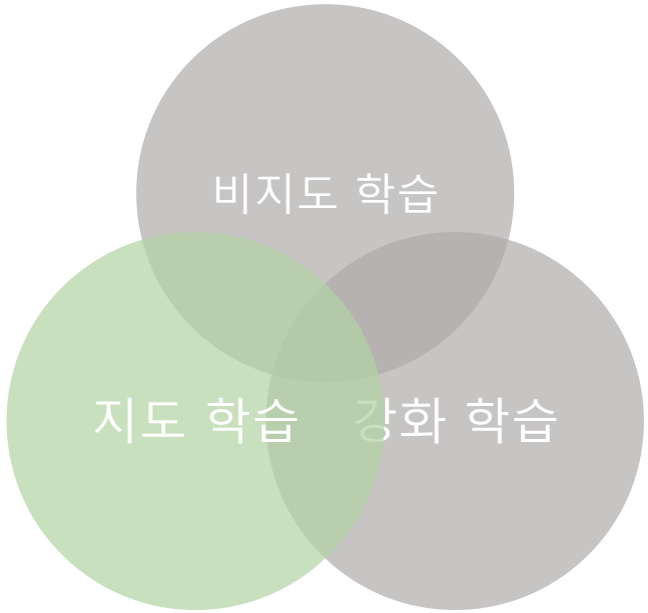
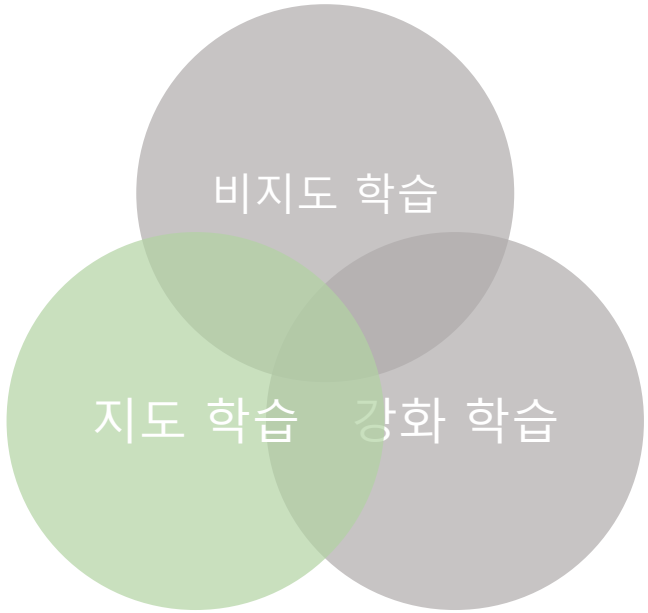


지도학습(supervised learning)

(= 입력과 타깃으로 모델을 훈련)

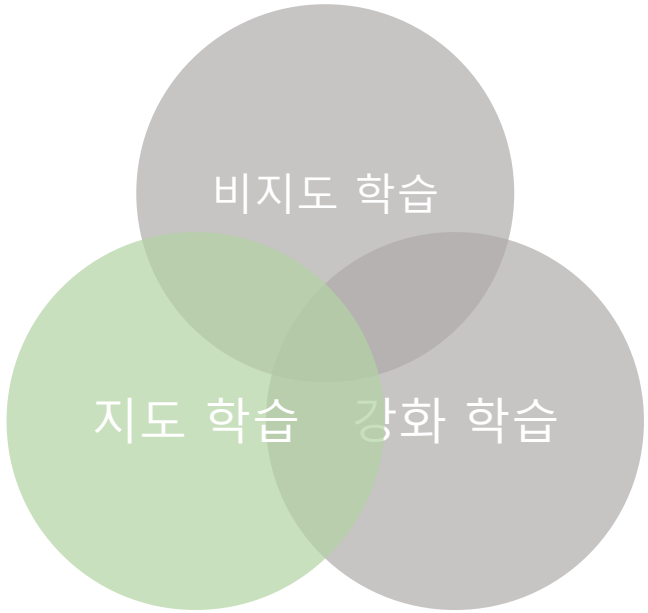
- 지도 학습은 인간이 프로그램을 지도해주는 것이다.
- 컴퓨터가 자율적으로 판단하는 것이 아니라 사람이 제시한 여러가지 기준을 토대로 확률 상 컴퓨터가 학습하는 기능이다.



지도학습 동작 방식

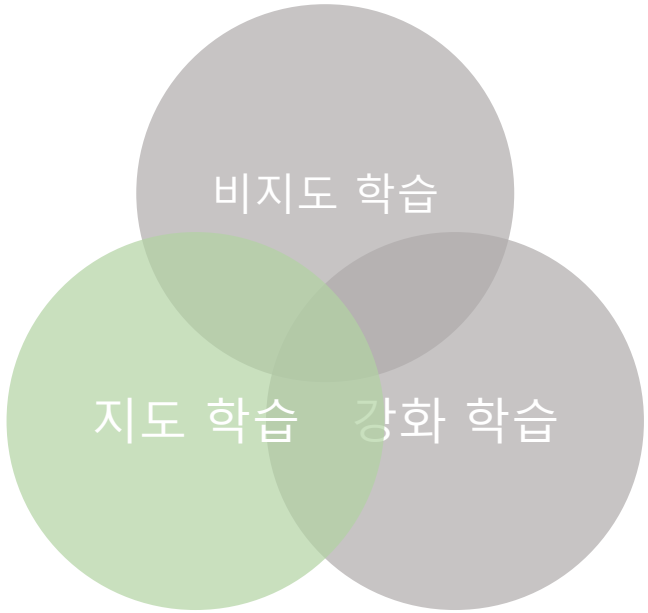
(= 입력과 타깃으로 모델을 훈련)

- 지도 학습 알고리즘은 레이블이 지정된 입력을 가져와 알려진 출력에 매핑합니다.
- 즉 대상 변수를 이미 알고 있다는 의미 입니다.
- 지도 학습 방법은 머신러닝 모델을 훈련하기 위해 외부 감독이 필요합니다.
- 따라서 이름이 지도가 되었습니다.
- 원하는 결과를 반환하려면 지침과 추가 정보가 필요합니다.



지도학습에서 많이 사용되는 알고리즘

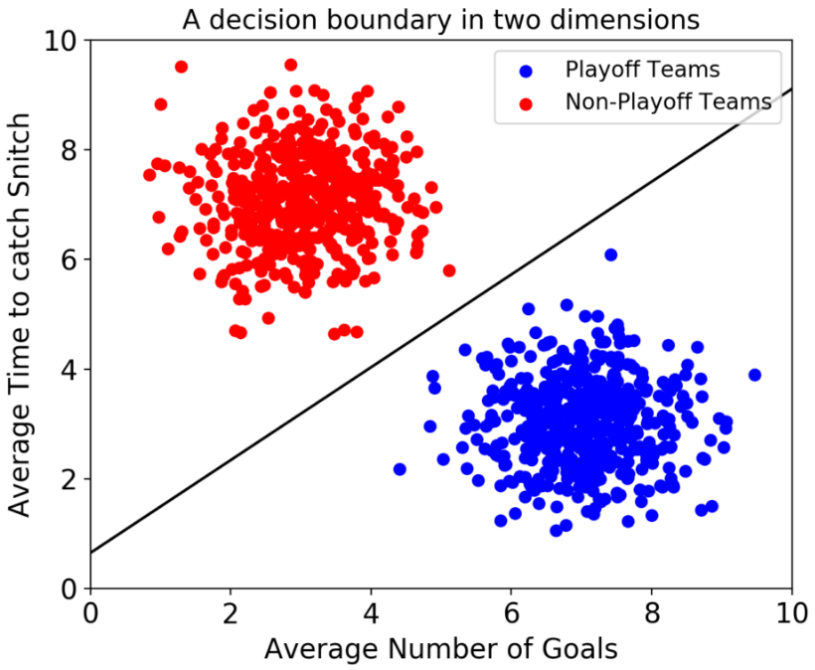
- Decision tree 의사결정 나무
- Logistic regression 로지스틱 회귀
- Support vector machine 서포트 백터 머신



