# 과제 1030

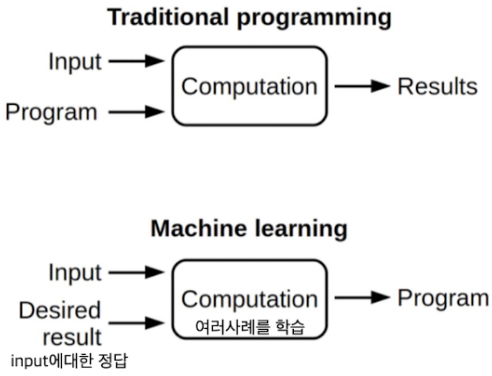
2315028 김성현

▼ 기계학습 기초

- 기계학습

  : 인간이 새로운 지식과 경험을 학습하는 것처럼 경험을 통해 컴퓨터를 지능적으로 만들고자 하는 것

  학습은 경험을 전제로해야됨.

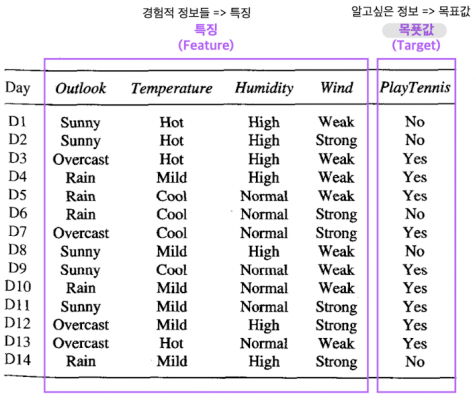



용어배경 ) 아서사무엘, 컴퓨터에서 명시적 프로그래밍 없이도 학습을 통해 특정과업을 수행할 수 있는 기술이라고 처음 정의함

톰미첼, 경험을 통해 나중에 유사하거나 같은 일을 더 효율적처리할 수 있도록 시스템의 구조나 파라미터를 변경하는 것이라고 정의함

⇒ 기계학습은 경험자료인 데이터로부터 모르는 것을 추론하기위한 알고리즘 설계 분야

▼ 데이터 구조

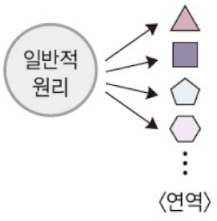


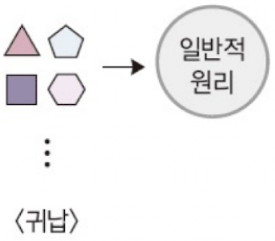

| Day | *Outlook* | *Temperature* | *Humidity* | *Wind* | *PlayTennis* |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

경험적 정보가되는 ⇒ 특징(feature)

최종적으로 알고싶은 정보 ⇒ 목표값 (target)

▼ 추론




연역적 추론

: 이론, 원리, 규칙으로부터 사례나 현상을 이해하는 추론방식

← 전문가 시스템, 지식기반시스템에서 많이 활용됨(if-then)

