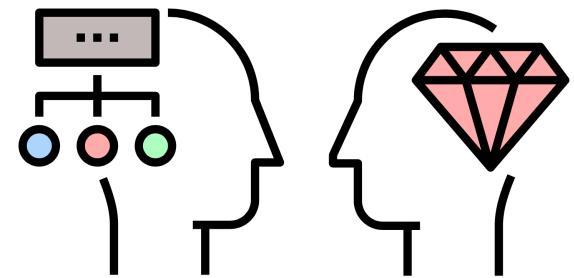


Machine Learning for Materials

5. Classical Learning

Aron Walsh

Department of Materials
Centre for Processable Electronics



Course Contents

1. Course Introduction
2. Materials Modelling
3. Machine Learning Basics
4. Materials Data and Representations
- 5. Classical Learning**
6. Artificial Neural Networks
7. Building a Model from Scratch
8. Recent Advances in AI
9. and 10. Research Challenge

ML Model Map

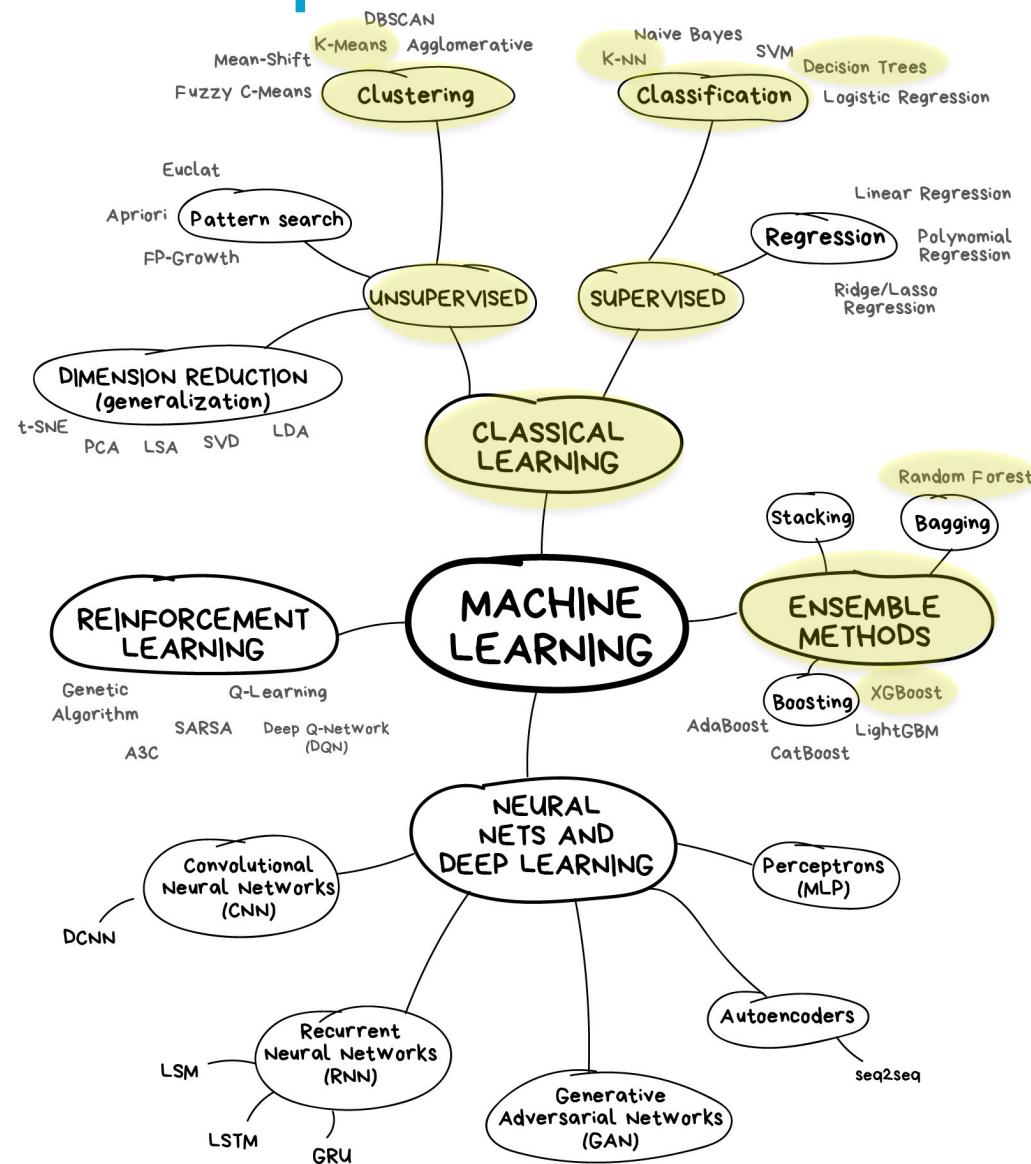
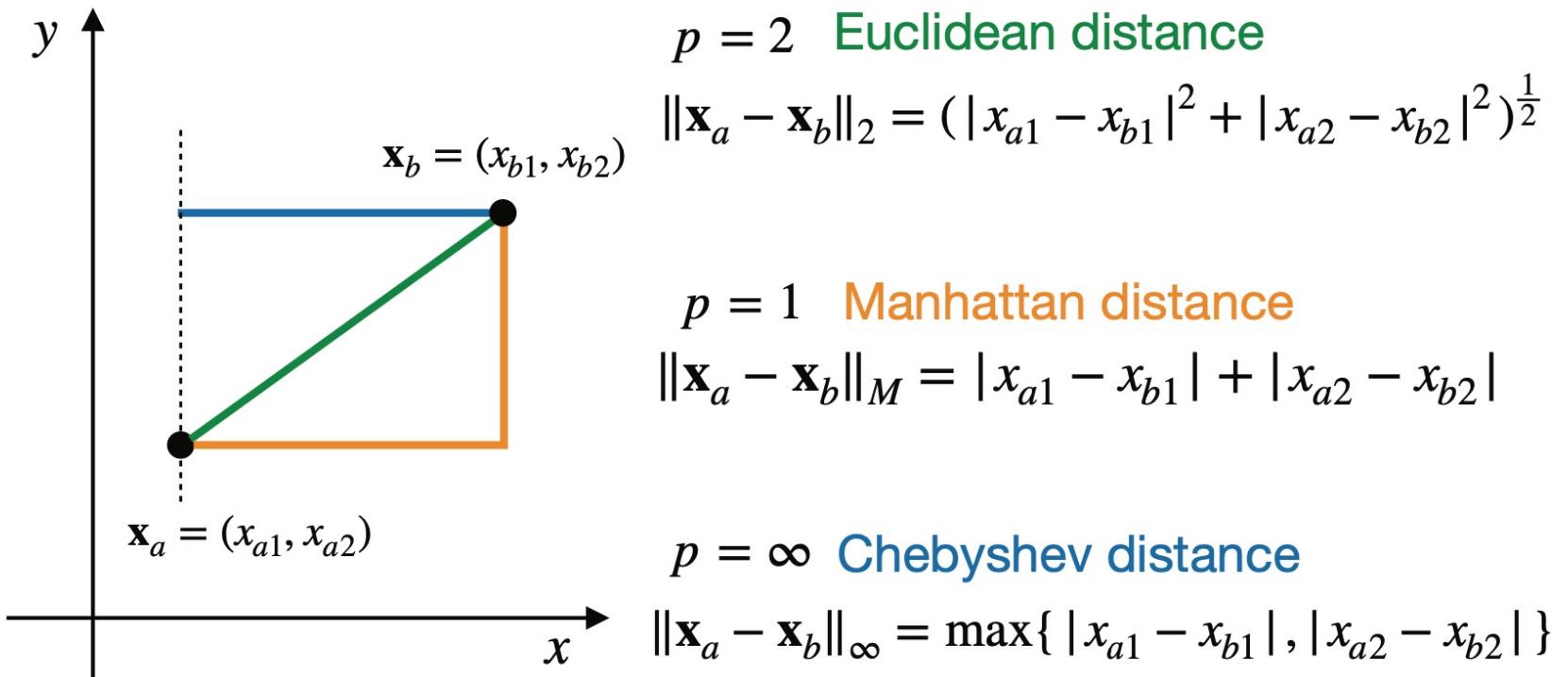


Image from https://vas3k.com/blog/machine_learning

Distance in High Dimensions

Minkowski distance is a convenient expression:

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



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 largest difference; highlight outliers in feature space

