

Low Code Machine Learning Course

Reza Rezazadegan

Shiraz University, Department of Mathematics and Computer Science, Fall 2024

www.dreamintelligent.com

Telegram: @rrezaza



Course Description.....	2
What is Artificial Intelligence (AI)?	3
From Symbolic AI to Machine Learning.....	3
Enters Machine Learning.....	4
AI as function approximation.....	5
Representing different data types numerically	6
AI as optimization.....	9
How an ML model is evaluated.....	13
Machine Learning vs Data Science.....	14
Unsupervised Learning.....	14
Manifold learning.....	15
Reinforcement learning.....	18
ML versus science.....	18
The lifecycle of a technology: how AI is becoming mainstream.....	20
Citizen data scientist.....	22
Edge AI.....	23
Introducing AutoML.....	24
The problems we study in this course	25
Binary classification	25
Regression.....	25
Multi-class classification.....	25
Clustering.....	25

Course Description

This course and accompanying workshop are an accessible introduction to AI and Machine Learning (ML). We are going to make use of Low Code Machine Learning (or AutoML) tools and try to have as few prerequisites as possible.

After an initial introduction to what AI and ML are, we review the life cycle of a technology and explain why AI and ML keep getting more and more accessible and easier to use and even train AI models. We then study four explicit ML problems:

1. two classification problems (binary and multi-class),
2. a regression problem,
3. a clustering problem ,
4. a problem in forecasting time series.

We take an Up-to-Bottom approach and explain (some of) the tools involved in solving these problems after making use of them. The guiding principle of this course is NOT that AutoML can replace a thorough ML training, but that:


- AutoML tools can help with ML and AI education
- Learning ML should involve as few software and programming idiosyncrasies as possible.

This is a leisurely course which is expected to extend beyond the originally designated number of lectures. Students are encouraged to give presentations on topics they are interested in, which we don't get around to explain as part of the course.

What is Artificial Intelligence (AI)?

The New York Times

NEW NAVY DEVICE LEARNS BY DOING; Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

 Share full article



July 8, 1958

Simply put, AI is enabling machines to do "intelligence" things, like humans do. But first, what is intelligence?

There are different definitions; however, intelligence can be reduced to learning from experience. Or in other words, the ability to *generalize* i.e. the ability to apply learned knowledge to new unseen situations.

A good example of the generalization capability of humans is their ability to guess the next numbers in a sequence, based on the knowledge of the initial ones e.g.

2, 6, 12, 20, 30?

From Symbolic AI to Machine Learning

In the early days, artificial intelligence involved logic, rules and inference. Good examples of such Symbolic AI are *expert systems* which drive conclusions based on a set of rules, e.g. for diagnosing diseases based on the symptoms.

Closely related is *fuzzy logic* which is used e.g. in automatic car transmission: making decisions for gear shift based on inputs such as car speed, throttle position and engine load.

Low Code Machine Learning Course

However, it was later revealed that logic and symbolic AI is not capable of complex problems related to e.g. Computer Vision and language. This happened e.g. by the failure of machine translation.

This resulted in the *AI winter*, starting in 1974 and continuing for most of the remaining part of the 20th century

"I think that in practical terms, it's a mirage, in the sense that if it's something that we think we can see on the horizon, in the sense that on our deathbeds it may be announced or our children will see it, that it's really there on the horizon, then I disagree with such a view." James Lighthill, 1973. From the Lighthill Debate on The Future of AI

Enters Machine Learning

The data-centric revolution in AI happened by observing that we can develop (*train*) a mathematical or statistical model based on existing data, so that it can make inferences (or predictions) for the cases it has not seen! Such a model would have the ability to generalize. This is called *Machine Learning*!

For example, imagine you want to have a model which can take a product review and decide whether it is positive or negative. The traditional method is to look for positive or negative words in the review and decide accordingly. (Although such a model may get confused when a review contains both positive and negative words.)

But the modern approach is to take a bunch of reviews which are already labeled as positive or negative and then feed them to a model which basically learns which words or phrases are more frequent in either type of review.

This line of thought is radically different from the original Symbolic AI and it has resulted in breakthroughs such as:

- Machine translation of text and speech
- Text to speech and speech to text
- Text, image, video and music generation

Low Code Machine Learning Course

- Question answering and dialogue Even mathematical reasoning
- Face and object detection in images
- Predicting product demand (Although this goes back to before Machine Learning!)

AI as function approximation

How can machines achieve feats such as those in the above list?

Looking at the list above, we see that most "intelligent" tasks involve mapping one type of data (which is involved in the human experience of the world) to another, e.g.

- image to text (image captioning)



A person is walking along a beach with a big dog



A black and white dog carries a tennis ball in its mouth



A soccer player takes a soccer ball in the grass



A man is doing a trick on a snowboard



A surfer dives into the ocean



A black and white dog leaps to catch a Frisbee

- text to text (translation, summarization, question answering)
- text to image (image generation)
- image to categories (object or face detection)



Low Code Machine Learning Course

- text to categories (sentiment analysis, text classification)

Therefore, if we can turn these data (text, images, etc.) into vectors of numbers, then AI would largely be reduced to approximating functions as follows!

We need a bunch of examples (a dataset) in which both the source and the target of the problem are known, for example:

- a bunch of images for which the name of the person in it, is known, or
- a bunch of reviews which are manually labeled as positive or negative.

The data associated with such examples is of the form

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

where each x_i or y_i is a vector of numbers and for each x_i (e.g. an image) we have the corresponding label y_i (e.g. a caption).

