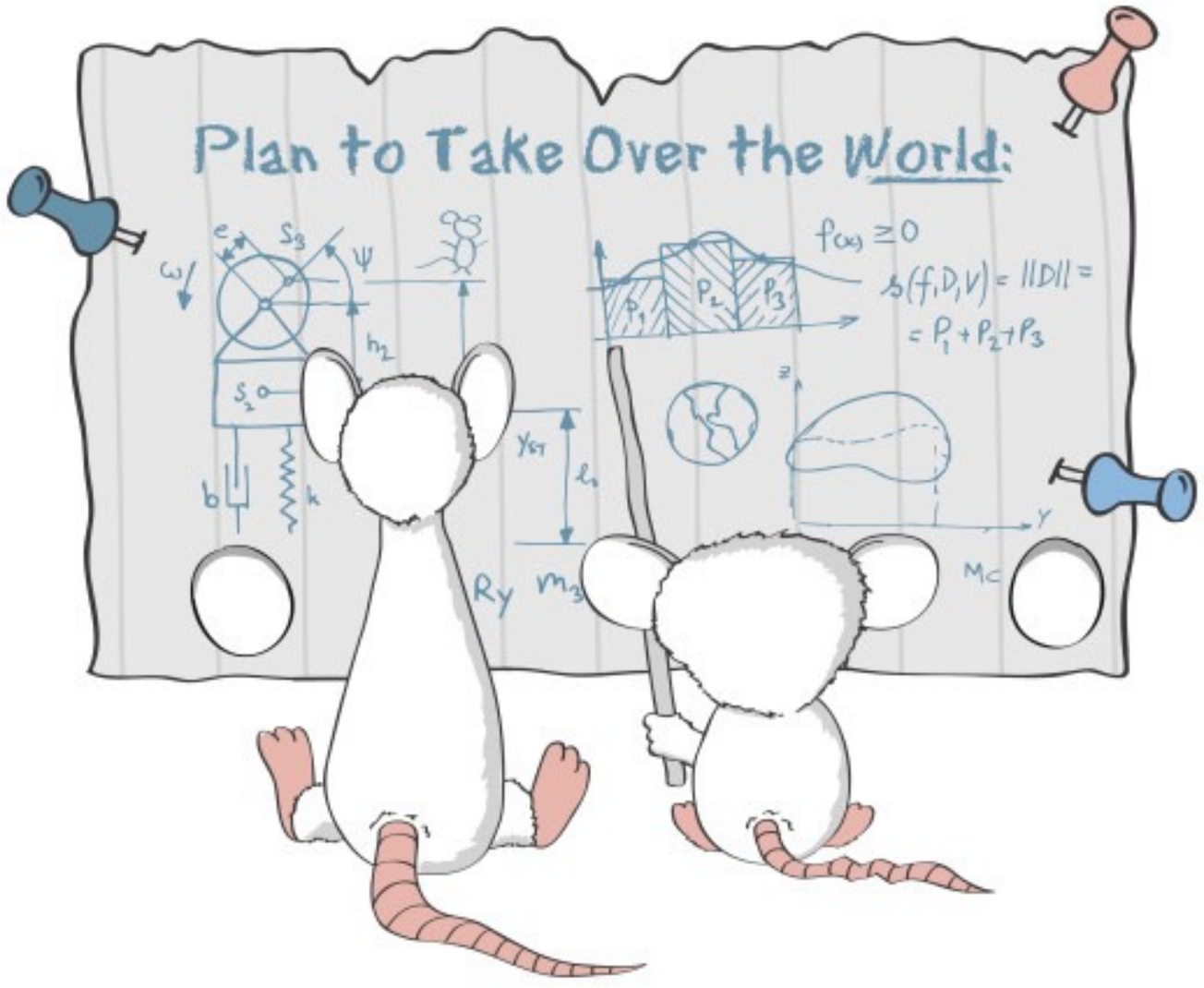


Work to Machine!

F5611 Machine Learning for Astronomers
by Martin Topinka

https://github.com/toastmaker/f5611-ML4A

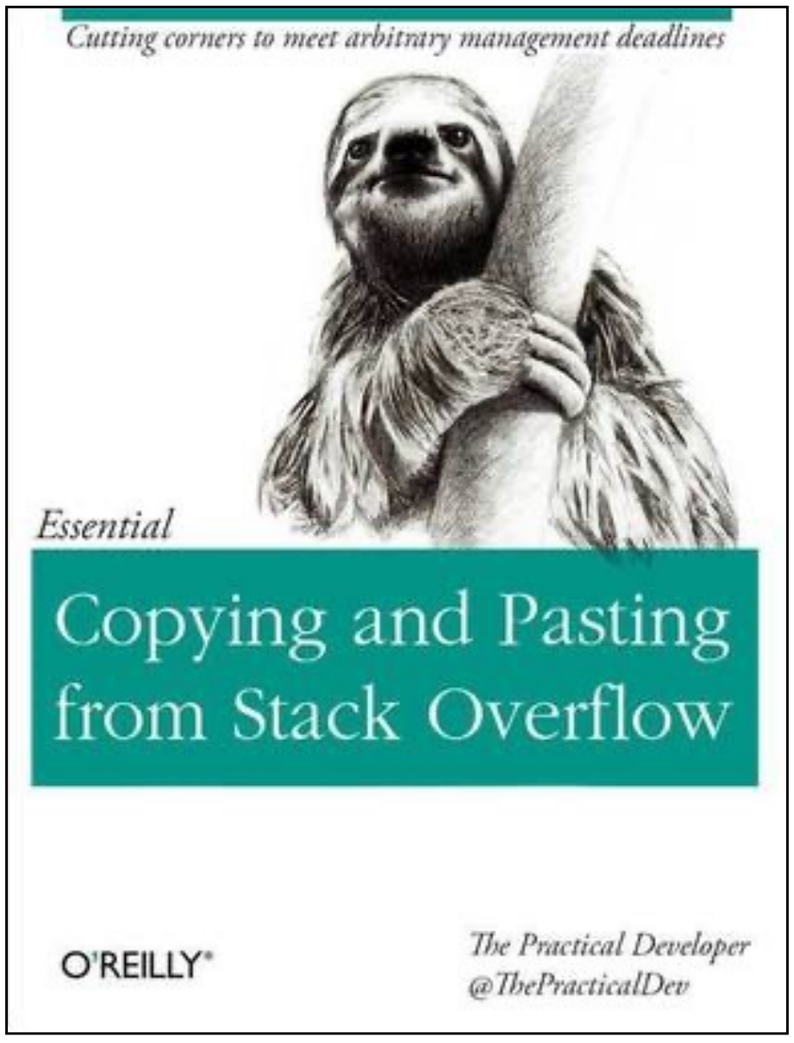
Every Thursday 15:00-16:50, 1/1

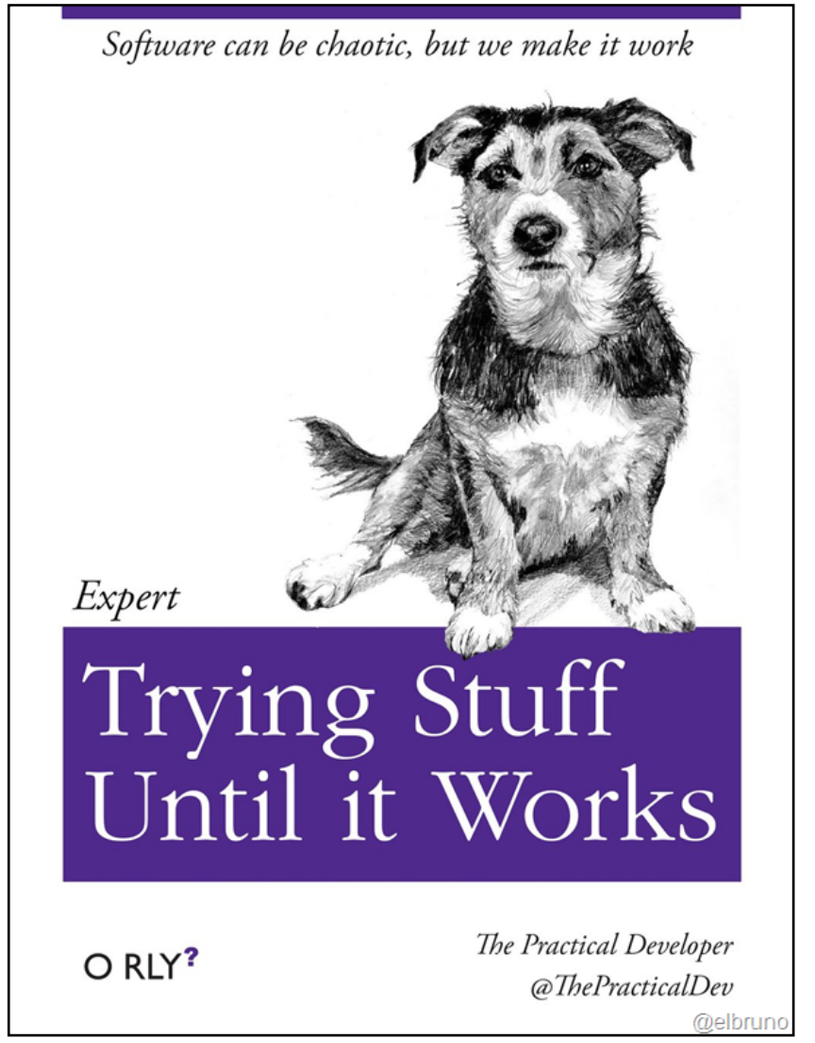
https://is.muni.cz/predmet/sci/podzim2021/F5611

