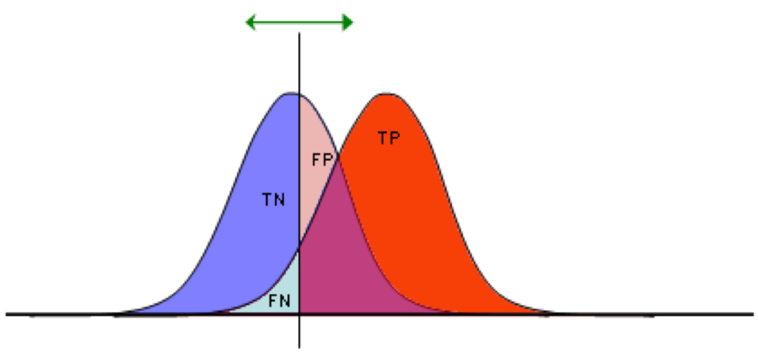
Classification performance – Part 2

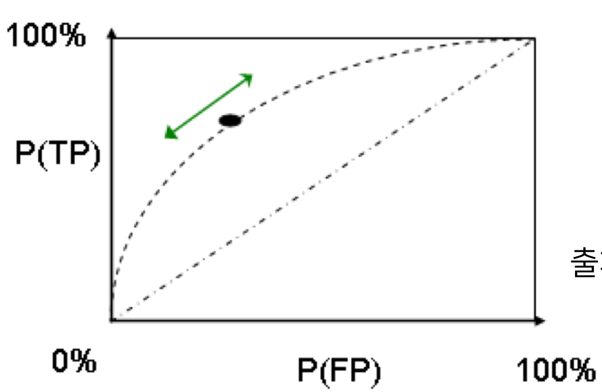
Taehoon Ko (thoon.koh@gmail.com)

Classification performance: ROC Curve

- Receiver operating characteristics (ROC) curve
  - Sort the records based on the P(positive class) in a descending order.
  - Compute the true positive rate and false positive rate by varying the cut-off.
  - Draw a chart where x & y axes are false & true positive rate, respectively.



TP	FP
FN	TN
1	1



출처: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Classification performance: Example 1 (Revisited)

- 예제1: 소매점에서 고객 구매 이력 데이터를 기반으로, 이 고객이 꾸준히 방문하는 VIP 고객인지 아닌지 예측하고자 함. (Test set = 10명의 고객)

X_1	…	X_p	Y
			1
			0
			1
			0
			0
			1
			0
			1
			1
			0

$$\Pr(Y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_p x_p)}}$$

$P(Y=1)$
0.97
0.15
0.54
0.58
0.24
0.75
0.42
0.80
0.45
0.70

Classification performance: Example 1 (Revisited)

- 데이터 포인트를 $P(Y = 1)$ 기준으로 내림차순 정렬

Y	$P(Y=1)$
1	0.97
0	0.15
1	0.54
0	0.58
0	0.24
1	0.75
0	0.42
1	0.80
1	0.45
0	0.70

Y	$P(Y=1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

Classification performance: Example 1 (Revisited)

- Cut-off value를 변화시키면서 True positive rate와 False positive rate를 계산

Classify as positive class

cut-off

Classify as negative class

Y	$P(Y=1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

Cut-off>0.97		Predicted class	
		1 (+)	0 (-)
Actual class	1 (+)	0	5
	0 (-)	0	5

- True positive rate (Sensitivity, Recall)
  = 0 / (0 + 5) = 0

- False positive rate (1-Specificity)
  = 0 / (0 + 5) = 0

Classification performance: Example 1 (Revisited)

- Cut-off value를 변화시키면서 True positive rate와 False positive rate를 계산

Y	$P(Y = 1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

Classify as positive class

cut-off

Classify as negative class

0.8<Cut-off<0.97

<table>
<tr><td colspan="2"></td><td colspan="2">Predicted class</td></tr>
<tr><td colspan="2"></td><td>1 (+)</td><td>0 (-)</td></tr>
<tr><td rowspan="2">Actual class</td><td>1 (+)</td><td>1</td><td>4</td></tr>
<tr><td>0 (-)</td><td>0</td><td>5</td></tr>
</table>

- True positive rate (Sensitivity, Recall)
= 1 / (1 + 4) = 0.2
- False positive rate (1-Specificity)
= 0 / (0 + 5) = 0

Classification performance: Example 1 (Revisited)

