Machine Learning for Economists: Part 1 – Introduction

Michal Andrle
International Monetary Fund

Washington, D.C.,
October, 2018

Disclaimer #1:

The views expressed herein are those of the authors and should not be attributed to the International Monetary Fund, its Executive Board, or its management.

What is ‘Machine Learning’? Is it ‘**Magic**’?

“Machine learning is the science of getting computers to act without being explicitly programmed. ”

Standford ML Coursera Course

“Machine Learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world.”

NVIDIA

“Each machine learning problem can be precisely defined as the problem of improving some measure of performance **P** when executing some task **T**, through some type of training experience **E**. ”

T. Mitchell

What is ‘Machine Learning’ (ML)?

Machine Learning is not magic. . .

Machine learning is a field of study using mathematics, statistics, information theory and other related field creating models to detect patterns in data and use those to predict unseen outcomes or help with decisions under uncertainty.

ML has deep roots in and overlaps with statistics and information theory.

Machine learning is a set of procedures to create and estimate **models** interpreting and/or predicting data:

$$\mathbf{X} \rightarrow \mathbf{ALGORITHM} \rightarrow \mathbf{Y}$$

Examples of ML Applications: The X's and Y's

Economics & Finance

- Crisis prediction, default detection
- Now-casting and forecasting (equity, forex, . . .)
- Risk analysis and measurement, fraud transaction detection
- . . .

General Applications:

- Document classification
- Email spam filtering, fraud detection
- Image classification, fingerprints analysis
- Speech recognition, text and speech translation
- Disease prediction, X-Ray and MRI analysis, DNA exploration
- Sport analysis and forecasting
- . . .

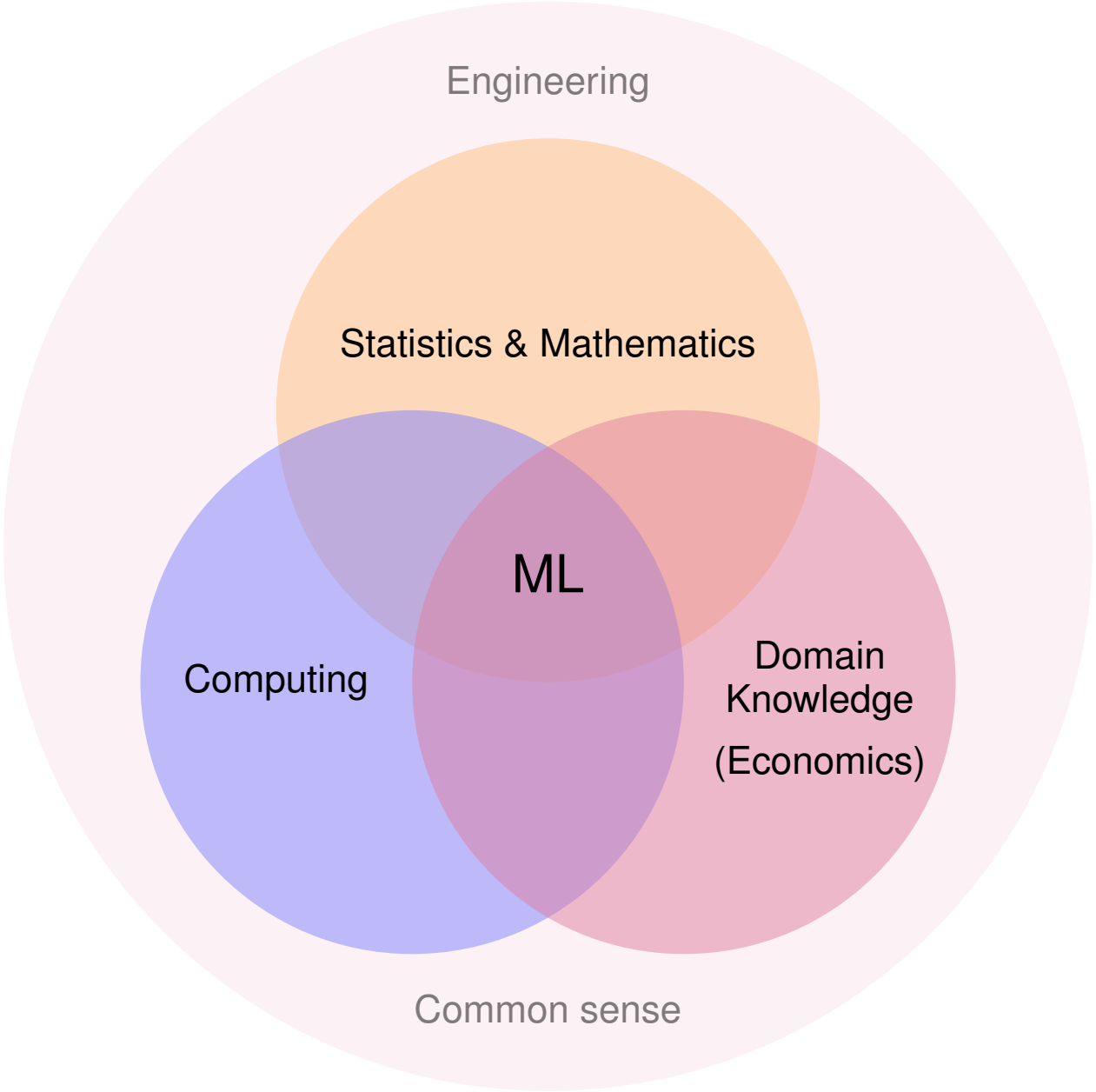
Is Machine Learning NEW? Why now?

