Data Analysis in Software Engineering

Javier Dolado
U. País Vasco/Euskal Herriko Unibertsitatea

Daniel Rodríguez
Universidad de Alcalá

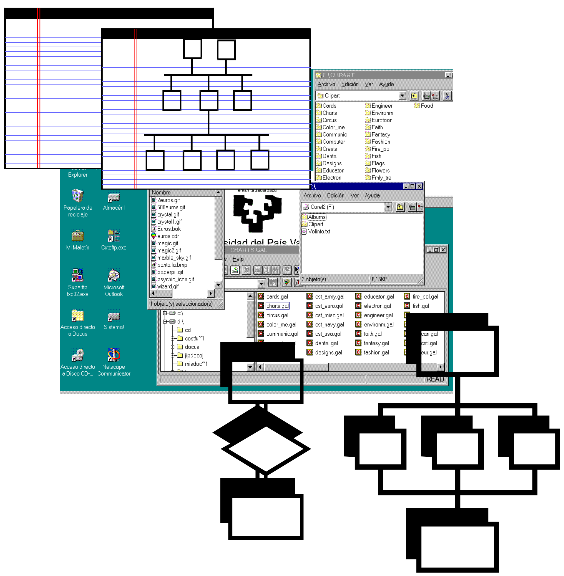
Javier Tuya
Universidad de Oviedo

PRESI TIN2013-46928-C3-1-R, TIN2013-46928-C3-2-R

Outline

- **Problems of Software Engineering, Data Analysis and Data Mining**
  - Sofware Cost Estimation, Software Size Estimation
  - Process measurement and estimation
  - Software Quality/Testing
- **Methods**
  - Supervised or Predictive:
    - Regression, Genetic Programming, Decision trees, k-NN, etc.
  - Unsupervised:
    - Clustering, Assocition rules
  - Others: Semisupervised learning, text mining, SNA, etc.
  - Experimentation and Hypothesis Tests (comparison of methods)
- **Tools**
- **Results and Discussion**

Problem: Prediction

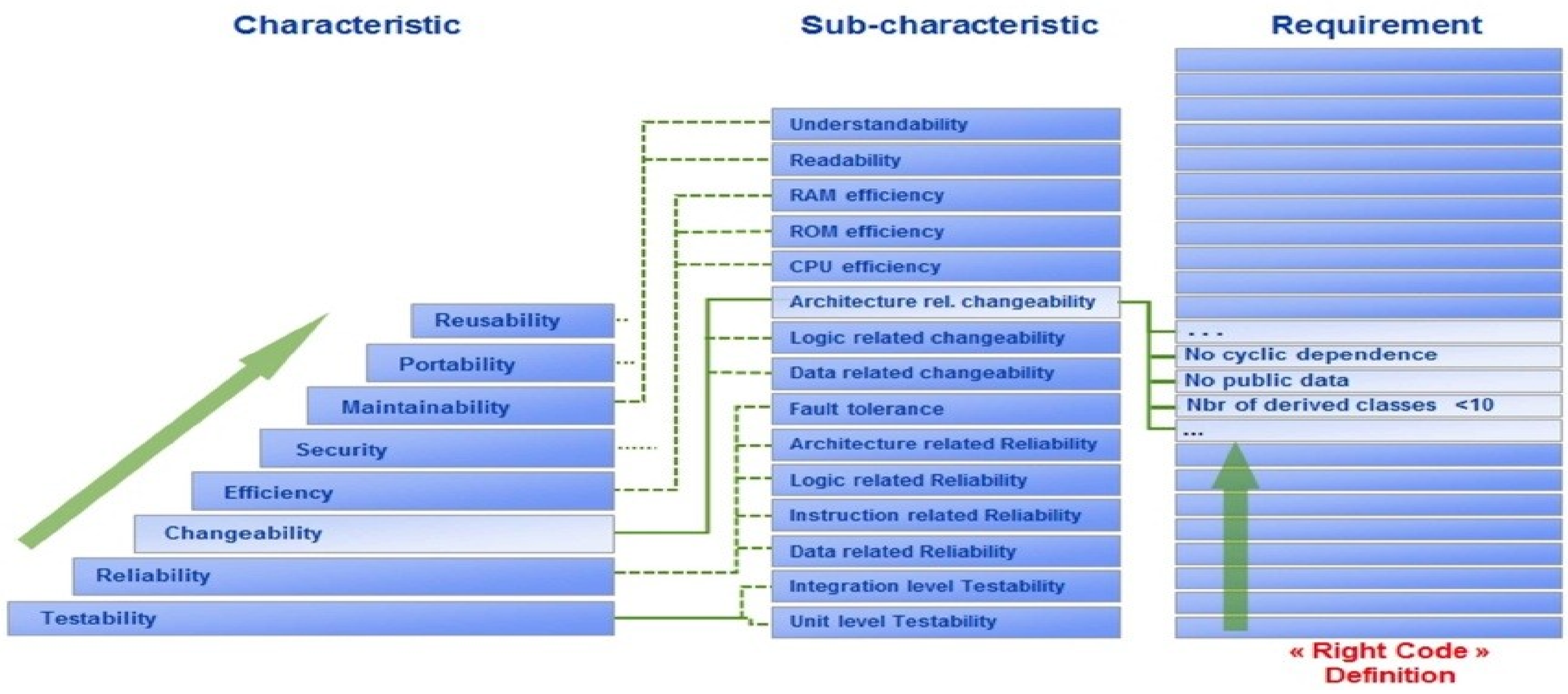


Parameters, data collected, previous projects, etc.

