Advanced Metrics

Ivan Corneillet

Data Scientist

GENERAL ASSEMBLY

Learning Objectives

After this lesson, you should be able to:

- Evaluate a binary classification model using advanced metrics such as confusion matrix, ROC, and AUC curves
- Explain the trade-offs between false positives and false negatives

Here’s what’s happening today:

- Confusion Matrix
- True Positive and False Positive Rates
- ROC and AUC

Accuracy and misclassification rate

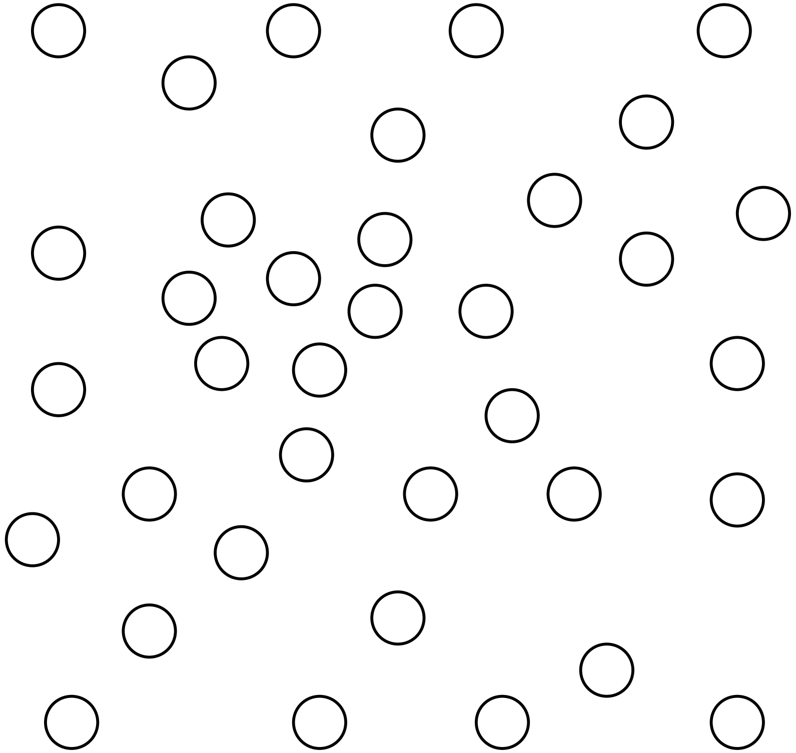
- Accuracy is only one of several metrics used when solving for a classification problem
    - E.g., if we know a prediction is 75% accurate, accuracy doesn't provide any insight into why the 25% was wrong. Was it wrong *equally* across all class labels? Did it just guess one class label for all predictions and 25% of the data was just the other label?
- It's important to look at other metrics to fully understand the problem

- Accuracy
    - How many observations that we predicted were correct? This is a value we'd want to increase (like R^2)

- Misclassification rate
    - Directly opposite of accuracy
    - Of all the observations we predicted, how many were incorrect? This is a value we'd want to decrease (like the mean squared error)

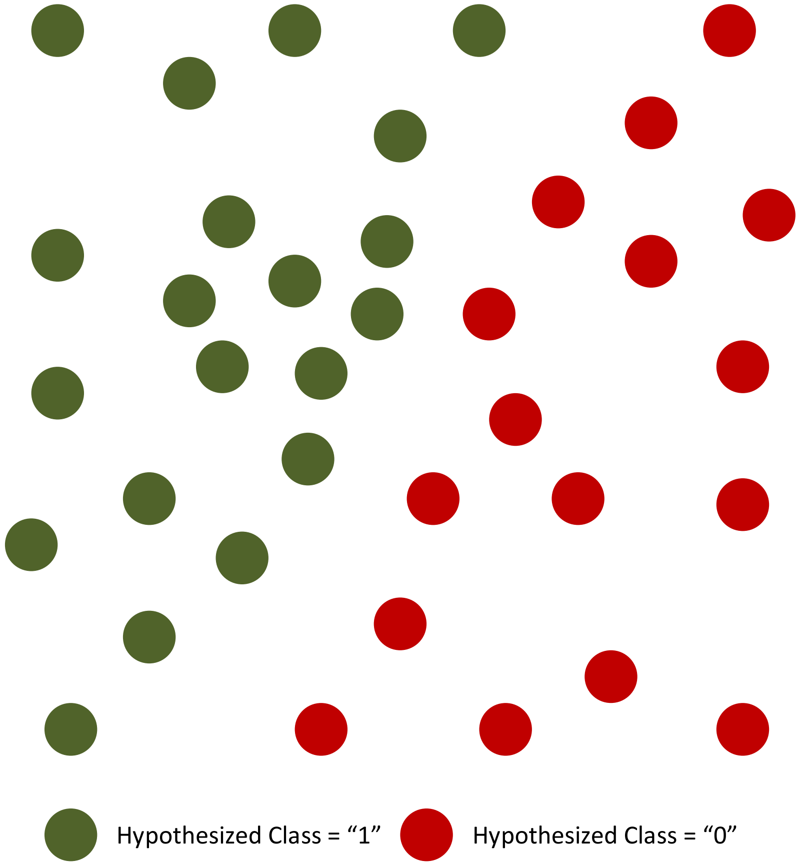
Confusion Matrix

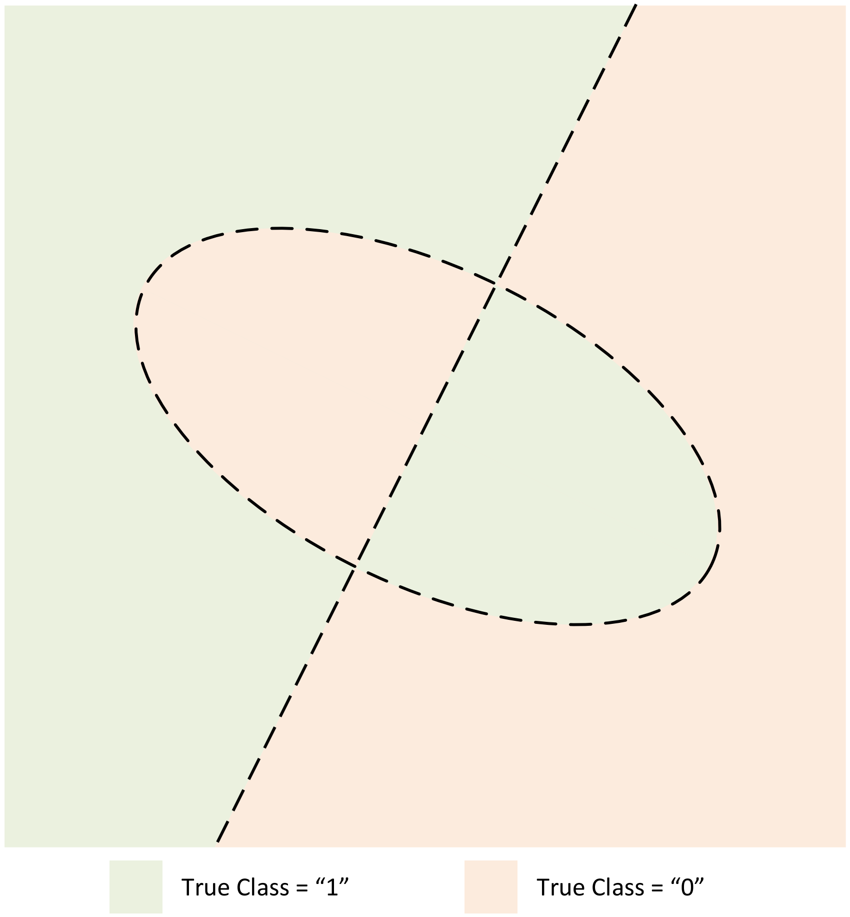
Stepping back | Let’s say we want to classify this data:



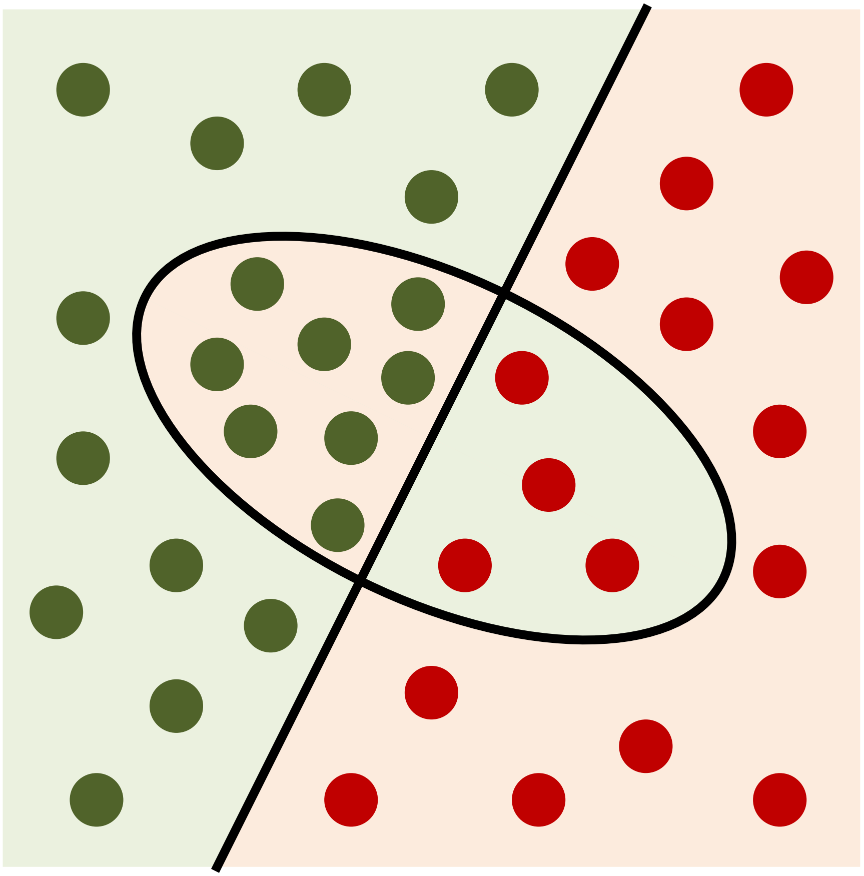
○ Data to Predict

Hypothesized and true classes don’t necessarily match





We can rearrange these 4 possibilities into a 2x2 table



🟢 Hypothesized Class = “1”

🔴 Hypothesized Class = “0”

🟩 True Class = “1”

🟧 True Class = “0”

True Positive	False Positive
🟢	🟢
🔴	🔴
False Negative	True Negative

Confusion Matrix (a.k.a., Contingency Table or Error Matrix)

		True Class	
		1	0
Hypothesized Class	1 ●	● True Positives (TP)	● False Positives (FP) *(type I error)*
	0 ■	■ False Negatives (FN) *(type II error)*	■ True Negatives (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

- A confusion matrix is a specific table layout that allows visualization of the performance of a supervised learning algorithm
- Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class
- The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e., commonly mislabeling one as another)

Activity | Interpreting the confusion matrix

EXERCISE

DIRECTIONS (10 minutes)

1. Use the variables defined in the confusion matrix (TP, FN, FP, TN, P, and N) to calculate the answers to the following questions:
    a. Overall, how often is the classifier correct?
    b. When the classifier predicts yes, how often is it correct?
    c. How often does the yes condition actually occur in our sample?
    d. When it’s actually yes, how often does the classifier predict yes?
    e. When it’s actually no, how often does the classifier predict yes?
    f. When it's actually no, how often does it predict no?
    g. Overall, how often is the classifier wrong?

Activity | Interpreting the confusion matrix (cont.)

EXERCISE

Hypothesized Class		True Class 1	True Class 0
1	●	● True Positives (TP)	● False Positives (FP) *(type I error)*
0	■	■ False Negatives (FN) *(type II error)*	■ True Negatives (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

Question: Overall, how often is the classifier correct? Answer: $\frac{TP+TN}{P+N}$ $(accuracy)$	*When the classifier predicts yes, how often is it correct?* Answer: $\frac{TP}{TP+FP}$ *(precision)*
How often does the yes condition actually occur in our sample? Answer: $\frac{P}{P+N}$ *(prevalence)*	*When it's actually yes, how often does the classifier predict yes?* Answer: $\frac{TP}{P}$ *(TPR, sensitivity, recall)*
When it's actually no, how often does the classifier predict yes? Answer: $\frac{FP}{N}$ *(FPR, fall-out)*	*When it's actually no, how often does it predict no?* Answer: $\frac{TN}{N}$ *(specificity)*
Overall, how often is the classifier wrong? Answer: $\frac{FP+FN}{P+N}$ *(misclassification rate)*	

Activity | Interpreting the confusion matrix (cont.)

EXERCISE

DIRECTIONS (cont.)

2. Given a medical exam that tests for cancer ($1 = Cancer$, $0 = Cancer\ free$), use the variables defined in the confusion matrix (TP, FN, FP, TN, P, and N) to calculate the answers to the following questions:
   a. How often is it correct when it identify patients with cancer?
   b. How often does it correctly identify patients without cancer?
   c. How often does it trigger a “false alarm” by saying a patient has cancer when they actually don’t?
   d. How often does it correctly identify patients with cancer?
3. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions

Activity | Interpreting the confusion matrix (cont.)

EXERCISE

		True Class	
		Has Cancer	Doesn't have cancer
Hypothesized Class	Predict Cancer ●	● True Positives (TP)	● False Positives (FP) *(type I error)*
	Predict No Cancer ■	■ False Negatives (FN) *(type II error)*	■ True Negatives (TN)
	Total Columns	$P = TP + FN$	$N = FP + TN$

How often is it correct when it identify patients with cancer? Answer: $\frac{TP}{TP+FP}$ *(precision)*	*How often does it correctly identify patients without cancer?* Answer: $\frac{TN}{N}$ *(specificity)*
How often does it trigger a "false alarm" by saying a patient has cancer when they actually don't? Answer: $\frac{FP}{N}$ *(FPR, fall-out)*	*How often does it correctly identify patients with cancer?* Answer: $\frac{TP}{P}$ *(TPR, sensitivity, recall)*
How many patients have cancer? Answer: $\frac{P}{P+N}$ *(prevalence)*	

Activity | Interpreting the confusion matrix – Take 2

EXERCISE

DIRECTIONS (5 minutes)

1. We trained a binary classifier and got the following hypothesized probabilities ($\hat{p}$) for the samples in the table.
   a. What are the hypothesized classes ($\hat{c}$)?
   b. Are the samples true/false positive/negative?
2. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions

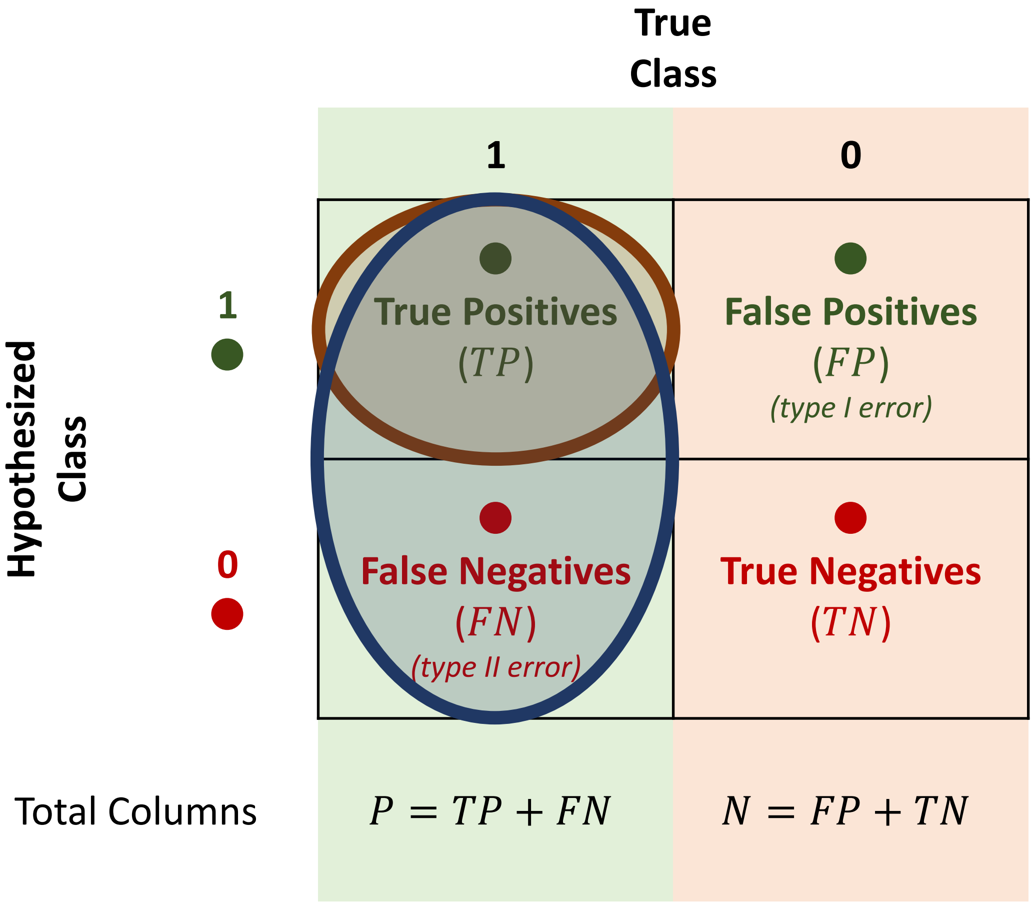
Activity | Interpreting the confusion matrix – Take 2 (cont.)

EXERCISE

#	$\hat{p} = P(c = 1)$	$\hat{c}$	c	*True/False Positive/Negative*
1	.44	**0**	**1**	**FN**
2	.29	**0**	**0**	**TN**
3	.98	**1**	**1**	**TP**
4	.69	**1**	**0**	**FP**
5	.07	**0**	**1**	**FN**

True and False Positive Rates

True Positive Rate, $TPR = \frac{TP}{P}$

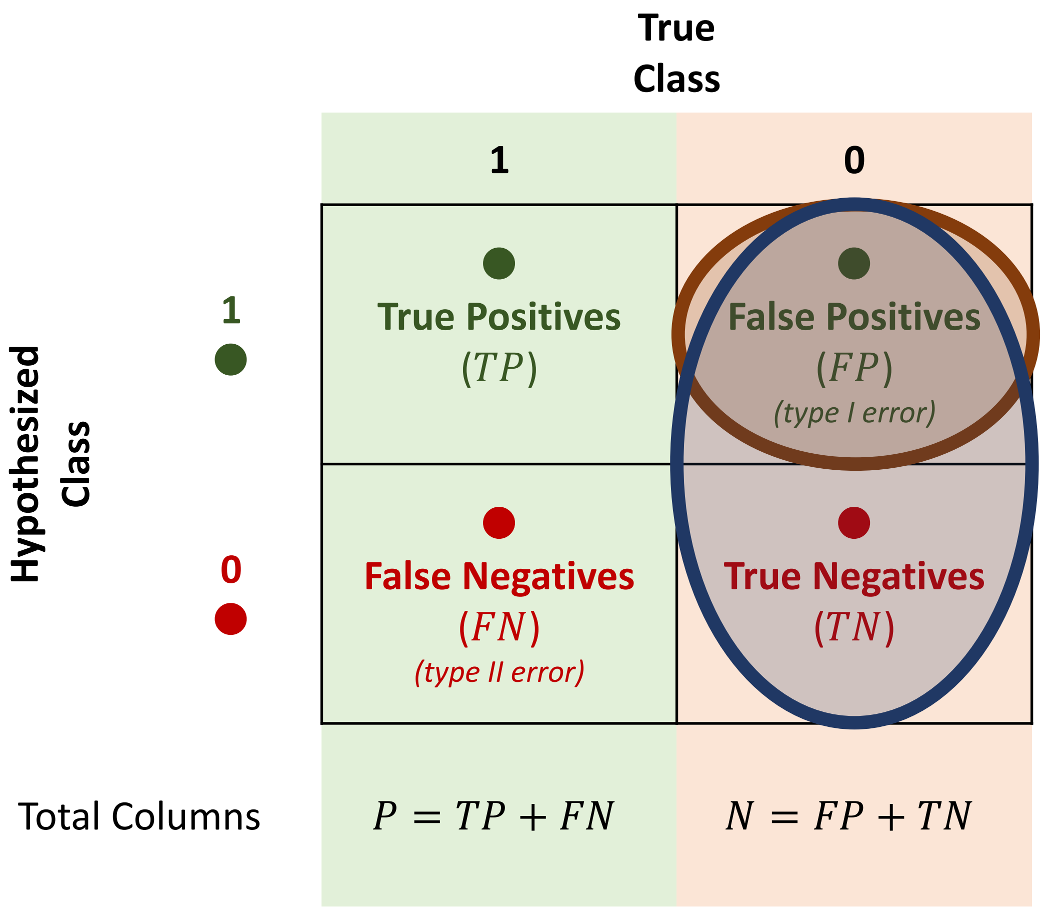


		True Class	
		1	0
Hypothesized Class	1 ●	● True Positives (TP)	● False Positives (FP) *(type I error)*
	0 ●	● False Negatives (FN) *(type II error)*	● True Negatives (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

- When it’s actually yes, how often does the classifier predict yes?
- A.k.a., “Sensitivity”

  - E.g., given a medical exam that tests for cancer, how often does it correctly identify patients with cancer?
  - Likewise, this can be inverted: how often does a test *correctly* identify patients without cancer

False Positive Rate, $FPR = \frac{FP}{N}$



		True Class	
		1	0
Hypothesized Class	1 ●	● True Positives (TP)	● False Positives (FP) *(type I error)*
	0 ●	● False Negatives (FN) *(type II error)*	● True Negatives (TN)
Total Columns		$P = TP + FN$	$N = FP + TN$

- When it’s actually no, how often does the classifier predict yes?
- A.k.a., “Fall-out”

- E.g., given a medical exam that tests for cancer, how often does it trigger a “false alarm” by saying a patient has cancer when they actually don’t?
- Likewise, this can be also inverted: how often does a test *incorrectly* identify patients as being cancer-free when they might actually have cancer!

