**기계학습 과제1 보고서**

16010980 이우석

1. 필요 라이브러리

텍스트이(가) 표시된 사진

자동 생성된 설명

* 필요한 라이브러리와 함수들을 import.
* load\_iris : 인공지능 모델을 학습시키기 위해 필요한 데이터셋.
* train\_test\_split : 데이터셋을 train 데이터와 test 데이터로 나누기 위해 필요한 함수.
* KNeighborsClassifier : KNN 분류 알고리즘.
* accuracy\_score : 성능을 점수로 표현하기 위해 필요한 함수.

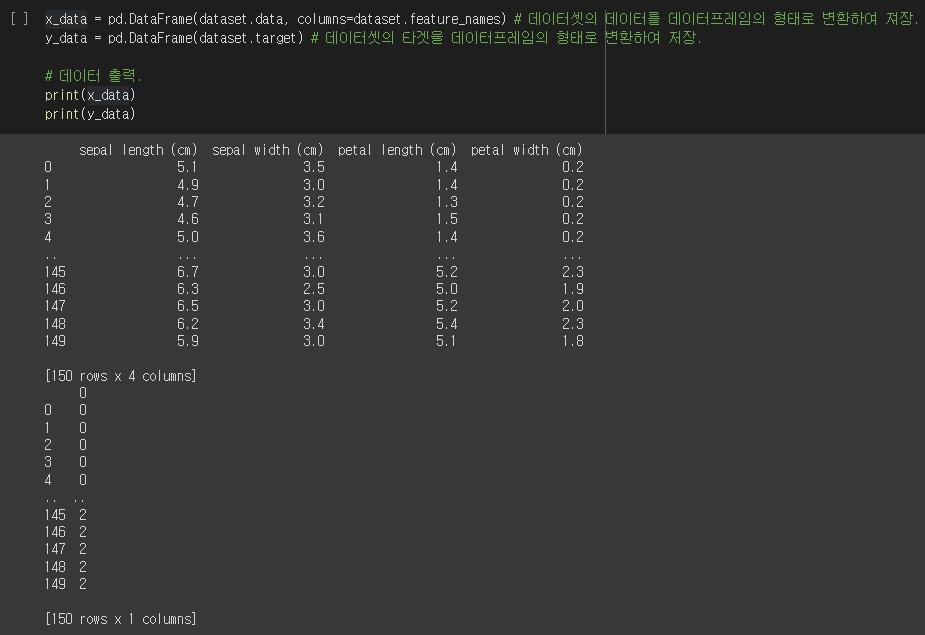
1. 데이터셋

텍스트이(가) 표시된 사진

자동 생성된 설명

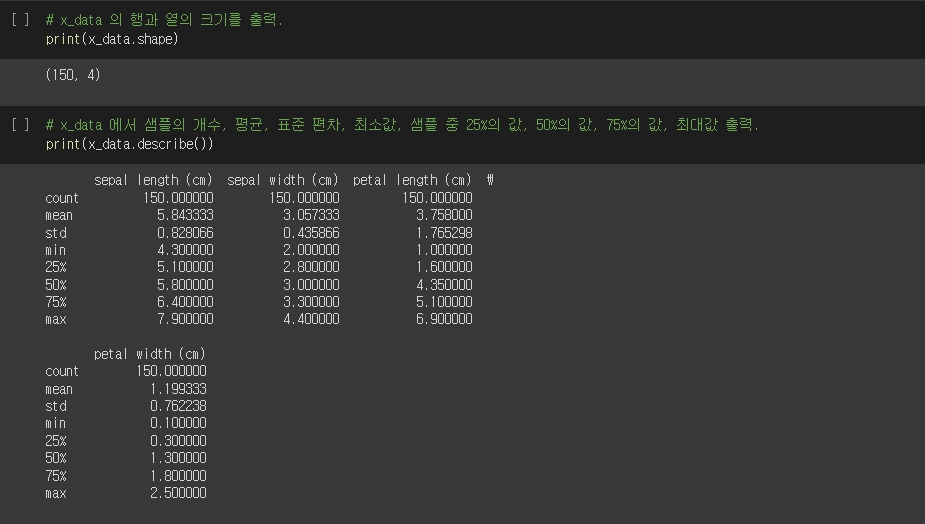
* load\_iris() : Iris 데이터 셋을 불러옴.
* dataset.DESCR : 해당 데이터셋에 대한 정보를 볼 수 있음.
* Number of Instances : 샘플의 개수.
* Number of Attributes : 열(column)의 개수.
* Attribute Information : 열(column)의 정보를 알 수 있음.
* Summary Statistics : 각 열에 대해서 샘플의 최소값, 최대값, 평균값 등을 알 수 있음.
* Missing Attribute Values: None : 값이 비어있는 샘플이 존재하는가를 알 수 있음. 사진에서는 ‘None’ 이라고 되어 있으므로, 값이 비어있는 샘플이 없음을 알 수 있음.

