CASE STUDY 1 · CHAPTER 2 · PART 2

조류 대발생 예측 (2)
모델링, 평가, 최종 예측

선형회귀와 회귀나무로 모델을 만들고, 교차검증으로 평가한 뒤, 140개 테스트 샘플의 조류 빈도를 예측한다

Part 1에서 완성한 clean.algae(198개, 결측치 없음)와 algae(198개, 회귀나무용 원본)를 이어서 사용합니다. 코드와 수치는 R을 직접 실행해 얻은 결과입니다.

RECAP

지난 시간 복습 (Part 1)

1

데이터

200개 수질 샘플, 11개 설명변수(계절·강 크기·유속 + 화학성분 8개), 7개 조류 빈도(a1~a7)

2

탐색

히스토그램·박스플롯·조건부 그래프로 분포와 이상치 확인

3

결측치 처리

62·199번 제거 후, 유사 사례(최근접 이웃) 방법으로 나머지 결측치를 채운 clean.algae(198개) 완성

오늘은 이 데이터를 이용해 실제 예측 모델을 만들고, 어떤 모델이 더 나은지 평가한 뒤, 140개 테스트 샘플의 조류 빈도를 예측합니다.

OVERVIEW

학습 목표

1. **회귀 모델 구축**
   선형회귀(lm)와 회귀나무(rpart) 모델을 R로 직접 만들고 결과를 해석할 수 있다.

2. **모델 평가**
   MAE·MSE·NMSE로 예측 성능을 계산하고, 교차검증으로 신뢰할 수 있는 성능을 추정할 수 있다.

3. **모델 선택과 결합**
   여러 모델의 성능을 비교하고, 가중 결합(combination) 전략을 이해할 수 있다.

4. **최종 예측**
   140개 테스트 샘플의 7개 조류 빈도를 예측하고 실제 정답과 비교해볼 수 있다.

REGRESSION

회귀 문제로 바라보기

- 예측 대상(a1~a7)이 수치(빈도)이므로, 이 문제는 회귀(regression) 문제
- 7개 조류를 예측해야 하므로, 사실상 7개의 독립적인 회귀 문제로 다룰 수 있음
- 이번 시간에는 먼저 조류 a1을 예로 들어 모델을 만들고 평가하는 과정을 살펴본 뒤, 7개 조류 전체로 확장

> *회귀 모델은 수치형 목표변수를 예측합니다. 반대로 목표변수가 범주(예/아니오, 클래스)라면 분류(classification) 문제가 됩니다.*

TWO MODELS

이번 시간에 다룰 두 가지 모델

선형회귀 (Linear Regression)

변수들의 가중합으로 목표값을 예측하는 가장 기본적인 통계 모델. 해석이 쉽지만 결측치가 있으면 사용할 수 없음.

회귀나무 (Regression Tree)

변수에 대한 조건(질문)을 반복해 데이터를 나누는 트리 구조 모델. 결측치를 자연스럽게 다룰 수 있고, 비선형 관계도 포착.

두 모델은 가정이 서로 다르고 결측치 처리 방식도 다르기 때문에, 선형회귀에는 clean.algae(채운 데이터)를, 회귀나무에는 algae(원본, 결측치 포함 가능)를 사용합니다.

LINEAR REGRESSION

선형회귀 모델 만들기: lm()

R CONSOLE

```
> lm.a1 <- lm(a1 ~ ., data=clean.algae[,1:12])
> summary(lm.a1)
```

- a1 ~ . : a1을 나머지 모든 변수로 예측하는 모델
- season/size/speed 같은 범주형 변수는 자동으로 더미(0/1) 변수로 변환됨

실행 결과 요약	값
Multiple R-squared	0.3739
Adjusted R-squared	0.3223
F-statistic (p-value)	7.245 (p = 2.2e-12)
유의한 변수(p<0.05)	sizesmall(*), NO3(**)

summary(lm.a1) 결과 해석하기

1

Residuals (잔차)

