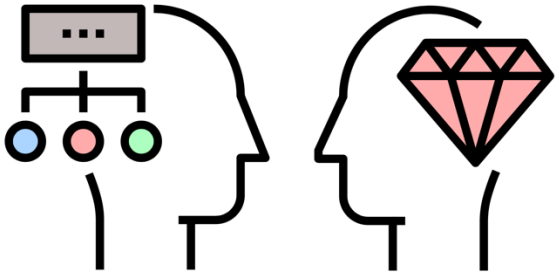
IMPERIAL
Department of Materials

Machine Learning for Materials

5. Classical Learning

Aron Walsh

Department of Materials
Centre for Processable Electronics



Module Contents

ML Model Map

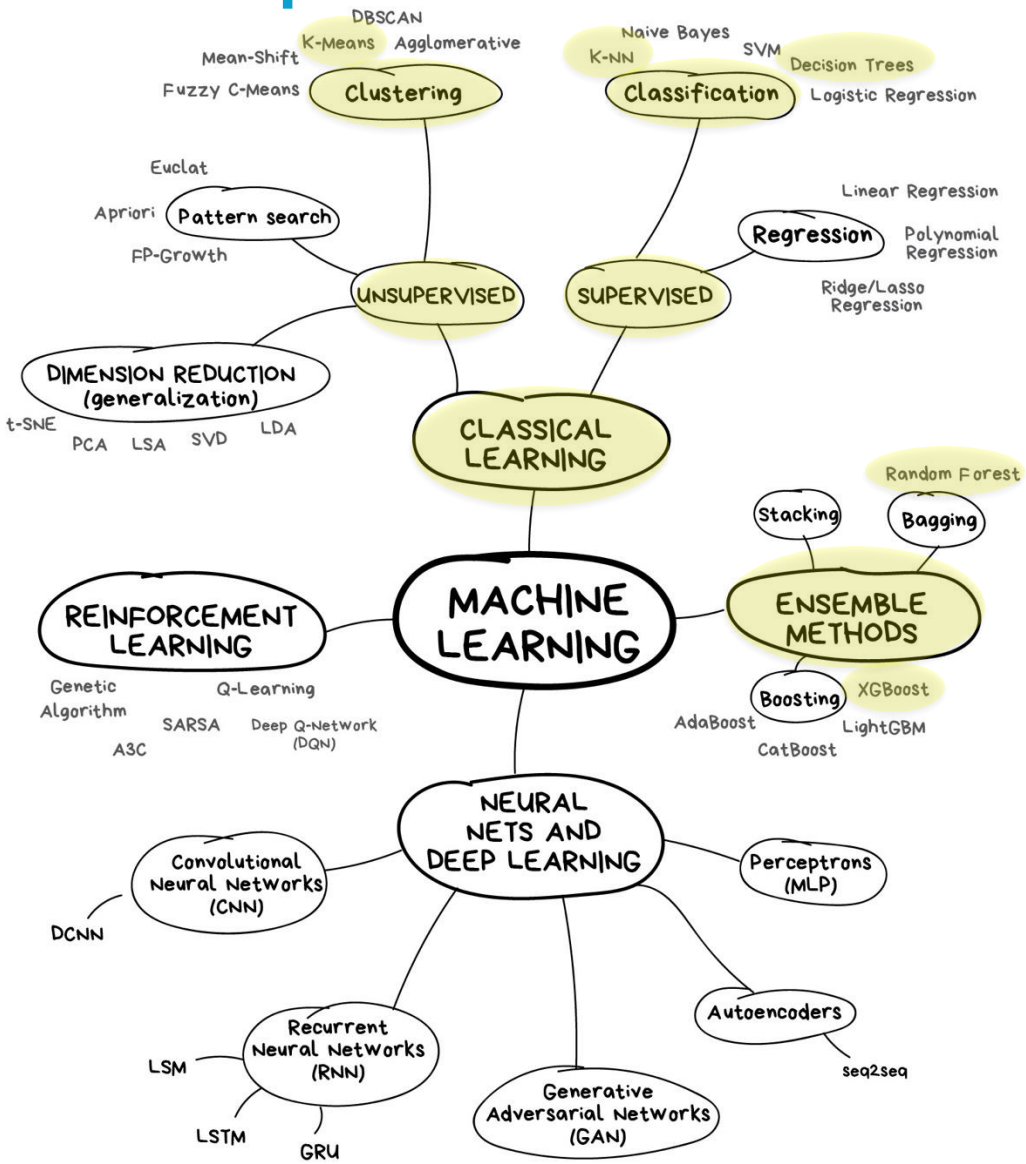
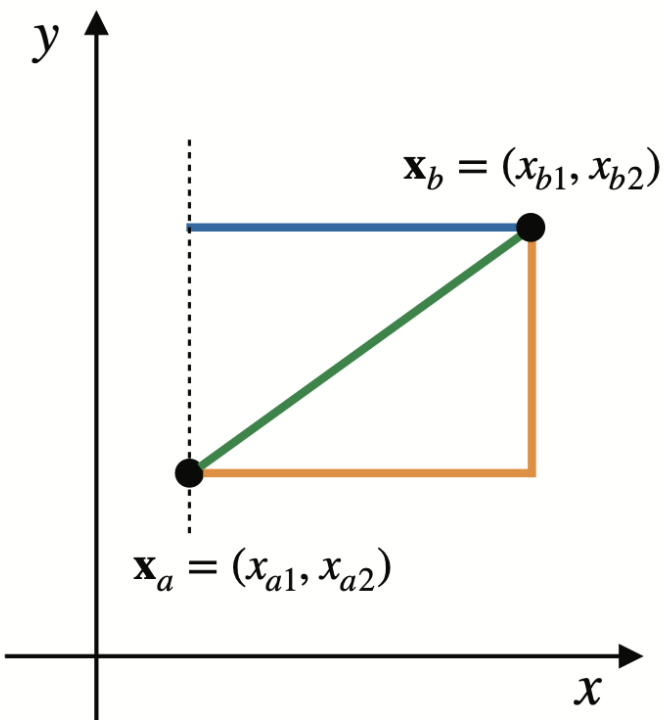


Image from https://vas3k.com/blog/machine_learning

Distance in High Dimensions

Minkowski distance is a convenient expression:

$$d(\mathbf{A},\mathbf{B}) = \left(\sum_{i=1}^{n} |A_i - B_i|^p\right)^{1/p}$$



$p = 2$ Euclidean distance

$$\|\mathbf{x}_a - \mathbf{x}_b\|_2 = (|x_{a1} - x_{b1}|^2 + |x_{a2} - x_{b2}|^2)^{\frac{1}{2}}$$

$p = 1$ Manhattan distance

$$\|\mathbf{x}_a - \mathbf{x}_b\|_M = |x_{a1} - x_{b1}| + |x_{a2} - x_{b2}|$$

$p = \infty$ Chebyshev distance

$$\|\mathbf{x}_a - \mathbf{x}_b\|_\infty = \max\{|x_{a1} - x_{b1}|, |x_{a2} - x_{b2}|\}$$

Image from C. Fu and J. Yang, Algorithms 14, 54 (2021)

Distance in High Dimensions

Distinction between distance measures

- **Euclidean** – straight line between points. Use when data is dense & continuous; features have similar scales
- **Manhattan** – distance following gridlines. Use when data has different scales or grid-like structure
- **Chebyshev** – maximum separation in one dimension. Use to emphasise the largest difference; highlight outliers in feature space

Class Outline

Classical Learning

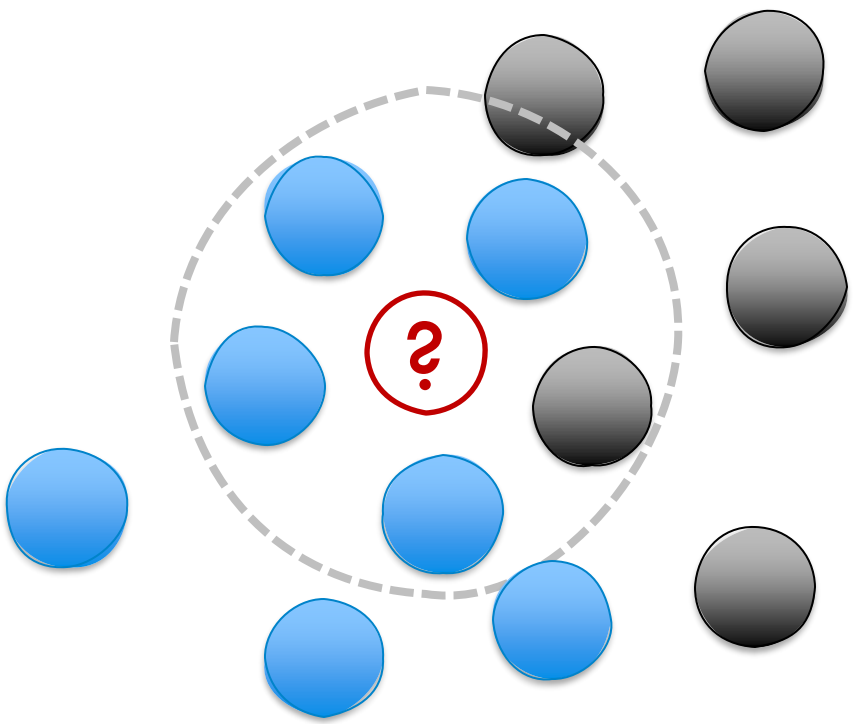
A. *k-nearest neighbours*

B. *k-means clustering*

C. *Decision trees and beyond*

k-Nearest Neighbours (*k*-NN)

