CS57300
PURDUE UNIVERSITY

FEBRUARY 28, 2019

DATA MINING

ANNOUNCEMENTS

- In-class midterm exam
  - 75 minutes (4:30-5:45pm, March 5, WANG 2599)
  - Closed-book, closed-notes
  - Non-programmable calculator is allowed
  - More on this later today

ANNOUNCEMENTS

- Assignment 3
  - For Naive Bayes classifier, please directly use your implementation in Assignment 2 (categorical values are transformed through label encoding), but use set up in Assignment 3 to split training/test set, and take random samples in the cross validation.
  - Learning rate & regularization parameters for Logistic regression (LR) and SVM: **Keep it as is**
  - **Bonus question (+2 points)**: Fine tune the hyperparameters of LR and SVM (i.e., learning rate and regularization parameters) to get the highest possible accuracy on the testSet (recall that you should not touch the testSet until you are satisified with your model). Report your tuning procedure, the hyperparameters you end up with, and the level of accuracy you get on the testSet.

PREDICTIVE MODELING: EVALUATION

WHAT WE'VE LEARNED SO FAR

- We've covered quite a bit of predictive models
  - Naive bayes
  - Decision trees
  - Nearest neighbors
  - Logistic regression
  - SVM
  - Neural networks

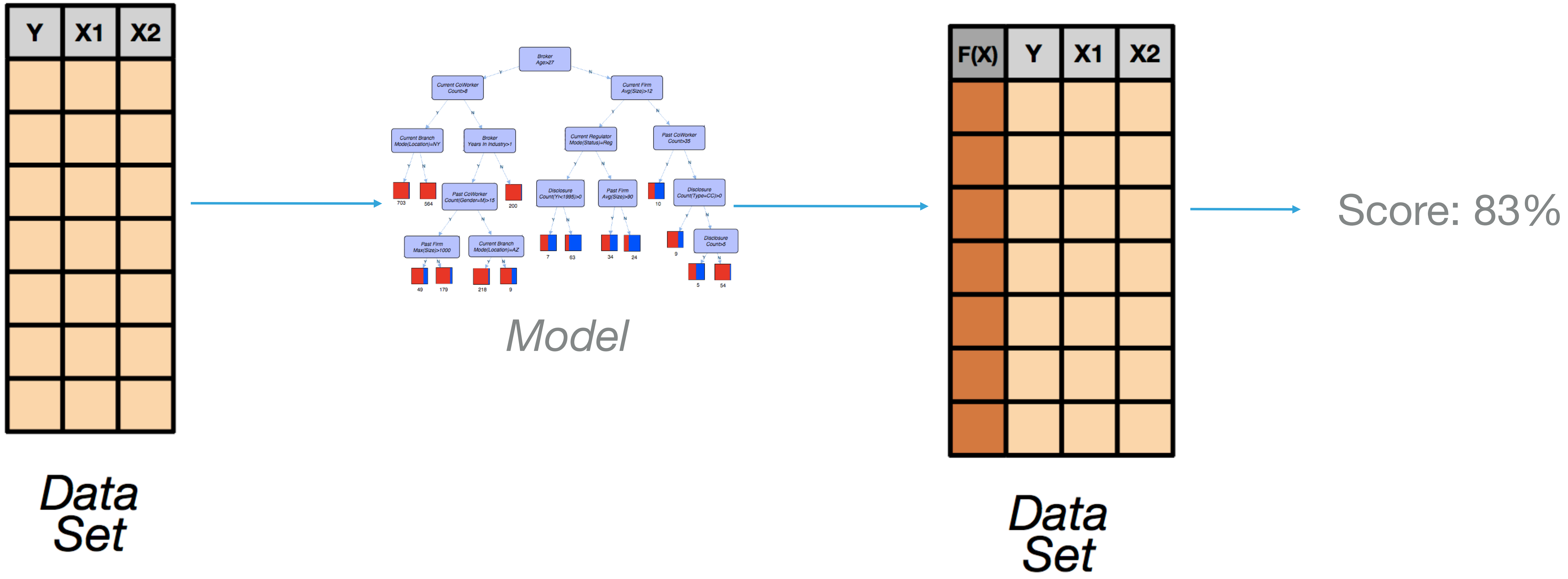
EMPIRICAL EVALUATION

- Given observed accuracy of a model on a limited amount of data, how well does this estimate generalize for additional examples?
- Given that one model outperforms another on some sample of data, how likely is it that this model is more accurate in general?
- When data are limited, what is the best way to use the data to both learn and evaluate a model?

EVALUATING CLASSIFIERS

- Goal: Estimate a classifier's performance on future (unseen) data
- **Approach 1**
  - Use the learned classifier to classify training data and estimate performance

