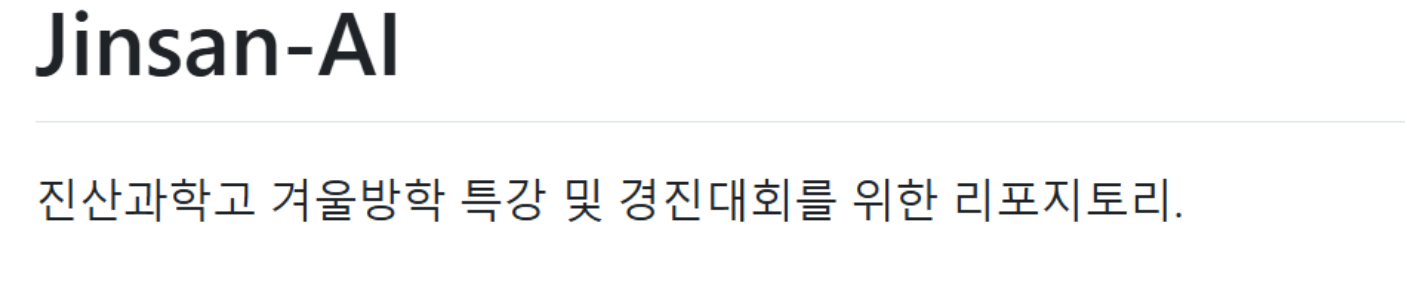
인천 진산과학과 특강
3일차

Prof. Jibum Kim (jibumkim@inu.ac.kr)

Department of Computer Science & Engineering

Incheon National University

- https://github.com/DevSlem
- **Repositories 클릭**
- **Jinsan-AI 클릭**
- **카톡 오픈채팅방 입장: 질의 응답/ 필요한 코드 등 요청 가능, 다음주 경진대회 시 질의 응답 가능**
- **1~2일차 정답파일 업로드 완료**



- 오늘 특강 내용
- 1. 트리 기반 ML (결정 트리), 별도 pdf 파일
- 2. 앙상블 기반 ML (random forest), 별도 pdf 파일
- 3. Cross validation
- 4. Fine-tuning and grid search, 별도 pdf 파일
- 5. 비지도학습 (clustering, 군집화)

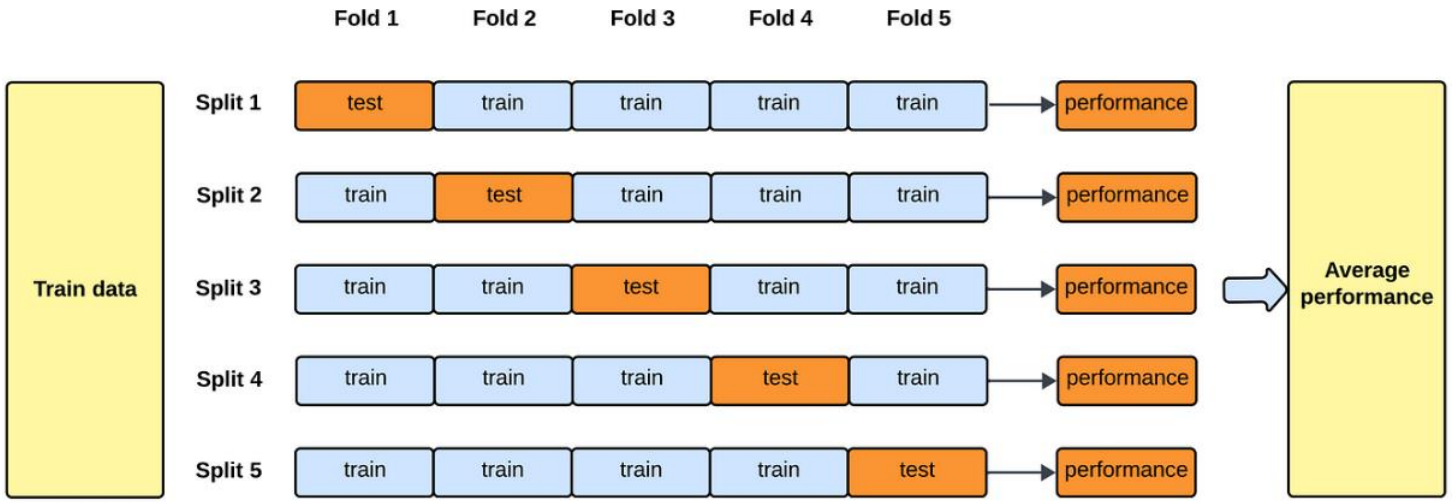
특강 내용

Cross validation (교차 검증)

- **Cross validation (교차 검증):** ML 모델에서 모델의 성능을 평가하고 overfitting을 방지하기위한 기법

- **목적:** 데이터를 여러 번 분할하여 학습과 평가를 반복적으로 수행함으로써, 데이터 분할 방식에 따른 성능 편향을 줄이고 일반화 성능을 더 정확히 측정

- **K-fold cross validation**
- **훈련 (train) 데이터를 K개로 나누고, 각 fold를 검증 데이터로 사용하여 K 번 반복**
- **모든 훈련 데이터에 대해서 다 test해보고 train도 시켜봄**
- **예: 5-fold cross validation**



- Cross validation이 필요한 이유
- 1. over-fitting 방지
- 훈련 데이터를 여러 번 나누어 ML 모델을 평가하므로, 특정 훈련 데이터에만 적합한 모델이 되는 것을 방지
- 2. 모델의 일반화 성능 평가
- 교차 검증을 통해 모델이 특정 훈련 데이터 뿐만 아니라 unseen 데이터에 대해서도 성능이 잘 유지되는지 확인
- 3. 모델 비교
- 여러 ML 모델들 성능을 비교시에 객관적 비교
- 4. 훈련 데이터의 효율적인 활용 가능
- 여러 번 나누어 훈련과 테스트를 반복하므로 데이터가 더 효과적으로 사용됨