Supervised ML model that labels a datapoint based on the properties of its neighbours



Euclidean distance in *n*-dimensions is a common metric to determine *k*-NN

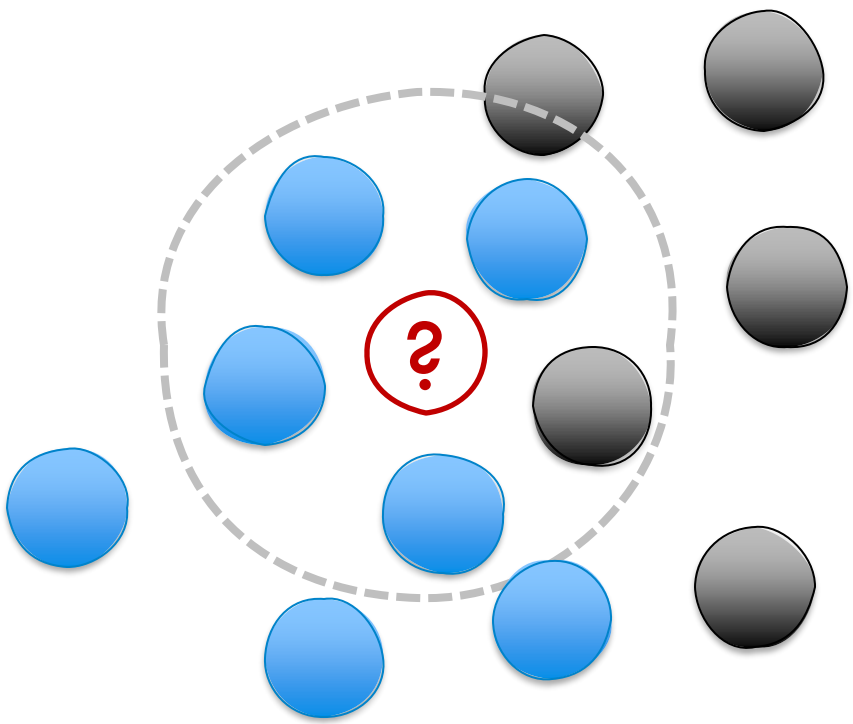
$$\sqrt{(p_1 - q_1)^2 + \cdots + (p_n - q_n)^2}$$

What is the most likely colour of the unknown point?

"Discriminatory Analysis" E. Fix and J. Hodges (1951)

k-Nearest Neighbours (k-NN)

k refers to the number of nearest neighbours to include in the majority vote



Predicted label

Nearest neighbours

$$\boxed{y = mode(k)}$$

Most common value

Here $k = 5$. The limit of $k = 1$ uses the closest neighbour only

"Discriminatory Analysis" E. Fix and J. Hodges (1951)

k-Nearest Neighbours (*k*-NN)

Components required to build a model

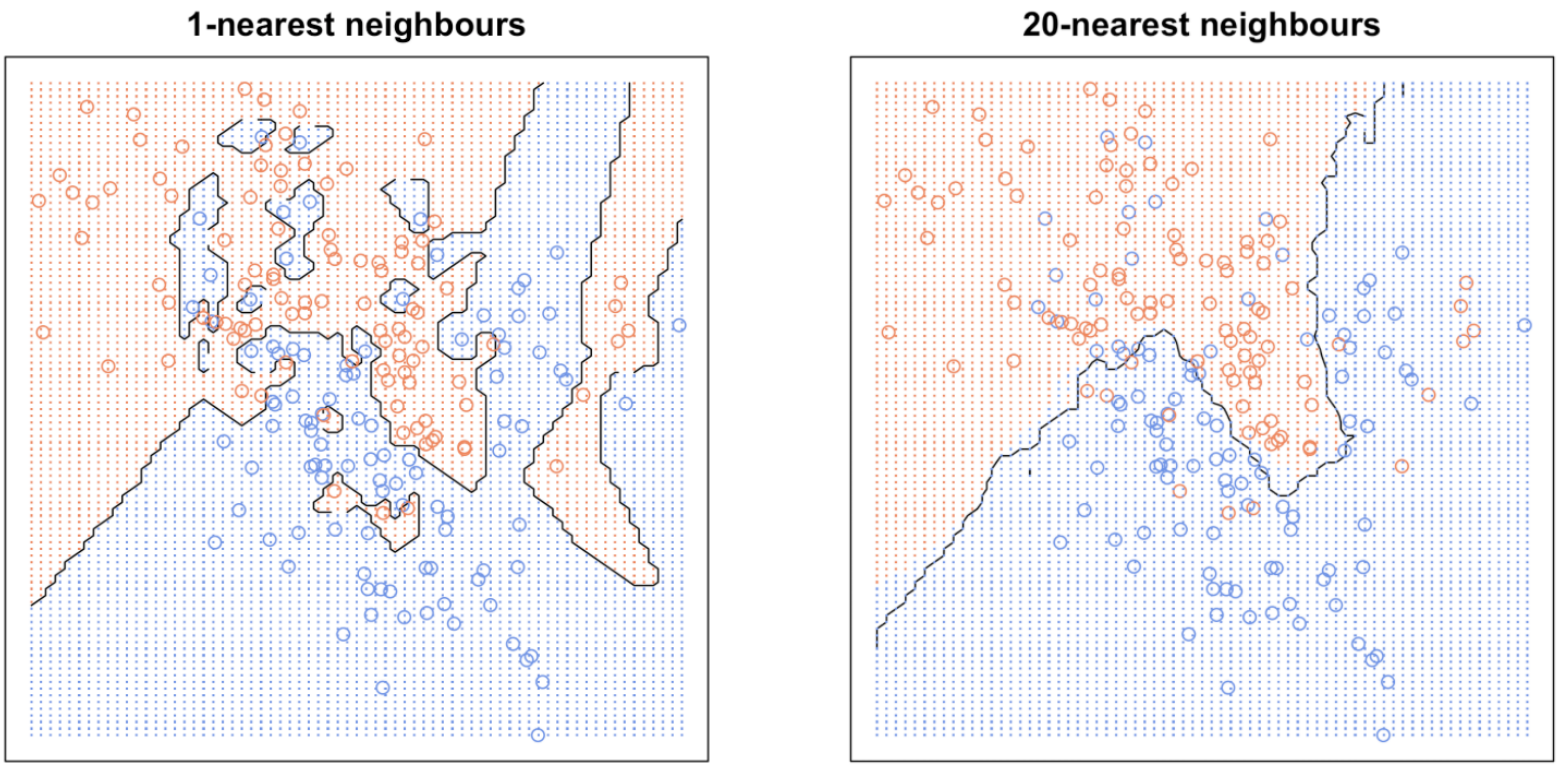
1. **Feature space:** how the object/data is defined in multi-dimensional space, e.g. materials properties such as density or hardness

2. **Distance metric:** method used to measure similarity between data points in feature space, such as Euclidean or Manhattan distance

3. **Training data:** a set of labelled examples with known features and corresponding class labels

k-Nearest Neighbours (*k*-NN)

k-NN can be used for **classification** (majority vote) or **regression** (neighbour weighted average) problems

