2. Clustering

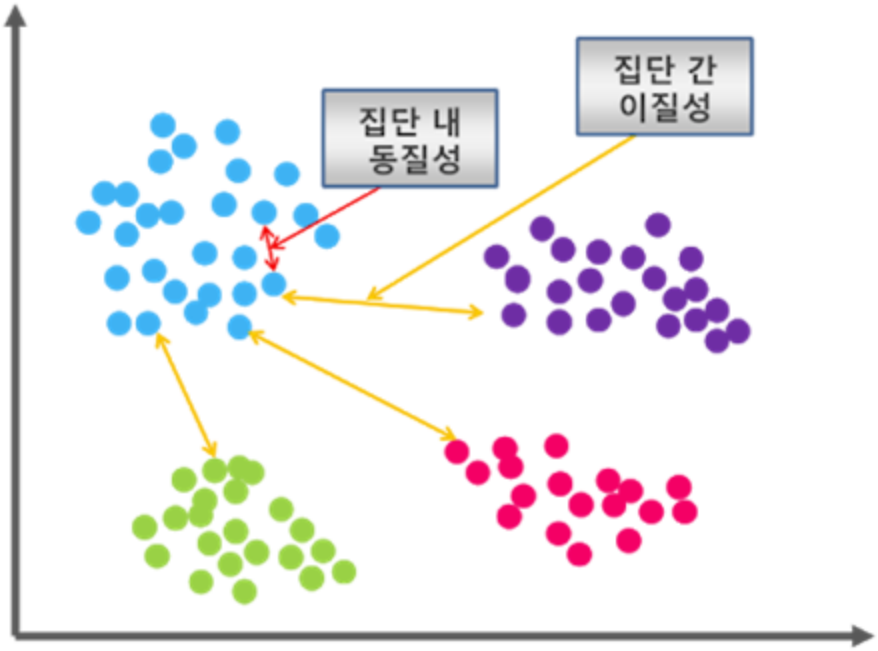
- 레이블이 없는 데이터를 사용하는, 또는 레이블이 있더라도 그것을 사용하지 않는 비지도 학습의 일종
- 정해진 정답이 없음

목적

- 예측성 분석이 아니기 때문에 주로 분석 초기 단계에서 데이터의 특성을 파악하기 위해서 활용됨
    - 어떤 목적 변수를 예측하기보다는 고객의 수입, 연령과 같이 비슷한 고객을 묶어서 몇 개의 의미있는 군집으로 나누는 것
    - 대상 개체를 유사하거나 서로 관련있는 항목끼리 묶어서 몇 개의 집단으로 그룹화하거나, 각 집단의 성격을 파악함으로써 데이터 전체의 구조에 대한 이해를 돕고자 하는 탐색적 분석 방법
    - 대용량의 데이터가 너무 복잡하게 분포되어 있을 때 이를 구성하고 있는 몇 개의 군집을 우선 살펴보고 이런 군집들의 특성을 살펴서 전체적인 윤곽을 잡을 수 있음
    - 숲이 너무 복잡해서 전체를 파악하기 힘들 때 이를 구성하고 있는 나무를 먼저 살펴보는 것과 비슷한 이치

원리

- 유사한 입력값끼리 묶어서 군집을 찾고 때로는 각 군집에 맞는 대표값을 찾음
    - 한 클러스터 안의 데이터 포인트끼리는 매우 비슷하고 다른 클러스터의 데이터 포인터와는 구분되도록 데이터를 나눔
    - 동일한 군집 내 개체들은 유사한 성격(집단내 동질성)을 가지고, 서로 다른 군집은 이질적인 성격(집단간 이질성)을 갖도록 군집이 형성됨



군집의 유형

- 상호 배반적 군집 : 각 관찰치가 군집 중 오직 하나에 속함(일반적)
- 계보적 군집 : 한 군집이 다른 군집에 포함, 상하종속 관계
- 중복 군집 : 두 개 이상의 군집에 한 관찰치가 소속
- 퍼지 군집 : 각 객체가 각 군집에 속할 확률이나 자격을 어떤 지표로 규정하는데, 이는 상호배반적, 계보적, 중복 등의 어느 형태를 취할 수 있음

군집 알고리즘 종류

- K-평균(KMeans)
  - 설명
    - 가장 간단하고 또 널리 사용하는 군집 알고리즘
    - 데이터의 어떤 영역을 대표하는 클러스터의 중심을 찾는 알고리즘
    - 데이터 포인트를 가장 가까운 클러스터 중심에 할당하고, 클러스터에 할당된 데이터 포인트의 평균으로 클러스터 중심을 다시 지정하는 두 단계를 반복함