The estimation of cost, si defects, quality, etc has always been a problem

Problem: Quality

- Technical Debt:
  - work to be done before a can be considered properly finished
  - **SQALE** method



Letouzey J.L., Ilkiewicz, M., Managing Technical Debt with the SQALE Method, IEEE Software, 29(6),2012,pp 44-51

Problem: Defect Prediction/testing

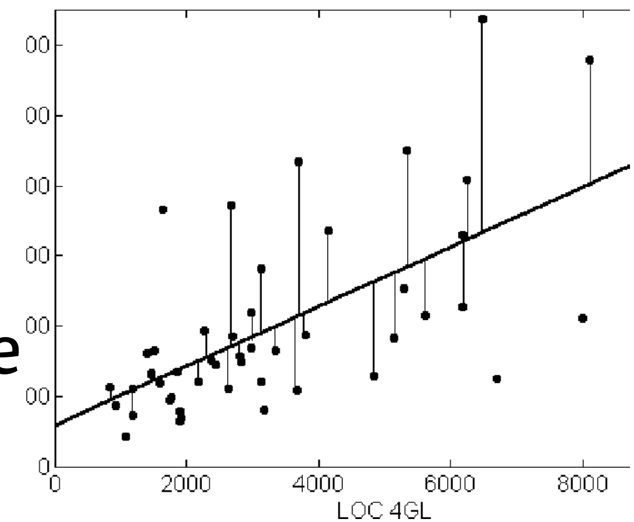
- Defect Prediction:
  - Which modules/classes/components are error-prone?
- Testing
  - Integration testing
    - Which test should we run?
    - In which order?

