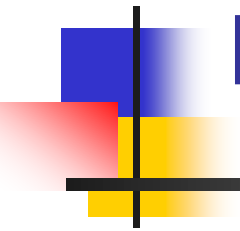


Data Science: Evaluation-2

Won Kim

2022

Roadmap: End-to-End Process

- 1. Objective Setting
- 2. Data Curation
- 3. Data Inspection
- 4. Data Preparation
- 5. Data Analysis
- 6. Evaluation
- 7. Deployment

Roadmap: Evaluation

- Motivation
- Evaluation Methods
- Ensemble Learning
- Evaluation Metrics

Roadmap: Evaluation Metrics

- For Regression
- For Classification

Acknowledgments

- http://homedir.jct.ac.il/~rosenfa/mining/auc.pptx
- https://www.cs.waikato.ac.nz/ml/weka/slides/Chapter5.pptx
- https://www.slideshare.net/AndrewFerlitsch/machine-learning-accuracy-and-confusion-matrix
- https://www.python-course.eu/confusion_matrix.php
- https://www.dataquest.io/blog/understanding-regression-error-metrics/

Data Science Terminology

- feature, attribute, predictor, observation, independent variable
  - in RDB/Excel: column, attribute
- sample, observation, event, case, tuple
  - In RDB/Excel: record, tuple, row
- label, class, target, response, dependent variable
  - (added to a record/row by a human)
- loss function (function for computing estimated error in prediction vs. actual value)
  - also called error function, cost function, objective function

Classifier vs. Regressor

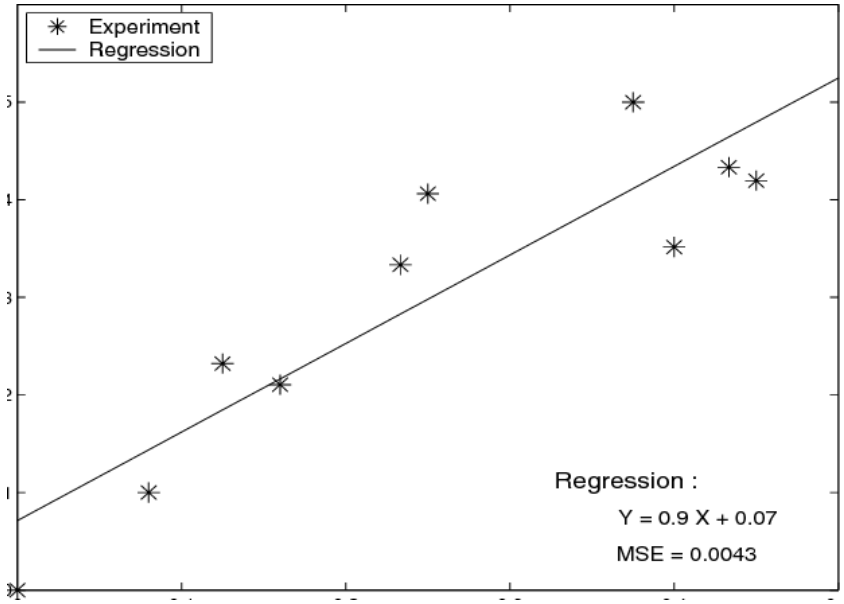
- Regressor outputs a continuous value
- Classifier outputs a discrete value

Performance of a Regression Model (1/2)

- Accuracy is measured as a loss function between the correct (expected) result and predicted result.
  - (e.g.) mean square error
- The objective is to minimize the loss when fitting a regression line.

Performance of a Regression Model (2/2)

- Minimize the mean squared error (MSE) cost function



$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

* n is the number of data points
* Y_i represents observed values
* $\hat{Y}_i$ represents predicted values

Measures of Regression Error (1/5)

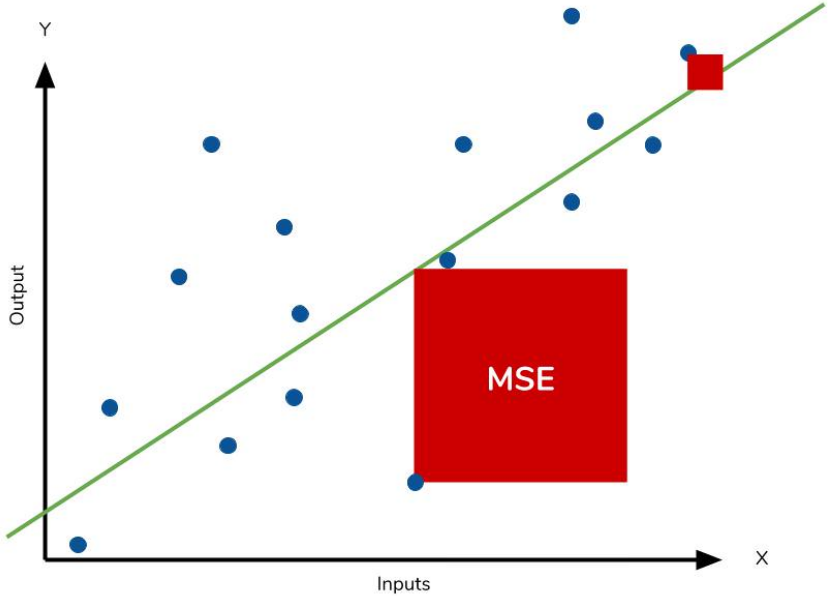
- MSE: Mean-squared-error
  - square the errors and find their average

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

- RMSE (root-mean-squared-error): square the errors, find their average, take the square root
  - Square root converts MSE back to the units used for the original output (y) values
- For both MSE and RMSE, outliers make errors bigger

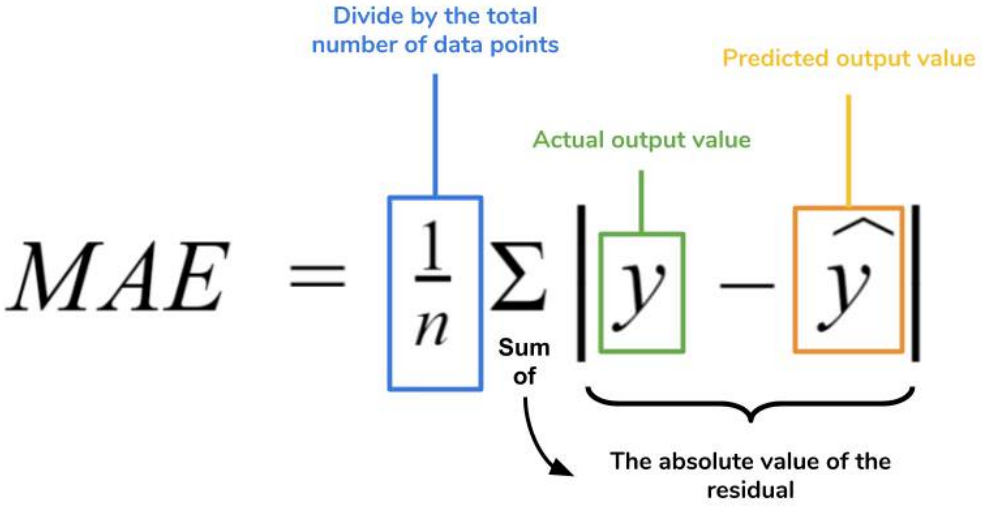
Measures of Regression Error (1/5)

- MSE (cont'd)
  - Large residuals get punished more (because of the square term)



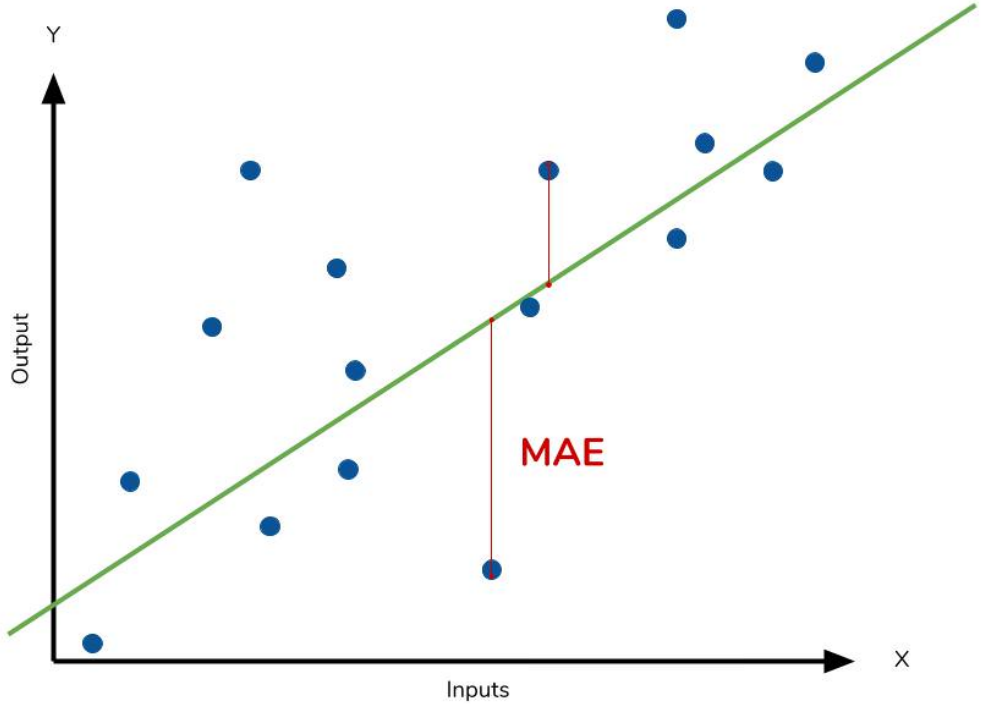
Measures of Regression Error (2/5)

- MAE: Mean Absolute Error
  - Ignores +/- signs of the errors
  - Gives an idea of the magnitude of errors



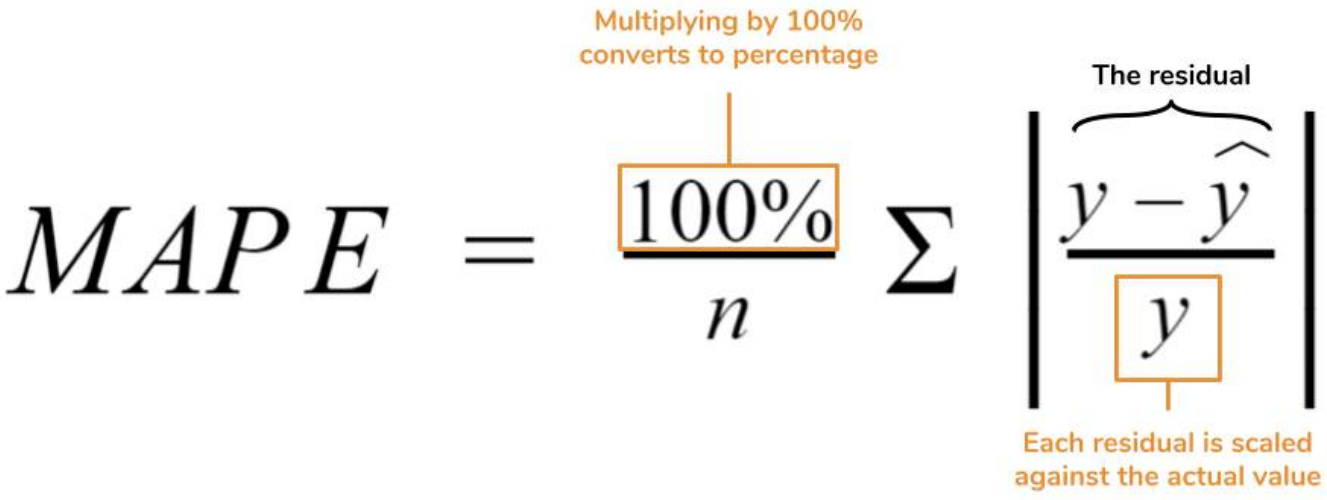
Measures of Regression Error (2/5)

