12주차:
모델의 성능 평가

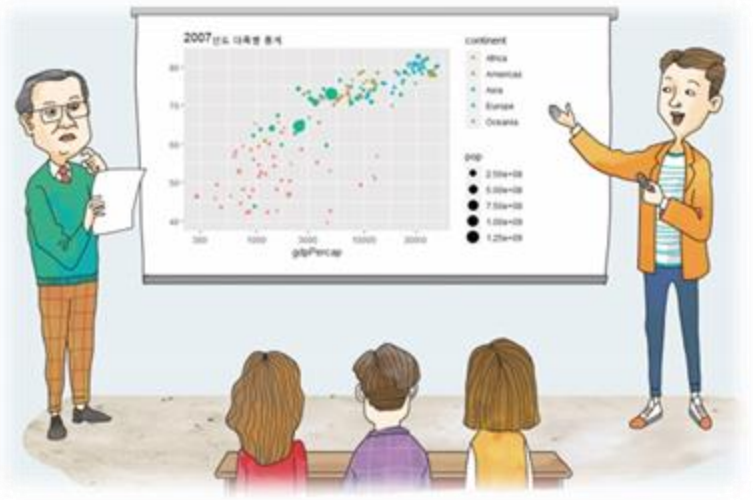
ChulSoo Park

School of Computer Engineering & Information Technology

Korea National University of Transportation

10 CHAPTER

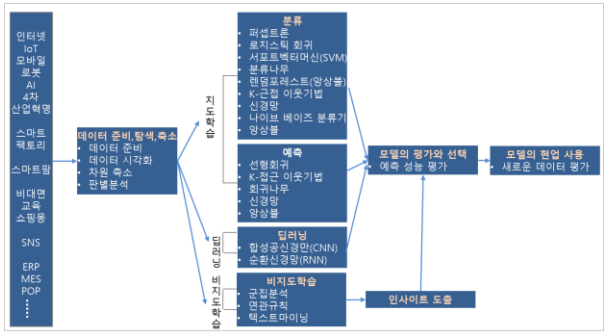
모델의 성능 평가



CONTENTS

10.5 모델 선택

모델의 성능 평가



데이터 사이언스에서 **데이터 준비,전처리, 모델링** 등은 작업 파이프라인에서 핵심적인 단계로 생각하지만, 학습된 모델의 **성능을 평가**하는 것 또한 중요한 단계이다.

모델링 결과를 평가 시 고려할 3가지 기준

모델링: 모델링은 이용가능 한 데이터에 대한 정확한 예측이 아닌, 미래(새로운) 데이터에 대한 정확한 예측을 고려하여 진행되어야 함.

과적합: 통제된 데이터 속에서 Memorization 상태

일반화(Generalization): 모델의 능력은 학습에 사용되지 않았던 데이터를 적용해 본 결과로 정의

10.5 모델 선택

평가지표(evaluation metrics)

분류(classification), 회귀(regression), 랭킹, 군집화(clustering), 토픽모델링(topic modeling) 등 각각의 모델마다 적절한 평가지표는 다르지만, Accuracy, Precision-recall과 같이 여러 모델링에 일반적으로 유용하게 쓰이는 지표들이 있다.

1. 정확도(Model Accuracy) :

분류 모델 측면에서 모델 정확도는 전체 표본 중 **정확히 분류된 표본의 수**

Accuracy = Number of correct predictions / Total number of predictions

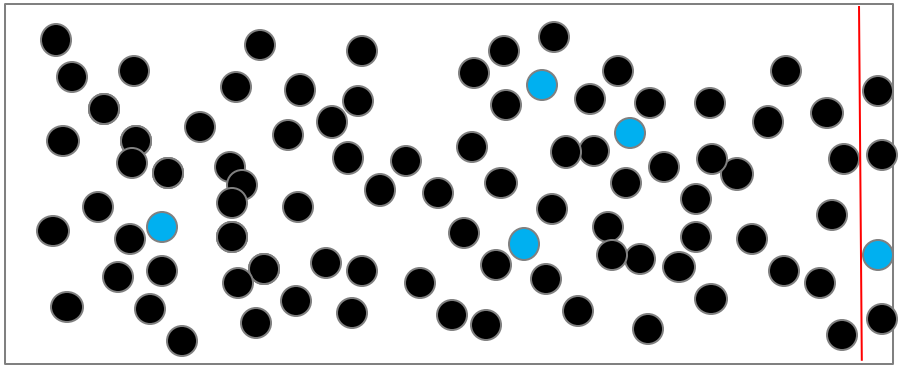
Or for binary classification models, the accuracy can be defined as:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Actual \ Predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Confusion Matrix

True Positive (TP) : 모델이 정답(Positive)을 맞추었을 때
True Negative (TN) : 모델이 오답(Negative)을 맞추었을 때
False Positive (FP) : 모델이 오답(Negative)을 정답(Positive)으로 잘못 예측했을 때
False Negative (FN) : 모델이 정답(Positive)을 오답(Negative)으로 잘못 예측했을 때



10.5 모델 선택

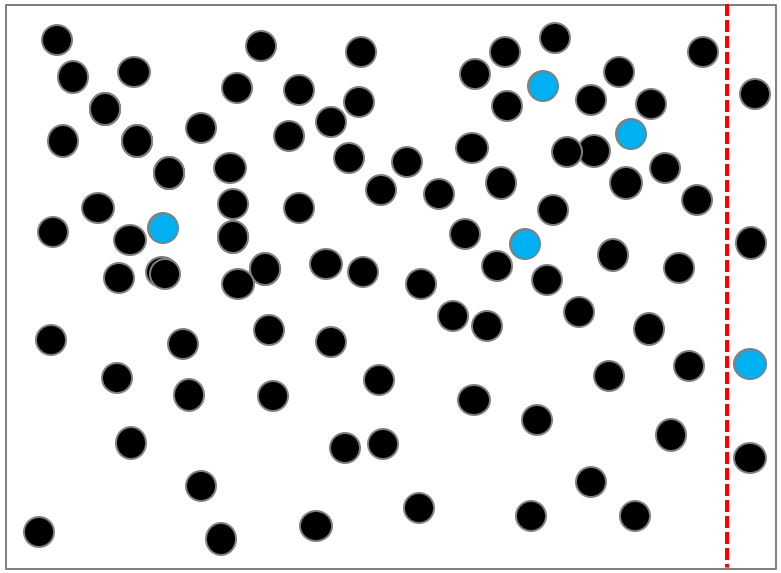
평가지표(evaluation metrics)

1. 정확도(Model Accuracy) :

분류 모델 측면에서 모델 정확도는 전체 표본 중 **정확히 분류된 표본의 수**

세상에 암환자는 0.1 %로 추정 (10000면 중 10명)

암환자 진단 정확도



의사의 정확도 : 99.9

철수의 정확도 : 99.8

부류 불균형 : 한 부류는 발 생 빈도가 높고 다른 한 쪽은 낮은 것. 이 때 정밀도와 재현율 사용
예) 환자,불량품,거부 등

부류 균형일 때 오류 형태를 자세히 따져야 하는 경우는 혼돈 행렬 사용

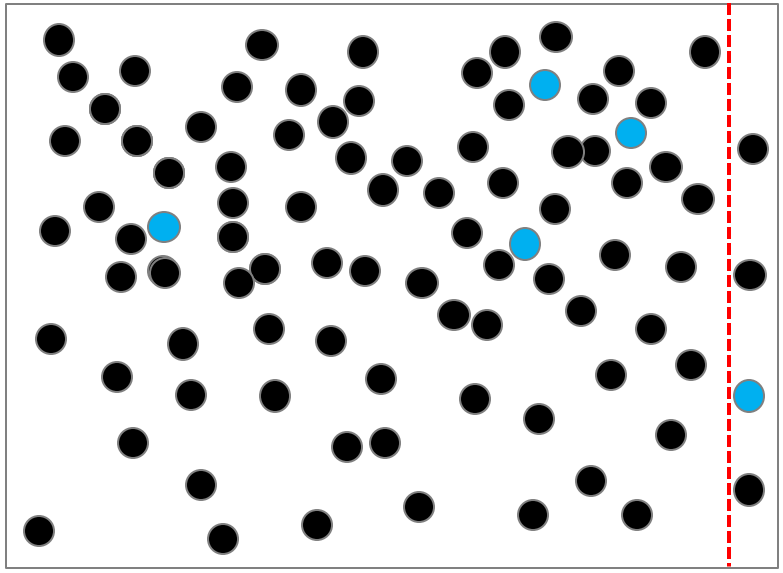
10.5 모델 선택

2. 정밀도(Precision) and 재현율(Recall)