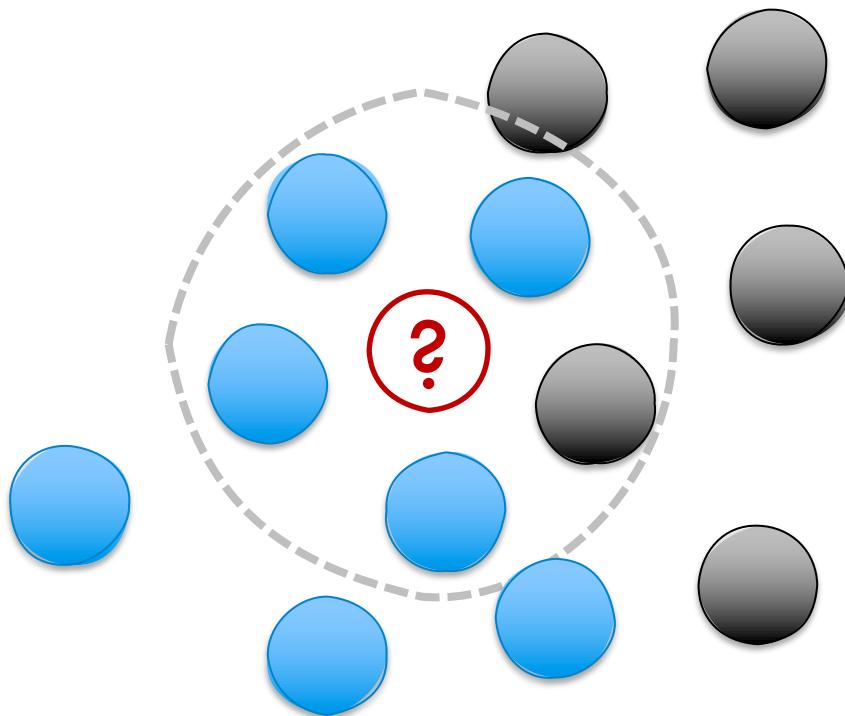
Class Outline

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



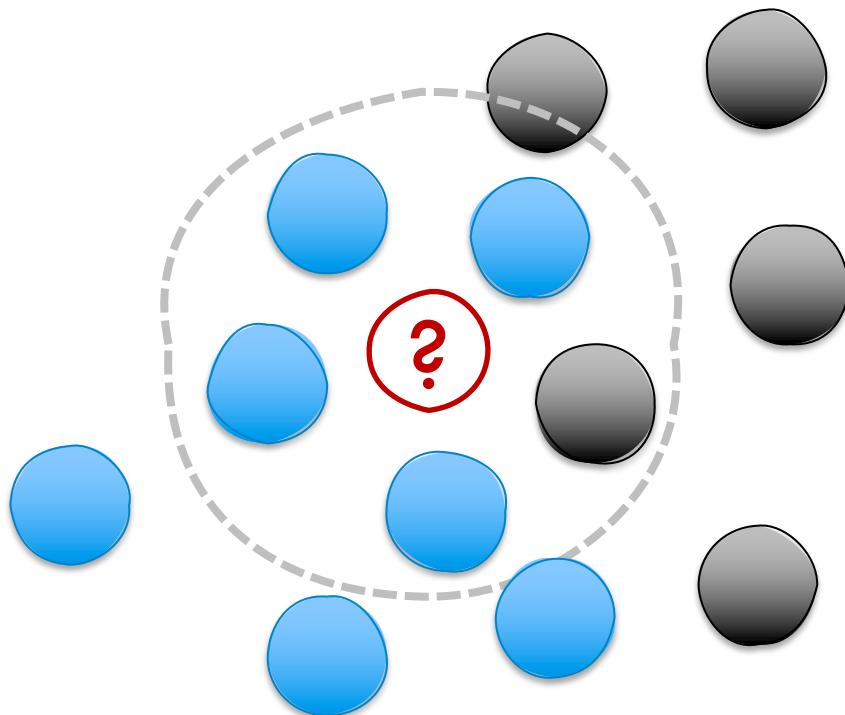
Euclidean distance in *n*-dimensions is a common metric to determine *k*-NN

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

What is the most likely colour of the unknown point?

k -Nearest Neighbours (k -NN)

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



Predicted label	Nearest neighbours
$y = \text{mode}(k)$	
	Most common value

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

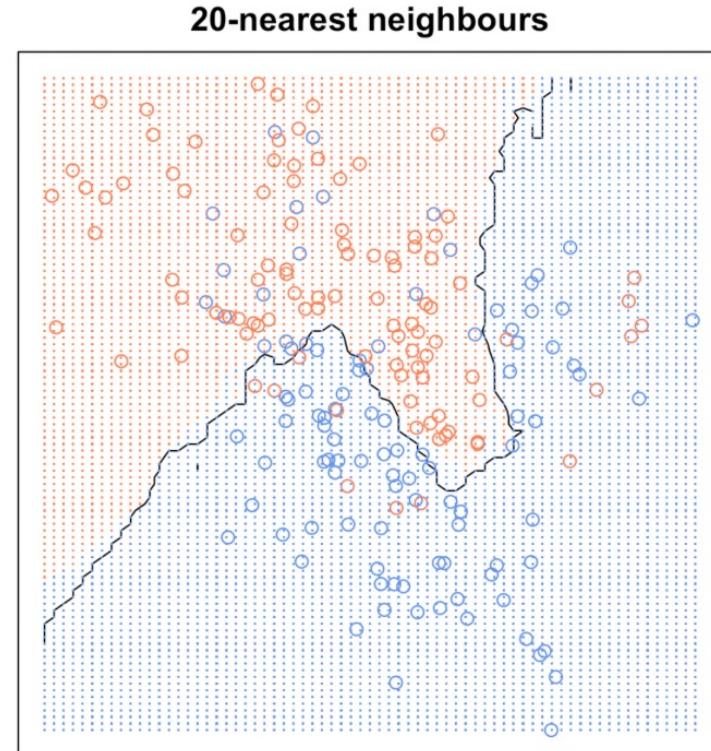
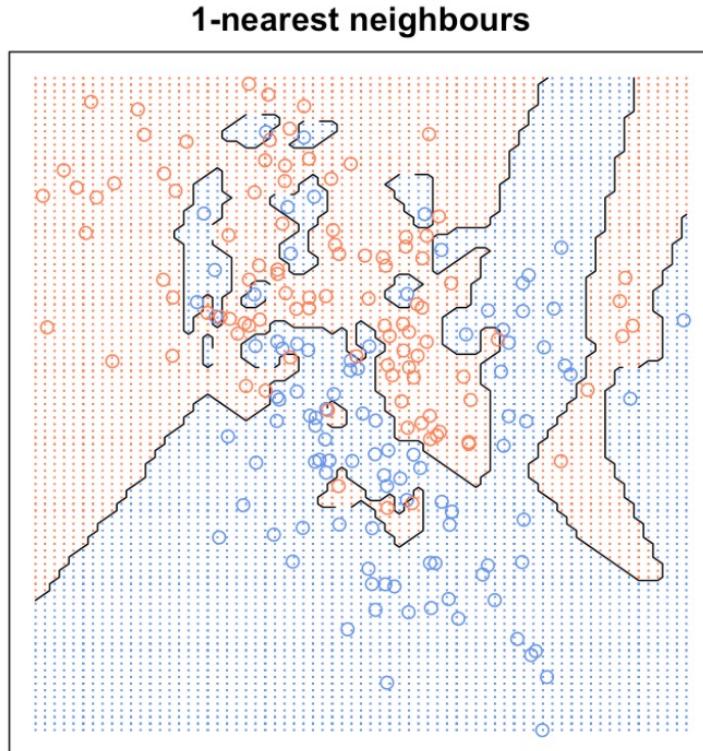
***k*-Nearest Neighbours (*k*-NN)**

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:** How the object/data is separated in multi-dimensional space, e.g. Euclidean or Manhattan distance measures
- 3. Training data:** Labelled examples are required; features and their corresponding classes

k-Nearest Neighbours (*k*-NN)