We want to find a *rule* or a *function* $f(x)$ that maps each x_i to its corresponding y_i i.e. $f(x_i) = y_i$, or at least close to it: $f(x_i) \simeq y_i$.

First let's see how different types of data can be represented numerically.

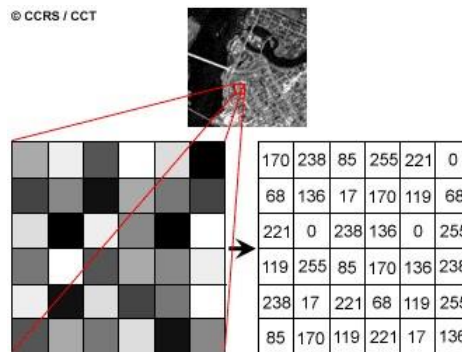
Representing different data types numerically

Remember that computers can only understand numbers!

Data representation is the task of representing different data types as vectors in a Euclidean space \mathbb{R}^d in such a way that, roughly speaking, data samples that are similar to each other are mapped to nearby vectors in \mathbb{R}^d . Each component of such a vector is called a *feature* of the data.

Numerical quantities such as temperature, blood pressure, exchange rates, etc. are already given by numbers! If we have, say, 3 numerical components (*features*) in our data (say, age, blood pressure and fasting blood glucose of a person), we can think of each data point as a 3-dimensional vector.

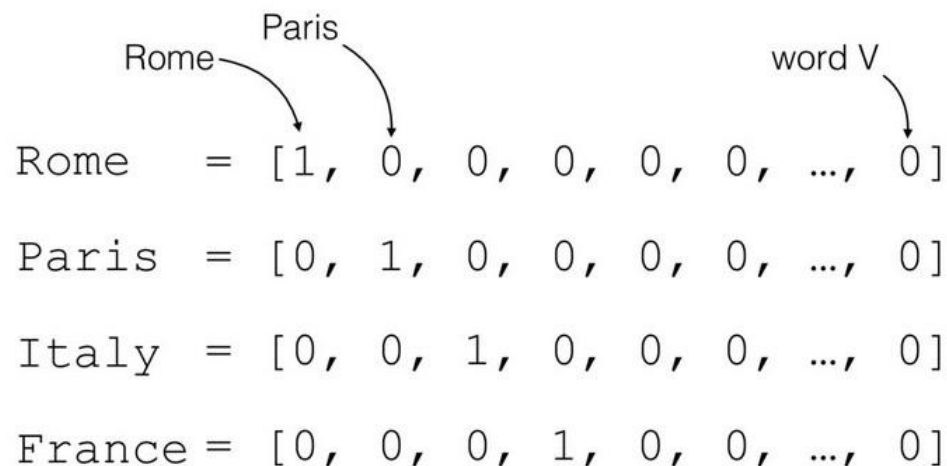
Images: A digital black-and-white image whose length and height are l and h pixels respectively, consists of $l \times h$ pixels, each of which is determined by the intensity of the image at that pixel: 0 for completely dark to 255 for completely bright. Thus, we can regard



such an image as an $l \times h$ matrix.

Color images have 3 values per pixel, called *channels*: corresponding to the intensity of the Red, Green and Blue values. Satellite images may contain UV and infrared spectrum as well and therefore can involve many channels.

Text: Computers can only understand numbers and so, we have to somehow turn words into numbers. One of the simplest ways to do so is called *1-hot encoding*. This means that if the



Low Code Machine Learning Course

words in the vocabulary are w_1, w_2, \dots, w_N we then represent each word by an N -dimensional vector such that the vector which has 1 at position i and zeros elsewhere.

This is not a great solution as these vectors are very high dimensional and all of them are orthogonal to each other, even the vectors corresponding to synonymous words! In other words, this method does not capture the semantic relations between words!

For this reason, other word representation (or word embedding) methods have been developed which try to map related words (words that occur near each other in a corpus of texts) to nearby vectors in the Euclidean space. This way, since computers can understand vector similarity, they can understand relatedness of words as well. This gives a good approximation for word meanings.

Further reading:

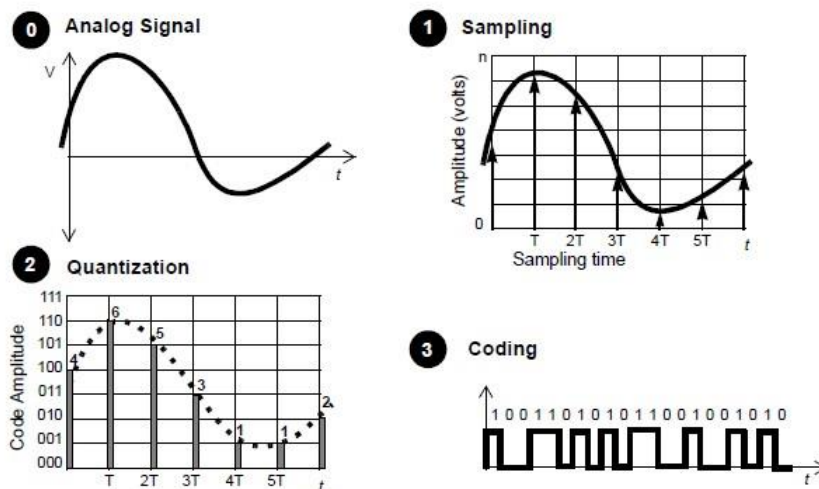


Efficient Estimation of Word Representations in Vector Space

by T Mikolov · 2013 · Cited by 44526 — We propose two novel model architectures for computing continuous vector representations of words from very large data sets.