- **실습:Iris dataset에 대하여 KNN 모델 사용**
- **Cross validation 적용**

```
# 3. 5-Fold Cross Validation 수행
cv_scores = cross_val_score(tree_clf, X, y, cv=5, scoring='accuracy')
```

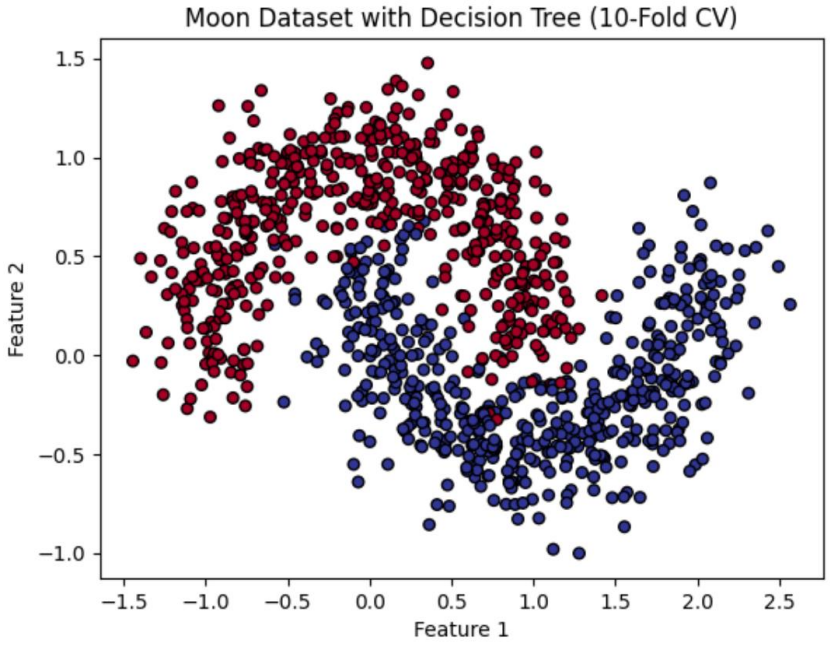
```
Cross-Validation Scores: [0.96666667 0.96666667 0.9        0.93333333 1.        ]
Mean Accuracy: 0.9533
Standard Deviation: 0.0340
```

- **ML 모델에서 여러 개의 하이퍼 파라미터가 존재하는 경우 cross validation과 뒤에 배울 grid search를 같이 사용하여 ML 모델의 어떤 파라미터가 분류기 성능이 가장 좋은지 확인할 수 있음**

- **실습**
- **Scikit-learn에는 moon dataset이라는 가상의 데이터셋을 만들수 있다**
- **이는 “make two interleaving half circles”를 만드는 것이다**
- **make_moons — scikit-learn 1.6.0 documentation**
- **간단한 ML 연습용으로 많이 쓰인다**

- **Moon dataset 만드는 코드**
- **데이터 샘플 수, noise 정도등을 조절 가능**

```
# 1. Moon 데이터셋 생성
X, y = make_moons(n_samples=1000, noise=0.2, random_state=42)
```



특강 내용

비지도학습과 군집화

- **Although most of the applications of ML today are based on supervised learning, the vast majority of the available data is unlabeled: we have the input features X, but we do not have the labels y.**
- **Unsupervised learning (비지도 학습)**
- **데이터에 대한 정답 (label)이 없는 경우에 학습을 진행하는 ML 기법**
- **데이터가 입력 값으로만 제공되고, 이에 대한 정답이 주어지지 않은 상태에서 데이터를 이해**

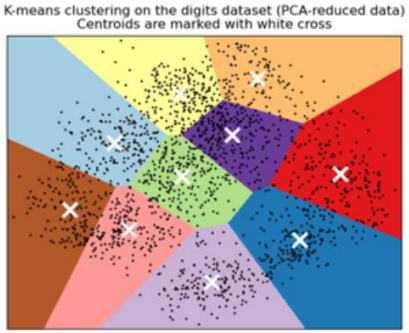
- **Scikit-learn에 가면 Clustering**
- **목표: automatic grouping of similar objects into sets**

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.

Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



- 비지도 학습의 목표에는 여러 가지가 있겠지만 가장 중요한 것은 데이터의 패턴 탐지이다
- 이는 군집화 (clustering)이란 방법을 통해서 가능하다
- Clustering is the task of grouping a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups
-

군집화 사용 예시 1: 고객 분류

- 어떤 사람이 몇년간 쇼핑몰을 성공적으로 운영해 왔다 (수백만명의 고객을 가지고 있다). 그 동안 한 종류의 홍보 판플렛을 만들어 발송했는데 이제부터는 고객의 취향을 분석하여 4-6종류의 판플렛을 만들어 맞춤 홍보를 하고자 한다. 일종의 개인화 (personalization) 홍보 전략이다. 고객에 대한 각종 정보는 DB에 저장되어 있어 이것을 기초 자료로 활용 가능하다
- 고객 정보는 월평균 구매액, 선호하는 물품의 종류와 수준, 결제 방법, 반품 성향, 직업, 성별, 나이, 거주 지역등 다양하다.
- Q) 수백만명의 고객이 있을때 이를 어떻게 4-6개의 그룹으로 나눌수 있을까?
- 해결 방법: 유사한 샘플(데이터)들끼리 모아서 4-5개의 그룹으로 만든다

- **군집화 사용 예시 2: 이상치 detection**
- **If you have clustered the users of your website based on their behavior, you can detect users with unusual behavior, such as an unusual number of requrests per second**
- **Anomaly detection is particularly useful in detecting defects in manufacturing, or for fraud detection**

- **금융 및 사기 탐지 (Fraud Detection)**
- 금융 거래 데이터를 클러스터링하여 정상 거래와 비정상 거래를 그룹화합니다. 클러스터 중심에서 멀리 떨어진 데이터는 사기 가능성이 높은 거래로 간주됩니다.
- 예시:
- 신용카드 거래: 클러스터링으로 일반적인 구매 패턴과 다른 거래(예: 비정상적으로 큰 금액, 낯선 위치에서의 거래)를 탐지.
- 보험 청구: 클레임 데이터를 클러스터링하여 의심스러운 보험 청구를 식별.

