Model Evaluation

Cross Validation, underfitting vs overfitting, learning curves

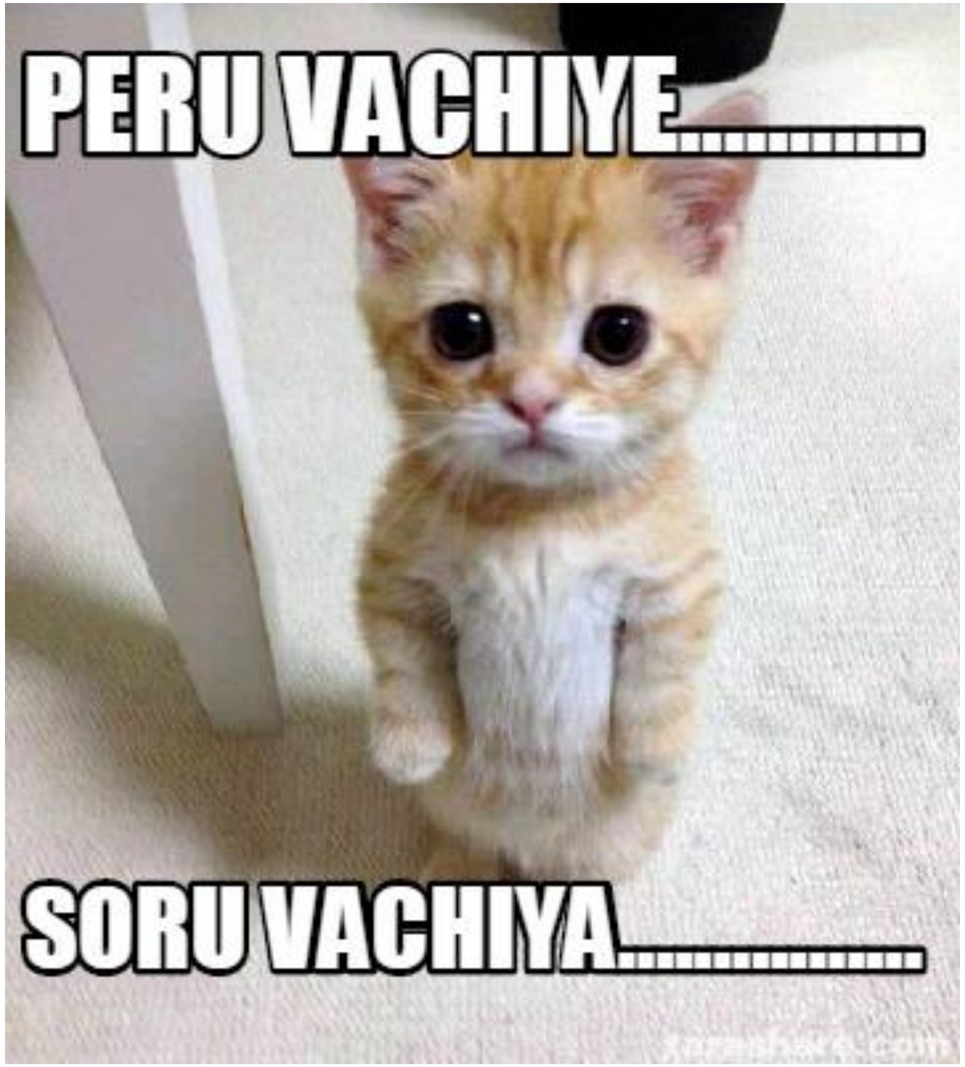
TRENDSTAR

THANKS FOR 1000 SUBSCRIBERS ON YouTube


Machi... Machiiiii......


Majaa Matrix Machi......

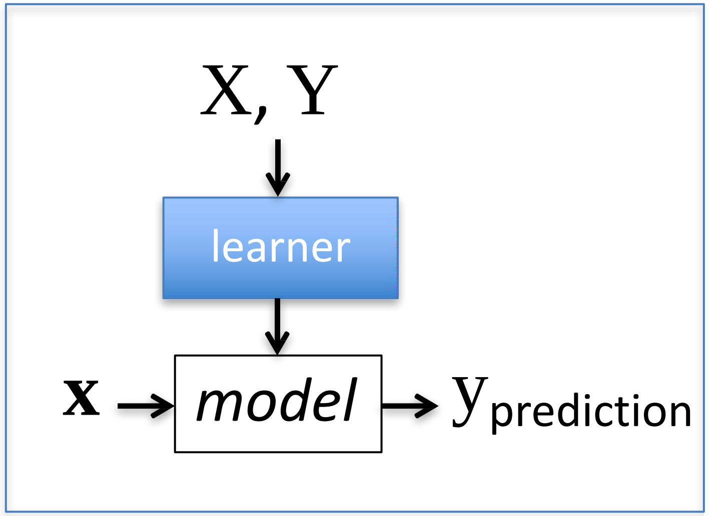
Majaa Matrix from now onwards…



Stages of 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 $y_i = f_{target}(\boldsymbol{x}_i)$



Train the model:

model ← *classifier*.train(X, Y)

Apply the model to new data:

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

$y_{prediction}$ ← *model*.predict($\mathbf{x}$)

Metrics

Regression	Classification
• Mean Absolute Error (MAE) • Root Mean Squared Error (RMSE) • R-Squared and Adjusted R-Squared	• Recall • Precision • F1-Score • Accuracy • Area Under the Curve (AUC)

Training Data and Test Data

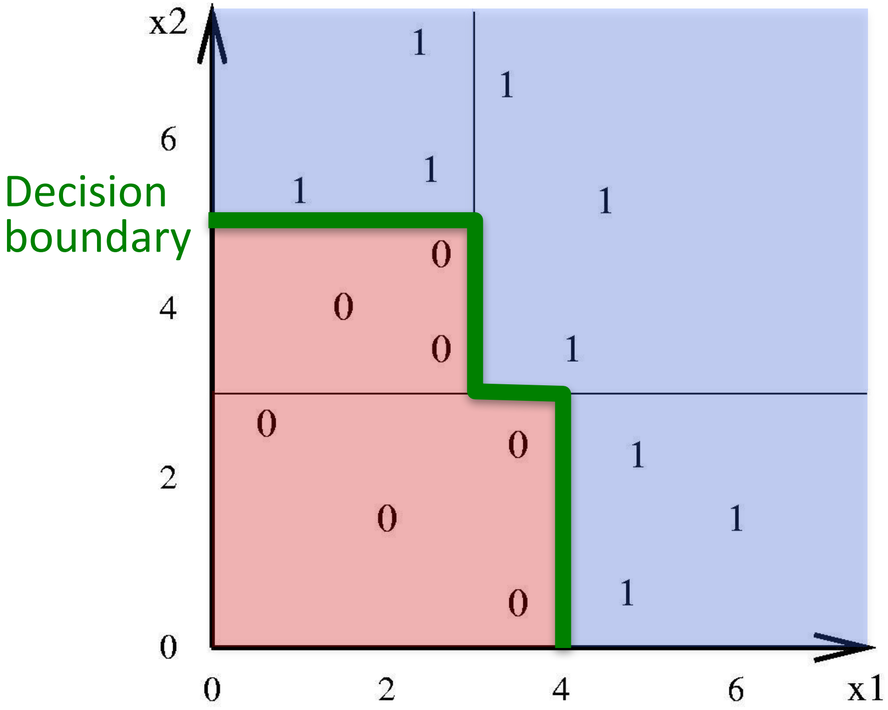
- Training data: data used to build the model
- Test data: new data, not used in the training process

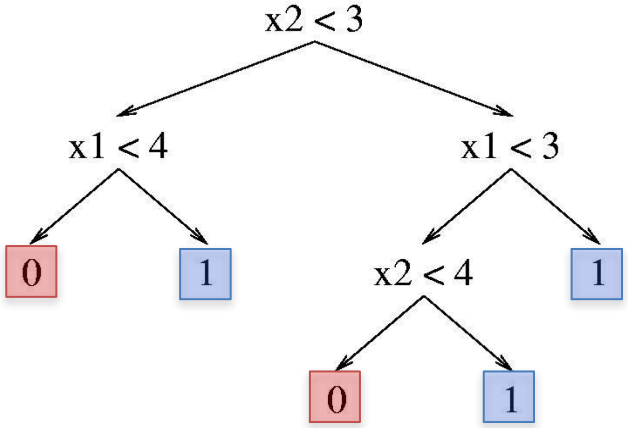
- Training performance is often a poor indicator of generalization performance
  - Generalization is what we really care about in ML
  - Easy to overfit the training data
  - Performance on test data is a good indicator of generalization performance
  - i.e., test accuracy is more important than training accuracy


