Supervised Learning

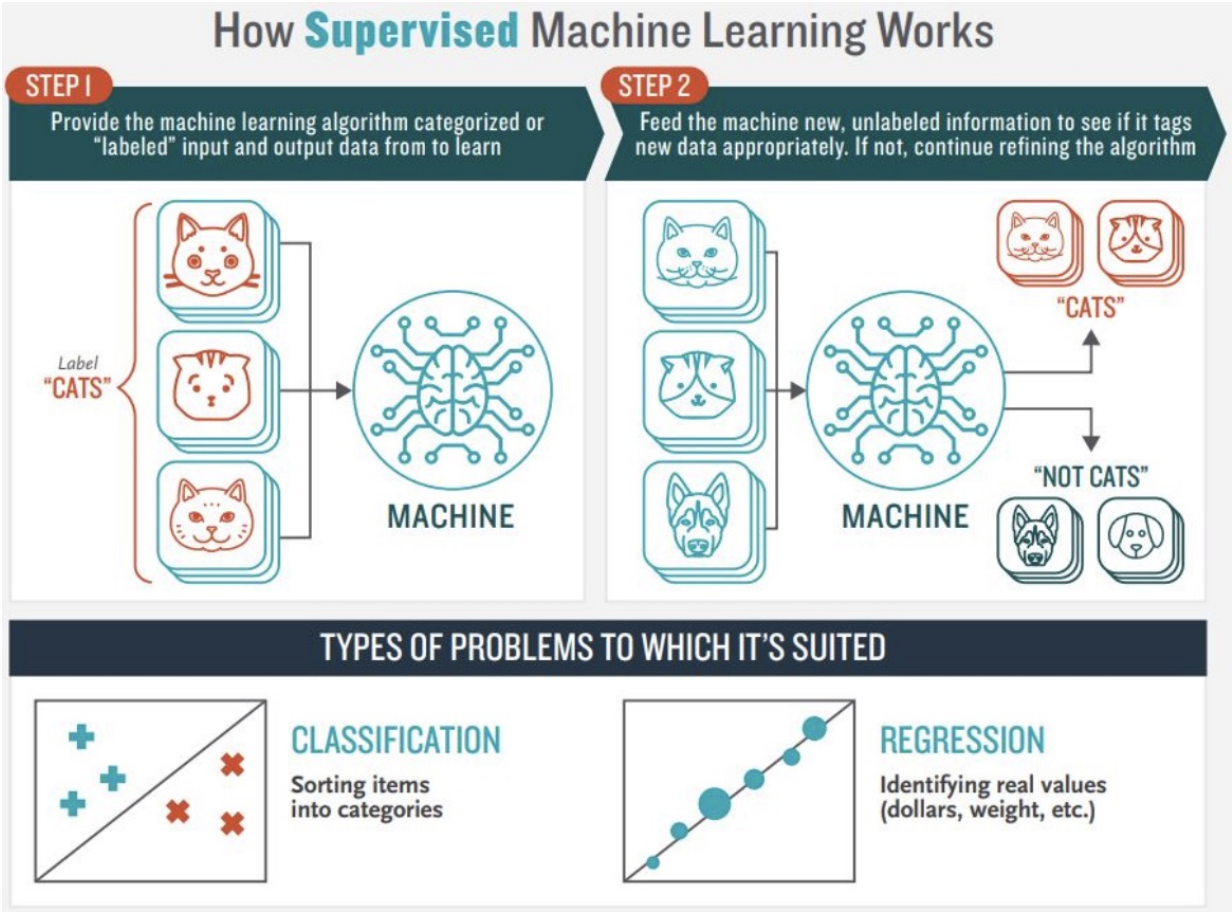
COSC 410: Applied Machine Learning

Spring 2022

Prof. Apthorpe

Outline

- Classification & Regression
- Training, Validation, & Test Sets
- Golden Rule of Supervised ML
- Cross-Validation
- Performance Metrics
  - Regression Error Functions
  - Accuracy & Imbalanced Classes
  - ROC curve & AUC
  - Precision/Recall & F_1 Scores
  - Confusion Matrices



Classification & Regression

Regression: Continuous labels

Classification: Binary or multi-class labels

Dataset

$$X_{m \times n(\times p \ldots)}$$

Examples

Attributes (Features)

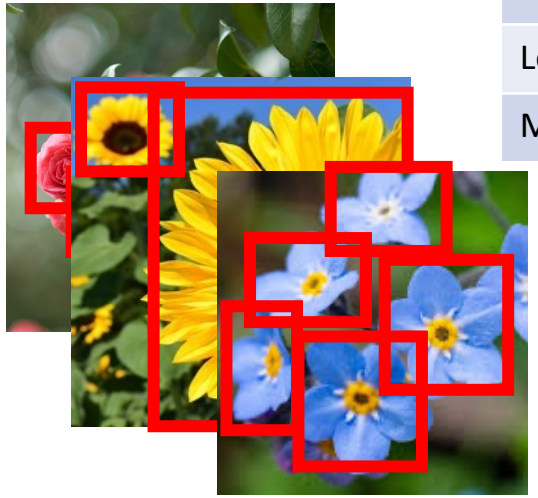
Age	Fam His.	Smokes
43	Y	Y
29	N	N
76	N	N

Labels

$$y_{m(\times k \ldots)}$$

Examples

Cancer Risk
High
Low
Med



Ultimate Goal

Train model $h()$ such that for data X and correct labels y

$$h(X) \approx y$$

"The model's predictions are close* to the real labels"