- **네트워크 보안 (Network Security)**
- **네트워크 트래픽을 클러스터링하여 정상적인 트래픽 패턴과 이상 패턴(해커 공격, DDoS 등)을 구분.**
- **정상 클러스터에서 멀리 떨어진 점을 잠재적 공격으로 간주.**
- **예시:**
- **침입 탐지 시스템(IDS): 정상적인 사용자 행동과 다른 트래픽을 이상치로 감지.**
- **로그 분석: 서버 로그 데이터를 분석해 비정상적인 로그인 시도나 활동을 탐지**

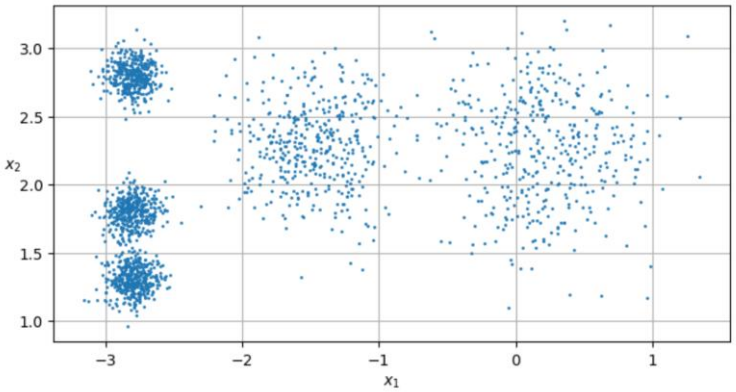
- **군집화 사용 예시 3: semi-supervised learning (반지도 학습)**
- **If you have a few labels, you could perform clustering and propagate the labels to all the instances in the same cluster.**
- **This technique can greatly increase the number of labels available for a subsequent supervised learning algorithm, and thus improve its performance**

- **Scikit-learn의 make_blobs**

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

- **무작위 데이터 생성시, 특히 정규분포 형태의 데이터를 생성하여 분류기/군집화 테스트 시에 사용 가능**

- **K-means 알고리즘과 군집화**
- 아래 데이터를 보면 분명하게 5개의 밀집된 군집 (blob)를 볼 수 있다
- 아래 데이터에 대하여 군집화 알고리즘 중에 가장 널리 쓰이는 K-means 알고리즘을 수행해 보려고 한다
- K-means 알고리즘에서는 군집의 개수인 K를 정해야 하는데 아래와 같이 군집의 개수가 명확하기 때문에 K=5로 정했다
- 실습파일



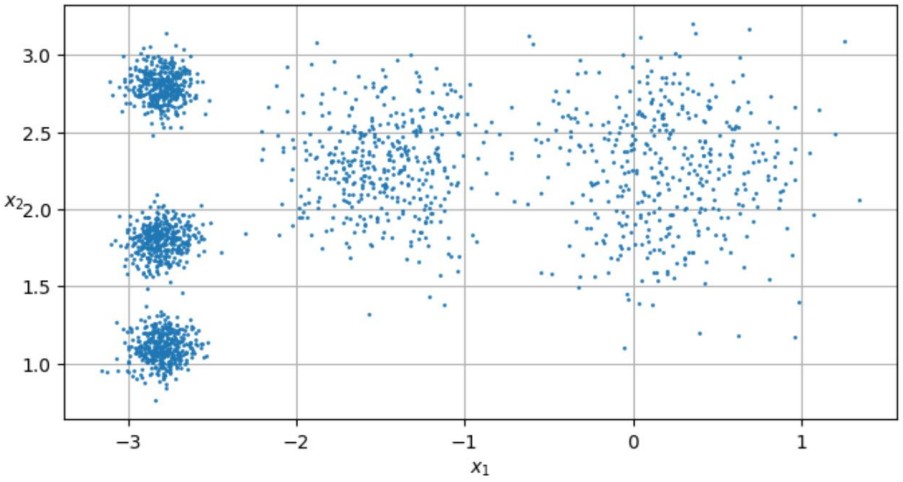

```
k = 5
kmeans = KMeans(n_clusters=k, n_init=10, random_state=42)
y_pred = kmeans.fit_predict(X)
```

- K-means 알고리즘을 수행하면 각각의 데이터가 5개의 cluster (군집) 중에 하나에 속하게 된다
- 즉, 0부터 4까지 5개의 군집 중에 하나에 속함

```
y_pred
```

```
array([0, 0, 4, ..., 3, 1, 0], dtype=int32)
```

