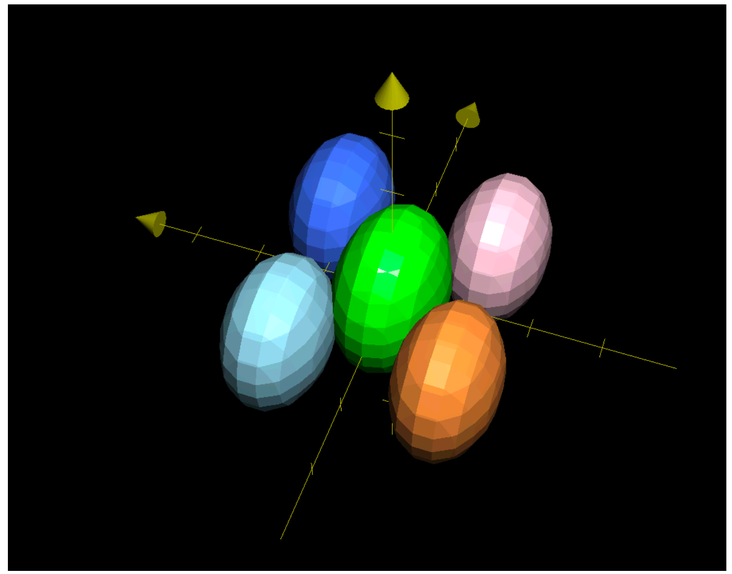
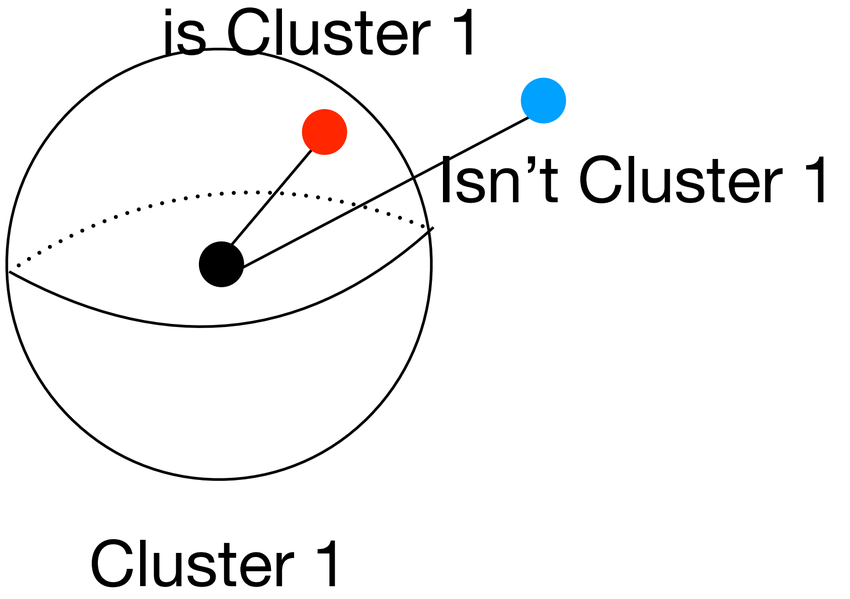
Term Project #2