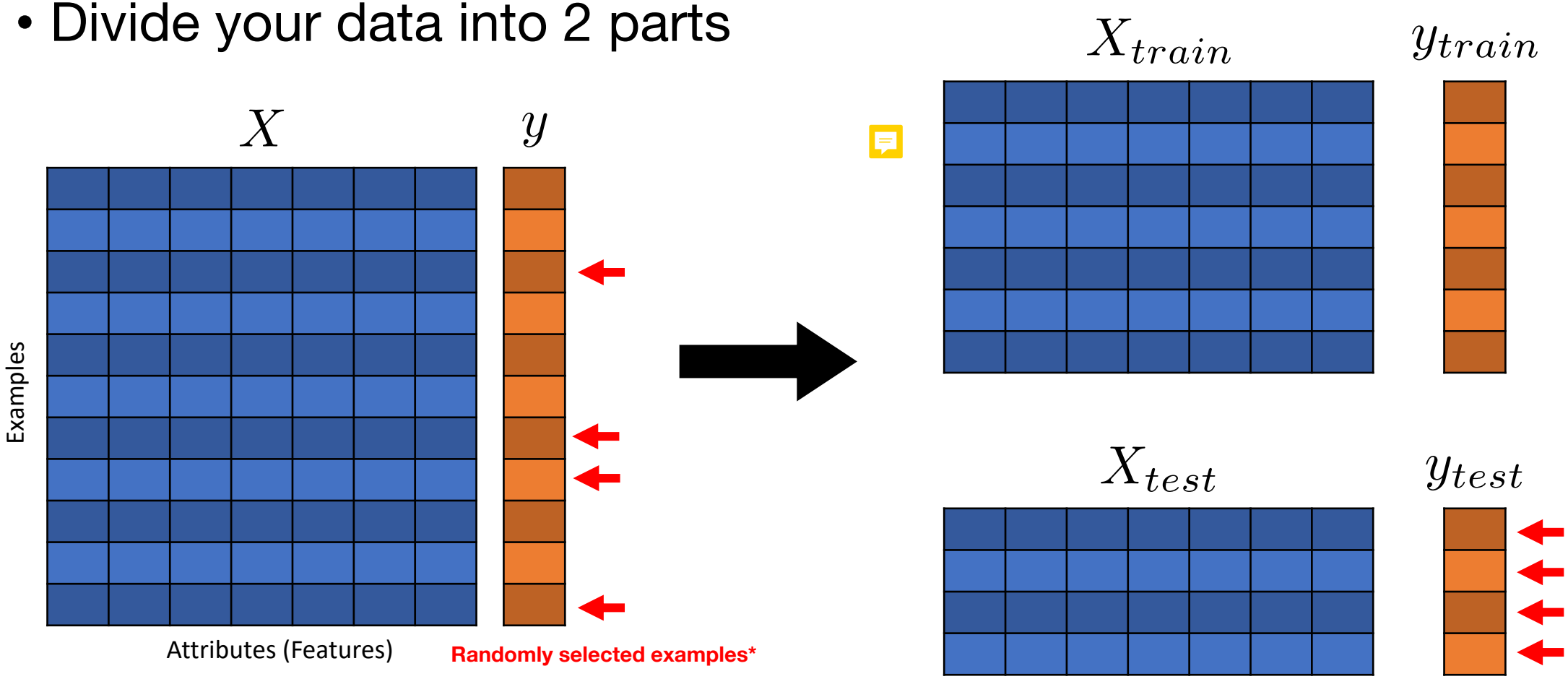
*Need to define "close"

Inherent Problem

- If you don't know the correct labels for new data, *how do you evaluate whether your model **generalizes**?*
  - Why is this important? 
- Solution:
  - Use the data and labels you do have for **both training** and **evaluation**

Training Sets & Test Sets

- Divide your data into 2 parts



Training Sets & Test Sets

- Use the **training set** to train the model $h()$
  - ★ Minimize the **training error** $E(h(X_{train}), y_{train})$
    - Choice of **error (loss) function**
      - Square error, absolute error, hinge loss, cross-entropy loss, …

- Use the **test set** to measure the model's **performance** and ability to **generalize** to new data
  - Measure the **test error** $E(h(X_{test}), y_{test})$

The **Golden Rule** of Supervised ML

Never train on the test set!!!

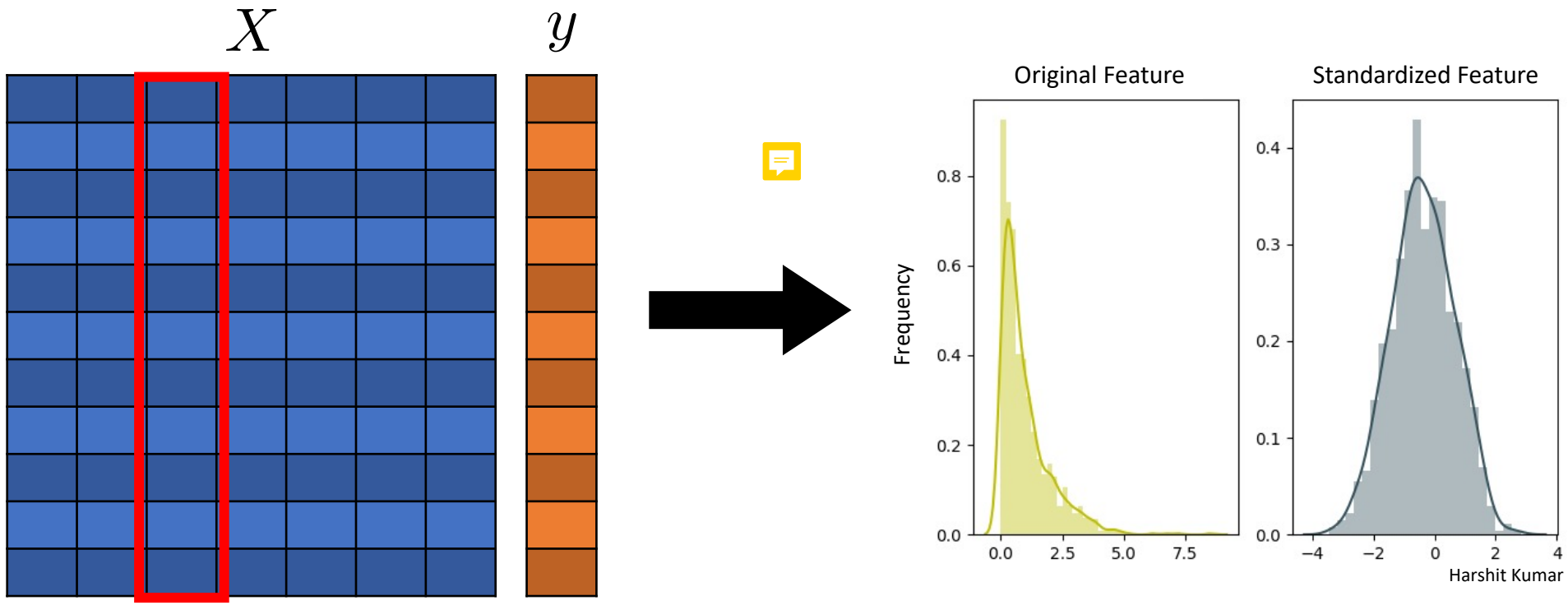
Nothing in your model should be influenced by the examples in the test set. This includes preprocessing steps, training examples, and hyperparameter settings!