- 또한, K-means 알고리즘을 수행하면 각 군집의 중심 위치 (centroid), 군집 중심도 출력해준다



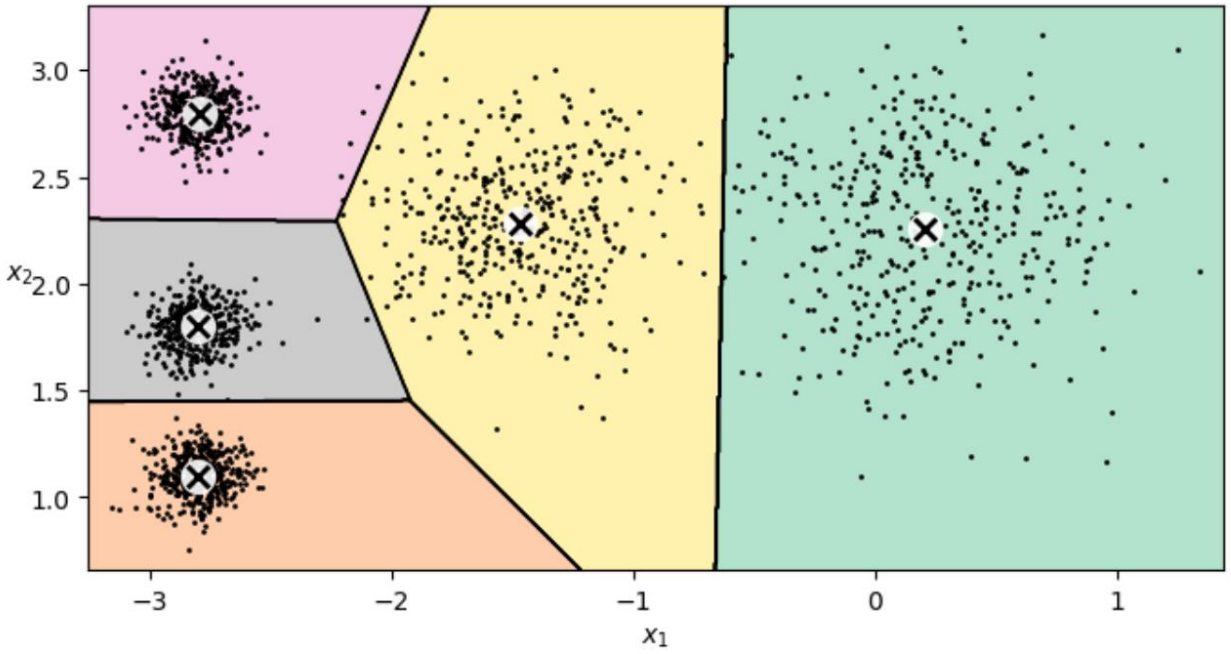

```
[18] kmeans.cluster_centers_

array([[ 0.20876306,  2.25551336],
       [-2.80052143,  1.10046968],
       [-2.79290307,  2.79641063],
       [-1.46679593,  2.28585348],
       [-2.80374386,  1.79979459]])
```

- **Clustering이 끝나면 새로운 데이터에 대해서도 predict도 가능하다**
- **Q: 올바른 예측인가?**

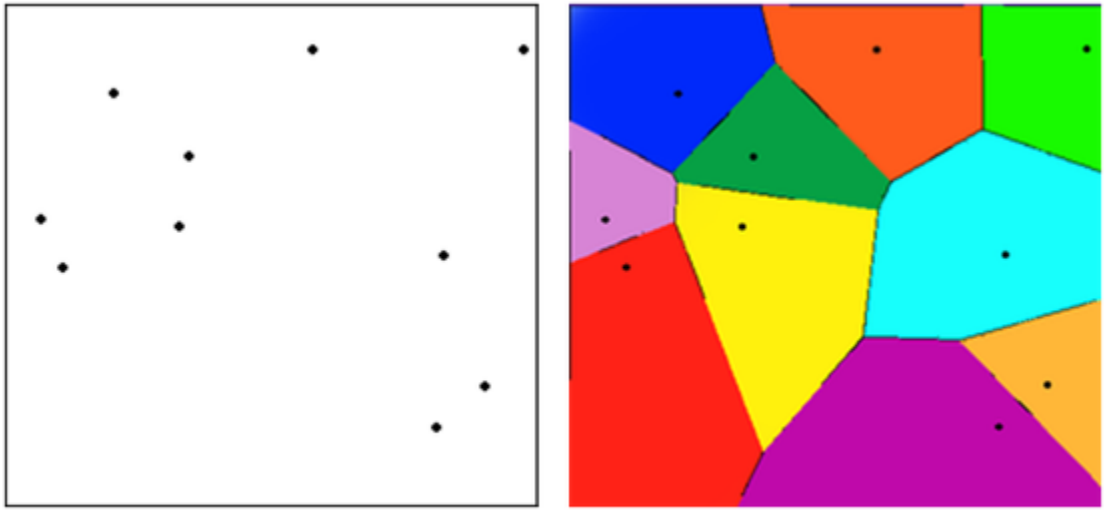
```
X_new = np.array([[0, 2], [3, 2], [-3, 3], [-3, 2.5]])
kmeans.predict(X_new)
```

```
array([0, 0, 2, 2], dtype=int32)
```

- **Clustering의 결정 경계도 확인 가능하다**



- **Voronoi diagram (보로노이 다이어그램)**
- 공간을 여러 개의 영역으로 나누는 수학적 도구
- 점들의 집합에 대해서 각 점으로부터 가장 가까운 모든 위치를 포함하는 영역을 정의



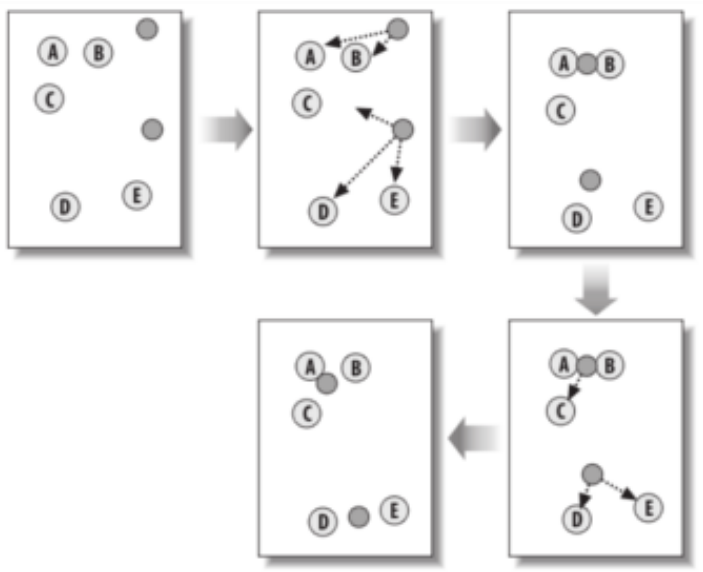
<그림 2> 그림출처: 위키피디아[1]

