Machine Learning for Clinicians:

Advances for Multi-Modal Health Data

Michael C. Hughes

A Tutorial at MLHC 2018, August 16, 2018

PART 1: Making and Evaluating Predictions

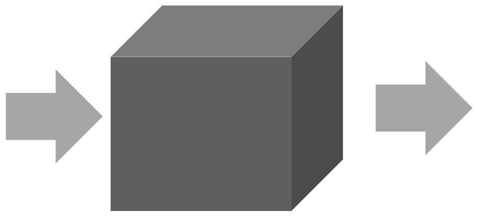
Evaluate: Confusion matrix, ROC curve, calibration, utilities
Predict: Linear/logistic regression, Decision Trees & Rand Forests,
Neural Nets, Gaussian Processes

Slides / Resources / Bibliography:
https://michaelchughes.com/mlhc2018_tutorial.html

PART 1: Making and Evaluating Predictions

Prediction for example n

x_n vector with D dims. $[x_{n1} \;\; x_{n2} \;\; \ldots \;\; x_{nD}]$



y_n scalar

"features" or "attributes"

"covariates"

"independent variables"

"outcomes"

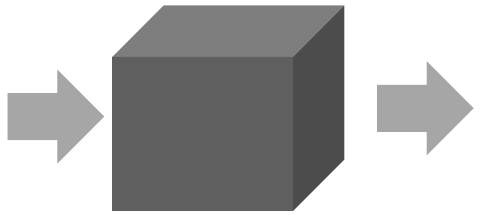
"targets"

"dependent variable"

Prediction Tasks

Regression

$[x_{n1} \;\; x_{n2} \; \ldots \; x_{nD}]$

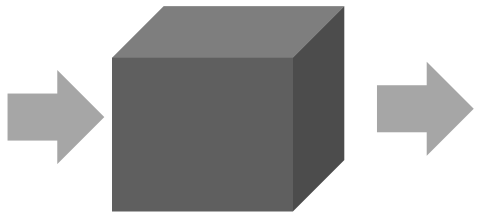


any real number

-3.4, 17.56, …

Binary Classification

$[x_{n1} \;\; x_{n2} \; \ldots \; x_{nD}]$

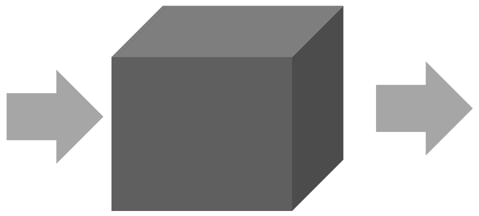


2 categories

0 or 1

Multi-class Classification

$[x_{n1} \;\; x_{n2} \; \ldots \; x_{nD}]$



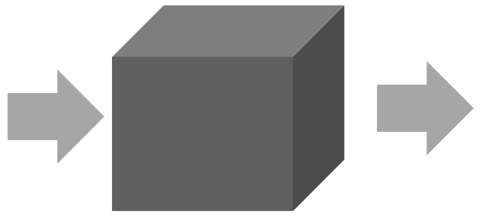
C categories

1,2,3, … or C

Probabilistic Prediction Tasks

Regression

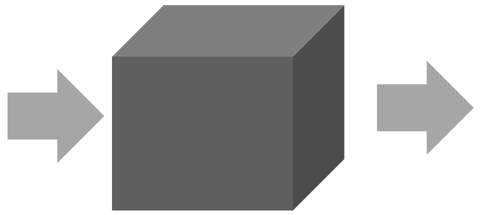
$[x_{n1} \;\; x_{n2} \;\; \ldots \;\; x_{nD}]$



pdf over reals

Binary Classification

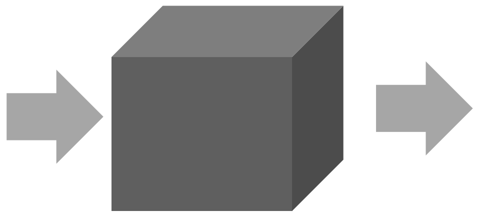
$[x_{n1} \;\; x_{n2} \;\; \ldots \;\; x_{nD}]$



Pr(y_n = 1)

Multi-class Classification

$[x_{n1} \;\; x_{n2} \;\; \ldots \;\; x_{nD}]$



Pr(y_n =class 1)
Pr(y_n =class c)
…
Pr(y_n =class C)

Evaluation First!

Recommendations:

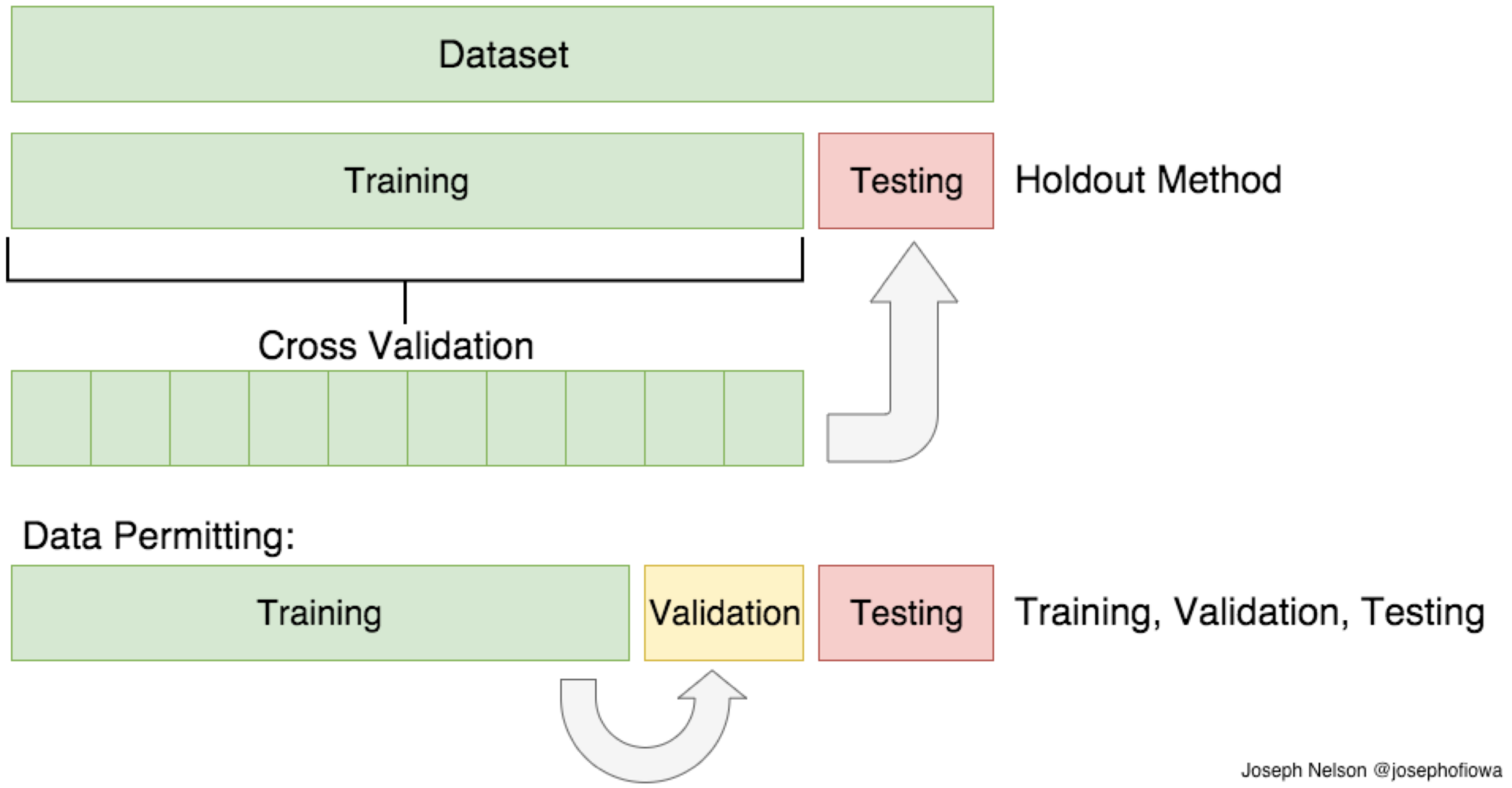
1) Spend as much time on designing evaluation as you do with model prototyping

2) Make diagnostic plots, not just tables

3) How to measure actual utility?

   Days of life extended, Dollars saved, etc.

Splitting Dataset: Train/Valid/Test

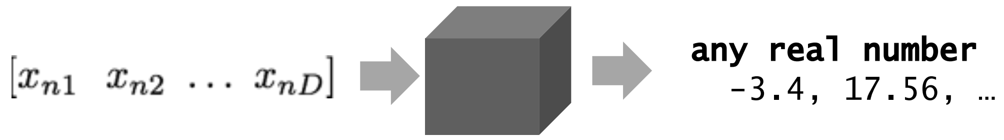


Random splits often not enough for healthcare applications!

Splitting Strategies

- Split by patient
  - Will my method generalize to new subjects?
- Split by hospital site
  - Will my method generalize to new doctors?
- Split by year
  - Is my method sensitive to specific transient features of the health system?

Perf. Metrics for Regression



Mean Squared Error

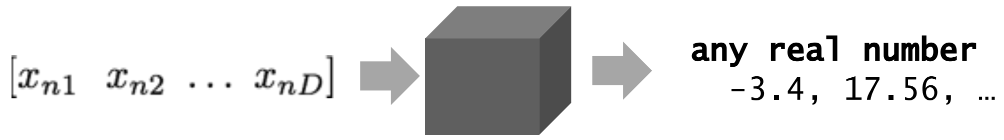
$$\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2$$

Mean Absolute Error

$$\frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n|$$

These metrics have units! (days, dollars, etc.)
Hard to interpret alone.
Need to be compared to baselines (simpler models).

Perf. Metrics for Regression



Predictive R^2

$$1 - \sum_{n=1}^{N} \frac{(y_n - \hat{y}_n)^2}{(y_n - \bar{y})^2}$$

$$\bar{y} = \mathrm{mean}(y_1, y_2, \dots y_N)$$

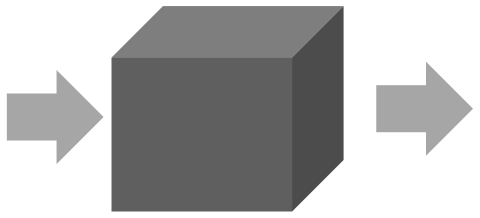
Unit-less
Best possible: 1.0
Worst possible: -inf

PR^2 of 0.0 means the predictions are as good as guessing the dataset mean.

Good practice: Report predR^2 *and* mean error

Perf. Metrics for Probabilistic Binary Classifier

$[x_{n1} \;\; x_{n2} \;\; \ldots \;\; x_{nD}]$

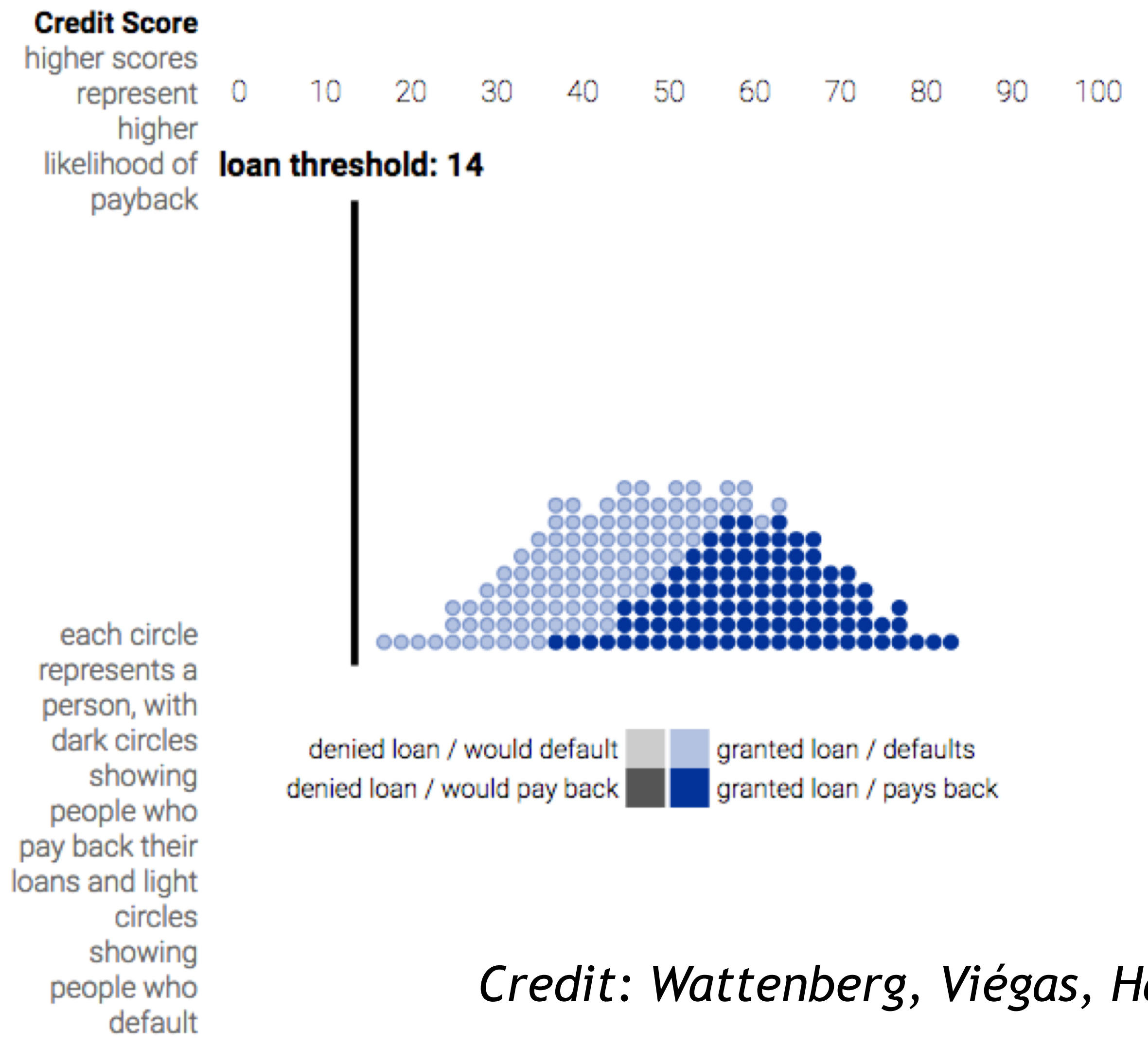


Pr(y_n = 1)