APPROACH 1

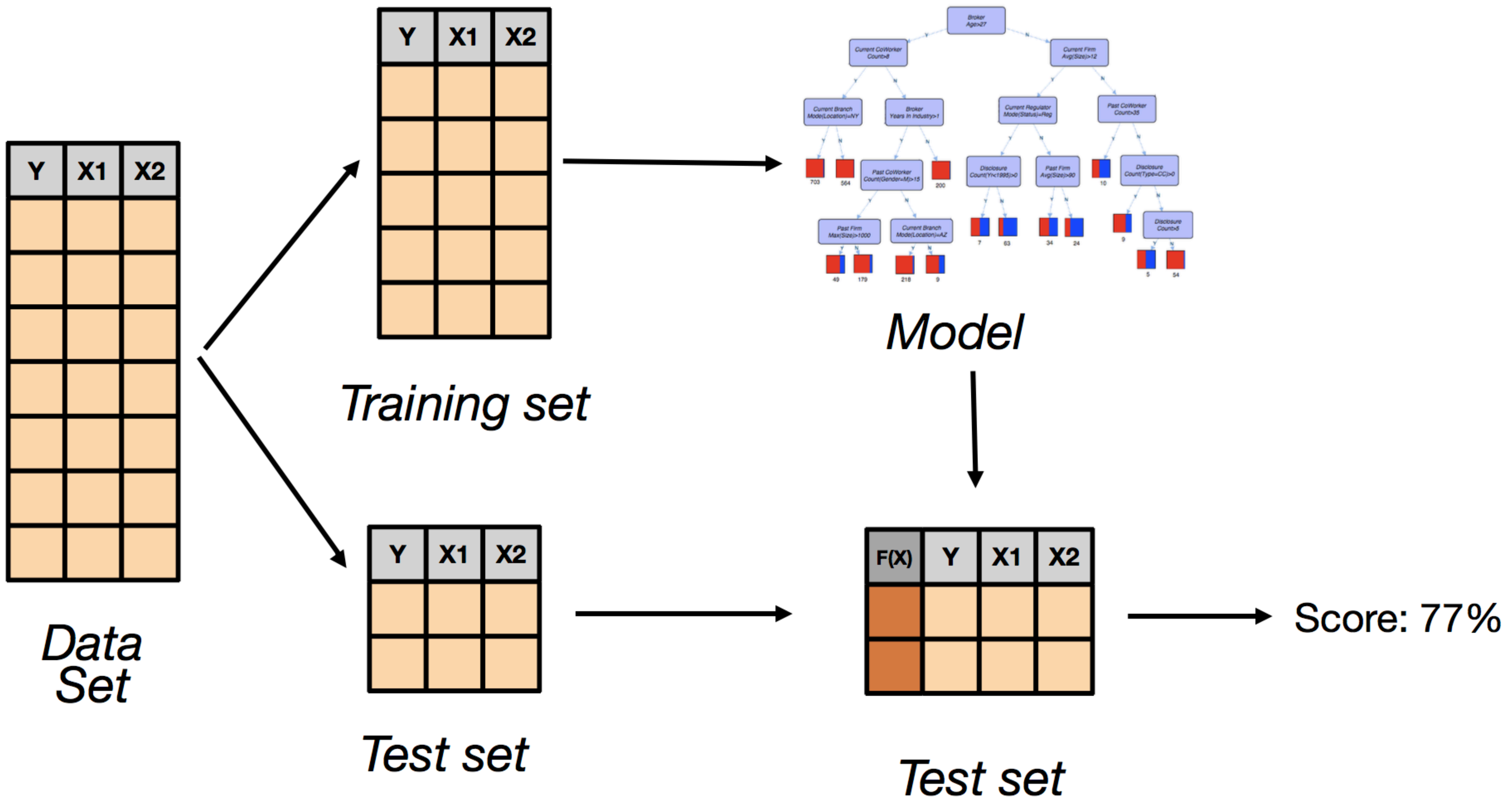


Typically produces a biased estimate of future performance

EVALUATING CLASSIFIERS

- **Approach 2**
  - Classify **disjoint** test set to estimate performance

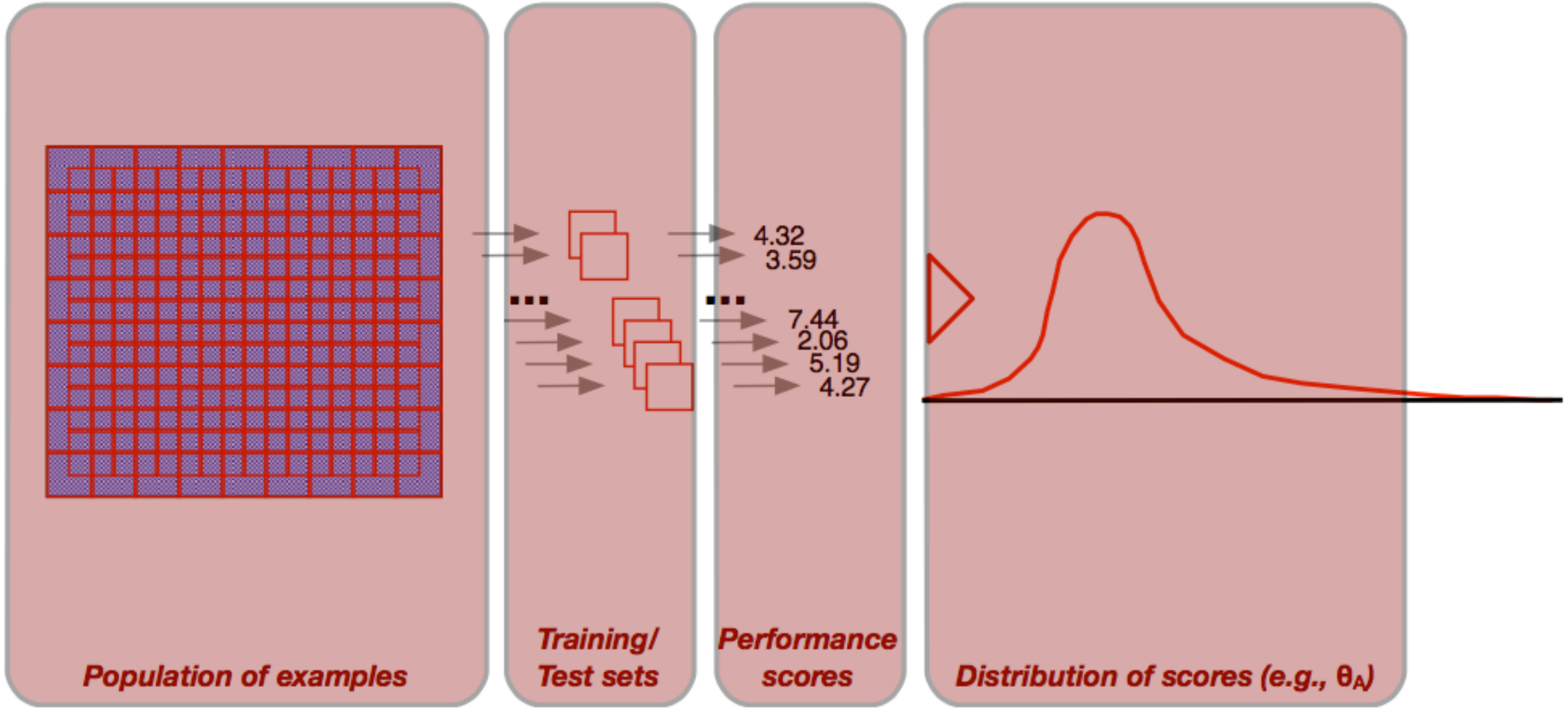
APPROACH 2



✓ An unbiased estimate of future performance

✗ But the estimate will vary due to size and makeup of test set

SAMPLING DISTRIBUTIONS



COMPARING CLASSIFIERS

- Given models A and B, how to decide which model has a better classification performance in general?
