A (short) introduction to Machine Learning (ML) test

Dr Mick Taylor
University of Sussex, UK

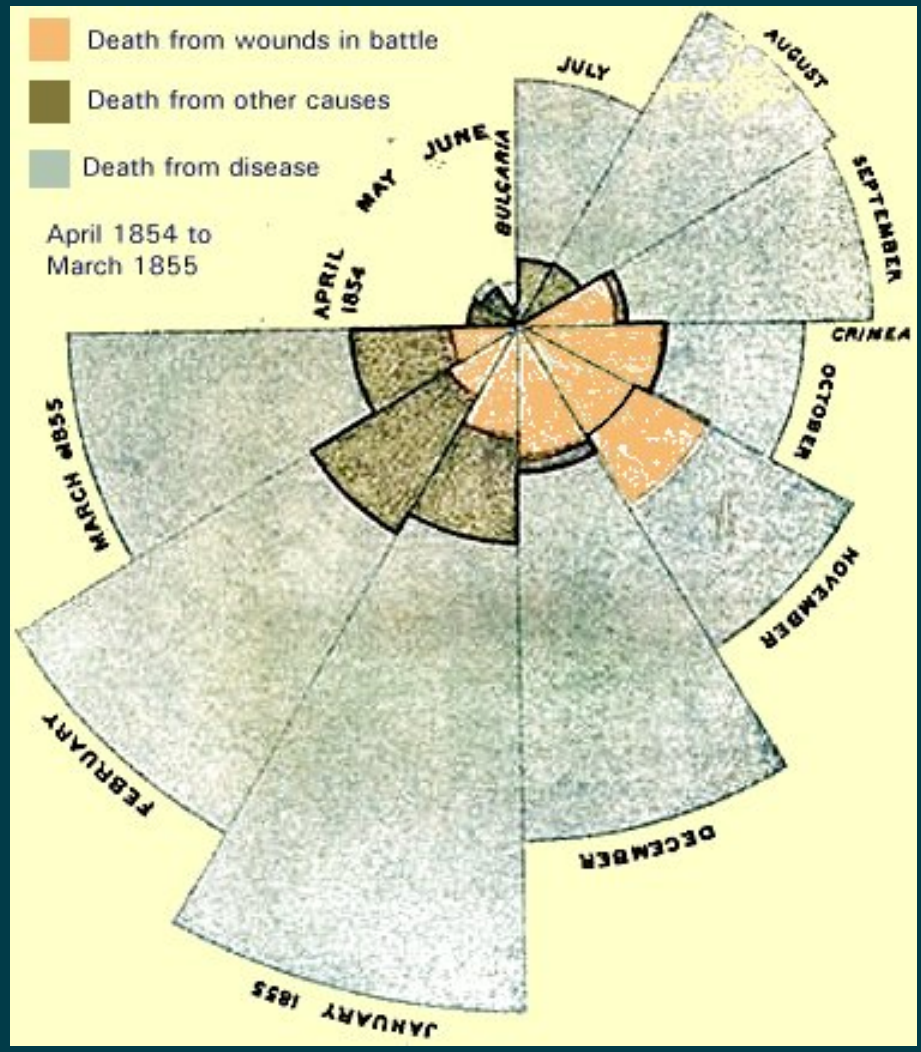
The Abdus Salam International Centre for Theoretical Physics
Physics Without Frontiers
IAEA International Atomic Energy Agency
unesco United Nations Educational, Scientific and Cultural Organization
UNIVERSITY OF SUSSEX

What is Machine Learning (ML)?

?

What is Machine Learning (ML)?

Florence Nightingale, 1820 - 1910



What is Machine Learning (ML)?

Computer model

What is Machine Learning (ML)?

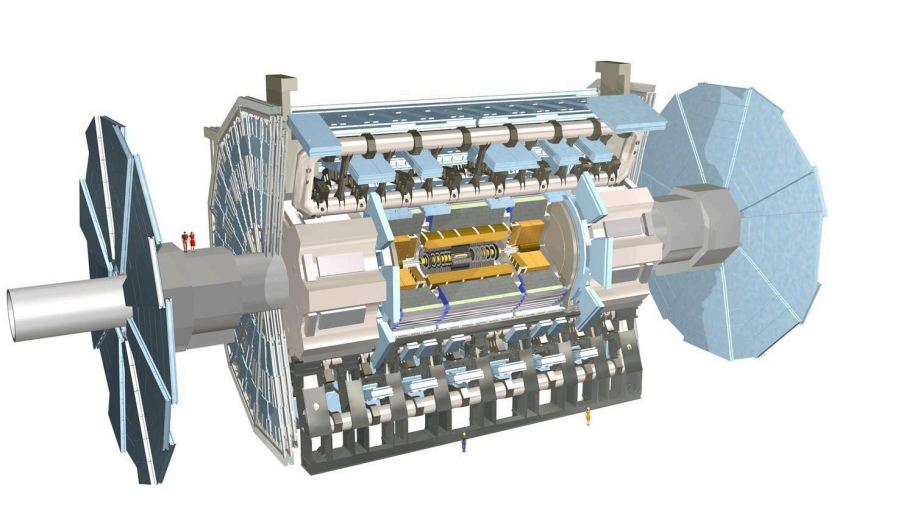
- A subset of A.I. involving computer programmes, often referred to as **models**, which:
  - are **trained** to achieve some **task**,
  - by performing **pattern recognition**.
  - These patterns are learned from some **example data**.
  - A key factor in ML is getting the computer to learn a **generalisable** mechanism for achieving the task

Why is ML useful?

- Potential to delegate trivial, repetitive or dangerous jobs to machines.
- Faster processing of information.
- Potential for deeper insights to be learned from more data.
- Vast volumes of data are being generated and stored digitally, more than any human could ever hope to look at and learn from.

Why is ML useful?

The ATLAS detector



Recorded	per event	per year
raw data	1.6 Mbytes	3 200 Tbytes
reconstructed data	1 Mbytes	2 000 Tbytes
physics data	0.1 Mbytes	200 Tbytes

(A terabyte is a million megabytes)

The 3200 terabytes of data are the equivalent of the content in:

- 160 million trees made into books.
- 7 km (4 miles) of CD-ROMs stacked on top of each other.
- 600 years of listening to songs.

Learning outcomes (part 1): tasks, models and features

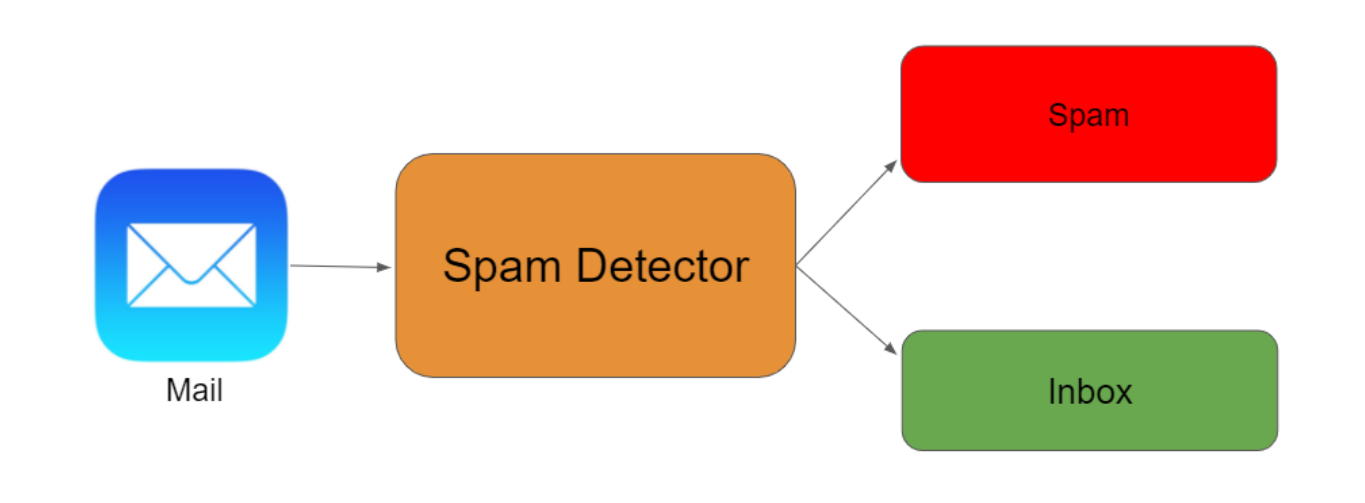
- Meet a few of machine learning **tasks** such as ***classification***, ***regression*** and ***clustering***.
- Learn what machine learning **models** are.
- Recognise the importance of **features** as an input to machine learning.

Mathematics and Machine Learning

- Machine learning is underpinned by a variety of mathematical theories and formalisms.
- Difficult to fully understand without at least some of the relevant mathematics:
  - probability theory
  - linear algebra
  - multivariable calculus
  - information theory
  - logic and set theory
- It is important to try and understand the intuition behind the mathematics. This influences the choice of what method to use in which situation.

Examples of ML tasks

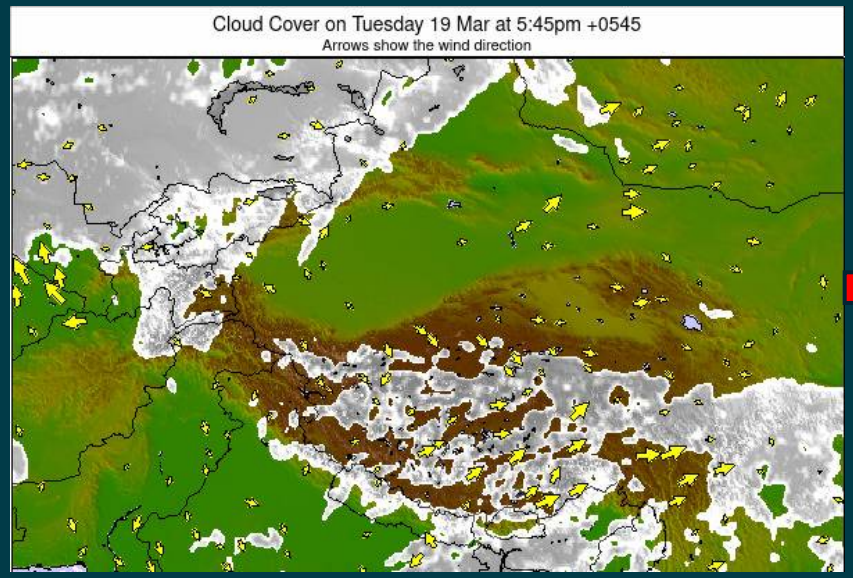
- Identifying whether an email contains irrelevant information (spam) or not (not spam).



