R 교육 세미나

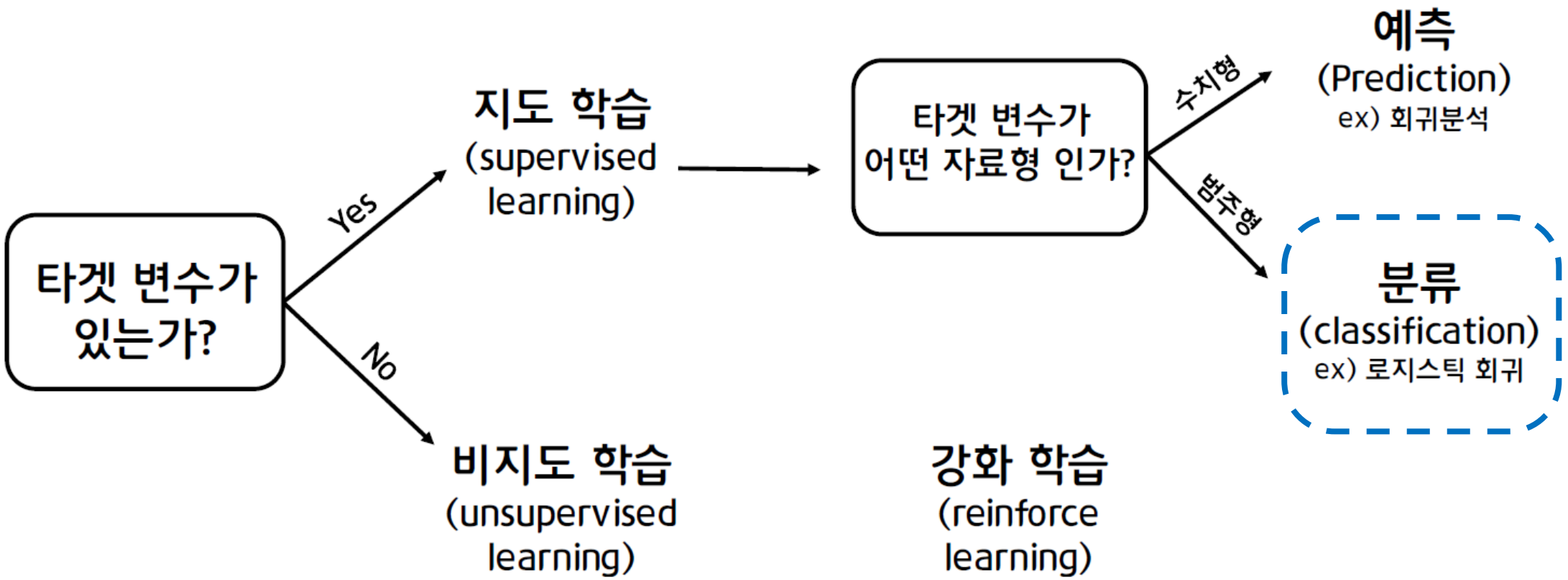
ToBig's 9기 최영제

KNN , LDA

Contents

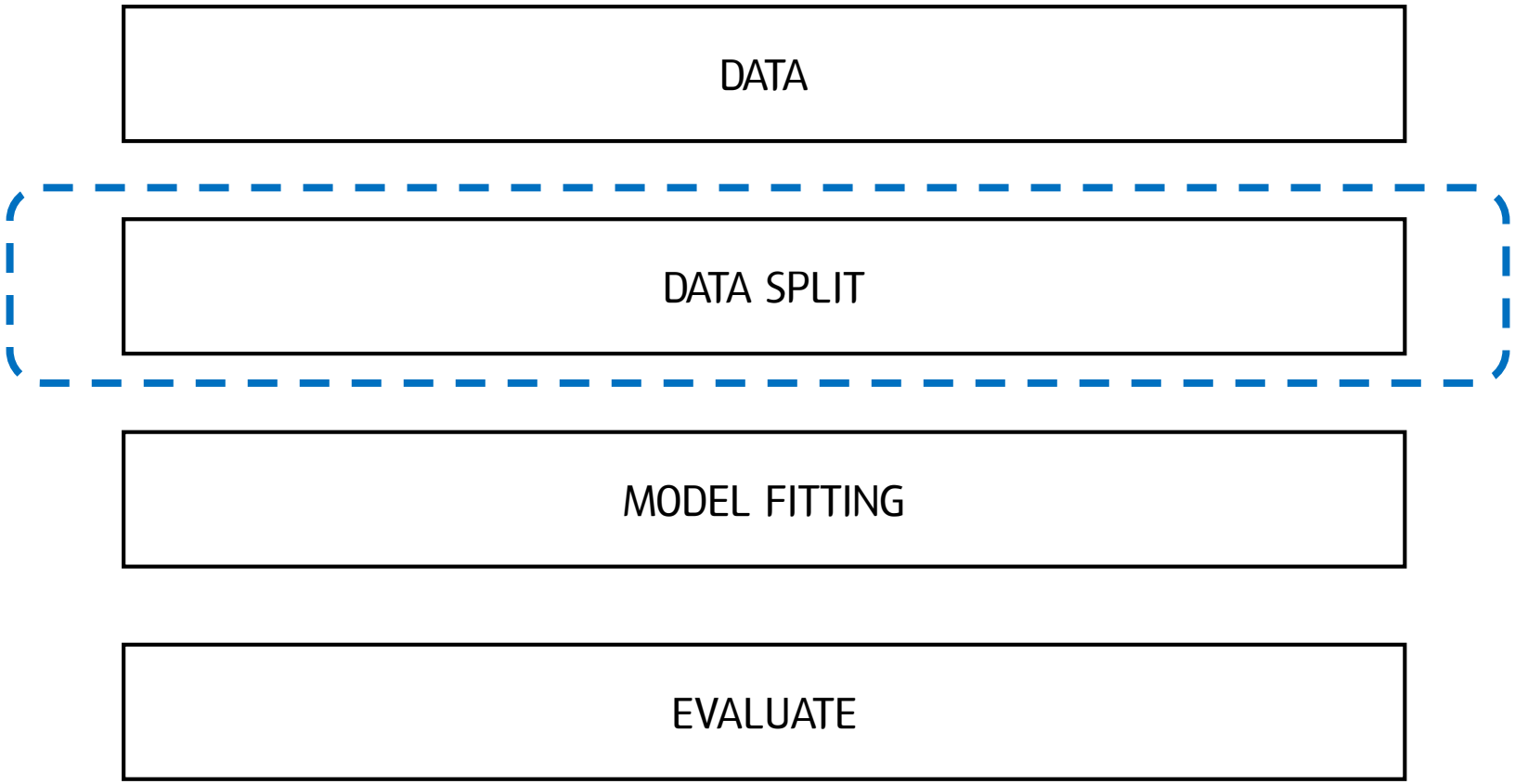
Unit 01 | ML 개요2

ML



Unit 01 | ML 개요2

ML



Unit 01 | ML 개요2

Data split

임의 추출

전체 데이터 : 9기 16명 10기 16명 = 32명

Train .75 : 32 * 0.75 = 24명
Test .25 : 32 * 0.25 = 8명

Unit 01 | ML 개요2

Data split

임의 추출

전체 데이터 : 9기 16명 10기 16명 = 32명

Train .75 : 32 * 0.75 = 24명
Test .25 : 32 * 0.25 = 8명

Train : 9기 16명 10기 8명
Test : 9기 0명 10기 8명

TEST_ACC가 1이더라도 좋은 분류기라 할 수 없다

Unit 01 | ML 개요2

Data split

비례 층화 추출

1 : 1

전체 데이터 : 9기 16명 10기 16명 = 32명

Train .75 : 32 * 0.75 = 24명
Test .25 : 32 * 0.25 = 8명

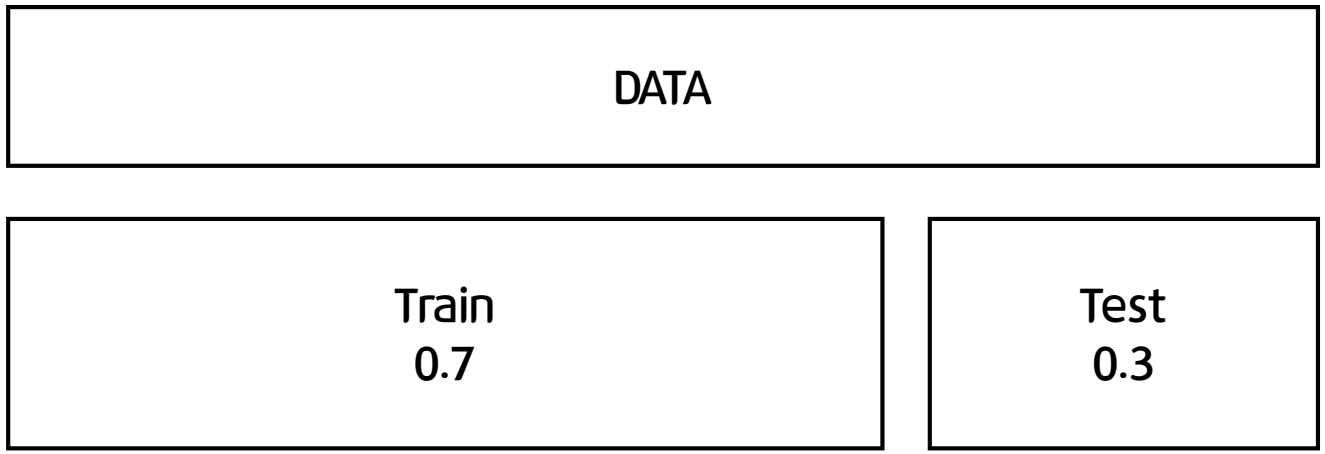
1 : 1

Train : 9기 12명 10기 12명
Test : 9기 4명 10기 4명

클래스 비율과 동일하게 train test 분할!

Unit 01 | ML 개요2

Data split



Test 데이터로 모델의 성능을 증가시키는 선택을 반복 시행

Unit 01 | ML 개요2

Data split

Test 데이터로 모델의 성능을
증가시키는 선택을 반복 시행

Test 데이터에 대해 과적합

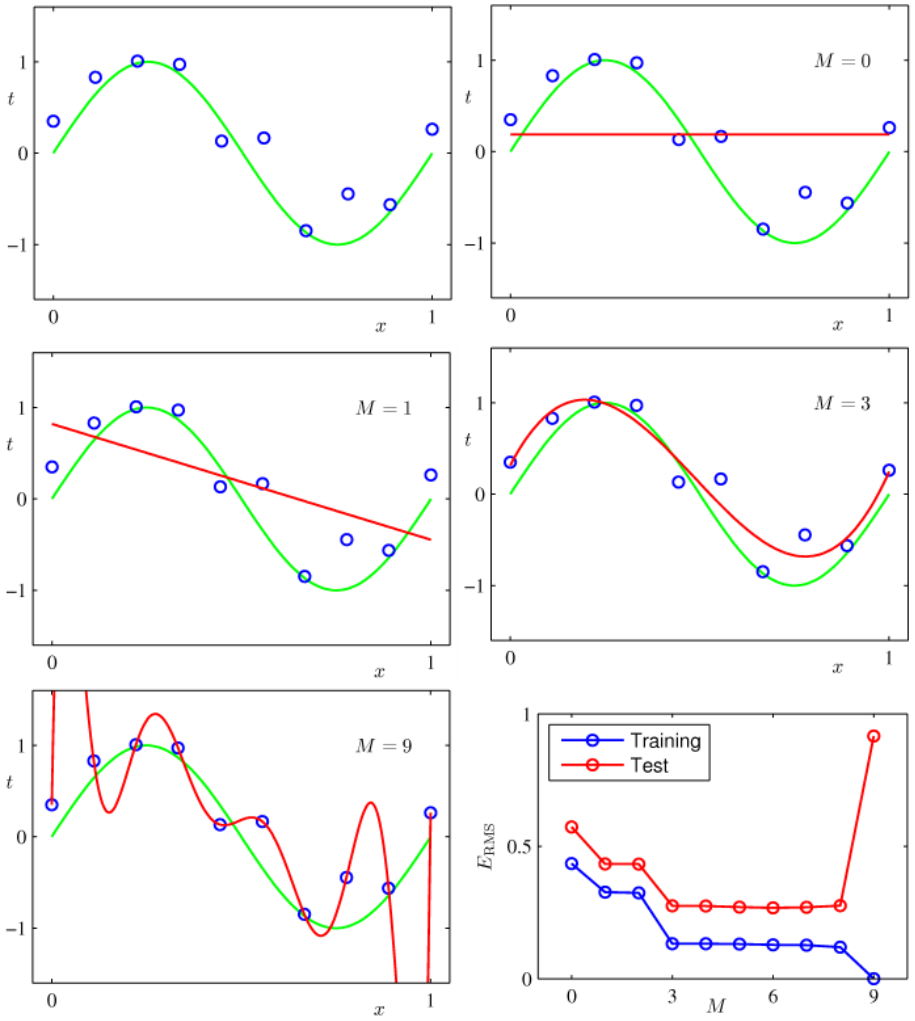
Unit 01 | ML 개요2

Data split

과적합 Over fitting

모델이 데이터에 잘 적합되면 예측력이 높아진다. 그러나 갖고 있는 데이터에 지나치게 적합되면 새로운 데이터가 들어왔을 때 좋은 예측력을 보이지 못한다. 이를 과적합이라고 한다.

왜 과적합은 예측력을 떨어트릴까? 모든 데이터에는 패턴과 잡음이 섞여있다. 패턴은 과거의 데이터와 미래의 데이터에 반복되지만, 잡음은 그렇지 않다. 모델이 데이터에 과적합 되면 이러한 잡음마저도 학습하기 때문에 새로운 데이터에 좋은 성능을 보여주지 못한다.