- Partition D_0 into two disjoint subsets, learn model on one subset, measure and compare performance on the other subset
- **Problem**: this is a point estimate of the model's performance, i.e., the estimate will vary due to size and makeup of test set

COMPARING CLASSIFIERS

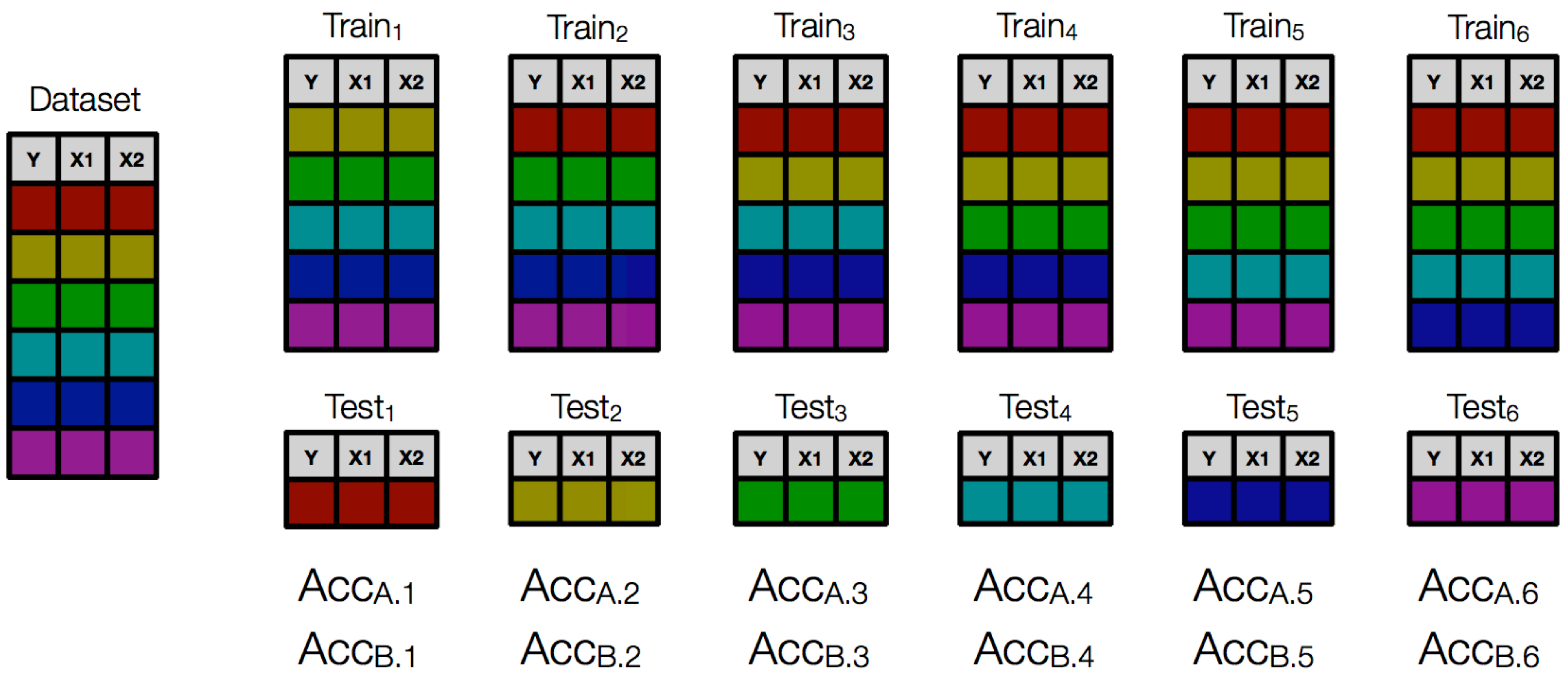
- Repeat Approach 2 for k times, i.e., randomly partition the entire dataset into disjoint training set and test set. Learn the model using the training set and evaluate on the test set.
- Compute the model's average performance over the k trials
- Plot average error and standard error bars
- Any problems?

