

Big data science Day 2



F. Legger - INFN Torino

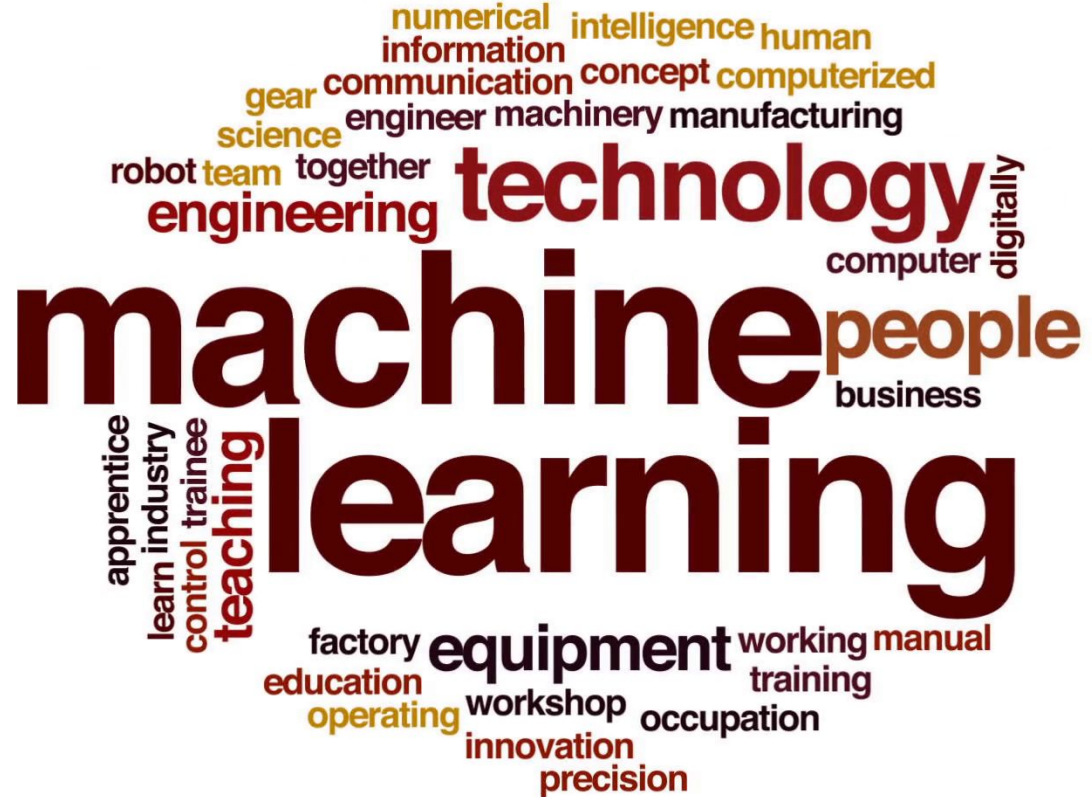
<https://github.com/Course-bigDataAndML/MLCourse-INFN-2022>

Yesterday

- Big data
- Analytics

Today

- Machine learning
 - Supervised models
 - Unsupervised models
 - Best practices



A PROPOSAL FOR THE
DARTMOUTH SUMMER RESEARCH PROJECT
ON ARTIFICIAL INTELLIGENCE

J. McCarthy, Dartmouth College
M. L. Minsky, Harvard University
N. Rochester, I. B. M. Corporation
C. E. Shannon, Bell Telephone Laboratories

*Our ultimate objective
is to make programs
that learn from their
experience as
effectively as humans
do*

[John McCarthy, 1958]

August 31, 1955

ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.



MACHINE LEARNING

Machine learning begins to flourish.



DEEP LEARNING

Deep learning breakthroughs drive AI boom.



1950's 1960's 1970's 1980's 1990's 2000's 2010's

Machine Learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead

[Wikipedia]

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 task in T , as measured by P , improves with experience E

[Tom Mitchell, 1997]

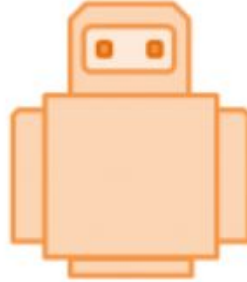
Machine Learning is the science of getting computers to act without being explicitly programmed

[Andrew Ng]

Machine Learning

Input Data

Information (+ Answers)



Output

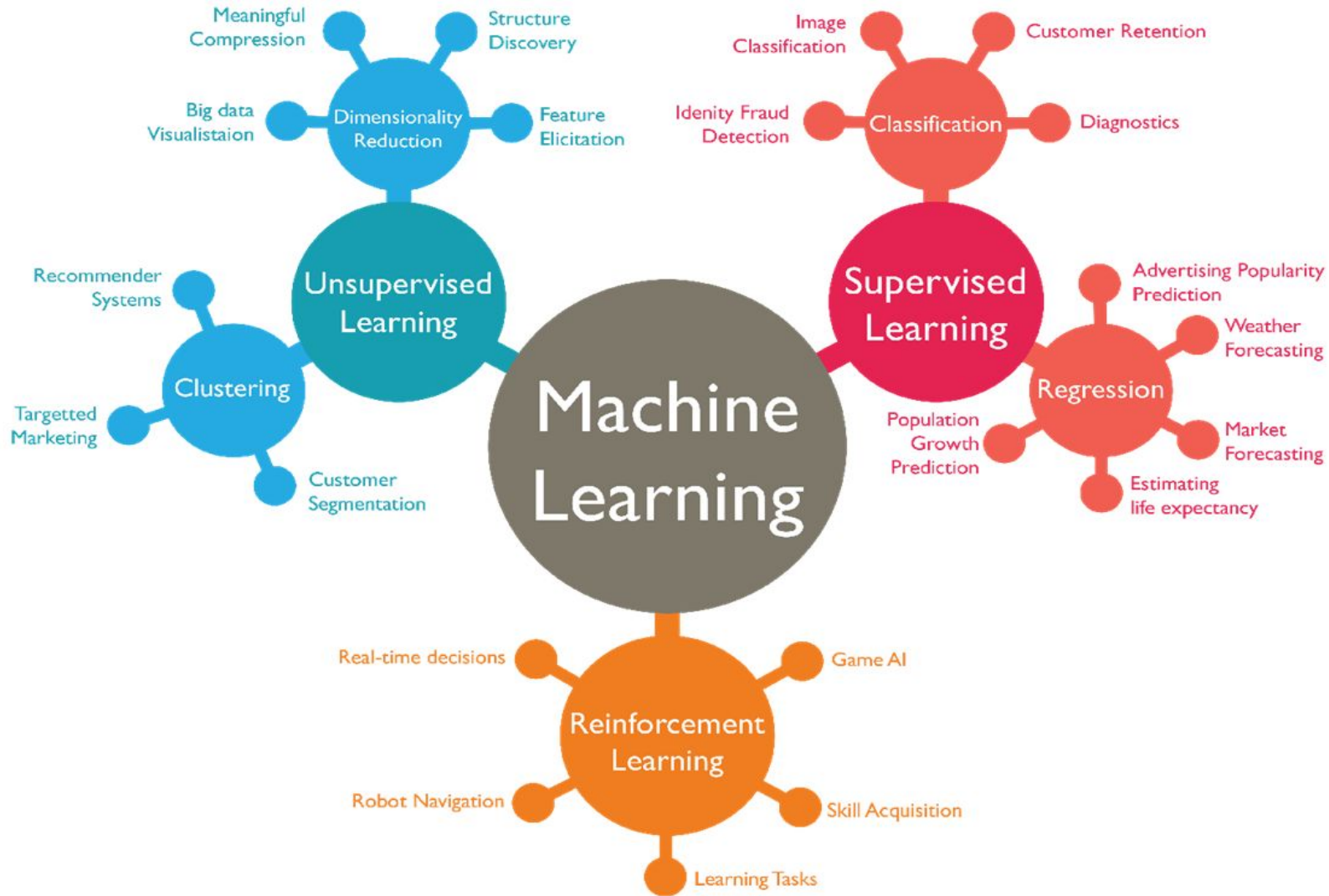
Optimum Model

- Relationships
- Patterns
- Dependencies
- Hidden structures

Questions?

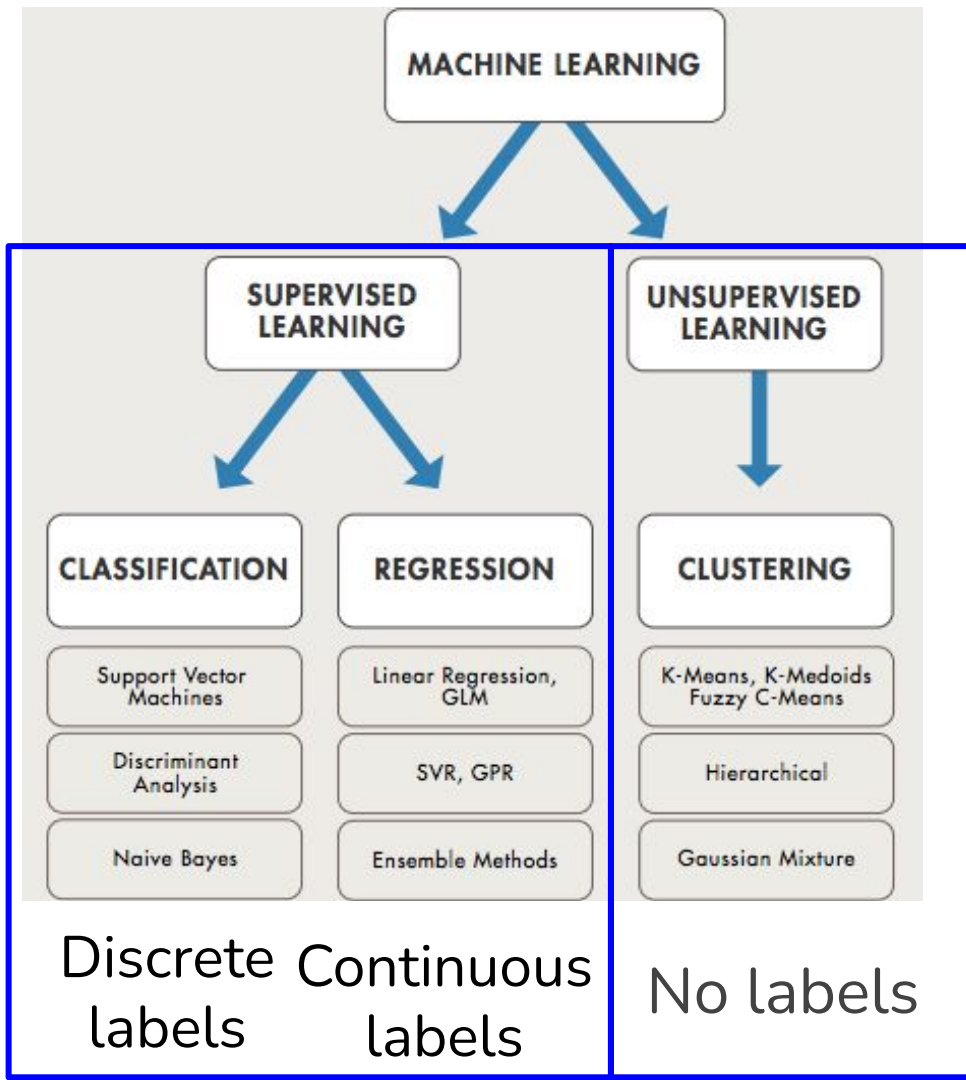
Algorithms + Techniques

Be able to
find
answers
from new
data



Are input data labelled?