2016026117 홍기범

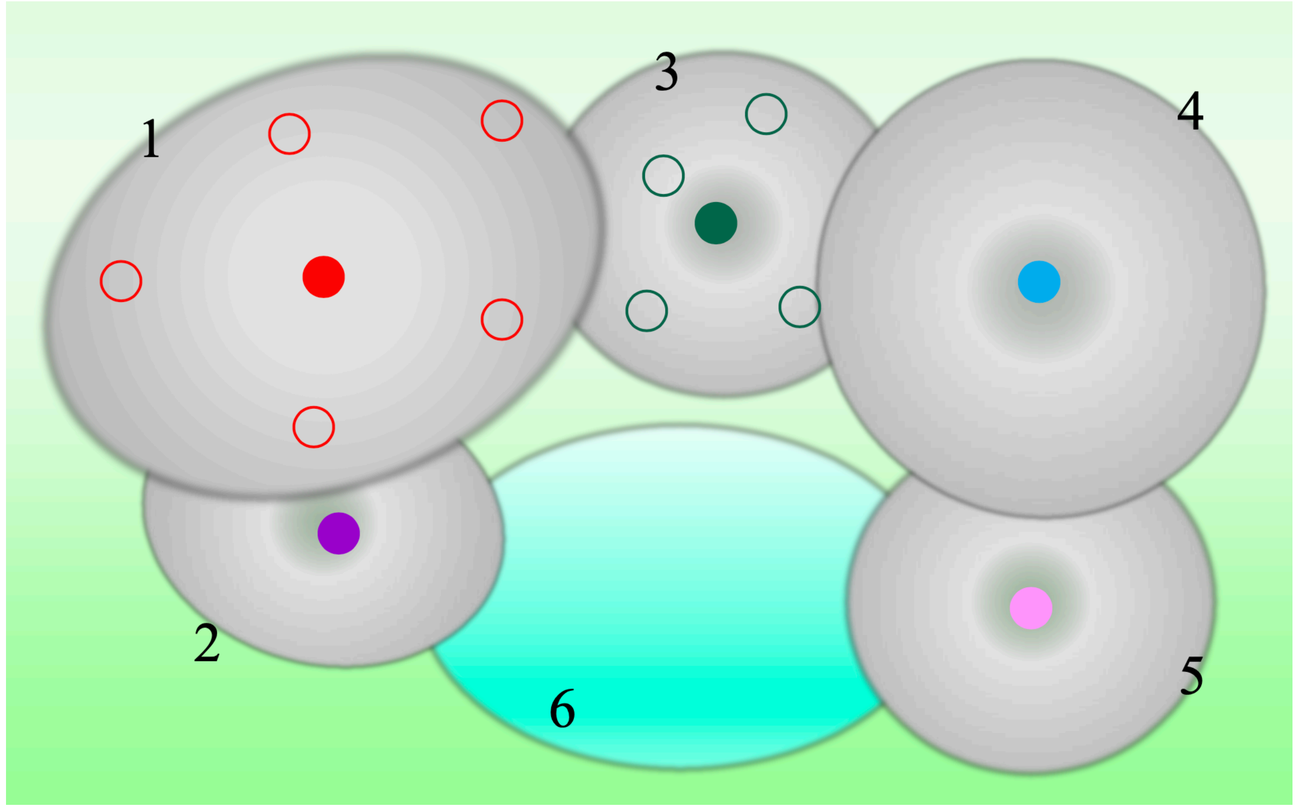
진행 과정

- Gaussian Distribution을 활용하여 서로 다른 5개의 Data Cluster를 만듭니다.
- 각각의 Cluster는 300개의 Data로 이루어져 있습니다.
- 이 때 만들어진, 총 1500개의 Data를 가지고 K - means Clustering을 활용하여 다시 Cluster를 구분해냅니다.
- 특정 cluster에 속하는 여부를 판단하기 위한 기준(최대치)을 설정합니다