OVERLAPPING TEST SETS

- Repeated sampling of test sets leads to overlap (i.e., dependence) among test sets… this will result in underestimation of variance
- Standard errors will be biased if performance is estimated from **overlapping** test sets *(Dietterich'98)*
- Recommendation: Use **cross-validation** to eliminate dependence between test sets

COMPARING CLASSIFIERS THROUGH CROSS VALIDATION

- Use k-fold cross-validation to get k estimates of performance for M_A and M_B

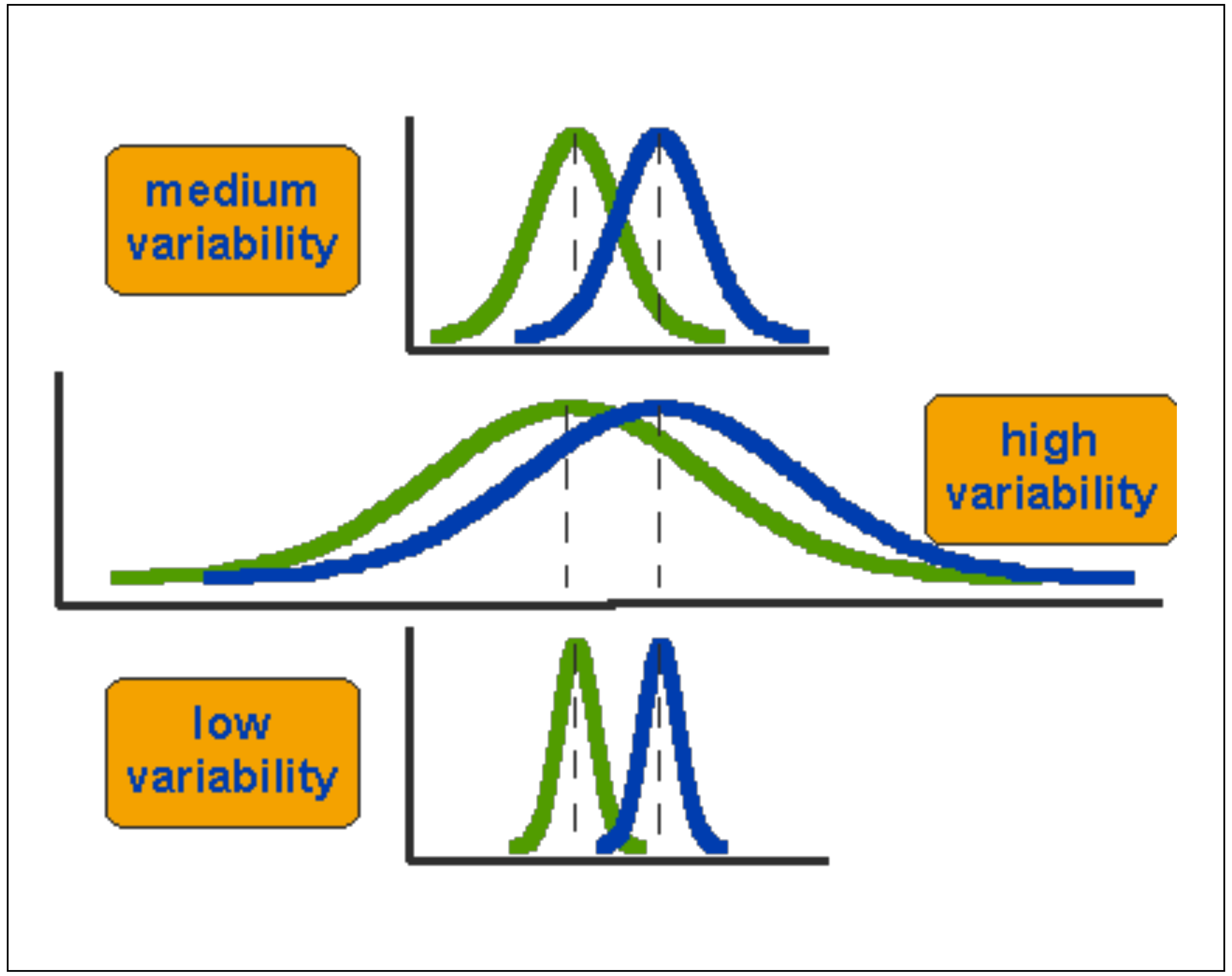


- Set of errors estimated over the test set folds provides empirical estimate of sampling distribution
- Mean is estimate of expected performance

ASSESSING SIGNIFICANCE

- Use **paired t-test** to assess whether the two distributions of errors are statistically different from each other

$Acc_{A.1}$	$Acc_{B.1}$
$Acc_{A.2}$	$Acc_{B.2}$
$Acc_{A.3}$	$Acc_{B.3}$
$Acc_{A.4}$	$Acc_{B.4}$
$Acc_{A.5}$	$Acc_{B.5}$
$Acc_{A.6}$	$Acc_{B.6}$



- Takes into account both the difference in means and the variability of the scores