Example of a (binary) classification task

Examples of ML tasks

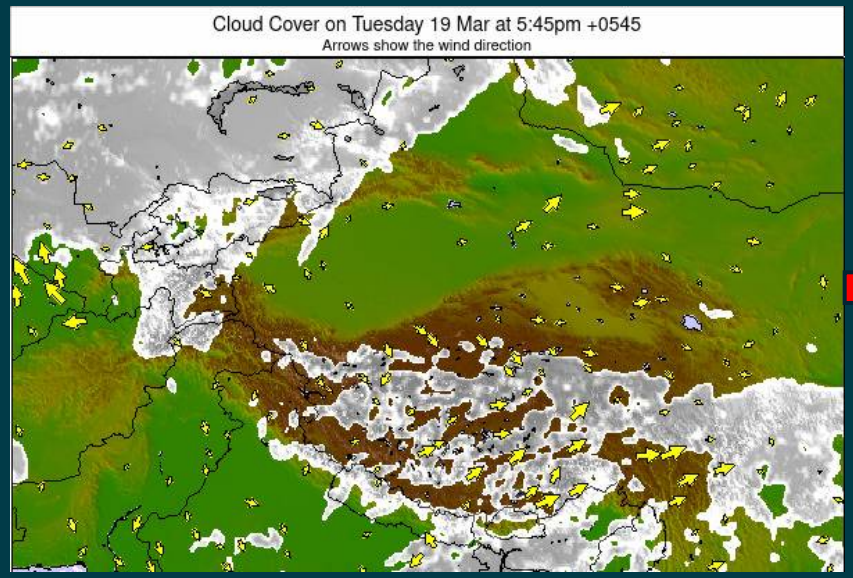
- Predicting from satellite images and environmental sensors whether tomorrow will be sunny or cloudy or rainy or thunderstorm etc.



Example of a (multiclass) classification task

Examples of ML tasks

- Predicting from satellite images and environmental sensors for tomorrow's temperature, wind, and humidity values.



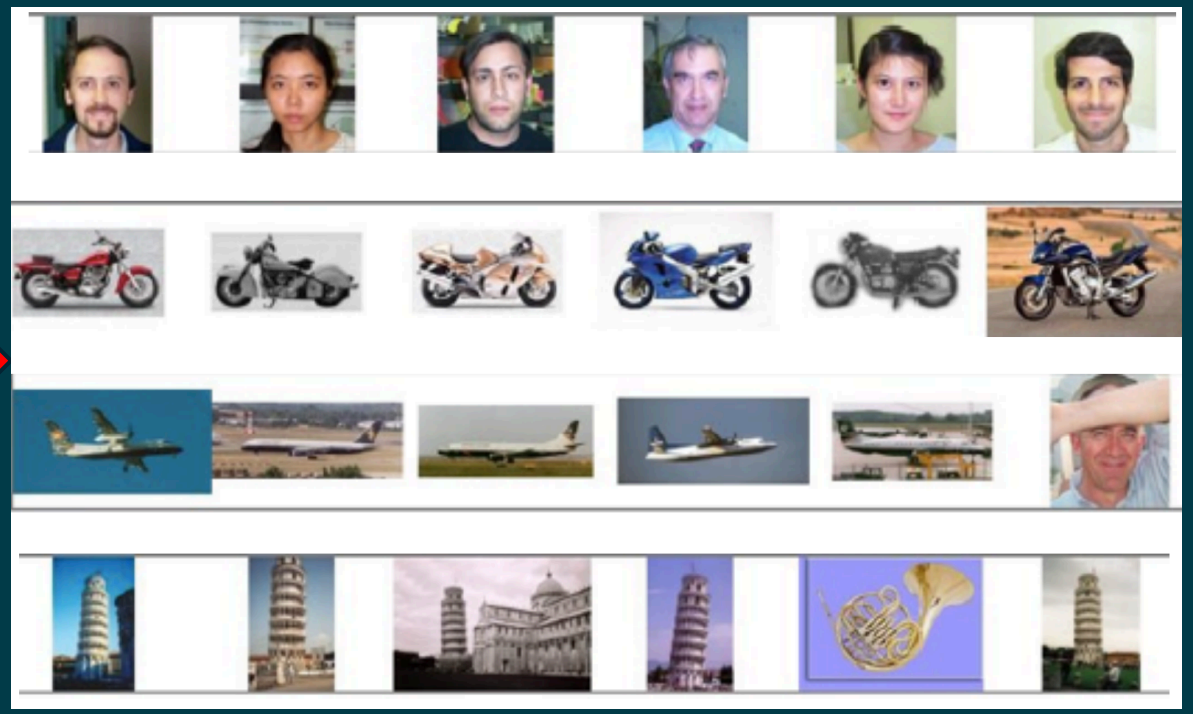
10 °C
Wind : 6mph
Humidity : 40%

Example of a regression task

Examples of ML tasks

- Grouping together images according to semantic similarities.

Set of images



Example of a clustering task

The power of ML

- Some problems are difficult or even impossible to formalise as a computer science problem, but humans can provide examples or feedbacks.
- Given an image of an animal, which animal is this?

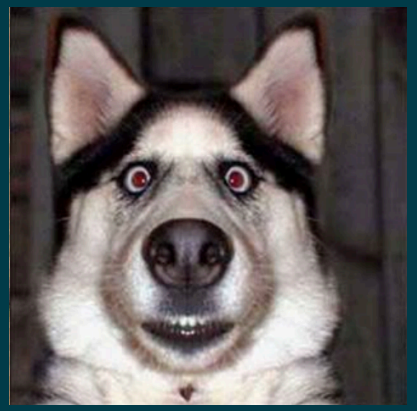


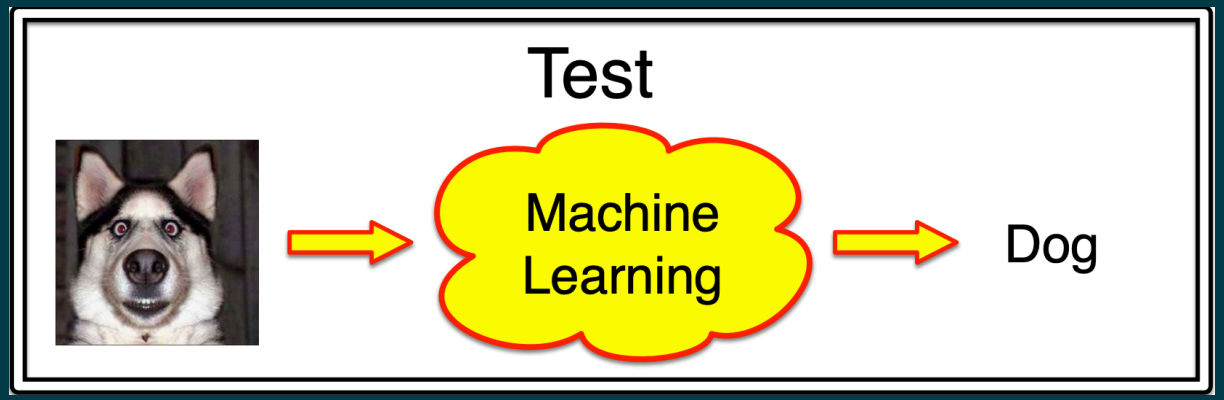
Image classification task

```
if ???
```

It is ***impossible*** to program a solution to this.

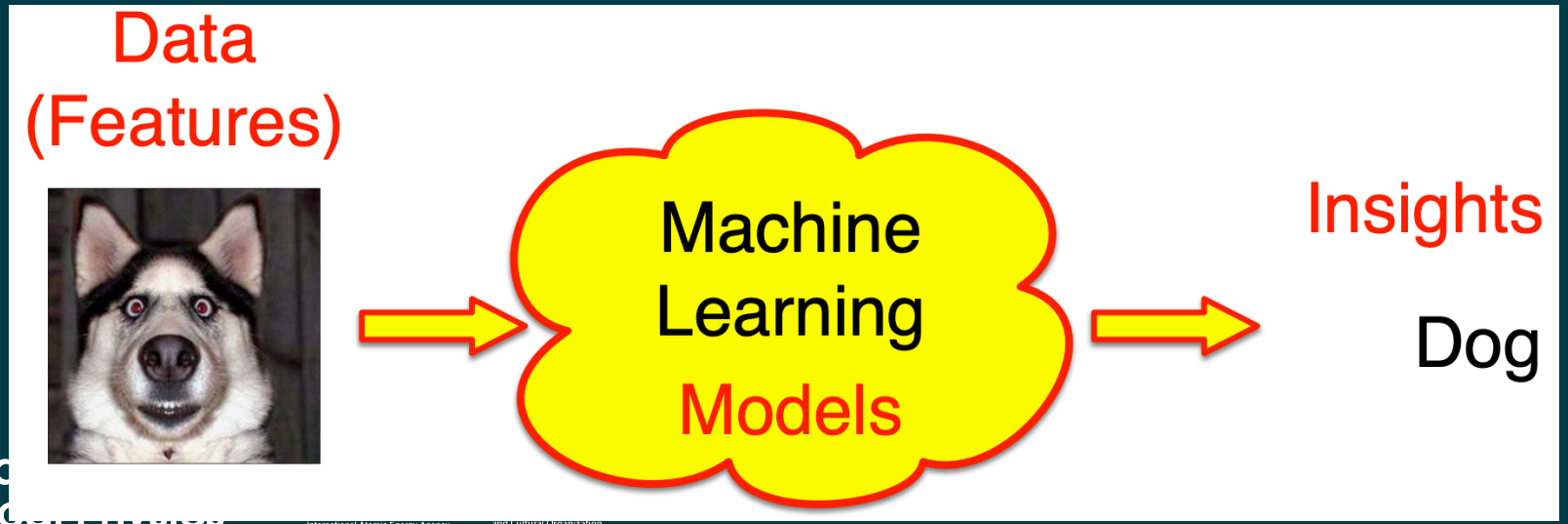
The power of ML

- The good news: we have labelled examples that we can use to train a computer to do this task!



Tasks, models and features

- Three main ingredients of machine learning:
- **Tasks**:
  - problems that require a mapping from data to desired outputs (insights).
- **Features**:
  - characteristics of the data used to describe domain objects.
- **Models**:
  - encode the required task mapping.



Core machine learning tasks

- **Classification**
  - Binary (C = 2) classification involves separating data into two distinct groups, often dnoted (+ and −)
    - Multi-class classification generalise to C > 2 different classes
- **Regression**
  - Involves mapping from data items to real values
- **Clustering**
  - Separating data into different clusters/concepts on the basis of their characteristics.

Some other ML tasks you may come across are: **dimensionality reduction**, **collaborative filtering** and **reinforcement learning.**

(Some) Machine learning tasks

Can you match the following ML tasks preformed on health data:

1. quantifying the risk of heart disease on the basis of personal health records

2. grouping people according to their genetic characteristics.