- **1. 지리학 및 위치 분석**
- **응용 사례:**
  - 도시 계획: 도시 내에서 병원, 소방서, 경찰서와 같은 공공시설이 담당하는 범위를 분석.
  - 배달 서비스: 각 배달 센터에서 담당할 최적의 지역을 계산.
  - 기상학: 특정 날씨 관측소의 관측 범위를 정의.
- **예시: 보로노이 다이어그램으로 각 공공시설이 담당하는 지역을 시각적으로 나타내면, 서비스가 제대로 분배되는지 확인할 수 있습니다.**

- **2. 통신 및 네트워크**
- **응용 사례:**
  - 셀룰러 네트워크: 기지국(Cell Tower)의 서비스 범위를 정의.
  - 무선 센서 네트워크: 각 센서가 커버하는 영역을 계산.
  - Wi-Fi 배치 최적화: 각 라우터의 커버리지가 최대화되도록 조정.
- **예시: 기지국의 위치와 커버리지 맵을 보로노이 다이어그램으로 나타내어 신호 겹침 문제를 해결.**

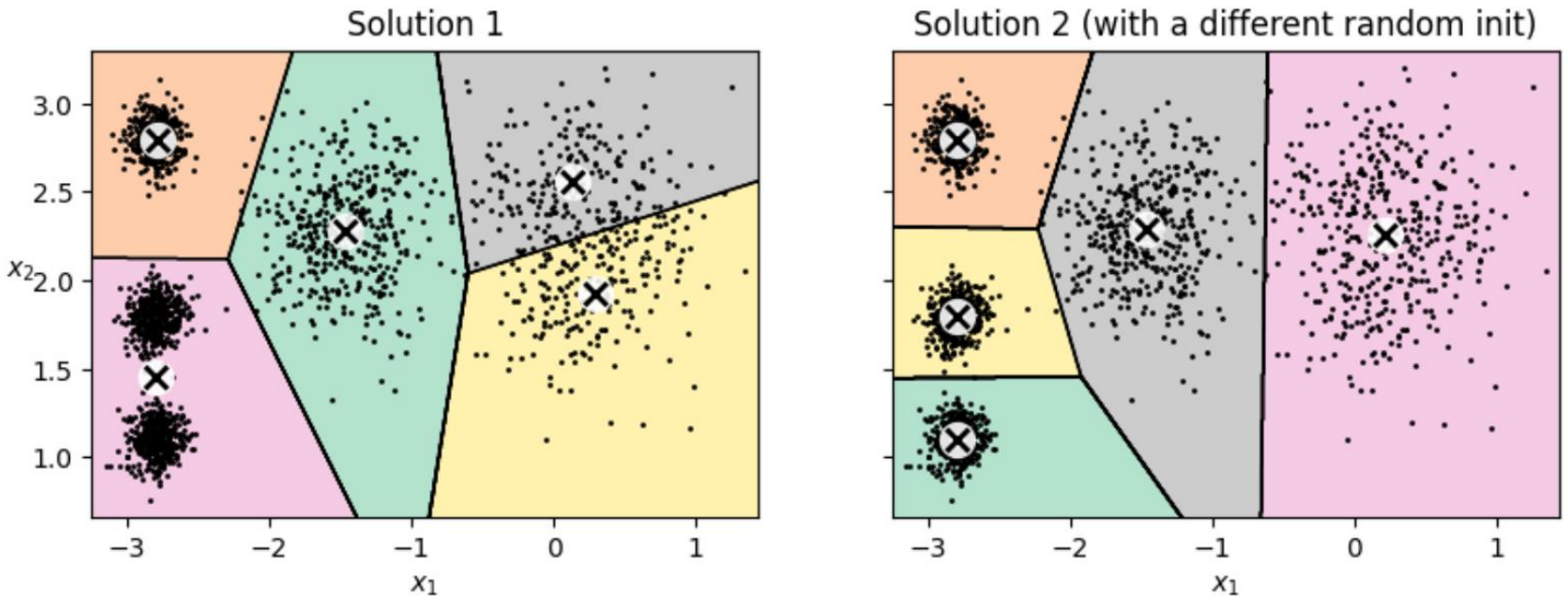
- K-means 알고리즘의 동작
- 1. K개의 군집 중심 (centroid)의 위치를 무작위로 초기화
- 2. Repeat until convergence (centroid의 위치가 더 이상 움직이지 않을때까지 반복)
- 1) Assign each data to the closest centroid
- 2) Update the centroid to be the mean of the instances that are assigned to them

- K=2이고 5개의 데이터 (A, B, C, D, E)가 있는 경우
- 1. 임의의 위치에 군집 중심 2개가 위치
- 2. 각각의 데이터가 가장 가까운 군집 중심을 찾음. 이 경우에는 A, B는 위쪽 군집 중심과 가깝고, C, D, E는 아래쪽 군집 중심과 가까움
- 3. A, B의 평균 위치가 새로운 군집 중심의 위치, C,D, E의 평균 위치가 새로운 군집 중심의 위치
- 4. 다시 2 반복. 이 경우 A, B, C는 위쪽 군집 중심과 가까움. D, E는 아래쪽 군집 중심과 가까움
- 5. 다시 3 반복.
- 언제까지 2,3 반복?



- **K-means 알고리즘의 variability**
- **In the original K-Means algorithm, the centroids are just initialized randomly, and the algorithm simply runs a single iteration to gradually improve the centroids, as we saw above.**
- **However, one major problem with this approach is that if you run K-Means multiple times (or with different random seeds), it can converge to very different solutions, as you can see below:**

- 최초 centroid의 위치에 따라서 K-means clustering의 결과는 달라짐



- 해결 방법?
- 1. centroid의 위치를 대략적으로 알고 있다면 최초 hypterparameter에서 centroid의 위치를 설정 가능
- 하지만, 대부분 어려움. 특히, 고차원 데이터 의 경우 더 어려움

- 2. 여러 개의 무작위 centroid의 위치에서 시작하여 그 중에 best solution을 찾음. hyperparameter로 설정 가능

n_init : *'auto' or int, default='auto'*

Number of times the k-means algorithm is run with different centroid seeds.

