Lecture 12

Model Evaluation 5:
Performance Metrics

STAT 479: Machine Learning, Fall 2019

Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat479-fs2019/

Some Announcements

The Undergraduate Statistics Club + the Data Science Club present...
An Interview Skills Workshop
By Amy Yang from SuccessWorks

When: Tuesday November 12 from 6-7pm

Where: Service Memorial Institute Room 133

Why: To learn how to nail that interview!

The Undergraduate Statistics Club + the Data Science Club present...
An Alumni Panel
With UW-Madison alumni focused in Technology, Data, and Analytics

When: Thursday November 21 from 6:30-7:30pm

Where: Genetics-Biotechnology Center Building Room 1441

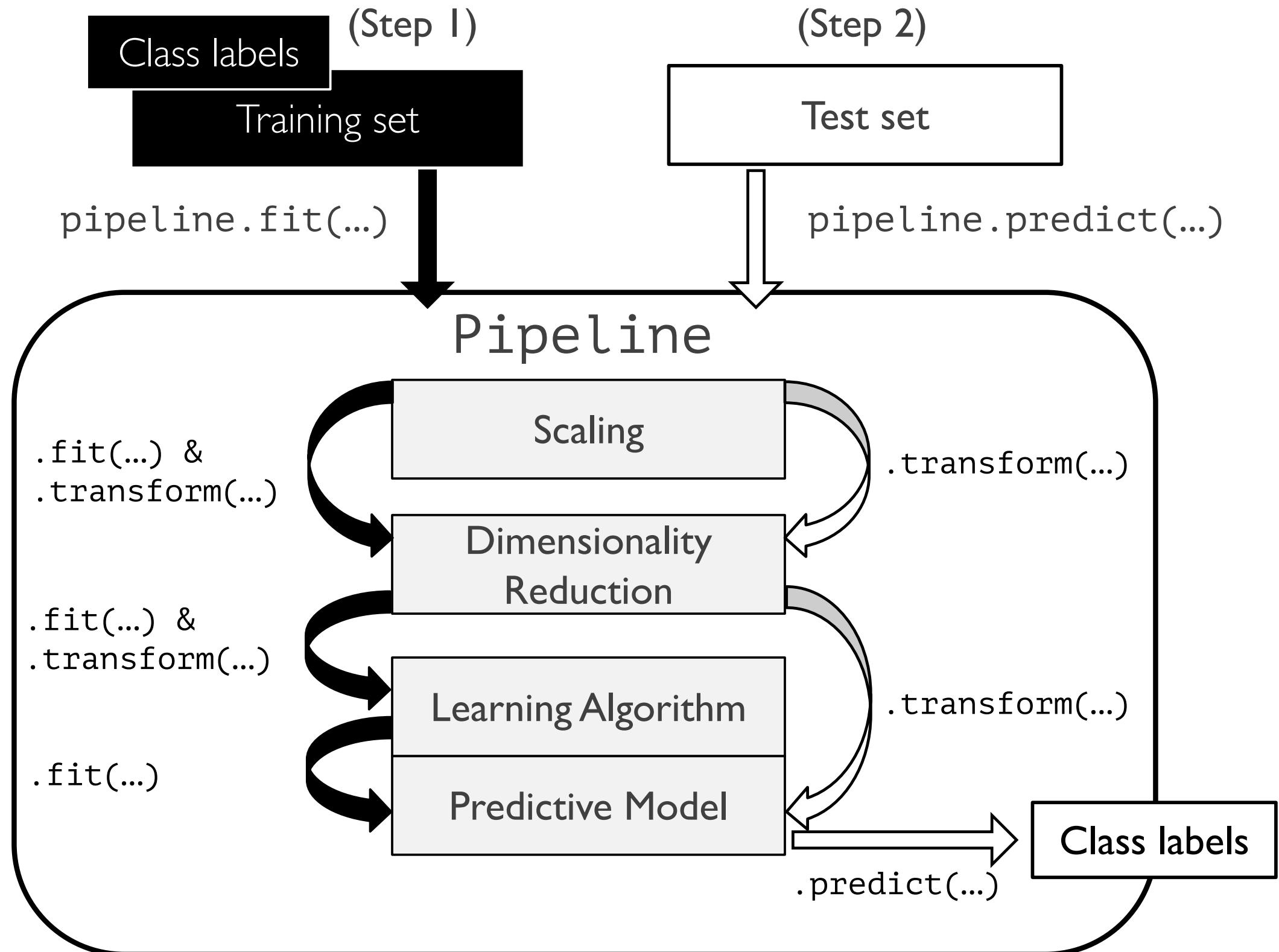**Why**: To get advice from professionals in the field!

