

모델 성능 평가 (Performance Evaluation)

Hyerim Bae

Department of Industrial Engineering, Pusan National University

hrbae@pusan.ac.kr

코로나 19 검사

KBS NEWS

분야별 ▼ 시사·다큐 ▼ TV뉴스 ▼

‘코로나19’ 쟁대믹

[이슈체크] 정부가 코로나19 ‘신속진단키트’ 도입 꺼리는 이유는?

발행 2020.09.19 (07:03) | 수정 2020.09.19 (09:03)

백지채크

1 4

가



Source: <http://news.kbs.co.kr/news/view.do?ncd=5008172&ref=A>

이런 이유로 **항원·항체 검사**는 **유전자 검사**에 비해 **정확도가 떨어지는** 걸로 보고됐습니다. 식약처는 그런 점을 고려해 진단시약의 허가 기준을 항원·항체 검사의 경우 임상적 민감도 70% 이상, 특이도 90% 이상을 충족하도록 했습니다. 민감도란 질병이 있는 사람을 질병이 있다고 진단할 확률을 뜻하고 특이도는 그 반대의 경우를 말합니다. 민감도 90%, 특이도 95% 이상인 유전자 검사의 승인 기준보다 낮은 수치입니다.

보건 당국은 현재로서는 이런 장점보다 **진단의 정확성이 가장 중요하다**는 입장입니다.

구분	유전자 검사	항원 검사	항체 검사
검사 목적	코로나19 바이러스 유전자 유무 확인	코로나19 바이러스 특정 단백질 유무 확인	코로나19 바이러스에 대한 항체 생성여부 확인
검사 물질	바이러스 유전자	바이러스 특정 단백질	체내 생성 항체
사용 검체	코 또는 목의 점액, 가래(객담)	코 또는 목의 점액	혈액
검사 시간	약 3 ~ 6시간	약 15분	약 15분
장점	정확도가 높아 확진용으로 사용	유전자 검사 대비 검사 시간 짧고 비용 낮음	과거 감염이력 확인 가능, 검사시간 짧고 비용 낮음
단점	과거 감염 이력 확인 불가 검사시간 길고 비용 높음	유전자 검사 대비 낮은 정확도, 확진용으로 사용 어려움	감염 초기 항체가 확인되지 않을 수 있고 검사당시 검체 내 바이러스 유무 직접 확인 어려움
측정 원리	바이러스 유전자를 증폭하여 감염여부 확인	바이러스와 결합한 특정 물질을 검출하여 바이러스 감염여부 확인	체내에 생성된 항체와 결합한 물질을 분석하여 항체 존재여부 확인
검사자 (사용자)	의료인 또는 검사 전문가	의료인 또는 검사 전문가	의료인 또는 검사 전문가

식품의약품안전처 자료.

<http://bael>

Contents

01

Overview

02

Lift

03

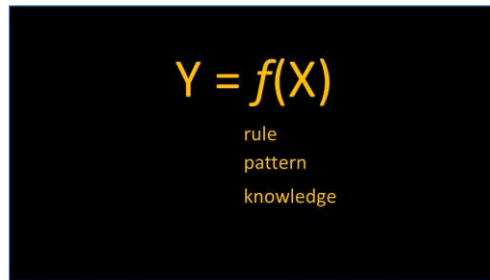
손실 평가(Cost evaluation)

모델 성능 평가

성능 평가의 필요성

- 분류나 예측을 위해 사용할 수 있는 다양한 방법론이 존재
 - 인공신경망을 쓸까? 의사결정 나무를 쓸까?
- 각각의 방법론에 대해 다양한 설정을 선택할 수 있음
 - Activation function을 ReLu를 쓸까? Sigmoid를 쓸까?
- 최선의 모델을 선택하기 위해 각 모델(또는 각 모델의 설정)을 평가해야 함
 - 기계학습의 정의를 다시한번 생각해보자.

- Finding ' f ' such that


$$Y = f(X)$$

rule
pattern
knowledge

- We use X and Y to find ' f '

모델의 출력 유형

- 수치형 데이터(연속형) L
 - 이번학기 산업데이터과학 중간고사 점수는?
- 클래스(Class)
 - 이번학기 산업데이터과학의 평점은?
- 경향성(Tendency): 특정 클래스에 속할 확률
 - 내가 이번학기에 산업데이터 과학에서 A+를 받을 확률은?