True positive and false positive rates

- We can split up the accuracy of each label by using true positive and false positive rates. Using them, we can get a much clearer picture of where predictions begin to fall apart

- A good classifier would have a true positive rate approaching 1, and a false positive rate approaching 0. In a binary problem (say, predicting if someone smokes or not), it would accurately predict all of the smokers as smokers, and not accidentally predict any of the non-smokers as smokers

ROC and AUC

ROC (receiver operating characteristic or relative operating characteristic) and AUC (Area Under the Curve)

Activity | Introduction to the ROC space

EXERCISE

DIRECTIONS (5 minutes)

1. Calculate TPR and FPR for the four confusion matrices in the handout and place them in the ROC space (TPR as a function of FPR)
2. How would you classify these four cases as a function of their performance (e.g., better or worse)
3. What does the ROC space tells you?
4. When finished, share your answers with your table

DELIVERABLE

Answers to the above questions

Activity | Introduction to the ROC space (cont.)

EXERCISE

A

TP = 63	FP = 28
FN = 37	TN = 72

B

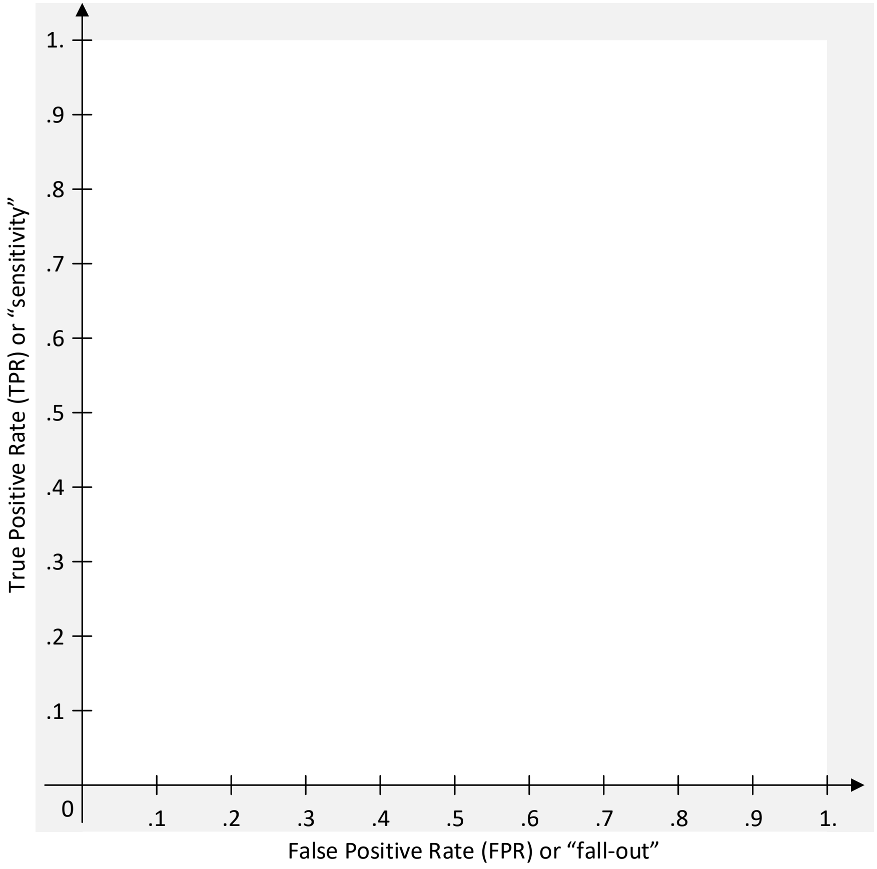
TP = 77	FP = 77
FN = 23	TN = 23

C

TP = 24	FP = 88
FN = 76	TN = 12

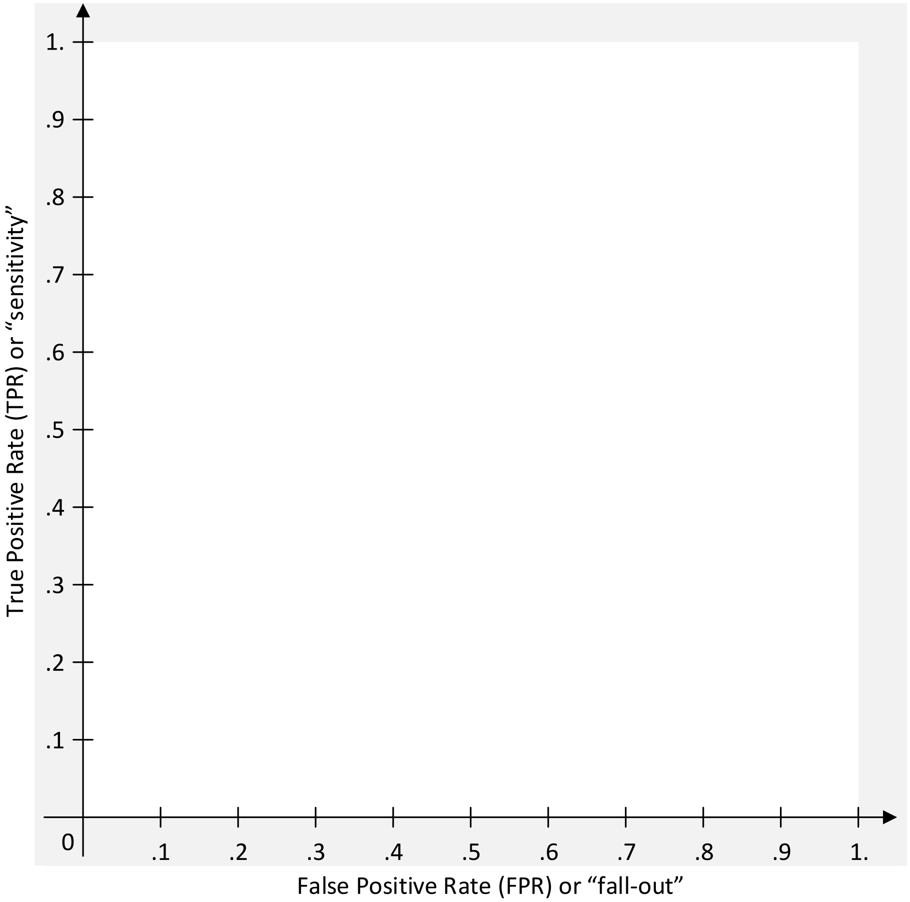
D

TP = 76	FP = 12
FN = 24	TN = 88



ROC (receiver operating characteristic) curve (a.k.a., relative operating characteristic curve)

