프레임워크 API 제공 알고리즘과 편리한 다양한 - 파이썬 머신러닝을 위한

scikit-learn

Machine Learning in Python

 Open source, commercially usable - BSD license Built on NumPy, SciPy, and matplotlib contexts GitHub Getting Started Release Highlights for 0.23

또는 pip install scikit-learn 설末 : conda install scikit-learn

print(sklearn. version import sklearn

0.21.3

예측하기 배 Kho 데이터 이용하여 붓꽃(Iris) 2. 사이킷런을

<u> 피</u>쳐를 꽃받침의 길이와 너비 너비, 세트는 꽃잎의 길이와 품종 예측 붗꽃 데이터 기반으로 꽃의

표 데이터 자

름이름

데이터

싸



ris virginica iris versicolor 배 Kho



sepal length - petal length sepal width - petal width

류 : <u>지도학습</u> 방법의 하나 山

됐 ↑ 데이터로 황 정 ◆ 때스트 입사 盂 데이터로 피처의 하라

집이匠

메시트

데이터,

꺆

데이터(피쳐, 레이블) →

sklearn.model selection import train test split from sklearn.tree import DecisionTreeClassifier from sklearn.datasets import load iris from

Accessible to everybody, and reusable in various

Simple and efficient tools for predictive data

analysis

рд import pandas as 로딩합니다 데이터 세트를

iris = load_iris()

데이더를 딦 세트에서 피처(feature)만으로 # iris.data는 Iris 데이터 numpy로 가지고 있습니다.

iris_data = iris.data

데이터를 跃) # iris.target은 붓꽃 데이터 세트에서 레이블(결정 numpy로 가지고 있습니다.

iris label = iris.target

print('iris target弘:', iris_label)

print('iris tanget閚:', iris.tanget_names)

변환합니다. # 붓꽃 데이터 세트를 자세히 보기 위해 DataFrame으로 iris_df = pd.DataFrame(data=iris_data, columns=iris.feature_names)

iris_df['label'] = iris.target iris_df.head(3) 0000000000 $- \approx \approx$

iris target号:['setosa' 'versicolor' 'virginica']

	sepal length (cm)	sepal width (cm)	sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) label	petal width (cm)	label
0	5.1	3.5	1.4	0.2	0
~	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0

파 데이터 3)학습 데이터와 테스트

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

사 統 황 학습 라다 4)의사결정트리로

dt_clf = DecisionTreeClassifier(random_state=11) # DecisionTreeClassifier 객체 생성 dt_clf.fit(X_train, y_train) 하라 사행 #

min_impurity_decrease=0.0, min_impurity_split=None, DecisionTreeClassifier(class_weight=None, criterion='gini', min_samples_leaf=1, min_samples_split=2, max_features=None, max_leaf_nodes=None, max_depth=None,

min_weight_fraction_leaf=0.0, presort=False,

random_state=11,splitter='best')

예측 수행 데이터로 모델 기반에서 테스트 5)학습된

집이단 # 학습이 완료된 DecisionTreeClassifier 객체에서 테스트 세트로 예측 수행. pred = dt_clf.predict(X_test)

6)DecisionTreeClassifer의 예측 성능 평가

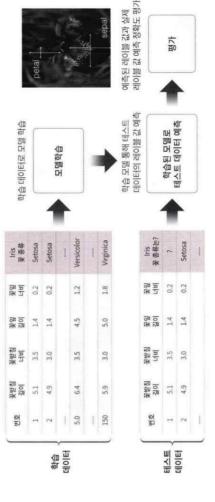
from sklearn.metrics import accuracy_score print('예측 정확도:{0:.4f}'.format(accuracy_score(y_test,pred)))

0.9333 정확도: 뺭

맞는지를 얼마나 정확하게 값과 메이를 실제 결과가 ·· 쌍 표 <u>정</u> 약 다 평가하는

끄로세스 뺭 포 집이匠 싸

학습 데이터를 기반으로 ML 알고리즘을 적용해 모델을 학습시킵니다 데이터를 학습 데이터와 테스트 데이터로 분리합니다 데이터 세트 분리 모델하습 학습된 ML 모델을 이용해 테스트 데이터의 분류(즉, 붓꽃 종류)를 예측합니다 이렇게 예측된 결괏값과 테스트 데이터의 실제 결괏값을 비교해 ML 모델 성능을 평가합니다. 예측 수행 평가



(붓꽃 데이터 세트 기반의 ML 분류 예측 수행 프로세스)

昭가