Voice and audio:

Audio amplitude is sampled at regular intervals. The samples are then quantized and encoded.



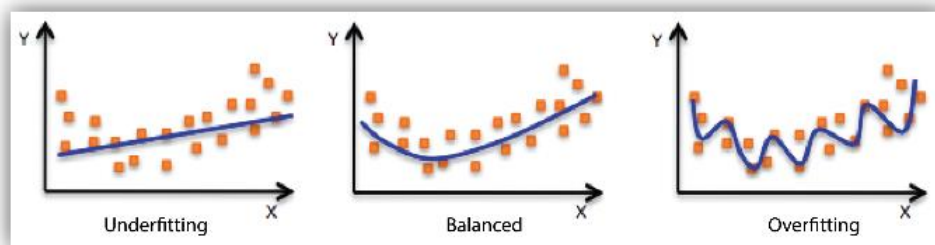
Video: video is just a series of pictures!

To recap, in *supervised machine learning* we have a bunch of data points $x_1, x_2, x_3, \dots, x_N$ each with a label y_i and we want to find a function $f(x)$ such that $f(x_i) \approx y_i$ for each i .

This would enable us to generalize to the case of x that is not in the training dataset.

AI as optimization

There are usually infinitely many functions which satisfy the condition $f(x_i) \approx y_i$ for $i = 1, \dots, N$.



For example, in the above image, you can see three different functions that approximate given data. (The meaning of image captions will be explained later.)

Therefore, we restrict ourselves to specific, *parametric families of functions* such as:

- **Linear functions:**

$$f(x) = ax + b$$

where a, b are the parameters.

- **Polynomial functions:**

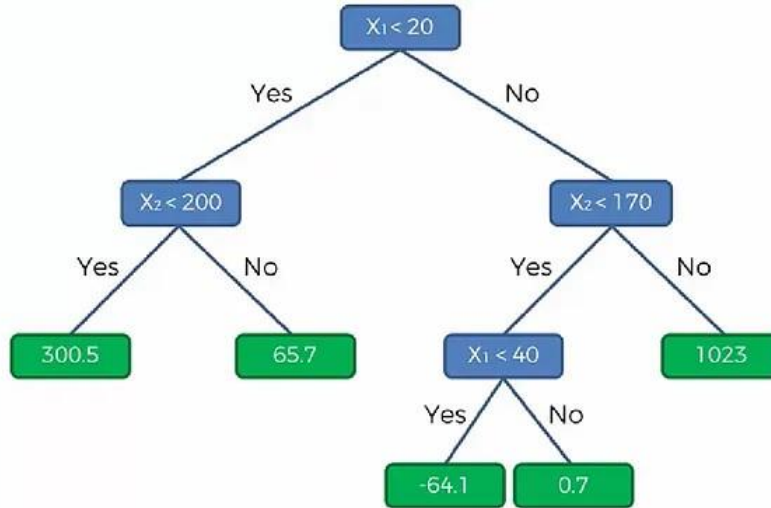
$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_dx^d$$

where the a_0, a_1, \dots, a_d are the parameters.

- **Logistic function:** It maps the real numbers into [0,1].

$$\sigma(x) = \frac{1}{1+e^x}$$

- Functions given by **decision trees:**



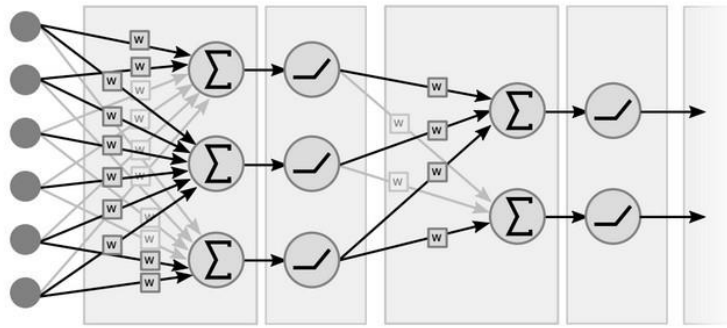
More complicated function families based on *neural networks*. We'll talk about them later in the course, because neural networks and deep learning is a big subject. However, we take a glimpse at them here.

The building block of a neural network is called a *perceptron* or *artificial neuron* which is a function of the form

$$\phi(x_1, x_2, \dots, x_d) = h(a_1x_1 + a_2x_2 + \dots + a_dx_d + a_{d+1})$$

where h is a nonlinear function (such as the sigmoid function above) and a_1, a_2, \dots, a_{d+1} are the parameters. A neural network is obtained by feeding the output of several perceptrons to another artificial neuron, and so on:

$$\phi(\phi_1(x), \phi_2(x), \dots, \phi_d(x)).$$



Neural networks are very effective for working with raw data types such as images, text and audio and are behind many of the achievements of AI, mentioned above.

