Text Classification

Classification & Evaluation

Ayoub Bagheri

Outline

- Text classification
- Algorithms
- Evaluation

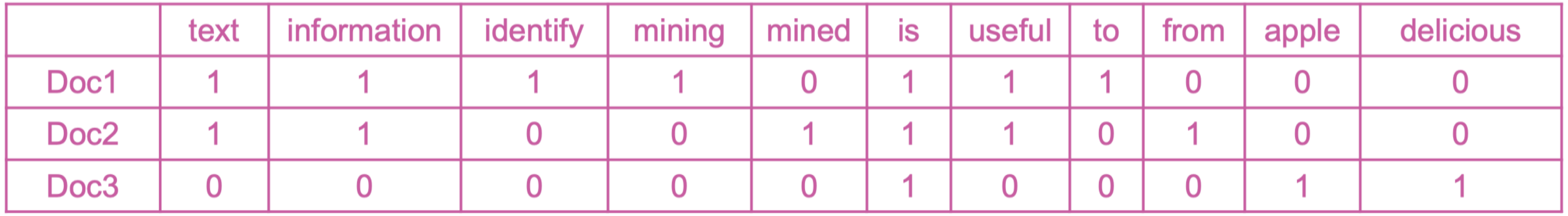
Bag-of-Words representation

Doc1: Text mining is to identify useful information.

Doc2: Useful information is mined from text.

Doc3: Apple is delicious.

Document-Term matrix (DTM):



	text	information	identify	mining	mined	is	useful	to	from	apple	delicious
Doc1	1	1	1	1	0	1	1	1	0	0	0
Doc2	1	1	0	0	1	1	1	0	1	0	0
Doc3	0	0	0	0	0	1	0	0	0	1	1

Text classification

- **Supervised learning**: Learning a function that maps an input to an output based on example input-output pairs.
  - infer a function from labeled training data
  - use the inferred function to label new instances

- Human experts annotate a set of text data
  - Training set

Document	Class
Email1…..	Not spam
Email2….	Not spam
Email3….	Spam
…	…

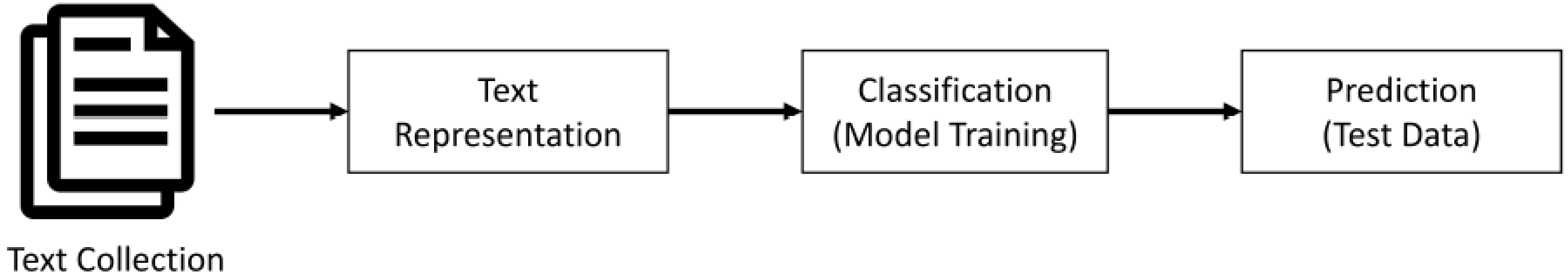
Applications

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis

Quiz?

- Which one **is not** a text classification task? (less likely to be)
  - Author's gender detection from text
  - Finding about the smoking conditions of patients from clinical letters
  - Grouping similar news articles
  - Classifying reviews into positive and negative sentiment

