Big data science Day 1

F. Legger - INFN Torino

https://github.com/Course-bigDataAndML/MLCourse-INFN-2022

Schedule

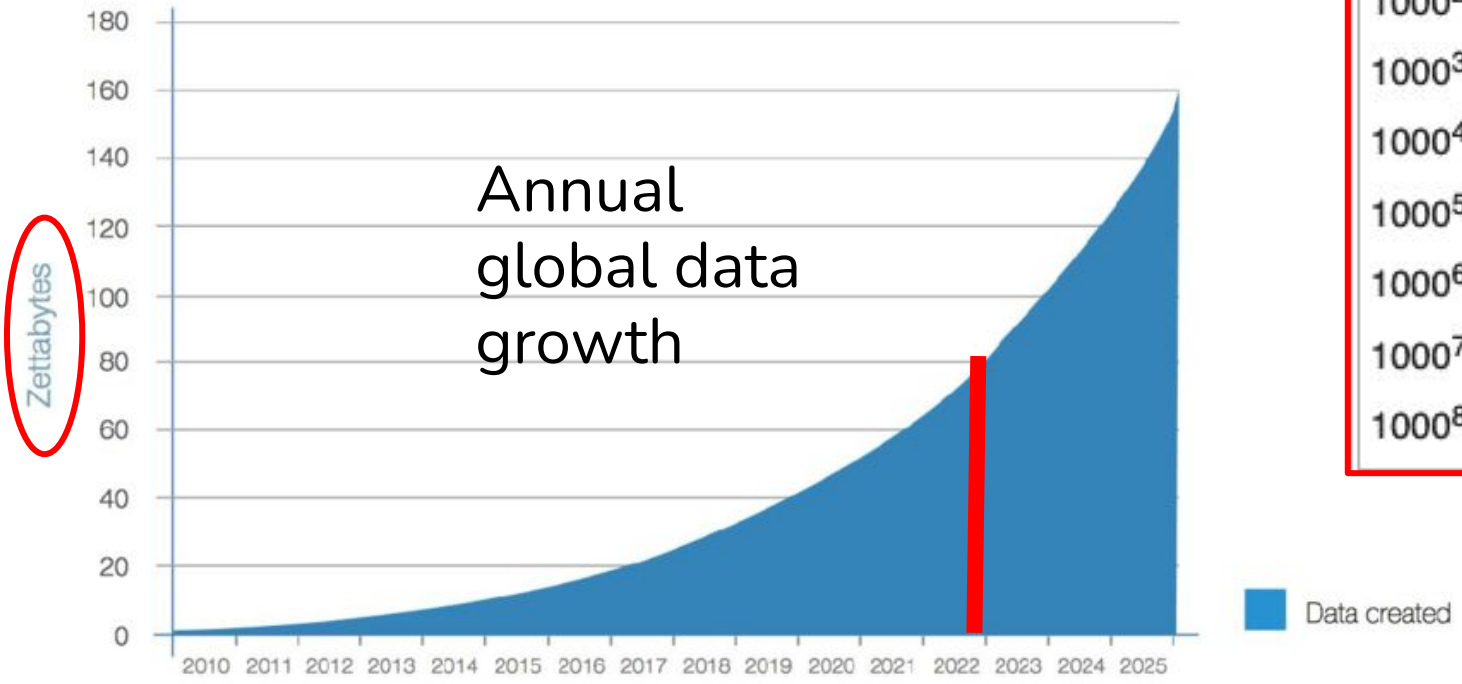
- **Mon, Nov. 7 - Wed, Nov. 9:**
  - 10:00 - 12:00 Lecture on big Data, ML
  - 14:00 - 16:00 Computing infrastructure+hands-on
- **Thursday, Nov. 10:**
  - Dive in on CNNs + hands-on on colab
- **Friday Nov 11:**
  - 10:00 - 12:00 Introduction to INFN Cloud
- **Test on Friday 14:00 - 16:00**
  - You must take the test to get the participation certificate!

Today

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

What is big data?

- Data that is too big to be analysed traditionally

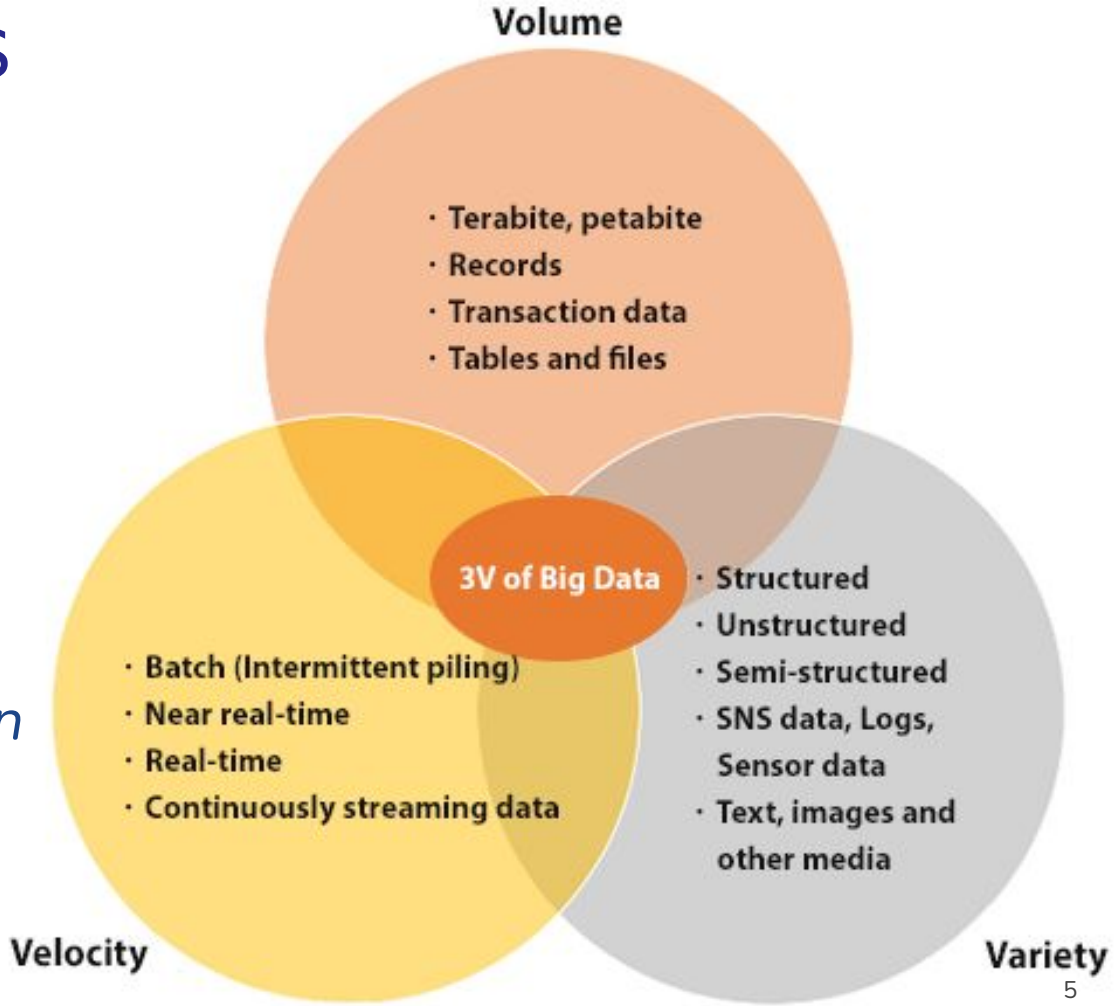


Decimal		
Value	Metric	
1000	kB	kilobyte
1000^2	MB	megabyte
1000^3	GB	gigabyte
1000^4	TB	terabyte
1000^5	PB	petabyte
1000^6	EB	exabyte
1000^7	ZB	zettabyte
1000^8	YB	yottabyte

https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

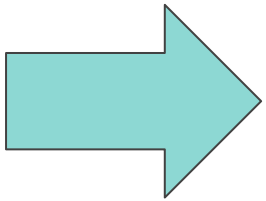
It’s all about Vs

“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

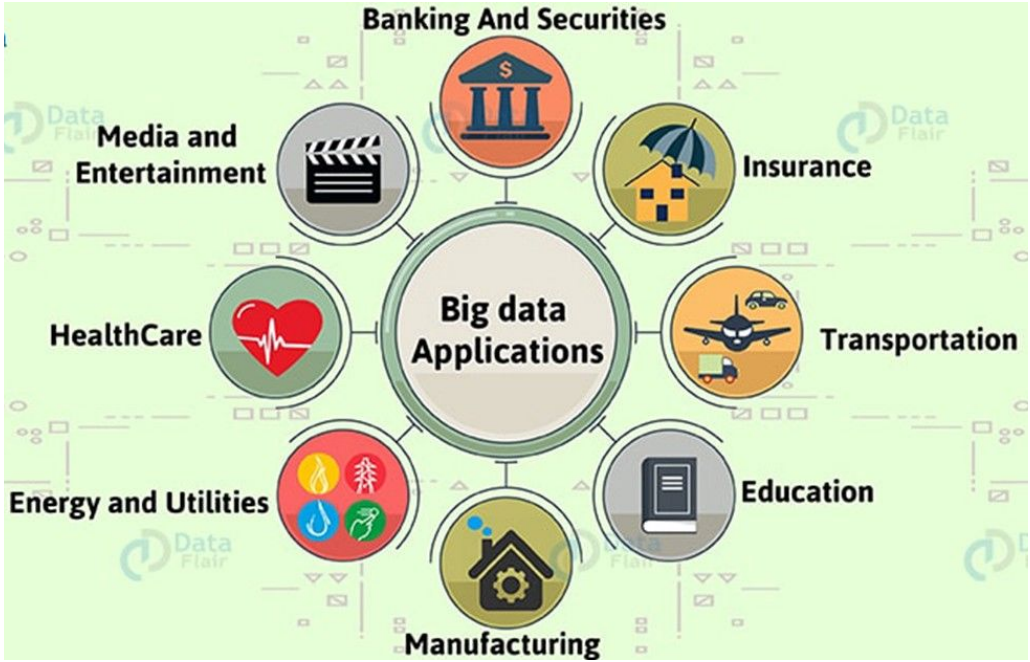


Complex and heterogeneous

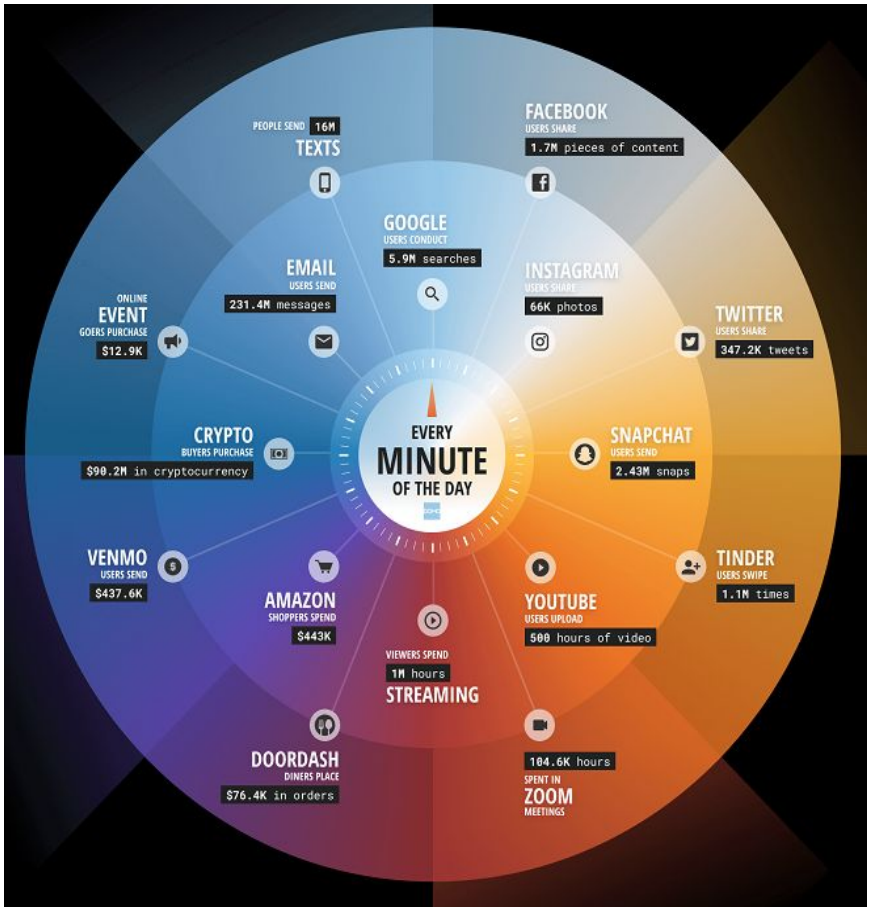
https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx

Big data sources: Business

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

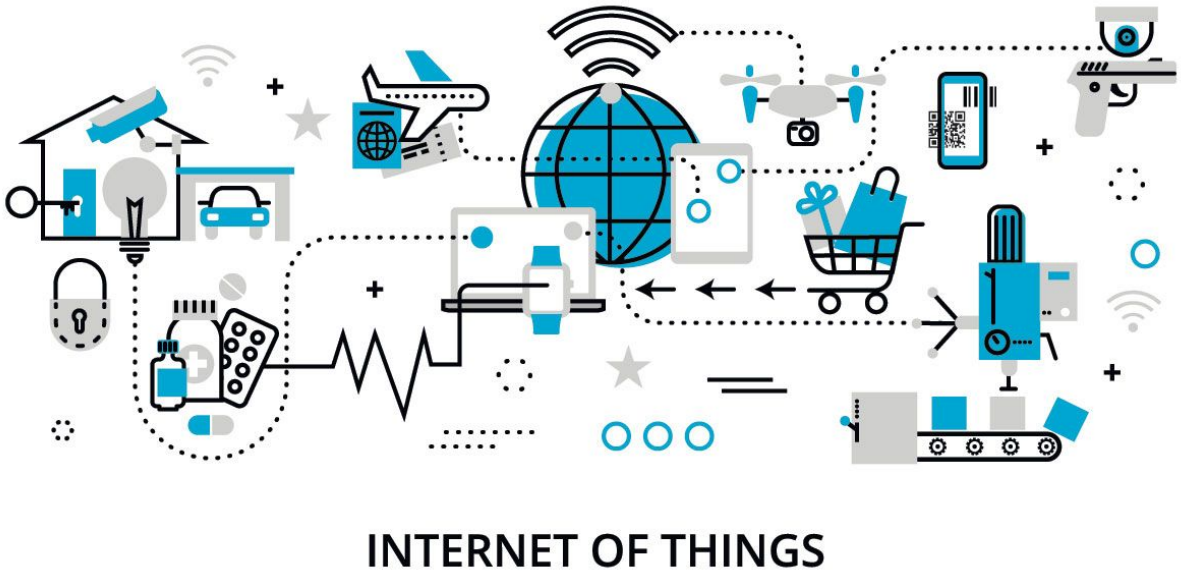


Big data sources: Human **(x10)**



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

Big data sources: Machine **(x100)**

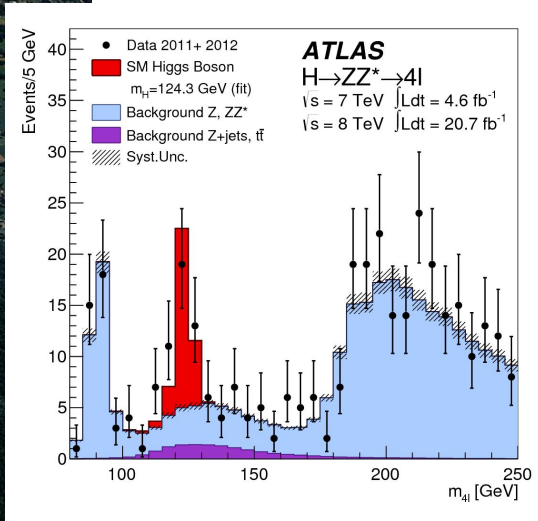


- **Internet of Things (IoT)**
  - Sensors: traffic, weather, mobile phone location, etc.
  - Security, surveillance videos, and images
  - Satellite images
  - Data from computer systems (logs, web logs)