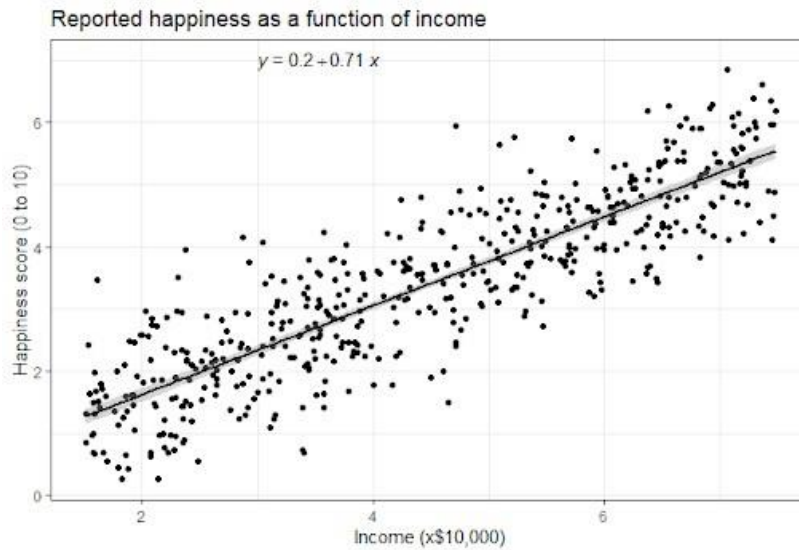
Let's denote a parametric function with $f_w(x)$ where w is the set of parameters, e.g. $w = (a, b)$ for linear regression.

Once such a family of functions $\{f_w(x)\}$ with parameter set w (also called a *model*) is chosen, finding the function f is reduced to an optimization problem: find the parameters w such that the average difference between $f_w(x_i)$ and y_i (for all (x_i, y_i) in the data) is minimized! Various optimization methods are used to solve such problems, the easiest one being using the derivative.

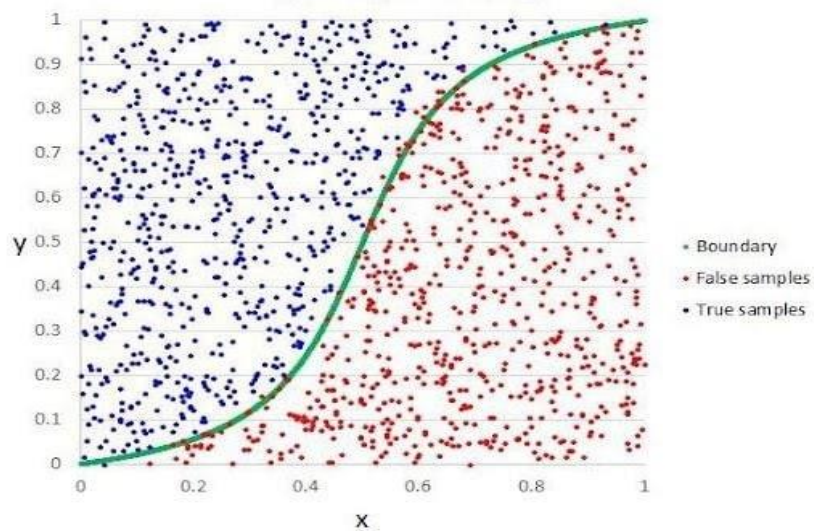
For the families of functions mentioned above, the corresponding models are called *linear regression*, *polynomial regression*, *logistic regression* and *decision tree learning*, respectively.

Low Code Machine Learning Course

Linear and logistic regression are the most fundamental ML models.



An example of linear regression: estimating reported happiness score as a function of income.



Logistic regression is used for classifying items.

The number of parameters used in a model, i.e. the length of the vector w is very important. It ranges from 2 for linear regression to more than a trillion for GPT 4.

Low Code Machine Learning Course

Note that some ML models are *probabilistic*, i.e. they compute the probability of the answer y given the data x , or in probabilistic notation: $P(y|x)$.

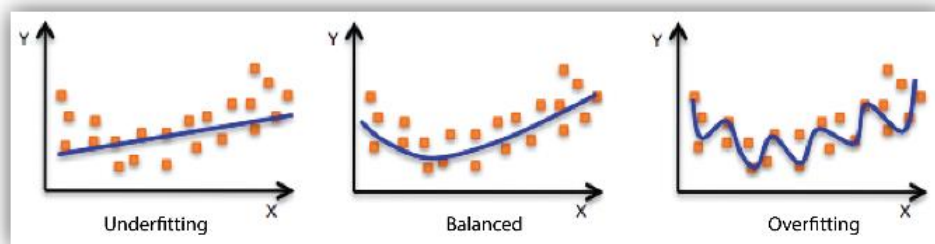
For example *Large Language Models (LLMs)* behind chatbots like ChatGPT can estimate the probability that a word would follow a sequence of words. This enables them to complete prompts, answer questions and generate text.

How an ML model is evaluated

We described intelligence as ability to generalize from learned knowledge. In evaluating an ML model, we need to see how well it can generalize to data it has not seen.

For this reason, we split our data D into two parts. One part (typically 70% to 90% of the data) for *training* the model, i.e. for solving the optimization problem mentioned above, and finding the function $f(x)$.

The rest is used for *testing* the model. This means that we evaluate the function $f(x)$ from the last step on the test data that "it has not seen" and evaluate how well it can predict the labels y_i .



The model on the right of this picture (which may come from a high degree polynomial) estimates (fits) the data very well but would not generalize well to unseen data. This is because the model has learned all the noise and irrelevant features of the data. This is called *overfitting*.