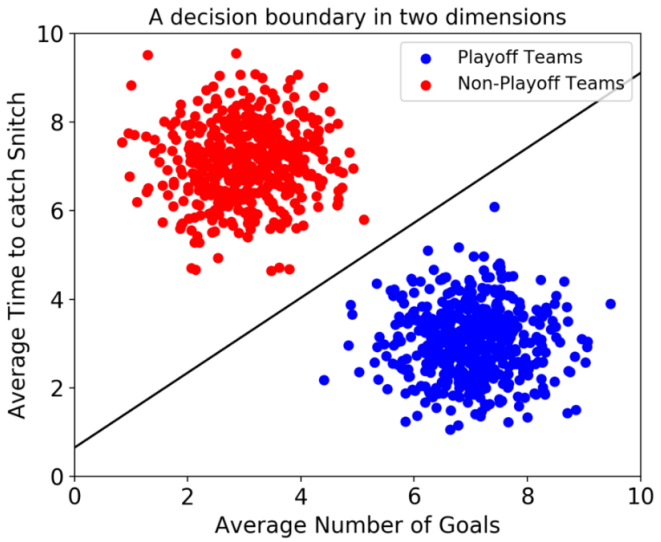
지도학습에서 많이 사용되는 알고리즘

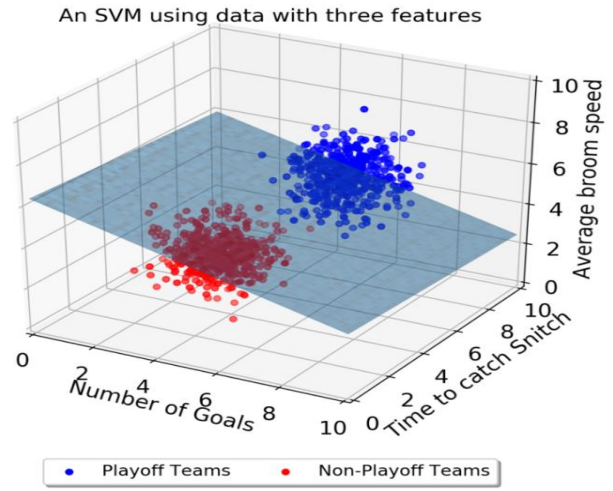
- Linear Regression 선형 회귀
- K Nearnest Neighbor K 가장 가까운 이웃
- Random Forest 랜덤 포레스트
- Naïve Bayes 나이브 베이즈

지도학습 알고리즘 : Support vector machine



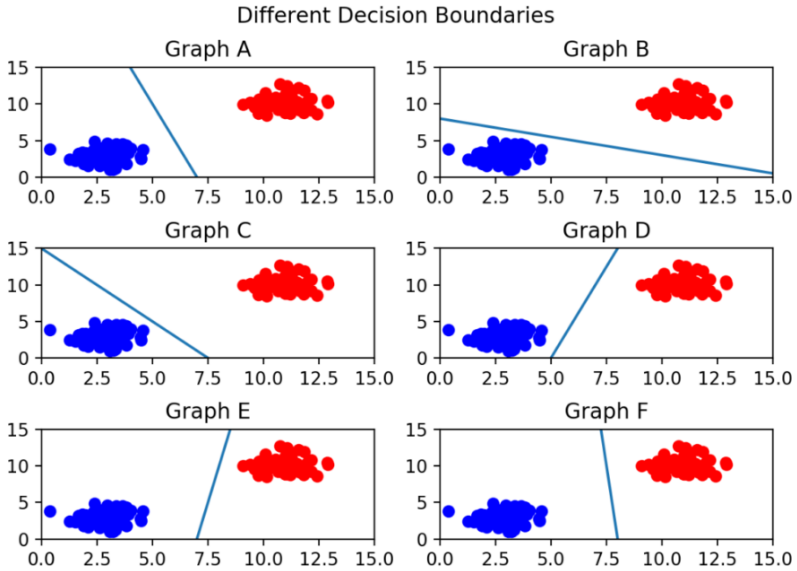
- 서포트 벡터 머신은 결정 경계, 즉 분류를 위한 기준 선을 정의하는 모델입니다.
- 그래서 분류되지 않은 새로운 점이 나타나면 어느 쪽에 속하는지 확인해서 분류 과제를 수행할 수 있게 됩니다.
- 결국 이 결정 경계라는 걸 어떻게 정의하고 계산하는지 이해하는 게 중요합니다.





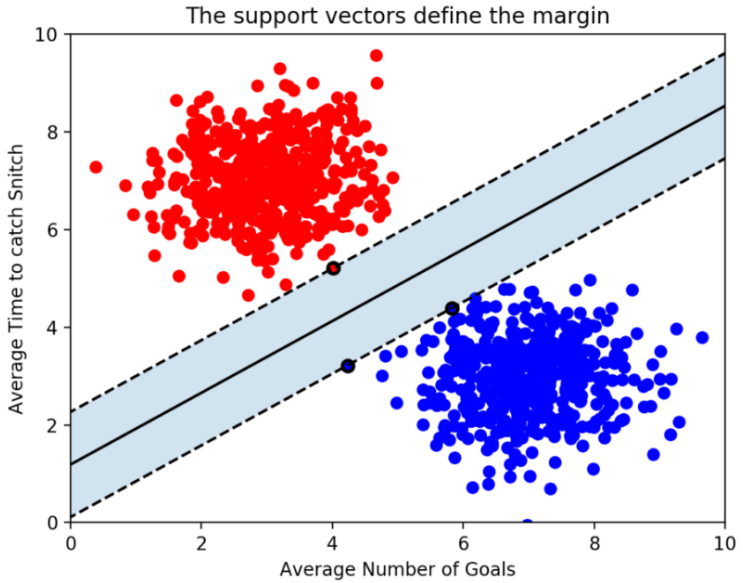
지도학습 알고리즘 : Support vector machine

- 만약 데이터에 2개의 속성만 있다면 결정경계는 간단한 선 형태가 됩니다.
- 그러나 속성이 3개로 늘어난다면 이렇게 3차원으로 그려야 합니다.
- 이때 결정경계는 선이 아닌 평면이 됩니다.
- 우리가 이렇게 시각적으로 인지할 수 있는 범위는 딱 3차원까지 입니다.
- 즉, 속성의 개수가 늘어날 수록 당연히 복잡해질 것입니다. 결정 경계도 단순한 평면이 아닌 고차원이 될 텐데 이를 초평면 이라고 부릅니다.