It is not new but it is making **rapid progress**...

Why is it so popular now?

- **Because it works!** Both in business and science...
- Increasingly abundant and informative data
- Commoditized open-source software solutions
- Applicable across most business and scientific areas
- Computers have caught up with the mathematics...

Applied Statistical Learning for Economists



Applied Machine Learning – Heads Up

90% of your COMPUTER's time:
Actual computation – optimizing, resampling, cross-validating

90% of YOUR time:
Data cleaning, manipulation, and feature engineering, reports...

Feature engineering: fancy term for data transformations...

- load, clean, and normalize data
- treat outliers, ponder the role of missing data
- create new variables' transformations
- treat categorical data correctly
- ...

Applied Machine Learning – Software Available

It is largely **irrelevant** what environment you use.

Fitting Ridge Regression in `R`:

```
m.lasso <- glmnet(X, Y, alpha = 0, lambda = myLambdas)
```

Fitting Elastic Net Regression in `R`:

```
eNet <- glmnet(X, Y, alpha = 0.5, lambda = myLambdas)
```

Fitting Elastic Net in `MATLAB`:

```
eNet = lasso(X, Y, 'Alpha', 0.5, 'Lambda', myLambdas)
```

Fitting Elastic Net in `PYTHON/scikitLearn`:

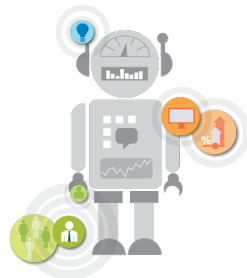
```
eNet = ElasticNet(alpha = myLambda, l1_ratio= 0.5)
eNet.fit(X, Y)
```

The **crucial thing** is to know what 'elastic net', `alpha` and `lambda` mean!

Human Learning about Machine Learning. . .

"A little learning is a dangerous thing;
drink deep, or taste not the Pierian spring:
there shallow draughts intoxicate the brain,
and drinking largely sobers us again."

Alexander Pope



TYPES OF MACHINE LEARNING

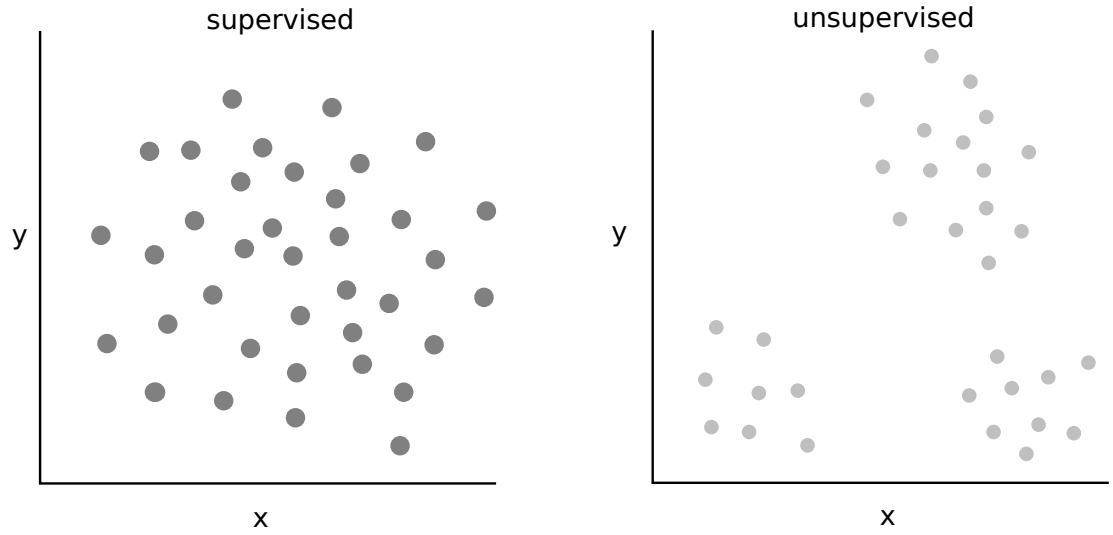
Basic Types of Machine Learning

1. **Supervised learning $\mathbf{y} = \mathbf{f}(\mathbf{x})$**
   - Using **labelled** data (matching *y*'s and *x*'s)
   - Explain outcome **y** as a function of **x**, to learn $p(\mathbf{y}|\mathbf{x})$
   - *Examples: regression and classification models . . .*

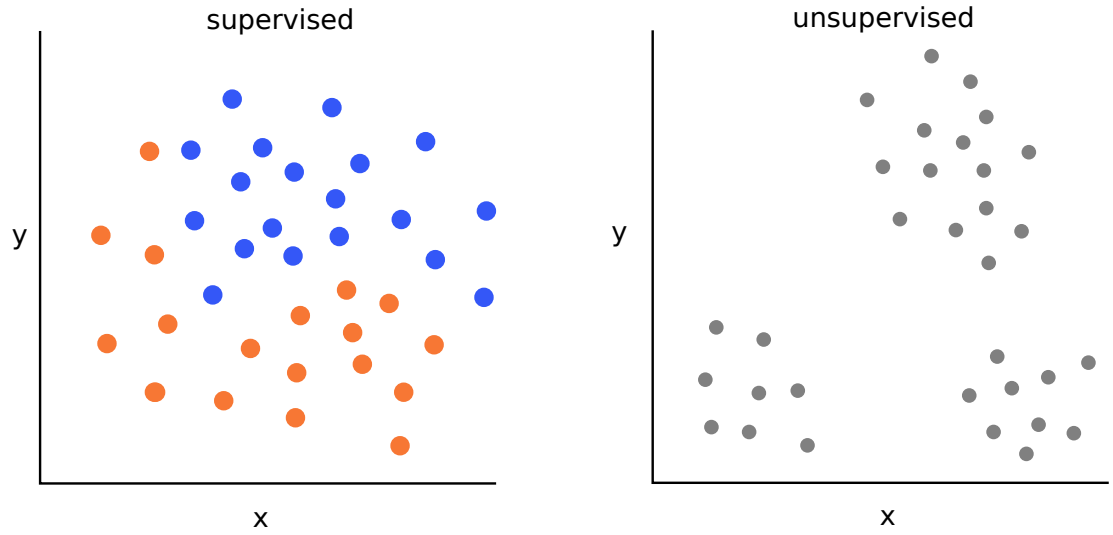
2. **Unsupervised learning**
   - No labeled data, all we know is **x**...
   - Learn properties of the joint data distribution $p(\mathbf{x})$
   - Pattern recognition, automated feature learning, dimensionality reduction, novelty/outlier detection, . . .
   - *Example: Clustering, principal components, auto-encoders, . . .*

