Introduction to Machine Learning

Topic 5, Module 1, Part 3

Duration: 1 Hour

</TECHUP_WOMEN>

1. What we'll cover

</TECHUP_WOMEN>

This module will introduce,

- Useful terminology
- Key concepts

Part 1

- Some basic mathematical background
- Our first learning system

Part 2

- A number of machine leaning algorithms from first principles
- Examples you can try for yourself

Part 3

Aim: to help you acquire the foundational knowledge required to apply machine learning in practice.

Edge Hill University

2. Learning in Name Only…

</TECHUP_WOMEN>

- 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.

Durham University · Edge Hill University · The University of Nottingham (United Kingdom · China · Malaysia) · University of York · institute of CODING · Office for Students

Edge Hill University

3. Classification Algorithms

</TECHUP_WOMEN>

- **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.

4. Types of Classifier

</TECHUP_WOMEN>

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?

5. 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.

6.Generative Decision Making

Edge Hill University

</TECHUP_WOMEN>

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.

Durham University · Edge Hill University · The University of Nottingham (UNITED KINGDOM · CHINA · MALAYSIA) · University of York · institute of CODING · Office for Students OfS

7. Activity 1 – Naïve Bayes

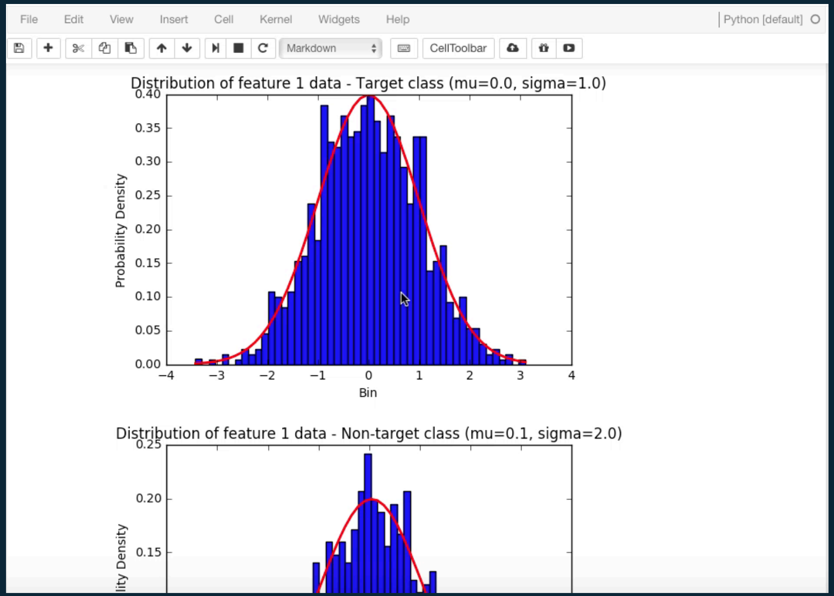
Overview

- We'll try and explore how Naïve Bayes works via some examples.
- We can work through 2 of the examples manually.
- The final example requires you to run some code using Python, Jupyter notebooks, and the Scikit-Learn API.

8. Tools: Scikit-Learn

</TECHUP_WOMEN>

- Machine Learning software Library
- Interacted with via Python Programming language.
- Widespread adoption.
- Require some programming experience.
- Pre-requisites can be tricky to understand for beginners.



https://jupyter.org/

https://scikit-learn.org

https://www.anaconda.com

Edge Hill University

9. Activity 1.1 – Estimate the probability

</TECHUP_WOMEN>

- Suppose we have a bag containing 8 marbles.
- 3 marbles are white, 3 are blue and 2 are yellow.

1. What is the chance of randomly picking out a blue marble, i.e. what is $\boldsymbol{P(marble = blue)}$? $= \frac{3}{8}$

2. What is the chance of picking a white marble, if we already picked a yellow marble out the bag: $\boldsymbol{P(marble = white|\ yellow\ marble\ picked)}$?

$$P(yellow) \times P(white|yellow\ marble\ picked)$$

$$P(yellow) = \frac{2}{8} \quad P(white|yellow\ marble\ picked) = \frac{3}{7}$$

$\frac{2}{8} \times \frac{3}{7} = \frac{3}{28} = 0.107\ to\ 3.d.p. = approx\ 10\%$ chance.

$\boldsymbol{P(x)}$ Can be read as, the probability of outcome x.

$\boldsymbol{P(x|y)}$ Can be read as, the probability of outcome x, given the outcome y.

