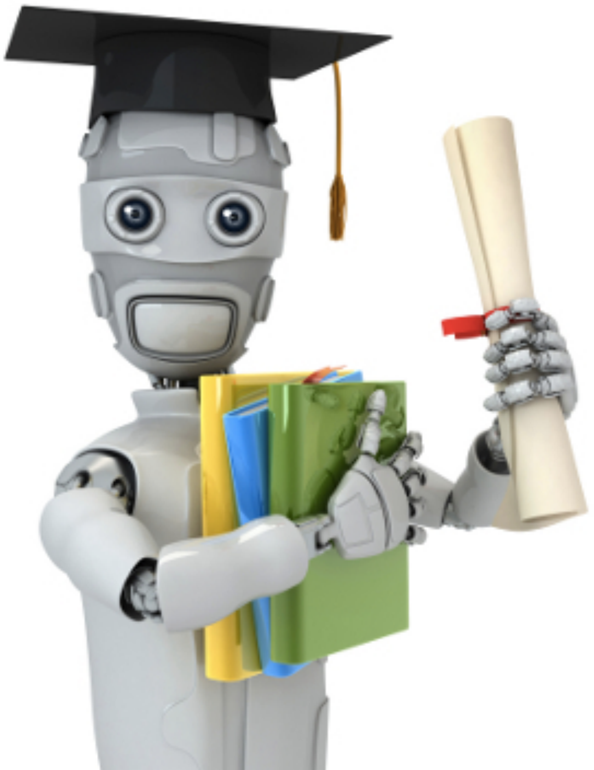
CLASSIFICATION AND CATEGORIZATION

INTRODUCTION TO DATA SCIENCE

ELI UPFAL



teaching datascience .org

MACHINE LEARNING PROBLEMS

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

EXAMPLE: TITANIC DATASET

Label Features

survived	pclass	sex	age	sibsp	parch	fare	cabin	embarked
0	3	male	22	1	0	7.25		S
1	1	female	38	1	0	71.2833	C85	C
1	3	female	26	0	0	7.925		S
1	1	female	35	1	0	53.1	C123	S
0	3	male	35	0	0	8.05		S
0	3	male		0	0	8.4583		Q
0	1	male	54	0	0	51.8625	E46	S
0	3	male	2	3	1	21.075		S
1	3	female	27	0	2	11.1333		S
1	2	female	14	1	0	30.0708		C
1	3	female	4	1	1	16.7	G6	S
1	1	female	58	0	0	26.55	C103	S
0	3	male	20	0	0	8.05		S

Can we predict survival from these features?

