Model Evaluation

Sachin Tripathi

IIT(ISM), Dhanbad

Topics to be covered

- Confusion Matrix
- Metrics for Regression
- Cross Validation

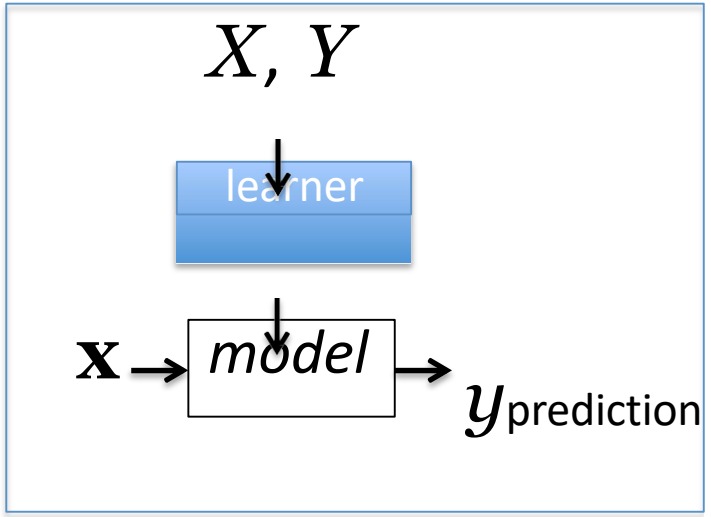
Stages of (Batch) Machine Learning

Given: labeled training data $X, Y = \{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n}$

- Assumes each $\boldsymbol{x}_i \leftarrow D(X)$ with $y_i = ftarget(\boldsymbol{x}_i)$

Train the model:

$model \leftarrow classifier.\text{train}(X, Y)$



Apply the model to new data:

- Given: new unlabeled instance $\boldsymbol{x} \leftarrow D(X)$

$y_{\text{prediction}} \leftarrow model.\text{predict}(\mathbf{x})$

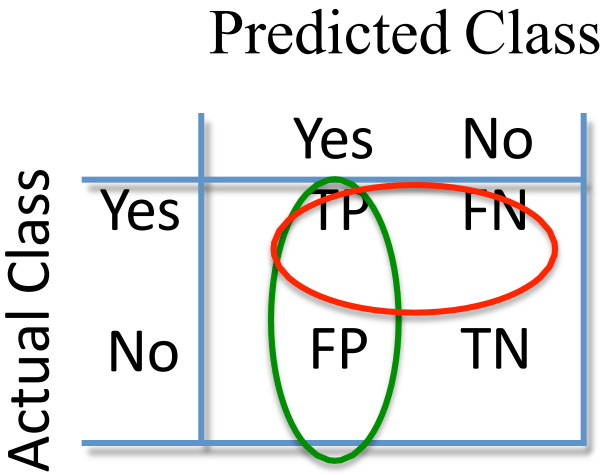
Classification Metrics

- Accuracy = $\frac{\#\ \text{correct predictions}}{\#\ \text{test instances}}$

$$\text{Error} = 1 - \text{Accuracy} = \frac{\#\ \text{incorrect predictions}}{\#\ \text{test instances}}$$

Confusion Matrix

- Given a dataset of P positive instances and N negative instances:



$$\text{accuracy} = \frac{TP + TN}{P + N}$$

- Imagine using classifier to identify positive cases (i.e., for information retrieval)

$$\text{Precision} = \frac{TP}{TP + FP}$$

Probability that a randomly selected result is relevant

$$\text{Recall} = \frac{TP}{TP + FN}$$

Probability that a randomly selected relevant document is retrieved

❑ F1 Score

Ideally in a good classifier, we want both precision and recall to be one which also means FP and FN are zero. Therefore we need a metric that takes into account both precision and recall. F1-score is a metric which takes into account both precision and recall.

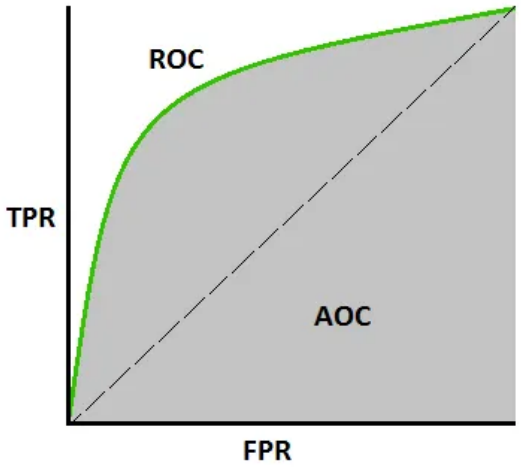
F1 Score becomes 1 only when precision and recall are both 1. F1 score becomes high only when both precision and recall are high. F1 score is the harmonic mean of precision and recall and is a better measure than accuracy.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

AUC -ROC Curve

What is the AUC - ROC Curve?

- AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability.
- It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.
- By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.
- The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.



Defining terms used in AUC and ROC Curve.

