An introduction to neural networks: from "classical" to deep learning

Marc Huertas-Company

EDE - Session 2020

SOME PRELIMINARY NOTES

I AM NOT A MACHINE LEARNING RESEARCHER

BUT I HAVE BEEN USING MACHINE LEARNING FOR THE LAST ~15 YEARS FOR MY RESEARCH IN ASTROPHYSICS

THESE LECTURES ARE INTENDED TO PROVIDE A **GENERAL INTRODUCTION TO** NEURAL NETWORKS

ML@EDE20

CYCLE 1: INTRODUCTION AU DEEP LEARNING:
tutorial session at 2pm today (Monday)

CYCLE 2: 3h LECTURES ON NEURAL NETWORKS AND DEEP LEARNING 9am (Monday) + Tutorials 2pm (Tuesday)

ALL: Séance Ouverte 5.30 pm

ML@EDE20

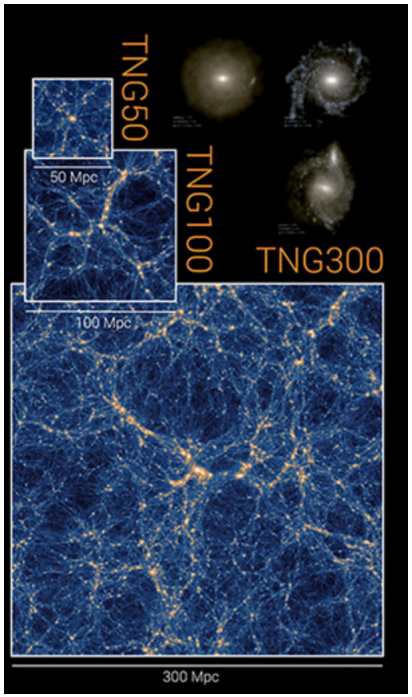
REQUIREMENTS FOR THE TUTORIALS:

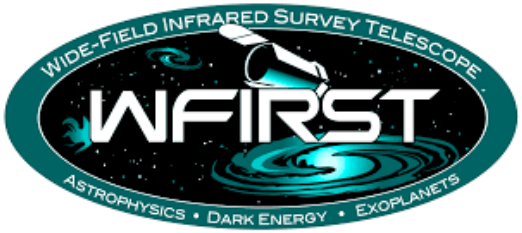
- GOOGLE DRIVE ACCOUNT WITH SOME FREE SPACE

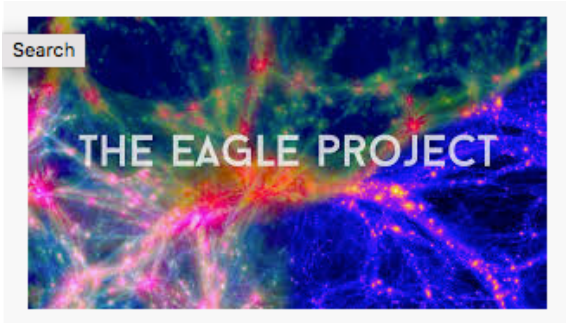
- INTERNET CONNECTION ?!

GITHUB REPO WITH MORE COMPLETE SLIDES AND COLAB TUTORIALS:
https://github.com/mhuertascompany/EDE20

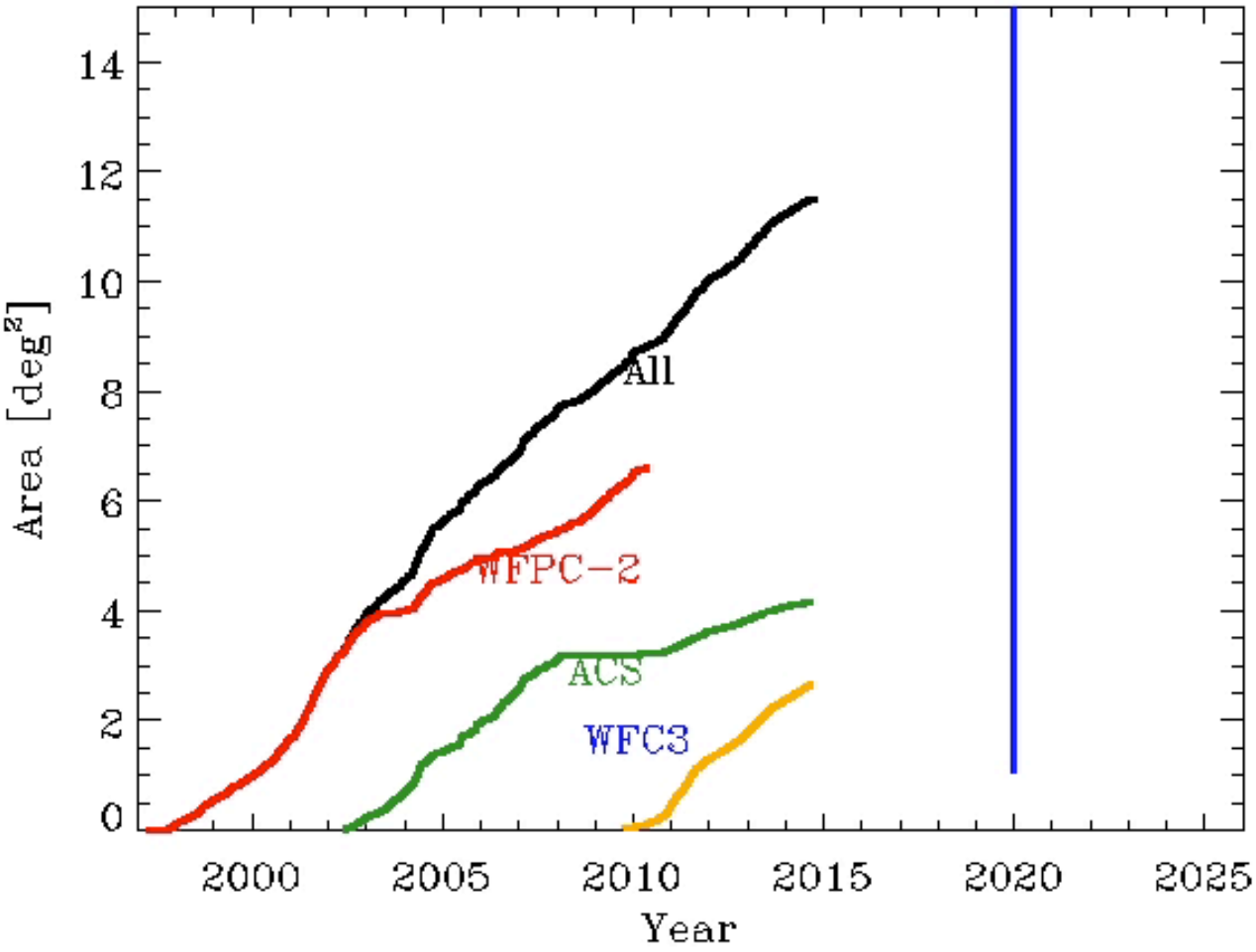
WE DO NOT HAVE THE CAPACITY TO "LOOK" AT FUTURE ASTRONOMICAL (BIG-DATA) SETS

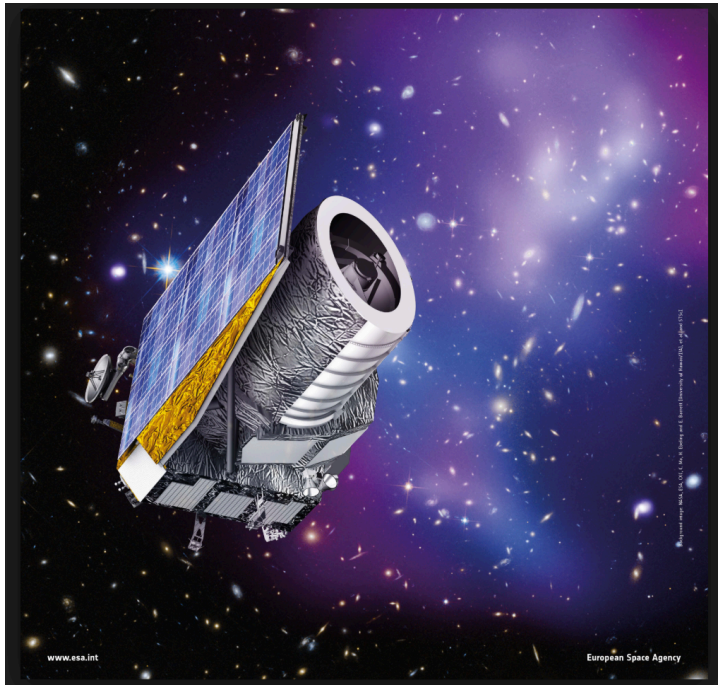






WE WILL SOON ENTER A NEW REGIME





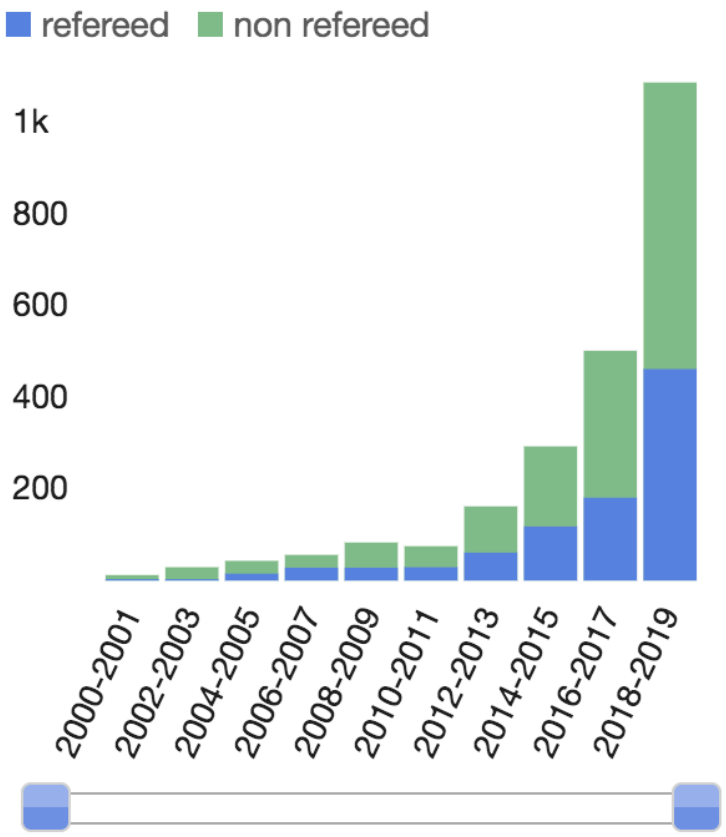
EUCLID space telescope (2021)