오분류 에러(Misclassification error)

- 에러(Error) = 데이터가 속한 클래스를 잘못 분류한 경우
- 에러율(Error rate) = 전체 데이터 중 오분류된 데이터의 비율

벤치마크(Benchmark)

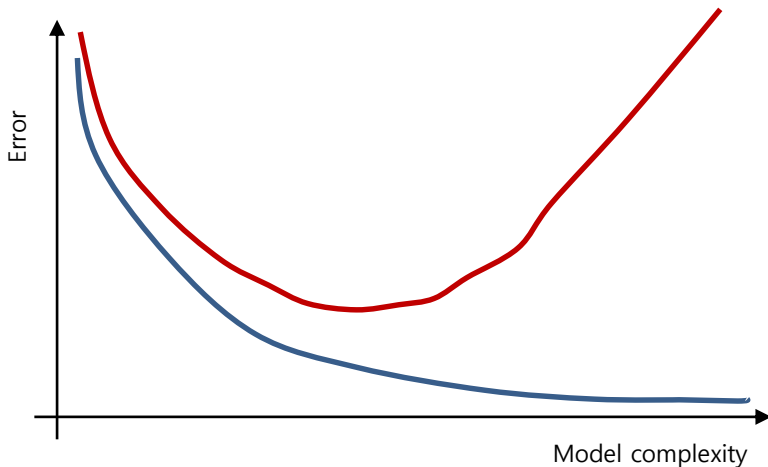
- Benchmark: 분류에 대한 **Benchmark** (Naïve rule): 모든 데이터를 가장 일반적인 클래스에 속한다고 분류
 - 일반적으로 모델의 성능이 벤치마크를 활용하는 것보다는 줄기를 기대함
 - 예외: 주어진 목표가 소수 클래스를 식별하는 것인 경우, Naïve rule보다 좋지 않은 규칙을 도입함으로써 더 좋은 성능을 낼 수도 있음
- **Prediction Benchmark**: Mean “학습데이터들의 평균값을 예측값으로 사용”

에러를 평가하기 위한 지표

- Error란: 예측값과 실제값의 차이
 - $e_i = \hat{y}_i - y_i$
- 평균 절대 오차(Mean Absolute Error: MAE (or MAD))
 - $1/n \sum_{i=1}^n |e_i|$
- 평균 오차(Average Error: AE)
 - $1/n \sum_{i=1}^n e_i$
- 평균 절대 백분율 오차(Mean Absolute Percentage Error: MAPE)
 - $100 \times 1/n \sum_{i=1}^n |e_i/y_i|$
- 평균 제곱근 오차(Rooted Mean Squared Error: RMSE)
 - $\sqrt{1/n \sum_{i=1}^n e_i^2}$
- 오차 제곱 합(Sum of Squared Error: SSE)
 - $\sum_{i=1}^n e_i^2$

훈련(Training) vs. 검증(Validation)

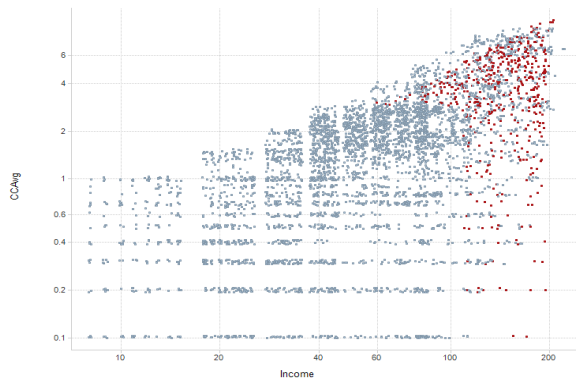
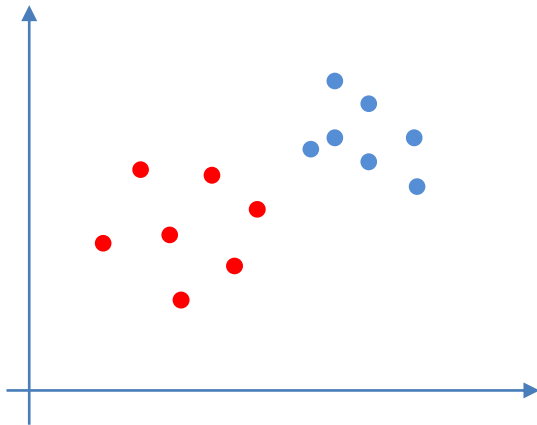
- 지도학습의 경우, 두 가지 오차의 평가를 통해 학습을 진행
 - 훈련 오차(Training error)
 - 검증 오차(Validation error)