or

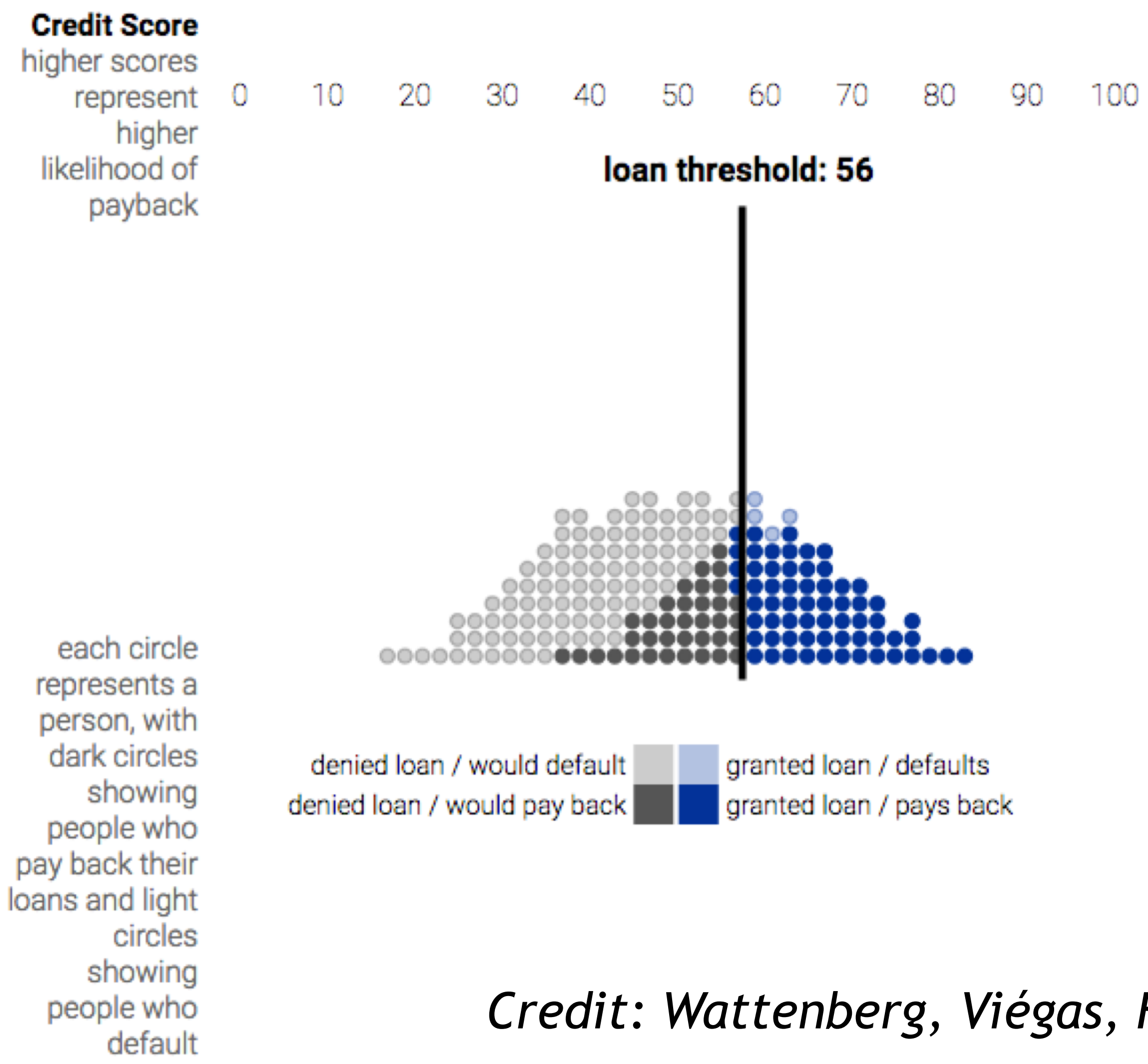
real score value

Thresholding to get Binary Decisions



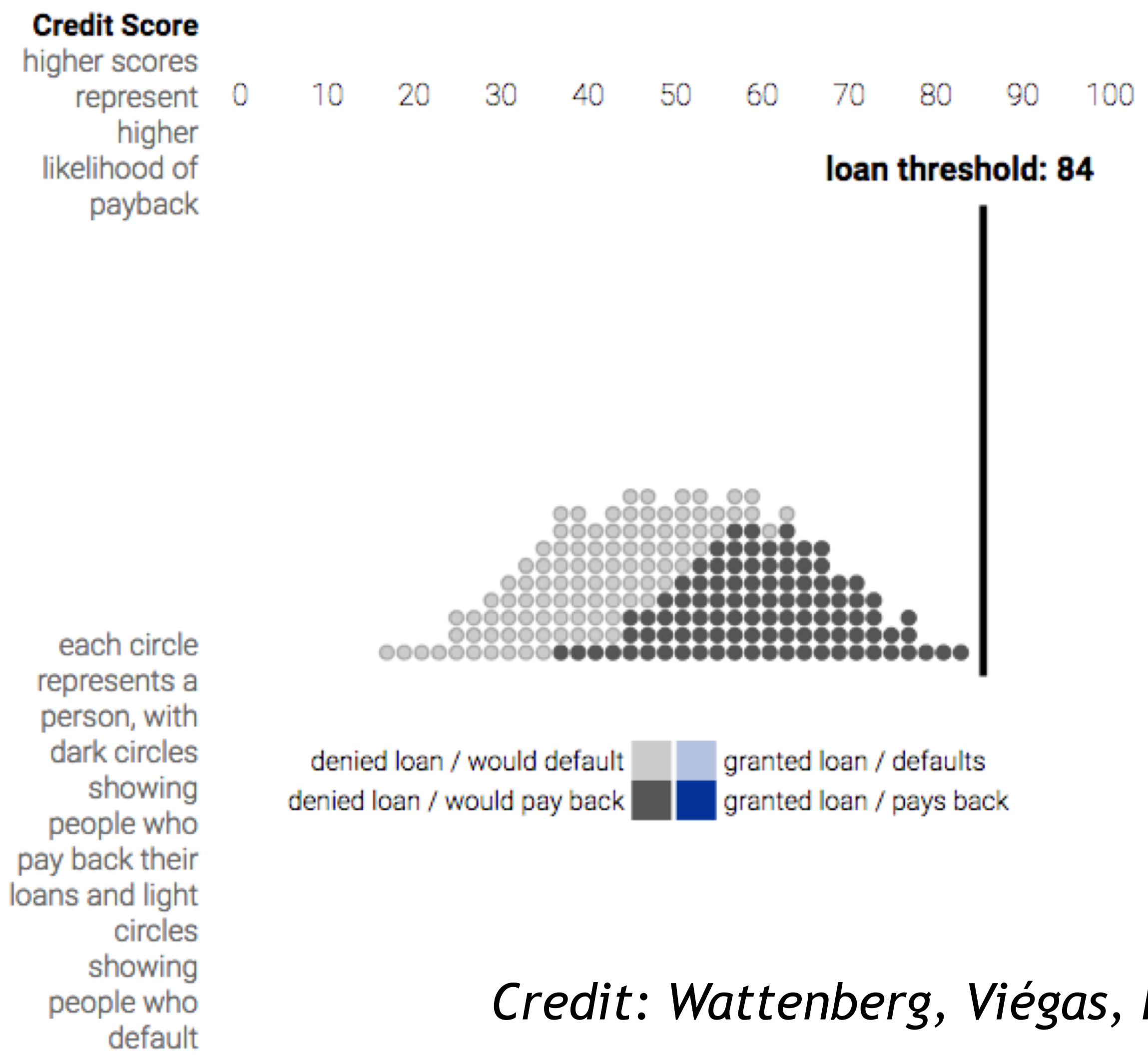
Credit: Wattenberg, Viégas, Hardt

Thresholding to get Binary Decisions



Credit: Wattenberg, Viégas, Hardt

Thresholding to get Binary Decisions



Credit: Wattenberg, Viégas, Hardt

Performance Metrics for Binary Classifiers

Two kinds:

1) Evaluate particular threshold

   Accuracy, TPR, FPR, PPV, NPV, etc.

2) Evaluate across range of thresholds

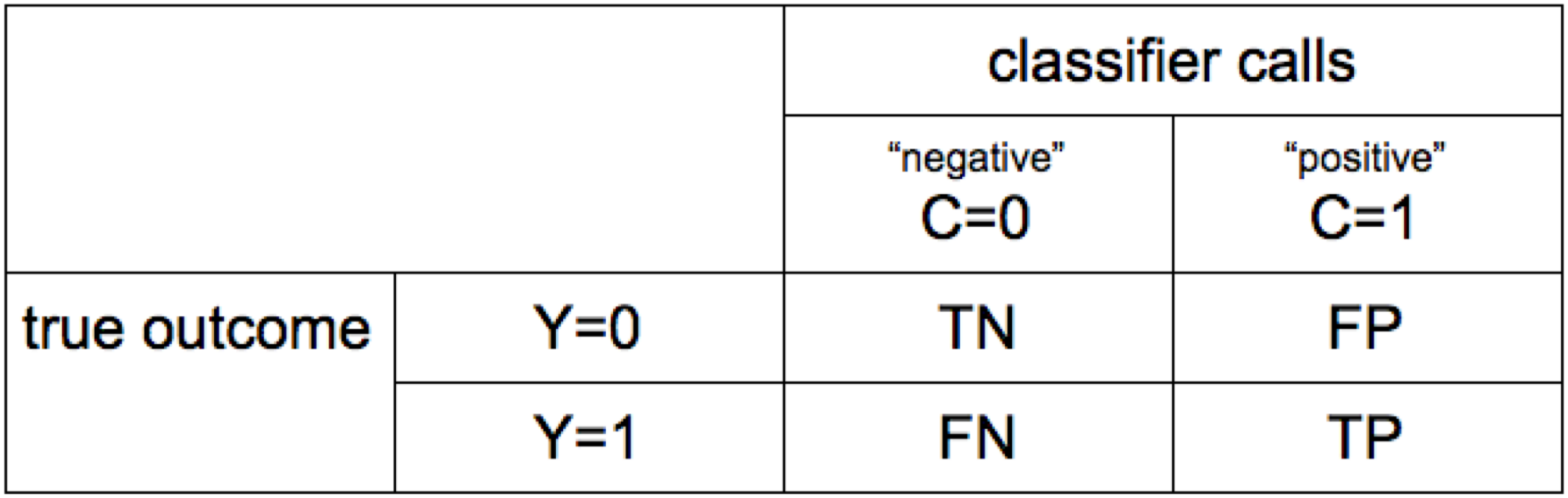
   ROC curve, Precision-Recall curve

Confusion Matrix:
Which mistakes do I make at given threshold?

Fundamental counts:

TN : true negative	TP : true positive
FN : false negative	FP : false positive

total num patients = TP + FP + TN + FN



		classifier calls	
		"negative" C=0	"positive" C=1
true outcome	Y=0	TN	FP
	Y=1	FN	TP

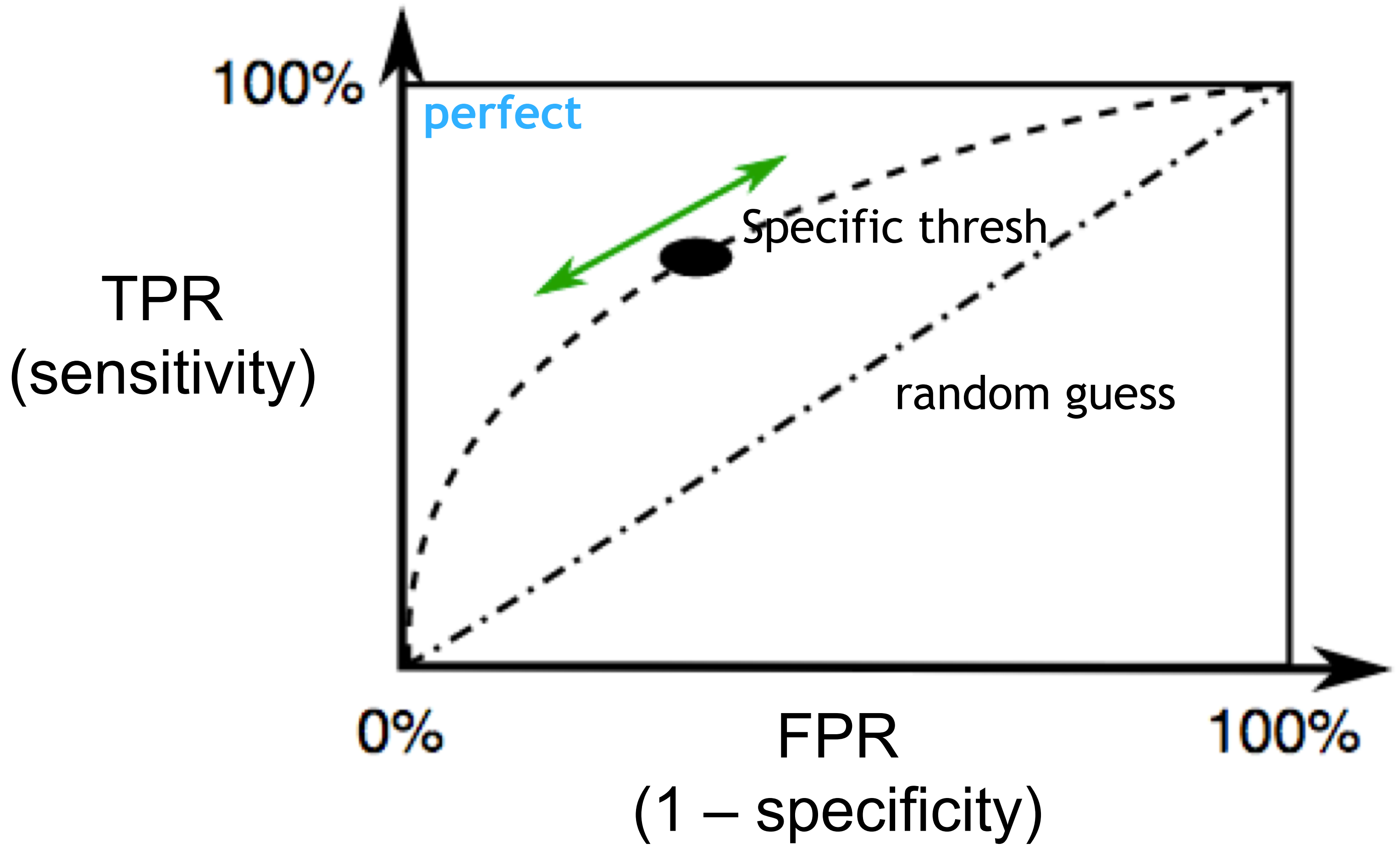
This table is called "confusion matrix". Always good to report it.

Metrics for Specific Threshold

METRIC	FORMULA	IN WORDS **"Probability that …" Or "How often the …"**	EXPRESSION
True Positive Rate (TPR)	TP --------------------- TP + FN	subject who is positive will be called positive	Pr(C = 1 | Y = 1)
False Positive Rate (FPR)	FP --------------------- FP + TN	subject who is negative will be called positive	Pr(C = 1 | Y = 0)
Positive Predictive Value (PPV)	TP --------------------- TP + FP	subject called positive will actually be positive	Pr(Y = 1 | C = 1)
Negative Predictive Value (NPV)	TN --------------------- TN + FN	subject called negative will actually be negative	Pr(Y = 0 | C = 0)

Use the metrics appropriate for your application.

ROC Curve (across thresholds)

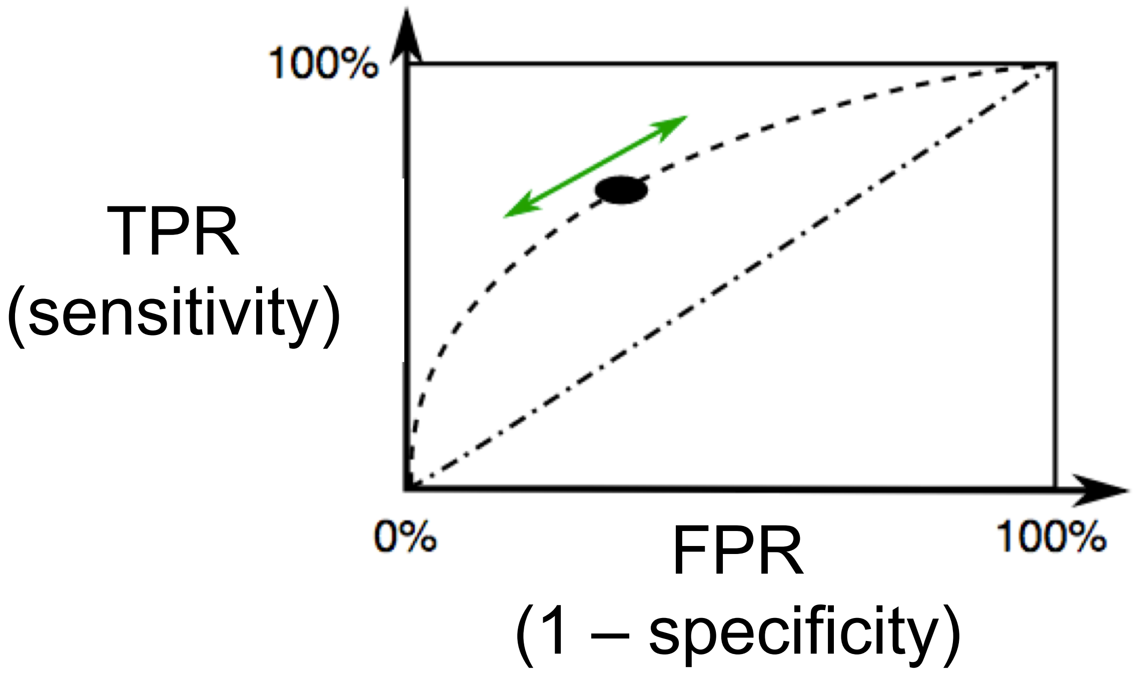


Area under ROC curve (AUROC or AUC)

Area varies from 0.0 – 1.0.   0.5 is random guess. 1.0 is perfect.

aka "C-statistic"

Graphical:

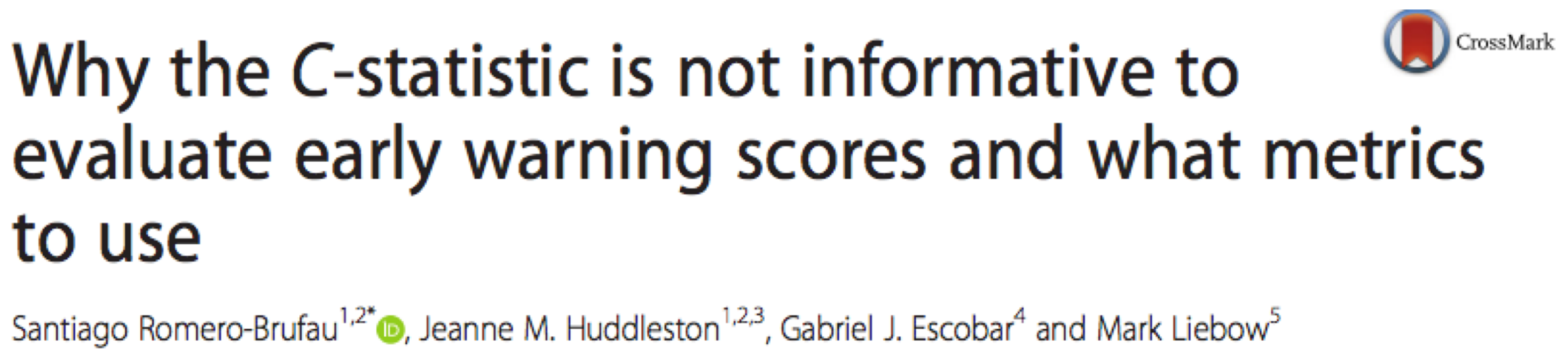


Probabilistic:

$$\text{AUROC} \triangleq \Pr(\hat{y}(x_i) > \hat{y}(x_j) | y_i = 1, y_j = 0)$$

For random pair of examples, one positive and one negative,
What is probability classifier will rank positive one higher?