TPR (True Positive Rate) / Recall /Sensitivity

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

Specificity

$$\text{Specificity} = \frac{TN}{TN + FP}$$

FPR

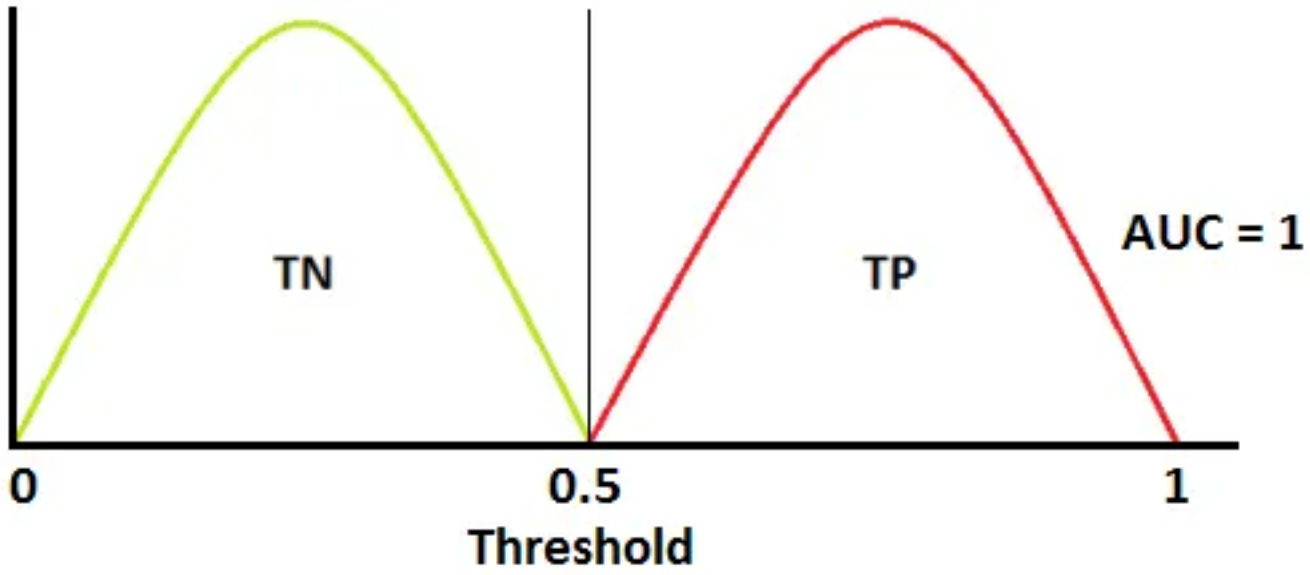
$$\text{FPR} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP}$$

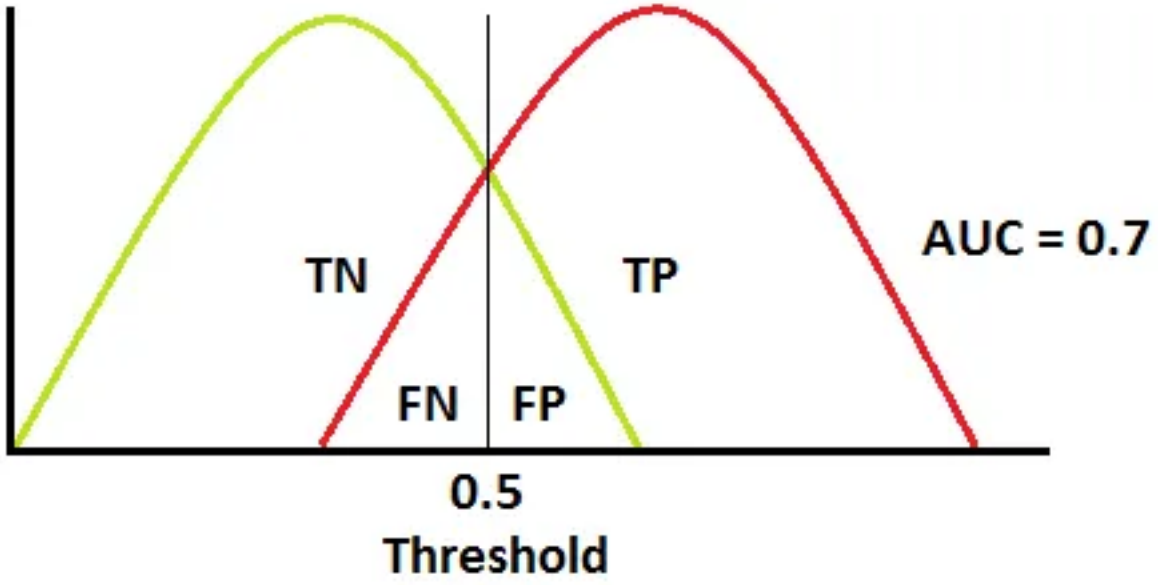
How to speculate about the performance of the model

- An excellent model has AUC near to the 1 which means it has a good measure of separability.
- A poor model has an AUC near 0 which means it has the worst measure of separability. In fact, it means it is reciprocating the result.
- It is predicting 0s as 1s and 1s as 0s.
- And when AUC is 0.5, it means the model has no class separation capacity whatsoever.

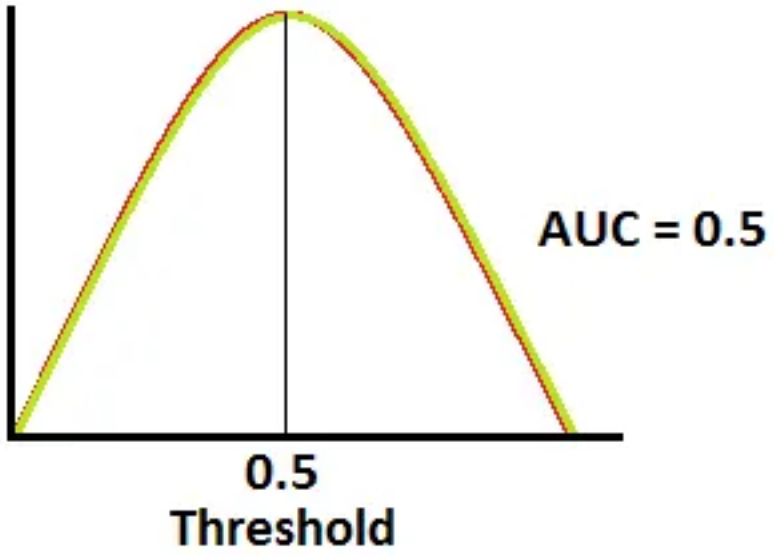
Note: Red distribution curve is of the positive class and the green distribution curve is of the negative class.



