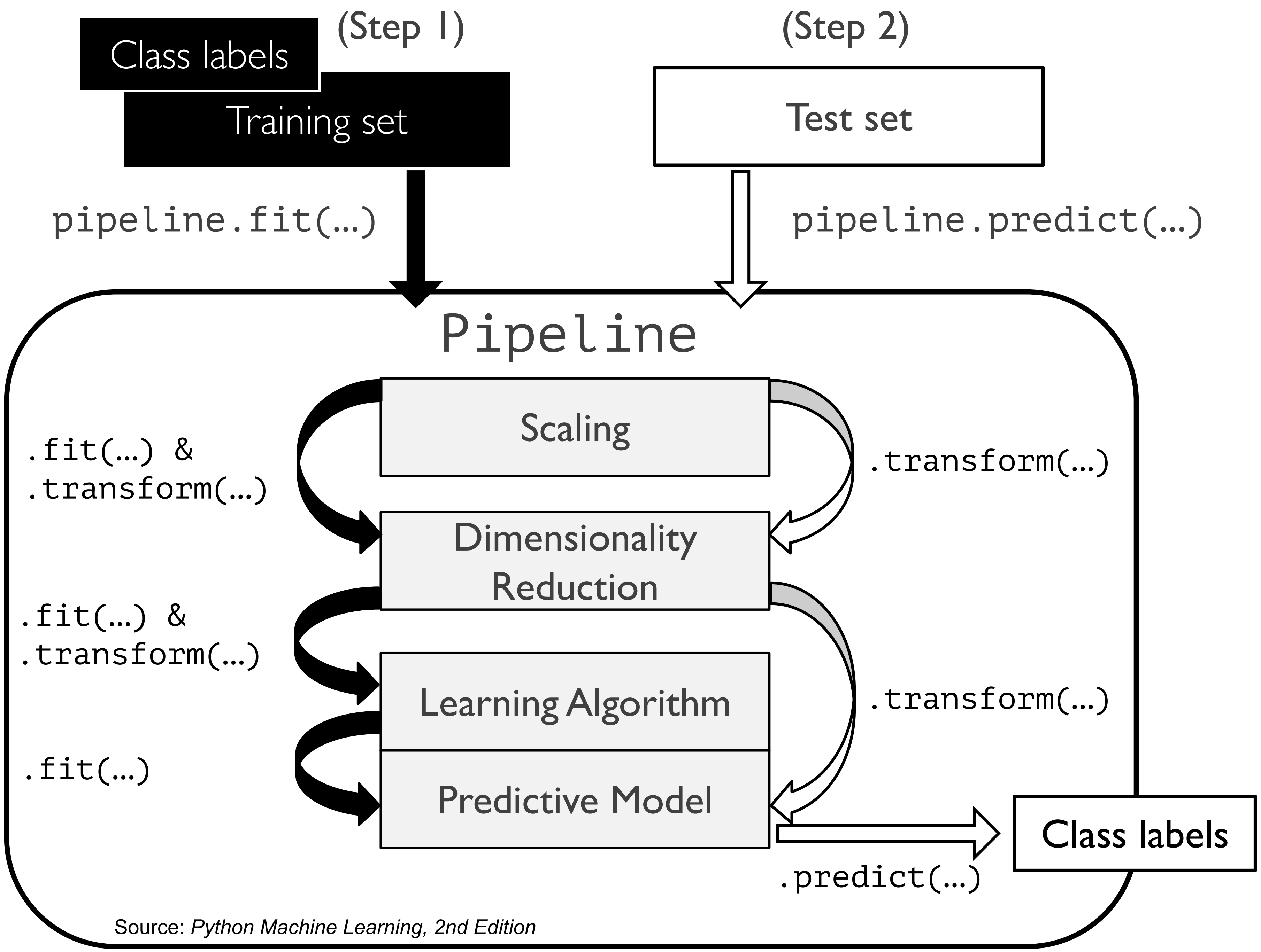
Lecture 12

Model Evaluation 5:
Performance Metrics

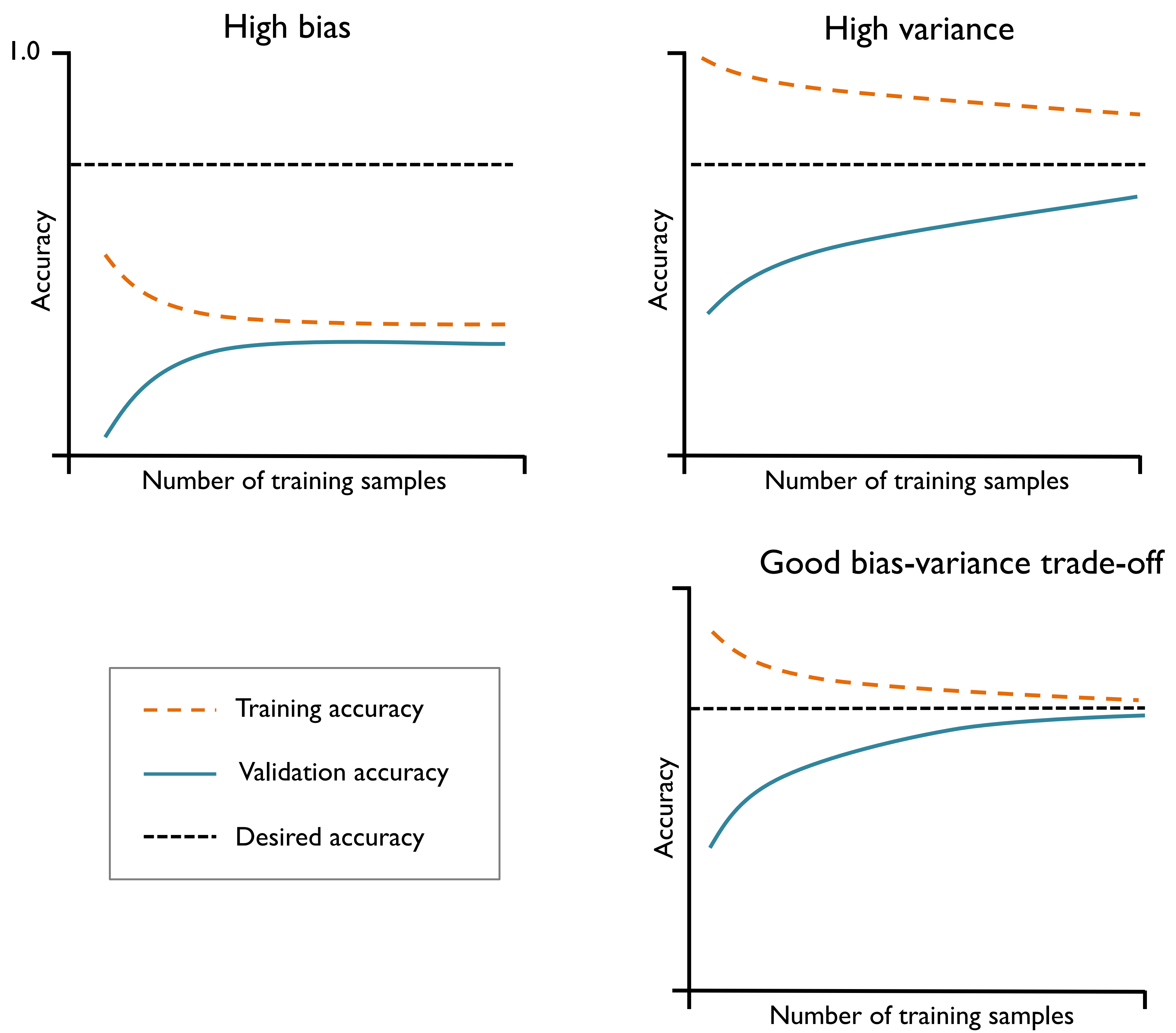
STAT 479: Machine Learning, Fall 2018
Sebastian Raschka
http://stat.wisc.edu/~sraschka/teaching/stat479-fs2018/

Recap



(Step 1)
Class labels
Training set
pipeline.fit(...)
(Step 2)
Test set
pipeline.predict(...)
Pipeline
.fit(...) &
.transform(...)
Scaling
.transform(...)
Dimensionality Reduction
.fit(...) &
.transform(...)
Learning Algorithm
.transform(...)
.fit(...)
