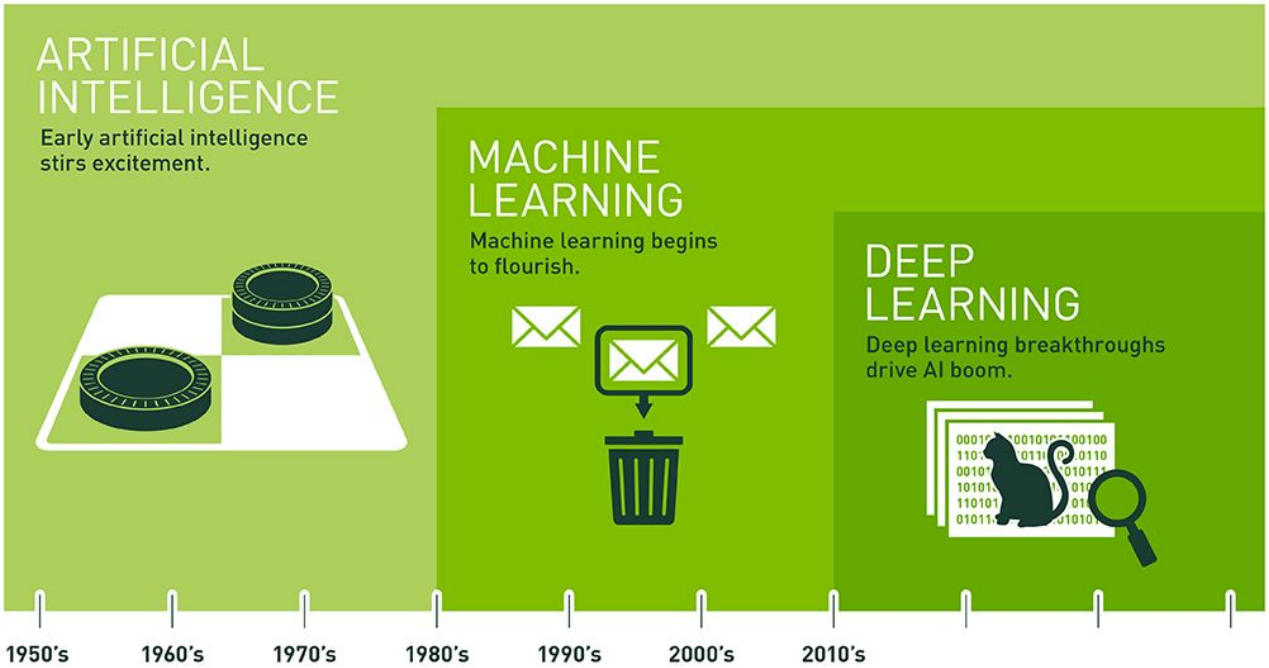
MACHINE LEARNING INTRODUCTION

Although AI has been around for decades, new advances have ignited a boom in practical applications through advancements in statistical techniques known as Deep Learning.

Deep learning powers self-driving cars, image/video recognition systems, natural language understanding, speech recognition/synthesis, life-saving advances in medicine and many more applications.

As an example, you are using DL-powered applications in Google, Facebook and Apple products that you use everyday.

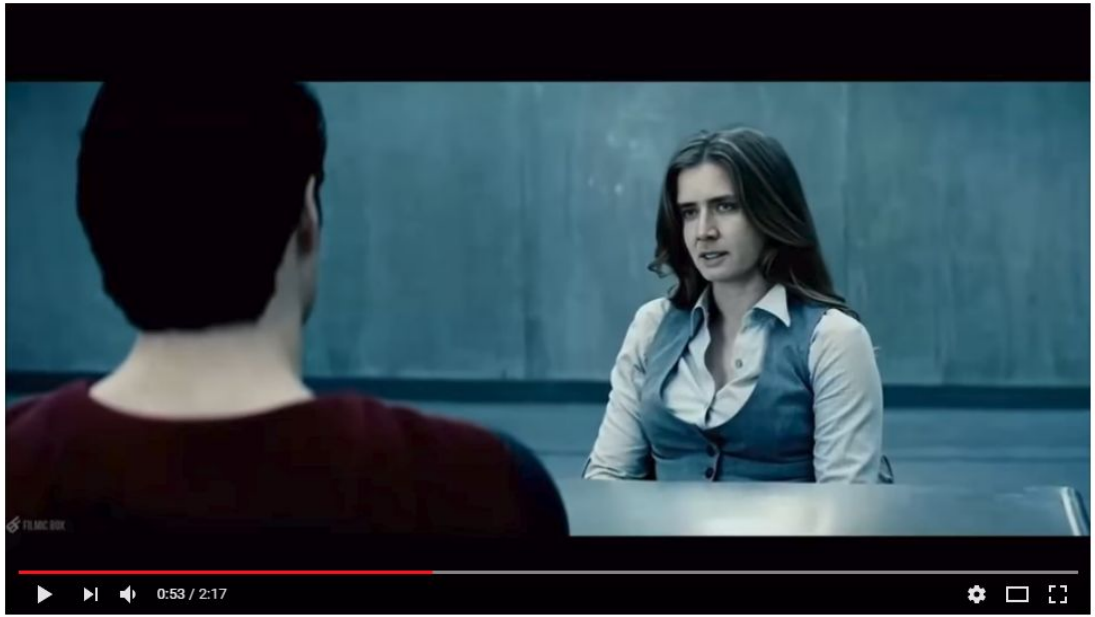


Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created larger disruptions.

DeepFakes

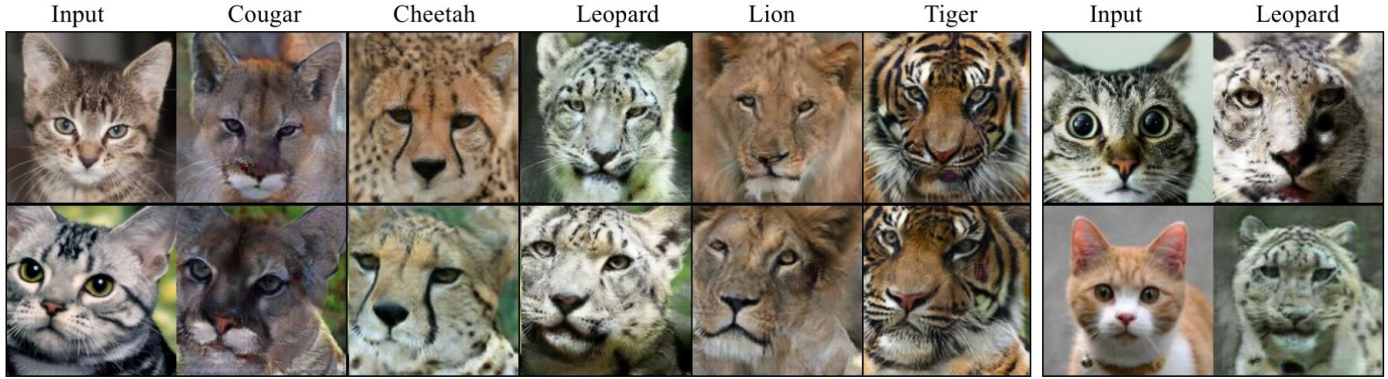
GAN

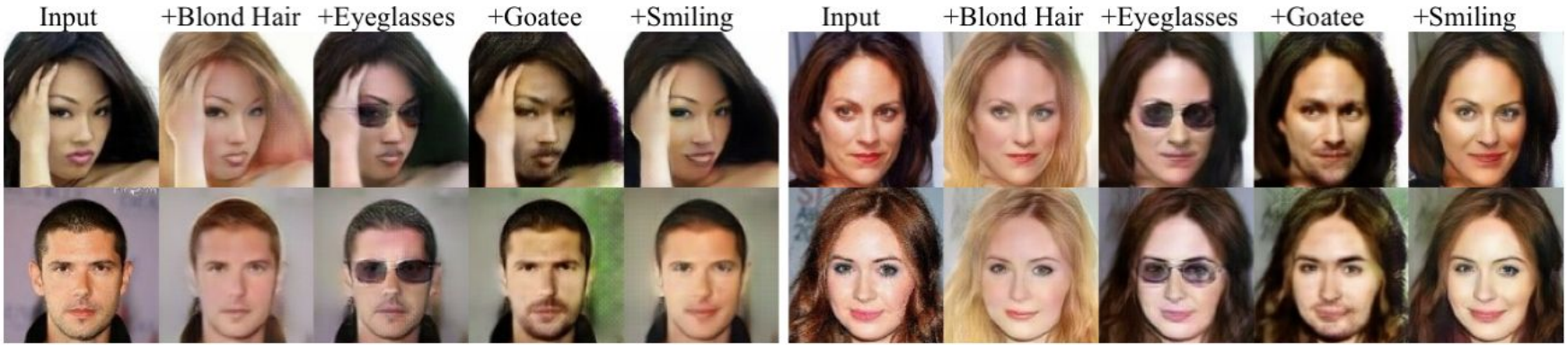
Nicholas Cage DeepFakes movie compilation



Nick Cage DeepFakes Movie Compilation

Demo: https://youtu.be/BU9YAHigNx8?t=42s

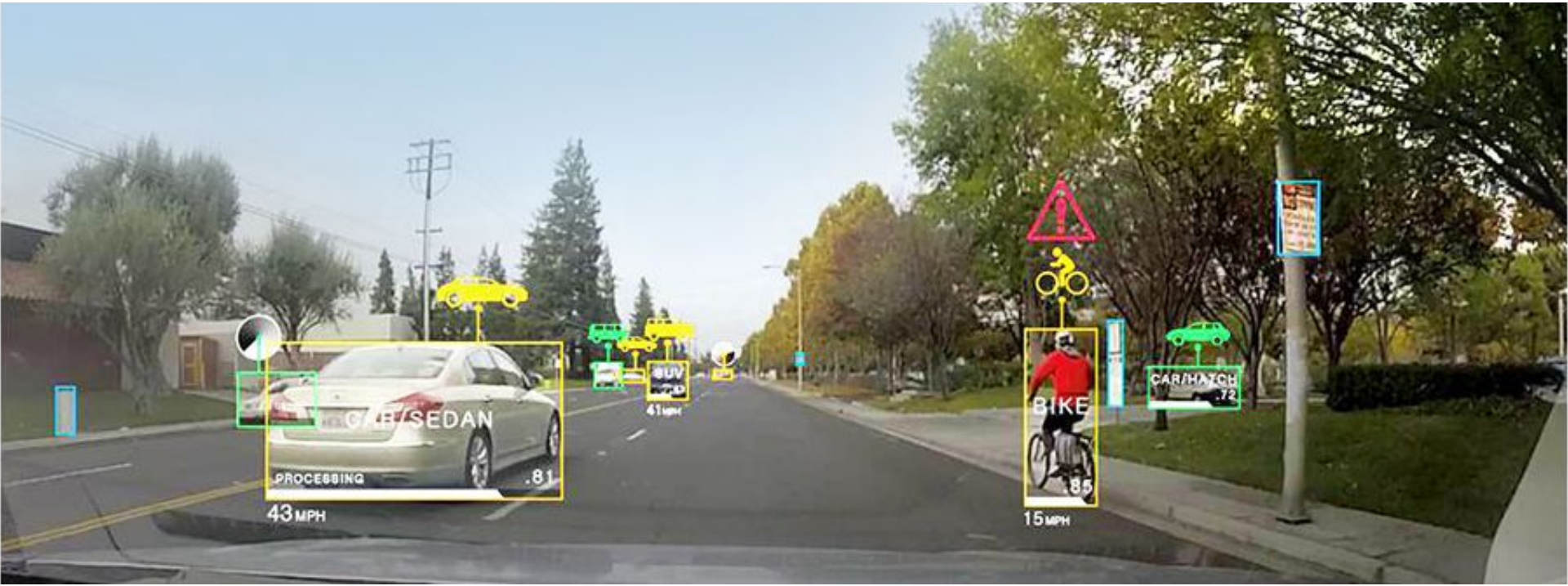

Input
Cougar
Cheetah
Leopard
Lion
Tiger
Input
Leopard


Input
+Blond Hair
+Eyeglasses
+Goatee
+Smiling
Input
+Blond Hair
+Eyeglasses
+Goatee
+Smiling

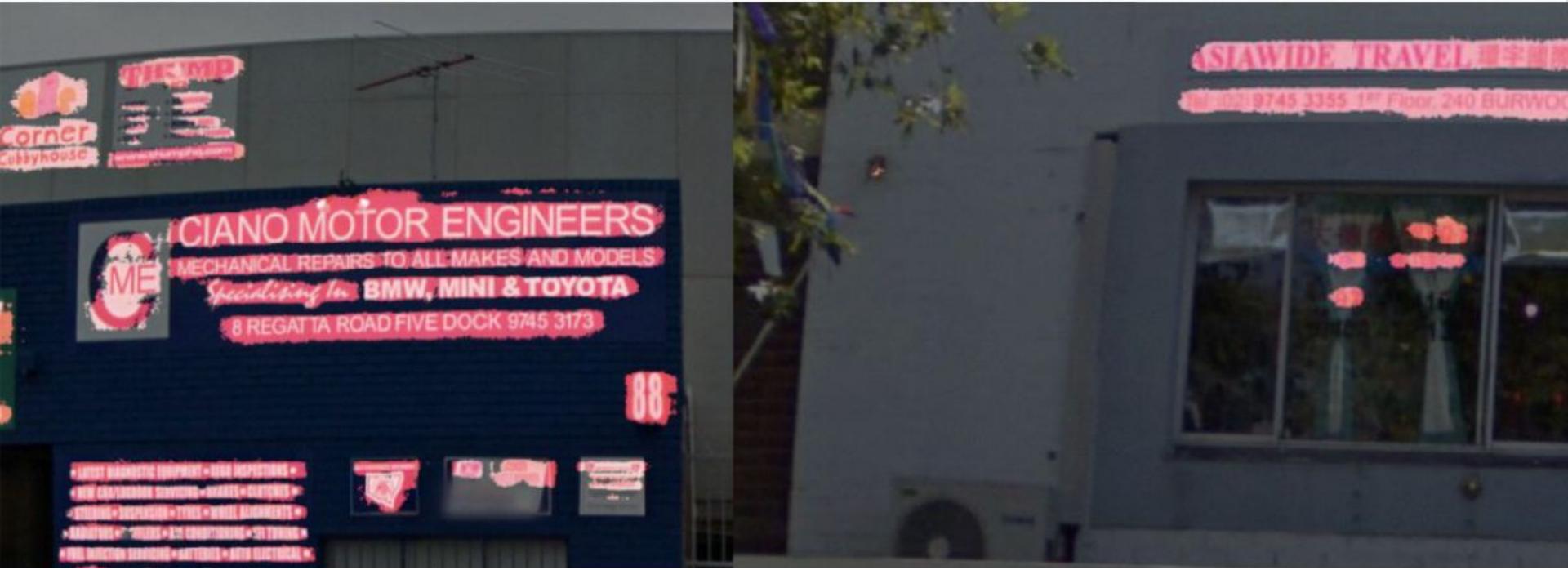
More info: https://github.com/mingyuliutw/UNIT