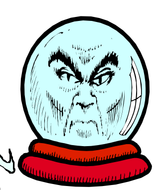
Strategy: Build *models* from **data**

Data

Important: data sources must be relevant and reliable

- Different methods are applied depending of the type of problem



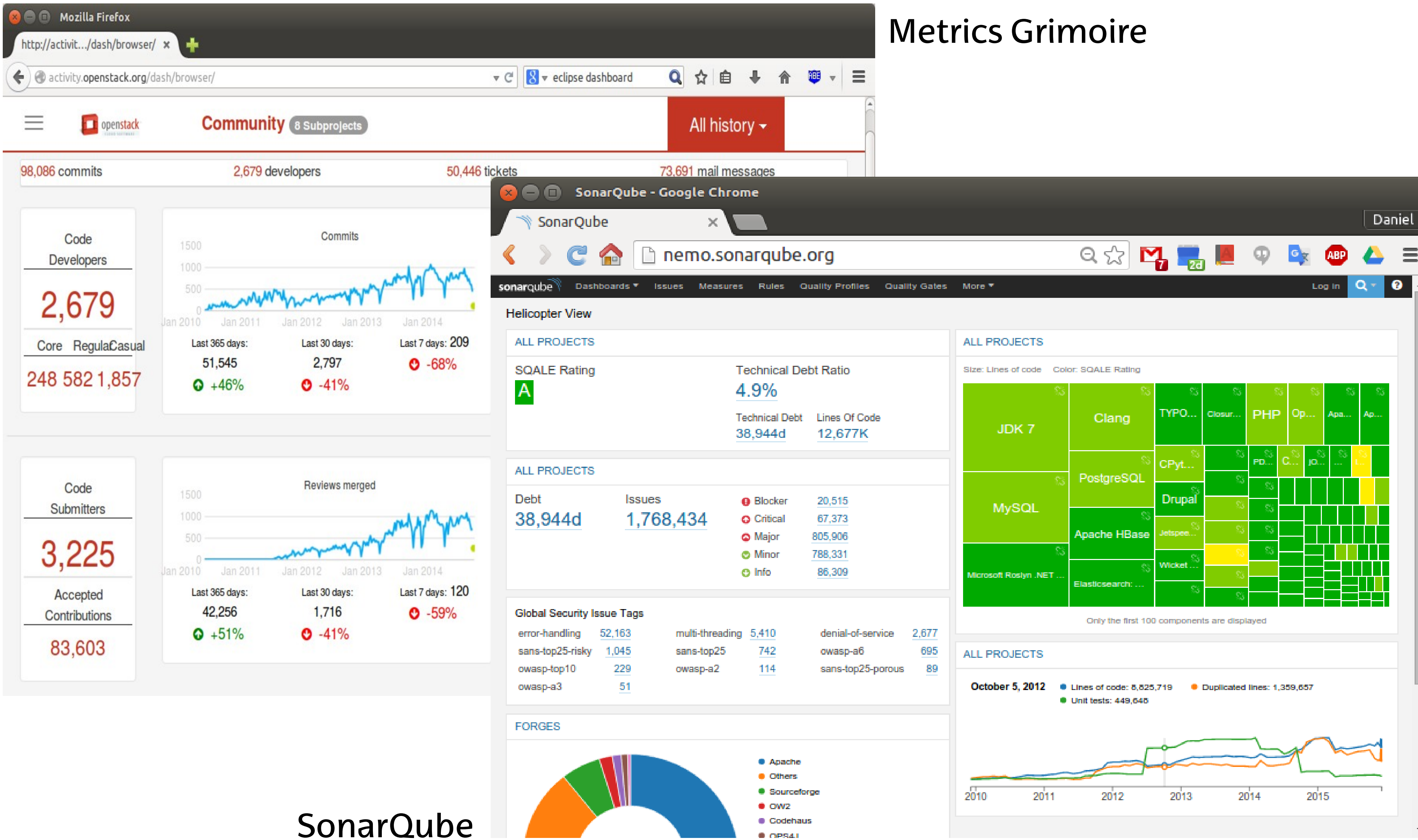


Or "Guesstimating"

Where **data** comes from

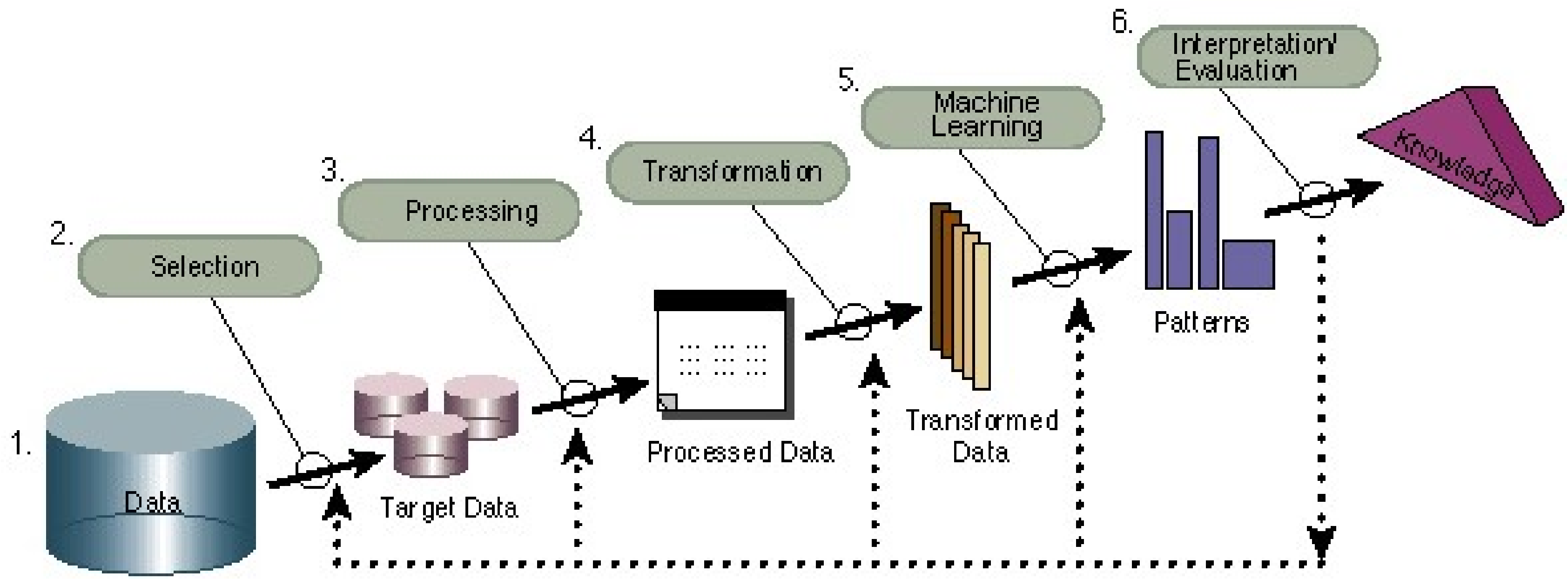
Where **data** comes from

Metrics Grimoire



SonarQube

Knowlege Discovery in Dbs (KDD)



An Overview of the Steps That Compose the KDD Process

(Fayyad *et al.*, 96)

Methods: Classification

- **Supervised learning** which aims to discover knowledge for classification or prediction (predictive)
  - Decision trees such as C4.5 (Quilan) or ID3.
  - Rule induction
  - Lazy techniques k-nearest neighbour (k-NN), CBR
  - RegresionNumeric prediction:
    - Regression Techniques, SVM, NN
  - Neural Networks
  - Statistical Techniques: Bayesian networks classifiers
  - Meta-techniques

A_1	...	A_n	C
$a_{1,1}$	...	$a_{1,n}$	c_1
...	...	...	...
$a_{m,1}$		$a_{m,1}$	c_m