- **K-means 알고리즘의 평가지표**
- **Inertia: 각 데이터와 그것이 속한 centroid의 거리 제곱합 (Sum of squared distance)으로 정의**
- **Scikit-learn에서 출력 가능**

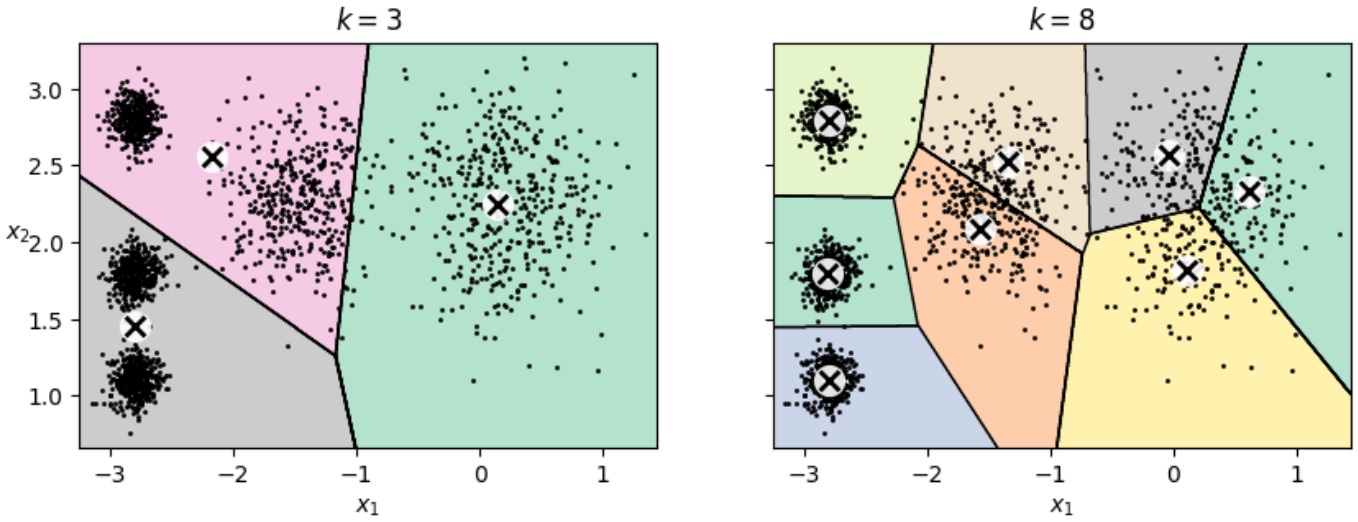
```
kmeans.inertia_
211.80447602862358
```

$$\text{Inertia} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

여기서:

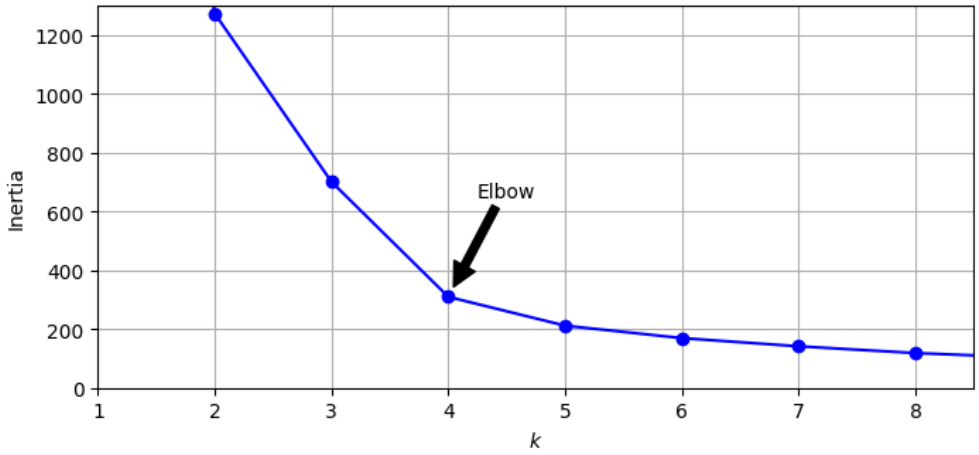
- k: 클러스터 개수
- C_i: i-번째 클러스터
- x: 클러스터 C_i에 속한 데이터 포인트
- μ_i: C_i의 중심(클러스터 센트로이드)
- $\|x - \mu_i\|^2$: 데이터 포인트 x와 클러스터 중심 μ_i 사이의 거리 제곱

- **최적의 군집 수 찾기**
- **In general, it will not be so easy to know how to set k (군집 수), and the result might be quite bad if you set it to the wrong value**
- **See the below figure that set k to 3 or 8, that results in fairly bad models.**



- **1. Elbow method**
- **You might be thinking that we could just pick the model with the lowest intertia**
- **Why not? Intertia 값 변화**
- **k=3: 653.2, k=5: 211.6, k=8: 119.1**
- **But, k가 증가함에 따라서 inertia는 계속 작아짐**

- k가 4까지는 inertia가 급격히 감소하지만 5부터는 상대적으로 천천히 감소한다
- This curve has roughly the shape of an arm, and there is an “elbow” at k=4
- This method is called an “elbow” method