귀납적 추론

: 여러 사례로부터 일반적인 원리, 패턴, 규칙을 이끌어내는 추론 방식

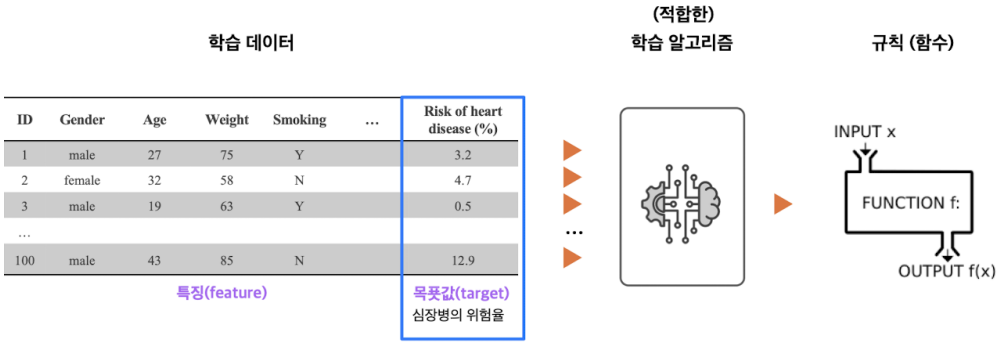
← 기계학습분야에서 많이 활용됨

---

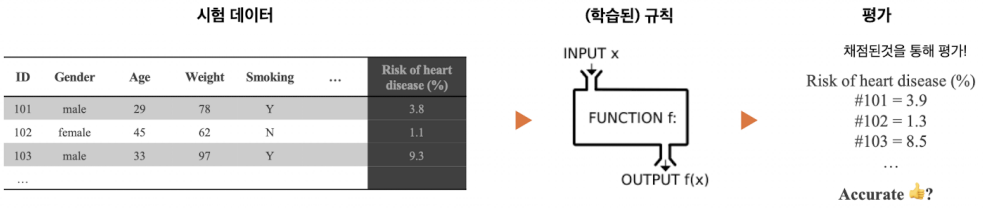
▼ 기계학습의 기본단계

학습데이터모으기 > 학습데이터 정제하기 > 모델 학습하기 > 평가 > 예측

- 모델학습




- 모델평가



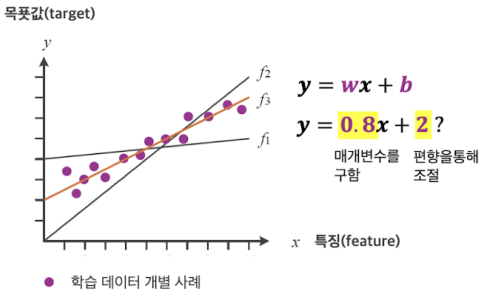

▼ 학습

: 학습데이터를 잘 대표할 수 있는 적합한 함수를 찾는 것 ( 함수의 **매개변수**를 찾아가는 과정 )

가중치 : 입력 값이 출력에 미치는 중요도를 조절하는 매개변수

편향 : 절편이라 부르며, 입력과 무관하게 기본적 출력을 조정하는 역할

매개변수 : 기울기(가중치), 절편(편향)




▼ 모델

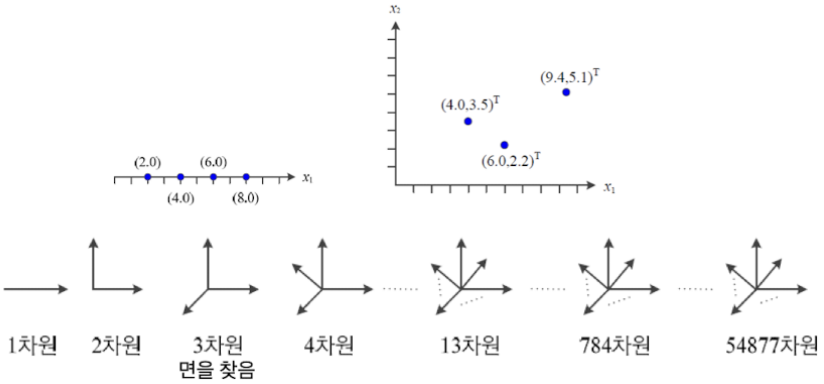
▼ 특징, 차원

특징의 개수 d에 따라 d차원의 특징 공간이 선형결합된 형태로 표현

( 고차원으로 갈수록 단순직선이 아닌 초평면으로 존재함.

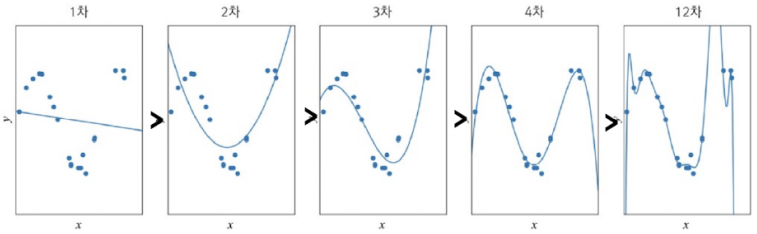
선을 찾기보단 경계를 찾는 것이 정화한 표현이 됨 )

$$y = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_dx_d + b$$




차원의 저주 : 데이터 수 대비 차원이 너무 많으면 전체 공간 내 **데이터가 희소**해지고, 계산이 어려워짐 ⇒ 불필요한 차원을 제거, 규제하거나 차원축소 기법등을 도입해 차원을 적절히 줄여주어야함