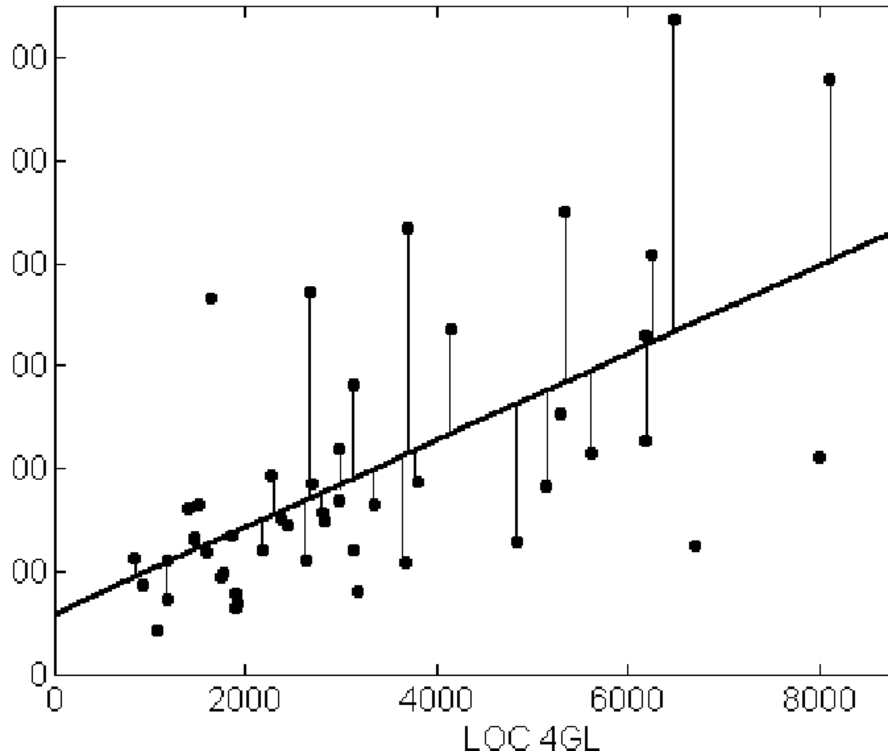
- **Unsupervised learning** which refers to the induction to extract interesting knowledge from data (descriptive)
  - Clustering (k-means, EM)
  - Association Rules (Apriori)

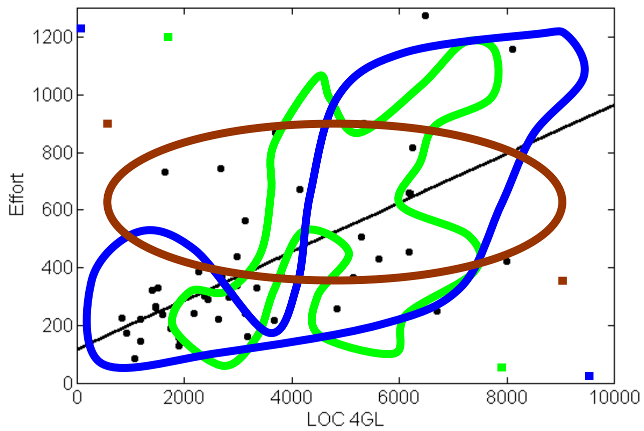
A_1	...	A_n
$a_{1,1}$	...	$a_{1,n}$
...	...	...
$a_{m,1}$		$a_{m,1}$

- Other approaches:
  - Time Series Analysis
  - Simulation
  - Semisupervised learning, Subgroup Discovery, etc.