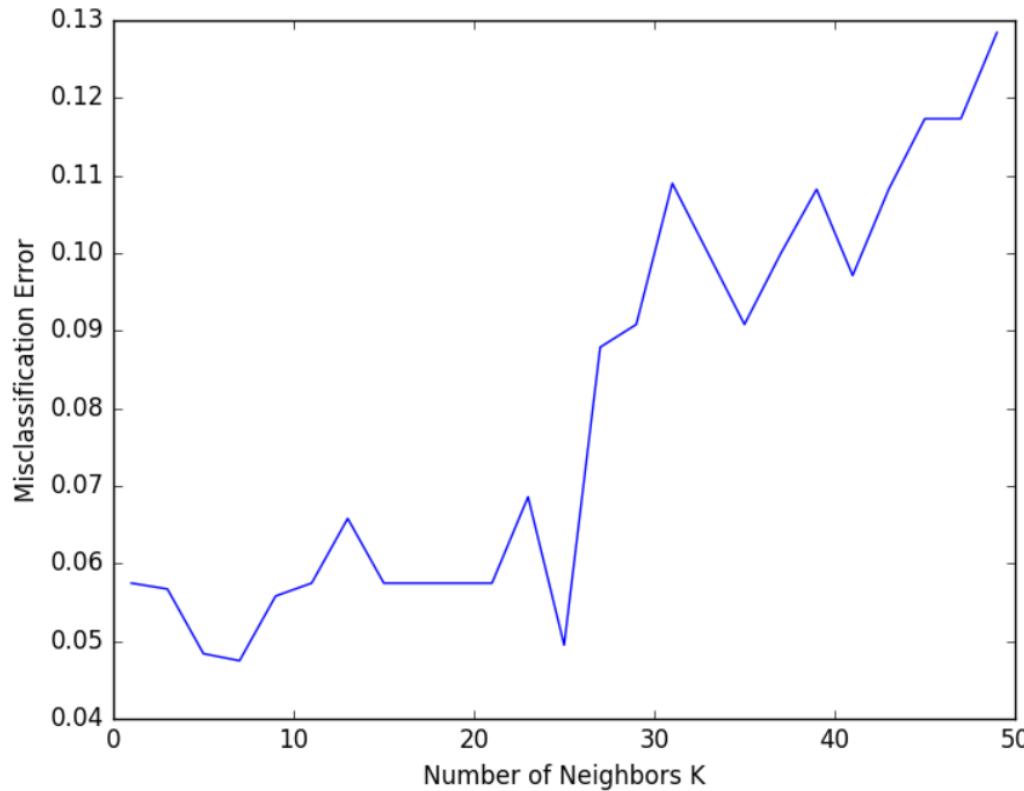
k-NN can be used for **classification** (majority vote) or **regression** (neighbour weighted average) problems



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

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)

Model Assessment

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

Metric	Formula	Interpretation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$	Coverage of actual negative sample
F1 score	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

k-Nearest Neighbours (*k*-NN)

Where a *k*-NN model may struggle:

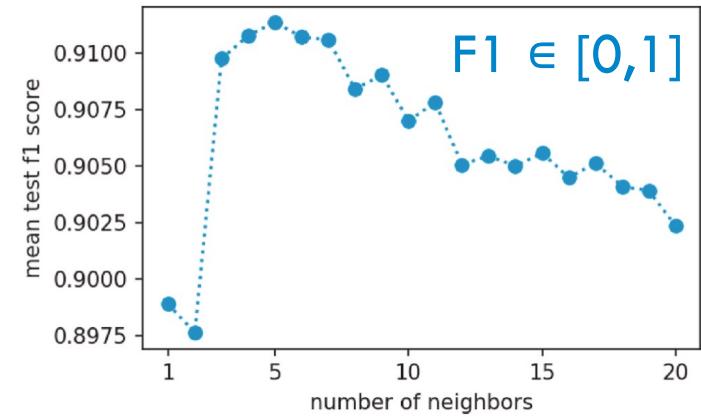
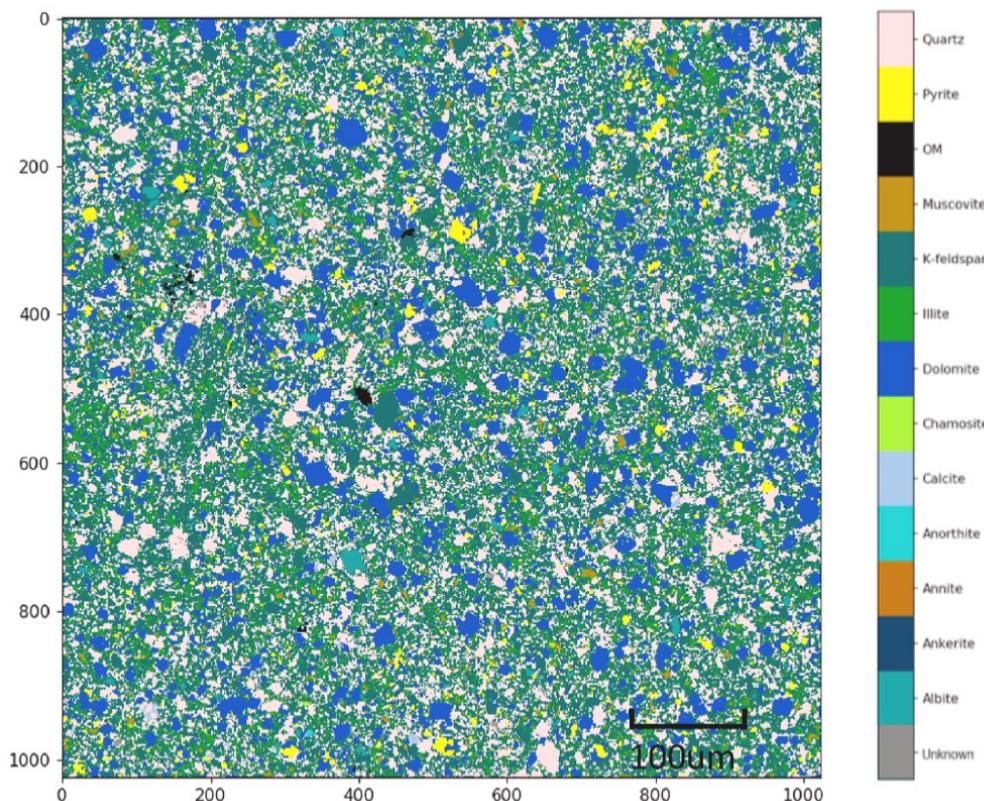
1. Imbalanced data – if there are multiple classes that differ in size, the smallest class may be overshadowed. Addressed by appropriate weighting

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

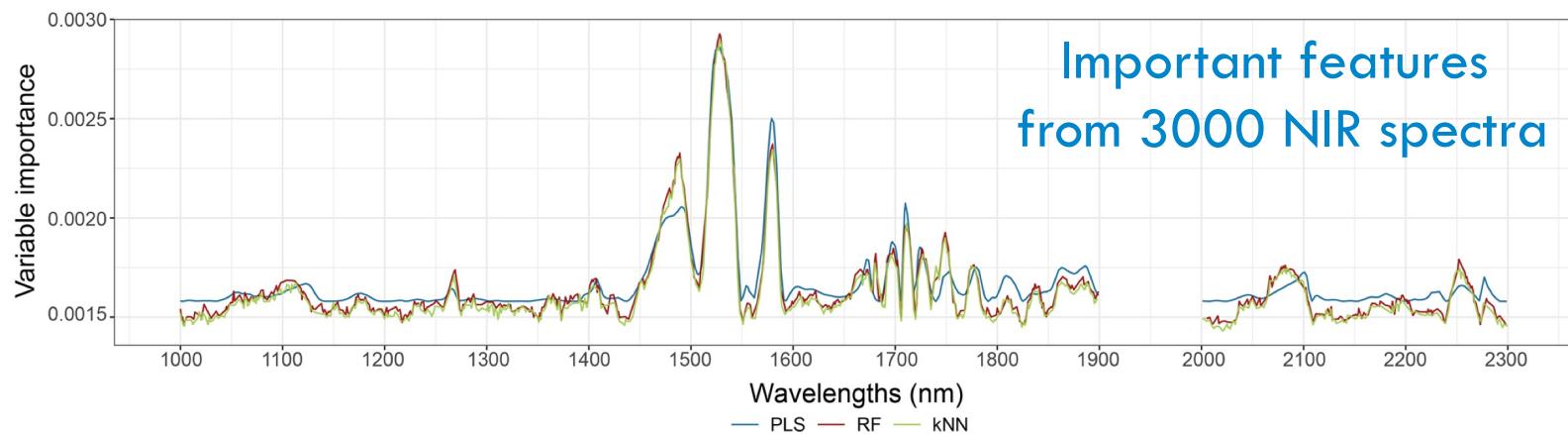
Classification of mixed mineral samples from microscopy (SEM-EDS) datasets