- 클러스터에 할당되는 데이터 포인트에 변화가 없을 때 알고리즘이 종료됨

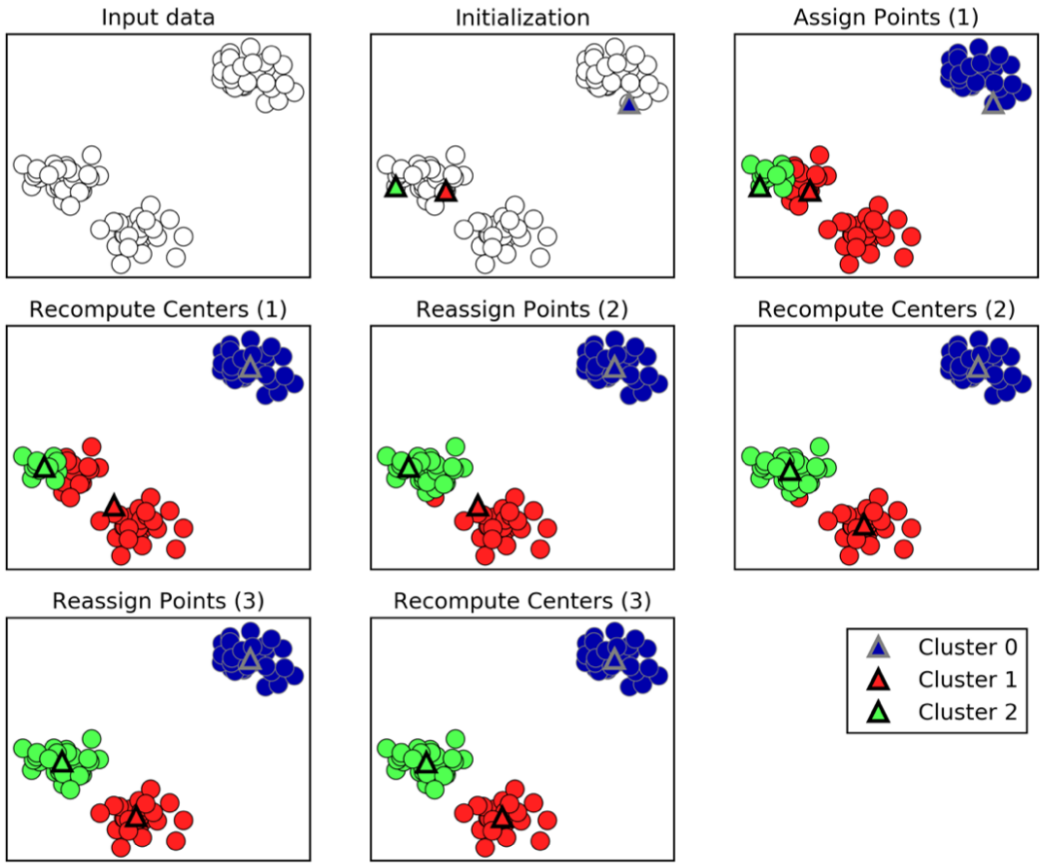


Figure 3-23. Input data and three steps of the k-means algorithm

- K-Means 알고리즘으로 찾은 클러스터 중심과 클러스터 경계

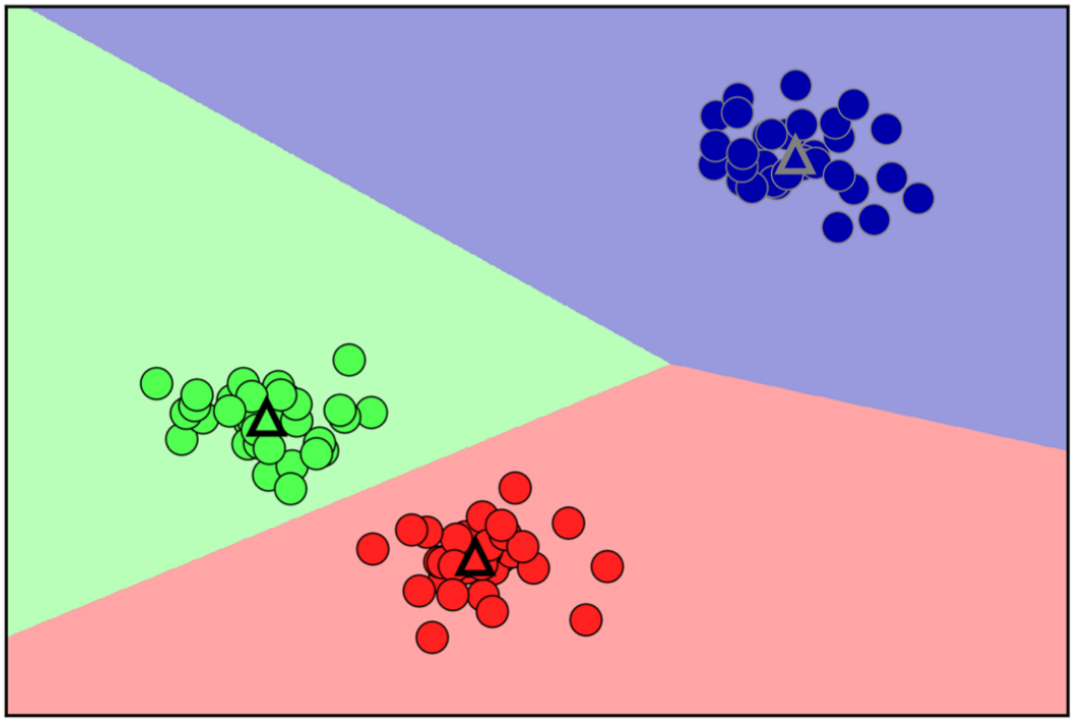
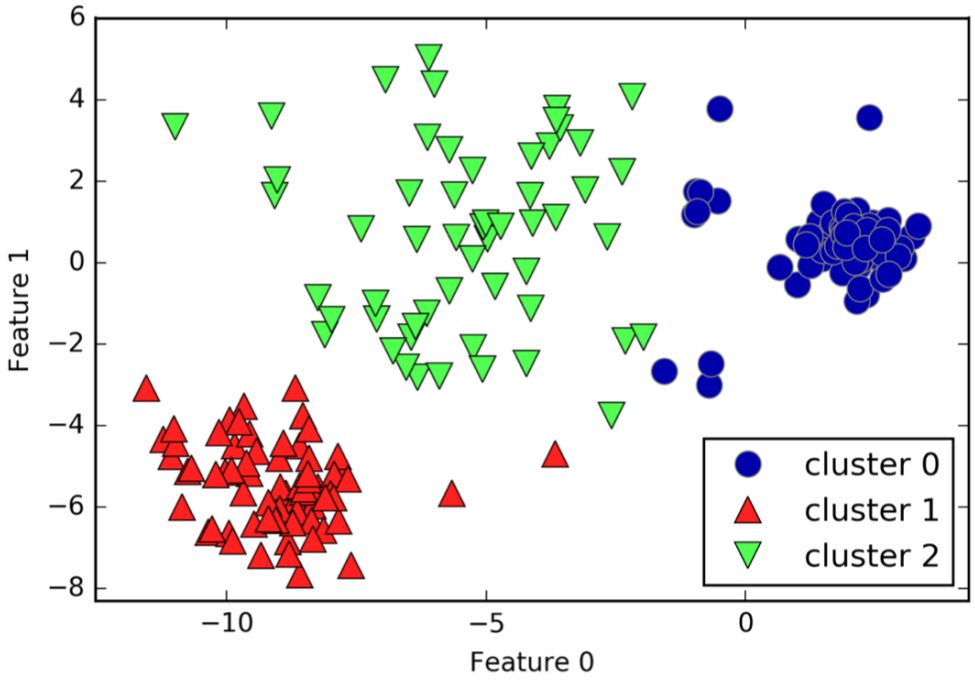


