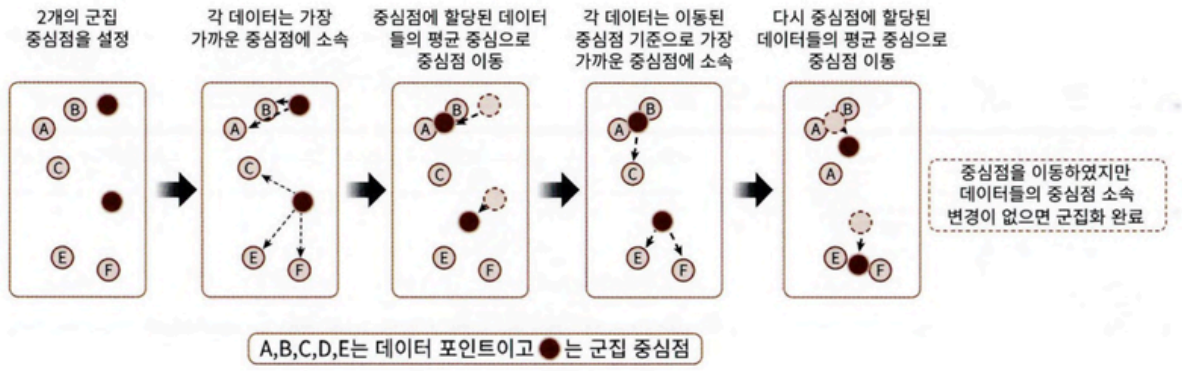
Week11_군집화(Clustering)_7장

K-평균 알고리즘

군집 중심점(centroid)라는 특정한 임의의 지점을 선택해 **해당 중심에 가장 가까운 포인트**들을 선택하는 군집화 기법

K-mean clustering 동작 과정



1. 군집 중심점 초기화: 구성하려는 군집 개수만큼 (적합한) 임의의 위치에 중심점을 가져다 놓기
2. 데이터 라벨링: 각 데이터를 데이터에서 가장 가까운 중심점에 소속시킴 (중심점 기준으로 군집화)
3. 군집 중심점 이동: 군집 중심점을 결정된 소속 데이터들의 평균 지점으로 이동시킴
4. 변경된 중심점을 기준으로 다시 데이터 라벨링
5. 군집 중심점 이동: 변경된 소속 데이터들의 평균 지점으로 이동
6. ...
7. 중심점 이동 후 데이터의 소속 변경이 이루어지지 않을 때까지 반복한 뒤 군집화 종료

⇒ **군집 중심점 설정 & 데이터 라벨링** 의 반복

장점

- 일반적인 군집화에서 가장 많이 활용되는 알고리즘
- 쉽고 간편

단점

- 거리 기반 알고리즘 (라벨링 기준이 '가까움'임) → 속성 개수가 너무 많을 경우 군집화 정확도 감소
  (→ PCA로 차원 감소 해야 할 수도 있음)
- 반복 횟수가 많을 경우 수행 시간 매우 느려짐
- 몇 개(k)의 군집(cluster)을 선택할지 가이드하기 어려움

사이킷런 KMeans 클래스

```
class sklearn.cluster.KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001,
                             precompute_distances='auto', verbose=0, random_state=None,
                             copy_x=True, n_jobs=1, algorithm='auto')
```

- n_clusters: 군집화할 개수 = 군집 중심점의 개수
- init: 초기에 군집 중심점의 좌표를 설정할 방식. 디폴트는 k-means++ 방식
- max_iter: 최대 반복 횟수 (이 횟수 이전에 반복 종료되면 군집화 종료)

KMeans 객체

- **fit(**데이터 세트**), fit_transform(**데이터 세트**)** 로 군집화 수행
- 군집화 완료된 KMeans 객체의 **labels_**, **cluster_centers_** 속성으로 군집화 관련 주요 속성 파악 가능
  - labels_: 각 데이터 포인트가 속한 군집 중심점 레이블
  - cluster_centers_: 각 군집 중심점 좌표 *[군집 개수, 피처 개수]* → 군집 중심점 좌표 시각화 가능

사이킷런 군집화용 데이터 생성기 make_blobs(), make_classification() API

두 생성기 모두 여러 클래스에 해당하면서, 하나의 클래스에 여러 군집이 분포될 수 있게 데이터 생성 가능

- make_blobs( ): 개별 군집의 중심점과 표준 편차 제어 기능 추가

- make_classification(): 노이즈 포함 데이터 생성 가능

+) make_circle(), make_moon(): 중심 기반 군집화로 해결하기 어려운 데이터 세트 만드는 데 사용됨

make_blobs()

피처 데이터 세트와 타깃 데이터 세트가 튜플로 반환됨

- n_samples: 생성할 총 데이터 개수. default=100
- n_features: 데이터의 피처 개수
- centers: int - 군집 개수 / ndarray - 개별 군집 중심점의 좌표
- cluster_std: 생성될 군집 데이터의 표준 편차
  - float - 군집 내에서 해당 표준편차를 가진 값 생성
  - float형 리스트 - 군집의 각 표준편차 지정

군집 평가(Cluster Evaluation)

⚠️ 대부분의 군집화 데이터 세트는 군집화 결과와 비교할 만한 타깃 레이블을 가지고 있지 않음

⚠️ 군집화와 분류의 차이

- 데이터 내에 숨어 있는 별도의 그룹을 찾아 의미 부여
- 동일한 분류 값 → 그 안에서 더 세분화된 군집화를 추구하거나 서로 다른 분류 값의 데이터도 더 넓은 군집화 레벨화 등의 영역을 가짐

