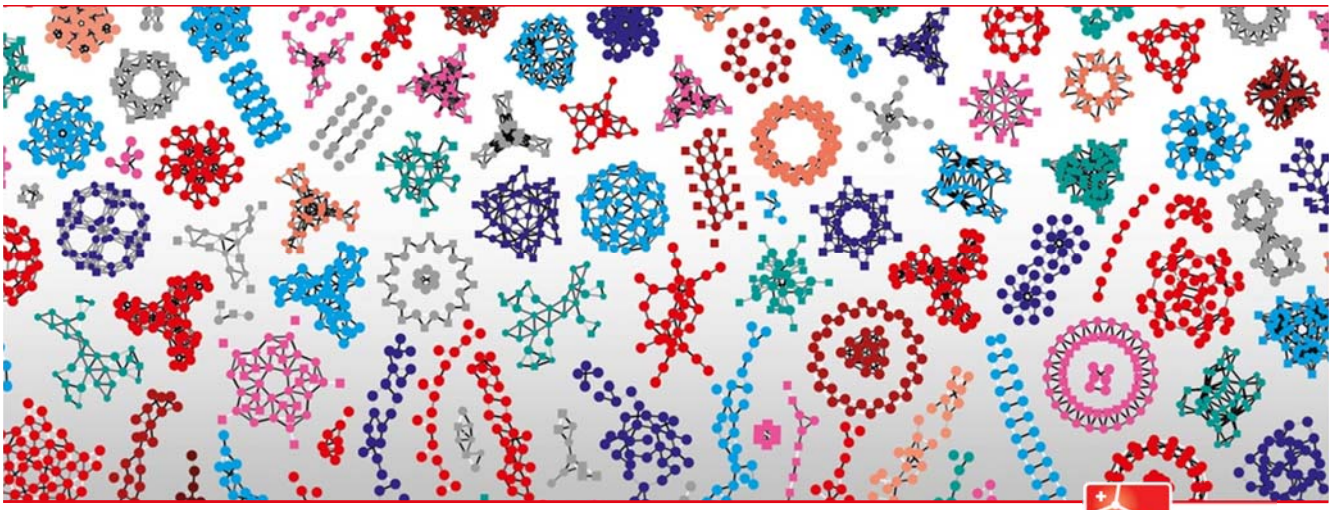


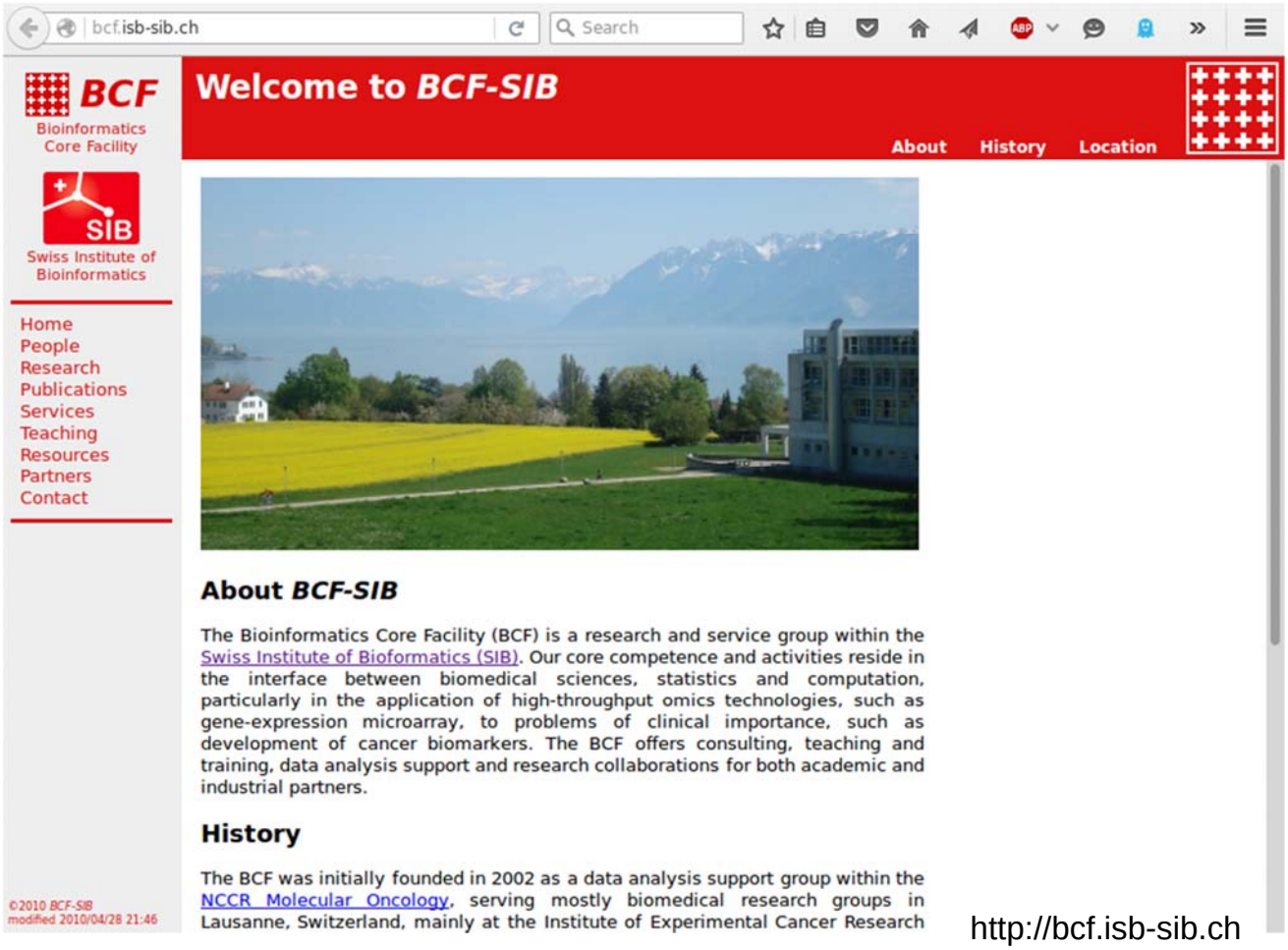
Swiss Institute of Bioinformatics

Machine Learning

Frédéric Schütz

November 2017

www.sib.swiss



http://bcf.isb-sib.ch

bcf.isb-sib.ch/Teaching.html

BCF Bioinformatics Core Facility

SIB Swiss Institute of Bioinformatics

Home
People
Research
Publications
Services
Teaching
Resources
Partners
Contact

Teaching and Training

Upcoming Past

The BCF provides researchers with educational support and practical training in the use of software and analysis methods. This includes the organization of seminars, workshops, statistical software training courses, and teaching in the regular curriculum at the University of Geneva, the University of Lausanne and the EPFL.

The range of topics we have covered includes:

- Introduction to statistics in biomedical sciences
- R statistical software and BioConductor
- Transcriptomics analysis (microarray analysis, RNAseq and qPCR)

These courses are available at both introductory or advanced level. Most courses are taught over a full week; some specialized workshops can be organized over one day, including:

- General statistics in biomedical sciences (for people who want to understand statistics but won't use them directly)
- Multivariate Analysis
- Integration of data from several sources
- Graphical representation of life science data
- Data analysis and reproducible research

We can also offer these courses "in-house", or develop custom courses tailored to your needs and level, according to your requirements. Please contact stat@isb-sib.ch if you have any question.

Upcoming

Our courses upcoming courses are announced on the SIB education web page. You can also sign up to remain informed about the education activities at the SIB.

The organization of our courses depends strongly on the interest of potential participants. If you have any question or suggestion, please contact stat@isb-

bcf.isb-sib.ch/Services.html

BCF Bioinformatics Core Facility

SIB Swiss Institute of Bioinformatics

Home
People
Research
Publications
Services
Teaching
Resources
Partners
Contact

Services

SIB Biostat Teaching Consulting Analysis Collaboration Embedding

SIB Biostatistics Support

The BCF provides a consulting service on biostatistics matters, on a mandate from (and partially funded by) the SIB and the Swiss Confederation. This service is aimed at all people active in life sciences in Switzerland. It includes training and teaching, consulting, data analysis, and research collaboration, with a focus on high-throughput technologies in genomics or proteomics.

The service can be provided on a collaborative basis or for a fee, depending on the circumstances: among other factors, the origin and goals of the request (academy or industry), the amount of work involved and our current workload will be taken into account in determining the service provided. For academic groups that require long-term support, we strongly advise to start a discussion at the grant-submission step, and to include a request for a part-time bioinformatician in the grant. By pooling such part-time positions, the BCF is able to offer a longer-term dedicated support.

Consulting usually starts with a short meeting discussing the questions asked. Often, this is enough to help the researcher solve the problem. In other cases, the meeting allows us to define the different possibilities for a forthcoming collaboration.

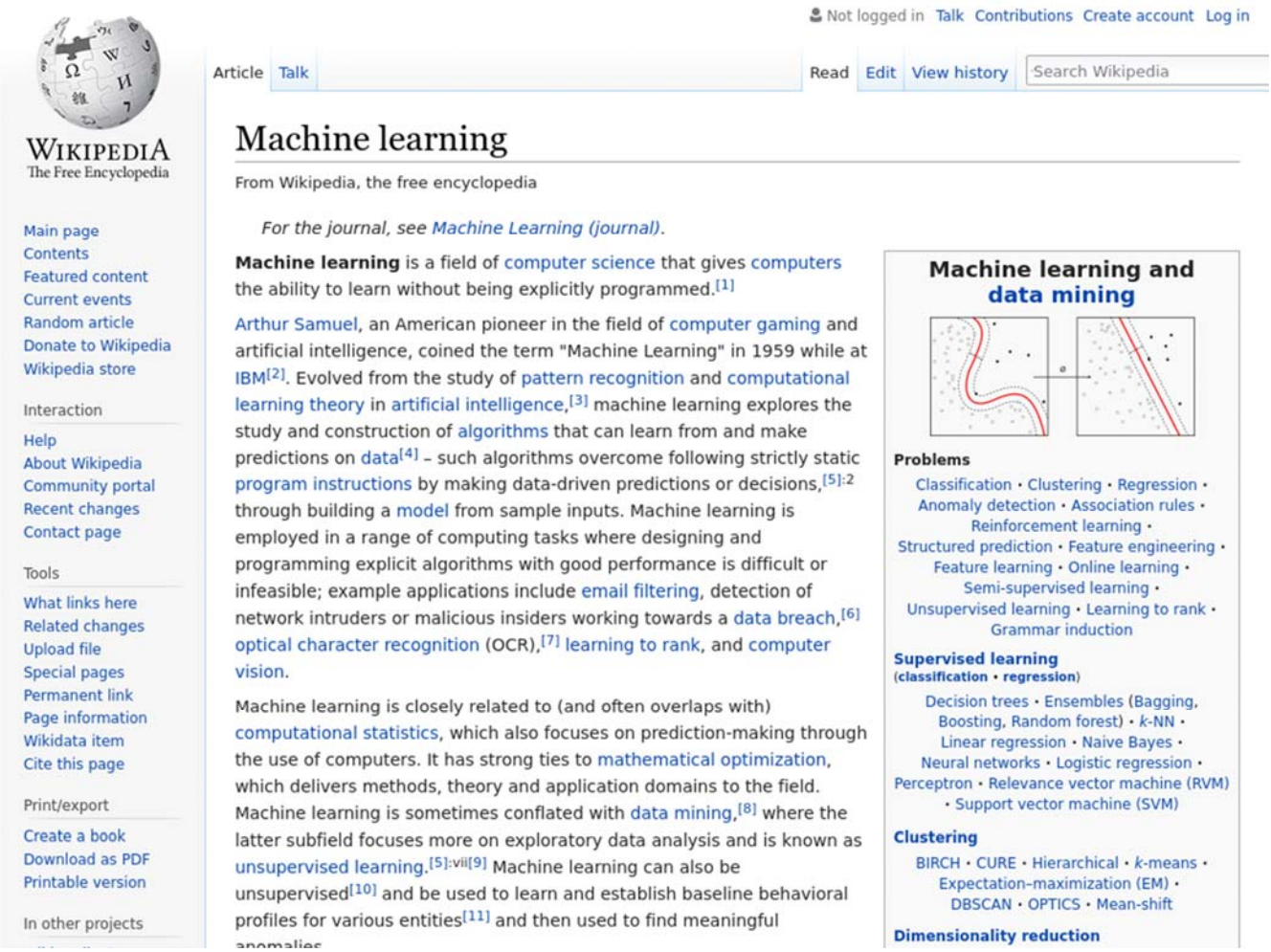
For more information, please contact us at stat@isb-sib.ch or by calling Frédéric Schütz at +41 21 692 40 94 or Charlotte Soneson at +41 21 692 40 91.

Teaching and Training

We provide short courses and workshops, as well as longer but low-intensity semester courses. More information about recent and upcoming courses is available on the SIB education web page. The Teaching page holds information about courses up to 2011. You can also sign up to remain informed about the education activities at the SIB.

http://bcf.isb-sib.ch/Services.html

Machine learning ?



Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects

Machine learning

From Wikipedia, the free encyclopedia

For the journal, see Machine Learning (journal).

Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed.[1]

Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM[2]. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence,[3] machine learning explores the study and construction of algorithms that can learn from and make predictions on data[4] – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions,[5]:2 through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach,[6] optical character recognition (OCR),[7] learning to rank, and computer vision.

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining,[8] where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.[5]:vii[9] Machine learning can also be unsupervised[10] and be used to learn and establish baseline behavioral profiles for various entities[11] and then used to find meaningful anomalies.

Machine learning and data mining

Problems
Classification • Clustering • Regression • Anomaly detection • Association rules • Reinforcement learning • Structured prediction • Feature engineering • Feature learning • Online learning • Semi-supervised learning • Unsupervised learning • Learning to rank • Grammar induction

Supervised learning
(classification • regression)
Decision trees • Ensembles (Bagging, Boosting, Random forest) • *k*-NN • Linear regression • Naive Bayes • Neural networks • Logistic regression • Perceptron • Relevance vector machine (RVM) • Support vector machine (SVM)

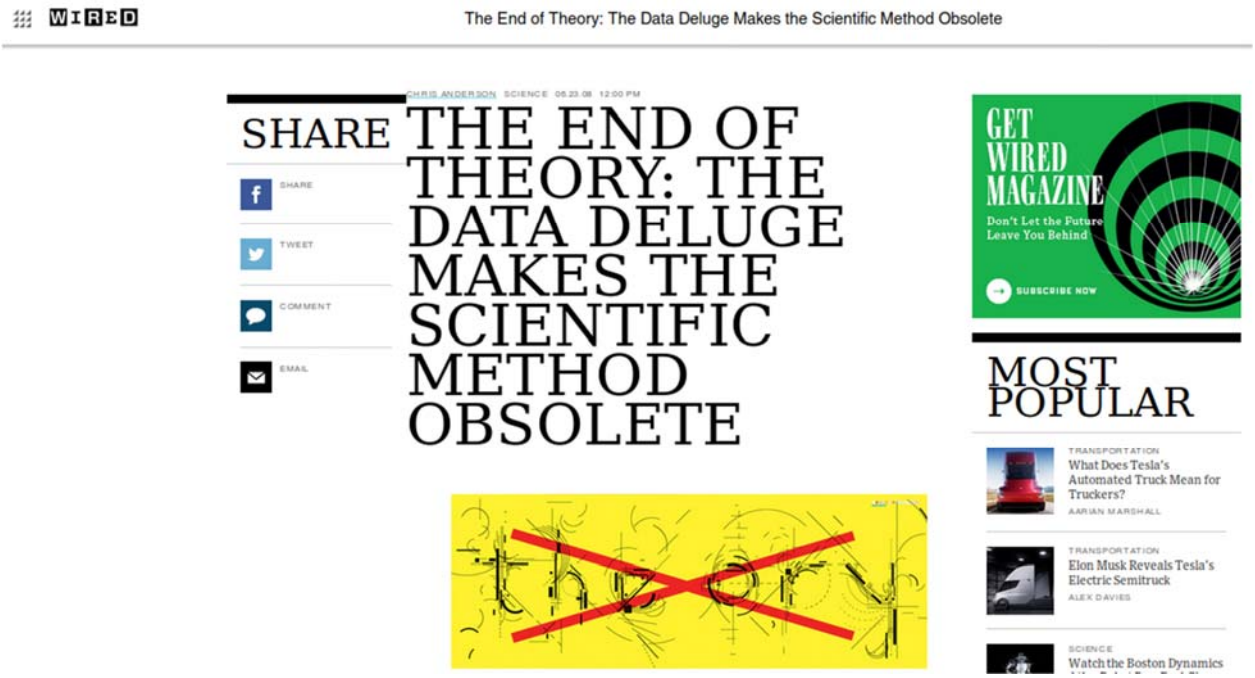
Clustering
BIRCH • CURE • Hierarchical • *k*-means • Expectation-maximization (EM) • DBSCAN • OPTICS • Mean-shift

Dimensionality reduction

Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed. […]

Machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs.

Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible.



https://www.wired.com/2008/06/pb-theory/

Today companies like Google, which have grown up in an era of massively abundant data, don't have to settle for wrong models. Indeed, they don't have to settle for models at all. […]

This is a world where massive amounts of data and applied mathematics replace every other tool that might be brought to bear. Out with every theory of human behavior, from linguistics to sociology. Forget taxonomy, ontology, and psychology. […]

With enough data, the numbers speak for themselves.

Scientists are trained to recognize that correlation is not causation, that no conclusions should be drawn simply on the basis of correlation between X and Y (it could just be a coincidence). Instead, you must understand the underlying mechanisms that connect the two. Once you have a model, you can connect the data sets with confidence. Data without a model is just noise.

But faced with massive data, this approach to science — hypothesize, model, test — is becoming obsolete.

Now biology is heading in the same direction.

There is now a better way. Petabytes allow us to say: "Correlation is enough." We can stop looking for models. We can analyze the data without hypotheses about what it might show. We can throw the numbers into the biggest computing clusters the world has ever seen and let statistical algorithms find patterns where science cannot.

https://xkcd.com/1838/

“ Models are opinions embedded in mathematics.”

– *Cathy O'Neil, "Weapons of Math Destruction"*

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers.

It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field.

Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.

According to the Gartner hype cycle of 2016, machine learning is at its peak of inflated expectations.

Effective machine learning is difficult because finding patterns is hard and often not enough training data is available; as a result, machine-learning programs often fail to deliver.

What is a statistical model ? One definition (from Terry Speed)

A **statistical model** is a set of equations involving random variables, with associated distributional assumptions, devised in the context of a **question** and a body of **data concerning some phenomenon**, with which **tentative answers** can be derived, along with **measures of uncertainty** concerning these answers.

questions + ***data*** $\xrightarrow[\textbf{\textit{model}}]{}$ ***answers*** + ***measures of uncertainty***

Differents families of machine-learning algorithms

- Association rules learning
- Bagging
- Bayesian classifiers
- Bayesian networks
- Boosting
- Deep learning
- Decision trees
- Discriminant analysis
- Generalized linear models
- Genetic algorithms
- Logistic and multinomial regression
- Multiple adaptive regression splines
- Nearest-neighbours
- Neural networks
- Partial least squares and principal component regression
- Random forest
- Reinforcement learning
- Rule-based classifiers
- Stacking
- Support vector machines
- …

Wikipedia + Journal of Machine Learning Research 15 (2014) 3133-318

Classifying machine learning tasks

(At least) two different types of machine learning algorithms

Supervised learning: the system is provided with existing inputs and the corresponding (expected) outputs, and must learn how to predict the correct output for new (future) inputs

Unsupervised learning: the system is provided with existing inputs, and it must learn from them in order to find structure in the data.

Examples of unsupervised learning

- Hierarchical clustering
- K-Means
- Principal component analysis

Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.

Two typical kind of outputs we want from a ML algorithm

Classification: the inputs belong to two or more classes, and the system must be able to assign new (future) inputs into one (or more) of these classes

Regression: the outputs are continuous instead of discrete.

