

Big data science

Day 1

F. Legger - INFN Torino

<https://github.com/leggerf/MLCourse-2022>



Course plan

- **Mon 2nd** : introduction to big data
- **Tu 3rd - Th 5th** OpenMP and Novel architectures, theory+hands-on
- **Fri 6th - Tue 10th**: Machine and Deep learning, theory + hands-on
- **Wed 11:** free for hands-on/questions/discussion/leftovers
- **Th 12:** 2-hour hands-on and final test
 - Final notebook (a summary of all hands-on sessions)
 - OpenMP exercises

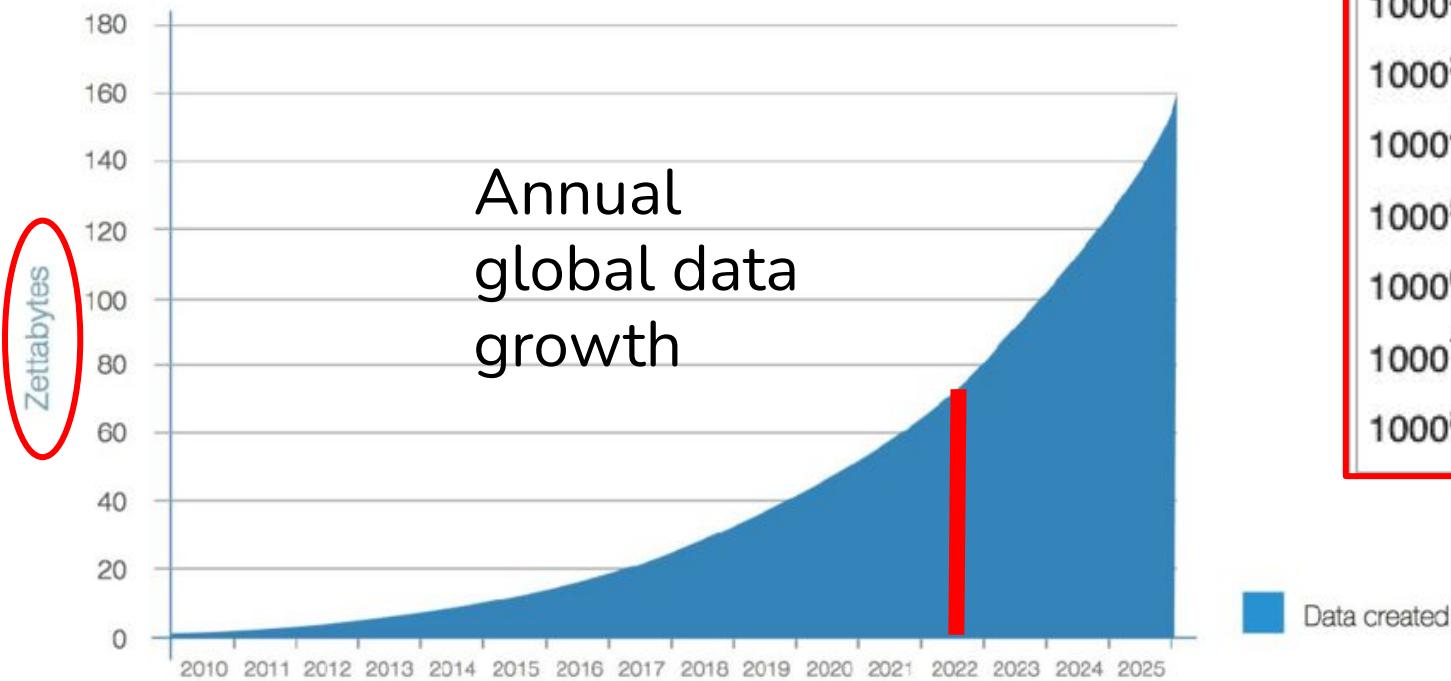
Today

- **Introduction to big data**
 - Definition, applications sources
 - **The big data pipeline**
 - Infrastructure, technologies
 - **Analytics**
 - Data mining, data structures



What is big data?

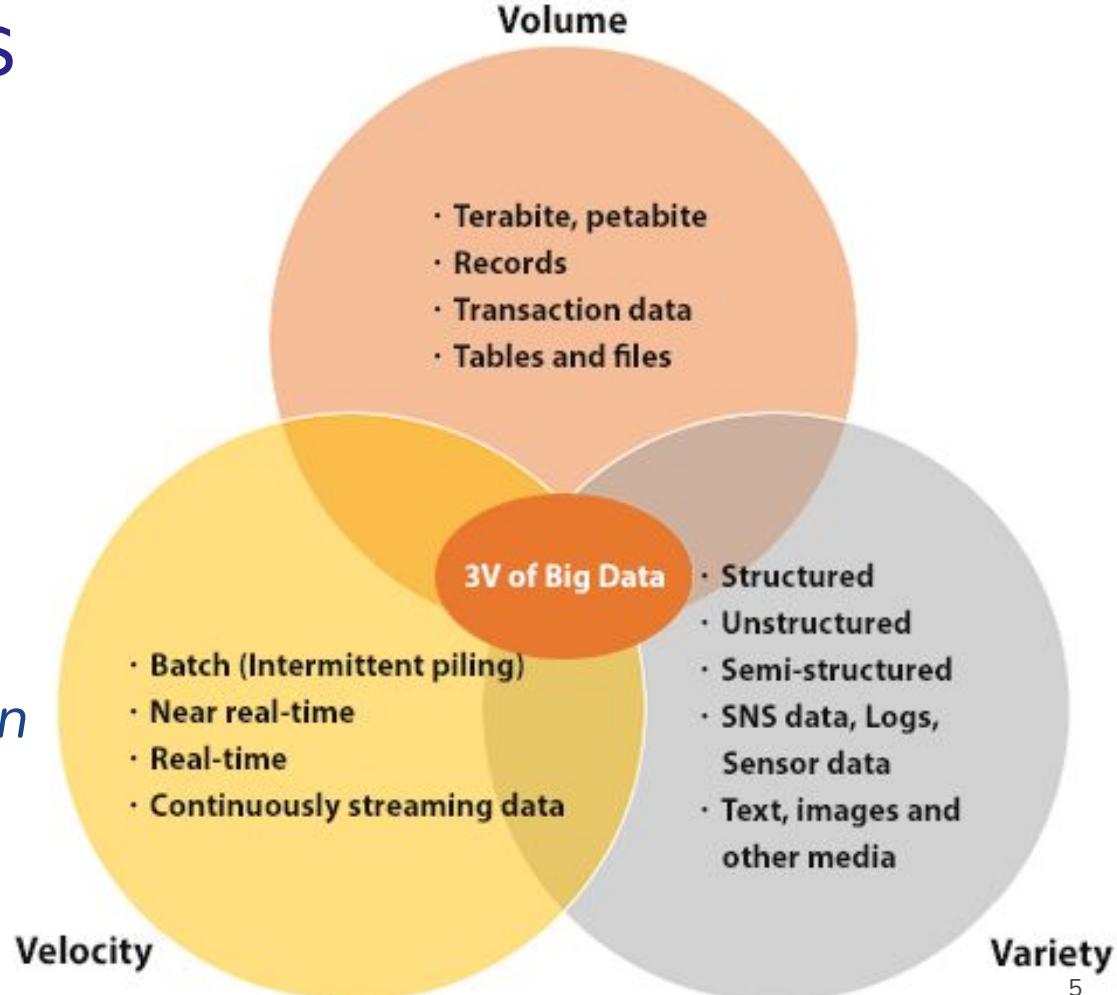
- Data that is too big to be analysed traditionally



Decimal	
Value	Metric
1000	kB kilobyte
1000^2	MB megabyte
1000^3	GB gigabyte
1000^4	TB terabyte
1000^5	PB petabyte
1000^6	EB exabyte
1000^7	ZB zettabyte
1000^8	YB yottabyte

It's all about Vs

“Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.” Gartner (2012)

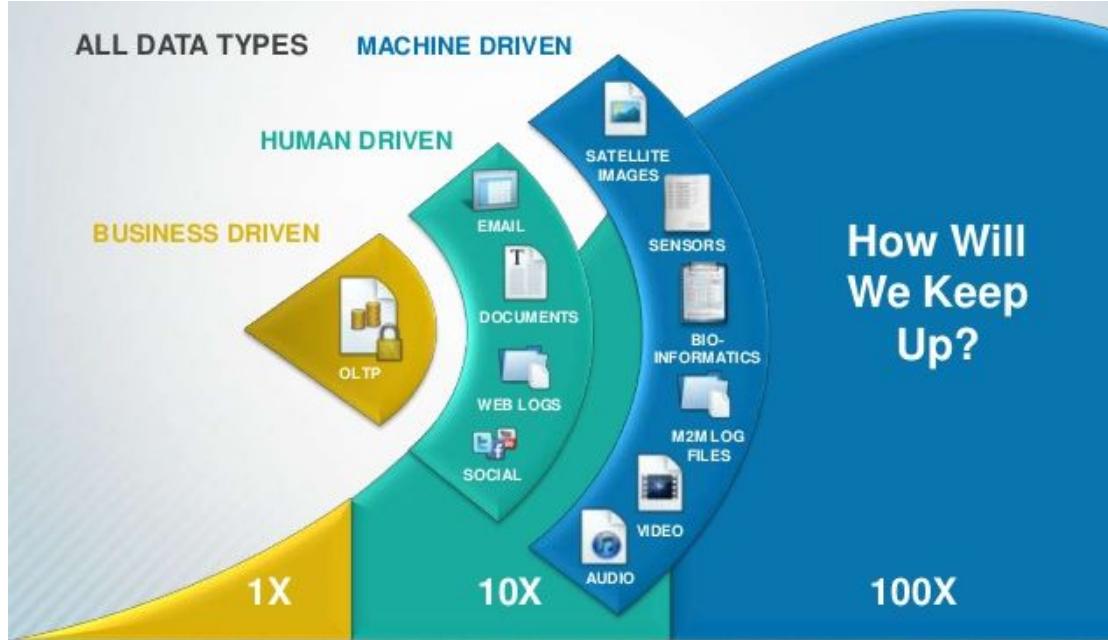


Even more Vs

- Veracity
- Variability
- Visualization
- Validity
- Vulnerability
- Volatility
- Value

