강의 계획 및 평가 방법

주차	날짜	강의 내용	과제	대면/비대면	평가
1	03/06	강의 소개		Online	
2	03/13	데이터 마이닝 절차	과제 1 (10%)	A704	
3	03/20	데이터 탐색 및 시각화		B224	
4	03/27	차원 축소		Online	
5	**04/03**	**예측성능 평가**		**Online**	
6	04/10	다중 선형 회귀분석		A704	
7	04/17	중간 프로젝트 발표		B224	30%
8	04/24	k-최근접이웃 알고리즘 나이브 베이즈 분류	과제 2 (10%)	Online	
9	05/01	분류와 회귀 나무		Online	
10	05/08	로지스틱 회귀분석		Online	
11	05/15	신경망		A704	
12	05/22	판별 분석		Online	
13	05/29	연관 규칙		Online	
14	06/05	군집 분석		A704	
15	06/12	기말 프로젝트 발표		B224	40%

Data Mining for Business Analytics

Ch. 05 Evaluating Predictive Performance

2023.04.03.

Contents

5.1 Introduction

Evaluation Metrics

- 예측모델의 성능 평가
  - ✓ 평균오차(Average error), MAPE, RMSE
- 분류모델의 성능평가
  - ✓ 정오행렬(Confusion matrix) 기반 측도: 정확도(Accuracy), 민감도 (Sensitivity), 특이도 (Specificity), 오분류 비용(Misclassification costs)
  - ✓ Cutoff 값의 선택과 분류성능 사이의 관계: ROC(Receiver Operating Characteristics) 도표
- Ranking이 목적인 경우
  - ✓ 향상 차트(Lift charts)

Outcome Variables in Supervised Learning

- 수치 값: 출력변수가 수치 값일 때(예: 주택가격)
- 클래스 소속도: 출력 변수가 범주 값일 때(예: 구매자/비구매자)
- 경향(Propensity): 출력변수가 범주 값일 때 클래스 소속도의 확률(예: 채무불이행 경향)
- Classifier: Classification Methods
  - ✓ Classification: 새로운 레코드들의 클래스 소속도를 예측
  - ✓ Ranking: 새로운 레코드들의 집합에서 관심있는 클래스에 속할 가능성이 가장 큰 것 탐지

5.2 Evaluating Predictive Performance

Naïve Benchmark: The Average

- 새로운 레코드에 대한 예측은 단순히 학습용 데이터의 레코들에 대한 평균값
- 좋은 예측 모델은 예측의 정확성 면에서 벤치마크 기준보다 우수한 성능을 내야 함

Basic Terms

변수명			변수 내역
편차(Deviation)	관측값-평균	$x_i - \bar{x}$	자료가 평균을 중심으로 얼마나 광범위하게 분포하고 있는가
잔차(Rresidual)	관측값과 회귀직선의 예측값과의 차이	$y_i - \hat{y}_i$	회귀모형의 적합도(모델 학습)
오차(Error)	실제값과 예측값의 차이	$y_i - \hat{y}_i$	데이터마이닝 성능평가에서 사용, 모형의 성능(실제값 예측)

5.2 Evaluating Predictive Performance

Prediction Accuracy Measures

변수명			변수 내역
예측 오차	Prediction error	$e_i = y_i - \hat{y}_i$	실제 출력값과 예측된 출력값의 차이
평균 오차	AE(Average Error)	$\frac{1}{n}\sum_{i=1}^{n} e_i$	예측이 평균적으로 반응의 예측을 초과하는지 미달하는지 확인
절대평균 오차/편차	MAE(Mean Absolute Error) or MAD(Mean Absolute Deviation)	$MAE = \frac{1}{n}\sum_{i=1}^{n} \lvert e_i \rvert$	평균 절대오차의 크기
평균 백분율 오차	MPE(Mean Percentage Error)	$100 \times \frac{1}{n}\sum_{i=1}^{n} \frac{e_i}{y_i}$	오차의 방향을 고려하여 예측이 실제 값에서 얼마나 벗어나는지에 대한 퍼센트
절대평균 백분율 오차	MAPE(Mean Absolute Percentage Error)	$100 \times \frac{1}{n}\sum_{i=1}^{n} \left\lvert \frac{e_i}{y_i} \right\rvert$	예측이 실제 값에서 평균적으로 벗어나는 정도를 백분율 점수로 표현
평균 제곱 오차의 제곱근	RMSE(Root Mean Squared Error)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}$	학습용 데이터(MAE) → 평가용 데이터(RMSE)

5.2 Evaluating Predictive Performance

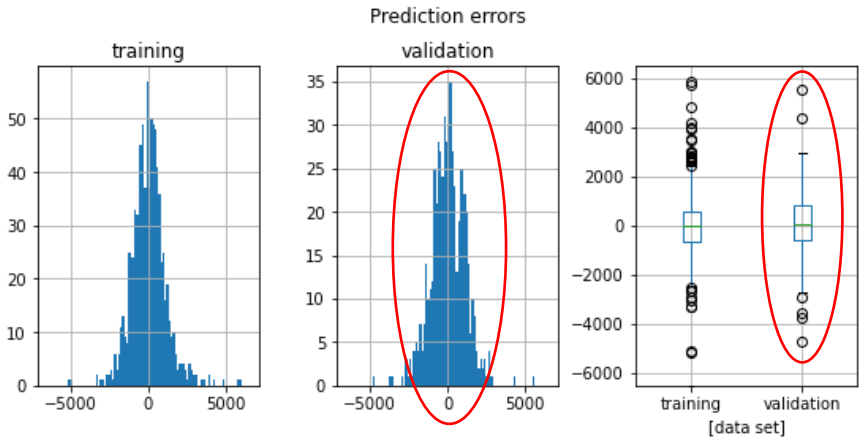
[실습] Table 5.1

Prediction Accuracy Measures

- ex) Toyota Corolla Car
- Training set: 861 / Validation set: 575
- 검증용 데이터에 대한 예측 오차의 결과: 대부분의 오차가 [−2,000, 2000] 범위 안에 있음
- 오차 분포는 유사

Comparing Training and Validation Performance

- 모델이 복잡해 질수록 학습 데이터에 "과적합"될 가능성이 큼 → 학습과 검증 오차의 차이가 더 커짐
- 극단적인 과적합: 학습 오차는 0, 검증 오차는 무시할 수 없는 값
- 아래 예) 학습용 데이터에 대한 성능이 검증용 데이터에 대한 성능 보다 약간 더 좋음.