THE MACHINE LEARNING FRAMEWORK

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

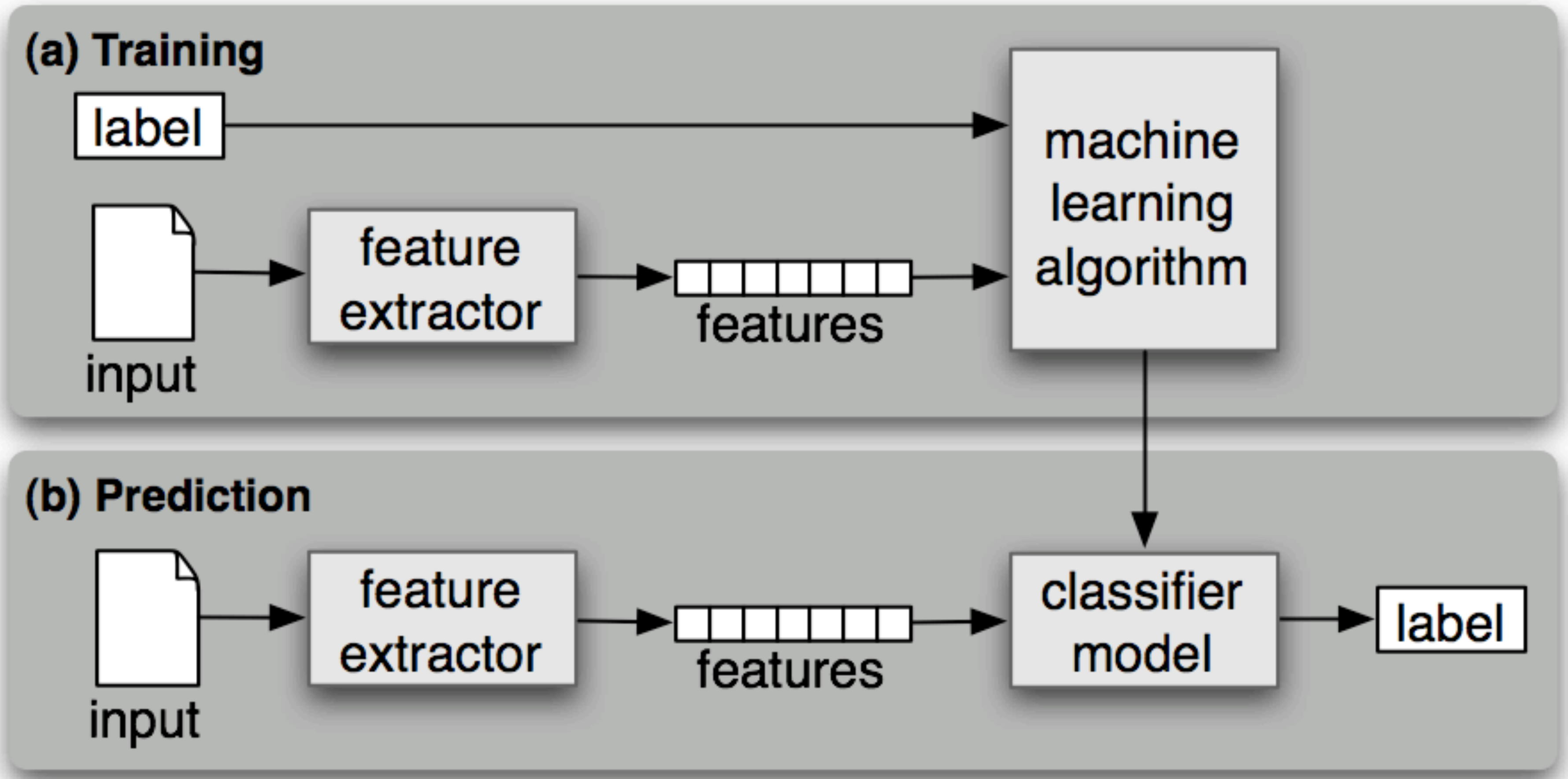
output prediction function features

Training: given a *training set* of labeled examples $\{(x_1,y_1), \ldots, (x_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set

Testing: apply f to a never before seen *test example* x and output the predicted value $y = f(x)$

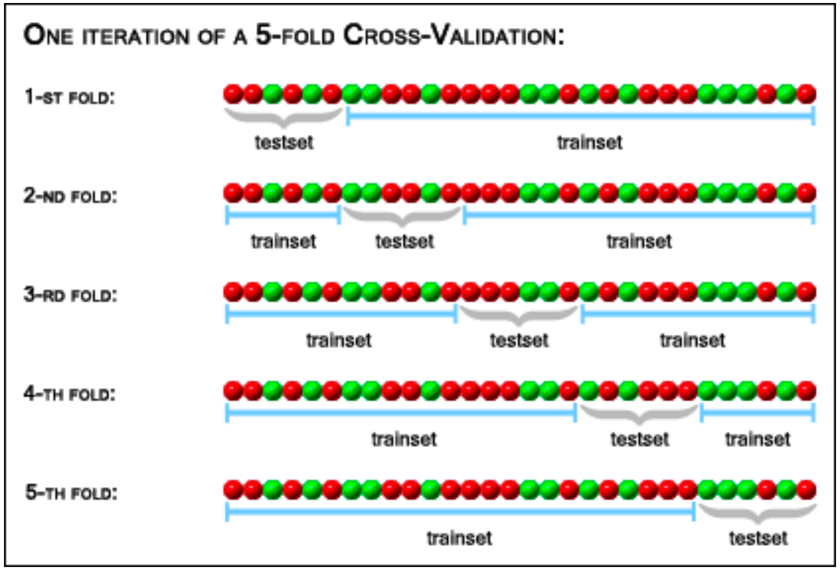
Slide credit: L. Lazebnik

ML PIPELINE (SUPERVISED)