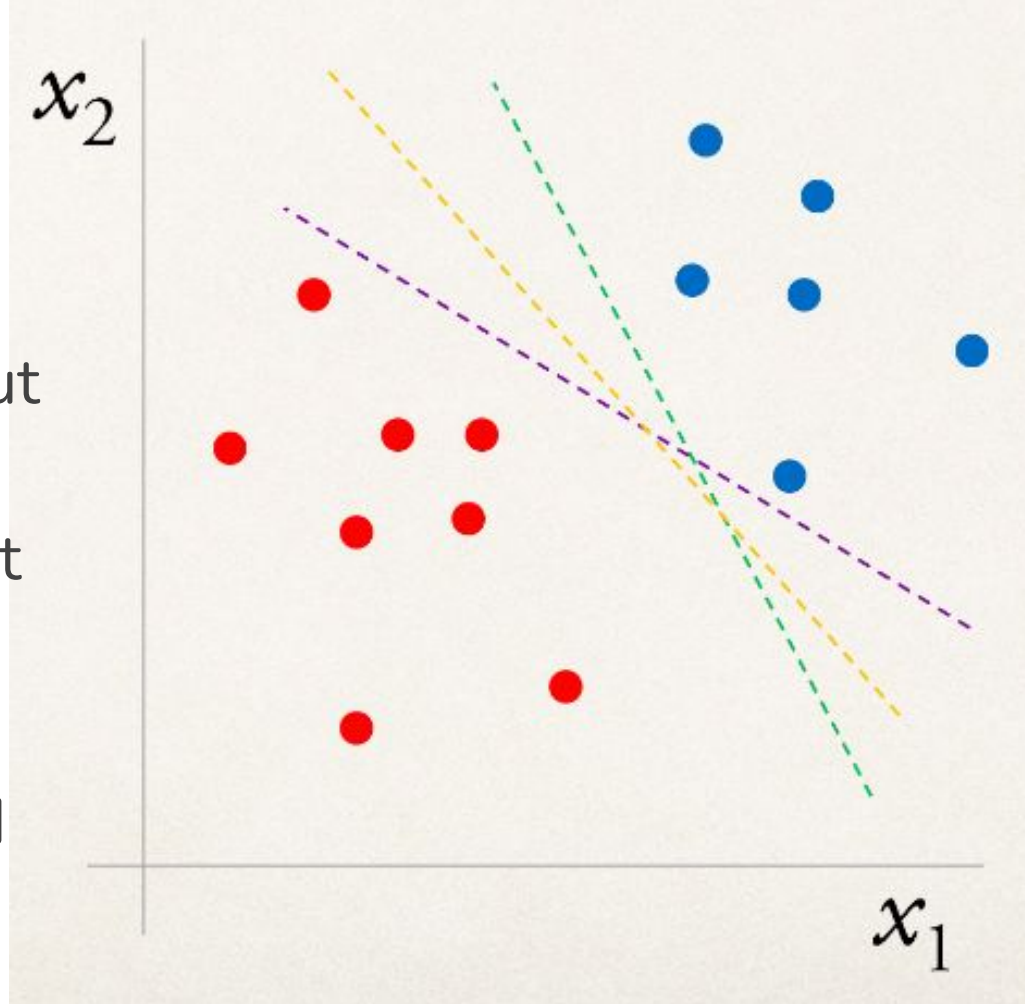
sensor1	sensor 2	sensor3	label
0.3	0.2	0.6	0
0.3	0.2	0.6	0
0.3	0.2	0.6	0
0.3	0.2	0.6	0
0.3	0.2	0.6	1
0.3	0.2	0.6	1
			1



Classification

Supervised, discrete labels

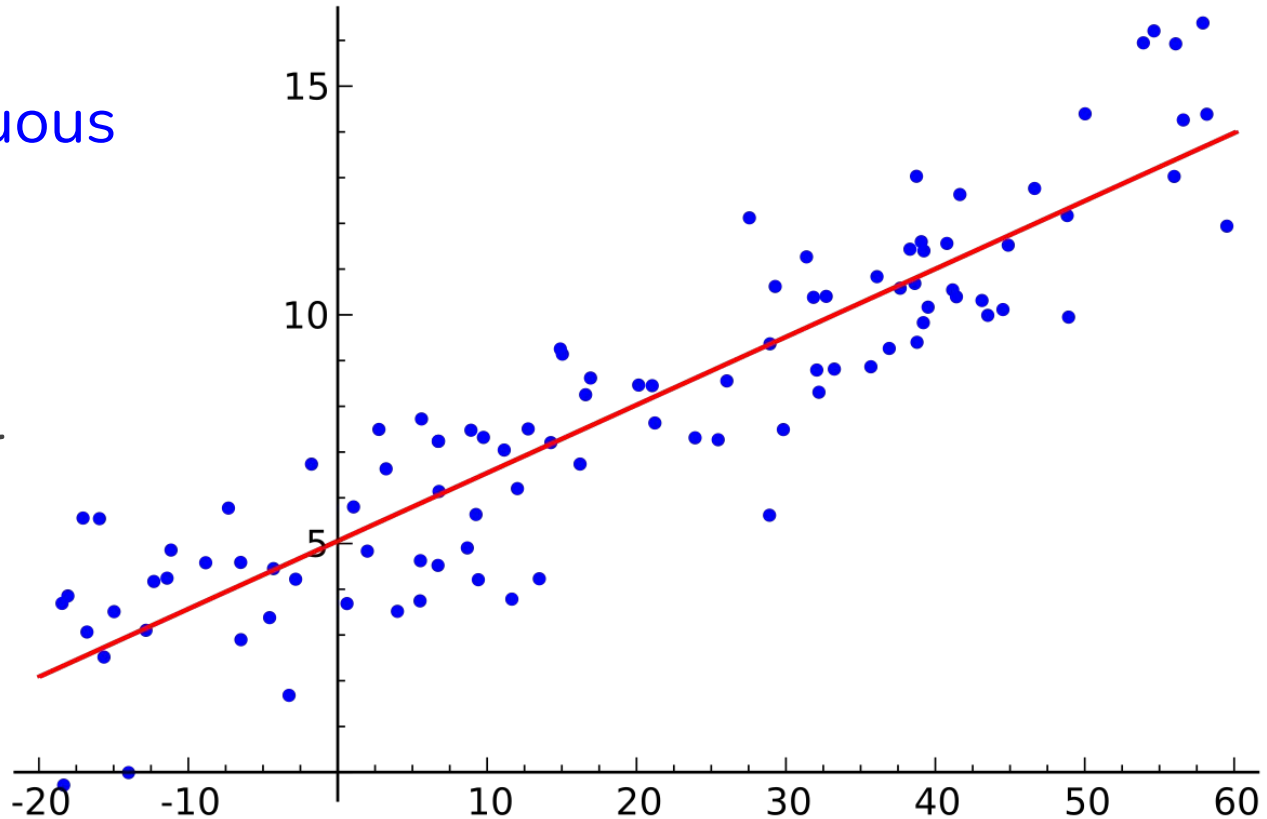
- Predict one or more output class
 - Businesses who target customers: good vs bad, stay or leave
 - **Signal vs background**



Regression

Supervised, continuous labels

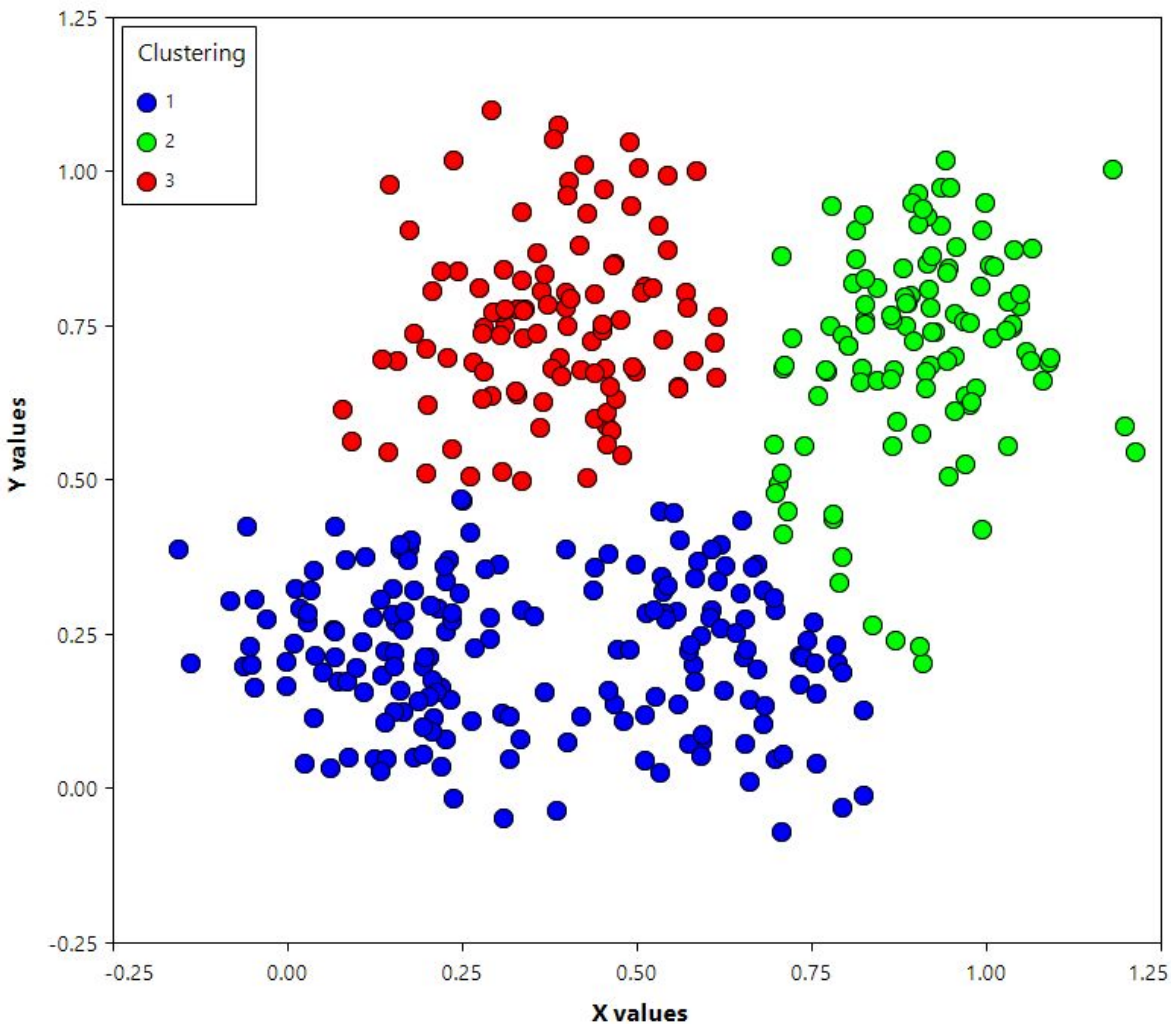
- Businesses who predict customer behavior: e.g. house prices, ...



Clustering

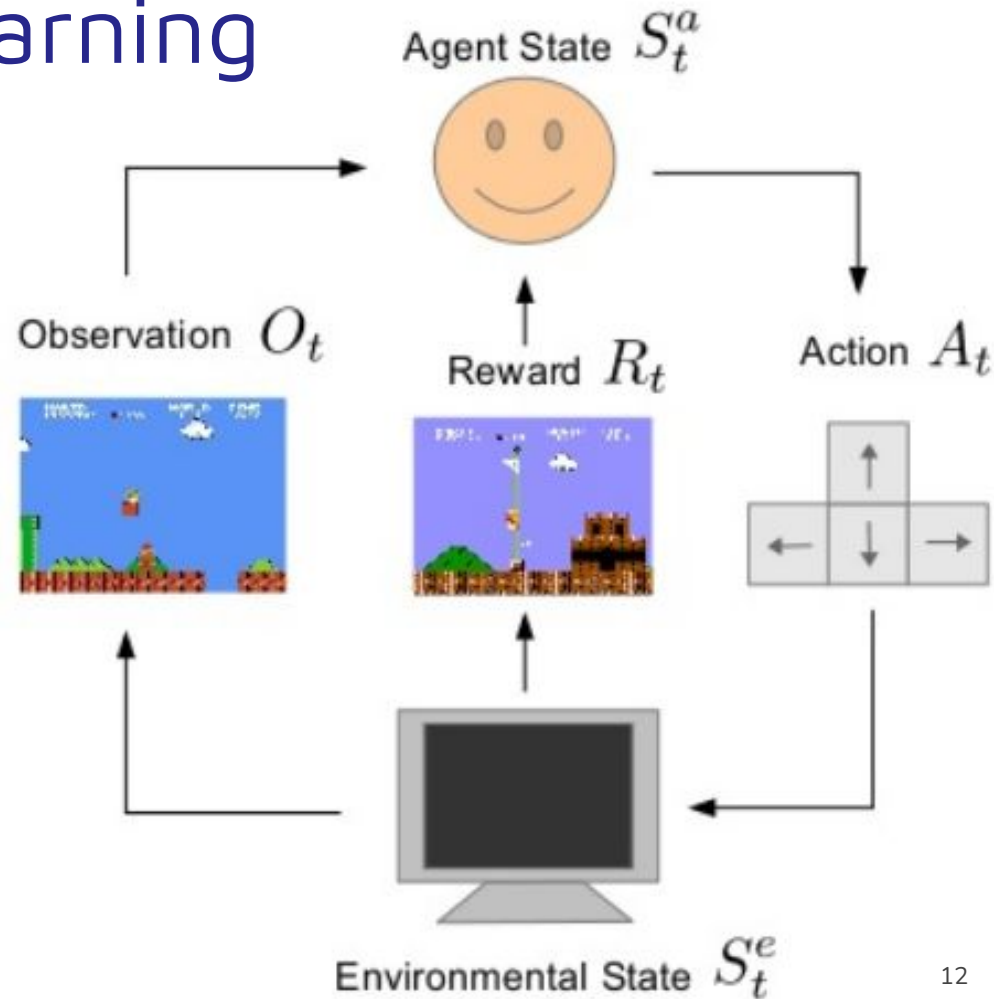
Unsupervised

- Businesses who identify customer categories
- Light vs heavy flavour jets
-



Reinforcement learning

- getting an agent to act in the world so as to maximize its rewards
- sparse and time delayed labels (**rewards**)



Remember...



