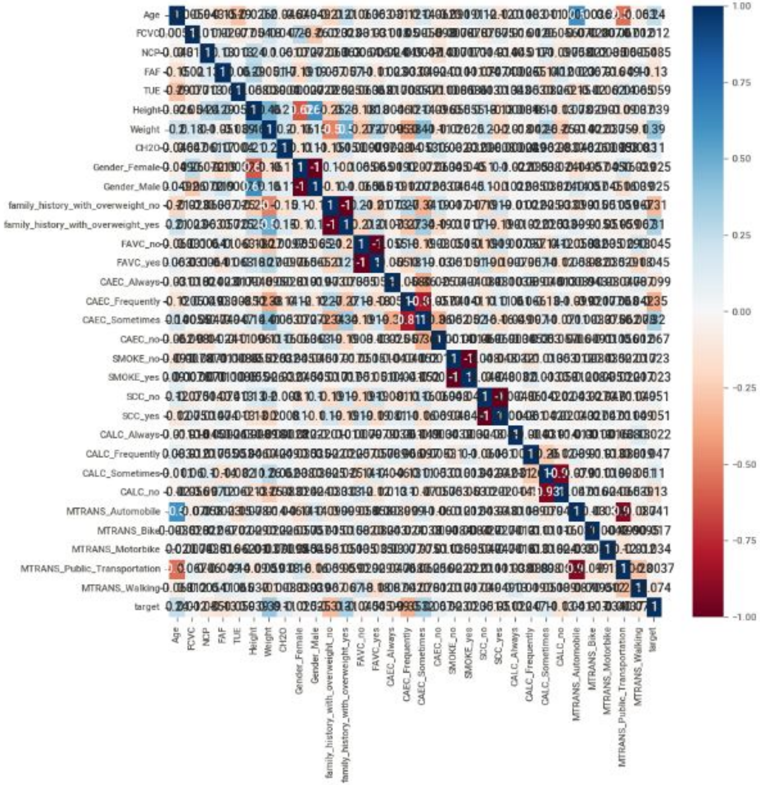
데이터 전처리

	Age	FCVC	NCP	FAF	TUE	Height	Weight	CH2O	Gender_Female	Gender_Male	...	SCC_yes	CALC_Always	CALC_Frequently
0	21.0	2.0	3.0	0.000000	1.0	-0.875589	-0.862558	-0.013073	1	0	...	0	0	0
1	21.0	3.0	3.0	3.000000	0.0	-1.947599	-1.168077	1.618759	1	0	...	1	0	0
2	23.0	2.0	3.0	2.000000	1.0	1.054029	-0.366090	-0.013073	0	1	...	0	0	1
3	27.0	3.0	3.0	2.000000	0.0	1.054029	0.015808	-0.013073	0	1	...	0	0	1
4	22.0	2.0	1.0	0.000000	0.0	0.839627	0.122740	-0.013073	0	1	...	0	0	0

2111 rows × 31 columns

상관도 분석

```
target                               1.000000
Weight                               0.387643
CAEC_Sometimes                       0.316962
family_history_with_overweight_yes   0.313667
Age                                  0.235660
CALC_Sometimes                       0.114104
CH2O                                 0.108868
CAEC_no                              0.066715
SCC_no                               0.050679
CALC_Frequently                      0.047318
FAVC_yes                             0.044582
MTRANS_Automobile                    0.041170
Height                               0.038986
Gender_Male                          0.024908
SMOKE_no                             0.023256
FCVC                                 0.012068
MTRANS_Public_Transportation        -0.003748
MTRANS_Bike                         -0.017351
CALC_Always                         -0.022484
SMOKE_yes                           -0.023256
Gender_Female                       -0.024908
MTRANS_Motorbike                    -0.034293
FAVC_no                             -0.044582
SCC_yes                             -0.050679
TUE                                 -0.059050
MTRANS_Walking                      -0.073823
NCP                                 -0.085367
CAEC_Always                         -0.099028
FAF                                 -0.129564
CALC_no                             -0.134716
family_history_with_overweight_no   -0.313667
CAEC_Frequently                     -0.351827
Name: target, dtype: float64
```