3. distinguishing people at risk from heart disease, from those not at risk.

a) **Classification**
b) **Regression**
c) **Clustering**

(Some) Machine learning tasks

Can you match the following ML tasks preformed on health data:

1. quantifying the risk of heart disease on the basis of personal health records

2. grouping people according to their genetic characteristics.

3. distinguishing people at risk from heart disease, from those not at risk.

a) **Classification**
b) **Regression**
c) **Clustering**

Data and features

- **Features** are the inputs from data that we use to create a machine learning **model**:
  - They are used to describe data objects
  - They represent particular characteristics of an instance (a single piece of data being analysed by a ML model)
  - They may be **numerical** (e.g. age, weight, income), **boolean** (e.g. is employed, is male), or **nominal/categorical** (nationality).

- **A model is only as good as the information it sees:**
  - Choice of features is therefore extremely important
  - Techniques for removing redundant features or transforming features in various ways are often crucial

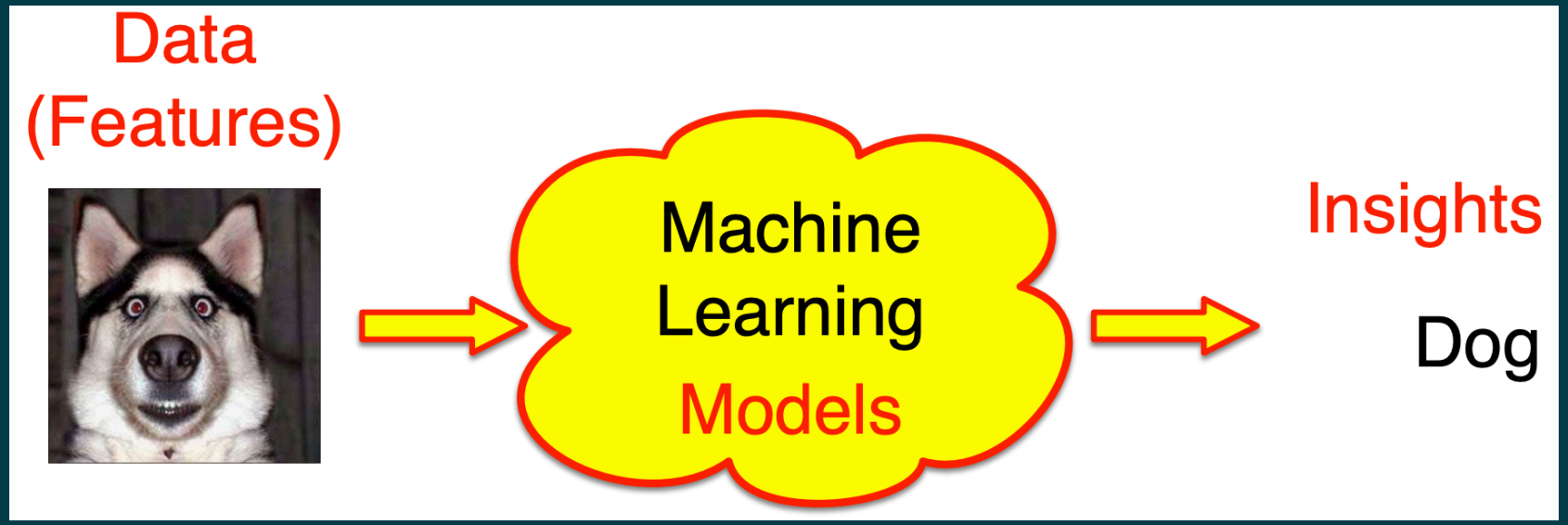
Machine Learning models

- **Models** form the central concept in machine learning:
  - They are a framework for making predictions from features.
  - They are parameterised with parameters learned from data.
  - They encode the mapping needed to solve a task.

- Models are unique to the task and data that you are working with.

Tasks, features and models: summary

Task: Classify a given image as either DOG or NOT DOG



Supervised vs unsupervised learning

- A supervised ML model is trained on data where each instance has a manually inputted label
  - Classification
  - Regression
- An unsupervised ML model is trained on data that has no labels – it is simply looking for patterns in the data
  - Clustering

Training/test split

- When using data to construct any ML model, it is important to first randomly split your data into 2 sets: **Training set** and **Test set**.
- Your model would be created using only the training set
- It would then be evaluated using the test set.
- This helps evaluate three main things about your ML model
  - Accuracy
  - Overfitting
  - Choice of hyperparameters

Instances and Instance Spaces

- Consider data to consist of a set of **instances**
  - an **instance** represents an object of interest
- Set of all possible instances is the **instance space**:
  - e.g. set of all email messages received
  - or, set of photos of animals

Notation

- We will denote the instance space by $\mathcal{X}$
- An individual instance will be denoted by x
- $x \in \mathcal{X}$

Labels and Label Spaces

- In supervised problems, each instance is associated with a **label**
- Set of all labels for a task is called the **label space**:
  - class labels C in classification tasks., e.g. C = {dog, cat}, or C = {signal, background}
  - real numbers $\subseteq$ R in regression tasks. e.g. temperature.

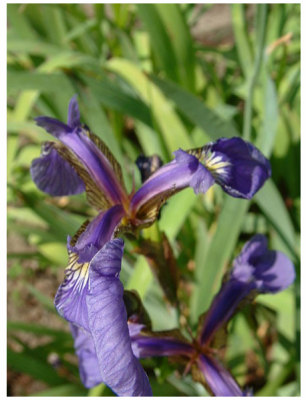
Notation

- We will denote arbitrary label space by $\mathcal{Y}$
- Labelling function $f : \mathcal{X} \to \mathcal{Y}$ maps from instances to labels
- Label associated with a given instance x denoted by $f(x)$

Binary Classification

- In the simplest classification case, we just have two class labels
  - positive (+ or +1)I negative (- or -1)
  - By convention the class of interest is labelled positive
- Binary classification task is to label instances with one or other of the class labels