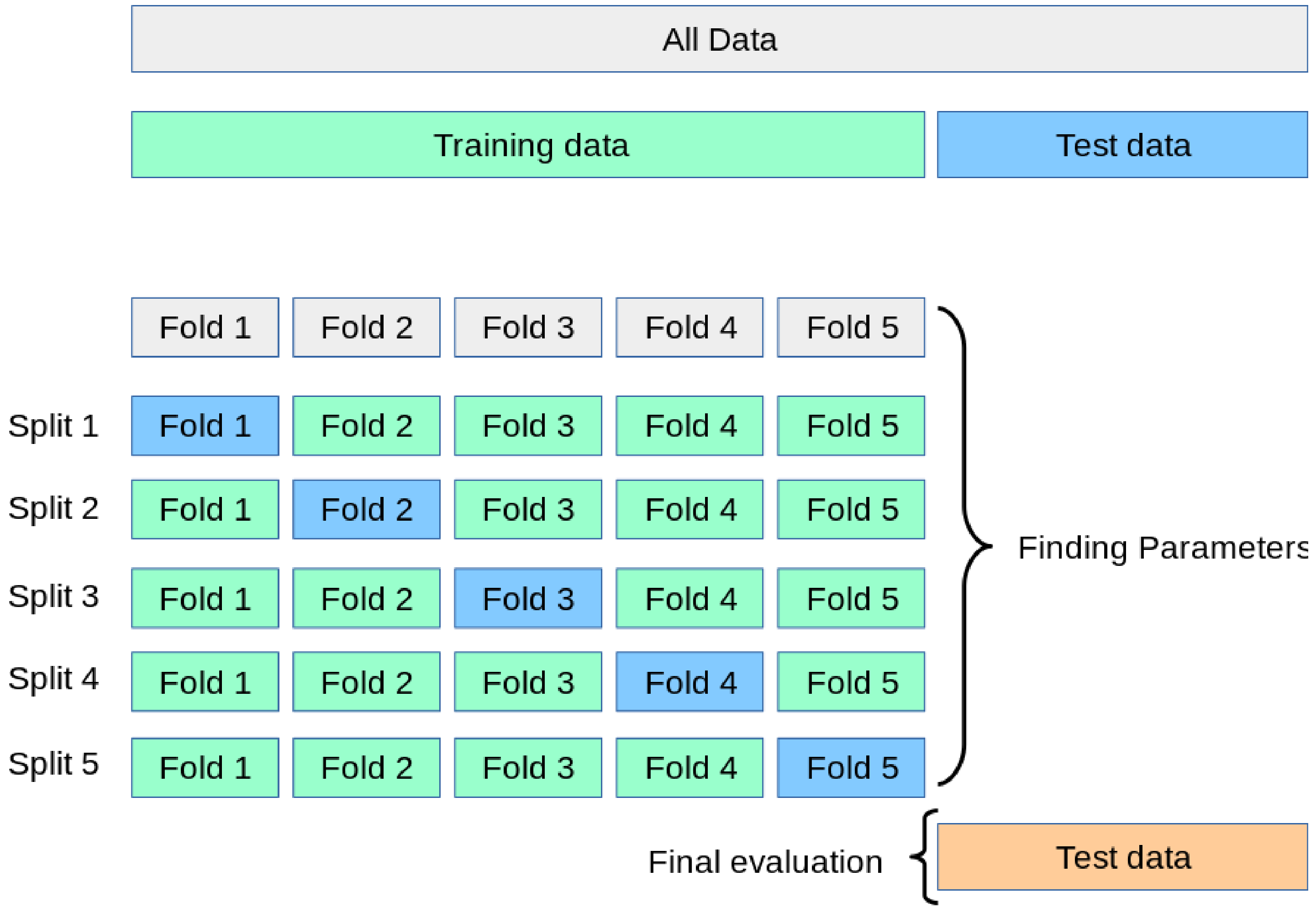
Simple pipeline



Preparing data

- Text pre-processing
- Data splitting
    - Training
    - Validation (development)
        - to tune the hyperparameters
    - Test
- Text representation

K-fold cross validation



All Data

Training data

Test data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

Finding Parameters

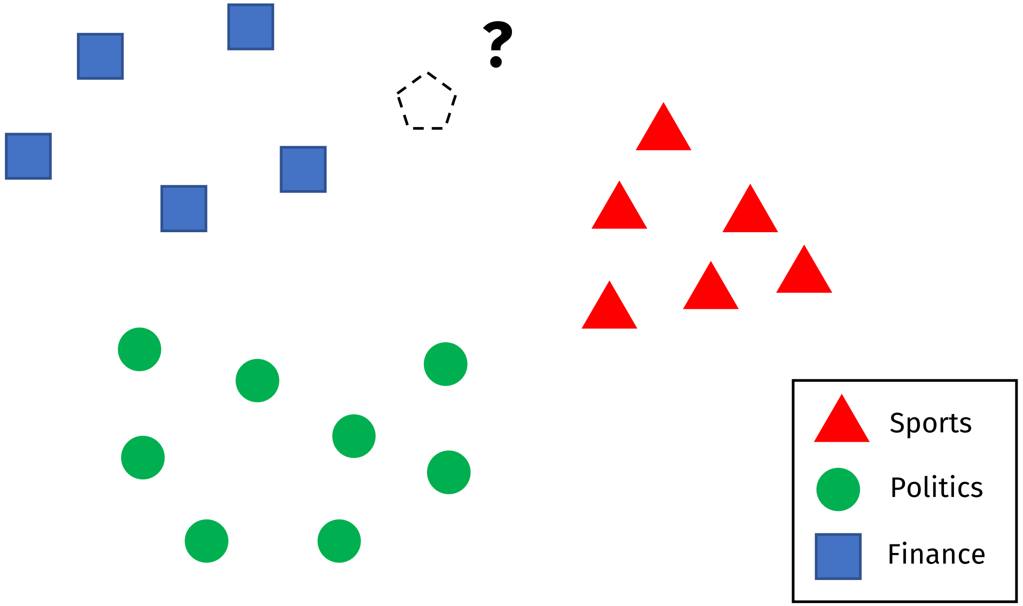
Final evaluation

Test data

https://scikit-learn.org/stable/modules/cross_validation.html

How to classify this document?

Documents by vector space representation



Text Classification: definition

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2,\ldots, c_J\}$

- *Output*: a predicted class $c \in C$

Classification Algorithms

Hand-coded rules

- Rules based on combinations of words or other features
- Accuracy can be high: If rules carefully refined by expert
- But building and maintaining these rules is expensive
- Data/Domain specifics

