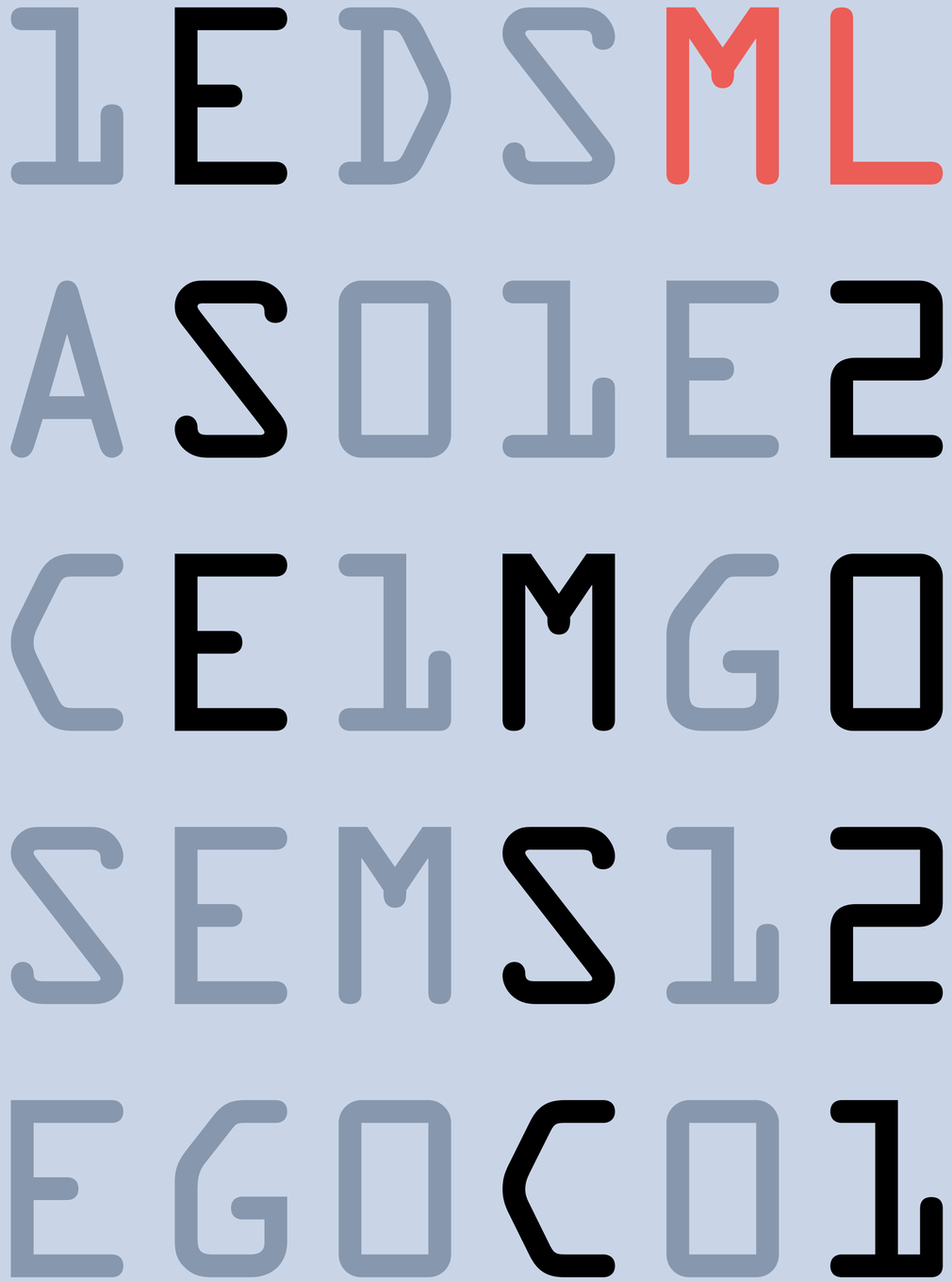


Machine Learning module

Lluis Guasch
George Strong
Carlos Cueto
Deborah Pelacani Cruz
Yao Jiashun
Raul Adriaensen
Alexander Campbell

Course structure

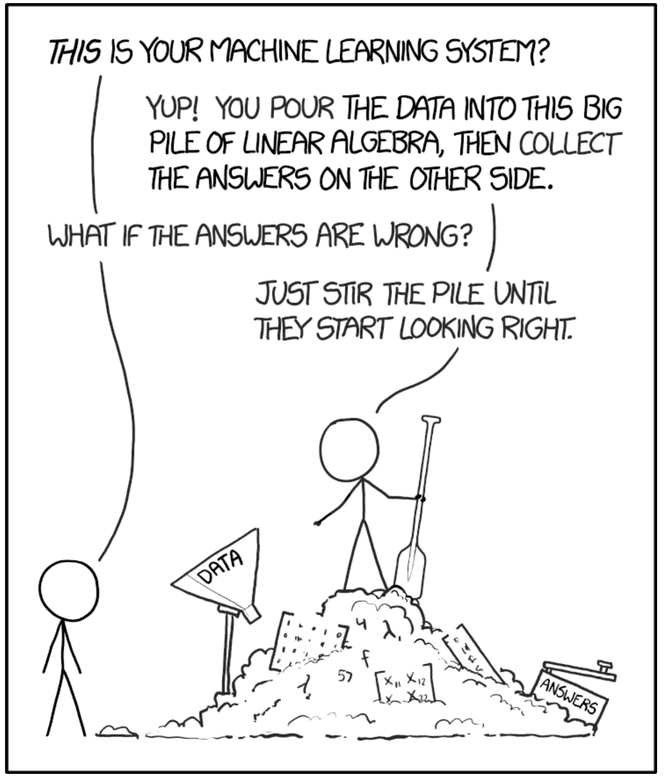
https://github.com/ese-msc-2021/ML_module

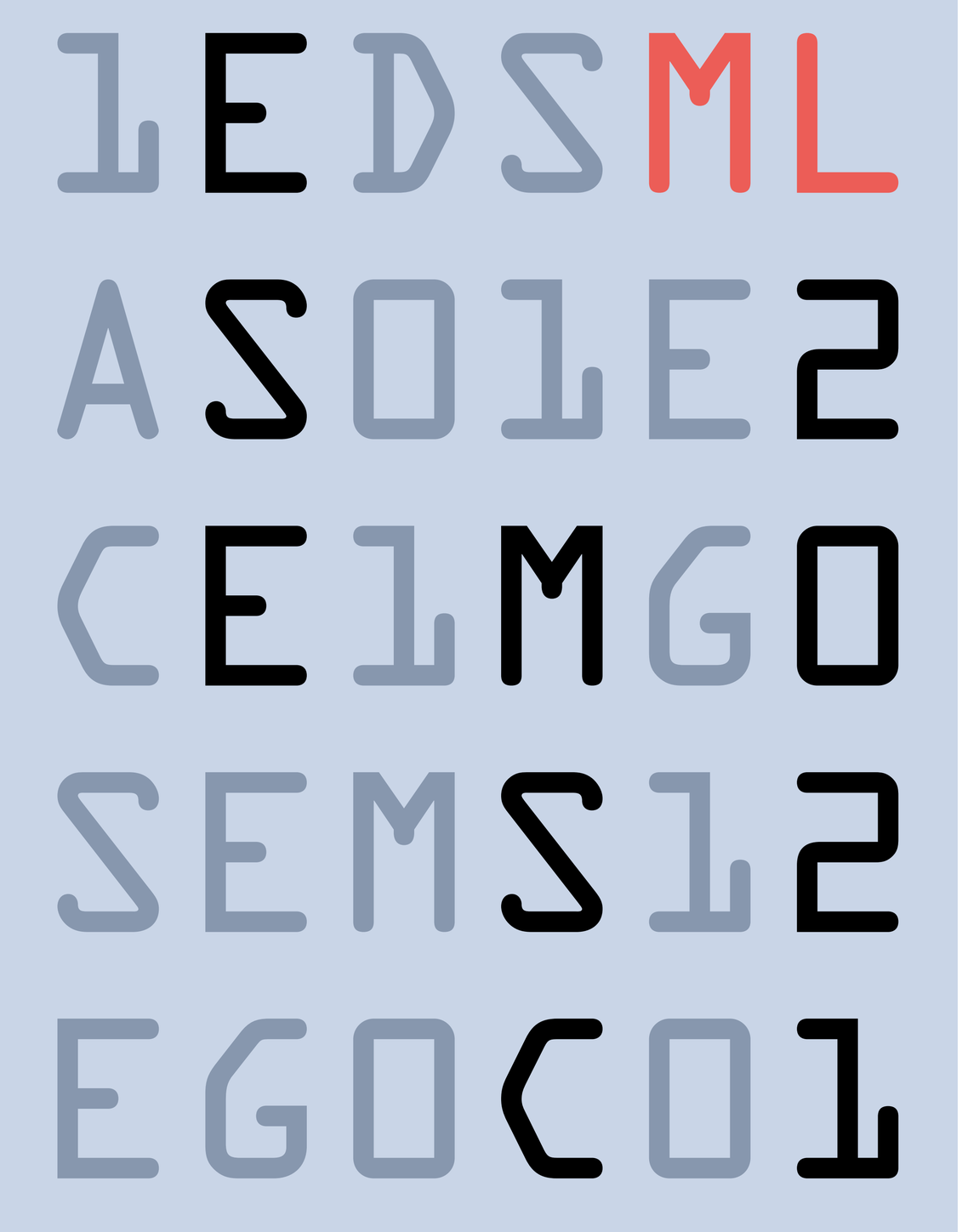
Teaching Team

- George Strong
- Carlos Cueto
- Deborah Pelacani Cruz
- Yao Jiashun
- Raul Adriaensen
- Alexander Campbell
- Lluis Guasch

Course Outcomes

- Understand well the principles of Machine and Deep Learning.
- Gain insights on what methods work on which situations, and why.
- Don’t treat ML as a black box.





1-Introduction to Machine Learning

Lluis Guasch

GTAs online:
Debbie
Raul

1. **What is ML?**
2. Unsupervised VS supervised learning
3. Linear regression
4. Logistic regression
5. k-Means and PCA

STATE OF THE ART

DALL·E 2: https://openai.com/dall-e-2/

‘An astronaut lounging in a tropical resort in space in a vaporwave style’

Societal Impact of ML

A recent study by researchers from Imperial (March 2022):

AI tool accurately predicts tumour regrowth in cancer patients

Exclusive: Tool predicts how likely tumours are to grow back in cancer patients after they have undergone treatment

reported by The Guardian

A comparison of machine learning methods for predicting recurrence and death after curative-intent radiotherapy for non-small cell lung cancer: Development and validation of multivariable clinical prediction models

Sumeet Hindocha,[a,b,c,d,k] Thomas G. Charlton,[e] Kristofer Linton-Reid,[d] Benjamin Hunter,[a,c,d,k] Charleen Chan,[c] Merina Ahmed,[f] Emily J. Robinson,[g] Matthew Orton,[h] Shahreen Ahmad,[e] Fiona McDonald,[a,c] Imogen Locke,[f] Danielle Power,[i] Matthew Blackledge,[j] Richard W. Lee,[a,k,l] and Eric O. Aboagye [c]**

Godfathers of Machine Learning

Yann LeCun, Geoffrey Hinton, and Yoshua Bengio:

Turing award:

https://www.youtube.com/watch?v=HzilDIhWhrE

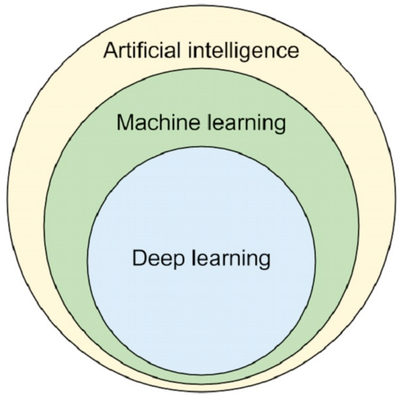
AAAI (Association for the Advancement of Artificial Intelligence) award:

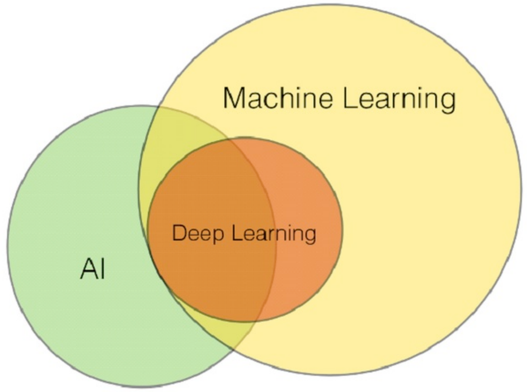
https://www.youtube.com/watch?v=UX8OubxsY8w

ML DEFINITION

From wikipedia:

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data.





Different interpretations of ML in relation to AI

A more formal definition:

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

Tom M. Mitchell

ML DEFINITION

Artificial intelligence:

Turing, A.M. (1950). Computing machinery and intelligence. Mind, 59, 433-460.

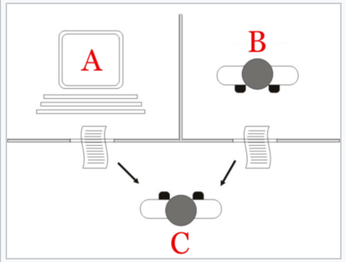
COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The

Turing test



The "standard interpretation" of the Turing test, in which player C, the interrogator, is given the task of trying to determine which player – A or B – is a computer and which is a human. The interrogator is limited to using the responses to written questions to make the determination.[1]

‘Turing’ (Voight-Kampff) test in Blade Runner

ML APPLICATIONS

Finance: predict patterns at short and long scales

Health: triage and diagnostic, image segmentation, automated data handling, and a long etc

Retail: Offer personal recommendations based on recent choices of products

Marketing: Organize customers into classes based on their past behavior

Automotive: self-driving vehicles

Media/Advertising: Generate images of non-existing people for synthetic scenes

Art: synthetic generation of music, text or paintings in the “style” of an existing artist

…and **Science**, of course

almost all applications rely on large amounts of data

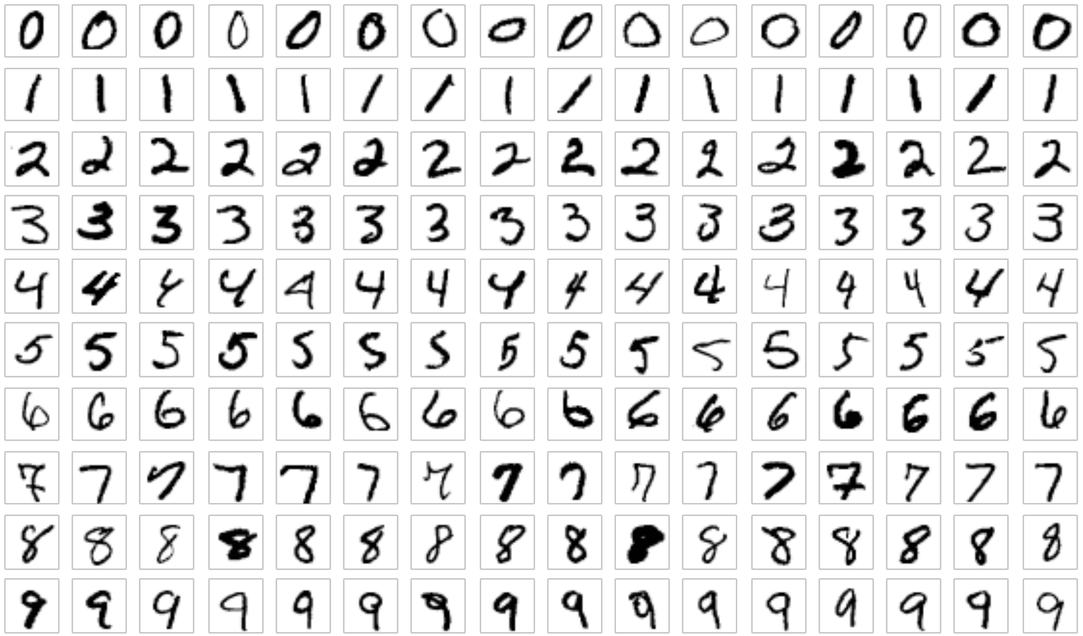
DATA VOLUMES

The data explosion behind machine learning:

- ~90% of the world's data was created in the last five years
- ~10^{18} bytes of data are created on the Internet every day
- In 2 days, as many data are created as from the beginning of history to 2003

MNIST Dataset

Classic ML example: the **MNIST** dataset



Problem:

Based on a Training Set of 60,000 labelled handwritten digit images (**the experience E**), learn an algorithm that automatically recognizes which digit a new unseen image represents (**the Task T**).

