T-Academy

Lab_02) Clustering

2019. Apr. 25.
SK플래닛 T아카데미
캐글 코리아
강천성

Clustering

클러스터링은 주어진 데이터들의 특성을 고려해 데이터 클러스터를 정의하고, 클러스터를 대표할 수 있는 대표점을 찾는 비지도 학습의 대표적인 알고리즘 입니다.
간단히 말해서, 비슷한 특성을 가진 데이터끼리 묶는다고 말할 수 있습니다.

1. k-Means 클러스터링

k-means 클러스터링은 대표적인 클러스터링 알고리즘 중 하나로, 각 클러스터에 할당된 데이터 포인트들의 평균 좌표를 이용해 중심점을 반복적으로 업데이트하며 클러스터를 형성하는 알고리즘 입니다.

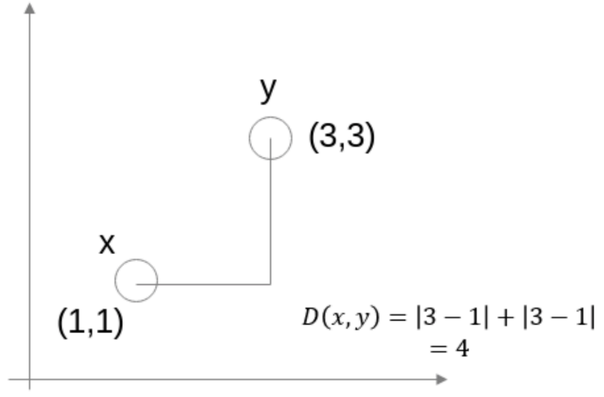
k-means Clustering 알고리즘은 3가지 단계로 이루어집니다.
Step 1. 각 데이터 포인트 i 에 대해 가장 가까운 중심점을 찾고, 그 중심점에 해당하는 클러스터를 할당합니다.
가까운 중심점을 찾을 때는, **유클리드 거리**를 사용합니다.
Step 2. 할당된 클러스터를 기반으로 새로운 중심점을 계산합니다. 중심점은 클러스터 내부 점들 좌표의 **산술 평균(mean)** 으로 합니다.
Step 3. 각 클러스터의 할당이 바뀌지 않을 때까지 반복합니다.

점과 점사이의 거리를 어떻게 측정할 수 있을까?

k-means clustering은 거리 기반 알고리즘이므로 여러가지 방법으로 거리를 측정할 수 있습니다.

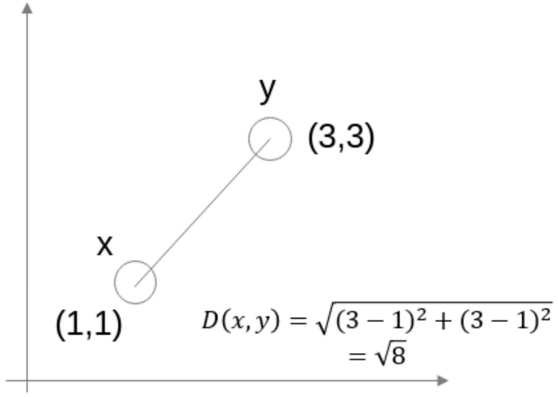
1. Manhattan Distance - 각 축에 대해 수직으로만 이동하여 계산하는 거리 측정방식

$$D(x, y) = \sum_{i=1}^{d} |x_i - y_i|$$



2. Euclidean Distance - 점과 점사이의 가장 짧은 거리를 계산하는 거리 측정방식

$$D(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$



• k-Means Clustering

1. 모델 불러오기 및 정의

```python
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3)
```