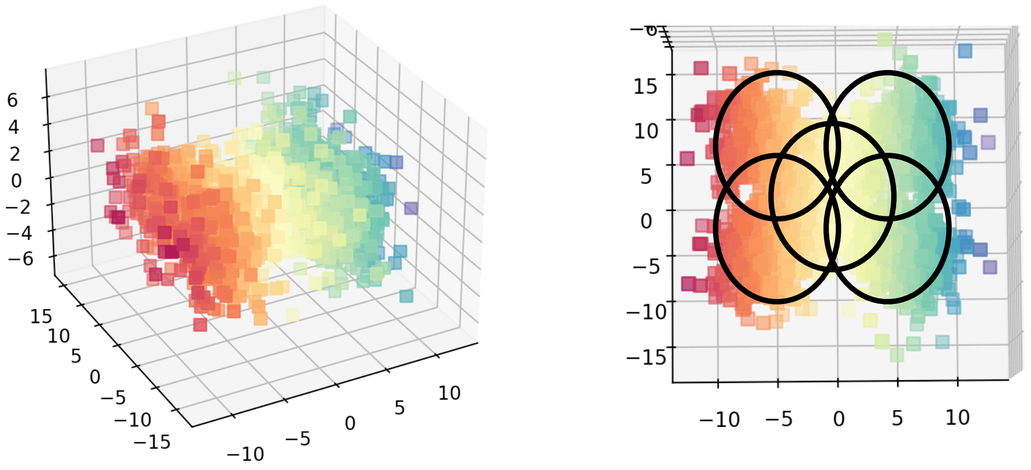
진행 과정

- 처음 Data Cluster를 구성하는데 사용했던 Gaussian Distribution을 이용해 다시 데이터를 만듭니다.
- 이 데이터가 K-means Clustering으로 구성된 각 cluster의 인식 기준에 부합하여 제대로 인식이 되는지 여부를 판단합니다.
- 각 Cluster마다 100개의 Data로 테스트를 진행합니다.
- 마지막으로, 기존 5개와 다른 Distribution을 구성하여 어떤 Cluster에도 속하지 않을 법한 Dataset을 만들고, 이를 기존의 5개의 cluster임을 판별하는 기준치에 대입하여 다른 cluster인지 여부를 판단합니다.

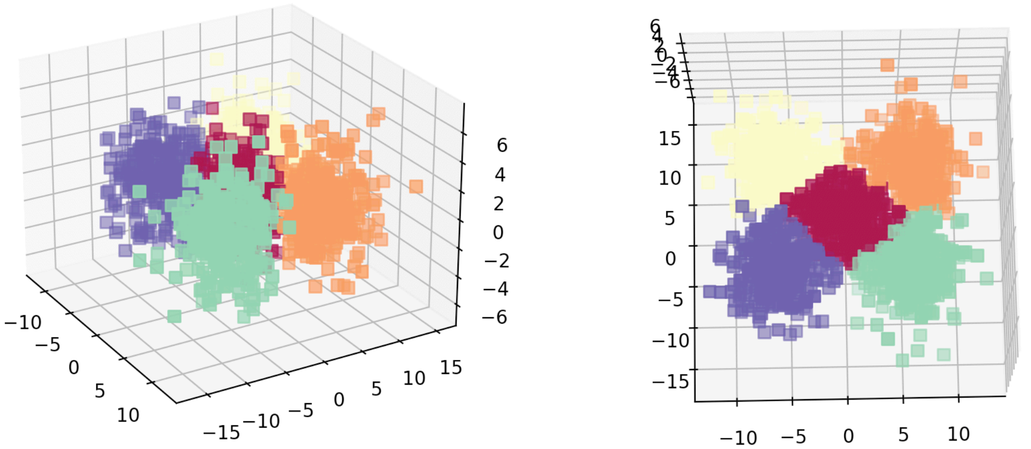


아이디어 및 방법

- 우선 Gaussian Distribution을 이용하여 Dataset을 만들어 냅니다.
- 5개의 cluster로 구분 되게끔 Data를 만들어 주어야 합니다.
- 각 cluster별로 300개씩 데이터를 만들어 줍니다.
- 또한 cluster간에 적절한 중첩을 허용하여 이상적이지 않은 상황에서도 K-Means clustering을 통해 dataset간의 구분을 수행하도록 합니다.



