

## Lecture 12

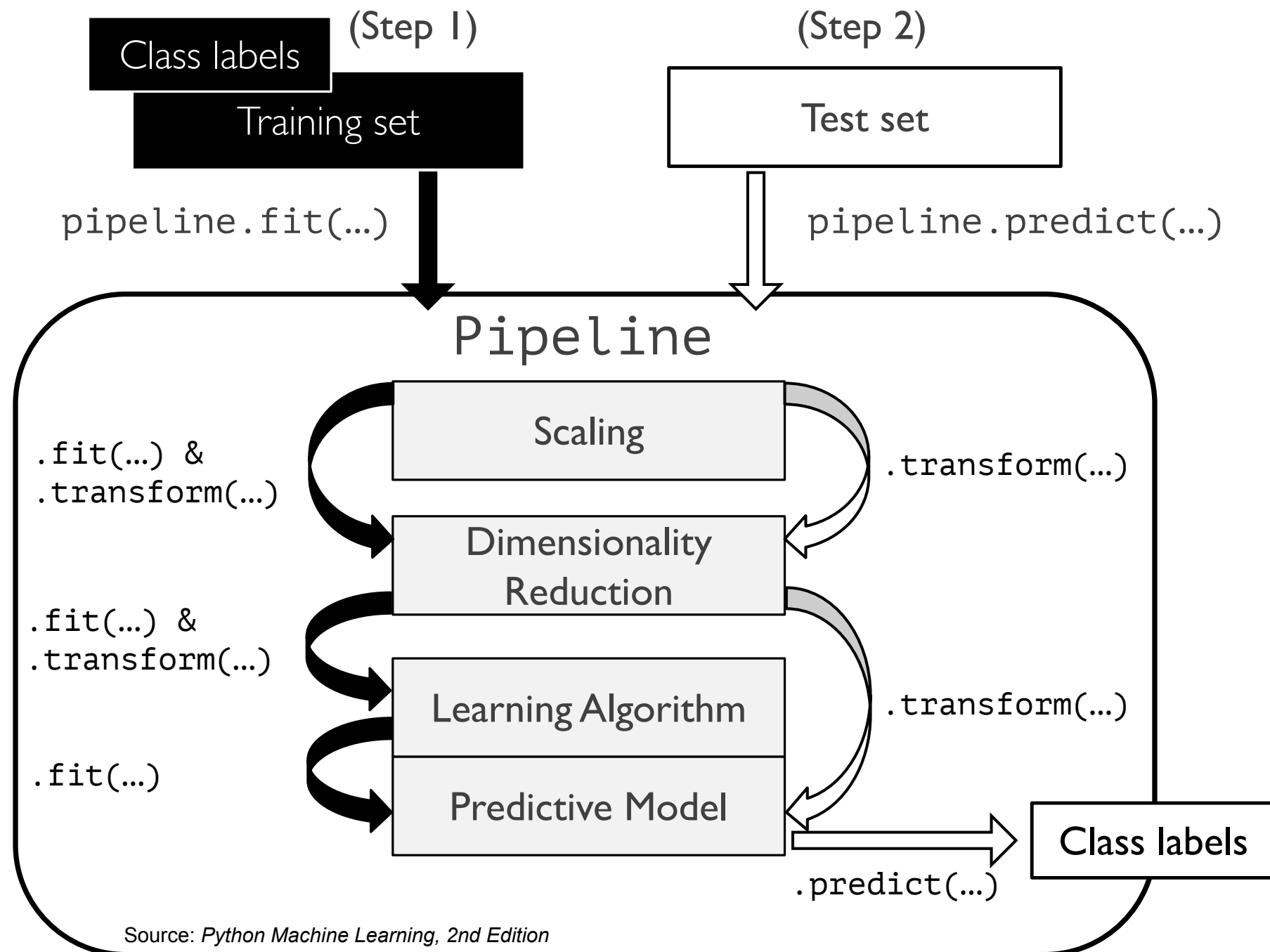
# Model Evaluation 5: Performance Metrics

