Artificial Intelligence

Foundations of Machine Learning I

Prof. Ignacio Olmeda

DEFINITION OF *MACHINE LEARNING*

- Learning is an important characteristic of any living system: organisms, from amoebas to humans, organisms adapt to the environment, trying to survive, they *learn*.

Learning is NOT a human characteristic

- In extension, we may also figure artificial systems that adapt to their environment and *learn*, in the words of Alan Turing in 1947:

  "what we want is a machine that can learn from experience."

- Intuitively machine learning is concerned with the question of how to infer knowldge from data using artificial systems

- An alternative view is how construct computer programs that automatically improve with experience; this improvement, in the context of AI is called *learning*.

- One of the earliest definitions of Machine Learning is due to Arthur Samuel (1959):

"the field of study that gives computers the ability to learn without being explicitly programmed"

- Mitchell (1997) provides an authoritative –formal- definition and some examples of *Machine Learning*:

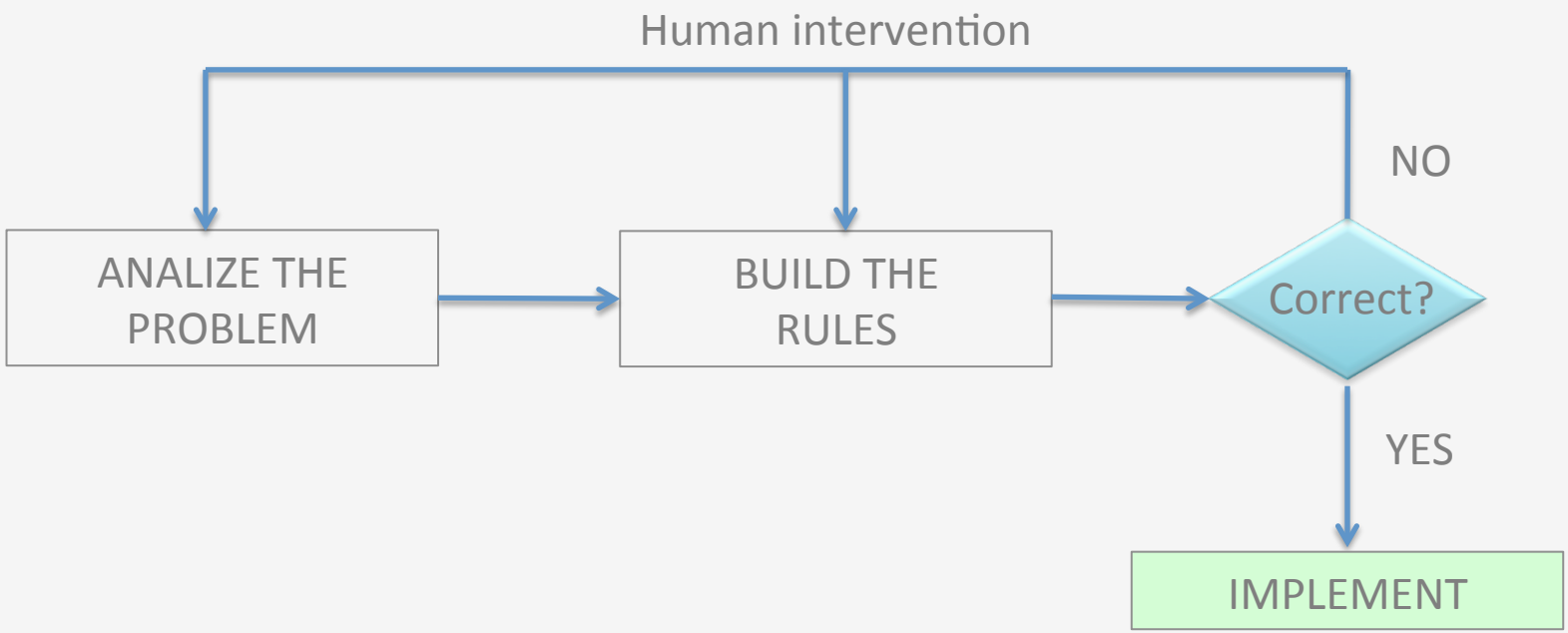
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

- Important concepts:

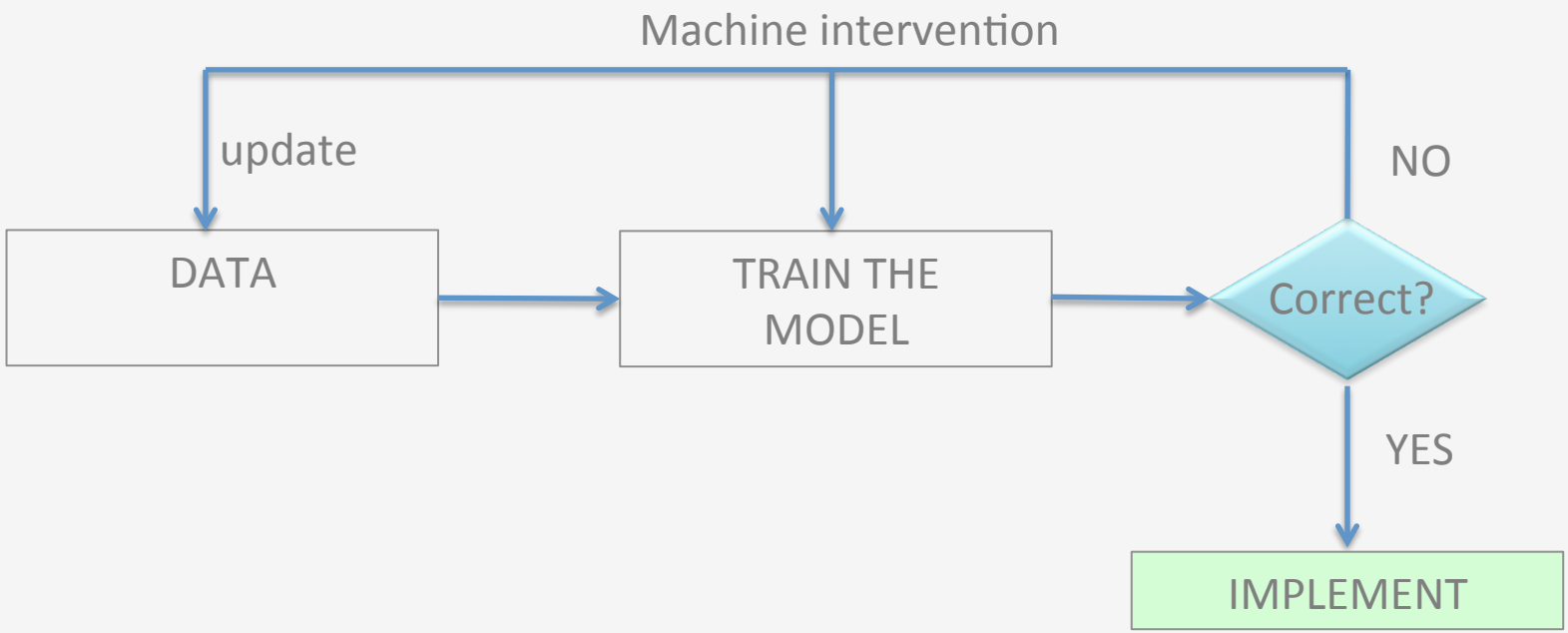
  E: *Experience*
  P: *Performance*
  T: *Task*

- ML allows to simplify enormously the process of building artificial decision systems.

- In the "old"paradigm of Expert Systems one has to iterate through a flow graph like this:



- While in ML the flow would be similar to this:



- Note also that the Expert Systems approach can be unfeasible when there are no clear “rules” or associations between input and output: it is very easy to tell whether some particular music likes us or not but very complicate or even impossible to explain the reasons why.

- Finally, ML will be useful in situations where, even though rules may exist, the environment changes at a fast pace, making difficult or impossible to implement a solution crafted “by-hand”.

- Examples of ML (Mitchell, 1997):

- A checkers learning problem:
  - *Task T*: playing checkers
  - *Performance measure P*: percent of games won against opponents
  - *Training experience E*: playing practice games against itself

- A handwriting recognition learning problem:
  - *Task T*: recognizing and classifying handwritten words within images
  - *Performance measure P*: percent of words correctly classified
  - *Training experience E*: a database of handwritten words with given classifications

- A robot driving learning problem:
  - *Task T*: driving on public four-lane highways using vision sensors
  - *Performance measure P*: average distance traveled before an error (as judged by human overseer)
  - *Training experience E*: a sequence of images and steering commands recorded while observing a human driver

- The power of Machine learning lies in the fact that it makes computers to program themselves.

- In "classical" programming, one program and a set of data produce some output, notice that the result is, again, data.

INPUT DATA + ALGORITHM = OUTPUT DATA

- In machine learning, a set of data and a desired output produce some program, notice that the result is a program, not data.

INPUT DATA + OUTPUT DATA = ALGORITHM

- So, in the output of machine learning algorithms are of a totally different nature as the inputs.

- Any ML algorithm has three basic components:

  1. Representations (*model*)
  2. Evaluation (*goal*)
  3. Optimization (*algorithm*)

- In terms of representation there are literaly hundreds of models: Decision trees, Neural networks, Support Vector Machines, Classification and Regression Trees or even Mixtures of models.

