

Work to Machine!

F5611 Machine Learning for Astronomers
by Martin Topinka

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

Every 2nd Tuesday 5 pm - 7pm, 0/1

<https://is.muni.cz/predmet/sci/podzim2020/F5611>

martin.topinka@gmail.com

Muni, HEA group, Math building, the top floor, the door on the very left

Assisted by Matej Kosiba

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

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

Software can be chaotic, but we make it work



Expert

Trying Stuff Until it Works

O RLY?

The Practical Developer
@ThePracticalDev
@elbunno

It would be a pure function if not for the side effects on your sanity



Turning Coffee Into Code

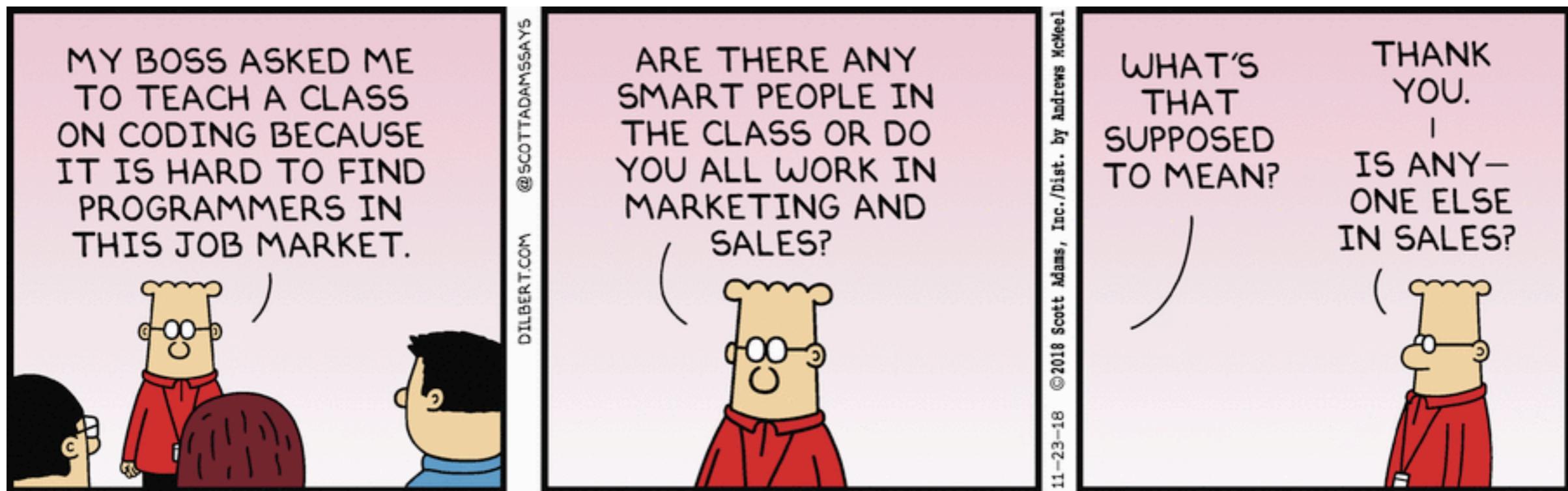
The Definitive Guide

O RLY?

@ThePracticalDev

- Yaser Abu-Mostafa (Caltech): *Learning From Data*, 2012
- Željko Ivezic, 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

Introduce yourself and your goals



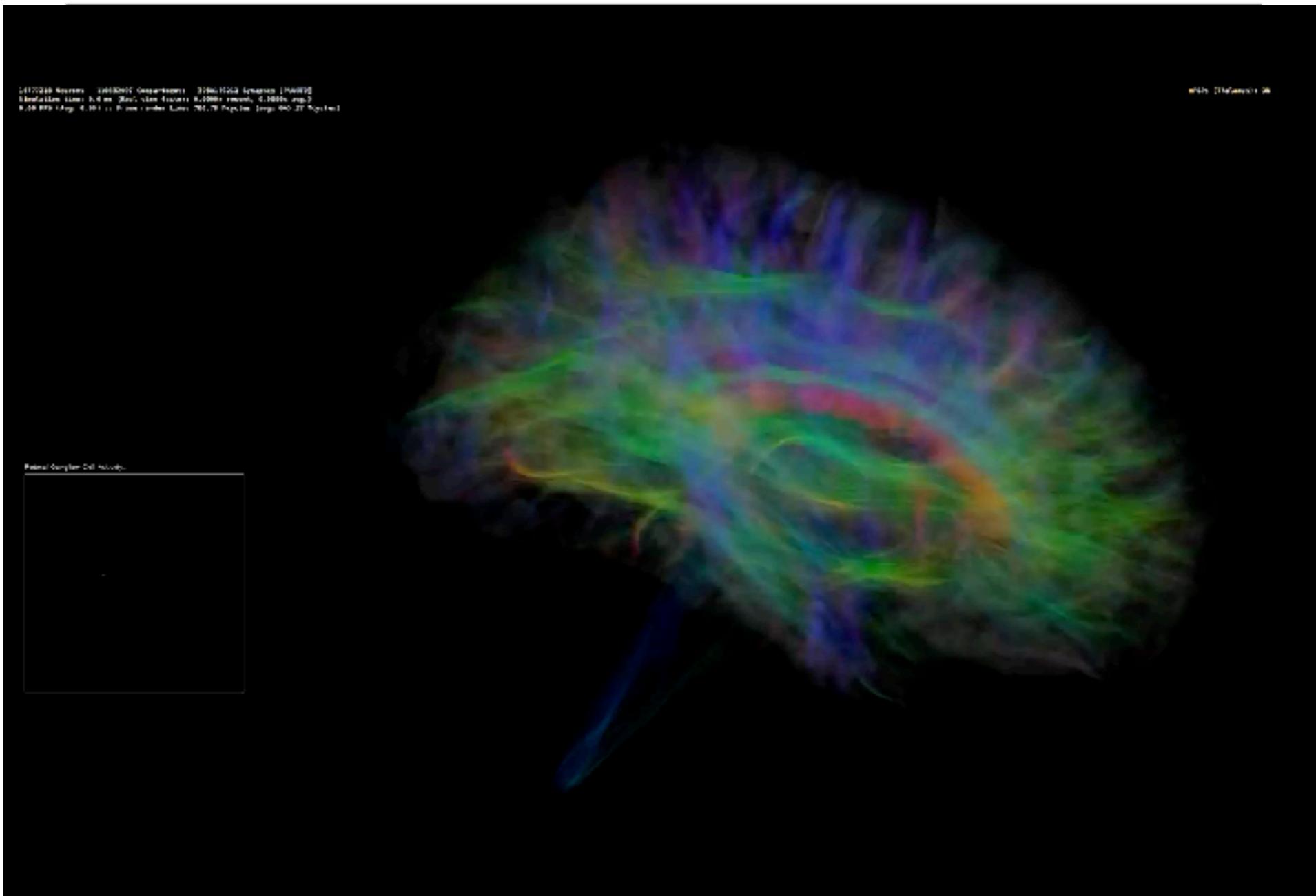
- Distinguish short GRBs flares from SGR flares from INTEGRAL
- Search for GRB afterglows in SDSS
- GRBs photometric redshift GROND
- Clustering of spectra (minimal spanning tree) in the distance-from-templates space in DFBS
- Search for M-stars, Carbon stars in DFBS
- JWST high-z: galaxy morphology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

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



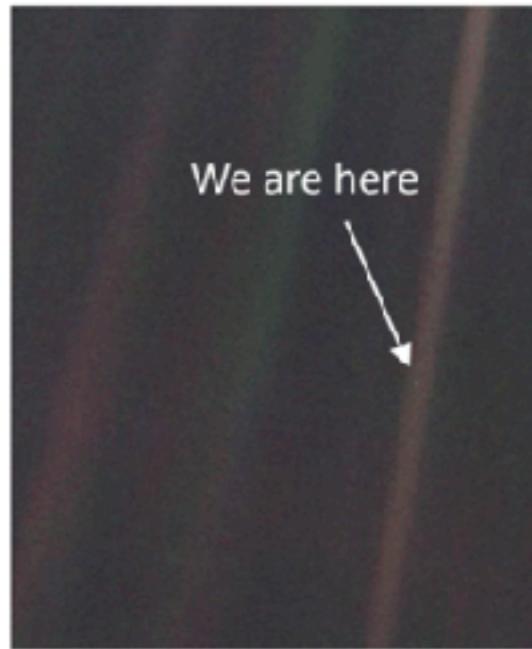
Frankenstein (1818)



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

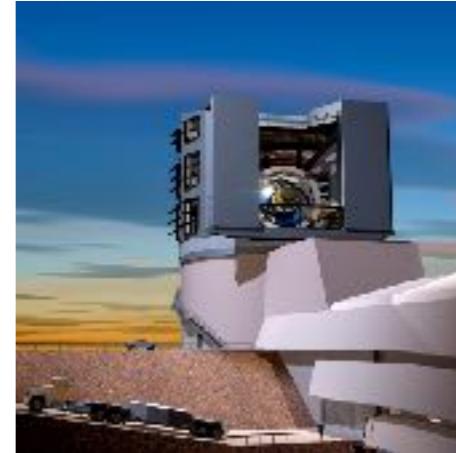
* Dates are for perspective and not as definitive historical record of invention or credit