정밀도 : 정답을 정답이라고 맞춘TP(True Positive)개수를 TP+FP(False Positive, 정답을 오답이라고 판단한 개수)로 나누는 것.

$$\text{정밀도(Precision)} = \frac{\text{TP}}{\text{TP+FP(정상인을 암환자로 오진)}}$$

암환자 진단 정밀도



의사의 정밀도 : 85%

철수의 정밀도 : 2%

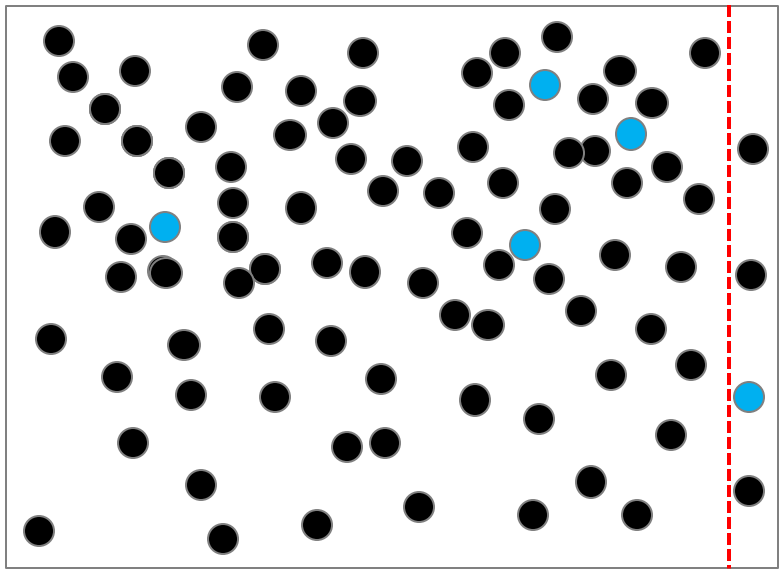
10.5 모델 선택

2. 정밀도(Precision) and 재현율(Recall)

재현률 : 정답을 정답이라고 맞춘 TP(True Positive)개수를 TP+FN로 나누는 것.

$$\text{재현률(Recall)} = \frac{\text{TP}}{\text{TP+FN(암환자를 정상인으로 오진)}}$$

암환자 진단 재현율



의사의 정확도 : 87%

철수의 정확도 : 30%

모델의 효율성을 평가하기 위해 precision과 recall을 모두 검토할 필요가 있다. 불행하게도, precision과 recall은 보통 반대되는 수치를 가진다. 즉, precision을 높이고자 하면 recall이 감소하고, 그 반대도 마찬가지다.

10.5 모델 선택

- 데이터 과학 대회에서는 모델의 성능을 겨룸
  - 현장에 설치할 예측 시스템을 만들 때도 성능이 가장 좋은 모델을 찾아야함
  - 좋은 모델을 찾으려면 주어진 데이터에 여러 모델을 적용하고 성능을 상호 비교해야 함
  - 이러한 것을 모델 선택이라 함
  - 동일한 모델에서도 최적의 하이퍼 매개변수를 설정해야 함

- 이 장에서 할 일
  - 첫째, 하이퍼 매개변수는 기본값으로 두고 모델 선택
  - 둘째, 하이퍼 매개변수 최적화와 모델 선택을 동시에 수행

10.5 모델 선택

- iris 데이터에 대해 5-겹 교차 검증을 4개 모델에 적용하는 코드

4가지 모델 : 결정트리, 랜덤 포리스트, SVM,k-nn : 정확률(Accuracy)

cross-validation

```
> # iris 4개 모델 적용 5-겹 교차 검증 SVM, k-NN 등등 #
> control = trainControl(method = 'cv', number = 5)
> r = train(Species~., data = iris, method = 'rpart', metric = 'Accuracy', trControl=control)
> f = train(Species~., data = iris, method = 'rf', metric = 'Accuracy', trControl = control)
> s = train(Species~., data = iris, method = 'svmRadial', metric = 'Accuracy', trControl = control)
> k = train(Species~., data = iris, method = 'knn', metric = 'Accuracy', trControl = control)
> resamp = resamples(list(decisiontree = r, randomforest = f, SVM = s, kNN = k))
> summary(resamp)
```

10.5 모델 선택

```
Console  C:/RSources/

Call:
summary.resamples(object = resamp)

Models: decisiontree, randomforest, SVM, kNN
Number of resamples: 5

Accuracy
                Min.   1st Qu.    Median  Mean   3rd Qu.      Max.
decisiontree 0.8666667 0.9000000 0.9333333 0.92 0.9333333 0.9666667
randomforest 0.9333333 0.9333333 0.9333333 0.96 1.0000000 1.0000000
SVM          0.8666667 0.9000000 0.9666667 0.94 0.9666667 1.0000000
kNN          0.9333333 0.9666667 1.0000000 0.98 1.0000000 1.0000000
             NA's
decisiontree    0
randomforest    0
SVM             0
kNN             0

Kappa
             Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
decisiontree  0.8    0.85   0.90 0.88    0.90 0.95    0
randomforest  0.9    0.90   0.90 0.94    1.00 1.00    0
SVM           0.8    0.85   0.95 0.91    0.95 1.00    0
kNN           0.9    0.95   1.00 0.97    1.00 1.00    0

> sort(resamp, decreasing = TRUE)
[1] "kNN"          "randomforest" "SVM"          "decisiontree"
```

정확률

resamples 함수로 정리한 결과를 summary와 sort 함수로 출력

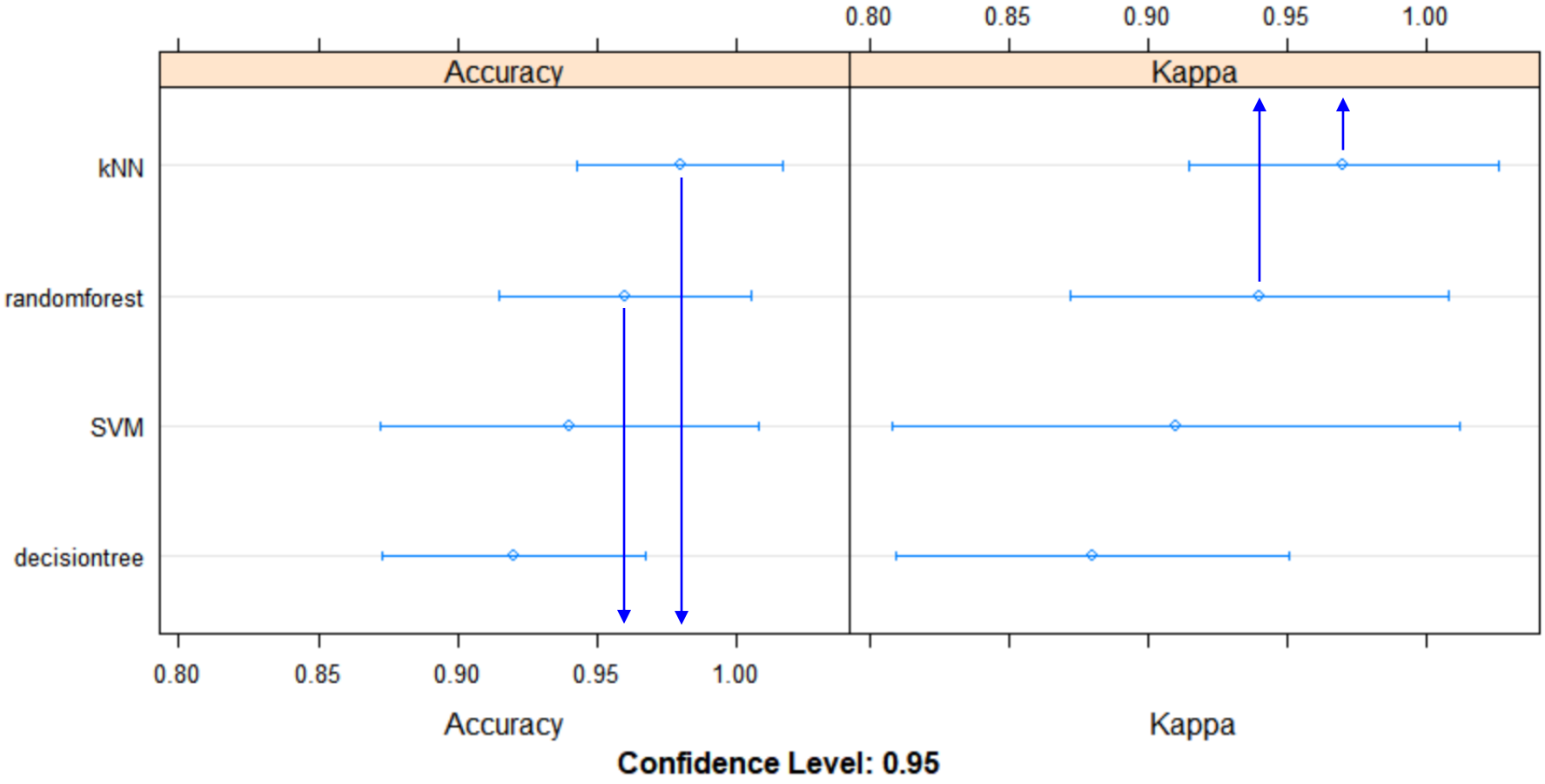
Mean (평균 정확률)을 기준으로 하면 *k*-NN이 98.0%로 가장 높고 결정 트리가 92.0%로 가장 낮음

