7️⃣

07 _ 군집화

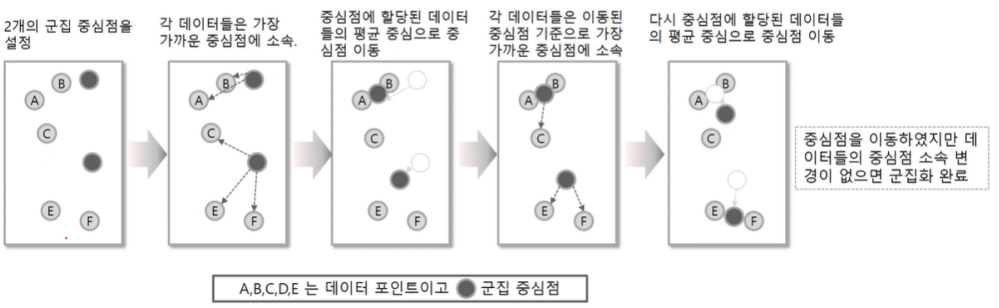
K-Means

군집화 (clustering)에서 가장 일반적으로 사용되는 알고리즘

- 임의의 지점(군집 중심점)을 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 군집화기법

> 💡 군집 중심점(centroid)
>
> 선택된 포인트의 평균지점으로 이동하고 이동된 중심점에서 다시 가까운 포인트를 선택,
>
> 다시 중심점을 평균지점으로 이동하는 프로세스를 반복적으로 수행한다.
>
> 모든 데이터포인트에서 더이상 중심점의 이동이 없을 경우, 반복을 멈추고 해당 중심점에 속하는 데이터 포인트들을 군집화한다.

1. 군집화의 기준이 되는 중심점을 군집화 개수만큼 설정하여 적합한 위치에 가져다 놓는다.
2. 각 데이터는 가장 가까운 중심점에 소속된다.
3. 소속이 결정되면 중심점이 소속된 데이터의 평균 중심으로 이동된다.
4. 중심점이 이동했기 때문에 데이터들이 가까운 중심점으로 소속을 변경한다.
5. 3-4과정을 반복한다.
6. 데이터들의 소속변경이 없으면 군집화를 종료한다.



K-평균의 장점

- 일반적인 군집화에서 가장 많이 활용되는 알고리즘
- 알고리즘이 쉽고 간결하다.
- 비지도학습으로 사전 라벨 데이터가 필요없다.

K-평균의 단점

- 거리 기반 알고리즘으로 속성(피처)의 개수가 매우 많을 경우 군집화 정확도가 떨어진다.(이를 위해 PCA 차원 감소를 적용해야 할 수도 있음)
- 반복을 수행하는데, 반복 횟수가 많을 경우 수행 시간이 매우 느려진다.
- 몇 개의 군집(cluster)을 선택해야 할지, 즉 K 설정에 대한 가이드하기가 어렵다.

사이킷런 KMeans 클래스 소개

사이킷런 패키지는 K-평균을 구현하기 위해 KMeans 클래스를 제공

```
# KMeans 초기화 파라미터
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')
```

- KMeans 초기화 파라미터 중 가장 중요한 파라미터는 **n_cluster**, **군집화 개수** , 즉 **군집 중심점의 개수**를 의미
- init는 초기에 군집 중심점의 좌표를 설정할 방식을 말하며, 보통은 임의로 중심을 설정하지 않고 k-means++ 방식으로 최초 설정

- max_iter는 최대 반복 횟수이며, 이 횟수 이전에 모든 데이터 중심점 이동이 없으면 종료
- Kmeans 알고리즘은 fit()/ fit_transform() 메서드를 이용해 수행
- Kmeans 객체는 군집화 수행이 완료돼 군집화와 관련된 주요 속성을 알 수 있음
  - labels_ : 각 데이터 포인트가 속한 군집 중심의 레이블
  - cluster_centers_: 각 군집 중심점 좌표(Shape 는 [군집개수, 피처개수]). 이를 이용하면 군집 중심점의 좌표가 어디인지 시각화 할 수 있음

