Edge Hill University

# A.I. for Astrophysics: Masterclass

Dr. Rob Lyon
Edge Hill University
robert.lyon@edgehill.ac.uk
@scienceguyrob

Edge Hill University

# Welcome

In the next few hours we'll learn how to apply ML to a real-world astrophysics problem. If you've never done machine learning before don't worry. I'll cover everything you need to know including:

- Machine learning basics.
- Python basics.
- The astronomy background you'll need to understand the challenge.

If you have questions at any point – just ask. We'll move through things slowly. We want this masterclass to be enjoyable!

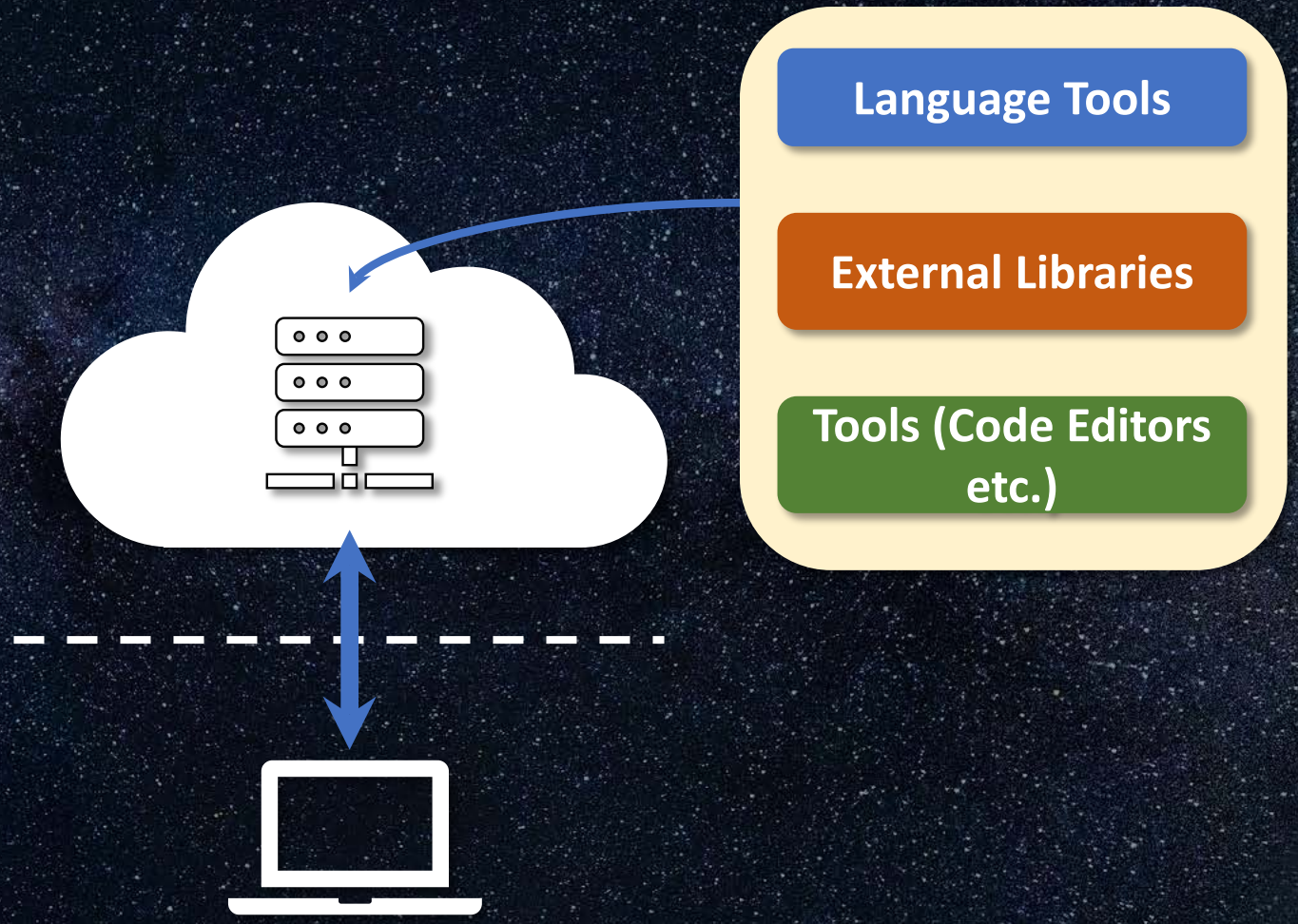
Edge Hill University

# Masterclass Environment

Edge Hill University

# Google Colab

- A cloud environment containing the tools need to write & execute Python programs.
- You'll need is a google account to use, e.g. a Gmail Account and the *Chrome* Web Browser.
- When you login to the Collaboratory, it creates a computer just for you.
- Inside you can create & execute Python code.
- Before proceeding, please create a Google account if you don't already have one.



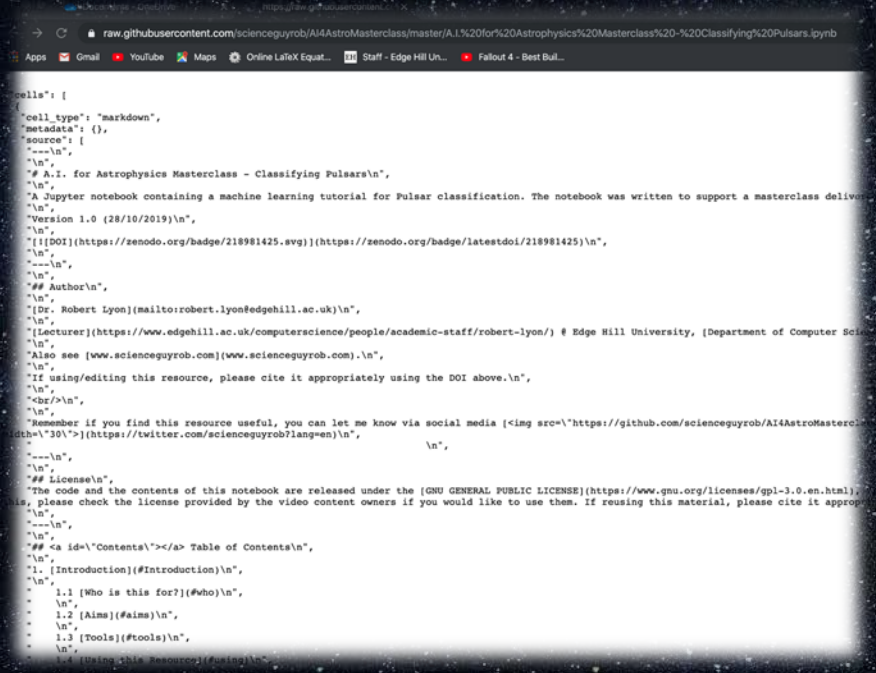
Edge Hill University

# Accessing the Colab

Edge Hill University

# Accessing the Masterclass Resource

1. Open Google Chrome.
2. Navigate to: https://github.com/scienceguyrob/AI4AstroMasterclass
3. Click the file: “`A.I. for Astrophysics Masterclass - Classifying Pulsars.ipynb`”
4. There is a button with the text “Raw” on it – click it. This will load the resource into your browser.
5. Save the content to your own machine – left click and choose the “Save as” option that appears on the popup menu. Keep the `.ipynb` file extension.
6. Open Google Colab: https://colab.research.google.com/
7. Click the File menu, click “Upload Notebook”. Choose the file you saved on your machine.



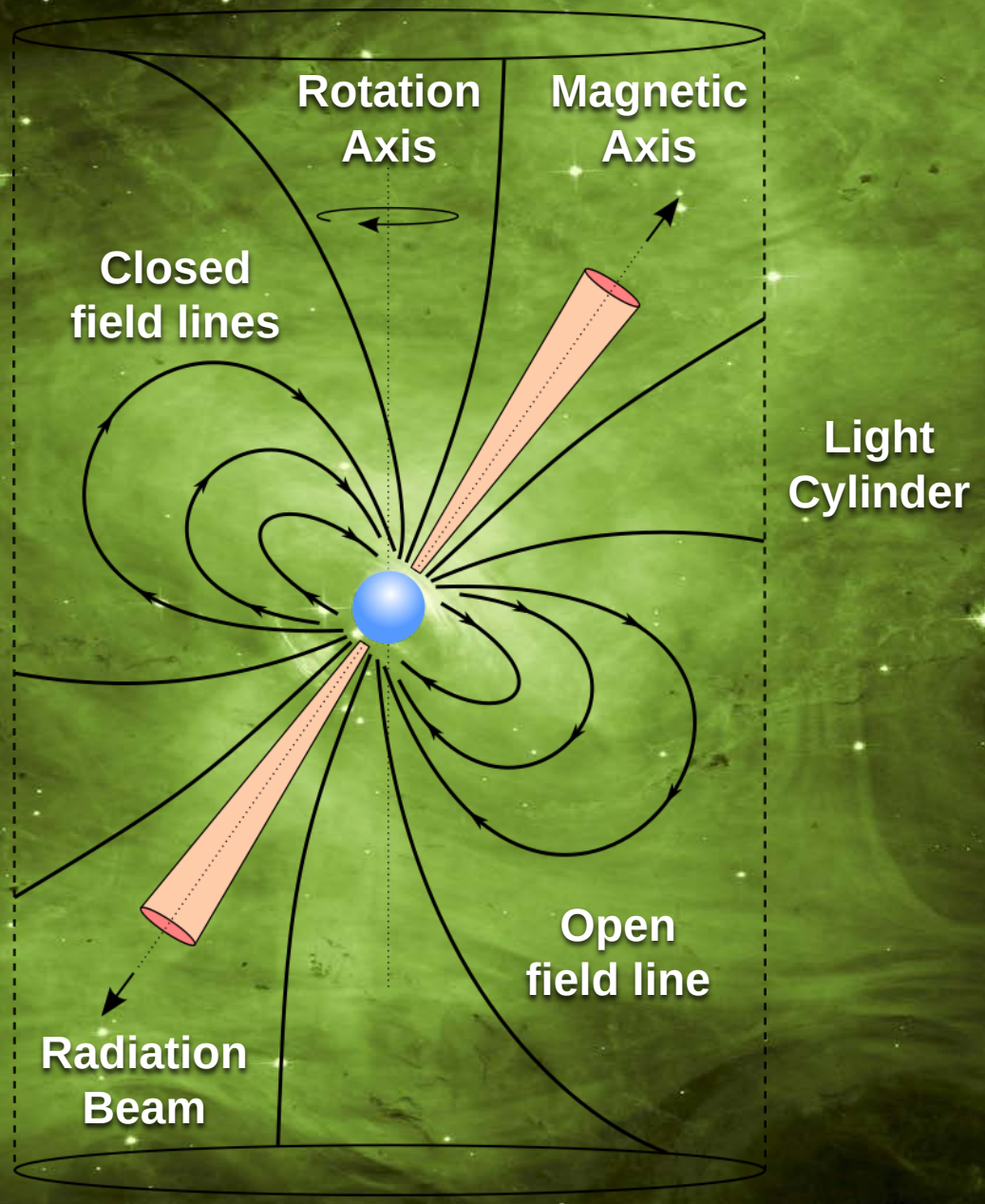
Edge Hill University

# Astrophysics Problem

Edge Hill University

# Radio Pulsars

- Stellar remnant
- Very dense
- ~20 km diameter
- Produce radio emission
- Very useful for science



