Machine Learning Algorithms

Dr. Leila
Rzayeva

Lecture 1

Introduction in Machine Learning

Course structure

- <30 hours of Oral Presentations
- <20 hours Practice: homework/in-class assignments -> 60%
- Final exam -> 40%

Lab “Targets”

- Weekly targets for your practical work
  - *Complete them on time!*
- You’re an adult – manage your own time
- All material on Moodle
- It is an honor code violation to intentionally refer to other’s complete assignments

Assignment

- Work on assignments individually (!!!)
- Conduct a deep study of any topic in ML that you want!
  (some suggestions will be provided)
- Produce a 3 page report

Outline

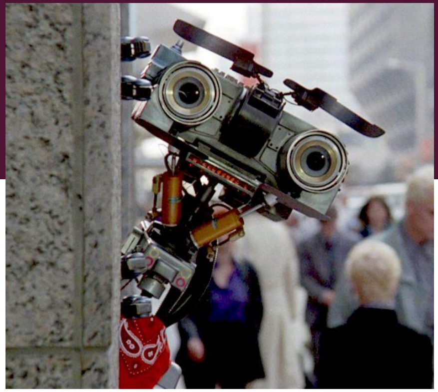
- Difference between AI and Machine Learning?
- Processes behind AI system
- Applications of AI & ML
- Basic Concepts of Machine Learning
- Rugby players and Ballet dancers example with Linear and Nearest Neighbour classifiers

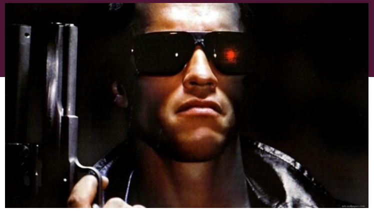
Artificial Intelligence

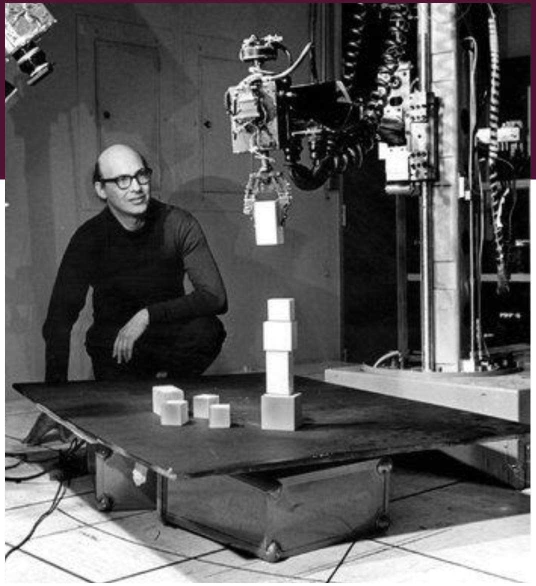
Worldwide A.I. investment to top $200bn by 2025

KPMG. July 31, 2018

"We view AI as an ecosystem that unlocks value by enhancing, accelerating, and automating decisions that drive growth and profitability."








Google


amazon

Artificial Intelligence

Intelligence

Seeing

Moving

Listening

Language

Learning

Thinking

Artificial Intelligence

Computer Vision
Seeing

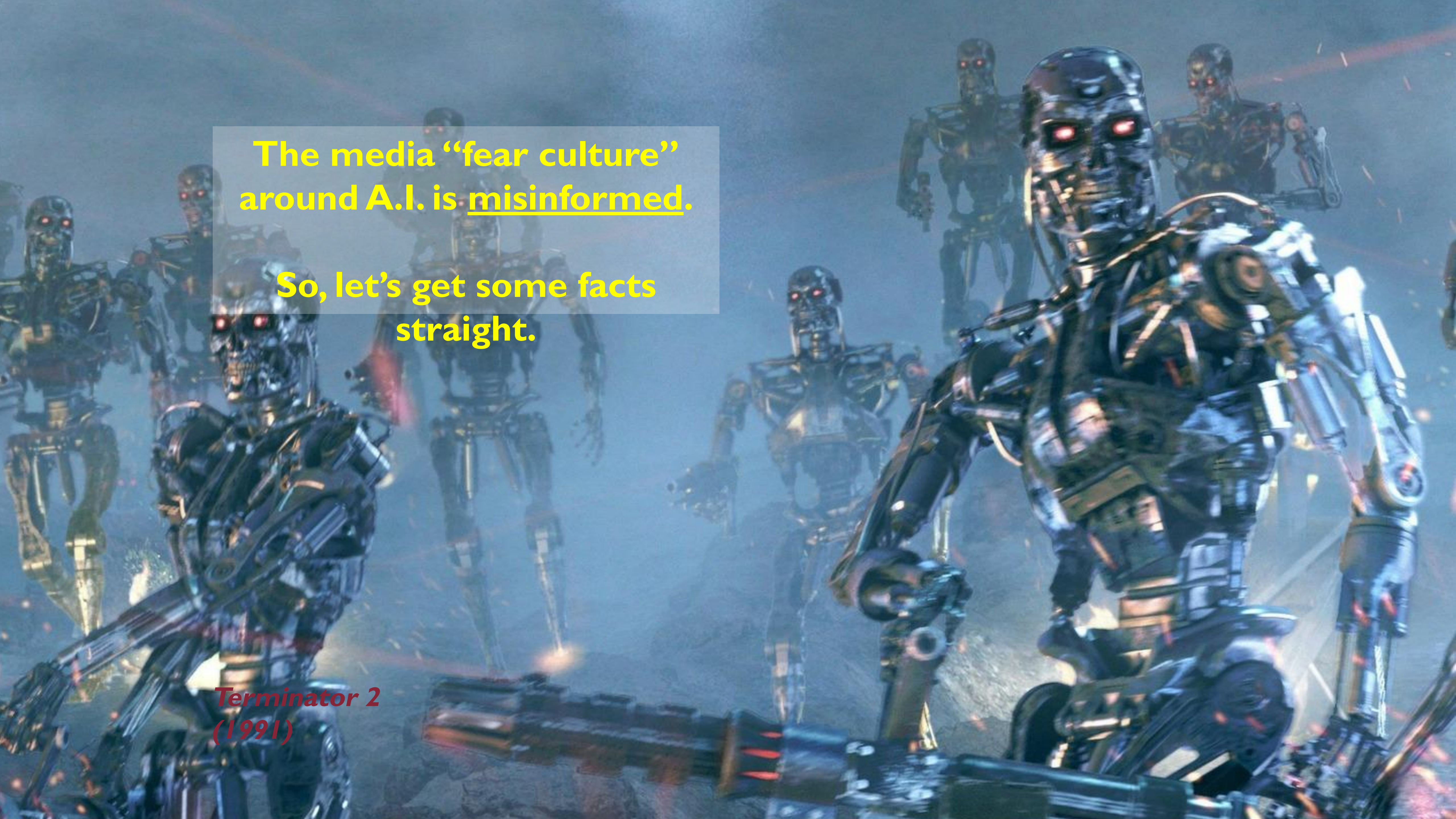
Robotics
Moving

Speech Recognition
Listening

NLP
Language

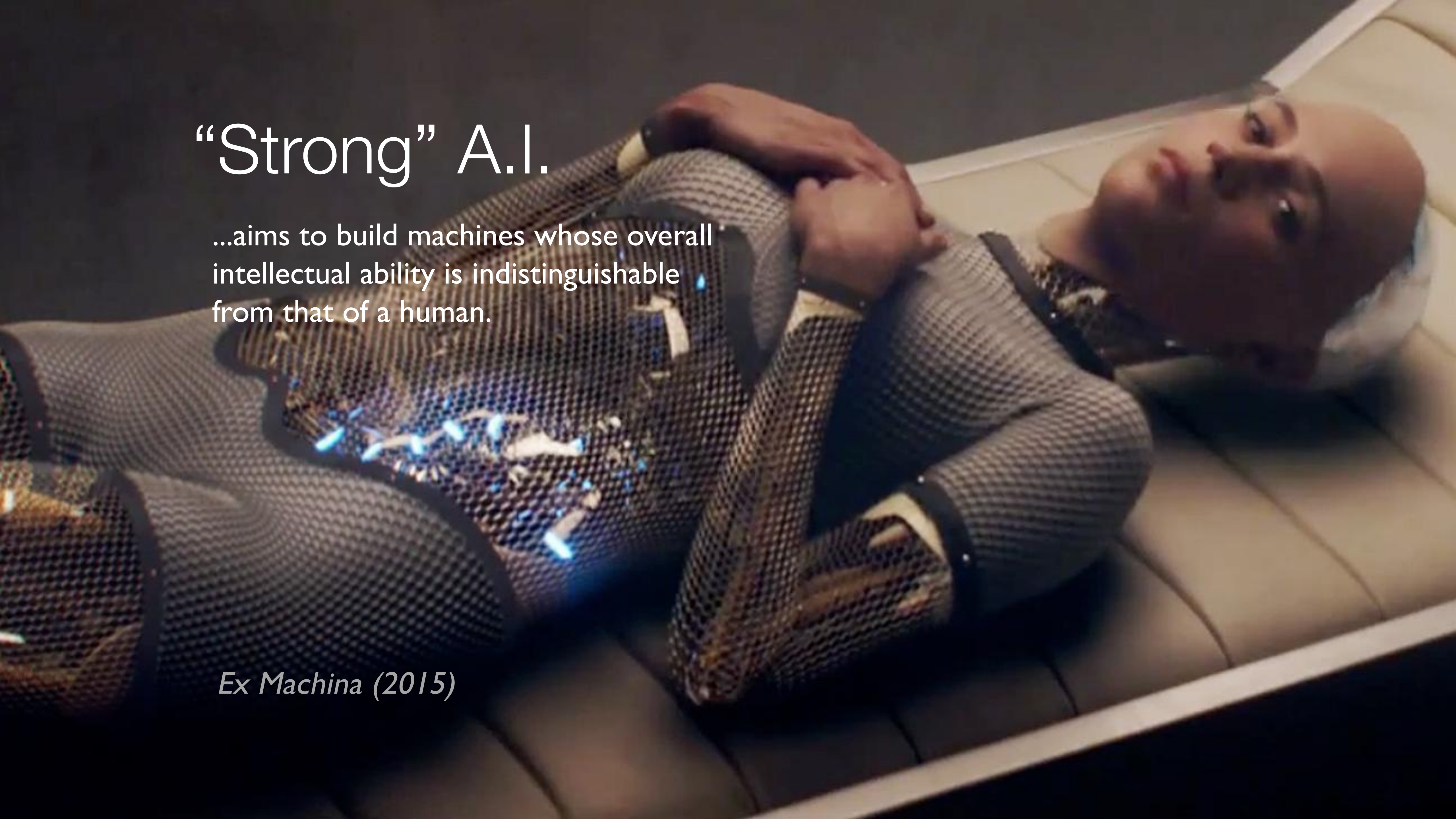
Machine Learning
Learning

Automated Reasoning
Thinking


The media "fear culture" around A.I. is misinformed.
So, let's get some facts straight.
Terminator 2 (1991)

"Strong" A.I.

...aims to build machines whose overall intellectual ability is indistinguishable from that of a human.



Ex Machina (2015)

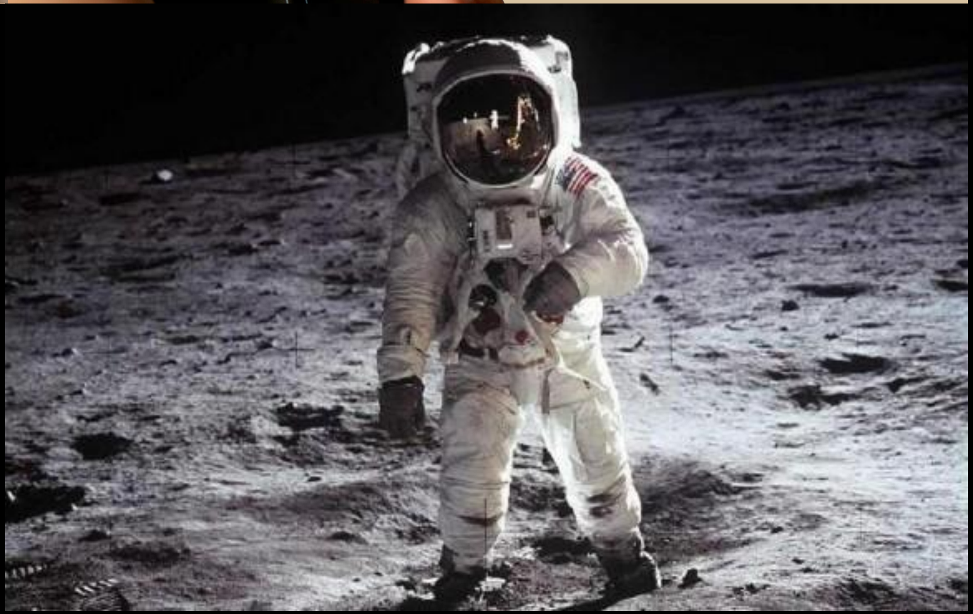
“Weak” A.I.

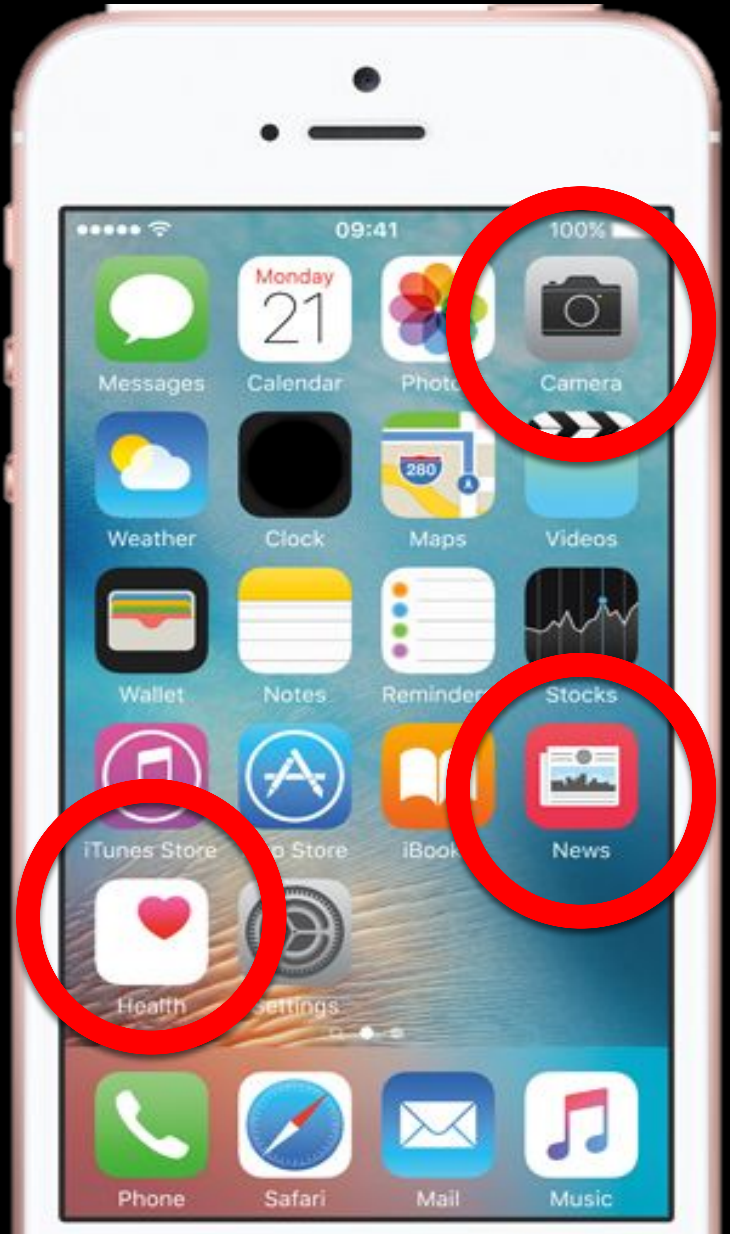
…aims to engineer commercially viable "smart" systems

Science Fiction

2050?

2500?





09:41
100%
Monday
21
Messages
Calendar
Camera
Weather
Clock
Maps
Videos
Wallet
Notes
Stocks
iTunes Store
News
Health
Phone
Safari
Mail
Music

