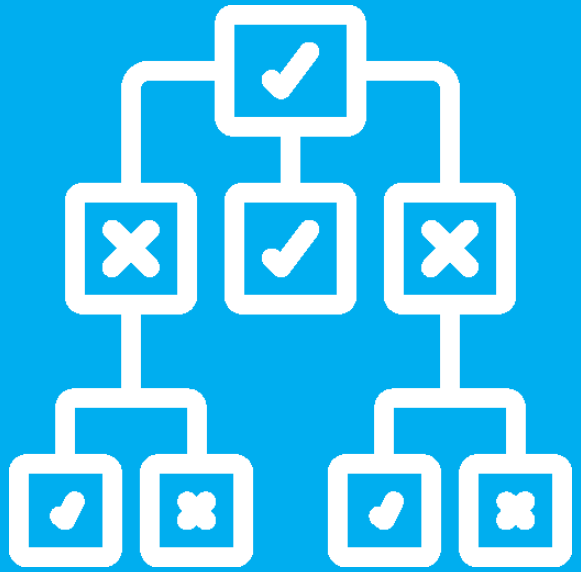
Warwick Business School

Data Science & Generative AI

Dr Michael Mortenson
Associate Professor (Reader)
michael.mortenson@wbs.ac.uk



Session 4:
ML Methodology II and Decision Trees

1.1 Data Engineering for Machine Learning

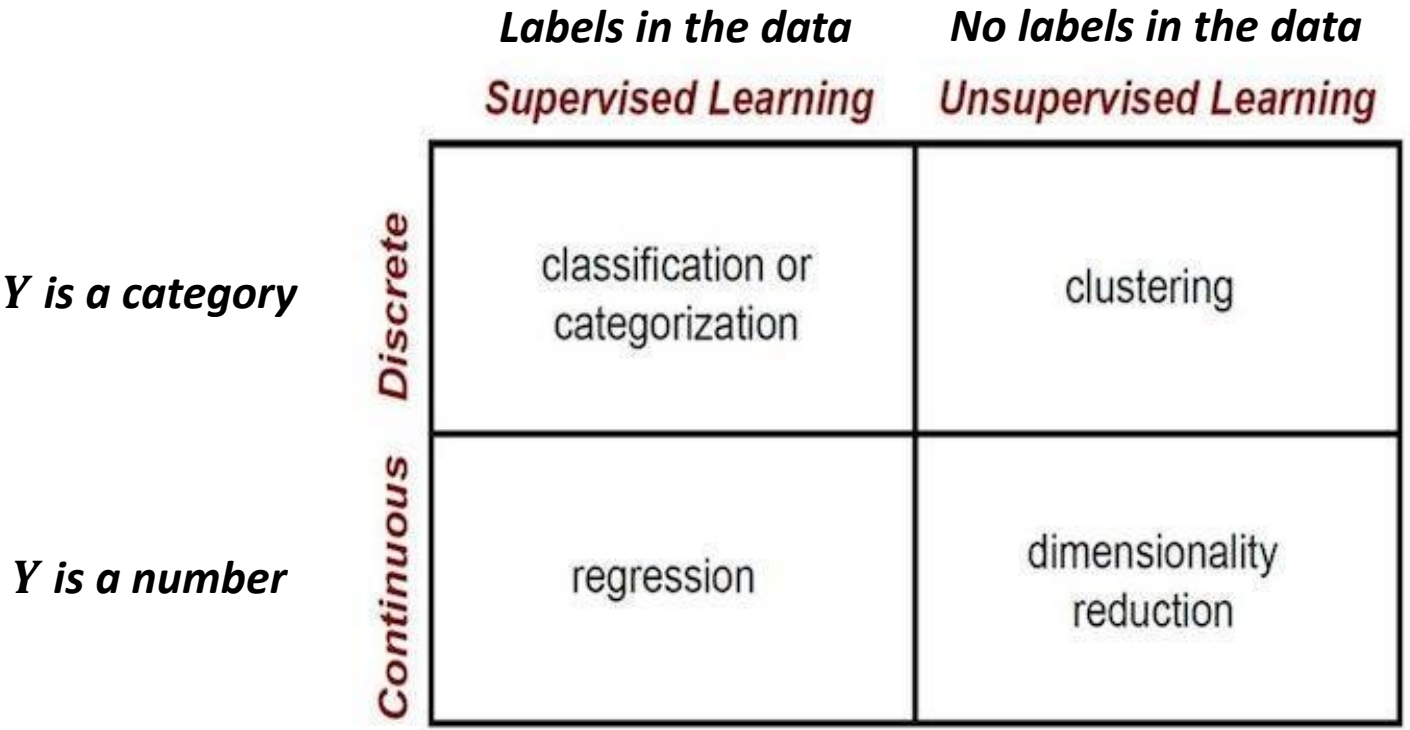


Image Credit: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

1.2 Scaling Data

- Typically we just scale all features regardless of model used:
  - Min-max scaling:

    $$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

    where x_i is an individual value (cell) in the feature (column) x.

  - Standardisation:

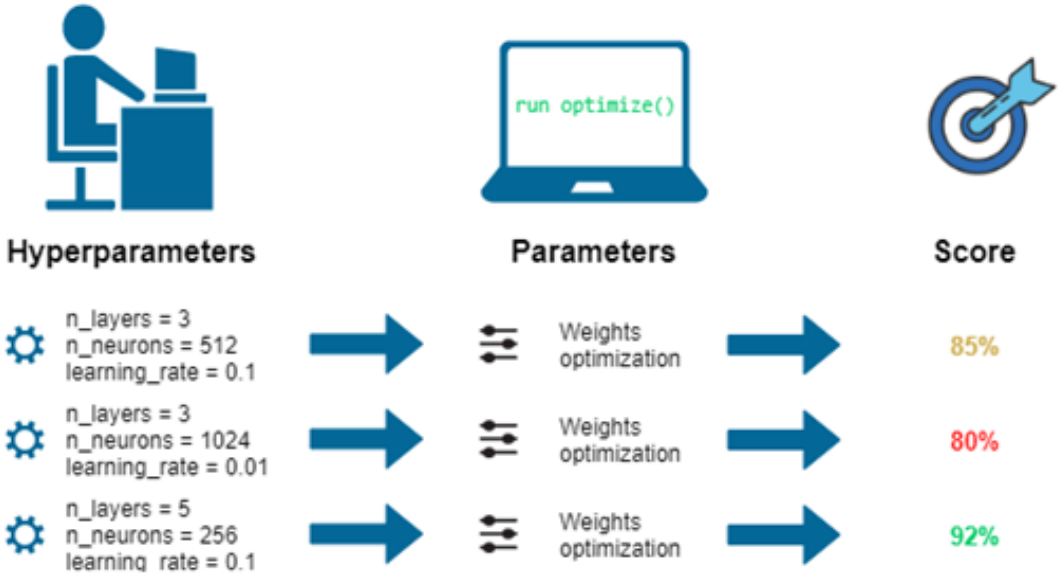
    $$z = \frac{x - \mu}{\sigma}$$

  Where x is a feature; $\max(x)$ is the maximum of x; $\min(x)$ its minimum; μ its mean; σ its standard deviation.

- These approaches will transform all the data to be on the same scale as each other, giving them all equal opportunity to influence the model.

1.3 Scores, Parameters and Hyperparameters

- **Score**: measures of the model's performance.
- **Parameters**: variables specific to the model that determine how it predicts. Parameters **are** the model.
- **Hyperparameters**: user defined parameters that govern the training process of the model. Mostly these balance under and over-fitting.



1.3 Scores, Parameters and Hyperparameters

- Choice of “score” for the model largely depends on task.
- For regression we would typically use:
  - Mean Squared Error: $MSE = \sum(y_i - \hat{y}_i)^2$
  - Mean Absolute Error: $MAE = \sum |y_i - \hat{y}_i|$
- These are very similar calculations to those we have already seen.
- For classification these will not work. If our labels are either 0 or 1 (e.g. “pass” or “fail”), our predictions are 0 or 1. It is not meaningful to measure the gap between label and prediction.
- Instead we may use something like accuracy:
  - $Accuracy = \frac{Correct\ Predicitons}{Total\ Predictions}$

1.3 Scores, Parameters and Hyperparameters

- Parameters are the things we learn when training an algorithm to become a model.
- In linear regression … e.g. $Y = \alpha + \beta x$:
  - Y and x are fixed … they are the data.
  - α and β are the parameters. When we learn a line we learn the slope of the line (β) and where it crosses the intercept (α) … the value of Y when x is zero.
- The parameters we learn are specific to the data (they are learned in relation to Y and x) and the choices we make such as regularisation methods.