Training accuracy 98%
Testing accuracy is also 98%

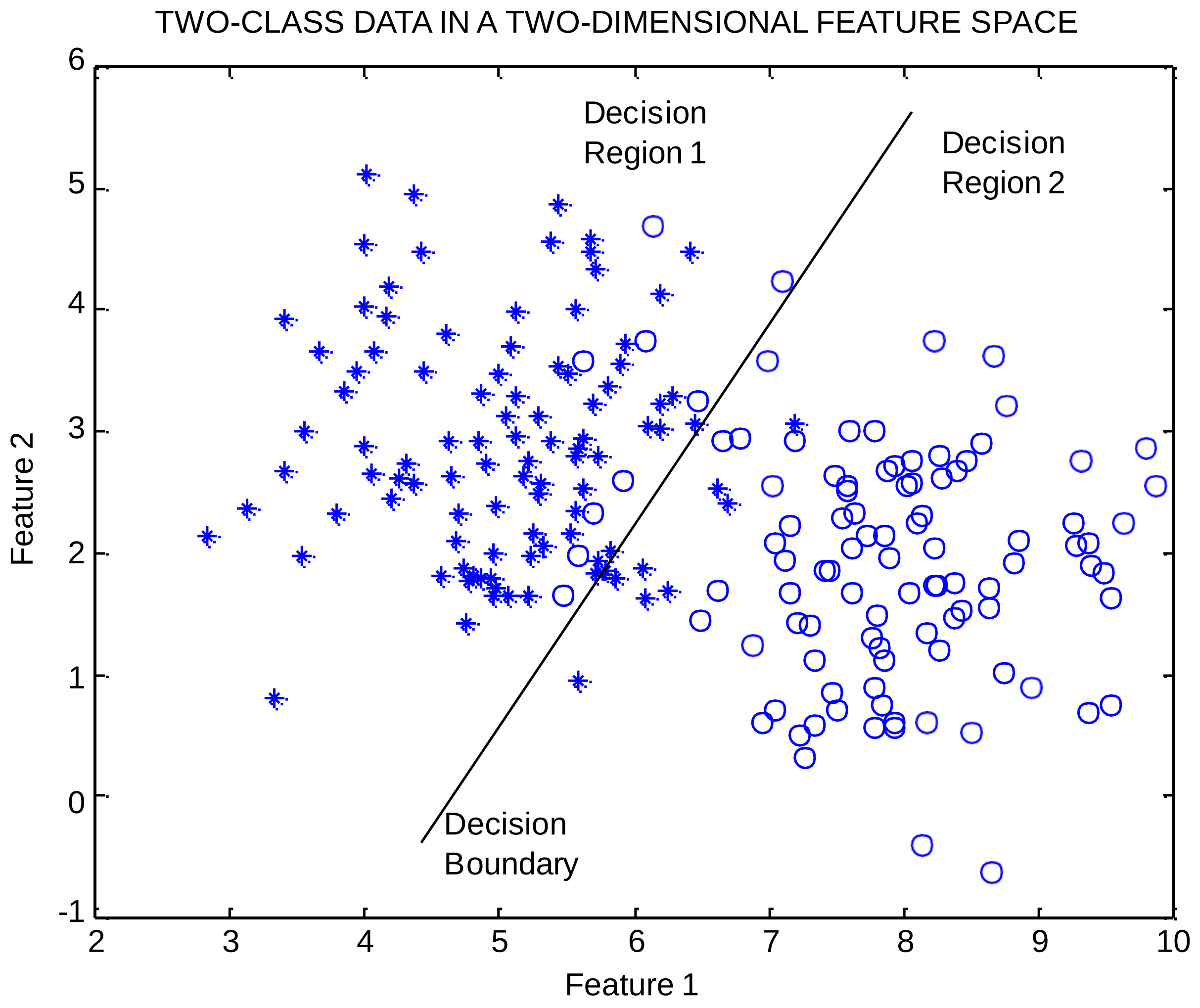
Decision Tree – Decision Boundary

- Decision trees divide the feature space into axis-parallel (hyper-)rectangles
- Each rectangular region is labeled with one label
  - or a probability distribution over labels



