



Machine Learning

Machine Learning 정의

Machine : 수학적 함수

- 컴퓨터를 학습하는 것이 아니라 컴퓨터로 수학적 방법을 통해 학습하는 것!
- Machine == Function == Model == Algorithm

Learning : 학습, 변화

- 변화는 긍정적일수도, 부정적일수도 있다 == Risk

Machine Learning 종류

지도 학습

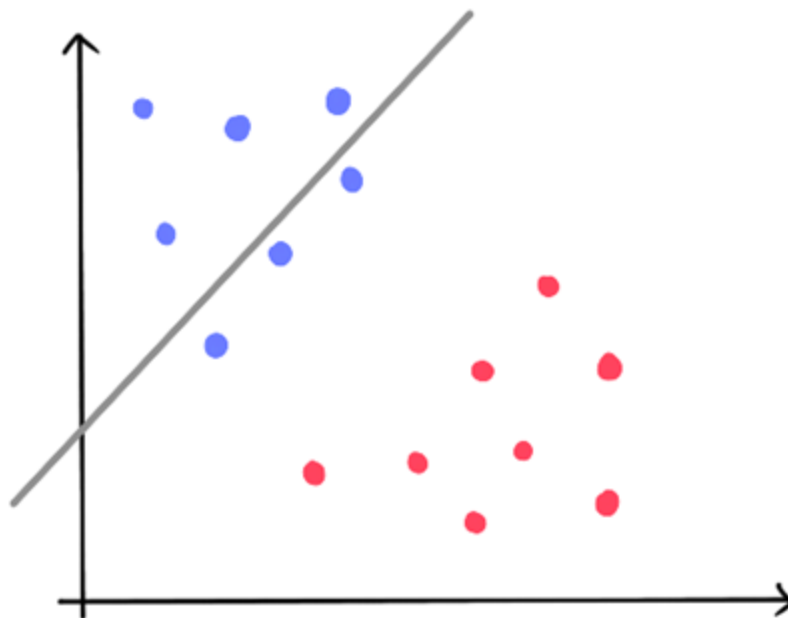
- Regression (수치 예측)
- Classification (범주 예측)