Edge Hill University

10. Activity 1.2 – Predict the outcome

</TECHUP_WOMEN>

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency (play vs no play)		
Weather	**No play**	**Play**
Overcast	0	4
Rainy	3	2
Sunny	2	3
Total	5	9

Likelihood	
Overcast	4/14=0.29
Rainy	5/14=0.36
Sunny	5/14=0.36
Play	9/14=0.64
No play	5/14=0.36

- We have data describing the use of sports facilities, along with the corresponding weather.
- Suppose it's sunny today. What is the chance that there will be play today, $P(play|sunny)$?

$$P(play|sunny)=\frac{P(sunny|play)\times P(Play)}{P(sunny)}$$

$$P(sunny) = \frac{5}{14}$$

$$P(play) = \frac{9}{14}$$

$$P(sunny|play) = \frac{3}{9}$$

$$\frac{\frac{3}{9}\times\frac{9}{14}}{\frac{5}{14}} = \frac{0.333 \times 0.643}{0.357} =$$

$0.600\ to\ 3.d.p. = approx\ 60\%$ chance.

Edge Hill University

11. Activity 1.2 – Predict the outcome

</TECHUP_WOMEN>

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency (play vs no play)		
Weather	**No play**	**Play**
Overcast	0	4
Rainy	3	2
Sunny	2	3
Total	5	9

Likelihood	
Overcast	4/14=0.29
Rainy	5/14=0.36
Sunny	5/14=0.36
Play	9/14=0.64
No play	5/14=0.36

- **Now do the same for, $\boldsymbol{P(no\ play|sunny)}$?**

$$P(sunny) = \frac{5}{14}$$

$$P(no\ play) = \frac{5}{14}$$

$$P(sunny|no\ play) = \frac{2}{5}$$

$$\frac{\frac{2}{5}\times\frac{5}{14}}{\frac{5}{14}} = \frac{0.400\times 0.357}{0.357} =$$

$0.400\ to\ 3.d.p. = approx\ 40\%$ chance.

Predict play, as chance of play greater than no play.

Durham University
Edge Hill University
The University of Nottingham UNITED KINGDOM · CHINA · MALAYSIA
University of York
institute of CODING
Office for Students OfS

Edge Hill University

12. Activity 1.3 – Python Environment

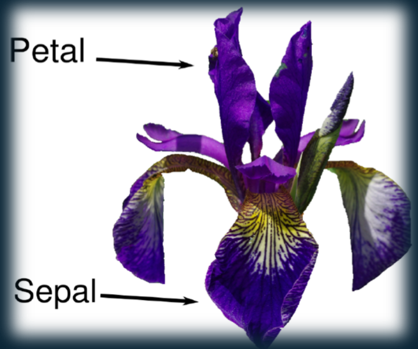
</TECHUP_WOMEN>

- Go to URL: https://colab.research.google.com/
- Login using your Gmail ID.
- Open a file: File -> New Python 3 Notebook
- Familiarize yourself with Jupyter Notebooks. Go through the Jupyter tutorials if unfamiliar with Jupyter. Also learn about markdown language.
- We'll use the Python Scikit APIs mentioned earlier, to build the machine learning models in this exercise.
- Execute the examples given in the file: IOC-5-1-Intro-to-ML.ipynb
- All practical examples will be in this file.

13. Activity 1.3 – Practical

</TECHUP_WOMEN>

- The notebook contains code that allows you to build a Naïve Bayes Classifier for a real dataset. Scroll down to the Naïve Bayes section in the notebook.
- The data we'll use contains details of the length and width of petals and sepals for various flowers. The goal is to correctly classify the petals assigning them their correct class label - setosa (0), versicolor (1), virginica (2).
- There are four features plus a class label for each example.
- Move through the cells in the notebook, executing them as you go.
- Each cell takes you through a stage in the ML process:



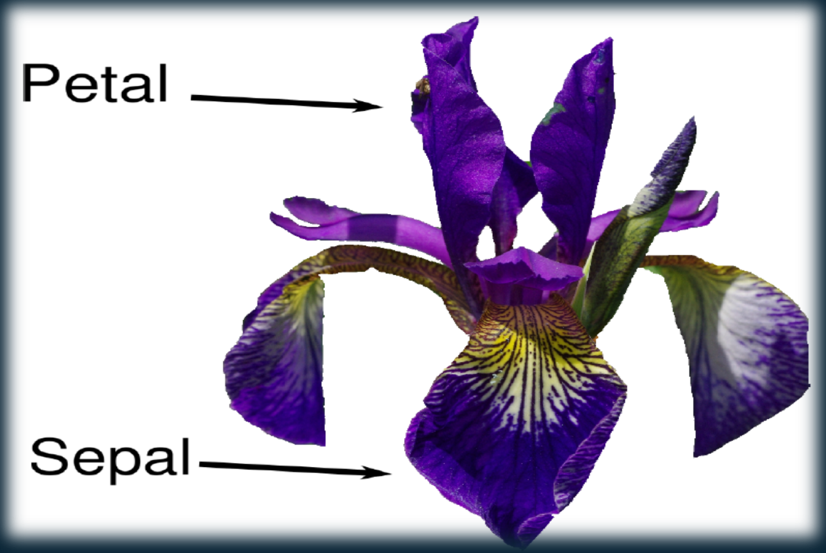
1. Loading the data.
2. Preparing the test / training data sets – here we split the data to build these.
3. Building the model.
4. Evaluating the model.
5. Applying the model to new data.

- Alter the settings for yourself, experiment with what happens. Change the training/test dataset split – what's the effect?

14. Activity 1.3 – Practical

Comments on Naïve Bayes:

- Very fast to train and predict.
- Easy to understand.
- Few parameters to tune.



Edge Hill University

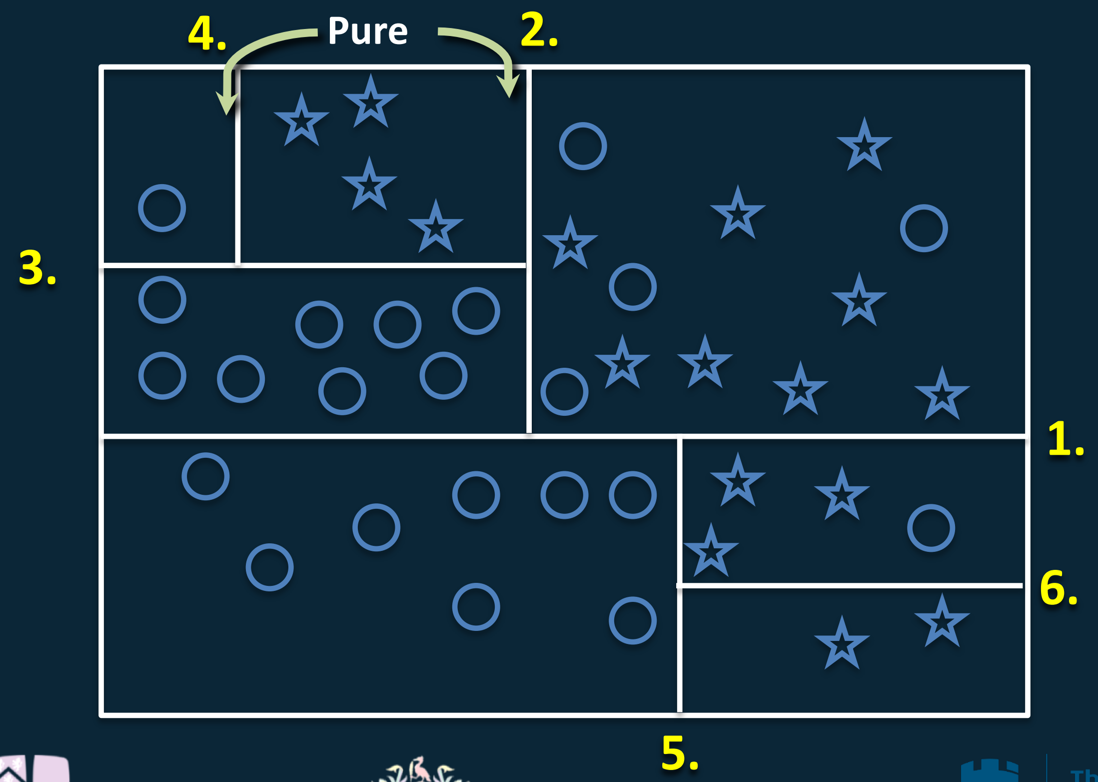
15. Tree Learning – Discriminative Model