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”

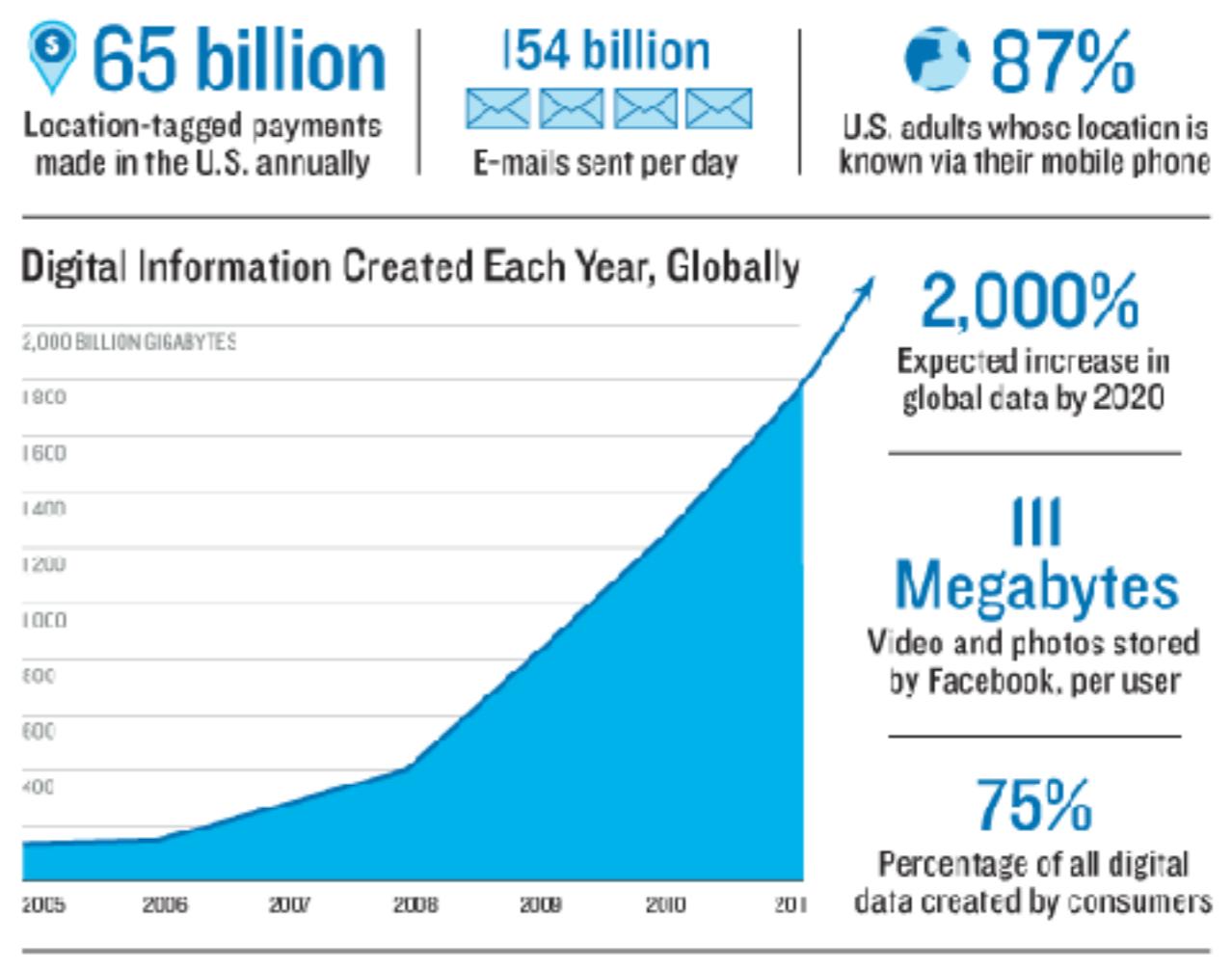
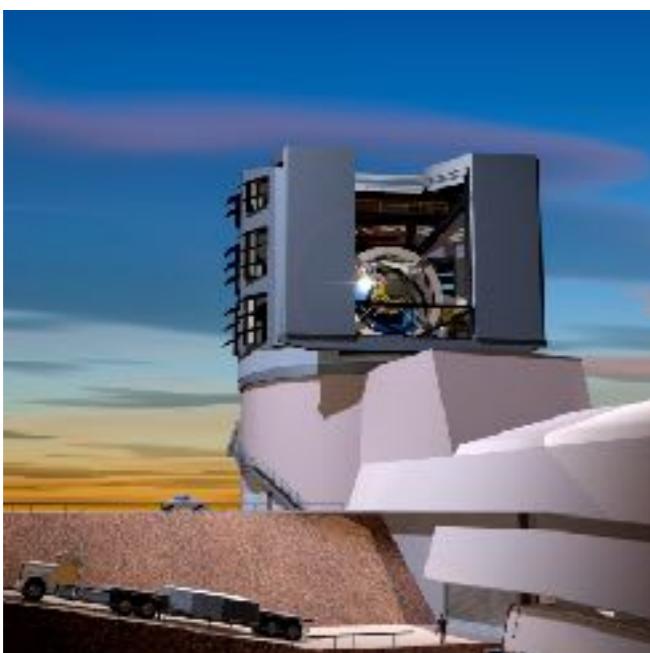
3Vs':

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

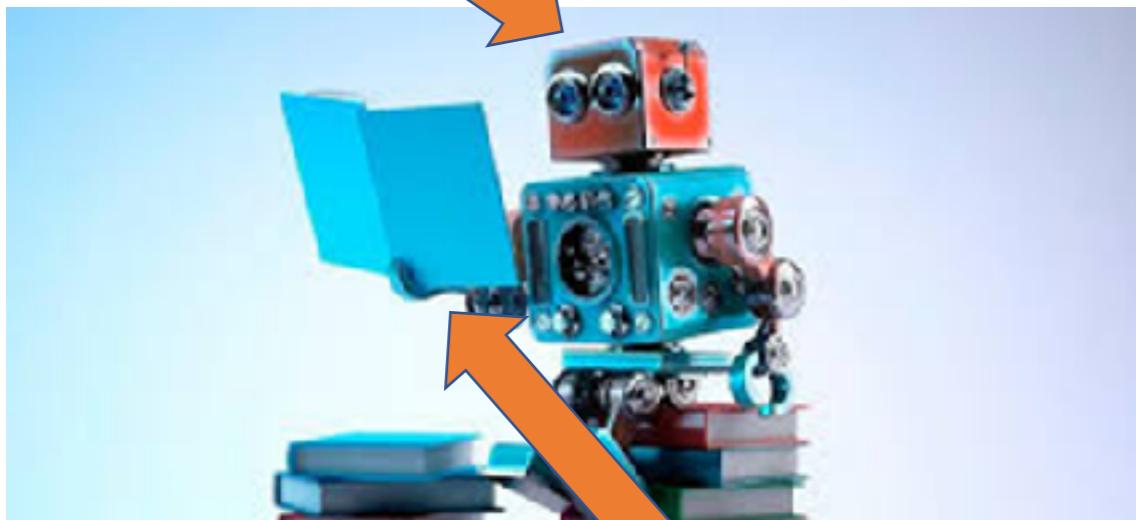
LSST

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?!?