More info: https://github.com/mingyuliutw/UNIT

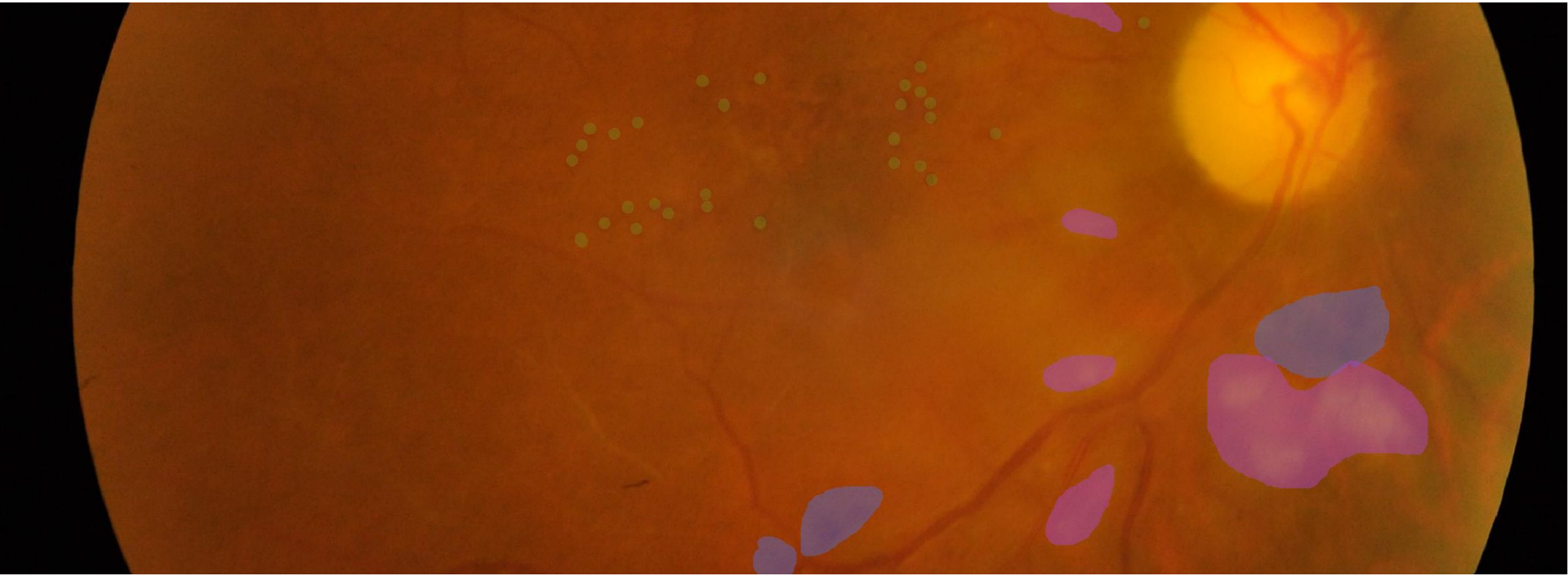
Self-driving cars



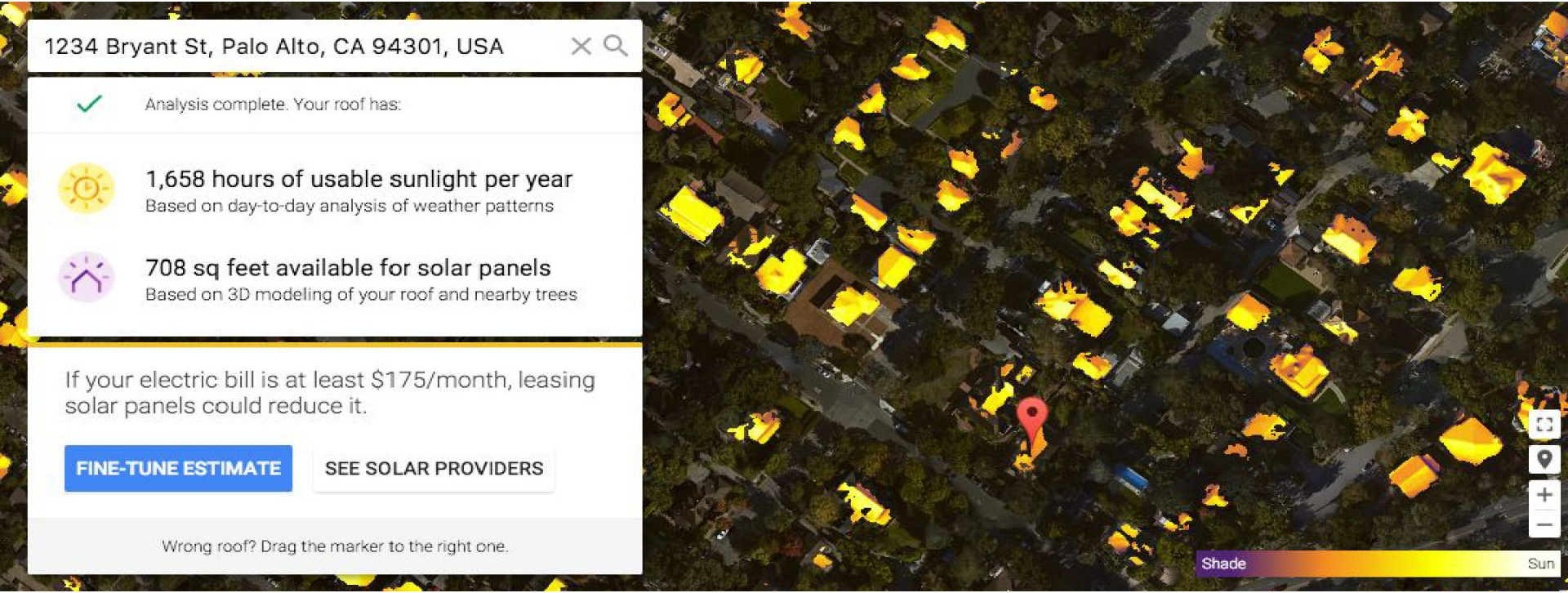
Text recognition



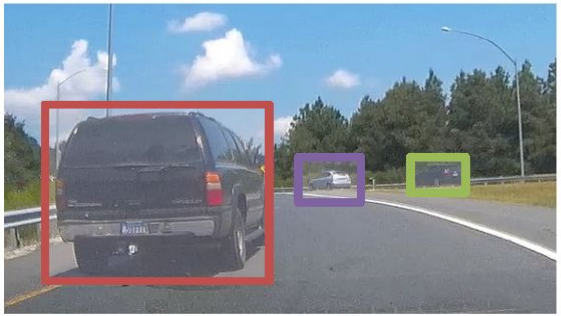
Medical applications

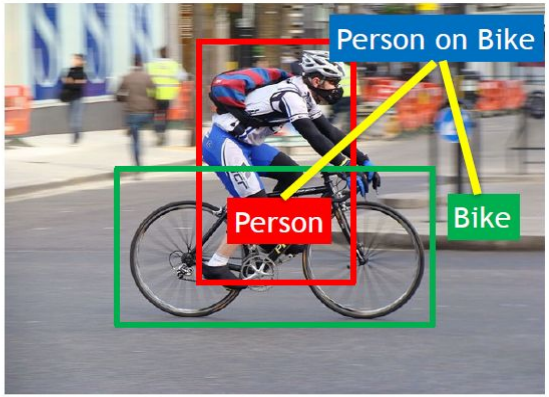


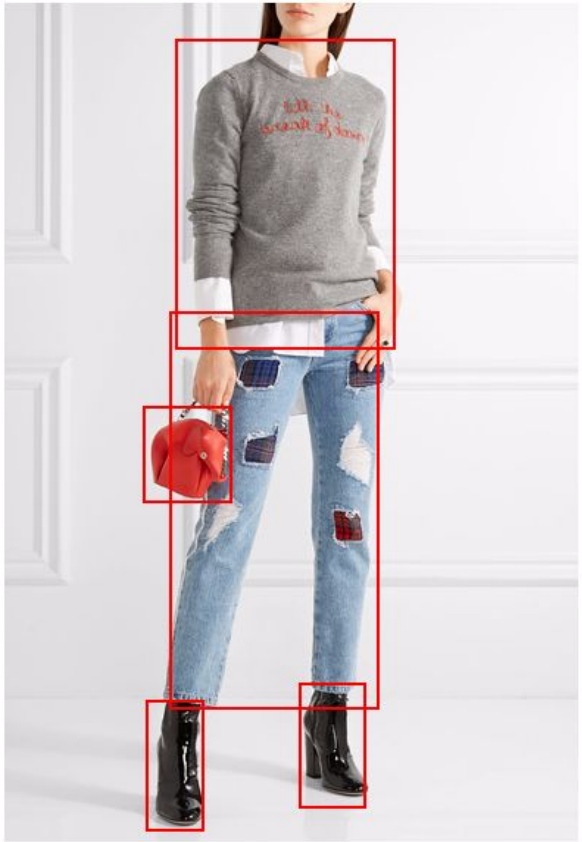
Google sunroof



Object detection - Action classification - Image captioning







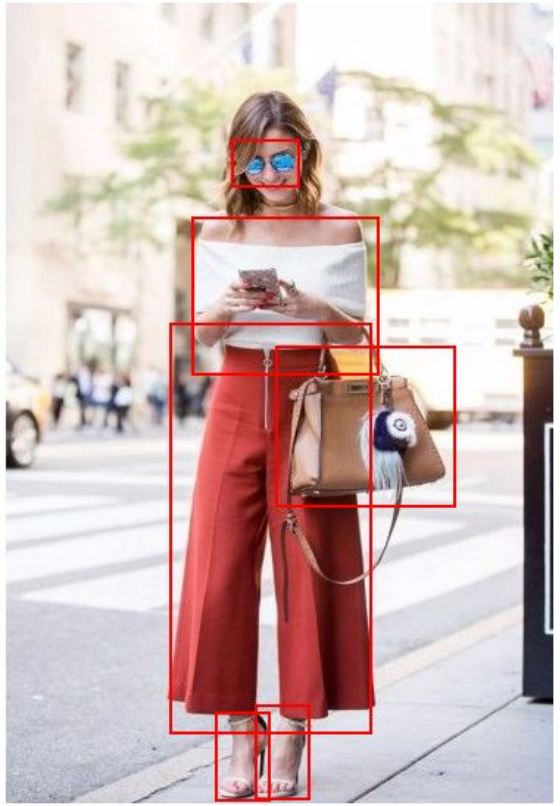
jeans

boots

boots

sweater

bag



bag

heels

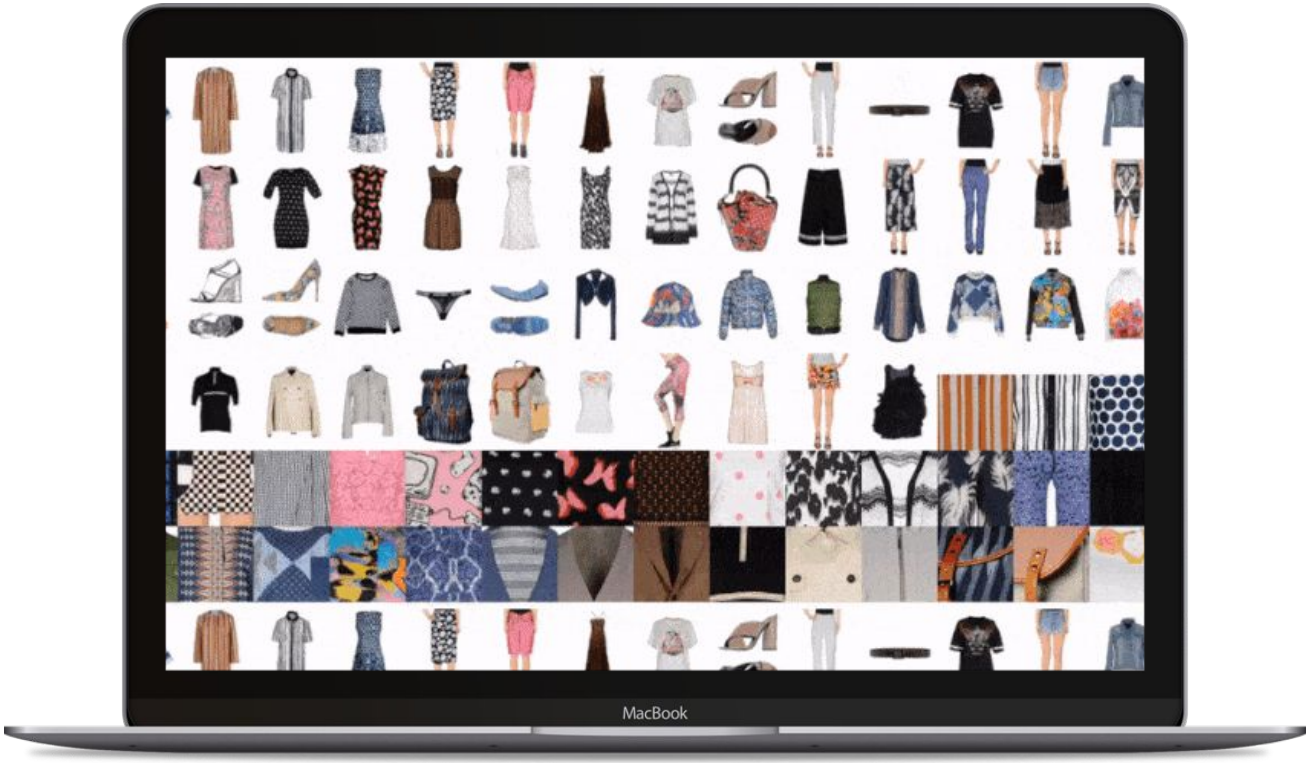
sunglasses

pants

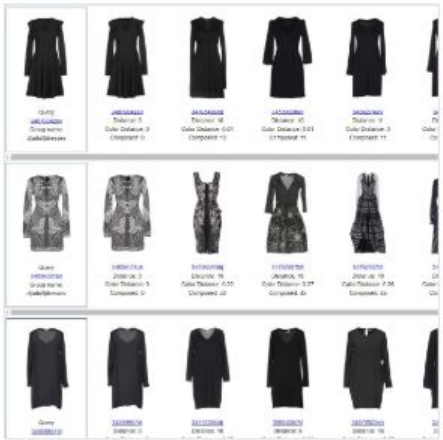
heels

top

Image similarity API



Yoda-similarity

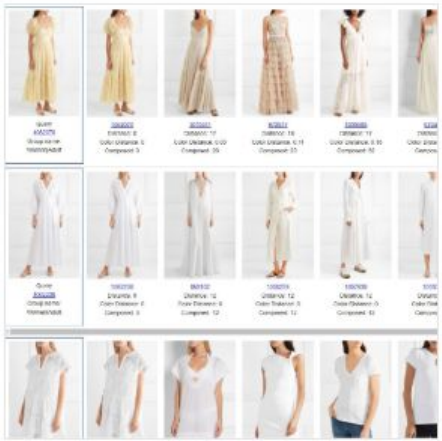


Similarity Browser Yoox Catalog

Visualizer for similarity results for Yoox catal…

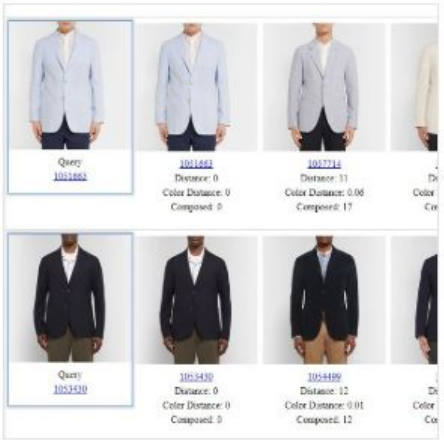
Similarity Browser NAP Catalog

Visualizer for similarity results for NAP catalog



Similarity Browser NAP Catalog (w/o macro…

Visualizer for similarity results for NAP catal…



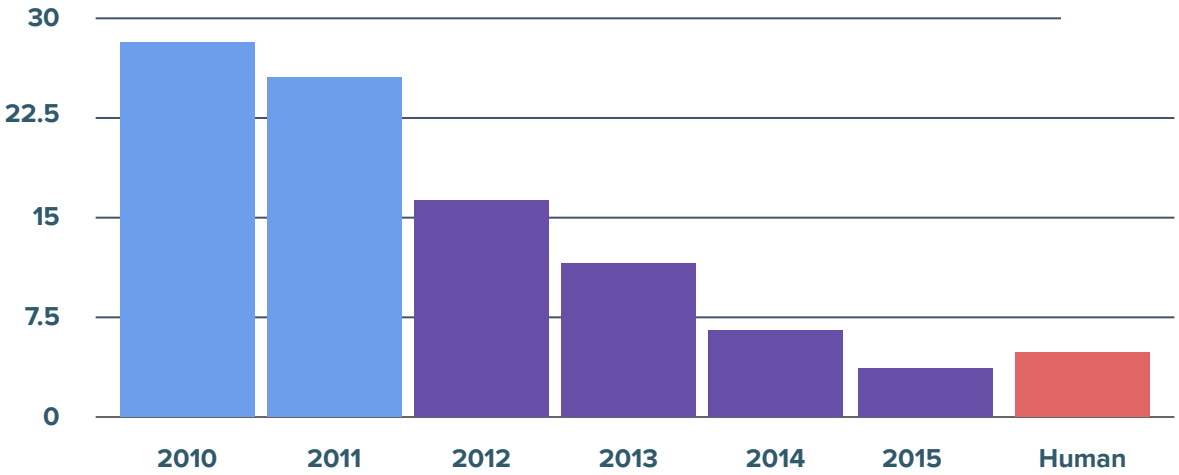
Similarity Browser MRP Catalog

Visualizer for similarity results for MRP catal…

Imagenet Challenge

Classification error %

1,000 object classes
1.5 million images

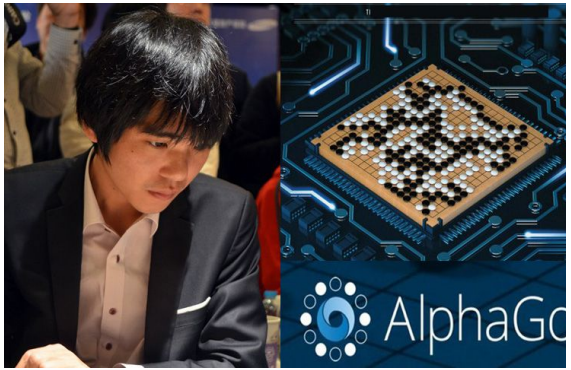


Better than human performance in image classification tasks

CHESS (1997)

World champion Kasparov vs Deep Blue (IBM)
Game complexity: 10^{50}

GO (2016)