Iris dataset



Iris setosa

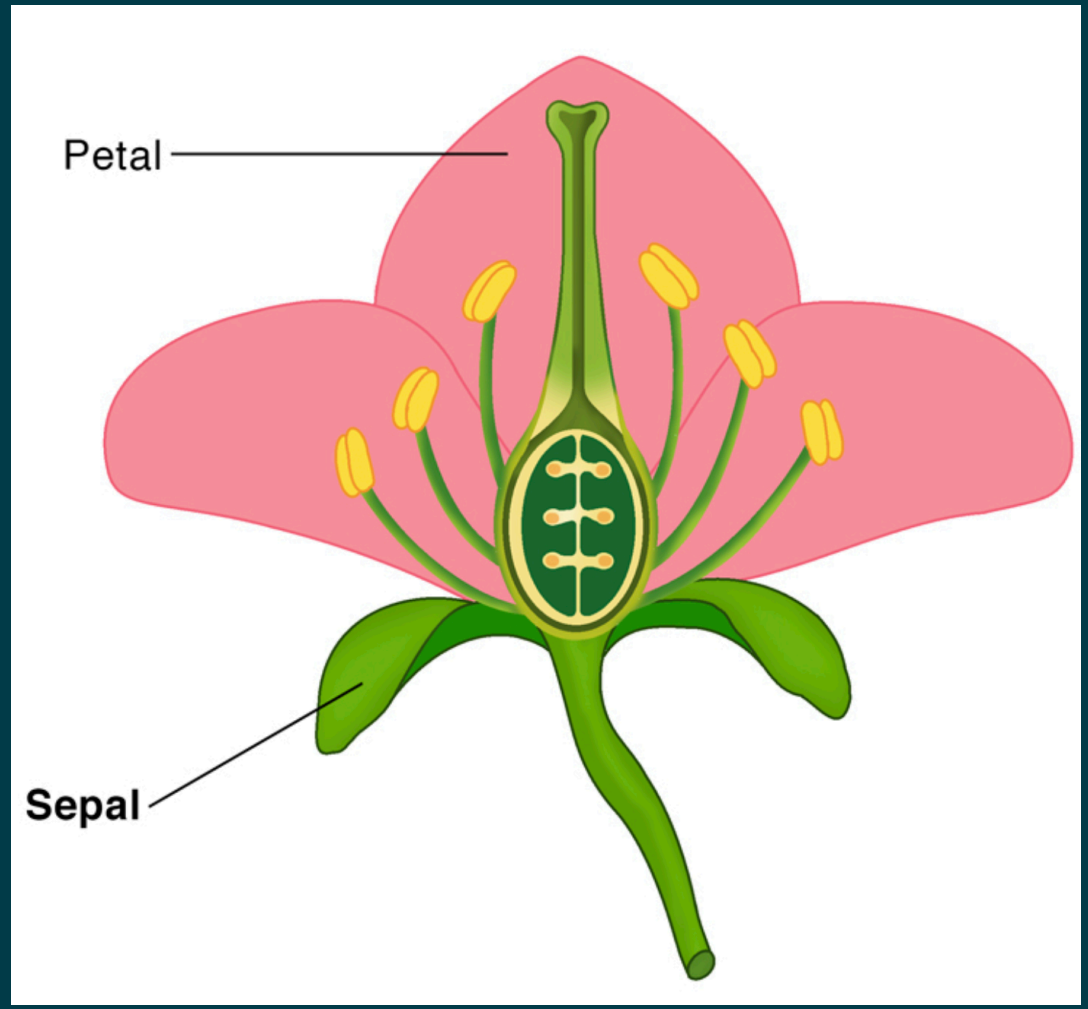
Setosa

Iris versicolor

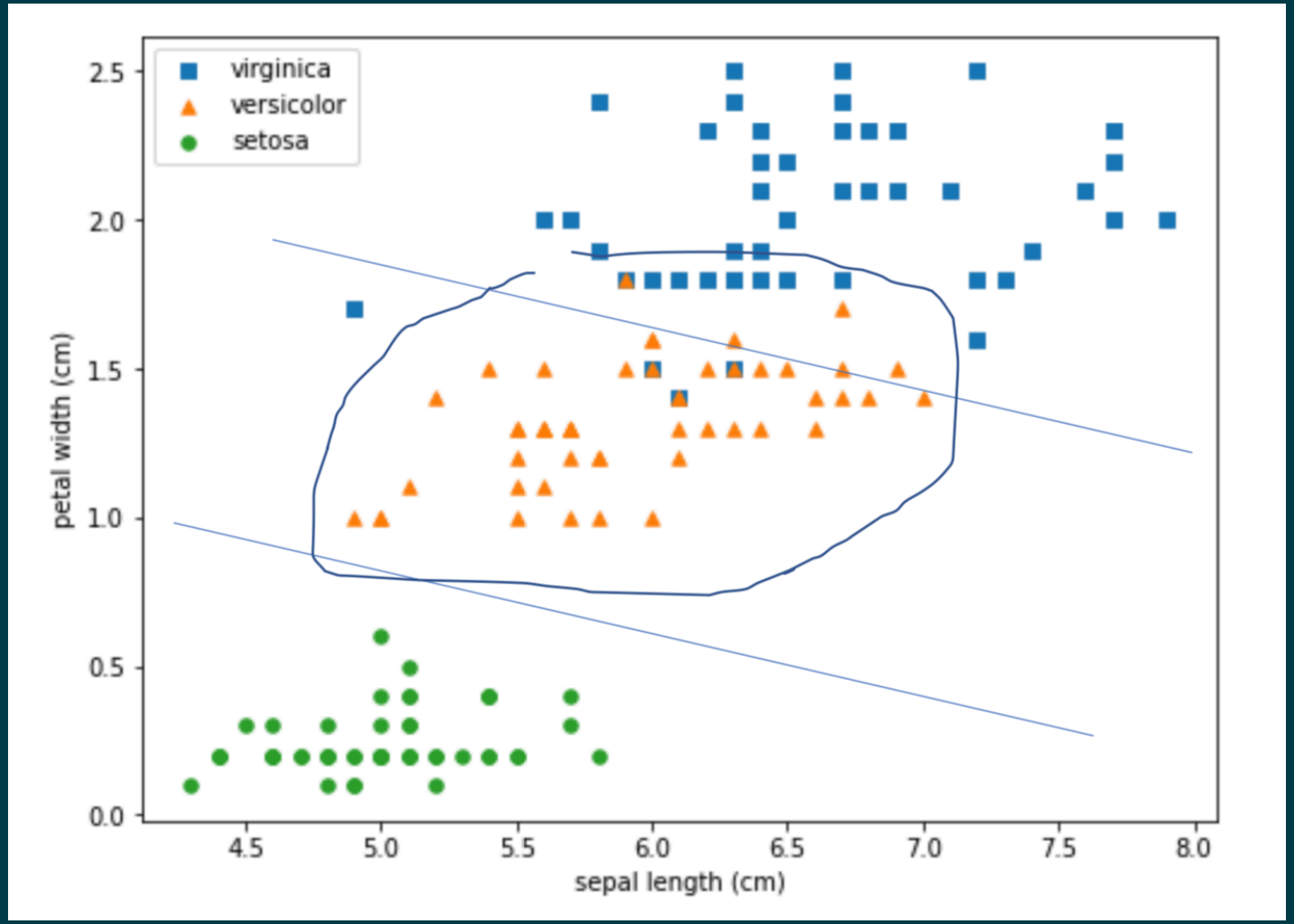
Versicolour

Iris virginica

Verginica



Linear vs non-linear boundaries

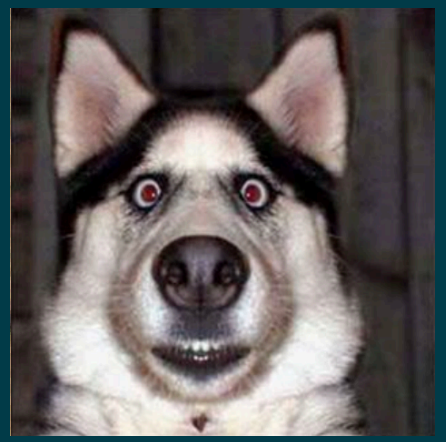


Learning a Classifier

- In practice, we do not know the true classification function f
- We have a training set of labelled instances, that is (x,f(x))
  - A set of manually annotated examples
- Use the examples to learn a model $\hat{f}: X \rightarrow C$
- $\hat{f}$ approximates the true classification function f
  - Should follow the training examples closely
  - Should generalise to whole of the instance space X

Binary Classification: Assessing Performance

- For a learned binary classifier, $\hat{f}$ approximates the true classification function f
  - $\hat{f}$ will not exactly be the same function as f
  - $\hat{f}$ will make errors in assigning class labels to instances
- How can we assess the performance of a learned binary classifier?



Cat

Binary Classification: Assessing Performance

- Use a **contingency table** (also known as **confusion matrix**)

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

Accuracy and error

- **Accuracy**: proportion of correctly predicted instances
- **Error**: proportion of incorrectly predicted instances

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

- **Accuracy: (30 + 90) / 150 = 0.8**
- **Error: (10 + 20) / 150 = 0.2**

Binary Classifications: More useful metrics

- Although the accuracy is important, it makes the unrealistic assumption that both classes are equally important to us. **Take the example of a face recognition system:**
- Imagine the + class is one particular identity, so the - class is all others.
- We have a trade off between always correctly identifying the + class, and mis-identifying the - class.
- If we err on the side of caution, i.e. always correctly find the + class, we will incorrectly find people from the -ve class.
- This can have major consequences!

Binary Classifications: More useful metrics

- **Take the example of a medical diagnosis system:**
- Imagine the + class is bad cancer (surgery), and the - class is benign (no surgery).
- We have a trade off between always correctly identifying the + class, and mis-identifying the - class.
- If we err on the side of caution, i.e. always correctly find the + class, we will incorrectly operate on people who would be fine otherwise, opening them up to unnecessary complications.
- Alternatively, we could be more optimistic and miss some malignant cases.
- The choice of metrics must be appropriate to the application.

Binary Classification: Per-Class Accuracy

- **True positive rate (sensitivity):** proportion of correctly predicted +ve instances
- **True negative rate (specificity)**: proportion of correctly predicted -ve instances

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

- **True positive rate (sensitivity): 30 / 50 = 0.6**
- **True negative rate (specificity): 90 / 100 = 0.9**

Binary Classification: Per-Class inaccuracy

- **False positive rate:** proportion of incorrectly predicted +ve instances
- **False negative rate**: proportion of incorrectly predicted -ve instances

	Predicted +	Predicted -	
Actual +	30	20	50
Actual -	10	90	100
	40	110	150

- **False positive rate: 10 / 50 = 0.6**
- **False negative rate: 20 / 100 = 0.9**

Quiz - Confusion Matrix Problem

- In a dog (+)/cat(-) classifier what does a false positive instance correspond to?

1. A dog incorrectly classified as a cat
2. A cat incorrectly classified as a dog
3. A dog correctly classifier as a dog
4. A cat classifying a dog as problematic

2. A cat incorrectly classified as a dog

Quiz - Confusion Matrix Problem

- In a facial recognition system designed to unlock a user's mobile phone by looking at the screen, where your face is a +ve instance and all other faces are –ve instances, what would be more likely to happen if the system was tuned to be more specific than sensitive?

1. Random people could unlock your phone
2. Sometimes you would be locked out of your phone

- **True positive rate (sensitivity):** proportion of correctly predicted +ve instances
- **True negative rate (specificity)**: proportion of correctly predicted -ve instances

2. Sometimes you would be locked out of your phone

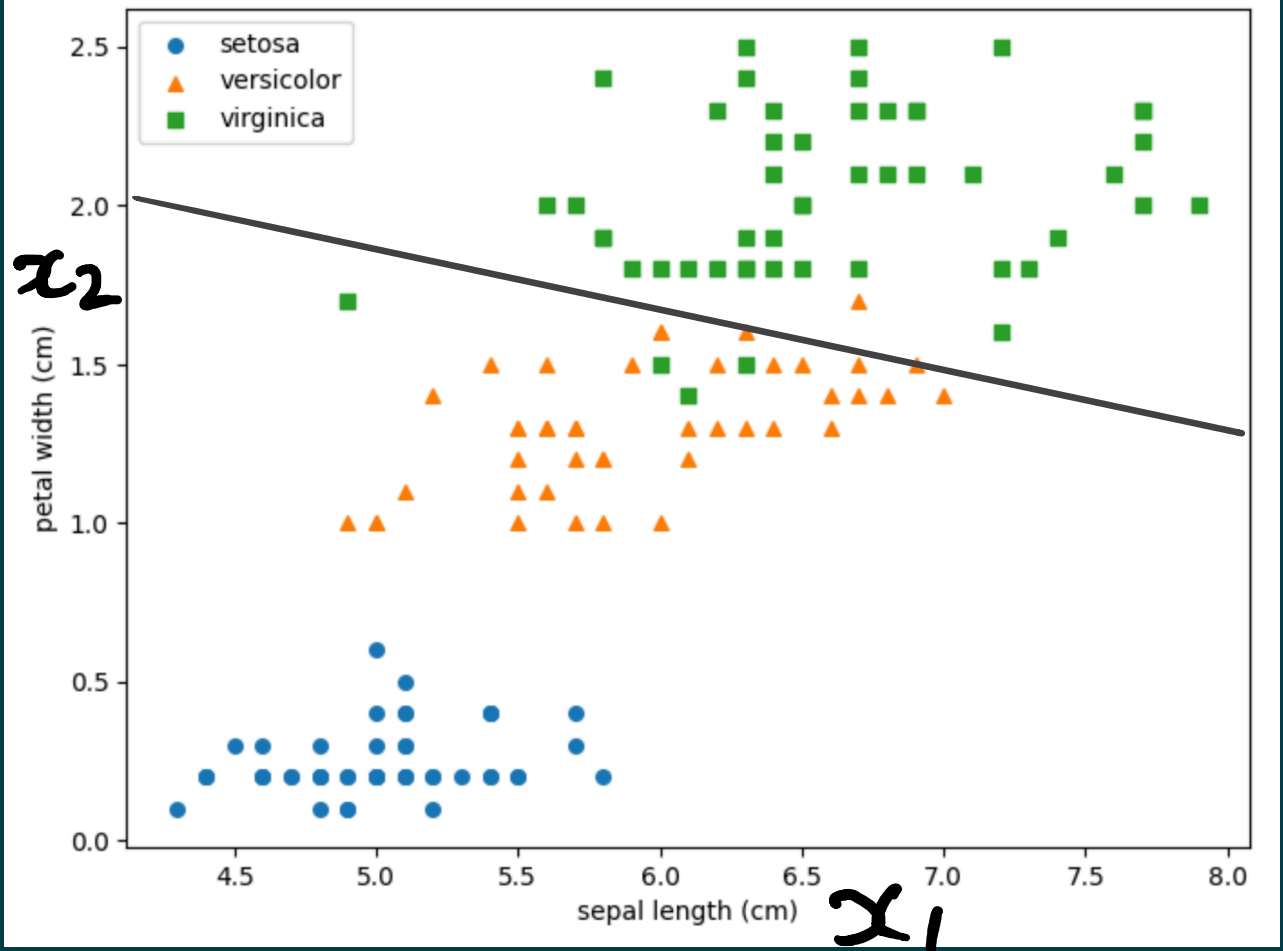
Models for (binary) classification

- **Logistic regression**
- Fits a linear decision boundary to the training data
- For two variables x_1, x_2, this would be a straight line:

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

- The model would optimise the weights based on the training data
- For every instance of data, it would assign a probability that it is one side of the design boundary or the other.