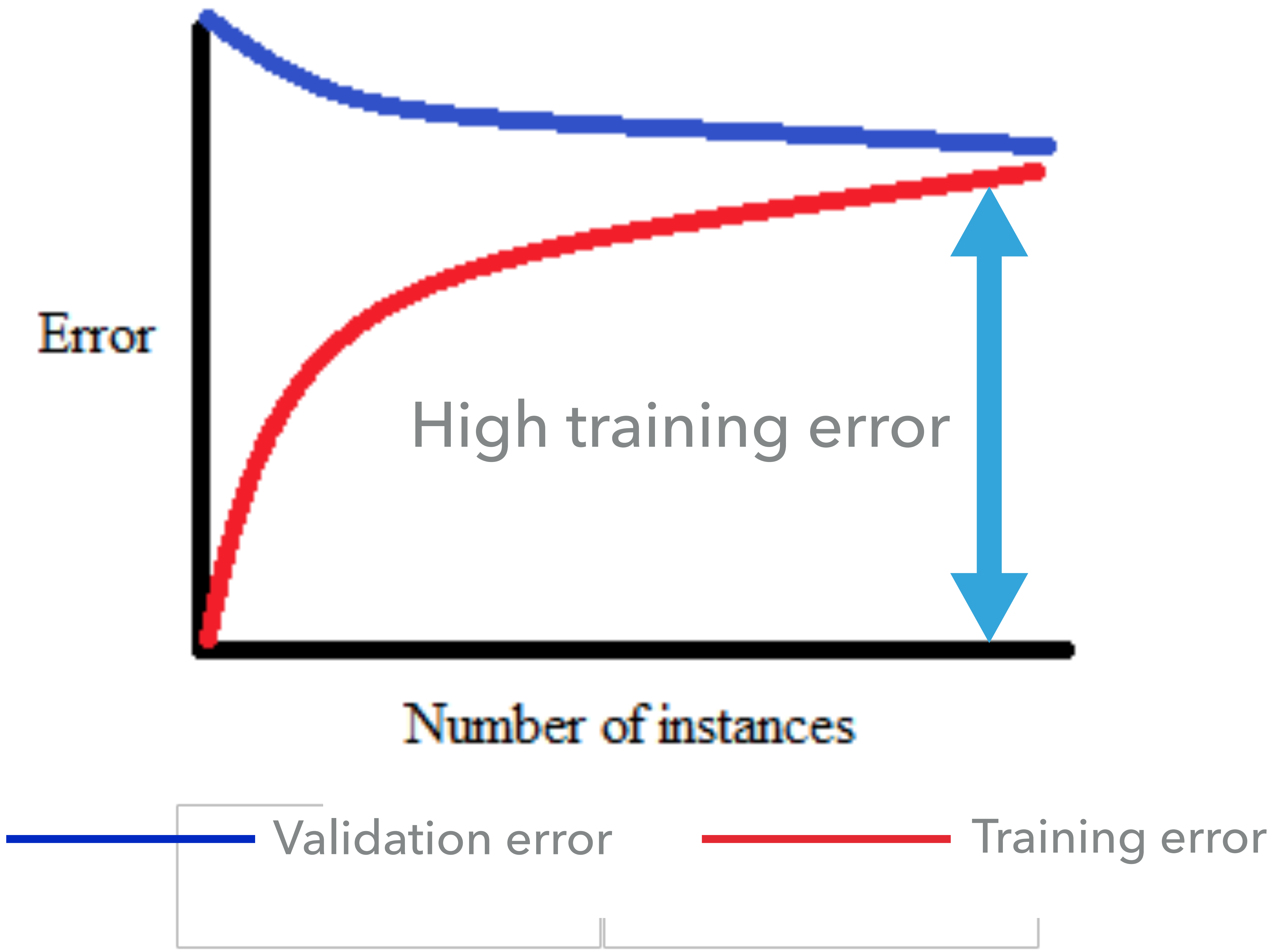
USING CROSS-VALIDATION FOR MODEL SELECTION / TUNING

- Model evaluation
  - Estimate model performance across k-fold cross validation trials
  - Use performance measurement as empirical sampling distribution for model performance
  - Evaluate difference between algorithms with statistical test
- Parameter tuning
  - Decision tree example: Choose threshold for split function with cross validation
    - Repeatedly learn model with different thresholds
    - Pick threshold that shows best cross-validation performance

PLOT LEARNING CURVE

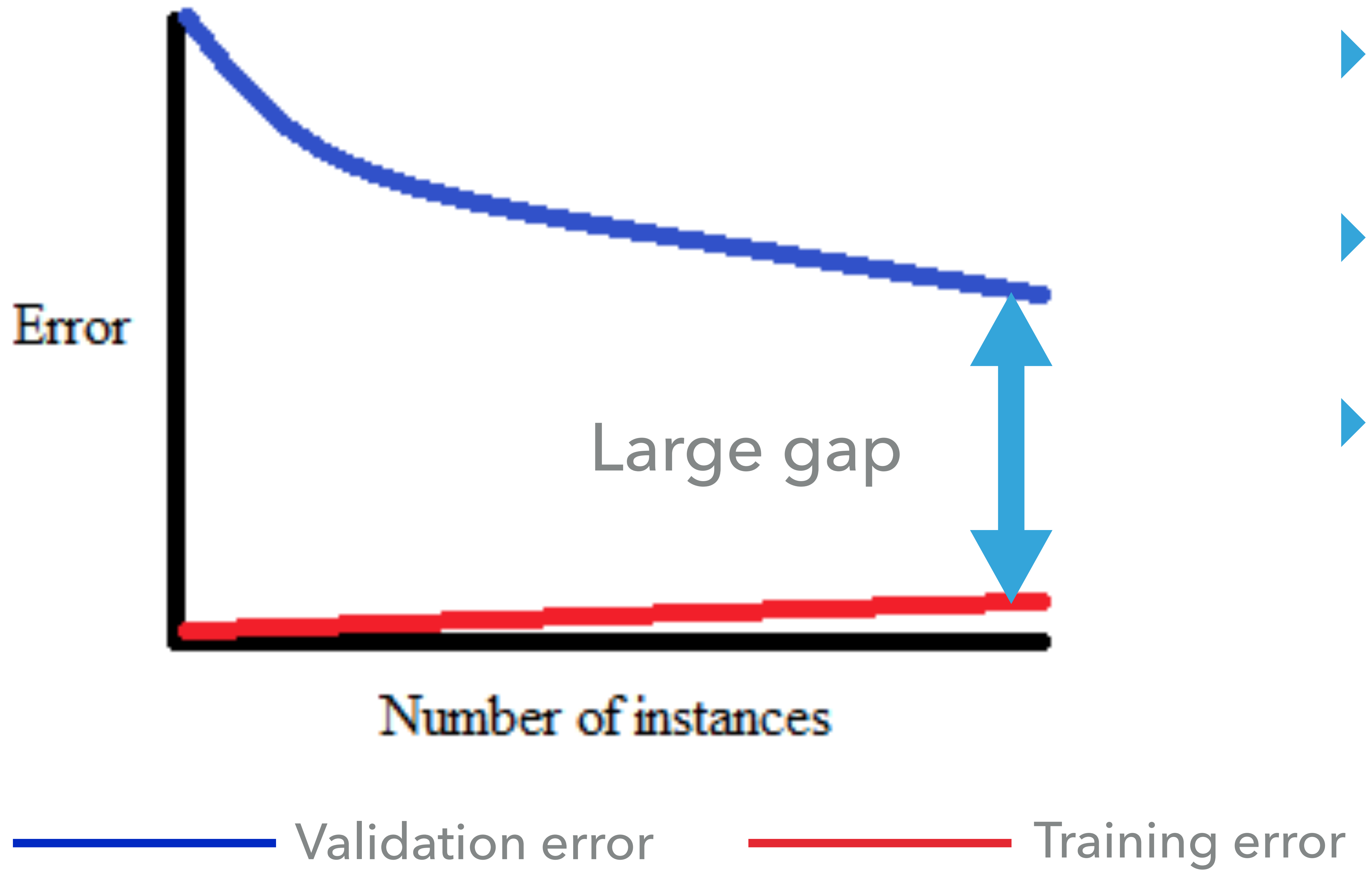
- For a given dataset S, partition it into K folds S_1, S_2, …, S_K
- For frac = [10, 20, …, 100]
  For i = 1:K
  Test set = S_i
  Randomly sample frac% of S_{-i} to construct the training set S_{train}
  Learn model on S_{train} (as a reference, you can estimate the learned model's performance on S_{train}, record it as $perf\_t_{k,\ frac}$)
  Evaluate model's performance on S_i, record it as $perf\_v_{k,\ frac}$
- Plot the training set size vs. model performance
  - Given a specific frac, model's performance is captured by the mean and standard errors of [$perf\_v_{1,\ frac}$, $perf\_v_{2,\ frac}$, …, $perf\_v_{K,\ frac}$]