- Cut-off value를 변화시키면서 True positive rate와 False positive rate를 계산

Y	$P(Y = 1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

cut-off

0.75<Cut-off<0.8		Predicted class	
		1 (+)	0 (-)
Actual class	1 (+)	2	3
	0 (-)	0	5

- True positive rate (Sensitivity, Recall)
  = 2 / 5 = 0.4

- False positive rate (1-Specificity)
  = 0 / 5 = 0

Classification performance: Example 1 (Revisited)

- Cut-off value를 변화시키면서 True positive rate와 False positive rate를 계산

Y	$P(Y=1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

cut-off

0.58<Cut-off<0.7		Predicted class	
		1 (+)	0 (-)
Actual class	1 (+)	3	2
	0 (-)	1	4

- True positive rate (Sensitivity, Recall) = 3 / 5 = 0.6
- False positive rate (1-Specificity) = 1 / 5 = 0.2

Classification performance: Example 1 (Revisited)

- Cut-off value를 변화시키면서 True positive rate와 False positive rate를 계산

Y	$P(Y=1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

cut-off

0.45<Cut-off<0.54		Predicted class	
		1 (+)	0 (-)
Actual class	1 (+)	4	1
	0 (-)	2	3

- True positive rate (Sensitivity, Recall) = 4 / 5 = 0.8
- False positive rate (1-Specificity) = 2 / 5 = 0.4

Classification performance: Example 1 (Revisited)

- Cut-off value를 변화시키면서 True positive rate와 False positive rate를 계산

Y	$P(Y = 1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

cut-off

Cut-off < 0.15		Predicted class	
		1 (+)	0 (-)
Actual class	1 (+)	5	0
	0 (-)	5	0

- True positive rate (Sensitivity, Recall) = 5 / 5 = 1

- False positive rate (1-Specificity) = 5 / 5 = 1

Classification performance: Example 1 (Revisited)

- Cut-off value를 변화시키면서 True positive rate와 False positive rate를 계산

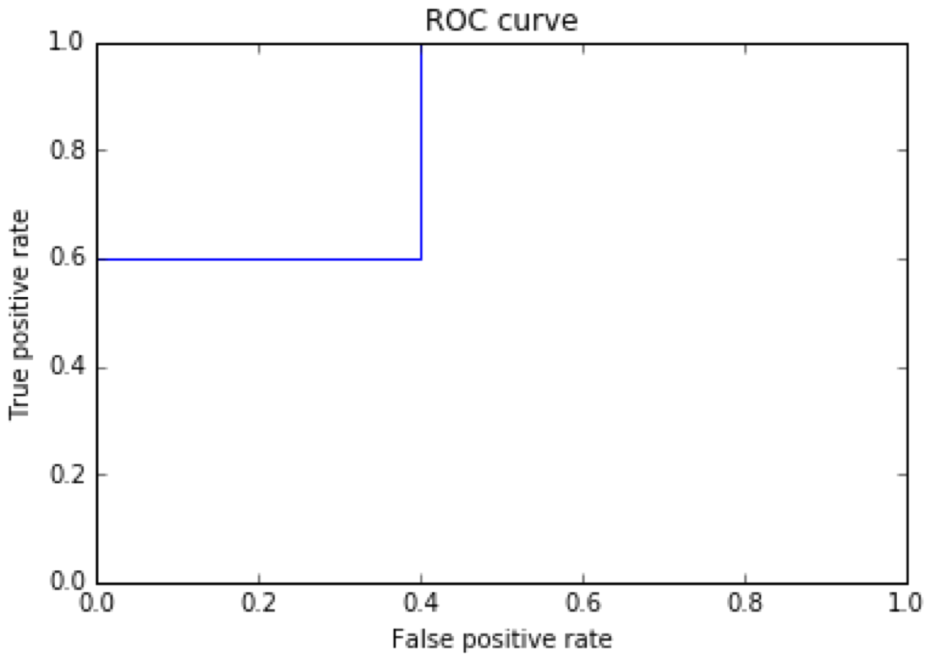
Y	$P(Y=1)$
1	0.97
1	0.8
1	0.75
0	0.7
0	0.58
1	0.54
1	0.45
0	0.42
0	0.24
0	0.15

