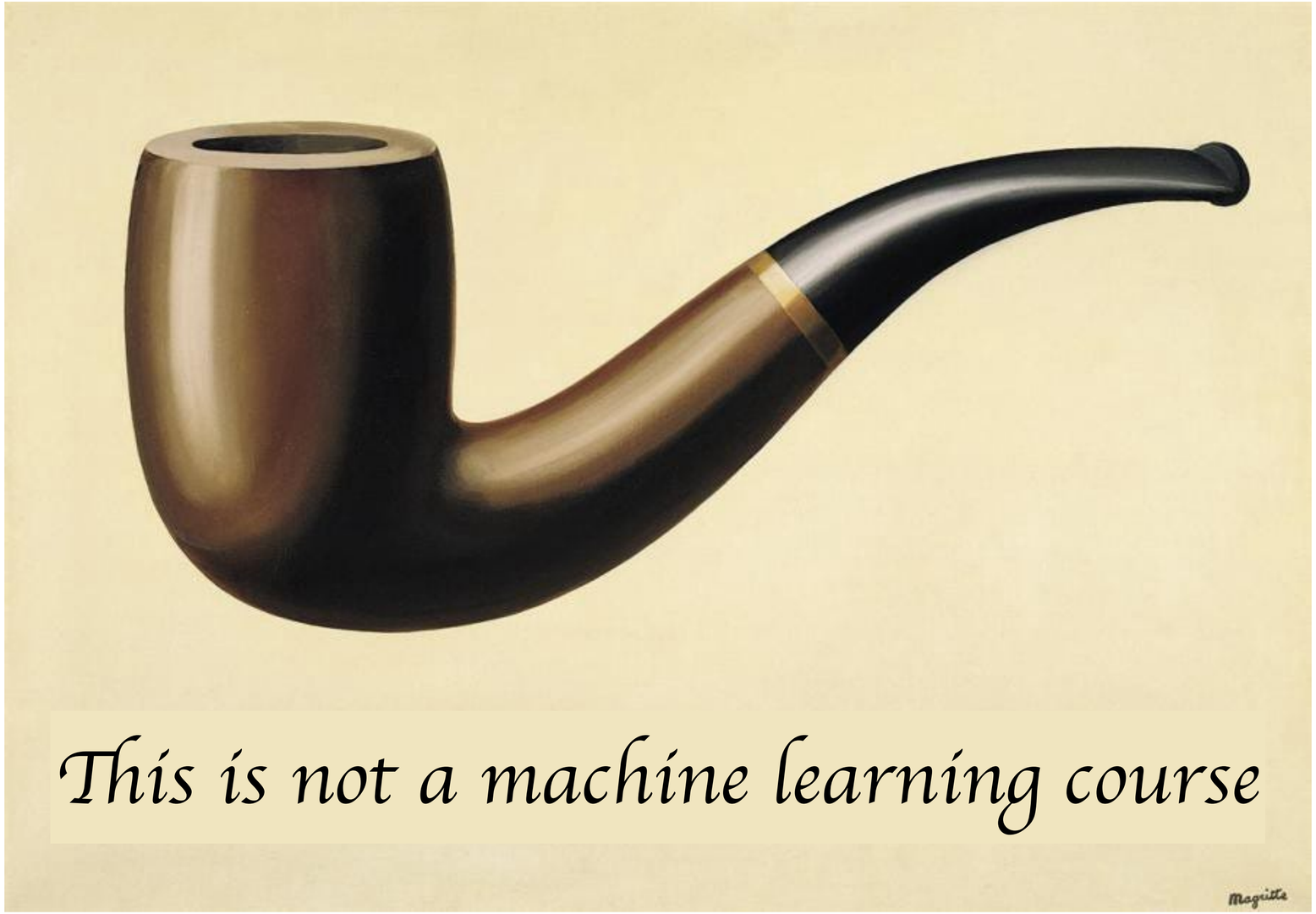
Applied Machine Learning #1

Manoel Horta Ribeiro
manoel@cs.princeton.edu

COS 424 / Fall 2026

The treachery of COS 424



MENU

APPETIZER

Brief supervised learning recap

MAIN COURSE

Metrics to Evaluate Models

DESSERT

Methods to Evaluate Models

MENU

APPETIZER

Brief supervised learning recap

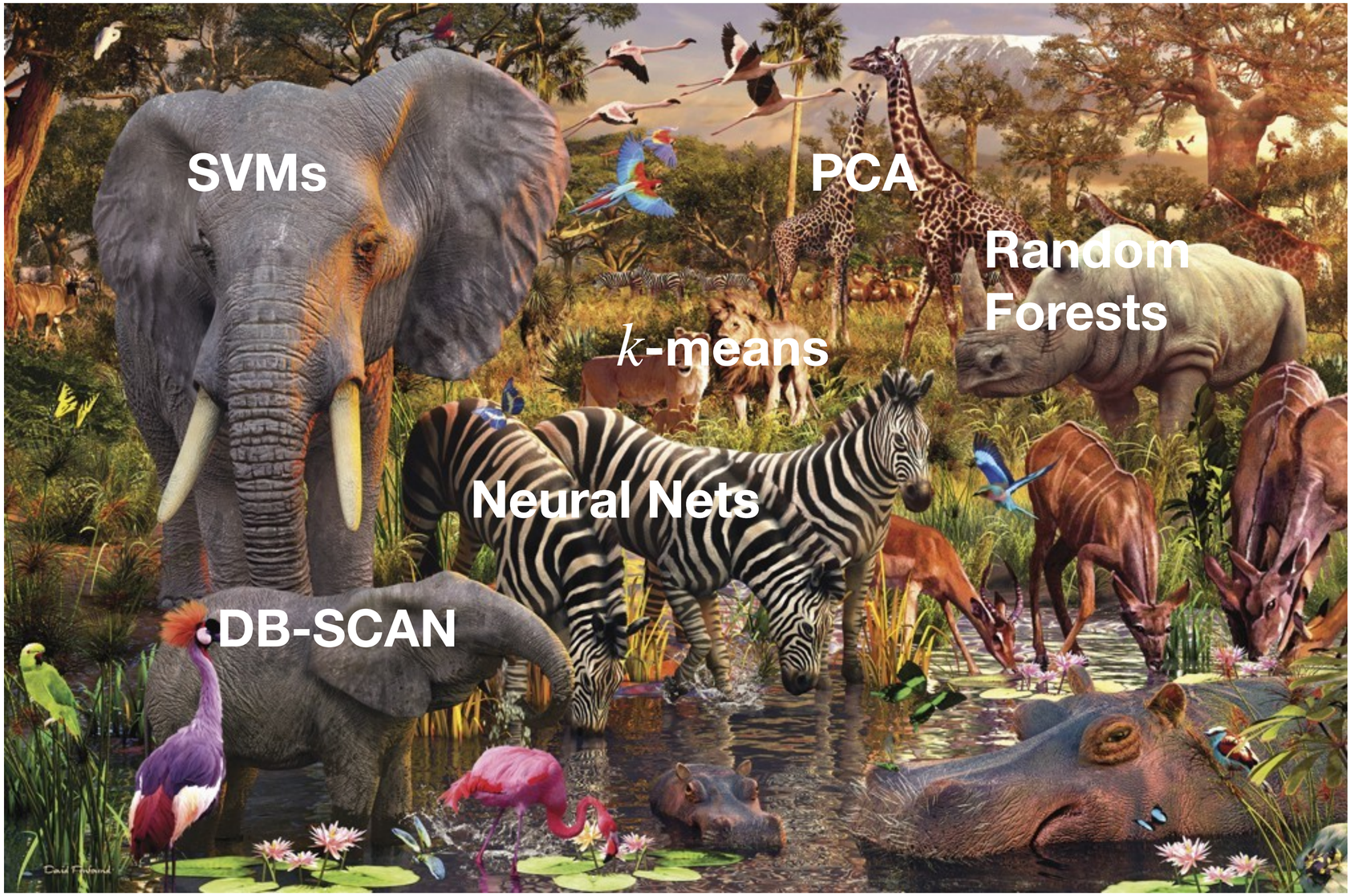
MAIN COURSE

Metrics to Evaluate Models

DESSERT

Methods to Evaluate Models

Early 2000s: A Zoo of Models

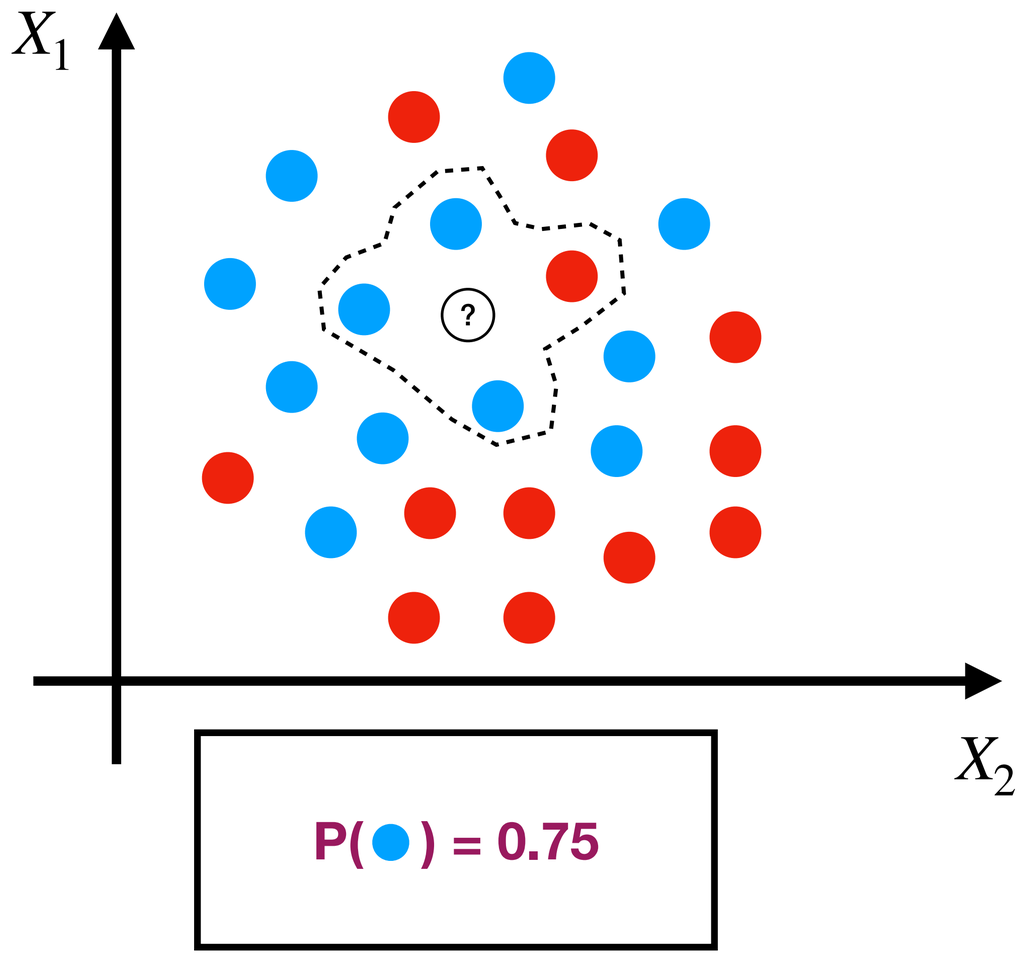


“Traditional” Supervised Learning

- Consider *features* $\{X_1, \ldots, X_n\}$ and an *outcome* Y
- Given a sample $\{x_1^i, x_2^i, \ldots, x_n^i, y_i\}$, learn $f(\mathbf{X}) \simeq Y$
- We want $f(\mathbf{X})$ to generalize to new, unseen samples!
  - If Y is continuous → **“regression,”**
  - If Y is categorical → **“classification.”**