Performance P measure will quantify the accuracy of the algorithm on a Test Set of 10,000 images.

1. What is ML?
2. **Unsupervised VS supervised learning**
3. Linear regression
4. Logistic regression
5. k-Means and PCA

Supervised VS Unsupervised Learning

Supervised learning algorithms experience a dataset containing features, but each example is also associated with a label or target.

For example, the Iris dataset is annotated with the species of each iris plant. A supervised learning algorithm can study the Iris dataset and learn to classify iris plants into three different species based on their measurements.

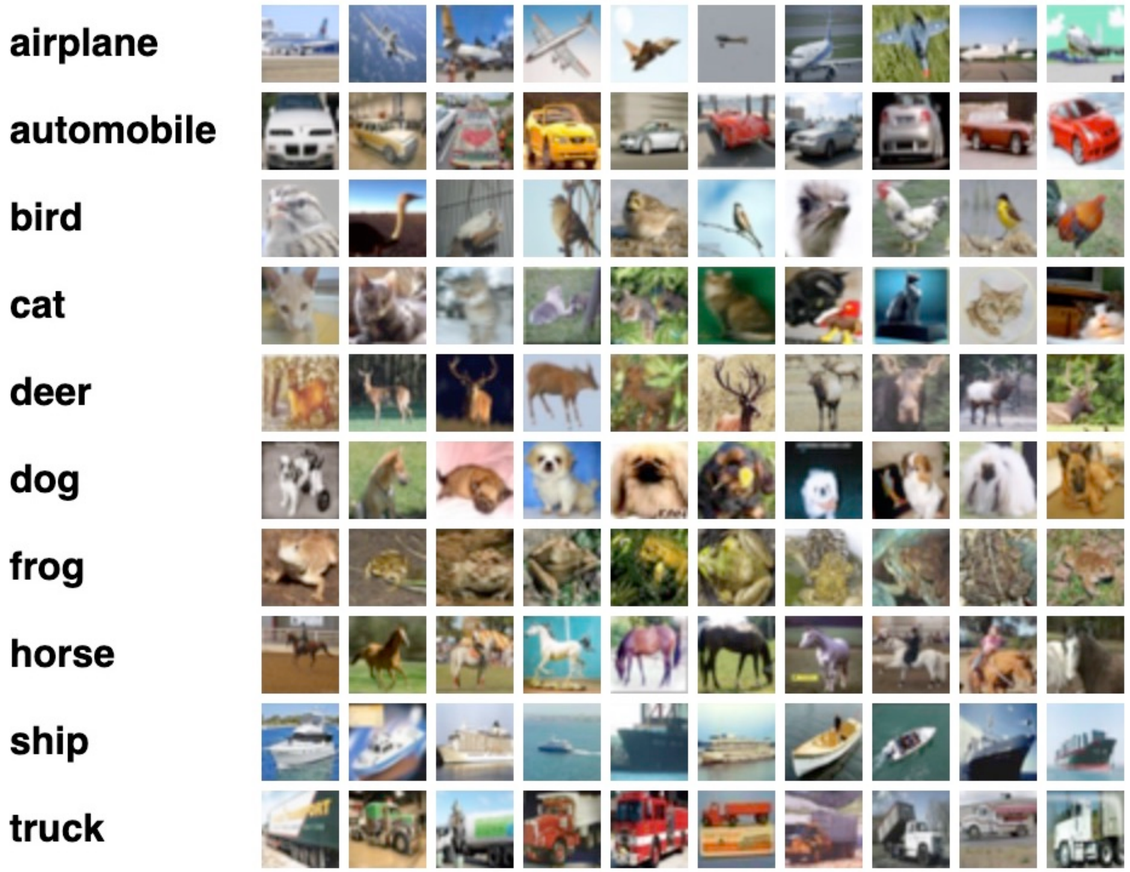
Unsupervised learning algorithms experience a dataset containing many features, then learn useful properties of the structure of this dataset.

In the context of deep learning, we usually want to learn the entire probability distribution that generated a dataset, whether explicitly, as in density estimation, or implicitly, for tasks like synthesis or denoising. Some other unsupervised learning algorithms perform other roles, like clustering, which consists of dividing the dataset into clusters of similar examples.

definitions from the book Deep Learning by Goodfellow et al, 2016

Supervised Learning examples

Classification:

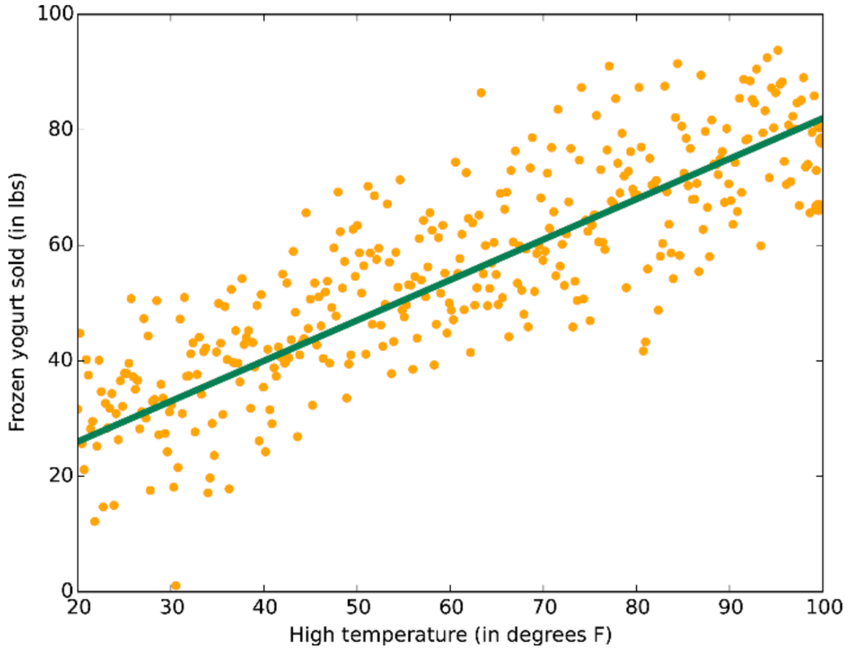


CIFAR-10 dataset

Task: classify new images

Supervised Learning examples

Regression:

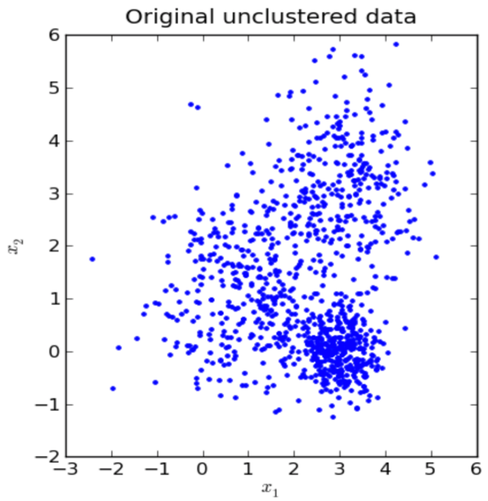


Task:

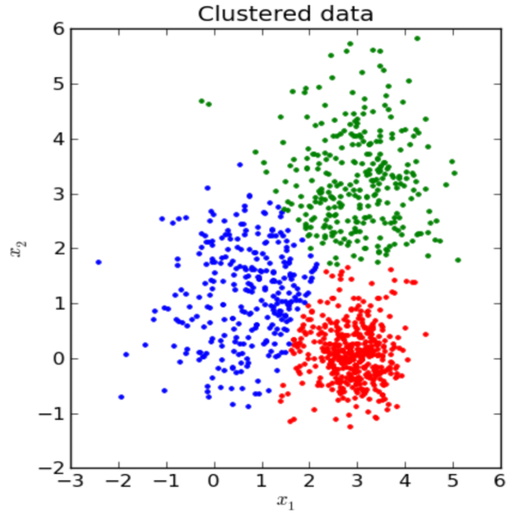
From the Training Set of pairs (Temperature, sales), the **Experience**, establish a formula for predicting sales when only the Temperature is known such that the averaged squared error is minimized (**Performance).**

Unsupervised Learning

Clustering:



Data does not have labels



Task:

Organise data (x1,x2) into clusters according to some pre-defined structure, so that new data points can be assigned to one of the clusters.

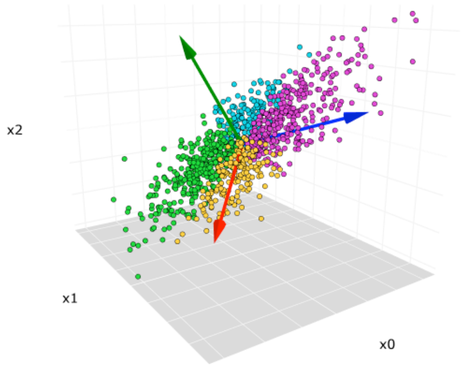
Examples:

- Sort costumers into different categories in order to target them by marketing
- Among many press articles, identify those which deal with the same story
- From a satellite image, identify classes of regions where the data seem to behave in a “similar” fashion

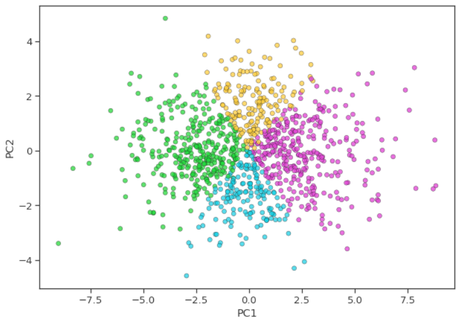
Unsupervised Learning

How do we explore structure in the data?

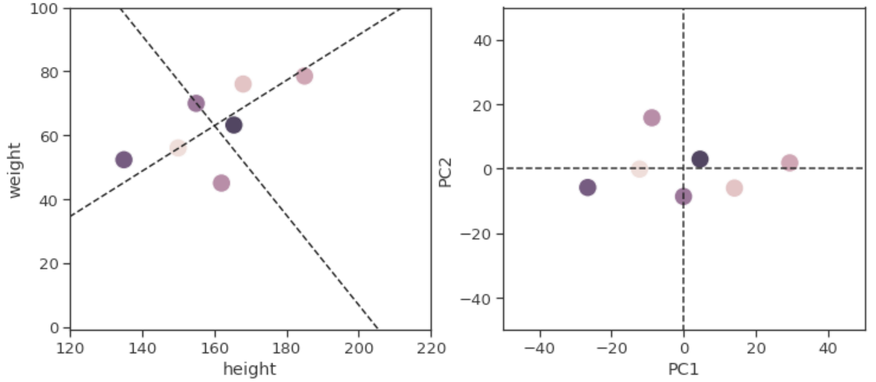
PCA: Principal Component Analysis



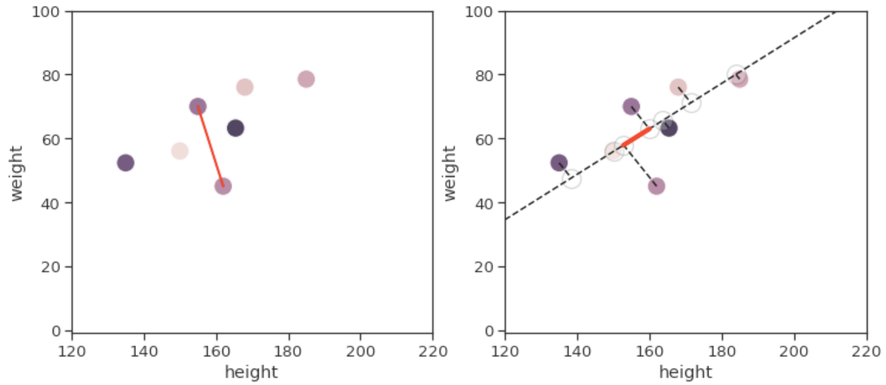
reduce dimensionality



2D (data) to 2D (2 components): **rotation**

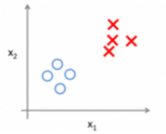


2D (data) to 1D (1 component): **projection**



SUPERVISED VS UNSUPERVISED LEARNING

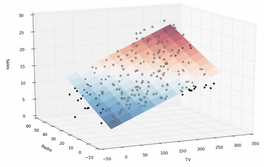
Summary:



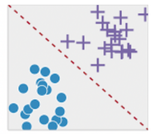
SUPERVISED LEARNING

Training Set: Data x with labels y

- **Regression**: label is a real number/vector

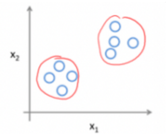


- **Classification**: label is a class or integer number/vector

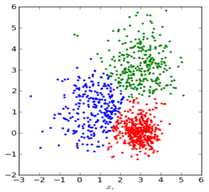


UNSUPERVISED LEARNING

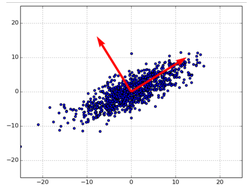
Training Set: Just Data x



- **Clustering**: organise data in groups

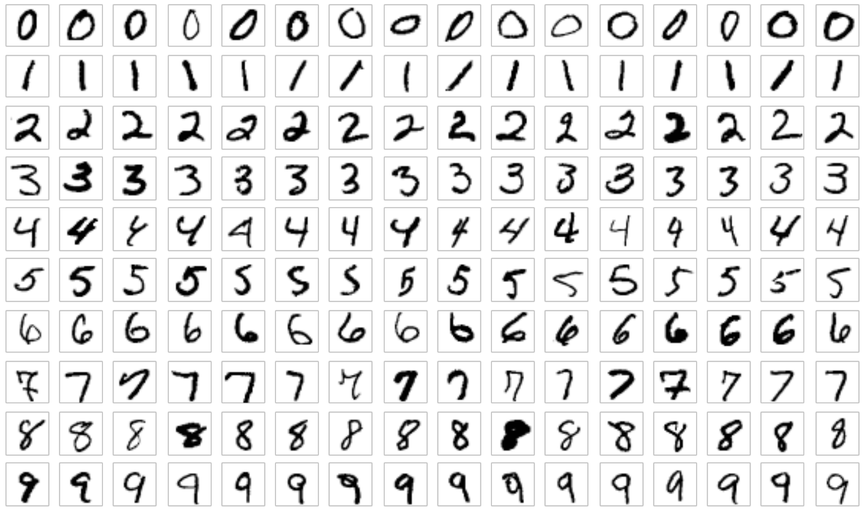


- **Dimensionality reduction**: compress data



Learning from Data

Labelled datasets can be used in unsupervised learning too. For example, MNIST:



SUPERVISED LEARNING:

Learn by using the images and their corresponding labels to be able to predict new labels on unseen data.