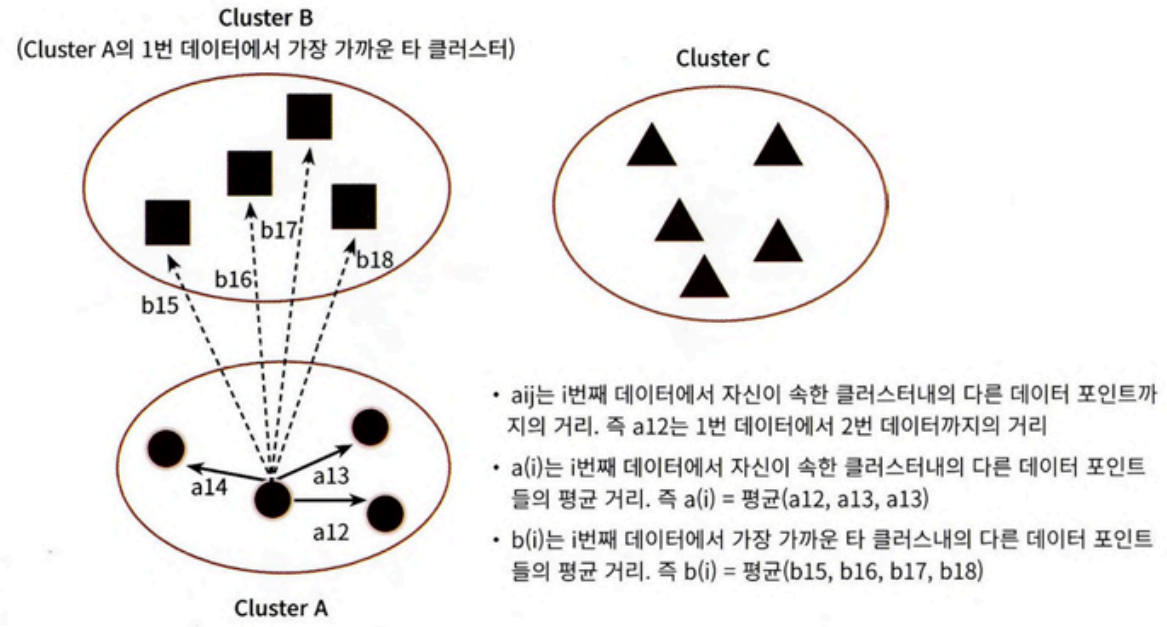
실루엣 분석 (silhouette analysis)

각 군집 간의 거리가 얼마나 효율적으로 분리돼 있는지를 나타내는 군집화 평가 방법

> “잘 분리되었다”
>
> = 다른 군집과의 거리는 떨어져 있고 동일 군집끼리의 데이터는 서로 가깝게 잘 뭉쳐 있다

실루엣 계수

: 개별 데이터가 같은 군집 내의 데이터와 얼마나 가깝게 군집화돼 있고, 다른 군집에 있는 데이터와는 얼마나 멀리 분리돼 있는지를 나타내는 지표



- a(i): 해당 데이터 & 같은 군집 내에 있는 다른 데이터 포인트 거리를 평균한 값
- b(i): 해당 데이터 & 다른 군집 중 가장 가까운 군집과의 평균 거리

실루엣 거리 계산

두 군집 간 거리가 얼마나 떨어져 있는가 b(i) - a(i)

→ 정규화 MAX(a(i), b(i))

⇒ i 번째 데이터 포인트의 실루엣 계수 값 s(i)

$$s(i) = \frac{(b(i) - a(i))}{(\max(a(i), b(i))}$$

- -1 ~ 1 사이의 값
- 1로 가까워질수록 근처 군집과 더 멀리 떨어져 있다는 뜻
- 0으로 가까워질수록 근처 군집과 가까워진다는 뜻

- - 값 = 다른 군집에 데이터 포인트가 할당되었다는 뜻

사이킷런 실루엣 분석을 통한 군집 평가

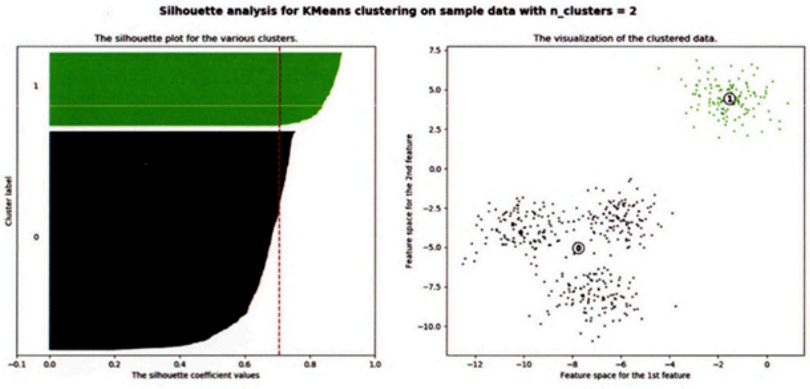
사이킷런 실루엣 분석 메소드

- 각 데이터 포인트의 실루엣 계수 계산 - **silhouette_samples**

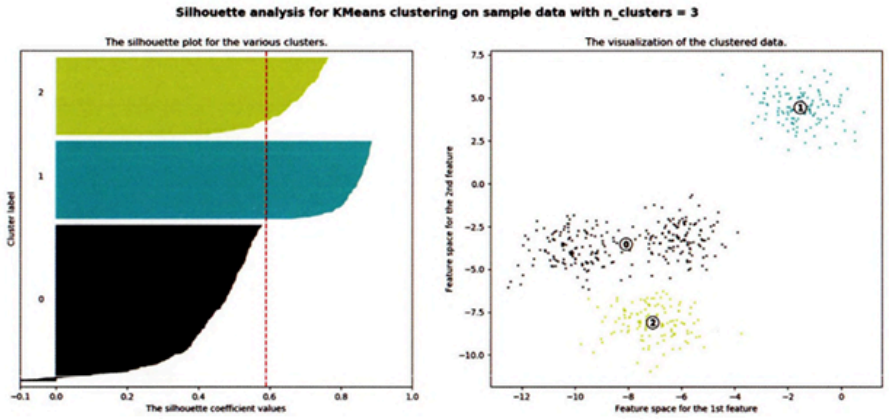
sklearn.metrics.silhouette_samples(*X, labels, metric='euclidean', **kwds*): 인자로 X feature 데이터 세트와 각 피처 데이터 세트가 속한 군집 레이블 값인 labels 데이터를 입력해주면 각 데이터 포인트의 실루엣 계수를 계산해 반환합니다.