- In terms of evaluation we have also dozens of performance measures: Accuracy, Precision and Recall, Mean Suared Error, Absolute Error, Confusion Matrices, Likelihood, Expected Utility, Entropy, Kullback-Leibler Divergence, Mahalanobis Distance…

- In terms of optimization we may also think on a variety of algorithms:

  - Deterministic: Gradient Descent and all variants (e.g. stochastic variance reduced gradient, SAGA, L-BFGS, Conjugate Gradient…) , Dynamic Programming methods

  - Heuristic: Simulated Annealing, Genetic Algorithms, Tabu Search, Particle Swarm Optimization, ….

- Combinations are explosive, we can employ different algorithms or performance measures under the same model, this makes the possible combinations almost unlimited:
    - A Deep Network trained with backprop to minimize the sum of squared errors
    - A Decision Tree, built using cross-entropy to maximize the number of correct classifications
    - A Deep Network trained with a Genetic Algorithm to maximize a non differentiable loss function
    - ….

- Most of the uses of ML are *predictive* in the sense that we want to know what will happen in the future or what will be the outcome of some particular action; examples are:

  - Forecasting the price of a particular criptocurrency
  - Predinting the next move of the opponent in a chess game
  - Forecasting whether one consumer will buy or not one product based on its web log history

- In some other cases the purpose is *descriptive*, we want to understand the hidden relationships in the data for different purposes; examples are:
  - Understanding risk factors in some medical diagnosis process
  - Understanding the relationships in the buying behaviour of consumers by analysing their comusumtion basket
  - Understanding the charateristics of some particular material based on its molecular structure.

- Both interests do not necessarily fit into any particular kind of learning but it is common that prediction employs supervised algorithms (defined latter) and description employs unsupervised (or semi supervised) algorithms (also defined latter).

- Notice that there is a close similarity between Machine Learning and Statistics since we also find both approaches in Statistics:

  - *Descriptive Statitics*: Analysis of data that helps to describe or summarize data in a meaningful way i.e. finding patterns in the data

  - *Inferential Statistics*: Techniques that allows us to make generalizations from samples taken from data.

- Since there is a close relationship between ML and Statistics, it is very useful to have a good knowledge of the formal methods employed in Statistics (and also Mathematics).

- It is even possible to stablish direct relationships between concepts of Machine Learning vs. Statistics, e.g. in the context of Deep Learning:

Synaptic Weights = Parameters

Learning = Estimation

Network = Model

Backprop Algorithm = Robbins-Monro's Stochastic Approximation Method

Supervisd Learning = Regression

Unsupervised Learning = Clustering

Performance = Loss Function

LEARNING TYPES

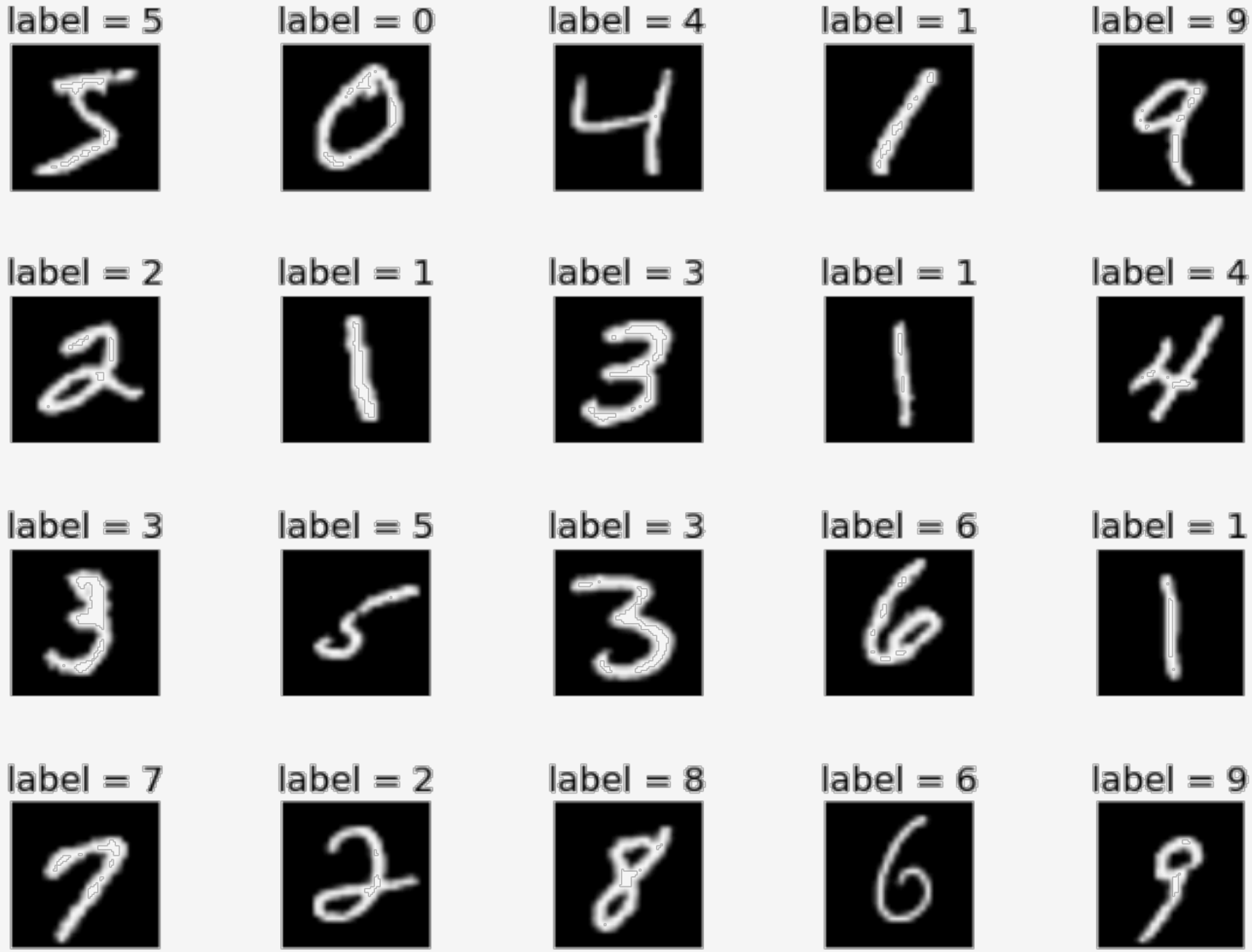
- In terms of the degree of intervention of an external "teacher" we can diferentiate among three kinds of learning: "Supervised", "Unsupervised" and "Reinforced".