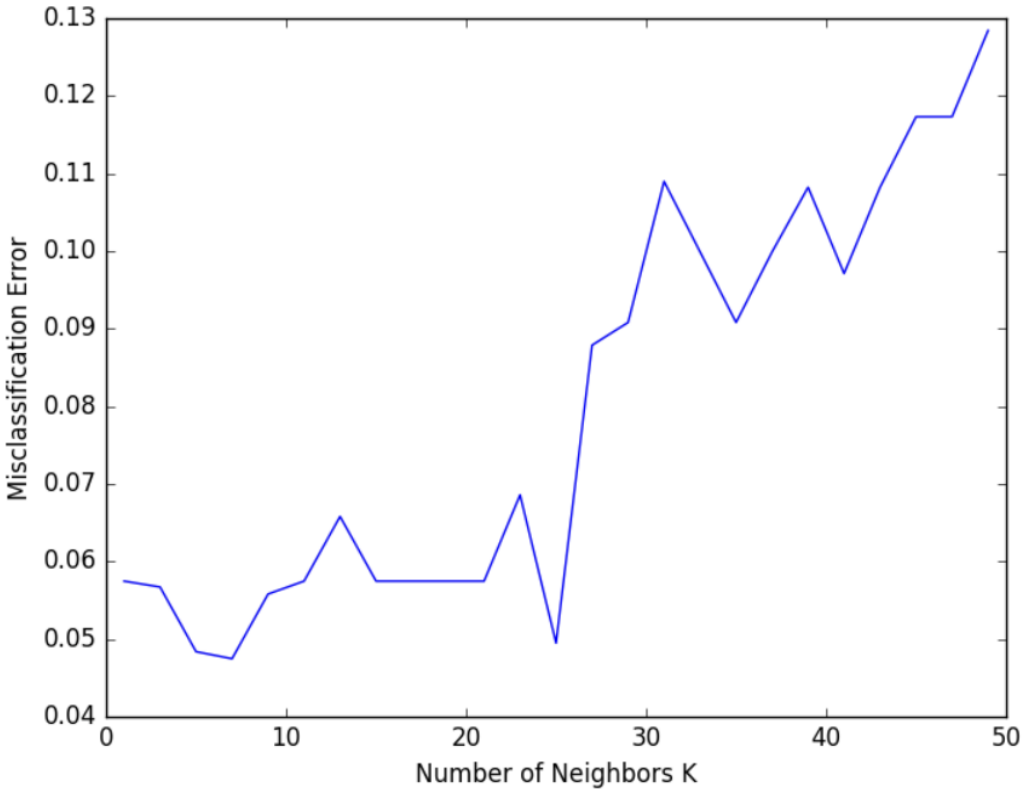


k is a hyperparameter (too small = overfit; large = underfit)

Image from https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor

k-Nearest Neighbours (*k*-NN)

k-NN can be used for **classification** (majority vote) or **regression** (neighbour weighted average) problems



k is a hyperparameter (too small = overfit; large = underfit)

Image from https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor

Model Assessment

Classification metrics: true positives (TP), true negatives (TN), false negatives (FN), false positives (FP)

Metric	Formula	Interpretation
Accuracy	(TP+TN)/(TP+TN+FP+FN)	Overall model performance
Precision	TP/(TP+FP)	Proportion of true positive out of all positive predictions
Recall/ Sensitivity	TP/(TP+FN)	Proportion of actual positives correctly identified
Specificity	TN/(TN+FP)	Proportion of actual negatives correctly identified
F1 score	2TP/(2TP+FP+FN)	Harmonic mean of precision and recall (for imbalanced classes)

k-Nearest Neighbours (*k*-NN)

Where a *k*-NN model may struggle:

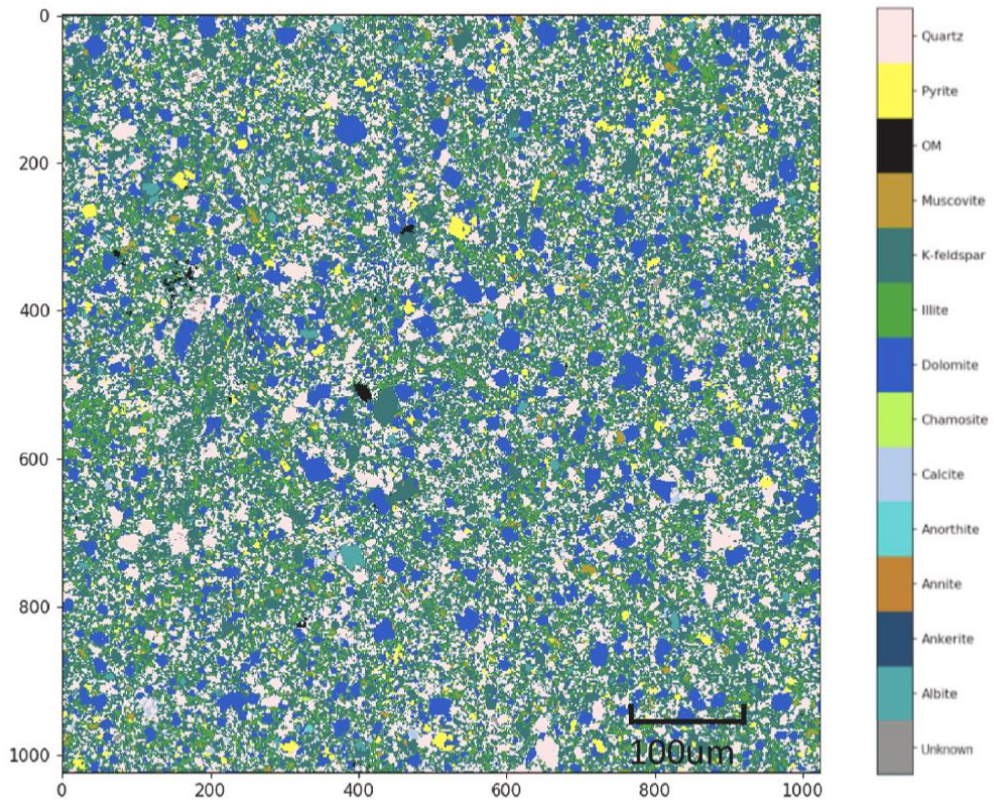
1. Imbalanced data – if there are multiple classes that differ in size, the smallest class may be overshadowed. Addressed by appropriate weighting

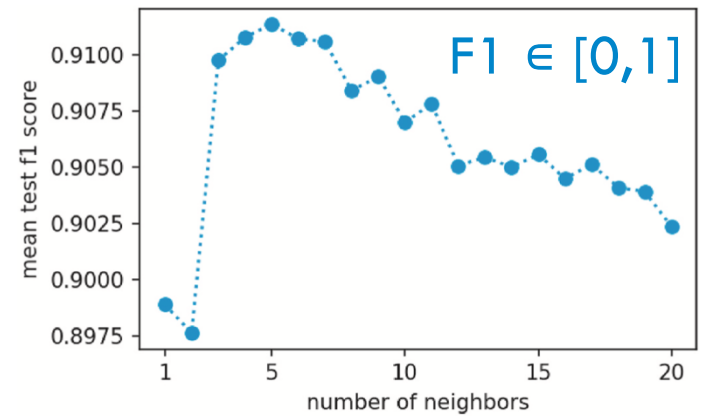
2. Too many dimensions – identifying nearest neighbours and calculating distances can be costly. It may be optimal to apply dimension reduction techniques* first

*Principal Component Analysis (PCA) is popular for this purpose (see Exercises)

k-NN Application: Microscopy

Classification of mixed mineral samples from microscopy (SEM-EDS) datasets



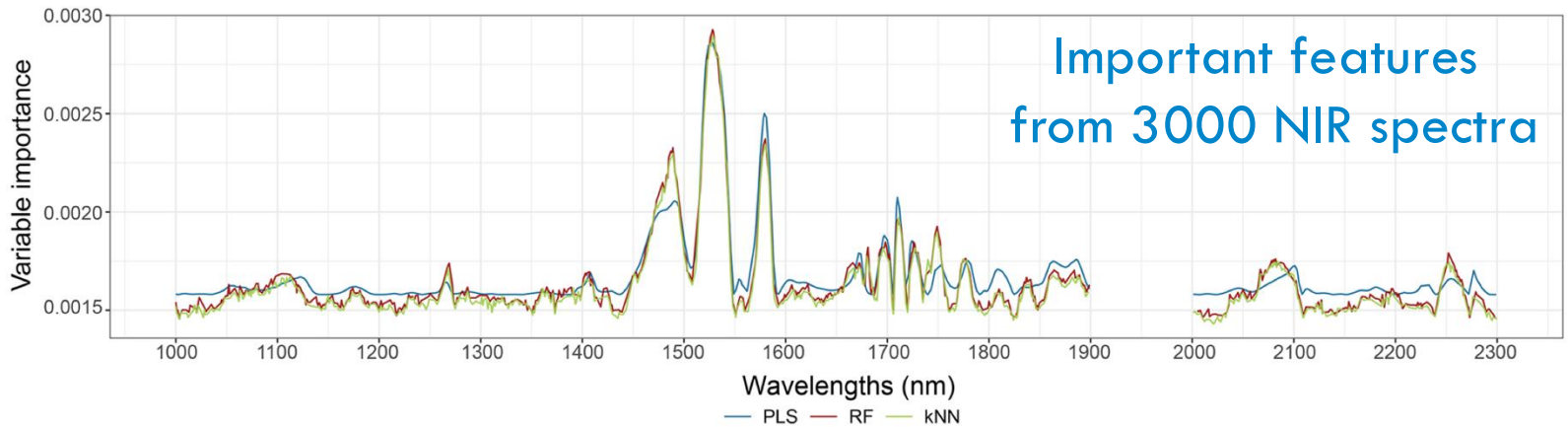


Learning models		F1-score		
		F1-micro	F1-weighted	F1-macro
Machine learning models	Logistic regression	0.8705	0.8691	0.8691
	Linear SVM	0.7980	0.7778	0.7778
	k-Nearest Neighbors	0.9150	0.9139	0.9139
	Random Forest	0.9238	0.9215	0.9215
	ANN	0.8869	0.8873	0.8873
Deep learning model	U-Net	0.8832	0.8784	0.7301