- 전체 데이터의 실루엣 계수 값 평균해 반환 - **silhouette_score**

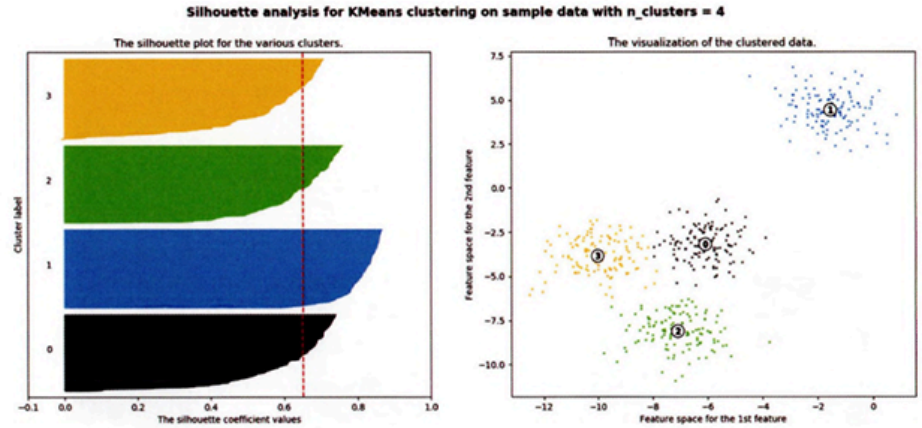
sklearn.metrics.silhouette_score(*X, labels, metric='euclidean', sample_size=None, **kwds*): 인자로 X feature 데이터 세트와 각 피처 데이터 세트가 속한 군집 레이블 값인 labels 데이터를 입력해주면 전체 데이터의 실루엣 계수 값을 평균해 반환합니다. 즉, np.mean(silhouette_samples())입니다. 일반적으로 이 값이 높을수록 군집화가 어느 정도 잘 됐다고 판단할 수 있습니다. 하지만 무조건 이 값이 높다고 해서 군집화가 잘 됐다고 판단할 수는 없습니다.



군집이 2개일 경우 평균 실루엣 계수 값: 0.704



군집이 3개일 경우 평균 실루엣 계수 값: 0.588



군집이 4개일 경우 평균 실루엣 계수 값: 0.65

→ 실루엣 계수를 통한 K-평균 군집 평가 방법

- 직관적 이해 쉬움
- 각 데이터별로 다른 데이터와의 거리 반복적으로 계산: 계산량 ⬆️
- 사이킷런 실루엣 계수 평가 API를 개인용 PC에서 수행할 경우 메모리 부족 등 에러 발생할 수 O

평균 이동 (Mean Shift)

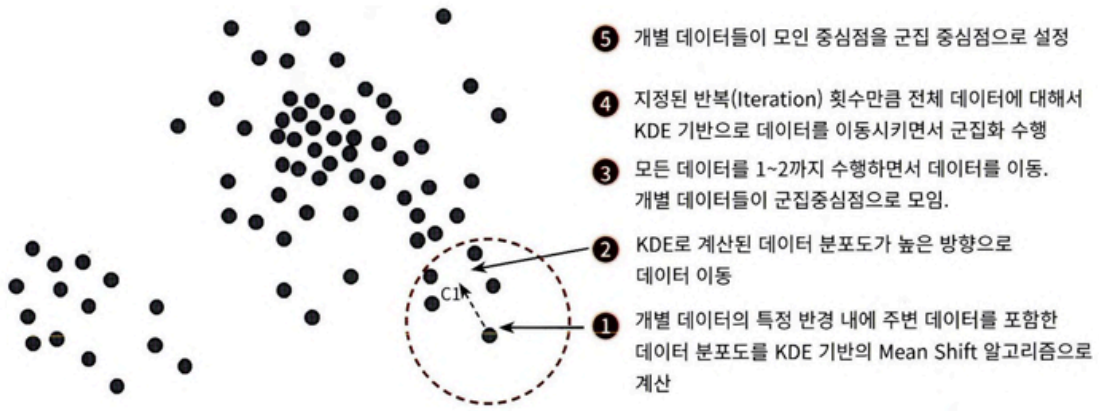
K-평균 알고리즘과 유사하게 군집 중심점을 군집의 중심으로 지속적으로 움직이며 군집화 수행

** 군집의 중심 = **데이터가 모여 있는 밀도가 가장 높은 곳**

- 확률 밀도 함수 (probability density function) 이용

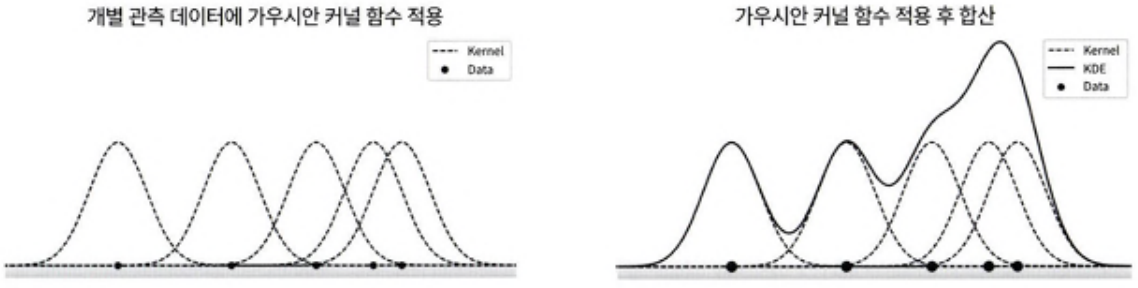
- 주어진 모델의 확률 밀도 함수를 찾기 위해 일반적으로 KDE(Kernel Density Estimation) 이용

평균 이동 동작 과정



PDF: 확률 변수의 분포를 나타내는 함수
ex. 정규분포, 감마 분포, t-분포

KDE: 커널 함수를 통해 어떤 변수의 확률 밀도 함수를 추정하는 대표적인 방법. 개별 관측 데이터에 커널 함수를 적용한 뒤, 이 적용 값을 모두 더한 후 개별 과측 데이터의 건수로 나눠 확률 밀도 함수 측정