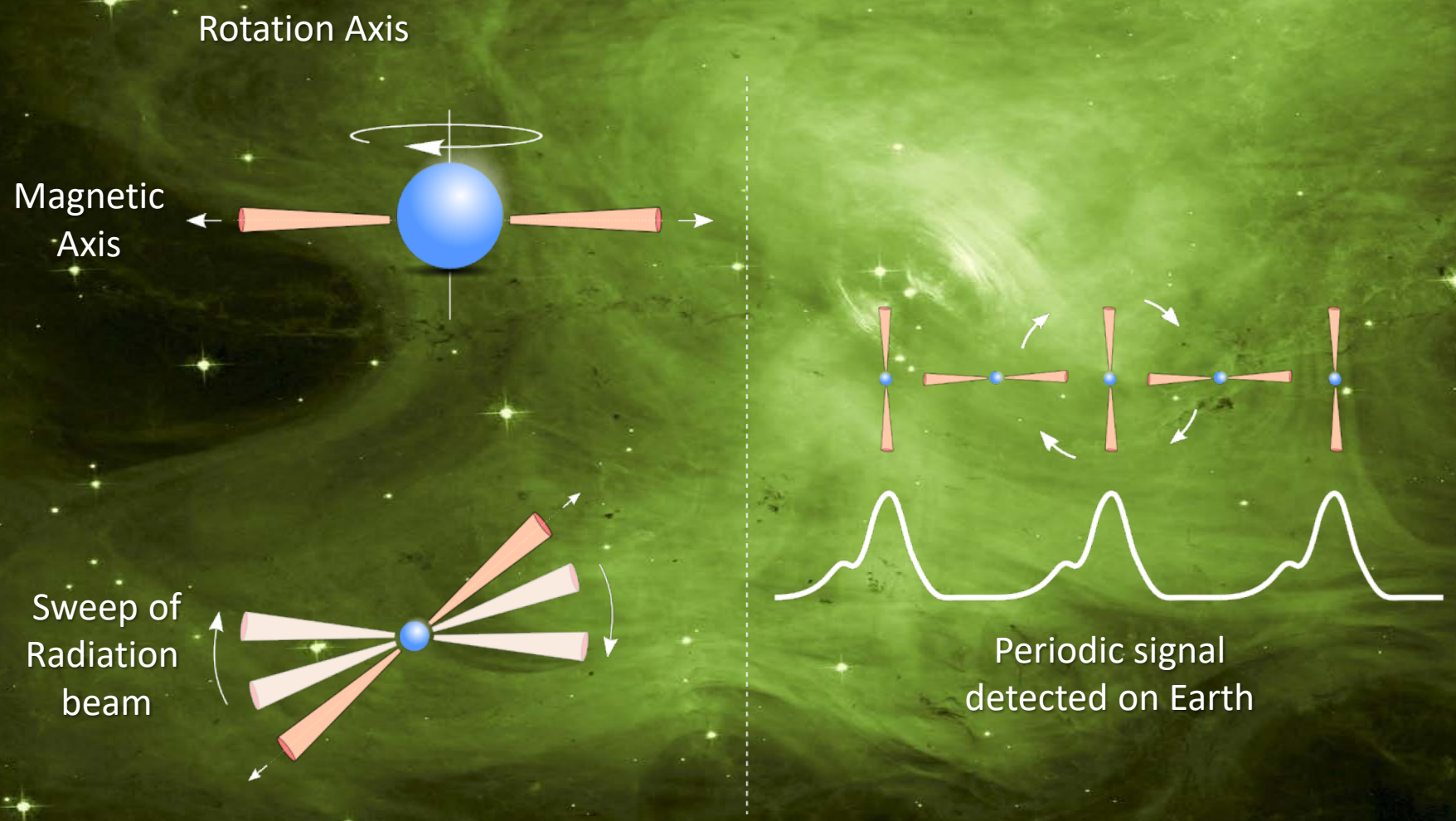
Edge Hill University

# Pulsar Birth

Credit: John Rowe Animations

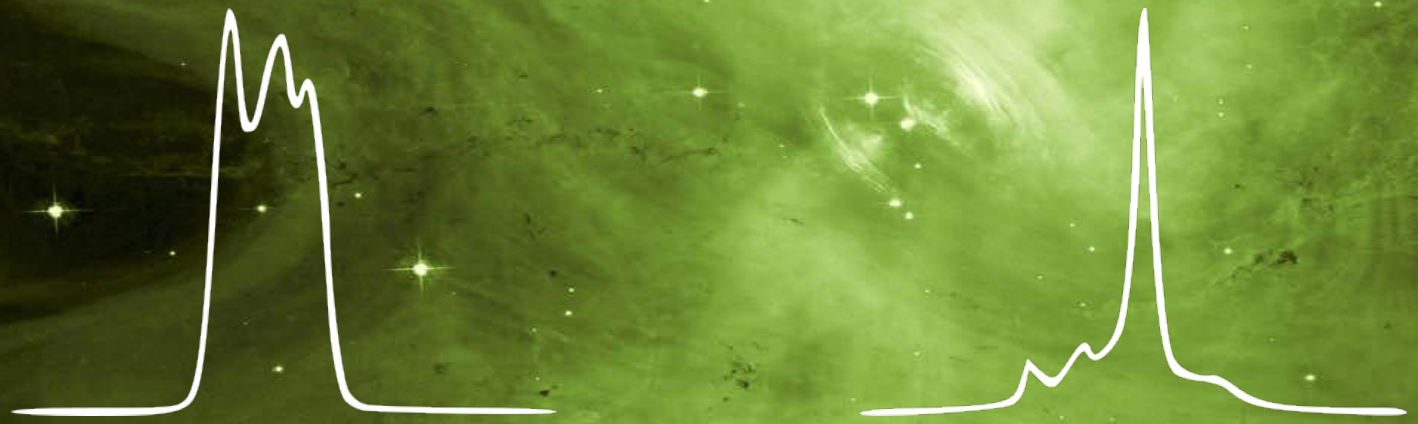
Edge Hill University

# Pulsar Signals



Edge Hill University

# Pulse Profiles

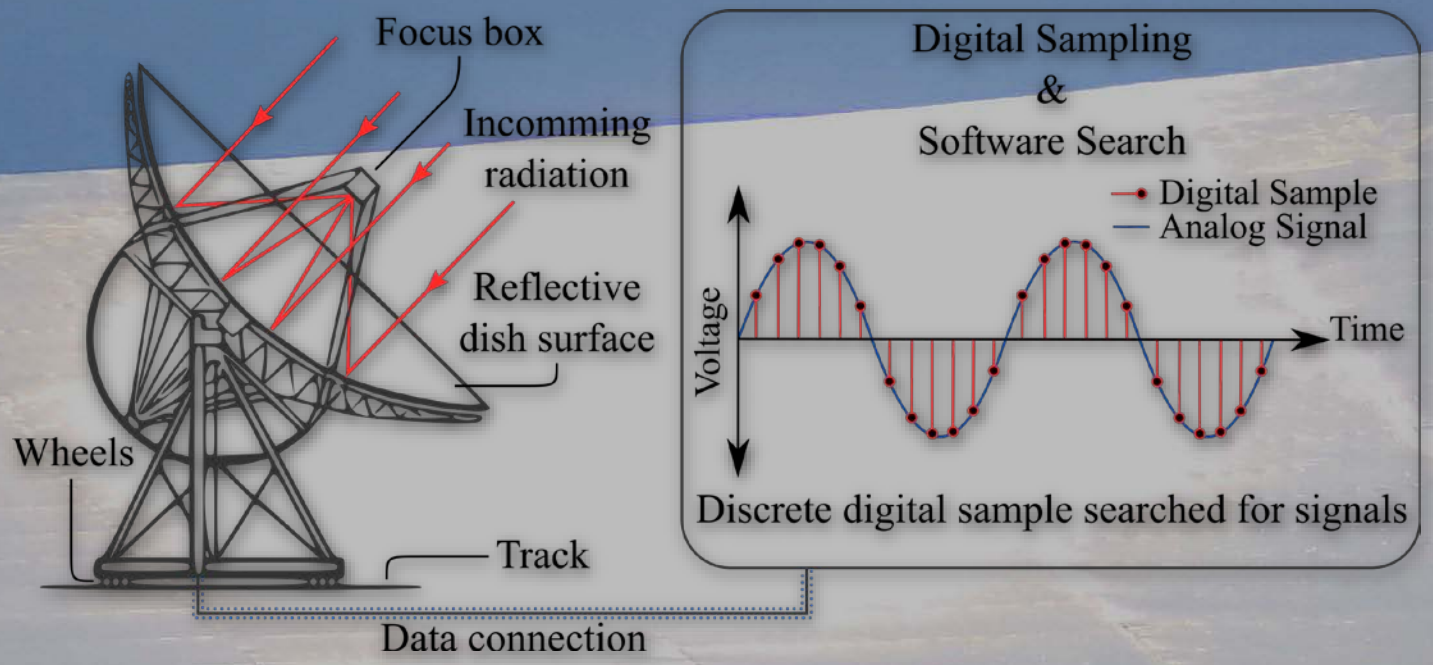


Complex Pulse Profiles

Edge Hill University

# Data Capture

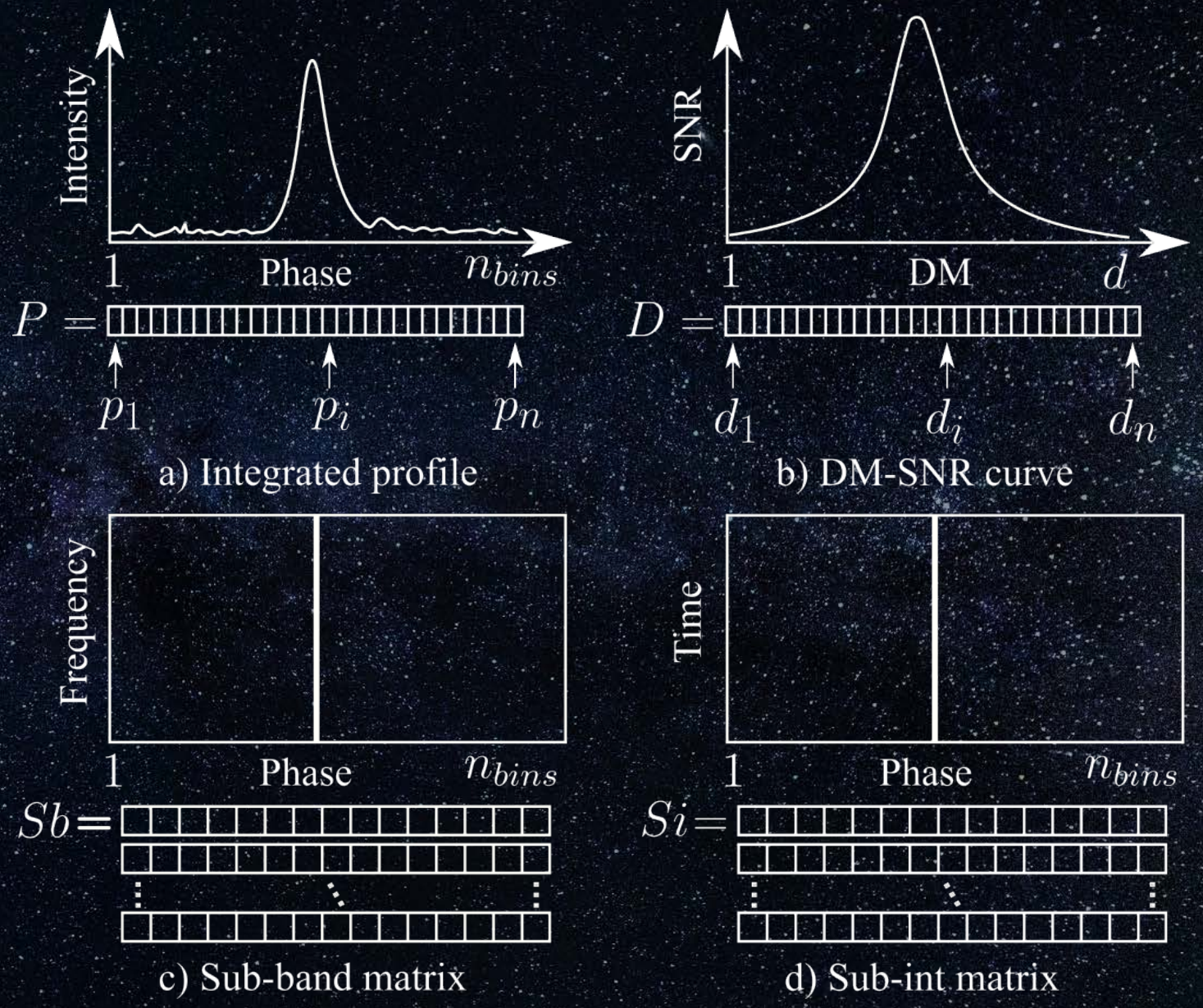
- Analog to digital samples
- Complex search pipeline applied
- Steps:
  - RFI mitigation
  - Dedispersion
  - FFT
  - Harmonic summing
  - Detection
  - Sifting



Edge Hill University

# Data Products

- **A candidate file**
- **Describes a signal detection**
- **Covers time and frequency space**
- **At least one per detection (duplicates!)**
- **Classification appears simple - things do get fuzzy!**



Edge Hill University

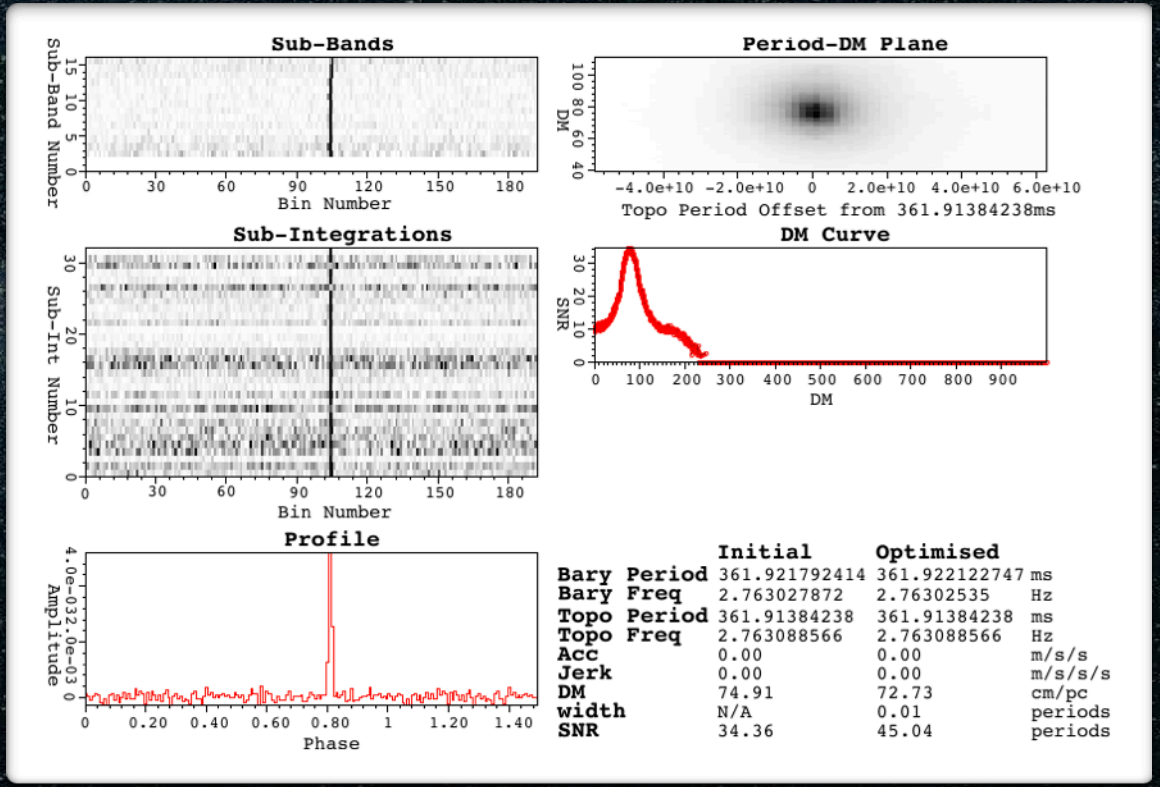
# What to look for?

- Clear pulse with a defined peak.
- Evidence that the signal persists in time, i.e. an indication of a periodic source.
- Evidence that the signal persists in frequency - pulsars are broadband emitters.
- A DM value greater than zero.
- Evidence of other effects, such as scintillation?

Edge Hill University

# Candidate Examples (1)

- Defined peak
- Clear DM-SNR peak
- Consistent in time
- Persistent in frequency
- Obvious pulsar!