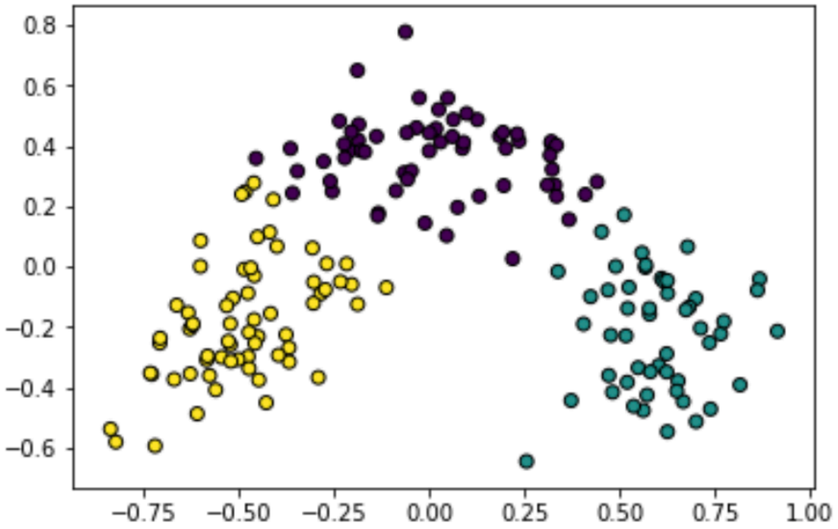
2. fit

```python
kmeans.fit(data)
```

3. predict

```python
cluster = kmeans.predict(data)
```

4. 결과 확인



2. Hierarchical Clustering

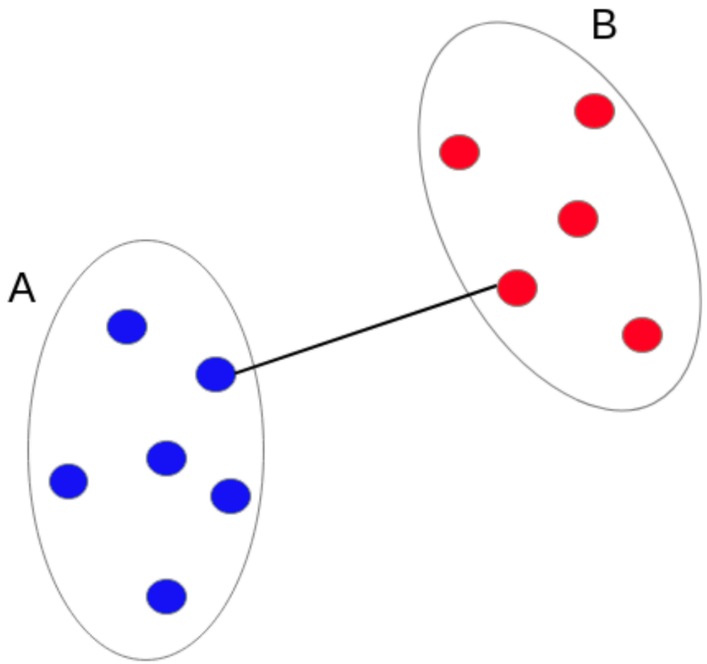
Hierarchical Clustering은 거리(Distance) 또는 유사도(Similarity)를 기반으로 클러스터를 형성하는 알고리즘 입니다.
k-means Clustering과 다르게 클러스터의 수를 설정해 줄 필요가 없으며, 클러스터 형태를 시각적으로 표현해주는 덴드로그램을 통해 적절한 클러스터의 수를 선택할 수 있습니다.
Hierachichal Clustering에는 Bottom-Up 방식의 Agglomerative Method와 Top-Down 방식의 Divisive Method로 나뉩니다.
이번 단원에서는 Agglomerative Method를 사용해 실습을 진행합니다.

Agglomerative Method를 사용한 Hierarchical Clustering 알고리즘은 3가지 단계로 이루어집니다.
Step 1. 각 데이터 포인트를 클러스터로 할당합니다. (n개의 클러스터)
Step 2. 가까운 클러스터끼리 병합합니다.
Step 3. 1개의 클러스터가 될 때까지 반복합니다.

