K평균 군집화

이론

- n개의 샘플로 구성된 피처 행렬X를 중심이 K개의 군집으로 분할

- K개의 중심을 임의로 고른다.

- 각 샘플에서 가장 가까운 중심을 해당 샘플을 포함한 군집의 중심으로 정한다.

- 군집별 샘플 피처값의 평균을 계산하여 이를 군집의 새로운 중심으로 업데이트 한다.

- 더는 중심이 변하지 않거나 변화가 임계값 이하일 때까지 반복한다.

1. 하이퍼파라미터

- n_clusters : 생성할 클러스터 수
- init : 초기화 방법 선택
  - k-means++ : 수렴 속도를 높이기 위한 초기 중심값을 설정 #default
  - random : 임의로 초기 중심값 선택
- n_init : 설정된 알고리즘을 반복하는 횟수.
- max_iter : 각각의 알고리즘 반복에서 이터레이션의 상한값
- tol : 조기 종료의 허용 오차
- algorithm : 학습에 사용할 알고리즘
  - lloyd : 로이드 알고리즘 (기본적인 EM-알고리즘) #default
  - elkan : 삼각 부등식 이용하는 엘칸 알고리즘을 적용, 메모리 소모가 크다 !

실제 사용 팁

- 거리 기반이기 때문에 피처를 스케일링 해야 좋다.

- tol, max_iter 등 때문에 완전히 수렴하기 전에 알고리즘이 멈춘다면 중심이 각각의 클러스터의 평균과 다를 수 있으므로 조심해야 한다.

계층적 군집화 모델 이론

- 병합적 군집화에서는 가까운 군집끼리 묶어 새로운 군집을 형성하므로 두 점 사이의 거리를 정의한 후 두 점 사이의 거리를 이용하여 군집 사이의 거리를 정의해야 한다.

- 두 샘플 사이의 거리

  - 두 샘플이 주어졌을 때 그 두 샘플 사이의 거리 메트릭을 정의/ 보통 유클리드, L1, L2, 코사인 거리 등을 사용

- 군집 사이의 거리

  - 연결법은 샘플 간 거리를 정의한 상태에서 두 군집 사이의 거리를 정의한 것으로, 단일 연결법, 최장 연결법, 평균 연결법, 와드 연결법 등 다양한 방식이 있음

<하이퍼파라미터>

- n_clusters (default: 2) -> 생성할 군집 수
- affinity (default: 'euclidean') -> 연결에 사용할 메트릭
  - 'percomputed' 사용자 지정 거리 행렬 사용. 이때 fit() 메서드가 유사도 행렬 대신 거리 행렬을 인자로 받게 된다.
  - 그 외 'l1', 'l2', 'manhattan', 'cosine'이 있음
- linkage (default: ward) -> 연결법 선택, 각각 와드 연결법, 완전 연결법, 평균 연결법, 단일 연결법을 의미함. 'ward'일 때는 affinity = 'euclidean'로 설정해야 한다.
  - Complete Linkage (최장 연결): 두 군집 간 최대 거리를 사용하여 클러스터 간의 거리를 측정
  - Single Linkage (최단 연결): 두 군집 간 최소 거리를 사용하여 클러스터 간의 거리를 측정
  - Average Linkage (평균 연결): 두 군집 간의 평균 거리를 사용하여 클러스터 간의 거리를 측정
  - Centroid Linkage (중심 연결): 두 군집의 중심 간 거리를 사용하여 클러스터 간의 거리를 측정
- distance_threshold (default: None) -> 군집화 기준 거리 설정. 군집의 범위를 제어하는 값
  - None 모든 거리에 대해 군집화를 수행하여 데이터 전체가 1개의 군집으로 병합될 때까지 계층 구조를 형성함
  - float 거리가 이 값 이상인 군집은 병합하지 않는다. n_clusters가 None이어야 동작함

실제 활용 팁

-계층적 군집화 모델은 군집 개수를 먼저 결정하지 않고 계층의 형태를 보고 사후에 군집 개수를 결정할 수 있다는 장점 이를 위한 방법으로 덴드로그램을 그려 계층을 시각화가능

-분석 방법은 크게

n_cluster에 적절한 값을 설정하는 경우

vs

n_cluster를 none으로 하고 distance_tresthold를 설정하여 그 거리 내에서 군집화 하는 경우

PCA

- 차원 축소 : 주어진 피처 공간의 차원 수를 줄이는 기법 -> 차원의 수가 증가할 경우 차원의 저주 현상 발생

- 일반적으로 샘플 수가 피처 수에 대해 지수적으로 증가해야함

- 피처 선택법 vs 피처 추출법

- 피처 선택법 : 전체 피처의 집합에서 일부 부분 집합을 선택하는 방법 => 머신러닝 방법론 이용

- 피처 추출법 : 주어진 데이터의 정보를 최대한 보존하는 새로운 피처 조합을 생성하는 방법 ex)PCA, Isomap, LLE, t-SNE

- n_components : 추출한 주성분의 수
  - None(default),float,mle
- whiten : 분산 조정 여부
  - False(default)
- svd_solver : 최적화 알고리즘 선택
  - full, arpack, randomized, auto(default)
- tol : svd_solver가 'arpack'일 때 유지할 특잇값에 대한 허용 요차를 설정
  - 0.0(default)
- iterated_power : svd_solver가 'randomized'일 때의 이터레이션 횟수
  - auto(default)
- random_state : svd_sover가 'arpack' 이나 'radnomized'일 때 랜덤성을 제어하고자 사용
  - None(default)

실제 사용 시 활용 팁

- PCA 클래스는 자동으로 평균 줌심화를 수행하므로 이 과정은 전처리에서 수행하지 않아도 괜찮지만 whitten 하이퍼파라미터의 기본값이 false이므로 분산의 스케일링은 PCA 클래스에서 이루어지지않음

- PCA 클래스 객체는 fit() 메서드를 이용해 n개의 주성분을 학습한다. 학습 후 객체는 transform() 메서드를 이용해 테스트 데이터로 학습한 결과 PCA를 수행 가능