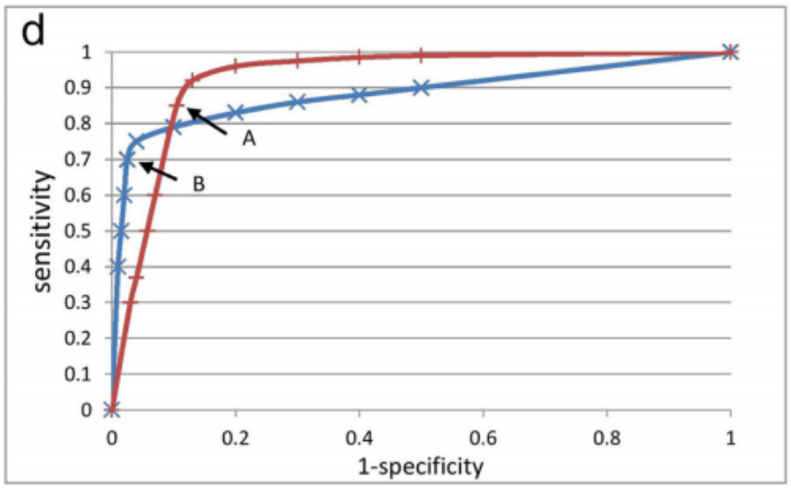
AUROC not always best choice



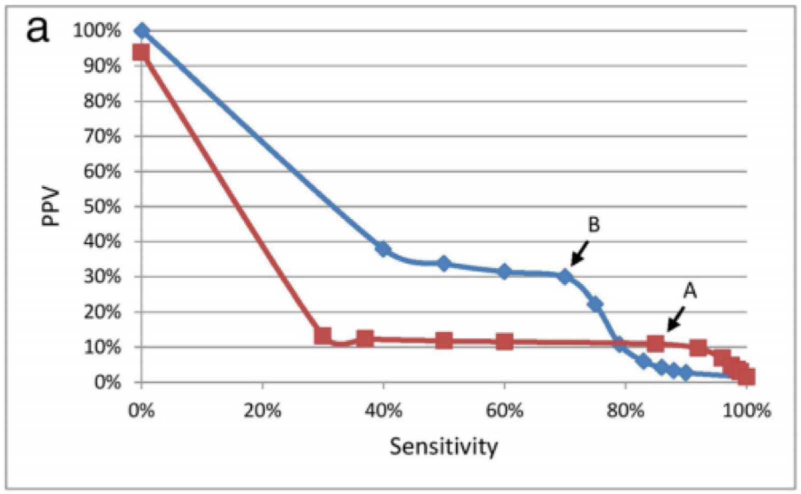
Why the C-statistic is not informative to evaluate early warning scores and what metrics to use

CrossMark

Santiago Romero-Brufau[1,2*], Jeanne M. Huddleston[1,2,3], Gabriel J. Escobar[4] and Mark Liebow[5]



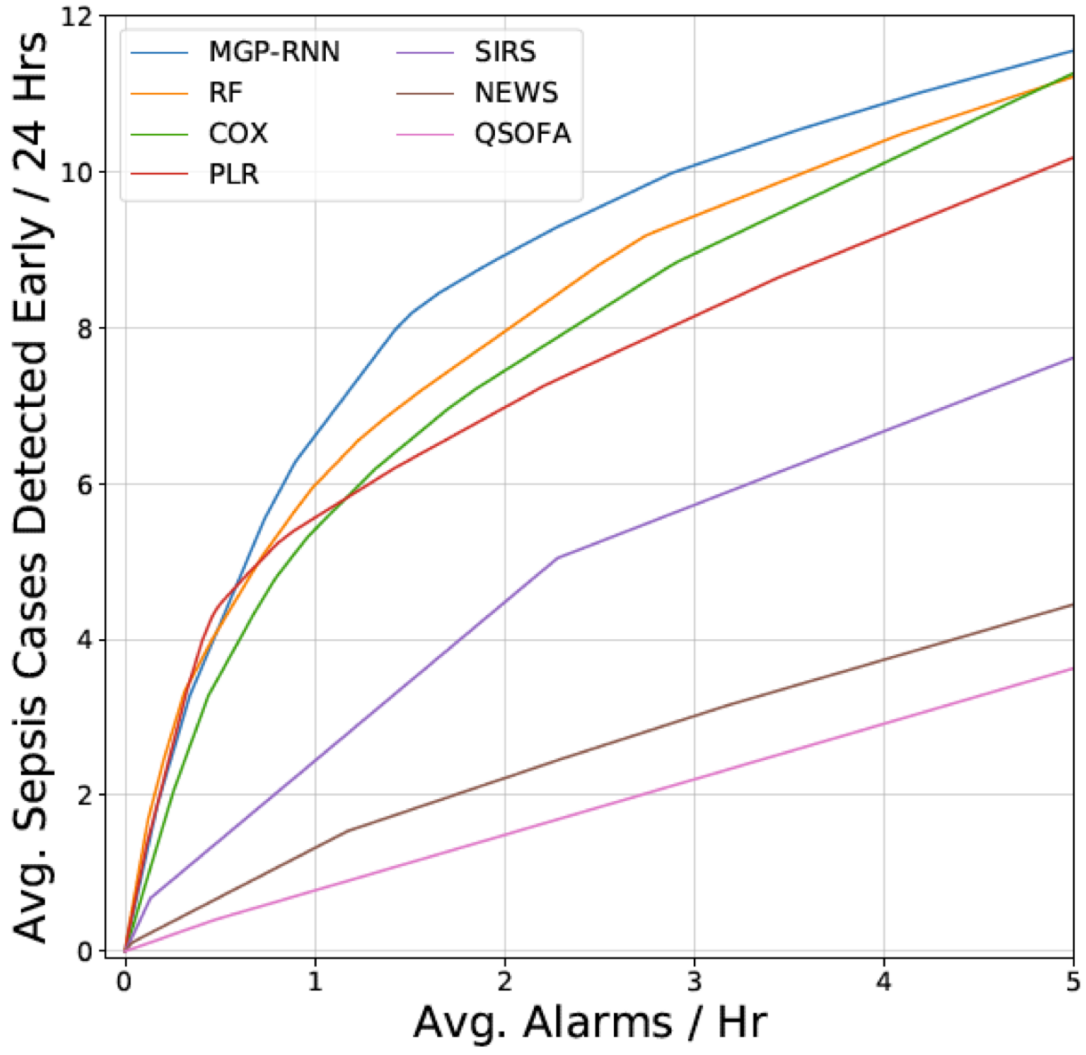
AUROC: red is better



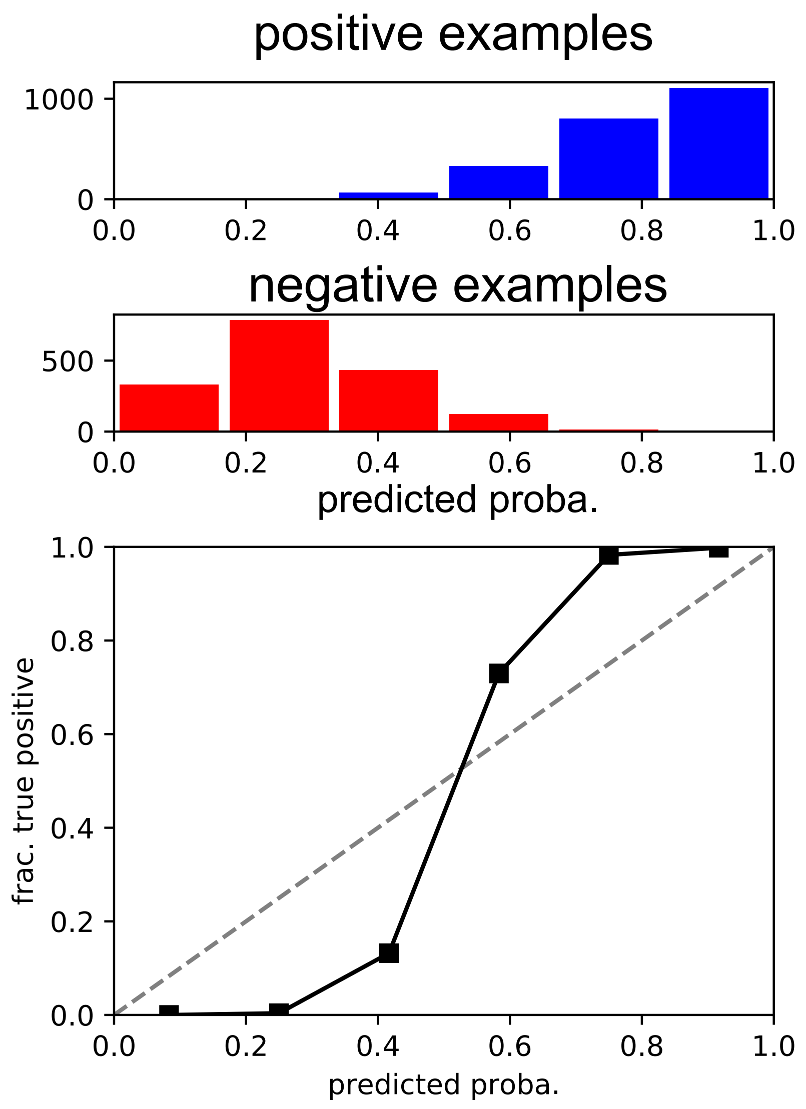
Blue much better for alarm fatigue

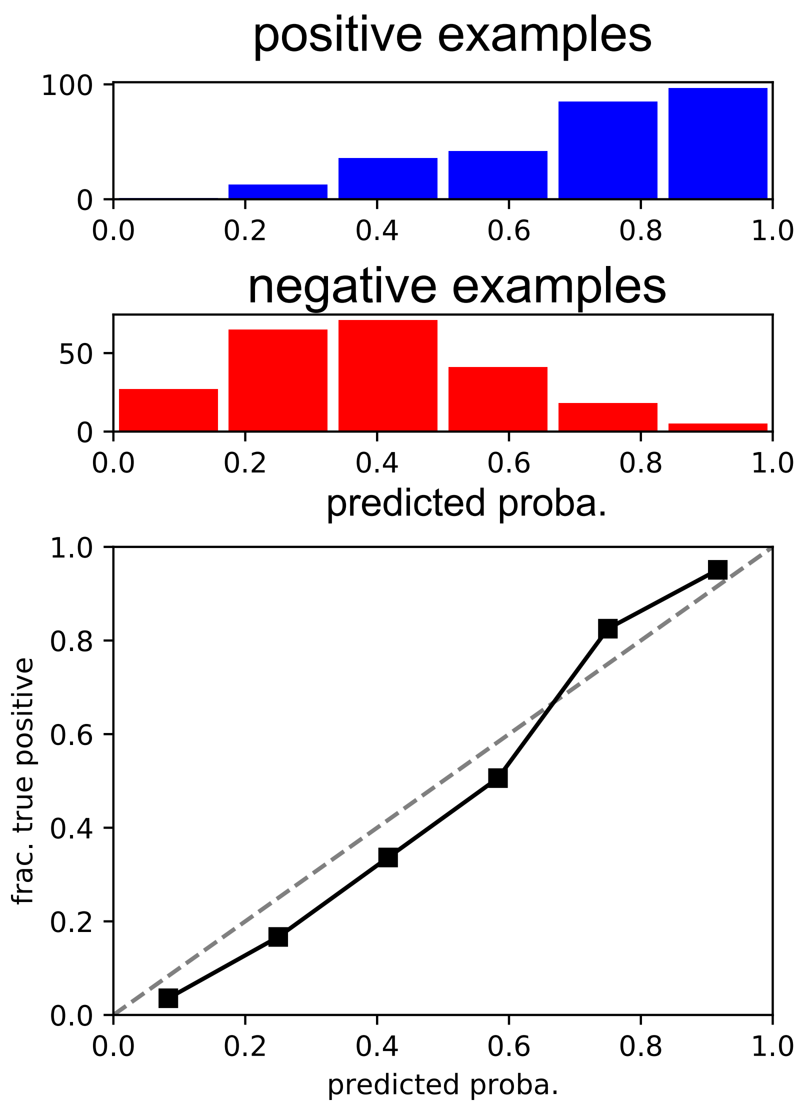
Domain Specific Evaluation!