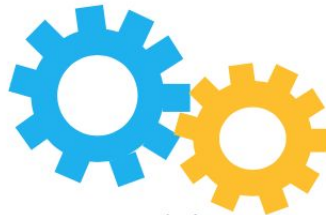
Machine learning involves
two mathematical entities



Model: a mathematical model
describes the relationship between
different aspects of the data



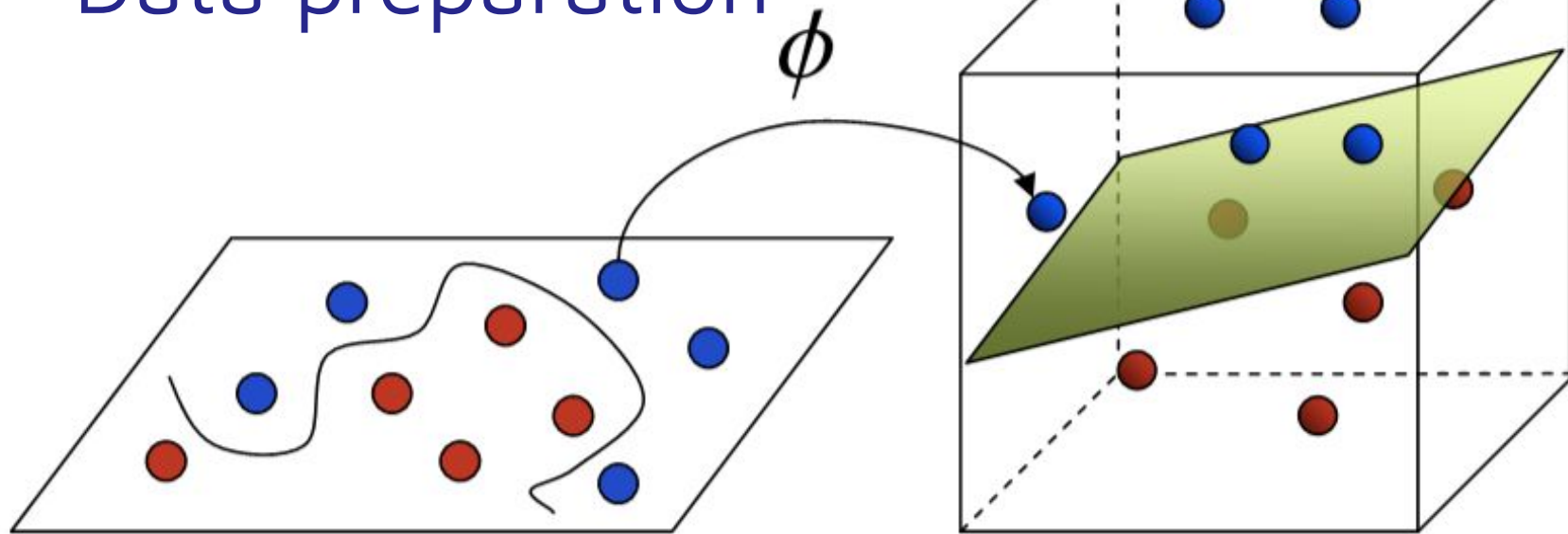
Features: a feature is a
representation of raw data



Model



Data preparation



Input Space

Feature Space

Raw data \longrightarrow

Preprocessing

Feature engineering

Raw Data

```
0 : {  
  house_info : {  
    num_rooms: 6  
    num_bedrooms: 3  
    street_name: "Shorebird Way"  
    num_basement_rooms: -1  
    ...  
  }  
}
```

Raw data doesn't come to us as feature vectors.

Feature Engineering

Feature Vector

```
[  
  6.0,  
  1.0,  
  0.0,  
  0.0,  
  0.0,  
  9.321,  
  -2.20,  
  1.01,  
  0.0,  
  ...,  
]
```

Process of creating features from raw data is **feature engineering**.

Example: supervised classification

Ingredients

- **Inputs:** X , e.g. timestamp, price, color, size, etc.
- **Features:** X , transformed inputs
- **Labels:** y

Recipe

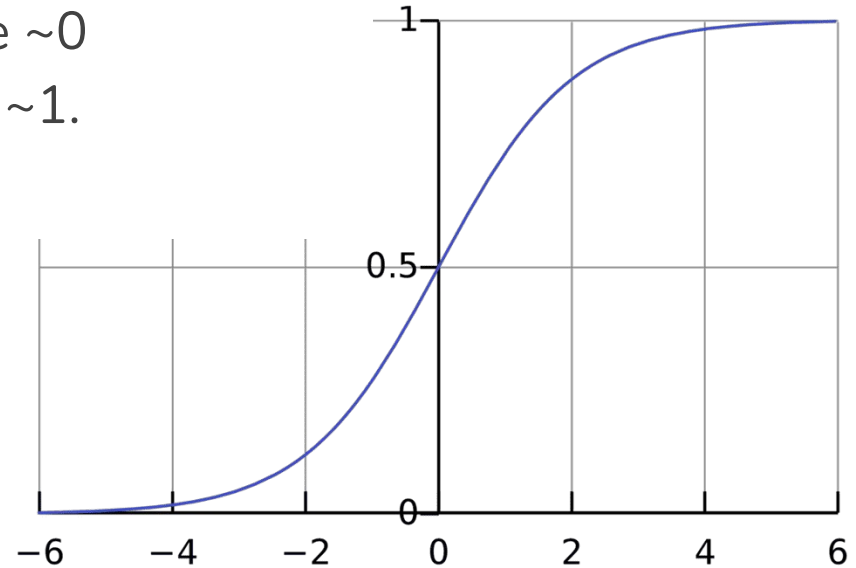


- **Weights:** W (matrix) contains the model parameters
 - **Predictions:** $z = \phi(W^T X)$ yields $(0,1)$
 - **Activation function:** ϕ (step function, e.g. sigmoid)
 - **Cost function == loss function == prediction error**, function of the model parameters W
-
- **Aim:** find weights W that minimize cost function

Activation function

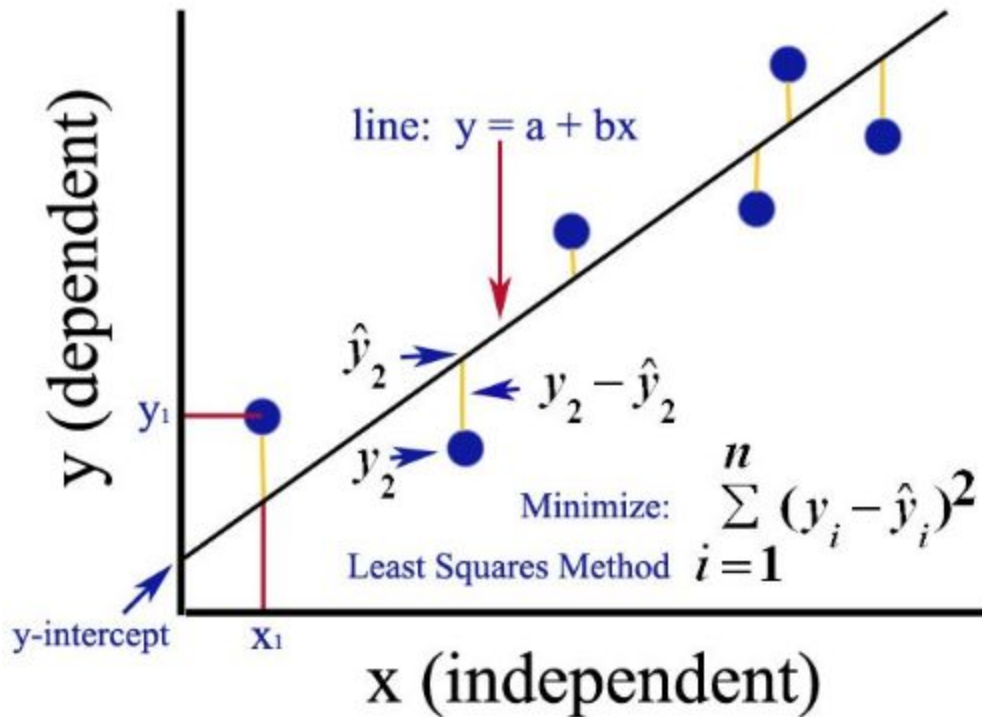
- Turns unbounded output into a known range/shape
- For example, **sigmoid** function only outputs numbers in the range (0, 1)
 - big negative numbers become ~0
 - big positive numbers become ~1.

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$



Another example, linear regression

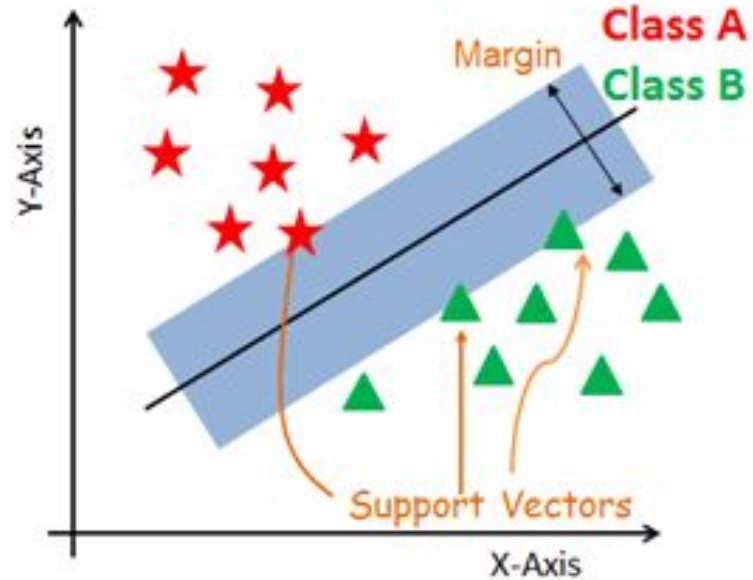
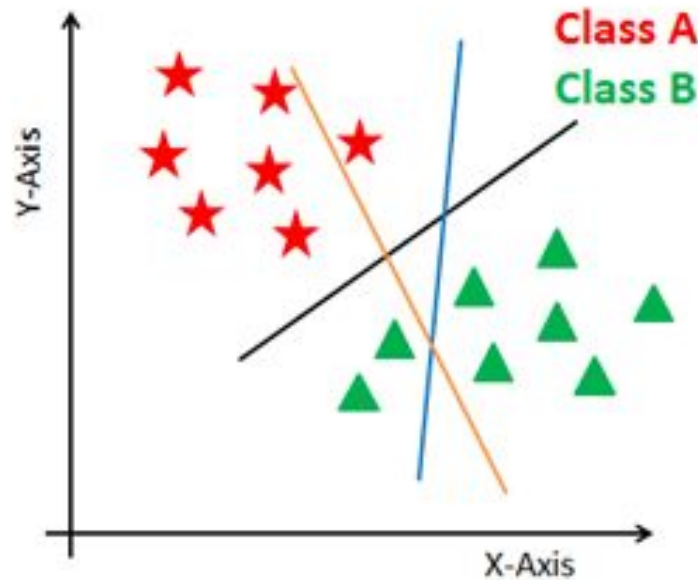
- Inputs (features): x_i
- Labels: y_i
- Model: $y = a + bx$
- Weight+bias (parameters to be found): a, b
- Cost function: **Mean Square Error (MSE)**
- No **activation function**: problem is linear



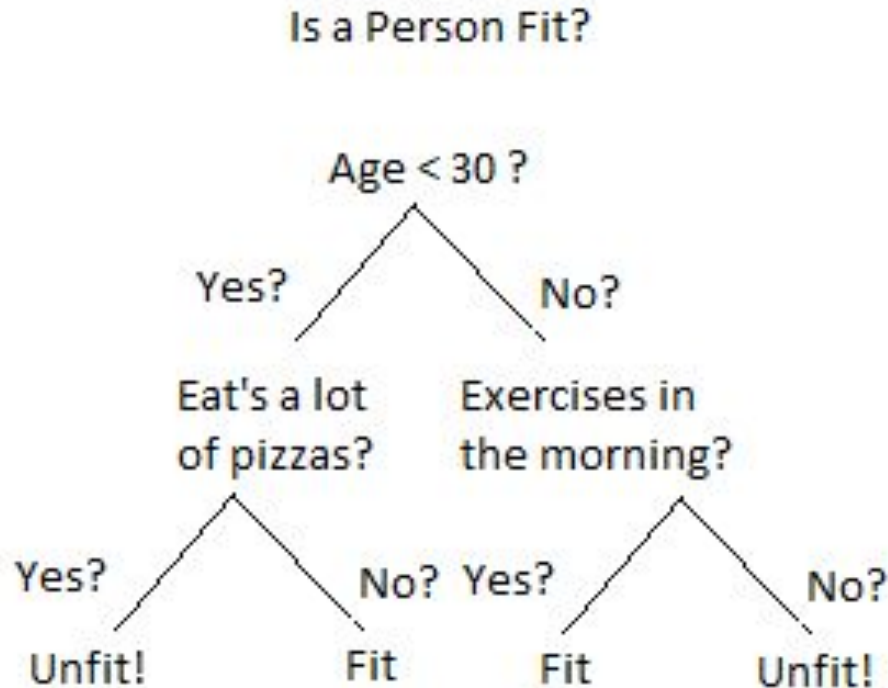
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