Predictive Model
Class labels
.predict(...)
Source: *Python Machine Learning, 2nd Edition*

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)

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} = 1 - ACC \tag{1}$$

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

Confusion Matrix for Multi-Class Settings

Predicted Labels

True Labels

	Class 0	Class 1	Class 2
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

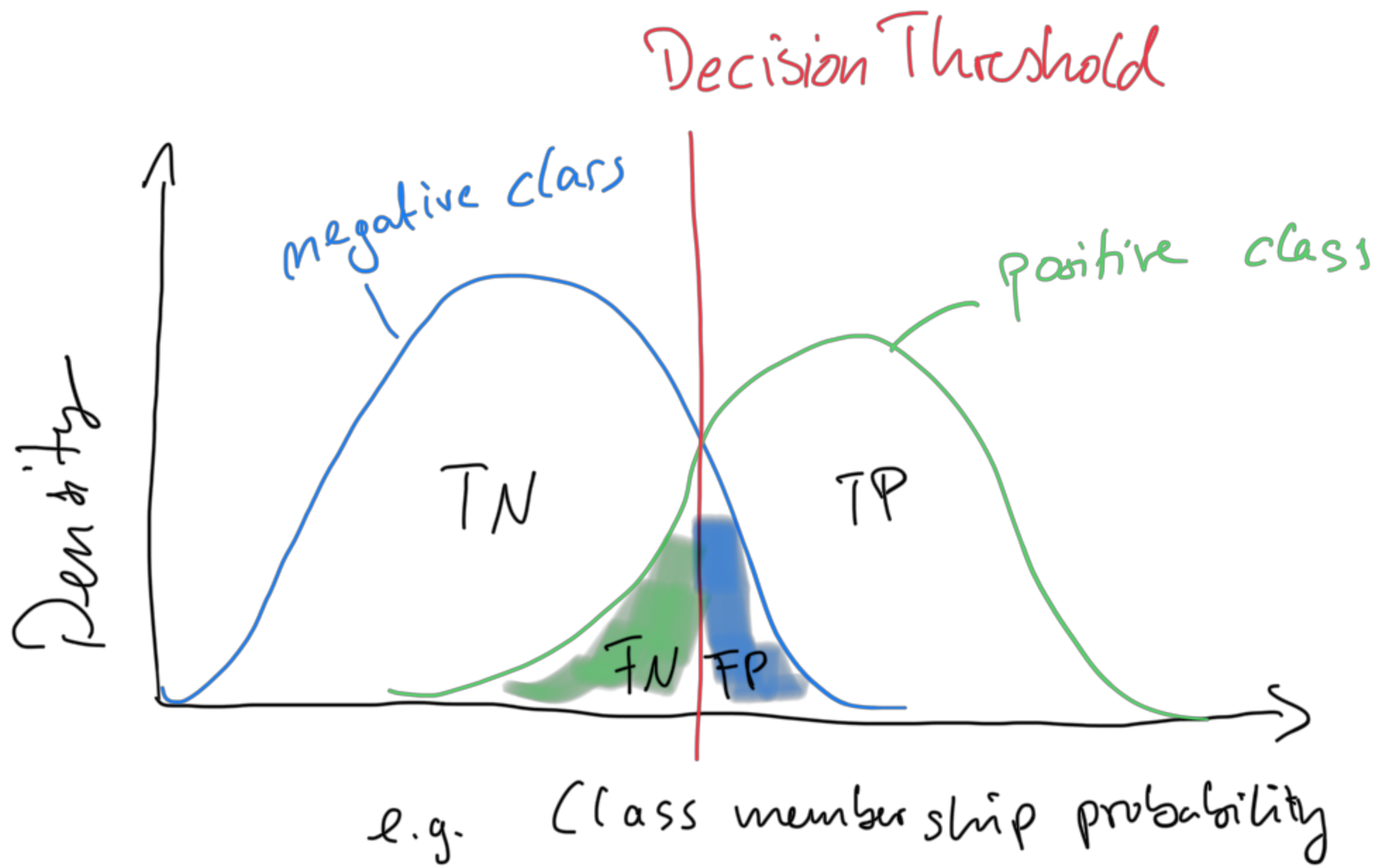
False Positive Rate and False Negative Rate