Futoma et al. 2017
Sepsis risk

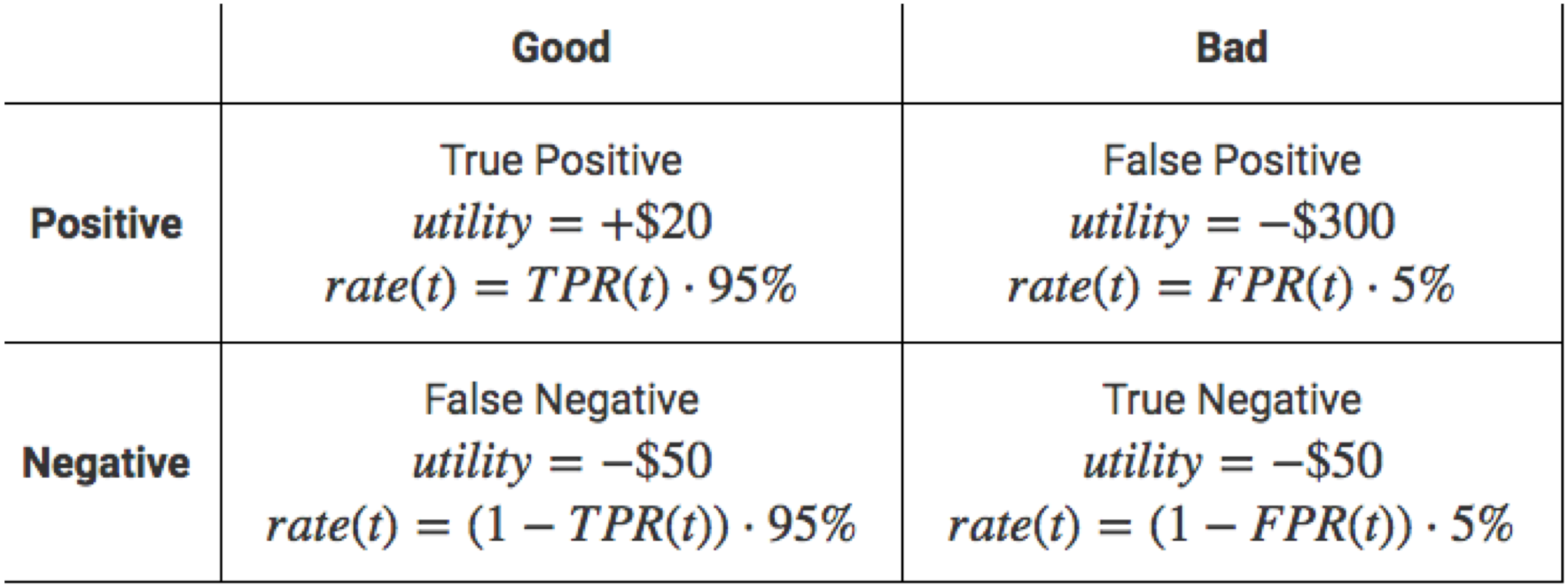


Classifier Calibration





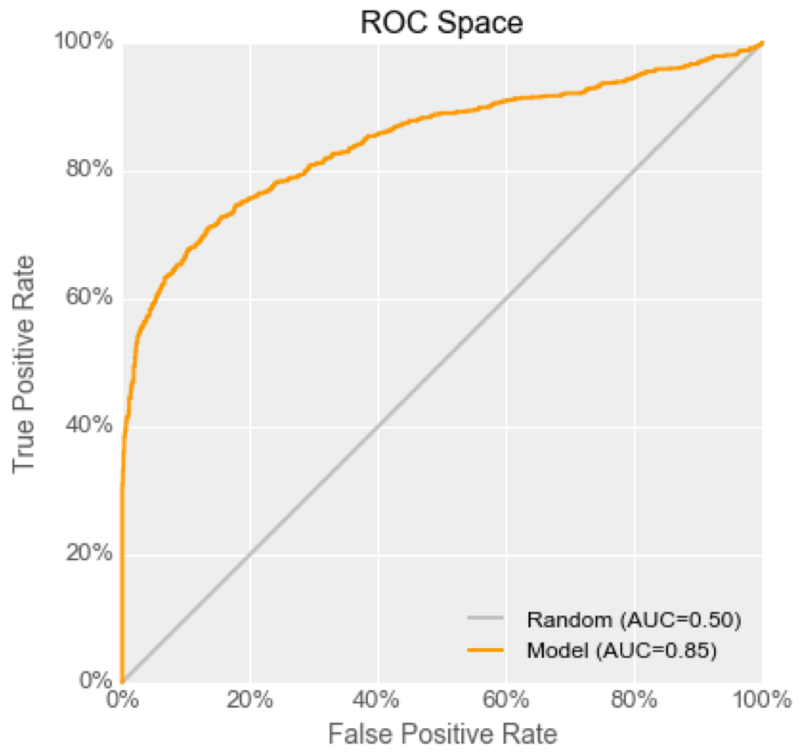
Quantifying costs of decisions

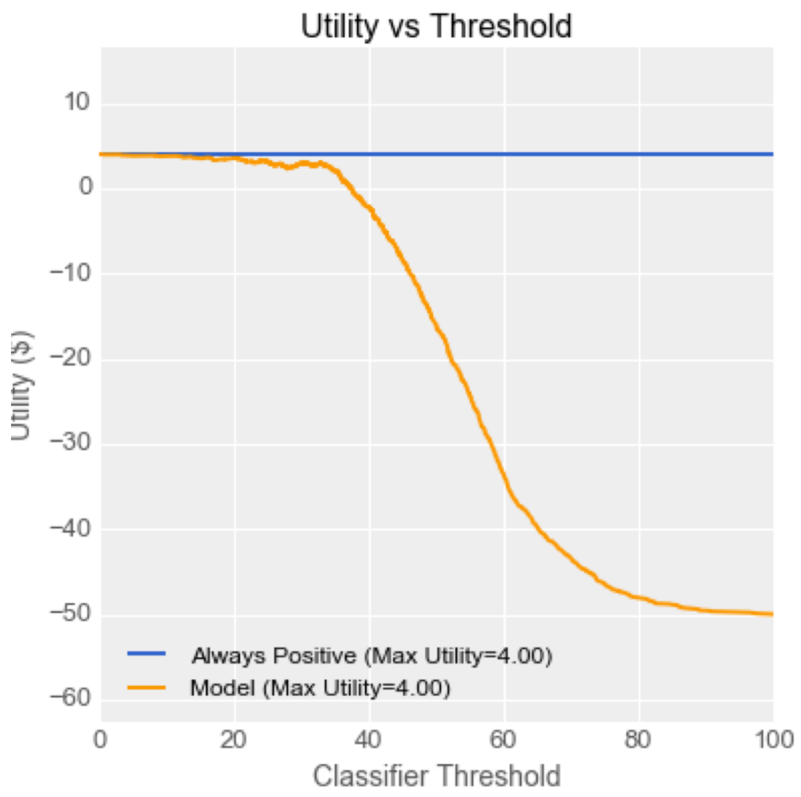


	Good	Bad
Positive	True Positive $utility = +\$20$ $rate(t) = TPR(t) \cdot 95\%$	False Positive $utility = -\$300$ $rate(t) = FPR(t) \cdot 5\%$
Negative	False Negative $utility = -\$50$ $rate(t) = (1 - TPR(t)) \cdot 95\%$	True Negative $utility = -\$50$ $rate(t) = (1 - FPR(t)) \cdot 5\%$

Credit: Nicolas Kruchten
http://blog.mldb.ai/blog/posts/2016/01/ml-meets-economics/

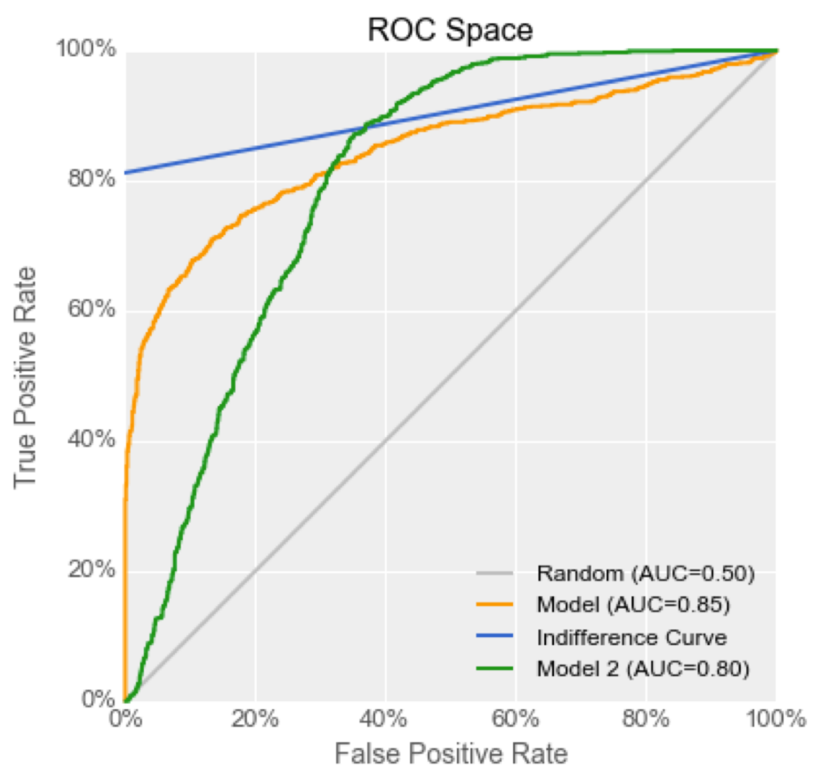
Making decisions from classifiers

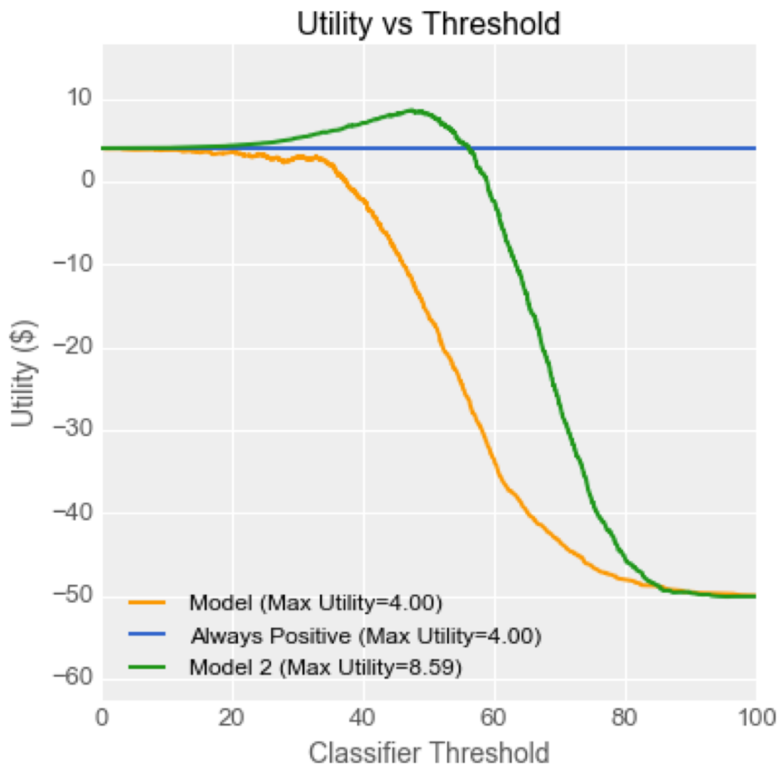




Credit: Nicolas Kruchten
http://blog.mldb.ai/blog/posts/2016/01/ml-meets-economics/

Making decisions from classifiers



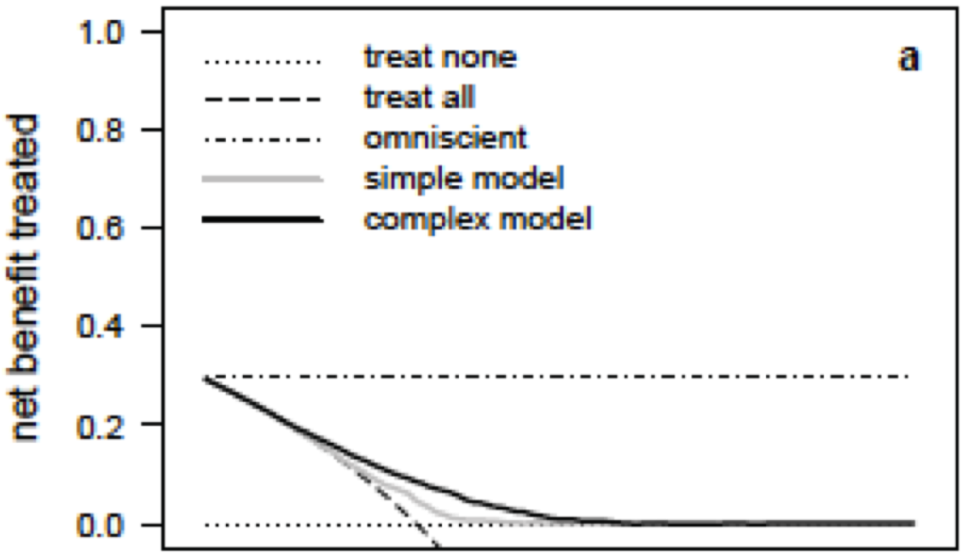


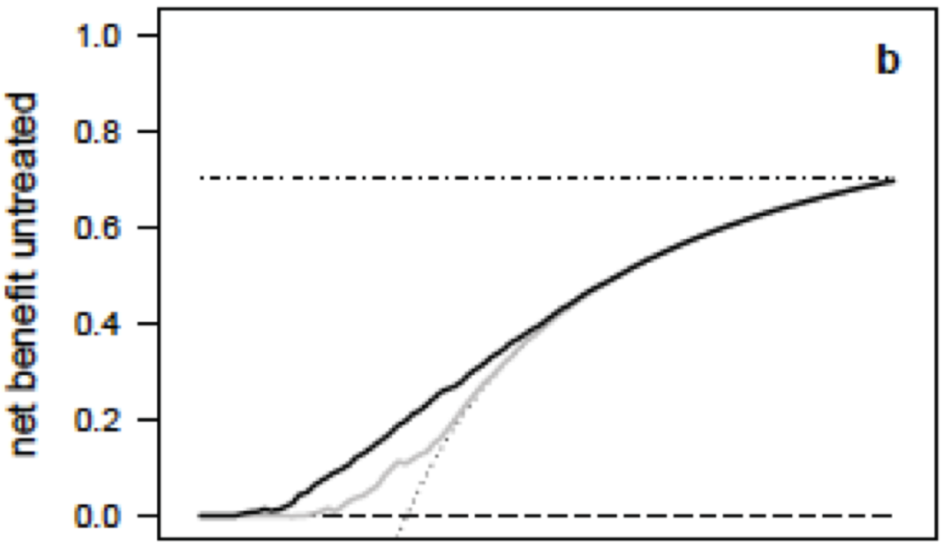
Model 2 (green) has lower AUC

… but has operating points with much higher utility!

Credit: Nicolas Kruchten
http://blog.mldb.ai/blog/posts/2016/01/ml-meets-economics/

Decision Curve Analysis





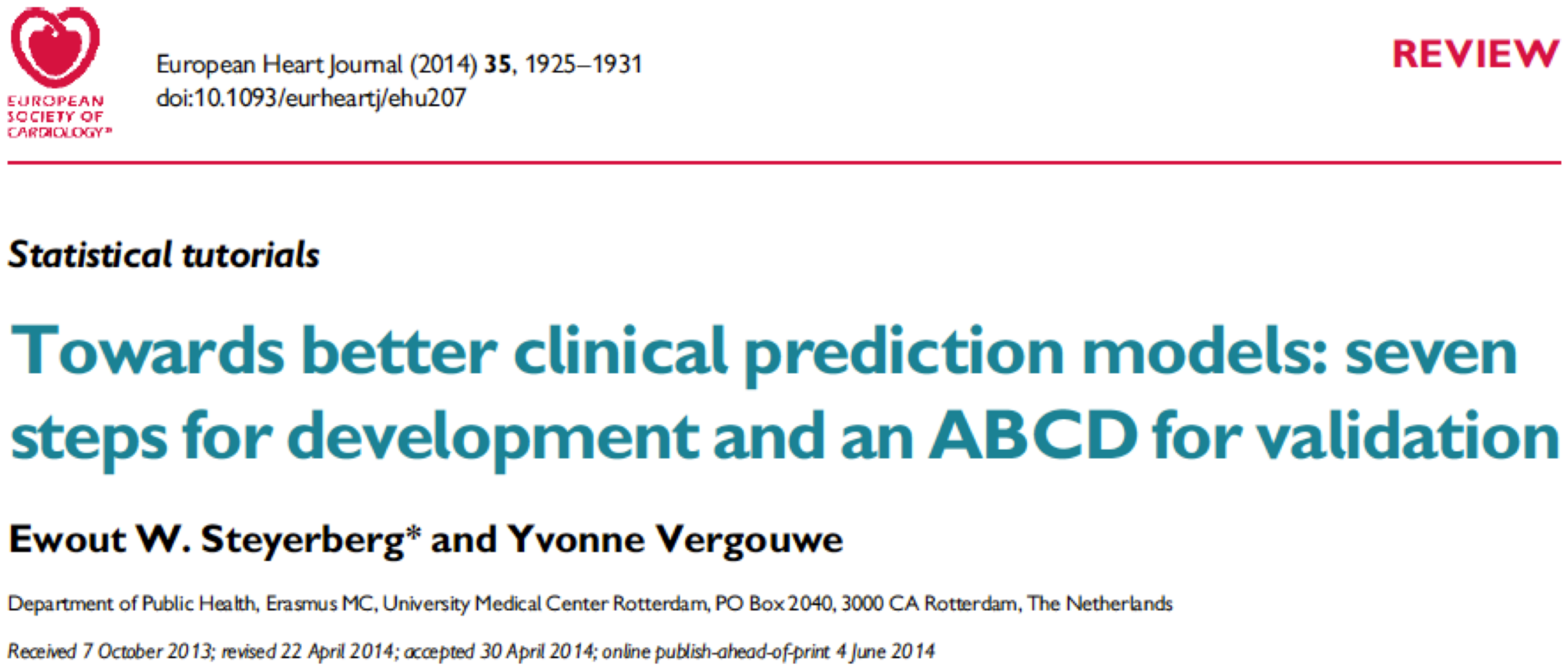
choice of threshold

choice of threshold

Credit: Rousson and Zumbrunn 2011

Earlier work: Vickers & Elkin 2006

Resource: ABCDs of validation



EUROPEAN SOCIETY OF CARDIOLOGY

European Heart Journal (2014) **35**, 1925–1931
doi:10.1093/eurheartj/ehu207

REVIEW

Statistical tutorials

Towards better clinical prediction models: seven steps for development and an ABCD for validation