분류 문제 해결을 위한 데이터 분리

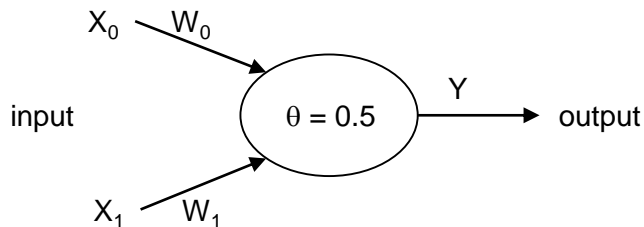
“높은 정확도의 데이터 분리(High separation of records)”는
예측 변수를 사용하여 분류 시, 높은 정확도를 달성하는 것

“낮은 정확도의 데이터 분리(Low separation of records)”는
예측 변수를 사용하는 것이 Naïve rule 대비 크게 향상되지 않음을 의미



선형 분리 가능성(Linearly Separable)

- And / XOR Gate 를 통해 선형 분리가 가능한가?



input		Output (by f)		
X_0	X_1	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

- AND

$$0 \times W_0 + 0 \times W_1 = 0 < 0.5$$

$$0 \times W_0 + 1 \times W_1 = W_1 < 0.5$$

$$1 \times W_0 + 0 \times W_1 = W_0 < 0.5$$

$$1 \times W_0 + 1 \times W_1 = W_0 + W_1 > 0.5$$

→ W_0, W_1 : 0.3 or 0.4

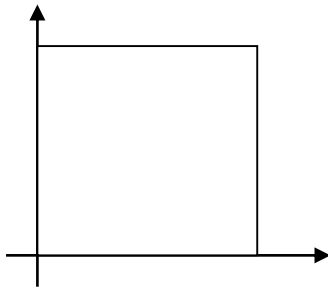
- XOR

$0 \times W_0 + 0 \times W_1 = 0$	< 0.5
$0 \times W_0 + 1 \times W_1 = W_1$	> 0.5
$1 \times W_0 + 0 \times W_1 = W_0$	> 0.5
$1 \times W_0 + 1 \times W_1 = W_0 + W_1$	< 0.5

→ W_0, W_1 do not exist that satisfy above

→ cannot solve XOR

XOR function



혼동행렬(Confusion Matrix)

- 201 “1”을 “1”로 올바르게 분류 ($n_{1,1}$)
- 85 “1”을 “0”으로 오분류 ($n_{1,0}$)
- 25 “0”을 “1”로 오분류 ($n_{0,1}$)
- 2689 “0”을 “0”으로 올바르게 분류 ($n_{0,0}$)

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

에러율(Error Rate)

전체 에러율 = $(25+85)/3000 = 3.67\%$

정확도 = $1 - \text{err} = (201+2689) = 96.33\%$

분류해야 할 클래스가 여러 개인 경우, 에러율은 아래와 같이 계산
(오분류 된 데이터 수)/(전체 데이터 수)

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	201	85
0	25	2689

$$\text{Error rate, err} = \frac{(n_{0,1} + n_{1,0})}{n}, \text{ accuracy} = 1 - \text{err}$$

분류 컷-오프(Cutoff for classification)

