**지도학습과 비지도학습 비교 분석**

30501 김도현

## **1페이지**

서론 - 학습 방식의 근본적 차이, 지도학습 vs 비지도학습  
 이 노트북은 머신러닝의 두 가지 핵심 패러다임인 지도학습(Supervised Learning)과 비지도학습(Unsupervised Learning)을 대표적인 알고리즘인 K-최근접 이웃(KNN)과 K-평균(K-Means)을 통해 비교 분석한다. 지도학습은 '정답'이 주어진 데이터(labeled data)를 사용하여 모델을 학습시키고, 새로운 데이터의 정답을 '예측'하거나 '분류'하는 것을 목표로 한다. 반면, 비지도학습은 정답이 없는 데이터(unlabeled data)에서 숨겨진 구조, 패턴, 유사성을 발견하여 데이터를 의미 있는 그룹으로 '군집화'하는 것을 목표로 한다. 이 노트북은 두 알고리즘을 직접 구현하고 라이브러리로 활용하는 과정을 통해 이러한 근본적인 차이를 명확히 보여준다.

## **2페이지**

지도학습의 대표주자, KNN 알고리즘 구현  
 첫 번째 파트에서는 지도학습 기반의 분류 알고리즘인 KNN을 다룬다. 노트북은 scikit-learn 라이브러리를 사용하기에 앞서, '키'와 '몸무게'라는 특성을 가진 '기린'과 '하마' 데이터를 분류하는 간단한 예제를 통해 알고리즘의 핵심 로직을 직접 구현했다. 새로운 데이터가 주어졌을 때, 기존의 모든 데이터 포인트와의 유클리드 거리를 계산하고, 가장 가까운 K개의 이웃을 찾은 뒤, 그 이웃들 중 다수를 차지하는 클래스로 새로운 데이터를 분류하는 전 과정을 단계별로 보여주었다. 이는 KNN이 데이터 포인트 간의 '거리'를 기반으로 예측을 수행하는 본질을 명확히 이해하게 한다.

## **3페이지**

Scikit-learn을 통한 KNN의 효율적 활용  
 직접 구현을 통해 원리를 이해한 후, scikit-learn이라는 강력한 머신러닝 라이브러리를 사용하여 KNN을 얼마나 효율적으로 적용할 수 있는지를 보여준다. 붓꽃(Iris) 데이터셋을 로드하고, train\_test\_split으로 데이터를 분할한 뒤, KNeighborsClassifier 클래스의 객체를 생성했다. 단 몇 줄의 코드, 즉 knn.fit(X\_train, Y\_train)으로 모델 학습을 완료하고, knn.predict(X\_test)로 새로운 데이터에 대한 예측을 수행했다. 마지막으로 metrics.accuracy\_score를 통해 모델의 성능을 정량적으로 평가함으로써, 라이브러리가 제공하는 추상화의 이점을 극명하게 보여주었다.

## **4페이지**

비지도학습의 시작, K-Means 클러스터링 원리  
 노트북의 후반부는 정답 레이블이 없는 데이터를 그룹화하는 비지도학습 알고리즘, K-평균(K-Means)을 소개한다. K-Means의 목표는 전체 데이터들을 사용자가 지정한 K개의 군집(cluster)으로 나누는 것이다. 알고리즘은 다음 두 단계를 수렴할 때까지 반복한다: 1) 할당(Assignment) 단계: 각 데이터 포인트를 현재 군집의 중심점(centroid) 중 가장 가까운 중심점에 할당한다. 2) 업데이트(Update) 단계: 각 군집에 속한 데이터 포인트들의 평균값을 계산하여 새로운 중심점으로 업데이트한다. 노트북에서는 이 반복 과정을 while 루프를 사용하여 직접 구현함으로써, 알고리즘이 어떻게 군집을 형성해 나가는지 시각적으로 이해할 수 있도록 돕는다.

## **5페이지**

Scikit-learn을 활용한 K-Means 구현 및 시각화  
 KNN과 마찬가지로 K-Means 역시 scikit-learn의 KMeans 클래스를 통해 매우 간단하게 구현할 수 있다. KMeans(n\_clusters=2)와 같이 군집의 수(K)를 지정하여 객체를 생성하고, fit 메소드를 데이터에 적용하면 알고리즘이 자동으로 실행되어 최적의 군집을 찾는다. 학습이 완료된 후 kmeans.labels\_ 속성을 통해 각 데이터 포인트가 어떤 군집에 할당되었는지 확인할 수 있으며, kmeans.cluster\_centers\_ 속성을 통해 최종적으로 결정된 K개 군집의 중심점 좌표를 얻을 수 있다. 노트북은 matplotlib을 이용해 원본 데이터와 함께 이 중심점을 시각화하여, 알고리즘의 결과물을 직관적으로 검증했다.

## **6페이지**

K-Means의 다양한 구현 방식 - 재귀적 접근  
 알고리즘에 대한 깊이 있는 이해를 돕기 위해, while 반복문 방식 외에 재귀 함수(recursive function)를 이용한 K-Means 구현도 선보였다. 이 접근법은 현재 중심점을 기준으로 데이터를 두 그룹으로 나누고, 각 그룹의 평균을 구해 새로운 중심점을 계산한 뒤, 이 새로운 중심점을 인자로 하여 자기 자신을 다시 호출하는 구조를 가진다. 재귀는 이전 중심점과 새로운 중심점이 더 이상 변하지 않을 때(즉, 알고리즘이 수렴했을 때) 멈추게 된다. 이는 K-Means의 반복적인 최적화 과정을 다른 프로그래밍 패러다임으로 표현한 것으로, 문제 해결에 대한 다양한 관점을 제공한다.

## **7페이지**

결론 - 목적에 따른 알고리즘의 선택 (분류 vs 군집)  
 본 KNN,Kmeans.ipynb는 지도학습의 '분류(Classification)'와 비지도학습의 '군집화(Clustering)'가 어떻게 다른 목표를 가지며, 그에 따라 알고리즘이 어떻게 다르게 동작하는지를 명확하게 대비시켰다. KNN은 정답 레이블을 사용하여 모델을 학습시키고 새로운 데이터의 소속을 결정하는 반면, K-Means는 데이터 자체의 내재된 구조를 파악하여 유사한 데이터끼리 묶어준다. 결국 어떤 머신러닝 문제를 해결하고자 할 때, '정답 데이터가 있는가?'라는 질문에 따라 KNN과 같은 지도학습을 선택할지, K-Means와 같은 비지도학습을 선택할지가 결정된다.