[training] Regression statistics

Mean Error (ME)	-0.0000
Root Mean Squared Error (RMSE)	1121.0606
Mean Absolute Error (MAE)	811.6770
Mean Percentage Error (MPE)	-0.8630
Mean Absolute Percentage Error (MAPE)	8.0054

[validation] Regression statistics

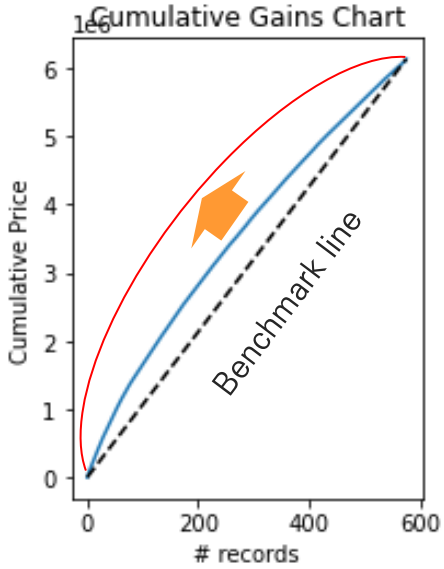
Mean Error (ME)	97.1891
Root Mean Squared Error (RMSE)	1382.0352
Mean Absolute Error (MAE)	880.1396
Mean Percentage Error (MPE)	0.0138
Mean Absolute Percentage Error (MAPE)	8.8744

5.2 Evaluating Predictive Performance

Cumulative Gains and Lift Charts

- 활용 목적: 새로운 레코드의 집합 중에서 누적 예측 값이 가장 높은 레코드들의 부분집합을 찾음 → Ranking
- 예) 렌터카 회사는 보유 차량을 최신 차종으로 유지하려고 함. 보통 중고차 중개업자를 통한 대량 판매. 일부 제한된 수의 자동차는 자체 판매경로를 통해서 판매하는 게 이익. → 자체 경로를 통해 재판매할 자동차를 선택하기 위한 모델 필요. → 리프트 차트는 수익에 대한 예측된 향상 정도를 보여줌

- 레코드들의 집합을 예측 값에 대해서 높은 것부터 낮은 것으로 배열
- x축: 누적된 레코드의 수
- y축: x에 대한 함수로 실제 값들을 누적시킨 값
- 대각선 직선: 각 레코드에 단순한 평균 예측 값을 배정하고 이 평균값을 누적시켜서 대각선이 되는 직선
- 모델의 향상 곡선(cumulative gain curve)이 벤치마크 대각선에서 멀어질 수록, 낮은 값의 결과 레코드들과 높은 값의 결과 레코드들을 잘 분류



5.2 Evaluating Predictive Performance

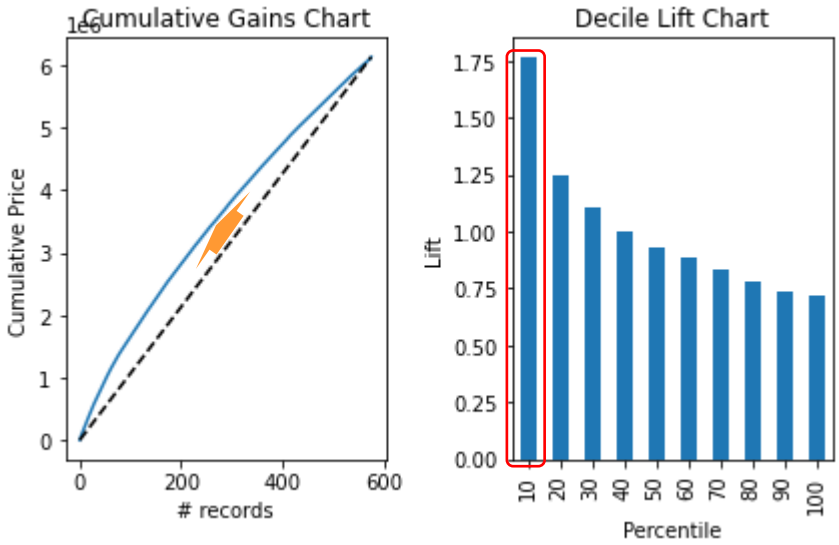
[실습] Figure 5.2

Cumulative Gains and Lift Charts

- 10분위 향상 차트: 배열된 레코드를 10개의 십분위로 집단화하고, 각 십분위에 대해서 단순한 벤치마크 향상 정도 대비 모델의 향상 비율(Lift)을 보여줌
- ex) Toyota Corolla: 검증 데이터(575대 차량)에 대한 결과