Ewout W. Steyerberg* and Yvonne Vergouwe

Department of Public Health, Erasmus MC, University Medical Center Rotterdam, PO Box 2040, 3000 CA Rotterdam, The Netherlands

Received 7 October 2013; revised 22 April 2014; accepted 30 April 2014; online publish-ahead-of-print 4 June 2014

Quick Tour of Common Predictors

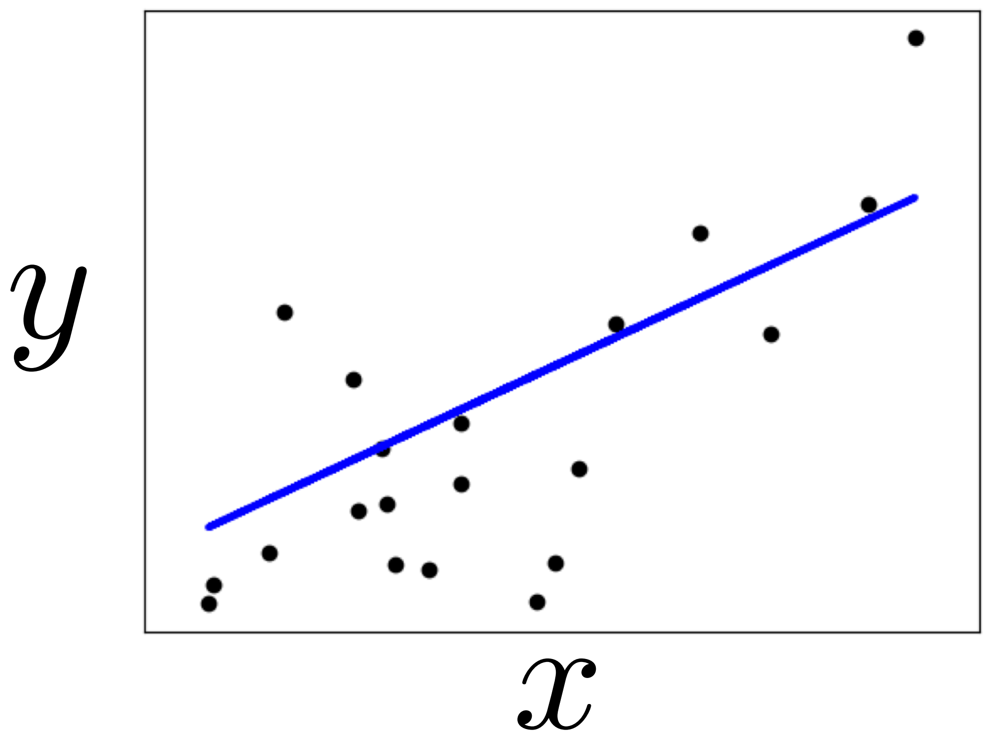
- Linear models
- Decision Trees / Random Forests
- Neural Networks
- Gaussian Processes

Keep in mind how each one:

- Trades off simplicity for flexibility
- Might be sensitive to hyperparameters
- Might be sensitive to initialization/optimization
- Might scale to very large datasets

Linear Regression

Suppose each x is one scalar (patient age). We wish to predict length of stay.



Training task

Estimate w, b to minimize squared error

Prediction task

Predict y for new x

$$\hat{y}(x_n) = wx_n + b$$

Linear Regression (multivariate)

$$\begin{array}{c} y \\ \begin{bmatrix} -4.8 \\ 2.3 \\ 4.6 \\ \dots \\ 1.2 \end{bmatrix} \\ \text{N examples} \end{array} \approx \begin{array}{c} \tilde{x} \quad \text{D features + bias} \\ \begin{bmatrix} 1.2 & 5.2 & 6.7 & 1 \\ -3.4 & 7.8 & 9.9 & 1 \\ 2.4 & -2.5 & 8.2 & 1 \\ & & \dots & \\ 5.1 & -6.2 & 2.3 & 1 \end{bmatrix} \end{array} \begin{array}{c} \tilde{w} \\ \begin{bmatrix} -1.1 \\ 7.3 \\ -2.3 \\ \mathbf{b = 2.5} \end{bmatrix} \end{array}$$

(Tilde means includes bias. Makes notation easy.)

$$\hat{y}(x_n) = w_1 x_{n1} + w_2 x_{n2} + \ldots + w_D x_{nD} + b$$

Regularization of linear models

To avoid overfitting (improve generalization), can penalize weights from taking extreme values. Especially needed when D >> N.

L2 regularization / “Ridge regression”

$$\sum_{n=1}^{N}(y_n - \tilde{w}^T\tilde{x}_n)^2 + \alpha\sum_d w_d^2$$

L1 regularization / “Lasso regression”

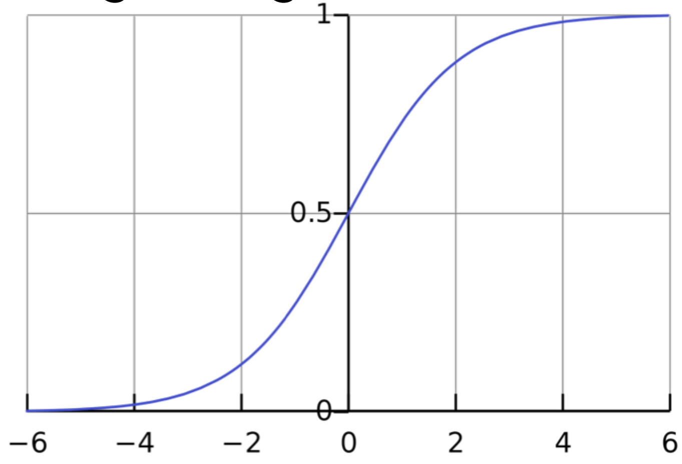
$$\sum_{n=1}^{N}(y_n - \tilde{w}^T\tilde{x}_n)^2 + \alpha\sum_d |w_d|$$

Logistic Regression

Probabilistic Binary classifier

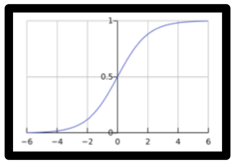
$$y \quad \begin{array}{r} 1 \\ 0 \\ 0 \\ \ldots \\ 0 \end{array} \qquad \tilde{x} \quad \begin{array}{rrrr} 1.2 & 5.2 & 6.7 & 1 \\ -3.4 & 7.8 & 9.9 & 1 \\ 2.4 & -2.5 & 8.2 & 1 \\ & & \ldots & \\ 5.1 & -6.2 & 2.3 & 1 \end{array}$$

Logistic sigmoid function



Map every real number onto unit interval
Interpret output as a probability

Prediction rule

$\hat{y}(x_n) =$  $\Big(w_1 x_{n1} + w_2 x_{n2} + \ldots + w_D x_{nD} + b\Big)$

Training objective

$$\min_{\tilde{w}=\{w,b\}} \sum_{n=1}^{N} \text{logistic\_loss}(y_n, \hat{y}(x_n, w, b))$$

Logistic Regression

PRO

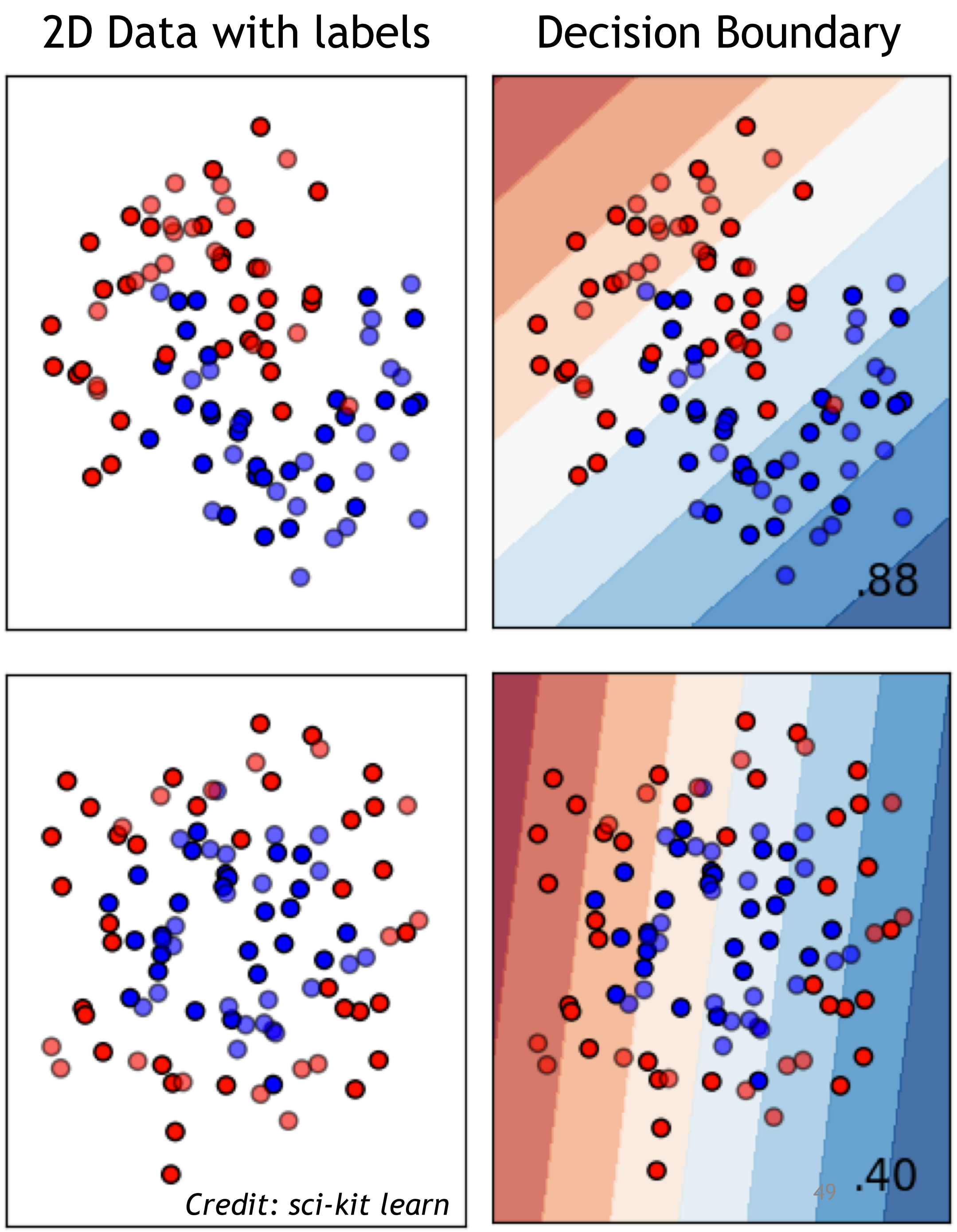
- Scales to many examples and many features
- Easy to inspect learned weights
- Easy optimization (convex)

CON

- Only linear boundaries
- Some minor data cleaning required (standardize numerical scale, categoricals to one-hot)

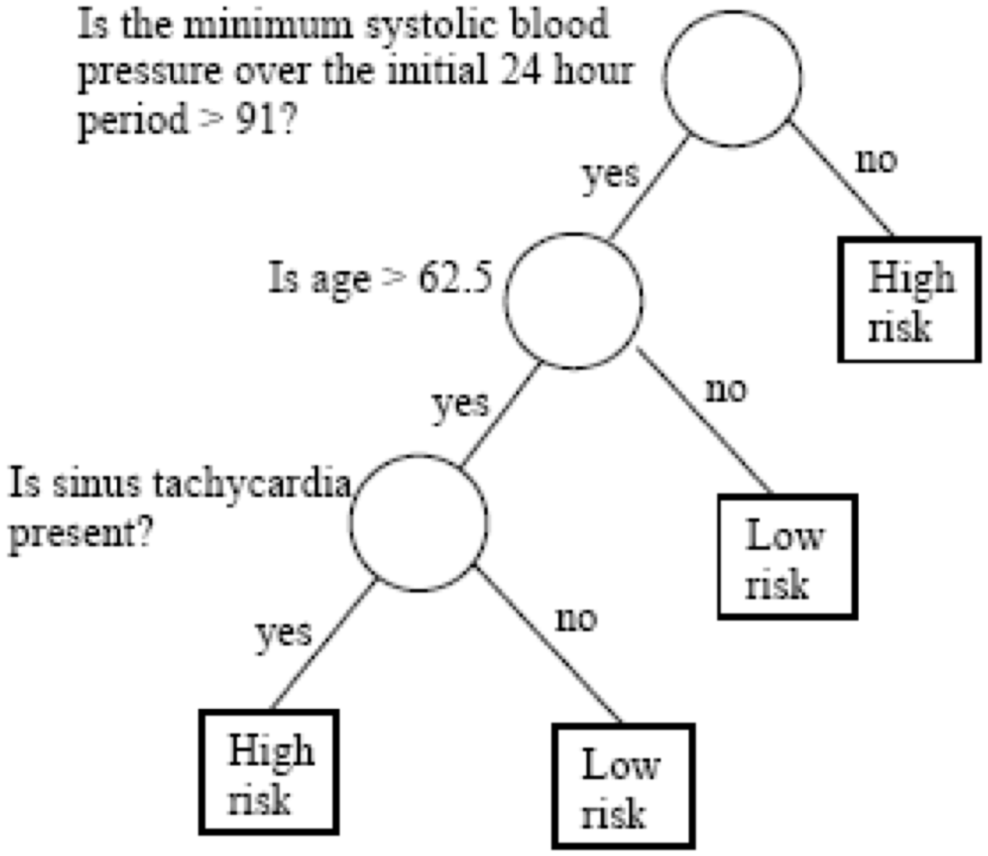
TUNE

- Reg. type? L1/L2
- Reg. strength?



Credit: sci-kit learn

Decision Trees


Is the minimum systolic blood pressure over the initial 24 hour period > 91?
yes
no
High risk
Is age > 62.5
yes
no
Low risk
Is sinus tachycardia present?
yes
no
High risk
Low risk

Resource: Ch. 9.2 of Elements of Statistical Learning

Training Decision Trees

Usually iterative procedure. At each step, greedily select best (variable, split-value) pair.

