Introduction to Machine Learning

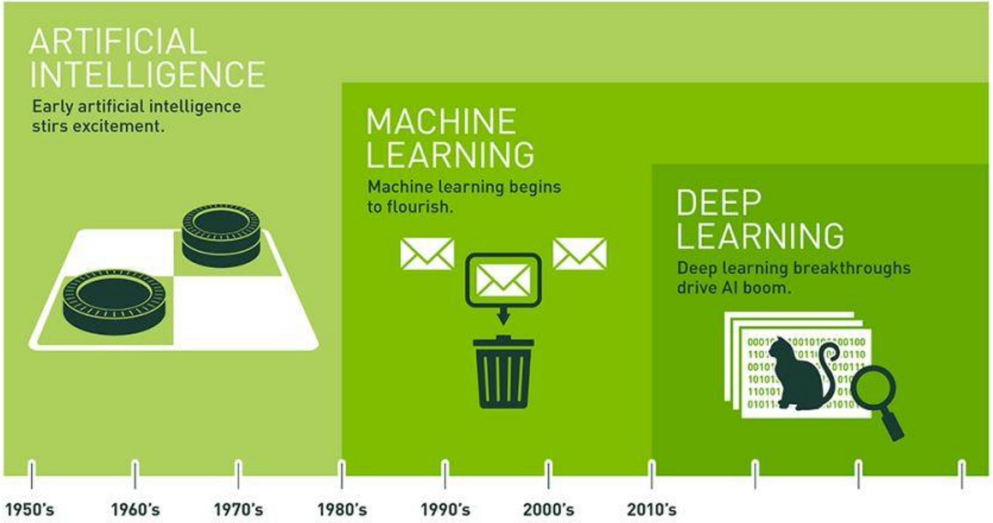
Wu Yu
武钰

DESAY SV automotive

Confidential

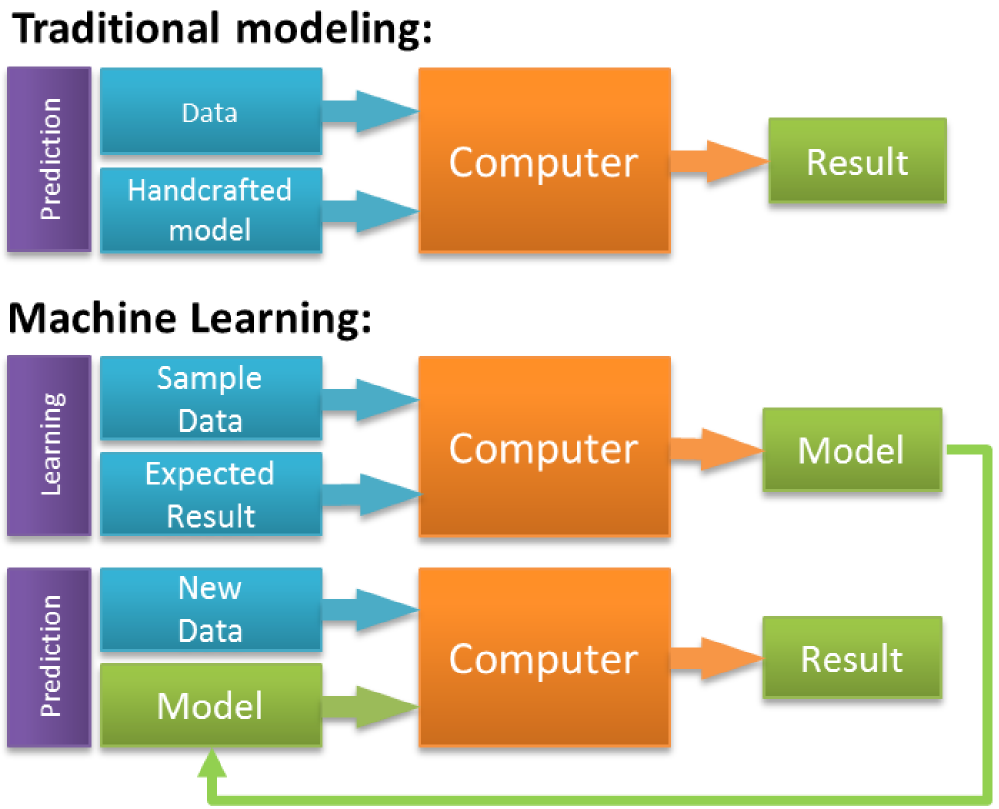
What is Machine Learning

- Branch of Artificial Intelligence
- Learn from data
- Make predictions or decisions



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Structure of Machine Learning