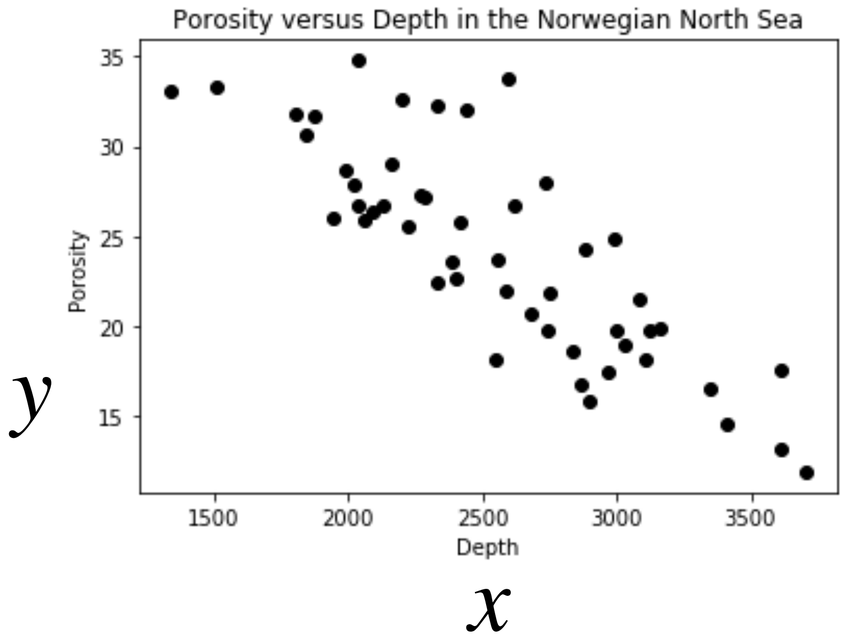
UNSUPERVISED LEARNING:

Ignoring labels, learn to classify the digits in different classes or groups according to the inherent structure in the images.

1. What is ML?
2. Unsupervised VS supervised learning
3. **Linear regression**
4. Logistic regression
5. k-Means and PCA

Linear regression is a form of **supervised learning**

'Standard' machine learning terminology and notation:



- m = number of training examples (here we have m=50 pairs of data)
- x = input variables/**features** (here x is depth)
- y = output variables/**target** variables (here y is porosity)
- x^i , y^i = the i^{th} training sample pair

What does Linear Regression do?

Fit a linear function to the data. In Machine Learning terminology, this function is called the **Hypothesis** $h_{\boldsymbol{\theta}}(\mathbf{x})$

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 \mathbf{x}$$

$\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are called the **parameters** (or the weights) of the model.

For each value of the feature x^i in the Training Set, we want $h_{\boldsymbol{\theta}}(x^i)$ to be close to the target y^i

In other words, we wish to minimize the **Cost Function** (or **Loss function**) $J(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$:

$$J(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \frac{1}{2m} \sum_{i=1}^{m} (y^i - h(x^i))^2 = \frac{1}{2m} \sum_{i=1}^{m} (y^i - \boldsymbol{\theta}_0 - \boldsymbol{\theta}_1 x^i)^2$$

Minimising the Cost Function

Let's assume that the bias term $\boldsymbol{\theta}_0$ is zero (for simplicity and without loss of generaility).

The cost function then becomes:

$$J(\boldsymbol{\theta}_1) = \frac{1}{2m}\sum_{i=1}^{m}(y^i - \boldsymbol{\theta}_1 x^i)^2 = \boldsymbol{\theta}_1^2(\sum_{i=1}^{m}(x^i)^2) - 2\boldsymbol{\theta}_1(\sum_{i=1}^{m}(x^i y^i) + (\sum_{i=1}^{m}(y^i)^2)$$

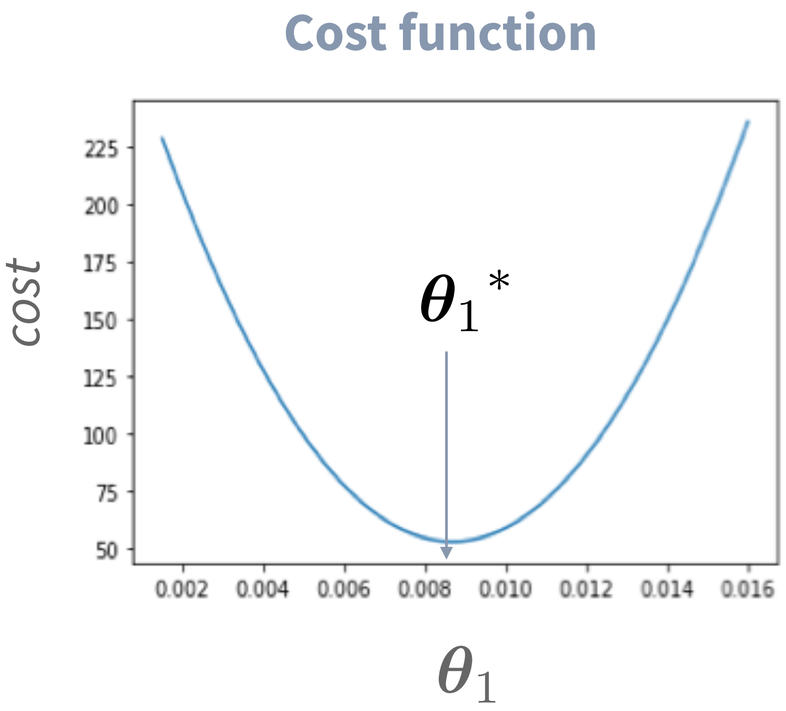
which as a **minimum** at:

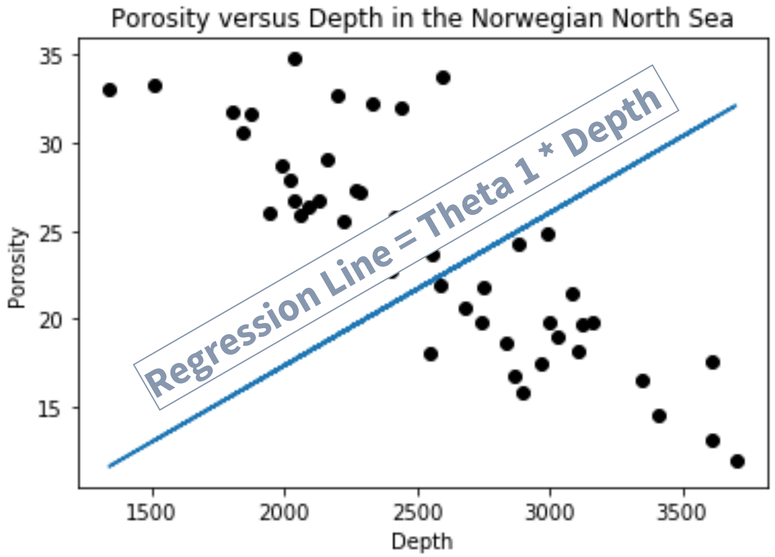
$$\frac{\partial J(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1} = 0 \longrightarrow \boldsymbol{\theta}_1{}^* = \frac{\sum_{i=1}^{m} x^i y^i}{\sum_{i=1}^{m}(x^i)^2}$$

normal equation

Minimising the Cost Function

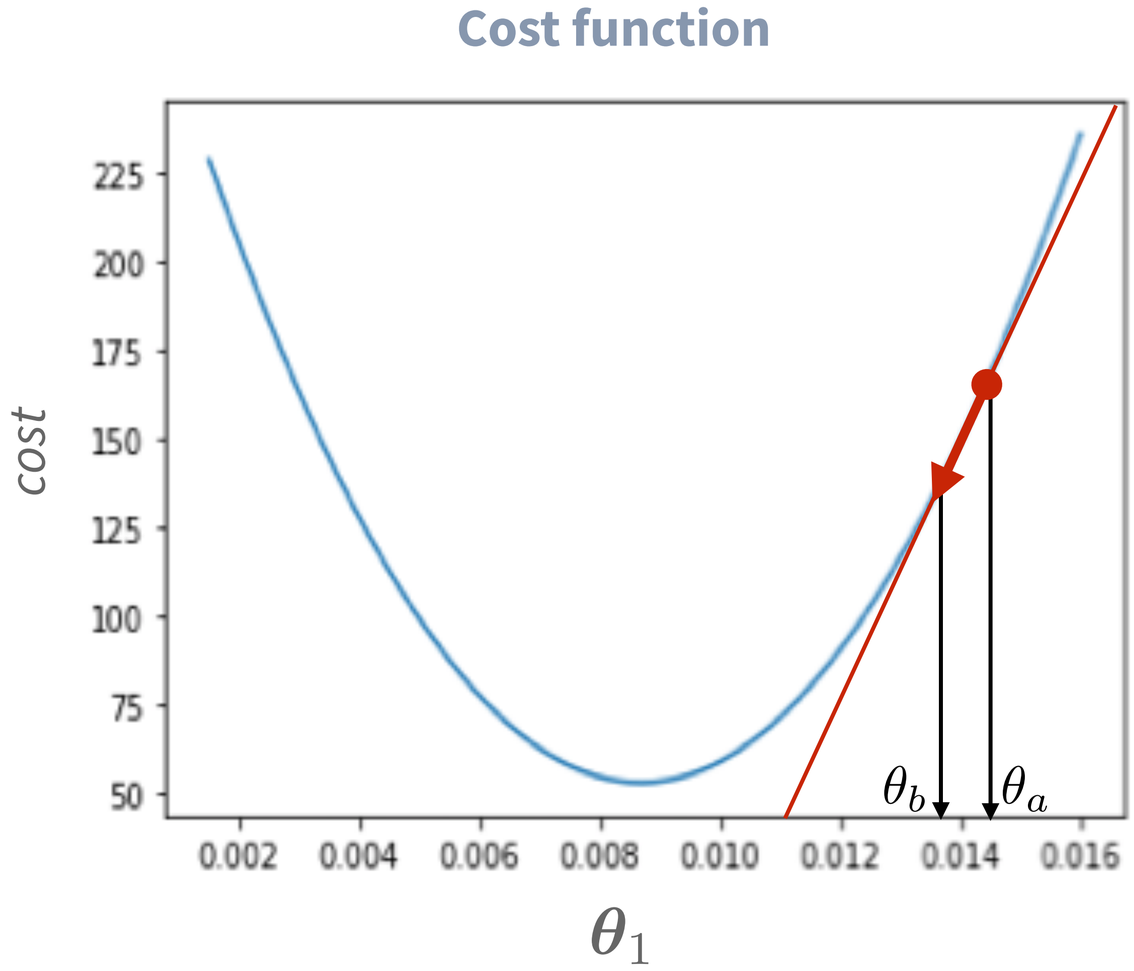
Let's assume that the bias term θ_0 is zero (for simplicity and without loss of generaility).





Gradient Descent

From before, the solution is when the derivative of the cost with respect to the parameter vanishes, but the derivative of the cost also gives us the slope of the curve at the point where it's calculated:



$$\theta_b = \theta_a \overset{!}{-} \alpha \frac{\partial J(\theta_a)}{\partial \boldsymbol{\theta}}$$

Learning rate

EXPRESSIONS INCLUDING BIAS TERM

If we add a bias term, the expression for the cost function and the values of the parameters are:

$$J(\boldsymbol{\theta_0},\boldsymbol{\theta_1}) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - \theta_0 - \theta_1 x^{(i)}\right)^2 = \frac{1}{2m}\sum_{i=1}^{m} y^{(i)2} + \frac{\boldsymbol{\theta_0^2}}{2} + \boldsymbol{\theta_1^2}\left(\frac{1}{2m}\sum_{i=1}^{m} x^{(i)2}\right)$$
$$-2\boldsymbol{\theta_0}\left(\frac{1}{2m}\sum_{i=1}^{m} y_i\right) + 2\boldsymbol{\theta_0}\boldsymbol{\theta_1}\left(\frac{1}{2m}\sum_{i=1}^{m} x_i\right) - 2\boldsymbol{\theta_1}\left(\frac{1}{2m}\sum_{i=1}^{m} x^{(i)}y^{(i)}\right)$$

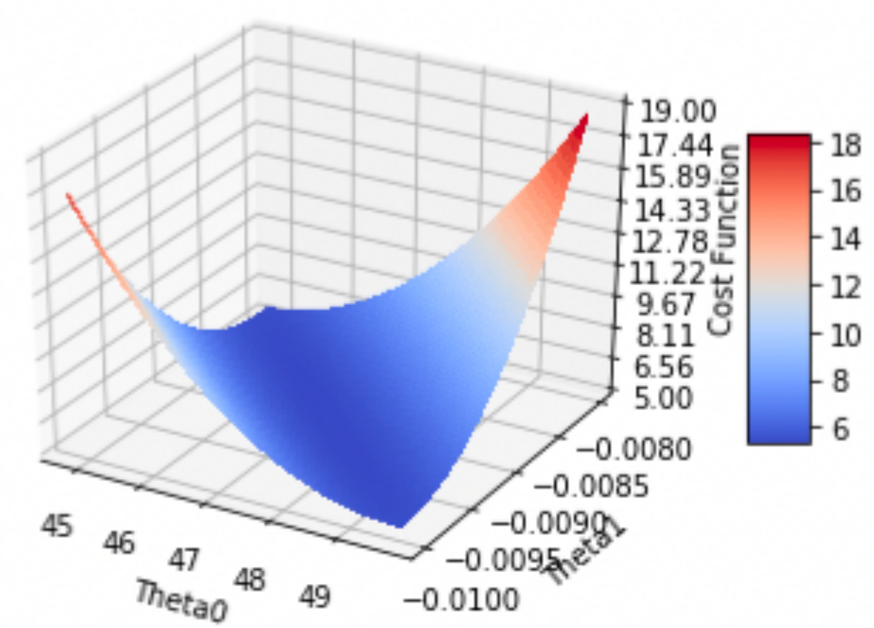
The cost function is simply a second degree polynomial in θ_0 and θ_1. It reaches its minimum for the values of $\theta_0$and θ_1 given by the so-called normal equations

$$\theta_0^* = \frac{\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)2}\right)\left(\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\right) - \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right)\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}y^{(i)}\right)}{\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)2}\right) - \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right)^2} \quad \theta_1^* = \frac{\frac{1}{m}\sum_{i=1}^{m} x^{(i)}y^{(i)} - \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right)\left(\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\right)}{\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)2}\right) - \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right)^2}$$

