

밑바닥부터 시작하는 데이터 과학



4회차

- 기계학습 기초
- 로지스틱 회귀
- 나이브베이지스
- k-nn



1. 기계학습 기초

기계학습 (Machine Learning)

1. 기계학습 기초

기계학습 기초

기계 학습의 방법

- 지도 학습 (Supervised Learning)
 - 학습에 사용될 데이터에 정답이 포함되어 있는 경우
- 비지도 학습 (Unsupervised Learning)
 - 학습에 사용될 데이터에 정답이 포함되어 있지 않은 경우
- 준 지도 학습 (Semi - supervised Learning)
 - 데이터의 일부에만 정답이 포함되어 있는 경우
- 온라인 학습 (online learning)
 - 새로 들어오는 데이터를 통해 모델을 끊임없이 조정하는 경우

모델 (Model)

- 다양한 변수간의 수학적 혹은 확률적 관계를 표현

하이퍼파라미터 (Hyperparameter)

- 기계학습 모델의 parameter
 - 베이저안 통계에서의 하이퍼파라미터와는 전혀 다르다.

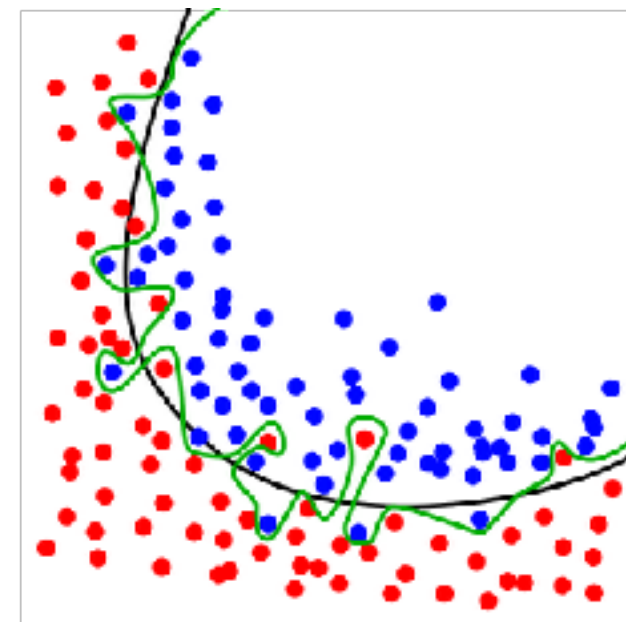


1. 기계학습 기초

기계학습 기초

오버피팅 문제

- 주어진 데이터에만 과하게 적합한 (overfitting) 모델이 만들어지는 문제
 - 새로 관찰된 데이터에 대해서는 전혀 맞지 않는 모델이 되어버린다.
- 이를 해소하기 위해서는 **학습용(train) 데이터**와 **검증용(test) 데이터**를 분리해서 모델링을 한다. (**validation** 이라고 한다)



정확도

- 기계학습 방법론에 따라 모델의 성능을 평가하기 위한 평가 지표가 주어진다.
- 그 중, 분류모델은 어떤 문제에 관한 모델인지에 따라 다른 방식으로 평가하는게 용이하다.
- 일반적으로 알려진 정확도 (Accuracy)
 - 정답을 맞춘 개수 / 전체 개수

1. 기계학습 기초

기계학습 기초

정확도의 단점

- 오차의 종류에 대한 구분이 없다.
 - 스팸 필터링에서 중요한 이메일을 삭제하게 되는 오차로 인한 비용은 잘못해 필터를 통과한 스팸 메일에 대한 비용보다 비쌀 것이다.
- 각 클래스의 자연적인 발생 빈도를 고려하지 못한다.
 - 태아가 다운증후군 등의 유전병을 보유할 확률을 0.1%라고 했을 때, 예측 모델이 모든 태아가 유전병에 대해 음성으로 예측한다면 예측률은 거의 완벽할 것이다.

분류모델의 성능 평가

- Confusion matrix
 - 스팸인데, 스팸으로 제대로 분류된것은 True Positive (TP)
 - 정상인데 정상으로 제대로 분류된것은 True Negative (TN)
 - 스팸인데 정상으로 잘못 분류된것은 False Negative (FN)
 - 실제로는 스팸 이메일인데, 정상으로 판별한 경우
 - 정상 이메일인데 스팸으로 잘못 검출된것은 False Positive (FP)
 - 정상 이메일인데 스팸으로 판별한 경우

		Predicted		
		Positive	Negative	
Observed	Positive	TP	FN	P
	Negative	FP	TN	N

1. 기계학습 기초

기계학습 기초

분류모델의 성능 평가

Recall 과 Precision

		Predicted		
		Positive	Negative	
Observed	Positive	TP	FN	P
	Negative	FP	TN	N

- Accuracy : (True positive + True negative) / (TP + FN + FP + TN)

- Recall : True positive / (True positive + False negative)

- 실제값 중, 정확하게 분류한 예측값 비율

스팸 이메일인데 스팸이라고 맞춘 수(TP)

(스팸 이메일인데 스팸이라고 맞춘 수(TP) + 스팸 이메일인데 정상이라고 분류된 경우(FN))

- Precision : True positive / (True positive + False positive)

- 예측값 중, 정확하게 분류한 실제값 비율

스팸 이메일인데 스팸이라고 맞춘 수(TP)

(스팸 이메일인데 스팸이라고 맞춘 수(TP) + 정상 이메일인데 스팸으로 분류된 경우 (FP))

1. 기계학습 기초

기계학습 기초

분류모델의 성능 평가

평균적으로 전체 5%가 스팸 이메일인 경우.
스팸 이메일을 분류하는 분류모델 두 가지를 만들었다.

1. 100개의 이메일 중 6개를 스팸으로 판별했고, 그 중 5개의 스팸이메일을 정확하게 분류했다.
2. 100개의 이메일 중 4개를 스팸으로 판별했고, 그 중 4개의 스팸이메일을 정확하게 분류했다.

각 모델의 정확도, recall, precision은 얼마인가.
그리고 어떤 모델이 더 좋은 모델인가?

1. 기계학습 기초

기계학습 기초

분류모델의 성능 평가

1. 100개의 이메일 중 6개를 스팸으로 판별했고, 그 중 5개의 스팸이메일을 정확하게 분류했다.

- 정확도 : 99/100
- Recall : 5/5 = 1

스팸 이메일인데 스팸이라고 맞춘 수(TP) = 5

(스팸 이메일인데 스팸이라고 맞춘 수(TP) (=5) + 스팸 이메일인데 정상이라고 분류된 경우(FN)) (=0)

- Precision : 5/6 = 0.83

스팸 이메일인데 스팸이라고 맞춘 수(TP) (=5)

(스팸 이메일인데 스팸이라고 맞춘 수(TP) (=5) + 정상 이메일인데 스팸으로 분류된 경우 (FP)) (=1)

1. 기계학습 기초

기계학습 기초

분류모델의 성능 평가

2. 100개의 이메일 중 4개를 스팸으로 판별했고, 그 중 4개의 스팸이메일을 정확하게 분류했다.