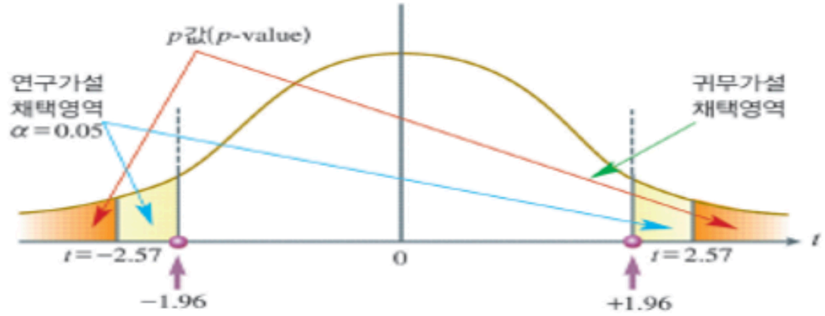
Clustering 전

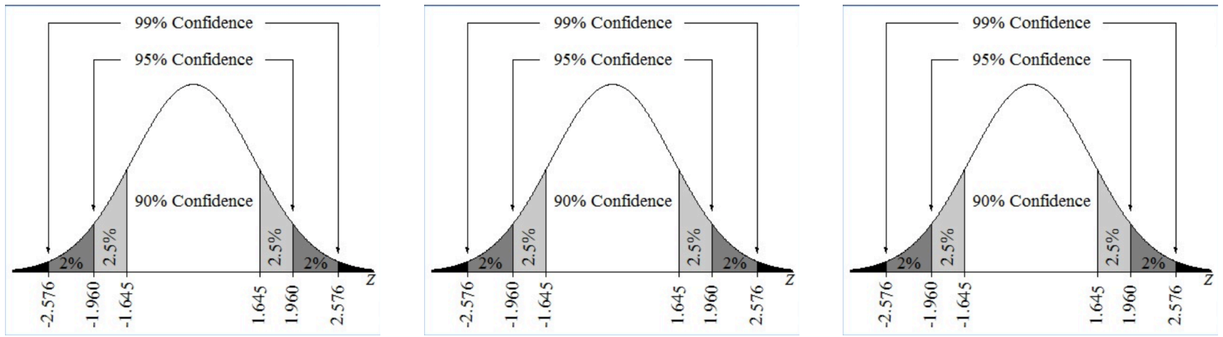


Clustering 후

아이디어 및 방법

- K-means Clustering를 통해 얻어낸 중심으로 부터 data가 얼마만큼 떨어져야 같은 cluster로 인정할 것인가에 대해 명확히 할 필요가 있습니다.
- Gaussian 분포를 활용해 데이터를 발생시켰습니다. 따라서 이 사실을 이용하여 clustering을 통해 생성된 데이터의 표준편차를 이용하여 해당 cluster임을 판단하는 허용구간을 설정할 수 있습니다.
- 우선 총 1500개의 data에 대해 clustering을 수행합니다. 이후 5개의 cluster로 재분류된 데이터 집단(cluster)에 대해서 각 축(x,y,z)별로 표준편차를 구합니다. 이후 같은 분포를 이용해 각 cluster에 대해 100개의 sample data을(총 500개)만들고, 이 5개의 cluster와 동떨어진 sample data 100개를 추가적으로 만들어냅니다.
- 이후 가설검정(two-tailed test)을 활용합니다.
- 각 축(x,y,z)별로 구한 5개의 cluster의 표준편차를 이용하여, 중심에 신뢰계수(c)*표준편차를 더하거나 뺀 값을 구간으로 설정합니다.