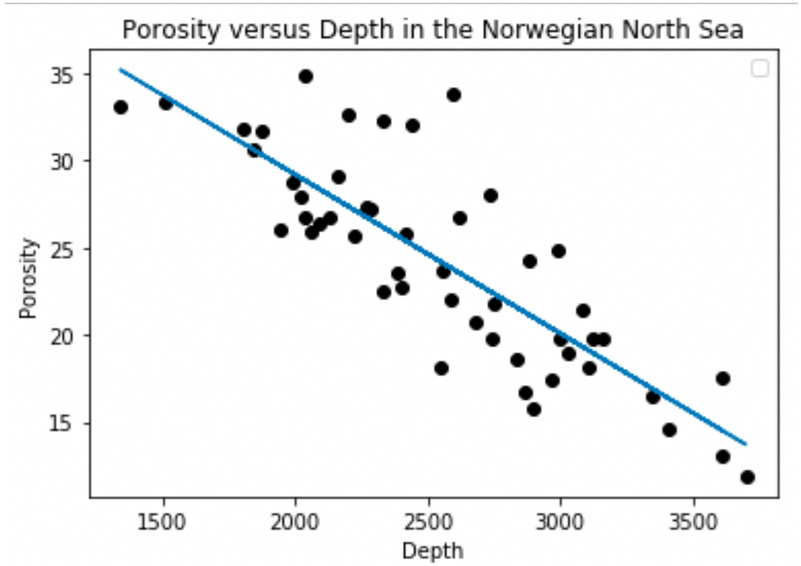
Expressions including Bias term

And now we can fit our data much better (thanks to the bias term):

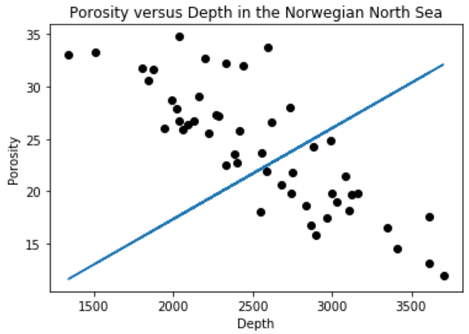
since this is a linear program there is only **1 minimum**, which is the solution:



with bias term:

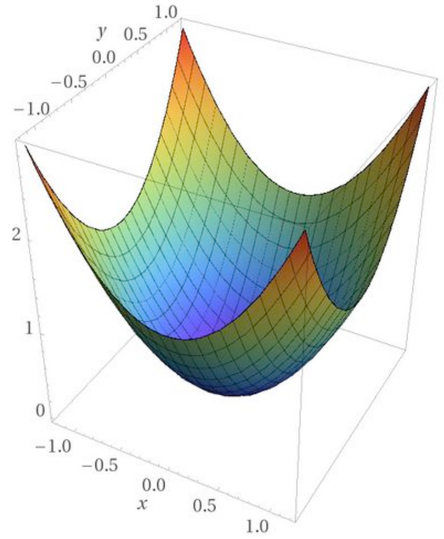


without bias term:



Convex VS non-Convex

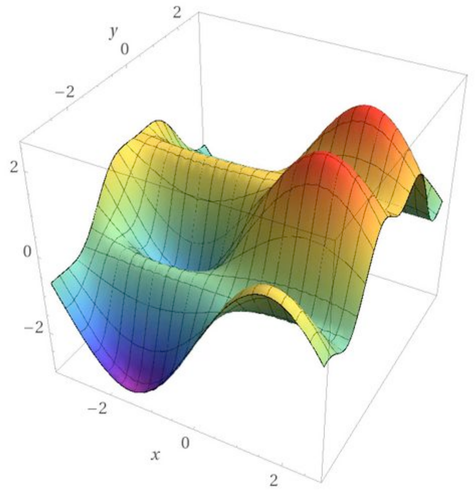
Linear relations between the hypothesis and the parameters result in quadratic cost functions:



Convex

(no local minima, only one solution)

But most real-world problems do not present this linear relation:



Non-convex

Loss Landscapes

If you want to have a play with loss landscapes:

https://losslandscape.com/

Multiple Linear Regression

From here:

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 \mathbf{x}$$

to here:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 \boldsymbol{x}_1 + \boldsymbol{\theta}_2 \boldsymbol{x}_2 + \cdots + \boldsymbol{\theta}_n \boldsymbol{x}_n$$

- n is the number of features in each data sample
- x_i is the value of the feature i
- y is the value of the target

Example of Multiple Linear Regression:

Predict house prices from size, location, number of bedrooms, etc…

VECTOR NOTATION

If we have n input features $\boldsymbol{x}_i$:

$$y = h_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 \boldsymbol{x}_1 + \boldsymbol{\theta}_2 \boldsymbol{x}_2 + \cdots + \boldsymbol{\theta}_n \boldsymbol{x}_n$$

with:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_n \end{bmatrix}$$

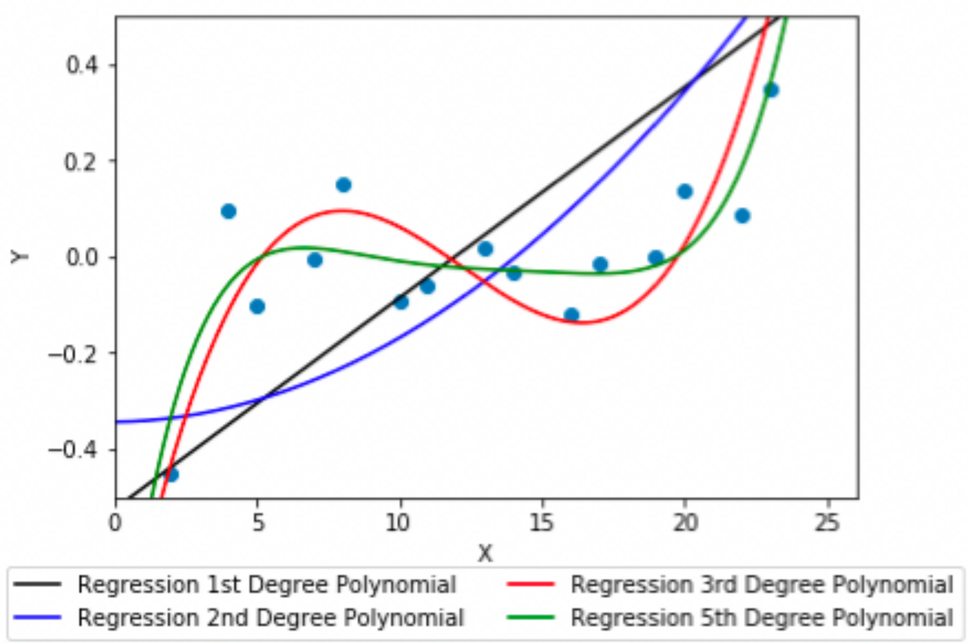
where each vector (for example $\boldsymbol{x}_1$) has m features

This vector notation allows us to write the hypothesis h as: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \mathbf{x}$

Non-linear Regression

Assume that the relation between $\boldsymbol{x}$ and h is not linear by using polynomials:

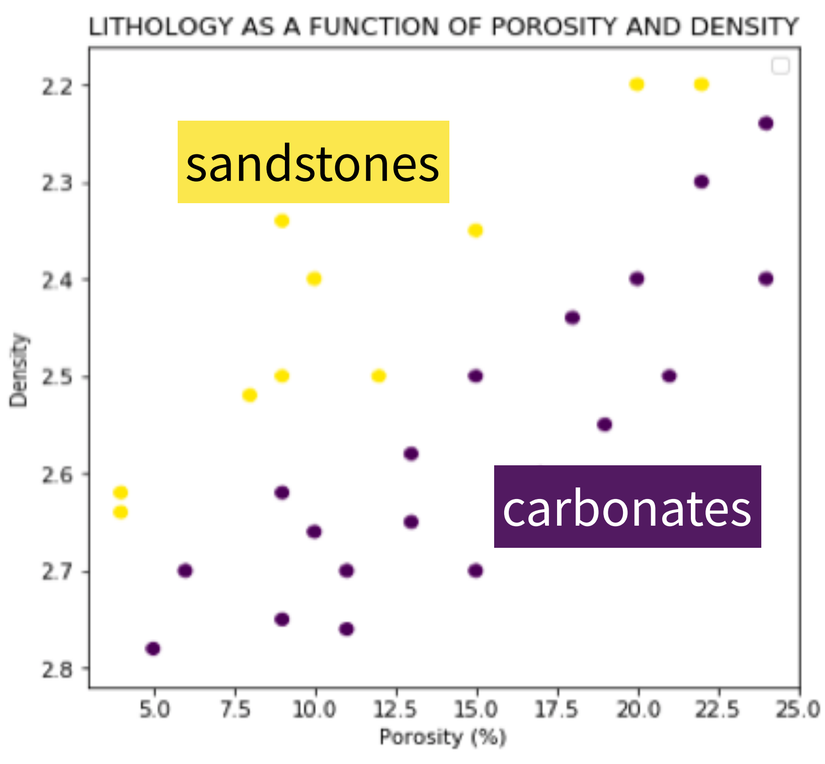
$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 \boldsymbol{x} + \boldsymbol{\theta}_2 \boldsymbol{x}^2 + \boldsymbol{\theta}_n \boldsymbol{x}^3 + \ldots$$



1. What is ML?
2. Unsupervised VS supervised learning
3. Linear regression
4. **Logistic regression**
5. k-Means and PCA

Logistic Regression

Example: predict Lithology from Density and Porosity ⟶ **Supervised Classification problem**



30 rock samples with Porosity, Density and Lithology (sandstone or carbonate) as a label.

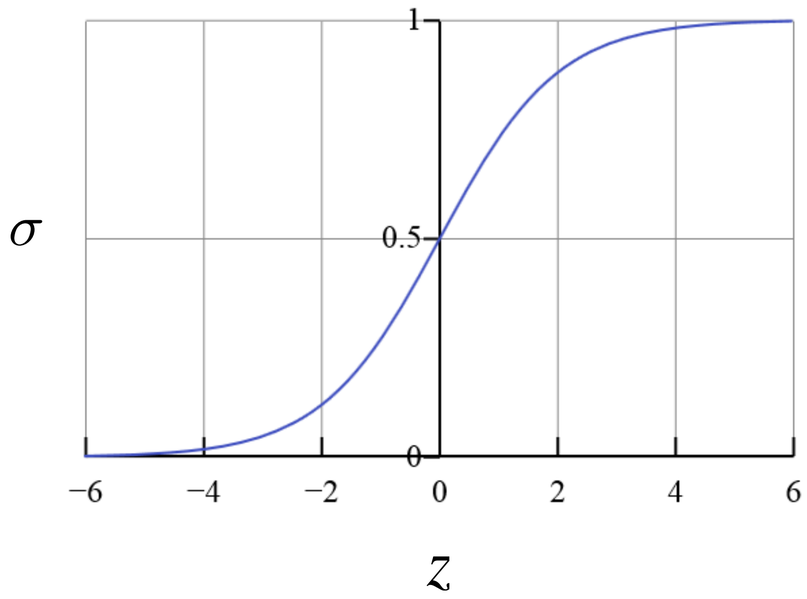
Why can't we use linear regression for this task (or multiple linear regression)?

The Sigmoid Function

We need to map real number pairs (porosity, density) to probability values (0,1)

The **sigmoid function** (aka the **logistic function**) takes any real value z and transforms it into a value between 0 and 1:

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



Important:

$$\sigma'(z) = (1 - \sigma(z))\sigma(z)$$

The Sigmoid Function

How to use the **sigmoid function**:

Take an output y from a regression equation with input $\boldsymbol{x}$:

$$y = \boldsymbol{\theta}^\top \mathbf{x}$$

where y can take any positive or negative value. Applying the sigmoid function to y we get a value between 0 and 1, which we can **interpret as the probability for the class to be 1**:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = P(y = 1 | \boldsymbol{\theta}, \boldsymbol{x}) = \sigma(\boldsymbol{\theta}^\top \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \boldsymbol{x}}}$$

Logistic Regression Cost Function

Cost function using for logistic regression:

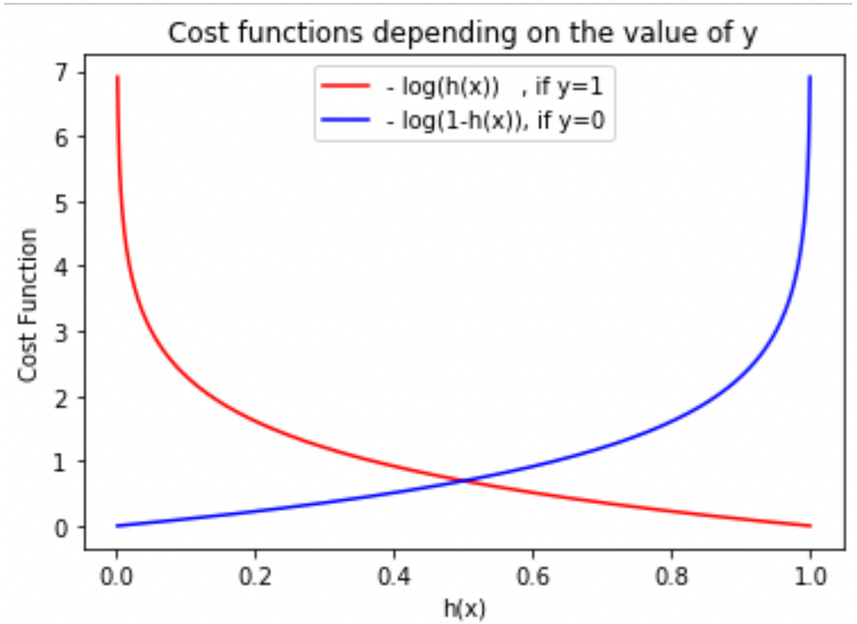
For one data point , how to measure the discrepancy between the actual data value y (equal to 0 or 1) and the estimated probability $h_{\boldsymbol{\theta}}(\boldsymbol{x})$?

if $(y = 1)$

$$C(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$

if $(y = 0)$

$$C(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -\log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$



combining the two into a single expression:

$$C(h_{\boldsymbol{\theta}}(\boldsymbol{x}), y) = -y \log(h_{\boldsymbol{\theta}}(\boldsymbol{x})) - (1 - y) \log(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x}))$$

Logistic Regression Cost Function

Cost function for a training dataset:

$$J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} C(h_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y^i) = -\frac{1}{m}\sum_{i=1}^{m} [y^i \log(h_{\boldsymbol{\theta}}(\boldsymbol{x}^i)) + (1-y^i)\log(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}^i))]$$

Minimise using gradient descent:

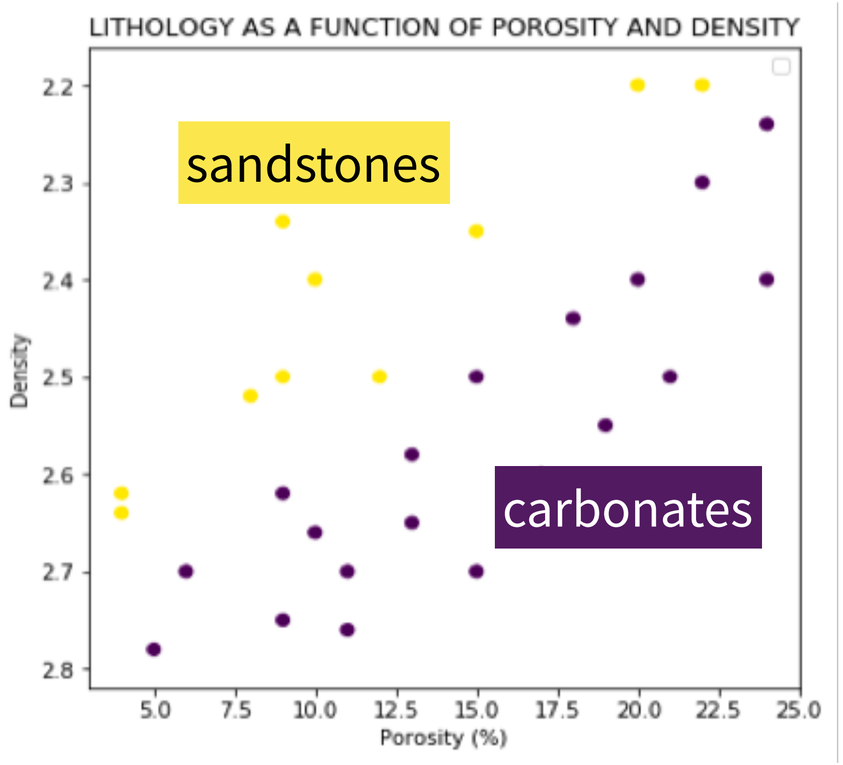
$$\frac{\partial J}{\partial \boldsymbol{\theta}_j} = \frac{1}{m}\sum_{i=1}^{m} [h_{\boldsymbol{\theta}}(\boldsymbol{x}^i) - y^i)]\boldsymbol{x}_j^i$$

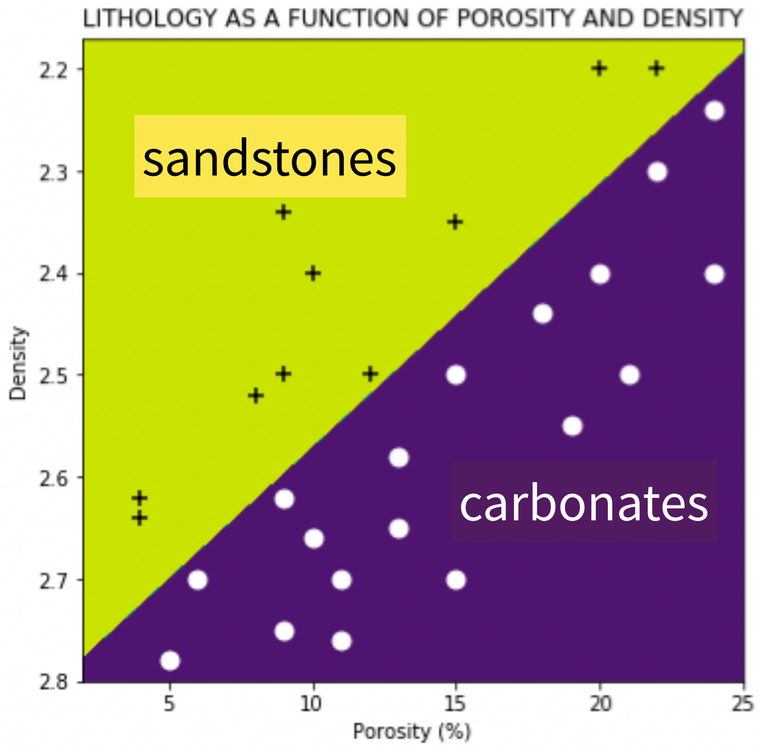
(afternoon exercise)

Logistic regression is used in Classification (not regression)

Logistic Regression in Action

Binary classification with logistic regression (with the previous lithology example):

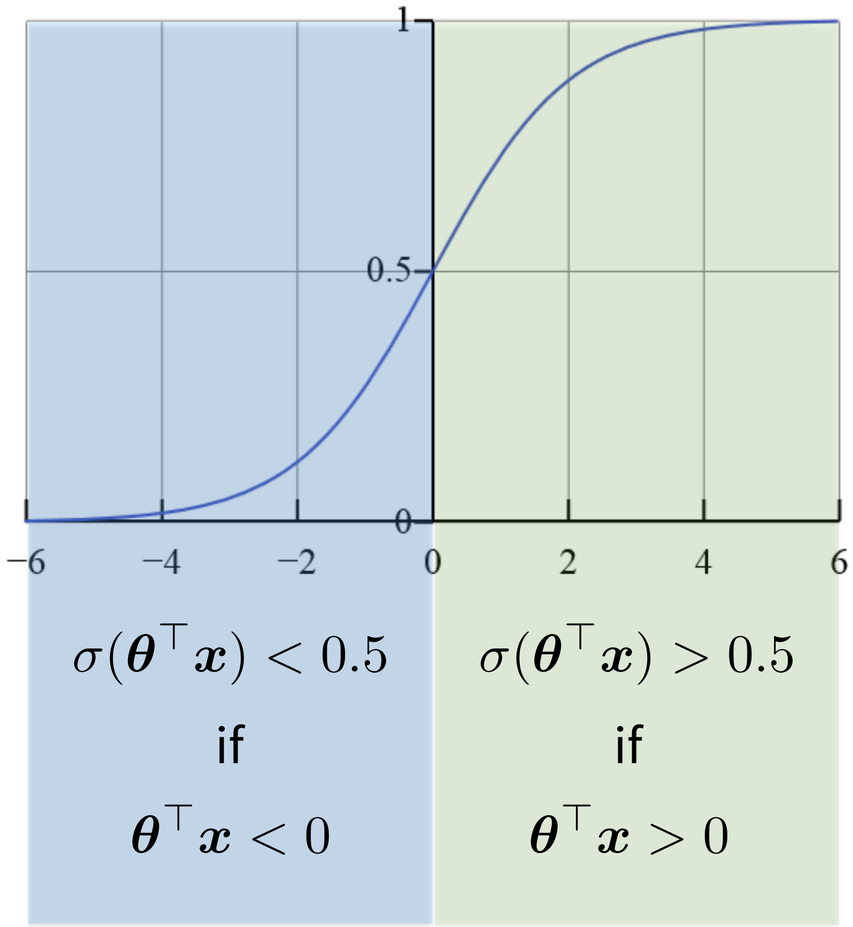




Logistic Regression in Action

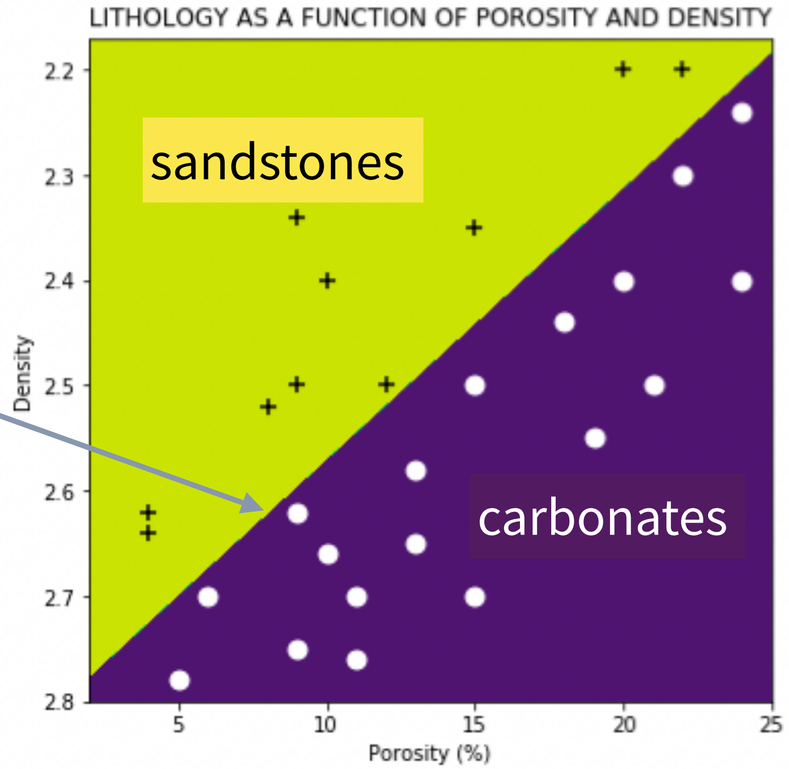
Definition of the **Decision Boundary**:

$$P(y = 1|\boldsymbol{\theta}, \boldsymbol{x}) = \sigma(\boldsymbol{\theta}^\top \boldsymbol{x})$$



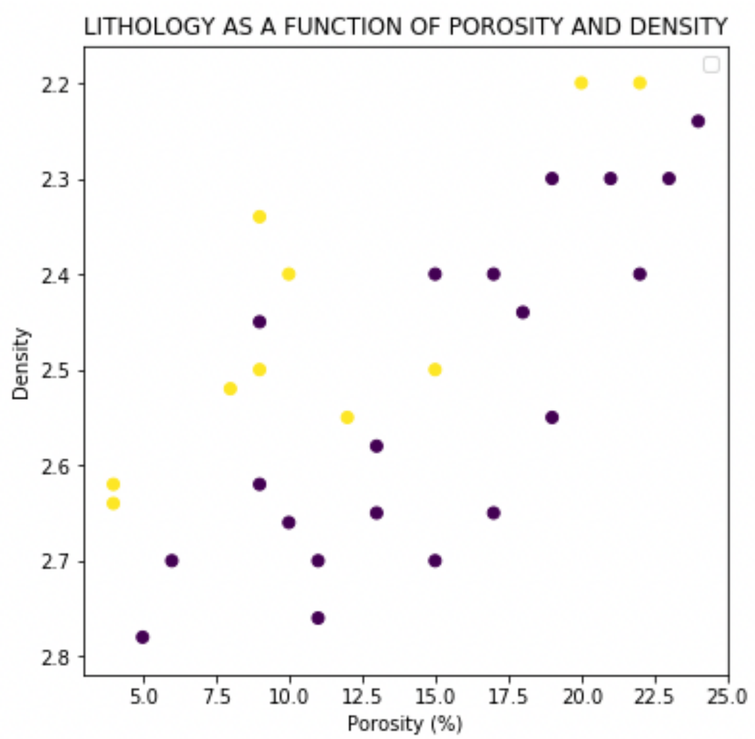
Decision boundary satisfies:

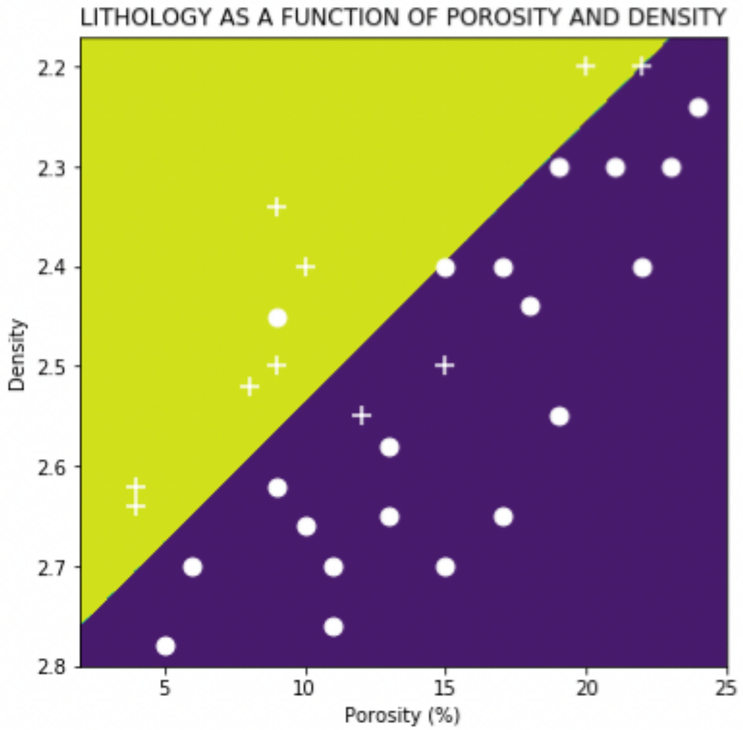
$$\boldsymbol{\theta}^\top \boldsymbol{x} = 0$$



Logistic Regression

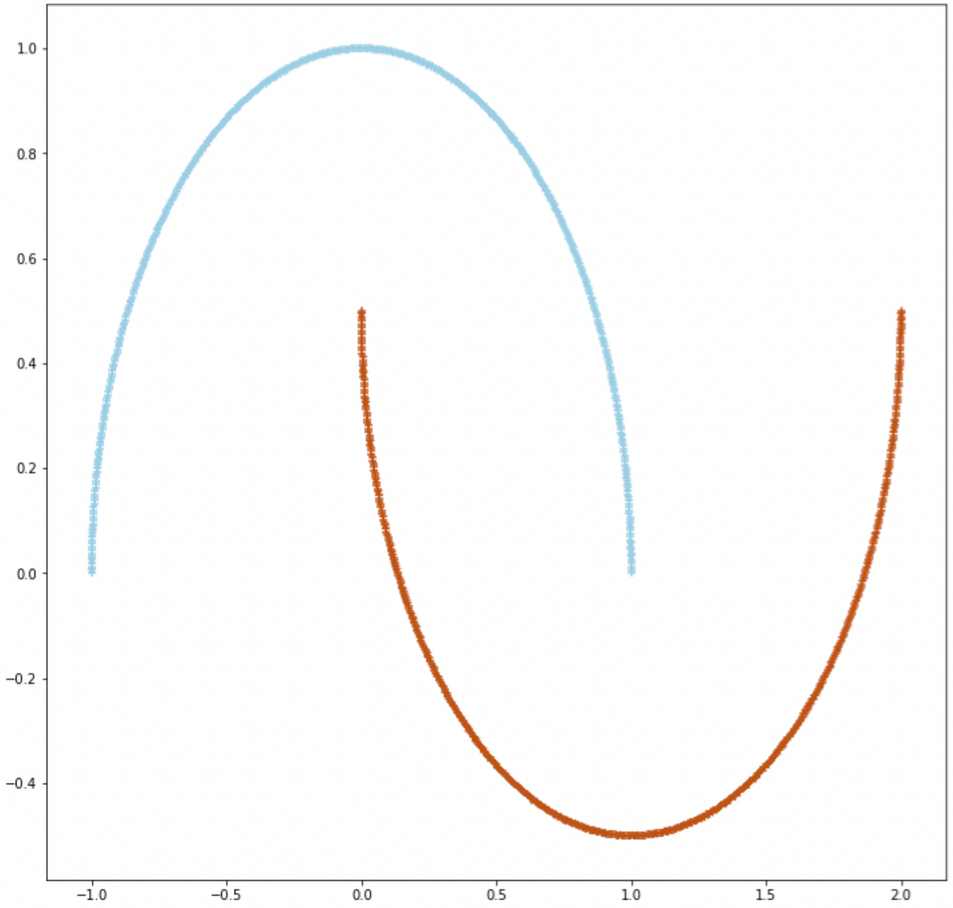
Data not linearly separated





Non-linear Logistic Regression

1000 data points and 2 classes (blue and red):