3. **Reinforcement learning**
   - Software agents take actions to maximize rewards, reacting to environment. . .
   - *Example: beating humans in Go! ;)*

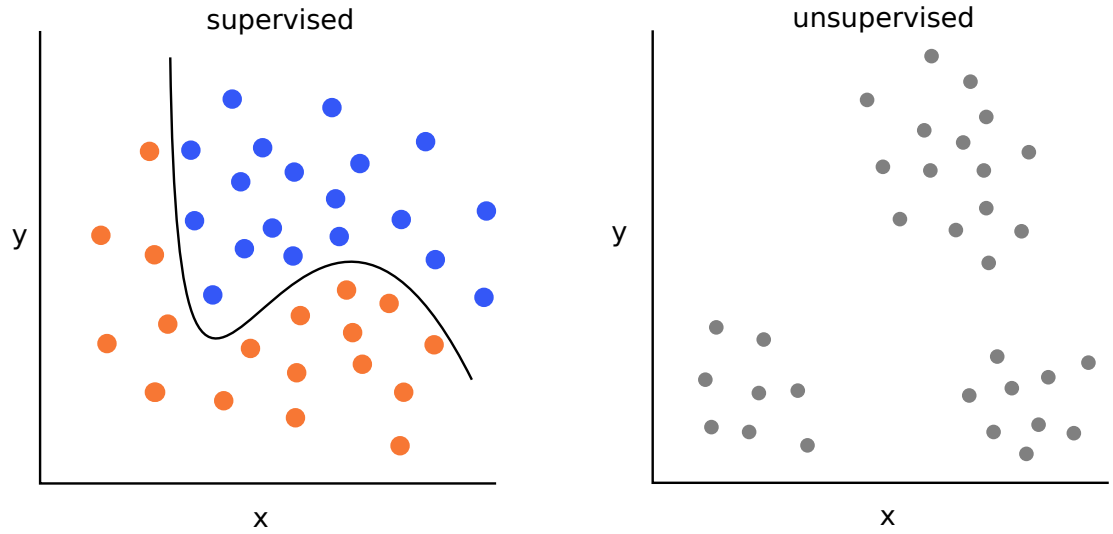
Supervised vs. Unsupervised



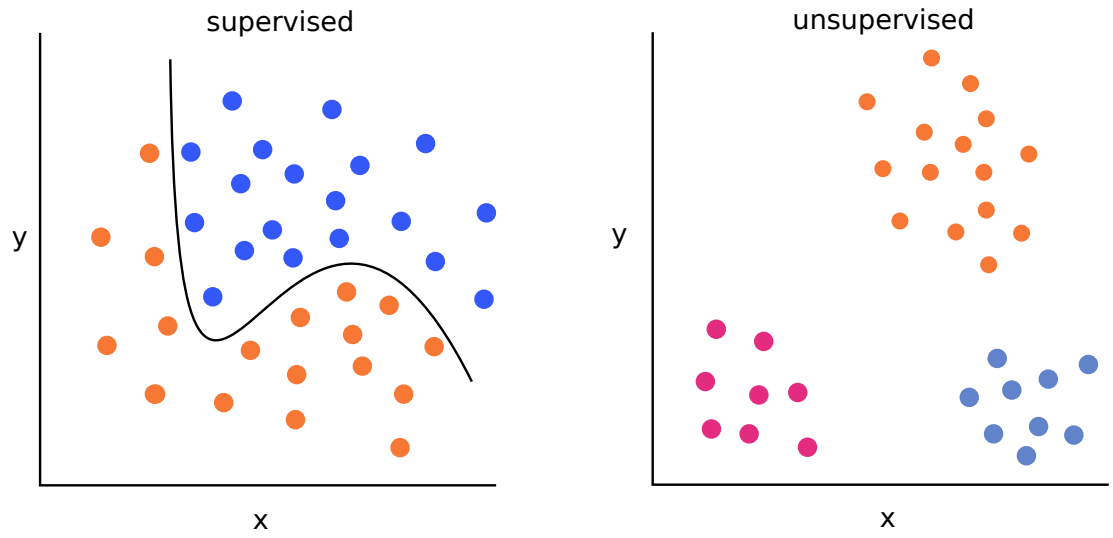
Supervised vs. Unsupervised

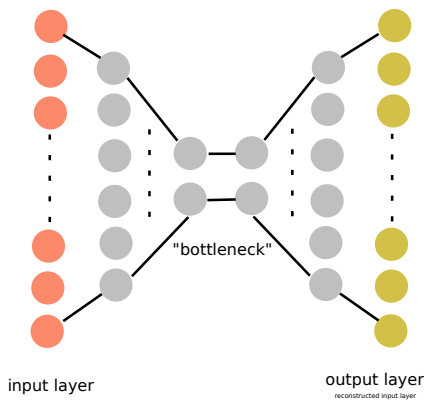


Supervised vs. Unsupervised



Supervised vs. Unsupervised





Supervised Learning: (1)

Explain outcome **y** as a mapping from **x**, to learn $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x})$, that **generalizes well** to new data.