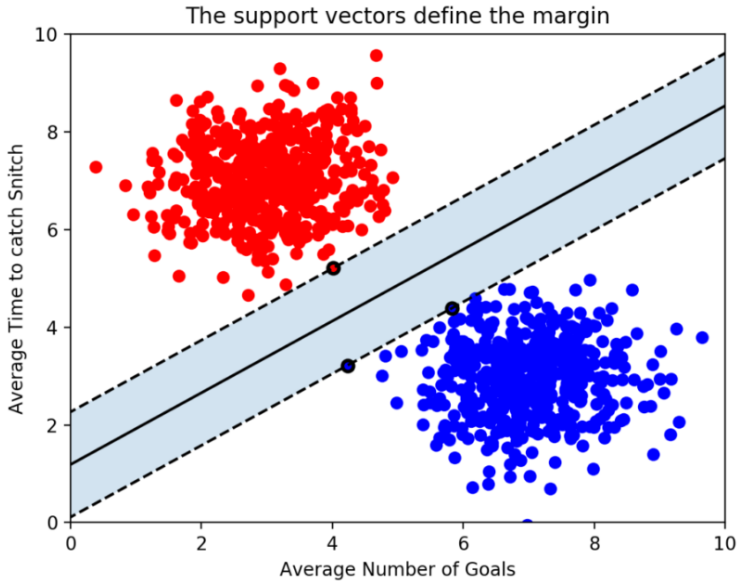
지도학습 알고리즘 : Support vector machine 최적 결정 경계

- C를 보면 선이 파란색 부류와 너무 가까워서 아슬아슬해 보입니다.
- F를 보면 두 클래스 즉 분류 사이에서 거리가 가장 멀기 때문에 적절합니다.
- 이제 결정 경계는 데이터 군으로부터 최대한 멀리 떨어지는 게 좋습니다.
- 실제로 서포트 벡터 머신이라는 이름에서 서포트 벡터는 결정 경계와 가까이 있는 데이터 포인트들을 의미 합니다.
- 이 데이터들이 경계를 정의하는 결정적인 역할을 하는 셈입니다.



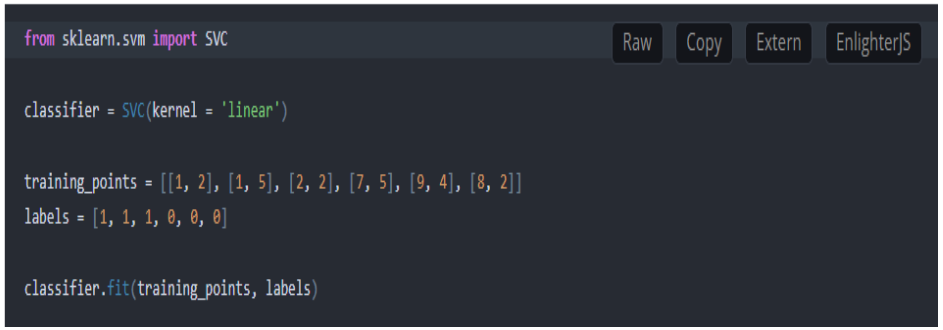
지도학습 알고리즘 : Support vector machine 마진(Margin)

- 마진은 결정 경계와 서포트 벡터 사이의 거리를 의미합니다.
- 가운데 실선이 하나 그어져 있는데 이게 바로 결정 경계가 됩니다.
- 그리고 그 실선으로부터 검은 테두리가 있는 빨간 점 1개, 파란 점 2개까지 영역을 두고 점선을 그어 놓았습니다.
- 점선으로부터 결정 경계까지의 거리가 바로 마진입니다.
- 최적의 결정 경계는 마진을 최대화 합니다.
- N개의 속성을 가진 데이터에는 최소 n+1개의 서포트 벡터가 존재합니다.
- SVM에서는 결정 경계를 정의하는 게 결국 서포트 벡터이기 때문에 데이터 포인트 중에서 서포트 벡터만 잘 골라내면 나머지 쓸 데 없는 수 많은 데이터 포인트들을 무시할 수 있어서 매우 빠르다.



지도학습 알고리즘 : Support vector machine scikit-learn

- SVM에서 결정 경계를 구하는 것은 상당히 복잡한 최적화 문제입니다.
- 파이썬 scikit-learn 라이브러리로 SVM을 구현해볼 수 있습니다.
- scikit-learn을 활용하면 모델을 구현할 때 사용되는 추가적인 개념도 익힐 수 있습니다.