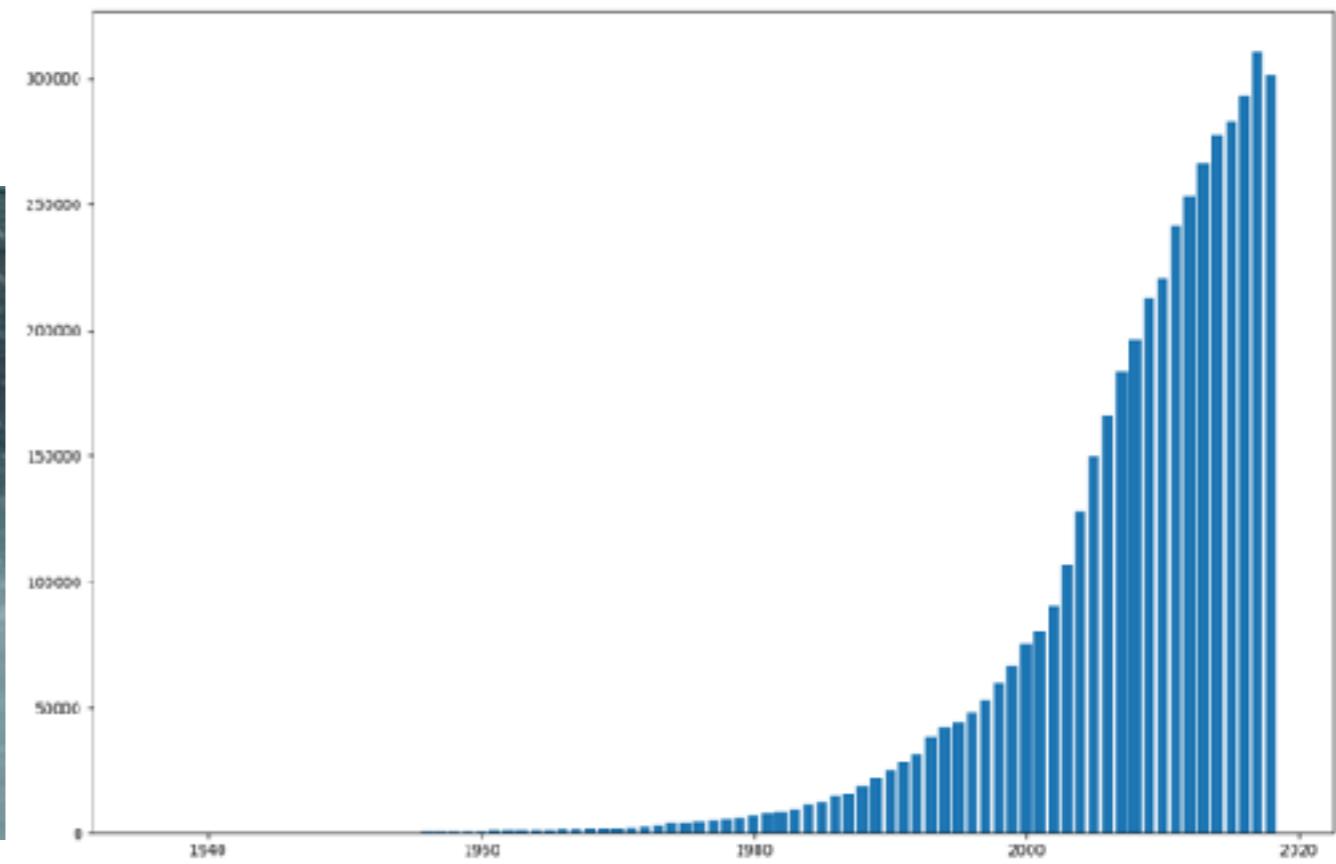
'AI IS THE NEW ELECTRICITY'



"Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don't think AI will transform in the next several years."

Andrew Ng

Former chief scientist at Baidu, Co-founder at Coursera



Number of ML papers on [arXiv.org](https://arxiv.org)

Garik Kasparov vs IBM's Deep Blue
(1997)



Lee Se-dol vs DeepMind Alpha Go
(2016)





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

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

Machine learning gives computers 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.

What is Machine Learning?

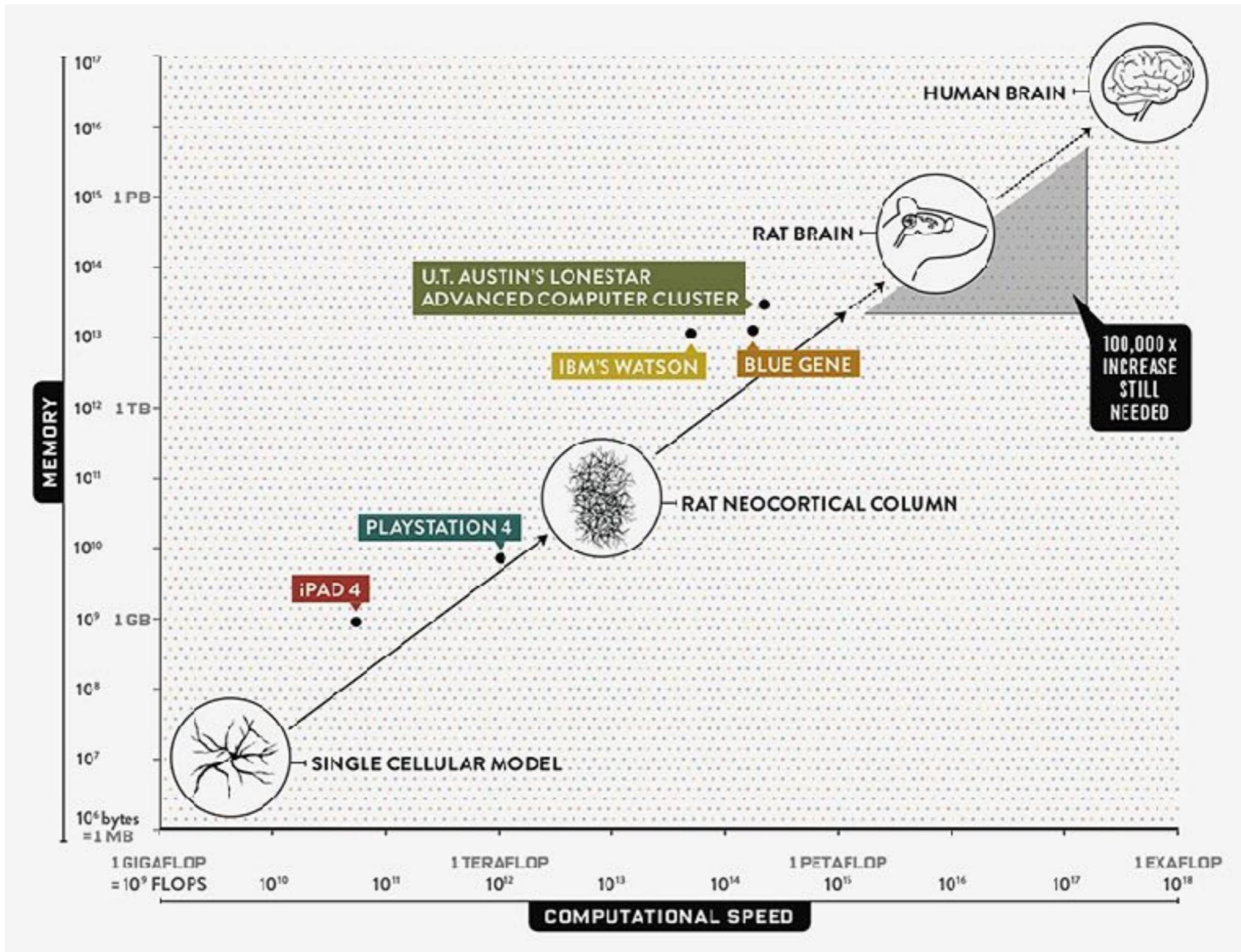
Machine learning is the process to automatically extract knowledge from data, usually with the goal of making predictions on new, unseen data.

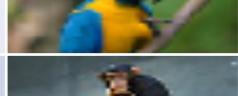
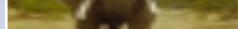
Central to machine learning is the concept of making decision automatically from data, without the user specifying explicit rules how this decision should be made.

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.