Examples :Regression and Curve Estimation

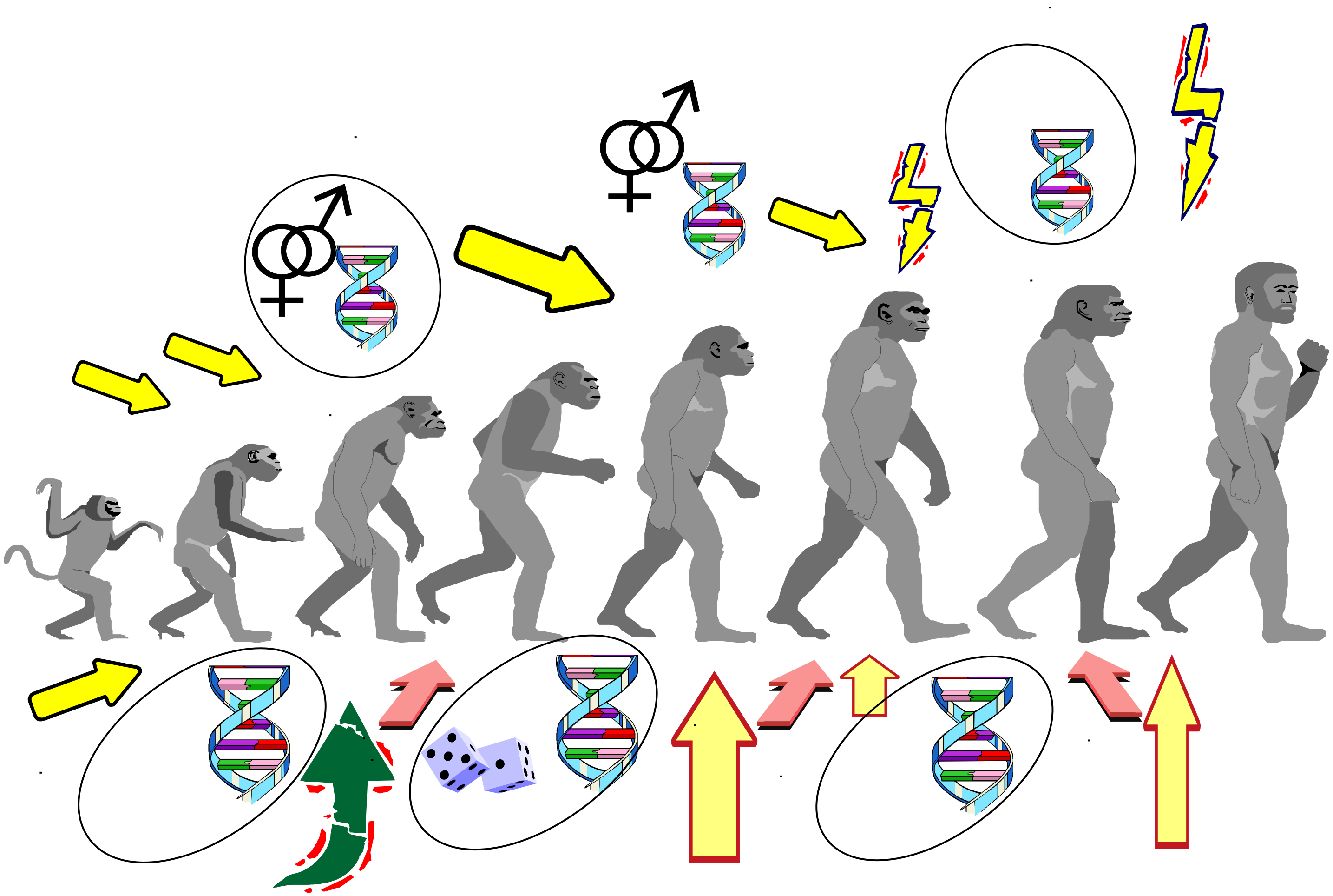
- Probably, the most used method for estimation.
- It is simple and it obtains results as good as other more complex methods