Logisitic regression



$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

In terms of the weights, what are the gradient and intercept of this line?

$$\text{gradient} = -\frac{w_1}{w_2}$$

$$\text{intercept} = -\frac{w_0}{w_2}$$

Decision trees

- Decision trees are powerful predictive models that allow predictions that are non-linear in the input.
- They have been applied to problems such as classification, probability density estimation and regression.
- Decision trees are readily interpreted, their decision making process can be understood by humans.
- Decision trees are highly expressive
  - good for separating complex data
  - but need to worry about over-fitting

Decision tree example

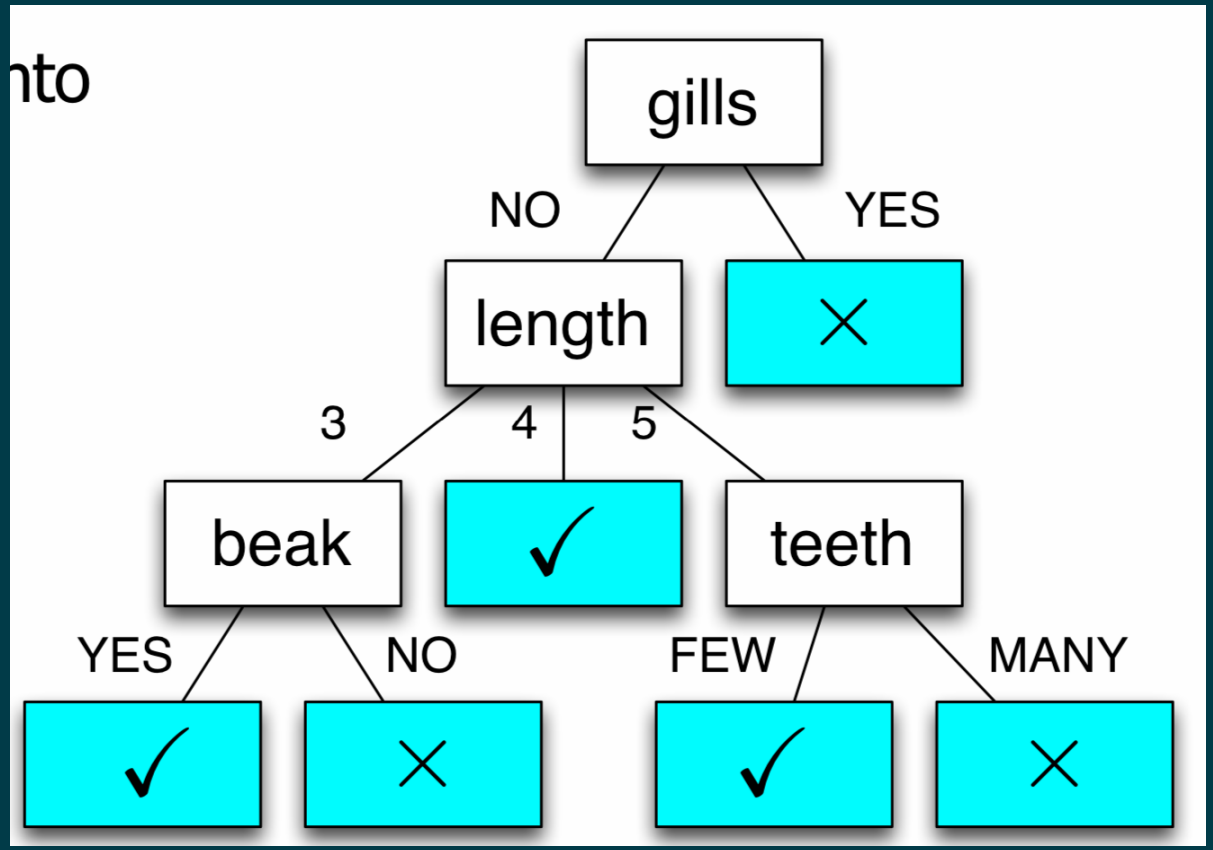
- Suppose we are building a binary classifier of sea creatures.
- We collect some properties that are assumed significant for identification:
  - Whether or not they have gills
  - The length of the creatures in metres
  - Whether they have a beak or not
  - Whether they have few or many teeth

Decision tree example

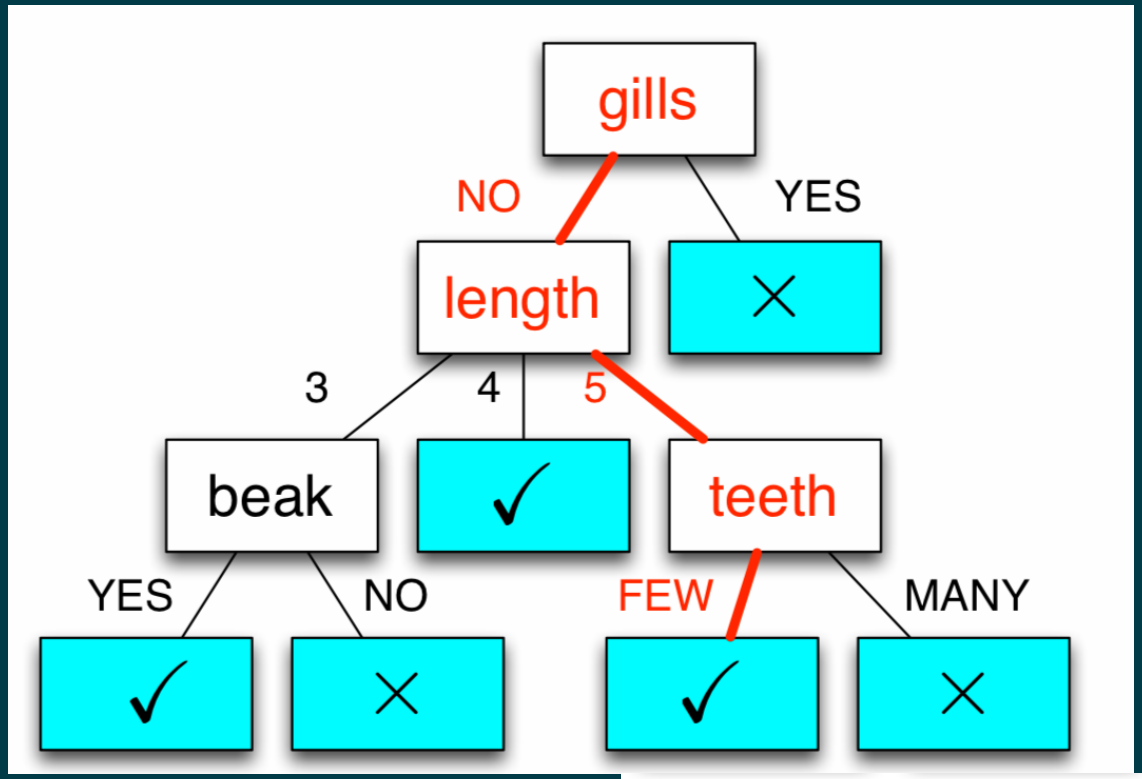
	gills	length	beak	teeth	class
example1	yes	3	no	few	×
example2	no	5	yes	few	✓
example3	no	3	yes	many	✓
example4	yes	5	no	many	×
example5	no	5	yes	many	×
example6	no	4	yes	many	✓
example7	no	5	no	few	✓
example...	no	3	no	few	×

Decision tree example

- Recast observed properties into a tree of questions (nodes):
  - nodes labelled by features
  - edges labelled by values of those features
  - leaf nodes labelled by class labels