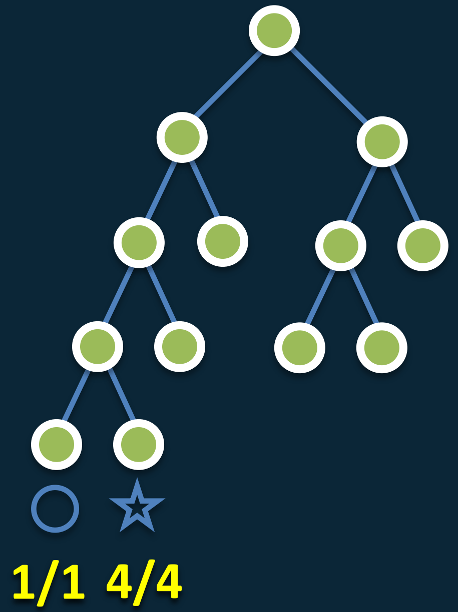
</TECHUP_WOMEN>

Suppose we have a dataset containing 2 classes:

- Stars
- Circles

Goal is the separate them as accurately as possible.

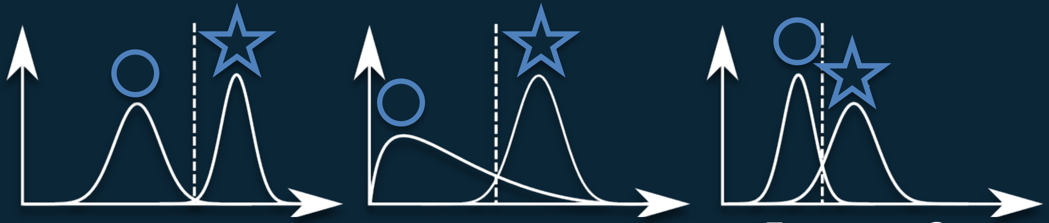




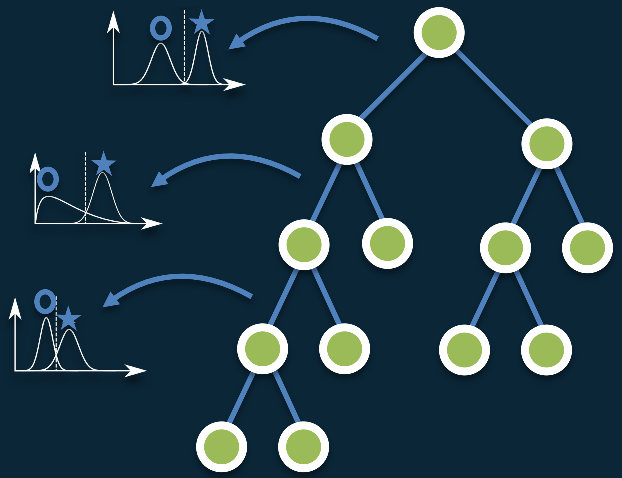
- Non-linear model.
- Successively discriminates into classes via partitioning the data.
- Each partition forms an extra decision boundary.
- Key question – which feature to split on at any given time?

Durham University · Edge Hill University · The University of Nottingham UNITED KINGDOM · CHINA · MALAYSIA · University of York · institute of CODING · Office for Students OfS