카파(Kappa) 상관계수는 2명의 관찰자(또는 평가자)의 신뢰도를 확보하기위한 확률로서 평가지표로 사용되는 상관계수이다. 2명이상에서 신뢰도를 얻기위해서는 플레이스 카파 상관계수(Fleiss' kappa)를 사용할수있다.

10.5 모델 선택

- dotplot 함수를 적용하면

```
> dotplot(resamp)
```



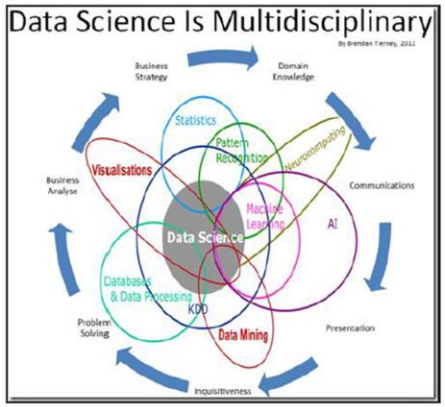
10.5 모델 선택

- 하이퍼 매개변수
  - 모델의 구조나 학습 방법을 제어하는 데 사용하는 변수 (9.5.4절 참조)
  - 예) SVM의 경우 커널 함수 종류, 랜덤 포리스트의 경우 결정 트리의 개수 등

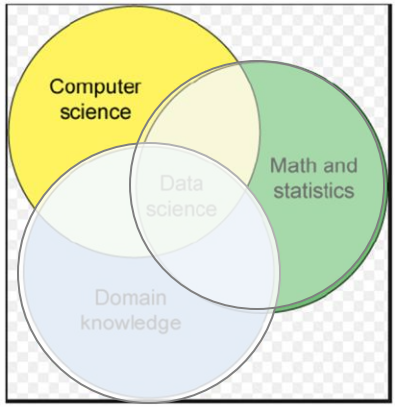
- 5가지 모델에 대한 하이퍼 매개변수 최적화
  - colon 데이터 (9.7.2절 참조)에 대해 SVM, 랜덤 포리스트, 결정 트리, *k*-NN, glm 이라는 다섯 가지 모델에 대해 모델 선택
  - SVM은 커널 함수, 랜덤 포리스트는 결정 트리 개수라는 하이퍼 매개변수를 동시에 최적화

10.5 모델 선택

- **colon cancer 데이터 살펴보기 : Domain knowledge(8장 강의 내용)**
  - 결측치가 존재하는 일반적인 colon cancer data 로지스틱 회귀 적용 실습



출처 : www.oralytics.com



데이터 사이언스 학과(대학원)

T(Tumor, 종양)인자
T0 : 종양의 근거가 없음
T1 : 점막층과 점막하층에 국한된 대장암
T2 : 고유근층까지 침습한 대장암
T3 : 장막층을 침습한 결장암 또는 직장간막층을 침습한
T4 : 인접한 다른 장기까지 침습한 대장암

N(Node, 림프절)인자
N0 : 림프절 전이 없음
N1 : 1~3개의 국소 림프절 전이
N2 : 4개 이상의 국소 림프절 전이
N3 : 비 전형적인 림프절 또는 큰 혈관 주위의 림프절 전

M(Metastasis, 원격전이)인자
M0 : 원격전이 없음
M1 : 원격전이 있음

Dukes (Astler-Coller 개정) 분류법
A기 : 점막에 국한된 대장암
B기
B1기 : 고유근층까지 침습한 대장암
B2기 : 대장암이 장막층을 뚫고 나간 상태
B3기 : 대장암이 인접한 장기에 유착되거나 침습한 상태

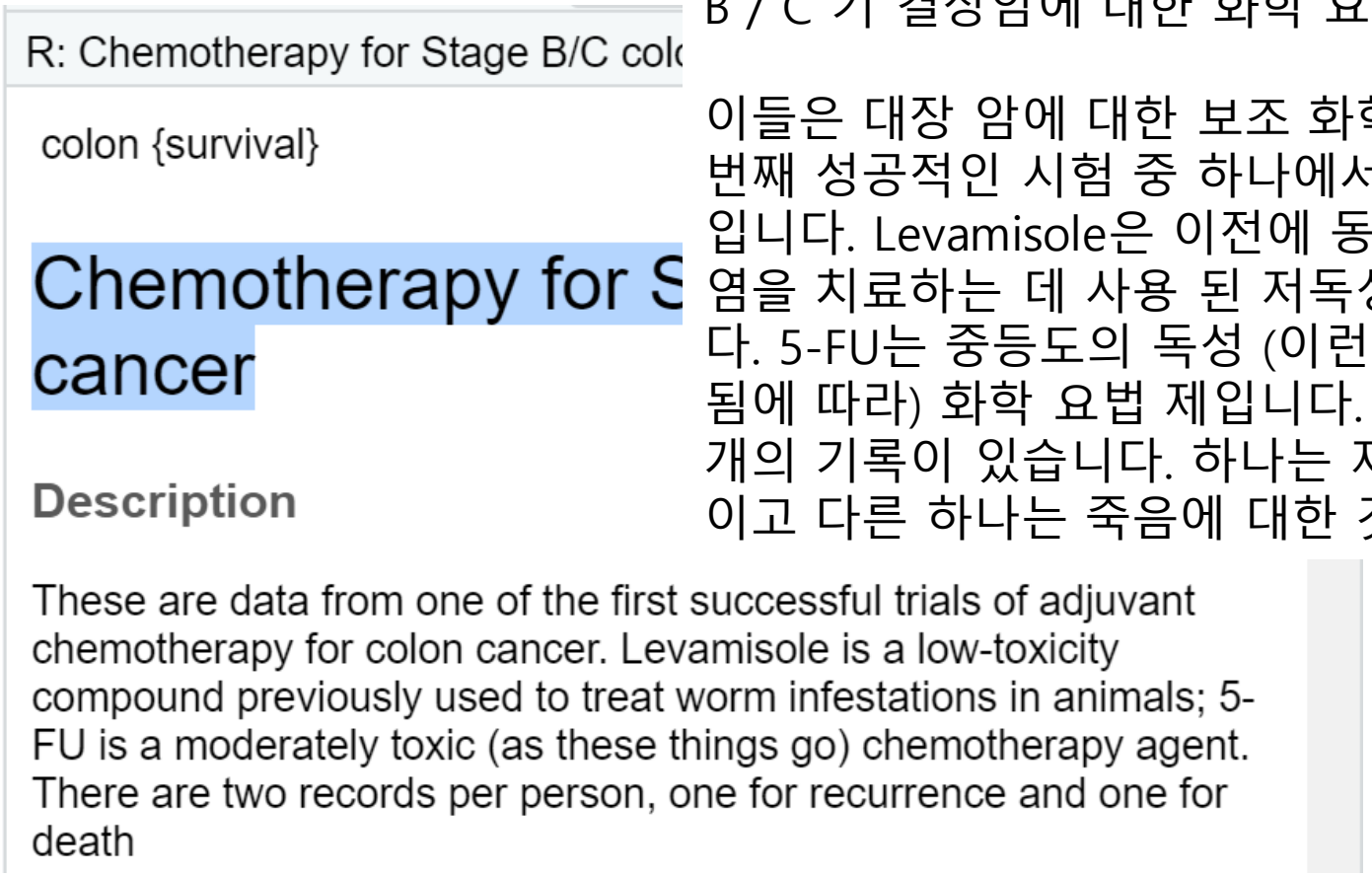
C기
C1기 : B1 + 국소 림프절 전이
C2기 : B2 + 국소 림프절 전이
C3기 : B3 + 국소 림프절 전이
D기 : 원격전이

