

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_validate
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier

wine = pd.read_csv('https://bit.ly/wine_csv_data')

data = wine[['alcohol', 'sugar', 'pH']].to_numpy()
target = wine['class'].to_numpy()

train_input, test_input, train_target, test_target = train_test_split(
    data, target, test_size=0.2, random_state=42
)

ss = StandardScaler()
ss.fit(train_input)
train_scaled = ss.transform(train_input)
test_scaled = ss.transform(test_input)
dt = DecisionTreeClassifier(random_state=42)

scores = cross_validate(dt, train_scaled, train_target)

print(f"기본 모델 교차 검증 평균 점수: {np.mean(scores['test_score']):.4f}\n")

```

기본 모델 교차 검증 평균 점수: 0.8566

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_validate, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier

wine = pd.read_csv('https://bit.ly/wine_csv_data')

data = wine[['alcohol', 'sugar', 'pH']].to_numpy()
target = wine['class'].to_numpy()

train_input, test_input, train_target, test_target = train_test_split(
    data, target, test_size=0.2, random_state=42
)

ss = StandardScaler()
ss.fit(train_input)
train_scaled = ss.transform(train_input)
test_scaled = ss.transform(test_input)

dt = DecisionTreeClassifier(random_state=42)
scores = cross_validate(dt, train_scaled, train_target)

params = {
    'max_depth': [5, 6, 7, 8, 9, 10],
    'min_samples_split': [2, 5, 10, 20],
    'min_samples_leaf': [1, 2, 4, 8]
}

gs = GridSearchCV(DecisionTreeClassifier(random_state=42),
                  params,
                  n_jobs=-1)

gs.fit(train_scaled, train_target)

best_cv_score = np.max(gs.cv_results_['mean_test_score'])

best_dt = gs.best_estimator_
test_score = best_dt.score(test_scaled, test_target)

print(f"하이퍼파라미터: {gs.best_params_}\n")
print(f"튜닝 후 최고 교차 검증 평균 점수: {best_cv_score:.4f}\n")
print(f"최종 모델 테스트 세트 점수: {test_score:.4f}\n")

하이퍼파라미터: {'max_depth': 8, 'min_samples_leaf': 8, 'min_samples_split': 20}
튜닝 후 최고 교차 검증 평균 점수: 0.8640
최종 모델 테스트 세트 점수: 0.8608

```

## 1. 수정 전 코드 (기준점 확인)

한 일: 결정 트리 모델을 '기본 설정(Default)'으로 딱 1개 만든다.

목적: 이 기본 모델의 성능(예: 85점)이 얼마인지 측정한다.

의미: 튜닝(최적화)을 하기 전의 '기준 점수'를 확인하는 과정이다.

## 2. 수정 후 코드 (최적화)

한 일: max\_depth 같은 하이퍼파라미터 값의 조합을 수십 개 만든다.

목적: 그리드 서치 기능을 이용해, 이 조합들을 모두 테스트하고 그중 가장 최적화된 성능의 조합을 찾아낸다.

의미: 모델이 낼 수 있는 최고 점수(88점)을 찾는 과정이다.

## 3. 핵심 비교

기준 점수(85점)와 튜닝 후 최고 점수(88점)를 비교한다.

점수가 올랐다면, 우리가 시간을 들여 하이퍼파라미터를 튜닝한 노력이 성공적이었음을 의미한다.

이 과정은 모델이 훈련 데이터에만 과대적합되는 것을 막고, 실제 테스트에서 더 잘 작동하도록(일반화 성능을 높이도록) 만드는 핵심 작업이다.