Computer Vision

Seeing

Artificial Intelligence

Robotics

Moving

Speech Recognition

Listening

Machine Learning

Learning

Automated Reasoning

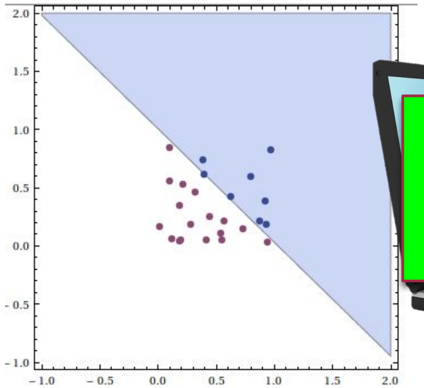
Thinking

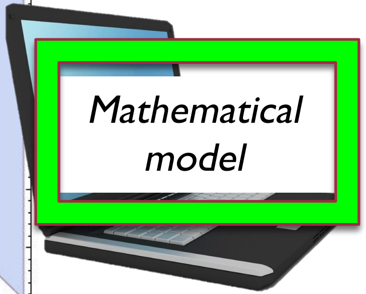
NLP

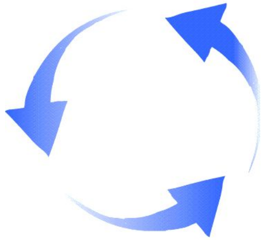
Language

Machine Learning

Examples of what we want

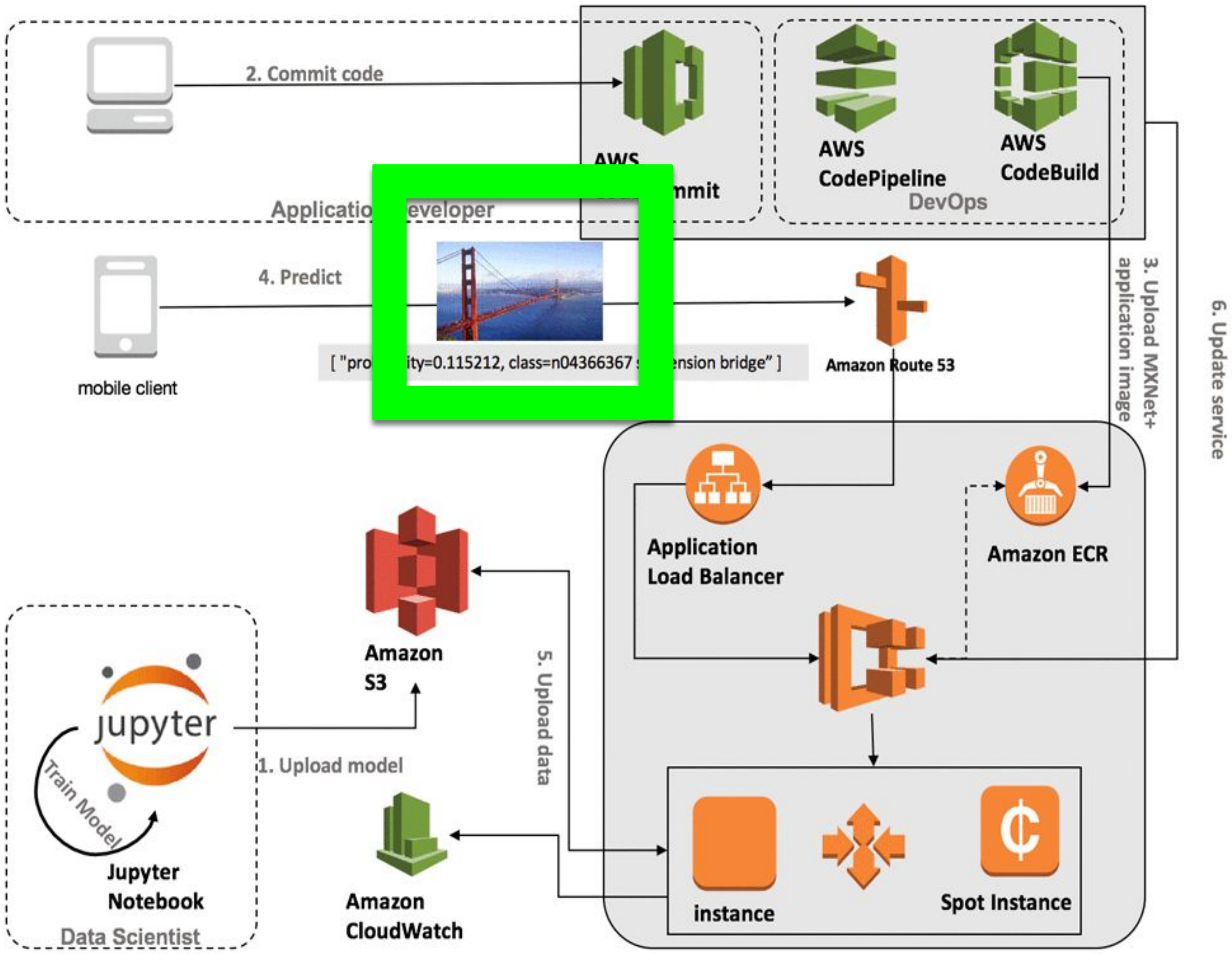






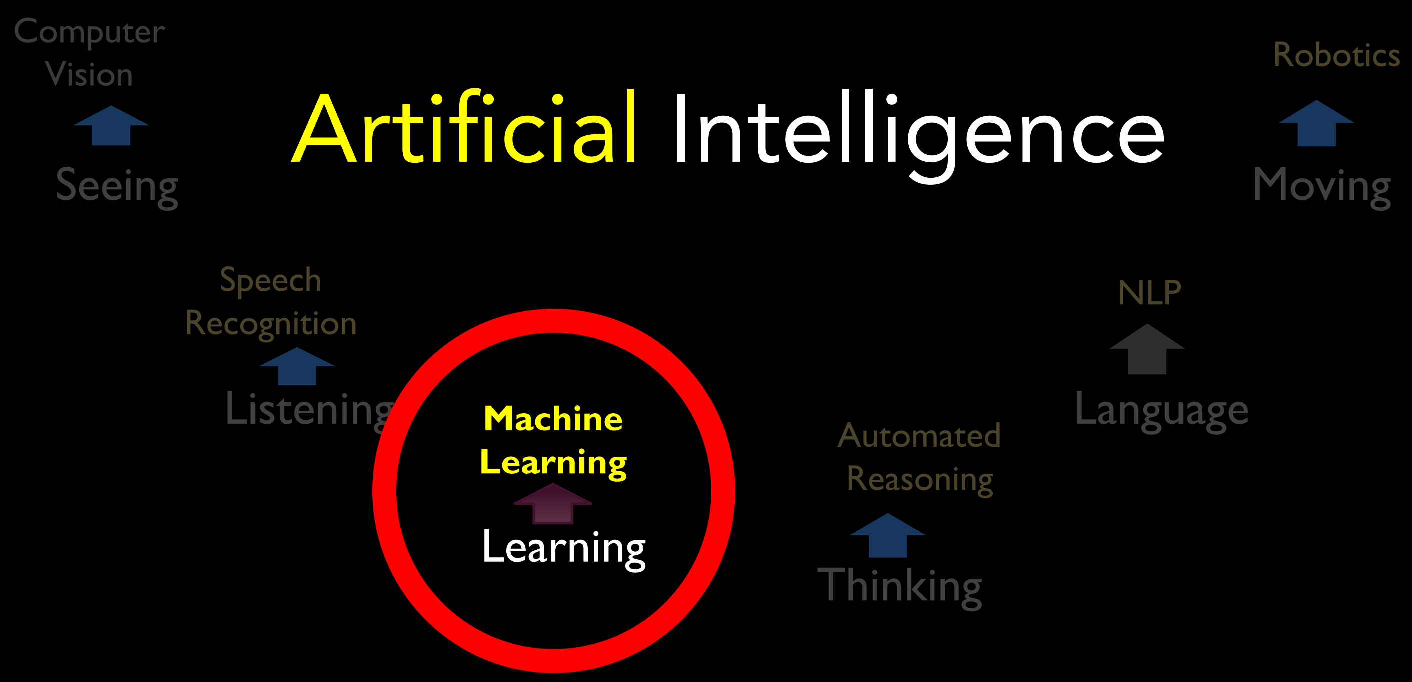
$$P = \frac{e^{(\beta + \alpha_1 X_1 + \ldots \alpha_n X_n)}}{1 + e^{(\beta + \alpha_1 X_1 + \ldots \alpha_n X_n)}}$$

Mathematical model, tuned for your task



2. Commit code
AWS
mmit
Applicatio
eveloper
AWS
CodePipeline
AWS
CodeBuild
DevOps
4. Predict
["pro
ity=0.115212, class=n04366367
ension bridge"]
mobile client
Amazon Route 53
3. Upload MXNet+
application image
6. Update service
Application
Load Balancer
Amazon ECR
Amazon
S3
5. Upload data
jupyter
Train Model
1. Upload model
Jupyter
Notebook
Data Scientist
Amazon
CloudWatch
instance
Spot Instance

Artificial Intelligence



Artificial Intelligence

Computer Vision
Seeing

Robotics
Moving

Speech Recognition
Listening

Machine Learning
Learning

Automated Reasoning
Thinking

NLP
Language

$$
\begin{aligned}
&\neg(\neg p \wedge q) \wedge (p \vee q) \\
&\equiv (\neg\neg p \vee \neg q) \wedge (p \vee q) \\
&\equiv (p \vee \neg q) \wedge (p \vee q) \\
&\equiv p \vee (\neg q \wedge q) \\
&\equiv p \vee \text{False} \\
&\equiv p
\end{aligned}
$$

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

$$h(\boldsymbol{x}_i) = sign\left(\sum_{j=1}^{s} \alpha_j \; y_j \; K(\boldsymbol{x}_j, \boldsymbol{x}_i) + b \right)$$

$$K(\boldsymbol{v}, \boldsymbol{v}') = \exp\left(\frac{||\boldsymbol{v} - \boldsymbol{v}'||^2}{2\gamma^2}\right)$$

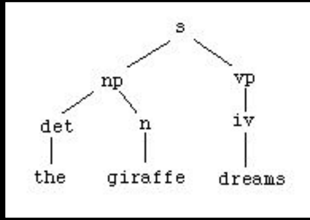
Artificial Intelligence

Vision
Robotics

Speech
Language

Reasoning

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$



```
X = np.array([ [0,0,1],[0,1,1],[1,0,1],[1,1,1] ])
y = np.array([[0,1,1,0]]).T
syn0 = 2*np.random.random((3,4)) - 1
syn1 = 2*np.random.random((4,1)) - 1
for j in xrange(60000):
    l1 = 1/(1+np.exp(-(np.dot(X,syn0))))
    l2 = 1/(1+np.exp(-(np.dot(l1,syn1))))
    l2_delta = (y - l2)*(l2*(1-l2))
    l1_delta = l2_delta.dot(syn1.T) * (l1 * (1-l1))
    syn1 += l1.T.dot(l2_delta)
    syn0 += X.T.dot(l1_delta)
```

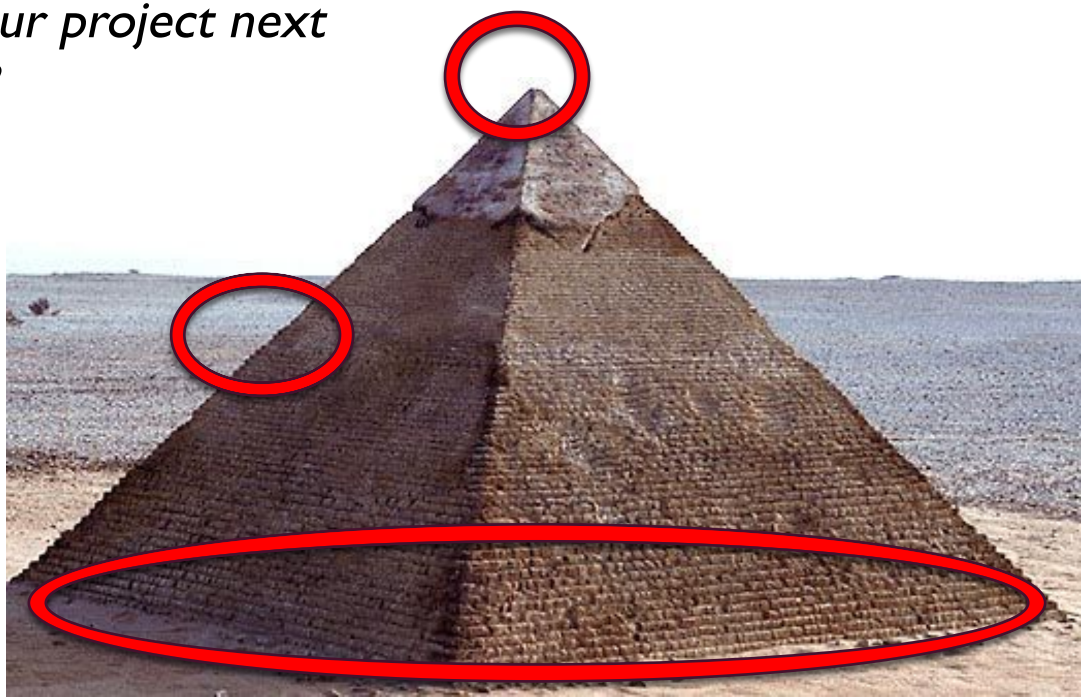
This
course…

Machine Learning Algorithms

Research in ML / AI
…Your project next year?

"Deep Learning"

This
course



Machine Learning

Artificial Intelligence

Computer Vision

Seeing

Speech Recognition

Listening

Machine Learning

Learning

Automated Reasoning

Thinking

NLP

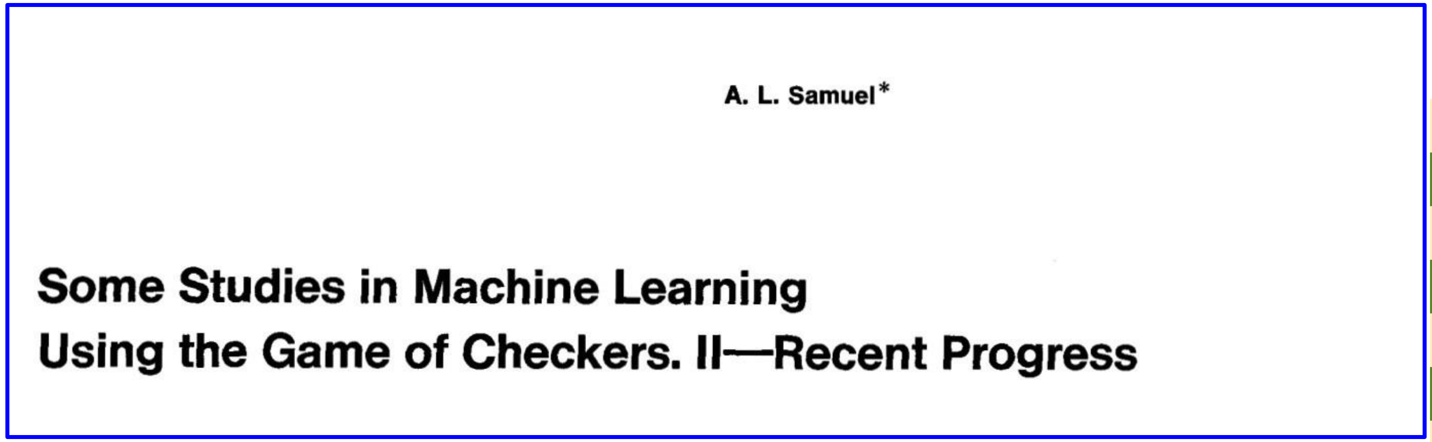
Language

Robotics

Moving

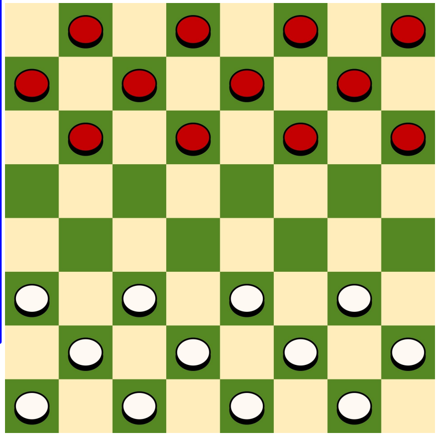
Definition of Machine Learning

Arthur Samuel (1959): Machine Learning is the field of study that gives the computer the ability to learn without being explicitly programmed.



A. L. Samuel*

Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress



Photos from Wikipedia

Tom Mitchell (1998): a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Experience (data): games played by the program (with itself)

Performance measure: winning rate

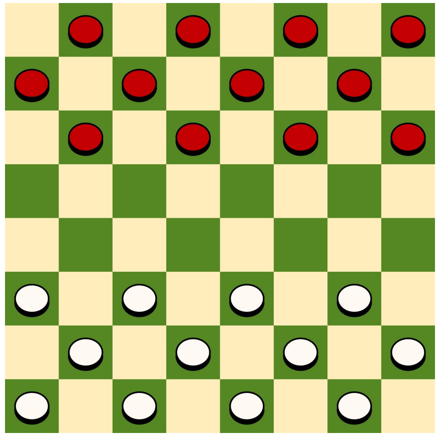


Image from Tom Mitchell's homepage

What are you?

Human ability	*Name of A.I. Research Field*
Seeing	Computer vision
Talking	Speech synthesis
Listening	Speech recognition
Understanding language	Natural Language Processing
Reasoning	Automated Reasoning
Consciousness	Philosophy / Cognitive Science
Walking / moving around	Robotics
Learning	**Machine Learning**

“Learning” is a process

- not specific to a substrate (e.g. biological neurons)

- can be mechanized, with a careful definition

Machine Learning algorithms need data

Predicting health of a patient needs measurements.

- Height
- Weight
- Systolic blood pressure
- Diastolic blood pressure
- Enzyme levels
- Blood sugar levels

Machine Learning algorithms need data

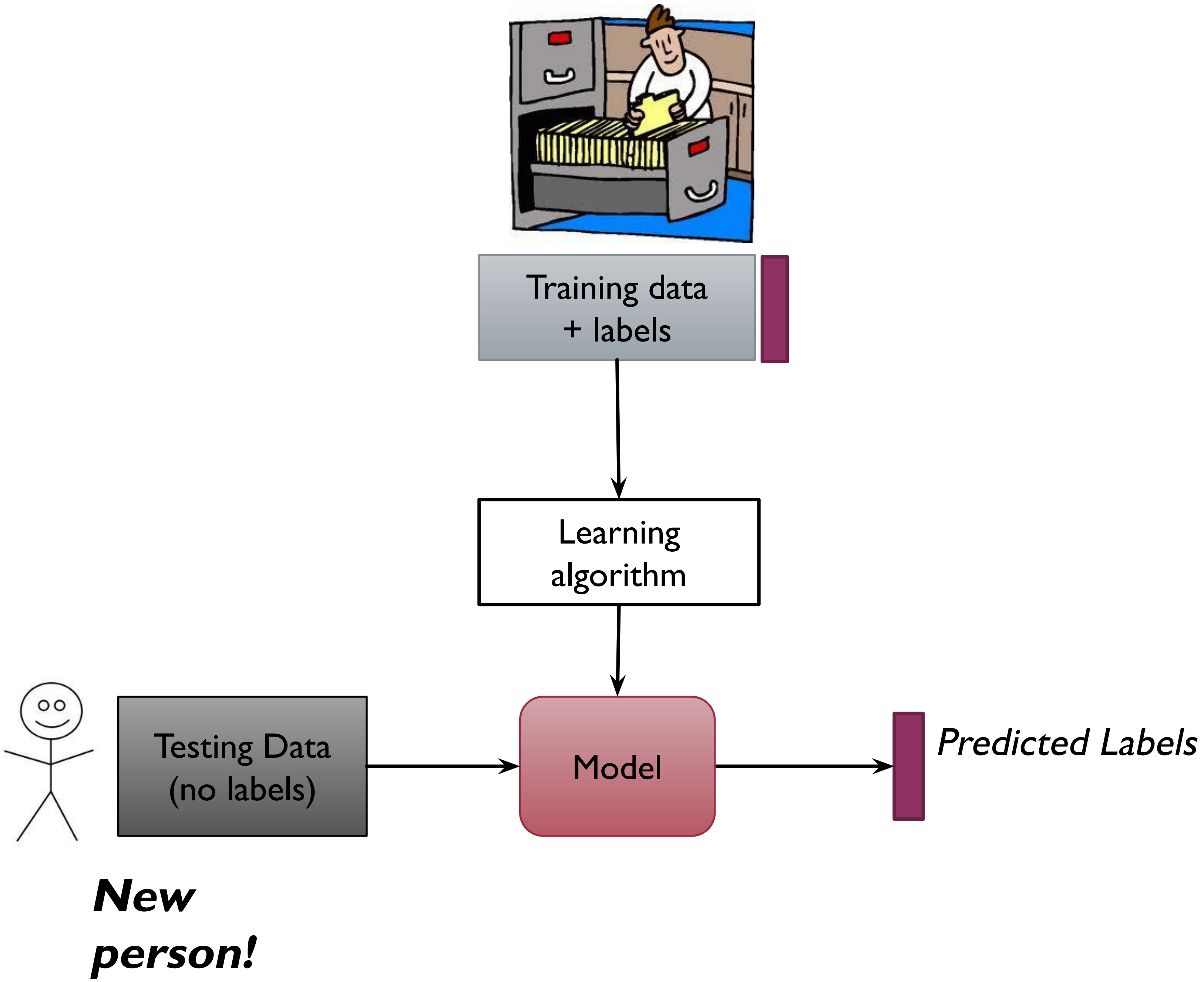
“Examples”