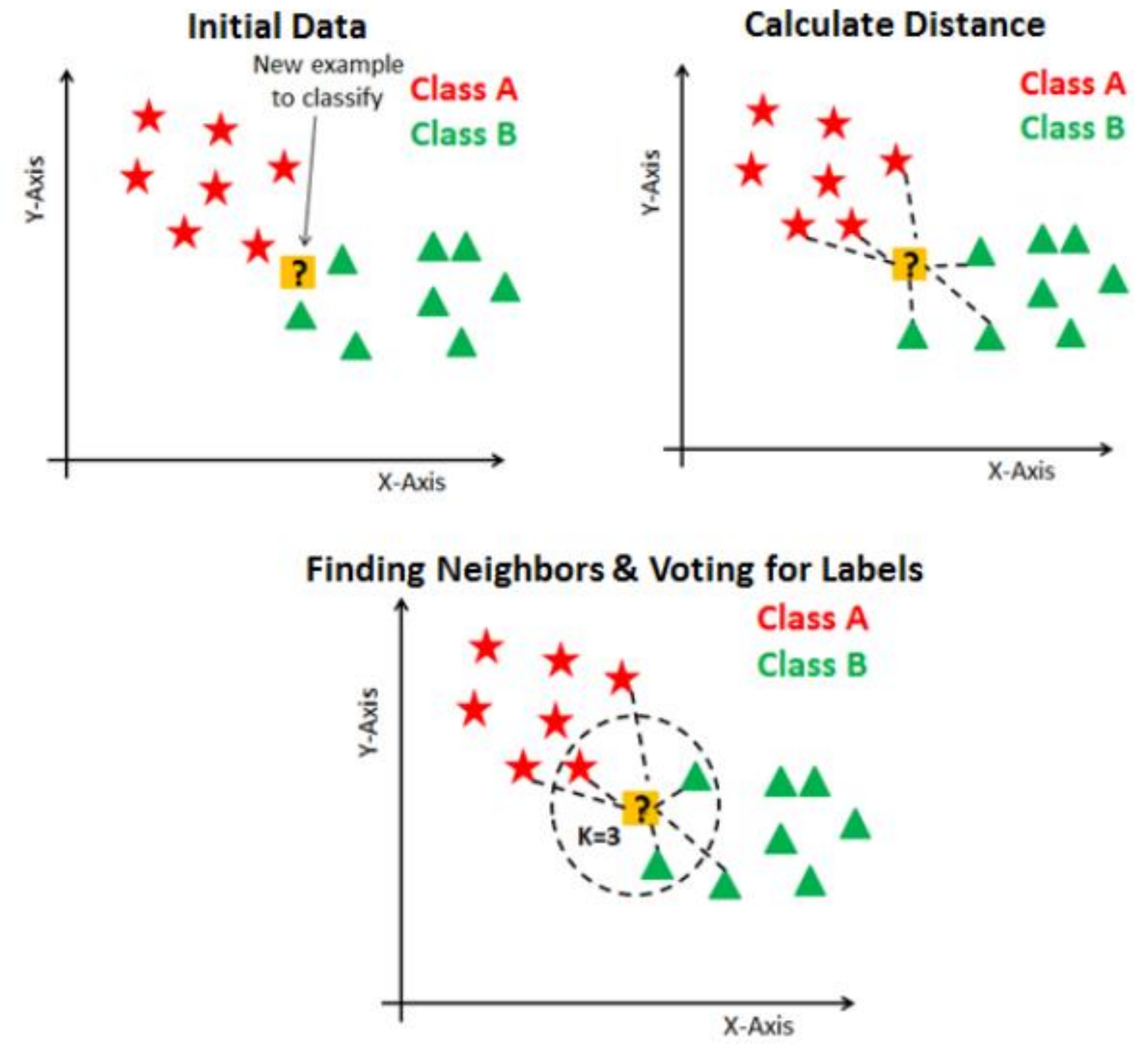
Supervised Machine Learning

- *Input:*
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$
- *Output:*
  - a learned classifier $y{:}d \rightarrow c$

Outline

- Naïve bayes
- Logistic regression
- Support-vector machines
- K-nearest neighbors
- Neural networks
- Deep learning

K-nearest neighbor



Naïve Bayes

y(

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

)=c

Bayes' rule

- For a document *d* and a class *c*

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Learning naïve Bayes

$$c_{MAP} = \underset{c\hat{I}\ C}{\operatorname{argmax}} P(c \mid d)$$

MAP is “maximum a posteriori” = most likely class

$$= \underset{c\hat{I}\ C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c\hat{I}\ C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Dropping the denominator

Learning naïve Bayes

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Document d represented as features x1..xn

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

Learning naïve Bayes

- Simply use the frequencies in the data

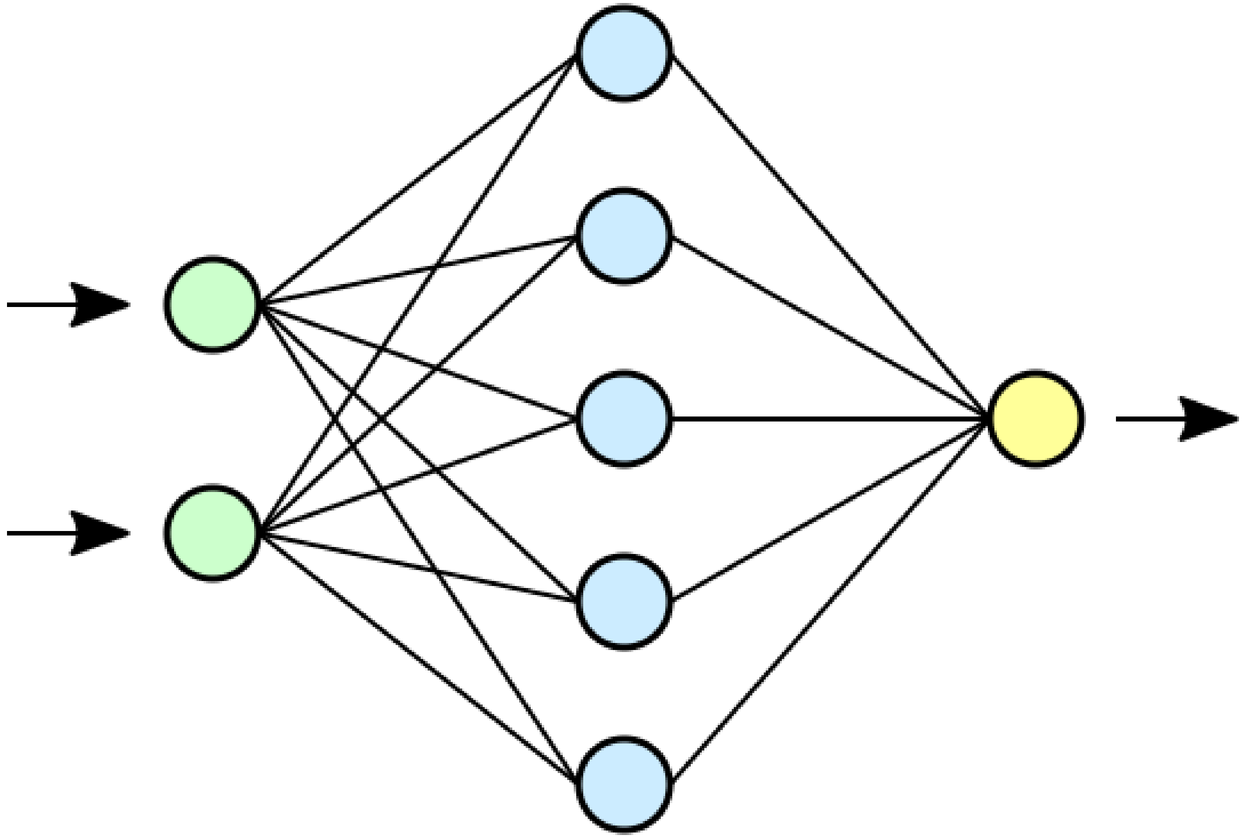
$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

