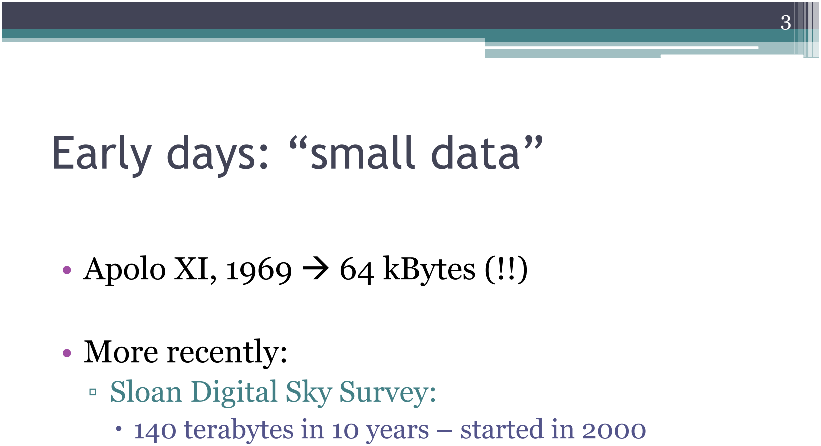
Ciência de Dados em Larga Escala

Inês Dutra and Zafeiris Kokkinogenis

DCC-FCUP
room 1.31
ines@dcc.fc.up.pt
zafeiris.kokkinogenis@gmail.com

23/24

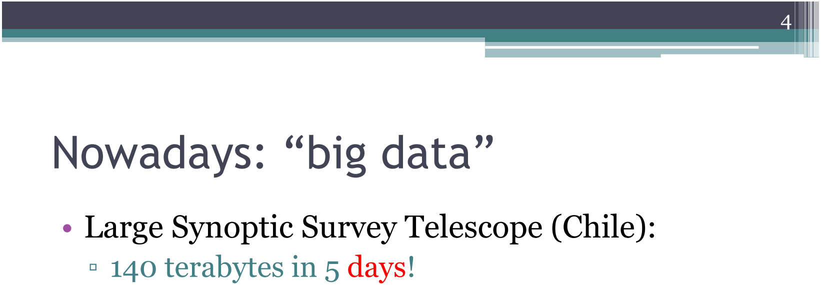
Introduction



Early days: “small data”

- Apolo XI, 1969 → 64 kBytes (!!)
- More recently:
  - Sloan Digital Sky Survey:
    - 140 terabytes in 10 years – started in 2000

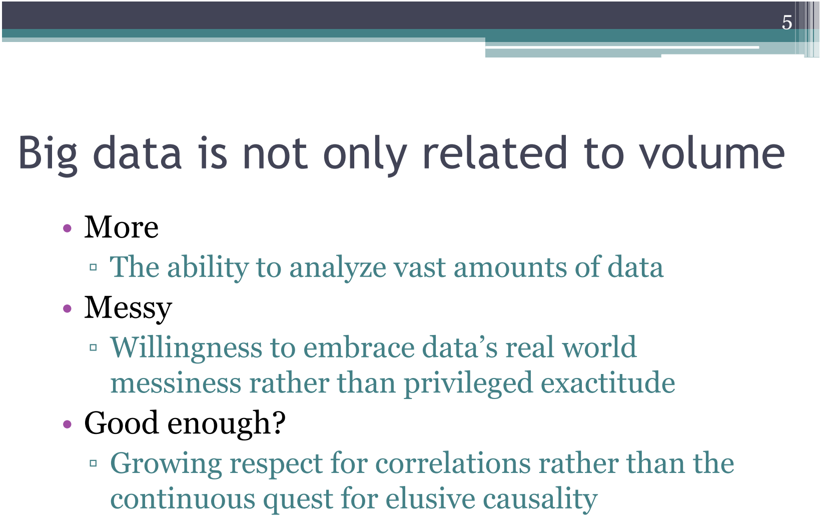
Introduction



Nowadays: “big data”

- Large Synoptic Survey Telescope (Chile):
  - 140 terabytes in 5 days!

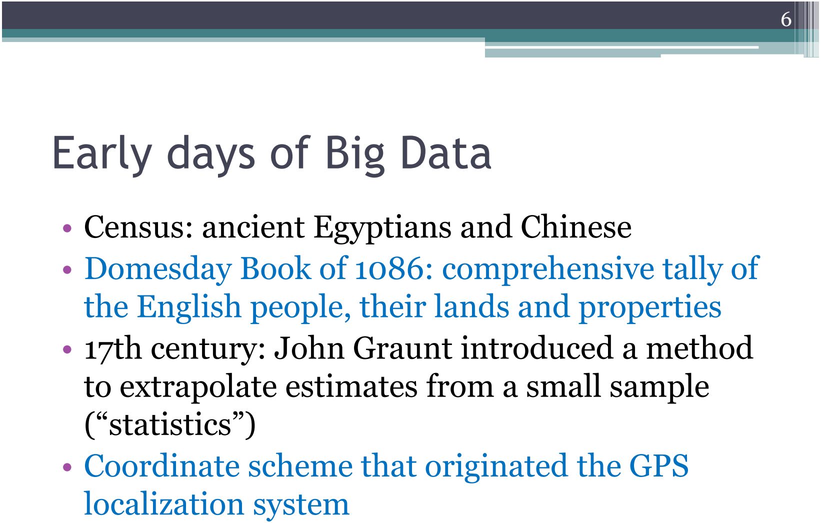
Introduction



Big data is not only related to volume

- More
  - The ability to analyze vast amounts of data
- Messy
  - Willingness to embrace data’s real world messiness rather than privileged exactitude
- Good enough?
  - Growing respect for correlations rather than the continuous quest for elusive causality

