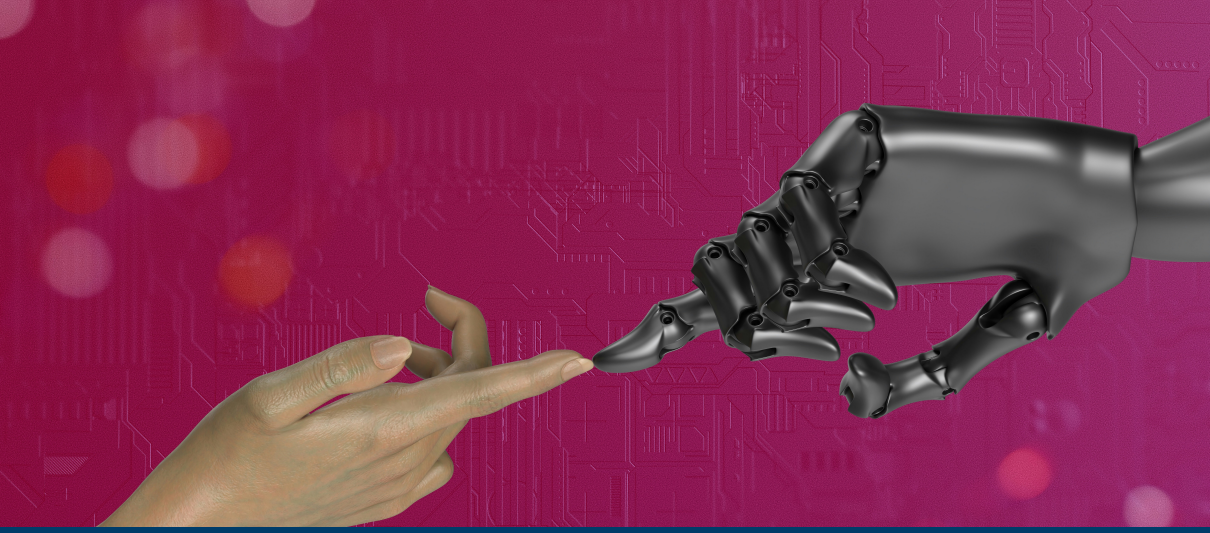


Hands-On Machine Learning

Dr. Derek Bridge | 2026/2027

UCC Coláiste na hOllscoile Corcaigh
University College Cork, Ireland

School of Computer Science and Information Technology
Scoil na Ríomheolaíochta agus na Teicneolaíochta Faisnéise

3. Binary Classification

The data we use here was collected by me when I was teaching a module (CS1109) between 2008 and 2014.

Classification

- **Classifiers** predict an object's **class** from a finite set of classes.
- E.g. given an email, predict whether the email is spam or ham.
- E.g. given a photo, predict whether it depicts a cat, dog or rabbit.
- (Compare with **regression**.)

Binary Classification

- In **binary classification**, there are just two classes.
- E.g. fail/pass, ham/spam, benign/malignant.
- (Contrast with **multiclass classification**.)

Positive/Negative

- In binary classification, it is common to refer to one class as the **negative class** and the other the **positive class**.
- It doesn't really matter which is which. But, usually, we treat the class that requires special action as the positive class.
- E.g. in spam filtering, ham is the negative class; spam is the positive class.
- What about tumour classification?
- This terminology is extended to other things too, e.g. we can refer to **negative examples** and **positive examples**.

Class exercise

- Consider:
  - Predicting tomorrow's rainfall.
  - Predicting whether we will have a white Christmas.
  - Predicting the sentiment of a tweet (negative, neutral or positive).
  - Predicting a person's sexual orientation.
  - Predicting a person's opinion of a movie on a rating scale of 1 star (rotten) to 5 stars (fab).
- Answer the following:
  - Which are regression and which classification?
  - If classification, which are binary and which are multiclass?
  - If binary, which is the positive class and which the negative?

Classes and Class Labels

- We assume we have a finite set of **labels**, $\mathcal{C}$, one per class.
- Given an object $\boldsymbol{x}$, a classifier will assign one of the labels $\hat{y} \in \mathcal{C}$ to the object.
- Software libraries often use integers for the labels (**label encoding**).
- E.g. given an email, a spam filter predicts $\hat{y} \in \{0, 1\}$, where 0 means ham and 1 means spam.
- But a classifier should not treat these as continuous, e.g. it should never output 0.5.
- Furthermore, where there are more than two labels, we should not assume a relationship between the labels.

