

3) 학습 데이터와 테스트 데이터 분리

```
X_train, X_test, y_train, y_test = train_test_split(iris_data,
iris_label, test_size=0.2, random_state=11)
```

4) 의사결정트리로 학습과 예측 수행

```
# DecisionTreeClassifier 객체 생성
dt_clf = DecisionTreeClassifier(random_state=11)
```

```
# 학습 수행
dt_clf.fit(X_train, y_train)
```

```
DecisionTreeClassifier(class_weight=None, criterion='gini',
max_depth=None,
                        max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort=False,
                        random_state=11, splitter='best')
```

5) 학습된 모델 기반에서 테스트 데이터로 예측 수행

```
# 학습이 완료된 DecisionTreeClassifier 객체에서 테스트 데이터
세트에 예측 수행.
pred = dt_clf.predict(X_test)
```

6) DecisionTreeClassifier의 예측 성능 평가

```
from sklearn.metrics import accuracy_score
print('예측 정확도: {0:.4f}'.format(accuracy_score(y_test, pred)))
```

예측 정확도: 0.9333

- 정확도 : 예측 결과가 실제 레이블 값과 얼마나 정확하게 맞는지를 평가하는 지표

* 붓꽃 데이터 세트 예측 프로세스