- MAE (cont'd)



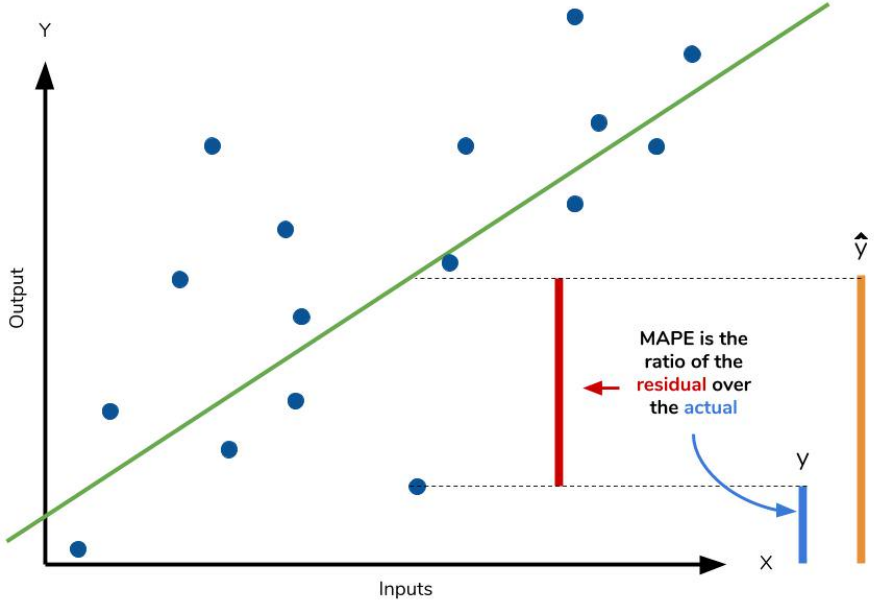
Measures of Regression Error (3/5)

- MAPE: Mean Absolute Percentage Error
  - percentage equivalent of MAE
  - Percentage is easier for people to comprehend
  - Less affected by the outliers (than MAE) because of the use of absolute residual



Measures of Regression Error (3/5)

- MAPE (cont'd)



- Problematic when y is 0 or very small (then the ratio becomes big)

Measures of Regression Error (3/5)

- MAPE (cont'd)
  - Biased towards predictions that are less than the actual values

$$MAPE = \frac{100\%}{n} \sum \left| \frac{y - \hat{y}}{y} \right|$$

$\hat{y}$ is smaller than the actual value

n = 1 $\hat{y}$ = 10 y = 20

MAPE = 50%

$\hat{y}$ is greater than the actual value

n = 1 $\hat{y}$ = 20 y = 10

MAPE = 100%

Measures of Regression Error (4/5)

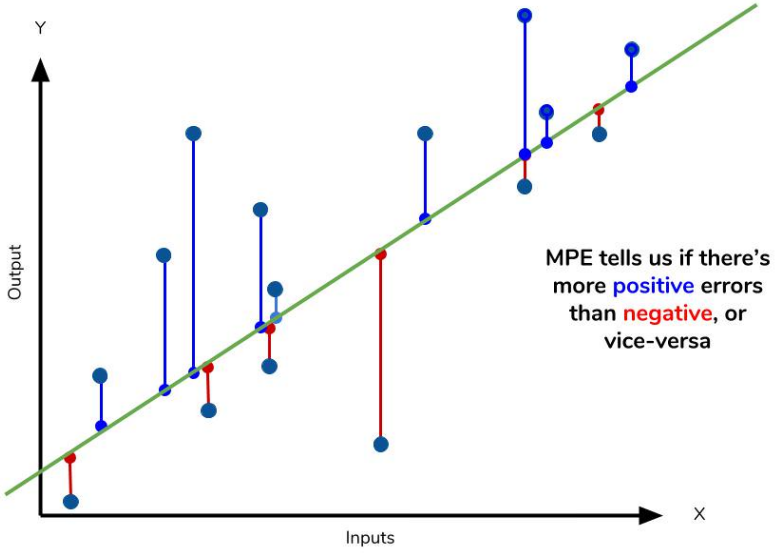
- MPE: Mean Percentage Error
  - MPE is MAPE minus the absolute value operation.

$$MPE = \frac{100\%}{n} \Sigma \left(\frac{y - \hat{y}}{y} \right)$$

  - Average Error: MPE without the percentage
    - (similar to MAE vs MAPE)
  - With MPE and Average Error, positive and negative errors cancel out, so we cannot talk about the model prediction performance.

Measures of Regression Error (4/5)

- MPE (cont'd)
  - However, if there are more negative or positive errors, this bias will show up in the MPE.
  - So it allows us to see if the model systematically underestimates (more negative errors) or overestimates (more positive errors)



Measures of Regression Error (5/5)

- Total SSE: total sum of squared errors

SSR = $\Sigma\,(\hat{y} - \bar{y})^2$ (measure of explained variation)

SSE = $\Sigma\,(y - \hat{y})^2$ (measure of unexplained variation)

SST = SSR + SSE = $\Sigma\,(y - \bar{y})^2$ (measure of total variation in y)

Roadmap: Evaluation Metrics

- For Regression
- For Classification

Roadmap: Measures for Classification

- Confusion Matrix
- ROC

Confusion Matrix (classification matrix)

- Summarizes the correct and incorrect classifications that a classifier produced for a dataset
- Rows and columns of the matrix correspond to the true and predicted classes, respectively.
- Calculate a confusion matrix for many different output thresholds (e.g., 0.1, 0.2 ⋯0.9 for “yes” or “no”).

Confusion Matrix

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

- TP and TN are the number of correct classifications.
- FP and FN are the number of incorrect classifications.

Example Confusion Matrix

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

201 1's correctly classified as "1"

85 1's incorrectly classified as "0"

25 0's incorrectly classified as "1"

2689 0's correctly classified as "0"

Error Rate

- accuracy = (TP+TN)/N
- miscalculation = (FP+FN)/N

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

Overall error rate = (25+85)/3000 = 3.67%

Accuracy = 1 – err = (201+2689) = 96.33%

If multiple classes, error rate is

(sum of misclassified records)/(total records)

Performance Measures Computed from the Confusion Matrix (1/2)

- precision = positive predictive value (PPV)
  = TP/(TP+FP)
  /* first column of confusion matrix */
  /* all positive predictions */
- recall = hit rate = sensitivity = true positive rate (TPR)
  = TP/(TP+FN) = 1 – FNR
  /* first row of confusion matrix */
  /* all positive predictions for actual True */
- F measure = harmonic mean of precision and recall
  = 2 x precision x recall / (precision + recall)

Performance Measures Computed from the Confusion Matrix (2/2)

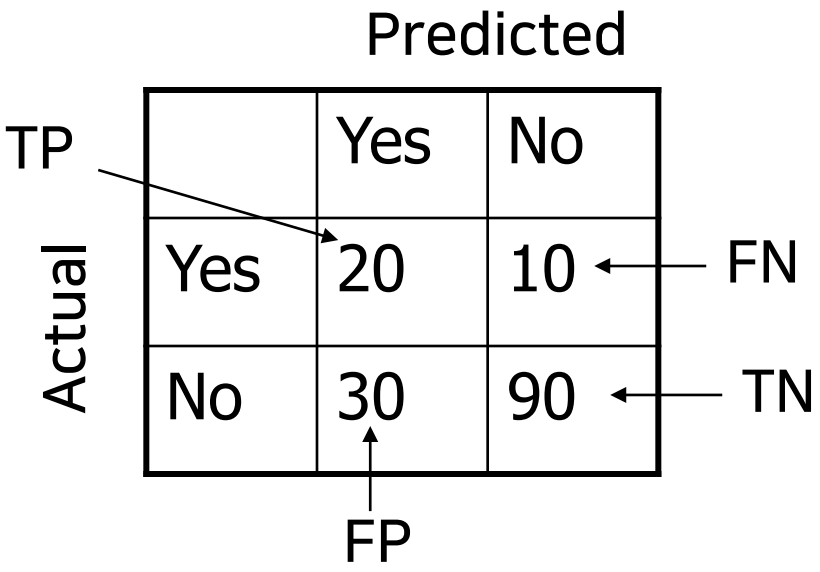
- false positive rate (FPR) = 1 - TNR

  FP/(FP+TN)
- specificity = selectivity = true negative rate (TNR) =

  TN/(TN+FP) = 1 - FPR

Example

Confusion Matrix



Precision = 20 / 50 = 0.4
Recall = 20 / 30 = 0.666
F-measure=2 x .4 x .666 / 1.0666 = .5

Generalizing the Confusion Matrix(1/2)

- 3-state example
- (e.g.) 25 animals: 7 cats, 8 dogs, 10 snakes.

	predicted			
actual		dog	cat	snake
	dog	6	2	0
	cat	1	6	0
	snake	1	1	8

- (e.g.) correctly predicted 6 of 8 actual dogs, but mistook 2 dogs as cats
- All correct predictions are located in the diagonal of the matrix

Generalizing the Confusion Matrix(2/2)

- Precision and Recall for the Multi-Class Case

$$\text{precision}_i = M_{ii} / \Sigma_j M_{ji}$$
$$\text{recall}_i = M_{ii} / \Sigma_j M_{ij}$$

- The precision for the example

  $\text{precision}_{dogs}=6/(6+1+1)=3/4=0.75$

  $\text{precision}_{cats}=6/(2+6+1)=6/9=0.67$

  $\text{precision}_{snakes}=8/(0+0+8)=1$

- The recall for the example

  $\text{recall}_{dogs}=6/(6+2+0)=3/4=0.75$

  $\text{recall}_{cats}=6/(1+6+0)=6/7=0.86$

  $\text{recall}_{snakes}=8/(1+1+8)=4/5=0.8$

	predicted			
actual		dog	cat	snake
	dog	6	2	0
	cat	1	6	0
	snake	1	1	8

Practical Perspectives (1/4)

- Algorithms optimized for accuracy tend to guess in favor of the preponderant class and be wrong most of the time with the minor classes.
- Precision and Recall, and F1, solve this problem.

Practical Perspectives (2/4)

- Precision
  - Measures 'exactness': % of times prediction was right
  - Useful when trying to minimize false positives
  - (e.g.) % of patients who really have cancer among those diagnosed (predicted) as having cancer

    (If you diagnosed 10 patients as having cancer (TP+FP), and 9 really have cancer (TP), precision is 90%.)
  - Problem
    - If you do not diagnose cancer in a patient who really has cancer, or you diagnose a healthy patient (FP).
    - Recall helps.

Practical Perspectives (3/4)

- Recall
  - Measures 'completeness': % of times prediction was right (true positive) among an entire class (actual positive)
  - Useful when trying to minimize false negatives
  - (e.g.) % of patients diagnosed as having cancer among all patients diagnosed

    (If you diagnosed 9 patients as having cancer, out of 20 patients diagnosed, recall is 45%.)
  - Problem
    - You can be accurate but have a low recall, or have a high recall but lose accuracy

Practical Perspectives (4/4)

- F1 score
  - Harmonic mean of precision and recall
  - Can maximize precision and recall together
  - Indicates a balance between precision and recall
  - It ranges from 0 to 1
    - Best value is 1 (perfect precision and recall) and worst is 0.

