기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 03. 지도학습 알아보기

정 정 민

programmers

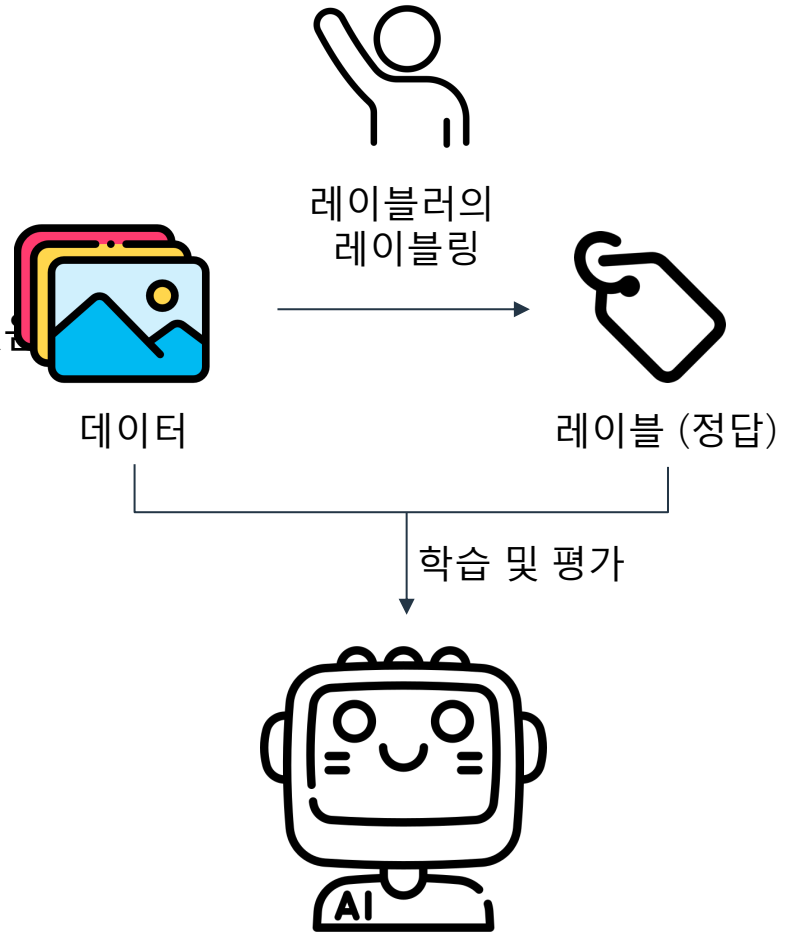
Chapter 08. 지도학습의 개념과 대표 알고리즘

1. 지도학습
2. 대표 알고리즘

지도학습

지도 학습, Supervised Learning

- 정답 레이블 정보를 활용해 알고리즘을 학습하는 학습 방법론
- 이 방법으로 학습되는 알고리즘은
  데이터와 정답인 레이블 사이의 관계를 파악하는 목적을 갖고 있음
- 특징 및 장점
  - 정답이 존재하므로 모델이 풀어야하는 문제가 비교적 쉽고 잘 학습 됨
  - 또한, 명확한 평가 수치가 존재하며 학습된 모델의 성능을 쉽게 측정할 수 있음
- 단점
  - 정답이 필요하므로 이를 위해 추가적인 시간, 노동, 비용이 필요
    - 정답을 매기는 행위에 필요한 전문 인력과 같은 추가 비용이 발생



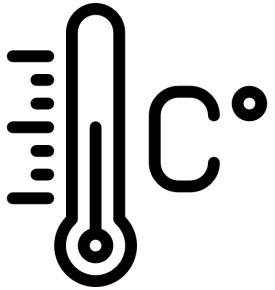
회귀 문제와 분류 문제

- 문제 (Task)란,
  - 쉽게 학창시절 우리가 풀었던 여러 문제와 비슷
  - 머신 러닝 기법을 활용해 해결하고자 하는 대상으로
  - 데이터를 통해 얻고자 하는 특정 목표나 결과를 의미

- 지도 학습에서 흔히 다루는 대표적인 두 문제
  - 회귀 문제 (Regression problem)
    - 주어진 입력 데이터에 대해 연속적인 숫자값을 예측하는 문제
  - 분류 문제 (Classification problem)
    - 주어진 입력 데이터가 어떤 범주(클래스, class)에 속하는지를 판별하는 문제

회귀 문제, Regression Problem

- 주관식 문제와 비슷
- 입력 데이터를 바탕으로 정확한 숫자 형태의 결과를 예측하는 문제
  - 정확한 숫자는 정수 혹은 실수 범위의 수
- 예를 들어,
  - 내일 주식 가격은? 53,228.3 원
  - 5년 뒤 나의 몸무게는? 73.2 kg



39.3 ° C

내일 서울의 온도는??