실제값과 예측값의 차이. 평균이 0에 가깝고 좌우 대칭일수록 바람직

2

계수(Coefficients)와 t-test

각 변수의 영향력(추정값)과 표준오차. p-value(Pr(>|t|))가 작을수록 해당 변수가 유의미

3

R-squared (결정계수)

모델이 설명하는 분산의 비율. 1에 가까울수록 적합도가 좋음 (a1 모델은 0.37로 낮은 편)

4

F-statistic

모델 전체가 목표변수와 무관하지 않다는 가설을 검정. p-value가 작으면(2.2e-12) 모델이 유의미

범주형 변수는 어떻게 처리될까? (더미 변수)

- k개의 값을 갖는 범주형(factor) 변수는 k-1개의 더미(0/1) 변수로 변환됨
- 예: season(4개 값) → seasonspring, seasonsummer, seasonwinter (3개 변수)
- 세 더미 변수가 모두 0이면 '기준값'인 autumn을 의미
- size(3개 값) → sizemedium, sizesmall / speed(3개 값) → speedlow, speedmedium

summary(lm.a1) 결과에 seasonspring, seasonsummer, seasonwinter, sizemedium, sizesmall, speedlow, speedmedium 이 개별 변수로 나타난 이유가 바로 이것입니다.

SIMPLIFICATION

모델 단순화 ①: anova()로 변수 중요도 보기

R CONSOLE

```
> anova(lm.a1)
```

변수	p-value	유의성
season	0.965	— (가장 낮음)
size	5.6e-08	***
speed	0.0022	**
oPO4	0.0001	***
NO3	0.0012	**
Chla	0.256	—

- anova() : 변수가 순서대로 모델에 추가될 때 오차가 얼마나 줄어드는지 검정
- season이 오차 감소에 가장 적게 기여(p=0.965) → 제거 후보 1순위
- update(lm.a1, . ~ . - season) 로 season을 제거한 lm2.a1 생성 → R^2 0.3739→0.3692로 거의 변화 없음 (anova 비교 p=0.71, 유의한 차이 없음)

SIMPLIFICATION

모델 단순화 ②: step()으로 자동 변수 선택

R CONSOLE

```
> final.lm <- step(lm.a1)
   ...  (AIC 기준으로 변수를 하나씩 제거)  ...
> summary(final.lm)
Call: lm(formula = a1 ~ size + mxPH + Cl + NO3 + PO4, ...)
```

- step() : AIC(Akaike Information Criterion)를 기준으로 변수를 자동으로 추가/제거하며 최적 모델 탐색 (기본은 후진제거)
- 11개 변수로 시작해 최종적으로 5개 변수(size, mxPH, Cl, NO3, PO4)만 남김

최종 모델(final.lm) 실행 결과 — R^2 = 0.3538, Adj R^2 = 0.3335, Residual SE = 17.49, F = 17.43 (p = 4.75e-16). 변수는 줄었지만 설명력은 원래 모델과 비슷 → 더 단순하면서 성능은 유지.

REGRESSION TREE

회귀나무 모델 만들기: rpart()

R CONSOLE

```
> library(rpart)
> rt.a1 <- rpart(a1 ~ ., data=algae[,1:12])
> rt.a1
1) root 198 90401.29 16.996
  2) PO4>=43.818 147 31279.12  8.980
    4) Cl>=7.8065 140 21622.83  7.493
      8) oPO4>=51.118 84  3441.15  3.846 *
      9) oPO4< 51.118 56 15389.43 12.963
       ...
```