지도학습 알고리즘 : Support vector machine 알고리즘 코드

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear')
training_points = [[1, 2], [1, 5], [2, 2], [7, 5], [9, 4], [8, 2]]
labels = [1, 1, 1, 0, 0, 0]
classifier.fit(training_points, labels)
print(classifier.predict([[3, 2]]))
```

지도학습 알고리즘 : Decision tree 의사결정 나무 장점

- 데이터 전처리를 거의 하지 않아도 된다.
- 이해라고 해석하기 쉽다.
- 분류와 회귀가 가능해 여러 용도로 사용할 수 있다.

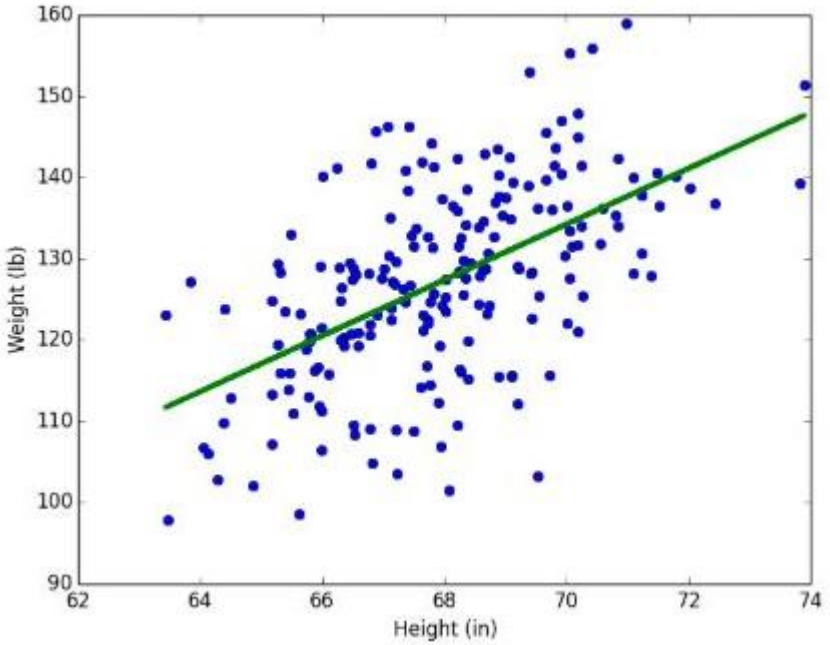
지도학습 알고리즘 : Decision tree 의사결정 나무 단점

- 과적합이 발생하기 쉽다.
- 데이터 변화에 민감하다.

지도학습 알고리즘 : Decision tree 알고리즘 구현

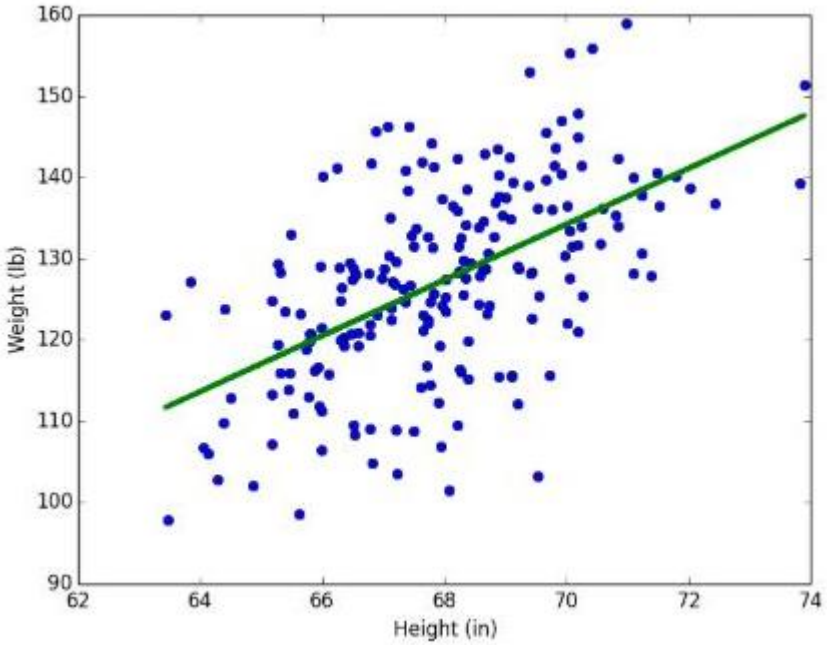
- https://ratsgo.github.io/machine%20learning/2017/07/08/treecode/

지도학습 알고리즘 : Linear Regression 선형 회귀



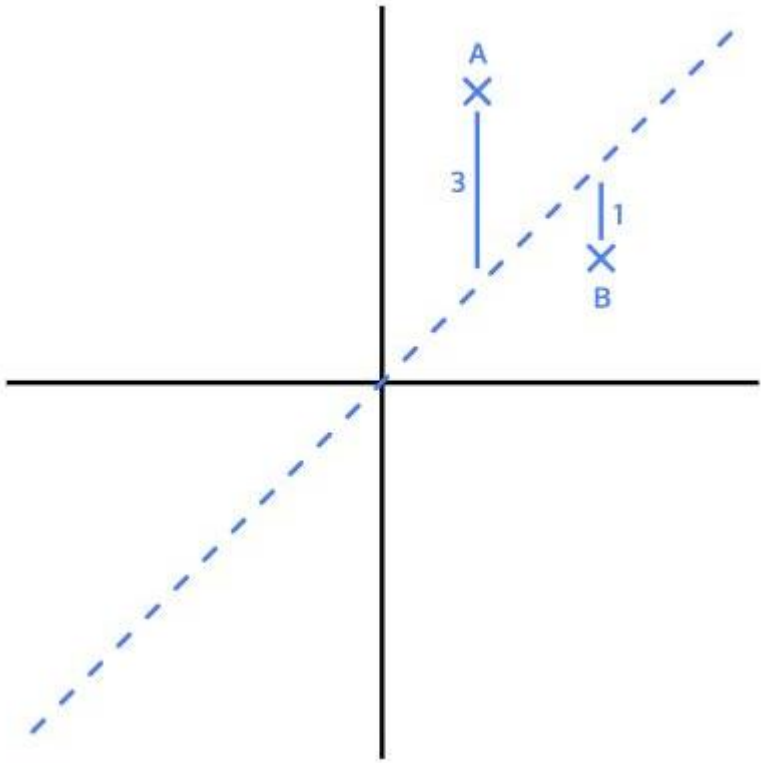
- 머신 러닝의 가장 큰 목적은 실제 데이터를 바탕으로 모델을 생성해서 만약 다른 입력 값을 넣었을 때 발생할 아웃풋을 예측하는 데에 있습니다.
- 이때 우리가 찾아낼 수 있는 가장 직관적이고 간단한 모델은 선이다.
- 그래서 데이터를 놓고 그걸 가장 잘 설명할 수 있는 선을 찾는 분석 방법을 선형 회귀 분석이라 부릅니다.

지도학습 알고리즘 : Linear Regression 선형 회귀



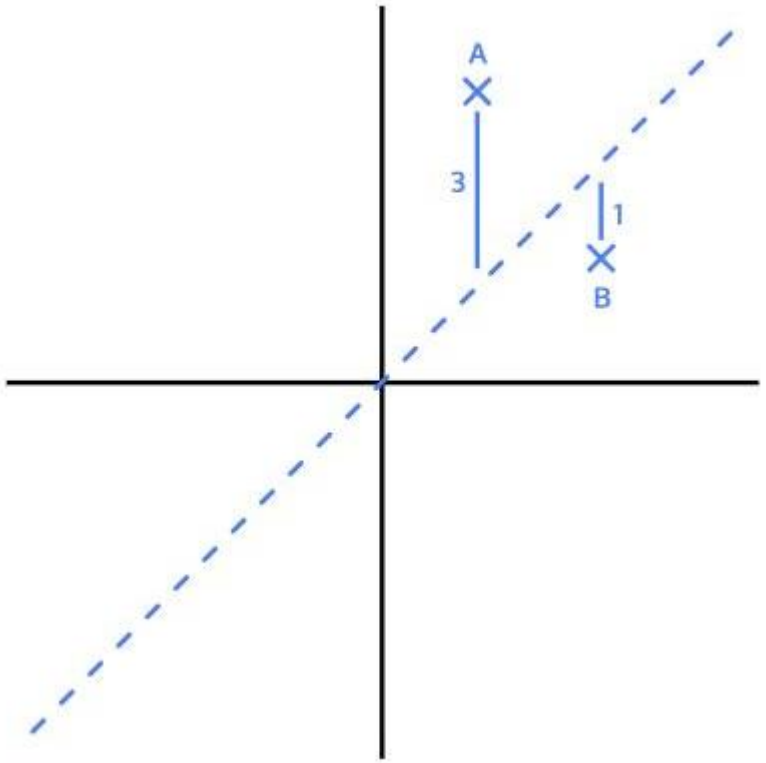
- 예를 들어 키와 몸무게 데이터를 펼쳐 놓고 그것들을 가장 잘 설명할 수 있는 선을 하나 그어놓게 되면 특정 인의 키를 바탕으로 몸무게를 예측할 수 있습니다.
- 당연히 근사치고 정확하지 않지만 최대한 가깝게 추정할 수 있다는 데에 의의가 있습니다.
- Y = mx + b
- 기울기 m, 절편 b에 따라 그 선의 모양이 정해지기 때문에 x를 넣었을 때 y를 구할 수 있습니다.
- 선형 회귀 분석의 목적도 결국 우리가 가진 데이터를 가장 잘 설명할 수 있는 m과 b를 얻는 것입니다.