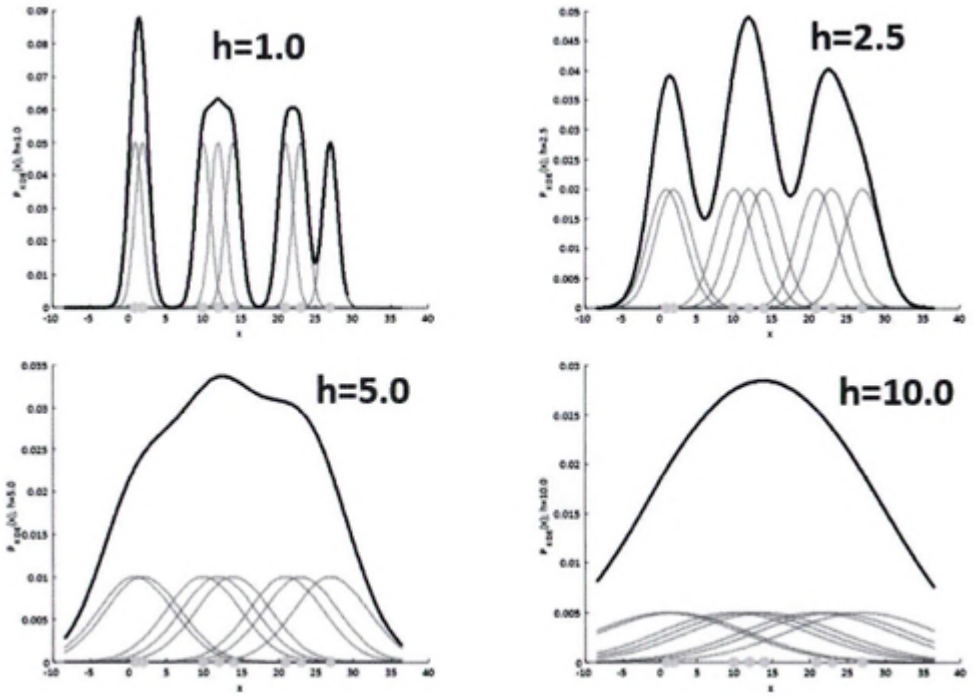


$$\mathrm{KDE} = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

KDE 커널 함수식

- 대역폭 h: KDE 형태를 부드럽거나 뾰족한 형태로 평활화(Smoothing)하는 데 적용

- 과적합, 과소적합이 일어나지 않도록 적절한 대역폭 h를 계산해야 함



사이킷런 MeanShift 클래스

- 파라미터
  - bandwidth: KDE의 대역폭 h

    → 사이킷런의 estimate_bandsidgh() 함수로 최적의 대역폭 계산 가능

장점

- 더 유연한 군집화 가능: 데이터 세트를 특정 형태나 특정 분포도 기반의 모델로 가정하지 않음
- 이상치의 영향력이 크지 않음
- 미리 군집의 개수 정할 필요 X (cf. K-평균 알고리즘)

단점

- 수행 시간이 오래 걸림
- bandwidth 크기에 따른 군집화 영향도가 ⬆️

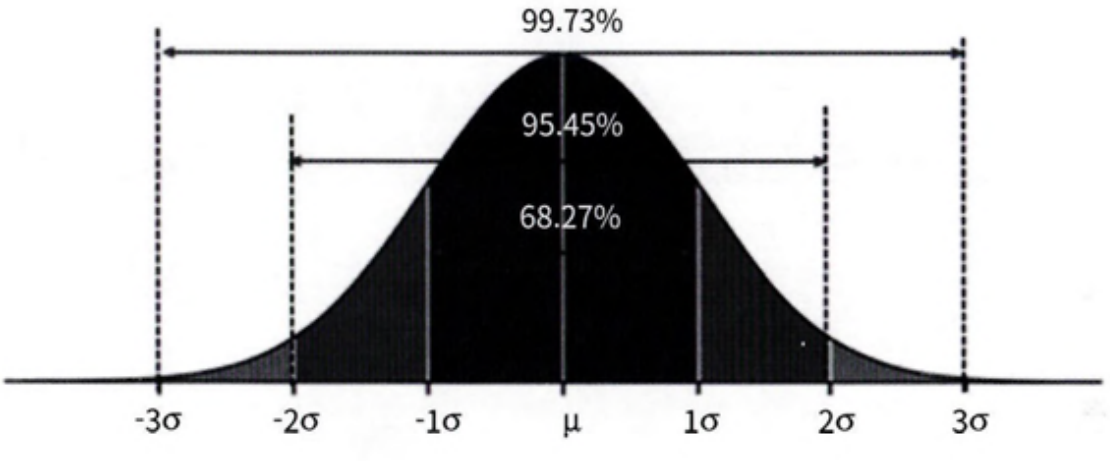
  → 컴퓨터 비전 영역에서 많이 사용

GMM (Gaussian Mixture Model)

군집화를 적용하고자 하는 데이터가 여러 개의 가우시안 분포를 가진 데이터 집합들이 섞여서 생성된 것이라는 가정 하에 군집화를 수행하는 방식

가우시안 분포 (정규 분포)