Figure 3-24. Cluster centers and cluster boundaries found by the k-means algorithm

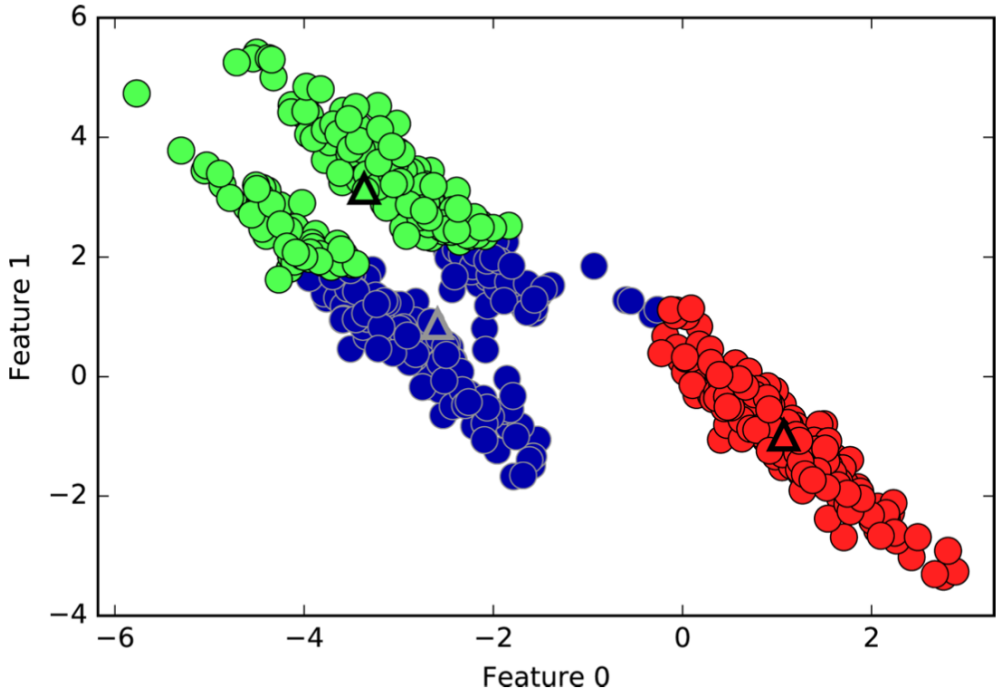
- 실패하는 경우

  데이터의 클러스터 개수를 정확하게 알고 있더라도 항상 이 알고리즘이 성공하는 것은 아님

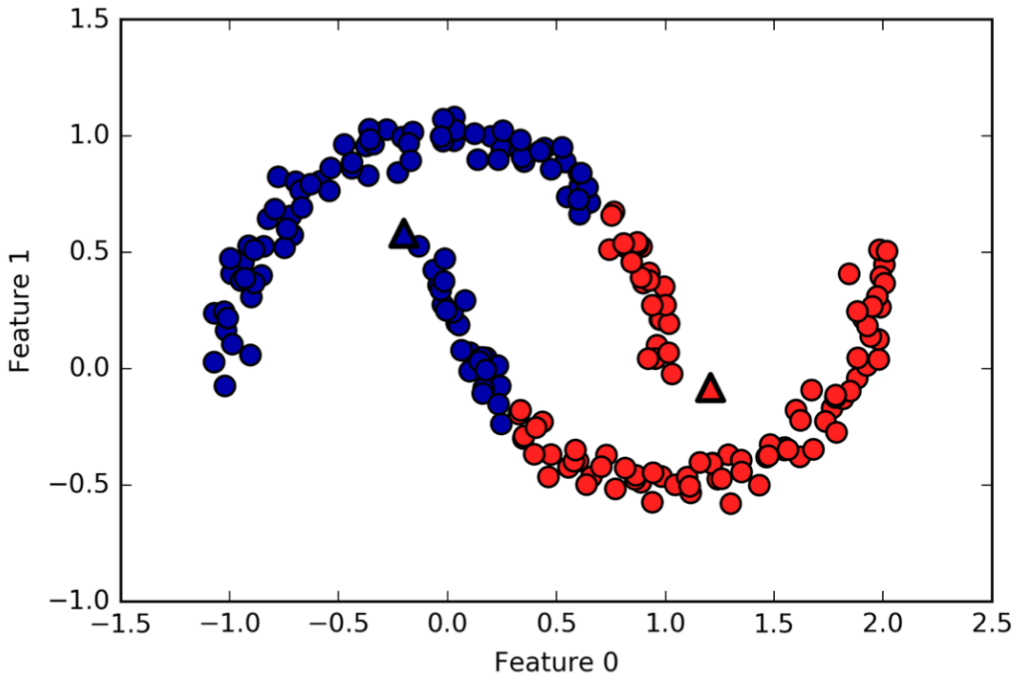
  - 각 클러스터를 정의하는 것이 중심 하나 뿐이므로 클러스터는 둥근 형태로 나타날 수 밖에 없음
  - 모든 클러스터의 반경이 똑같다고 가정하기 때문에 K-Means 알고리즘은 비교적 간단한 형태만 구분할 수 있음
  - 모든 클러스터의 반경이 똑같다고 가정하기 때문에 클러스터 중심 사이의 정확히 중간에 경계를 그림
    - 클러스터의 밀도가 다를 경우에 K-Means 알고리즘으로 찾은 클러스터링 할당



- 클러스터에서 모든 방향이 똑같이 중요하다고 가정함
  - 원형이 아닌 클러스터를 구분하지 못하는 K-Means 알고리즘

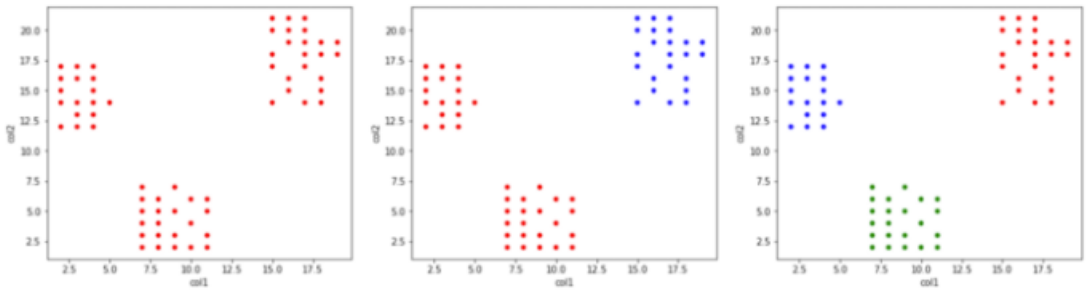


- 복잡한 형태의 데이터는 K-Means의 성능이 더 나빠짐



    - 장점
        - 비교적 이해하기 쉽고 구현도 쉬울 뿐만 아니라 비교적 빠르기 때문에 가장 인기 있는 군집 알고리즘