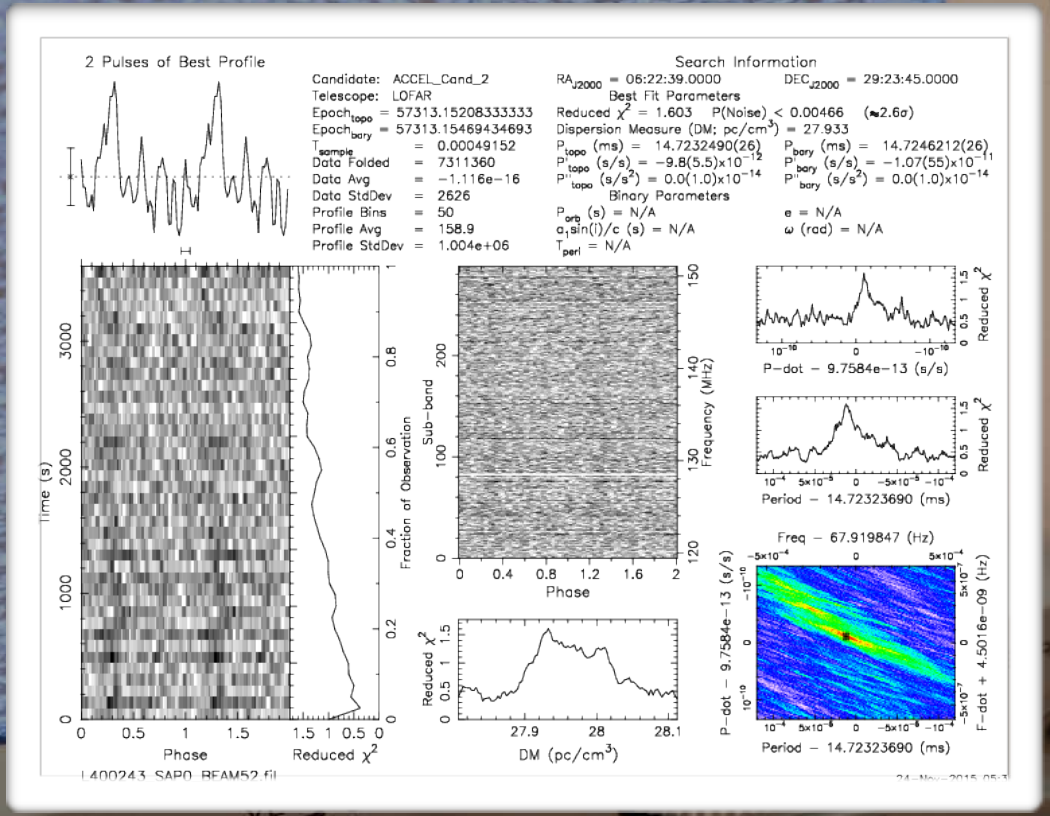



Credit: HTRU Collaboration.

Edge Hill University

# Candidate Examples (2)

- Clearly defined peak?
- Some persistence in time?
- A DM value > zero
- Labelled as noise

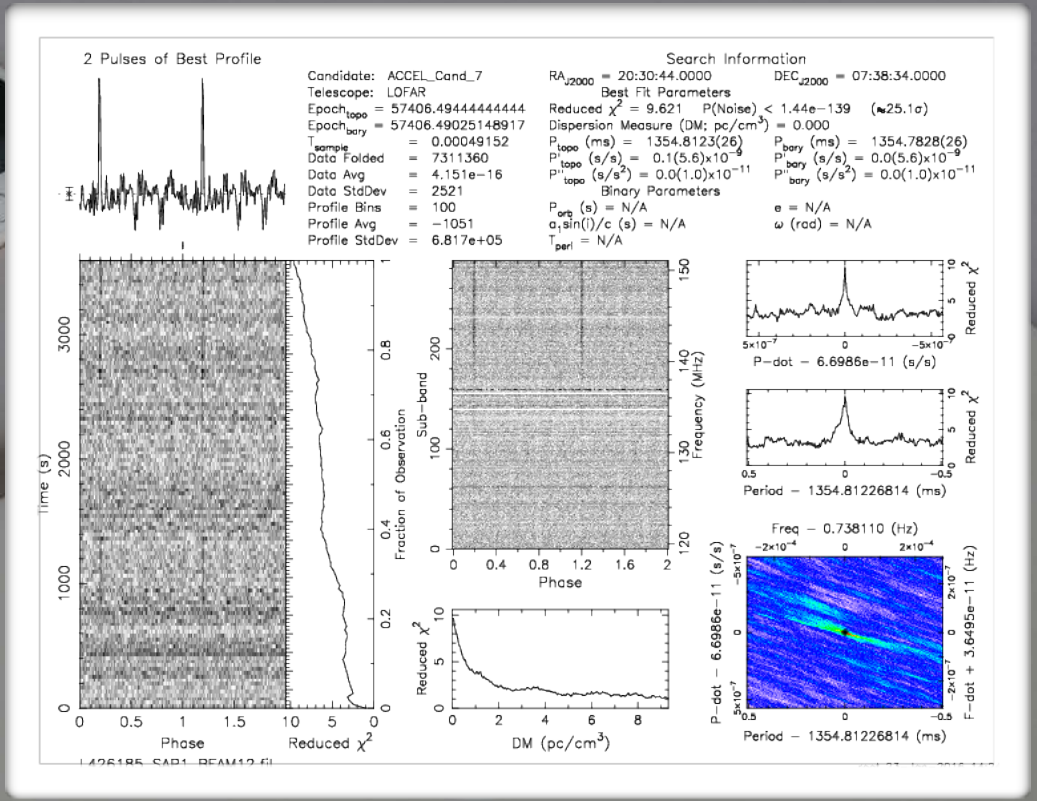



Credit: LOTAAS Collaboration (Chia Min Tan et. al.).

Edge Hill University

# Candidate Examples (3)

- Clearly defined peak?
- Some persistence in time?
- Some persistence in frequency?
- RFI




Credit: LOTAAS Collaboration (Chia Min Tan et. al.).

Edge Hill University

# Things go wrong in practice

- Not all candidates so easily categorised.
- Huge volume of candidates usually returned by telescopes.
- This makes the problem particularly hard to solve.

Edge Hill University

# Imbalanced Data too!

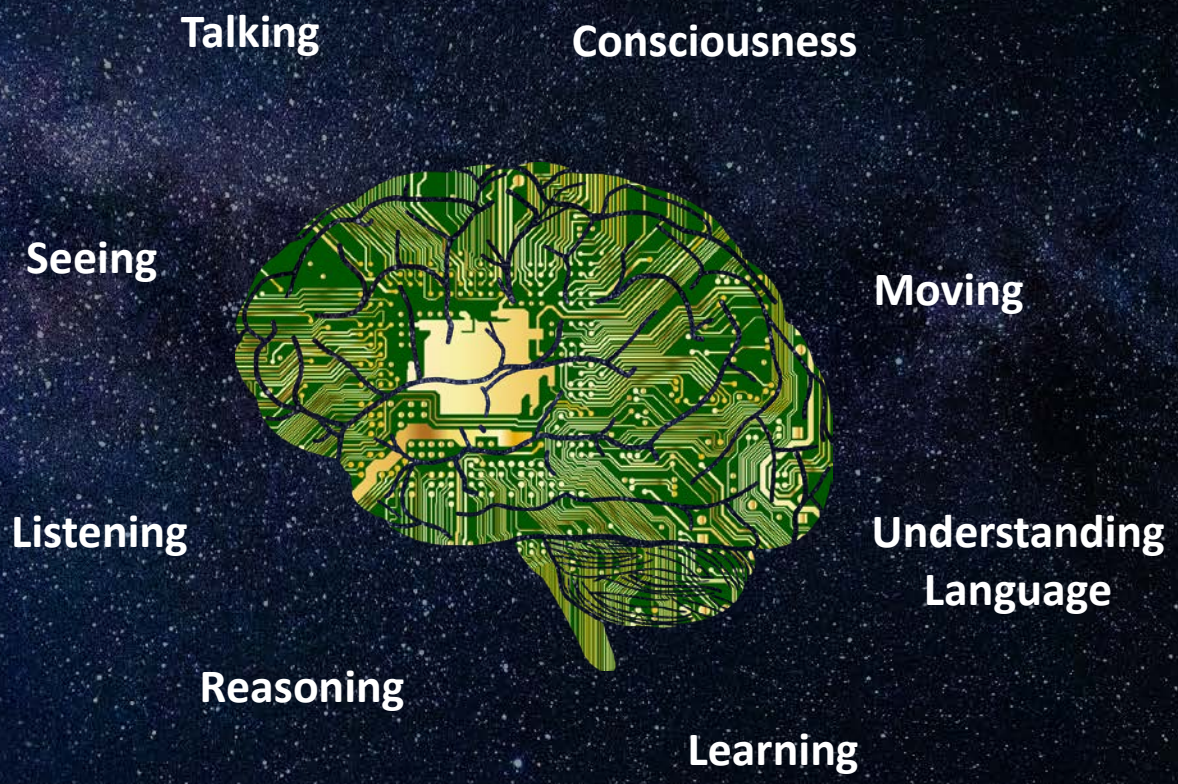
- Candidate ratio is 1 real pulsar to 10,000 non-pulsar examples (not a worse case).
- Given that class distributions drift over time, we have a changing problem too.

Edge Hill University

# Tackling this with A.I.

Edge Hill University

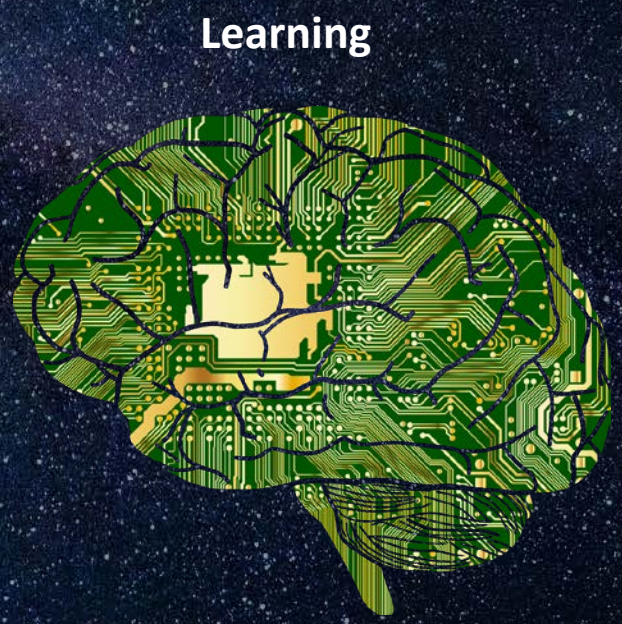
# Artificial Intelligence




- Artificial Intelligence (A.I.) is a field of study concerned with reproducing/replicating human intelligence.
- A.I. is an umbrella term – it is comprised of many sub-fields of study.
- Here we'll talk about replicating the human capacity for learning.
- This field is known more generally as "Machine Learning".

Edge Hill University

# Machine Learning




- Machine learning (ML) is a fascinating field. It combines insights from statistics, logic, psychology and neuroscience to build automated systems capable of “learning” by themselves.
- ML isn’t concerned with replicating exactly how humans learn, after all, we’re flawed!
- Instead it focuses on developing ways to make optimal decisions/predictions using available information.
- During this module you’ll acquire an understanding of ML that will allow you to apply it moving forward.

Edge Hill University

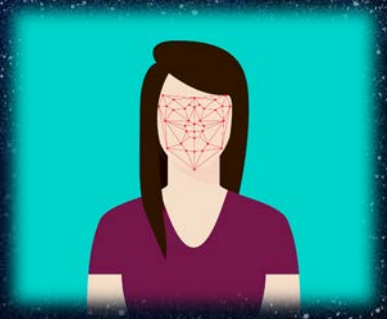
# Example Applications

**Searching for rare stars in astronomy data**

Image Credit: SARAO

**Identifying regions of disease in medical images**

**Climate Modelling**



**Facial Recognition**

**Voice Recognition**

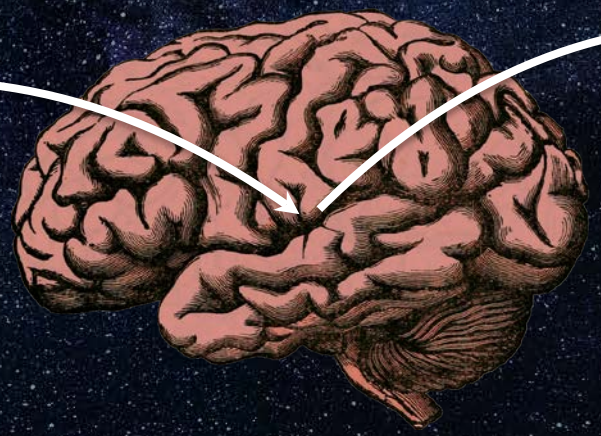
Uncredited images obtained from https://pixabay.com (Creative Commons License, no attribution required).

Edge Hill University

# Human Decision Making

- Using available information / past experience we try to make optimal decisions.

Data (Sight, sound, taste, touch, smell)



Actions /Decisions

Edge Hill University

# Quantifying Experience

- Experience is describable and quantifiable as data.
- You already intuitively understand what data is, but to understand ML, we must explain the terminology used in the field.

Variables / Attributes / Features

Ground Truth

examples

| Mass (Kg) | Height (cm) | Legs | Colour | Class / Label |
|---|---|---|---|---|
| 4.2 | 25.4 | 4 | Black | Cat |
| 18 | 43.5 | 4 | Brown | Dog |
| 0.1 | 5.1 | 4 | Grey/Black | Mouse |
| … | … | … | … | … |

Table 1. Characteristics of animals we've seen

As the ground truth is known, this is called a labelled dataset.

Edge Hill University

# Quantifying Experience

Example notation $x_i$

| | Mass (Kg) | Height (cm) | Legs | Colour |
|---|---|---|---|---|
| $x_1$ | 4.2 | 25.4 | 4 | Black |
| $x_2$ | 18 | 43.5 | 4 | Brown |
| $x_3$ | 0.1 | 5.1 | 4 | Grey/Black |

| Class / Label $y$ | |
|---|---|
| Cat | $y_1$ |
| Dog | $y_2$ |
| Mouse | $y_3$ |

Example notation $y_i$

Table 1. Characteristics of animals we've seen

Variables / Attributes / Features

Ground Truth

Array

$$x_1 = \{4.2, 25.4, 4, Black\} \qquad y_1 = \{Cat\}$$

$$x_2 = \{18, 43.5, 4, Brown\} \qquad y_2 = \{Dog\}$$

$$x_3 = \{0.1, 5.1, 4, Grey/Black\} \qquad y_3 = \{Mouse\}$$

Edge Hill University

# Quantifying Experience

**Variables / Attributes / Features**

$$x_1 = \{4.2, 25.4, 4, Black\}$$

$$x_2 = \{18, 43.5, 4, Brown\}$$

$$x_3 = \{0.1, 5.1, 4, Grey/Black\}$$

**Ground Truth**

$$y_1 = \{Cat\}$$

$$y_2 = \{Dog\}$$

$$y_3 = \{Mouse\}$$

$$E = \{(x_1, y_1), (x_2, y_1), \ldots, (x_n, y_1)\}$$

- **Experience is comprised of pairs of feature values and ground truth labels.**
- **The pairs are actually called "tuples".**

Edge Hill University

# Sometimes Knowledge is Incomplete

- Sometimes we don't know the ground truth. In such cases, we have to collect information, and provide it for ourselves.
- This can be costly and time consuming.

| | Variables / Attributes / Features | | | | Ground Truth | |
|---|---|---|---|---|---|---|
| | Ext. Temp (℃) | Int. Temp (℃) | Time (24hr) | Consumption (kWh) | Class / Label $y$ | |
| $x_1$ | 4.2 | 17.1 | 15:23 | 7500 | ? | $(x_1, ?)$ |
| $x_2$ | 33.4 | 27.8 | 18:01 | 7000 | ? | $(x_2, ?)$ |
| $x_3$ | 10.8 | 19.1 | 20:36 | 450 | Deactivated | $(x_3, ?)$ $(x_3, Deactivated)$ |
| | … | … | … | … | … | |

Table 2. Status of a heating control system

- If the ground truth is unknown, this is called an "unlabelled" dataset.
- Datasets may also be "partially-labelled".

