수치해석 Project#2

K-Means Clustering

Python 사용

컴퓨터소프트웨어학부 2015005187 최철훈

목차

Term Project 개요

이번 Term Project에서는 총 데이터가 분포하는 여러 경우를 생각하여 총 3가지 유형의 데이터를 생성하였다.

1. 데이터가 극단적으로 분포해 있는 경우
2. 데이터가 흩뿌려져 있는 경우
3. 데이터가 균일하게 분포하는 경우

랜덤 데이터 생성 극단적인 경우

첫 번째 경우의 데이터는 극단적인 경우로, 겹치는 부분이 없는 데이터, 균등한 데이터, 극단적으로 긴 데이터 등을 생성하였다.

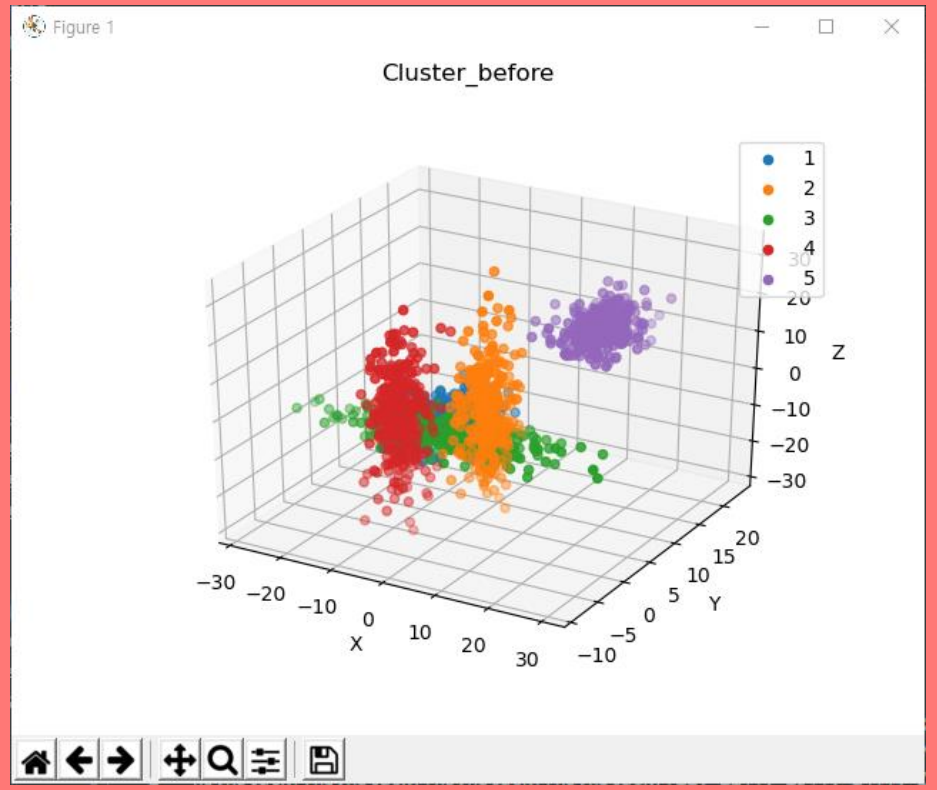
X1 ~ N(0, 3), Y1 ~ N(0, 3), Z1 ~ N(0, 3)

X2 ~ N(4, 2), Y2 ~ N(4, 2), Z2 ~ N(0, 10)

X3 ~ N(0, 10), Y3 ~ N(0, 2), Z3 ~ N(-4, 2)

X4 ~ N(-4, 2), Y4 ~ N(-4, 2), Z4 ~ N(4, 10)

X5 ~ N(15, 3), Y5 ~ N(15, 3), Z5 ~ N(15, 3)



랜덤 데이터 생성 흩뿌려진 경우

두 번째 경우의 데이터는 좀 넓게 흩뿌려진 경우로, 조금씩 겹치는 부분이 있도록 생성하였다.