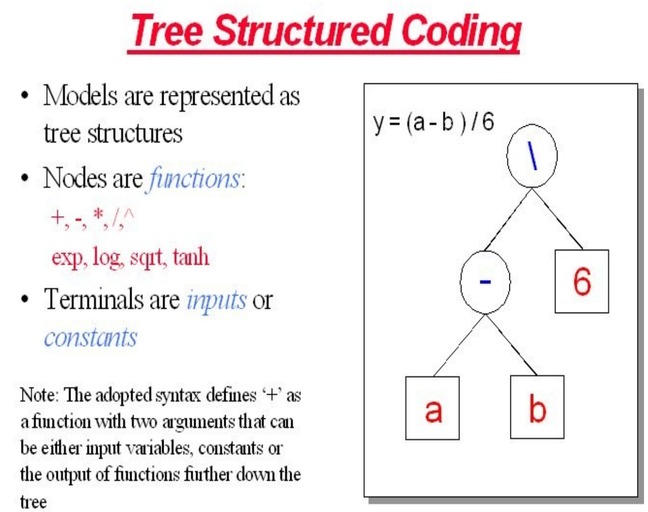


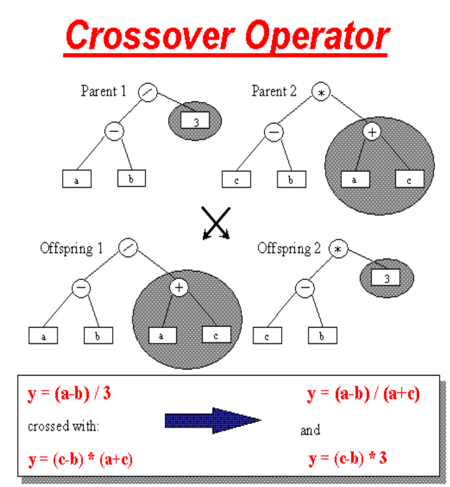


Example: Genetic Programming

- Tries to mimic one of the methods of evolution

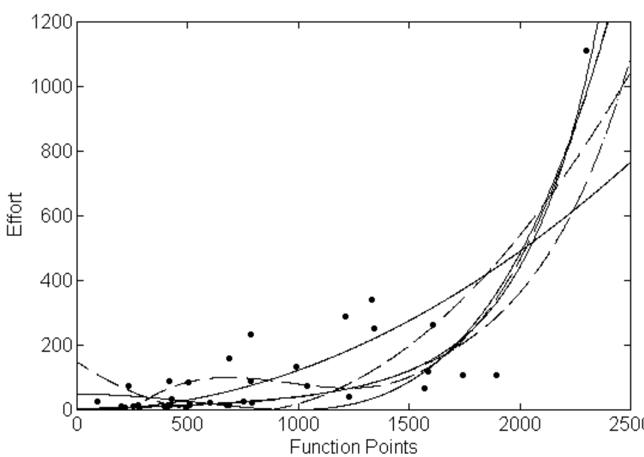


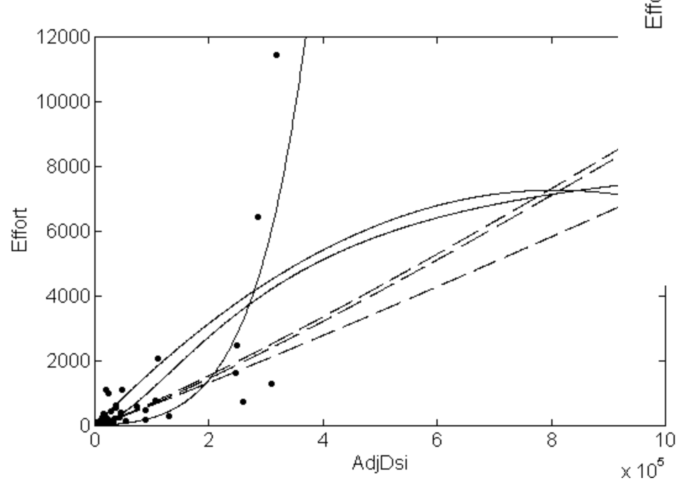


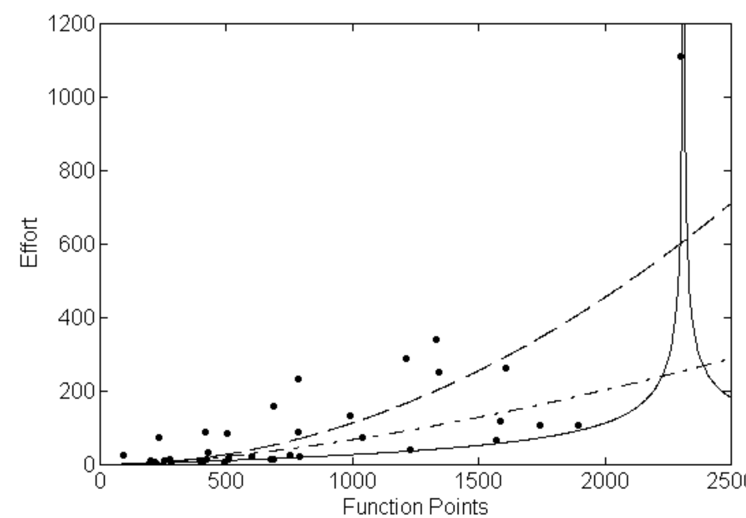


Example: Genetic Programming (cont)

- Genetic programming allows us to adjust almost any equation. GP gives always good results, with the proper adjustment of parameters.
- We can always find a "good model"

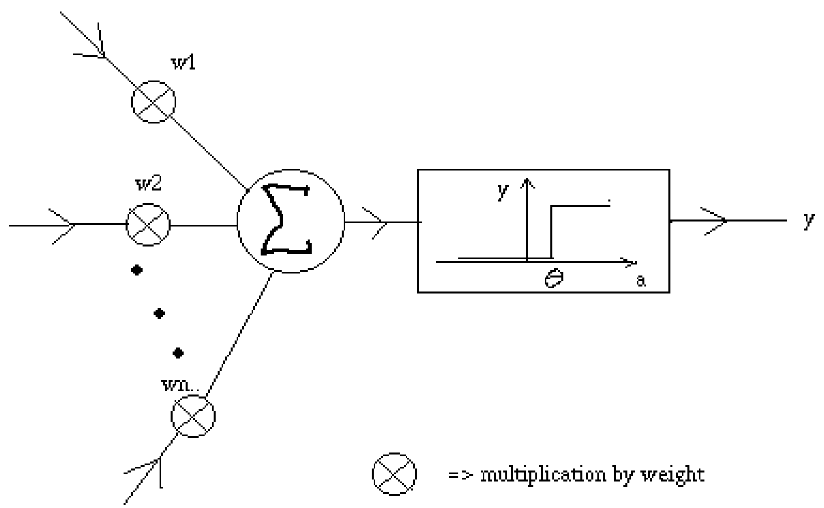


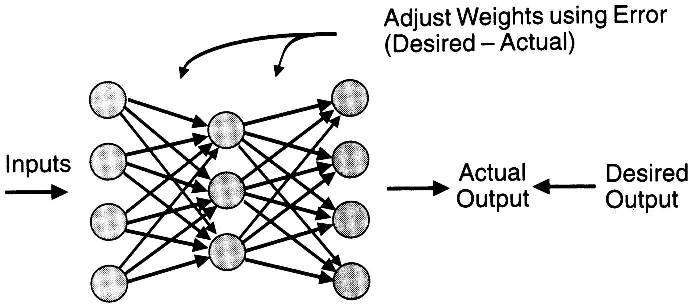




Example: Neural Networks

- All methods are based on a specific paradigm and purpose, therefore their application must be carefully examined
- Neural networks provide “moderate good predictions“





Example: k-NN

	Size	Effort	Defects	...	Quality
P1	123	64	218	...	High
P2	657	256	4783	...	High
...	...	...	...	...	...
P 98	349	118	2846	...	Low

Data

New case

150	?	200	...	?