1.3 Scores, Parameters and Hyperparameters

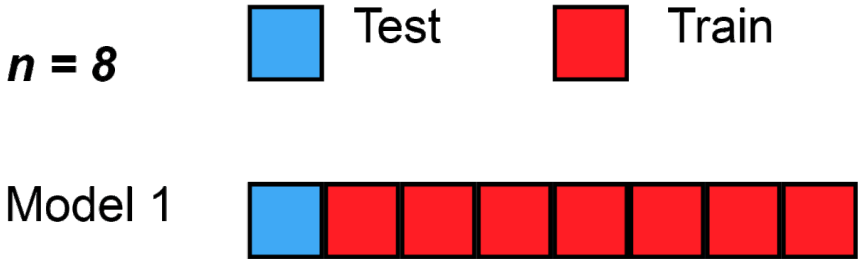
- Hyperparameters are the nobs and dials we can tweak to guide how the model is learned (normally managing bias vs variance).
- In L1 regularisation we modified the learning objective to add a penalty to the OLS objective … e.g. $OLS + \alpha \cdot \sum |\beta|$. The value of α determines how much we pay attention to the facilitator (line of best fit / least error) or the policeman (minimising the size of the β values). But how do we decide what α should be?
- Potentially we may have some theory on what this should be – if our data is very messy maybe a higher α. However, in ML we care more about results than theory, so normally we just solve this experimentally…

1.3 Scores, Parameters and Hyperparameters

- **Grid search:** We don't know which hyperparameters are best, so we can just search for them. If we have determined the best metric(s) to use (e.g. accuracy), we can compare different combinations of hyperparameters and simply choose the combination that returns the best result. E.g.
  - `criterion = ['gini', 'information gain']`
  - `max_depth = [None, 3, 5]`
  - `min_samples_split = [None, 5, 10]`
  - `max_features = [0.8, 'sqrt', 'log2']`
- This gives 2*3*3*3 = 54 permutations that we search to find the best combination.