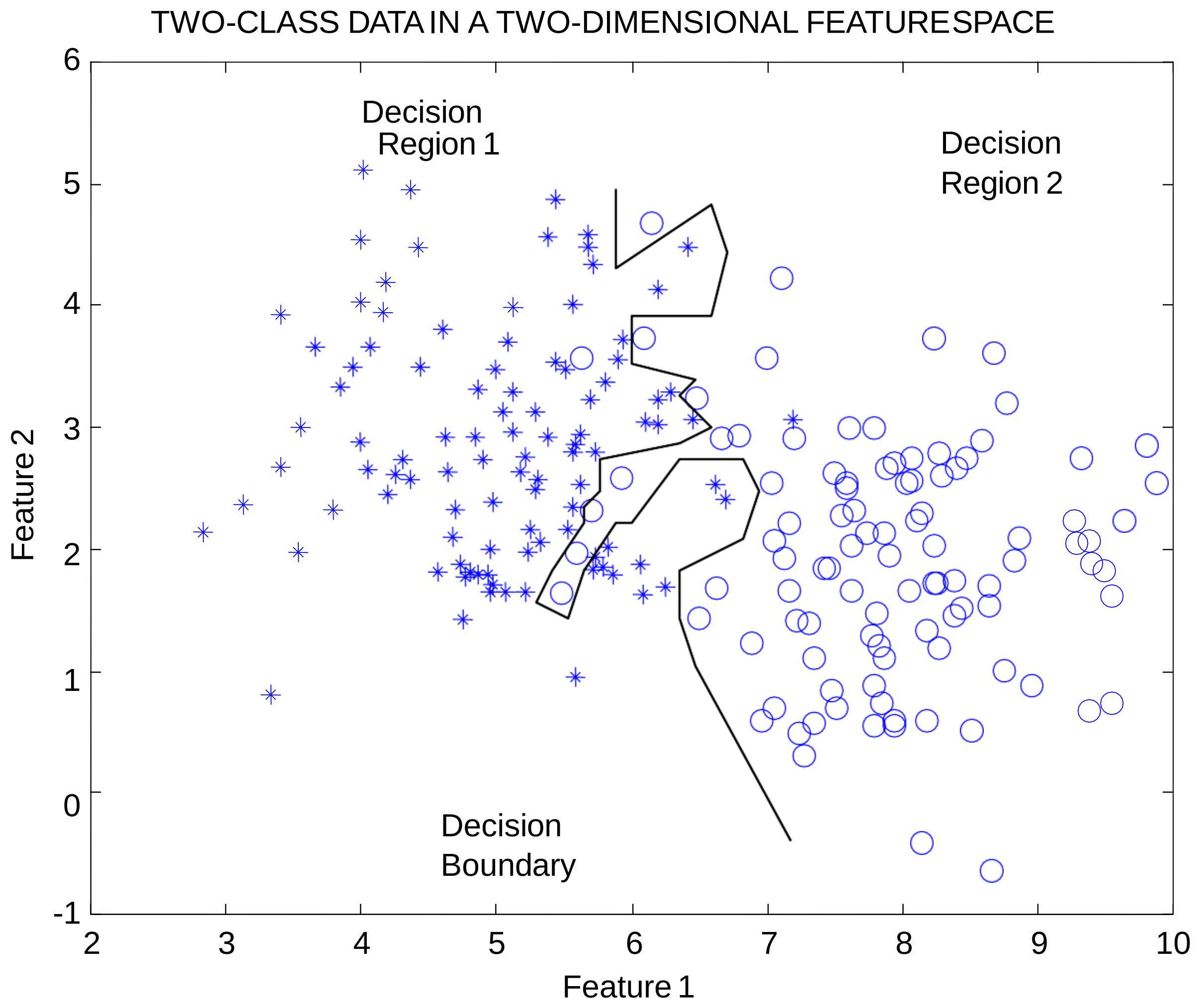


Simple Decision Boundary



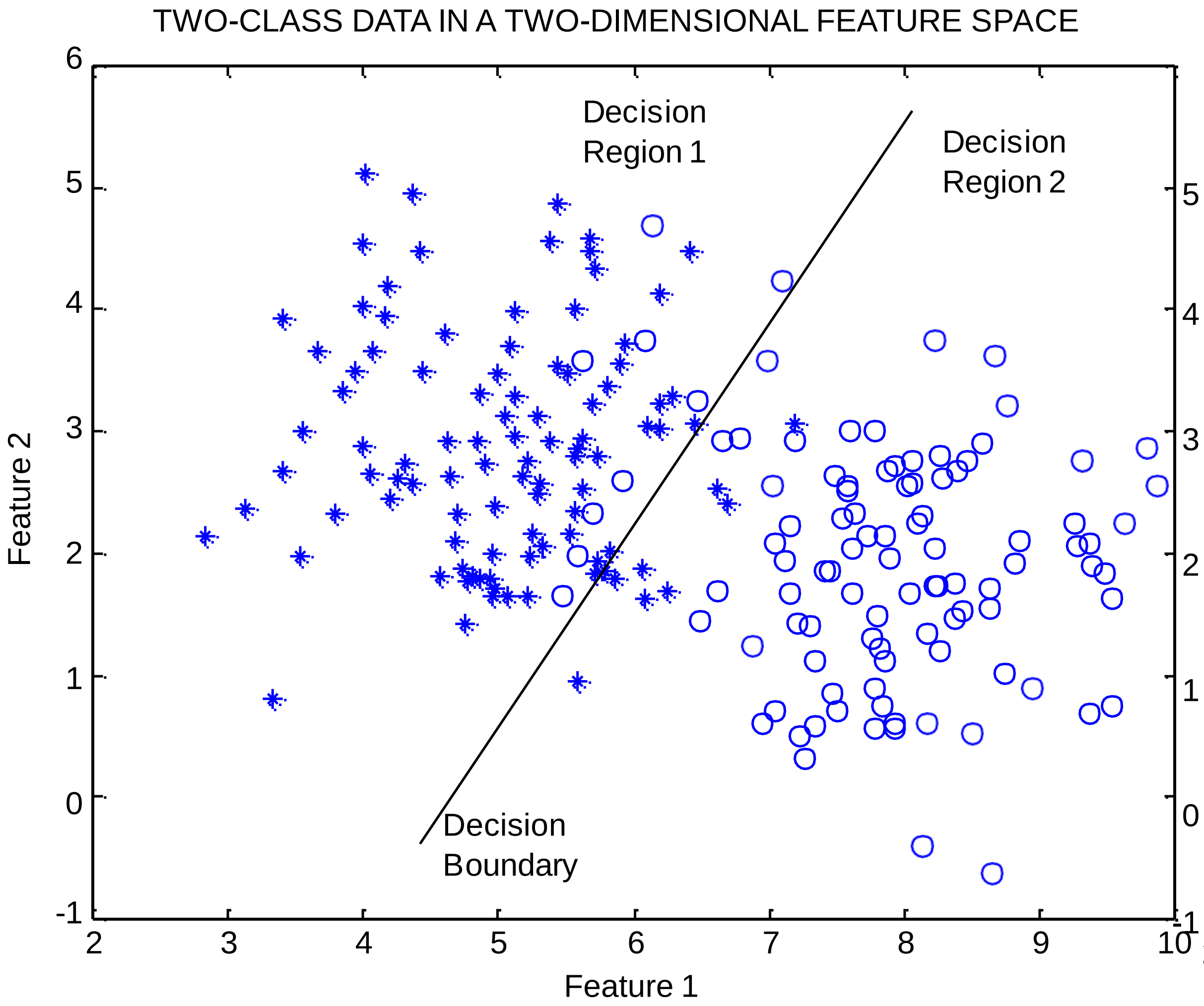
Slide by Padhraic Smyth, UCIrvine

More Complex Decision Boundary



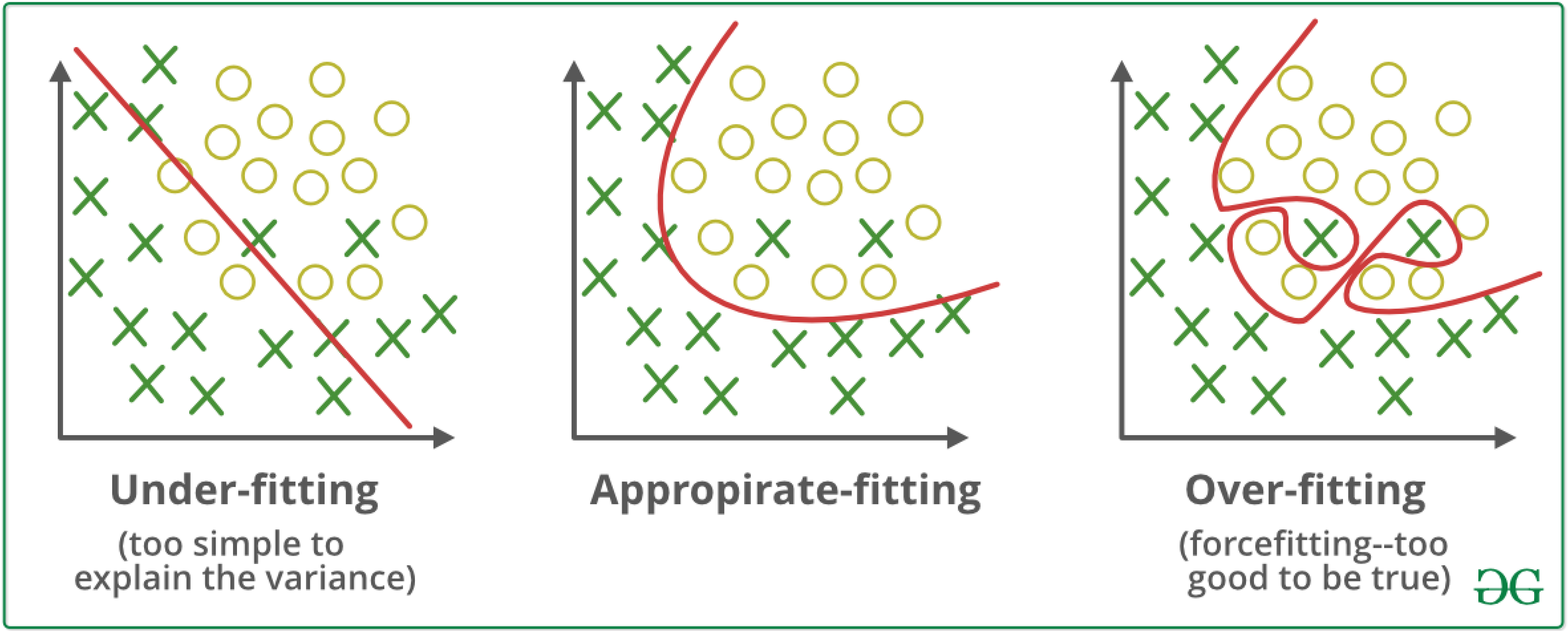
Slide by Padhraic Smyth, UCIrvine


TROLL GASM
RENDUM ONNUKU ONNU SALACHATHU ILLA

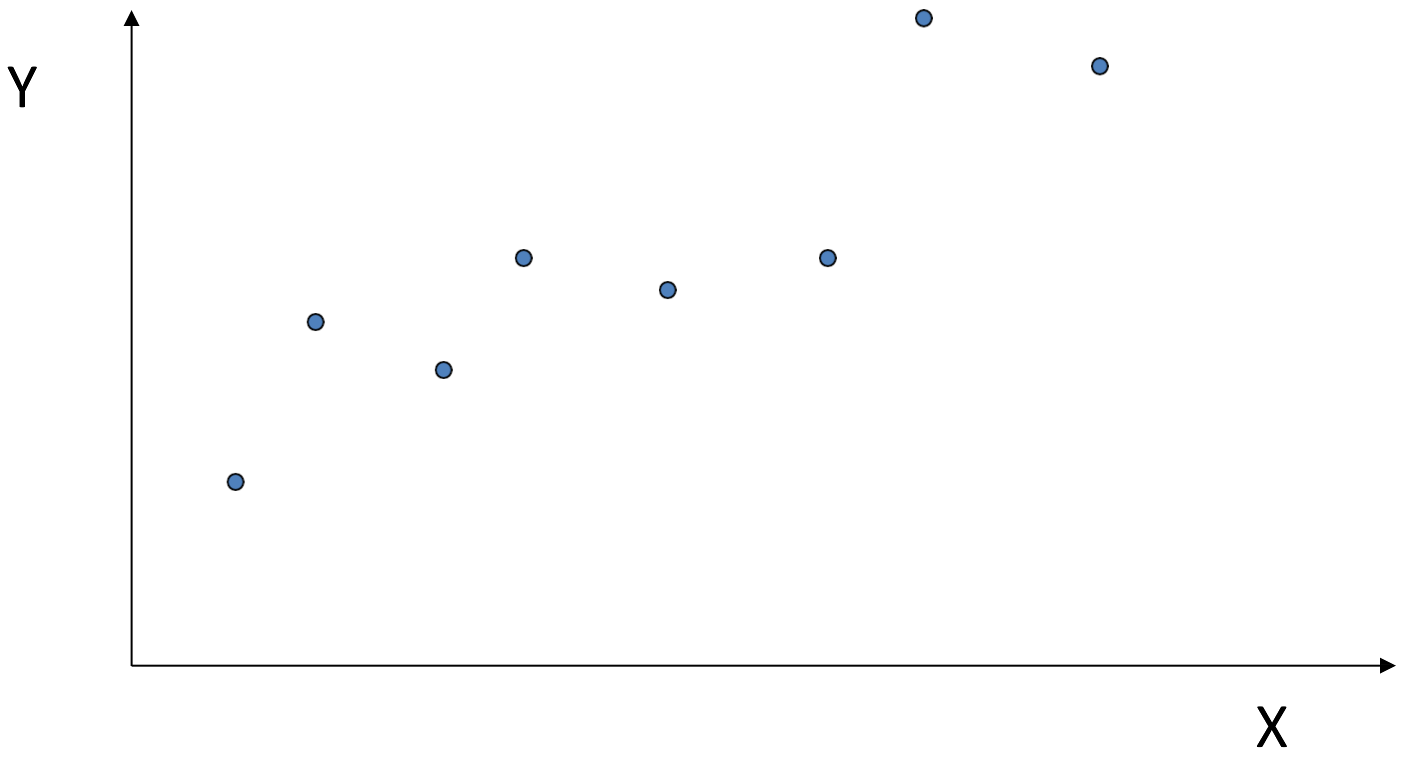

TWO-CLASS DATA IN A TWO-DIMENSIONAL FEATURE SPACE
Decision Region 1
Decision Region 2
Decision Boundary
Feature 1
Feature 2
6
5
4
3
2
1
0
-1
2
3
4
5
6
7
8
9
10


TWO-CLASS DATA IN A TWO-DIMENSIONAL FEATURESPACE
Decision Region 1
Decision Region 2
Decision Boundary
Feature 1
Feature 2
6
5
4
3
2
1
0
-1
2
3
4
5
6
7
8
9
10