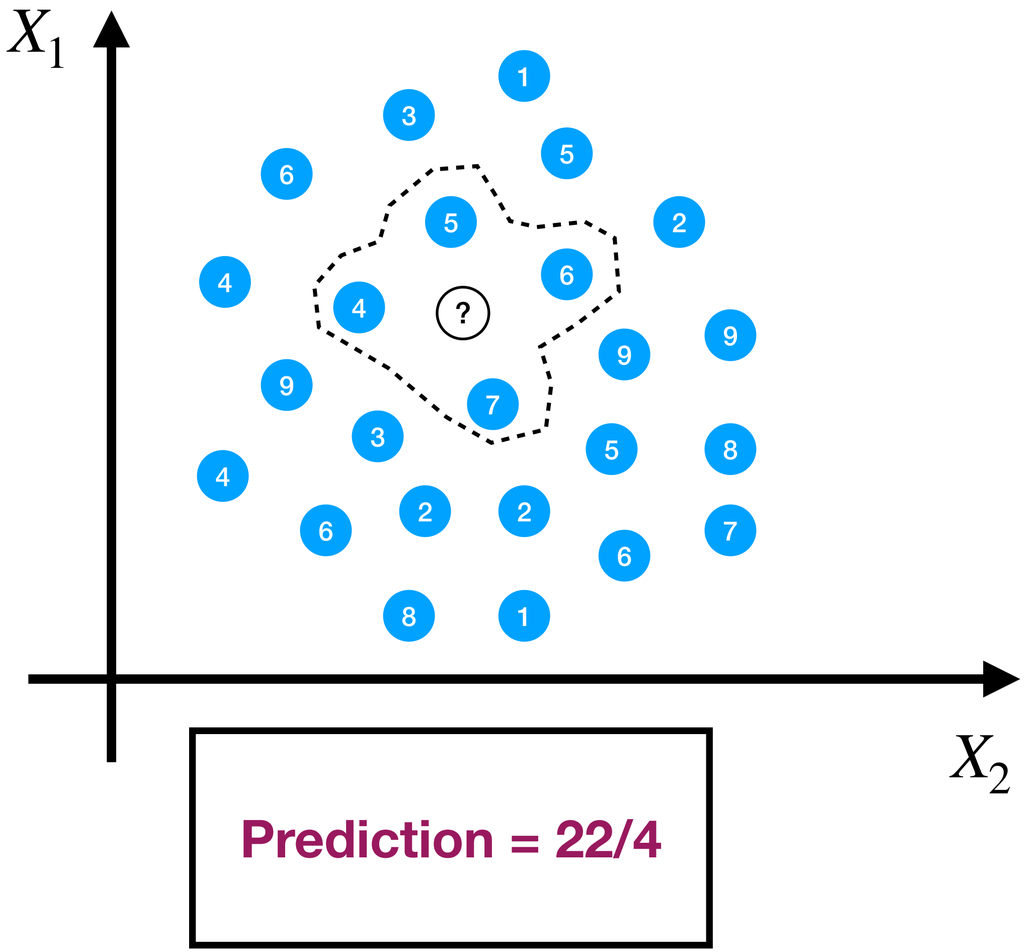
Simple: k-nearest neighbors

- Find k the nearest neighbors according to some distance.
- Predict the “average” category or “average” value.
- Decisions:
  - Distance? $d(x, y) = \|x - y\|$
  - Weighting neighbors?



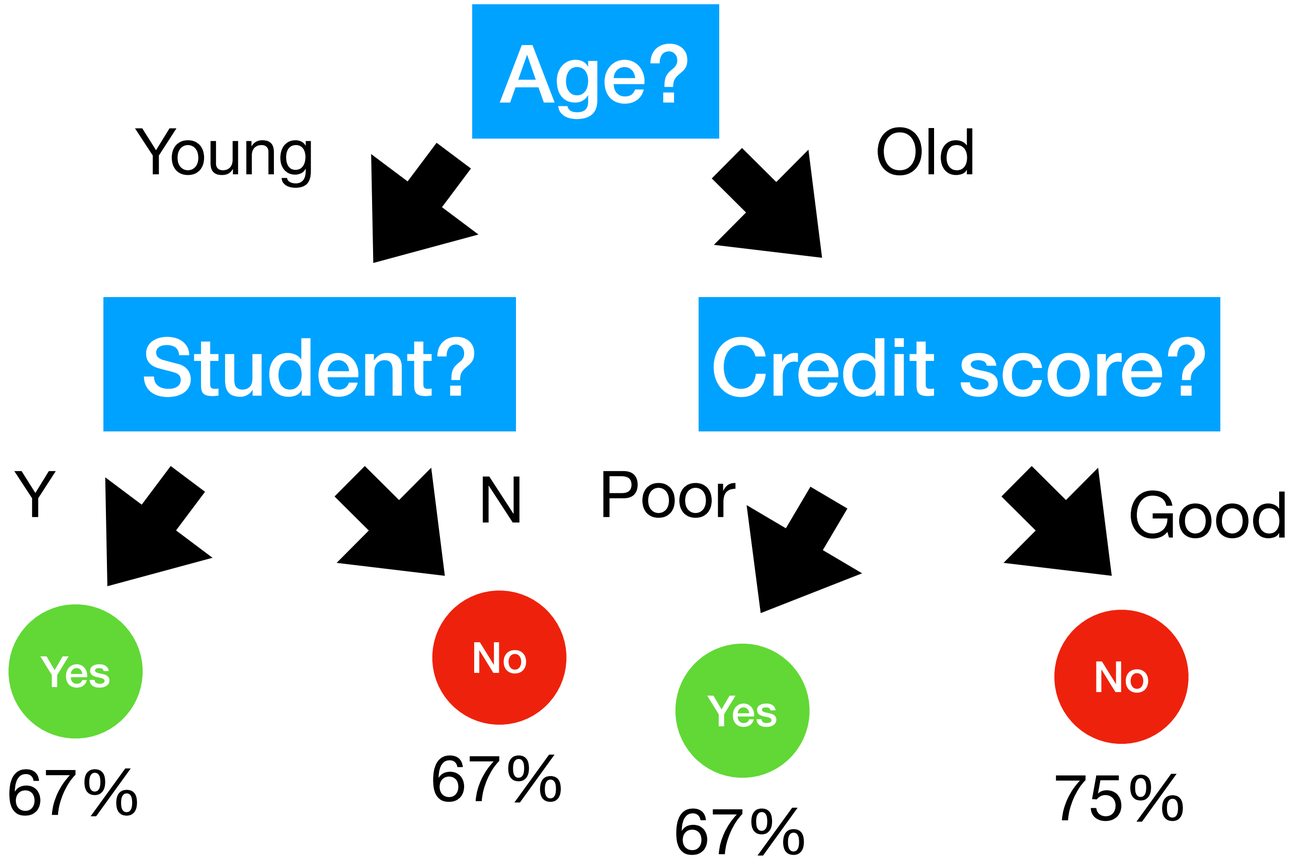
Simple: k-nearest neighbors

- Regression version is similar!
- Take the nearest neighbors and aggregate their values to produce a prediction!