- 정확도 : 99/100
- Recall : $4/5 = 0.8$

스팸 이메일인데 스팸이라고 맞춘 수(TP) = 4

(스팸 이메일인데 스팸이라고 맞춘 수(TP) (=4) + 스팸 이메일인데 정상이라고 분류된 경우(FN)) (=1)

- Precision : $4/4 = 1$

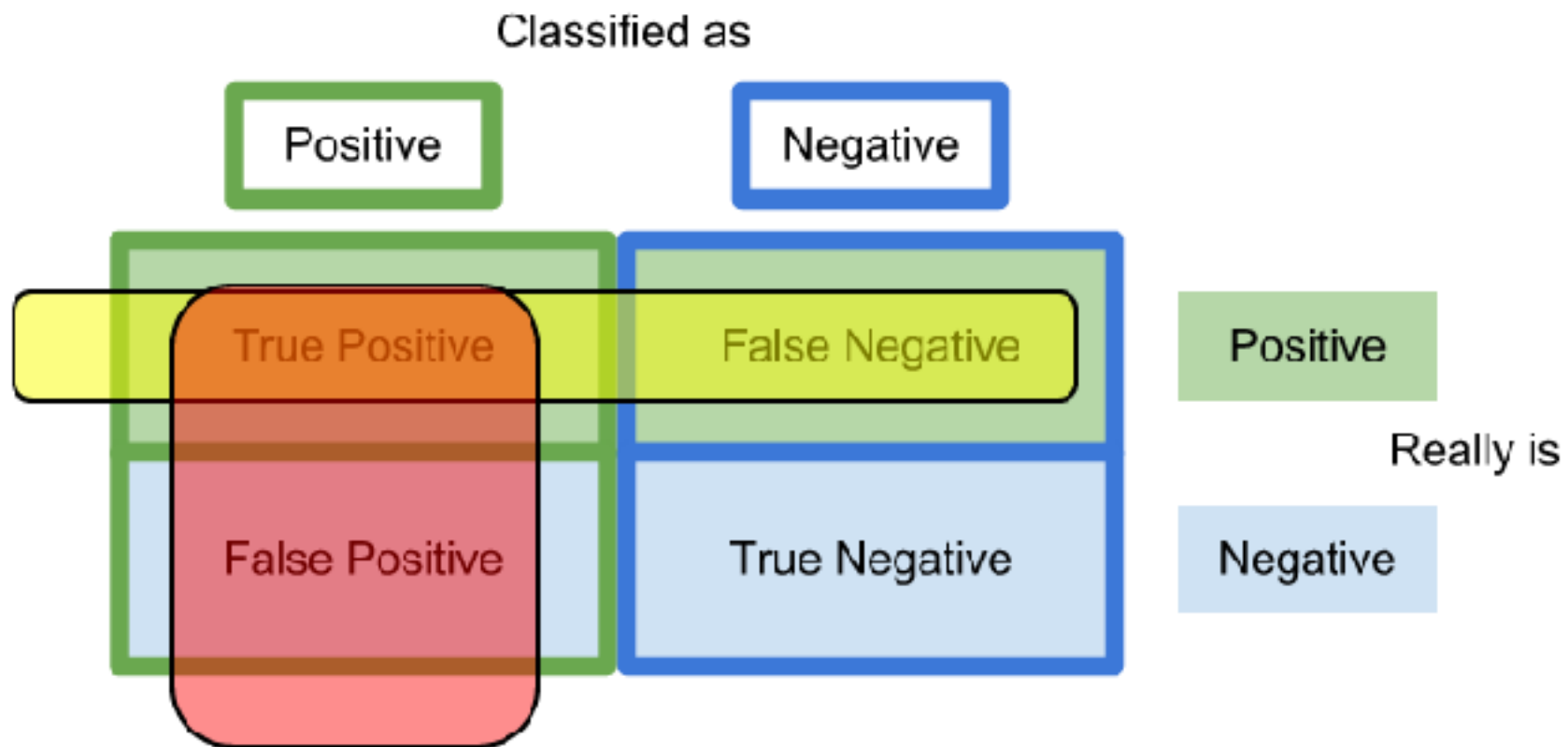
스팸 이메일인데 스팸이라고 맞춘 수(TP) (=4)

(스팸 이메일인데 스팸이라고 맞춘 수(TP) (=4) + 정상 이메일인데 스팸으로 분류된 경우 (FP)) (=0)

1. 기계학습 기초

기계학습 기초

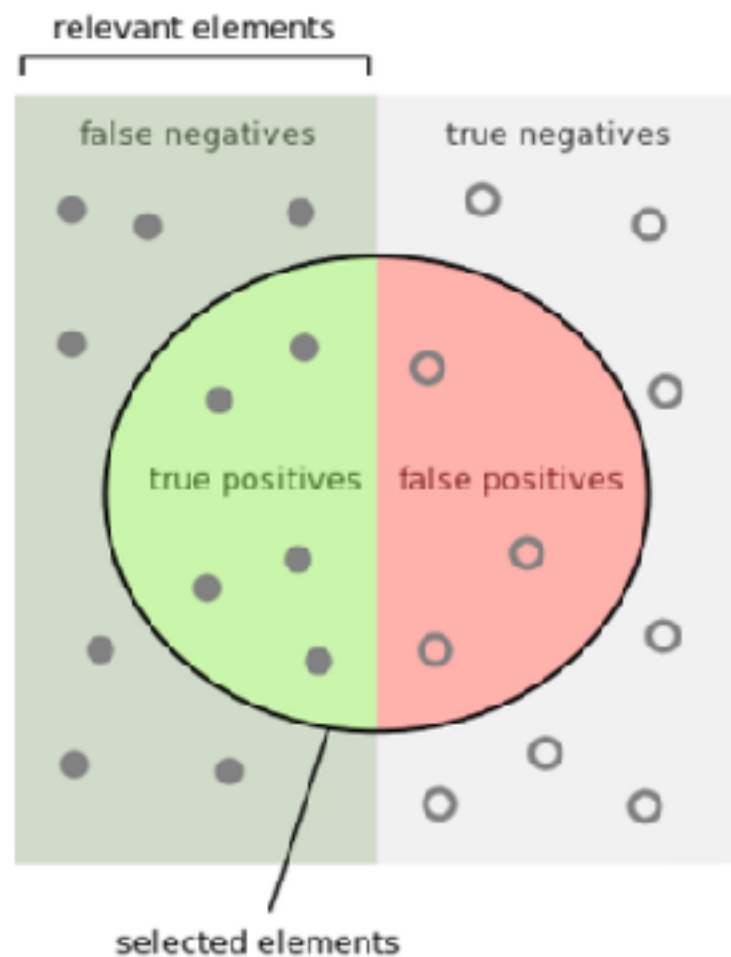
분류모델의 성능 평가





1. 기계학습 기초


기계학습 기초


분류모델의 성능 평가



Accuracy =  =
$$\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$$

Precision =  =
$$\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$$

Recall =  =
$$\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$$

False Positive Rate = (Fall-out) =  =
$$\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$$