Underfitting vs Overfitting

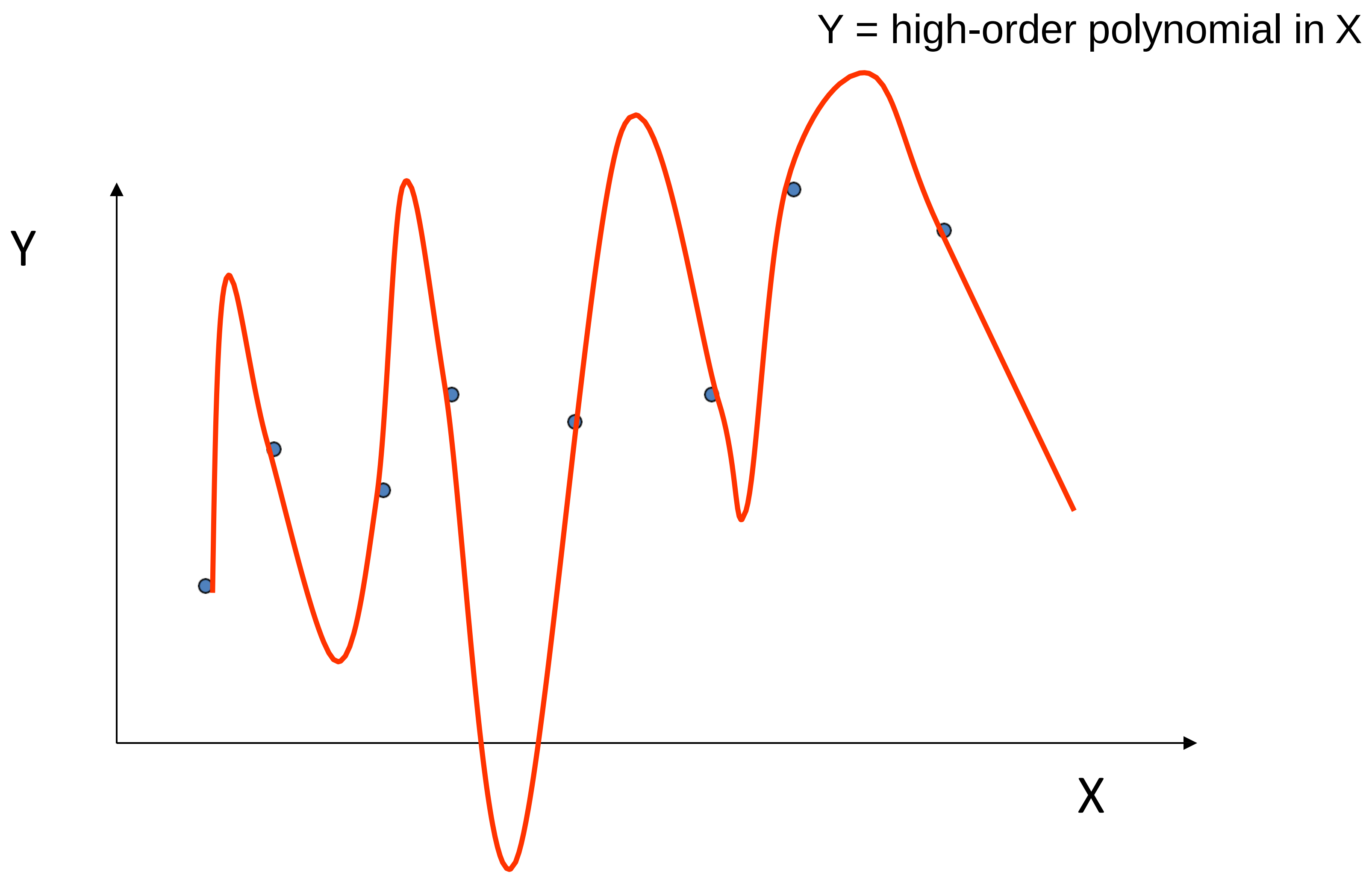


Fitting a Regression Model



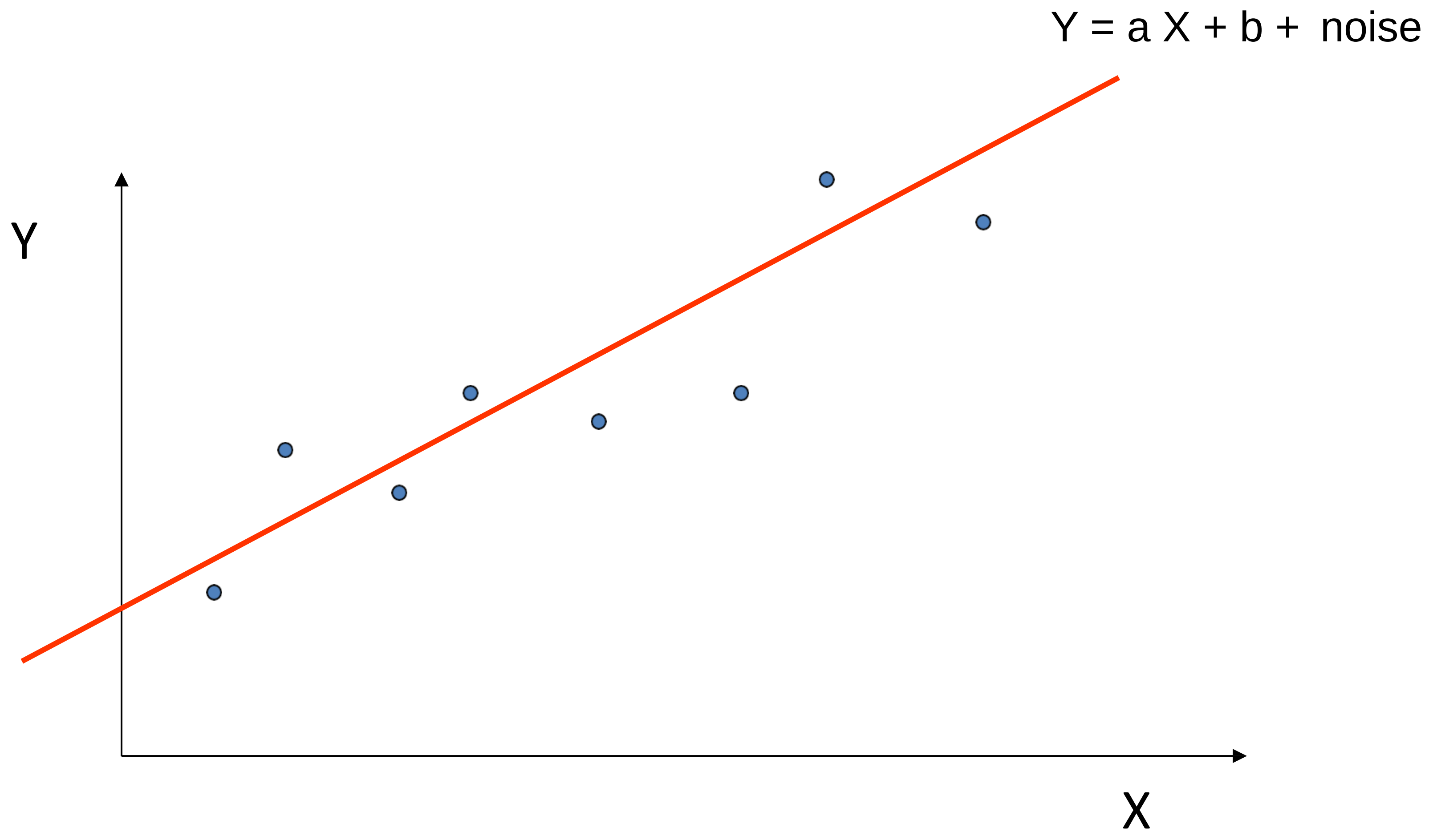
Slide by Padhraic Smyth, UCIrvine

A Complex Model



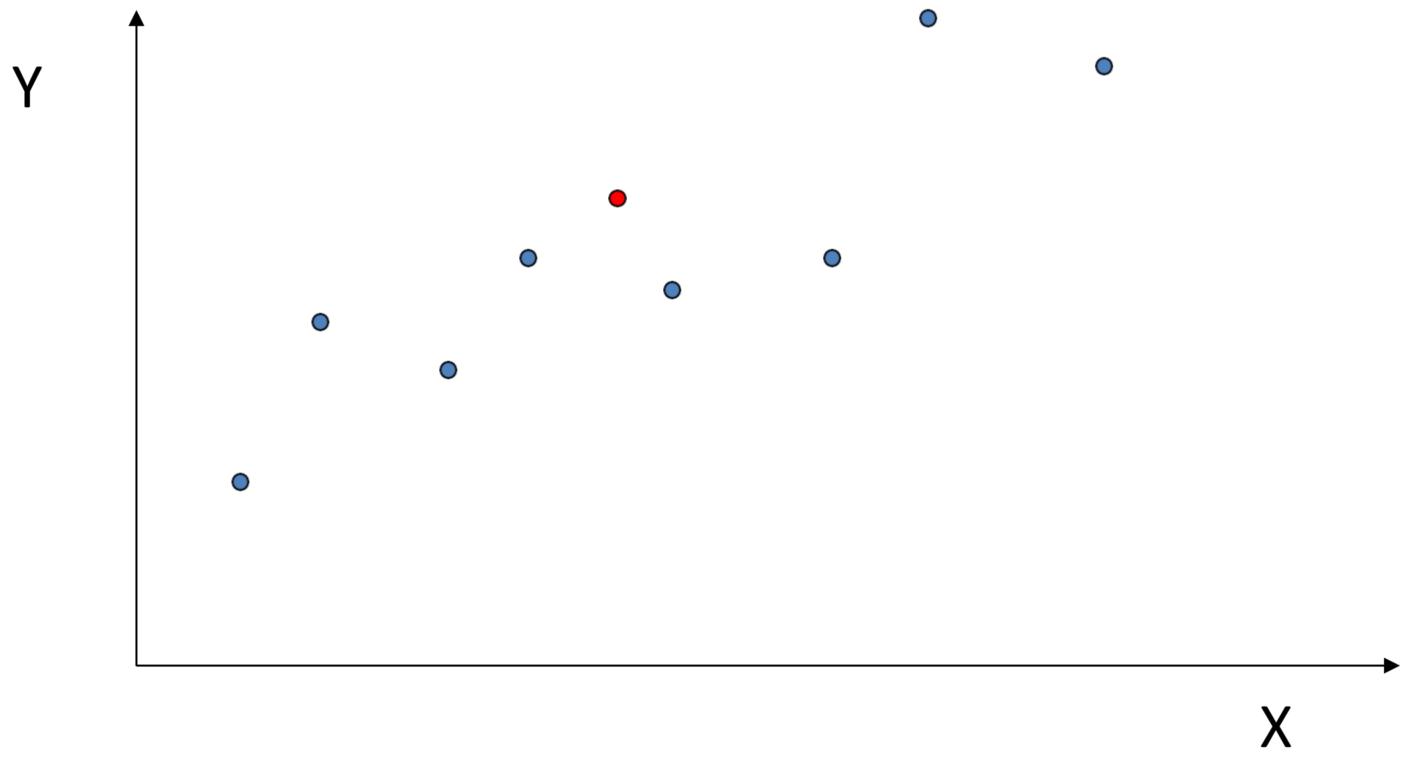
Slide by Padhraic Smyth, UCIrvine

The True (simpler) Model



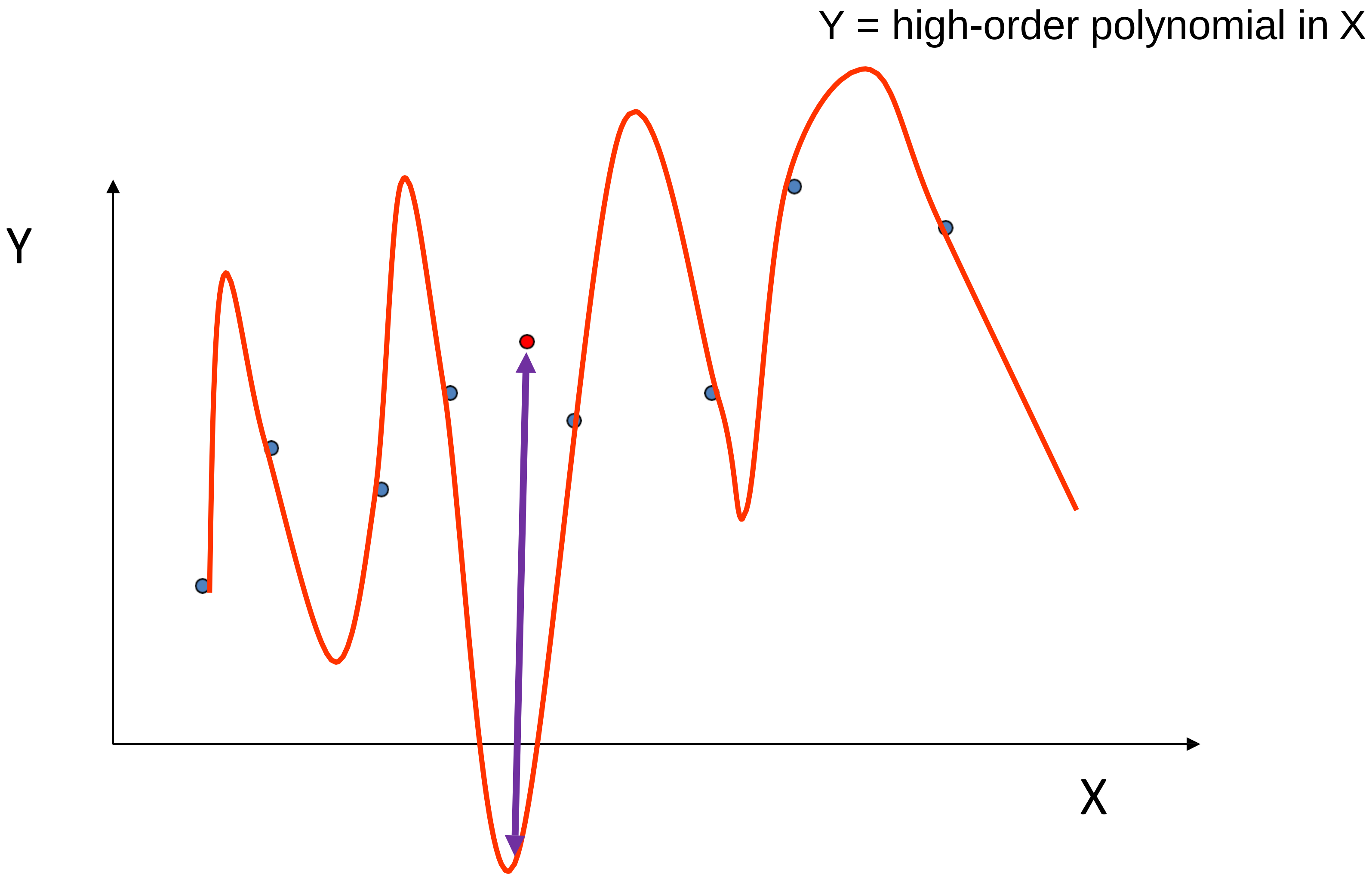
Slide by Padhraic Smyth, UCIrvine

Example: The Overfitting Phenomenon



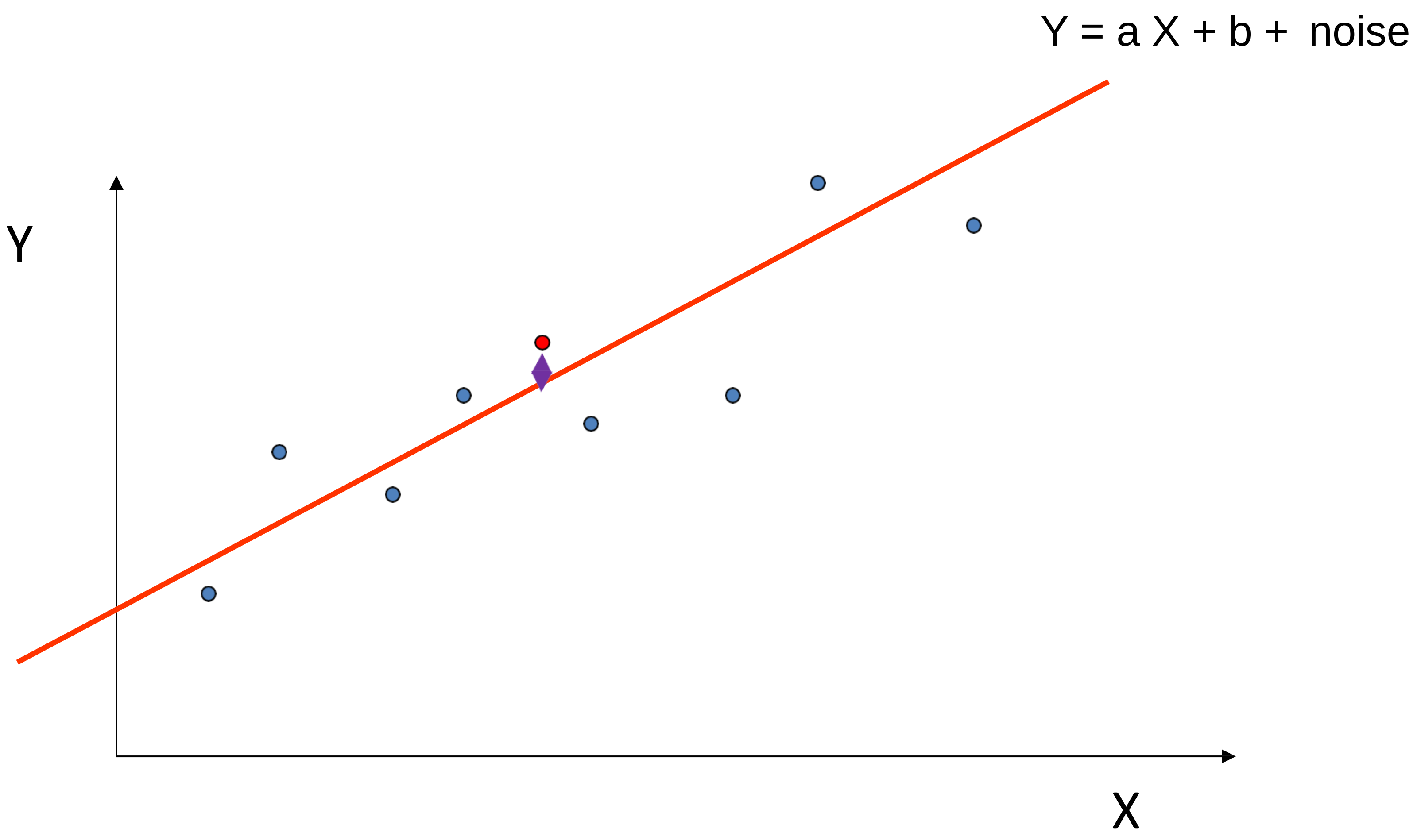
Slide by Padhraic Smyth, UCIrvine

A Complex Model



Slide by Padhraic Smyth, UCIrvine