2. Silhouette score 사용

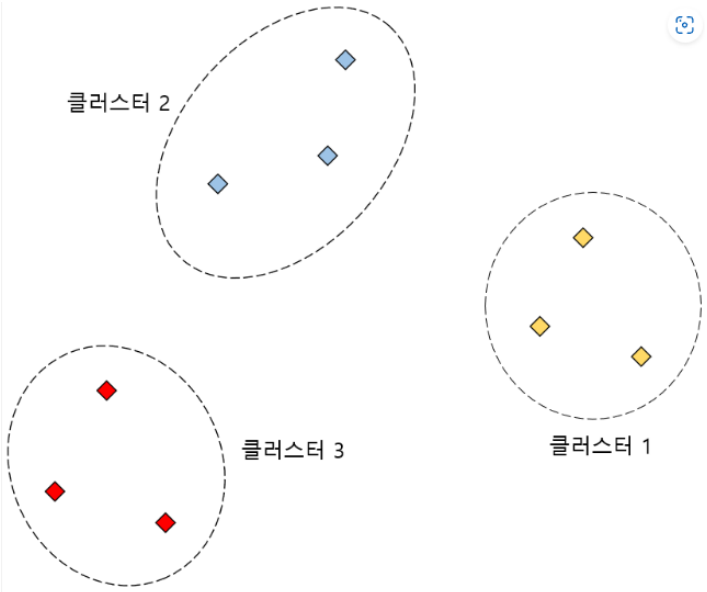
- **Silhouette score란? 데이터들이 자신이 속한 군집내에서 얼마나 잘 밀집되어 있고, 다른 군집과 얼마나 명확히 분리되어 있는지를 나타낸다. 이 지표는 데이터 하나하나에 대해 계산되고, 전체 데이터에 대해 평균값으로 최종 계산됨**

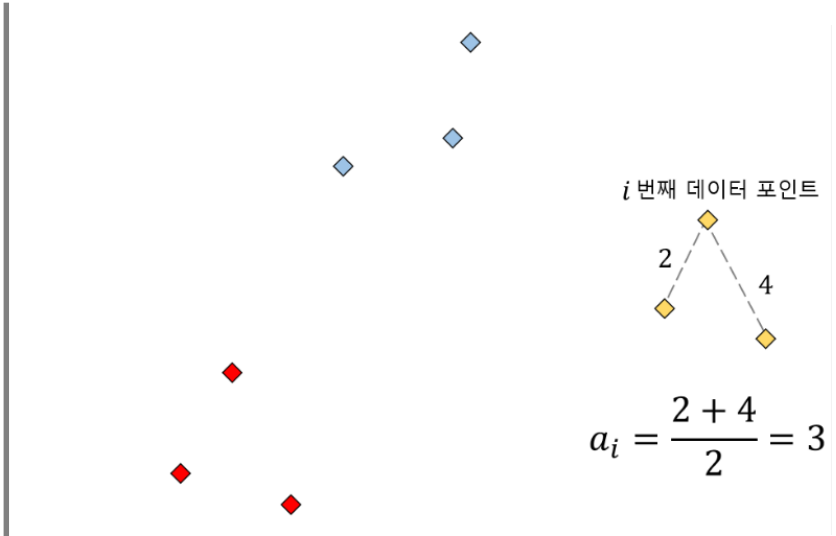
- 한 데이터 포인트 i에 대해 Silhouette Coefficient $s(i)$는 다음과 같이 계산됩니다:

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

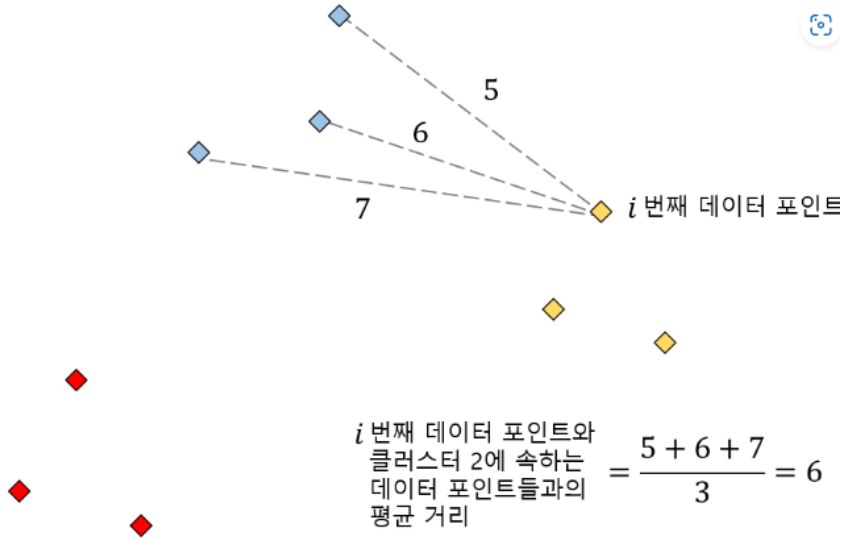
  - $a(i)$: 데이터 포인트 i가 속한 클러스터 내에서의 평균 거리 (Intra-cluster distance)
    - 동일 클러스터 내 다른 데이터 포인트들과의 평균 거리
  - $b(i)$: 데이터 포인트 i가 속하지 않은 다른 클러스터 중 가장 가까운 클러스터와의 평균 거리 (Inter-cluster distance)

- 예: 총 9개의 데이터에 대해서 3개의 군집, 클러스터 (군집) 1의 한 데이터에 대해서 실루엣 스코어 계산
- i번째 데이터가 속한 cluster내 다른 데이터와의 평균 거리 (a_i)=3





- i번째 데이터에서 가장 가까운 cluster와의 평균 거리 (b_i)=6
- i번째 데이터의 실루엣 스코어 계산 = 0.5
- 이것이 무슨 의미인가?



$$S_i = \frac{b_i - a_i}{\max\{a_i,\ b_i\}} = \frac{6-3}{\max(3,6)} = \frac{3}{6} = 0.5$$

