인공지능 과제 리포트

과제 제목: IRIS classification using KNN

학번: B911197
이름: 최준영

1. 과제 개요

iris꽃의 4가지 특성 데이터를 기반으로 학습된 모델을 생성한다. 그 후 KNN알고리즘을 사용하여 특성값만으로 데이터가 나타내는 품종을 예측한다.

2. 구현 환경

Pycharm IDE를 사용하여 개발하였습니다.

3. 알고리즘에 대한 설명

한 iris개체는 4가지 특성을 가지며 4차원 벡터로 표현됩니다.
KNN모델의 학습데이터와 테스트 개체의 거리를 연산합니다.
거리연산은 유클리트 방식을 사용하였습니다.
그 후 가장 가까운 K개의 학습데이터 개체들을 선정합니다.
선정된 K개의 개체들의 품종들을 다수결방식, 가중치방식을 적용하여 최종적으로 테스트 개체의 품종을 결정합니다.

4. 데이터에 대한 설명

4.1 Input Feature

2차원 numpy배열 타입으로 구현되어 있습니다.
2차원 배열의 요소는 하나의 iris개체를 의미하며 4차원 벡터로 구현되어 있습니다.
요소들은 한 개체의 4가지 특성을 나타냅니다. 특성정보는 아래와 같습니다.
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
코드내 iris_train_inputs식별자는 학습에 사용될 데이터를 의미합니다.
iris_test_inputs식별자는 테스트에 사용될 데이터를 의미합니다.

4.2 Target Output

output class의 수 등에 대한 설명

output class는 총 3가지 입니다.
0, 1, 2값으로 표현됩니다. class정보는 아래와 같습니다.
['setosa' 'versicolor' 'virginica']
iris_train_results식별자는 학습데이터 개체들의 실제품종을 의미합니다.
iris_test_results식별자는 테스트데이터 개체들의 실제 품종을 의미합니다.
output데이터들의 타입은 1차원 리스트로 구현되어 있습니다.

5. 소스코드에 대한 설명

```
train_scalar = MyScalar()
train_scalar.caculate_means_and_stds(iris_train_inputs)
iris_train_inputs_normalized = train_scalar.normalize(iris_train_inputs)

test_scalar = MyScalar()
test_scalar.caculate_means_and_stds(iris_test_inputs)
iris_test_inputs_normalized = test_scalar.normalize(iris_test_inputs)
```

특징들의 스케일을 normalize하기 위해 KNNClass.py에 MyScalar클래스를 구현하였습니다.
클래스 객체를 생성하고 caculate_means_and_stds매서드로 평균과 표준편차를 구해야하는 데이터를 전달합니다. 그 후 normalize함수를 호출하여 전달한 데이터를 normalize한 배열을 생성합니다.
그렇게 생성한 배열을 iris_train_inputs_normalized, iris_test_inputs_normalized에 각각 저장합니다.

```
classes = [[],[],[]]

for index in range(len(iris_train_inputs_normalized)):
    classes[iris_train_results[index]].append(iris_train_inputs_normalized[index])

fig = plt.figure(figsize=(6, 6))
ax = fig.add_subplot(111, projection='3d')
colorList = ['Greens', 'winter', 'Reds']

for index in range(len(classes)):
    xs = np.array(classes[index])[:,0]
    ys = np.array(classes[index])[:,1]
    zs = np.array(classes[index])[:,2]
    cmin, cmax = 0, 2
    color = np.array([(cmax - cmin) * np.random.random_sample() + cmin for i in range(len(classes[index]))])
    ax.scatter(xs, ys, zs, c=color, marker='o', s=20, cmap=colorList[index])

plt.show()
```