The test set is **only** for reporting your final performance

Imagine you don't even have the test set when preparing your model

Common Ways to Break the Golden Rule

- **Preprocessing based on entire dataset**
  - E.g. scaling the variance of each feature to a normal distribution



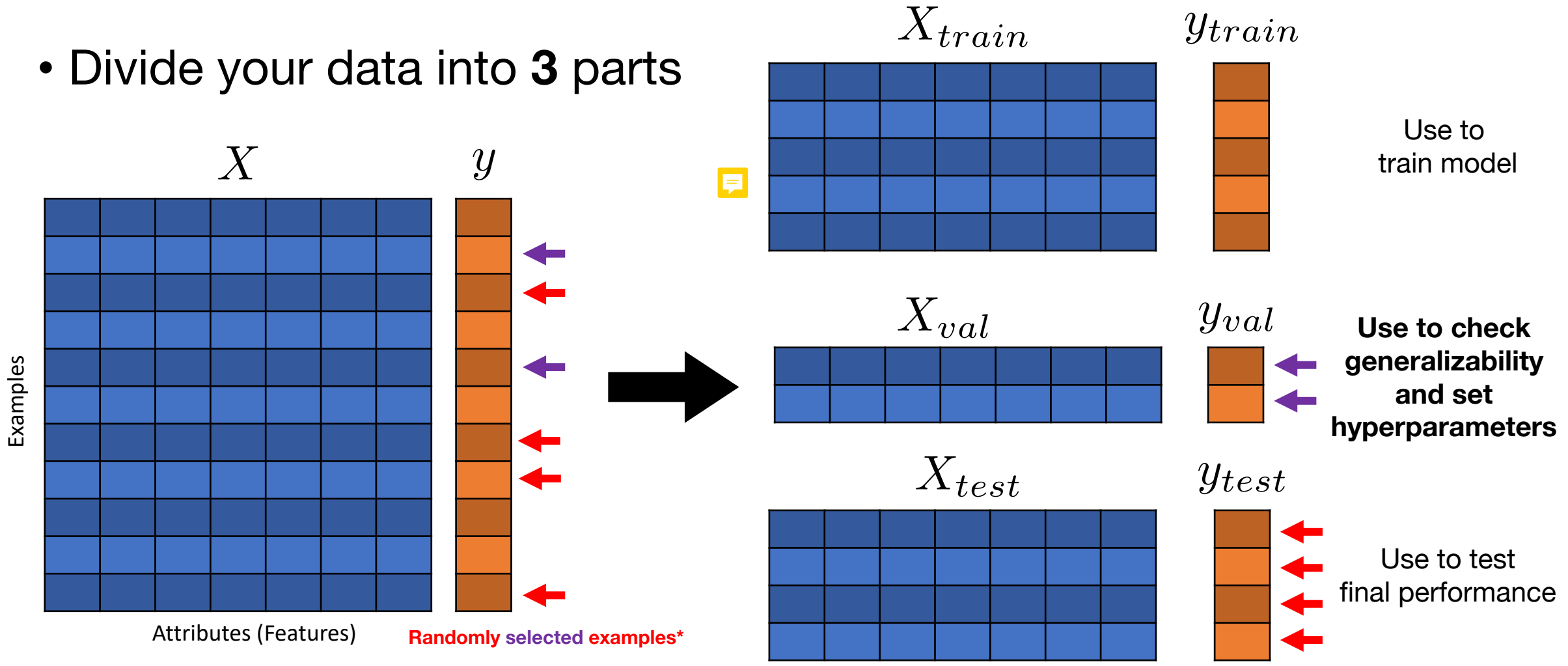
Common Ways to Break the Golden Rule

- **Modifying model based on test error**
  - Train on the training set (good)
  - Test on the test set (good)
  - Think “that worked poorly, let me modify some parameters…” (BAD)

- Why is this bad?
- Why is this tempting?
  - Without testing on data you didn’t use to train, how do you improve generalization?