Machine Learning vs Data Science

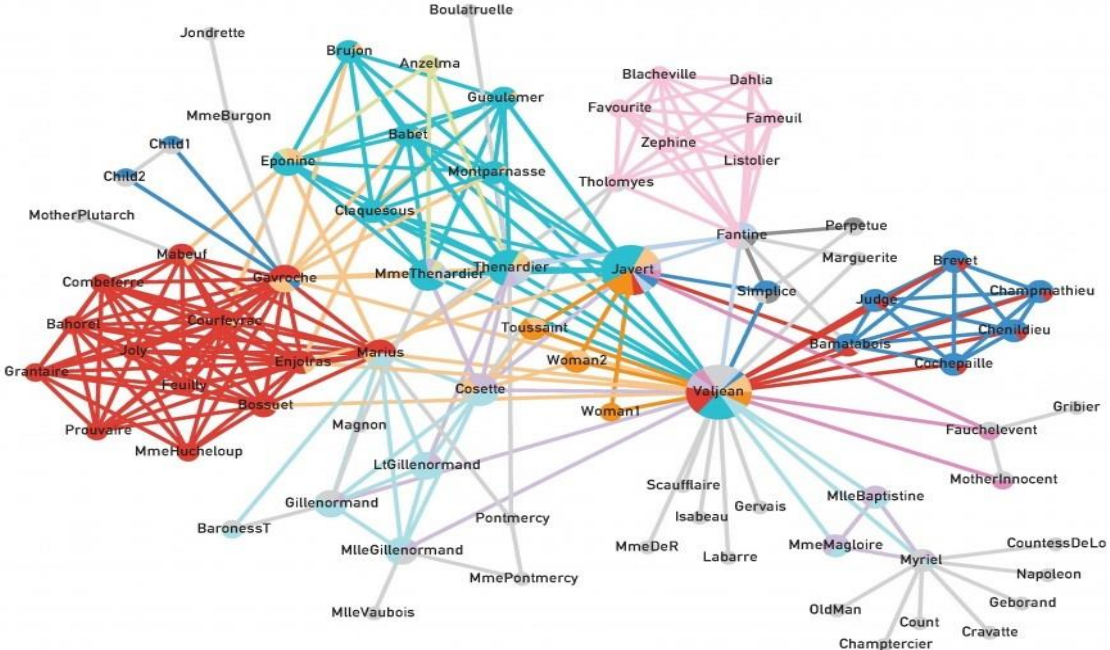
Even though both ML and Data Science deal with data, in Data Science the emphasis is on analyzing data and obtaining insights from it, whereas in ML the emphasis is on developing and analyzing methods for learning from data.

Unsupervised Learning

What we talked about so far is *supervised learning* in which the labels y_i are provided. In *unsupervised learning*, the labels are not provided and we try to optimize a problem defined on the features $\{x_i\}$ alone.

For example:

- We may want to group (or *cluster*) the data points together, in such a way that similar instances are grouped together. This is called *clustering*.
- We may have a network (such as a social network) and find the *communities* in the network, i.e. groups of individuals which are well-connected to each other but less connected to the rest of the network. This is called *community detection*.



Low Code Machine Learning Course

An example of community detection in the network of characters in the novel *Les Miserables*. (More on network analysis [here](#).)

As another example, the problem of *word representation* mentioned above is also unsupervised because it tries to assign similar vectors to words that occur together in texts often, and the meanings of the words are not given to the system in advance. It only uses the co-occurrence of words in sentences!

In each of the examples above we have an optimization problem too. We try to optimize a function that evaluates the fitness of the clusters or the communities found.

Unsupervised learning is actually more fundamental than supervised learning, because most of the data we have is unlabeled, and labeling data is expensive and time consuming. A great portion of what we humans learn happens in an unsupervised way too.

Manifold learning

Another problem in unsupervised learning is *dimension reduction* or *manifold learning*. The basic idea is that data is often represented by more coordinates (features) than needed. This is a special case of the scientific principle that *coordinates are not canonical*.

For example:

- In an image of a natural world, the value of a pixel is usually similar to the pixels around it. And this is behind image compression techniques such as JPEG.
- A matrix can be put in a canonical form and be represented by far fewer numbers.

Remember I said we represent each type of data as vectors in some Euclidean space R^N .

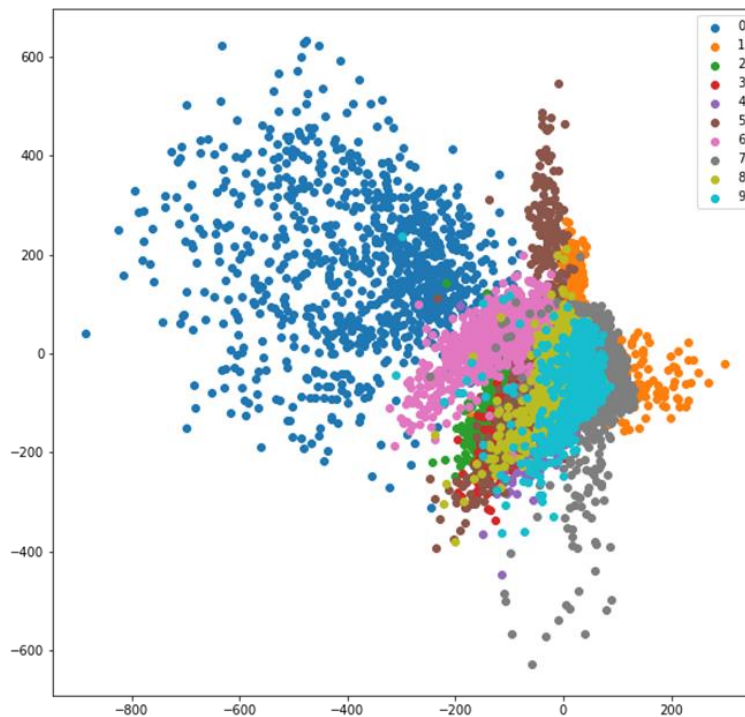
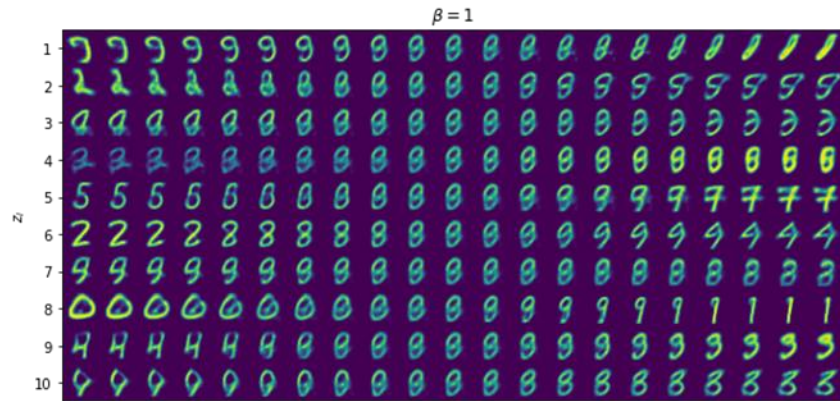
For example:

$$\Phi : \text{NaturalPictures} \rightarrow R^N.$$

However, as argued above, the image of Φ usually has a dimension n much smaller than N . In other words, most of the elements in R^N correspond to random and meaningless pictures.

Low Code Machine Learning Course

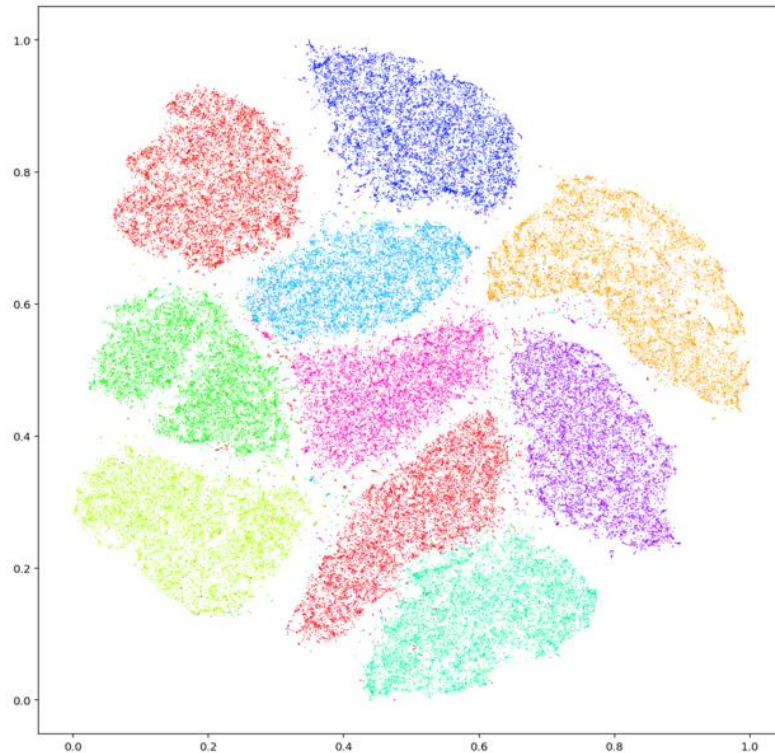
In dimension reduction and manifold learning, we try to reduce the dimension of this data from N to n . Or in other words, find alternative coordinates on the image of Φ , instead of using the coordinates induced from \mathbb{R}^N .



A 2-parameter family of handwritten digit images. In the top image you see the actual number shapes, and how one digit shapes morphs into another. In the bottom image, the distribution of digit images across the two resulting coordinates are depicted. Here the dimension has been reduced from the number of pixels in the images ($28 \times 28 = 784$) to only 2. [Source](#)

Low Code Machine Learning Course

Below is another low-dimensional projection of the same dataset of digit images, obtained using another algorithm called t-SNE. It has clearly separated the points corresponding to different digits, even though it had no knowledge of the labels.



Yann LeCun likens unsupervised learning to the bulk of the AI cake and supervised learning to the icing on the case.

- **"Pure" Reinforcement Learning (cherry)**
 - ▶ The machine predicts a scalar reward given once in a while.
 - ▶ **A few bits for some samples**
- **Supervised Learning (icing)**
 - ▶ The machine predicts a category or a few numbers for each input
 - ▶ Predicting human-supplied data
 - ▶ **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
 - ▶ The machine predicts any part of its input for any observed part.
 - ▶ Predicts future frames in videos
 - ▶ **Millions of bits per sample**

■ (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

Reinforcement learning

We have not talked about the aspect of intelligence that involves action and behavior. This is the subject of Reinforcement Learning (RL) which tries to maximize the rewards of an agent in an environment, for example a chess player. In this course we do not focus on RL.

ML versus science

As discussed above, Machine Learning models learn complex relations between the independent variables (also called features) and the dependent variable (also called response variable). And they do this by means of examples given to them. More precisely an ML method finds the model that can best approximate the training data, within its family of models.

This is as opposed to what science does: science starts from some underlying principles and tries to explain the world.