- In *Supervised Learning* we try to find the relationship between an *dependent* variable Y and some *explanatory* variables X, using labeled data.

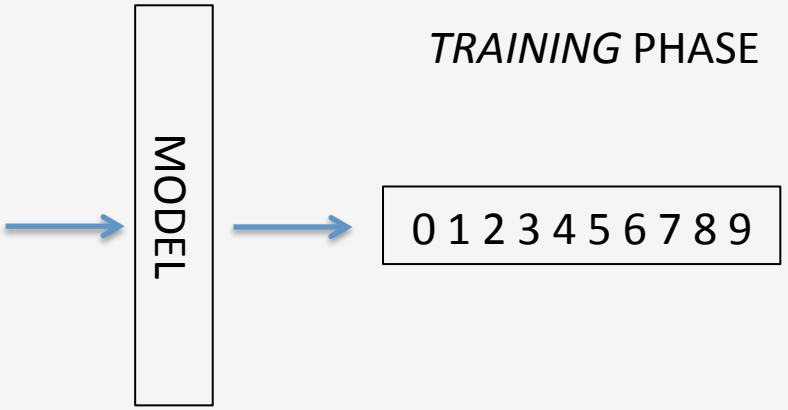
- This means that we have to know, in advance, which are the correct labels for a set of examples that are used to build the model.

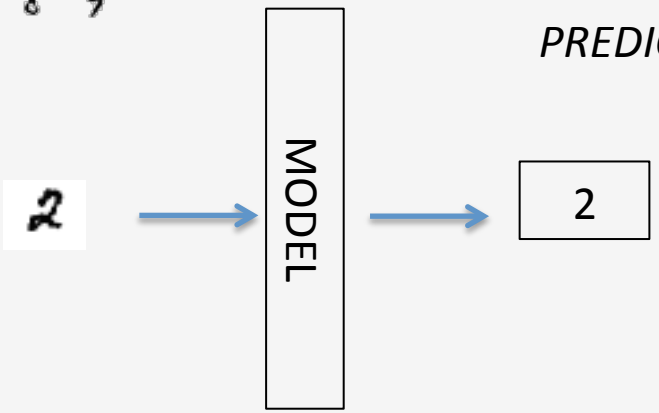
- For example, the MNIST database
  http://yann.lecun.com/exdb/mnist/
  contains a set of examples of hanwritten digits with the corresponding labels.

- This example corresponds to an area called *Optical Character Recognition* (OCR) which was highly sucessful in the 90's, and was used, for example recognizing the numbers of ZIP codes in envelopes.


label = 5
label = 0
label = 4
label = 1
label = 9
label = 2
label = 1
label = 3
label = 1
label = 4
label = 3
label = 5
label = 3
label = 6
label = 1
label = 7
label = 2
label = 8
label = 6
label = 9

- This database can be used to train a model and then use that model to identify a new unseen example of a handwritten digit:





- The *predictor variables* (also called *independent variables* or *features*) allow to determine the *responses* (also called the *dependent variables*)

- Formally, in the training phase we have a set of observations (X,Y) and the problem consist on finding the optimal parameters Θ that index some model f ; these parameters should be optimal under some performance measure.

$$Y = f_{\Theta}(X)$$

Y: responses
X: independent variables

- For example, in the case of MNIST the performance could be the number of misclassifications on a particular database.