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

- 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

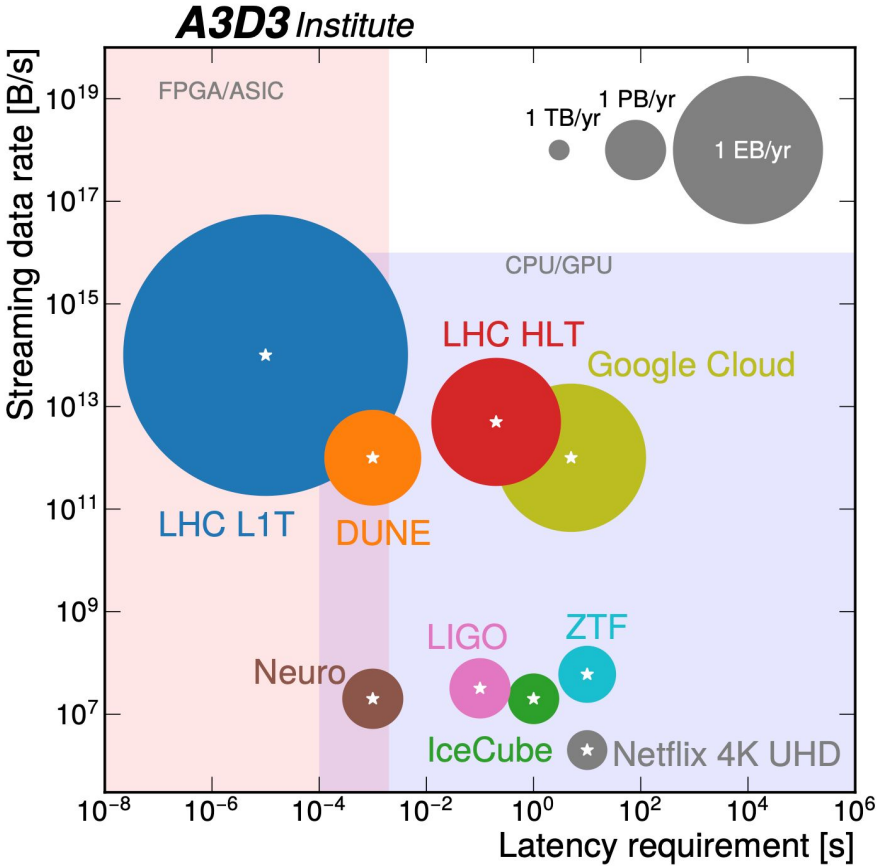
We cannot store this!

10^9 (LHC events/s) x 1 MB/event = 1 PB / second

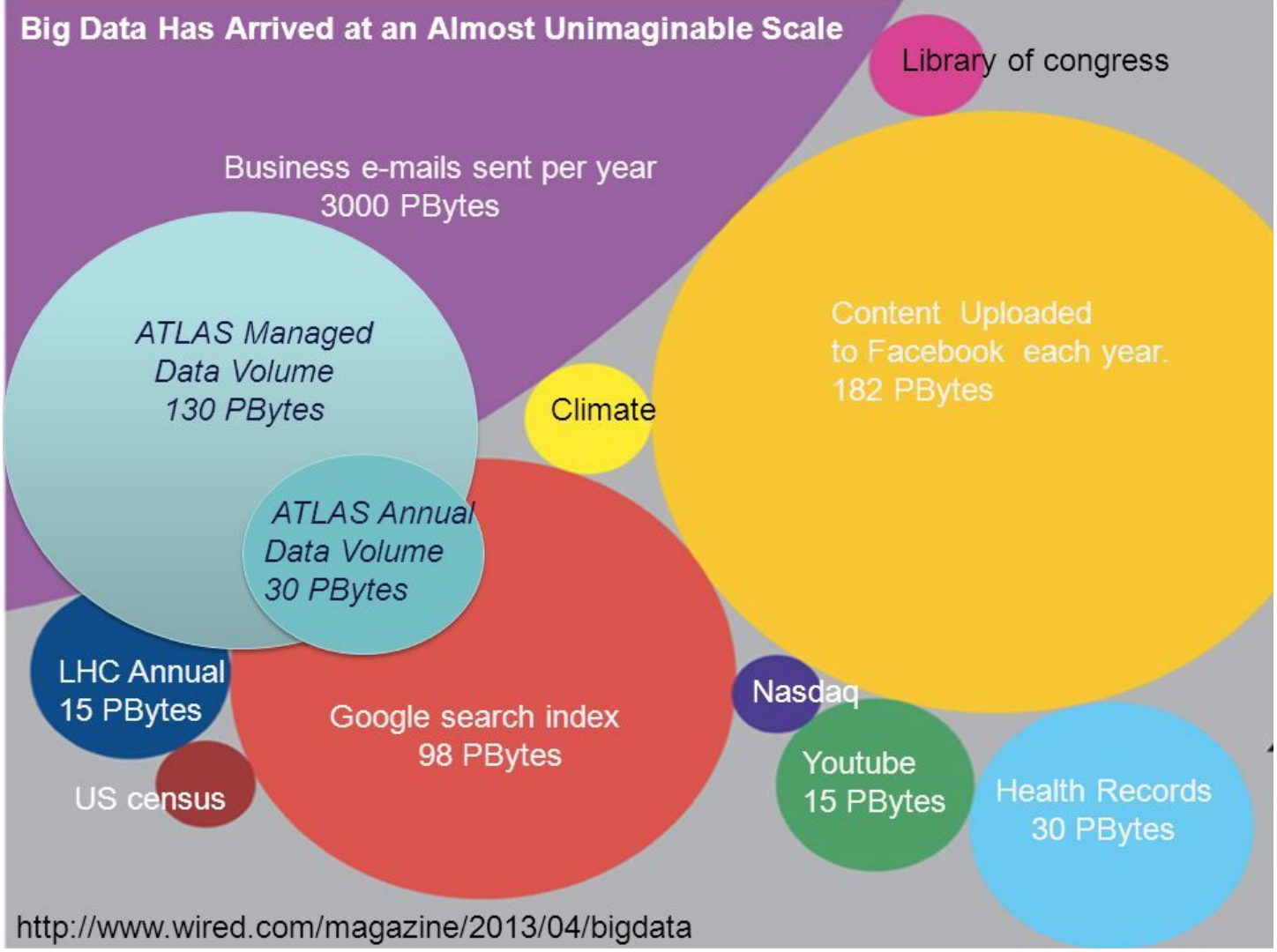
- Only one in a billion is an Higgs boson!
- Trigger system (real time): "empty" pictures immediately thrown away: 1GB/second
- Data stored, processed and analysed using the **Worldwide LHC Computing Grid (WLCG)**

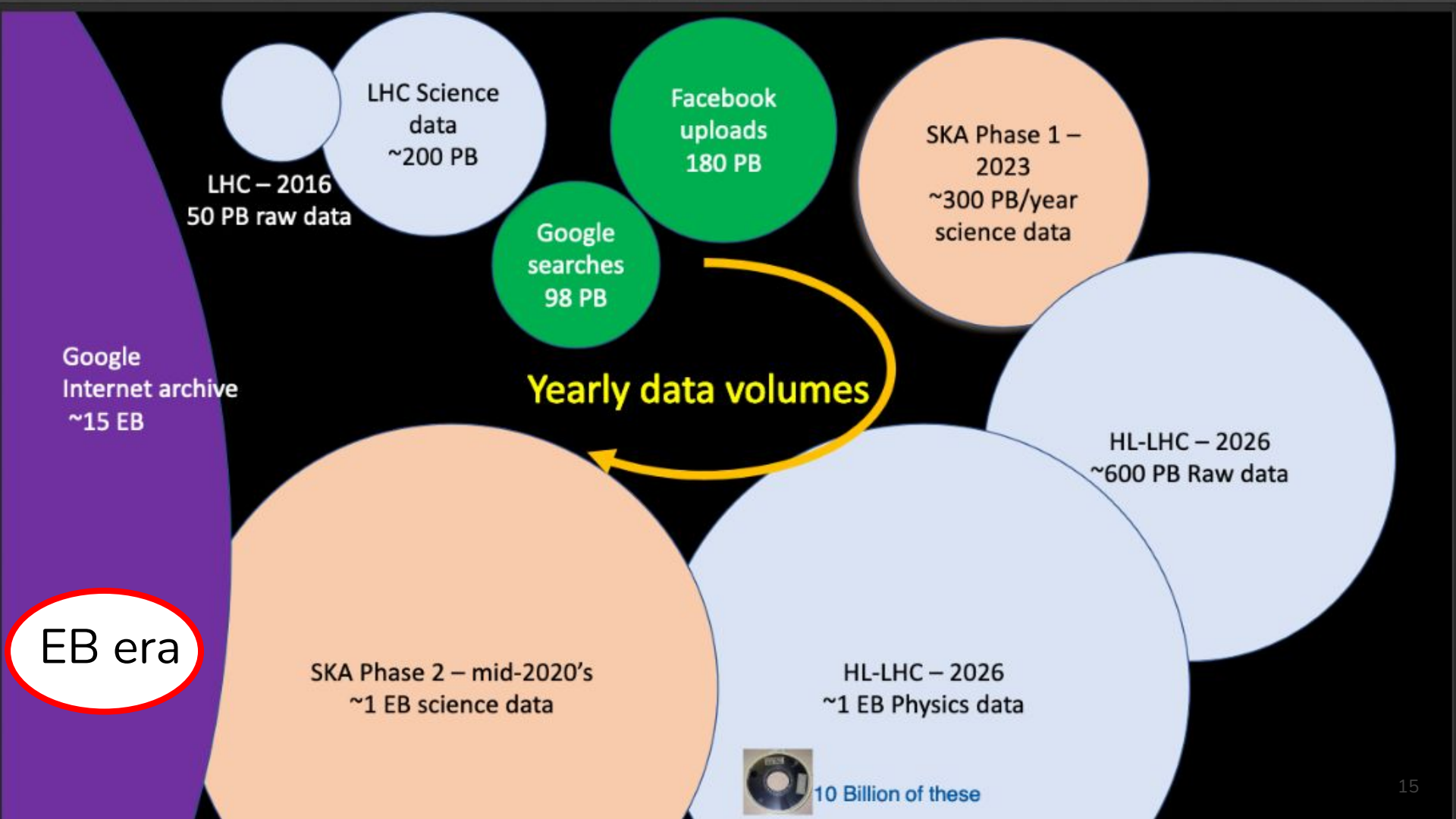
10^3 (events/s) x 10^7 (s/year) x 1 MB/event = 10 PB / year

Big data at the LHC – online systems

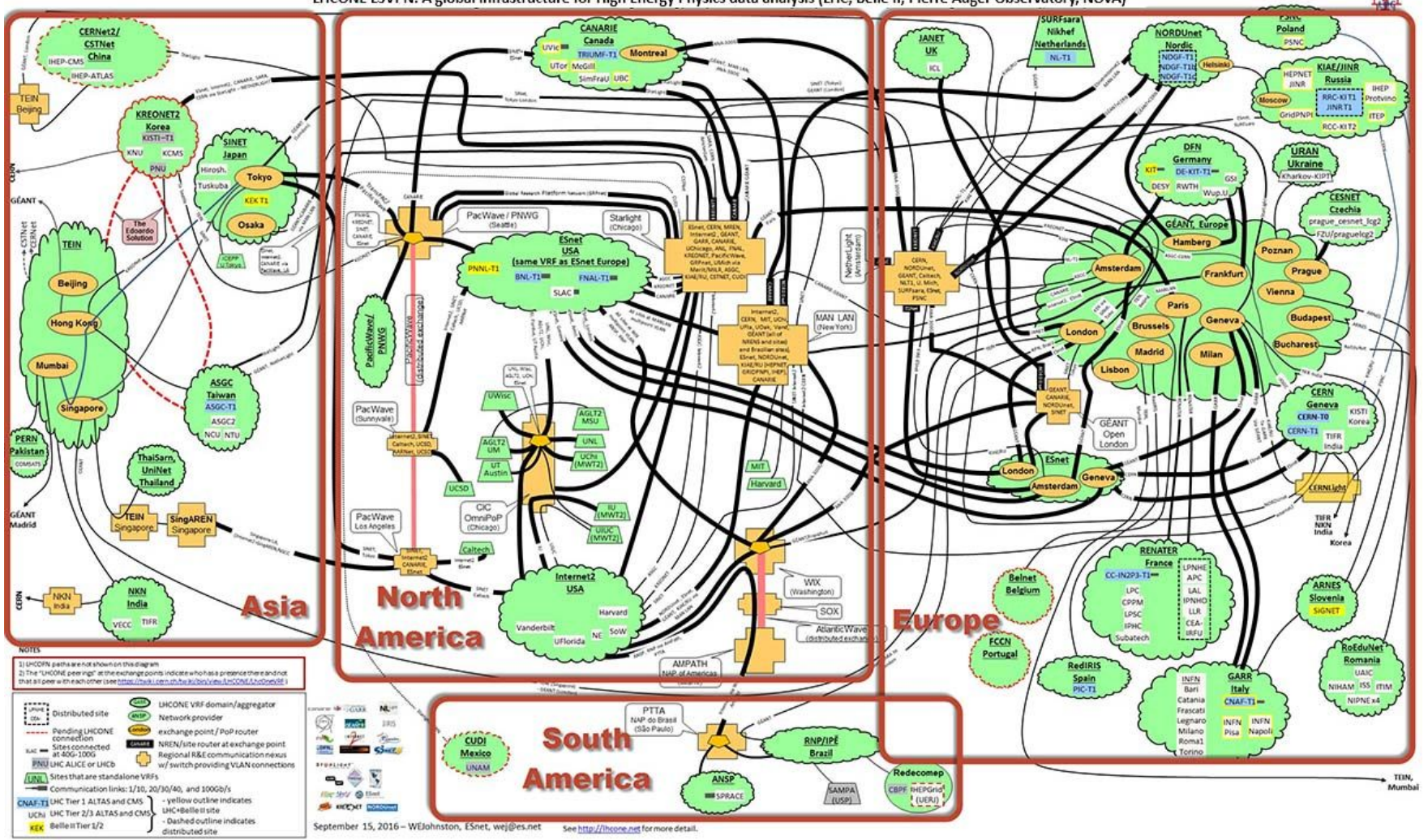


2013,
PB era


Big Data Has Arrived at an Almost Unimaginable Scale
Library of congress
Business e-mails sent per year
3000 PBytes
ATLAS Managed
Data Volume
130 PBytes
Climate
Content Uploaded
to Facebook each year.
182 PBytes
ATLAS Annual
Data Volume
30 PBytes
LHC Annual
15 PBytes
Google search index
98 PBytes
Nasdaq
US census
Youtube
15 PBytes
Health Records
30 PBytes
http://www.wired.com/magazine/2013/04/bigdata