▼ 다항모델




▼ 과적합(과대적합, overfitting) : 모델이 훈련에 사용한 특정데이터에 너무 최적화되어 그외 새로운 데이터에서 성능이 저하되는 상태

- 사례

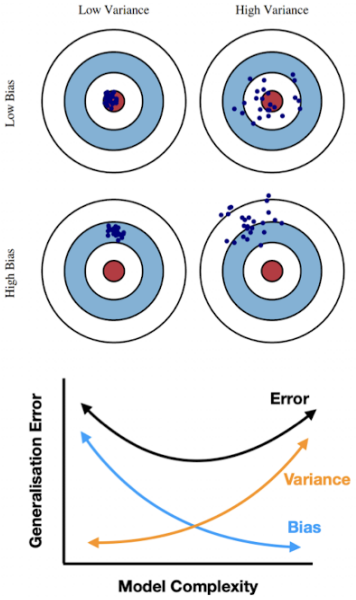
  모델이 너무 복잡한 경우 / 학습을 너무 오랫동안 한 경우 / 데이터가 부족한 경우

▼ 과소적합(underfitting) : 모델이 데이터패턴을 충분히 학습하지 못하여 예측성능이 낮은상태

- 사례

  모델이 너무 단순한 경우 ex_1차선형함수 / 특징이 데이터를 충분히 표현하지 못한 경우 / 데이터가 부족하거나 너무 적은 시간만 학습해 학습이 충분치 않은 경우

---




- 편향(bias) : 모델의 예측이 기댓값과 벗어나는 경향

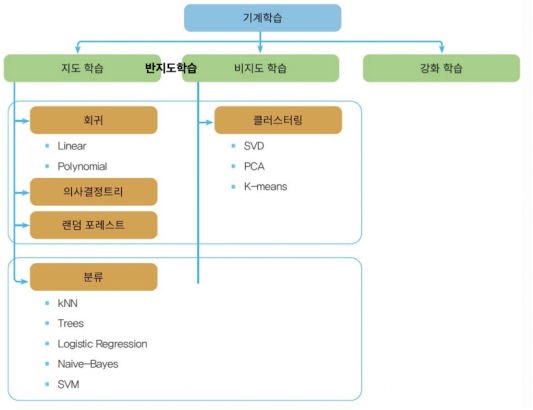
  → 일반적으로 모델이 단순한경우에 강한 편향이 나타남

- 분산(variance) : 훈련 데이터 안의 변동 때문에 모델의 예측이 달라지는 정도

  → 모델이 복잡한 경우, 큰 분산이 나타남

⇒ 기계학습에서는 낮은편향과 낮은분산이 목표이지만 둘은 trade-off관계이므로 편향 희생을 최소로하고 분산을 최대로 낮추는 전략이 필요!

▼ 학습방법




- 지도학습(supervised learning)

  **입력(문제) - 출력(답)** 쌍의 데이터로부터 새로운 입력에 대한 출력을 결정할 수 있는 패턴 추출

- 비지도학습(unsupervised learning)

  출력에대한 **정보가 없는 데이터**로부터 필요한 패턴 추출

- 강화학습(reinforcement learning)

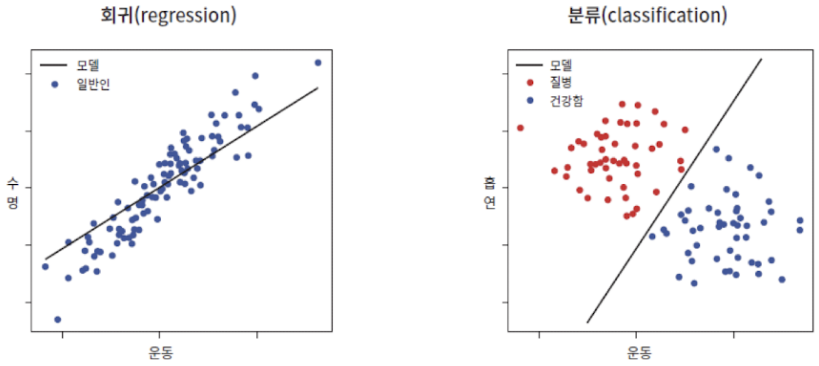
출력에대한 정확한 정보를 제공하지는 않지만, **평가정보(reward)**가 주어지는 문제에대해 각 상태의 행동을 결정