Lecture 12

Model Evaluation 5:
Performance Metrics
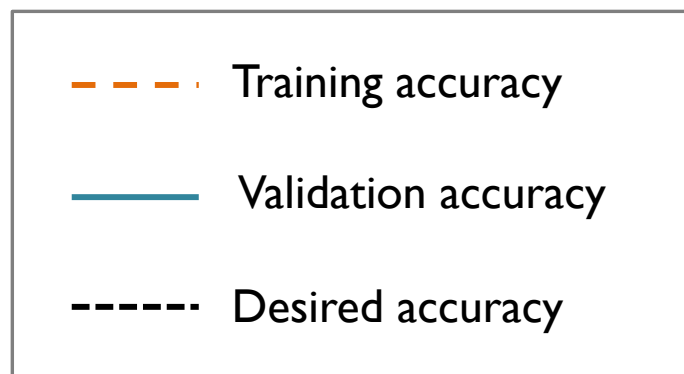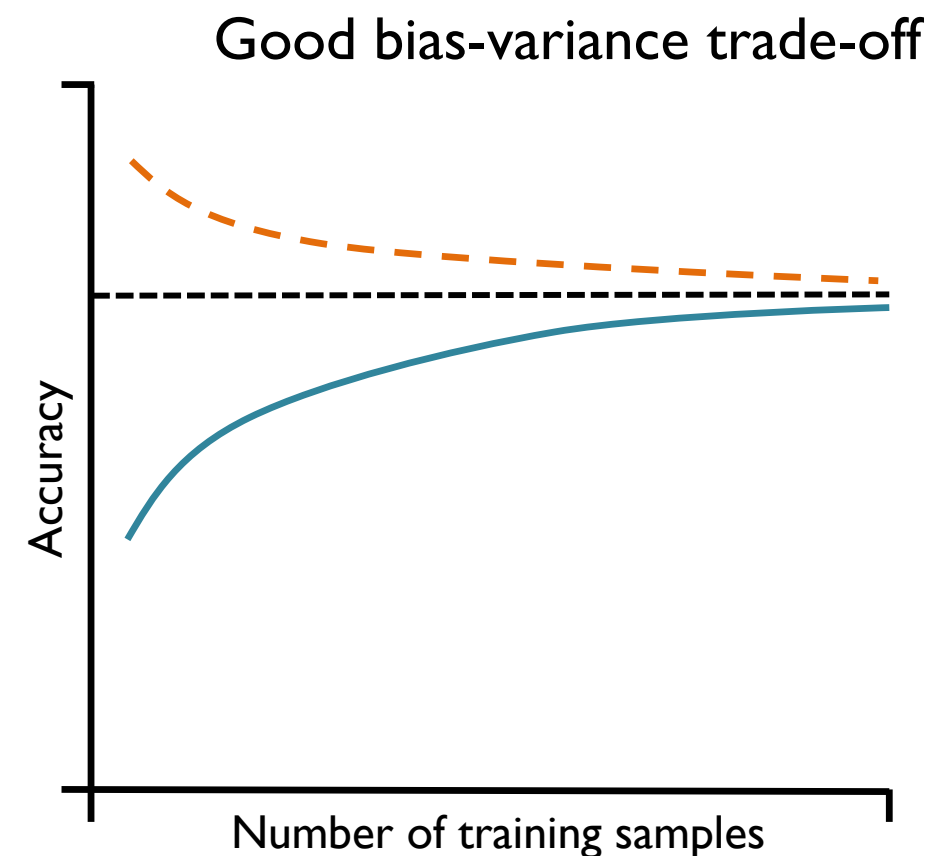
STAT 479: Machine Learning, Fall 2019
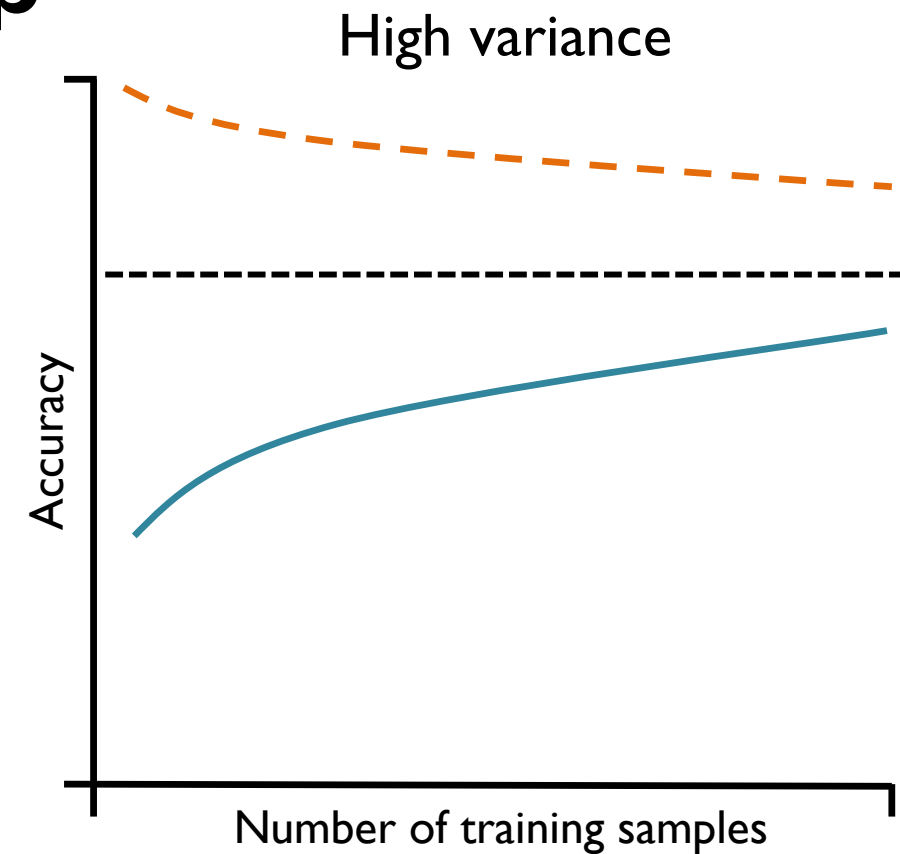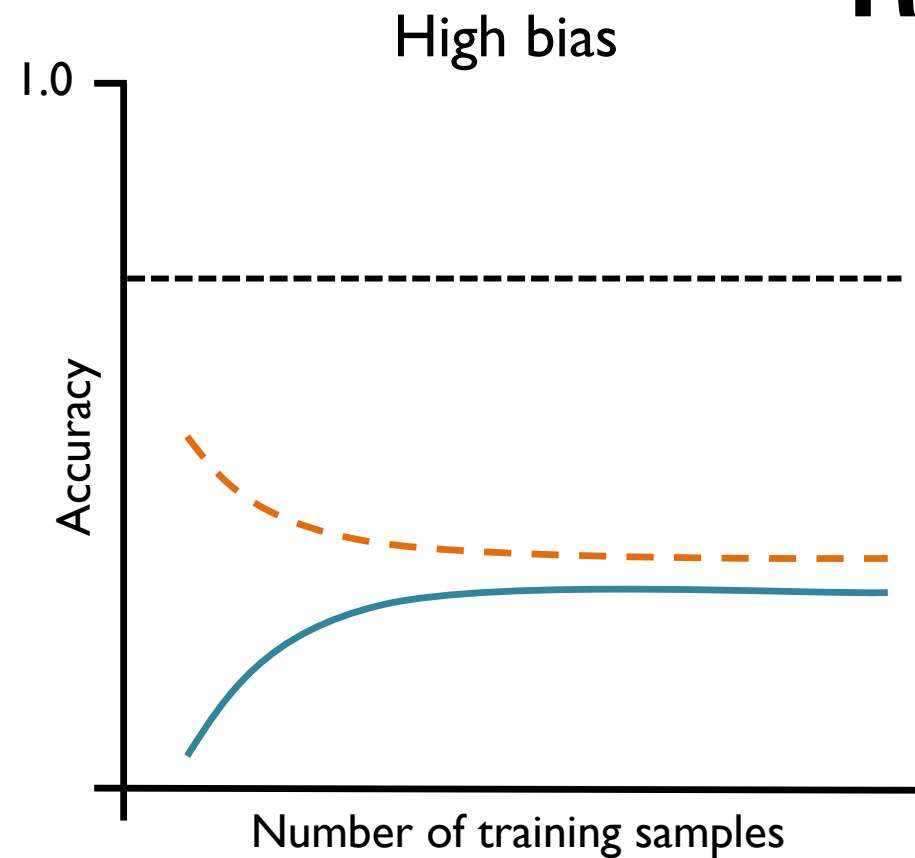
Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat479-fs2019/