STAT 479: Machine Learning, Fall 2018

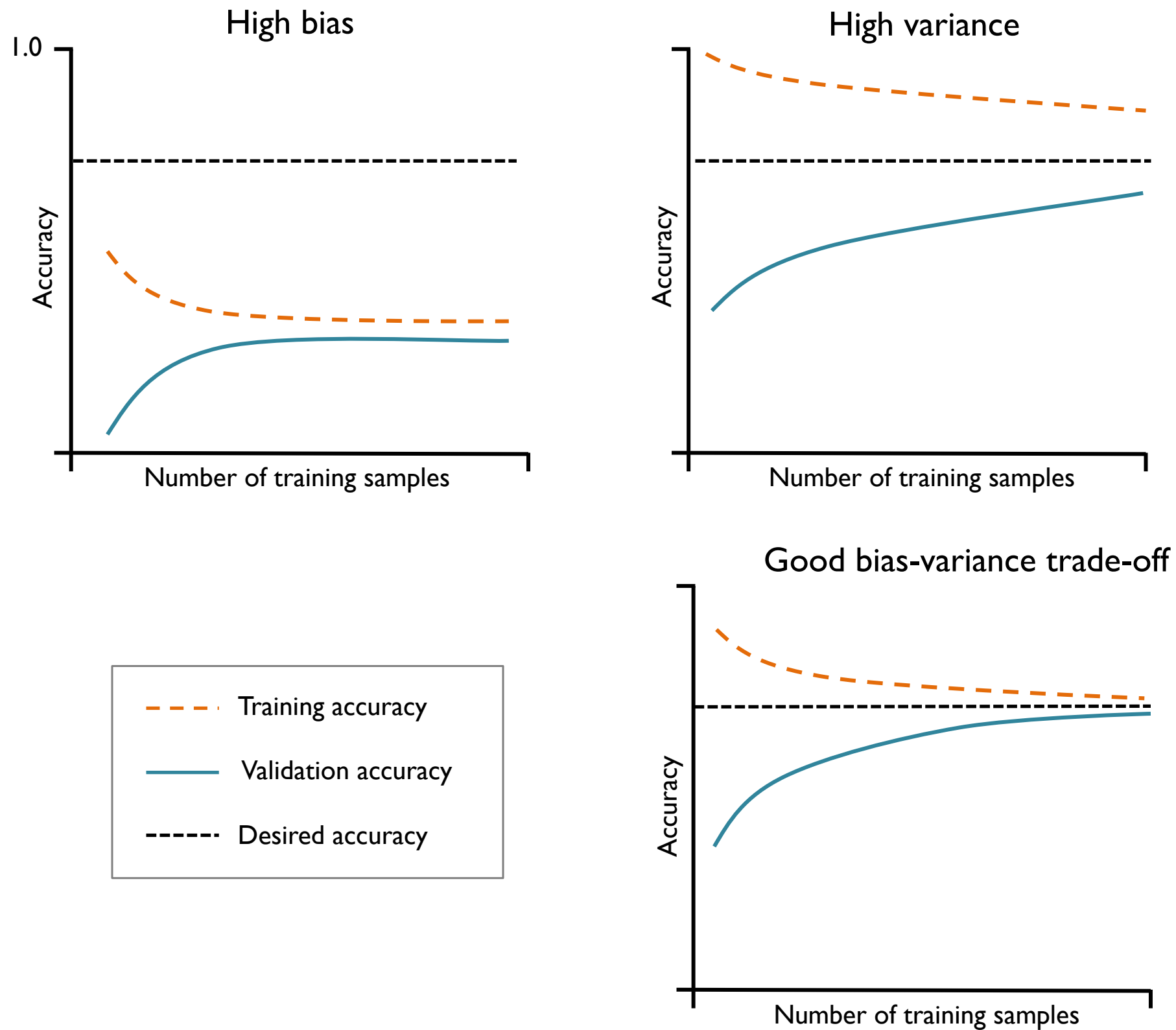
Sebastian Raschka

<http://stat.wisc.edu/~sraschka/teaching/stat479-fs2018/>

# Recap



# Recap



Source: *Python Machine Learning, 2nd Edition*

# 2x2 Confusion Matrix

		Predicted class	
		$P$	$N$
Actual Class	$P$	True Positives (TP)	False Negatives (FN)
	$N$	False Positives (FP)	True Negatives (TN)

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} = 1 - ACC \quad (1)$$

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR \quad (2)$$

# Confusion Matrix for Multi-Class Settings

		Predicted Labels		
		Class 0	Class 1	Class 2
True Labels	Class 0	$T(0,0)$		
	Class 1		$T(1,1)$	
	Class 2			$T(2,2)$

Confusions matrices are traditionally for binary class problems but we can easily generalize it to multi-class settings

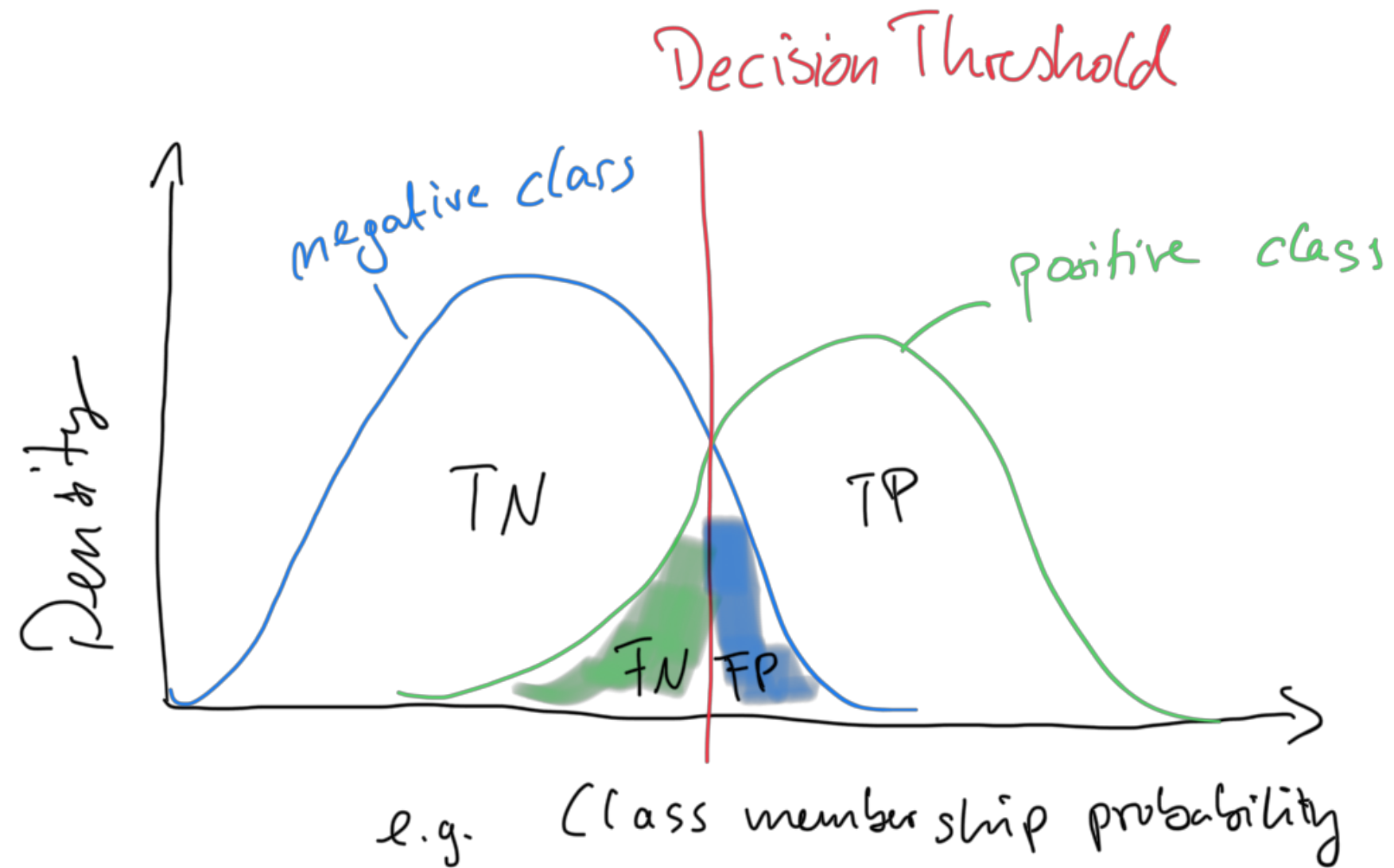
# False Positive Rate and False Negative Rate