비지도 학습

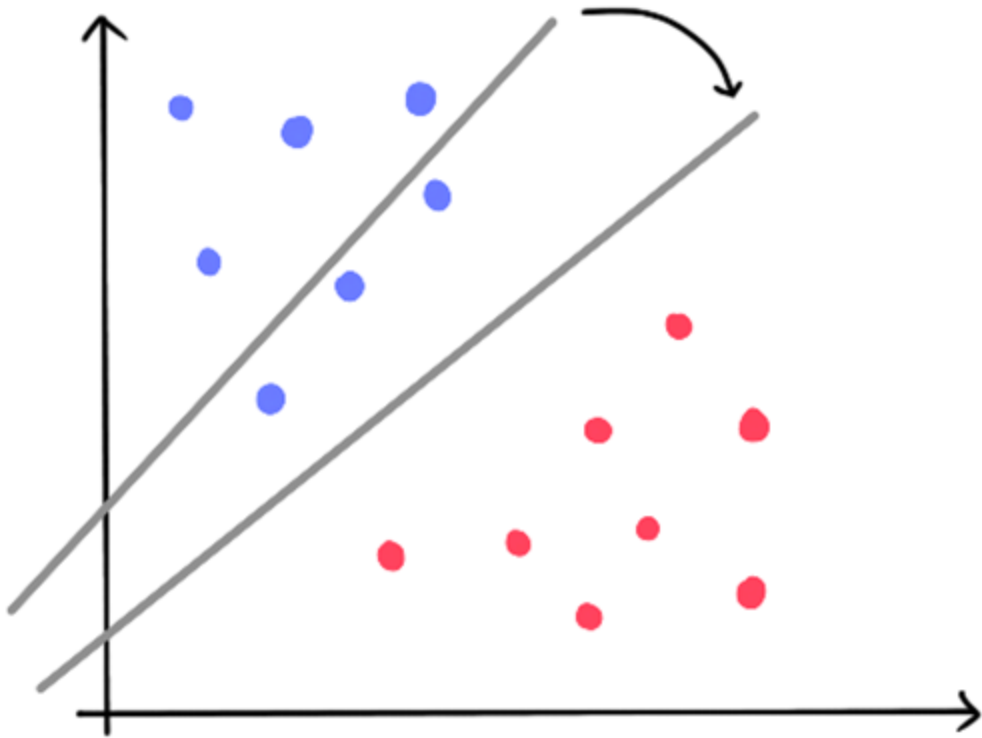
Classification로 예를 들어 이해하기



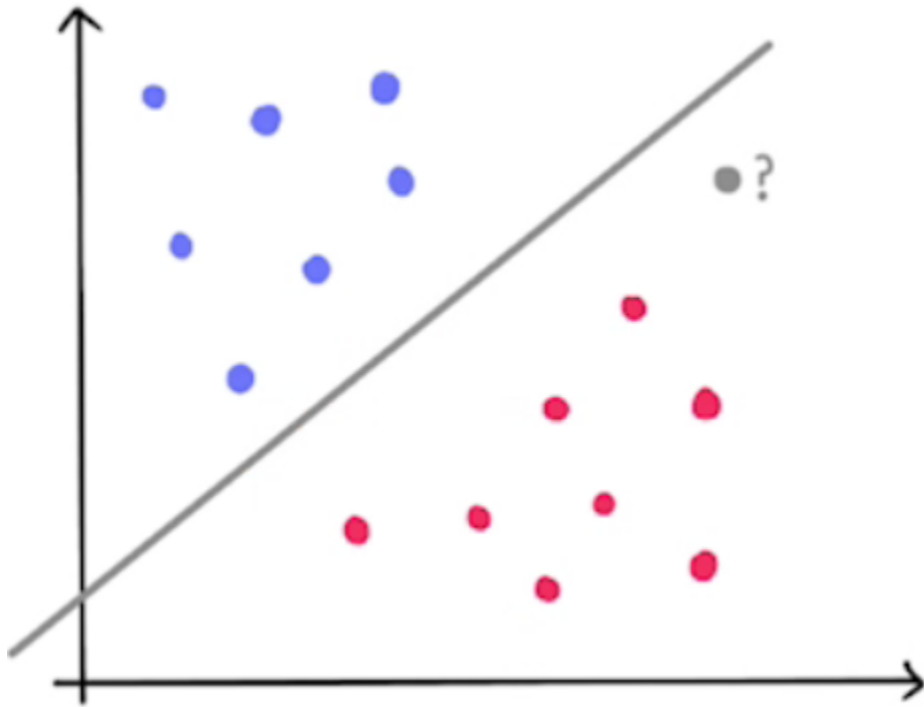
빨간색 점과 파란색 점(Data)을 분류하려 한다.



빨간색 점과 파란색 점을 나누는 선(Function, Machine) $y = wx + b$ 를 긋는다.



이 선의 위치를 변경(Change, Learning)하며 두 종류의 점을 분리하는 적절한 w , b (parameters)를 찾는다.(최적화)



점들을 분리하는 $y = wx + b$ 함수(Model)를 가지고 새로운 점(미래의 Data)의 색을 예측한다.

