

Learning method

- Supervised learning : labeled data, exit loss,
 - ↳ ex) classification, regression,
 - ↳ $O, X / A, C$ \rightarrow $\frac{A-C}{A}$ \rightarrow $\frac{A-C}{A}$ 가격 어느 정도
- Un Supervised learning : no labeled data, no loss
 - ↳ ex) Clustering, GAN, K-mean,

\Rightarrow labeled 된 data 부족해 등장하였다.

- Semi Supervised learning
 - ↳ Labeled 100개로 학습, 그 후 un labeled 1000개 예측 후 그 예측 값 토대로 학습한다.
- Self Supervised learning
 - ↳ un labeled data 제공해 학습가능하게 만들어 (feature 등을 활용) Supervised learning 진행

Confusion Matrix

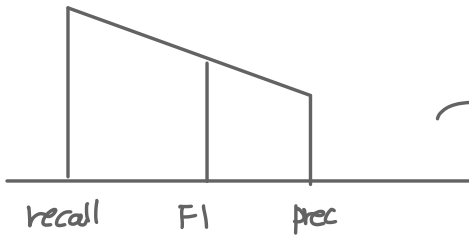
		Actual	
		T	F
Pred	T	TP	FP
	F	FN	TN

⇒ 10개중 dataset 이 9개가 T,
1개가 F 인 모델은 무조건 T라고만
하게된다. 이것을 하려면
precision, recall, ROC curve 나쁨

- precision (정밀도) (모델양성) = $\frac{TP}{TP + FP}$ (남지 않는다 예측했음에 실제 맞은 비율)
- recall (재현율) (hit rate) (정답양성) = $\frac{TP}{TP + FN}$ (실제 맞은 남지중 모델이 맞다고 한 비율)

⇒ 한쪽만 고려하는것은 올바른 모델 아님

F1 - score = $2 \times \frac{1}{\frac{1}{\text{prec}} + \frac{1}{\text{recall}}} = 2 \frac{\text{prec} \times \text{recall}}{\text{prec} + \text{recall}}$ (조화 평균)



기하학적 의미
(각분포를 재가산 조화평균)

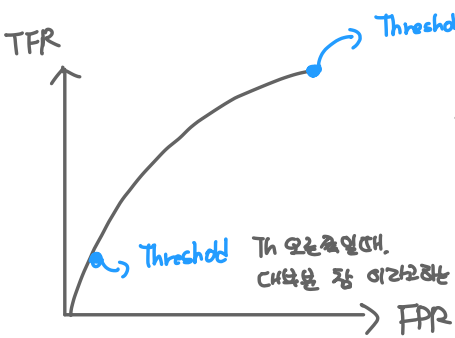
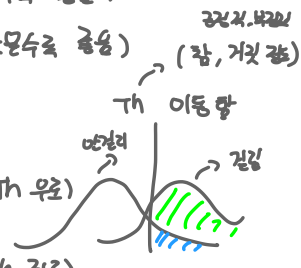
ROC Curve

=> 성능 평가 그래프는 왼쪽부터 높음수록 좋음

TPR (True Positive Rate) => 앓음 많이라고 한 비율 (높음수록 좋음)
 (FPR (False ") => 앓지않는데 앓이라고 한 비율 (낮음수록 좋음)

=> 하지만 둘은 trade-off 한다

혹시나 라며 앓이라고 판단 앓이한 의사 => TPR ↑, FPR ↑ (Th 무리)
 앓이 아니라고 " " => TPR ↓, FPR ↓ (Th 리로)



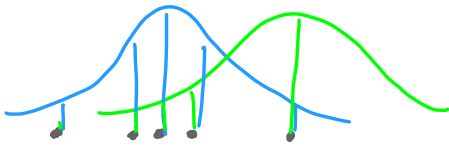
Threshold 왼쪽일때 즉, 대부분 앓이라고 한 부정적인 의사
 => FPR이 낮고 TPR이 높음수록 좋음 (좌. 위)
 => 모든 Threshold (모든 의사선생님) 를 고려하여 TPR, FPR 을 보겠다.