- After the optimal parameters are found, the model is generally used to forecast unseen examples X'

$$\hat{Y} = f_{\hat{\Theta}}(X')$$

- Notice that the bracket over Y means that, in fact, we obtain predictions (not real data).

- After predictions are found we can calculate the overall performance with real data by computing some distance between predictions and real observations:

$$performance = g(\hat{Y}, Y)$$

- Note that the preceeding is an example of a *classification* problem, we have ten *classes* or *labels* (numbers 0 to 9) and the task is to correctly classify a handrwritten number.

- One important problem in supervised learning is *dimensionality reduction* which consists on the process of discovering the explanatory variables that account for the greatest changes in the response variable.

- Dimensionality reduction is, many times, a previous step in ML e.g. when the high number of features make it too cubersome to deal with them (due to the computational effort needed).

- The problem of transforming raw data into a usable dataset is called *feature engineering* and many times is a labor-intensive process that demands time –and skills- from the data analyst .

- Supervised learning can be also employed in *regression* problems where the response is not restricted to belong to a finite set of labels.

- For example, in predicting the prices of houses we may think that the problem is:

$$price = f_{\Theta}(size, no.rooms, dist.school)$$

- In some other regression examples, the response variable is not limited to a finite number of classes but it can only take some bounded values.

- For example, in a *credit scoring system* we may use as input variables the income of the applicant, age, whether she has a car etc. and the output will consist on the probability of returning a consumer loan, that can be any number but only between zero and one

- In this case:

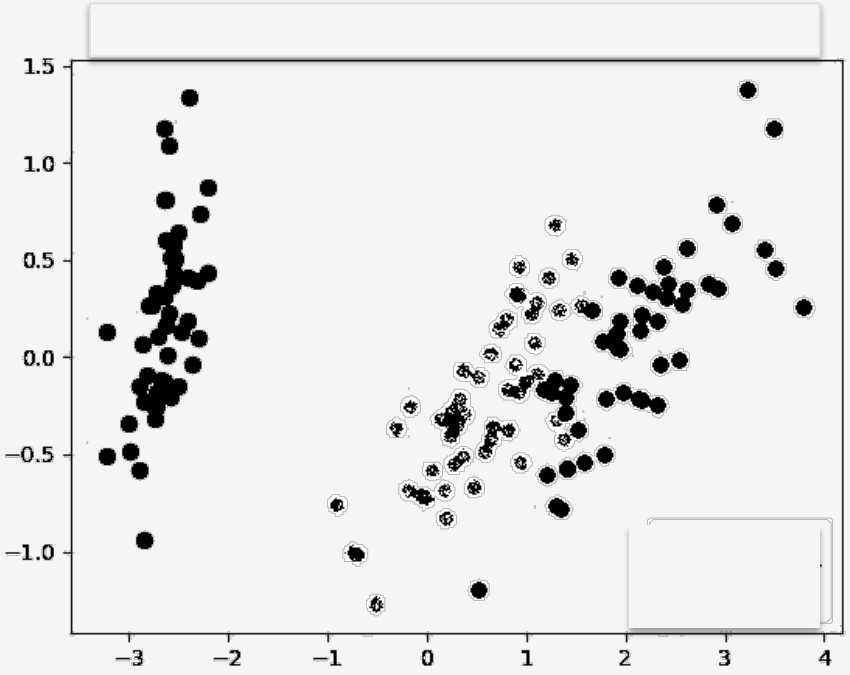
$$prob = f_{\Theta}(income, age, car(yes / no)...) \in [0,1]$$