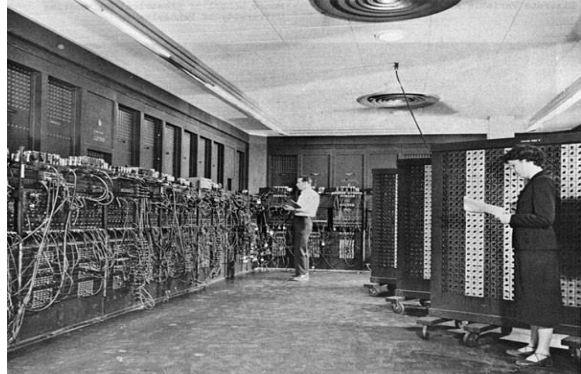
Comparing ML to the scientific method:

Aspect	Scientific Method	Machine Learning
Purpose	To understand underlying principles and causality.	To optimize predictive models from data.
Starting Point	Hypothesis-driven.	Data-driven.
Approach	Theory and controlled experimentation.	Model training and error minimization.
Explanation vs. Prediction	Focuses on explaining phenomena (causal inference).	Focuses on prediction accuracy (correlation).
Iteration	Theory is refined over time based on evidence.	Model is iteratively improved based on feedback from data.
Data	Data is used to test or validate theories.	Data is used to train models and optimize predictions.
Error Handling	Errors and uncertainties quantified in experiments.	Errors are reduced by optimizing model parameters.

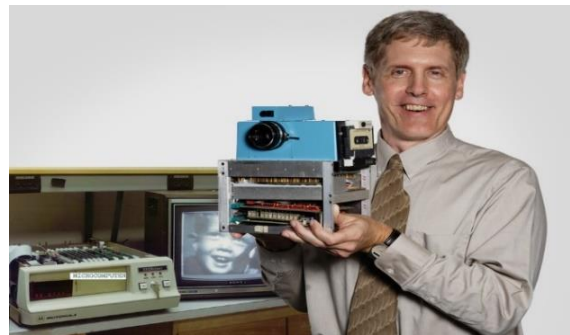
There are also attempts at combining the two, such as [Physics-aware neural networks](#).

The lifecycle of a technology: how AI is becoming mainstream

- First computer (ENIAC, 1945)



- First digital camera with a resolution of 0.01 megapixels!

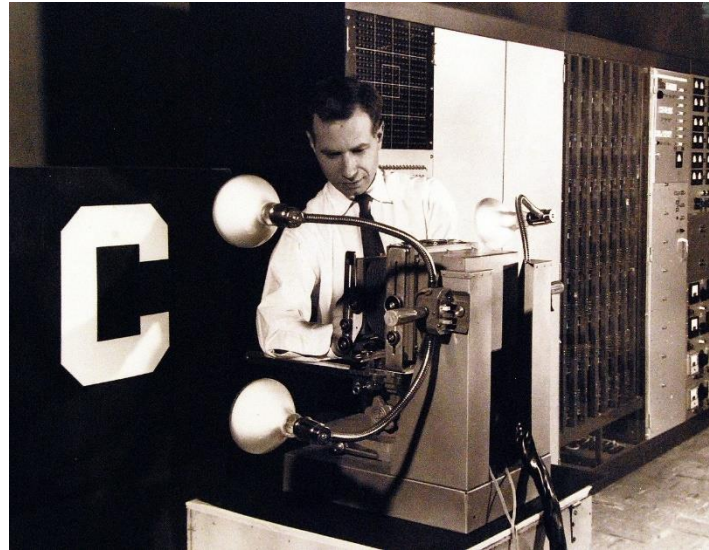


- First cellphone!



Low Code Machine Learning Course

But nowadays everybody has a camera, a cellphone and a computer, all in one package, in his or her pocket! Similarly the throughput of wireless systems has been increasing (3G, 4G, 5G,...), the capacity of magnetic disks has been increasing, etc. (As a side, this is related to the *improvement rate of a technology*. Some technologies improve faster than others. See e.g. [this paper](#).)



The same is true about AI. At first (1960s to 1980s) it was a rather esoteric kind of science.

The Perceptron AI machine in 1958. The same machine about which NY Times had written a headline.

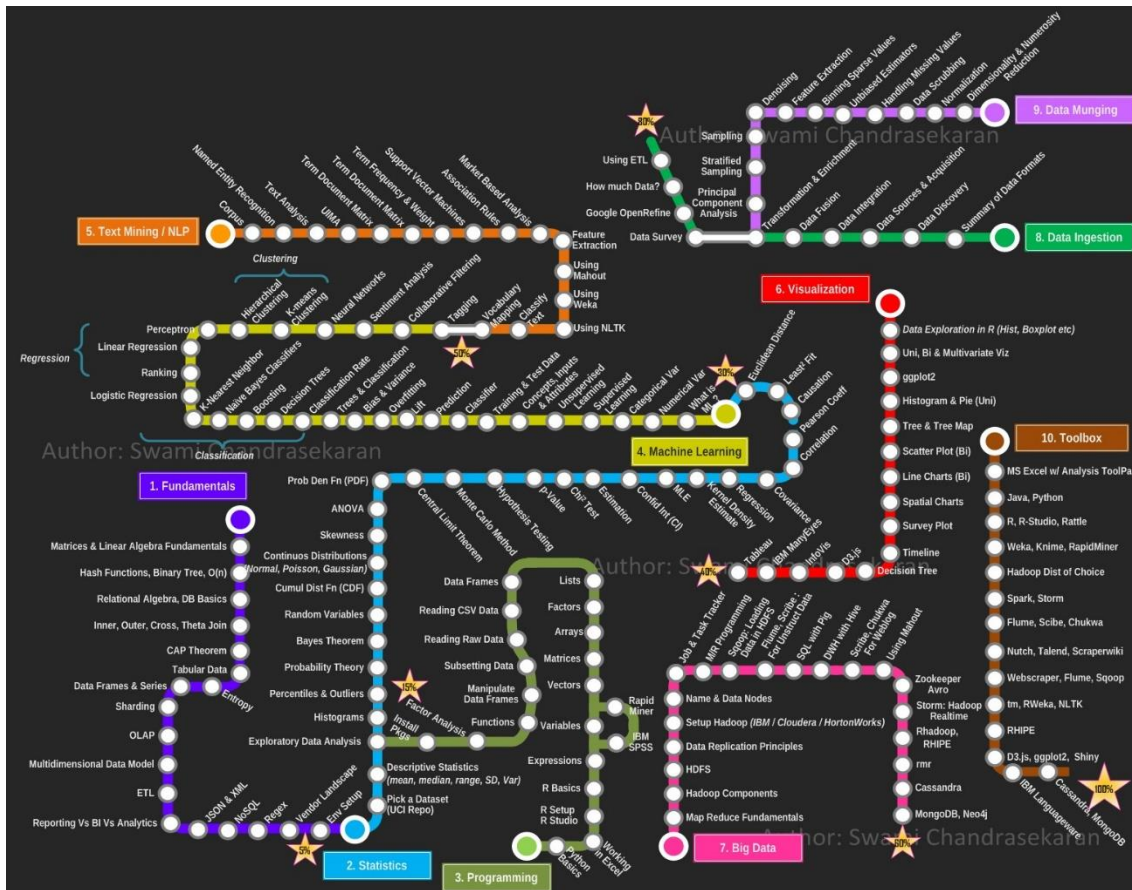
Even around the year 2000 there were no easy-to-use software packages or programming libraries for working with AI and researchers had to code models from scratch.