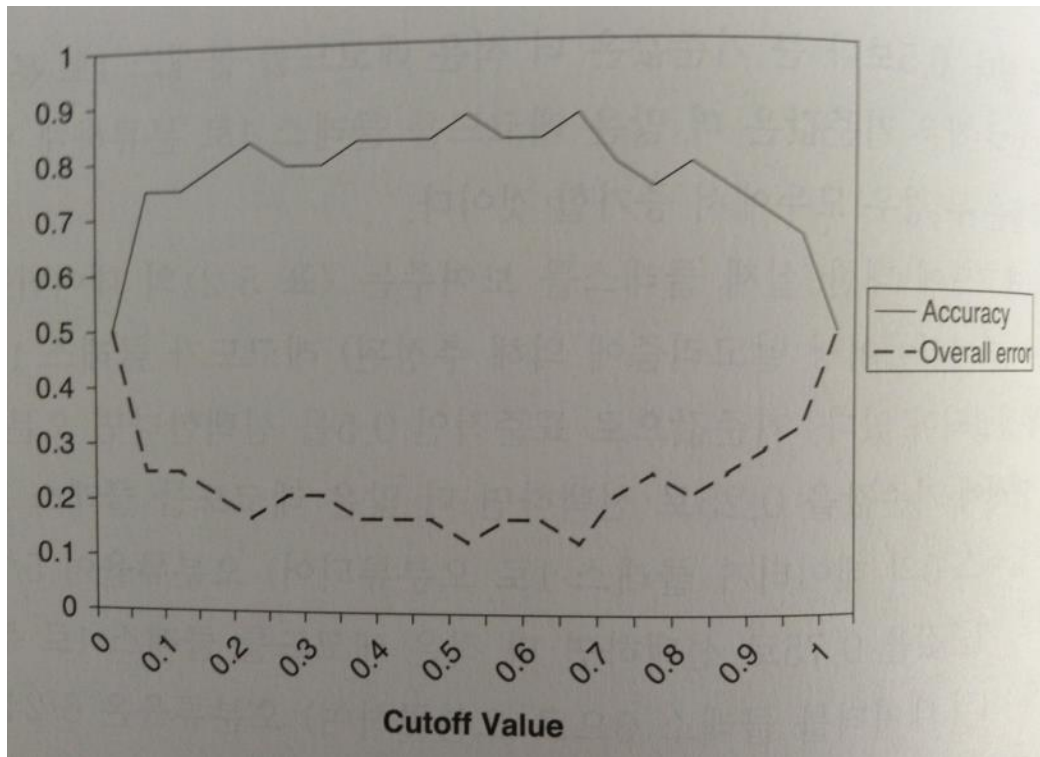
대부분의 데이터마이닝 알고리즘은 2 단계 프로세스를 통해 분류를 수행함

1. 각각의 데이터에 대해 클래스 “1”에 속할 확률 계산
 2. 컷 오프 값을 기준삼아 클래스 분류
- 기본 컷오프 값은 0.5
 - 클래스에 속할 확률이 0.5 이상이라면 “1” 로 분류
 - 클래스에 속할 확률이 0.5 미만이라면 “0”으로 분류
 - 컷 오프 값은 사용자가 정의할 수 있음
 - 일반적으로 에러율은 컷오프 = 0.5 일 때 가장 낮음

컷 오프 테이블(Cutoff Table)

- Cutoff = 0.5 인 경우, 13개의 데이터를 “1”로 분류 (에러율 = ?)
- Cutoff = 0.8 인 경우, 7개의 데이터를 “1”로 분류

Actual Class	Prob. of "1"	Actual Class	Prob. of "1"
1	0.996	1	0.506
1	0.988	0	0.471
1	0.984	0	0.337
1	0.980	1	0.218
1	0.948	0	0.199
1	0.889	0	0.149
1	0.848	0	0.048
0	0.762	0	0.038
1	0.707	0	0.025
1	0.681	0	0.022
1	0.656	0	0.016
0	0.622	0	0.004



리프트(Lift)

분류 문제에서 특정 클래스의 중요도가 높을 때

“회사가 파산할지를 예측하는 것이 지불능력을 유지할 지를 예측하는 것보다 더 중요하다.”