1. **Regression:**
   - Output **y** is **quantitative**, input **x** unconstrained
   - E.g. GDP growth, income, stock price, ...

2. **Classification:**
   - Output **y** is **categorical**, input **x** unconstrained
   - Categories may be unordered or ordered
   - E.g.: crisis (`yes|no`), growth (`up|nil|down`), state (`low|mid|high`)

Key differences in choosing a **measure of success**: $D(y_i, \hat{y}_i)$.

Supervised Learning: (2)

Most supervised learners can be adapted to **both** regression and classification tasks.

Classification:

1. **Binary** – two classes
2. **Multiclass**
   - One-vs.-All (OvA):
     - N models created for N classes
     - red vs. (green+blue), blue vs. (red+green), green vs. (red+blue)
   - All-vs.-All (AvA):
     - N*(N-1)/2 models created for N classes
     - red vs. green, red vs. blue, . . . , blue vs. green

Supervised Learning: Type of $f(\mathbf{x})$

1. **Parametric:**
   - Uses known functional forms and **fixed** set of parameters
   - No matter how much data, number of parameters unchanged
   - Simplifies the process of learning
   - May limit what can be learned

   Example: (Non)-Linear Regression, Neural Nets, LDA, ...

2. **Non-Parametric:**
   - Flexible set of rules, adapts to given data, the form is not pre-determined
   - Easily deals with nonlinearity
   - Data hungry, simple to "overfit"

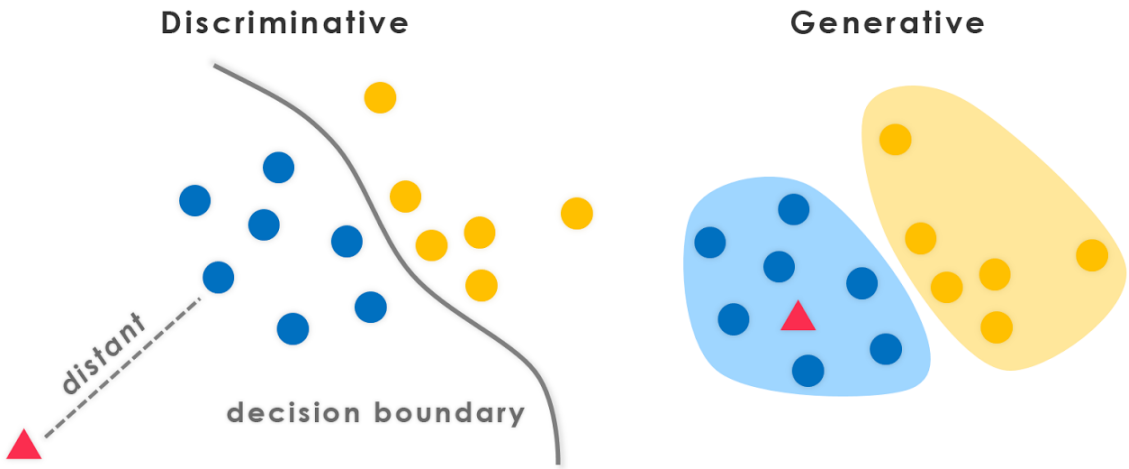
   Example: Clustering, k-Nearest Neighbors, Trees, Splines, ...

Here, param./non-param. is not a statement about **distributional assumptions**.

Discriminative vs. Generative Classification

1. **Discriminative models**
   - Learns the conditional prob. of y, given inputs: $p(y|x)$
   - Cannot generate (y, x) as no info about x
   - Focuses on boundaries between classes
   - You see zebra and elephant. . . which is zebra?

2. **Generative models**
   - Learns the joint probability of inputs and labels $p(x, y)$
   - Focuses on distribution of the classes, can generate $p(x|y)$
   - You saw zebras and elephants. . . draw a zebra!



Discriminative vs. Generative Classification

‘Do not learn more than you need to. . . ’

‘Folk Wisdom’ based on Vapnik (1998):

“one should solve the [classification] problem directly and never solve a more general problem as an intermediate step [such as modeling $p(x|y)$]”

However, Ng and Jordan (2001) challenge this view for ‘big data’ cases.

The Elements of Learning

The Problem:

An **unknown** mapping, u, from input space, $\mathcal{X}$, to output space, $\mathcal{Y}$, $u : \mathcal{X} \rightarrow \mathcal{Y}$, such that $y_j = f(x_j)$.

Hypothesis Set:

A set $\mathcal{R}(f)$ containing candidate mappings (models, hypotheses) mean to approximate the unknown mapping u.

Observed Data:

A finite set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$, or $\mathcal{D} = \{(x_1, x_2), \ldots, (x_k, x_k)\}$.

Algorithm:

An algorithm, $\mathcal{A}$, that uses observed data, $\mathcal{D} \in$, to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$, $f \in \mathcal{R}(f)$.

Error Measure (Distance/Loss/Utility Function)

A metric $\mathcal{E}$ used by the algorithm $\mathcal{A}$ to choose a mapping g.