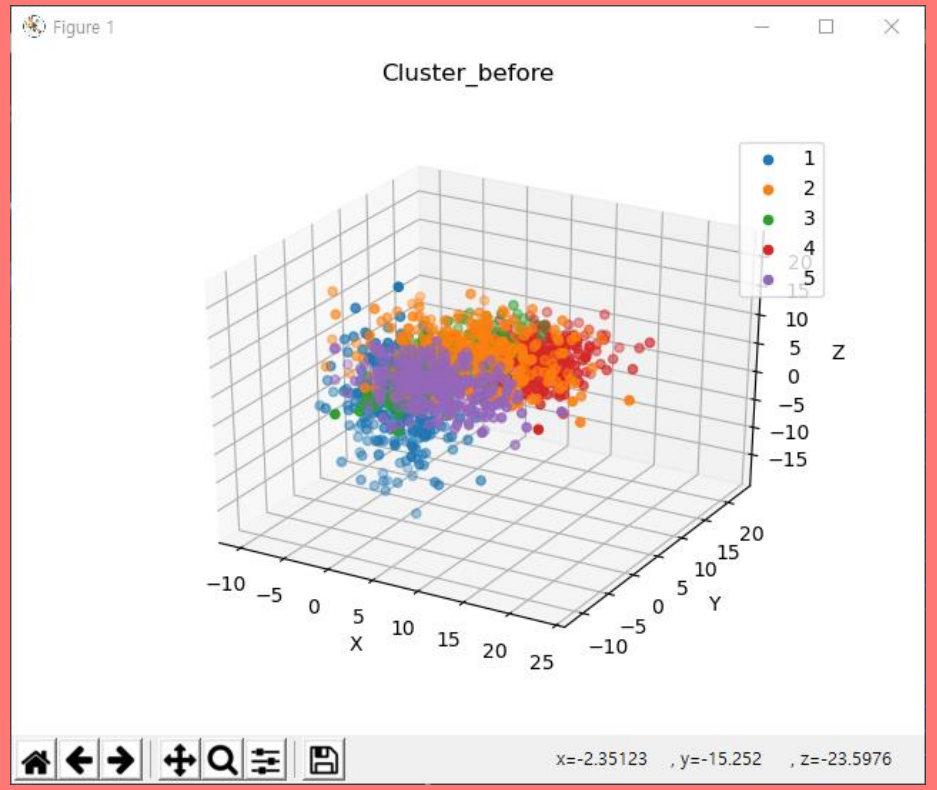
X1 ~ N(0, 3), Y1 ~ N(0, 2), Z1 ~ N(0, 6)

X2 ~ N(6, 5), Y2 ~ N(4, 4), Z2 ~ N(8, 3)

X3 ~ N(2, 2), Y3 ~ N(6, 6), Z3 ~ N(4, 2)

X4 ~ N(10, 3), Y4 ~ N(10, 4), Z4 ~ N(5, 2)

X5 ~ N(5, 4), Y5 ~ N(-2, 2), Z5 ~ N(6, 3)



랜덤 데이터 생성 균일한 경우

세 번째 경우의 데이터는 균일한 경우로, 조금씩 겹치고 매우 균일하게 뭉쳐서 잘 분포해 있도록 생성하였다.

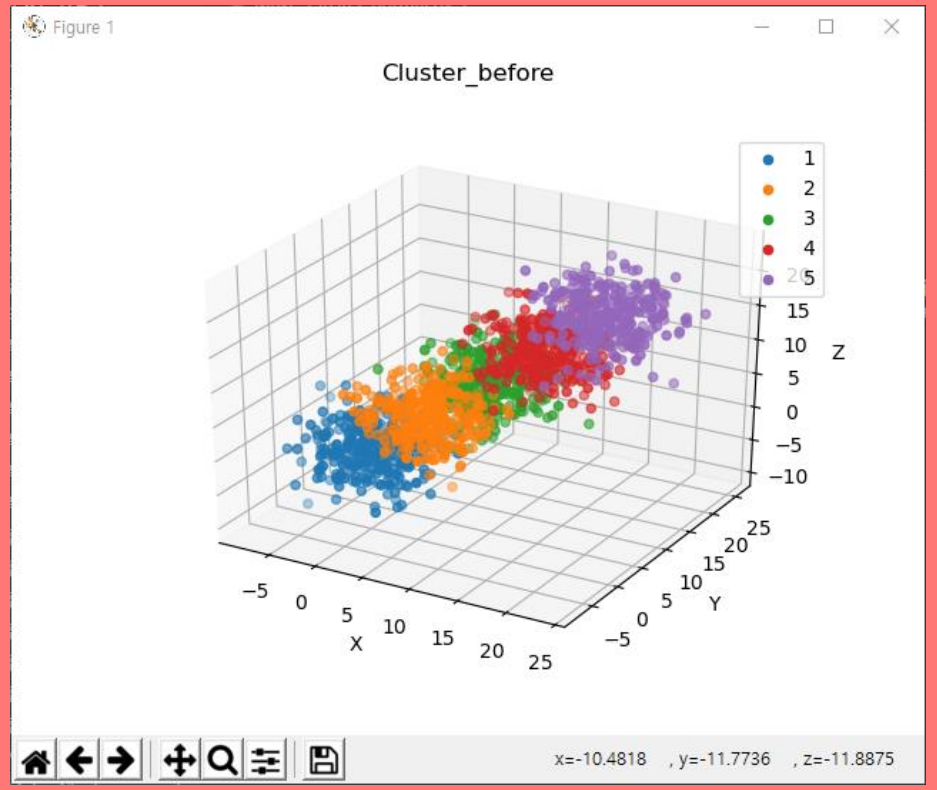
X1 ~ N(0, 3), Y1 ~ N(0, 3), Z1 ~ N(0, 3)