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{3}$$

$$TPR = \frac{TP}{P} = \frac{TP}{FN + TP} \tag{4}$$

- 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

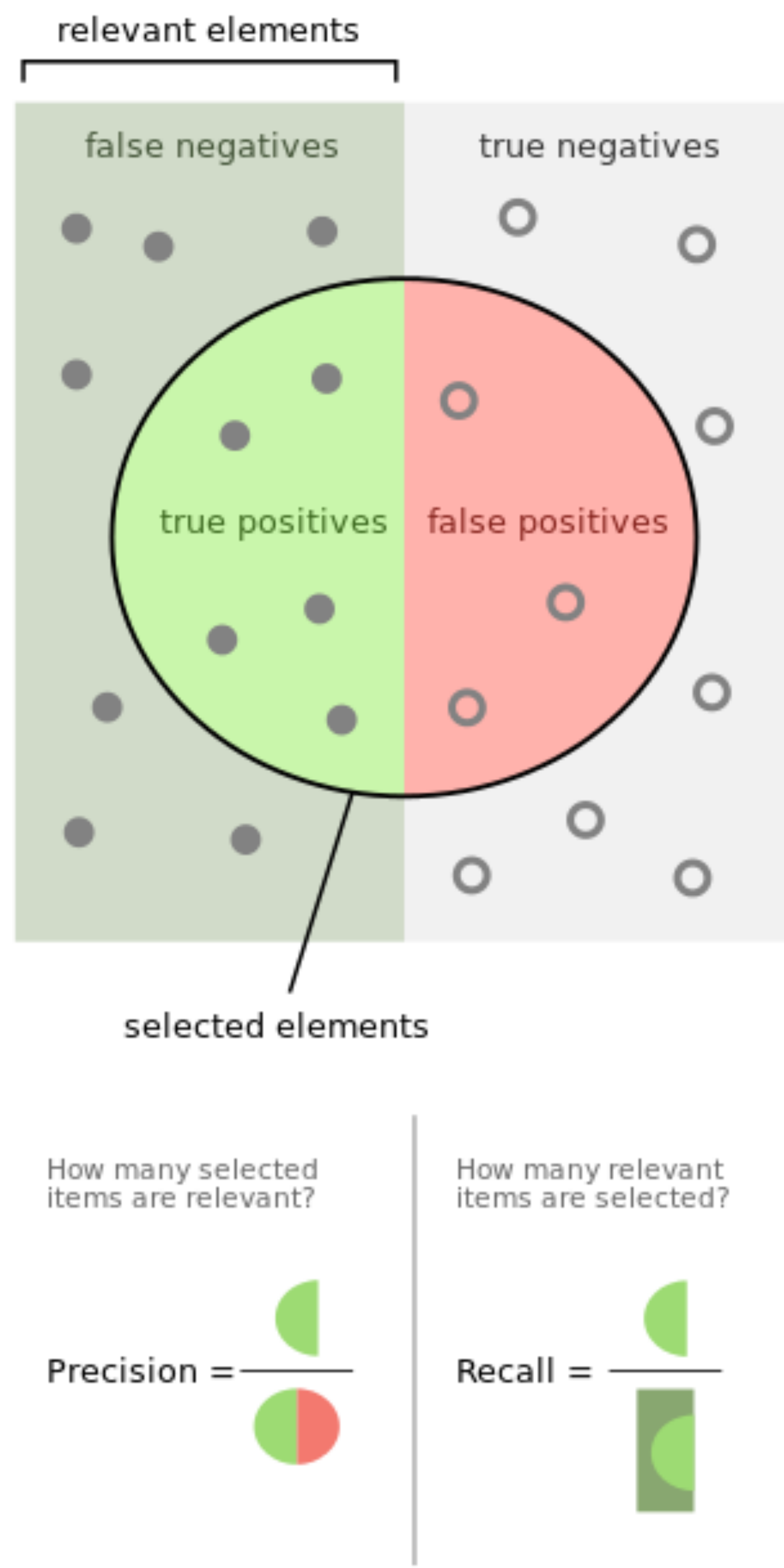
$$PRE = \frac{TP}{TP + FP} \tag{5}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP} \tag{6}$$

$$F_1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC} \tag{7}$$

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

Precision and Recall



https://en.wikipedia.org/wiki/Precision_and_recall

Sensitivity and Specificity

$$SEN = TPR = REC = \frac{TP}{P} = \frac{TP}{FN + TP} \tag{8}$$

$$SPC = TNR = \frac{TN}{N} = \frac{TN}{FP + TN} \tag{9}$$

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.