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

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	RF	<i>k</i> NN
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

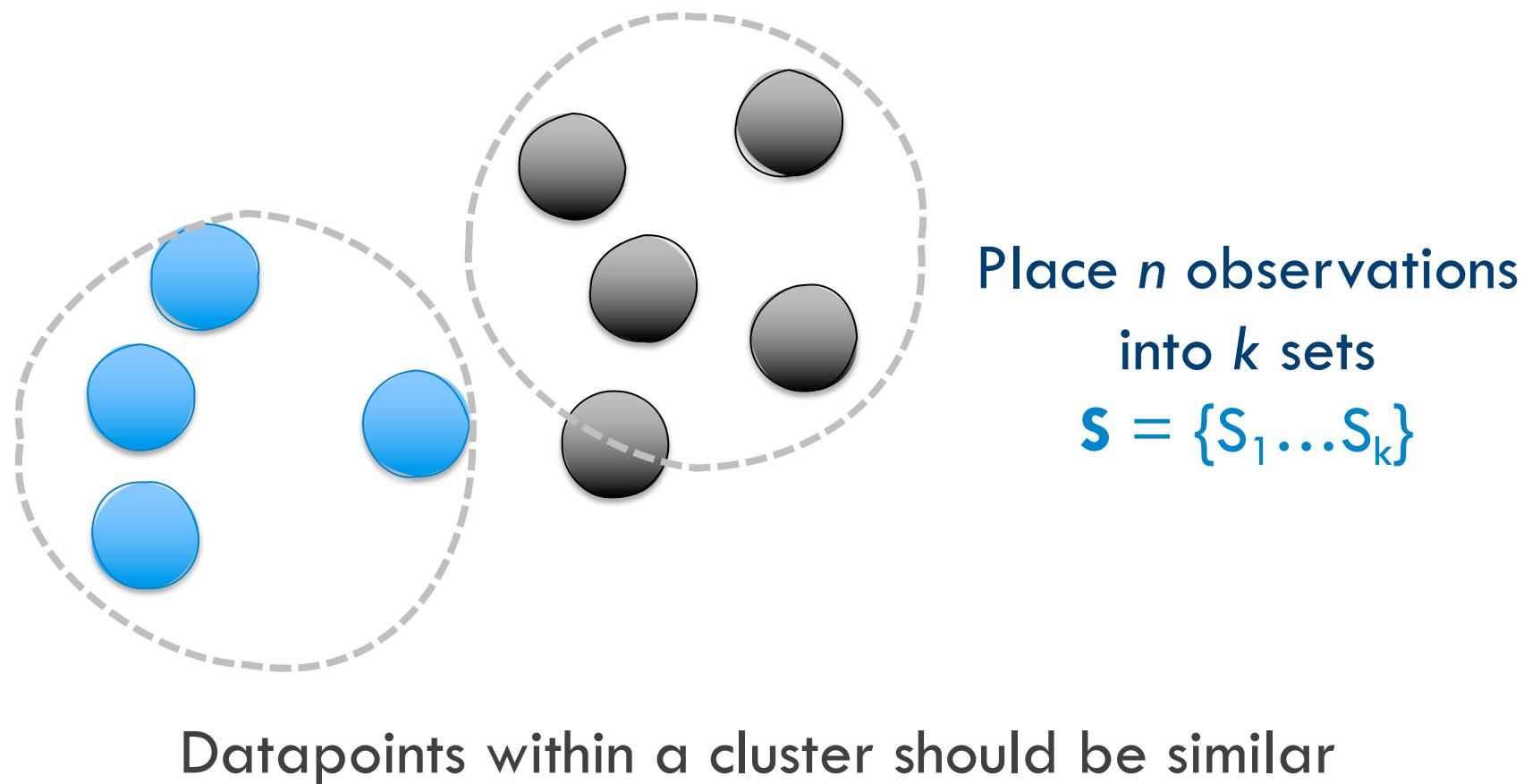
Class Outline

Classical Learning

- A. *k-nearest neighbours*
- B. *k-means clustering*
- C. *Decision trees and beyond*

k-Means Clustering

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



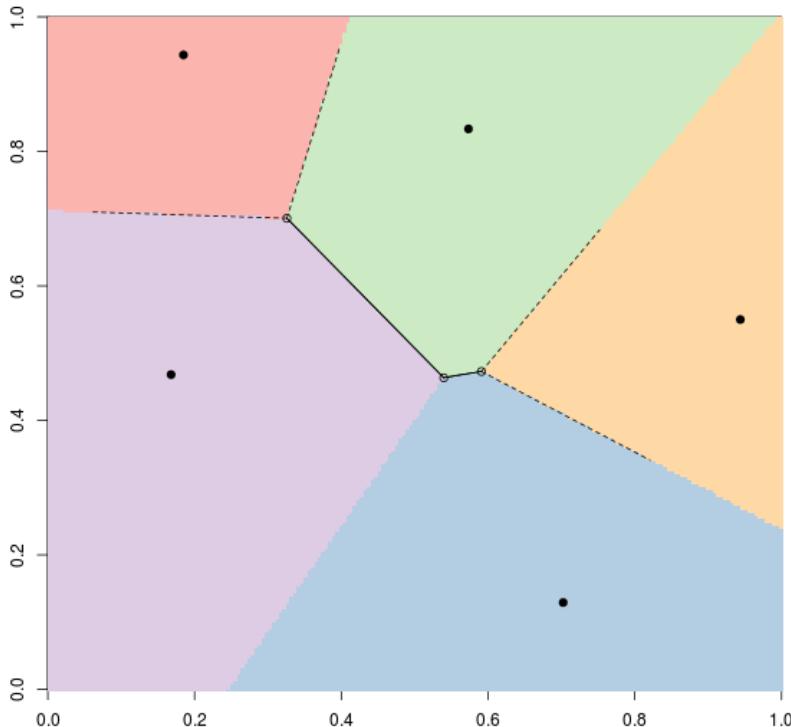
k-Means Clustering

Main components of a *k*-means model:

- 1. Initialisation:** Choose the number of clusters k that you want to identify in your dataset.
Centroids can be distributed randomly
- 2. Distance metric:** Similar to k -NN, you need a distance measure to define the similarity or dissimilarity, e.g. Euclidean or Manhattan
- 3. Assignment:** Each point is assigned to the nearest cluster based on distance and then iterated...

k-Means Clustering

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



Place n observations
into k sets

$$S = \{S_1 \dots S_k\}$$

Minimise within cluster
sum of squares (WSS)

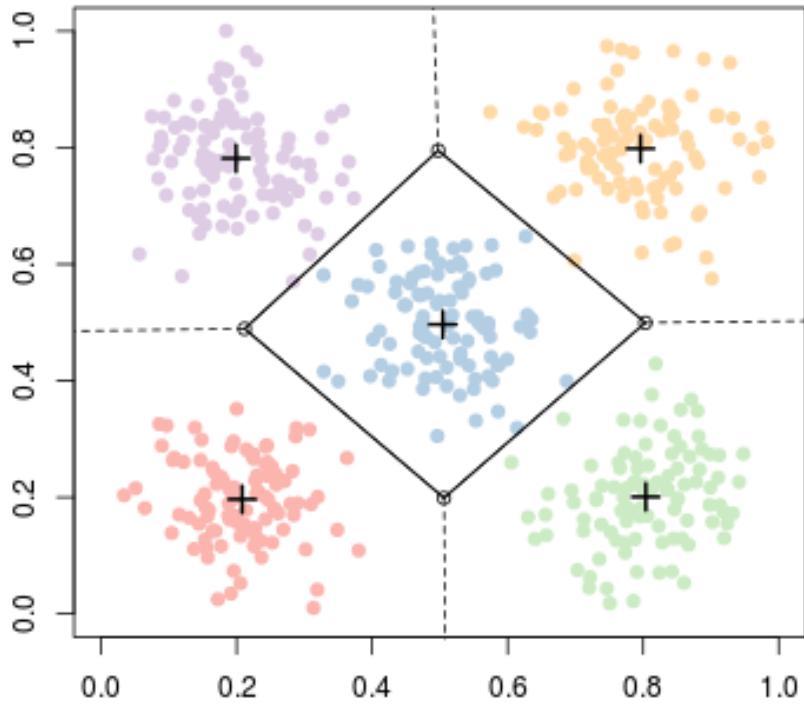
$$J = \sum |x_i - \mu_k|^2$$

centroid
of cluster k

An iterative algorithm is used to minimise cluster variance

k-Means Clustering

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



Place n observations
into k sets

$$S = \{S_1 \dots S_k\}$$

Minimise within cluster
sum of squares (WSS)