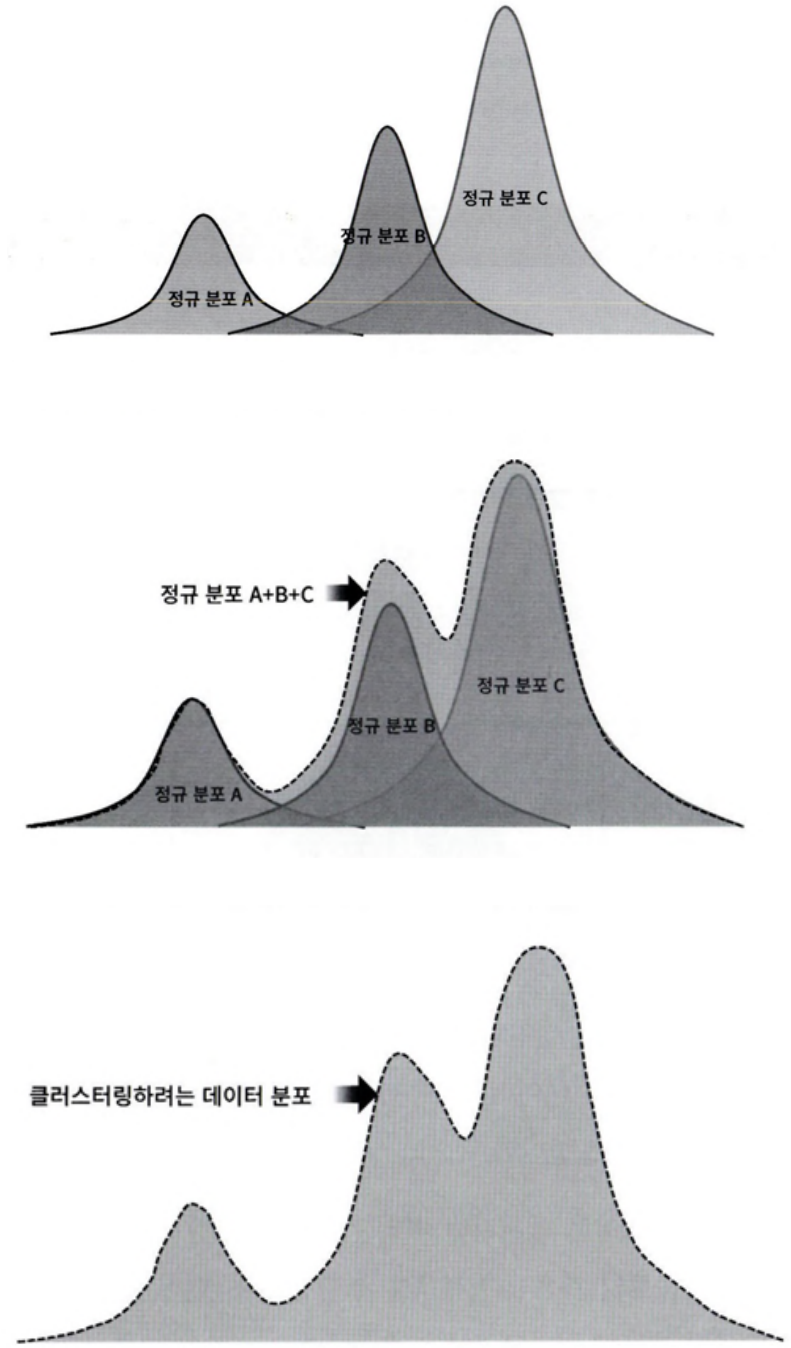
: 좌우 대칭형의 종 형태를 가진 통계학에서 가장 잘 알려진 연속 확률 함수



- 평균 μ를 중심으로 높은 데이터 분포도를 가지고 있음
- 좌우 표준편차 1에 전체 데이터의 68.27%
- 좌우 표준편차 2에 전체 데이터의 95.45%
- 표준 정규 분포: 평균이 0이고 표준편차가 1인 정규 분포

GMM

어떤 데이터 세트를 서로 다른 세 개의 정규 분포 형태를 가진 여러 확률 분포 곡선으로 구성하면 다음과 같음



이러한 **서로 다른 정규 분포에 기반해 군집화**를 수행하는 것이 GMM 군집화 방식

= 개별 데이터가 여러 정규 분포 중 어떤 정규 분포에 속하는지 결정하는 방식

모수 추정

- 개별 정규 분포의 평균과 분산
- 각 데이터가 어떤 정규 분포에 해당되는지의 확률

을 추정하는 것

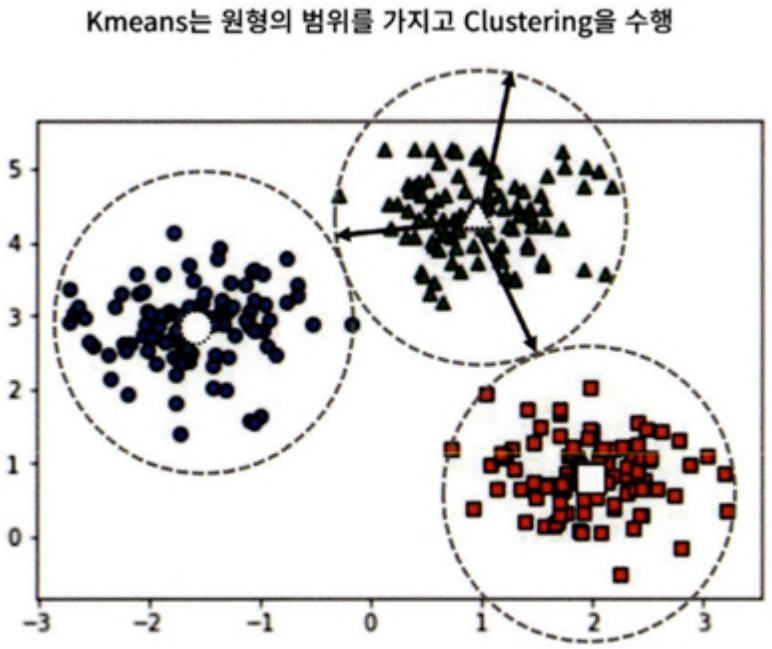
→ EM (Expectation and Maximization) 방법 적용