C. Li et al, J. Pet. Sci. Eng. 200, 108178 (2020)

k-NN Application: Vibrational Spectra

Dating of historical books based on near-infrared spectral signatures



Three Models: Partial Least Squares | Random Forest | *k*-NN

groups	subsets	n_s	PLS RMSECV$_{100}$ ± SD (year)	RF RMSECV$_{100}$ ± SD (year)	kNN RMSECV$_{100}$ ± SD (year)
Point	Gutter	1000	12.81 ± 0.10	5.89 ± 0.08	1.32 ± 0.30
	Center	1000	12.07 ± 0.10	6.36 ± 0.06	2.37 ± 0.20
	Margin	1000	12.94 ± 0.11	5.36 ± 0.08	1.65 ± 0.26
All		1000	14.24 ± 0.35	8.10 ± 0.17	5.89 ± 0.69
		3000	12.00 ± 0.04	4.68 ± 0.04	1.57 ± 0.13

F. Coppola et al, J. Am. Chem. Soc. 145, 12305 (2023)

Class Outline

Classical Learning

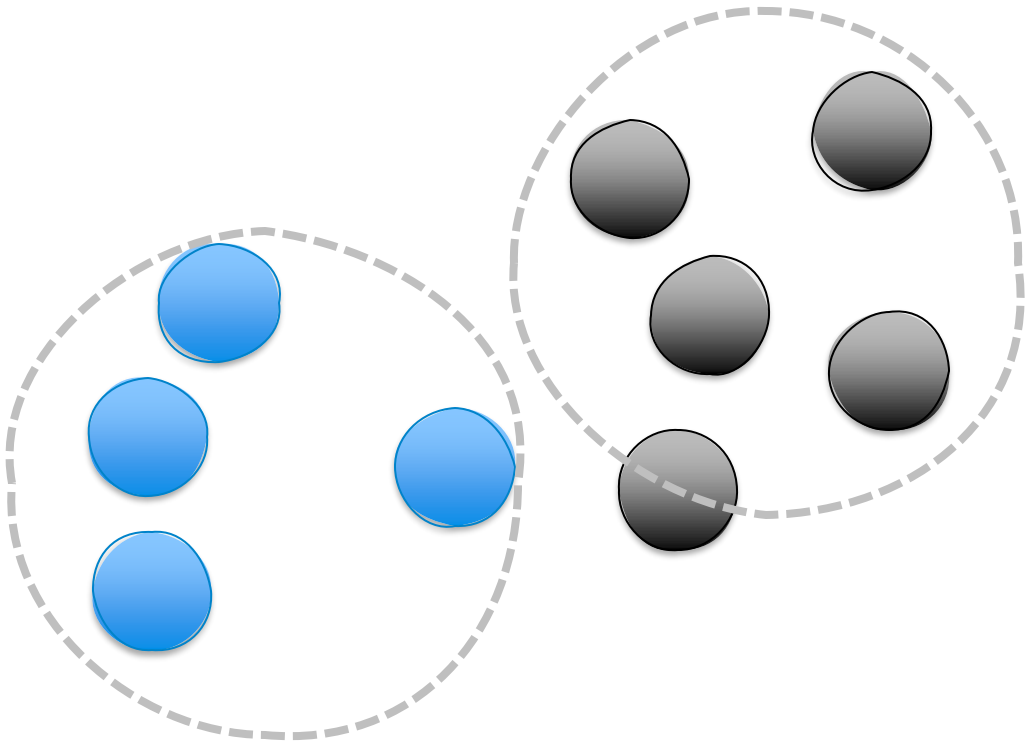
A. *k-nearest neighbours*

B. k-means clustering

C. *Decision trees and beyond*

k-Means Clustering

Unsupervised model that groups data into clusters, where *k* is the number of clusters identified



Place *n* observations into *k* sets

$\mathbf{S} = \{S_1 \ldots S_k\}$

Datapoints within a cluster should be similar

"Sur la division des corps matériels en parties" H. Steinhaus (1957)

k-Means Clustering

Main components of a *k*-means model

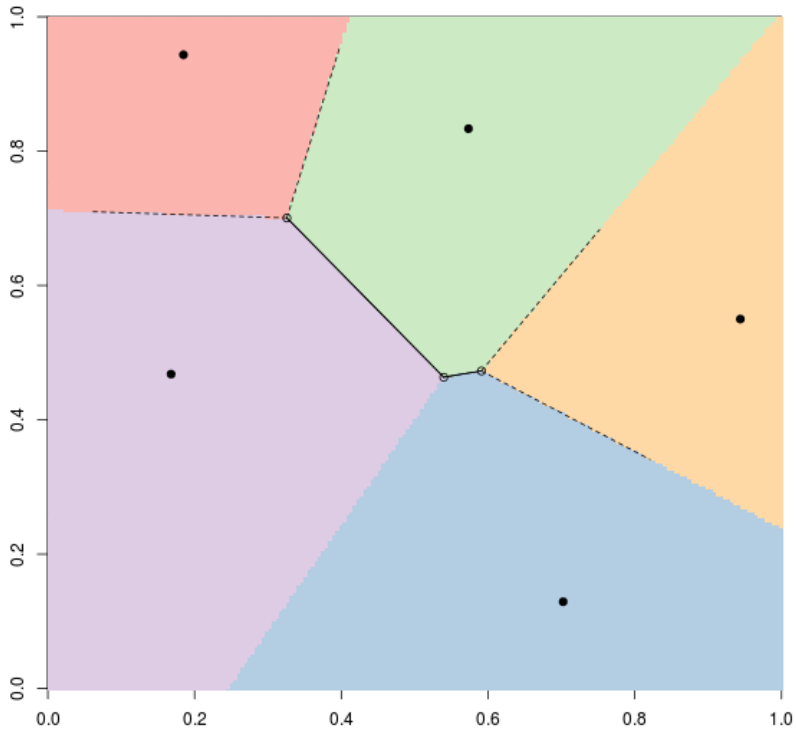
1. **Initialisation:** Choose the number of clusters *k* to identify. Centroids can be distributed randomly

2. **Distance metric:** Similar to *k*-NN, a distance measure is required to define the similarity or dissimilarity, e.g. Euclidean or Manhattan

3. **Assignment:** Each point is assigned to the nearest centroid. The mean of all points in each cluster is calculated. This process iterates until convergence

k-Means Clustering

Unsupervised model groups data into clusters, where *k* is the number of clusters identified



Place *n* observations into *k* sets

$$\mathbf{S} = \{S_1 \ldots S_k\}$$

Minimise within cluster sum of squares (WSS)

$$\mathbf{J} = \sum |x_i - \mu_k|^2$$

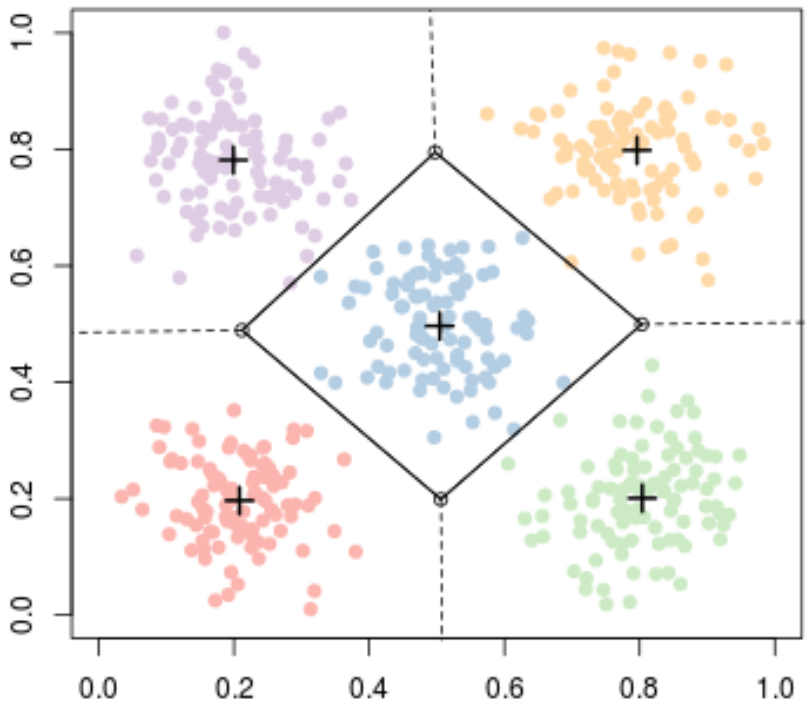
centroid of cluster *k*

An iterative algorithm is used to minimise cluster variance

Animation from https://freakonometrics.hypotheses.org/19156

k-Means Clustering

Unsupervised model groups data into clusters, where *k* is the number of clusters identified