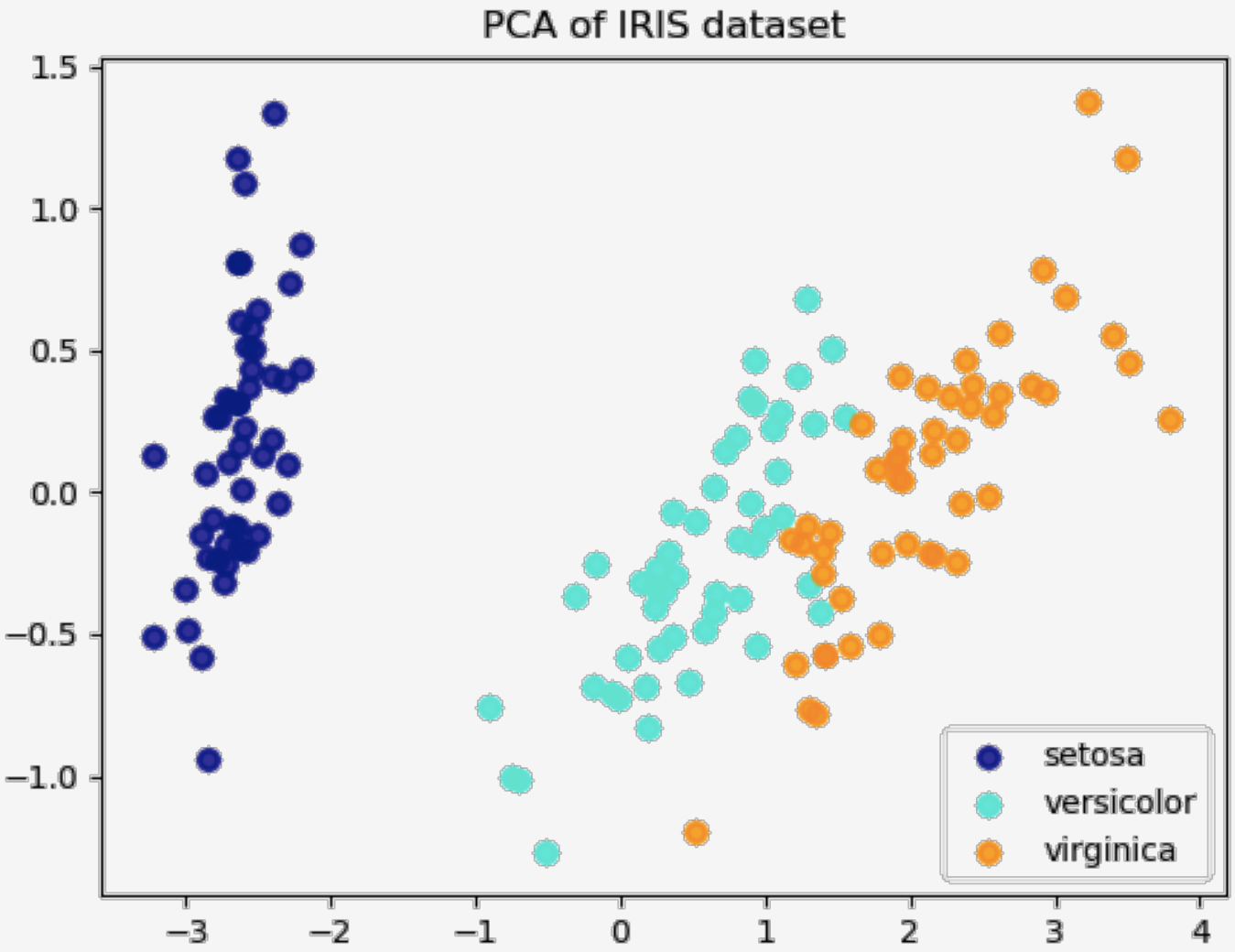
- Note also that supervised problems are not perfectly defined, for example, in the case of credit scoring the output could be a rating (unbounded), a class (default/non-default), or a probability ([0,1]).

- Some examples of the techniques that are used in supervised learning are the following:
    - Linear Regression
    - Logistic Regression
    - k-Nearest Neighbors
    - Decision Trees
    - Support Vector Machines (SVMs)
    - Feedforward Neural Networks

- In *Unsupervised Learning* we do not know the labels of the examples X and the objective is to clasify them into K categories where K can be a specified or unespecified number.

- For example, the well-known iris data (https://archive.ics.uci.edu/ml/datasets/iris) contains examples of three kinds of flowers, and specimens must be classified in each one of the three classes according to the petal and sepal size.




PCA of IRIS dataset
1.5
1.0
0.5
0.0
−0.5
−1.0
−3
−2
−1
0
1
2
3
4
setosa
versicolor
virginica

- Note that in unsupervised learning the system tries to learn without a "teacher".

- One example of unsupervised learning is customer classification, we would like to obtain typologies of consumers using socioeconomic data as well as data of past transactions.

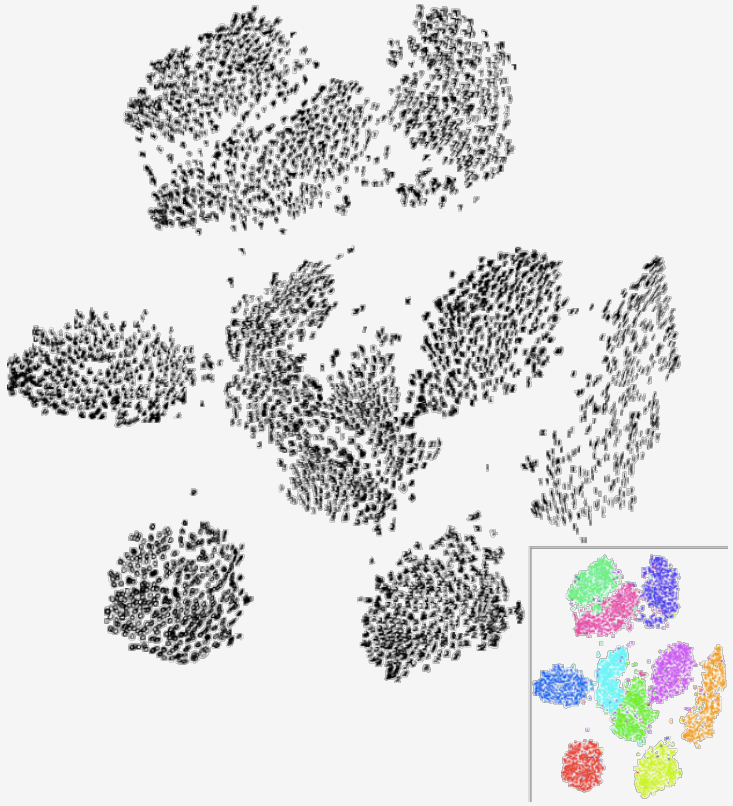
- After the typologies have been established, firms may target each of the groups with specific marketing strategies.