위 코드는 학습데이터중 3가지 특성을 3차원 산점도에 표현하기위한 코드입니다.
classes리스트의 3요소는 3품종의 데이터를 보유합니다.
scatter함수를 사용할 때 품종마다 서로다른 색으로 표시되게 하기위해 colorList를 사용합니다.

```python
def weighted_majority_vote(self, k_nearest_neighbor):
    class_dic = {
        0: 0,
        1: 0,
        2: 0
    }

    # "1 / (1+distance)"를 가중치로 더하여 사용합니다.
    for result, distance in k_nearest_neighbor:
        class_dic[result] += 1 / (1+distance)

    max_count_class = 0
    max_count = 0
    for result, _ in k_nearest_neighbor:
        if max_count < class_dic[result]:
            max_count = class_dic[result]
            max_count_class = result

    return max_count_class
```

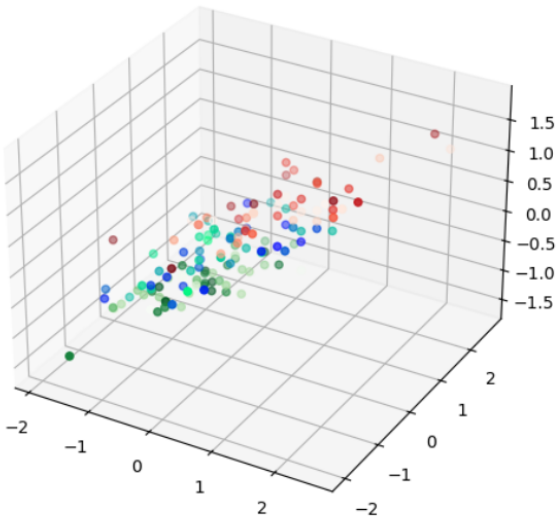
전달받은 인접이웃들 중에서 테스트 개체의 품종을 선별하기위해 weighted majority vote를 사용한 코드입니다.
class_dic은 3가지 품종이 가지는 값을 저장하는데 이 값이 가장큰 품종을 테스트 개체의 품종의로 선정합니다.
이 값은 k_nearest_neighbor리스트의 요소들의 품종별 " 1/(1+거리) " 값의 합입니다.
따라서 거리가 가까울 수록 해당 품종이 선정될 가능성이 높아집니다.

6. 학습 과정에 대한 설명

train input데이터를 시각화 할때 특징들 마다 표현단위가 상이하여 시각상의 인접이웃과 실제 인접이웃이 일치하지 않는 상황이 발생하였습니다. 정규화를 통해서 이 문제를 해결할 수 있었습니다. (특성값 - 평균) / 표준편차 방식을 사용하여 정규화를 하면 모든 데이터가 평균이 0이고 표준편차는1인 스케일을 가지도록 정규화할 수 있었습니다. 정규화를 마친 후 시각화한 데이터는 iris의 각 특징들이 유사한 분포를 가지게 되어 테스트 개체의 인접이웃에 대한 정보를 시각적으로도 파악할 수 있었습니다.

가중 다수결투표를 구하는 과정에서 길이를 반비례하는 공식을 사용할 경우 효과적으로 구현이 가능했습니다.

7. 결과 및 분석



특징 3개를 정규화한 후 표현한 3차원 산점도입니다.
품종별로 색을 달리하였습니다.