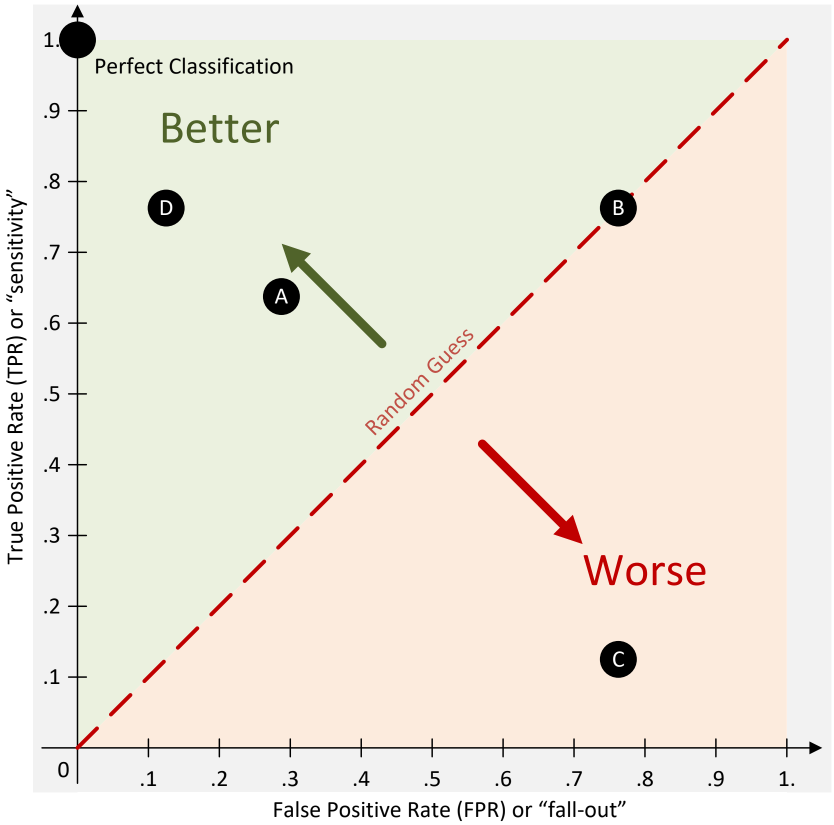
- An ROC curve plots the true positive rate (TPR) (or "sensitivity") against the false positive rate (FPR) (or "fall-out") at various threshold settings to illustrate the performance of a binary classifier system. The ROC curve is thus the sensitivity as a function of fall-out



ROC curves demonstrate several things:

- It shows the trade-off between sensitivity and fall-out (any increase in sensitivity will be accompanied by an increase in fallout)
  - The closer the **points** are in the left-hand border and then the top border of the ROC space, the more accurate the classifier is
  - The closer the **points** come to the 45-degree diagonal of the ROC space, the less accurate the classifier is

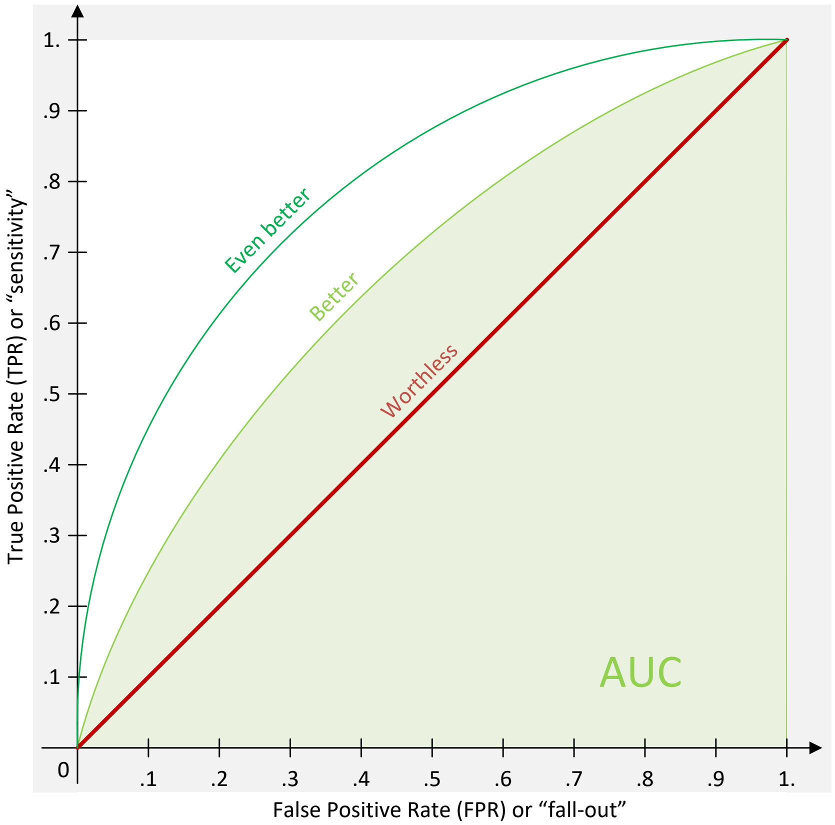
ROC Space



ROC curves demonstrate several things: (cont.)

- The area under the curve (AUC) is a measure of classifier accuracy
  - The closer the **curve** follows the left-hand border and then the top border of the ROC space, the more accurate the classifier is
  - The closer the **curve** comes to the 45-degree diagonal of the ROC space, the less accurate the classifier is

ROCs and AUCs



Plotting an ROC curve

- ❶ Discard $\hat{c}$ (hypothesized class) and whether it is a true/false positive/negative
- ❷ Order the trained sample by their decreasing hypothesized probabilities $\hat{p}$ (from more confident to have a ‘1’ down to less confident to have a ‘1’)
- ❸ Discard the original ranking from the dataset as well as $\hat{p}$
- ❹ Start at $(0, 0)$
- ❺ For each training sample in the sorted order
  - If $c = 1$, move up by $^1/_P$
  - If $c = 0$, move up by $^1/_N$
- ❻ If not already at $(1, 1)$, go all the way to the right, then up all the way to $(1, 1)$

Let's plot the ROC for the following trained binary classifier

EXAMPLE

#	$\hat{p}$	$\hat{c}$	c	*True/False Positive/Negative*
1	.44	0	1	FN
2	.29	0	0	TN
3	.98	1	1	TP
4	.69	1	0	FP
5	.07	0	1	FN

➊ Discard $\hat{c}$ (hypothesized class) and whether it is a true/false positive/negative

EXAMPLE

#	$\hat{p}$	c
1	.44	1
2	.29	0
3	.98	1
4	.69	0
5	.07	1

❷ Order the trained sample by their decreasing hypothesized probabilities $\hat{p}$ (from more confident to have a ‘1’ down to less confident to have a ‘1’)

EXAMPLE

#’ (ranking by decreasing probabilities)	# (ranking from dataset)	$\hat{p}$	c
1	3	.98	1
2	4	.69	0
3	1	.44	1
4	2	.29	0
5	5	.07	1

❸ Discard the original ranking from the dataset as well as $\hat{p}$ (cont.)

EXAMPLE

#’ (ranking by decreasing probabilities)	c
1	1
2	0
3	1
4	0
5	1

Let’s plot the ROC/AUC for the following trained binary classifier (cont.)