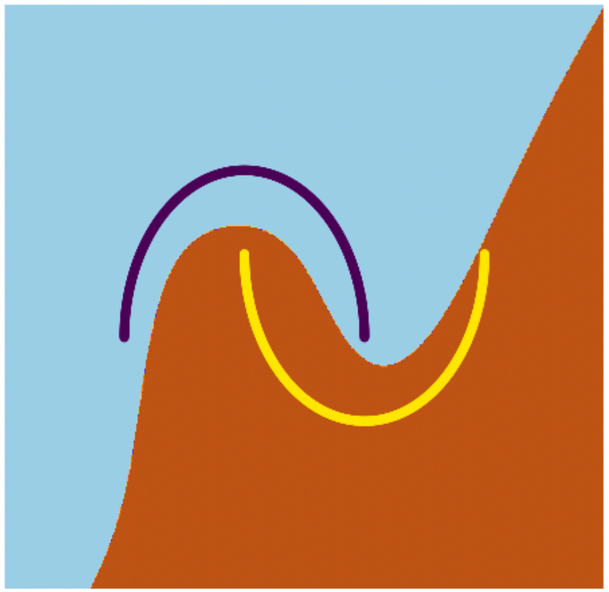
Task:

Predict the color at any point in the plane

Non-linear Logistic Regression

Simple example of logistic regression:

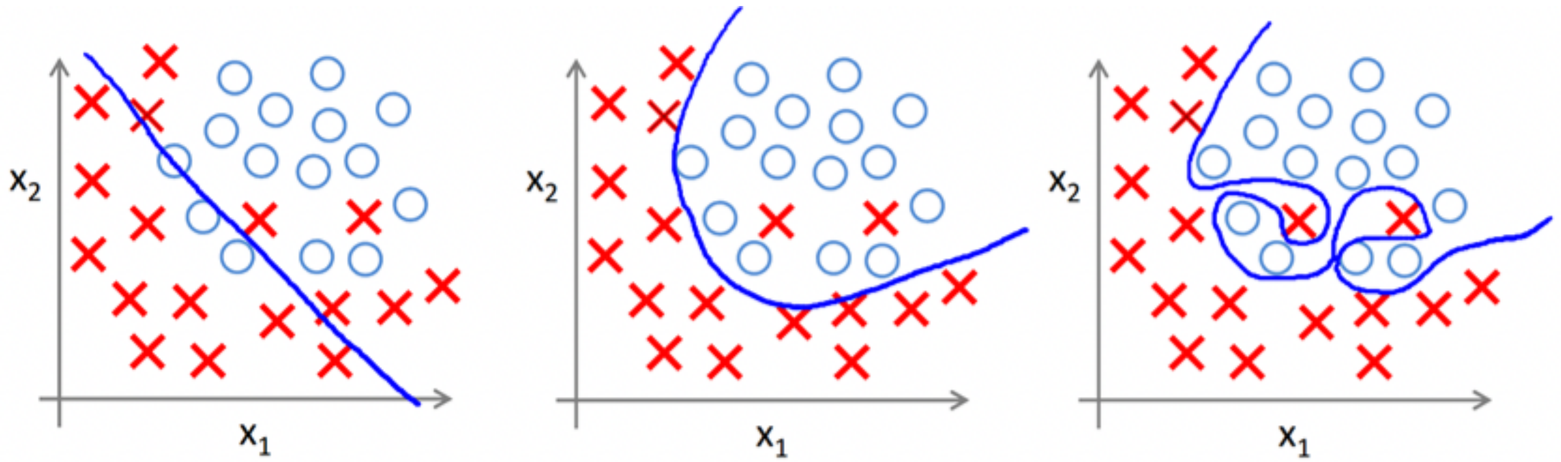
$$h_{\boldsymbol{\theta}}(x_1, x_2) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \cdots + \theta_{16} x_1^5 + \theta_{17} x_2^5 + \theta_{18} x_1^4 x_2 + \cdots + \theta_{21} x_1^2 x_2^3)$$



Decision Boundary is a polynomial of degree 5

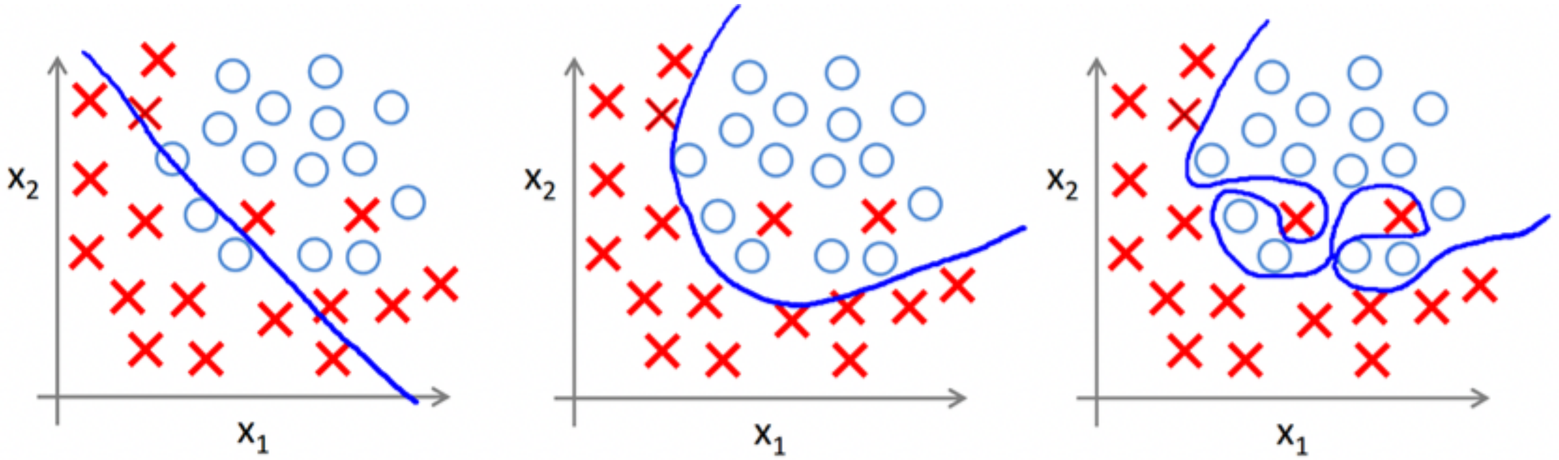
Non-linear Logistic Regression

How do we choose the degree of the polynomial?



Non-linear Logistic Regression

Application to MNIST



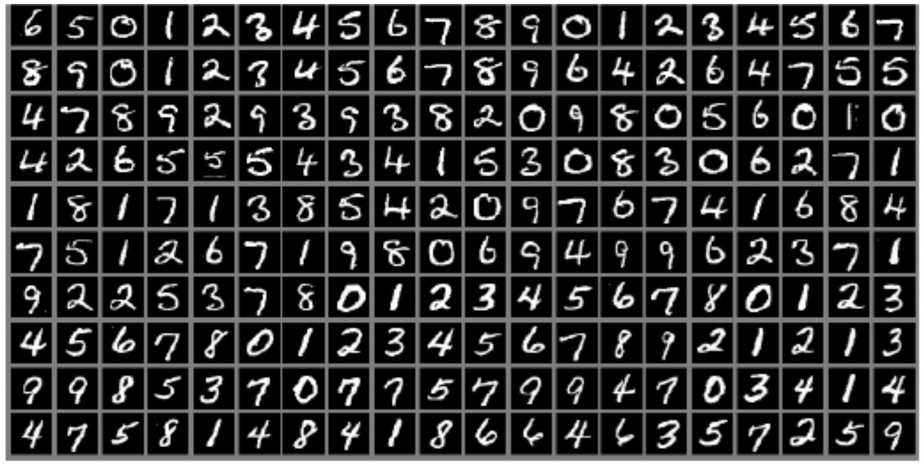
First degree polynomial (could do better)

Second degree polynomial (reasonable)

n^{th} degree polynomial (**overfitting!**)

Non-linear Logistic Regression

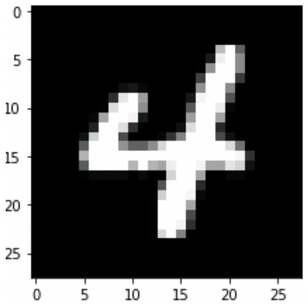
Application to MNIST (60000 training samples & 10000 test samples):



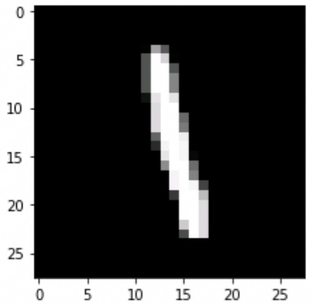
a few samples of MNIST

labels are the digits between 0 and 9

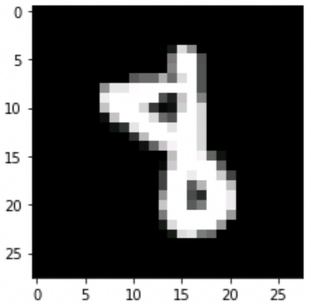
label=4



label=1



label=8



28x28 pixel images with values (0-255)

Non-linear Logistic Regression

Application to MNIST (60000 Training examples): **Softmax Regression** is used, a generalization of Logistic Regression for more than 2 classes (next session)

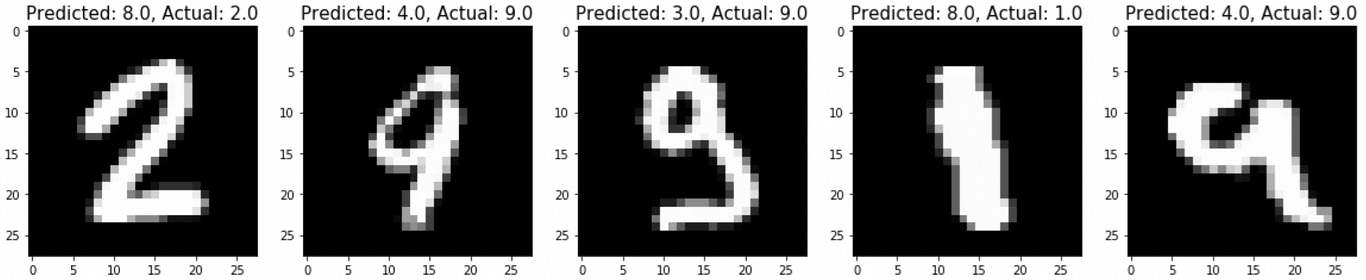
Results:

<u>60000 Training Images:</u>

Mean Accuracy: 0.94
Misclassified Images: 3893 (6.5%)

<u>10000 Test Images:</u>

Mean Accuracy: 0.92
Misclassified Images: 817 (8.2%)

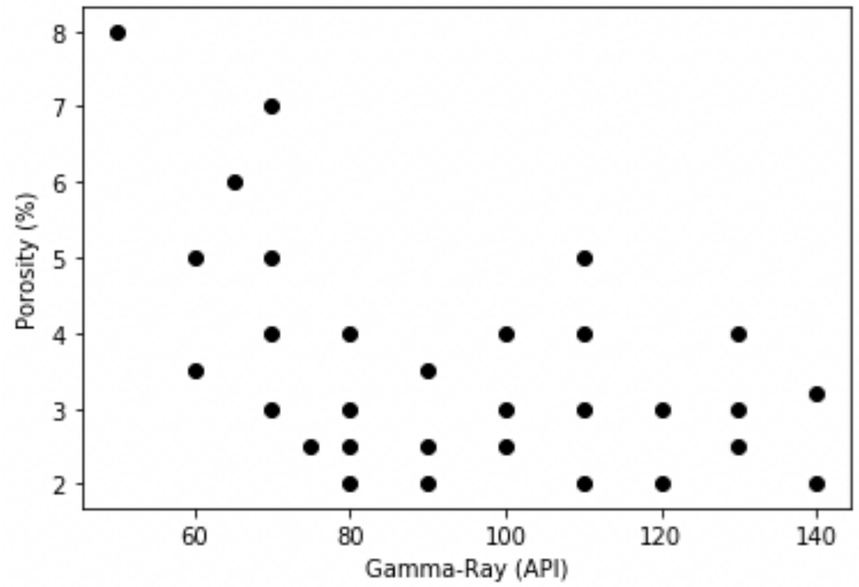


misclassified images

1. What is ML?
2. Unsupervised VS supervised learning
3. Linear regression
4. Logistic regression
5. **k-Means and PCA**

CLUSTERING

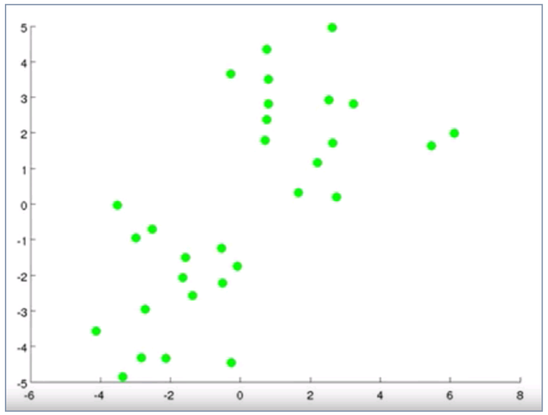
Clustering is an **Unsupervised Learning** approach



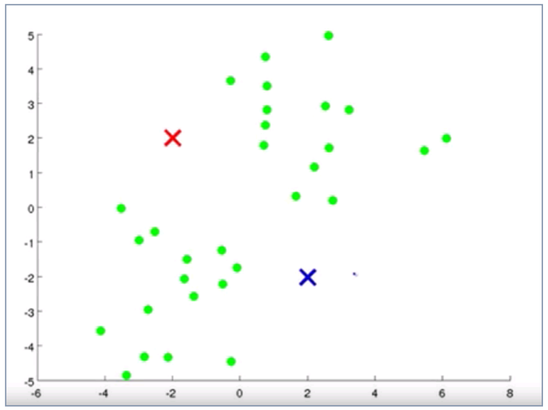
The Training set has **no labels** y^1 , it only has two **features** x^1

Clustering automatically groups the input training examples into a small number of clusters of ‘similar’ examples

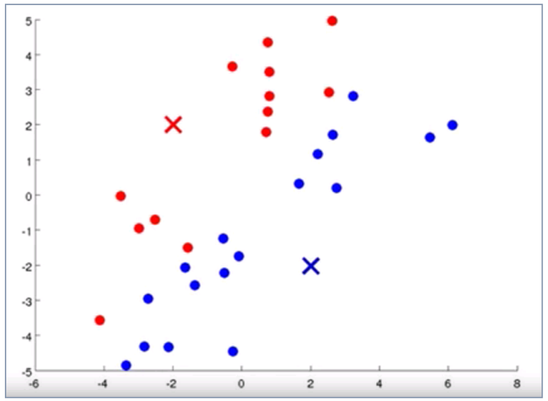
Visual description of the k-means algorithm:



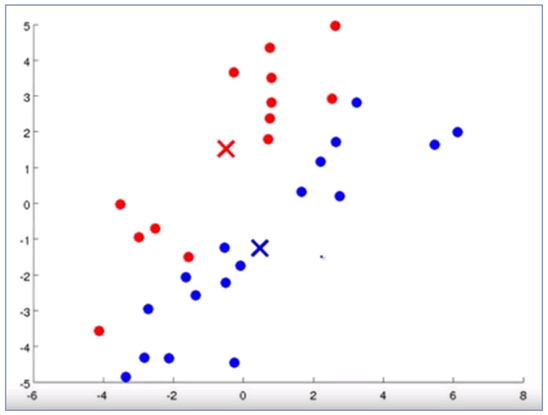
1- original unclassified dataset



2- random initialisation of centroids

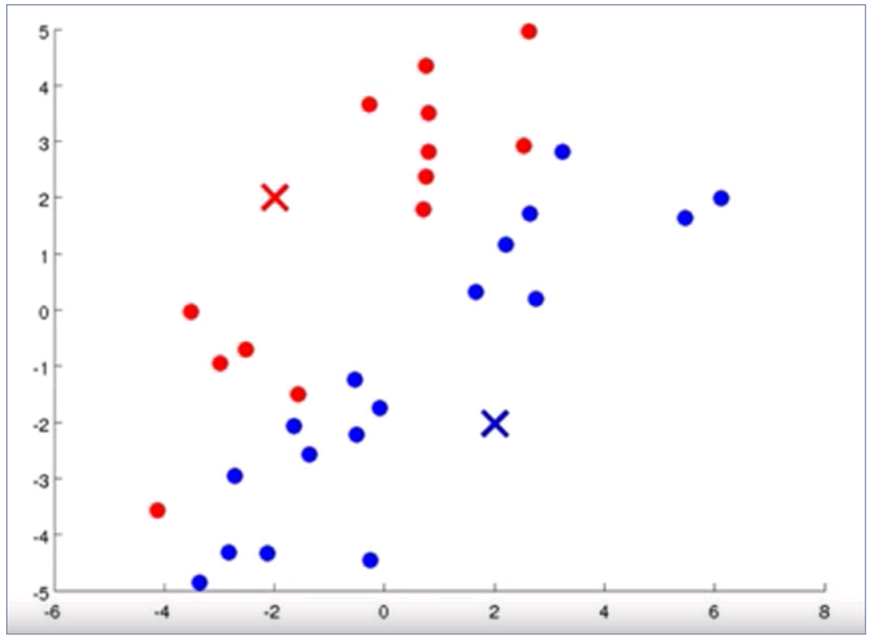


3- assign points closests to the centroid

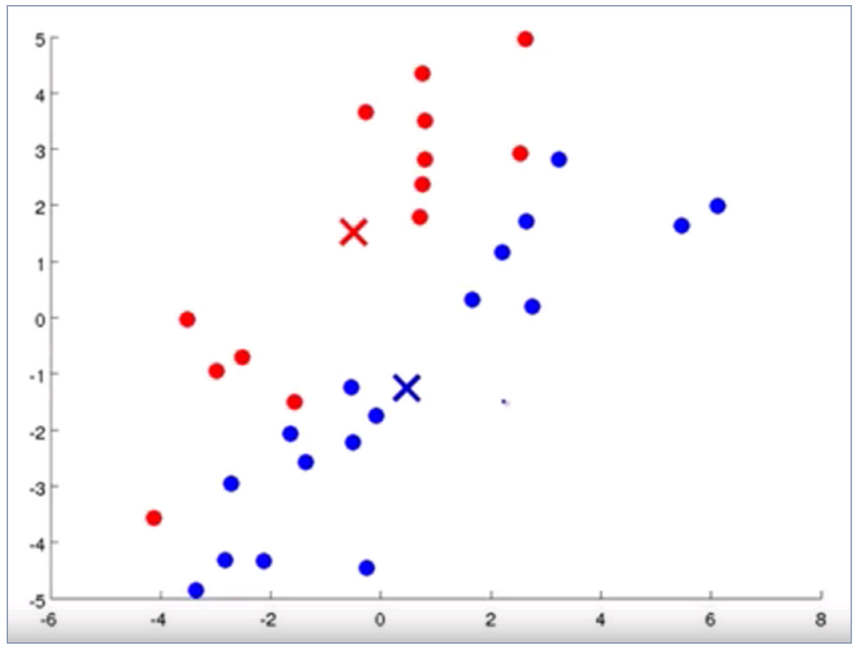


4- compute center of mass of clusters

Visual description of the k-means algorithm:

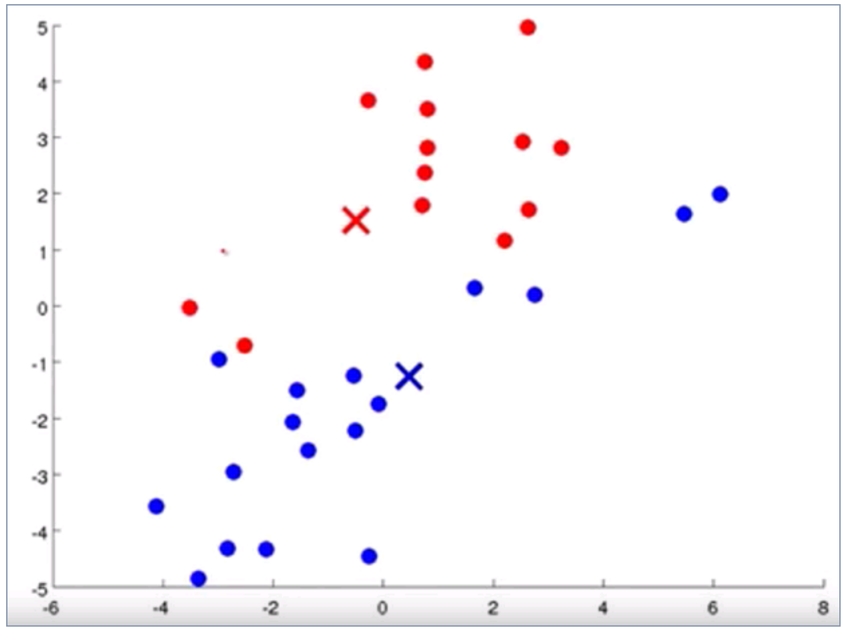


cluster assignment 1

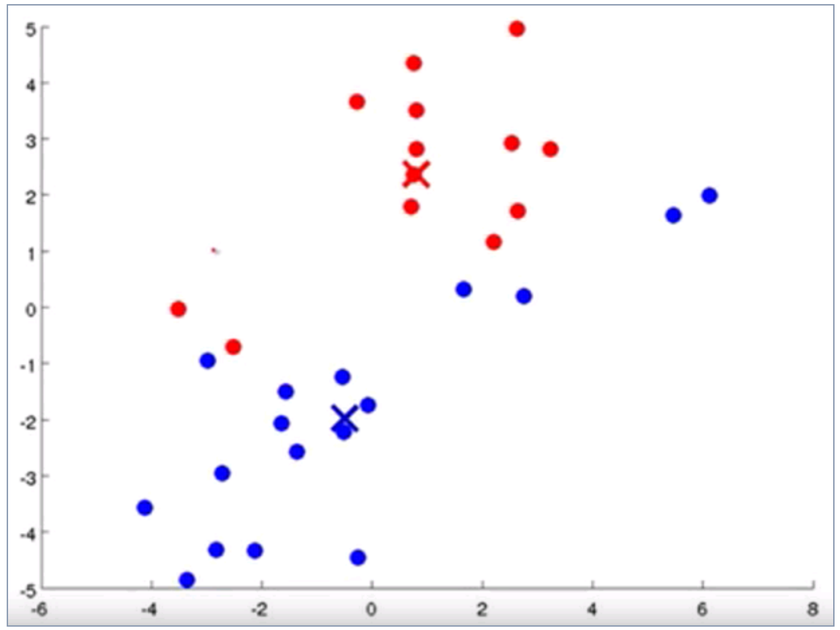


centroid assignment 1

Visual description of the k-means algorithm:

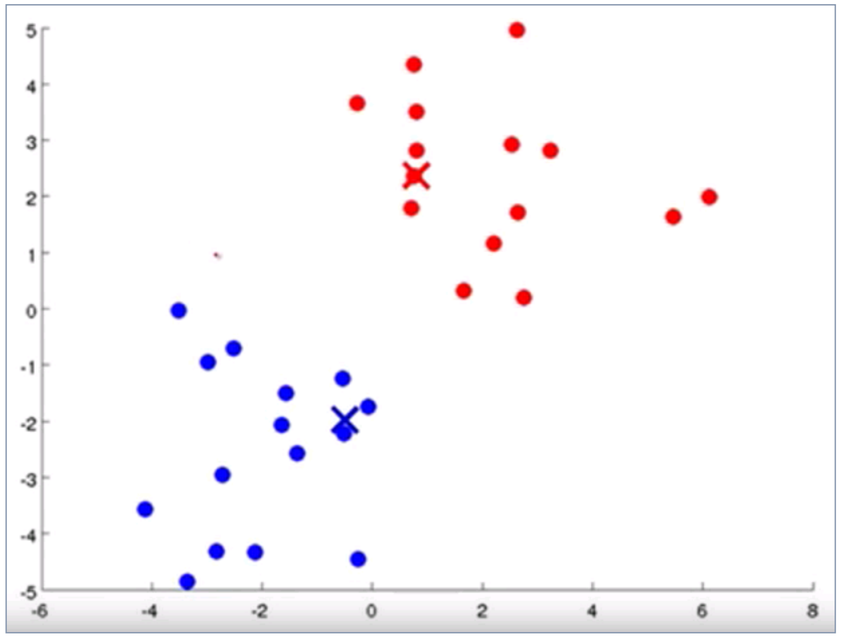


cluster assignment 2

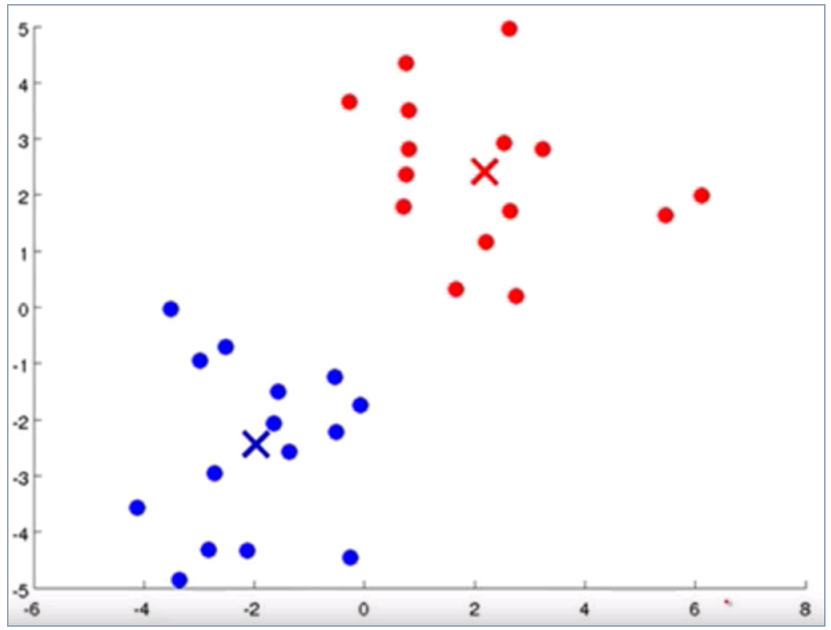


centroid assignment 2

Visual description of the k-means algorithm:



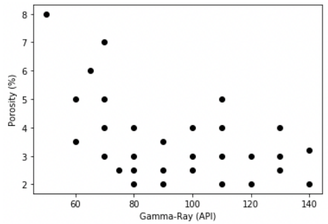
cluster assignment 3

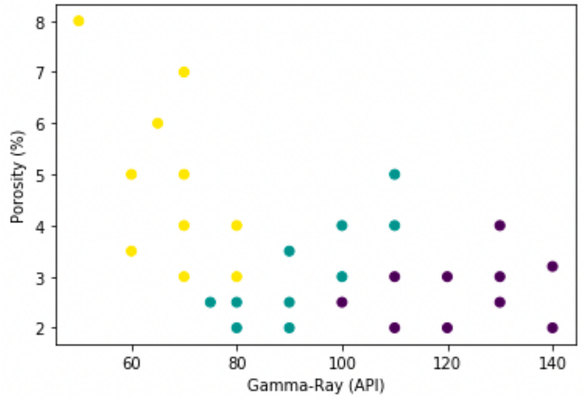


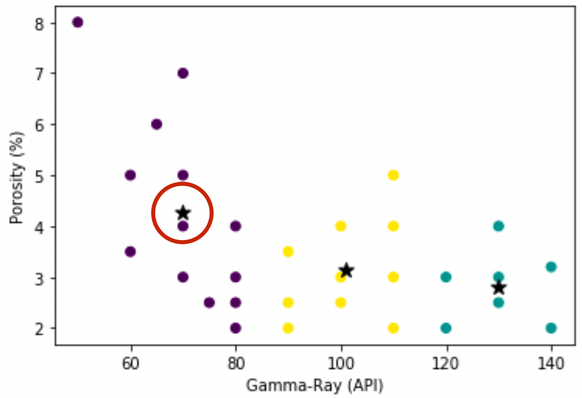
final centroids
(**in this example**, they don't change position anymore)

K-MEANS

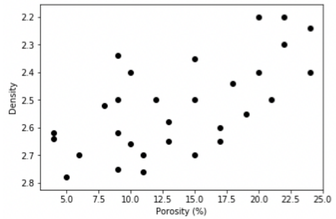
k-means performance depends heavily on the data structure we want to capture:

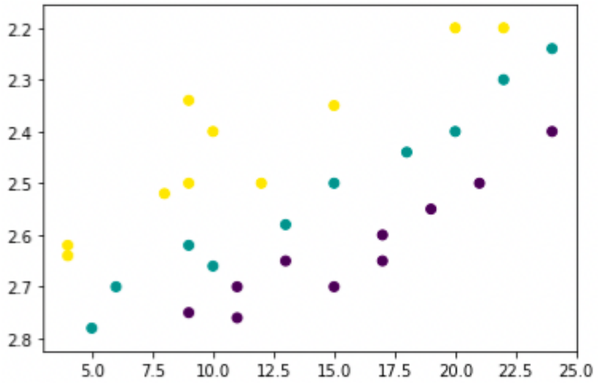


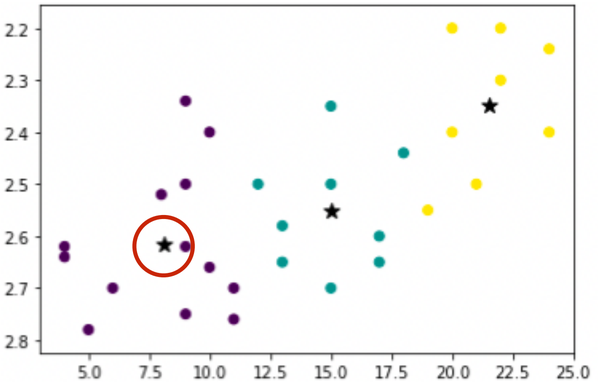




Pretty good



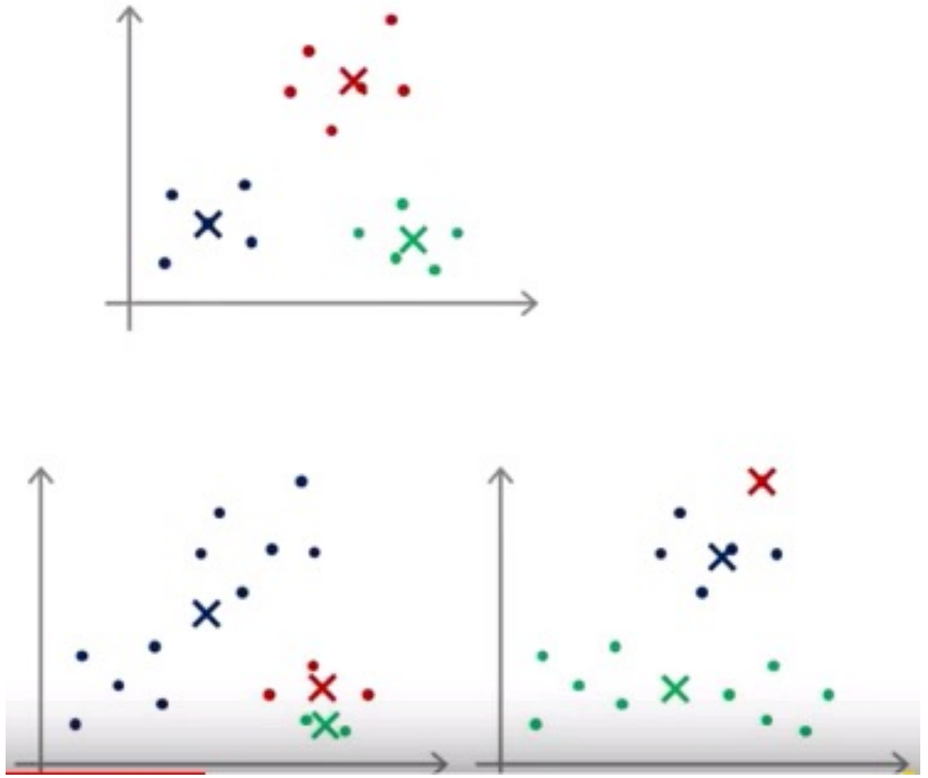




Poor match

Initial centroids are random, which means that the results are not unique

Three examples of possible clusters for the same dataset, one corresponds to global minimum, the two others to local minima.



Loss function for k-means:

The k-Means Loss Function is, for k classes and m data points (with $k \ll m$):

$$J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_k\right) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2$$

Classes associated with each data point

Class centroids

Data Points

Centroid of $x^{(i)}$'s class

Best approach: run say 100 k-Means with different random initializations of centroid, calculate Loss Function J for each of them, and pick run associated with lowest value of J.

Loss function for k-means:

The k-Means Loss Function is, for k classes and m data points (with $k \ll m$):

$$J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_k\right) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2$$

Classes associated with each data point

Class centroids

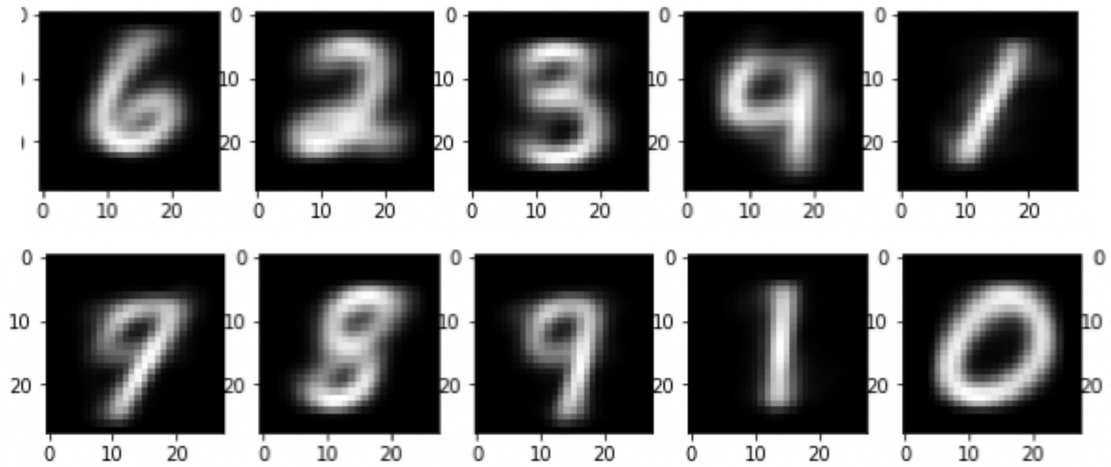
Data Points

Centroid of $x^{(i)}$'s class

Best approach: run say 100 k-Means with different random initializations of centroid, calculate Loss Function J for each of them, and pick run associated with lowest value of J.

Application to MNIST:

Each image is a centroid (class mean)



Problems:

- classes 4, 5, and 7 not represented
- three classes look like 9s, and two like 1s

How can we improve this poor result? With **PCA**

Principal Component Analysis (PCA)

Preliminary data normalisation for PCA:

Training Set vectors: $x^{(1)}, x^{(2)}, \ldots \ldots, x^{(m)}$. (no label)

Calculate mean of coordinates of the training vectors: $\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$

Calculate standard deviation s_j of coordinates of the training vectors: $s_j^2 = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)2} - \mu_j^2$

Replace each input feature by normalized value: $\boldsymbol{x_j^{(i)} := \frac{x_j^{(i)} - \mu_j}{s_j}}$

Principal Component Analysis (PCA)

Dimensionality reduction with PCA:

Training Set vectors: $x^{(1)}, x^{(2)}, \ldots \ldots, x^{(m)}$ (no label).

To project data from n-dimensional to k-dimensional space, calculate covariance matrix:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}\right)\left(x^{(i)}\right)^{T}$$

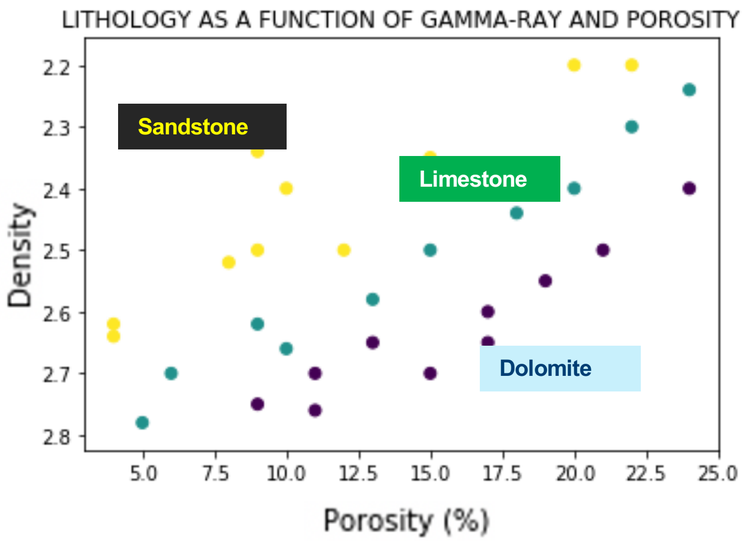
Then compute eigenvalues $(\lambda_i)_{i=1\ldots m}$ of matrix Σ

Keep the p largest eigenvalues $(\lambda_i)_{i=1\ldots p}$ and project on the space of dimension p defined by the associated p eigenvectors, also called principal components.

Principal Component Analysis (PCA)

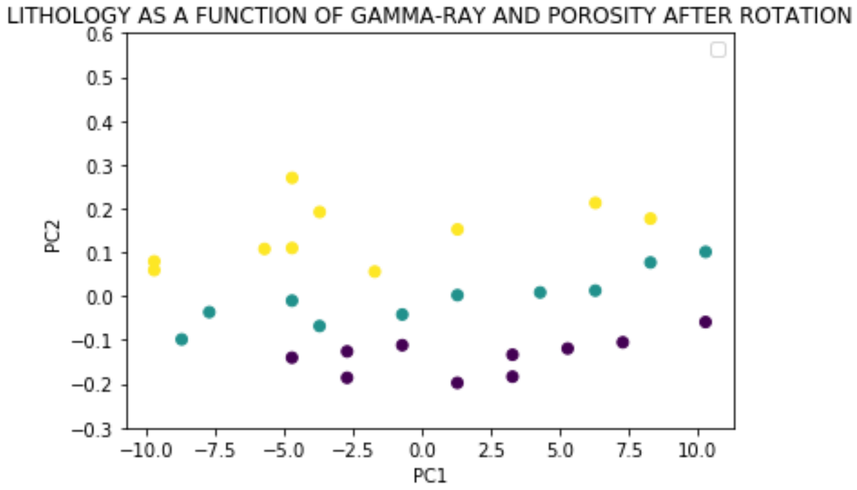
Dimensionality reduction with PCA on 2D example dataset:

Original Data Space



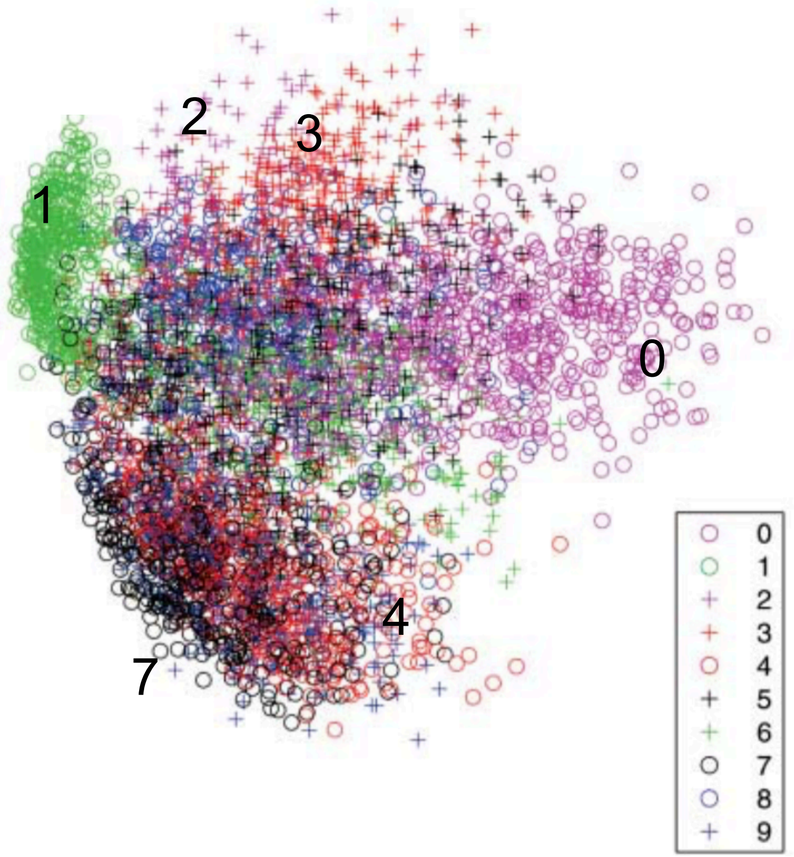
PCA

Principal Components Space



Principal Component Analysis (PCA)

Dimensionality reduction with PCA on MNIST dataset:

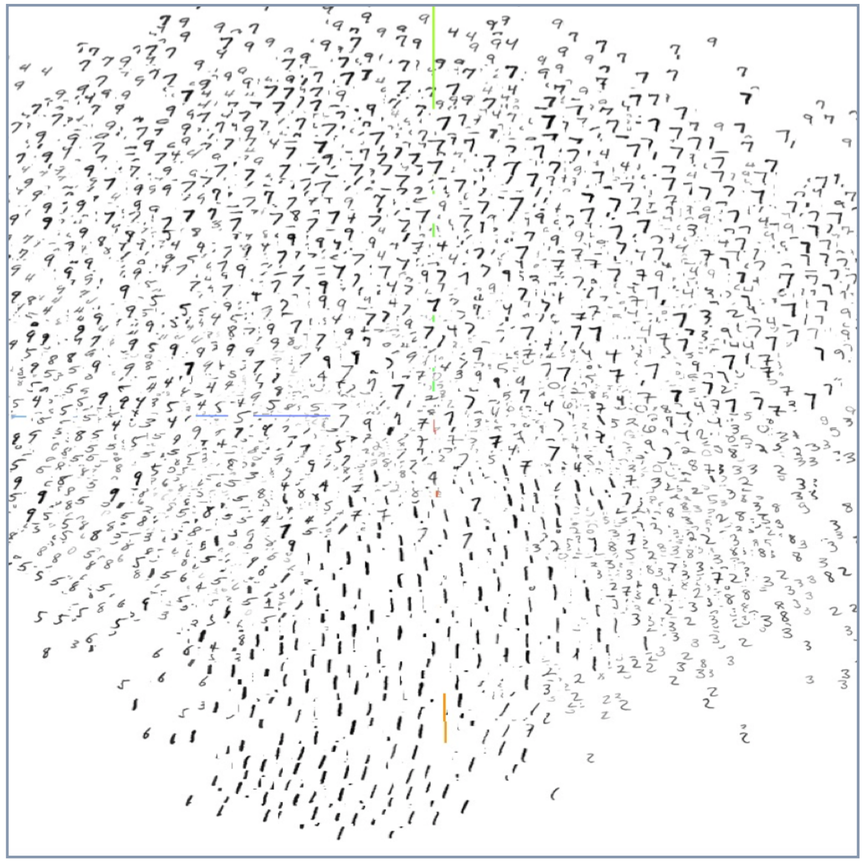


The two first principal components for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. The labels were not used for PCA, they are just posted on the PCA results.

From Hinton and Salakhutdinov, Science, Science, July 2006

Principal Component Analysis (PCA)

3D PCA on MNIST dataset:



https://projector.tensorflow.org/

Mathematics of PCA:

PCA transforms the data to a new coordinate system such that the greatest variance after projection of the data lies on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on….

If X is the $m \times n$ data matrix (m data points, n features), each row vector $(x_{ij})_{j=1,\dots,n}$ of dimension n is mapped into a new vector $(t_{il})_{l=1,\dots,p}$ of dimension p which is a linear combination of the n coordinates of $(x_{ij})_{j=1,\dots,n}$:

$$t_{il} = \sum_{j=1}^{n} x_{ij} w_{jl}$$

We want the vector $(t_{i1})_{i=1,\dots,n}$ to maximize its norm, or the first weight vector to satisfy:

$$w_1 = argmax\left(\sum_{i=1}^{m} t_{i1}^2\right) = argmax(w_1^T X^T X w_1)$$

Rayleigh Theorem says that the maximum value of the above norm is the largest eigenvalue of $X^T X$, which occurs when w is the corresponding eigenvector.

https://en.wikipedia.org/wiki/Principal_component_analysis

SUMMARY

- Differences between supervised and unsupervised learning
- Regression as a first ‘machine learning’ method
- Logistic regression as the elementary ML supervised classificaiton method
- k-means and PCA as the elementary ML unsupervised method
- Networks are extensions and generalisations of these concepts
- Mathematical formulations are not yet unified because ML is an emerging and exponentially growing field.