TPR	FPR
0	0
0.2	0
0.4	0
0.6	0
0.6	0.2
0.6	0.4
0.8	0.4
1	0.4
1	0.6
1	0.8
1	1

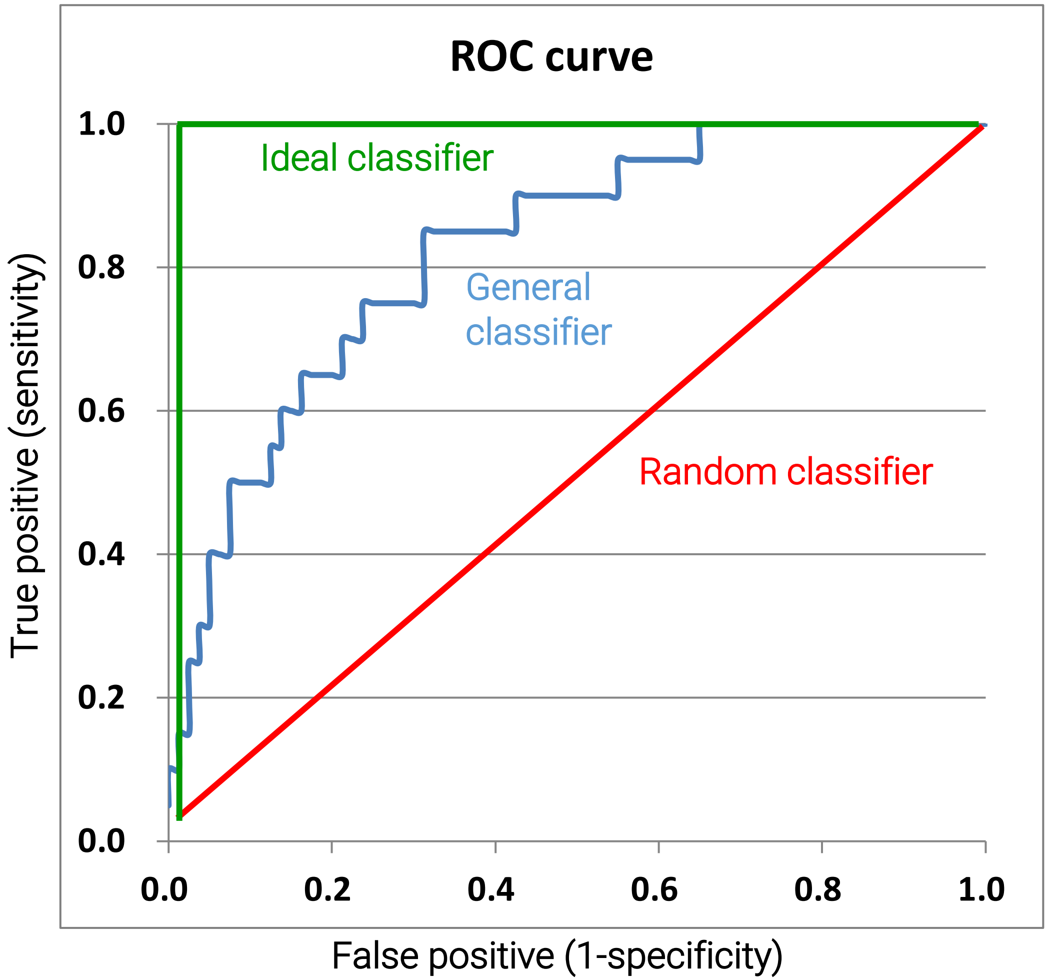
Classification performance: Example 1 (Revisited)

- Draw ROC curve

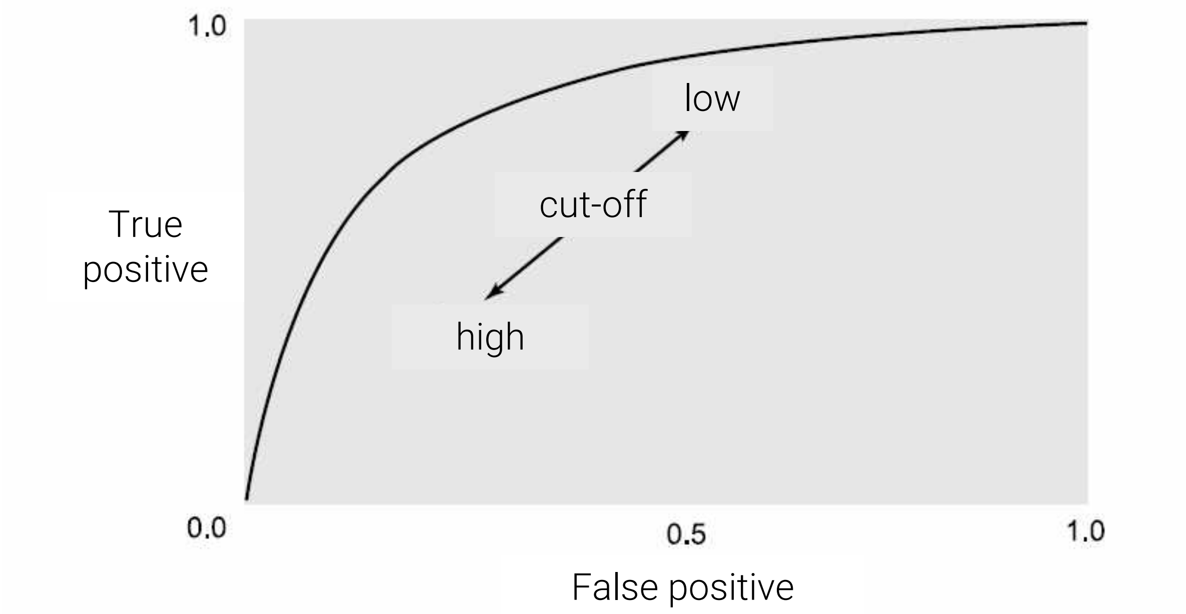
```
%matplotlib inline
from matplotlib import pyplot as plt

tpr = [0,0.2,0.4,0.6,0.6,0.6,0.8,1,1,1,1]
fpr = [0,0,0,0,0.2,0.4,0.4,0.4,0.6,0.8,1]

plt.plot(fpr,tpr)
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.show()
```