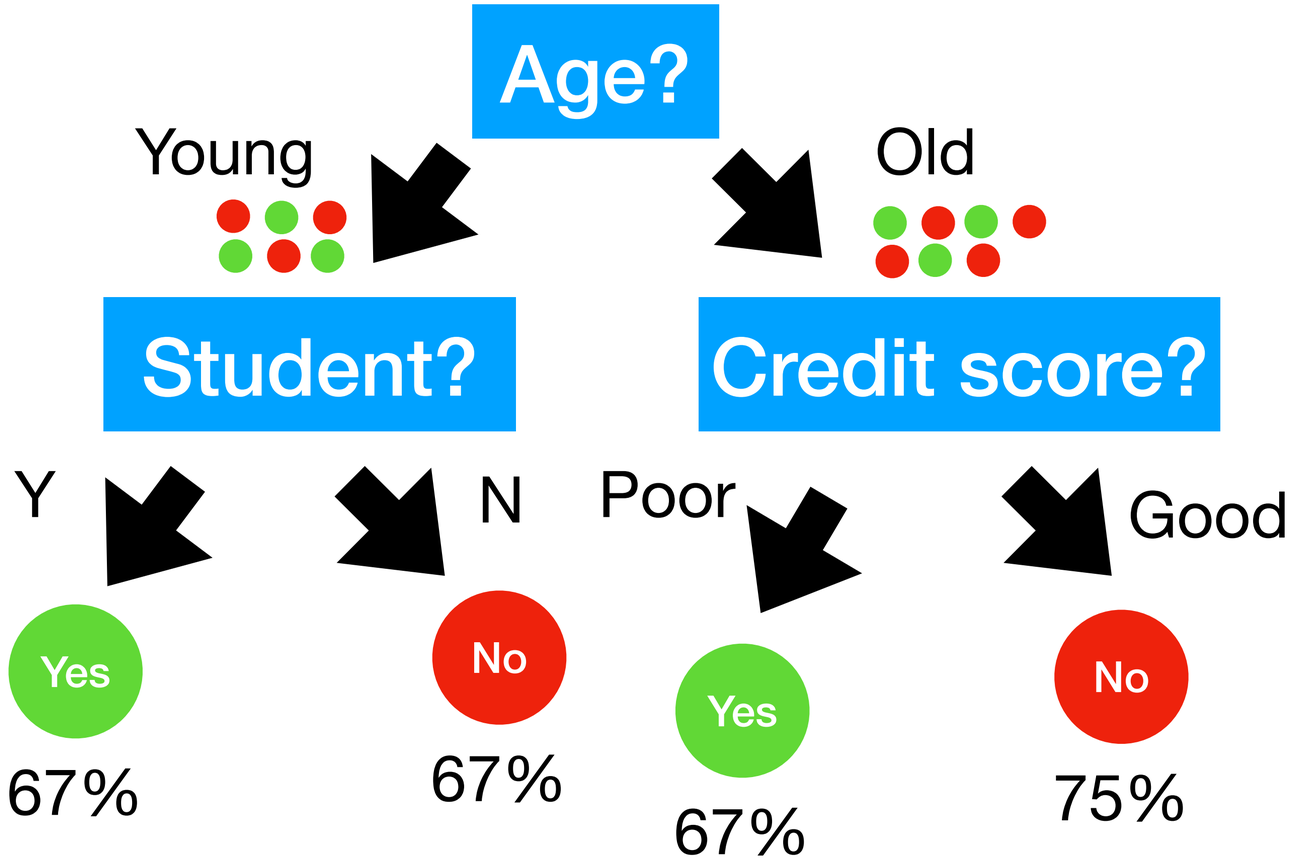
Decision Trees

Age	Student	Credit	Buys Computer
Young	Yes	Poor	No
Young	Yes	Poor	Yes
Young	Yes	Poor	Yes
Young	No	Poor	No
Young	No	Poor	No
Young	No	Good	Yes
Old	Yes	Poor	No
Old	No	Poor	Yes
Old	No	Poor	Yes
Old	No	Good	No
Old	No	Good	No
Old	No	Good	Yes
Old	No	Good	No



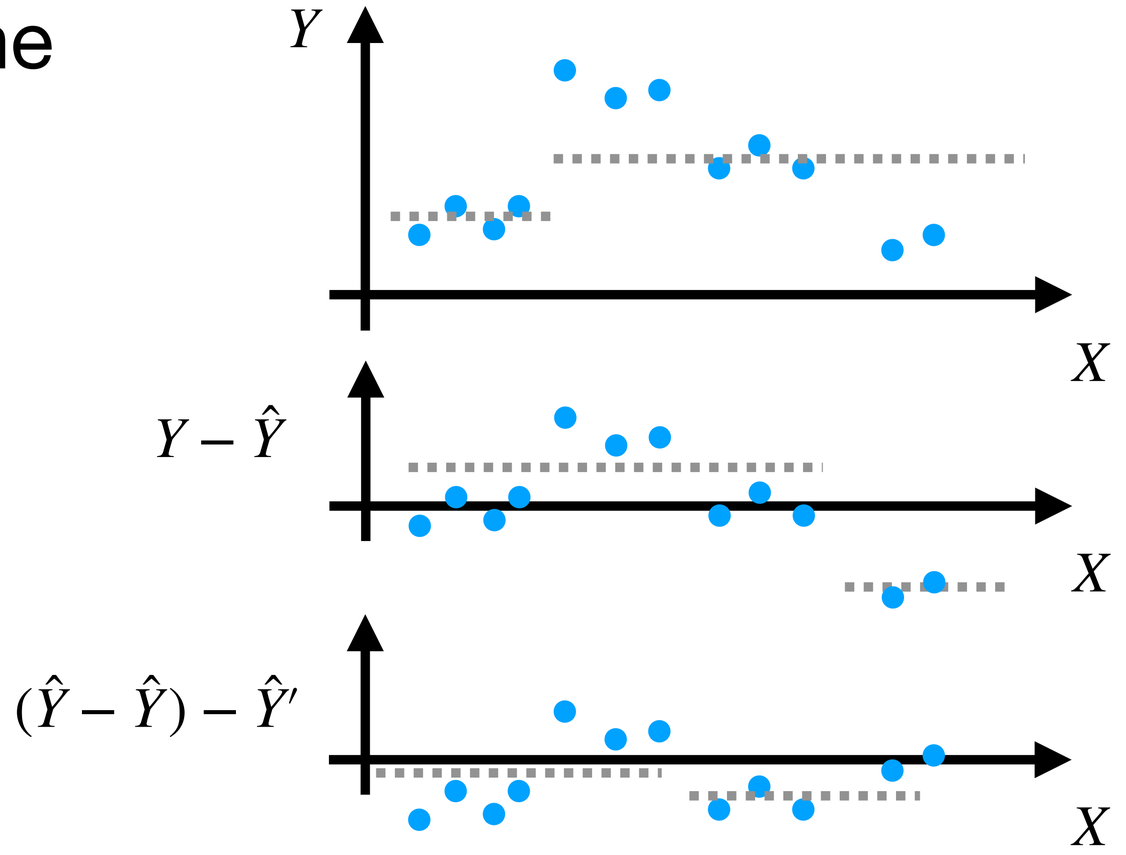
Decision Trees

- Building trees:
  - All data “starts” in a root node
  - Select a split (usually, the one that reduces entropy the most!)
  - Recursively repeat on each child
- Important parameters:
  - Maximum tree depth
  - Stopping criteria



The Star: Gradient Boosting

- Dominated competitive Machine Learning (e.g., Kaggle).
- Very hard to beat! Easy to use!
- **Idea**: series of “weak” models fit successively on the residuals.

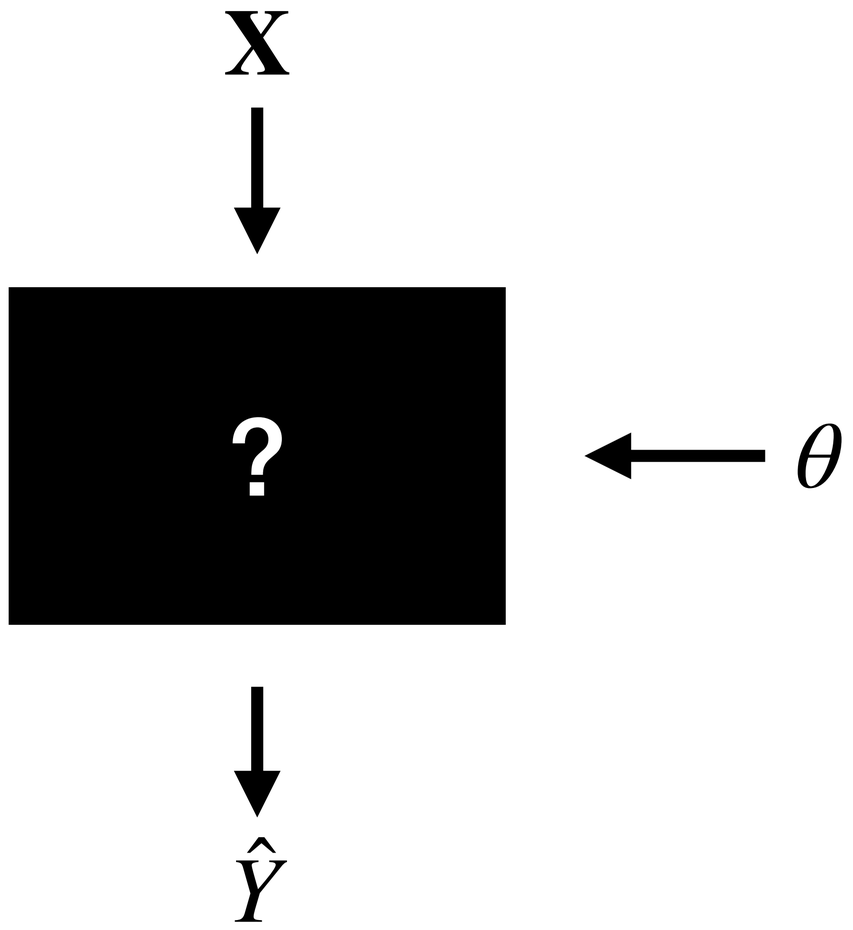


LightGBM

DMLC XGBoost

In Practice

- Models provide a predictive function $f(\mathbf{X}; \theta)$ based on some data $\mathbf{X}$ and hyper-parameters θ.
- The exact model does not matter so much…
- How to assess their performance?