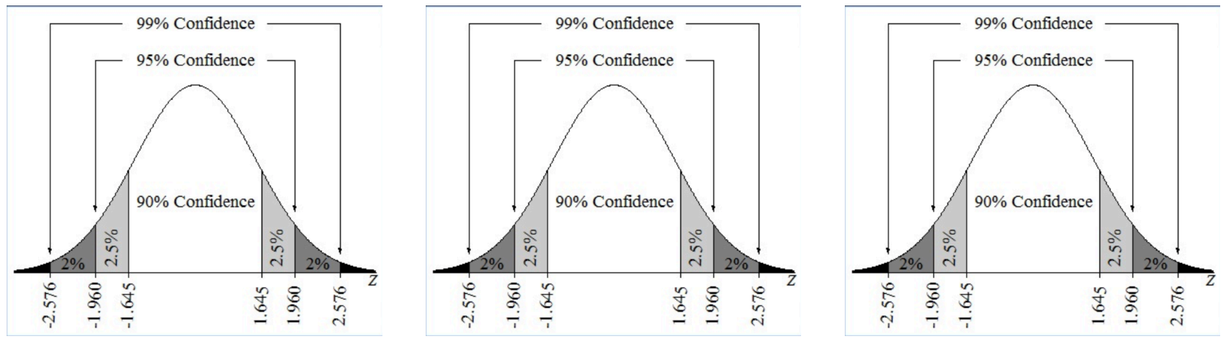


$$c\sigma_x \qquad c\sigma_y \qquad c\sigma_z$$

$$\text{sqrt}\left(c\sigma_x^2+c\sigma_y^2+c\sigma_z^2\right)$$

아이디어 및 방법

- 그리고 각 구간값을 제곱한 후 더한 뒤, 제곱근을 씌워주어 특정 cluster임을 판단하는 기준인 거리를 구합니다.
- Sample data와 cluster 중심사이의 거리와, 이 허용치를 비교하여 어느 cluster에 속한 데이터인지 판단합니다.
- 허용치 보다 중심과 sample data 사이의 거리가 작다면 그 데이터는 해당 cluster에 속하는 것으로 판별할 수 있습니다.
- 허용치를 벗어날 경우, 해당 cluster가 아닌 것으로 판단합니다.
- 다른 distribution을 활용해서 100개의 sample data를 만든 경우, 5개의 cluster coverage에서 벗어나게 되면 어떤 cluster에도 해당하지 않음을 표시하면 되겠습니다.



$c\sigma_x$ $c\sigma_y$ $c\sigma_z$

$\text{sqrt}(c\sigma_x^2+c\sigma_y^2+c\sigma_z^2)$

is Cluster 1

Isn't Cluster 1

D = $\text{sqrt}(c\sigma_x^2+c\sigma_y^2+c\sigma_z^2)$

Cluster 1

참고

사용한 Gaussian Distribution 정보

- Cluster 0 : (0,0,0), std_x = 2, std_y = 3, std_z = 2
- Cluster 1 : (6,6,0), std_x = 2, std_y = 3, std_z = 2
- Cluster 2 : (6,-6,0), std_x = 2, std_y = 3, std_z = 2
- Cluster 3 : (-6,6,0), std_x = 2, std_y = 3, std_z = 2
- Cluster 4 : (-6,-6,0), std_x = 2, std_y = 3, std_z = 2
- Other than 0-4(different distribution) : (10,10,10), std_x = 2, std_y = 3, std_z = 2

결과

신뢰계수 z = 2 일때

```
[[ 5.99336197 -5.87240953  0.11622023]
 [ 6.1598383   6.07770646  0.02653127]
 [-6.09376129  6.36412993 -0.26290163]
 [-5.85257252 -6.52712617  0.15996012]
 [-0.41227107 -0.01373835  0.02986681]]
```