사이킷런 GaussianMixture 클래스

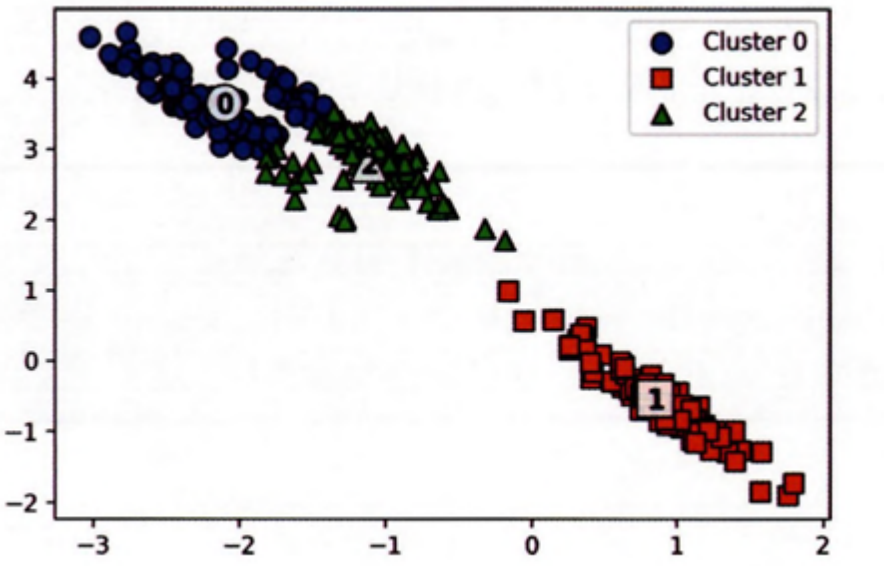
- n_components: gaussian mixture 모델의 총 개수 = 군집 개수 를 정하는 중요한 파라미터

GMM과 K-means 비교

K-means

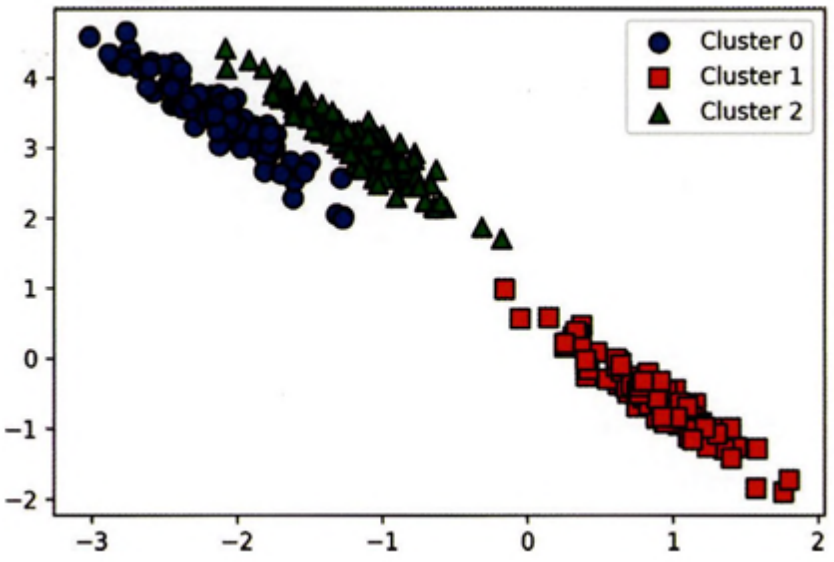


거리 기반 군집화 → 데이터 세트가 원형의 범위를 가질수록 K-means의 군집화 효율이 높아짐



→ 데이터 세트가 원형이 아닐 경우 KMeans의 군집화 정확성 떨어짐

GMM



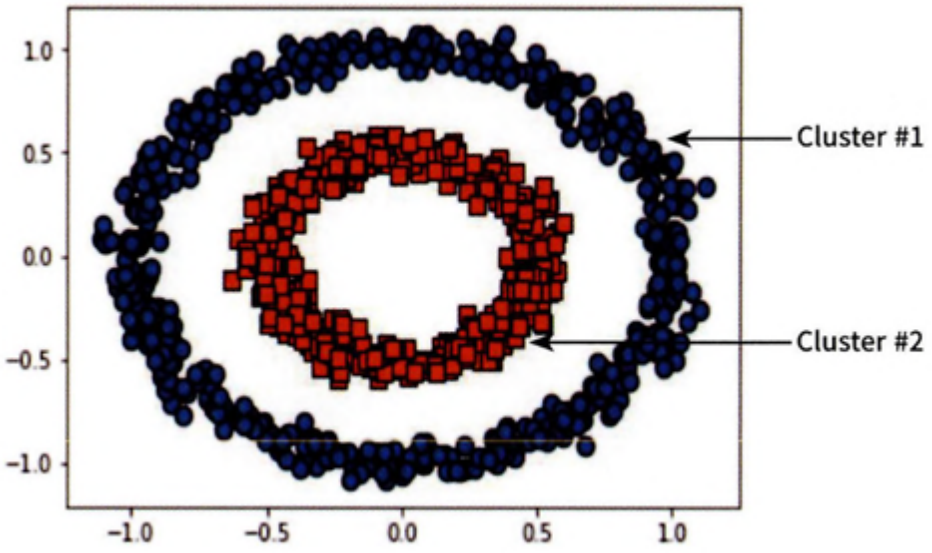
확률 기반 군집화 → 데이터 분포 방향에 따라 정확하게 군집화됨

DBSCAN

(Density Based Spatial Clustering of Applications with Noise)

⇒ **밀도 기반 군집화** 대표적인 알고리즘

특정 공간 내 데이터 밀도 차이를 기반으로 하는 알고리즘
→ 복잡한 기하학적 분포도를 가진 데이터 세트에 대해서도 군집화 잘 수행함



DBSCAN 파라미터

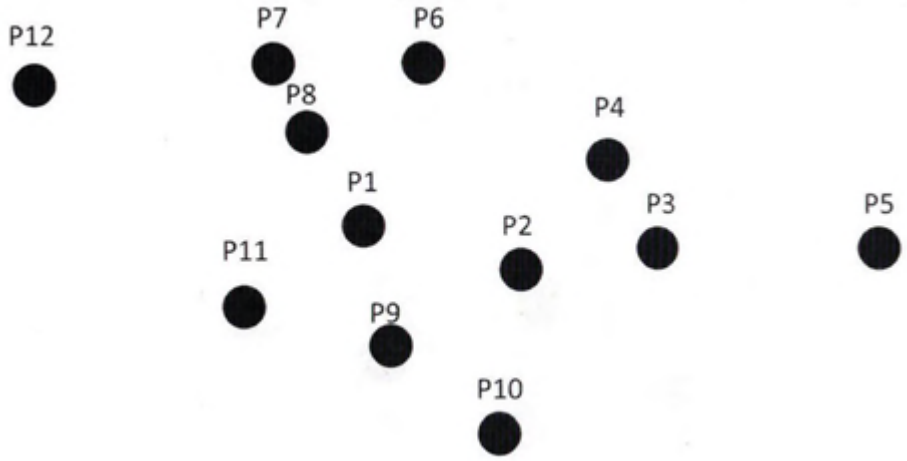
- 입실론 주변 영역 (**epsilon**): 개별 데이터를 중심으로 입실론 반경을 가지는 원형의 영역
- 최소 데이터 개수 (**min points**): 개별 데이터의 입실론 주변 영역에 포함되는 타 데이터 개수