esa

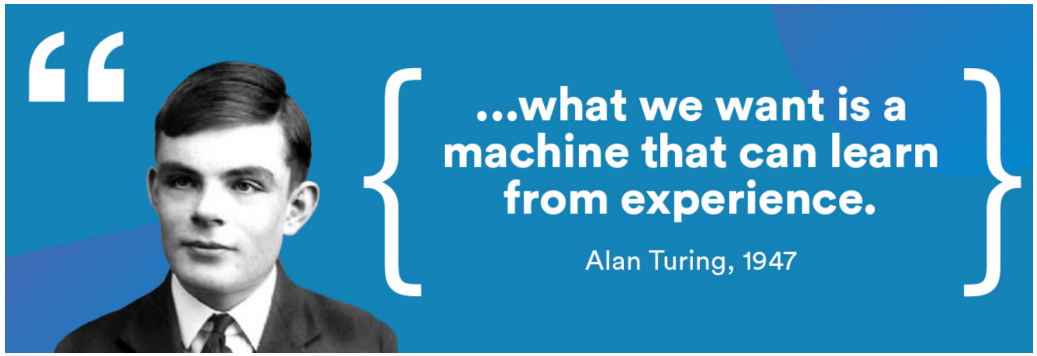
NASA

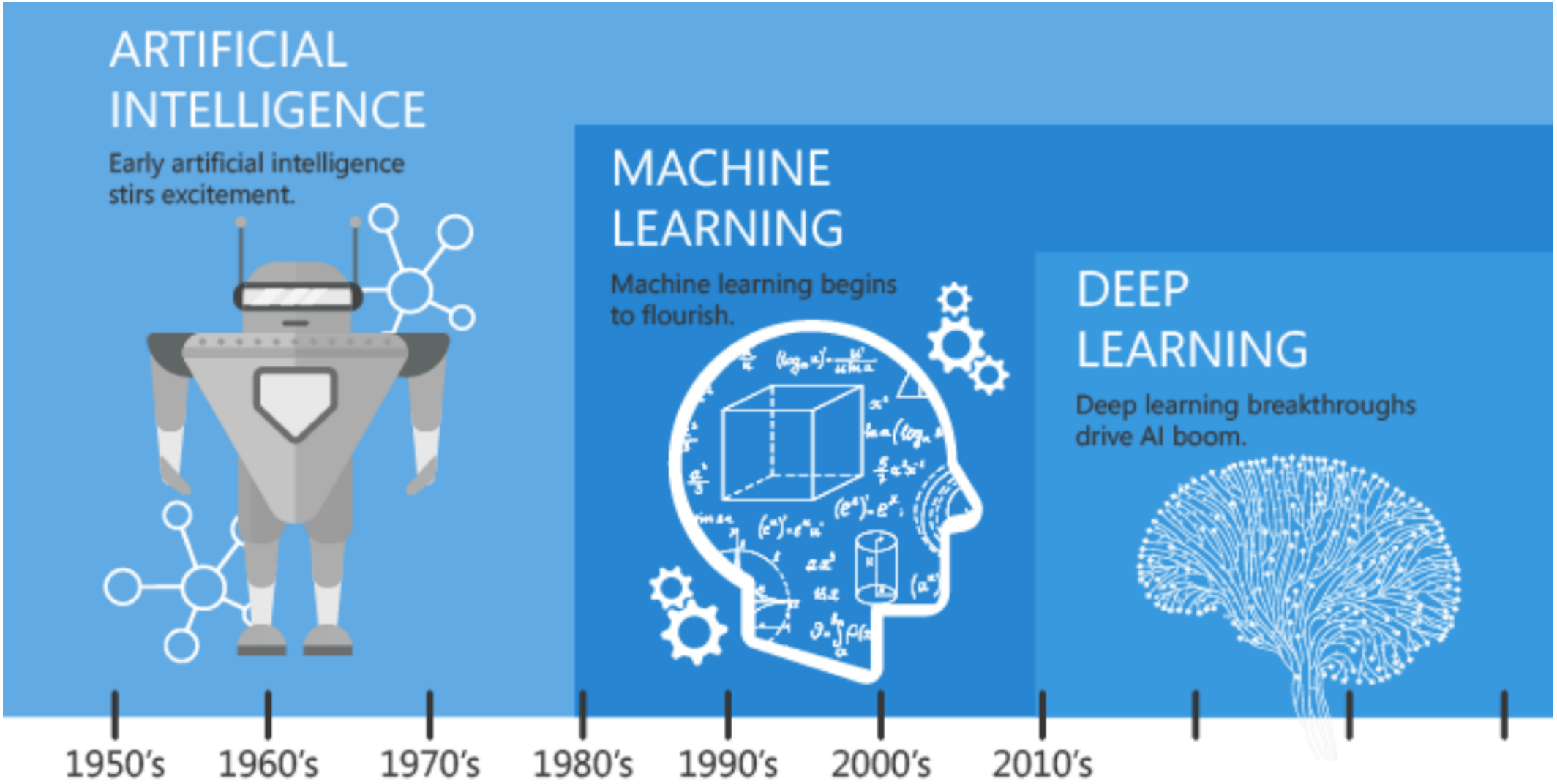
(Thanks to J. Brinchmann)

AI TO THE RESCUE?

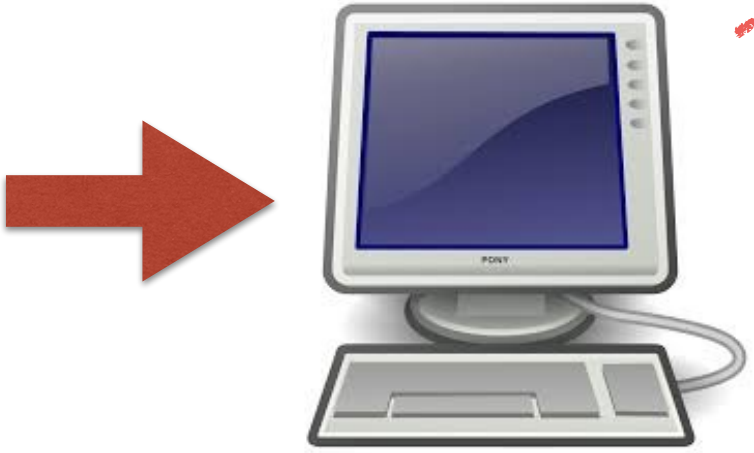


SOURCE: ADS


...what we want is a machine that can learn from experience.
Alan Turing, 1947


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

BEFORE 2012....

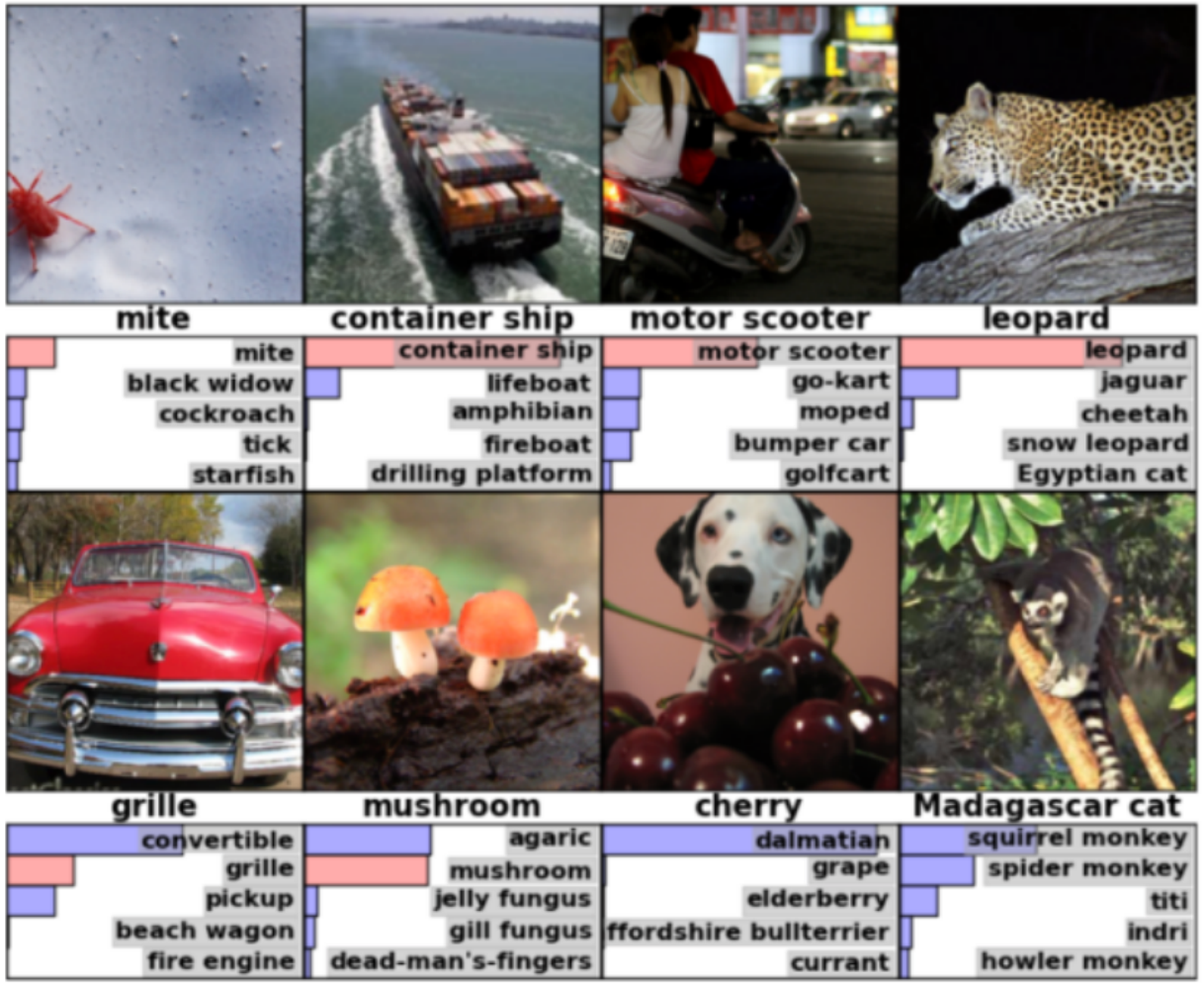


CAT?

DOG?