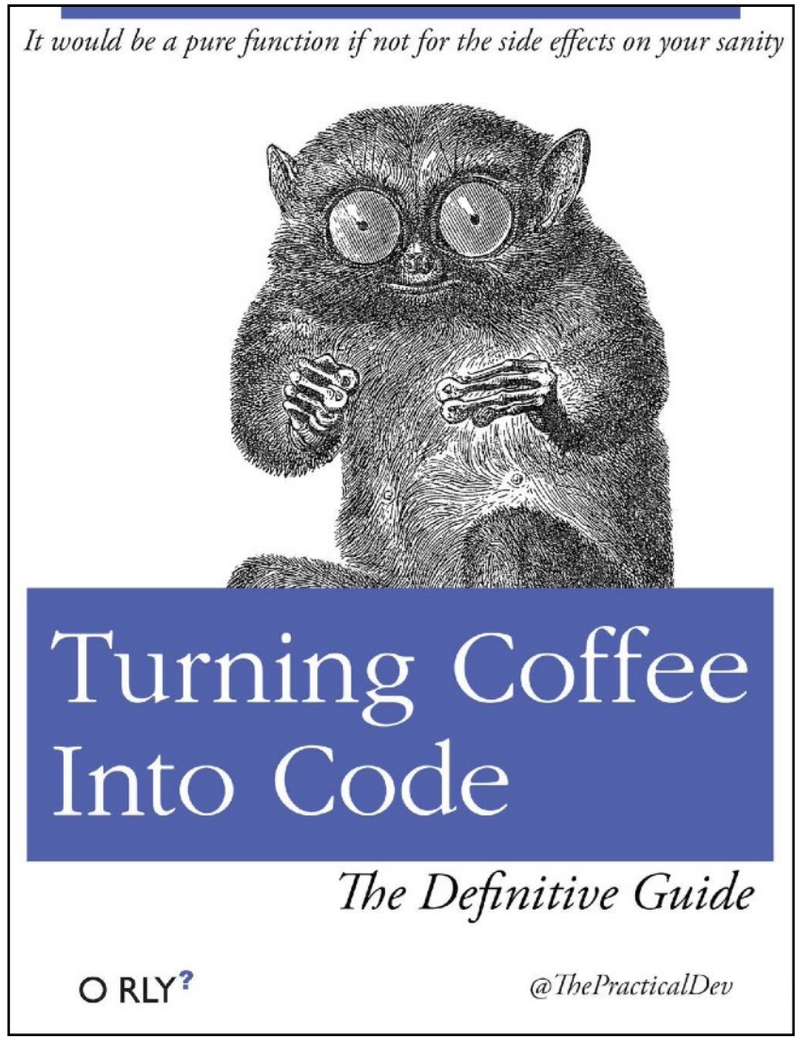
233480@mail.muni.cz

Assisted by **Matej Kosiba** and **Tomáš Plšek**

- (Lecture + Q&A) + (Hands-on Session — labs)
- Project => Zápočet (good attendance may help)
- Guest lecture/seminar
- (This is not either a Python course, either a theoretical course)







- Yaser Abu-Mostafa (Caltech): *Learning From Data*, 2012
- Željko Ivezič, Andrew Connolly, Jake Vanderplas: *Statistics, Data mining and Machine Learning in Astronomy*, 2014
- Aurelien Geron: *Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow*, 2019
- Francois Chollet: *Deep Learning with Python*, 2017
- https://www.deeplearningbook.org/

Further Reading/Watching

- scikit-learn.org documentation, great gallery, tons of examples

- Andrew Ng Standford course (I passed it and it's really good, the code is in Matlab)
  https://www.coursera.org/learn/machine-learning

  http://cs229.stanford.edu/syllabus-autumn2018.html
  http://cs231n.stanford.edu/

- Yaser Abu-Mostafa Caltech course — #1 course online (in my humble opinion, I passed it, it requires no coding, very good but simple math explanation)
  https://work.caltech.edu/telecourse.html

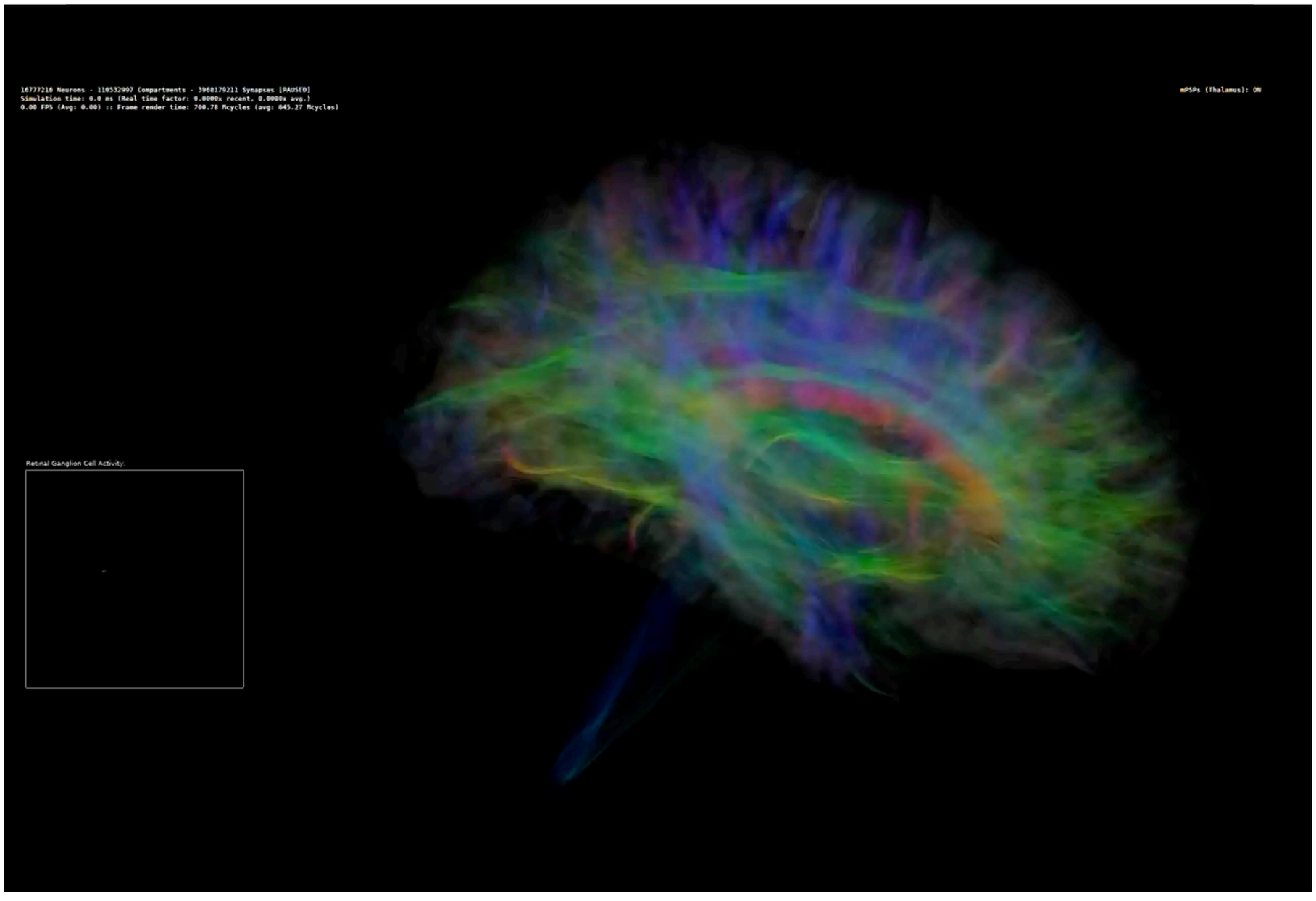
- MIT 6.S191 Introduction to Deep Learning
  http://introtodeeplearning.com

Syllabus

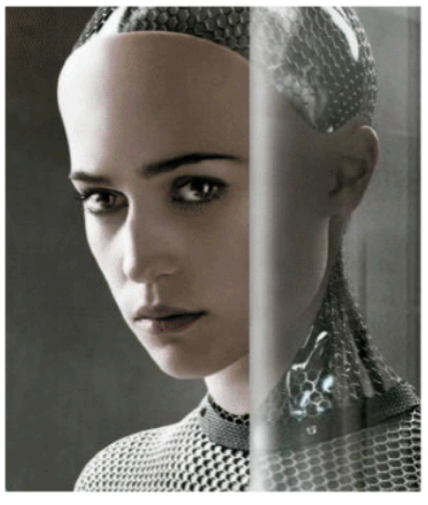
- Introduction to machine learning, history...
- Principles of machine learning
- Supervised, unsupervised machine learning
- Classification vs regression
- Loss function, accuracy measures
- Bias-variance tradeoff
- Curse of dimensionality
- Python based software for machine learning
- Basic machine learning algorithms (SVM, KNN, K-mean, Logistic regression, Decision Trees, Random Forest)
- Feature selection, data reduction (PCA)
- Advanced algorithms (bagging, boosting, voting)
- Introduction to scikit-learn
- First touch of scikit-learn API
- Hands on session scikit-learn with GRB classification, QSO's vs stars…
- Model validation, hyper-parameter fine tuning
- Imbalanced classes
- Neural network, perceptron
- Deep learning neural networks
- Regularisation, dropout
- Deep learning with Convolutional Neural Networks
- Encoder-Decoder, Auto-encoder
- GAN
- Training data generators
- Introduction to Keras/TensorFlow
- Hands on session in Keras (developing a NN to classify stars/QSOs; developing a deep convNN auto-encoder for finding transients)
- Optional: Gaussian Processes

“AI began with an ancient wish to forge the gods.”

- Pamela McCorduck, *Machines Who Think, 1979*



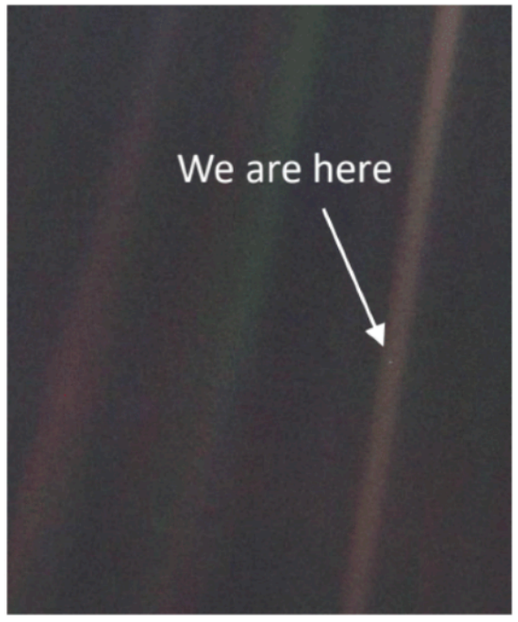
Frankenstein (1818)



Ex Machina (2015)

Visualized here are 3% of the neurons and 0.0001% of the synapses in the brain.