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \quad (3)$$

$$TPR = \frac{TP}{P} = \frac{TP}{FN + TP} \quad (4)$$

- Think of it in a spam classification problem (what are true positives, and if you had to pick one at the expense of the other: would you rather decrease the FPR or increase the TPR?)

# False Positive Rate and False Negative Rate



# Precision, Recall, and F1 Score

$$PRE = \frac{TP}{TP + FP} \quad (5)$$

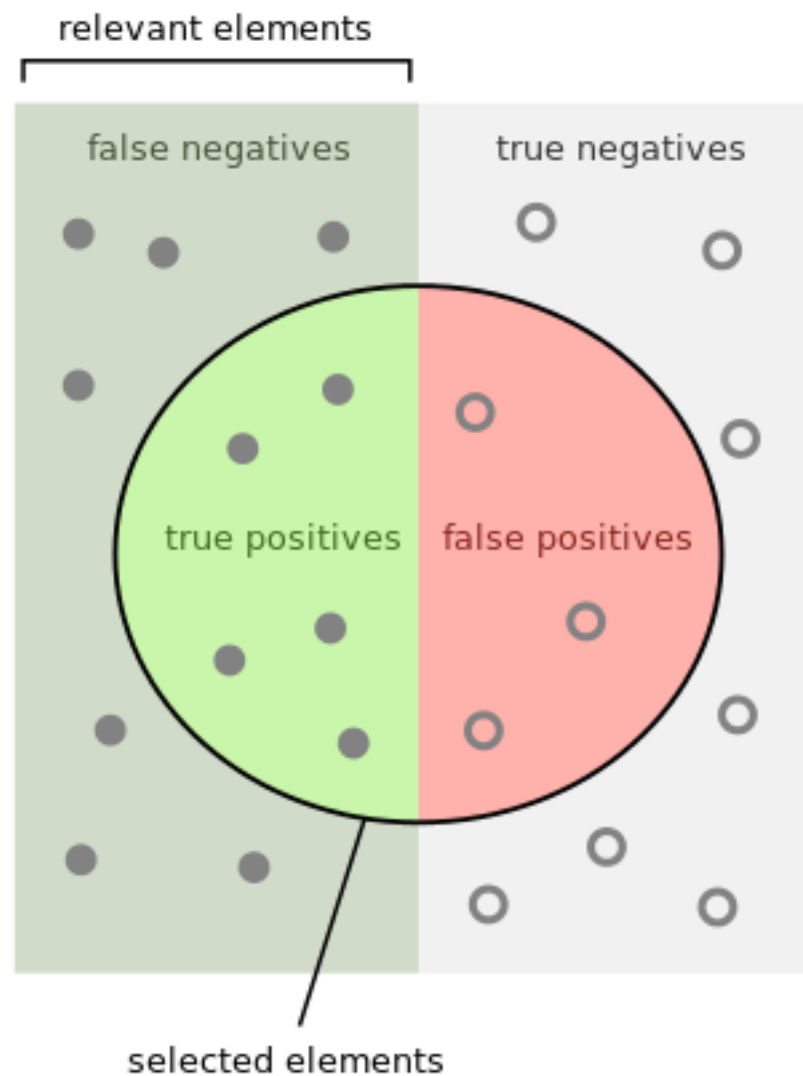
$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP} \quad (6)$$

$$F_1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC} \quad (7)$$

- Terms that are more popular in Information Technology
- Recall is actually just another term for True Positive Rate (or "sensitivity")



# Precision and Recall



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

[https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall)

# Sensitivity and Specificity

$$SEN = TPR = REC = \frac{TP}{P} = \frac{TP}{FN + TP} \quad (8)$$

$$SPC = TNR = \frac{TN}{N} = \frac{TN}{FP + TN} \quad (9)$$

Sensitivity measures the recovery rate of the Positives and complimentary, the Specificity measures the recovery rate of the Negatives.

# Matthew's Correlation Coefficient

- Matthews correlation coefficient (MCC) was first formulated by Brian W. Matthews [1] in 1975 to assess the performance of protein secondary structure predictions
- The MCC can be understood as a specific case of a linear correlation coefficient (Pearson  $r$ ) for a binary classification setting
- Considered as especially useful in unbalanced class settings.
- The previous metrics take values in the range between 0 (worst) and 1 (best)
- The MCC is bounded between the range 1 (perfect correlation between ground truth and predicted outcome) and -1 (inverse or negative correlation) — a value of 0 denotes a random prediction.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (10)$$

[1] Brian W Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)- Protein Structure*, 405(2):442–451, 1975.