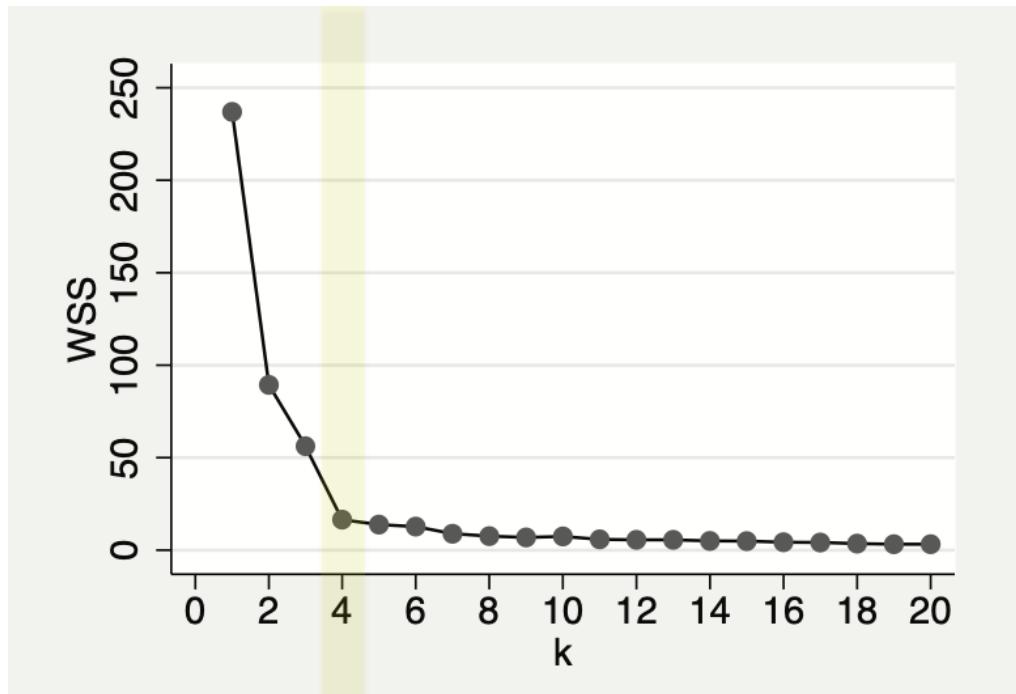
$$J = \sum |x_i - \mu_k|^2$$

centroid
of cluster k

An iterative algorithm is used to minimise cluster variance

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

k-Means Clustering

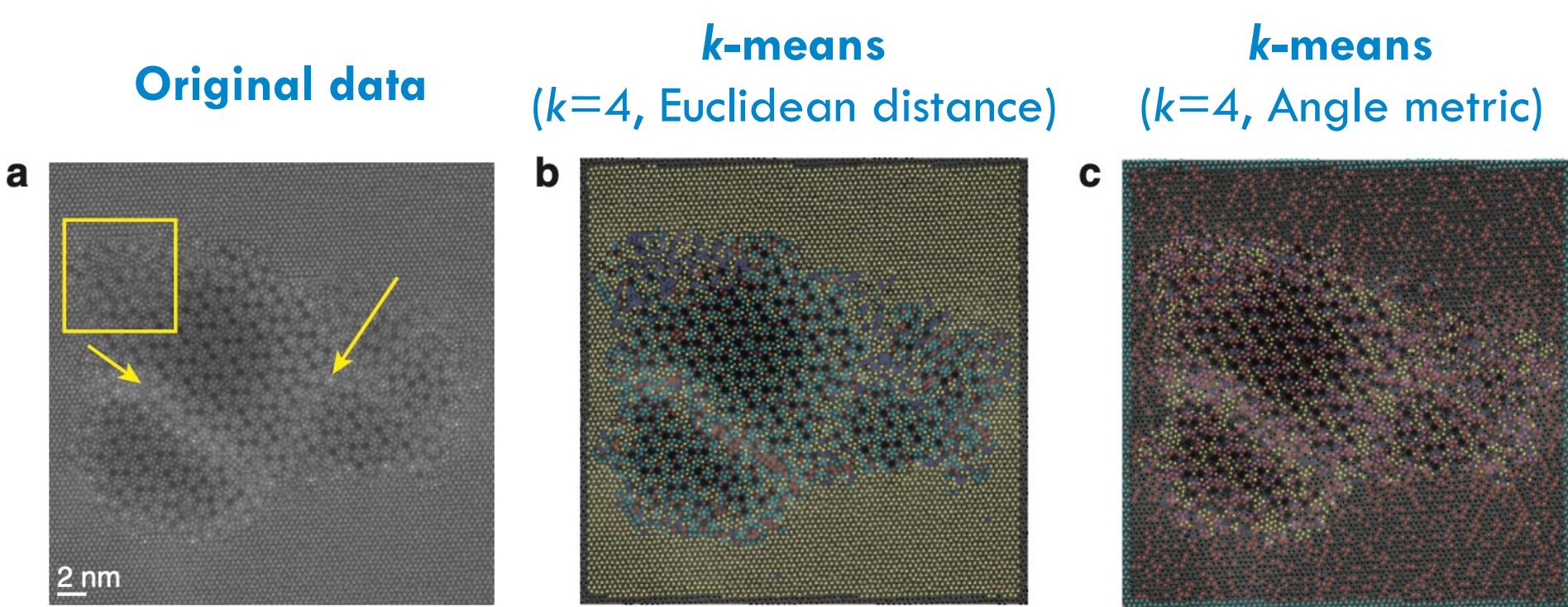
The strength of *k*-means is simplicity,
but it has limitations:

- 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



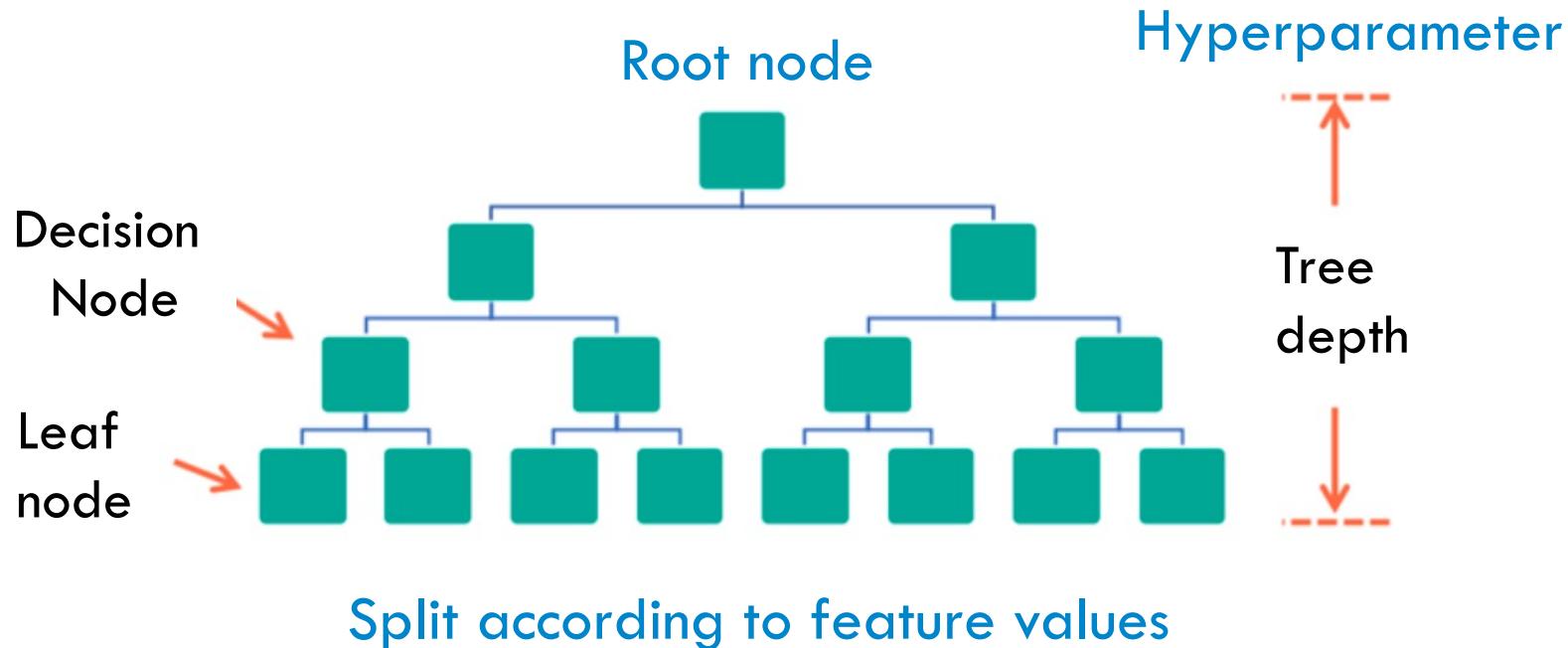
Class Outline

Classical Learning

- A. *k-nearest neighbours*
- B. *k-means clustering*
- C. ***Decision trees and beyond***