Regularization: Prevent overfitting by setting

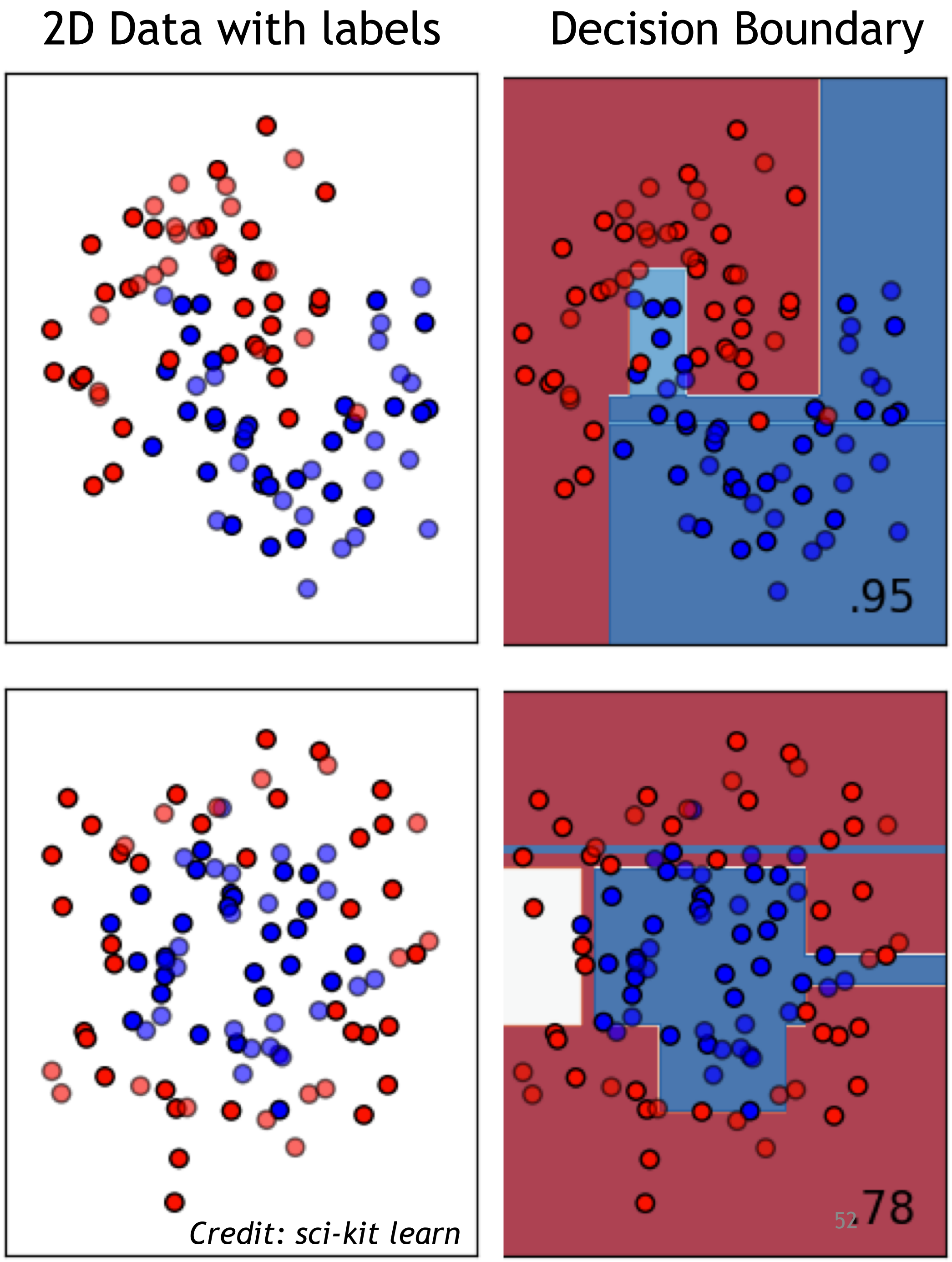
- max tree depth
- min. number of examples per leaf node

Decision Trees

PRO
- More flexible boundaries than linear models
- Easy to use features of many types (binary, real)
- Little data cleaning required

CON
- Unordered category features require huge trees
- Decision boundaries can only be "axis aligned"
- Exact tree structure very sensitive to training data

TUNE
- Max depth
- Min node size
- Splitting criteria

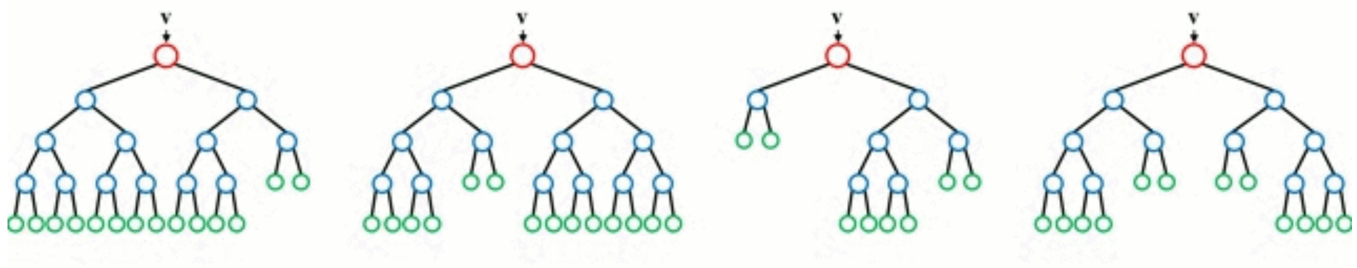


Random Forest

Train many trees, each one by:

- Sample N examples with replacement from training set
- Choose from a random subset of features at each greedy split

Final prediction: average of all trees



Credit: Hgkim0331/Wikimedia Commons

Random Forest

PRO

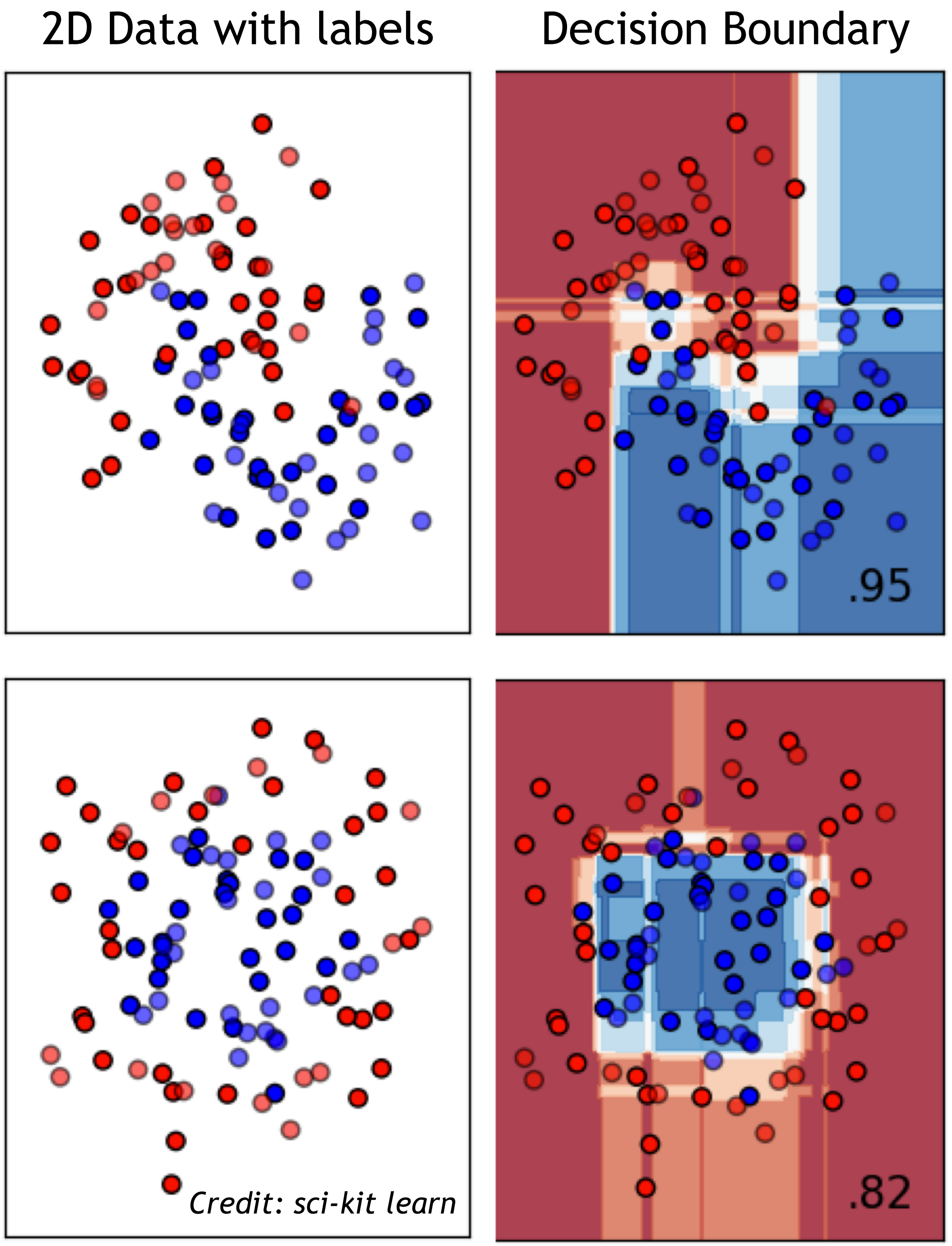
- Flexible boundaries
- Easy to use features of many types (binary, real)
- Little data cleaning required
- Less variance in predictions than single decision tree. Usually leads to better generalization.

CON

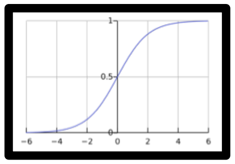
- Harder to interpret than single decision tree

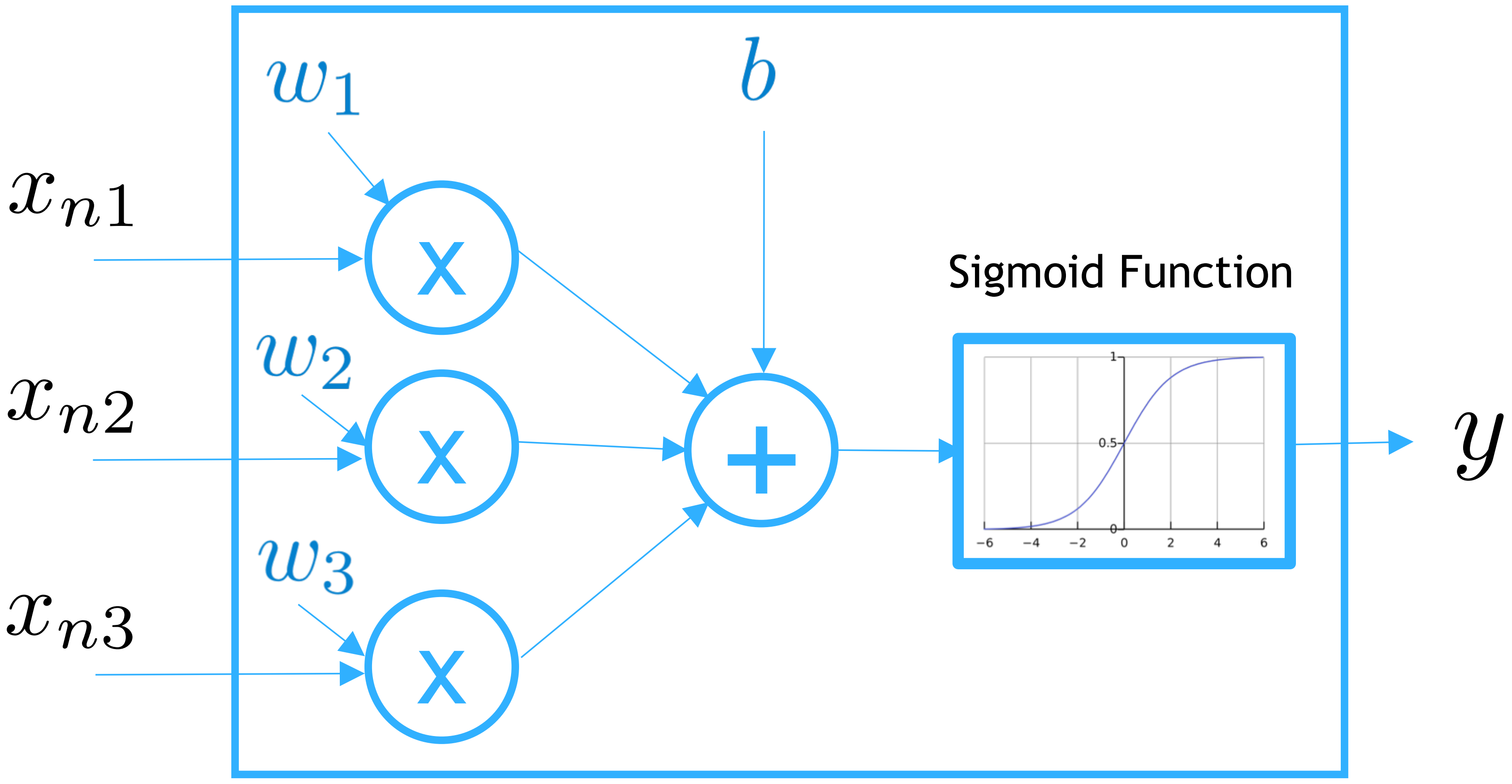
TUNE

- How many trees?
- All decision tree hypers



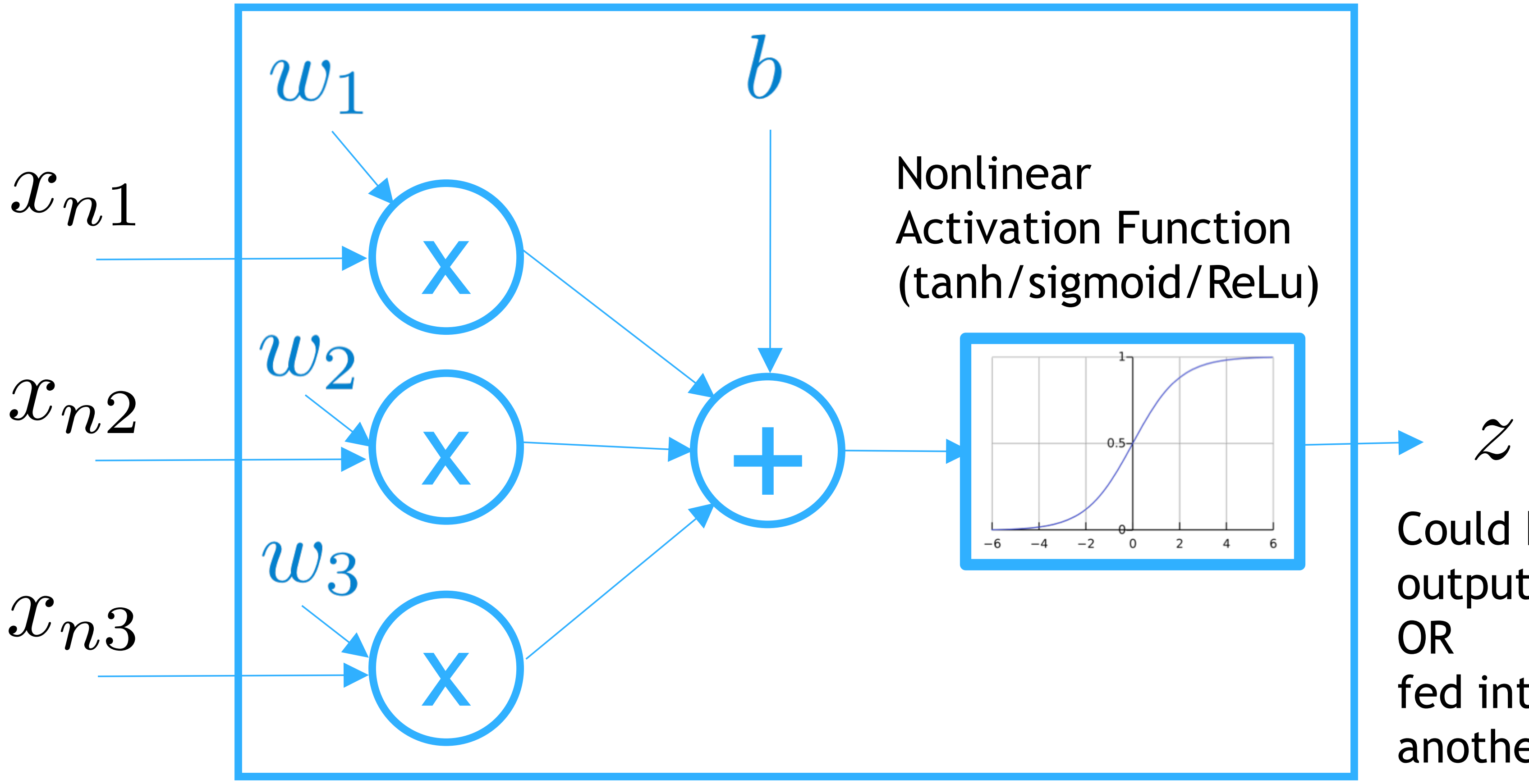
Logistic Regression as a “neuron”

$\hat{y}(x_n) =$  $\left(w_1 x_{n1} + w_2 x_{n2} + \ldots + w_D x_{nD} + b\right)$