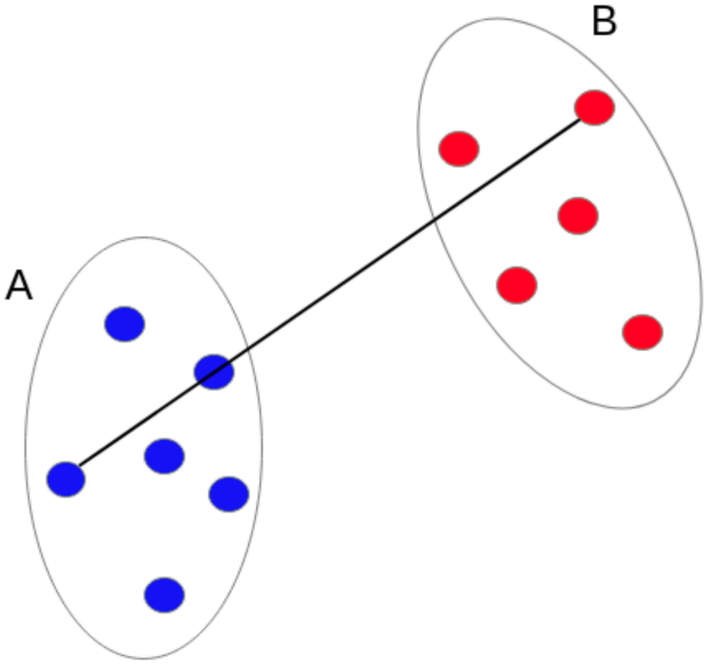
어떻게 가장 가까운 클러스터를 찾을 수 있을까?

방금전 거리 측정 방법으로 맨하탄 거리, 유클리디언 거리에 대해 배웠었습니다.
k-means에서는 각 클러스터의 중심점 간의 거리로 클러스터간 거리를 계산했었습니다.
이번 수업에서는 새로운 클러스터간 거리를 계산하는 방법에 대해 알아보겠습니다.

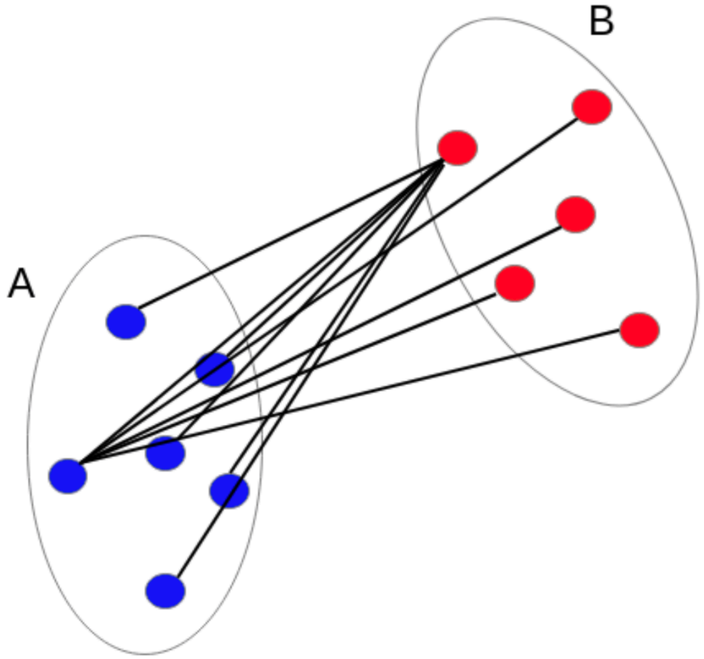
1. Single Linkage - 두 클러스터 내의 가장 가까운 점 사이의 거리



2. Complete Linkage - 두 클러스터 내의 가장 먼 점 사이의 거리



3. Average Linkage - 두 클러스터 내의 모든 점 사이의 평균 거리



• Hierarchical Clustering(Bottom-Up, Single Linkage)

1. 모델 불러오기 및 정의

```
from sklearn.cluster import AgglomerativeClustering
single_clustering = AgglomerativeClustering(linkage='single', n_clusters=3)
```