(regression: a measure of the relation between the mean of a variable and the values of other variables)

Classification

Historically, *objects* are classified into *groups*

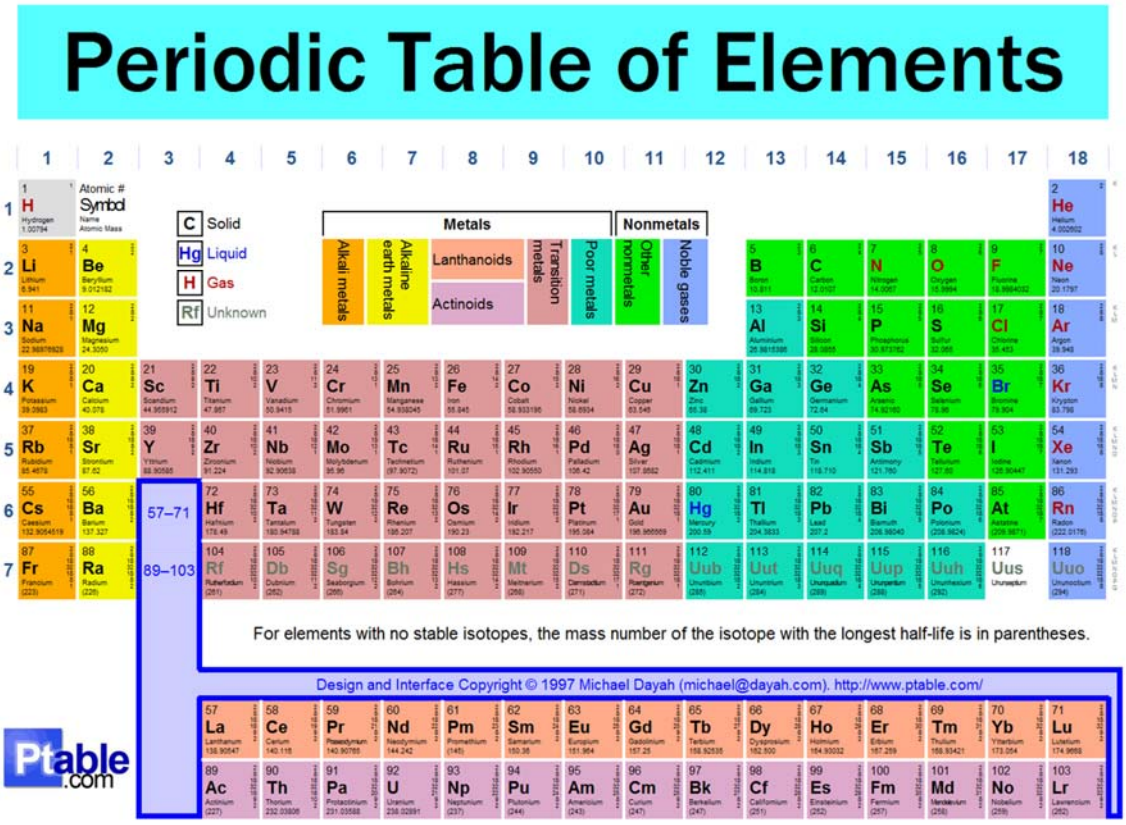
- periodic table of the elements (chemistry)
- taxonomy (zoology, botany)

Why classify?

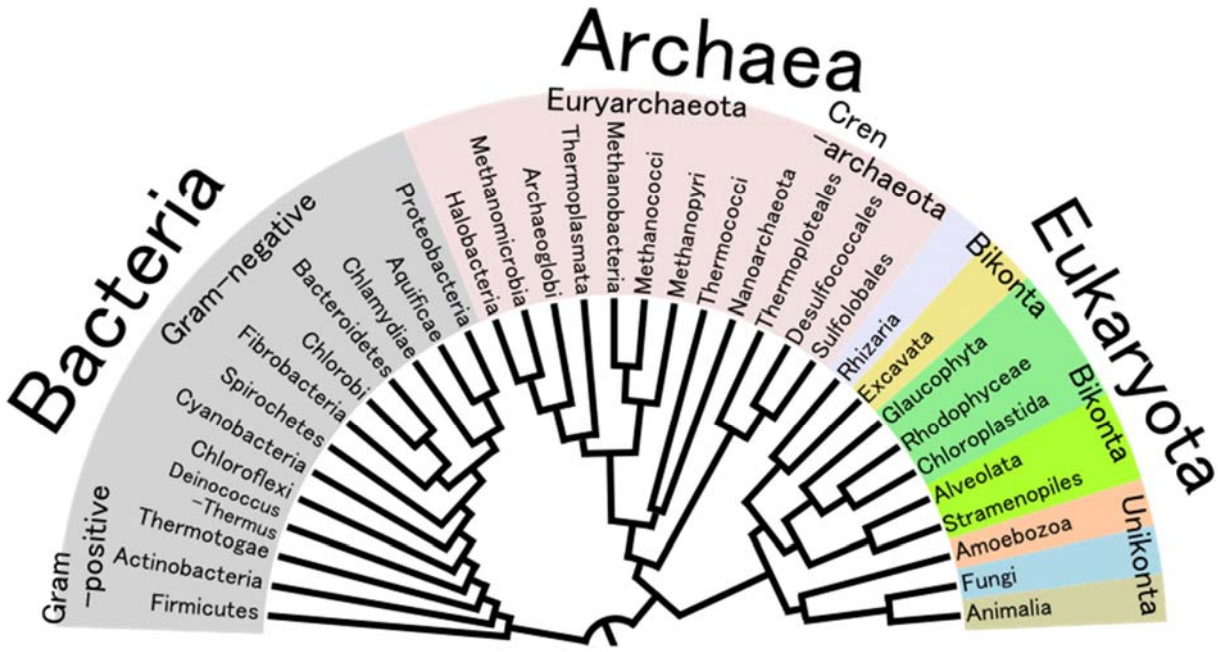
- organizational convenience, convenient summary
- prediction
- explanation

Note: these aims do not necessarily lead to the same classification; e.g. *SIZE* of object in hardware store vs. *TYPE/USE* of object

Example of classification



Example of classification



Typical questions in statistics

Class comparison

« Which measurements are significantly different between the two (or more) experimental conditions ?»

Class discovery

(unsupervised learning)

« Can I identify homogeneous subgroups of samples which are characterized by similar measurements profiles ?»

Class prediction

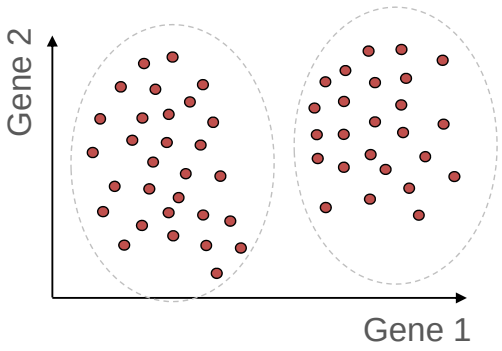
(supervised learning)

« Can I find a rule to classify my samples in known groups using my measurements » ?

Class discovery vs class prediction

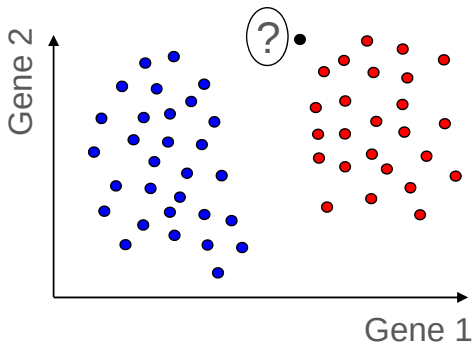
Example: patients from which we obtained measurements (e.g. gene expression)

Class discovery



Find natural groups in the data (e.g. sets of patients with similar gene expression)

Class prediction



Given previous measurements for which the grouping is known (**red** and **blue**), can we predict the group to which a new observation belong ?

Examples of class prediction questions in biology and medicine

- Does a patient have a predisposition for a given disease ?
- What is the prognosis for this patient ?
- What will be the response of this patient to a given drug ?
- Is this tumour benign or malign ?
- What type is this tumour ?
- Which treatment should we use ?
- Does this new organism look like anything known already ?

VOLUME 30 · NUMBER 12 · APRIL 20 2012

JOURNAL OF CLINICAL ONCOLOGY

ORIGINAL REPORT

Identification of a Poor-Prognosis *BRAF*-Mutant–Like Population of Patients With Colon Cancer

Vlad Popovici, Eva Budinska, Sabine Tejpar, Scott Weinrich, Heather Estrella, Graeme Hodgson, Eric Van Cutsem, Tao Xie, Fred T. Bosman, Arnaud D. Roth, and Mauro Delorenzi

VOLUME 30 · NUMBER 12 · APRIL 20 2012

JOURNAL OF CLINICAL ONCOLOGY

ORIGINAL REPORT

A B S T R A C T

Purpose

Our purpose was development and assessment of a *BRAF*-mutant gene expression signature for colon cancer (CC) and the study of its prognostic implications.

Materials and Methods

A set of 668 stage II and III CC samples from the PETACC-3 (Pan-European Trails in Alimentary Tract Cancers) clinical trial were used to assess differential gene expression between c.1799T>A (p.V600E) *BRAF* mutant and non-*BRAF*, non-*KRAS* mutant cancers (double wild type) and to construct a gene expression–based classifier for detecting *BRAF* mutant samples with high sensitivity. The classifier was validated in independent data sets, and survival rates were compared between classifier positive and negative tumors.

Results

A 64 gene-based classifier was developed with 96% sensitivity and 86% specificity for detecting *BRAF* mutant tumors in PETACC-3 and independent samples. A subpopulation of *BRAF* wild-type patients (30% of *KRAS* mutants, 13% of double wild type) showed a gene expression pattern and had poor overall survival and survival after relapse, similar to those observed in *BRAF*-mutant patients. Thus they form a distinct prognostic subgroup within their mutation class.

Conclusion

A characteristic pattern of gene expression is associated with and accurately predicts *BRAF* mutation status and, in addition, identifies a population of *BRAF* mutated-like *KRAS* mutants and double wild-type patients with similarly poor prognosis. This suggests a common biology between these tumors and provides a novel classification tool for cancers, adding prognostic and biologic information that is not captured by the mutation status alone. These results may guide therapeutic strategies for this patient segment and may help in population stratification for clinical trials.

Machine learning and R

CRAN: R packages and task views

cran.r-project.org

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods

cran.r-project.org/web/views/MachineLearning.html

CRAN Task View: Machine Learning & Statistical Learning

Maintainer: Torsten Hothorn
Contact: Torsten.Hothorn at R-project.org
Version: 2012-10-30

Several add-on packages implement ideas and methods developed at the borderline between computer science and statistics - this field of research is usually referred to as machine learning. The packages can be roughly structured into the following topics:

- *Neural Networks* : Single-hidden-layer neural network are implemented in package nnet (shipped with base R). Package RSNNS offers an interface to the Stuttgart Neural Network Simulator (SNNS).
- *Recursive Partitioning* : Tree-structured models for regression, classification and survival analysis, following the ideas in the CART book, are implemented in rpart (shipped with base R) and tree. Package rpart is recommended for computing CART-like trees. A rich toolbox of partitioning algorithms is available in Weka , package RWeka provides an interface to this implementation, including the J4.8-variant of C4.5 and M5. The Cubist package fits rule-based models (similar to trees) with linear regression models in the terminal leaves, instance-based corrections and boosting. The C50 package can fit C5.0 classification trees, rule-based models, and boosted versions of these.

  Two recursive partitioning algorithms with unbiased variable selection and statistical stopping criterion are implemented in package party. Function `ctree()` is based on non-parametrical conditional inference procedures for testing independence between response and each input variable whereas `mob()` can be used to partition parametric models. Extensible tools for visualizing binary trees and node distributions of the response are available in package party as well.

Bioconductor

www.bioconductor.org

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

Home | Install | Help | Developers | About

About Bioconductor

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, 610 software packages, and an active user community. Bioconductor is also available as an Amazon Machine Image (AMI).

Use Bioconductor for...

Microarrays
Import Affymetrix, Illumina, Nimblegen, Agilent, and other platforms. Perform quality assessment, normalization, differential expression, clustering, classification, gene set enrichment, genetical genomics and other workflows for expression, exon, copy number, SNP, methylation and other assays. Access GEO, ArrayExpress, Biomart, UCSC, and other community resources.

Variants
Read and write VCF files. Identify structural location of variants and compute amino acid coding changes for non-synonymous variants. Use SIFT and PolyPhen database packages to predict consequence of amino acid coding changes.

Sequence Data
Import fasta, fastq, ELAND, MAQ, BWA, Bowtie, BAM, gff, bed, wig, and other sequence formats. Trim, transform, align, and manipulate sequences. Perform quality assessment, ChIP-seq, differential expression, RNA-seq, and other workflows. Access the Sequence Read Archive.

Transcription Factors
Find candidate binding sites for known transcription factors via sequence matching.

Mailing Lists — Subscribe »

Events
Next Generation Data Analysis Workshop

News
Bioconductor 2.11 released

Bioconductor: classification software

www.bioconductor.org/packages/release/BiocViews.html#___Classification

biocviews

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

Home | Install | Help | Developers | About

Home » BiocViews

All Packages

Bioconductor version 2.11 (Release)

- Software (608)
  - Annotation (82)
  - AssayDomains (236)
  - AssayTechnologies (357)
  - Bioinformatics (394)
    - Classification (33)
    - Clustering (52)
    - Enrichment (14)
    - MultipleComparisons (44)
    - Networks (23)
    - Preprocessing (91)
    - QualityControl (54)
    - SequenceMatching (14)
    - TimeCourse (17)
    - Visualization (102)
  - BiologicalDomains (80)
  - Infrastructure (167)

Packages

Package	Maintainer	Title
BioSeqClass	Li Hong	Classification for Biological Sequences
cancerclass	Daniel Kosztyla	Development and validation of diagnostic tests from high-dimensional molecular data
Clonality	Irina Ostrovnaya	Clonality testing
clst	Noah Hoffman	Classification by local similarity threshold
clstutils	Noah Hoffman	Tools for performing taxonomic assignment.
CMA	Christoph Bernau	Synthesis of microarray-based classification
CPImage	Henrik Failmezger, Yinyin Yuan	CPImage a package to classify cells and calculate tumour cellularity
ctc	Antoine Lucas	Cluster and Tree Conversion.
DirichletMultinomial	Martin Morgan	Dirichlet-Multinomial Mixture Model Machine Learning for Microbiome Data
eisa	Gabor Csardi	Expression data analysis via the Iterative Signature Algorithm
ExpressionView	Gabor Csardi	Visualize biclusters identified in gene expression data
fastseg	Guenter Klambauer	fastseg - a fast segmentation algorithm
flowPhyto	David M. Schruth	Methods for Continuous Flow Cytometry
gene2pathway	Holger Froehlich	Prediction of KEGG pathway membership for individual genes based on InterPro domain signatures

Some classification methods and R packages)

• Classification And REgression Training	`caret`
• K-nearest neighbours	`class`
• Linear Discriminant Analysis (LDA)	`MASS, sda`
• Quadratic Discriminant Analysis (QDA)	`MASS`
• Classification trees	`rpart`
• Support Vector Machines (SVM)	`e1071`
• Random Forest	`randomForest`

etc.

Our program

Introduction

Examples of machine-learning algorithms

- Nearest-neighbors
- Linear discriminant analysis

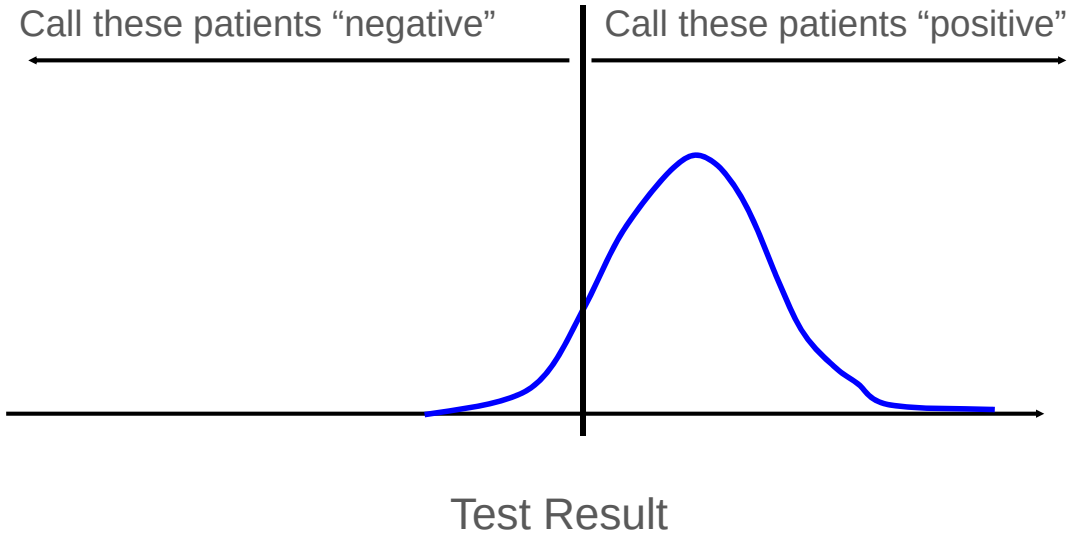
Assessing the performance of machine-learning algorithms

Some more machine learning algorithms

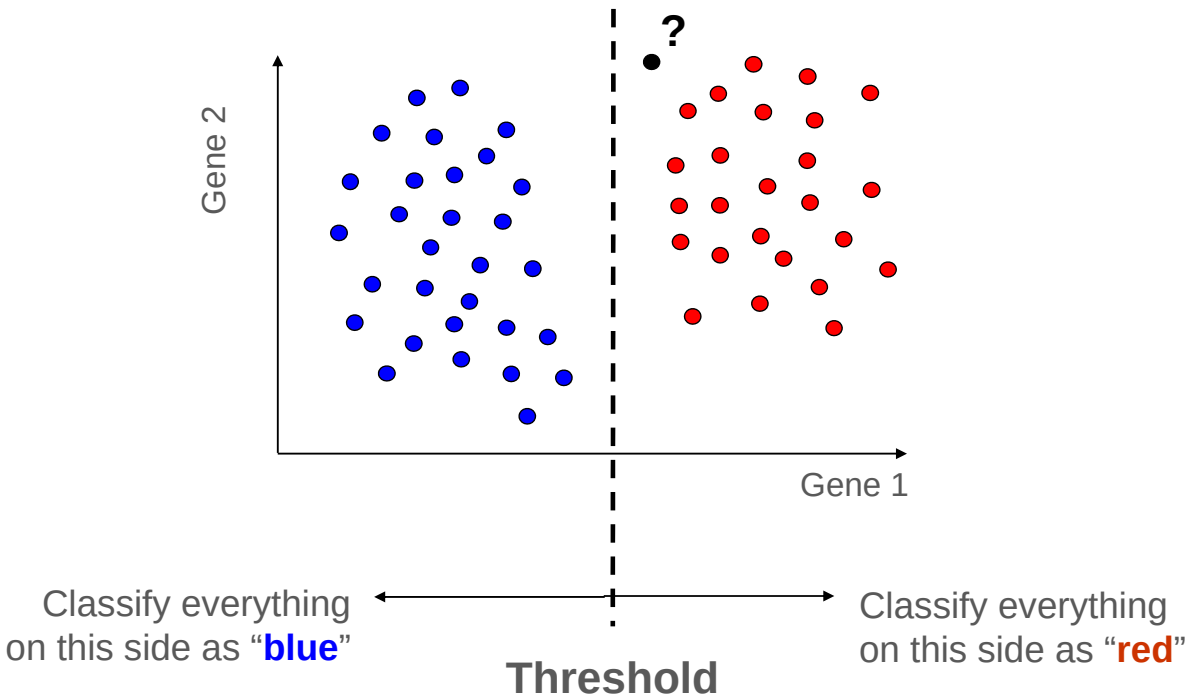
- Random Forests
- Support Vector Machines

Regression vs Classification

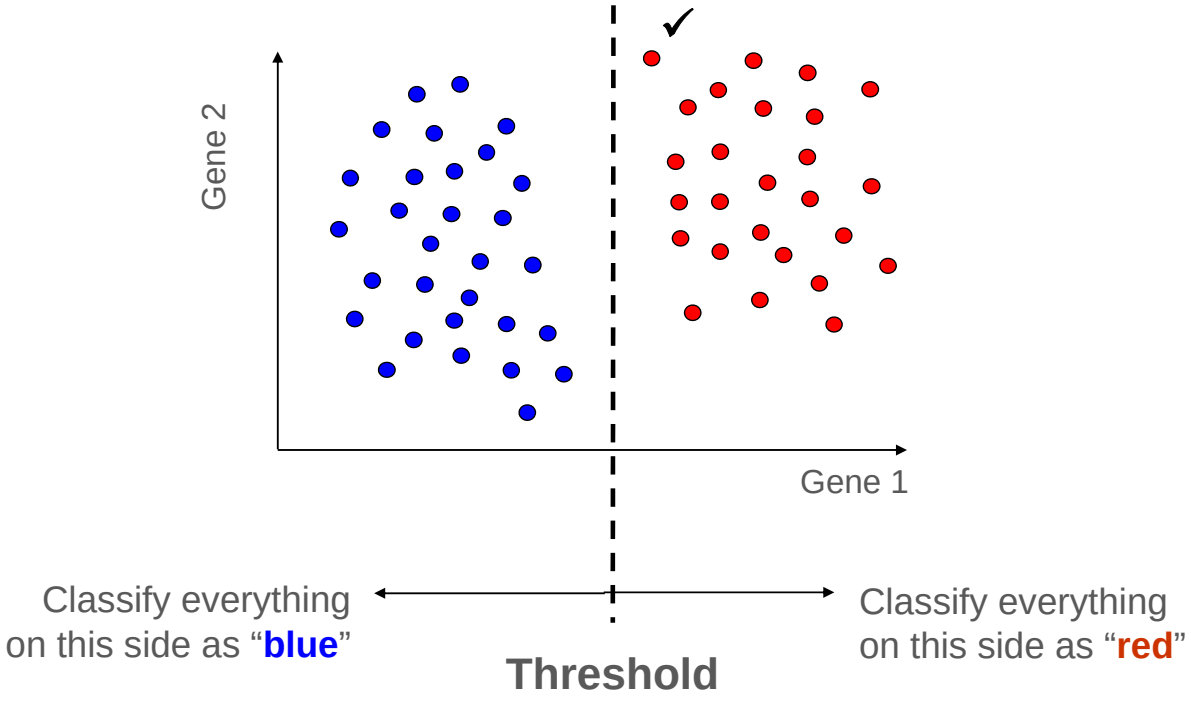
Many classification tools are based on regression models with a suitable threshold:



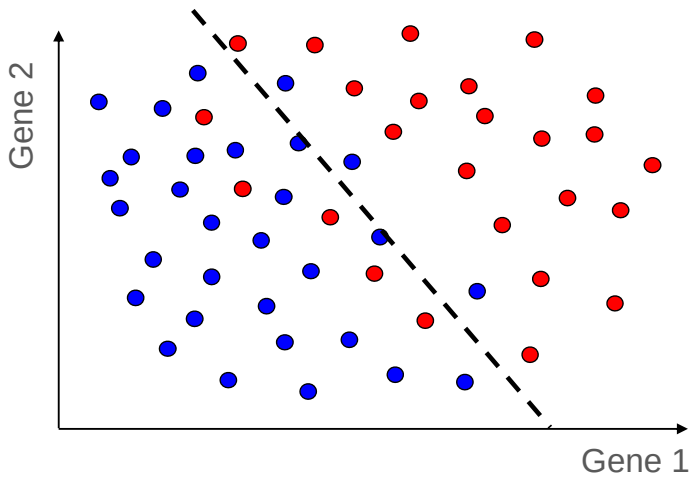
Class prediction: easy case



Class prediction: easy case



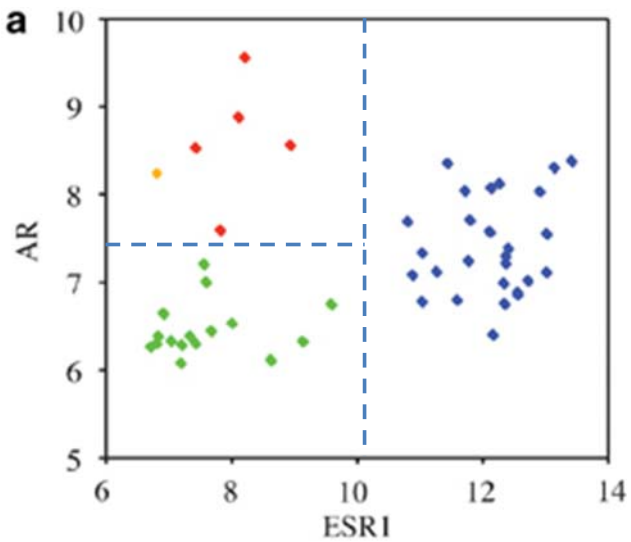
Class prediction: in practice



Class prediction: in practice

- The two groups are not perfectly separated (and the example was still a pretty good case…)
- One variable (gene) is not sufficient to assign patients to groups
- With high throughput methods, we may be talking about 10’000 measurements instead of 2

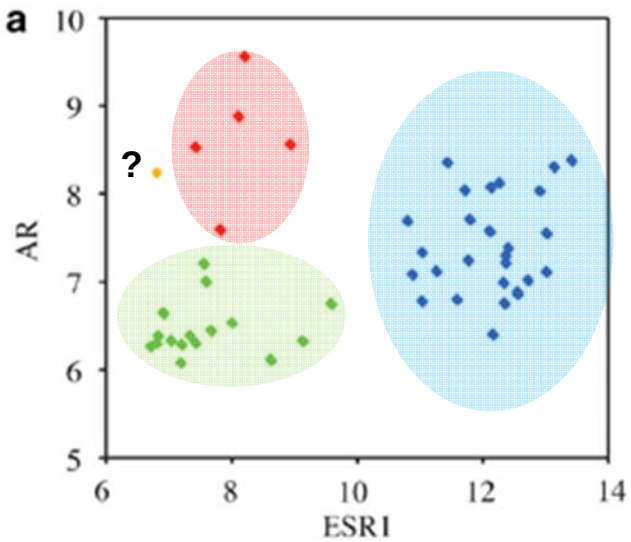
Example: classifying breast tumours



Blue points represent “oestrogen receptor (ER) status positive” determined by immunohistochemistry.

Pierre Farmer et al. **Identification of molecular apocrine breast tumours by microarray analysis.** *Oncogene* (2005) **24,** 4660–4671

Example: classifying breast tumours

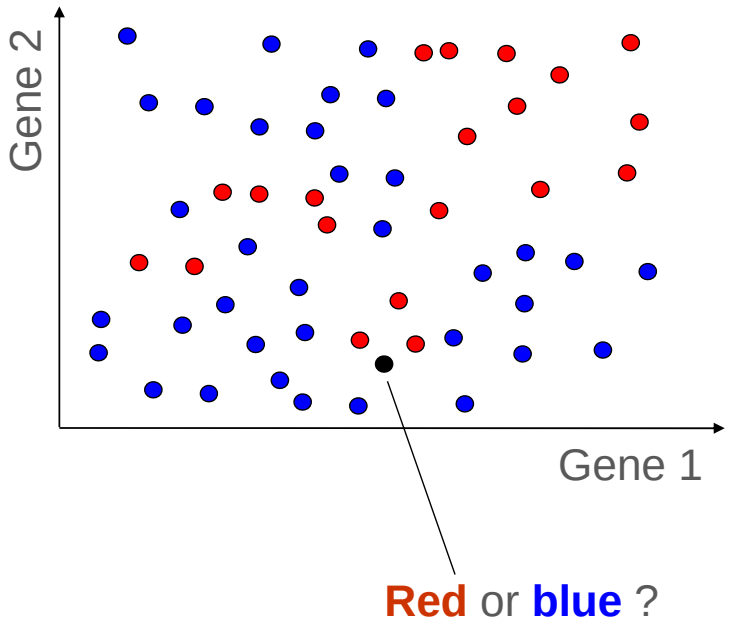


Blue points represent “oestrogen receptor (ER) status positive” determined by immunohistochemistry.

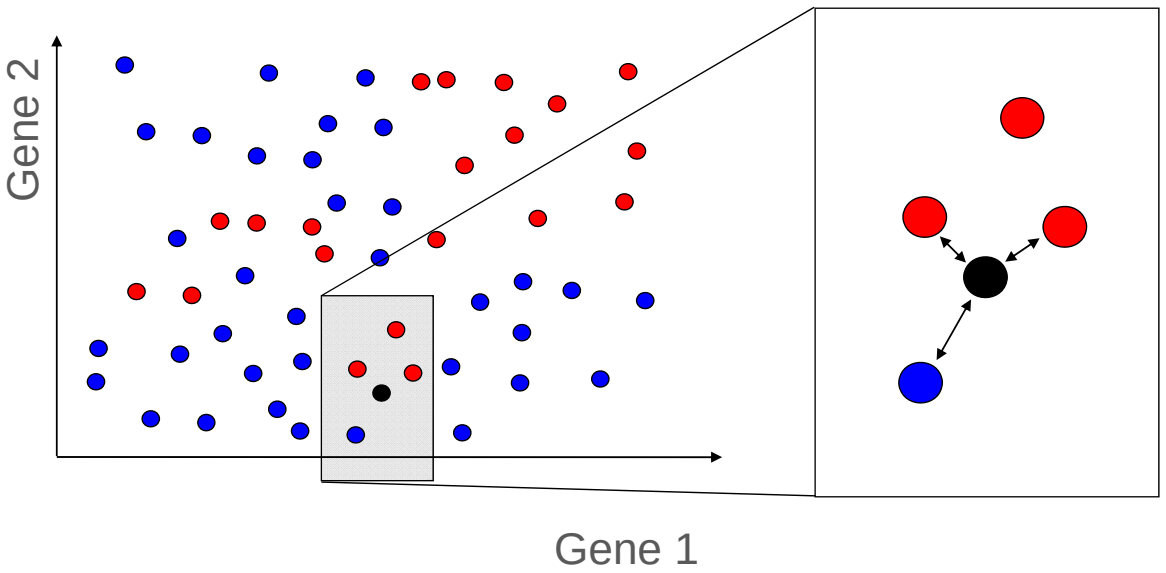
Pierre Farmer et al. **Identification of molecular apocrine breast tumours by microarray analysis.** *Oncogene* (2005) **24,** 4660–4671

The k-nearest neighbors algorithm (k-NN)

Example: 3-nearest neighbors



The 3 nearest neighbors vote



2 **red** vs 1 **blue**: the point is assigned to "**red**"

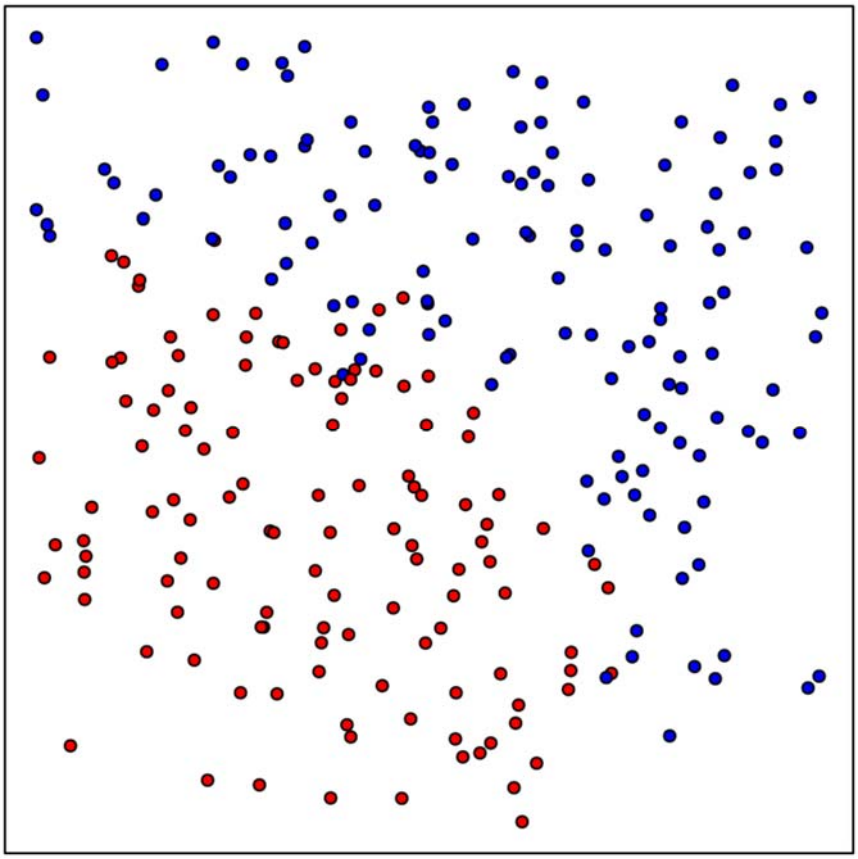
k-nearest neighbors algorithm (k-NN)

1. Choose a value for k
2. Find the k observations in the learning set that are closest to the new observation
3. Predict the class by a majority vote

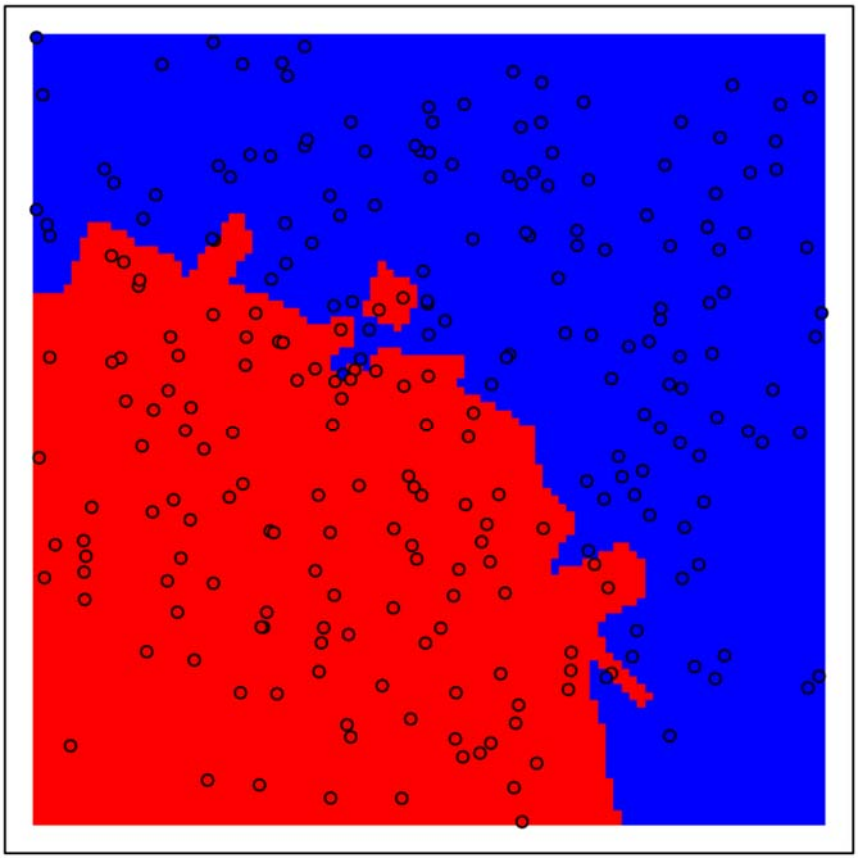
k-nearest neighbors (k-NN)

- Typical values for k: 3 or 5
- Usually determined from the learning data (value that produces the "best" result)
- Very simple method, with surprisingly good performance
- Also usable for regression (average values instead of voting)