- 세금 사기
- 신용 불이행
- 지연된 항공편 예측

위와 같은 예시에서, 더욱 주의가 필요한 클래스를 잘 식별하기 위해 더 큰 전체 에러율을 감수할 수 있음

상황에 따른 정확도 측정 지표

“C₁” 을 올바르게 분류하는 것이 중요하다면

Sensitivity(민감도) = 클래스 C₁ 이 올바르게 분류된 비율(%)

$$\frac{n_{1,1}}{(n_{1,0} + n_{1,1})}$$

Specificity(특이도) = 클래스 C₀ 가 올바르게 분류된 비율(%)

$$\frac{n_{0,0}}{(n_{0,0} + n_{0,1})}$$

False positive rate = 실제 C₁ 이 아니지만, C₁ 으로 분류된 비율(%)

$$\frac{n_{0,1}}{(n_{0,0} + n_{0,1})}$$

False negative rate = 실제 C₀ 가 아니지만, C₀ 으로 분류된 비율(%)

$$\frac{n_{1,0}}{(n_{1,0} + n_{1,1})}$$

정밀도(Precision) 와 재현율(Recall)

- 정밀도(Precision)
 - 모델이 클래스 “1” 이라고 분류한 데이터 중 실제 클래스가 “1”인 데이터의 비율
- 재현율(Recall)
 - 실제 클래스가 “1”인 데이터 중에서 모델이 “1”이라고 분류한 데이터의 비율
- F1 score
 - 정밀도와 재현율의 조화 평균

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

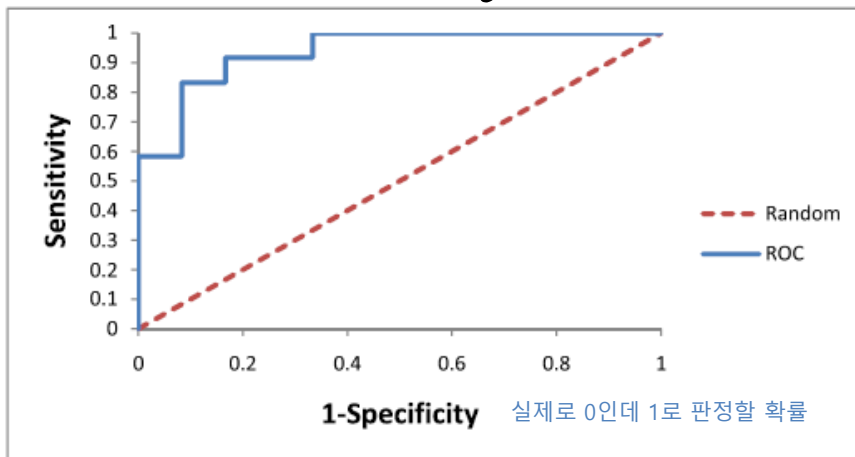
		True condition	
		True	False
Predicted condition	True	True Positive	False Positive
	False	False Negative	True Negative

ROC 커브(ROC Curve)

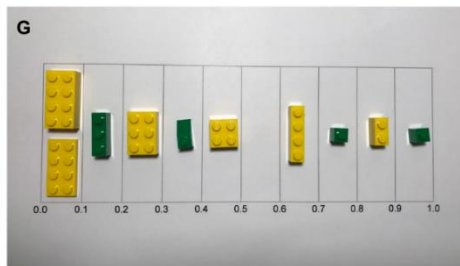
- Random하게 뽑으면?

0 안에 1로 판정할 확률과
1 안에 1로 판정할 확률이 같다!

실제로 1인
데 1로 판
정할 확률

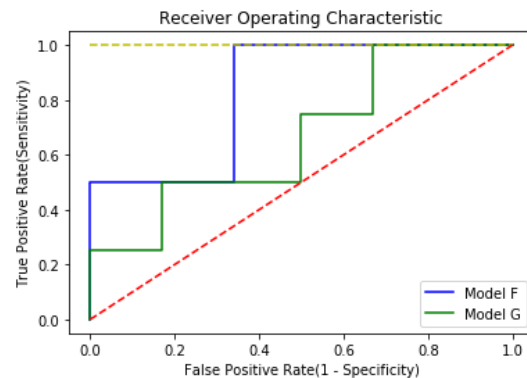


실제로 0인데 1로 판정할 확률



홀수 블록 임계값	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
맞춘 홀수(전체4개)	4	4	4	4	3	2	2	2	2	2	0
맞춘 짝수(전체6개)	0	1	3	4	4	4	5	6	6	6	6
정확도	40%	50%	70%	80%	70%	60%	70%	80%	80%	80%	60%
민감도	100%	100%	100%	100%	75%	50%	50%	50%	50%	50%	0%
특이도	0%	16.6%	50%	66.6%	66.6%	66.6%	83.3%	100%	100%	100%	100%