소프트 보팅과 하드 보팅을 통한 학습

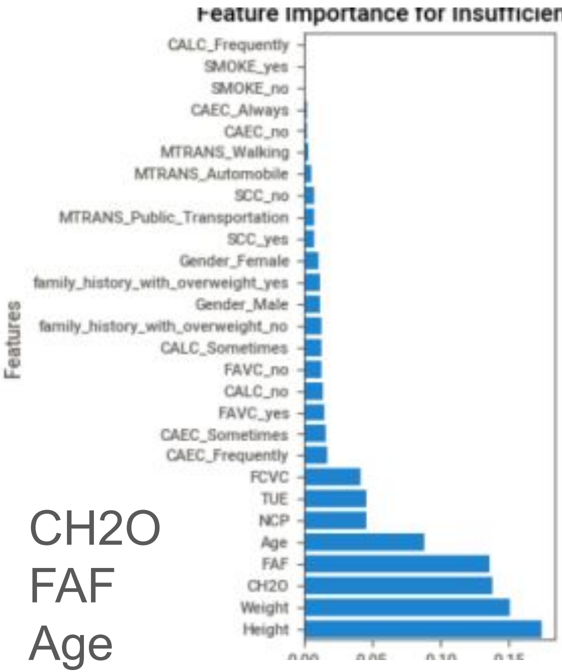
```
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# 데이터 준비
X = X_features_ohe  # 피처
y = pig['NObeyesdad']  # 타겟

# 훈련 데이터와 테스트 데이터 분리
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 개별 모델 정의
dt_clf = DecisionTreeClassifier(random_state=156)
log_reg = LogisticRegression(max_iter=1000, random_state=156)
rf_clf = RandomForestClassifier(random_state=156, max_depth=20, max_features='sqrt', min_samples_leaf=1, min_samples_split=5, n_estimators=200
gb_model = GradientBoostingClassifier(learning_rate= 0.05, max_depth= 3, min_samples_split= 2, n_estimators= 200, subsample= 0.8, random_state
# 하드 보팅 분류기
hard_voting_clf = VotingClassifier(estimators=[
    ('log_reg', log_reg),
    ('decision_tree', dt_clf),
    ('rf_clf', rf_clf),
    ('gb_model',gb_model)
], voting='hard')  # 하드 보팅

# 소프트 보팅 분류기
soft_voting_clf = VotingClassifier(estimators=[
    ('log_reg', log_reg),
    ('decision_tree', dt_clf),
    ('rf_clf', rf_clf),
    ('gb_model',gb_model)
], voting='soft')  # 소프트 보팅
```

모델 훈련, 예측, 평가

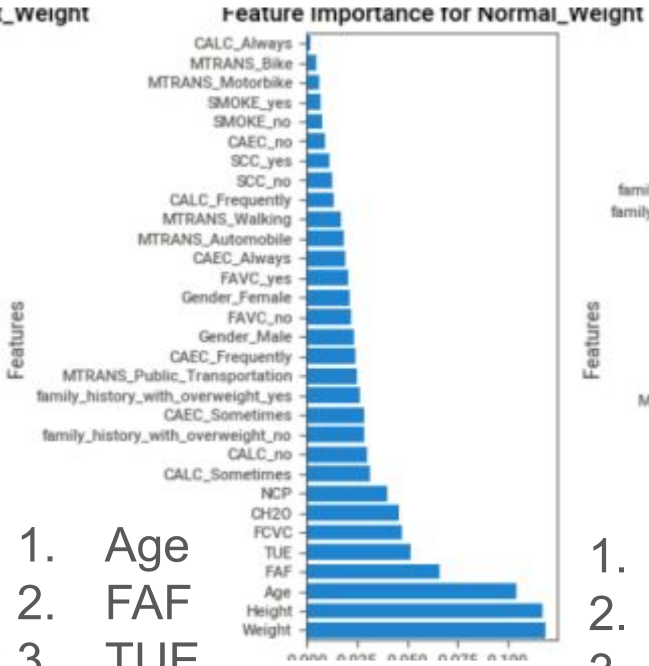
```python
# 모델 훈련
hard_voting_clf.fit(X_train, y_train)
soft_voting_clf.fit(X_train, y_train)

# 예측
y_pred_hard = hard_voting_clf.predict(X_test)
y_pred_soft = soft_voting_clf.predict(X_test)

# 정확도 평가
accuracy_hard = accuracy_score(y_test, y_pred_hard)
accuracy_soft = accuracy_score(y_test, y_pred_soft)

print(f"Hard Voting Accuracy: {accuracy_hard:.4f}")
print(f"Soft Voting Accuracy: {accuracy_soft:.4f}")
```

```
Hard Voting Accuracy: 0.9598
Soft Voting Accuracy: 0.9669
```