Edge Hill University

# Feature Values

- Features can be categorical or numerical.
- Usually we "discretise" categorical features as this makes them easier for many algorithms to work with. This is either done automatically using tools, or manually.

| Mass (Kg) | Height (cm) | Legs | Colour |
|---|---|---|---|
| 0.229 | 25.4 | 4 | 1 |
| 1.0 | 43.5 | 4 | 2 |
| 0 | 5.1 | 4 | 4 |
| … | … | … | … |

| Class / Label $y$ |
|---|
| 1 |
| 2 |
| 3 |
| … |

Table 1. Characteristics of animals we've seen

- We may also "normalise" our data from time to time.

Edge Hill University

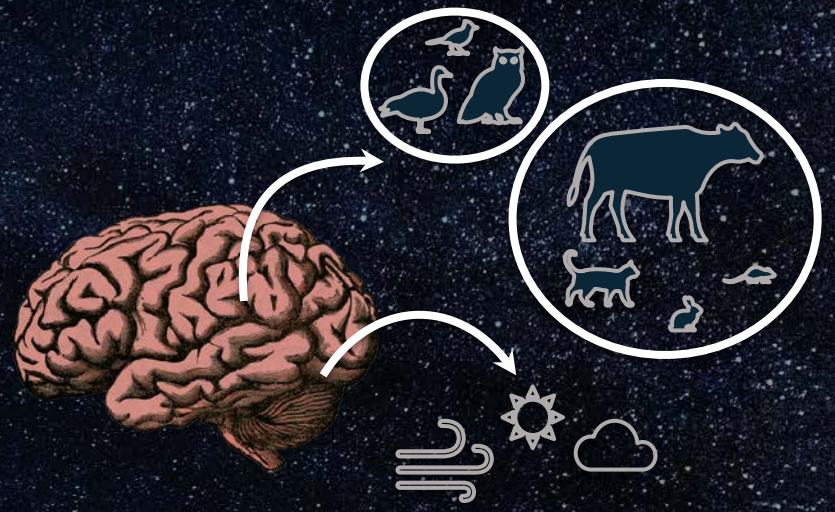
# Human Decision Making

**Features (Variables)**

$$x_i = \{ \text{Height, Mass, } \ldots, \text{Age} \}$$

**Labels ( Feedback )**

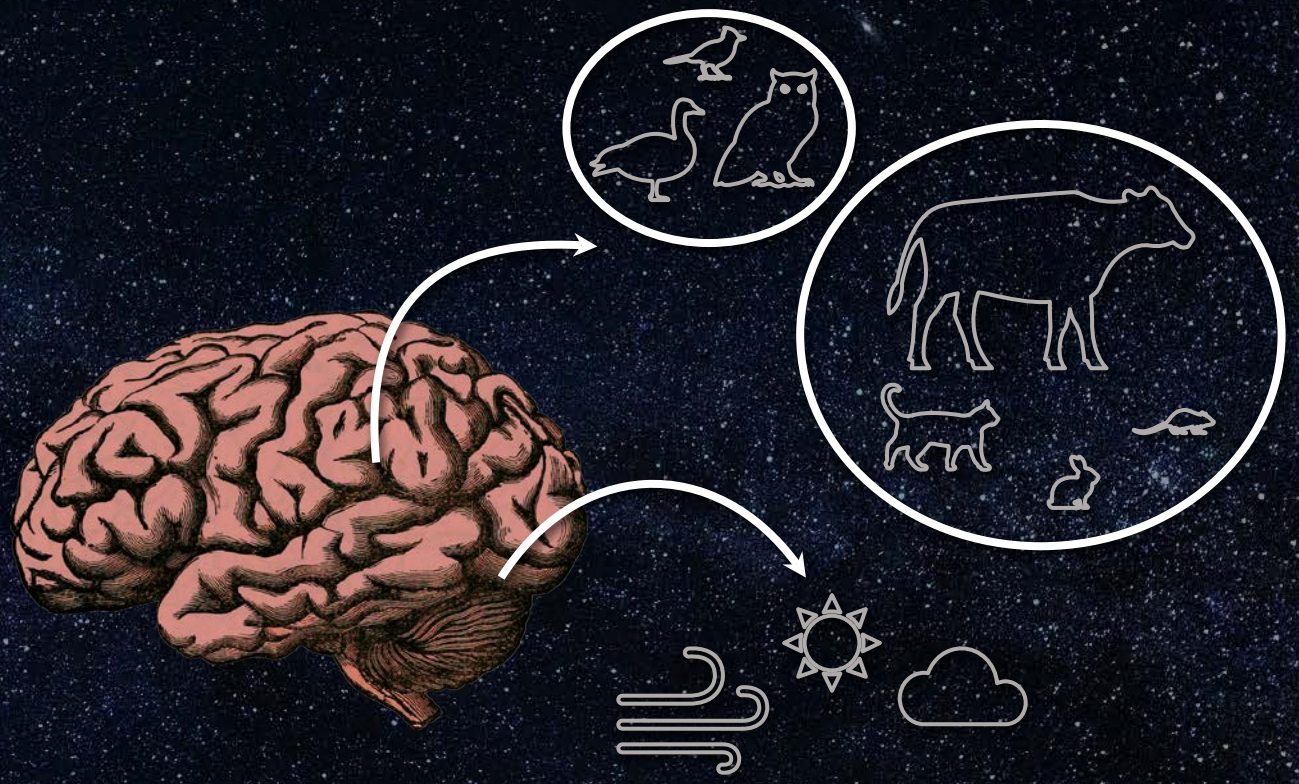
$$y_i = \{Cat, Dog, \ldots, Bird\}$$



Edge Hill University

# Human Decision Making

- **Humans capable of accurately,**
  - **Clustering**
  - **Classifying**
  - **Predicting**
- **Made possible via a feedback cycle.**



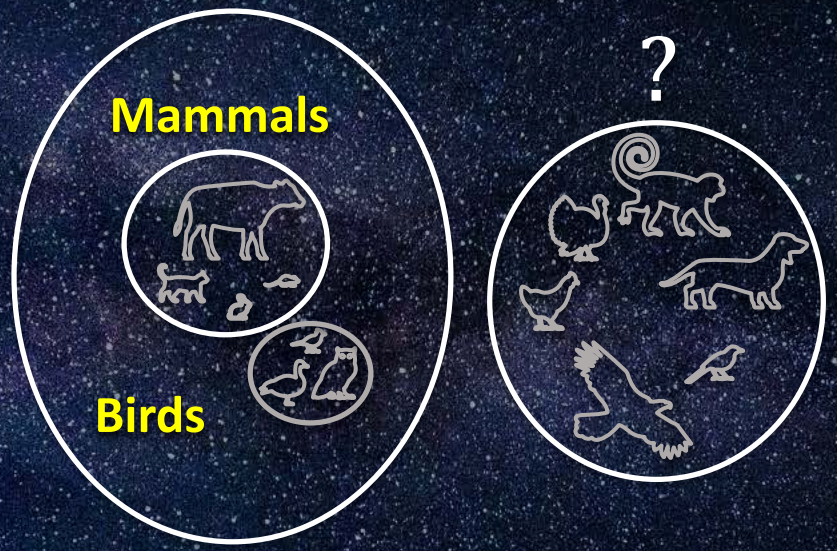
$$E = \{(x_1, y_1), (x_2, y_1), \ldots, (x_n, y_1)\}$$

**Feedback**

Edge Hill University

# Approaches to Learning

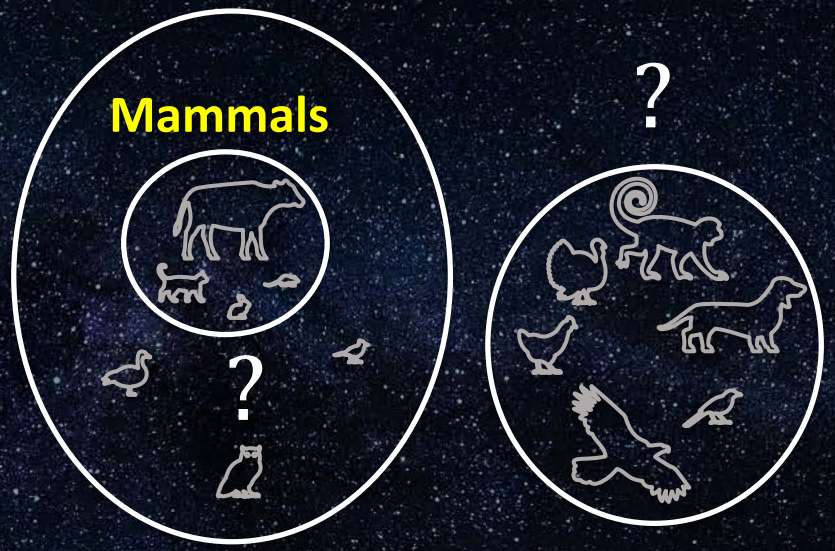
## Supervised Learning




$E$ Contains labels for mammals & birds, goal is to correctly group unseen animals into mammal & bird classes.

Labeled data

## Semi-supervised Learning




$E$ Contains examples labels for mammals only, goal is to correctly group unseen animals into mammal & bird classes.

Partially-Labeled data

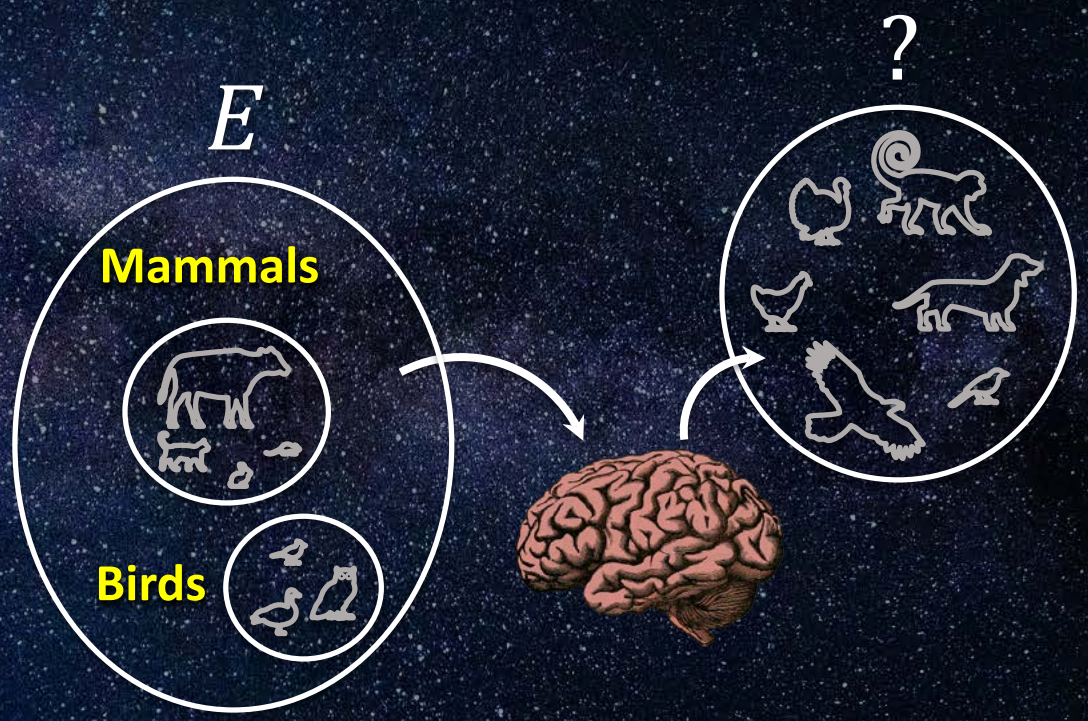
## Unsupervised Learning

$E$ Contains no labels. Instead we use the data to self describe class separation.

Unlabeled data

Edge Hill University

# Classification




- These problems are “binary” classification tasks.
- Called binary, as there are two options (mammal or bird).
- Such classification tasks are everywhere, and we do them every day.
- To complete them we make predictions.
- Classification problems in the real-world are usually far more complex, i.e. more than 2 potential classes to predict.
- These are known as multi-class problems. Let’s try one.

Edge Hill University

# Example

Feline

Lutrinae
(e.g. Otter)

Canine

Mongoose

Viverridae
(e.g Civet)

Image Credit: Ran Kirlian - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=4164754
Uncredited images obtained from https://pixabay.com (Creative Commons License, no attribution required).

Edge Hill University

# Could you generalise?

- In the last slide, we saw an animal you may not be familiar with.
- Even if you have extensive knowledge of the animal kingdom, you may not have known the animal was a Fossa.
- Perhaps this animal was not described in the dataset, $E$ .
- Even if you did not know it was a Fossa, you could still tell it was a type of mammal.
- This application of your knowledge, from past experience, illustrates your ability to generalise beyond known facts.
- Humans are gifted generalisers - but they are susceptible to two problems:

**Overfitting** **Underfitting**

Edge Hill University

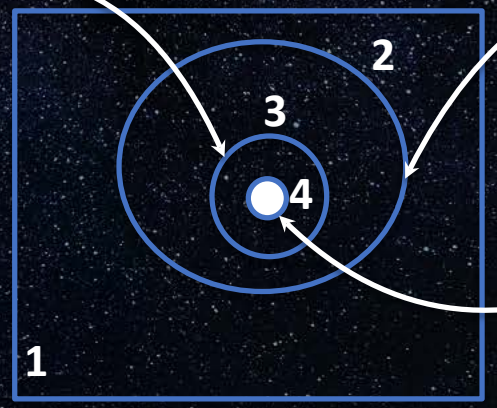
# Making Optimal Decisions

- Many have tried to understand how to make optimal decisions.
- We know we should use available evidence at all times.
- Humans are not always so thorough - we do make bad decisions.
- We are biased decision makers - we often use instinct and personal experience to decide.