Recap

Class labels

Training set

Test set

`pipeline.fit(…)`

`pipeline.predict(…)`

Pipeline

`.fit(…) &`
`.transform(…)`

Scaling

`.transform(…)`

`.fit(…) &`
`.transform(…)`

Dimensionality Reduction

`.fit(…)`

Learning Algorithm

`.transform(…)`

Predictive Model

Class labels

`.predict(…)`

Source: *Python Machine Learning, 2nd Edition*

Recap

High bias

High variance

Good bias-variance trade-off

- - - - - Training accuracy
--------- Validation accuracy
- - - - - Desired accuracy

Source: *Python Machine Learning, 2nd Edition*

2x2 Confusion Matrix

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

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

Confusion Matrix for Multi-Class Settings

Predicted Labels

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

True Labels

Confusions matrices are traditionally for binary class problems but we can be readily generalized it to multi-class settings

False Positive Rate and False Negative Rate

$$\text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 1 - \text{FNR}$$
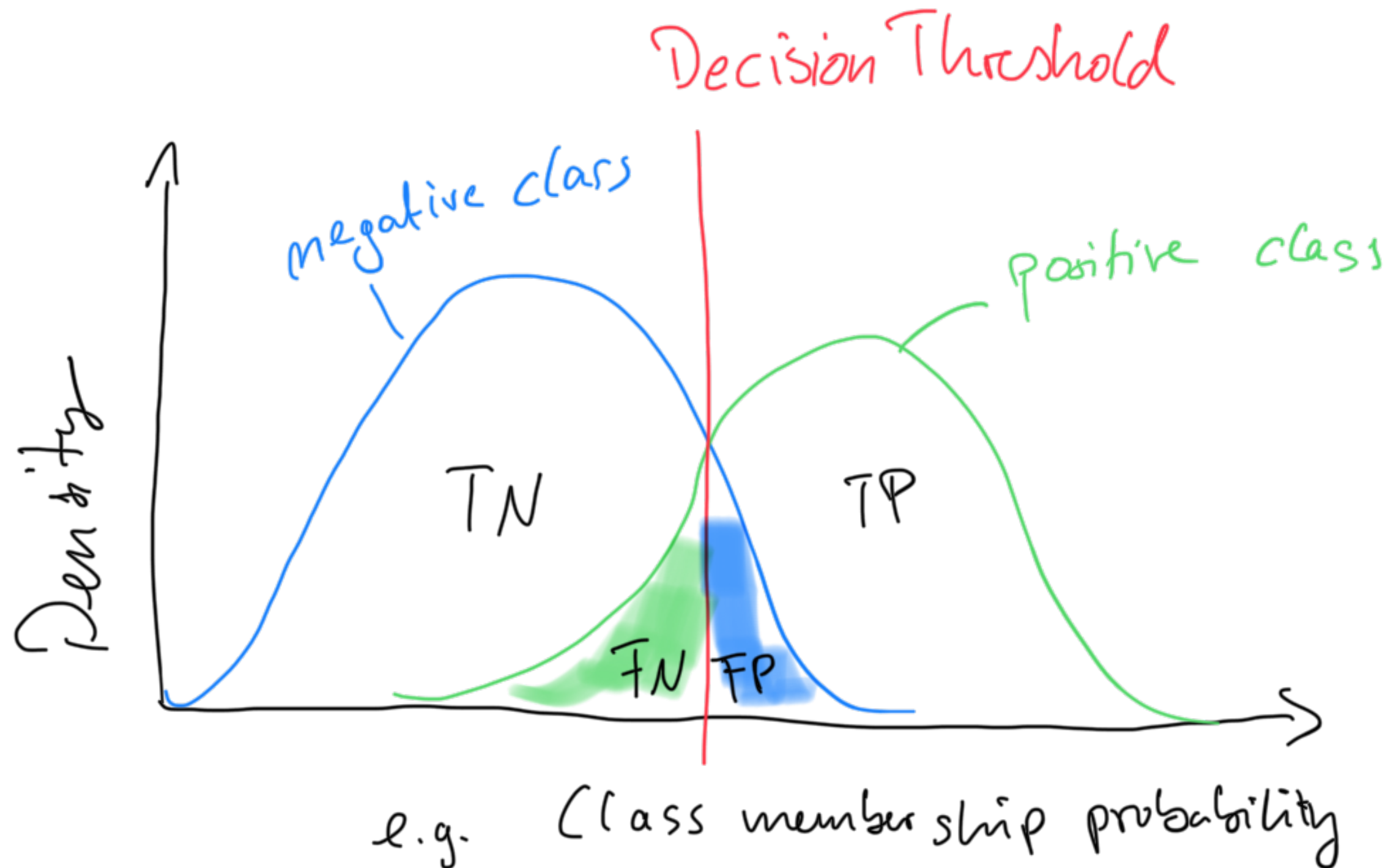
$$\text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{\text{FP}}{\text{FP} + \text{TN}} = 1 - \text{TNR}$$

$$\text{FNR} = \frac{\text{FN}}{\text{P}} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 1 - \text{TPR}$$

$$\text{TNR} = \frac{\text{TN}}{\text{N}} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 1 - \text{FPR}$$