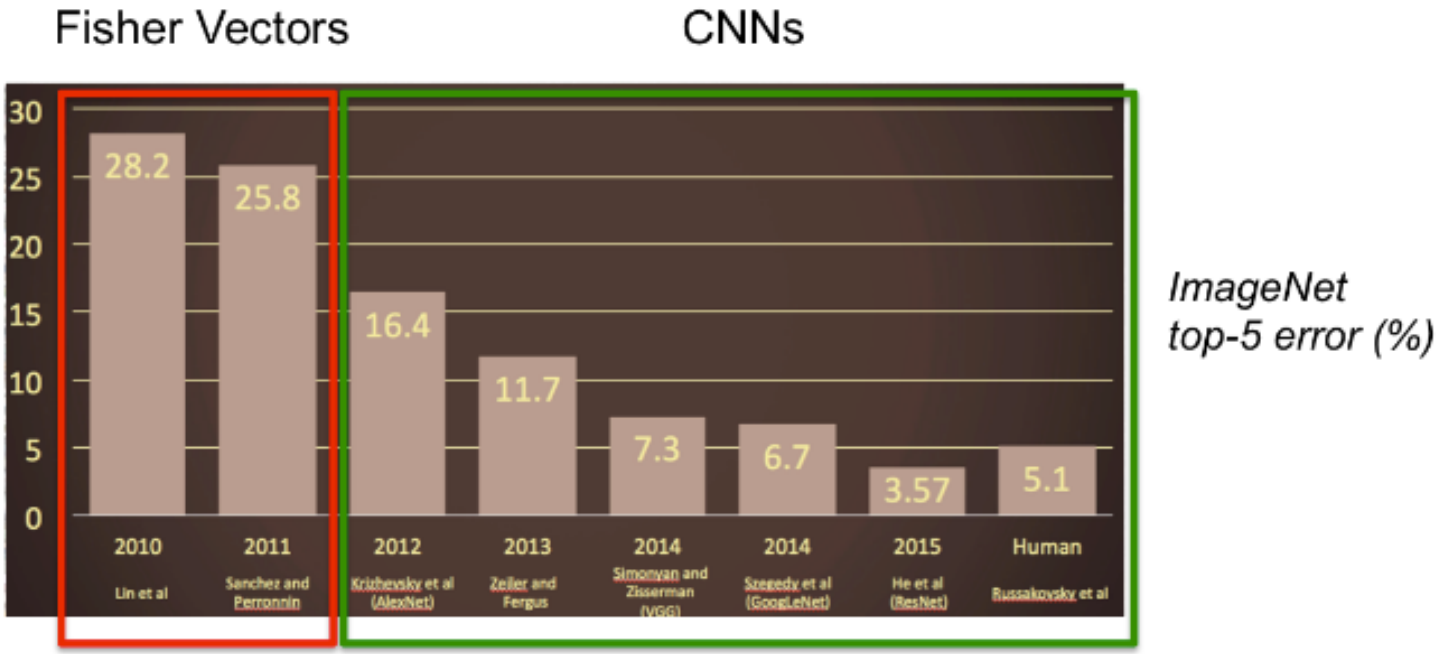
TRIVIAL HUMAN TASKS REMAINED CHALLENGING FOR COMPUTERS

AFTER 2012



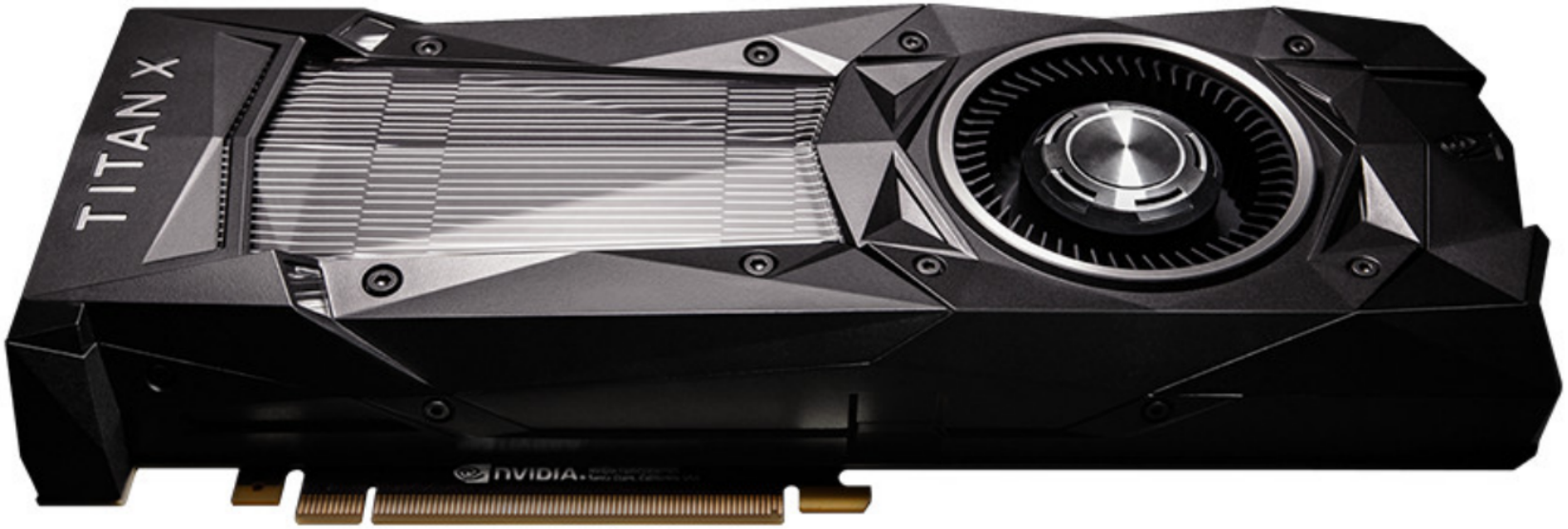
IT HAS BECOME TRIVIAL….

THIS IS A CHANGE OF PARADIGM!



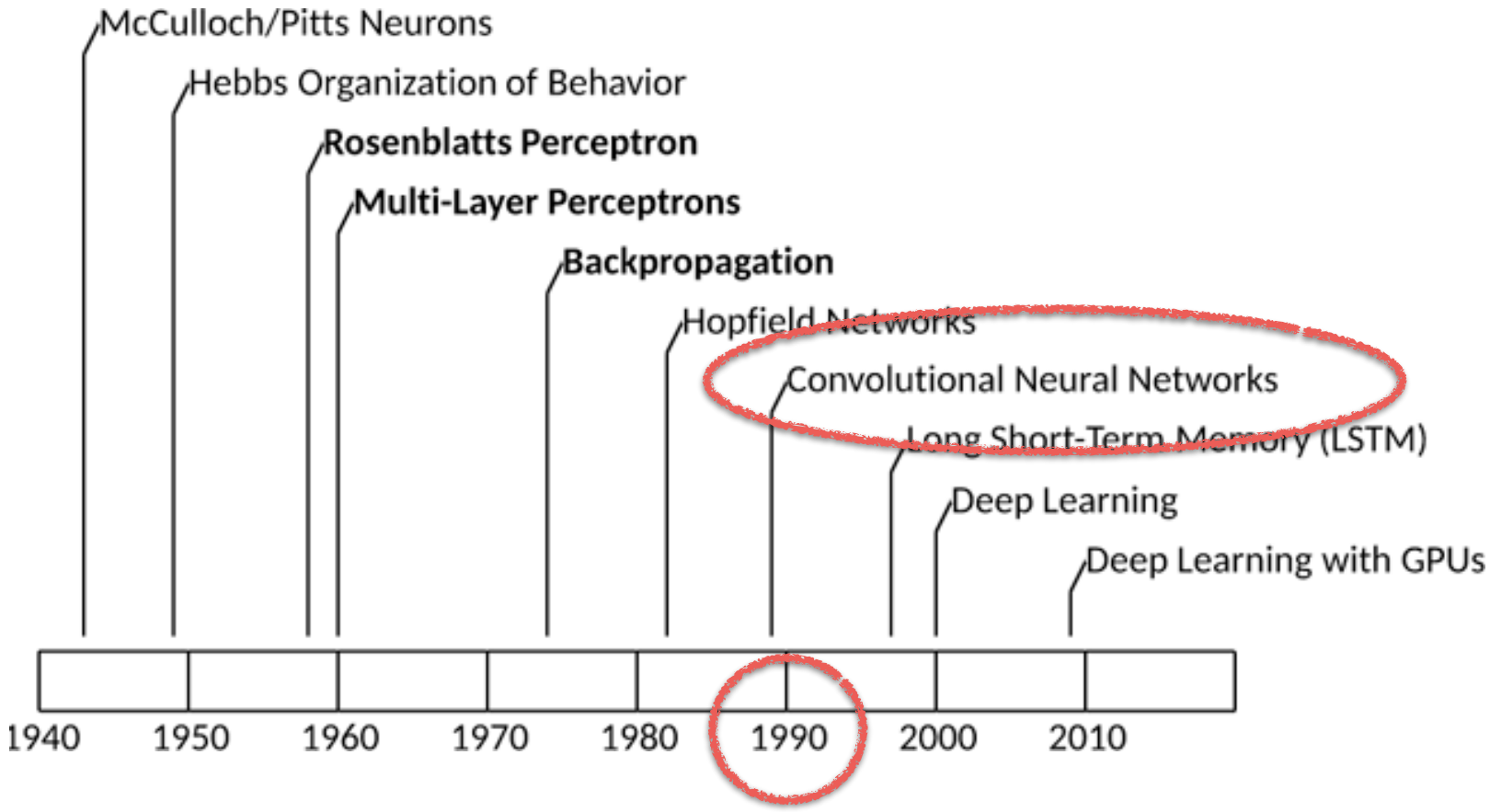
ONE OF THE MAIN REASONS OF THIS BREAKTHROUGH IS THE AVAILABILITY OF VERY LARGE DATASETS TO LEARN

COMBINED WITH THE TECHNOLOGY TO PROCESS ALL THIS DATA



ONE OF THE MAIN REASONS OF THIS BREAKTHROUGH IS THE AVAILABILITY OF VERY LARGE DATASETS TO LEARN

BUILDING BLOCKS OF DEEP LEARNING ARE FROM THE 90s



PROGRAM FOR TODAY

- **PART I: INTRODUCTION**
  - UNSUPERVISED / SUPERVISED
  - GENERAL STEPS TO “TEACH A MACHINE”
  - LOSS FUNCTION, OPTIMIZATION
- **PART II: THE FOUNDATIONS OF NEURAL NETWORKS**
  - PERECPTRON, NEURON DEFINITION
  - LAYER OF NEURONS, HIDDEN LAYERS
  - ACTIVATION FUNCTIONS
  - OPTIMIZATION [GRADIENT DESCENT, LEARNING RATES]
  - BACKPROPAGATION
  - NEURAL NETWORKS AS STATISTICAL MODELS: LOSS FUNCTIONS

PROGRAM FOR TODAY

- **PART III: THE FOUNDATIONS OF DEEP LEARNING**
  - CONVOLUTIONS AS NEURONS
  - CNNs [POOLING, DROPOUT]
  - VANISHING GRADIENT / BATCH NORMALIZATION
- **PART IV: BEYOND CLASSIFCATION - IMAGE2IMAGE NETWORKS**
  - FCNNs (IMAGE2IMAGE NETWORKS)
  - OBJECT DETECTION
  - INSTANCE AND SEMANTIC SEGMENTATION
  - (A VERY BRIEF INCURSION INTO UNSUPERVISED LEARNING: AUTOENCODERS)

REFERENCES

SEVERAL SLIDES / INFOS SHOWN HERE ARE INSPIRED/ TAKEN FROM OTHER WORKS / COURSES FOUND ONLINE

- Deep Learning: Do-It-Yourself! [Bursuc, Krzakala, Lelarge]
- DEEPLEARNING.AI [COURSERA, Ng, Bensouda, Katanforoosh]
- MACHINE LEARNING LECTURES [Keck]
- EPFL DEEP LEARNING COURSE [Fleuret]
- DEEPMIND / UCL Lecture series [DeepMind]

+ many others!

Thanks to all of them!

PART I: AN INTRODUCTION TO "CLASSICAL" SUPERVISED MACHINE LEARNING

WHAT DOES MACHINE LEARNING DO?