EVALUATION – CROSS-VALIDATION

- Error type:
  - Training error: fraction of errors on training set
  - Generalization error: expected fraction of error on new items
- Estimating generalization error:
  - Hold-out training set - test on fresh items
  - Cross validation, k-fold, leave-one-out,…



CONFUSION TABLE

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

numerical form

predicted→ real ↓	*Class_pos*	*Class_neg*
Class_pos	114	86
Class_neg	7	93

→

percentage form

predicted→ real ↓	*Class_pos*	*Class_neg*
Class_pos	38%	29%
Class_neg	2%	31%

numerical form

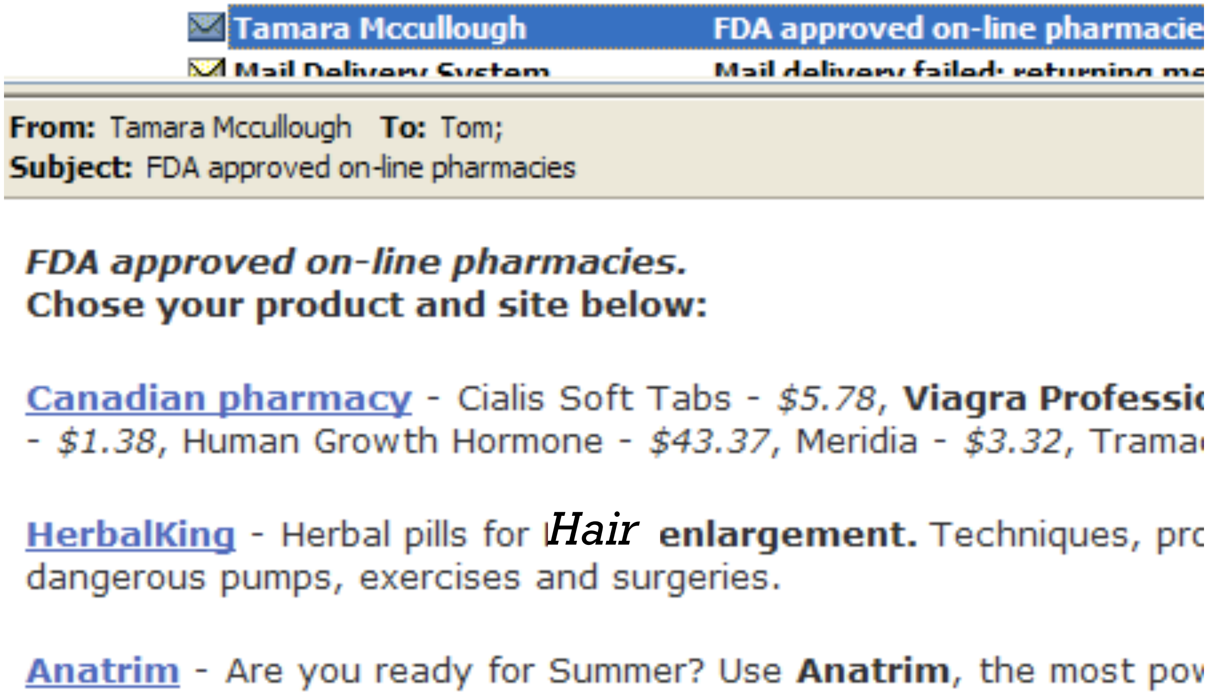
predicted→ real ↓	*Class_1*	*Class_2*	*Class_3*
Class_1	94	16	10
Class_2	21	113	16
Class_3	4	4	92

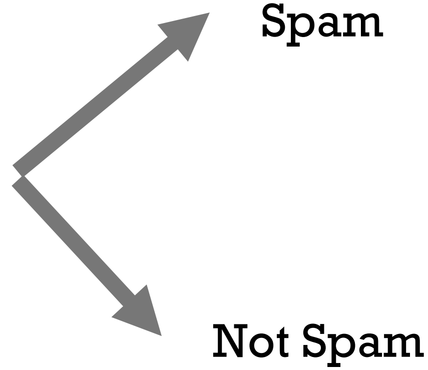
→

percentage form

predicted→ real ↓	*Class_1*	*Class_2*	*Class_3*
Class_1	25%	4%	3%
Class_2	6%	31%	4%
Class_3	1%	1%	25%