Supervised learning

Support vector machines (SVG): supervised classification

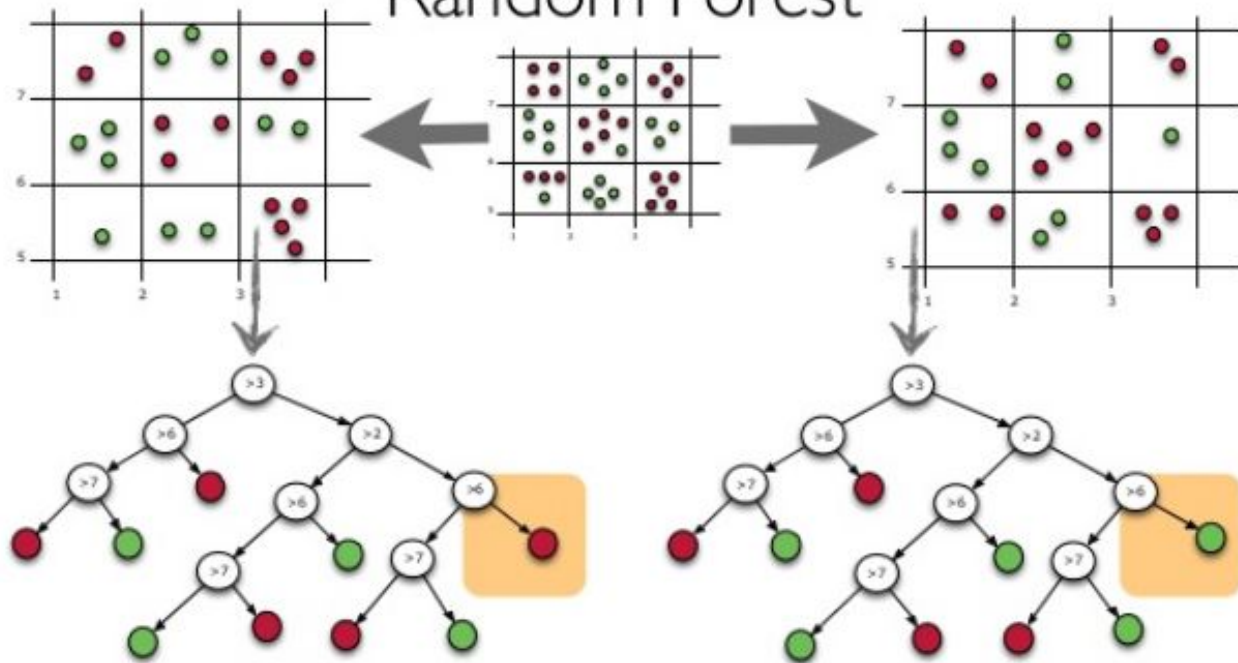


Decision trees: supervised classification



Typically used in combinations
(Random forest,
Gradient Tree
Boosting)

Random Forest



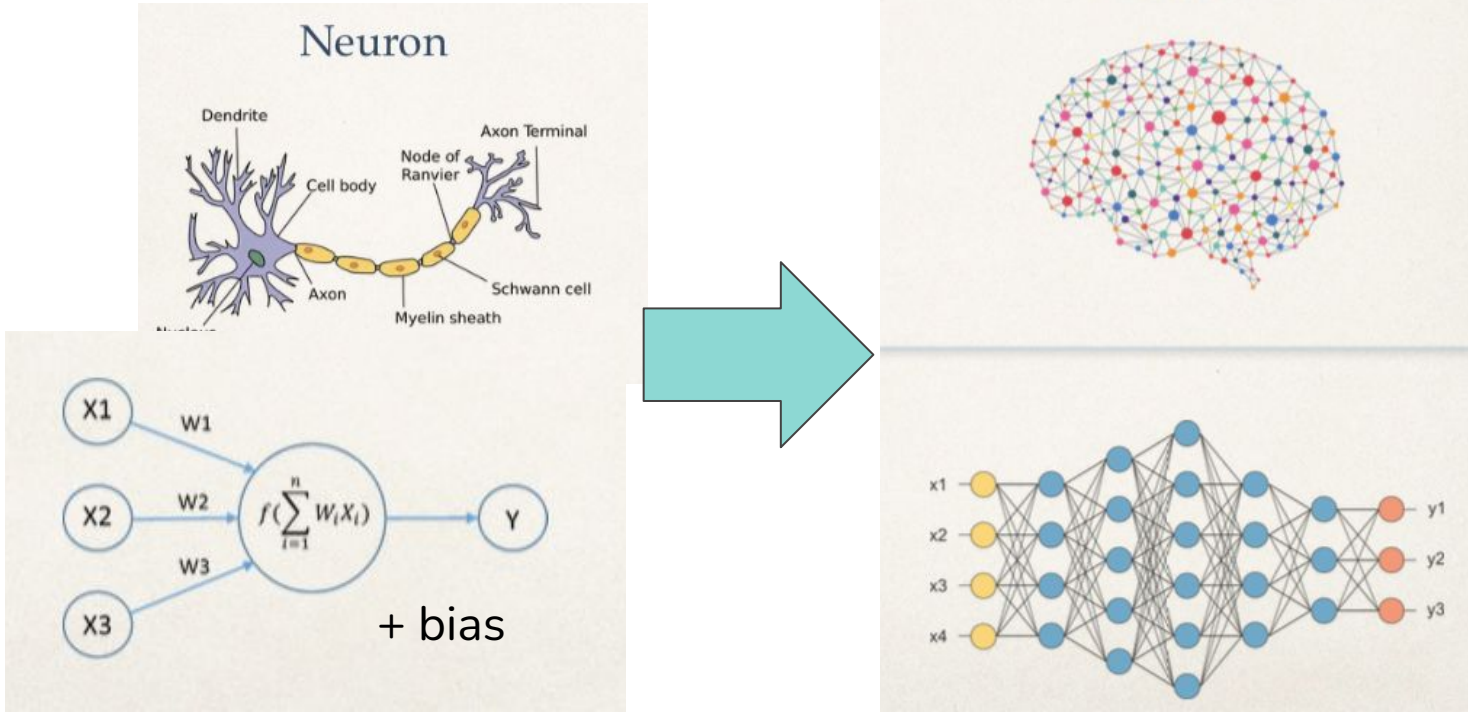
- Each tree sees part of the training sets and captures part of the information it contains

Ensembles

- **Bagging**
 - building multiple models (typically of the *same* type) from different subsamples of the training dataset
- **Boosting**
 - building multiple models (typically of the *same* type) each of which learns to fix the predictions errors of a prior model in the chain
- **Stacking**
 - building multiple models (typically of *different* types) and a supervisor model that learns how to best combine the predictions of the primary model
- **Weighting|Blending**
 - combine multiple models into single prediction using different weight functions

Neural networks: supervised classification

- Basic unit: **Neuron**. A neuron takes inputs, does some math with them, and produces **one** output



Training a neural network

Name	Weight (lb)	Height (in)	Gender
Alice	133	65	F
Bob	160	72	M
Charlie	152	70	M
Diana	120	60	F

- Predict gender from weight and height

Feature engineering

- Symmetrize numeric values
- Category -> numbers

Name	Weight (lb)	Height (in)	Gender
Alice	133	65	F
Bob	160	72	M
Charlie	152	70	M
Diana	120	60	F

Name	Weight (minus 135)	Height (minus 66)	Gender
Alice	-2	-1	1
Bob	25	6	0
Charlie	17	4	0
Diana	-15	-6	1



Ingredients

- **n**, number of samples: ?

Name	Weight (minus 135)	Height (minus 66)	Gender
Alice	-2	-1	1
Bob	25	6	0
Charlie	17	4	0
Diana	-15	-6	1

Ingredients

- **n**, number of samples: 4 (Alice, Bob, Charlie, Diana)
- **Inputs:** ?

Name	Weight (minus 135)	Height (minus 66)	Gender
Alice	-2	-1	1
Bob	25	6	0
Charlie	17	4	0
Diana	-15	-6	1

Ingredients

- **n**, number of samples: 4 (Alice, Bob, Charlie, Diana)
- **Inputs: \mathbf{X}** , dimension = 2
 - **weights** (\mathbf{X}_1) and **heights** (\mathbf{X}_2)
- **Features: ?**

Name	Weight (minus 135)	Height (minus 66)	Gender
Alice	-2	-1	1
Bob	25	6	0
Charlie	17	4	0
Diana	-15	-6	1

Ingredients

- n , number of samples: 4 (Alice, Bob, Charlie, Diana)
- **Inputs:** \mathbf{X} , dimension = 2
 - **weights** (\mathbf{X}_1) and **heights** (\mathbf{X}_2)
- **Features:** $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, transformed inputs
- $\mathbf{y}_{\text{true}} : ?$

Name	Weight (minus 135)	Height (minus 66)	Gender
Alice	-2	-1	1
Bob	25	6	0
Charlie	17	4	0
Diana	-15	-6	1

Ingredients

- n , number of samples: 4 (Alice, Bob, Charlie, Diana)
- **Inputs:** \mathbf{X} , dimension = 2
 - **weights** (\mathbf{X}_1) and **heights** (\mathbf{X}_2)
- **Features:** $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, transformed inputs
- \mathbf{y}_{true} : true value of y (Gender)

Name	Weight (minus 135)	Height (minus 66)	Gender
Alice	-2	-1	1
Bob	25	6	0
Charlie	17	4	0
Diana	-15	-6	1

Model: 1 input layer

- Inputs: $\mathbf{x} = (x_1, x_2)$

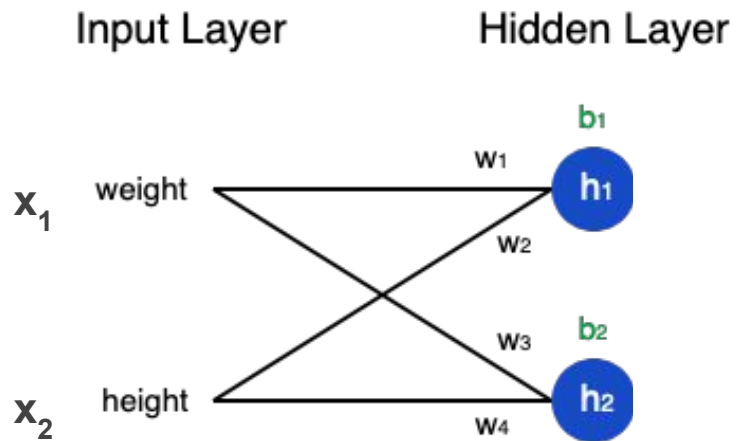
Input Layer

x_1 weight

x_2 height

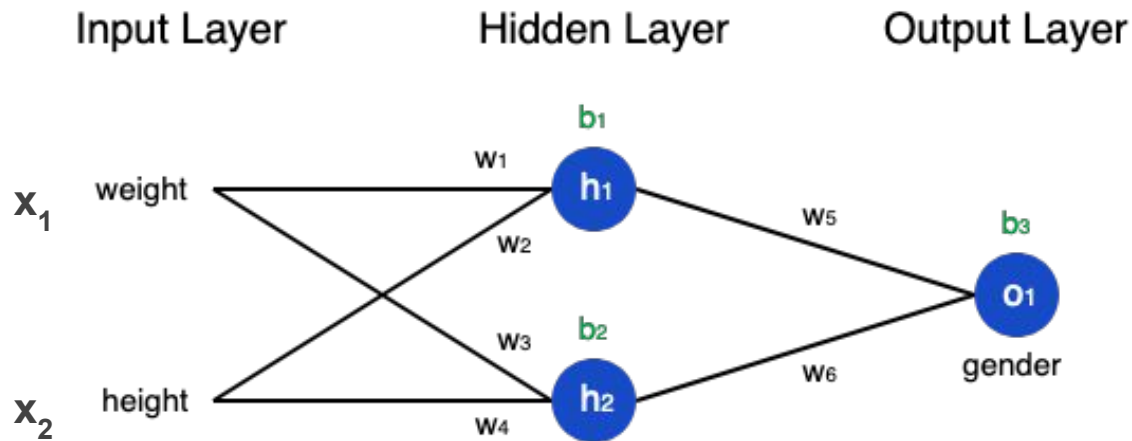
Model: 1 hidden layer with two neurons

- Inputs: $\mathbf{x} = (x_1, x_2)$
- Outputs of the hidden layer: \mathbf{h}



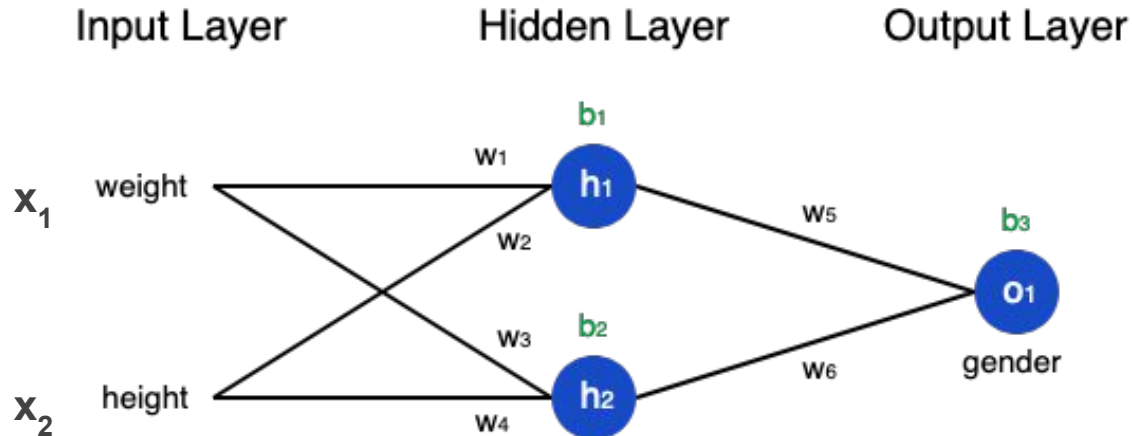
Model: 1 hidden layer with two neurons

- Inputs: $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$
- Outputs of the hidden layer: \mathbf{h}
- \mathbf{y}_{pred} : predicted value of $y = \mathbf{o}$
- Unknown parameters: weights \mathbf{w} and biases \mathbf{b}