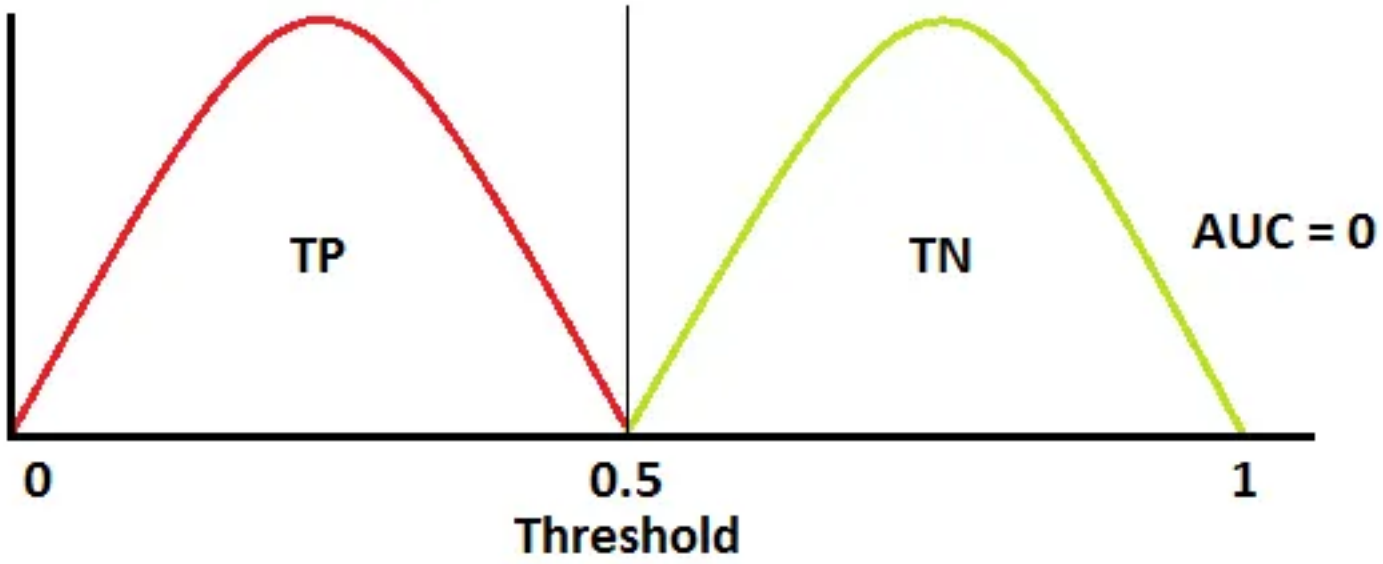
- This is an ideal situation.
- When two curves don’t overlap at all means model has an ideal measure of separability.
- It is perfectly able to distinguish between positive class and negative class.



- When two distributions overlap, we introduce type 1 and type 2 errors.
- Depending upon the threshold, we can minimize or maximize them.
- When AUC is 0.7, it means there is a 70% chance that the model will be able to distinguish between positive class and negative class.



- This is the worst situation.
- When AUC is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class.



- When AUC is approximately 0, the model is actually reciprocating the classes.
- It means the model is predicting a negative class as a positive class and vice versa.

Evaluation Metrics for Regression Models

- Regression models and techniques are extremely popular in Machine Learning across several industries.
- These models are efficient in accomplishing several tasks, such as:
- Estimate the price value of houses, cars, tech products, and others;
- Determine the optimal drug dosages based on the characteristics of a patient.
- Estimate the future demand for transportation services;
- Predict future sales based on historical data, events, and market trends.

Mean Absolute Error:-

- The Mean Absolute Error gives us the average value of the tota absolute differences between the predicted values output by the model and the actual values in the dataset.
- It is expressed in the same unit scale as the data measured, which makes it a straightforward metric to interpret.
- Values closer to 0 are considered better.

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

Mean Squared Error

- The formula for computing the Mean Square Error is very similar to the one we use to compute the Mean Absolute Error.
- This time, however, we square the differences between actual and predicted values fo Y.
- By squaring the differences, we penalize larger errors more than smaller errors, makin it an ideal choice to evaluate models for tasks in which larger errors may lead to undesirable outcomes.
- The closer to 0, the better the model's performance.

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

Root Mean Squared Error

- By using the Root Mean Squared Error we are able to bring the Mean Square Error bac to the same unit scale as the observed data, which makes it more intuitive.
- It still holds the same characteristics as the Mean Squared Error, penalizing larger errors more than smaller errors.
- The closer to 0, the better the model's performance.

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

Coefficient of Determination (R^2)

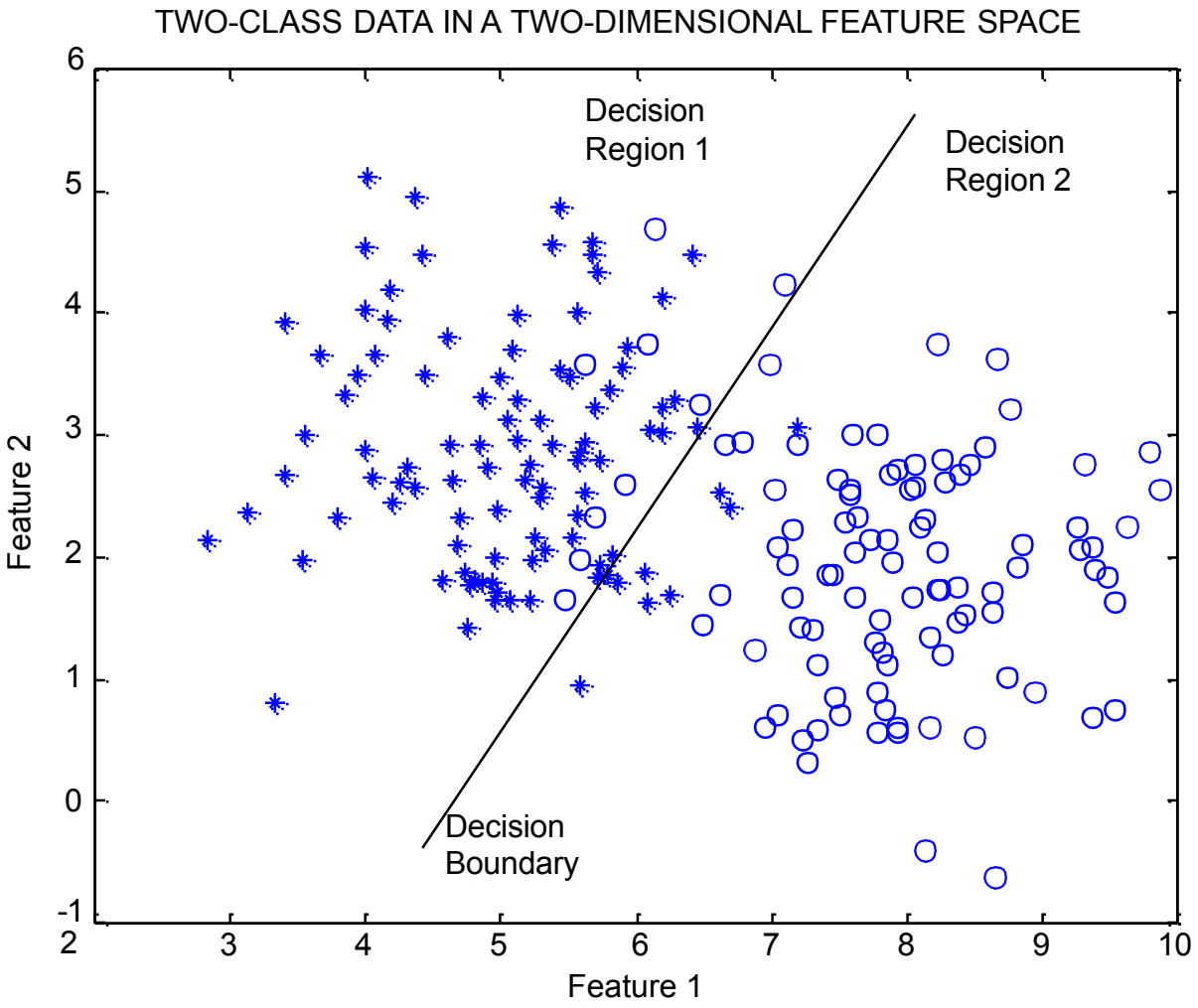
- The Coefficient of Determination — also referred to as R-Squared — is a measure that tells us how well a regression model fits the actual data.
- It quantifies the degree to which the variance in the dependent variable is predictable from the independent variables.
- Values closer to 1.0 indicate a better model.