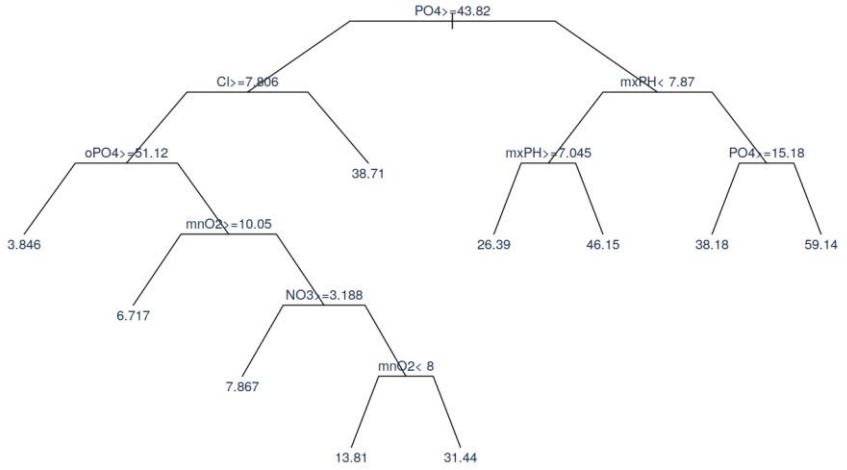
- algae(원본, 결측치 포함 가능)를 그대로 사용 — 회귀나무는 결측치를 내부적으로 처리
- rpart()가 사용한 데이터는 198개 샘플, 전체 평균(root) 16.996에서 시작해 조건을 따라 나뉨
- 결과: PO4, Cl, oPO4, mnO2, NO3, mxPH 변수를 사용한 10개의 leaf(잎) 노드로 구성된 트리

REGRESSION TREE

회귀나무 시각화

R CONSOLE

```
> plot(rt.a1, uniform=T, branch=0.5, margin=0.1)
> text(rt.a1, cex=0.8)
```



REGRESSION TREE

회귀나무는 어떻게 읽을까?

1

뿌리(root)에서 시작

전체 198개 샘플의 평균값(16.996)에서 출발

2

각 노드는 조건(질문)

예: 'PO4 ≥ 43.82인가?' — 참이면 왼쪽, 거짓이면 오른쪽 가지로 이동

3

잎(leaf)에 도달하면 예측 완료

더 이상 나뉘지 않는 마지막 노드의 평균값이 그 조건을 만족하는 샘플들의 예측값

4

변수는 자동 선택됨

트리가 스스로 가장 정보량이 큰 변수와 기준값을 선택 — 모든 변수가 트리에 등장할 필요는 없음

PRUNING

과적합 방지: printcp()로 트리 크기 살펴보기

R CONSOLE

```
> printcp(rt.a1)
```

nsplit	rel error	xerror	xstd
0	1.000	1.015	0.131
1	0.594	0.800	0.124
2	0.522	0.754	0.123
3	0.491	0.650	0.107
6	0.405	0.645 (최소)	0.110
9 (전체)	0.355	0.684	0.111

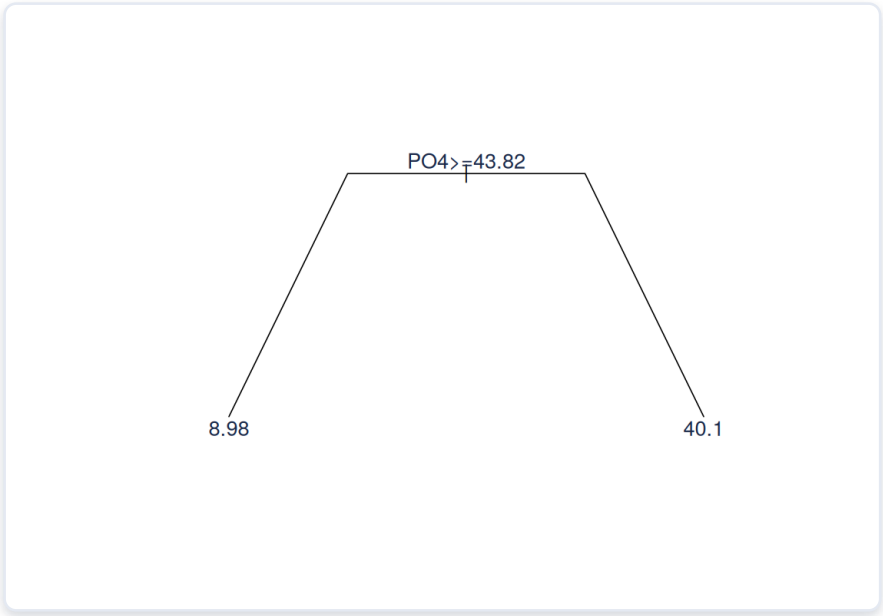
- rel error : 훈련 데이터에서의 상대오차 (트리가 커질수록 계속 감소 — 과적합 위험)
- xerror : 10-fold 교차검증으로 추정한 신뢰할 수 있는 상대오차 (더 중요!)