** Harmonic Mean

- Harmonic mean is the reciprocal of the average of the reciprocals @@)
- Harmonic mean = n / (1/a + 1/b + 1/c + ⋯)

  where a, b, c,.. are values and n is how many values
- Calculation Steps
  - Calculate the reciprocal (1/value) for every value.
  - Find the average of those reciprocals (add them and divide by how many there are)
  - Then do the reciprocal of that average (=1/average)

** Harmonic Mean: Example

- Harmonic mean of 1, 2 and 4
  - Find the reciprocals of 1, 2 and 4

    1/1 = 1, 1/2 = 0.5, 1/4 = 0.25
  - Add them up

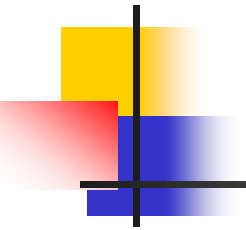
    1 + 0.5 + 0.25 = 1.75
  - Divide by how many

    average = 1.75/3
  - The reciprocal of that average is our answer

    Harmonic Mean = 3/1.75 = 1.714 (to 3 places)

Exercise 1

- A computer program for recognizing dogs in photographs identifies 8 dogs in a picture containing 12 dogs and some cats.
- Of the 8 identified as dogs, 5 actually are dogs (true positives), while the rest are cats (false positives).
- What is the program's precision?
- What is the program's recall?

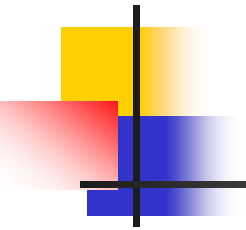


Answers

- The program's precision is 5/8
- The program’s recall is 5/12.

Exercise 2

- A search engine returns 30 pages; only 20 of the 30 pages were relevant. It failed to return 40 additional relevant pages.
- What is the search engine's precision?
- What is the search engine's recall?



Answers

- The search engine’s precision is 20/30 = 2/3.
- The search engine’s recall is 20/60 = 1/3.
- In this case, precision is "how useful the search results are", and recall is "how complete the results are".

Estimated (Desired) Accuracy Varies with the Validation Dataset Size

	Err							
	0.01	0.05	0.10	0.15	0.20	0.30	0.40	0.50
± 0.025	250	504	956	1,354	1,699	2,230	2,548	2,654
± 0.010	657	3,152	5,972	8,461	10,617	13,935	15,926	16,589
± 0.005	2,628	12,608	23,889	33,842	42,469	55,741	63,703	66,358

- columns: error (miscalculation) rate
- rows: estimated (desired) accuracy

- Example
  - If we want desired accuracy within (+/-) 0.01 of miscalculation rate 0.05, we need a validation dataset with 3,152 records.

Cutoff for Classification

- Most data mining algorithms classify via a 2-step process:
  - For each record,

    1. Compute probability of belonging to class "1".
    2. Compare to cutoff value, and classify accordingly.
- Default cutoff value is 0.50.

  If >= 0.50, classify as "1"

  If < 0.50, classify as "0"
- Typically, error rate is lowest for cutoff = 0.50
- But often we can use different cutoff values.

Example Cutoff Table

Actual Class	Probability of 1
1	0.995976726
1	0.987533139
1	0.984456382
1	0.980439587
1	0.948110638
1	0.889297203
1	0.847631864
0	0.762806287
1	0.706991915
1	0.680754087
1	0.656343749
0	0.622419543

Actual Class	Probability of 1
1	0.505506928
0	0.47134045
0	0.337117362
1	0.21796781
0	0.199240432
0	0.149482655
0	0.047962588
0	0.038341401
0	0.024850999
0	0.021806029
0	0.016129906
0	0.003559986

- The table contains the actual class (e.g., "iPhone owner/non-owner) for 24 records, sorted by the probability that the record will be classified as 1.

Example Confusion Matrix (for Cutoff 0.5)

Cut off Prob.Val. for Success (Updatable)	0.5

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	11	1
non-owner	2	10

Example Confusion Matrix (for Cutoff 0.25 and 0.75)

- With cutoff of 0.25, we classify more records as 1's and the misclassification rate goes up (more 0's misclassified as 1's) to 5/24
- With cutoff of 0.75, we classify fewer records as 1's and the misclassification rate goes up (more 1's misclassified as 0's) to 6/24

Cut off Prob.Val. for Success (Updatable)	0.25

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	11	1
non-owner	4	8

Cut off Prob.Val. for Success (Updatable)	0.75

Classification Confusion Matrix		
	Predicted Class	
Actual Class	owner	non-owner
owner	7	5
non-owner	1	11

Problems with Accuracy

- Example: mammography
  - The disease occurs < 1%.
  - So let a trained monkey always say there is no disease (without even looking at the film/image)!
  - The monkey's accuracy is (0+99)/(0+99+0+1) = 99% without even going to medical school!
- Need a different measure!

When One Class is More Important

- In many cases it is more important to identify members of one class
  - tax fraud
  - credit default
  - response to promotional offer
  - detecting electronic network intrusion
  - predicting delayed flights
- In such cases, accept greater overall error, in return for better identifying the important class for further attention

Confusion Matrix for 2 Classes

		Predicted class	
		C_0	C_1
Actual class	C_0	$n_{0,0}$= C0 correctly classified as C0	$n_{0,1}$= C_0 incorrectly classified as C_1
	C_1	$n_{1,0}$= C_1 incorrectly classified as C_0	$n_{1,1}$= C_1 correctly classified as C_1

Alternate Accuracy Measures

If "C_1" is the important class

- Sensitivity = % of "C_1" class correctly classified

  $$\text{Sensitivity} = n_{1,1} / (n_{1,0} + n_{1,1})$$

- Specificity = % of "C_0" class correctly classified

  $$\text{Specificity} = n_{0,0} / (n_{0,0} + n_{0,1})$$

- False positive rate = % of predicted "C_1's" that were not "C_1's"
- False negative rate = % of predicted "C_0's" that were not "C_0's"

Confusion Matrix
Using Pandas and Scikit-Learn

Acknowledgments

- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
- https://datatofish.com/confusion-matrix-python/
- https://www.python-course.eu/confusion_matrix.php

Basic Function: actual vs predicted Matrixes

```
>>> from sklearn.metrics import confusion_matrix
>>> y_true = [2, 0, 2, 2, 0, 1]
# y_true is a matrix of actual (target) values

>>> y_pred = [0, 0, 2, 2, 0, 2]
# y_pred is a matrix of data predicted by a classifier

>>> confusion_matrix(y_true, y_pred)

array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]])
```

	pred-0	pred-1	pred-2
true-0	2	0	0
true-1	0	0	1
true-2	1	0	2

Mapping the Two Matrixes to the Confusion Matrix

A confusion matrix C is such that $C_{i,j}$ is equal to the number of observations known to be in group i but predicted to be in group j.

Thus in binary classification, the count of true negatives is $C_{0,0}$,
false negatives is $C_{1,0}$,
true positives is $C_{1,1}$ and
false positives is $C_{0,1}$.

Extracting the 4 Entries of a 2-State Confusion Matrix

```
>>> tn, fp, fn, tp = confusion_matrix([0, 1, 0, 1],
                [1, 1, 1, 0]).ravel()
>>> (tn, fp, fn, tp)

(0, 2, 1, 1)
```

	pred-0	pred-1
true-0	0	2
true-1	1	1

Numpy.ravel() function

- The ravel() function is used to create a contiguous flattened array.

```
>>> import numpy as np
>>> x = np.array([[1, 2, 3], [4, 5, 6]])
>>> print(np.ravel(x))
[1 2 3 4 5 6]

 # ravel “order”:
       C (default, ‘C’, row-major order),
       F (‘Fortran’, column-major order),
       A, K
>>> print(np.ravel(x, order='F'))
[1 4 2 5 3 6]
```

Example: Dataset

y_Predicted	y_Actual
1	1
1	0
0	0
1	1
0	0
1	1
1	0
0	0
1	1
0	0
0	1
0	0

Create a Pandas DataFrame

```
import pandas as pd

data = {'y_Predicted': [1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0],
        'y_Actual':    [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0]}

df = pd.DataFrame(data,
                  columns=['y_Actual','y_Predicted'])
print (df)
```

Result of Running the Code

	y_Predicted	y_Actual
0	1	1
1	1	0
2	0	0
3	1	1
4	0	0
5	1	1
6	1	0
7	0	0
8	1	1
9	0	0
10	0	1
11	0	0

Create the Confusion Matrix

```
confusion_matrix = pd.crosstab(df['y_Actual'],
    df['y_Predicted'], rownames=['Actual'],
    colnames=['Predicted'])
print (confusion_matrix)
```

Full Python Code

```
import pandas as pd

data = {'y_Predicted': [1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0],
        'y_Actual':    [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0]}

df = pd.DataFrame(data,
        columns=['y_Actual','y_Predicted'])

confusion_matrix = pd.crosstab(df['y_Actual'],
    df['y_Predicted'], rownames=['Actual'],
    colnames=['Predicted'])
print (confusion_matrix)
```

Resulting Confusion Matrix

Predicted	0	1
Actual		
0	5	2
1	1	4

Displaying the Confusion Matrix Using Seaborn

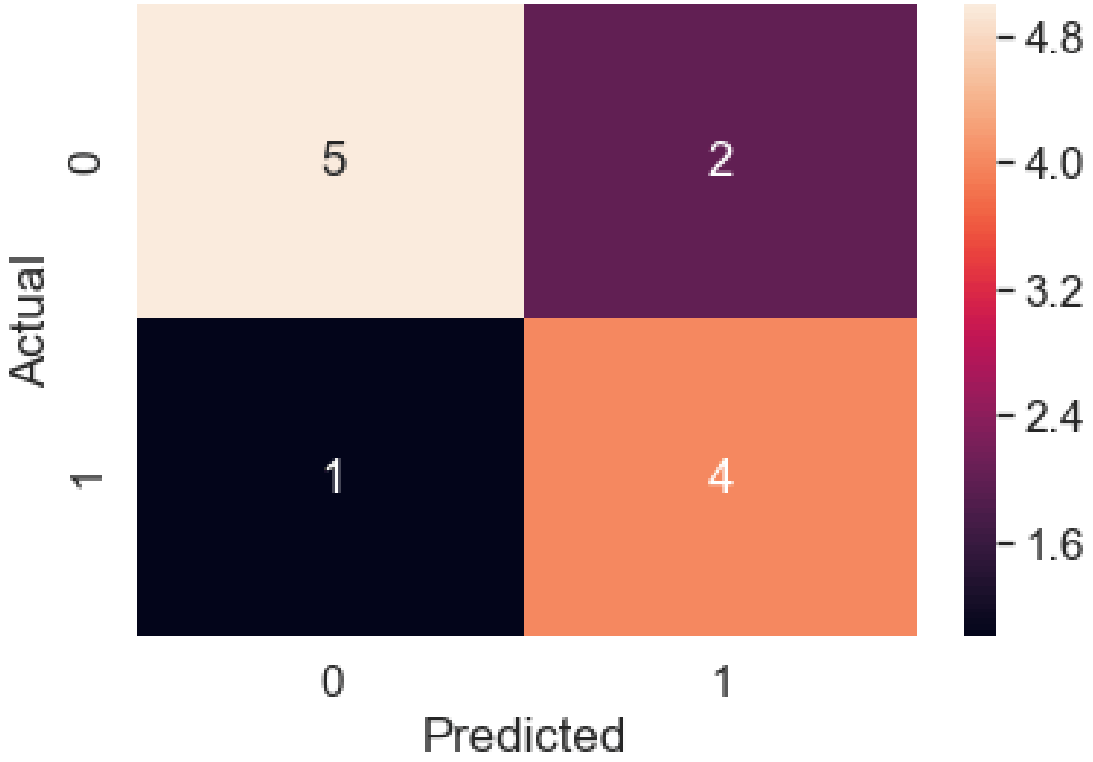
```
import pandas as pd
import seaborn as sn

data = {'y_Predicted': [1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0],
        'y_Actual':    [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0]}

df = pd.DataFrame(data,
        columns=['y_Actual','y_Predicted'])
confusion_matrix = pd.crosstab(df['y_Actual'],
        df['y_Predicted'], rownames=['Actual'],
        colnames=['Predicted'])

sn.heatmap(confusion_matrix, annot=True)
```