지도학습 알고리즘 : 선형 회귀에서 발생하는 오차



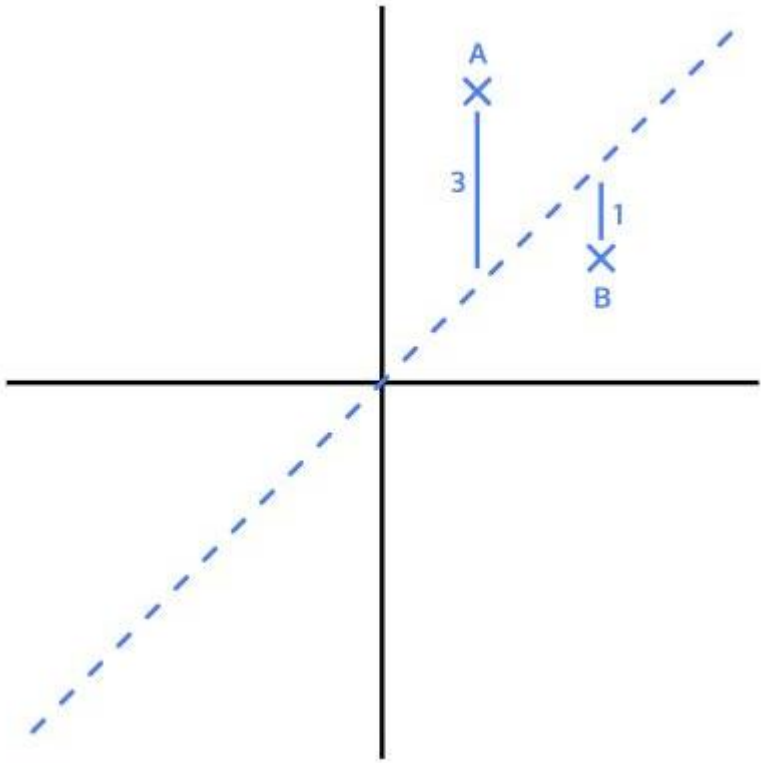
- 데이터들을 놓고 선을 긋는다는 것은 결국 대충 어림잡아본다는 뜻인데 그러면 당연히 선은 실제 데이터와 약간의 차이가 발생합니다.
- 그림을 보면 A는 3, B는 1만큼의 손실이 발생했습니다.
- 그런데 엄밀히 보면 + 또는 – 방향을 고려하지 않고 얘기한 것입니다.
- 선과 실제 데이터 사이에 얼마나 오차가 있는지 구하려면 양수와 음수 관계없이 동일하게 반영되도록 모든 손실에 제곱을 해주는 게 좋습니다.
- 이런 방식으로 손실을 구하는 것을 평균 제곱 오차(MSE)라고 부릅니다.

지도학습 알고리즘 : 선형 회귀에서 손실을 구하는법

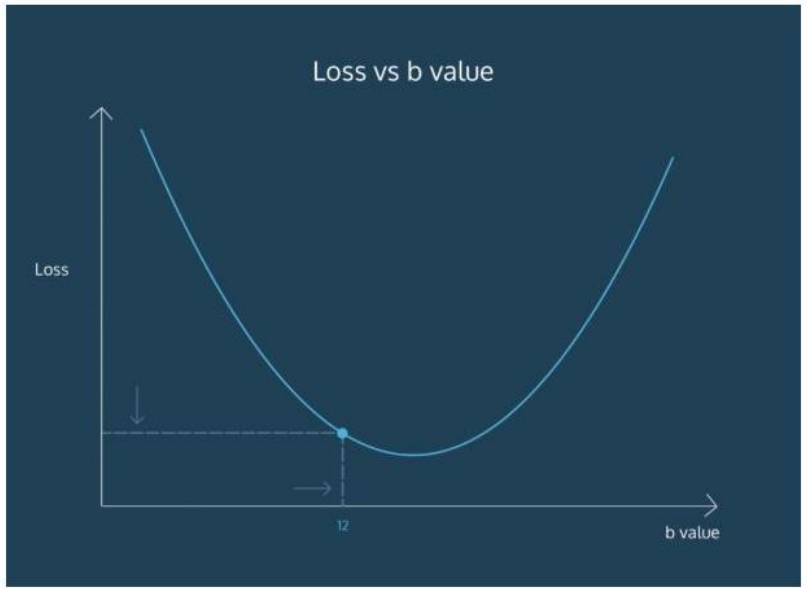


- 평균 제곱 오차(Mean Squared Error)
- 평균 절대 오차(Mean Absolute Error)
- 결정 계수(Coefficient of Determination)

지도학습 알고리즘 : 선형 회귀 모델의 목표



- 선형 회귀 모델의 목표는 모든 데이터로부터 나타나는 오차의 평균을 최소화할 수 있는 최적의 기울기와 절편을 찾는 것 입니다.



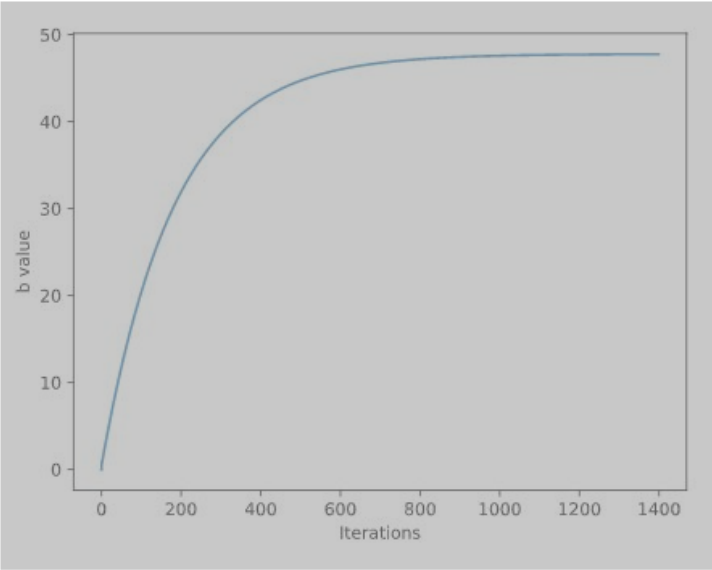
$$\frac{2}{N}\sum_{i=1}^{N} -(y_i - (mx_i + b))$$

$$\frac{2}{N}\sum_{i=1}^{N} -x_i(y_i - (mx_i + b))$$

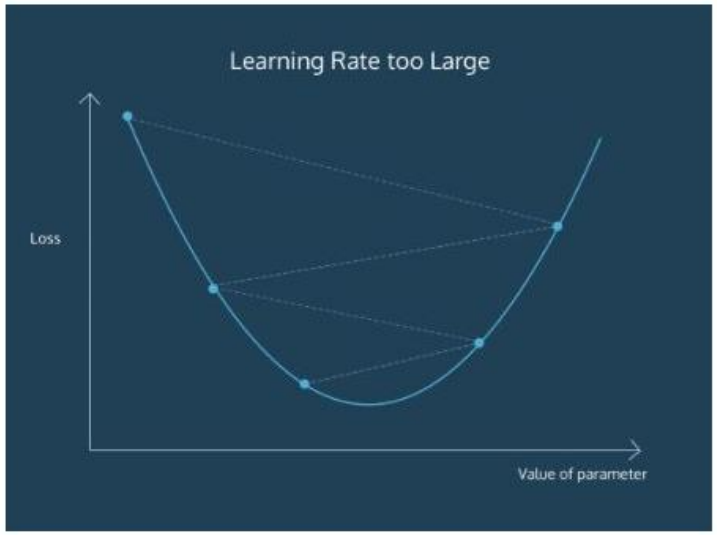
지도학습 알고리즘 : 선형 회귀 손실 최소화 경사 하강법

- 머신러닝에서 사용하는 모형은 매우 복잡하기 때문에 선형 회귀 분석에서도 최적의 기울기와 절편을 구할 수 있는 마땅한 방법이 없습니다.
- 그나마 단서가 있다면 위에서 설명한 손실을 함수로 나타내면 이렇게 아래로 볼록한 모양이라는 것입니다.
- 파라미터의 임의로 정한 다음에 조금씩 변화시켜가며 손실을 점점 줄여가는 방법으로 최적을 파라미터를 찾아갑니다.
- 예를 들어 절편을 구할 때 사용하는 공식은 옆과 같은데 이 값이 최소가 되도록 계속 b값을 변화시키는 것입니다.
- 기울기를 구할 때 사용하는 공식은 옆과 같은데 마찬가지로 이 값이 최소가 되도록 m값을 변화시키는 것입니다.

지도학습 알고리즘 : 선형 회귀 모델 수렴(Convergence)



- 선형 회귀 분석을 수행하면 기울기와 절편을 계속 변경해가면서 최적을 값을 찾게 될 텐데 이것을 언제까지 할지 정해줘야 합니다.