TEXT FEATURES





Bag of Words

$$\begin{pmatrix} Viagra \\ Soft \\ Herbel \\ Pills \\ Are \\ \ldots \end{pmatrix}$$

N-Grams

$$\begin{pmatrix} herbel\ pills \\ pills\ for \\ for\ Hair \\ Hair\ enlargement \\ enlargement\ Techniques \\ \ldots \end{pmatrix}$$

TOKENIZATION AND STEMMING

WORKING WITH TEXT

TOKENIZATION

Input: "*Friends, Romans and Countrymen*"

Output: Tokens

- *Friends*
- *Romans*
- *and*
- *Countrymen*

A token is an instance of a sequence of characters

COMMON STEPS

- **Remove Stop Words (a, an, the, to, be, …)**
- **Normalization to terms**
  - **deleting periods:** U.S.A. → USA
  - **deleting hyphens:** *anti-discriminatory* → *antidiscriminatory*
  - **Abbreviations**: Massachusetts Institute of Technology → MIT
  - **Case-folding:** Meal → meal, Brown → brown
  - **Language-issues:** *Tuebingen, Tübingen* → *Tubingen*
  - **asymmetric expansion***: windows* → *window*
  - *…*
  - *What examples above are problematic?*
- **Thesauri and soundex**
  - ***car* = *automobile*** ***color* = *colour***
- ***Stemming***

STEMMING

Reduce terms to their "roots" before indexing

"Stemming" suggest crude affix chopping

- language dependent
- e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.

for example compressed and compression are both accepted as equivalent to compress.

for exampl compress and compress ar both accept as equival to compress

PORTER'S ALGORITHM

Commonest algorithm for stemming English

- Results suggest it's at least as good as other stemming options

Conventions + 5 phases of reductions

- phases applied sequentially
- each phase consists of a set of commands
- sample convention: *Of the rules in a compound command, select the one that applies to the longest suffix.*

TYPICAL RULES IN PORTER

sses → ss

ies → i

ational → ate

tional → tion

Weight of word sensitive rules

(m>1) EMENT →

- *replacement → replac*
- *cement → cement*

OTHER STEMMERS

Other stemmers exist, e.g., Lovins stemmer

- http://www.comp.lancs.ac.uk/computing/research/stemming/general/lovins.htm
- Single-pass, longest suffix removal (about 250 rules)

Full morphological analysis – at most modest benefits for retrieval

Do stemming and other normalizations help?

- English: very mixed results. Helps recall for some queries but harms precision on others
  - E.g., operative (dentistry) ⇒ oper
- Definitely useful for Spanish, German, Finnish, …
  - 30% performance gains for Finnish!