✓ 향상 정도에 있어서 예측 성능이 베이스라인 모델보다 좋음: 향상 곡선이 기준모델의 직선보다 더 높은 위치

✓ 예측된 판매를 가장 높게 하는 자동차들의 상위 10%를 선택하면, 임의의 10%를 선택하는 것에 비해서 1.75배 높은 수익을 얻음



5.3 Judging Classifier Performance

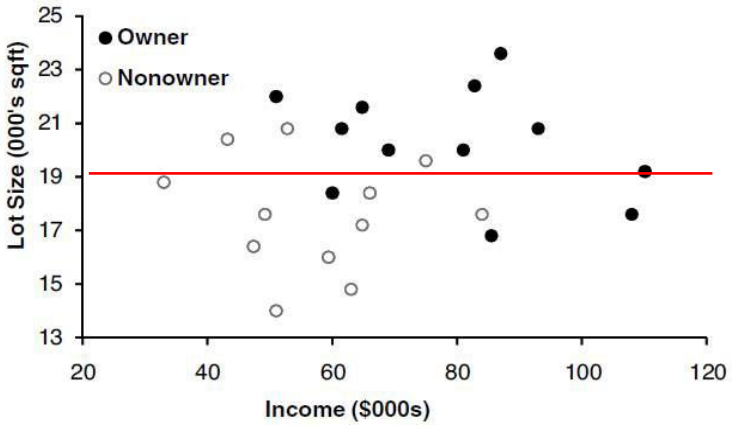
Benchmark: The Naïve rule

- 가장 지배적인 클래스에 속한다고 분류(다수결의 원칙): 베이스라인이나 벤치마크로 활용 가능

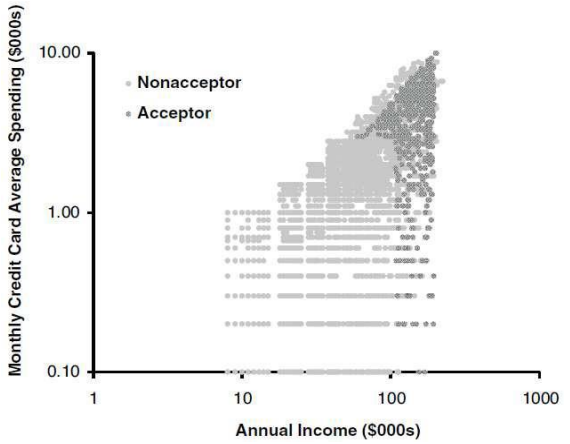
Class Separation

- 예측변수 정보로 클래스들이 잘 분리된다면, 작은 데이터셋으로도 좋은 분류기를 찾을 수 있지만, 그렇지 않다면 매우 큰 데이터셋으로도 분류를 잘 할 수 없음 → 예측변수의 선택이 중요

(a) Data 개수(n=24, 각 class 12개 씩)가 작아도 예측변수(income & lot size)에 의해 class "Owner/Nonowner" 분리가 용이



(b) Data 개수는 많지만(5,000개) 예측변수(income & credit card spending)에 의해 class "Acceptor/Nonacceptor" 분리가 어려움



5.3 Judging Classifier Performance

The Confusion (Classification) Matrix

- 분류 결과의 정확성을 평가하여 최적의 분류모델을 선택
- 대각선 셀: 올바른 분류의 개수
- 올바른 분류의 추정 값과 오분류율 파악

	Predicted Class 0	Predicted Class 1
Actual 0	2689	25
Actual 1	85	201

Using the Validation Data

- 학습 데이터를 사용하여 분류기 구축
- 검증 데이터에 적용하여 예측된 분류 결과 산출
- 분류 결과를 Confusion Matrix로 요약
- 과적합 검증: 학습 데이터와 검증 데이터의 Confusion Matrix 비교

5.3 Judging Classifier Performance

Accuracy Measure

		Predicted Class	
		C_1	C_2
Actual Class	C_1	True $n_{1,1}$ = 올바르게 분류된 C_1의 개수	False $n_{1,2}$= C_2로 잘못 분류된 C_1의 개수
	C_2	False $n_{2,1}$ = C_1으로 잘못 분류된 C_2의 개수	True $n_{2,2}$ = 올바르게 분류된 C_2의 개수

추정된 오분류율 Or 전체 오차율	Estimated Misclassified Rate or Overall Error Rate	$err = \frac{n_{1,2} + n_{2,1}}{n}$
전체 정확도	Overall Accuracy	$accuracy = \frac{n_{1,1} + n_{2,2}}{n} = 1 - err$

	Predicted Class 0	Predicted Class 1
Actual 0	2689	25
Actual 1	85	201

- Overall error rate = (25+85)/3000 = 3.67%
- Accuracy = 1 – err = (201+2689)/3000 = 96.33%

✓ If multiple classes, error rate is:
(sum of misclassified records)/(total records)

5.3 Judging Classifier Performance

[실습] Table 5.5, Figure 5.4

Propensities and Cutoff for Classification

- 경향(Propensities): 레코드(개체)가 각 클래스에 속할 확률
- 분류: cutoff 점수를 이용해 클래스 소속도 예측
- Cutoff: 특정 클래스에 속할 분류 기준 값. 이진분류의 경우 일반적으로 0.5

- Cutoff value = 0.5 → misclassification rate = 3/24
- Cutoff value = 0.25 → misclassification rate = 5/24
- Cutoff value = 0.75 → misclassification rate = 6/24

- Stable accuracy:
  - ✓ Accuracy: around 0.8
  - ✓ Cutoff value: (0.2, 0.8)

cutoff = 0.5

Confusion Matrix (Accuracy 0.8750)

Actual	Prediction nonowner	owner
nonowner	10	2
owner	1	11

cutoff = 0.25

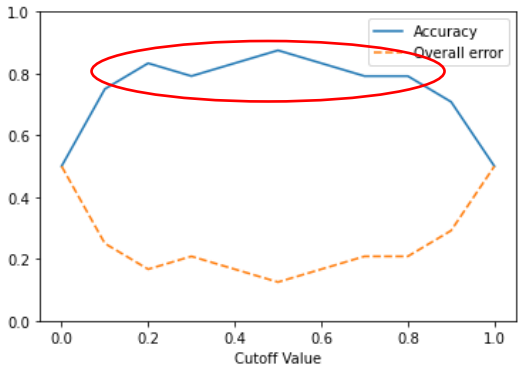
Confusion Matrix (Accuracy 0.7917)

Actual	Prediction nonowner	owner
nonowner	8	4
owner	1	11

cutoff = 0.75

Confusion Matrix (Accuracy 0.7500)

Actual	Prediction nonowner	owner
nonowner	11	1
owner	5	7



5.3 Judging Classifier Performance

[실습] Table 5.5

Propensities and Cutoff for Classification

- Cutoff value = 0.5 → misclassification rate = 3/24
- Cutoff value = 0.25 → misclassification rate = 5/24
- Cutoff value = 0.75 → misclassification rate = 6/24

[분류기로 추정된 "owner" 클래스가 될 확률(경향)과 실제 클래스 (24개 레코드)]

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

cutoff = 0.5

Confusion Matrix (Accuracy 0.8750)

Actual \ Prediction	nonowner	owner
nonowner	10	2
owner	1	11

cutoff = 0.25

Confusion Matrix (Accuracy 0.7917)

Actual \ Prediction	nonowner	owner
nonowner	8	4
owner	1	11

cutoff = 0.75

Confusion Matrix (Accuracy 0.7500)

Actual \ Prediction	nonowner	owner
nonowner	11	1
owner	5	7

5.3 Judging Classifier Performance

Propensities and Cutoff for Classification

- 오분류 비율이 증대함에도 불구하고 0.5가 아닌 다른 cutoff 값을 사용하는 이유

→ 오분류 비용이 비대칭적(asymmetric)이기 때문. 즉, “1을 1로 제대로 맞히는 것이, 0을 0으로 제대로 분류하는 것보다 중요”