# Balanced Accuracy / Average Per-Class Accuracy

		Predicted Labels		
		Class 0	Class 1	Class 2
True Labels	Class 0	T(0,0)		
	Class 1		T(1,1)	
	Class 2			T(2,2)

$$ACC = \frac{T}{n}$$

		Predicted Labels		
		Class 0	Class 1	Class 2
True Labels	Class 0	3	0	0
	Class 1	7	50	12
	Class 2	0	0	18

$$ACC = \frac{3 + 50 + 18}{90} \approx 0.79$$

$$APC\ ACC = \frac{83/90 + 71/90 + 78/90}{3} \approx 0.86$$

# Balanced Accuracy / Average Per-Class Accuracy

		Predicted Labels	
		Class 0	Neg Class
True Labels	Class 0	3	0
	Neg Class	7	80

		Predicted Labels	
		Class 1	Neg Class
True Labels	Class 1	50	19
	Neg Class	0	21

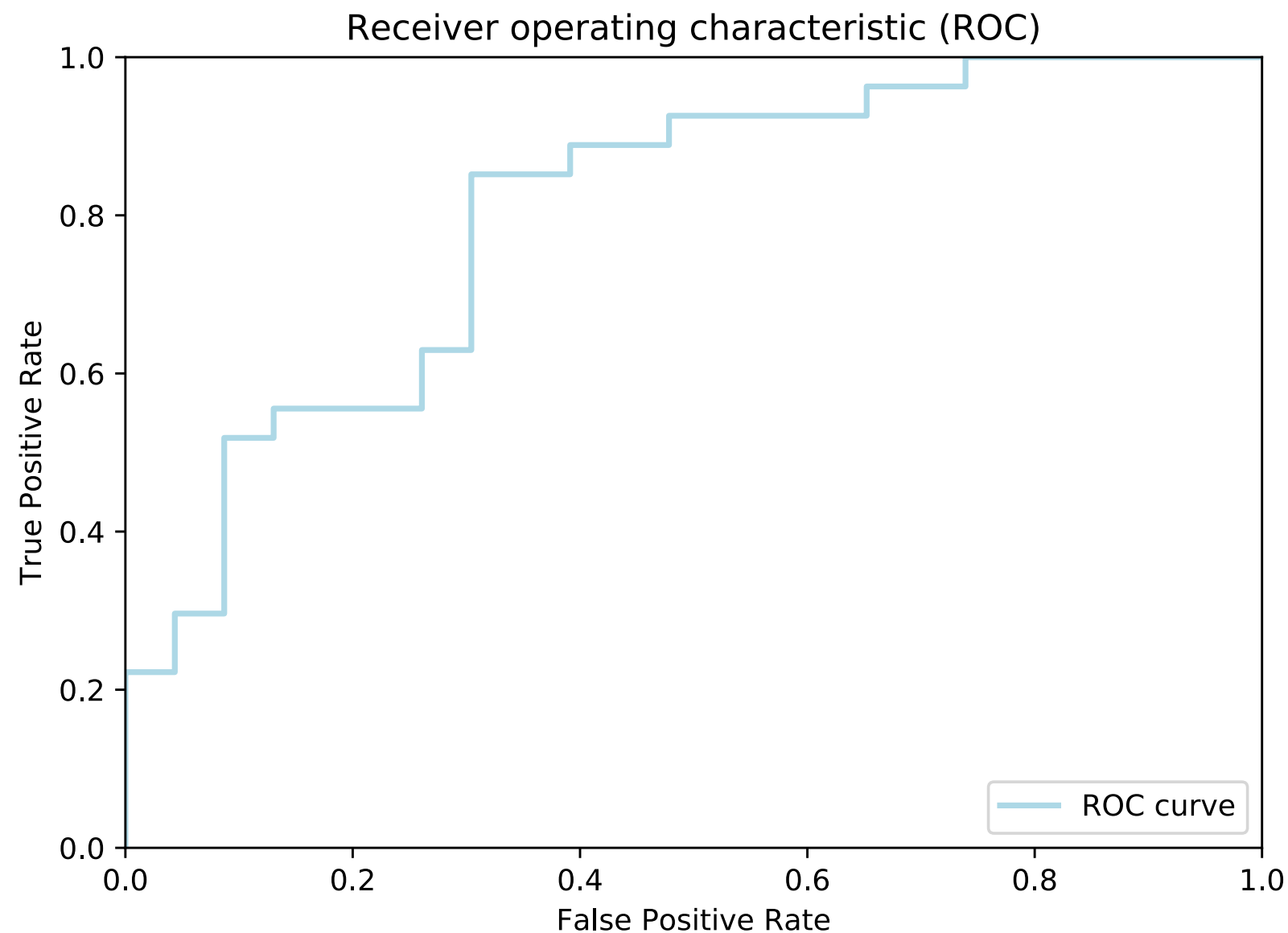
		Predicted Labels	
		Class 2	Neg Class
True Labels	Class 2	18	0
	Neg Class	12	60