EXAMPLE

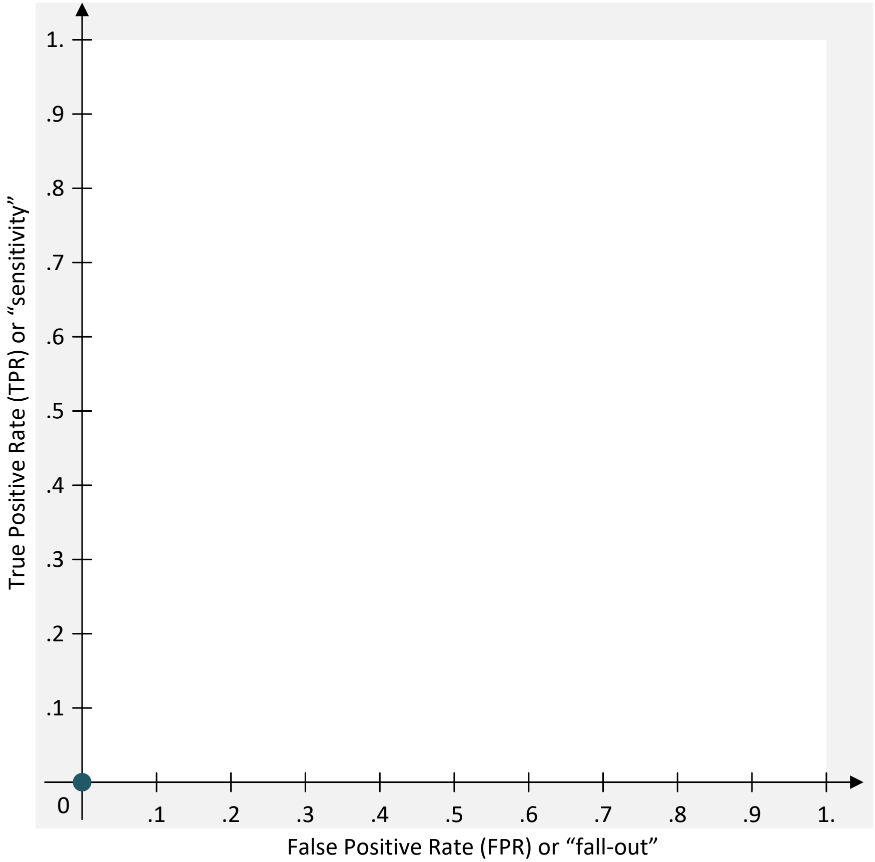
#’ (ranking by decreasing probabilities)	c
1	1
2	0
3	1
4	0
5	1

- $P = 3 \rightarrow {}^{1}/_{P} = {}^{1}/_{3}$
- $N = 2 \rightarrow {}^{1}/_{N} = {}^{1}/_{2}$

❹ Start at (0, 0)

EXAMPLE

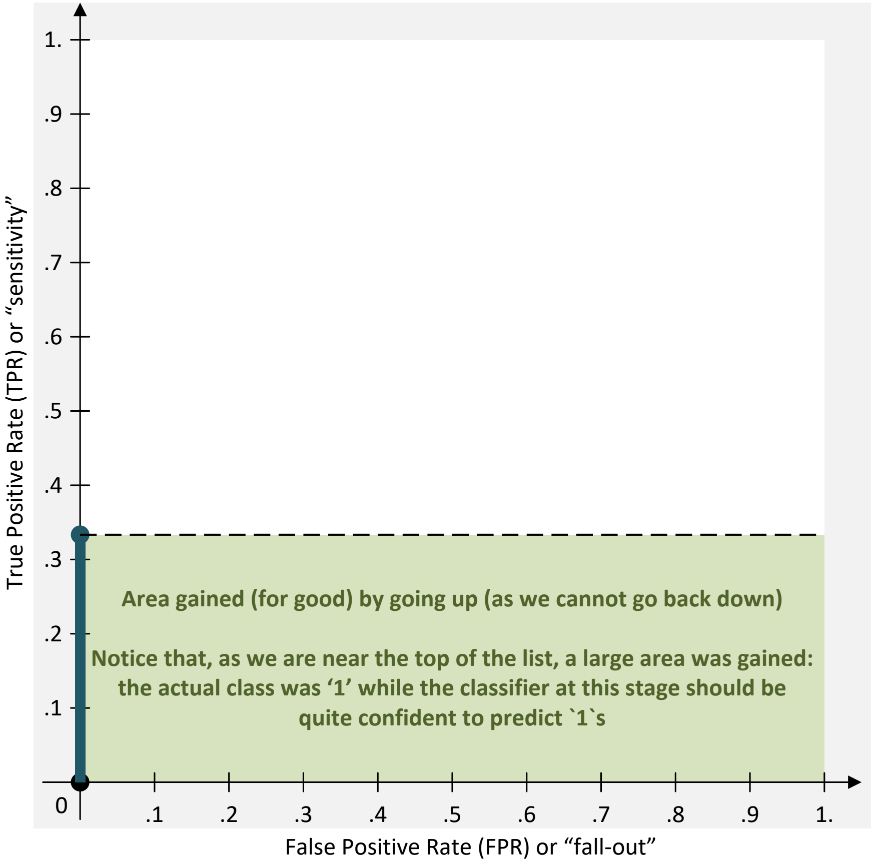
#′ (ranking by decreasing probabilities)	c
1	1
2	0
3	1
4	0
5	1



❺ Because $c = 1$, move up by ${}^{1}/_{P} = {}^{1}/_{3}$

EXAMPLE

#' (ranking by decreasing probabilities)	c
1	**1**
2	0
3	1
4	0
5	1



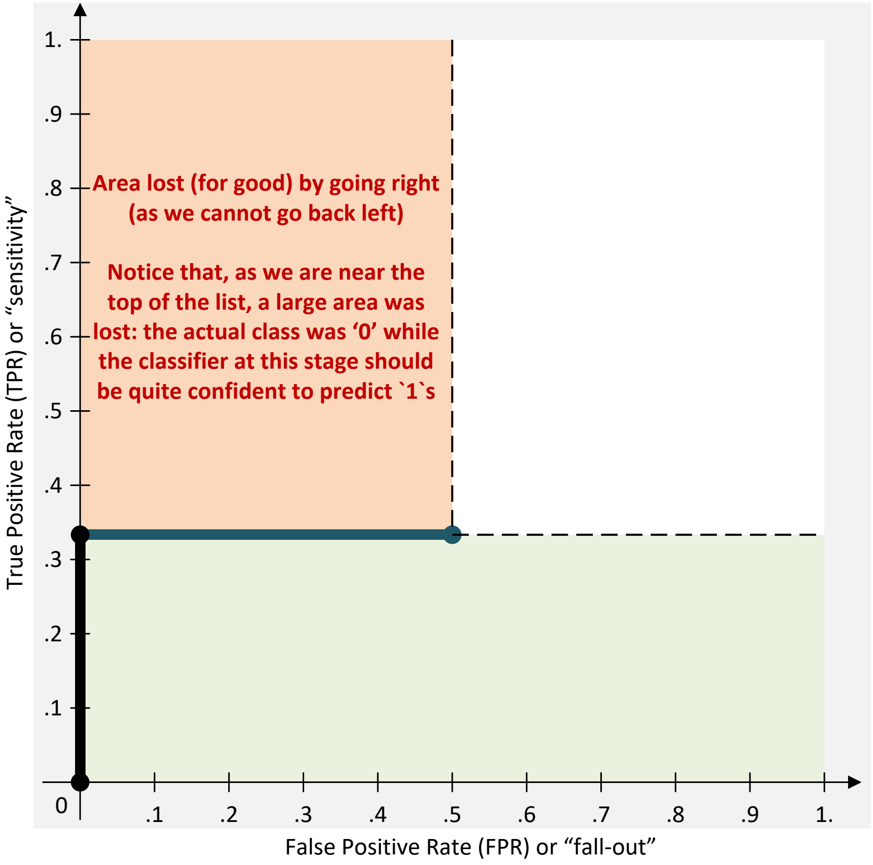
Area gained (for good) by going up (as we cannot go back down)