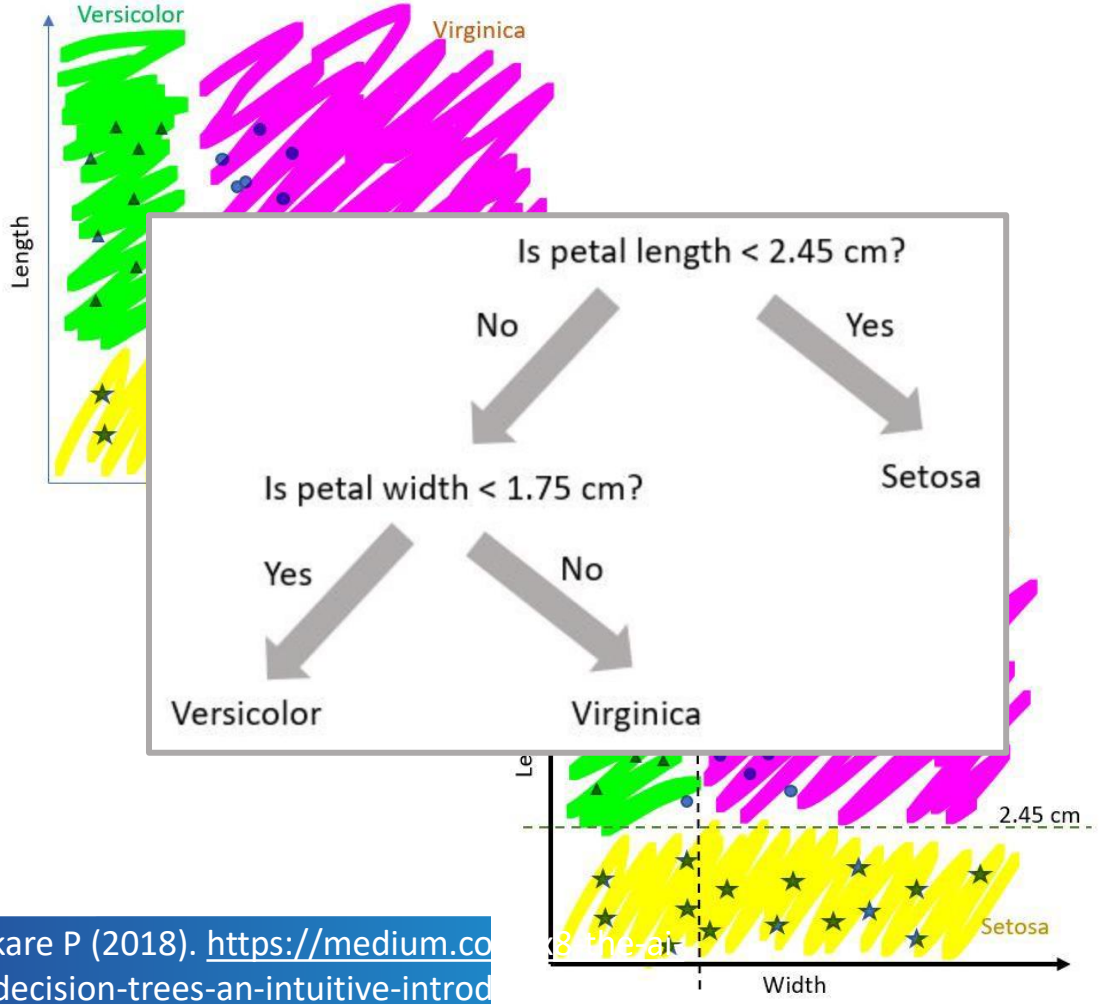
1.3 Scores, Parameters and Hyperparameters

- Split your training data (randomly) in to k separate chunks (e.g. 8);
- For each hyperparameter combination (chosen by grid or random):
  1. Allocate the 1st chunk as test, train on the remaining 7 and then test the model on the test set (1st chunk);
  2. Repeat step 1 seven times each time changing the test set (2nd chunk, 3rd chunk … k^{th} chunk);
  3. Average the results of the eight experiments to get the score for that hyperparameter combination.



1.4 Decision Trees by Eye

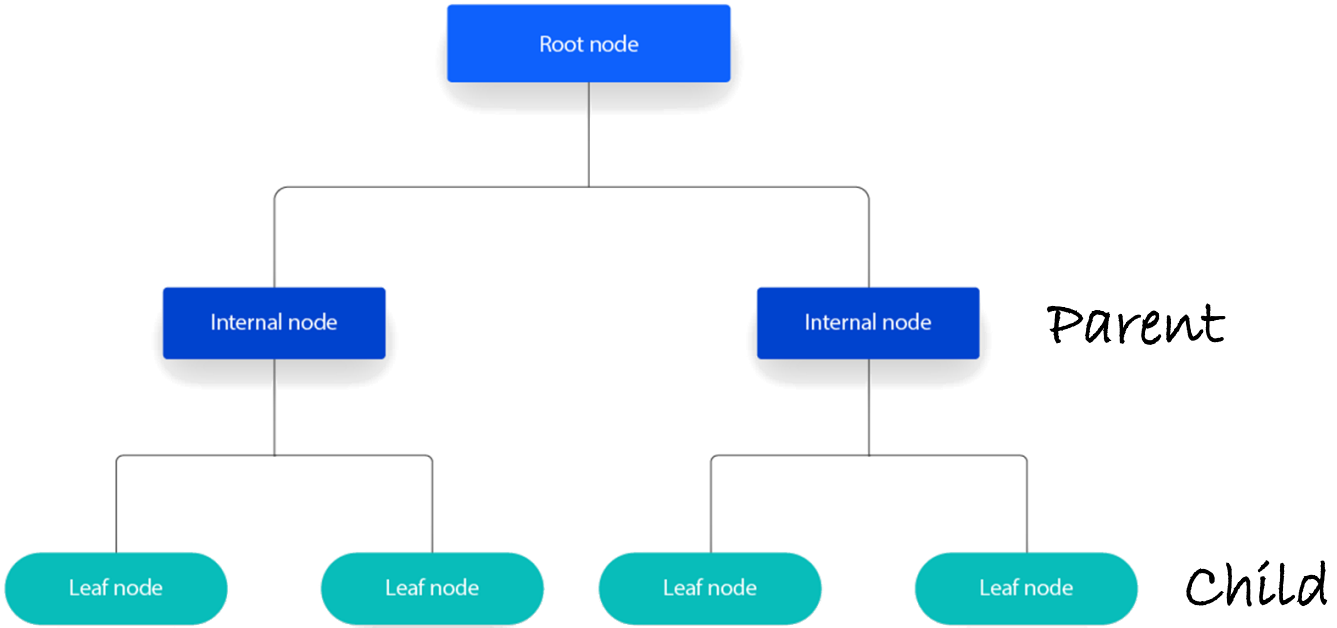
1. Plot the data onto a graph based on the most influential features
2. Identify a line which separates one or more class based on a specific value rule
3. Repeat (2) until all the classes are separated
4. Generalise the above to create an overall decision rule for the problem