Resulting Display



Adding Totals at the Margins of the Confusion Matrix

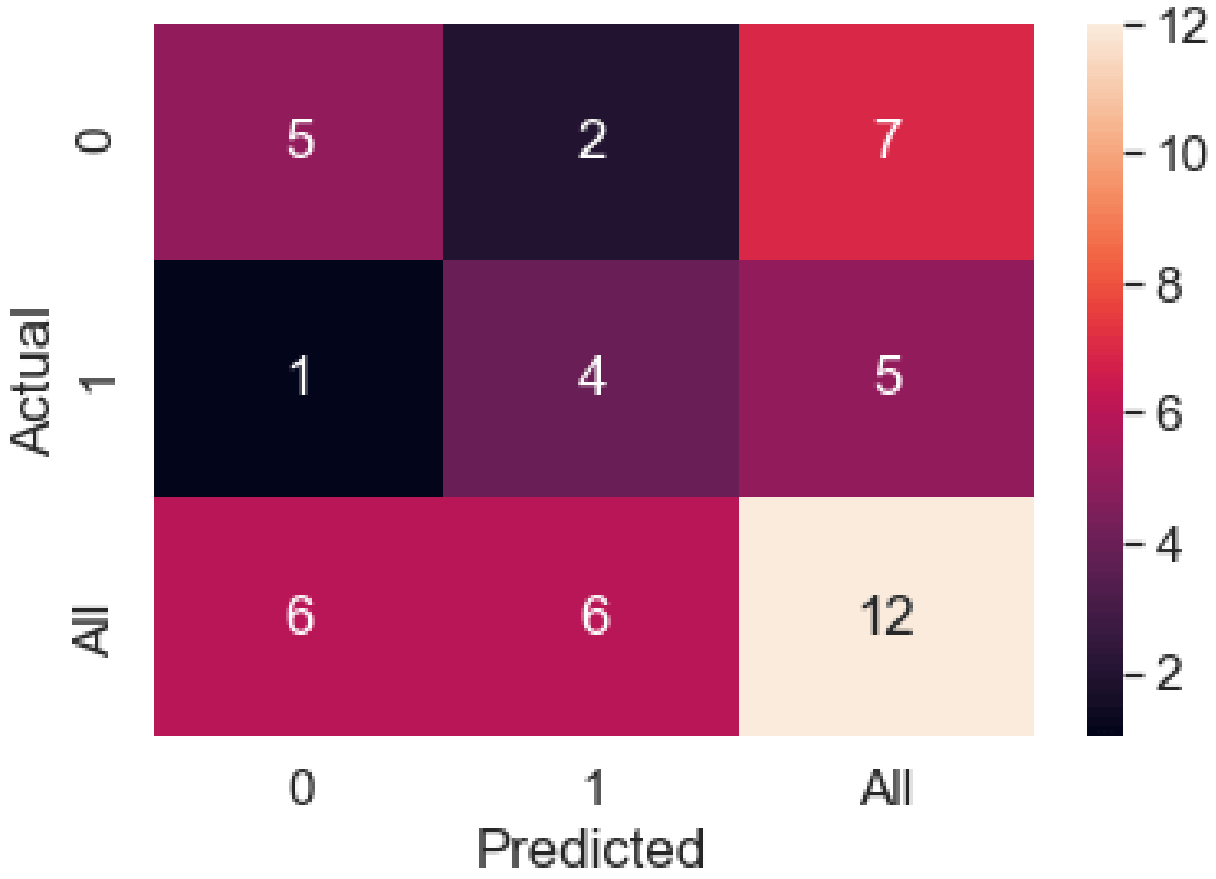
```
import pandas as pd
import seaborn as sn

data = {'y_Predicted': [1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0],
        'y_Actual':    [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0]}

df = pd.DataFrame(data,
   columns=['y_Actual','y_Predicted'])
confusion_matrix = pd.crosstab(df['y_Actual'],
   df['y_Predicted'], rownames=['Actual'],
   colnames=['Predicted'], margins = True)

sn.heatmap(confusion_matrix, annot=True)
```

Resulting Display



Roadmap: Measures for Classification

- Confusion Matrix
- ROC

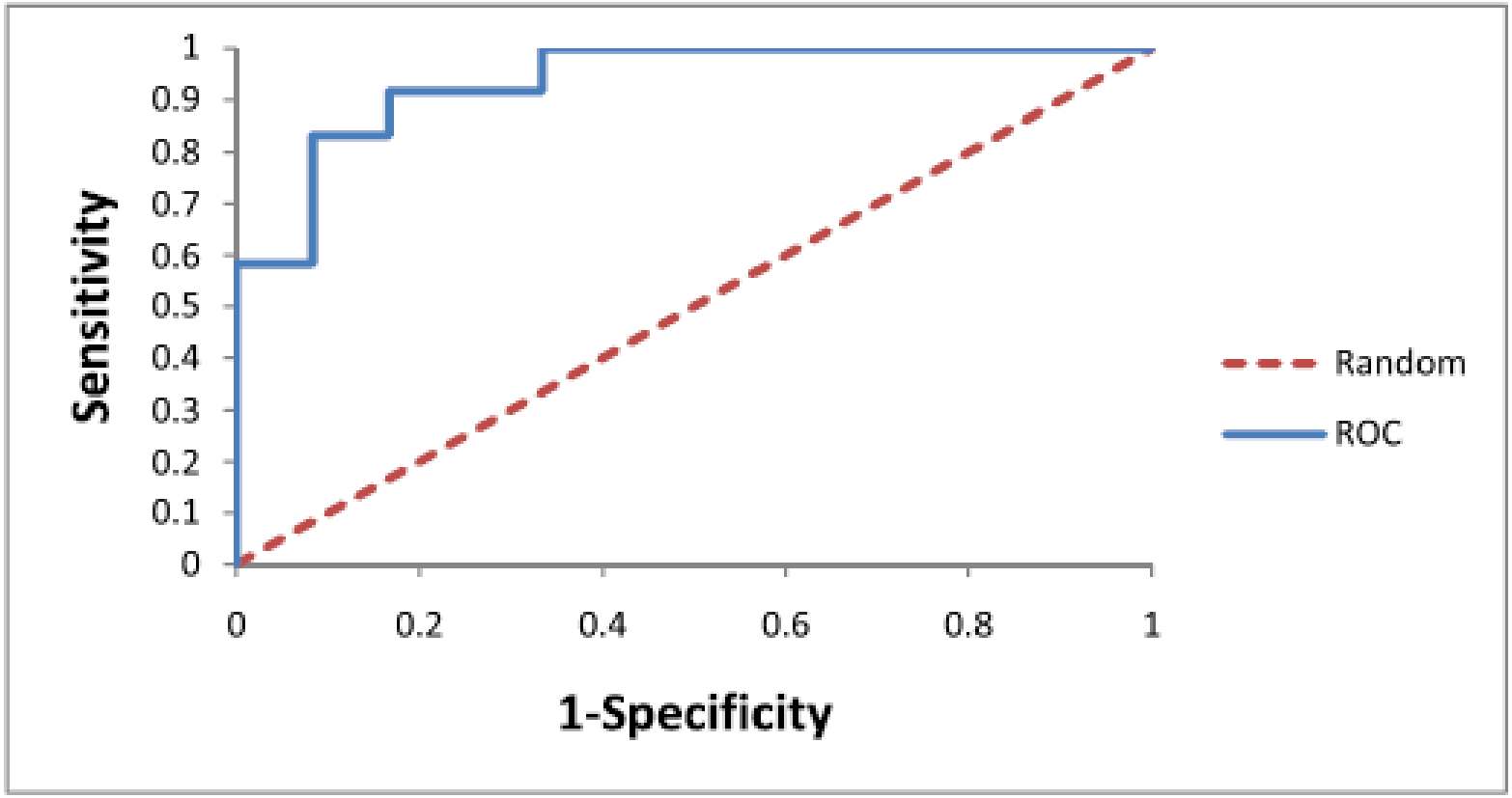
Acknowledgments

- http://www.washburn.edu/faculty/boncella/XLMiner/Lecture%204%20-%20Model%20Evaluation.ppt
- http://people.sju.edu/~ggrevera/cscCV/ComputerVision8.ppt

ROC Curve (1/2)

- ROC = Receiver Operating Characteristic
- Visual comparison of classification models
- The area under the curve (AUC) is a measure of the accuracy of the model.
- Started in electronic signal detection theory (1940s - 1950s)
- Has been used in medicine, radiology, biometrics, and other areas for many decades
- Is increasingly used in evaluating classifiers in machine learning

Example ROC Curve



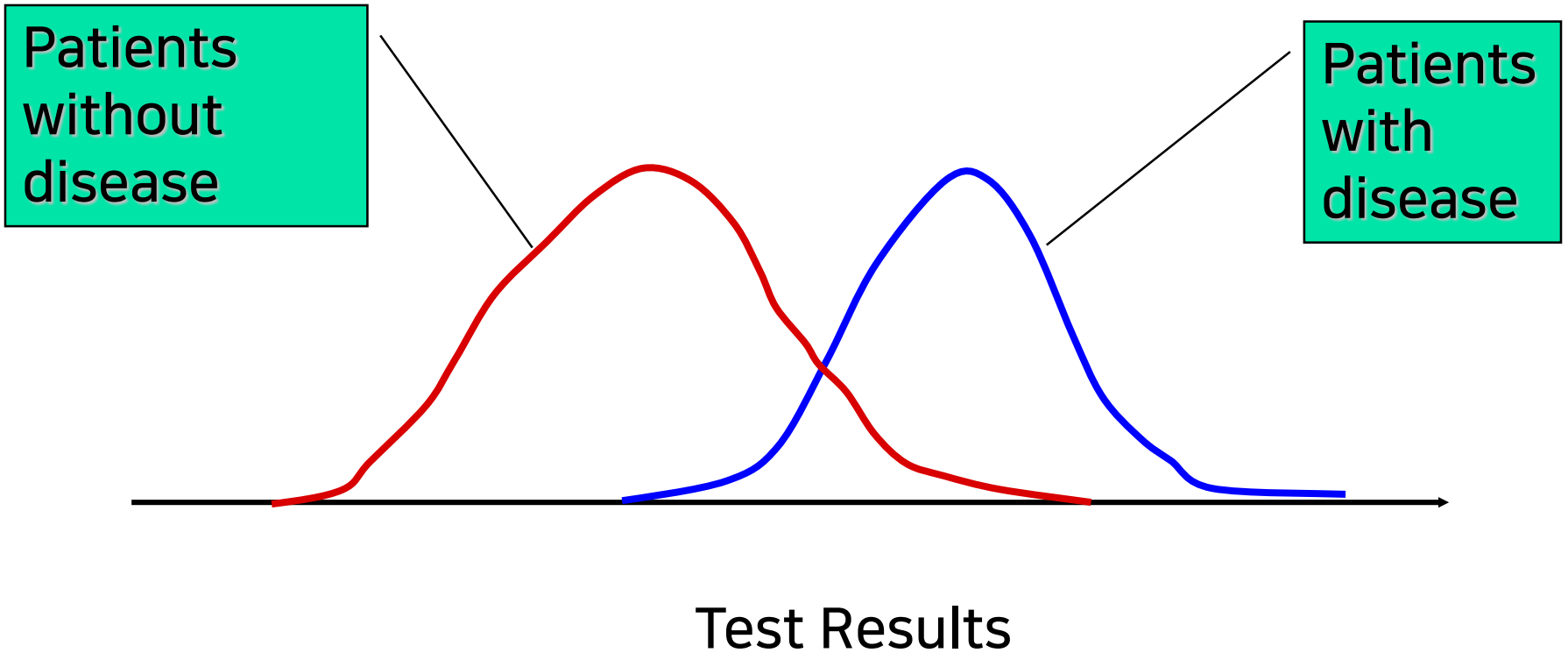
ROC Curve (2/2)

- Created from the confusion matrix
- Shows the tradeoff between the false positive rate (X-axis) and the true positive rate (Y-axis)