	Predict Class 0	Predict Class 1
Actual 0	80	15
Actual 1	2	3

Accuracy (for class 0)= 80/100
Accuracy (for class 1)= 3/100

	Predict Class 0	Predict Class 1
Actual 0	80	10
Actual 1	2	8

Accuracy (for class 0)= 80/100
Accuracy (for class 1) = 8/100

cutoff = 0.5

Confusion Matrix (Accuracy 0.8750)

	Prediction	
Actual	nonowner	owner
nonowner	10	2
owner	1	11

cutoff = 0.25

Confusion Matrix (Accuracy 0.7917)

	Prediction	
Actual	nonowner	owner
nonowner	8	4
owner	1	11

cutoff = 0.75

Confusion Matrix (Accuracy 0.7500)

	Prediction	
Actual	nonowner	owner
nonowner	11	1
owner	5	7

5.3 Judging Classifier Performance

Performance in Case of Unequal Importance of Classes

- C_1 클래스의 소속도를 예측하는 것이 C_2 클래스 보다 더 중요하다고 가정
- ex) C_1 : 파산 / C_2 : 지불능력 있음
- 아래 두 가지 measure들의 균형을 맞추는 값으로 cutoff 값을 찾는 것이 중요

Sensitivity (민감도: Recall)

- 중요한 클래스의 멤버를 올바르게 감지하는 능력
- C_1 멤버를 올바르게 분류하는 비율 $\frac{n_{1,1}}{n_{1,1}+n_{1,2}}$

Specificity (특이도)

- C_2 멤버를 올바르게 제외하는 능력
- C_2 멤버를 올바르게 분류하는 비율 $\frac{n_{2,2}}{n_{2,1}+n_{2,2}}$

		Predicted Class	
		C_1	C_2
Actual Class	C_1	True $n_{1,1}$ = 올바르게 분류된 C_1의 개수	False $n_{1,2}$= C_2로 잘못 분류된 C_1의 개수
	C_2	False $n_{2,1}$ = C_1으로 잘못 분류된 C_2의 개수	True $n_{2,2}$ = 올바르게 분류된 C_2의 개수

5.3 Judging Classifier Performance

Performance in Case of Unequal Importance of Classes

		Predicted Class: Positive [0] (more important)	Predicted Class: Negative [1]
Actual Class	Positive [0]	True Positive (TP)	False Negative (FN)
	Negative [1]	False Positive (FP)	True Negative (TN)

ROC Curve

Sensitivity (민감도: Recall) True Positive Rate(TPR)

$$\frac{TP}{TP + FN}$$

False Negative Rate (FNR)

$$1 - Sensitivity = \frac{FN}{TP + FN}$$

ROC Curve

False Positive Rate (FPR)

$$1 - Specificity = \frac{FP}{FP + TN}$$

Specificity (특이도) True Negative Rate(TNR)

$$\frac{TN}{FP + TN}$$

Precision (정밀도)

$$\frac{TP}{TP + FP}$$

Accuracy (정확도) $\frac{TP + TN}{n}$

F1 Score $\frac{2 \times P \times R}{P + R}$

5.3 Judging Classifier Performance

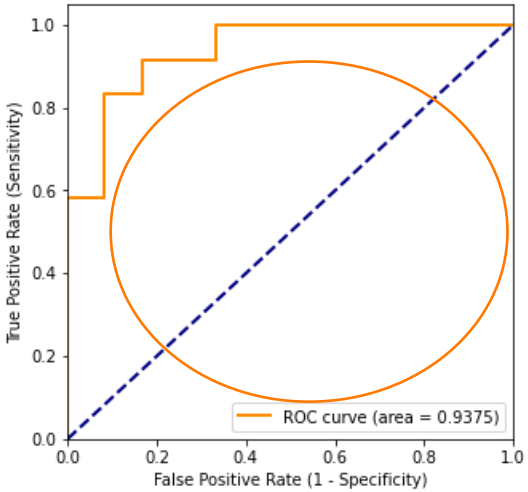
[실습] Figure 5.5

Performance in Case of Unequal Importance of Classes

ROC(Receiver Operating Characteristic) Curve

- x-축: 1-specificity(False Positive Rate): 실제로 "1"인 클래스를 "0" 클래스로 잘못 예측한 비율 (0에 가까울수록 좋음)
- y-축: Sensitivity(True Positive Rate): 실제로 "1"인 클래스를 "1" 클래스로 바르게 예측한 비율 (1에 가까울 수록 좋음)

		Predicted Class	
		Positive [0] (more important)	Negative [1]
Actual Class	Positive [0]	True Positive (TP) [Sensitivity(Recall) 부분]	False Negative (FN)
	Negative [1]	False Positive (FP) [1-Specificity 부분]	True Negative (TN)



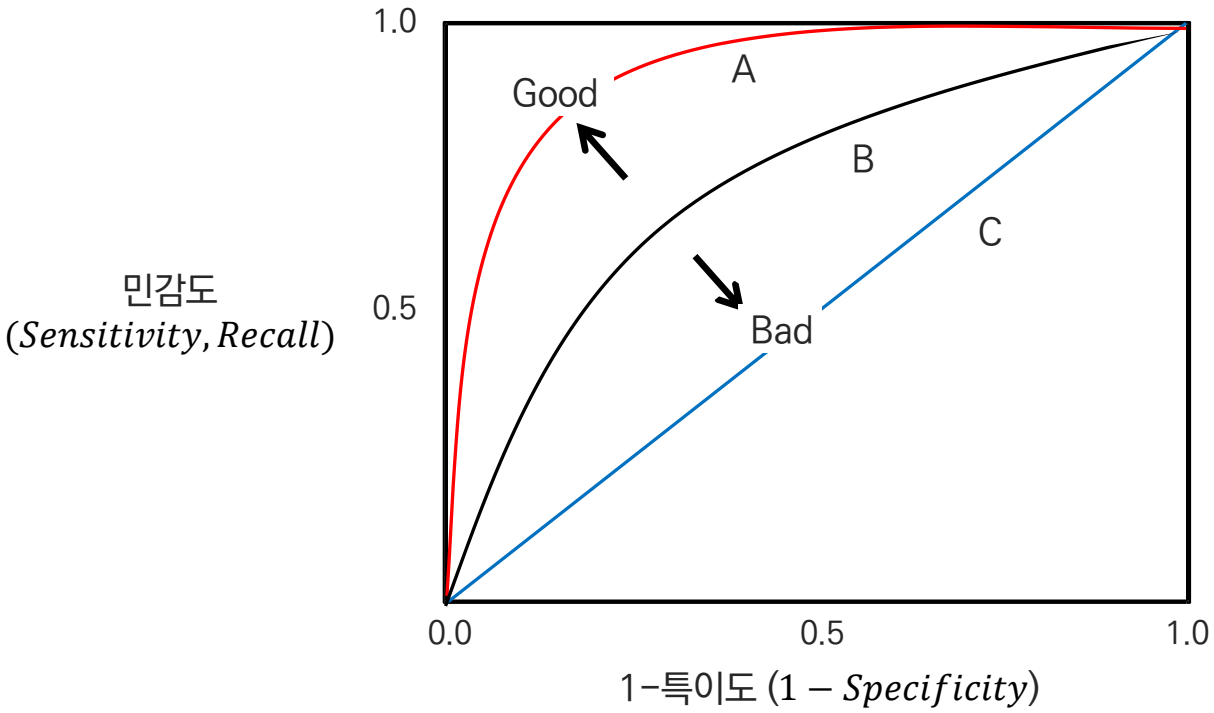
AUC(Area Under the Curve): 곡선 아래 영역. 넓을 수록 좋음