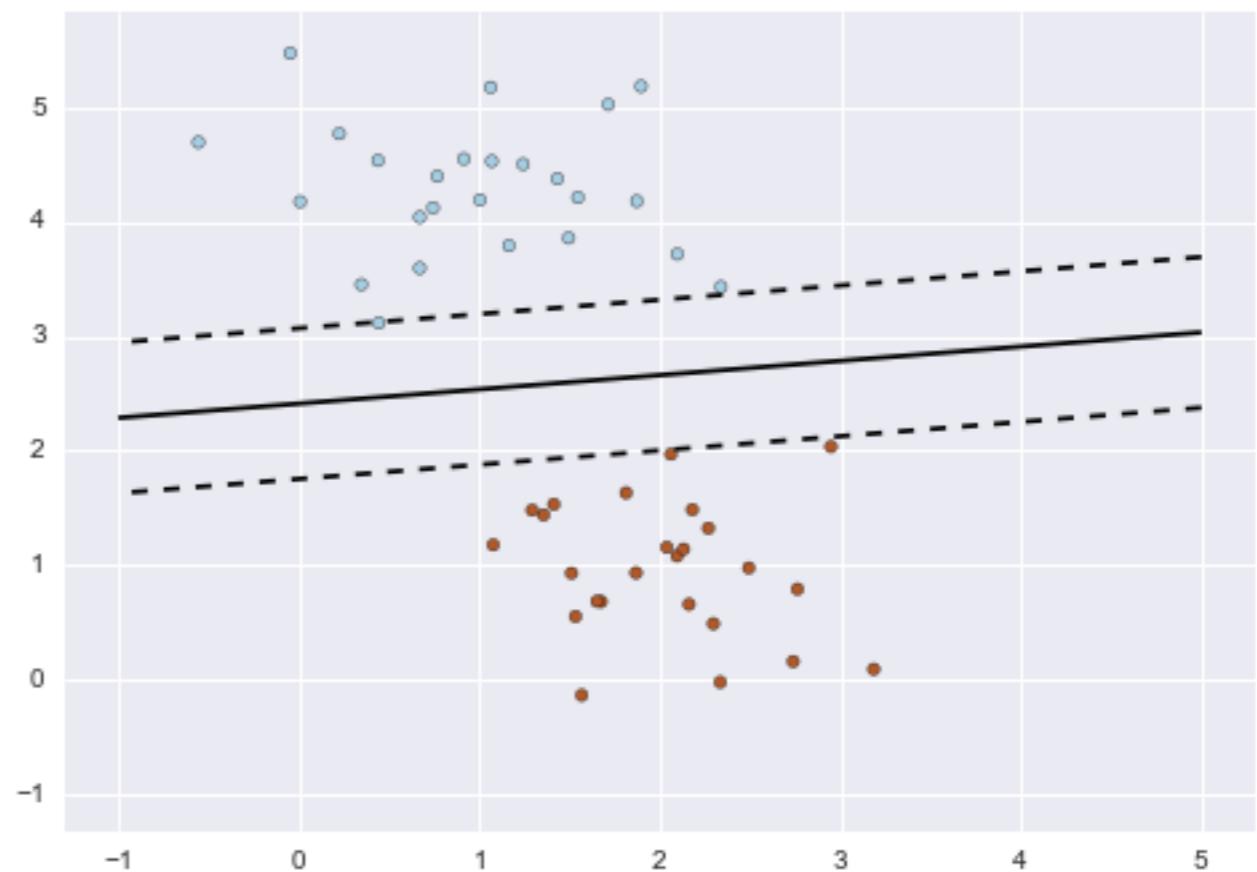
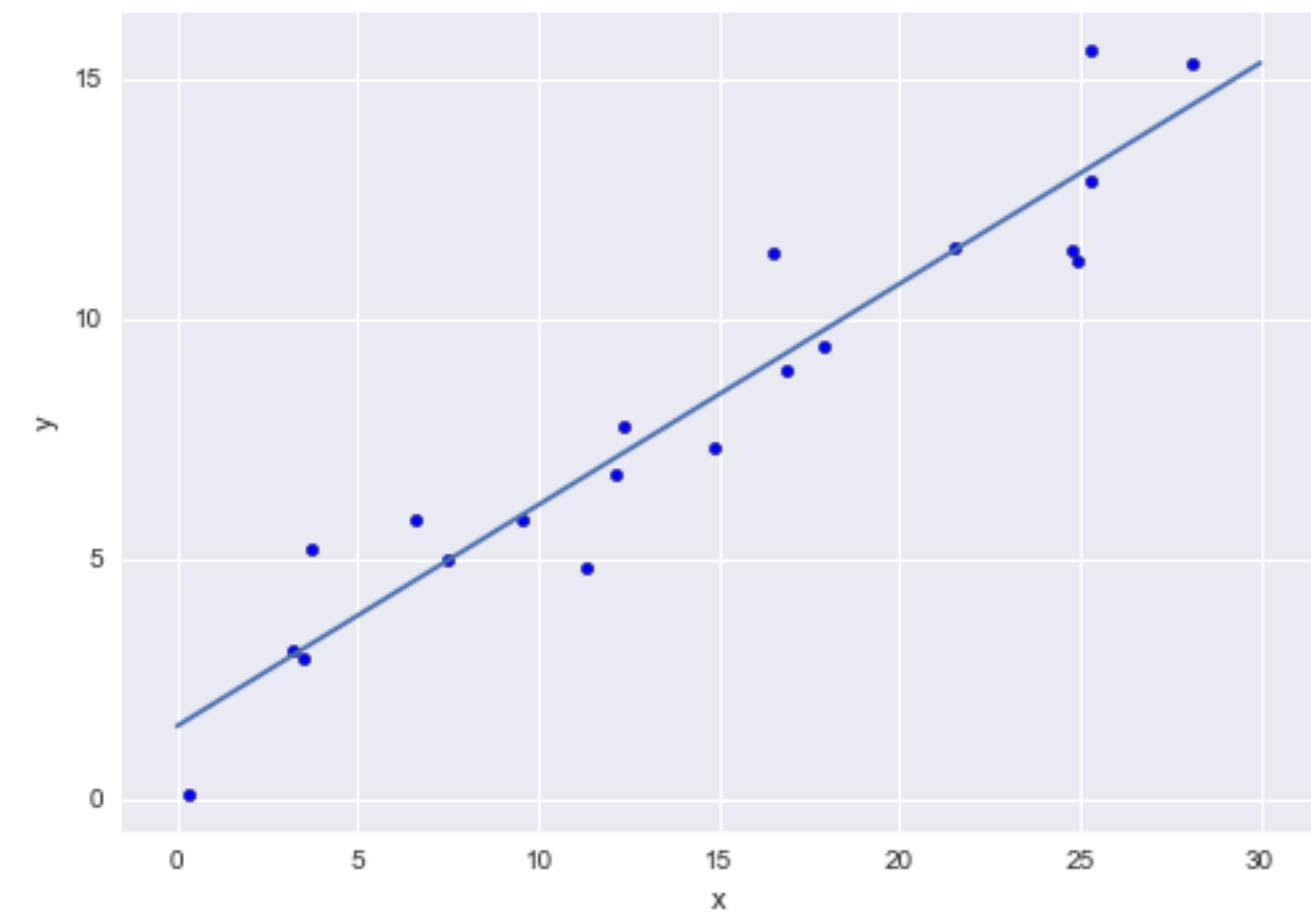
ML is the set of algorithms with tuneable parameters that can learn and adjust the values of these parameters from previously seen data and generalising for predictions of new yet unseen data.

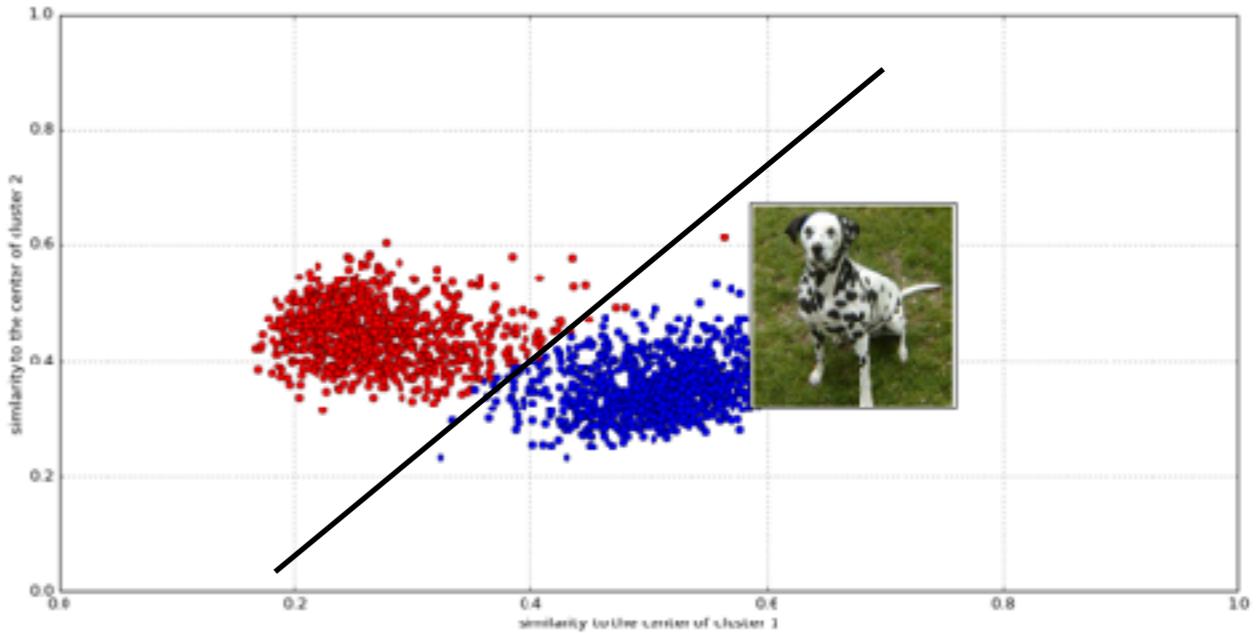
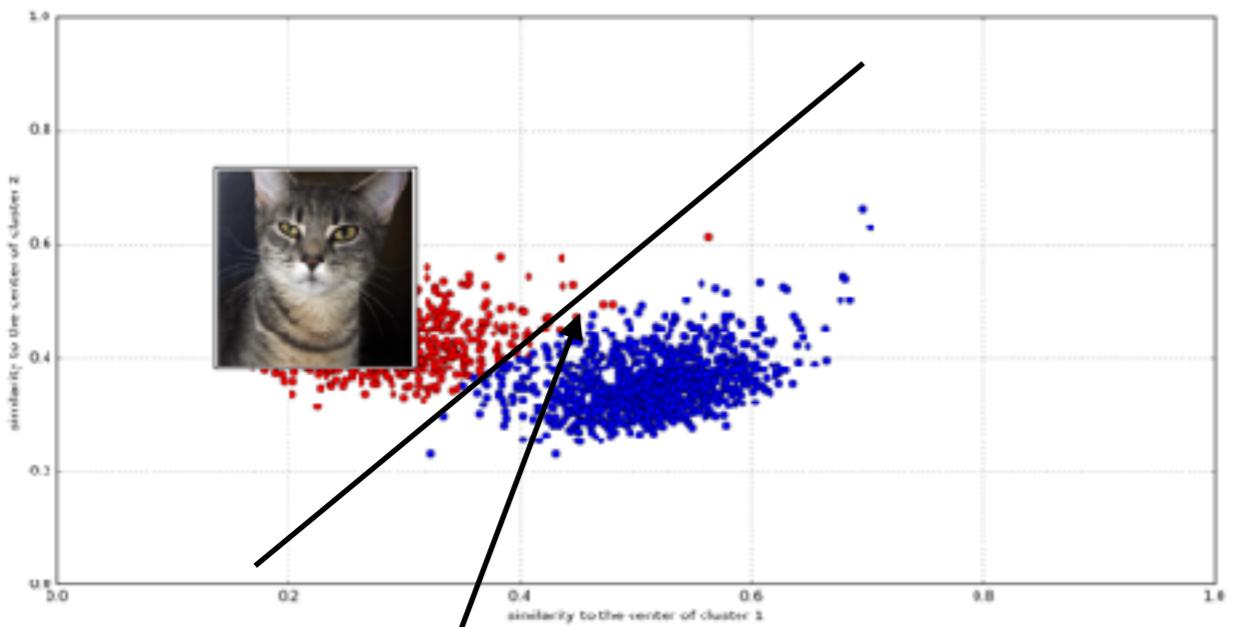
ML has many names: Pattern Recognition, Statistical Data Modelling, Learning from Data, Deep Learning, Computational Statistics