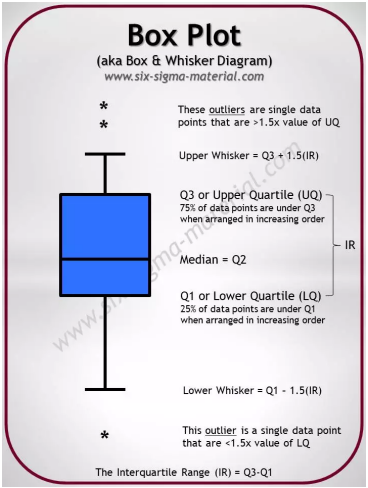
Question

- Suppose there are three classes $\{1, 2, 3\}$.
- Suppose we are classifying object $\boldsymbol{x}$ and we happen to know that its actual class label $y = 3$.
- One classifier predicts $\hat{y} = 1$.
- Another classifier predicts $\hat{y} = 2$.
- Which classifier has done better?

Stratified Holdout

- Splitting a dataset using holdout might result in partitions that do not reflect the distribution of examples within the classes:
  - Examples of one class might be under-represented in the training set or test set.
  - Examples of one class might even be completely absent from the training set or test set.
- **Stratified holdout**: the proportion of examples of each class in the overall dataset is respected in the partitioning into training and test sets.
- Although this fixes the distribution with respect to the classes, you may still get 'lucky' or 'unlucky' in other ways. So we will still want a large dataset for holdout.

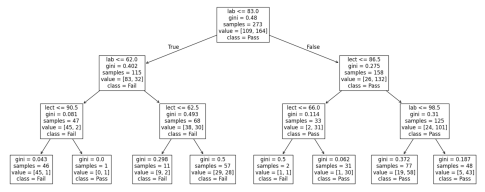
How to Read a Box-and-Whisker Plot



Decision Trees for Classification

A **Decision Tree** is a model, comprising a hierarchy of simple rules.

Inference: start at the root and keep going left or right, as appropriate.



- `gini`: is the **Gini impurity** of a node (with values in $[0.0, 0.5]$).
- `samples`: how many training examples the node applies to.
- `value`: how many training examples of each class this node applies to.
- `class`: the prediction if we stop at this node (the majority class of the training examples that this node applies to)

Gini Impurity

Assume binary classification for simplicity: classes $\mathcal{C} = \{c, c'\}$.

Then at a particular node in the tree,

$$Gini = 1 - (Prob(c)^2 + Prob(c')^2)$$

- Gini measures how often an example would be incorrectly labeled if it was randomly labeled according to the distribution of labels for this node.
- A Gini of 0 means no impurity: all examples that this node applies to belong to the same class.
- Values closer to 0.5 imply a lot more impurity.
- (If you're Googling this, make sure you're reading about Gini Impurity or the Gini index, not the Gini coefficient!).

Learning a Decision Tree

The CART Algorithm (Classification and Regression Tree)

- CART can learn trees for classification and for regression.
- For regression, it uses MSE in place of Gini.
- For classification, it can use entropy in place of Gini — but it doesn't usually make much difference!
- In the case of classification, it can learn trees for binary classification ($|\mathcal{C}| = 2$) and multiclass classification ($|\mathcal{C}| > 2$).
- It only produces binary trees (hence yes/no questions in non-leaf nodes).
- It can handle both numeric-valued and nominal-valued features and even missing values — although this may not be true of the scikit-learn implementation.
- It is recursive.
- It is greedy.

The CART Algorithm

for *each feature* x_i *and each value* v, **do**
split the dataset into two;

X_{left} are the examples in X for which $x_i \leq v$;

X_{right} are the examples in X for which $x_i > v$;

Then calculate the CART score: $\frac{|X_{left}|}{|X|}\text{Gini}(X_{left}) + \frac{|X_{right}|}{|X|}\text{Gini}(X_{right})$;

end

From the above, choose the feature x_i and value v with lowest score;

if *a stopping criterion has been reached (e.g. maximum depth or if no split reduces impurity)* **then**
return;

end

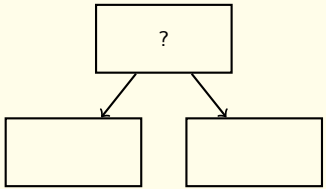
else
Recursively call CART on X_{left};

Recursively call CART on X_{right};

end

Example

What will go in the root?



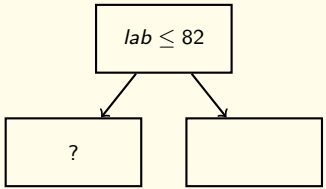
For each feature and each value, we compute the CART score.

$lect \leq 1$	0.477
$lect \leq 2$	0.472
$lect \leq 3$	0.469
$lect \leq 4$	0.469
$lect \leq 5$	0.466
⋮	⋮
$lab \leq 1$	0.472
$lab \leq 2$	0.472
$lab \leq 3$	0.472
$lab \leq 4$	0.472
$lab \leq 5$	0.469
⋮	⋮

The lowest CART score is $lab \leq 82 : 0.328$.

Example

What will go in the left-hand child?



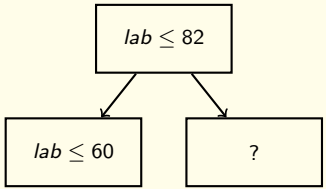
For each feature and each value, we compute the CART score.