* 곡선의 형태도 => classification 을 얼마나 더 잘하는지 나타내는 척도

MLE (Maximum likelihood estimation)

pdf = 확률밀도함수 $\int_{-\infty}^{\infty} f(x) dx = 1$ 이므로 확률 잘 나오지 않게끔

pdf를 모를때 data를 통해 pdf의 파라미터를 추정하는 방법



=> • 데이터를 얻었다면 이 데이터는
파라미터를 추론하는 데 사용된다

=> 직관적으로 보면 파라미터를 이룰 likelihood (가능도)의 공으로 계산

가능도 (likelihood)를 $p(x|\theta)$ 라 하며 이의 공인 likelihood function의 정의

$$p(x|\theta) = \prod_{k=1}^n p(x_k|\theta) \quad \text{이때 } p(x_k|\theta) \text{의 값이 가장 커지게 하는}$$

θ 값을 찾는것이 목적 \Rightarrow 따라서 이 도는 것 찾을

그때 찾는 분포의 평균값 (확률 잘 나오도록 조정값으로 쓰인다.)

* 로그도 공차환산기 때문에 \log 를 붙일수 있으며 $\times \rightarrow +$ 되기때문에
 \log likelihood 자주 사용한다

$$L(x|\theta) = \log p(x|\theta) = \log \prod_{k=1}^n p(x_k|\theta) = \sum \log p(x_k|\theta)$$

Overfitting 막는 방법

⇒ dataset 늘리기

⇒ data augmentation

⇒ 모델 complex 줄이기

⇒ 가중치 규제 (regularization) $L1$ norm, $L2$ norm

⇒ Drop Out ⇒ 신경망 일부는 사용하지는 것

Logistic Regression (2도함수)

⇒ data가 어떤 범주에 속할 확률은 0~1 값으로 예측하고 그 확률에 따라 가능성이 더 높은 범주에 속하는 것으로 분류



⇒ Linear regression은 $-\infty \sim \infty$ 으로 값이 뻗어 나간다. 이는 말지않는 상황들이 있기 때문에 확률로 만들어 범위를 0~1로 만들어 표현

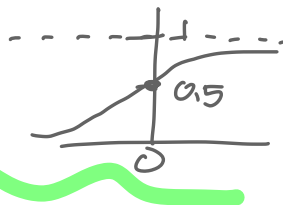
$$\text{Odds} = \frac{\text{일어날 확률}}{\text{안 일어날 확률}} = \frac{x}{1-x} \quad \log(\text{Odds}) = 0 \sim 1 \text{ 개}$$

여기서 (feature들 \times coeff)_n + intercept 0으로 표현함.

$$z = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$

이를 sigmoid로 표현

$$h(z) = \frac{1}{1+e^{-z}} \Rightarrow$$



Soft max regression

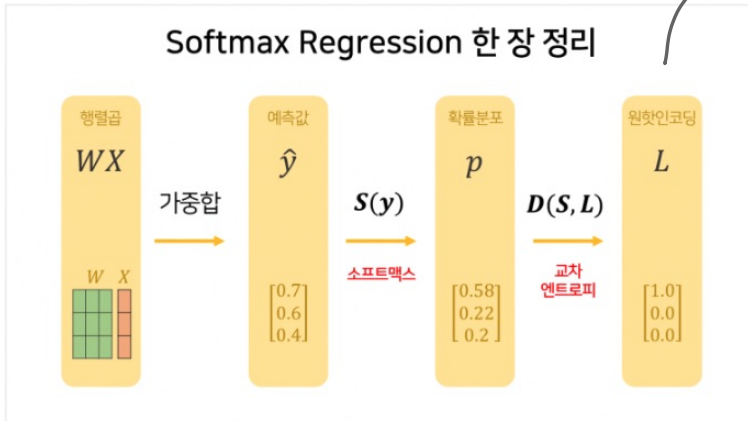
⇒ 로지스틱 회귀를 multi classification O/C.