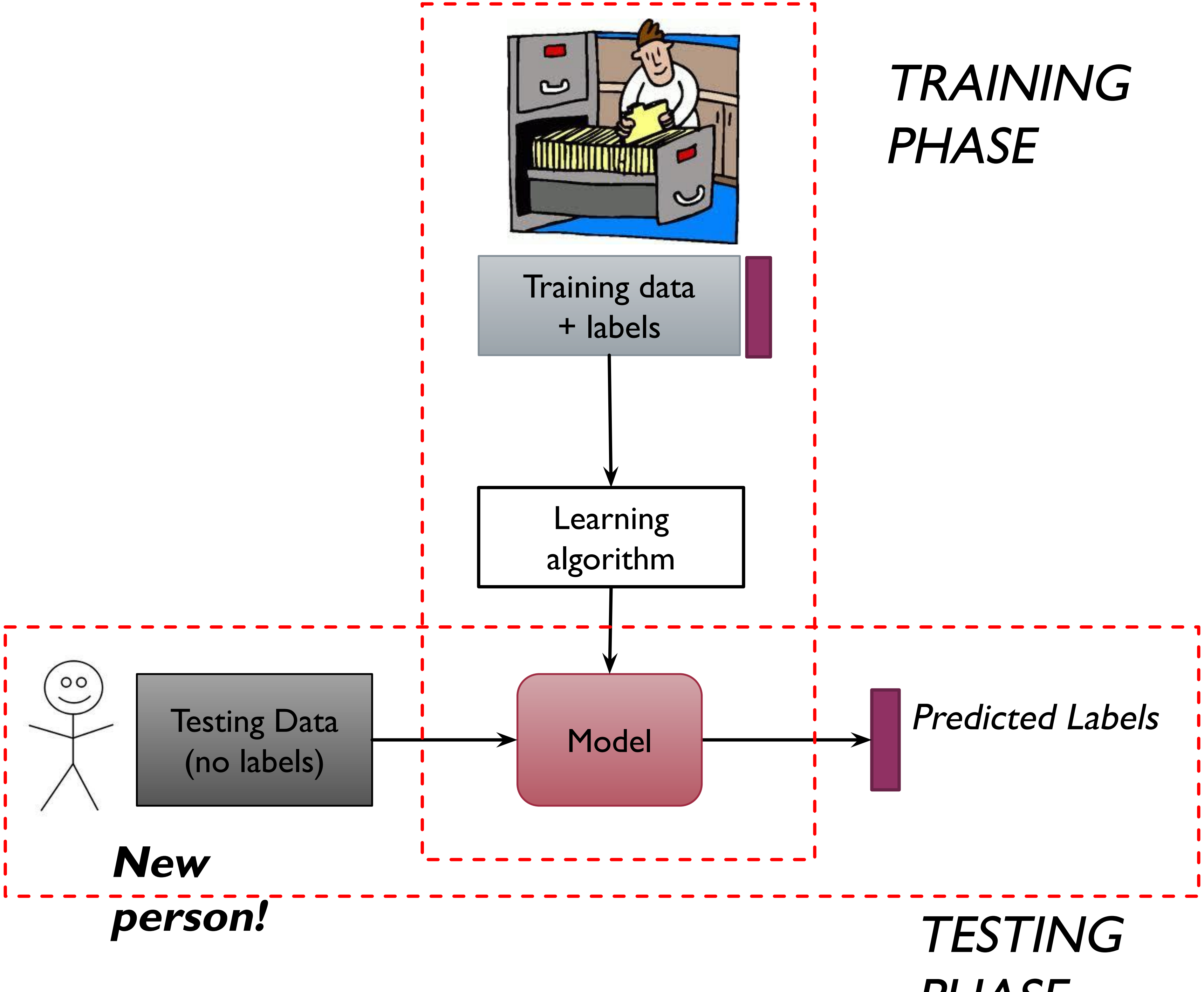
height	weight	BP	enzyme	Health?
70	64	3	1	1
23	86	5	0	1
56	49	5	1	0
50	88	3	0	0
12	50	1	0	1
56	66	2	1	0
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
56	1	5	0	0

“Features”

Class, or “label”

Historical data in health records for example.


Training data
+ labels
Learning
algorithm
Testing Data
(no labels)
Model
Predicted Labels
New
person!

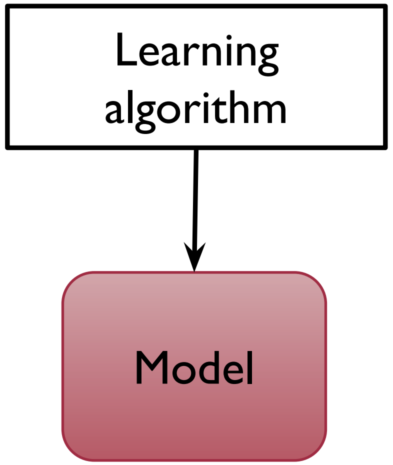


TRAINING PHASE
Training data + labels
Learning algorithm
Testing Data (no labels)
Model
Predicted Labels
New person!
TESTING

ML algorithms make mistakes

Predicting health.
Quite a hard problem even for trained professional!

Next… Need to QUANTIFY performance of our algorithms.



TAXONOMY OF MACHINE LEARNING (A SIMPLISTIC VIEW BASED ON TASKS)

gender	age	label
M	48	sick
M	67	sick
F	53	healthy
M	49	sick
F	32	healthy
M	34	healthy
M	21	healthy

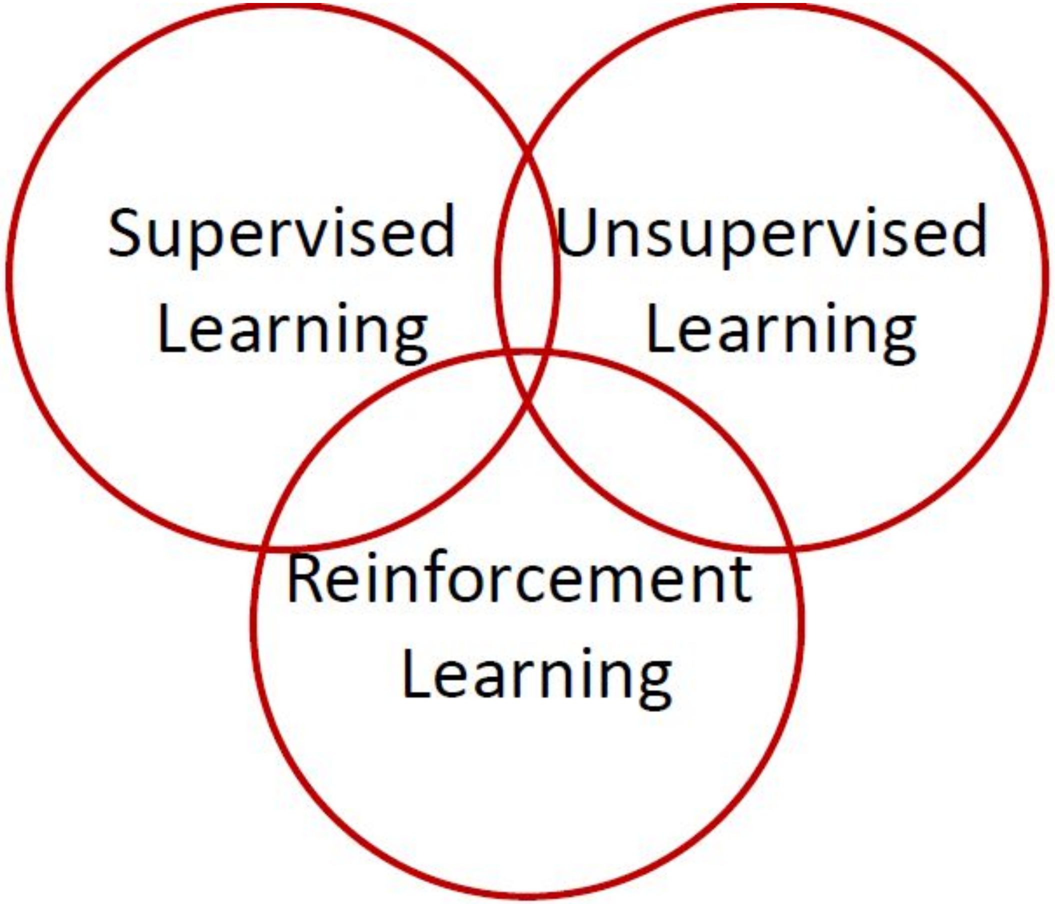
Supervised Learning

Unsupervised Learning

Reinforcement Learning

gender	age
M	48
M	67
F	53
M	49
F	34
M	21

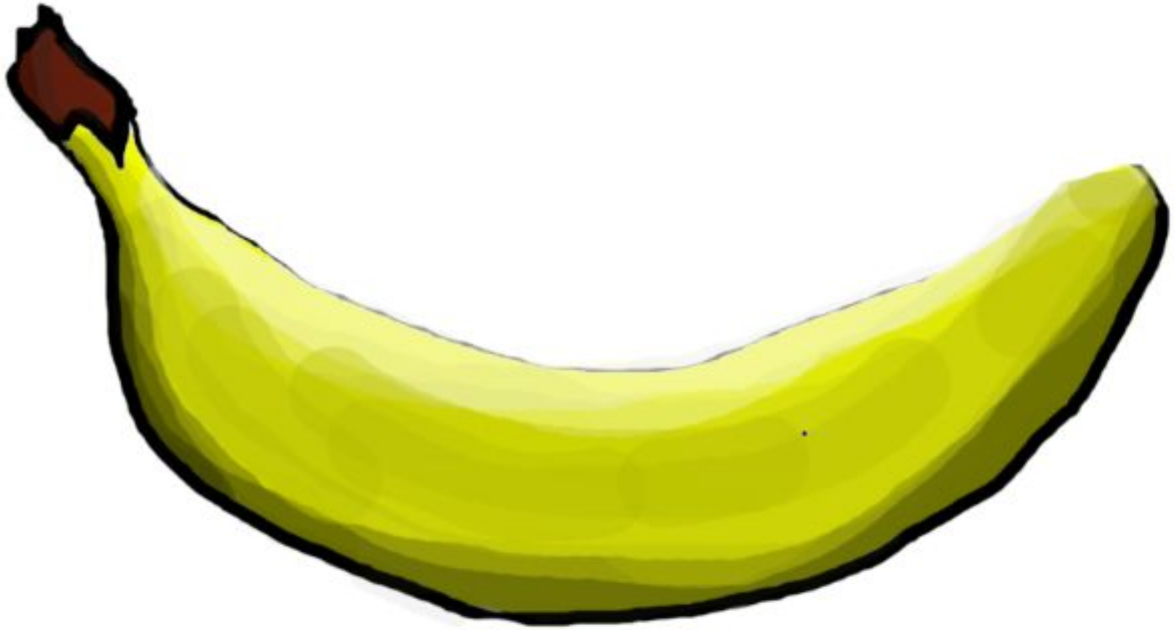
TAXONOMY OF MACHINE LEARNING (A SIMPLISTIC VIEW BASED ON TASKS)



Semi-supervised learning

can also be viewed as tools/methods

SUPERVISED LEARNING ALGORITHMS



EXAMPLE OF SUPERVISED LEARNING ALGORITHMS:

- Linear Regression
- Logistic Regression
- Nearest Neighbor
- Gaussian Naive Bayes
- Decision Trees
- Support Vector Machine (SVM)
- Random Forest

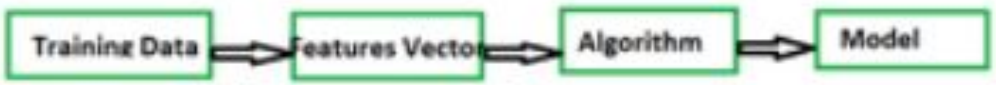
SUPERVISED LEARNING ALGORITHMS

Advantages:

- Supervised learning allows collecting data and produces data output from previous experiences.
- Helps to optimize performance criteria with the help of experience.
- Supervised machine learning helps to solve various types of real-world computation problems.

Disadvantages:

- Classifying big data can be challenging.
- Training for supervised learning needs a lot of computation time. So, it requires a lot of time.



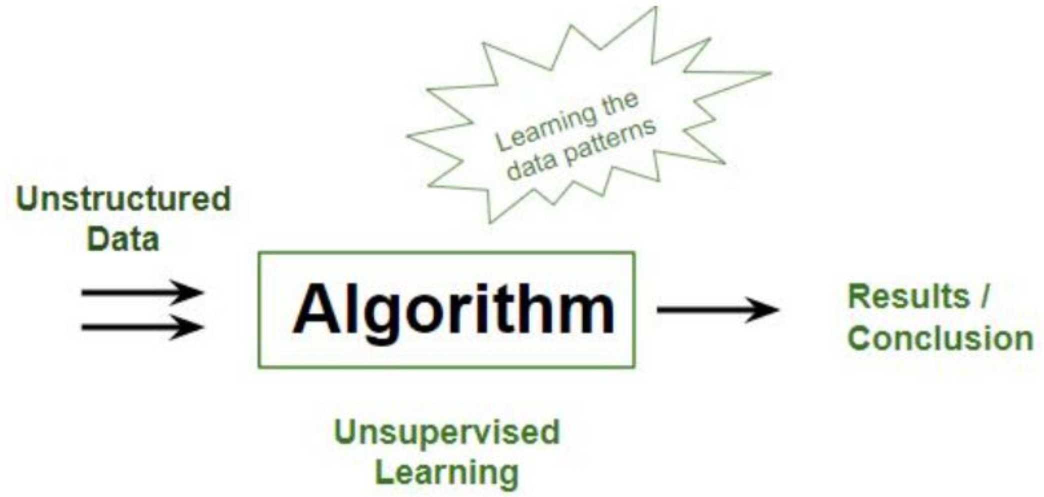
Steps

UNSUPERVISED LEARNING ALGORITHMS

- **Unsupervised learning algorithms** (unsupervised algorithms) are another type of algorithms. In unsupervised learning algorithms, only objects are known, and there are no answers. Although there are many successful applications of these methods, they tend to be more difficult to interpret and evaluate.
- **Examples of machine learning tasks without a teacher:**
- Identifying topics in a set of posts If you have a large collection of text data, you can aggregate them and find common topics. You have no preliminary information about what topics are covered there and how many of them. So there are no known answers.

EXAMPLES OF MACHINE LEARNING TASKS WITHOUT A TEACHER:

EXAMPLES OF MACHINE LEARNING TASKS WITHOUT A TEACHER:



CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3
10	Female	30	19	72
11	Male	67	19	14
12	Female	35	19	99
13	Female	58	20	15
14	Female	24	20	77
15	Male	37	20	13
16	Male	22	20	79
17	Female	35	21	35

Figure A

TYPES OF UNSUPERVISED LEARNING:

TYPES OF UNSUPERVISED LEARNING:

Clustering
Exclusive (partitioning)
Agglomerative
Overlapping
Probabilistic

Clustering Types:
K-means clustering (*DBSCAN, BIRCH*)
Hierarchical clustering
Principal Component Analysis
Singular Value Decomposition
Independent Component Analysis

MACHINE LEARNING TASKS WITHOUT A TEACHER:

- When solving machine learning tasks with and without a teacher, it is important to present your input data in a format that is understandable to a computer.
- Often the data is presented in the form of a table. Every data point you want to explore (every email, every customer, every transaction) is a row, and every property that describes that data point (say, customer age, amount, or transaction location) is a column. You can describe users by age, gender, account creation date and frequency of purchases in your online store. You can describe the image of the tumor using grayscale for each pixel or using the size, shape and color of the tumor.

DISCUSS EXAMPLES

- In machine learning, each object or row is called a sample or a data point, and the columns-properties that describe these examples are called characteristics or features.
- Later we will focus in more detail on the topic of data preparation, which is called feature extraction or feature engineering. However, you should keep in mind that no machine learning algorithm will be able to make a prediction based on data that does not contain any useful information.
- For example, if the only sign of a patient is his last name, the algorithm will not be able to predict his gender. This information is simply not in the data. If you add one more sign – the name of the patient, then things will already be better, because often, knowing the name of a person, you can judge his gender.

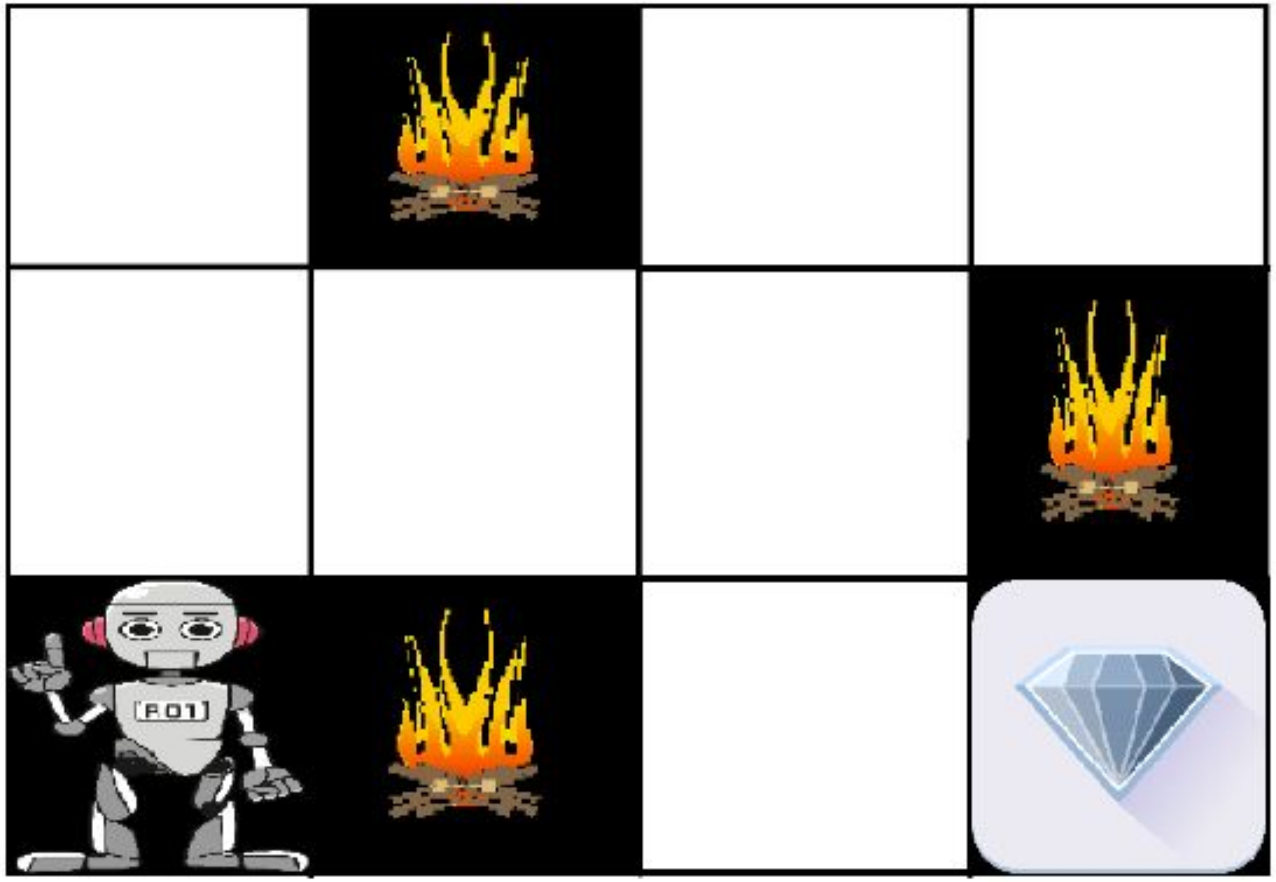
SEMI-SUPERVISED LEARNING:

REINFORCEMENT

Supervised vs. Unsupervised Machine Learning

Parameters	Supervised machine learning	Unsupervised machine learning
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data that is not labeled
Computational Complexity	Simpler method	Computationally complex
Accuracy	Highly accurate	Less accurate
No. of classes	No. of classes is known	No. of classes is not known
Data Analysis	Uses offline analysis	Uses real-time analysis of data
Algorithms used	Linear and Logistics regression, Random forest, Support Vector Machine, Neural Network, etc.	K-Means clustering, Hierarchical clustering, Apriori algorithm, etc.