History of Deep Learning Ideas and Milestones*

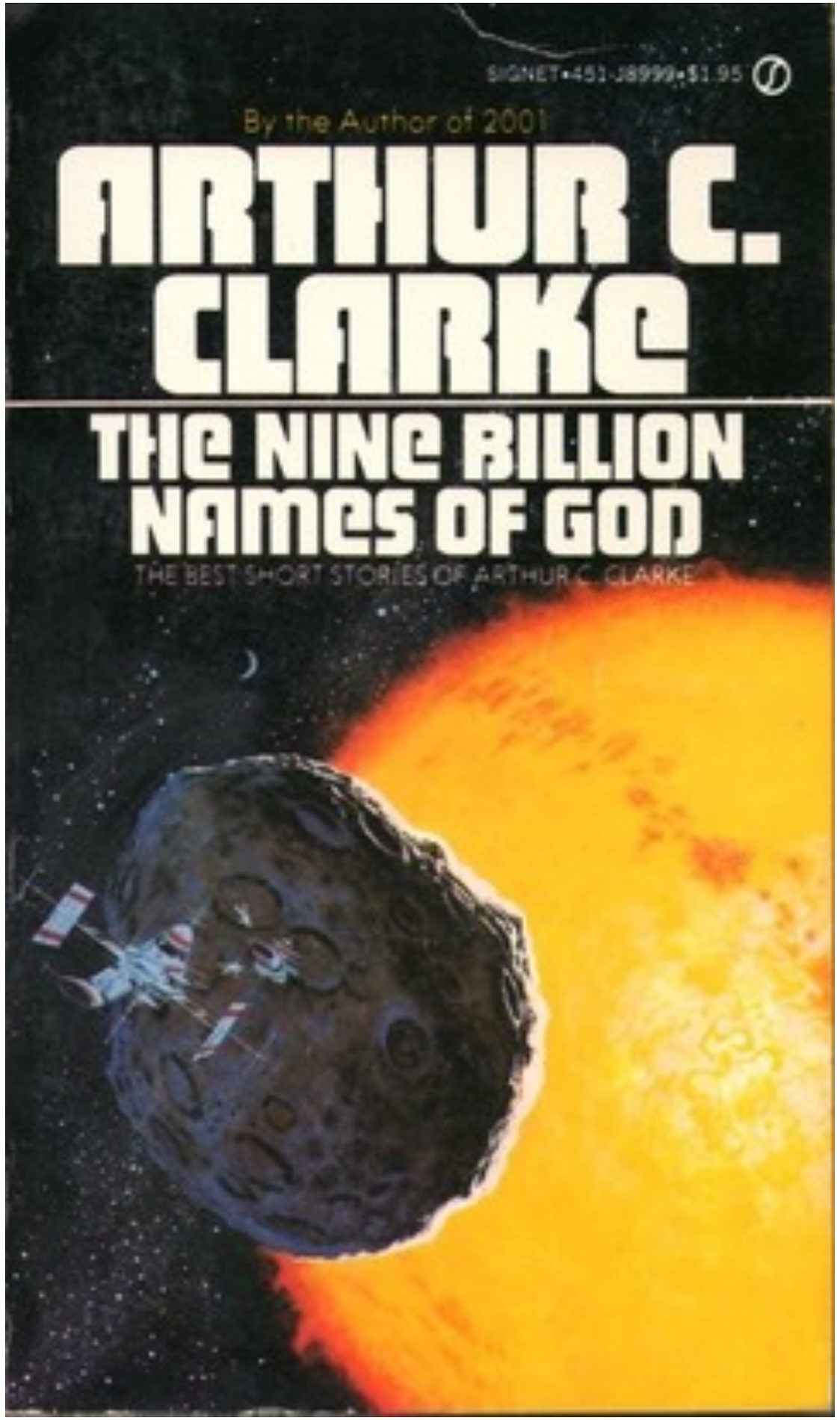


Perspective:

- Universe created
  13.8 billion years ago
- Earth created
  4.54 billion years ago
- Modern humans
  300,000 years ago
- Civilization
  12,000 years ago
- Written record
  5,000 years ago

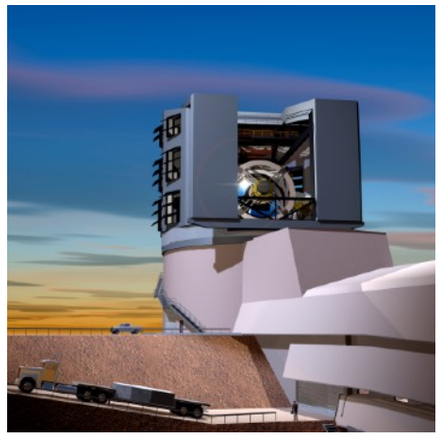
- 1943: Neural networks
- 1957: Perceptron
- 1974-86: Backpropagation, RBM, RNN
- 1989-98: CNN, MNIST, LSTM, Bidirectional RNN
- 2006: “Deep Learning”, DBN
- 2009: ImageNet
- 2012: AlexNet, Dropout
- 2014: GANs
- 2014: DeepFace
- 2016: AlphaGo
- 2017: AlphaZero, Capsule Networks
- 2018: BERT

* Dates are for perspective and not as definitive historical record of invention or credit


SIGNET•451-J8999•$1.95
By the Author of 2001
ARTHUR C. CLARKE
THE NINE BILLION NAMES OF GOD
THE BEST SHORT STORIES OF ARTHUR C. CLARKE

Big Data in a Nutshell

- Data volume doubles every 18 months (Moore's law)
- LSST:
  - 10 years long movie of the sky
  - 30’000 GB/night (size of entire SDSS)
  - 1’000’000 transient alerts/night in differential imaging (faster and more reliable than catalog cross-matching)
  - 50% rubbish
  - Human time/attention does **not** scale :-(



Automated real-time classification is needed!

The Fourth Paradigm: Data-Intensive Scientific Discovery

Era of “Data-driven discoveries”

3Vs':

- Volume (large archives)
- Velocity (continuous flow)
- Variety (complexity)

LSST

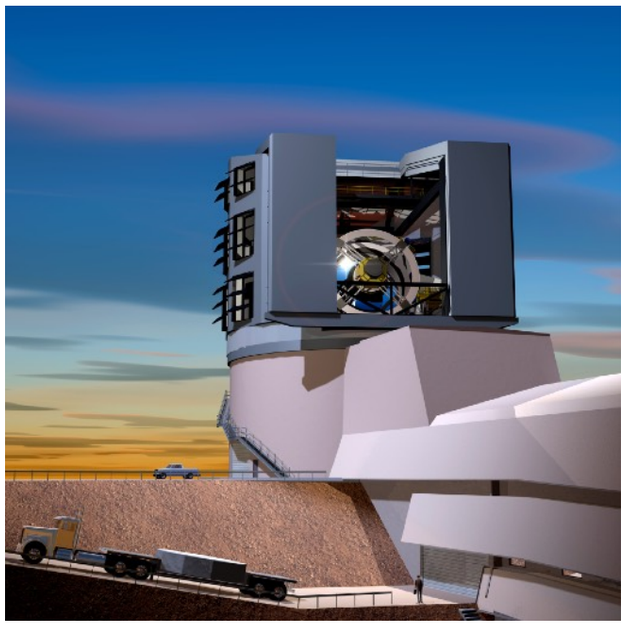
DR11 37 10^9 objects, 7 10^{12} sources,
5.5 million 3.2 Gigapixel images