5.3 Judging Classifier Performance

Performance in Case of Unequal Importance of Classes

ROC(Receiver Operating Characteristic) Curve

AUC(Area Under the Curve): 곡선 아래 영역. 넓을 수록 좋음



5.3 Judging Classifier Performance

Asymmetric Misclassification Costs

- 분류 오류라는 관점에서 분류기의 가치를 평가 → 오분류 셀의 비용/이익(Cost/Benefit) 측정
- Confusion Matrix를 사용하여 검증 데이터의 각 레코드에 대한 오분류 비용의 기댓값 계산 → 전체 기대비용(또는 이득)을 기준으로 분류기 비교

ex) 우편물을 통한 구매 제의

- 1,000개 메일, 평균 1% respond
- “0”=not respond / “1”=respond
- Naïve rate: 모두를 “0”으로 분류

  → Error rate = 1%
- 오른쪽과 같은 분류 결과 가정
- Error rate = 100*(20+2)/1,000 = 2.2%

  → Naïve rate 보다 높음

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

5.3 Judging Classifier Performance

Asymmetric Misclassification Costs

ex) 우편물을 통한 구매 제의

- 1,000개 메일, 평균 1% 응답
- "0"=not respond / "1"=respond
- 가정: Profit = $10 per "a respond", Cost = $1 per "sending an offer"
- Naïve: 모두를 "0"으로 분류

  → Profit = $0 / Cost = $100(10 * $10)
- 분류 결과

  → Profit = $60 / Cost = $48

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

Profit	Predict Class 0	Predict Class 1
Actual 0	0	−$20
Actual 1	0	$80

Costs	Predict Class 0	Predict Class 1
Actual 0	0	$20
Actual 1	$20 (Opportunity Cost)	$8

5.4 Judging Ranking Performance

[실습] Figure 5.6

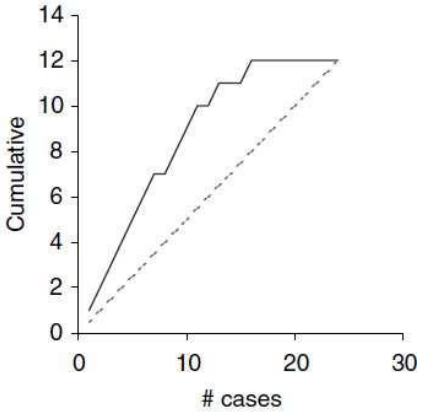
Gains and Lift Charts for Binary Data

- 랭킹(Ranking): 관심있는 클래스에 속할 가능성이 가장 큰 집단 추출
- 경향(Propensity): 출력변수가 범주값일 때 클래스 소속도의 확률
- 향상차트(lift chart, gain chart): 상대적으로 적은 수의 데이터를 선택하여 상대적으로 큰 비율의 응답자 추출
- 특정한 클래스가 상대적으로 드물지만 다른 클래스보다 훨씬 더 관심있는 경우: 탈세, 채무 불이행, 메일링에 대한 응답자
- 목표: 특정한 클래스 소속도의 경향에 따라서 레코드들의 랭크 순위 획득

Sorting by Propensity

- 레코드들의 집합을 경향(중요한 클래스, 예를 들면 C_1 에 속하는 경향)에 따라 내림차순으로 정렬
- 각 행에서 C_1 멤버(실제 클래스 C_1)의 누적 수 계산
- x축: 누적된 레코드의 수
- y축: x에 대한 함수로 실제값들을 누적시킨 값

Serial no.	Predicted prob of 1	Actual Class	Cumulative Actual class
1	0.995976726	1	1
2	0.987533139	1	2
3	0.984456382	1	3
4	0.980439587	1	4
5	0.948110638	1	5
6	0.889297203	1	6
7	0.847631864	1	7
8	0.762806287	0	7
9	0.706991915	1	8
10	0.680754087	1	9
11	0.656343749	1	10
12	0.622419543	0	10
13	0.505506928	1	11
14	0.47134045	0	11
15	0.337117362	0	11
16	0.21796781	1	12
17	0.199240432	0	12
18	0.149482655	0	12
19	0.047962588	0	12
20	0.038341401	0	12
21	0.024850999	0	12
22	0.021806029	0	12
23	0.016129906	0	12
24	0.003559986	0	12