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

[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

True Labels

	Class 0	Class 1	Class 2
Class 0	T(0,0)		
Class 1		T(1,1)	
Class 2			T(2,2)

Predicted Labels

True Labels

	Class 0	Class 1	Class 2
Class 0	3	0	0
Class 1	7	50	12
Class 2	0	0	18

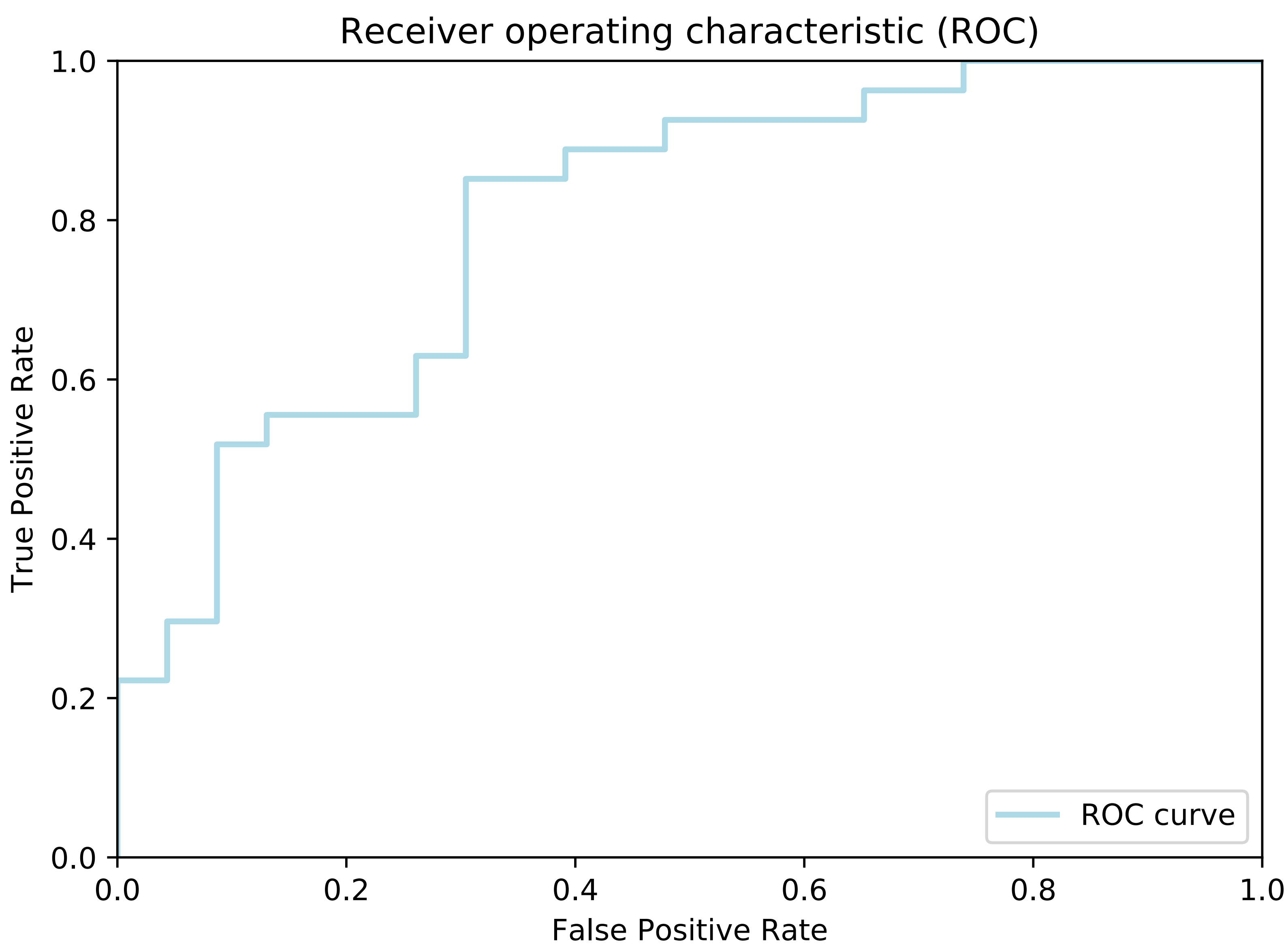
$$ACC = \frac{T}{n}$$

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

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

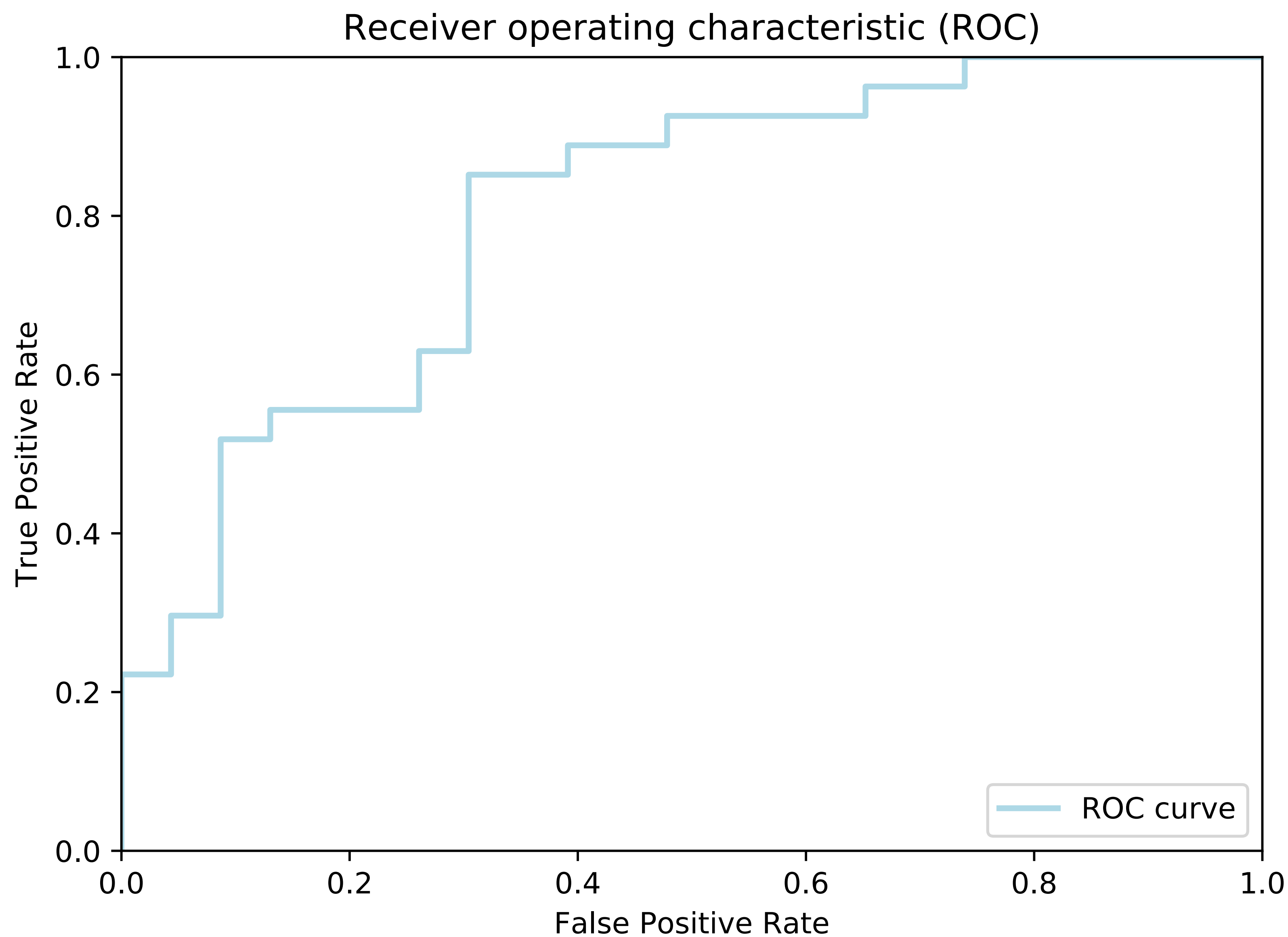
Receiver Operating Characteristic curve (ROC curve)

- 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 [**RA**dio **D**irection **A**nd **R**anging] images)