- 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
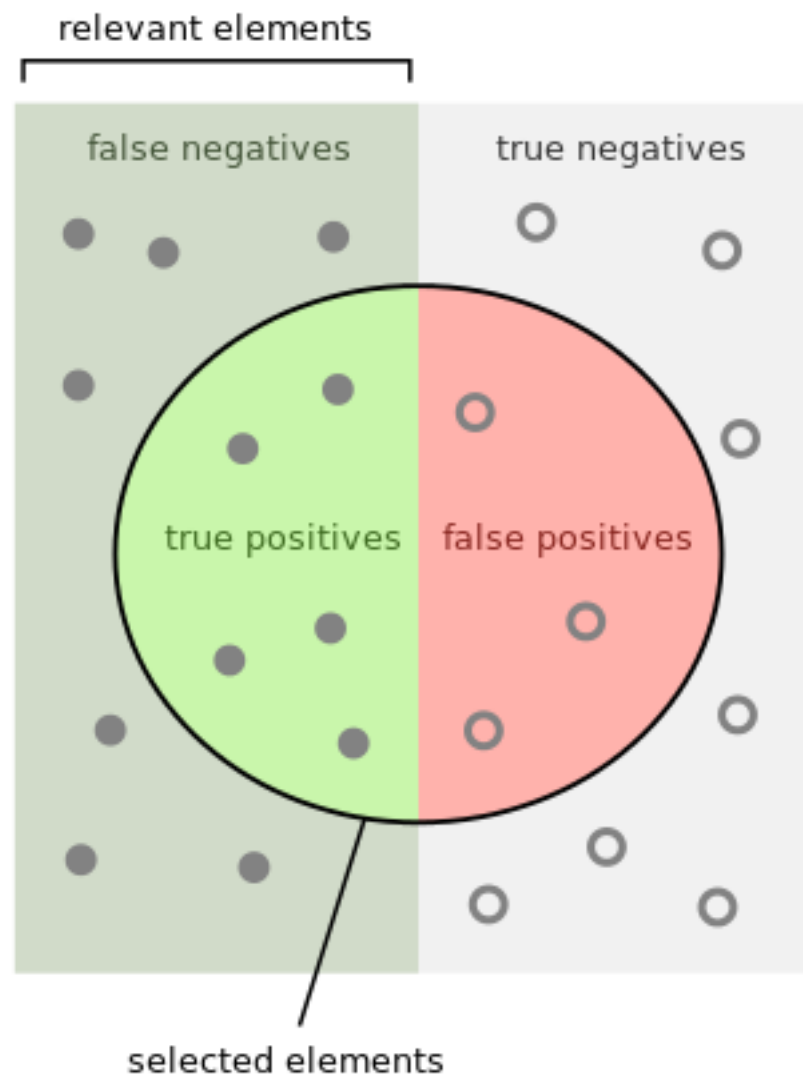
Precision, Recall, and F1 Score

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

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

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

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

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 (SEN) measures the recovery rate of the Positives and complimentary, *Specificity (SPC)* 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.