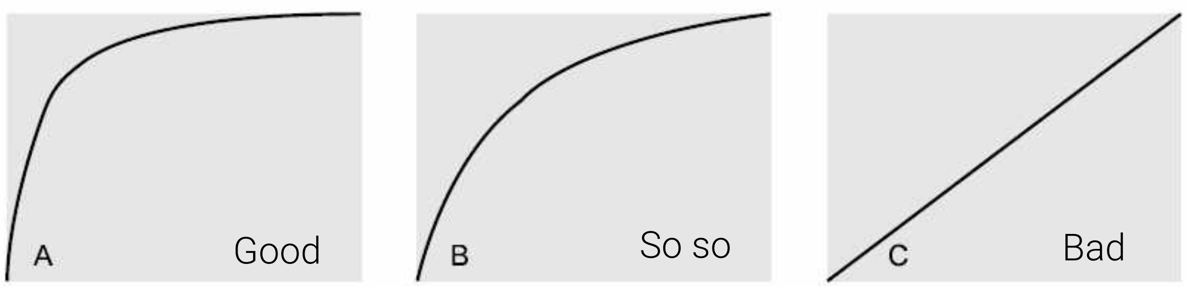


Classification performance: ROC Curve



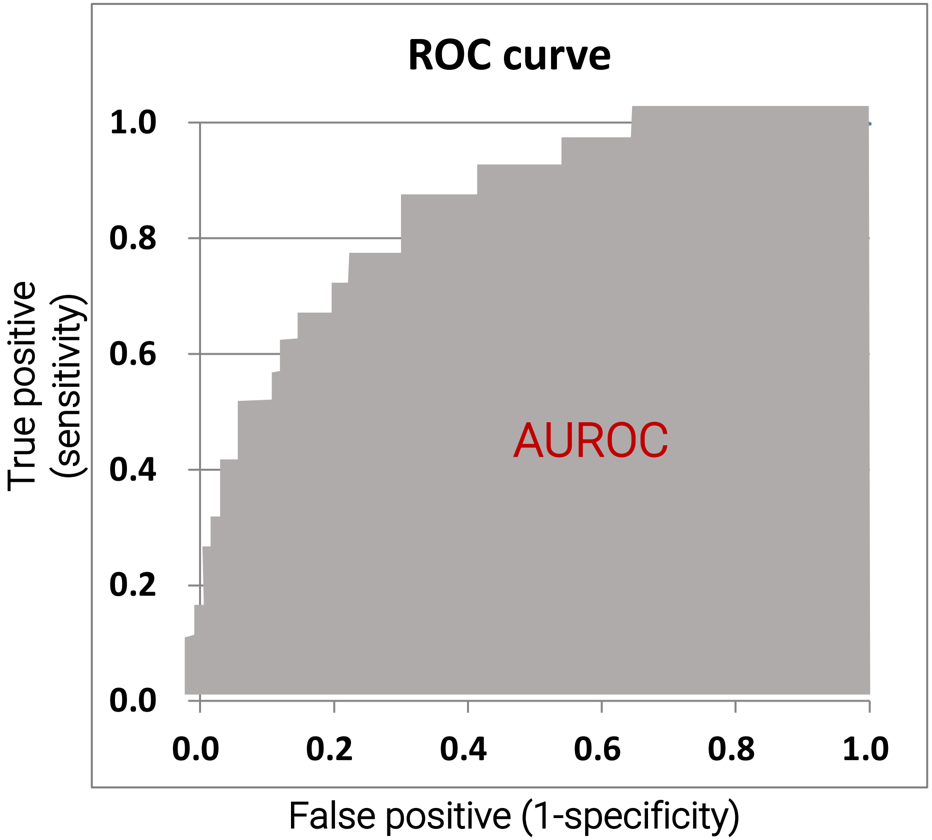
Classification performance: ROC Curve





Classification performance: AUROC

- Area under ROC curve (AUROC or AUC)
  - ROC curve 아래의 면적
  - Ideal classifier: AUROC = 1
  - Random classifier: AUROC = 0.5
  - In general, 0.5 < AUROC < 1
  - AUROC가 클 수록 분류 모델의 성능이 좋음.



Classification performance: Profit and cost

- Asymmetric error costs
  - 두 가지 형태의 error costs
    - Positive class인 포인트들을 negative class로 잘못 분류했을 때의 cost
    - Negative class인 포인트들을 positive class로 잘못 분류했을 때의 cost
  - 일반적으로 positive class인 포인트들을 잘못 분류했을 때의 cost가 그 반대의 경우보다 크다.
    - ex) 암 진단, 보험 사기 탐지, VIP 고객 탐지, 제품 불량 예측 등.
- Profits
  - 포인트들을 제대로 분류했을 때 발생하는 profit
  - 일반적으로 positive class인 포인트들을 잘 분류했을 때의 profit이 그 반대의 경우보다 크다.

Classification performance: Profit and cost

- Example: Response to promotional offer
  - Suppose we send an offer to 1000 people, with 1% average response rate (“1” = response, “0” = non-response).
  - “Naïve rule”: Classify everyone as “0”.

Confusion Matrix		Predicted	
		1	0
Actual	1	0	10
	0	0	990

  - Misclassification error = 1%
  - Accuracy = 99%.

출처: 강필성 교수 강의노트

Classification performance: Profit and cost

- Example: Response to promotional offer
  - Classification model

Confusion Matrix		Predicted	
		1	0
Actual	1	8	2
	0	20	970

  - Misclassification error = 2.2%
  - Accuracy = 97.8%

출처: 강필성 교수 강의노트

Classification performance: Profit and cost

- Consider profits and costs.
  - Assign profit/cost for each cell of confusion matrix.