가지가 늘어난다고 xerror가 항상 줄지는 않음 → 무작정 큰 트리가 좋은 것은 아님

PRUNING

1-SE 규칙으로 최적 크기 선택하기

- 1-SE 규칙: 최소 xerror + 표준오차(xstd) 이내에서 가장 작은(단순한) 트리를 선택
- 우리 결과: 최소 xerror = 0.645(nsplit=6), 허용 한계 = 0.645+0.110 = 0.755
- 한계 이내의 가장 작은 트리는 nsplit=2 (3개의 잎) — 복잡도 대비 성능이 좋은 균형점
- prune(rt.a1, cp=값) 으로 원하는 크기까지 가지치기 가능



예시: prune(rt.a1, cp=0.08)로 자른 트리 (PO4 기준 단 1번만 분기) — 지나치게 단순화된 예시로, 실전에는 1-SE 규칙의 cp값을 사용

EVALUATION

예측 성능, 어떻게 측정할까?

1

MAE (평균절대오차)

mean(|예측값 - 실제값|) — 오차의 평균 크기. 목표변수와 같은 단위라 해석이 쉬움

2

MSE (평균제곱오차)

mean((예측값 - 실제값)²) — 큰 오차에 더 큰 벌점을 부여

3

NMSE (정규화 평균제곱오차)

MSE ÷ (단순히 평균만 예측했을 때의 MSE) — 1보다 작으면 '평균 예측'보다 낫다는 뜻

NMSE는 값의 스케일에 영향받지 않아, 서로 다른 조류(a1~a7)의 성능을 공정하게 비교할 때 특히 유용합니다.

EVALUATION

선형회귀 vs 회귀나무: 훈련 데이터 성능

R CONSOLE

```
> lm.predictions.a1 <- predict(final.lm, clean.algae)
> rt.predictions.a1 <- predict(rt.a1, algae)
```

지표	선형회귀 (lm)	회귀나무 (rt)
MAE	13.10	8.48
MSE	295.04	161.92
NMSE	0.646	0.355

이 수치는 책의 원본 예제와 R/rpart 버전 차이로 조금 다를 수 있지만, 이번 실행에서는 이 데이터(clean.algae/algae)와 코드로 직접 계산한 실제 결과입니다. 훈련 데이터 자체에 대한 성능이므로 과대평가(과적합)될 수 있다는 점에 유의 — 신뢰할 수 있는 비교는 잠시 후 교차검증으로 진행합니다.