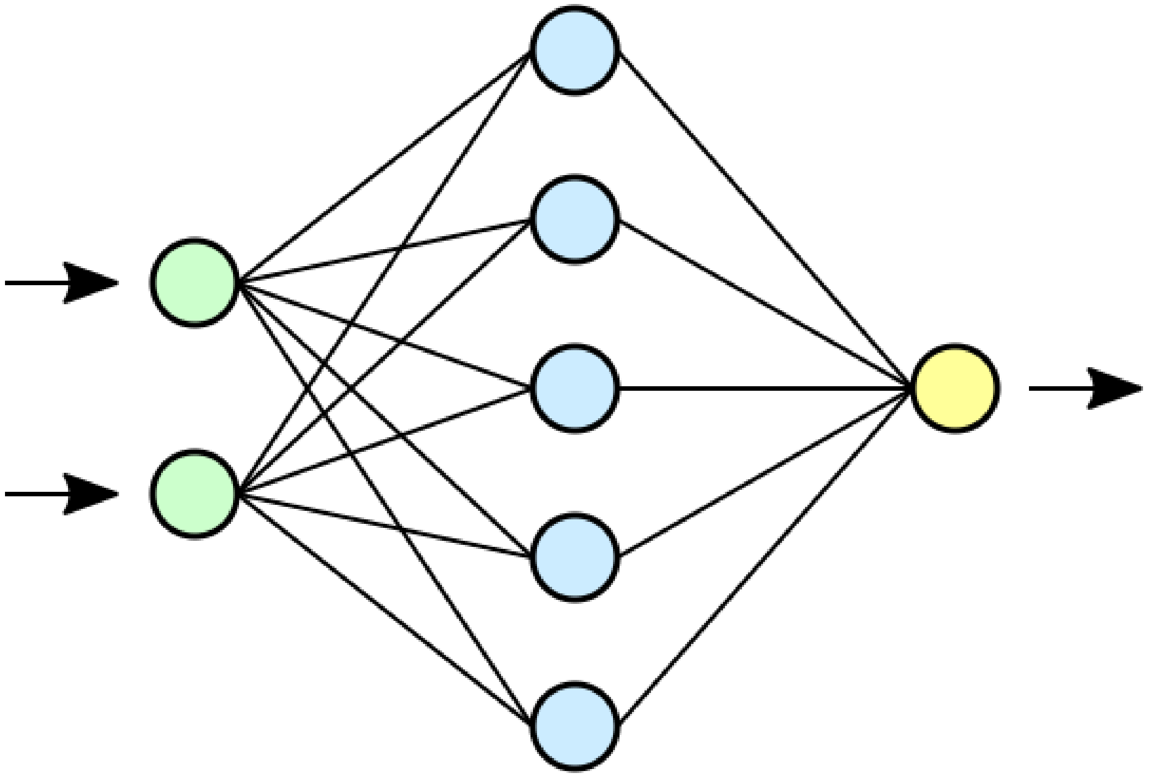
Neural networks, “deep learning”

- Compositional approach to curve-fitting;
- “Biologically inspired”
  (but don’t take that too seriously);
- Sound cool.

Neural network



Neural network



"Hidden" nodes:

Example:

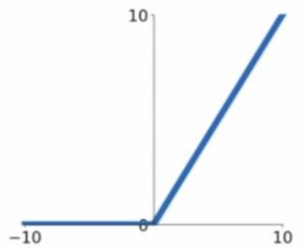
$$h_1 = f(w_{11}x_1 + w_{12}x_2)$$

Output:

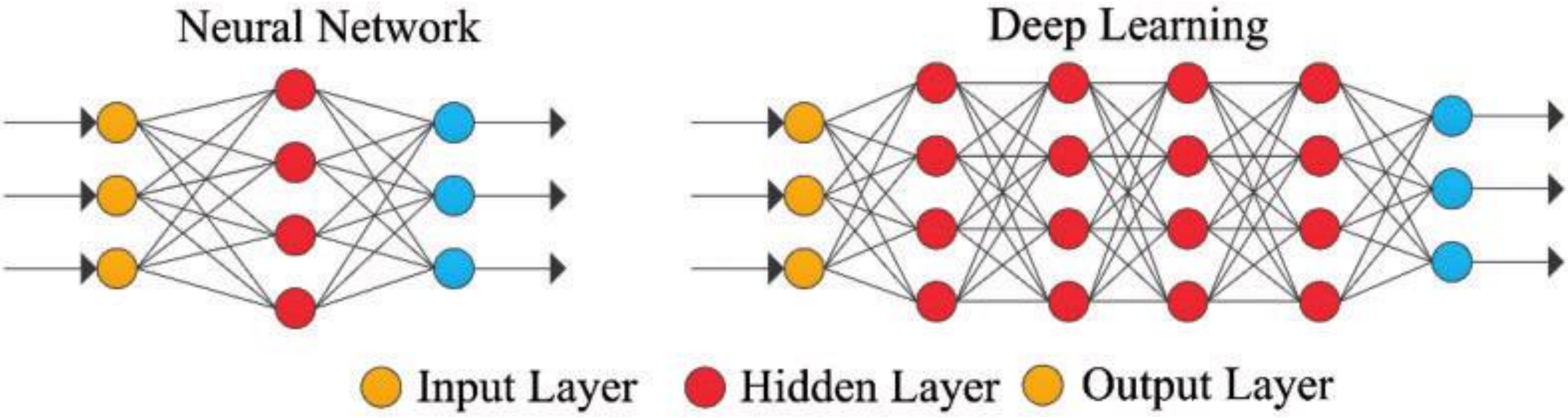
$$y = f(w_{21}h_1 + w_{22}h_2 + \cdots + w_{25}h_5)$$

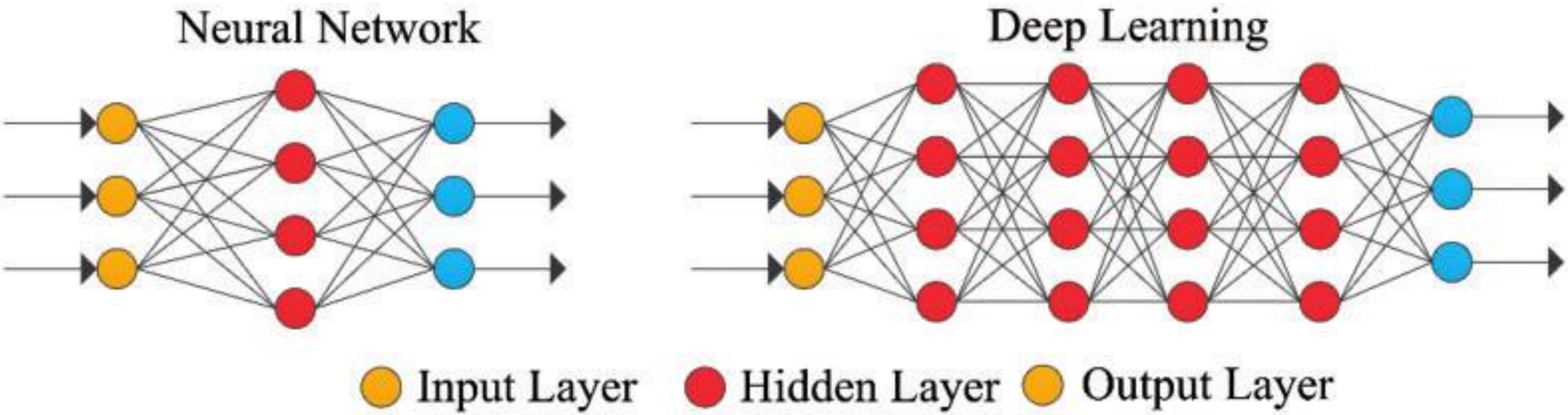
"Activation function":

- RElu $f(z) =$
- …



What makes a neural net "deep"?