1. 데이터셋을 데이터프레임 형태로



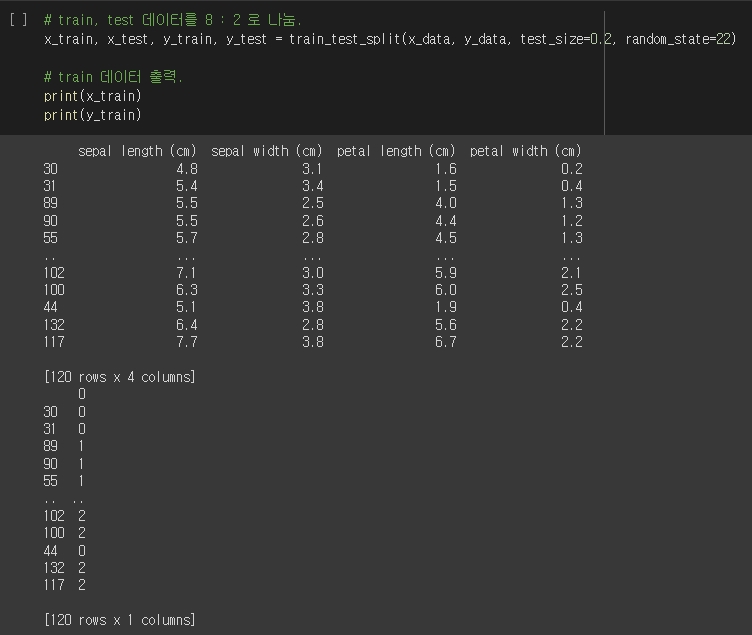
* pd.DataFrame() : 데이터프레임을 생성.
* 데이터셋을 이용하여 인공지능 모델을 학습시키기 전에 데이터셋을 데이터프레임의 형태로 변환하는 전처리 과정을 거침.

1. 데이터프레임 확인



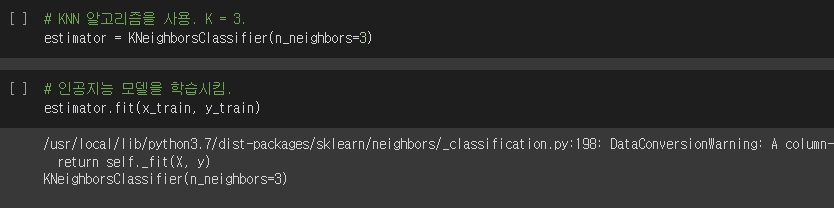
* x\_data.shape : 현재 x\_data 라는 데이터프레임의 행과 열의 크기를 알 수 있음.
* x\_data.describe() : 현재 x\_data 라는 데이터프레임의 종합적인 정보를 알 수 있음.
* 데이터프레임 형태로 잘 변환되었는지 확인.
* 현재 데이터들의 종합적인 정보를 확인.

1. 데이터셋을 분리



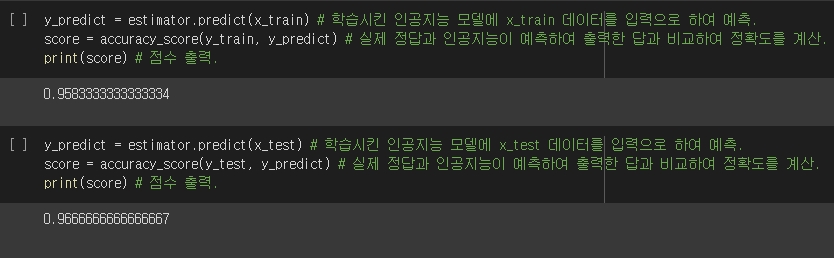
* train\_test\_split() : train 데이터와 test 데이터를 분리.
* train 데이터와 test 데이터를 8 : 2 로 분리하는 전처리 과정을 거침.
* 이러한 과정을 거침으로써, 인공지능 모델이 train 데이터에 과적합(Overfitting)이 되지는 않았는지 확인할 수 있음. 궁극적으로 과적합을 예방하기 위함이다.
* 데이터셋이 train 데이터와 test 데이터로 잘 나누어졌는지 출력을 통해 확인.

1. KNN 알고리즘을 사용하여 인공지능 모델을 학습



* KNeighborsClassifier() : KNN 알고리즘을 이용.
* fit() : 인공지능 모델을 적합 또는 학습시킴.
* 본격적으로 인공지능 모델을 학습시키는 부분.

1. 학습시킨 인공지능 모델을 통해 예측



* predict() : 학습시킨 인공지능 모델에 입력을 주어 출력을 예측하도록 함.
* accuracy\_score() : 실측치와 예측치를 비교하여 점수를 계산.
* train 데이터에 대해서 계산한 점수와 test 데이터에 대해서 계산한 점수를 비교하여 현재 학습시킨 인공지능이 train 데이터에 과적합 되지는 않았는지 확인할 수 있음.