<https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx>

Big data sources: Business



- Traditional Business Systems
 - Commercial transactions
 - Banking/stock records
 - E-commerce
 - Credit cards
 - Medical records

Big data sources: Human (x10)



- **Social Networks**
 - Twitter and Facebook
 - Blogs and comments
 - **Video conferences, Zoom, Teams**
- Videos: YouTube, Netflix
- Internet searches
- **Deliveries**
- User-generated maps
- E-Mail

Big data sources: Machine (x100)



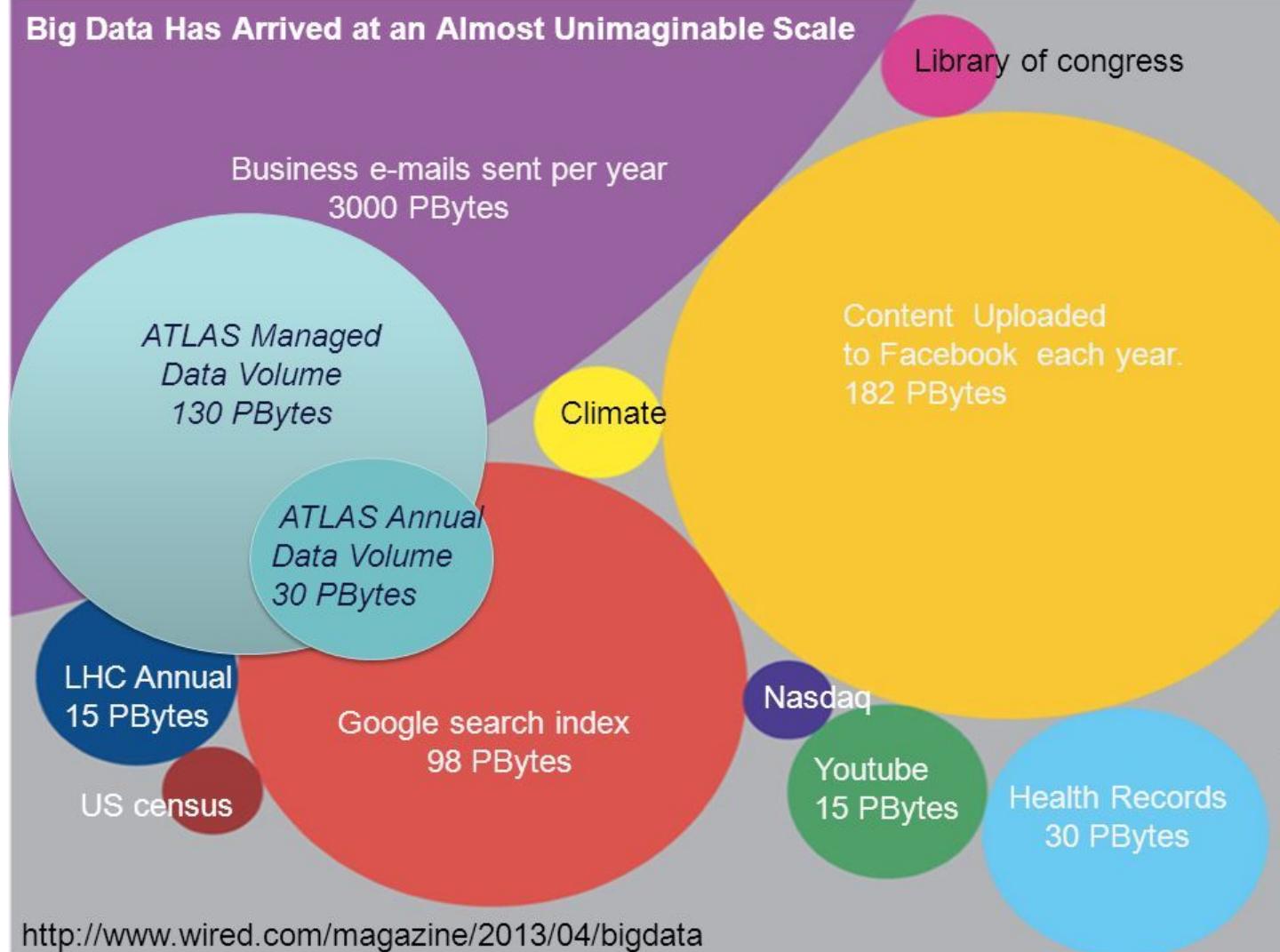
- **Internet of Things (IoT)**
 - Sensors: traffic, weather, mobile phone location, etc.
 - Security, surveillance videos, and images
 - Satellite images
 - Data from computer systems (logs, web logs)

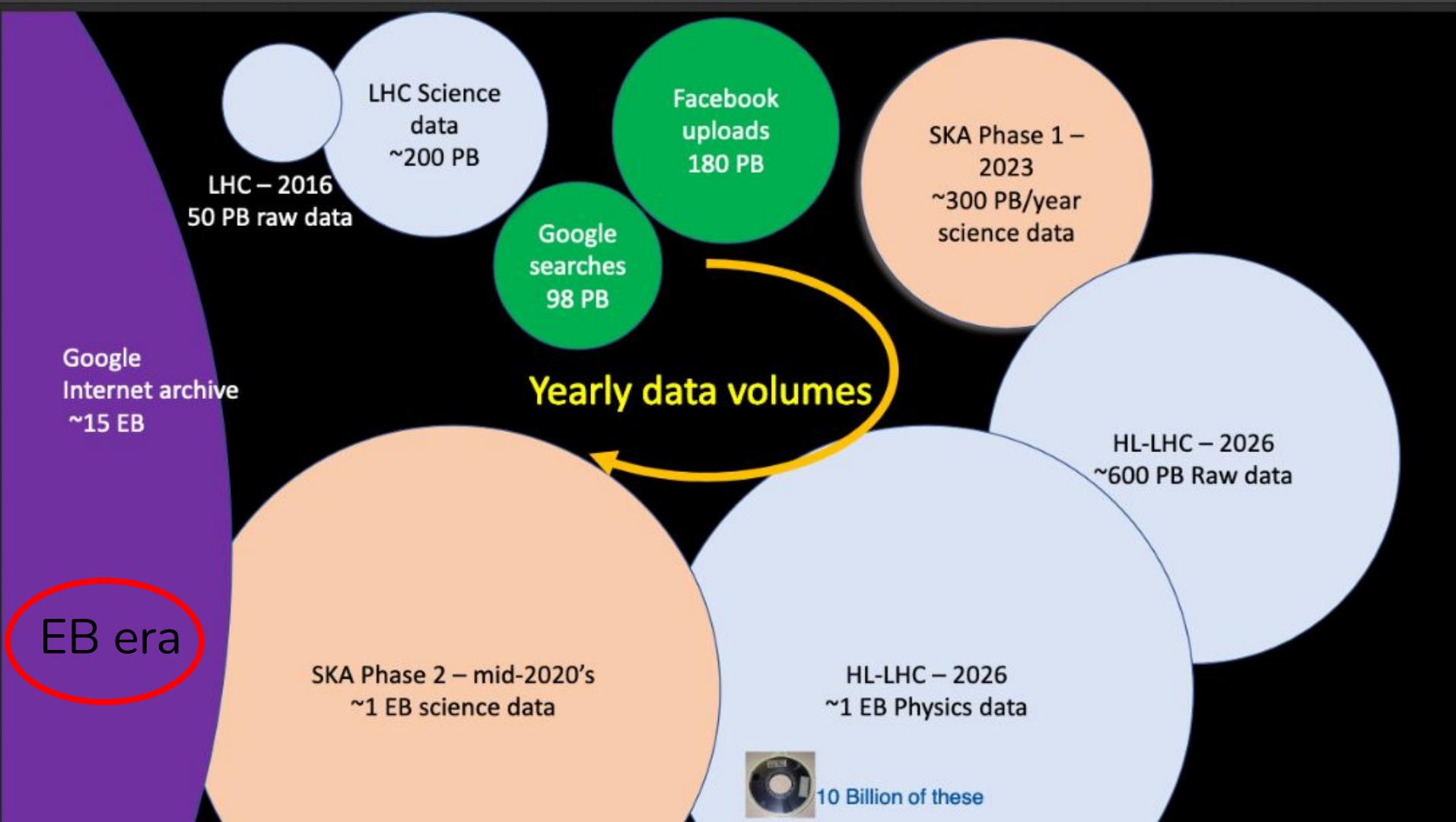
Big data applications

- **Entertainment:** Netflix and Amazon make shows and movie recommendations to their users.
- **Insurance:** predict illness, accidents and price their products accordingly.
- **Driverless Cars:** Google's driverless cars collect about 1 GB/s.
- **Automobile:** Rolls Royce fits hundreds of sensors into its engines and propulsion systems. Real time data are used to schedule maintenance.
- **Government:** analyse patterns and influence election results (Cambridge Analytica Ltd.), people movement during lockdown