MANY CLASSIFIERS TO CHOOSE FROM

Decision Trees

K-nearest neighbor

Support Vector Machines

Logistic Regression

Naïve Bayes

Random Forrest

Bayesian network

Randomized Forests

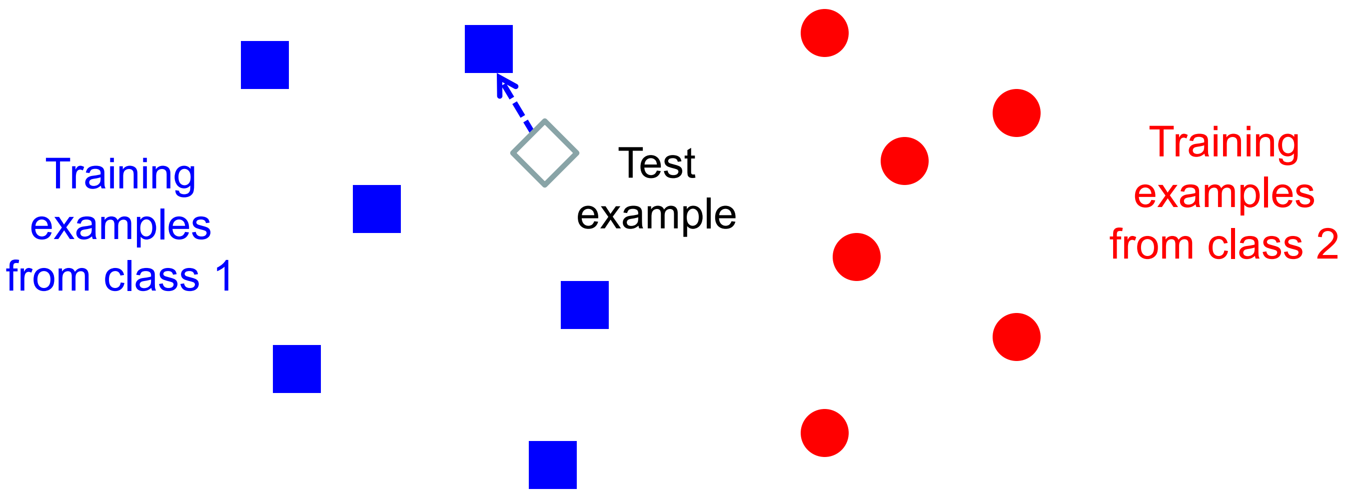
Boosted Decision Trees

RBMs

….