The True (simpler) Model

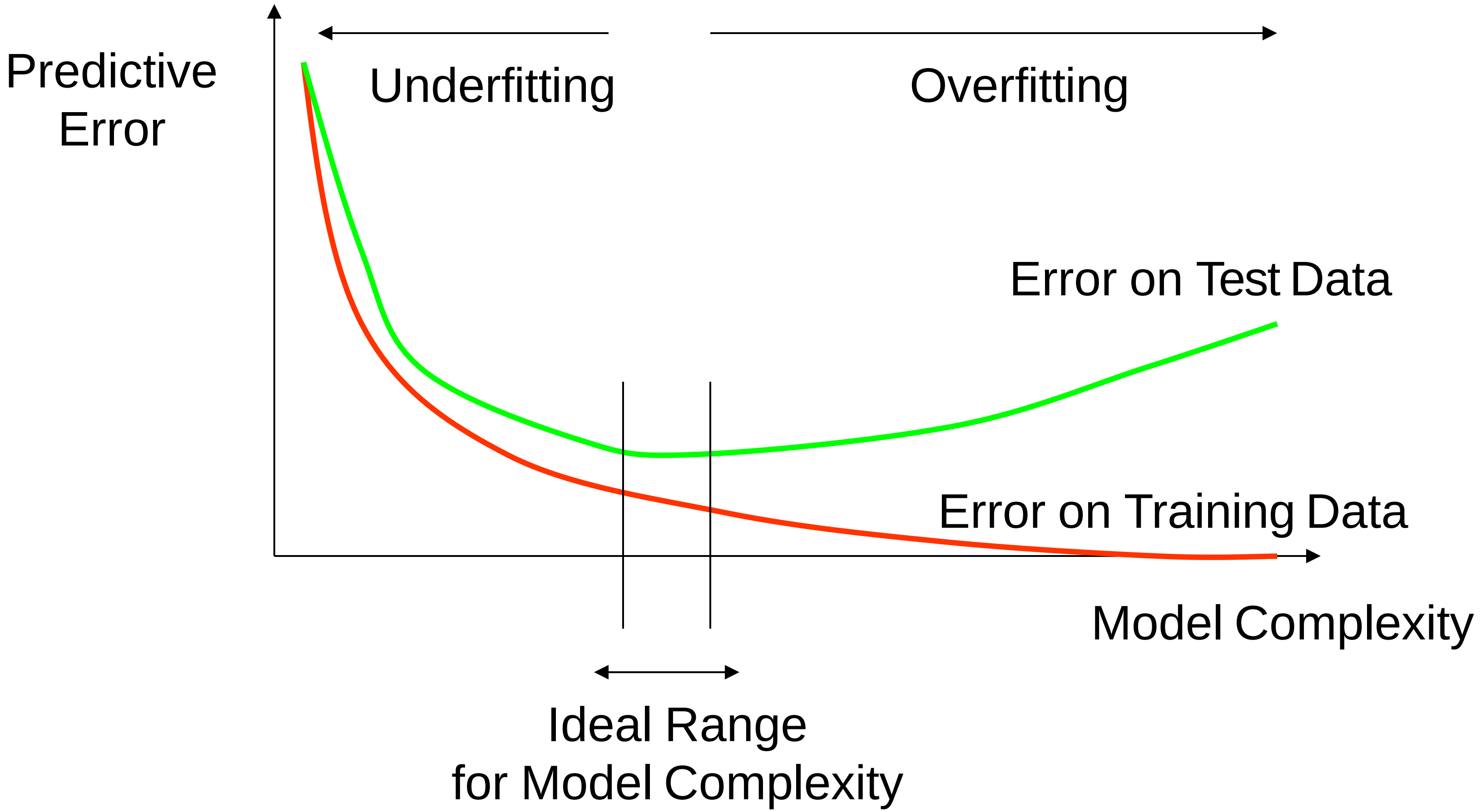


When underfitting and overfitting happens?

Slide by Padhraic Smyth, UCIrvine

How Overfitting Affects Prediction

How Overfitting Affects Prediction



Slide by Padhraic Smyth, UCIrvine

Comparing Classifiers

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

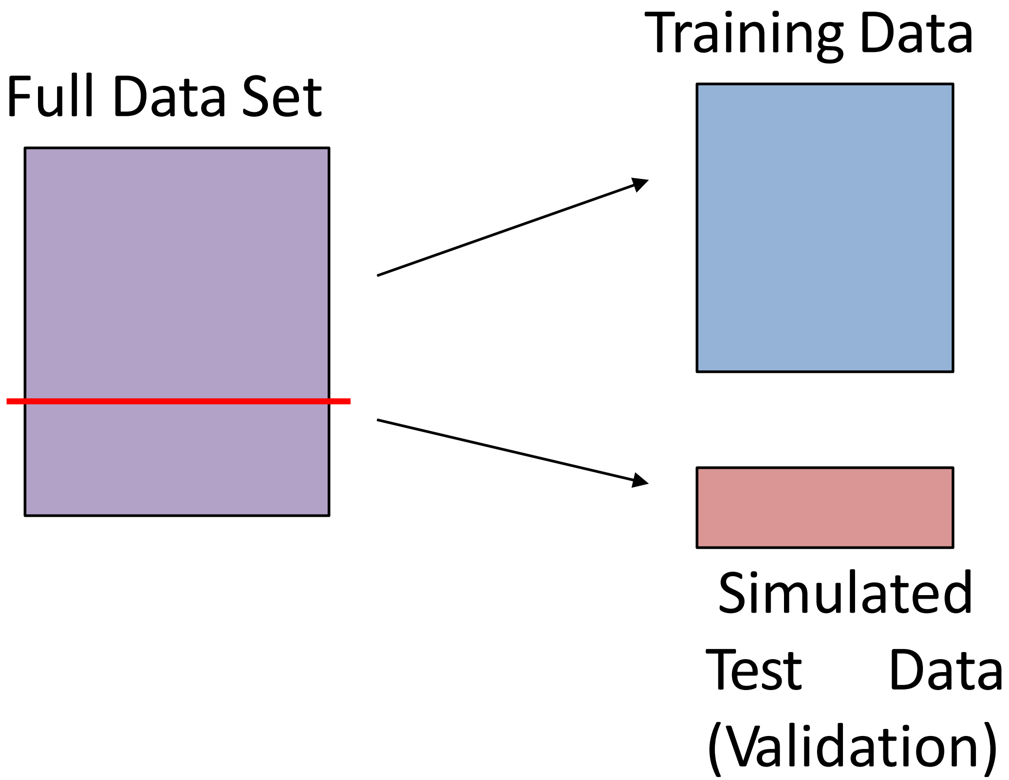
Can we use training accuracy to choose between them?

- 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"...

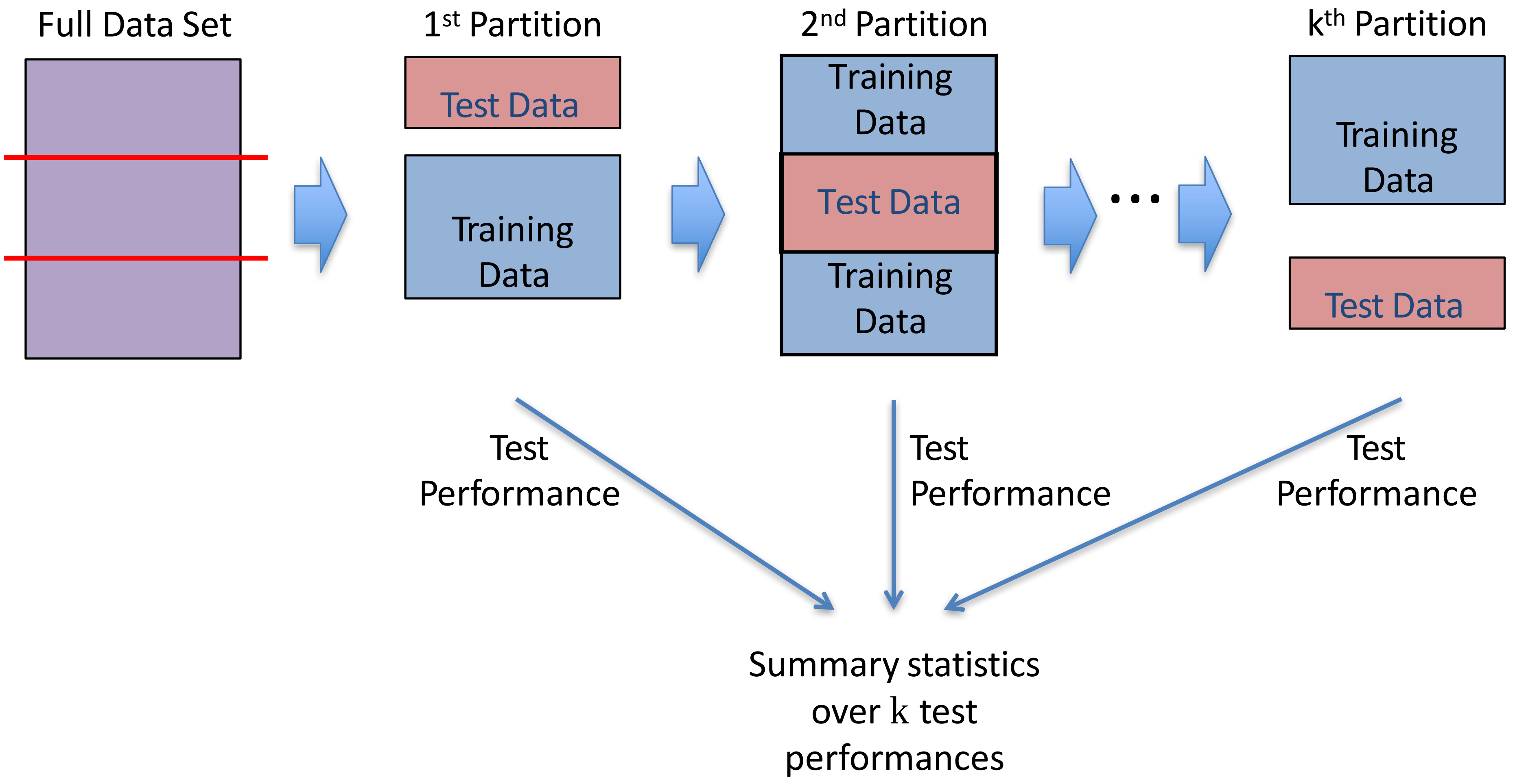
...and then test each model's accuracy on the "test" data

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 <u>k disjoint subsets</u> (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



More on Cross-Validation

- Cross-validation generates an approximate estimate of how well the classifier will do on "unseen" data
  - As $k \rightarrow n$, 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

Learning Curve

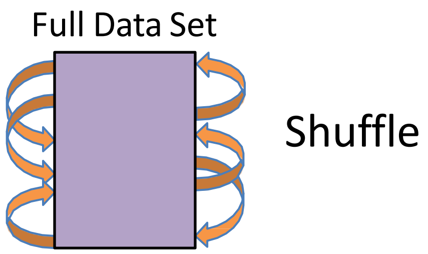
- Line plot of learning (y-axis) over experience (x-axis).