분류 문제, Classification Problem

- 5지선다형 객관식 문제와 비슷
- 입력으로 주어지는 데이터를 정해진 보기 중 하나로 분류하는 문제
  - 보기 : 클래스(Class)
- 분류 문제의 세분화 : 주어진 클래스의 수 & 모델이 결과로 출력하는 수 등에 따라 나뉨
  - 이진 분류 문제 : 주어지는 클래스가 2 개인 경우
  - 다중 클래스 문제 : 모델이 여러 클래스를 내보내야 하는 경우
    - 예를 들어, 고양이 → 동물, 포유류와 같이 여러 종류로 분류

선풍기
고양이 ✓
개구리
의자

긍정 ✓
부정

"웨이드 같은 남자 어디 없나요 ㅠㅠ"

분류 vs 회귀

- 분류 문제와 비교해보면
  - 내일 온도를 예측한다면 : 회귀
  - 내일 날씨를 예측한다면 : 분류 (맑음, 비, 흐림, 눈 중 택 1)

  - 내일 주식의 가격을 예측한다면 : 회귀
  - 내일 주식의 등락을 예측한다면 : 분류 (오른다, 내린다)

  - 사진에 나온 사람의 나이를 예측한다면 : 회귀
  - 사진에 나온 사람의 나이대를 예측한다면 : 분류 (10대, 20대, 등등..)

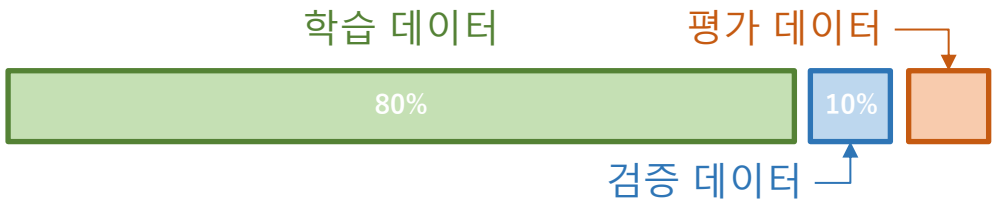
- 즉,
  - 분류: 보기 중 선택의 문제
  - 회귀 : 정확한 숫자 값을 찾는 문제

데이터 분할 : 학습 / 검증 / 평가

- 시험을 보는 학생의 공부 방법을 잠깐 생각해볼까요?
  - **이론지** : 학습을 통해 정보를 습득하고 이해하는 과정에서 사용
  - **모의고사** : 습득한 정보를 연습하고 중간 중간 학습 상태를 확인
  - **시험** : 모의고사를 통해 최적의 공부 상태를 만들고 시험을 진행

- 모의고사는 시험과는 다름
  - 모의고사에서 좋은 점수를 받았다고 꼭 시험에서 좋은 점수를 받는 것은 아님
  - 단 본인의 공부 정도를 판단하는 척도로 활용

- 머신 러닝 모델도 비슷하게 이론지, 모의고사, 시험을 활용
- 이것 각각들의 이름은
  - 이론지 : 학습 데이터 (train data)
  - 모의고사 : 검증 데이터 (validation data)
  - 시험 : 평가 데이터 (test data)



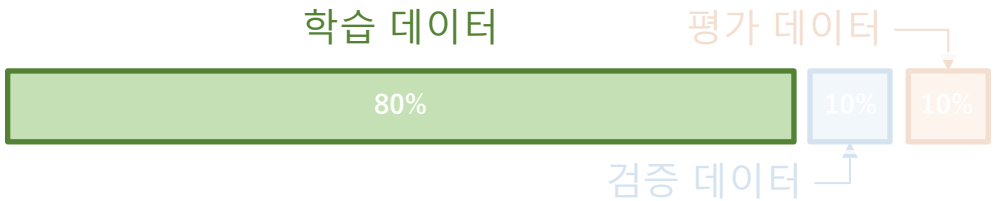

```
# 전체 데이터를 기준으로
All_data = Load_All_dataset()

# 학습 데이터 준비 (80%)
train_dataset = split_from(All_data, 0.8)
# 검증 데이터 준비 (10%)
val_dataset = split_from(All_data, 0.1)
# 평가 데이터 준비 (10%)
test_dataset = split_from(All_data, 0.1)
```