군집 평가(Cluster Evaluation)

- 군집화는 classification과 달리 타깃 레이블을 가지고 있지 않고, 동일한 분류 값에 속하더라도 그 안에서 더 세분화된 군집화를 추구하거나 서로 다른 분류 값의 데이터도 더 넓은 군집화 레벨화 등의 영역을 가지고 있음
- 군집 평가를 위한 방법으로 **실루엣 분석**을 이용
-

<실루엣 분석의 개요>

- **실루엣 분석**은 **각 군집 간의 거리는 떨어져 있고 동일 군집끼리의 데이터는 서로 가깝게 잘 뭉쳐있는지,** 즉 얼마나 효율적으로 잘 분리돼있는지를 나타냄

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

- **실루엣 계수 s(i)** : 개별 데이터가 가지는 군집화 지표로 해당 데이터가 같은 군집 내의 데이터와 얼마나 가깝게 군집화 되어있고, 다른 군집 데이터와는 얼마나 멀리 분리돼있는지를 나타냄
  - a(i) : i번째 데이터에서 자신이 속한 클러스터 내의 다른 데이터 포인트들의 평균 거리
  - b(i) : i번째 데이터에서 가장 가까운 타 클래스터 내의 다른 데이터 포인트들의 평균 거리

    - 1 에서 1 사이 값을 가짐.
    - 1 에 가까울 수록 근처의 군집과 더 멀리 떨어지고 0에 가까울수록 근처의 군집과 가까워짐. -값은 다른 군집에 데이터 포인트가 할당됐음을 뜻함
- 실루엣 분석을 위한 사이킷런 메서드
    - *sklearn.metrics.silhouette_samples(X, labels, metric='euclidean', *kwds*) : 각 데이터 포인트의 실루엣 계수 반환
    - *sklearn.metrics.silhouette_score(X, labels, metric='euclidean', sample_size=None, **kwds)*: 전체 데이터의 실루엣 계수 값의 평균 반환 = np.mean(silhouette_samples())

평균 이동

- 좋은 군집화의 조건
    1. 전체 실루엣의 평균값인 silhouette_score() 값은 **0~1 사이의 값을 가지며 1에 가까울 수록 좋음**
    2. **전체 실루엣 계수의 평균값과 더불어 개별 군집 평균값의 편차가 크지 않아야 함.** 예를 들어, 전체 실루엣 계수의 평균값은 높지만, 특정 군집의 실루엣 계수 평균값만 유난히 높고 다른 군집들의 실루엣계수 평균값은 낮으면 좋은 군집화 조건이 아님

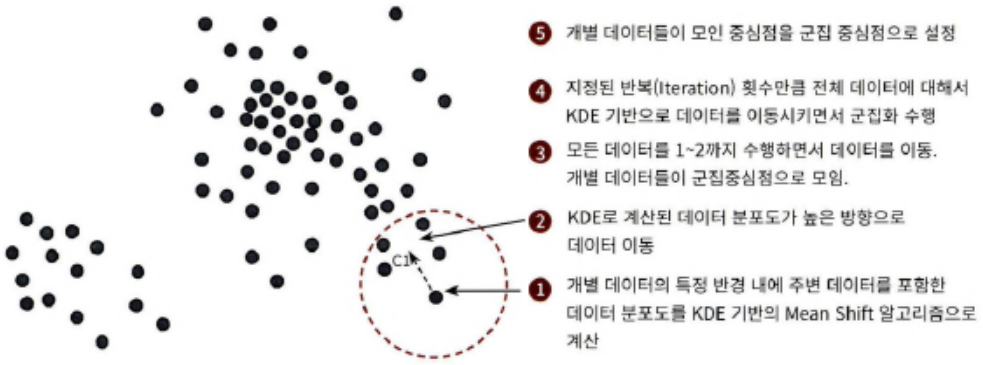
K-평균과 유사하게, 군집의 중심을 지속적으로 움직이면서 군집화를 수행함