- 대장암(colorectal cancer) = 결장암(colon cancer) + 직장암(rectal cancer)
- 대장암 진행 정도 표기법
  - ✓ AJCC(American Joint Committee on Cancer) : A,B,C,D로 표현
  - ✓ 국제표준 TNM 분류법 : 로마자로 I, II, III,IV로 표현
- 등급 판정 기준 : T인자, N인자,M인자 종합
- 5년 생존율 : A기(90%), B기(60%), C기(40%), D기(5% 미만)
- 최근 수출후 생존률을 유의하게 높일 수 있는 5-FU 효과 인정

출처 : 현대의학, 자연과학 그리고 의용공학의 세계 (https://blog.daum.net/inbio880/16096552)

10.5 모델 선택

- **colon cancer 데이터 살펴보기 : Description(8장 강의 내용)**
  - 결측치가 존재하는 일반적인 colon cancer data 로지스틱 회귀 적용 실습



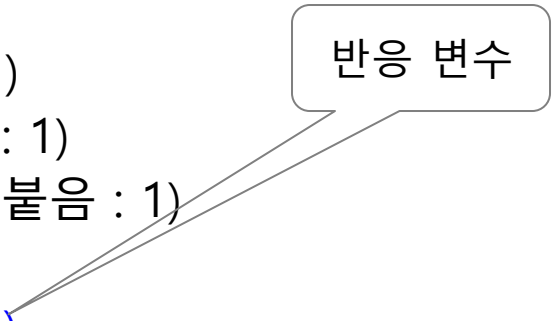
B / C 기 결장암에 대한 화학 요법 기술

이들은 대장 암에 대한 보조 화학 요법의 첫 번째 성공적인 시험 중 하나에서 얻은 데이터입니다. Levamisole은 이전에 동물의 벌레 감염을 치료하는 데 사용 된 저독성 화합물입니다. 5-FU는 중등도의 독성 (이런 일들이 진행됨에 따라) 화학 요법 제입니다. 한 사람당 두 개의 기록이 있습니다. 하나는 재발에 대한 것이고 다른 하나는 죽음에 대한 것입니다

10.5 모델 선택

colon cancer 데이터 살펴보기 : Description of 16개 변수(8장 강의 내용)

- id : 환자 번호
- study : 모든 샘플이 1(모두 조사에 참여)
- rx : 치료 방법(Observation, Levamisole, Levamisole+5-FU)
- sex : 성별(여성 : 0, 남성 : 1)
- age : 나이
- obstruct : 결장의 폐쇄 여부(폐쇄 안 됨 : 0, 폐쇄 : 1)
- perfor : 결장의 구멍 여부(구멍 없음 : 0, 구멍 있음 : 1)
- adhere : 인접 장기와 붙었는지 여부(붙지 않음 : 0, 붙음 : 1)
- nodes : 암세포가 있는 림프절의 수
- status : 재발/사망 여부 (완치 : 0, 재발 또는 사망 :1)
- differ : 암세포의 조직학적 분화 정도(well : 1, moderate : 2, poor :3)
- extent : 암세포가 침습한 깊이(submucosa:1, muscle:2, serosa: , 인접 장기:4
- surg : 수술 후 등록기까지의 기간 (short : 0, long :1)
- node4 : 양성 림프절 수가 4개 이상인지 여부(4개 미만 :0, 4개 이상 :1)
- time : etype까지의 일수
- etype : 재발 또는 사망(재발 : 1, 사망 :2)



10.5 모델 선택

- colon cancer 데이터 살펴보기 : Description of 16개 변수

Console C:/RSources/

```
> str(colon)
'data.frame':   1858 obs. of  16 variables:
 $ id       : num  1 1 2 2 3 3 4 4 5 5 ...
 $ study    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ rx       : Factor w/ 3 levels "Obs","Lev","Lev+5FU": 3 3 3 3 1 1 3 3 1 1 ...
 $ sex      : num  1 1 1 1 0 0 0 0 1 1 ...
 $ age      : num  43 43 63 63 71 71 66 66 69 69 ...
 $ obstruct: num  0 0 0 0 0 0 1 1 0 0 ...
 $ perfor   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ adhere   : num  0 0 0 0 1 1 0 0 0 0 ...
 $ nodes    : num  5 5 1 1 7 7 6 6 22 22 ...
 $ status   : num  1 1 0 0 1 1 1 1 1 1 ...
 $ di
 $ ex
 $ su
 $ nc
 $ ti
 $ et
```

```
> head(colon,10)
   id study       rx sex age obstruct perfor adhere nodes status differ extent surg node4 time etype
1   1     1 Lev+5FU   1  43        0      0      0     5      1      2      3    0     1 1521     2
2   1     1 Lev+5FU   1  43        0      0      0     5      1      2      3    0     1  968     1
3   2     1 Lev+5FU   1  63        0      0      0     1      0      2      3    0     0 3087     2
4   2     1 Lev+5FU   1  63        0      0      0     1      0      2      3    0     0 3087     1
5   3     1     Obs   0  71        0      0      1     7      1      2      2    0     1  963     2
6   3     1     Obs   0  71        0      0      1     7      1      2      2    0     1  542     1
7   4     1 Lev+5FU   0  66        1      0      0     6      1      2      3    1     1  293     2
8   4     1 Lev+5FU   0  66        1      0      0     6      1      2      3    1     1  245     1
9   5     1     Obs   1  69        0      0      0    22      1      2      3    1     1  659     2
10  5     1     Obs   1  69        0      0      0    22      1      2      3    1     1  523     1
```

10.5 모델 선택

- 모델 선택하고 하이퍼 매개변수 최적화하기: colon 데이터

Console Jobs

C:/RSources/

```
> library(survival)
> clean_colon = na.omit(colon)
> clean_colon = clean_colon[c(TRUE, FALSE), ]
> clean_colon$status = factor(clean_colon$status)
> # ------------------------------------------------------
> control = trainControl(method = 'cv', number = 10)
> formular = status~rx+sex+age+obstruct+perfor+adhere+nodes+differ+extent+surg+node4
> # ------------------------------------------------------
> L = train(formular, data = clean_colon, method = 'svmLinear', metric = 'Accuracy', trControl = control)
> LW = train(formular, data = clean_colon, method = 'svmLinearWeights', metric = 'Accuracy', trControl = control)
> P = train(formular, data = clean_colon, method = 'svmPoly', metric = 'Accuracy', trControl =  control)
> R = train(formular, data = clean_colon, method = 'svmRadial', metric = 'Accuracy', trControl = control)
> RW = train(formular, data = clean_colon, method = 'svmRadialWeights', metric = 'Accuracy', trControl = control)
> f100 = train(formular, data = clean_colon, method = 'rf', ntree = 100, metric = 'Accuracy', trControl = control)
> f300 = train(formular, data = clean_colon, method = 'rf', ntree = 300,metric = 'Accuracy', trControl = control)
> f500 = train(formular, data = clean_colon, method = 'rf', ntree = 500,metric = 'Accuracy', trControl = control)
> r = train(formular, data = clean_colon, method = 'rpart', metric = 'Accuracy', trControl = control)
> k = train(formular, data = clean_colon, method = 'knn', metric = 'Accuracy', trControl = control)
> g = train(formular, data = clean_colon, method = 'glm', metric = 'Accuracy', trControl = control)
> # ------------------------------------------------------
> resamp = resamples(list(linear_regression = L, lr_Weight = LW, polynomial = P, RBF = R, Weight = RW, rf100 = f10
0, rf300 = f300, rf500 = f500, tree = r, knn = k, glm = g))
> summary(resamp)
```