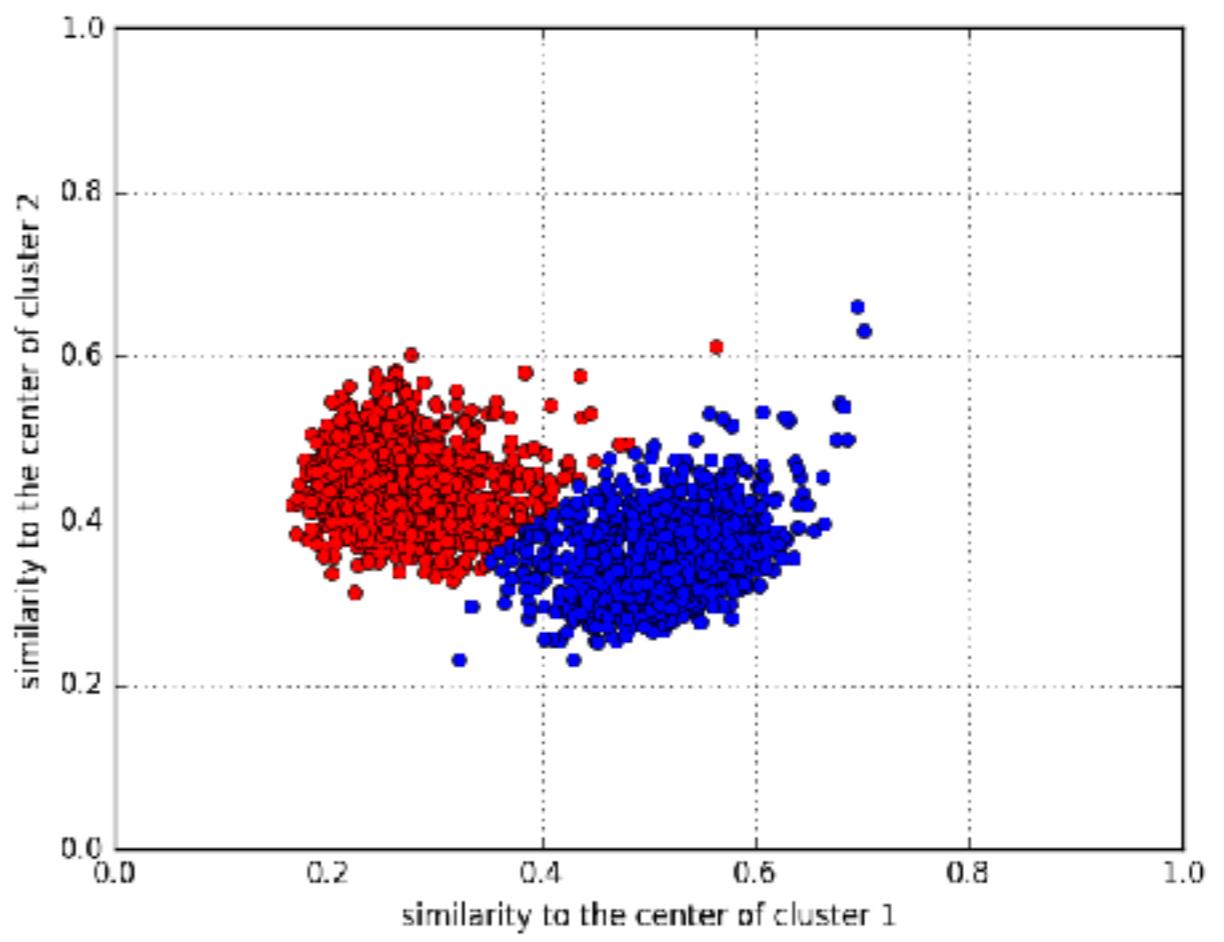
Name	# of neurons / # of synapses	Visuals
<i>Caenorhabditis elegans</i>	302	
<i>Hydra vulgaris</i>	5,600	
<i>Homarus americanus</i>	100,000	
<i>Blatta Orientalis</i>	1,000,000	
Nile Crocodile	80,500,000	
Digital Reasoning NN (2015)	~86,000,000 (est.) / 1.6E11	
<i>Rattus Rattatouillensis</i>	200,000,000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





Trying two clusters

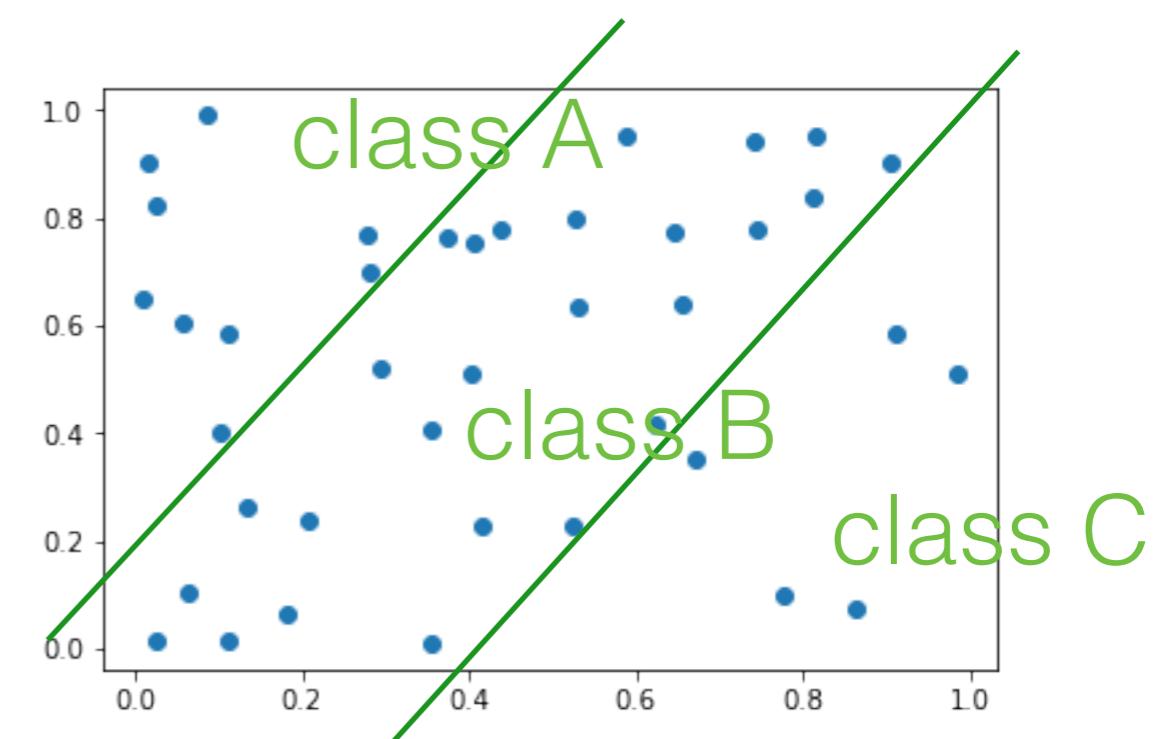
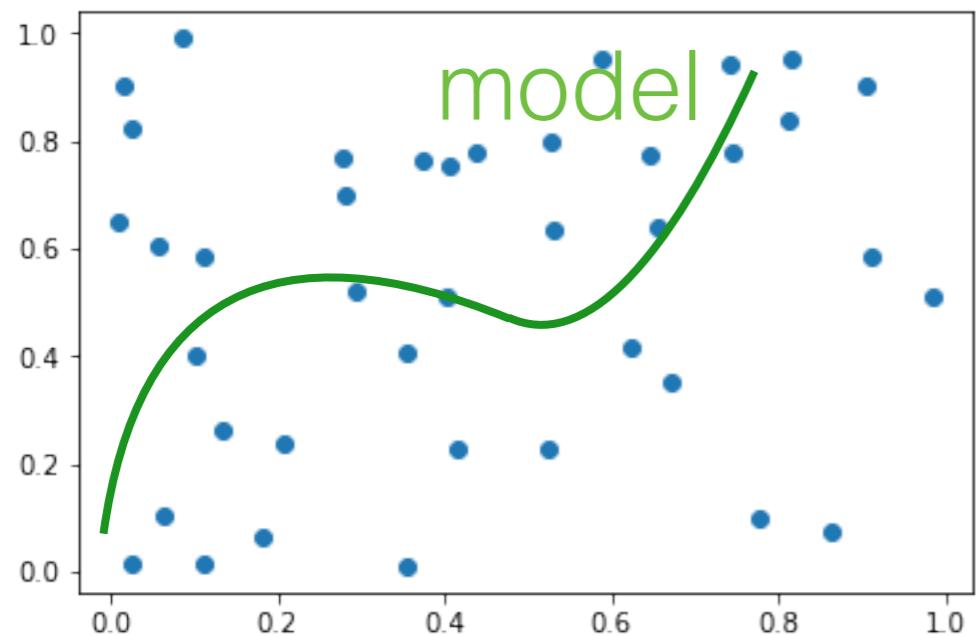