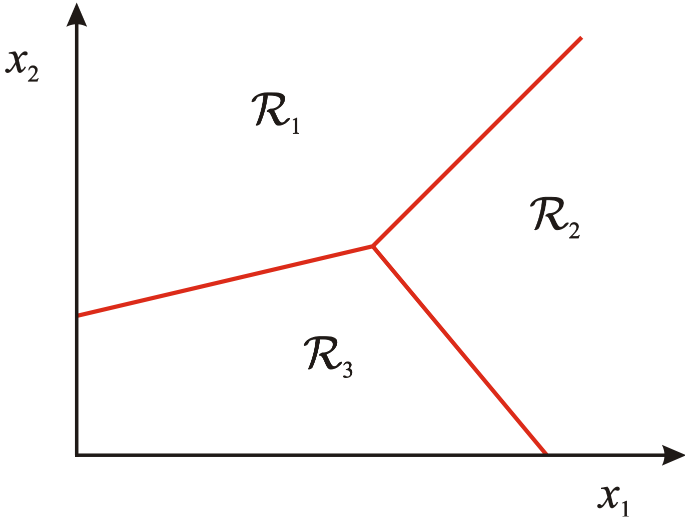
Categories of Machine Learning

Machine Learning Problems

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Classification

- Assign input vector to one of two or more classes
- Any decision rule divides input space into *decision regions* separated by *decision boundaries*



Examples:

Spam email filtering:

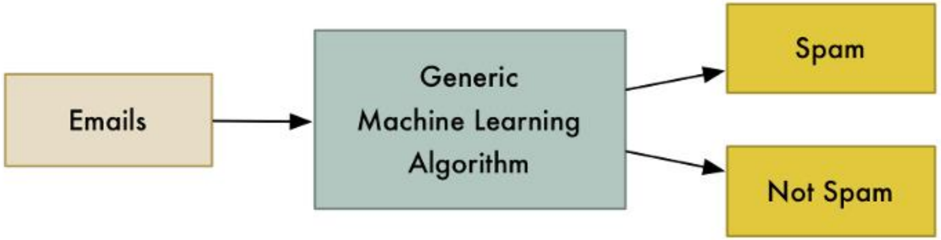


Image recognition:

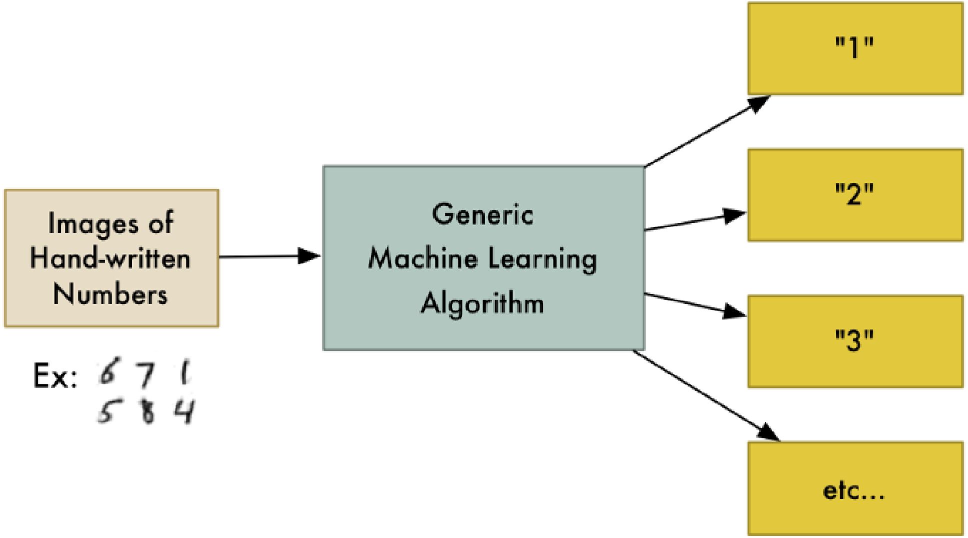


Image recognition

- Apply a prediction function to a feature representation of the image to get the desired output:

F() = “apple”

F() = “tomato”

F() = “cow”