P45	123	64	218	...	High
P64	143	68	267	...	Low

Similar ones

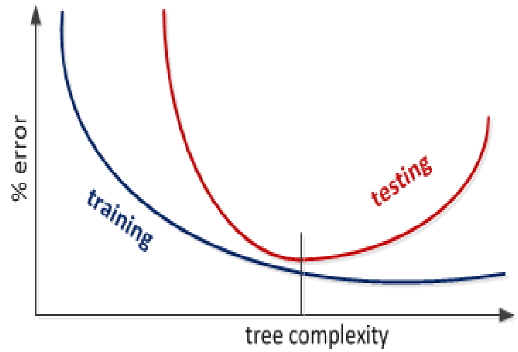
Combine

Estimate

150	76	200	...	High

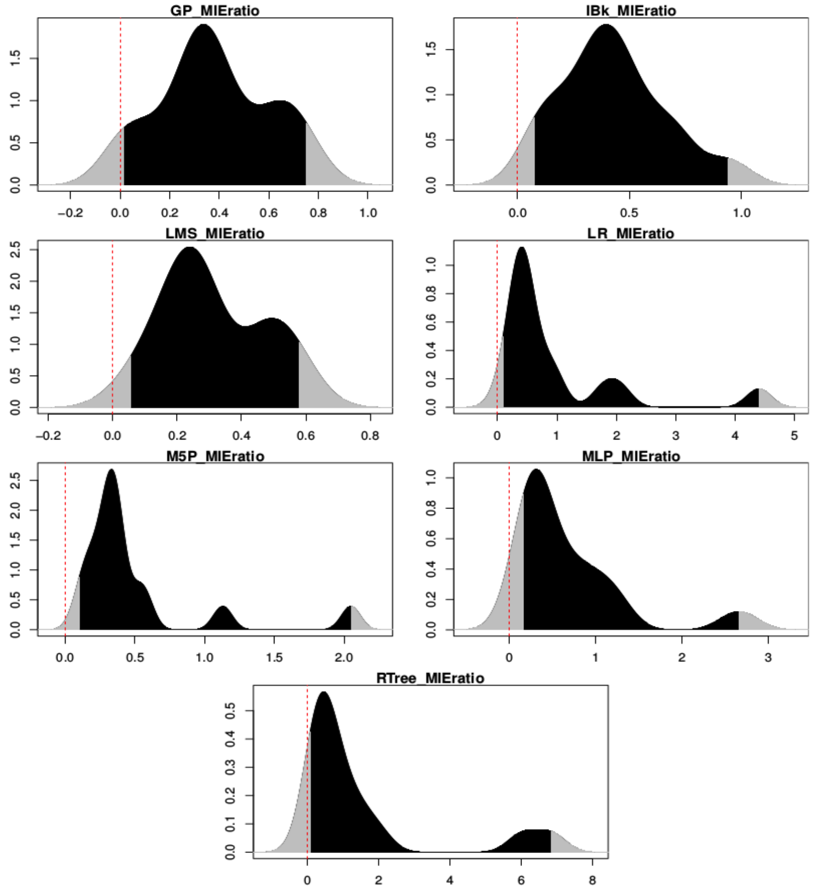
Evaluation of methods

- Dividing into training and testing datasets
  - Holdout, Cross Validation, LOO
- Need to be careful with
  - Overfitting vs underfitting
  - Imbalance, overlaping, etc.
- Many evaluation measures
  - Continuous (numeric) classes (MRE, RSME, etc)
  - Discrete classes (many based on the confusion matrix)



		Predicted		
		Positive	Negative	
Actual	Positive	TP True Positive	FN False Negative (Type II error)	TPrate=TP/(TP+FN) (Sensitivity, Recall)
	Negative	FP False Positive (Type I error)	TN True Negative	TNrate=TN/(FP+TN) (Specificity)
		PPV=TP/(TP+FP) Positive Predictive Value (Confidence, Precision)	NPV=TN/(FN+TN) Negative Predicted Value	Accuracy= TP+FP/(TP+TN+FP+FN)