X2 ~ N(4, 3), Y2 ~ N(4, 3), Z2 ~ N(4, 3)

X3 ~ N(8, 3), Y3 ~ N(8, 3), Z3 ~ N(8, 3)

X4 ~ N(12, 3), Y4 ~ N(12, 3), Z4 ~ N(12, 3)

X5 ~ N(16, 3), Y5 ~ N(16, 3), Z5 ~ N(16, 3)

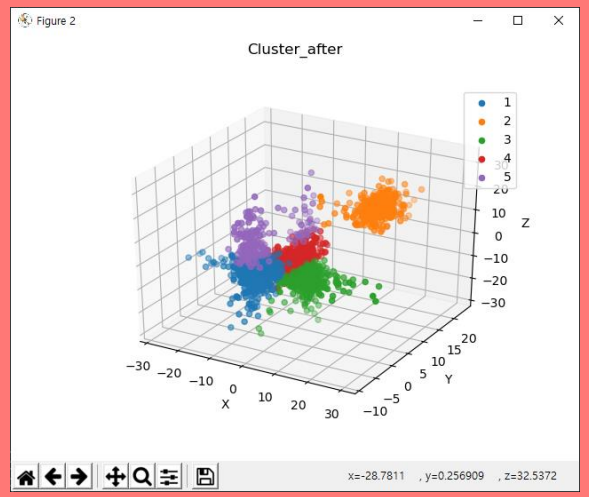


K-Means Clustering 분류결과

앞선 3가지의 경우를 K-Means Clustering으로 분류하였다. 각 클러스터의 평균 좌표와 시각화한 결과이다.

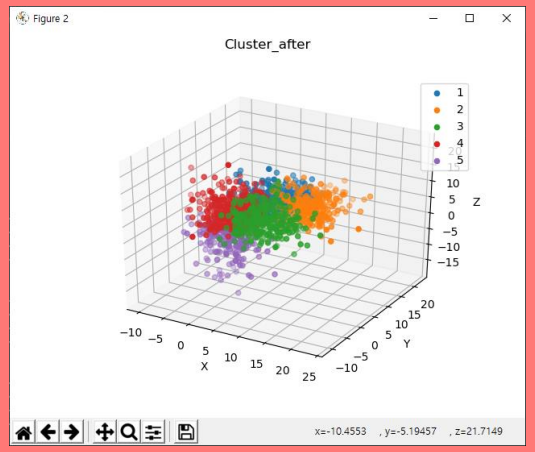
극단적

```
[[-5.1005 -2.1295 -3.6508]
 [14.7835 14.651  15.2906]
 [ 6.9038  2.1721 -7.3894]
 [ 1.9272  1.7192  0.9793]
 [-2.3981 -2.0686 12.5732]]
```



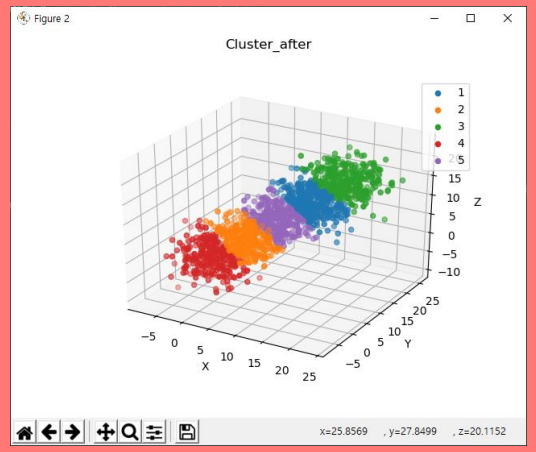
흩뿌림

```
[[ 3.1526 10.5007  4.8733]
 [10.9108  9.0481  6.0063]
 [ 7.6182 -1.0488  6.5441]
 [ 0.6394  0.6262  6.1716]
 [-0.4148  0.1151 -3.2563]]
```