Introduction



Early days of Big Data

- Census: ancient Egyptians and Chinese
- Domesday Book of 1086: comprehensive tally of the English people, their lands and properties
- 17th century: John Graunt introduced a method to extrapolate estimates from a small sample (“statistics”)
- Coordinate scheme that originated the GPS localization system

Some definitions

- **Big data**:
  - Early definition:
    - volume of information that does not fit into memory

Some definitions

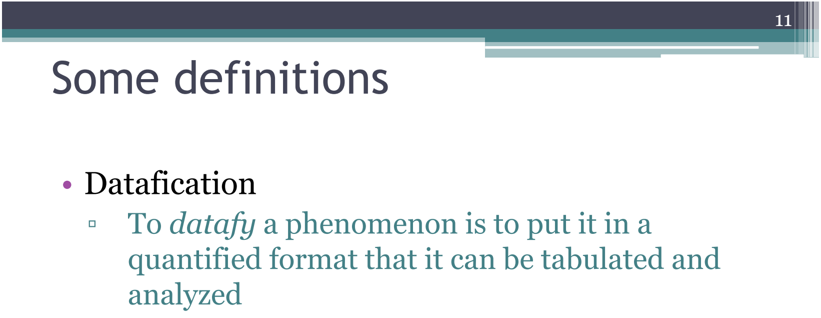
- **Big data**:
  - Early definition:
    - volume of information that does not fit into memory
  - More modern definition:
    - the ability of society to harness information in novel ways to produce useful insights or goods and services of significant value
    - things one can do at a large scale that cannot be done at a smaller one, to extract insights…

Introduction

Some definitions

- **Big data**:
  - Early definition:
    - volume of information that does not fit into memory
  - Modern definition:
    - the ability of society to harness information in novel ways to produce useful insights or goods and services of significant value
    - things one can do at a large scale that cannot be done at a smaller one, to extract insights…
  - BUT, no common agreement ☹

Introduction

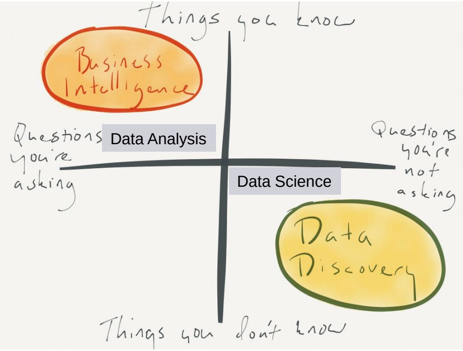


Some definitions

- Datafication
  - To *datafy* a phenomenon is to put it in a quantified format that it can be tabulated and analyzed

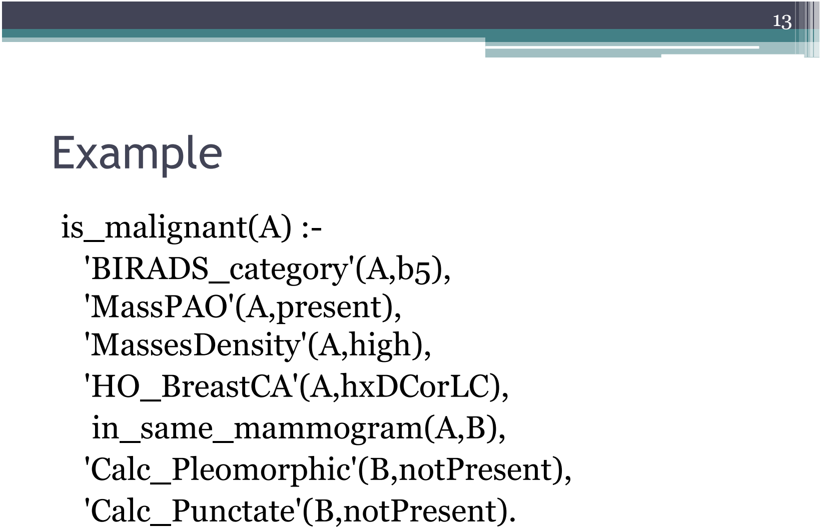
Some definitions

- Data Analysis x Data Science



http://www.applieddatalabs.com/content/new-reality-business-intelligence-and-big-data