        - 대용량 데이터셋에도 잘 작동하지만 scikit-learn은 아주 큰 대규모 데이터셋을 처리할 수 있는 miniBatchKMeans도 제공함
    - 단점
        - 무작위 초기화를 사용하여 알고리즘의 출력이 난수 초기값에 따라 달라짐
        - 클러스터의 모양을 가정하고 있어서 활용 범위가 비교적 제한적이며, 또 찾으려 하는 클러스터의 개수를 지정해야 만 한다는 것
- 분리 군집(Divisive Clustering)
    - 설명
        - 시작할 때 하나의 포괄적인 클러스터로 시작(전체 데이터 셋이 하나의 클러스터)하고, 어떤 종료 조건을 만족할 때까지 클러스터를 분할함
        - 하향식 클러스터링 접근 방법(계층적 군집)
    - 원리
        - 처음에는 데이터 세트의 모든 포인트가 하나의 단일 클러스터에 속함

- 클러스터를 가장 유사한 두 개의 클러스터로 분할
- 원하는 수의 클러스터를 얻을 때까지 반복적으로 진행하여 새 클러스터를 형성



(저자별 이미지), 첫 번째 이미지 : 모든 데이터 포인트가 하나의 클러스터에 속하고, 두 번째 이미지 : 1 개의 클러스터가 이전 단일 클러스터에서 분리됨, 세 번째 이미지 : 추가로 1 개의 클러스터가 이전 클러스터 세트에서 분리됩니다.

