11주차 예습 과제

7.1 K-평균 알고리즘 이해

- K-평균: 군집 중심점(centroid)를 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 군집화 기법
  1. 만들고자 하는 군집 개수만큼 임의로 중심점을 잡음
  2. 각 데이터는 거리 계산을 통해 가장 가까운 중심점의 군집에 소속됨
  3. 군집별 평균 위치로 중심점 이동
  4. 중심점이 이동했기 때문에 다시 각 데이터가 가장 가까운 중심점을 기준으로 소속을 재조정
  5. 새롭게 형성된 군집을 기준으로 중심점을 다시 평균 위치로 이동
  6. 데이터의 군집 소속이 더 이상 바뀌지 않으면 알고리즘을 종료한다.
- 장점: 군집화에서 가장 많이 활용되는 알고리즘으로, 쉽고 간결함
- 단점:
  - 거리 기반 알고리즘이기 때문에 속성의 개수가 많으면 정확도가 떨어짐
  - 반복 횟수가 많으면 수행 시간이 매우 느려짐
  - 군집 개수 선택이 어려움
- 사이킷런 패키지는 KMeans 클래스를 제공
  - 주요 초기화 파라미터:
    - n_clusters: 생성할 군집 개수
    - init: 초기 중심점을 어떻게 정할지 결정하는 방식 (default='k-means++')
    - max_iter: 한 번의 실행에서 중심점 재계산 및 군집 재배정 과정을 반복하는 최대 횟수
  - 주요 속성:
    - labels_: 각 데이터 포인트가 어떤 군집에 속했는지 나타냄
    - cluster_centers_: 군집 중심점들의 좌표 → (군집 개수, 피처 개수)
- 군집화 테스트 데이터 생성:
  - make_blobs(): 원하는 개수의 중심점과 분포를 가진 데이터 생성 가능

→ 피처 데이터 세트와 타깃 데이터 세트가 튜플 형태로 반환

- n_samples: 생성할 데이터 개수 (default=100)
- n_features: 피처 개수
- centers: 군집의 개수 또는 개별 중심점 좌표를 지정
- cluster_std: 각 군집 데이터의 표준편차

- make_classification(): 노이즈를 포함한 데이터를 만드는데 유용
- make_circle(), make_moon(): 비선형 패턴의 군집 데이터를 생성

7.2 군집 평가

- 군집화는 대부분 정답 레이블이 없어 성능 평가가 어려움
- 실루엣 분석: 군집 간의 거리가 얼마나 잘 분리되어 있는지를 평가하는 대표적인 군집화 성능 측정 방법 → 실로엣 계수에 기반
- 실루엣 계수: 개별 데이터가 가지는 군집화 지표
  - 같은 군집 내 데이터들과 얼마나 가깝게 모여 있는지
  - 다른 군집의 데이터들과는 얼마나 멀리 떨어져 있는지 나타냄

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

  - a(i) = i번째 데이터에서 같은 군집 내의 다른 데이터까지의 평균 거리
  - b(i) = i번째 데이터에서 가장 가까운 타 군집 내의 다른 데이터까지의 평균 거리

    ⇒ b(i)-a(i) = 두 군집 간의 거리

    ⇒ 두 값 중 더 큰 값으로 나눠주어서 정규화 (-1~1 값으로 조정)
- 사이킷런 제공 메서드:
  - sklearn.metrics.silhouette_samples(): 각 데이터의 실루엣 계수 계산
  - sklearn.metrics.silhouette_score(): 전체 평균 실루엣 계수 계산
- 좋은 군집화 조건:
  1. 전체 실루엣 계수 평균(silhouette_score)이 1에 가까울수록 좋음
  2. 전체 평균뿐 아니라 개별 군집의 평균 실루엣 계수도 균일해야 함