MENU

—

APPETIZER

Brief supervised learning recap

—

MAIN COURSE

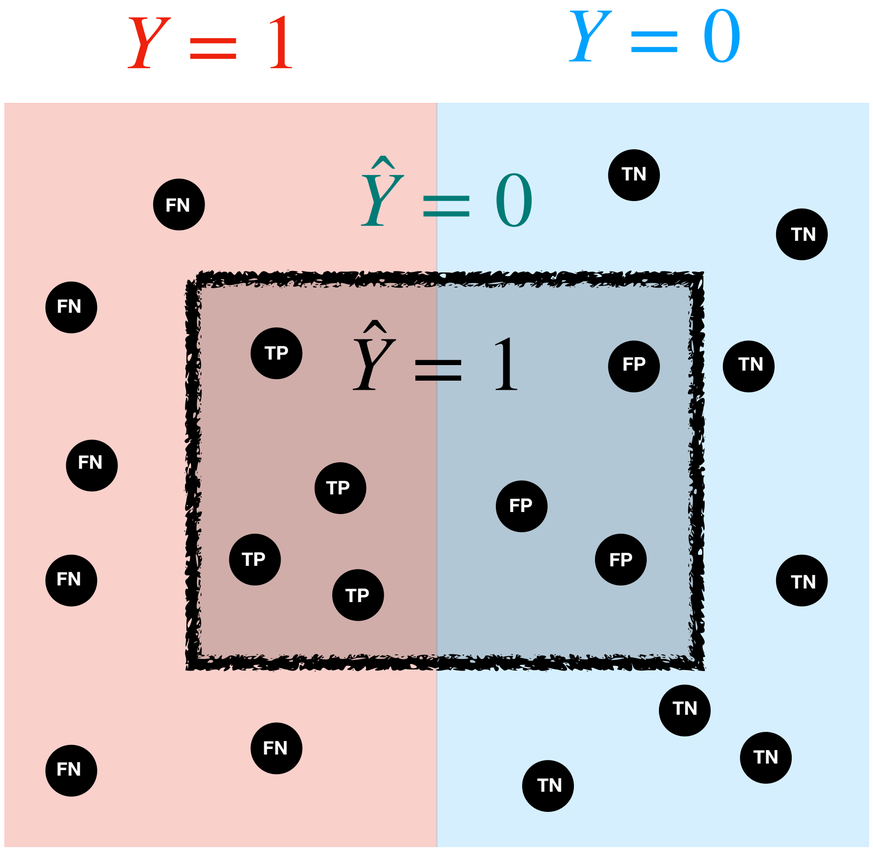
Metrics to Evaluate Models

—

DESSERT

Methods to Evaluate Models

Performance Metrics



Y is the true label

$\hat{Y}$ is the predicted label

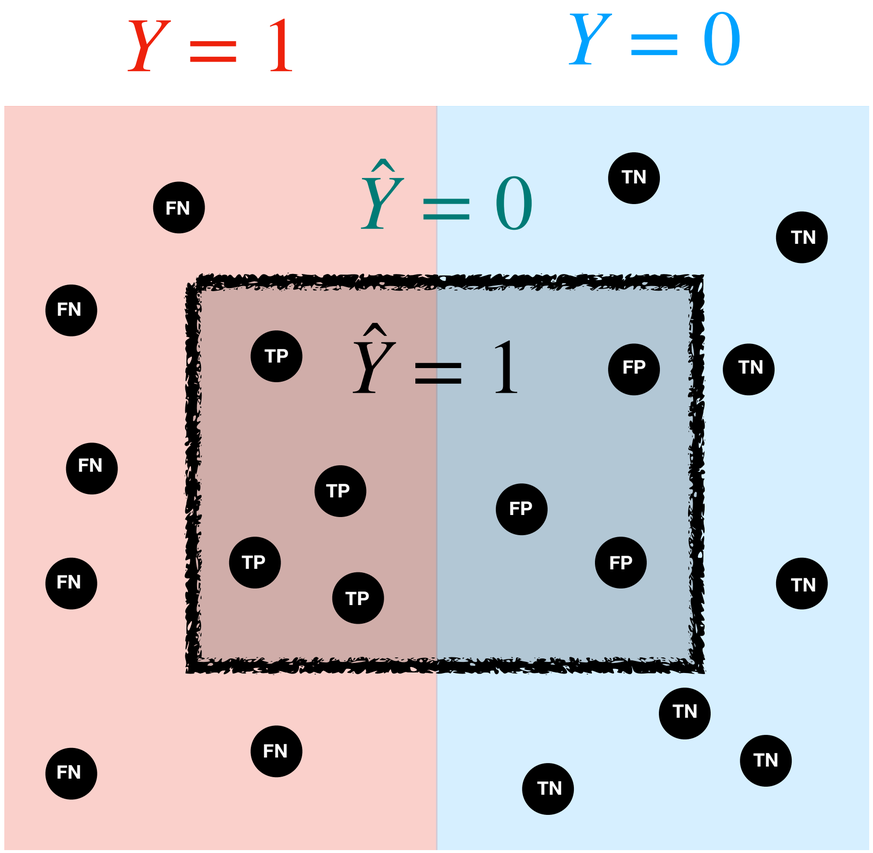
$Y = 1, \hat{Y} = 1$: **T**rue **P**ositives

$Y = 0, \hat{Y} = 0$: **T**rue **N**egatives

$Y = 0, \hat{Y} = 1$: **F**alse **P**ositives

$Y = 1, \hat{Y} = 0$: **F**alse **N**egatives

Performance Metrics



Accuracy:

$$\frac{\text{True Positive} \quad \text{True Negative}}{\text{True Positive} \quad \text{True Negative} \quad \text{False Positive} \quad \text{False Negative}}$$

Precision:

$$\frac{\text{True Positive}}{\text{True Positive} \quad \text{False Positive}}$$

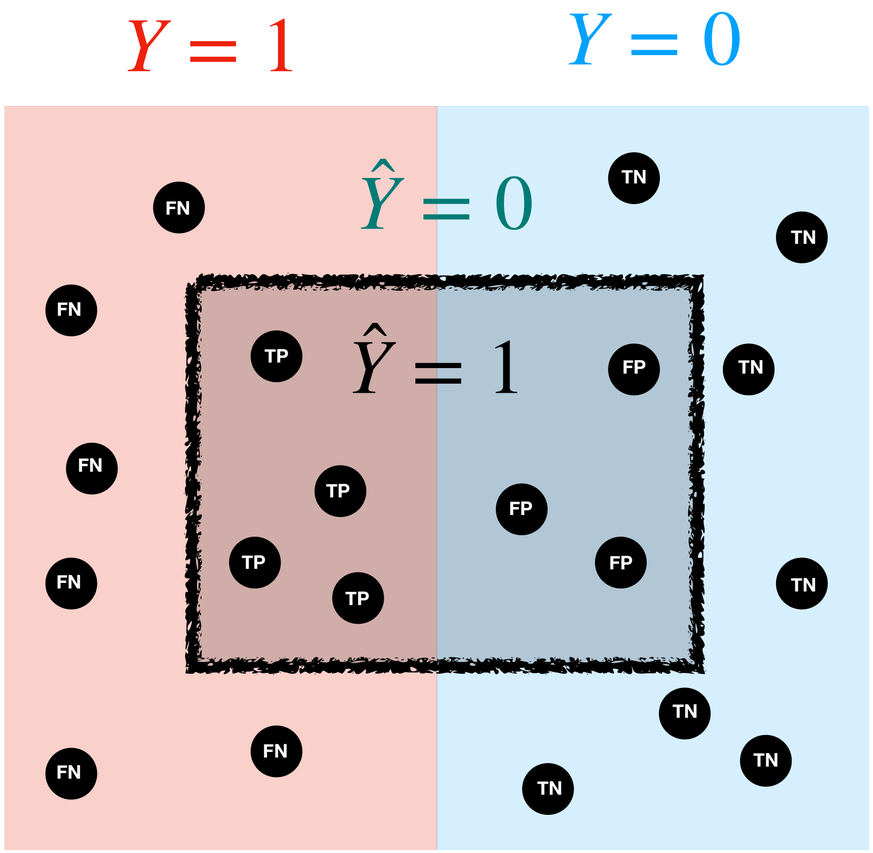
If the test says you have the flu, what is the chance of you actually having it?

Recall:

$$\frac{\text{True Positive}}{\text{True Positive} \quad \text{False Negative}}$$

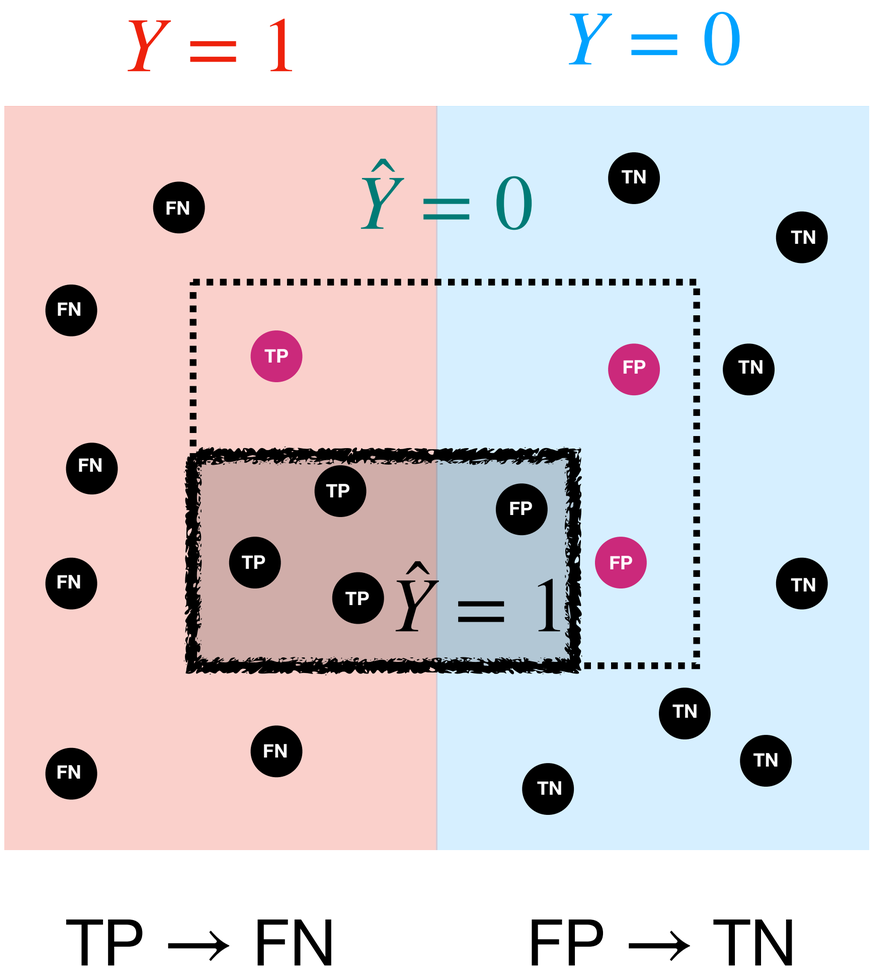
What percentage of flu cases will the test catch?

Decision Thresholds



- Most classifiers yield a *probability* for a class. E.g., $P(Y = 1) = \alpha$!
- Typically, if $\alpha \geq 0.5$, we predict $\hat{Y} = 1$ (and $\hat{Y} = 0$ otherwise)
- But 0.5 is really an arbitrary threshold τ! We can vary it!

Decision Thresholds



If we increase τ, i.e., if we are more strict, we <u>usually</u> **increase** precision and **decrease** recall!