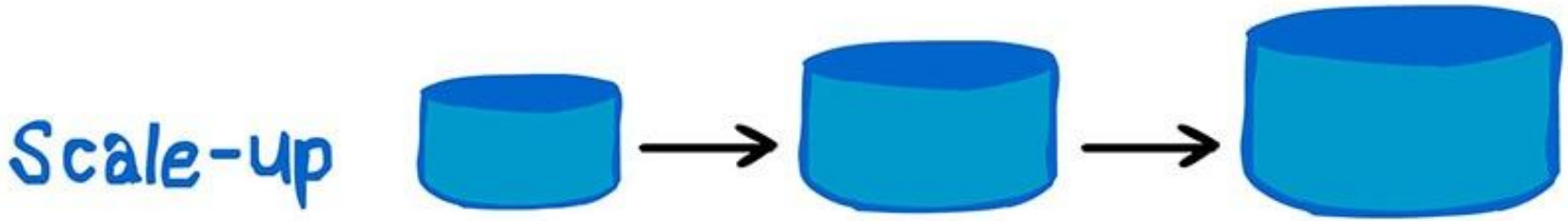

LHC – 2016
50 PB raw data
LHC Science data
~200 PB
Facebook uploads
180 PB
Google searches
98 PB
SKA Phase 1 – 2023
~300 PB/year
science data
Google
Internet archive
~15 EB
Yearly data volumes
HL-LHC – 2026
~600 PB Raw data
EB era
SKA Phase 2 – mid-2020's
~1 EB science data
HL-LHC – 2026
~1 EB Physics data
10 Billion of these


11/28/2013 11:44:13 am
11:40 am
11:51 am
Running jobs: 214268
Transfer rate: 42.74 GiB/sec
World
LHC
Computing
Grid
US Dept of State Geographer
© 2013 Google
Image Landsat
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
42 countries
170 computing centres
Over 2 million tasks daily
1 million computer cores
2 exabytes of storage: 1.2 Tape + 0.8 Disk
Google earth

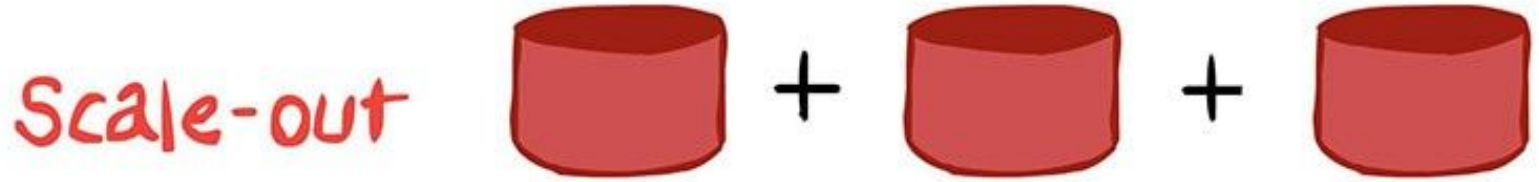

LHCONE L3VPN: A global infrastructure for High Energy Physics data analysis (LHC, Belle II, Pierre Auger Observatory, NOvA)
Asia
North America
Europe
South America
CERNet2/ CSTNet China
IHEP-CMS
IHEP-ATLAS
TEIN Beijing
KREONET2 Korea
KISTI-T1
KNU
KCMS
PNU
SINET Japan
Hirosh.
Tuskuba
Tokyo
KEK T1
Osaka
The Edoardo Solution
GÉANT
TEIN
Beijing
Hong Kong
Mumbai
Singapore
ASGC Taiwan
ASGC-T1
ASGC2
NCU
NTU
PERN Pakistan
COMSATS
ThaiSarn, UniNet Thailand
GÉANT Madrid
TEIN Singapore
SingAREN Singapore
CERN
NKN India
VECC
TIFR
CANARIE Canada
UVic
Montreal
UTor
McGill
SimFraU
UBC
PacWave / PNWG (Seattle)
Starlight (Chicago)
ESnet USA (same VRF as ESnet Europe)
PNNL-T1
BNL-T1
FNAL-T1
SLAC
PacificWave/ PNWG
PacificWave (distributed exchange)
PacWave (Sunnyvale)
PacWave Los Angeles
UWISC
AGLT2 MSU
AGLT2 UM
UNL
UCHi (MWT2)
UT Austin
UCSD
CIC OmniPoP (Chicago)
IU (MWT2)
UIUC (MWT2)
Caltech
MIT
Harvard
Internet2 USA
Vanderbilt
Harvard
UFlorida
NE
SoW
WIX (Washington)
SOX
AtlanticWave
AMPATH NAP of Americas
MAN LAN (New York)
NetherLight (Amsterdam)
JANET UK
ICL
SURFsara Nikhef Netherlands
NL-T1
NORDUnet Nordic
NDGF-T1
NDGF-T1d
NDGF-T1c
Helsinki
PSNC Poland
KIAE/JINR Russia
HEPNET JINR
RRC-KI T1
JINR T1
IHEP Protvino
ITEP
Moscow
GridPNPI
RCC-KI T2
URAN Ukraine
Kharkov-KIPT
DFN Germany
KIT
DE-KIT-T1
DESY
RWTH
GSI
Wup.U
CESNET Czechia
prague_cesnet_lcg2
FZU/praguelcg2
GÉANT, Europe
Hamberg
Amsterdam
Frankfurt
Poznan
Prague
Vienna
Paris
Brussels
Geneva
Budapest
London
Madrid
Milan
Bucharest
Lisbon
GÉANT Open London
ESnet
London
Amsterdam
Geneva
CERN Geneva
CERN-T0
CERN-T1
KISTI Korea
TIFR India
CERNLight
TIFR NKN India
Korea
RENATER France
CC-IN2P3-T1
LPNHE
APC
LAL
IPNHO
LLR
CEA-IRFU
LPC
CPPM
LPSC
IPHC
Subatech
Belnet Belgium
FCCN Portugal
RediRIS Spain
PIC-T1
ARNES Slovenia
SIGNET
RoEduNet Romania
UAIC
NIHAM
ISS
ITIM
NIPNE x4
GARR Italy
CNAF-T1
INFN Bari
Catania
Frascati
Legnaro
Milano
Roma1
Torino
INFN Pisa
INFN Napoli
CUDI Mexico
UNAM
PTTA NAP do Brasil (São Paulo)
RNP/IPÊ Brazil
ANSP
SPRACE
SAMPA (USP)
Redecomep
CBPF
RHEPGrid (UERJ)
TEIN, Mumbai
NOTES
1) LHCOPN paths are not shown on this diagram
2) The "LHCONE peerings" at the exchange points indicate who has a presence there and not that all peer with each other (see https://twiki.cern.ch/twiki/bin/view/LHCONE/LhcOneVrf)
Distributed site
Pending LHCONE connection
Sites connected at 40G-100G
LHC ALICE or LHCb
Sites that are standalone VRFs
Communication links: 1/10, 20/30/40, and 100Gb/s
LHC Tier 1 ALTAS and CMS
LHC Tier 2/3 ALTAS and CMS
Belle II Tier 1/2
LHCONE VRF domain/aggregator
Network provider
exchange point / PoP router
NREN/site router at exchange point
Regional R&E communication nexus w/ switch providing VLAN connections
- yellow outline indicates LHC+Belle II site
- Dashed outline indicates distributed site
September 15, 2016 – WEJohnston, ESnet, wej@es.net
See http://lhcone.net for more detail.

Scale up vs scale out

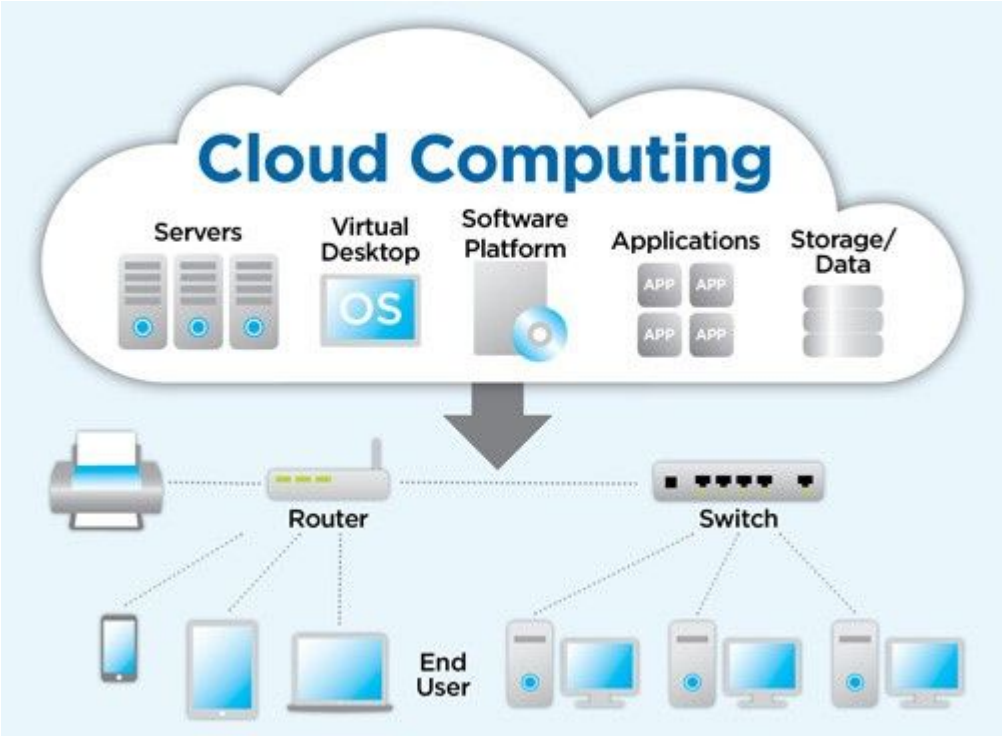
HPC, super computers



Grid, cloud



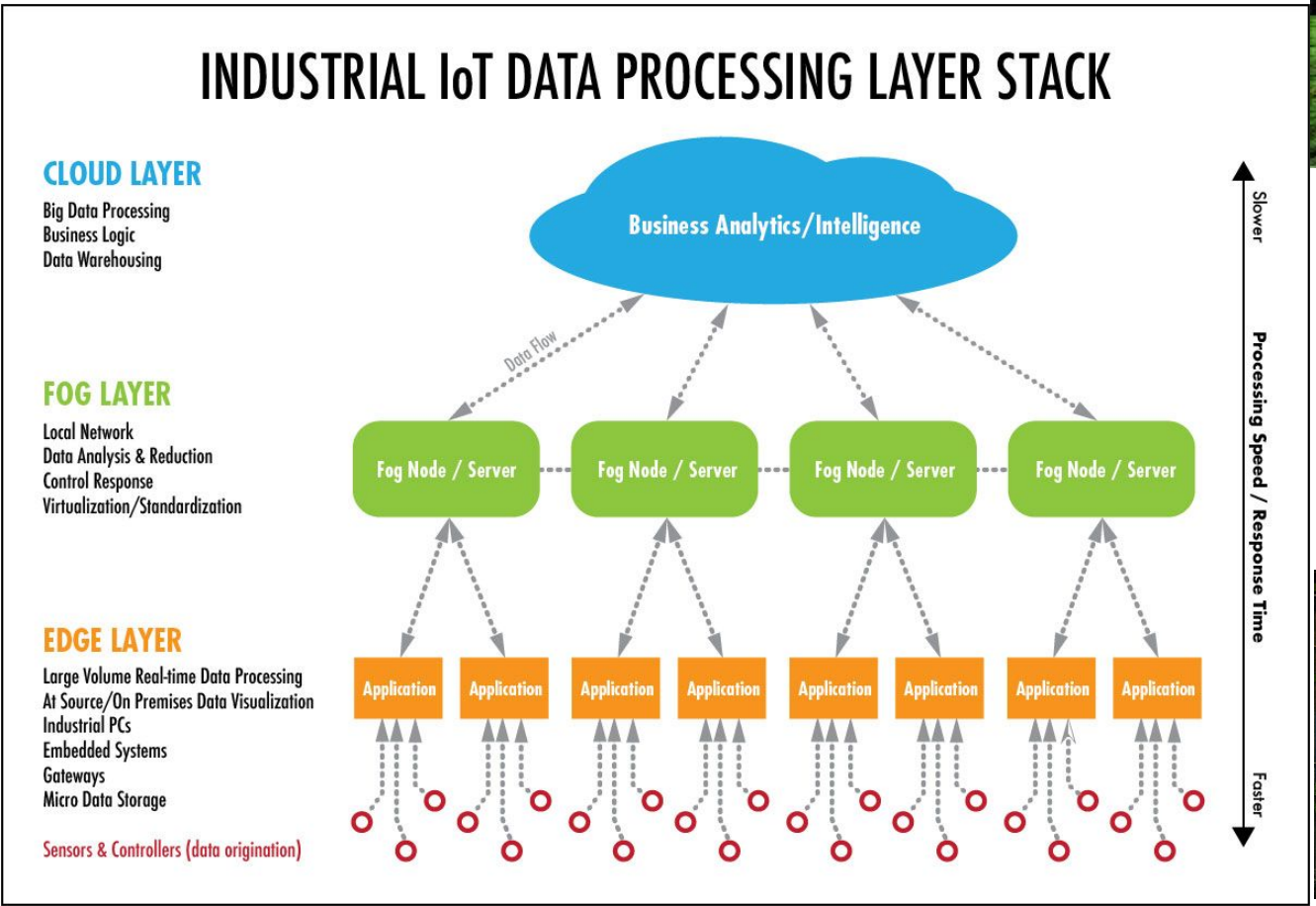
Cloud computing

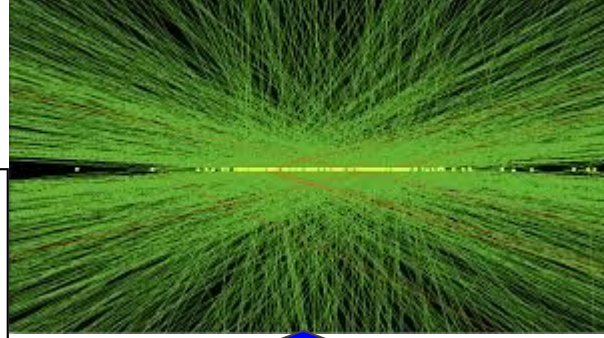


On-demand *availability of computer system resources, especially* ***data storage and computing power****, without direct active management by the user*

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



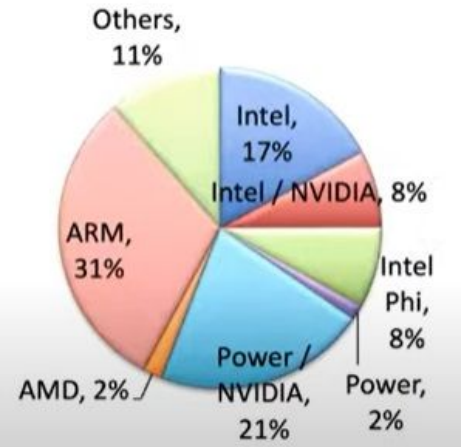


High Performance Computing (HPC)

- Typically involves supercomputers
  - Large computing power combining machines with large number of cores connected with high speed network
- Top 500 #1:
  - ORNL's **Frontier** First to Break the Exaflop Ceiling (May 30, 2022)
  - the first true exascale machine with an HPL score of 1.102 Exaflop/s.