REINFORCEMENT LEARNING ALGORITHMS



REINFORCEMENT LEARNING ALGORITHMS

Main points in Reinforcement learning –

- Input: The input should be an initial state from which the model will start
- Output: There are many possible outputs as there are a variety of solutions to a particular problem
- Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.
- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

DIFFERENCE BETWEEN REINFORCEMENT LEARNING AND SUPERVISED LEARNING:

Reinforcement learning	Supervised learning
Reinforcement learning is all about making decisions sequentially. In simple words, we can say that the output depends on the state of the current input and the next input depends on the output of the previous input	In Supervised learning, the decision is made on the initial input or the input given at the start
In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions	In supervised learning the decisions are independent of each other so labels are given to each decision.
Example: Chess game	Example: Object recognition

REINFORCEMENT LEARNING ALGORITHMS

Types of Reinforcement: There are two types of Reinforcement:

1. **Positive –**
   Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior. Advantages of reinforcement learning are:
   - Maximizes Performance
   - Sustain Change for a long period of time
   - Too much Reinforcement can lead to an overload of states which can diminish the results
2. **Negative –**
   Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided. Advantages of reinforcement learning:
   - Increases Behavior
   - Provide defiance to a minimum standard of performance
   - It Only provides enough to meet up the minimum behavior

CATEGORIZING BASED ON REQUIRED OUTPUT

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:

1. **Classification:** When inputs are divided into two or more classes, the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".

2. **Regression:** Which is also a supervised problem, A case when the outputs are continuous rather than discrete.

3. **Clustering:** When a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

DISCUSS EXAMPLES OF REINFORCEMENT LEARNING

Various Practical applications of Reinforcement Learning –

- RL can be used in robotics for industrial automation.
- RL can be used in machine learning and data processing
- RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.

RL can be used in large environments in the following situations:

1. A model of the environment is known, but an analytic solution is not available;
2. Only a simulation model of the environment is given (the subject of simulation-based optimization)
3. The only way to collect information about the environment is to interact with it.

SCIENCE WITH PYTHON

- The amount of digital data that exists is growing at a rapid rate, doubling every two years, and changing the way we live. It is estimated that by 2020, about 1.7MB of new data will be created every second for every human being on the planet. This means we need to have the technical tools, algorithms, and models to clean, process, and understand the available data in its different forms for decision-making purposes.
- *Data science* is the field that comprises everything related to cleaning, preparing, and analyzing unstructured, semistructured, and structured data. This field of science uses a combination of statistics, mathematics, programming, problem-solving, and data capture to extract insights and information from data.

THE STAGES OF DATA SCIENCE

- Figure 1-1 shows different stages in the field of data science. Data scientists
- use programming tools such as Python, R, SAS, Java, Perl, and C/C++
- to extract knowledge from prepared data. To extract this information,
- they employ various fit-to-purpose models based on machine leaning
- algorithms, statistics, and mathematical methods.

Figure 1-1. *Data science project stages*

WHY PYTHON?

- Python is a dynamic and general-purpose programming language that is used in various fields. Python is used for everything from throwaway scripts to large, scalable web servers that provide uninterrupted service 24/7.
- It is used for web programming, and application testing. It is used by scientists writing
- applications for the world's fastest supercomputers and by children first learning to program. It was initially developed in the early 1990s by Guido van Rossum and is now controlled by the not-for-profit Python Software Foundation, sponsored by Microsoft, Google, and others.
- The first-ever version of Python was introduced in 1991. Python is now at version 3.*x*, which was released in February 2011 after a long period of testing. Many of its major features have also been backported to the backward-compatible Python 2.6, 2.7, and 3.6. GUI and database programming, client- and server-side

BASIC FEATURES OF PYTHON PYTHON PROVIDES NUMEROUS FEATURES; THE FOLLOWING ARE SOME OF THESE

- ***Easy to learn and use*:** Python uses an elegant syntax, making the programs easy to read. It is developer-friendly and is a high-level programming language.
- ***Expressive***: The Python language is expressive, which means it is more understandable and readable than other languages.
- ***Interpreted***: Python is an interpreted language. In other words, the interpreter executes the code line by line. This makes debugging easy and thus suitable for beginners.
- ***Cross-platform*:** Python can run equally well on different platforms such as Windows, Linux, Unix, Macintosh, and so on. So, Python is a portable language.
- ***Free and open source*:** The Python language is freely available at www.python.org. The source code is also available.

BASIC FEATURES OF PYTHON

- *Object-oriented*: Python is an object-oriented language with concepts of classes and objects.
- • *Extensible*: It is easily extended by adding new modules implemented in a compiled language such as C or C++, which can be used to compile the code.
- • *Large standard library*: It comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, and reading and modifying files.
- • *GUI programming support*: Graphical user interfaces can be developed using Python.
- • *Integrated*: It can be easily integrated with languages such as C, C++, Java, and more.

PORTABLE PYTHON EDITORS (NO INSTALLATION REQUIRED)

- These editors require no installation:
- *Azure Jupyter Notebooks*: The open source Jupyter Notebooks was developed by Microsoft as an analytic playground for analytics and machine learning.
- ***Python(x,y)***: Python(x,y) is a free scientific and engineering development application for numerical computations, data analysis, and data visualization based on the Python programming language, **Qt graphical user interfaces**, and Spyder interactive scientific development environment.
- ***WinPython***: This is a free Python distribution for the Windows platform; it includes prebuilt packages for ScientificPython.
- ***Anaconda***: This is a completely free enterprise ready Python distribution for large-scale data processing, predictive analytics, and scientific computing.

TABULAR DATA AND DATA FORMATS

- Data is available in different forms. It can be unstructured data, semistructured data, or structured data.
- Python provides different structures to maintain data and to manipulate it such as variables, lists, dictionaries, tuples, series, panels, and data frames. Tabular data can be easily represented in Python using lists of tuples representing the records of the data set in a data frame structure.
- Though easy to create, these kinds of representations typically do not enable important tabular data manipulations, such as efficient column selection, matrix mathematics, or spreadsheet-style operations. Tabular is a package of Python modules for working with tabular data. Its main object is the tabarray class, which is a data structure for holding and manipulating tabular data. You can put data into a tabarray object for more flexible and powerful data processing.
- The Pandas library also provides rich data structures and functions designed to make working with structured data fast, easy, and expressive. In addition, it provides a powerful and productive data analysis environment.
- A Pandas data frame can be created using the following constructor:

pandas.DataFrame(data, index, columns, dtype, copy)

PANDAS DATA FRAME

A Pandas data frame can be created using various input forms such as the following:

- List
- Dictionary
- Series
- Numpy ndarrays

Another data frame

PYTHON PANDAS DATA SCIENCE LIBRARY

- Pandas is an open source Python library providing high-performance data manipulation and analysis tools via its powerful data structures. The name Pandas is derived from “panel data,” an econometrics term from multidimensional data. The following are the key features of the Pandas library:
- Provides a mechanism to load data objects from different formats
- Creates efficient data frame objects with default and customized indexing
- Reshapes and pivots date sets
- Provides efficient mechanisms to handle missing data
- Merges, groups by, aggregates, and transforms data
- Manipulates large data sets by implementing various functionalities such as slicing, indexing, subsetting, deletion, and insertion
- Provides efficient time series functionality

TECHNICAL REQUIREMENTS

- We will use various Python packages, such as NumPy, SciPy, scikit-learn, and Matplotlib, during the course of this book to build various things. If you use Windows, it is recommended that you use a SciPy-stack-compatible version of Python. You can check the list of compatible versions at http:/ / www. scipy. org/ install. html. These distributions come with all the necessary packages already installed. If you use MacOS X or Ubuntu, installing these packages is fairly straightforward. Here are some useful links for installation and documentation:
- **NumPy: https:/ / www. numpy. org/ devdocs/ user/ install. html.**
- **SciPy: http:/ / www. scipy. org/ install. html.**
- **Scikit-learn: https:/ / scikit- learn. org/ stable/ install. html.**
- **Matplotlib: https:/ / matplotlib. org/ users/ installing. html.**

A PANDAS SERIES

```
In [1]: #Create series from array using pandas and numpy
        import pandas as pd
        import numpy as np
        data = np.array([90,75,50,66])
        s = pd.Series(data,index=['A','B','C','D'])
        print (s)

A    90
B    75
C    50
D    66
dtype: int32
```

```
In [5]: #Create series from dictionary using pandas
        import pandas as pd
        import numpy as np
        data = {'Ahmed' : 92, 'Ali' : 55, 'Omar' : 83}
        s = pd.Series(data,index=['Ali','Ahmed','Omar'])
        print (s)

Ali      55
Ahmed    92
Omar     83
```

A PANDAS DATA FRAME

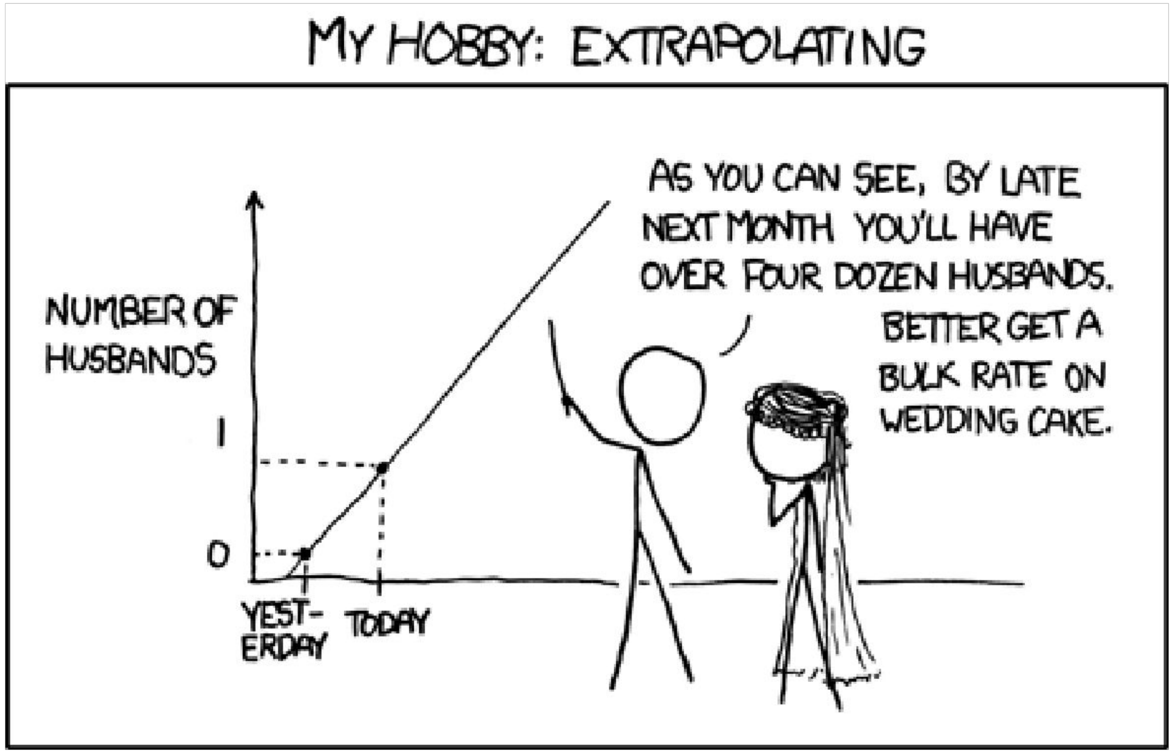
- A *data frame* is a two-dimensional data structure. In other words, data is aligned in a tabular fashion in rows and columns. In the following table, you have two columns and three rows of data. Listing 2 shows how to create a data frame using the Pandas library.

```
In [10]: # Creating a Data Frame Using the Pandas Library
         import pandas as pd
         data = [['Ahmed',35],['Ali',17],['Omar',25]]
         DataFrame1 = pd.DataFrame(data,columns=['Name','Age'])
         print (DataFrame1)

    Name  Age
0  Ahmed   35
1    Ali   17
2   Omar   25

In [11]: DataFrame1[1:]

Out[11]:
```

	Name	Age
1	Ali	17
2	Omar	25

Linear Model

Linear Models



A Problem to Solve with Machine Learning

Distinguish rugby players from ballet dancers.

You are provided with a few examples.

Almaty rugby club (16).

Astana ballet troupe (10).

Task

Generate a program which will correctly classify ANY player/dancer in the <u>world</u>.

Hint

We shouldn't "fine-tune" our system too much so it only works on the local clubs.

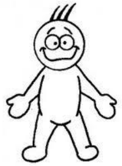
Taking measurements….

We have to process the people with a computer, so it needs to be in a computer-readable form.

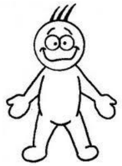
What are the distinguishing characteristics?

1. *Height*
2. *Weight*
3. *Shoe size*
4. *Gender*

Terminology



"Examples"

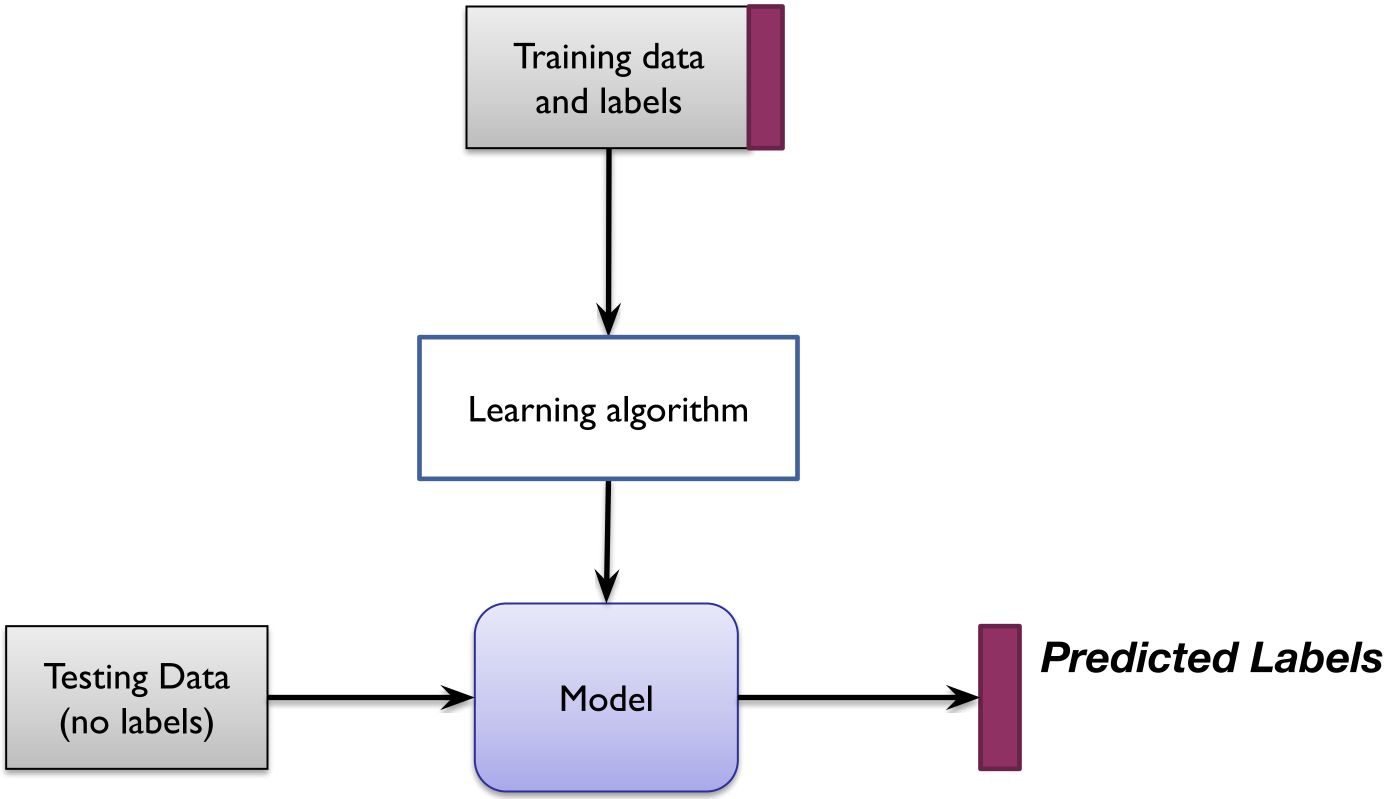


id	height	weight	shoe size	sex	Ballet?
1	70	64	3	1	1
2	23	86	5	0	1
3	56	49	5	1	0
4	50	88	3	0	0
5	12	50	I	0	1
6	56	66	2	1	0
…	…	…	…	…	…
…	…	…	…	…	…
…	…	…	…	…	…
…	…	…	…	…	…
N	56	1	5	0	0

"Features"

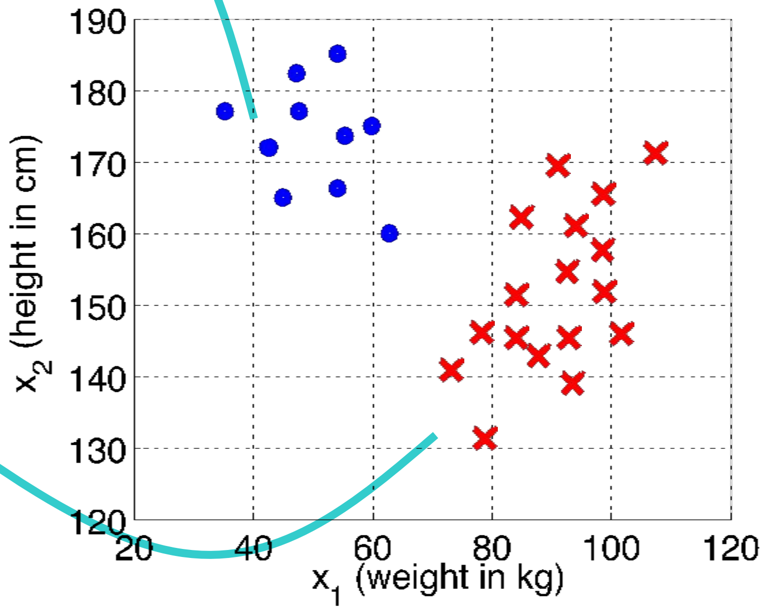
Class, or "label"

THE SUPERVISED LEARNING PIPELINE



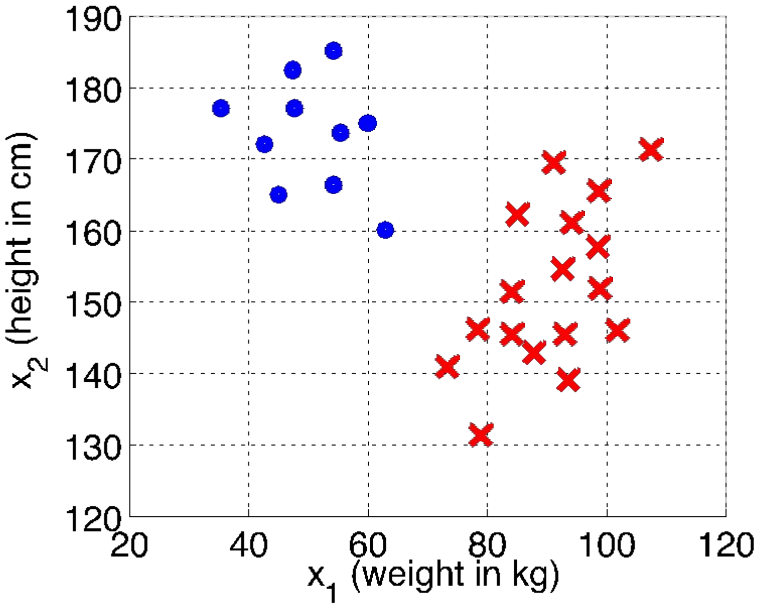
Taking measurements….