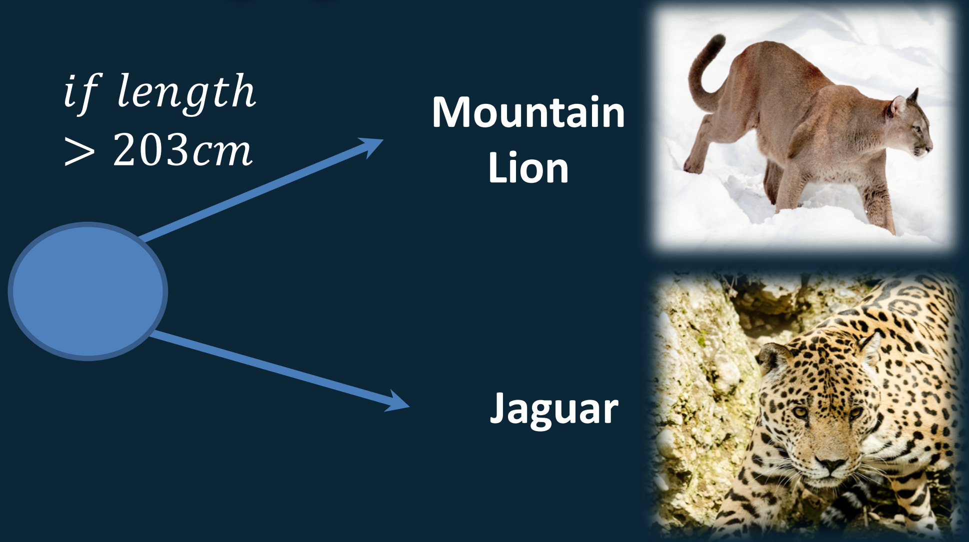
16. Tree Split Choice



Feature 1. Distribution | Feature 2. Distribution | Feature 3. Distribution



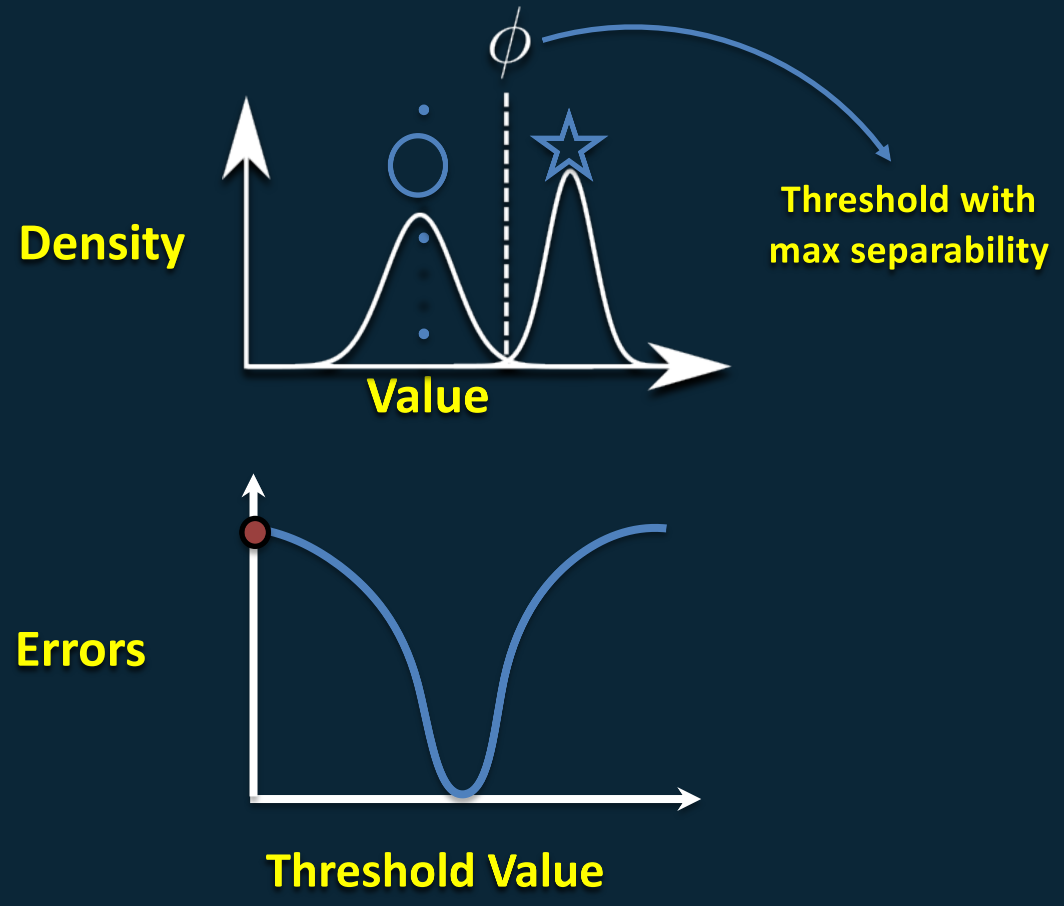
- At each spilt point, find the feature that maximises the separation between each class.
- Done by considering the statistical distribution of the features.
- Via an optimisation method (search over the possible split-points, find the best).