2. 로지스틱 회귀

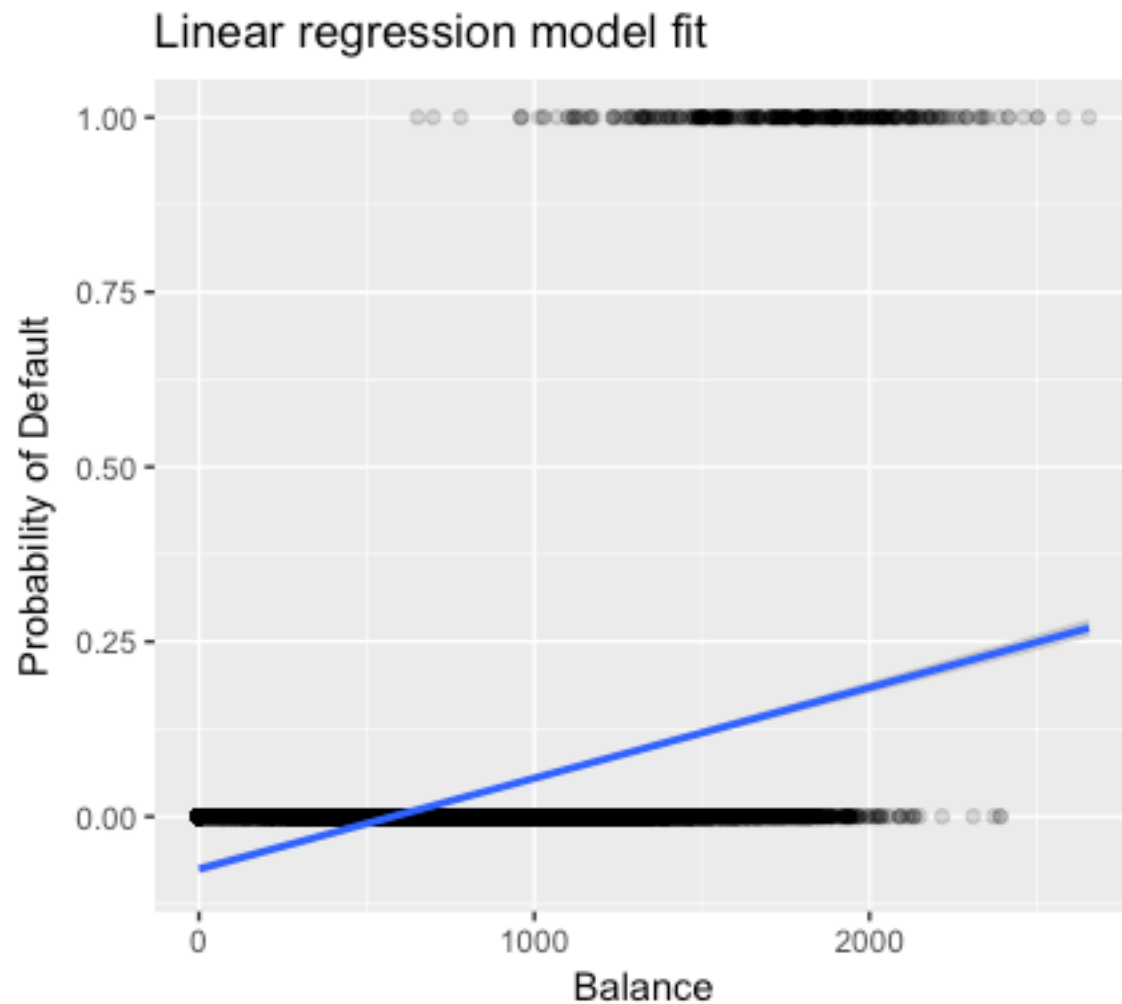
로지스틱회귀 (Logistic Regression)

2. 로지스틱 회귀

Motivation

선형회귀의 경우에는 종속 변수(y)가 범주형 데이터인 경우에는 사용하기 어렵다.

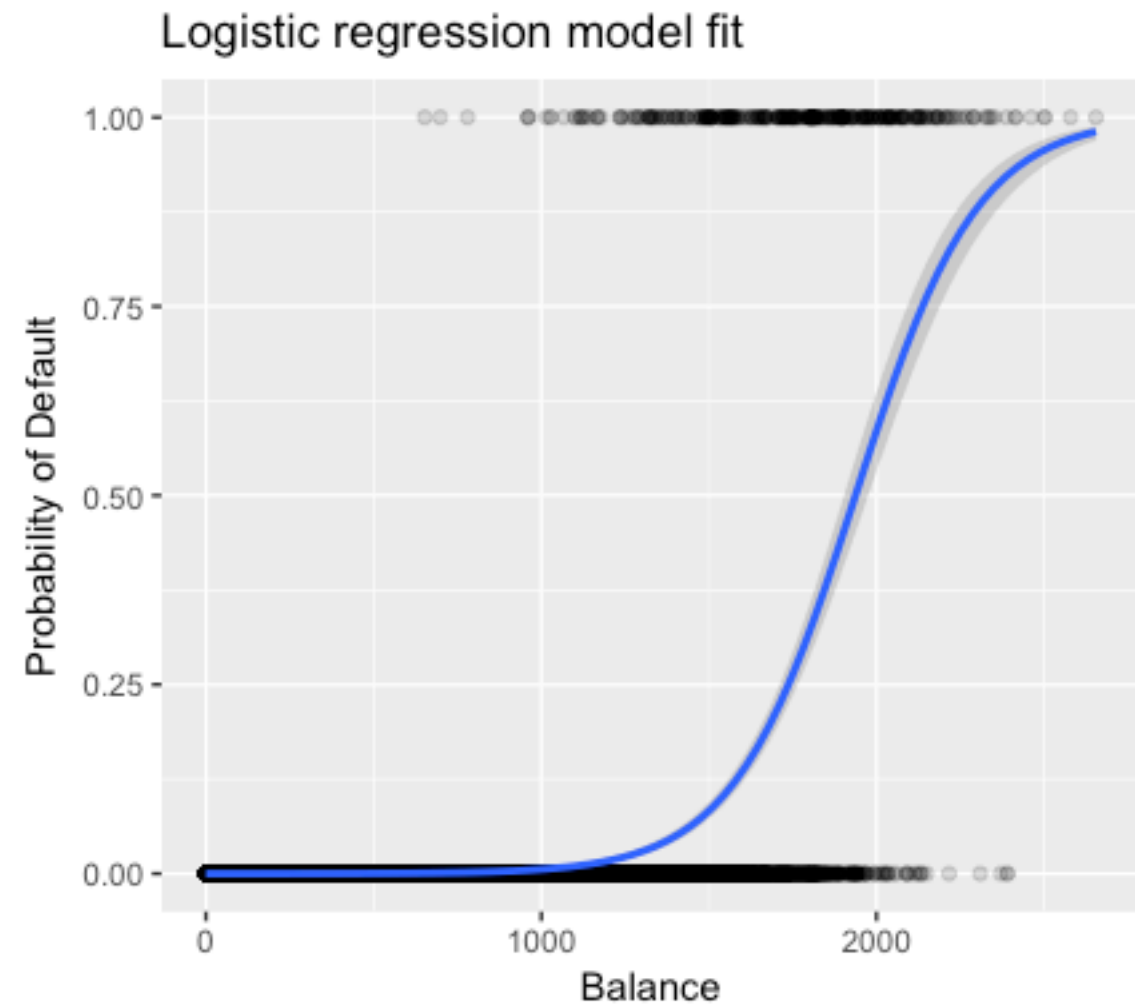
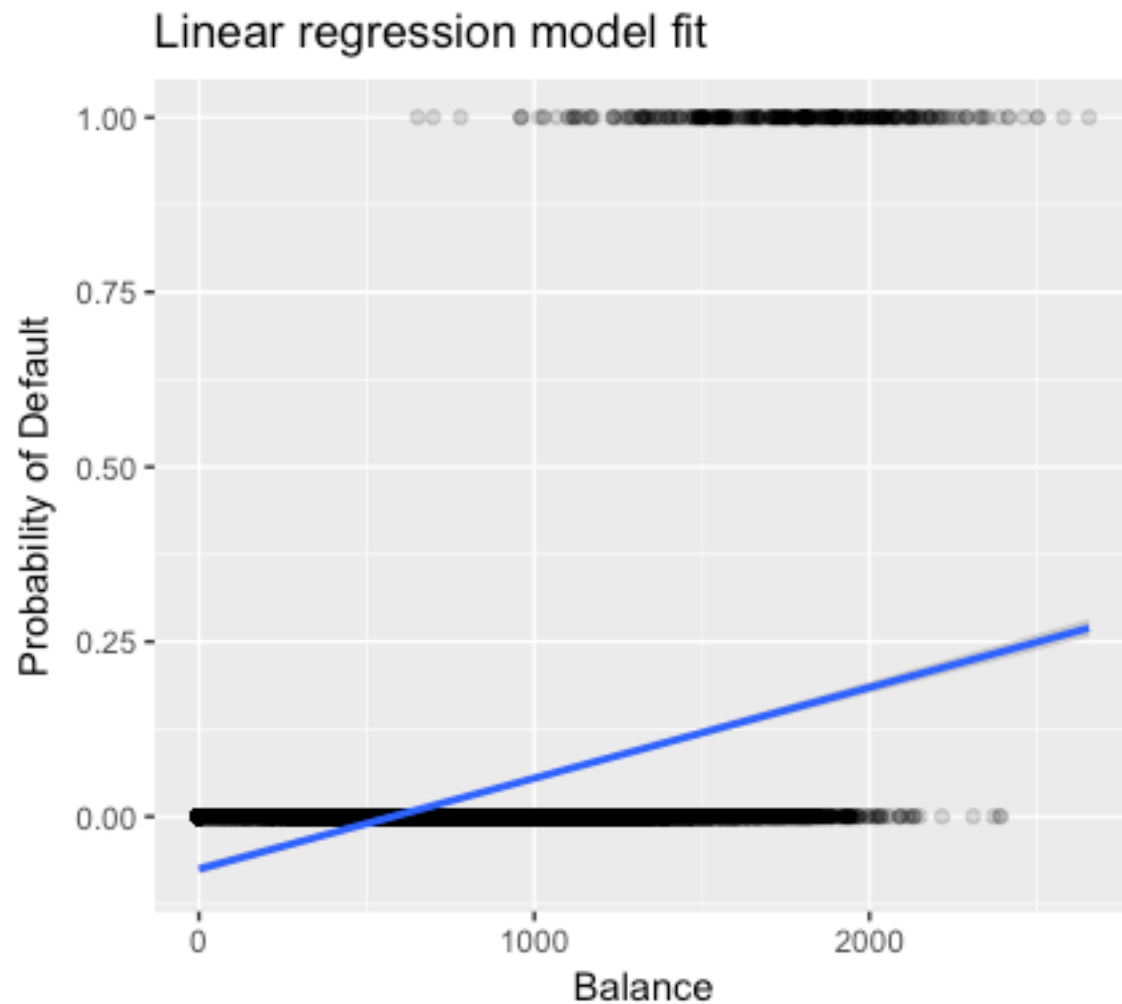
- 예시 : 잔고 (x)에 대한 채무불이행 ($y, (0, 1)$) 예측