Image recognition

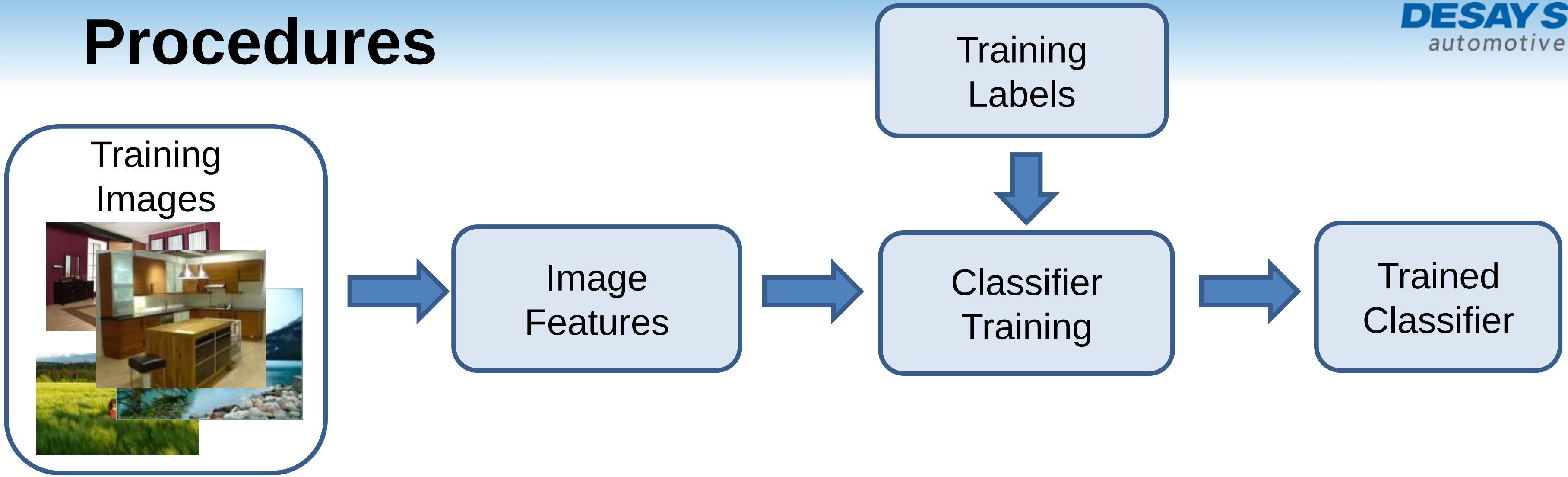
$$y = F(\mathbf{x})$$

Output Prediction function Image feature

Training:
Given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function F by minimizing the prediction error on the training set

Testing:
Apply F to a test *example* $\mathbf{x}$ and output the predicted value $y = F(\mathbf{x})$

Procedures


Training Labels
Training Images
Image Features
Classifier Training
Trained Classifier