Some math

- For each neuron: $y_j = b_j + f \sum_i x_i w_{ij}$, f activation function
- For the net:
 - $h_1 = f(w_1 x_1 + w_2 x_2) + b_1$
 - $h_2 = f(w_3 x_1 + w_4 x_2) + b_2$
 - $o_1 = f(w_5 h_1 + w_6 h_2) + b_3$



Model training

- Training the network == Find weights **w** and biases **b** that minimize the loss

$$L(w_1, w_2, w_3, w_4, w_5, w_6, b_1, b_2, b_3)$$

- Loss function **L: MSE**

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_{\text{true}} - y_{\text{pred}})^2$$

Back propagation

- Minimization taking partial derivatives

If only Alice
in the
dataset, $n=1$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y_{pred}} * \frac{\partial y_{pred}}{\partial h_1} * \frac{\partial h_1}{\partial w_1}$$

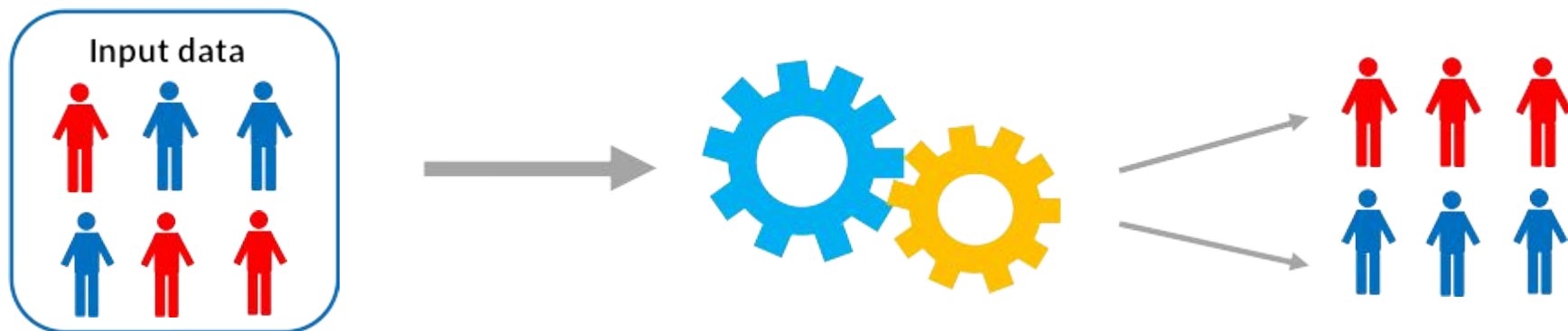
$$\begin{array}{l} L = (1 - y_{pred})^2 \\ \frac{\partial L}{\partial y_{pred}} = \frac{\partial (1 - y_{pred})^2}{\partial y_{pred}} \end{array} \quad \left| \quad \begin{array}{l} y_{pred} = o_1 = f(w_5 h_1 + w_6 h_2 + b_3) \\ \frac{\partial y_{pred}}{\partial h_1} = w_5 * f'(w_5 h_1 + w_6 h_2 + b_3) \end{array} \right. \quad \begin{array}{l} h_1 = f(w_1 x_1 + w_2 x_2 + b_1) \\ \frac{\partial h_1}{\partial w_1} = x_1 * f'(w_1 x_1 + w_2 x_2 + b_1) \end{array}$$



Unsupervised learning

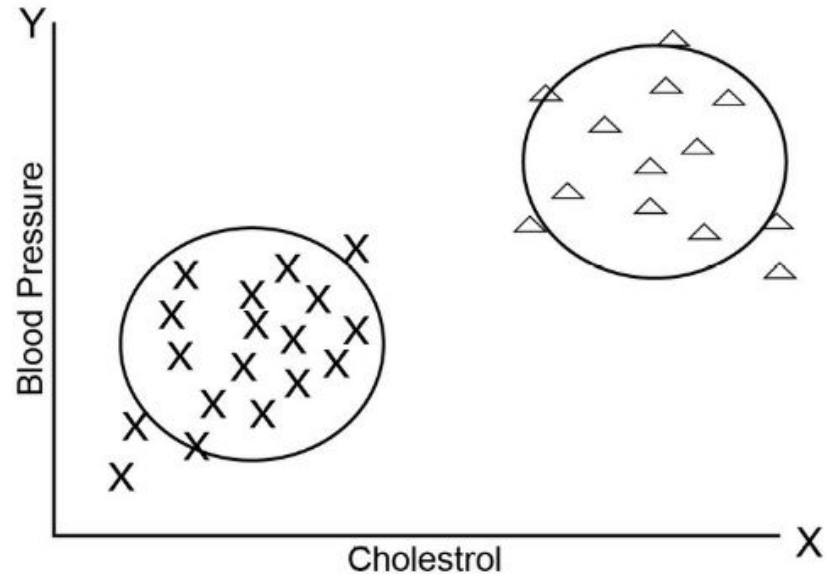
Challenges

- No label (ground truth) in input dataset
- The system must have the ability to recognize patterns in the data without explicitly being told what patterns to identify



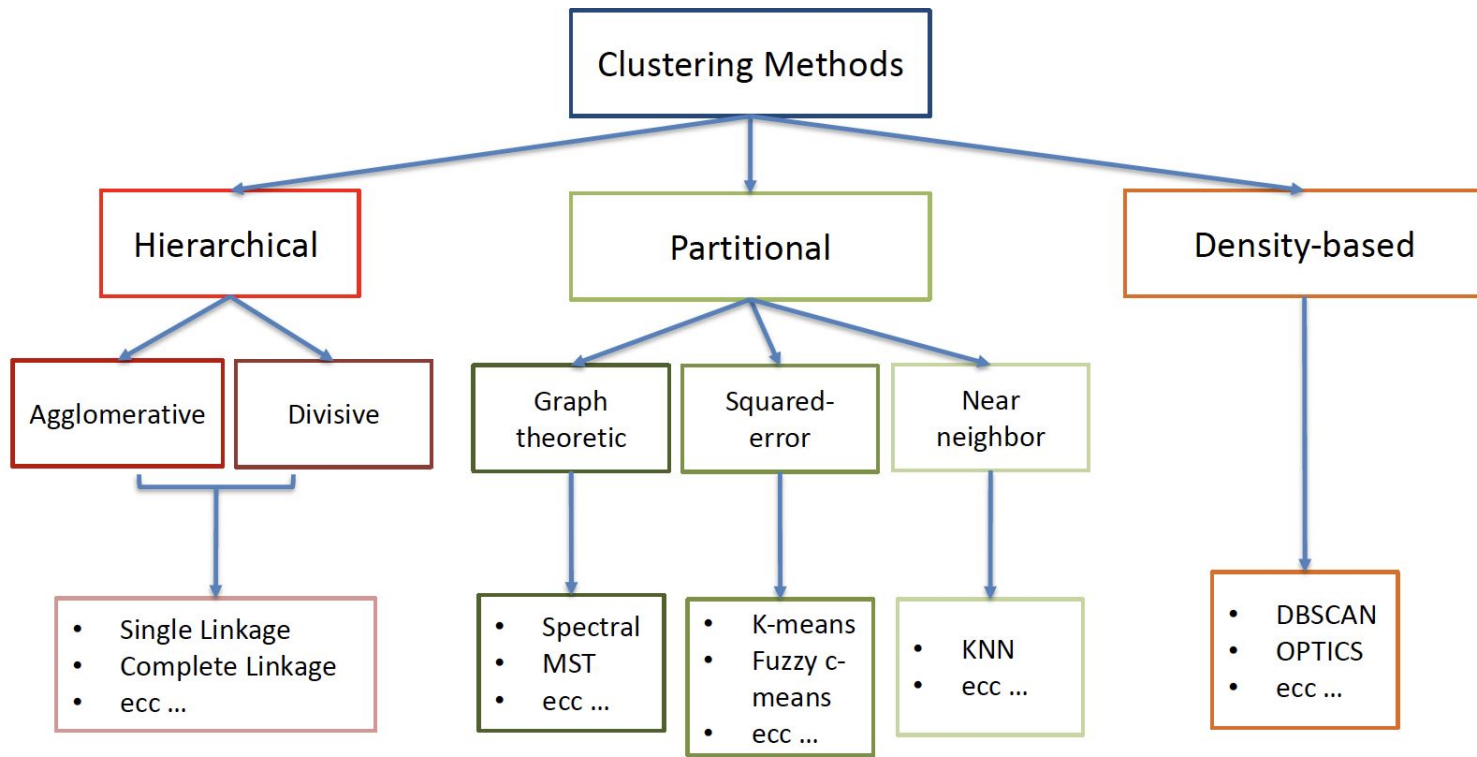
Clustering

- The most basic type of unsupervised learning is clustering
- Definition: Clustering is a set of *methods or algorithms that are used to find natural groupings* according to predefined properties of variables in a dataset
- Clustering is mostly used when we don't have labeled data – data with predefined classes
- Clustering uses various properties inside the dataset



Clustering applications

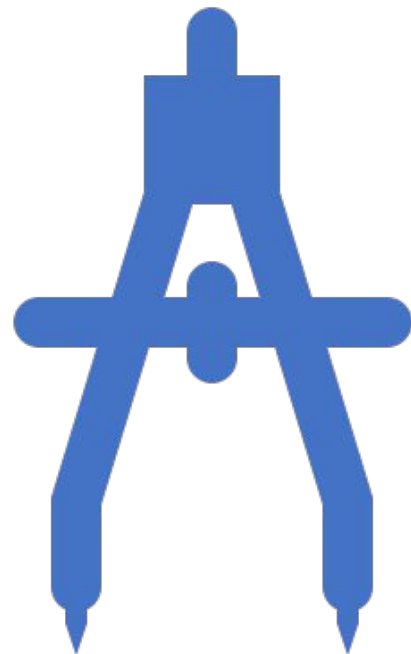
- **Exploratory data analysis:** When we have unlabeled data, we often do clustering to explore the underlying structure and categories of the dataset
- **Generate training data:** Sometimes, after processing unlabeled data with clustering methods, it can be labeled for further training with supervised learning algorithms
- **Natural language processing:** Clustering can be used for the grouping of similar words, texts, articles, or tweets, without labeled data
- **Anomaly detection:** You can use clustering to find outliers



- **explicit methods:** the number of clusters is imposed by the researcher (agglomerative hierarchical clustering and k-means)
- **implicit methods:** the number of modules is adapted on the dataset analyzed according to other information suggested by the researcher (affinity propagation)