Prediction and Causal Inference (A)

Most widely-known application of ML/SL are **predictive**.

- Image recognition, voice recognition, ...
- Economic forecasting/nowcasting, FX forecasting, stock analytics
- Default prediction
- ...

Statitistical learning can be used also for **causal inference**.

- Flexible models for propensity estimation, more robust to specification error
- Targeted Learning (van der Laan and Rose)
- Causal Trees (Athey et al.)
- ...

We'll be dealing mostly with predictive applications.

ML: Methods & Principles

1. **METHODS:**
   - **flexible** models/algorithms with strong representational capacity
   - handling large amount of explanatory variables
   - focusing on flexible, non-linear models

2. **PRINCIPLES:**
   - **truth unknown**, use flexible tools and Occam's razor
   - strong focus on **out-of-sample performance** (generalization)
   - choosing the model that does not **overfit** the in-sample information (**training sample**) by using **test** samples
   - techniques for **adaptive** representational capacity, **regularization** (shrinkage, priors, . . .)

Let's make these concepts more specific. . .

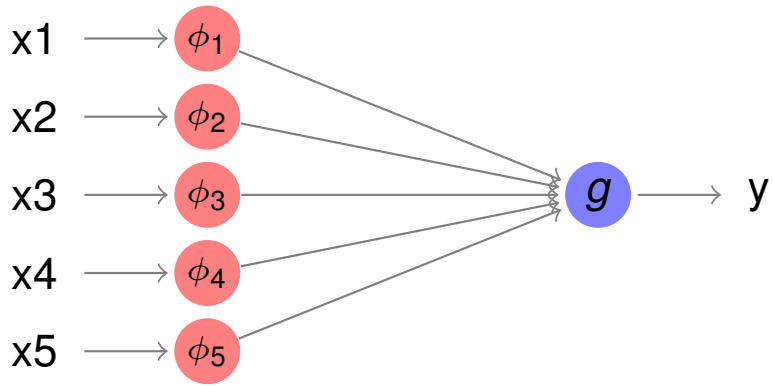
Prominent Examples of Supervised Learners, $f(\mathbf{x})$

1. **Linear Regression**
2. **Artificial Neural Network**

3. **Trees**
4. **k-Nearest Neighbors**

Example #1: [Linear] Regression

$$y_i = g[\beta_0 + \beta_1\phi_1(x_1) + \cdots + \beta_k\phi_k(x_k)]$$



Linear regression with $g(x) \equiv 1 \times x$, logistic regression, $g(x) = \text{logit}^{-1}(x)$

In practice, the Linear regression and/or logistic regression (classifier) are often estimated by other methods than 'OLS' (ridge regression, LASSO, . . .)

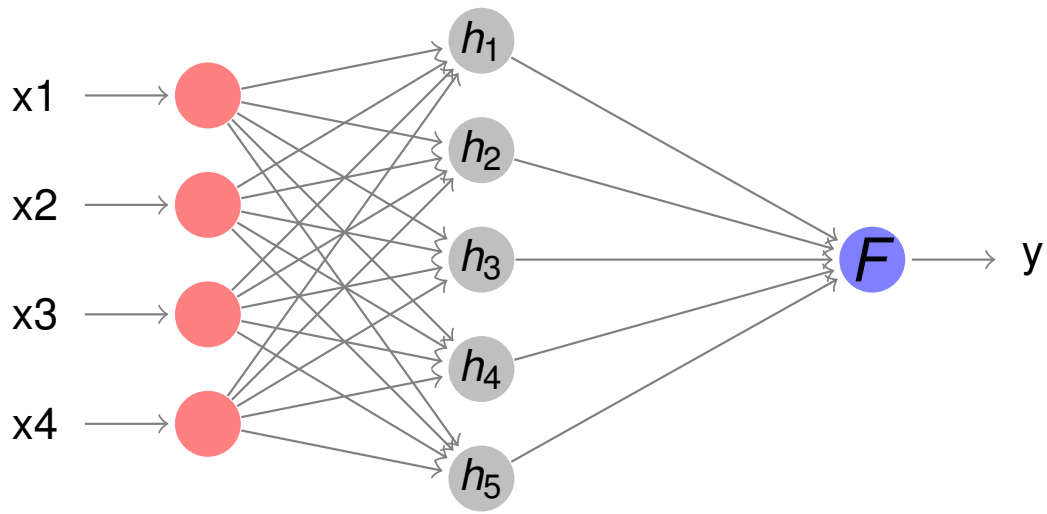
Example #2: Neural Net

Artificial Feed-Forward Neural Network with One Hidden Layer

(Neural networks are universal approximants.)

$$
\begin{aligned}
y &= F(\beta_0 + \beta_1 h_1 + \cdots + \beta_r h_r) \\
h_j &= f(\lambda_{0,j} + \lambda_{1,j} x_1 + \cdots + \lambda_{k,j} x_k)
\end{aligned}
$$