DETECTING PROBLEMS WITH LEARNING CURVES



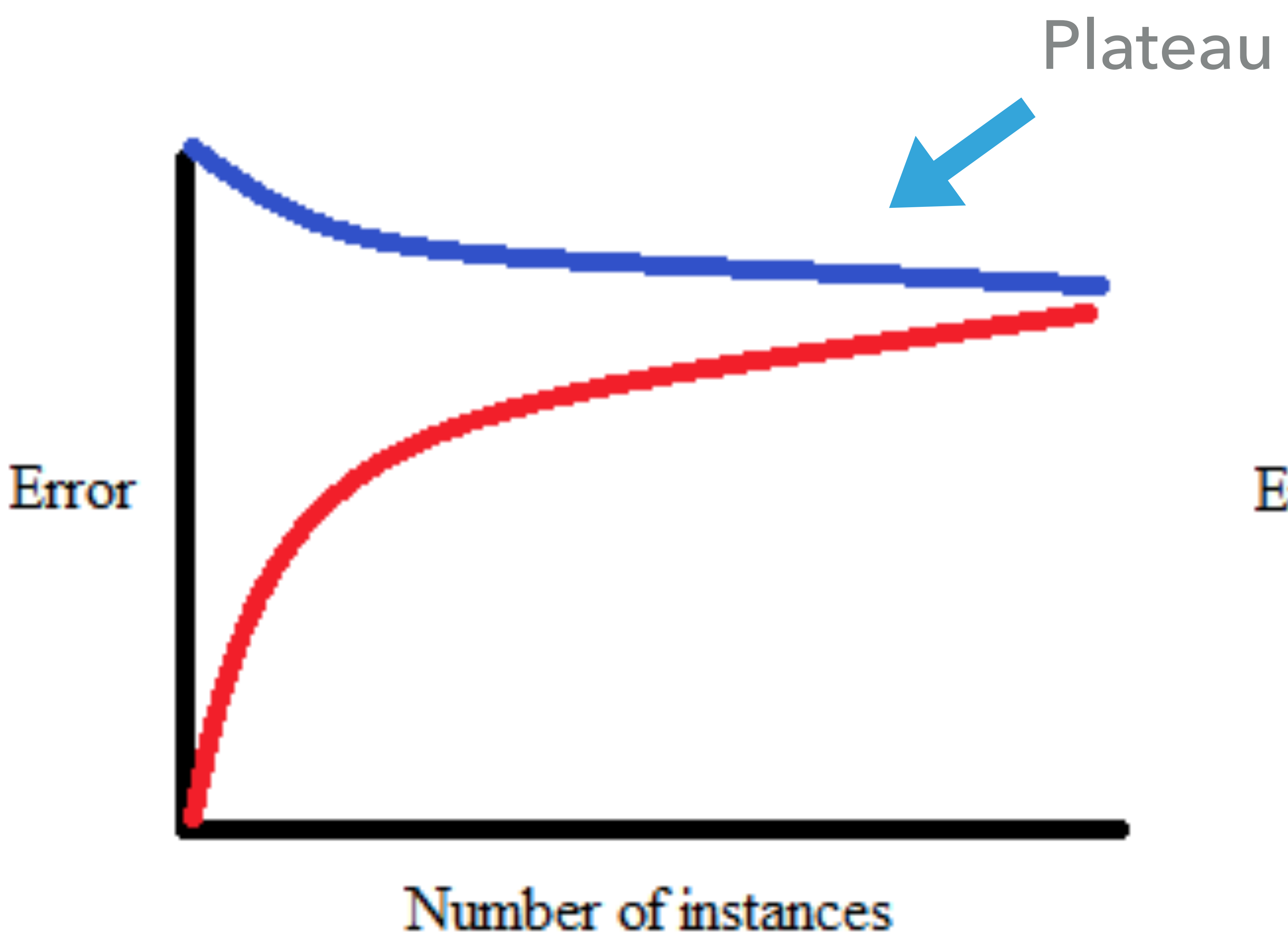
- High bias, low variance
- Underfitting: models are over-simplified