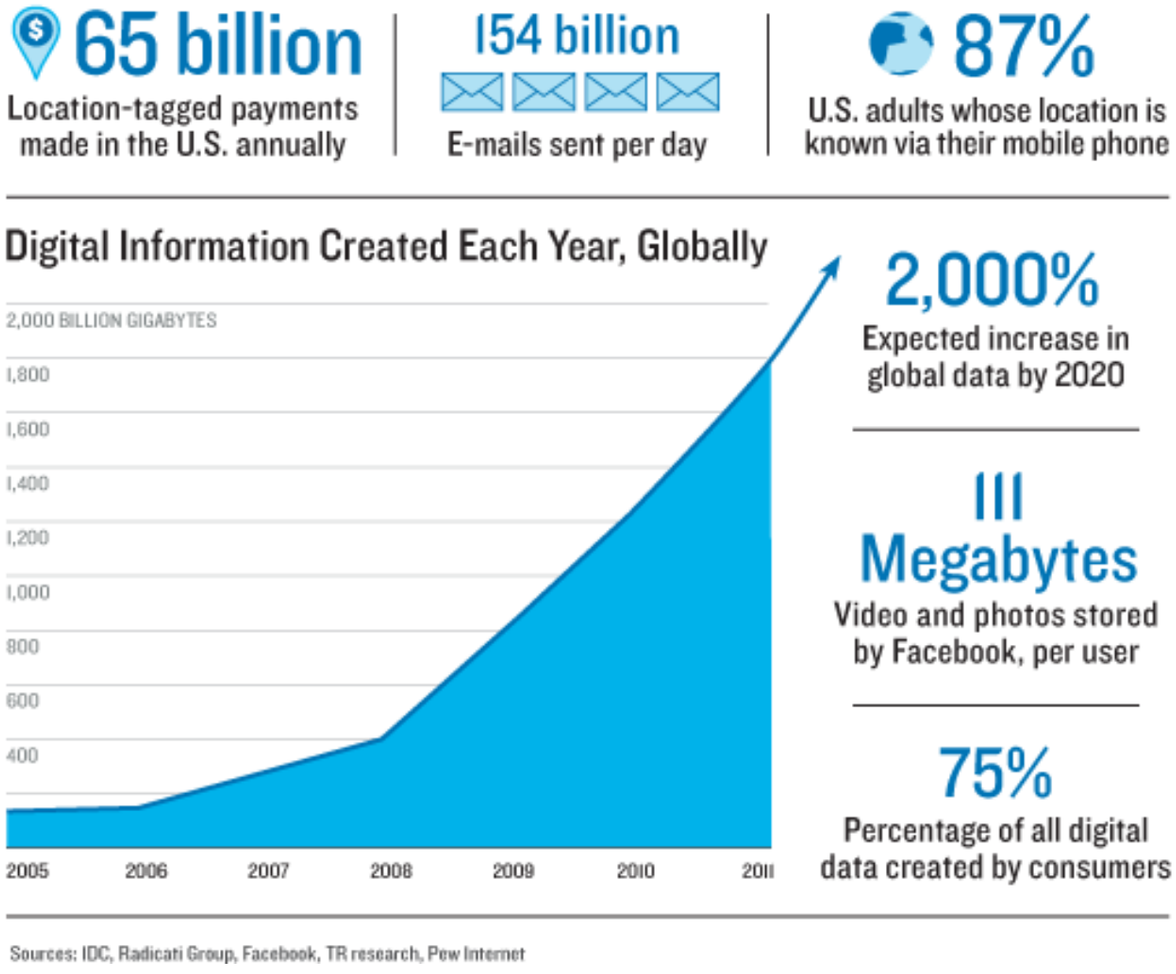
30 terabytes of data nightly

Final volume of raw image data = 60 PB

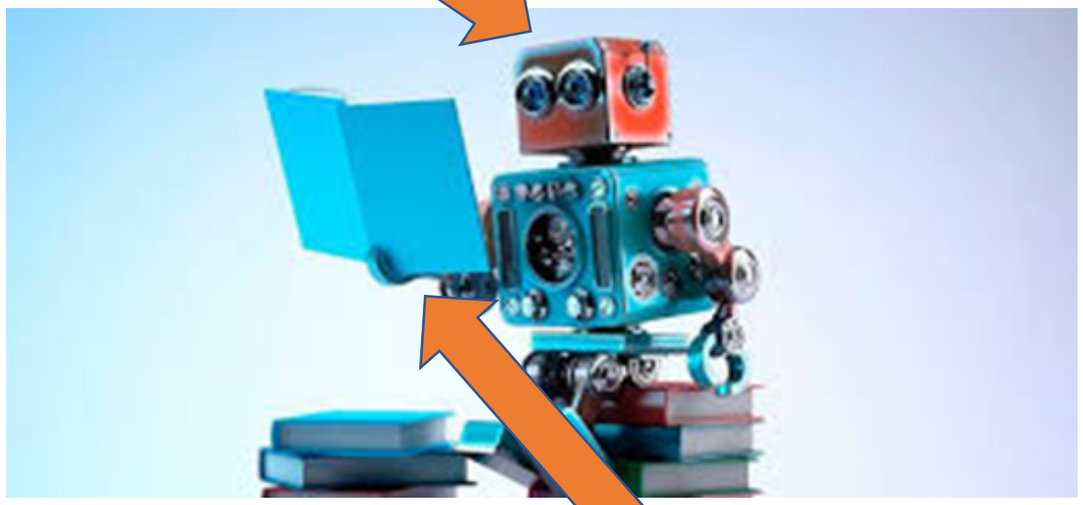
Final image collection (DR11) = 0.5 EB

Final catalog size (DR11) = 15 PB

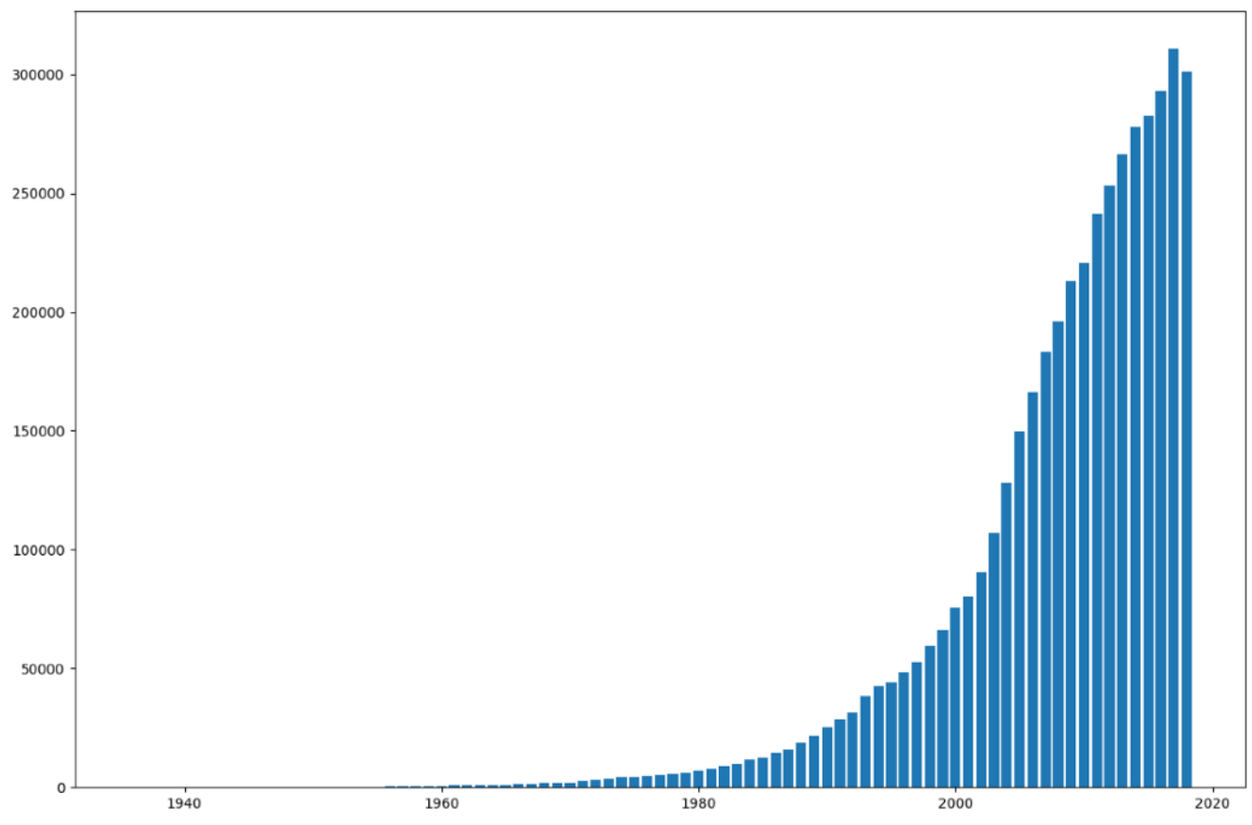




Machine…



… Learning?!?



Number of ML papers on arXiv.org

Garik Kasparov vs IBM's Deep Blue
(1997)

Lee Se-dol vs DeepMind Alpha Go
(2016)

What is Machine Learning?

Machine learning is the process to automatically extract knowledge from data, usually with the goal of making predictions on new, unseen data.

Central to machine learning is the concept of making decision automatically from data, without the user specifying explicit rules how this decision should be made.

The second central concept is **generalisation**. The goal of a machine learning algorithm is to predict on new, previously unseen data. We are not interested in marking an email as spam or not, that the human already labeled. Instead, we want to make the users life easier by making an automatic decision for new incoming mail.

ML is the set of algorithms with tuneable parameters that can learn and adjust the values of these parameters from previously seen data and generalising for predictions of new yet unseen data.

ML has many names: Pattern Recognition, Statistical Data Modelling, Learning from Data, Deep Learning, Computational Statistics

Some use-cases

- Self-driving cars
- Fraud detection in banks
- Spam/ham
- Learn cooking by watching you-tube videos
- Object recognition in an image
- All sort of classification tasks when you know classes
- Clustering, new class detection, outlier detections, exploring multi-dim spaces, finding correlations, finding groups
- Replacing parts of a complicated simulation with ML engine
- Search for transits in exoplanets, time-series prediction
- Artificial real-looking catalogue building
- Denoising, data-compressing
- Feature selection

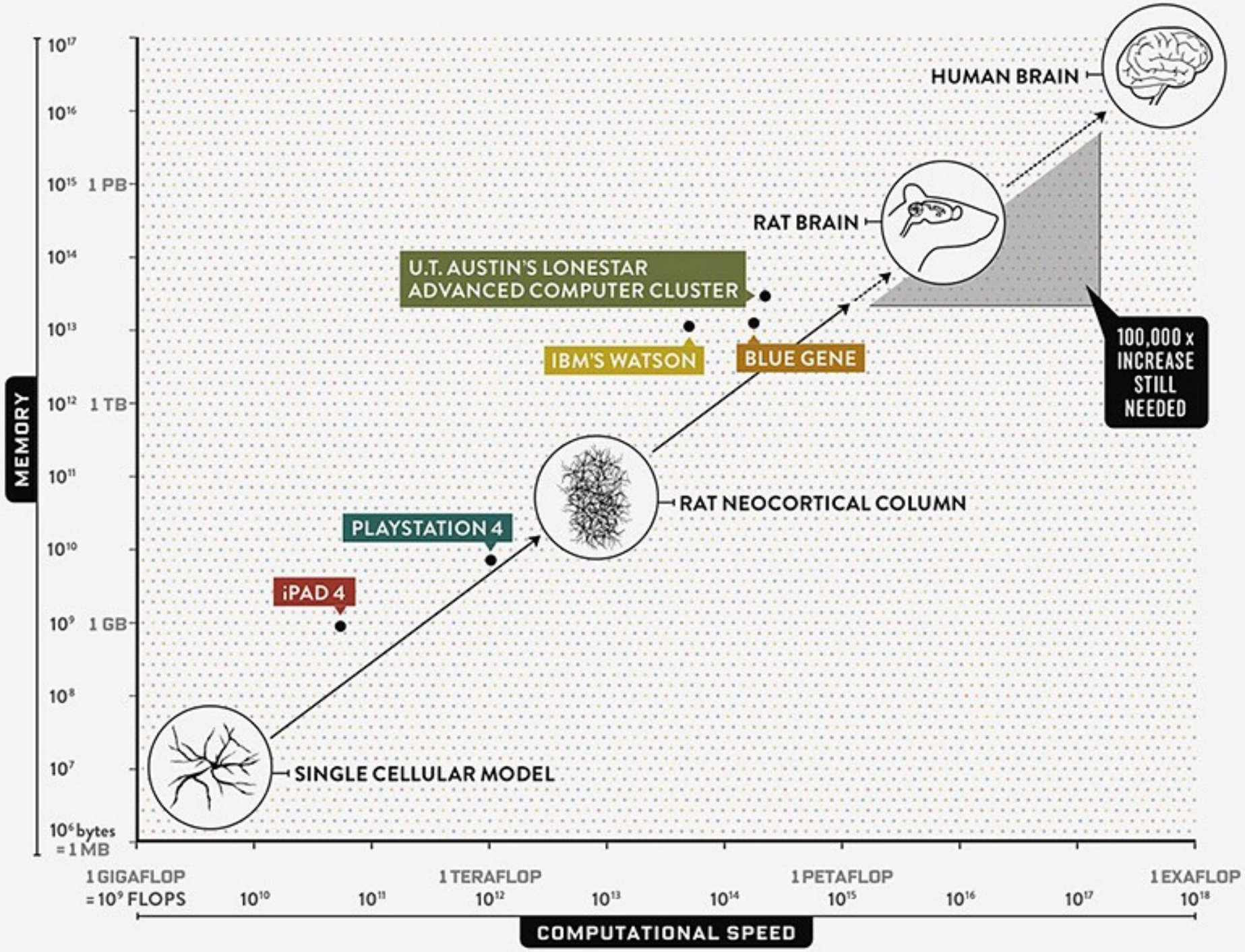
- Distinguish short GRBs flares from SGR flares from INTEGRAL
- Search for GRB afterglows in SDSS
- GRBs photometric redshift GROND
- Clustering of spectra (minimal spanning tree) in the distance-from-templates space in DFBS
- Search for M-stars, Carbon stars in DFBS
- JWST high-z: galaxy morphology classification
- JWST high-z: search for bars in galaxies
- JWST high-z: mock up catalog generation
- GRB localisation

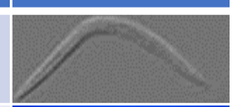
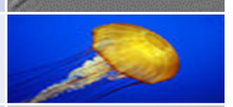
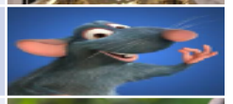
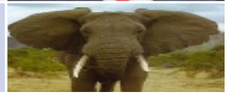
Machine learning is the science of how to teach computers to make decision.