- 파라미터를 계속 조정을 하다 보면 어느정도 최적의 값으로 수렴합니다.
- 옆 그림을 보면 1000번 반복하니까 b값이 결국 47에 수렴하는 것을 알 수 있습니다.

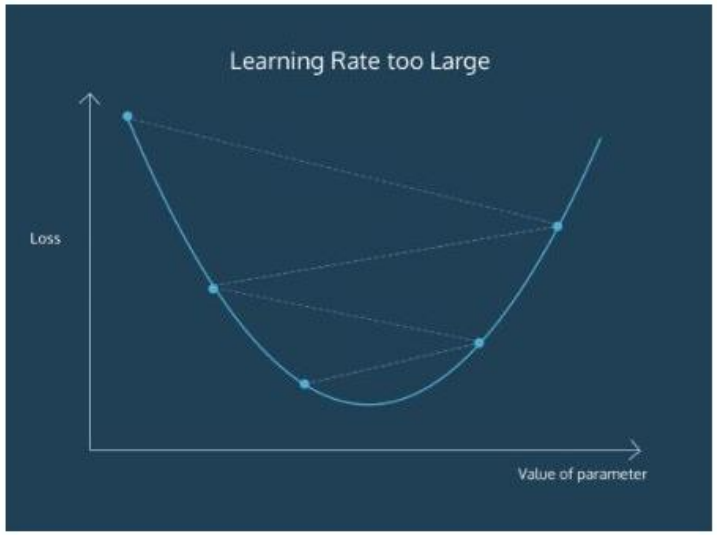


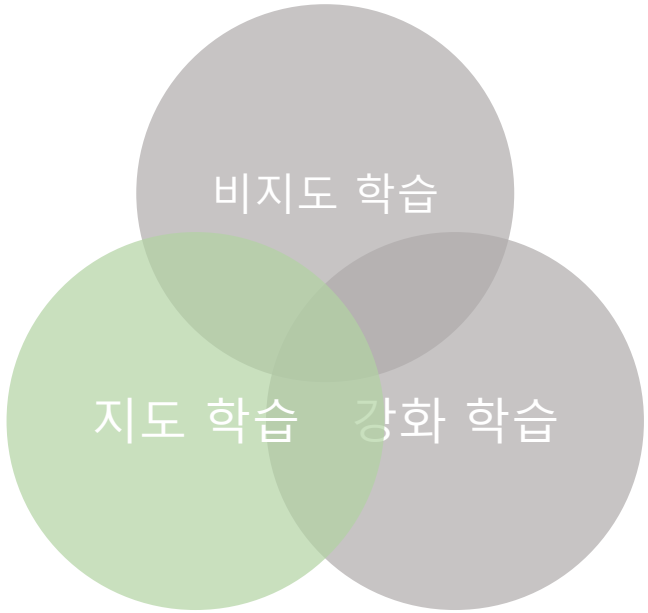
지도학습 알고리즘 : 선형 회귀 모델 학습률(Learning Late)

- 학습률이라는 것을 정해줄 필요가 있습니다.
- 학습률을 크게 설정하면 최적의 값을 제대로 찾지 못합니다.
- 일을 빨리 하긴 하겠지만 대충하는 것입니다.
- 그렇다고 학습률을 작게 설정하면 최적의 값으로 수렴할 때까지 시간이 오래 걸립니다.
- 그래서 모델을 학습시킬 때는 최적의 학습률을 찾는 게 중요합니다.
- 효율적으로 파라미터를 조정하면서도 결국 최적의 값을 찾아 수렴할 수 있을 정도로 해야합니다.

지도학습 알고리즘 : 선형 회귀 모델 알고리즘 구현

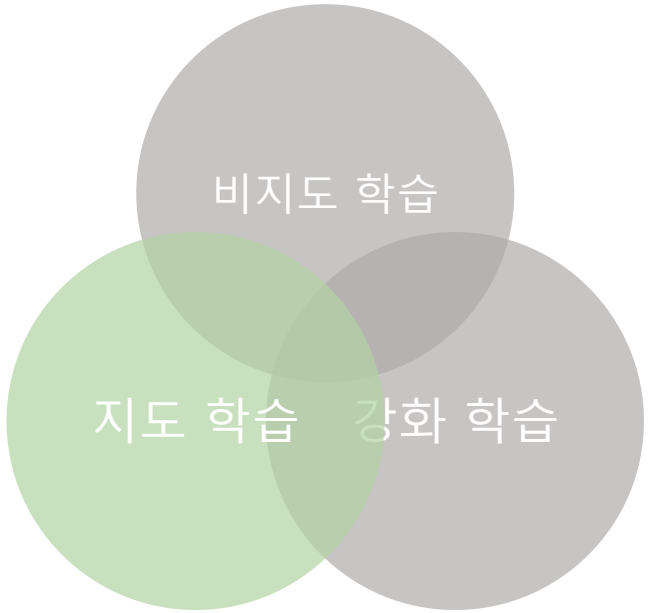
- https://hleecaster.com/ml-linear-regression-example/





지도학습의 한계

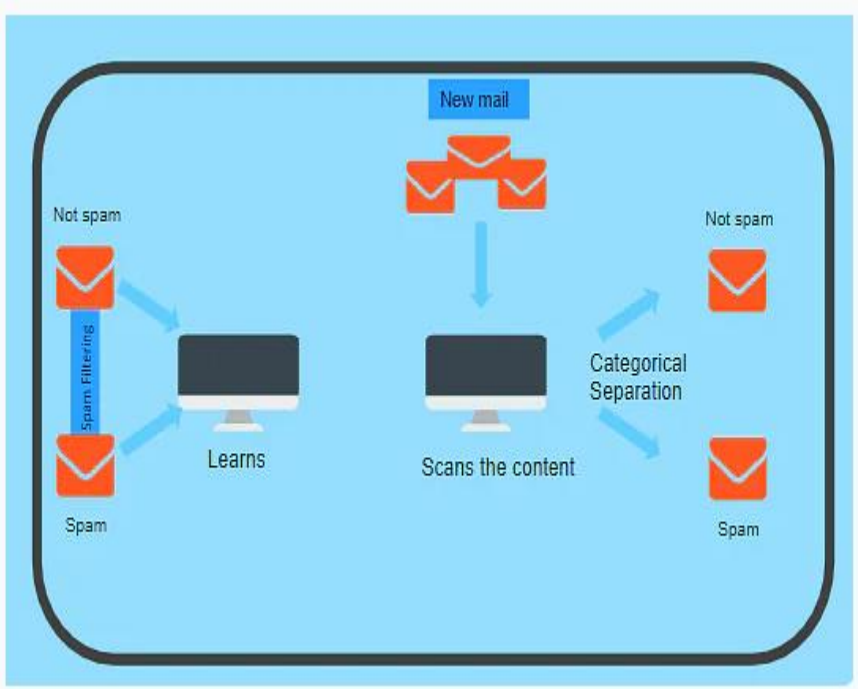
- 지도학습의 핵심은 훈련데이터다.
- 정교한 프로그램이 되려면 거대한 훈련 데이터가 필요한데 이런 데이터를 수집하거나 만드는 과정은 만만치 않으며 데이터 중에 오차가 큰 데이터가 있을 때도 문제가 된다.
- 입력들이 새로 들어왔을 때 실시간으로 모델이 최신으로 되지 않는다는 것이다.
- 새로 입력된 데이터를 포함하여 새로 훈련을 시켜야 한다.
- 모델이 한 번 만들어지면 그 모델에 대한 예상 값만 반환할 뿐 환경 변화에 적응하기 어렵다.



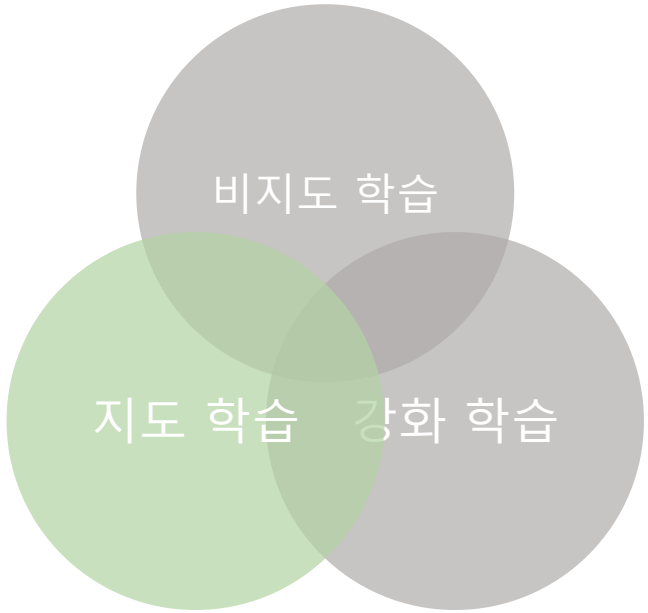
지도학습의 두가지 형태 : 분류

- 분류는 출력 변수가 예를 들어, 예 또는 아니오, 남성 또는 여성, 참 또는 거짓 등의 범주형(2개이상의 클래스 포함) 데이터 일 때 사용됩니다.

지도학습의 두가지 형태 : 분류 예시

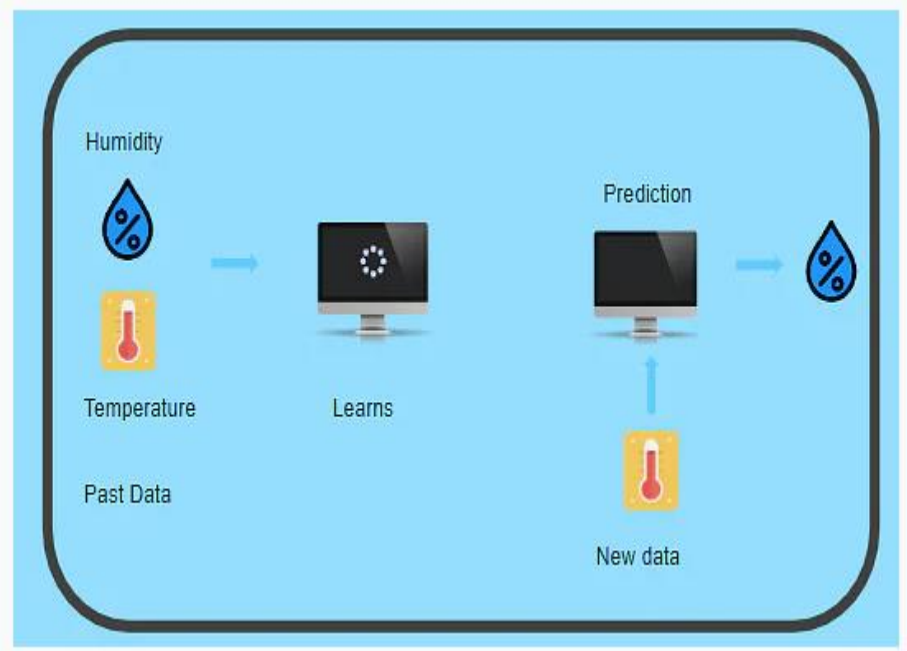


- 메일이 스팸인지 여부를 예측하려면 먼저 기계에 스팸 메일이 무엇인지 가르쳐야 됩니다.
- 이것은 많은 스팸 필터를 기반으로 수행됩니다.
- 메일 내용 검토, 메일 헤더, 검토 및 잘못된 정보가 포함되어 있는지 검색합니다.
- 이미 블랙리스트에 올라간 스패머로부터 협박을 받는 블랙리스트 필터와 특정 키워드 등