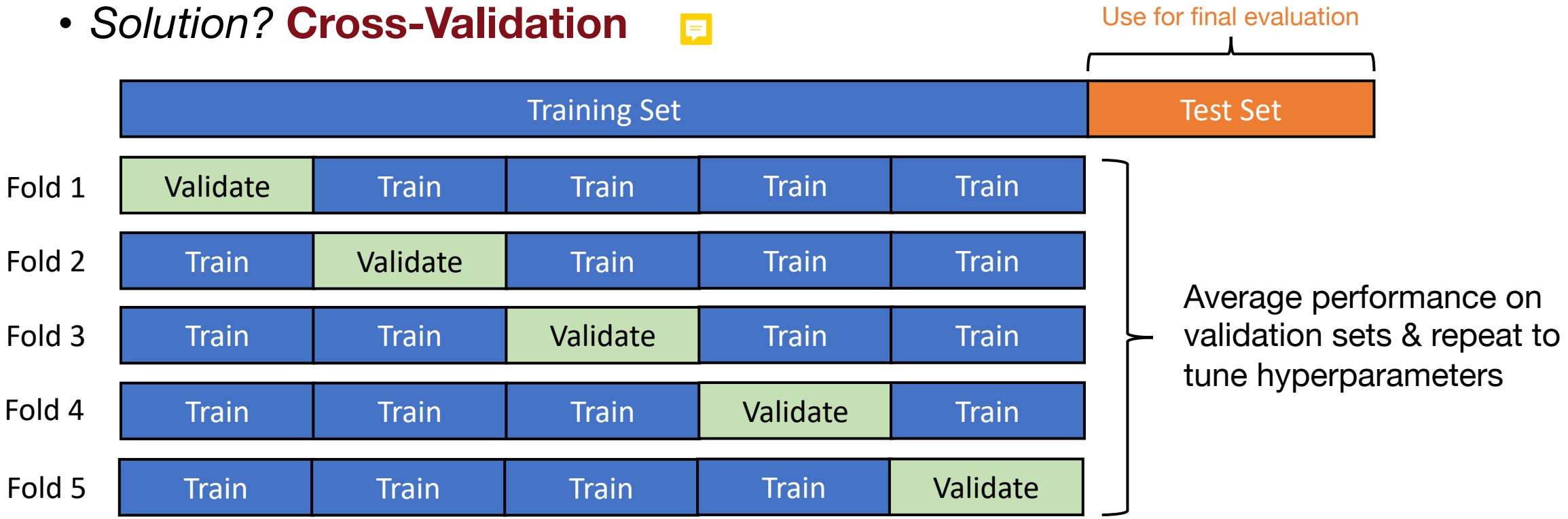
Training / Validation / Test Sets

- Divide your data into **3** parts



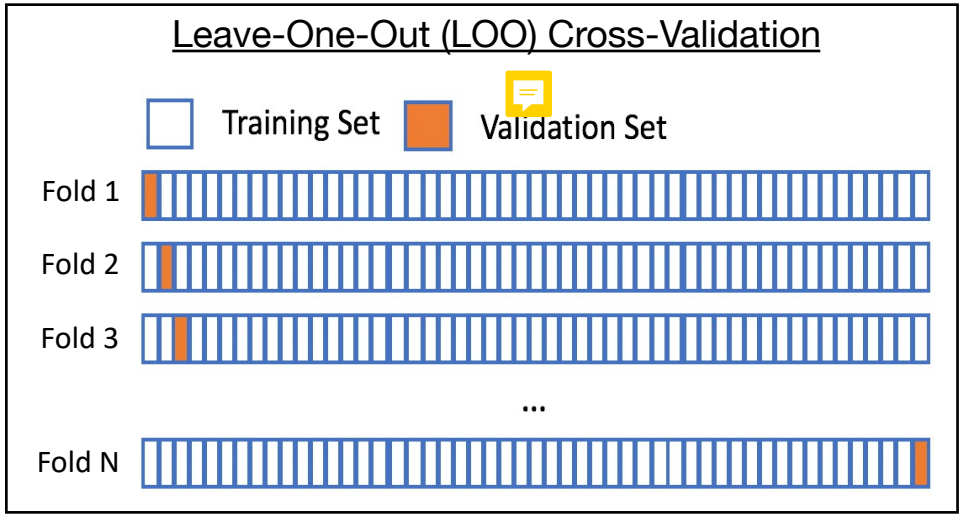
Cross-Validation

- Train / validate / test split leaves much **less data for training**
  - Can reduce final performance (more data usually improves models)
- *Solution?* **Cross-Validation**



Cross-Validation

- How many folds?
  - **More** folds
    - **Better estimation of model performance**
    - Computation intensive
      - Need to train model from scratch for each fold
  - **Fewer** folds *(5 or 10 standard)*
    - **Worse estimation of model performance**
    - Fewer computations (practical for deep learning)



Performance Metrics

How to Measure ML Success

Performance Metrics

- **Performance metric** functions
  - Produce **value** indicating the success of the model
  - Compare predicted labels to actual labels