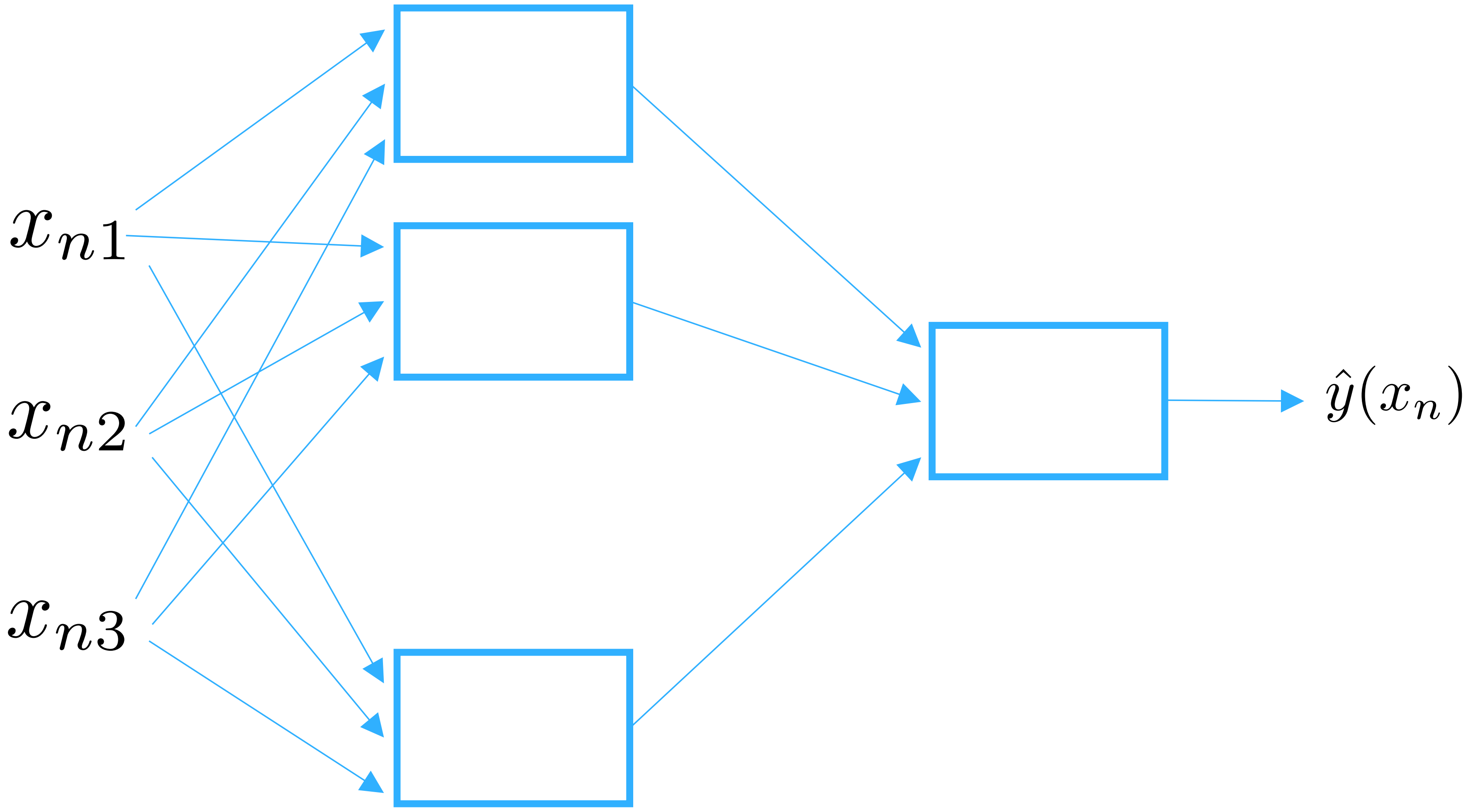
Simple Many-to-one Neuron

The basic unit of neural networks for regression/classification

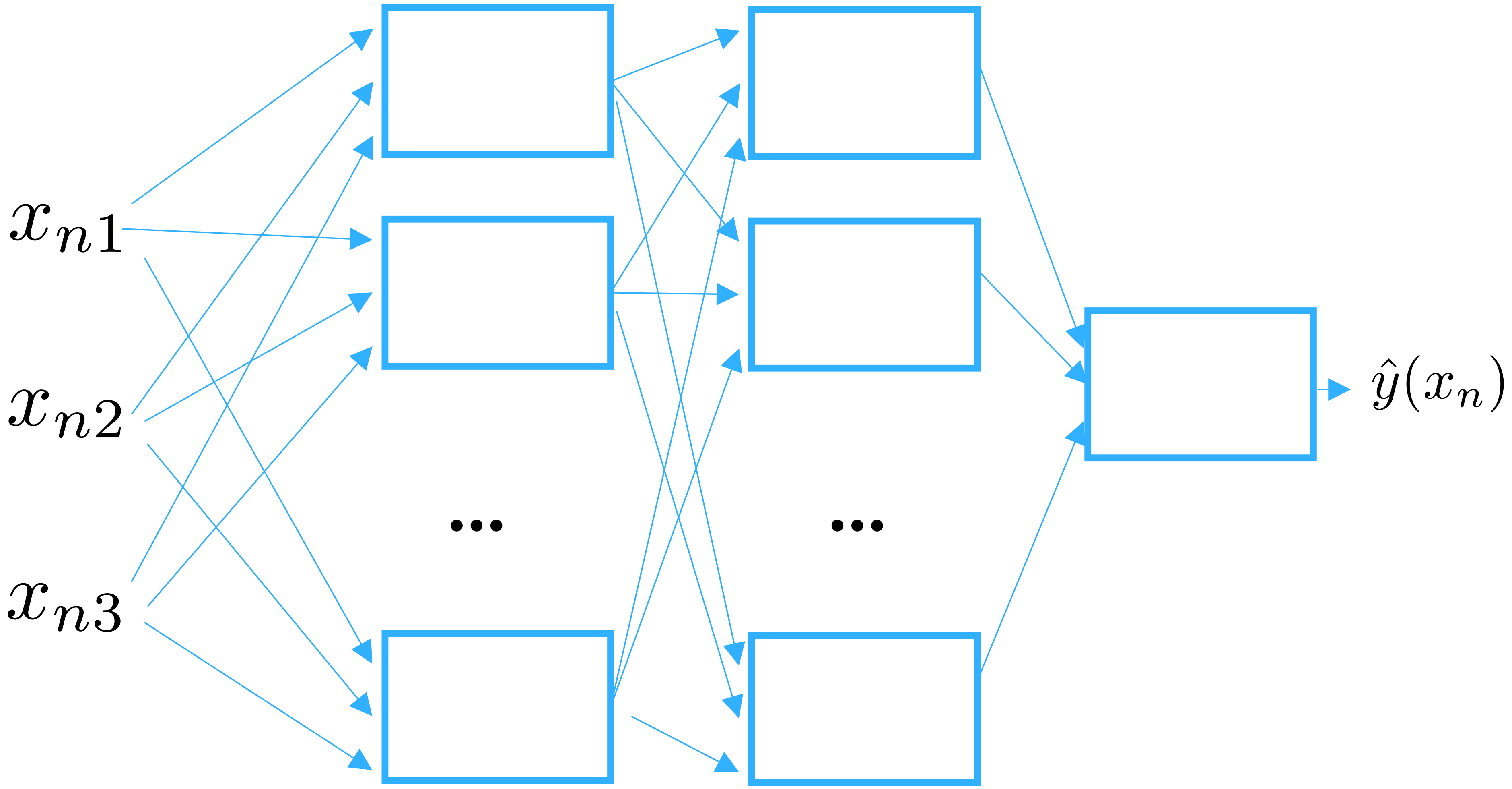


Could be final output,
OR
fed into to another layer!

NN with 1 hidden layer



Multilayer Perceptron

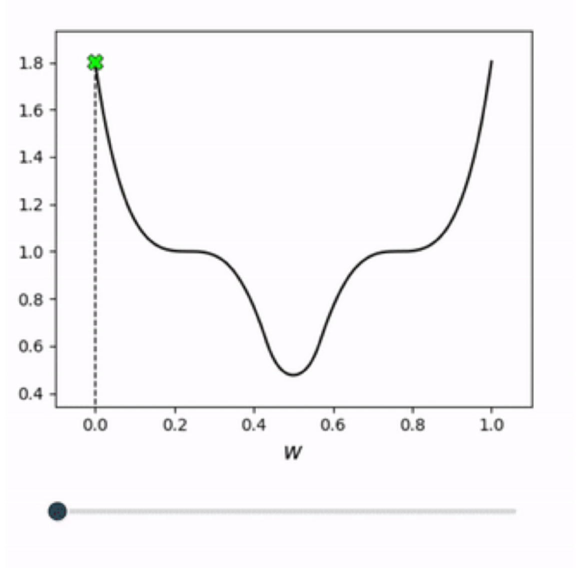


Training MLPs

$$\min_{w} \quad \sum_{n=1}^{N} \mathrm{loss}(y_n, \hat{y}(x_n, w)) + \mathrm{regularizer}(w)$$

“Backpropagation”
Gradient descent using chain rule

Credit: Jeremy Watt



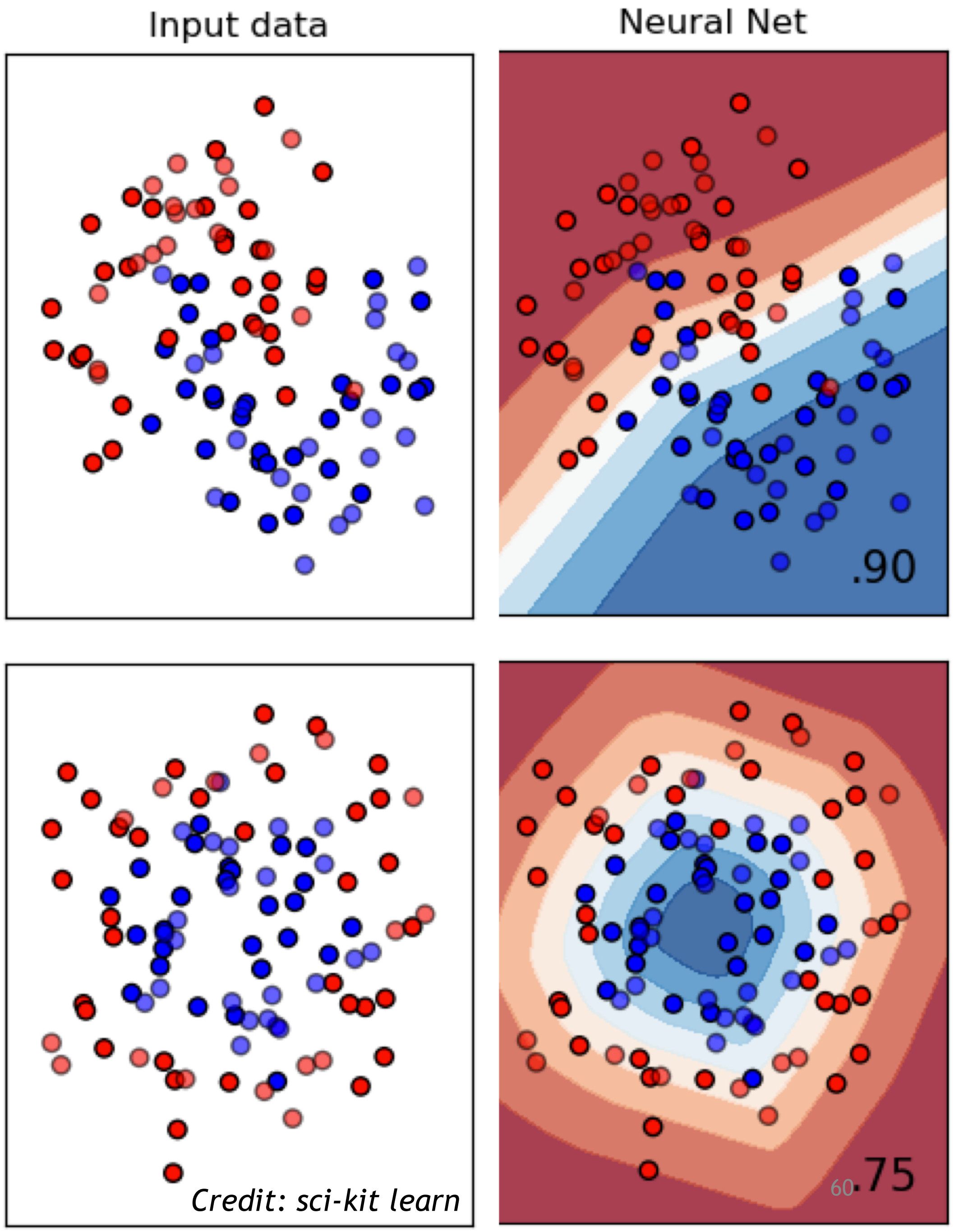
- Scalable: process data one mini-batch at a time
- Modern software: automatic differentiation
  - You provide the loss function + prediction architecture
  - Software does all gradient computations

MLP
Neural Net

PRO
- Flexible boundaries
- Built from composable pieces
- Great software available
- Fast to test, scales well

CON
- Easy to overfit
- Sensitive to init.: Can get different networks from same train data

TUNE
- Activation function?
- How many layers?
- How many units per layer?
- Regularization (L1/L2)?
- Batch size / learning rate



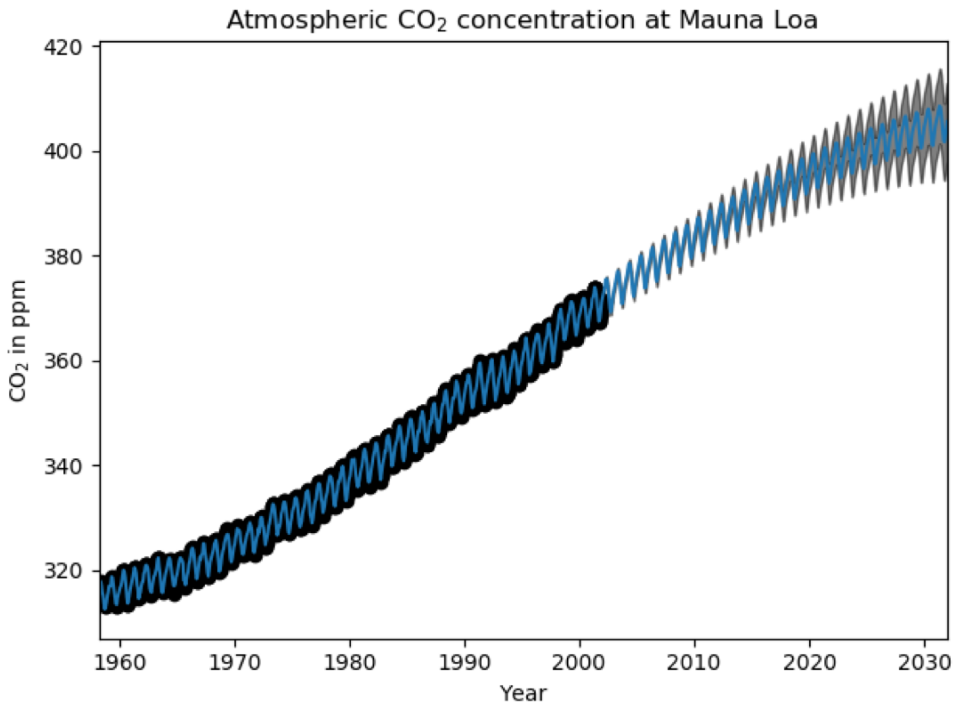
Tradeoffs

	Decision Boundaries	Parameters to tune	Interpretation
Logistic Regr.	Linear	Reg. type: L2 or L1 Reg. strength?	Inspect weights
Decision Tree	Axis-aligned	Max. depth Min. node size Split criteria	Inspect tree
Random Forest	Flexible	Num. trees Max. depth Min. node size Split criteria	Feature importance weights
MLP	Flexible	**Num. layers?** **Num. units per layer?** **Reg. type and strength?** Activation function?	**See Part 3**

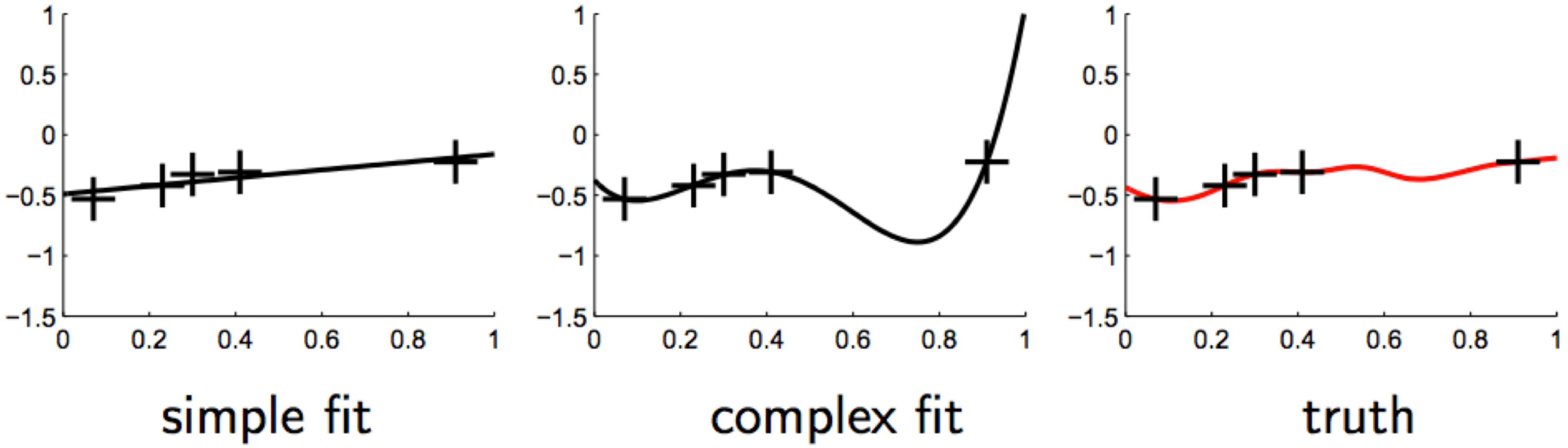
What if …

- We care about uncertainty in predictions?
- We care about models that extrapolate?
  - Need to bake in domain-knowledge