균일

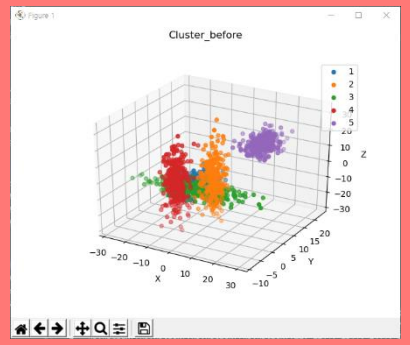
```
[[11.9944 11.9695 11.783 ]
 [ 3.8859  3.9757  3.7614]
 [16.226  16.2555 16.4836]
 [-0.0348 -0.2619 -0.027 ]
 [ 7.8812  7.4465  8.1019]]
```

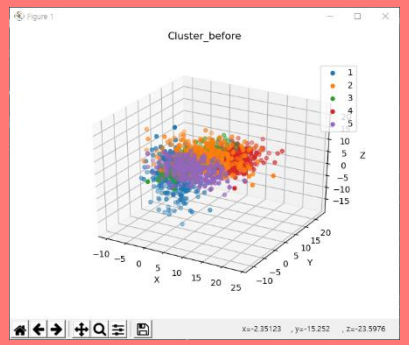


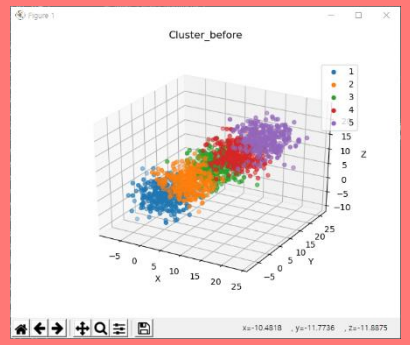
K-Means Clustering 비교

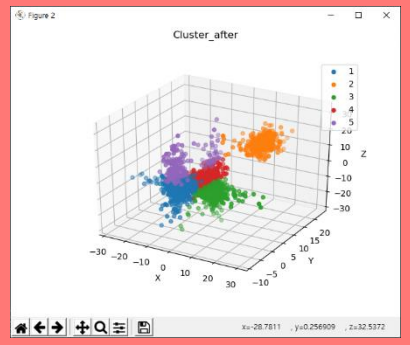
원래 생성된 데이터와 클러스터링을 하고 난 후의 데이터 분포를 비교하였다.

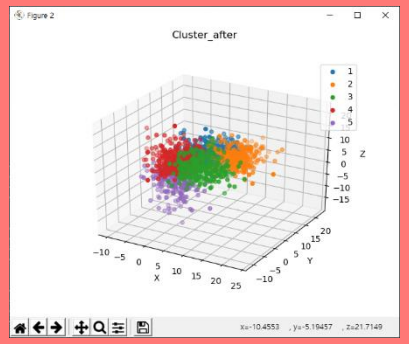
첫 번째 경우는 완전 다르게 클러스터링 되었다. 두 번째 경우는 흩뿌려진게 오밀조밀 모여서 클러스터링 되었다. 세 번째 경우는 원래의 분포에서 경계가 명확하게 나뉘어졌다.

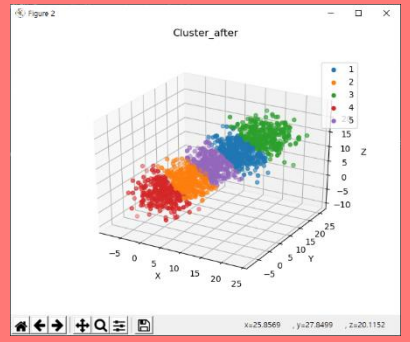












Test 개요

1. 테스트 데이터를 각각의 경우에 생성했던 데이터와 동일한 분포로 100개씩 생성하였다.
2. 해당 데이터는 클러스터링으로 구한 평균과의 거리가 가장 작은 클러스터에 속하는 것으로 판별하였다.
3. 직접 거리들을 구해보니 7보다 작은 값들이 대부분이었다. 그러므로 평균과의 거리가 7이상인 점은 어떤 클러스터에도 속하지 않는다고 판별하여 0으로 라벨링하였다.
4. 앞서 구했던 분포의 순서와 클러스터링 후 분포의 순서가 다르므로 클러스터링 된 평균을 보고 순서를 매칭시켜야한다.
5. 6번째 테스트는 어느 클러스터와도 겹치지 않게 설정하였다.