World champion Lee Sedol vs AlphaGo (Google)
Game complexity: 10^{170}

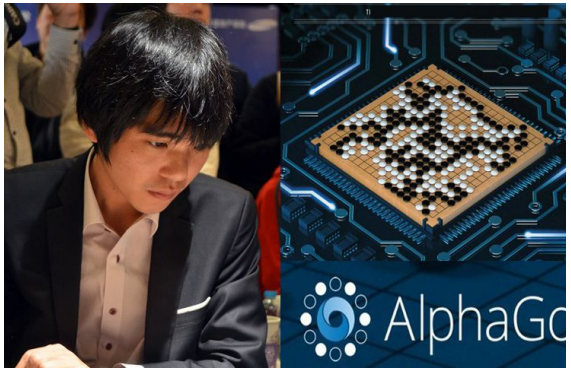
Where Deep Blue mainly relied on brute computational force to evaluate millions of positions, AlphaGo also relied on neural networks and reinforcement learning, which more closely resemble human decision-making

CHESS (2017)

AlphaZero vs StockFish

4-hour training time starting from scratch

GO (2017)

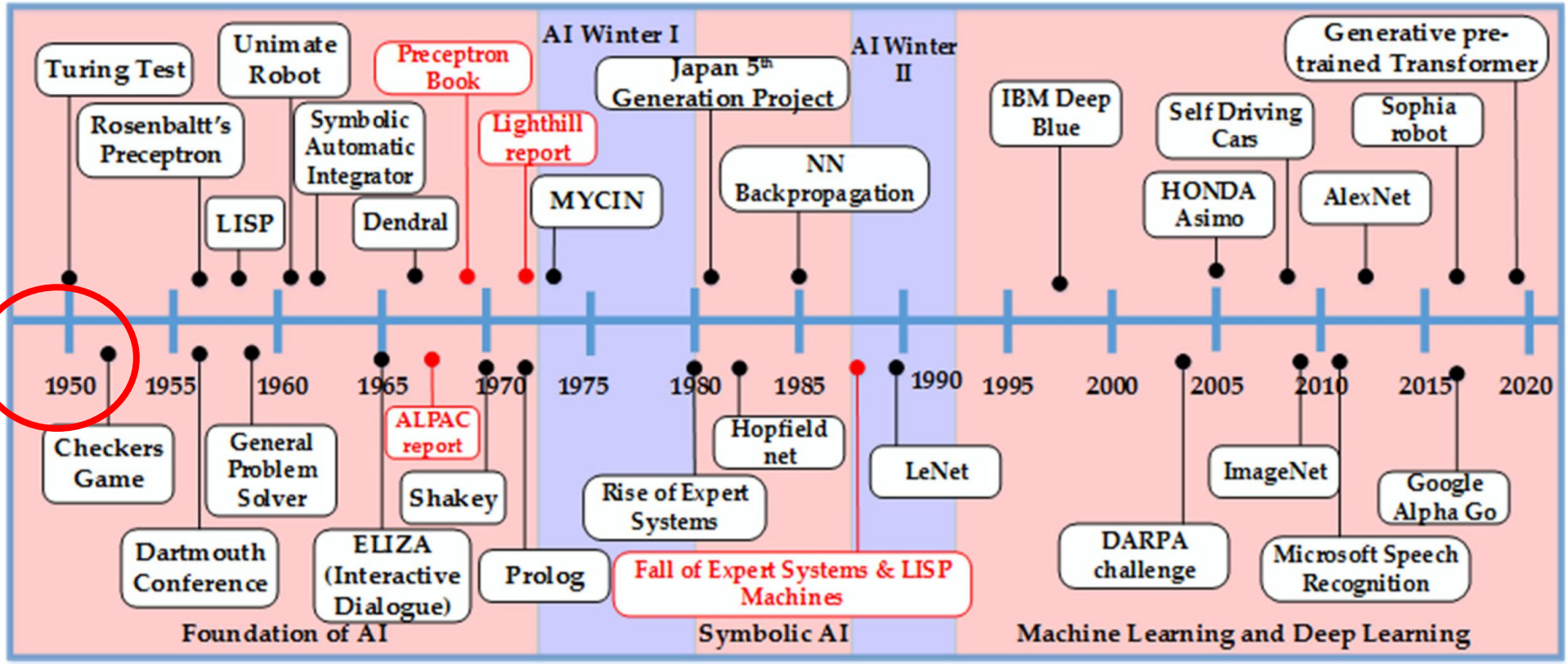


AlphaGo vs AlphaGo Zero (Google)

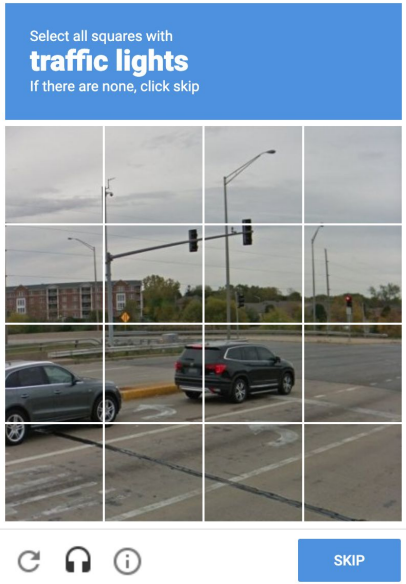
3-day training time starting from scratch

Based on reinforcement learning

AlphaZero searches fewer positions that its predecessor (80k per second vs 70M per second)
GM Magnus Carlsen: “I always wondered how it would be if a superior species landed on earth and showed us how they play chess. I feel now I know”


Turing Test
Rosenbaltt's Preceptron
Unimate Robot
LISP
Symbolic Automatic Integrator
Dendral
Preceptron Book
Lighthill report
MYCIN
AI Winter I
Japan 5th Generation Project
NN Backpropagation
AI Winter II
IBM Deep Blue
Self Driving Cars
HONDA Asimo
AlexNet
Sophia robot
Generative pre-trained Transformer
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
Checkers Game
General Problem Solver
ALPAC report
Shakey
Hopfield net
LeNet
ImageNet
Google Alpha Go
Dartmouth Conference
ELIZA (Interactive Dialogue)
Prolog
Rise of Expert Systems
Fall of Expert Systems & LISP Machines
DARPA challenge
Microsoft Speech Recognition
Foundation of AI
Symbolic AI
Machine Learning and Deep Learning

2020


Select all squares with
traffic lights
If there are none, click skip
SKIP

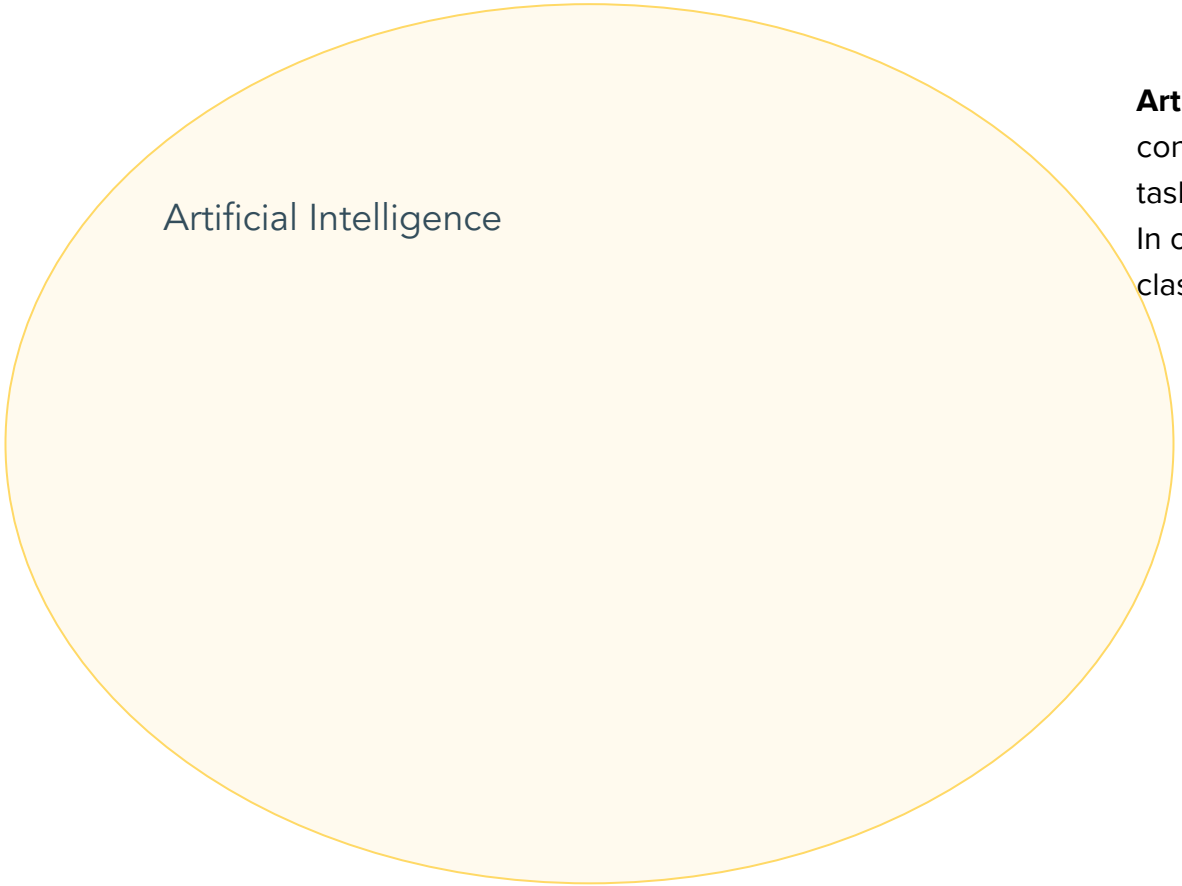

AutoDraw
Fast drawing for everyone.


Google Maps

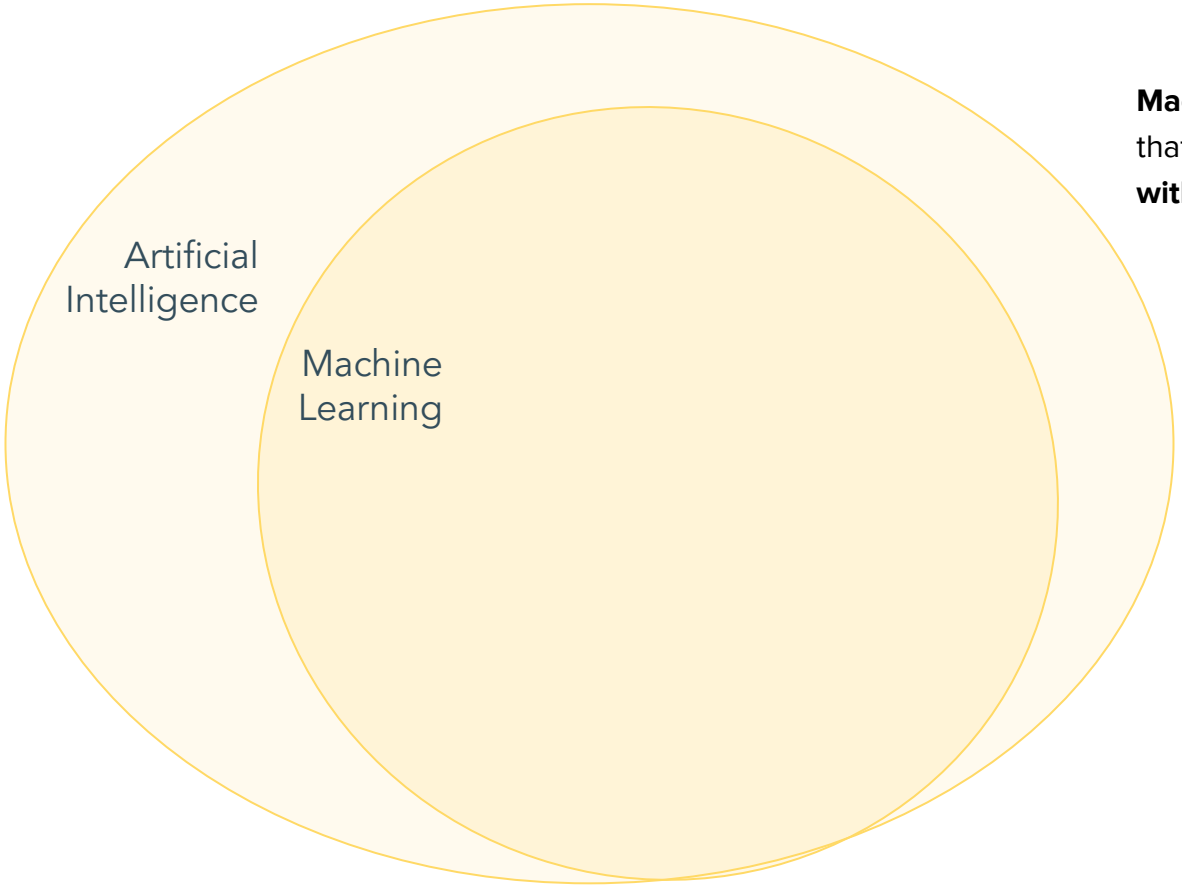

Google
Analytics


Google
Translate

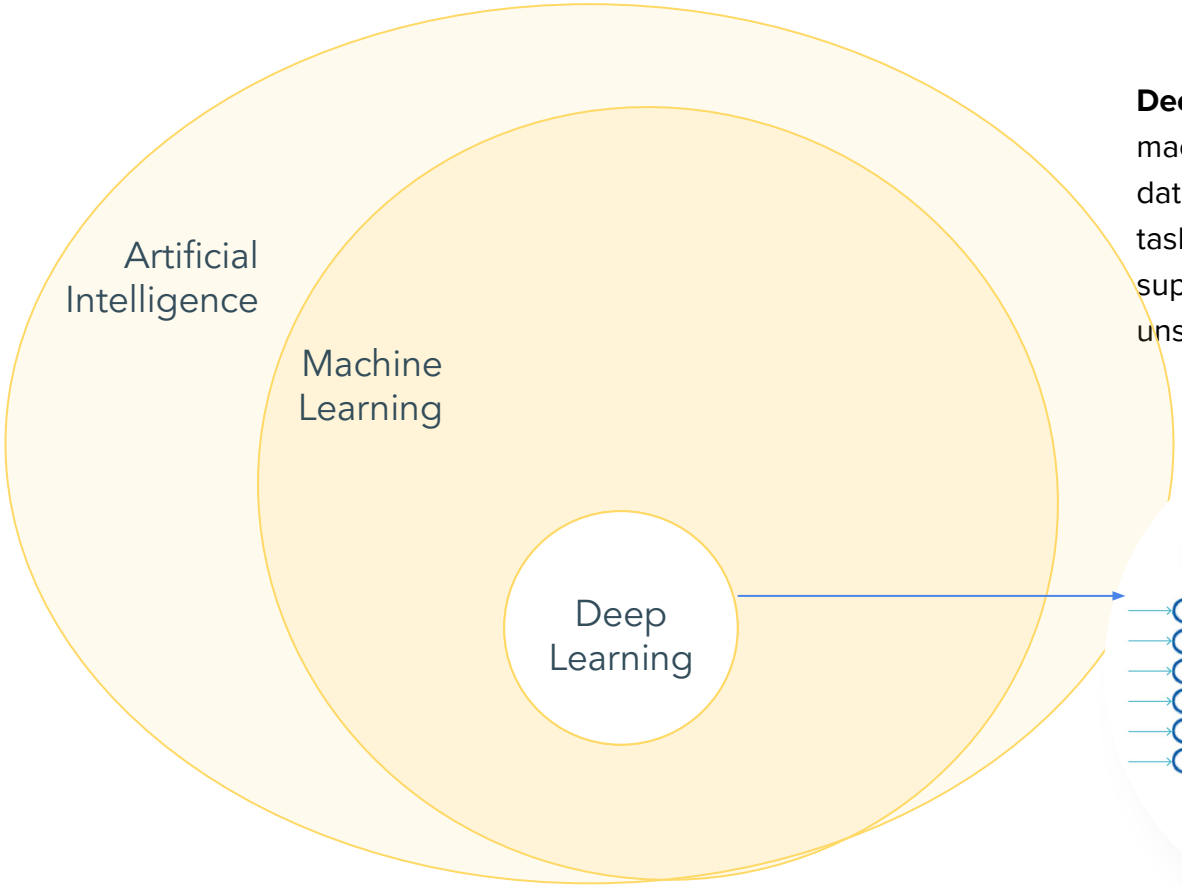
What is Artificial Intelligence?



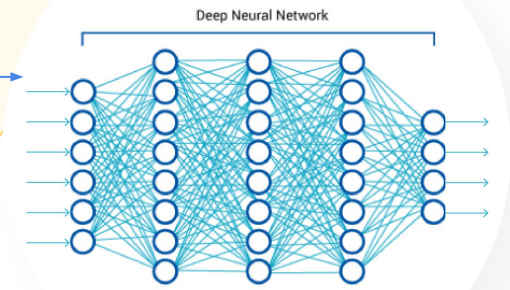
Artificial Intelligence (AI) is the broader concept of machines being able to carry out tasks in a way that we would consider “smart”. In other words, we are not able to write a classic computer program to perform the task.