  - Example:
    - $10: net profit for the responders if the offer is sent.
    - $10: net cost for not sending offer for the responders.
    - $1: net cost for sending an offer.

<table>
<tr><td colspan="2" rowspan="2">Confusion Matrix</td><td colspan="2">Predicted</td></tr>
<tr><td>1</td><td>0</td></tr>
<tr><td rowspan="2">Actual</td><td>1</td><td>$9</td><td>-$10</td></tr>
<tr><td>0</td><td>-$1</td><td>0</td></tr>
</table>

  - Total profit for the naïve rule: 10*(-$10) = -$100
  - Total profit for classification model: 8*($9)+2*(-$10)+20*(-$1) = **$32* (Best)**

출처: 강필성 교수 강의노트

Classification performance: Profit and cost

- Profit과 cost를 정확히 할당할 수 있는가?
  - 매우 어려운 문제.
    - ex) 암 예측

Confusion Matrix		Predicted	
		1	0
Actual	1	Reduce diagnosis cost / Save one's life	Increase diagnosis cost / Lose one's lfe
	0	Misdiagnosis cost	0

  - 경제학 등 일부 분야에서는 이러한 profit과 cost를 잘 정의하여 모델의 성능을 평가하는 경우도 있음

출처: 강필성 교수 강의노트

Class 별 cost를 다르게 주는 방법

- Class별 weight / cost를 줘서 모델링하는 경우
  - 예제: 암 환자 10명, 정상 환자 990명 ➔ Class-imbalanced data
  - 암 환자에 대해서 더 큰 가중치를 부여하여, 모델링에 반영하는 방법

- In scikit-learn,
  - Classifier 클래스 중에 parameter로 [class_weight]라는 것이 있는 경우, 각 클래스에 다른 가중치를 주는 것이 가능

Class 별 cost를 다르게 주는 방법

- Example:
  - http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

class_weight : dict or 'balanced', default: None

Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one.

The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

New in version 0.17: class_weight='balanced' instead of deprecated *class_weight='auto'*.