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

[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 (APC) Accuracy

Predicted Labels

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

True Labels

Predicted Labels

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

True Labels

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

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

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

Balanced Accuracy / Average Per-Class Accuracy

Predicted Labels

	Class 0	Neg Class
Class 0	3	0
Neg Class	7	80

True Labels

Predicted Labels

	Class 1	Neg Class
Class 1	50	19
Neg Class	0	21

True Labels

Predicted Labels

	Class 2	Neg Class
Class 2	18	0
Neg Class	12	60

True Labels

Predicted Labels

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

True Labels

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

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"
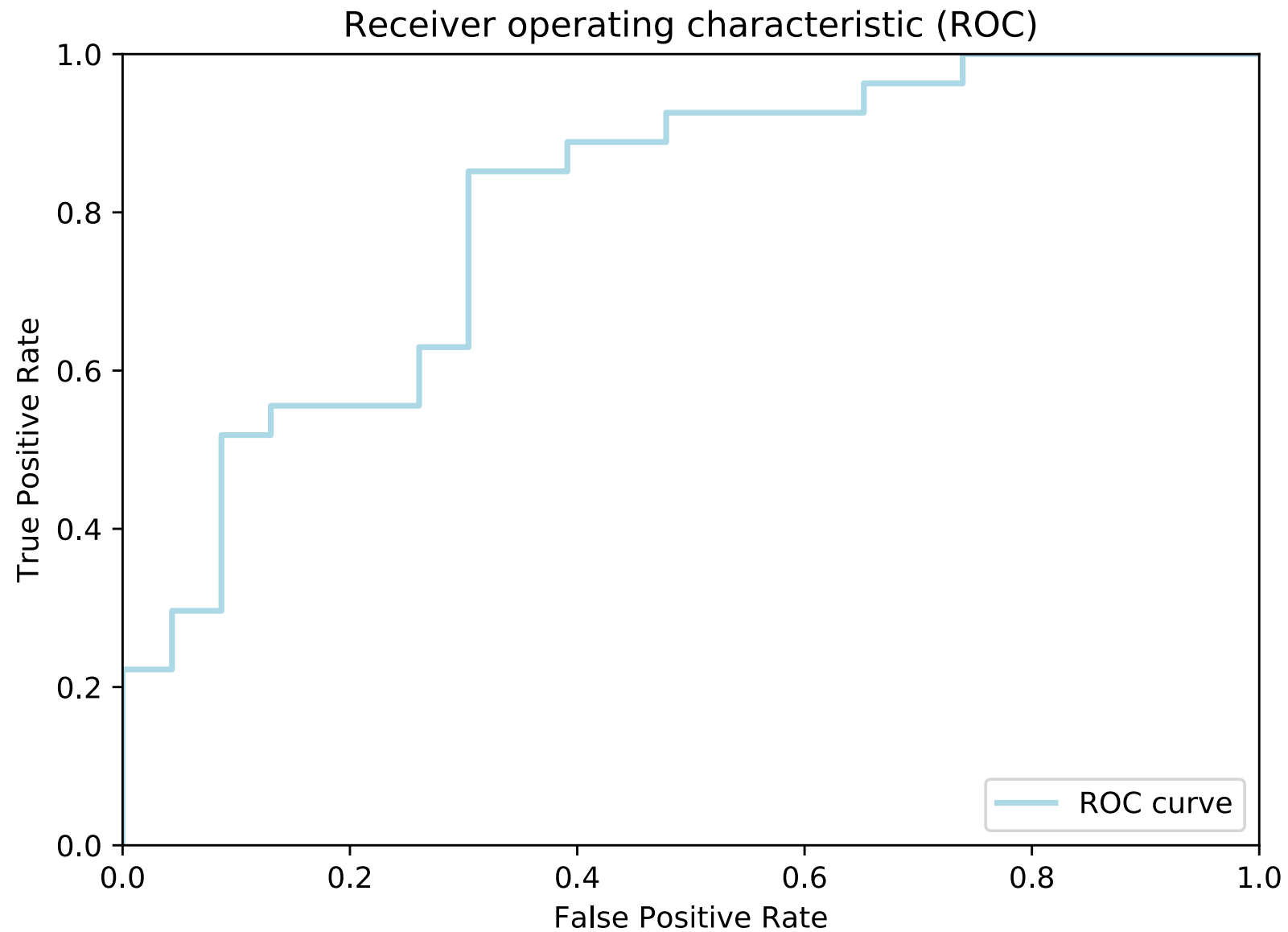 (analysis of radar [**RA**dio **D**irection **A**nd **R**anging] images)
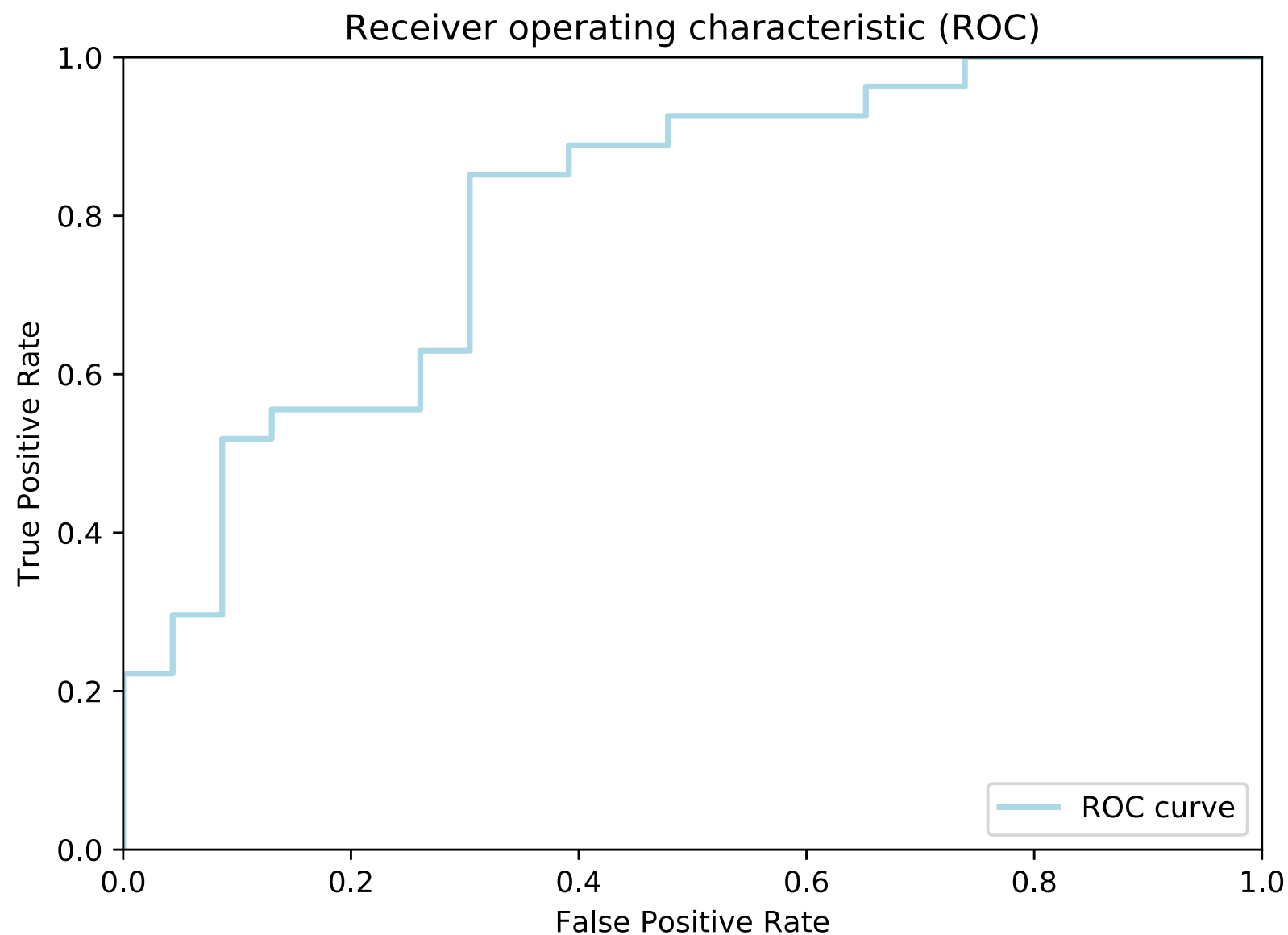
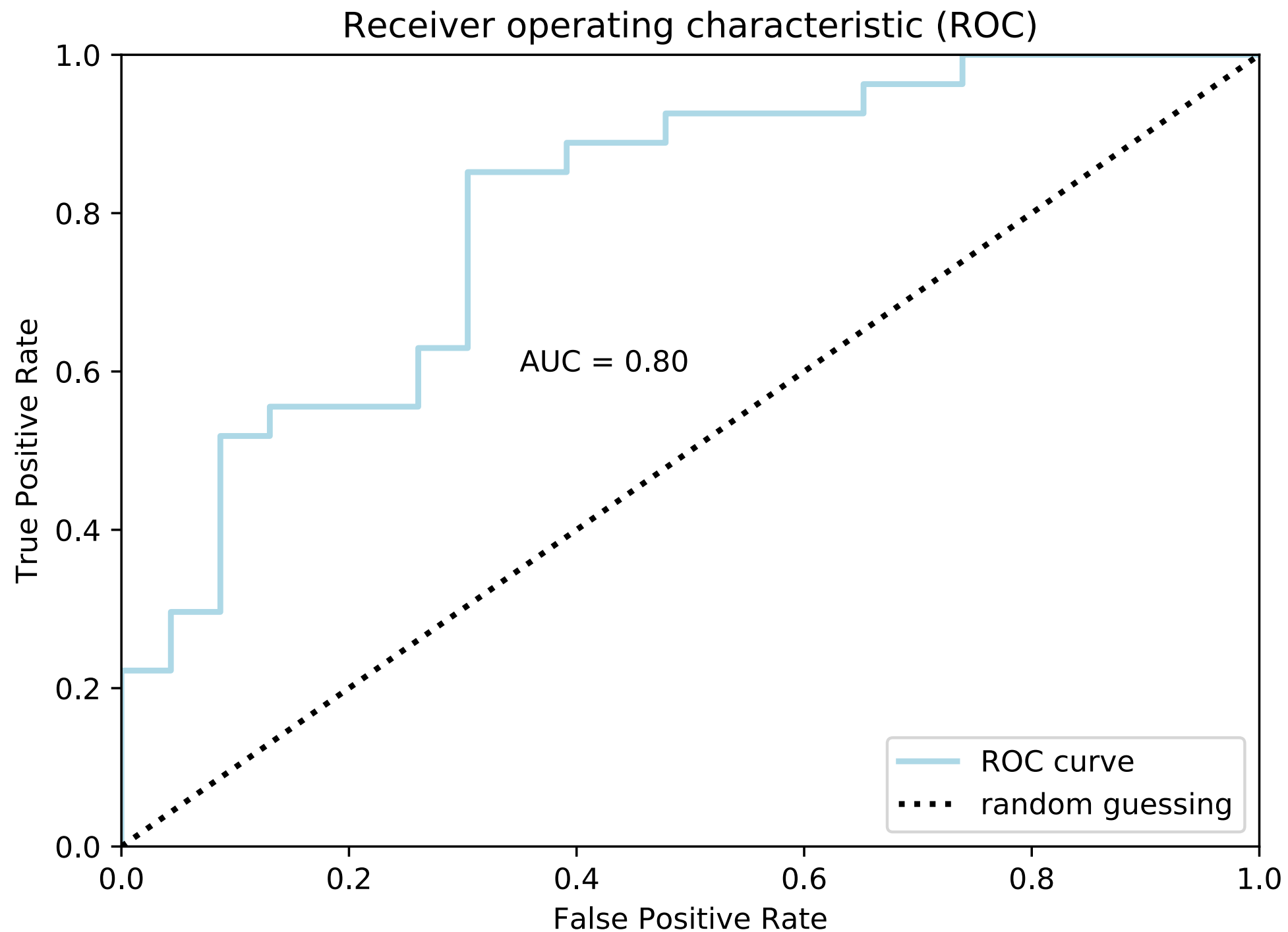# RECAP: False Positive Rate and False Negative Rate
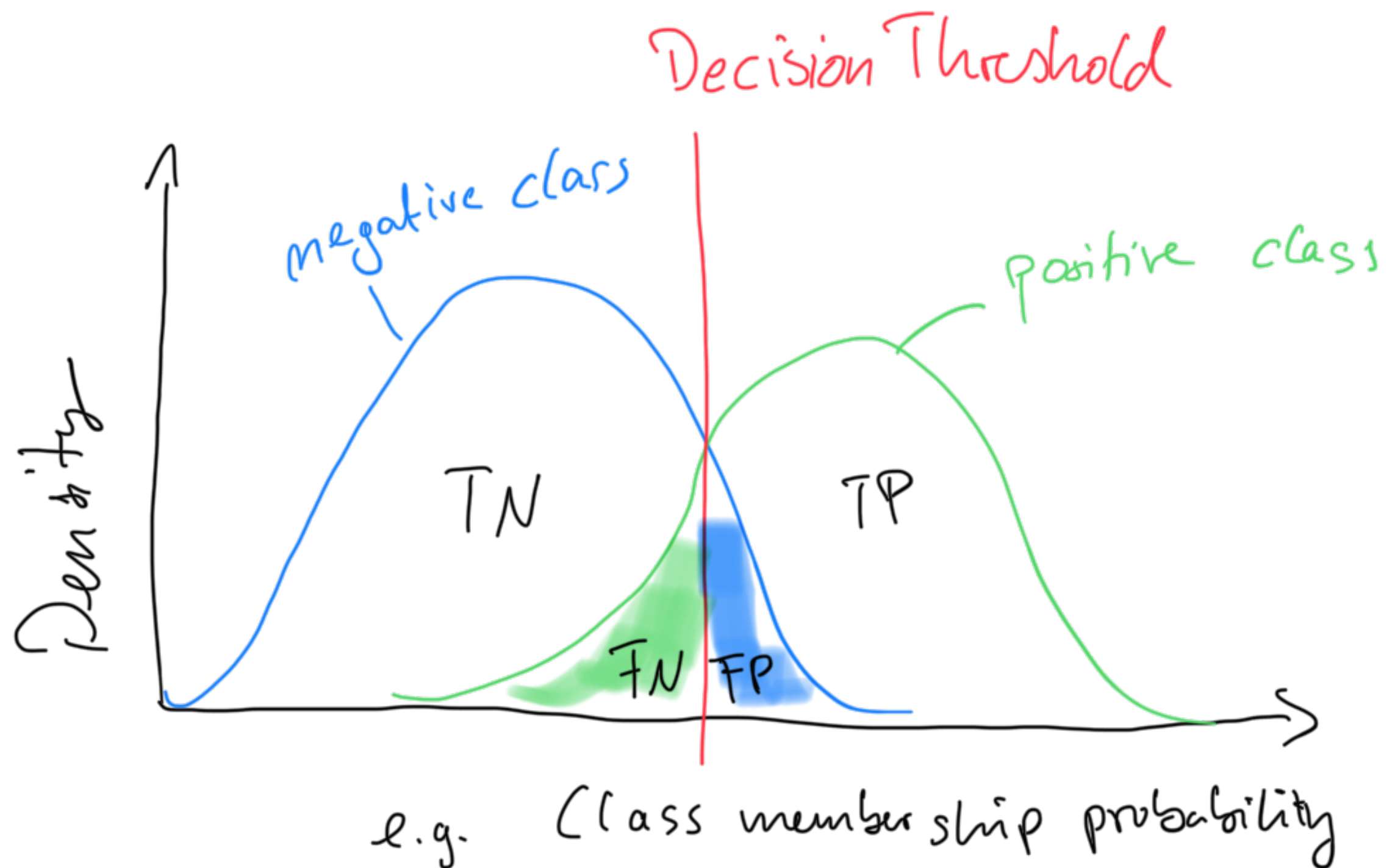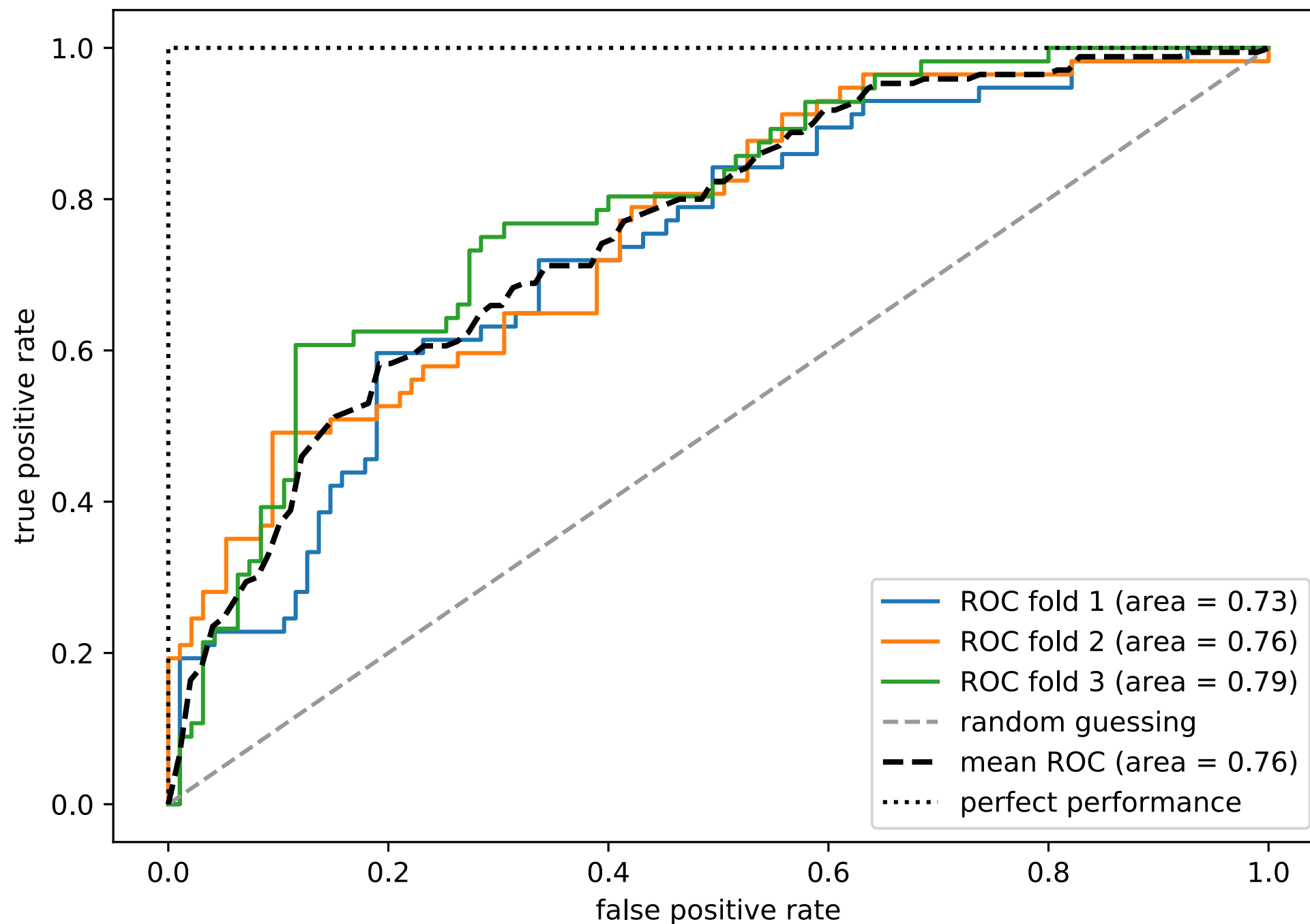
Receiver Operating Characteristic curve (ROC curve)