Decision tree example

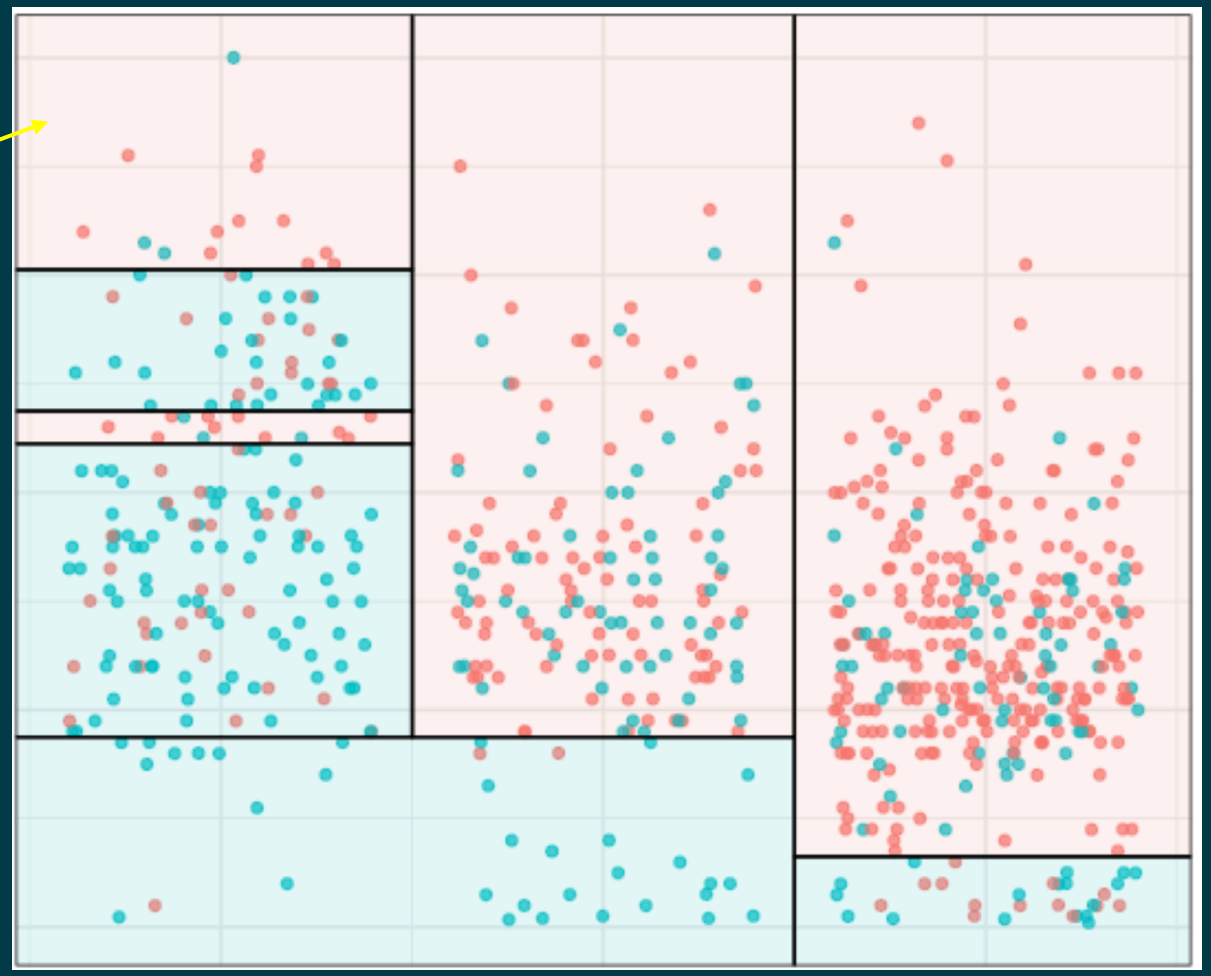


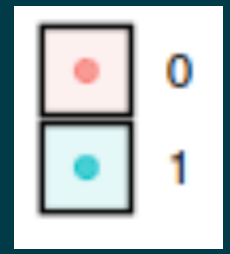
A conjunction of tests

$$\text{gills=no} \wedge \text{length=5} \wedge \text{teeth=few}$$

Decision tree – assigning probabilities

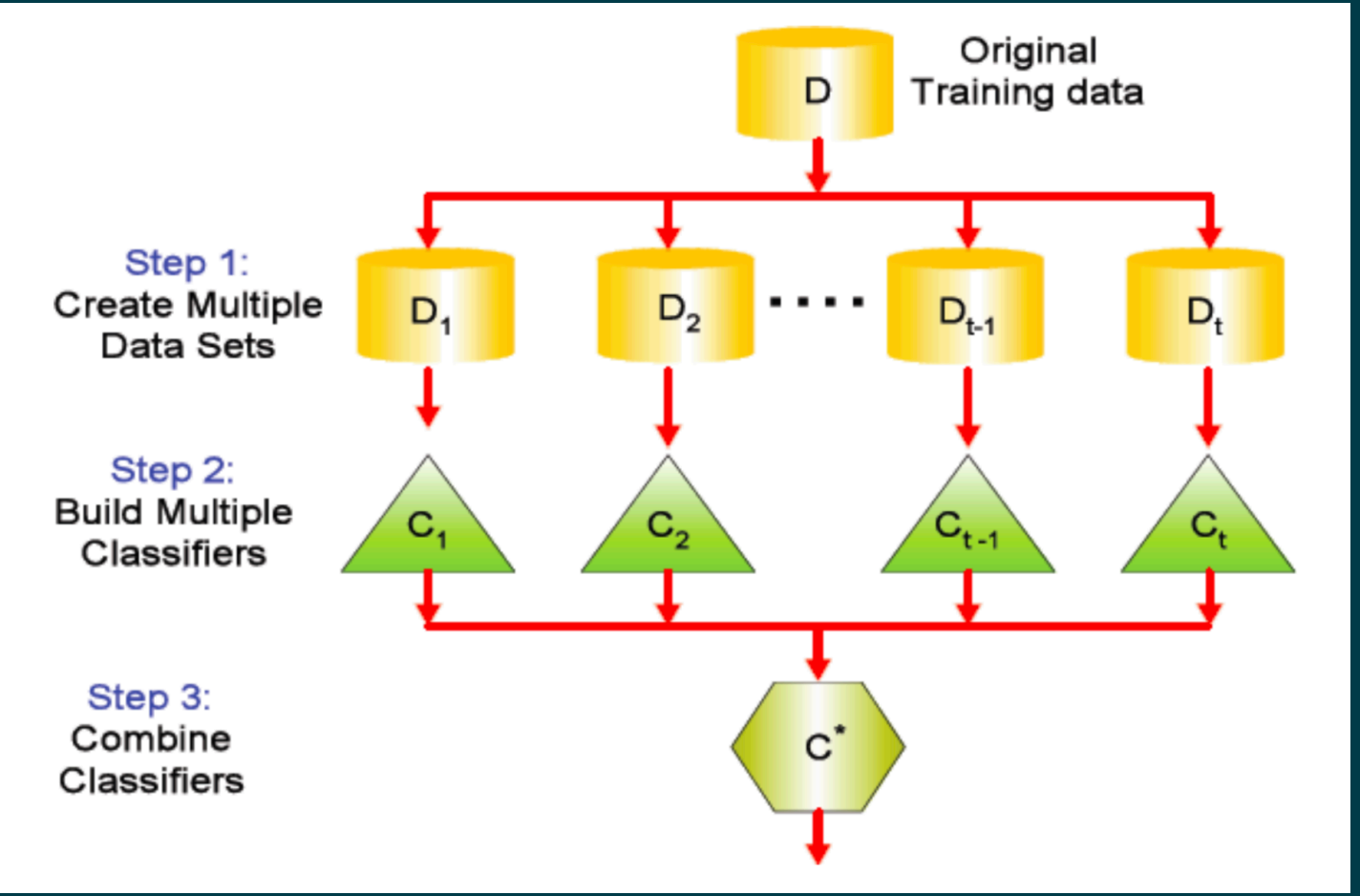
P(0) = 11/14
P(1) = 3/14





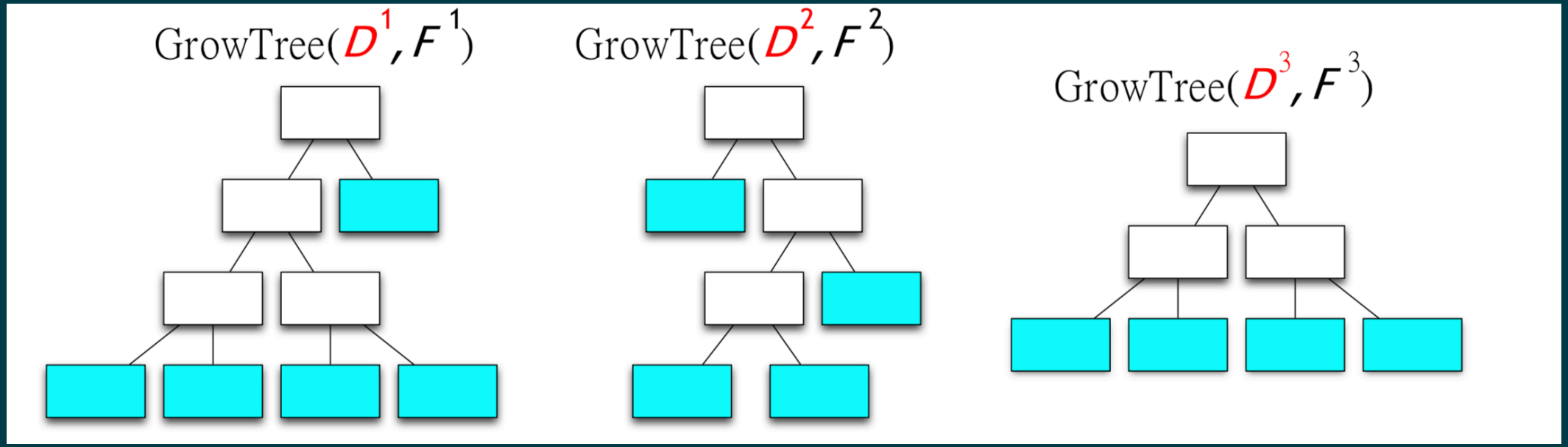