		Predicted Labels		
		Class 0	Class 1	Class 2
True Labels	Class 0	3	0	0
	Class 1	7	50	12
	Class 2	0	0	18

$$APC\ ACC = \frac{83/90 + 71/90 + 78/90}{3} \approx 0.86$$

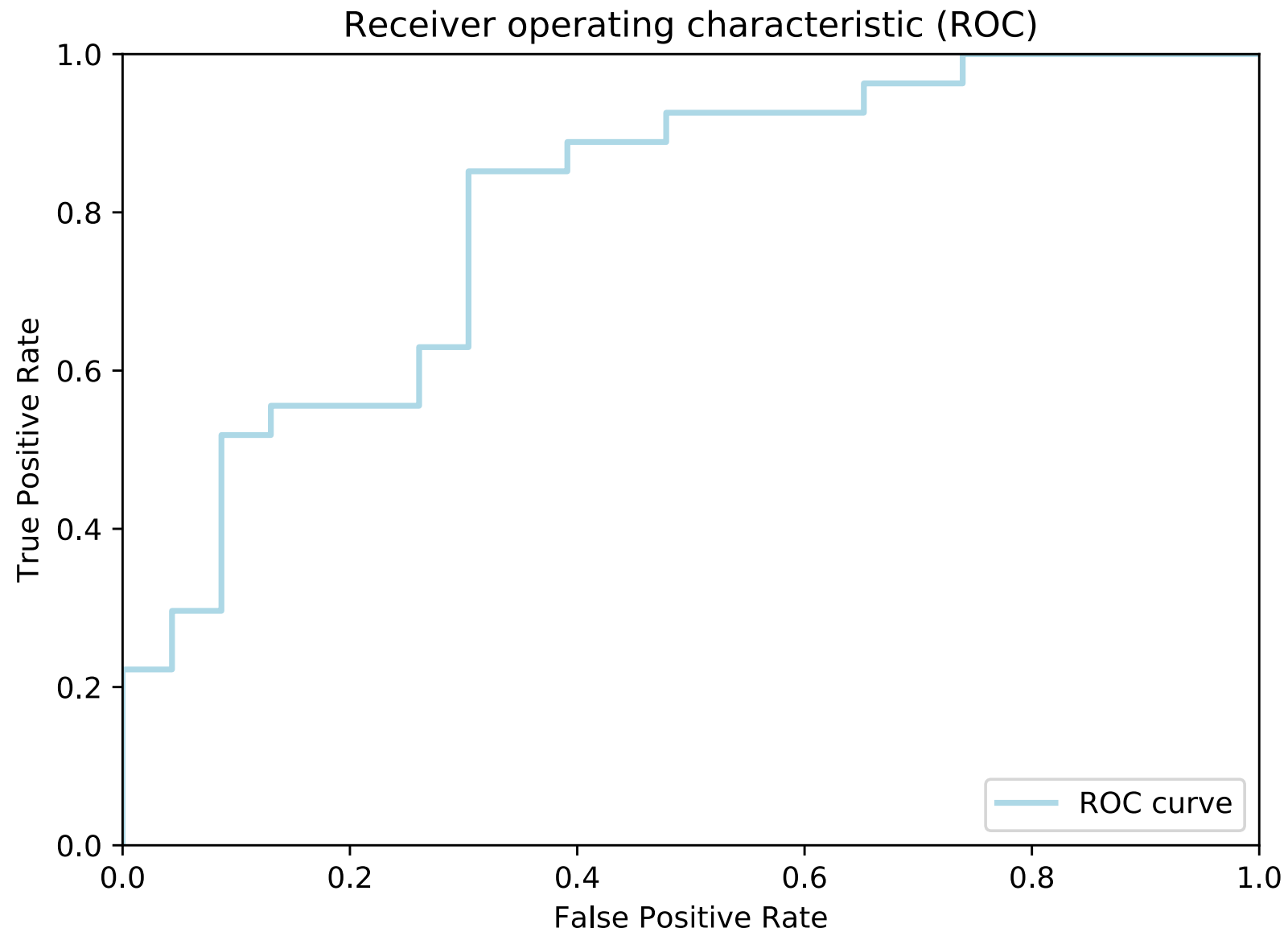
# Receiver Operating Characteristic curve (ROC curve)

- Trade-off between True Positive Rate and False Positive Rate
- ROC can be plotted by changing the prediction threshold
- ROC term comes from "Radar Receiver Operators"  
(analysis of radar [**R**adio **D**irection **A**nd **R**anging] images)