Slide credit: D. Hoiem

CLASSIFIERS: NEAREST NEIGHBOR

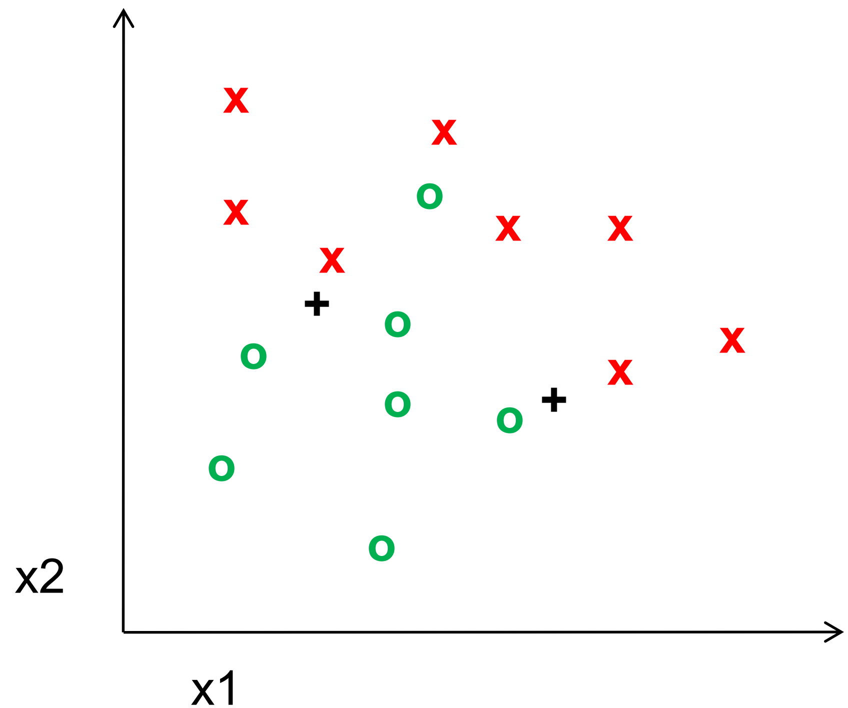


f(**x**) = label of the training example nearest to **x**

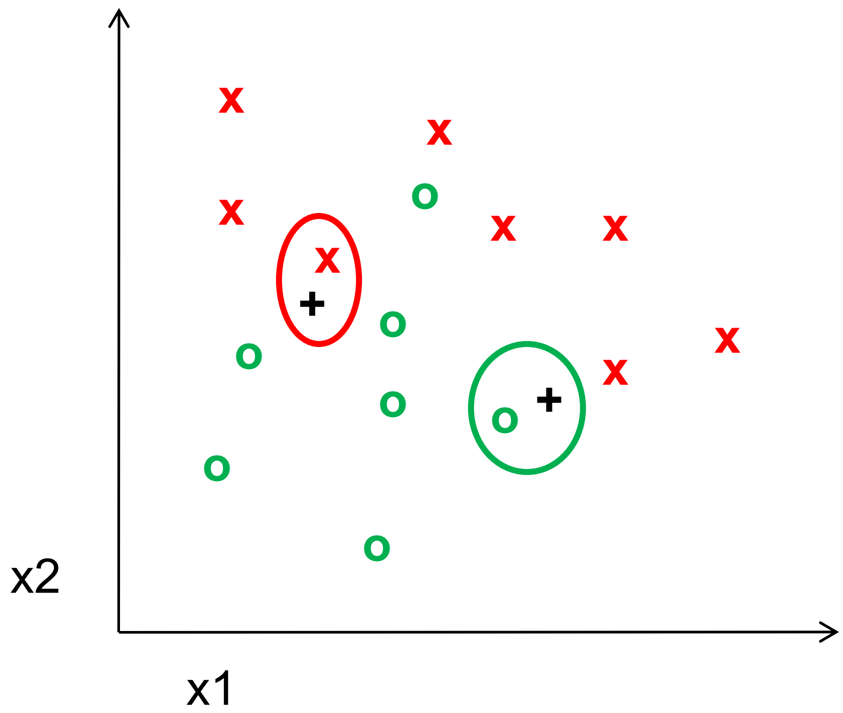
- All we need is a distance function for our inputs
- No training required!