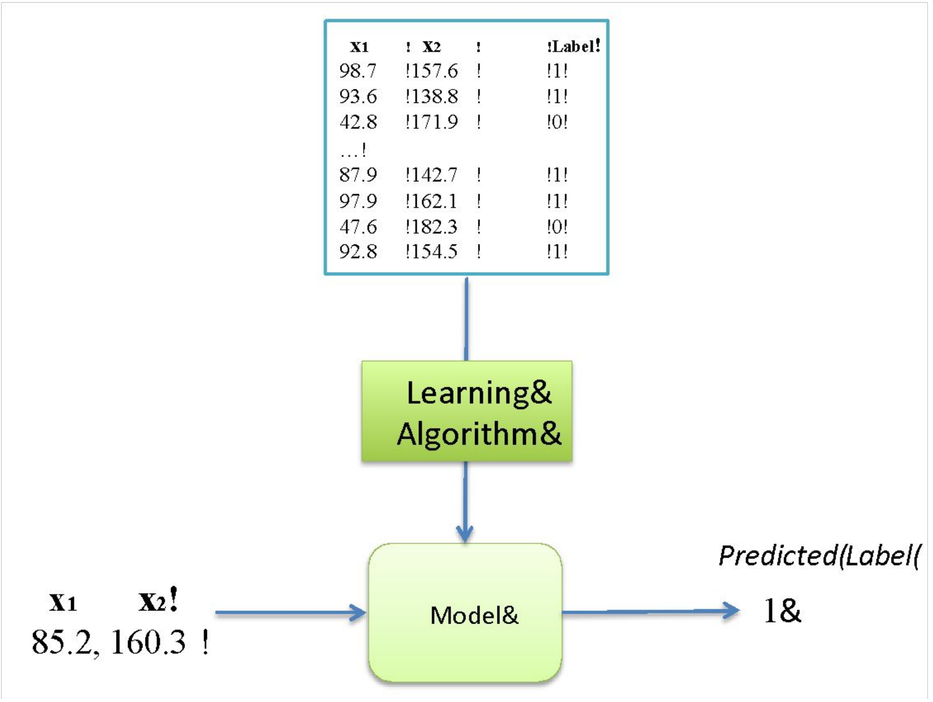
Person	Weight	Height
1	63kg	190cm
2	55kg	185cm
3	75kg	202cm
4	50kg	180cm
5	57kg	174cm
…	…	
16	85kg	150cm
17	93kg	145cm
18	75kg	130cm
19	99kg	163cm
20	100kg	171cm
…	…	

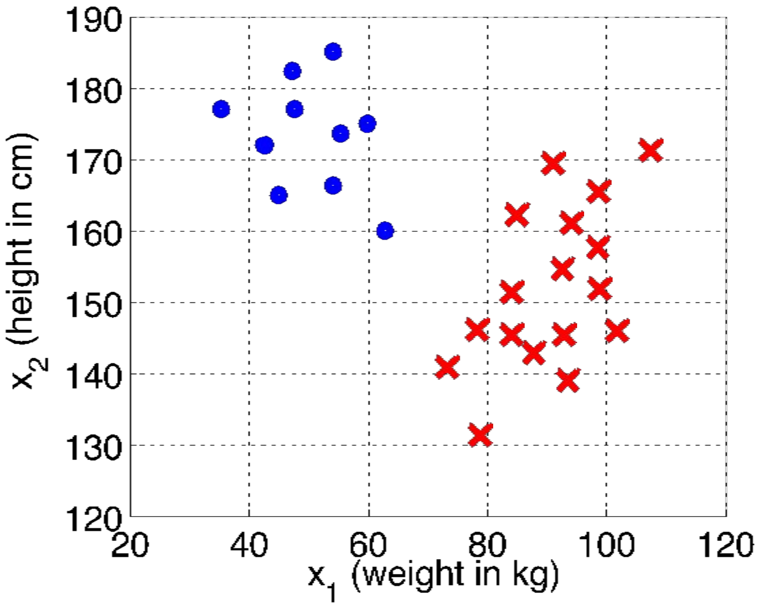


A Problem

x_1,	x_2,	y (label)
98.79,	157.59,	1
93.64,	138.79,	1
42.89,	171.89,	0
...		
...		
87.91,	142.65,	1
97.92,	162.12,	1
47.63,	182.26,	0
92.72,	154.50,	1



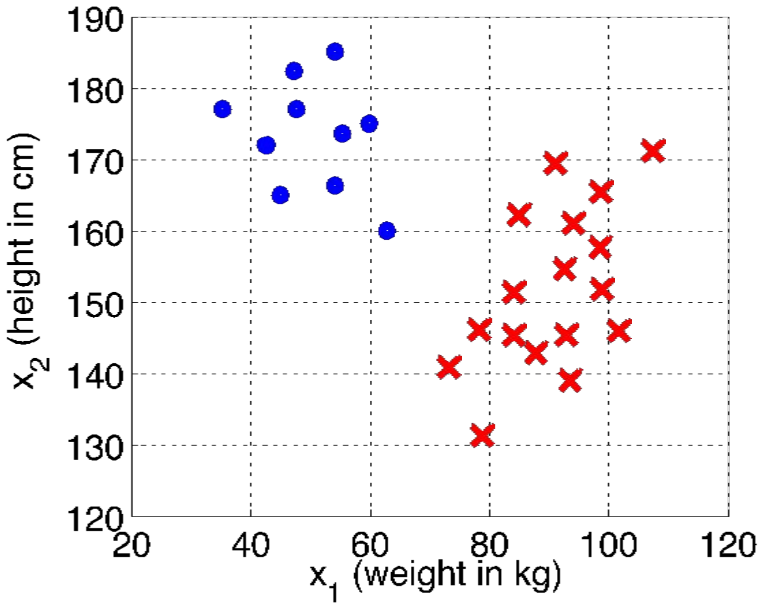


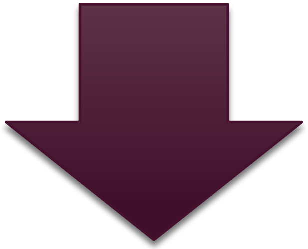


A simple computer program will solve this….

$$\text{if } x_1 > 70 \text{ then } \hat{y} = 1 \text{ else } \hat{y} = 0$$

$$\text{if } x_1 > 70 \text{ then } \hat{y} = 1 \text{ else } \hat{y} = 0$$



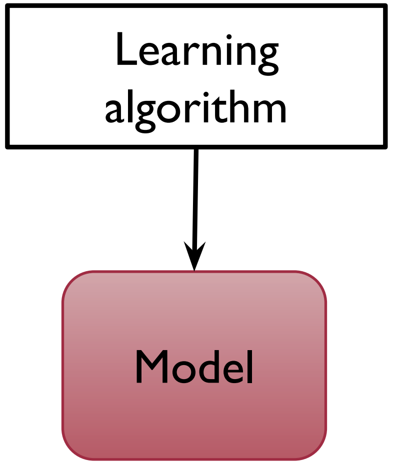


$$\text{if } f(x) > 0 \text{ then } \hat{y} = 1 \text{ else } \hat{y} = 0$$

where

...

$$f(x) = (x_1 - t), \text{ with } t = 70,$$



f(x) .... is a MODEL

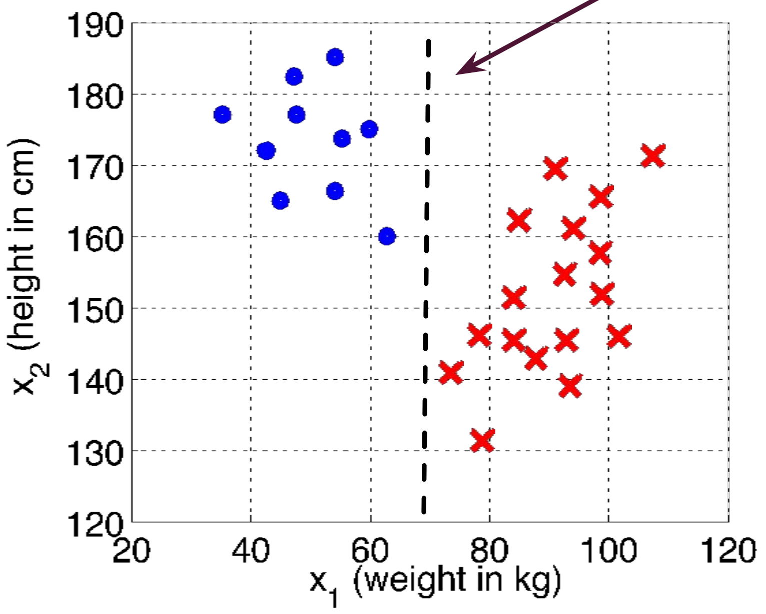
The "Decision Stump" is a linear model

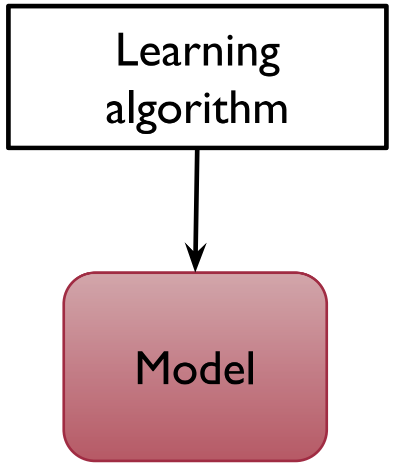
$$\text{if } f(x) > 0 \text{ then } \hat{y} = 1 \text{ else } \hat{y} = 0$$

where

$$f(x) = (x_1 - t)$$

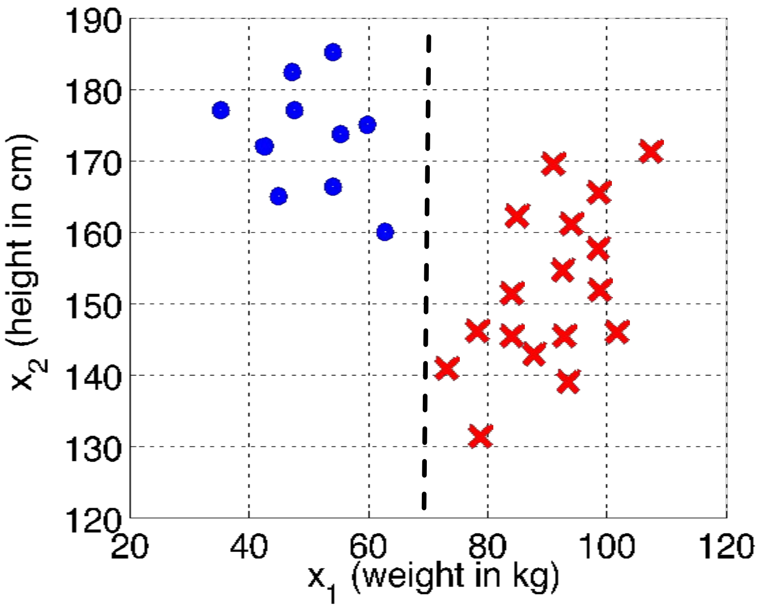
"*Decision Boundary*"

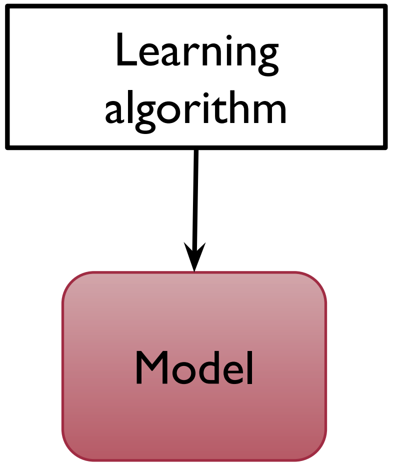


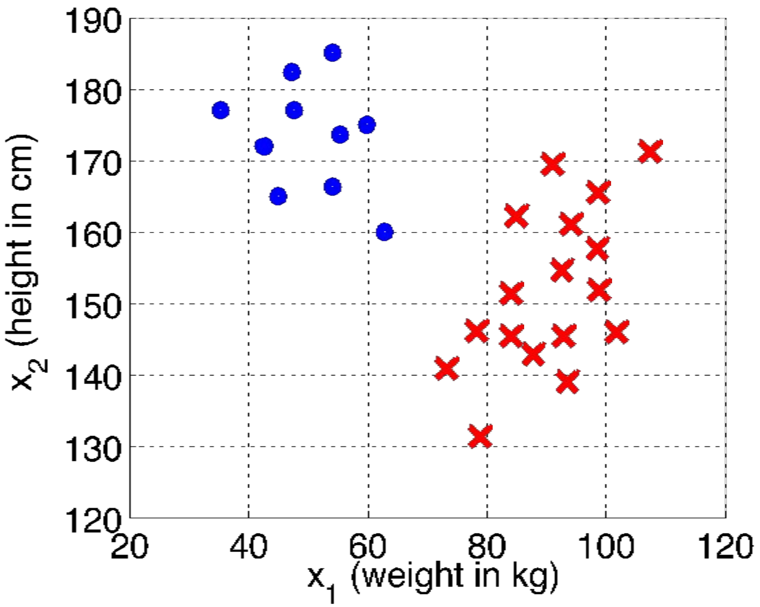


Learning Algorithm for Decision Stump: *Line search*

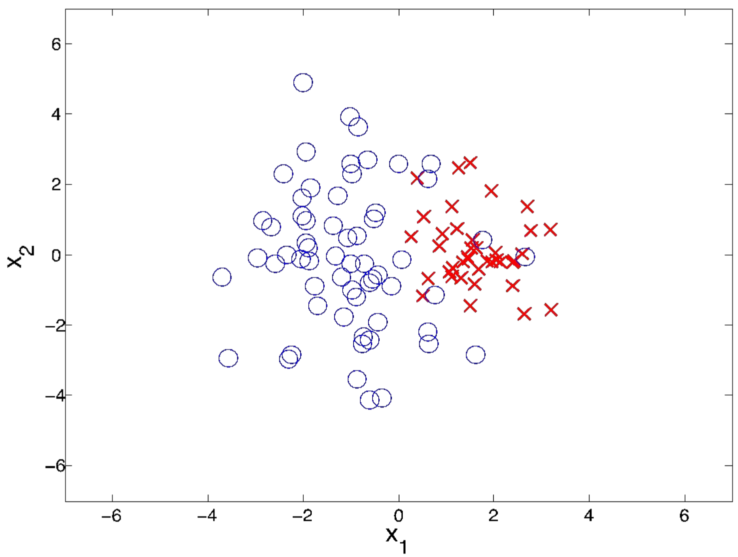
stepsize 1, *minErr* 99999
for t = min(x) *to* max(x) *by stepsize do*
  numErrs = *numberOfErrors*(t)
  if numErrs $\leq$ *minErr then*
    minErr *numErrs*
    t_{best} t
  end if
end for
return t_{best}