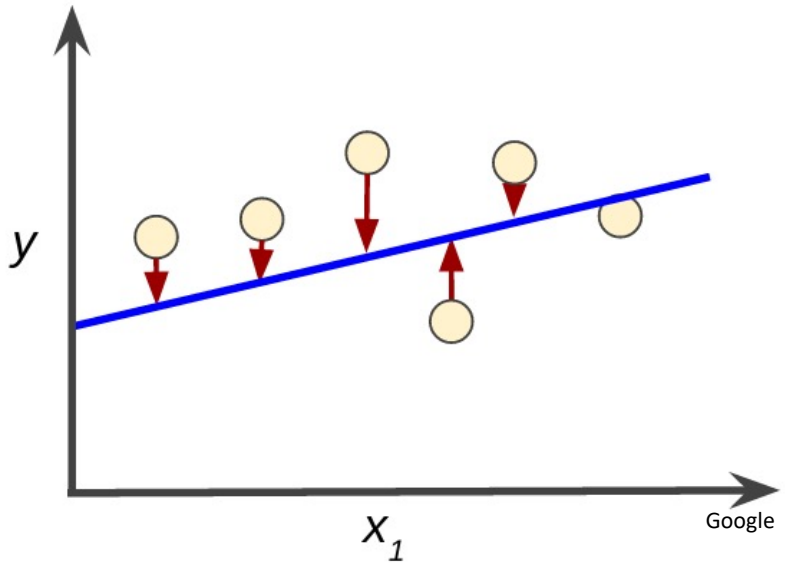
Regression Loss Functions

- Mean square error

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

- Mean absolute error

$$MAE = \frac{\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid}{n}$$

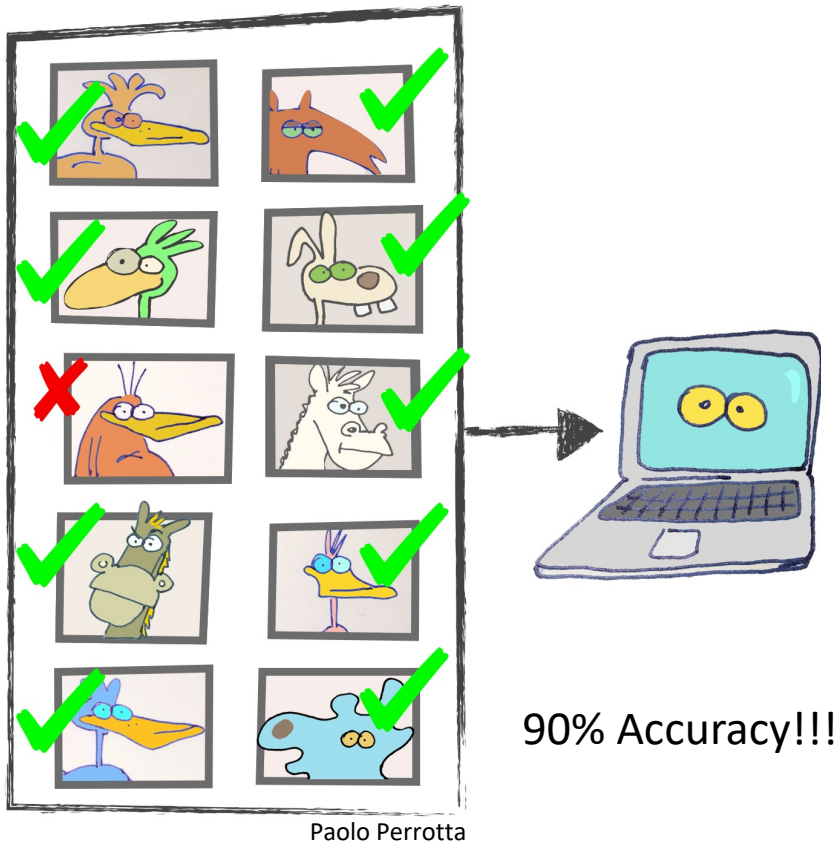


Classification **Accuracy**

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Number of Total Predictions}}$$

- Accuracy is intuitive, but deceptive if classes are unbalanced

Trivial Solution: Always guess "Not a platypus"

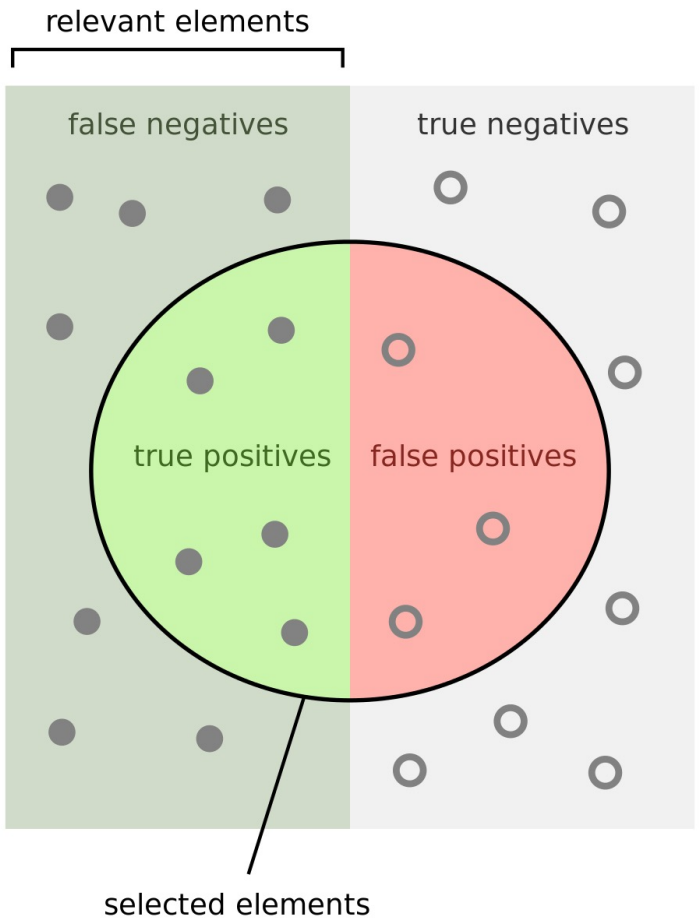


Paolo Perrotta

90% Accuracy!!!

This definition works for binary classification
P/R can also be generalized to multiclass classification

Precision/Recall & F_1 Scores



How many selected items are relevant?

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

High precision is good!

How many relevant items are selected?

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

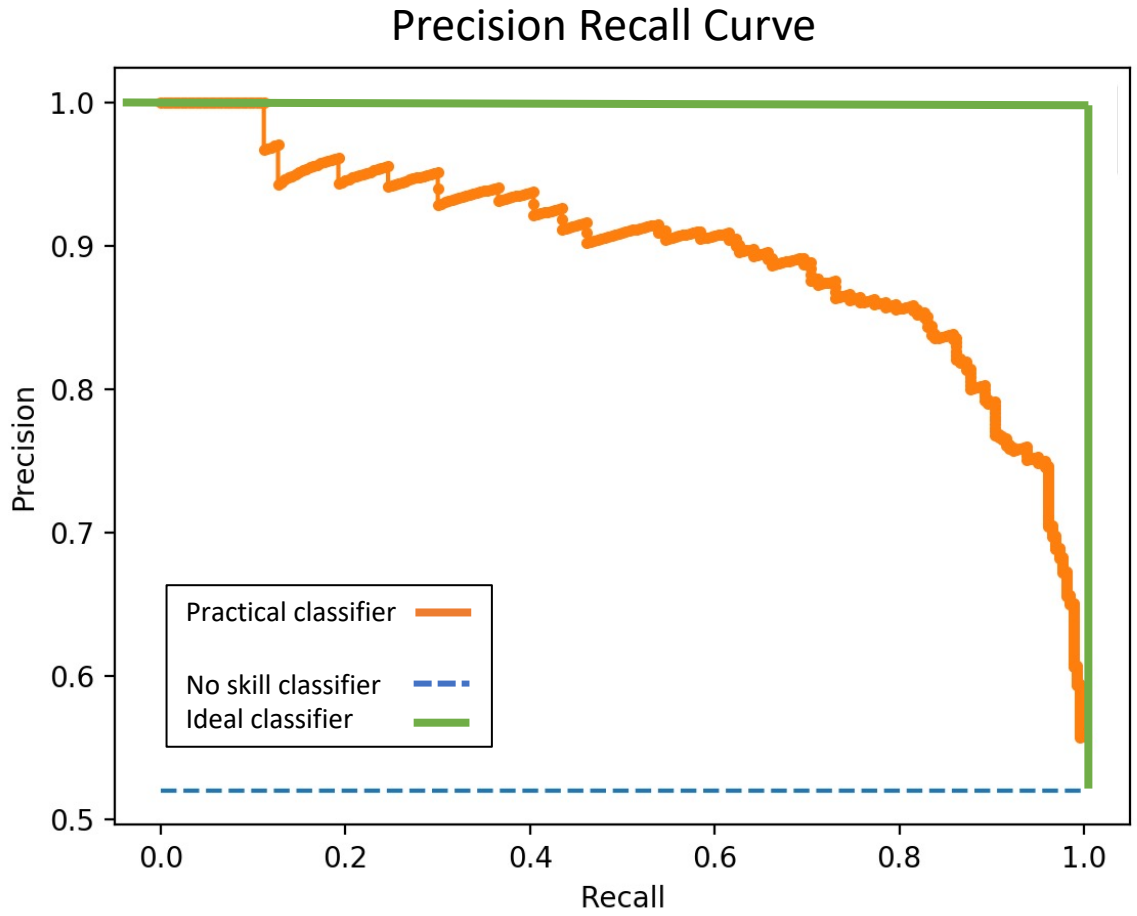
High recall is good!