Test 극단적인 경우

test6는 X, Y, Z 모두 N(-12, 3)으로 생성하여 겹치지 않는 것을 볼 수 있다.

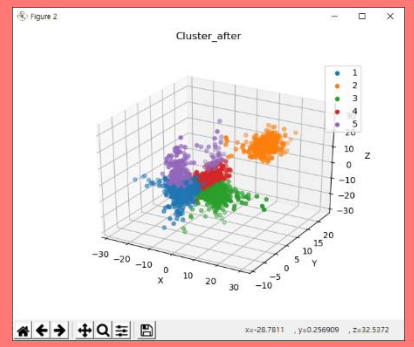
첫 번째의 경우는 원래의 분포와 너무나도 다르게 클러스터링 되었기에 해당 테스트가 어느 클러스터로 판별이 나야 제대로 분류한 것인지를 알 수 없어 분류가 실패했다고 볼 수 있다.

원래 분포

X1 ~ N(0, 3), Y1 ~ N(0, 3), Z1 ~ N(0, 3)
X2 ~ N(4, 2), Y2 ~ N(4, 2), Z2 ~ N(0, 10)
X3 ~ N(0, 10), Y3 ~ N(0, 2), Z3 ~ N(-4, 2)
X4 ~ N(-4, 2), Y4 ~ N(-4, 2), Z4 ~ N(4, 10)
X5 ~ N(15, 3), Y5 ~ N(15, 3), Z5 ~ N(15, 3)

클러스터링 된 평균 및 결과

```
[[-5.1005 -2.1295 -3.6508]
 [14.7835 14.651  15.2906]
 [ 6.9038  2.1721 -7.3894]
 [ 1.9272  1.7192  0.9793]
 [-2.3981 -2.0686 12.5732]]
```