홀수 블록 임계값	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
맞춘 홀수(전체4개)	4	4	3	3	2	2	2	2	1	1	0
맞춘 짝수(전체6개)	0	2	2	3	3	4	4	5	5	6	6
정확도	40%	60%	50%	60%	50%	60%	60%	70%	60%	70%	60%
민감도	100%	100%	75%	75%	50%	50%	50%	50%	25%	25%	0%
특이도	0%	33.3%	33.3%	50%	50%	66.6%	66.6%	83.3%	83.3%	100%	100%



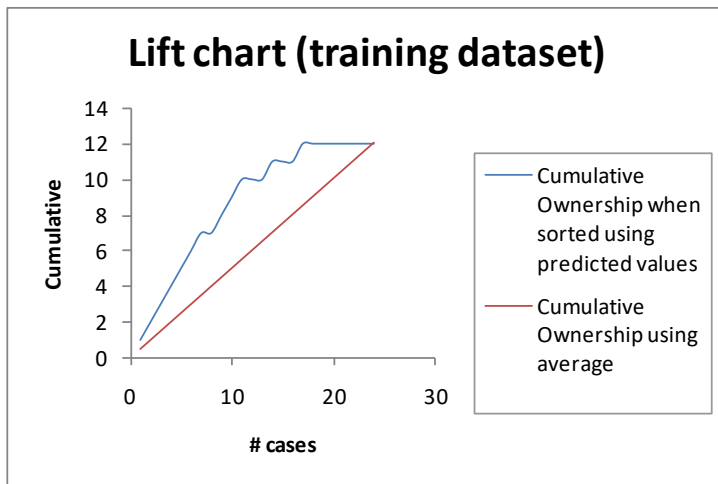
Lift and Decile(십분위수) Charts

Lift와 Decile Chart는 특정 클래스(중요도가 높은 클래스)를 잘 분류해야 하는 상황에서 성능을 평가하는 데 유용함

- 조사해야 할 세금 기록 수
- 대출을 승인해 줄 고객
- 메일을 보내야 할 고객 수

Lift Chart – cumulative performance

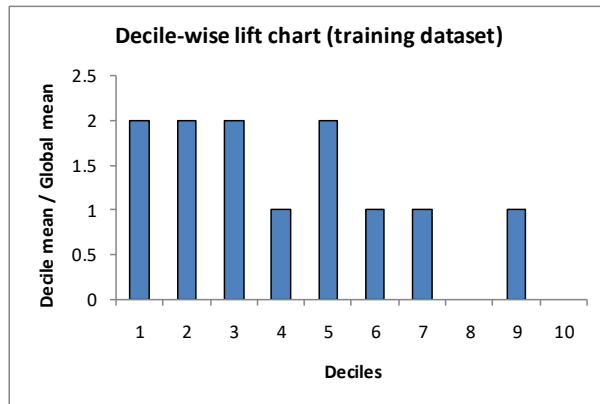
좋은 분류기(분류 모델)는 적은수의 데이터만으로도 높은 성능 향상을 보임



Actual Class	Prob. of "1"	Actual Class	Prob. of "1"
1	0.996	1	0.506
1	0.988	0	0.471
1	0.984	0	0.337
1	0.980	1	0.218
1	0.948	0	0.199
1	0.889	0	0.149
1	0.848	0	0.048
0	0.762	0	0.038
1	0.707	0	0.025
1	0.681	0	0.022
1	0.656	0	0.016
0	0.622	0	0.004

십분위수 차트(Decile Chart)

In “most probable” (top) decile, model is twice as likely to identify the important class (compared to avg. prevalence)



Lift vs. Decile Charts

Both embody concept of “moving down” through the records, starting with the most probable