Keep doing

$$z = g^{(n_h)}(g^{(\dots)}(g^{(2)}(g^{(1)}(\mathbf{x}))))$$

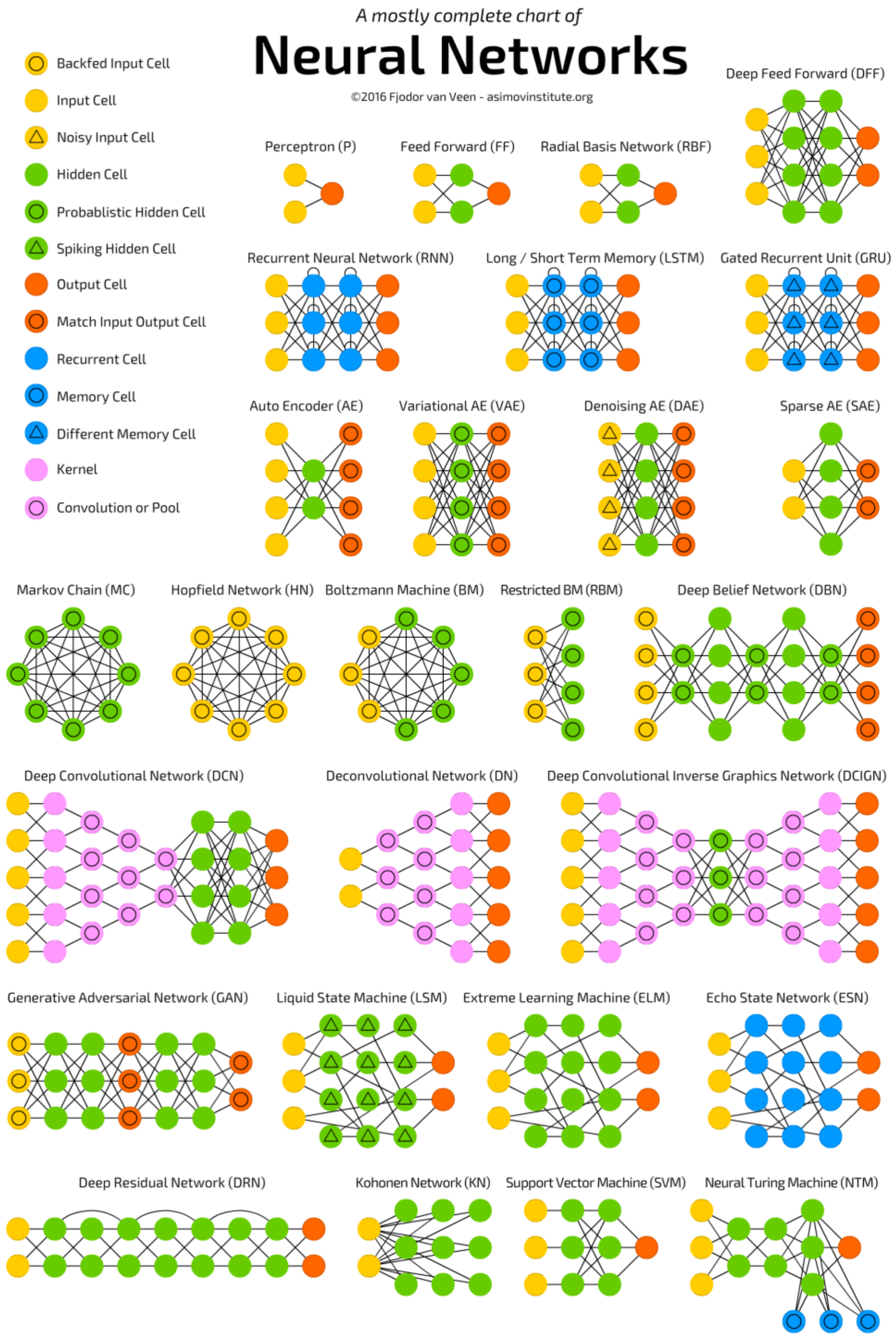
then $y \approx f(z)$.

Deep learning

- Output of each hidden layer is input to subsequent one
- Allow representation learning by building complex features out of simpler ones
- Go deep: exponential advantages, less overfitting
- Aggressive parameterization + aggressive regularization
- Compositional: efficient parametrization
- Learn relevant features: “End-to-end”

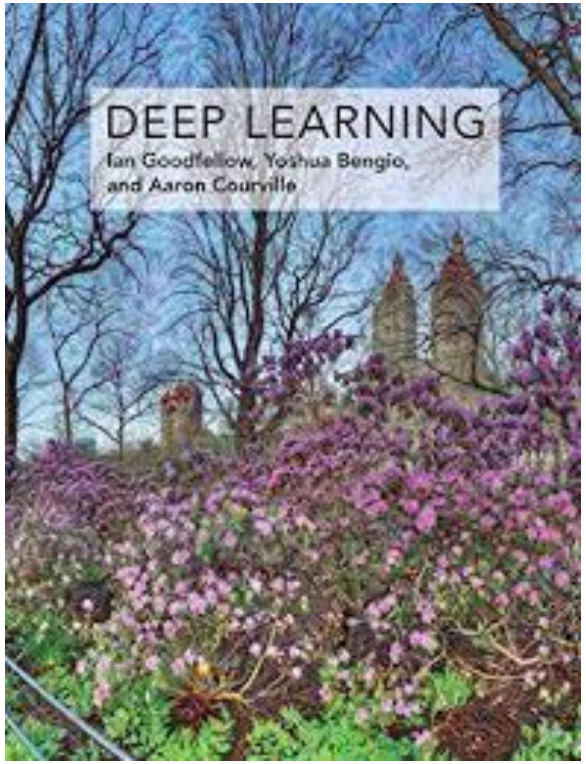
Different architectures

- By adjusting the arrows, layers, and activation functions, you can create models that are tailored to specific data, e.g.
- Convolutional (CNN): images, text, sound
- Recurrent (RNN): time series, text
- Graph (GNN): networks
- …