Ensemble learning

- To reduce model variance and overfitting

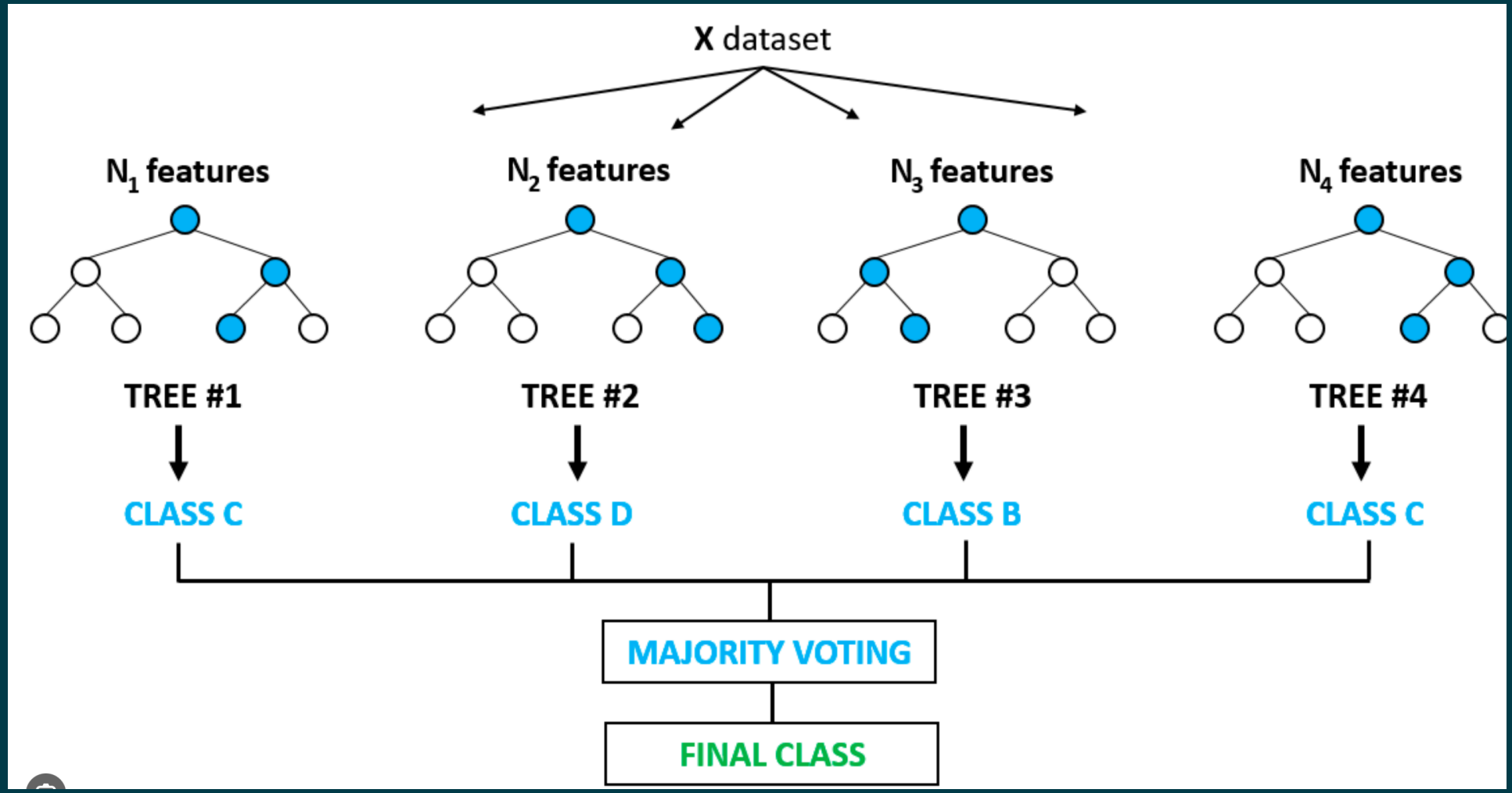


Random forest

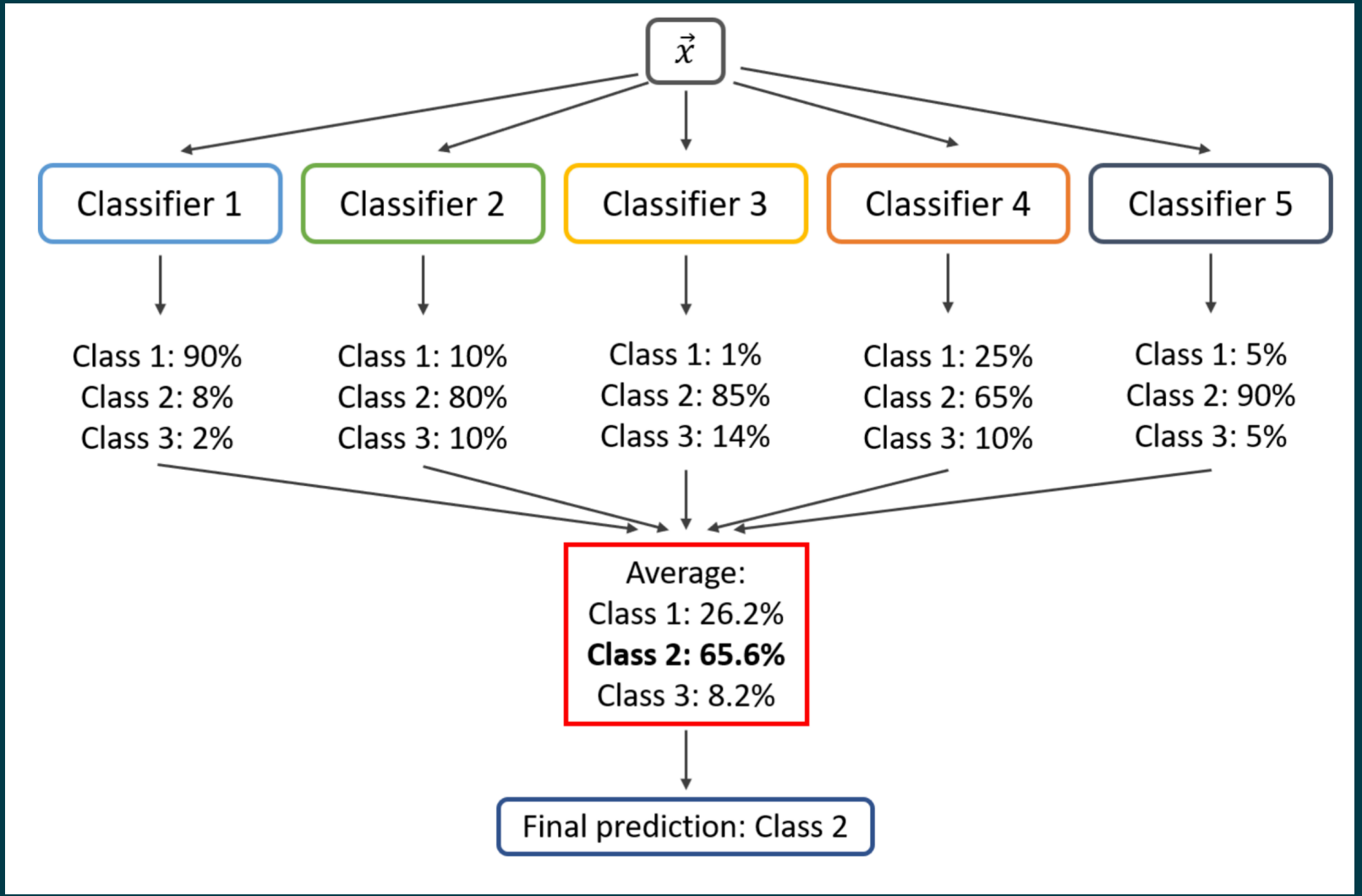
- A forest is an ensemble of trees.
- Each tree is slightly different from the others.
- Two sources of randomness in the trees:
  - Random sampling of the training data
  - Random subset of data features