(로지스틱은 성공/실패 2개 분류, soft max는 A,B,C 처럼 3개 이상 classification)

- 회색을 포함한 1의 확률 분포로 바꾸기 위해 이를 $S(y)$ 로 표현 되고 이를 거쳐 확률 분포 p 로 바뀐다

$$S(b_i) = \frac{e^{b_i}}{\sum_k e^{b_k}}$$

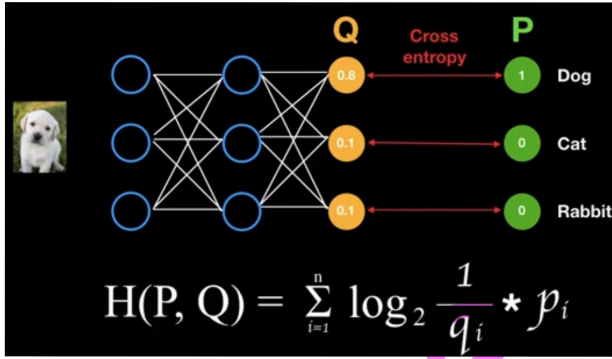
전체 그림



1 2 3. 가중치 따르니
0과 1로 가중치 없애고
표준화 효율적임!

Cross Entropy

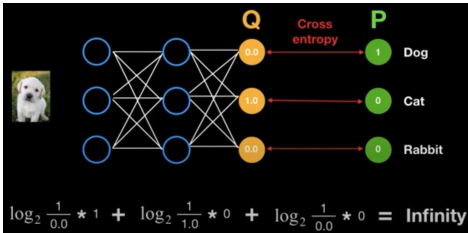
↳ 전달수 있는 정보 (모델의 결과) 는 구한 엔트로피



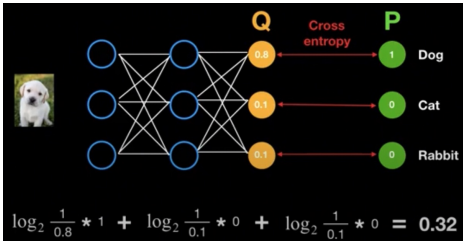
→ 그냥 엔트로피 식은

$$= \log_2 \frac{1}{p_i} * p_i$$

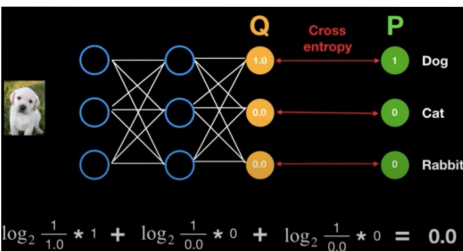
예측한 Q와 실제 결과인 P를 (Cross) 둘을 얼마나 근사하게 표현하고 싶은지



→ 학습이 안되는 상태 (완전 틀림)
Cross Entropy 는 극값 ∞



⇒ 어느정도 학습 되어 (꽤나 맞음)
0.32 나옴



⇒ 그냥 100% 다맞출 경우 = 0

loss Function 으로 사용
⇒ 이를 줄이도록 업데이트 한다.

❏ Cross Entropy function

$$\text{cost}(w) = - \sum_{j=1}^k b_j \log(p_j)$$

↳ k 개의 Class가 있을때 b_j 는 one-hot-vector의 값
 p_j 는 j 번째 Class의 확률 (\hat{b}_j 으로도 됨)

실제로 c 가 one-hot-vec 이고, $p_c = 1$ 이면 $-|\log| = 0$ 이므로,

$\text{cost}(w)$ 를 최소화 하는 방향으로 해야 하고, data n 개 class k 개일

$$\text{cost}(w) = - \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k b_j^{(i)} \log(p_j^{(i)}) \text{ 가 되어야 한다.}$$

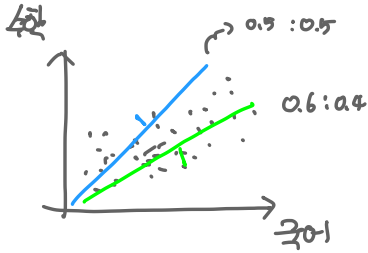
❏ Binary Cross Entropy (이진이라 한단것 뿐!)

$$\begin{aligned} \sum_{j=1}^2 b_j \log p_j &= b_1 \log p_1 + b_2 \log p_2 \\ &= b_1 \log p_1 + (1 - b_1) \log (1 - p_1) \end{aligned}$$

$$\approx \text{cost}(w) = - \frac{1}{n} \sum_{i=1}^n (b^{(i)} \log p^{(i)} + (1 - b^{(i)}) \log (1 - p^{(i)}))$$

PCA (Principal Component Analysis)

⇒ 데이터 분포에서 어떤 선으로 잘 설명 시킬때 데이터 분포 잘 표현 되었나? (각각 어떤 공할까? 변형 비율 (비율) 작)