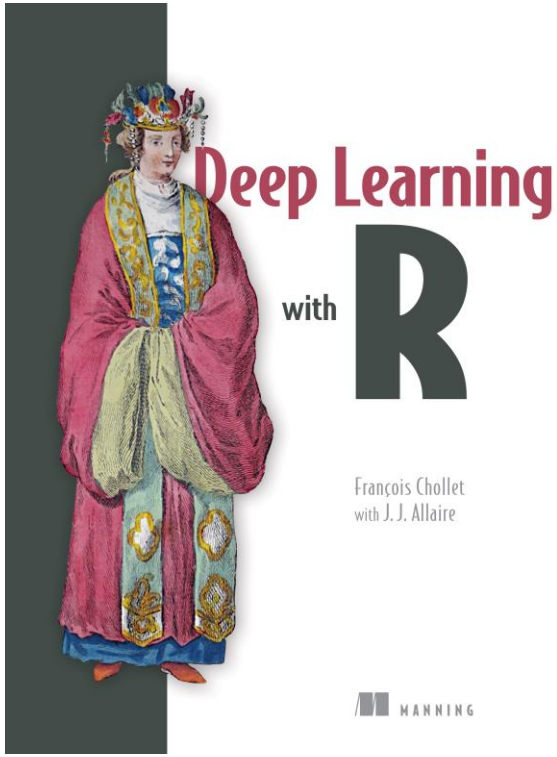

A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

Deep learning in practice

- Good places to start:
  - https://keras.rstudio.com/
- ISLR Chapter 10



Goodfellow et al.



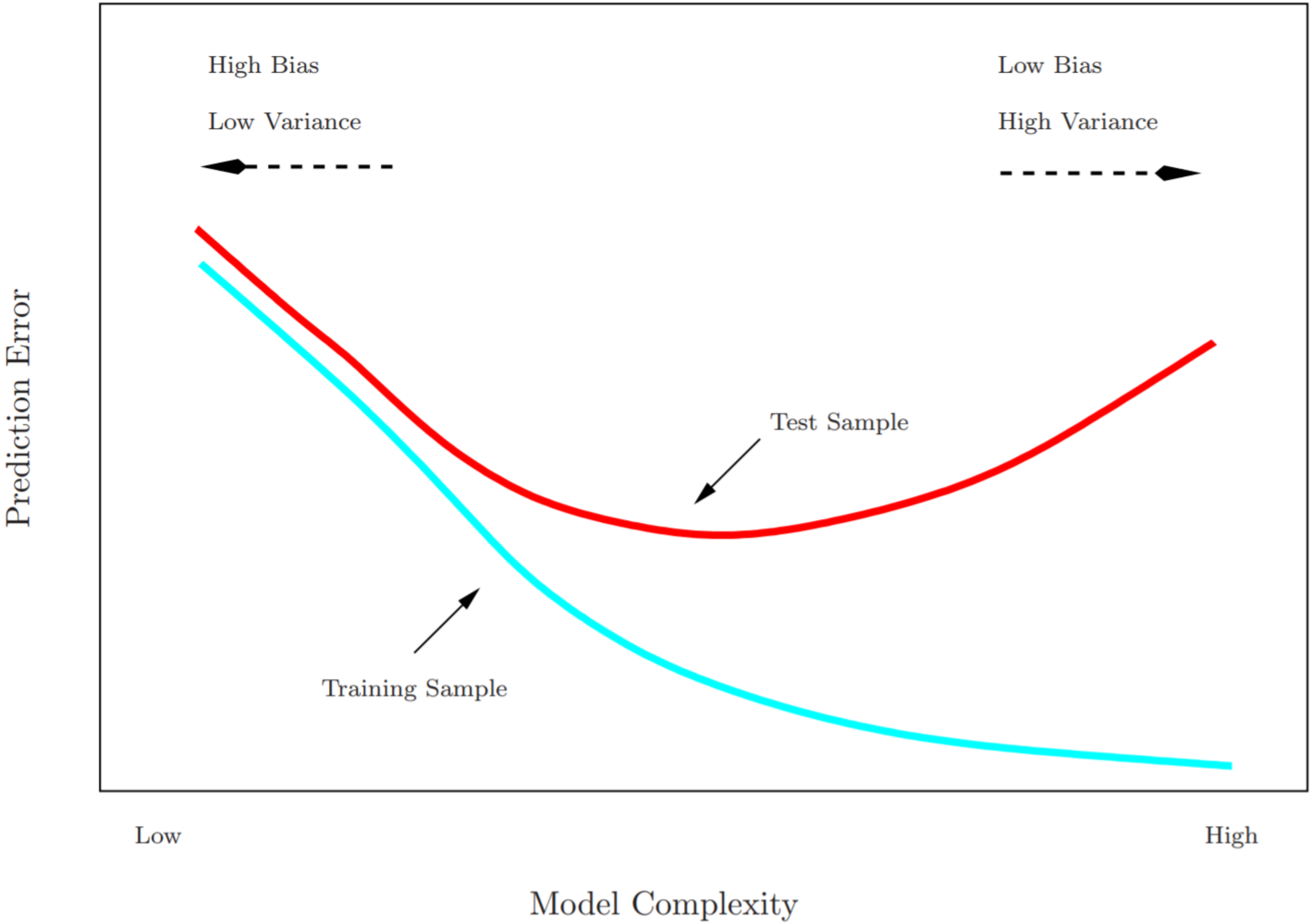
Chollet (R/Python version)