이론지 : 학습 데이터 (train data)

- 순수하게 학습을 하는 과정에서 사용하는 데이터

- 갖고 있는 전체 데이터 중 가장 많은 비율을 차지함
  - 정해진 정답은 없지만 약 80% 정도를 사용
  - 이 분류에 속한 데이터가 많으면 많을수록 성능이 좋아질 가능성이 커짐



모의고사 : 검증 데이터 (Validation data)

- 학습을 진행하는 중간 과정에서
  머신 러닝 모델이 어느 정도 학습 되었는지를 주기적으로 확인하는데 사용하는 데이터

- 검증 과정은 학습 중간에 진행되는 평가라고 생각할 수 있음
- 모의고사를 통해 전체 시험 범위 중 부족한 단원을 찾는 것과 같이
- 학습의 정도를 판단할 수 있음

- 전체 데이터 중 약 10% 정도를 할당

시험 : 평가 데이터 (Test data)

- 학습의 과정과는 별도의 과정
- 최후의 머신 러닝 모델이 생성된 후, 학습한 모델의 최종 성능을 평가하기 위해 사용되는 데이터

- 시험 공부 기간 중 중간/기말고사 시험지를 미리 볼 수 없듯
- 평가 데이터는 학습 과정에서는 절대 사용되지 않음

- 완성된 머신 러닝 모델이 서비스 혹은 제품과 같이
  실제 사용 시나리오 과정에서 보게 될 데이터라는 가정으로 만들어진 데이터

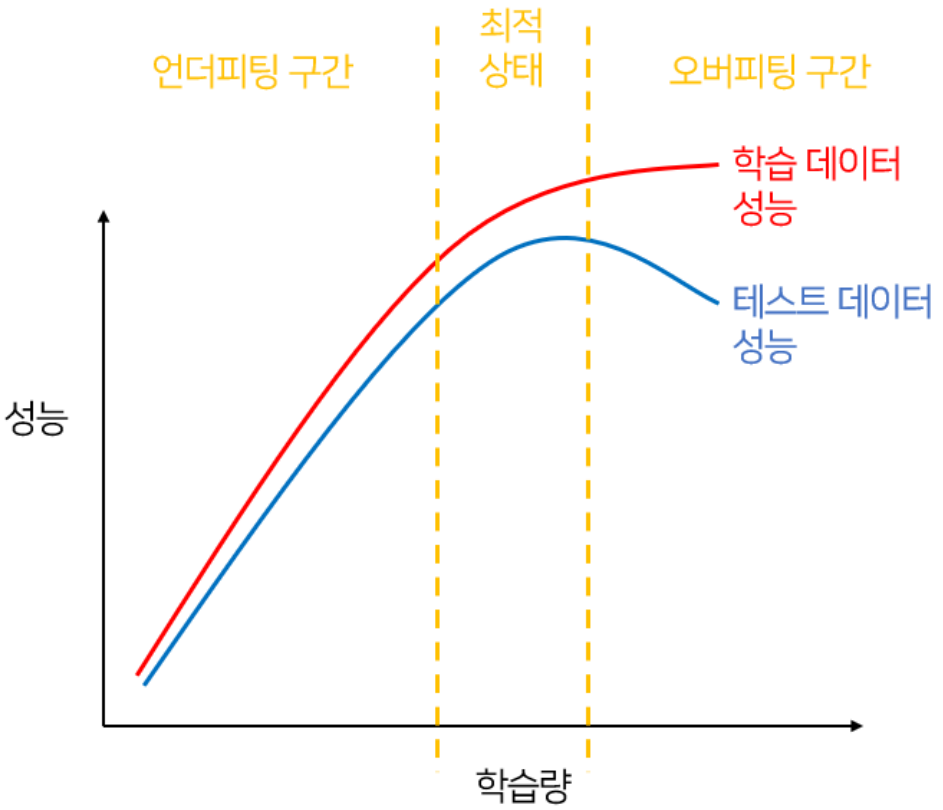
- 전체 데이터 중 약 10% 정도를 할당

과적합, Overfitting

- 머신 러닝 모델이 특정 훈련 데이터에 지나치게 학습되어
- 새로운 데이터나 테스트 데이터에서 잘 작동하지 않는 상태를 의미

- 이런 상태를 일반화 능력(generalization)이 떨어진 상태라고 표현
- 학습 데이터에 포함된 특정 패턴이나 디테일 그리고 작은 노이즈까지 학
  - 단순히 데이터를 외워버린 경우!