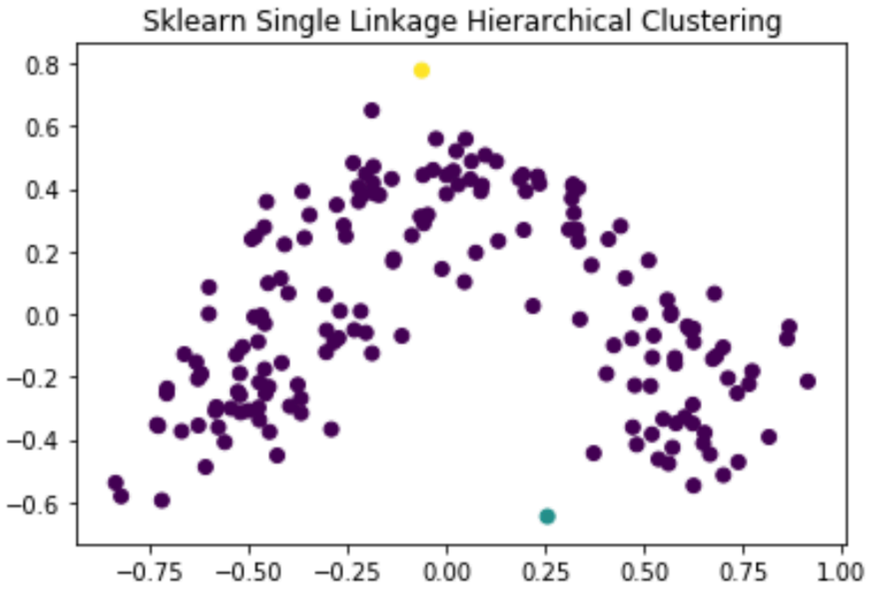
2. fit

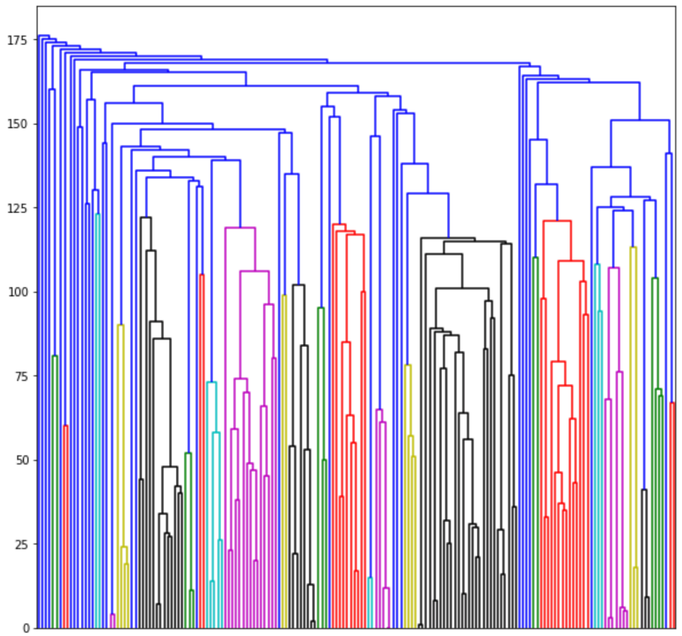
```
single_clustering.fit(data)
```

3. predict

```
single_cluster = single_clustering.labels_
```

4. 결과 확인





• Hierarchical Clustering(Bottom-Up, Complete Linkage)

1. 모델 불러오기 및 정의

```
complete_clustering = AgglomerativeClustering(linkage='complete', n_clusters=3)
```