Evaluation

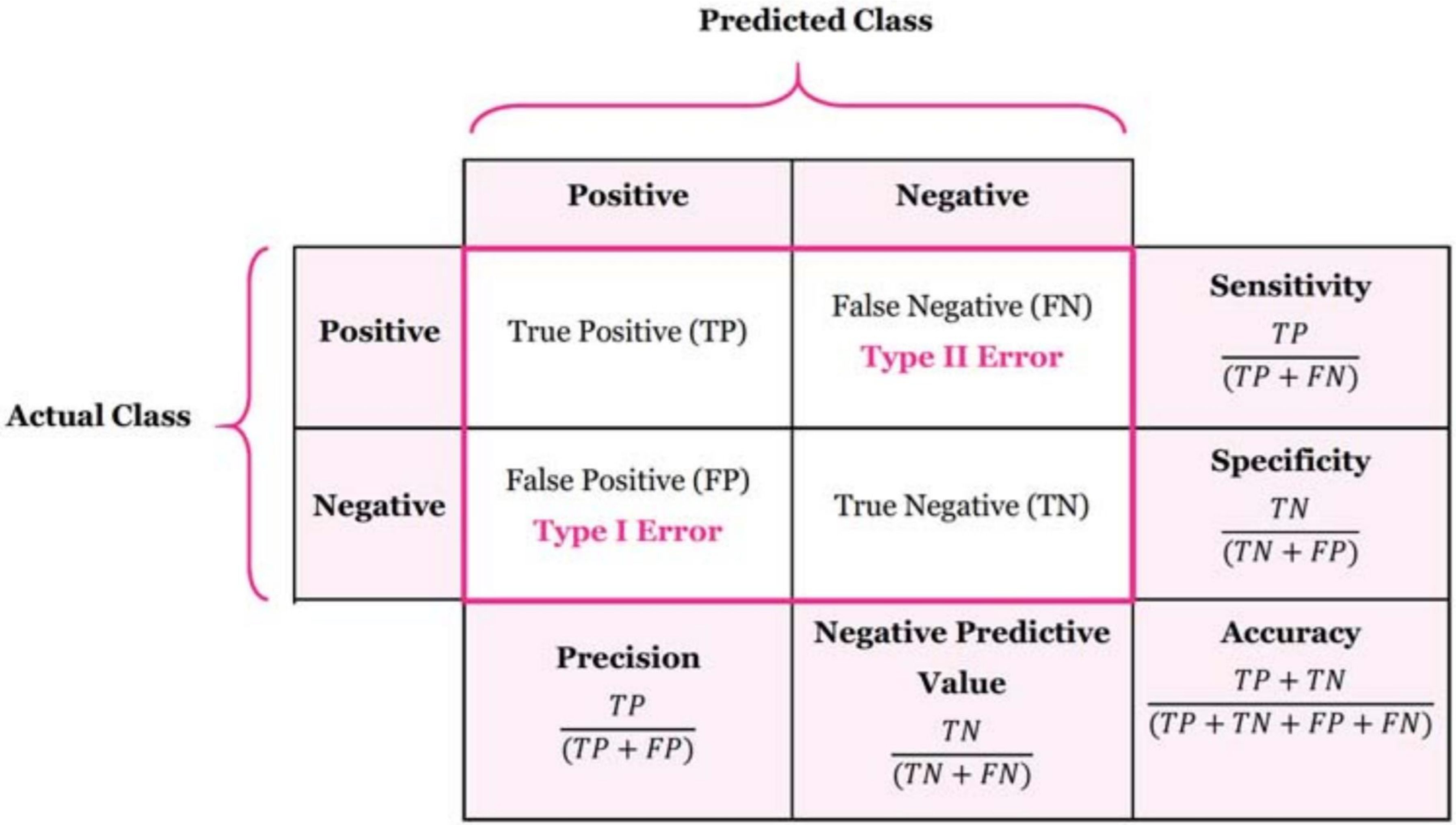
No free lunch

“Any two optimization algorithms are equivalent when their performance is averaged across all possible problems”

(Wolpert & MacReady)


High Bias
Low Variance
Low Bias
High Variance
Test Sample
Training Sample
Prediction Error
Low
High
Model Complexity