In software cost estimation there are two methods that perform reasonably well …

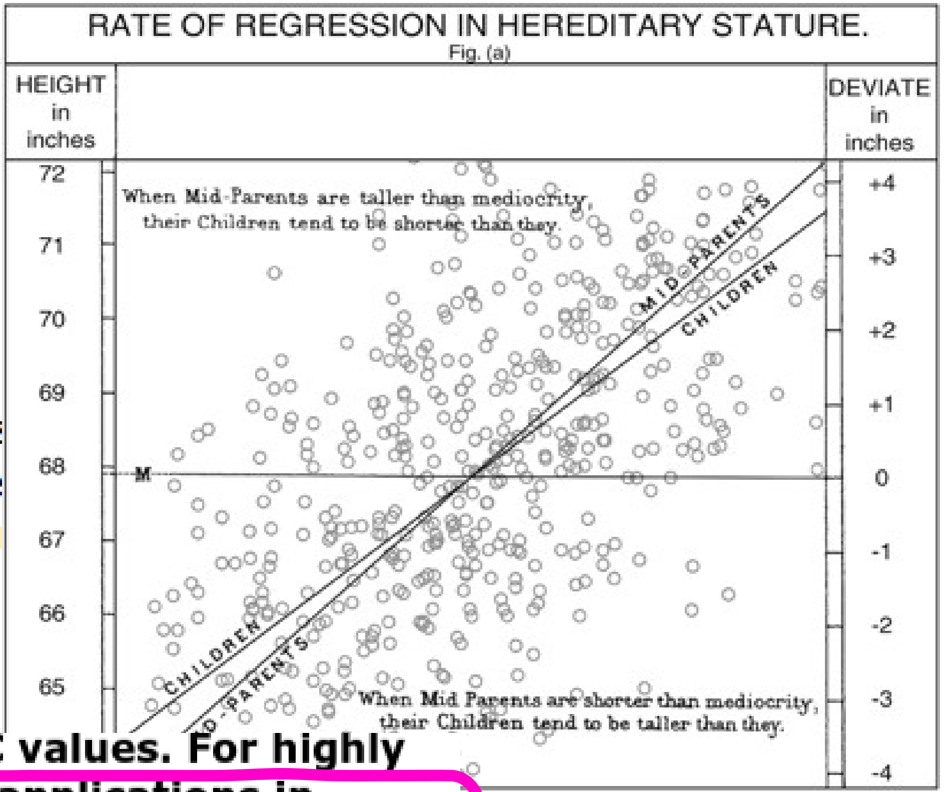


	Qtle. 2.5%-97.5%	HPD low-upper	M-Hast. 2.5%-97.5%
GP	0.021-0.725	0.015-0.751	0.273-1.417
Bk	0.096-0.859	0.073-0.943	0.317-0.733
MS	0.088-0.566	0.056-0.581	0.239-0.493
LR	0.162-3.582	0.103-4.397	0.569-1.962
M5P	0.124-1.727	0.102-2.048	0.33-0.831
MLP	0.171-2.161	0.168-2.662	0.44-1.216
RTree	0.169-6.56	0.096-6.841	0.78-4.506

Table 3: This table shows different probabilistic intervals for each one of the 7 methods ($\alpha = 0.05$) for the data of the MIEratios. Scale is 0-∞. Lower values are better.

Don't underestimate the value of simple methods...

Sir Francis Galton, 1886



Article

European Journal of Human Genetics (2009) **17,** 1070–1075; doi:10.1038/ejhg.2009.5; published online 18 February 2009

Predicting human height by Victorian genomic methods

Yurii S Aulchenko[1,2,7], Maksim V Struchalin[1,3,7], Nadez
M Belonogova[2,4], Tatiana I Axenovich[2], Michael N Wee
Albert Hofman[1], Andre G Uitterlinden[6], Manfred Kayse
Ben A Oostra[1], Cornelia M van Duijn[1], A Cecile J
W Janssens[1] and Pavel M Borodin[2,4]

genomic profile should explain to reach certain AUC values. For highly heritable traits such as height, we conclude that in applications in which parental phenotypic information is available (eg, medicine), the Victorian Galton's method will long stay unsurpassed, in terms of both discriminative accuracy and costs. For less heritable traits, and in situations in which parental information is not available (eg, forensics), genomic methods may provide an alternative, given that

Results

- We've applied many statistical methods to different Soft Eng problems including, cost, time, defects and others.
- We have applied Equivalence Hypothesis Testing to several software engineering experiments
- A big problem: Show me the data!
  - Public data is not always relevant to our specific domain
  - It is much better to collect the data within the organization
- There is no “best method“
  - No free lunch theorem
  - They need to be understood and tuned
  - Bayesian Networks can be applied in the sw testing area

Discussion

- Many methods available that are easy to apply, however...
    - their way of working (theory) needs to be understood
    - they need to be tuned! (many parameters)
- Many tools available:
    - For Software Engineering (data collection and metrics).
    - For machine learning:
        - Open source: R, Weka, Python (scikit learn, ScyPy),
        - Closed: Matlab, mathematica...
- Data from public sources cannot be applied to other settings in a straightforward way
    - It's almost unavoidable to use 'within-company' data

Acknowledgements

PROJECTS

“Testing of data persistence and user perspective under new paradigms“

“Gamificación y prototipado de procesos para la detección temprana de oportunidades en la producción del software“

PRESI TIN2013-46928-C3-1-R, TIN2013-46928-C3-2-R

Ministerio de Economía y Competitividad