원래 분포로 생성한 테스트 데이터의 클러스터링 된 평균과의 라벨링

```
test1
[1. 4. 1. 1. 1. 4. 4. 1. 4. 1. 4. 0. 4. 4. 4. 4. 1. 4. 4. 0. 3. 4. 4. 4.
 0. 1. 4. 4. 1. 4. 4. 4. 0. 4. 4. 4. 4. 1. 4. 4. 4. 4. 4. 4. 4. 4. 1. 4.
 4. 4. 1. 1. 4. 3. 0. 4. 4. 1. 4. 1. 1. 1. 4. 4. 4. 4. 4. 4. 4. 4. 1. 3.
 4. 4. 4. 0. 1. 1. 4. 4. 4. 1. 1. 4. 1. 4. 4. 4. 4. 4. 4. 1. 4. 1. 4. 4. 0.
 4. 4. 4. 4.]
test2
[3. 4. 0. 0. 3. 0. 4. 0. 4. 0. 4. 4. 0. 0. 3. 4. 4. 4. 4. 3. 3. 4. 4. 0.
 0. 5. 0. 3. 0. 4. 3. 4. 3. 0. 0. 4. 0. 0. 3. 4. 4. 3. 0. 0. 4. 4. 3. 0.
 4. 4. 3. 4. 3. 0. 3. 3. 0. 0. 0. 3. 0. 3. 4. 4. 0. 0. 0. 0. 4. 0. 0. 3.
 4. 0. 4. 4. 4. 0. 0. 4. 4. 3. 3. 4. 4. 0. 4. 0. 4. 4. 3. 0. 0. 5. 4. 3.
 4. 4. 0. 3.]
test3
[3. 1. 0. 1. 1. 3. 0. 0. 1. 3. 4. 0. 4. 4. 0. 3. 3. 0. 1. 0. 1. 0. 4. 3.
 3. 0. 1. 1. 1. 4. 4. 0. 0. 1. 0. 4. 0. 0. 0. 0. 0. 3. 1. 0. 1. 4. 3. 1.
 0. 3. 1. 0. 3. 0. 3. 4. 1. 1. 1. 0. 4. 0. 1. 4. 0. 0. 1. 4. 3. 0. 0. 0.
 1. 3. 3. 0. 1. 4. 0. 1. 0. 1. 3. 1. 0. 3. 0. 4. 0. 1. 1. 4. 0. 0. 1. 1.
 0. 1. 1. 0.]
```

```
test4
[1. 0. 5. 5. 0. 1. 1. 1. 5. 0. 0. 0. 0. 5. 1. 1. 1. 0. 1. 0. 5. 1. 1. 0.
 0. 1. 1. 5. 0. 5. 1. 1. 5. 0. 5. 5. 1. 5. 0. 1. 5. 0. 0. 1. 1. 5. 5. 0.
 5. 1. 0. 1. 0. 5. 5. 0. 4. 1. 5. 1. 0. 5. 0. 5. 5. 5. 0. 1. 5. 5. 5. 1.
 0. 5. 5. 1. 1. 5. 1. 0. 5. 5. 5. 5. 1. 0. 0. 1. 5. 0. 1. 0. 1. 5. 1. 0.
 0. 0. 1. 0.]
test5
[2. 2. 0. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 0. 2. 0. 0. 2. 0. 0. 0. 2. 2. 2.
 2. 2. 2. 2. 2. 0. 2. 2. 2. 2. 2. 0. 2. 2. 2. 2. 2. 0. 2. 2. 2. 2. 2. 2. 2.
 0. 2. 2. 2. 2. 2. 2. 2. 0. 2. 2. 2. 2. 2. 2. 0. 0. 2. 2. 2. 0. 2. 0. 2.
 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 0. 2.
 2. 0. 2. 2.]
test6
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

Test 흩뿌려진 경우

test6는 X, Y는 N(15, 2)로, Z는 N(12, 2)로 생성하여 2와의 거리가 7이하인 데이터가 조금 확인된다.

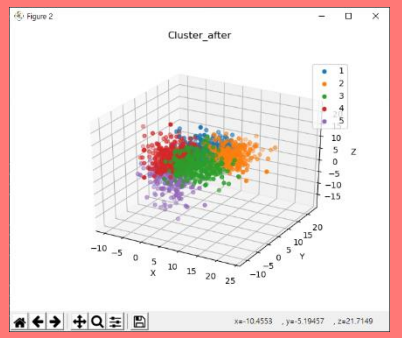
두 번째의 경우 test1은 5, test4는 2, test5는 3으로 분류되면 제대로 분류된 것이다. 나머지 2개의 test는 클러스터링 된 평균이 원래의 평균과 너무 다르게 나와 어느 클러스터에 속해야 한다고 판별할 수 없다. test1, 4, 5는 그래도 잘 분류되었다고 볼 수 있다.

원래 분포

X1 ~ N(0, 3), Y1 ~ N(0, 2), Z1 ~ N(0, 6)
X2 ~ N(6, 5), Y2 ~ N(4, 4), Z2 ~ N(8, 3)
X3 ~ N(2, 2), Y3 ~ N(6, 6), Z3 ~ N(4, 2)
X4 ~ N(10, 3), Y4 ~ N(10, 4), Z4 ~ N(5, 2)
X5 ~ N(5, 4), Y5 ~ N(-2, 2), Z5 ~ N(6, 3)

클러스터링 된 평균 및 결과

```
[[ 3.1526 10.5007  4.8733]
 [10.9108  9.0481  6.0063]
 [ 7.6182 -1.0488  6.5441]
 [ 0.6394  0.6262  6.1716]
 [-0.4148  0.1151 -3.2563]]
```



원래 분포로 생성한 테스트 데이터의 클러스터링 된 평균과의 라벨링

```
test1
[5. 5. 4. 5. 4. 5. 5. 3. 0. 0. 5. 4. 5. 5. 4. 0. 5. 5. 4. 4. 4. 4. 4. 4.
 4. 5. 5. 5. 5. 5. 4. 3. 5. 5. 3. 4. 4. 5. 4. 5. 0. 0. 5. 5. 4. 3. 5. 5.
 5. 4. 5. 4. 5. 0. 5. 4. 5. 5. 4. 4. 0. 4. 4. 5. 5. 0. 0. 5. 5. 5. 3. 5.
 4. 5. 4. 5. 5. 5. 5. 5. 4. 5. 0. 5. 5. 5. 5. 5. 0. 5. 5. 0. 5. 5. 4. 0.
 4. 5. 4. 4.]
test2
[2. 1. 0. 1. 4. 0. 3. 2. 4. 4. 3. 3. 1. 4. 0. 2. 3. 1. 2. 4. 0. 3. 4. 0.
 1. 1. 3. 3. 2. 3. 3. 0. 2. 3. 4. 3. 3. 2. 1. 4. 3. 0. 2. 0. 0. 4. 4. 0.
 1. 1. 2. 4. 0. 1. 0. 2. 2. 0. 3. 1. 4. 4. 1. 1. 4. 3. 3. 3. 4. 3. 2. 3.
 0. 0. 1. 2. 0. 0. 1. 0. 3. 0. 1. 4. 0. 4. 3. 2. 3. 3. 4. 3. 0. 4. 3. 2.
 3. 3. 2. 2.]
test3
[1. 4. 1. 0. 4. 1. 4. 1. 1. 5. 1. 1. 1. 4. 4. 1. 1. 1. 4. 1. 3. 1. 4. 4.
 4. 0. 1. 4. 1. 1. 4. 1. 1. 0. 4. 4. 4. 1. 4. 5. 4. 4. 1. 1. 1. 1. 4. 1.
 1. 1. 4. 1. 5. 1. 4. 1. 0. 1. 1. 1. 1. 1. 0. 1. 4. 1. 1. 1. 4. 4. 4. 4.
 4. 1. 0. 1. 1. 1. 1. 5. 5. 1. 4. 1. 1. 1. 0. 3. 4. 1. 0. 1. 4. 4. 1. 1.
 4. 1. 4. 0.]
```