▼ 지도학습

: 입력-출력 쌍으로 이루어진 학습데이터에서 y=f(x)일때 해당 f를 **근사(approximation)하는 함수 h를** 구하는 것

**h** - 가설(hypothesis), 모형(model), 함수(function)라고 불림

여러 가설 중 데이터에 최적합하는 함수를 찾아야함. 가설이 테스트 결과를 정확히 예측하면 가설이 잘 **일반화(generalizatino)**된것임




- 회귀(regression) : 출력이 연속적임. 연속형

  → 학습 데이터에 부합되는 출력값이 실수인 함수를 찾는 문제

  회귀문제의 학습, 회귀모델 ) 출력이 실수인 학습데이터가 주어질때, 입력에서 출력으로의 매핑함수를 학습

- 분류(classification) : 출력이 유한한 개수의 값. 범주형

  → 데이터들을 몇개의 범주(class)로 대응시키는 문제. ex_이진분류, 다중분류

  분류문제의 학습, 분류모델 ) 학습데이터를 잘 분류할 수 있는 함수를 찾는 것

  함수의 형태는 수학적함수 혹은 규칙임

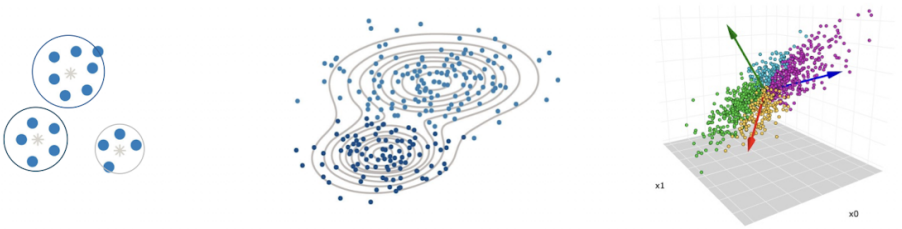
  classifier 분류기 ) 학습된 함술르 이용해 데이터를 분류하는 프로그램

▼ 비지도학습

: 목푯값, 레이블이 없는 데이터에서 특정 패턴을 찾는 것.

데이터에 잠재한 구조, 계층구조, 숨겨진 사용자 집단을 찾는 것

문서들을 주제에따라 구조화하는 것, 로그정보를 사용해 사용패턴을 찾아내는 것



군집화 / 밀도추정 / 차원축소

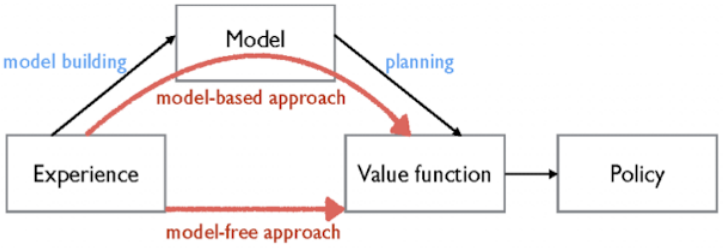
- 군집화(clustering) : 유사성에 따라 비슷한 특징을 가진 데이터를 분할하는 기법
- 밀도추정(density estimation) : 범주별 데이터를 만들었을 것으로 추정되는 확률분포를 찾는 기법
- 차원축소(dimensionality reduction) : 고차원의 데이터의 손실을 최소화하며 저차원으로 변환하는 기법
- 이상치 탐지(anomaly detection) : 데이터 내에서 예상치 못한 패턴을 찾는 기법