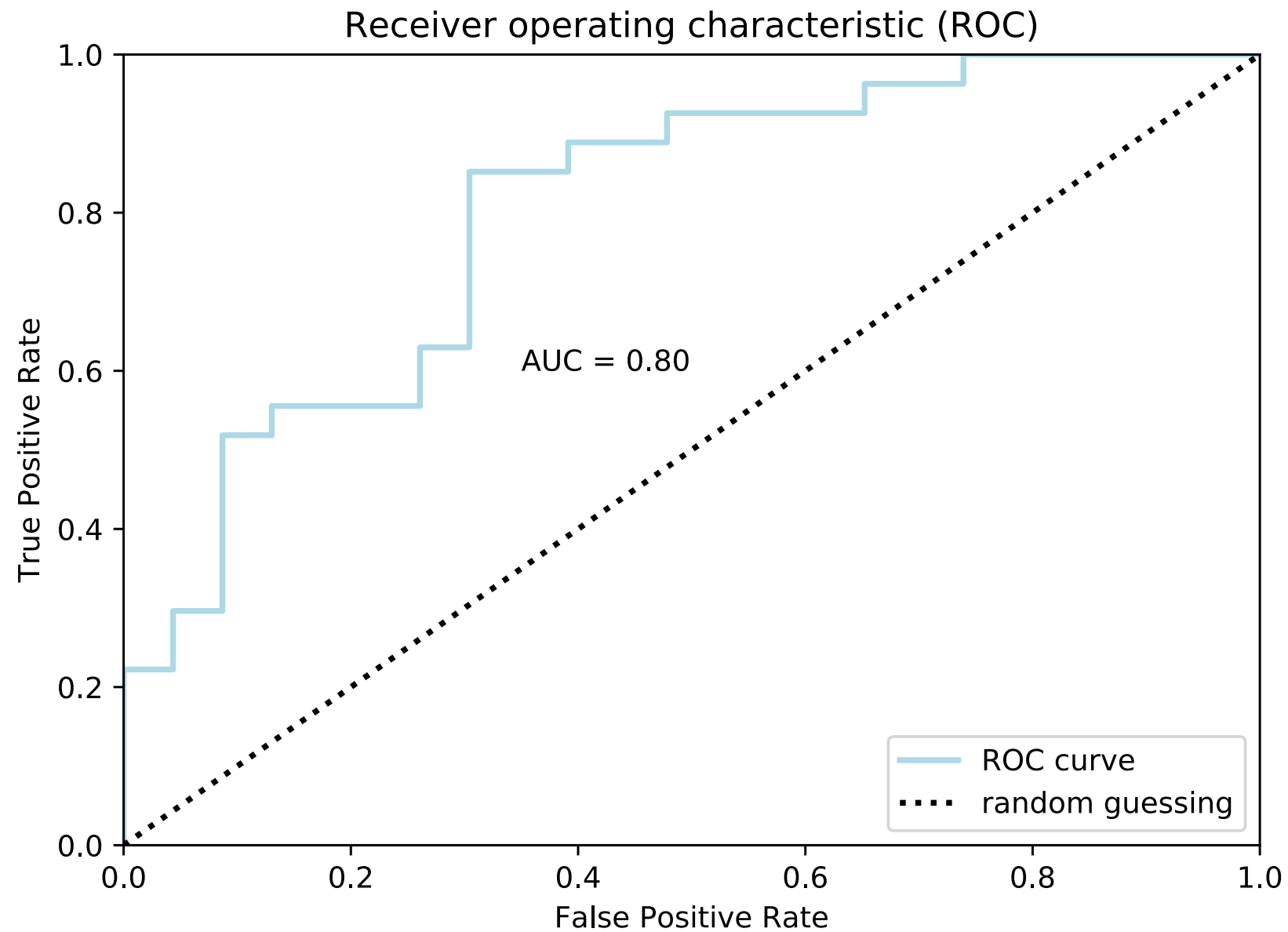


# Receiver Operating Characteristic curve (ROC curve)

- $(1, 1)$  = Perfect Prediction
- $(0.5, 0.5)$  = Random Prediction

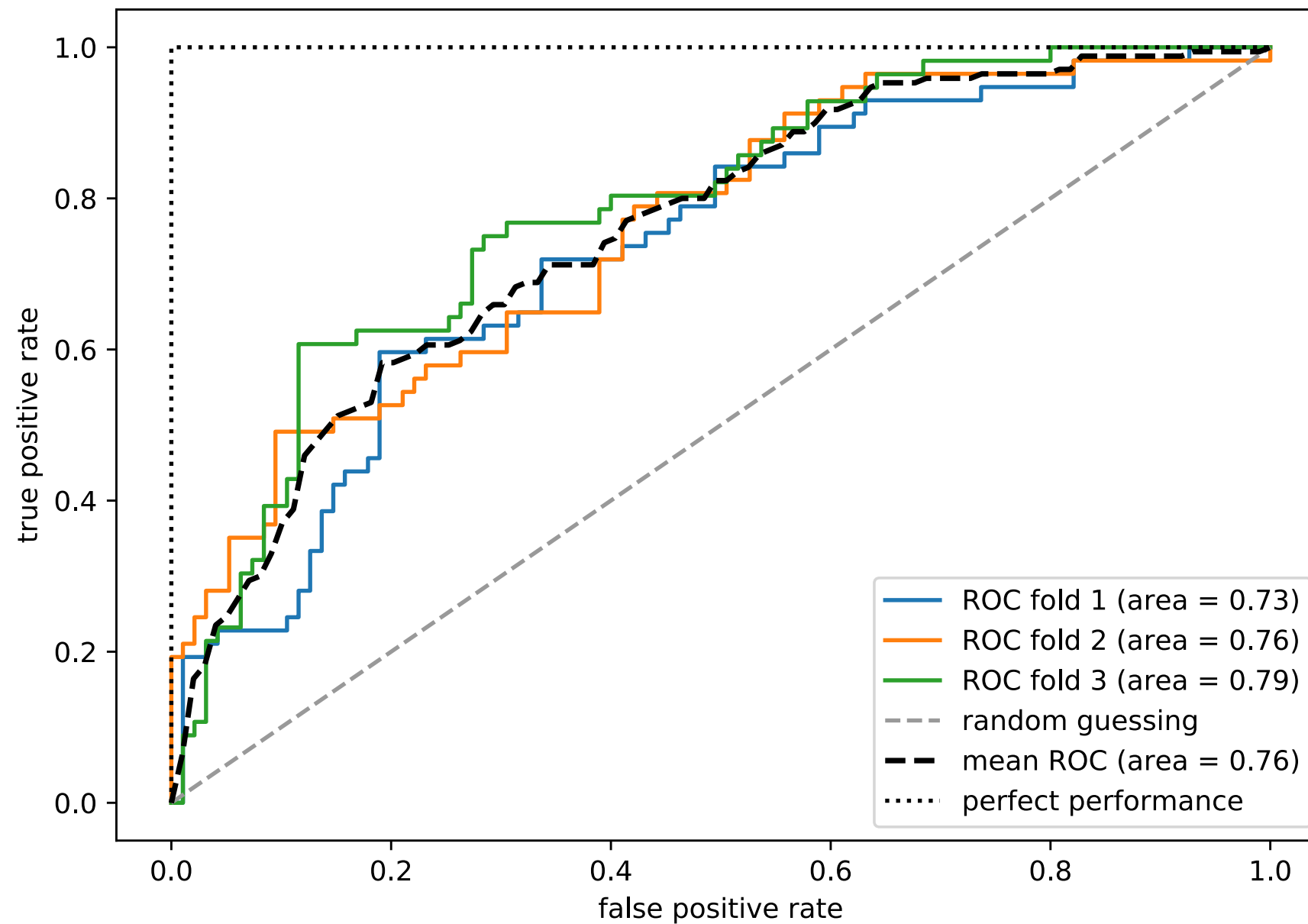


# ROC Area Under the Curve (AUC)

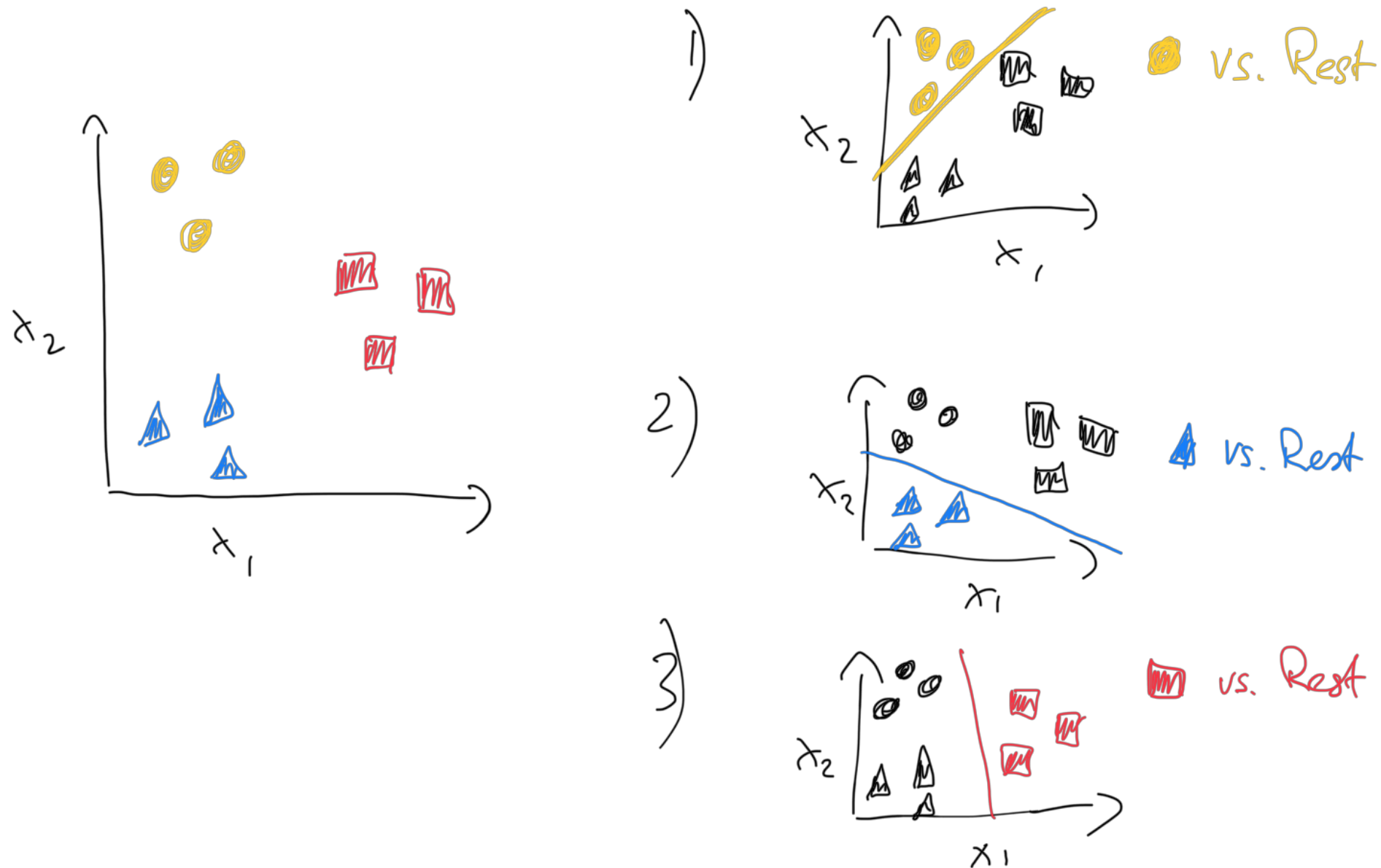




# ROC and k-Fold Cross-Validation



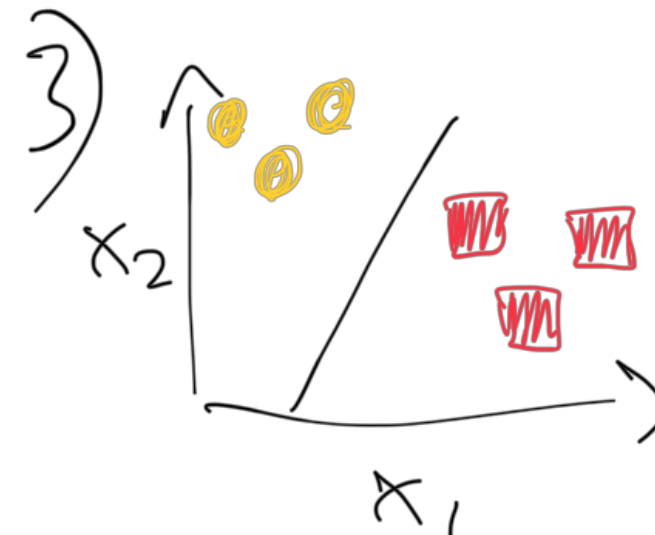
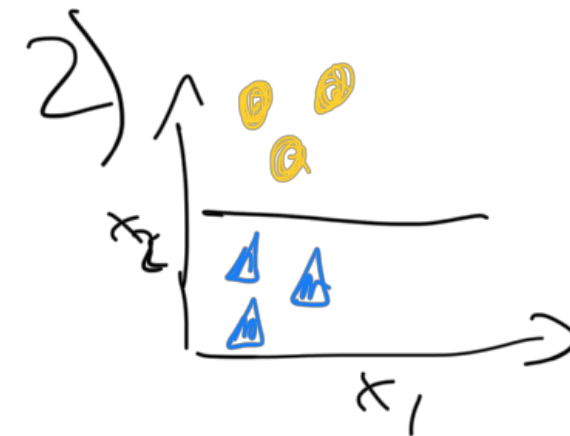
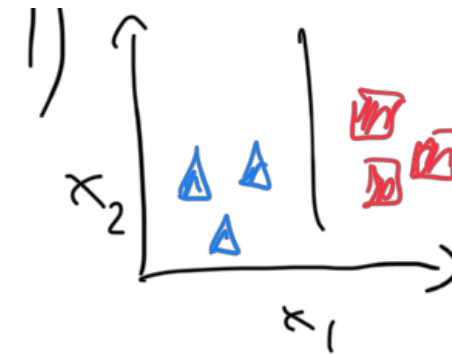
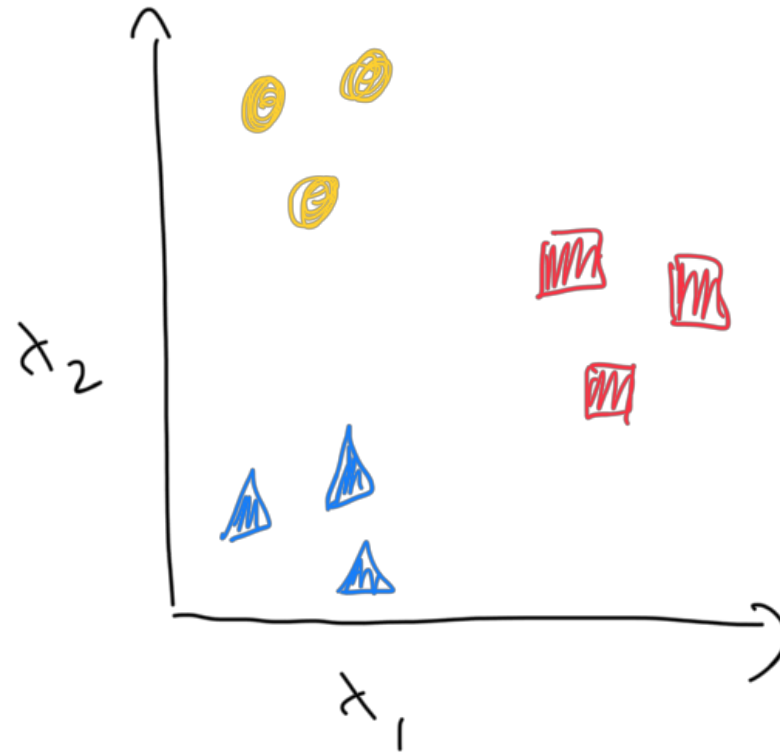
# Binary Classifiers and One-vs-Rest (OvR) / One-vs-All (OvA)



Big O:  $O(?)$

Choose the class with the highest confidence score

# Binary Classifiers and One-vs-One (OvO) / All-vs-All (AvA)



$\text{num\_classes} \times (\text{num\_classes} - 1) / 2$

Big O:  $O(?)$

Select the class by majority vote (and use confidence score in case of ties)