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

Training Data and Test Data

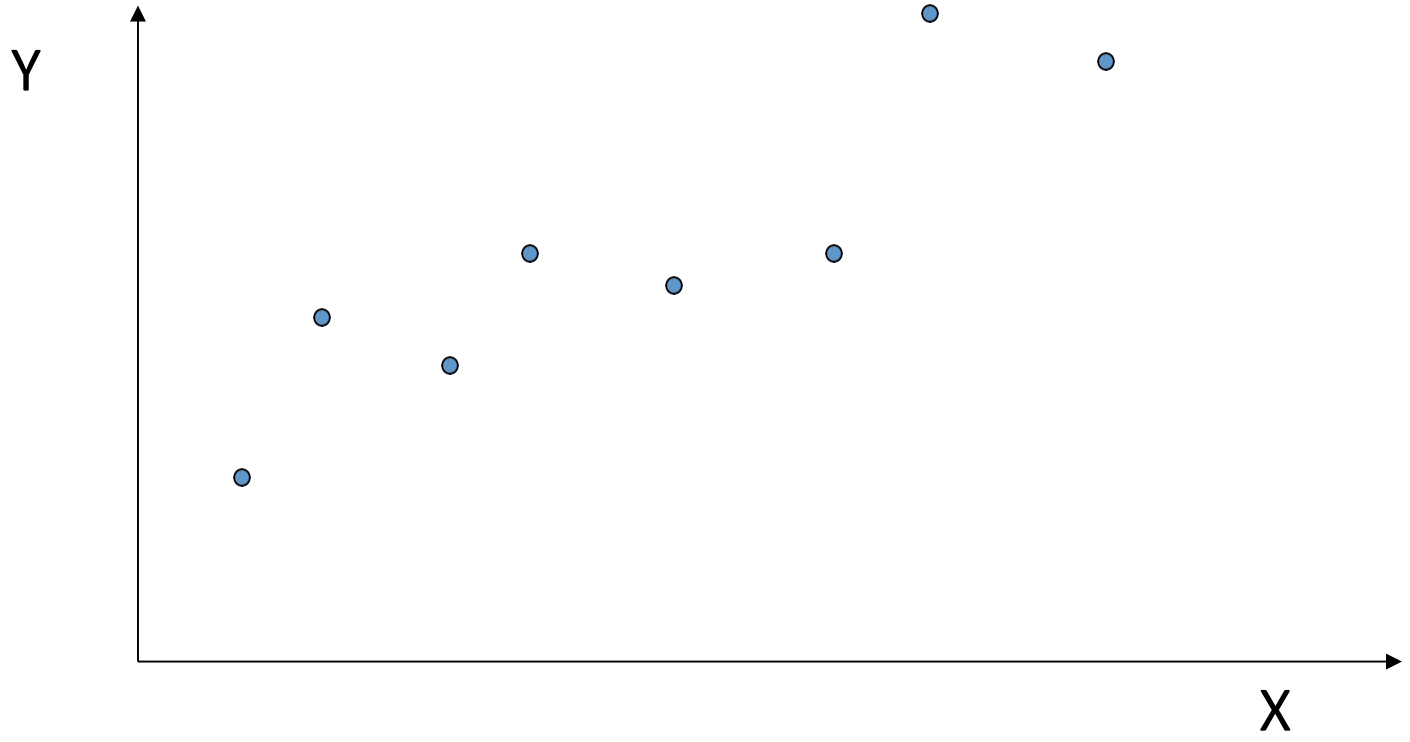
- Training data: data used to build the model
- Test data: new data, not used in the training process
- Training performance is often a poor indicator of generalization performance
  - Generalization is what we really care about in ML
  - Easy to overfit the training data
  - Performance on test data is a good indicator of generalization performance
    - i.e., test accuracy is more important than training accuracy