$lect \leq 1$	0.4
$lect \leq 2$	0.398
$lect \leq 3$	0.396
$lect \leq 4$	0.396
$lect \leq 5$	0.395
⋮	⋮
$lab \leq 1$	0.398
$lab \leq 2$	0.398
$lab \leq 3$	0.398
$lab \leq 4$	0.398
$lab \leq 5$	0.396
⋮	⋮

The lowest CART score is $lab \leq 60 : 0.325$.

Example

What will go in the right-hand child?



For each feature and each value, we compute the CART score.

$lect \leq 57$	0.275
$lect \leq 58$	0.275
$lect \leq 59$	0.275
$lect \leq 60$	0.275
$lect \leq 61$	0.275
⋮	⋮
$lab \leq 84$	0.275
$lab \leq 85$	0.275
$lab \leq 86$	0.275
$lab \leq 87$	0.275
$lab \leq 88$	0.275
⋮	⋮

The lowest CART score is $lect \leq 84 : 0.275$.

A Variation of Classification

- Given an object $\boldsymbol{x}$, a classifier outputs a label, $\hat{y} \in \mathcal{C}$.
- Instead, a classifier could output a probability distribution over the labels $\mathcal{C}$.
- E.g. given an email $\boldsymbol{x}$, a spam filter might output $\langle 0.2, 0.8 \rangle$ meaning the probability that $\boldsymbol{x}$ is *ham* is 0.2, and the probability that it is *spam* is 0.8.
- The probabilities must sum to 1.

Questions

- In fact, a binary classifier can output just one probability, rather than two. Why?
- How do you think a Decision Tree calculates these probabilities?

Error Estimation for Classifiers

The main evaluation metric is **accuracy** (ACC).

(If you want an error estimate, then compute $1 - ACC$.)

Accuracy (ACC)

$ACC = \frac{1}{m} \sum_{\boldsymbol{x} \in \boldsymbol{X}, y \in \boldsymbol{y}} int(\hat{y} = y)$

Alternatives to Accuracy (for Private Study) — Binary Classification

Confusion Matrix

		Predicted Class: Negative	Predicted Class: Positive
Actual Class	**Negative**	True Negatives (TN)	False Positives (FP)
	Positive	False Negatives (FN)	True Positives (TP)

Alternatives to Accuracy (for Private Study) — Binary Classification

TNR, FPR, FNR, TPR, NPV, FDR, FOR, PPV

		Predicted Class: Negative	Predicted Class: Positive		
Actual Class	**Negative**	True Negatives (TN)	False Positives (FP)	*True Negative Rate* $TNR = \frac{TN}{TN+FP}$	*False Positive Rate* $FPR = \frac{FP}{TN+FP}$
	Positive	False Negatives (FN)	True Positives (TP)	*False Negative Rate* $FNR = \frac{FN}{FN+TP}$	*True Positive Rate* $TPR = \frac{TP}{FN+TP}$

Negative Predicted Value $NPV = \frac{TN}{TN+FN}$

False Discovery Rate $FDR = \frac{FP}{FP+TP}$

False Omission Rate $FOR = \frac{FN}{TN+FN}$

Positive Prediction Value $PPV = \frac{TP}{FP+TP}$

Alternatives to Accuracy (for Private Study) — Binary Classification

		Predicted Class: Negative	Predicted Class: Positive
Actual Class	Negative	True Negatives (TN)	False Positives (FP)
	Positive	False Negatives (FN)	True Positives (TP)

Accuracy

$$ACC = \frac{TP + TN}{FP + FN + TP + TN}$$

Precision and Recall

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

$$Recall = \frac{TP}{FN + TP} = TPR$$

F1 score

$$F1 = 2\frac{Prec \times Rec}{Prec + Rec}$$

Alternatives to Accuracy (for Private Study) — Multiclass Classification

Suppose $|\mathcal{C}| > 2$. Take each class $c \in \mathcal{C}$. Treat c as the positive class and all other classes as the negative class. Then we can calculate TP_c — the True Positive for class c, and similarly TN_c, FP_c, FN_c.

Micro-Averaged Precision

$$PRE_{micro} = \frac{\sum_{c \in \mathcal{C}} TP_c}{\sum_{c \in \mathcal{C}} FP_c + \sum_{c \in \mathcal{C}} TP_c}$$

Macro-Averaged Precision

$$PRE_{macro} = \frac{\sum_{c \in \mathcal{C}} PRE_c}{|\mathcal{C}|}$$

Hyperparameter

The maximum depth of a Decision Tree is an example of a **hyperparameter** — an input to the training algorithm.

Underfitting and Overfitting

- **Underfitting**: the model is not complex enough.
- **Overfitting**: the model is too complex.

The maximum depth is one factor influencing whether a Decision Tree underfits or overfits.

Image credits

	Photo from myirelandtour.com
	Taken from https://www.six-sigma-material.com