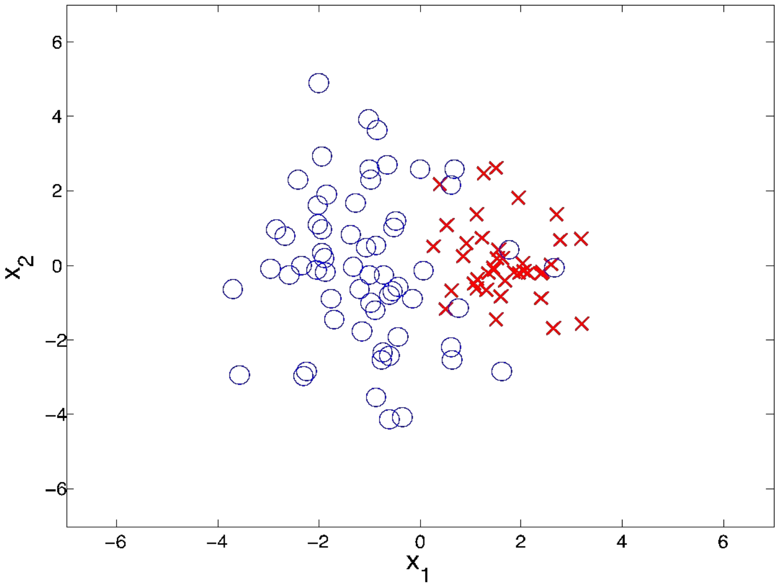
LINEARLY SEPARABLE



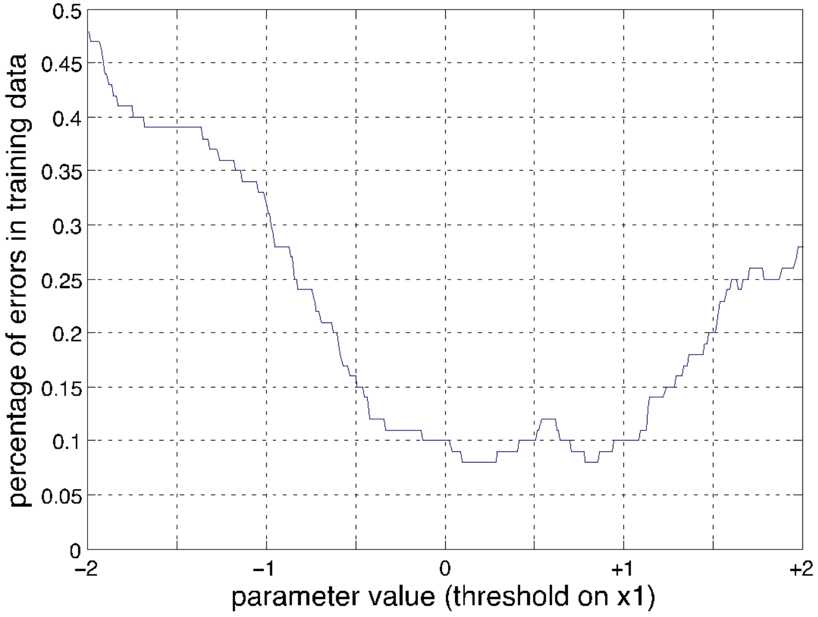
NON-LINEARLY SEPARABLE

Learning Algorithm for Decision Stump: *Line search*

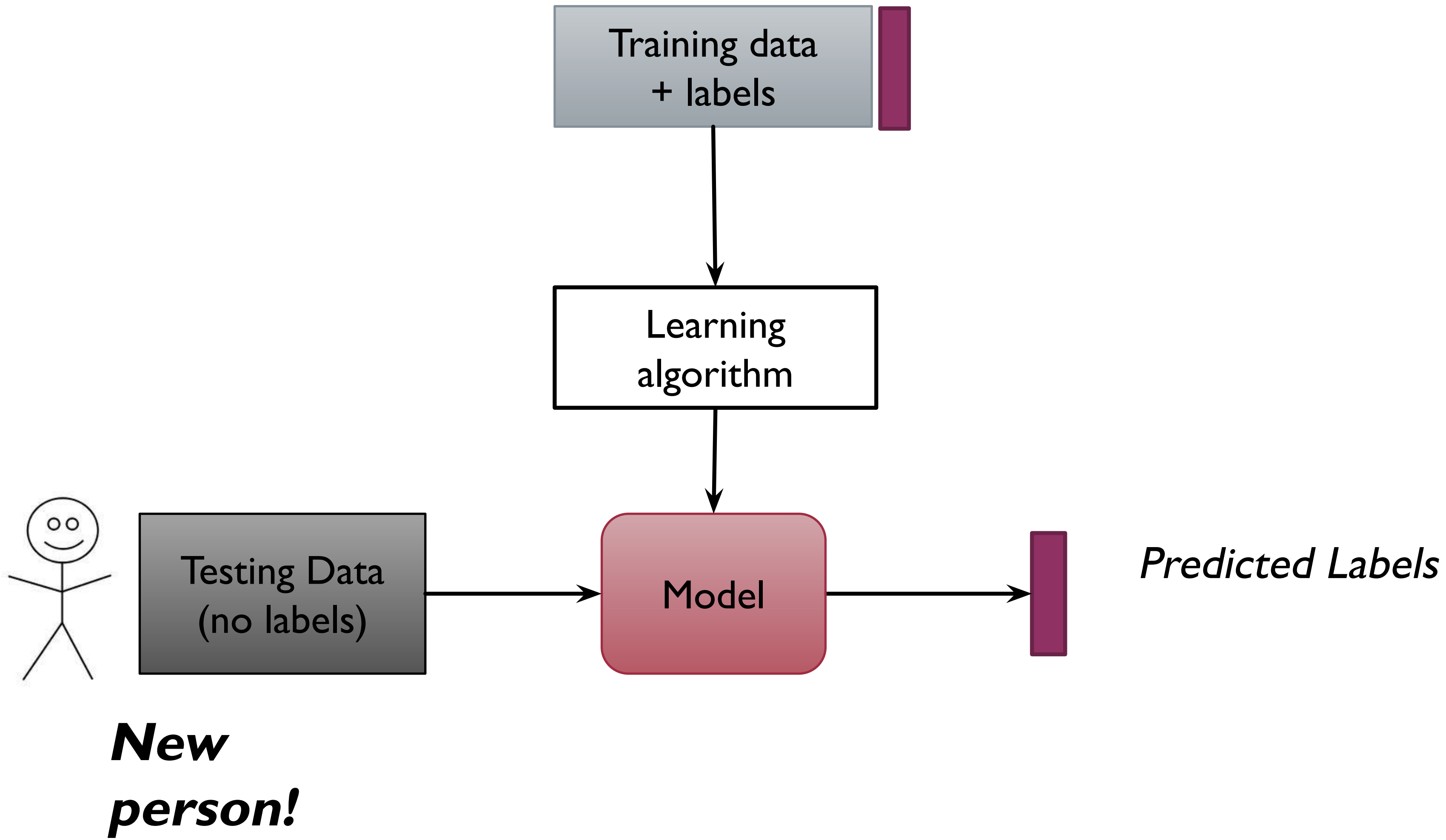
$stepsize \quad 1,\ minErr \quad 99999$
for $t = \min(x)$ *to* $\max(x)$ *by stepsize do*
  $numErrs = numberOfErrors(t)$
  if $numErrs \leq minErr$ *then*
    $minErr \quad numErrs$
    $t_{best} \quad t$
  end if
end for
return t_{best}

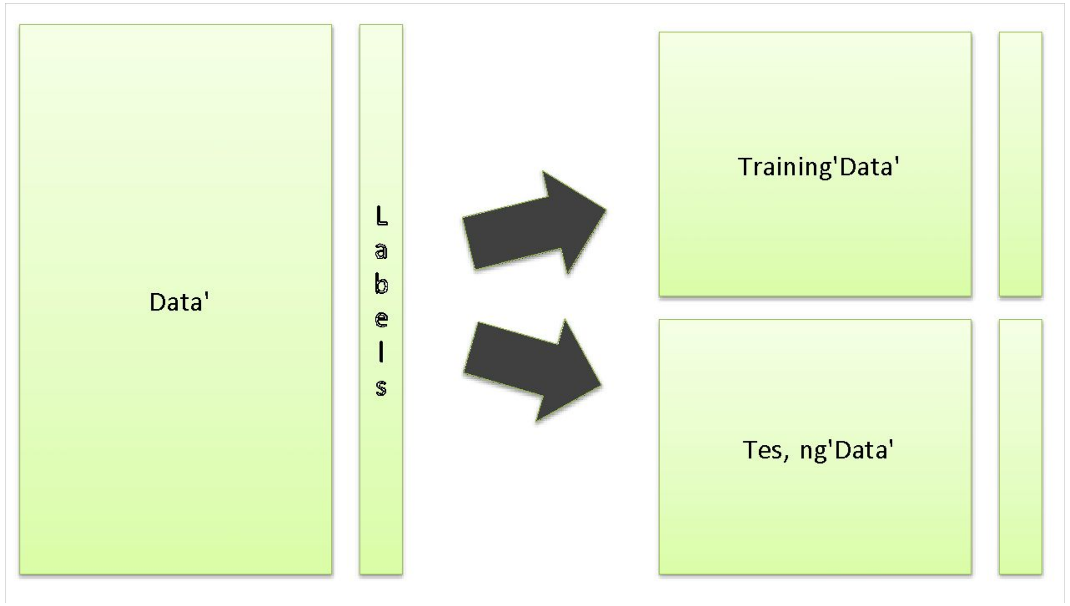


"Error landscape"



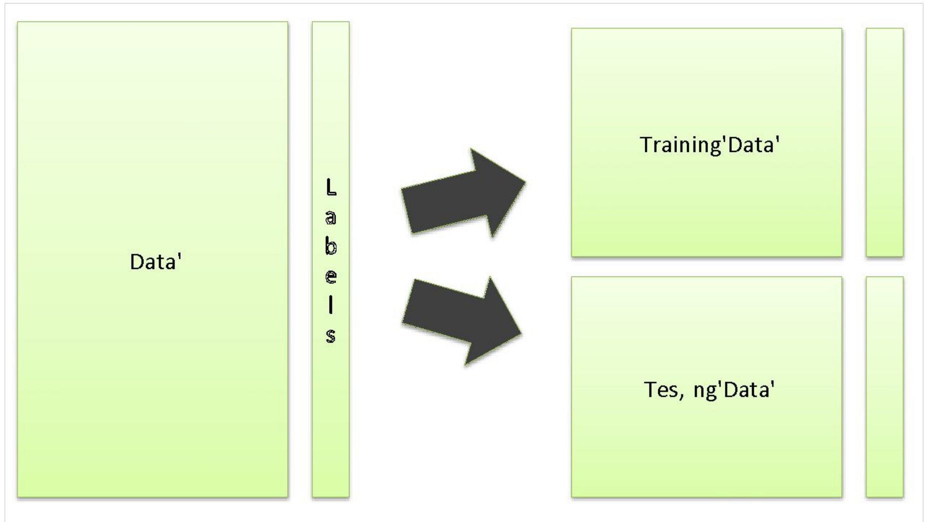
Training = driving lessons
Testing = driving test





LESSONS
....

THEN THE TEST
!

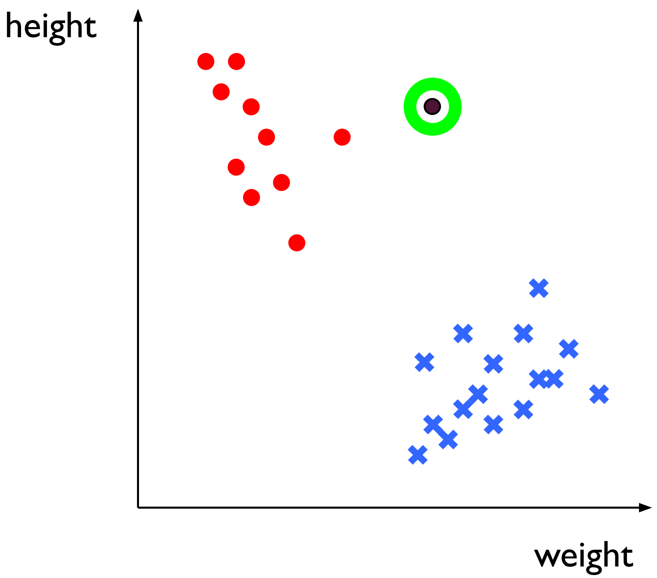


Evaluating a Model

1. Split the data randomly in half, into training and testing sets.
2. Train a model on the *training set*.
3. Test this model on the *testing set* and record the error rate.
4. Repeat this procedure many times, and calculate the average error rate.

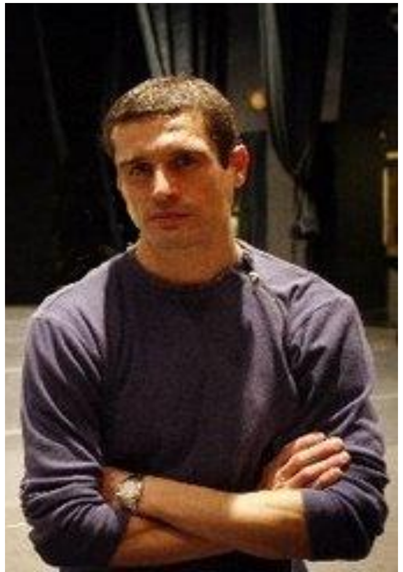
The Nearest Neighbour Classifier

The Nearest Neighbour Rule



Person	Weight	Height
1	63kg	190cm
2	55kg	185cm
3	75kg	202cm
4	50kg	180cm
5	57kg	174cm
…	…	
16	85kg	150cm
17	93kg	145cm
18	75kg	130cm
19	99kg	163cm
20	100kg	171cm
…	…	

"TRAINING" DATA

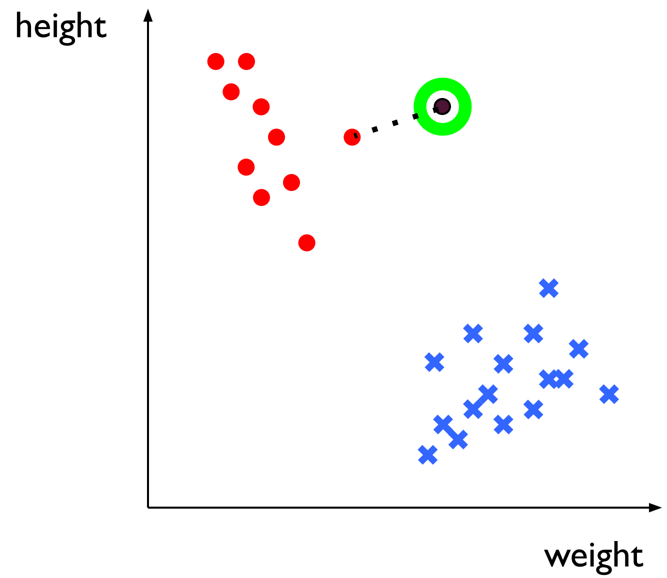


"TESTING" DATA

Who's this guy?
- player or dancer?

height = 180cm
weight = 78kg

The Nearest Neighbour Rule



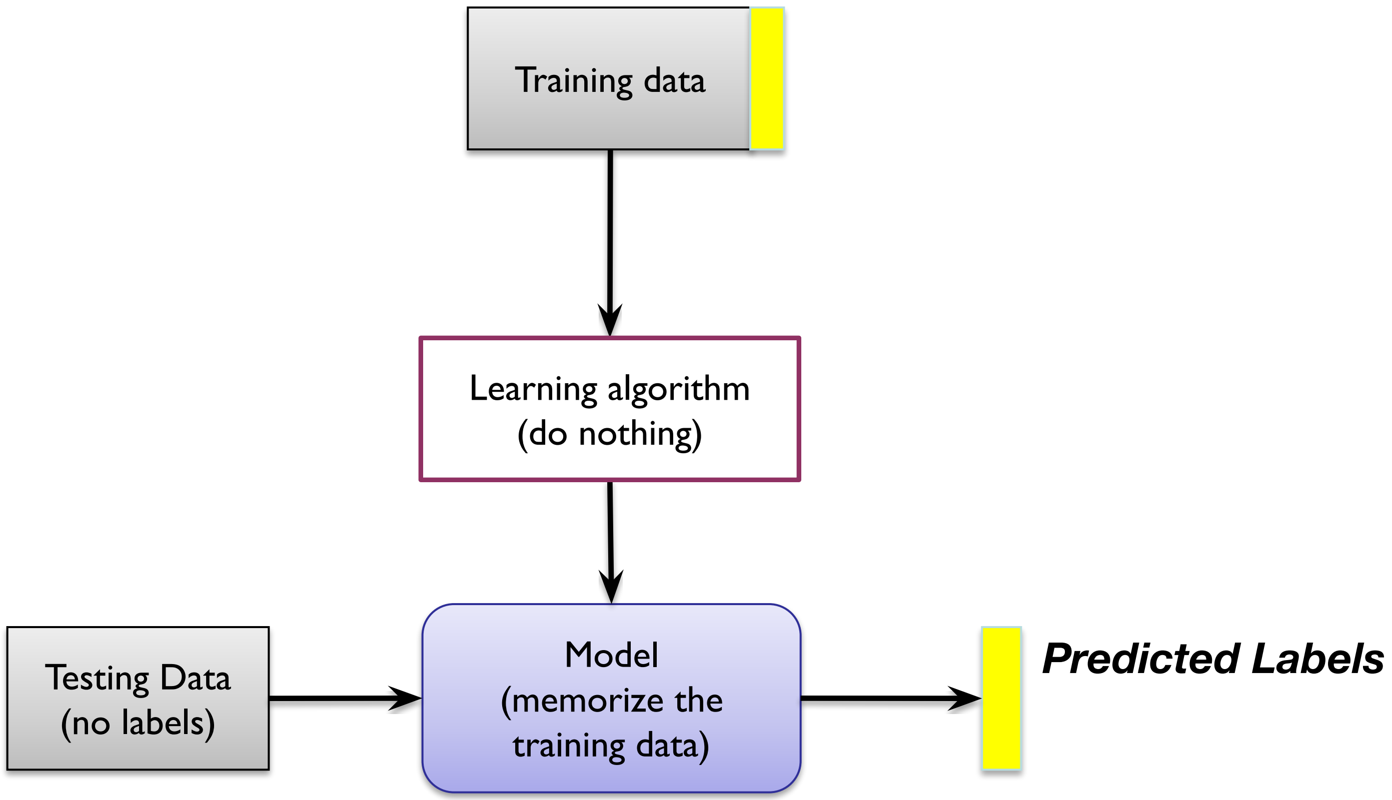
Person	Weight	Height
1	63kg	190cm
2	55kg	185cm
3	75kg	202cm
4	50kg	180cm
5	57kg	174cm
…	…	
16	85kg	150cm
17	93kg	145cm
18	75kg	130cm
19	99kg	163cm
20	100kg	171cm
…	…	

"TRAINING" DATA

height = 180cm
weight = 78kg

```
1.  Find nearest neighbour
2.  Assign the same class
```

Supervised Learning Pipeline for Nearest Neighbour

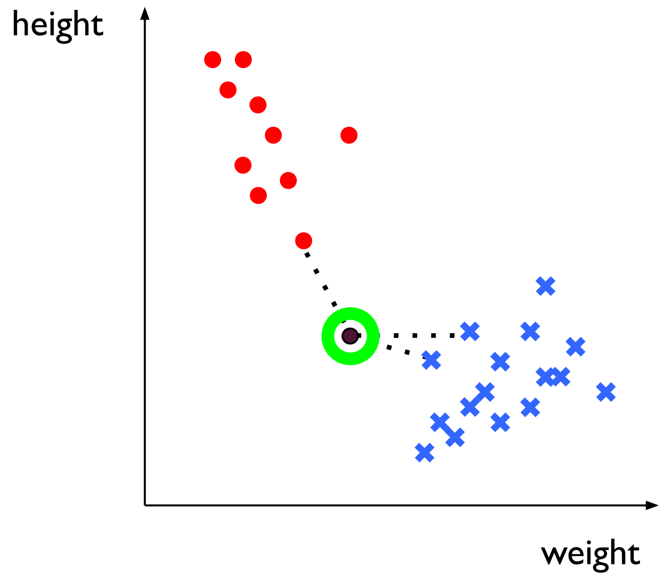


The K-Nearest Neighbour Classifier

```
Testing point x
For each training datapoint x'
   measure distance(x,x')
End
Sort distances
Select K nearest
Assign most common class
```

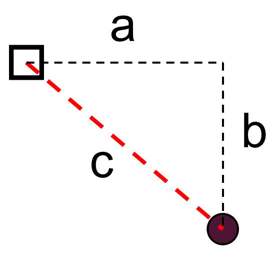
Person	Weight	Height
1	63kg	190cm
2	55kg	185cm
3	75kg	202cm
4	50kg	180cm
5	57kg	174cm
…	…	
16	85kg	150cm
17	93kg	145cm
18	75kg	130cm
19	99kg	163cm
20	100kg	171cm
…	…	

"TRAINING" DATA



Quick reminder: Pythagoras' theorem

```
. . .
   measure distance(x,x')
. . .
```