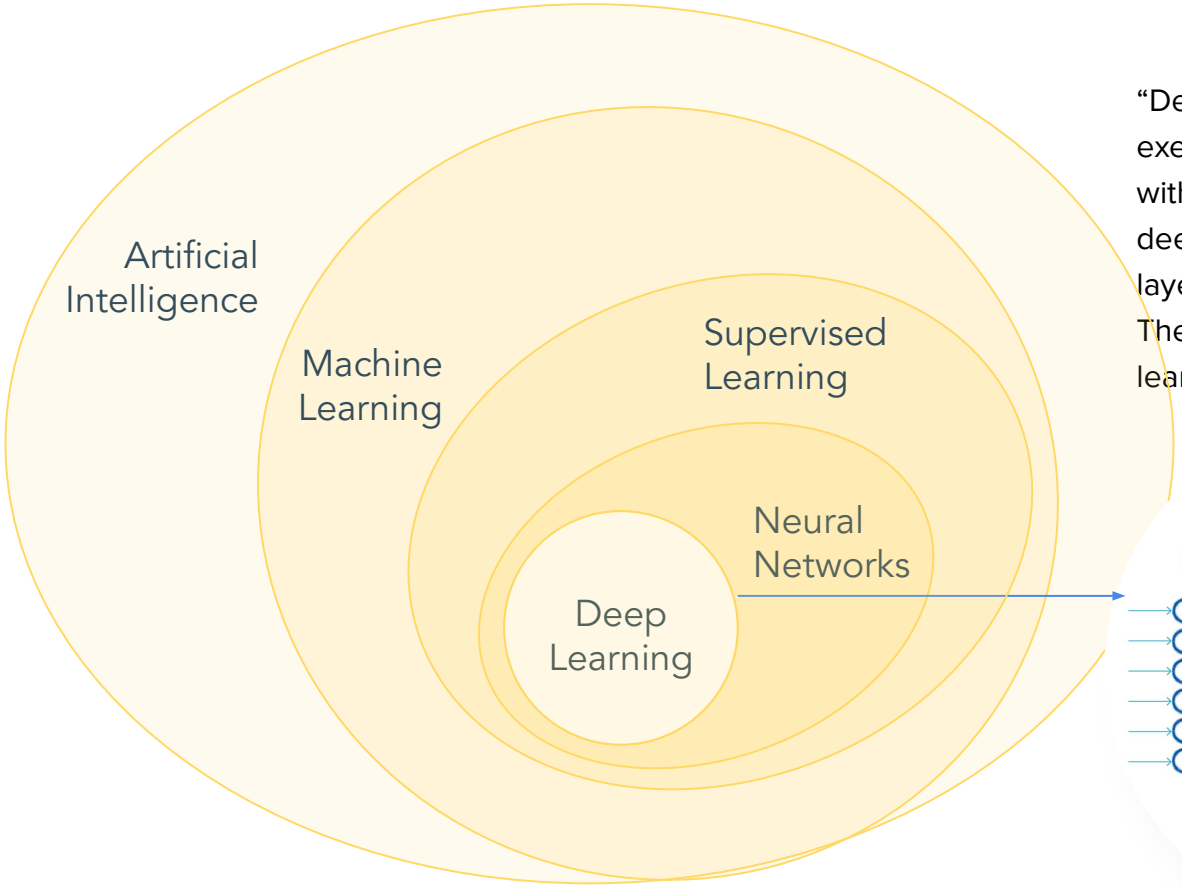


Machine learning is a broad set of techniques that enable computers to perform a task **without being explicitly programmed**.

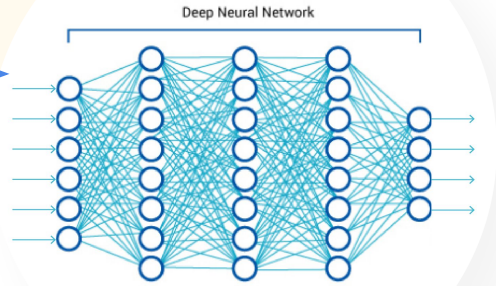


Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, partially supervised or unsupervised.



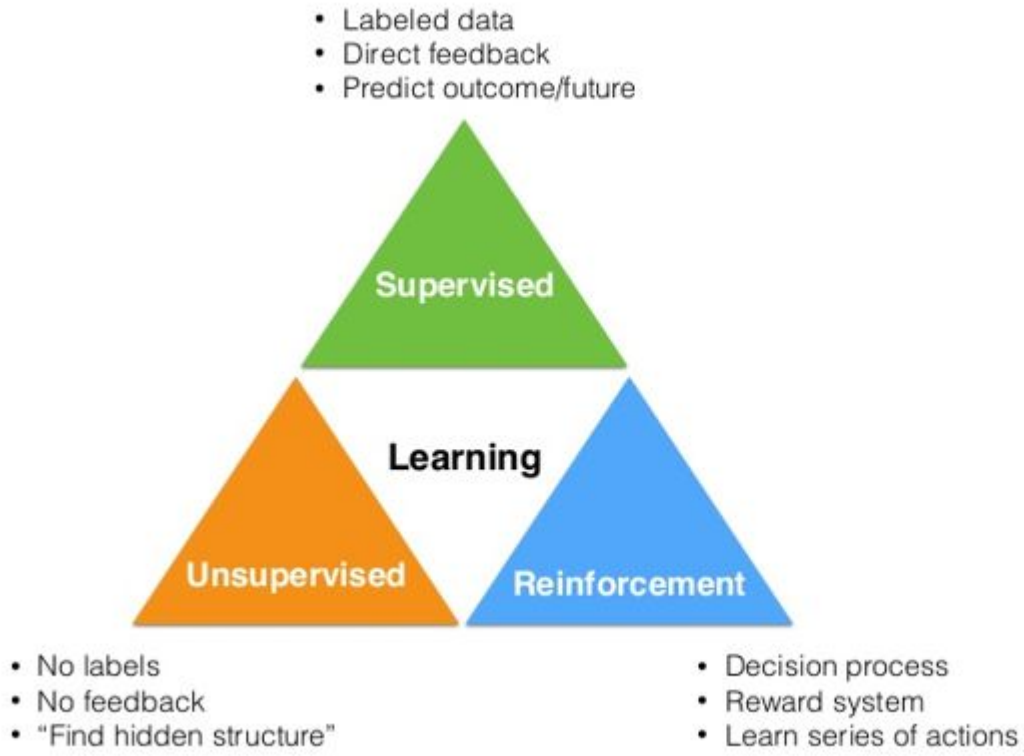


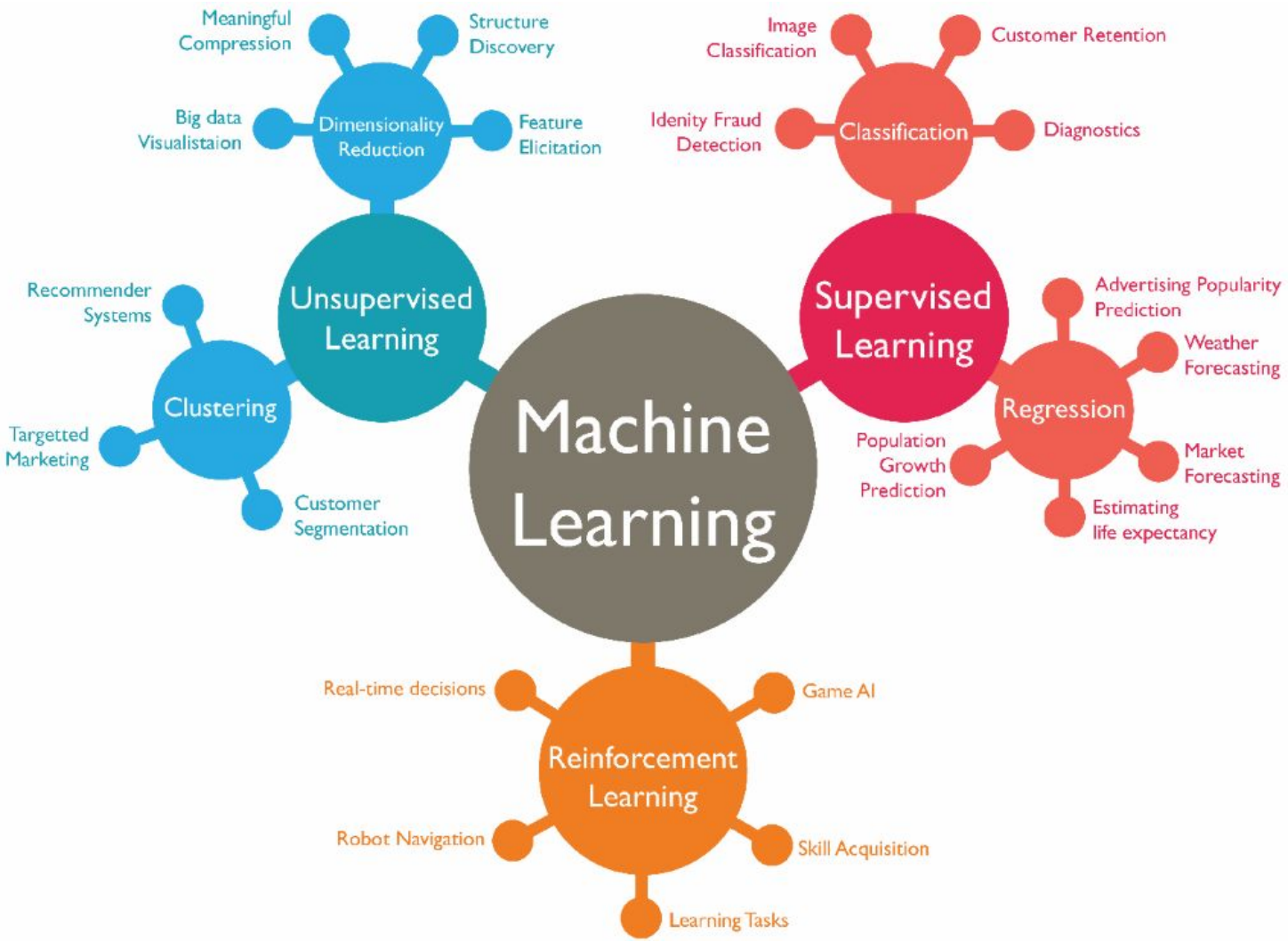
"Deep architectures are perhaps best exemplified by **multi-layer neural networks** with several hidden layers. In general terms, deep architectures are composed of multiple layers of parameterized non-linear modules. The parameters of every module are subject to learning." [Yoshua Bengio and Yann LeCun]

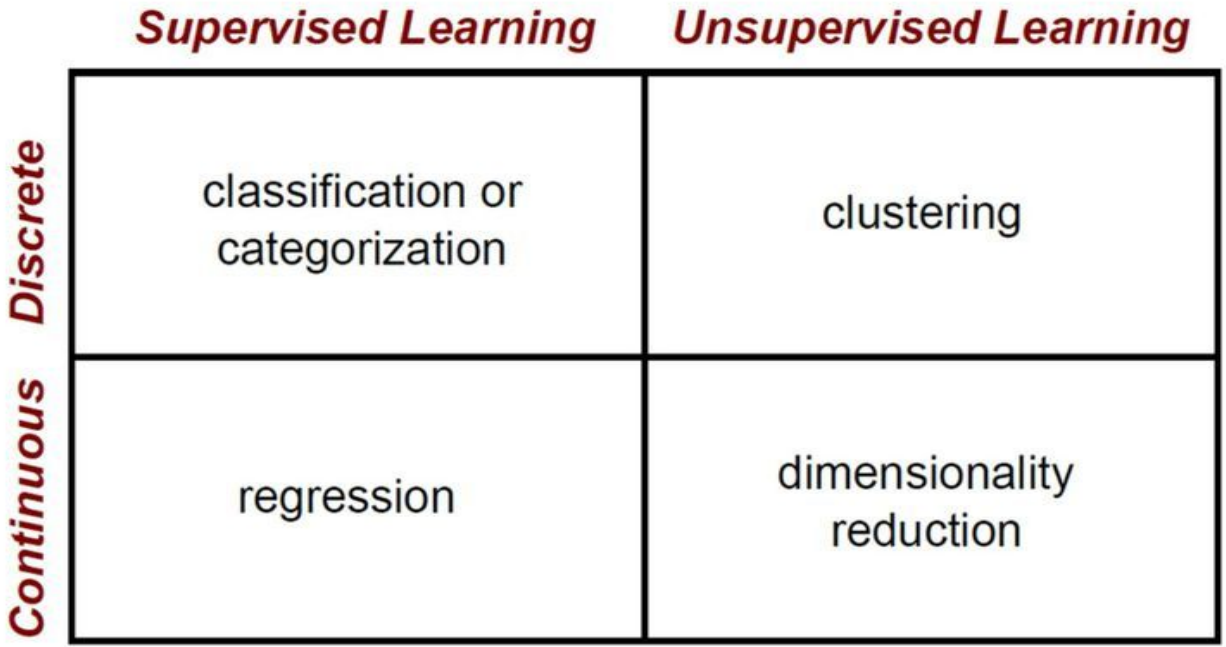


Basic concepts

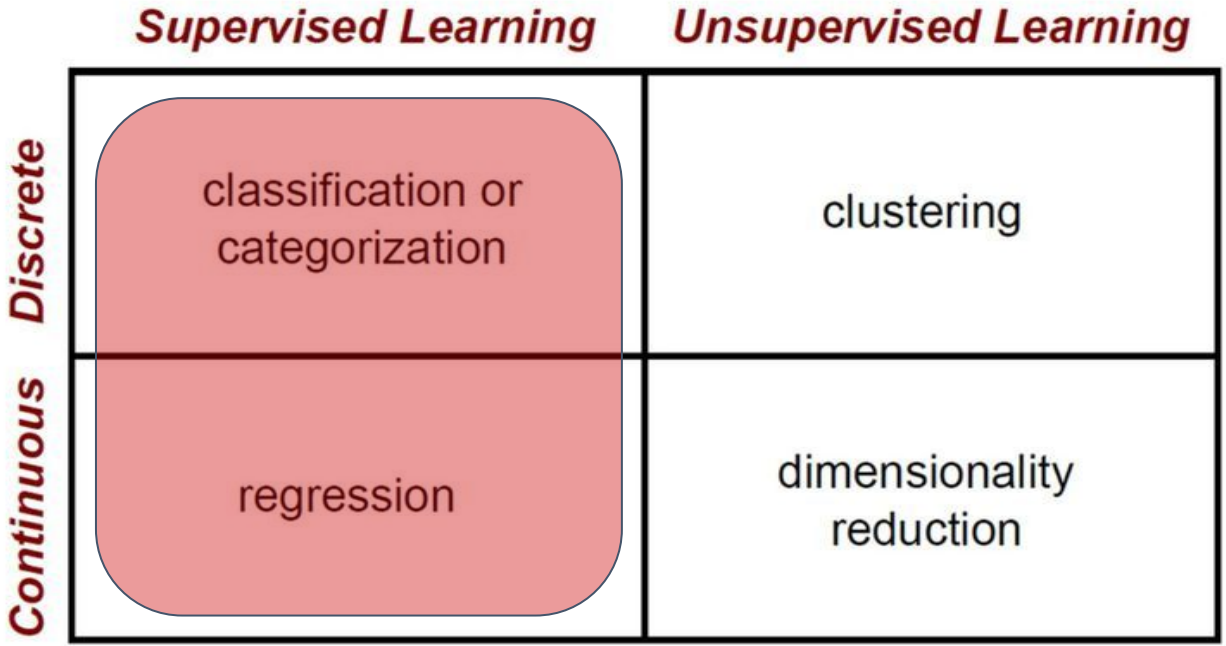
Machine learning


• Labeled data
• Direct feedback
• Predict outcome/future
Supervised
Learning
Unsupervised
Reinforcement
• No labels
• No feedback
• "Find hidden structure"
• Decision process
• Reward system
• Learn series of actions


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Feature Elicitation
Big data Visualistaion
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating life expectancy
Population Growth Prediction
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

