1주차 문제풀이

이론 문제 1. rainmapmap

▼ 클릭

다음중 Supervised learning과 Unsupervised learning에 대하여 틀리게 설명한 것을 골라 주세요.

1. Supervised learning은 지도학습이라고 불리며 특정 사진을 보고 개인지 고양이인지 분류하는 모델을 만들 수 있다.
2. Unsupervised learning은 비지도학습이라고 불리며 여러 동물 사진을 특징 별로 군집화를 할 수 있다.
3. Supervised learning은 feature가 여러 개여도 문제없이 학습이 가능하다.
4. **Unsupervised learning은 데이터 중 training set이 없어도 학습이 가능하다.**

→ 지도학습과 비지도학습 모두 training set가 필요함

이론 문제 2. DATASET을 구분해보자

▼ 클릭

문제내용

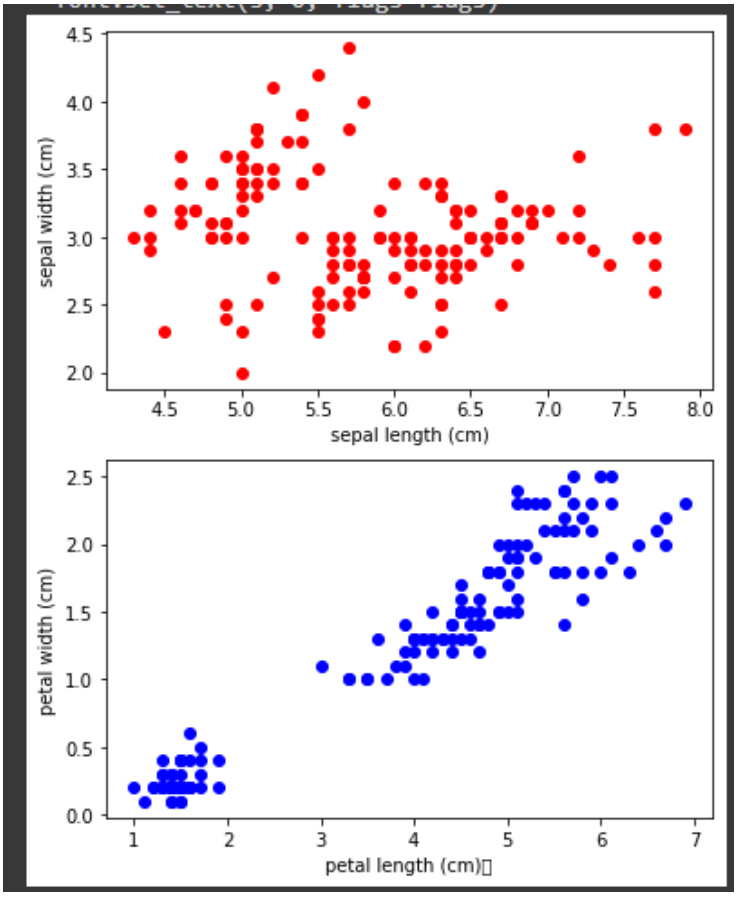
Dataset에 대한 설명 중 틀린 것을 모두 골라 주세요.

1. 학습이 끝난 후 모델 평가에 쓰이는 Test set은 Train set에서 사용하지 않은 데이터여야 한다.
2. 1.의 이유는 평가를 할 때에는 모델이 본 적있는 데이터가 아닌 처음 보는 데이터로 해야하기 때문이다.
3. **Dataset을 Train set : Test set = 1 : 1 비율로 나누면 Overfitting이 발생할 수 있다.**
4. Overfitting을 방지하기 위한 방법으로 Validation set를 만드는 방법이 있다.
5. **Validation set의 사용 목적은 Test set과 일치한다.**

실습문제 3. 라이브러리폭탄

▼ 클릭

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
iris = load_iris()

#데이터프레임 사용
iris['data']
df = pd.DataFrame(iris['data'], columns=iris['feature_names'])
df

#그래프 그리기
plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
plt.scatter(df['sepal length (cm)'], df['sepal width (cm)'], c='r')
plt.xlabel('sepal length (cm)')
plt.ylabel('sepal width (cm)')