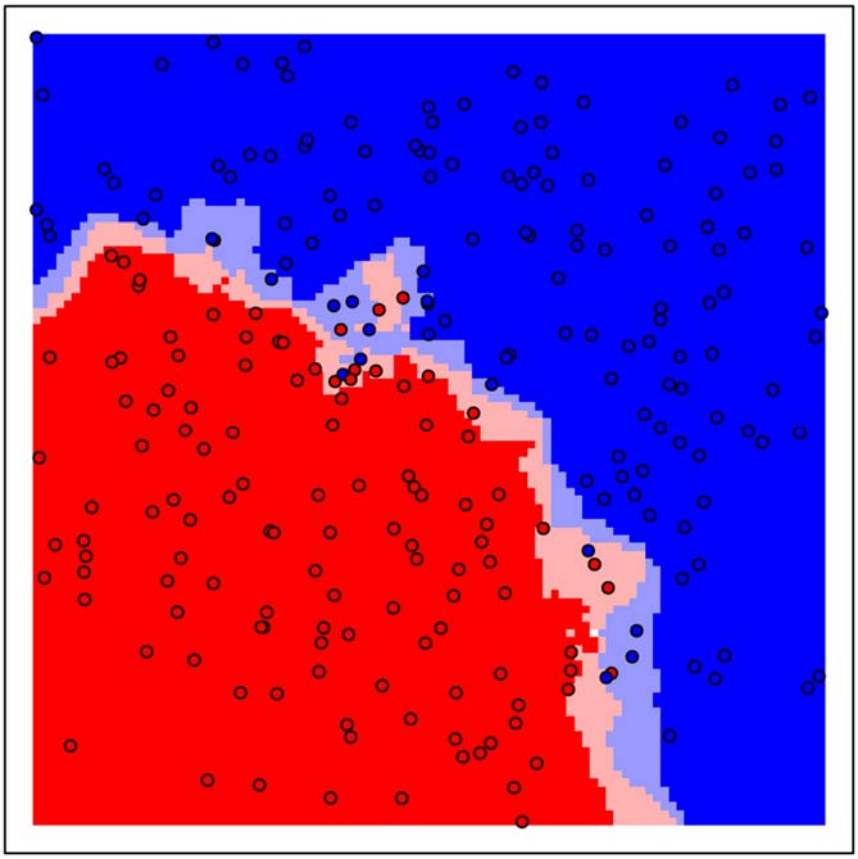
Example



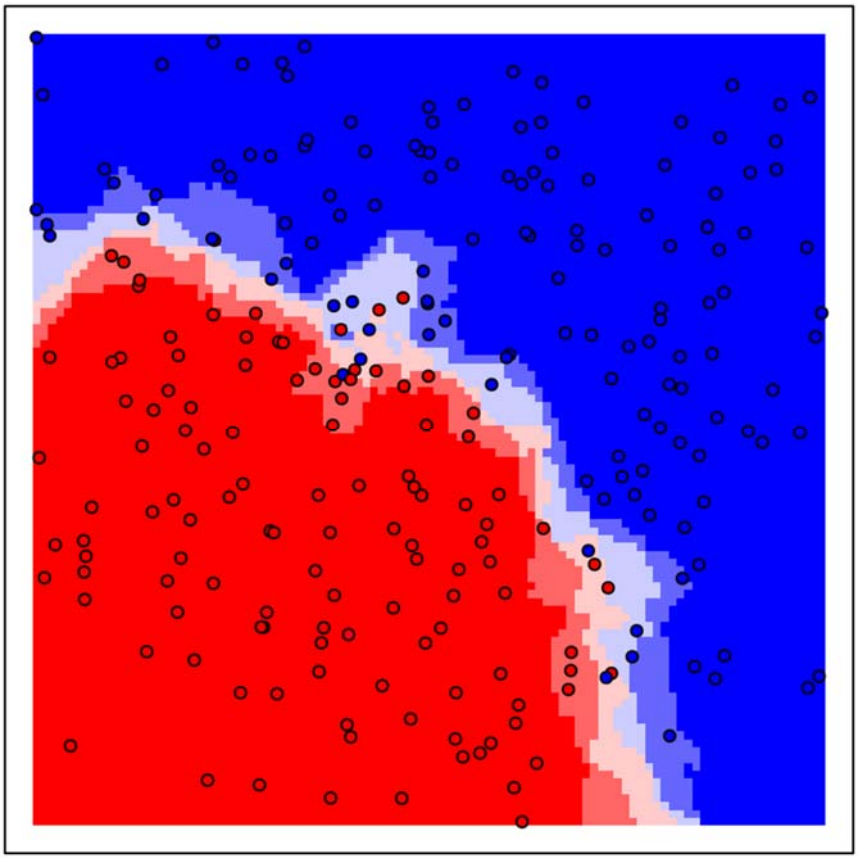
1-NN



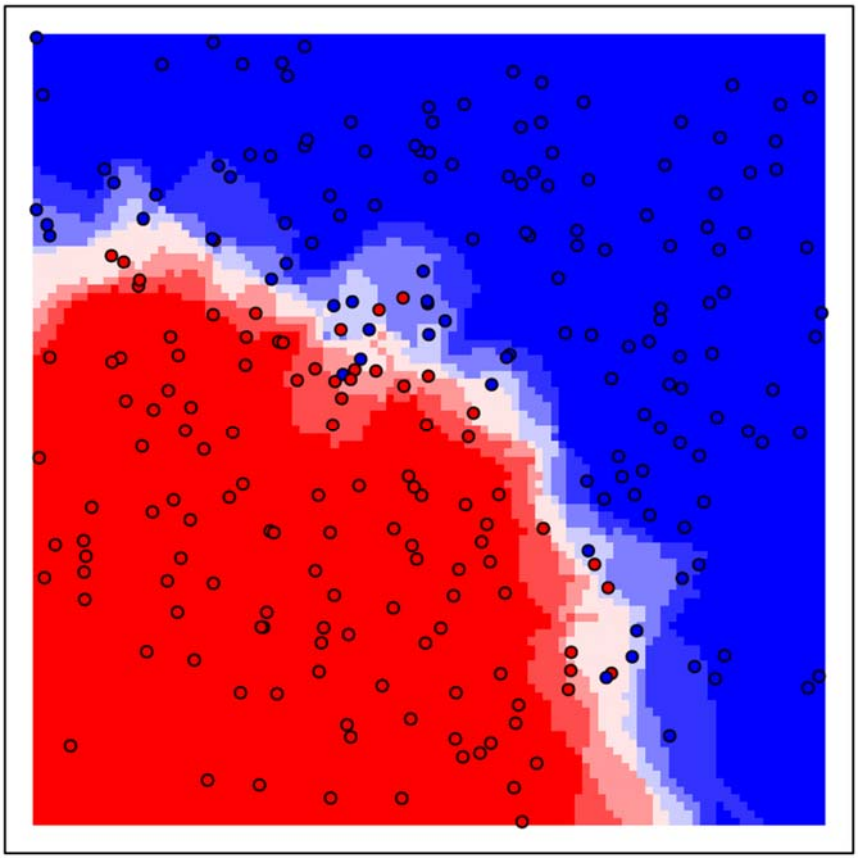
3-NN



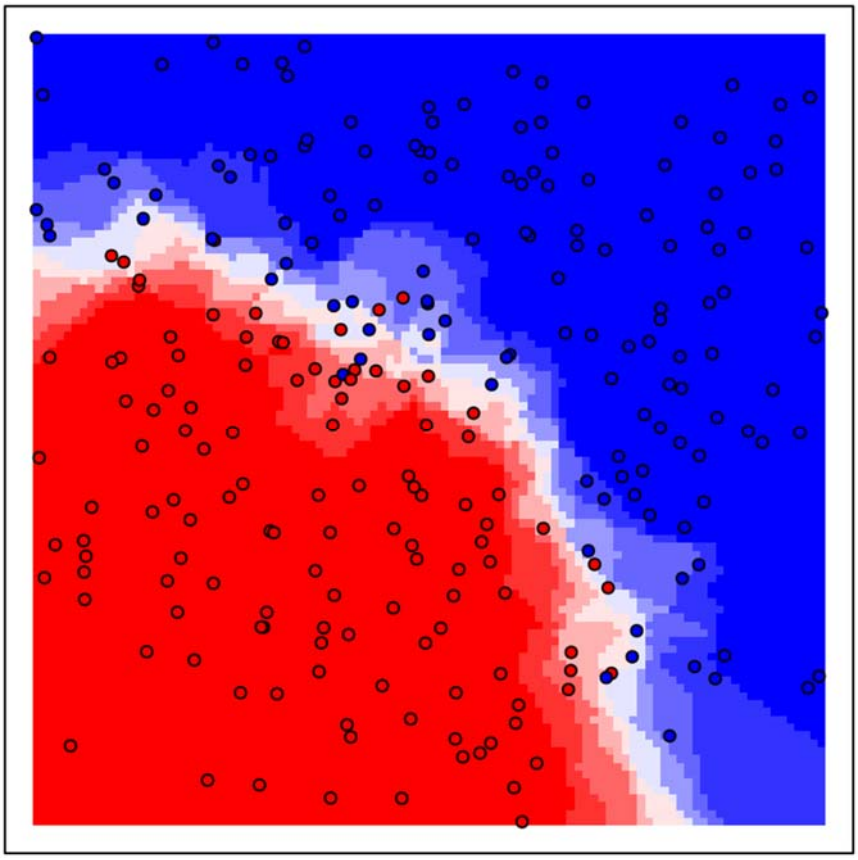
5-NN



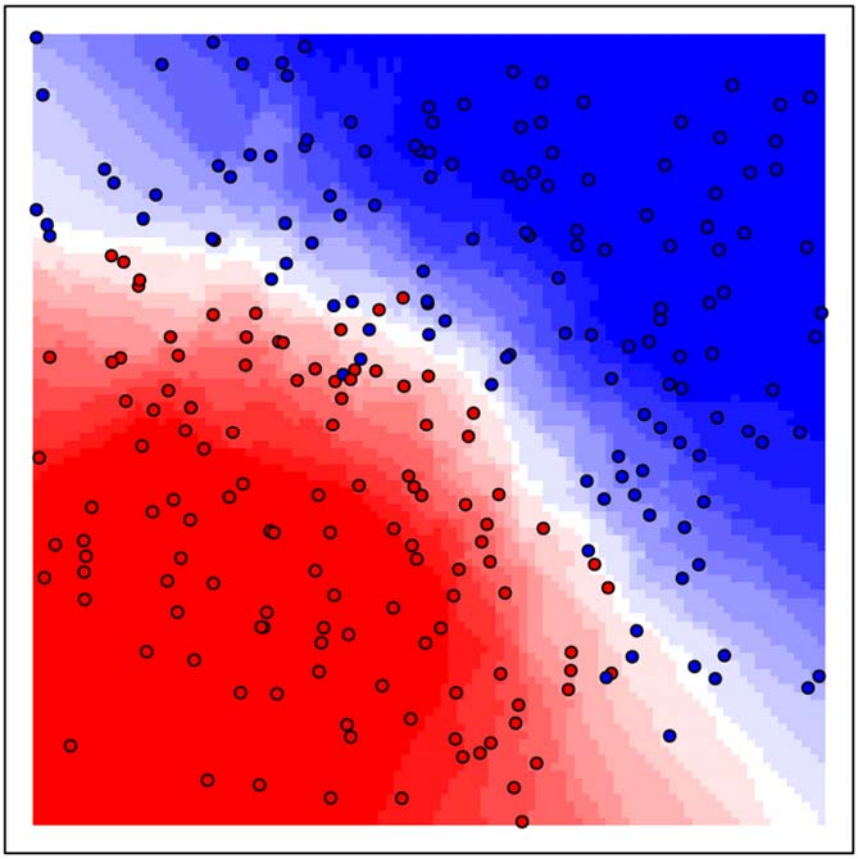
7-NN



9-NN



51-NN

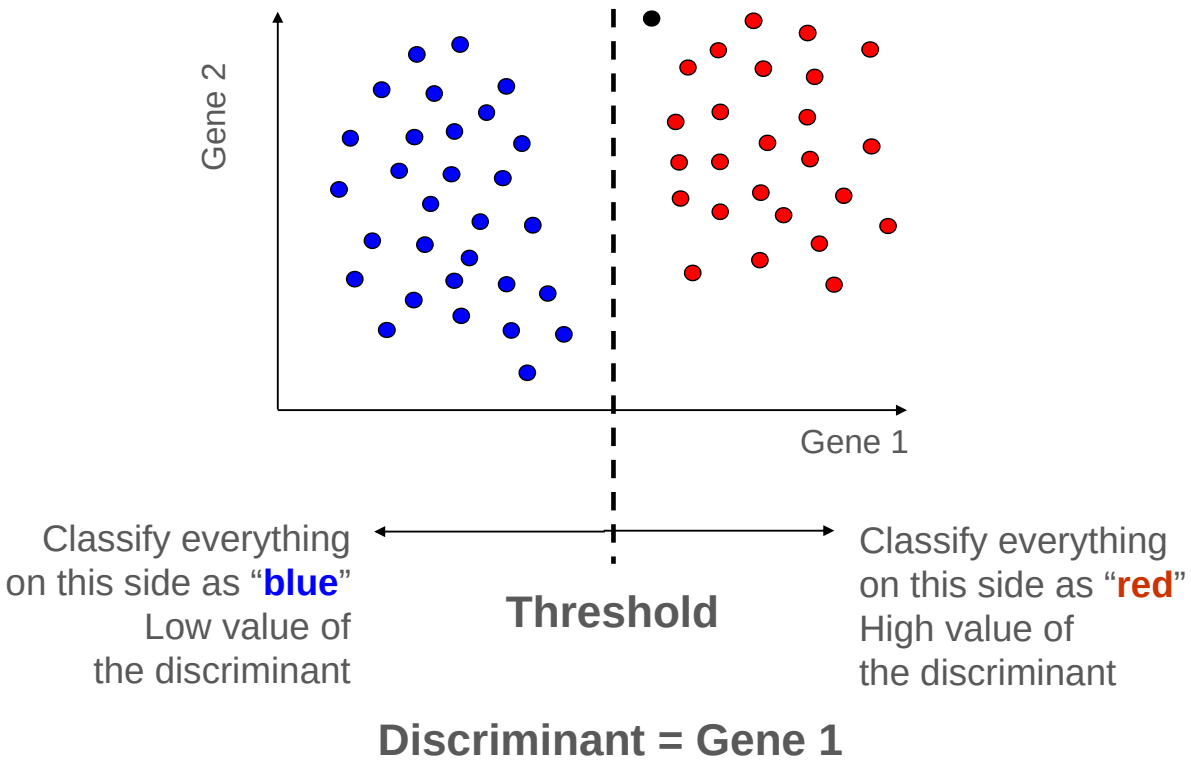


Linear discriminant analysis

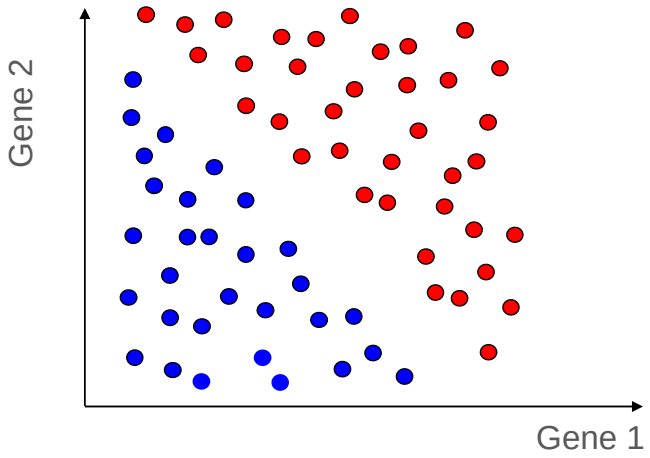
Linear Discriminant Analysis

- Suggested by R.A. Fisher in 1935
- Procedure to find a **linear combination** of the observed variables that best separates (**discriminates**) two classes of objects.
- Using the "new variable", objects from the same class are close together, and objects from a different class are further away.
- Straightforward to calculate
- Can easily be extended to more than two classes
- Similar idea to Principal Component Analysis (PCA) (unsupervised method)
- Often forgotten in favour of PCA

Back to the easy case



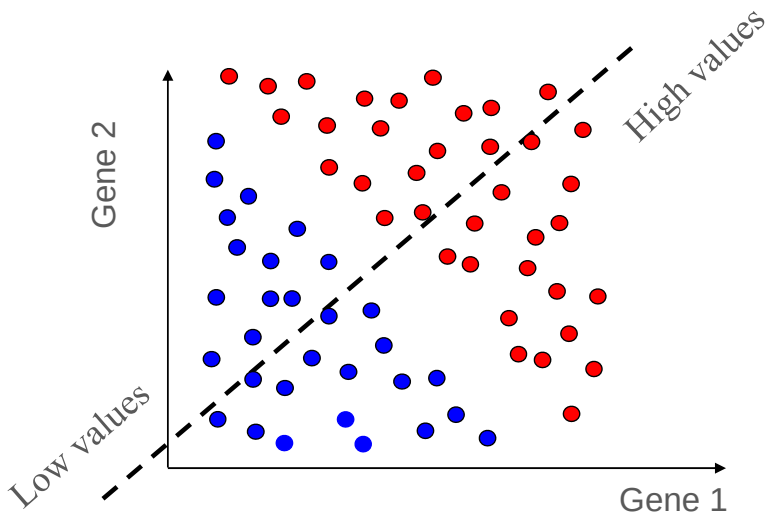
Linear Discriminant Analysis: Example



The two groups are well separated

Neither Gene1 nor Gene2 are able to discriminate between the two categories

Linear Discriminant Analysis: Example

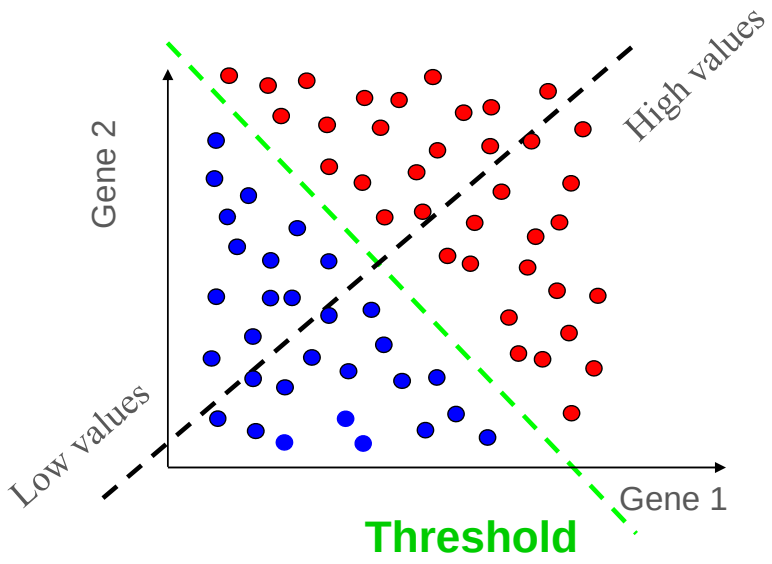


However, the linear combination discriminates well between the two groups:

L = Gene1 + Gene2

- **Blue** points tend to have smaller L values
- **Red** points tend to have bigger L values

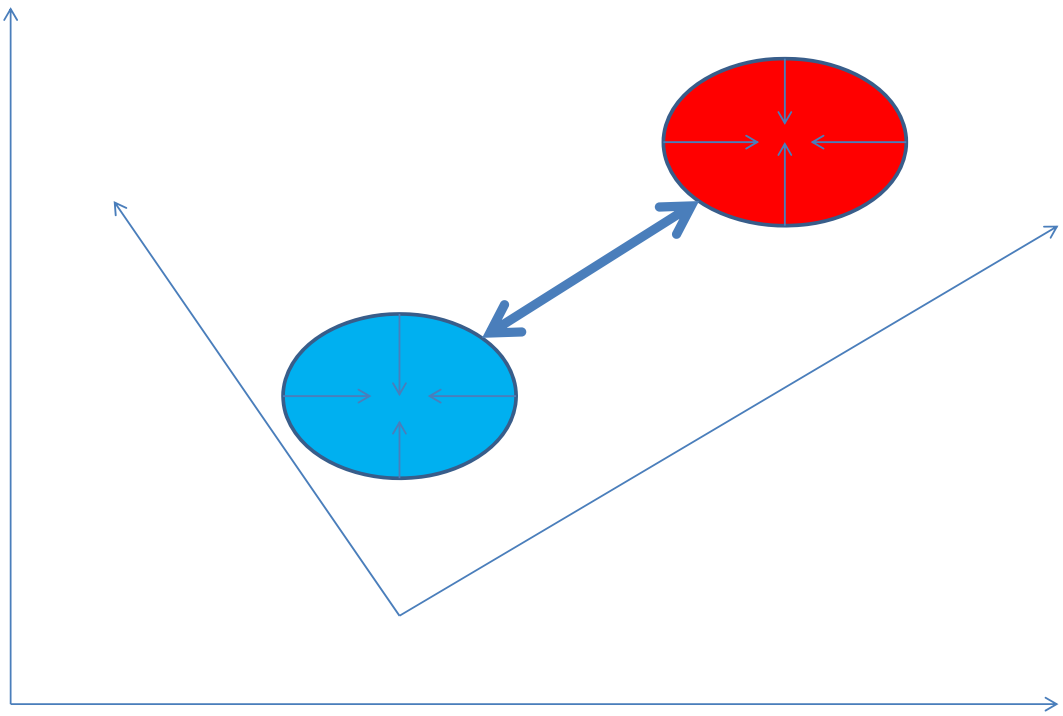
Linear Discriminant Analysis: Example



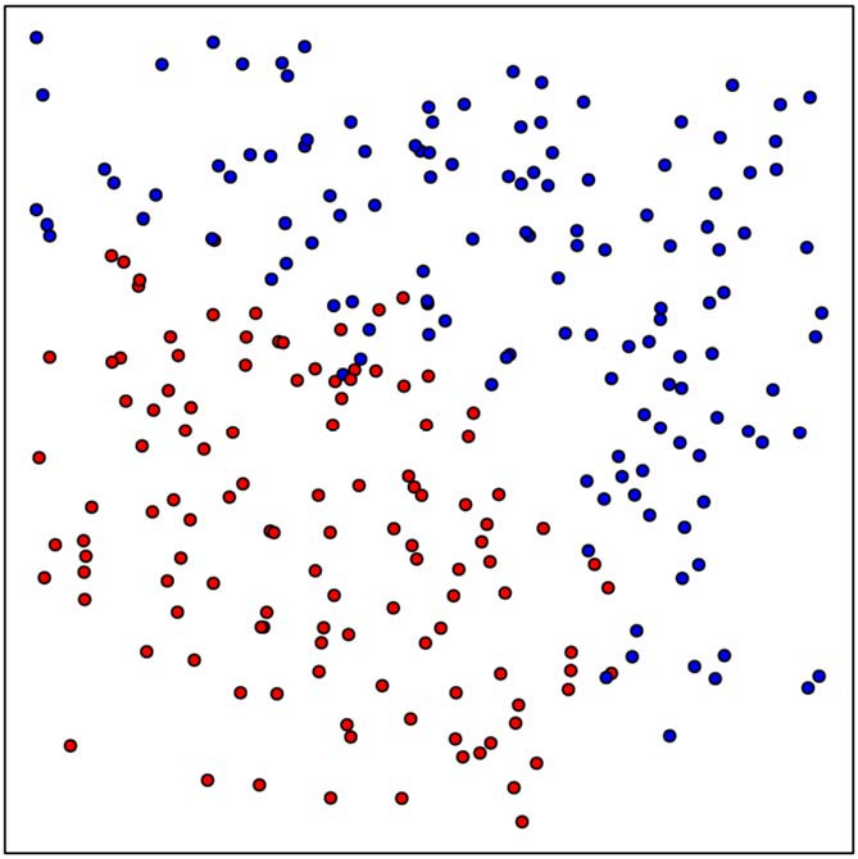
A threshold is set in between the mean of the two groups:

- Points with a value L above the threshold are classified as red
- Points with a value L below the threshold are classified as blue

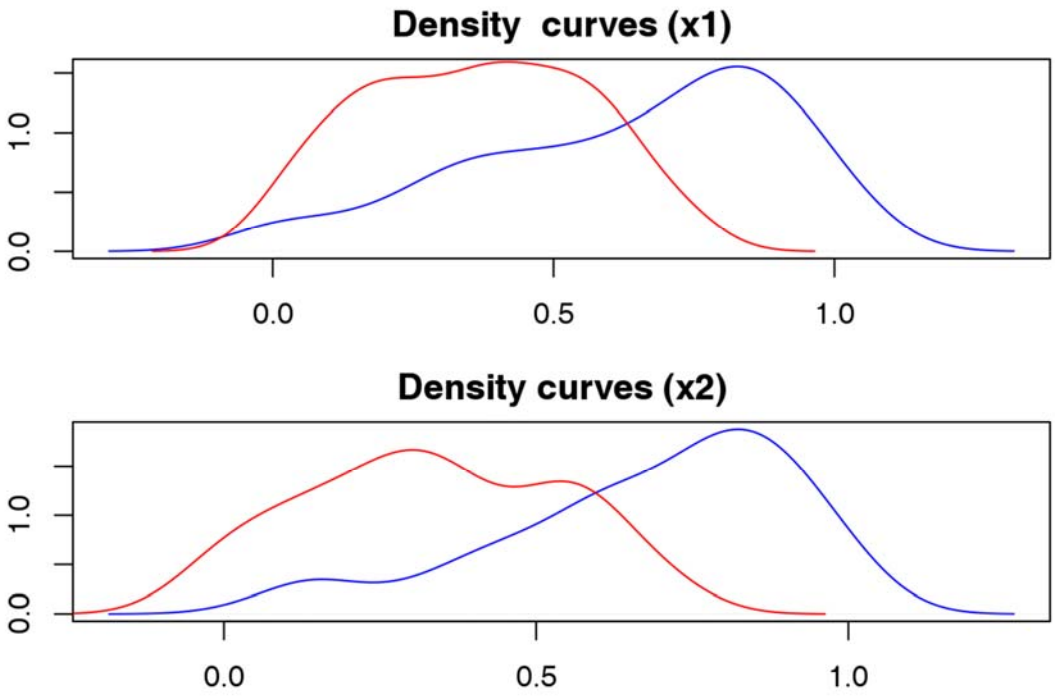
What does LDA do ?