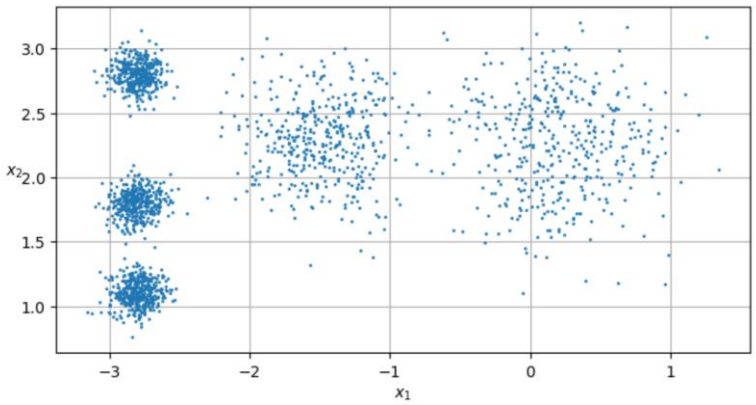
- **s(i)가 1에 가까울수록 해당 데이터는 잘 군집화된 것**
- **극단적으로 a(i)=0이면?**
- **s(i)가 음수이면 해당 데이터는 자신의 군집과 가까운 것 보다, 다른 군집에 더 가까운 경우 => 잘못된 군집에 배정**

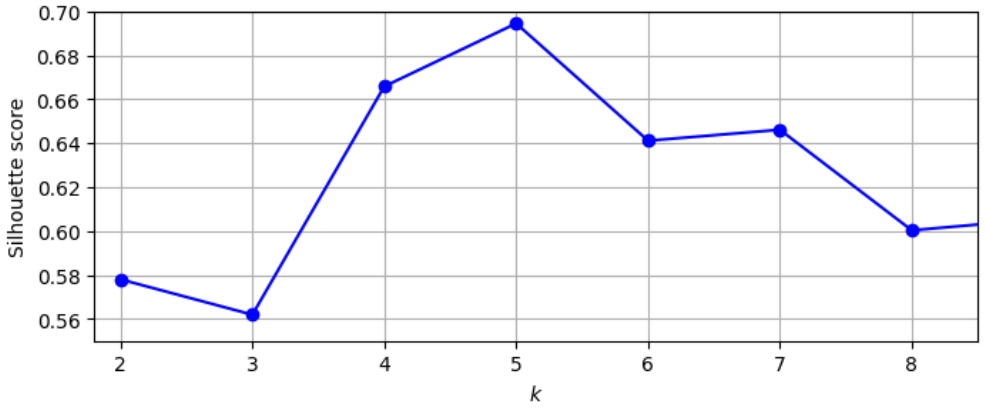
한 데이터 포인트 i에 대해 Silhouette Coefficient $s(i)$는 다음과 같이 계산됩니다:

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

- $a(i)$: 데이터 포인트 i가 속한 클러스터 내에서의 평균 거리 (Intra-cluster distance)
  - 동일 클러스터 내 다른 데이터 포인트들과의 평균 거리
- $b(i)$: 데이터 포인트 i가 속하지 않은 다른 클러스터 중 가장 가까운 클러스터와의 평균 거리 (Inter-cluster distance)

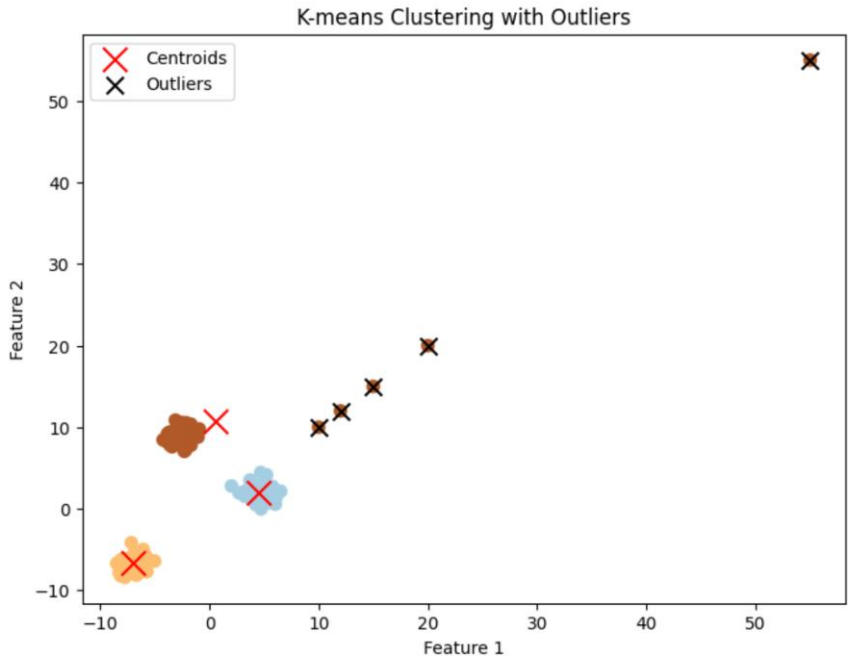
- k (군집 개수)의 변화에 따른 실루엣 스코어 변화
- k값이 5일때 가장 크다. 최적?
- k값이 4일때도 괜찮은 선택





- K-means 군집화의 약점
- 1. 초기 centroid 선택에 민감
- 2. 군집의 형태를 원형 형태라고 가정
- 3. 군집의 개수 K를 미리 지정해야 한다
- 4. outlier에 민감

- **(실습): Outlier가 몇 개 존재하면 centroid의 위치가 군집 바깥으로 이동할 수도 있다**




```
Cluster Centers (including outliers):
[[ 4.49951001  1.93892013]
 [-6.95170962 -6.67621669]
 [ 0.5460171  10.66194522]]
```

- 실습:
- 1. iris dataset을 이용한 군집화
- 2. 최적의 군집 수 K값 찾아봄
- 1) elbow
- 2) 실루엣 스코어