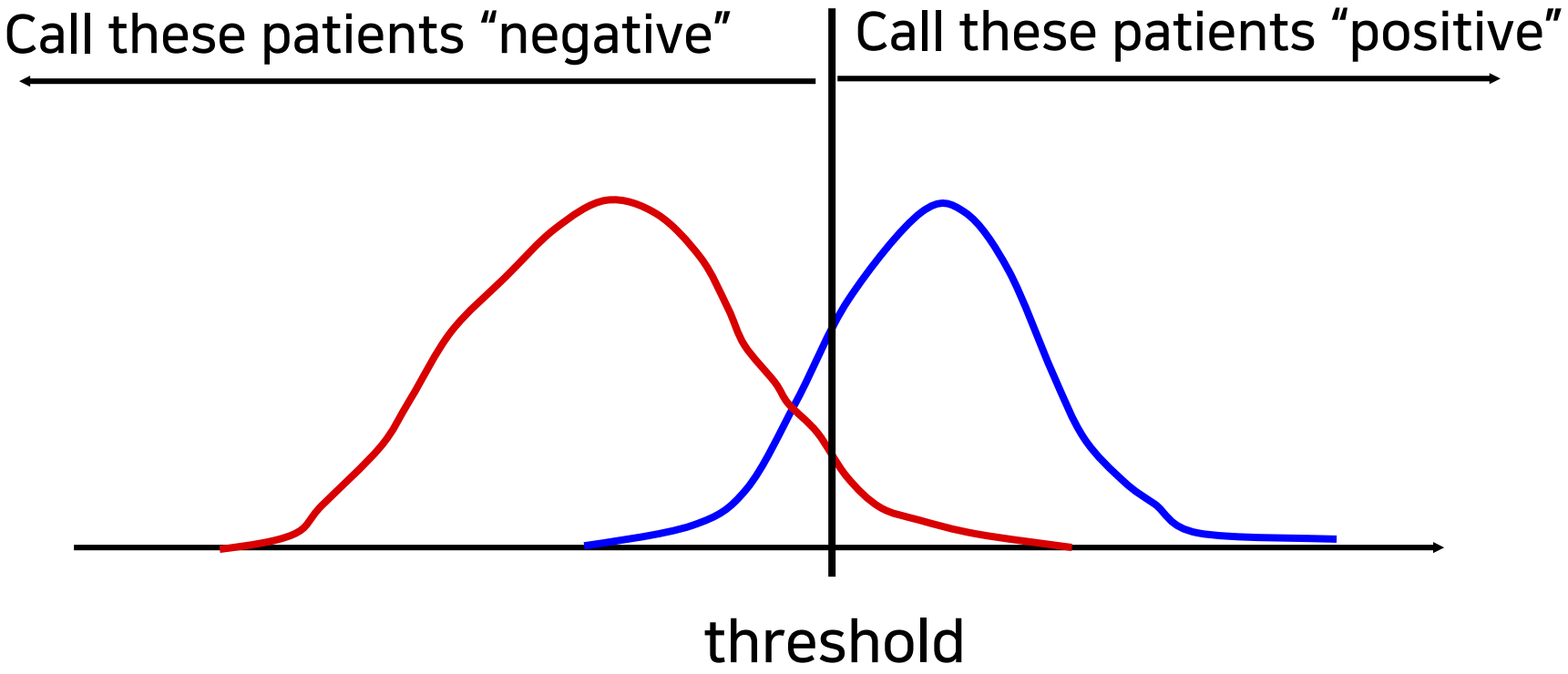
		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the test.

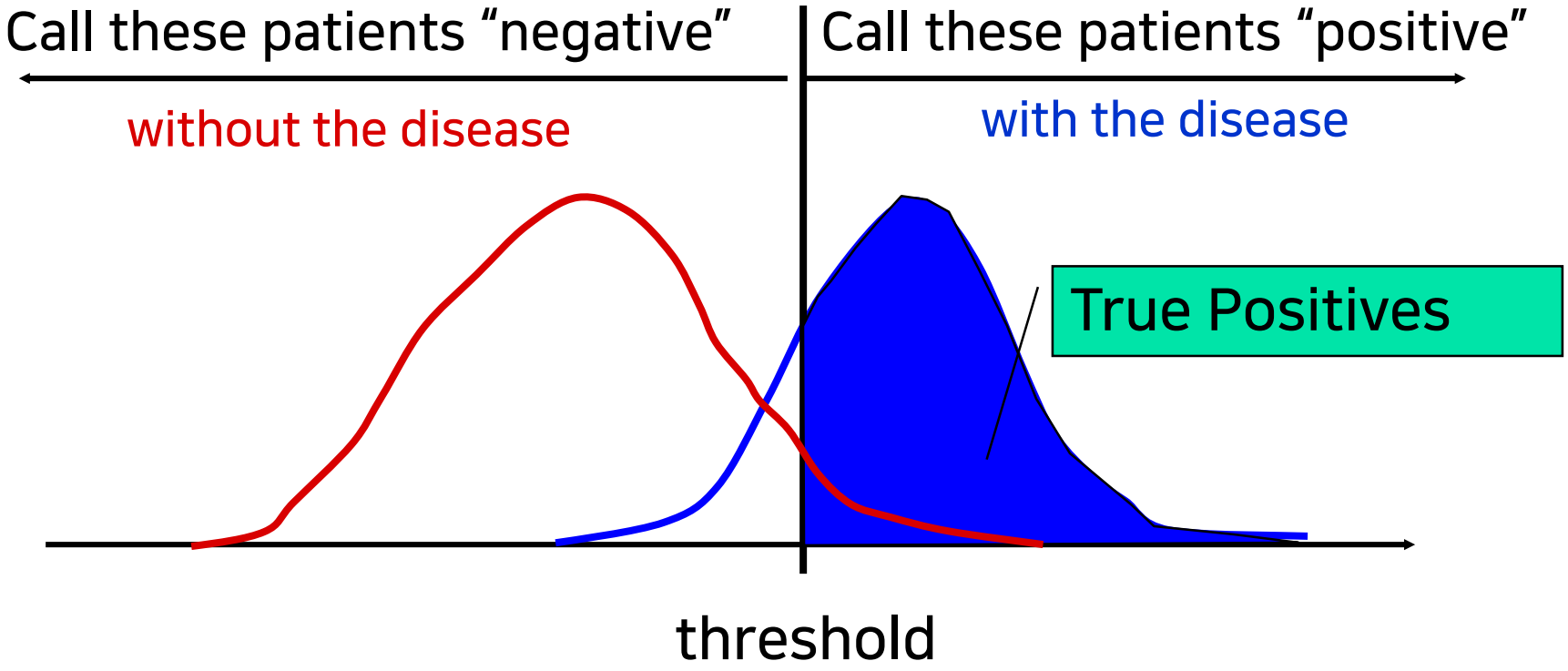
Example: results of classification for patients with a certain disease