Notice that, as we are near the top of the list, a large area was gained: the actual class was '1' while the classifier at this stage should be quite confident to predict `1`s

❺ Because $c = 0$, move right by ${}^{1}/_{N} = {}^{1}/_{2}$

EXAMPLE

#' (ranking by decreasing probabilities)	c
1	*1*
2	**0**
3	1
4	0
5	1



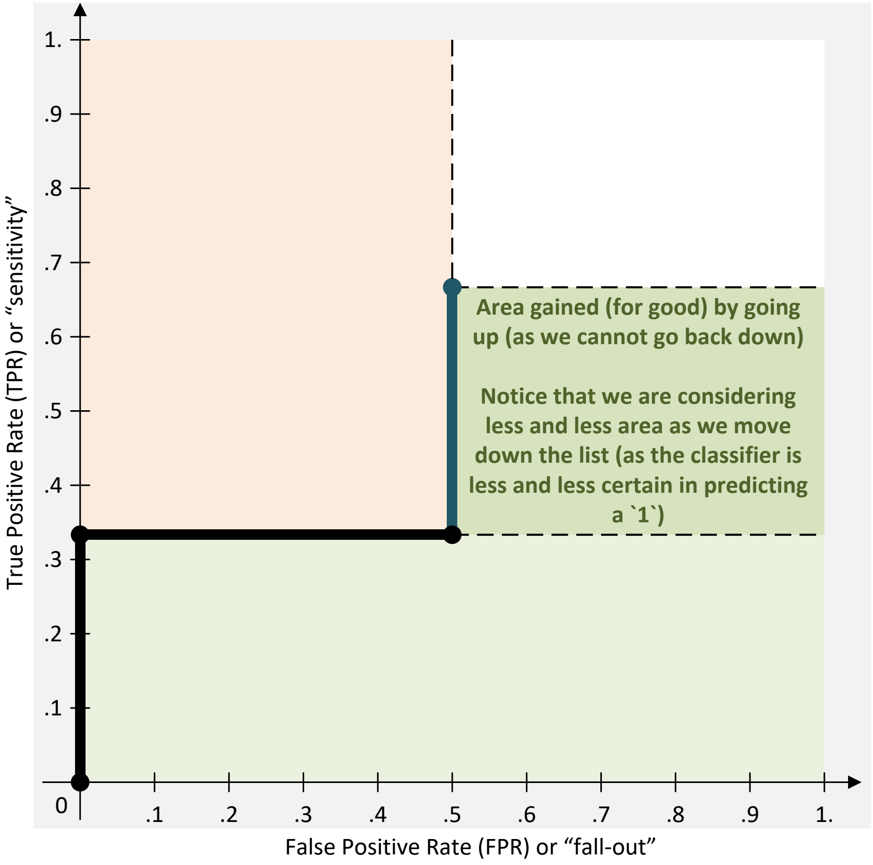
Area lost (for good) by going right (as we cannot go back left)

Notice that, as we are near the top of the list, a large area was lost: the actual class was '0' while the classifier at this stage should be quite confident to predict `1`s

❺ Because $c = 1$, move up by $^1/_P = {}^1/_3$

EXAMPLE

#' (ranking by decreasing probabilities)	c
1	*1*
2	*0*
3	**1**
4	0
5	1



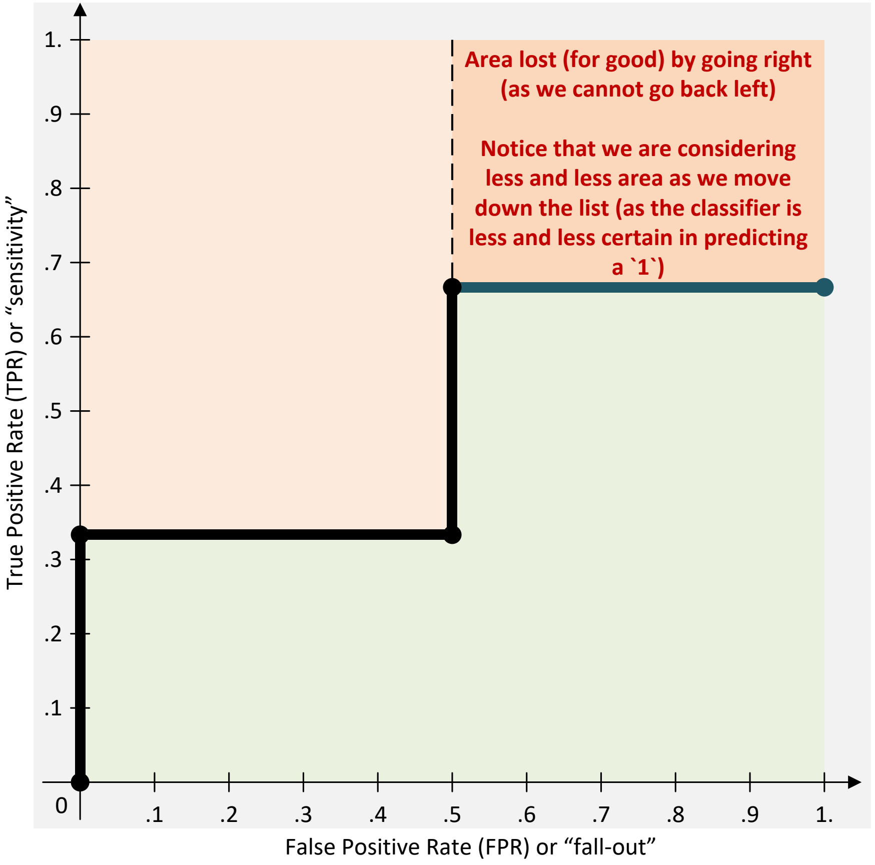
Area gained (for good) by going up (as we cannot go back down)

Notice that we are considering less and less area as we move down the list (as the classifier is less and less certain in predicting a `1`)

❺ Because $c = 0$, move left by $^1/_N = ^1/_2$

EXAMPLE

#’ (ranking by decreasing probabilities)	c
1	*1*
2	*0*
3	*1*
4	**0**
5	1



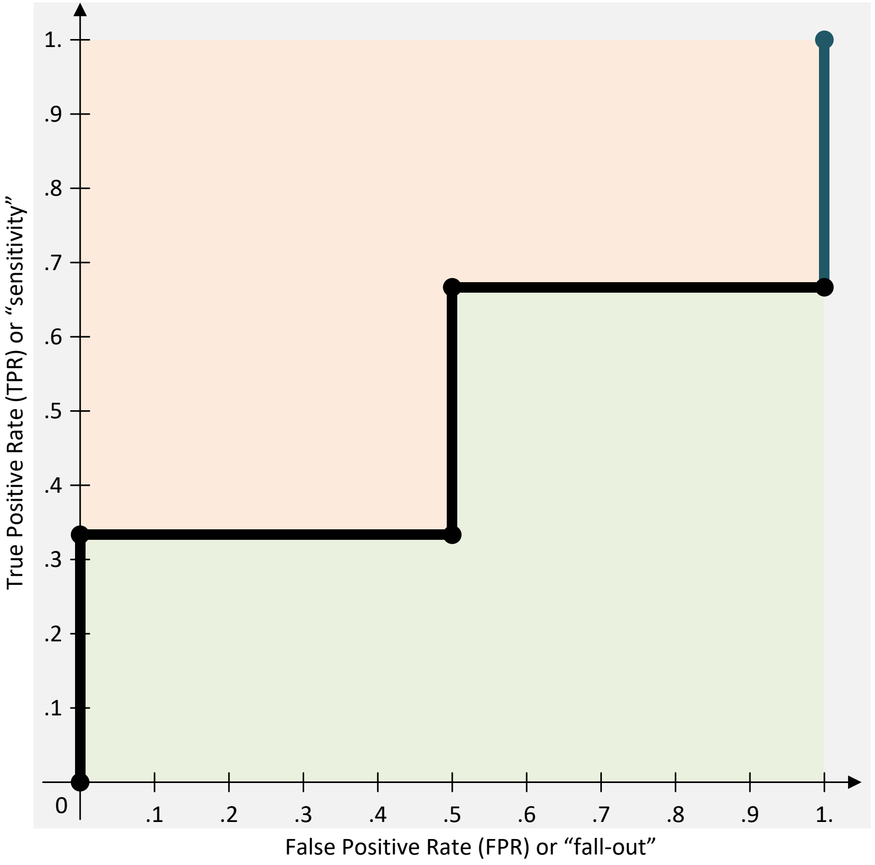
Area lost (for good) by going right (as we cannot go back left)