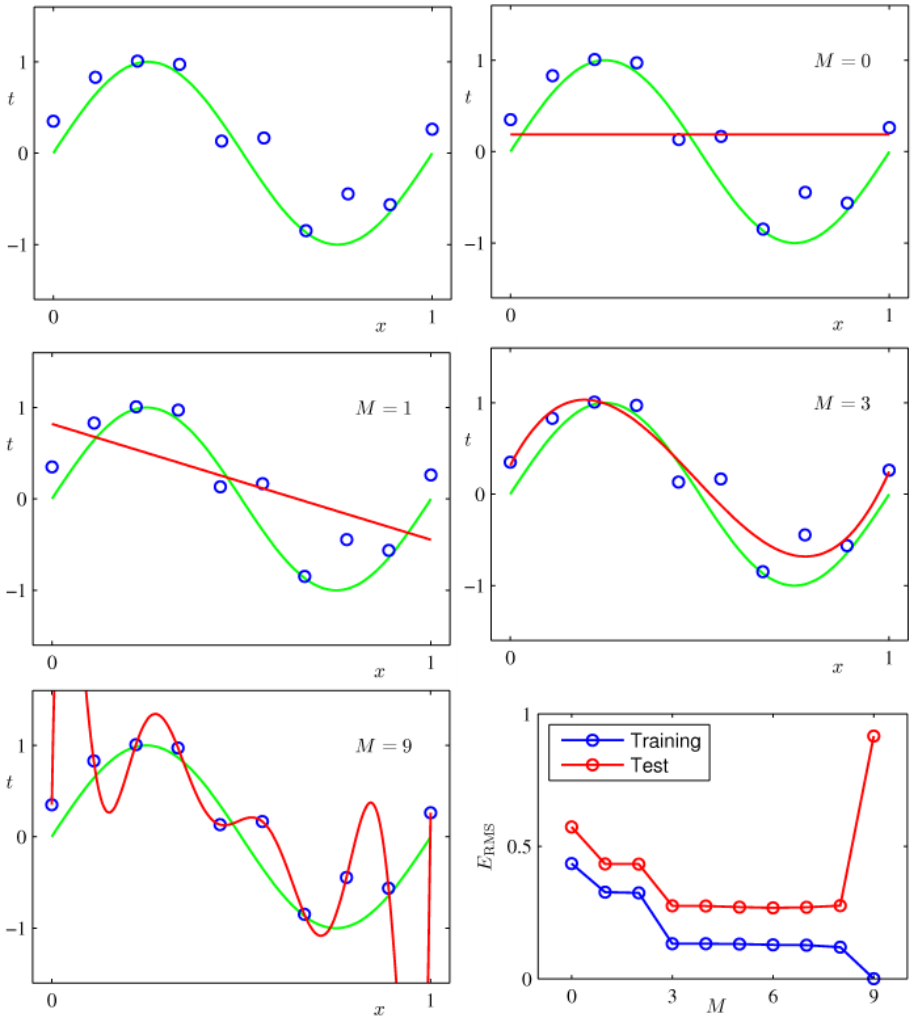


Unit 01 | ML 개요2

Data split

과적합 Over fitting

오른쪽 그림의 파란색 점은 초록색 직선에서 잡음을 더해서 만든 데이터이다. 모델이 단순하면(M = 숫자가 낮으면) Train데이터도 잘 못 맞추지만 모델이 복잡해지면 완벽하게 적합되어 error는 0이 된다.
그러나 train error가 0일 때의 모형(M=9,빨간색 직선)은 실제 분포(초록색)와 많은 차이가 있음을 알 수 있다.



Data split

전체 데이터를 train, test 두 그룹으로만 분리할 경우 발생할 수 있는 문제

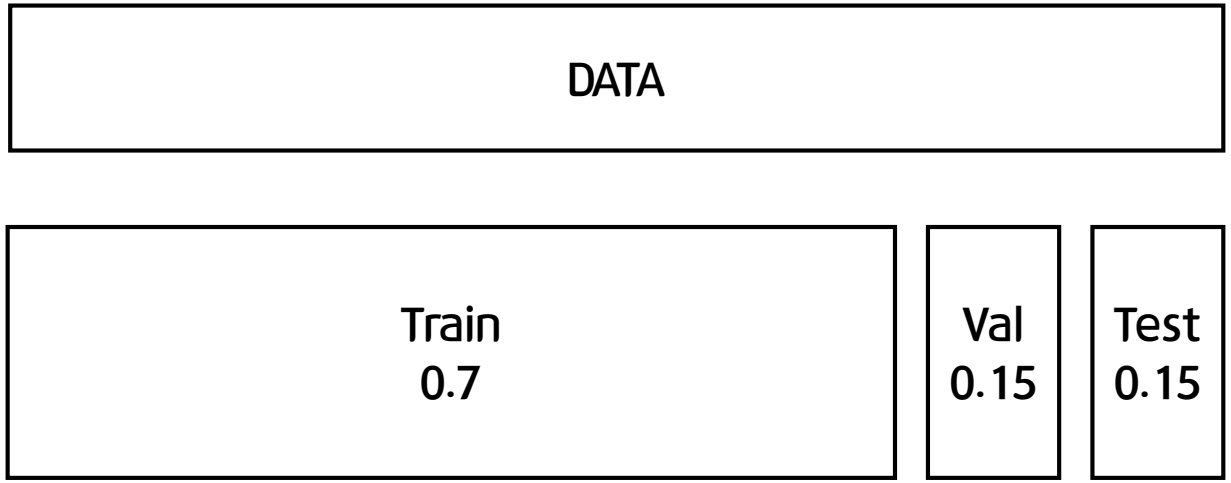
데이터를 나눈 목적에 위배된다.

미래 성능에 대한 정확한 추정치를 얻으려면 테스트 데이터 셋에 대한 성능이 모델에 영향을 미치지 않게 해야한다.
반복되는 테스트 결과를 바탕으로 최고의 모델을 선택하기 때문에 이 규칙은 사용자도 모르게 위반하기가 쉽다.
가장 좋은 결과를 선별했기 때문에 테스트 성능은 처음 보는 데이터에 대해 편향되지 않은 성능 척도가 아니다.

따라서 전체 데이터가 주어진 경우 훈련, 테스트, 검증 데이터 셋으로 나누는게 좋다.

Unit 01 | ML 개요2

Data split



홀드아웃 Hold out

데이터 셋의 크기
Val 데이터셋의 구성

Unit 01 | ML 개요2

Data split

홀드아웃 방법

* 하이퍼 파라미터 : 모델의 특성을 결정, 사용자가 지정하는 옵션
ex) KNN의 k, Random Forest의 ntree, mtry

Train 데이터로 학습을 진행, Val 데이터로 하이퍼 파라미터 튜닝
가장 좋은 성능을 보인 모델을 Test 데이터로 검증 후 성능 평가

유의사항

홀드아웃 방법은 데이터를 위와 같이 나눈 방식 자체가 성능 추정에 민감한 영향을 미친다.
데이터셋이 충분히 크지 않거나 분할 시 데이터가 고루 퍼져있지 않으면 치명적인 문제 발생

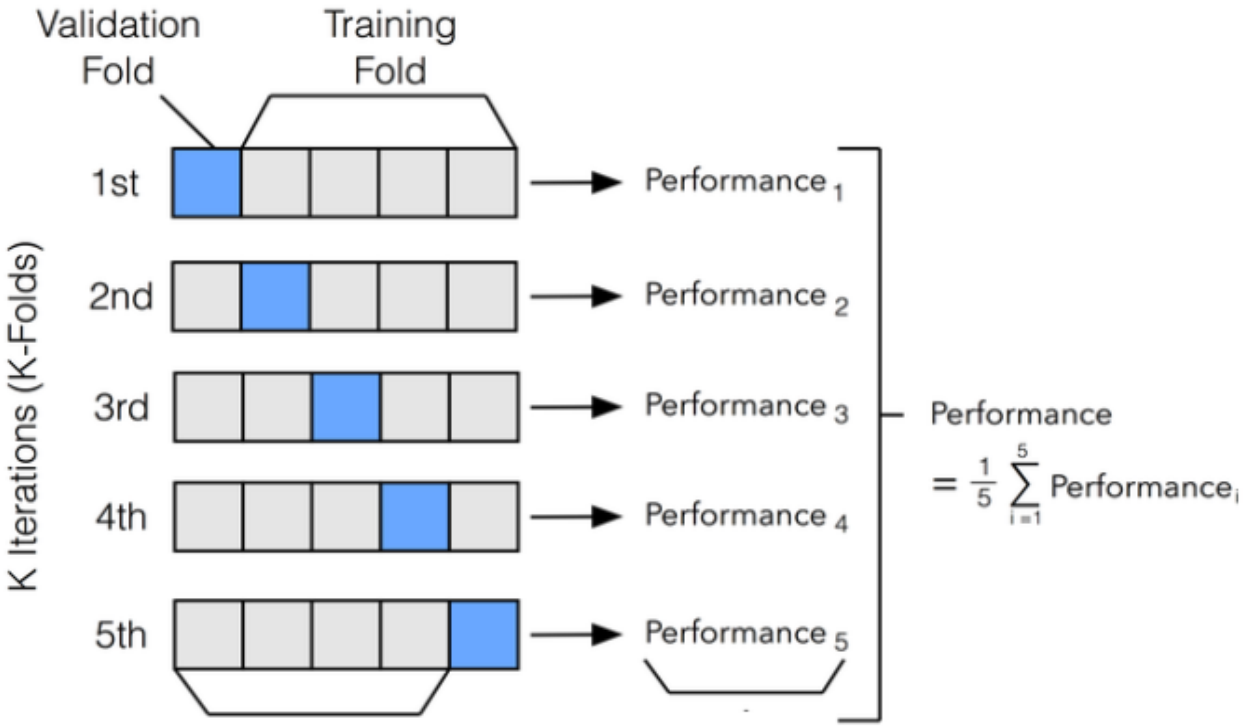
Unit 01 | ML 개요2

Data split

K-fold 교차검증 Cross validation

일반적으로 5, 10 사용

* LOOCV leave-one-out cross-validation
  : K -> N(데이터 개수)



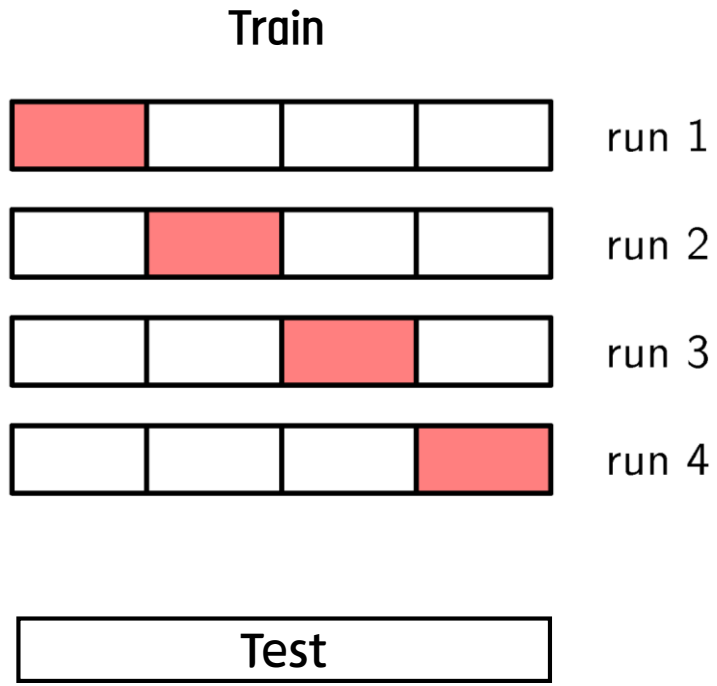
K-fold 교차검증

Ex) K = 4

1. 데이터를 4분할
2. 1번 데이터 셋이 val, 나머지 2~4번 데이터 셋으로 train
3. 2번 데이터 셋이 val, 나머지 1,3~4번 데이터 셋으로 train

...

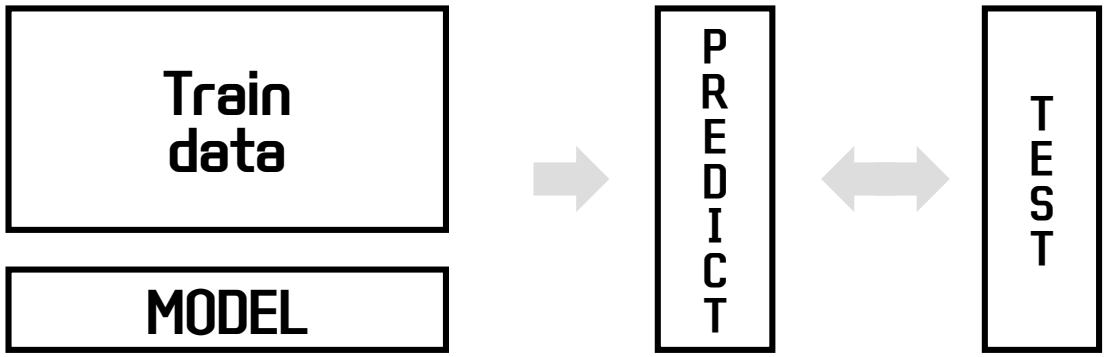
4. 각각의 과정에서 나온 로스들을 평균
5. 최적의 하이퍼 파라미터 탐색 후 test 데이터를 통해 성능 확인



위와 같은 과정을 통해 데이터의 특정 부분에에 편향되는 것을 방지 할 수 있다.

Unit 01 | ML 개요2

Evaluate



수치예측 = MSE, MAE, Huber loss 등

분류 = 정확도, Cross entropy(log loss) 등

Unit 01 | ML 개요2

Evaluate

MSE

$$MSE = \frac{1}{N}\sum_{i}(y_i - \hat{y}_i)^2$$

평균 제곱 오차(MSE), 가장 널리 쓰임
실제값(y)과 예측값(y hat)의 차이를 제곱한 다음 평균낸 것이다.
제곱을 한 이유는 오차(실제값-예측값)의 부호를 양수로 변환하기 위해 사용한다.

MAE

$$MAE = \frac{1}{N}\sum_{i}|y_i - \hat{y}_i|$$

평균 절대 오차(MAE)
부호를 양수로 바꿈에 있어 절대값을 사용한다.
MSE는 오차를 제곱하기 때문에 오차가 커지면 커질 수록 손실이 제곱으로 커진다.
따라서 100개중 99개가 잘맞아도 하나에서 크게 틀리면 영향을 많이 받는다.
MAE는 절대값을 취하기 때문에 상대적으로 이상점에 영향을 적게 받는다.

Huber loss

$$L_\delta(e) = \begin{cases} \frac{1}{2}e^2 & \text{for } |e| \le \delta, \\ \delta(|e| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

후버 로스
일정한 범위(델타)를 정해서 그 안에 있으면 오차를 제곱하고
밖에 있으면 절대값을 취한다.

편향-분산 Trade off

$$MSE = E[(y - \hat{y})^2] = Var(\hat{y}) + Bias(\hat{y})^2 + Var(\epsilon)$$

MSE는 위와 같이 다시 정의할 수 있다. Var(분산) + Bias(편향) + Var(노이즈)
노이즈는 데이터 자체가 가지는 태생적인 한계로 어떤 모델을 사용하건 줄일 수 없다.
반면 분산과 편향은 어떤 모델을 선택하느냐에 따라 줄일 수 있는 오차이다.