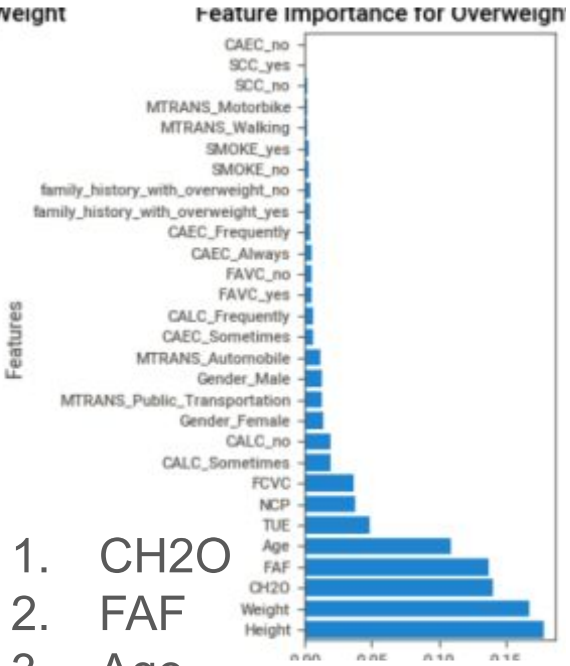
저체중 그룹, 정상체중 그룹, 과체중 1 그룹



1. CH2O
2. FAF
3. Age
4. NCP
5. TUE

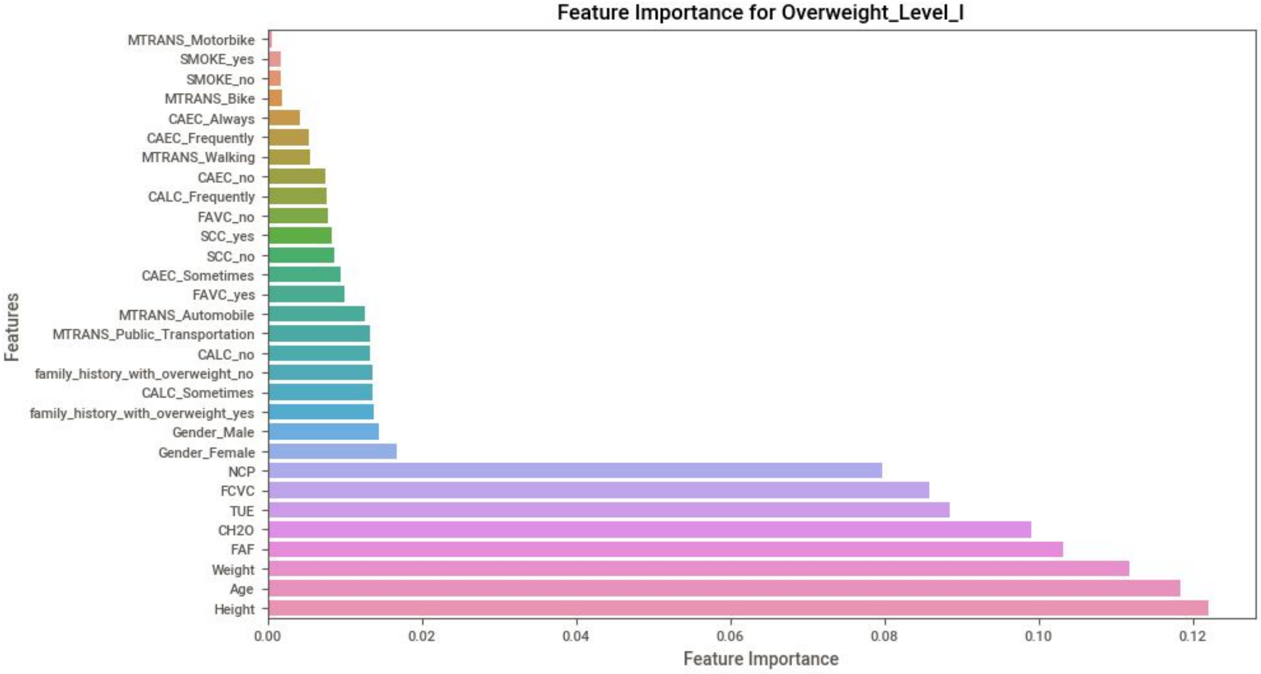


1. Age
2. FAF
3. TUE
4. FCVC
5. CH2O



1. CH2O
2. FAF
3. Age
4. Tue
5. NCP