Notice that we are considering less and less area as we move down the list (as the classifier is less and less certain in predicting a `1`)

❺ Because $c = 1$, move up by ${}^{1}/_{P} = {}^{1}/_{3}$

EXAMPLE

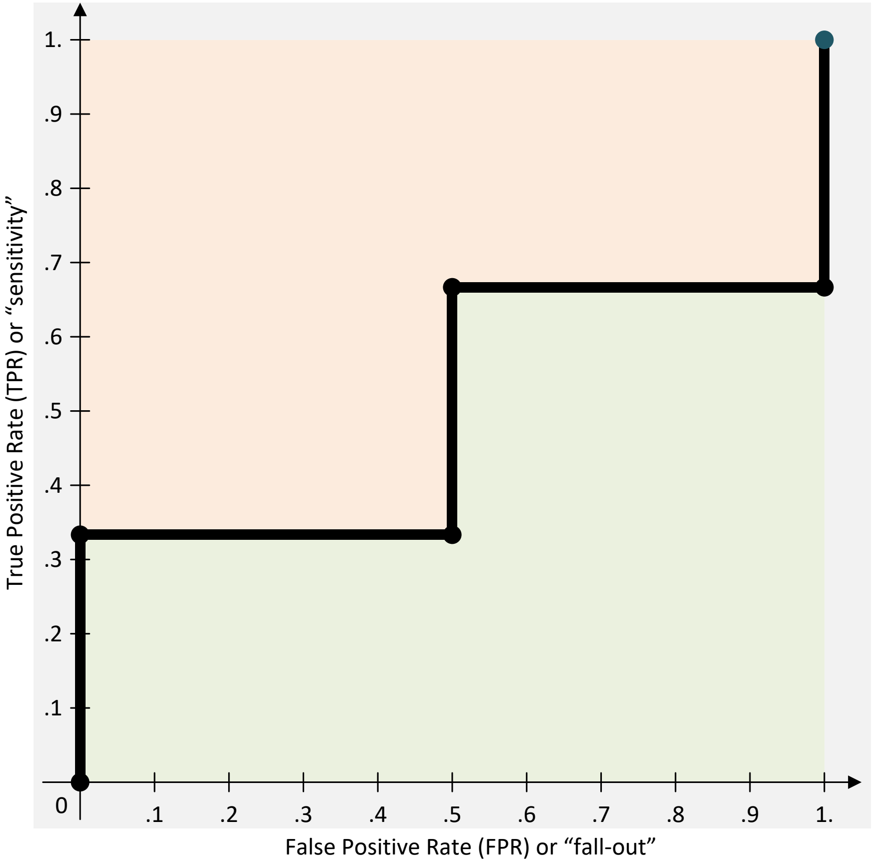
#' (ranking by decreasing probabilities)	c
1	*1*
2	*0*
3	*1*
4	*0*
5	**1**



❻ If not already at $(1, 1)$, go all the way to the right, then up all the way to $(1, 1)$

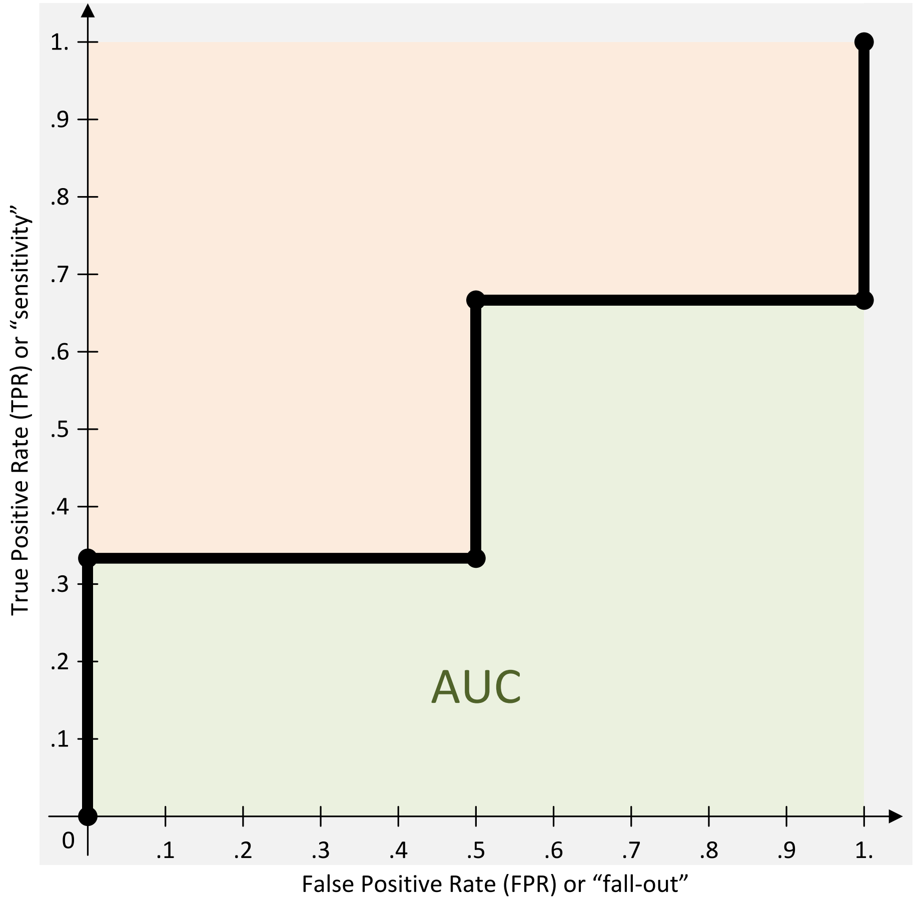
EXAMPLE

#' (ranking by decreasing probabilities)	*c*
1	*1*
2	*0*
3	*1*
4	*0*
5	*1*



Let’s plot the ROC/AUC for the following trained binary classifier (cont.)

EXAMPLE



Plotting an ROC curve (cont.)

- Notes
  - We don’t rely on a threshold (e.g., .5) for plotting ROC curves. Indeed, moving up or right is independent of $\hat{p}$ (we discarded it in step ❸) and only relies on a decreasing ranking of $\hat{p}$ and then c
  - As a matter of fact, you can use ROC curves to select the best threshold but we won’t address it here