Procedures

DESAY SV automotive

Training Labels

Training Images

Image Features

Classifier Training

Trained Classifier

Test Image

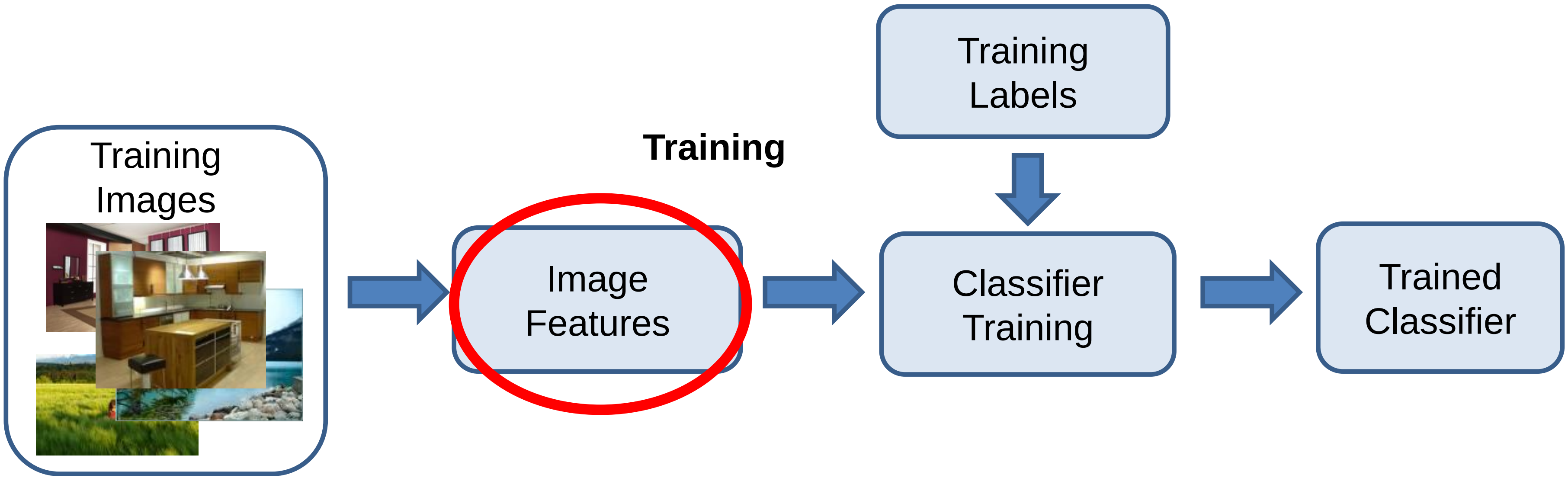
Image Features

Trained Classifier

Prediction

Outdoor

Image Features

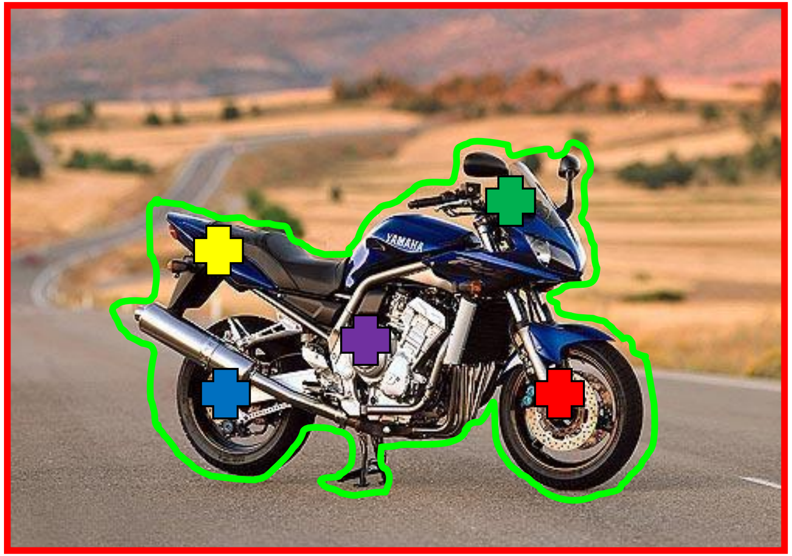


General Principles of Representation

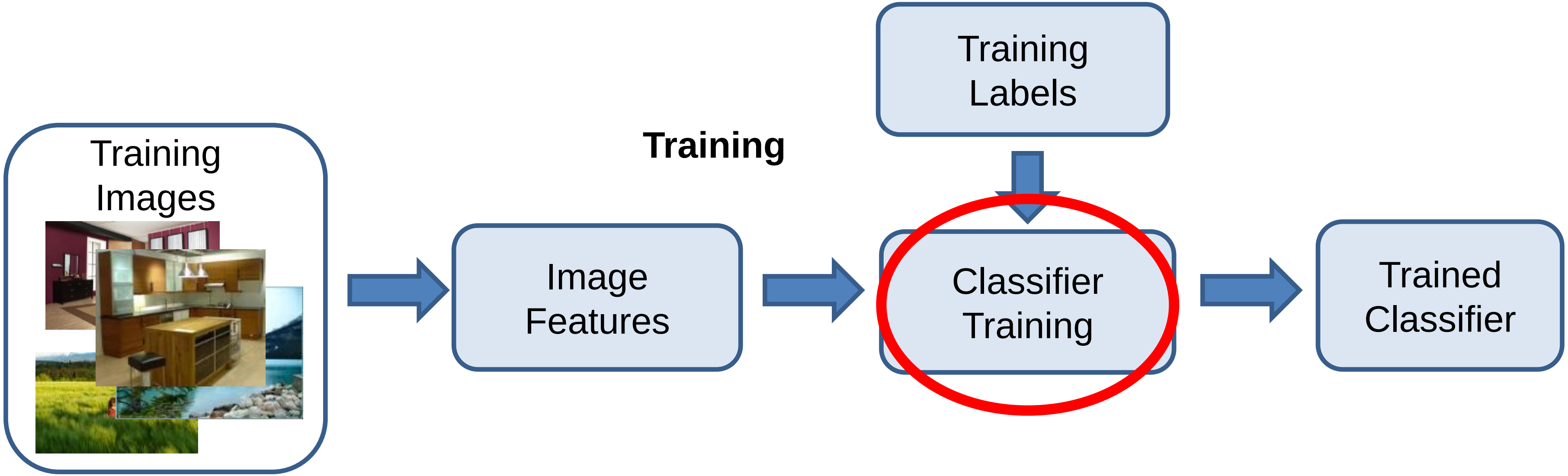
- **Coverage**
Ensure that all relevant info is captured

- **Concision**
Minimize number of features without sacrificing coverage

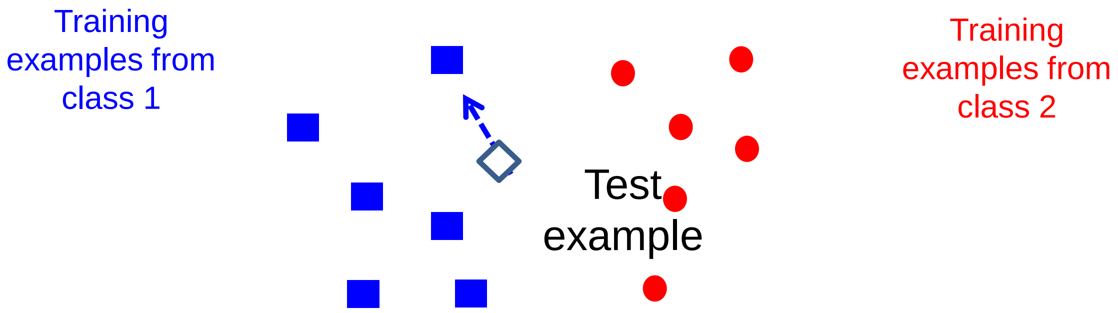
- **Directness**
Ideal features are independently useful for prediction