Personal Knowledge & Experience

$E$

Set of all knowledge on a subject




Available Knowledge & Experience

Knowledge used to make a decision

Edge Hill University

# So what is Automated Learning?

Edge Hill University

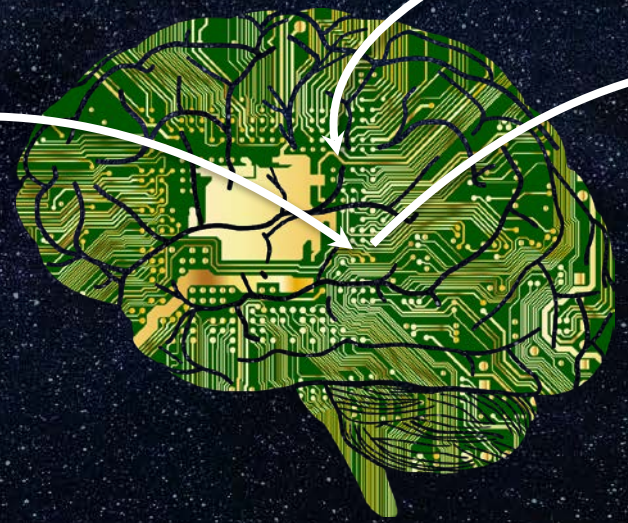
# Automated Decision Making

- Machine Learning is concerned with making optimal decisions primarily using the tools of statistics.

Biased cognitive process

Biased input data

Data (feature values, measures, outcomes…)



Decisions & Predictions

$$E = \{(x_1, y_1), (x_2, y_1), \ldots, (x_n, y_1)\}$$

Edge Hill University

# Automated Decision Making's Flaws

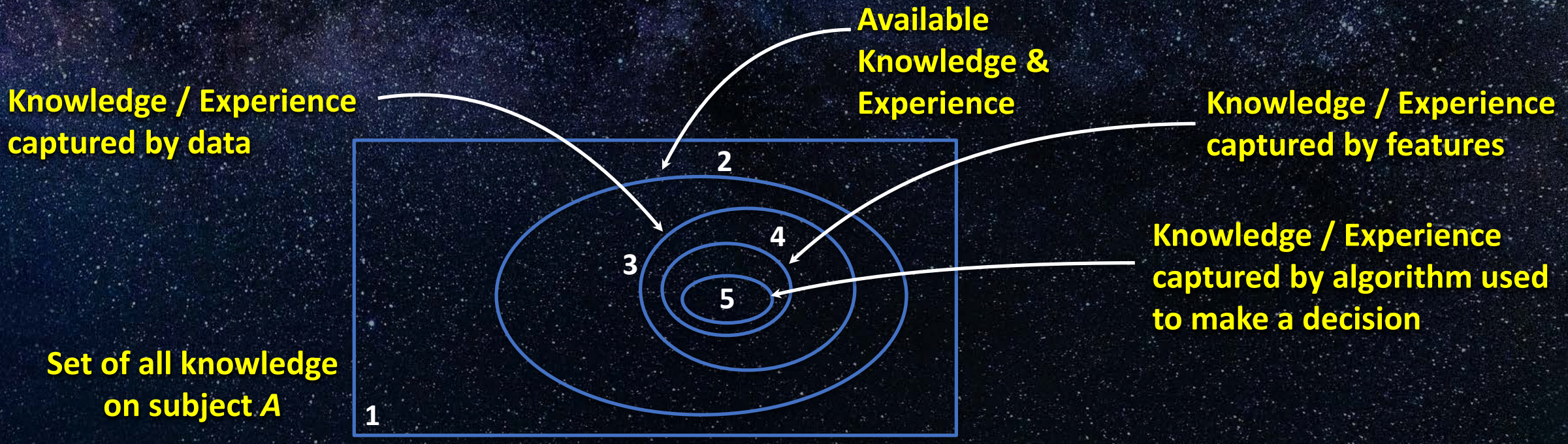
- **Biased input data** - For example, suppose you want to teach an algorithm to recognize a specific disease in a group of patients.
- You provide the algorithm with data describing patients who are all aged between 60 to 80.
- You then run the algorithm on patients aged 18 to 30. The algorithm performs badly, as it's knowledge is biased toward recognizing disease in much older patients.

- **Biased cognitive process** - For instance suppose we try to teach an algorithm to predict when a train on the Tokyo rail network will be late.
- Trains on this network are exceptionally punctual.
- To achieve the best overall performance, just never predict that a train will be late.

Edge Hill University

# Making Optimal Automated Decisions

- Algorithms can process more quantifiable data than humans.
- However the data may be biased/incomplete.
- Individual algorithms also have intrinsic biases.
- Algorithms can therefore be just as fallible as humans!



Edge Hill University

# From Experience to Training Data

- Experience is known as "training data".
- An algorithm is ‘taught’ using training data.
- Training data can be labelled or unlabelled.
- We evaluate performance on a disjoint dataset called "test data" - this data must be labelled (ground truth known).

$$Training\ data = \{(x_1, y_1), (x_2\ , y_1), \ldots, (x_n, y_1)\}$$

Edge Hill University

# Training vs Test Data

**There is an easy way to understand the difference between training and test data.**

- **Training data represents the knowledge given before an exam.**
- **Test data is a disjoint set of information, that represents the exam – we use this to test performance.**

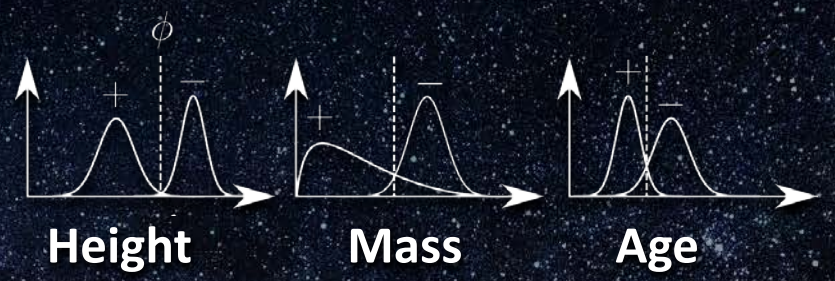
**Set of all knowledge on subject *A***

Edge Hill University

# Features & Labels Revisit

**Features (Variables)**

$$x_i = \{x_i^1, \ldots, x_i^m\}$$




**Labels ( Feedback )**

$$y_i \in \{-1, 0, 1\}$$

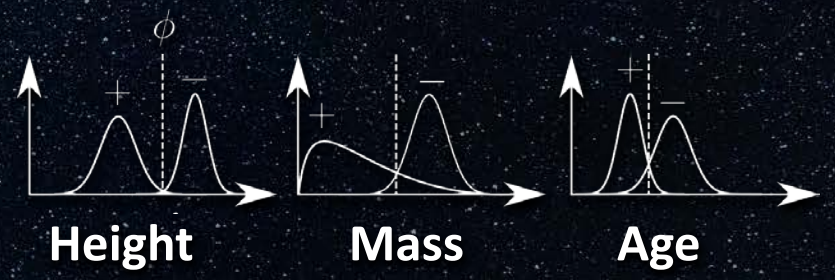
**e.g. -1 = no disease etc**

$$Training\ data = \{(x_1, y_1), (x_2, y_1), \ldots, (x_n, y_1)\}$$

Edge Hill University

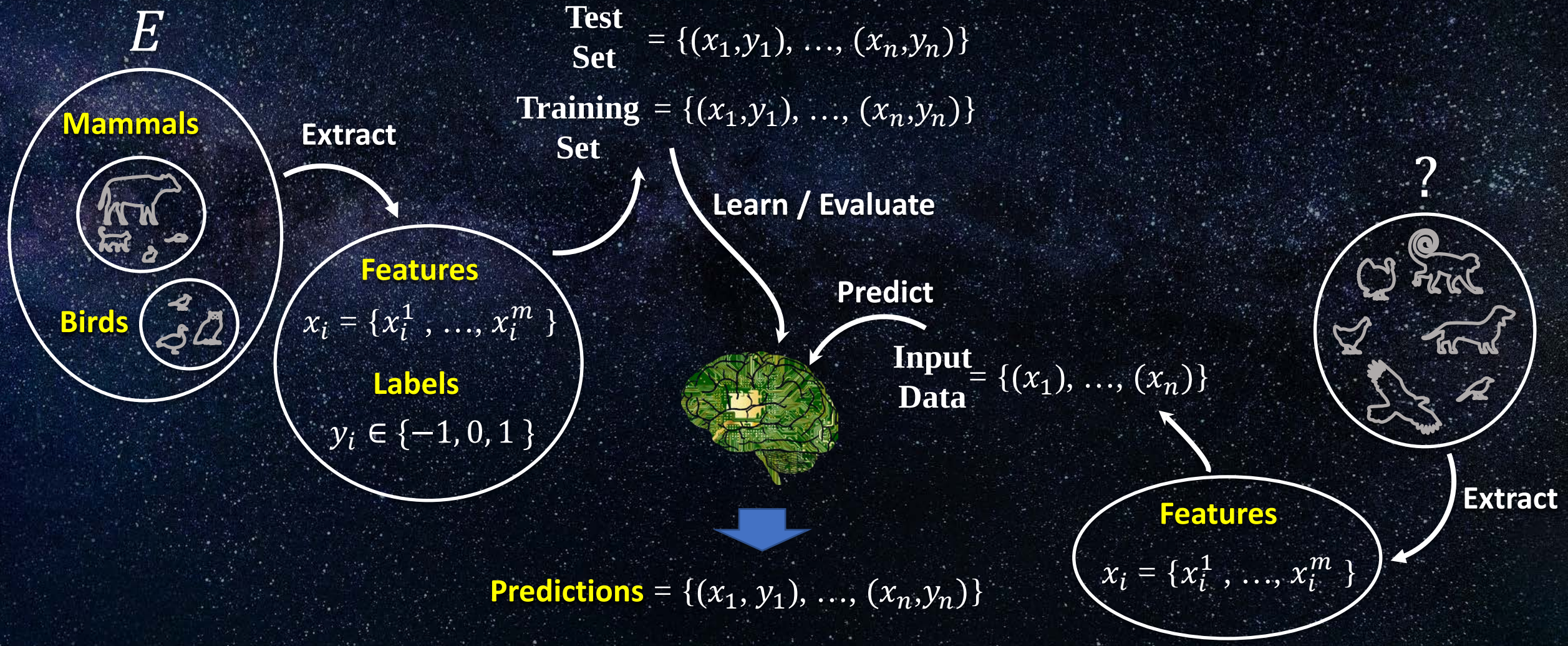
# Feature Design

- In principle feature design involves studying your data.
- Considering it's properties.
- Extracting information you believe will help with class separation.
- In practice this involves,
  1. considering as many candidate features as possible.
  2. Considering their useful ness in turn.
  3. Selecting a sub-set of features you think will be effective.
  4. Testing those, see what happens.
  5. Return to step 1 if results are not as expected.
  6. Collect new data?
  7. Deriving new features?



Edge Hill University

# Classification Process



Edge Hill University

# So How is “Learning” Done?

- **Learning via trial and error – learning from mistakes made.**

**We need to be at work, but don’t know what time the bus arrives to get us there. Work starts at 09:00, and the journey takes roughly 40 minutes.**

**We start to wait at the bus stop each day, recording some details.**

| Day | Arrived at bus stop | Bus arrived |
|---|---|---|
| Mon | 08:32 | 08:45 |
| Tue | 08:17 | 08:30 |
| Wed | 08:12 | 08:15 |

| Late for work $y$ |
|---|
| Yes |
| Yes |
| No |

**In this case learning involves finding an arrival time at the bus stop, that lets you make it to work on time. Each time you’re late, you’ve made an error that you learn from.**

Edge Hill University

# So How is “Learning” Done?

- In ML error is quantified and minimised using the tools of mathematics.
- This sort of error minimisation is done using functions.

We can reduce finding the best time to wait at the bus stop, to a mathematical problem, i.e. find a value for $t$ that minimizes the number of times you’re late.

Many potential values for $t$ work, but clearly it must be 08:15 or earlier!

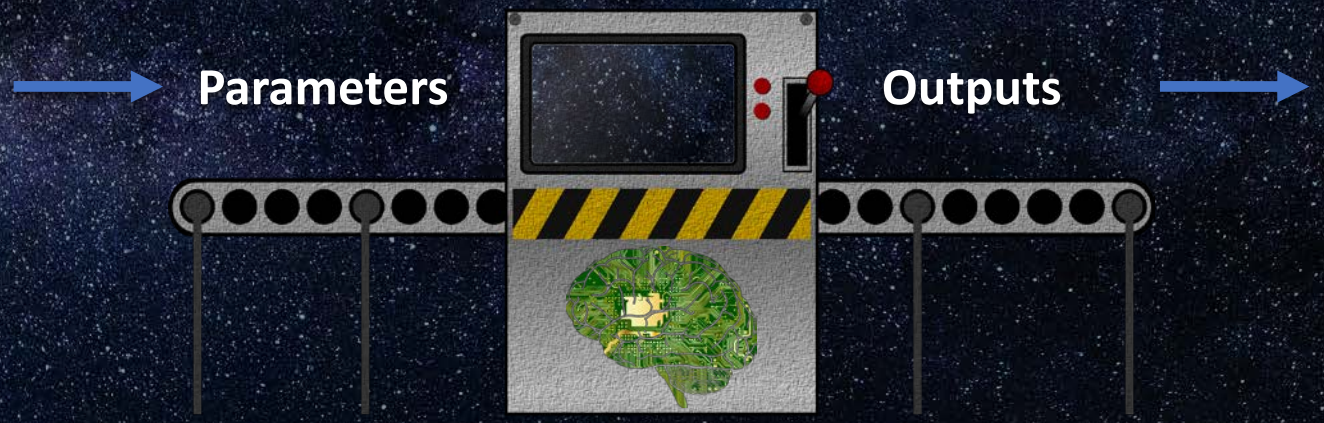
| Day | Arrived at bus stop $t$ | Bus arrived |
|---|---|---|
| Mon | 08:27 | 08:45 |
| Tue | 08:17 | 08:30 |
| Wed | 08:12 | 08:15 |

| Late for work $y$ |
|---|
| Yes |
| Yes |
| No |

Edge Hill University

# Functions

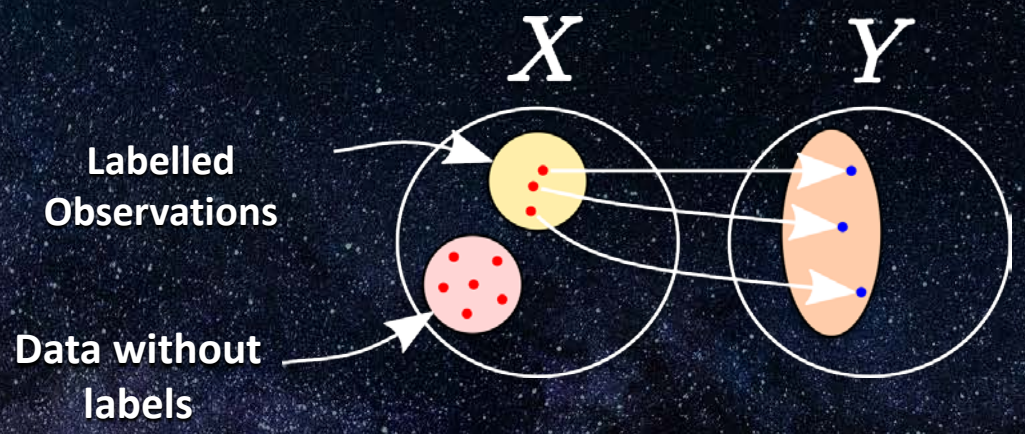
- Functions can be thought of as simple input/output boxes.
- Machine learning algorithms are functions that ingest data, and produce some output.
- How they do this is hard to convey without some basic mathematics!