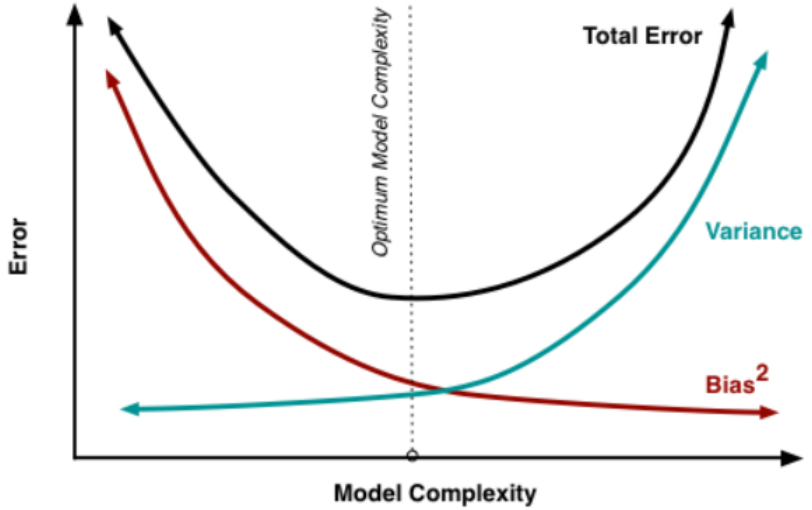
$$Var(\hat{y}) = E[(\hat{y} - E(\hat{y}))^2] \qquad Bias(\hat{y}) = E[(E(\hat{y}) - y)^2]$$

분산은 모델을 통해 예측한 값(y hat)이 그 예측값들의 평균(E(y hat))으로부터 얼만큼 퍼져있는지를 보여준다.
다시 말해 모델에 실제 데이터를 집어 넣었을 때 나오는 예측값이 얼마나 큰 변동성을 갖는지를 나타낸다.
한편 편향은 예측한 값의 평균(E(y hat))과 실제 값(y)의 차이로 모델이 맞추지 못하는 부분을 나타낸다.
즉 모델에 데이터를 넣었을 때 나오는 예측값이 전반적으로 얼마나 정확하게 예측하는지를 보여준다.

편향-분산 Trade off

일반적으로 복잡한 모델일 수록 비교적 예측값을 정확하게 추정하기 때문에 편향이 낮고 분산은 높다.
반면 단순선형모형과 같이 단순한 모델일 수록 편향은 높고 분산은 낮다.

MSE가 적은 모델을 만들기 위해서는 분산과 편향 모두 줄이는 것이 좋다.
그러나 모델의 복잡도 관점에서 봤을 때 둘은 상충관계에 놓인다.
복잡한 모델의 경우 분산이 높고 편향이 낮으며(=over fitting 가능성)
단순한 모델은 분산이 낮고 편향이 높아 (= under fitting 가능성)
둘 모두를 만족시킬 수 없다.
따라서 최선은 전체 오차(Total Error)를 최소화 하는 선에서
분산과 편향이 적절히 섞인 모델을 구축하는 것이다.



Q & A

KNN

Unit 02 | KNN

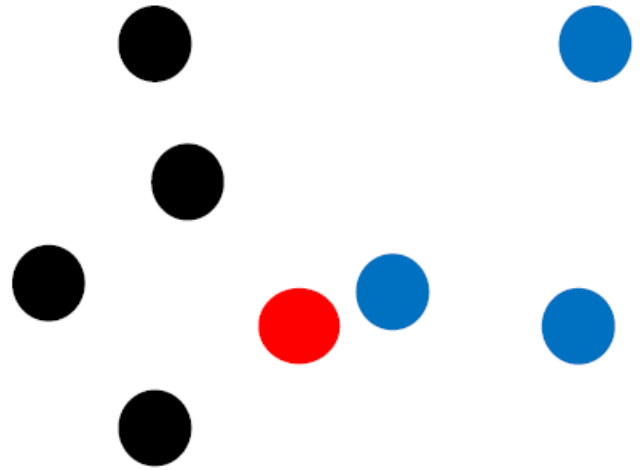
KNN

K-NN 이란?

K 개의

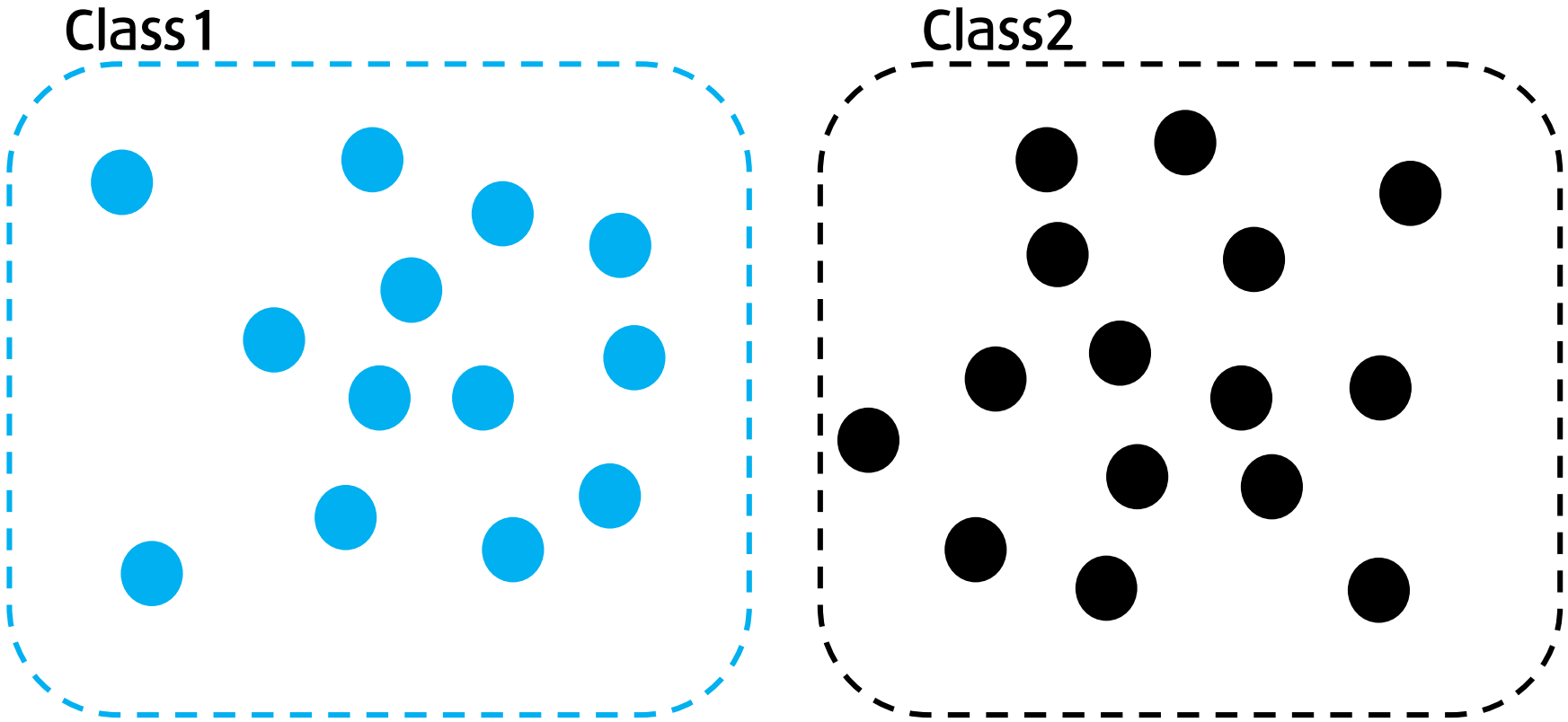
Nearest (가까운)

Neighbor (이웃)