Machine learning gives computes the ability to learn without being explicitly programmed.

Machine learning algorithms can figure out how to perform tasks by generalising from examples (experience).

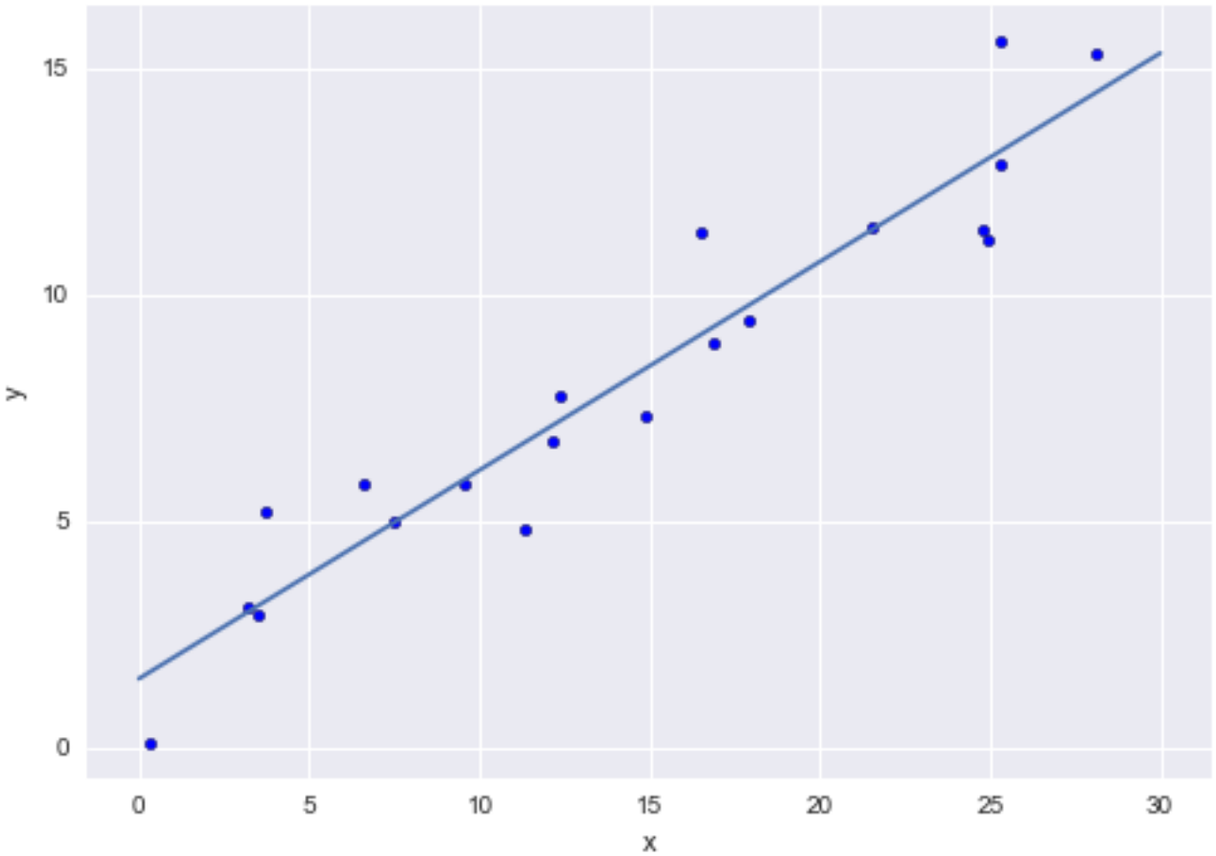
If we could easily sense more than 3 dimensions we would probably not need machine learning.


MEMORY
10^{17}
10^{16}
10^{15} 1 PB
10^{14}
10^{13}
10^{12} 1 TB
10^{11}
10^{10}
10^{9} 1 GB
10^{8}
10^{7}
10^{6} bytes = 1 MB
HUMAN BRAIN
RAT BRAIN
U.T. AUSTIN'S LONESTAR ADVANCED COMPUTER CLUSTER
IBM'S WATSON
BLUE GENE
100,000 x INCREASE STILL NEEDED
RAT NEOCORTICAL COLUMN
PLAYSTATION 4
iPAD 4
SINGLE CELLULAR MODEL
1 GIGAFLOP = 10^{9} FLOPS
10^{10}
10^{11}
1 TERAFLOP
10^{12}
10^{13}
10^{14}
1 PETAFLOP
10^{15}
10^{16}
10^{17}
1 EXAFLOP
10^{18}
COMPUTATIONAL SPEED

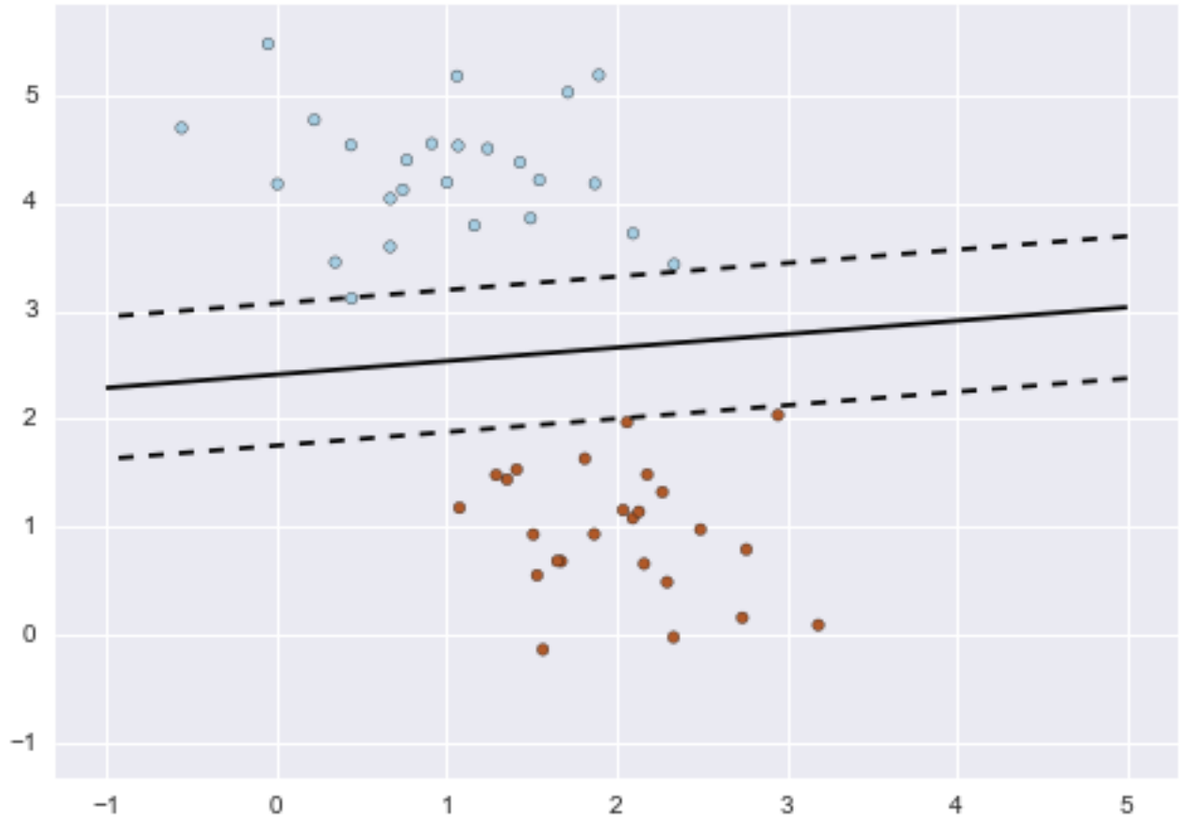
Name	# of neurons / # of synapses	Visuals
Caenorhabditis elegans	302	
Hydra vulgaris	5,600	
Homarus americanus	100,000	
Blatta Orientalis	1,000,000	
Nile Crocodile	80,500,000	
Digital Reasoning NN (2015)	**~86,000,000 (est.) / 1.6E11**	
Rattus Rattatouillensis	200,000,000	
Blue and yellow macaw	1,900,000,000	
Chimpanzee	28,000,000,000	
Homo Sapiens Sapiens	**86,000,000,000 / 1.5E14**	
African Elephant	257,000,000,000	

- **Supervised** - classification, regression (fitting)
- **Unsupervised** - clustering, graphs/trees, transformations, typically in multi-dim, outlier detection
- **Semi-supervised**, genetic algorithms, GANs...

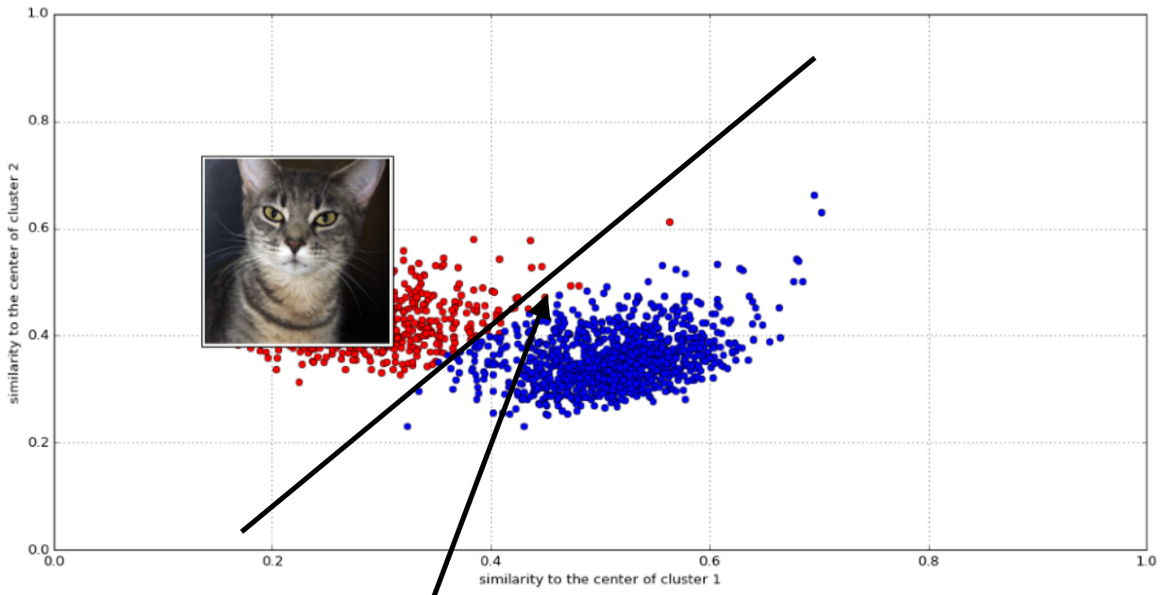
- regression (fitting)