2. 로지스틱 회귀

Motivation

| 0, 1을 맞추는 문제가 아닌 0, 1이 될 확률을 예측하는 문제로 바꾸어보자.





2. 로지스틱 회귀

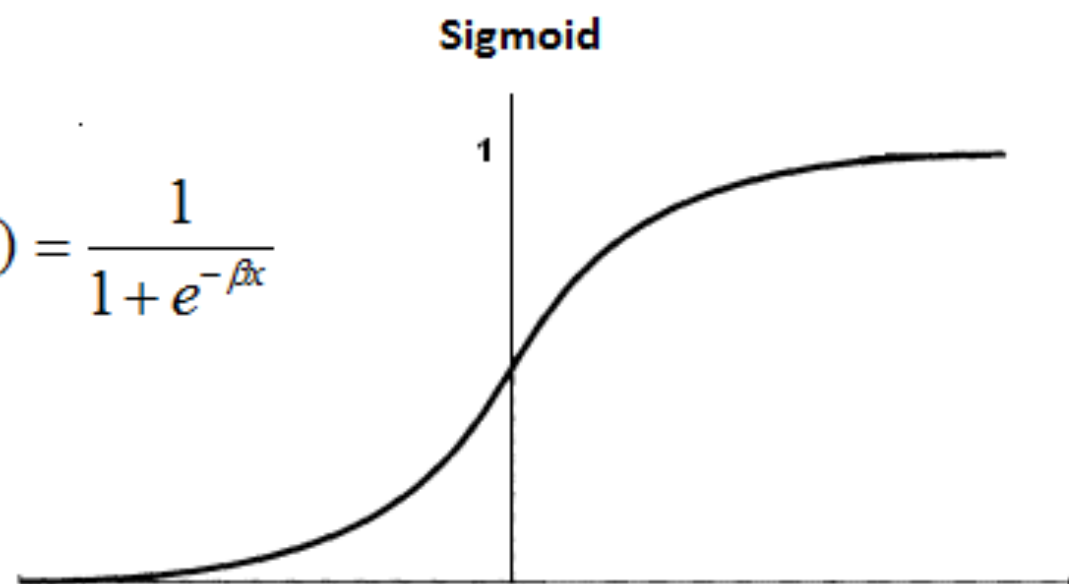
Mathematical Background

로지스틱 함수

- 결과 값이 항상 0과 1사이의 값만 가지도록 변형한다.
 - 이런 형태로 최대값과 최소값에 정해져 있는 함수들을 시그모이드 함수라고 한다.
 - 시그모이드 함수 중 대표적인 함수가 로지스틱 함수이므로 구분없이 사용하기도 한다.

$$\theta(x;w) = \frac{1}{1 + \exp(-w^T x)}$$

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$



2. 로지스틱 회귀

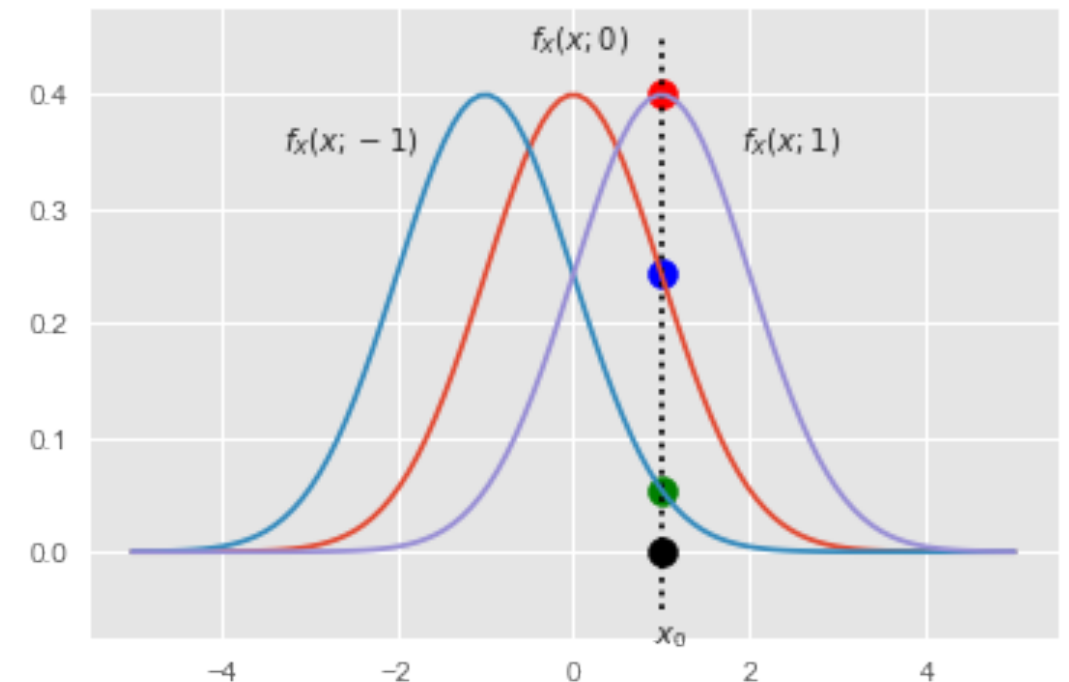
Mathematical Background

최대우도추정법 (Maximum Likelihood Estimation)

- 주어진 샘플 x 에 대해 **likelihood**를 가장 크게 해주는 모수 θ 를 찾는 방법.
- 데이터가 어떤 확률 분포 $f_x(x; \theta)$ 에서 나왔고, x 에 대한 확률 값을 알아야 되는 상황을 가정해보자.
 - θ : 확률 모형의 모수(parameter)집합
 - θ 가 주어진 경우에는 x 에 대한 확률값을 바로 계산할 수 있다.
 - (함수의 관점에서는 θ 는 주어져있고, x 만을 변수로 가정한다.)
- 하지만 현실세계에서는 주어진 데이터에 대한 θ 를 알고있기 힘들다.
- 그렇다면 주어진 데이터에서 가장 그럴듯한 θ 는 무엇인지 알면, 반대로 x 에 대한 확률 값을 계산할 수 있지 않을까?
- **likelihood 함수** : $L(\theta; x)$
- 주어진 x 에 대한 θ 의 상대적 가능성을 계산할 수 있다.
 - likelihood를 가능도라고도 부른다.