- 이러한 모든 특성들을 메일에 점수를 매기고 스팸 점수를 부여하는 데 사용됩니다.
- 이 메일의 총 스팸점수가 낮을수록 스팸이 아닐 가능성이 높습니다.
- 새 수신 메일의 콘텐츠, 레이블 및 스팸 점수를 기반으로 알고리즘은 해당 메일을 받은 편지함 또는 스팸 폴더에 넣을지 여부를 결정합니다.



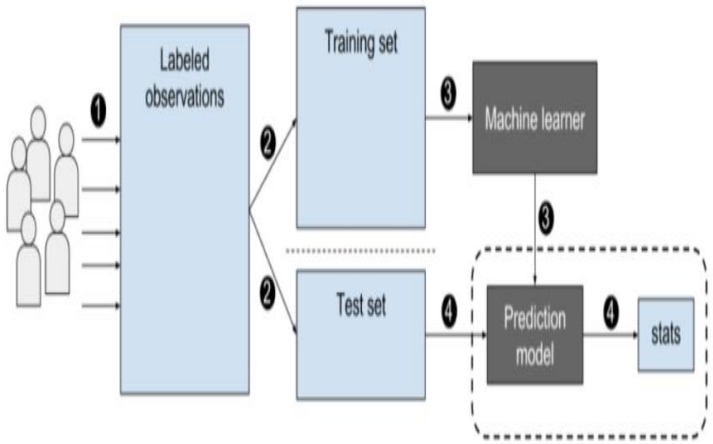
지도학습의 두가지 형태: 회귀

- 회귀는 출력 변수가 실수 또는 연속 값일 때 사용됩니다.
- 이 경우 둘 이상의 변수 사이에 관계를 가집니다.
- 즉, 예를 들어, 경력에 따른 급여 또는 키에 따른 체중 등 , 한 변수의 변경이 다른 변수의 변경과 연관됩니다.



지도학습의 두가지 형태: 회귀 예시

- 그럼 습도와 온도라는 두 가지 변수를 고려해 보겠습니다.
- 여기서 '온도'는 독립 변수이고 '습도'는 종속 변수입니다.
- 온도가 높아지면 습도가 낮아집니다.
- 이 두 변수는 모델에 제공되고 기계는 이들 간의 관계를 학습합니다.
- 기계가 훈련된 후에는 주어진 온도를 기반으로 습도를 예측할 수 있습니다.

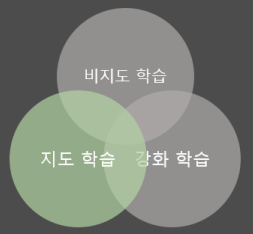


Input	Target
구름의 양 (the amount of cloud)	우천 (Rain)
0.67156	X
0.31158	X
0.99740	○
0.11598	X
0.49894	X
0.13498	X
0.77894	○

7 Input

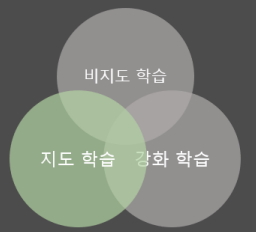
지도학습 그 외 예시 1 : 훈련 데이터

- 데이터 예시를 보면 프로그램한테 구름의 양에 따라서 실제로 비가 내렸는지에 대한 과거 데이터들을 넣어주고 이것을 훈련 데이터라고 부른다.
- 과거의 입력과 출력에 대한 데이터들을 주고 이 데이터의 패턴을 확인하여 기준을 만드는 게 지도 학습이다.
- 이 훈련된 데이터에 새로운 입력을 넣으면 그 예상 결과를 나타내 준다.



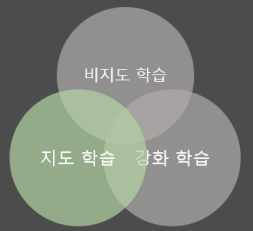
지도학습 그 외 예시 2 : 패턴분류

- 패턴 분류에서는 여러 가지 패턴 값을 학습하여 예측 또는 결과값을 출력합니다.
- 이 출력 값은 미리 정해진 분류 값이며 '라벨(이름)'이 붙어 있습니다.
- 예를 들어 개인가 치킨인가? 라는 문제에서 개와 치킨 라벨이 주어진 여러 사진 데이터집합(패턴)패턴으로 학습하여 어느 것인지 식별하는 시스템을 들을 수 있다,

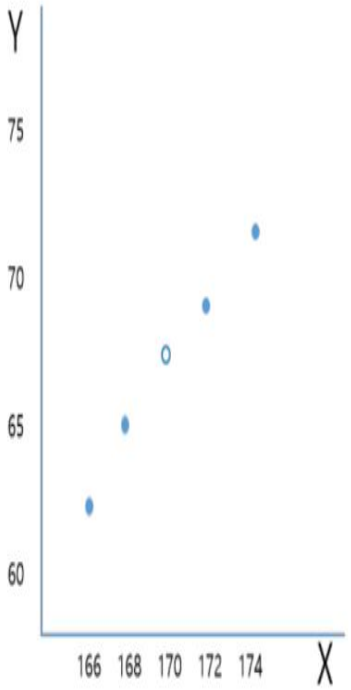