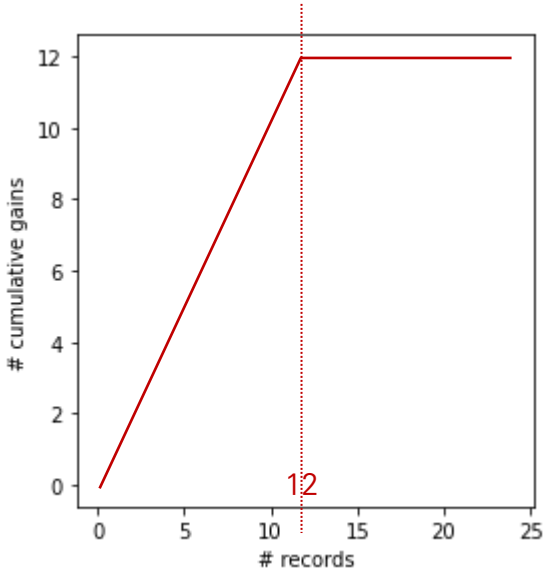
5.4 Judging Ranking Performance

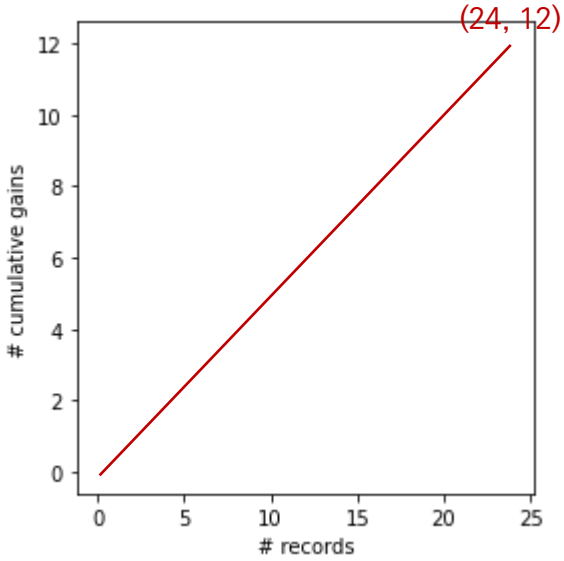
Gains and Lift Charts for Binary Data

Interpreting the Cumulative Gain Chart

- 이상적인 랭킹(Ranking) 성능: 모든 “1”의 클래스를 앞쪽에 위치시키고(실제 “1” 클래스의 데이터는 가장 높은 경향을 가져 표의 위쪽에 있음), 모든 “0”을 뒤쪽에 위치시킴 → 향상차트는 “1”이 축적될 때까지는 기울기 1인 대각선이다가, “0”이 시작되면 수평선이 되는 형태를 가짐
- 벤치마크(Baseline Model): Actual class 컬럼의 값을 random shuffle. (0, 0)과 (24, 12)를 잇는 대각선

Serial no.	Predicted prob of 1	Actual Class	Cumulative Actual class
1	0.995976726	1	1
2	0.987533139	1	2
3	0.984456382	1	3
4	0.980439587	1	4
5	0.948110638	1	5
6	0.889297203	1	6
7	0.847631864	1	7
8	0.762806287	0	7
9	0.706991915	1	8
10	0.680754087	1	9
11	0.656343749	1	10
12	0.622419543	0	10





5.4 Judging Ranking Performance

Gains and Lift Charts for Binary Data

Interpreting the Cumulative Gain Chart

- lift curve의 의미: 이 model을 사용하게 되면 random하게 case들을 선택하는 것 보다 이 “lift” 만큼 올바를 확률이 높아짐
- e.g.) 10개를 class 1이라고 선택하면 9/10의 확률(lift curve)로 올바른 classification이지만, random 하게 선택한다면 10×12/24=5, 즉 5/10의 확률(reference line)로 올바른 classification을 한 것이 됨

Serial no.	Predicted prob of 1	Actual Class	Cumulative Actual class
1	0.995976726	1	1
2	0.987533139	1	2
3	0.984456382	1	3
4	0.980439587	1	4
5	0.948110638	1	5
6	0.889297203	1	6
7	0.847631864	1	7
8	0.762806287	0	7
9	0.706991915	1	8
10	0.680754087	1	9
11	0.656343749	1	10
12	0.622419543	0	10
13	0.505506928	1	11
14	0.47134045	0	11
15	0.337117362	0	11
16	0.21796781	1	12
17	0.199240432	0	12
18	0.149482655	0	12
19	0.047962588	0	12
20	0.038341401	0	12
21	0.024850999	0	12
22	0.021806029	0	12
23	0.016129906	0	12
24	0.003559986	0	12

5.4 Judging Ranking Performance

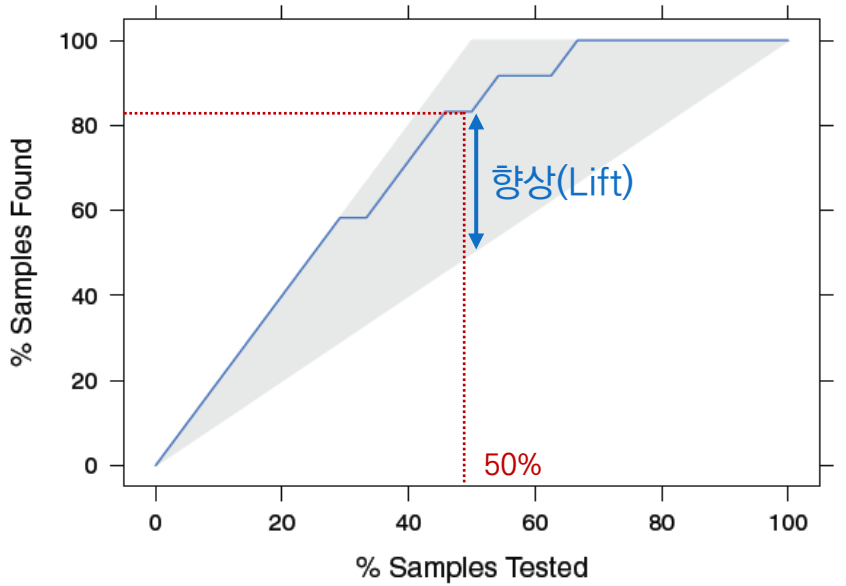
Gains and Lift Charts for Binary Data

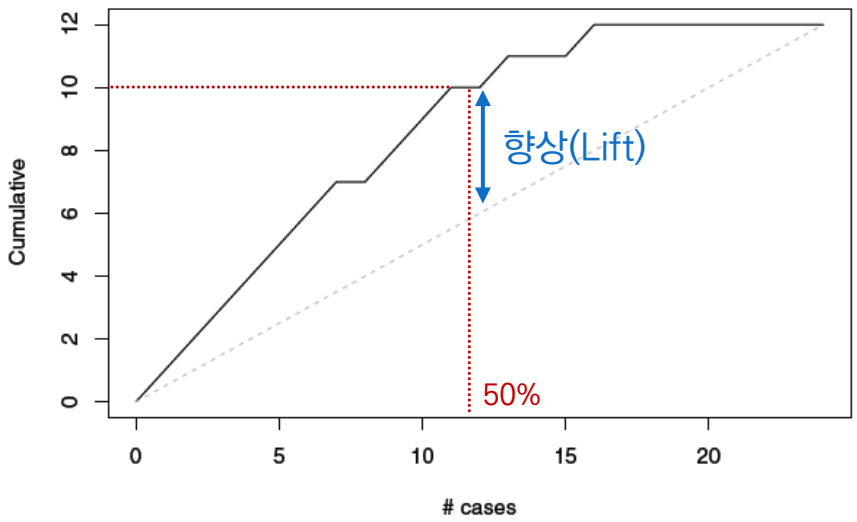
Interpreting the Cumulative Gain Chart

- e.g.) 올바른 분류 확률(50% 데이터)
  - ✓ 분류기: class 1, 10개 선택, 10/12의 확률
  - ✓ random: class 1, 6개 선택, 6/12의 확률