Back to our earlier dataset



Univariate summary: density plots

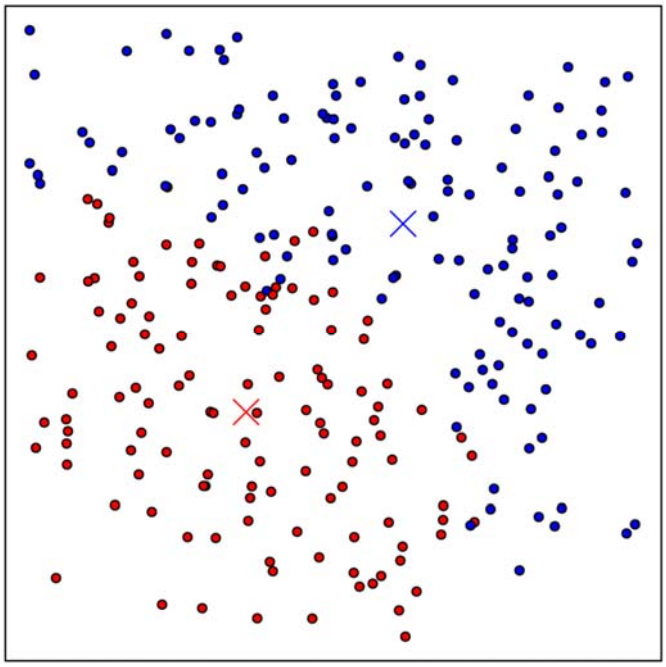


Linear Discriminant analysis using MASS

```
> library(MASS)
> class <- lda( group ~ x1 + x2)
> class
Call:
lda(group ~ x1 + x2)

Prior probabilities of groups:
    1     2
0.524 0.476

Group means:
         x1        x2
1 0.6346950 0.6808438
2 0.3628336 0.3359276

Coefficients of linear discriminants:
         LD1
x1 -3.647709
x2 -4.507556
```

Linear Discriminant analysis using MASS

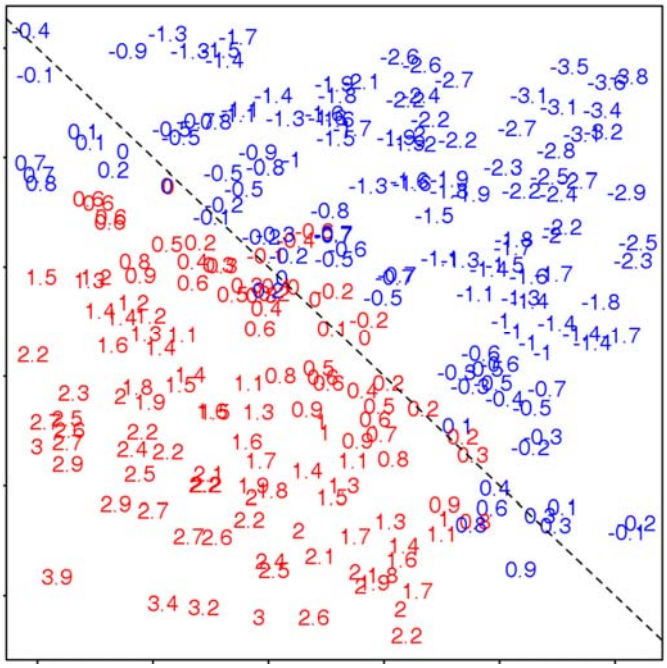

```
> library(MASS)
> class <- lda( group ~ x1 + x2)
> class
Call:
lda(group ~ x1 + x2)

Prior probabilities of groups:
    1     2
0.524 0.476

Group means:
         x1        x2
1 0.6346950 0.6808438
2 0.3628336 0.3359276
```


```
Coefficients of linear discriminants:
         LD1
x1 -3.647709
x2 -4.507556
```

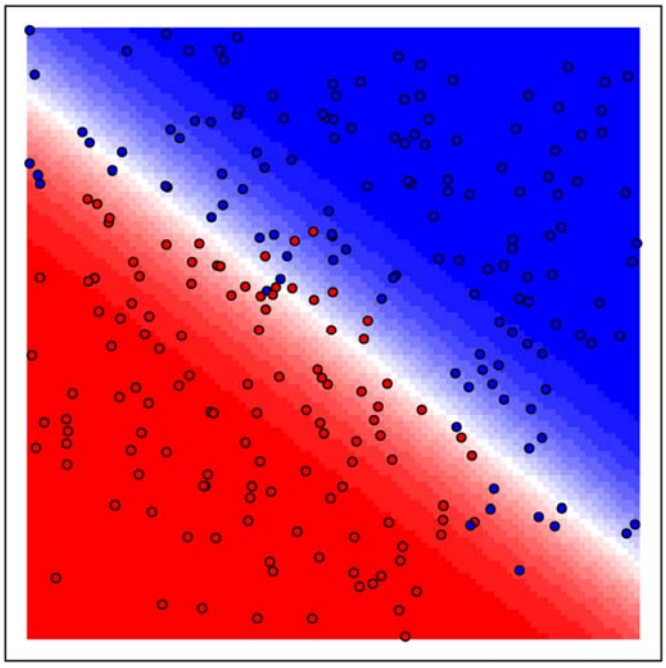
Linear Discriminant analysis using MASS

```
> library(MASS)
> class <- lda( group ~ x1 + x2)
> class
Call:
lda(group ~ x1 + x2)

Prior probabilities of groups:
    1     2
0.524 0.476

Group means:
          x1        x2
1 0.6346950 0.6808438
2 0.3628336 0.3359276

Coefficients of linear discriminants:
          LD1
x1 -3.647709
x2 -4.507556
```



Linear Discriminant analysis using MASS

```
> library(MASS)
> class <- lda( group ~ x1 + x2)
> class
Call:
lda(group ~ x1 + x2)

Prior probabilities of groups:
    1     2
0.524 0.476

Group means:
          x1        x2
1 0.6346950 0.6808438
2 0.3628336 0.3359276

Coefficients of linear discriminants:
          LD1
x1 -3.647709
x2 -4.507556
```



Assessing performance

Mechanisms of disease

Use of proteomic patterns in serum to identify ovarian cancer

Emanuel F Petricoin III, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, Lance A Liotta

572 THE LANCET • Vol 359 • February 16, 2002 • www.thelancet.com

Methods Proteomic spectra were generated by mass spectroscopy (surface-enhanced laser desorption and ionisation). A preliminary "training" set of spectra derived from analysis of serum from 50 unaffected women and 50 patients with ovarian cancer were analysed by an iterative searching algorithm that identified a proteomic pattern that completely discriminated cancer from non-cancer. The discovered pattern was then used to classify an independent set of 116 masked serum samples: 50 from women with ovarian cancer, and 66 from unaffected women or those with non-malignant disorders.

Findings The algorithm identified a cluster pattern that, in the training set, completely segregated cancer from non-cancer. The discriminatory pattern correctly identified all 50 ovarian cancer cases in the masked set, including all 18 stage I cases. Of the 66 cases of non-malignant disease, 63 were recognised as not cancer. This result yielded a sensitivity of 100% (95% CI 93–100), specificity of 95% (87–99), and positive predictive value of 94% (84–99).

Mechanisms of disease

Use of proteomic patterns in serum to identify ovarian cancer

Emanuel F Petricoin III, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, Lance A Liotta

572 THE LANCET • Vol 359 • February 16, 2002 • www.thelancet.com

Methods Proteomic spectra were generated by mass spectroscopy (surface-enhanced laser desorption and ionisation). A preliminary "training" set of spectra derived from analysis of serum from 50 unaffected women and 50 patients with ovarian cancer were analysed by an iterative searching algorithm that identified a proteomic pattern that completely discriminated cancer from non-cancer. The discovered pattern was then used to classify an independent set of 116 masked serum samples: 50 from women with ovarian cancer, and 66 from unaffected women or those with non-malignant disorders.

Findings The algorithm identified a cluster pattern that, in the training set, completely segregated cancer from non-cancer. The discriminatory pattern correctly identified all 50 ovarian cancer cases in the masked set, including all 18 stage I cases. Of the 66 cases of non-malignant disease, 63 were recognised as not cancer. This result yielded a sensitivity of 100% (95% CI 93–100), specificity of 95% (87–99), and positive predictive value of 94% (84–99).

Prediction

Truth

	Cancer	No	Total
Cancer	50	0	50
No cancer	3	63	66
Total	53	63	116

Results

		Predicted class		
		Cancer	Healthy	Total
True class	Cancer	50	0	50
	Healthy	3	63	66
	Total	53	63	116

Confusion matrix

		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

Confusion matrix

		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

Confusion matrix: what we want to optimize

		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

Total number of errors: 3 + 0 = **3**

MECHANISMS OF DISEASE

Mechanisms of disease

Use of proteomic patterns in serum to identify ovarian cancer

Emanuel F Petricoin III, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, Lance A Liotta

572 THE LANCET • Vol 359 • February 16, 2002 • www.thelancet.com

Methods Proteomic spectra were generated by mass spectroscopy (surface-enhanced laser desorption and ionisation). A preliminary "training" set of spectra derived from analysis of serum from 50 unaffected women and 50 patients with ovarian cancer were analysed by an iterative searching algorithm that identified a proteomic pattern that completely discriminated cancer from non-cancer. The discovered pattern was then used to classify an independent set of 116 masked serum samples: 50 from women with ovarian cancer, and 66 from unaffected women or those with non-malignant disorders.

Findings The algorithm identified a cluster pattern that, in the training set, completely segregated cancer from non-cancer. The discriminatory pattern correctly identified all 50 ovarian cancer cases in the masked set, including all 18 stage I cases. Of the 66 cases of non-malignant disease, 63 were recognised as not cancer. This result yielded a sensitivity of 100% (95% CI 93–100), specificity of 95% (87–99), and positive predictive value of 94% (84–99).

Prediction

Truth	Cancer	No	Total
Cancer	50	0	50
No cancer	3	63	66
Total	53	63	116

Different types of errors

True positive

False negative

		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

False positive

True negative

Different types of errors: true and false positive rates

TP

FN

		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

FP

TN

True positive rate: (**sensitivity**)

$$\frac{TP}{TP + FN}$$

False positive rate: (**1 – specificity**)

$$\frac{FP}{FP + TN}$$

Different types of errors: true and false positive rates

		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

TP FN FP TN

True positive rate: (**sensitivity**) $\frac{TP}{TP + FN} = 100\%$

False positive rate: (**1 – specificity**) $\frac{FP}{FP + TN} = 4.6\%$

Different types of errors: PPV and NPV

		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

TP FN FP TN

Positive predictive value: (**PPV, precision**) $\frac{TP}{TP + FP}$

Negative predictive value: (**NPV**) $\frac{TN}{TN + FN}$

Different types of errors: PPV and NPV

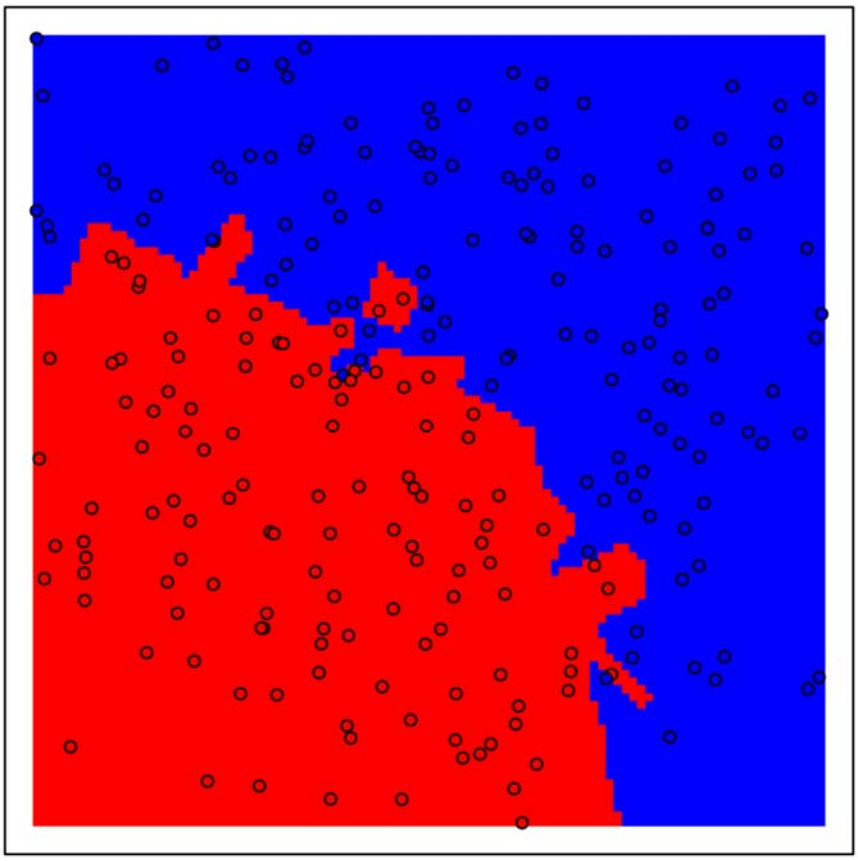
		Predicted class	
		Cancer	Healthy
True class	Cancer	50	0
	Healthy	3	63

TP FN FP TN

Positive predictive value: $\frac{TP}{TP + FP} = 94\%$
(**PPV, precision**)

Negative predictive value: $\frac{TN}{TN + FN} = 100\%$
(**NPV**)

Back to the 1-NN classifier



Results: 1-NN

		Predicted class	
		Blue	Red
True class	Blue	131	0
	Red	0	119

Results: 3-NN and 5-NN

		Predicted class	
		Blue	Red
True class	Blue	127	4
	Red	4	115

Results: 7-NN

		Predicted class	
		Blue	Red
True class	Blue	127	6
	Red	4	113

Results: 9-NN

		Predicted class	
		Blue	Red
True class	Blue	127	9
	Red	4	110

Results: 51-NN

		Predicted class	
		Blue	Red
True class	Blue	121	3
	Red	10	116

Some rules for assessment

Each observation can either be used for fitting the model or assessing it (but not both !)

You can use an observation as many times as you like for exploration/learning, but you can only use it once for confirmation. If you use it more than once, you are learning again (and not assessing).

To assess a model, you **must** use data independent of the data you used to train the model – otherwise you will be over-optimistic.

Inspired from "R for Data Science", Garrett Grolemund and Hadley Wickham.
http://r4ds.had.co.nz/model-intro.html

Ideally: a independent dataset

More realistically: randomly split your data in two pieces **before you begin using it**:

50% will be used to train the model (learning set or training set)

50% will be used to test the model (testing set)

An even better approach (suggested by H. Wickham)

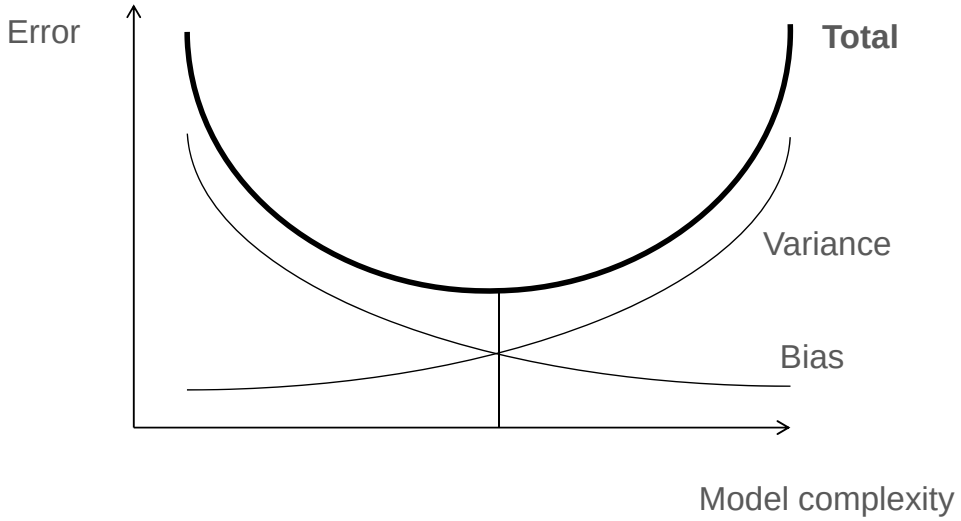
Split your data into three pieces before you begin the analysis:

- 60% of your data goes into a training set. You're allowed to do anything you like with this data.
- 20% goes into a query set. You can use this data to compare models by hand, but you're not allowed to use it as part of an automated process.
- 20% is held back for a test set. You can only use this data ONCE, to test your final model.

An even better approach (suggested by H. Wickham)

This partitioning allows you to explore the training data, occasionally generating candidate hypotheses that you check with the query set. When you are confident you have the right model, you can check it once with the test data.

Bias-variance trade-off



Bias: model misses important features of the underlying model (underfitting)
Variance: model is sensitive to noise in the data (overfitting)

Mechanisms of disease

Use of proteomic patterns in serum to identify ovarian cancer

Emanuel F Petricoin III, Ali M Ardekani, Ben A Hitt, Peter J Levine, Vincent A Fusaro, Seth M Steinberg, Gordon B Mills, Charles Simone, David A Fishman, Elise C Kohn, Lance A Liotta

572 THE LANCET • Vol 359 • February 16, 2002 • www.thelancet.com

Methods Proteomic spectra were generated by mass spectroscopy (surface-enhanced laser desorption and ionisation). A preliminary "training" set of spectra derived from analysis of serum from 50 unaffected women and 50 patients with ovarian cancer were analysed by an iterative searching algorithm that identified a proteomic pattern that completely discriminated cancer from non-cancer. The discovered pattern was then used to classify an independent set of 116 masked serum samples: 50 from women with ovarian cancer, and 66 from unaffected women or those with non-malignant disorders.

Findings The algorithm identified a cluster pattern that, in the training set, completely segregated cancer from non-cancer. The discriminatory pattern correctly identified all 50 ovarian cancer cases in the masked set, including all 18 stage I cases. Of the 66 cases of non-malignant disease, 63 were recognised as not cancer. This result yielded a sensitivity of 100% (95% CI 93–100), specificity of 95% (87–99), and positive predictive value of 94% (84–99).

Prediction

Truth	Cancer	No	Total
Cancer	50	0	50
No cancer	3	63	66
Total	53	63	116

BIOINFORMATICS

Vol. 20 no. 5 2004, pages 777–785
DOI: 10.1093/bioinformatics/btg484

Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments

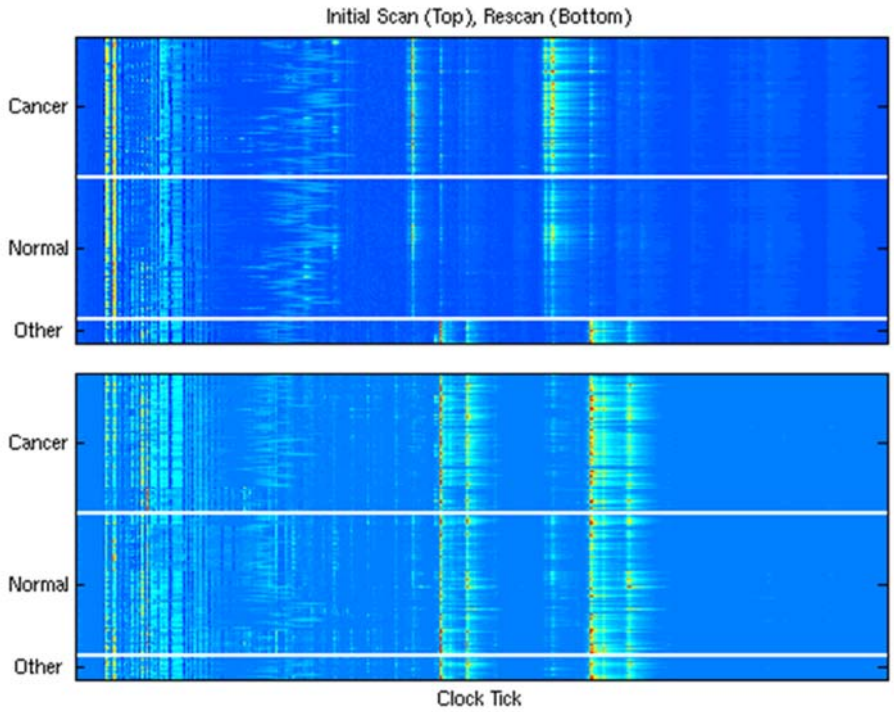
Keith A. Baggerly*, Jeffrey S. Morris and Kevin R. Coombes

ABSTRACT

Motivation: There has been much interest in using patterns derived from surface-enhanced laser desorption and ionization (SELDI) protein mass spectra from serum to differentiate samples from patients both with and without disease. Such patterns have been used without identification of the underlying proteins responsible. However, there are questions as to the stability of this procedure over multiple experiments.