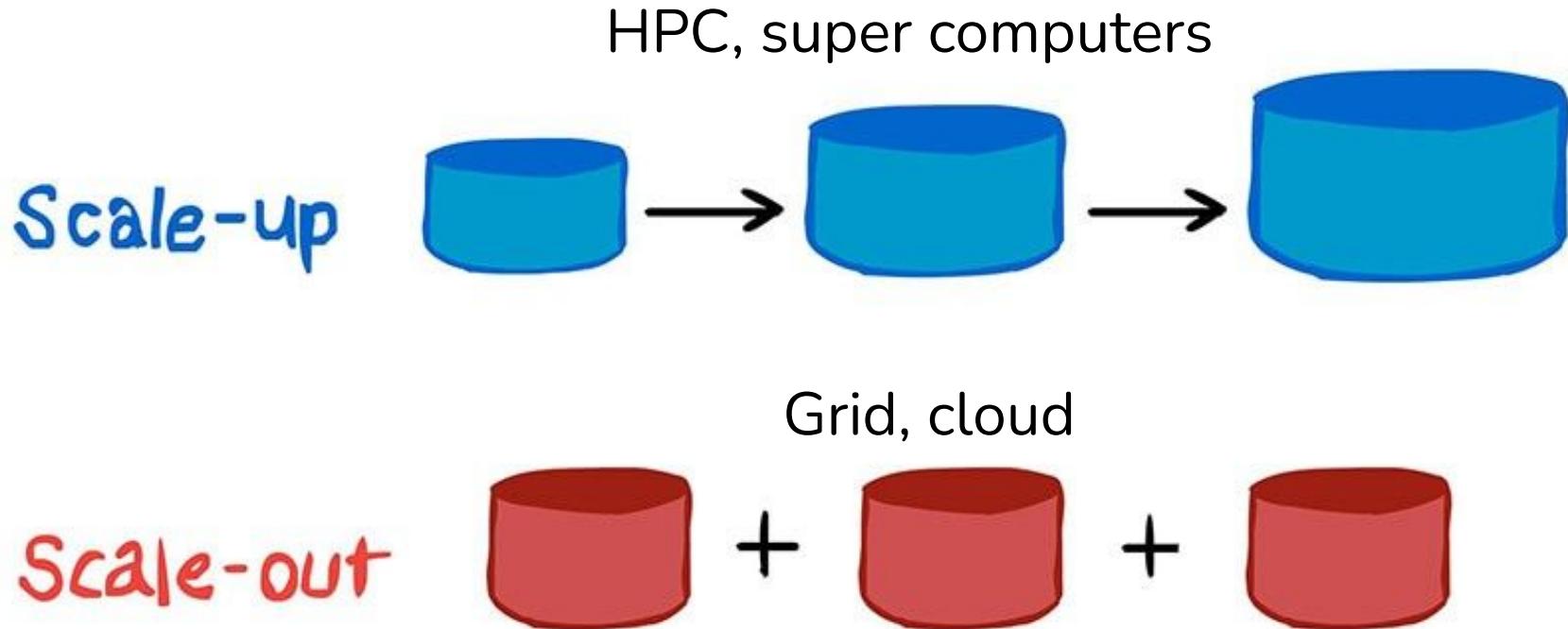
Big Data Has Arrived at an Almost Unimaginable Scale

2013,
PB era



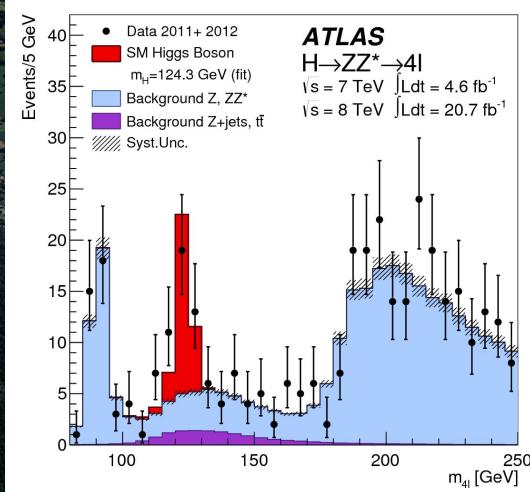


Scale up vs scale out





- pp (or Pb-Pb) collisions
- 4 experiments (ATLAS, CMS, LHCb, ALICE)
- Discovery of Higgs boson
- Nobel prize for physics 2013



Big data @LHC

- ATLAS/CMS: 100-megapixel digital cameras that take 40 million “pictures” per second == **40 TB raw data/second**
- Trigger system (real time): “empty” pictures immediately thrown away: **1GB/second**
- Only **one in a billion** is an Higgs boson!
- Data stored, processed and analysed using the **Worldwide LHC Computing Grid (WLCG)**