Place *n* observations into *k* sets

$\mathbf{S} = \{S_1 \ldots S_k\}$

Minimise within cluster sum of squares (WSS)

$\mathbf{J} = \sum |x_i - \mu_k|^2$

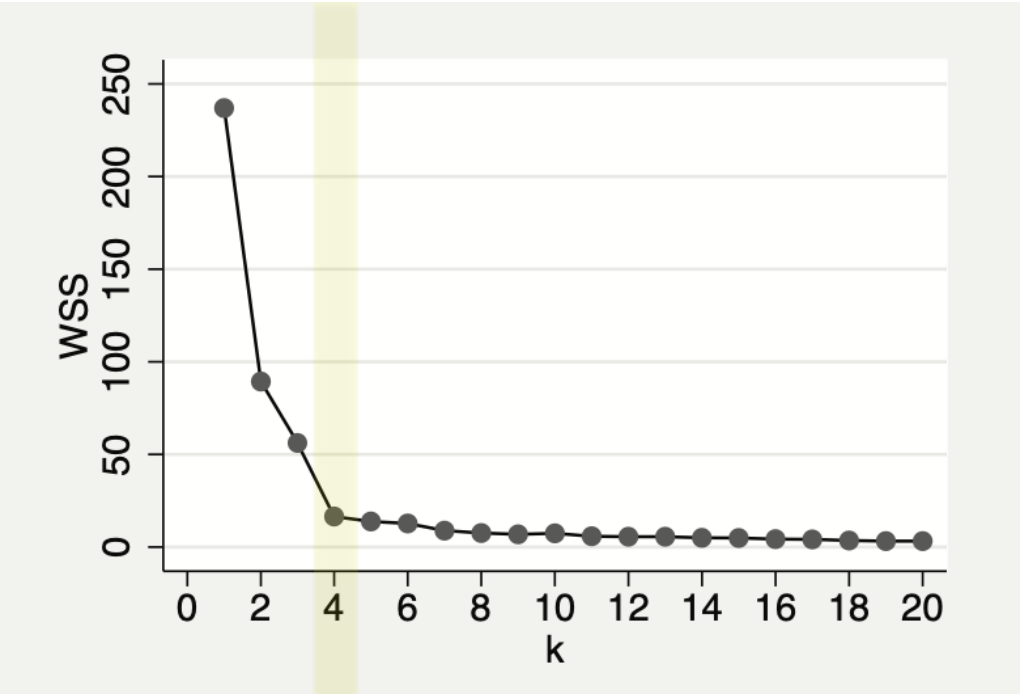
centroid of cluster *k*

Note the linear (piecewise straight) cluster boundaries

Image from https://freakonometrics.hypotheses.org/19156

k-Means Clustering

k is a hyperparameter. How many clusters to choose?



A *scree plot* shows how within-cluster scatter decreases with *k*

The kink at $k = 4$ suggests the optimal number

As *k* increases, the similarity within a cluster increases, but in the limit of $k = n$, each cluster is only one data point

A. Makles, Stata Journal 12, 347 (2012)

k-Means Clustering

The strength of *k*-means is simplicity, but it has limitations:

1. No dual membership – even if a data point falls at a boundary, it is assigned to one cluster only

2. Clusters are discrete – no overlap or nesting is allowed between clusters

Extended techniques such as spectral clustering compute the probability of membership in each cluster

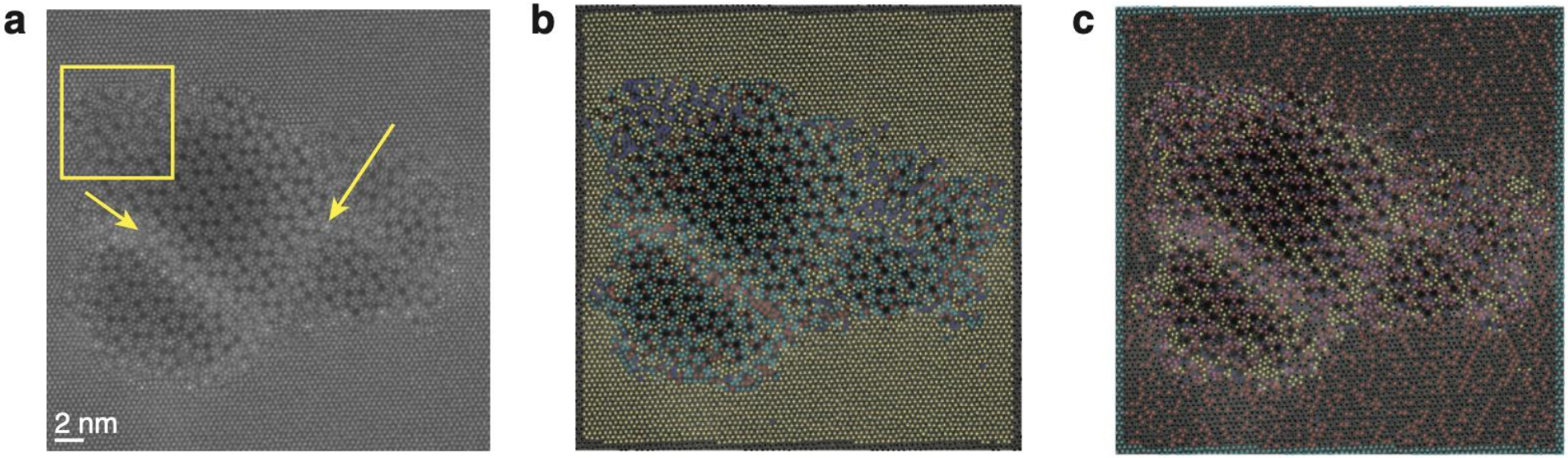
k-Means Application: Microscopy

Clustering in STEM images of multicomponent (Mo–V–Te–Ta) metal oxides

Original data

***k*-means**
(*k*=4, Euclidean distance)

***k*-means**
(*k*=4, Angle metric)



Two representations of the local atomic environment are used for grouping into clusters

A. Belianinov et al, Nature Commun. 6, 7801 (2015)

Class Outline

Classical Learning

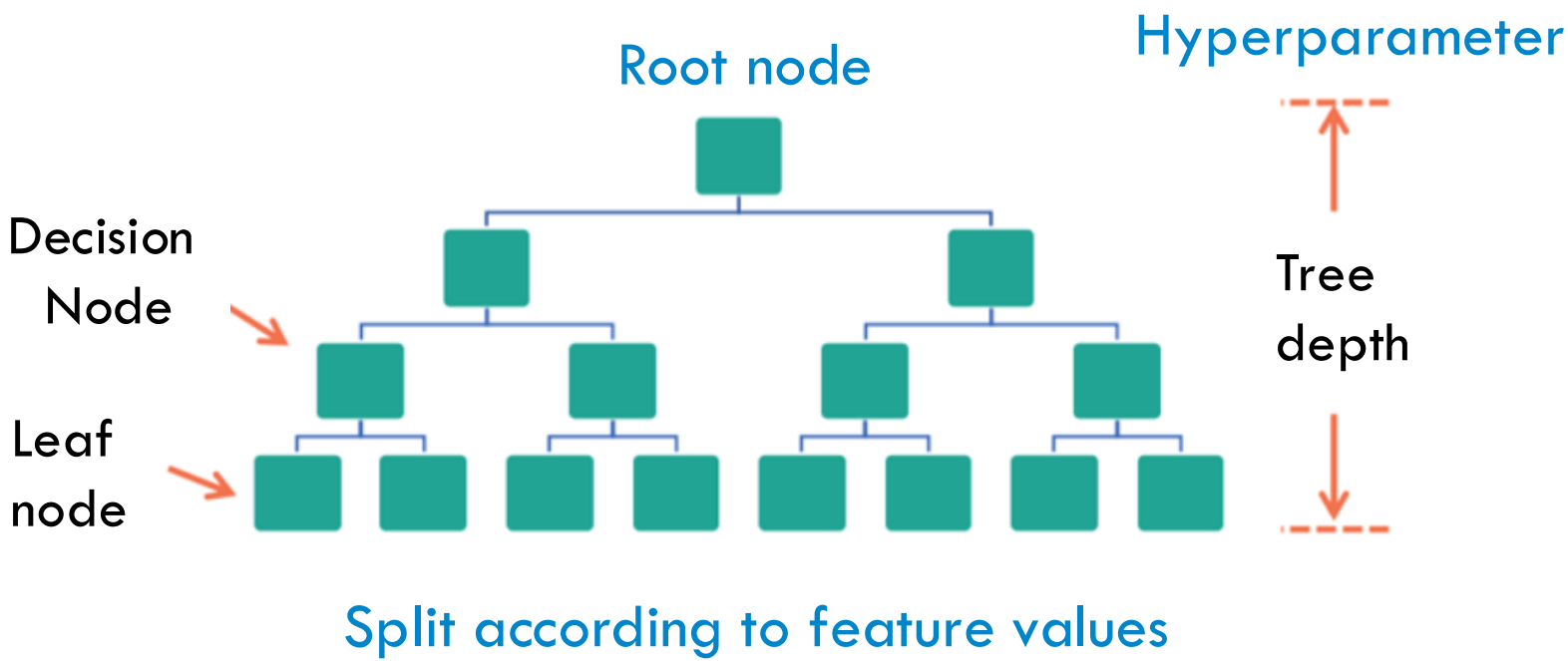
A. *k-nearest neighbours*

B. *k-means clustering*

C. Decision trees and beyond

Decision Trees

Supervised tree-like model splits data multiple times according to feature values (decision rules)