The difference between a physicist and an astronomer:

The physicist sees random 2D data and draws a curved line in it saying it's the model that describes the data.

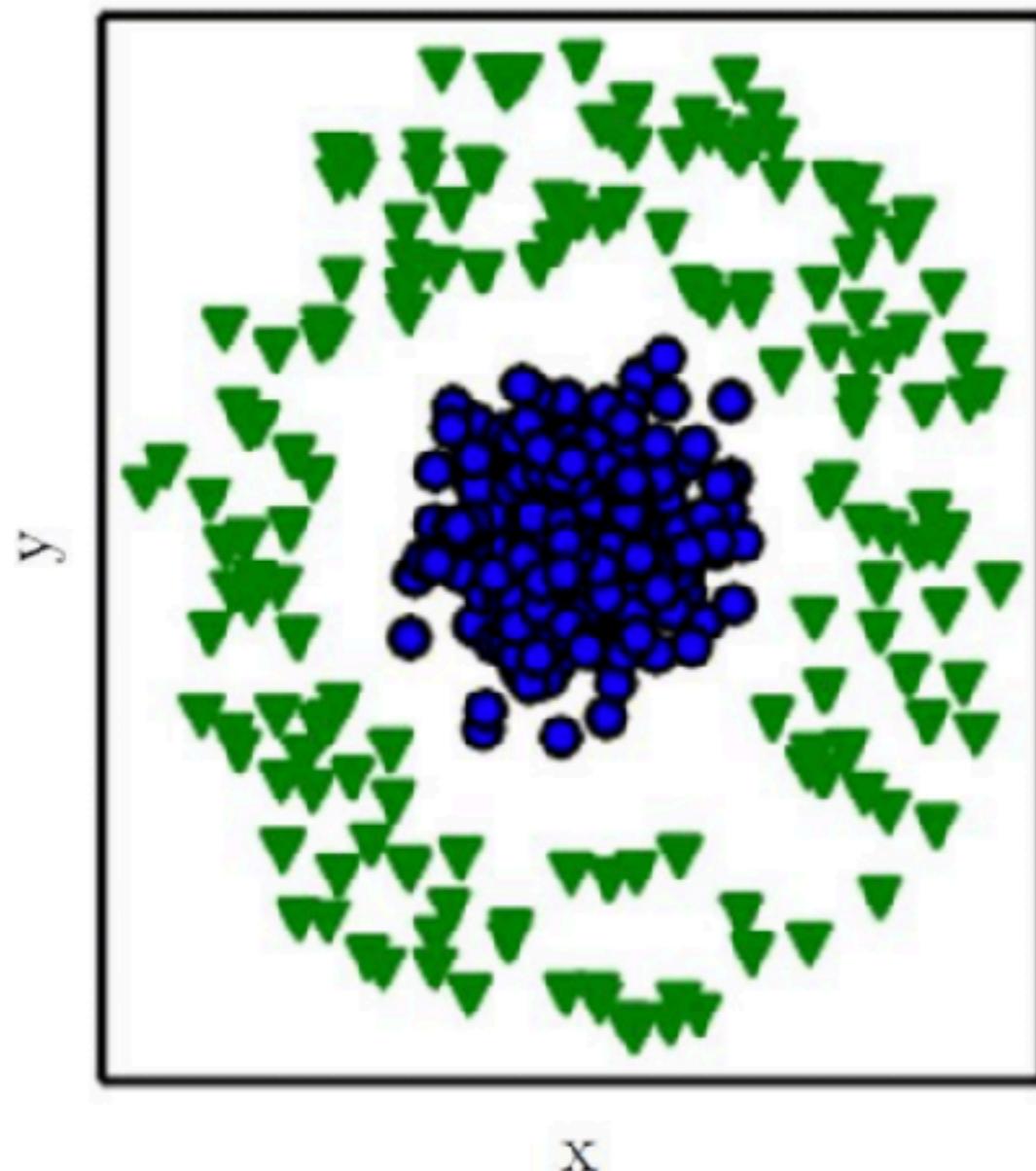
The astronomer draws two parallel lines, saying these points belong to class A, these to class B and this is class C.

— Andy Lawrence, private communication

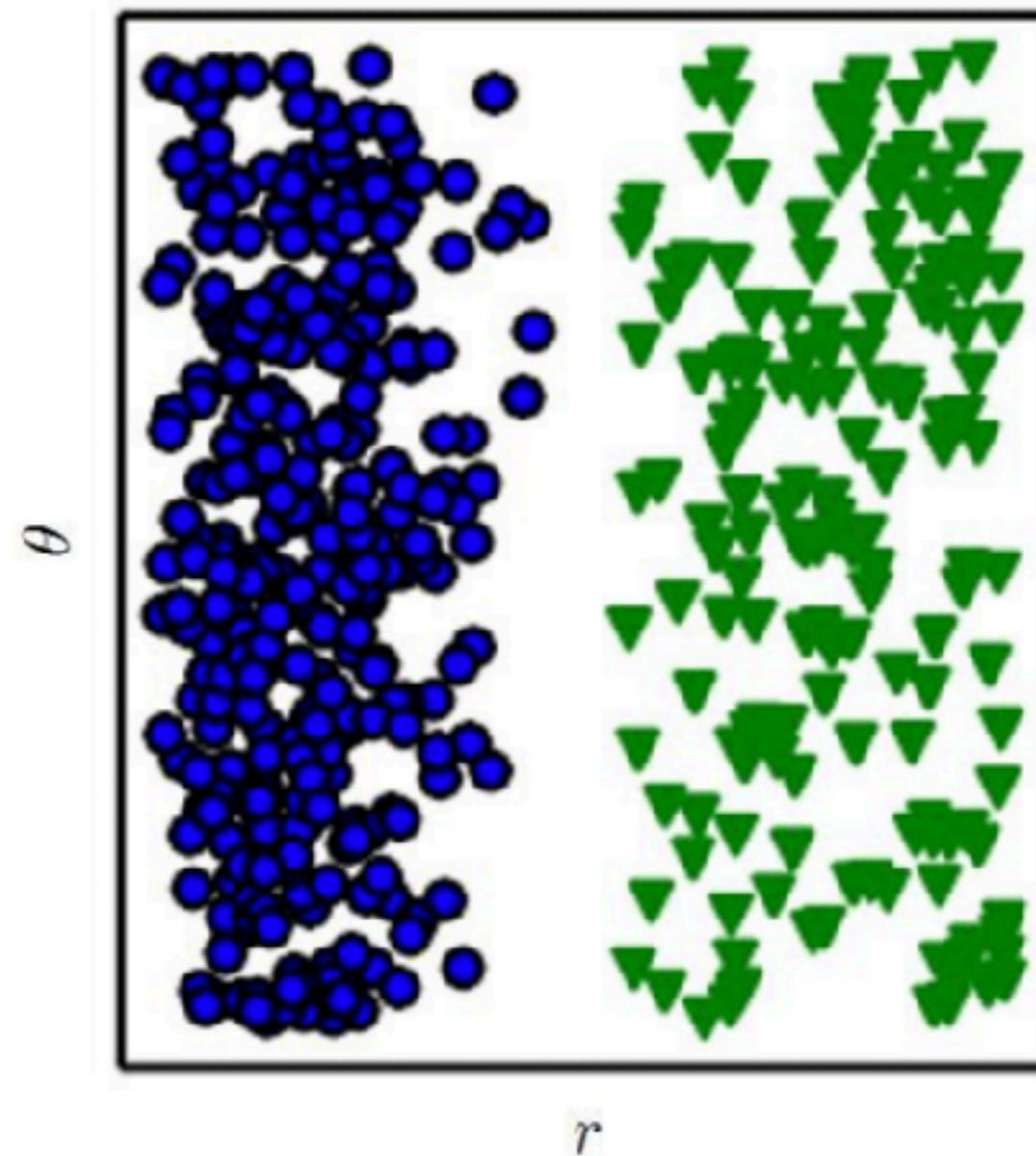


Representation Matters

Cartesian coordinates



Polar coordinates



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

Regression vs Classification

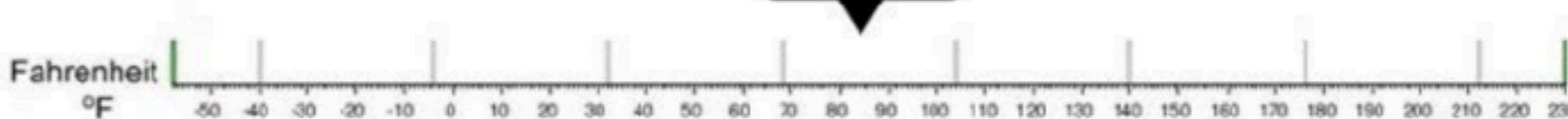


Regression

What is the temperature going to be tomorrow?