- What does **high precision**, **low recall** mean conceptually?
  - In a real-world context? When is this good/bad?
- What does **low precision, high recall** mean conceptually?
  - In a real-world context? When is this good/bad?

Performance Metrics

- Most ML algorithms are tunable
  - Allow you to trade-off different performance priorities
  - Try a range of values to see **performance curve**

Precision/Recall & F_1 Scores

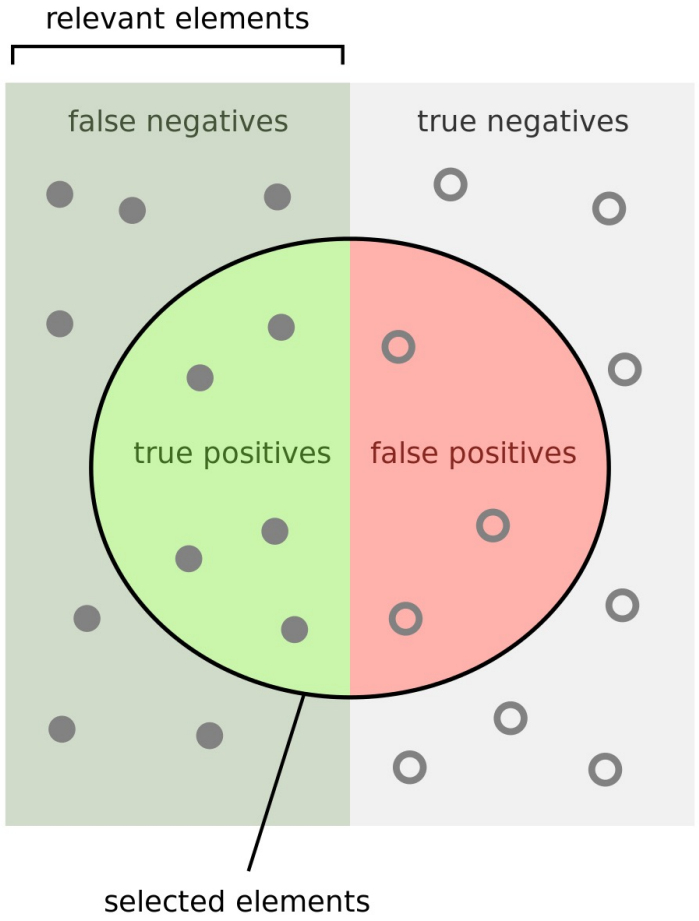


- Often report the **F_1 score** (**harmonic mean** of precision & recall)

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- Harmonic mean gives more weight to low values → why does this matter?

Receiver Operating Characteristic (ROC)



True Positive Rate = $\frac{\text{true positives}}{\text{relevant elements}}$

High TPR is good!

False Positive Rate = $\frac{\text{false positives}}{\text{true negatives}}$

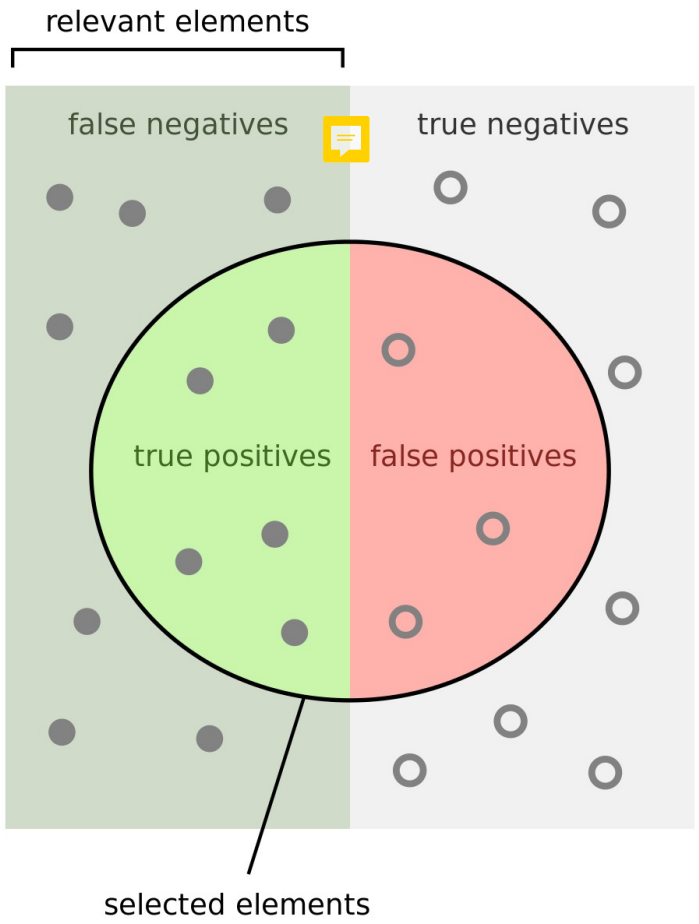
Low FPR is good!