- Other application is chatbots: when interacting with a chatbot one of the problems is to detect the *intention* of the consumer and then cluster customers with similar intentions.

- Unsupervised learning is widely employed in dimensionality reduction i.e. problems where we want to have a simplified version of the data.

- A particular application of this are *visualization algorithms* where one wants to have a graphical, intuitive view of the data.

- For example, this is a graphical representation in 2-D of the MNIST database



- Note that, in some cases, the problem can be considered as supervised or unsupervised depending on the objectives or the availability of data.

- We can even employ a miture of both, for example, we can employ unsupervised learning to clusterize customers with similar intentions and then to employ supervised learning to characterize the intentions of a specific group.

- Unsupervsed learning is used also as a previous step when one wants to reduce the dimensionality of the problem.

- Another application of unsupervised learning is *anomaly detection* where one wants to detect observations that can be considered as *anomalies* or *outliers*.

- These anomalies could distort the normal process of model construction so that such anomalies are previously eliminated in the training phase.

- Some example of techniques used in unsupervised learning are:
    - K-Means
    - DBSCAN
    - Hierarchical Cluster Analysis (HCA)
    - Isolation Forest
    - Principal Component Analysis (PCA)
    - Locally-Linear Embedding (LLE)

- Finally, there exists an approach that is in some sense a mixture of supervised and unsupervised learning and which is refered as *semi-supervised learning*.

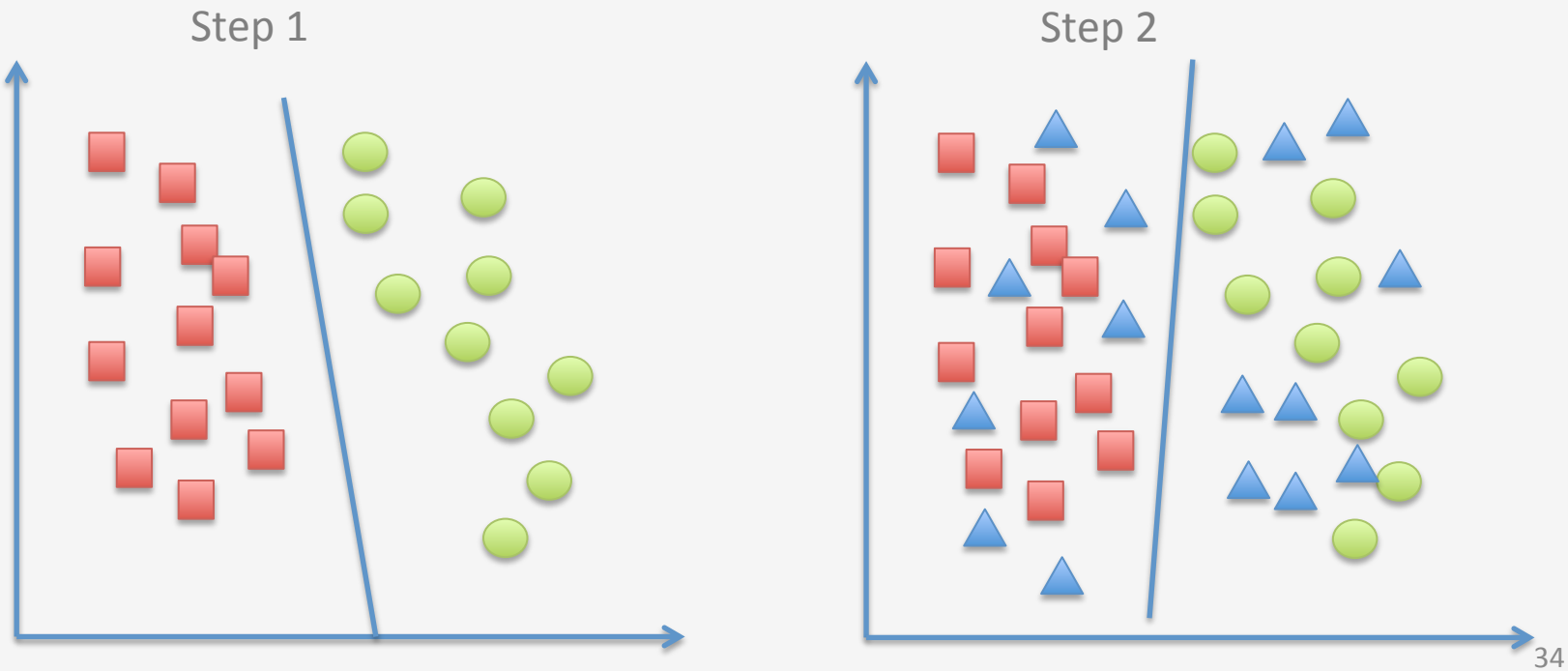
- Semi-supervised learning is a technique employed in situations where the dataset contains labeled and unlabeled examples and the number of unlabeled examples is much higher than of labeled examples.