Gradient Descent

경사 하강법

모든 학습의 원리가 되는 것, Optimization Method

$y = wx + b$ (w: weight, 가중치 / b: bias, 편향)

Loss Function

실제 값과 예측 값의 차이 \Rightarrow 오차 (Error, Loss, Cost)

$$L(y, y_{\text{hat}}) = (y - y_{\text{hat}})^2$$

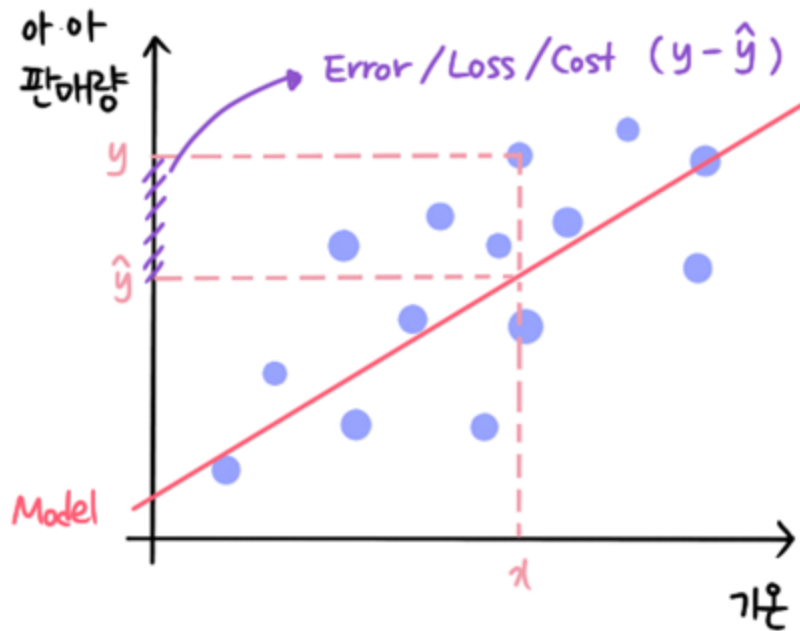
Mean Squared Error

Loss들의 평균

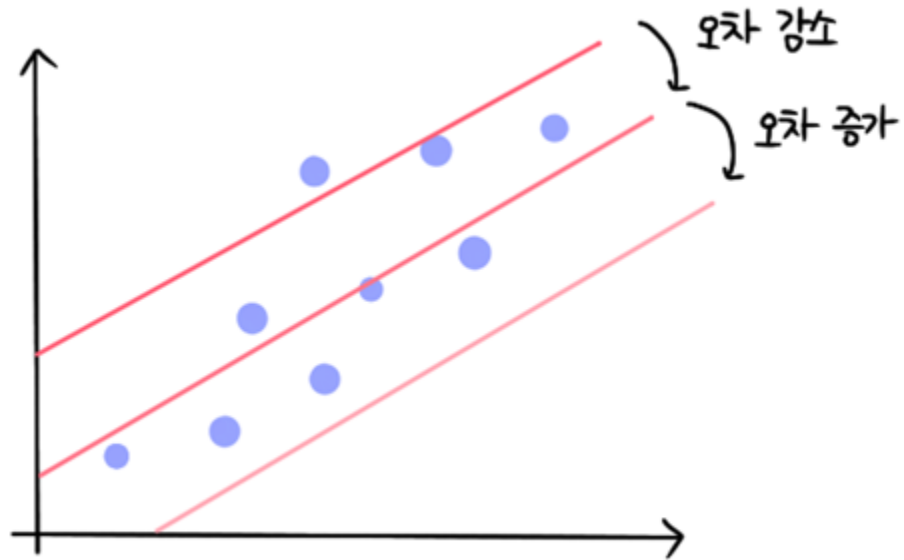
$$\text{mean}(L(y, \hat{y})) = \text{mean}(y - (wx + b))^2$$