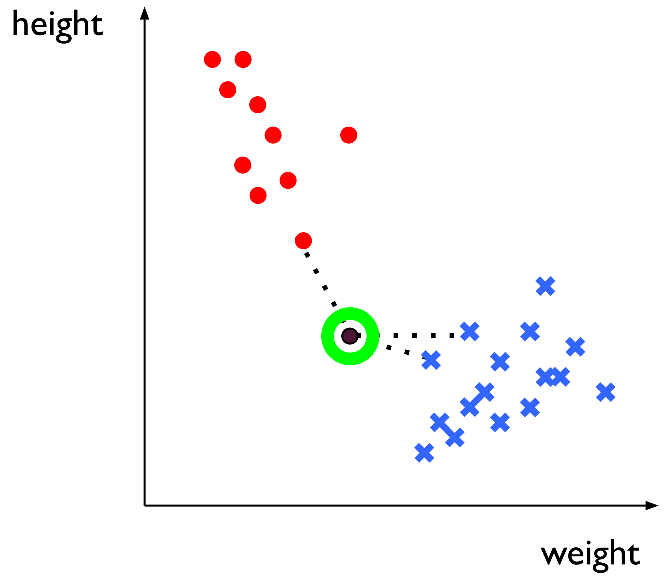


$$a^2 + b^2 = c^2$$

So.... $$c = \sqrt{a^2 + b^2}$$

a.k.a. "Euclidean" distance

$$\text{distance}(x, x') = \sqrt{\sum_i (x_i - x'_i)^2}$$



The K-Nearest Neighbour Classifier

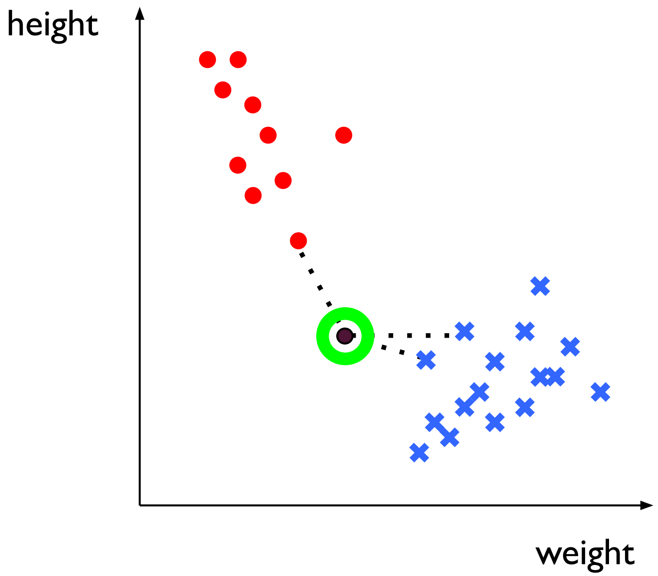
```
Testing point x
For each training datapoint x'
   measure distance(x,x')
End
Sort distances
Select K nearest
Assign most common class
```

Person	Weight	Height
1	63kg	190cm
2	55kg	185cm
3	75kg	202cm
4	50kg	180cm
5	57kg	174cm
…	…	
16	85kg	150cm
17	93kg	145cm
18	75kg	130cm
19	99kg	163cm
20	100kg	171cm
…	…	

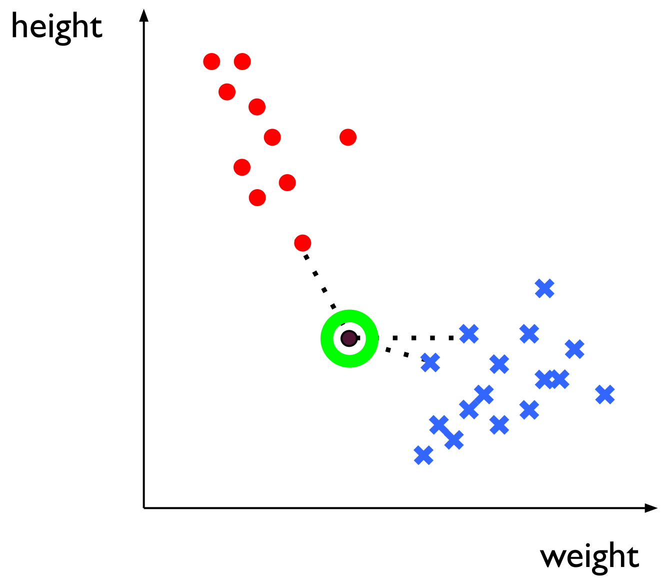
"TRAINING" DATA

Seems sensible.

But what are the disadvantages?



The K-Nearest Neighbour Classifier



Person	Weight	Height
1	63kg	190cm
2	55kg	185cm
3	75kg	202cm
4	50kg	180cm
5	57kg	174cm
…	…	
16	85kg	150cm
17	93kg	145cm
18	75kg	130cm
19	99kg	163cm
20	100kg	171cm
…	…	

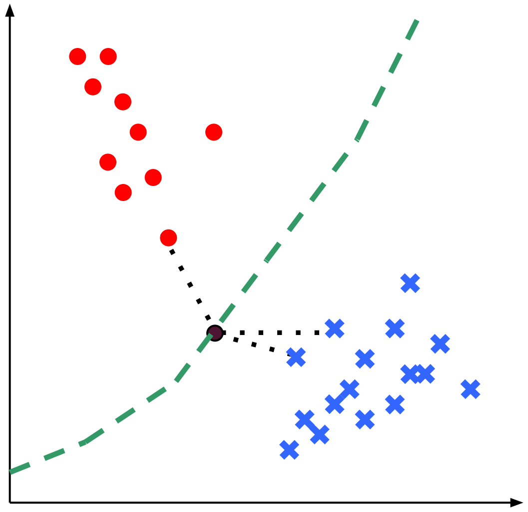
"TRAINING" DATA

Here I chose k=3.

What would happen if I chose k=5?

What would happen if I chose k=26?

The K-Nearest Neighbour Classifier



Person	Weight	Height
1	63kg	190cm
2	55kg	185cm
3	75kg	202cm
4	50kg	180cm
5	57kg	174cm
…	…	
16	85kg	150cm
17	93kg	145cm
18	75kg	130cm
19	99kg	163cm
20	100kg	171cm
…	…	

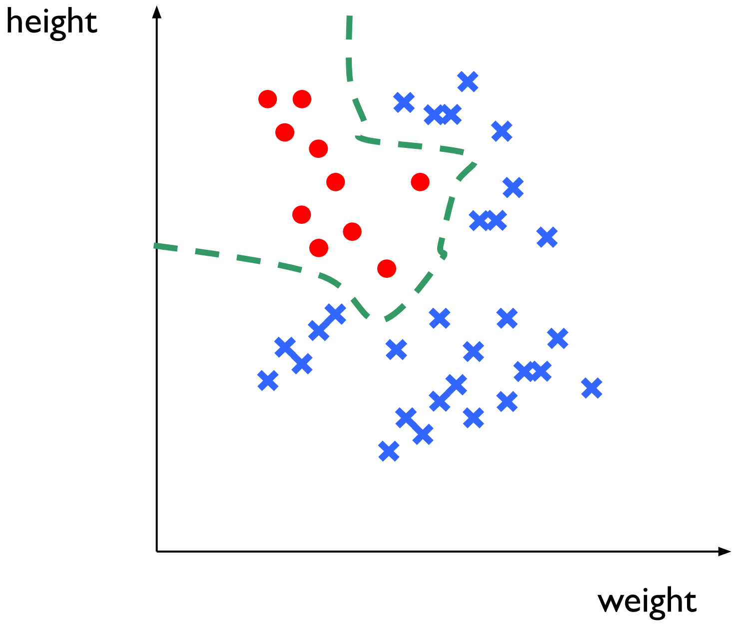
"TRAINING" DATA

*Any point on the left of this **"boundary"** is closer to the **red circles**.*

*Any point on the right of this "**boundary"** is closer to the **blue crosses**.*

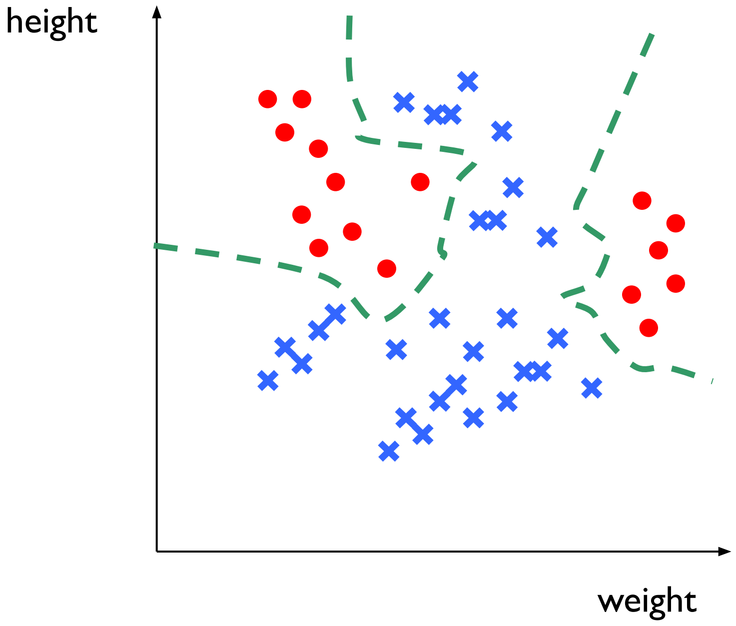
*<u>This is called the **"decision boundary".**</u>*

Where's the decision boundary?



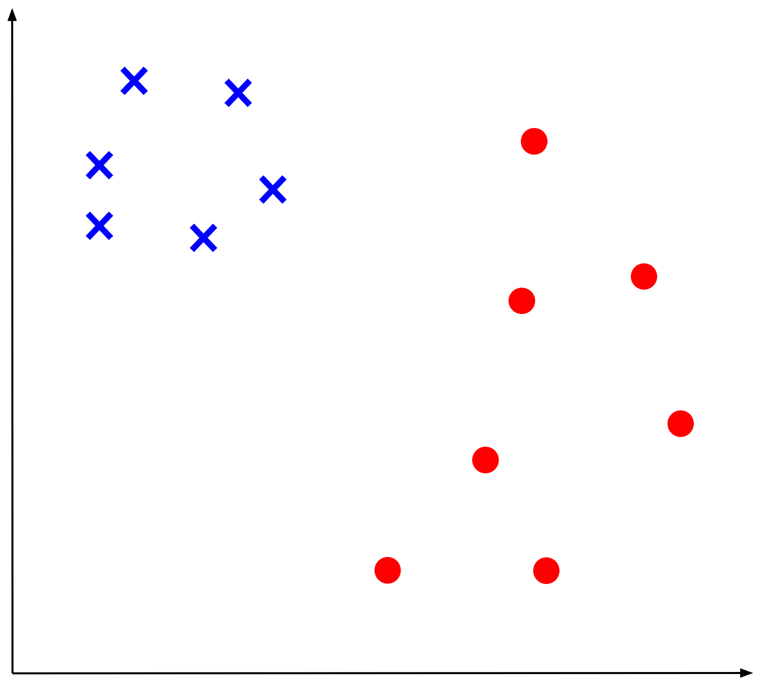
Not always a simple straight line!

Where's the decision boundary?



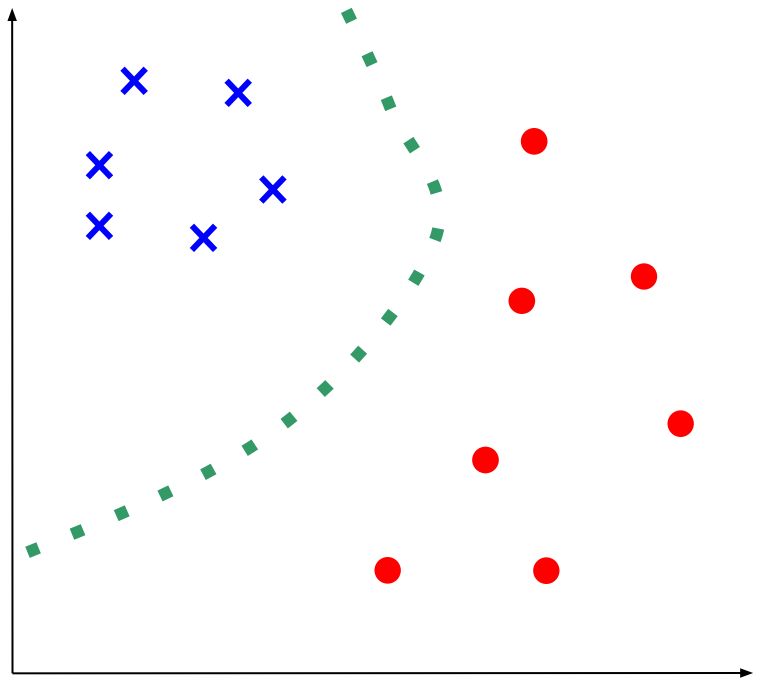
Not always contiguous!

The *most important* concept in Machine Learning

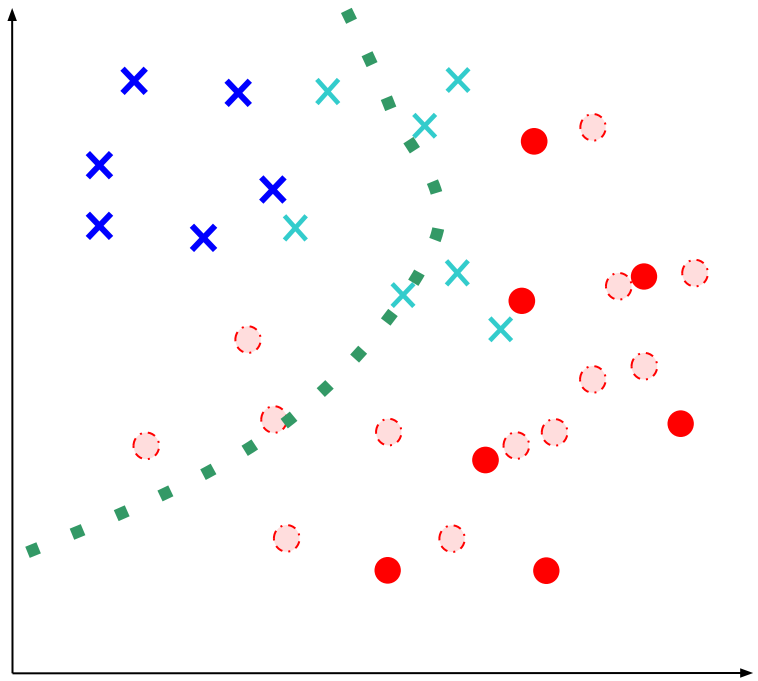


The *most important* concept in Machine Learning

Looks good so far…



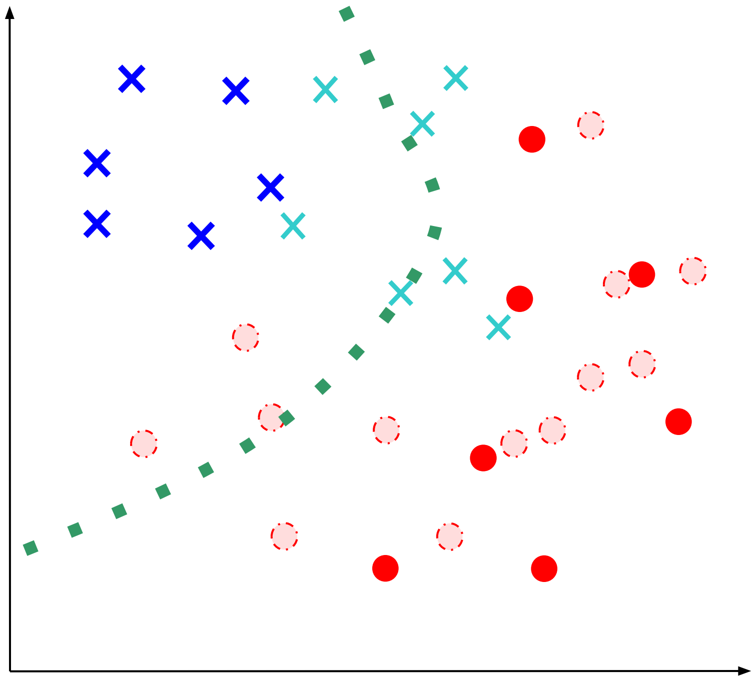
The *most important* concept in Machine Learning



Looks good so far…

Oh no! Mistakes!
What happened?

The *most important* concept in Machine Learning



Looks good so far…

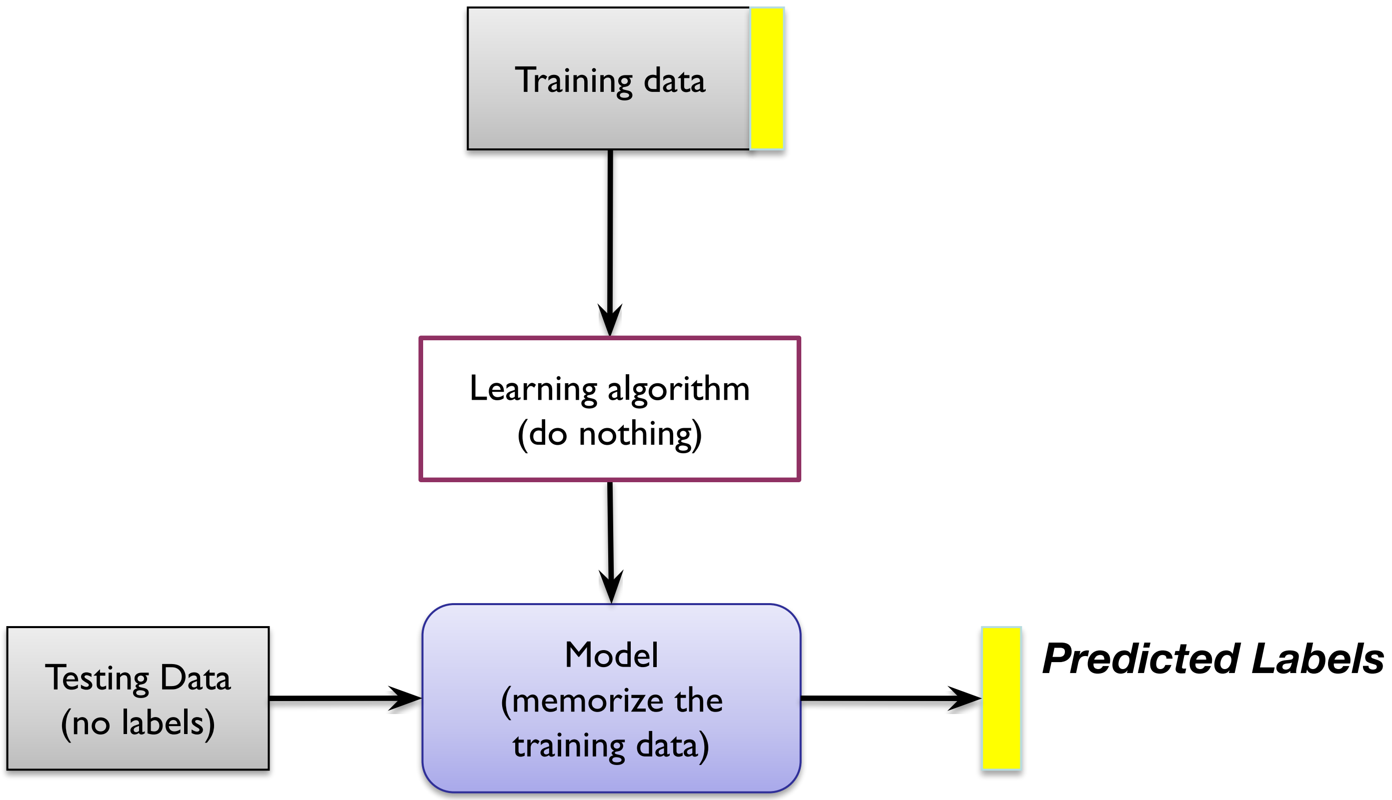
Oh no! Mistakes!
What happened?