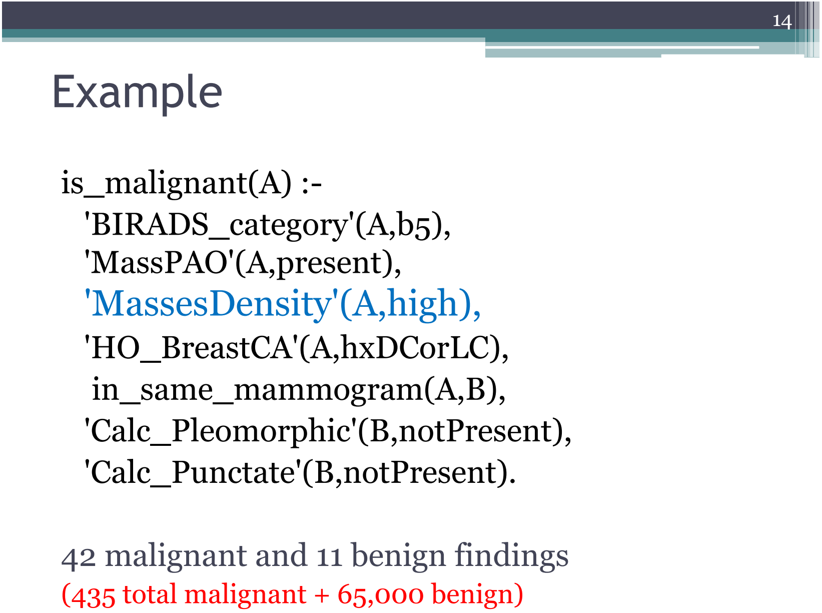
Introduction



Example

```
is_malignant(A) :-
   'BIRADS_category'(A,b5),
   'MassPAO'(A,present),
   'MassesDensity'(A,high),
   'HO_BreastCA'(A,hxDCorLC),
    in_same_mammogram(A,B),
   'Calc_Pleomorphic'(B,notPresent),
   'Calc_Punctate'(B,notPresent).
```

Introduction



Example

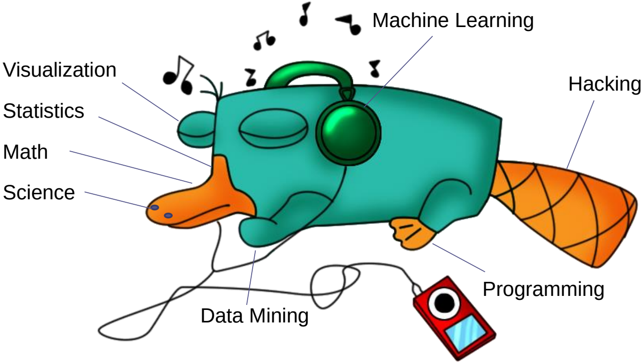
```
is_malignant(A) :-
   'BIRADS_category'(A,b5),
   'MassPAO'(A,present),
   'MassesDensity'(A,high),
   'HO_BreastCA'(A,hxDCorLC),
    in_same_mammogram(A,B),
   'Calc_Pleomorphic'(B,notPresent),
   'Calc_Punctate'(B,notPresent).
```

42 malignant and 11 benign findings

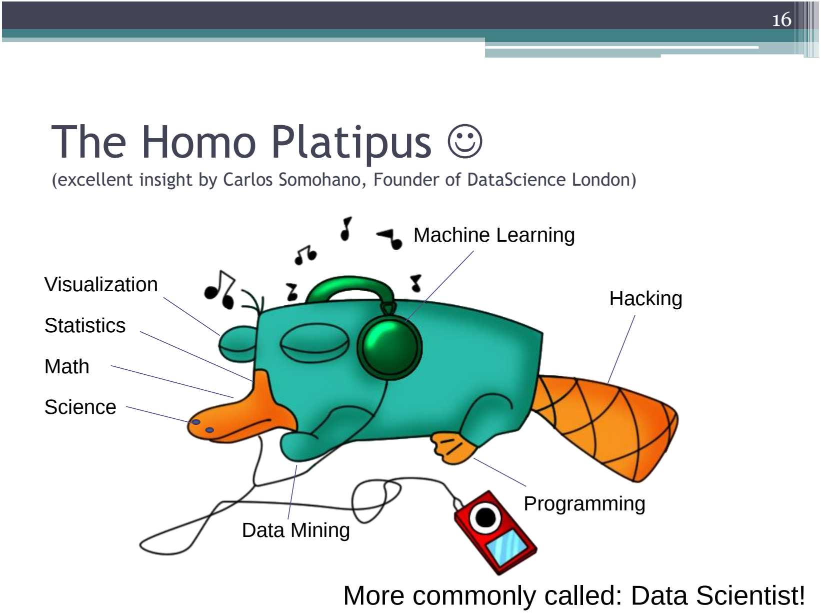
(435 total malignant + 65,000 benign)

The Homo Platipus ☺

(excellent insight by Carlos Somohano, Founder of DataScience London)