World LHC Computing Grid



US Dept of State Geographer

© 2013 Google

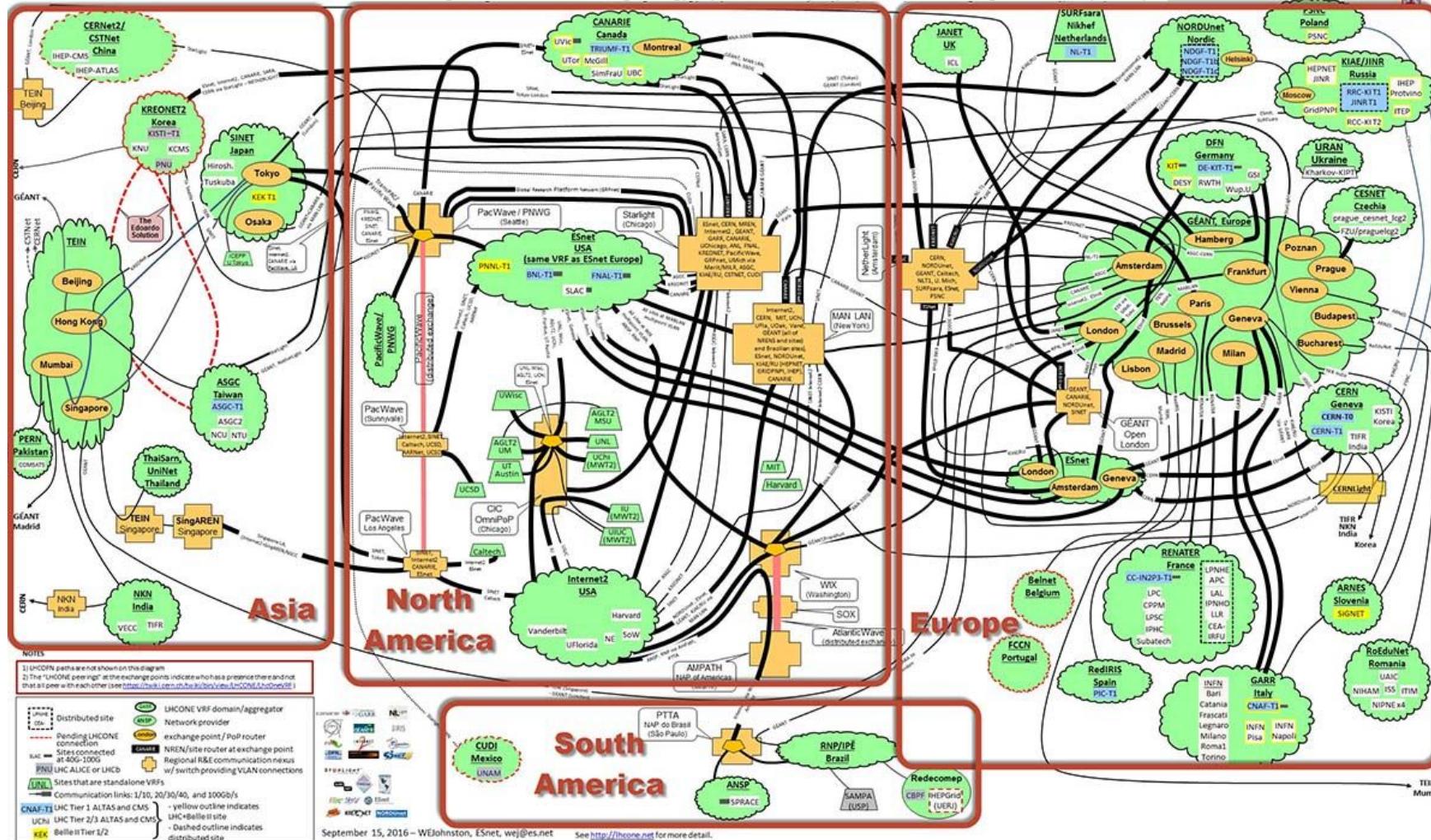
Image Landsat

Data SIO, NOAA, U.S. Navy, NGA, GEBCO

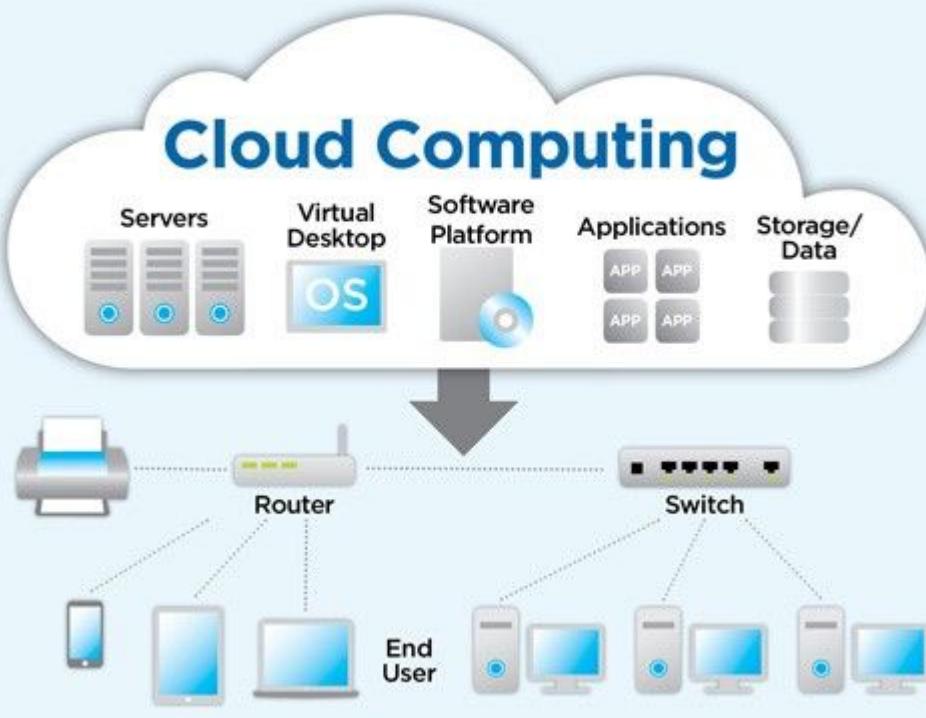
Google earth

- 42 countries
- 170 computing centres
- Over 2 million tasks daily
- 1 million computer cores
- 1 exabyte of storage

LHCONE L3VPN: A global infrastructure for High Energy Physics data analysis (LHC, Belle II, Pierre Auger Observatory, NOvA)



Cloud computing



On-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user

1. Resources Pooling
2. On-Demand Self-Service
3. Easy Maintenance
4. Large Network Access
5. Availability
6. Automatic System
7. Economical
8. Security
9. Pay as you go (commercial)
10. Measured Service

Edge and fog computing

INDUSTRIAL IoT DATA PROCESSING LAYER STACK

CLOUD LAYER

Big Data Processing
Business Logic
Data Warehousing

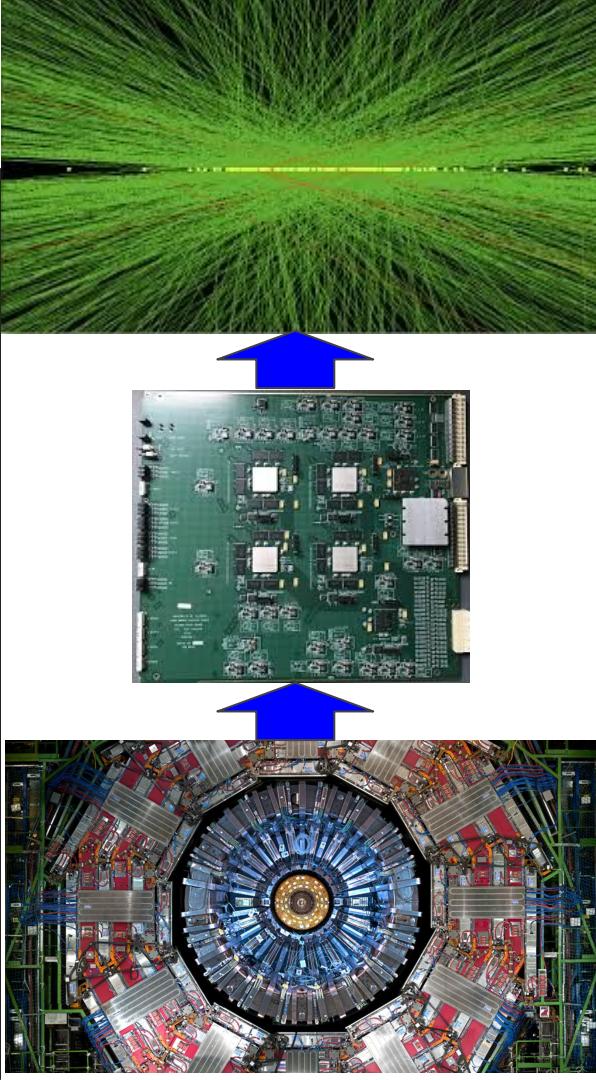
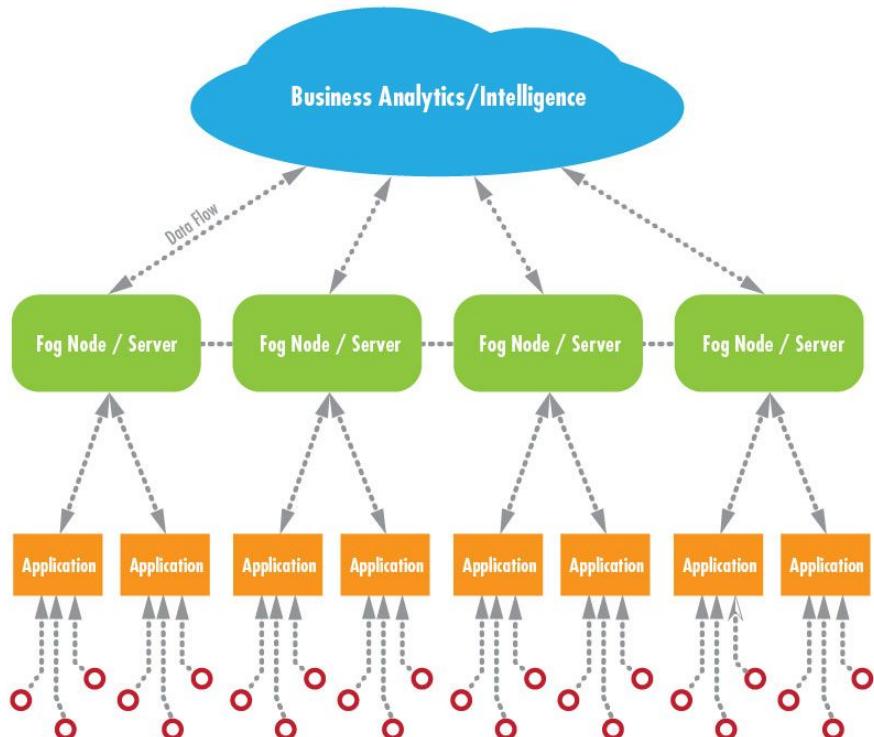
FOG LAYER

Local Network
Data Analysis & Reduction
Control Response
Virtualization/Standardization

EDGE LAYER

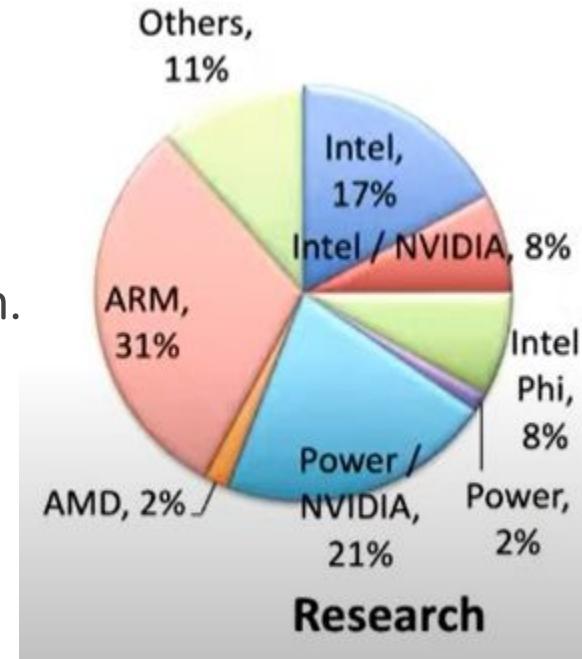
Large Volume Real-time Data Processing
At Source/On Premises Data Visualization
Industrial PCs
Embedded Systems
Gateways
Micro Data Storage

Sensors & Controllers (data origination)



High Performance Computing (HPC)

- Typically involves supercomputers
- Top 500 #1: Fugaku
 - A64FX ARM v8.2-A,
 - Scalable Vector Extension (SVE) instructions and a 512-bit implementation.
 - on-die Tofu-D network BW (~400 Gbps)



Rank	System	Cores	Rmax [TFlop/s]	Rpeak [TFlop/s]	Power (kW)
1	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	537,212.0	29,899
2	Summit - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	Sierra - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China				
5	Perlmutter - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States				