1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	1,110,144	151.90	214.35	2,942
4	**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	148.60	200.79	10,096
5	**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	94.64	125.71	7,438
6	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44	15,371

Top European HPC



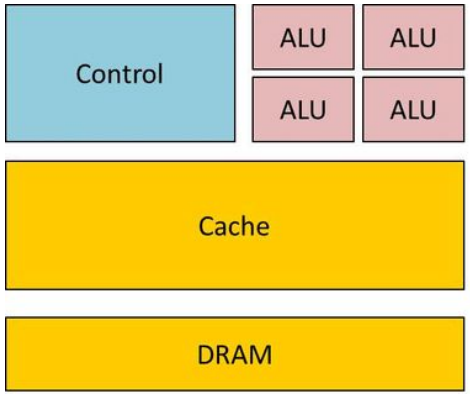
https://www.top500.org/

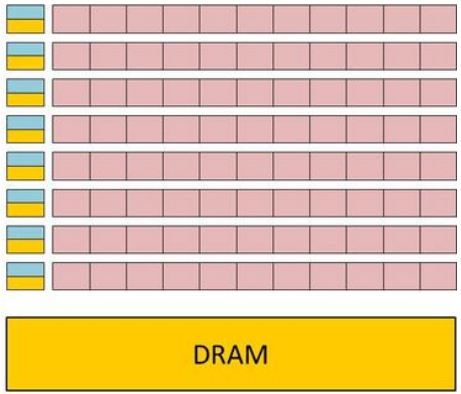
Graphical Processing Unit (GPU)

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





- Performances
  - Excels at matrix and vector operations
  - Gaming, rendering...and ML
- Power consumption (more FLOPS/Watt)

Keep in mind

- **GPUs** have high performance for floating point arithmetic
  - limited amount of memory available (may not hold large models or large batches of data)
  - rate at which data can be moved from CPU to GPU
  - memory bandwidth for GPUs must be seen relative to the amount of FLOPS: if one has more floating point units, a higher bandwidth is needed to keep them occupied
- **Matrix operations:** typically bandwidth cost is larger than multiplication cost
  - Particularly important for many small multiplications
- Be aware of power consumption and **overheating** when placing multiple GPUs close to one another!

Heterogeneous architectures

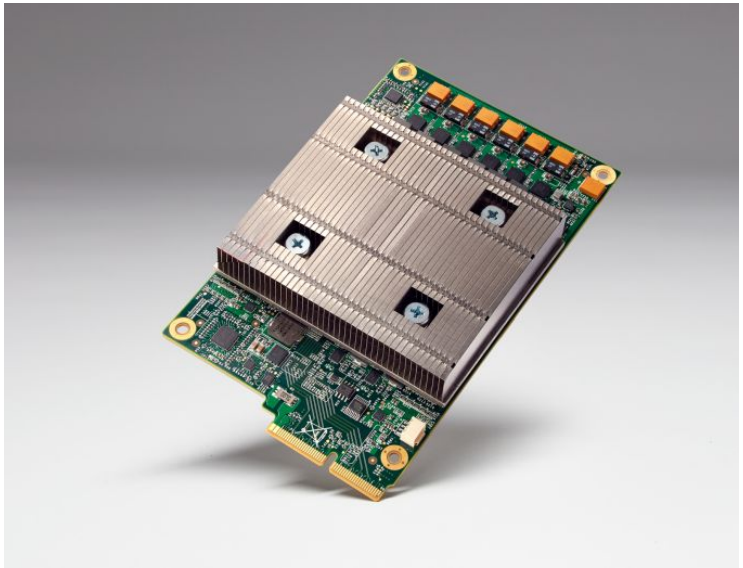
CPU
- Small models
- Small datasets
- Useful for design space exploration

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: Tensor(flow) Processing Unit
FPGA: Field Programmable Gate Array

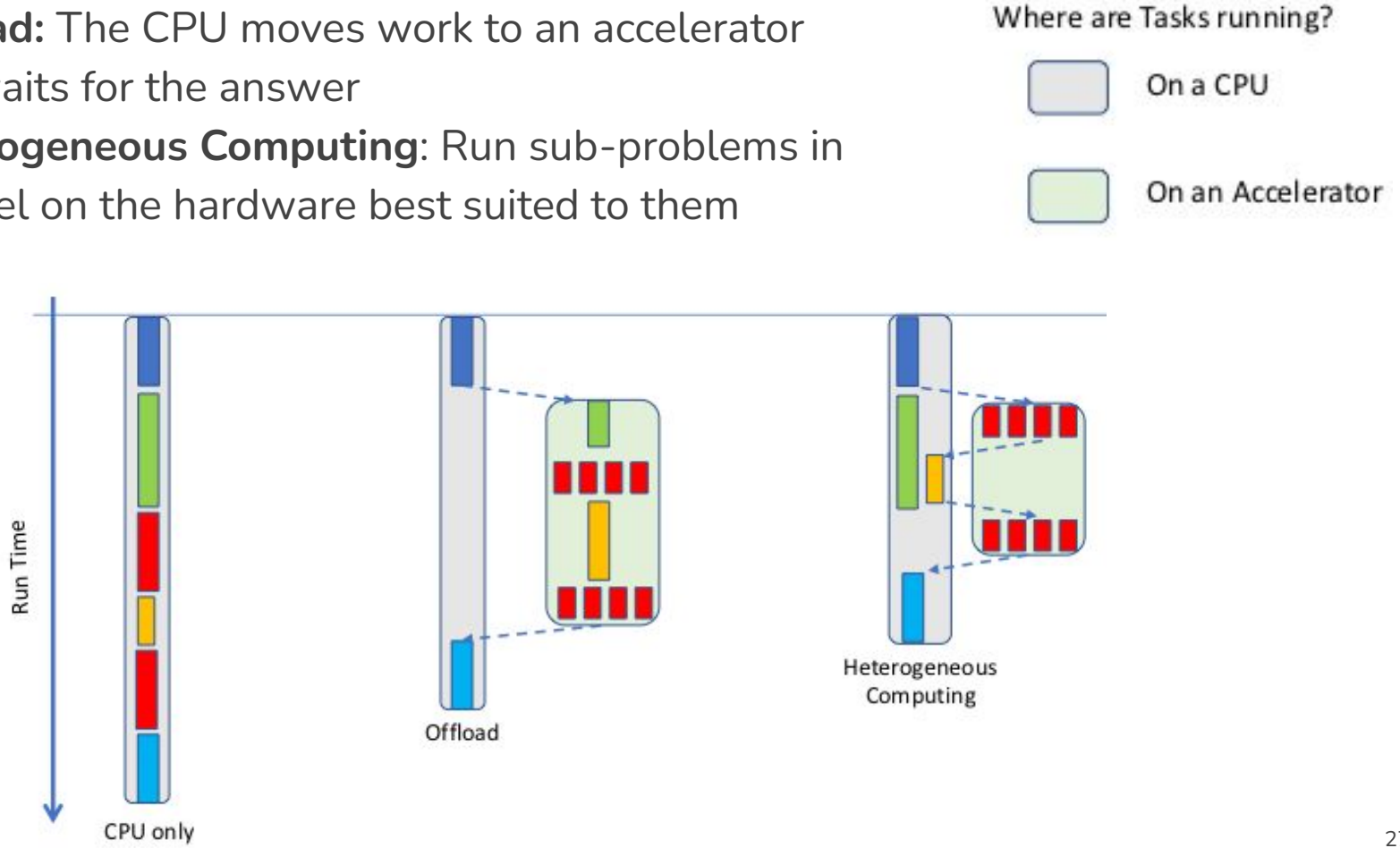


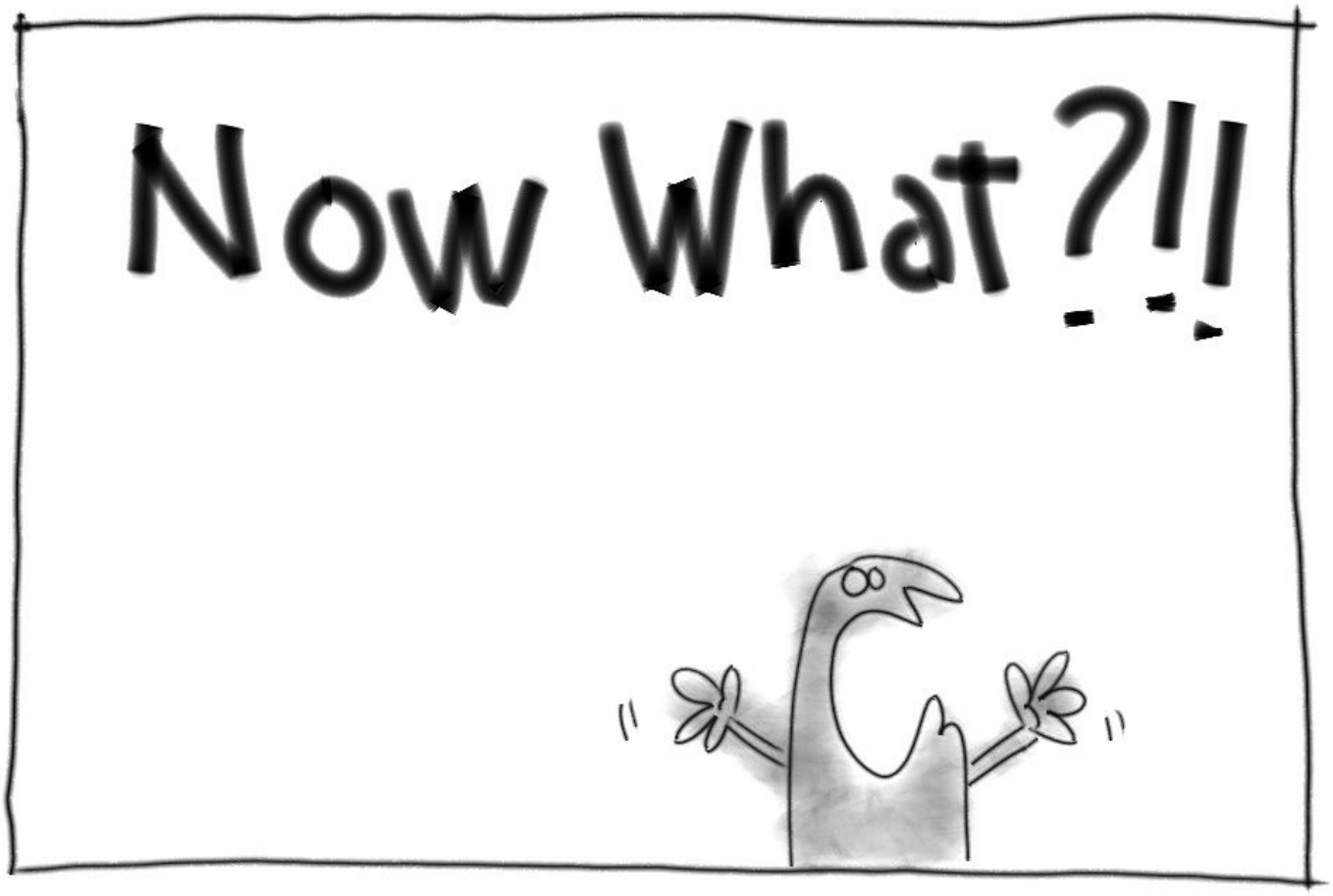
Options

- **NVIDIA (CUDA)**
  - Market leader, large user community, best support from DL frameworks
  - Can be used in python, C/C++, fortran, Matlab
- **AMD (HIP)**
  - Limited support for pyTorch and Tensorflow
  - Slightly behind in terms of performances
- **INTEL**
  - Xeon Phi: Poor support
  - Habana Gaudi **(AI processor)**-based AWS EC2 instances available since Oct 2021
- **Google TPU**
  - Good performances, more powerful than cloud GPUs
  - best for training, for prototype and inference better use cheaper alternatives
- **Amazon AWS and Microsoft Azure**
  - Powerful, easy to scale, expensive

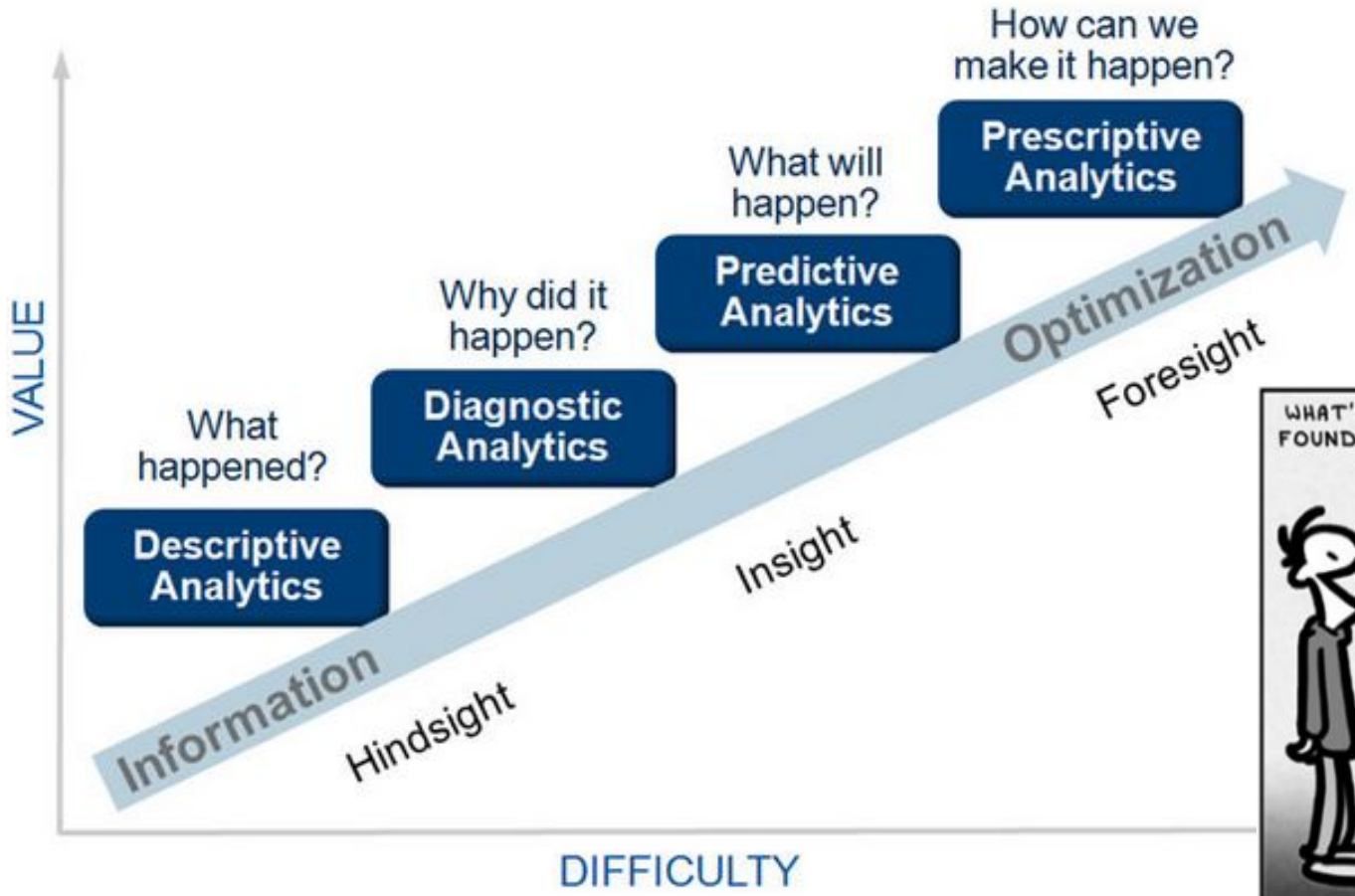
Heterogeneous programming: **Kokkos, Alpaka, SyCL**

- **Offload:** The CPU moves work to an accelerator and waits for the answer
- **Heterogeneous Computing**: Run sub-problems in parallel on the hardware best suited to them

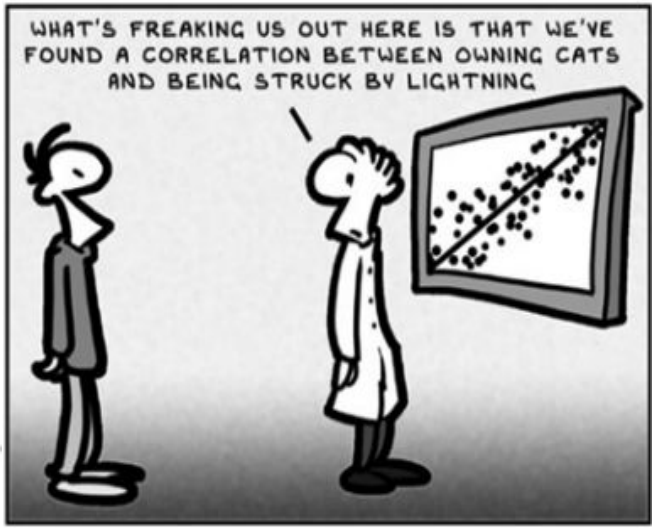



Now What?!!

