

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

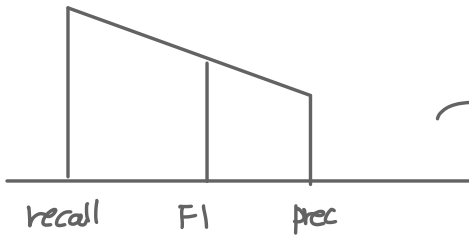
		Actual	
		T	F
Pred	T	TP	FP
	F	FN	TN

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

- precision (정밀도) = $\frac{TP}{TP + FP}$ (남시 많다 예측했음에
(모델 많음) 실제 많은 만큼)
- recall (재현율) = $\frac{TP}{TP + FN}$ (실제 많은 남시 중
(hit rate) (정답 많음) 모델이 많다고 한 확률)

⇒ 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)

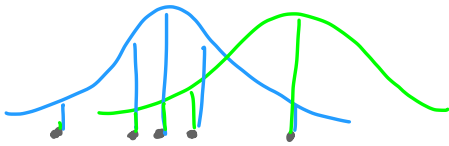


기하학적 의미
(각분포를 재가산 조화평균)

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 추론할 때 분포에서 맞는지 여부를 추정

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

가능도 (likelihood)를 $p(x|\theta)$ 라 하며 이들의 곱을 likelihood function 으로 정의

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

θ 값을 찾는것이 목적 \Rightarrow 따라서 이 도는 것 찾기

그때 찾는 분포의 평균값 (확률 잘 나오게끔 조정값으로 쓰게함.)

* 로그도 곱셈보다 덧셈에 의해 분할을 쉬게하여 $\times \rightarrow +$ 되기때문에
 \log likelihood 자주 사용함

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

Logistic Regression (지도학습)

⇒ 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

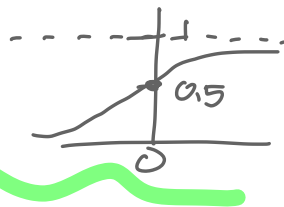
$$\text{Odds} = \frac{\text{일어날 확률}}{\text{안 일어날 확률}} = \frac{x}{1-x} \quad \log(\text{Odds}) = 0 \sim 1 \text{ 개}$$

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

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

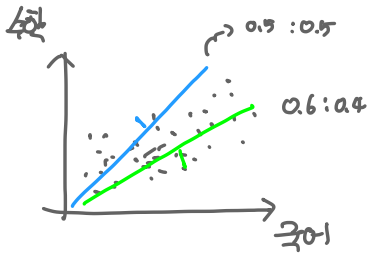
이를 sigmoid로 표현

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



PCA (Principal Component Analysis)

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



⇒ 이렇게 변형 비율 다

5:5로 변형할지 6:4로 변형할지
데이터 분포 따라 우리가 잘 변형할지
다르다! 최적의 projection 할 line 찾기

↳ 2차원 데이터를 2개의 변수로 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 생선할 때, 모든 사이즈 고쳐할 수 없어
S.M.L로 사이즈 나눌 때 이용 가능 하다.

* Centroid = Cluster의 중심

데이터 분할, K 결정. Centroid 설정

while (변함 있음)

모든 데이터 Centroid와 유클리드 거리로 분류

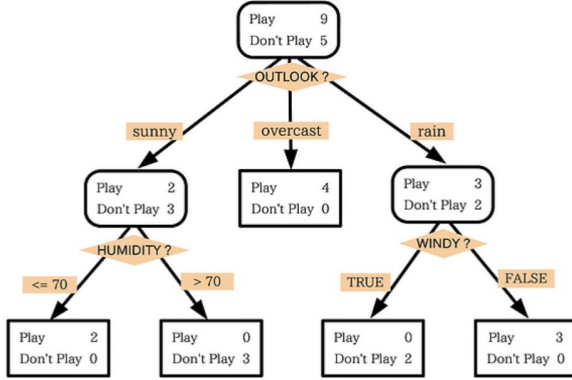
Centroid를 cluster된 데이터의 공간으로 다시 설정

다시 새로운 Centroid로 모든 데이터 유클리드 거리로 분류

⇒ K 최적 값은 무라 단점

Decision Tree

Dependent variable: PLAY



⇒ 결문문 먼저 분류하여
input 이 무엇인지 예측함

(불확실성)

⇒ 결문문 어떤것을 먼저 선택하는가에 따라 다르게 어떤 사용도는 개념 **entropy**

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

entropy

일반적 동전	50%	50%	1	⇒ 앞뒤 맞추기 어렵, 불확실성 높다
앞면만 있는 동전	100%	0%	0	⇒ 확실함, 불확실성 0
구리동전	90%	10%	0.47	

⇒ 정수의 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,

Soft max regression

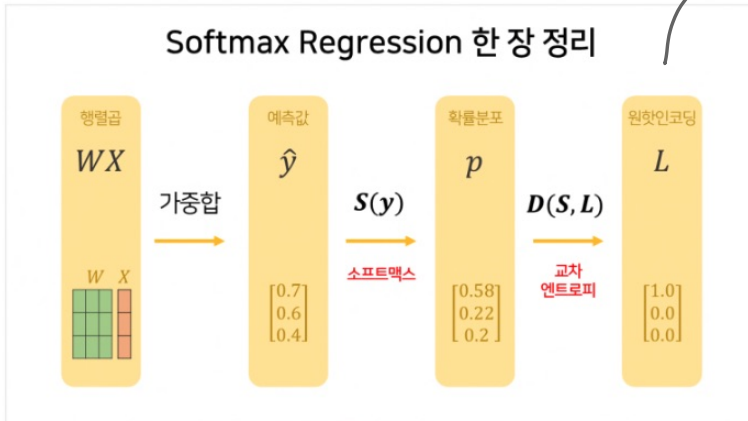
⇒ 로지스틱 회귀를 multi classification O/C.

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

- 확률 조합인 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!

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.