인공지능 과제 리포트

과제 제목: iris data 분류/KNN 구현

학번: B511226
이름: 홍현승

1. 과제 개요

150행으로 이루어진 iris data-> 140개의 train data와 10개의 test data로 분할
1개의(1행) irisdata는 4개의 feature을 가지고 있고, label로 3개의 class인 Setosa, Versicolor, Virginca에 해당된다.
train 데이터 140개를 공간상에 찍어두고, 10개의 test data로, 140개의 데이터 중 어느 데이터랑 가장 근접한지(KNN, weighted KNN) 판단하여 출력해보는 과제.

2. 구현 환경

OS: Window
Integrated Development Environment: Jupyter
Using library: math / sklearn / numpy

3. 알고리즘에 대한 설명

KNN 알고리즘은 기존 데이터(training data)를 학습 시킨 후, 이 데이터중 test data가 가장 가까운 부류가 무엇인지 선택하는 알고리즘이다. 이때 "가장 가까운 부류를 선택"하는 방식에는 majority vote 방식과, weighted majority vote 방식이 있다. 이때 사용자가 지정하는 k, 즉 몇 개의 근접 data를 잡을 것인지를 판단하게 되는데, k가 너무 적으면 data underfitting이, k가 너무 커지면 data overfitting이 일어나 판단 오류가 생기게 된다.

4. 데이터에 대한 설명

4.1 Input Feature
150*5 iris data -> (140*4 training data, 140*1 training data의 label data) / (10*4 test data, 10*1 test data의 label data)로 train test split구현,
4.2 Target Output
test data로 train data와 근접한 class를 예측 -> 예측 값은 10*1행렬(각각의 값은 setosa/versicolor/virginica중 하나임)을, 실제 test data의 class와 일치하는지 비교

5. 학습 과정에 대한 설명

1)140개의 training data를 공간에 찍어두고 10개의 test data를 넣음
2)각각의 test data(10개)의 training data와의 거리를 구함
->def distance(p1, p2):

```
    p1 = np.array(p1)
    p2 = np.array(p2)
    return np.sqrt(np.sum(np.power((p2-p1),2)))
```

3)"test data 1개"와 "train data 140개"의 거리가 담겨있는 dist 리스트(1*140행렬)를 sorting 한다 ->이러면 가까운 거리부터 앞에 오게 된다
->temp = dist.argsort() =>cf)이때 temp를 찍어보면, 값이 아닌 index가 결과로 나온다!!
4)k(k = 3,5,10)에 따라 temp(인덱스가 담겨있다)에서 앞에 3개/5개/10개를 뽑는다.

5)3개/5개/10개중 voting을 거쳐 가장 많이 존재하는 값을 선택한다.

6. 결과 및 분석

Majority Vote, K = 3

```
for i in range(10):
    print("Test Data Index: "+ str(i), end =' ')
    classify(train_data, test_data[i], 3, train_label)
    print("True class: "+ flowers[int(test_label[i])])# flowers 딕셔너
```

```
Test Data Index: 0 computed class: Setosa True class: Setosa
Test Data Index: 1 computed class: Setosa True class: Setosa
Test Data Index: 2 computed class: Setosa True class: Setosa
Test Data Index: 3 computed class: Versicolor True class: Versicolor
Test Data Index: 4 computed class: Versicolor True class: Versicolor
Test Data Index: 5 computed class: Versicolor True class: Versicolor
Test Data Index: 6 computed class: Virginica True class: Virginica
Test Data Index: 7 computed class: Versicolor True class: Virginica
Test Data Index: 8 computed class: Virginica True class: Virginica
Test Data Index: 9 computed class: Virginica True class: Virginica
```

Majority Vote, K = 5

```
for i in range(10):
    print("Test Data Index: "+ str(i), end =' ')
    classify(train_data, test_data[i], 5, train_label)
    print("True class: "+ flowers[int(test_label[i])])
```

```
Test Data Index: 0 computed class: Setosa True class: Setosa
Test Data Index: 1 computed class: Setosa True class: Setosa
Test Data Index: 2 computed class: Setosa True class: Setosa
Test Data Index: 3 computed class: Versicolor True class: Versicolor
Test Data Index: 4 computed class: Versicolor True class: Versicolor
Test Data Index: 5 computed class: Versicolor True class: Versicolor
Test Data Index: 6 computed class: Virginica True class: Virginica
Test Data Index: 7 computed class: Versicolor True class: Virginica
Test Data Index: 8 computed class: Virginica True class: Virginica
Test Data Index: 9 computed class: Virginica True class: Virginica
```