Regression Analysis로 예를 들어 이해하기

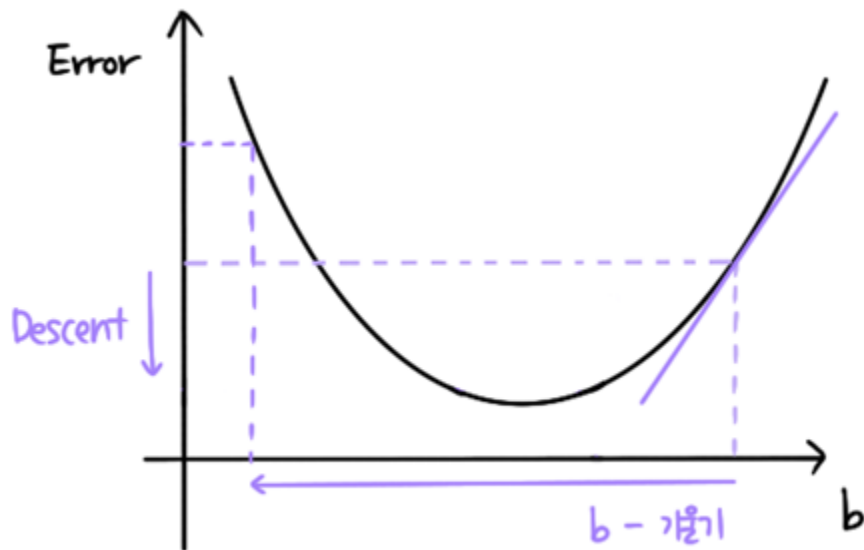
학습 평가 기준 : 오차 함수, 예측 모델의 경우 MSE



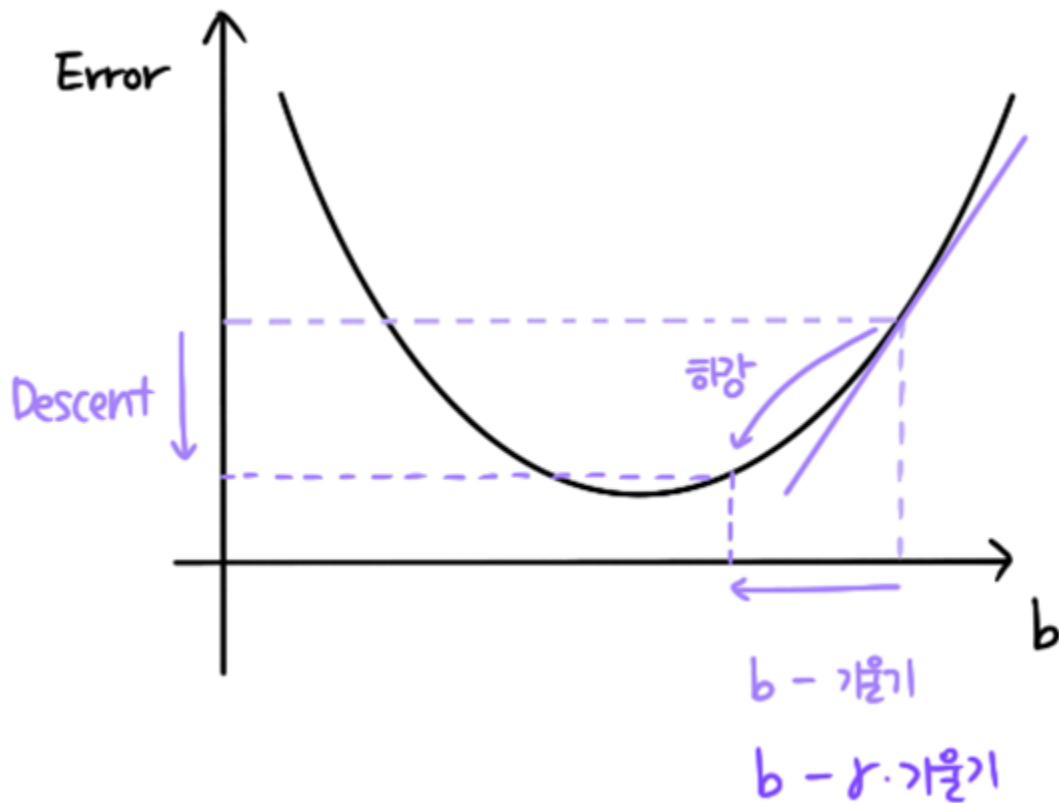
실제값과 예측값의 차이 : Error, Loss, Cost



$y = wx + b$ 에서 b 의 값이 바뀌면 오차가 감소하다가 어느 순간 오차가 다시 증가



b 에서 접선의 기울기를 빼면 경사(Gradient)가 하강(Descent $\Rightarrow b = b - \text{기울기}$)
하지만 너무 많이 이동하면 오차가 다시 증가하게 됨