Simple Decision Boundary



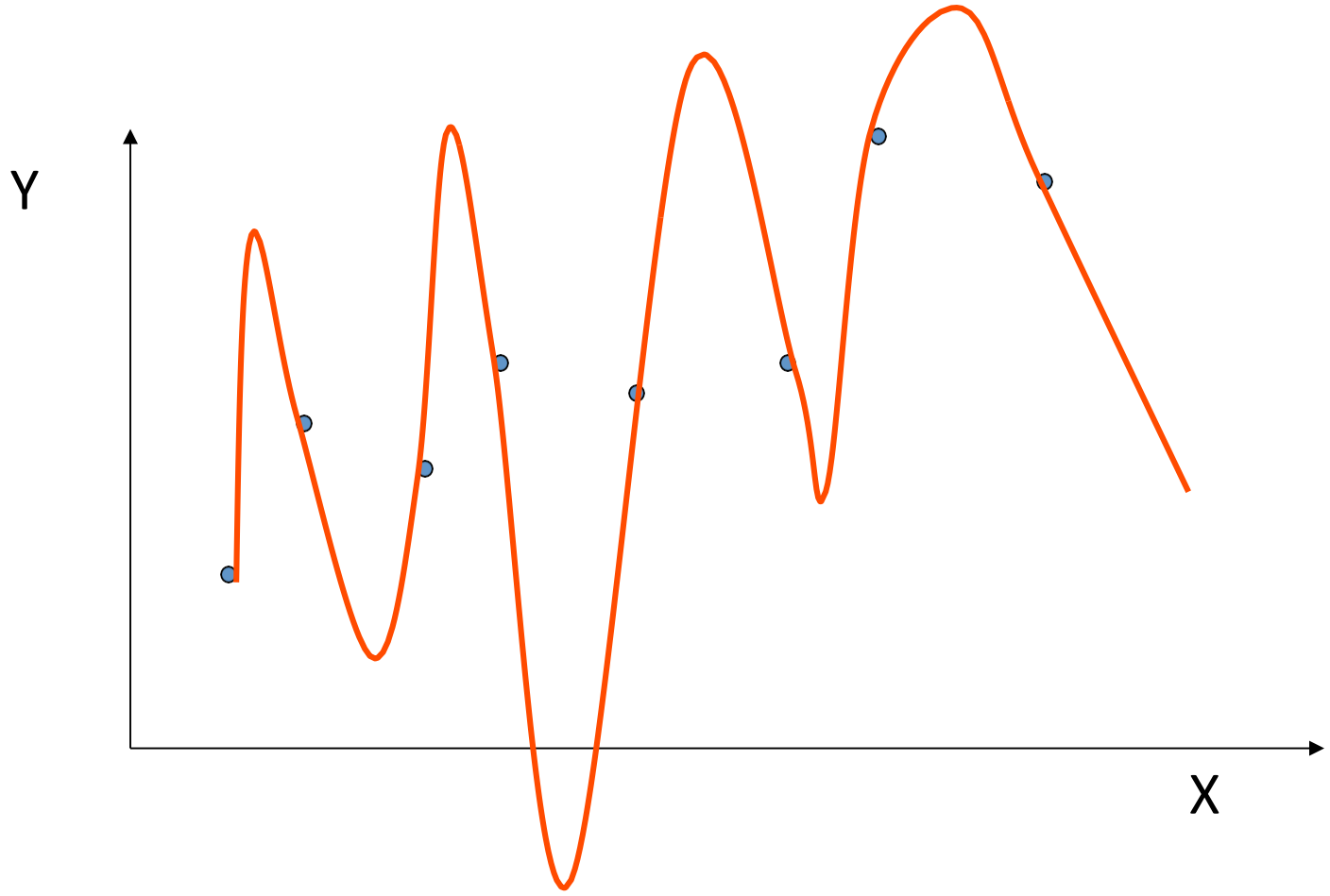
More Complex Decision Boundary

Example: The Overfitting Phenomenon

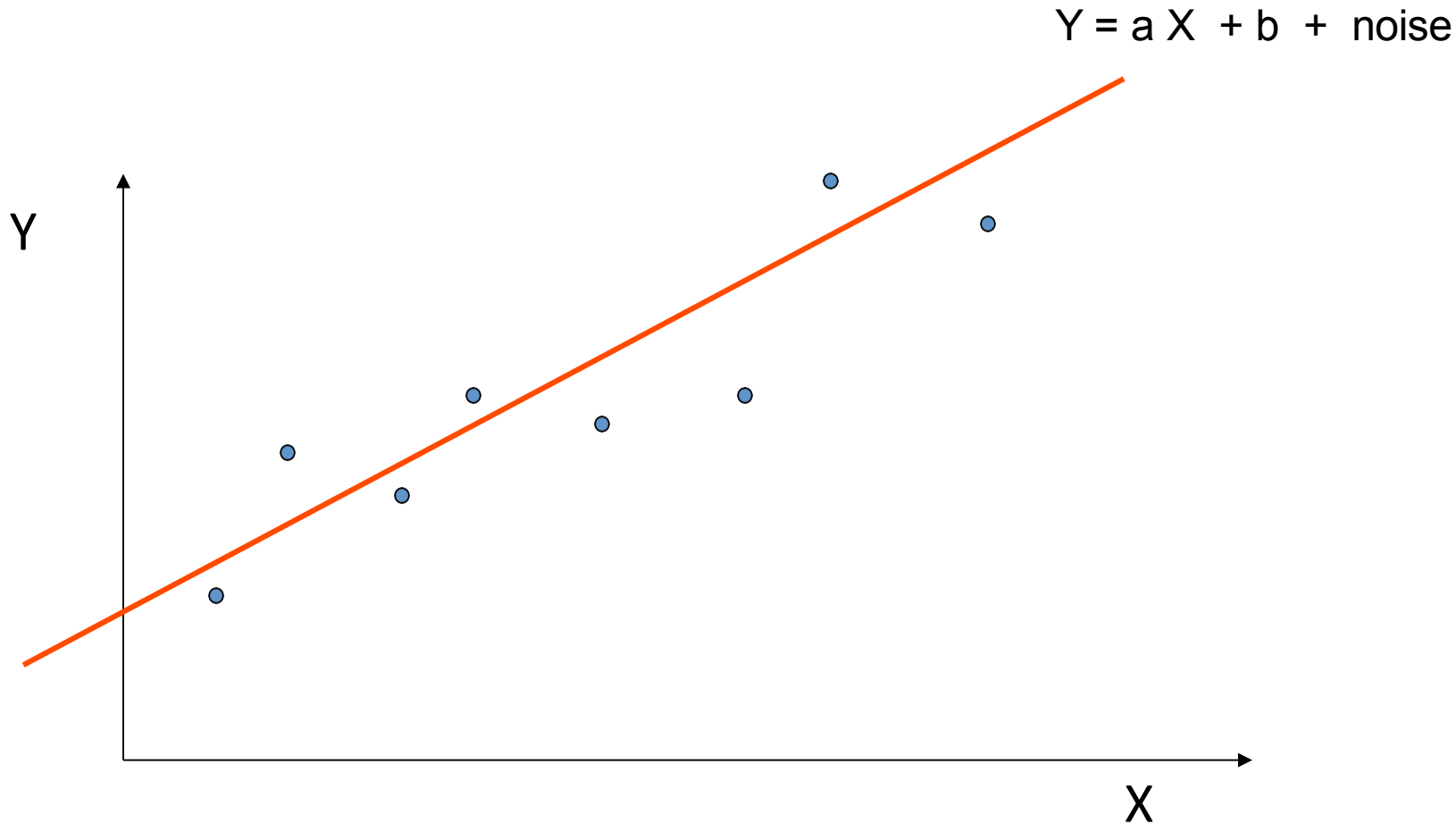


A Complex Model

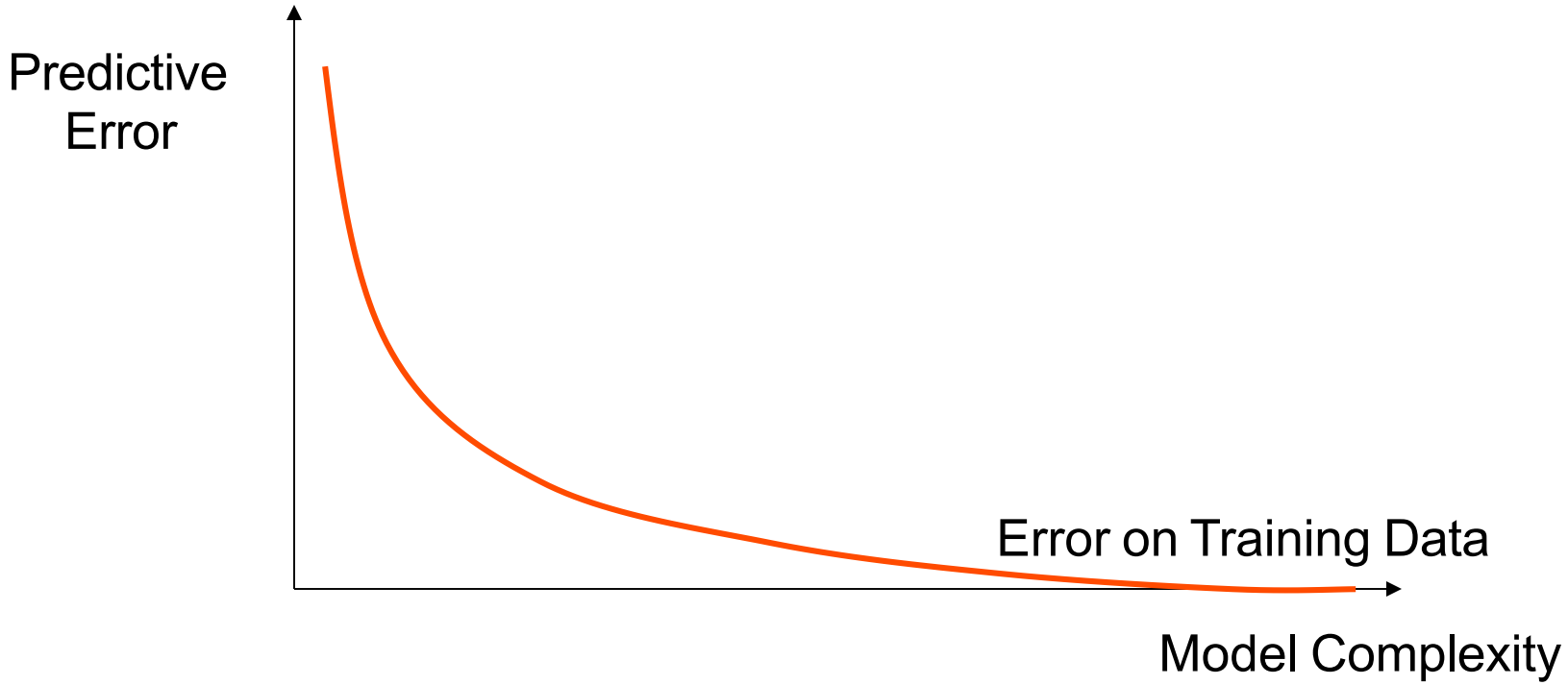
Y = high-order polynomial in X



The True (simpler) Model



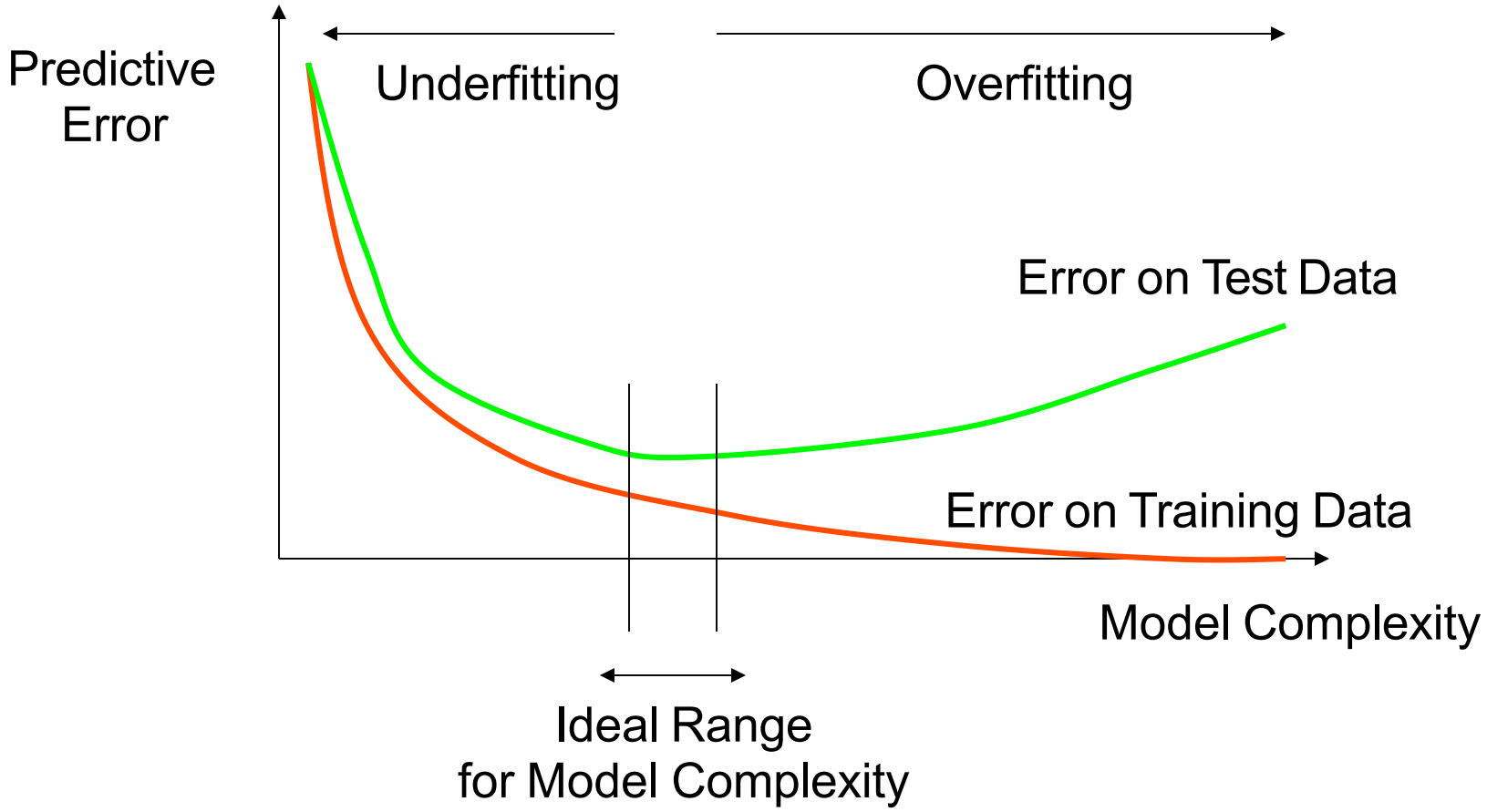