For several years now, we have had great programming tools that help us work with ML.



Low Code Machine Learning Course

Now AI has come to mainstream by cellphone apps such as voice recognition and chatbots like ChatGPT. However it's still tough to learn ML: you have to learn Pandas, Scikit, Matplotlib, Pytorch or TensorFlow,...



The roadmap for learning data science by [Swami Chandrasekaran](#).

Technology democratization is the process by which a technology rapidly becomes more accessible to more people. This is happening for AI and ML too and they become more and more accessible to the layman.

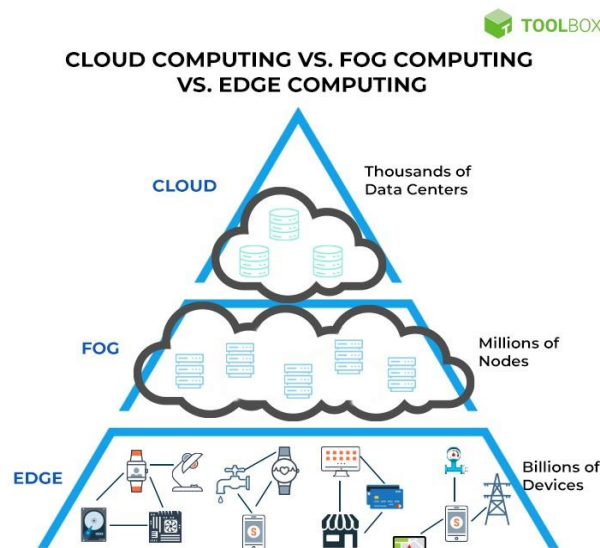
Citizen data scientist

Citizen Data Scientists are power users who can perform both simple and moderately complicated analytical tasks that would previously have required more technical expertise.

Further reading: [The Risks of Empowering “Citizen Data Scientists”](#)

Edge AI

Edge AI is the practice of running AI models on the "edge" devices such as cellphones and laptops, instead of cloud computing clusters and super computers.



For many AI tasks such as voice recognition, data has to be transferred to the cloud, be processed there and then the result be transferred back to the edge device. However more efficient AI models make it possible to run them on the edge devices themselves. For example ChatGPT (from 2022) has 175 billion parameters but the language model Phi-3-mini (from 2024) has only 3.8 billion parameters and its performance is far better than what the number of its parameters would suggest.

Further reading: [Forget ChatGPT: why researchers now run small AIs on their laptops](#)

Forget ChatGPT: why researchers now run small AIs on their laptops

Artificial-intelligence models are typically used online, but a host of openly available tools is changing that. Here's how to get started with local AIs.

By [Matthew Hutson](#)

Introducing AutoML

AutoML or Low Code ML tools make it possible to do machine learning with very little programming. Various AutoML tools have been developed:



In this course we make use of an AutoML tool called PyCaret for teaching you machine learning. It is a Python library.



The problems we study in this course

For the practical problems we study in this course you either need to use Google Colab, or install [Python](#) (3.11), [Pycaret](#) and [VSCode](#) on your own machine. Pycaret can be installed by running the following command in terminal:

```
pip install pycaret[full]
```

The jupyter notebooks for this course are available [on github](#).

Remember that in supervised learning, when the labels y_i are continuous, we call it a *regression problem* and when the labels are discrete, we call it a *classification problem*.

Binary classification

When we have only two classes, the problem is called a *binary* classification problem. Otherwise it is called *multi-class classification*.

Predicting diabetes based on patient's data such as age, number of pregnancies, etc.

Regression

Predicting health insurance charges based on a person's data such as age, gender, BMI, etc.

Multi-class classification

Predicting iris species based on flower features.

Detecting handwritten digits.

Clustering

Clustering is an unsupervised problem of grouping data points according to their similarity to each other.

Forecasting time series

Predicting the number of airline passengers based on historical data.