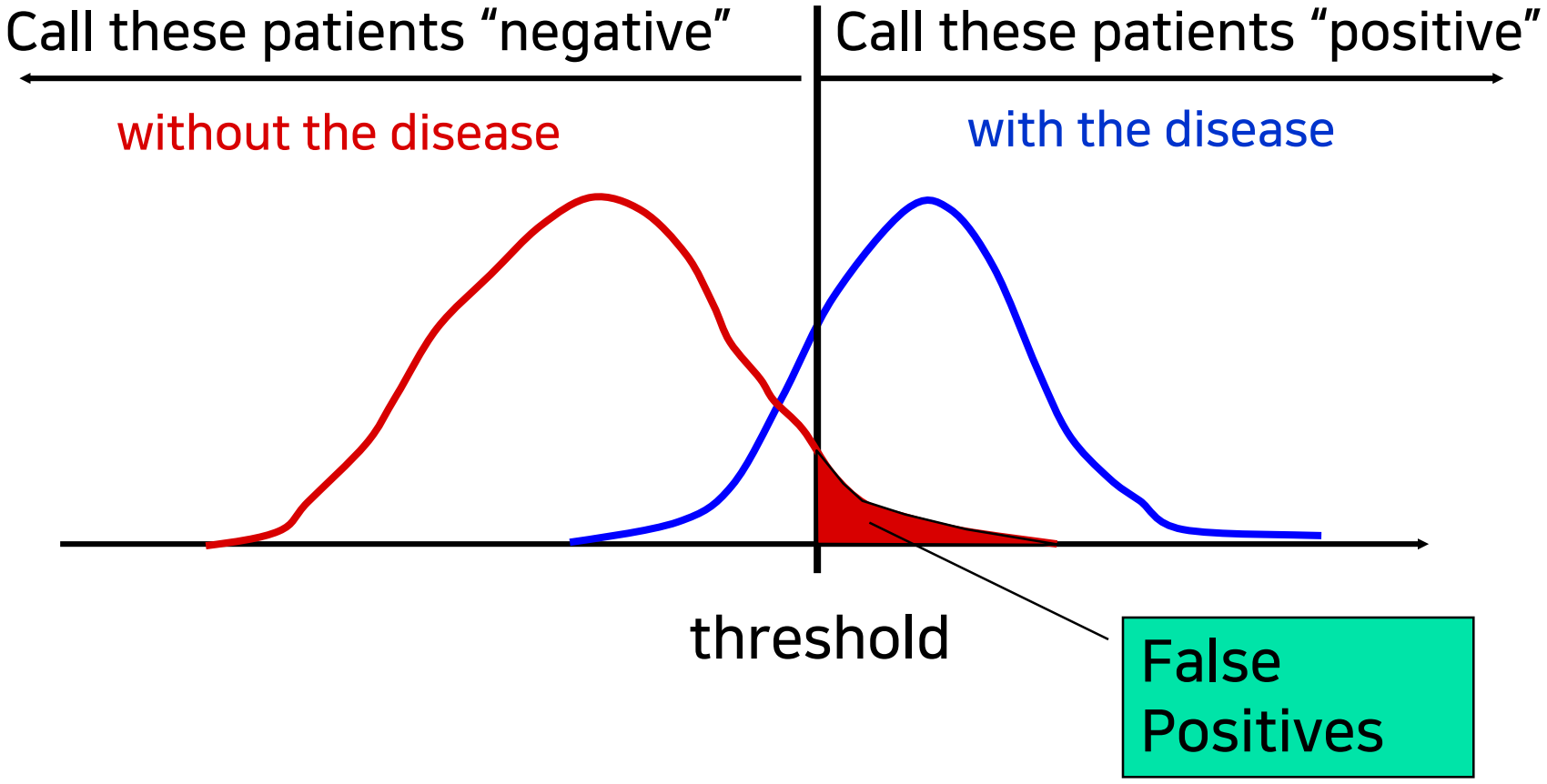


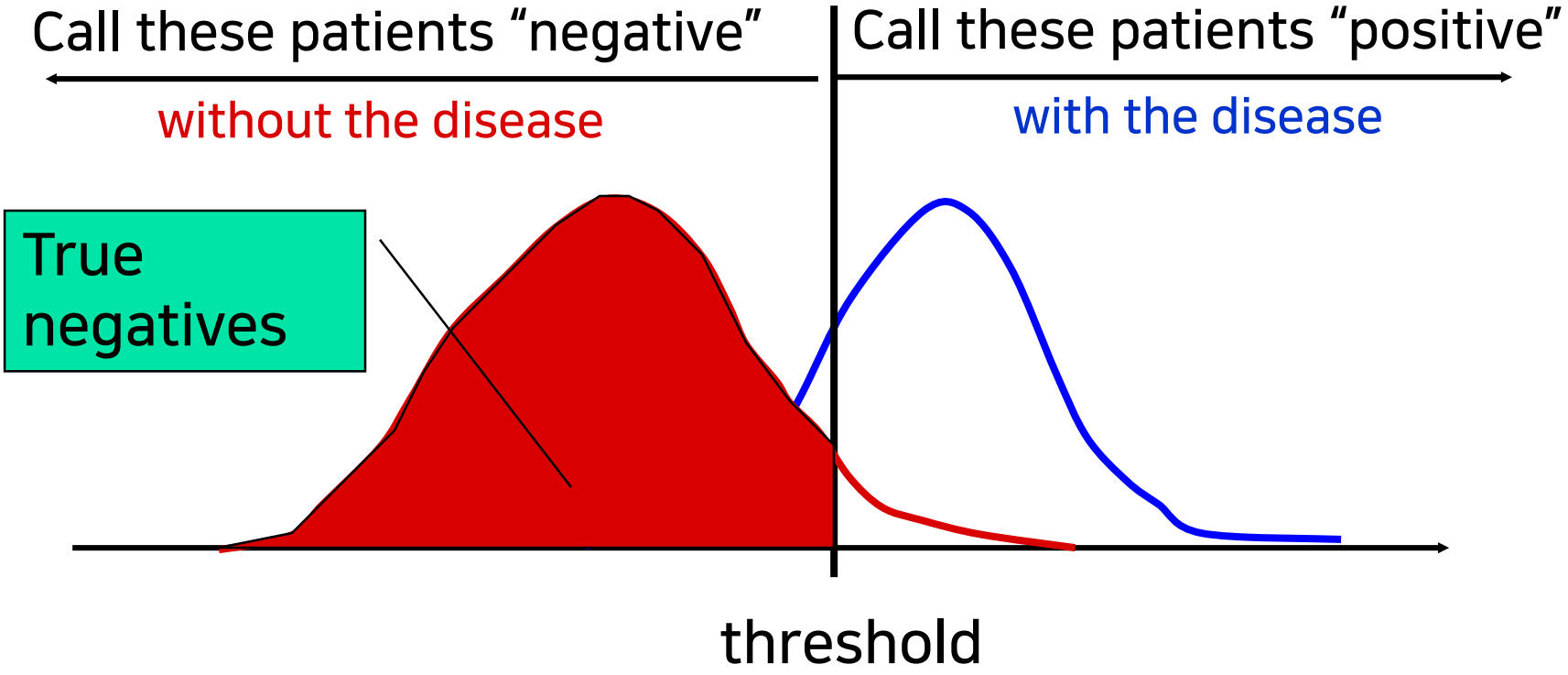
Set a Decision Threshold (Cutoff) for the Confusion Matrix

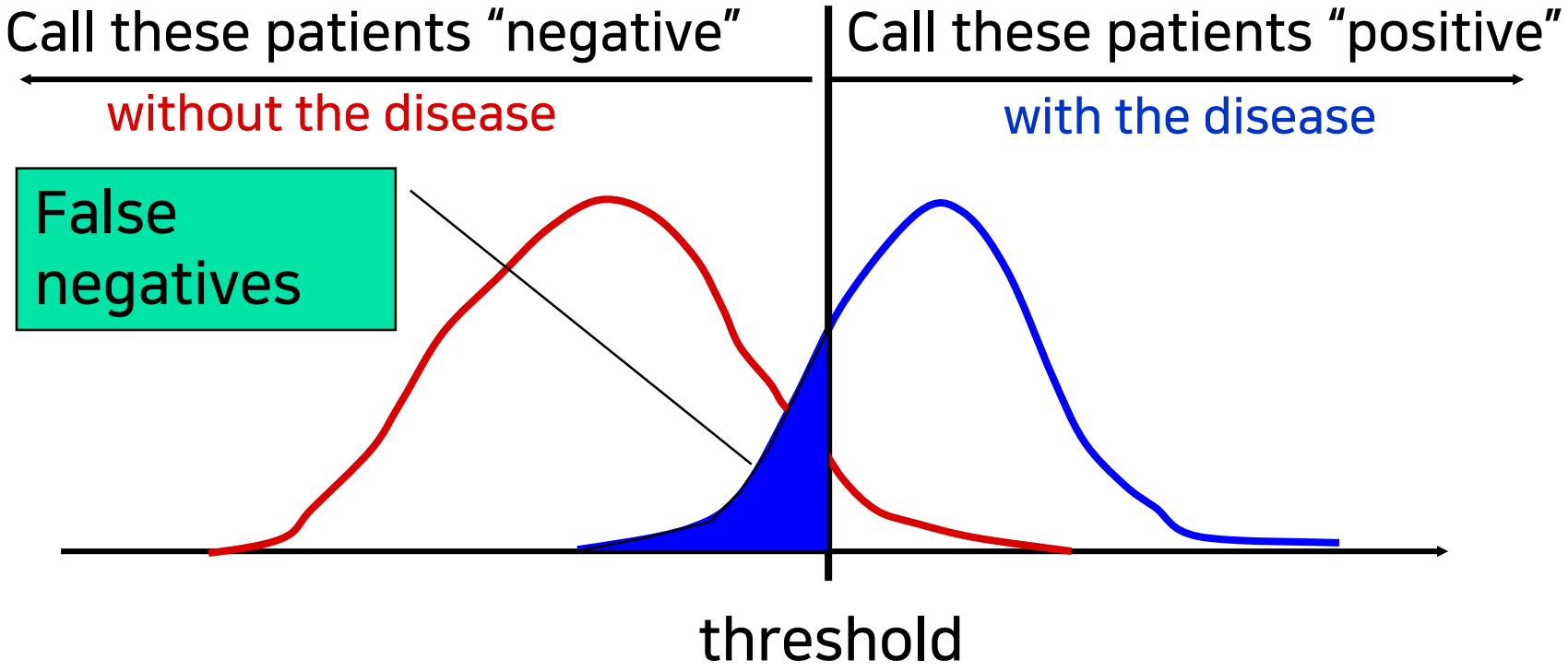


Some definitions …



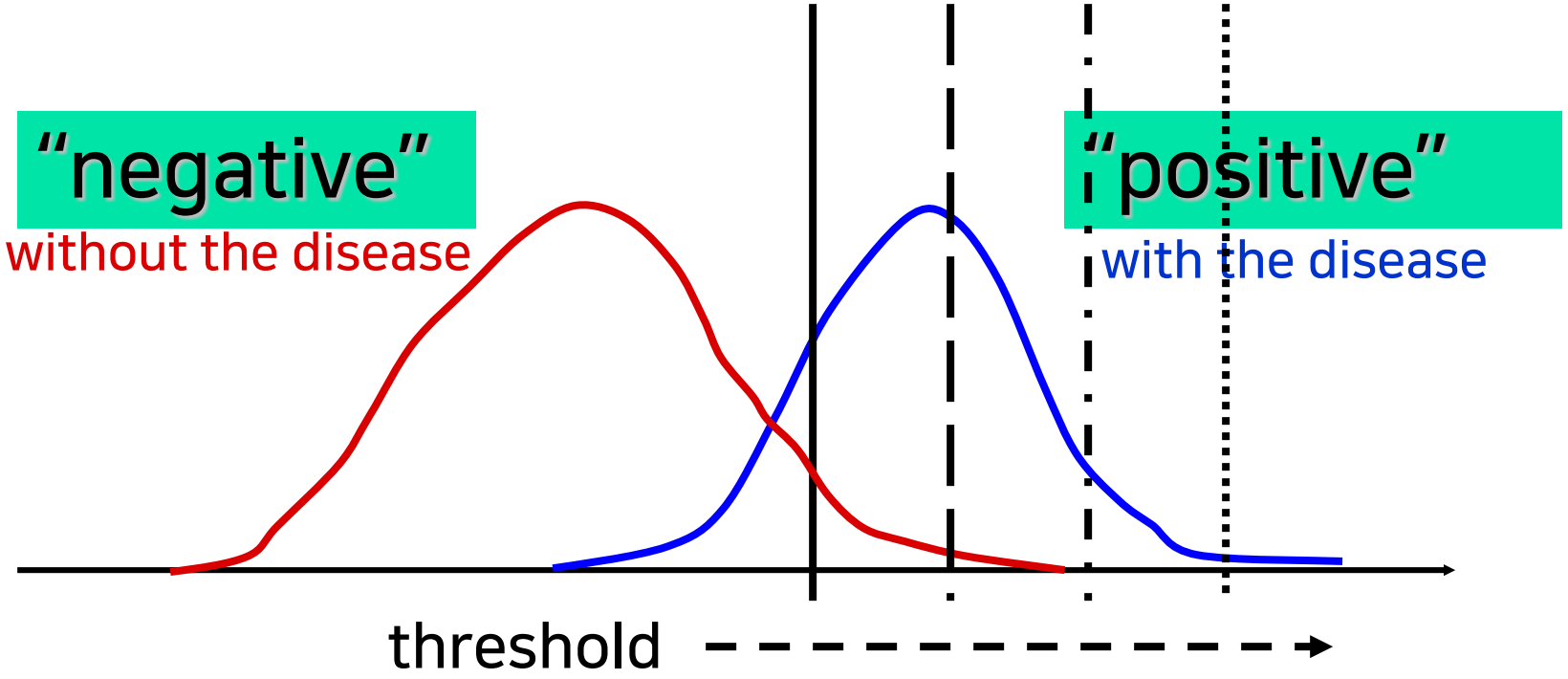

Call these patients “negative”
Call these patients “positive”
without the disease
with the disease
threshold
False Positives


Call these patients “negative”
Call these patients “positive”
without the disease
with the disease
True negatives
threshold


Call these patients “negative”
Call these patients “positive”
without the disease
with the disease
False negatives
threshold