품종 분류 테스트 결과입니다.
먼저 k = 3, 5, 7에 대해 다수결로 도출한 결과입니다.

```
----------obtain_majority_vote매서드 사용----------


k값이 3일 때의 결과입니다.

Test Data Index: 0 Compuited Class: setosa, True class: setosa
Test Data Index: 1 Compuited Class: setosa, True class: setosa
Test Data Index: 2 Compuited Class: setosa, True class: setosa
Test Data Index: 3 Compuited Class: setosa, True class: setosa
Test Data Index: 4 Compuited Class: versicolor, True class: versicolor
Test Data Index: 5 Compuited Class: versicolor, True class: versicolor
Test Data Index: 6 Compuited Class: versicolor, True class: versicolor
Test Data Index: 7 Compuited Class: virginica, True class: virginica
Test Data Index: 8 Compuited Class: virginica, True class: virginica
Test Data Index: 9 Compuited Class: virginica, True class: virginica
```

```
k값이 5일 때의 결과입니다.

Test Data Index: 0 Compuited Class: setosa, True class: setosa
Test Data Index: 1 Compuited Class: setosa, True class: setosa
Test Data Index: 2 Compuited Class: setosa, True class: setosa
Test Data Index: 3 Compuited Class: setosa, True class: setosa
Test Data Index: 4 Compuited Class: versicolor, True class: versicolor
Test Data Index: 5 Compuited Class: versicolor, True class: versicolor
Test Data Index: 6 Compuited Class: versicolor, True class: versicolor
Test Data Index: 7 Compuited Class: virginica, True class: virginica
Test Data Index: 8 Compuited Class: virginica, True class: virginica
Test Data Index: 9 Compuited Class: virginica, True class: virginica
```

```
k값이 7일 때의 결과입니다.

Test Data Index: 0 Compuited Class: setosa, True class: setosa
Test Data Index: 1 Compuited Class: setosa, True class: setosa
Test Data Index: 2 Compuited Class: setosa, True class: setosa
Test Data Index: 3 Compuited Class: setosa, True class: setosa
Test Data Index: 4 Compuited Class: versicolor, True class: versicolor
Test Data Index: 5 Compuited Class: versicolor, True class: versicolor
Test Data Index: 6 Compuited Class: versicolor, True class: versicolor
Test Data Index: 7 Compuited Class: virginica, True class: virginica
Test Data Index: 8 Compuited Class: virginica, True class: virginica
Test Data Index: 9 Compuited Class: virginica, True class: virginica
```

가중 다수결 투표로 도출한 결과입니다.

```
----------weighted_majority_vote매서드 사용----------


k값이 3일 때의 결과입니다.

Test Data Index: 0 Compuited Class: setosa, True class: setosa
Test Data Index: 1 Compuited Class: setosa, True class: setosa
Test Data Index: 2 Compuited Class: setosa, True class: setosa
Test Data Index: 3 Compuited Class: setosa, True class: setosa
Test Data Index: 4 Compuited Class: versicolor, True class: versicolor
Test Data Index: 5 Compuited Class: versicolor, True class: versicolor
Test Data Index: 6 Compuited Class: versicolor, True class: versicolor
Test Data Index: 7 Compuited Class: virginica, True class: virginica
Test Data Index: 8 Compuited Class: virginica, True class: virginica
Test Data Index: 9 Compuited Class: virginica, True class: virginica
```

```
k값이 5일 때의 결과입니다.

Test Data Index: 0 Compuited Class: setosa, True class: setosa
Test Data Index: 1 Compuited Class: setosa, True class: setosa
Test Data Index: 2 Compuited Class: setosa, True class: setosa
Test Data Index: 3 Compuited Class: setosa, True class: setosa
Test Data Index: 4 Compuited Class: versicolor, True class: versicolor
Test Data Index: 5 Compuited Class: versicolor, True class: versicolor
Test Data Index: 6 Compuited Class: versicolor, True class: versicolor
Test Data Index: 7 Compuited Class: virginica, True class: virginica
Test Data Index: 8 Compuited Class: virginica, True class: virginica
Test Data Index: 9 Compuited Class: virginica, True class: virginica
```

```
k값이 7일 때의 결과입니다.

Test Data Index: 0 Compuited Class: setosa, True class: setosa
Test Data Index: 1 Compuited Class: setosa, True class: setosa
Test Data Index: 2 Compuited Class: setosa, True class: setosa
Test Data Index: 3 Compuited Class: setosa, True class: setosa
Test Data Index: 4 Compuited Class: versicolor, True class: versicolor
Test Data Index: 5 Compuited Class: versicolor, True class: versicolor
Test Data Index: 6 Compuited Class: versicolor, True class: versicolor
Test Data Index: 7 Compuited Class: virginica, True class: virginica
Test Data Index: 8 Compuited Class: virginica, True class: virginica
Test Data Index: 9 Compuited Class: virginica, True class: virginica
```