▼ 강화학습

: 데이터가 아닌 에이전트가 환경과 상호작용을 통해 직접 경험을 축적하며 시행착오(trial and error)를 통해 학습하는 방법

에이전트가 환경과 상호작용을 통해 보상(reward)이 최대가 되도록 주어진 상태(state)에서 취할 수 있는 적합한 행동(action)을 찾는 것

---




- 용어

보상 - 행동에 대한 결과의 평가치로 주어짐

정책 - 강화학습 에이전트가 행동을 결정할때 사용하는 규칙

가치함수 - 각 상태에서 특정행동을 선택함에따라 전이된 상태가 얼마나 좋은지 보상의 기댓값을 나타냄. → 에이전트가 더 나은 정책을 학습할 수 있도록 도움

모형화 - 모델기반 강화학습(: 모형에 기반한 상태전이 및 보상을 예측해 최적의 계획을 세움) / 모델프리 강화학습(: 모형없이 직접경험만을 통해 학습함)

- 행동 선택 전략

이용 - 지금까지 학습한 정보로 가장 높은 보상을 제공할 것을 예상한 행동 선택

→ 이미 알고있는 정보를 최대한 활용해 보상을 극대화

탐험 - 충분히 학습되지 않은 새로운 행동을 선택

→ 에이전트는 아직 발견하지 못한 최적의 정책을 찾을 수 있음

---

▼ numpy, matplotlib

## Numpy

: 수학 및 과학 분야의 수치 연산을 위한 파이썬 패키지
- 특히, 벡터, 행렬 등 계산할 때 빠른 고성능 계산이 가능하여 대량의 데이터를 처리하는데 유리함

▼ ndarray : 다차원 array형태를 ndarray객체를 제공함

- 필요성 )

행렬 및 벡터연산을 위해 다차원 array를 사용해야함

- 속성 )

  - ndarray.ndim: 배열의 차원 수
  - ndarray.shape: 각 차원의 크기를 나타내는 튜플
  - ndarray.size: 배열에 포함된 전체 요소 개수
  - ndarray.dtype: 배열에 저장된 요소의 데이터 타입

- 형 변환 )

  -ndarray.astype(자료형) : 배열을 특정 자료형으로 변환

  - int8, int16, int32, int64
  - float16, float32, float64, float128
  - complex64, ...
  - bool

---

- 넘파이 배열 생성 )

  - np.arange(): 원하는 숫자 범위 내 특정 간격에 따른 배열 생성
  - np.ones(): 1로 가득찬 배열 생성
  - np.zeros(): 0으로 가득찬 배열 생성
  - np.full(): 특정 값으로 가득찬 배열 생성
  - np.linspace(): 원하는 숫자 범위 내 원하는 개수의 요소를 가진 배열 생성

- 랜덤값 배열 생성 )

  - np.random.rand(): 0과 1 사이의 무작위 값이 들어간 배열 생성 (균등분포, uniform dist.)

- np.random.randn(): -1과 1 사이의 무작위 값이 들어간 배열 생성 (정규분포, normal dist.)
- np.random.randint(): 특정 범위 내 무작위 정수값 들어간 배열 생성

---

- 넘파이 배열구조의 재배열 )

  - np.reshape(변경할 배열, 차원)
  - ndarray.reshape(차원)

> 넘파이 배열간의 연산은 반복문 없이도, 내부적으로 벡터 내 성분 간 연산처리 가능함(벡터화 계산)

- 벡터의 내적구하기(dot product)

내적은 벡터의 같은 성분끼리 각각 곱해 합한 값, 스칼라 값으로 반환

- np.dot(벡터1, 벡터2)
- 벡터1 @ 벡터2

---

- 넘파이 배열 응용연산(통계량, 고급연산)

  - np.sum(): 배열 요소 전체 합산
  - np.mean(): 배열 요소 전체 평균
  - np.median(): 배열 요소 중앙값
  - np.var(): 배열 요소 분산
  - np.std(): 배열 요소 표준편차

- 그외 수치계산

  - np.exp(): 자연상수 e의 지수함수
  - np.log(): 자연상수 e의 로그함수
  - np.sqrt(): 제곱근

## Matplotlib