  → 1.8배 향상(Lift)

- 좋은 분류기: 적은 수의 데이터를 선택해도 높은 향상 정도 보장
- 왼쪽과 위쪽으로 치우칠수록 향상도 높음





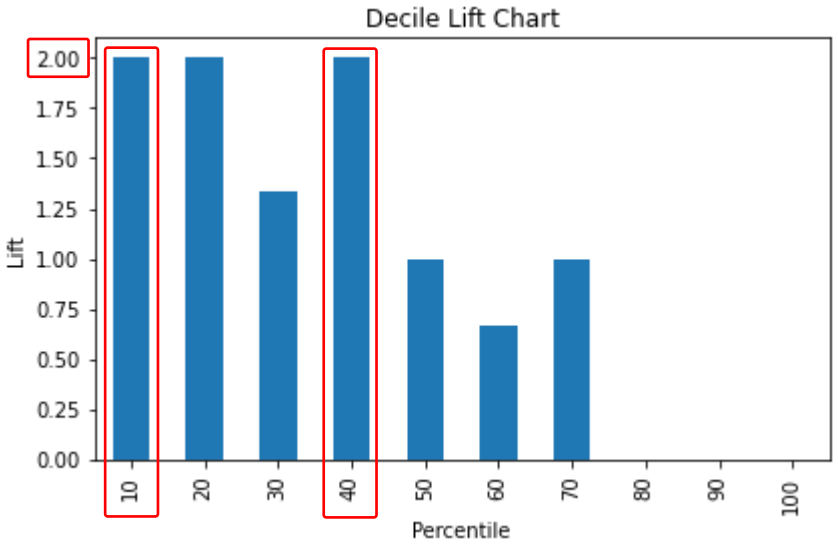
5.4 Judging Ranking Performance

[실습] Figure 5.7

Gains and Lift Charts for Binary Data

Decile Lift Chart

- 십분위 차트: 모든 향상 정보를 10개의 막대로 표현
- 왼쪽의 첫번째 막대: “(경향이 가장 커서) 가장 “1”이 될 가능성이 높다”고 랭크된 레코드의 10%를 취하면, 임의로 선택하는 것보다 두 배 많은 “1”을 얻을 수 있음
- 가장 큰 경향을 갖는 상위 40%의 레코드를 선택해도 여전히 임의로 하는 것보다 2배 좋은 성능 보장

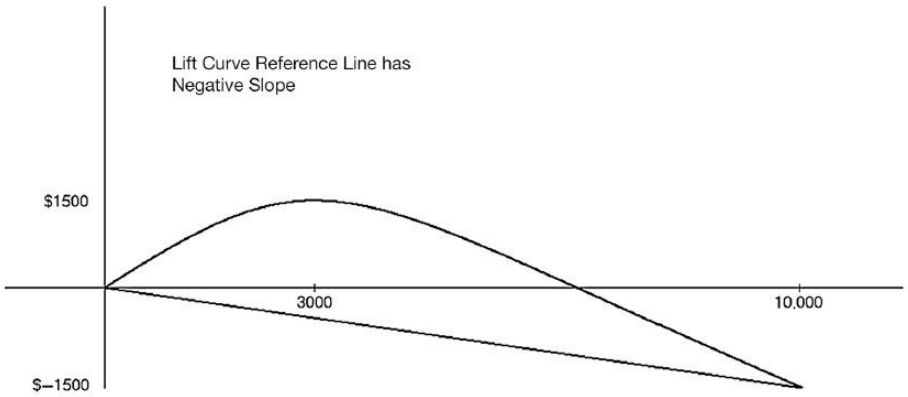


5.4 Judging Ranking Performance

Gains and Lift Charts Incorporating Costs and Benefit

비용/이익을 고려한 향상 차트 생성

1. 예측된 성공 확률값의 크기 순서대로 레코드를 정렬 (성공 = 관심 클래스에 속함)
2. 각 레코드에 대해 실제 결과값과 연관된 비용(이익) 기록
3. 리프트 곡선에서 가장 높은 경향(확률)을 가진 레코드(첫 행의 레코드)의 x 좌표 값은 1이고, y 좌표 값은 step 2에서 계산된 비용(이익)값
4. 다음 레코드에 대해서도 실제 결과값과 연관된 비용(이익)을 다시 계산. 이전 레코드의 비용(이익)에 다음 레코드의 비용(이익)을 더한다. [이 합계값=두번째 y 좌표값], [이때 x 좌표값=2]
5. 모든 레코드를 분석할 때까지 step 4 반복 수행. 모든 점들을 연결
6. 참조선은 첫번째(시작) 점에서 [y 좌표값=총 순이익], [x 좌표값=n(레코드의 총 개수)인 점까지 이은 직선