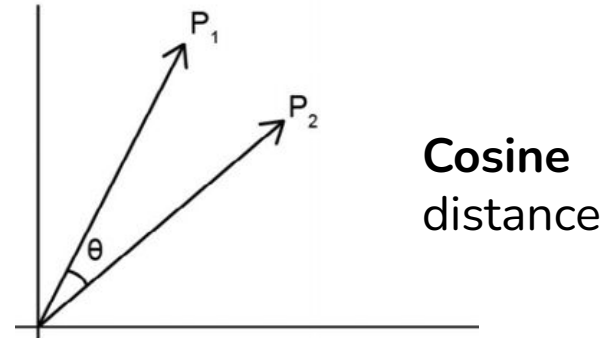
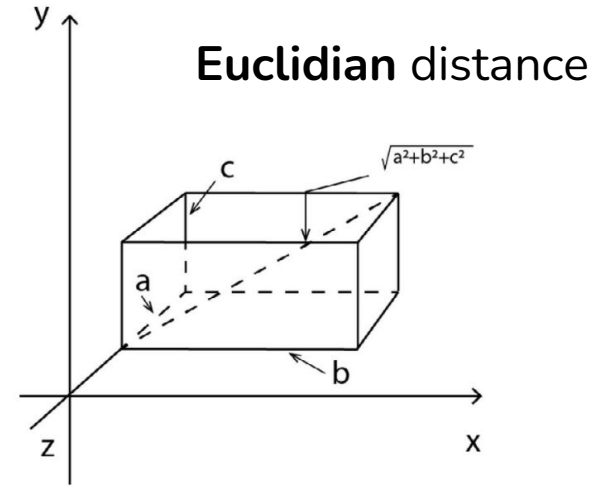
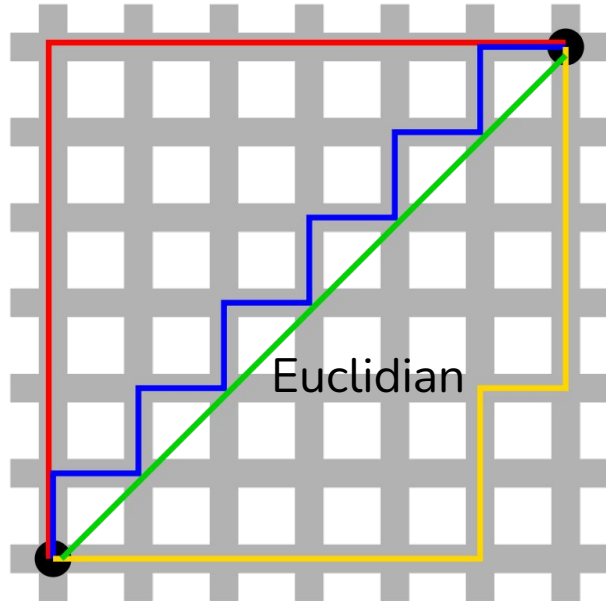
K-means clustering

- This algorithm finds natural groupings in accordance with a **predefined similarity or distance** measure
 - the distance or metric is a measurement of closeness between data points
 - Examples:
 - Euclidean distance;
 - Manhattan distance;
 - Cosine distance.



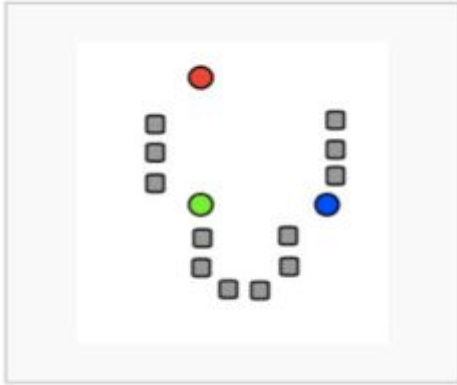
K-means: Distances

Taxicab or **Manhattan** distance:
sum of the projections along all
axis

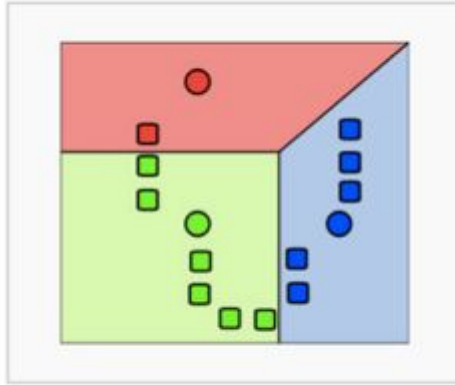


K-means: how it works

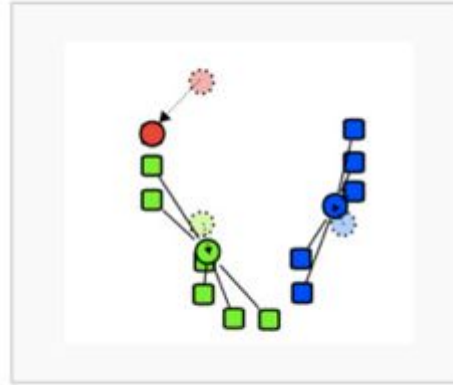
You must define **k**, the **number of clusters**, and which **distance** to use



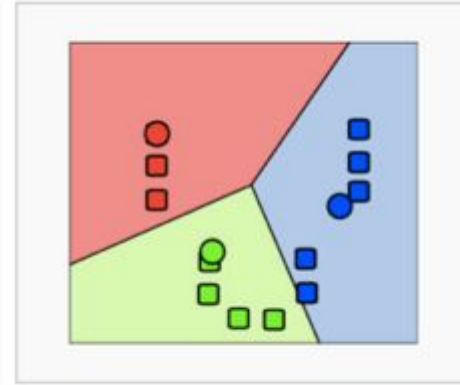
1. k initial "means" (in this case $k=3$) are randomly generated within the data domain (shown in color).



2. k clusters are created by associating every observation with the nearest mean. The partitions here represent the [Voronoi diagram](#) generated by the means.



3. The [centroid](#) of each of the k clusters becomes the new mean.

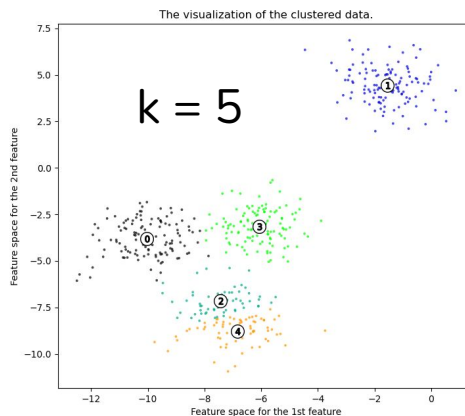
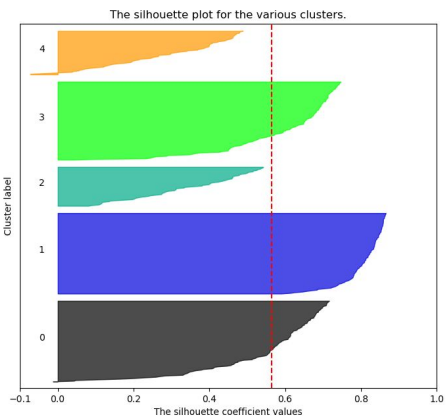
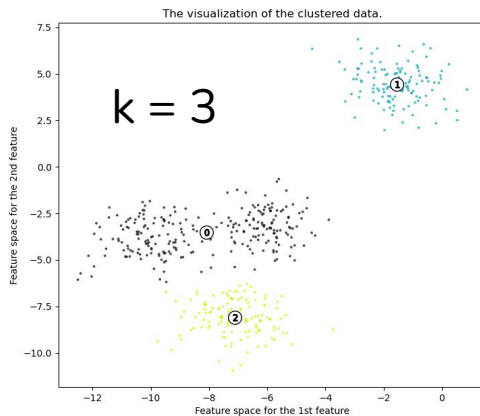
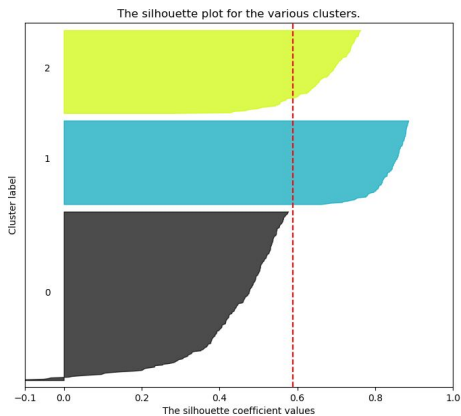
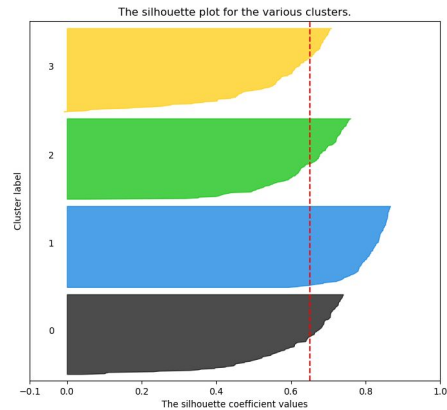
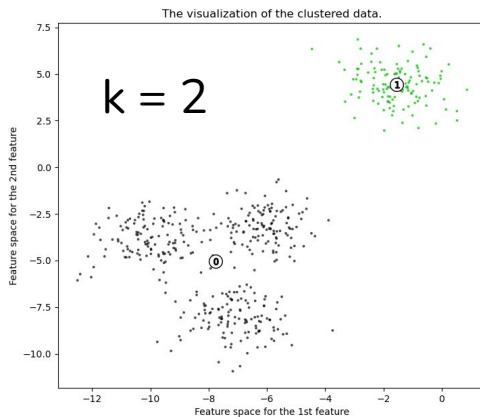
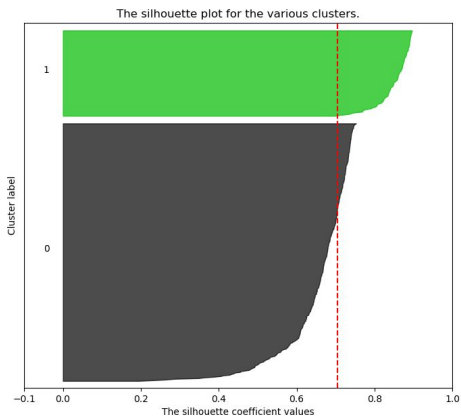


4. Steps 2 and 3 are repeated until convergence has been reached.

How to choose the number of clusters?

- The Silhouette method
- The Elbow method or WSS
- The Gap statistics

Silhouette score

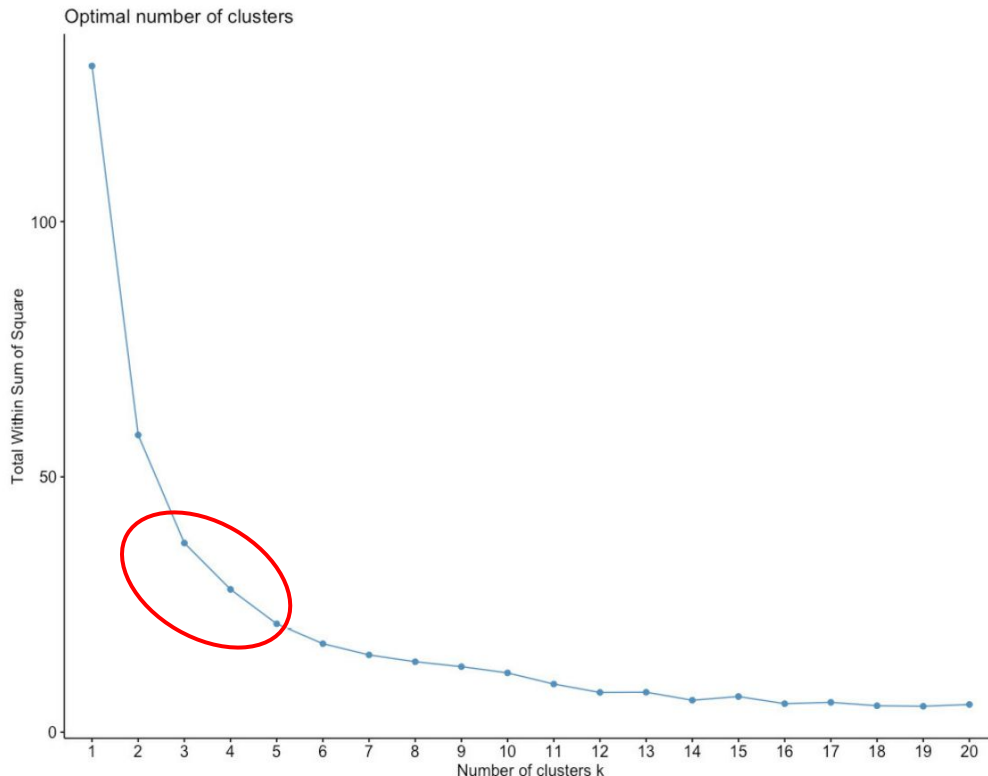


The Elbow method or WSS

Within-Cluster-Sum of Squared Errors:

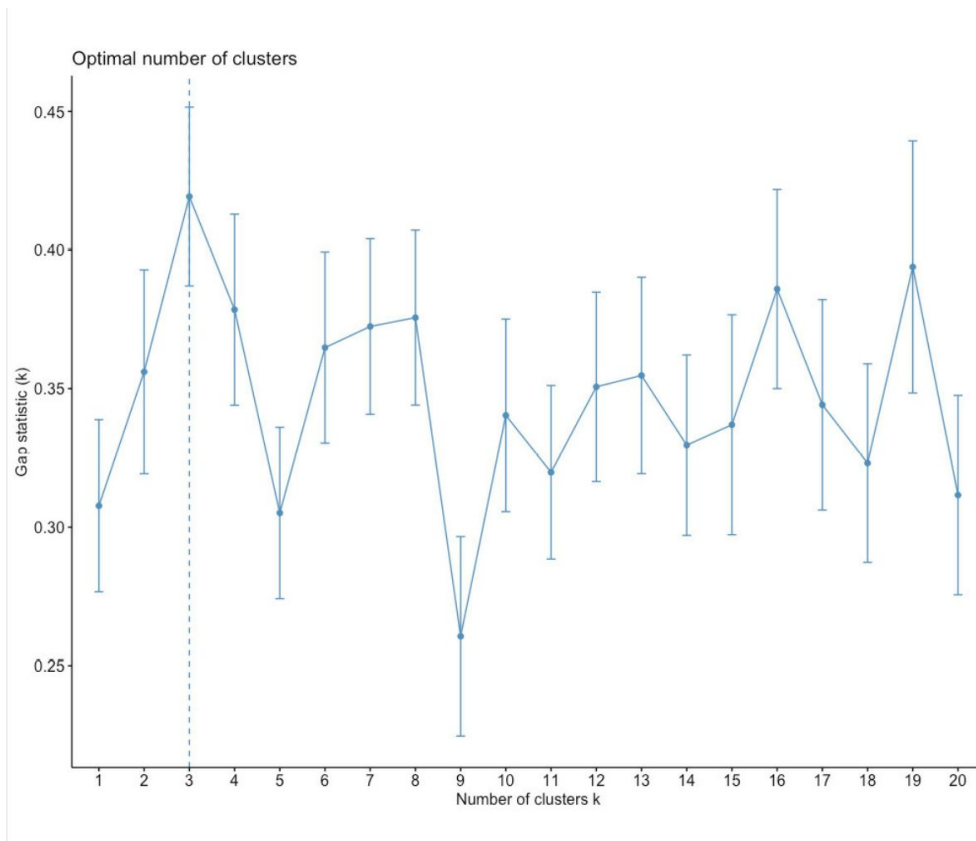
- The square of the distance of each point from the centre of the cluster (Squared Errors)
- The WSS score is the sum of these Squared Errors for all the points

$$WSS = \sum_{i=1} \sum_{x \in C_i} d(x, \bar{x}_{C_i})^2$$



Gap statistics

The Gap statistic is calculated by comparing the WSS value for the clusters generated on our dataset versus a reference dataset in which there are no apparent clusters, i.e. a random distribution of data points between the minimum and maximum values of our dataset



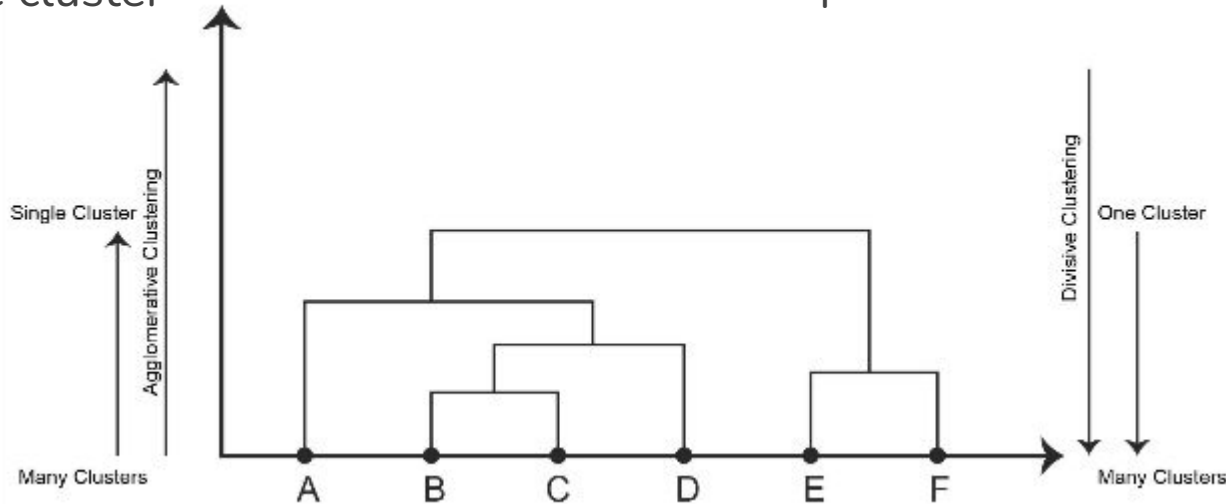
Hierarchical clustering

- **Agglomerative** clustering

- First, each data point is assumed to be a single cluster
- Then the most similar clusters are merged until all data points are in a single cluster

bottom-up
approach

Easier to
implement



- **Divisive** clustering

- all data points are initially assumed to be in a single cluster
- Then the cluster is split into multiple clusters until each data point is a cluster on its own

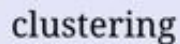
Top down
approach

More
efficient and
precise







classification



Spectral Clustering
GMM



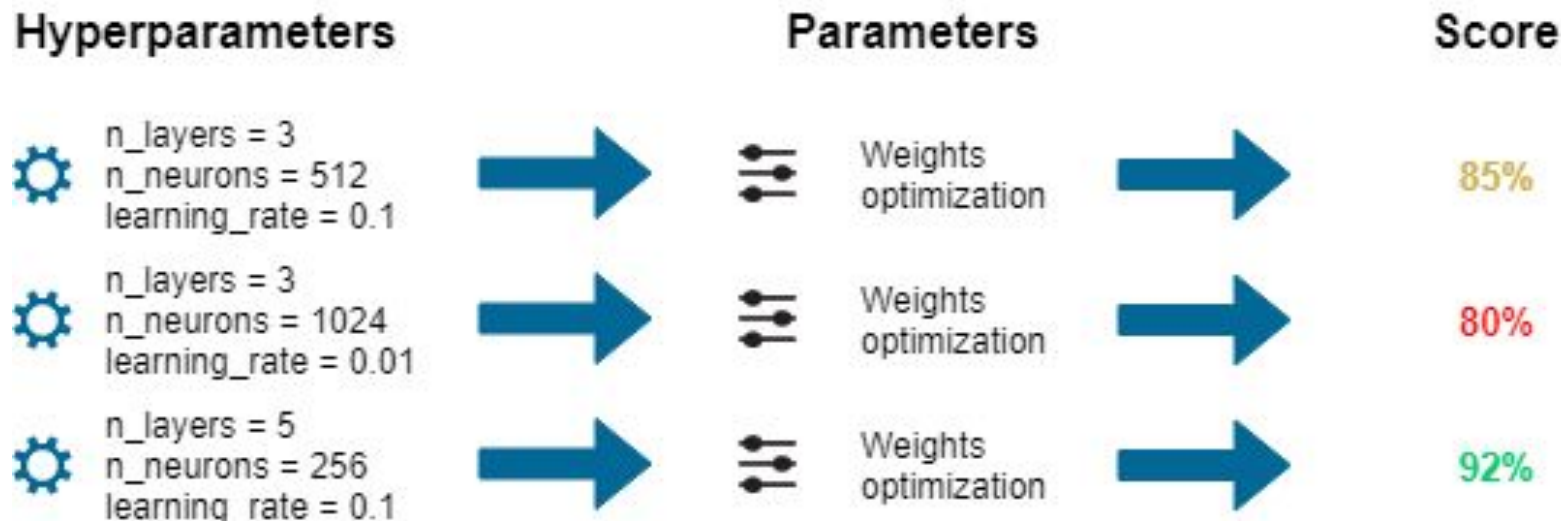
dimensionality
reduction

	TYPE	NAME	DESCRIPTION	ADVANTAGES	DISADVANTAGES
Linear		Linear regression	The “best fit” line through all data points. Predictions are numerical.	Easy to understand – you clearly see what the biggest drivers of the model are.	<ul style="list-style-type: none"> ✗ Sometimes too simple to capture complex relationships between variables. ✗ Tendency for the model to “overfit”.
		Logistic regression	The adaptation of linear regression to problems of classification (e.g., yes/no questions, groups, etc.)	Also easy to understand.	<ul style="list-style-type: none"> ✗ Sometimes too simple to capture complex relationships between variables. ✗ Tendency for the model to “overfit”.
Tree-based		Decision tree	A graph that uses a branching method to match all possible outcomes of a decision.	Easy to understand and implement.	<ul style="list-style-type: none"> ✗ Not often used on its own for prediction because it's also often too simple and not powerful enough for complex data.
		Random Forest	Takes the average of many decision trees, each of which is made with a sample of the data. Each tree is weaker than a full decision tree, but by combining them we get better overall performance .	A sort of “wisdom of the crowd”. Tends to result in very high quality models. Fast to train.	<ul style="list-style-type: none"> ✗ Can be slow to output predictions relative to other algorithms. ✗ Not easy to understand predictions.
		Gradient Boosting	Uses even weaker decision trees, that are increasingly focused on “hard” examples .	High-performing.	<ul style="list-style-type: none"> ✗ A small change in the feature set or training set can create radical changes in the model. ✗ Not easy to understand predictions.
Neural networks		Neural networks	Mimics the behavior of the brain. Neural networks are interconnected neurons that pass messages to each other. Deep learning uses several layers of neural networks put one after the other.	Can handle extremely complex tasks - no other algorithm comes close in image recognition.	<ul style="list-style-type: none"> ✗ Very, very slow to train, because they have so many layers. Require a lot of power. ✗ Almost impossible to understand predictions.

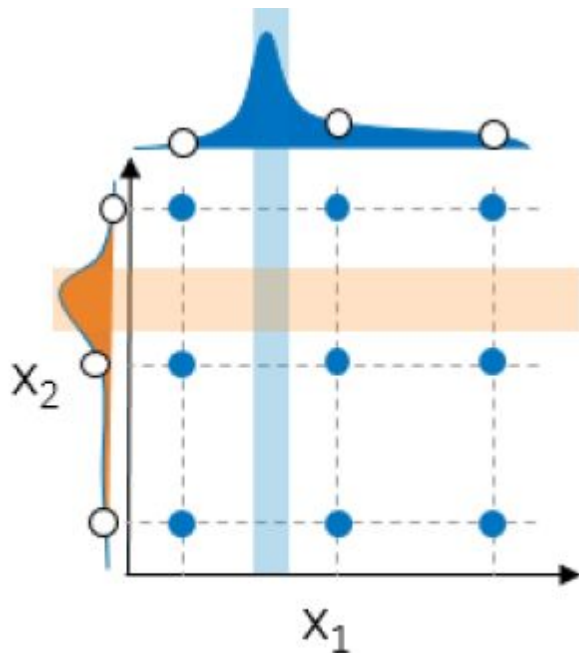
*All models are wrong, but
some are useful (George Box)*

Hyperparameters vs parameters

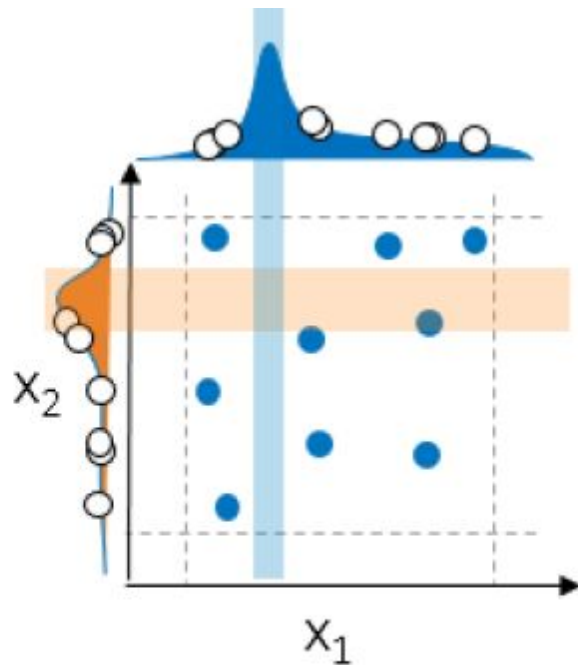
- Model **parameters** are learned during training when we optimize a loss function
- **Hyperparameters** are not model parameters and they cannot be directly trained from the data



Hyperparameter tuning



(a) Standard Grid Search



(b) Random Search

Modeling Algorithm

Tune

hyperparameters (tuning options)

- Polynomial order, penalty parameter, ...
- Network configuration, solver options, ...
- Max tree depth, splitting criterion, ...