- 병합군집(Aggomerative Clustering)
  - 설명
    - 상향식 클러스터링 접근 방식(계층적 군집)
    - 시작할 때 각 포인트를 하나의 클러스터로 지정하고, 그 다음 어떤 종료 조건을 만족할 때까지 가장 비슷한 두 클러스터를 합쳐나감
    - scikit-learn에서 사용하는 종료 조건은 클러스터 개수로, 지정된 개수의 클러스터가 남을 때까지 비슷한 클러스터를 합침
    - 2차원 데이터 셋에서 세 개의 클러스터를 찾기 위한 병합 군집의 과정

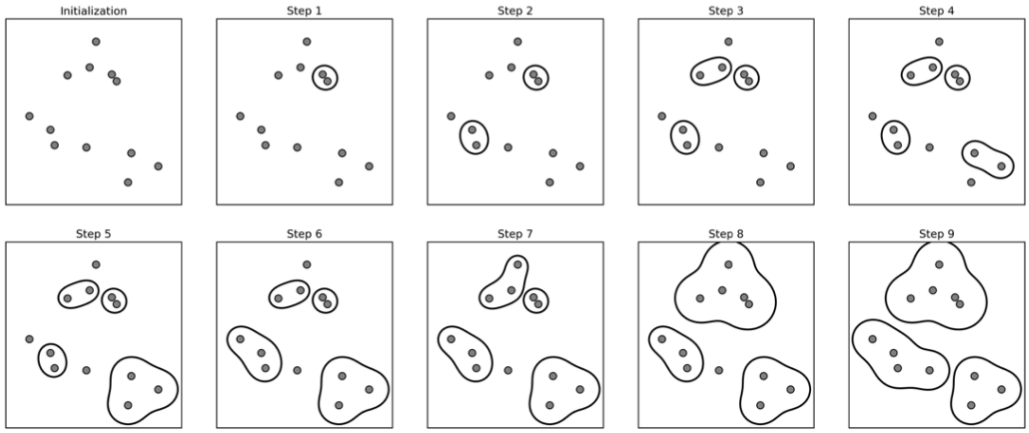


Figure 3-33. Agglomerative clustering iteratively joins the two closest clusters

- 매개변수
  - affinity : 거리 함수
    - euclidean : linkage가 ward라면 무조건 euclidean
    - l1
    - l2
    - manhattan
    - cosine
    - precomputed
  - linkage : 가장 비슷한 클러스터를 측정하는 방법을 지정
    - ward : 모든 클러스터 내의 분산을 가장 작게 증가시키는 두 클러스터를 합침.
      - 크기가 비교적 비슷한 클러스터가 만들어짐
      - 병합되는 클러스터의 분산을 최소화
      - 대부분의 데이터셋에 알맞아서 많이 사용
    - complete : 두 세트의 모든 관측치 간의 최대 거리를 사용
      - 클러스터에 속한 포인트 수가 많이 다를 때는 average나 complete가 더 나을 수 있음
    - average : 두 세트의 각 관측치 거리의 평균을 사용
    - single :두 세트의 모든 관측치 사이의 최소 거리를 사용
- 장점
  - 전체 데이터의 분할 계층도를 만들어주며 덴드로그램을 사용해 손쉽게 확인 가능
  - 클러스터 개수를 지정하지 않아도 됨
- 단점
  - 알고리즘 특성상 병합 군집은 새로운 데이터 포인트에 대해서는 예측을 할 수 없음
  - 복잡한 데이터셋을 구분하지 못함

    - 두 클러스터를 합치기로 결정했으면 돌이킬 수 없기 때문에(반복할 수 없음) 불안정함
    - 계산 시간 소모가 큼
    - 성능이 데이터 스케일링에 매우 의존함
- 계층군집분석(Hierarchical Clustering)과 덴드로그램
    - 병합 군집으로 생성한 계층적 군집과 번호가 매겨진 데이터 포인트