Introduction

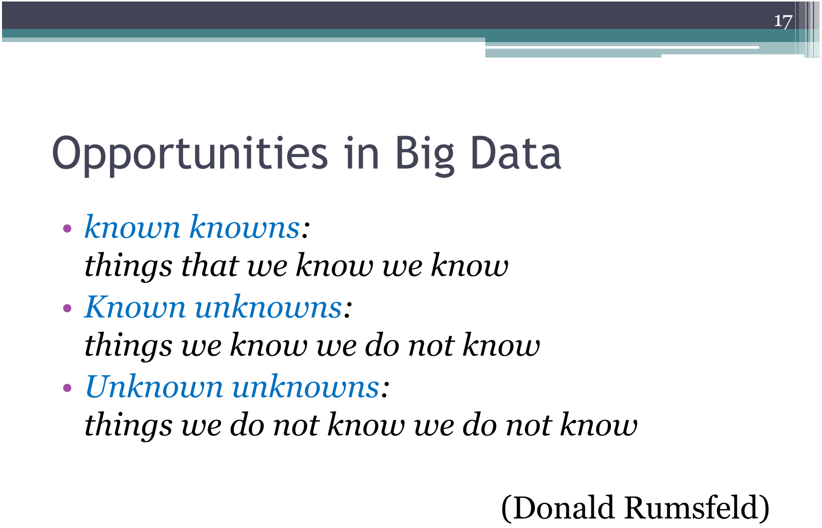


The Homo Platipus ☺

(excellent insight by Carlos Somohano, Founder of DataScience London)

More commonly called: Data Scientist!

Introduction



Opportunities in Big Data

- *known knowns*:
  things that we know we know
- *Known unknowns*:
  things we know we do not know
- *Unknown unknowns*:
  things we do not know we do not know

(Donald Rumsfeld)

Introduction

Big Data

- Unstructured x Structured
- Different sources of information
- Data from multiple tables
- Different formats

Data $\rightarrow$ Information $\rightarrow$ Understanding $\rightarrow$ Wisdom

Introduction

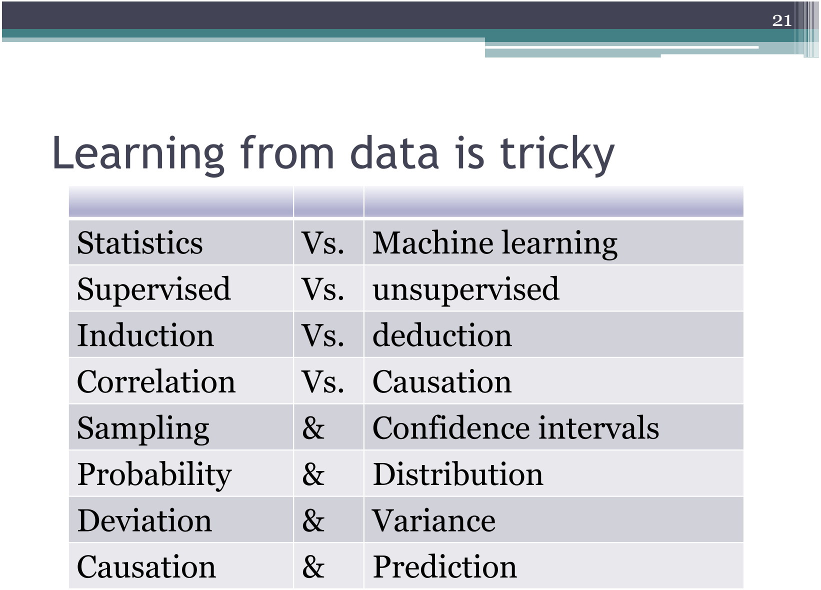
Some data science principles

- Systems are complex
- Data is dirty: deal with it!
- SvOT = LoL! (*)
- Data munging, taming and wrestling $>$ 70% time
- Simplification. Reduction. Distillation.
- Curiosity. Empiricsim. Skepticism.

(Somehano, DataScience London)

(*) SvOT: Single version Of Truth!

Introduction

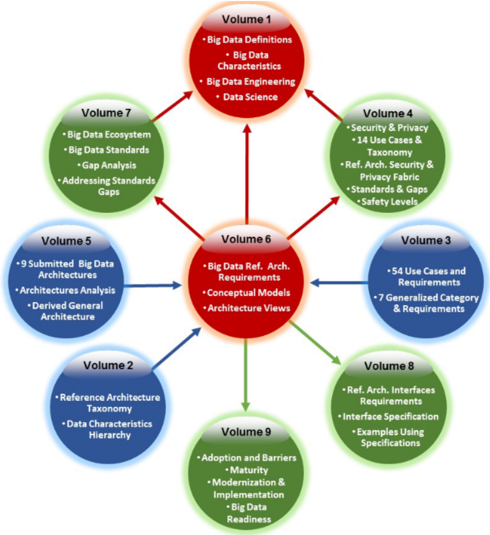


Learning from data is tricky

Statistics	Vs.	Machine learning
Supervised	Vs.	unsupervised
Induction	Vs.	deduction
Correlation	Vs.	Causation
Sampling	&	Confidence intervals
Probability	&	Distribution
Deviation	&	Variance
Causation	&	Prediction

Introduction

National Institute of Standards and Technology (NIST)



Volume 1
- Big Data Definitions
- Big Data Characteristics
- Big Data Engineering
- Data Science

Volume 7
- Big Data Ecosystem
- Big Data Standards
- Gap Analysis
- Addressing Standards Gaps

Volume 4
- Security & Privacy
- 14 Use Cases & Taxonomy
- Ref. Arch. Security & Privacy Fabric
- Standards & Gaps
- Safety Levels