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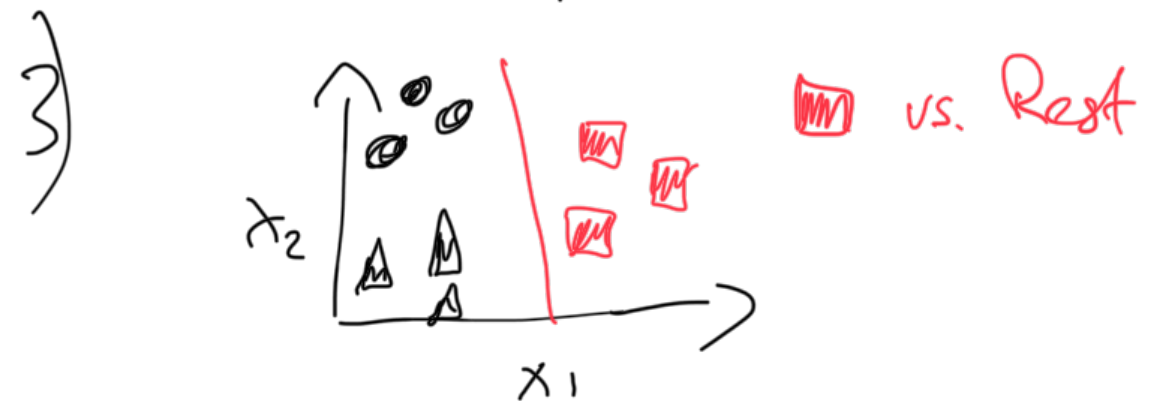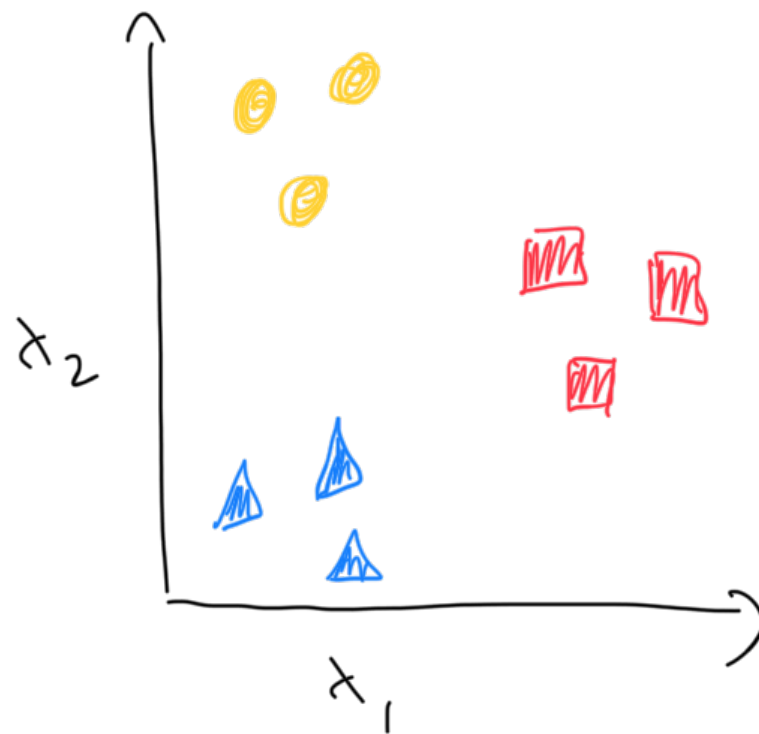- ?.? = Random Prediction

