랜덤포레스트 분류

#01. 패키지 참조하기

```
import warnings
warnings.filterwarnings('ignore')

from matplotlib import pyplot as plt
import seaborn as sb
from pandas import read_excel, DataFrame, melt
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score, cross_validate
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import GridSearchCV
from imblearn.over_sampling import SMOTE

from sklearnex import patch_sklearn
from daal4py.oneapi import sycl_context
patch_sklearn()
```

```
Intel(R) Extension for Scikit-learn* enabled (https://github.com/intel/scikit-learn-in
```

#02. 데이터 가져오기

필드명	설명
id	고유 식별 번호
age	나이
height(cm)	키
weight(kg)	몸무게
waist(cm)	허리둘레
eyesight(left)	시력(왼쪽)
eyesight(right)	시력(오른쪽)
hearing(left)	청력(왼쪽)
hearing(right)	청력(오른쪽)
systolic	수축기 혈압(mmHg 단위)
relaxation	휴식 혈압(mmHg 단위)
fasting blood sugar	공복 혈당 수치(mg/dL 단위)

필드명	설명
Cholesterol	콜레스테롤 수치(mg/dL 단위)
triglyceride	중성지방 수치(mg/dL 단위)
HDL	고밀도 지단백 수치 (mg/dL)
LDL	저밀도 지단백 수치 (mg/dL)
hemoglobin	헤모글로빈 수치(g/dL)
Urine protein	소변내 단백질 수준
serum creatinine	혈청 크레아티닌 수치(mg/dL)
AST	아스파르트 아미노전이효소(AST) 수준
ALT	알라닌아미노 전이효소 수준
Gtp	감마-글루타밀 전이효소 수준
dental caries	1인당 치아우식증 유무를 나타내는 값(0: 없음, 1: 있음)
smoking	흡연상태(0: 비흡연자, 1: 흡연자)

```
origin = read_excel("https://data.hossam.kr/G02/smoker_status.xlsx")
origin.head().T
```

	0	1	2	3	4
age	55.0	70.0	20.0	35.0	30.0
height(cm)	165.0	165.0	170.0	180.0	165.0
weight(kg)	60.0	65.0	75.0	95.0	60.0
waist(cm)	81.0	89.0	81.0	105.0	80.5
eyesight(left)	0.5	0.6	0.4	1.5	1.5
eyesight(right)	0.6	0.7	0.5	1.2	1.0
hearing(left)	1.0	2.0	1.0	1.0	1.0
hearing(right)	1.0	2.0	1.0	1.0	1.0
systolic	135.0	146.0	118.0	131.0	121.0
relaxation	87.0	83.0	75.0	88.0	76.0
fasting blood sugar	94.0	147.0	79.0	91.0	91.0
Cholesterol	172.0	194.0	178.0	180.0	155.0
triglyceride	300.0	55.0	197.0	203.0	87.0
HDL	40.0	57.0	45.0	38.0	44.0
LDL	75.0	126.0	93.0	102.0	93.0
hemoglobin	16.5	16.2	17.4	15.9	15.4
Urine protein	1.0	1.0	1.0	1.0	1.0
serum creatinine	1.0	1.1	0.8	1.0	0.8

	0	1	2	3	4
AST	22.0	27.0	27.0	20.0	19.0
ALT	25.0	23.0	31.0	27.0	13.0
Gtp	27.0	37.0	53.0	30.0	17.0
dental caries	0.0	1.0	0.0	1.0	0.0
smoking	1.0	0.0	1.0	0.0	1.0

#03. 데이터 전처리

데이터 확인

```
print(origin.shape)
```

```
(159256, 23)
```

```
origin.isnull().any()
```

```
age                    False
height(cm)             False
weight(kg)             False
waist(cm)              False
eyesight(left)         False
eyesight(right)        False
hearing(left)          False
hearing(right)         False
systolic               False
relaxation             False
fasting blood sugar    False
Cholesterol            False
triglyceride           False
HDL                    False
LDL                    False
hemoglobin             False
Urine protein          False
serum creatinine       False
AST                    False
ALT                    False
Gtp                    False
dental caries          False
smoking                False
dtype: bool
```