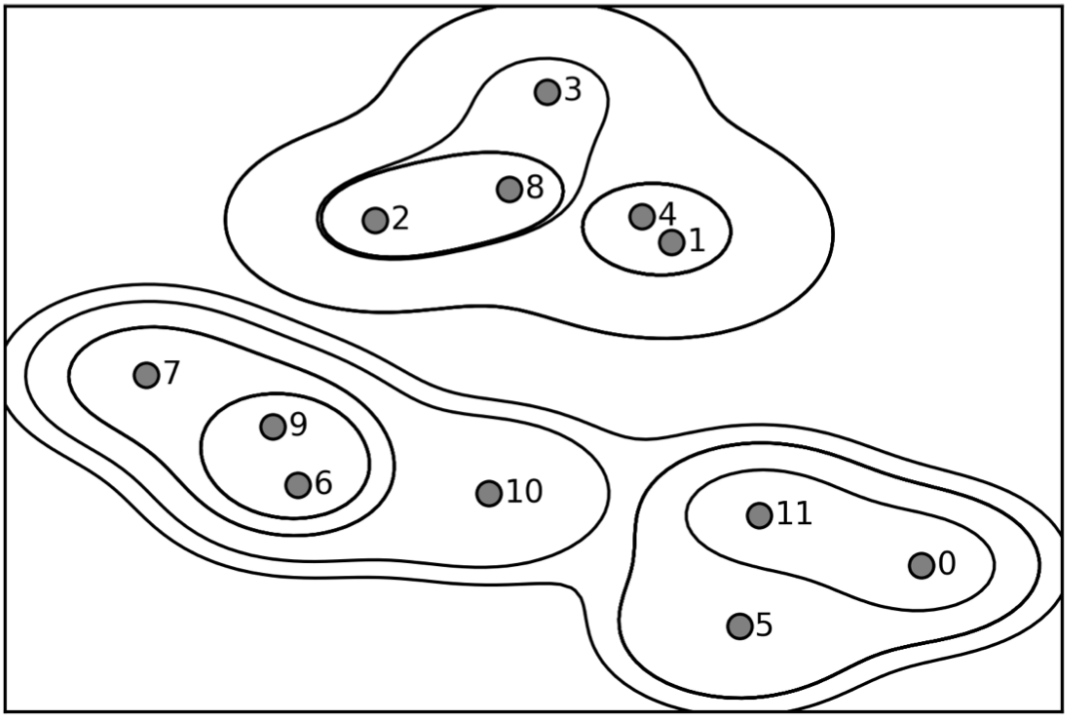
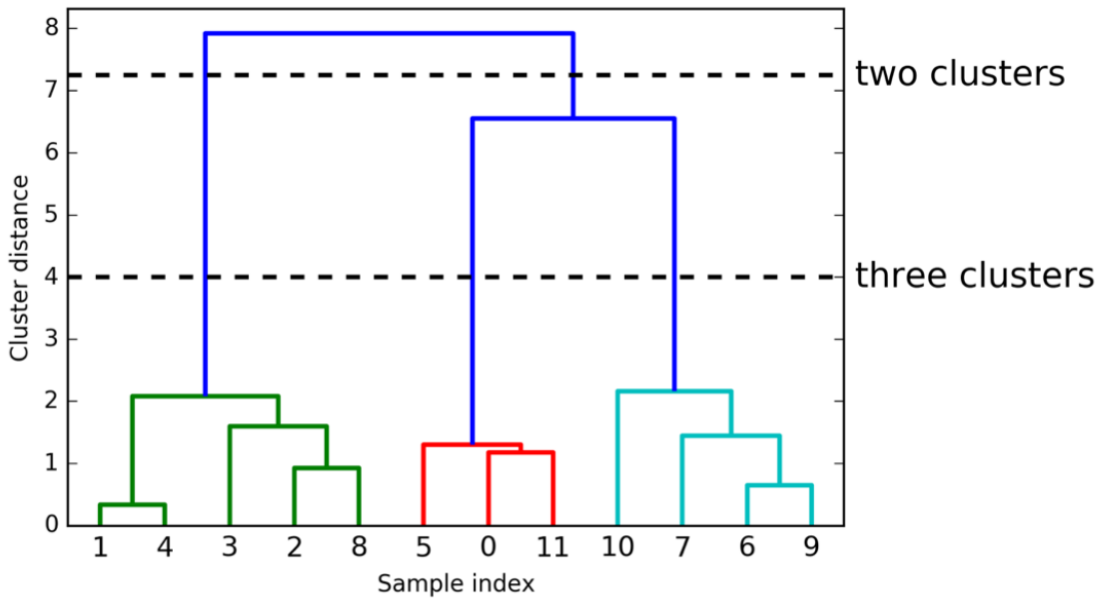


Figure 3-35. Hierarchical cluster assignment (shown as lines) generated with agglomerative clustering, with numbered data points (cf. Figure 3-36)



- DBSCAN(Density-based spatial clustering of applications with noise)
  - 설명
    - 특성 공간에서 가까이 있는 데이터가 많아 붐비는 지역의 포인트(밀집 지역)를 찾음
    - 데이터의 밀집 지역이 한 클러스터를 구성하며 비교적 비어있는 지역을 경계로 다른 클러스터와 구분됨
    - 밀집 지역에 있는 포인트를 핵심 샘플(=핵심 포인트)이라고 하며 두개의 매개변수 min_samples와 eps가 있음
    - 한 데이터 포인트에서 eps거리 안에 데이터가 min_samples 개수 만큼 들어 있으면 이 데이터 포인트를 핵심 샘플로 분류함
    - eps보다 가까운 핵심 샘플은 DBSCAN에 의해 동일한 클러스터로 합쳐짐
  - 원리
    - 시작할 때 무작위로 포인트를 선택한 후 그 포인트에서 eps 거리 안의 모든 이웃을 살펴봄
    - 만약 어떤 클러스터에도 할당되지 않았다면 바로 전에 만든 클러스터 레이블을 할당함
    - 만약 핵심 샘플이면 그 포인트의 이웃을 차례로 방문함