Majority Vote, K = 10

```
for i in range(10):
    print("Test Data Index: "+ str(i), end =' ')
    classify(train_data, test_data[i], 10, train_label)
    print("True class: "+ flowers[int(test_label[i])])
```

```
Test Data Index: 0 computed class: Setosa True class: Setosa
Test Data Index: 1 computed class: Setosa True class: Setosa
Test Data Index: 2 computed class: Setosa True class: Setosa
Test Data Index: 3 computed class: Versicolor True class: Versicolor
Test Data Index: 4 computed class: Versicolor True class: Versicolor
Test Data Index: 5 computed class: Versicolor True class: Versicolor
Test Data Index: 6 computed class: Virginica True class: Virginica
Test Data Index: 7 computed class: Virginica True class: Virginica
Test Data Index: 8 computed class: Virginica True class: Virginica
Test Data Index: 9 computed class: Virginica True class: Virginica
```

Weighted Majority Vote, K = 3

```
for i in range(10):
    print("Test Data Index: "+ str(i), end =' ')
    weighted_classify(train_data, test_data[i], 3, train_label)
    print("True class: "+ flowers[int(test_label[i])])
```

```
Test Data Index: 0 computed class: Setosa True class: Setosa
Test Data Index: 1 computed class: Setosa True class: Setosa
Test Data Index: 2 computed class: Setosa True class: Setosa
Test Data Index: 3 computed class: Versicolor True class: Versicolor
Test Data Index: 4 computed class: Versicolor True class: Versicolor
Test Data Index: 5 computed class: Versicolor True class: Versicolor
Test Data Index: 6 computed class: Virginica True class: Virginica
Test Data Index: 7 computed class: Versicolor True class: Virginica
Test Data Index: 8 computed class: Versicolor True class: Virginica
Test Data Index: 9 computed class: Virginica True class: Virginica
```

Weighted Majority Vote, K = 5

```
for i in range(10):
    print("Test Data Index: "+ str(i), end =' ')
    weighted_classify(train_data, test_data[i], 5, train_label)
    print("True class: "+ flowers[int(test_label[i])])
```

```
Test Data Index: 0 computed class: Setosa True class: Setosa
Test Data Index: 1 computed class: Setosa True class: Setosa
Test Data Index: 2 computed class: Setosa True class: Setosa
Test Data Index: 3 computed class: Versicolor True class: Versicolor
Test Data Index: 4 computed class: Versicolor True class: Versicolor
Test Data Index: 5 computed class: Versicolor True class: Versicolor
Test Data Index: 6 computed class: Virginica True class: Virginica
Test Data Index: 7 computed class: Versicolor True class: Virginica
Test Data Index: 8 computed class: Versicolor True class: Virginica
Test Data Index: 9 computed class: Versicolor True class: Virginica
```

Weighted Majority Vote, K = 10

```
for i in range(10):
    print("Test Data Index: "+ str(i), end =' ')
    weighted_classify(train_data, test_data[i], 10, train_label)
    print("True class: "+ flowers[int(test_label[i])])
```

```
Test Data Index: 0 computed class: Setosa True class: Setosa
Test Data Index: 1 computed class: Setosa True class: Setosa
Test Data Index: 2 computed class: Setosa True class: Setosa
Test Data Index: 3 computed class: Versicolor True class: Versicolor
Test Data Index: 4 computed class: Versicolor True class: Versicolor
Test Data Index: 5 computed class: Versicolor True class: Versicolor
Test Data Index: 6 computed class: Virginica True class: Virginica
Test Data Index: 7 computed class: Versicolor True class: Virginica
Test Data Index: 8 computed class: Versicolor True class: Virginica
Test Data Index: 9 computed class: Virginica True class: Virginica
```

majority vote의 경우, K = 3,5 인 경우 한 개가 불일치 하는 결과를 보였으나, K = 10인 경우 Data underfitting 문제가 해결되어 모든 데이터가 다 정확하게 예측 되었다.

weighted majority vote의 경우 weight를 주는 방식에 따라서 결과가 상승하기도 하고 하락하기도 하는 경향을 보였다.