K-means clustering을 통해 생성된 center, 순서대로 cluster 0-4에 대한 center임, K-means clustering 수행시 만들어지는 center들은 매번 순서가 바뀌므로 거리를 구할 때 순서에 대한 재보정 작업을 매번 거쳤음.

```
Cluster 0 Test :
classifed as cluster 0 : 96
classifed as cluster 1 : 1
classifed as cluster 2 : 0
classifed as cluster 3 : 0
classifed as cluster 4 : 2
classifed as nowhere : 1
```

```
Cluster 1 Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 97
classifed as cluster 2 : 0
classifed as cluster 3 : 0
classifed as cluster 4 : 2
classifed as nowhere : 1
```

```
Cluster 2 Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 0
classifed as cluster 2 : 93
classifed as cluster 3 : 0
classifed as cluster 4 : 7
classifed as nowhere : 0
```

```
Cluster 3 Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 0
classifed as cluster 2 : 2
classifed as cluster 3 : 92
classifed as cluster 4 : 5
classifed as nowhere : 1
```

```
Cluster 4 Test :
classifed as cluster 0 : 7
classifed as cluster 1 : 2
classifed as cluster 2 : 5
classifed as cluster 3 : 6
classifed as cluster 4 : 80
classifed as nowhere : 0
```

```
Free Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 2
classifed as cluster 2 : 0
classifed as cluster 3 : 0
classifed as cluster 4 : 0
classifed as nowhere : 98
```

Free Test : 10,10,10에서 발생시킨, 기존 5개 distribution과 다른 distribution에서 data를 발생시킨 후 cluster간의 포함 여부 체크

Nowhere : 0-4 중 어떤곳에도 포함되지 않음

결과

신뢰계수 z = 2.58 일때

```
[[-5.83207399 -6.08222135 -0.12052623]
 [-6.08799135  6.04716882 -0.13044885]
 [ 0.1056193   0.07663232  0.1180949 ]
 [ 5.75664425 -6.53226818 -0.04110003]
 [ 5.97250306  6.07093386 -0.02123285]]
```

```
Cluster 0 Test :
classifed as cluster 0 : 96
classifed as cluster 1 : 1
classifed as cluster 2 : 3
classifed as cluster 3 : 0
classifed as cluster 4 : 0
classifed as nowhere : 0
```

```
Cluster 1 Test :
classifed as cluster 0 : 5
classifed as cluster 1 : 89
classifed as cluster 2 : 6
classifed as cluster 3 : 0
classifed as cluster 4 : 0
classifed as nowhere : 0
```

```
Cluster 2 Test :
classifed as cluster 0 : 2
classifed as cluster 1 : 5
classifed as cluster 2 : 84
classifed as cluster 3 : 4
classifed as cluster 4 : 5
classifed as nowhere : 0
```

```
Cluster 3 Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 0
classifed as cluster 2 : 6
classifed as cluster 3 : 94
classifed as cluster 4 : 0
classifed as nowhere : 0
```

```
Cluster 4 Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 0
classifed as cluster 2 : 5
classifed as cluster 3 : 0
classifed as cluster 4 : 95
classifed as nowhere : 0
```

```
Free Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 0
classifed as cluster 2 : 0
classifed as cluster 3 : 0
classifed as cluster 4 : 25
classifed as nowhere : 75
```

결과

신뢰계수 z = 3일때

```
[[-0.05396278 -0.04341564  0.06230225]
 [-6.01650329  6.05369618  0.17037395]
 [-6.17639183 -6.20804282 -0.01942742]
 [ 6.08148402 -6.29646001  0.15148531]
 [ 5.97134543  6.35573854 -0.14339374]]
```

```
Cluster 0 Test :
classifed as cluster 0 : 89
classifed as cluster 1 : 4
classifed as cluster 2 : 3
classifed as cluster 3 : 3
classifed as cluster 4 : 1
classifed as nowhere : 0
```

```
Cluster 1 Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 100
classifed as cluster 2 : 0
classifed as cluster 3 : 0
classifed as cluster 4 : 0
classifed as nowhere : 0
```