    - 이런 식으로 계속 진행하여 클러스터는 eps 거리 안에 더 이상 핵심 샘플이 없을 때까지 자라남
    - 그런 다음 아직 방문하지 못한 포인트를 선택하여 다음과 같은 과정을 반복함
    - 한 데이터셋에 DBSCAN을 여러번 실행하면 핵심 포인트의 군집은 항상 같고 매번 같은 포인트를 잡음으로 레이블함
    - 그러나, 경계 포인트는 한 개 이상의 클러스터 핵심 샘플의 이웃일 수 있기 때문에 경계 포인트가 어떤 클러스터에 속할지는 포인트를 방문하는 순서에 따라 달라짐
    - 보통 경계 포인트는 많지 않으며 포인트 순서 때문에 받는 영향도 적어 중요한 이슈는 아님
- 포인트의 종류
    - 핵심 포인트
    - 경계 포인트
        - 핵심 포인트에서 eps거리 안에 있는 포인트
    - 잡음 포인트
        - 하얀색
- min_samples와 eps 매개변수를 바꿔가며 DBSCAN으로 계산한 클러스터 할당

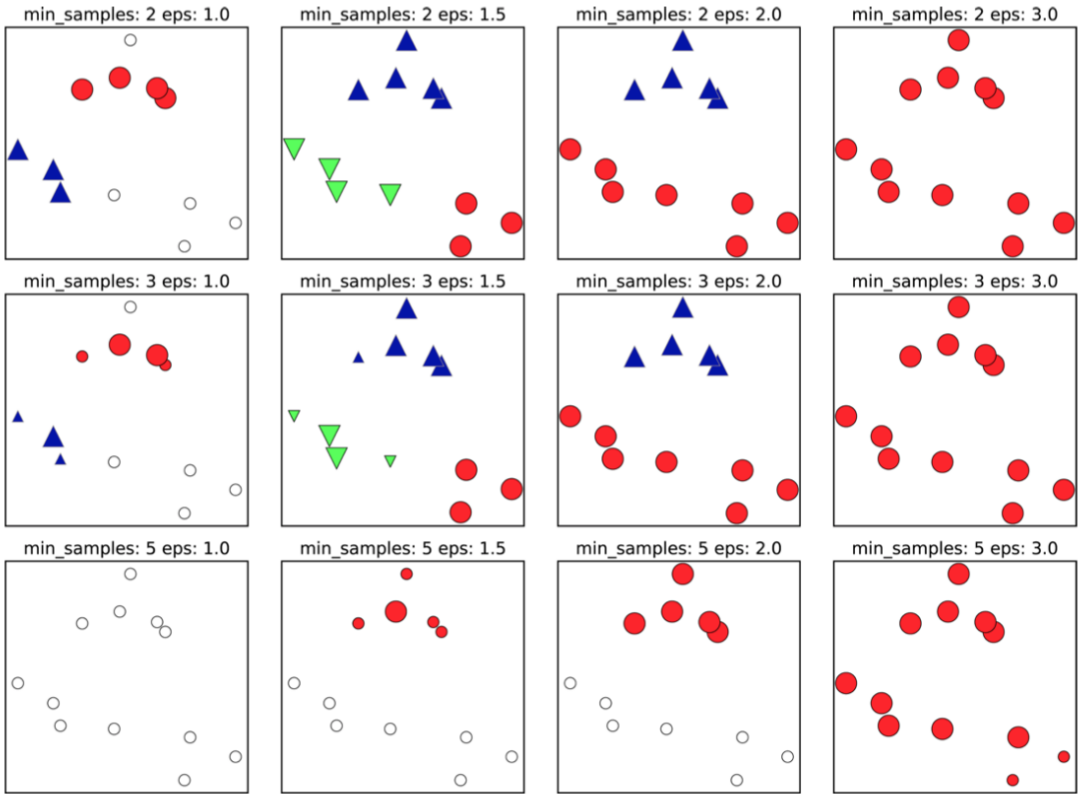


Figure 3-37. Cluster assignments found by DBSCAN with varying settings for the min_samples and eps parameters

- eps
  - eps를 증가시키면 하나의 클러스터에 더 많은 포인트가 포함되어 클러스터를 커지게 하지만 여러 클러스터를 하나로 합치게도 만듬
  - eps 매개변수는 가까운 포인트의 범위를 결정하기 때문에 더 중요
  - eps를 매우 작게 하면 어떤 포인트도 핵심 포인트가 되지 못하고, 모든 포인트가 잡음 포인트가 될 수 있음
  - eps를 매우 크게하면 모든 포인트가 단 하나의 클러스터에 속하게 될 것임
  - 간접적으로 몇 개의 클러스터가 만들어질지 제어함
  - 적절한 eps값을 찾으려면 StandardScaler나 MinMaxScaler로 모든 특성의 스케일을 비슷한 범위로 조정해주는 것이 좋음
- min_samples
  - min_samples를 키우면 핵심 포인트 수가 줄어들며 잡음 포인트가 늘어남