the machine is told what to look for

SUPERVISED

the machine is NOT told what to look for

UN-SUPERVISED

TWO VERY BROAD TYPES OF MACHINE LEARNING ALGORITHMS

CAT

CAT

SUPERVISED LEARNING

the machine is told what to look for

DOG

HUMAN LABELLING

DOG

SUPERVISED LEARNING

CAT

CAT

the machine is told what to look for

DOG

HUMAN LABELLING

TRAINING SET OF LABELED EXAMPLES

DOG

CAT

CAT

DOG

DOG

SUPERVISED LEARNING

ML

CAT

CAT

CAT

DOG

DOG

SUPERVISED LEARNING

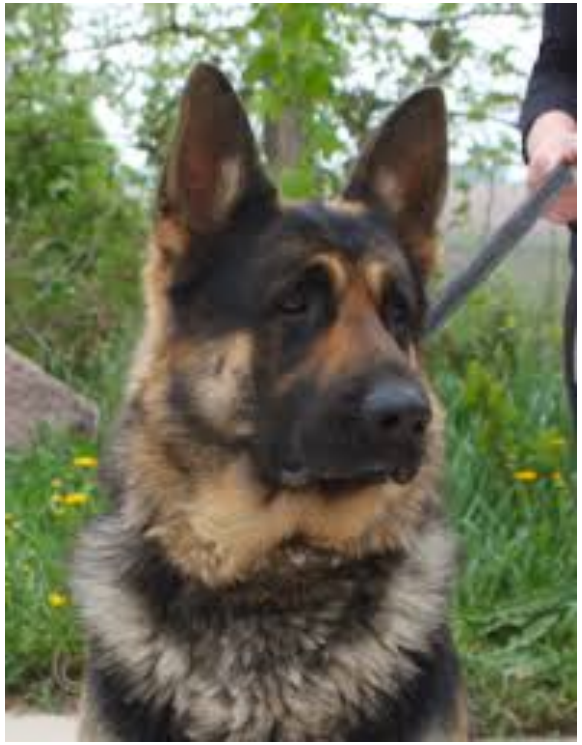
ML

DOG

UNSUPERVISED LEARNING

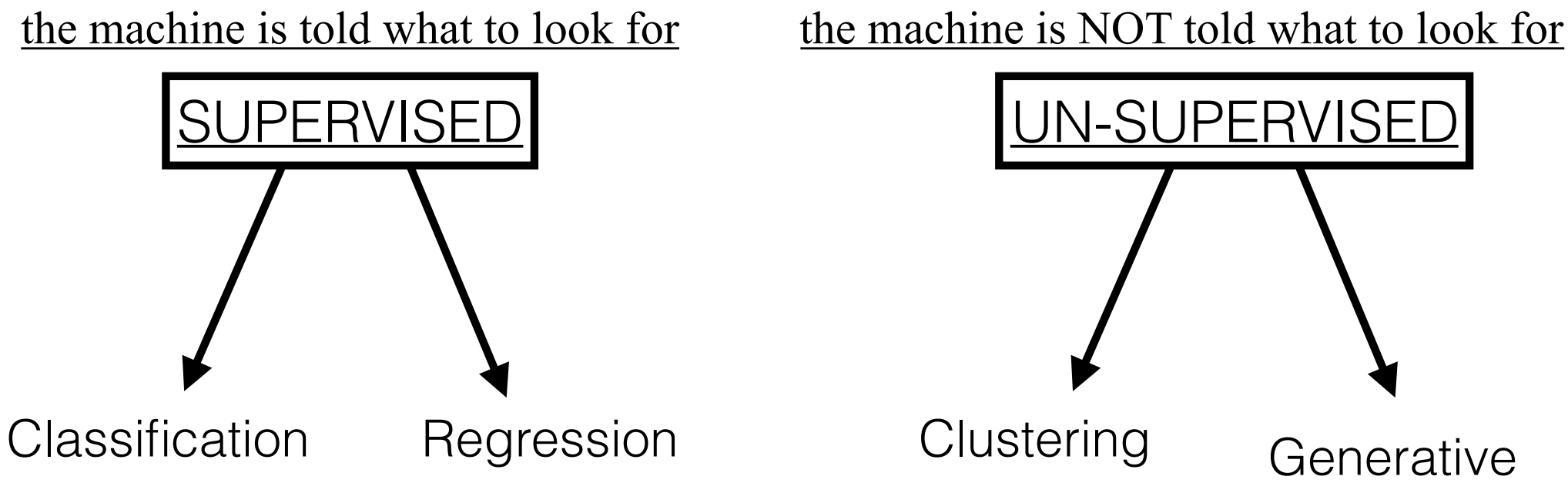
UNSUPERVISED LEARNING

UNSUPERVISED LEARNING

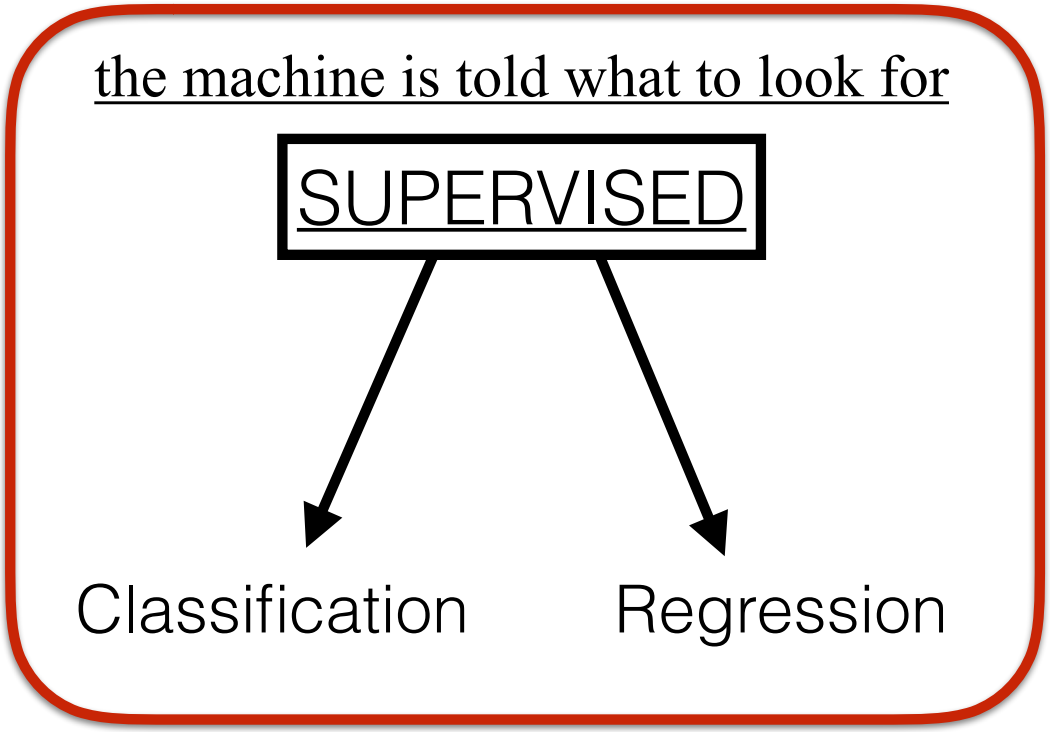


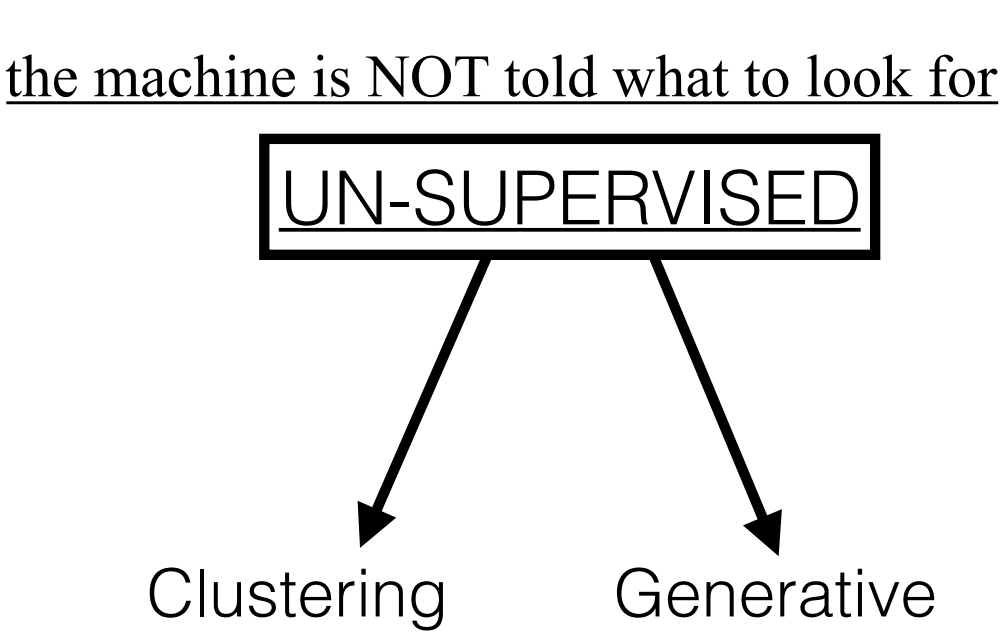
THE DEFINITION OF CLASSES IS SOMETIMES NOT OBVIOUS

WHAT DOES MACHINE LEARNING DO?



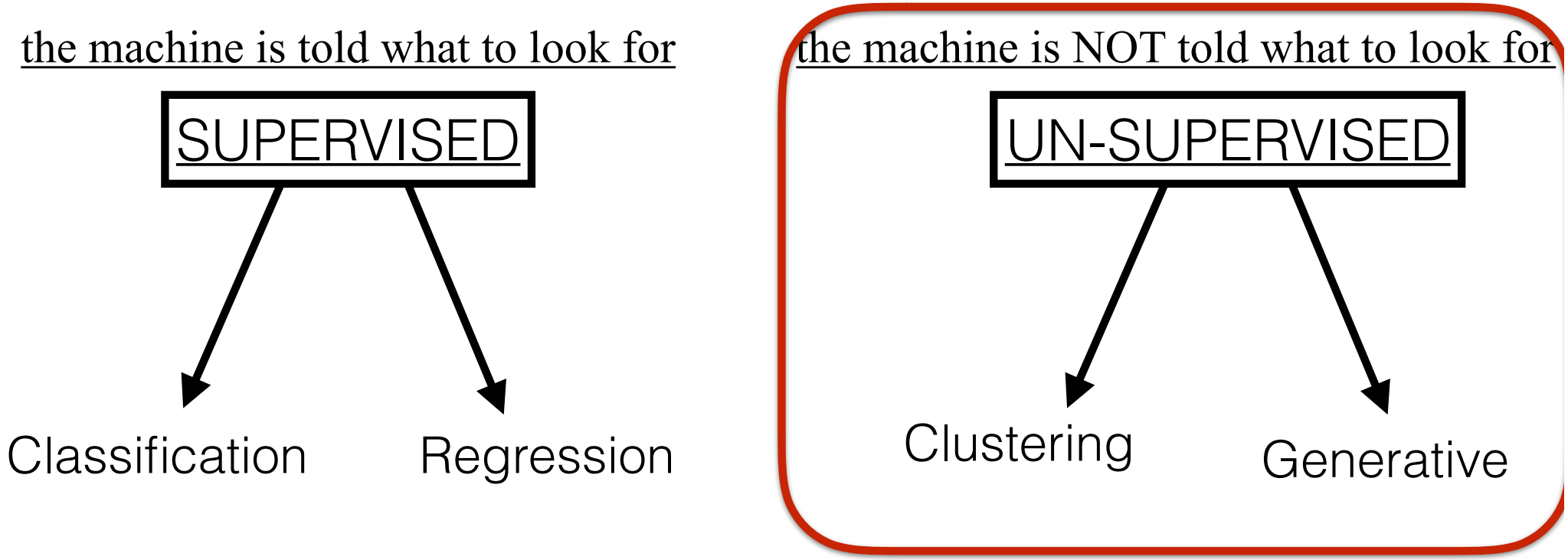
WHAT DOES MACHINE LEARNING DO?





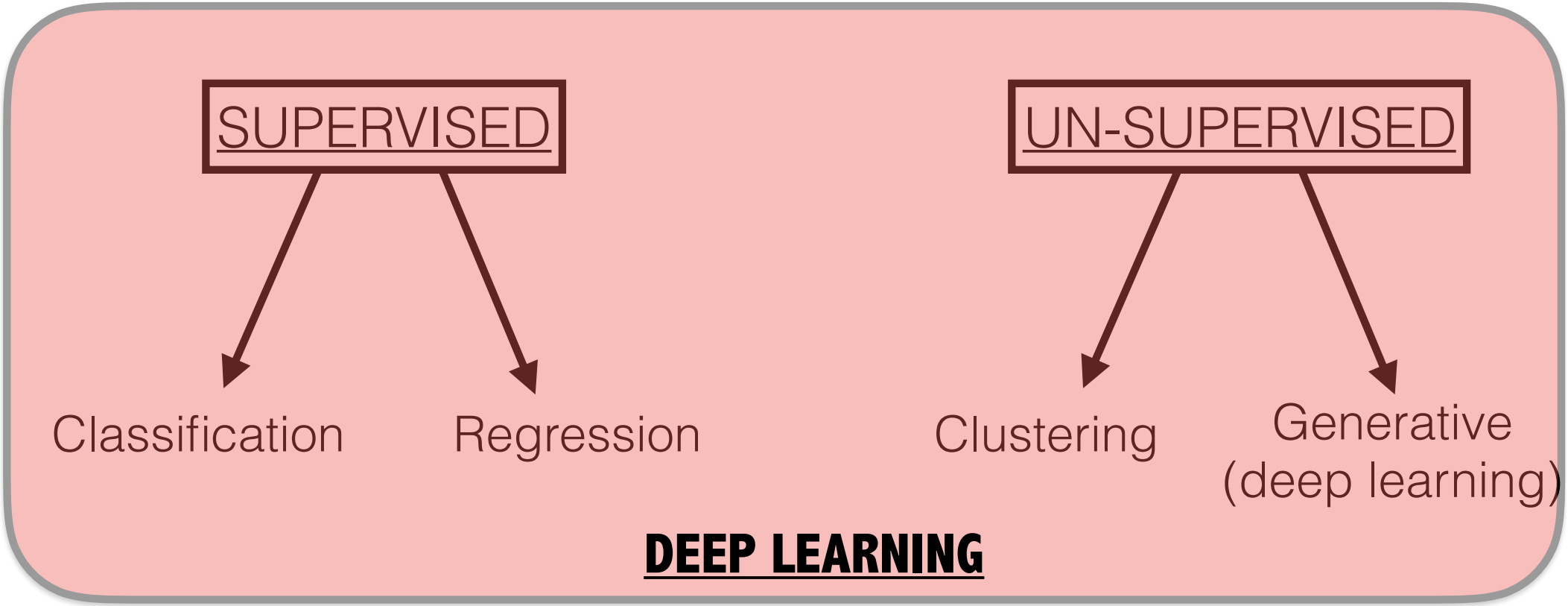
MOSTLY FOCUS ON THIS

WHAT DOES MACHINE LEARNING DO?



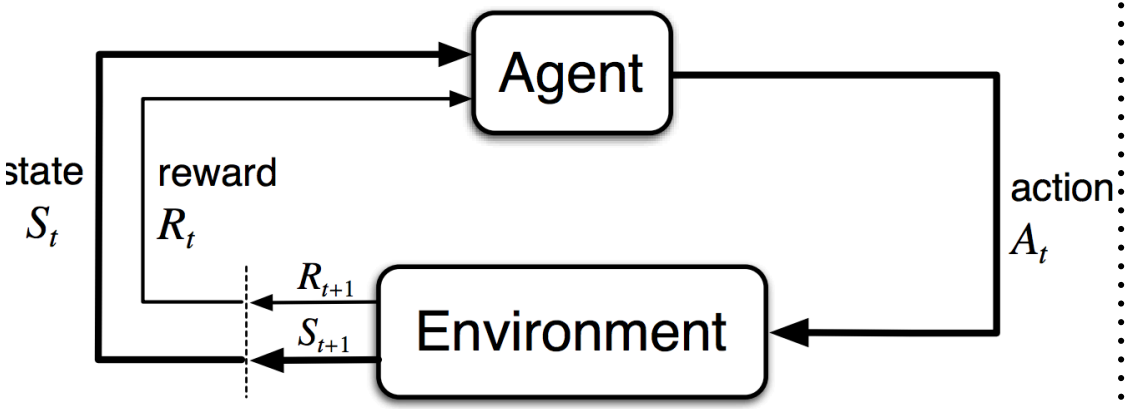
MAYBE VERY BRIEFLY TODAY (OR CYCLE3?)