Analytics


How can we make it happen?
Prescriptive Analytics
What will happen?
Predictive Analytics
Why did it happen?
Diagnostic Analytics
What happened?
Descriptive Analytics
Optimization
Foresight
Insight
Hindsight
Information
VALUE
DIFFICULTY

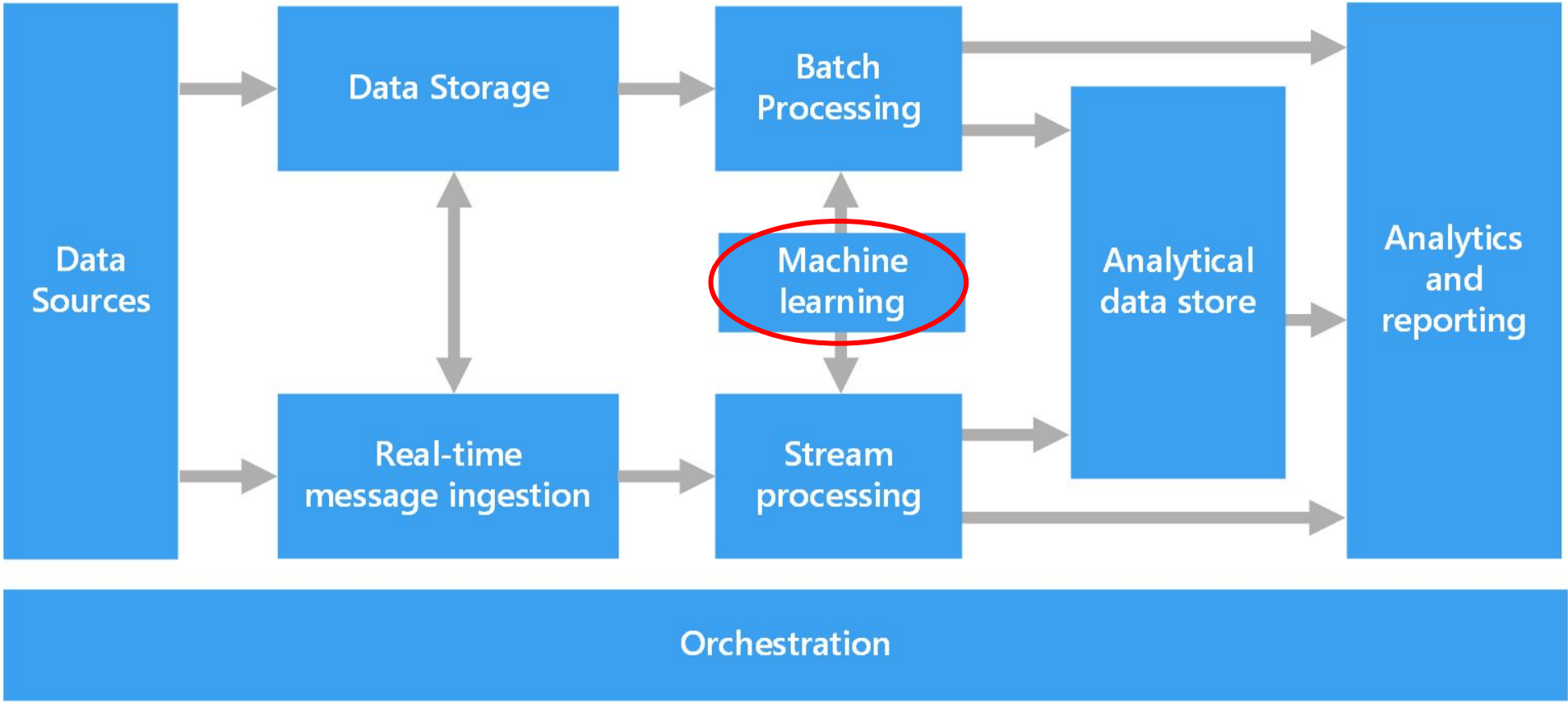

SHOULD WE ALWAYS IGNORE WHAT THE DATA SAYS, OR IS THIS MORE OF A ONE-TIME THING?


WHAT'S FREAKING US OUT HERE IS THAT WE'VE FOUND A CORRELATION BETWEEN OWNING CATS AND BEING STRUCK BY LIGHTNING

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

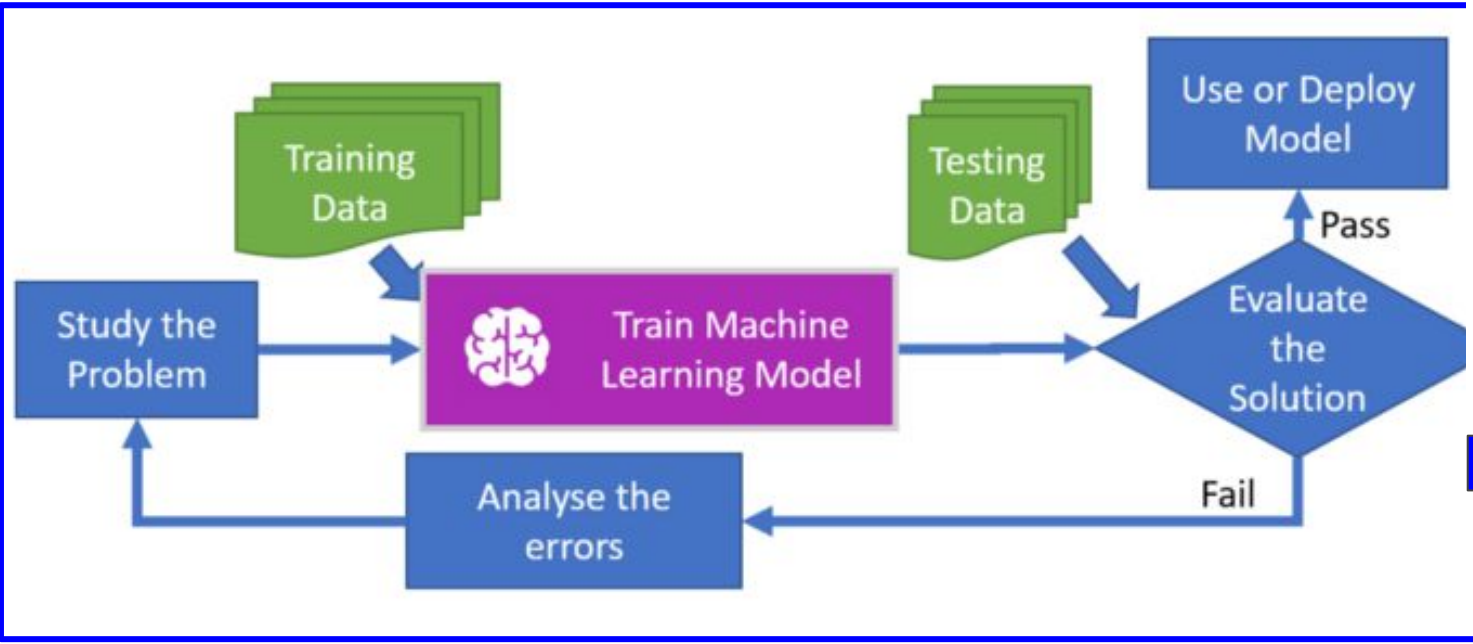
The big data pipeline

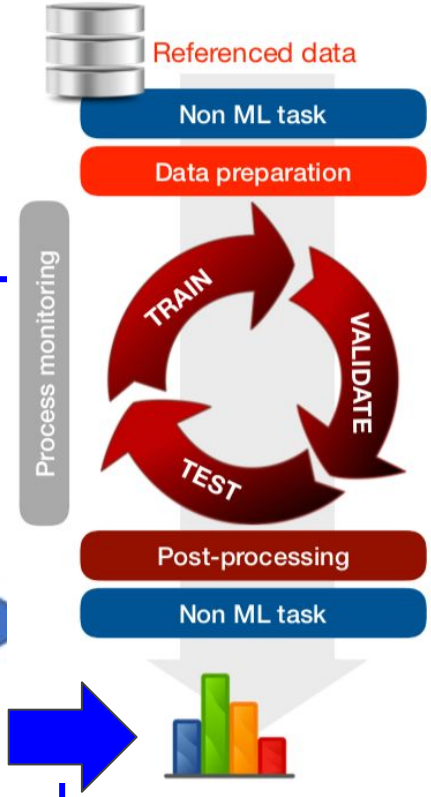


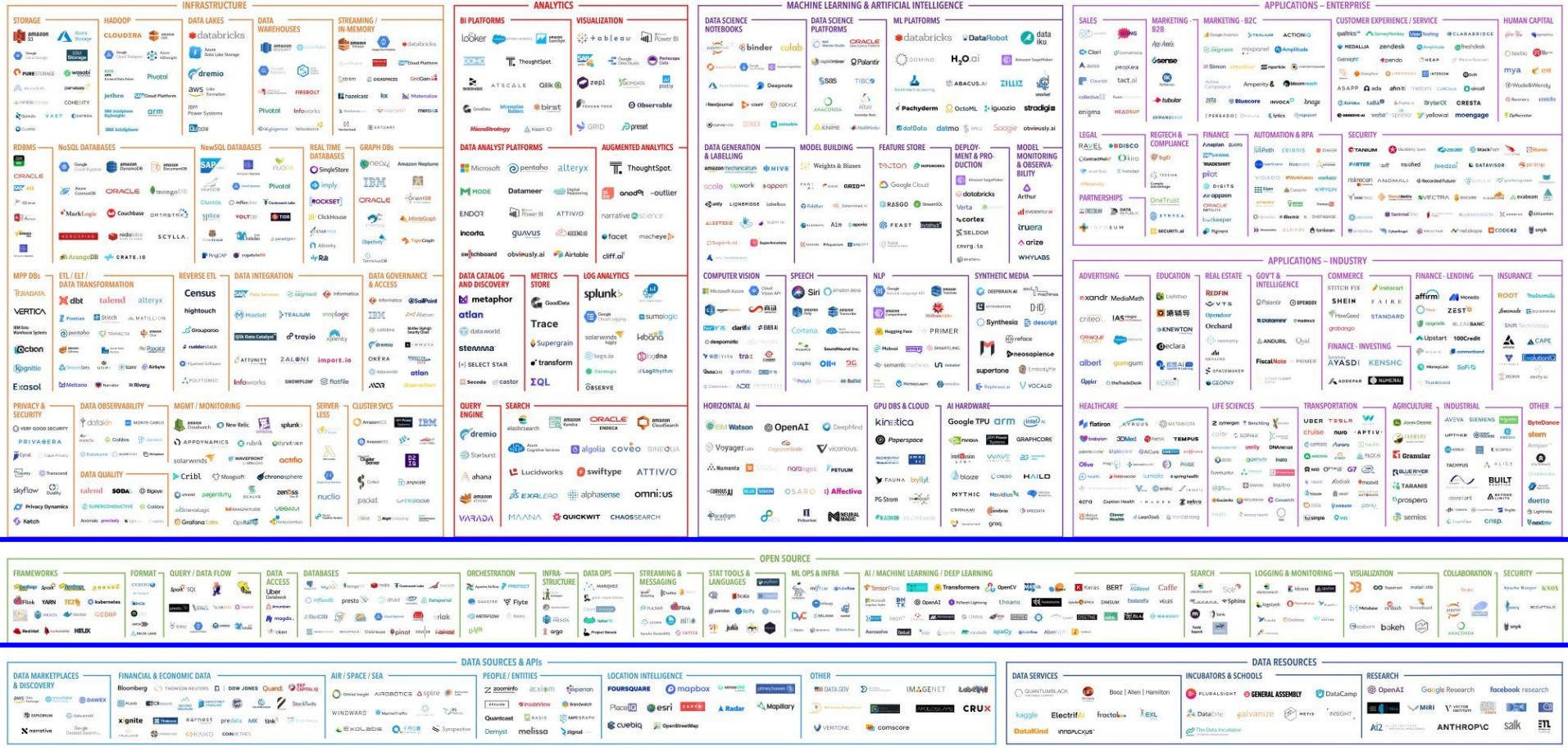
The machine learning pipeline

Input data is typically split into:

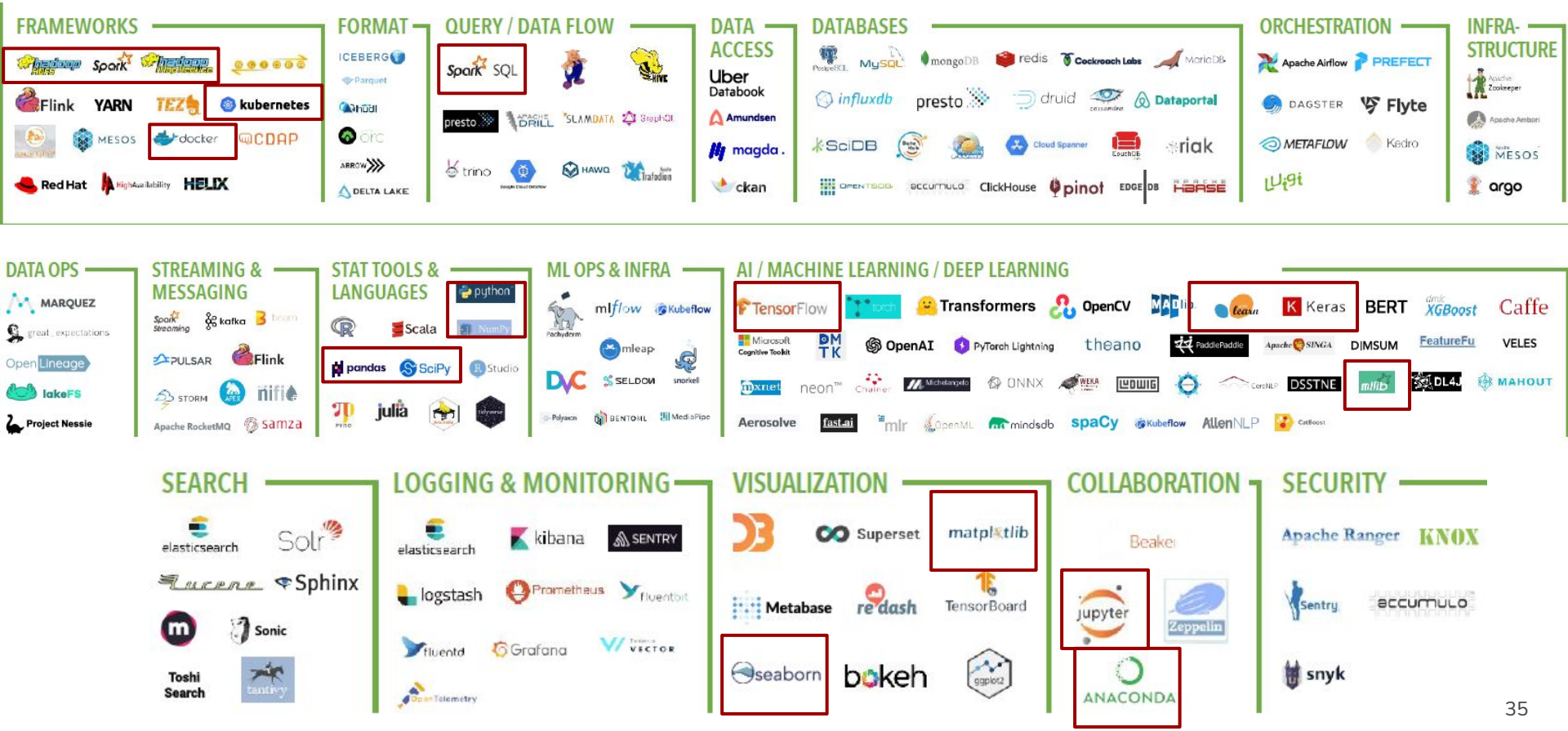
- Training dataset
- Testing dataset

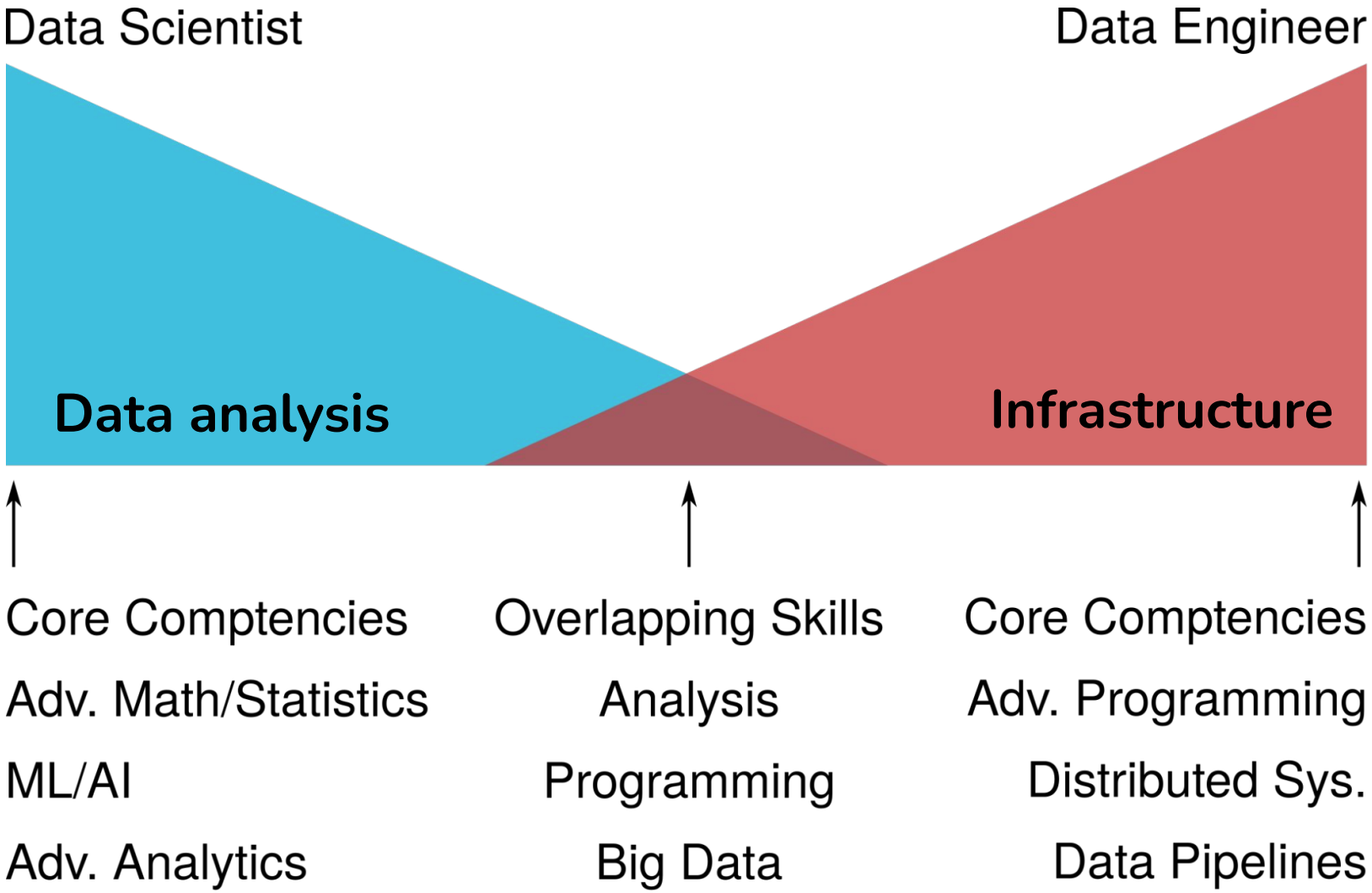


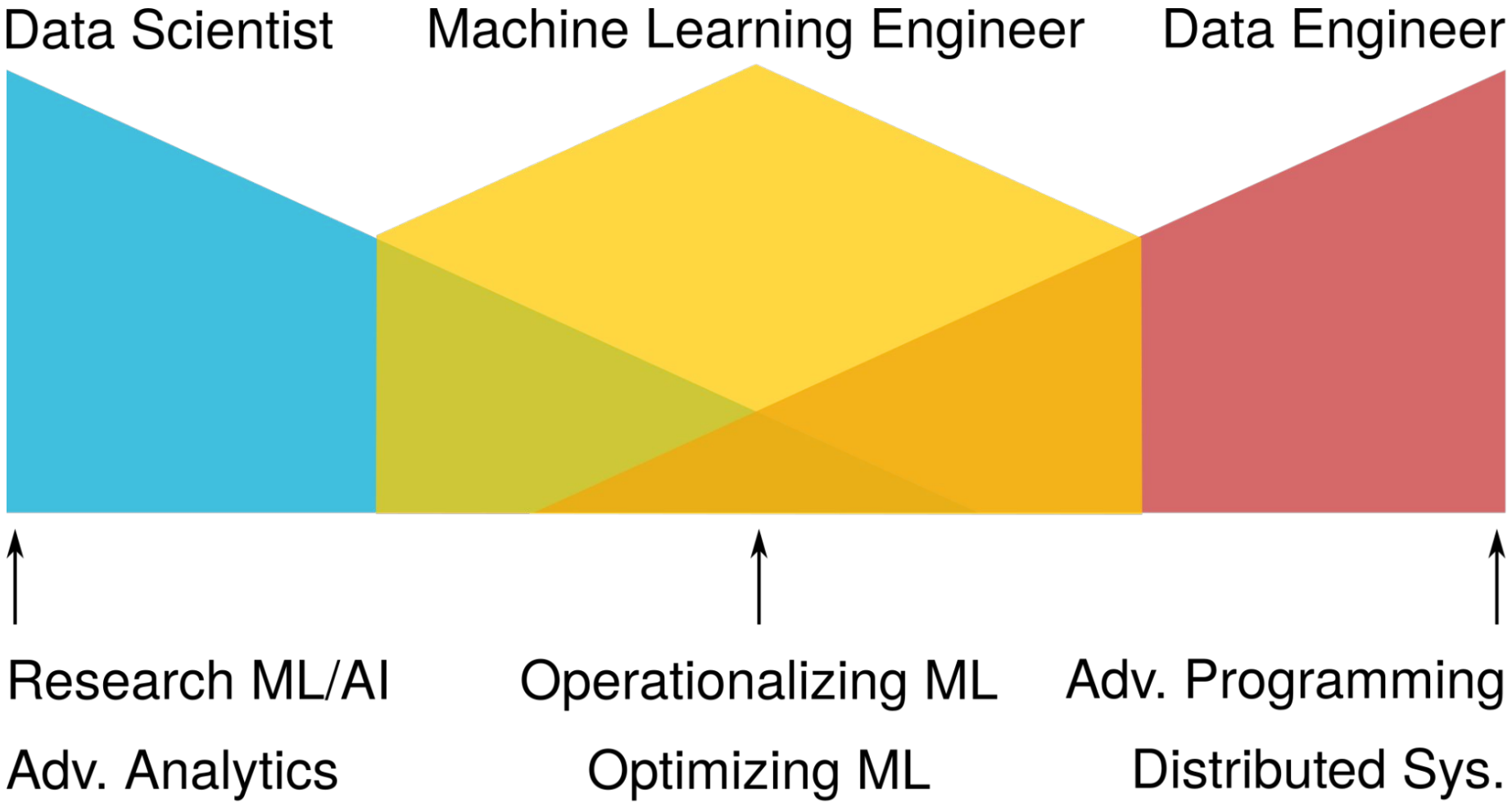




Open Source


FRAMEWORKS
Spark
Flink
YARN
TEZ
kubernetes
MESOS
docker
CDAP
Red Hat
HELIX
FORMAT
ICEBERG
Parquet
ORC
ARROW
DELTA LAKE
QUERY / DATA FLOW
Spark SQL
HIVE
presto
APACHE DRILL
SLAMDATA
GraphQL
trino
HAWQ
Trafodion
DATA ACCESS
Uber
Databook
Amundsen
magda.
ckan
DATABASES
PostgreSQL
MySQL
mongoDB
redis
Cockroach Labs
MariaDB
influxdb
presto
druid
cassandra
Dataportal
SciDB
Cloud Spanner
CouchDB
riak
OPENTSDB
accumulo
ClickHouse
pinot
EDGE DB
HBASE
ORCHESTRATION
Apache Airflow
PREFECT
DAGSTER
Flyte
METAFLOW
Kedro
Luigi
INFRA-STRUCTURE
Apache Zookeeper
Apache Ambari
MESOS
argo
DATA OPS
MARQUEZ
great_expectations
OpenLineage
lakeFS
Project Nessie
STREAMING & MESSAGING
Spark Streaming
kafka
beam
PULSAR
Flink
STORM
nifi
Apache RocketMQ
samza
STAT TOOLS & LANGUAGES
python
NumPy
Scala
pandas
SciPy
RStudio
julia
ML OPS & INFRA
Pachyderm
mlflow
Kubeflow
mleap
DVC
SELDON
snorkel
Polyaxon
BENTOML
MediaPipe
AI / MACHINE LEARNING / DEEP LEARNING
TensorFlow
torch
Transformers
OpenCV
MXNet
learn
Keras
BERT
XGBoost
Caffe
Microsoft Cognitive Toolkit
DMTK
OpenAI
PyTorch Lightning
theano
PaddlePaddle
SINGA
DIMSUM
FeatureFu
VELES
mxnet
neon
Chainer
Michelangelo
ONNX
Weka
LOGWIG
CoreNLP
DSSTNE
mlflib
DL4J
MAHOUT
Aerosolve
fast.ai
mlr
OpenML
mindsdb
spaCy
Kubeflow
AllenNLP
CatBoost
SEARCH
elasticsearch
Solr
Lucene
Sphinx
Sonic
Toshi Search
tantivy
LOGGING & MONITORING
elasticsearch
kibana
SENTRY
logstash
Prometheus
fluentbit
fluentd
Grafana
VECTOR
OpenTelemetry
VISUALIZATION
Superset
matplotlib
Metabase
redash
TensorBoard
seaborn
bokeh
ggplot2
COLLABORATION
Beaker
jupyter
Zeppelin
ANACONDA
SECURITY
Apache Ranger
KNOX
Sentry
accumulo
snyk


Data Scientist
Data Engineer
Data analysis
Infrastructure
Core Comptencies
Adv. Math/Statistics
ML/AI
Adv. Analytics
Overlapping Skills
Analysis
Programming
Big Data
Core Comptencies
Adv. Programming
Distributed Sys.
Data Pipelines

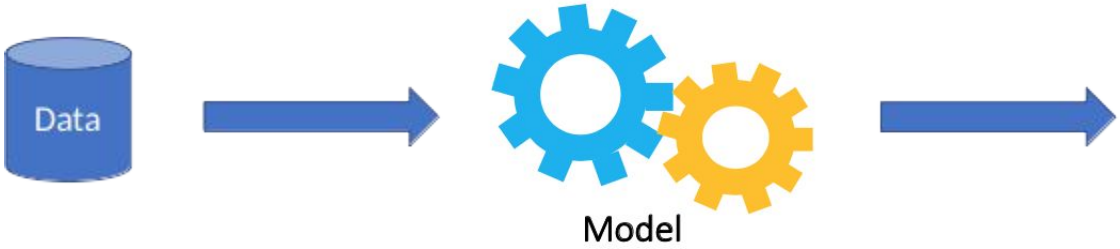

Data Scientist
Machine Learning Engineer
Data Engineer
Research ML/AI
Adv. Analytics
Operationalizing ML
Optimizing ML
Adv. Programming
Distributed Sys.

Machine Learning

Machine learning involves two mathematical entities

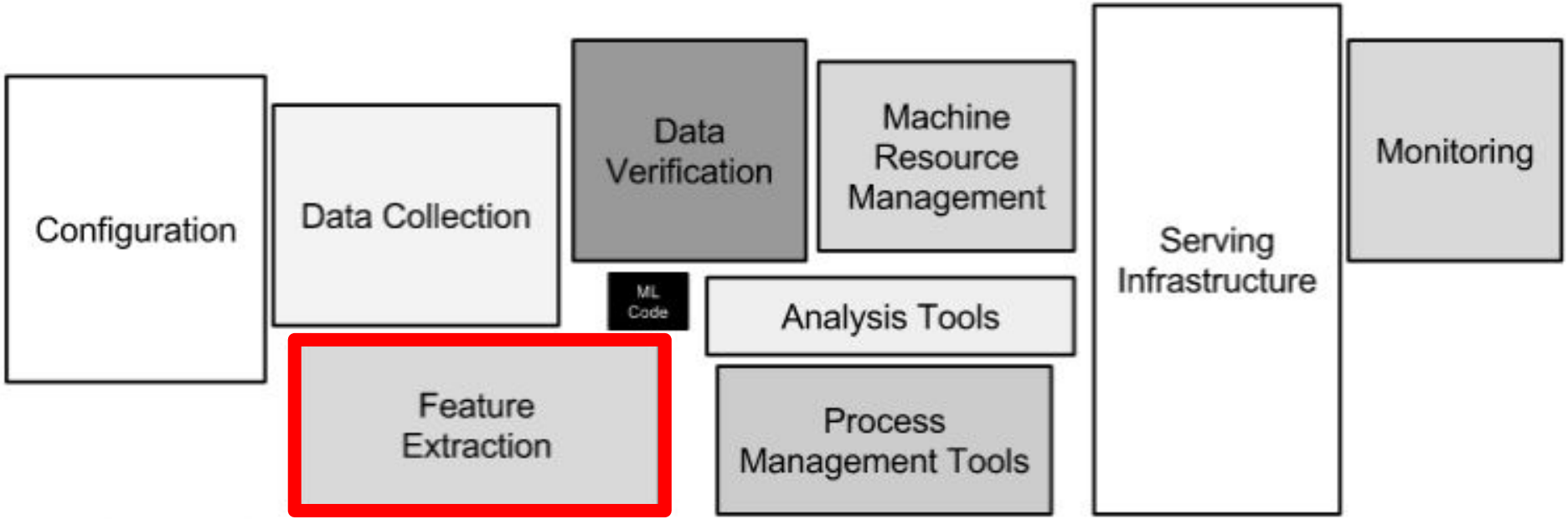
Model: a mathematical model describes the relationship between different aspects of the data