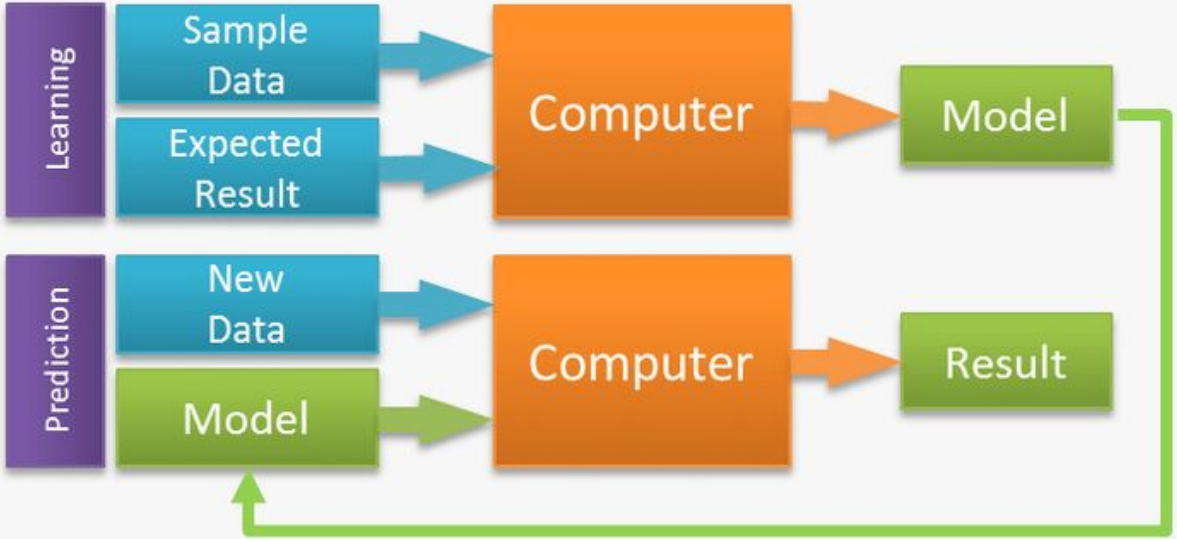

Supervised Learning
Unsupervised Learning
Discrete
Continuous
classification or categorization
clustering
regression
dimensionality reduction

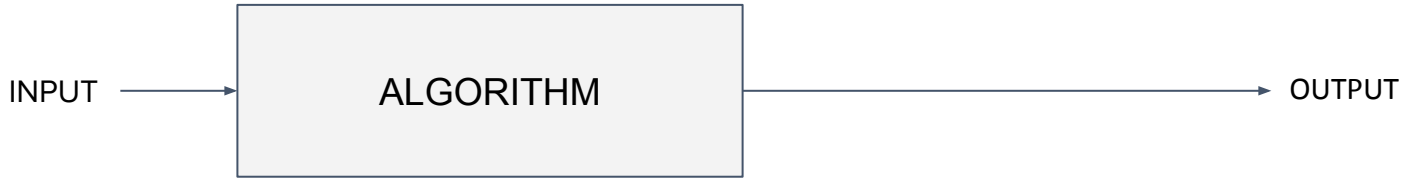
Training dataset with m samples: $(x^{(1)},y^{(1)}),..(x^{(m)},y^{(m)})$
Each $x^{(i)},y^{(i)}$ can be multi-dimensional



Traditional modeling:

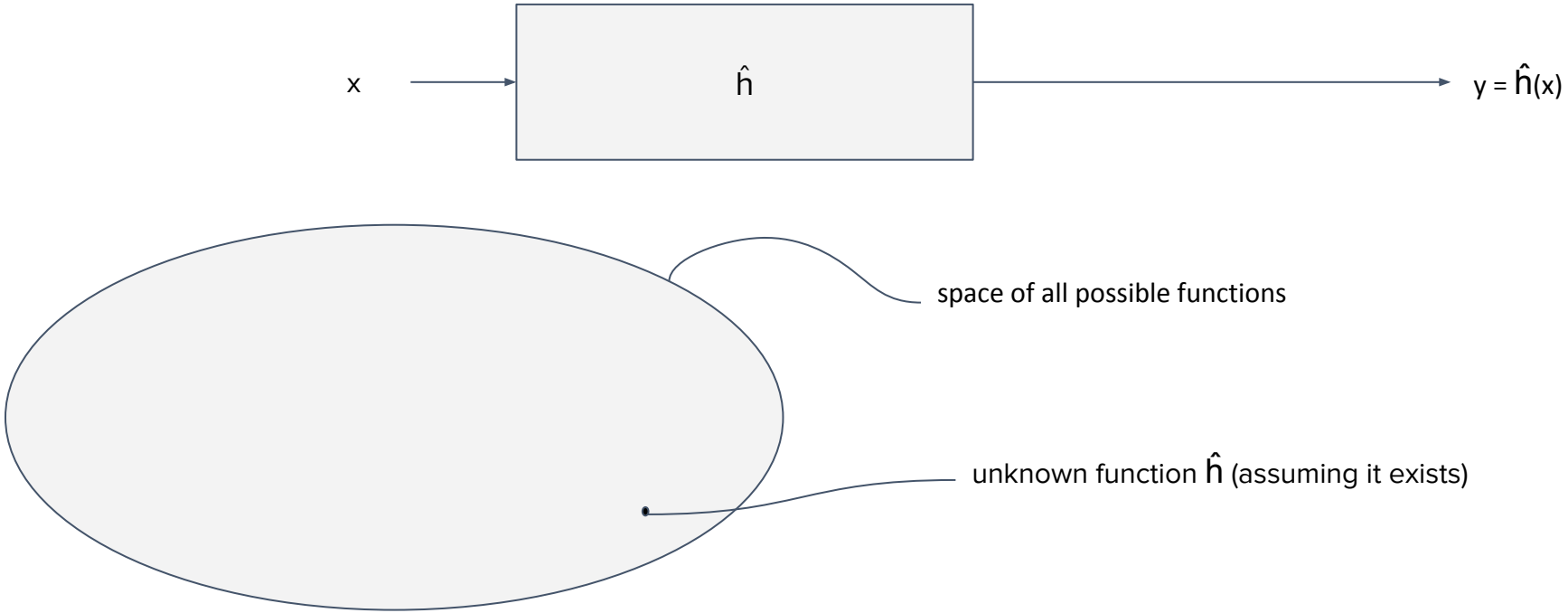
Machine Learning:

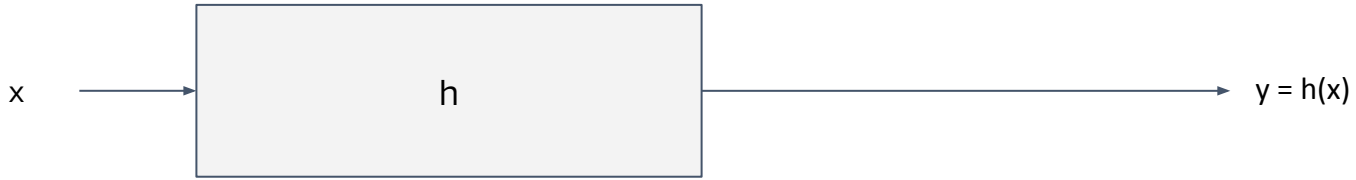




Machine learning is a broad set of techniques that enable computers to perform a task **without being explicitly programmed**.

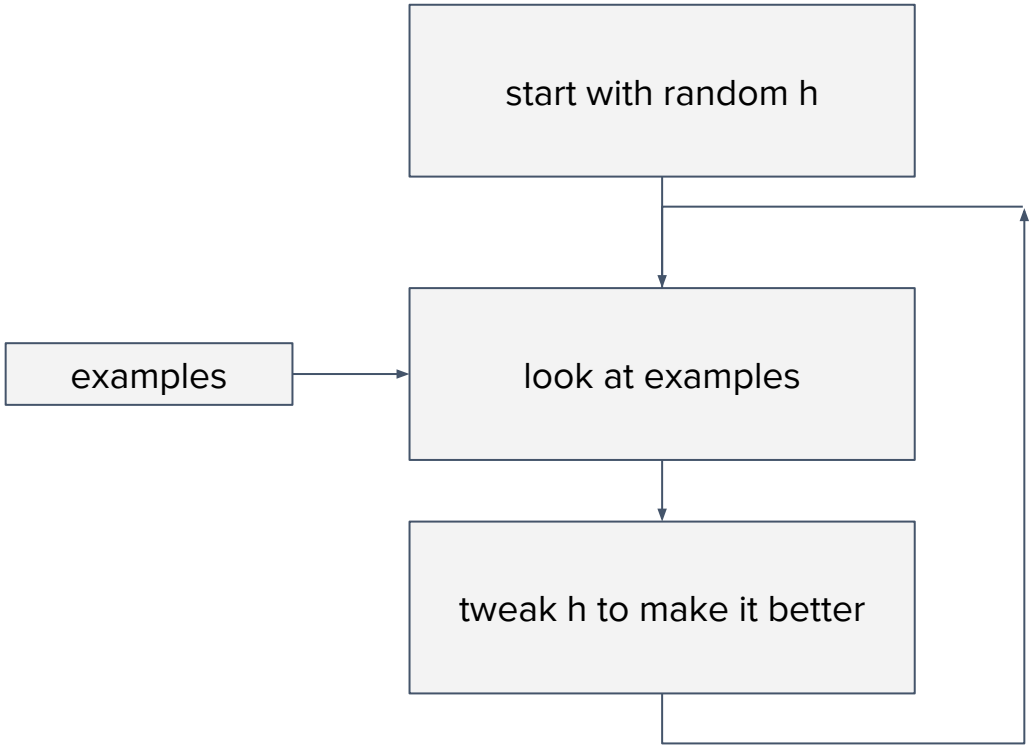
$\hat{h}$ = unknown function that implements the algorithm





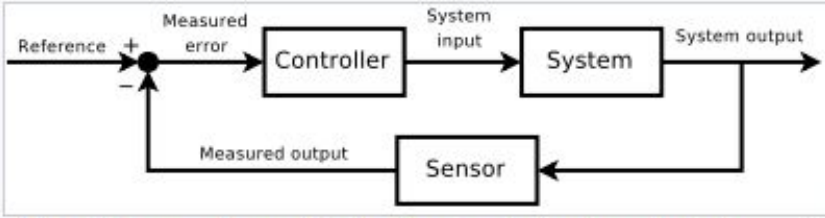
h = hypothesis function

How to find a good candidate for h ?

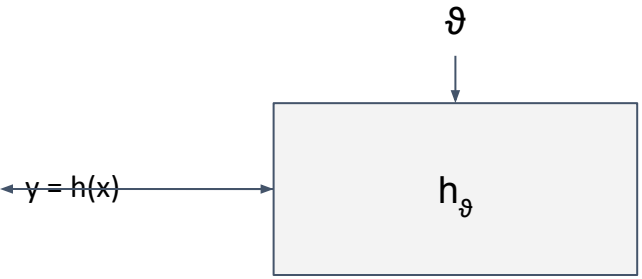

start with random h
examples
look at examples
tweak h to make it better

Control theory: harnessing the power of feedback loops.

Wide applicability in many fields: physics, biology, climate science, mechanical engineering, electronic engineering, software engineering, social sciences, economics, etc.

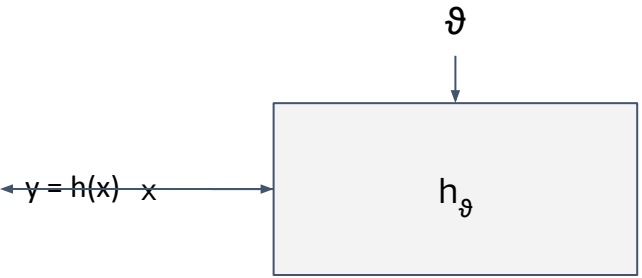


A block diagram of a negative feedback control system. Illustrates the concept of using a feedback loop to control the behavior of a system by comparing its output to a desired value, and applying the difference as an error signal to dynamically change the output so it is closer to the desired output



h = hypothesis function

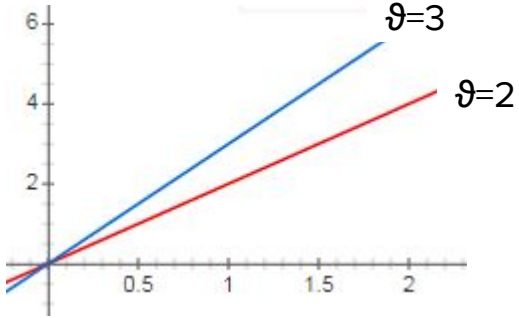
- instead of choosing a specific function for h, let's make it depend on one or more **parameters ϑ**,
  so that $h_ϑ$ represents a **SET of functions**
- by changing the parameters, the candidate h changes.



h = hypothesis function

- instead of choosing a specific function for h, let's make it depend on one or more **parameters ϑ**, so that $h_ϑ$ represents a **SET of functions**
- by changing the parameters, the candidate h changes.

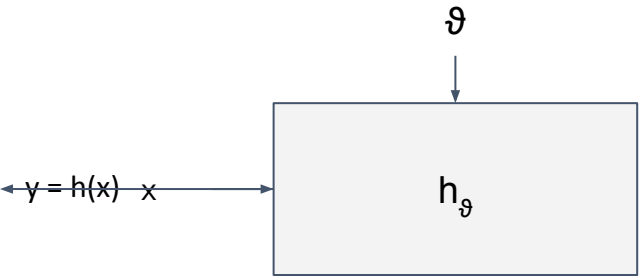
Example: h(x) = 2x ⟶ $h_ϑ(x) = ϑx$



ϑ=2 ⟶ h(x) = 2x

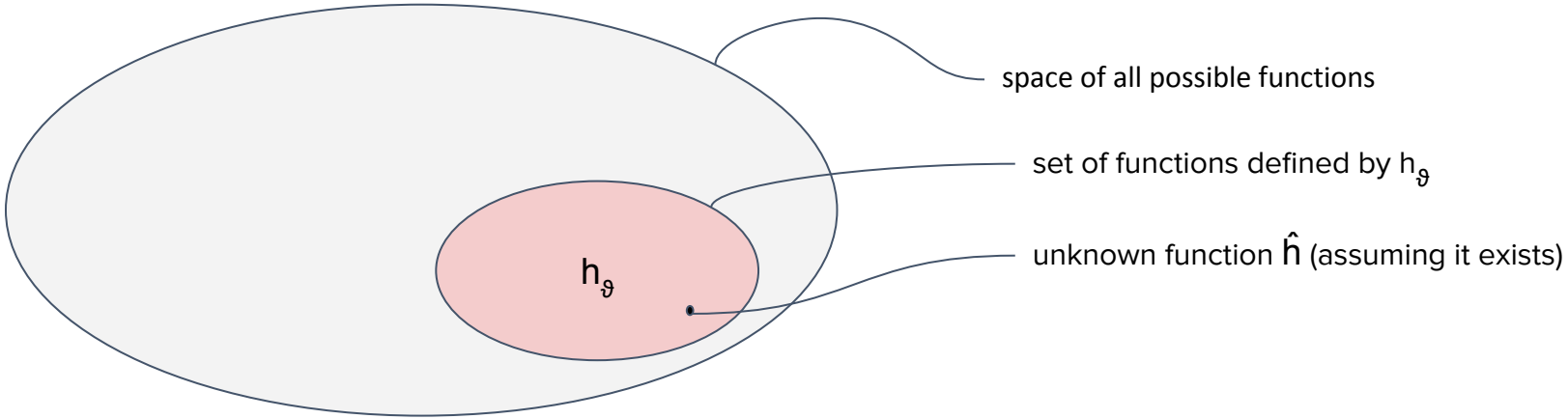
ϑ=3 ⟶ h(x) = 3x

....



h = hypothesis function

- instead of choosing a specific function for h, let's make it depend on one or more **parameters ϑ**, so that h_ϑ represents a **SET of functions**
- by changing the parameters, the candidate h changes.



How to choose the parameterized form of h ?

How to choose parameter values?

LINEAR REGRESSION

LET'S CONSIDER A FUNCTION y that can be calculated from a linear combination of the input variables (x)

When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression

KEY TERMINOLOGY

Labels

A **label** is the thing we're predicting the `y` variable in simple linear regression. The label could be the future price of a house, the kind of animal shown in a picture, the meaning of an audio clip, or just about anything.

Features

A **feature** is an input variable the `x` variable in simple linear regression. A simple machine learning project might use a single feature, while a more sophisticated machine learning project could use millions of features, specified as:

x1,x2,...xN

Examples/Instance

An **example** is a particular instance of data, **X**. (We put **X** in boldface to indicate that it is a vector.)