Volume 5
- 9 Submitted Big Data Architectures
- Architectures Analysis
- Derived General Architecture

Volume 6
- Big Data Ref. Arch. Requirements
- Conceptual Models
- Architecture Views

Volume 3
- 54 Use Cases and Requirements
- 7 Generalized Category & Requirements

Volume 2
- Reference Architecture Taxonomy
- Data Characteristics Hierarchy

Volume 8
- Ref. Arch. Interfaces Requirements
- Interface Specification
- Examples Using Specifications

Volume 9
- Adoption and Barriers
- Maturity
- Modernization & Implementation
- Big Data Readiness

Figure 1: NBDIF Documents Navigation Diagram Provides Content Flow Between Volumes

Introduction

The various V's of "V"ig Data (😊)

- **Validity** refers to appropriateness of the data for its intended use.
- Value refers to the inherent wealth, economic and social, embedded in any dataset.
- **Variability** refers to changes in dataset, whether data flow rate, format/structure, semantics, and/or quality that impact the analytics application.
- **Variety** refers to data from multiple repositories, domains, or types.
- **Velocity** refers to the rate of data flow.

Introduction

The various V's of "V"ig Data (😊)

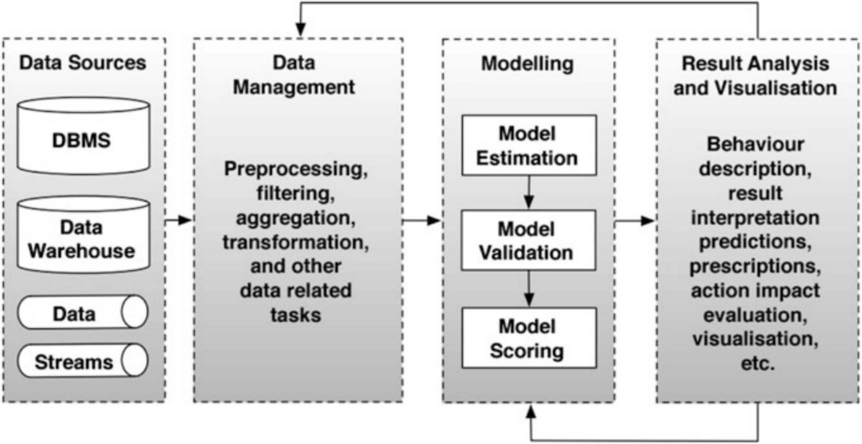
- **Veracity** refers to the accuracy of the data.
- **Vertical Data Scientist** is a subject matter expert in specific disciplines involved in the overall data science process.
- **Vertical scaling** (aka optimization) is the activity to increase data processing performance through improvements to algorithms, processors, memory, storage, or connectivity.
- **Volatility** refers to the tendency for data structures to change over time.
- **Volume** refers to the size of the dataset.

Google says if you have to think about how to manage your data prior to gaining any insight into it, then it's big data.

Data Mining and Machine Learning: recap

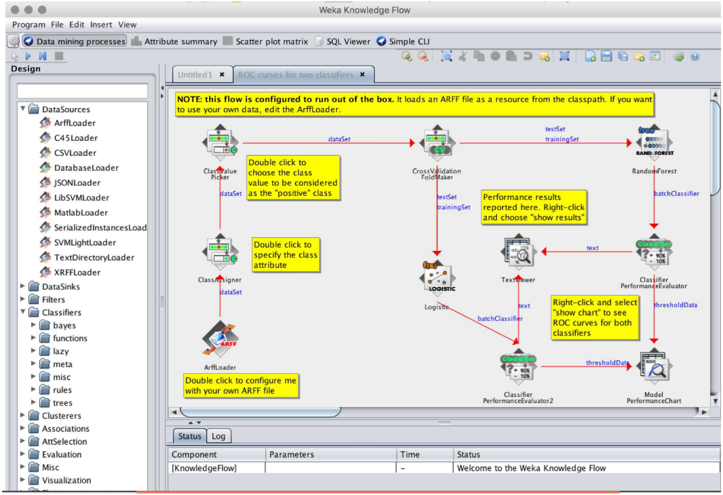
- Workflow (Dataflow - Knowledgeflow):
  - Data preprocessing
    - transformation: normalization, standardization, averaging, median, denoising, filtering
    - preparation: depends on the task, algorithm, package or library being used
  - Machine learning task, algorithm
  - Validation: cross-validation, bootstrapping
- Workflow tools: WEKA KnowledgeFlow, RapidMiner, Orange3, Taverna, Condor DAGMan, Pegasus, Google Dataflow, Google Composer (Apache Airflow)

Data Mining and Machine Learning: workflow


Data Sources
DBMS
Data Warehouse
Data Streams
Data Management
Preprocessing, filtering, aggregation, transformation, and other data related tasks
Modelling
Model Estimation
Model Validation
Model Scoring
Result Analysis and Visualisation
Behaviour description, result interpretation predictions, prescriptions, action impact evaluation, visualisation, etc.