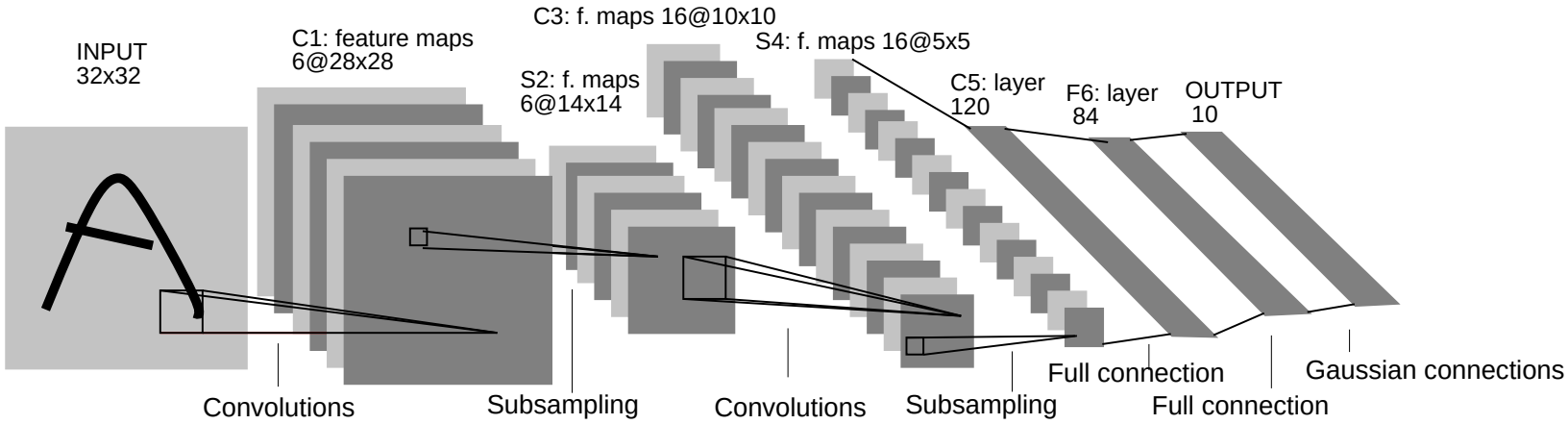
Input layer Hidden layer Output layer

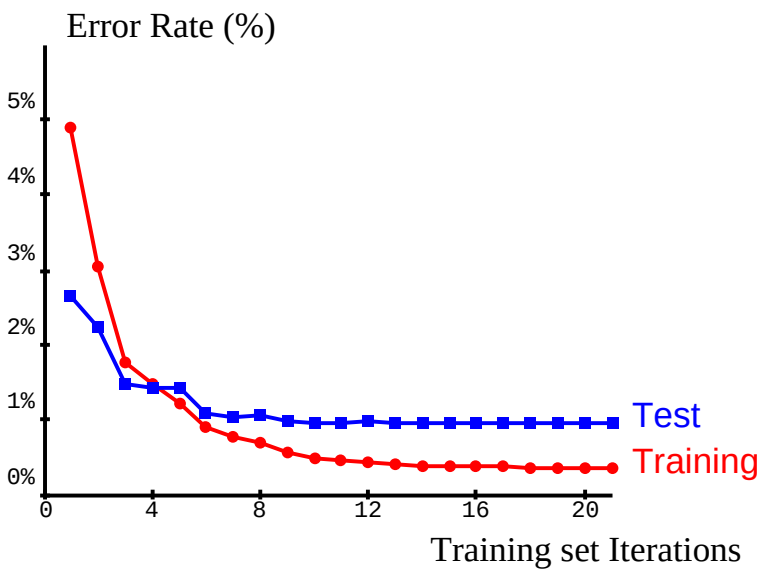


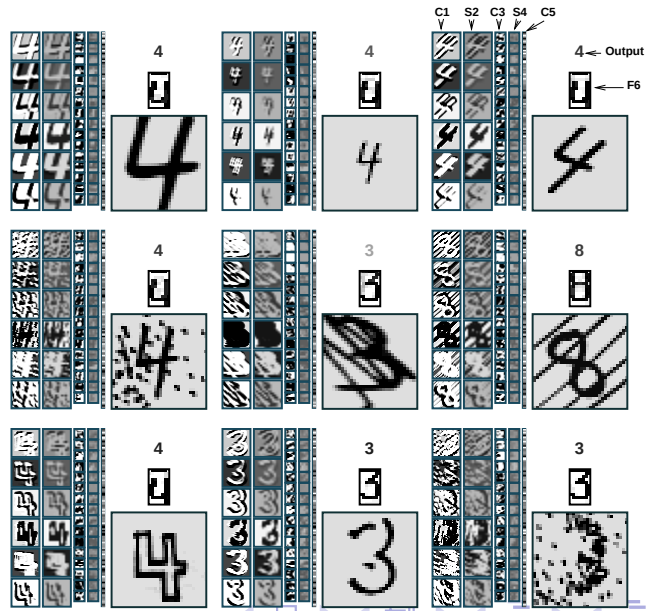
Functions F and f are **nonlinear activation** functions.

Example #2b: 'The First' Convolutional Network

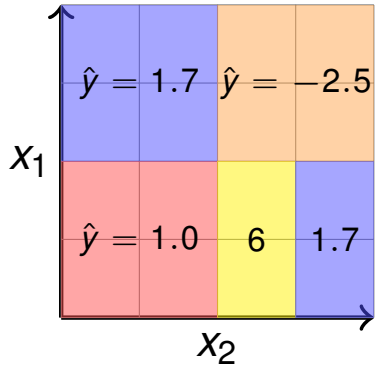
Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner:
Gradient-Based Learning Applied to Document Recognition, Proc. of the IEE, November 1998