```
test4
[2. 1. 2. 1. 2. 2. 1. 2. 2. 2. 2. 2. 2. 2. 0. 2. 2. 2. 2. 2. 2. 1. 2. 2.
 2. 2. 1. 2. 2. 2. 2. 1. 0. 2. 2. 2. 2. 2. 1. 2. 2. 1. 2. 0. 2. 2. 2. 2.
 0. 2. 2. 2. 2. 2. 2. 2. 3. 1. 0. 1. 2. 2. 0. 1. 2. 2. 0. 0. 2. 1. 2. 0.
 2. 2. 0. 2. 0. 2. 0. 1. 1. 2. 2. 2. 2. 1. 2. 1. 2. 2. 2. 2. 2. 0. 2. 2.
 2. 1. 0. 2.]
test5
[3. 4. 4. 0. 5. 3. 3. 3. 3. 4. 3. 4. 3. 3. 3. 3. 3. 4. 3. 3. 3. 4. 3. 5.
 4. 4. 4. 3. 3. 0. 0. 3. 3. 0. 0. 5. 3. 3. 3. 4. 3. 3. 5. 3. 4. 3. 3. 0.
 4. 3. 0. 3. 3. 3. 3. 3. 4. 4. 3. 3. 4. 4. 4. 4. 3. 3. 3. 4. 3. 4. 4. 3.
 4. 3. 3. 3. 4. 3. 4. 3. 3. 3. 0. 4. 3. 3. 3. 3. 3. 4. 4. 3. 4. 3. 4. 4.
 3. 3. 4. 3.]
test6
[0. 0. 0. 0. 2. 0. 0. 0. 0. 0. 0. 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2.
 0. 0. 0. 0. 0. 0. 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

Test 균일한 경우

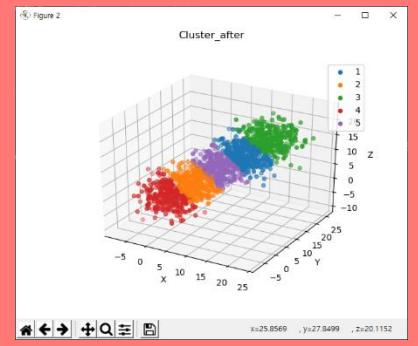
test6는 X, Y, Z 모두 N(-5, 3)으로 생성하여 4와의 거리가 7이하인 점이 꽤 있어 4로 판별한 것을 확인할 수 있다.
세 번째의 경우는 역시 클러스터링이 원래의 분포와 동일하게 진행된 것을 확인할 수 있다. test1은 4, test2는 2, test3는 5, test4는 1, test5는 3으로 분류되면 잘 분류된 것이다.

원래 분포

X1 ~ N(0, 3), Y1 ~ N(0, 3), Z1 ~ N(0, 3)
X2 ~ N(4, 3), Y2 ~ N(4, 3), Z2 ~ N(4, 3)
X3 ~ N(8, 3), Y3 ~ N(8, 3), Z3 ~ N(8, 3)
X4 ~ N(12, 3), Y4 ~ N(12, 3), Z4 ~ N(12, 3)
X5 ~ N(16, 3), Y5 ~ N(16, 3), Z5 ~ N(16, 3)

클러스터링 된 평균 및 결과

```
[[11.9944 11.9695 11.783 ]
 [ 3.8859  3.9757  3.7614]
 [16.226  16.2555 16.4836]
 [-0.0348 -0.2619 -0.027 ]
 [ 7.8812  7.4465  8.1019]]
```