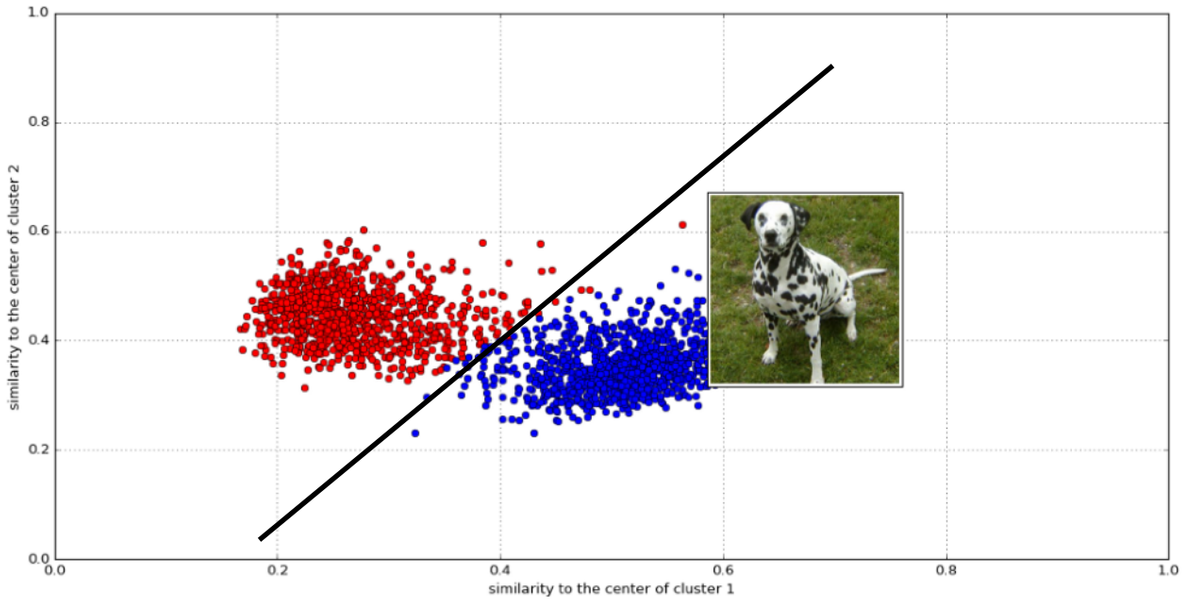


- classification



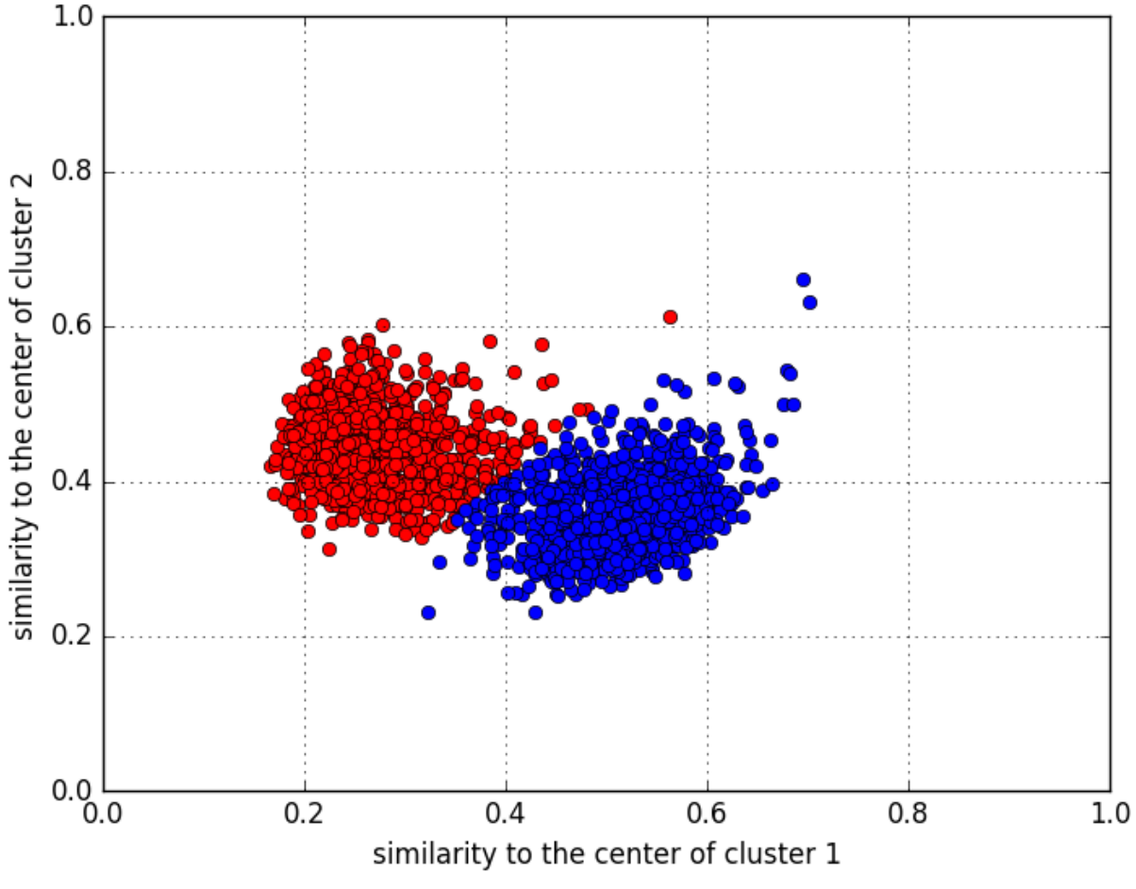
- may look simple, but imagine it in a high dim vector space





?

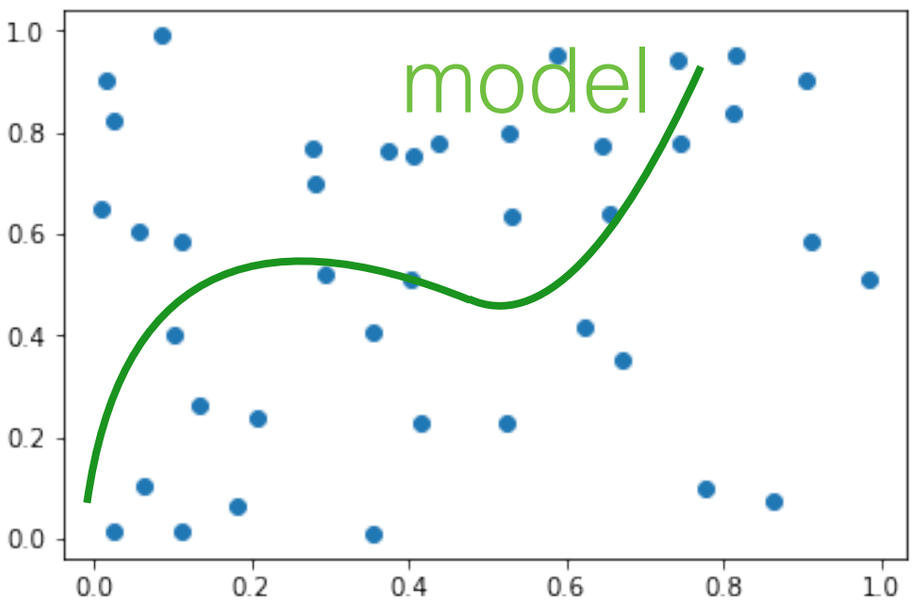
Trying two clusters

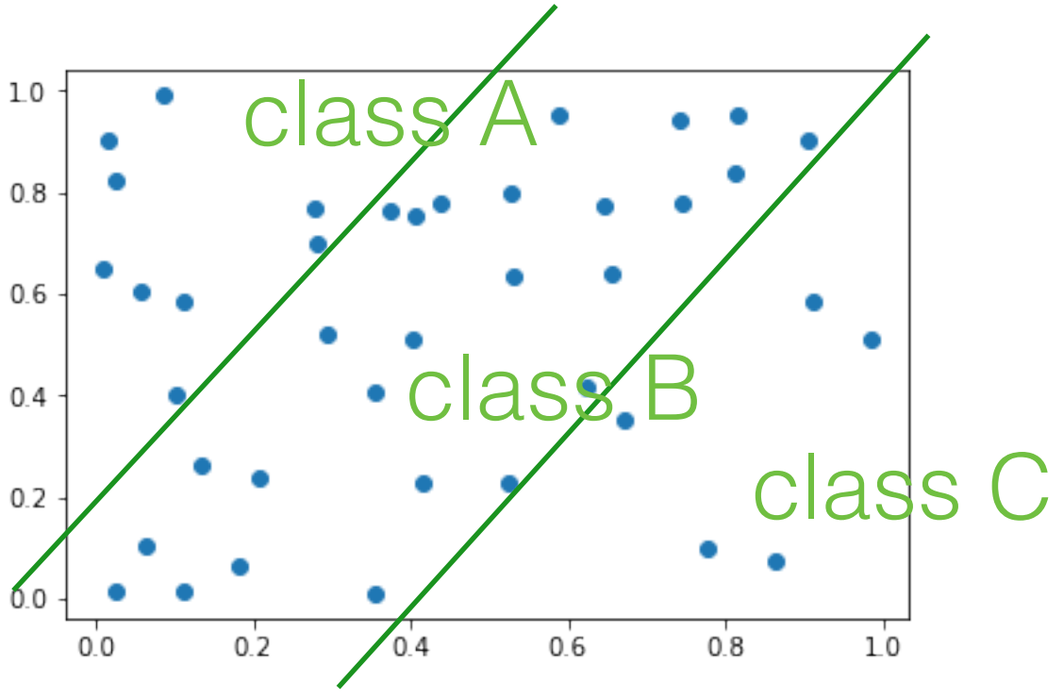


The difference between a physicist and an astronomer:

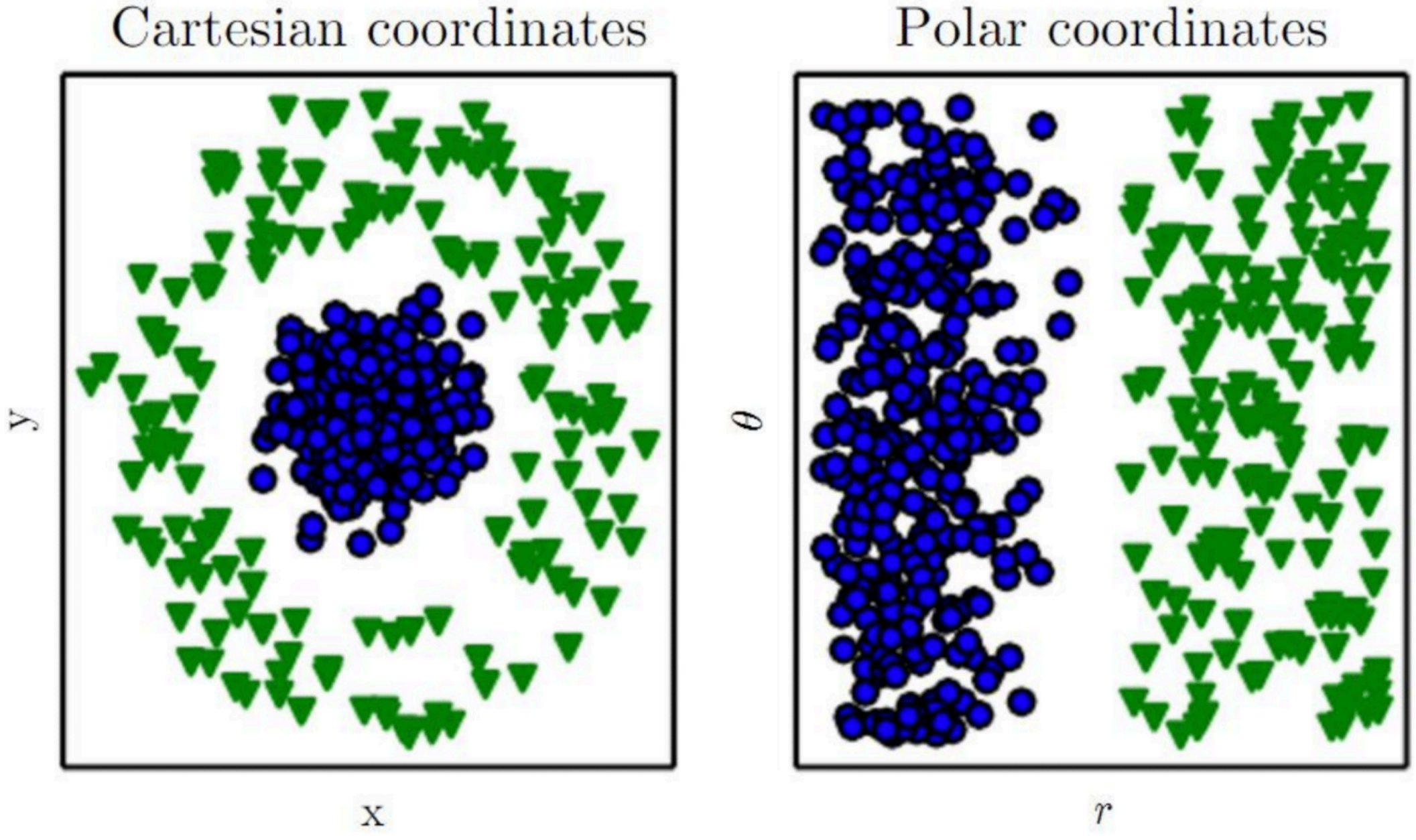
The physicist sees random 2D data and draws a curved line in it saying it's the model that describes the data. The astronomer draws two parallel lines, saying these points belong to class A, these to class B and this is class C.

—- Andy Lawrence, private communication





Representation Matters



Task: Draw a line to separate the **green triangles** and **blue circles**.

Regression vs Classification

- Regression: the result is a (real) number
  Example: distance to a galaxy, stellar temperature, an age of a person in an image

- Classification: the result is a label of a group
  Example: AGN vs star

- Often it can be interchanged and thought as regression-like
  Example: cold vs hot star instead of $T < 5000$ K vs $T > 5000$ K, short GRBs vs long GRBs instead of $t < 2$ s vs $t > 2$ s

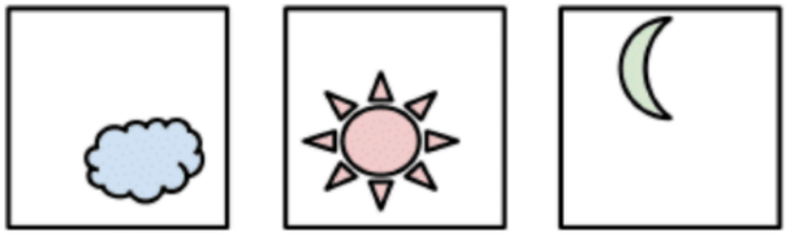
- Classification is typically used when the division is rough (sunny | cloudy | rainy), and it's **always** used when the **metric** in the number representation is **useless or misleading**
  Example: Type 1 supernovae are not smaller either larger than Type 2 supernovae, it's just a label

Multi-Class vs Multi-Label

Multi-Class

C = 3

Samples



Labels (t)

[0 0 1] [1 0 0] [0 1 0]

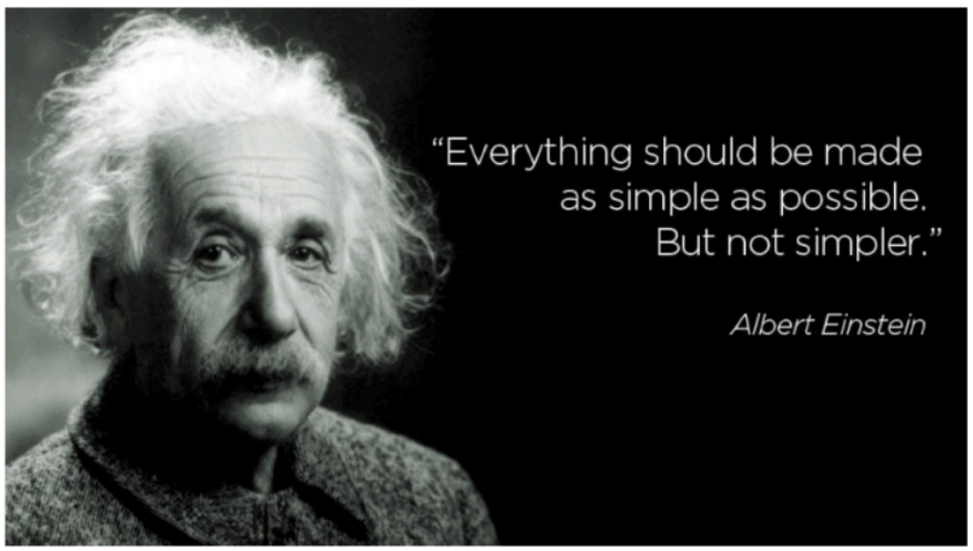
Multi-Label

Samples

Labels (t)

[1 0 1] [0 1 0] [1 1 1]

First Steps: **Start Simple**



Input Image:

TensorFlow Model:

Neural Network

Output:

Spiral galaxy

(with 87% confidence)

1

2

3

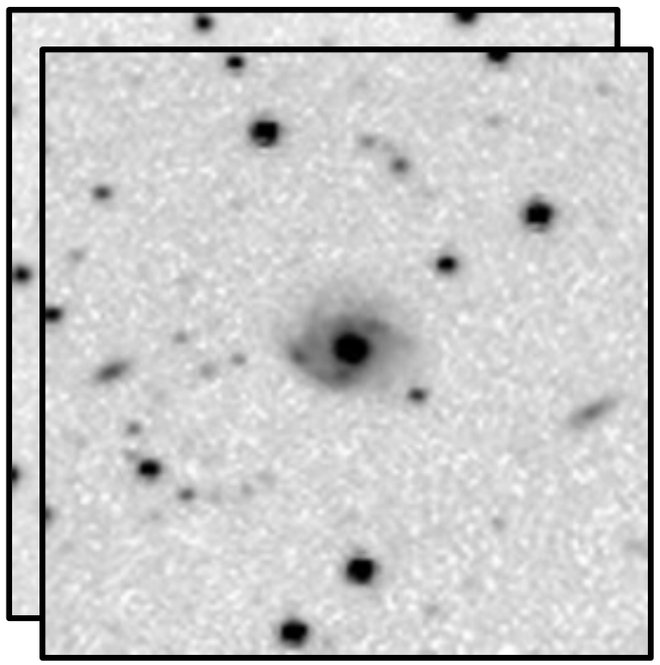
4

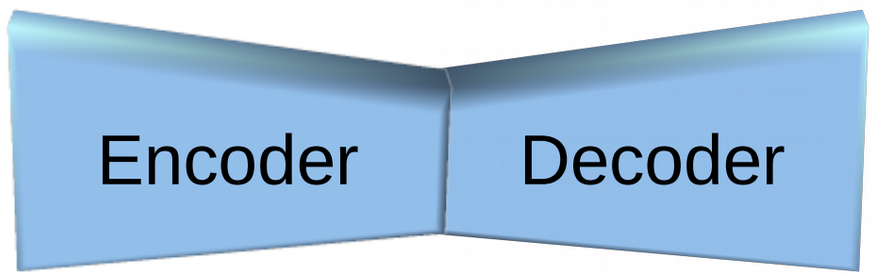
5

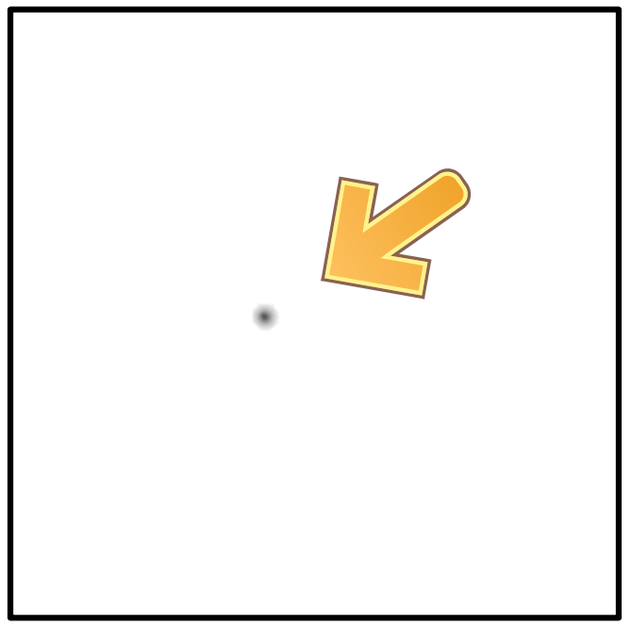
6

```python
# import tensorflow and keras (tf.keras not "vanilla" Keras)
import tensorflow as tf
from tensorflow import keras

# get data
(train_images, train_labels), (test_images, test_labels) = \
keras.datasets.mnist.load_data()

# setup model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer=tf.train.AdamOptimizer(),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# train model
model.fit(train_images, train_labels, epochs=5)

# evaluate
test_loss, test_acc = model.evaluate(test_images, test_labels)
print('test accuracy:', test_acc)

# make predictions
predictions = model.predict(test_images)
```