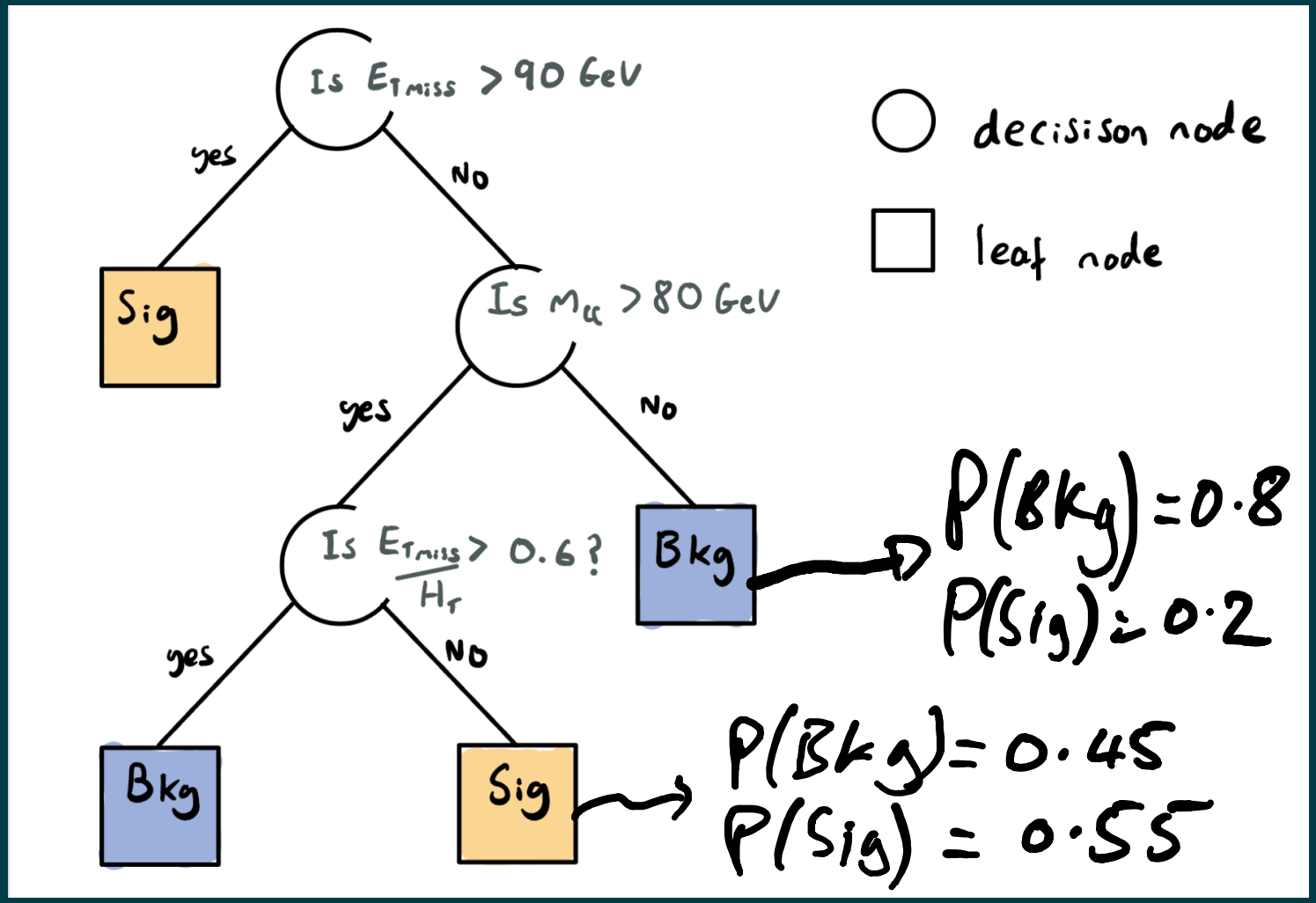
Majority (or hard) voting



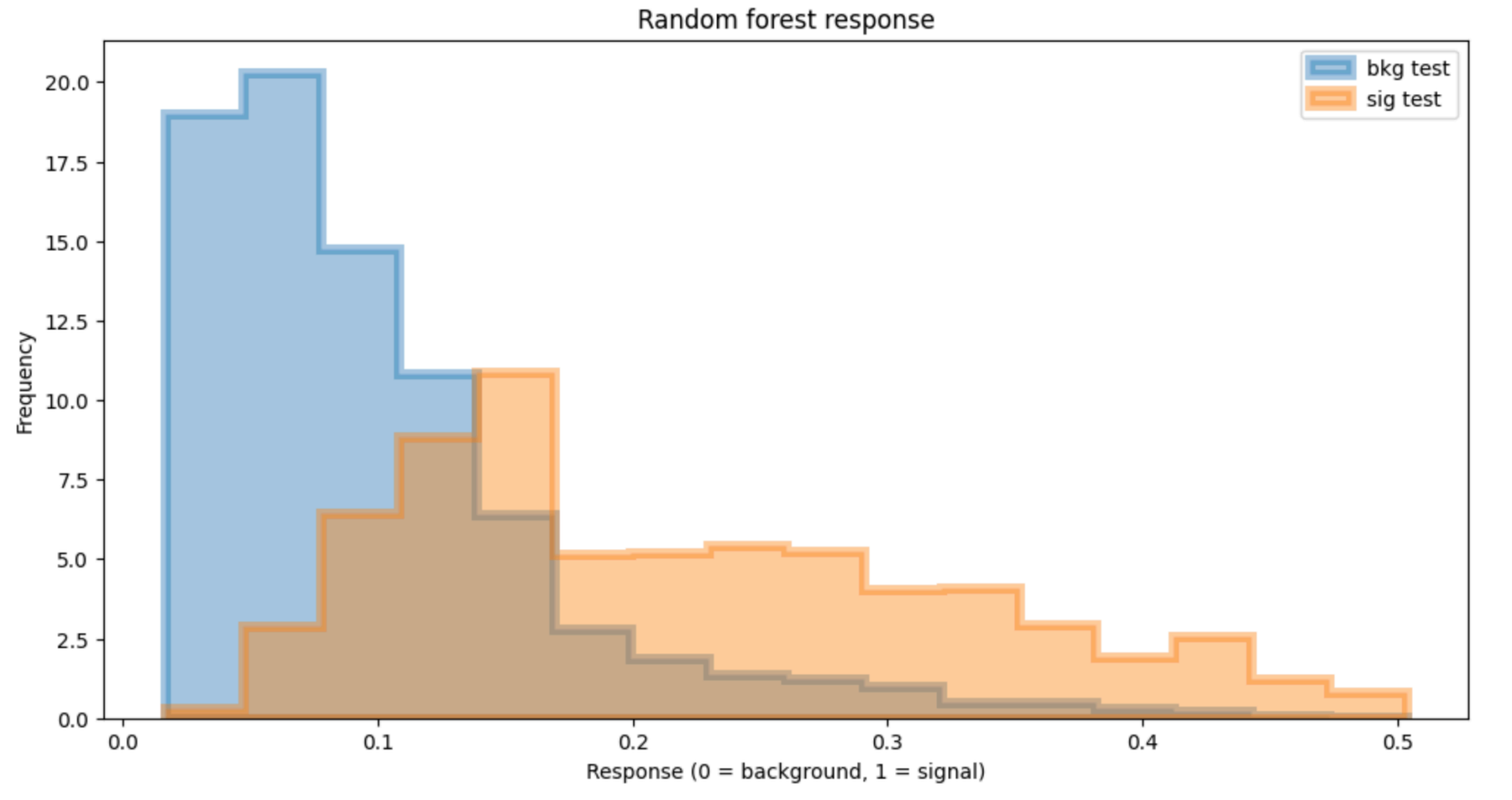
Soft voting


$\vec{x}$
Classifier 1
Classifier 2
Classifier 3
Classifier 4
Classifier 5
Class 1: 90%
Class 2: 8%
Class 3: 2%
Class 1: 10%
Class 2: 80%
Class 3: 10%
Class 1: 1%
Class 2: 85%
Class 3: 14%
Class 1: 25%
Class 2: 65%
Class 3: 10%
Class 1: 5%
Class 2: 90%
Class 3: 5%
Average:
Class 1: 26.2%
Class 2: 65.6%
Class 3: 8.2%
Final prediction: Class 2

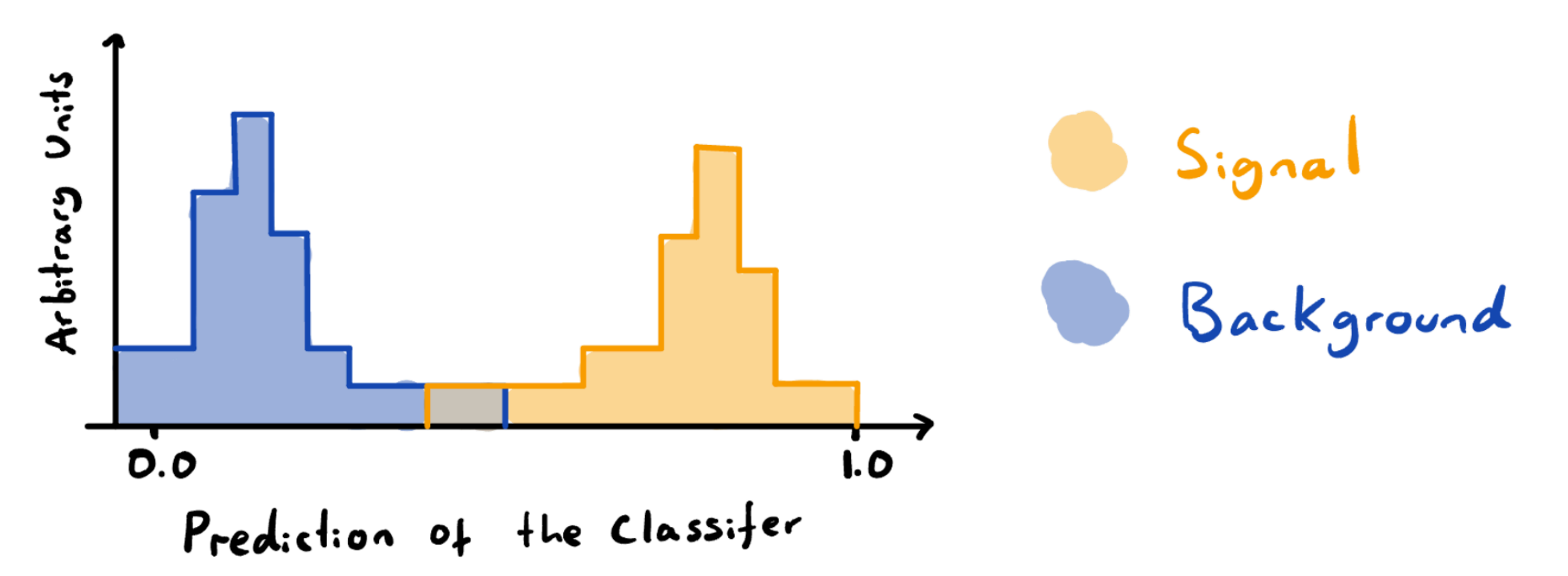
Random Forests and dark matter



Random forests and dark matter



Random forests and dark matter



Scikit-learn: Python ML library

scikit-learn

Machine Learning in Python

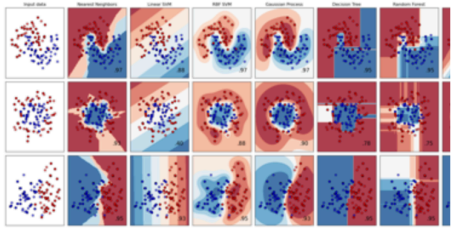
Getting Started | Release Highlights for 1.4 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

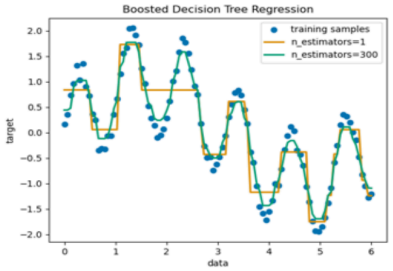


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

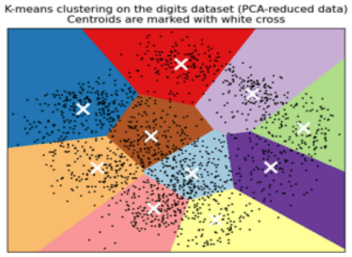


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Model selection

Comparing, validating and choosing parameters and models.

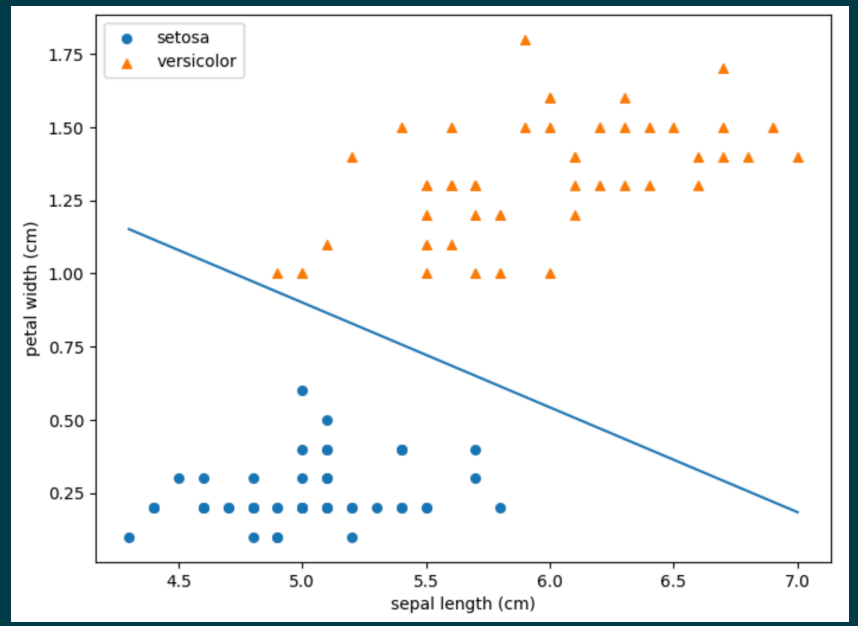
Preprocessing

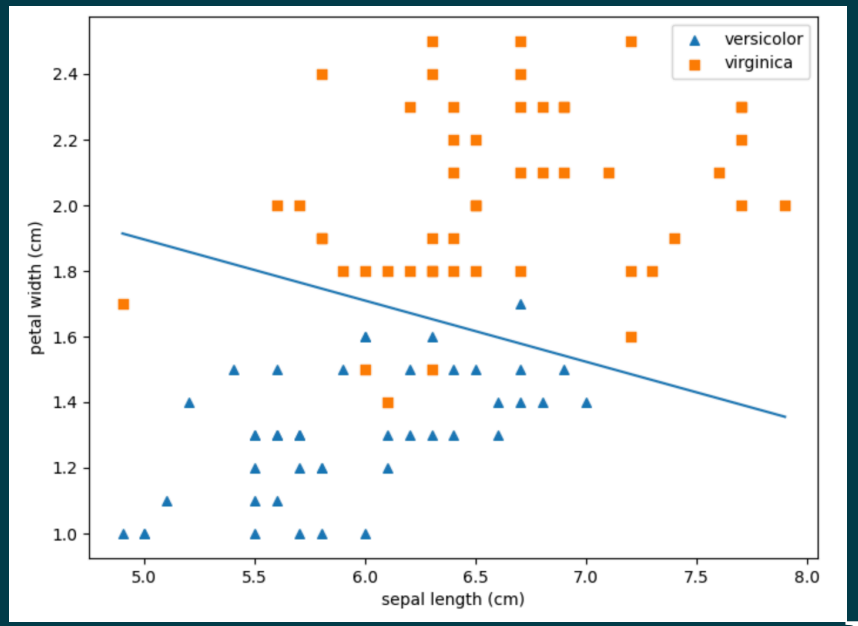
Feature extraction and normalization.

Applications: Transforming input data such as

Scikit-learn

```
# Create a classifier object
classifier = linear_model.LogisticRegression()
# Fit it to the data
classifier.fit(X,y)
```





Scikit-learn

```
proportion_training = 0.2
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, train_size=proportion_training,
                                                                    stratify=y)
```

```
# Now train the classifier on the training set
classifier.fit(X_train, y_train)
# We can predict on the test set
y_test_pred = classifier.predict(X_test)
```

```
print(iris.target_names[:])
metrics.plot_confusion_matrix(classifier, X_test, y_test,
```