그러나, K-평균이 중심에 소속된 데이터의 평균 거리 중심으로 이동하는데 반해,

평균 이동은 데이터가 모여있는 밀도가 가장 높은 곳으로 이동시키면서 군집화하는 방법

- 평균 이동 군집화는 데이터의 분포도를 이용해 군집 중심점을 찾음
    - 군집 중심점은 데이터 포인트가 모여있는 곳이라는 생각에서 착안
    - 이를 위해 확률 밀도 함수를 이용 함
    - 확률 밀도 함수가 피크인 점(가장 집중적으로 데이터가 모여 있을)을 군집 중심점으로 선정하며
    - 주어진 모델의 확률 밀도 함수를 찾기 위해서 KDE(Kernel Density Estimation)을 이용

- 주변 데이터와의 거리 값을 KDE 함수 값으로 입력한 뒤, 그 반환 값을 현재 위치에서 업데이트하면서 이동하는 방식



- KDE(Kernel Density Estimation)

  커널 함수를 통해 어떤 변수의 확률 밀도 함수를 추정하는 대표적인 방법 개별 데이터 각각에, 커널 함수를 적용한 값을 모두 더한 뒤 데이터 건수로 나눠 확률 밀도 함수를 추정한다.

  - 확률 밀도 함수 PDF(Probability Density Function)

    확률 변수의 분포를 나타내는 함수 (정규 분포, 감마 분포, t-분포 등)

    확률 밀도 함수를 알면 특정 변수가 어떤 값을 갖게 될지에 대한 확률을 알게 되므로, 이 를 통해 변수의 특성, 확률 분포 등 변수의 많은 요소를 알 수 있다.

> 💡 MeanShift 클래스
>
> 사이킷런은 평균 이동 군집화를 위해 MeanShift 클래스를 제공
>
> 가장 중요한 파라미터 : bandwidth (KDE의 대역폭 h와 동일)
>
> 대역폭 크기 설정이 군집화의 품질에 큰 영향을 미치기 때문에 최적의 대역폭 계산을 위해 `estimate_bandwidth()` 함수를 제공

- 평균 이동의 장점
  - 데이터 세트의 형태를 특정 형태로 가정한다든가, 특정 분포 기반의 모델로 가정하지 않기 때문에 유연한 군집화 가능
  - 이상치의 영향력도 크지 않으며, 미리 군집의 개수를 정하지 않아도 된다.
- 평균 이동의 단점

        - 수행 시간이 오래 걸리고, bandwidth의 크기에 따른 군집화 영향도가 크다.
    - 활용
        - 컴퓨터 비전 영역에서 많이 사용
        - 이미지나 영상 데이터에서, 특정 개체를 구분하거나 움직임을 추적하는데 뛰어난 역할

GMM(Gaussian Mixture Model)

군집화를 적용하고자 하는 데이터가 여러 개의 가우시안 분포(GaussianDistribution)를 가진

데이터 집합들이 섞여서 생성된 것이라는 가정하에 군집화를 수행하는 방식

- GMM(Gaussian Mixture Model)은 데이터를 여러 개의 가우시안 분포가 섞인 것으로 간주
- 여러 개의 정규 분포 곡선을 추출하고, 개별 데이터가 어떤 정규 분포에 속하는지 결정

→ 이와 같은 방식을 모수 추정이라고 하는데, 모수 추정은 대표적으로 2가지를 추정

1. 개별 정규 분포의 평균과 분산
2. 각 데이터가 어떤 정규 분포에 해당되는지의 확률

모수 추정을 위해 GMM은 EM(Expectation and Maximization) 방법을 적용한다.