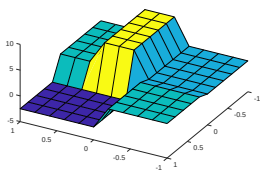


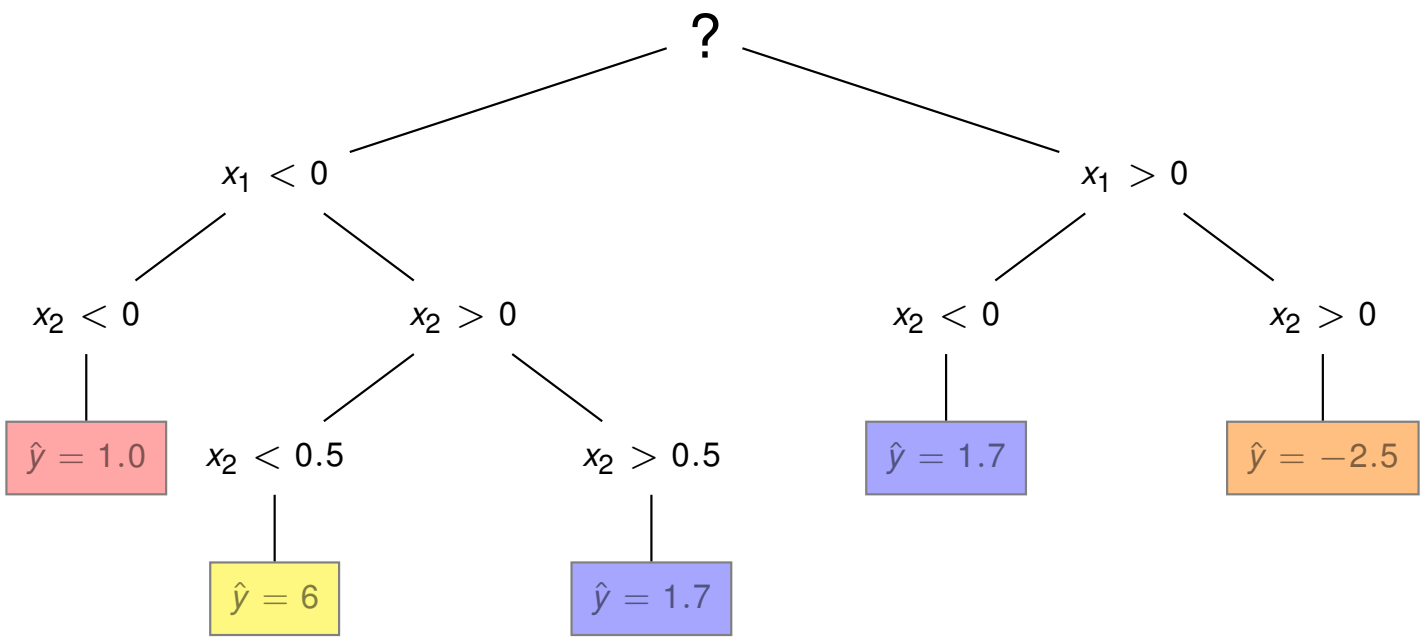




Example #3: Trees

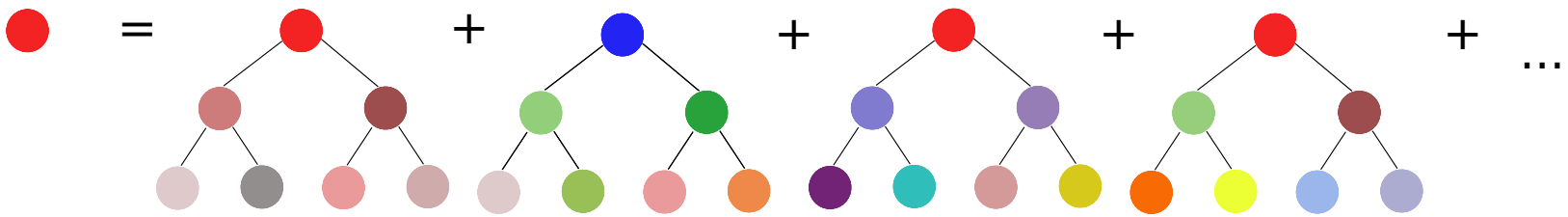




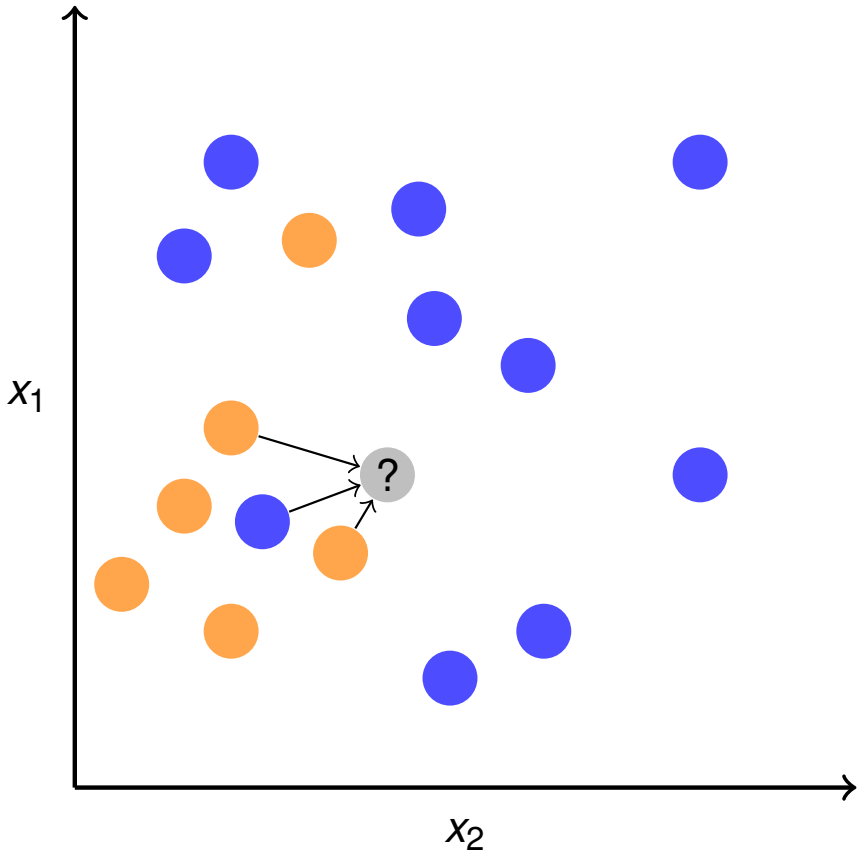


Example #3b: Tree Ensembles

Random Forests, Bagged Trees



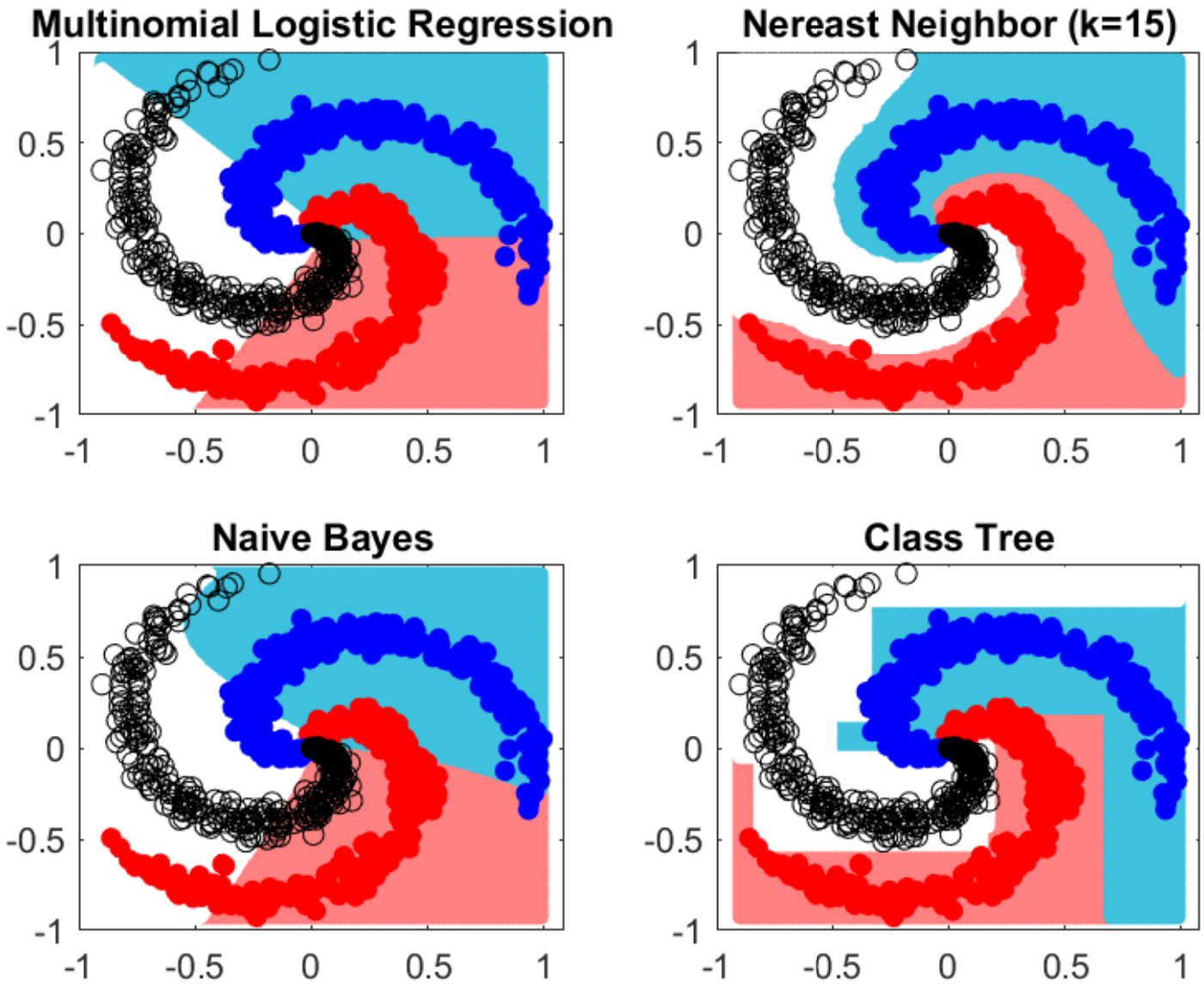
Example #4: k-Nearest Neighbors (kNN)



? $= 2\times$ $+ 1\times$ $=$

kNN with $k = 3$
x_1, x_2 categorical

Classification Examples. . .



No Free Lunch Theorem (Vapnik)

If you make no assumption about the data, there is NO reason to prefer one model over any other models. . .

A priori, no model is guaranteed to outperform the others.

Generalization and Overfitting

A hallmark of statistical/machine learning is the emphasis on models to **generalize** well to new, out-of-sample data

A model that performs much worse on out-of-sample data than on in-sample (training) data is **overfit**.

If the **model complexity** (capacity, degrees of freedom) is in excess of the information in data sample, the model may overfit. . .

ML provides set of techniques and principles focused on **combating overfitting** (regularization, cross-validation, . . .)

Thank you for your patience. . .