과체중 그룹-Overweight_Level_I

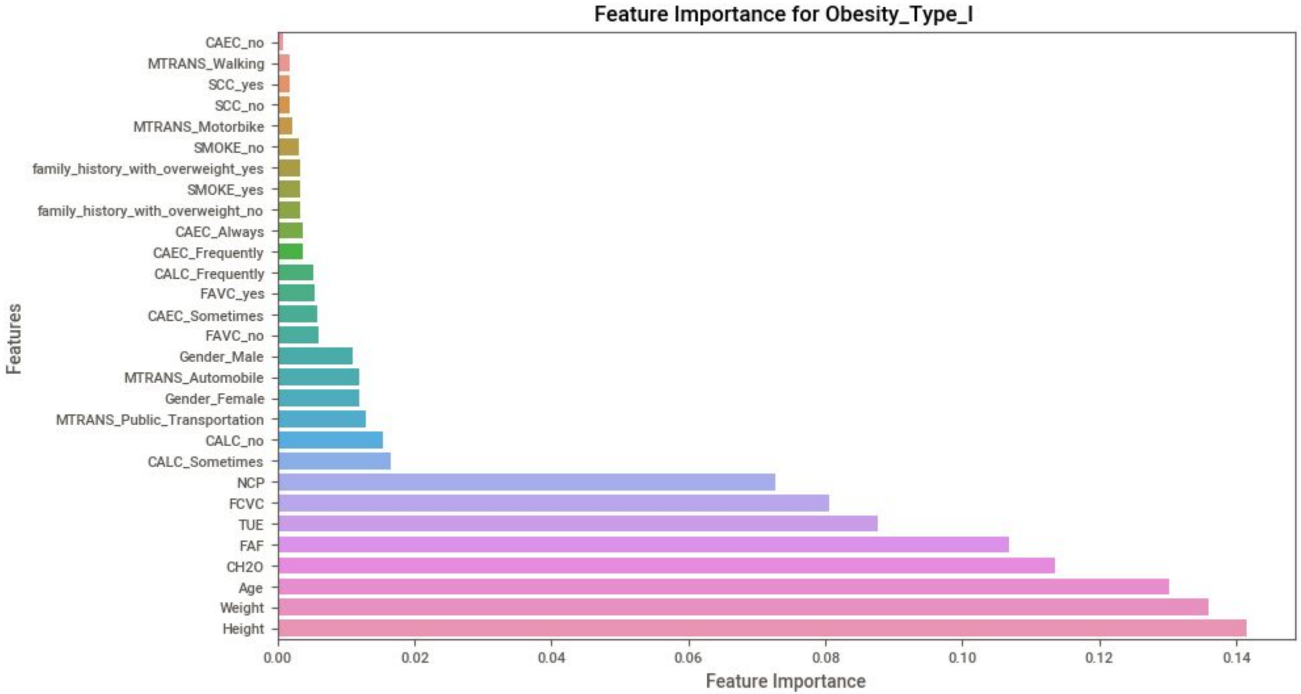


1. FAF
2. CH2O
3. TUE
4. FCVC
5. NCP

중요 인자

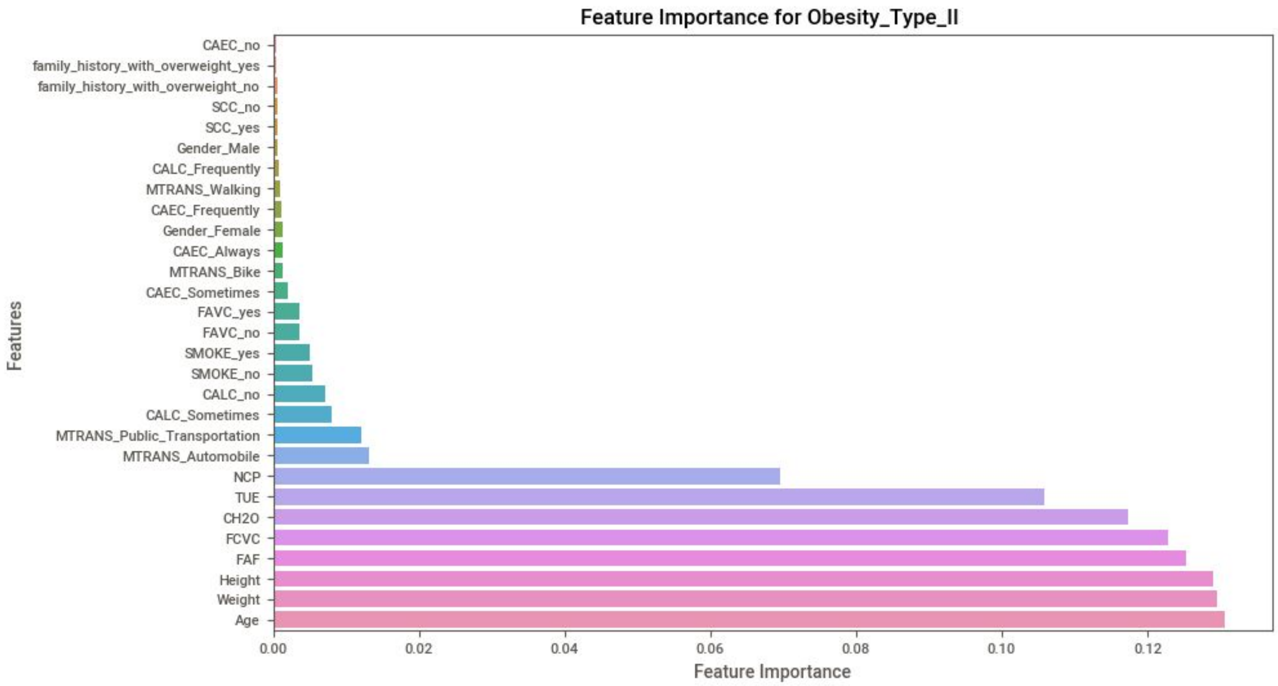
- FAF: 운동량
- CH2O: 물 섭취량
- TUE: 전자기기 사용
- FCVC: 채소섭취량
- NCP: 하루에 먹는 끼니 수