DBSCAN

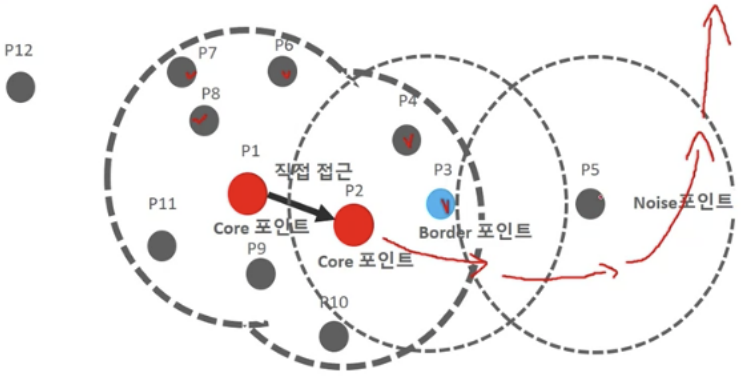
대표적인 밀도 기반 군집화 알고리즘

특정 공간 내 데이터 밀도 차이를 기반 알고리즘으로 하고 있어 복잡한 기하학적 분포를 가진 데이터에도 군집화를 잘 수행한다.

DBSCAN 주요 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

핵심 포인트(Core Point): 주변 영역 내에 최소 데이터 개수 이상의 타 데이터를 가지고 있을 경우 해당 데이터를 핵심 포인트라고 합니다.
이웃 포인트(Neighbor Point): 주변 영역 내에 위치한 타 데이터를 이웃 포인트라고 합니다.
경계 포인트(Border Point): 주변 영역 내에 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않지만 핵심 포인트를 이웃 포인트로 가지고 있는 데이터를 경계 포인트라고 합니다.
잡음 포인트(Noise Point): 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않으며, 핵심 포인트도 이웃 포인트로 가지고 있지 않는 데이터를 잡음 포인트라고 합니다



- 수행 방법
  1. P1 데이터 기준으로 입실론 반경(eps=0.6)내에 포함한 데이터(min_samples=6)가 7개 (자신 P1 포함하여, 이웃 데이터 P2, P6, P7, P9, P11)로 최소 6개 이상을 만족하므로 P1은 핵심 포인트(Core Point)
  2. 다음으로 P2 데이터를 보면, P2 역시 반경 내에 6개 (자신 P2, P1, P3, P4, P9, P10) 데이터를 갖고 있으므로 핵심 포인트
  3. 핵심 포인트 P1의 이웃 ↔ 데이터 포인트 P2 역시 핵심 포인트일 경우 ⇒ P1에서 P2를 연결하여 *[직접 접근]*
  4. 특정 핵심 포인트에서 *[직접 접근]*이 가능한 다른 핵심 포인트들을 서로 연결하면서 군집화를 구성 ⇒ 이런 군집화 영역을 확장해나가는 것이 DBSCAN의 군집화 방식
  5. P3 데이터의 경우, 이웃 데이터로 P2, P4 2개이므로 군집을 구분할 수 있는 핵심 포인트는 될 수 없음
     a. 하지만 이웃 데이터 중에 핵심 포인트인 P2를 가지고 있음
     b. 이렇게, 자신은 핵심 포인트가 아니지만, 이웃 데이터로 핵심 포인트를 가지고 있는 데이터를 경계 포인트(Border Point)라고 부름
     c. 경계 포인트는 *군집의 외곽*을 형성
  6. P5와 같이 반경내 최소 데이터를 갖고 있지도 않고, 핵심 포인트를 이웃 데이터로 가지고 있지 않은 데이터를 잡음 포인트(Noise Point)라고 함
  - 핵심 포인트(core point) : 주변 영역 내 최소 데이터 갯수 이상의 타 데이터를 가지고 있을 경우
  - 이웃 포인트(neighbor point): 주변 영역 내에 위치한 타 데이터

- 경계 포인트(border point): 주변 영역 내에 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않지만 핵심 포인트를 갖고 있는 데이터
- 잡음 포인트(noise point): 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않으며, 핵심 포인트도 갖고 있지 않은 데이터