WHAT DOES MACHINE LEARNING DO?



ACTUALLY, THERE IS A THIRD AND FORTH TYPE THAT WE WILL NOT HAVE TIME TO COVER

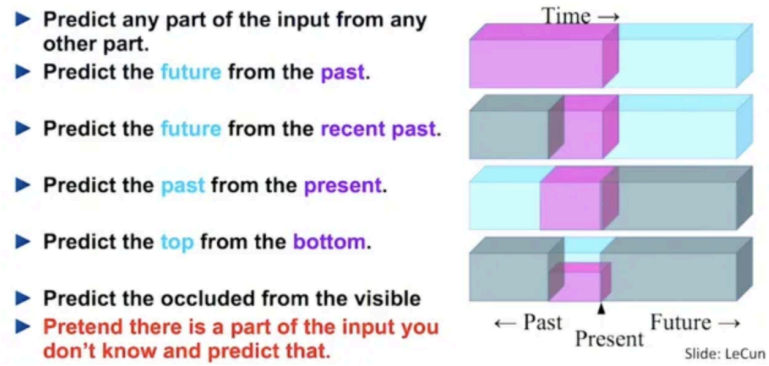
REINFORCMENT LEARNING:



source: Sutton & Barto

ITERATIVE LEARNING THROUGH TRIAL/ERROR: USED FOR ALPHAGO FOR EXAMPLE

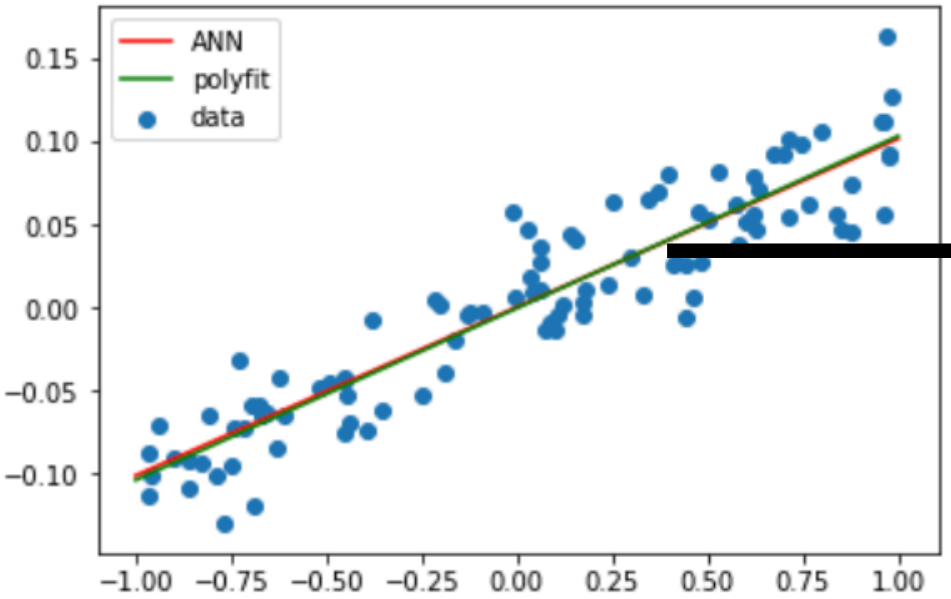
SELF-SUPERVISED LEARNING:



> That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.
>
> Y. LeCun

a self-supervised learning system attempts to predict parts of its inputs based on the other parts of its inputs

THRE IS NO MAGIC IN MACHINE LEARNING,
LINEAR REGRESSION IS MACHINE LEARNING



free parameters

model

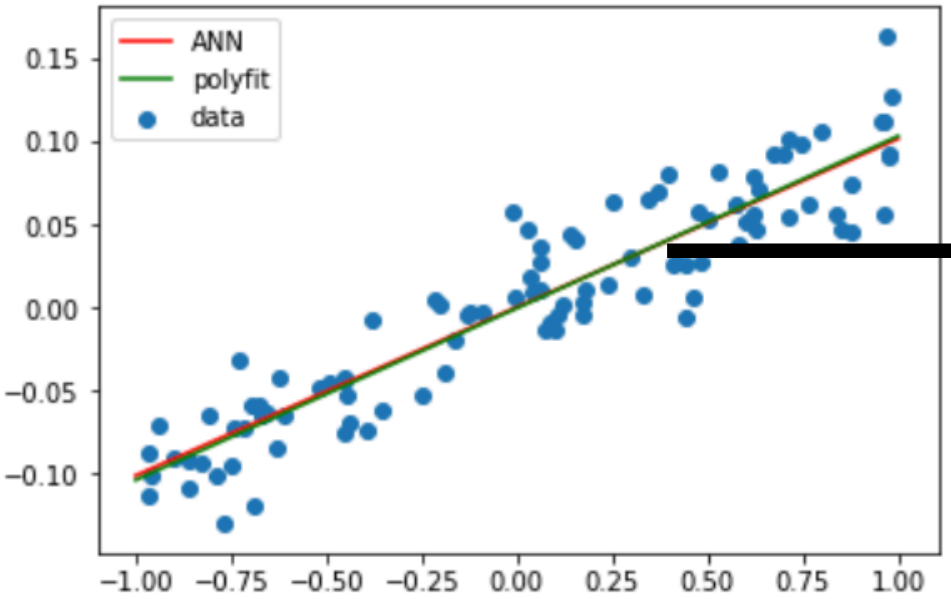
data

$$y = f_w(x) = w_1.x + b$$

we fit a linear model to the data by minimizing the chi2 error:

$$\frac{1}{N}\sum(x^2 - f_w(x)^2)$$

THRE IS NO MAGIC IN MACHINE LEARNING,
LINEAR REGRESSION IS MACHINE LEARNING



weights

neural network

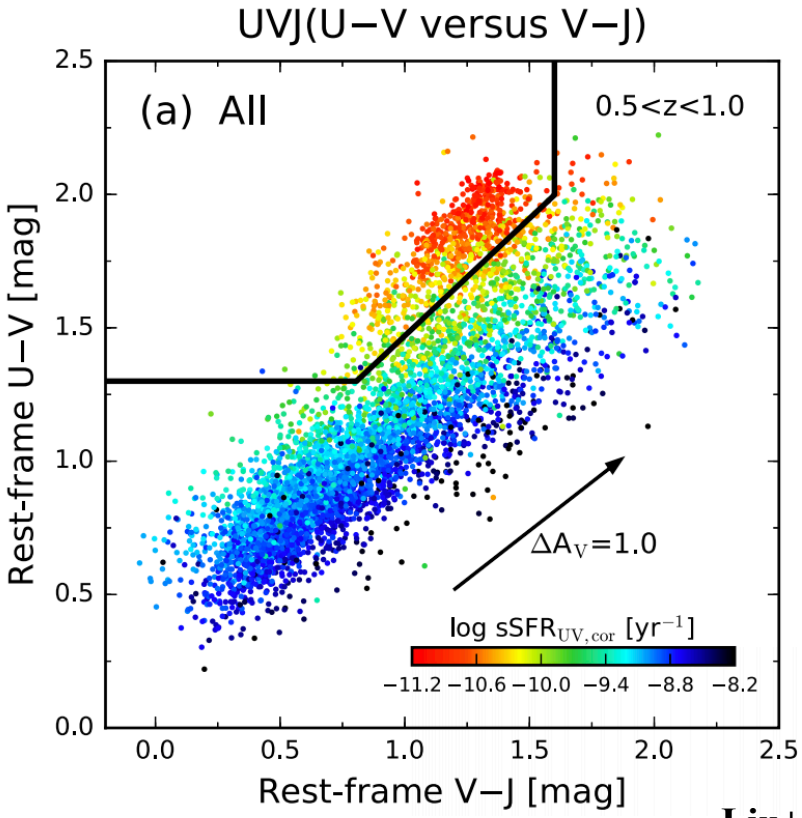
features

$$y = f_w(x) = w_1.x + b$$

we train a neural network with a mse loss function:

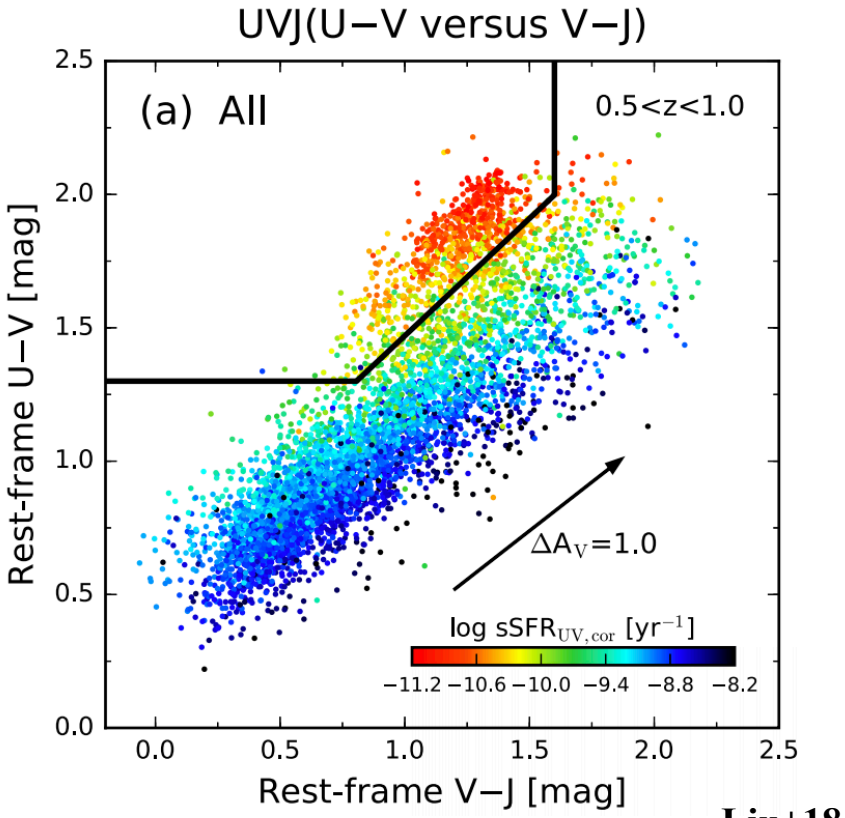
$$\frac{1}{N}\sum(x^2 - f_w(x)^2)$$

THRE IS NO MAGIC IN MACHINE LEARNING, AND IT IS ACTUALLY PRETTY SIMPLE



Liu+18

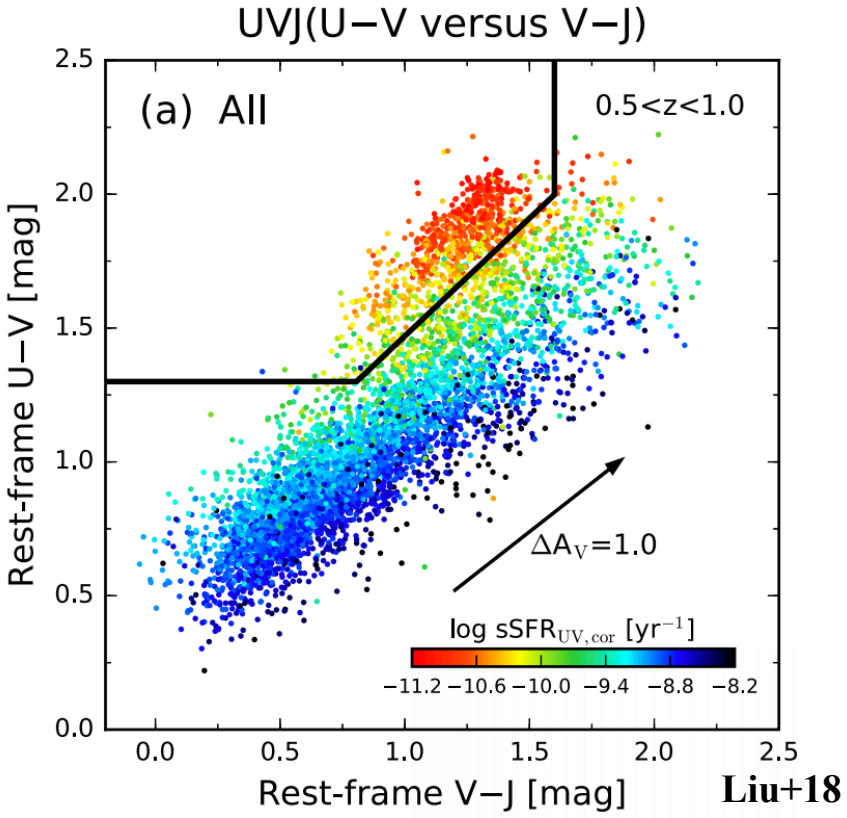
$$f_W(\vec{x}) = \vec{y}$$



Liu+18

$$f_W(\vec{x}) = \vec{y}$$

LABEL
Q , SF

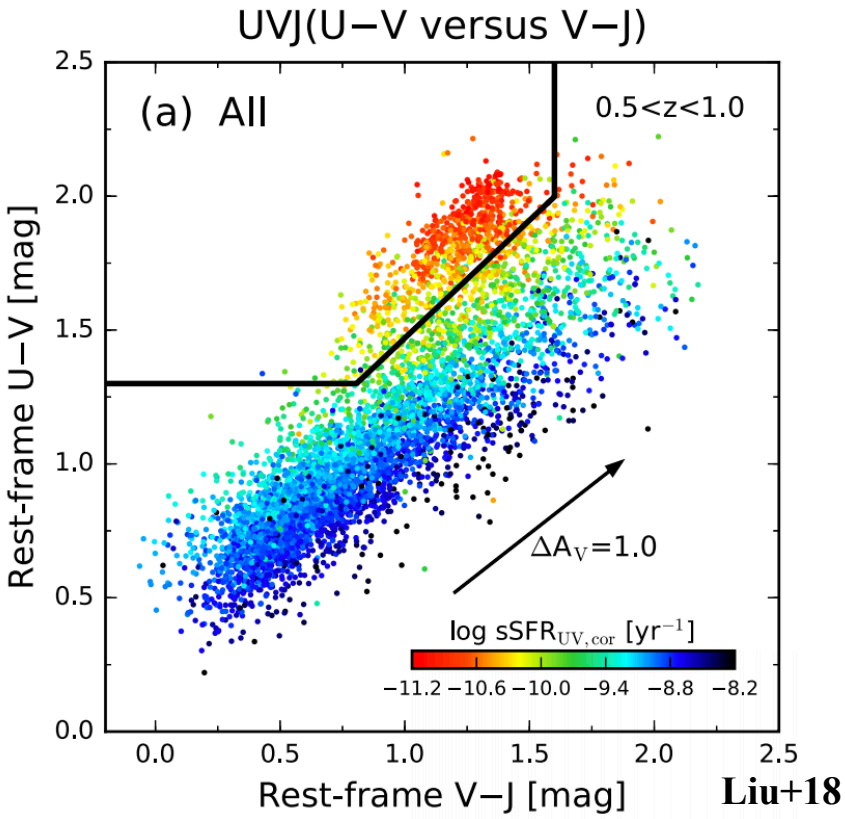


$$f_{\vec{W}}(\vec{x}) = \vec{y}$$

LABEL

Q(0) , SF(1)

(U-V, V-J) FEATURES



$$f_{\vec{W}}(\vec{x}) = \vec{y}$$

LABEL

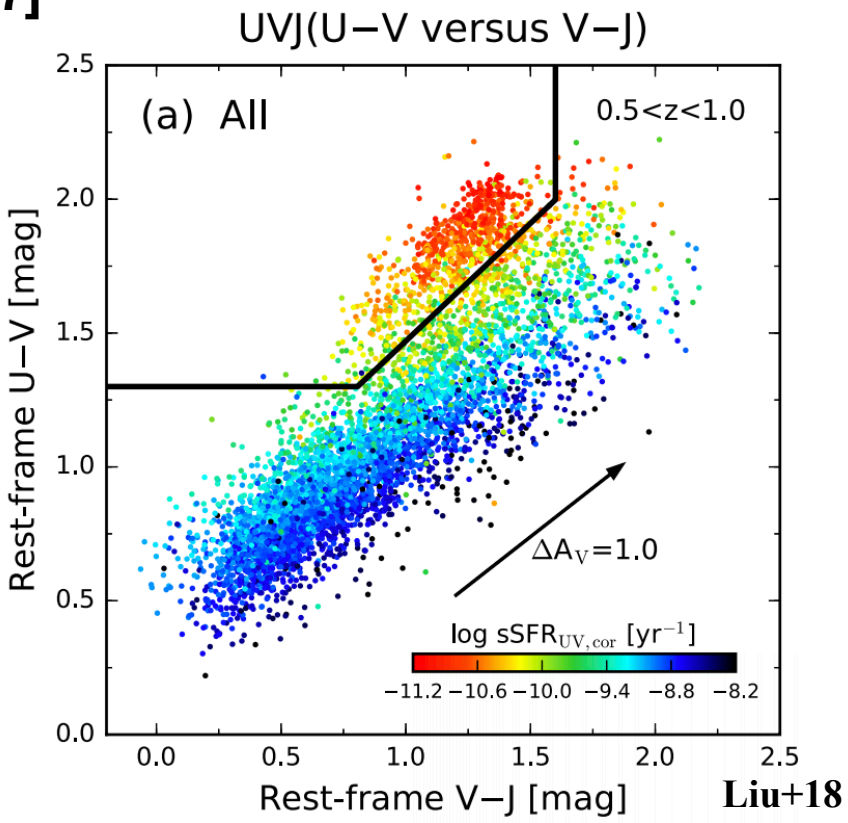
Q(0) , SF(1)

NETWORK FUNCTION

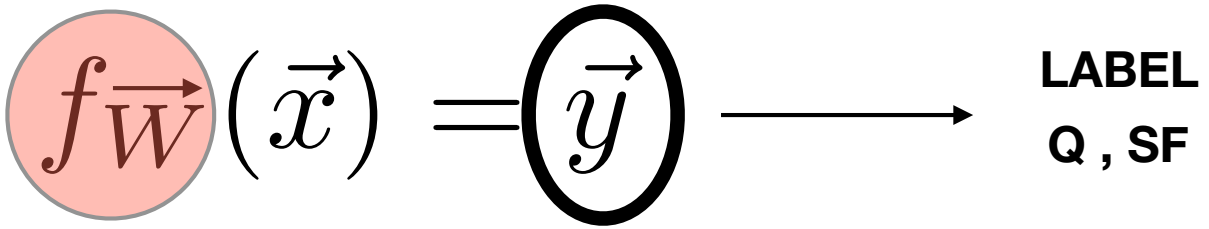
(U-V, V-J) FEATURES

sgn[(u-v)-0.8*(v-j)-0.7]

WEIGHTS



“CLASSICAL”
MACHINE LEARNING



sgn[(u-v)-W1*(v-j)-W2]

REPLACE THIS BY A GENERAL
NON LINEAR FUNCTION WITH SOME PARAMETERS W

SUPERVISED LEARNING

Given a dataset with known labels (measurements) - find a function that can assign (predict) measurements for an unlabeled dataset

Training set

$(\vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_n})$ → Measurements (colors, fluxes, spectra indices…)

$(\vec{y_1}, \vec{y_2}, \vec{y_3}, ..., \vec{y_n})$ → Label (morphology, object type, redshift …)

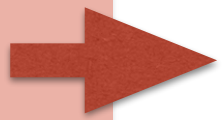
SUPERVISED LEARNING

Given a dataset with known labels (measurements) - find a function that can assign (predict) measurements for an unlabeled dataset

Training set

$(\vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_n})$

$(\vec{y_1}, \vec{y_2}, \vec{y_3}, ..., \vec{y_n})$



$f_W(\vec{x}) = \vec{y}$

?

SUPERVISED LEARNING

Unlabeled set

$(\vec{x_1}', \vec{x_2}', \vec{x_3}', ..., \vec{x_n}')$

$(\vec{y_1}', \vec{y_2}', \vec{y_3}', ..., \vec{y_n}')$

Training set

$(\vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_n})$

$(\vec{y_1}, \vec{y_2}, \vec{y_3}, ..., \vec{y_n})$

$f_W(\vec{x}) = \vec{y}$

?

$$(\vec{x_1}, \vec{x_2}, \vec{x_3}, \ldots, \vec{x_n}) \qquad \vec{x} \in \mathbb{R}^d$$

$$(\vec{y_1}, \vec{y_2}, \vec{y_3}, \ldots, \vec{y_n}) \qquad \vec{y} \in \mathbb{R} \qquad \vec{y} \in \mathbb{N}$$

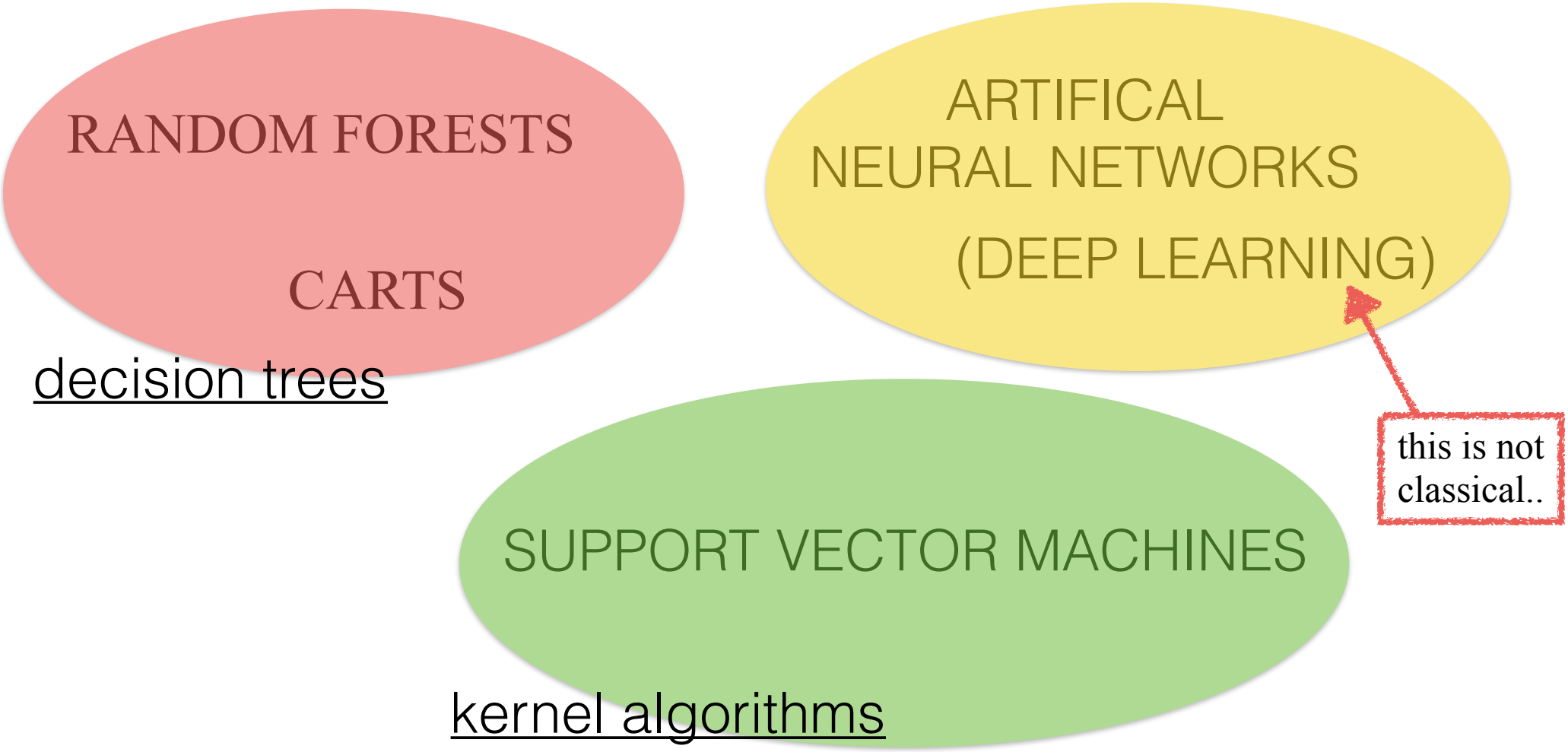
GENERAL GOAL: Find a (non-linear) function that outputs the correct class / measurement for a given input object:

$$f_W(\vec{x})$$

Number of parameters - can be large

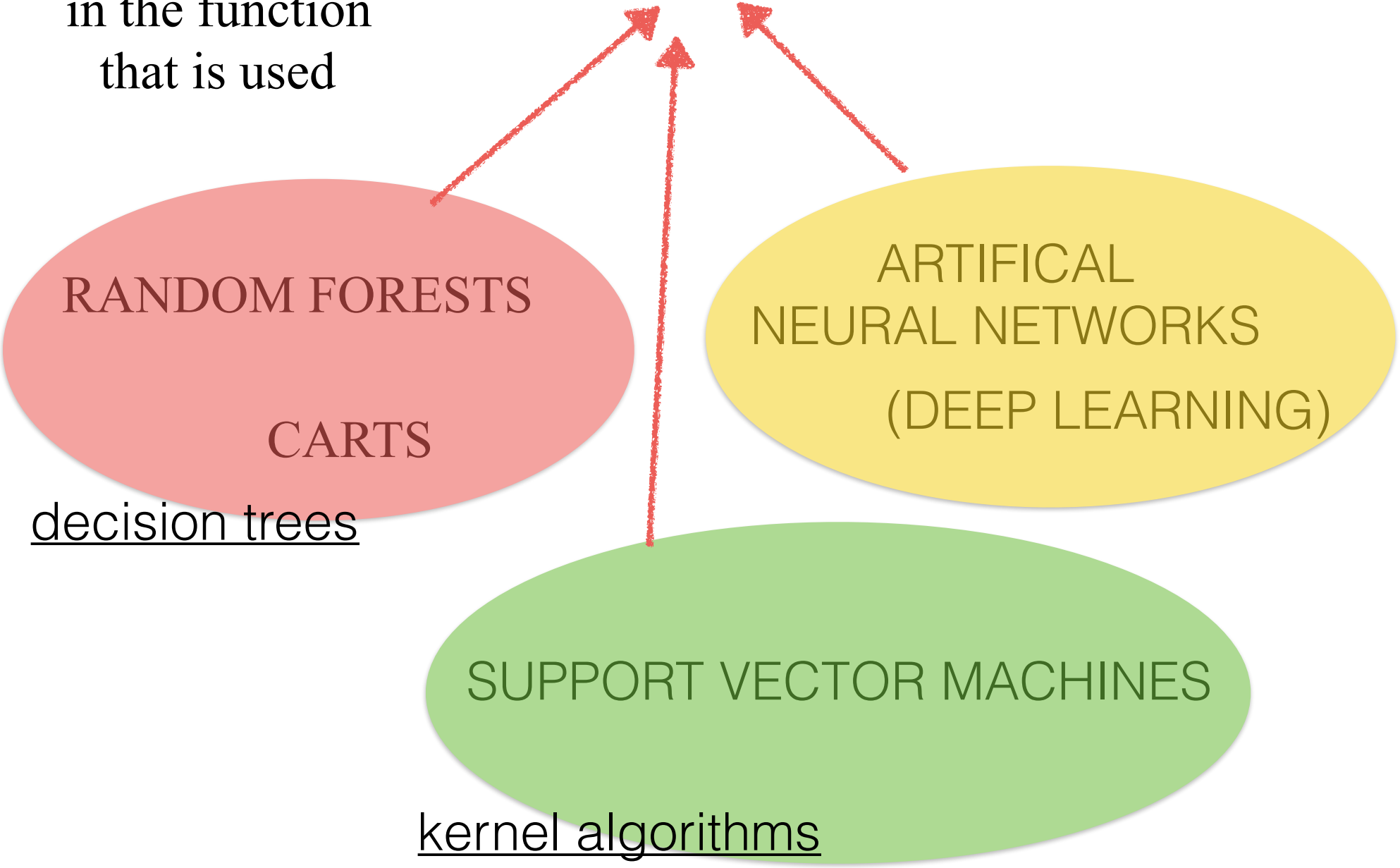
It is translated into a minimization problem : find W such as the prediction error is minimal <u>over all unseen vectors</u>

Different "classical" supervised machine learning methods



The differences are in the function that is used

$$f_W(\vec{x})$$



TO SUMMARIZE: supervised ML is an optimization problem

we need therefore 2 key ingredients:

1. A LOSS FUNCTION

2. A MINIMIZATION OR OPTIMIZATION ALGORITHM

TO SUMMARIZE: supervised ML is an optimization problem

we need therefore 2 key ingredients:

1. A LOSS FUNCTION

2. A MINIMIZATION OR OPTIMIZATION ALGORITHM

THIS IS COMMON TO ALL MACHINE LEARNING ALGORITHMS

1. DEFINE A LOSS FUNCTION

$$loss(F_W(.), \vec{x_i}, \vec{y_i})$$

For example: $(F_W(\vec{x_i}) - \vec{y_i})^2$ (MSE LOSS FUNCTION)

2. MINIMIZE THE EMPIRICAL RISK WITH OPTIMIZATION

$$\Re_{empirical}(W) = \frac{1}{N} \sum_{i}^{N} [loss(W, \vec{x}, \vec{y})]$$

MINIMIZE THE RISK

EMPIRICAL RISK?

$$\Re_{empirical}(W) = \frac{1}{N}\sum_{i}^{N}[loss(W, \vec{x}, \vec{y})]$$

WE ARE MINIMIZING WITH RESPECT TO A FINITE NUMBER OF OBSERVED EXAMPLES

EMPIRICAL RISK?

$$\Re_{empirical}(W) = \frac{1}{N}\sum_{i}^{N}[loss(W, \vec{x}, \vec{y})]$$

WE ARE MINIMIZING WITH RESPECT TO A FINITE NUMBER OF OBSERVED EXAMPLES

OBSERVED DATASET

EMPIRICAL RISK?

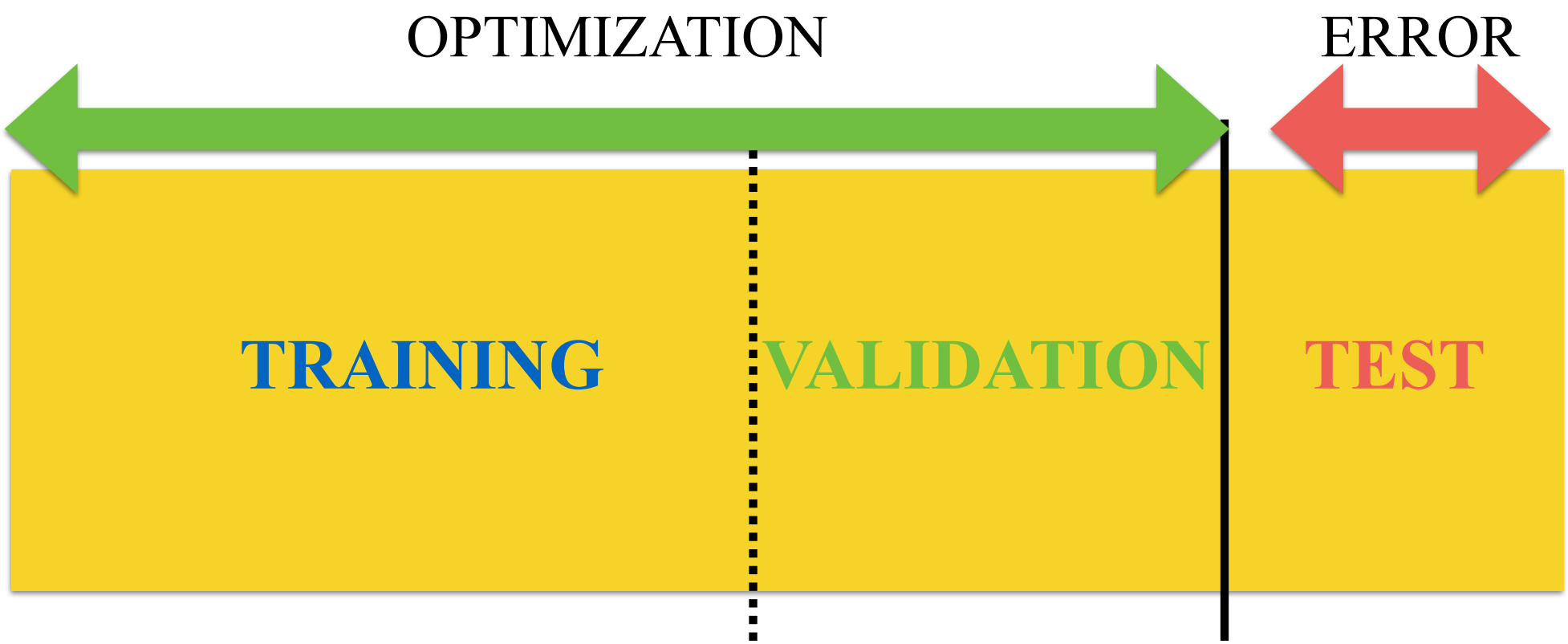
$$\Re_{empirical}(W) = \frac{1}{N}\sum_{i}^{N}[loss(W, \vec{x}, \vec{y})]$$

WE ARE MINIMIZING WITH RESPECT TO A FINITE NUMBER OF OBSERVED EXAMPLES

ALL “GALAXIES IN THE UNIVERSE”

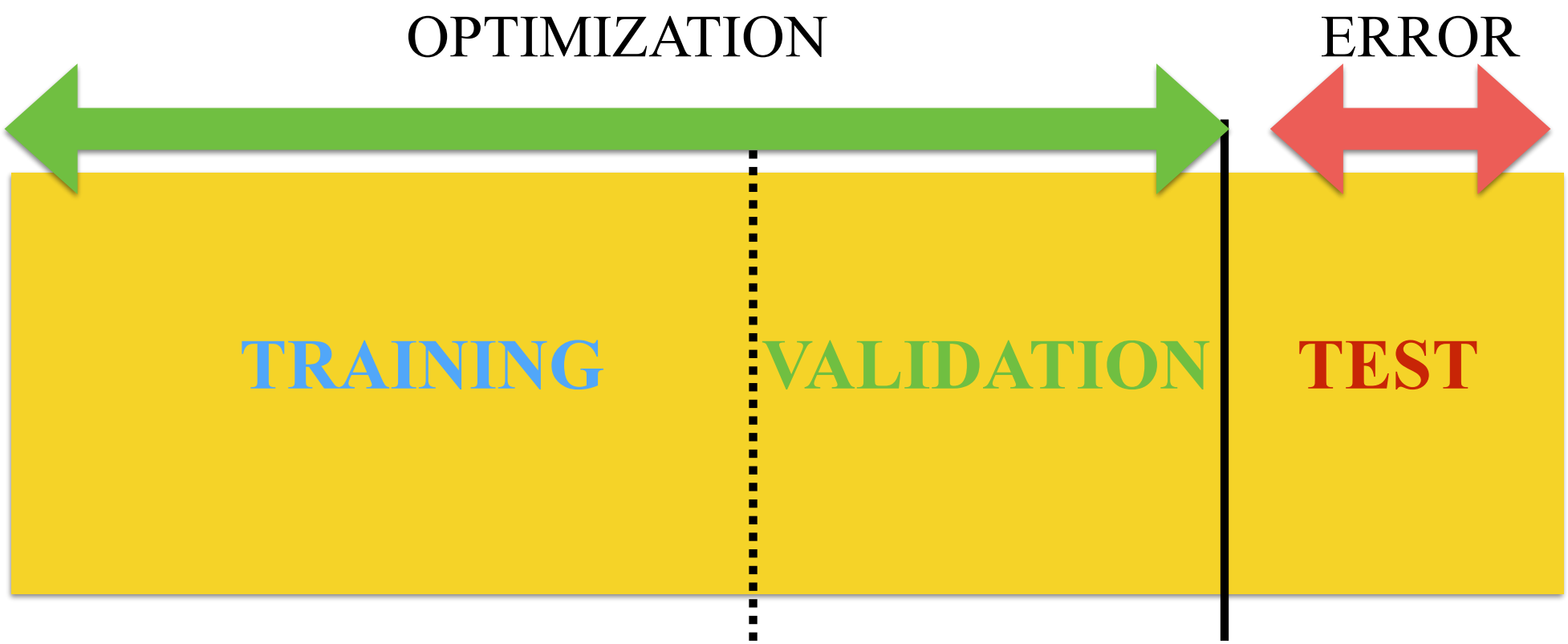
OBSERVED DATASET

IN PRACTICE



training set: use to train the classifier
validation set: use to monitor performance in real time - check for overfitting
test set: use to train the classifier

IN PRACTICE



NO CHEATING! NEVER USE TRAINING TO VALIDATE YOUR ALGORITHM!