Unit 02 | KNN

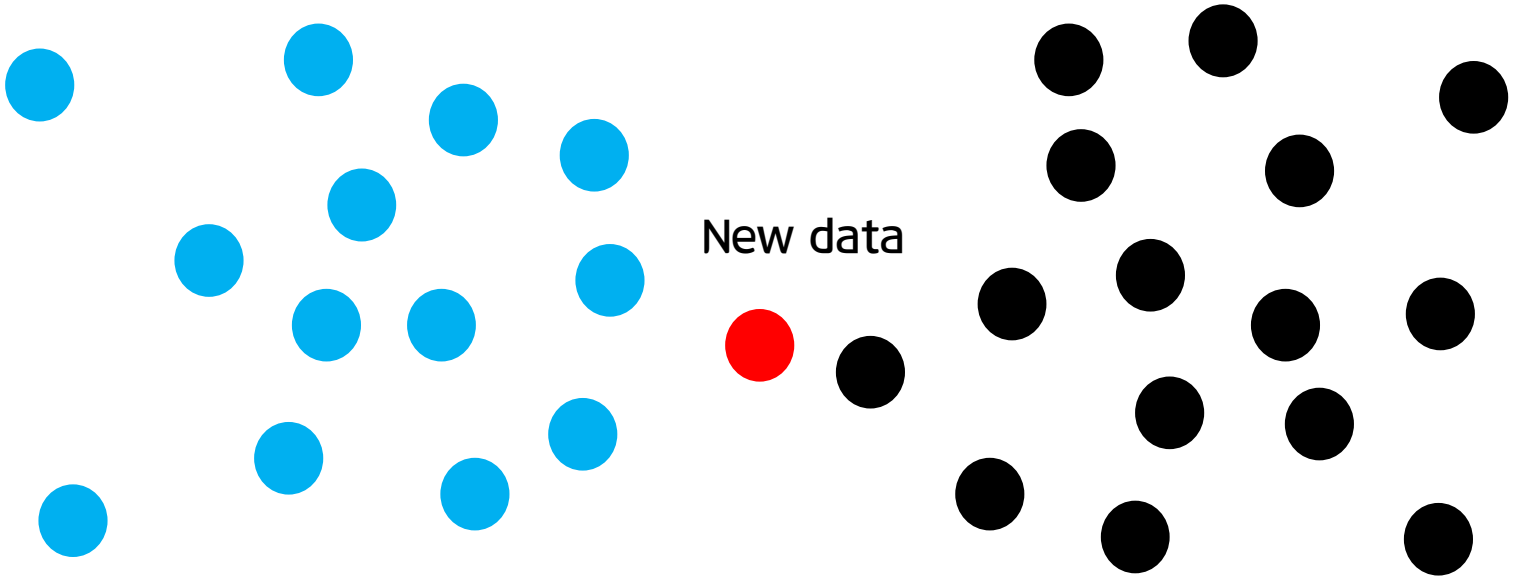
KNN



Unit 02 | KNN

KNN

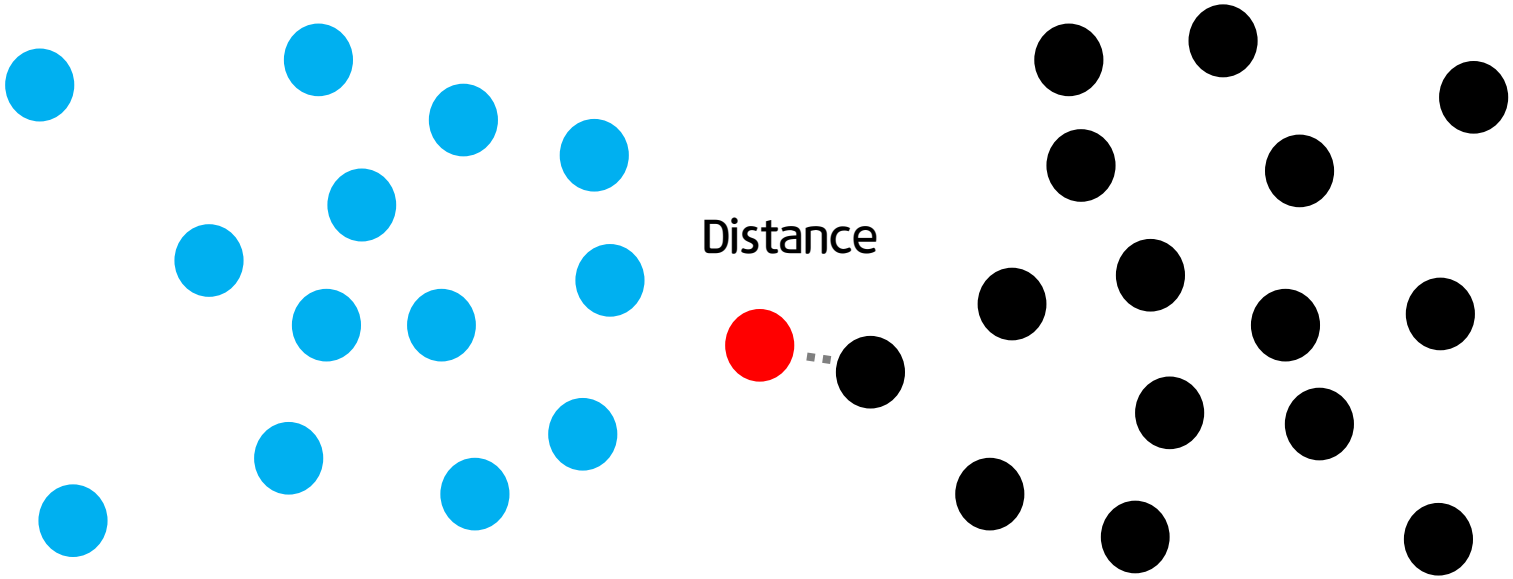
K = 1



Unit 02 | KNN

KNN

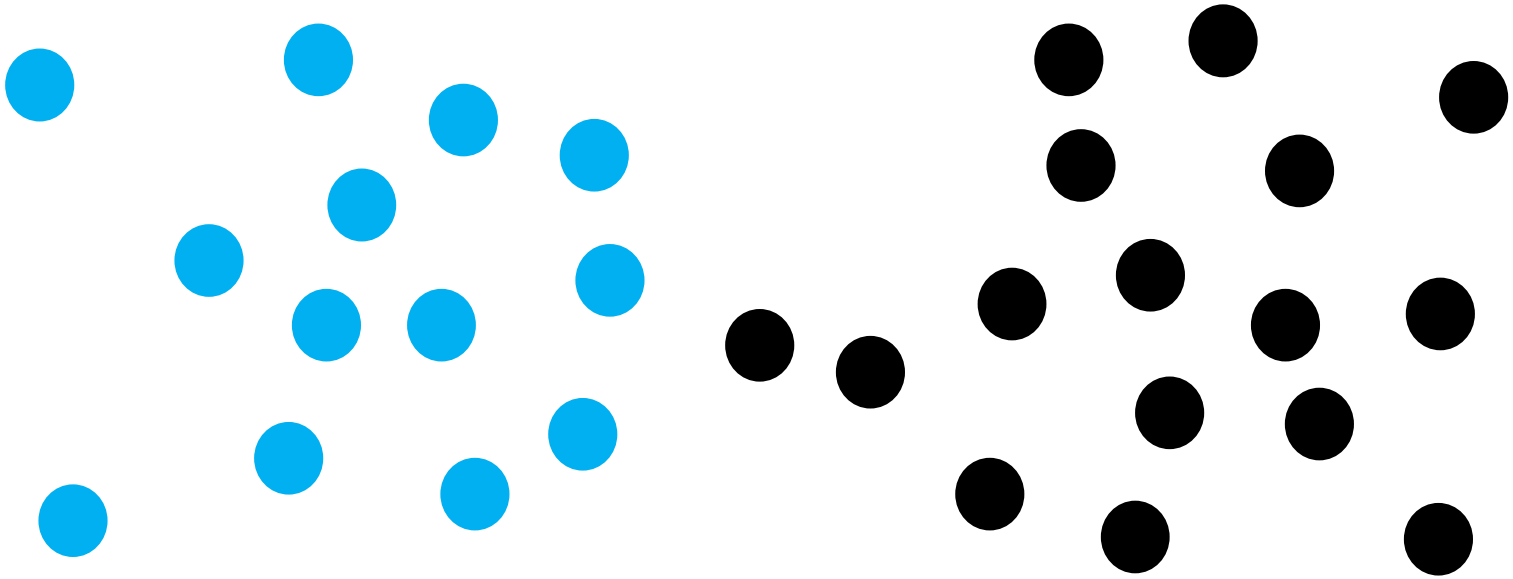
K = 1



Unit 02 | KNN

KNN

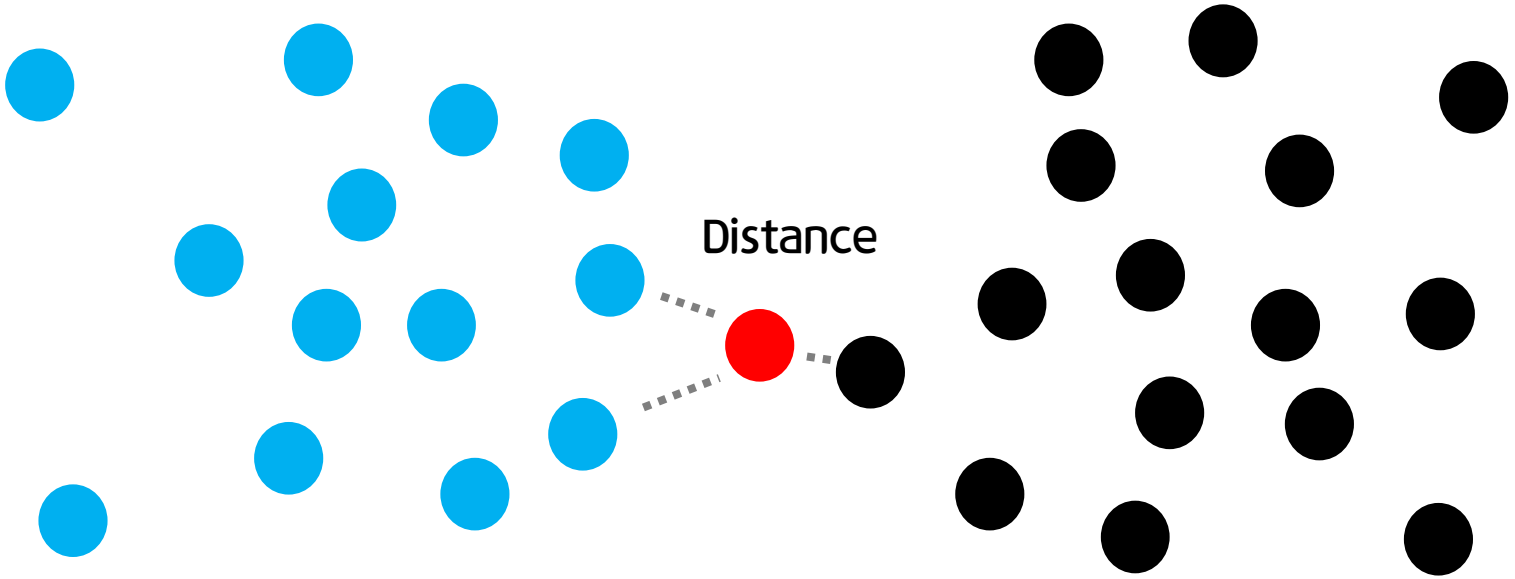
K = 1



Unit 02 | KNN

KNN

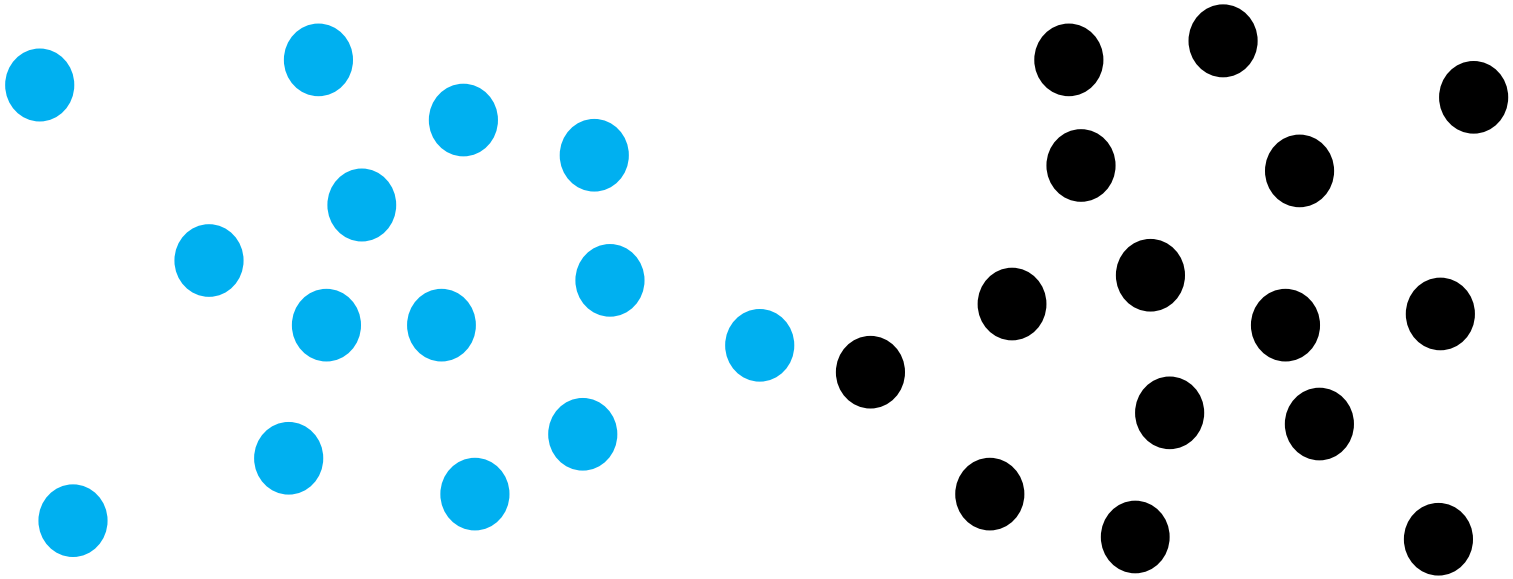
K = 3



Unit 02 | KNN

KNN

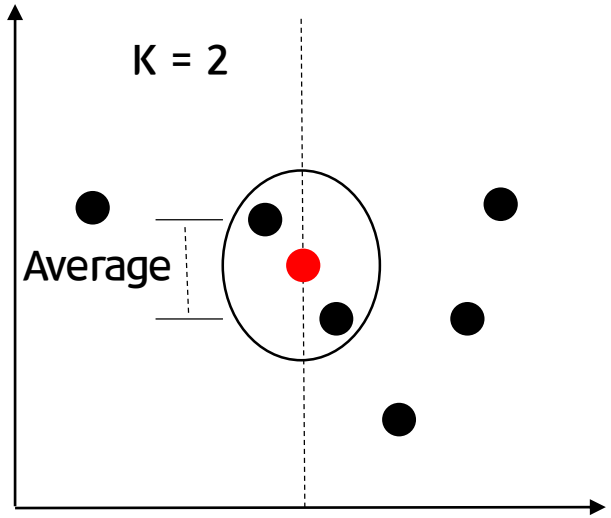
K = 3



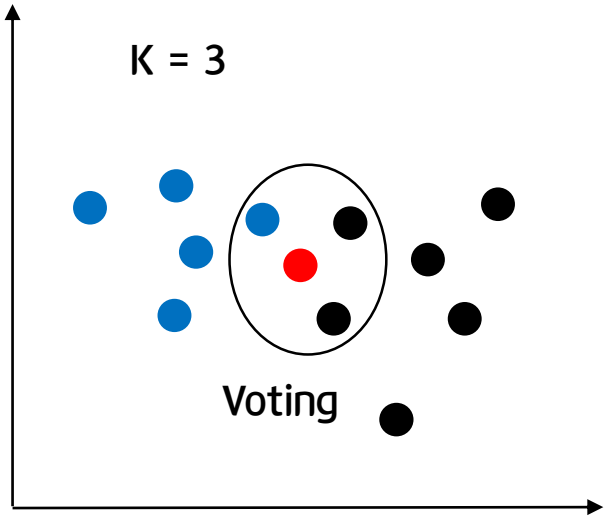
Unit 02 | KNN

KNN

KNN : 기억하고 있는 학습 데이터 중
k개의 가장 가까운 사례를 사용하여 **수치 예측** 및 **분류**



KNN Regressor



KNN Classifier

Unit 02 | KNN

Hyper parameter

KNN : 기억하고 있는 학습 데이터 중
k개의 가장 가까운 사례를 사용하여 수치 예측 및 분류

Hyperparameter

Distance(거리) & K

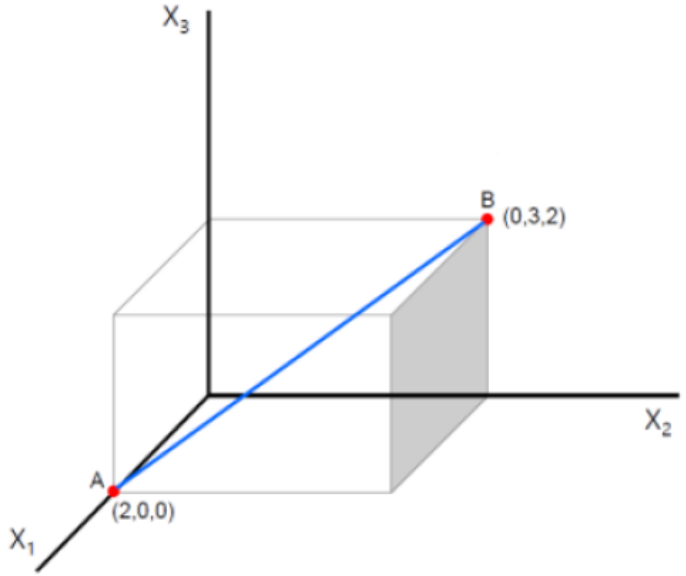
Unit 02 | KNN

Hyper parameter

Distance 를 구하는 방법 ?

Euclidean Distance

$$X = (x_1, x_2, \ldots, x_n)$$
$$Y = (y_1, y_2, \ldots, y_n)$$
$$d_{(X,Y)} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



$$d_{(A,B)} = \sqrt{(0-2)^2 + (3-0)^2 + (2-0)^2} = \sqrt{17}$$

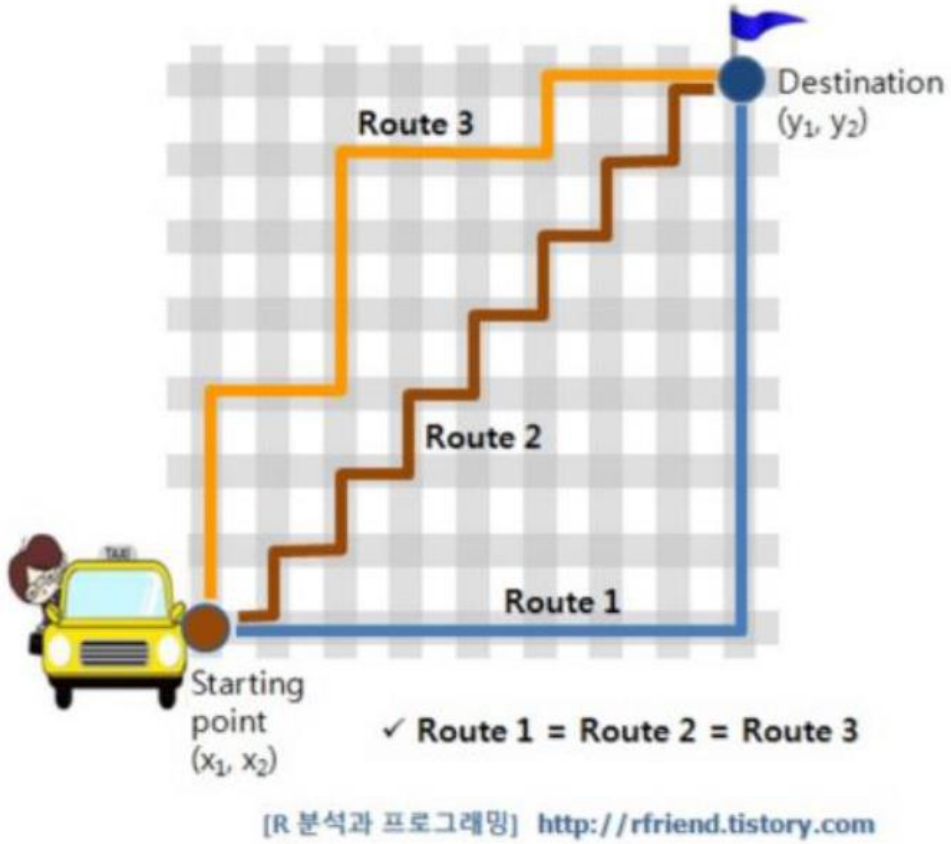
Unit 02 | KNN

Hyper parameter

Distance 를 구하는 방법 ?

Manhattan Distance

$$d_{Manhattan(X,Y)} = \sum_{i=1}^{n} |x_i - y_i|$$



Unit 02 | KNN

Hyper parameter

	Manhattan Distance	Euclidean Distance
k=1	78.42% (exponent=10)	81.86% (exponent=10)
k=3	86.00% (exponent=10)	86.57% (exponent=9)
k=5	86.42% (exponent=10)	86.57% (exponent=5)

TABLE II. WEIGHTED KNN CLASSIFICATION RESULTS

거리에 따라서 성능의 차이가 존재

Unit 02 | KNN

Hyper parameter

Distance를 구하는 방법?