# Macro and Micro Averaging

$$PRE_{micro} = \frac{TP_1 + \dots + TP_c}{TP_1 + \dots + TP_c + FP_1 + \dots + FP_c}$$

$$PRE_{macro} = \frac{PRE_1 + \dots + PRE_c}{c}$$

Micro-averaging is useful if we want to weight each instance or prediction equally, whereas macro-averaging weights all classes equally to evaluate the overall performance of a classifier with regard to the most frequent class labels.

# Dealing with Class Imbalance

## User Guide

- 1. Introduction
  - 1.1. API's of imbalanced-learn samplers
  - 1.2. Problem statement regarding imbalanced data sets
- 2. Over-sampling
  - 2.1. A practical guide
    - 2.1.1. Naive random over-sampling
    - 2.1.2. From random over-sampling to SMOTE and ADASYN
    - 2.1.3. Ill-posed examples
    - 2.1.4. SMOTE variants
  - 2.2. Mathematical formulation
    - 2.2.1. Sample generation
    - 2.2.2. Multi-class management
- 3. Under-sampling
  - 3.1. Prototype generation
  - 3.2. Prototype selection
    - 3.2.1. Controlled under-sampling techniques
      - 3.2.1.1. Mathematical formulation
    - 3.2.2. Cleaning under-sampling techniques
      - 3.2.2.1. Tomek's links
      - 3.2.2.2. Edited data set using nearest neighbours
      - 3.2.2.3. Condensed nearest neighbors and derived algorithms
      - 3.2.2.4. Instance hardness threshold
- 4. Combination of over- and under-sampling

[https://imbalanced-learn.readthedocs.io/en/stable/user\\_guide.html](https://imbalanced-learn.readthedocs.io/en/stable/user_guide.html)

# Reading Assignment

- *Python Machine Learning, 2nd Edition*  
Chapter 6: Learning Best Practices for Model Evaluation and Hyperparameter Tuning