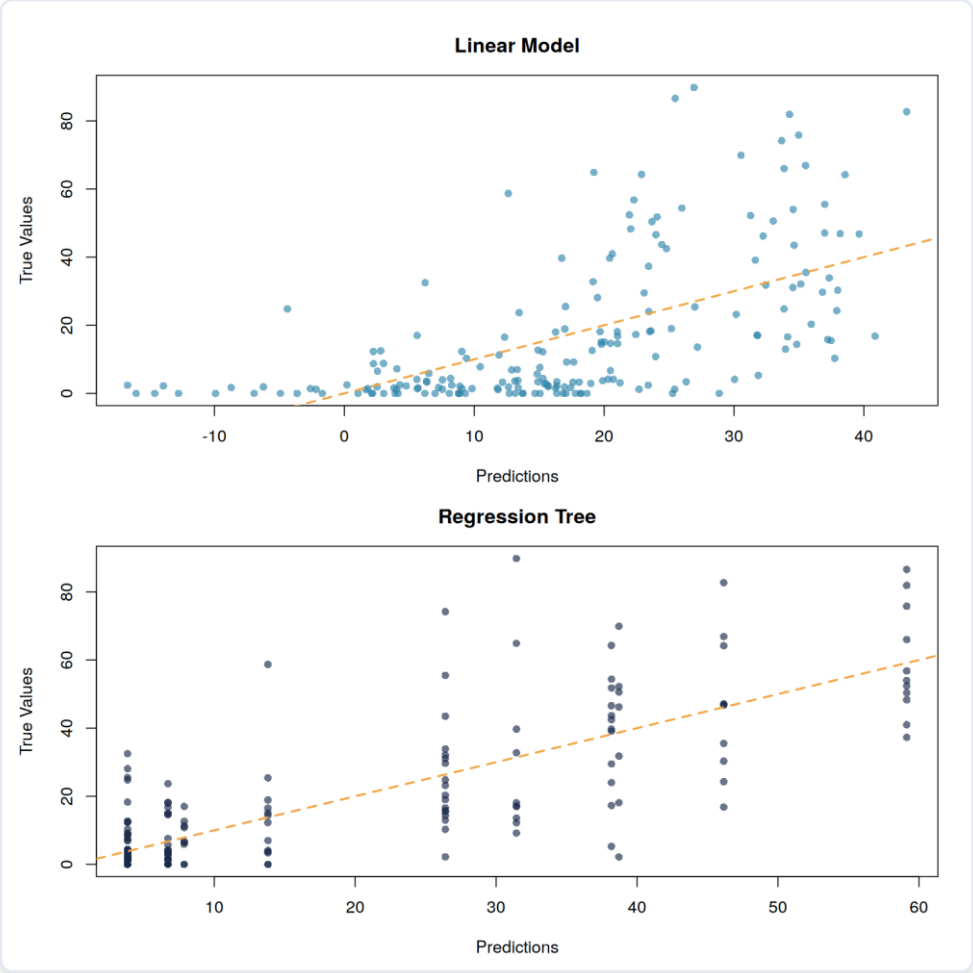
EVALUATION

예측값 vs 실제값 시각화

R CONSOLE

```
> plot(lm.predictions.a1, algae[,'a1'])
> abline(0,1,lty=2)
> plot(rt.predictions.a1, algae[,'a1'])
> abline(0,1,lty=2)
```

- 점선(대각선)에 가까울수록 예측이 정확함
- 두 모델 모두 큰 값에서 오차가 커지는 경향
- 선형회귀는 음수 예측을 만들어내는 문제가 있음(다음 슬라이드에서 보정)



EVALUATION

도메인 지식 활용하기: 음수 예측 보정

R CONSOLE

```
> sensible.lm.predictions.a1 <-
+   ifelse(lm.predictions.a1 < 0, 0, lm.predictions.a1)
> mean(abs(sensible.lm.predictions.a1 - algae[,'a1']))
```

- 조류의 발생 빈도는 절대 음수일 수 없음 — 이런 도메인 지식을 모델에 반영
- ifelse()로 음수 예측을 0으로 잘라내는 간단한 후처리만으로 성능이 향상됨

지표	보정 전	보정 후
MAE	13.10	12.47
NMSE	0.646	0.626

CROSS VALIDATION

왜 훈련 데이터 성능만으로는 부족할까?