↑ Precision:

$$\frac{\downarrow \text{True Positive}}{\downarrow \text{True Positive} + \downarrow \text{False Positive}}$$

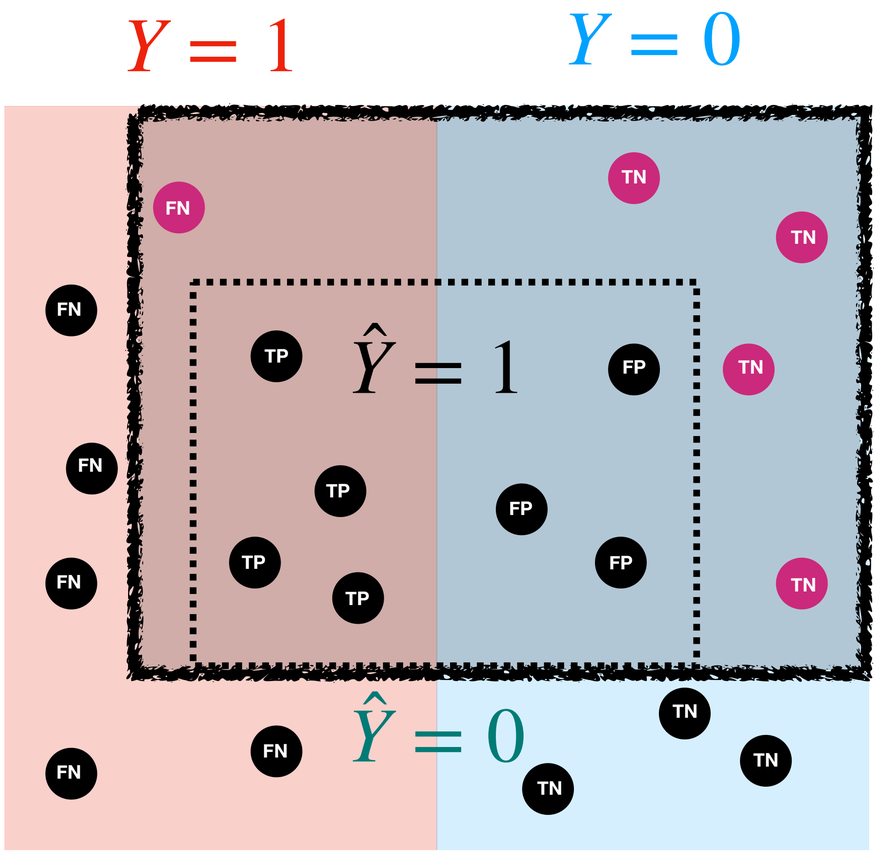
False Positives decrease;
True Positives decrease;

↓ Recall:

$$\frac{\text{True Positive} \downarrow}{\downarrow \text{True Positive} + \uparrow \text{False Negative}}$$

True Positives decrease;
False Negatives increase;

Decision Thresholds



FN → TP TN → FP

If we decrease τ, i.e., if we are less strict, we <u>usually</u> **decrease** precision and **increase** recall!

↓ Precision:

$$\frac{\uparrow \text{True Positive}}{\uparrow \text{True Positive} + \uparrow \text{False Positive}}$$

False Positives decrease;
True Positives decrease;

↑ Recall:

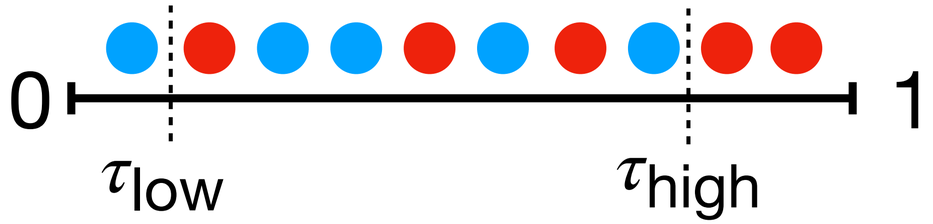
$$\frac{\uparrow \text{True Positive}}{\uparrow \text{True Positive} + \downarrow \text{False Negative}}$$

True Positives Increase;
False Positives decrease;

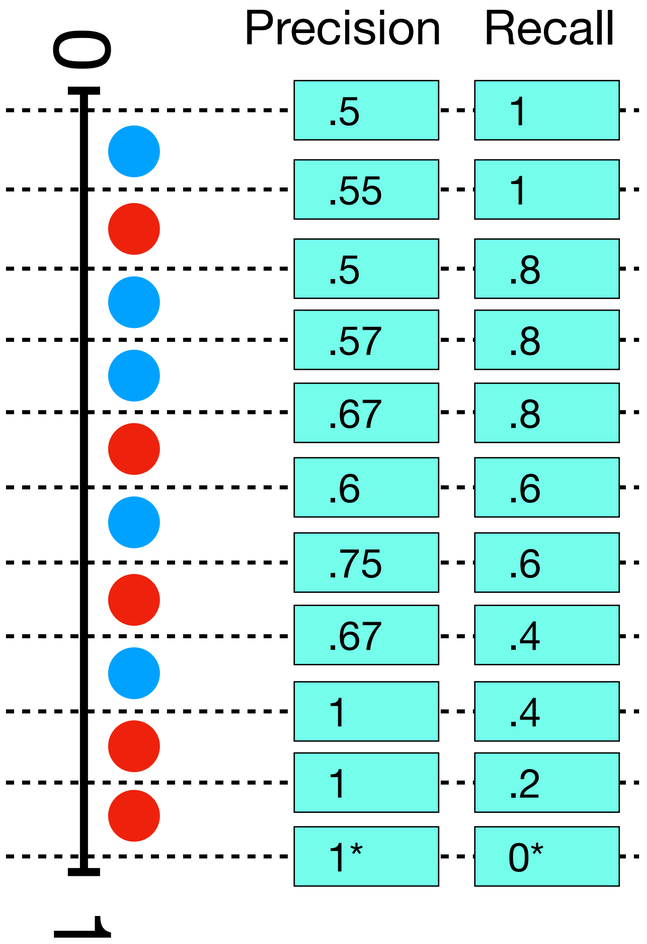
Intuition: Medical Test

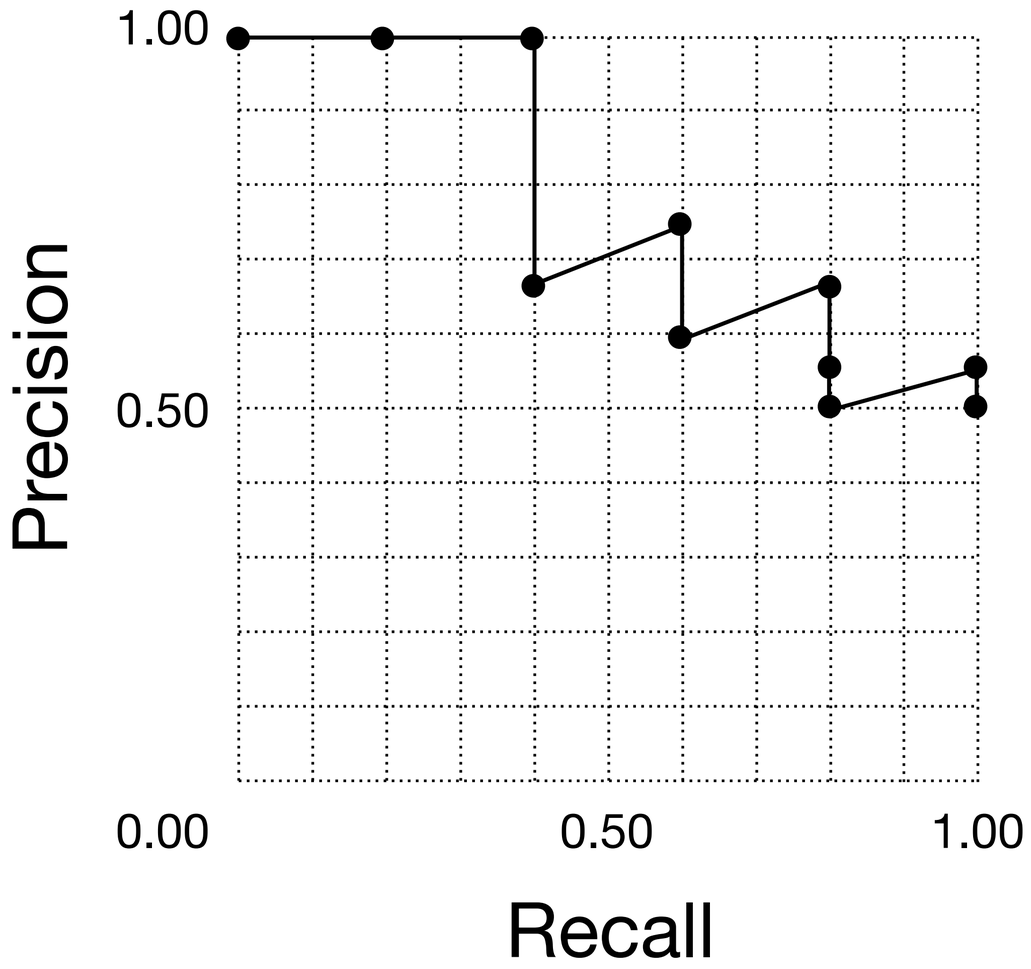
- A medical test screens for an infectious disease.
  - The test outputs a number [0,1]
  - Higher numbers correlate with positive outcomes
  - Suppose you return a positive test if the number $\geq \tau$

- Assume the following n=10 results:
  - Maximum precision comes at the expense of many false negatives
  - Maximum recall comes at the cost of many false positives
  - In-between these two values, you trade off precision and recall!