The algorithm used to minimize is called **OPTIMIZATION**

THERE ARE SEVERAL OPTIMIZATION TECHNIQUES

Optimization

THERE ARE SEVERAL OPTIMIZATION TECHNIQUES

THEY DEPEND ON THE MACHINE LEARNING ALGORITHM

Optimization

THERE ARE SEVERAL OPTIMIZATION TECHNIQUES

THEY DEPEND ON THE MACHINE LEARNING ALGORITHM

NEURAL NETWORKS USE THE GRADIENT DESCENT AS WE WILL SEE LATER

$$W_{t+1} = W_t - \lambda_h \bigtriangledown f(W_t)$$

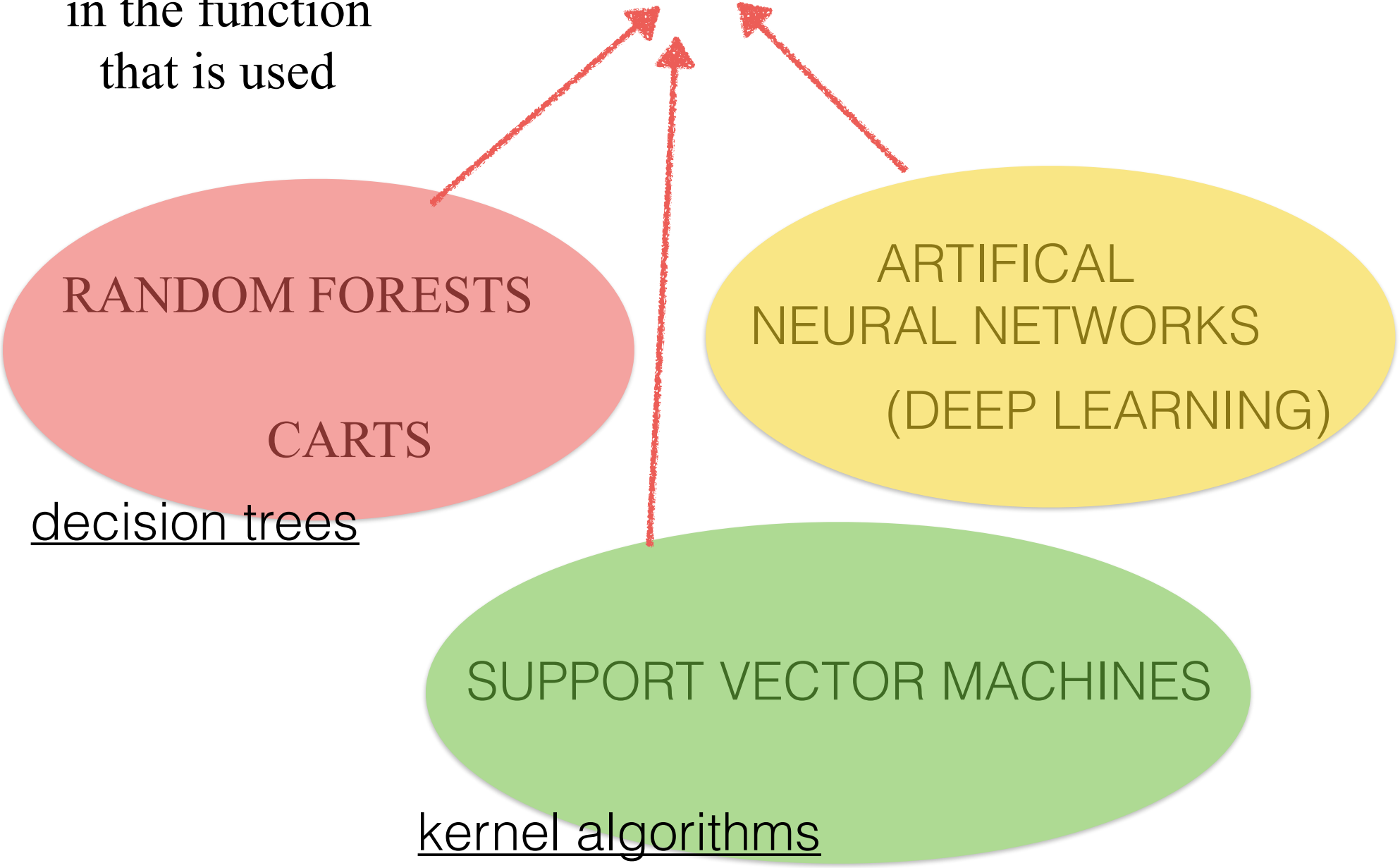
weights to be learned

epoch

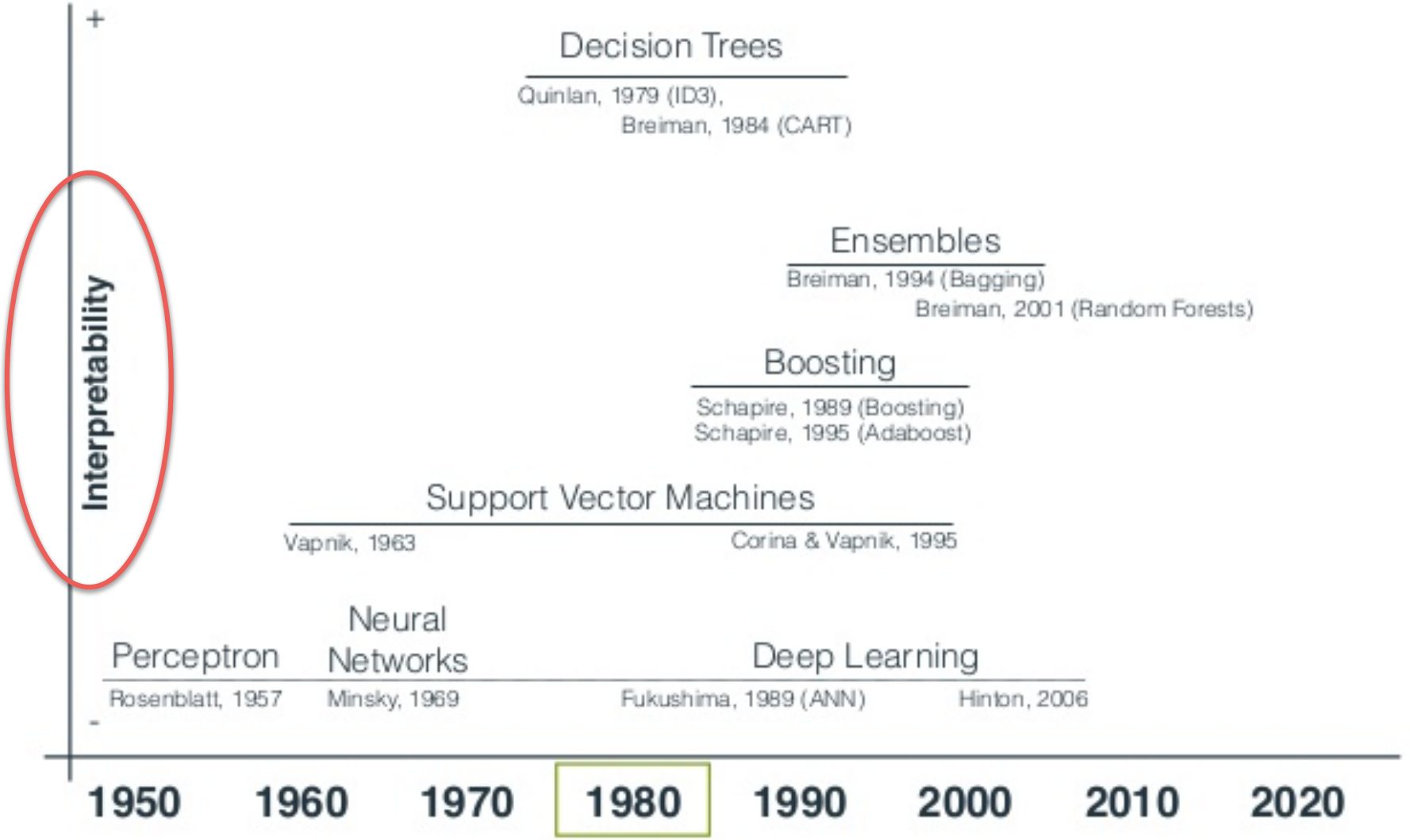
learning rate
(hyper parameter)

The differences are in the function that is used

$$f_W(\vec{x})$$

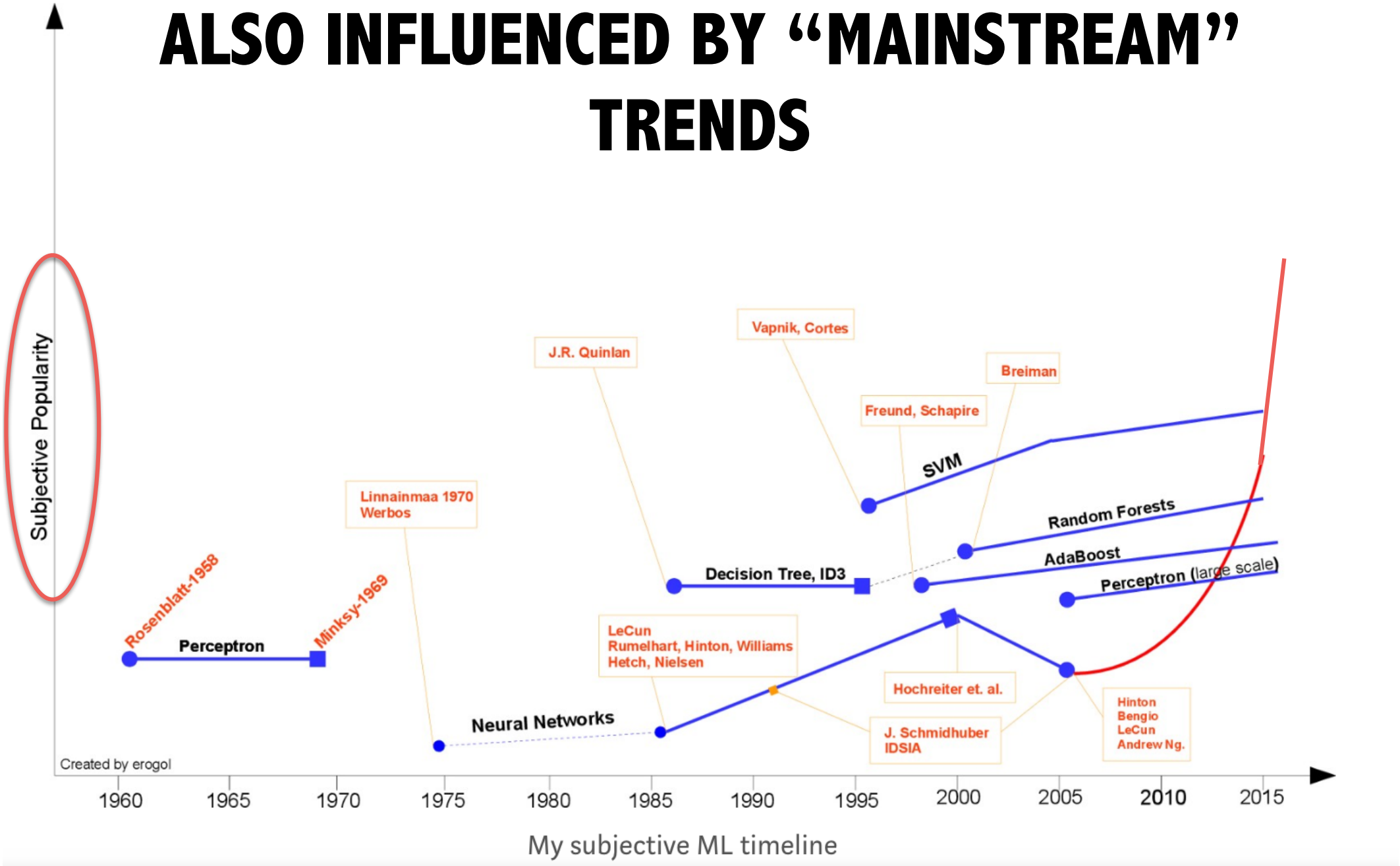


USING ONE OR ANOTER METHOD DEPENDS ON YOUR MAIN OBJECTIVES



credit

ALSO INFLUENCED BY "MAINSTREAM" TRENDS



Source