Slide credit: L. Lazebnik

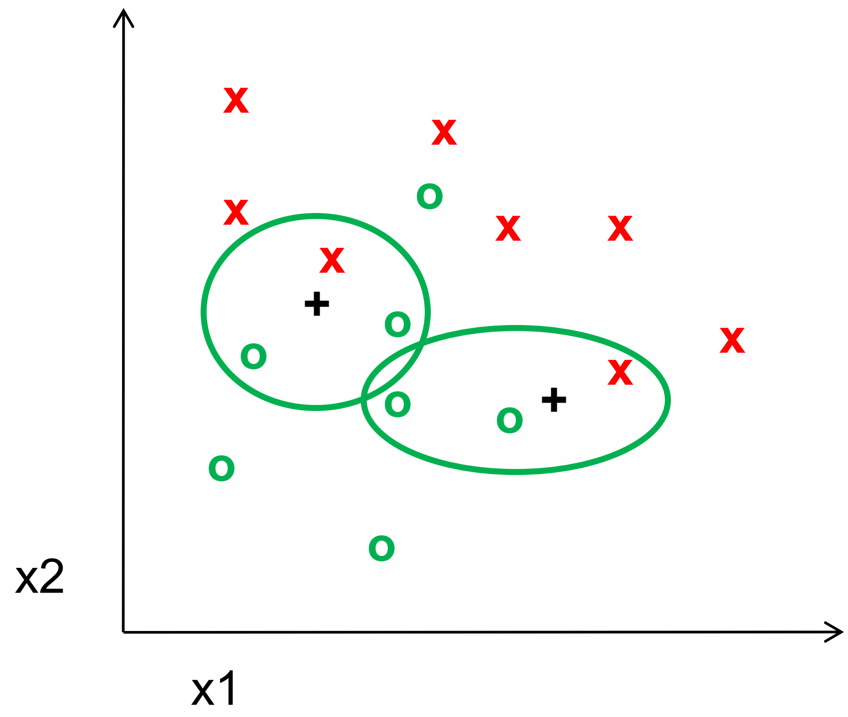
K-NEAREST NEIGHBOR



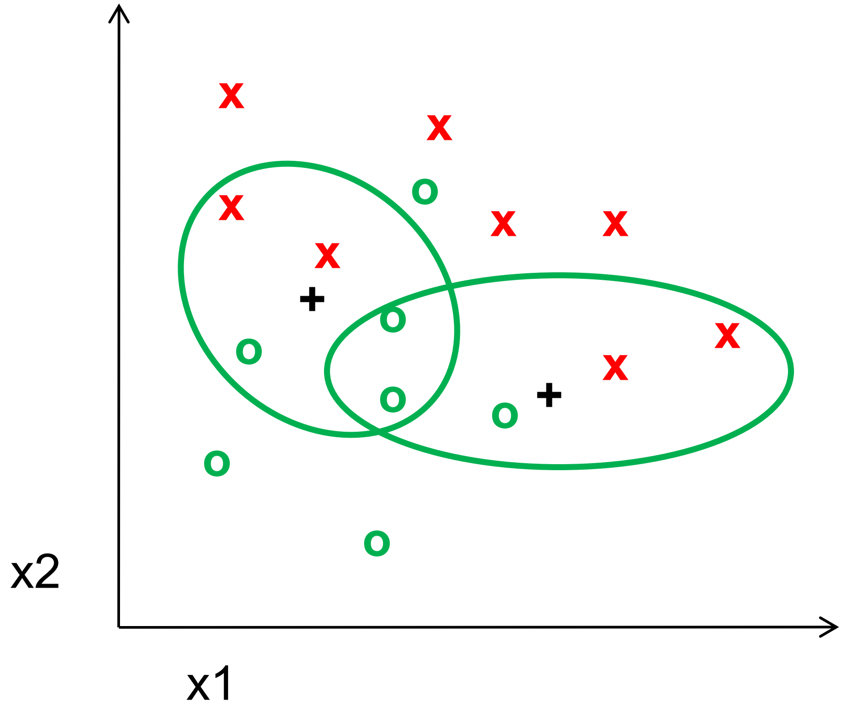
1-NEAREST NEIGHBOR



3-NEAREST NEIGHBOR

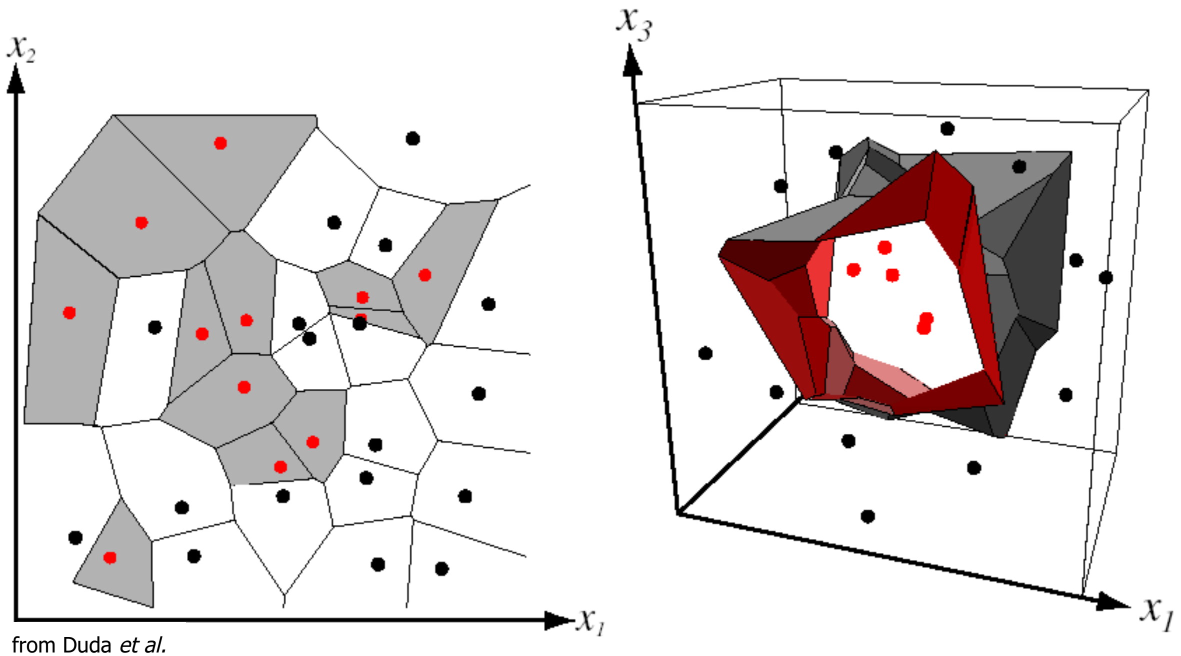


5-NEAREST NEIGHBOR



DECISION BOUNDARIES KNN

Assign label of nearest training data point to each test data point



Voronoi partitioning of feature space
for two-category 2D and 3D data

Source: D. Lowe