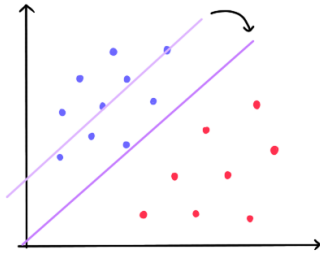
$$b = b - r * \text{기울기}$$

0~1 사이의 값을 가지는 감마를 기울기에 곱해서 너무 많이 이동하지 않도록 조정

(너무 작으면 학습 시간이 오래 걸리고 너무 크면 발산하게 된다)

감마 : **Learning Rate, Step Size**

Step 값으로 무슨 값을 줄지는 직접 정하는 것, 해보지 않고는 모른다 \Rightarrow Hyperparameter



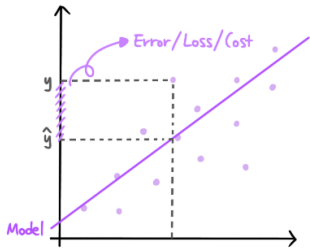
(DATA)
점들을 색깔별로 분류하기

(Label) (Label)
파란색 점들과 빨간색 점들을 구분

(Function, Machine)
선을 그리고 이 선이 두 종류의 점들 나누는

(parameter) (요인화) (Learning, Change)
적절한 w, b 찾기 (선을 변경하여)

(Model) (여러 DATA)
이 선을 가지고 새로운 점은 무슨 색깔지 예측하기



y : 연속형 데이터
 $y = wx + b$ (w : weight, 가중치
 b : bias, 편향)

Loss Function () : 실제값과 예측값의 차이
 $L(y, \hat{y}) = (y - \hat{y})^2$

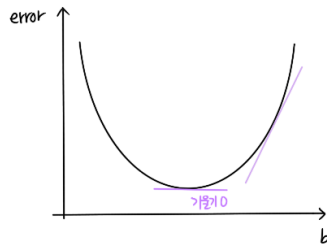
MSE (Mean Squared Error) : Loss들의 평균

$$\text{Cost}(y, \hat{y}) = \text{mean}(L(y, \hat{y})) \\ = \text{mean}((y - (wx + b))^2)$$

$$b = b - \delta \cdot db \\ w = w - \delta \cdot dw \quad (\delta: 0 \sim 1)$$

$b, w \Rightarrow \text{hyperparameter}$

가중치 0 \Rightarrow 더 이상 학습이 이뤄지지 않음



```
def Machine(x, w, b):
```

```
    y-hat = (w * x) + b
```

```
    return y-hat
```

```
def Gradient(x, y, w, b):
```

```
    y-hat = Machine(x, w, b)
```

```
    db = np.mean((y - y-hat) * (-2 * x))
```

```
    dw = np.mean((y - y-hat) * (-2))
```

```
    Loss = np.mean((y - y-hat) ** 2)
```

```
    return db, dw, Loss
```

```
def Learning(x, y, w, b, step):
```

```
    db, dw, Loss = Gradient(x, y, w, b)
```

```
    ub = b - (step * db)
```

```
    uw = w - (step * dw)
```

```
    Loss = Loss
```

```
    return ub, uw, Loss
```

```
for i in range(0, 1500):
```

```
    uw, ub, Loss = Learning(DATA.inputs, DATA.outputs, w, b, step)
```

```
    b = ub
```

```
    w = uw
```

```
    Error.append(Loss)
```