Precision Recall Curve



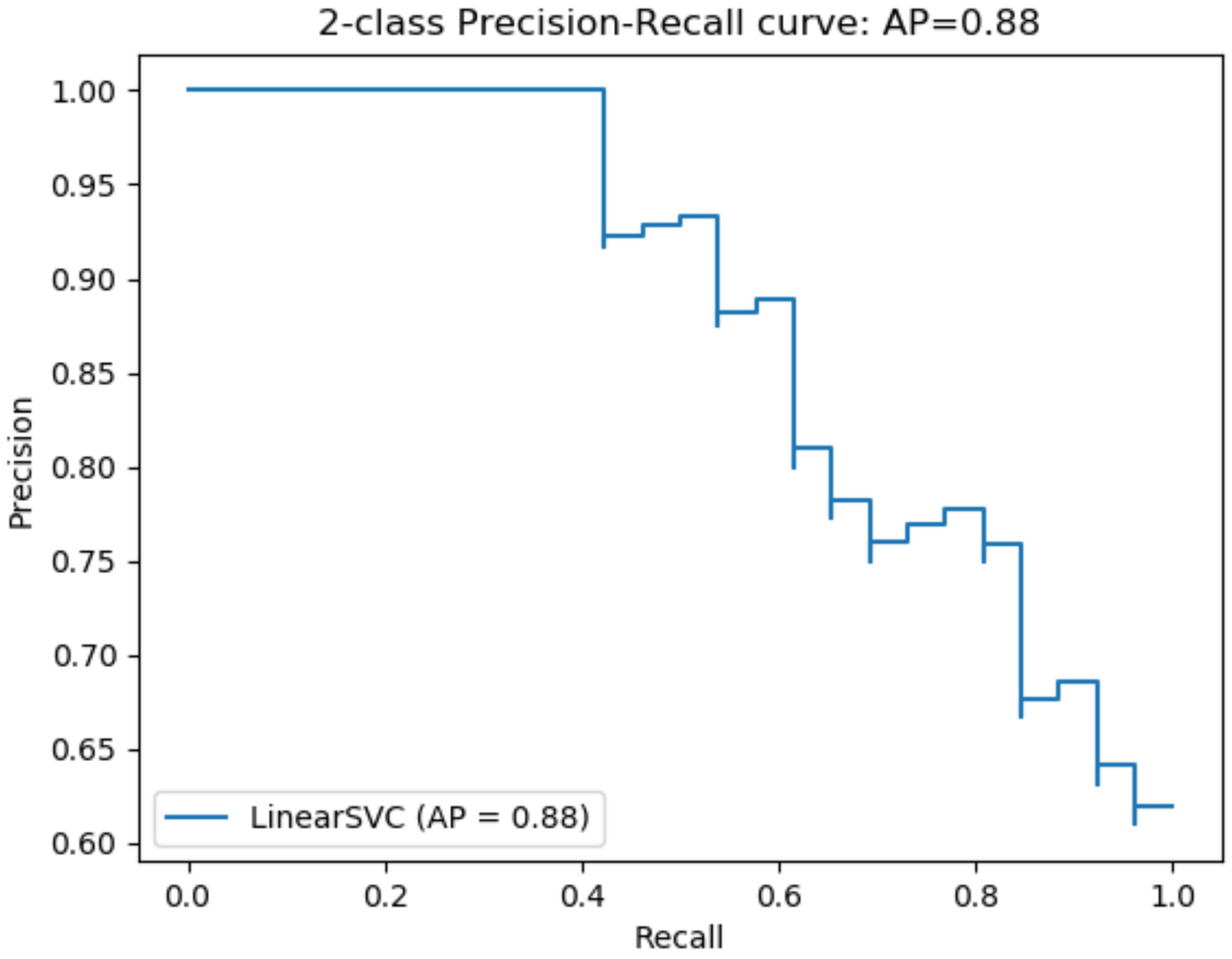


Precision Recall Curve

- F-1 score is the harmonic mean of precision & recall.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Typically, the curve is created with interpolation!

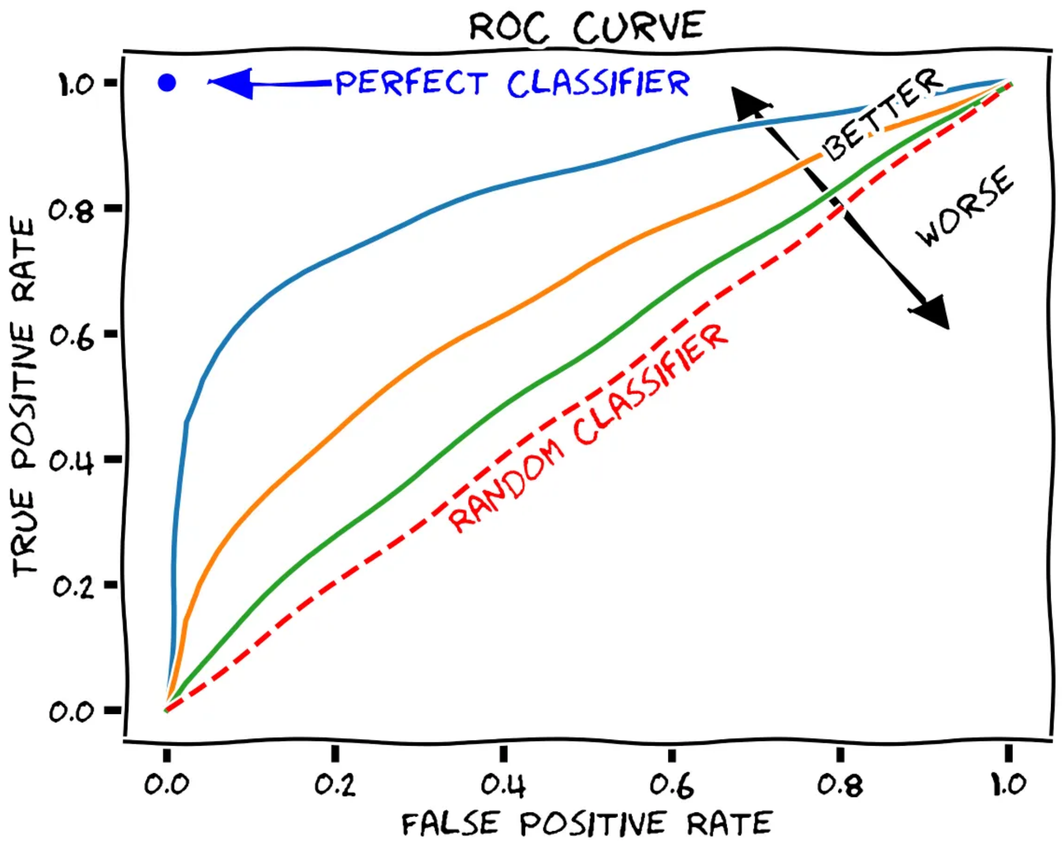


ROC Curve

- **R**eceiver **O**perating **C**haracteristic (WW2 name!)

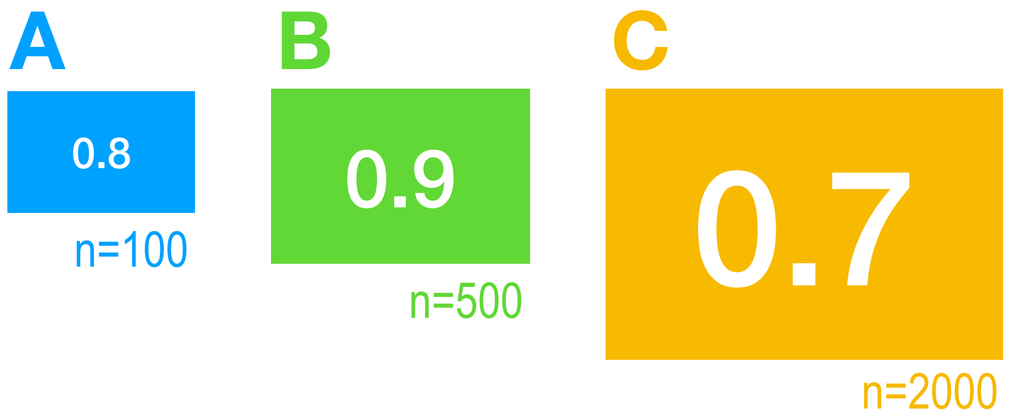
$$\text{False Positive Rate} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}}$$

- Simply an FPR-Recall curve
- Often used in medicine.



What if You Have Multiple Labels?

- Most metrics we discussed can be computed per label.
  - e.g., {A, B, C} → {{A}, {B, C}}



- **Key decision:**
  - average per label
  - average per datapoint

$$\text{macro} = \frac{0.8 + 0.9 + 0.7}{3} = 0.8$$

$$\text{micro} = \frac{0.8 \times 100 + 0.9 \times 500 + 0.7 \times 2000}{2600} \simeq 0.74$$

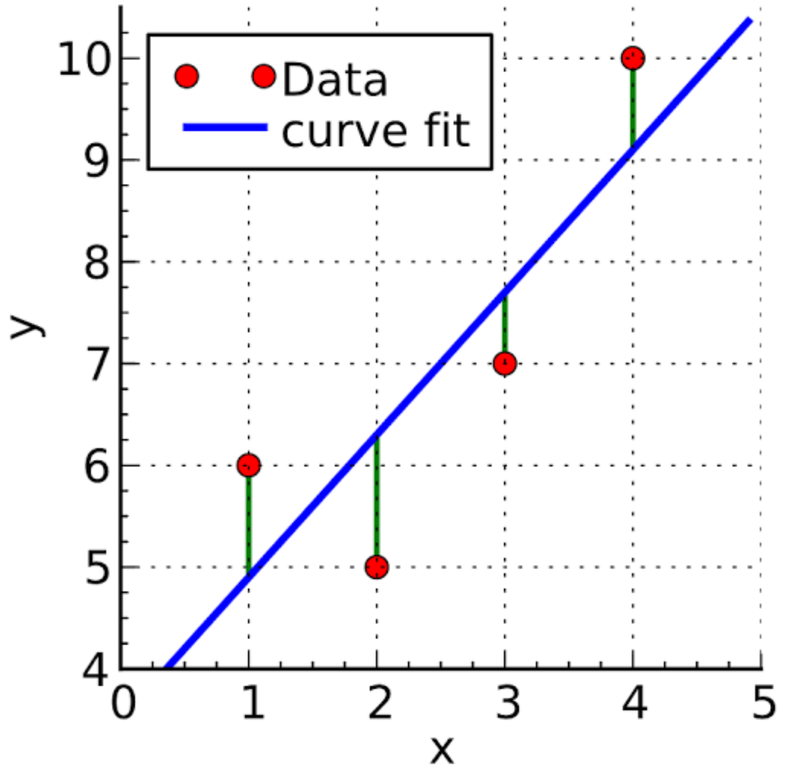
Regression-related metrics

Regression is typically more straightforward to assess.