Classifiers

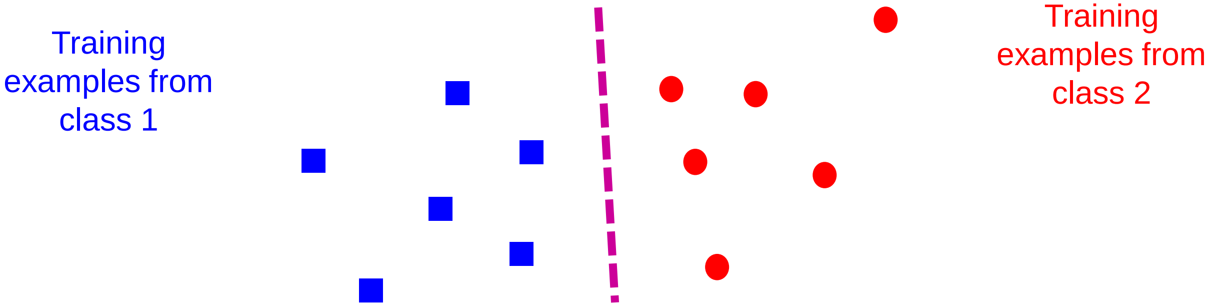


Classifiers: Nearest neighbor



- F(**x**) is label of the training example nearest to **x**
- All we need is a distance function for our inputs
- No training required!

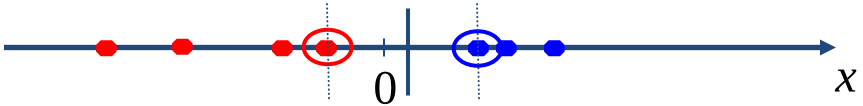
Classifiers: Linear SVM



- There is a *linear function* to separate the classes:
- $F(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$

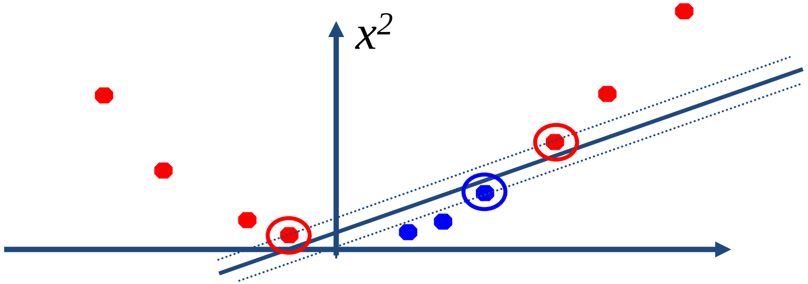
Classifiers: Nonlinear SVM

- Datasets that are linearly separable work out great:



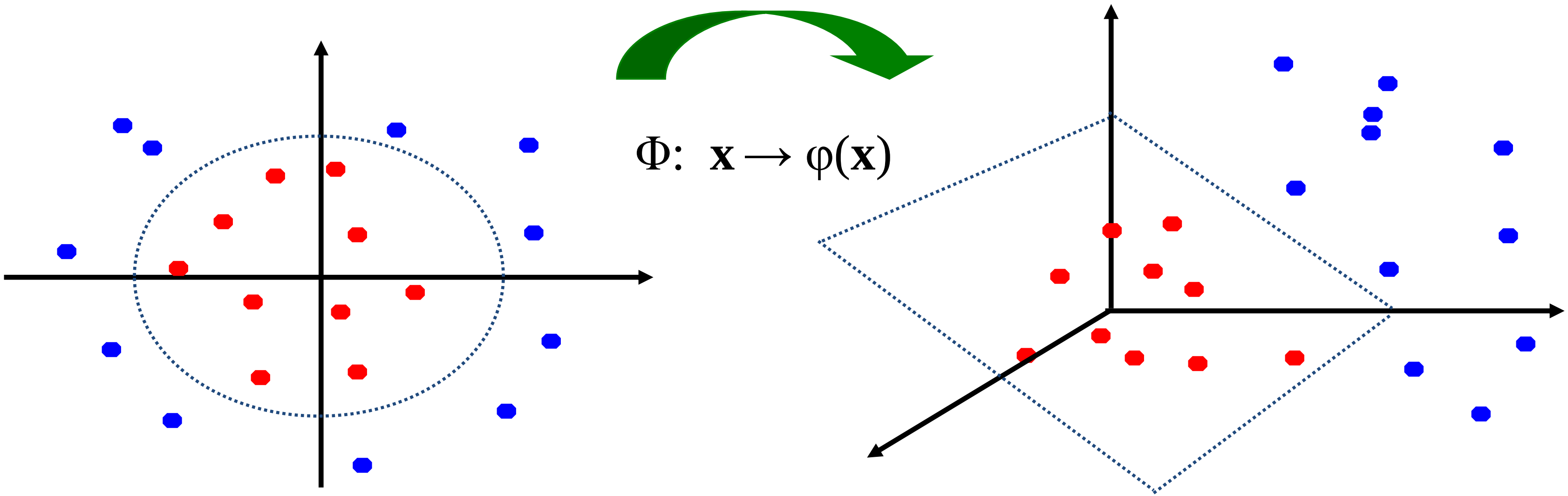
- But what if the dataset is just too hard?