Decision Trees

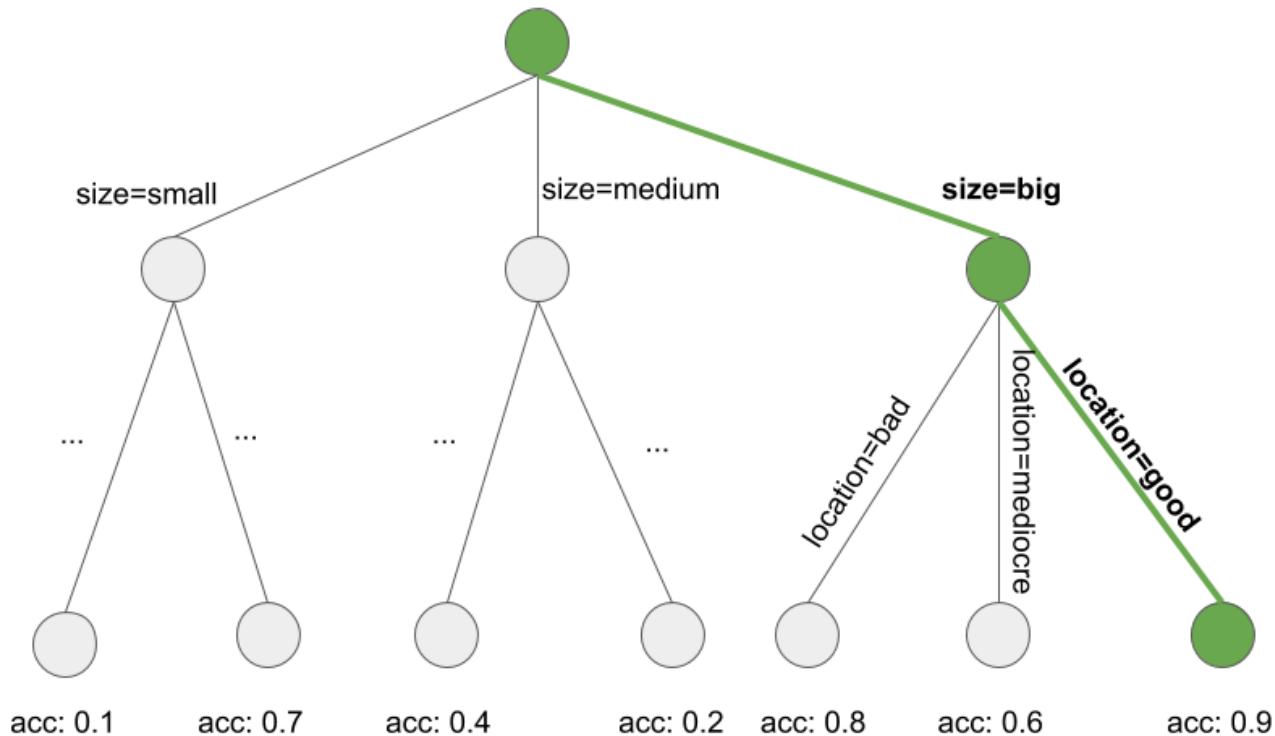
Supervised tree-like model splits data multiple times according to feature values (decision rules)



Can be used for classification or regression problems

Decision Trees

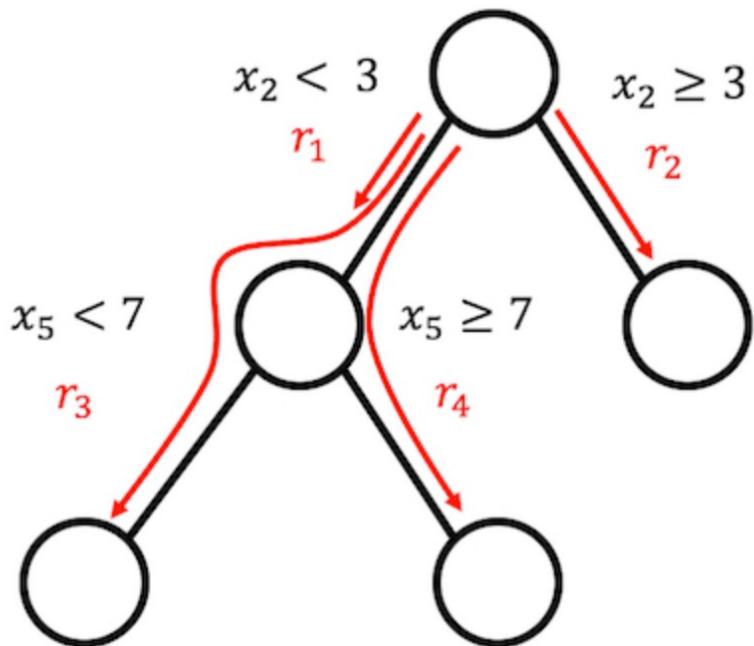
An interpretable model. Each prediction can be broken down into a sequence of decisions



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

Decision Trees

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



Coded as a sequence of rules



```
IF Feature 1 <= Threshold 1  
THEN Class Label A  
ELSE  
IF Feature 2 <= Threshold 2  
THEN Class Label B  
ELSE  
Class Label C
```

4 rules can be generated from this tree with 3 terminal nodes

Decision Trees

Main steps to build a decision tree model:

- 1. Feature selection:** Choose which features, from the labelled data, are most important for the tree
- 2. Splitting criteria:** Determine the best feature and test combination at each decision node,
e.g. based on information gain
- 3. Tree building:** An algorithm to recursively apply splitting criteria and expand to create child nodes until stopping criteria is met

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 many models through majority voting or averaging

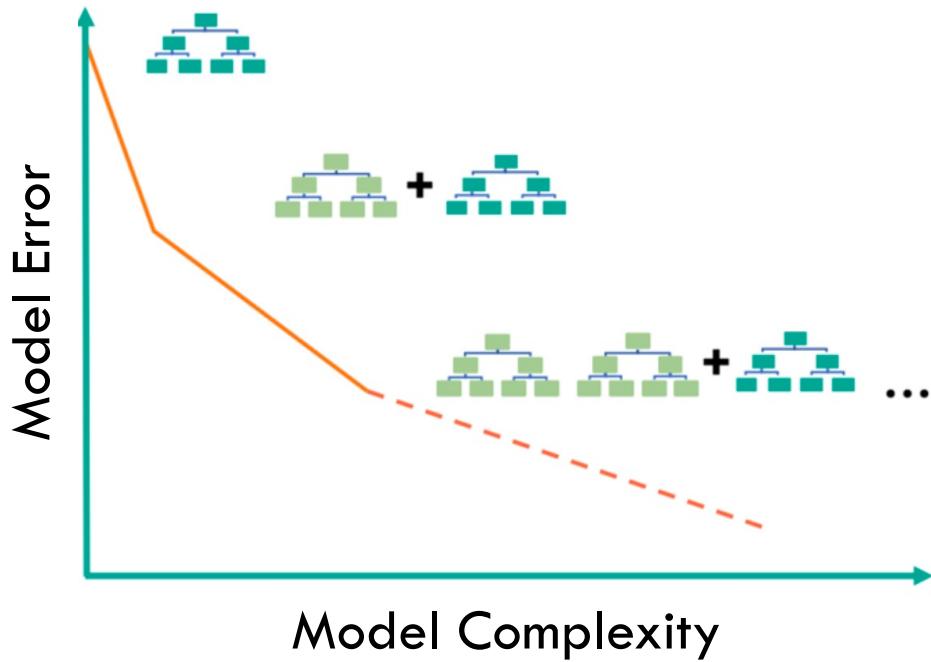
Model 1	Red	Red	Green	Green	Green	60% Accurate
Model 2	Green	Red	Red	Green	Red	40% Accurate
Model 3	Green	Red	Green	Red	Green	60% Accurate
Ensemble	Green	Red	Green	Green	Green	80% Accurate