- 훈련에 사용한 데이터로 측정한 성능은 낙관적으로 편향됨 (과적합, overfitting)
- 정말 중요한 것은 '한 번도 보지 못한 새로운 데이터'에 대한 성능
- K-fold 교차검증: 데이터를 K개로 나눈 뒤, 그중 1개는 평가용, 나머지 K-1개는 훈련용으로 삼아 K번 반복 → K개의 성능 점수를 평균
- 이번 실습은 재현 가능하도록 set.seed()로 난수를 고정했습니다

> *일반적으로 K=10을 많이 사용합니다. 데이터가 작을수록 교차검증이 특히 중요합니다 — 우리 데이터는 198개뿐이라 더욱 그렇습니다.*

CROSS VALIDATION

10-fold 교차검증 실행 (조류 a1)

R CONSOLE

```
> set.seed(20260825)
> cv10.res <- cross.validation(algae, clean.algae)
> cv10.res
```

모델	평균 NMSE	표준편차
선형회귀 (lm)	0.722	0.161
회귀나무 (rt)	0.701	0.267

두 모델의 평균 성능은 비슷하지만(0.72 vs 0.70), 회귀나무의 fold별 편차(표준편차 0.267)가 선형회귀(0.161)보다 훨씬 큽니다 — 즉 데이터 나누기에 따라 성능이 더 들쭉날쭉합니다.

두 모델의 차이는 통계적으로 유의미할까?

R CONSOLE

```
> t.test(cv10.res$lm$fold.res, cv10.res$rt$fold.res,
+   paired=T)

        Paired t-test
t = 0.209, df = 9, p-value = 0.839
95 percent confidence interval:
 -0.204  0.246
mean of the differences: 0.0208
```

- 대응표본 t검정(paired t-test): 같은 10개 fold에서 측정한 두 모델의 성능 차이가 0인지 검정
- p-value = 0.839 ≫ 0.05 → 두 모델의 성능 차이가 통계적으로 유의미하다고 볼 수 없음
- 결론: 조류 a1 하나만 놓고 보면, 선형회귀와 회귀나무 중 어느 쪽이 더 낫다고 확신할 수 없음

7개 조류 전체로 확장하기

R CONSOLE

```
> cv.all <- function(all.data, clean.data, n.folds=10) {
+   for (i in 1:n.folds)
+     for (a in 1:7) {
+       # a번째 조류(target)에 대해 lm, rt 모델을 만들고
+       # 가중 결합(comb) NMSE까지 함께 계산
+     }
+   list(lm=..., rt=..., comb=...)
+ }
> all.res <- cv.all(algae, clean.algae)
```

- 바깥쪽 for문(10-fold) 안에 안쪽 for문(7개 조류)을 중첩 — 조류마다, fold마다 lm/rt 모델을 새로 학습
- 이번에는 두 모델의 예측을 가중평균한 '결합(combination)' 모델의 성능도 함께 추정

7개 조류의 교차검증 NMSE 비교

조류	a1	a2	a3	a4	a5	a6	a7
lm	0.722	0.830	0.971	1.263	0.928	1.199	1.300
rt	0.691	1.000	1.000	1.000	1.000	1.000	1.103
comb	0.602	0.813	0.914	0.842	0.849	0.852	1.028

- 결합(comb) 모델이 a7을 제외한 모든 조류에서 가장 낮은(좋은) NMSE를 기록
- 회귀나무의 rt=1.000은 트리 성장이 되지 않아(변수 관계가 약해) '평균만 예측'한 것과 같은 성능이라는 의미
- a7은 모든 모델이 1에 가깝거나 넘음 → 이 조류는 화학 성분만으로 예측하기 특히 어려움을 시사

MODEL COMBINATION

결합(Combination) 모델은 어떻게 만들어질까?

R CONSOLE

```
wl <- 1 - perf.lm / (perf.lm + perf.rt)
wr <- 1 - wl
comb.preds <- wl * lm.preds + wr * rt.preds
```

- 두 모델의 예측을 단순 평균하는 대신, 교차검증에서 측정한 성능(NMSE)에 반비례하는 가중치를 부여
- 성능이 좋은(NMSE가 작은) 모델일수록 더 큰 가중치를 받음
- 서로 다른 모델을 섞으면(앙상블), 한 모델의 약점을 다른 모델이 보완해 전체적으로 더 안정적인 예측을 얻을 수 있음