Decile chart does this in decile chunks of data

Y axis shows ratio of decile mean to overall mean

Lift chart shows continuous cumulative results

Y axis shows number of important class records identified

Asymmetric Costs

오분류에 대한 비용(평가)는 달라질 수 있음

오분류에 대한 비용은 특정 클래스가 다른 클래스들의 비해 높을 수 있음
(중요도가 다를 수 있음)

다른 관점에서, 올바른 분류를 하는 것에 대한 중요도가 클래스 별로 다를 수 있음

예제 – 프로모션 제안에 대한 응답 상황

평균 응답률이 1%인 1000명에게 프로모션 제안 메일을 보낸다고 가정
(“1” = 응답함, “0” = 응답하지 않음)

- “Naïve rule”에 따르면 (모든 사람을 “0”으로 분류) 전체 에러율은 1%
- 데이터마이닝 기법을 사용하여 8개의 “1” 클래스를 “1”로 올바르게 분류할 수 있음
20개의 “0”을 “1”로, 2개의 “0”을 “1”로 오분류하는 비용이 발생

혼동행렬(Confusion Matrix)

에러율(Error rate) = $(2+20) = 2.2\%$ (naïve rule 보다 좋은 성능)

	Predict as 1	Predict as 0
Actual 1	8	2
Actual 0	20	970

오분류에 대한 비용 및 이득 개념 도입

가정:

- 클래스 “1”을 잘 분류했을 때의 이득: \$10
- 제안 메일을 발송하는데 드는 비용: \$1

Then:

- naïve rule에 따르면, 모두 “0”으로 분류하므로 비용이 발생하지 않음
 - 이득도 발생하지 않음
- 데이터마이닝 기법 활용시, 28개의 제안 메일 발송
 - 8명의 응답으로 인해 $8 * \$10 = \80 이득 발생
 - 20명이 응답하지 않았으므로 \$20 비용 발생
 - 972명에 대해서는 아무런 행동을 하지 않음(이득, 비용 모두 발생하지 않음)
- 최종 이득 = \$60

Profit Matrix

	Predict as 1	Predict as 0
Actual 1	\$80 8	0 2
Actual 0	(\$20) 20	0 990

Lift (again)

Adding costs to the mix, as above, does not change the actual classifications

Better: Use the lift curve and change the cutoff value for “1” to maximize profit

참고: 기회비용

- 비용과 이익을 각각 고려하는 것보다 모든 것을 비용으로 계산하는 것이 가장 좋음
- 판매 이익 대신 판매 손실의 기회 비용으로 생각

비용 및 이득(편익) 고려한 Lift Curve

- Sort records in descending probability of success
- For each case, record cost/benefit of actual outcome
- Also record cumulative cost/benefit
- Plot all records
 - X-axis is index number (1 for 1st case, n for nth case)
 - Y-axis is cumulative cost/benefit
 - Reference line from origin to y_n (y_n = total net benefit)

Lift Curve May Go Negative

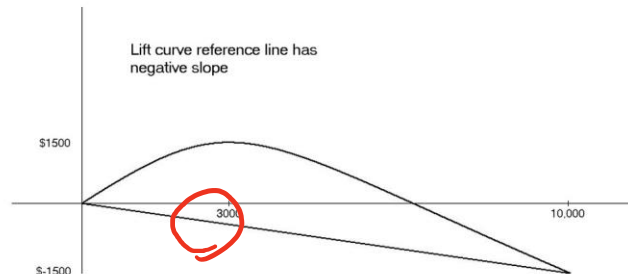
If total net benefit from all cases is negative, reference line will have **negative slope**

Nonetheless, goal is still to use cutoff to select the point where net benefit is at a maximum

Negative slope to reference curve

Cost for sending one mail 0.65\$, benefit from the respondent 25\$, response rate 2%,

* If we send to 10,000 people? $(0.02 * \$25 * 10,000) - (0.65 * 10,000) = -1500$



Summary

- Model evaluation
 - Evaluation metrics are important for comparing across DM models, for choosing the right configuration of a specific DM model, and for comparing to the baseline
 - Major metrics: confusion matrix, error rate, predictive error
 - Other metrics when
 - one class is more important
 - asymmetric costs