- We can map it to a higher-dimensional space:

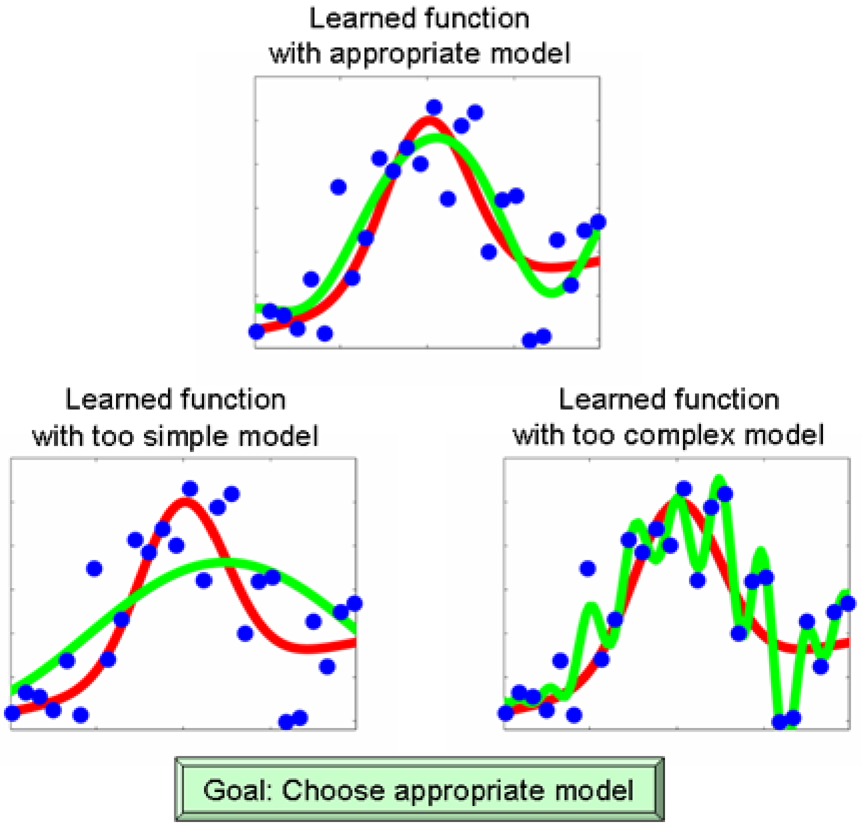


Classifiers: Nonlinear SVM

General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Model Selection



Generalization error

$$\text{Error} = \text{Noise}^2 + \text{Bias}^2 + \text{Variance}$$

Noise²: Unavoidable error

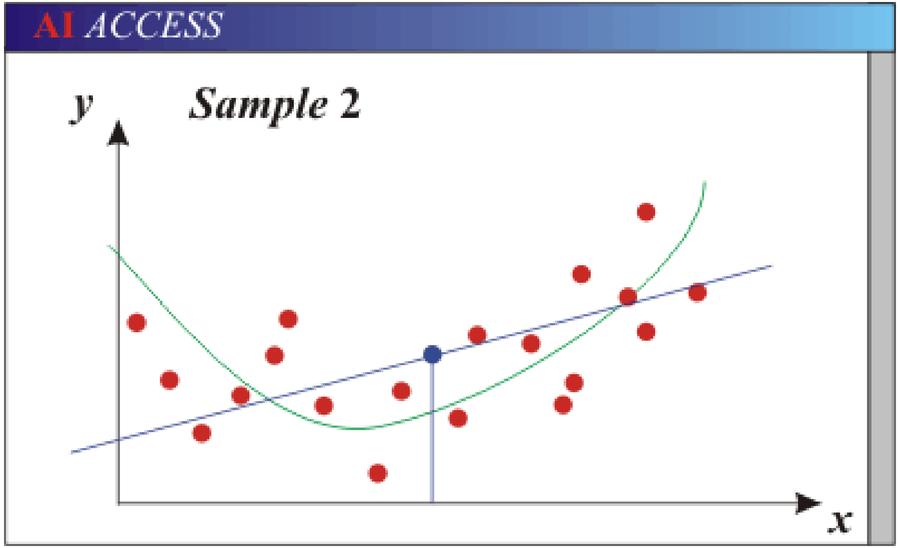
Bias²: Error due to inaccurate assumptions/simplifications made by the model