10.5 모델 선택

- 모델 선택하고 하이퍼 매개변수 최적화하기: colon 데이터

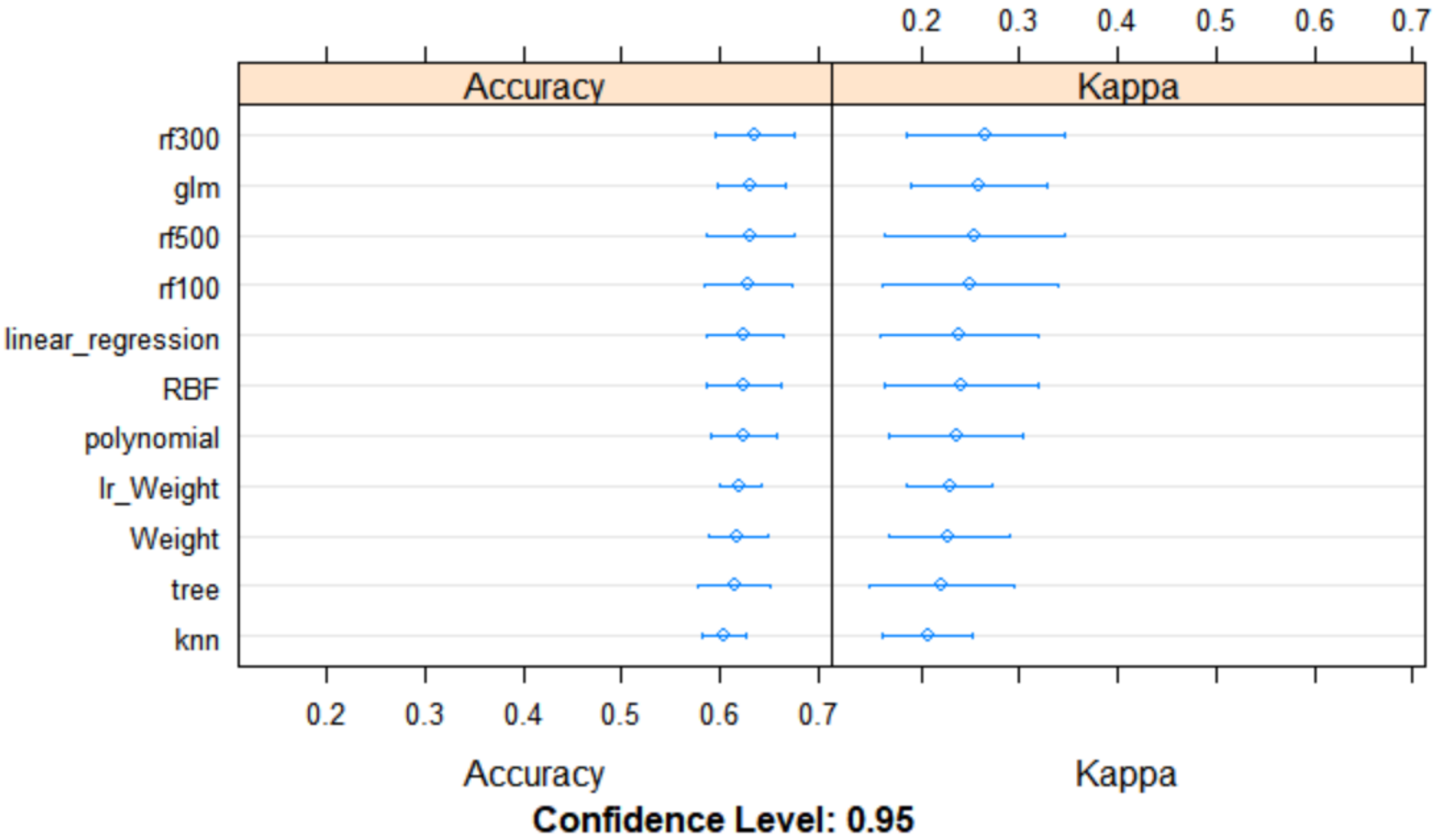
Console C:/RSources/

```
> summary(resamp)

Call:
summary.resamples(object = resamp)

Models: linear_regression, lr_Weight, polynomial, RBF, Weight, rf100, rf300, rf500, tree, knn, glm
Number of resamples: 10

Accuracy
                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
linear_regression 0.5280899 0.6004852 0.6327886 0.6249745 0.6404494 0.7191011    0
lr_Weight         0.5730337 0.6011236 0.6179775 0.6204801 0.6345761 0.6741573    0
polynomial        0.5568182 0.5898876 0.6179775 0.6238253 0.6430988 0.7078652    0
RBF               0.5393258 0.5898876 0.6292135 0.6239658 0.6676136 0.6853933    0
Weight            0.5393258 0.5948672 0.6235955 0.6182584 0.6488764 0.6741573    0
rf100             0.5280899 0.5955056 0.6327886 0.6284857 0.6713483 0.7159091    0
rf300             0.5168539 0.6260534 0.6441522 0.6362743 0.6713483 0.6966292    0
rf500             0.5393258 0.5747574 0.6404494 0.6306818 0.6856167 0.7078652    0
tree              0.5617978 0.5730337 0.6022727 0.6148366 0.6629213 0.6966292    0
knn               0.5617978 0.5920582 0.6011236 0.6047370 0.6151685 0.6741573    0
glm               0.5393258 0.6147217 0.6348315 0.6317799 0.6732316 0.6853933    0
Kappa
                     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
linear_regression 0.03610108 0.1903078 0.2568966 0.2387912 0.2687120 0.4319632    0
lr_Weight         0.13193018 0.1864546 0.2239031 0.2295828 0.2653814 0.3359918    0
polynomial        0.10251046 0.1684734 0.2233546 0.2364607 0.2785797 0.4078813    0
RBF               0.07268107 0.1679900 0.2502675 0.2411227 0.3332061 0.3662258    0
Weight            0.06841971 0.1798680 0.2396997 0.2289153 0.2900932 0.3450901    0
rf100             0.04642857 0.1835871 0.2633673 0.2503752 0.3352889 0.4258873    0
rf300             0.02892667 0.2462885 0.2834353 0.2660342 0.3337477 0.3893266    0
rf500             0.06984451 0.1398896 0.2750146 0.2543548 0.3673214 0.4123921    0
tree              0.10702341 0.1392077 0.1989678 0.2210114 0.3146277 0.3874586    0
knn               0.12193271 0.1788125 0.1992386 0.2067144 0.2278653 0.3500378    0
glm               0.06984451 0.2259558 0.2652447 0.2582878 0.3384890 0.3681542    0

> sort(resamp, decreasing = TRUE)
 [1] "rf300"             "glm"               "rf500"             "rf100"             "linear_regression"
 [6] "RBF"               "polynomial"        "lr_Weight"         "Weight"            "tree"
[11] "knn"
```

Mean (평균 정확률)을 기준으로 하면
rf300이 63.62%로 가장 높고
k-NN이 60.47%로 가장 낮음

10.5 모델 선택

- 모델 선택하고 하이퍼 매개변수 최적화하기: colon 데이터



10.5 모델 선택

- 실제 상황에서 하이퍼 매개변수 최적화는 훨씬 복잡

  - 대부분 경우 하이퍼 매개변수가 아주 많음
    - ✓ 예) SVM의 경우 커널 함수로 다항식 커널을 선택했다면 몇 차 다항식을 사용할지 지정해야 함. 또한 C를 얼마로 설정할 지 정해야 함(C : 여백과 잘 못 분류하는 샘플 수)
  - 하이퍼 매개변수의 모든 조합을 고려한다면 매우 시간이 많이 걸리는 작업
    - ✓ 왜냐하면 r개의 하이퍼 매개변수가 있는데 각각 q개의 서로 다른 값을 조사한다면 q^r 개의 조합이 발생
  - 실제적으로는 함수가 제공하는 기본값으로 성능을 측정한 다음, 하이퍼 매개변수 각각을 조금씩 변화시키며 최적 설정을 찾음