7.3 평균 이동

- 평균 이동(Mean Shift): 데이터가 많이 모여 있는 방향으로 점을 조금씩 이동시키면서 군집의 중심을 찾음
  - 데이터 분포의 확률 밀도 함수를 기반으로 군집 중심을 찾음
  - 확률 밀도 추정을 위해 KDE(Kernel Density Estimation)를 사용
  - 군집의 개수 지정 X
- KDE: 개별 관측 데이터에 커널 함수를 적용한 뒤 이 적용 값을 모두 더한 후 데이터의 건수로 나눠 확률 밀도 함수를 추정

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

  - 대역폭 h: 작을수록 좁고 뾰족하게 확률 밀도 함수를 추정 → 과적합하기 쉬움, 군집의 개수 많아짐
- 사이킷런은 `MeanShift` 클래스를 제공
  - bandwidth: KDE의 대역폭 h 설정하는 초기화 파라미터
  - estimate_bandwidth(): 최적의 대역폭 계산 함수
- 장점:
  - 데이터 세트의 분포를 가정하지 않기 때문에 유연한 군집화 가능
  - 이상치의 영향 적음
  - 군집의 개수 정할 필요 없음
- 단점: 오래 걸리고 bandwidth의 영향이 매우 큼
- 일반적으로 컴퓨터 비전 영역에서 많이 사용됨

7.4 GMM (Gaussian Mixture Model)

- GMM: 데이터가 여러 개의 가우시안 분포(=정규분포)를 가진 데이터 집합들이 섞어서 생성된 것이라고 가정하여 군집화를 수행하는 방법
- 모수 추정:

  ⇒ 전체 데이터 분포에서 여러 개의 정규 분포 곡선을 추출하고 개별 데이터가 어떤 정규 분포에 속하는지 결정하는 것. 즉 다음 2가지를 추정하는 것임.

  1. 개별 정규 분포의 평균 / 분산 추정

2. 각 데이터가 어떤 정규 분포에 해당되는지의 확률

- 사이킷런은 `GaussianMixture` 클래스 지원
  - n_components: gaussian mixture 모델의 총 개수
- GMM vs K-평균
  - K-평균: 원형의 범위에서 군집화를 수행 → 데이터가 원형의 범위를 가질수록 군집화 효율이 높아짐
  - GMM: K-평균보다 유연하게 다양한 데이터 세트에 적용됨

7.5 DBSCAN (Density Based Spatial Clustering of Applications with Noise)

- DBSCAN: 핵심 포인트를 연결하면서 군집화를 구성하는 밀도 기반 군집화 알고리즘
  - 분포가 기하학적으로 복잡한 데이터도 효과적으로 군집화 가능
  - 중요 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 데이터의 개수
  - 데이터 분류:
    - 핵심 포인트(Core Point): 주변 영역 내에 최소 데이터 개수 이상의 타 데이터를 가지고 있는 경우
    - 이웃 포인트(Neighor 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 이웃 포인트로 가지고 있는 데이터
    - 잡음 포인트(Noise Point): 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않으며, 핵심 포인트도 이웃 포인트로 가지고 있지 않는 데이터
  - 군집화 방식:
    - 핵심 포인트의 이웃 중 또 다른 핵심 포인트가 있으면 직접 접근 가능 → 이들을 서로 연결하면서 군집 영역을 확장함.
    - 경계 포인트는 군집의 외곽을 형성
- 사이킷런은 `DBSCAN` 클래스 지원

    - 주요 초기화 파라미터:
        - eps: 입실론 주변 영역의 반경
        - min_samples: 핵심 포인트가 되기 위해 입실론 주변 영역 내에 포함돼야 할 데이터의 최소 개수 (자신 포함하므로 min points+1)
    - 노이즈의 군집 레이블은 -1임
- 적절한 eps와 min_sample 파라미터를 통해 최적을 군집을 찾는게 중요
    - eps가 커지면 노이즈 개수 작아짐
    - min_samples는 커지면 노이즈 개수 커짐