Minkowski distance

$$X = (x_1, x_2, \ldots, x_n)$$

$$Y = (y_1, y_2, \ldots, y_n)$$

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

- p=1이면 맨하탄 거리
- p=2이면 유클리드 거리

Usage

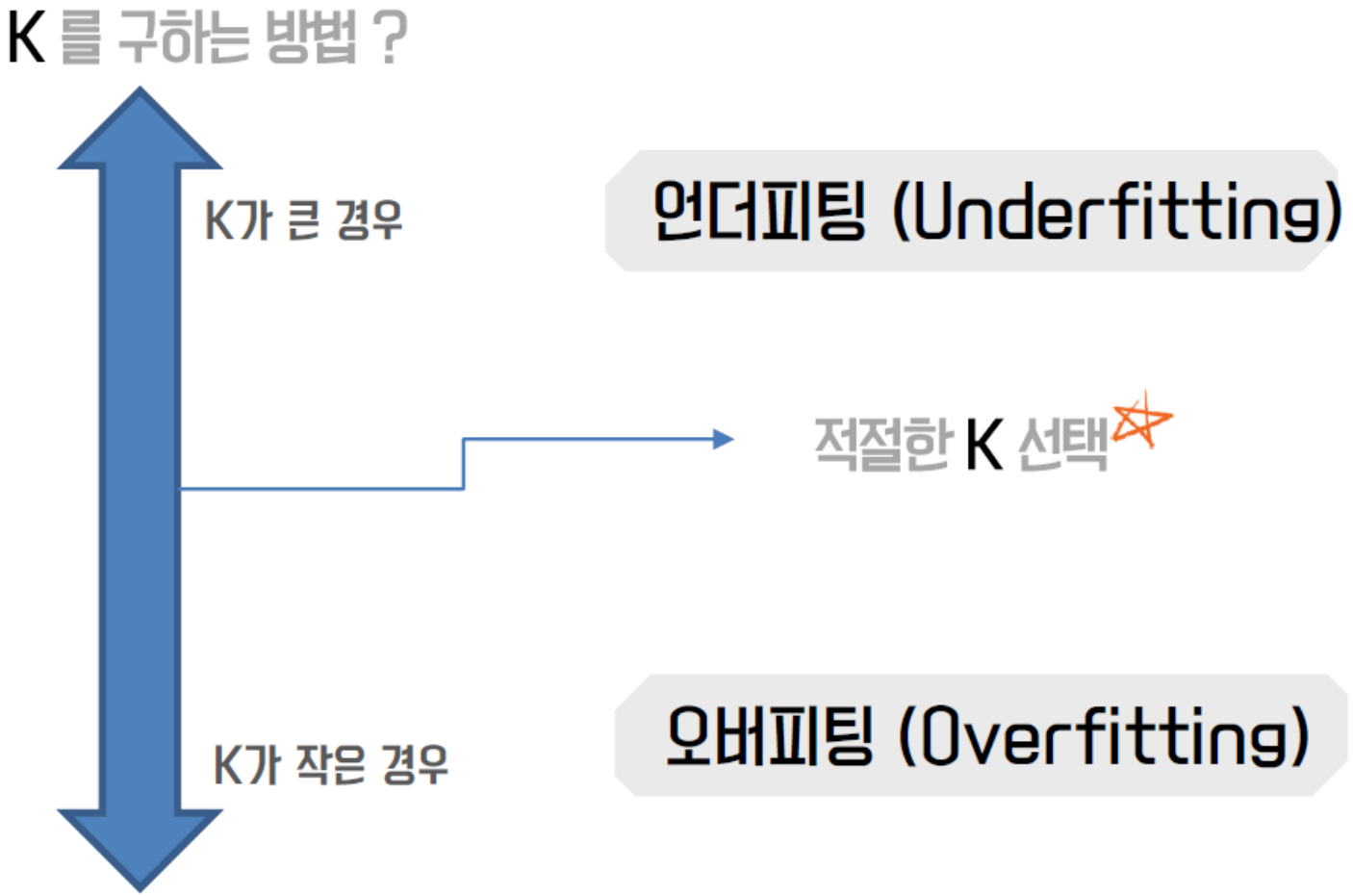
```
kknn(formula = formula(train), train, test, na.action = na.omit(),
        k = 7, distance = 2, kernel = "optimal", ykernel = NULL, scale=TRUE,
        contrasts = c('unordered' = "contr.dummy", ordered = "contr.ordinal"))
kknn.dist(learn, valid, k = 10, distance = 2)
```

Arguments

formula	A formula object.
train	Matrix or data frame of training set cases.
test	Matrix or data frame of test set cases.
learn	Matrix or data frame of training set cases.
valid	Matrix or data frame of test set cases.
na.action	A function which indicates what should happen when the data contain 'NA's.
k	Number of neighbors considered.
distance	Parameter of Minkowski distance.

Unit 02 | KNN

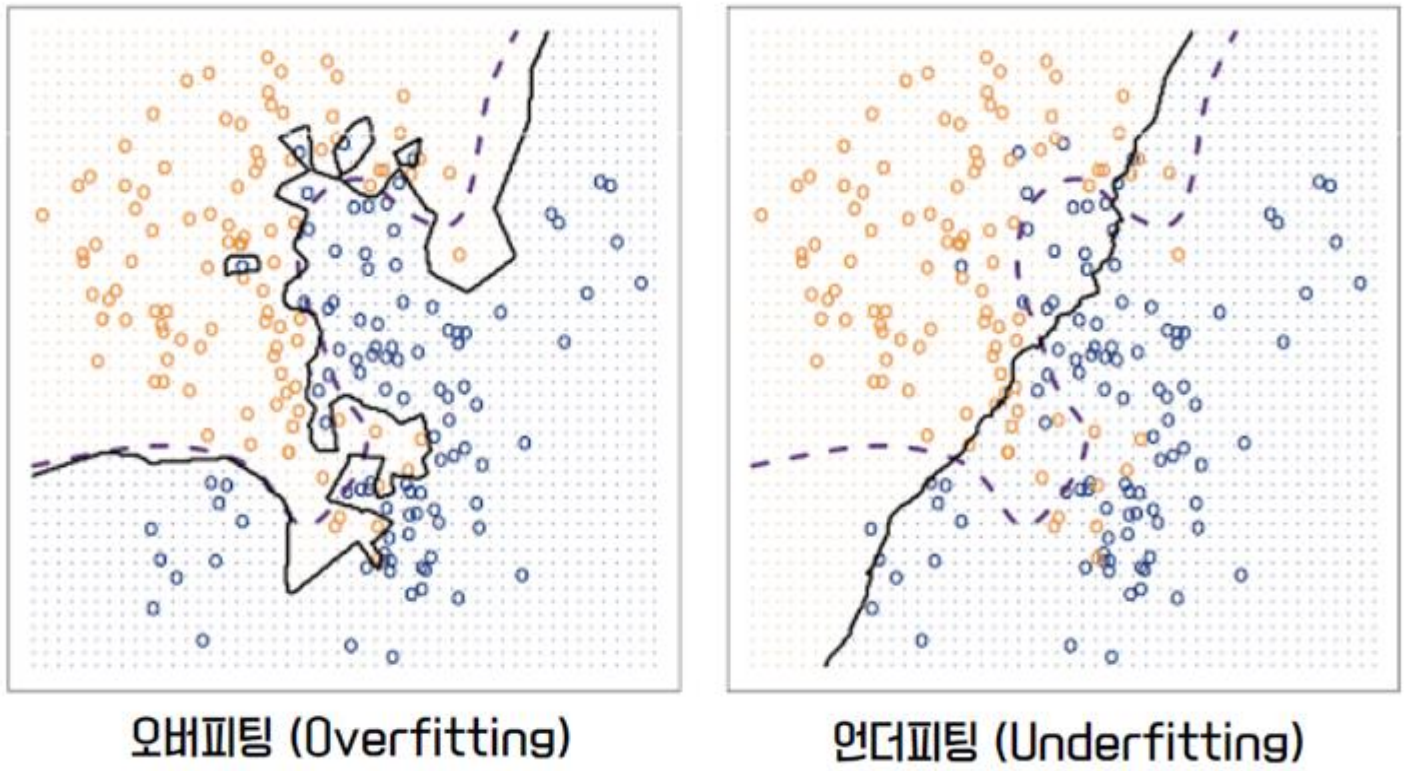
Hyper parameter



Unit 02 | KNN

Hyper parameter

K를 구하는 방법 ?



KNN 고려사항

1. Distance기반 알고리즘
   - 변수들의 단위에 민감

2. 단순 평균, Voting으로 결정하지 않으려면?

Unit 02 | KNN

KNN

1. 같은 의미 다른 결과

X1	X2($)
1	5
2	6
4	4

$$Distance(1,3) = \sqrt{(1-4)^2 + (5-4)^2}$$
$$= 3.162278$$

$$Distance(2,3) = \sqrt{(2-4)^2 + (6-4)^2}$$
$$= 2.828427$$

X1	X2(₩)
1	5000
2	6000
4	4000

$$Distance(1,3) = \sqrt{(1-4)^2 + (5000-4000)^2}$$
$$= 1000.004$$

$$Distance(2,3) = \sqrt{(2-4)^2 + (6000-4000)^2}$$
$$= 2000.001$$

1. 특정 변수에 집중

이웃	특성1	특성2	특성3
N1	0.8	400	0.5
N2	12	134,000	0.9
N3	0	20,000	1.1
N4	67	32,000	0.1

$$\text{distance}(N3, N4) = \sqrt{(0-67)^2 + (20{,}000-32{,}000)^2 + (1.1-0.1)^2}$$

$$= \sqrt{(4489 + 144{,}000 + 1.0)}$$

1. Feature scaling

변수들의 단위를 맞추어 준다!

Min-max normalization

$$X_{new} = \frac{X - min(X)}{max(X) - min(X)}$$

Z-score normalization

$$X_{new} = \frac{X - \mu}{\sigma}$$

Weighted K-NN

단순히 평균, 다수결로 값을 결정하지 않고
거리에 따라서 영향력을 달리 주고 싶을 때 사용

$$\text{유사도} = \frac{1}{\text{거리}} \qquad \text{가중치} = \frac{\text{유사도}}{\text{모든 이웃의 유사도 합}}$$

W-KNN

2. Classification

$$\text{유사도} = \frac{1}{\text{거리}} \qquad \text{가중치} = \frac{\text{유사도}}{\text{모든 이웃의 유사도 합}}$$

New Data
(K=4)

클래스	이웃	특성1	특성2	거리	유사도	가중치
A	N1	0.012	0	1	1	0.48
B	N2	0.179	1	2	0.5	0.24
C	N3	0	0.147	3	0.33	0.16
B	N4	1	0.237	4	0.25	0.12

KNN

A:1, B:2, C:1

B로 분류

W-KNN

A:1*0.48, B:1*0.24+1*0.12, C:1*0.16

A로 분류

2. Regression

$$\text{유사도} = \frac{1}{\text{거리}} \qquad \text{가중치} = \frac{\text{유사도}}{\text{모든 이웃의 유사도 합}}$$

New Data
(K=4)

이웃	체지방률	거리	유사도	가중치
N1	15.4	1	1	0.48
N2	17.2	2	0.5	0.24
N3	12.3	3	0.33	0.16
N4	11.5	4	0.25	0.12

KNN

= (15.4+17.2+12.3+11.5)/4

= 14.1

W-KNN

= (15.4*0.48+17.2*0.24+12.3*0.16+11.5*0.12)

= 14.868

Unit 02 | KNN

KNN 장점

1. 학습 과정이 없다

2. 직관적인 이해 용이

3. 데이터가 충분히 많으면 좋은 성능을 보인다

Unit 02 | KNN

KNN 단점

1. 메모리가 많이 필요하다

2. 지나치게 기존 사례에 의존적

3. 데이터가 많을 수록 검색 시간이 길어진다

4. 회귀의 경우 주변부에서 왜곡 현상이 나타날 수 있다

Unit 02 | KNN

KNN 단점

주변부에서 왜곡

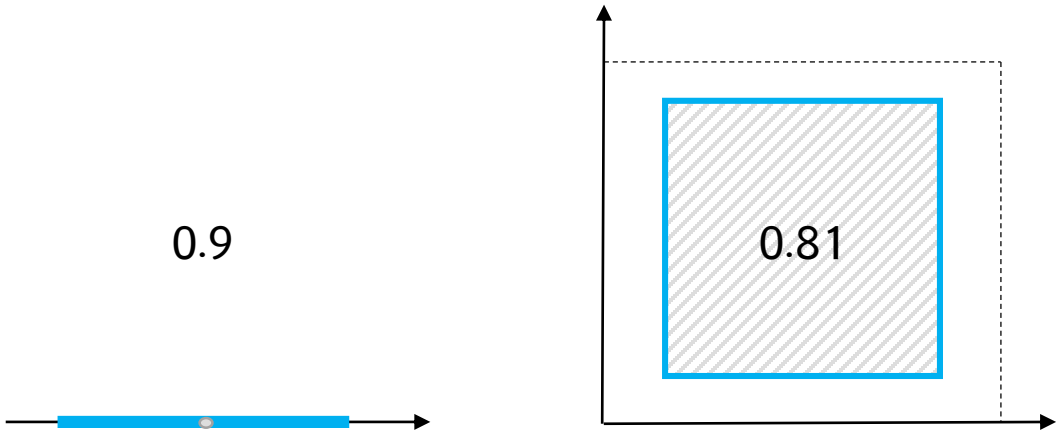
- 기존 사례의 최대치를 벗어나는 새로운 사례 : 과소 예측
- 기존 사례의 최소치를 벗어나는 새로운 사례 : 과대 예측

Unit 02 | KNN

KNN 단점

차원의 저주

*중심부로부터 90%이내의 데이터 비율



KNN 단점

차원의 저주

중심부로부터의 거리는 전체의 90%로 모두 동일하나 데이터는 희박

주변부에서 왜곡이 일어날 수 있는 KNN에 치명적

-> PCA 등 차원 축소의 개념을 통해 해결 가능!

Unit 02 | KNN

정리

하이퍼 파라미터

회귀

차원의 저주

가중치

KNN

분류

학습

주변부에서 왜곡

정규화

Unit 02 | KNN

정리

KNN은 학습이 일어나지 않아 lazy algorithm이라고도 한다.
새로운 사례가 들어오면 기존 사례와 가장 근접한 사례를 통해 예측(평균) 및 분류(다수결)를 진행한다.
대표적 하이퍼 파라미터로는 Distance와 K가 있으며
거리 기반 알고리즘이기 때문에 변수들의 단위가 다를 경우 정규화를 시켜주어야 한다.
분류와 예측에서 거리에 따른 영향력을 달리 주고 싶은 경우 가중치를 부여한 w-knn을 사용한다.

데이터가 충분히 많은 경우 좋은 성능을 보이는 장점이 있으나
그에 비례하여 참조 시간, 메모리 사용량이 증가하는 단점이 있다.
또한 Knn은 수치 예측의 경우 주변부에서 왜곡이 일어나는 단점이 존재한다.
이는 변수가 많은 경우 차원의 저주와 맞물려 더욱 심한 문제를 야기시킨다.
pca등 차원 축소 방법으로 해결할 수 있다.