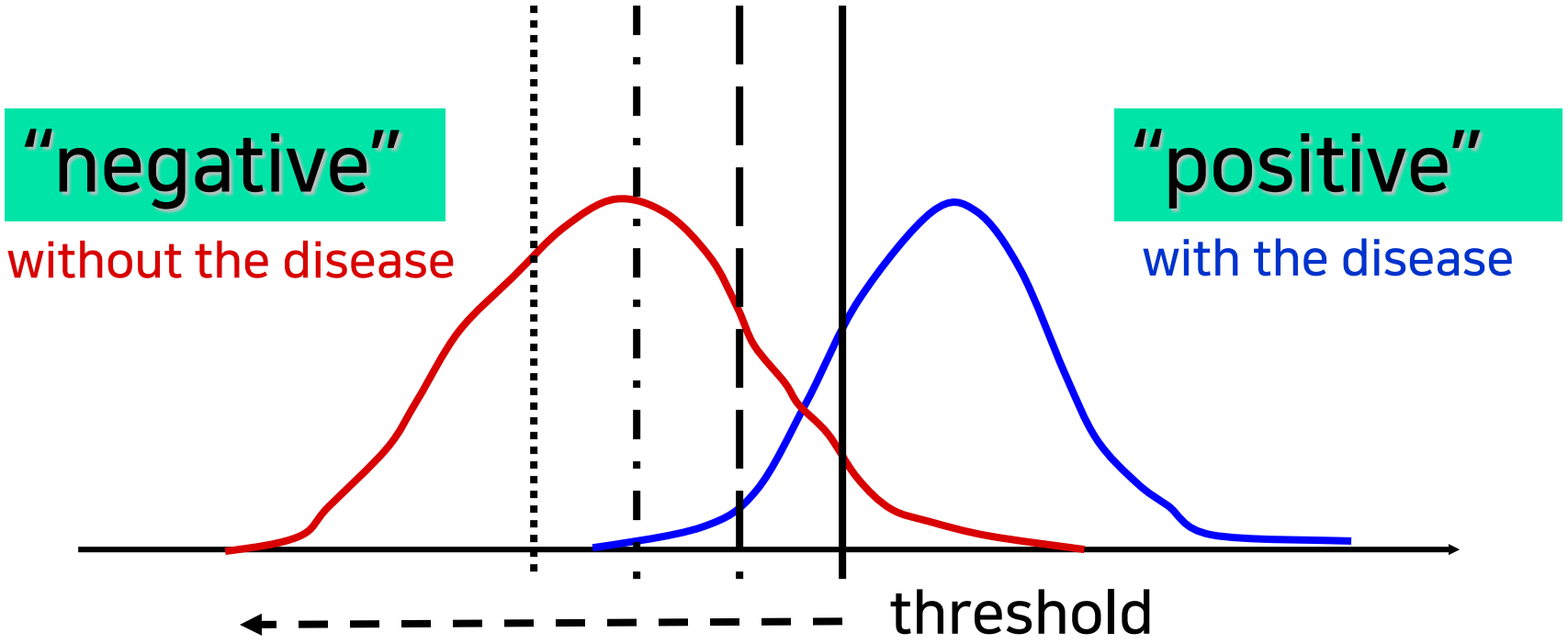
Moving the Threshold: right

Negative (TN, FN) increases

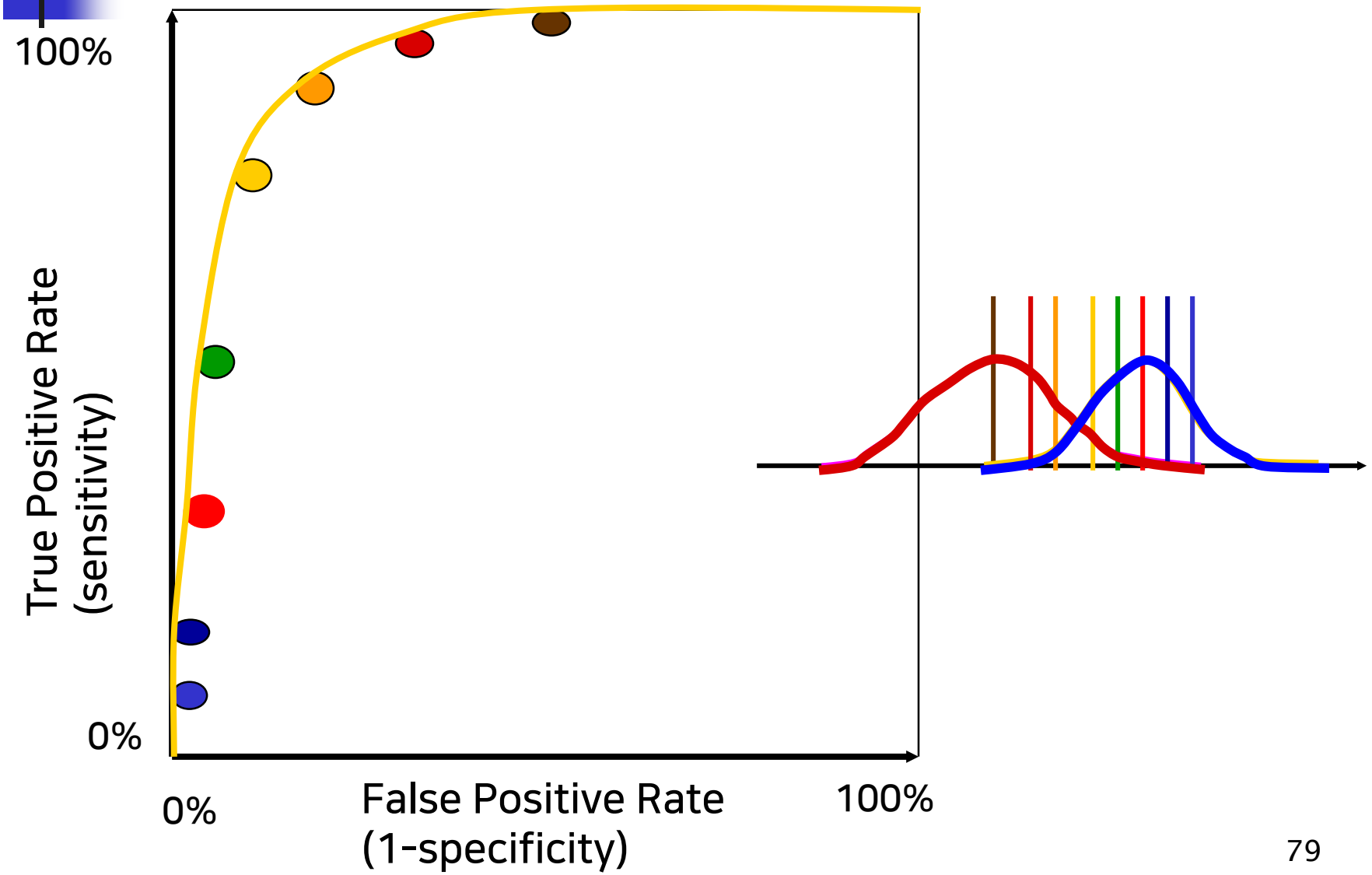


Moving the Threshold: left

Positive (TP, FP) increases



How the Changing Thresholds Correspond to Different Points on the ROC Curve (* Match the colors between the 2 plots *)



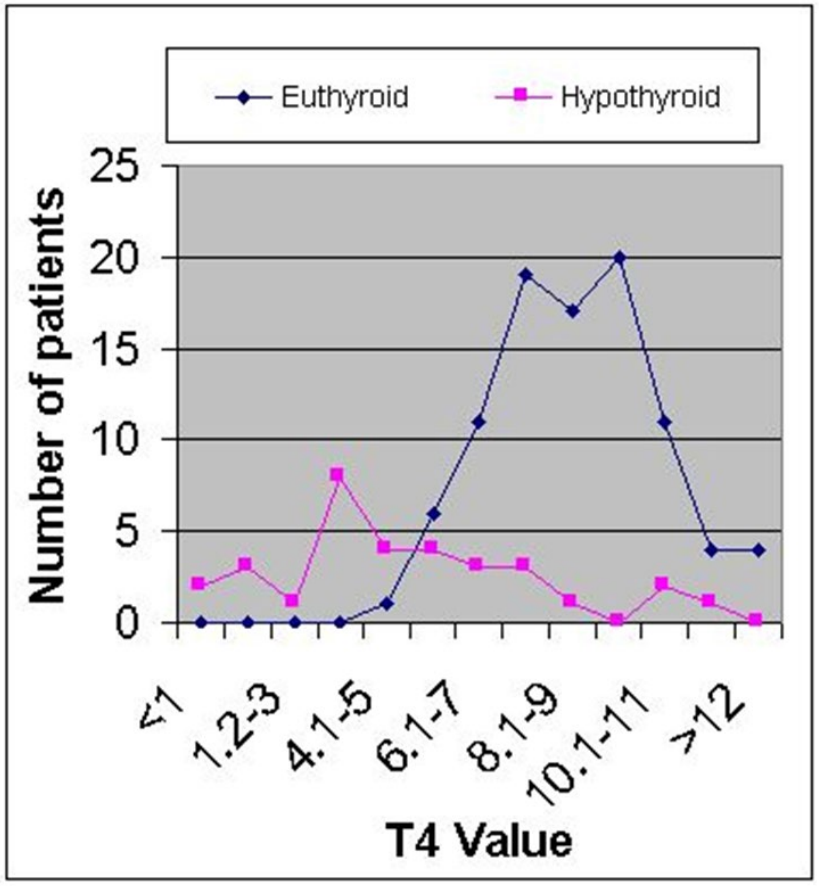
How to Construct an ROC Curve

- Sort the test (prediction) results in the *predicted positive* order.
- Select N thresholds (cutoffs).
- For each threshold, build a classifier with confusion matrix.
- Plot the TP and FP rates of the N classifiers.

Example

- Predicting Hypothyroidism
  - TSH (thyroid stimulating hormone or thyrotropin) levels are the "gold standard."
  - How good are the T4 (thyroxine) levels at predicting hypothyroidism?
    - ** A T4 test measures the amount of the T4 hormone in the blood.

Example Test Results:
positive(hypothyroid) and negative(euthyroid) distributions

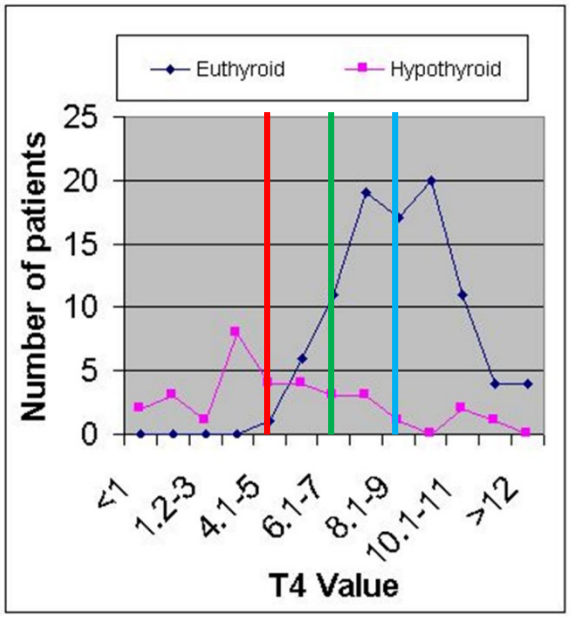


(# of patients) sorted in positive predicted order

T4 value	Hypothyroid	Euthyroid (normal)
<= 5	18	1
5.1 - 7	7	17
7.1 - 9	4	36
> 9	3	39
totals	32	93

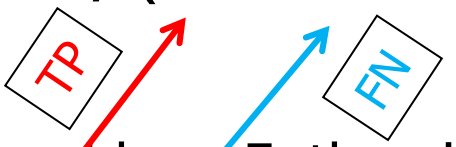
Computing Steps (1/5)

- Select 3 Thresholds
  - T4 <= 5 is hypo; T4 > 5 is normal
  - T4 <= 7 is hypo; T4 > 7 is normal
  - T4 < 9 is hypo; T4 >= 9 is normal
- Calculate the sensitivity (True Positive Rate, recall) and specificity (True Negative Rate) for each threshold.



Computing Steps (2/5)

- For the threshold T4 <= 5 hypo & T4 > 5 normal
- Calculate the sensitivity (TPR)
  - TP / (TP+FN) = 18 / (18+7+4+3) = 18/32 = 0.56



T4 value	Hypothyroid	Euthyroid
<=5	18	1
5.1 - 7	7	17
7.1 - 9	4	36
>= 9	3	39
Totals:	32	93

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

Computing Steps (3/5)

- For the threshold T4 <= 5 hypo & T4 > 5 normal
- Calculate the specificity (TNR)
  - TN / (TN+FP) = (17+36+39) / (17+36+39+1) =

92 / 93 = 0.99

TN + FP

TN

T4 value	Hypothyroid	Euthyroid
<=5	18	1
5.1 - 7	7	17
7.1 - 9	4	36
>=9	3	39
Totals:	32	93

		Predicted class	
		Yes	No
Actual class	Yes	TP: True positive	FN: False negative
	No	FP: False positive	TN: True negative

Computing Steps (4/5)

- Calculate the sensitivity and specificity for the other two thresholds
  - T4 <= 7 is hypo; T4 > 7 is normal.
  - T4 < 9 is hypo; T4 >= 9 is normal.