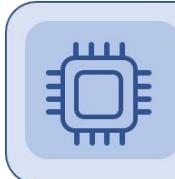
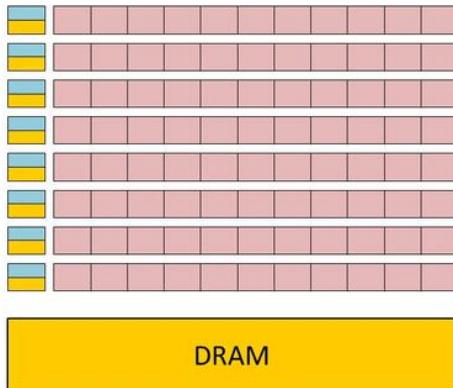
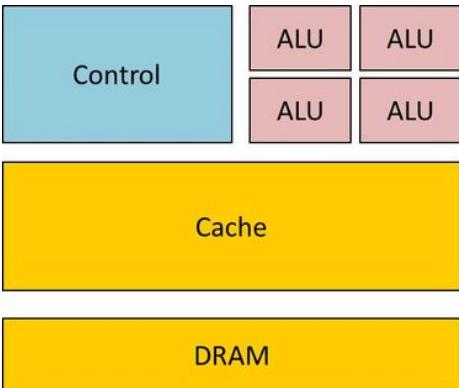
Top European HPCs

8	JUWELS Booster Module - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, Atos Forschungszentrum Juelich (FZJ) Germany	449,280	44,120.0	70,980.0	1,764
9	HPC5 - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, Dell EMC Eni S.p.A. Italy	669,760	35,450.0	51,720.8	2,252

Heterogeneous architectures

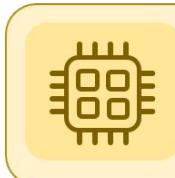
CPU vs GPU

- few very complex cores
 - single-thread performance optimization
 - transistor space dedicated to complex ILP
 - few die surface for integer and fp units
- hundreds of simpler cores
 - thousand of concurrent hardware threads
 - maximize floating-point throughput
 - most die surface for integer and fp units



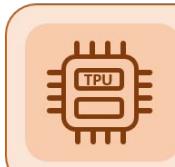
CPU

- Small models
- Small datasets
- Useful for design space exploration



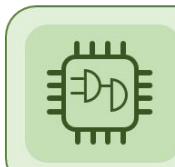
GPU

- Medium-to-large models, datasets
- Image, video processing
- Application on CUDA or OpenCL



TPU

- Matrix computations
- Dense vector processing
- No custom TensorFlow operations



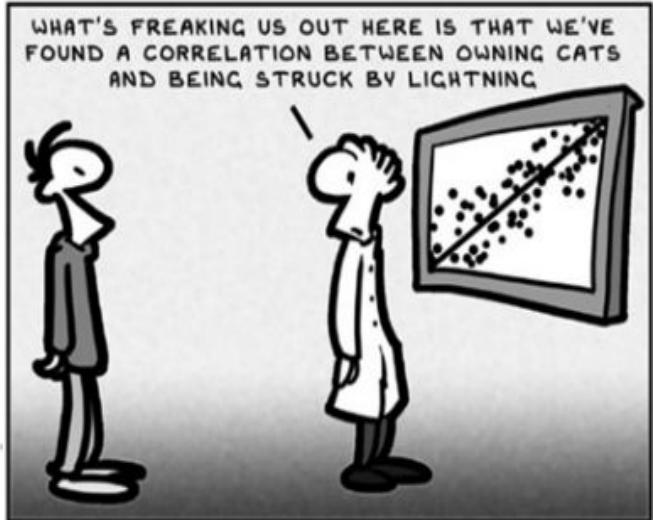
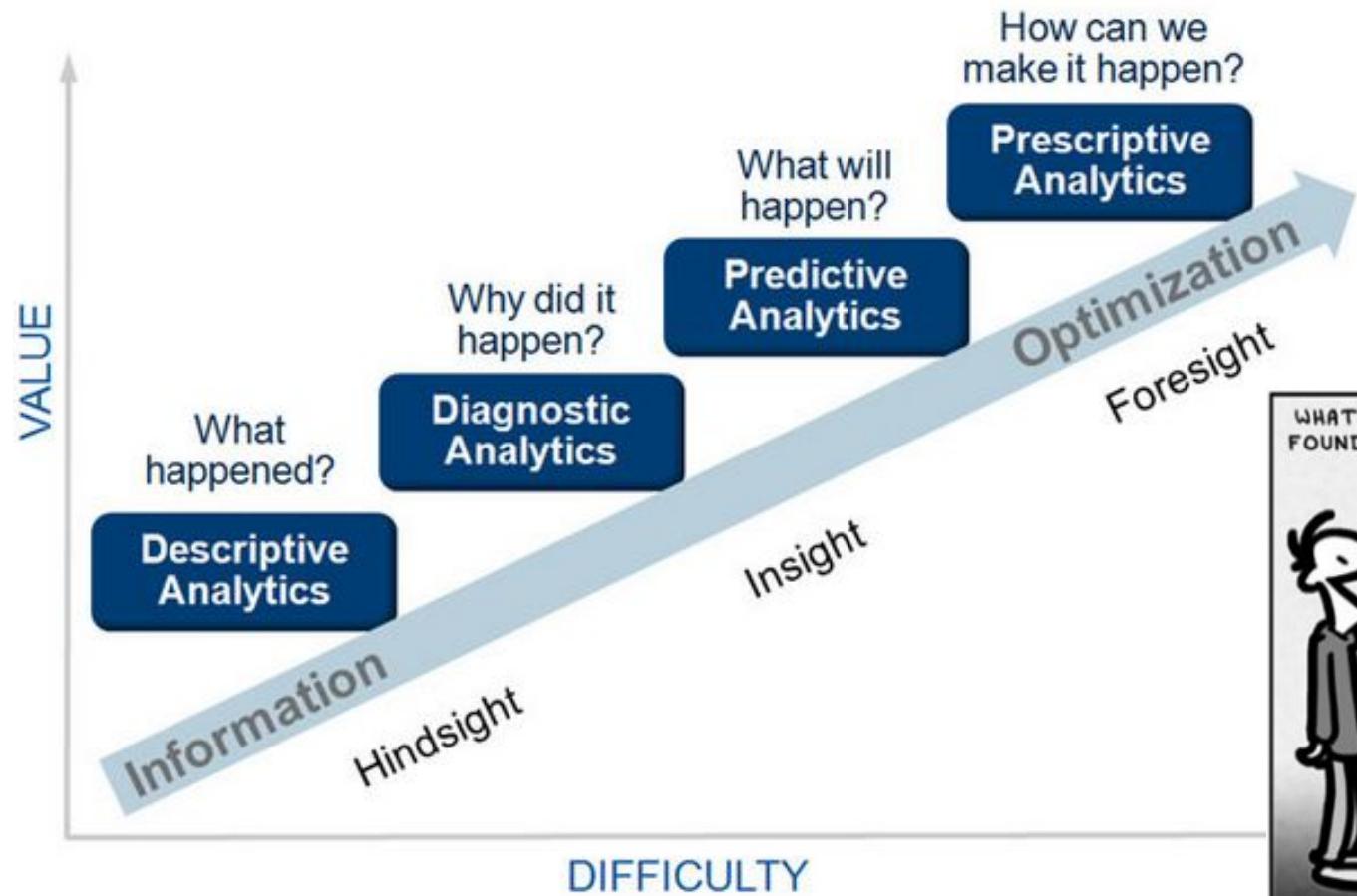
FPGA

- Large datasets, models
- Compute intensive applications
- High performance, high perf./cost ratio