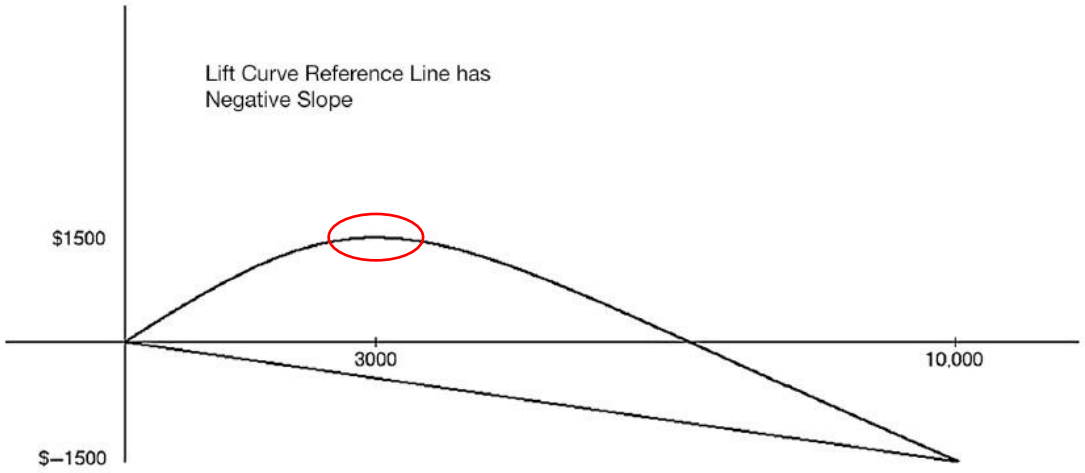


5.4 Judging Ranking Performance

Gains and Lift Charts Incorporating Costs and Benefit

비용/이익을 고려한 향상 차트

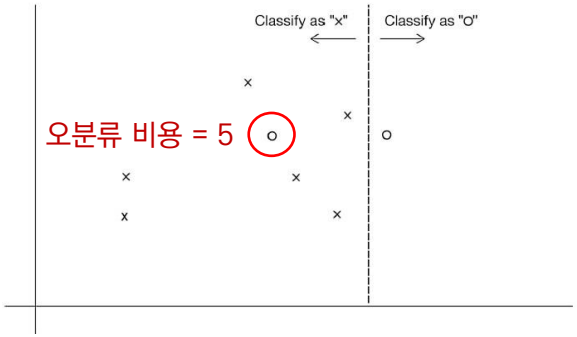
- 전체 데이터셋에 대한 순가치가 음수라면 비용과 이득이 음의 기울기를 갖도록 참조선이 그려질 수 있음
- 전체: 10,000명 / 메일링 비용 = $0.65 / 응답 이익 = $25 / 전체 응답비율 = 2%
- 기대 순가치 = (10,000*0.02*$25)−(10,000*$0.65) = $5,000 − $6,500 = − $1,500
- 최적 지점: 리프트 곡선이 최대일 때



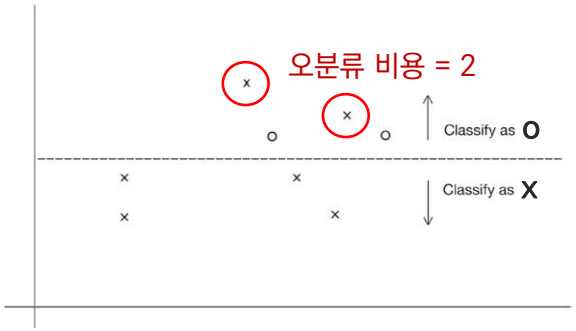
5.5 Oversampling

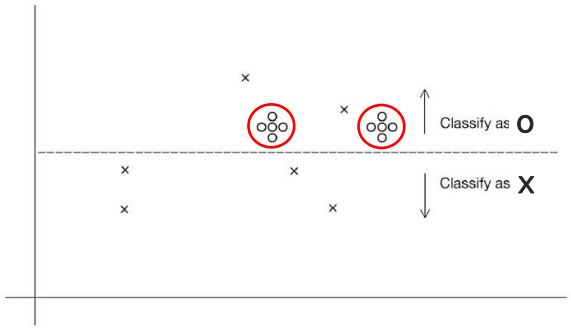
확대 샘플링

- 클래스들이 매우 불균등한 비율일 경우, 단순 무작위 샘플링을 사용하면 희귀한 클래스를 너무 적게 생성해서 우세한 클래스로부터 구별하는 것에 대한 유용한 정보를 얻지 못 할 수 있음
- 성층 샘플링(Stratified sampling), 가중 샘플링(Weighted sampling), 축소 샘플링(Undersampling) → 확대 샘플링(Oversampling) 용어 사용



- ex) x: 비응답자, o: 응답자 / x,y 축 = 예측변수
- 수직 점선: 동일한 비용의 가정 하에 최상의 분류결과, 하나의 오분류
- 만약 o를 x로 분류하는 비용=5, x를 놓치는 비용=1 → 오분류 비용 = 5 → 두 번째 그림에서는 오분류 비용 = 2
- 4개의 추가적인 o가 기존의 o에 더해지면 분류 알고리즘이 적절한 분류선을 자동으로 결정 → 단순한 무작위 샘플링으로 얻어진 것보다 적절한 결과 도출





5.5 Oversampling

Oversampling the Training Set

가중 샘플링(Weighted Sampling)

- 응답(yes)와 비응답(no)으로 데이터셋을 구분
- 응답(yes)에서 50%를 선택, 비응답도 같은 개수 선택해서 학습
- 나머지 응답 데이터를 검증 데이터로 활용
- 원래의 응답/비응답 비율을 유지하는 개수로 비응답 데이터를 검증 데이터로 활용
- 테스트 셋이 필요하다면 검증 데이터에서 무작위 추출 가능

Evaluating Model Performance Using Non-oversampled Validation Set

- 확대 샘플링된 데이터로 모델을 학습하더라도, 원래 데이터(즉, 확대 샘플링되지 않은 데이터)로 검증

5.5 Oversampling

Oversampling the Training Set

확대 샘플링된 검증 셋만 존재하는 경우의 모델 성능 평가

- 확대 샘플링된 데이터만 존재하는 경우에도 여전히 모델의 실제 데이터에 대한 분류 성능 평가 가능
- 하지만, 샘플링 과정에서 실제보다 적게 나타난 레코드의 클래스를 복구하기 위해 검증 셋의 재조정 필요

Adjusting the Confusion Matrix for Oversampling

- ex) 전체: 1,000 / 응답률 25배 확대 샘플링
  - ✓ 응답(1): 전체 데이터의 2%, 샘플의 50%
  - ✓ 비응답(0): 전체 데이터의 98%, 샘플의 50%
- 확대 샘플링 가중치(Oversampling weights)
  - ✓ 전체 데이터에서 응답 = 샘플에서 25명의 응답 가치(50/2)
  - ✓ 전체 데이터에서 비응답 = 샘플에서 0.5102명의 비응답 가치(50/98)
- 확대 샘플링 오분류율 = (80+110)/1,000 → 19%
- 재조정된 오분류율 = (3.2+215.6)/1,000 → 21.9%

[Oversampled Data (Validation)]

	Predict 0	Predict 1	Total
Actual 0	390	110	500
Actual 1	80	420	500
Total	470	530	1000

[Reweighted Data]

	Predict 0	Predict 1	Total
Actual 0	390/0.5102 =764.4	110/0.5102 =215.6	500
Actual 1	80/25 = 3.2	420/25 = 16.8	500