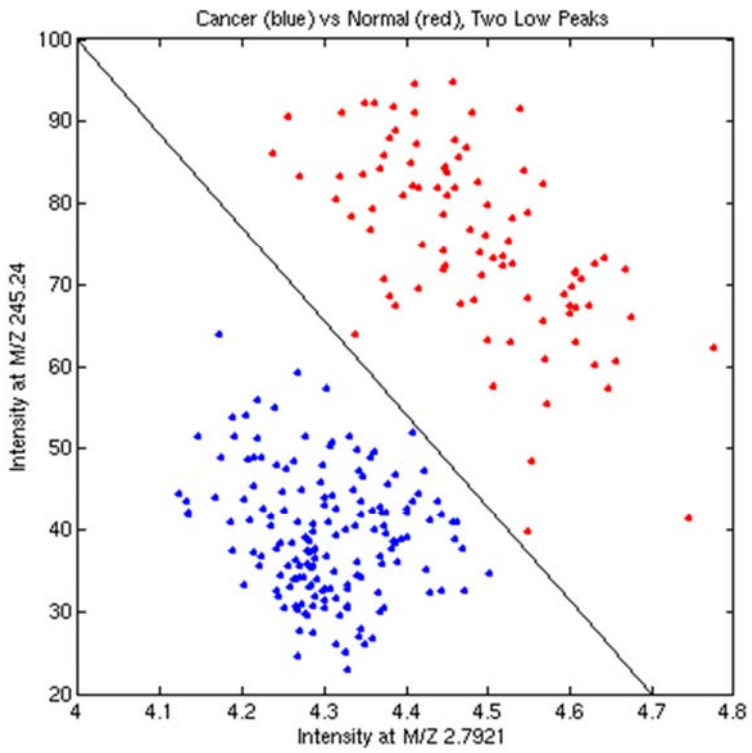
Results: We compared SELDI proteomic spectra from serum from three experiments by the same group on separating ovarian cancer from normal tissue. These spectra are available on the web at http://clinicalproteomics.steem.com. In general, the results were not reproducible across experiments. Baseline correction prevents reproduction of the results for two of the experiments. In one experiment, there is evidence of a major shift in protocol mid-experiment which could bias the results. In another, structure in the noise regions of the spectra allows us to distinguish normal from cancer, suggesting that the normals and cancers were processed differently. Sets of features found to discriminate well in one experiment do not generalize to other experiments. Finally, the mass calibration in all three experiments appears suspect. Taken together, these and other concerns suggest that much of the structure uncovered in these experiments could be due to artifacts of sample processing, not to the underlying biology of cancer. We provide some guidelines for design and analysis in experiments like these to ensure better reproducible, biologically meaningfully results.

Processing trumps biology

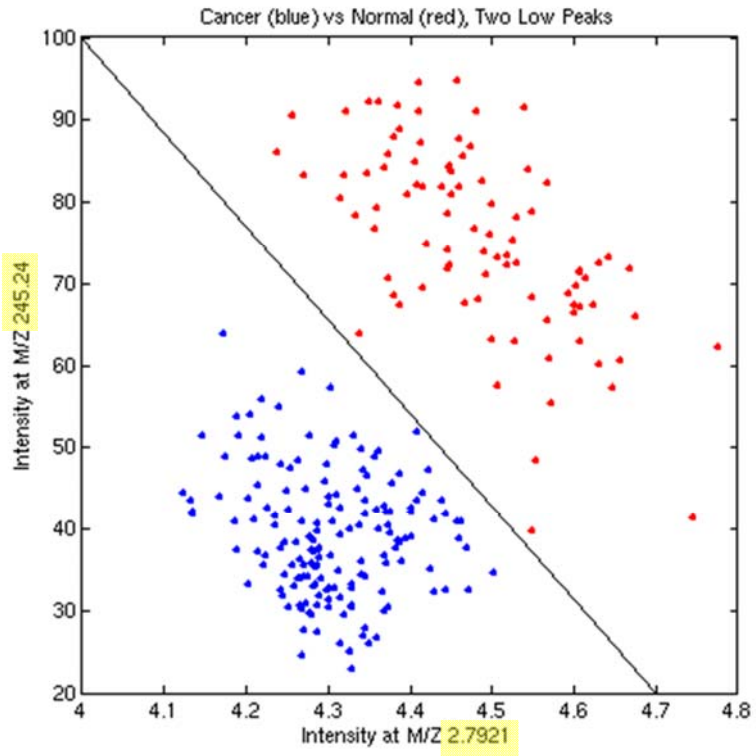


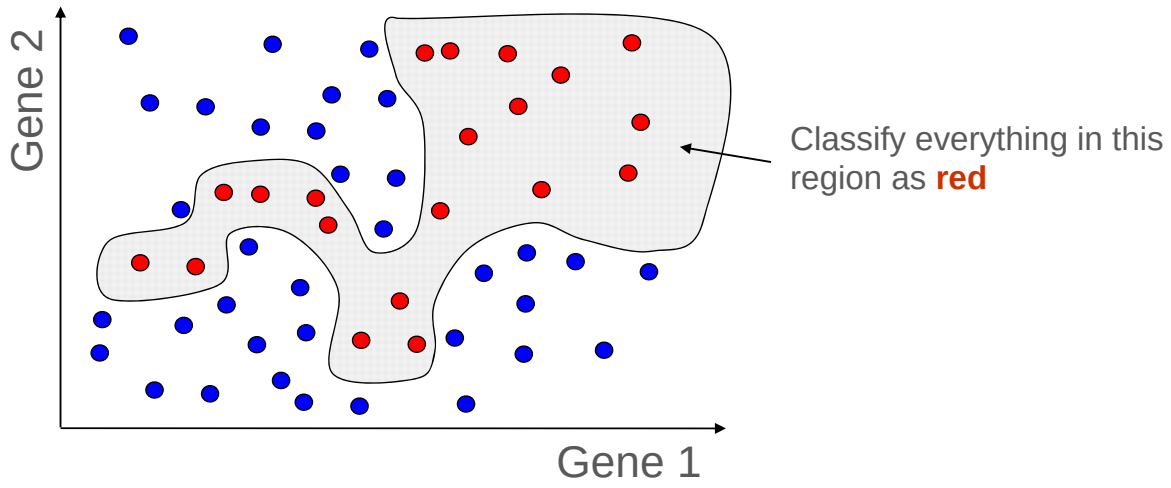
Slide courtesy of Keith Baggerly

Two peaks that allow correct classification of all samples



*Two **NOISE** peaks that allow correct classification of all samples*





Perfect classifier for this data

But probably not so good with any new data

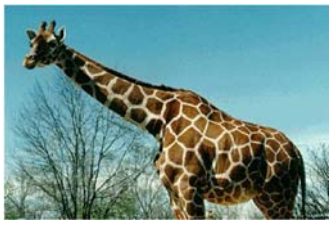
- It is easy to create classifiers which fits the training data perfectly
- It is harder to find classifiers which still works as well when validated on new data
- A classifier must ALWAYS be tested on data independent from the one used to actually train the classifier.
- This is particularly important in cases where we have
  - Few samples
  - Many different measurements
- If not careful, it is always possible to find a classifier that works well for your training data !

Discrimination

Training

Labels : A zebra ! A giraffe !

Test : A zebra or giraffe?



Predicted Label : I predict a giraffe

Discrimination

Training

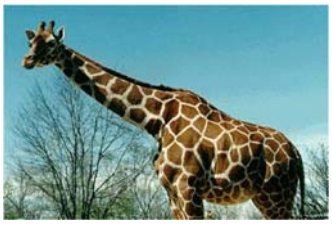
Labels: A zebra A giraffe

Discrimination: example of overfitting (and confounding factor)

Training

Labels : A zebra ! A giraffe !

Test : A zebra or giraffe?



Predicted Label : I predict a zebra !

How to avoid overfitting ?

Build your classifier using a dataset.

Use a second, **independent**, dataset to assess the performance of your classifier.

(either a really independent dataset or a training/learning split)

But if your dataset is too small to be partitioned, you have a problem…

3-fold Cross validation

Learning	Learning	Testing
Learning	Testing	Learning
Testing	Learning	Learning

Error
Error
Error

Average error

Cross-validation: «V-fold cross validation» (CV)

The learning set is divided randomly into V subsets of (nearly) equal size.

V Classifiers are built leaving each set out in turn; the test set error rate is computed on the set left out, and averaged.

Special case: «leave-one-out cross-validation»: the test set consists of only ***one*** sample.

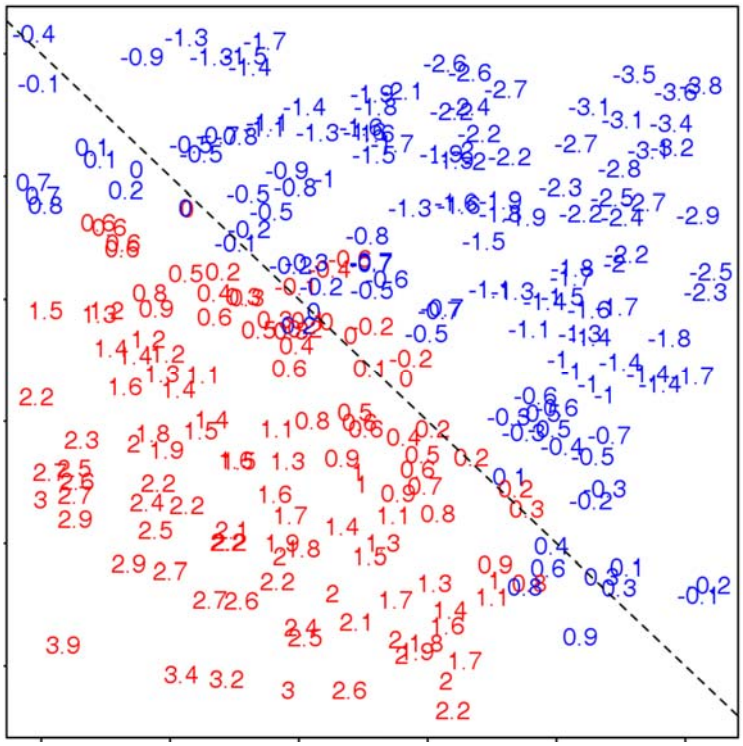
- Cross validation does not provide a single model
- Each step produces a different model

- Cross-validation allows you to assess the performance of a method for building a classifier rather than a single model

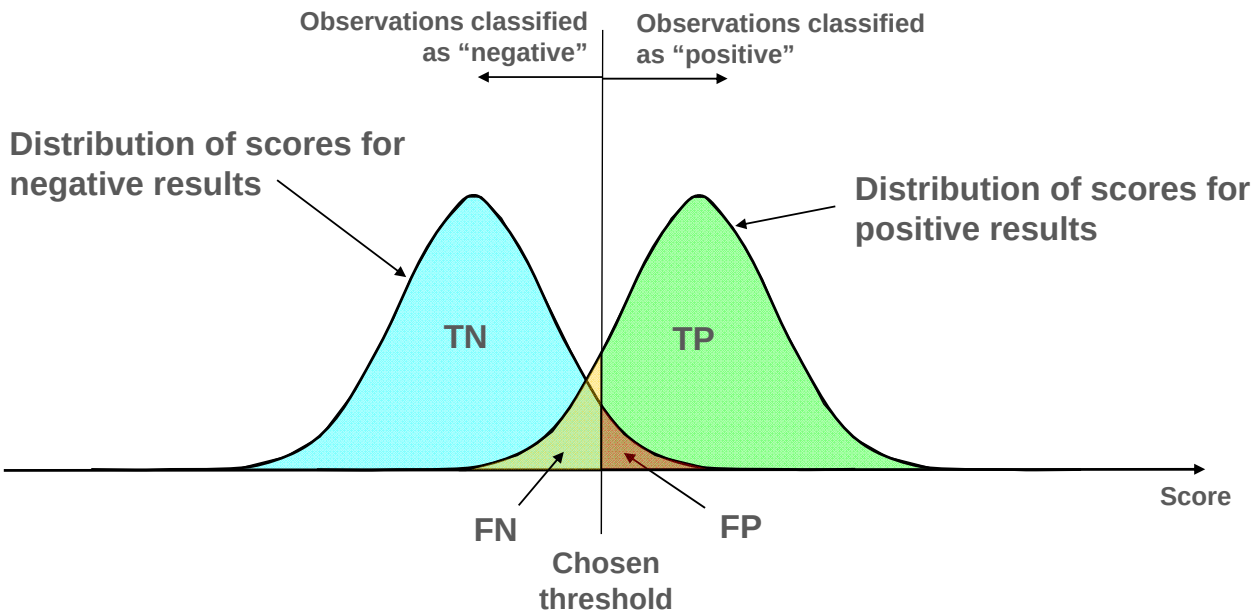
- Very useful for testing parameters
- Example: how many neighbours in the kNN algorithm ?

What does it mean when our classification depends on a continuous score ?

Back to our LDA example

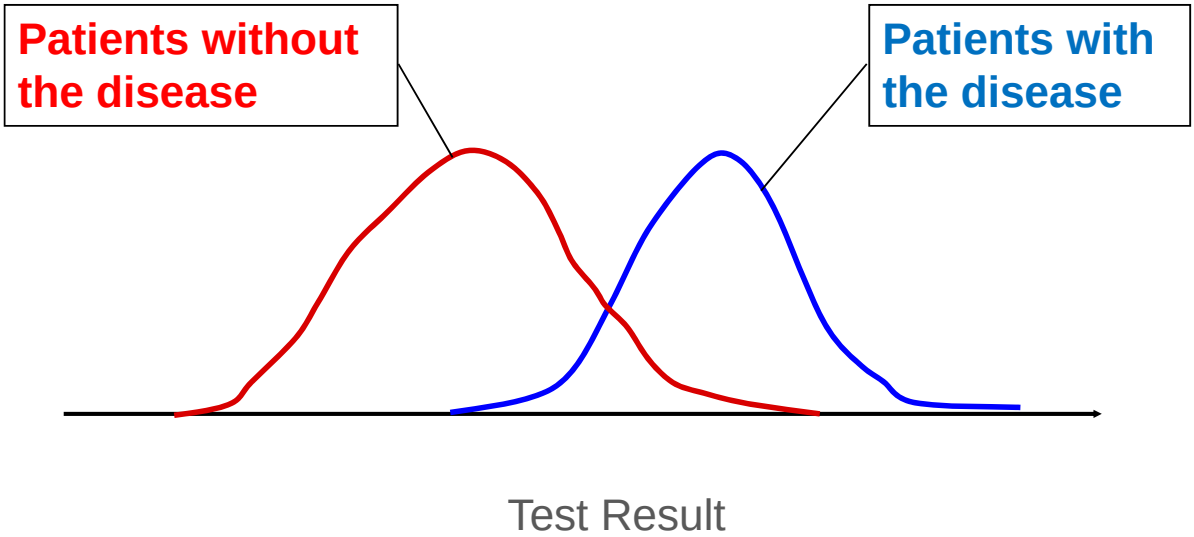


Graphical representation

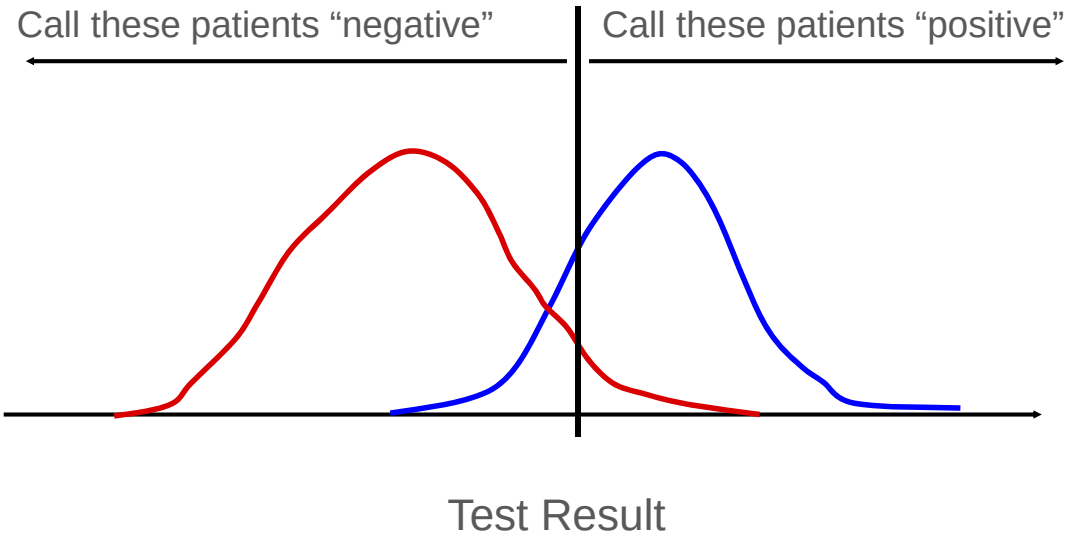


The two distributions overlap, so that it is impossible to use this score to perfectly discriminate between positive and negative results.