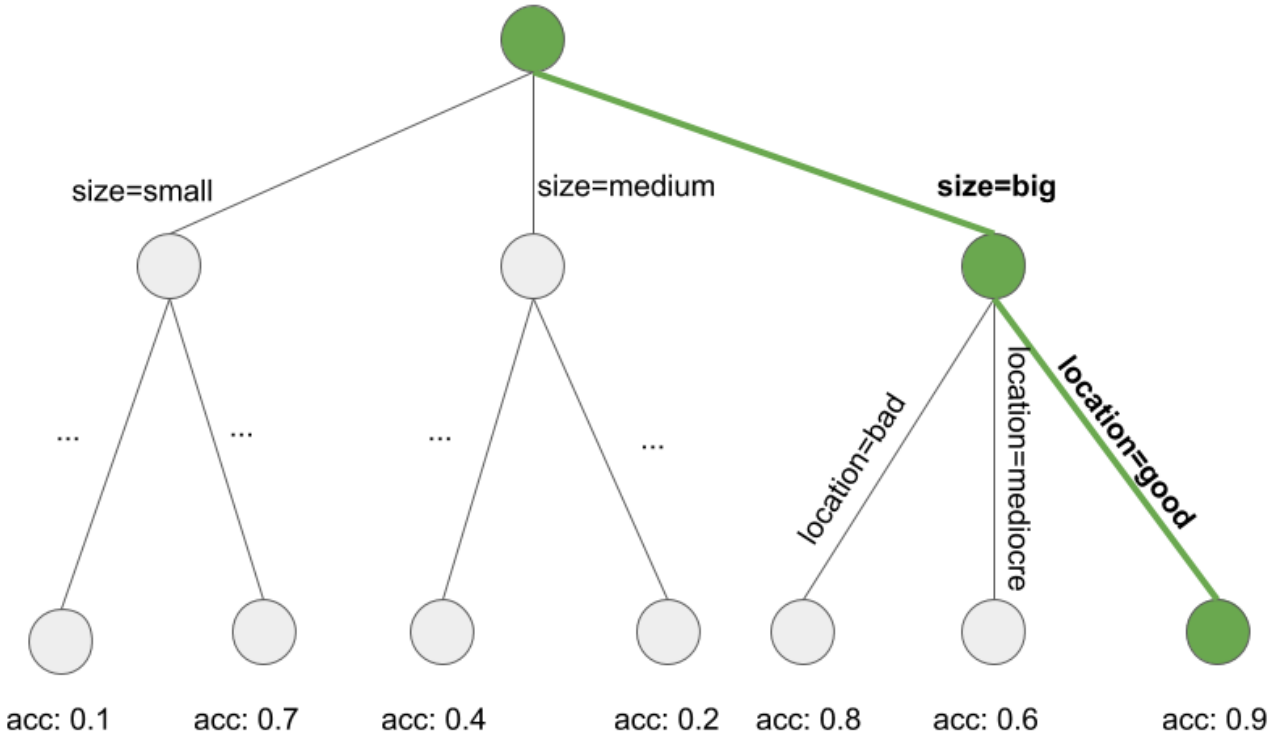


Can be used for classification or regression problems

J. N. Morgan and J. A. Sonquist, J. Am. Stat. Assoc. 58, 302 (1963), etc.

Decision Trees

An interpretable model. Each prediction can be broken down into a sequence of decisions



CART is a common training algorithm (e.g. in scikit-learn)

Image from https://christophm.github.io/interpretable-ml-book

Decision Trees

An interpretable model. Each prediction can be broken down into a sequence of decisions

Tree to assign class *N*

```
                [ Start ]
               Is x2 < 3?
                /        \
            Yes           No
           /               \
    [ Node 1 ]          [ Node 2 ]
    Is x5 < 7?          Is x5 ≥ 7?
     /        \          /        \
    Yes       No       Yes        No
   /           \        \          \
Class A     Class B  Class C     Class D
```

Pseudo-code

```
def assign_class(x2, x5):
    if x2 < 3:
        if x5 < 7:
            return "Class A"
        else:
            return "Class B"
    else:
        if x5 >= 7:
            return "Class D"
        else:
            return "Class C"
```

6 decisions are made leading to 4 terminal nodes

Decision Trees

Main steps to build a decision tree model

1. **Feature selection:** Identify the relevant features from the data that contribute to decision-making
2. **Splitting criteria:** Determine the best feature and test combination at each node using metrics such as information gain
3. **Tree building:** Recursively apply splitting criteria to grow child nodes, stopping when a predefined condition is met (e.g. maximum depth)

Decision Trees

A simple model that is applicable to many problems, but with limitations

1. **Instability** – a slight change in training data can trigger changes in the split and a different tree. Vulnerable to overfitting

2. **Inaccuracy** – the "greedy" method of using the best binary question first may not lead to the best overall model

There are many extensions of simple decision trees…

Ensemble Models

Combine predictions from multiple models through majority voting or averaging

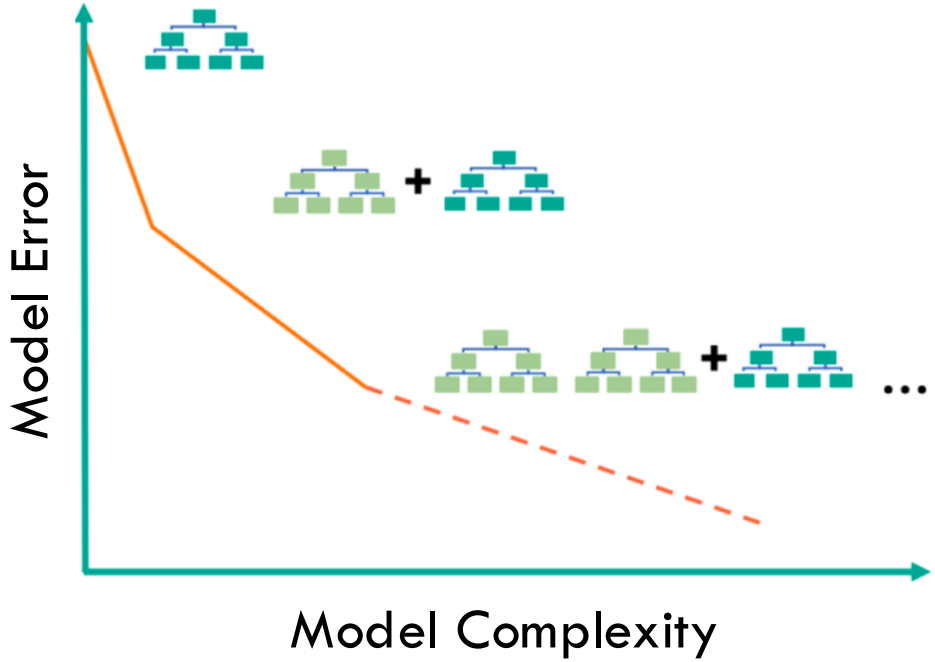
Model 1						60% Accurate
Model 2						40% Accurate
Model 3						60% Accurate
Ensemble						80% Accurate

An ensemble formed by majority voting yields higher accuracy than the separate models

Increased predictive power comes at the cost of reduced interpretability (a step towards “black boxes”)

From 木 to 林 to 森

Decision trees can be combined for more powerful classification & regression models