This definition works for binary classification
ROC can also be generalized to multiclass classification

Wikipedia

This definition works for binary classification
P/R can also be generalized to multiclass classification

Precision/Recall & F_1 Scores



How many selected items are relevant?

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

High precision is good!

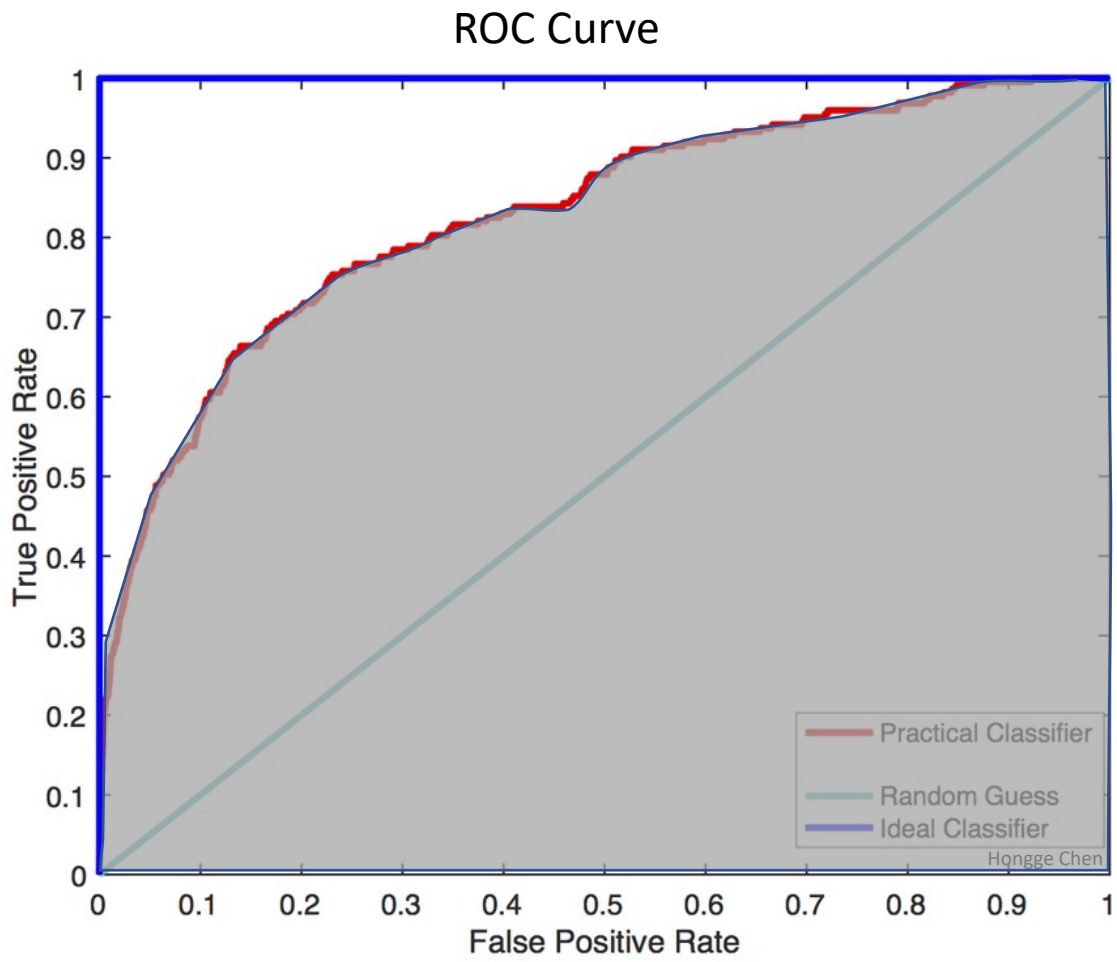
How many relevant items are selected?

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

High recall is good!