예시) 정규분포를 가지는 확률 변수의 **평균**은 모르는 상태

- x_0 으로 주어졌을 때, 평균=1일때 likelihood가 가장 크다.



2. 로지스틱 회귀

Logistic Mathematics

전제조건

- Y를 확률 변수로 생각해보자.
- 분류(예측)하고자 하는 값이 0과 1로만 이루어져있다면, Y는 **베르누이 확률 변수**로 생각할 수 있다.
- likelihood는 0과 1이 될 확률을 변수로 가지는 함수로 정의할 수 있다.

$$P(y=y_i) = P^{y_i} (1-P)^{1-y_i}$$

$$L = \prod_i P^{y_i} (1-P)^{1-y_i}$$

2. 로지스틱 회귀

Logistic Mathematics

로지스틱회귀의 likelihood함수

- 관측치가 i 개 있고, y 의 결과는 0과 1만 있는 이항로지스틱 모델의 파라미터 w 가 주어졌다고 가정해보자.
- y_i 는 $\theta(w^T x_i)$ 의 확률로 1, $1 - \theta(w^T x_i)$ 의 확률로 0이 된다.
 - 여기서 θ 는 로지스틱 함수이다.
- 이 경우 likelihood 함수는 다음과 같이 쓸 수 있다.

$$L = \prod \theta(w^T x_i)^{y_i} \{1 - \theta(w^T x_i)\}^{1-y_i}$$

로그 변환

$$\downarrow$$
$$\ln L = \sum y_i \ln\{\theta(w^T x_i)\} + \sum (1 - y_i) \ln\{1 - \theta(w^T x_i)\}$$

- log로 변환하는 이유는 계산을 단순화 시키기 위해서이다.

2. 로지스틱 회귀

Logistic Mathematics

Log Likelihood를 최대화 시키는 w

- LL을 로지스틱 함수로 정리하면

$$\begin{aligned} LL &= \log \prod_{i=1}^N \theta_i(x_i; w)^{y_i} (1 - \theta_i(x_i; w))^{1-y_i} \\ &= \sum_{i=1}^N (y_i \log \theta_i(x_i; w) + (1 - y_i) \log(1 - \theta_i(x_i; w))) \\ &= \sum_{i=1}^N \left(y_i \log \left(\frac{1}{1 + \exp(-w^T x_i)} \right) + (1 - y_i) \log \left(\frac{\exp(-w^T x_i)}{1 + \exp(-w^T x_i)} \right) \right) \end{aligned}$$

- LL을 최대화 시키는 w 를 구하기 위해 LL를 w 로 미분한다.

$$\frac{\partial LL}{\partial w} = \sum_{i=1}^N \frac{\partial LL}{\partial \theta_i(x_i; w)} \frac{\partial \theta_i(x_i; w)}{\partial w}$$

합성함수의 미분법 !

2. 로지스틱 회귀

Logistic Mathematics

Log Likelihood를 최대화 시키는 w

$$\frac{\partial LL}{\partial w} = \sum_{i=1}^N \boxed{\frac{\partial LL}{\partial \theta_i(x_i; w)}} \frac{\partial \theta_i(x_i; w)}{\partial w}$$

$$\frac{\partial LL}{\partial \theta_i(x_i; w)} = \left(y_i \frac{1}{\theta_i(x_i; w)} - (1 - y_i) \frac{1}{1 - \theta_i(x_i; w)} \right)$$

2. 로지스틱 회귀

Logistic Mathematics

Log Likelihood를 최대화 시키는 w

$$\frac{\partial LL}{\partial w} = \sum_{i=1}^N \frac{\partial LL}{\partial \theta_i(x_i; w)} \frac{\partial \theta_i(x_i; w)}{\partial w}$$

$$\frac{\partial \theta_i(x_i; w)}{\partial w} = \frac{\partial}{\partial w} \frac{1}{1 + \exp(-w^T x_i)}$$

$$= \frac{\exp(-w^T x_i)}{(1 + \exp(-w^T x_i))^2} x_i = \theta_i(x_i; w)(1 - \theta_i(x_i; w))x_i$$

2. 로지스틱 회귀

Logistic Mathematics

Log Likelihood를 최대화 시키는 w

- 두 식을 곱하면

$$\begin{aligned}\frac{\partial LL}{\partial w} &= \sum_{i=1}^N \left(y_i \frac{1}{\theta_i(x_i; w)} - (1 - y_i) \frac{1}{1 - \theta_i(x_i; w)} \right) \theta_i(x_i; w)(1 - \theta_i(x_i; w))x_i \\ &= \sum_{i=1}^N \left(y_i(1 - \theta_i(x_i; w)) - (1 - y_i)\theta_i(x_i; w) \right) x_i \\ &= \sum_{i=1}^N \left(y_i - \theta_i(x_i; w) \right) x_i\end{aligned}$$

- 이 값은 w 에 대한 비선형 함수이므로 선형 모형과 같이 간단하게 그래디언트가 0이 되는 모수 w 값에 대한 수식을 구할 수 없으며 수치적인 최적화 방법(numerical optimization)을 통해 최적 모수 의 값을 구해야 한다.

2. Naive Bayes

Naive Bayes

조건부 독립이라는 간단한(naive) 가정을 통해 데이터를 분류하는 방법론

2. Naive Bayes

Naive Bayes

나이프 베이즈 모델은 베이즈 정리를 이용하여, 분류하고자 하는 대상의 각 분류별 확률을 측정하여, 그 확률이 큰 쪽으로 분류하는 방법을 취한다.

베이즈 정리 Remind

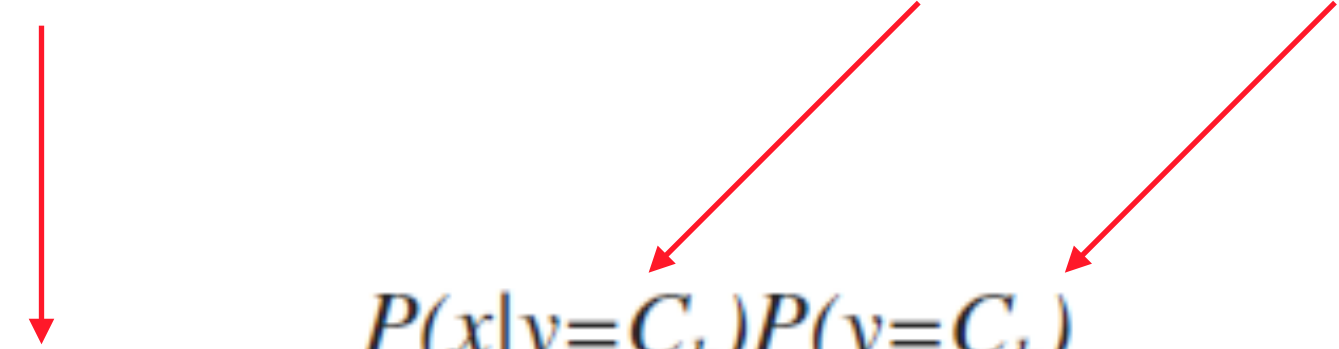
- 사건 B가 발생함으로써 (사건 B가 진실이라는 것을 알게 됨으로써 즉, 사건 B의 확률 $P(B)=1$ 이라는 것을 알게 됨으로써) 사건 A의 확률이 어떻게 변화하는지 표현한 정리
- 사건 B가 발생하였다는 것은 우리가 찾는 샘플이 사건 B라는 부분집합에 포함되어 있다는 새로운 정보를 취득하였다는 의미이다.