Random Forests

Ensemble of independent decision trees

Gradient Boosted Regression

Ensemble of coupled decision trees

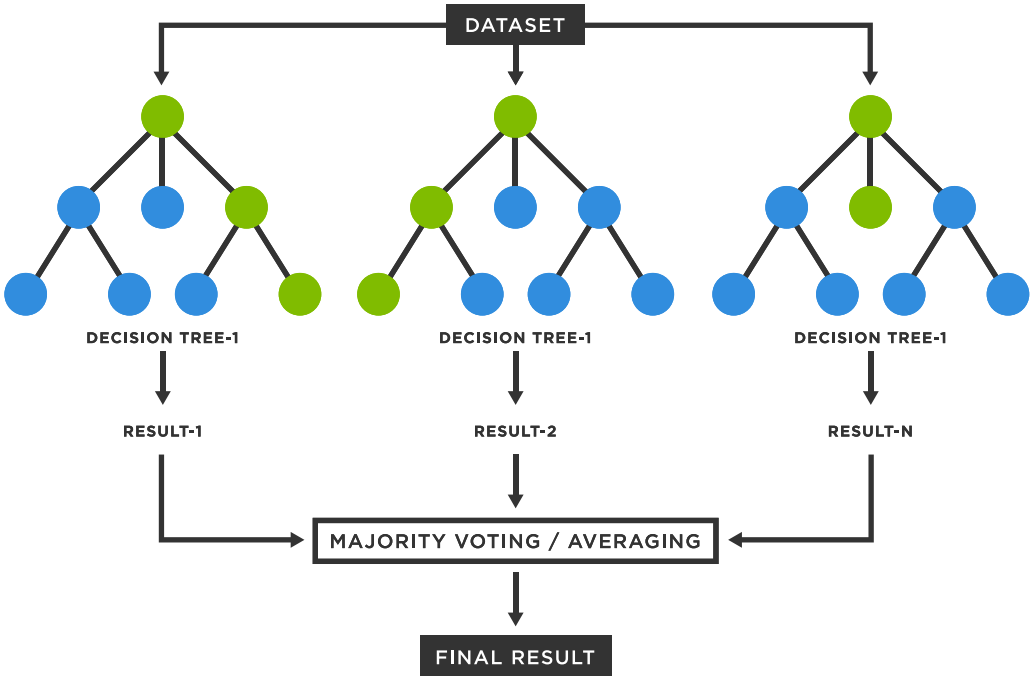
$$\mathbf{y} = \sum_{i=1}^{n} \gamma_i \, \mathrm{tree}_i(\mathbf{x})$$

Figure from D. W. Davies et al, Chem. Mater. 31, 7221 (2019)

Random Forests

Model built from an ensemble of decision trees.

Hyperparameters: no. trees, max depth, samples…



Bagging Method

Each tree is generated from a random subset of training data and a random subset of features (bootstrap aggregation)

Correct predictions can be reinforced, while (uncorrelated) errors are canceled out

Decision Forests: T. K. Ho, IEEE Trans. Pattern Anal. Mach. Intell. 20, 832 (1995)

Gradient Boosted Regression (GBR)

Algorithm that combines "weak learners" (decision trees) to build the best model

$$\mathbf{y} = \gamma_1 \, \mathrm{tree}_1(\mathbf{x}) + \gamma_2 \, \mathrm{tree}_2(\mathbf{x}) + \cdots \gamma_\mathrm{n} \, \mathrm{tree}_n(\mathbf{x})$$

GBR Approach

1. Use a weak learner (tree_1) to make predictions
2. Iteratively add trees to optimise the model (following the error gradient); scikit default of $n = 100$

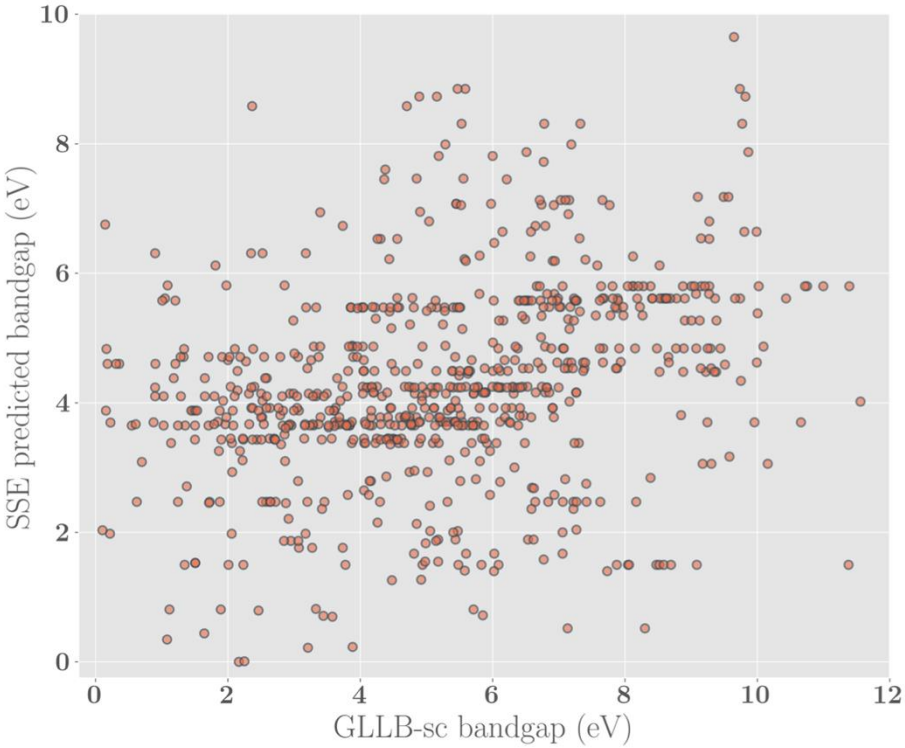
"When in doubt, use XGBoost"
Kaggle competition winner Owen Zhang

XGBoost: T. Chen and C. Guestrin, arXiv 1603.0275 (2016)

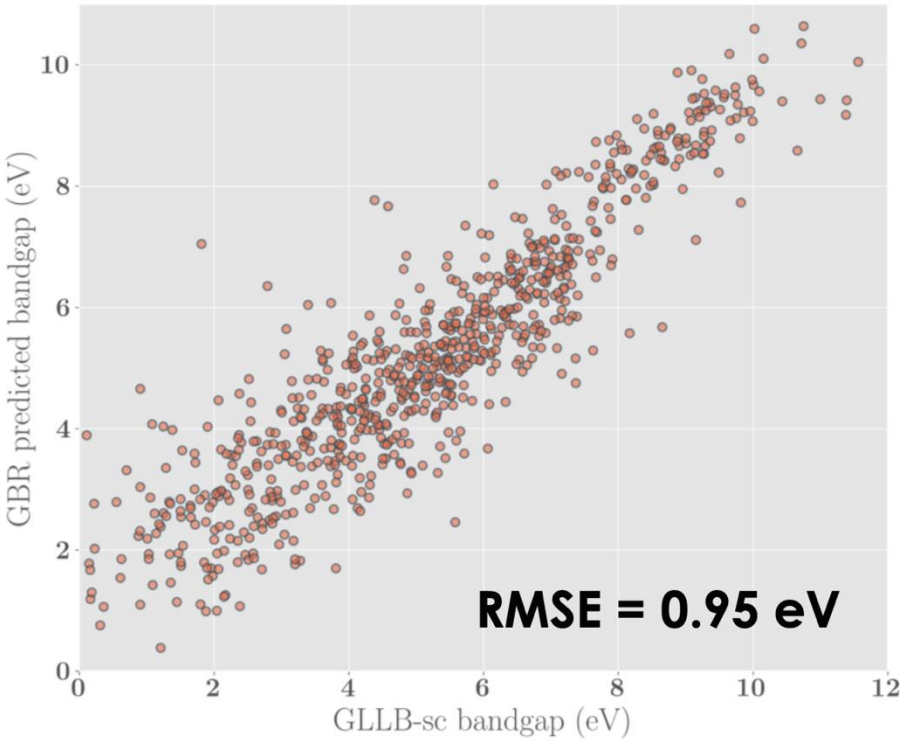
GBR Application: Band Gaps

Predictions of metal oxide band gaps from a dataset of 800 materials (GLLB/DFT; Castelli 2015)

Solid-state energy scale (SSE)



Gradient boosted regression (GBR)



Models use compositional information only (no structure)

D. W. Davies et al, Chem. Mater. 31, 7221 (2019)

GBR Application: Band Gaps

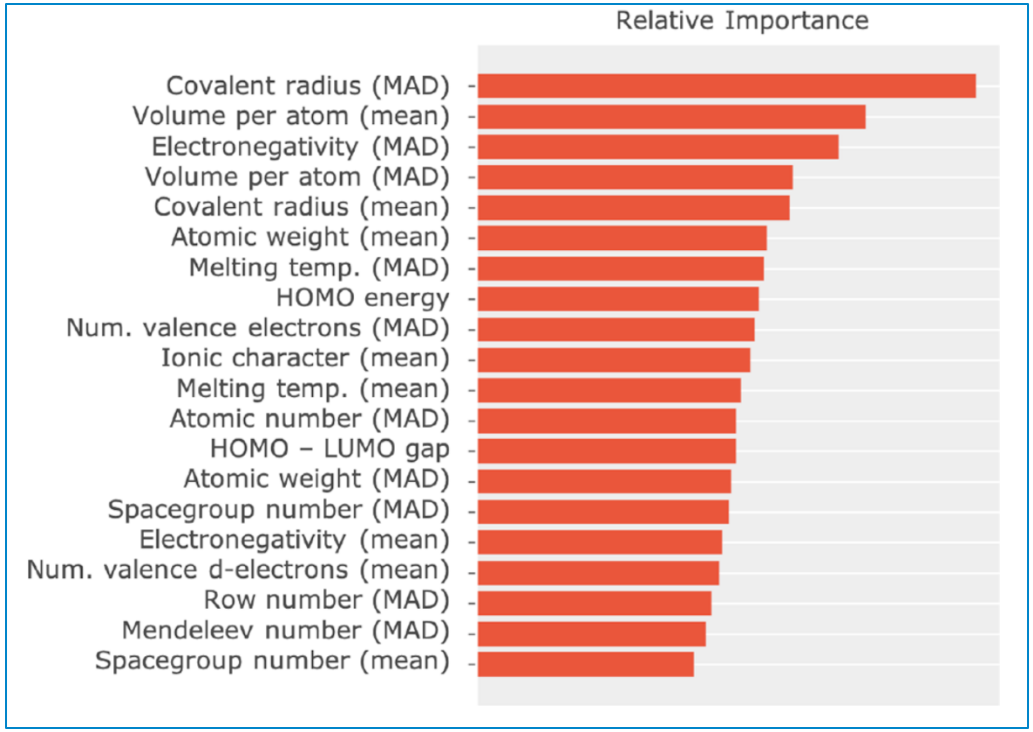
Predictions of metal oxide band gaps from a dataset of 800 materials (GLLB/DFT; Castelli 2015)