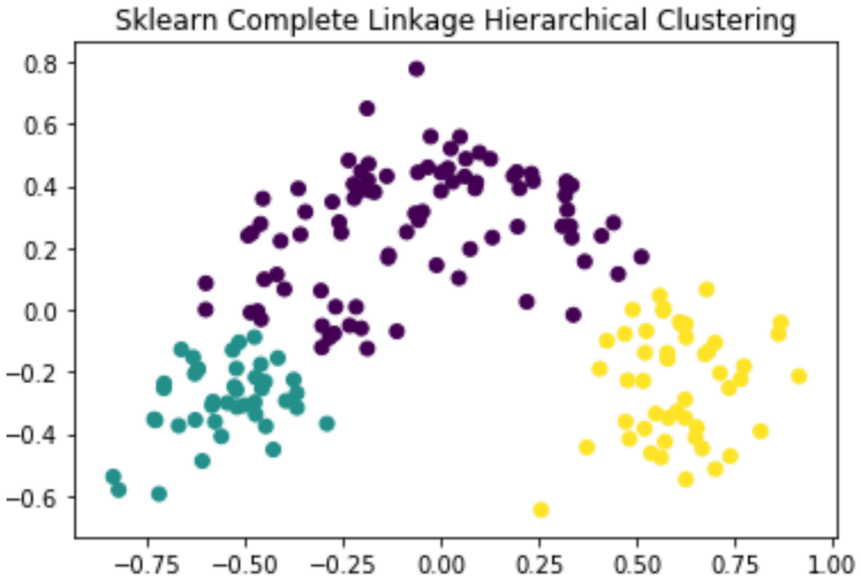
2. fit

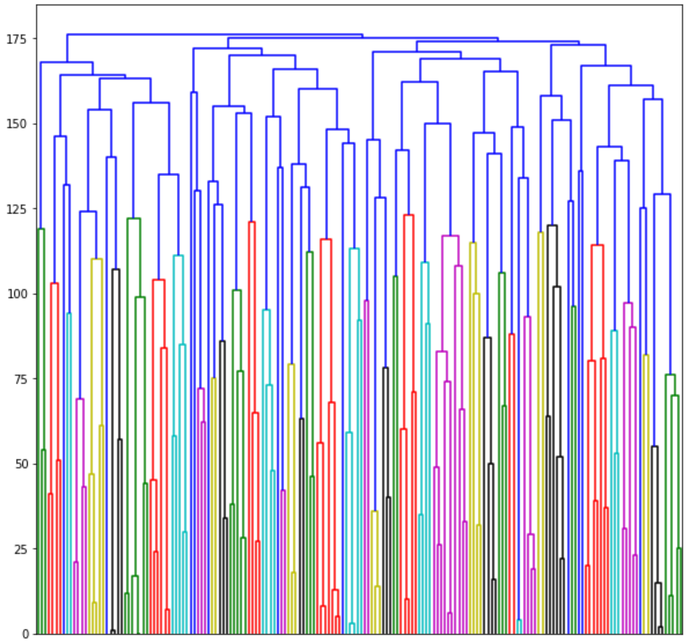
```
complete_clustering.fit(data)
```

3. predict

```
complete_cluster = complete_clustering.labels_
```

4. 결과 확인





• Hierarchical Clustering(Bottom-Up, Average Linkage)

1. 모델 불러오기 및 정의

```
average_clustering = AgglomerativeClustering(linkage='average', n_clusters=3)
```