Unit 02 | KNN

참고

- 교차검증, 하이퍼파라미터
- http://yamalab.tistory.com/44
- https://www.slideshare.net/RickyPark3/5model-evaluation-and-improvement
- https://tensorflow.blog/%EB%A8%B8%EC%8B%A0-%EB%9F%AC%EB%8B%9D%EC%9D%98-%EB%AA%A8%EB%8D%B8-%ED%8F%89%EA%B0%80%EC%99%80-%EB%AA%A8%EB%8D%B8-%EC%84%A0%ED%83%9D-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-%EC%84%A0%ED%83%9D-3/

- **맨하탄, 유클리드거리**
- **http://rfriend.tistory.com/tag/%EB%A7%A8%ED%95%98%ED%83%84%20%EA%B1%B0%EB%A6%AC**
- **https://redtea.kr/pb/pb.php?id=free&no=6926**

- **Knn**
- **http://horae.tistory.com/entry/KNN-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98-1**
- **http://www.datamarket.kr/xe/board_mXVL91/9153 - 투빅스 1기 김보섭**
- **https://www.semanticscholar.org/paper/A-Video-Semantic-Analysis-Method-Based-on-Kernel-Dai-Zhan/1cbf40e492ac9fa389ddcb78b7f362dd30cf4408 - A Video Semantic Analysis Method Based on Kernel Discriminative Sparse Representation and Weighted KNN,**

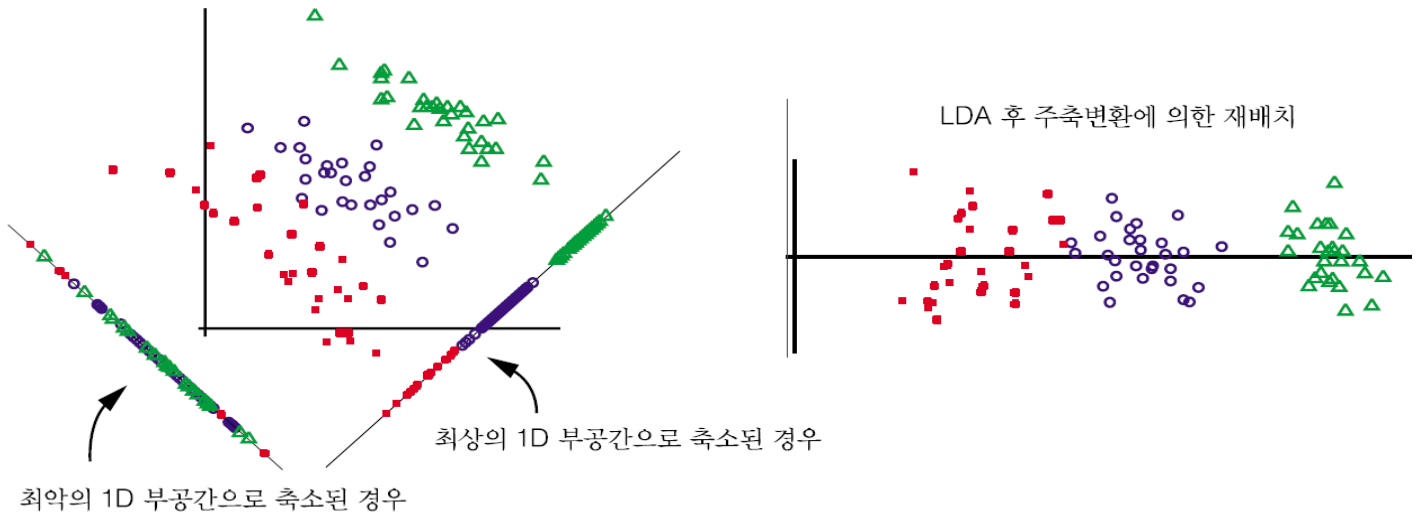
Q & A

LDA

Unit 03 | LDA

LDA

Is LDA a dimensionality reduction technique or a classifier algorithm?



[그림 11-2] LDA에 의한 차원 축소

LDA 차원축소

주성분 분석법(PCA)은 데이터의 최적 표현의 견지에서 데이터를 축소하는 방법인데 반하여

선형판별 분석법(LDA)은 데이터의 최적 분류의 견지에서 데이터를 축소하는 방법이라고 할 수 있다.

➔ (목적) 가능한 클래스간의 분별 정보를 최대한 유지시키면서 차원을 축소시키는 것

해당 부분은 PCA가 끝나고 자료를 통해서 제공!

Unit 03 | LDA

선형 분류

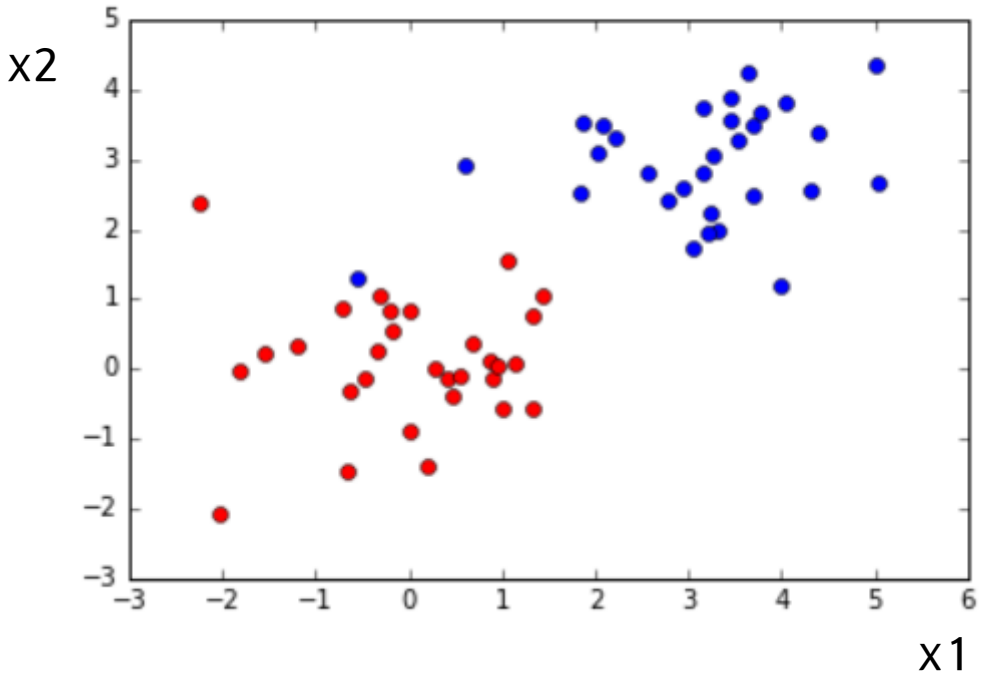
최근접 이웃의 단점과 장점 모두 기존의 사례들을 직접 활용함에서 나옴

⇒ 분류 경계선을 학습하는 방법

선형 분류 : y = 0이 가장 적절한 경계선이 되도록 w와 b를 결정

Unit 03 | LDA

선형 분류

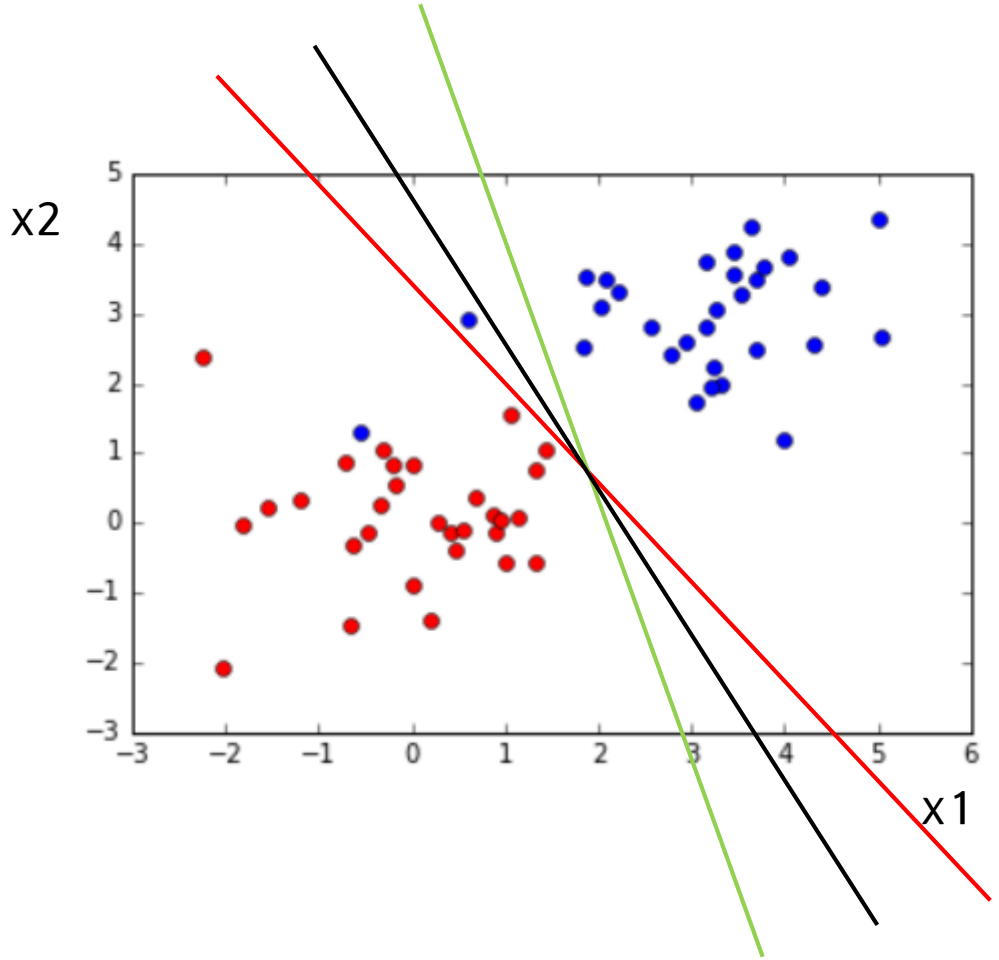


분류 경계선이 직선이라고 가정

$$y = w_1 x_1 + w_2 x_2 + b$$

Unit 03 | LDA

선형 분류

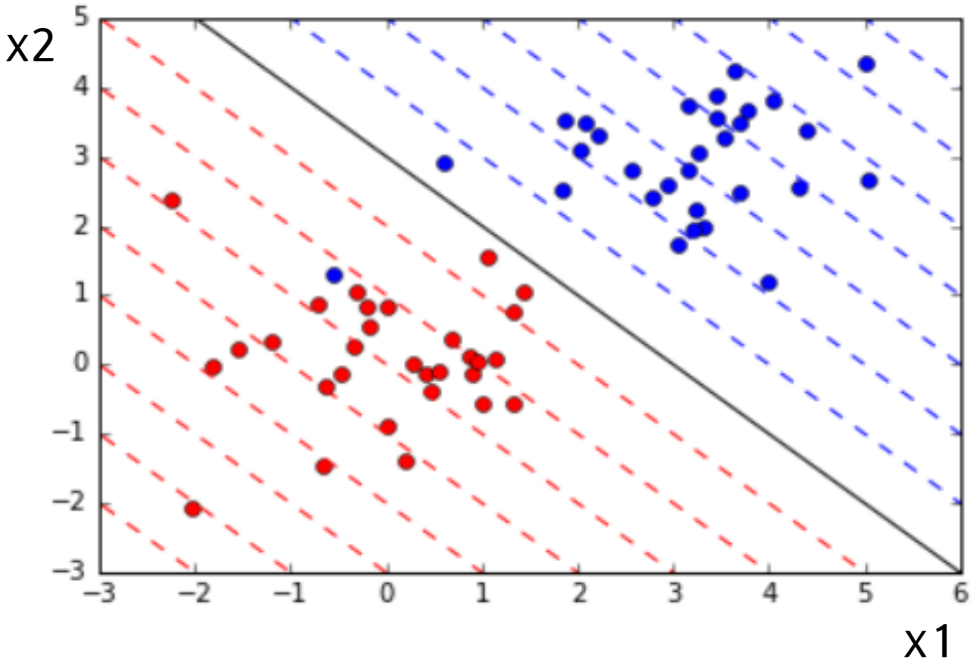


분류 경계선이 직선이라고 가정

$$y = w_1 x_1 + w_2 x_2 + b$$

Unit 03 | LDA

선형 분류



* 손실함수를 최적화하는 과정을 통해 결정

해당 data에 가장 알맞은 w가

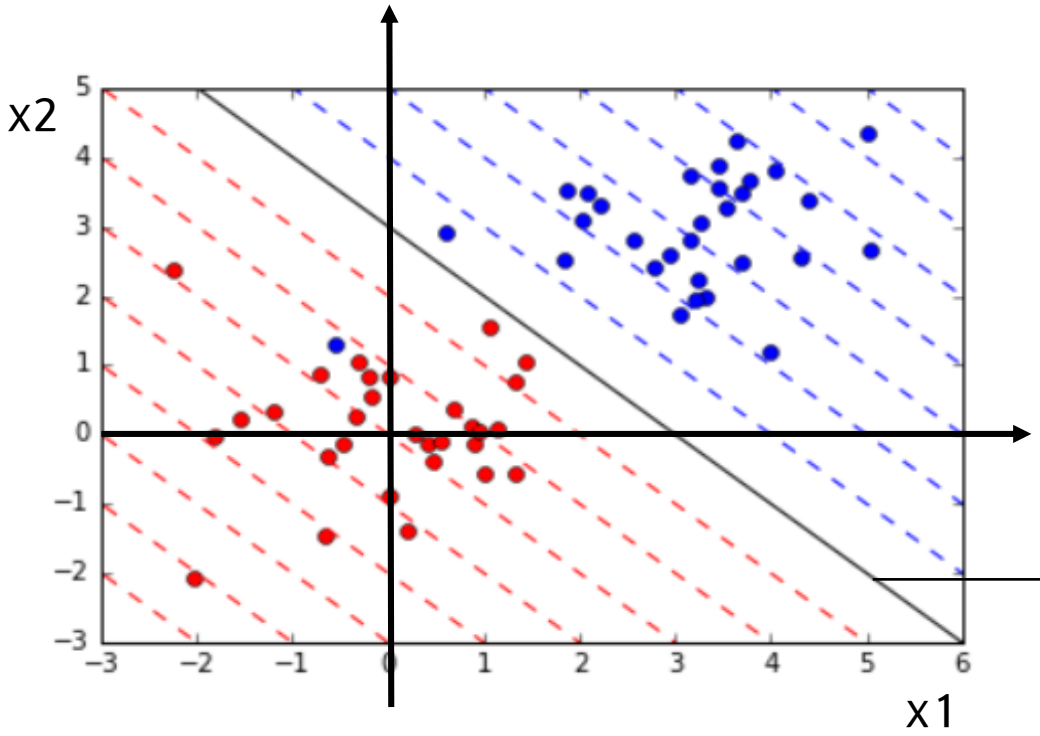
$w_1 = w_2 = 1$ 이면

$$y = x_1 + x_2 + b$$

절편 b에 따라서 다양하게 분류 경계선이 생김

Unit 03 | LDA

선형 분류



해당 data에 가장 알맞은 w가

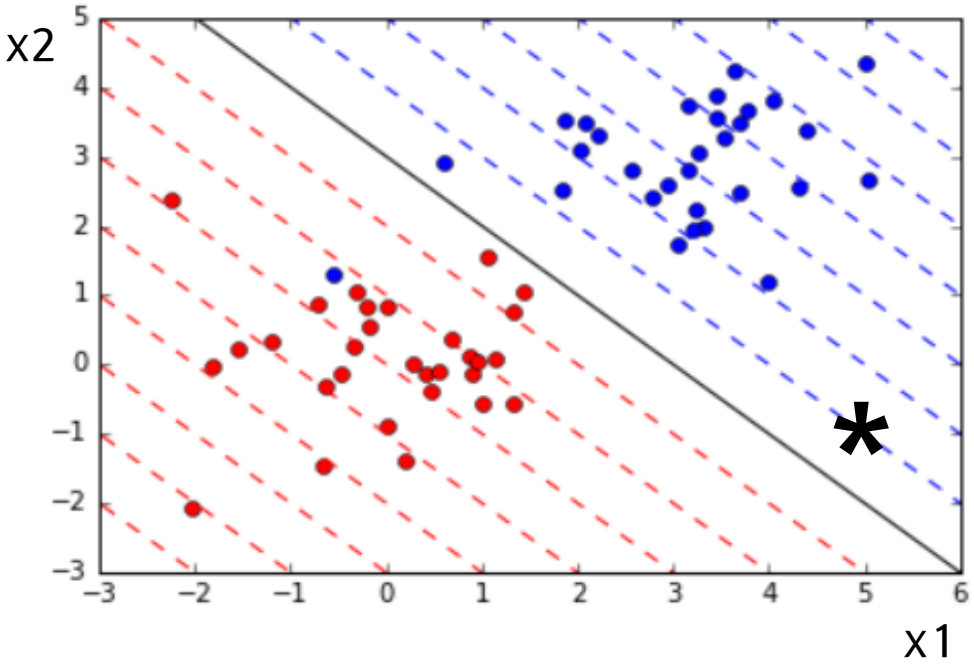
$w_1 = w_2 = 1$ 이면

$$y = x_1 + x_2 + b$$

이 실선이 최적! 그때의 x1+x2 = 3
Y = 3+b
따라서 b = -3

Unit 03 | LDA

선형 분류



새로운 데이터 Z(x1,x2)가 들어왔을 때
구축된 식 **y = x1+x2-3**에 대입을 하여
양수이면 파란색, 음수이면 빨간색으로 분류!