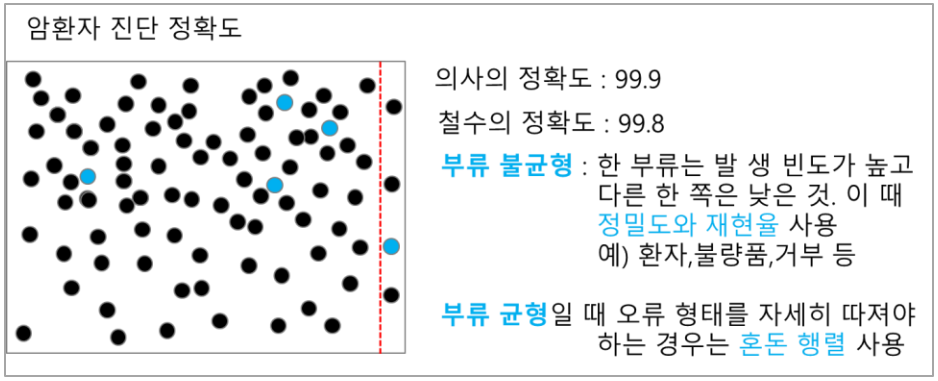
10.6 정밀도와 재현율

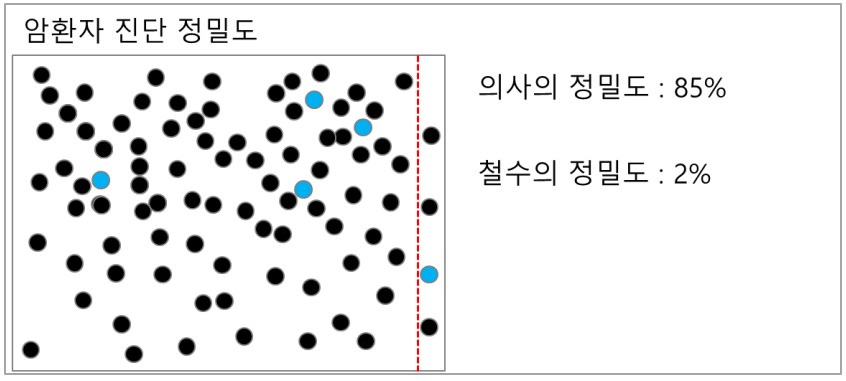
정확률이 의미가 없는 상황

- 예) 1000명당 1명꼴로 암환자라면 의사가 무턱대고 정상이라고 판정해도 정확률이 99.9%(오진율은 0.1%)인 명의가 됨
- 부류 불균형인 상황에서는 다른 성능 척도를 사용해야 함

이 절에서는

- 정상인과 환자, 정상품과 불량품, 승인과 거부처럼 2부류 분류 문제에서 모델의 성능을 보다 세밀하게 측정해 주는 척도로 정밀도와 재현율을 공부





10.6 정밀도와 재현율

- 두 부류를 긍정(positive)과 부정(negative)으로 구분하는 방법
  - 예) 의사의 진단: 목적은 환자를 가려내기 위함 → 환자를 긍정, 정상을 부정으로 봄
  - 예) 반도체 불량품 검사: 불량품이 긍정, 정상품이 부정
  - 예) 신용 카드 승인 시스템: 불승인이 긍정, 승인이 부정

- 예측의 네 가지 경우
  - 모델의 성능을 평가할때 사용되는 지표
  - 예측값이 실제 관측값을 얼마나 정확히 예측했는지 보여주는 행렬
  - TP(True Positive) : 참 긍정, 예) 환자를 환자로 분류
  - FP(False Positive) : 거짓 긍정, 예) 정상인을 환자로 분류
  - FN(False Negative) : 거짓 부정, 예) 환자를 정상인으로 분류
  - TN(True Negative) : 참 부정, 예) 정상인을 정상인으로 분류

- 그라운드 트루스(정답)을 붙이는 방법
  - 예) 불량품 검출: 생산라인에서 숙련자가 붙인 레이블
  - 예) 신용카드 승인: 사후에 들어온 도난 신고 정보를 사용

10.6 정밀도와 재현율

혼동 행렬

		ground truth	
		긍정	부정
예측값	긍정	TP	FP
	부정	FN	TN

Predicted

Actual	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Confusion Matrix

혼동 행렬에서 계산할 수 있는 네 가지 성능

TPR(True Positive Rate(참 긍정률) = $\frac{TP}{TP+FN} = \frac{TP}{A}$ (재현율 또는 민감도)

FPR(False Positive Rate(거짓 긍정률) = $\frac{FP}{FP+TN} = \frac{FP}{B}$ (특이도)

FNR(False Negative Rate(거짓 부정률) = $\frac{FN}{TP+FN} = \frac{FN}{A}$

TNR(True Negative Rate(참 부정률) = $\frac{TN}{FP+TN} = \frac{TN}{B}$

A:정답의 긍정(환자,불량,거부), B:정답의 부정(정상인,정상품, 승인)

10.6 정밀도와 재현율

- 정보검색에서 자주 사용하는 정밀도(precision)와 재현율(recall)

  정밀도 $= \frac{TP}{TP+FP}$

  재현율$= \frac{TP}{TP+FN}$ (TPR과 같음)

- 의료 분야에서 주로 사용하는 특이도(specificity)와 민감도(sensitivity)

  특이도 $= \frac{FP}{FP+TN}$ (FPR과 같음)

  민감도 $= \frac{TP}{TP+FN}$ (TPR과 같음)

10.6 정밀도와 재현율

암 판정 예

- 정밀 검사를 통한 최종 암 판정을 그라운드 트루스, 의사의 초진을 모델의 예측으로 간주

	1	2	3	4	5	6	7	8	9	10
Ground truth	N	N	P	N	P	P	N	N	N	P
초진(모델의 예측)	N	P	P	N	N	P	N	N	P	P
	TN	FP	TP	TN	FN	TP	TN	TN	FP	TP

		ground truth	
		긍정	부정
예측값	긍정	TP=3	FP=2
	부정	FN=1	TN=4

$$\text{정밀도} = \frac{TP}{TP+FP} = \frac{3}{3+2} = 0.6$$

$$\text{재현율} = \frac{TP}{TP+FN} = \frac{3}{3+1} = 0.75$$

$$\text{특이도} = \frac{FP}{FP+TN} = \frac{2}{2+4} = 0.33$$

$$\text{민감도} = \frac{TP}{TP+FN} = \frac{3}{3+1} = 0.75$$

10.7 ROC 곡선과 ACU

- 많은 시스템이 부류에 속할 확률을 출력
  - 예) 닥터 왓슨은 긍정(환자)일 확률을 출력함. 0.99라면 환자일 확률이 매우 높아 당장 입원 치료, 0.11이면 정상으로 판정, 0.48이면 판정을 보류하고 정밀 검사 의뢰

암 판정 예

	1	2	3	4	5	6	7	8	9	10
Ground truth	N	N	P	N	P	P	N	N	N	P
닥터 왓슨	0.26	0.81	0.73	0.11	0.20	0.48	0.23	0.11	0.61	0.99

불량품을 정상품으로 판단하는 것도 문제이고

정상품을 불량품으로 판단하는 것도 문제이다

10.7 ROC 곡선과 ACU

- 임계값(threshold)을 설정하여 확률을 긍정/부정으로 변환

임계값에 따라 확률을 레이블로 변환

	1	2	3	4	5	6	7	8	9	10
Ground truth	N	N	P	N	P	P	N	N	N	P
닥터 왓슨	0.26	0.81	0.73	0.11	0.20	0.48	0.23	0.11	0.61	0.99
T=1.00	N	N	N	N	N	N	N	N	N	N
T=0.75	N	P	N	N	N	N	N	N	N	P
T=0.05	N	P	P	N	N	N	N	N	P	P
T=0.25	P	P	P	N	N	P	N	N	P	P
T=0.00	P	P	P	P	P	P	P	P	P	P