$f(x, y) = x + y$ e.g. $f(x = 2, y = 4) = 2 + 4 = 6$

Edge Hill University

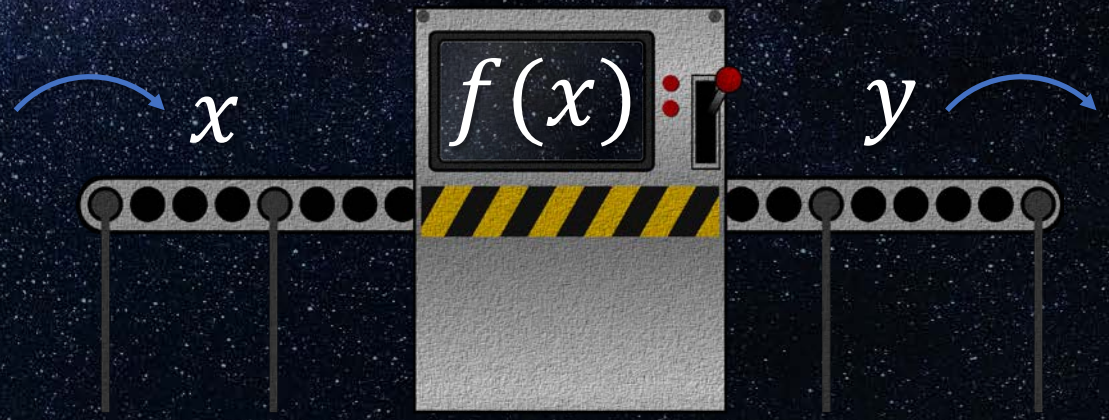
# Functions That "Map"




**E.g.**

$$f(x) = 0, if\ feature_1 \leq 0$$
$$1, otherwise$$

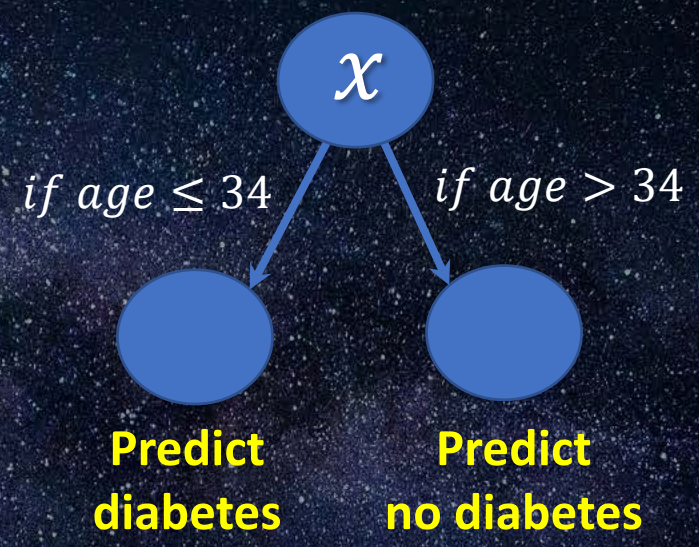
| $feature_1$ | $feature_2$ |
|---|---|
| 7.4 | 29 |
| 6.6 | 24 |
| 3.1 | 11 |
| … | … |
| 0.5 | 2 |

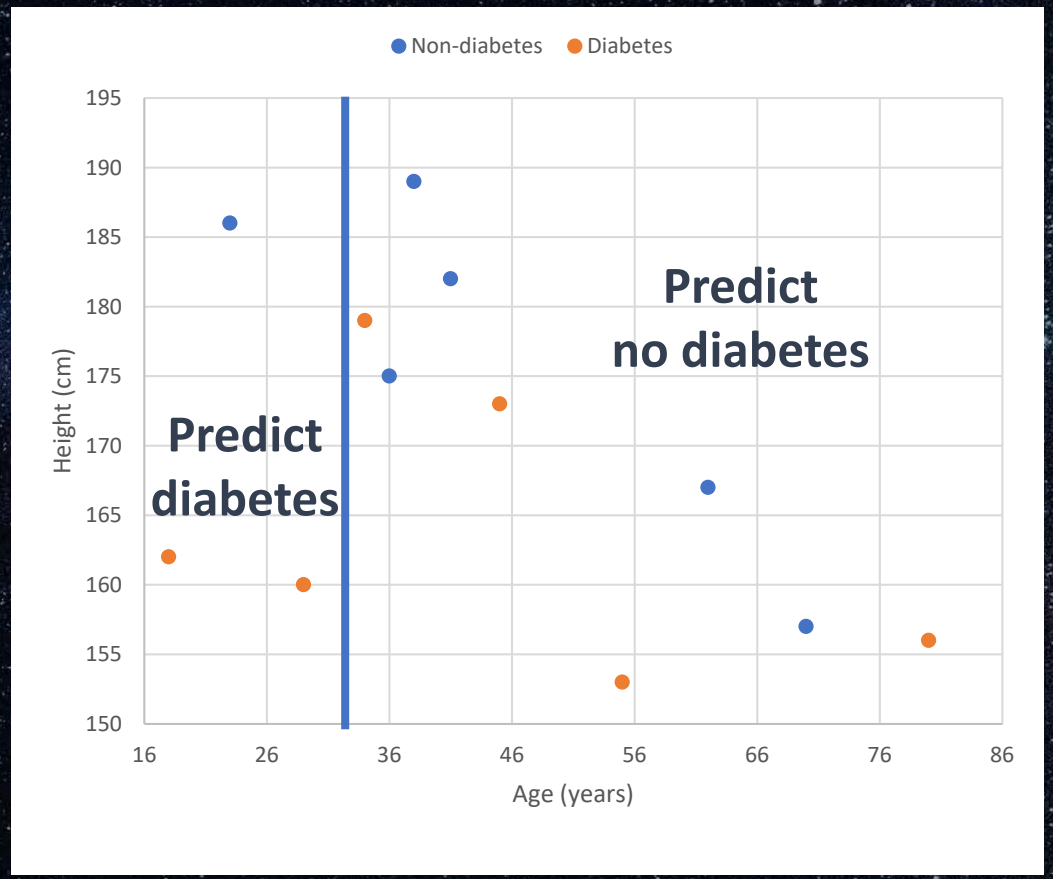



| $y$ |
|---|
| 1= Diabetic |
| 1= Diabetic |
| 0 = Non-diabetic |
| … |
| Unknown |

Edge Hill University

# Decision Stump

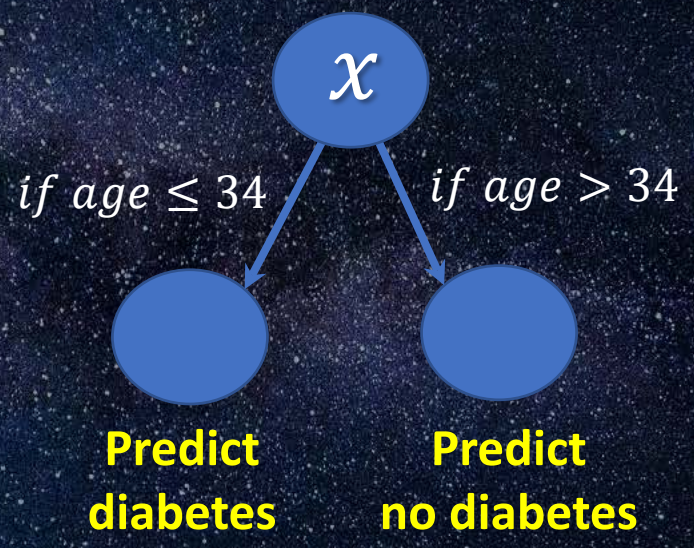



- We've just covered our first ML algorithm - the "Decision stump".
- It is a linear separator, or a linear model.
- It simply looks for a feature to split on, in a optimal way.
- It produces a linear separator between two classes.

Edge Hill University

# Decision Stump

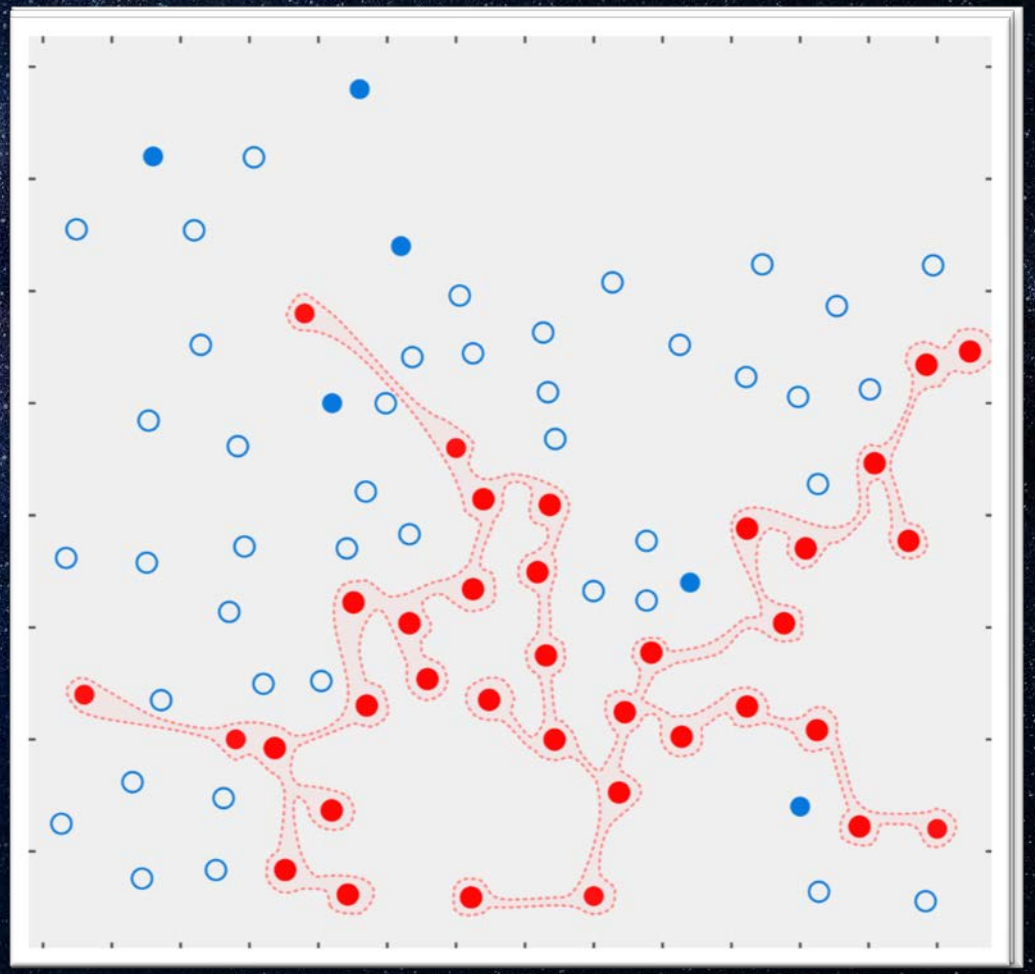



- For each feature in the data, the stump will search for a split-point, that minimises the error rate.
- In our example we used age as a feature, and our threshold = 34.

Edge Hill University

# Decision Boundary

Shaded region predicts diabetes



- You've now seen a linear model in action, and learned what a decision boundary is.
- Real-world problems are complex - so too are the decision boundaries needed to accurately separate data.
- This means linear models don't always work well. More complex ML algorithms are required.
- When we generalise well we can produce effective decision boundaries.
- However sometimes we don't perform well in practice.
- We can underfit to our data.
- We overfit to our data.
- Decision boundaries allow us to visualise this happening.

Edge Hill University

# Learning in Name Only…

- What we've talked about so far - this isn't really learning right?
- It's a search for parameters that minimise some error rate.
- Learning effectively reduced to a parameter search!
- When this works well it provides the illusion of intelligence – but this isn't how you or I learn.

Edge Hill University

# Classification Algorithms

- Classification algorithms come in all shapes and sizes.
- Inside they are comprised of functions, that attempt to perform the mapping from $x$ to $y$ in different ways.
  (Examples to labels)

- What they have in common:

All are evaluated against some ***feedback*** metric, which guides their learning - call this the error rate.

They must choose (find) parameter values that minimise the error rate.

Edge Hill University

# Classification Algorithms

Edge Hill University

# Types of Classifier

## Discriminative

- Learns the differences between the examples in a training set.
- Use those differences to classify and make predictions.
- We often think similarly, like in the Fossa example.

## Generative

- Learns a “model” of the classes in the training set.
- This model can be then compared against.
- We don’t usually think this way, e.g. given all the cats we’ve seen, what is the probability that this animal is a cat?