Ex) z(5,-1) -> 5-1-3 = 1
양수 = 파란색 class

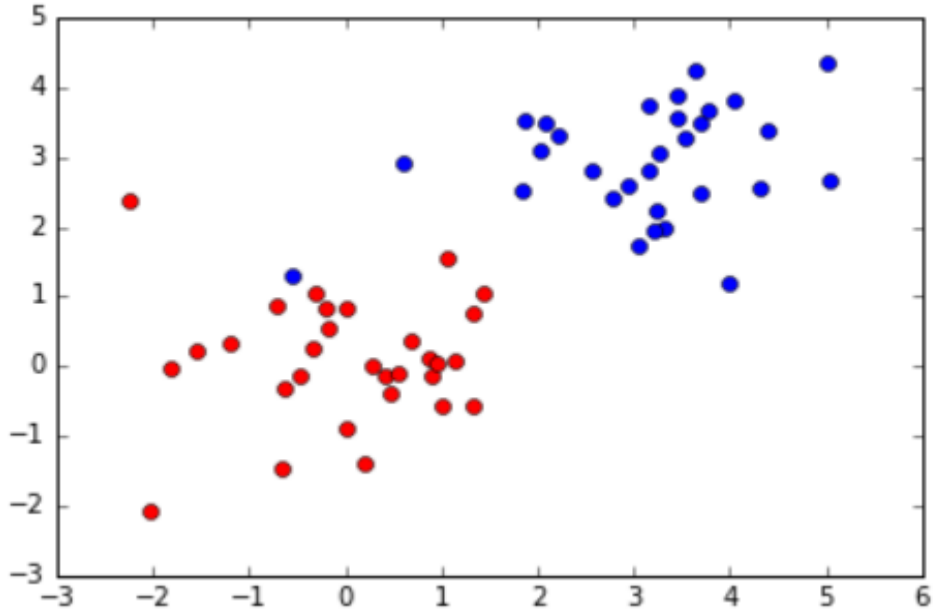
Unit 03 | LDA

LDA

선형 분류기 중 하나인 LDA!

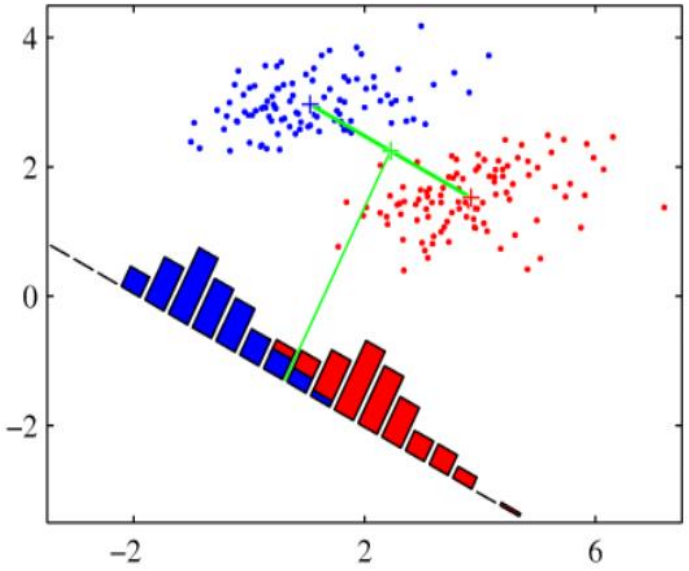
두집단의 평균(=중심점)을 멀게 만드는
W가 좋은 파라미터이다!

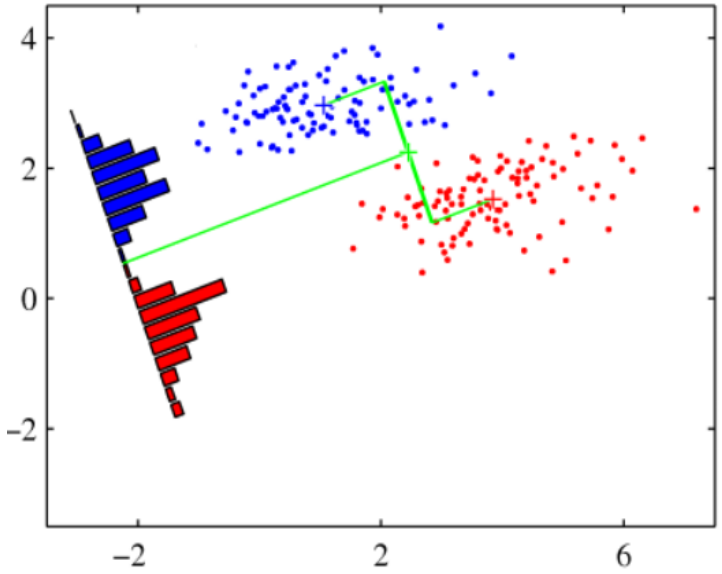
= 클래스 간의 분산을 크게



Unit 03 | LDA

LDA

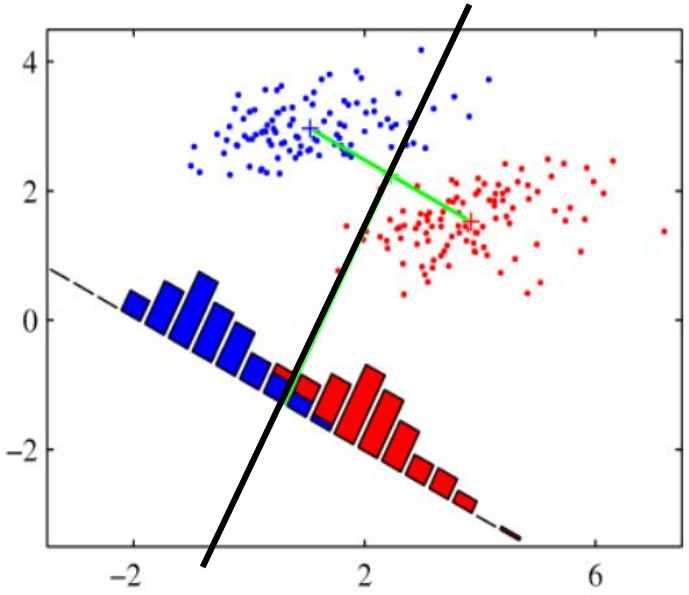




Unit 03 | LDA

LDA

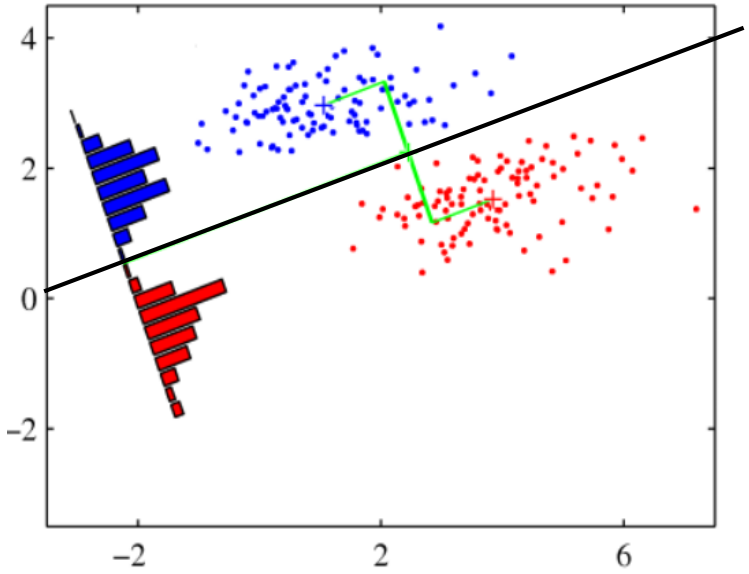
Decision boundary



중심점간의 거리

>

Decision boundary

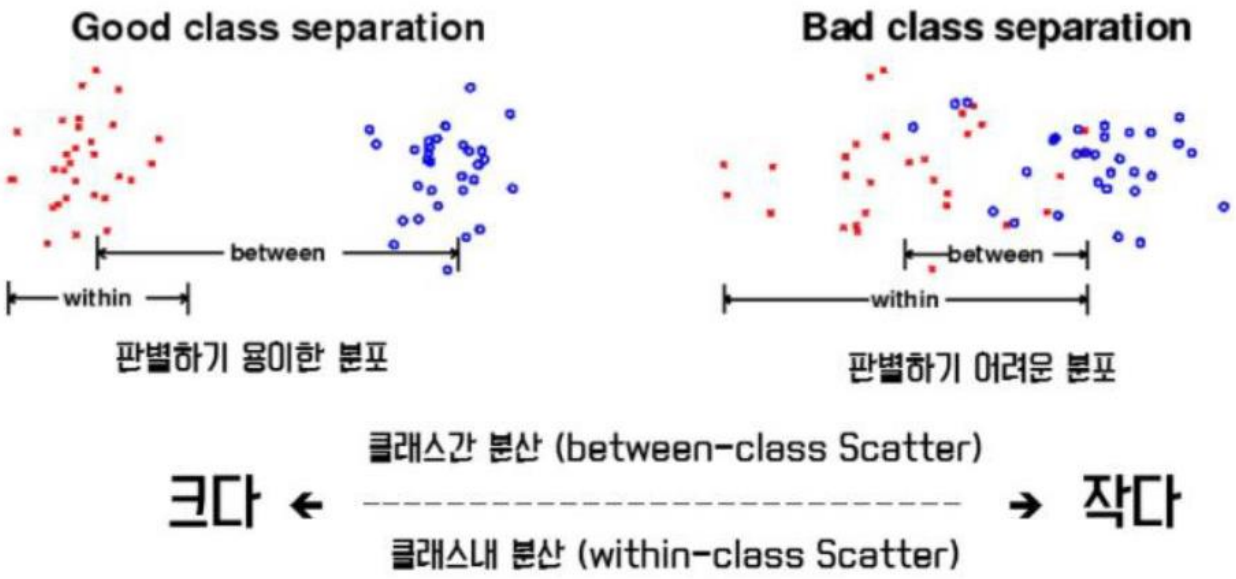


단순히 두 클래스의 중심점을 멀게
만드는 것만으론 충분하지 않다!

각 클래스 안에서의 분산 또한 고려!

Unit 03 | LDA

LDA



클래스 간의 분산은 크게
클래스 내의 분산은 작게

$$J = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

LDA의 목적함수!

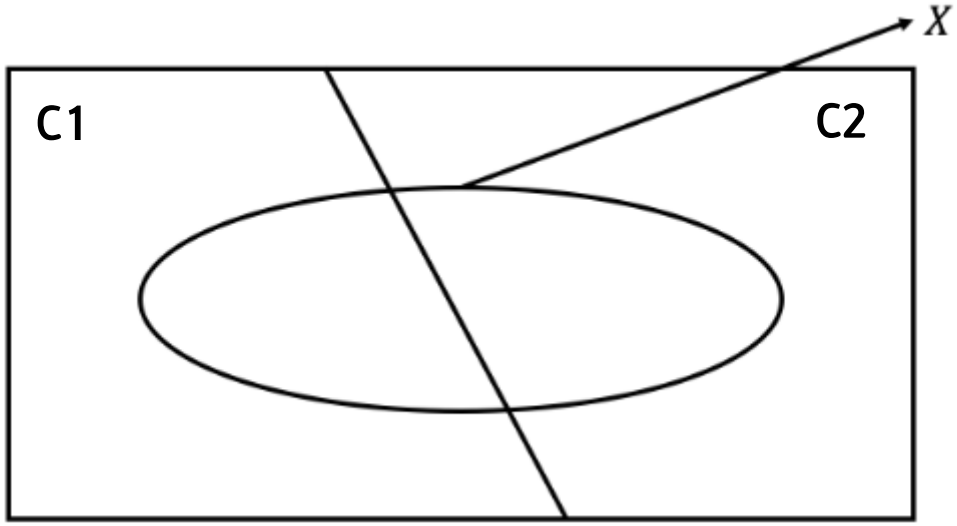
*자세한 내용은 PCA이후 자료를 통해 제공

Unit 03 | LDA

LDA

작동원리를 확률 모형으로부터 도출해보자!

클래스 = C1, C2



LDA assumption

가정1. 데이터가 다변량 정규분포를 따른다.

가정2. 범주 각각의 분산이 동일하다.

Unit 03 | LDA

LDA

LDA는 새로운 데이터(x)가 들어오면

C1일 확률과 = $P(C_1|x)$
C2일 확률을 = $P(C_2|x)$

구하고 큰 쪽으로 분류 진행!

사후확률 (=예측할 때) $P(C_i|x)$ = 우도 (=학습할 때) $P(x|C_i)$ 사전확률 $P(C_i)$ / $P(x)$

0. 사후확률(=판별함수) P(Ci|x)

새로운 데이터가 주어졌을 때(=정답을 모를 때) 특정 범주에 들어갈 확률

=> 범주 분류를 위한 확률(스코어)를 내주는 함수를 판별함수(discriminant function)라고 한다.