plt.subplot(2,1,2)
plt.scatter(df['petal length (cm)'], df['petal width (cm)'], c='b')
plt.xlabel('petal length (cm) ')
plt.ylabel('petal width (cm)')
plt.show(
```



실습문제 4.

▼ 클릭

문제내용

3번 문제의 연장선입니다!

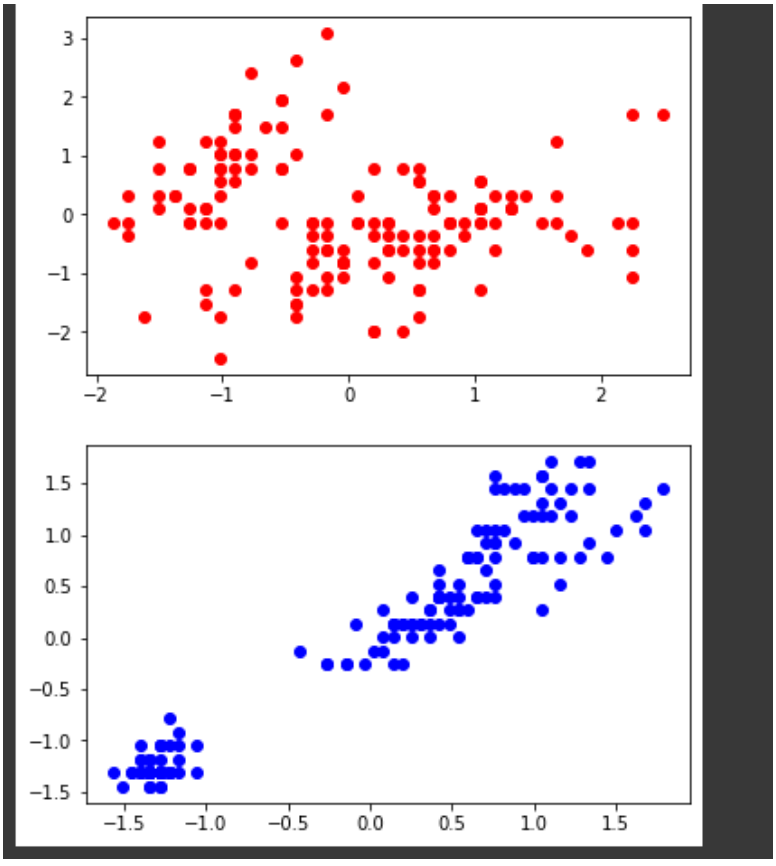
2장에서 데이터 전처리를 하는 법을 배웠어요! 그러면 이 iris data로도 한번 해볼까요?

3번 문제를 푸셨다면 총 4개의 feature가 있는 것을 확인하셨을 텐데 각 feature별로 평균과 표준편차를 이용해서 표준화를 해주세요!

방법은 자유이고 3번의 출력처럼 scatter plot를 코드와 함께 제출해주세요!

```
import numpy as np
df = df.rename({'sepal length (cm)': 'splength',
                        'sepal width (cm)': 'spwidth',
                        'petal length (cm)': 'ptlength',
                        'petal width (cm)': 'ptwidth'}, axis='columns')
```

```
print(df)

#그래프 그리기
mean = np.mean(df['splength'], axis=0)
std = np.std(df['splength'], axis=0)
scaled_spl = (df['splength']-mean)/std

mean = np.mean(df['spwidth'], axis=0)
std = np.std(df['spwidth'], axis=0)
scaled_spw = (df['spwidth']-mean)/std

mean = np.mean(df['ptlength'], axis=0)
std = np.std(df['ptlength'], axis=0)
scaled_ptl = (df['ptlength']-mean)/std

mean = np.mean(df['ptwidth'], axis=0)
std = np.std(df['ptwidth'], axis=0)
scaled_ptw = (df['ptwidth']-mean)/std

plt.figure(figsize=(6,8))
plt.subplot(2,1,1)
plt.scatter(scaled_spl, scaled_spw, c='r')

plt.subplot(2,1,2)
plt.scatter(scaled_ptl,scaled_ptw, c='b')
plt.show()
```



실습문제 5. KNN이 뭘까요

▼ 클릭

문제타입

- [ ] 이론
- [x] 실습

문제내용

와인 데이터 셋으로 KNN을 활용한 코드입니다.

해당 parameter(매개변수)에 여러 값을 넣어보면서 정확도가 높은 값이 나오게 해볼까요?

두 개의 빈칸을 채우고 해당 값들이 어떤 역할을 하는지 작성해주세요

채점 기준에 정확도가 높은가는 포함되지 않습니다.

빈칸 두개 + 역할 이 채점 기준입니다.

```
from sklearn import datasets,preprocessing
import numpy as np
from matplotlib import pyplot as plt

wine = datasets.load_wine(as_frame=True)

x_scaled = preprocessing.scale(wine.data)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, wine.target, test_size= 0.2    )

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors= 5 )

knn.fit(X_train, y_train)
knn.score(X_test, y_test)
```

💡 답안)
test_size 매개변수는 train_test_split() 함수를 이용해 데이터를 train과 test 세트로 나눌 때 비율을 지정하는 함수이다. 기본값은 0.25이다.
n_neighbors는 kneighbors 메서드에서 몇개의 이웃을 참고할지 결정한다.