데이터 포인트

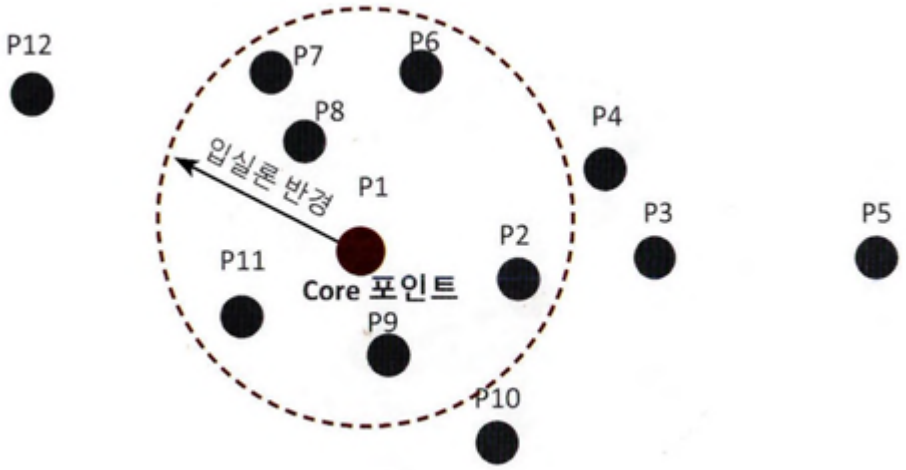
- 핵심 포인트(Core point): 주변 영역 내에 최소 데이터 개수 이상의 타 데이터를 가지고 있을 경우
- 이웃 포인트(Neighbor Point): 주변 영역 내에 위치한 타 데이터
- 경계 포인트(Border Point): 핵심 포인트가 아니지만 핵심 포인트를 이웃 포인트로 가지고 있는 데이터
- 잡음 포인트(Noise Point): 핵심 포인트가 아니고 핵심 포인트를 이웃 포인트로 가지고 있지도 않은 데이터

DBSCAN 군집화 과정

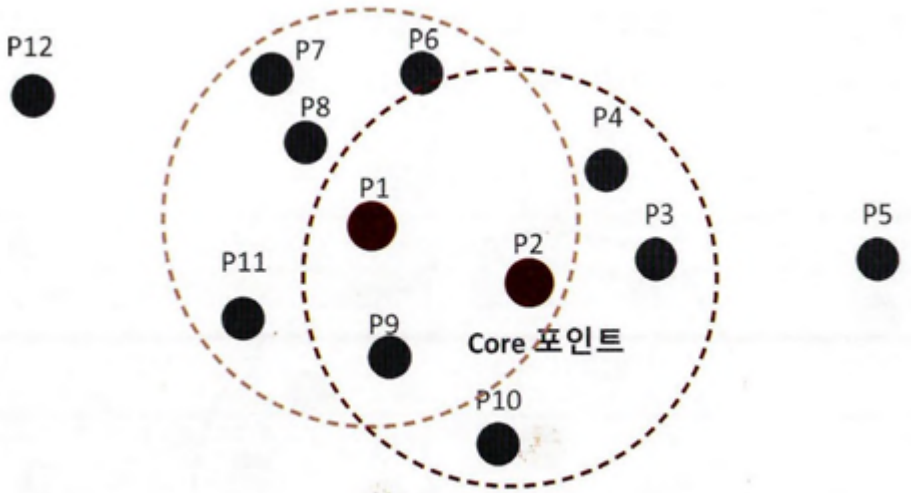
1. P1~P12까지의 12개 데이터 세트에 대해 적용, 특정 입실론 반경 내에 포함될 최소 데이터 세트를 6개로 가정



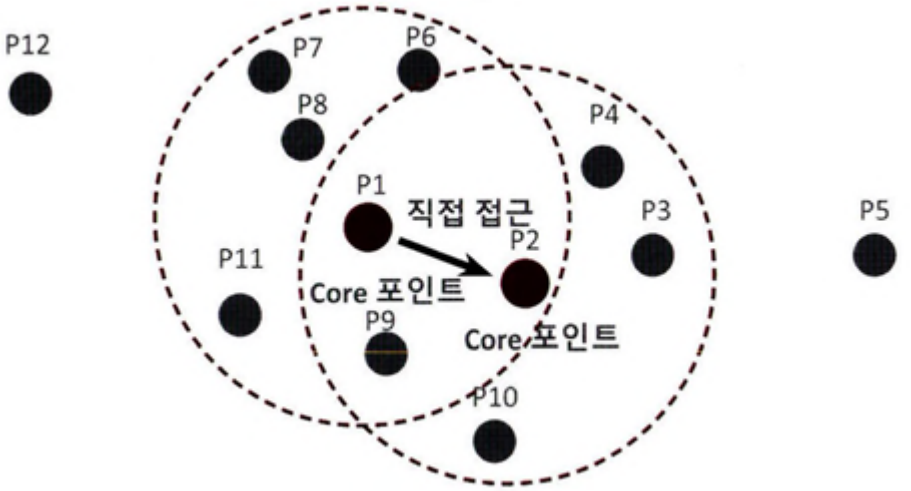
2. P1 데이터: 입실론 반경 내에 포함된 데이터 → 7개 ⇒ 핵심 포인트!