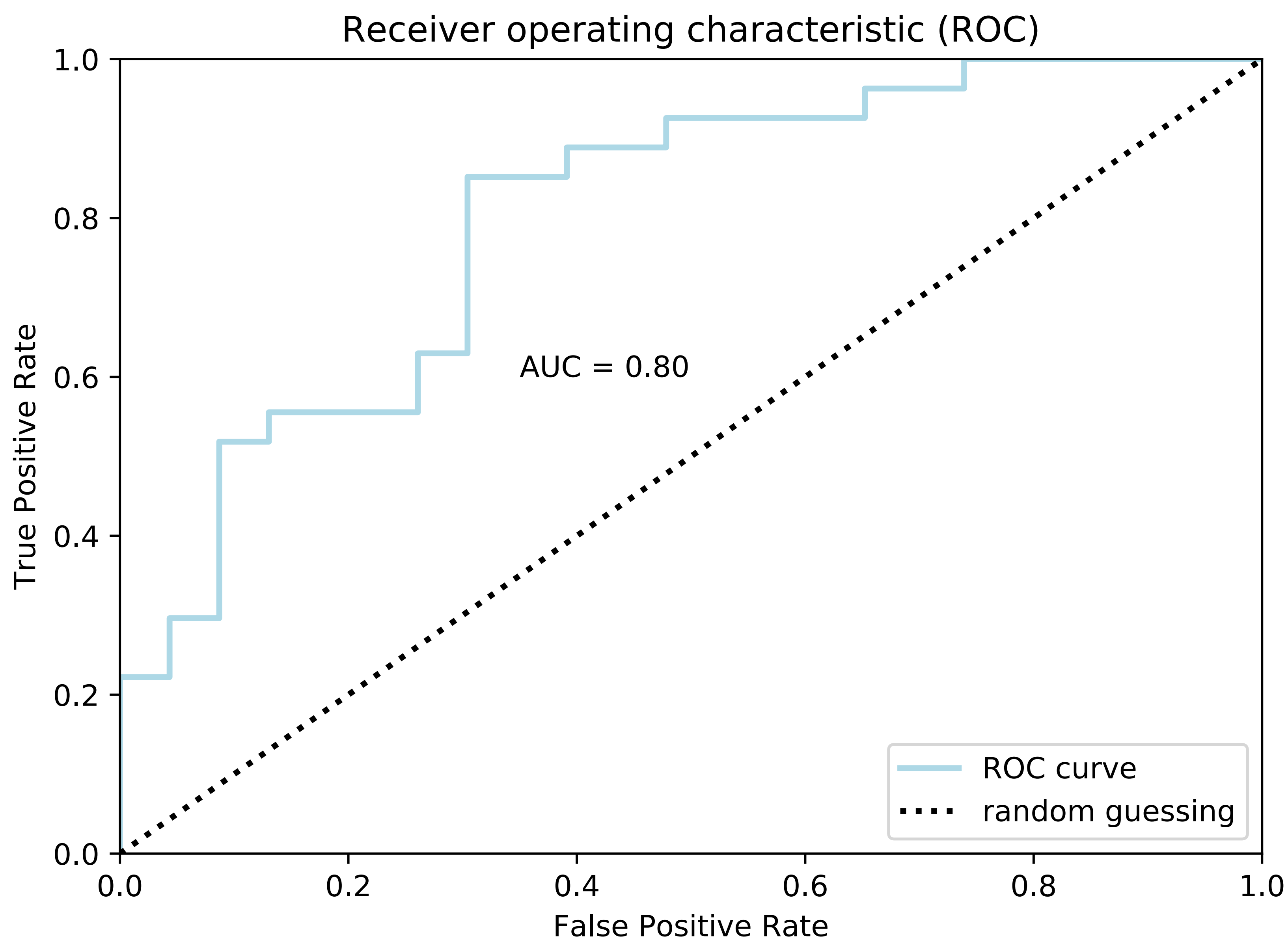


Receiver Operating Characteristic curve (ROC curve)

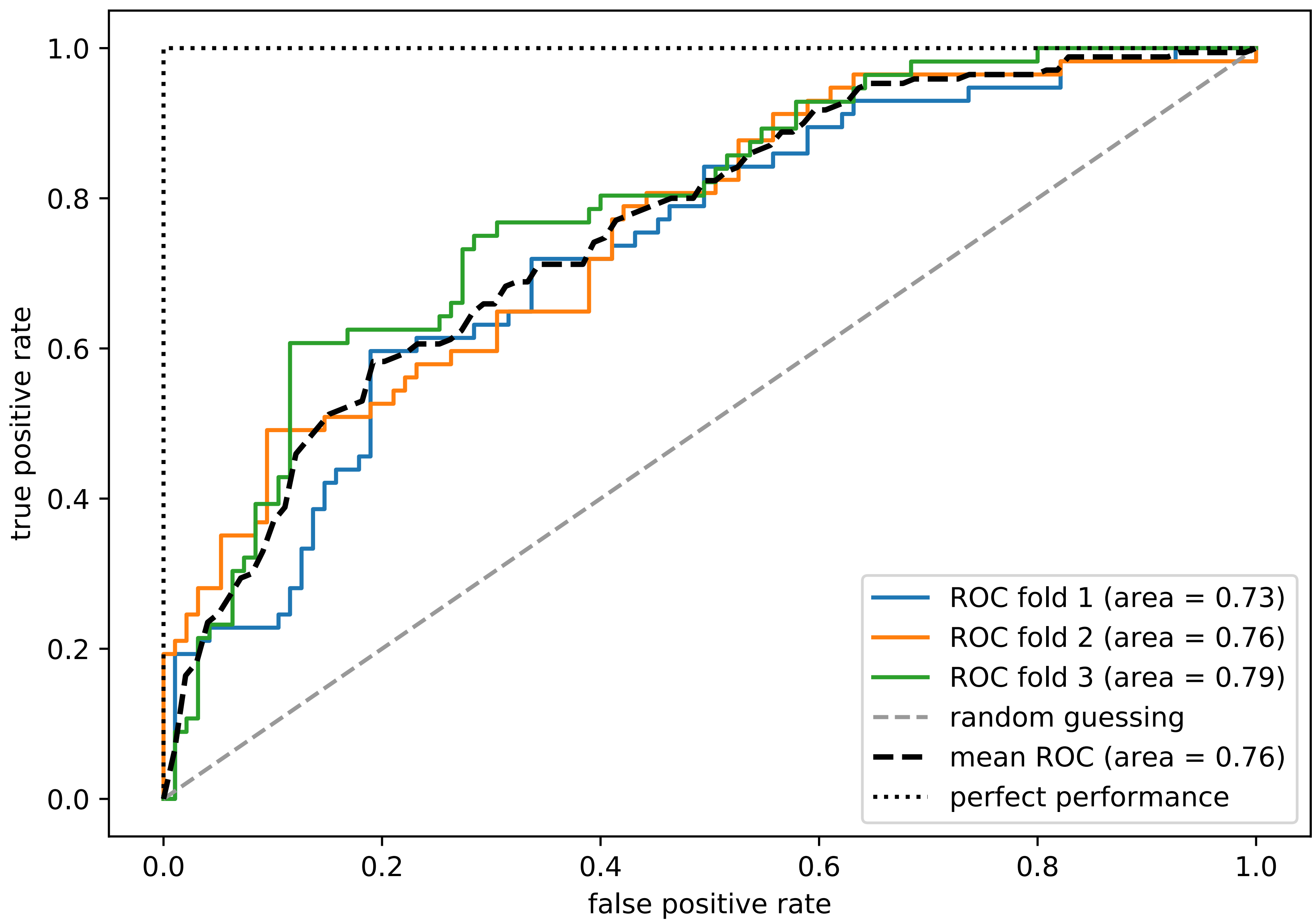
- ?.? = Perfect Prediction
- ?.? = Random Prediction