비만 그룹-Obesity_Type_I



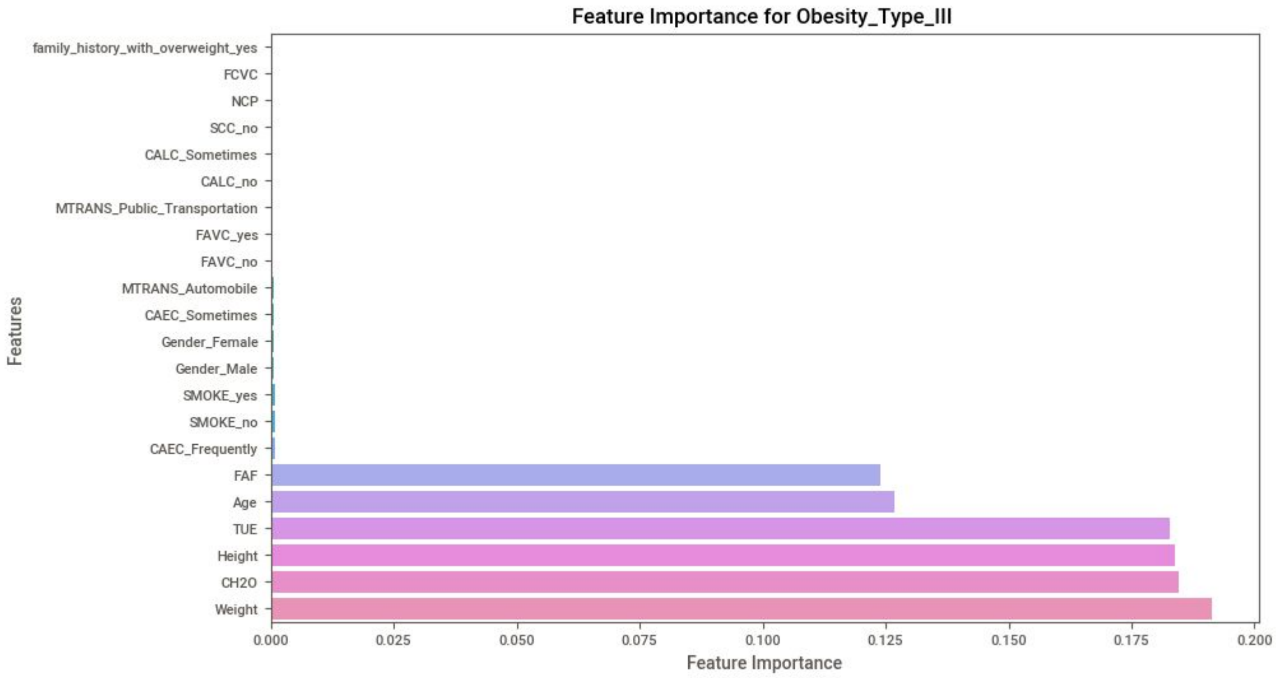
1. CH2O
2. FAF
3. TUE
4. FCVC
5. NCP

비만 그룹-Obesity_Type_II



1. FAF
2. FCVC
3. CH2O
4. TUE
5. NCP

비만 그룹-Obesity_Type_III



1. CH2O
2. TUE
3. FAF