ML (weak AI) ~~(weak AI)~~

Supervised
Eg: Face Detection
identification
of numbers, animals
etc.

Unsupervised
Eg: classify photos
w & w/o focus.
without explicit labels
E-commerce - classify
customers in groups

Reinforcement
Learning to perform
actions in an environment
eg: Self-driving cars
Computer Games,
Robotics.

Measuring AI - Turing Test

ML initially started as a quest for AI
& mimicking brain dynamics,
but has now evolved to solving practical
problems using algos which have
little or no resemblance to brains.

<u>Debate</u>: AI ⊂ ML or ML ⊂ AI
or AI & ML just have an overlap.

Goal of ML → Generalisation of learning to new &
unknown <u>datasets</u> & <u>scenarios</u> (AGI).
↳ pre-trained + last layer

often requires domain knowledge

Bias - Variance Tradeoff | Decomposition -
~~best~~ way to estimate generalisation error.

Bias: wrong assumption about training data (underfitting)
Variance: High sensitivity to small fluctuation & random noise.
(~~this~~ overfitting)

$\hat{f}$: Learned function

$y = f(x) + \epsilon$
↳ Actual function.

ϵ: zero mean
σ^2 variance.

$$\text{Bias} = E_D[\hat{f}] - f$$

$$\text{Variance} = E_D\left[\left(E_D[\hat{f}] - \hat{f}\right)^2\right]$$

Expected Error on unseen sample:

$$\text{Bias}^2 + \text{Var} + \sigma^2$$
↳ Var of <u>y</u> or ϵ

Supervised Learning : Regression vs. Classification

ML is another form of ~~curf~~ curve fitting, or function approximation, but where the functional form of the curve is not explicitly obvious.
[not binary but gray scale]

Linear Regression ⟶ Linear classification Models
[Logistic, Linear SVM]
↓
~~Probabilistic~~ Non-linear classification models
(SVM with kernels, Decision Trees)
↓
ANN ⟶ Deep Learning.

Model encompassing a broad family of functions & changing parameters helps in navigating this set of functions.

Learning = Training + Testing
Generalisation

Underfitting : High Bias & Low variance

Overfitting : Low Bias & High Variance
↳ reduce model complexity
use regularisation

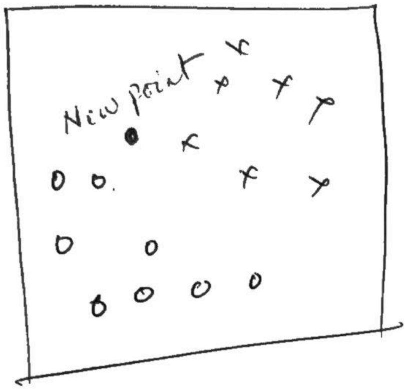
	Testing Error High	Testing Error Low
Training Error High	Underfitting	Fluke
Training Error Low	Overfitting	Ideal Good fit

- Naive Bayes' & Bayes' optimal classifier
- Maximum Likelihood Estimator & Maximum Aposteriori Probability
- Linear Regression
- Logistic Regression
- Support Vector Machines
- Decision Trees

~~ANNs~~

- K-nearest Neighbours
- ANNs.

K- Nearest Neighbours



Measure distance from K- nearest neighbours & classify based on majority rule.

Curse of Dimensionality

Consider a D-dimensional unit cube.

If we want to do our KNN estimate based on a fraction, f, of data points, we need to take a cube around our test point of edge size,

$$e_D(f) = f^{1/D}$$

If $f = 1\% = 0.01$,

$D = 3 \Rightarrow e_D(f) = 0.215$

$D = 10 \Rightarrow e_D(f) = 0.63 \rightarrow$ very large cube!!

$D = 100 \Rightarrow e_D(f) = 0.96$!!!

Blessing of Dimensionality

If Signal to Noise ratio is high.

High Dimensional data ~~has~~ can have very interesting & geometric properties which could be exploited for improving accuracy.

KNN - Algo - pseudo - code

		Predicted 0	Predicted 1
Actual	0	TN	FP
	1	FN	TP

Accuracy -

$$\frac{TN + TP}{Total}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F1 = \frac{2}{1/\text{Prec} + 1/\text{Recall}}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$: True Positive Rate

(Sensitivity)

$$\text{Specificity} = \frac{TN}{TN + FP}$$: True Negative Rate

~~Sensitivity~~