TPU: Tensorflow Processing Unit

FPGA: Field Programmable Gate Array

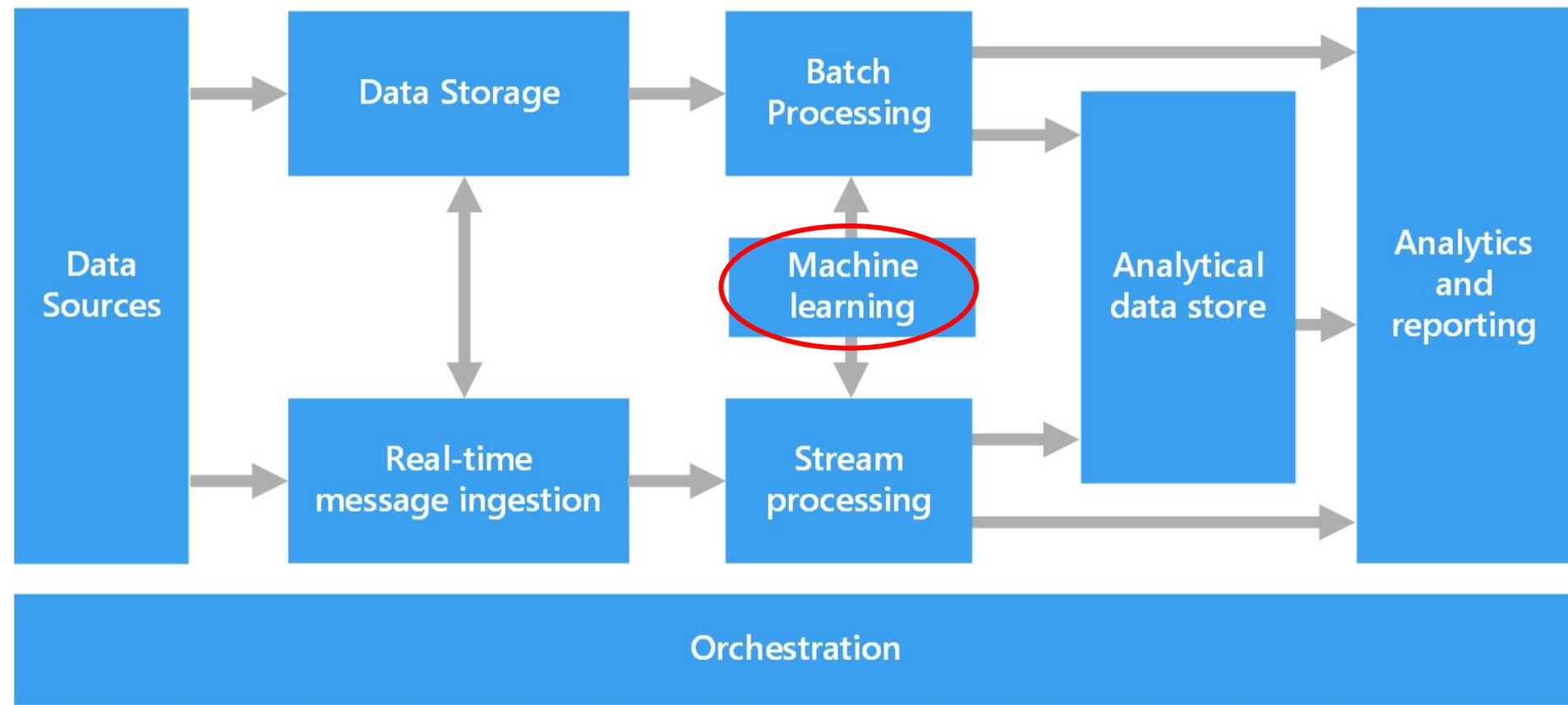
Analytics



Analytics applications

- **Churn prediction**
 - Customers switching from one company to another
- **Recommendation system**
 - Netflix reported that **2/3** of the movies watched are recommended
 - Google News stated that recommendations generate **38%** more click-through
 - Amazon claimed that **35%** sales come from recommendations
- **Sentiment analysis**
- **Operational analytics**
 - Automatization
- **Medicine**
 - Remote diagnosis, prevention

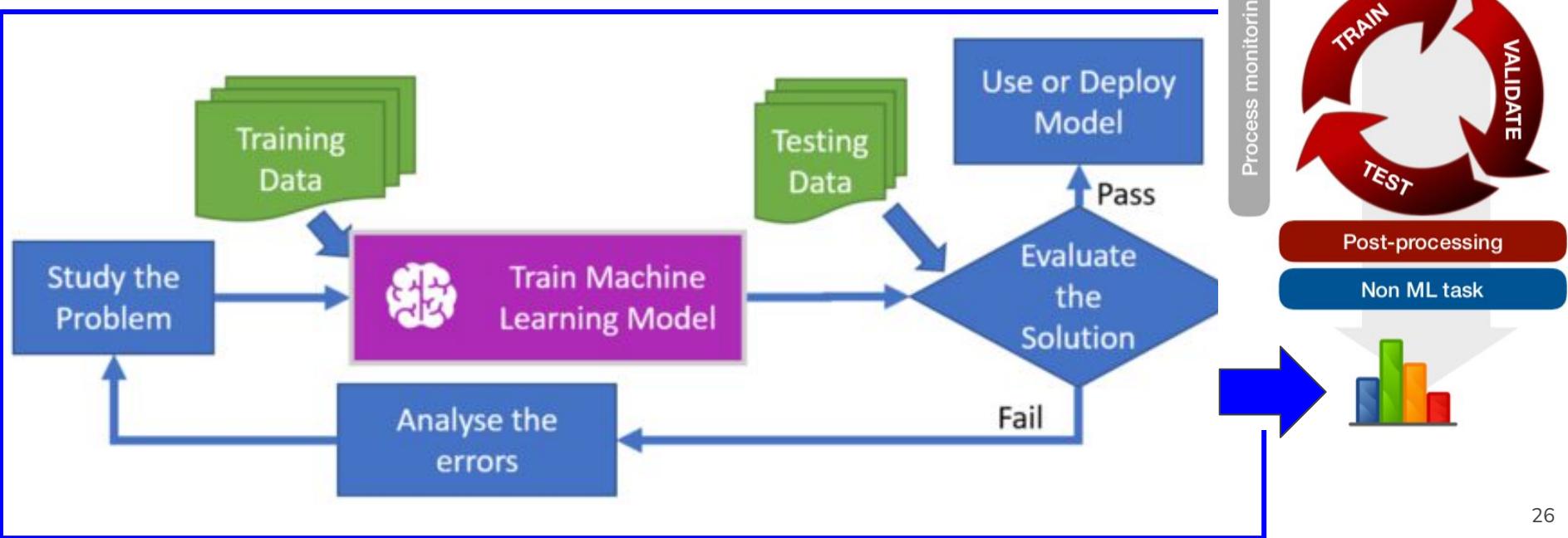
The big data pipeline

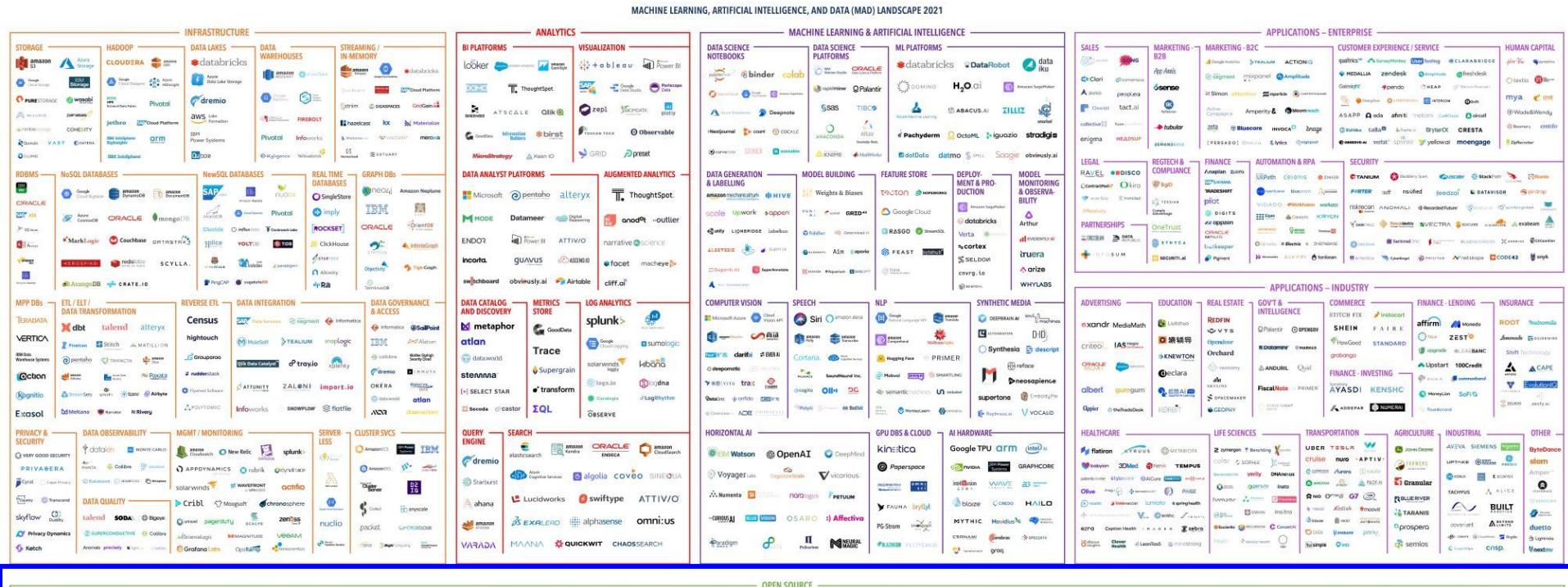


The machine learning pipeline

Input data is typically split into:

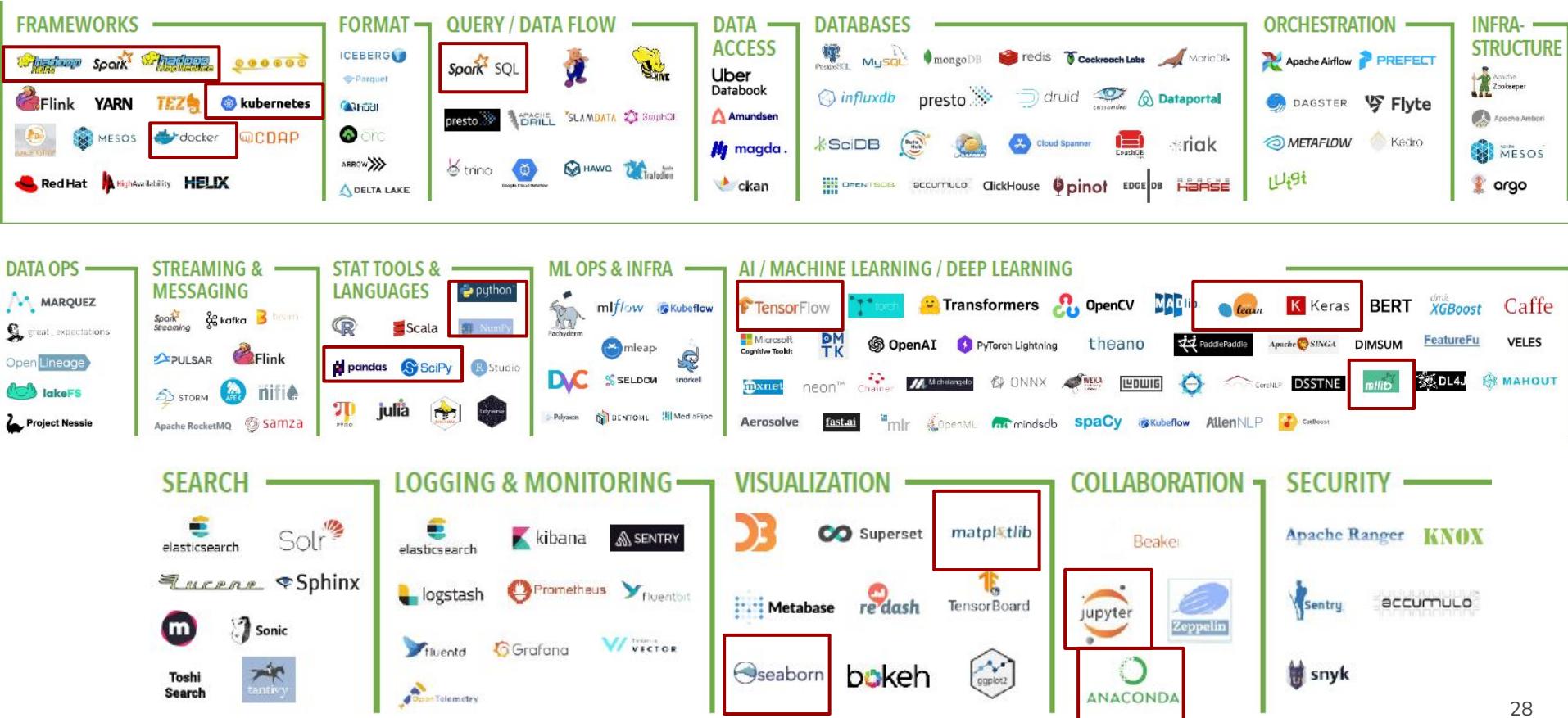
- Training dataset
- Testing dataset





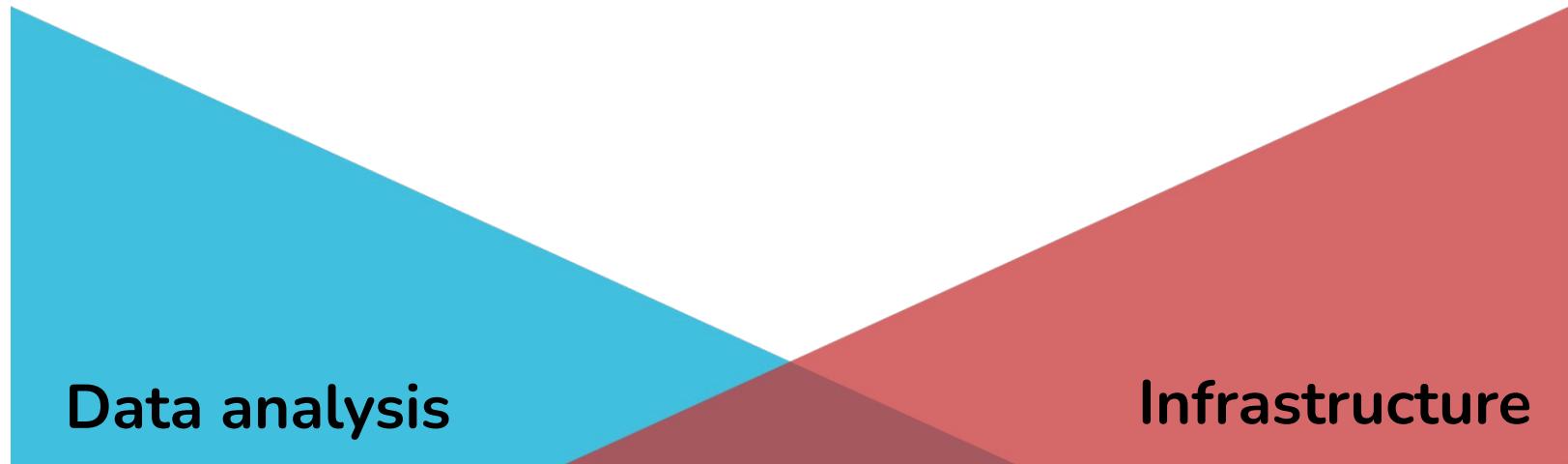
FIRSTMARK

Open Source



Data Scientist

Data Engineer



↑
Core Competencies

Adv. Math/Statistics

ML/AI

Adv. Analytics

↑
Overlapping Skills

Analysis

Programming

Big Data

↑
Core Competencies

Adv. Programming

Distributed Sys.

Data Pipelines

Data Scientist

Machine Learning Engineer

Data Engineer



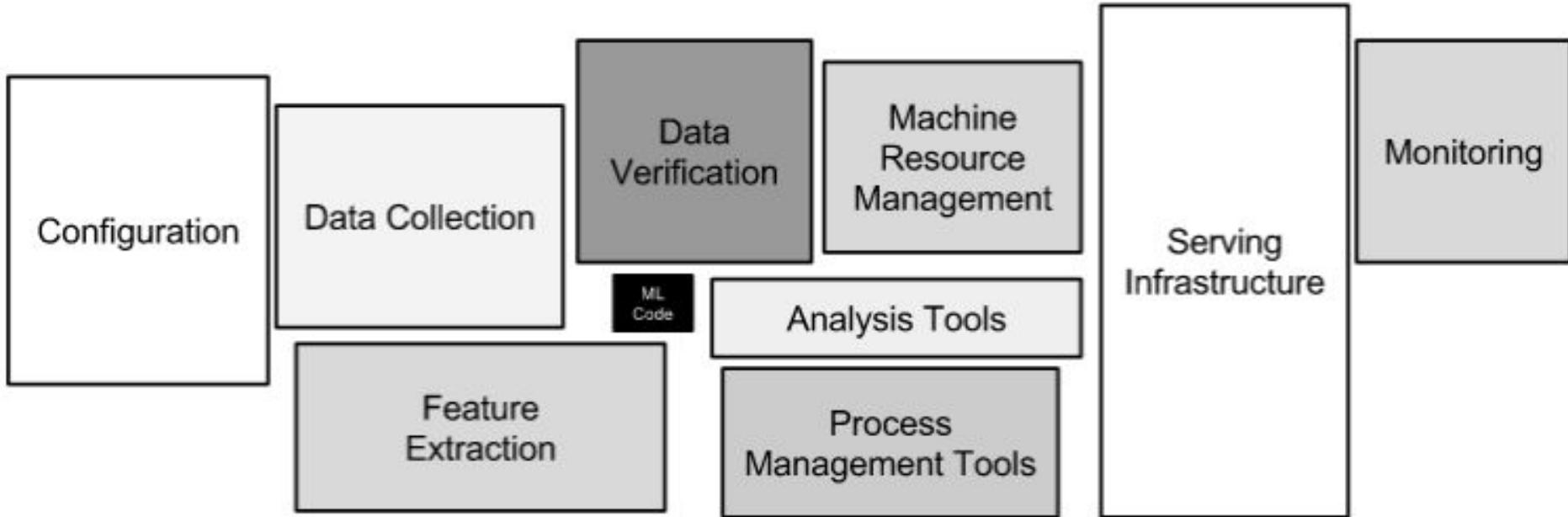
↑
Research ML/AI
Adv. Analytics

↑
Operationalizing ML
Optimizing ML

↑
Adv. Programming
Distributed Sys.