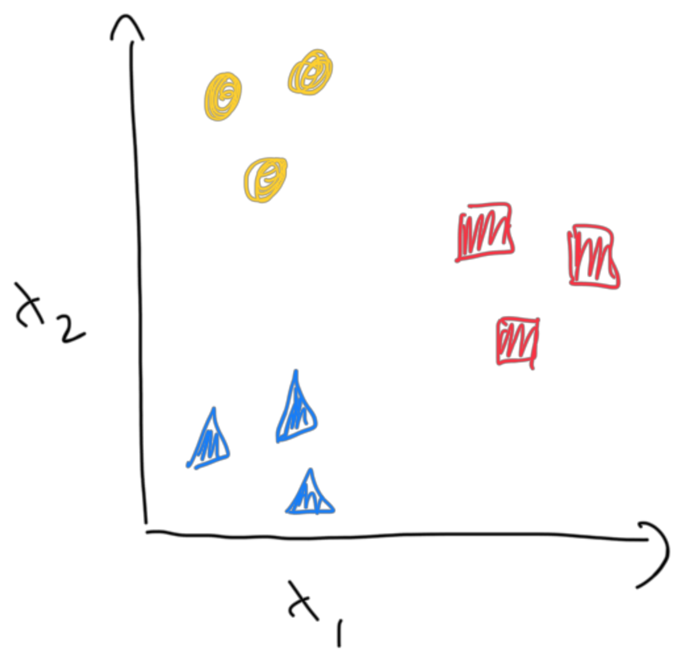
ROC Area Under the Curve (AUC)