Model hyperparameters

tree-specific parameters	
min. compounds needed to split nodes	65
max. depth of tree	20
min. compounds required at leaf nodes	1
max. features considered per tree	86
boosting parameters	
fraction of compounds to fit each tree	0.9
learning rate	0.01
number of decision trees	1000

20 most important features

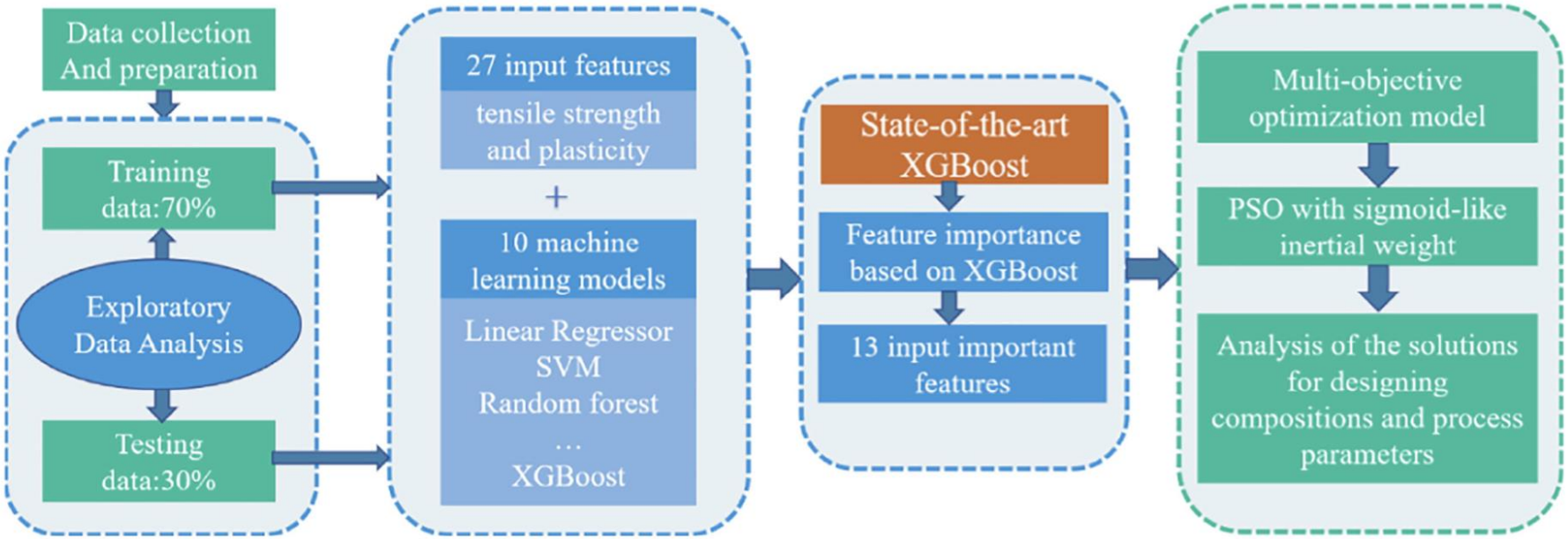
(from 149 generated using Matminer)



D. W. Davies et al, Chem. Mater. 31, 7221 (2019)

GBR Application: Steel

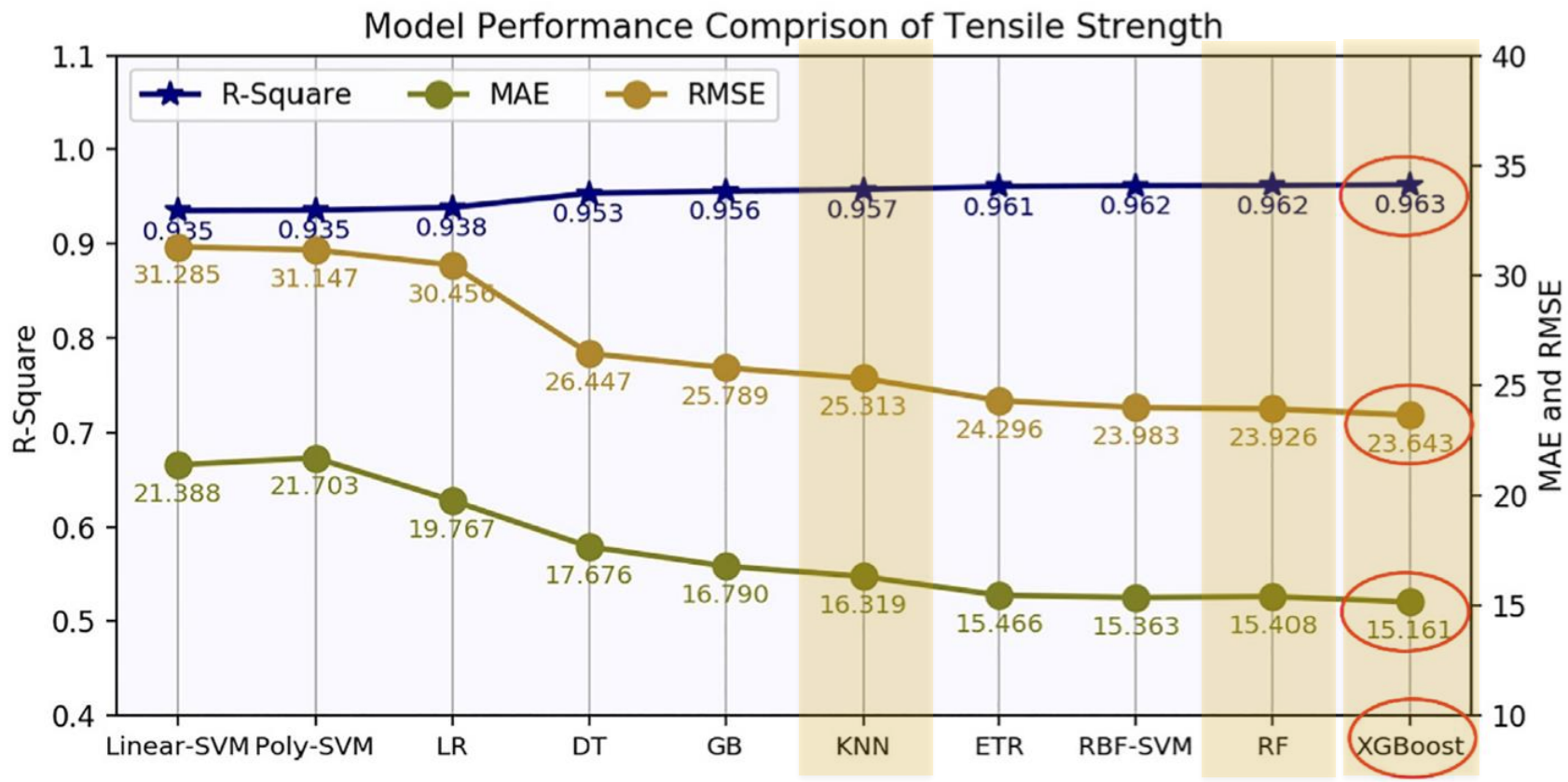
Multivariable optimisation of steel strength and plasticity using 63,000 samples



K. Song, et al, Comp. Mater. Sci. 174, 109472 (2020)

GBR Application: Steel

Multivariable optimisation of steel strength and plasticity using 63,000 samples



K. Song, et al, Comp. Mater. Sci. 174, 109472 (2020)

Class Outcomes

1. Describe the *k*-nearest neighbour model
2. Describe the *k*-means clustering model
3. Explain how a decision tree works and their combination in ensemble methods
4. Assess which types of model could be suitable for a particular problem

Activity:

Metal or insulator?