Confusion matrix



Predicted Class

		Positive	Negative	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

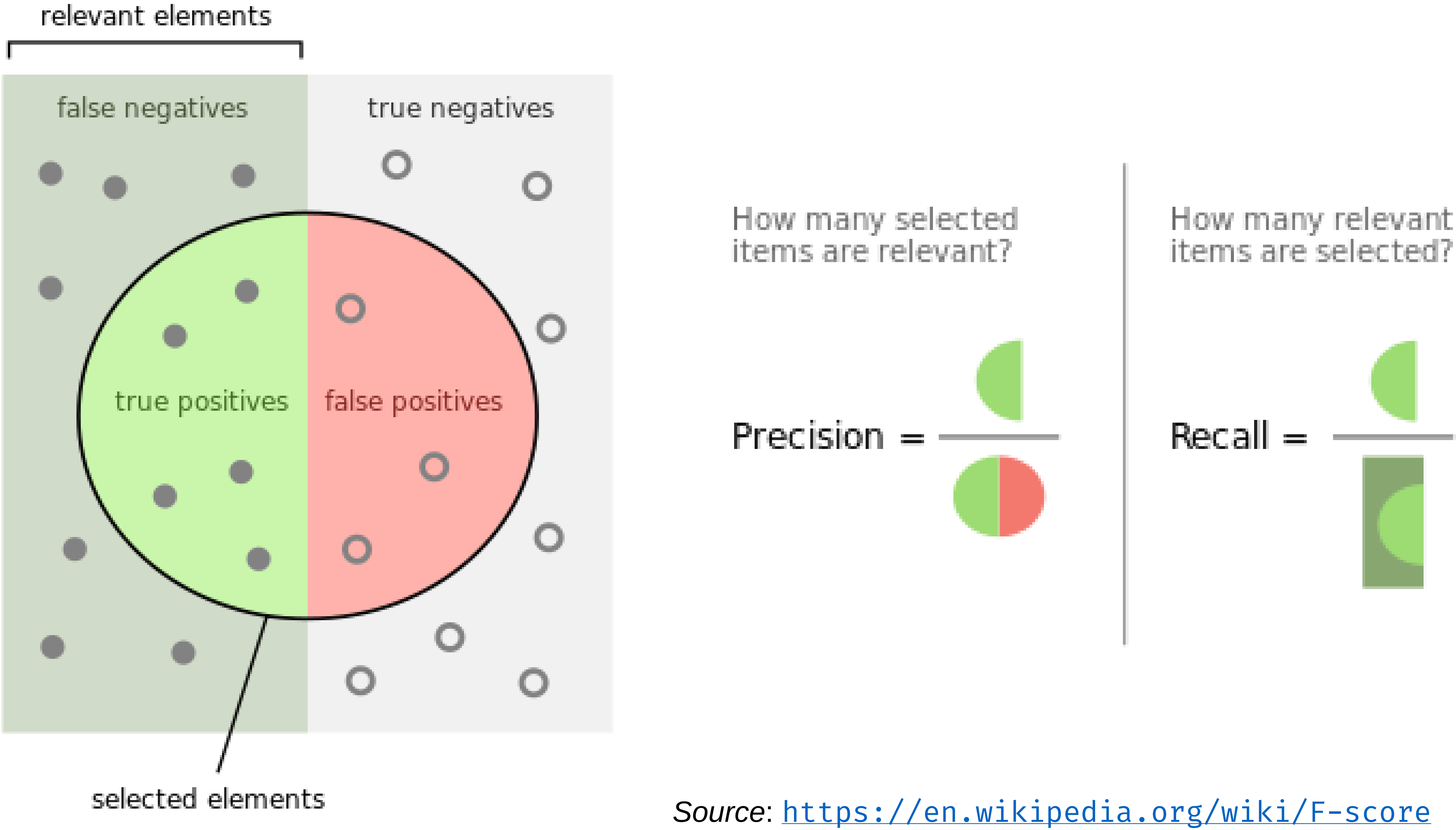
Accuracy

- Accuracy is a valid choice of evaluation for classification problems which are well balanced and not skewed.

Precision and recall

- **Precision**: % of selected items that are correct
  Recall: % of correct items that are selected

- Precision is a valid choice of evaluation metric when we want to be very sure of our prediction.
- Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.



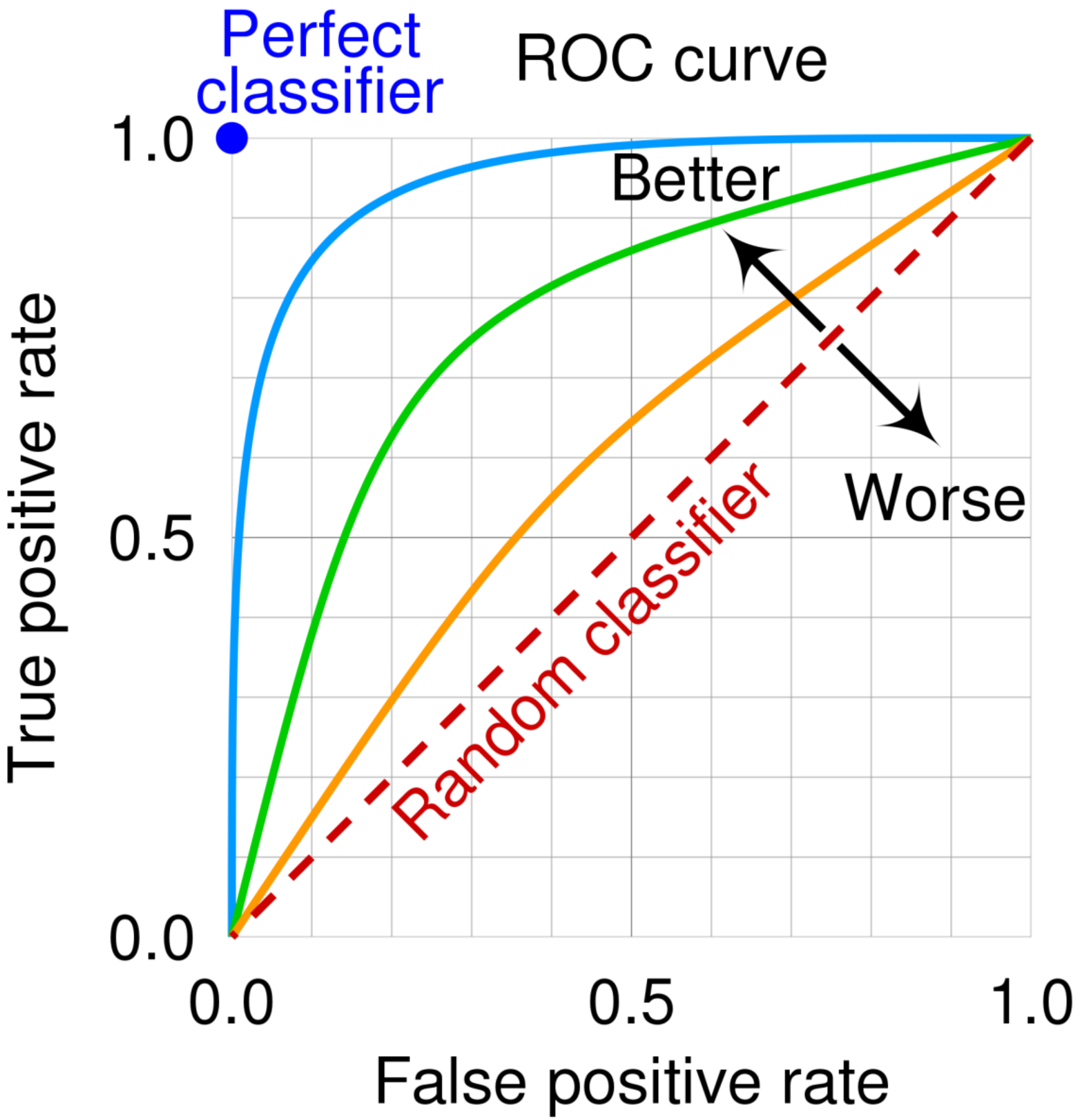
Source: https://en.wikipedia.org/wiki/F-score

A combined measure: F

- A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{a\frac{1}{P} + (1 - a)\frac{1}{R}} = \frac{(b^2 + 1)PR}{b^2 P + R}$$

- The harmonic mean is a very conservative average
- People usually use balanced F1 measure
  - i.e., with $\beta = 1$ (that is, $\alpha = ½$): $F = 2PR/(P+R)$


Perfect classifier
ROC curve
1.0
Better
Worse
0.5
Random classifier
0.0
0.0
0.5
1.0
True positive rate
False positive rate

Practical

Multiclass text classification of 20newsgroups articles.

Questions?