PREDICTION

84°



Classification

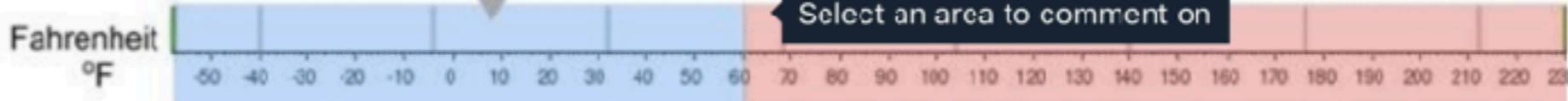
Will it be Cold or Hot tomorrow?

PREDICTION

HOT

COLD

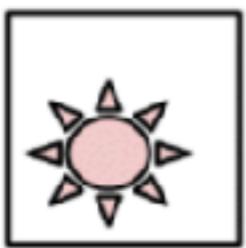
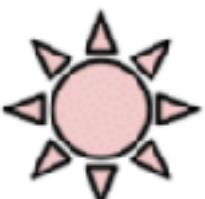
Select an area to comment on



Multi-Class vs Multi-Label

Multi-Class

Samples



Labels (t)

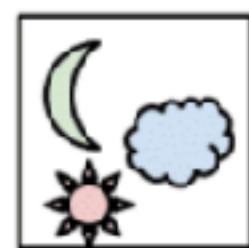
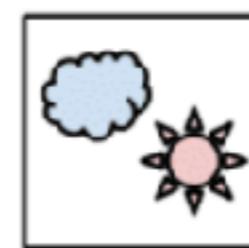
[0 0 1]

[1 0 0]

[0 1 0]

Multi-Label

Samples



Labels (t)

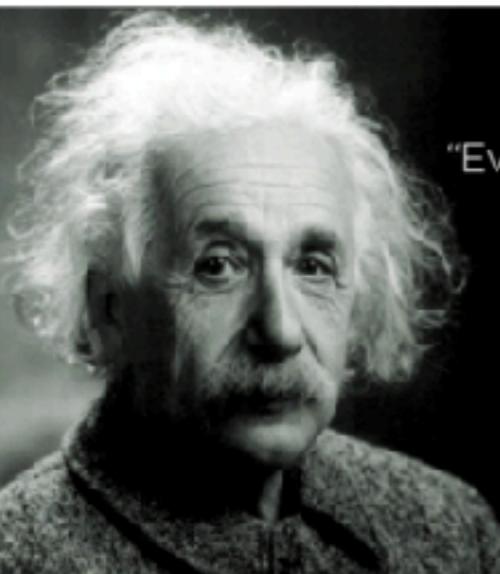
[1 0 1]

[0 1 0]

[1 1 1]

$C = 3$

First Steps: Start Simple



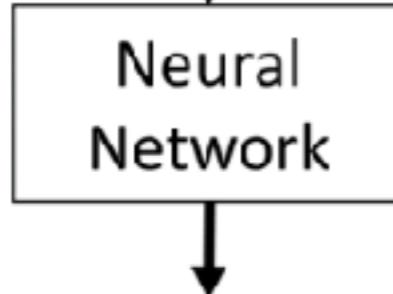
"Everything should be made
as simple as possible.
But not simpler."

Albert Einstein

Input Image:



TensorFlow
Model:



Output:

(with 87% confidence)