Assunção, M.D., Calheiros, R.N., Bianchi, S., Netto, M.A.S., Buyya, R.: Big data computing and clouds: trends and future directions. J. Parallel Distrib. Comput. 79–80, 3–15 (2015)

Example of workflow in WEKA



NOTE: this flow is configured to run out of the box. It loads an ARFF file as a resource from the classpath. If you want to use your own data, edit the ArffLoader.

Double click to choose the class value to be considered as the "positive" class

Double click to specify the class attribute

Performance results reported here. Right-click and choose "show results"

Right-click and select "show chart" to see ROC curves for both classifiers

Double click to configure me with your own ARFF file

Component	Parameters	Time	Status
[KnowledgeFlow]		-	Welcome to the Weka Knowledge Flow

`java -jar weka.jar` ⇒ KnowledgeFlow

Example of workflow with Orange3

(installation needed, go to https://orange.biolab.si/)

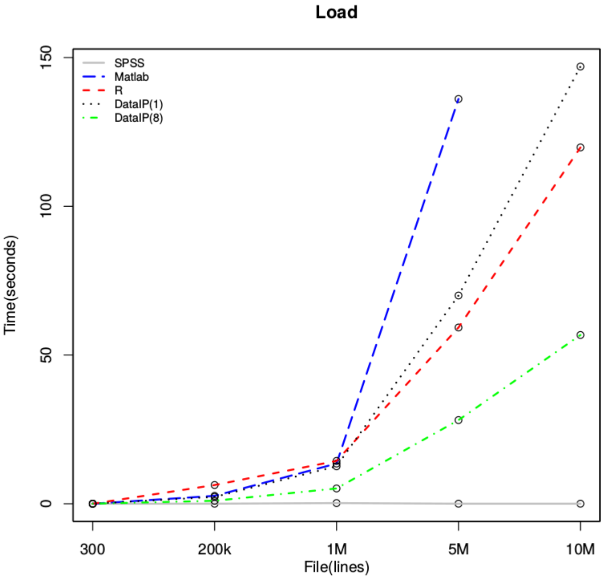
Limitations

- Most systems and tools for data analysis are not scalable
- I/O, memory, computing power

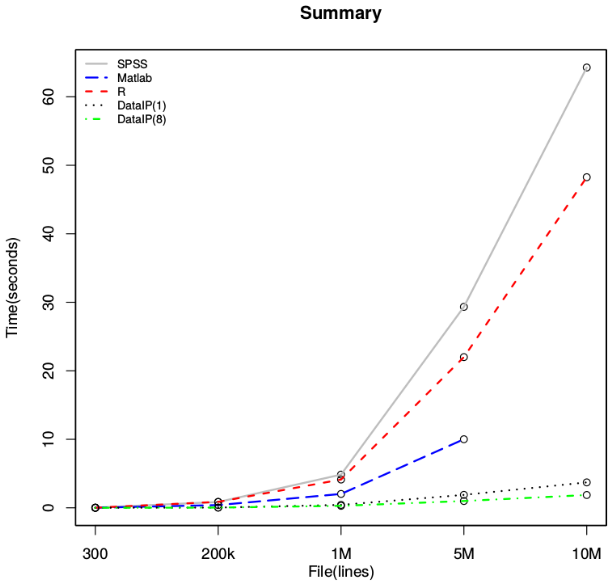
Scalability

- Computational resources: memory, CPU, I/O, storage
- I/O:
    - Experiment 1: SPSS, MatLab, R and DataIP (in-house implementation)
        - dataset of patients, originally 200K entries, 6 numeric variables without nulls
        - varying sizes: 300, 200k, 1M, 5M, 10M
    - Experiment 2: job that needs to fetch data files from a remote site