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

Single decision trees can be combined for more powerful classification & regression models



Random Forests

Ensemble of independent decision trees

Gradient Boosted Regression

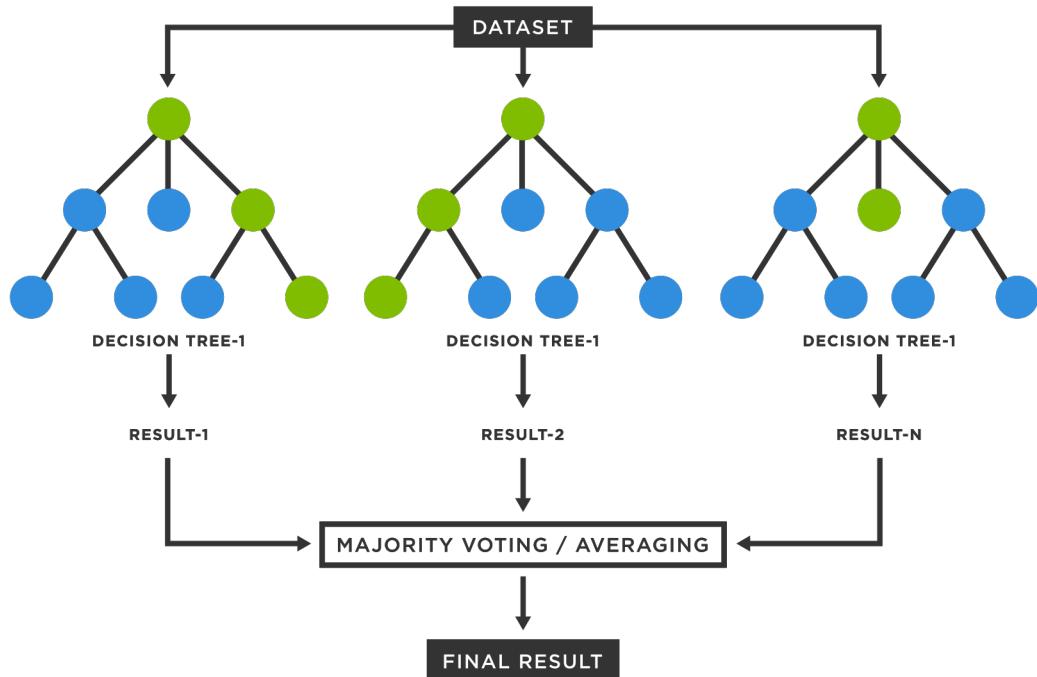
Ensemble of coupled decision trees

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

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

Gradient Boosted Regression (GBR)

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

$$y = \gamma_1 \text{tree}_1(x) + \gamma_2 \text{tree}_2(x) + \dots \gamma_n \text{tree}_n(x)$$

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 $n = 100$

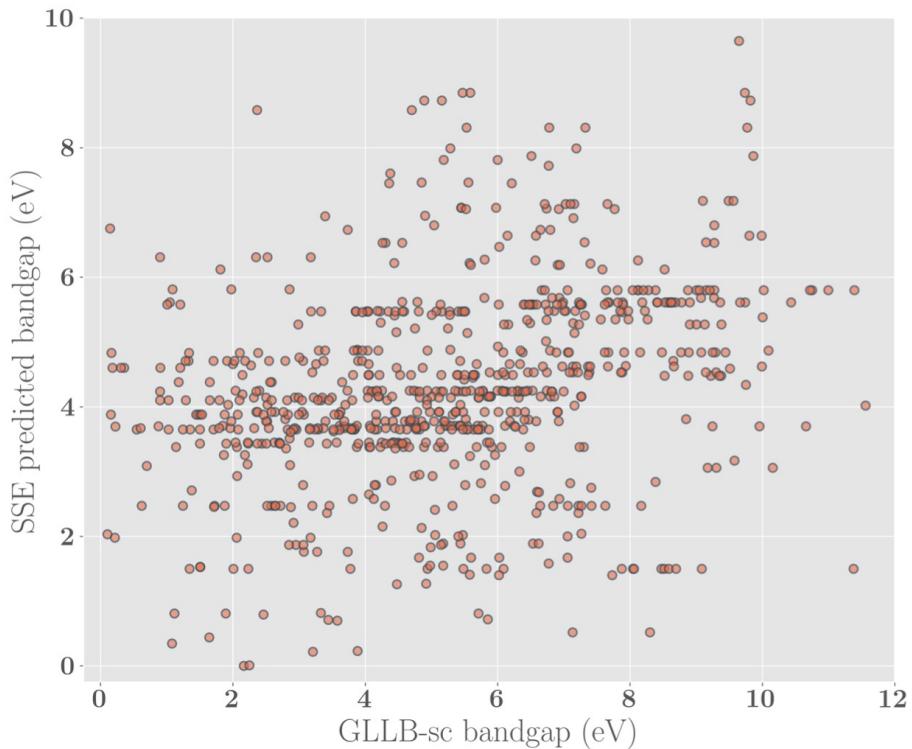
“When in doubt, use XGBoost”

Kaggle competition winner Owen Zhang

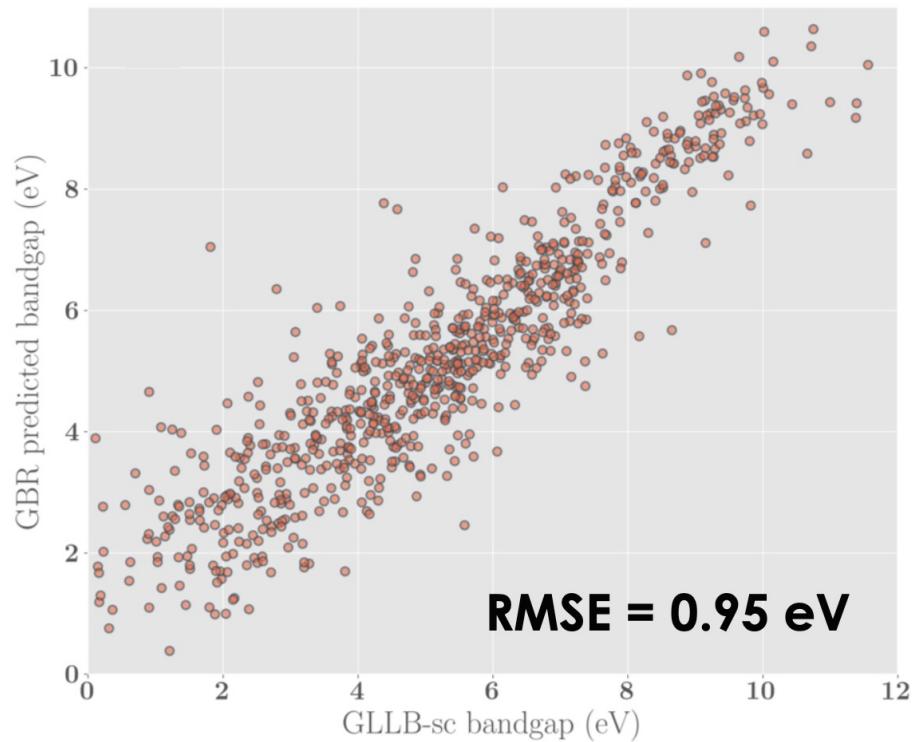
GBR Application: Band Gaps

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)