⇒ 이렇게 변형 비율 다

5:5로 변형한지 6:4로 변형한지
데이터 분포따라 우리가 잘 변형한지
다르다! 최적의 projection 할 line 찾기

↳ 2차원 데이터를 2개의 변수를 찾아야 한다

principle component (주성분) = data의 분산이 가장 큰 방향 벡터 (잘 퍼짐)

ex) 얼굴 인식, 1000개 사진 받아 1000개 주성분 벡터들 분산이 큰 20개 받아서 사용 (앞면부터 얼굴 전반적 형태, 뒤면부터 세부적)

▣ K-mean clustering

⇒ 키와 몸무게에 따라 옷을 생생 할때, 모든 사이즈 고거할수 없어
S.M.L로 사이즈 나눌때 이용가능 하다.

* Centroid = Cluster의 중심

데이터 분할, K 결정. Centroid 설정

while (변함 있음)

모든 데이터 Centroid와 유클리드 거리로 분류

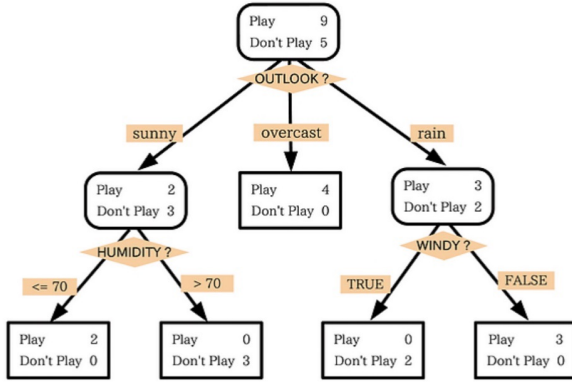
Centroid를 cluster된 데이터의 공간으로 다시 설정

다시 새로운 Centroid로 모든 데이터 유클리드 거리로 분류

⇒ K 최적 값문라 단점

Decision Tree

Dependent variable: PLAY



=> 결문문 먼저 분류하여
input 이 무엇인지 예측함

(불확실성)

=> 결문문 어떤것을 먼저 선택하냐에 따라 다르게 어떤 사용도는 개념 **entropy**

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i$$

↓
확률 ↓
정보의 양 entropy

일반적 동전	50%	50%	1	=> 앞뒤 맞추기 어렵, 불확실성 높다
앞면만 있는 동전	100%	0%	0	=> 확실함, 불확실성 0
구분리권 동전	90%	10%	0.47	

=> 정류의 수가 2^n 개일때 entropy 최댓값 n

=> Decision tree 만들때 **entropy 낮은 (확실한)** 결문문 우선
으로해 초반에 잘 걸러낸다

Naive Bayes Classify

$$P(B|A) = \frac{P(A|B) \times P(B)}{P(A)}$$

NBR 을 사용해 Classify 한다

(
ex) SPAM 메일 분류

$$P(\text{spam} | \text{free}) = \frac{P(\text{free} | \text{spam}) \times P(\text{spam})}{P(\text{free})}$$

(메일에 free란
단어가 있을 때 spam일
확률)

=> 하지만 단어가 train 할때 없으면 확률 0 이 되는데

각 확률에 대해 분모 분자에 +1 더해주는 Laplace Smoothing 사용하기도
한다,

SVM (support vector machine)

=> margin 을 최대화 (class 잘 분류) 하는 선을 찾는 것

- * Support Vector = 선과 가장 가까운 point
- * margin = support vector 와 선의 distance
- * decision boundary = 두 data 구분하는 선
- * robust 하다 = outlier의 영향을 덜 받는다.

		mean	median
(,)	1 2 3 4 5	3	3
	1 2 3 4 100	22	3
		↓	↓
		not robust	robust
		(outlier에 취약. 강건)	

-> 무조건 margin이 큰 선이 분류 잘하리 않을까?

과해서 나온 데이터를 분류하는 법이 많지 않을까?

그 범위 안에서 margin 최대화 하는 선 찾는 것

but 무조건 outlier 있는 경우 어느정도 outlier 무시라고 선찾을

-> Gamma 과다 decision boundary 휘어질, 작다. 직선에 가깝다.