Features: a feature is a representation of raw data

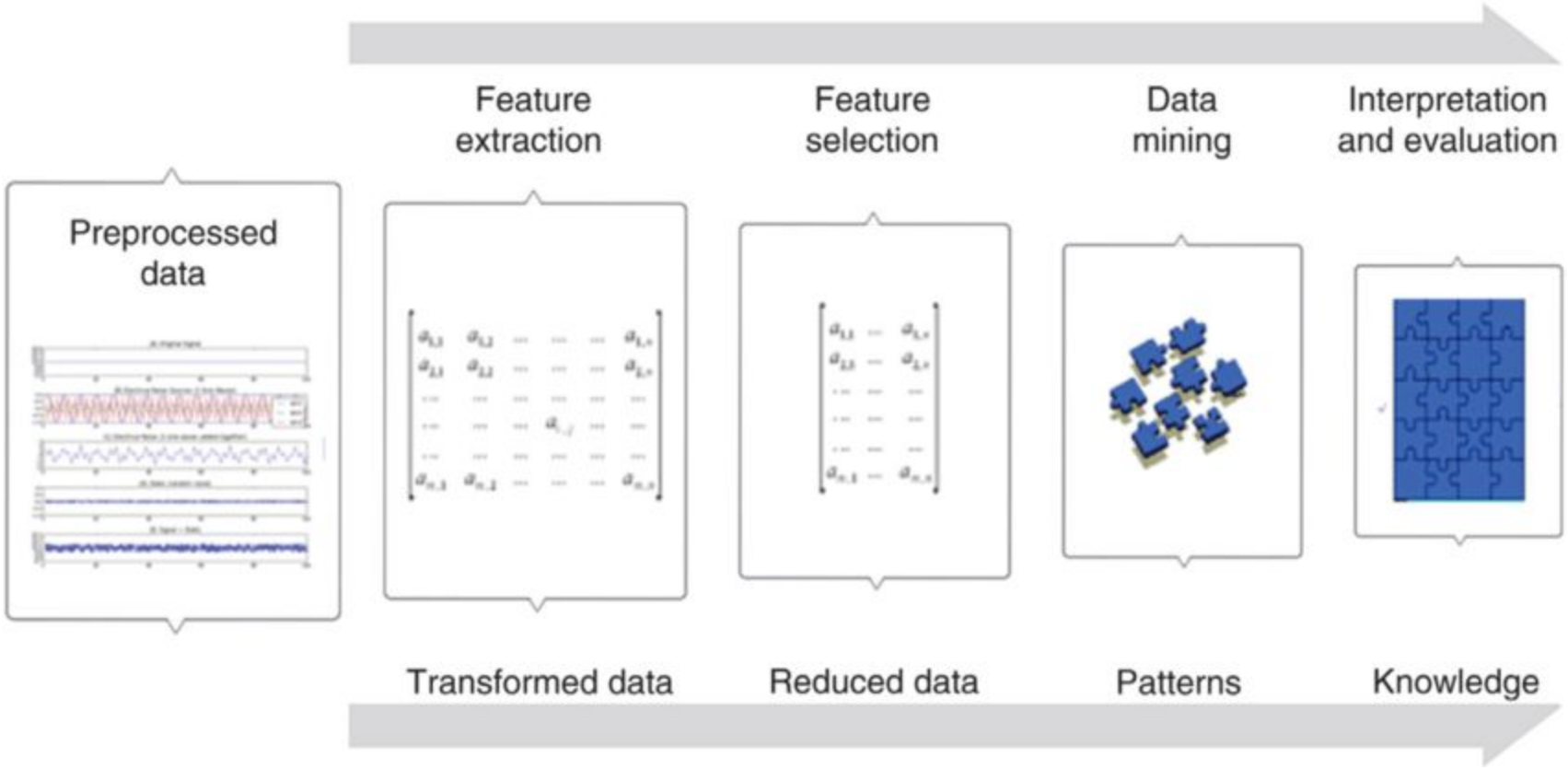


Typical ML workflow

Where is ML?



https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

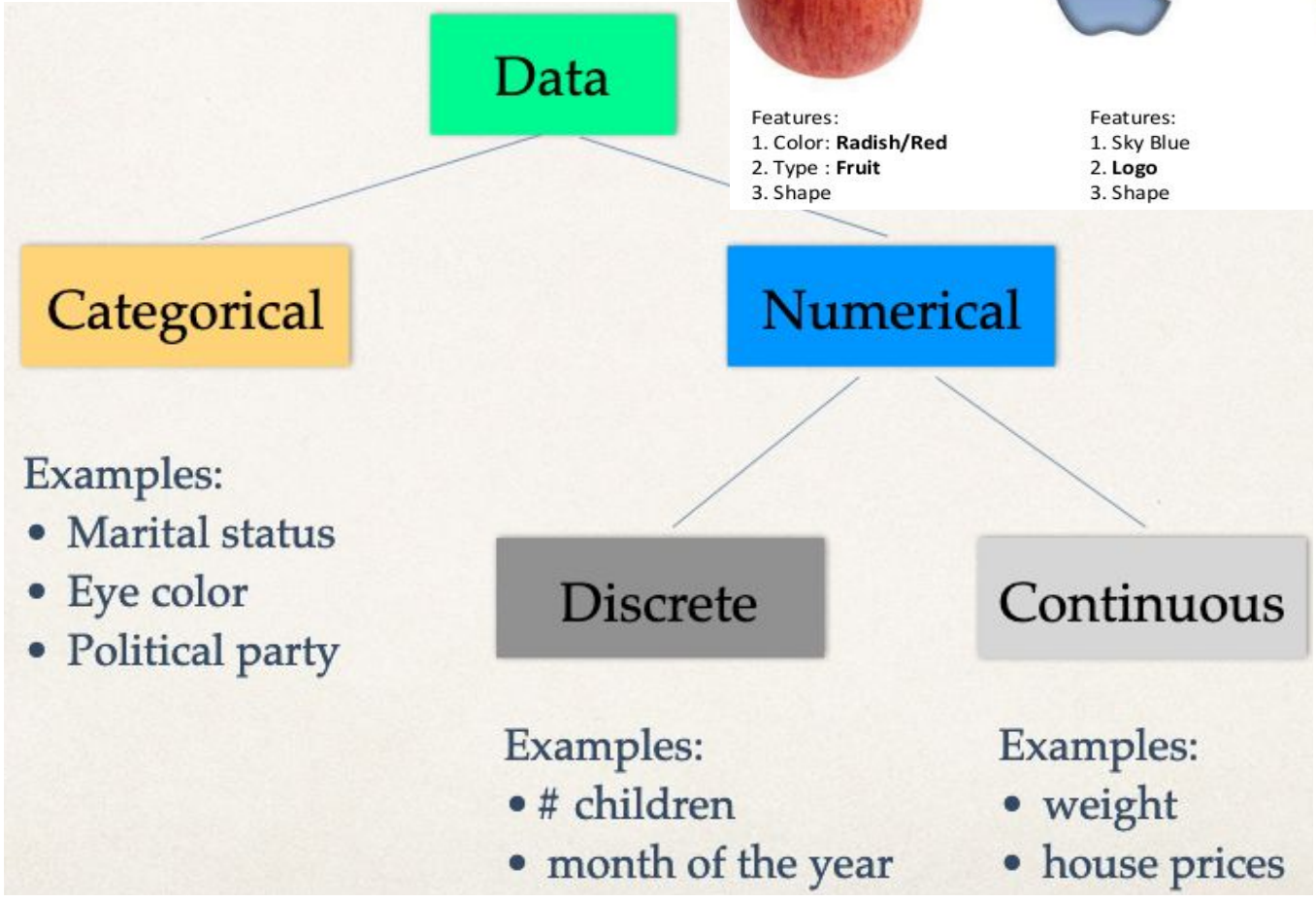

Feature extraction
Feature selection
Data mining
Interpretation and evaluation
Preprocessed data
Transformed data
Reduced data
Patterns
Knowledge

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 types

Typically strings



Features:
1. Color: **Radish/Red**
2. Type : **Fruit**
3. Shape

Features:
1. Sky Blue
2. **Logo**
3. Shape

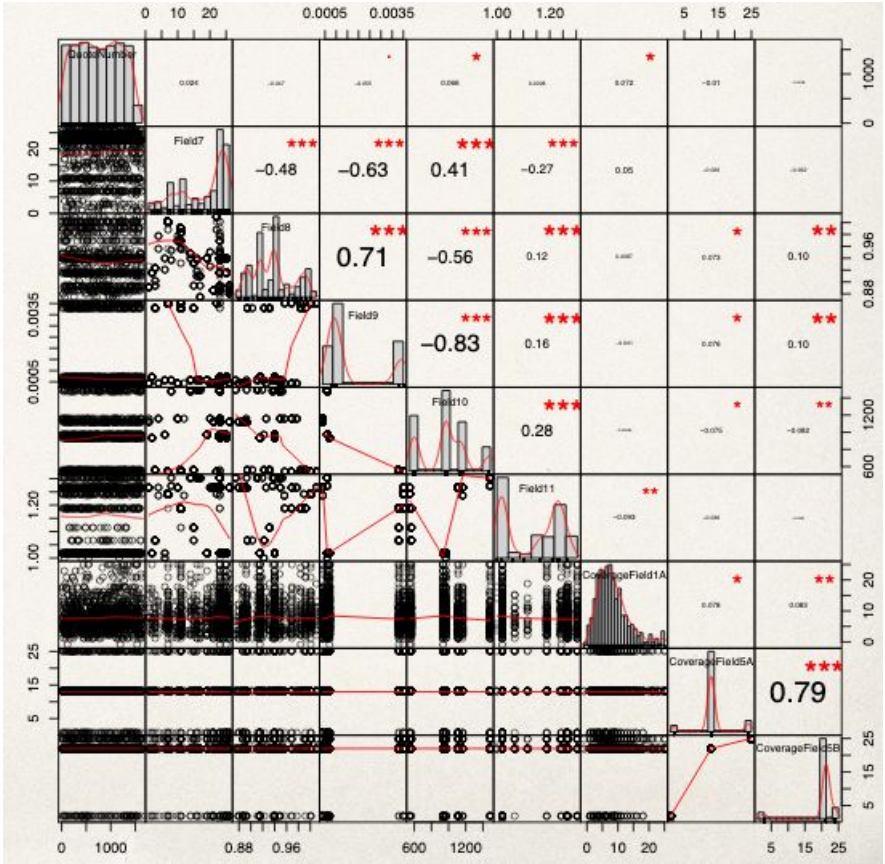
Features:
1. **Yellow**
2. **Fruit**
3. Shape

Numerical data

- Numerical data is already in a format easy to manage by mathematical models, but this doesn't mean that feature engineering is no longer necessary!
- Does the **magnitude** or the **scale** matters for the implemented model?
- What is the feature **distribution**?
  - For example the training process of a linear regression model assumes that features have a Gaussian distribution;
- Multiple features can be composed together into more complex features **Low level vs high level**
- In order to reduce the computational time, it may be necessary to **prune** the input features using automatic feature selection techniques

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



Data transformation

- 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 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

Feature scaling or normalization

- Models such as linear regression, logistic regression, neural networks, are affected by the scale of the input
- If the model is sensitive to the scale of the input features, feature scaling could help
- We can consider three types of scaling or normalization
  - **Min-Max** scaling
  - **Variance** scaling
  - **L2 normalization**

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

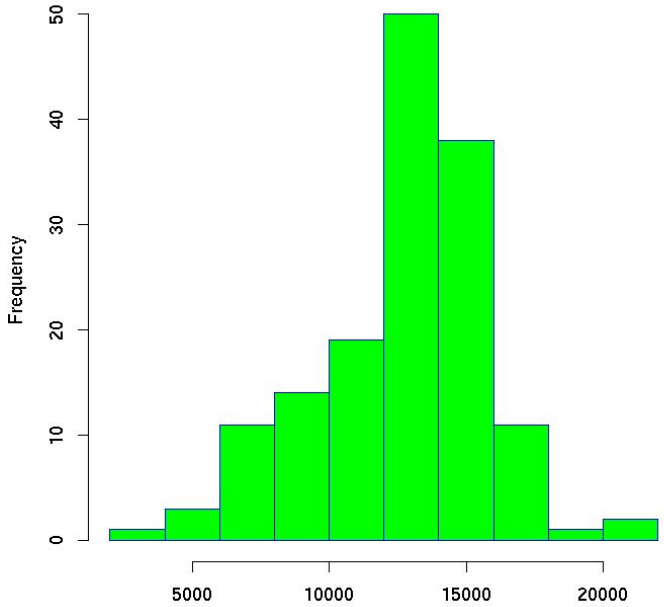
$$\tilde{x} = \frac{x}{\sqrt{x_1^2 + x_2^2 + \ldots + x_m^2}}$$

$$\tilde{x} = \frac{x - mean(x)}{\sqrt{var(x)}}$$

Important: feature scaling always divides the feature by a constant. Therefore, it does not change the shape of the single feature distribution!