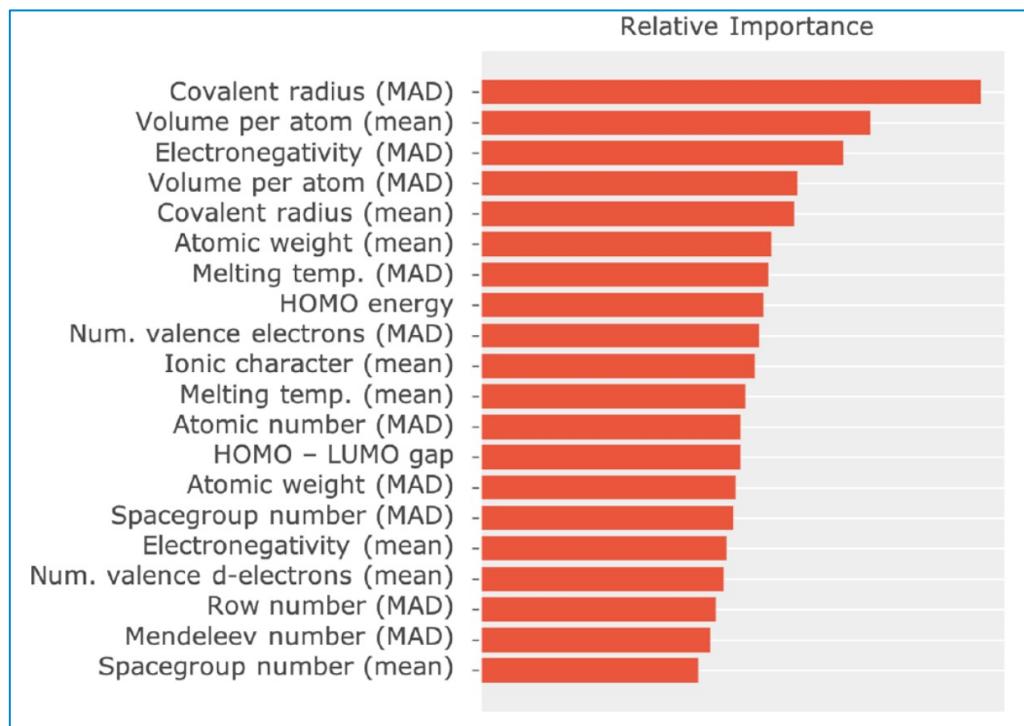
GBR Application: Band Gaps

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

Model hyperparameters

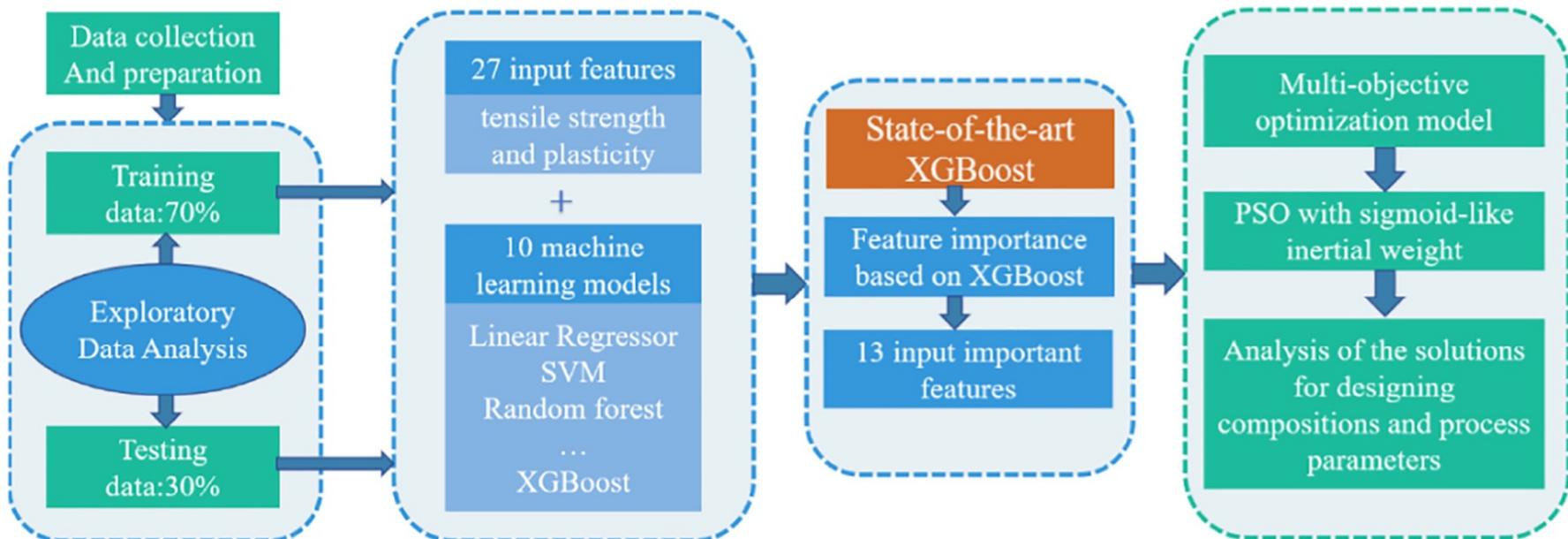
tree-specific parameters	
min. compounds needed to split nodes	65
max. depth of tree	20
min. compounds required at leaf nodes	1
max. features considered per tree	86
boosting parameters	
fraction of compounds to fit each tree	0.9
learning rate	0.01
number of decision trees	1000

20 most important features
(from 149 generated using Matminer)



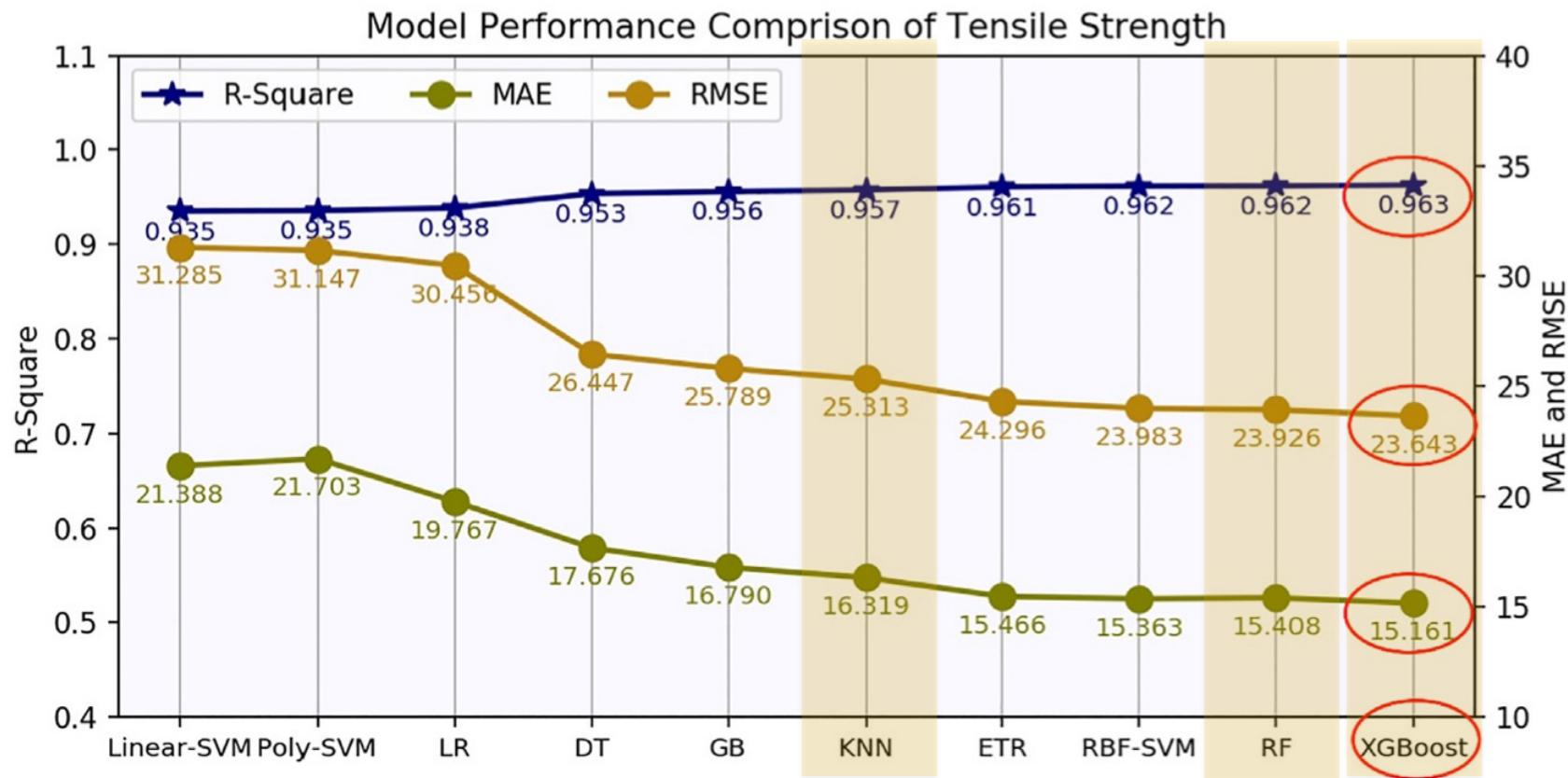
GBR Application: Steel

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



GBR Application: Steel

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



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:

Metals vs insulators