Model

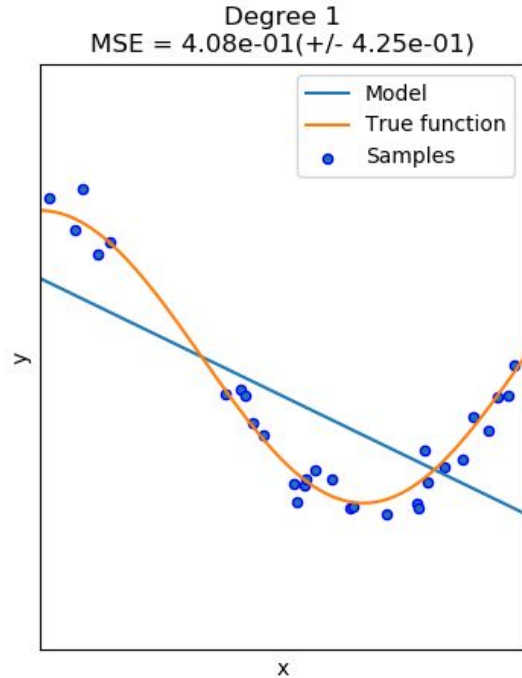
Train

model parameters

- Regression coefficients
- Neural net weights
- Tree splitting rules

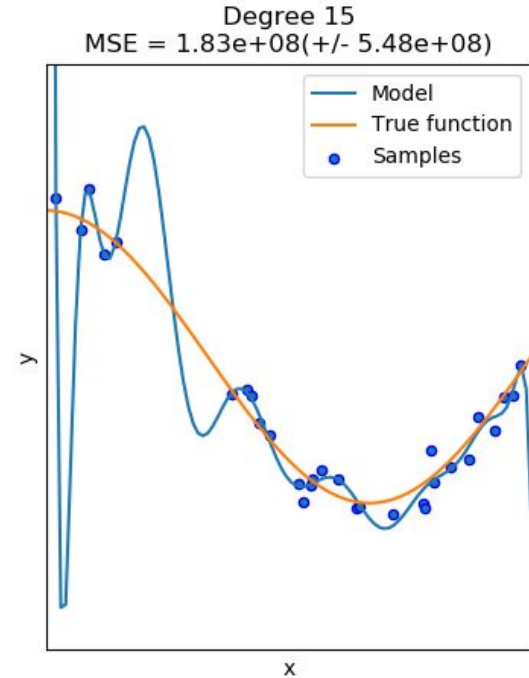
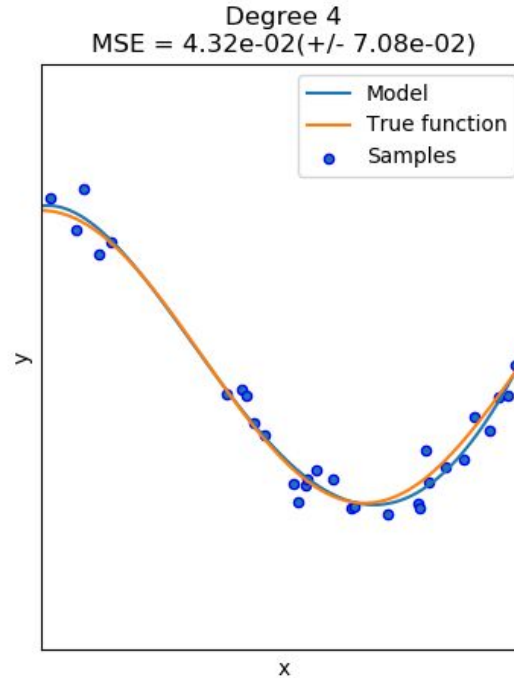
...

Overfitting / underfitting



Underfitting

Model doesn't have enough parameters to describe data



Overfitting

Model has too many parameters

Classification metrics

- **ROC**: Receiver Operating Characteristics
- **AUC**: Area under the curve
- **TPR**: True positive rate
- **FPR**: False positive rate
- **TNR/FNR**: True/False negative rate

Confusion matrix

		True class	
		p	n
Hypothesized class	Y	True Positives	False Positives
	N	False Negatives	True Negatives

Column totals:

P

N

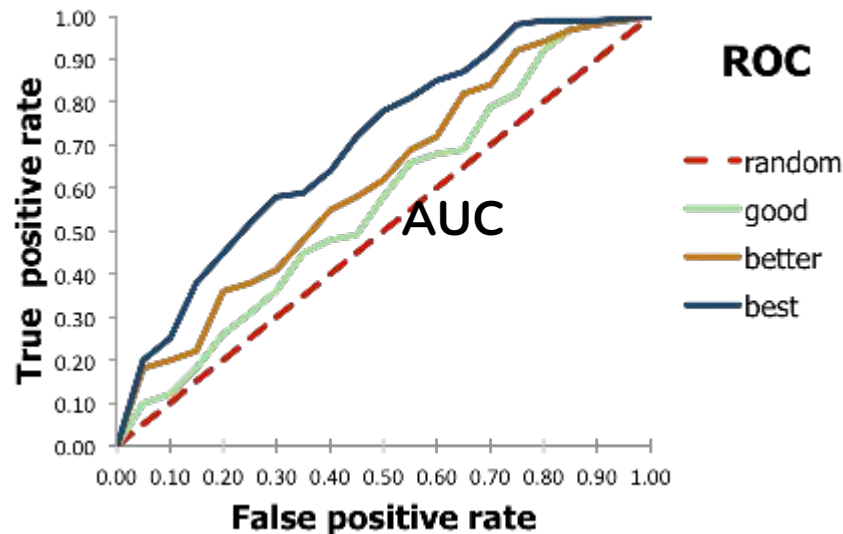
$$\text{fp rate} = \frac{FP}{N}$$

$$\text{tp rate} = \frac{TP}{P}$$

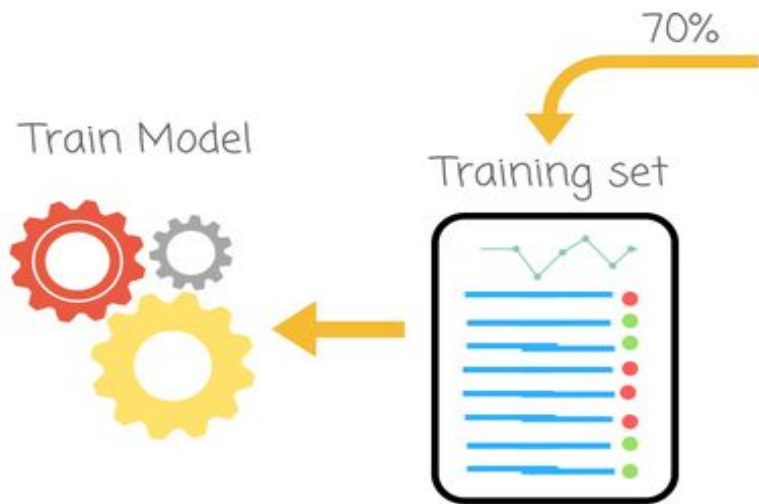
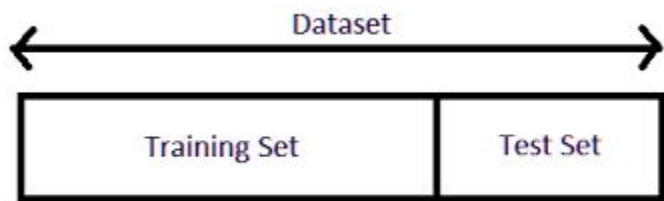
$$\text{precision} = \frac{TP}{TP+FP} \quad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N}$$

$$\text{F-measure} = \frac{2}{1/\text{precision} + 1/\text{recall}}$$



Training and test set



Entire Dataset



30%

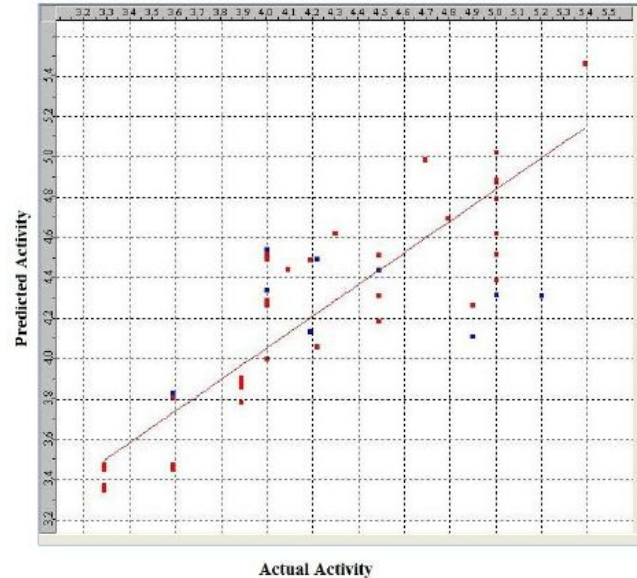


Test set

Test labels

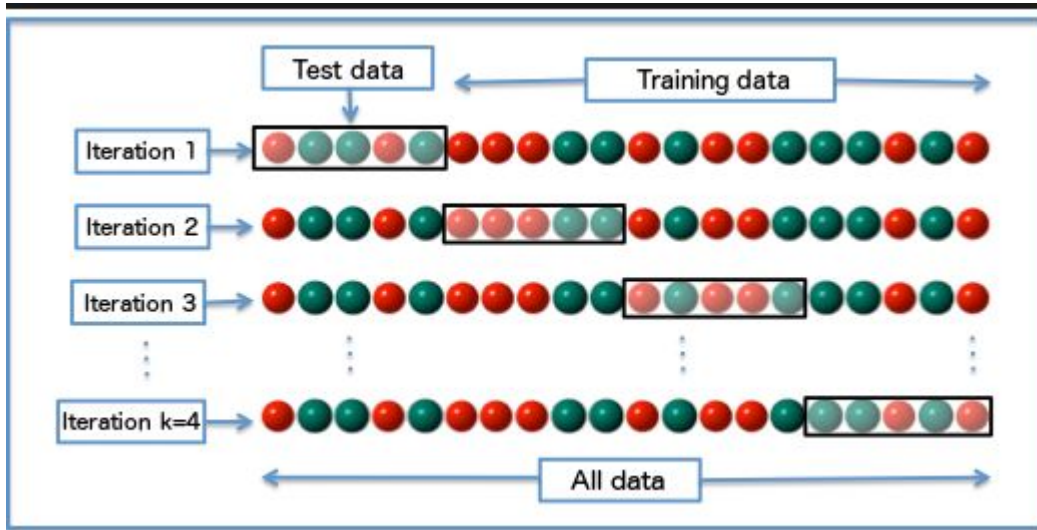
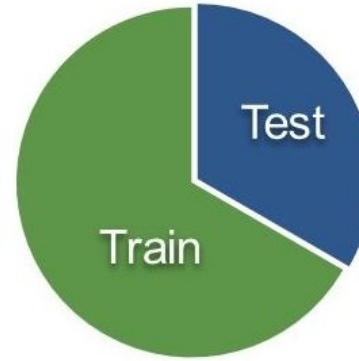


Used later for testing



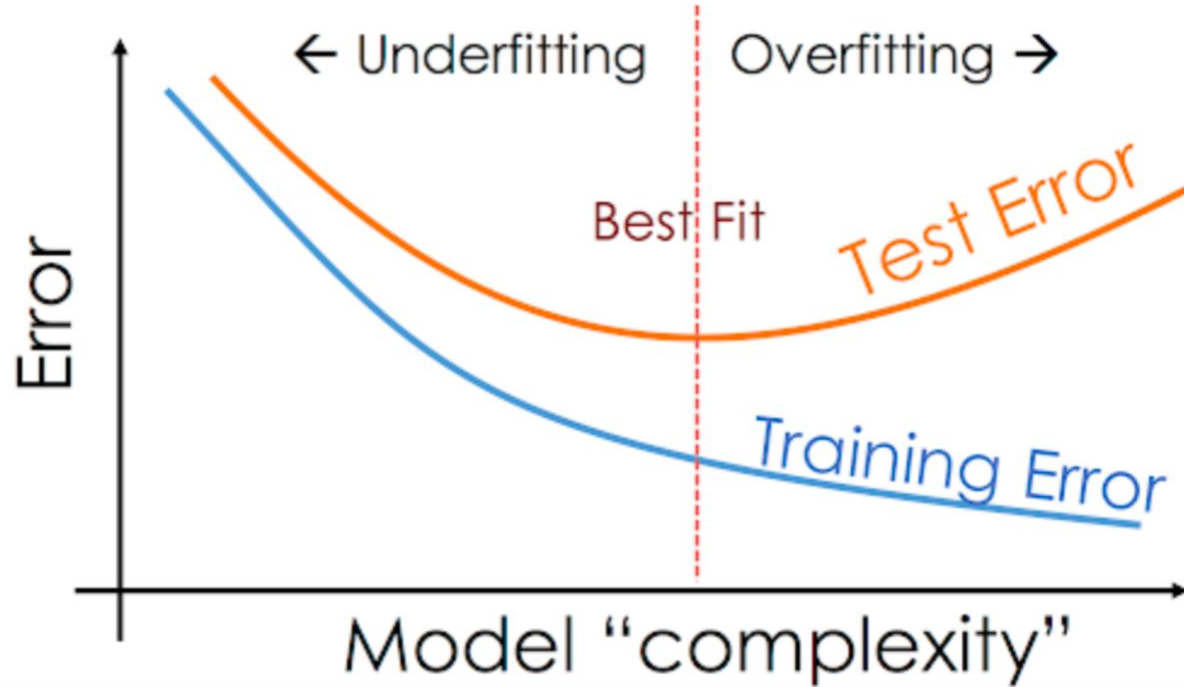
Cross-validation

- Train/test split
- K-folds cross validation



Under/Over-fitting check

Loss, or could
be any other
metrics of
interest



Or training epochs (number of iterations)