Edge Hill University

# Generative Decision Making

**Probability of mystery disease**

$P(disease) = 0.1$

**Probability of red spotty rash**

$P(symptom) = 0.1$

**Probability of red rash for those with disease**

$P(symptom|disease) = 0.8$

$$P(hypothesis|data) = \frac{P(data|hypothesis) \times P(hypothesis)}{P(data)}$$

**Bayes' Theorem**

$$P(disease|symptoms) = \frac{0.8 \times 0.1}{0.1}$$

$$= 0.80 = 80\%$$

**So the probability of having the mystery disease is 80%, if you have a red rash.**

Edge Hill University

# Generative Decision Making

**Probability of mystery disease**

$P(disease) = 0.1$

**Probability of red spotty rash**

$P(symptom) = 0.1$

**Probability of red rash for those with disease**

$P(symptom|disease) = 0.8$

**40 to 60 year old's**

$P(disease) = 0.001 = 0.1\%$

**Mystery disease = Chicken pox!**

$$P(disease|symptoms) = \frac{0.8 \times 0.001}{0.1}$$

$$= 0.8\%$$

**This shows that the data used during training, must be representative of the data to be faced in the real-world.**

**We've actually just learned our next ML algorithm - Naïve Bayes. It makes predictions using probability.**

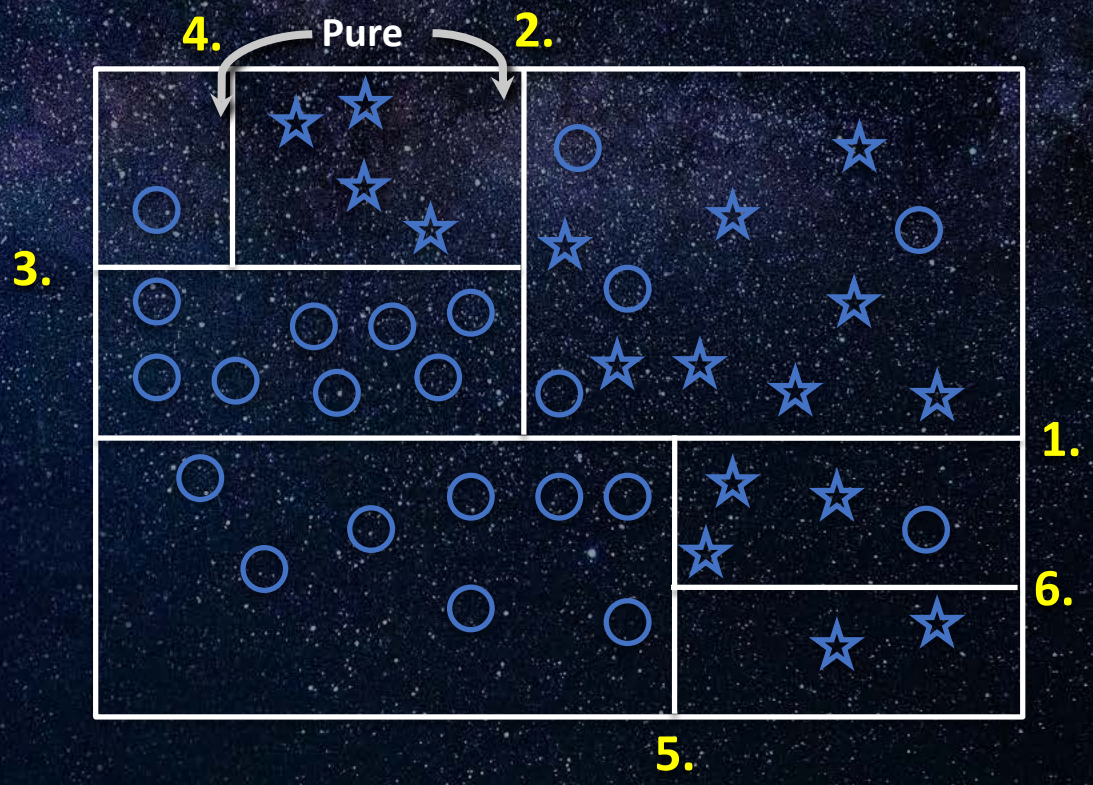
Edge Hill University

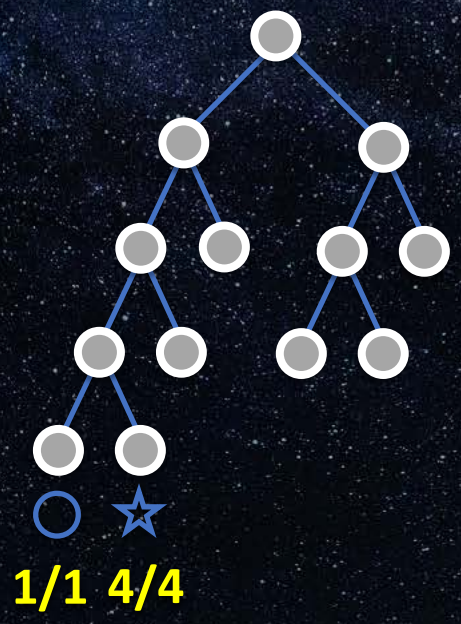
# Tree Learning – Discriminative Model

Suppose we have a dataset containing 2 classes:

- Stars
- Circles

Goal is the separate them as accurately as possible.

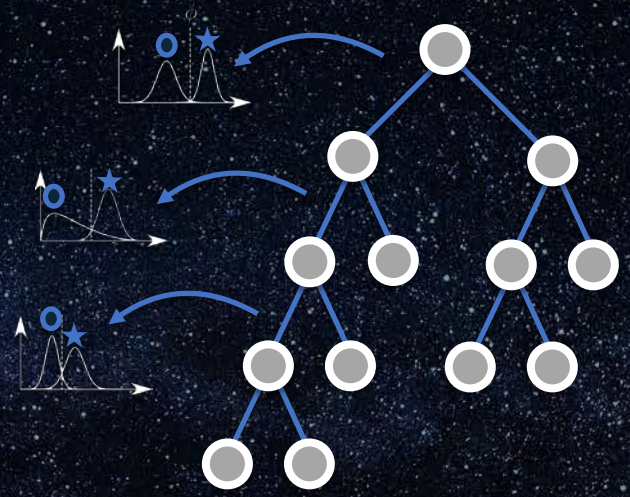


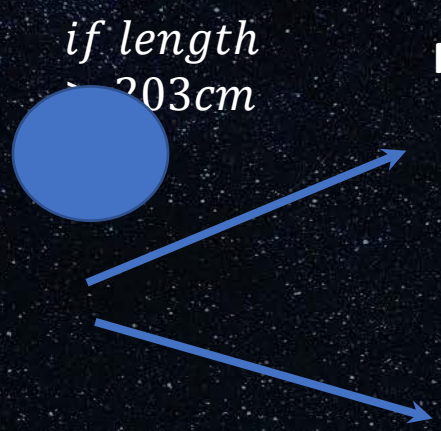

- Non-linear model.
- Successively discriminates into classes via partitioning the data.
- Each partition forms an extra decision boundary.
- Key question – which feature to split on at any given time?

Edge Hill University

# Tree Split Choice



- At each spilt point, find the feature that maximises the separation between each class.
- Done by considering the statistical distribution of the features.
- Via an optimisation method (search over the possible split-points, find the best).



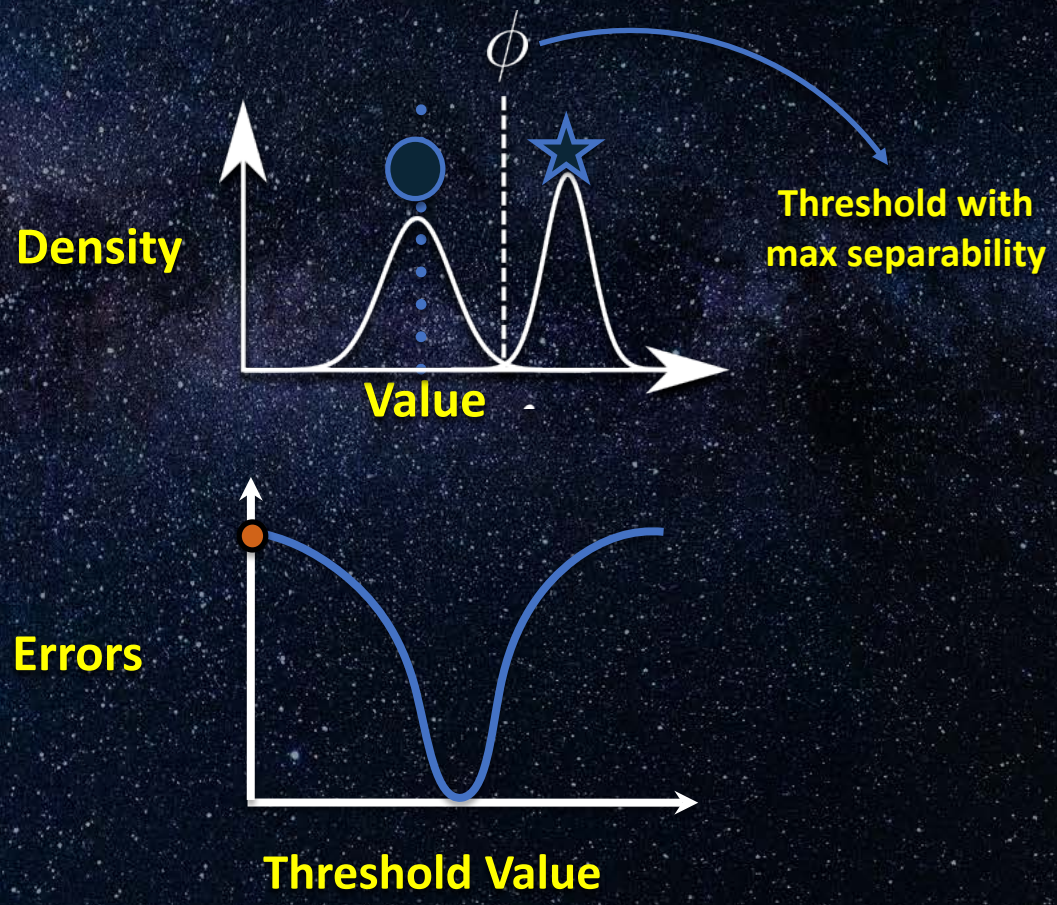

Mountain Lion

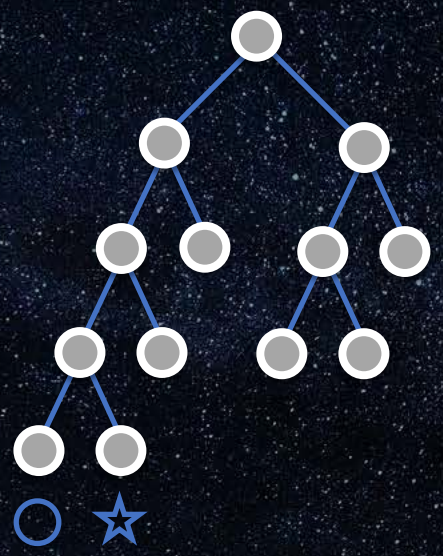
Jaguar

Mass roughly equal

Edge Hill University

# Splitting Guided by Error

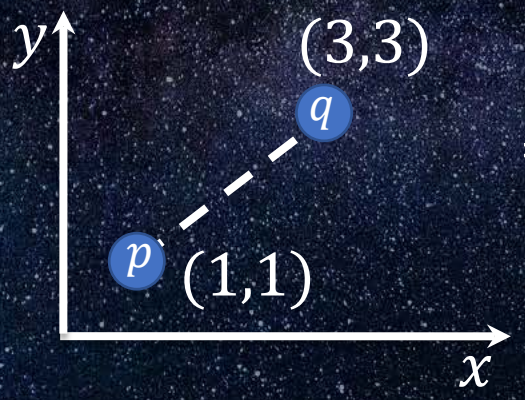


Edge Hill University

# Unsupervised learning

- To make a prediction for a data point, unsupervised algorithms try to group them together with the points closest to them in the training set – i.e. group according to the "nearest neighbors".
- Closest / nearest neighbours are computed using distance measures e.g. Euclidean distance:




$$d(p,q) = \sqrt{(q_1 - p_1)^2 + \ldots + (q_n - p_n)^2}$$

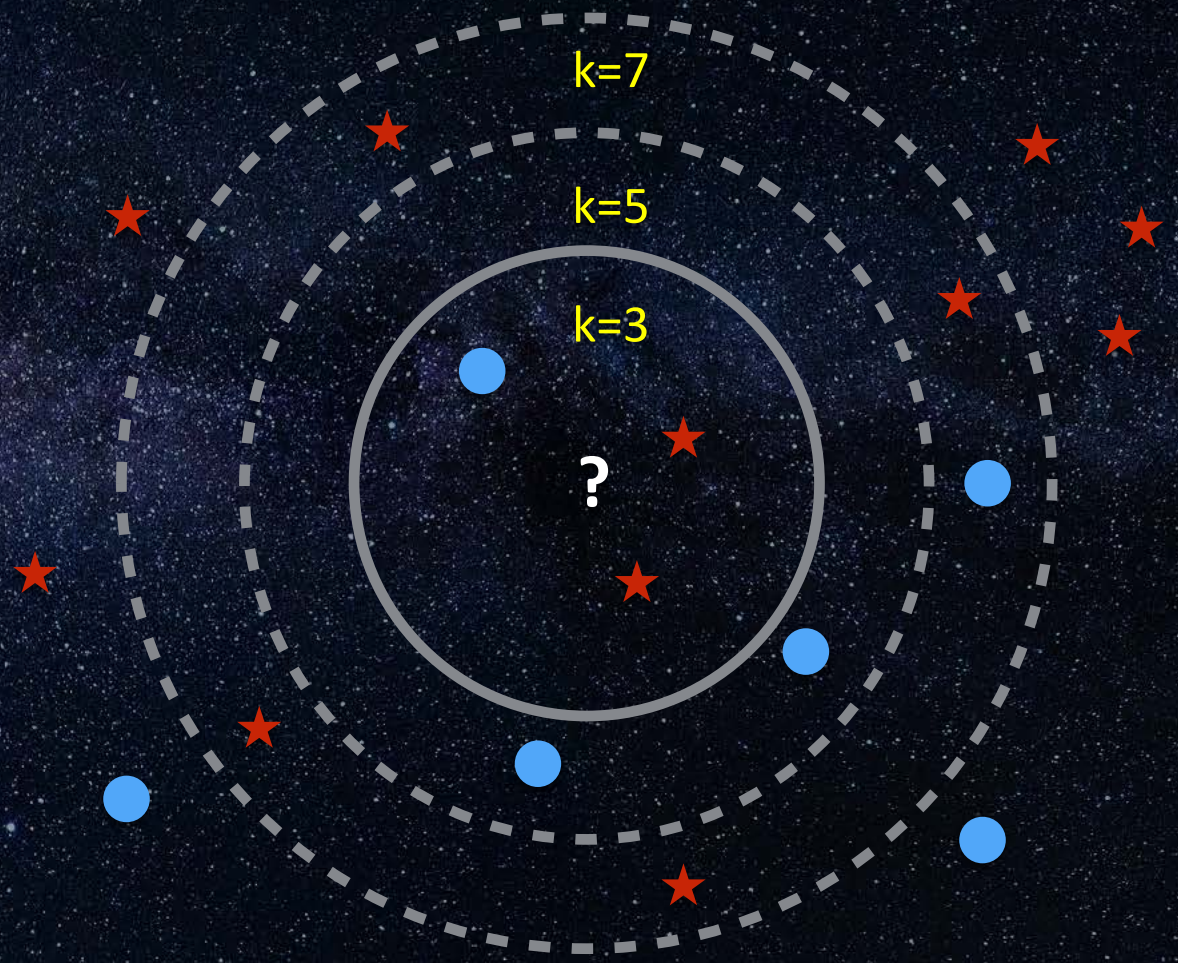
$$= \sqrt{(3-1)^2 + (3-1)^2}$$

$$= \sqrt{(2)^2 + (2)^2} = \sqrt{4+4} = 2.828 \text{ to } 3.\text{d.p}$$

- Once we find the nearest neighbours of a data point via a method like above, we determine the most popular label amongst it's nearest neighbours.
- We assign the majority label of the neighbours to the unlabeled data point.
- We need to determine how many of it's nearest neighbours, $\boldsymbol{k}$, we should use to decide.