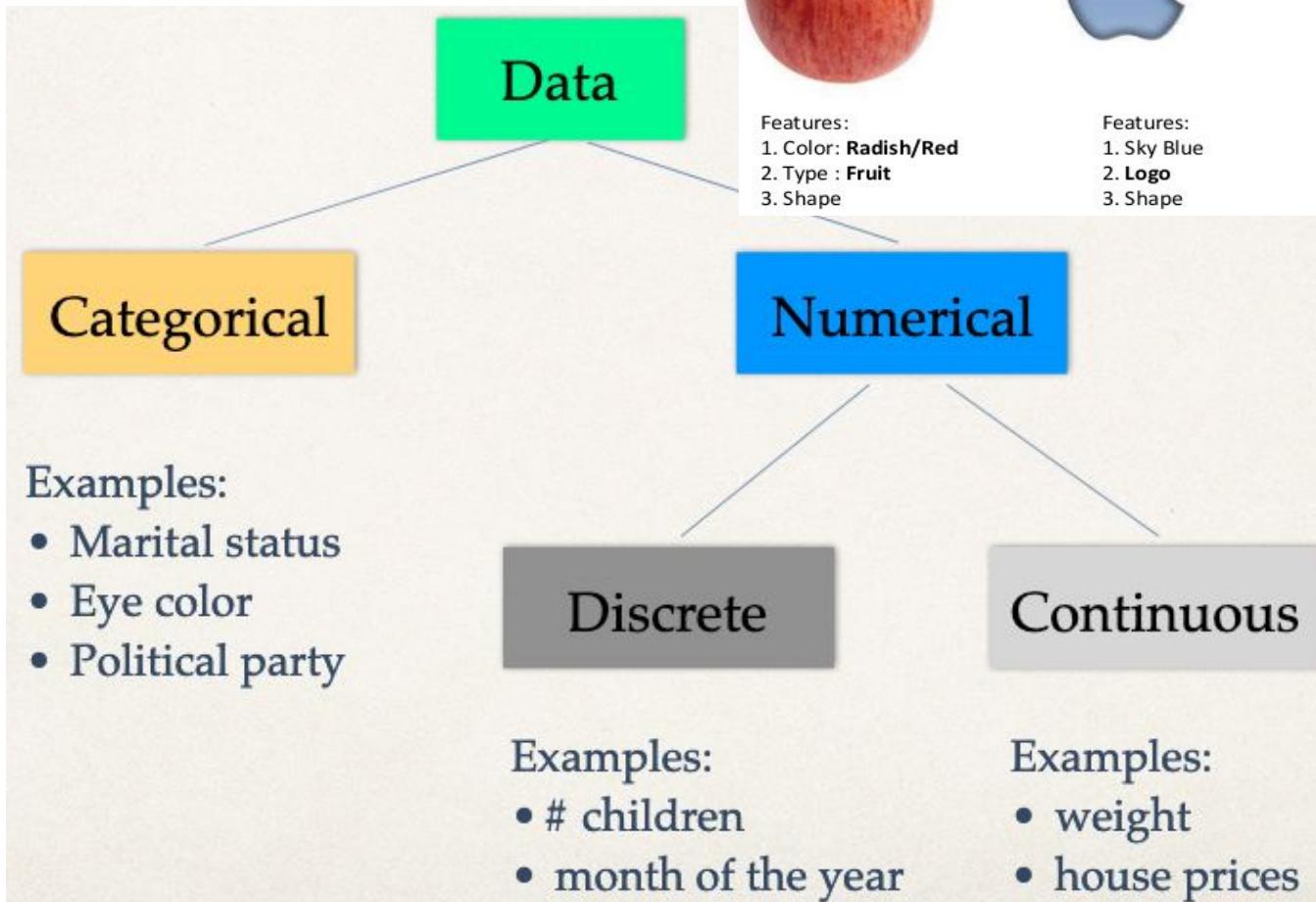
Typical ML workflow

Where is ML?



Data types

Typically
strings



Data preprocessing

- Most of the time will be spent in this step
- Data clean-up, data transformation, feature engineering
 - **data transformation**
 - scaling and normalization
 - encoding, aggregation features, log-transformation (to remove outliers)
 - **data visualization, exploration**
 - **data augmentation, bucketing, binning, ...**
 - **dimensionality reduction**
- Your programming skills will be required here: **R, Python, ...**

Data transformation

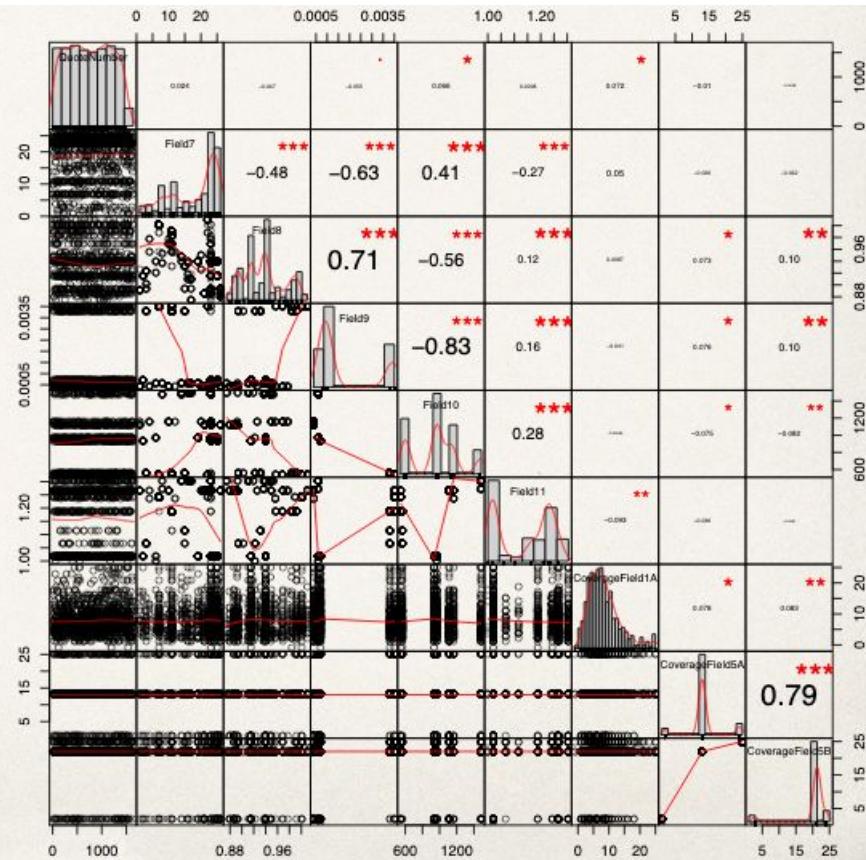
$$x' = \frac{x - \bar{x}}{\sigma}$$

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Data transformation and aggregation: log, sum of values, average, ...
- Scaling: a technique to scale data to a given range [0,1] or any other range
- Normalization/Standardization: a technique to scale data to mean with zero and unit-variance
- Augmentation: a technique to create additional data based on input sample which slightly differ from it, e.g. image rotation, flip, scale, crop, etc.
- Bucketing/Binning: a technique to place similar values into buckets/bins

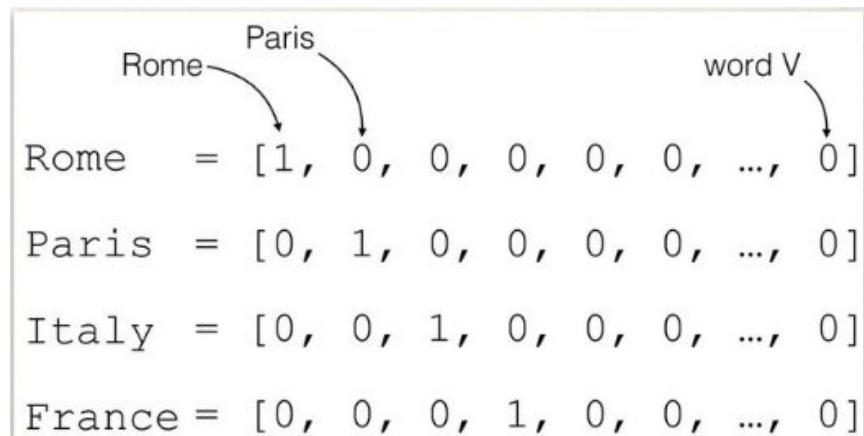
Data visualization

- Graphical representation may reveal important features of the data
 - find correlations, identify range, etc.
- Identify features which may require transformations, e.g. see outliers or skewness (asymmetry in probability distribution) in data
- It helps to identify a strategy how to deal with different features



One-hot encoding

- Technique to handle **categorical** data
- “One-Hot” refers to a state in electrical engineering where all of the bits in a circuit are 0, except a single bit with a value of 1 (“hot”)
- It represents a **categorical column as a vector of words**
- You need to define the word vector for the full set of data (train + test datasets)
 - Issues with NULL or missing data
 - delete rows with missing data
 - input data for missing values
 - Problematic with high cardinality



Leave-one-out encoding

- Effective by high cardinality
- Y is what we want to predict
- Encode UserID:
 - Train dataset:
 - Take mean of Y's for all rows with same UserID except the one you want to encode
 - multiply random noise
 - Test dataset
 - No Y, just use frequency of UserID

Split	UserID	Y	mean_y	random	newID
Train	A1	0	0.667	1.05	0.70035
Train	A1	1	0.333	0.97	0.32301
Train	A1	1	0.333	0.98	0.32634
Train	A1	0	0.667	1.02	0.68034
Test	A1	-	0.5	1	0.5
Test	A1	-	0.5	1	0.5
Train	A2	0			

Word embedding

- A way to capture multi-dimensional relationships between categories
 - you define a dimension of word vector up-front
 - it projects categorical variables into another phase space, e.g. days may be sunny or rainy, season or off season, Sunday and Saturday may have similar effect while other days may be treated independently
 - Use neural networks or other ML algorithms to train the model to find the best representation of embedded variables

Frequency based word embedding

- **Count Vector**

- Corpus C of D documents $\{d_1, d_2, \dots, d_D\}$ and N unique tokens (words) in C
- The N tokens will form our dictionary and the size of the Count Vector matrix M will be given by $D \times N$. Each row in the matrix M contains the frequency of tokens in $D(i)$

- **TF-IDF Vector**

- Similar to Count vector, but frequency is calculated with respect to all documents

- **Co-Occurrence Vector**

- Based on frequency of words appearing together (for example, it is)³⁹

	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	Document 7	Document 8
Term(s) 1	10	0	1	0	0	0	0	2
Term(s) 2	0	2	0	0	0	18	0	2
Term(s) 3	0	0	0	0	0	0	0	2
Term(s) 4	6	0	0	4	6	0	0	0
Term(s) 5	0	0	0	0	0	0	0	2
Term(s) 6	0	0	1	0	0	1	0	0
Term(s) 7	0	1	8	0	0	0	0	0
Term(s) 8	0	0	0	0	0	3	0	0

↑
Document Vector

(

)

)

)

Prediction based word embedding

- Word2vec based on neural networks
 - Continuous Bag of words (**CBOW**): predicts the probability of a word given a context
 - Skip-Gram model: predicts the context given a word

[https://www.analyticsvidhya.com/blog/2017/06/word-embedding_s-count-word2veec/](https://www.analyticsvidhya.com/blog/2017/06/word-embedding-s-count-word2veec/)