Adapted from: Karkare P (2018). https://medium.co
community/decision-trees-an-intuitive-introd

1.5 Decision Trees by Algorithm

- Decision trees find a set of *if > else* conditions in which to split the data into classes.



Source: https://www.ibm.com/topics/decision-trees

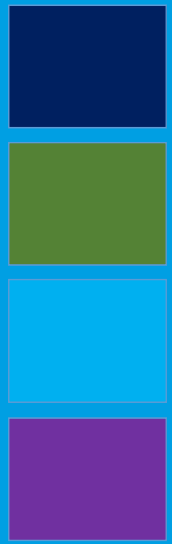
Session Aims

Introduction

Scores for Classification

Decision Trees Tutorial

Questions



2.1 Classification Metrics

- The most logical measure is accuracy:

$$accuracy = \frac{correct\_predictions}{total\_predictions}$$

- However, this only measures some of the error …

2.2 When Accuracy is Inaccurate (or at least misleading)

- The University of Warwick has developed a model to predict when students may (sadly) decide to drop out of their course.
- Based on 2024/25 data (not really I made all of this up), 9 out of 10 students complete their studies.
- We have developed a model and in training it shows an accuracy of 88%.
- Is the model any good?

2.3 Precision, Recall and F1

- **Precision** (when I predict 1, how often am I right):

$$precision = \frac{TP}{TP+FP}$$

- **Recall** (how many of the 1's do I find):

$$recall = \frac{TP}{TP+FN}$$

- **F1 Score** (bit of both):

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

		Predicted 0	Predicted 1
Actual	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

2.3 Precision, Recall and F1

- **Precision** (when I predict 1, how often am I right):