Receiver operating characteristic (ROC)

ROC Area Under the Curve (AUC)
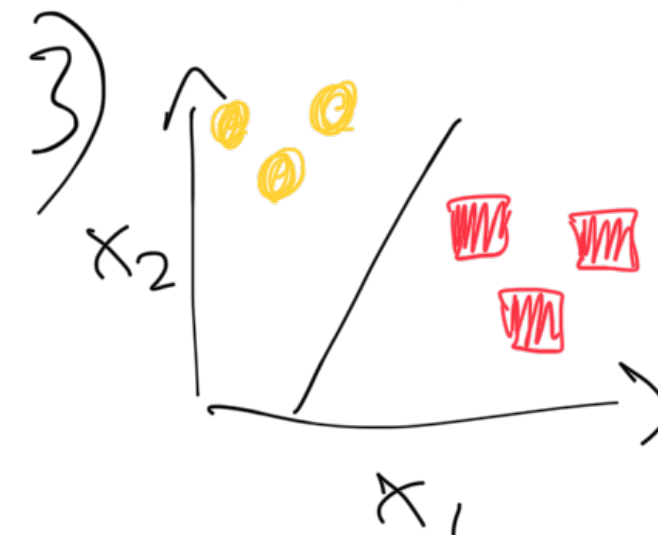
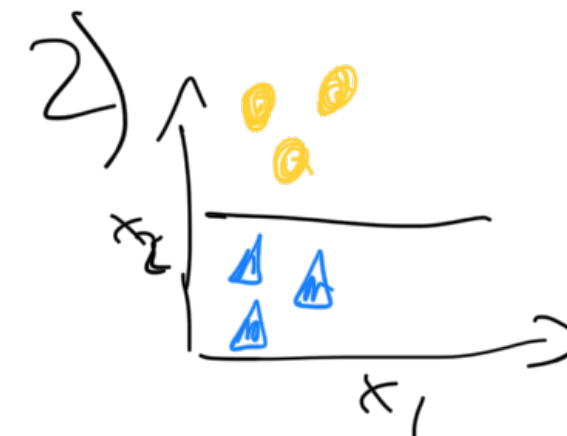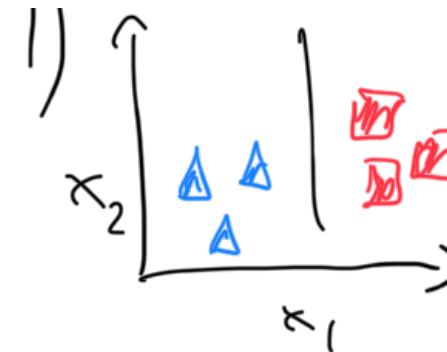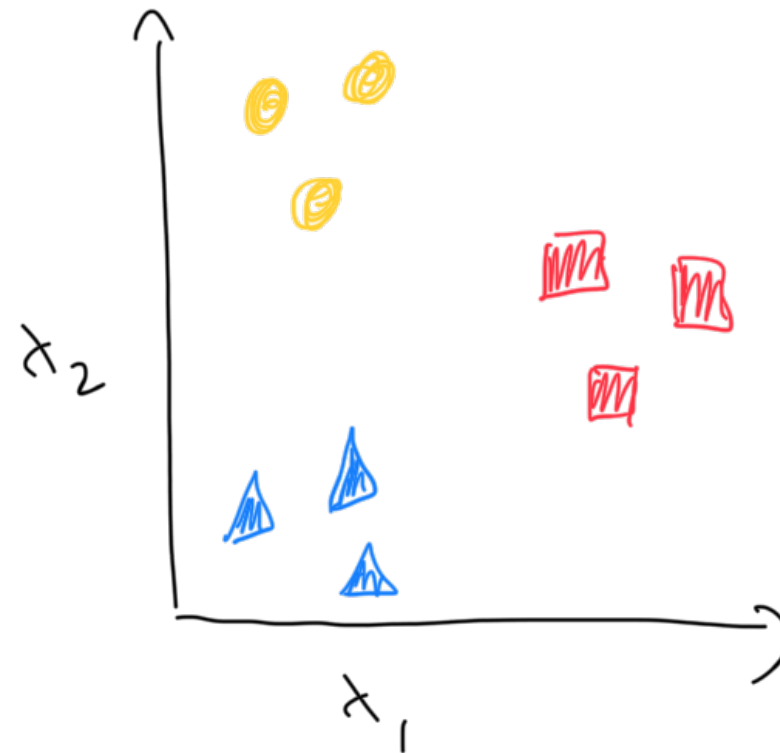# ROC and k-Fold Cross-Validation

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

Then, 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.

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

Reading Assignment

No lecture notes, instead see

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

  (code notebook: https://github.com/rasbt/python-machine-learning-book-2nd-edition/blob/master/code/ch06/ch06.ipynb)

Code Examples

https://github.com/rasbt/stat479-machine-learning-fs19/tree/master/12_eval5-metrics/code