How Overfitting Affects Prediction



Slide by Padhraic Smyth, UCIrvine

How Overfitting Affects Prediction

How Overfitting Affects Prediction



Comparing Classifiers

- Say we have two classifiers, C1 and C2, and want to choose the best one to use for future predictions

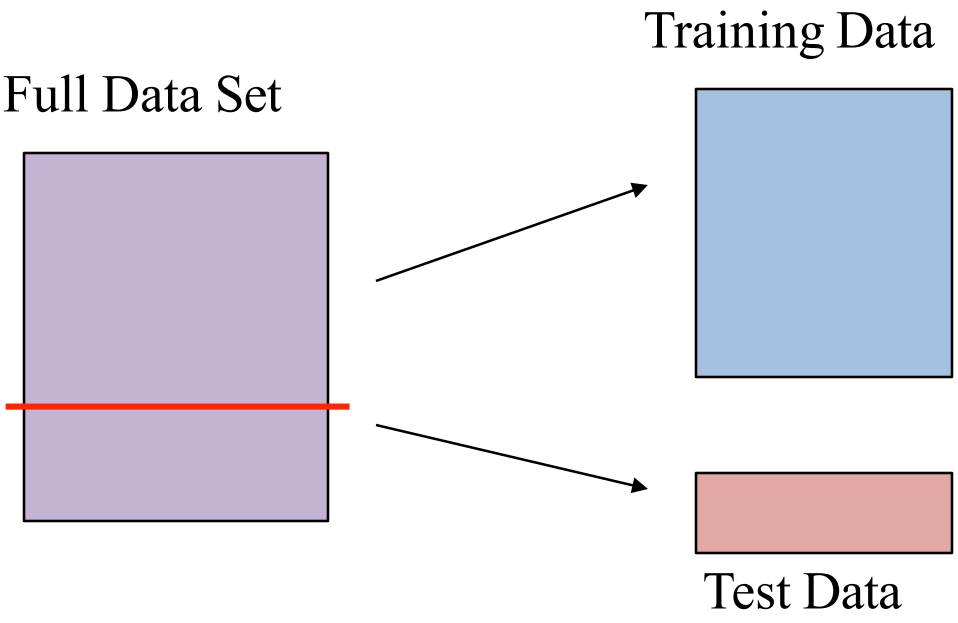
- Can we use training accuracy to choose between them?