- 임계값이 낮으면 거짓 긍정(FP)이 많아짐 → 멀쩡한 사람이 암환자 진단
- 임계값이 높으면 거짓 부정(FN)이 많아짐 → 암환자를 정상으로 판정하여 매우 위험
- 임계값 = 컷오프(cutoff)

10.7 ROC 곡선과 ACU

- ROC 곡선(Receiver Operating Characteristic curve)
  - [표 10-5]의 각 행에서 FPR과 TPR 계산하고, FPR을 가로축, TPR을 세로축으로 놓고 그린 그래프

임계값에 따른 FPR과 TPR의 변화

임계값	1.00	0.75	0.50	0.25	0.00
FPR	0.0	0.167	0.333	0.5	1.0
TPR	0.0	0.25	0.5	0.75	1.0

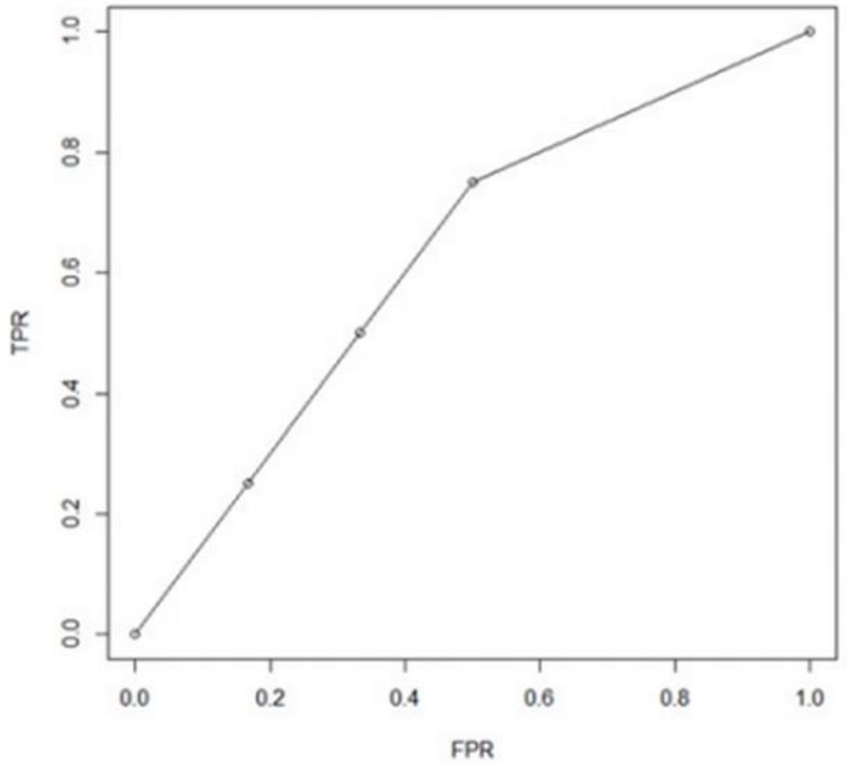


그림 10-6 ROC 곡선

10.7 ROC 곡선과 ACU

- AUC (Area Under Curve)
  - ROC 곡선의 아래쪽 영역의 면적
  - AUC는 0~1사이의 값을 가지는데 1에 가까울수록 좋은 예측 성능 (극단적으로 왼쪽 위 구석을 지나는 ROC 곡선은 최적 성능으로서 AUC는 1)
  - [그림 10-7]의 경우 오른쪽의 AUC가 더 큼

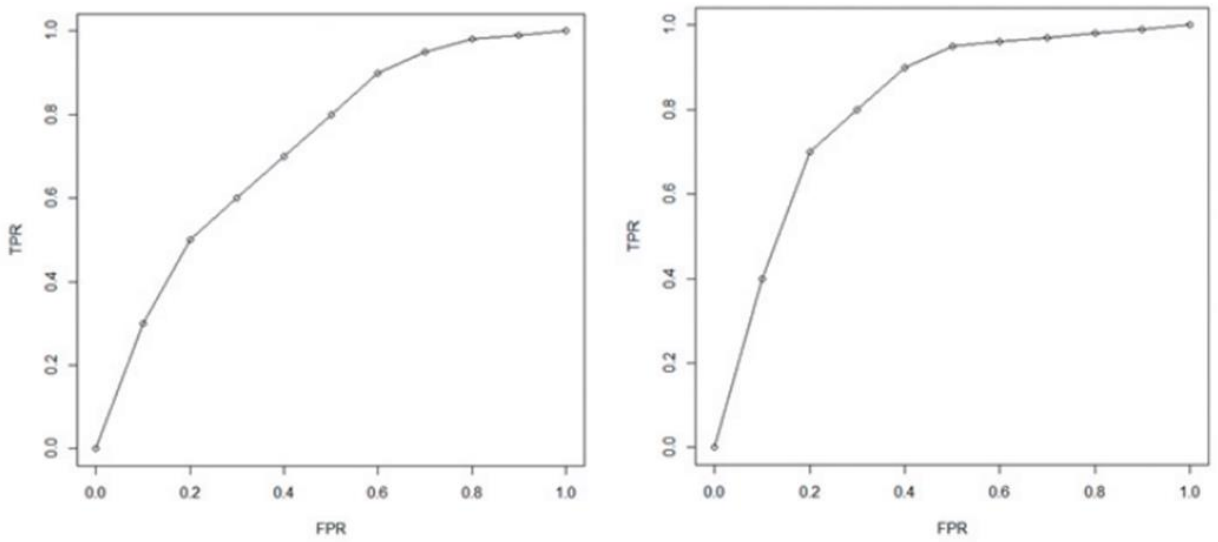


그림 10-7 두 모델의 ROC 곡선

10.7 ROC 곡선과 ACU

ROCR 라이브러리로 AUC 자동 계산

Console C:/RSources/

```
> library(ROCR)
> labels = c(0, 0, 1, 0, 1, 1, 0, 0, 0, 1)
> predictions = c(0.26, 0.81, 0.73, 0.11, 0.20, 0.48, 0.23, 0.11, 0.61, 0.99)
>
> p = prediction(predictions, labels)
> roc = performance(p, measure = 'tpr', x.measure = 'fpr')
>
> auc = performance(p, measure = 'auc')
> auc@y.values
[[1]]
[1] 0.7083333
```