2. Naive Bayes

Naive Bayes

나이브 베이즈 모델은 베이즈 정리를 이용하여, 분류하고자 하는 대상의 각 분류별 확률을 측정하여, 그 확률이 큰 쪽으로 분류하는 방법을 취한다.

- 분류하고자 하는 대상의 확률(사후 확률, posteriori)을 계산하기 위해 likelihood, 사전확률 (prior)이 필요하다.

$$P(y=C_k|x) = \frac{P(x|y=C_k)P(y=C_k)}{P(x)}$$


2. Naive Bayes

Naive Bayes

| 사전 확률 계산

- 주어진 데이터를 바탕으로 전체 데이터에서 각 클래스별 비율을 계산한다.
 - 전체 데이터 : 100
 - 클래스1 : 10
 - $P(\text{클래스1}) = 10/100$
 - 클래스2 : 90
 - $P(\text{클래스2}) = 90/100$

| Likelihood 계산

- 모든 x 가 상호 **독립**이라는 가정하에 각 클래스에 대한 조건부 확률을 구한다.
(= 결합 확률을 단순 곱으로 구할 수 있게된다)
 - **likelihood 확률 분포**
 - 베르누이 분포
 - 다항분포
 - 정규분포
 - x 가 연속변수일 때는 정규분포, 여러 개의 값을 가진 카테고리값인 경우 (ex. 단어) 다항분포를 가정한다.

$$P(x|y=C_k) = \prod_{i=1}^P P(x_j | y=C_k)$$

2. Naive Bayes

Naive Bayes

학습 과정 예시

- 영화 리뷰 데이터를 통해 긍정, 부정 리뷰 판별

1. 각 클래스 (긍정, 부정)의 사전 확률을 계산한다.

- 각 클래스의 비율을 계산한다.
 - 전체 100개 리뷰 중, 긍정 리뷰 67개, 부정리뷰 33개가 있다면,
 - 긍정 리뷰의 사전 확률 : 0.67
 - 부정 리뷰의 사전확률 : 0.33

2. 모든 x에 대해 likelihood를 계산한다.

- $P(x|y = \text{부정})$, $P(x|y = \text{긍정})$
- 여기서 x는 리뷰의 단어 (혹은 어절) 이다.
 - [정우성이 나오는 영화라서 너무 좋았다.] : 긍정
 - [외계인이 나와서 너무 싫었다.] : 부정
 - [외계인도 나와서 너무 좋았다.] : 긍정

- $P(\text{정우성} | y = \text{부정}) : 0/1$, $P(\text{정우성} | y = \text{긍정}) : 1/2$
- $P(\text{외계인} | y = \text{부정}) : 1/1$, $P(\text{외계인} | y = \text{긍정}) : 1/2$

2. Naive Bayes

Naive Bayes

학습 과정 예시

- 영화 리뷰 데이터를 통해 긍정, 부정 리뷰 판별

3. 새로운 리뷰에 대한 확률을 예측한다.

- 긍정 리뷰의 사전 확률 : 0.67
- 부정 리뷰의 사전 확률 : 0.33
- $P(\text{정우성} \mid y = \text{부정}) : 0/1, P(\text{정우성} \mid y = \text{긍정}) : 1/2$
- $P(\text{외계인} \mid y = \text{부정}) : 1/1, P(\text{외계인} \mid y = \text{긍정}) : 1/2$
- 외계인이 나와서 흥미로웠어요!
 - $P(Y = \text{긍정} \mid x) : P(\text{외계인} \mid y = \text{긍정}) * P(y = \text{긍정})$
 - $0.5 * 0.67 = 0.335$
 - $P(Y = \text{부정} \mid x) : P(\text{외계인} \mid y = \text{부정}) * P(y = \text{부정})$
 - $1 * 0.33 = 0.33$
- $P(Y = \text{긍정} \mid x) > P(Y = \text{부정} \mid x) \rightarrow$ 이 리뷰는 긍정리뷰이다!

4. K-NN (K- nearest neighbor)

K-NN (K- nearest neighbor)

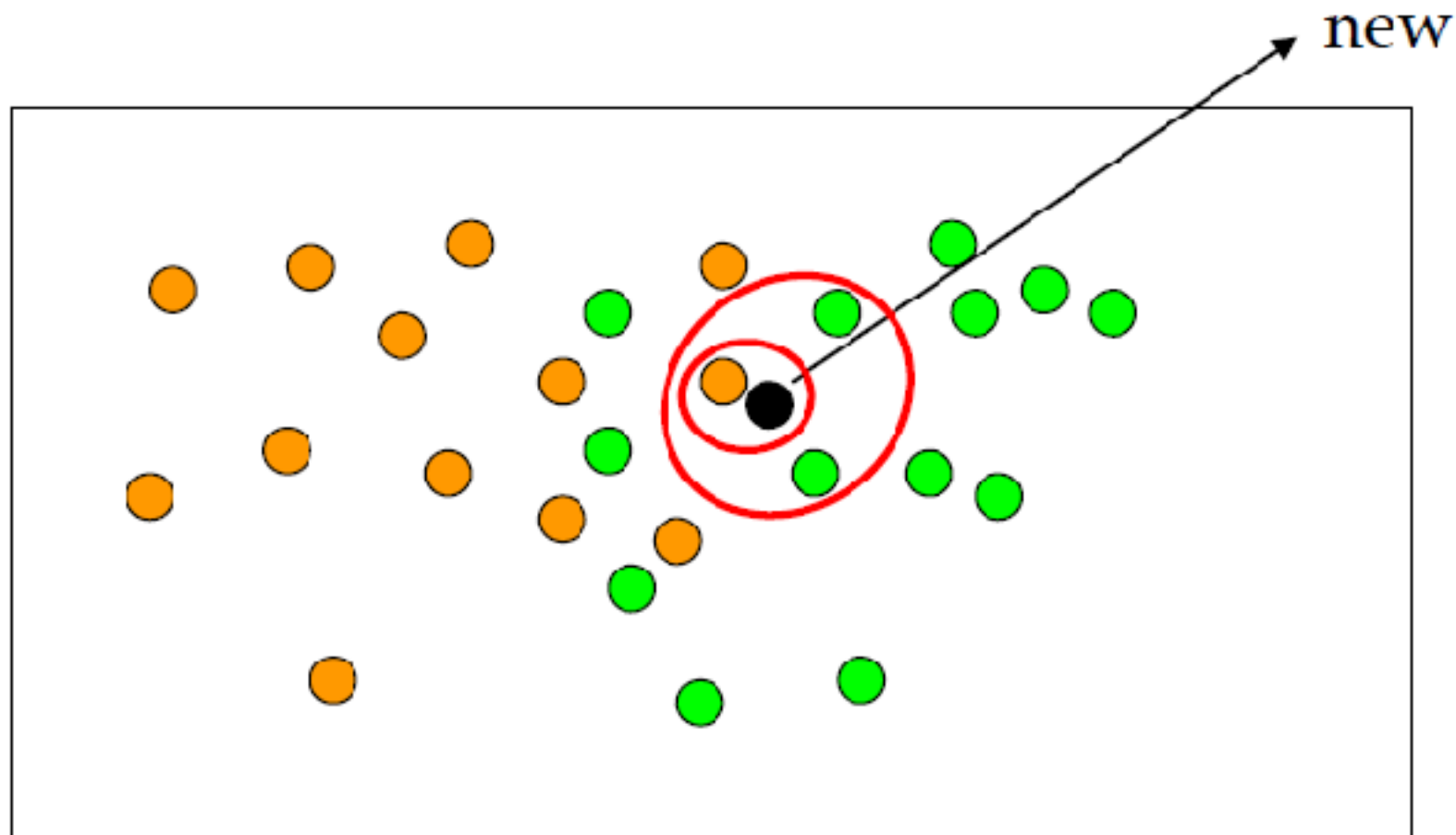
새로운 데이터가 주어졌을 때 기존 데이터 가운데 가장 가까운 k개 이웃의 정보로 새로운 데이터를 예측하는 방법론