LINEAR REGRESSION

$$\mathcal{Y}=\vartheta(\mathcal{X})$$

Or in extended format

$$y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_n X_n$$

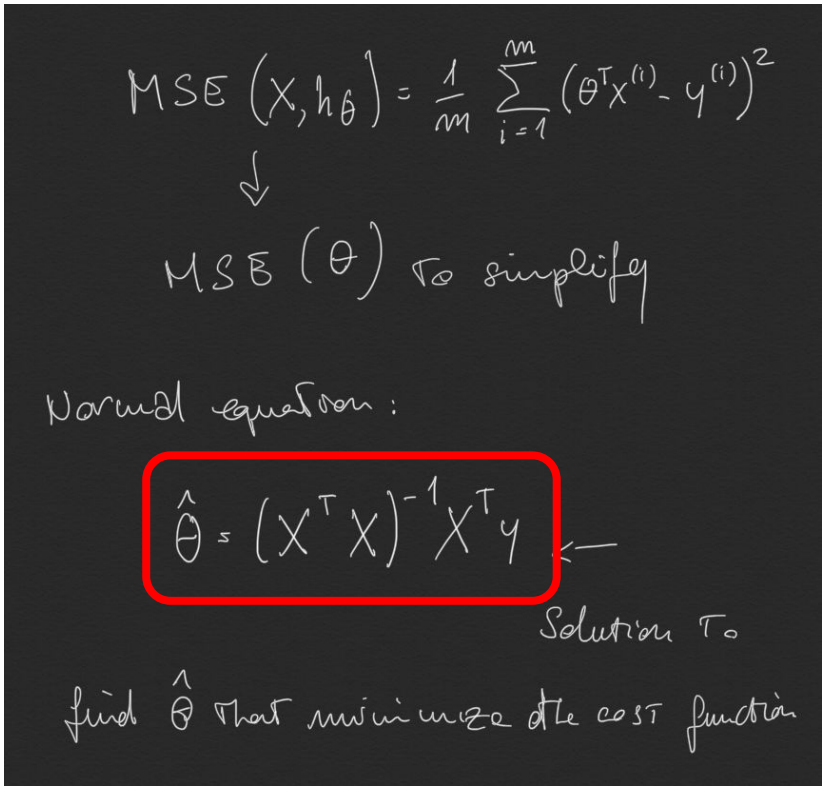
Or in vectorized form:

$$\hat{y} = h_\theta(x) = \theta \cdot x$$

Let’s find a LOSS function

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \hat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

LINEAR REGRESSION: we have a closed form eq



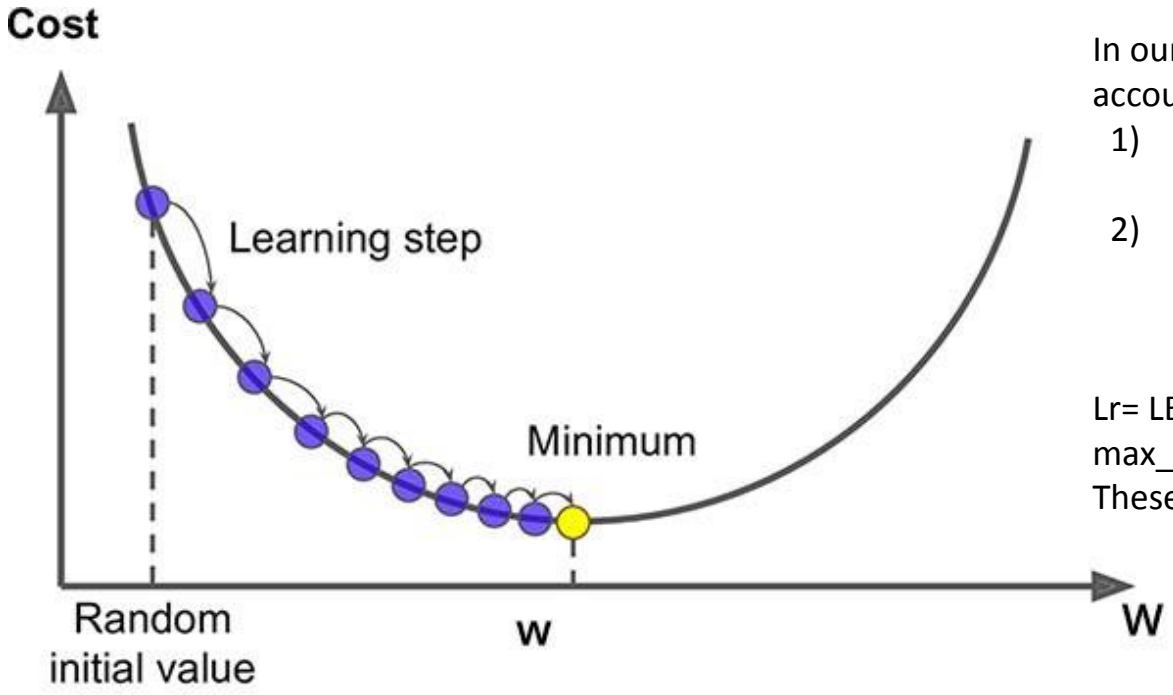
In this case we're lucky, a solution to the equation that minimizes the cost function exists ! and it's called the **NORMAL EQUATION**

To be calculated it needs the inverse matrix,
Using sci-kit-learn works with a slightly different algorithm, using the pseudoinverse called SVD: Singular Value Decomposition for efficiency reasons (speed) and since it's not always possible to invert a matrix, but you can always get a pseudo-inverse matrix.

In any case even if faster than using normal equation, SVD takes time when the features are high, so we need something better.

LET’S MOVE TO JUPYTER FOR THE DEMO

GRADIENT DESCENT

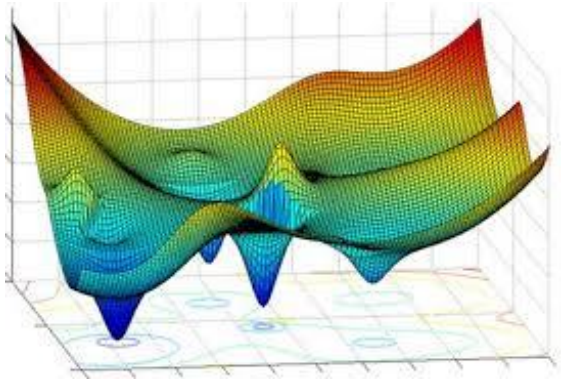


In our algorithm we need to take into account:

1) How many cicles are we willing to take to reach the minimum
2) How much should we tweak (change) our weight on a single cycle

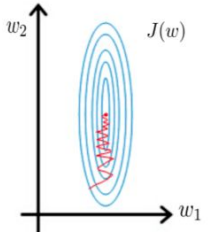
Lr= LEARNING RATE
max_iter= Max iterations
These are called **HYPERPARAMETERS**

GRADIENT DESCENT fine tuning



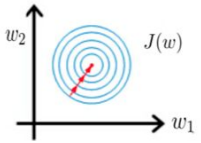
Gradient descent without scaling

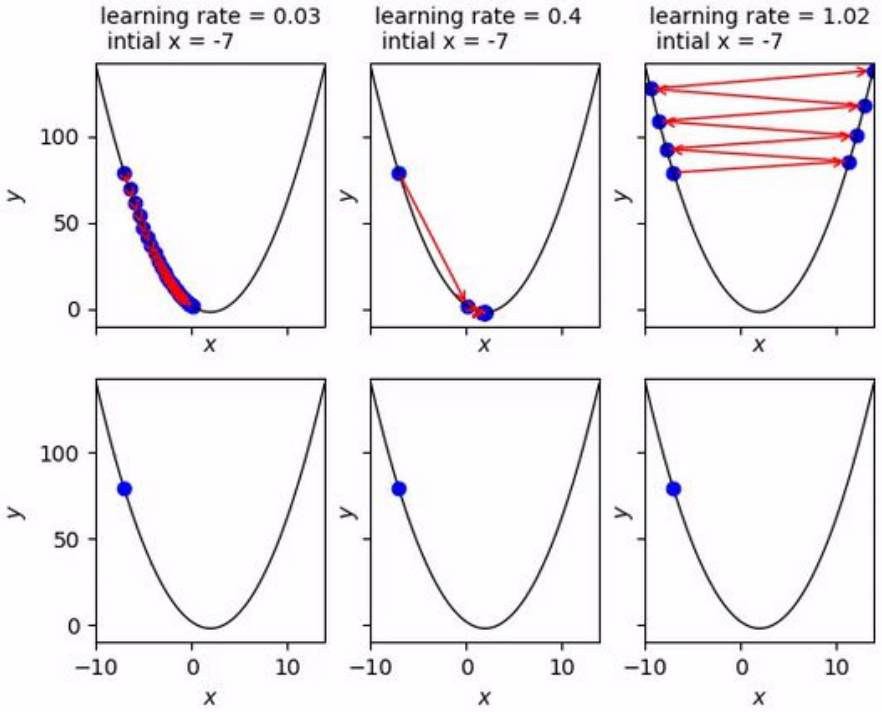
$x_1 >> x_2$



Gradient descent after scaling variables

$0 \le x_1 \le 1$
$0 \le x_2 \le 1$





GRADIENT DESCENT IMPROVED

SGD:

Note that in each iteration (also called update), only the gradient evaluated at a single point

X(i)

instead of evaluating at the set of all samples.

The key difference compared to standard (Batch) Gradient Descent is that only one piece of data from the dataset is used to calculate the step, and the piece of data is picked randomly at each step.

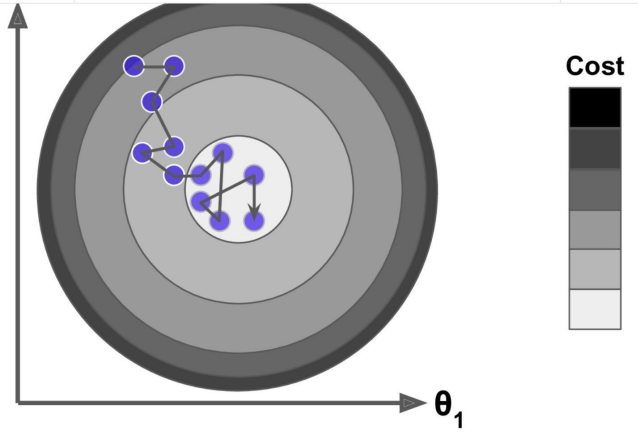


Figure 4-9. Stochastic Gradient Descent

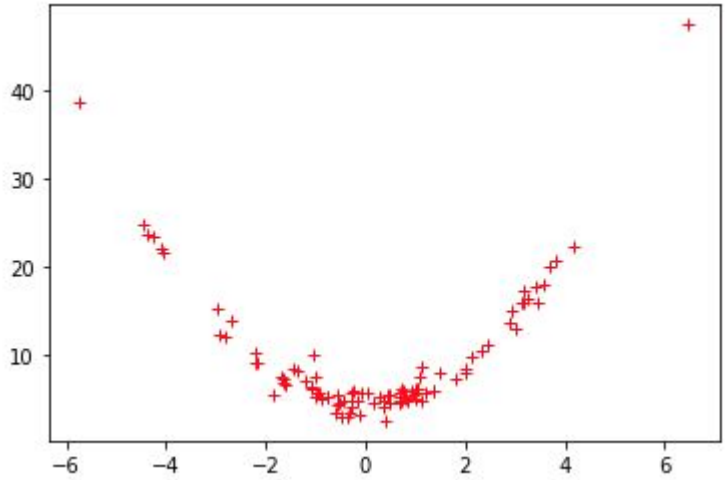
- Choose an initial vector of parameters w and learning rate η.
- Repeat until an approximate minimum is obtained:
  - Randomly shuffle examples in the training set.
  - For $i = 1, 2, \ldots, n$, do:
    - $w := w - \eta \nabla Q_i(w)$.

POLYNOMIAL REGRESSION

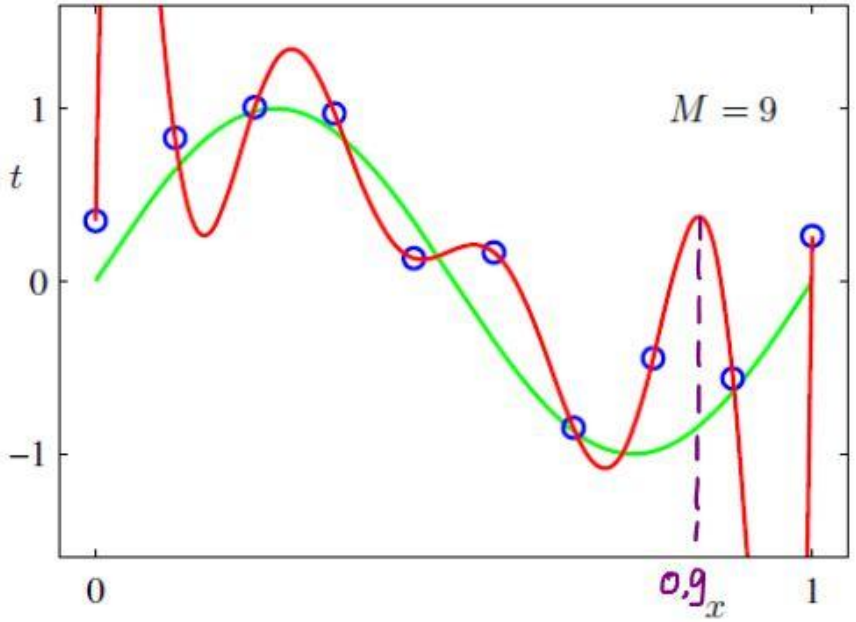
What if my data are clearly not linear?

There's a linear model that can fit non-linear Data.

Let's look at the example in JPYNB



$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n + \varepsilon.$$



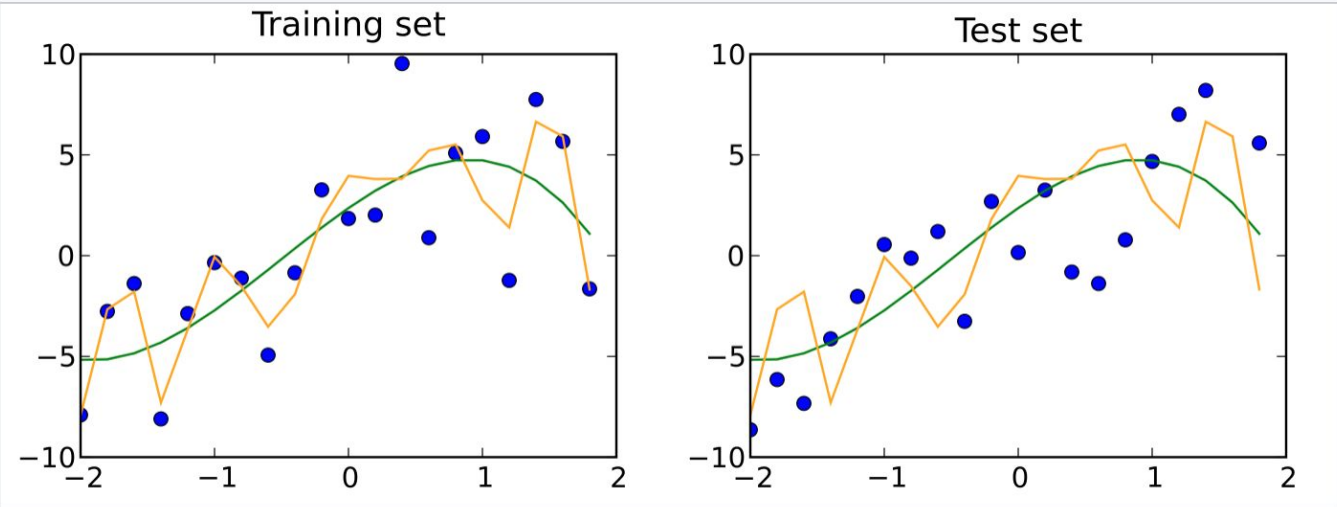
OVERFITTING

Looking at the **red** line vs the **green** line it's clear
that the model used is trying to include all the instances
losing its ability to represent a general solution.
We want to avoid that.

Test dataset [edit]

A test dataset is a dataset that is independent of the training dataset, but that follows the same probability distribution as the training dataset. If a model fit to the training dataset also fits the test dataset well, minimal overfitting has taken place (see figure below). A better fitting of the training dataset as opposed to the test dataset usually points to overfitting.

A test set is therefore a set of examples used only to assess the performance (i.e. generalization) of a fully specified classifier.[7][8]



A training set (left) and a test set (right) from the same statistical population are shown as blue points. Two predictive models are fit to the training data. Both fitted models are plotted with both the training and test sets. In the training set, the MSE of the fit shown in orange is 4 whereas the MSE for the fit shown in green is 9. In the test set, the MSE for the fit shown in orange is 15 and the MSE for the fit shown in green is 13. The orange curve severely overfits the training data, since its MSE increases by almost a factor of four when comparing the test set to the training set. The green curve overfits the training data much less, as its MSE increases by less than a factor of 2.

sklearn.model_selection.train_test_split¶

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem.