- 과적합은 머신 러닝 모델이 경계해야하는 현상이며
- 이를 위해 아래와 같은 방법들이 존재
  - 데이터 양 늘리기
  - 머신 러닝 모델의 복잡도를 줄이기
  - 규제와 같은 정규화 기법 사용하기
  - 등등



https://blog.naver.com/mmismin/223104767970

손실 함수 (Loss Function)

- 머신 러닝 모델이 얼마나 잘 하고 있는지 또는 못하고 있는지를 수치화한 손실(Loss)을 구하는 함수
- 다시 말해, 모델의 예측값과 실제 정답 사이의 차이를 측정하는 지표
- 일반적으로, 손실(Loss)이 작을수록 모델의 성능이 좋다고 볼 수 있음

- 따라서 머신 러닝 모델을 학습하는 과정은 손실(Loss)를 줄이는 과정으로 진행 됨
- 손실 함수의 종류는 해결하고자 하는 문제의 유형(회귀, 분류 등)에 따라 다름
  - 회귀 문제 : 평균 제곱 오차 (Mean Squared Error)
  - 분류 문제 : 교차 엔트로피 (Cross Entropy)
    - 이진 분류 문제 : 로그 손실 (Log Loss)

파라미터(Parameter)와 최적화(Optimization)

- 머신 러닝 모델의 파라미터란 해당 모델이 내부적으로 갖고 있는 변수를 의미
- 이 변수는 모델이 데이터로부터 학습하는 패턴 관계를 표현하며 모델의 예측 성능에 직접적인 영향을 미침

- 파라미터의 구조와 조합은 모델마다 다양하며 이 변수의 값은 학습의 과정으로 찾아야 함
  - 즉, 성능이 좋은 모델은 적절한 구조의 파라미터로 구성되며
  - 파라미터의 구체적인 값은 데이터를 이용한 학습으로 찾게 됨

- 최적화란 머신 러닝에서 모델의 성능을 최대화하거나, 오류를 최소화하기 위해
- 모델의 파라미터를 조절하는 과정을 의미
- 즉, Loss 값이 최소가 되는 파라미터를 찾는 것을 목표로 함

- 최적화 적용 과정은 머신 러닝 모델에 따라 상이할 수 있음
  - 최적의 해를 한번에 구하는 경우
  - 점진적이고 반복적으로 해를 구하는 경우

대표 알고리즘

분류 문제

- **로지스틱 회귀 (Logistic Regression)**
  - 이진 분류 문제에 적합한 구조
  - 확률을 직접 예측하는 확률 추정 접근으로 결과를 예측

- **결정 트리 분류기 (Decision Tree Classifier)**
  - 데이터를 잘 분할하는 결정 트리를 사용하여 분류를 수행
  - 직관적이고 이해가 쉬움

- **랜덤 포레스트 (Random Forest)**
  - 여러 결정 트리의 결합으로 앙상블 기법에 해당
  - 높은 정확도를 보이면서도 과적합 문제를 방지함

- **서포트 벡터 머신 (Support vector Machine, SVM)**
  - 데이터를 최적으로 분리하는 결정 경계를 찾는데 강력한 알고리즘
  - 어려운 형태의 데이터라도 비선형 계산이 가능한 다양한 커널 트릭있어 해를 구할 수 있음

회귀 문제

- **선형 회귀 (Linear Regression)**
  - 기본적이고 널리 사용되는 회귀 알고리즘
  - 독립 변수와 종속 변수 간의 선형 관계를 모델링
- **라쏘 회귀 혹은 릿지 회귀 (Lasso & Ridge Regression)**
  - 규제 기법을 이용해 과적합을 방지하고 일반화 성능이 향상된 선형 모델
- **결정 트리 회귀 (Decision Tree Regression)**
  - 결정 트리를 이용해 회귀 문제에 적용
- **서포트 벡터 회귀 (Support Vector Regression, SVR)**
  - 분류 모델인 SVM을 회귀에 적용한 알고리즘
- **K-최근접 이웃 회귀 (K-Nearest Neighbors Regression)**
  - 주어진 데이터 포인트에서 가장 가까운 K개의 이웃 데이터의 평균으로 예측값을 결정
  - 간단하면서도 데이터 자체만을 활용한 추정(비모수적 추적)이 가능

E.O.D