4. K-NN (K- nearest neighbor)

K-NN

수학적인 가정없이 새로운 데이터가 들어왔을 때, k개의 가장 가까운 (=이웃) 기존 데이터로 새로운 데이터의 결과값을 추론한다. (k -means와는 완전히 다른 모델임)

- k = 1 이면 new = 오렌지색
 - k = 3 이면 new = 녹색
- 아주 간단한 모델이며 분류 (classification), 회귀 (regression)에 모두 적용 가능함

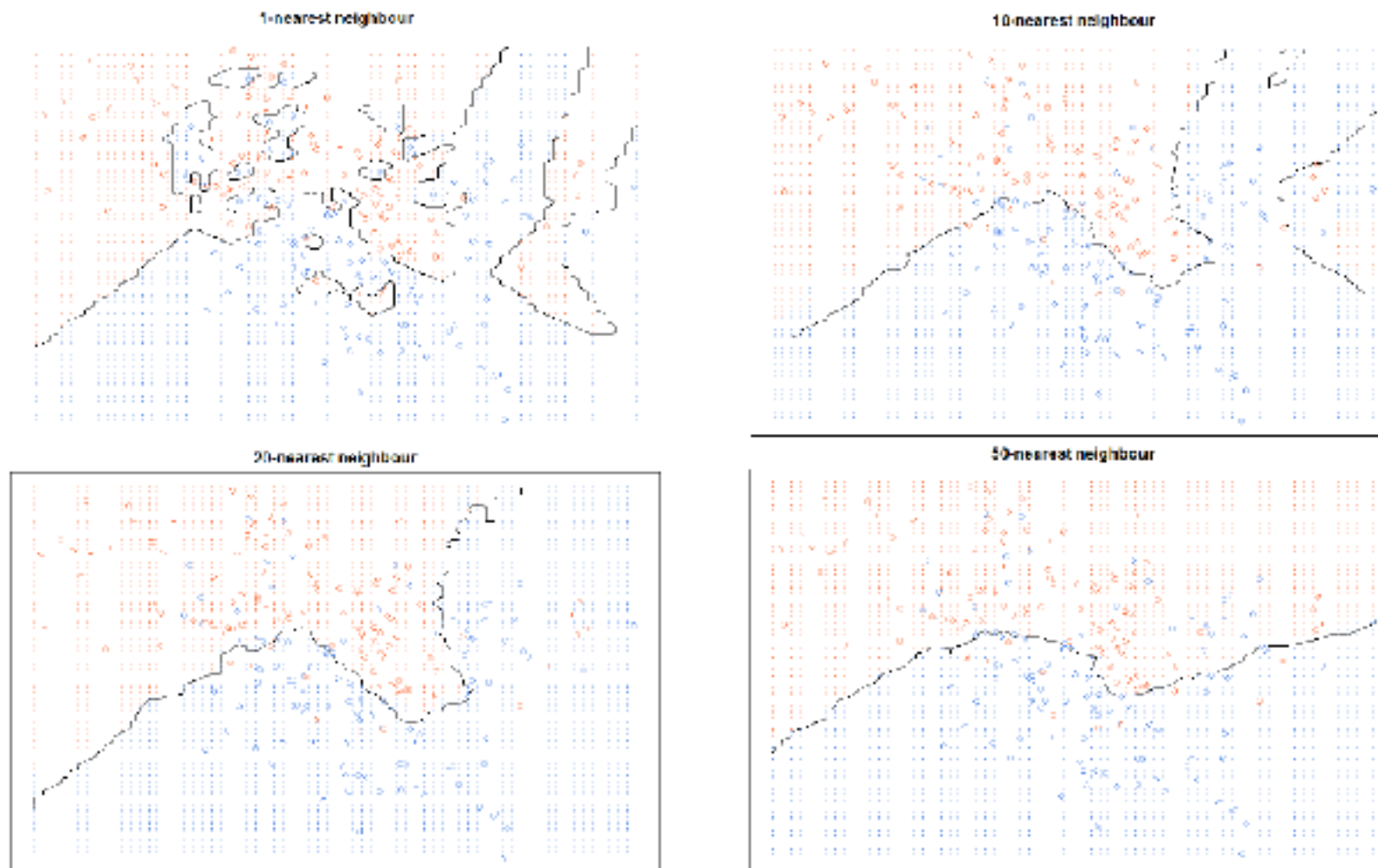


4. K-NN (K- nearest neighbor)

K-NN

하이퍼파라미터 : k (이웃), 거리 측정 방법

- k가 지나치게 작을 경우 : 지역적특성을 지나치게 반영함 (overfitting)
- k가 지나치게 클 경우 : 과하게 정규화한다 (underfitting)

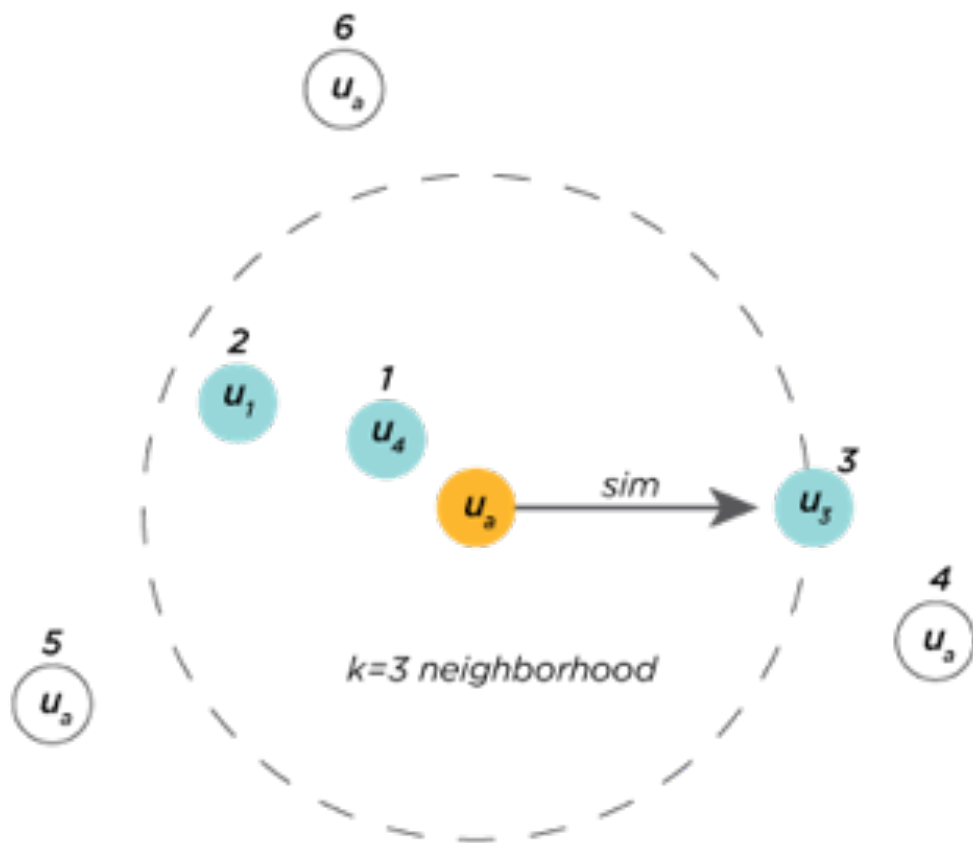


4. K-NN (K- nearest neighbor)

K-NN

활용 예시 (추천 시스템)