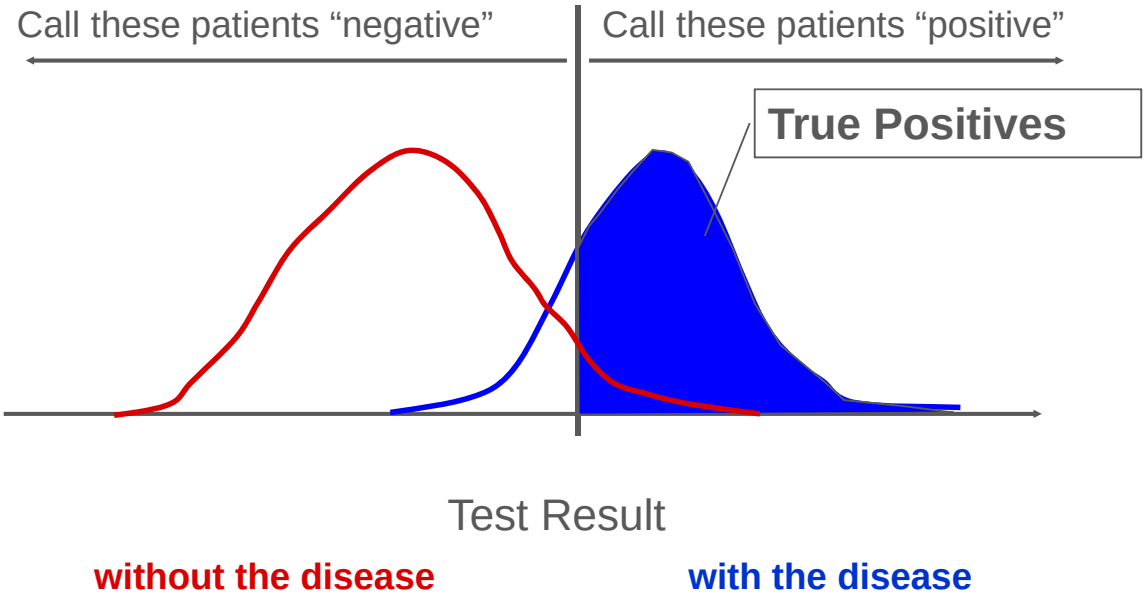
Specific Example

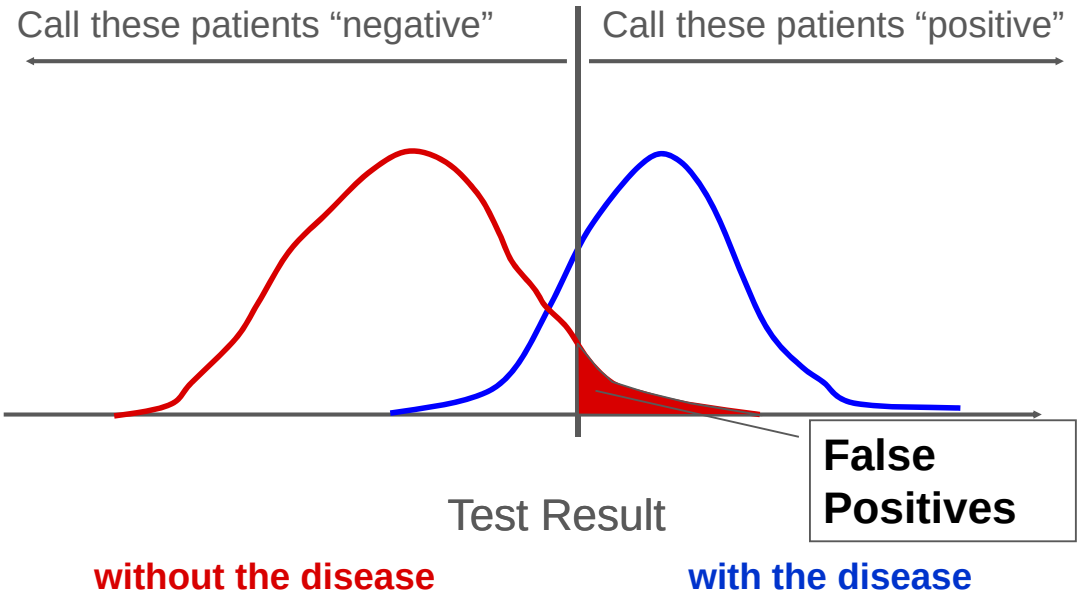


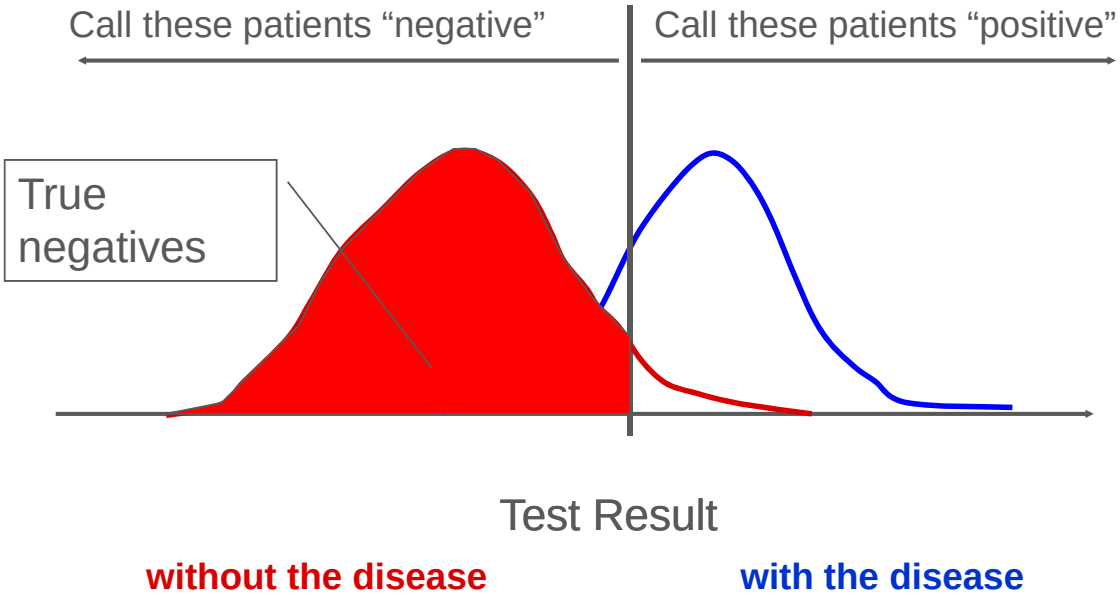
Threshold

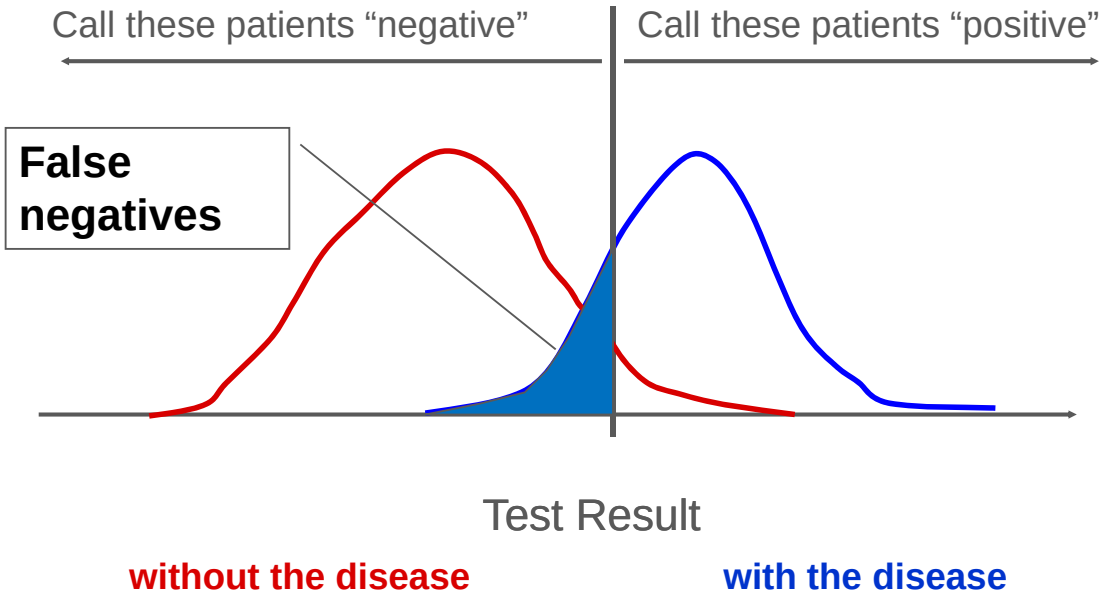


Some definitions ...

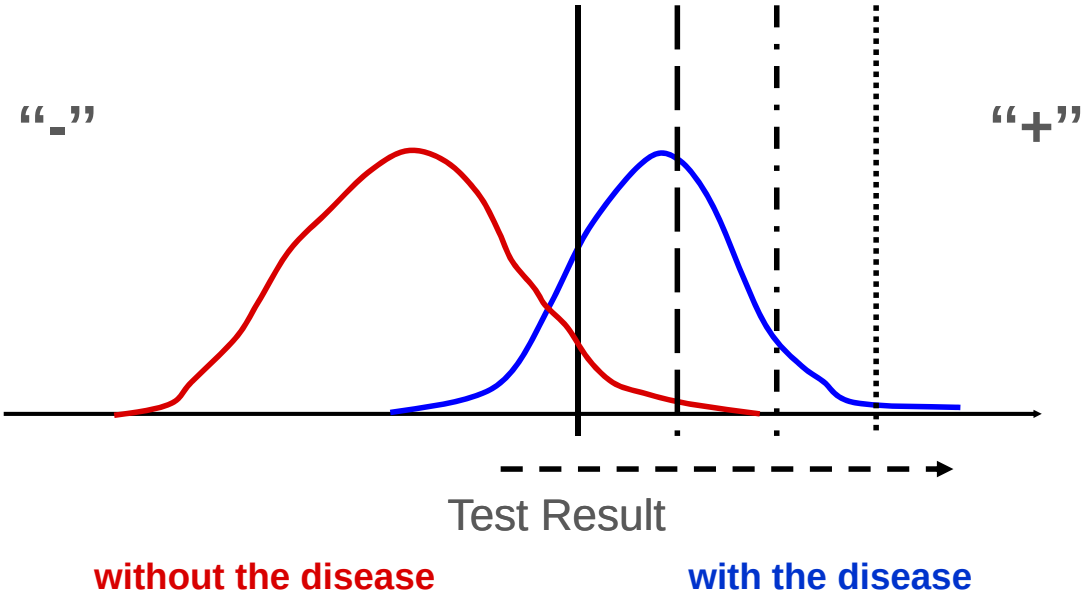




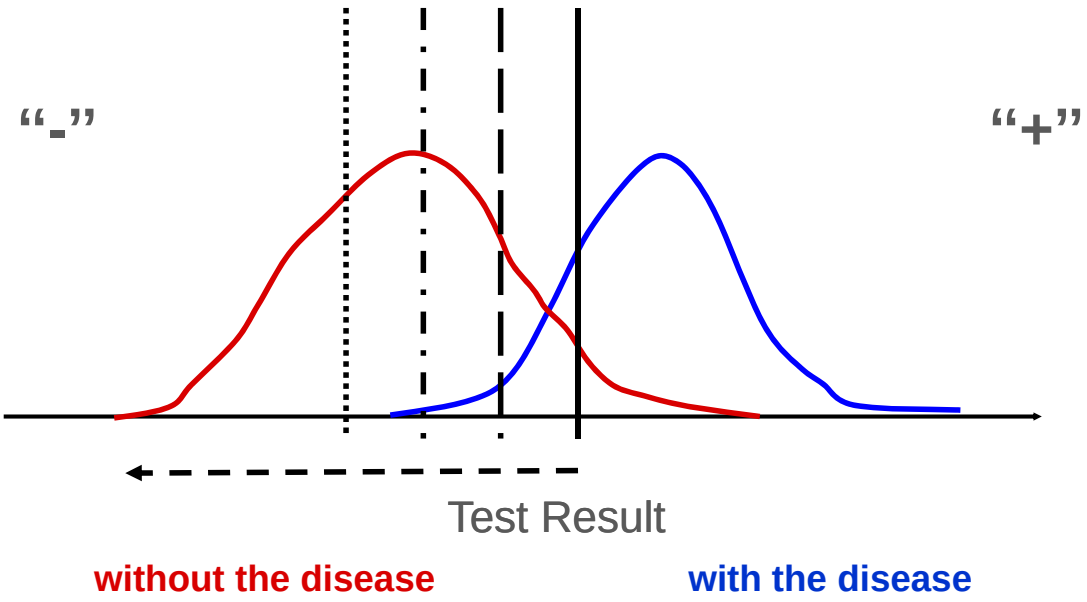

Call these patients “negative”
Call these patients “positive”
True negatives
Test Result
without the disease
with the disease


Call these patients “negative”
Call these patients “positive”
False negatives
Test Result
without the disease
with the disease

Moving the Threshold: right



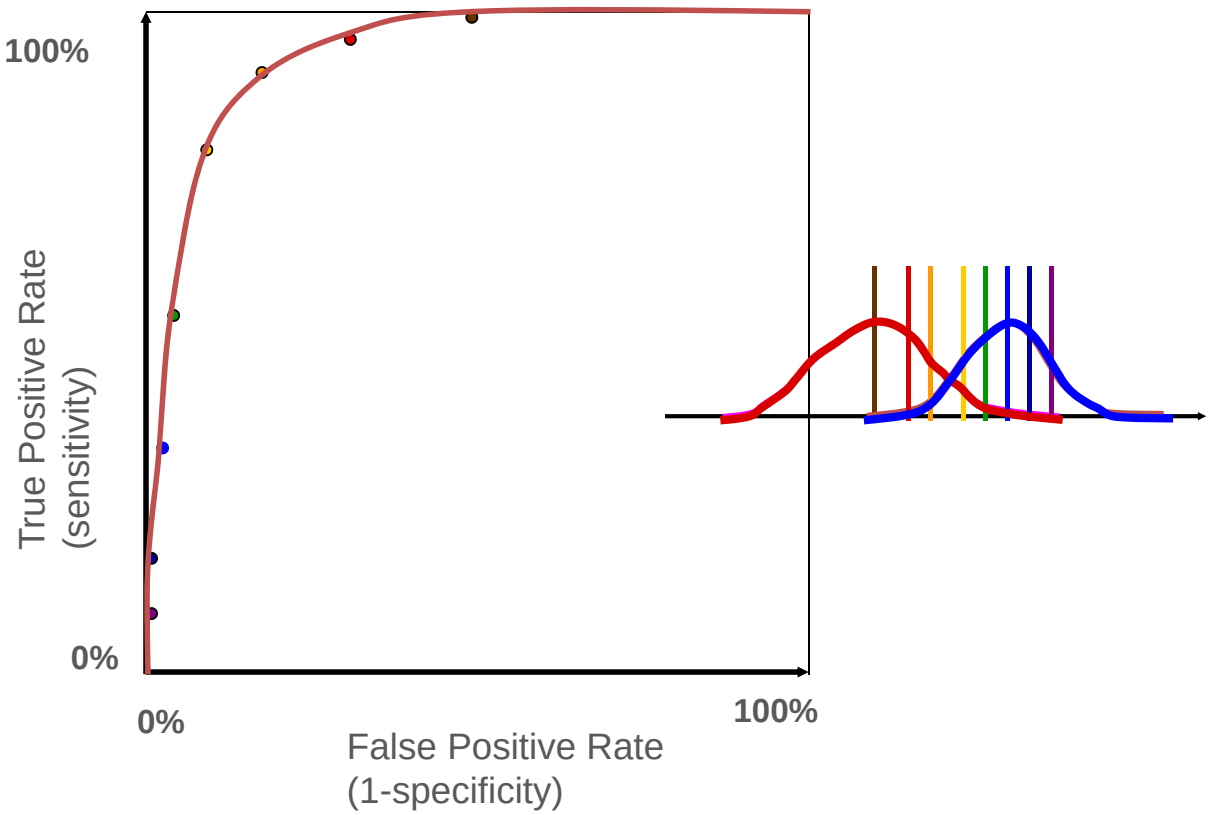
Moving the Threshold: left



Receiver Operating Characteristic (ROC) curves

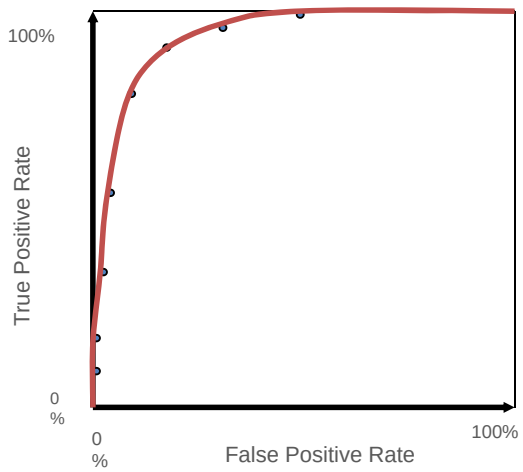
- Started in electronic signal detection theory (1940s - 1950s)
- Has become very popular in biomedical applications, particularly radiology and imaging
- Also used in machine learning applications to assess classifiers
- Can be used to compare tests/procedures

ROC curve

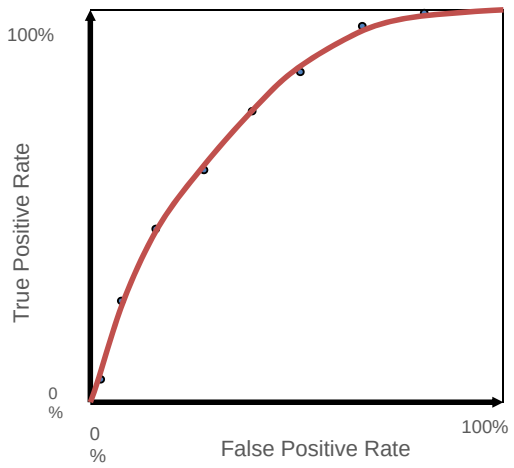


ROC curve comparison

A good test:

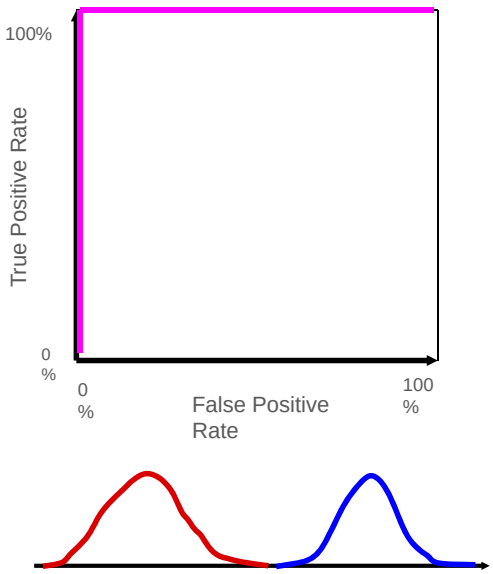


A poor test:



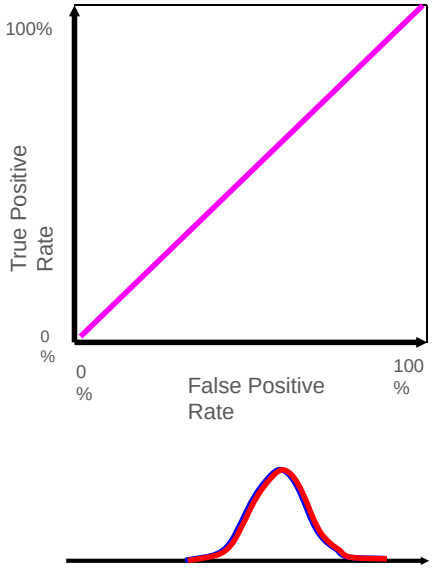
ROC curve extremes

Best Test



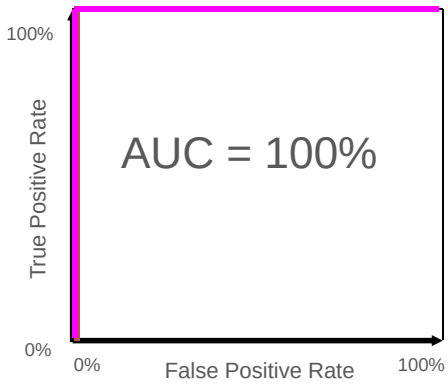
The distributions don't overlap at all

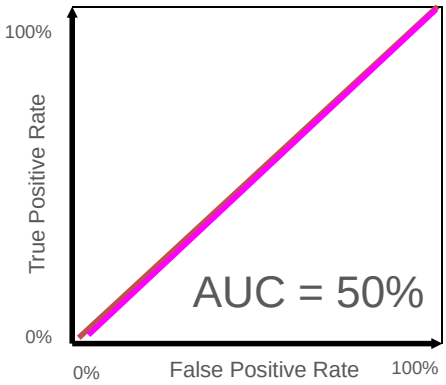
Worst test

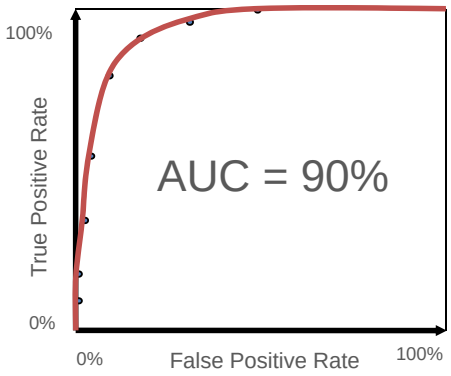


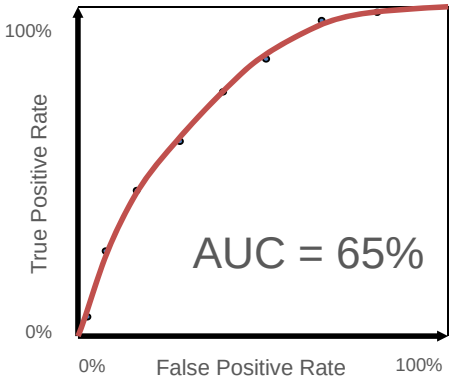
The distributions overlap completely

AUC for ROC curves









Area under ROC curve (AUC)

- ***Overall measure*** of test performance
- ***Comparisons*** between two tests based on differences between (estimated) AUC
- For continuous data, AUC equivalent to ***Mann-Whitney U-statistic*** (nonparametric test of difference in location between two populations)

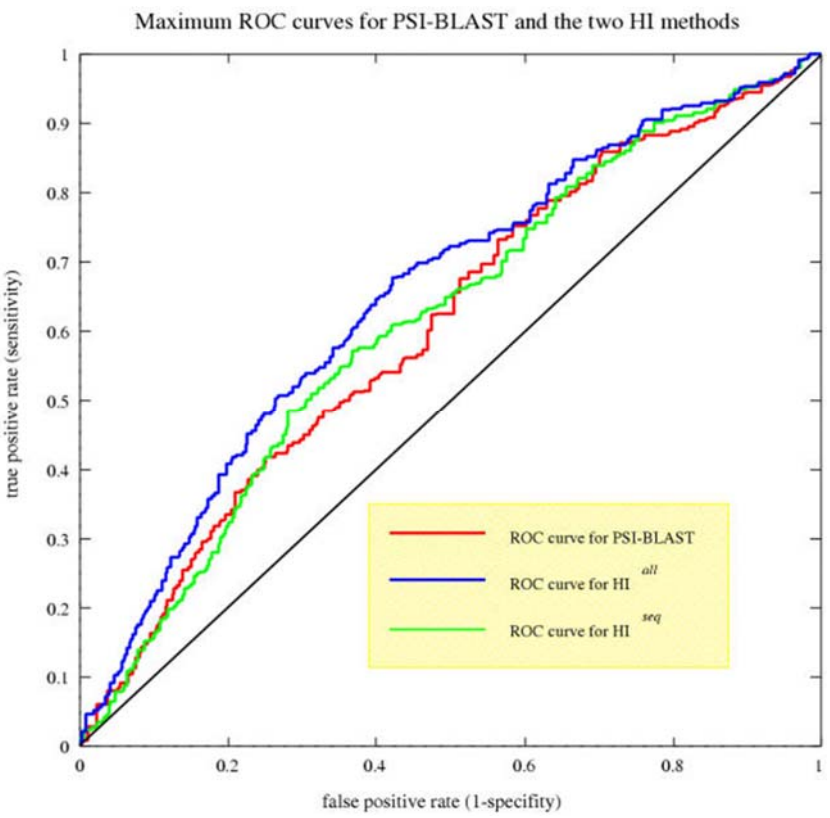
Problems with AUC

- ***No clinically relevant meaning***
- A lot of the area is coming from the range of ***large false positive*** values, no one cares what's going on in that region (need to examine restricted regions)
- The curves might ***cross***, so that there might be a meaningful difference in performance that is not picked up by AUC

Examples using ROC analysis

- Threshold selection for ‘tuning’ an already trained classifier (e.g. neural nets)
- Defining signal thresholds in DNA microarrays (Bilban *et al*.)
- Comparing test statistics for identifying differentially expressed genes in replicated microarray data (Lönnstedt and Speed)
- Assessing performance of different protein prediction algorithms (Tang *et al*.)
- Inferring protein homology (Karwath and King)

Example: Homology Induction ROC



VOLUME 30 · NUMBER 12 · APRIL 20 2012

JOURNAL OF CLINICAL ONCOLOGY

ORIGINAL REPORT

Identification of a Poor-Prognosis *BRAF*-Mutant–Like Population of Patients With Colon Cancer

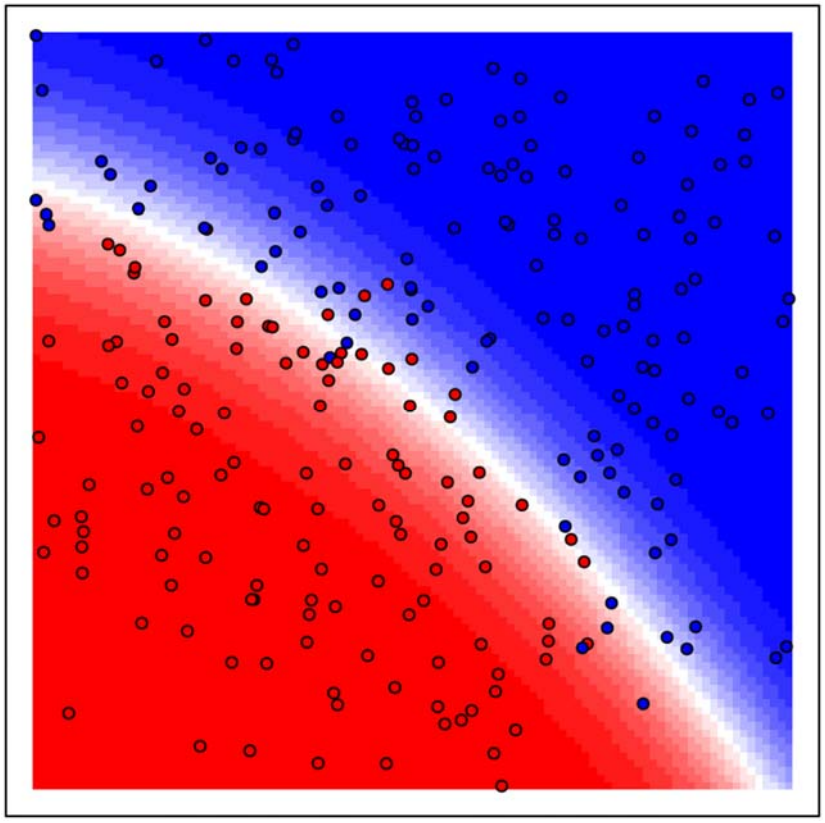
Vlad Popovici, Eva Budinska, Sabine Tejpar, Scott Weinrich, Heather Estrella, Graeme Hodgson, Eric Van Cutsem, Tao Xie, Fred T. Bosman, Arnaud D. Roth, and Mauro Delorenzi

For signature generation, an adapted version of the top scoring pairs algorithm[22] (multiple top scoring pairs [mTSP]; Data Supplement) was used, resulting in gene pairs deemed as the most informative in the process of classifier construction. The final classification model consisted of two groups of genes (G1 and G2), and the prediction was made comparing the averages of these groups: If, for a given sample, the average of G1 was smaller than the average of G2, then the sample was predicted to be BRAFm, otherwise WT2.