원래 분포로 생성한 테스트 데이터의 클러스터링 된 평균과의 라벨링

```
test1
[4. 4. 4. 4. 2. 4. 4. 4. 0. 4. 2. 4. 2. 4. 4. 4. 4. 2. 4. 4. 0. 4. 4. 4.
 4. 4. 0. 4. 0. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 2. 4. 4.
 4. 0. 4. 4. 4. 4. 4. 4. 4. 4. 2. 4. 4. 0. 4. 4. 4. 4. 4. 4. 0. 2. 4. 4.
 4. 4. 4. 2. 2. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 0. 4. 4. 4. 4. 4. 4. 4. 4.
 4. 4. 4. 4.]
test2
[5. 2. 2. 2. 4. 2. 2. 2. 5. 2. 2. 0. 2. 2. 0. 2. 5. 4. 2. 4. 2. 2. 2. 0.
 2. 2. 2. 2. 5. 0. 5. 2. 2. 4. 2. 2. 2. 2. 2. 5. 2. 5. 2. 2. 4. 4. 5. 2.
 2. 2. 4. 2. 2. 2. 2. 2. 2. 2. 4. 2. 2. 4. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 4. 0. 2. 2. 2. 5. 4. 2. 2. 2. 4. 2. 2. 4. 2. 2. 0. 5. 5. 2. 2. 5. 2. 2.
 2. 5. 4. 2.]
test3
[5. 5. 5. 5. 1. 5. 0. 5. 1. 5. 1. 2. 5. 5. 5. 1. 5. 5. 1. 5. 2. 1. 5. 5.
 0. 2. 5. 5. 5. 2. 1. 5. 1. 5. 1. 5. 5. 5. 5. 0. 5. 5. 5. 2. 5. 1. 5. 5.
 2. 5. 0. 5. 5. 5. 2. 5. 5. 1. 5. 5. 5. 1. 5. 5. 5. 5. 5. 5. 2. 0. 5. 1.
 5. 5. 5. 5. 5. 2. 1. 5. 1. 5. 5. 5. 5. 5. 5. 5. 5. 5. 1. 0. 2. 0. 5. 5.
 5. 2. 5. 5.]
```

```
test4
[1. 1. 1. 1. 1. 0. 3. 1. 1. 1. 1. 1. 1. 1. 1. 5. 5. 0. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 5. 1. 1. 1. 3. 1. 1. 1. 1. 1. 1. 1. 3. 1. 1. 1. 1. 1. 5.
 1. 3. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 5. 1. 1. 3.
 1. 5. 1. 3. 1. 1. 1. 3. 1. 1. 0. 5. 0. 1. 0. 1. 1. 1. 3. 1. 1. 1. 3. 1.
 1. 1. 5. 5.]
test5
[1. 3. 3. 3. 1. 3. 3. 3. 0. 3. 3. 3. 3. 3. 1. 3. 3. 3. 1. 3. 0. 3. 3. 0.
 3. 1. 3. 3. 3. 1. 3. 3. 1. 3. 1. 3. 3. 1. 0. 3. 3. 3. 0. 3. 0. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 1. 3. 3. 3.
 1. 1. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 3. 0. 1. 3. 3. 3. 1. 3. 3.
 3. 3. 3. 1.]
test6
[0. 0. 0. 0. 0. 0. 0. 0. 4. 4. 4. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4. 0. 0.
 0. 0. 0. 0. 0. 4. 0. 4. 0. 0. 4. 0. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4.
 4. 0. 4. 0. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4. 0. 4. 0. 4. 4. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4. 0. 0. 4. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

마무리 알게 된 점

1. 데이터의 선택이 얼마나 중요한지를 알게 되었다. 이번 프로젝트에서 처음 학습 데이터를 어떻게 뽑느냐에 따라서 분류의 성공과 실패가 결정되었다.

2. 데이터의 범위가 겹치더라도 겹치는 범위가 중간에서 형성된다거나 많이 겹치게 되면 학습의 결과가 의도와는 전혀 다르게 나온다는 것을 알게 되었다.

3. 아마 이번 프로젝트를 단순 거리비교로 분류하는 K-Means Clustering이 아닌 다른 더 정교하게 분류하는 모델을 사용했더라면 더 정확하게 분류할 수 있었을 것 같다.

마무리

감사합니다.

https://github.com/cheol-hoon/Numerical_Analysis