- User Based Collaborative Filtering
- 거리상으로 가까운 유저의 item 정보를 바탕으로 추천



	i_1	i_2	i_3	i_4	i_5	i_6
u_3	?	?	4.0	3.0	?	?
u_1	?	4.0	4.0	2.0	1.0	2.0
u_2	3.0	?	?	?	5.0	1.0
u_3	3.0	?	?	3.0	2.0	2.0
u_4	4.0	?	?	2.0	1.0	1.0
u_5	1.0	1.0	?	?	?	?
u_6	?	1.0	?	?	1.0	1.0
	3.5	4.0			1.3	

Recommendations: i_2, i_1

4. K-NN (K- nearest neighbor)

K-NN

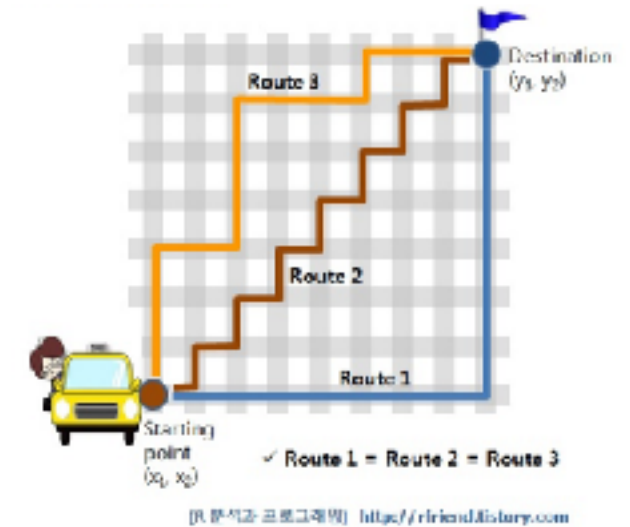
거리 지표

- Euclidean Distance

- 두 관측치의 최단 거리를 의미

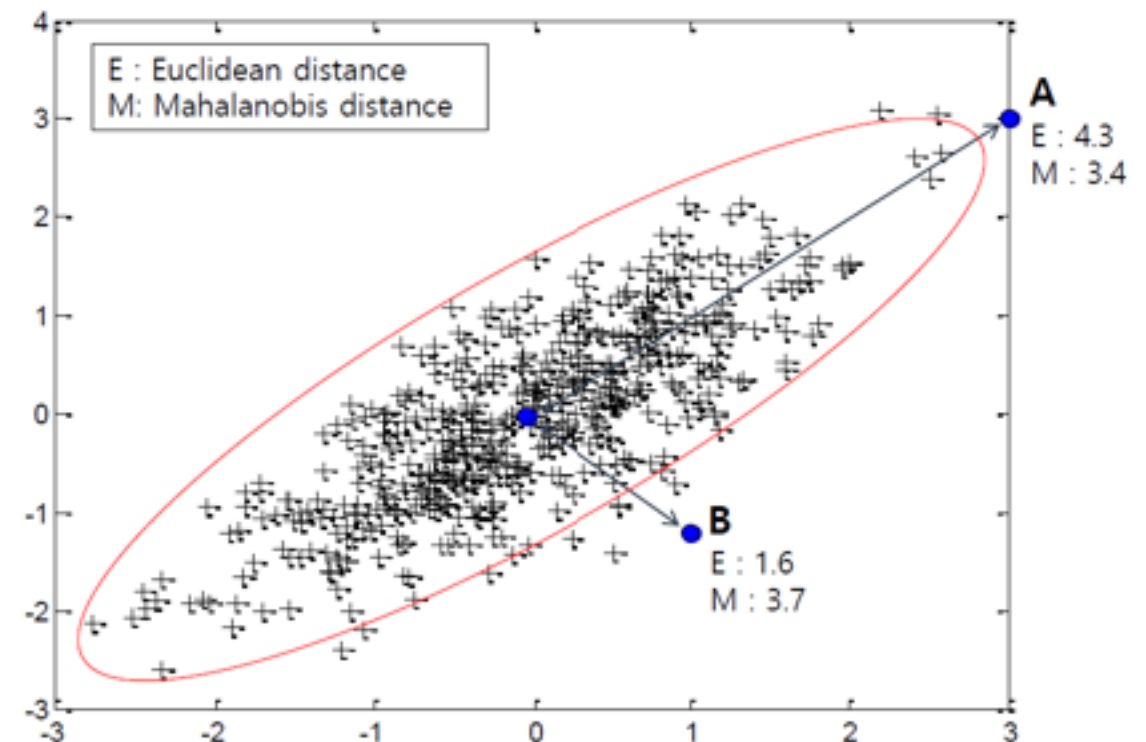
- Manhattan Distance

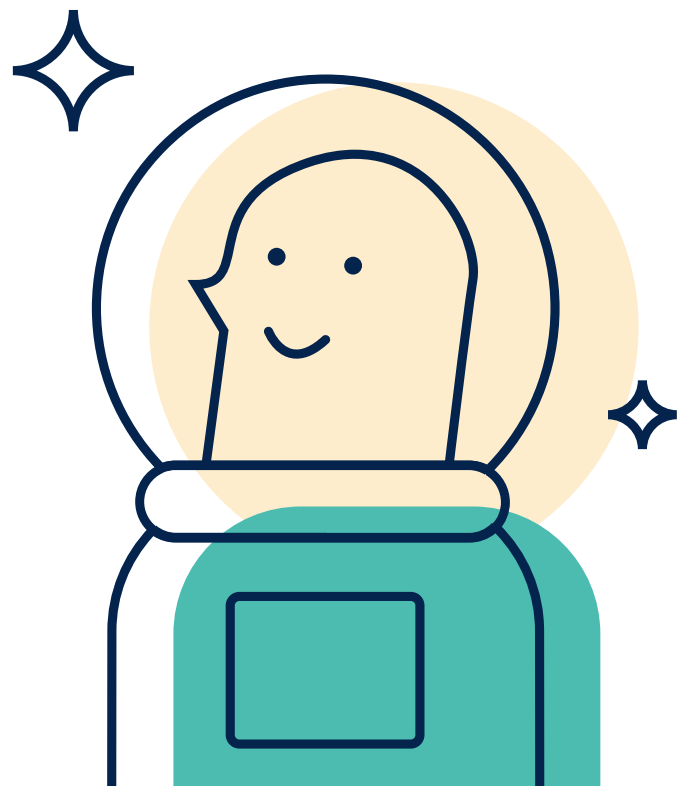
- A에서 B로 이동할 때 각 좌표축 방향으로만 이동할 경우에 계산되는 거리



- Mahalanobis Distance

- 변수 내 분산, 변수간 공분산을 모두 반영하여 거리를 계산하는 방식





다음에
또!
같이!
만나요!