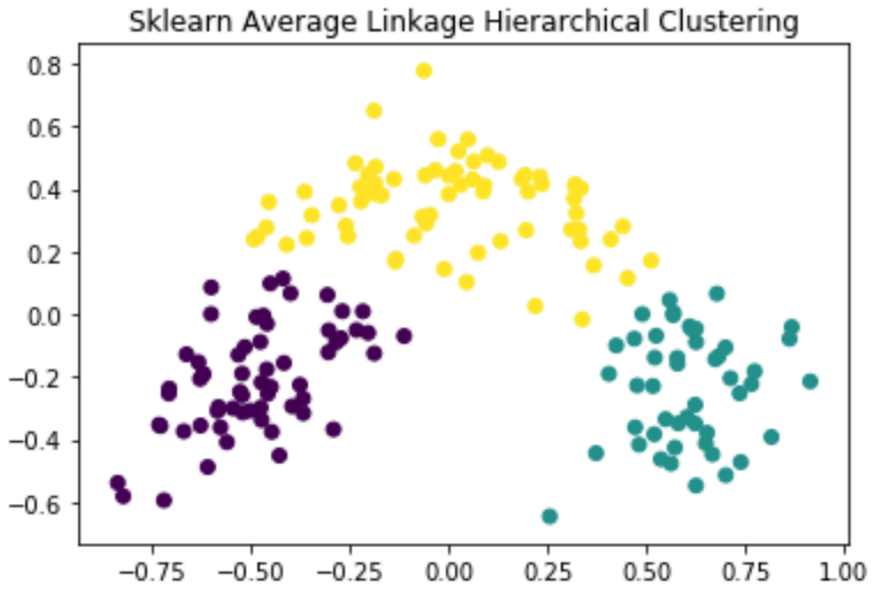
2. fit

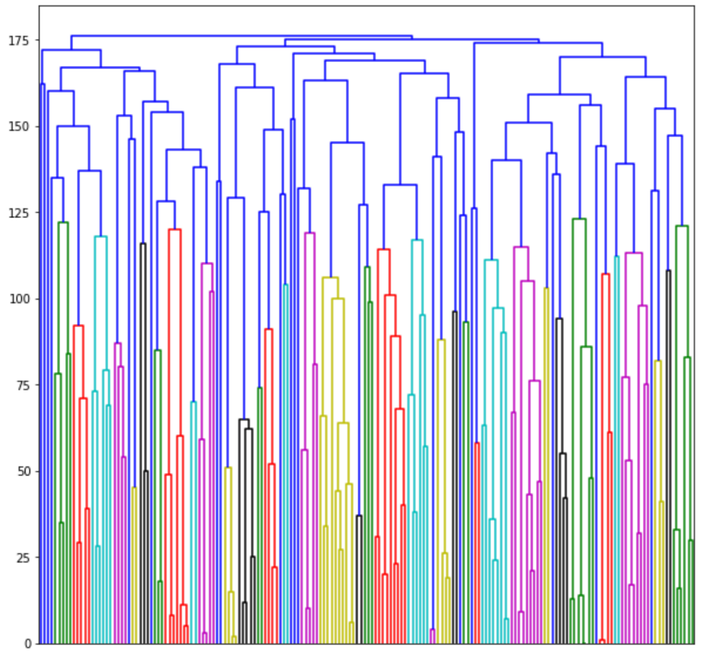
```
average_clustering.fit(data)
```

3. predict

```
average_cluster = average_clustering.labels_
```

4. 결과 확인





Evaluation

Silhouette

- 실루엣 값은 한 클러스터 안의 데이터들이 다른 클러스터와 비교해서 얼마나 비슷한가를 나타냅니다.
- 같은 클러스터 내의 점들간 거리는 가깝고(cohesion) 서로 다른 클러스터 간의 거리는 멀수록(separation) 높은 값을 얻을 수 있습니다.
- 실루엣 값이 1에 근접한다는 것은 같은 클러스터 내의 평균거리가 다른 클러스터와의 평균거리보다 가깝다는 것을 의미합니다.
- 일반적으로 실루엣 값이 0.5보다 크다면 데이터가 잘 클러스터링 되었다는 것을 나타냅니다.

실루엣 공식은 다음과 같습니다.

$$S_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$$

a_i : 같은 클러스터 내의 모든 점들 간 평균 거리

b_i : $\bar{d}$ = (i, c)의 최솟값

$\bar{d}$ = (i, c) : 다른 클러스터 c와 i번째데이터와의 평균 거리

직관적으로 수식을 이해해보겠습니다. a_i는 같은 클러스터 내의 데이터 들이 잘 모여있다면 적은 값을 나타내고, b_i는 각 클러스터들이 멀리 떨어져있다면 큰 값을 나타내게 됩니다.
따라서 수식 S_i에 따르면, 아주 잘 형성된(같은 클러스터는 가깝고 다른 클러스터끼리는 먼) 클러스터 형태일 때 분모는 b_i이 되고,
분자는 b_i에서 아주 작은 값인 a_i가 빠져 1에 가까운 실루엣 값을 얻을 수 있습니다.

가장 좋은 클러스터를 형성하는 클러스터의 수를 찾아보자

k-means 클러스터링과 Average Linkage를 사용한 Hierarchical 클러스터링에서 가장 높은 점수의 클러스터 수는 무엇인지 알아보겠습니다.

Silhouette 스코어링은 Sklearn의 metrics 패키지에 있습니다.