Time for Terms

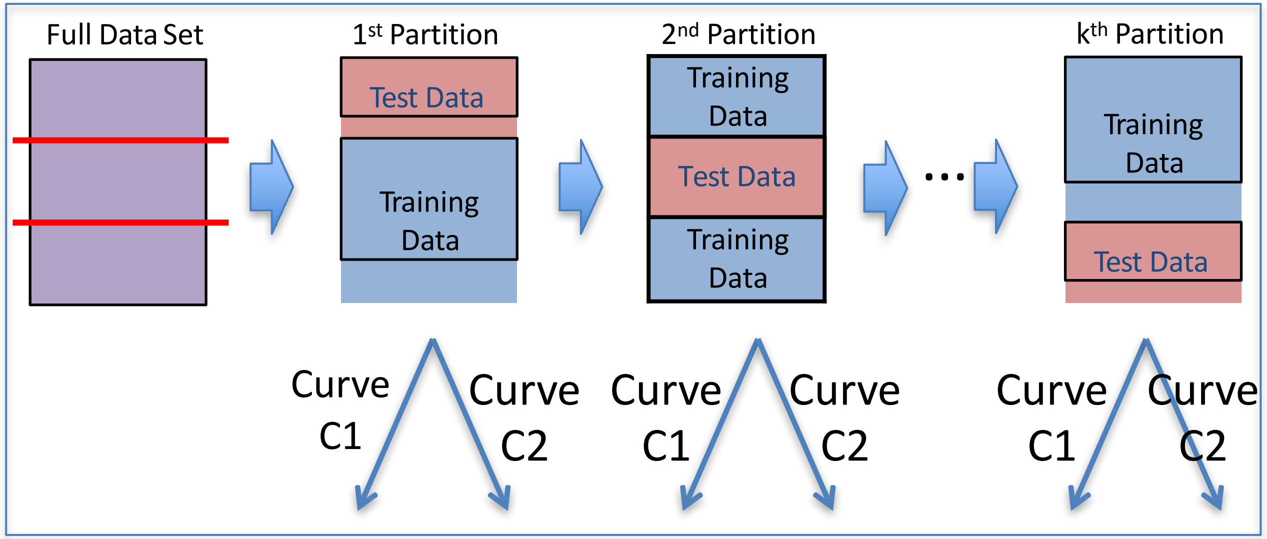
- **Train Learning Curve**: Learning curve calculated from the training dataset that gives an idea of how well the model is learning.
- **Validation Learning Curve**: Learning curve calculated from a hold-out validation dataset that gives an idea of how well the model is generalizing.
- **Optimization Learning Curves**: Learning curves calculated on the metric by which the parameters of the model are being optimized, e.g. loss.
- **Performance Learning Curves**: Learning curves calculated on the metric by which the model will be evaluated and selected, e.g. accuracy.