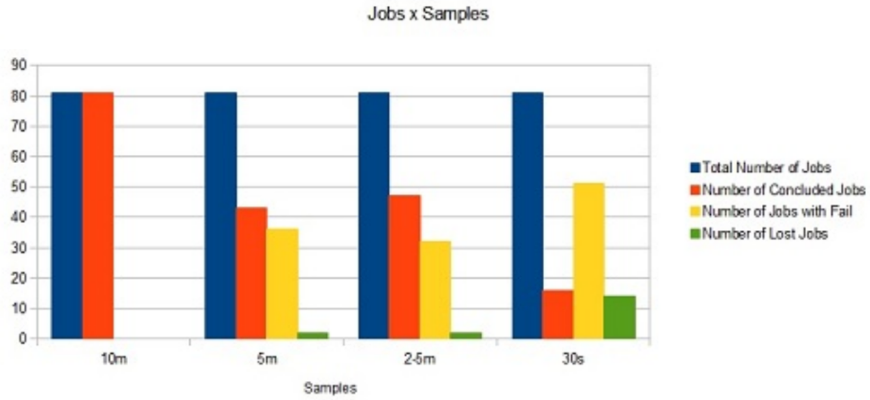
Scalability: Experiment 1, I/O



Scalability: Experiment 1, simple computing: summary



Scalability: Experiment 2, file transfer



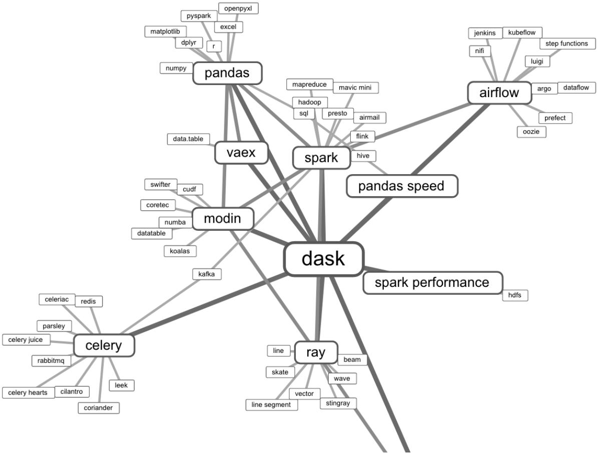
Scalability

- Alternatives
  - break file in multiple smaller files that can be read in parallel: useful if the reading can be done in parallel
  - undersampling: need to be careful about data distribution
  - use of specific hardware and software: distributed disks, distributed file systems, distributed databases, in-memory databases, parallel and distributed software
  - work with compressed files: zip, parquet, CSR, CSR5 (for sparse matrices) etc

Scalability

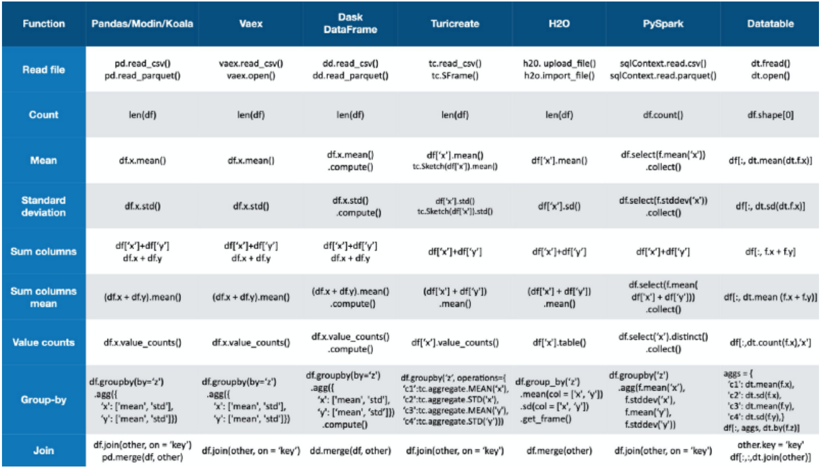
- Python libraries normally used to handle large scale data:
  - Joblib
  - Dask
  - Modin
  - Vaex
  - Koalas
  - Ray
  - Rapids-AI
  - ...

Alternative libraries for big data



source: `https://www.datarevenue.com/en-blog/pandas-vs-dask-vs-vaex-vs-modin-vs-rapids-vs-ray`

Differences in libraries



Function	Pandas/Modin/Koala	Vaex	Dask DataFrame	Turicreate	H2O	PySpark	Datatable

Read file	pd.read_csv()
	pd.read_parquet()
		vaex.read_csv()
		vaex.open()
			dd.read_csv()
			dd.read_parquet()
				tc.read_csv()
				tc.SFrame()
					h20. upload_file()
					h2o.import_file()
						sqlContext.read.csv()
						sqlContext.read.parquet()
							dt.fread()
							dt.open()

Count	len(df)	len(df)	len(df)	len(df)	len(df)	df.count()	df.shape[0]

Mean	df.x.mean()	df.x.mean()	df.x.mean()
			.compute()
				df['x'].mean()
				tc.Sketch(df['x']).mean()
					df['x'].mean()	df.select(f.mean('x'))
						.collect()
							df[:, dt.mean(dt.f.x)]

Standard deviation	df.x.std()	df.x.std()	df.x.std()
			.compute()
				df['x'].std()
				tc.Sketch(df['x']).std()
					df['x'].sd()	df.select(f.stddev('x'))
						.collect()
							df[:, dt.sd(dt.f.x)]

Sum columns	df['x']+df['y'] df.x + df.y	df['x']+df['y'] df.x + df.y	df['x']+df['y'] df.x + df.y	df['x']+df['y']	df['x']+df['y']	df['x']+df['y']	df[:, f.x + f.y]
Sum columns mean	(df.x + df.y).mean()	(df.x + df.y).mean()	(df.x + df.y).mean() .compute()	(df['x'] + df['y']) .mean()	(df['x'] + df['y']) .mean()	df.select(f.mean(df['x'] + df['y'])) .collect()	df[:, dt.mean (f.x + f.y)]

Value counts	df.x.value_counts()	df.x.value_counts()	df.x.value_counts()
			.compute()
				df['x'].value_counts()	df['x'].table()	df.select('x').distinct()
						.collect()
							df[:,dt.count(f.x),'x']