FINAL PREDICTIONS

140개 테스트 샘플(Eval.txt) 준비하기

R CONSOLE

```
> test.algae <- read.table('Eval.txt', ...)
> data4dist <- rbind(algae[,1:11], test.algae[,1:11])
> dist.mtx <- as.matrix(daisy(data4dist, stand=T))
> clean.test.algae <- test.algae
> for(r in which(!complete.cases(test.algae)))
+   clean.test.algae[r, which(is.na(test.algae[r,]))] <-
+     apply(..., 2, central.value)
```

- 테스트 데이터는 목표변수(a1~a7)가 없는 11개 설명변수만 포함
- 훈련+테스트 데이터를 합쳐서 거리행렬을 계산해야 테스트 샘플의 결측치도 '가장 비슷한 이웃'으로 채울 수 있음
- 회귀나무는 결측치를 그대로 다룰 수 있으므로 원본 test.algae를 그대로 사용

FINAL PREDICTIONS

최종 예측 만들기

R CONSOLE

```
> lm.models <- lm.all(clean.algae, clean.test.algae)
> rt.models <- rt.all(algae, test.algae)
> final <- final.preds(lm.models$preds, rt.models$preds, all.res)
```

샘플	a1	a2	a3	a4	a5	a6	a7
1	7.97	7.68	4.04	3.61	6.52	4.12	2.48
2	12.67	7.99	3.73	1.15	5.19	7.80	2.59
3	15.15	7.78	3.51	2.32	4.43	5.53	1.95
4	14.37	6.34	5.51	1.65	3.60	4.90	1.69

각 조류마다 7개(a1~a7)의 lm 모델과 rt 모델을 200개 전체 훈련 데이터로 새로 학습한 뒤, all.res의 교차검증 성능으로 가중 결합해 140개 테스트 샘플의 예측값을 만듭니다. (위 표는 처음 4개 샘플)

FINAL PREDICTIONS

실제 정답(Sols.txt)과 비교하기

R CONSOLE

```
> algae.sols <- read.table('Sols.txt', ...)
> apply((final-algae.sols)^2, 2, mean) / apply(base.sq.errs, 2, mean)
```

조류	a1	a2	a3	a4	a5	a6	a7
MAE	11.24	7.09	4.07	1.73	5.44	7.63	2.70
NMSE(대비 기준선)	0.544	0.940	0.886	0.661	0.887	0.878	1.012

- 기준선(baseline) = 훈련 데이터의 평균값만 예측하는 가장 단순한 모델
- a1·a4는 기준선보다 크게 개선(NMSE 0.54, 0.66) — 화학 성분이 예측에 유용했다는 뜻
- a7만 기준선보다 살짝 나쁨(NMSE 1.01) — 예측이 어려운 조류였다는 앞선 관찰과 일치

SUMMARY

Chapter 2 전체 정리

- 데이터 탐색 → 결측치 처리 → 모델 구축(선형회귀·회귀나무) → 교차검증 평가 → 모델 결합 → 최종 예측
- 선형회귀와 회귀나무는 성능이 비슷했지만(a1 기준 통계적으로 유의한 차이 없음), 결합 모델이 대부분의 조류에서 더 안정적
- R로 전체 데이터마이닝 파이프라인을 처음부터 끝까지 직접 구현해 봄

다음 장 (Chapter 3)

주식시장 수익률 예측 — 시계열 데이터, MySQL 연동, 거래 신호와 시뮬레이션 트레이더

PRACTICE

직접 해보기

1

다른 조류로 반복

a1 대신 a2에 대해 lm()과 rpart() 모델을 만들고, MAE/NMSE를 비교해보세요.

2

트리 크기 바꿔보기

prune(rt.a1, cp=값)에서 cp를 다르게 주어 트리 크기가 어떻게 달라지는지 관찰해보세요.

3

가중치 직접 계산

all.res의 a3 결과로 wl, wr 가중치를 손으로 계산해보고, comb의 NMSE(0.914)와 비교해보세요.