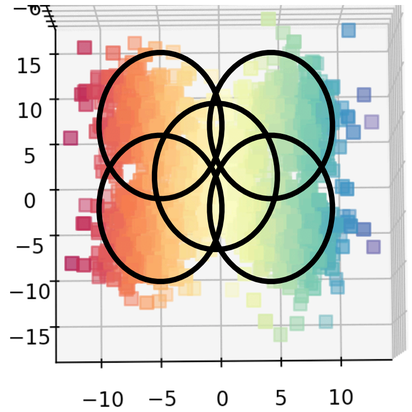
```
Cluster 2 Test :
classifed as cluster 0 : 7
classifed as cluster 1 : 1
classifed as cluster 2 : 92
classifed as cluster 3 : 0
classifed as cluster 4 : 0
classifed as nowhere : 0
```

```
Cluster 3 Test :
classifed as cluster 0 : 2
classifed as cluster 1 : 0
classifed as cluster 2 : 0
classifed as cluster 3 : 98
classifed as cluster 4 : 0
classifed as nowhere : 0
```

```
Cluster 4 Test :
classifed as cluster 0 : 6
classifed as cluster 1 : 0
classifed as cluster 2 : 0
classifed as cluster 3 : 0
classifed as cluster 4 : 94
classifed as nowhere : 0
```

```
Free Test :
classifed as cluster 0 : 0
classifed as cluster 1 : 0
classifed as cluster 2 : 0
classifed as cluster 3 : 0
classifed as cluster 4 : 52
classifed as nowhere : 48
```

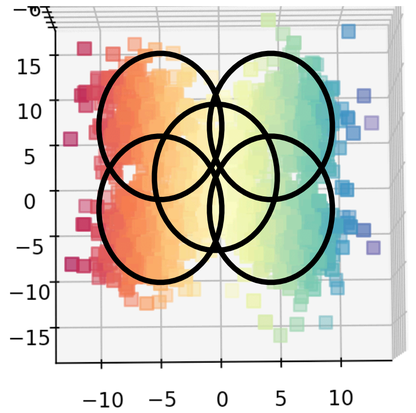
결과 분석



- (0,0,0)에서 발생시킨 data의 경우, 나머지 4개의 cluster와 모두 겹치는 경향이 있어 제일 인식률이 떨어지는 경향이 있습니다. (0,0,0)을 중심으로한 cluster임을 86%정도의 정확도로 판별하였고, 나머지는 4개의 cluster에 나누어서 인식이 되는 것을 확인할 수 있습니다. 이는 처음 5개의 분포를 세팅을 할 때 미리 예상한 결과이기도 합니다.

- 나머지의 경우 평균 95%정도의 정확도를 보였습니다. 다만 신뢰계수를 다르게 하였을 때 결과값이 조금씩 달라지는데, 신뢰계수가 커질수록 cluster 0-4에서 발생시킨 sample data의 인식률은 커지나, 전혀 다른 분포를 만든 뒤 발생시킨 data cluster가 cluster 0-4 와 전혀 다른 cluster임을 인식하는 정확도는 떨어지는 경향을 보였습니다.

결과 분석



- 이는 신뢰계수를 높임으로써 발생하는 일종의 tradeoff로 판단됩니다. 신뢰계수를 높이면 cluster 0-4를 판별하는 허용치가 높아지게 되고, 이에따라 cluster 0-4에 속하게 되는 data의 수가 늘어나지만, 한편으로는 어느 cluster에도 속하지 않아야할 data들이 cluster 0-4에 속하게 되면서 cluster 0-4의 data와 전혀 다른 distribution에서 발생시킨 data를 구분하는 능력은 떨어질 것입니다.

- 따라서 cluster 0-4이외의 cluster를 명확히 구분하고 싶으면 신뢰계수를 낮추고, cluster 0-4의 인식률을 높이고 싶으면 신뢰계수를 높여 작업을 진행하면 되겠습니다.

- 다만 신뢰계수를 너무 키우게 되면 cluster 0-4 사이의 구분이 모호해 질 수 있으므로, 너무 작게하면 cluster의 모든 데이터를 포함할 수 없으므로 이 점을 염두해 두어야 합니다.