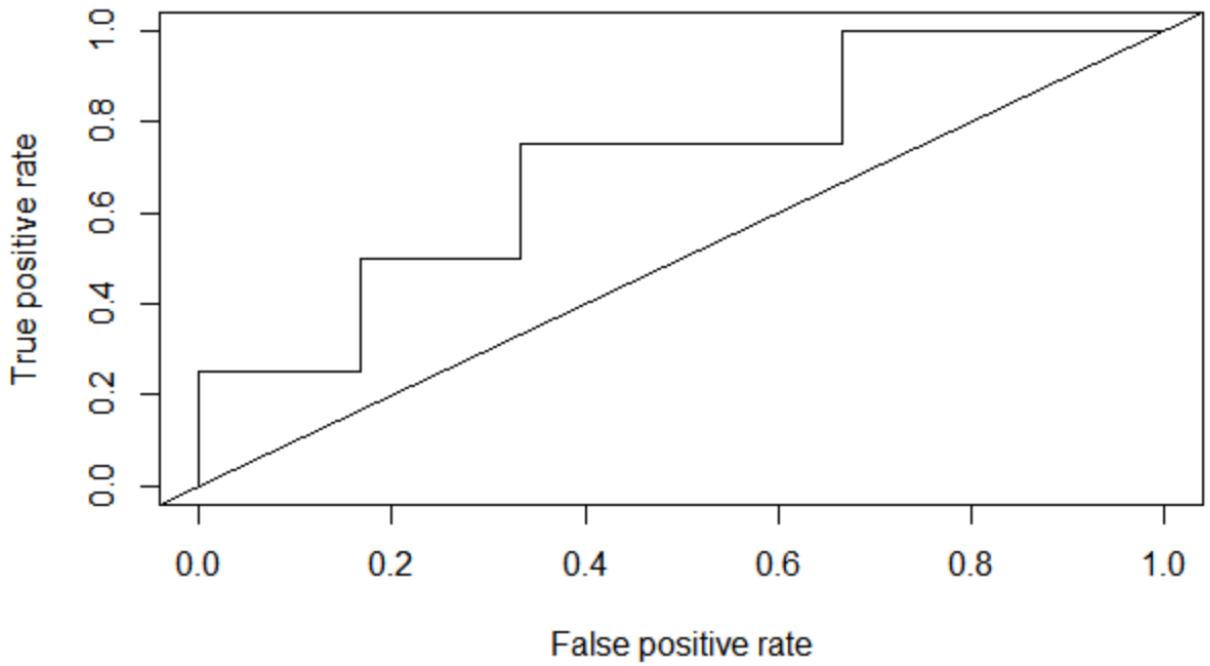
```
> str(p)
Formal class 'prediction' [package "ROCR"] with 11 slots
  ..@ predictions:List of 1
  .. ..$ : num [1:10] 0.26 0.81 0.73 0.11 0.2 0.48 0.23 0.11 0.61 0.99
  ..@ labels     :List of 1
  .. ..$ : Ord.factor w/ 2 levels "0"<"1": 1 1 2 1 2 2 1 1 1 2
  ..@ cutoffs    :List of 1
  .. ..$ : num [1:10] Inf 0.99 0.81 0.73 0.61 ...
  ..@ fp         :List of 1
  .. ..$ : num [1:10] 0 0 1 1 2 2 3 4 4 6
  ..@ tp         :List of 1
  .. ..$ : num [1:10] 0 1 1 2 2 3 3 3 4 4
  ..@ tn         :List of 1
  .. ..$ : num [1:10] 6 6 5 5 4 4 3 2 2 0
  ..@ fn         :List of 1
  .. ..$ : num [1:10] 4 3 3 2 2 1 1 1 0 0
```

① labels 변수에 정답 저장
② predictions 변수에 모델의 출력 값 저장
③ prediction함수의 첫번째 인자에 예측값, 두번째 인자에 정답, → 결과 p에 저장
④ performance 함수는 유사 정보를 축출하여 변수 roc에 저장
⑤ auc 계산 : performance 함수의 measure option을 acu로 설정

10.7 ROC 곡선과 ACU

- ROCR 라이브러리로 AUC 자동 계산

```
Console  C:/RSources/
> plot(roc)
> abline(a = 0, b = 1)
```



10.7 ROC 곡선과 ACU

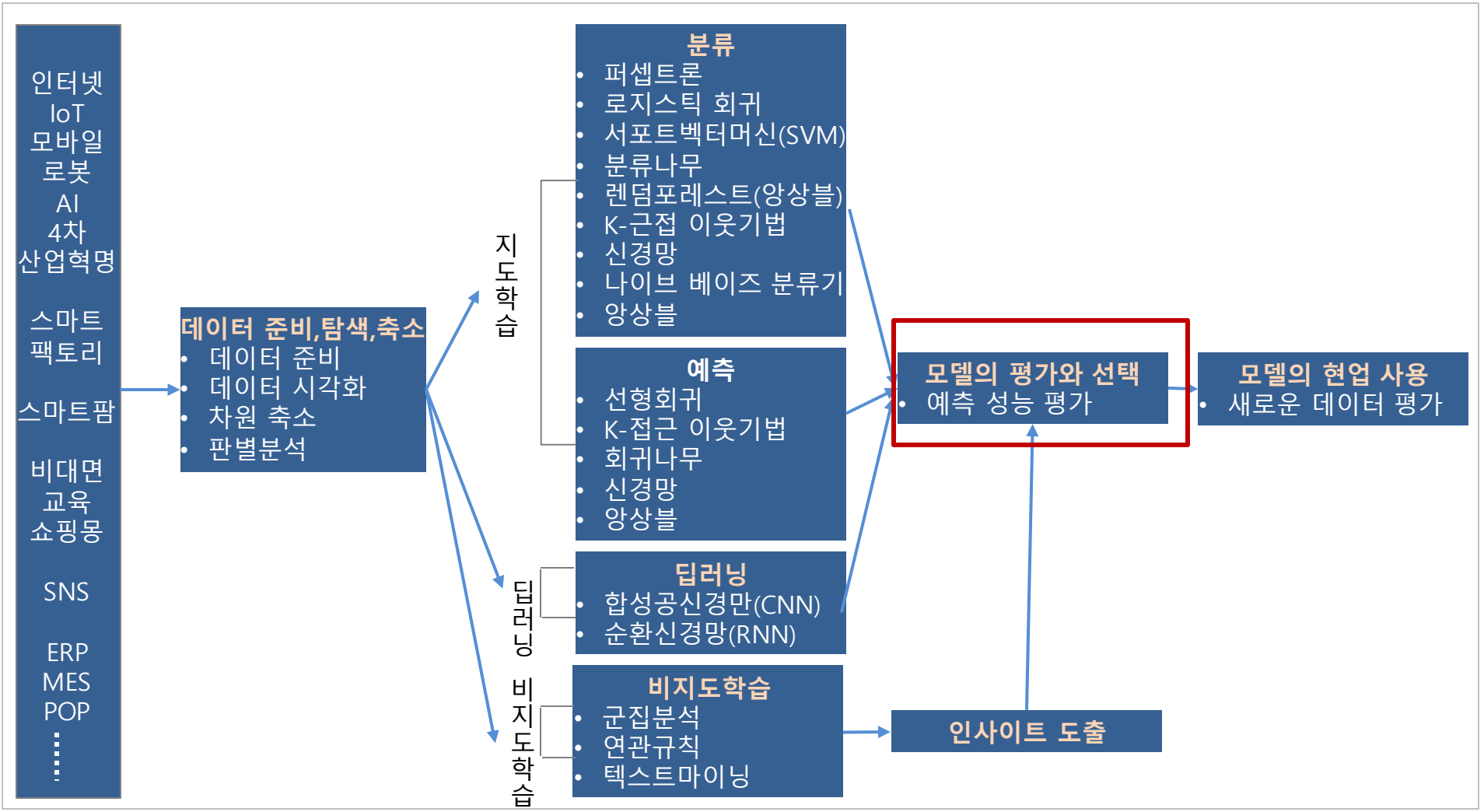
R 언어에서 @을 쓰는 경우

- @은 어떤 객체의 slot를 지칭함.
- 교재에서는 ROCR 라이브러리의 prediction과 performance 함수에서 사용
- R의 slot은 객체에 속한 멤버 변수

- 예) auc@y.vlaues는 auc 객체의 y.values라는 slot임
- 확인 방법 : slotNames(auc)

```
Console C:/RSources/
> slotNames(auc)
[1] "x.name"       "y.name"       "alpha.name"   "x.values"     "y.values"     "alpha.values"
```

요약


인터넷
IoT
모바일
로봇
AI
4차
산업혁명
스마트
팩토리
스마트팜
비대면
교육
쇼핑몽
SNS
ERP
MES
POP
데이터 준비,탐색,축소
데이터 준비
데이터 시각화
차원 축소
판별분석
지도학습
분류
퍼셉트론
로지스틱 회귀
서포트벡터머신(SVM)
분류나무
렌덤포레스트(앙상블)
K-근접 이웃기법
신경망
나이브 베이즈 분류기
앙상블
예측
선형회귀
K-접근 이웃기법
회귀나무
신경망
앙상블
딥러닝
딥러닝
합성공신경만(CNN)
순환신경망(RNN)
비지도학습
비지도학습
군집분석
연관규칙
텍스트마이닝
모델의 평가와 선택
예측 성능 평가
모델의 현업 사용
새로운 데이터 평가
인사이트 도출

Thank you


continent
Africa
Americas
Asia
Europe
Oceania
pop
2.50e+08
5.00e+08
7.50e+08
1.00e+09
1.25e+09
90
70
60
50
40
300
1000
3000
10000
30000
gdpPercap