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{Y_i}$$

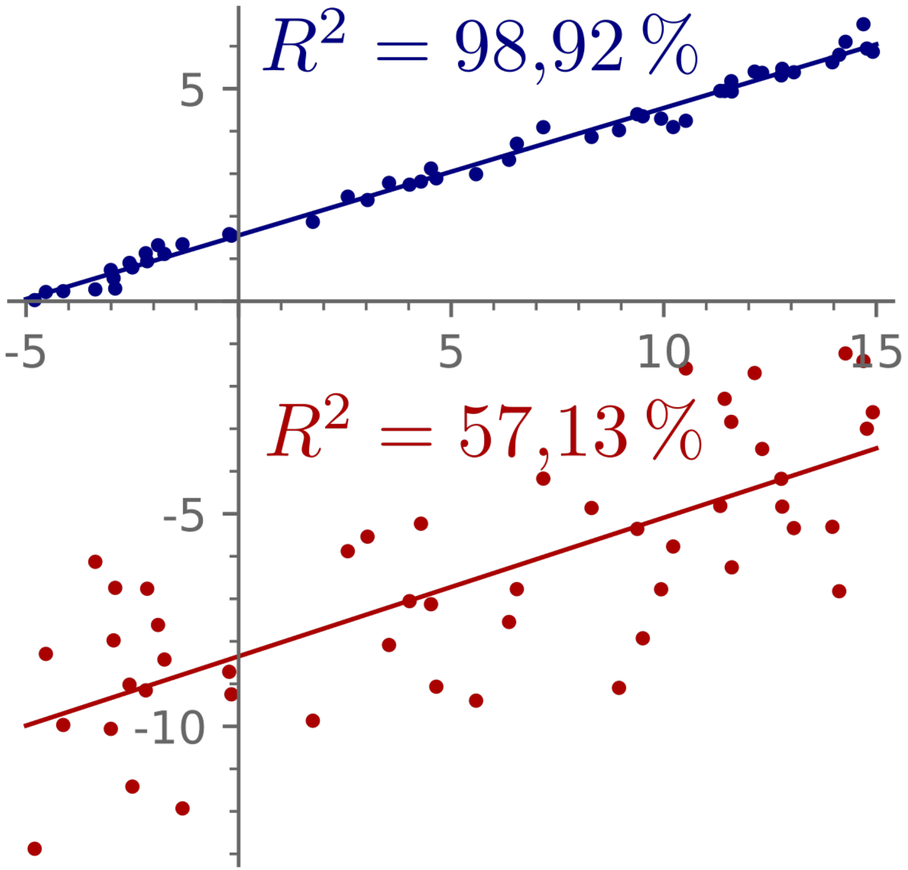


Coefficient of Determination

- How much of the variance in the data does your model explain?

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}$$

- Two extremes:
  - Always predict $\bar{Y} \rightarrow R^2 = 0$
  - $\hat{Y}_i = Y_i, \forall i \rightarrow R^2 = 1$



MENU

APPETIZER
Brief supervised learning recap

—

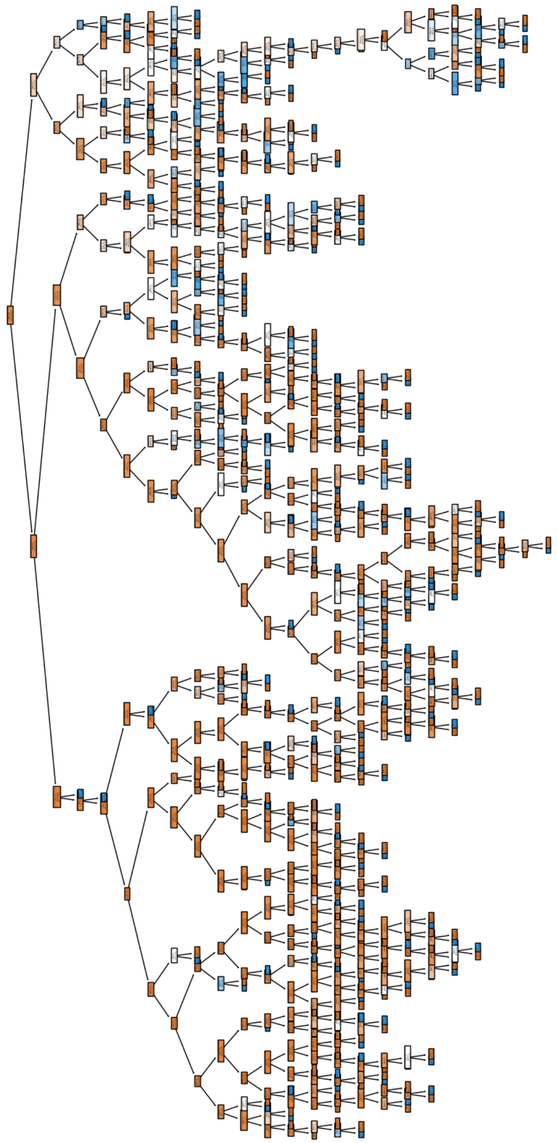
MAIN COURSE
Metrics to Evaluate Models

—

DESSERT
Methods to Evaluate Models

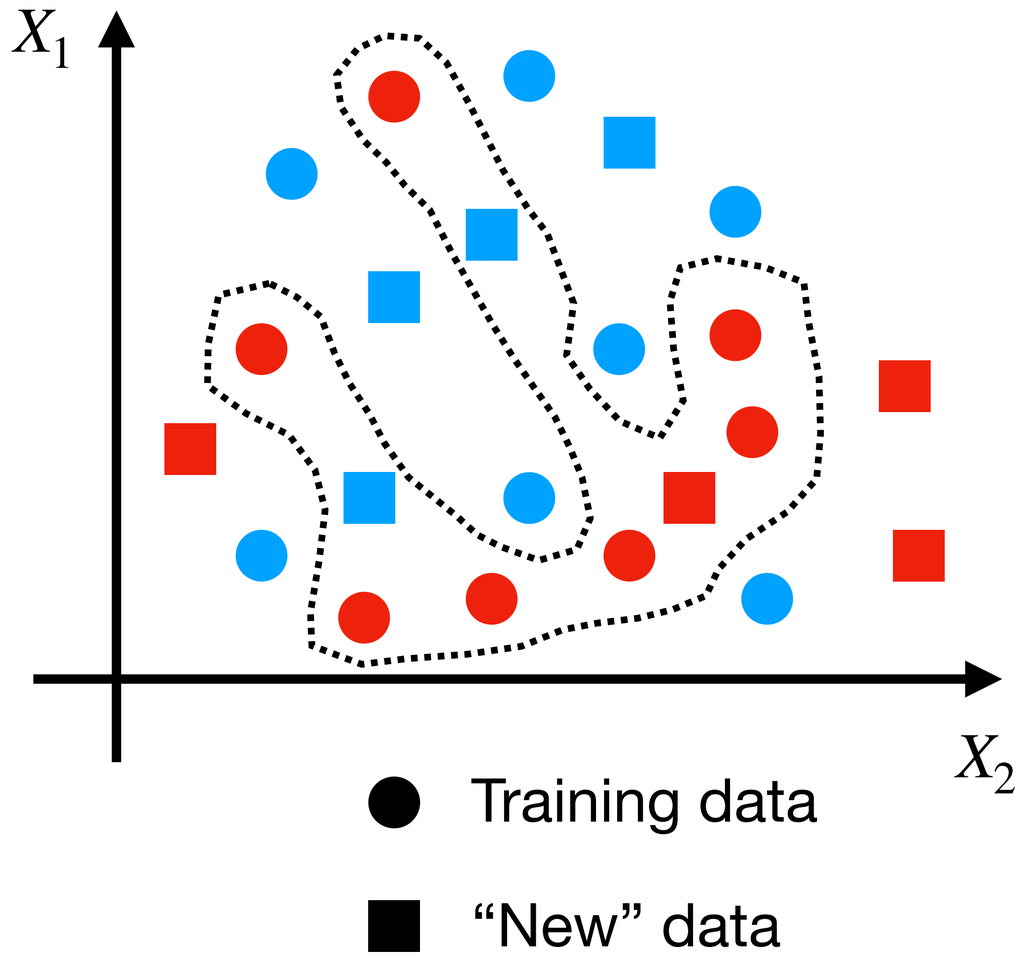
Typical story

- One creates a very complex model, e.g., a very deep decision tree!
- They achieve excellent performance, e.g., 95% accuracy
- They then apply this model to other data, and performance drops tremendously, e.g., to 70% accuracy



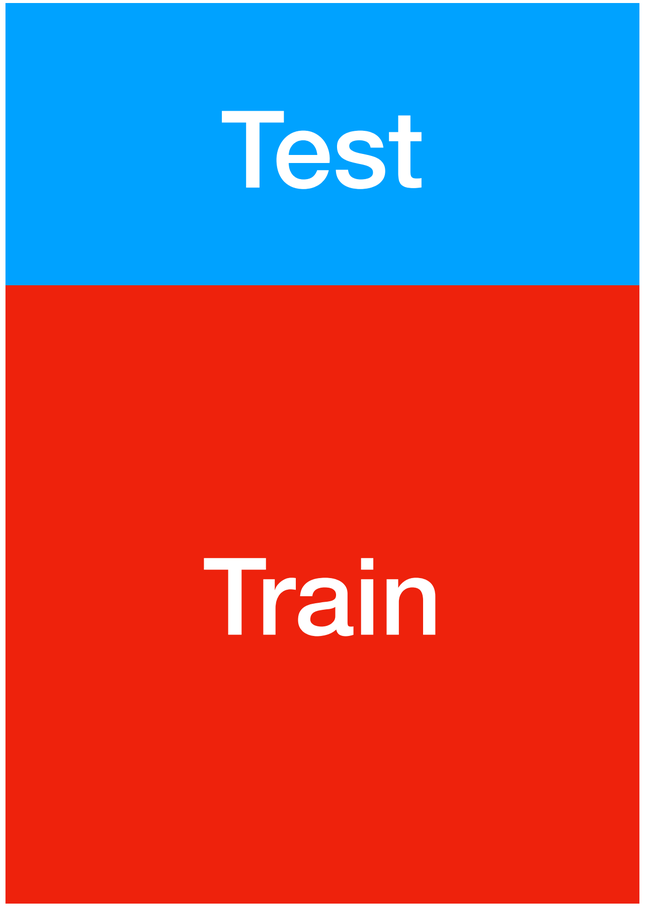
In the real world…

- Oftentimes models underperform outside of training!
- More complex models can fit the training data better…
  - E.g., suppose no data points have exactly the same features. A deep enough decision tree can achieve 100% accuracy!
- But this may lead them to learn “noise” from the training data