Group-by	df.groupby(by='z')
	.agg({
	'x': ['mean', 'std'],
	'y': ['mean', 'std']})
		df.groupby(by='z')
		.agg({
		'x': ['mean', 'std'],
		'y': ['mean', 'std']})
			df.groupby(by='z')
			.agg({
			'x': ['mean', 'std'],
			'y': ['mean', 'std']})
			.compute()
				df.groupby('z', operations={
				'c1':tc.aggregate.MEAN('x'),
				'c2':tc.aggregate.STD('x'),
				'c3':tc.aggregate.MEAN('y'),
				'c4':tc.aggregate.STD('y')})
					df.group_by('z')
					.mean(col = ['x', 'y'])
					.sd(col = ['x', 'y'])
					.get_frame()
						df.groupby('z')
						.agg(f.mean('x'),
						f.stddev('x'),
						f.mean('y'),
						f.stddev('y'))
							aggs = {
							'c1': dt.mean(f.x),
							'c2': dt.sd(f.x),
							'c3': dt.mean(f.y),
							'c4': dt.sd(f.y),}
							df[:, aggs, dt.by(f.z)]

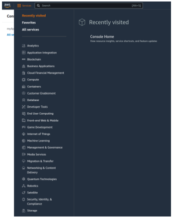
Join	df.join(other, on = 'key')
	pd.merge(df, other)
		df.join(other, on = 'key')	dd.merge(df, other)	df.join(other, on = 'key')	df.merge(other)	df.join(other, on = 'key')	other.key = 'key'
							df[:,:,dt.join(other)]

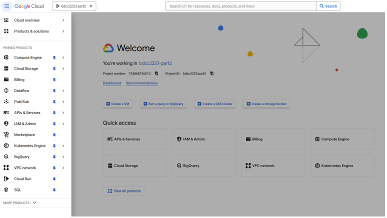
source: `https://towardsdatascience.com/`

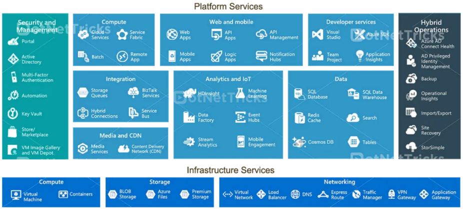
`beyond-pandas-spark-dask-vaex-and-other-big-data-technologies-battling-head-to-head-a453a1f8cc13`

Scalability

The need for clouds... (???)







Scalability

Amazon AWS, Azure and Google Cloud platform most popular, but many others...

- public: like public utilities (water, electricity etc). Operate at a scale far larger than any private cloud, can offer a broad set of powerful features:
  - elasticity
  - fine-grained billing
  - high reliability due to geographic distribution
  - wide variety of resource types
  - rich sets of platform services
  - equally importantly, they can achieve substantial economies of scale
- private: operated by a private institution or individual to provide computing, storage, and/or other services to a more limited audience.
- hybrid: may be used to run selected tasks on public clouds: a process termed cloud bursting
- open source: OpenStack, OpenNebula etc.
- community x academic

Other taxonomy

- SaaS: Software as a Service or Storage as a Service
- PaaS: Platform as a Service
- IaaS: Infrastructure as a Service

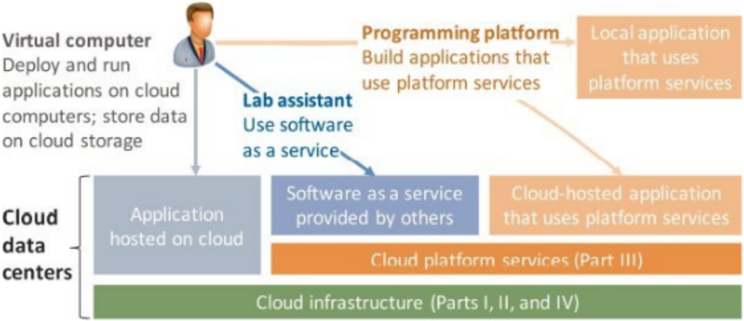
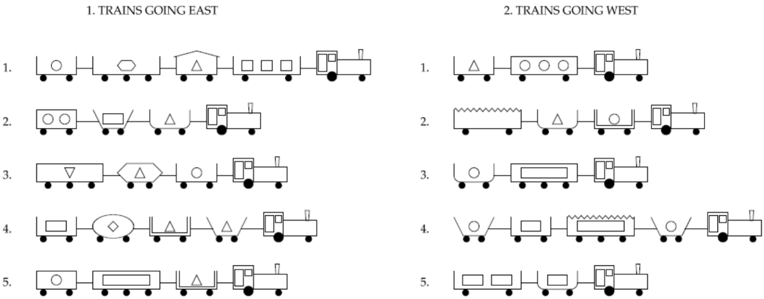

Virtual computer
Deploy and run applications on cloud computers; store data on cloud storage
Lab assistant
Use software as a service
Programming platform
Build applications that use platform services
Local application that uses platform services
Cloud data centers
Application hosted on cloud
Software as a service provided by others
Cloud-hosted application that uses platform services
Cloud platform services (Part III)
Cloud infrastructure (Parts I, II, and IV)

Figure 1.1: Scientists can use clouds in three distinct ways: As a source of on-demand computing and storage on which to run their own software (left); as a source of software that can be run over the network (center) as a source of new platform capabilities that can allow development of new types of software (right).

Example: Michalski's trains



Representation in a single table

Usual: aggregate all tables in only one!

Patient	Location	Size	Date	Calcifications
P1	C	0.1	20050403	F, A
P1	C	0.2	20060412	F
P1	9	0.1	20060412	A
P2	12	0.3	20050415	M
...	...	...	...	...