Quantization or binning

- Raw counts that span several orders of magnitude are problematic for many models (for example linear models or clustering algorithms as k-means)
  - contain the scale by quantizing the count
  - group features into bins (like histograms)
    - **Fixed-width binning**: each bin contains the total counts from a specific numeric range
    - **Adaptive binning**: to avoid the problem that with a fixed-width solution many bins could be empty

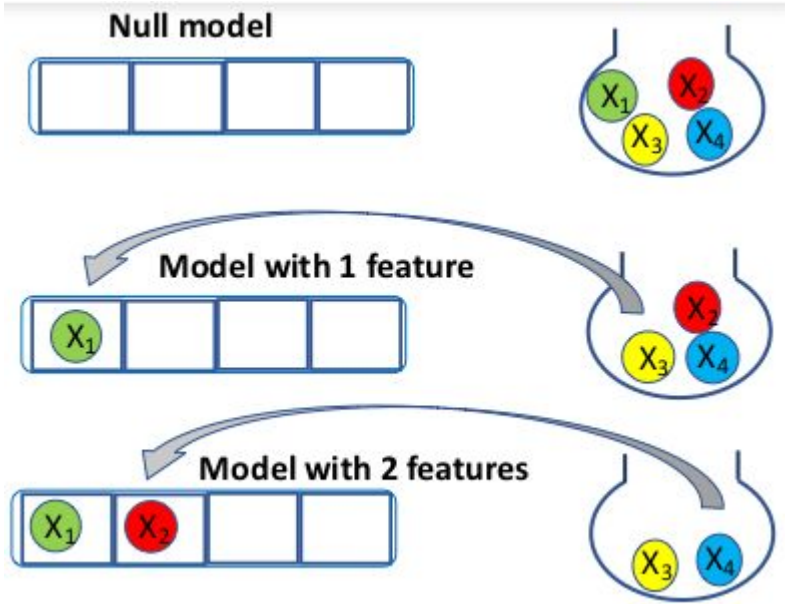


Feature selection

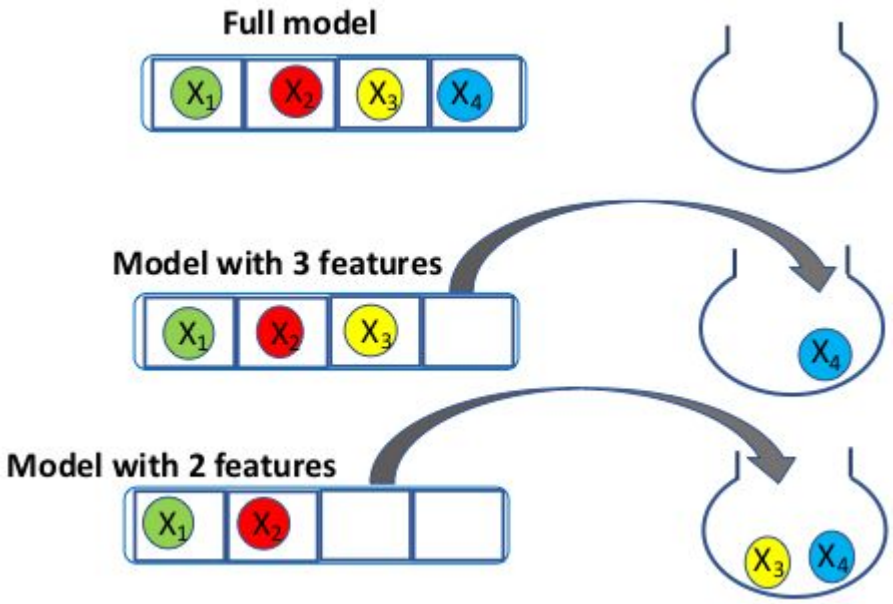
- Feature selection is a procedure that selects a subset of relevant and informative features to be used in a model
  - **Filter** methods
    - For example, remove one of two highly correlated variables
  - **Wrapper** methods
    - use a predictive model to score each feature subset used to train a model, which is tested on a control set
    - For example, stepwise regression or classification
  - **Embedded** methods
    - combine advantages of both Filter and Wrapper methods, but do not separate the learning process from the feature selection process (embedded)

Stepwise regression/classification

Forward selection



Backward elimination

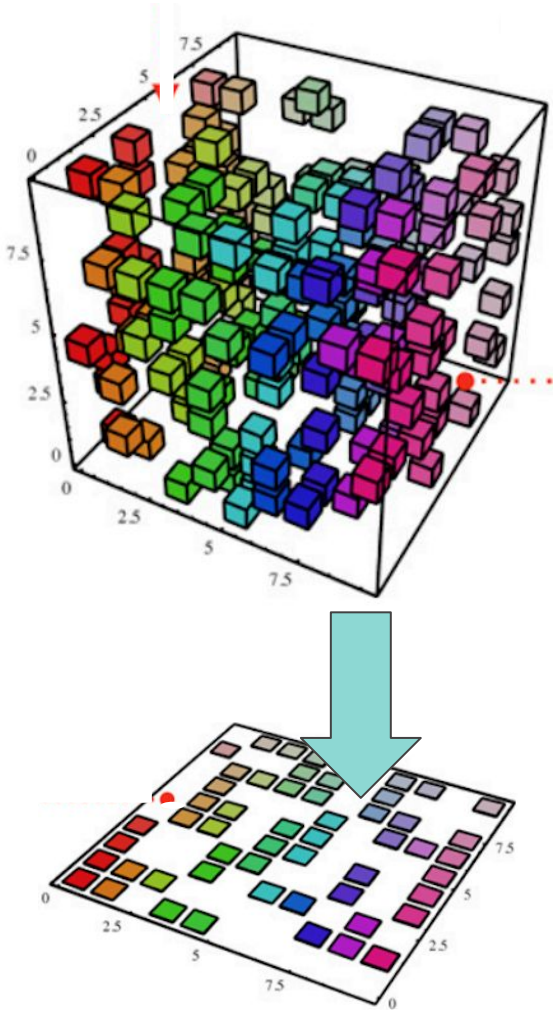


Dimensionality reduction

Why is dimension reduction something that we are interested in doing?

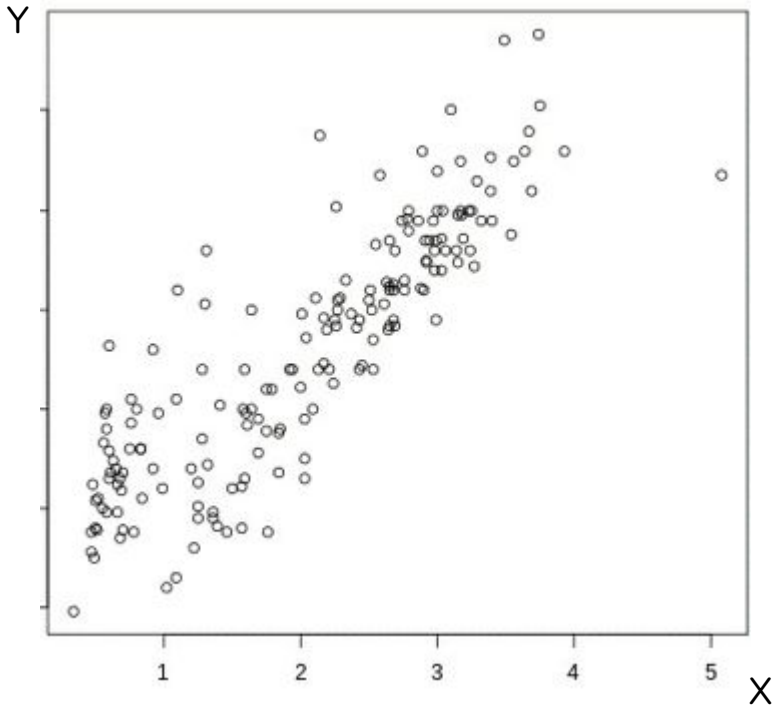
One reason may be for the sake of compressing data. If a dataset is particularly large, it may be useful to reduce the dimensions of the data

A more interesting reason for dimension reduction is that it provides insights into the underlying structure of the data



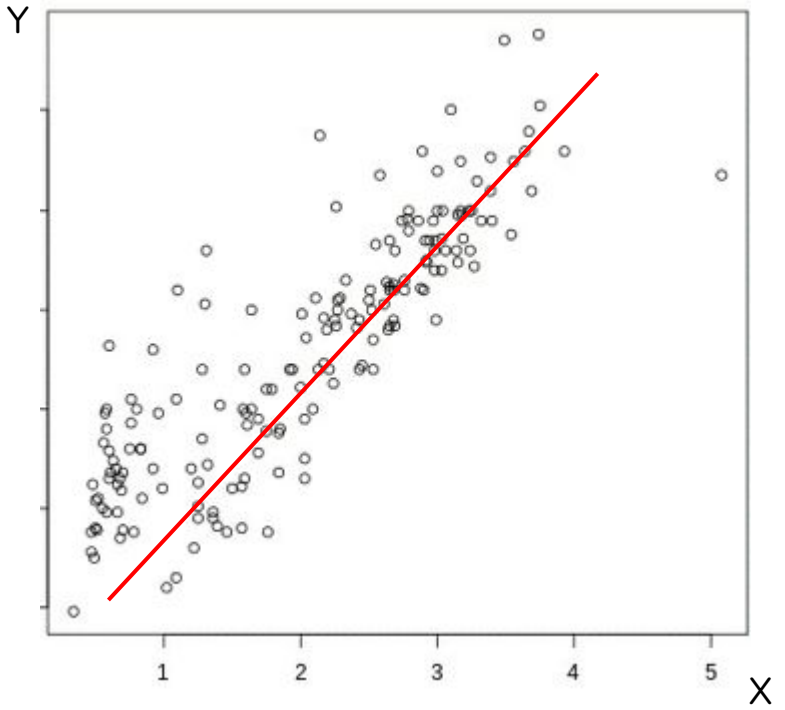
Example

- 2-dimension dataset with strong correlation between X and Y



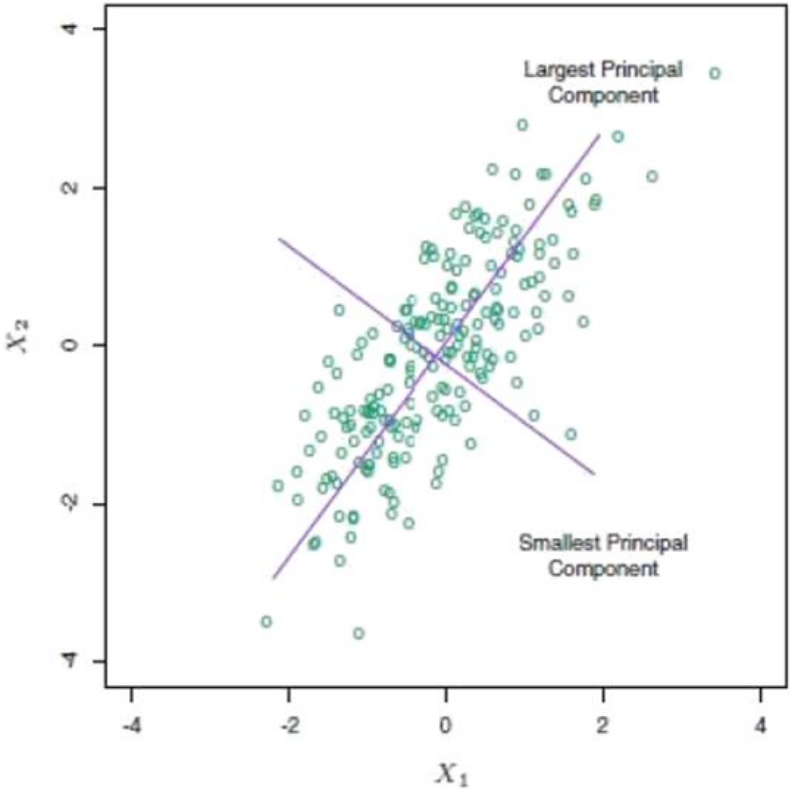
Example

- 2-dimension dataset with strong correlation between X and Y
- We can replace Y with a linear fit
- Then the input dataset is left with only 1 dimension (X)



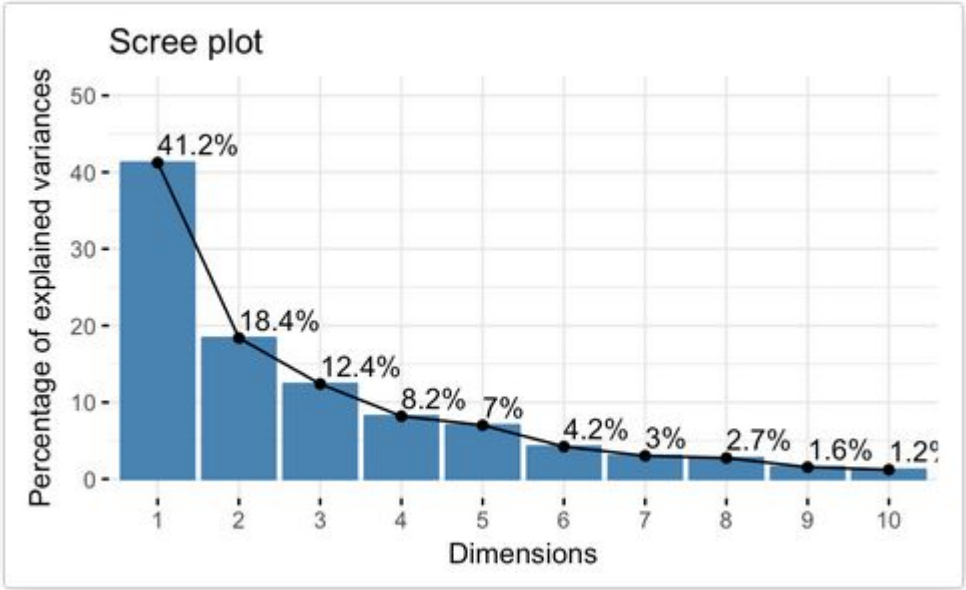
Principal Component Analysis (PCA)

- PCA is a technique that can be used to simplify a dataset:
  - linear transformation that chooses a new coordinate system such that greatest variance by any projection of the dataset comes to lie on the first axis (then called the first principal component);
  - the second greatest variance on the second axis, and so on



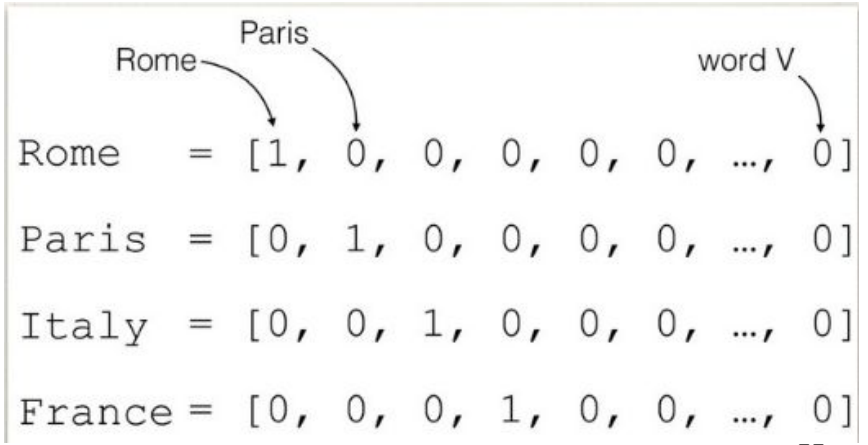
Principal Component Analysis (PCA)

- A **scree plot** is used to assess the relative importance of the new dimensions
- PCA can be used for reducing dimensionality by eliminating the last principal components
- *IMPORTANT: data must be normalized before using PCA*



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

Mean of [1,1,0]

mean_y* random

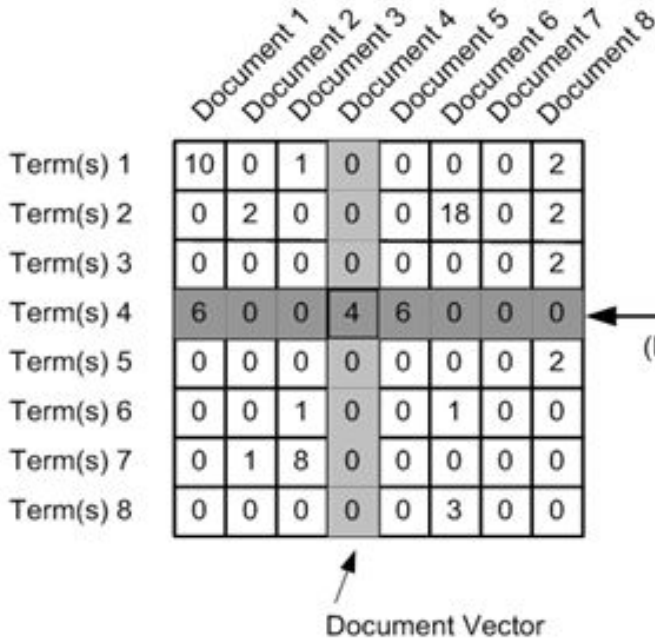
Split	UserID	Y	mean_y	random	newID
Train	A1	~~0~~	0.667	1.05	0.70035
Train	A1	1	0.333	0.97	0.32301
Train	A1	1	0.333	0.98	0.32634
Train	A1	0	0.667	1.02	0.68034
Test	A1	-	0.5	1	0.5
Test	A1	-	0.5	1	0.5
Train	A2	0			

Word embedding

- representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning
  - you define a dimension of word vector up-front
  - A way to capture multi-dimensional relationships between categories
  - 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

Frequency based word embedding

- **Count Vector**
  - Corpus C of D documents {d1,d2.....dD} 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 X 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)

Prediction based word embedding

- Use neural networks or other ML algorithms to train the model to find the best representation of embedded variables
- **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