1

2

3

4

5

6

```
# 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()

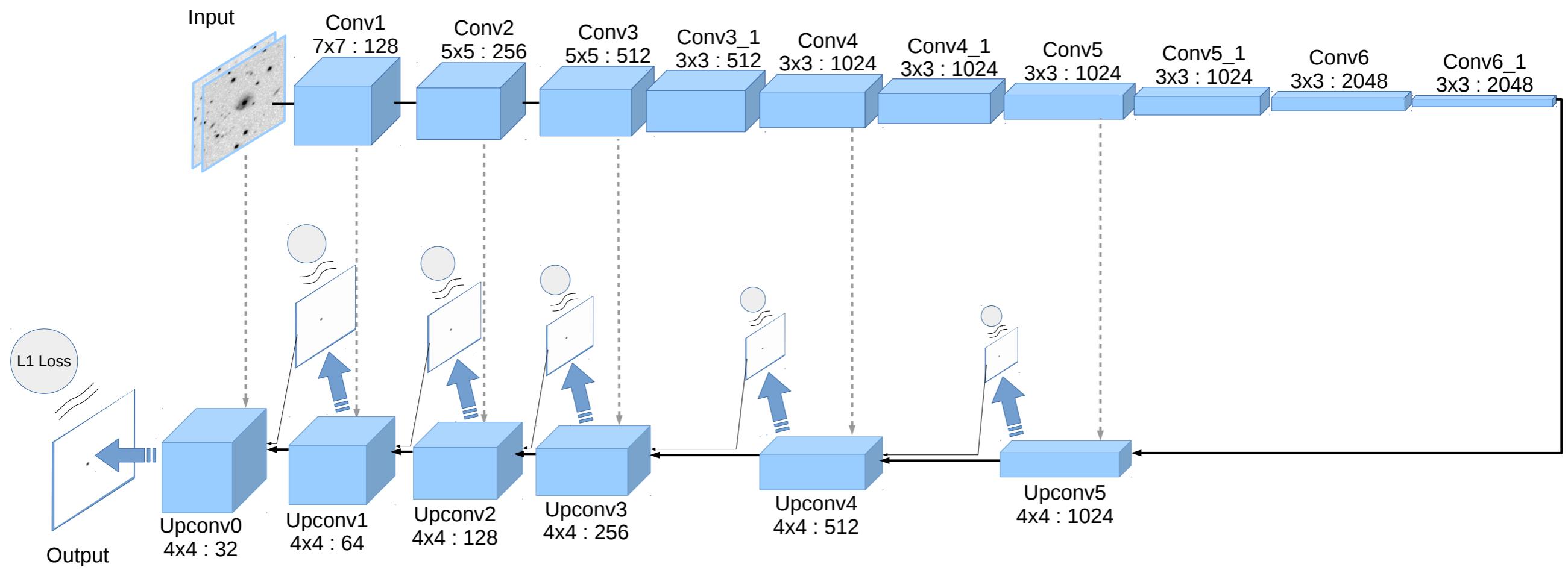
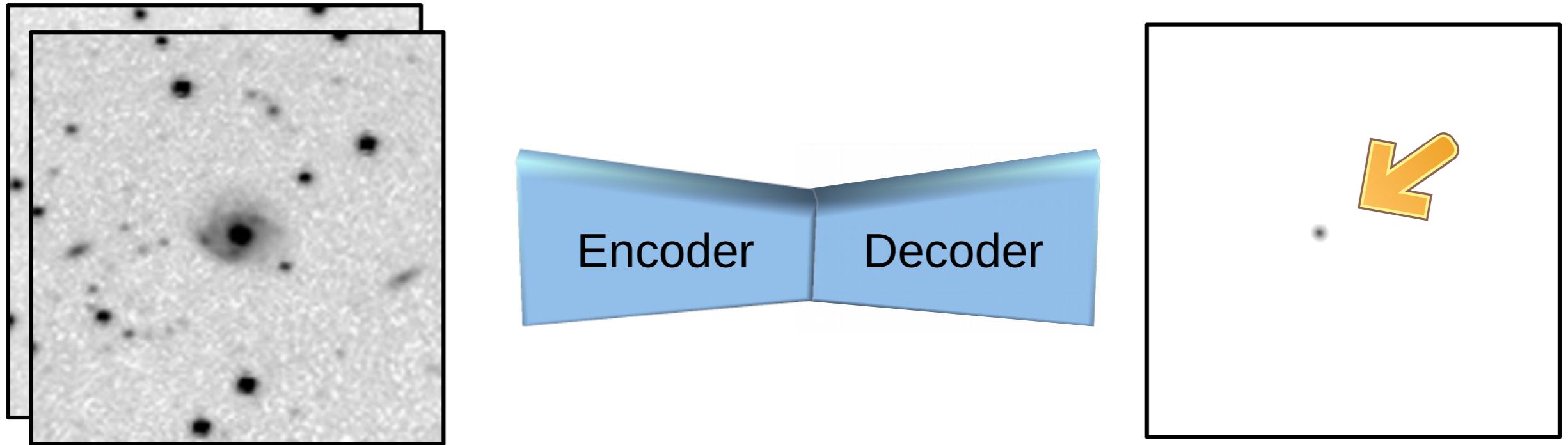
# setup model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

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

# train model
model.fit(train_images, train_labels, epochs=5) # Select an area to comment on

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

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



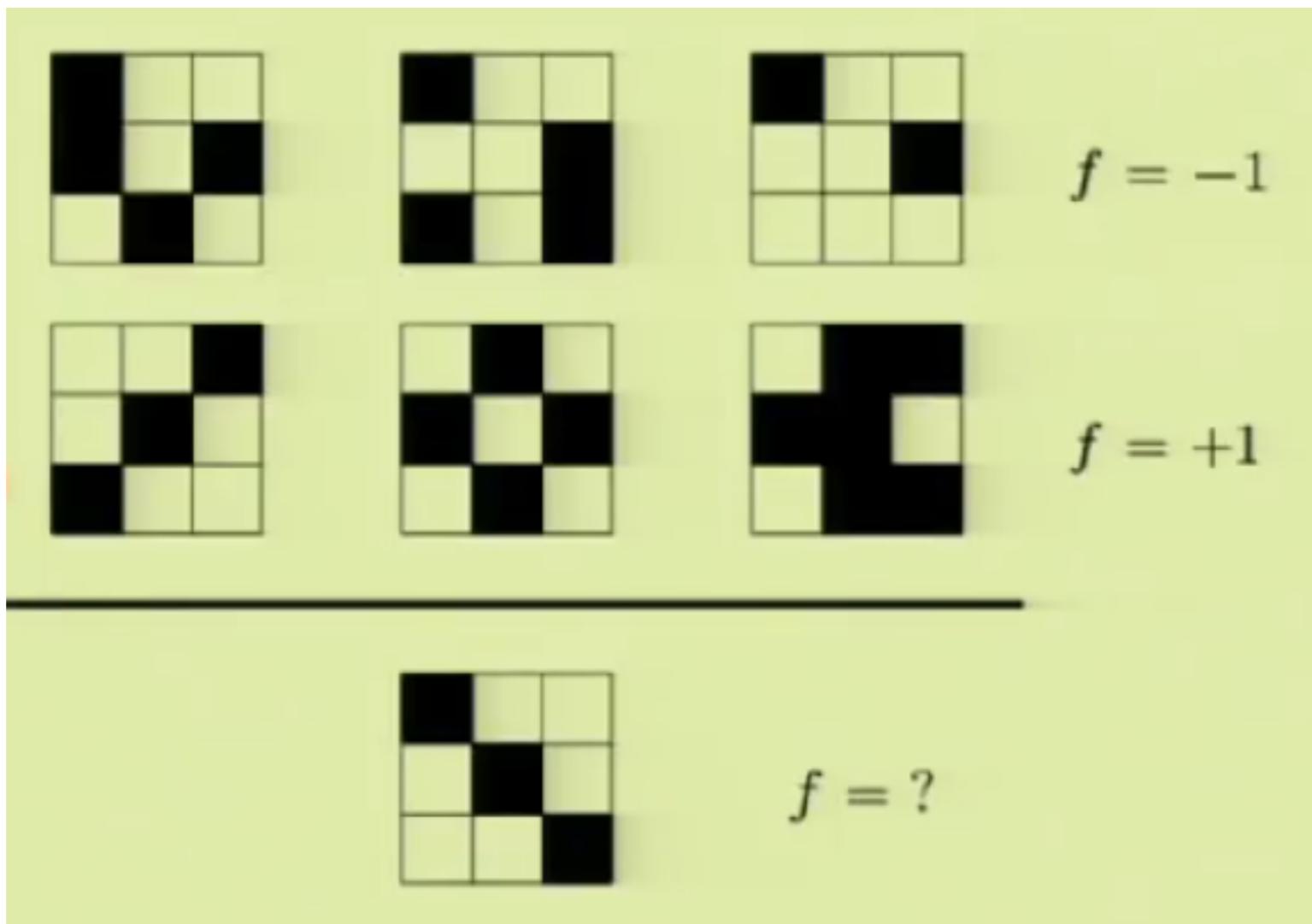
60 %	Data preparation (searching, query, downloading, formatting, cleaning, normalising)
10 %	ML coding
30 %	Fine tuning and interpretation

The art of **asking the right question**

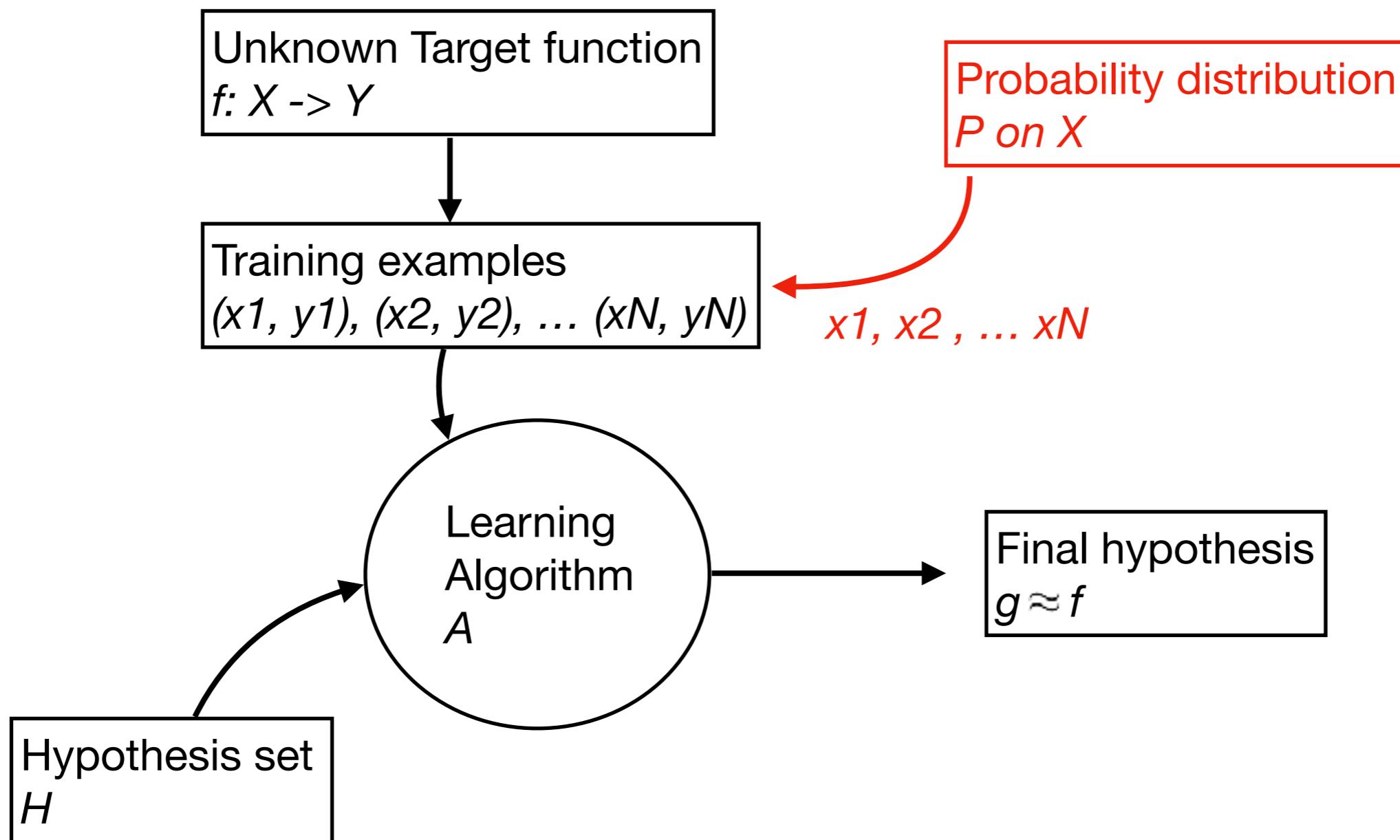
Essentials of Learning

- 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...

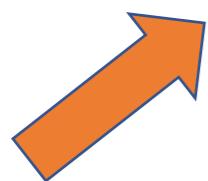
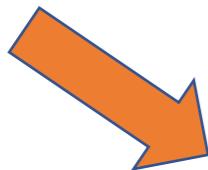
Python is slow

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 $y_t = \beta^T x_t + \rho_t + \epsilon_t$ 
Spark	Distributed Computing	
Scikit-learn	Machine Learning Toolbox	
Keras	Deep Learning	
TensorFlow	Deep Learning	
Open-cv	Computer Vision	



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>