DETECTING PROBLEMS WITH LEARNING CURVES

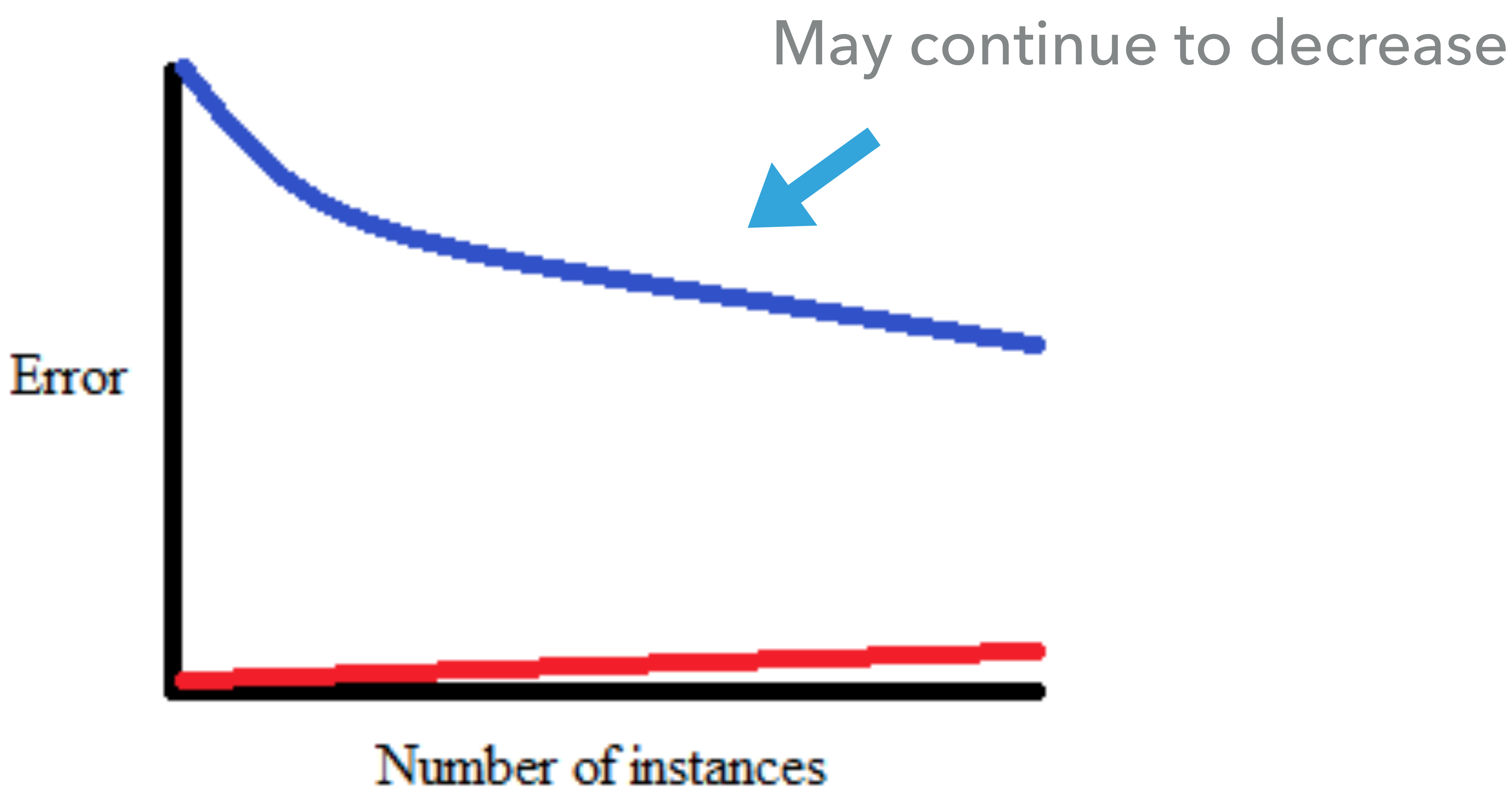


- Low bias, high variance
- Overfitting: models are over-complex
- Consider regularization, adding terms in scoring functions to penalize complexity, etc.

DETECTING PROBLEMS WITH LEARNING CURVE



More training data won't help



More training data may help

BEYOND ACCURACY: CONTINGENCY TABLE SCORE FUNCTIONS

- True positive (**TP**):
  positive prediction that is correct
- True negative (**TN**):
  negative prediction that is correct
- False positive (**FP**):
  positive prediction that is incorrect
- False negative (**FN**):
  negative prediction that is incorrect

		Actual	
		+	–
Predicted	+	TP	FP
	–	FN	TN

BEYOND ACCURACY

- Accuracy = (TP + TN) / (TP + TN + FP + FN) *% predictions that are correct*
- Misclassification = (FP+FN) / (TP+TN+FP+FN) *% predictions that are incorrect*
- Recall/Sensitivity = TP / (TP + FN) *% positive instances that are predicted positive*
- Precision = TP / (TP + FP) *% positive predictions that are correct*
- Specificity = TN / (TN + FP) *% negative instances that are predicted negative*
- F1 = 2 (P·R) / (P+R) *%harmonic mean of precision and recall*

MORE SCORING FUNCTIONS FOR PROBABILISTIC CLASSIFIERS

- Absolute loss: $\frac{1}{n}\sum_{i=1}^{n}|p(y_i = t_i) - 1.0|$ where t is true label
- Squared loss: $\frac{1}{n}\sum_{i=1}^{n}[p(y_i = t_i) - 1.0]^2$ where t is true label
- Likelihood/conditional likelihood: $\prod_{i=1}^{n} p(y_i = t_i)$ where t is true label

ROC CURVES

- Receiver Operating Characteristic (ROC) curve
- Plots the true positive rate (sensitivity) against the false positive rate (1-specificity) for different classification thresholds