- No!
  - e.g., C1 = pruned decision tree, C2 = 1–NN

    training_accuracy(1--NN) = 100%, but may not be best

Instead, choose based on test accuracy...

Training and Test Data



Idea:

Train each model on the "training data"...

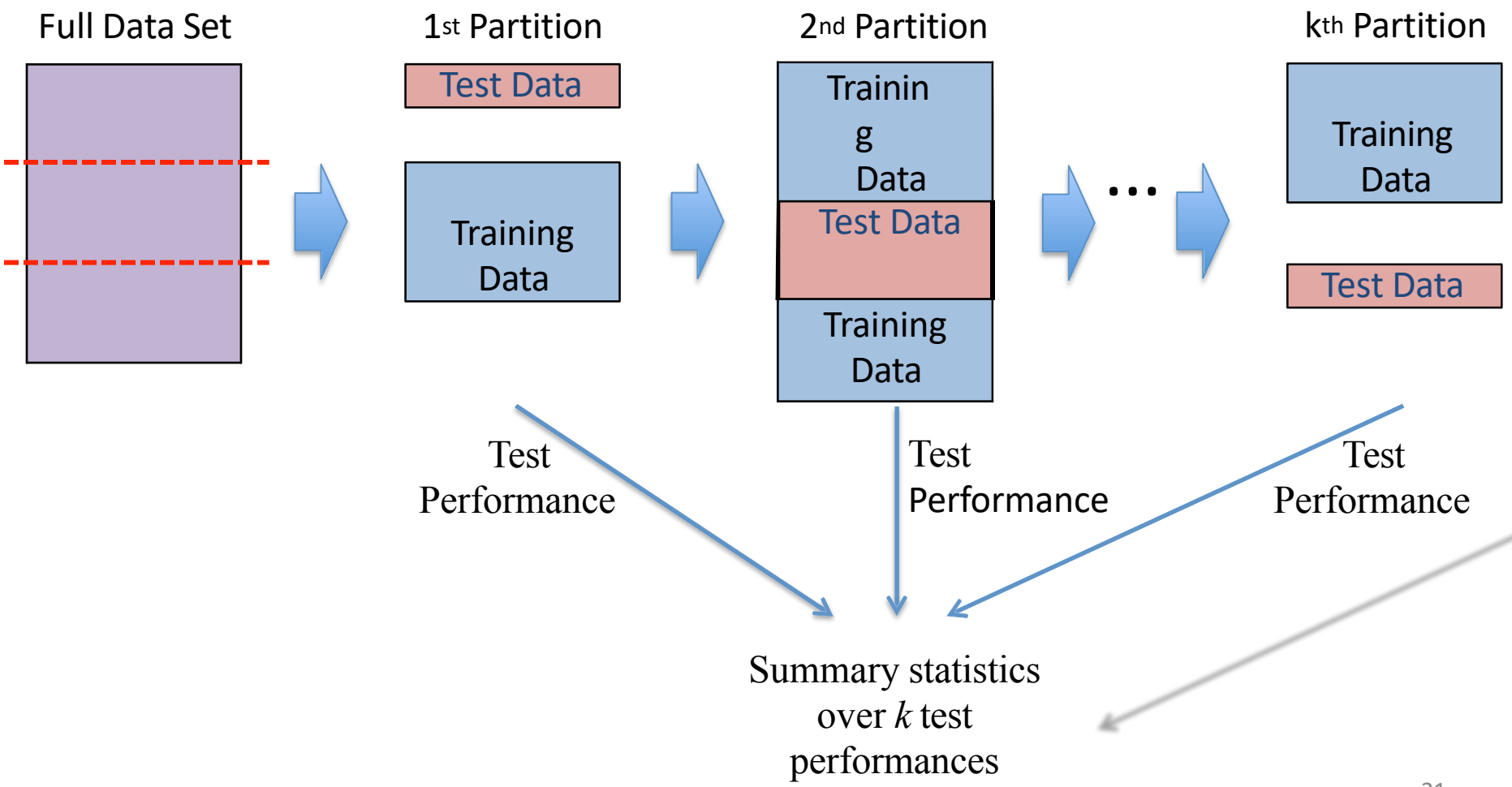
…and then test each model's accuracy on the test data

