핸즈온 머신러닝 CH9 : 비지도 학습

알고리즘이 레이블이 없는 데이터를 바로 사용하는 것

- 군집

  비슷한 샘플을 클러스터로 모아
  데이터 분석, 고객 분류, 추천 시스템, 검색 엔진, 이미지 분할, 준지도 학습, 차원 축소 등에 사용

- 이상치 탐지

  정상 데이터가 어떻게 보이지는 학습 → 그다음 비정상 샘플을 감지
  ex. 제조 라인에서 결함 제품을 감지하거나 시계열 데이터에서 새로운 트랜드를 찾아

- 밀도 추정

  데이터셋 생성 확률 과정의 확률 밀도 함수 **PDF** 를 추정
  밀도 추정은 이상치 탐지에 널리 사용됨

  밀도가 매우 낮은 영역에 놓인 샘플이 **이상치**일 가능성이 높아

  데이터 분석과 시각화에 유용

군집

▼ K-평균

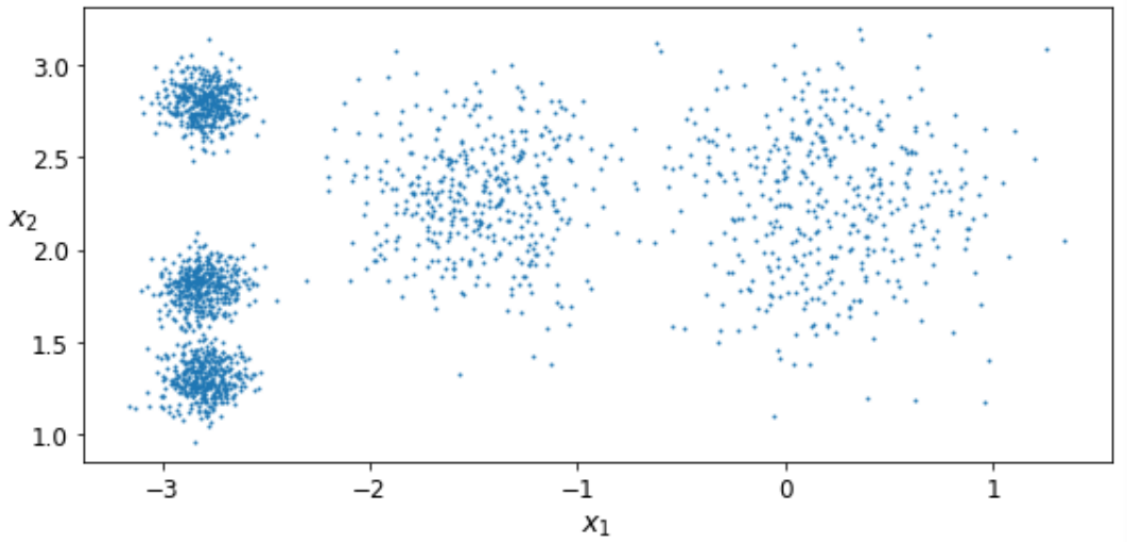
```
from sklearn.datasets import make_blobs

blob_centers = np.array(
    [[ 0.2,  2.3],
     [-1.5 ,  2.3],
     [-2.8,  1.8],
     [-2.8,  2.8],
     [-2.8,  1.3]])
blob_std = np.array([0.4, 0.3, 0.1, 0.1, 0.1])

X, y = make_blobs(n_samples=2000, centers=blob_centers,
                  cluster_std=blob_std, random_state=7)

#그래프
def plot_clusters(X, y=None):
    plt.scatter(X[:, 0], X[:, 1], c=y, s=1)
    plt.xlabel("$x_1$", fontsize=14)
```

```
    plt.ylabel("$x_2$", fontsize=14, rotation=0)
plt.figure(figsize=(8, 4))
plot_clusters(X)
save_fig("blobs_plot")
plt.show()
```



▼ 훈련과 예측

K-평균 군집 알고리즘 : 클러스터 중심을 찾고 각 샘플에 가까운 클러스터에 할당

```
from sklearn.cluster import KMeans

K=5
kmeans = Kmeans(n_clusters=k, random_state=42)
y_pred=kmeans.fit_predict(X)
# 각 샘플은 5개의 클러스터 중 하나로 할당됨

y_pred # 0~4 의 값으로 할당됨
y_pred is kmeans.labels_ #True 값 나와

kmeans.cluster_centers_ # 5개의 클러스터 중심 (센트로이드)
kmeans.labels_ # y_pred와 같은 건가요???

#새로운 샘플의 레이블 예측
X_new = np.array([[0, 2], [3, 2], [-3, 3], [-3, 2.5]])
kmeans.predict(X_new) # 0,0,3,3 으로 각각 나옴
```

▼ 하드 군집 vs 소프트 군집

- 하드 군집 : 각 샘플에 대해 가장 가까운 클러스터를 선택,
  샘플을 하나의 클러스터에 할당
- 소프트 군집 : 클러스터마다 샘플에 점수를 부여

▼ K-평균 알고리즘

가장 빠르고 간단한 군집 알고리즘 중 하나

- k 개의 센트로이드를 랜덤하게 초기화 : k 개의 샘플을 랜덤하게 선택하고, 센트로이드를 그 위치에 놓습니다.
- 센트로이드가 더 이상 움직이지 않을 때(수렴)까지 다음을 반복
  - 각 샘플을 가장 가까운 센트로이드에 할당
  - 센트로이드에 할당된 샘플의 평균으로 센트로이드를 업데이트

센트로이드 초기화 방법

```
#inint 매개변수에 센트로이드 리스트를 담은 넘파이 배열 지정, n_init=1로 설정

good_init = np.array([[-3,3],[-3,2],[-3,1],[-1,2],[0,2]])
kmeans = KMeans(n_clusters=5, init=good_init, n_init=1)
```

KMeans 클래스는 알고리즘을 n_inint 번 실행하여 이니셔가 가장 낮은 모델을 반환

가우시안 혼합

샘플이 파라미터가 알려지지 않은 여러 개의 혼합된 가우시안 분포에서 생성되었다고 가정하는 확률 모델

하나의 가우시안 분포에서 생성된 모든 샘플은 하나의 클러스터를 형성

(일반적으로 이 클러스터는 타원형)

```
from sklearn.mixture import GaussianMixture

gm=GaussianMixture(n_components=3, n_init=10)
gm.fit(X)


gm.weights_ #추정한 파라미터
gm.means_
gm.covariances_
```

- 가우시안 혼합을 사용한 이상치 탐지

  밀도가 낮은 지역에 있는 모든 샘플을 이상치로 봄

- 클러스터 개수 선택하기

  BIC나 AIC와 같은 이론적 정보 기준을 최소화하는 모델을 찾아