MAIN ISSUES OF ALL THE ALGORITHMS

- BIAS
- OVERFITTING
- BIAS / VARIANCE TRADE OFF

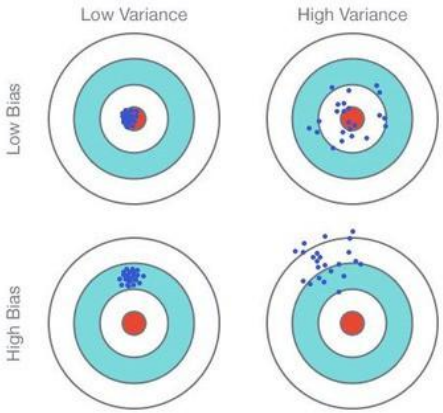
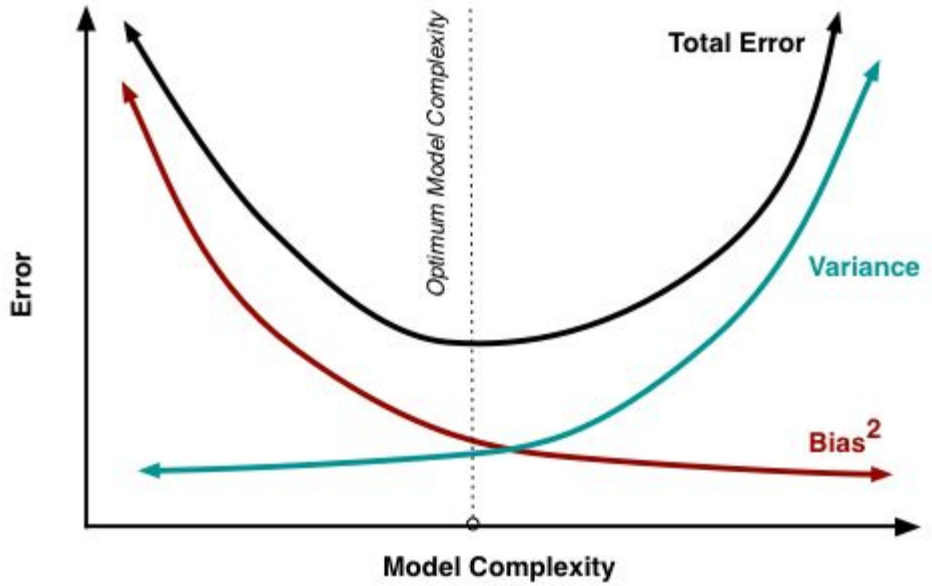


Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off



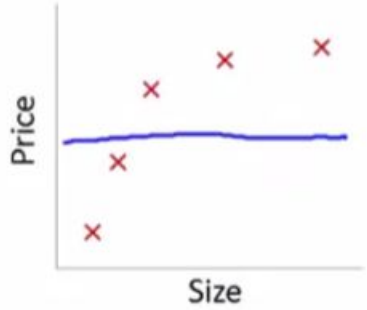
REGULARIZED LINEAR MODELS

A simple way to look at it is to think of introducing a parameter that measures complexity and a function that will increase our cost function every time we increase the complexity of the model itself.
That means finding a solution that should minimise the cost function (as usual) BUT BOOSTING the simpler solutions.
Ridge and Lasso are regularized linear models, that follow this approach, introducing hyperparameters to keep control of complexity.

Linear regression with regularization

Model: $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

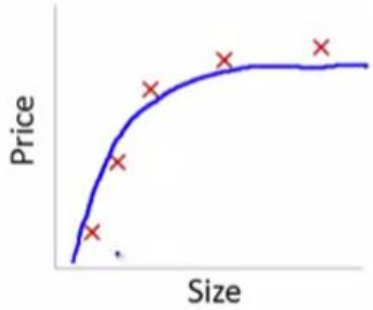


Large λ

→ High bias (underfit)

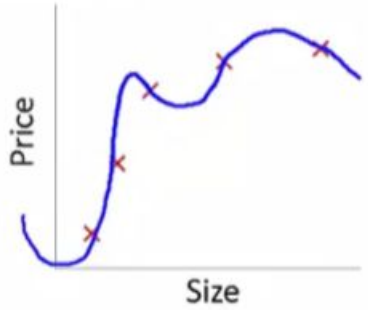
→ $\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$

$h_\theta(x) \approx \theta_0$



Intermediate λ

"Just right"



Small λ

High variance (overfit)

→ $\lambda = 0$

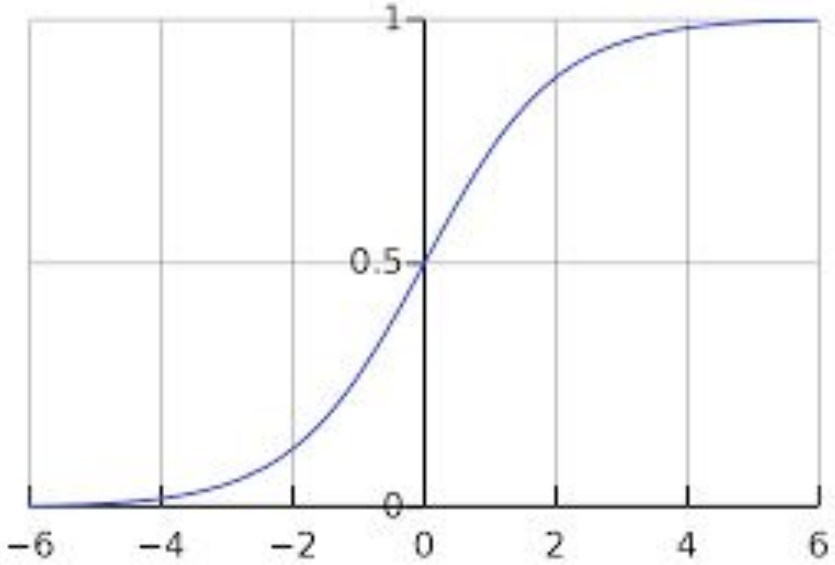
LOGISTIC REGRESSION: **Classifier**

Is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. It uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

In other words it is commonly used to estimate the probability that an instance belongs to a specific class

LOGISTIC REGRESSION

- It is a statistical approach
- It uses a sigmoid function
- It can take categorical variables
- Output will be between 0 and 1
- Used to solve classification problems



$$p = h_{\vartheta}(x) = \sigma(x^{T}\vartheta)$$

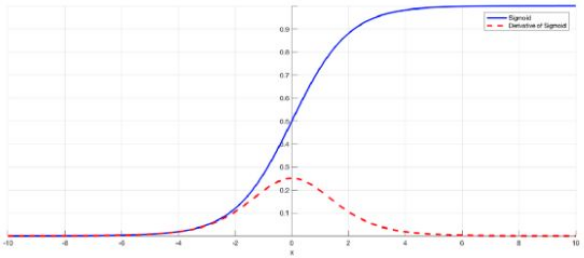
$$\sigma = 1/(1+e^{-t})$$

LOGISTIC REGRESSION MATH

- Cost function $c(\vartheta)$ called Log Loss function
- No closed form solution for that, so **NO NORMAL EQUATION** available
- But it's a convex function with one minimum !
- We can use Gradient Descent !

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

Derivative of the Sigmoid Function



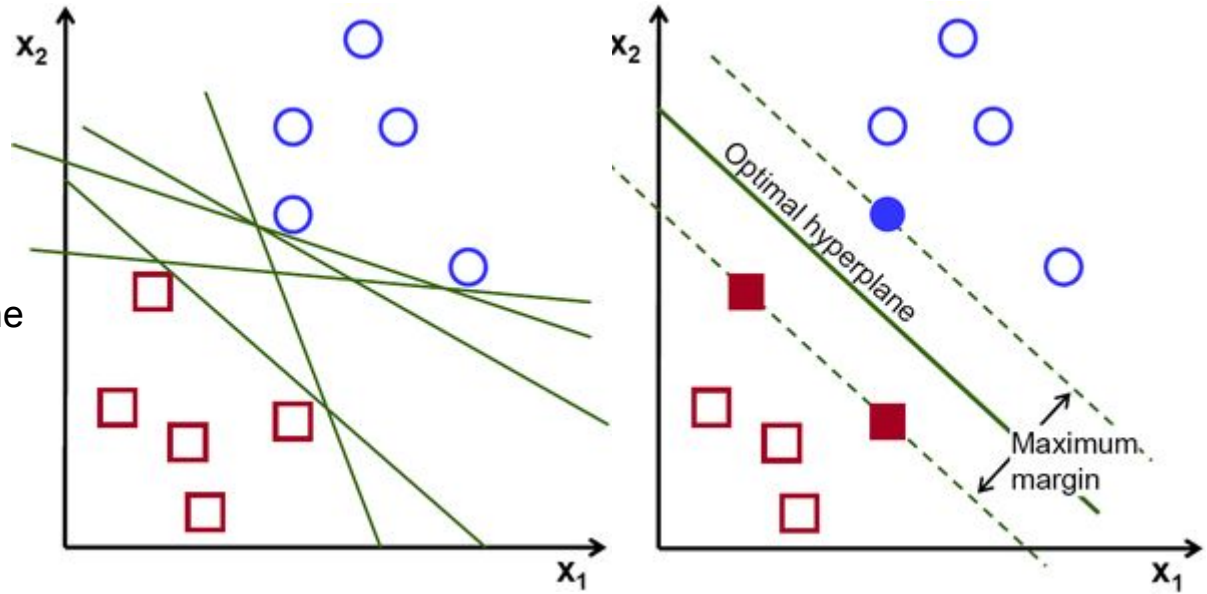
$$c(\vartheta) = \begin{cases} -\log(p) & \text{if } y=1 \\ -\log(1-p) & \text{if } y=0 \end{cases}$$

SUPPORT VECTOR MACHINES

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen.

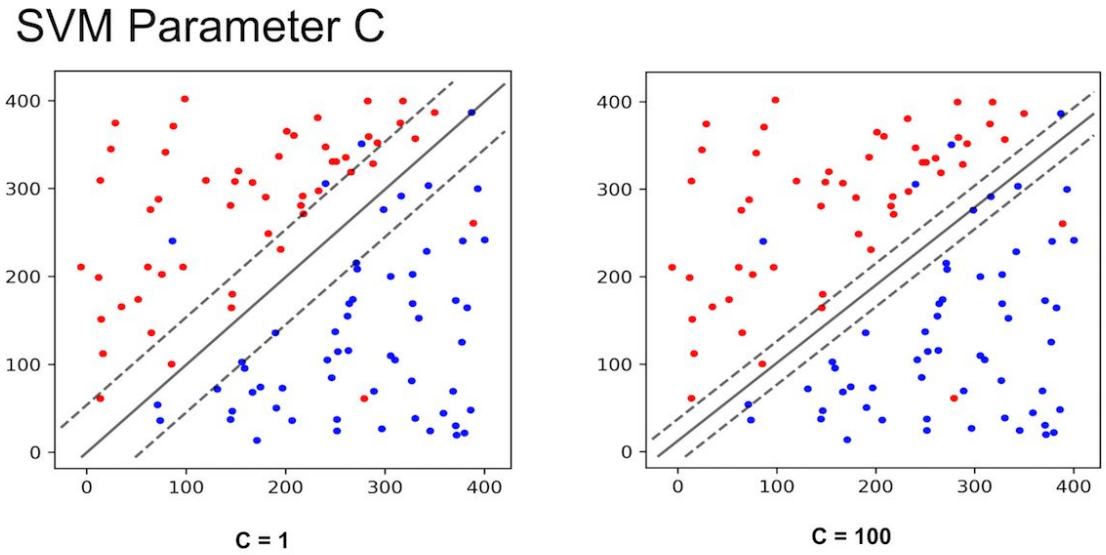
Our objective is to find a plane that has the **maximum margin**, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.



SUPPORT VECTOR MACHINES

SVM sensitivity to outliers can be set using Soft Margin, through parameter C.
Margin violation should be avoided, but in this case the model is capable of a better generalization/

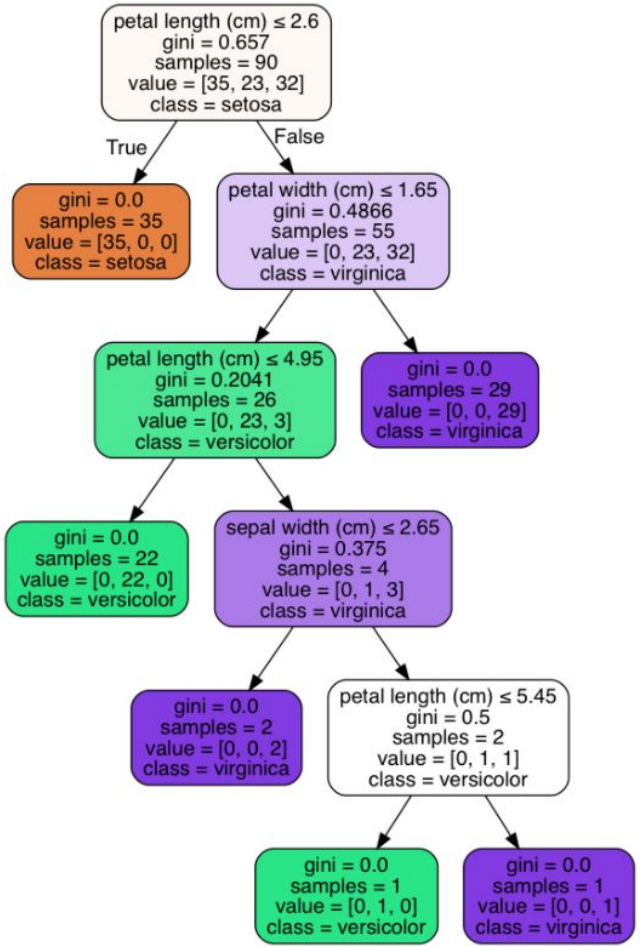
Another surprising capability of SVM is to be very flexible in case of non linearly separable data, using feature extension as Polynomial Features, or other Kernel techniques to avoid feature explosion in case of high degree polynomials.



DECISION TREES

Versatiles ML algorithms for both classification and regression.
Capable of fitting complex datasets. They are easy to understand, look at the graph on the right.

The objective is to subdivide the set with rules and try to move each class in a specific node and keep the classes separated as much a s possible.



DECISION TREES

Computationally: prediction is very fast, since it's based on simple comparisons. Generally quick also to train, unless we're talking about huge datasets.
The algorithm used to train the Decision Trees in Scikit-learn is called **CART**.

Classification and Regression Tree

Impurity Criterion

Gini Index

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

p_j: proportion of the samples that belongs to class c for a particular node

Entropy

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

p_j: proportion of the samples that belongs to class c for a particular node.

*This is the the definition of entropy for all non-empty classes ($p \neq 0$). The entropy is 0 if all samples at a node belong to the same class.

DECISION TREES

Computationally: prediction is very fast, since it's based on simple comparisons. Generally quick also to train, unless we're talking about huge datasets.
The algorithm used to train the Decision Trees in Scikit-learn is called **CART**.
Classification and Regression Tree

Main hyperparameters: max_depth, min_samples_split, min_samples_leaf...etc.

Impurity Criterion

Gini Index

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

p_j: proportion of the samples that belongs to class c for a particular node

Entropy

$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

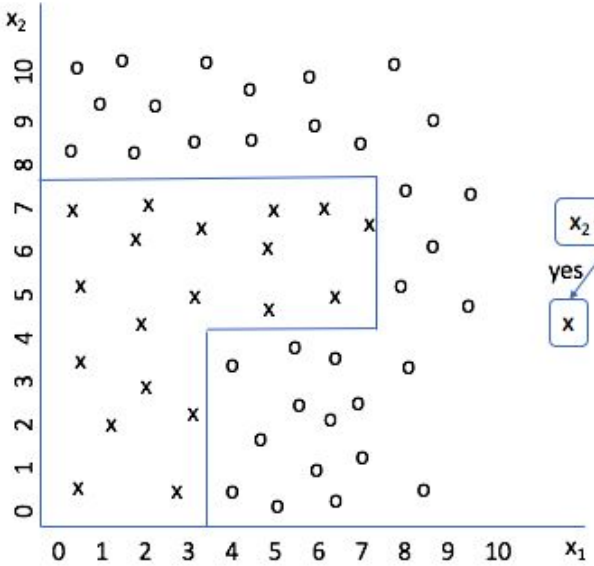
p_j: proportion of the samples that belongs to class c for a particular node.

*This is the the definition of entropy for all non-empty classes ($p \neq 0$). The entropy is 0 if all samples at a node belong to the same class.