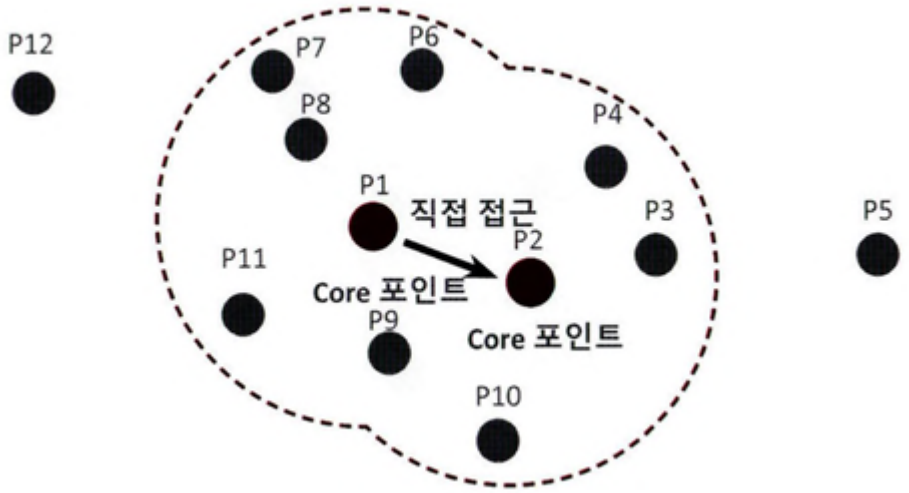
3. P2 포인트 → 반경 내 6개 데이터 ⇒ 핵심 포인트



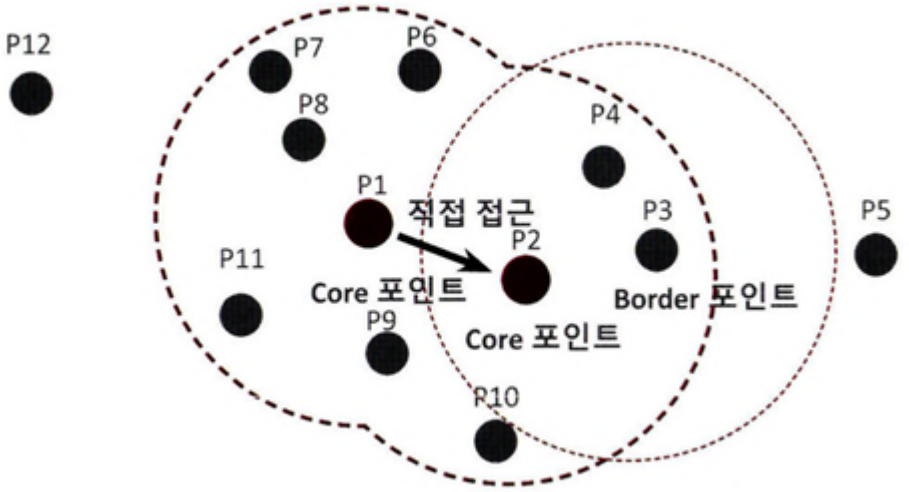
4. 핵심 포인트 P1의 이웃 데이터 포인트 P2가 핵심 포인트일 경우 P1—P2 직접 접근 가능



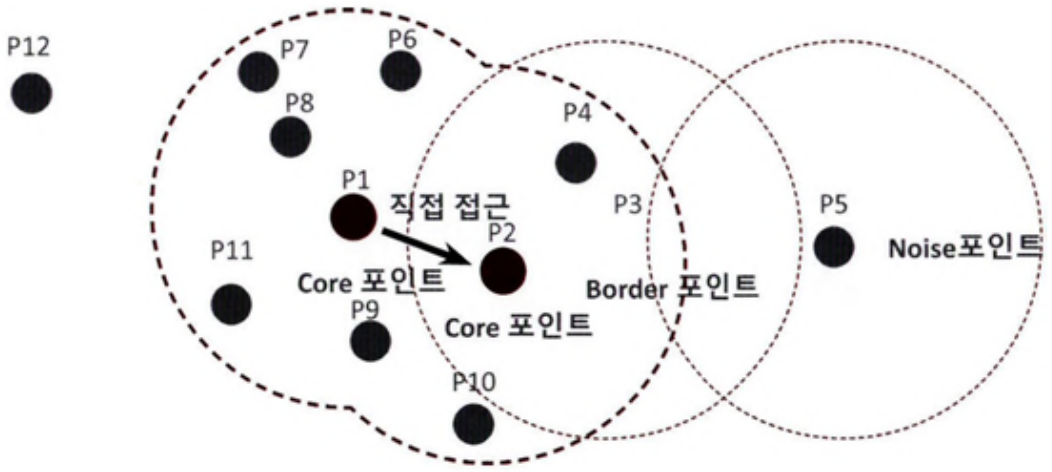
5. 특정 핵심 포인트에서 직접 접근이 가능한 다른 핵심 포인트를 서로 연결하면서 군집화 구성
   = 점차적으로 군집 영역을 확장해 나가는 방식



6. P3 데이터는 반경 내 포함되는 이웃 데이터가 2개 → 핵심 포인트 X
   but 이웃 데이터로 핵심 데이터를 포함 ⇒ 경계 포인트: 군집 외곽 형성



7. P5: 반경 내 최소 데이터 X, 이웃 데이터로 핵심 포인트 X ⇒ 잡음 포인트



사이킷런 DBSCAN 클래스

- eps: 입실론 주변 영역 반경
- min_samples: 핵심 포인트가 되기 위해 입실론 주변 영역 내에 포함되어야 할 데이터의 최소 개수 - 1

실습 - Customer Segmentation

다양한 기준으로 고객을 분류하는 기법

→ 고객의 어떤 요소를 기반으로 군집화할 것인가를 결정하는 것이 중요함

RFM 기법

- RECENCY (R): 가장 최근 상품 구입일 ~ 오늘
- FREQUENCY (F): 상품 구매 횟수
- MOMENTARY VALUE (M): 총 구매 금액