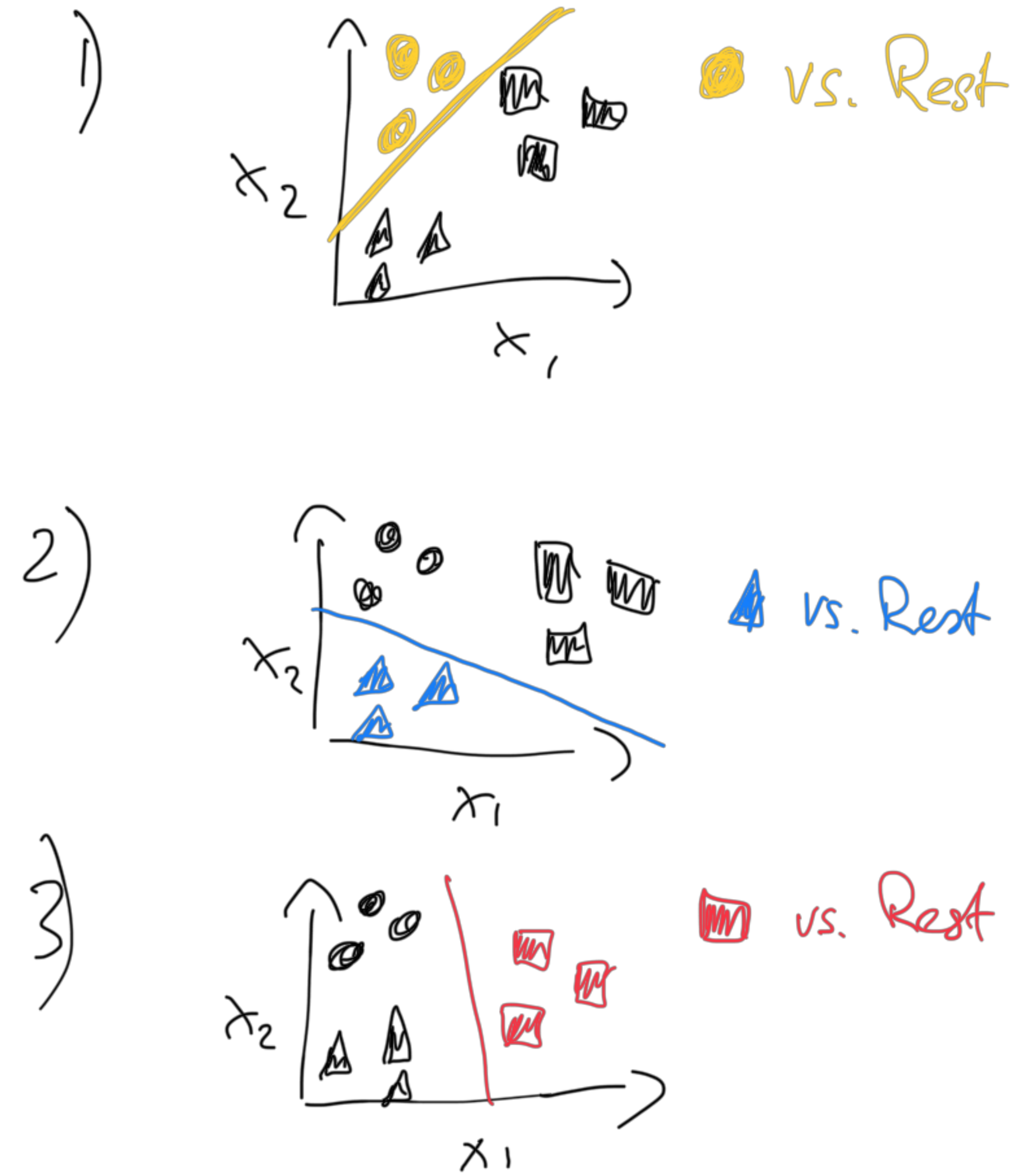


ROC and k-Fold Cross-Validation



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

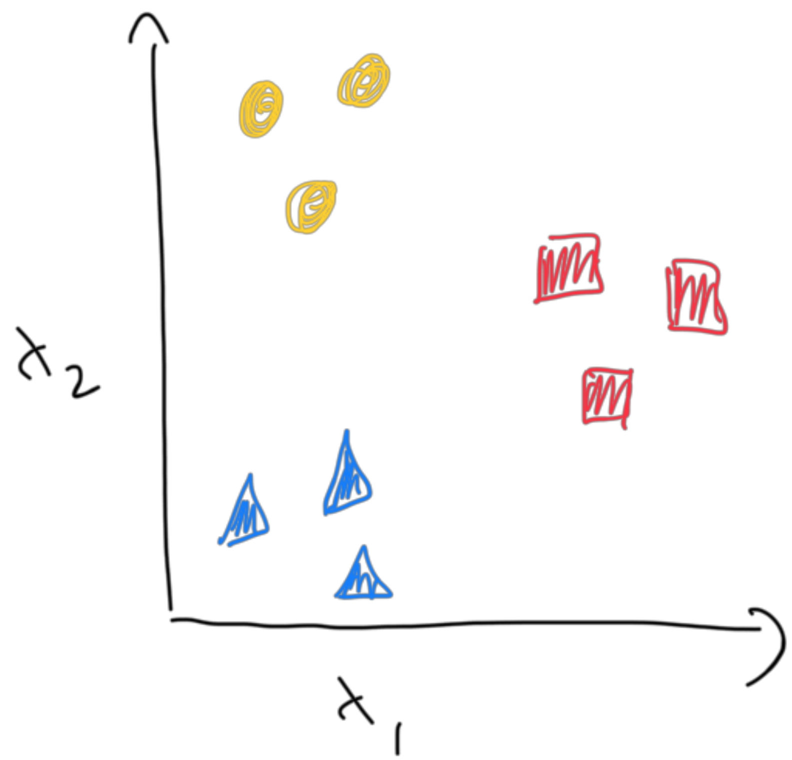


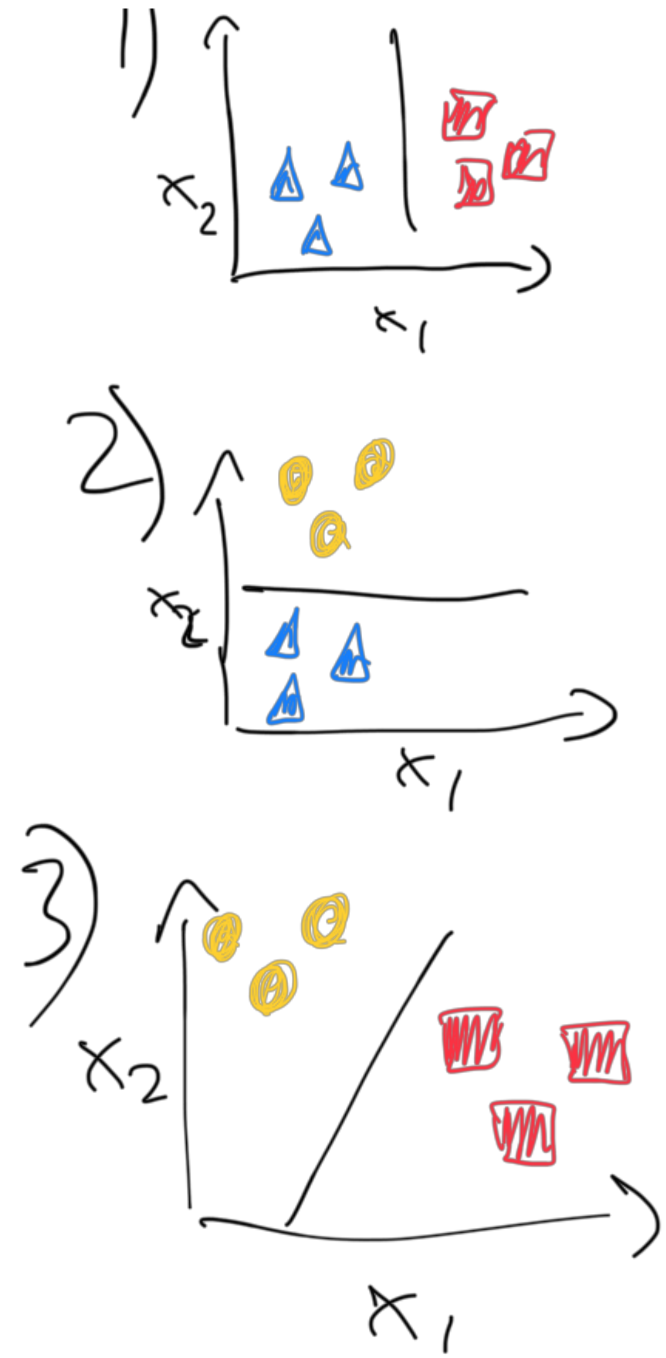


Big O: O(?)

Choose the class with the highest confidence score

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





num_classes x (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

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

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

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.

Note that macro-averaged recall is essentially the same as balanced accuracy

Dealing with Class Imbalance

User Guide

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