        - 덜 조밀한 지역에 있는 포인트들이 잡음 포인트가 될 것인지, 아니면 하나의 클러스터가 될 것인지를 결정하는데 중요한 역할을 함
        - min_samples를 늘리면 min_samples수보다 작은 클러스터들은 잡음 포인트가 됨
        - 클러스터의 최소 크기를 결정함
    - 장점
        - 클러스터의 개수를 미리 지정할 필요가 없음
        - 복잡한 형상도 찾을 수 있음
        - 어떤 클래스에도 속하지 않는 포인트를 구분할 수 있음
        - 비교적 큰 데이터셋에도 적용 가능
        - 크기가 많이 다른 클러스터를 만들어냄
    - 단점
        - 병합 군집이나 K-Menas 보다는 다소 느림
- EM 알고리즘(Expectation Maximization)
- 그래프 기반 클러스터링 기법(Spectral Clustering)
- Birch
- SOM
- Mean Shift Clustering
- Affinity Propagation
- Optics(A Cluster-Ordering Method)

군집 알고리즘의 비교와 평가

- 지표
    - 타깃값으로 군집 평가하기
        - ARI(Adjusted rand index)와 NMI(normalized mutual information)

- 클러스터 레이블은 그 자체로 의미가 있는 것이 아니며 포인트들이 같은 클러스터에 속해 있는가만이 중요함
- ARI 점수 비교

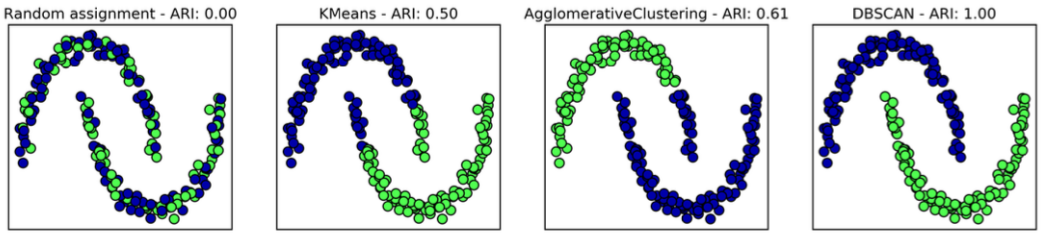


Figure 3-39. Comparing random assignment, k-means, agglomerative clustering, and DBSCAN on the two_moons dataset using the supervised ARI score

- 타깃값 없이 군집 평가하기
  - 실루엣 계수(Silhouette coefficient)
    - 실제로 잘 작동하지 않음
    - 클러스터의 밀집 정도를 계산하는 것
    - 높을수록 좋으며 최대 점수는 1
    - 밀집된 클러스터가 좋긴 하지만 모양이 복잡할 때는 밀집도를 활용한 평가가 잘 들어맞지 않음
    - 실루엣 계수 비교

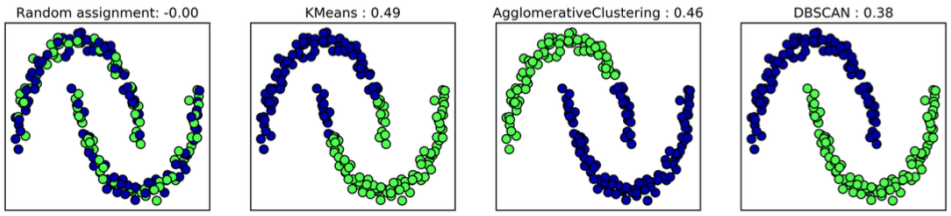


Figure 3-40. Comparing random assignment, k-means, agglomerative clustering, and DBSCAN on the two_moons dataset using the unsupervised silhouette score—the more intuitive result of DBSCAN has a lower silhouette score than the assignments found by k-means

    - DBSCAN의 점수가 더 낮지만 Kmeans의 실루엣 점수가 더 높음

- 클러스터 평가에 더 적합한 전략은 견고성 기반의 지표임
- 군집 모델이 매우 안정적이거나 실루엣 점수가 높다고 하더라도 군집에 어떤 유의미한 것이 있는지 또는 군집이 데이터의 흥미로운 면을 반영하고 있는지는 여전히 알 수 없음
- 클러스터가 우리 기대에 부합하는지 알 수 있는 유일한 방법은 클러스터를 직접 확인하는 것 뿐임

참고

- 블로그
  - https://muzukphysics.tistory.com/108
  - https://leedakyeong.tistory.com/entry/군집분석이란-What-is-clustering-algorithm
  - https://m.blog.naver.com/PostView.nhn?blogId=slykid&logNo=221970736285&isFromSearchAddView=true
  - https://genius12.tistory.com/229
  - https://sosal.kr/1055
  - https://techblog-history-younghunjo1.tistory.com/93
  - https://datascienceschool.net/03 machine learning/16.01 군집화.html
    - 필수
- 책
  - 파이썬 라이브러리를 활용한 머신러닝