Edge Hill University

# $K$-nearest Neighbours




| K value | Prediction |
|---|---|
| 3 | ★ |
| 5 | ● |
| 7 | ★ |

Edge Hill University

# Evaluation

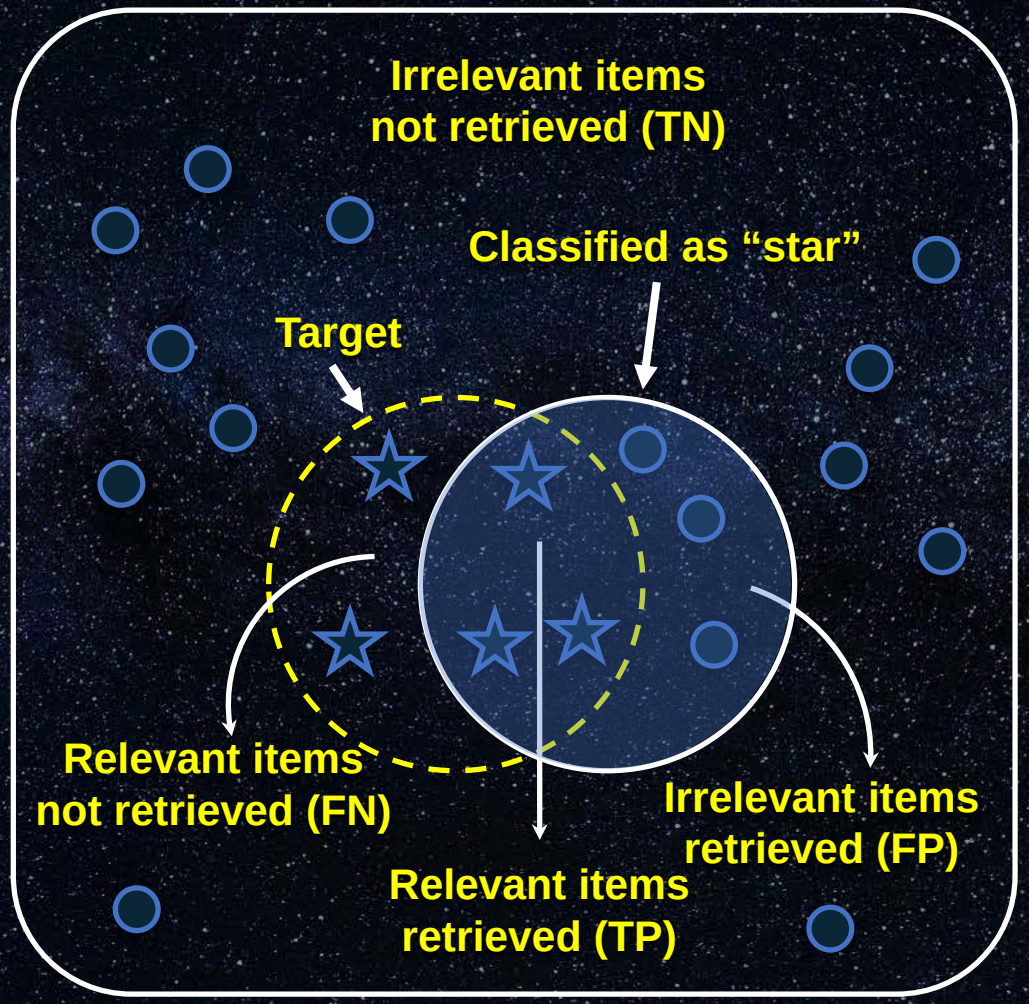
By counting the outcomes, we can evaluate performance. To do this we use "metrics" that quantify performance.

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

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

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

Many, many more metrics!




True Positives (TP)

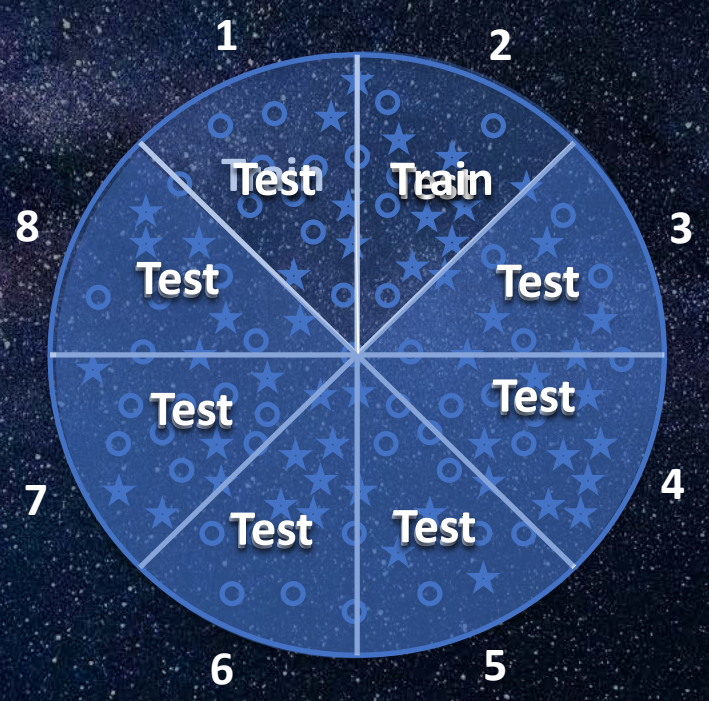
False Negatives (FN)

False Positives (FP)

True Negatives (TN)

Edge Hill University

# Practical Evaluation




N-fold Stratified Cross Validation

1. Randomly split labelled data in to $n$ equally sized portions called ”folds”. Ensure each fold has the correct proportion of examples for each class, when compared to the complete labelled data set.
2. Label the folds using numbers.
3. Use the first fold to train a model, then test on the remaining folds.
4. Record the result achieved.
5. Now use the second fold to train the model. All other folds (including the the first) now used for testing.
6. Repeat this process until all folds have been used for training.
7. Aggregate the results on each fold, and compute averages. This allows a more rounded impression of performance.

Edge Hill University

# Summary of Process (1)

Re-design

Training

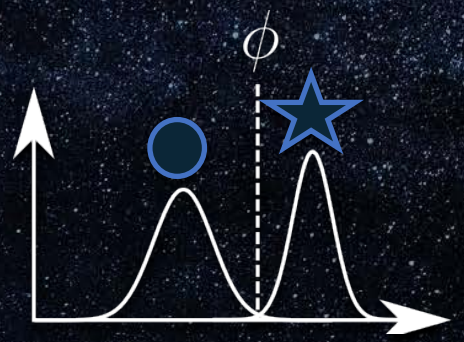
Test Data

Collect Data

Features

| Example | $feature\ 1$ | $feature\ 2$ |
|---|---|---|
| $x_1$ | 7.4 | 29 |
| $x_2$ | 6.6 | 24 |
| $x_3$ | 3.1 | 11 |
| … | … | … |
| $x_n$ | 0.5 | 2 |

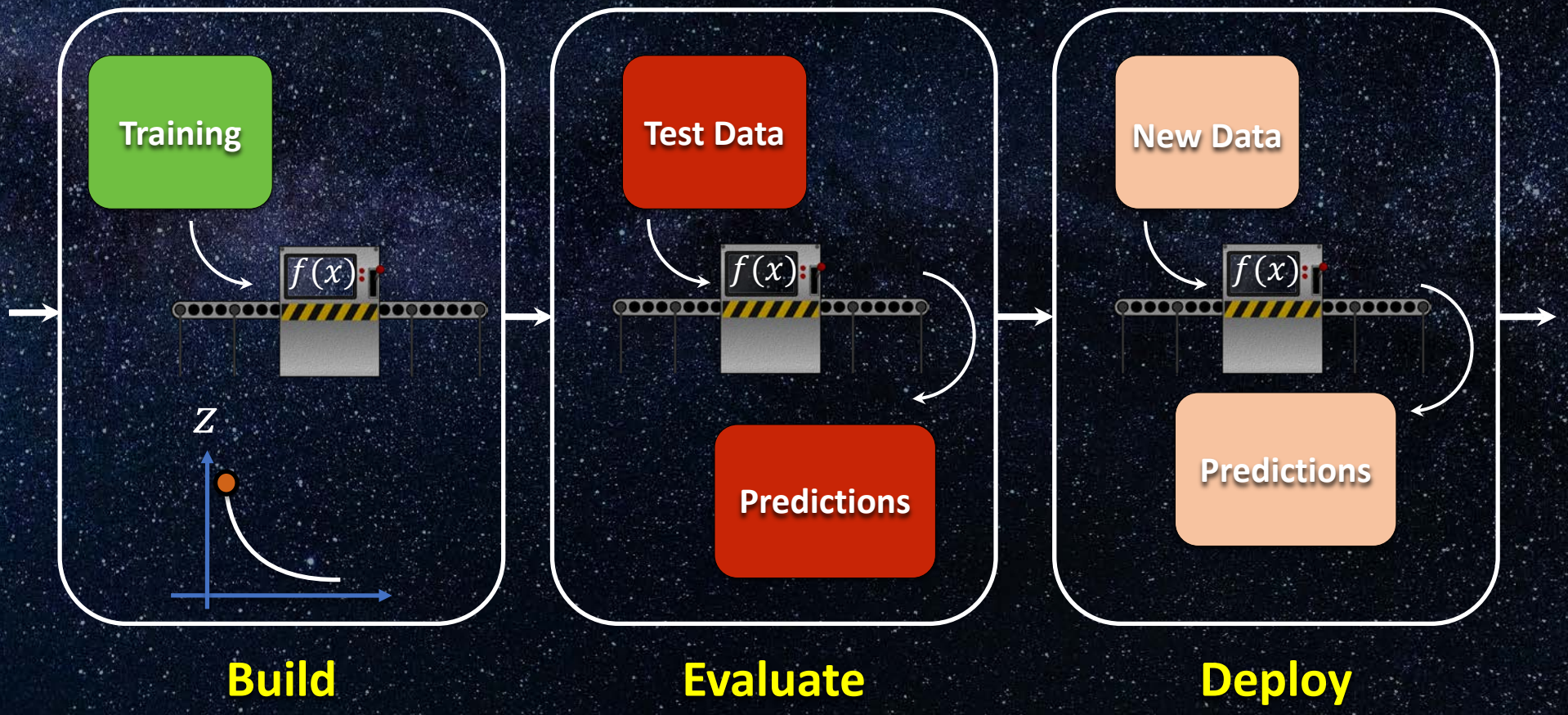
Design / Extract




Evaluate

Edge Hill University

# Summary of Process (2)



Edge Hill University

# Python

Edge Hill University

# Types of Languages

- There are two different types of coding language:

"Low Level" Languages

- Hard for humans to read.
- Easy for machines to read.
- Can be processed efficiently by CPUs.
- Often written in binary or simple numerical format.
- Can be used to control specific parts of the CPU.
- The total lines of code required to do simple tasks can become very large!

"High Level" Languages

- Easy for humans to read.
- Hard for machines to read.
- Cannot be processed efficiently by CPUs.
- Often written using a combination of human language and numbers.
- The total lines of code required to do simple tasks is small!
- Human language is after all more expressive.

Edge Hill University

# Translating to Machine Code

- Machine code clearly doesn't need translating for the CPU to understand it.
- All other programming languages must be translated. There are two main approaches for doing this.

**Compiled**

Written code

↓

Compiler

↓

Executable machine code file

**Interpreted**

Written code

↓

Interpreter

↓

Executable machine code file

Edge Hill University

# Python

- Python is an interpreted programming language.
- When you run Python code in code academy, a python interpreter reads each line, converts it to machine code, and this is run by the CPU.
- You've been learning Python 3. Thus, you've been using an interpreter capable of translating Python 3 code to machine code.
- Programming languages improve over time. If you wrote code Python 2 code, a Python 3 interpreter would *not* understand it. Thus ,you'd get errors.
- What's important to understand here:
  - Python is interpreted, you therefore need an interpreter to run Python code.
  - You must use an interpreter that understand the code your writing.
  - Any computer with a suitable Python interpreter will be able to run your code.

Edge Hill University

To the Colab!

Edge Hill University

# Questions & Thank For Listening

@scienceguyrob robert.lyon@edgehill.ac.uk

## Acknowledgements

- Uncredited images obtained from https://pixabay.com (Creative Commons License, no attribution required).

## More Information:

https://github.com/scienceguyrob/AI4AstroMasterclass