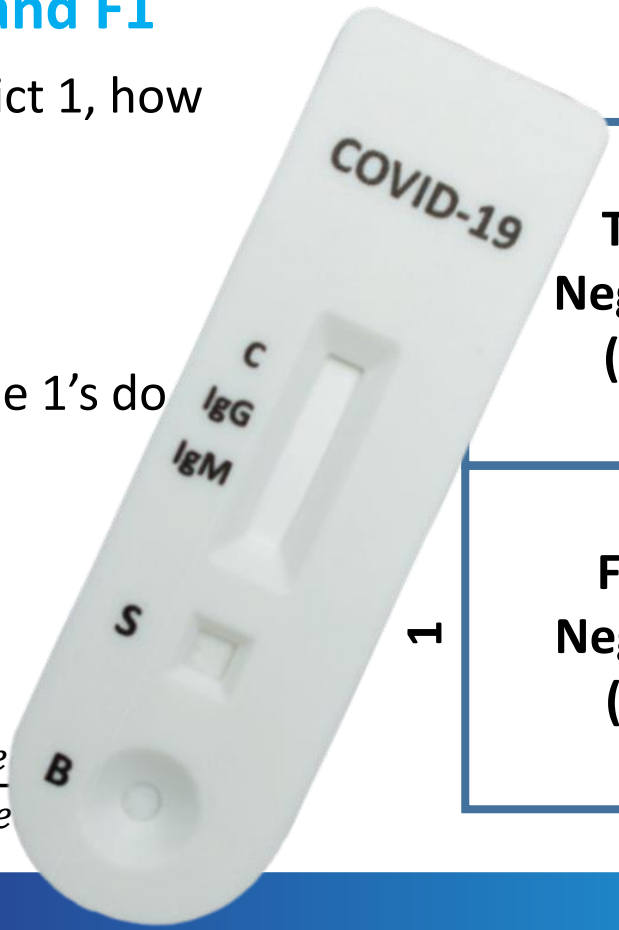
$$precision = \frac{TP}{TP+FP}$$

- **Recall** (how many of the 1's do find):

$$recall = \frac{TP}{TP+FN}$$

- **F1 Score** (bit of both):

$$F1 = 2 \cdot \frac{precision \cdot re}{precision + re}$$



		Predicted 0	Predicted 1
	0	True Negative (TN)	False Positive (FP)
	1	False Negative (FN)	True Positive (TP)

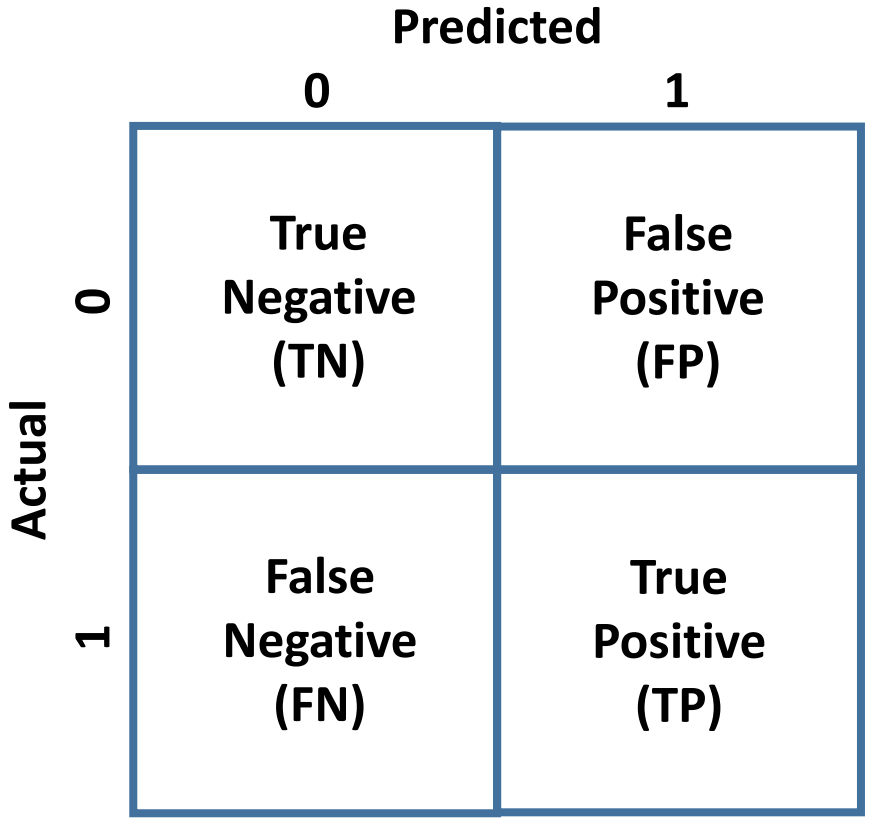
2.3 Precision, Recall and F1

- **Macro averaging**:

$$precision = \frac{precision_0 + precision_1}{2}$$

- **Micro averaging:**

$$precision = \frac{TP_0 + TP_1}{TP_0 + TP_1 + FP_0 + FP_1}$$



Session Aims

Introduction

Scores for Classification

Decision Trees Tutorial

Questions