- **Silhouette, Hierarchical clustering**

```
from sklearn.metrics import silhouette_score

best_n = 1
best_score = -1

for n_cluster in range(2, 11):
    kmeans = KMeans(n_clusters=n_cluster)
    kmeans.fit(data)
    cluster = kmeans.predict(data)
    score = silhouette_score(data, cluster)

    print('클러스터의 수 : {}, 실루엣 점수 : {:.2f}'.format(n_cluster, score))
    if score > best_score :
        best_n = n_cluster
        best_score = score

print('가장 높은 실루엣 점수를 가진 클러스터 수 : {}, 실루엣 점수 : {:.2f}'.format(best_n, best_score))
```

```
클러스터의 수 : 2, 실루엣 점수 : 0.49
클러스터의 수 : 3, 실루엣 점수 : 0.57
클러스터의 수 : 4, 실루엣 점수 : 0.49
클러스터의 수 : 5, 실루엣 점수 : 0.42
클러스터의 수 : 6, 실루엣 점수 : 0.42
클러스터의 수 : 7, 실루엣 점수 : 0.40
클러스터의 수 : 8, 실루엣 점수 : 0.40
클러스터의 수 : 9, 실루엣 점수 : 0.39
클러스터의 수 : 10, 실루엣 점수 : 0.39
가장 높은 실루엣 점수를 가진 클러스터 수 : 3, 실루엣 점수 : 0.57
```

- **Silhouette, Hierarchical clustering**

```
from sklearn.metrics import silhouette_score

best_n = 1
best_score = -1

for n_cluster in range(2, 11):
    average_clustering = AgglomerativeClustering(n_clusters=n_cluster, linkage='average')
    average_clustering.fit(data)
    cluster = average_clustering.labels_
    score = silhouette_score(data, cluster)

    print('클러스터의 수 : {}, 실루엣 점수 : {:.2f}'.format(n_cluster, score))
    if score > best_score :
        best_n = n_cluster
        best_score = score

print('가장 높은 실루엣 점수를 가진 클러스터 수 : {}, 실루엣 점수 : {:.2f}'.format(best_n, best_score))
```

```
클러스터의 수 : 2, 실루엣 점수 : 0.49
클러스터의 수 : 3, 실루엣 점수 : 0.56
클러스터의 수 : 4, 실루엣 점수 : 0.48
클러스터의 수 : 5, 실루엣 점수 : 0.42
클러스터의 수 : 6, 실루엣 점수 : 0.37
클러스터의 수 : 7, 실루엣 점수 : 0.34
클러스터의 수 : 8, 실루엣 점수 : 0.34
클러스터의 수 : 9, 실루엣 점수 : 0.37
클러스터의 수 : 10, 실루엣 점수 : 0.33
가장 높은 실루엣 점수를 가진 클러스터 수 : 3, 실루엣 점수 : 0.56
```

• Reference

- Wikipedia, Clustering : https://ko.wikipedia.org/wiki/클러스터_분석
- Wikipedia, Manhattan distance : https://ko.wikipedia.org/wiki/맨해튼_거리
- Wikipedia, Euclidean distance : https://ko.wikipedia.org/wiki/유클리드_거리
- Sklearn, Wine dataset : https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html
- Sklearn, k-Means Clustering : https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- Sklearn, Hierarchical Clustering : https://www.google.com/url?q=http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html&sa=U&ved=0ahUKEwj_2aiGvt7hAhXLi7wKHei8CNsQFggEMAA&client=internal-uds-cse&cx=016639176250731907682:tjtqbvtvij0&usg=AOvVaw0zVZAVTxgORo-7LbgrNv_o
- Sklearn, Silhouette : https://www.google.com/url?q=http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html&sa=U&ved=0ahUKEwi5lrTZwd7hAhUqCqYKHWCZCTEQFggEMAA&client=internal-uds-cse&cx=016639176250731907682:tjtqbvtvij0&usg=AOvVaw0-ZT8AJZRmR-qNpN-62Ei-
- Sklearn, Silhouette Example : https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py
- Scipy, Dendrogram : https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.cluster.hierarchy.dendrogram.html