Select an area to comment on




Encoder
Decoder



Input
Conv1
7x7 : 128
Conv2
5x5 : 256
Conv3
5x5 : 512
Conv3_1
3x3 : 512
Conv4
3x3 : 1024
Conv4_1
3x3 : 1024
Conv5
3x3 : 1024
Conv5_1
3x3 : 1024
Conv6
3x3 : 2048
Conv6_1
3x3 : 2048
L1 Loss
Output
Upconv0
4x4 : 32
Upconv1
4x4 : 64
Upconv2
4x4 : 128
Upconv3
4x4 : 256
Upconv4
4x4 : 512
Upconv5
4x4 : 1024

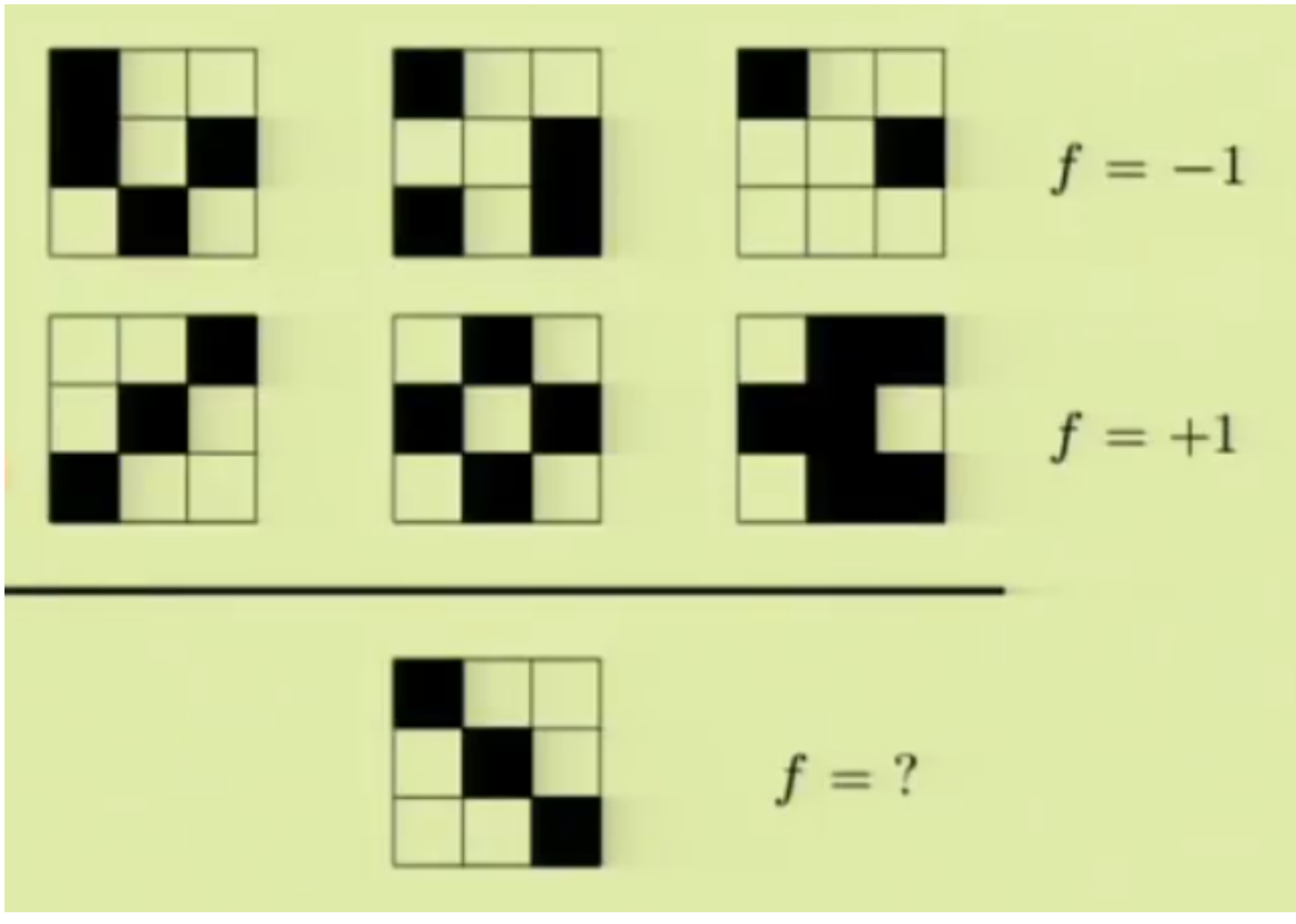
60 %	Data preparation (searching, query, downloading, formatting, cleaning, normalising)
10 %	ML coding
30 %	Fine tuning and interpretation

The art of **asking the right question**

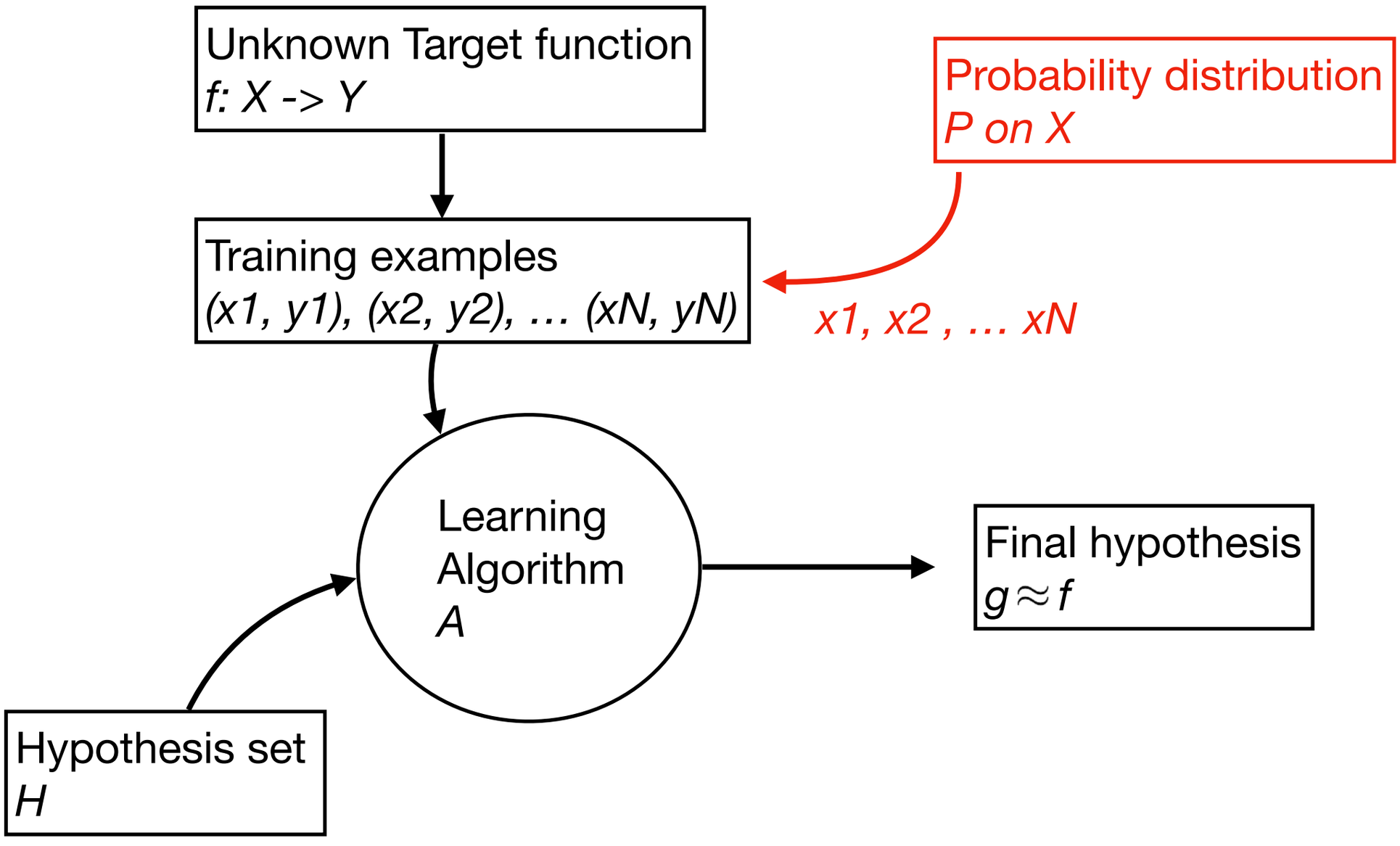
Essentials of Learning

- Pattern must exist
- Mapping “target function” is unknown or expensive to calculate
- We have the data (and computing resources…)
- Data sample is representative

Target function...



Learning Diagram



Python is fast
for writing, testing and developing code because it is

- **interpreted**
- **dynamically typed**
- **high-level**

```
/* Hello world in Java */
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello world");
   }
}
```

vs

```
# Hello world in Python
print("Hello world")
```

not speaking of C where a string is a null-terminated byte array...

Python is slow
Because it is a high-level, interpreted and dynamically-typed language. Each operation requires overhead. Many repeated operations (in loop) becomes significant.
Luckily, there is NumPy!
It works with arrays in chunks of memory like in C. Moreover, it provides slicing, index masking etc...

Name	Use	Logo
Pandas	Data Analysis	pandas $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$
Spark	Distributed Computing	Spark
Scikit-learn	Machine Learning Toolbox	scikit learn
Keras	Deep Learning	Keras A deep learning library
TensorFlow	Deep Learning	TensorFlow
Open-cv	Computer Vision	OpenCV

There are also other machine learning software packages, some even in Python, e.g. Facebook uses pyTorch… but we will KISS (Keep It Simple!)

Software prerequisites

- Python 3

- Anaconda anaconda.org with Python 3.7

- You may create an environment
  `conda create --name=ml python=3.6 anaconda`
  then to jump into the freshly installed environment
  `source activate ml`
  at to end the session
  `source deactivate ml`

- More info on environments in anaconda:
  https://conda.io/docs/user-guide/tasks/manage-environments.html

- *scikit-learn*, *pandas*, *scikit-image*, *seaborn*, *tensorflow*, *keras* (anaconda may suggest to downgrade some of the libraries to Python 3.6, but it's ok, the environment works as an independent container)
  `conda install xxx` or `conda install -c conda-forge xxx`

- `conda update conda ; conda update anaconda ; conda update xxx`

- Cheat sheets…

http://colab.research.google.com