k-Fold Cross-Validation

- Why just choose one particular "split" of the data?
  - ➢ In principle, we should do this multiple times since performance may be different for each split

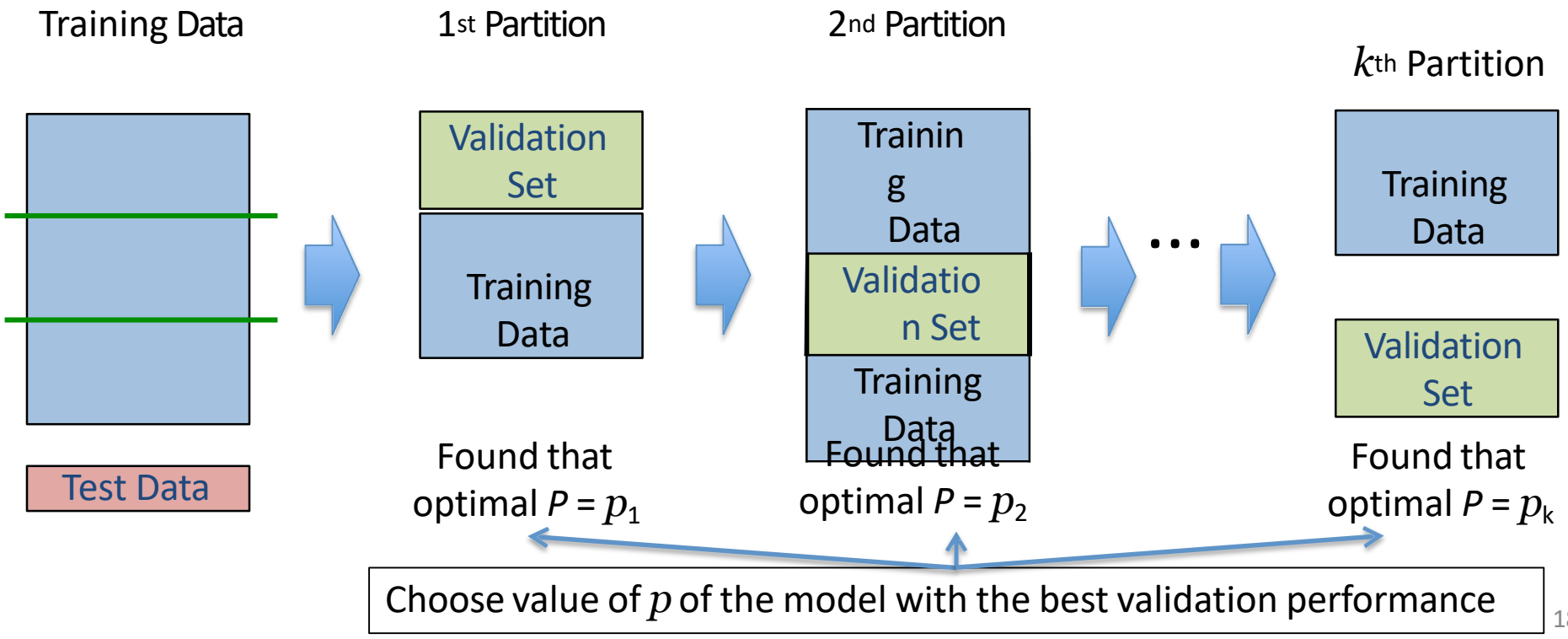
- *k*-Fold Cross-Validation (e.g., $k=10$)
  - ➢ Randomly partition full data set of n instances into k disjoint subsets (each roughly of size n/k)
  - ➢ Choose each fold in turn as the test set; train model on the other folds and evaluate
  - ➢ Compute statistics over k test performances, or choose best of the k models
  - ➢ Can also do "leave-one-out CV" where $k = n$

Example 3-Fold CV



Optimizing Model Parameters

- Can also use CV to choose value of model parameter P
  - Search over space of parameter values $p \in values(P)$
  - Evaluate model with $P = p$ on validation set
  - Choose value p' with highest validation performance
  - Learn model on full training set with $P = p$'



More on Cross--Validation

- Cross--validation generates an approximate estimate of how well the classifier will do on "unseen" data
  - As $k \to n$, the model becomes more accurate (more training data)
  - ...but, CV becomes more computationally expensive
  - Choosing $k < n$ is a compromise

- Averaging over different partitions is more robust than just a single train/validate partition of the data

- It is an even better idea to do CV repeatedly!

Thank You