We didn't have all the data.

We can never assume that we do.

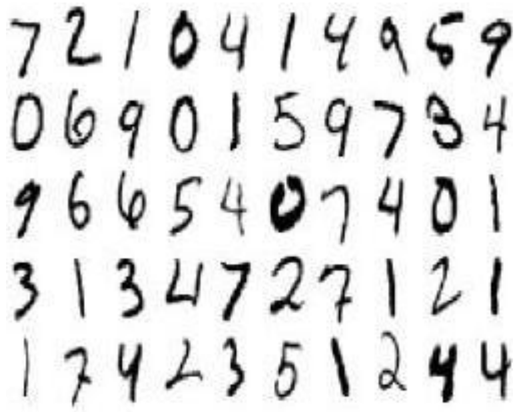
This is called "OVER-FITTING"
to the small dataset.

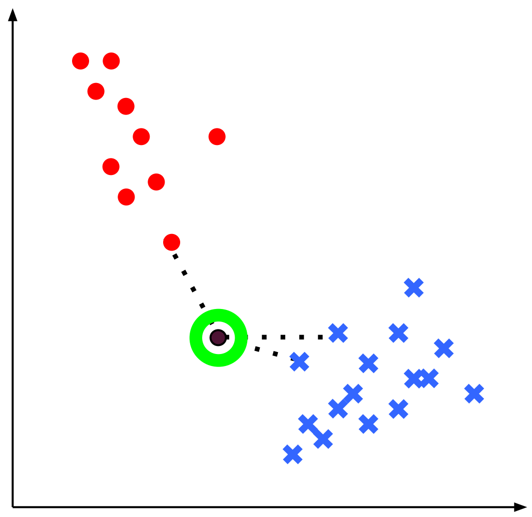
Pretty dumb! Where's the learning!



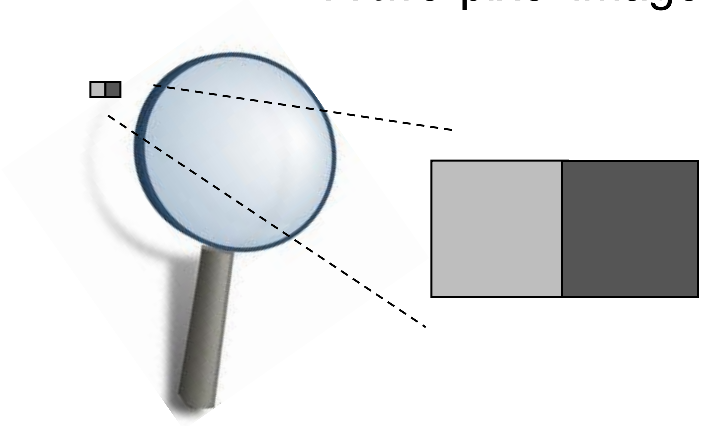
Now, how is this problem like

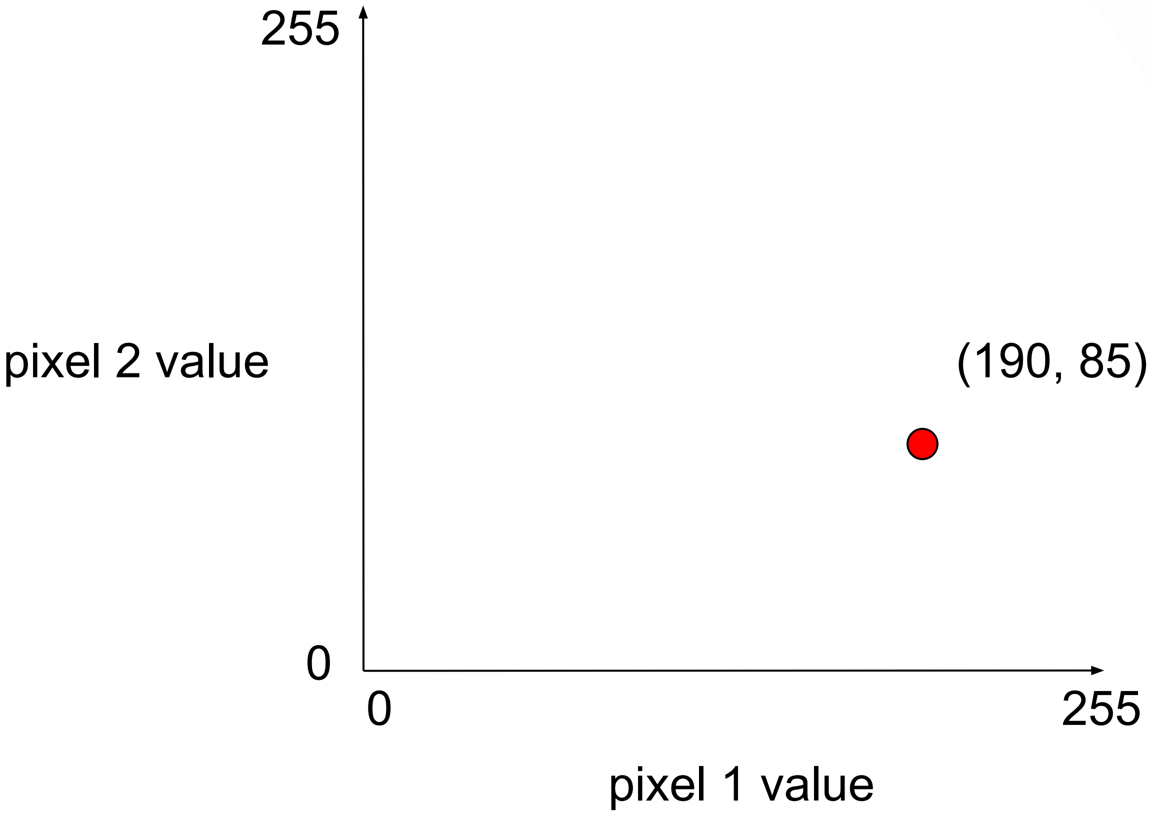
handwriting recognition?





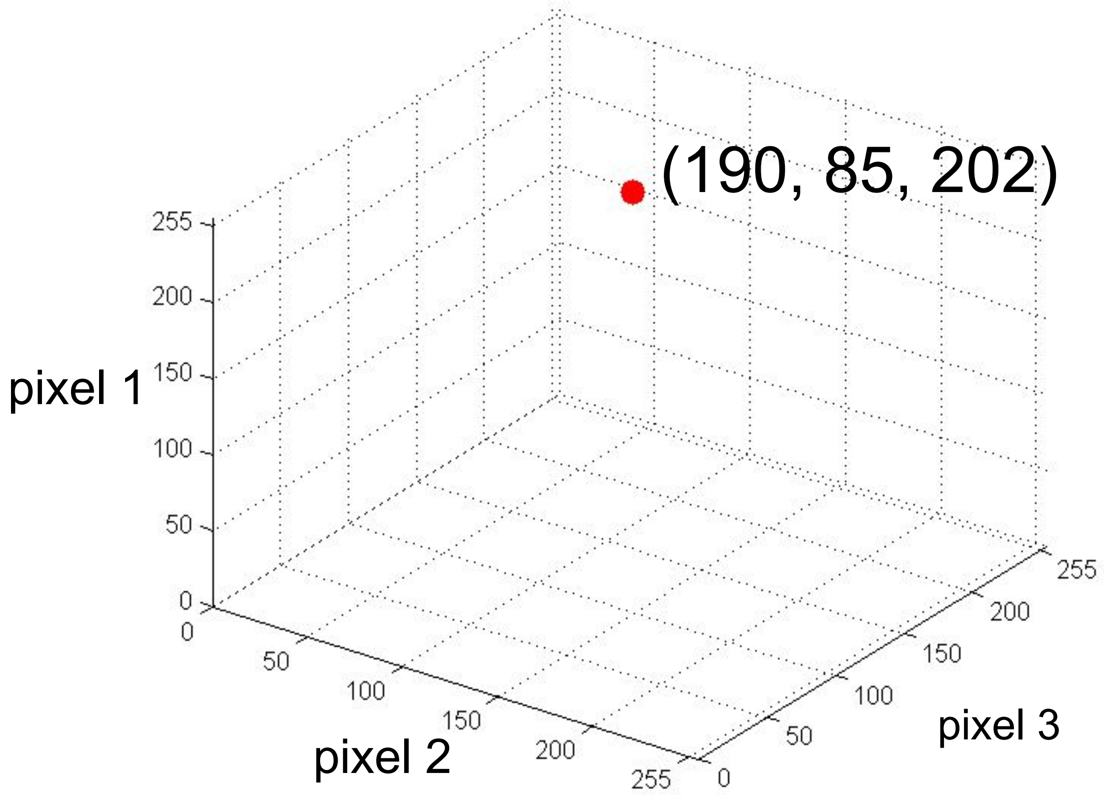
Let's say the measurements are pixel values.





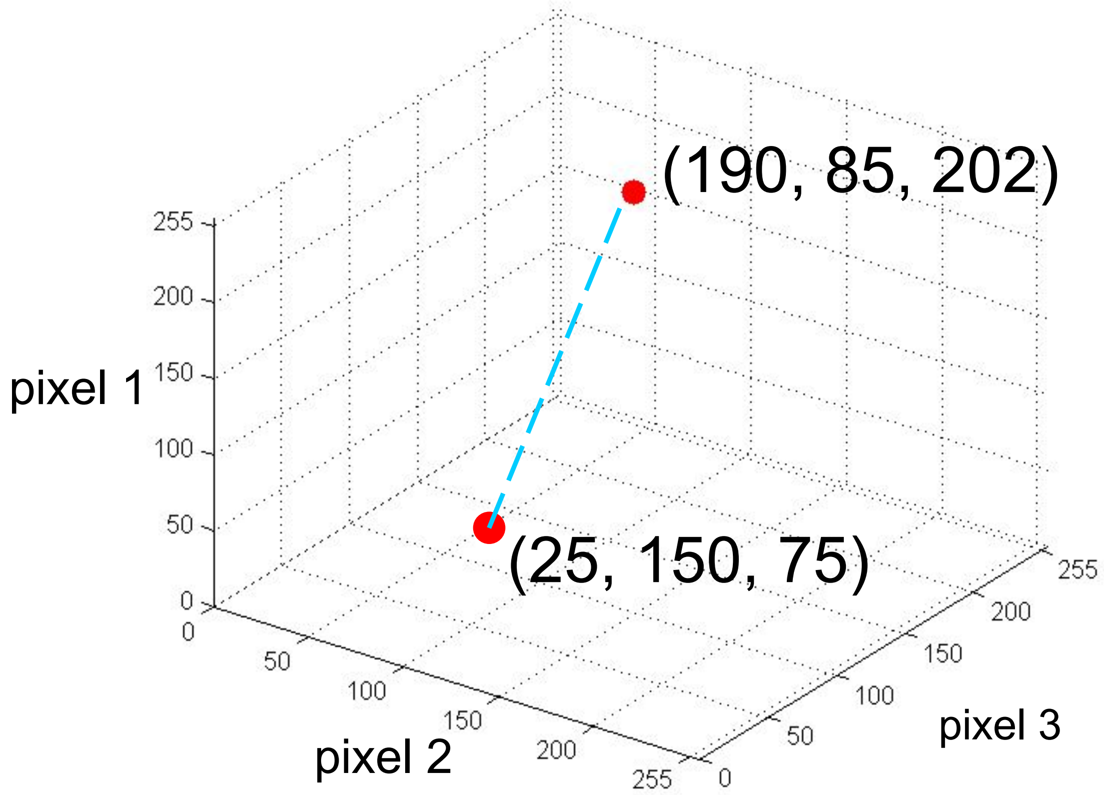
Three dimensions…

A three-pixel image



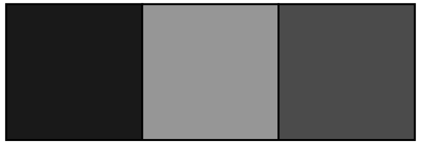
This 3-pixel image is represented by a SINGLE point in a 3-D space.

Distance between images



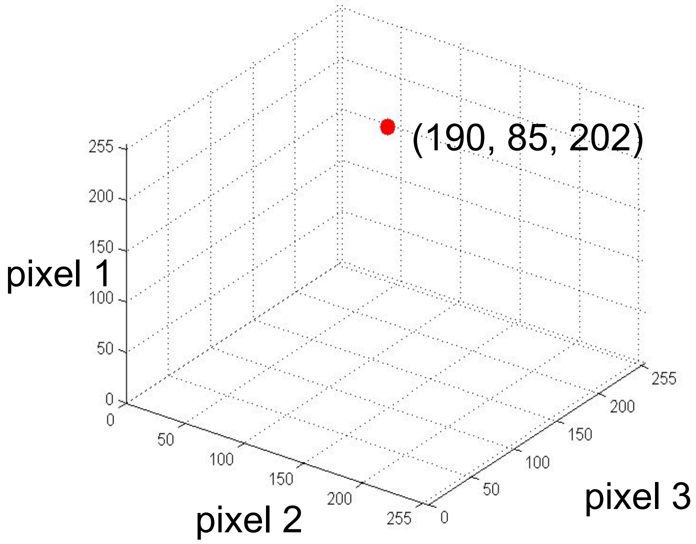
A three-pixel image

Another 3-pixel image



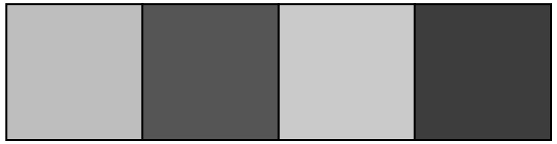
Straight line distance between them?

4-dimensional space? 5-d? 6-d?

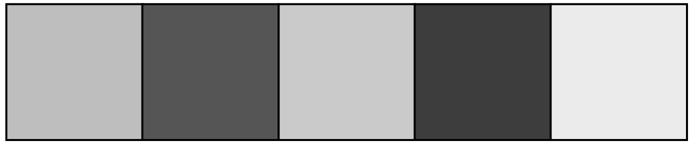


A three-pixel image

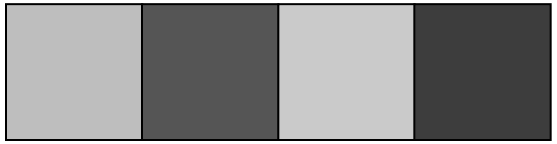
A four-pixel image.



A five-pixel image

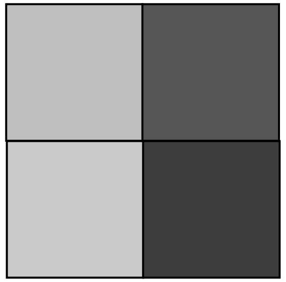


A four-pixel image.

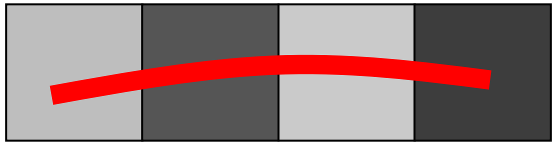


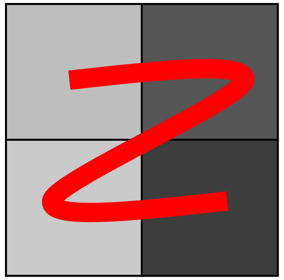
(190, 85, 202, 10)

A different four-pixel image.



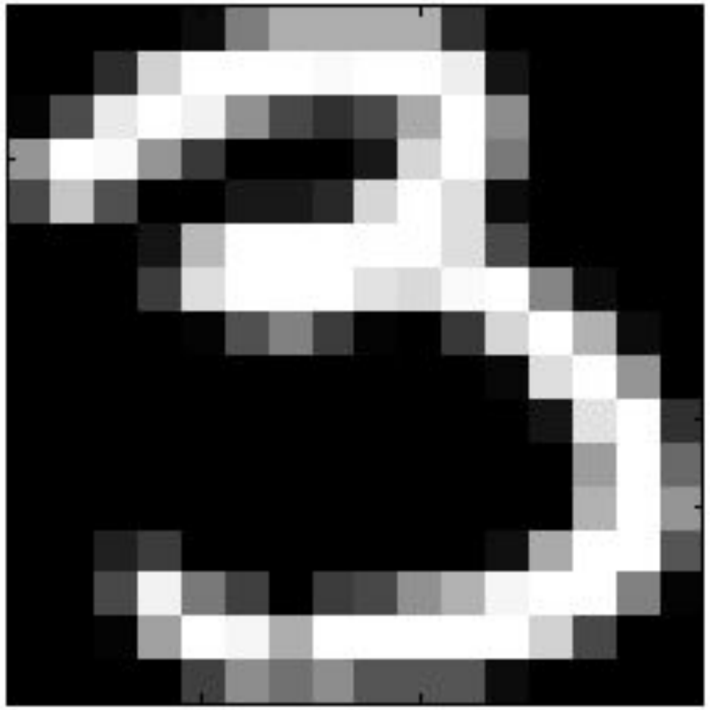
(190, 85, 202, 10)
Same 4-dimensional vector!



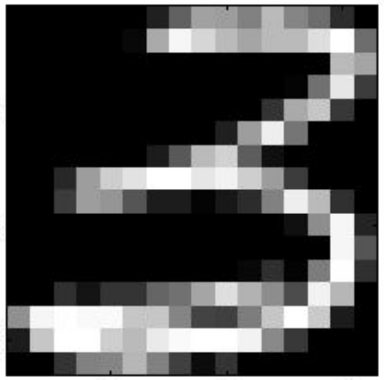


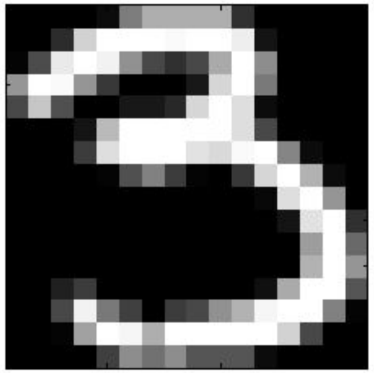
Assuming we read pixels in a systematic manner, we can now represent any image as a <u>single point in a high dimensional space.</u>

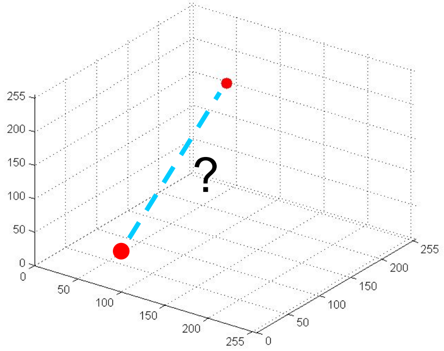
16 x 16 pixel image.   How many dimensions?



We can measure distance in 256 dimensional space.







$$\text{distance}(x, x') = \sqrt{\sum_{i=1}^{i=256} (x_i - x'_i)^2}$$

Which is the **nearest neighbour** to our ‘3’ ?