DECISION TREES

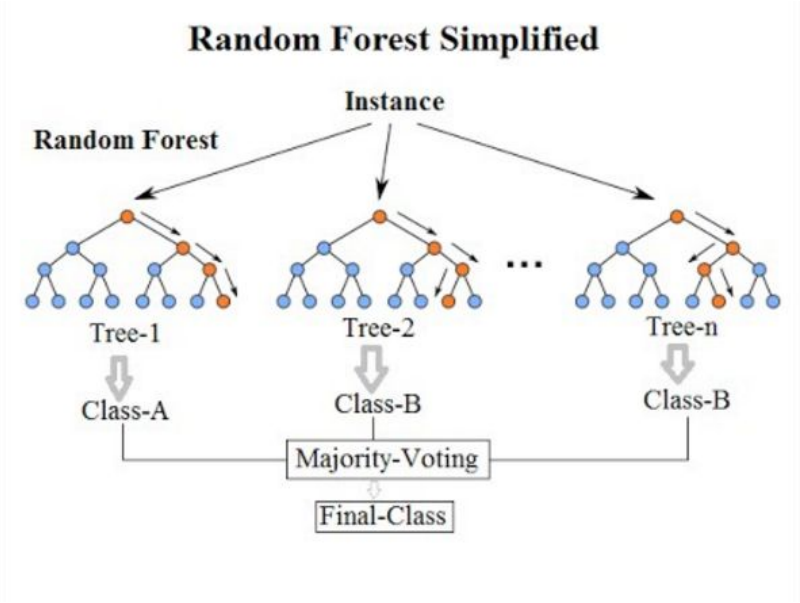
Interesting facts about decision trees:

- non parametric models
- white box Machine Learning
- low tuning of hyperparameters
- no need to normalize data
- tend to overfit
- high variance ➜ unstable



RANDOM FOREST

Random Forest is the first example of ENSEMBLE LEARNING

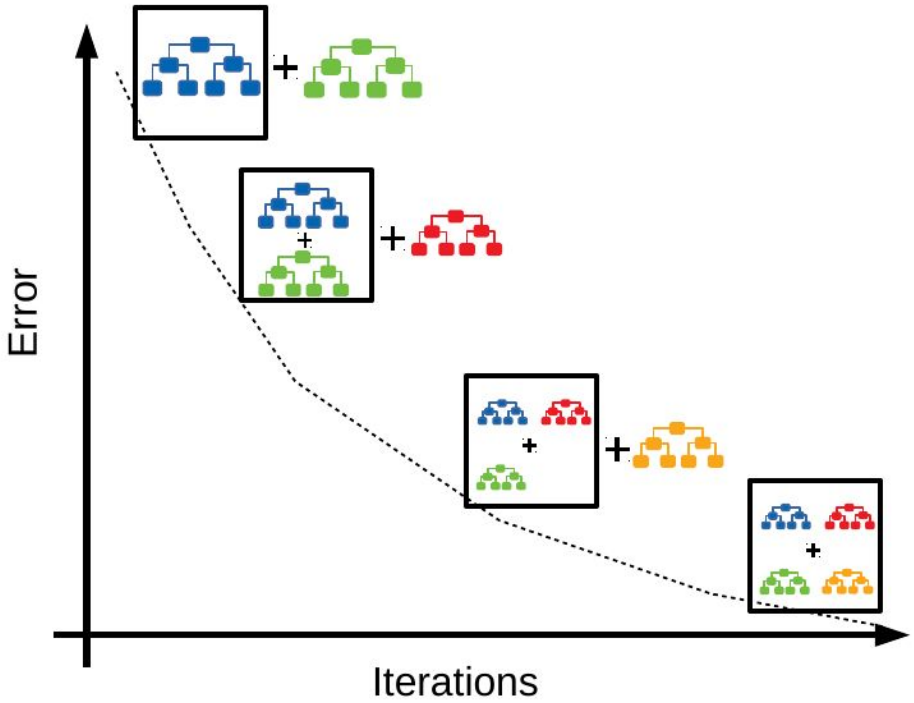


- Train many decision trees, each one on a random sample data
- Combine their output:
    - Regression: **AVERAGE**
    - Classification: **MODE**

GRADIENT BOOSTING

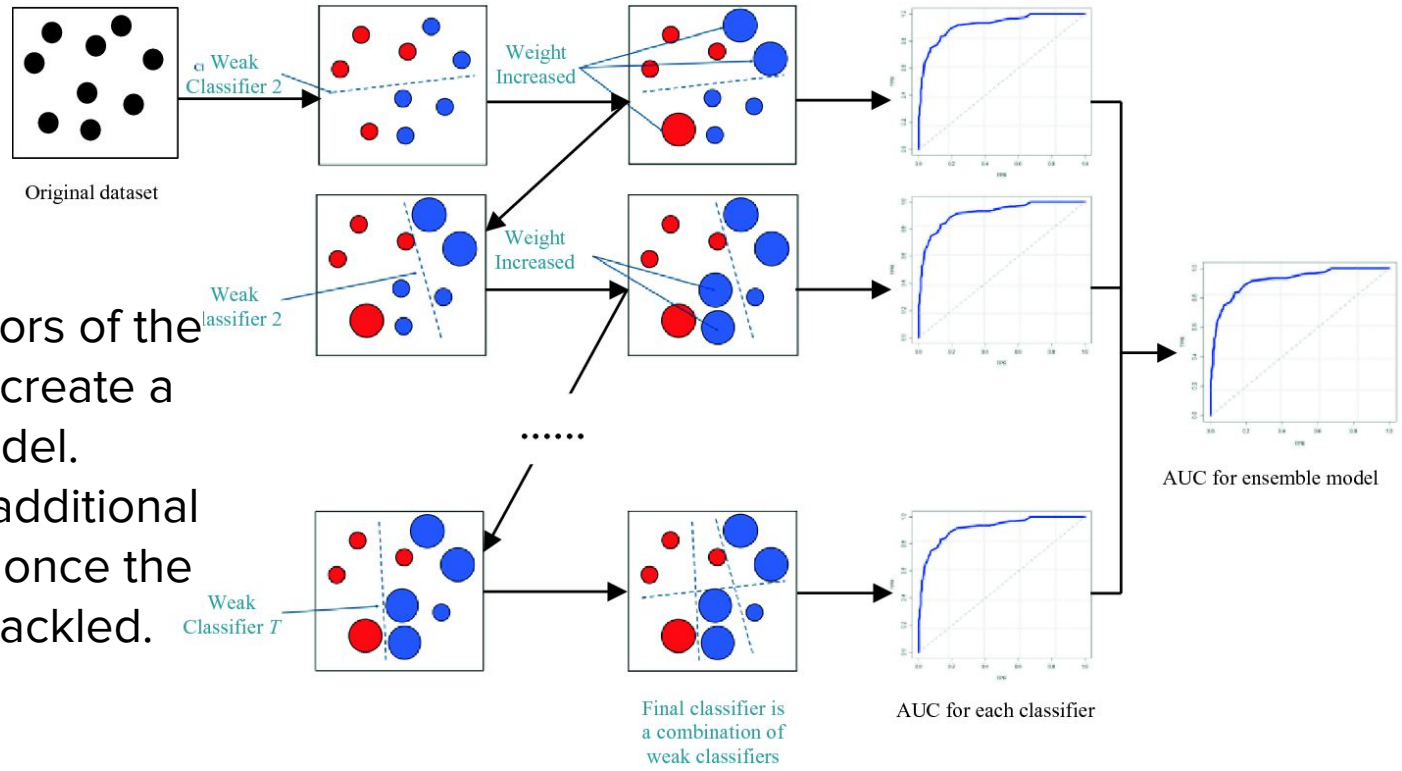
Is a form of ENSEMBLE LEARNING and uses multiple weak learners to create a strong learner

-



GRADIENT BOOSTING

Working on the errors of the previous model to create a better decision model.
The weight of the additional trees are modified once the errors are getting tackled.



UNSUPERVISED MODELS: CLUSTERING

Unsupervised Learning:

No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

WHY UNSUPERVISED LEARNING

Why Unsupervised Learning is needed despite of these issues?

- Annotating large datasets is very costly and hence we can label only a few examples manually.

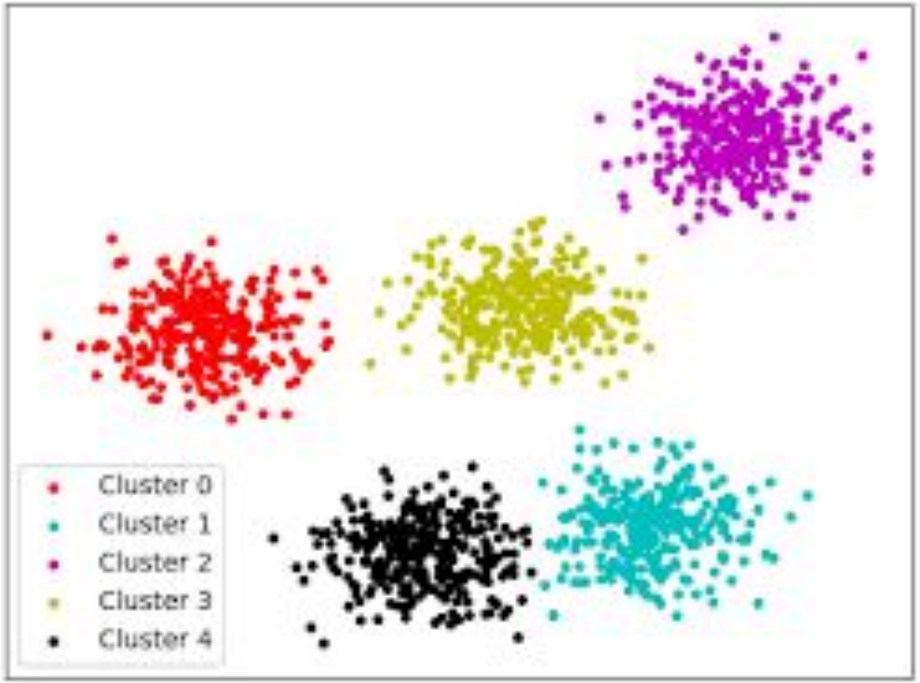
  Example: Speech Recognition
- There may be cases where we don’t know how many/what classes is the data divided into.

  Example: Data Mining
- We may want to use clustering to gain some insight into the structure of the data before designing a classifier.

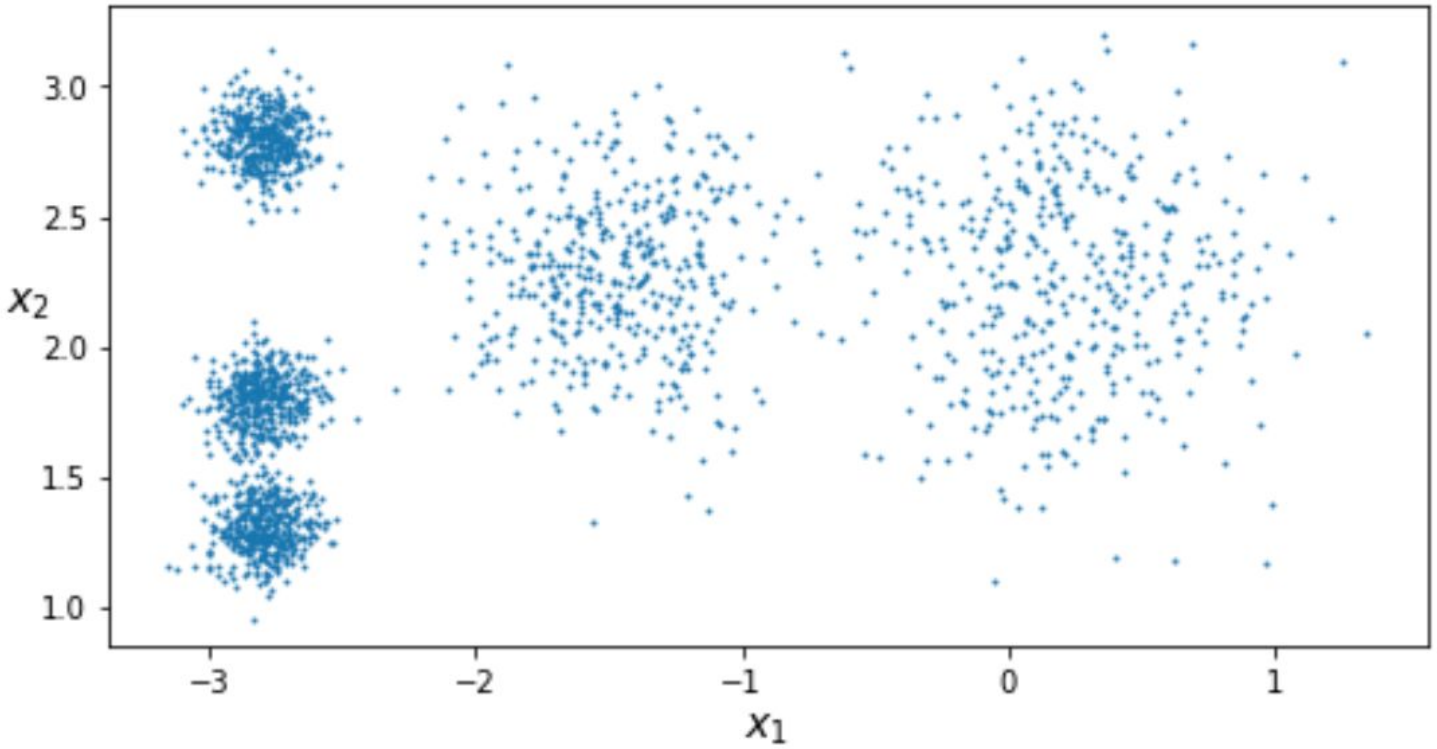
CLUSTERING

Uses:

- customer segmentation
- data analysis
- dimensionality reduction technique
- for anomaly detection
- for search engines
- to segment an image
- for semi-supervised learning



K MEANS ALGORITHMS

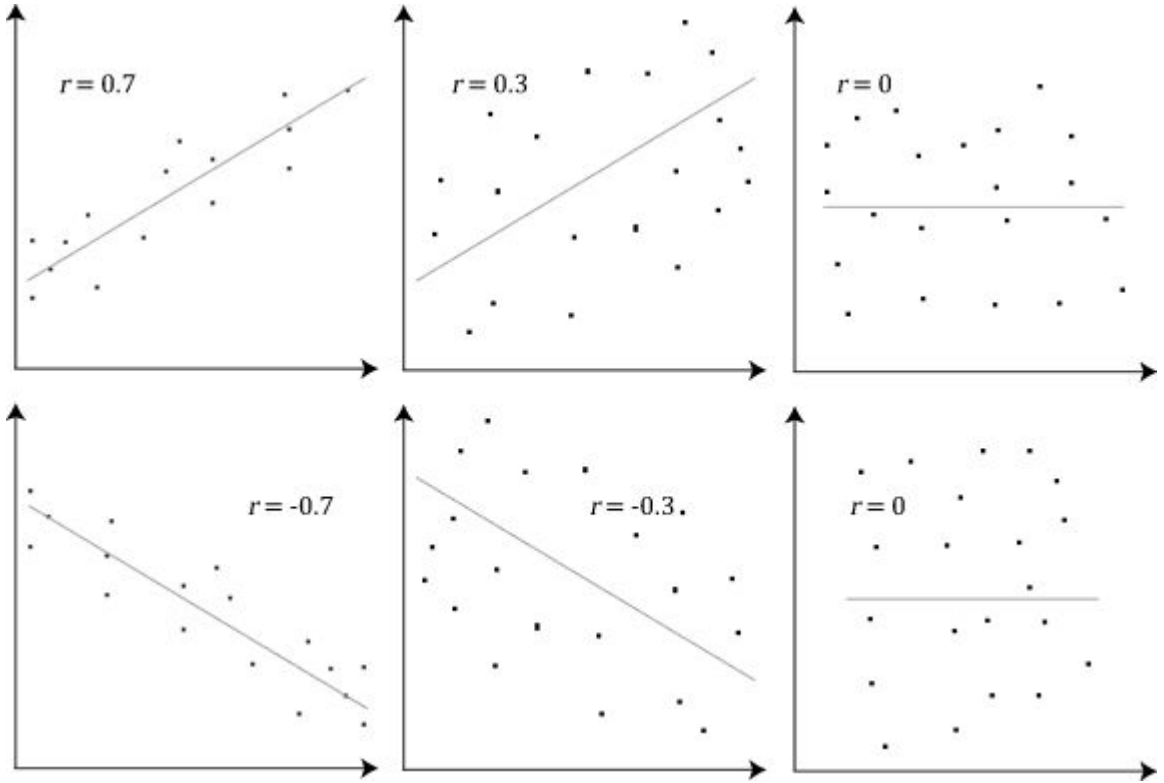


HYPER PARAMETERS OPTIMIZATION WITH GridSearch

Grid search is essentially an optimization algorithm which lets you select the best parameters for your optimization problem from a list of parameter options that you provide, hence automating the 'trial-and-error' method. Although it can be applied to many optimization problems, it is most popularly known for its use in machine learning to obtain the parameters at which the model gives the best accuracy.

PEARSON CORRELATION COEFFICIENT

$$r = \frac{1}{n-1}\sum\left(\frac{x-\bar{x}}{s_x}\right)\left(\frac{y-\bar{y}}{s_y}\right)$$



Measure	Populations	Samples
Size	N	n
Mean	$\mu = \frac{\sum x}{N}$	$\bar{x} = \frac{\sum x}{n}$
Variance	$\sigma^2 = \frac{\sum (x-\mu)^2}{N}$	$s^2 = \frac{\sum (x-\bar{x})^2}{n-1}$
Standard Deviation	$\sigma = \sqrt{\frac{\sum (x-\mu)^2}{N}}$	$s = \sqrt{\frac{\sum (x-\bar{x})^2}{n-1}}$

THE WORKFLOW 1

Data Preprocessing

Select Algorithm & Framework

Train & Tune Your Model

Integrate & Deploy

THE WORKFLOW 2

Select & Prepare Training Data

Choose & Optimize Your ML Model

Setup & Manage Environment For Training

Train & Tune Model (Trail & Error)

Deploy Your ML Model To Production

Scale & Manage Production Environment