판별함수= $\sigma_i(x) = P(C_i|x)$
*범주의 개수만큼 필요

$$\propto P(x|C_i)P(C_i)$$
$$\propto \ln P(x|C_i) + \ln P(C_i)$$

상대적인 크기만 필요,
계산의 편의를 위해 식 변형

1. 우도 $P(x|C_i)$

범주정보가 주어졌을 때(=정답을 알 때) x가 나타날 확률
=> 학습 데이터 내에 있는 C_i 들의 분포

정규분포 확률함수와
정규분포를 따르는 데이터의 평균과 분산
두가지 파라미터로 우도를 구할 수 있다.

ex) μ_1 = 첫번째 변수의 평균,
σ_{12} = 첫번째 변수와 두번째 변수의 공분산

$$\mu = \begin{bmatrix} \mu_1 \\ \cdots \\ \mu_p \end{bmatrix} \quad P$$

$$\Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1p} \\ \cdots & \cdots & \cdots \\ \sigma_{p1} & \cdots & \sigma_{pp} \end{bmatrix} \quad P$$

P = 변수 개수

Unit 03 | LDA

LDA

우도= $P(x|\,\mathrm{Ci}\,) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i)\right]$ ← 다변량 정규분포 식

위에서 변형한 판별함수에 대입

$$
\begin{aligned}
ln P(x|\,\mathrm{Ci}\,) &= \ln\frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i) \\
&= \ln\frac{1}{(2\pi)^{p/2}} + \ln\frac{1}{|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i) \\
&= -\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i)
\end{aligned}
$$

Unit 03 | LDA

LDA

우도= $P(x|\,\mathrm{Ci}\,) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i)\right]$ ← 다변량 정규분포 식

위에서 변형한 판별함수에 대입

$$\begin{aligned}\sigma_i(x) &= P(W_i|x)\\ &\propto P(x|W_i)P(W_i)\\ &\propto \ln P(x|W_i) + \ln P(W_i)\end{aligned}$$

$$\begin{aligned}lnP(x|\,\mathrm{Ci}\,) &= \ln\frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i)\\ &= \ln\frac{1}{(2\pi)^{p/2}} + \ln\frac{1}{|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i)\\ &= -\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x-\mu_i)^T\Sigma_i{}^{-1}(x-\mu_i)\end{aligned}$$

미지수가 하나도 없는 상수!

Unit 03 | LDA

LDA

우도= $P(x|\,\mathrm{Ci}\,) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right]$ ← 다변량 정규분포 식

위에서 변형한 판별함수에 대입

$$
\begin{aligned}
ln P(x|\,\mathrm{Ci}\,) &= \ln\frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \\
&= \ln\frac{1}{(2\pi)^{p/2}} + \ln\frac{1}{|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \\
&= -\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)
\end{aligned}
$$

변수 개수

분산 평균 데이터

미지수가 하나도 없는 상수!

Unit 03 | LDA

LDA

우도= $P(x|C_i) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} exp\left[-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right]$ ← 다변량 정규분포 식

위에서 변형한 판별함수에 대입

$$
\begin{aligned}
ln P(x|C_i) &= \ln \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \\
&= \ln \frac{1}{(2\pi)^{p/2}} + \ln \frac{1}{|\Sigma_i|^{1/2}} - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \\
&= -\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)
\end{aligned}
$$

변수 개수

분산 평균 데이터

= 값을 대입해서 P(x|Ci)를 구할 수 있다

2. 사전확률 $P(C_i)$

해당 범주의 데이터 개수를 전체 데이터 개수로 나눈다

사전확률

$$P(C_i) = \frac{C_i \text{ 데이터 개수 } (i=1,2)}{\text{전체 데이터 개수}}$$

3. 분류 경계 Decision boundary

$= \sigma_1(x)$ $= \sigma_2(x)$

범주가 2개인 경우, $P(C_1|x) = P(C_2|x)$인 지점이 바로 분류 경계

3. 분류 경계 Decision boundary

$= \sigma_1(x)$ $= \sigma_2(x)$

범주가 2개인 경우, $P(C_1|x) = P(C_2|x)$인 지점이 바로 분류 경계

$$\sigma_1(x) - \sigma_2(x) = 0$$

$$-\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1) + \ln P(C_1)$$

$$-(-\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_2| - \frac{1}{2}(x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2) + \ln P(C_2)) = 0$$

$$(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1) - (x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2) - \ln\frac{|\Sigma_2|}{|\Sigma_1|} + 2\ln\frac{P(C_1)}{P(C_2)} = 0$$

3. 분류 경계 Decision boundary

$= \sigma_1(x)$ $= \sigma_2(x)$

범주가 2개인 경우, P(C1|x) = P(C2|x)인 지점이 바로 분류 경계

$$\sigma_1(x) - \sigma_2(x) = 0$$

$$-\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1) + \ln P(C_1)$$

우도 ← → 사전확률

$$-(-\frac{p}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_2| - \frac{1}{2}(x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2) + \ln P(C_2)) = 0$$

$$(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1) - (x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2) - \ln\frac{|\Sigma_2|}{|\Sigma_1|} + 2\ln\frac{P(C_1)}{P(C_2)} = 0$$

3. 분류 경계 Decision boundary

*LDA 가정 2. 범주 각각의 분산은 동일

$$(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1)-(x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2)-\ln\frac{|\Sigma_2|}{|\Sigma_1|}+2\ln\frac{P(\mathrm{C}_1)}{P(\mathrm{C}_2)}=0$$

$$\Sigma=\left[\frac{n_1-1}{(n_1-1)+(n_2-1)}\right]\Sigma_1+\left[\frac{n_2-1}{(n_1-1)+(n_2-1)}\right]\Sigma_2$$

3. 분류 경계 Decision boundary

*LDA 가정 2. 범주 각각의 분산은 동일

$$(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1)-(x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2)-\ln\frac{|\Sigma_2|}{|\Sigma_1|}+2\ln\frac{P(C_1)}{P(C_2)}=0$$

$=0$

$$\Sigma_1=\Sigma_2=\Sigma$$

$$\Sigma=\left[\frac{n_1-1}{(n_1-1)+(n_2-1)}\right]\Sigma_1+\left[\frac{n_2-1}{(n_1-1)+(n_2-1)}\right]\Sigma_2$$

$$(\mu_1-\mu_2)^T\Sigma^{-1}x-\frac{1}{2}(\mu_1-\mu_2)^T\Sigma^{-1}(\mu_1+\mu_2)-\ln\frac{P(C_1)}{P(C_2)}=0$$

3. 분류 경계 Decision boundary

*LDA 가정 2. 범주 각각의 분산은 동일

$$(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)-(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)-\ln\frac{|\Sigma_2|}{|\Sigma_1|}+2\ln\frac{P(C_1)}{P(C_2)}=0$$

(=0)

$$\Sigma_1=\Sigma_2=\Sigma$$

$$\Sigma=\left[\frac{n_1-1}{(n_1-1)+(n_2-1)}\right]\Sigma_1+\left[\frac{n_2-1}{(n_1-1)+(n_2-1)}\right]\Sigma_2$$

$$(\quad A \quad)^{-1}x-\quad b \quad=0$$

= Ax+b의 선형식!
=> 결정 경계가 직선이다!

해당 식에 x를 넣어서 0보다 크면
C1으로 분류, 작으면 C2로 분류!

3. 분류 경계 Decision boundary

$$(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1)-(x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2)-\ln\frac{|\Sigma_2|}{|\Sigma_1|}+2\ln\frac{P(\mathrm{C}_1)}{P(\mathrm{C}_2)}=0$$

두 범주의 분산이 다르다면?

3. 분류 경계 Decision boundary

$$(x-\mu_1)^T{\Sigma_1}^{-1}(x-\mu_1)-(x-\mu_2)^T{\Sigma_2}^{-1}(x-\mu_2)-\ln\frac{|\Sigma_2|}{|\Sigma_1|}+2\ln\frac{P(C_1)}{P(C_2)}=0$$

$$-\frac{1}{2}x^T({\Sigma_1}^{-1}-{\Sigma_2}^{-1})x+(\mu_1{\Sigma_1}^{-1}+\mu_2{\Sigma_2}^{-1})x-\frac{1}{2}(\mu_1^2{\Sigma_1}^{-1}-\mu_2^2{\Sigma_1}^{-1})+\frac{1}{2}\ln\frac{|\Sigma_2|}{|\Sigma_1|}-\ln\frac{P(C_1)}{P(C_2)}=0$$

= $Ax^2+bx+c=0$ 형태의 2차식!
=> 결정 경계가 곡선이다!

이를 QDA라고 한다

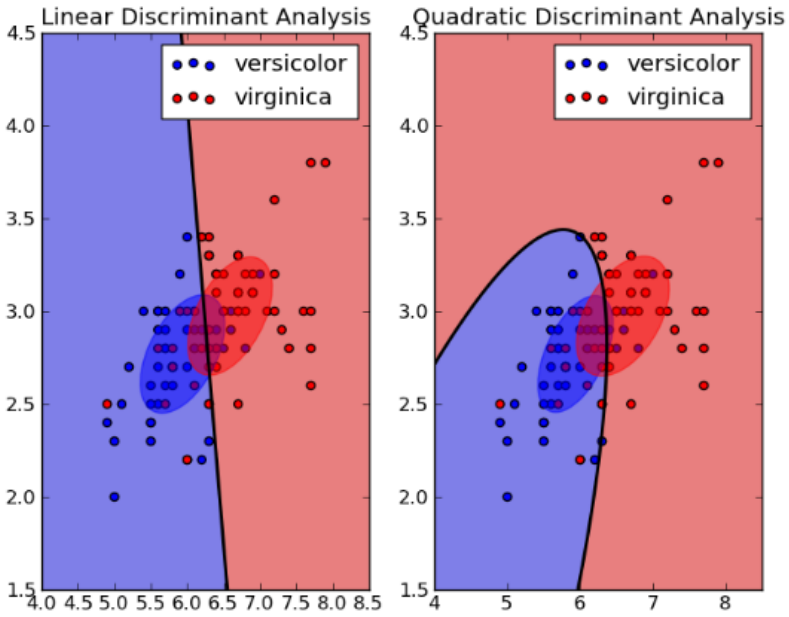
Unit 03 | LDA

QDA

QDA Quadratic discriminant analysis

- 각 범주의 공분산행렬에 대한 가정을 하지 않는다.

- 비선형 결정경계를 갖는다
  => LDA와 비교하여 더 유연하다



Unit 03 | LDA

QDA

QDA Quadratic discriminant analysis

⇒ 결론적으로 LDA는 QDA에 비해서 선형적인 모형을 갖기 때문에 유연성은 떨어지지만 낮은 분산(Variance)을 갖는다.
⇒ 그러나 만약 데이터가 LDA의 가정에 맞지 않으면 LDA의 편향(Bias)는 굉장히 높아진다.

⇒ 따라서 훈련 데이터가 상대적으로 적어 분산을 줄이는 것이 중요하다면, LDA를 선택하는 것이 좋으며,
훈련데이터가 충분히 크고 분류기의 분산이 주요 관심사가 아니거나 클래스에 대한 분산이 공통적이 아니면 QDA를 사용

Unit 02 | KNN

참고

LDA & QDA

- http://scikit-learn.org/stable/modules/lda_qda.html
- https://ratsgo.github.io/machine%20learning/2017/03/21/LDA/
- http://yamalab.tistory.com/41
- https://towardsdatascience.com/is-lda-a-dimensionality-reduction-technique-or-a-classifier-algorithm-eeed4de9953a
- https://web.stanford.edu/class/stats202/content/lec9.pdf
- http://www.datamarket.kr/xe/board_jPWY12/41770 - 투빅스 6기 장재석

과제

분석과제!!

1. 변수추가

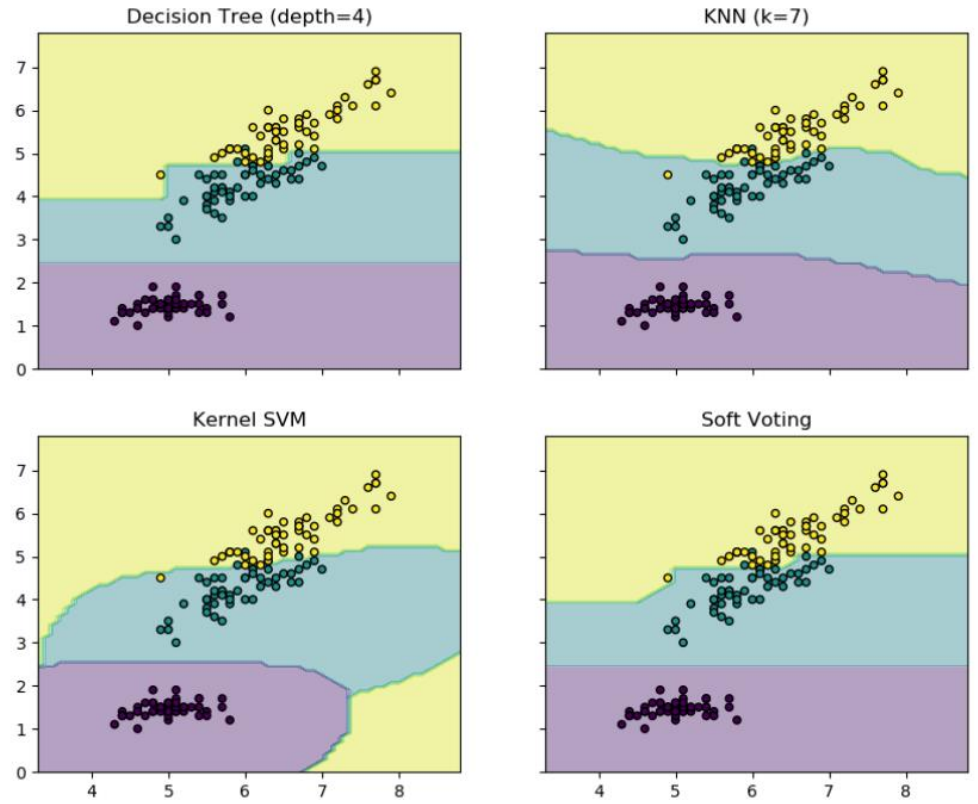
필드명	필드개수	내용
DT	1	총 체류시간(st_t의 총합)
PV	1	총 페이지뷰(st_c의 총합)
COV	1	총 22개의 사이트카테고리(cate)에 얼마나 다양하게 접속했는지에 대한 비율 ("서로 다른 카테고리 수 /22"로 계산)
Day	1	총 접속 일수

과제

Voting Classifier

Target변수 = gender

1. 분석 과제를 KNN, LDA로 해보기
2. 지금까지 배운 분류기 Logistic, KNN, LDA 앙상블해서 해보기



Unit

R 단축키

R studio

Ctrl + enter	해당 블록, 커서가 있는 줄의 코드 실행
Ctrl + shift + c	주석 생성 or 제거
Ctrl + shift + n	새로운 script창 생성
Ctrl + A	스크립트 전체 행 선택
Ctrl + D	해당 줄 삭제
Alt + shift + 좌우 방향키	해당 방향의 줄 선택
Alt + shift + 상하 방향키	해당 줄 복사
Alt + 상하 방향키	해당 줄 이동
;	개행한 것과 동일한 효과
Ctrl + shift + m	(dplyr)패키지 안의 pipe함수 %>%
Shift + click	주석된 링크 실행

http://codedragon.tistory.com/1523

들어주셔서 감사합니다.