Variance: Error due to variance of training samples

Under Fitting

Model is too “simple” to represent all the relevant class characteristics:

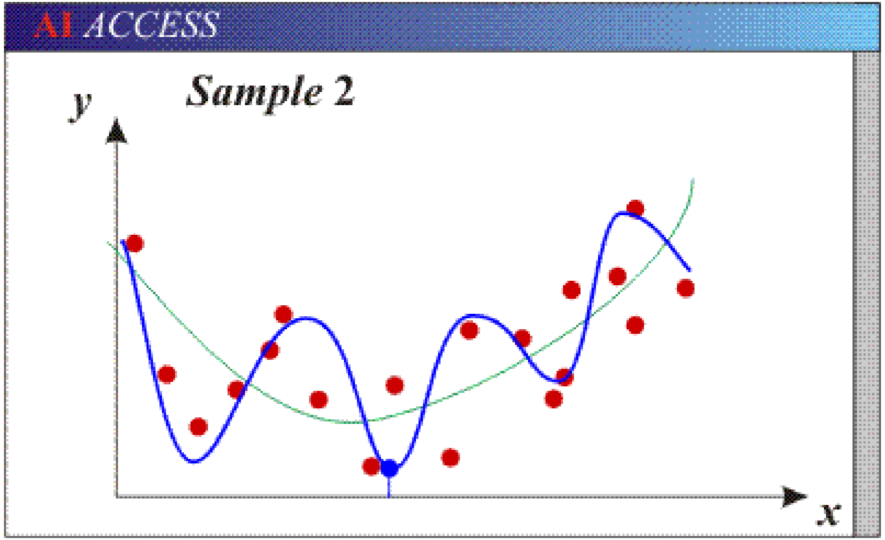
- High bias and low variance
- High training error and high test error
- Not enough flexibility



Over Fitting

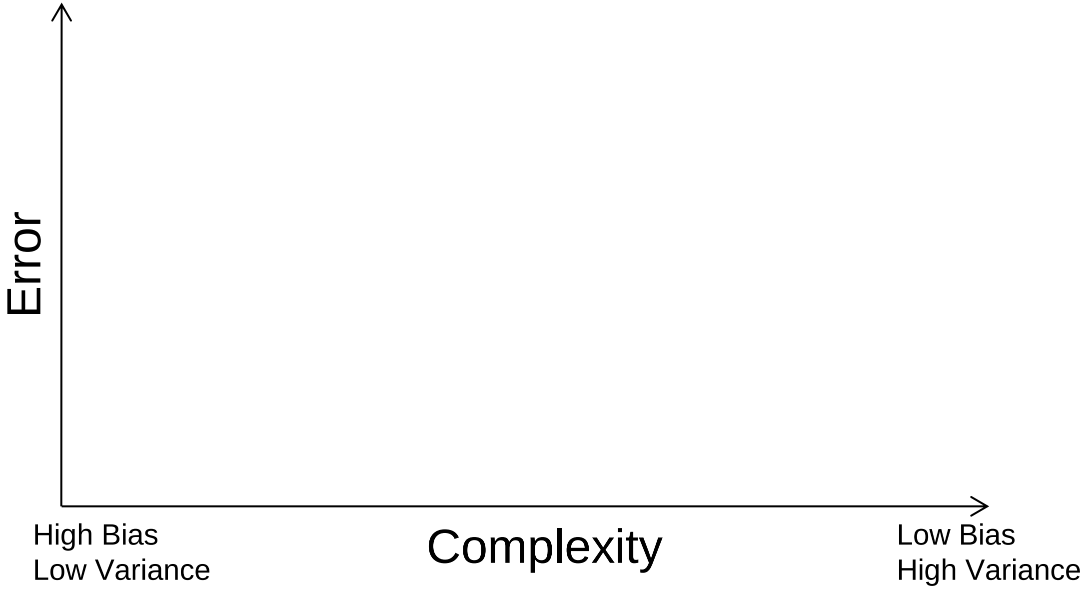
Model is too “complex” and fits irrelevant characteristics (noise) in the data:

- Low bias and high variance
- Low training error and high test error
- Too much sensitivity to the sample



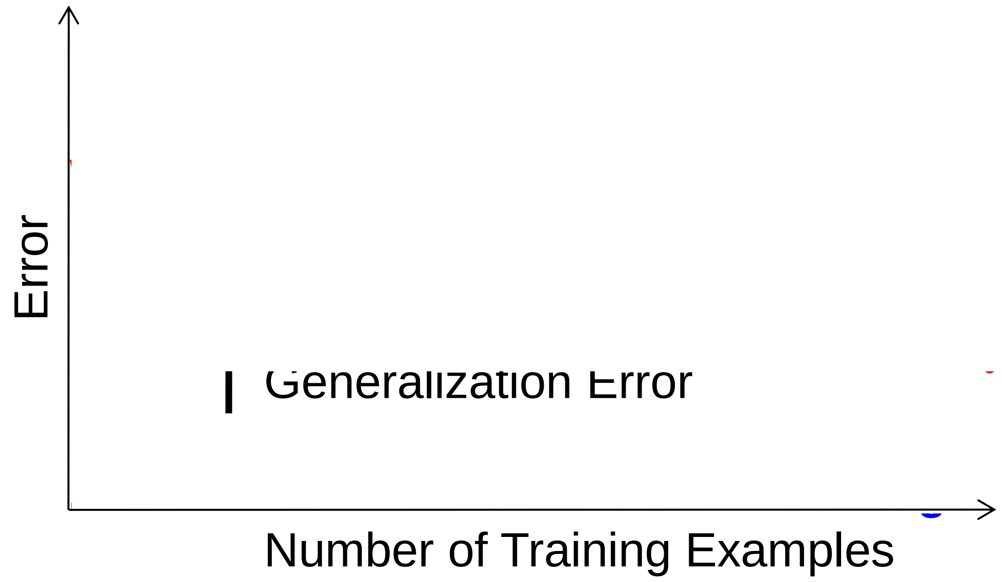
Bias-Variance Trade-off

Underfitting **Overfitting**



Bias-Variance Trade-off

Fixed prediction model



The perfect classification algorithm

Objective function

- Encode the right loss for the problem

Parameterization

- Makes assumptions that fit the problem

Training algorithm

- Find parameters that maximize objective on training set

Inference algorithm

- Solve for objective function in evaluation

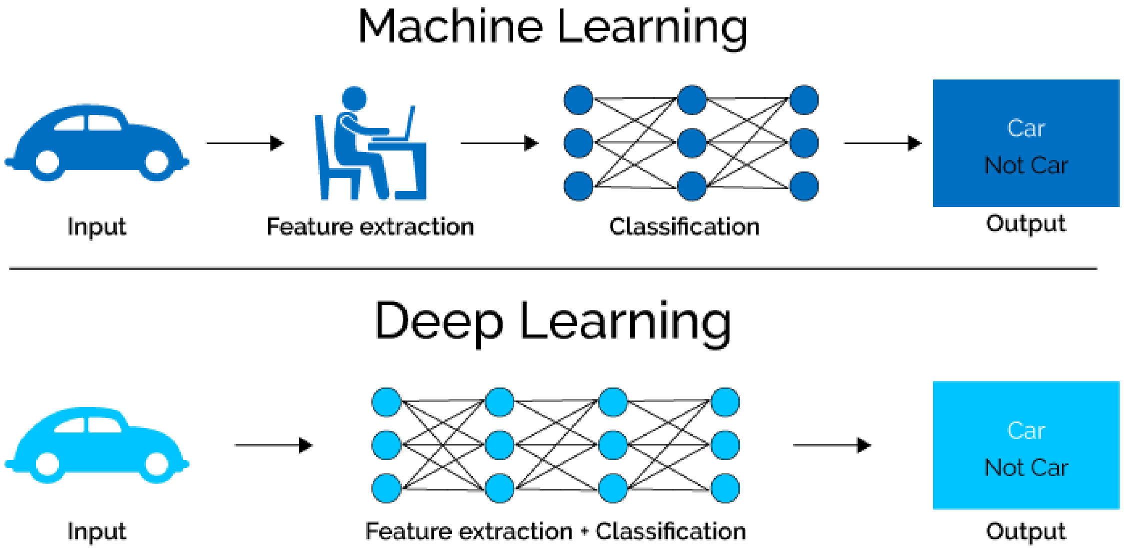
More about classifiers

- Machine learning algorithms are tools, not dogmas.
- Try simple classifiers first.
- Better to have smart features and simple classifiers than simple features and smart classifiers.
- Use increasingly powerful classifiers with more training data (bias-variance tradeoff).

Different types of classifiers

- SVM, Adaboost, …
- Neural Networks

Deep learning



Thank You !