Need: tell model to capture seasonal periodicity



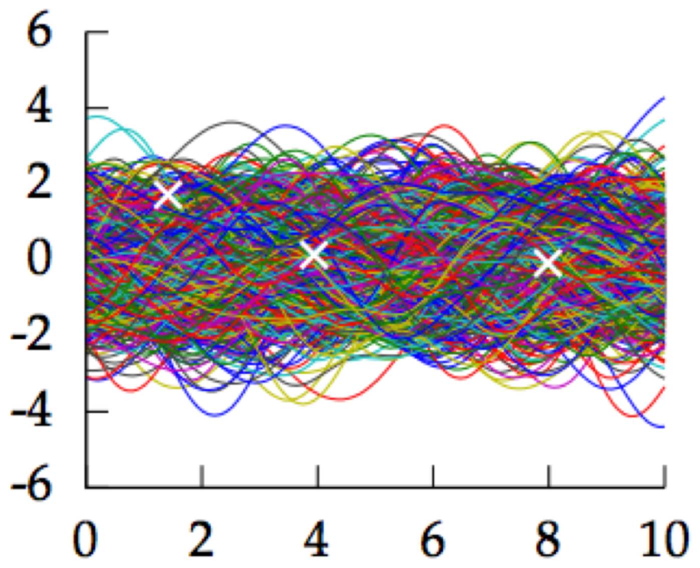
Gaussian Process Regression



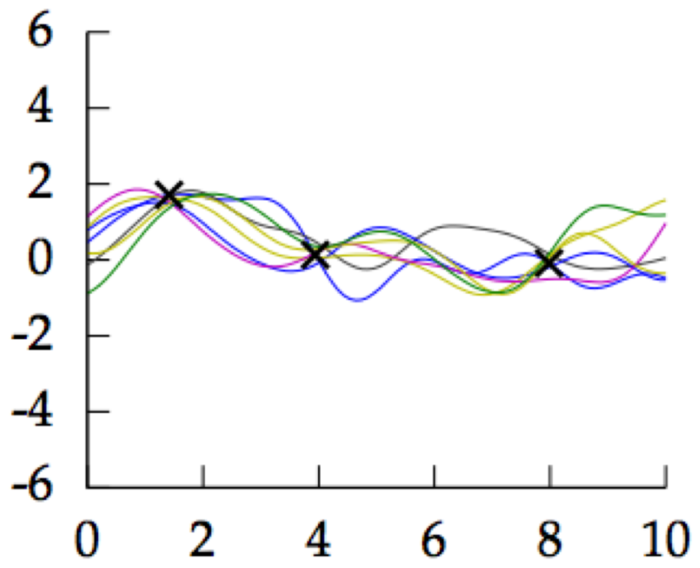
Gaussian process (GP) predictors can provide principled tools for:

- Specifying useful prior knowledge
- Estimating uncertainty
- Avoiding overfitting (Bayesian Occam's razor)
- Avoiding underfitting (model complexity can grow with more data)

Gaussian Process Regression

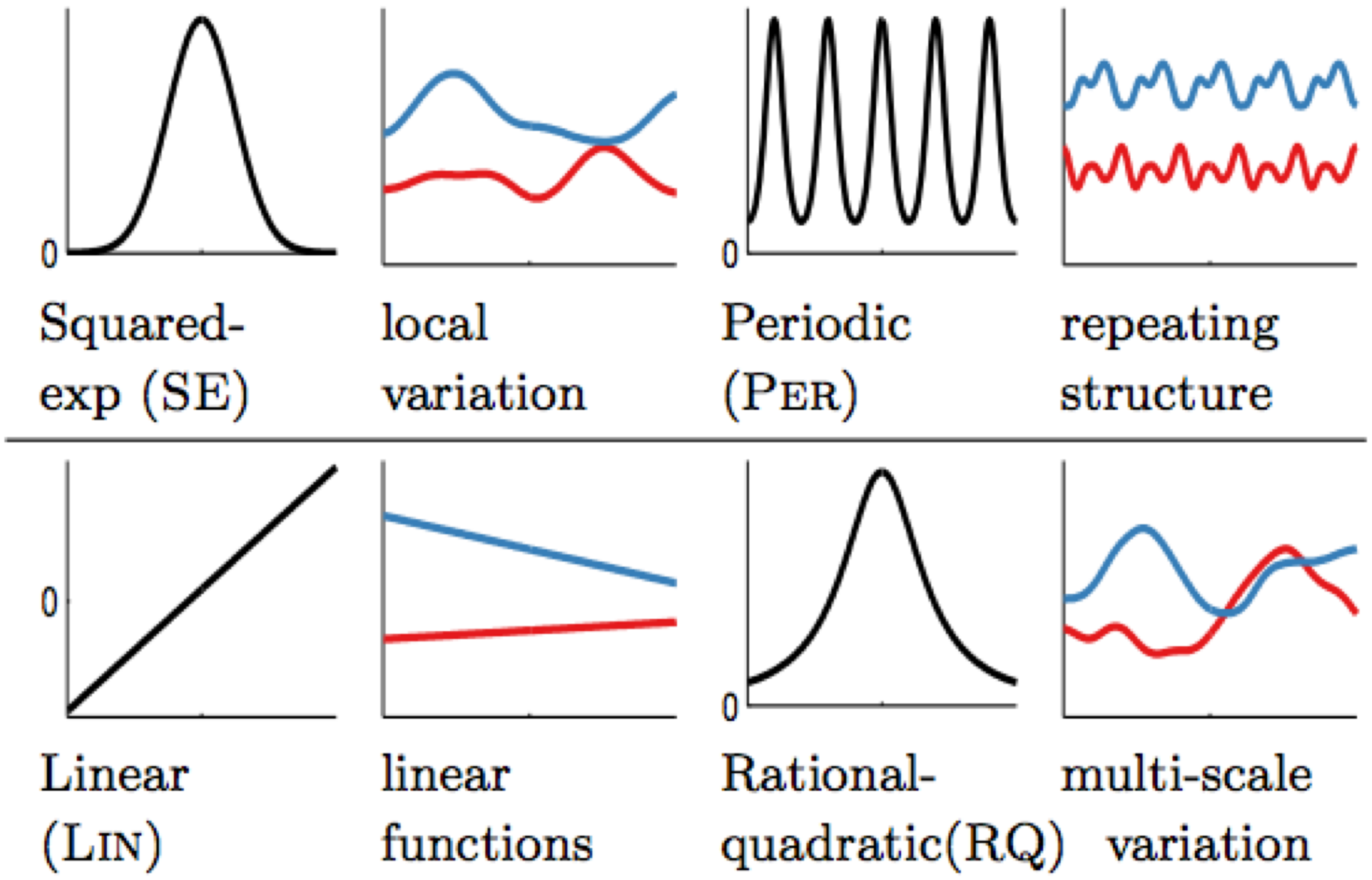


prior



posterior

Modeling dependencies with covariance functions (“kernels”)

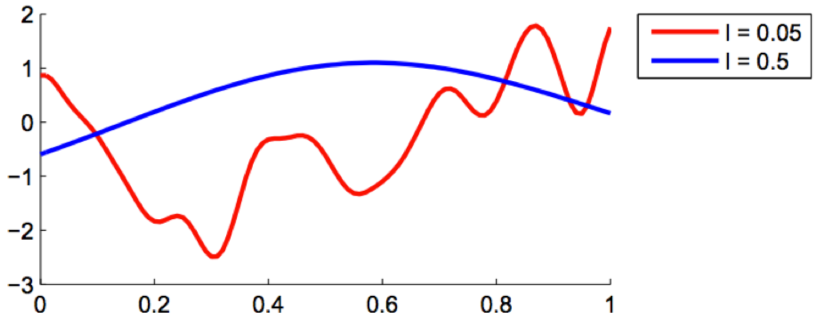


Credit: Duvenaud et al. ICML 2013

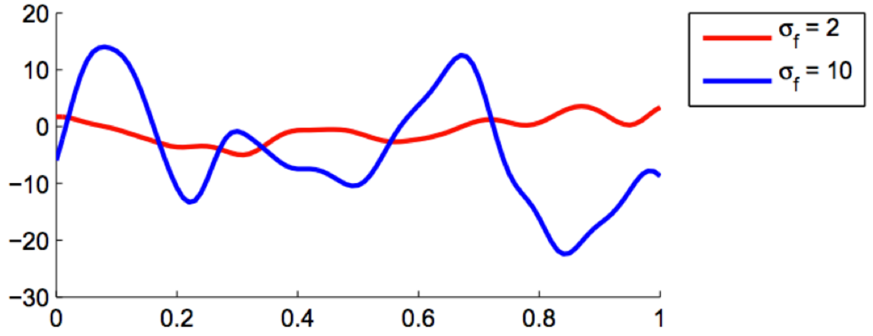
Modeling dependencies with squared exponential kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{d=1}^{D} (x_{d,i} - x_{d,j})^2 / \ell_d^2 \right)$$

Lengthscale controls distance between peaks



Variance controls output magnitude



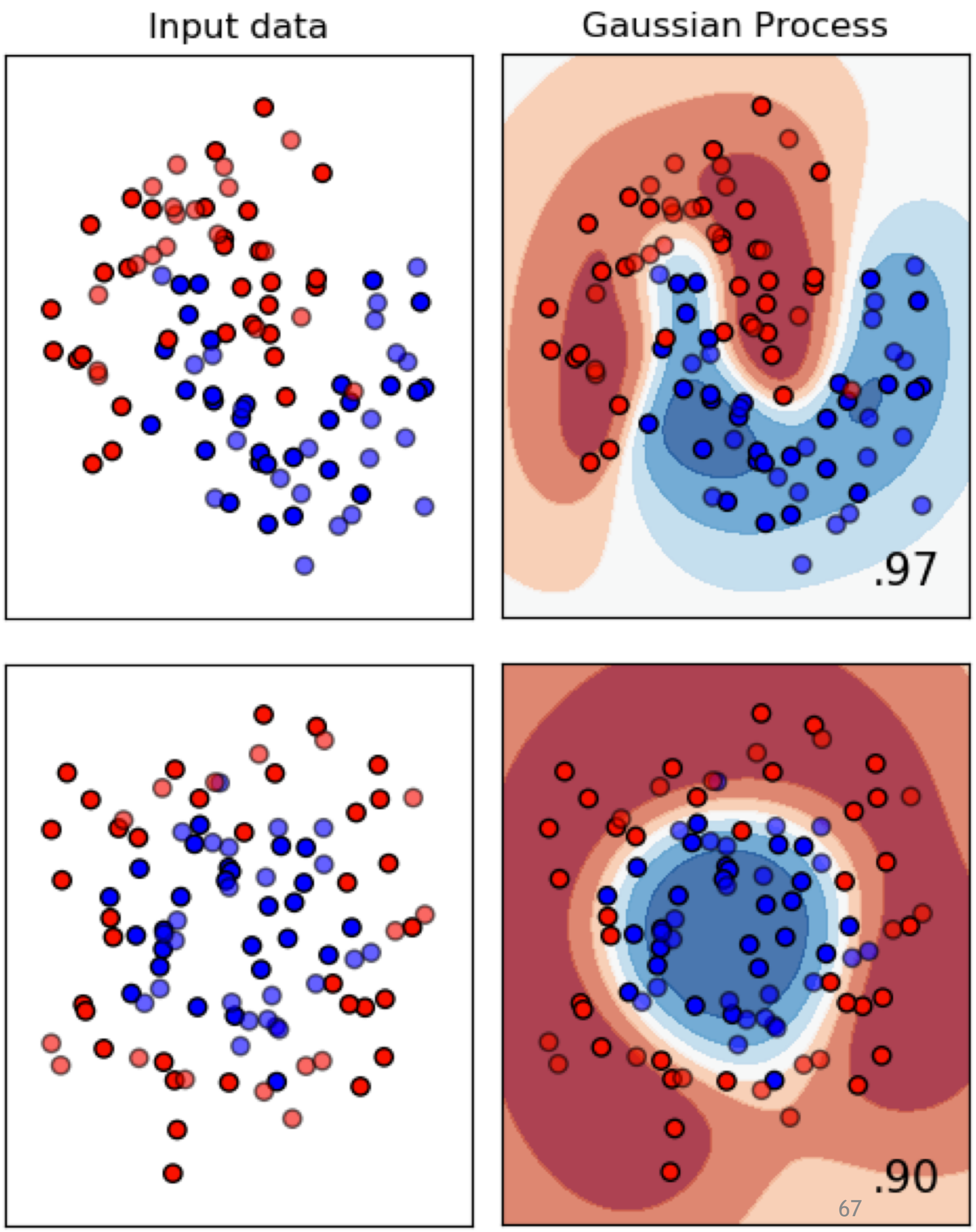
Credit: Iain Murray' slides

Gaussian Processes

PRO
- Flexible boundaries
- Hard to overfit
- Manage uncertainty

CON
- **Scale poorly** (cubic in number of examples)

TUNE
- Kernel function?
- Kernel hyperparameters?

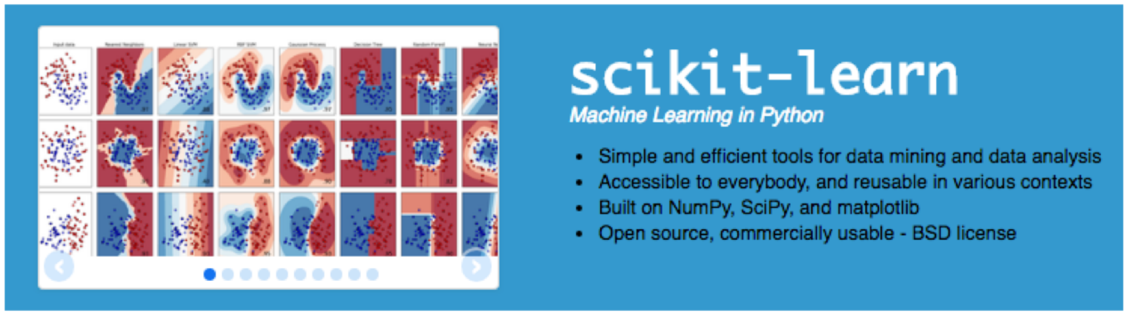


Takeaway: Prediction Challenges

- Choose the right evaluation metric?
- Choose the right input features?
- Choose the right method and training metric?
  - Is this aligned with your evaluation?
- Choose the right hyperparameters?
  - Which ones?
  - What selection strategy?

Takeaway: Prediction Software

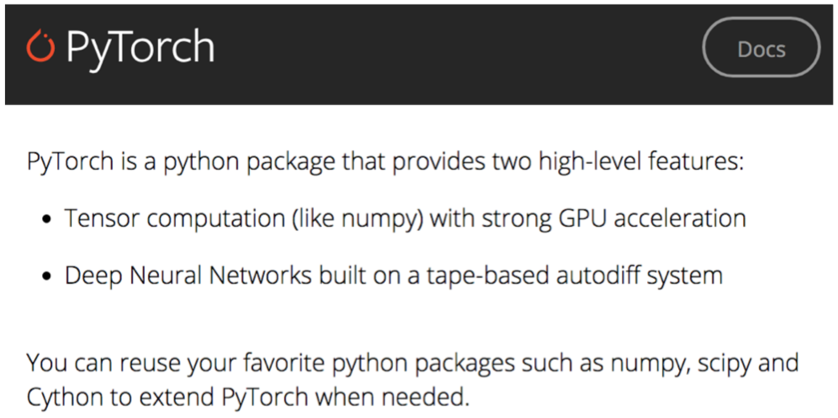
Python
- Scikit learn



- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Linear models, Decision trees, RFs, MLPs

- Tensorflow or PyTorch



PyTorch is a python package that provides two high-level features:

- Tensor computation (like numpy) with strong GPU acceleration
- Deep Neural Networks built on a tape-based autodiff system

You can reuse your favorite python packages such as numpy, scipy and Cython to extend PyTorch when needed.

For fancier NNs