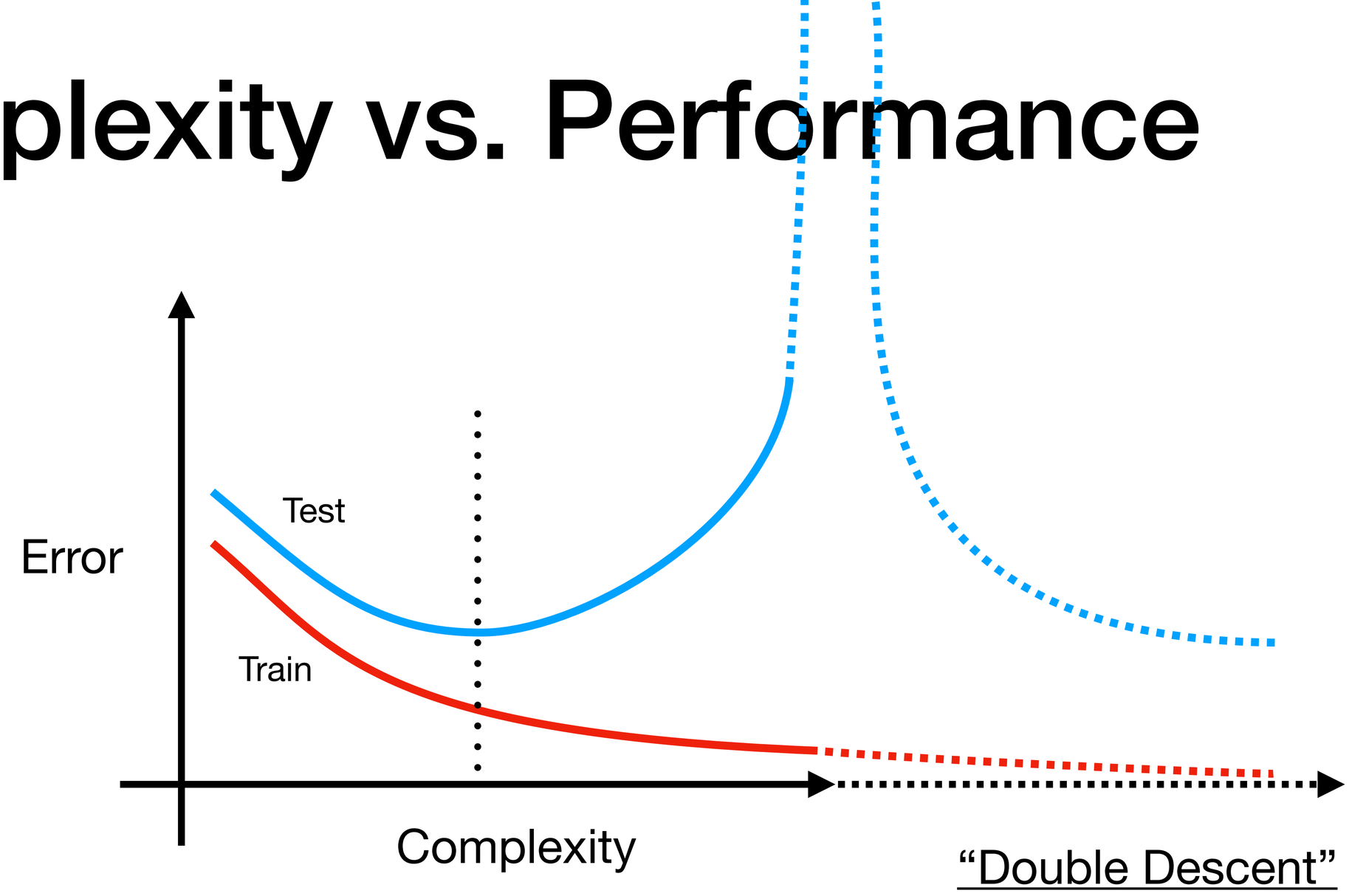


Train-Test Split

- **Idea**: split the data into two disjoint sets
  - **Training set**: anything goes
  - **Test set:** where we evaluate performance
- This works surprisingly well!
  - Hardt, 2026**:** with n datapoints we can evaluate k models with error about $\sqrt{log(k)/n}$,

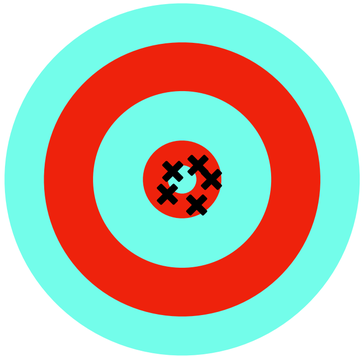


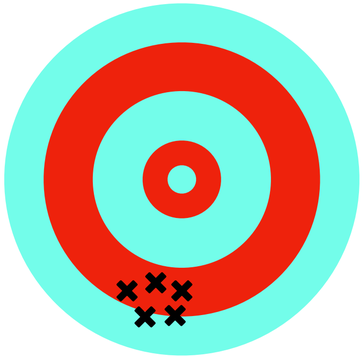
Complexity vs. Performance



Bias and Variance

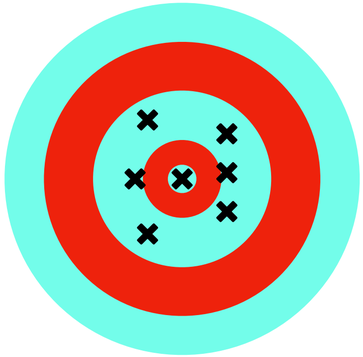
Low Bias, Low Variance
Predictions are:
on average, correct!
vary little!





High Bias, Low Variance
Predictions are:
on average, wrong
vary little!

Low Bias, High Variance
Predictions are:
on average, correct
vary a lot!



High Bias, High Variance
Predictions are:
on average, wrong
vary a lot!

Hyperparameter Tuning

- ML algorithms often have parameters that are not learned from data!
  - The k in k nearest neighbors!
  - Maximum tree depth in decision trees!
- "Hyperparameter tuning" is the process of finding the best configuration of these parameters!

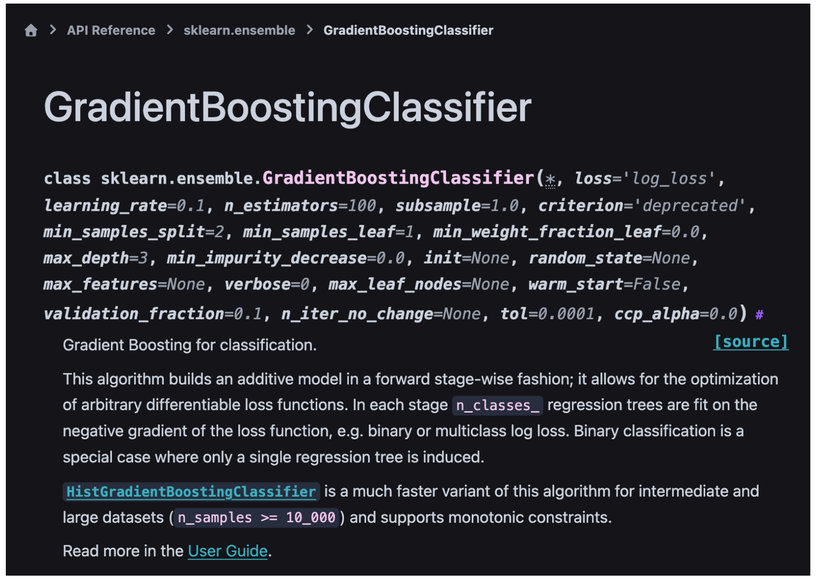
Grid search:
Test every single combination in a predefined set of values!

Random search:
Samples random combinations of parameters!

Bayesian Optimization:
Build a probabilistic model to guess and test hyperparameters!

Typical story

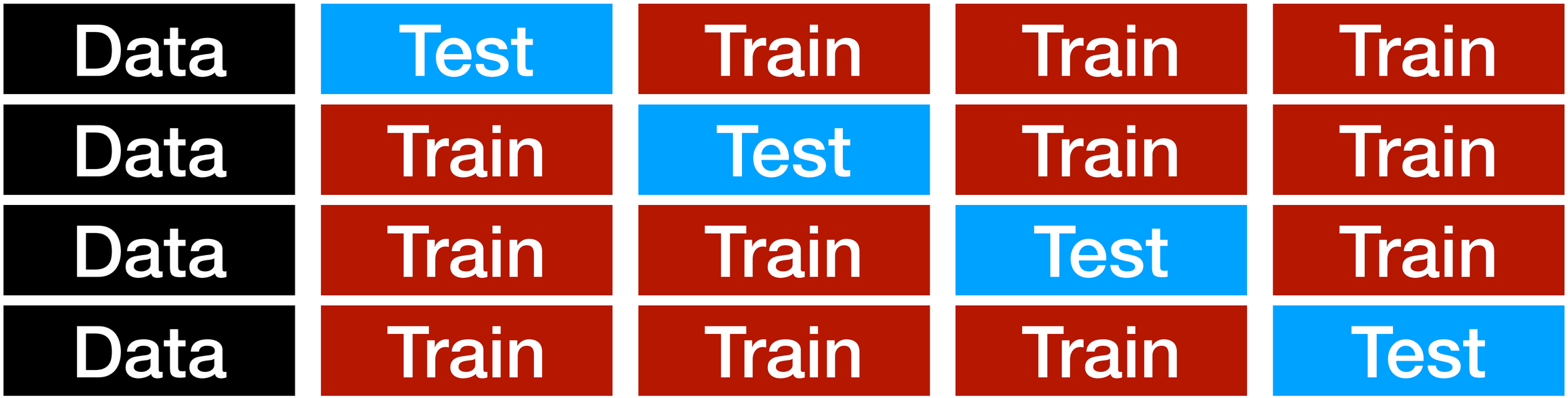
- One does hyperparameter tuning to achieve the best result
- They achieve excellent performance, e.g., 90% accuracy **on the test set**!
- They then apply this model to other data, and performance drops tremendously, e.g., to 80% accuracy



Train-Validation-Test Split

- **Idea**: split the data into three disjoint sets
  - **Training set**: used to fit and train the model
  - **Validation set:** used to evaluate the model during development for hyperparameter tuning
  - **Test Set:** held out completely until the very end of development.
- Still, we cannot peek at the test set too much! Model performance may vary in the real world for other reasons.

k-Fold Cross-Validation



Cross-validation estimates the performance of a training procedure!


Give us some feedback!
https://tinyurl.com/reasoningwithdata