- Note that,in general, to have more examples, even if they are unlabeled, it is beneficial for ML systems that require datasets as big as possible.

- Semi-supervised learning essentially consists on two steps:

    1. In the first step examples are grouped according to the shared features

    2. In the second step, after one of the examples of each of the groups is labeled, then the rest of the remaining examples are also labeled



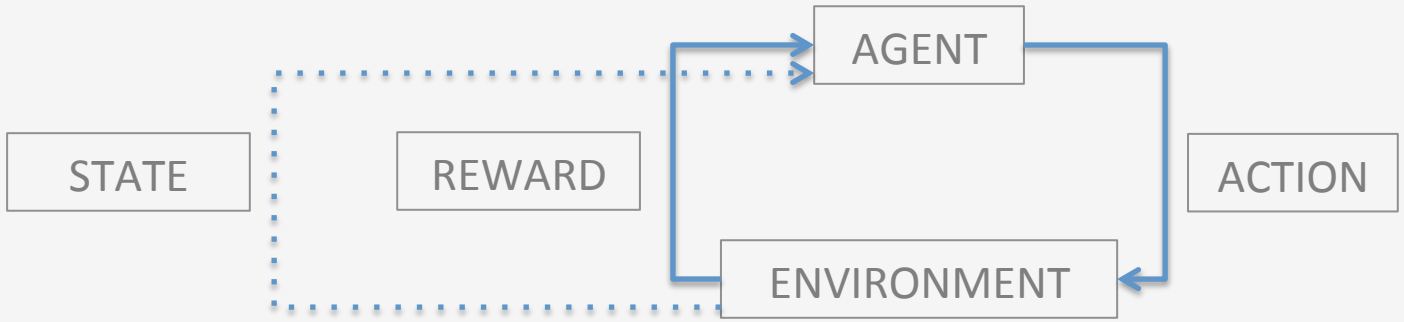
- In *reinforcement learning* the objective consists on using observations gathered from the interaction with the environment and taking actions that maximize some reward.

- The learning system is called an *agent* in this context and we asume it can observe the *environment* and perform some actions.

- The agent obtains some *rewards* in return (or *penalties* in the form of negative rewards).

- The agent must learn by itself (changing its *state*) what is the best strategy -called a *policy*- to get the most reward over time.

- The action would be optimal if it maximizes the expected average reward.



- Reinforced learning can be viewed as an intermediate situation between supervised and unsupervised learning, there is some feedback but not as strong as in supervised learning or non-existent in unsupervised learning.

- Reinforced Learning differs signifficantly from the other two paradigms in several aspects but the most important is that rewards are delayed: past performance affects future estate of the system (other architectures share this characteristic though, e.g. *recurrent networks*).

- Some example of techniques used in reinforced learning are:

    - Q-Learning
    - State-Action-Reward-State-Action (SARSA)
    - Deep Q Network (DQN)
    - Dyna-Q
    - Deep Deterministic Policy Gradient (DDPG)
    - Twin Delayed Deep Deterministic Policy Gradients (TD3)

- Note that learning can happen *incrementally* (the algorithm acts and learns at the same time, as we humans do) or *sequentially*.

- In the first case we say that the algorithm learns in an *online* mode while in the second we refer it as a *batch* learning or *offline* learning.

- In online learning, the model is trained incrementally feeding data sequentially, either individually or by small groups called *mini-batches*.

- Since, in many models, data is encoded in the form of parameters (e.g. the synaptic weights in a deep learning model) then online learning allows to discard training data as soon as parameters are updated with the new data, saving memory space.

- In general, both kinds of learning are compatible and the use of one or another depends on the graduality in the changes introduced in the training data as well as the speed needed to produce a *recall*.

- Availability of computing resources (CPU, memory and disk space, network and I/O…) often determines whether batch or incremental learning is appropriate.

- Note that incremental learning imposes the need of having some automatic control on the quality of data: if new data is corrupted then the system will change gradually so that we would not notice the degradation of the model.