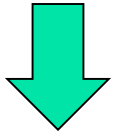
Computing Steps (5/5): Result

T4 value	Hypothyroid	Euthyroid
<= 5	18	1
5.1 - 7	7	17
7.1 - 9	4	36
> 9	3	39
total	32	93

threshold	sensitivity	specificity
5	0.56	0.99
7	0.78	0.81
9	0.91	0.42

Plotting an ROC Curve (1/2)

threshold	sensitivity	specificity
5	0.56	0.99
7	0.78	0.81
9	0.91	0.42

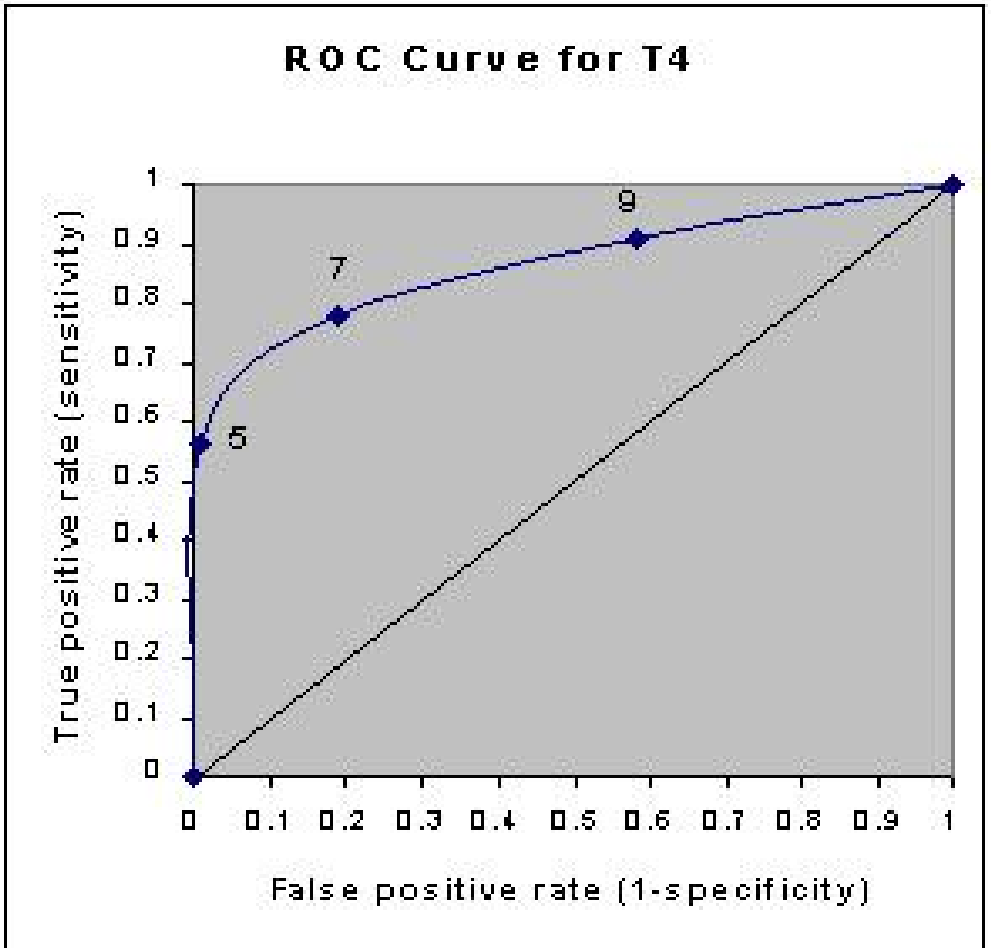


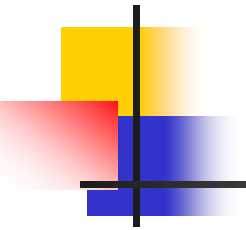
1 - specificity

threshold	True Positive	False Positive
5	0.56	0.01
7	0.78	0.19
9	0.91	0.58

Plotting an ROC Curve (2/2)

threshold	TPR	FPR
5	0.56	0.01
7	0.78	0.19
9	0.91	0.58





Exercise

- Walkthrough the ROC example, and confirm correctness of the computations shown.

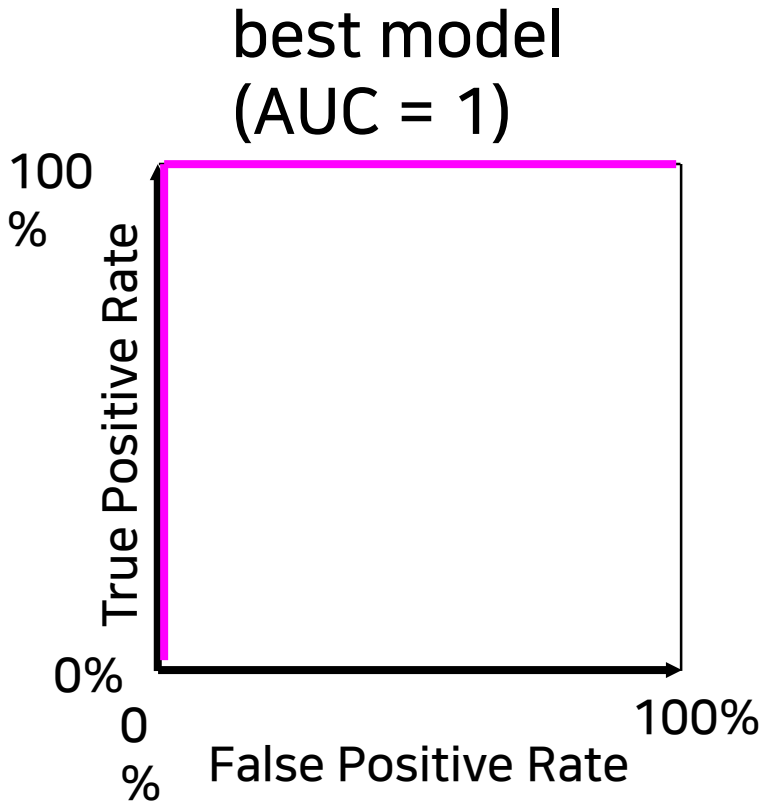
Interpreting an ROC Curve

- ROC curve is a plot of the true positive rate against the false positive rate for the different possible thresholds (cutoff points) of a test.
- An ROC curve demonstrates several things:
  - It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
  - The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
  - The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.
  - The area under the curve is a measure of accuracy.

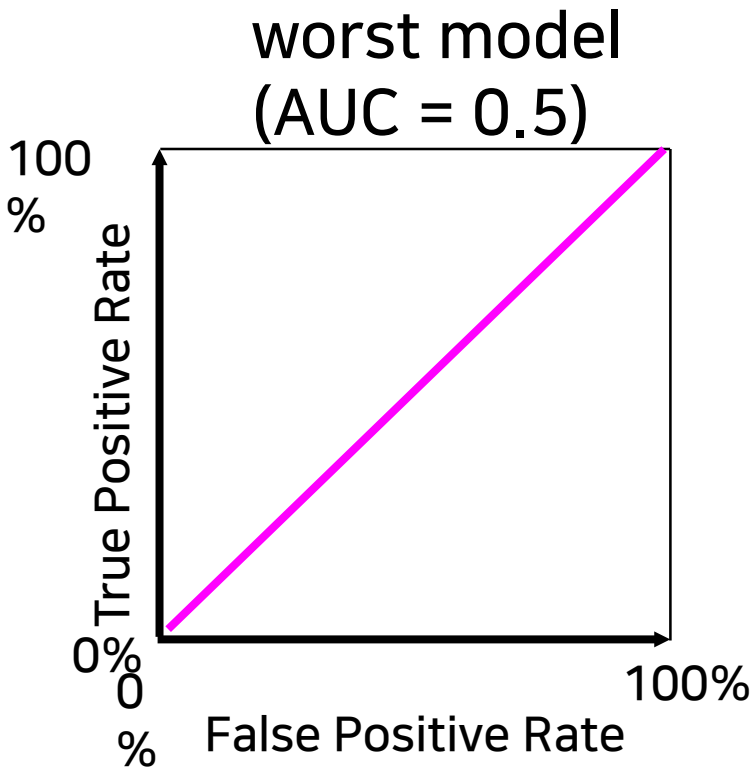
Area Under Curve (AUC)

- AUC = area under the ROC curve
- best case: ACU = 1.0
  - x (0, 1), y (0, 1) → x * y = 1.0
- worst case: ACU = 0.5
  - random guess
- others: between 0.5 and 1.0

ROC Curve Extremes



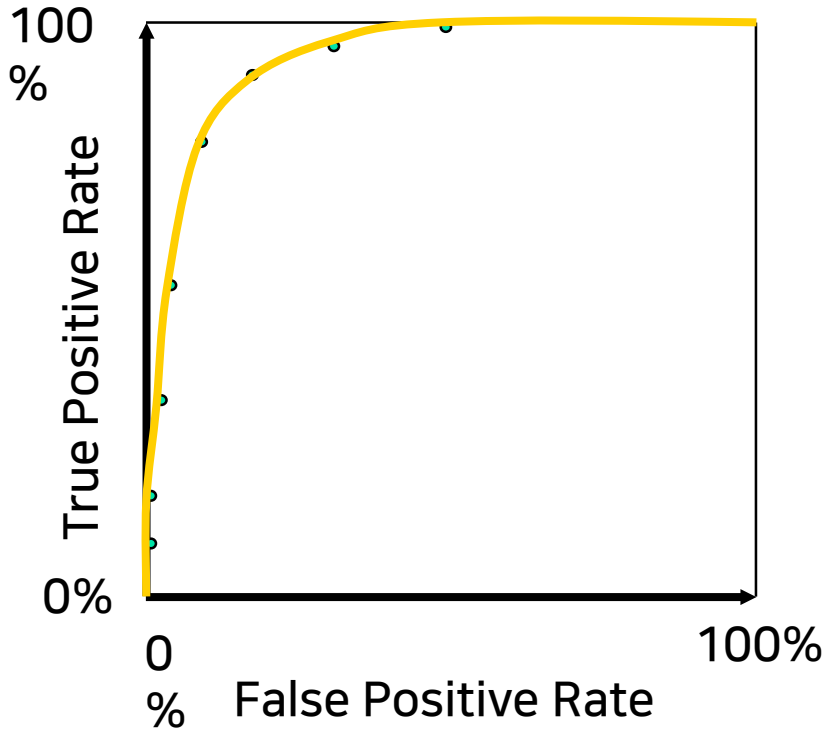
The distributions don't overlap at all



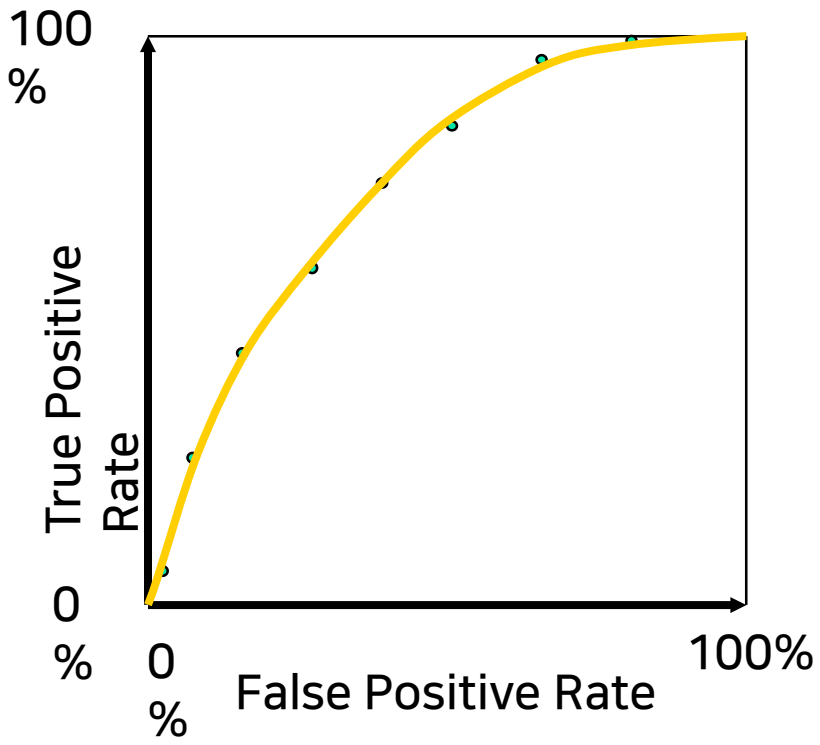
The distributions overlap completely

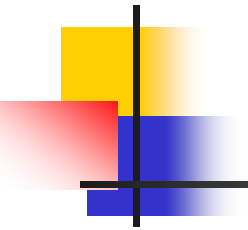
ROC Curve Comparison

good model



poor model





End of SubModule