지도학습 그 외 예시 2 : 패턴분류

- 기계 학습 중 딥러닝 에서는 패턴 분류에 해당하는 값(개, 치킨)의 분류 기준을 스스로 찾을 수 있다.
- 즉 딥러닝(인공신경망)에서는 패턴 분류 알고리즘을 자동을 만드는 것입니다,
- 이렇게 지도학습을 완료한 모델은 처음 보는 사진이라도 개와 치킨을 구분할 수 있게 됩니다.
- 훈련 데이터의 질, 양, 모델 알고리즘에 따라 다르겠지만 훈련 데이터로 사용된 사진과 너무 많이 차이가 나는 사진이 입력되면 판단을 못하는 경우도 있습니다.

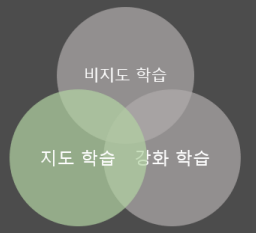


키	몸무게
166	63
168	65
170	?
172	69
174	71
176	74



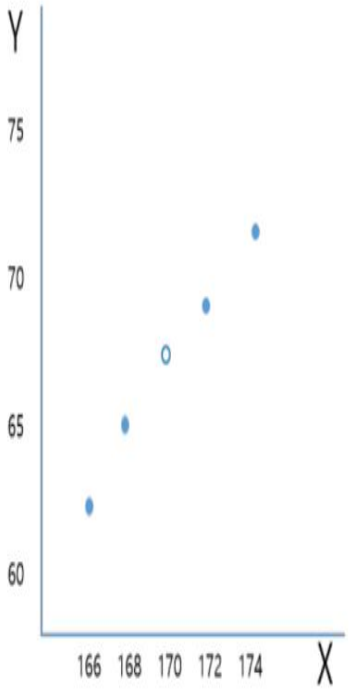
지도학습 그 외 예시 3 : 희귀분석

- 희귀 분석은 입력과 출력이 연속되는 수치로 주어졌을 때 이들의 관계로부터 새로운 출력 값을 도출하는 문제입니다.
- 예를 들어 정상 체중 범위에 있는 사람의 키와 몸무게 데이터 집합을 각각 X축, Y축에서 선형(직선, 곡선 등)으로 표현하여 특정 키를 가진 A라는 사람의 몸무게를 구하는 문제 등입니다.
- 희귀분석은 모델의 특성으로 인해 지도학습에서 최적의 결과를 도출합니다.

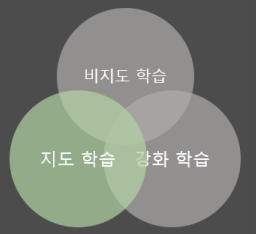


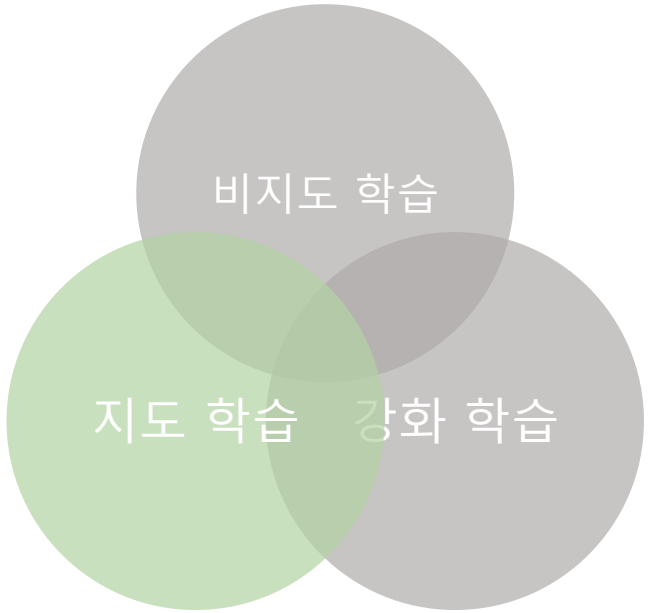
지도학습 실제적용1

키	몸무게
166	63
168	65
170	?
172	69
174	71
176	74



- 희귀분석은 모델의 특성으로 인해 지도학습에서 최적의 결과를 도출합니다.





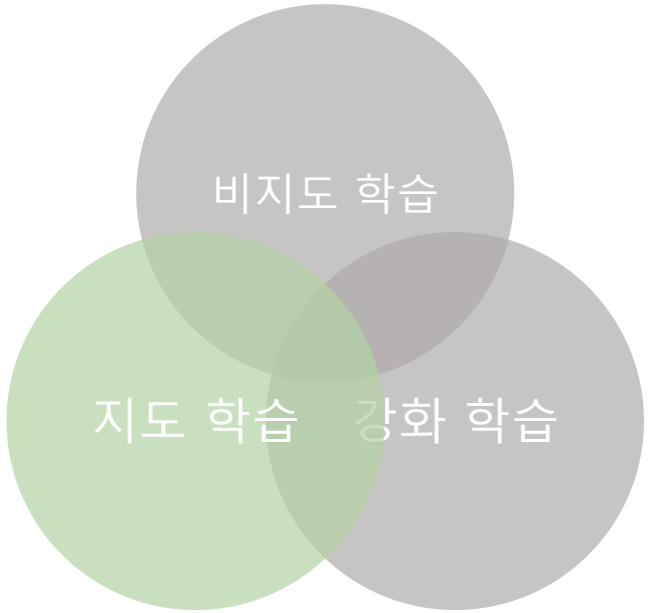
지도학습의 실제 적용

- 위험성 평가 (Risk Assessment)

  -> 지도 학습은 회사의 위험 포트폴리오를 최소화하기 위해 금융 서비스 또는 보험 영역의 위험 평가 하는 데 사용됩니다.

- 이미지 분류 (Image Classification)

  -> 이미지 분류는 지도 머신 러닝을 시연하는 주요 사용 사례 중 하나입니다. 예를 들어 Facebook은 태그가 지정된 사진 앨범의 사진에서 친구를 인식할 수 있습니다.



지도학습의 실제 적용

- 사기 탐지 (Fraud Detection)

  -> 사용자의 거래의 진위 여부를 식별합니다.

- 시각적 인식 (Visual Recognition)

  -> 사물, 장소, 사람, 행동, 이미지를 식별하는 머신 러닝 모델의 능력입니다.