We also defined a *BRAF* score (BS) as the difference between the average expression of G2 genes and the average expression of G1 genes (from the mTSP model) and used it to analyze the stratification for different threshold values (a threshold of 0 leading to the original decision rule). An alternative threshold for the *BRAF* score was obtained as the value that maximized Matthews correlation coefficient[23] on the PETACC-3 data set.

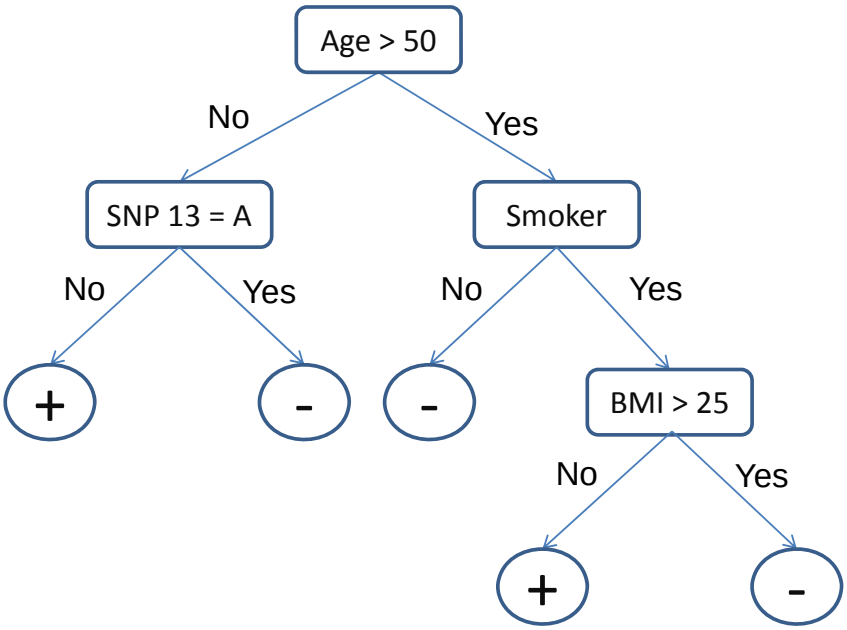
The performance of the classifier was estimated by repeated (10 times) stratified five-fold cross-validation, following the MAQC-II guidelines,[24] and measured in terms of sensitivity, specificity, and error rate. The final *BRAF* classifier was built from all BRAFm and WT2 samples in the PETACC-3 data set and then applied to the full PETACC-3 data set (including KRASm) and independent validation sets for the analysis of stratification of the population (Data Supplement). Because the stage II subgroup of PETACC-3 is smaller and not fully representative, the analysis of the prognostic value of the signature is focused on stage III subgroup. However, results for both stages are given (Data Supplement).

Quadratic discriminant analysis

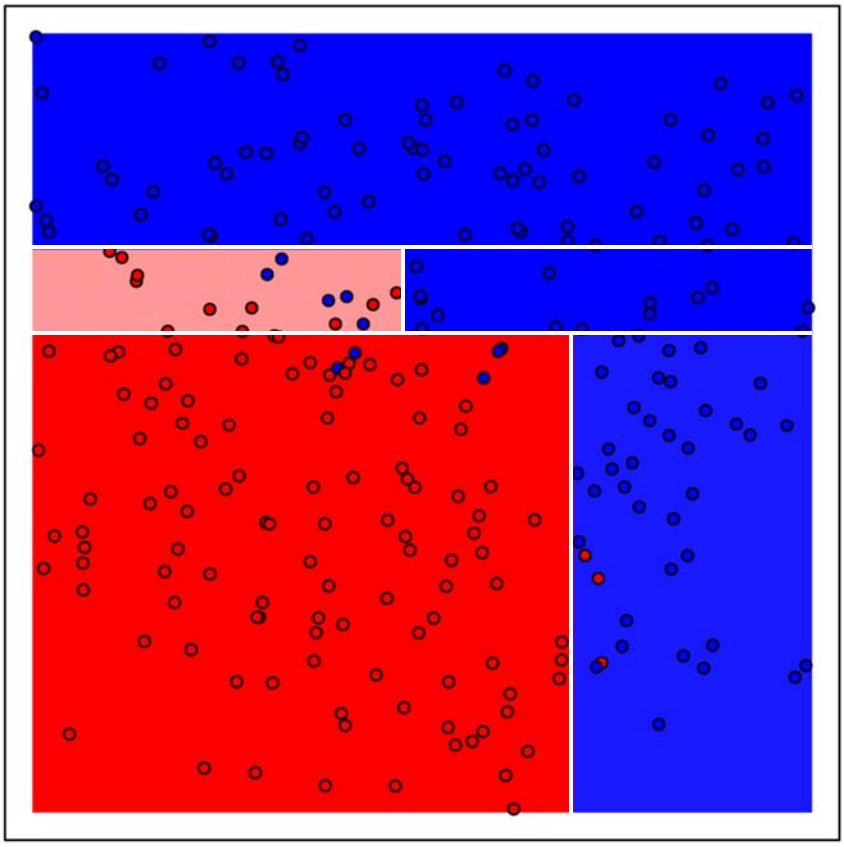


Random Forests

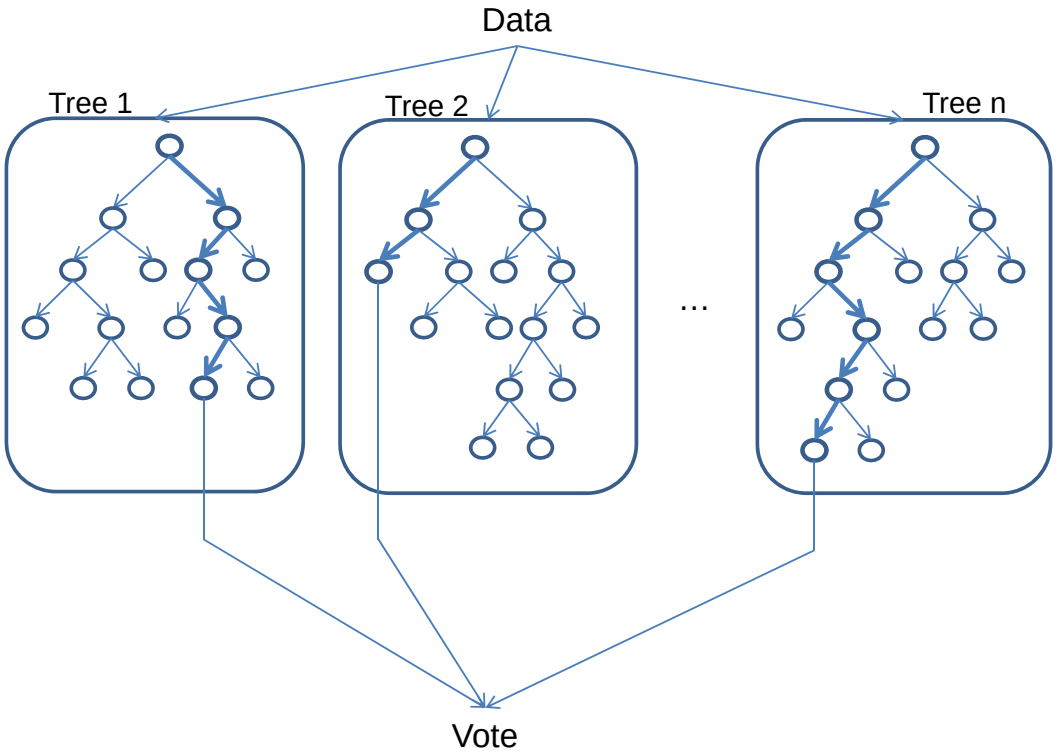
Decision trees

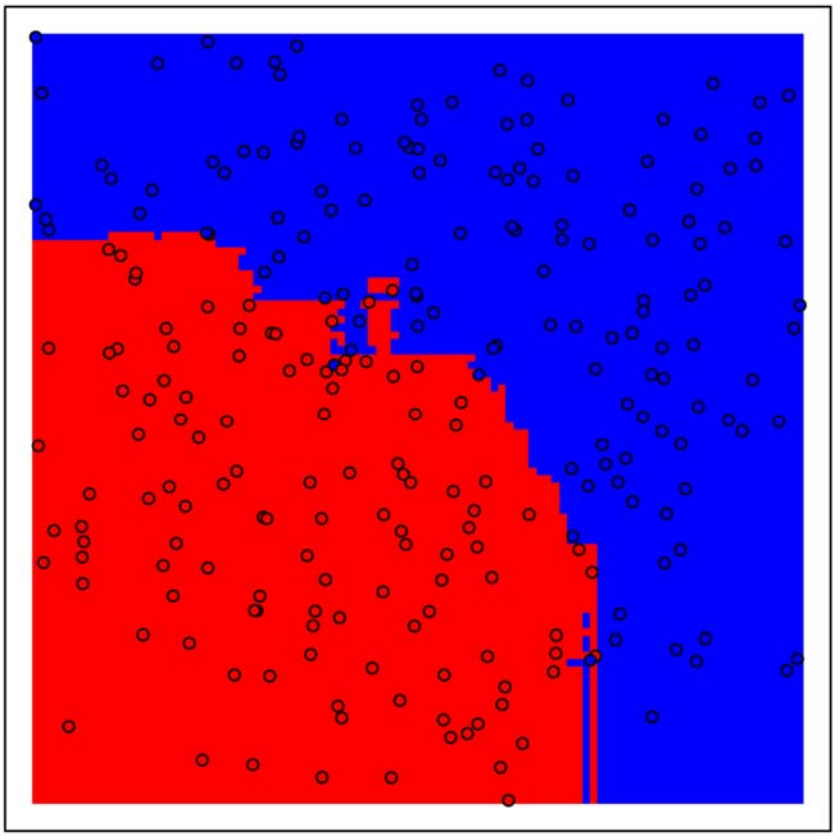


```
library(rpart)
fit = rpart(as.factor(group) ~ x1 + x2)

> fit
n= 250

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 250 119 1 (0.52400000 0.47600000)
   2) x2>=0.6112646 99  10 1 (0.89898990 0.10101010)
     4) x2>=0.7299728 71   1 1 (0.98591549 0.01408451) *
     5) x2< 0.7299728 28   9 1 (0.67857143 0.32142857)
      10) x1>=0.4927158 14   0 1 (1.00000000 0.00000000) *
      11) x1< 0.4927158 14   5 2 (0.35714286 0.64285714) *
   3) x2< 0.6112646 151  42 2 (0.27814570 0.72185430)
     6) x1>=0.7148586 40   3 1 (0.92500000 0.07500000) *
     7) x1< 0.7148586 111   5 2 (0.04504505 0.95495495) *
```



Random forest

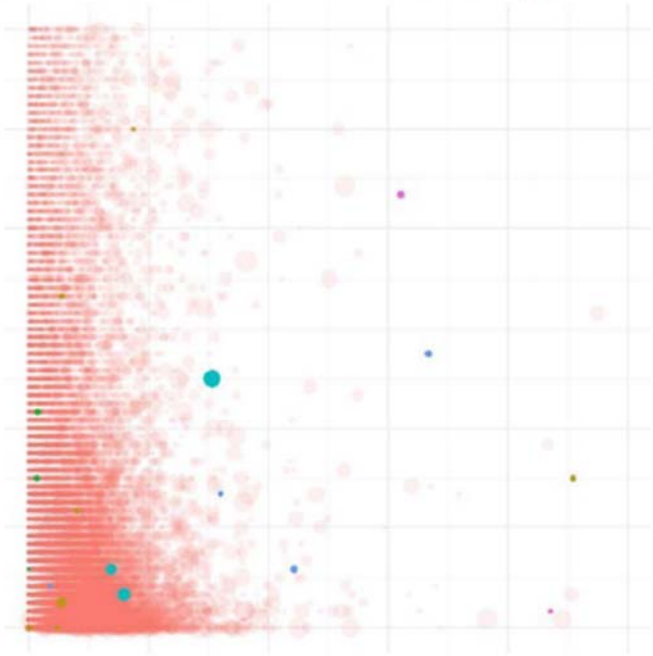




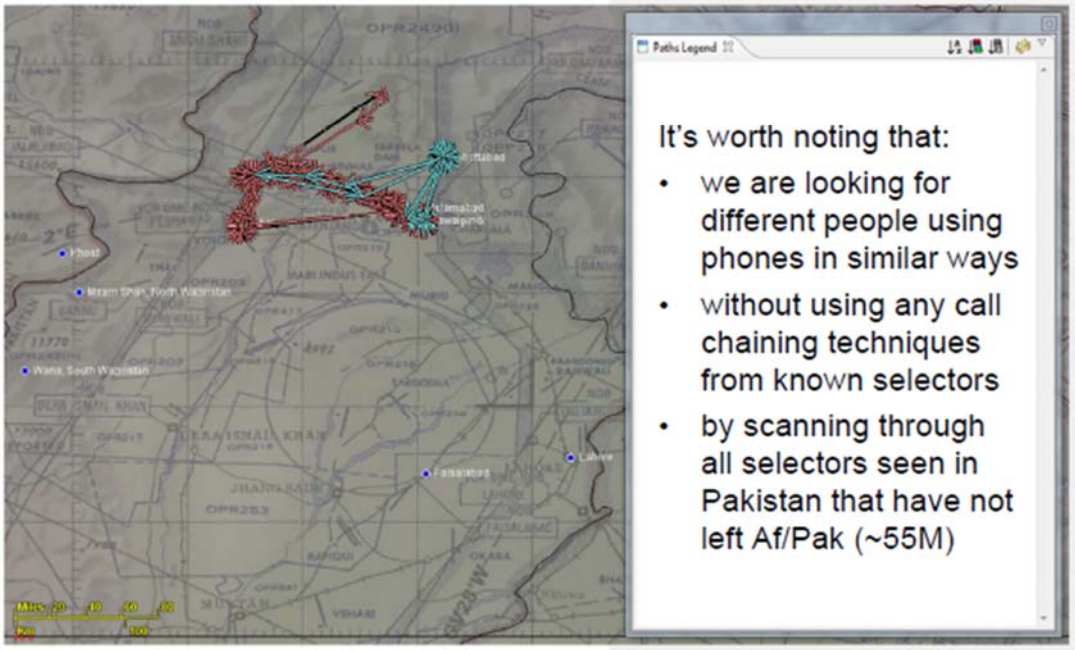
SKYNET:
Courier Detection via Machine Learning

[redacted], R66F/JHU
[redacted], R66F
[redacted], R66F
[redacted], T1211
[redacted], T1211
[redacted], S2I51
[redacted], S2I5/TD

June 5, 2012



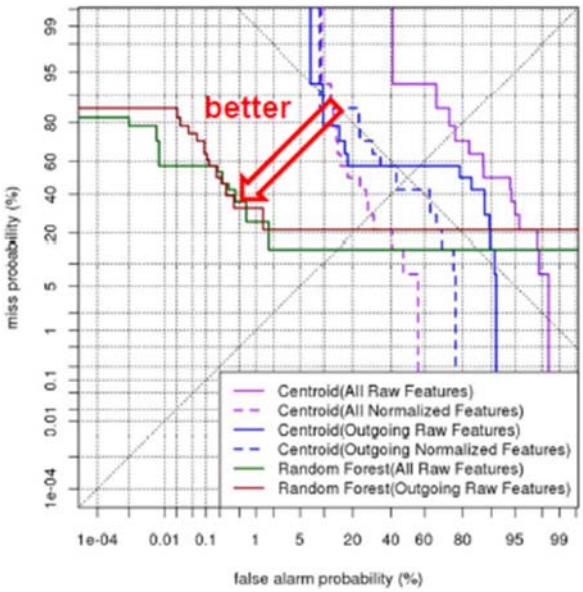
Given a handful of courier selectors, can we find others that "behave similarly" by analyzing GSM metadata?



Statistical algorithms are able to find the couriers at very low false alarm rates, if we're allowed to miss half of them

Random Forest Classifier

- 7 MSISDN/IMSI pairs
- Hold each pair out and then try to find them after learning how to distinguish remaining couriers fro n other Pakistanis
  (using 100k random selectors here)
- Assume that random draws of Pakistani selectors are nontargets
- 0.18% False Alarm Rate at 50% Miss Rate



We've been experimenting with several error metrics on both small and large test sets

Training Data	Classifier	Features	100k Test Selectors		55M Test Selectors	
			False Alarm Rate at 50% Miss Rate	Mean Reciprocal Rank	Tasked Selectors in Top 500	Tasked Selectors in Top 100
None	Random	None	50%	1/23k (simulated)	0.64 (active/Pak)	0.13 (active/Pak)
Known Couriers	Centroid	All	20%	1/18k		
		Outgoing	43%	1/27k		
	Random Forest		0.18%	1/9.9	5	1
+ Anchory Selectors						

Random Forest:

- 0.18% false alarm rate at 50% miss rate
- 7x improvement over random performance when evaluating its tasked precision at 100

Conclusion:
which algorithm to use ?

Differents families of machine-learning algorithms

We evaluate **179 classifiers** arising from **17 families** (discriminant analysis, Bayesian, neural networks, support vector machines, decision trees, rule-based classifiers, boosting, bagging, stacking, random forests and other ensembles, generalized linear models, nearest-neighbors, partial least squares and principal component regression, logistic and multinomial regression, multiple adaptive regression splines and other methods), implemented in Weka, R (with and without the caret package), C and Matlab, including all the relevant classifiers available today. We use **121 data sets**, which represent **the whole UCI** data base (excluding the large-scale problems) and other own real problems, in order to achieve significant conclusions about the classifier behavior, not dependent on the data set collection. **The classifiers most likely to be the bests are the random forest** (RF) versions, the best of which (implemented in R and accessed via caret) achieves 94.1% of the maximum accuracy overcoming 90% in the 84.3% of the data sets. However, the difference is not statistically significant with the second best, the SVM with Gaussian kernel implemented in C using LibSVM, which achieves 92.3% of the maximum accuracy. A few models are clearly better than the remaining ones: random forest, SVM with Gaussian and polynomial kernels, extreme learning machine with Gaussian kernel, C5.0 and avNNet (a committee of multi-layer perceptrons implemented in R with the caret package). The random forest is clearly the best family of classifiers (3 out of 5 bests classifiers are RF), followed by SVM (4 classifiers in the top-10), neural networks and boosting ensembles (5 and 3 members in the top-20, respectively).

Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?
Fernandez-Delgado et al, Journal of Machine Learning Research 15 (2014) 3133-318