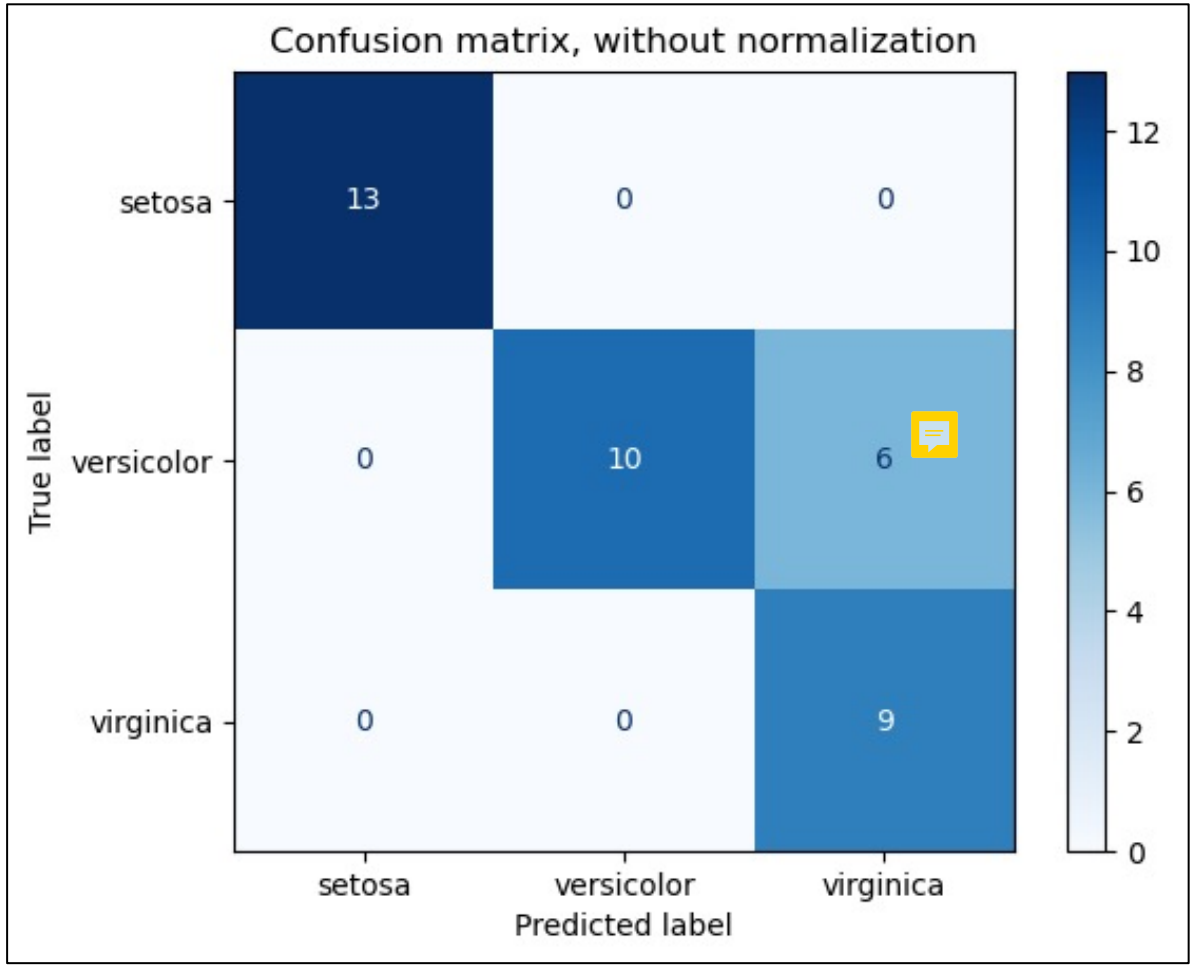
- What does **high precision**, **low recall** mean conceptually?
  - In a real-world context? When is this good/bad?
- What does **low precision, high recall** mean conceptually?
  - In a real-world context? When is this good/bad?

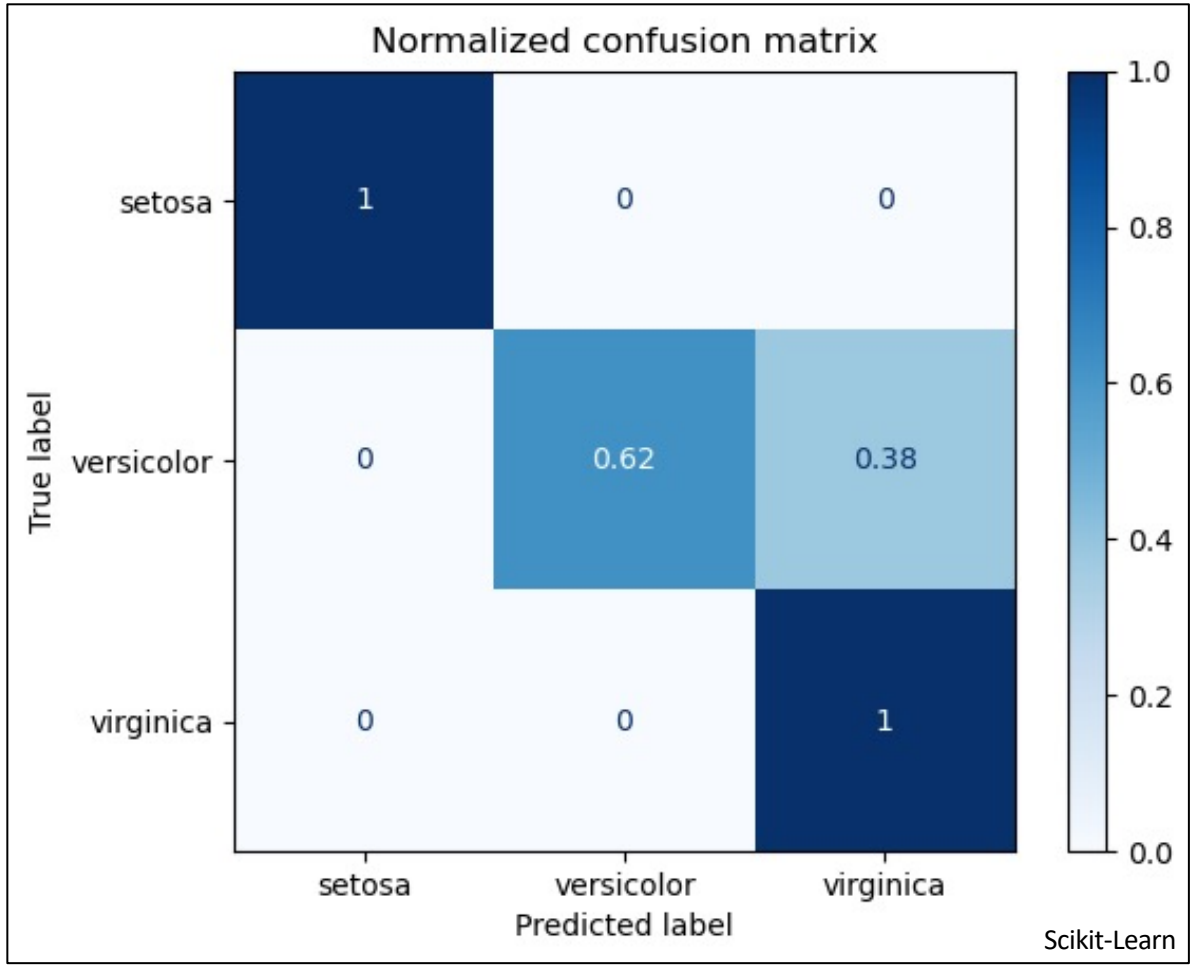
Receiver Operating Characteristic (ROC)



- Often report the area **under the curve (AUC)** as indicator of model performance
  - Ideal AUC = 1

Confusion Matrices





Scikit-Learn

Programming Practice

SupervisedLearning.ipynb