Building Learning Curves

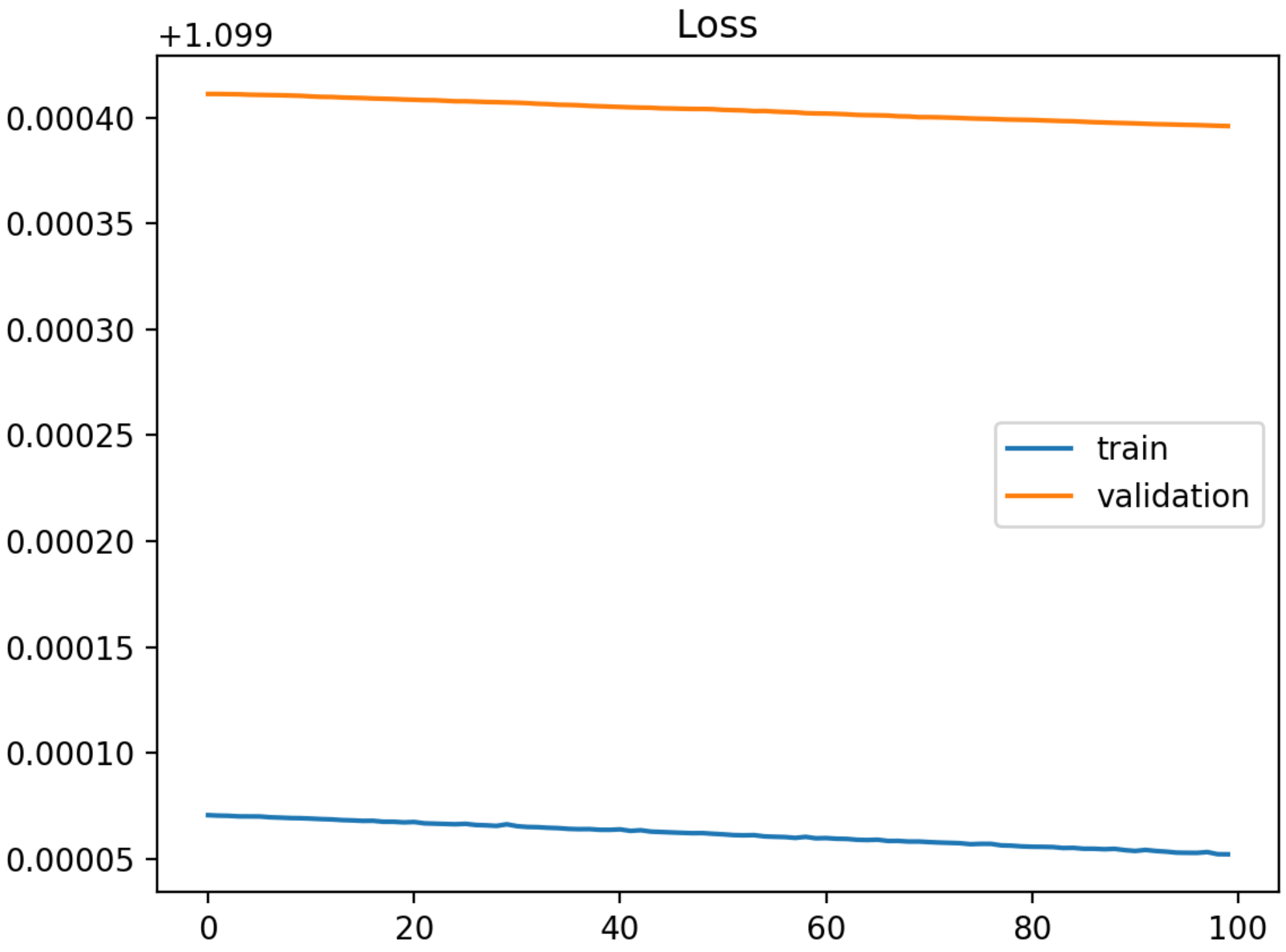
a.) Randomize Data Set



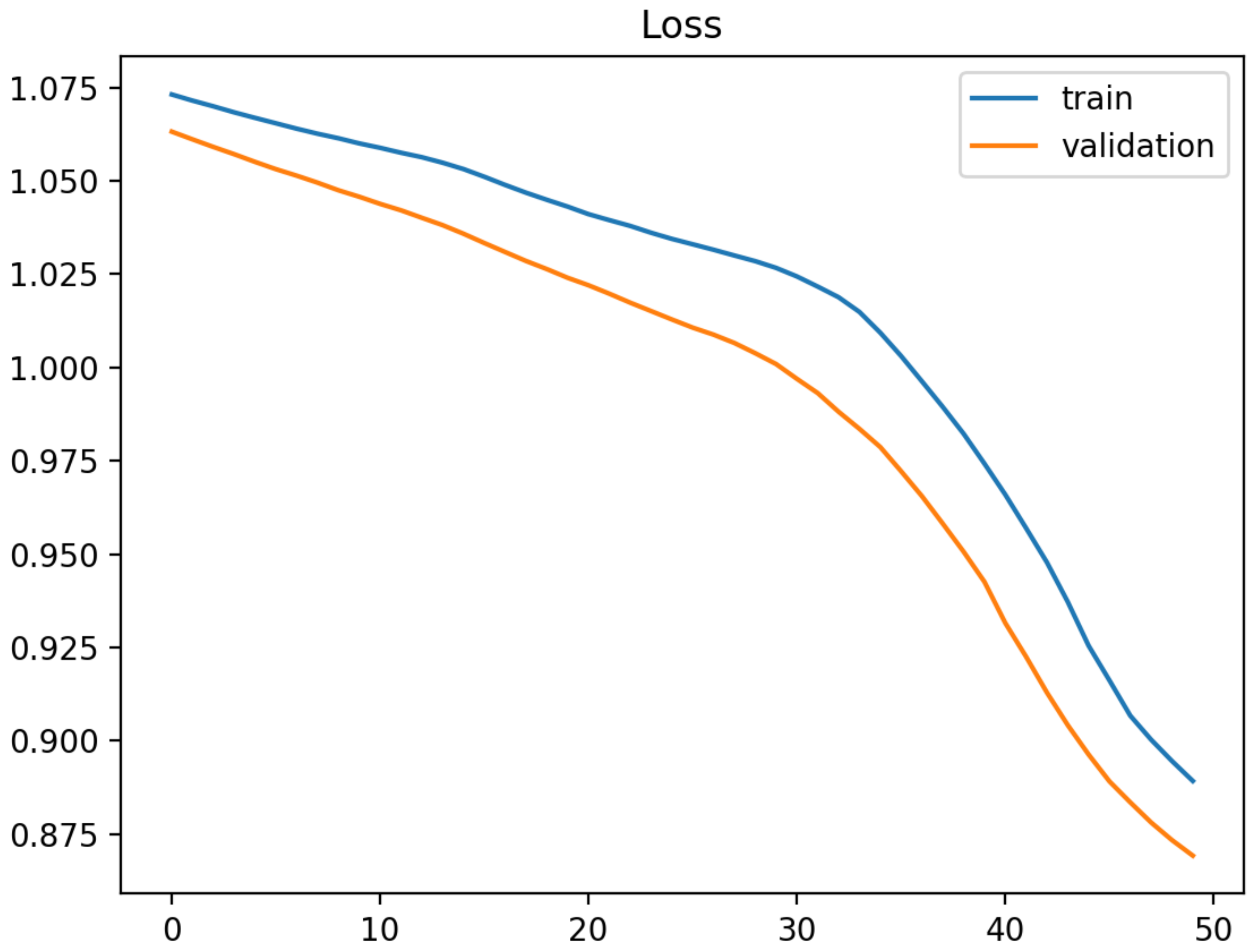
b.) Perform k-fold CV



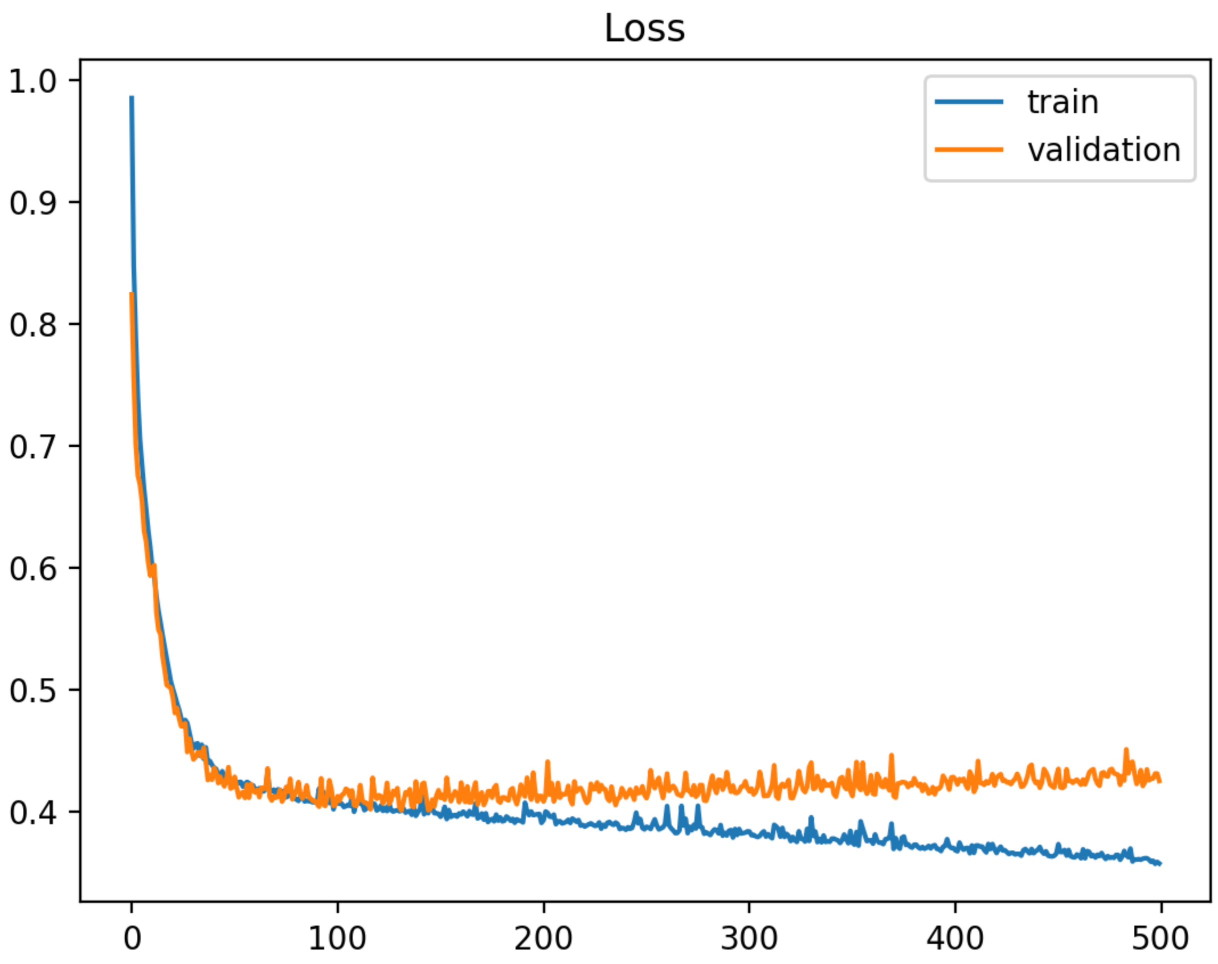
Example of Training Learning Curve Showing An Underfit Model That Does Not Have Sufficient Capacity



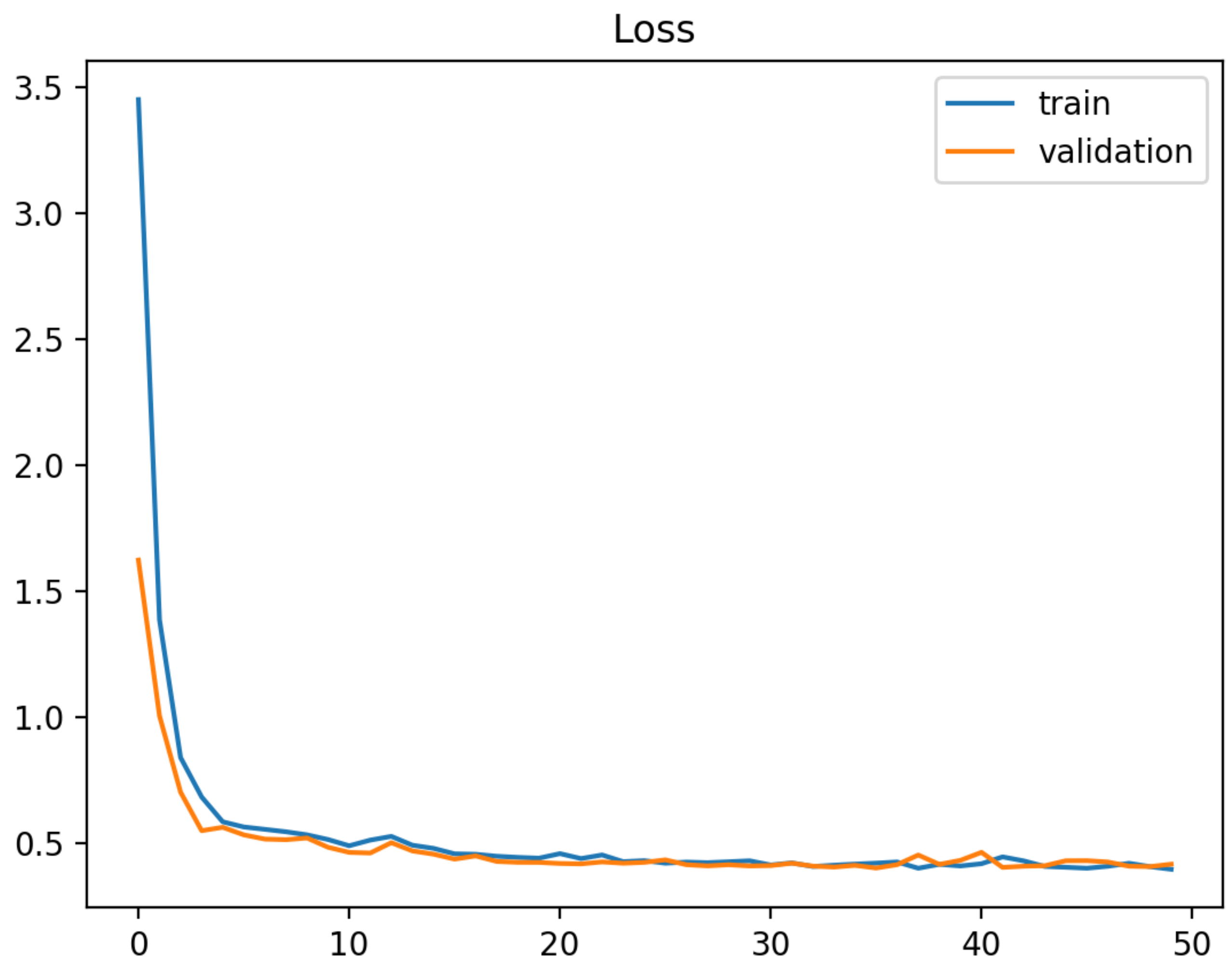
Example of Training Learning Curve Showing an Underfit Model That Requires Further Training



Example of Train and Validation Learning Curves Showing an Overfit Model



Example of Train and Validation Learning Curves Showing a Good Fit



AP International
Training, testing performance rendume balanced ah irukanum
AP International
Oh apdiyane... sarine sarine...