P(Y)	True class
0.94	+
0.84	-
0.67	+
0.58	+
0.51	+
0.42	+
0.16	-
0.12	+
0.07	-

P(Y)	True class	Predict class
0.94	+	+
0.84	-	-
0.67	+	-
0.58	-	-
0.51	+	-
0.42	+	-
0.16	-	-
0.1	-	-
0.07	-	-

TPR = 1/4
FPR = 0/5

P(Y)	True class	Predict class
0.94	+	+
0.84	-	+
0.67	+	-
0.58	-	-
0.51	+	-
0.42	+	-
0.16	-	-
0.1	-	-
0.07	-	-

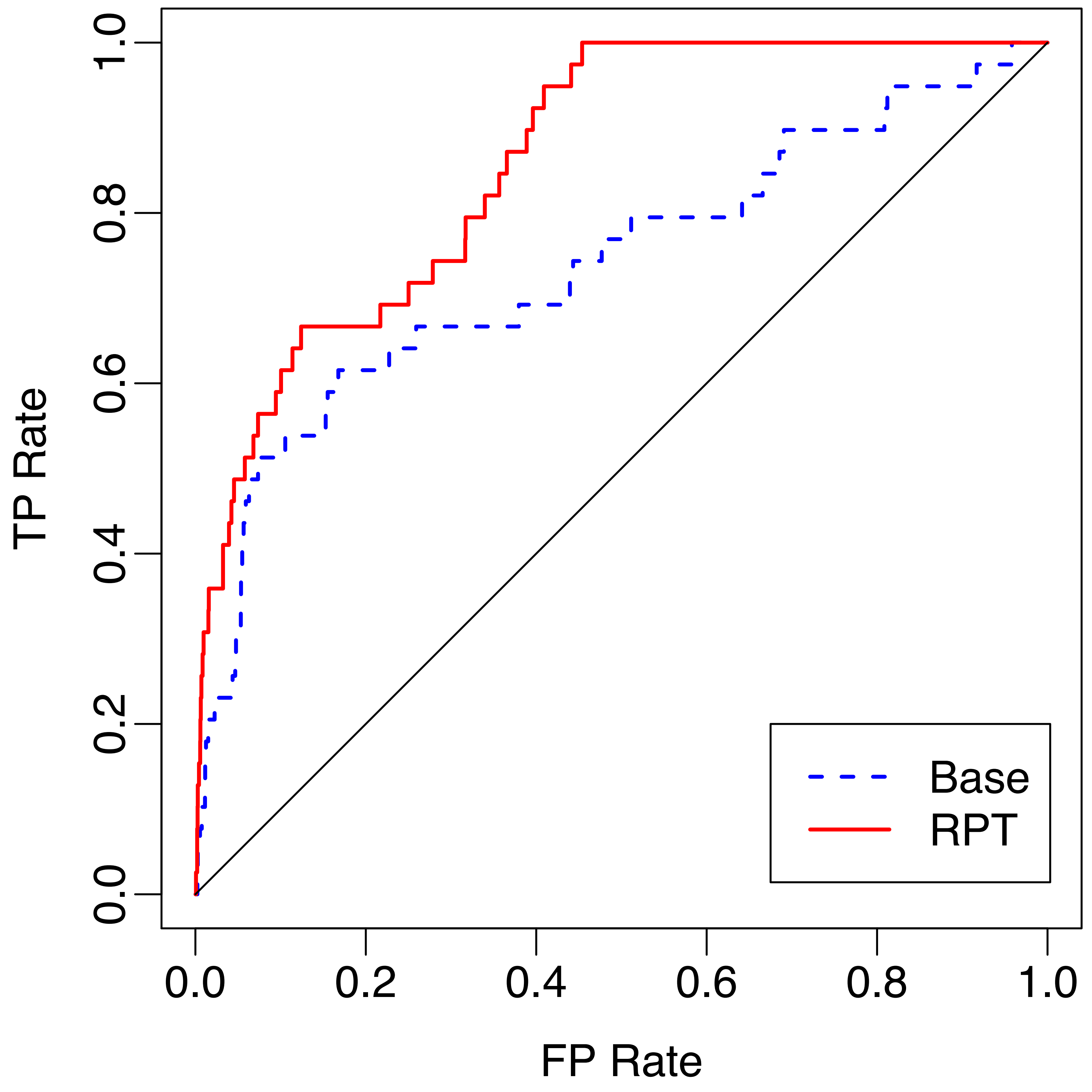
TPR = 1/4
FPR = 1/5

P(Y)	True class	Predict class
0.94	+	+
0.84	-	+
0.67	+	+
0.58	-	-
0.51	+	-
0.42	+	-
0.16	-	-
0.1	-	-
0.07	-	-

TPR = 2/4
FPR = 1/5

AUC

- Evaluates performance over varying costs and class distributions
  - Can summarize with area under the curve (AUC)
  - AUC of 0.5 is random
  - AUC of 1.0 is perfect



MIDTERM REVIEW

EXAM CONTENT & STRUCTURE

- Content
  - All the materials that we covered in class up until now (including today's class)
  - Readings posted on course calendar
- Structure
  - Conceptual multiple choice / true or false / short questions
  - Long questions testing your understanding of specific data mining algorithms

EXAM TOPICS

- Math review (probability and statistics, linear algebra, sampling, hypothesis testing, etc.)
- Elements of data mining algorithms
- Exploratory data analysis (data visualization, dimensionality reduction)
- Predictive modeling
  - Naive Bayes, decision trees, nearest neighbors, logistic regression, SVM, perceptron, neural networks…
  - Optimization
  - Evaluation

PREDICTIVE MODELING

- For each type of predictive model:
  - What's the knowledge representation?
  - How to learn the model? (What is the model space? What is the scoring function? What is the search algorithm?)
  - Special issues (How to deal with categorical/continuous variables? How to deal with overfitting? etc.)

GOOD LUCK!