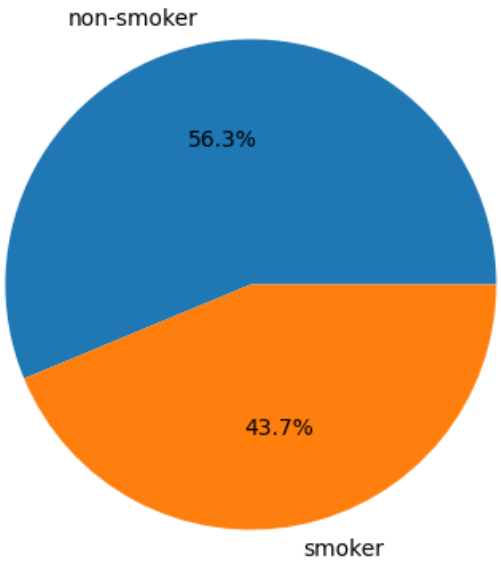
```
origin.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159256 entries, 0 to 159255
```

```
Data columns (total 23 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   age                  159256 non-null  int64
 1   height(cm)           159256 non-null  int64
 2   weight(kg)           159256 non-null  int64
 3   waist(cm)            159256 non-null  float64
 4   eyesight(left)       159256 non-null  float64
 5   eyesight(right)      159256 non-null  float64
 6   hearing(left)        159256 non-null  int64
 7   hearing(right)       159256 non-null  int64
 8   systolic             159256 non-null  int64
 9   relaxation           159256 non-null  int64
 10  fasting blood sugar  159256 non-null  int64
 11  Cholesterol          159256 non-null  int64
 12  triglyceride         159256 non-null  int64
 13  HDL                  159256 non-null  int64
 14  LDL                  159256 non-null  int64
 15  hemoglobin           159256 non-null  float64
 16  Urine protein        159256 non-null  int64
 17  serum creatinine     159256 non-null  float64
 18  AST                  159256 non-null  int64
 19  ALT                  159256 non-null  int64
 20  Gtp                  159256 non-null  int64
 21  dental caries        159256 non-null  int64
 22  smoking              159256 non-null  int64
dtypes: float64(5), int64(18)
memory usage: 27.9 MB
```

목적변수 비율 확인

```
plt.figure(figsize=(5, 5))
plt.pie(origin['smoking'].value_counts(), labels=['non-smoker', 'smoker'],
autopct='%1.1f%%')
plt.show()
plt.close()
```



변수값 분리

```
x = origin.drop(['smoking'], axis=1)
y = origin['smoking']
x.shape, y.shape
```

```
((159256, 22), (159256,))
```

훈련, 검증 데이터 분할

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state = 123)
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

```
((119442, 22), (39814, 22), (119442,), (39814,))
```

데이터 불균형 해소

```
smote_sampler = SMOTE(sampling_strategy="minority", random_state=777)
x_sm, y_sm = smote_sampler.fit_resample(x_train, y_train)
print(x_sm.shape, y_sm.shape)

y_sm.value_counts().sort_index()
```

```
(134520, 22) (134520,)
```

```
0    67260
1    67260
Name: smoking, dtype: int64
```

#04. 훈련모델 적합

RandomForestClassifier 하이퍼파라미터

파라미터	설명
n_estimators	결정트리의 갯수를 지정 (기본값=10), 성능에 비례, 속도에 반비례
min_samples_split	노드를 분할하기 위한 최소한의 샘플 데이터 수. 과적합을 제어하는데 사용(기본값=2), 값이 작을 수록 분할 노드가 증가하여 과적합 가능성이 높아짐
min_samples_leaf	리프노드가 되기 위한 최소한의 샘플 데이터 수. 과적합을 제어하는데 사용
max_features	최적의 분할을 위해 고려할 최대 feature 개수(기본값=auto) int형일 경우 갯수, float형일 경우 비율, auto일 경우 전체 feature 만큼 선정
max_depth	트리의 최대 깊이(기본값=None). `max_depth`가 `None`일 경우 완벽하게 클래스 값이 결정되거나 데이터 개수가

파라미터	설명
	`min_samples_split` 에서 설정한 값보다 작아질 때 까지 분할
max_leaf_nodes	리프노드의 최대 개수

```
rfc = RandomForestClassifier(random_state=777)

params = {
    'n_estimators': [20, 50, 100],
    'max_depth': [5, 30, 100],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 5, 10],
}

grid = GridSearchCV(rfc, param_grid=params, cv=5, n_jobs=-1)
grid.fit(x_sm, y_sm)

print("최적의 하이퍼 파라미터: ", grid.best_params_)
print("최적의 모델 평균 성능(훈련데이터): ", grid.best_score_)

best_model = grid.best_estimator_
y_pred = best_model.predict(x_test)
print("최종 모델의 성능(테스트 데이터): ", accuracy_score(y_test, y_pred))
```

분류 보고서

```
              precision    recall  f1-score   support

           0       0.83      0.75      0.79     22343
           1       0.71      0.80      0.76     17471

    accuracy                           0.77     39814
   macro avg       0.77      0.78      0.77     39814
weighted avg       0.78      0.77      0.77     39814
```

precision : 정밀도, 양성 클래스라고 예측한 샘플 중 실제로 양성 클래스에 속하는 샘플 수의 비율

recall : 재현율, 실제로 양성 클래스에 속한 샘플 중에서 양성 클래스라고 예측한 샘플 수의 비율

f1-score : 정밀도와 재현율의 가중 조화평균값

support : 각 Label에 대한 실제 샘플 수

accuracy : 정확도. 전체 샘플 중 맞게 예측한 샘플 수의 비율

macro avg : 단순 평균값(샘플 수의 불균형을 고려하지 않는 값)

weighted avg : 각 클래스에 속하는 표본의 개수로 가중 평균을 낸 값(샘플 수의 불균형을 고려한 값)

```
print(classification_report(y_test, y_pred))
```