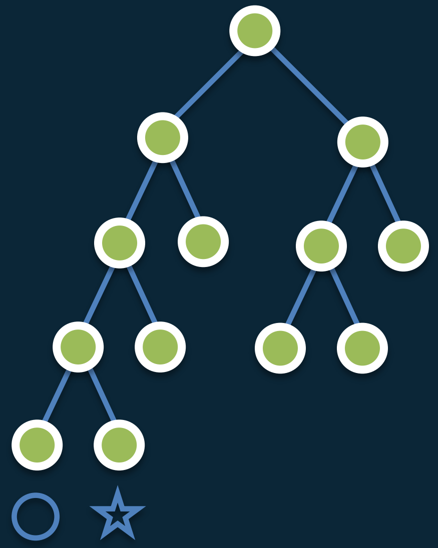


Edge Hill University

17. Splitting Guided by Error

</TECHUP_WOMEN>



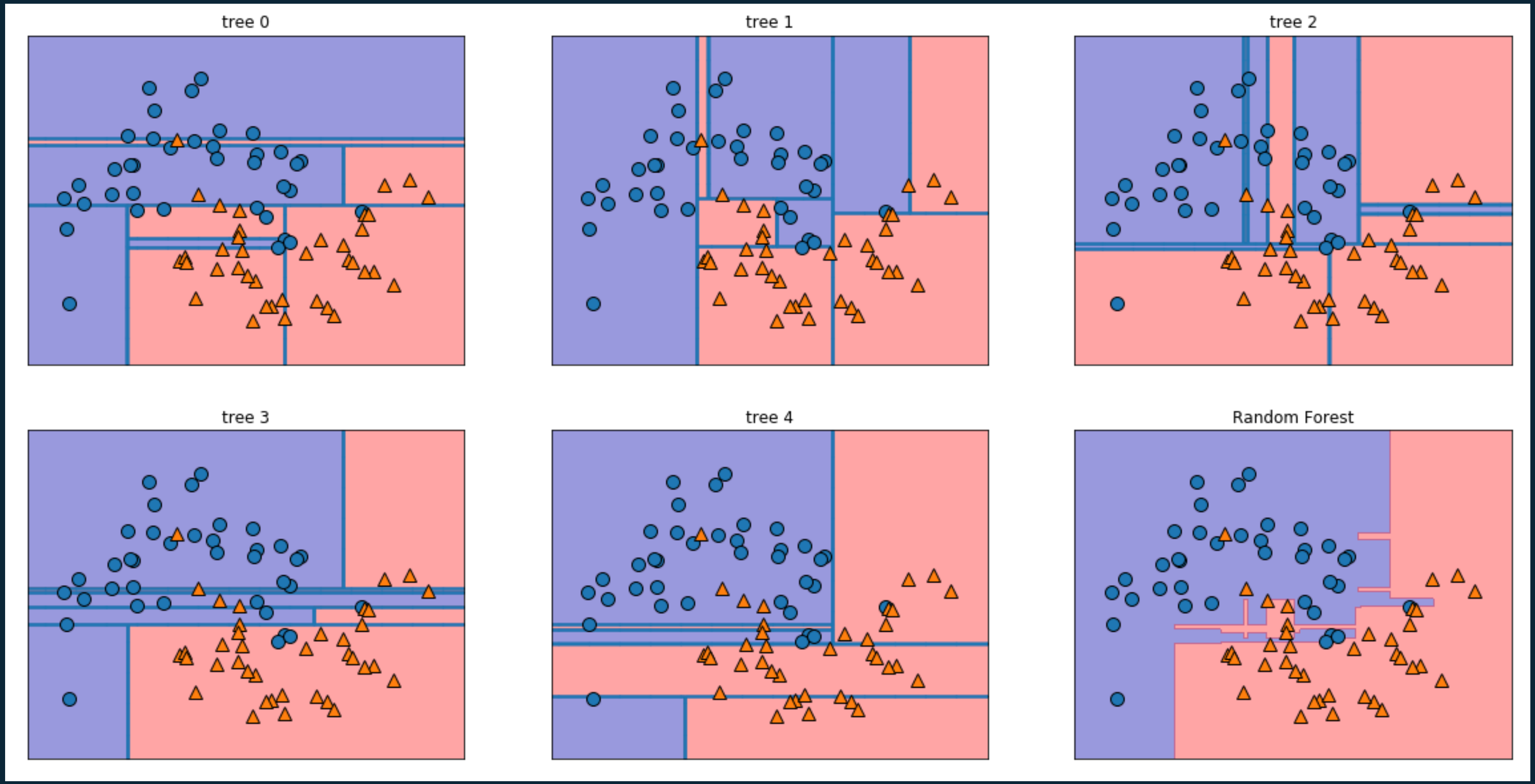


18. Activity 2 – Practical

- The notebook contains code that allows you to build a decision tree for a real dataset. Scroll down to the Decision tree section.
- Here we apply the decision tree to the to the Breast Cancer Dataset.
- Step through the cells in the notebook.
- You'll find that accuracy on the test set increases after restricting the depth of the tree to 4 layers.
- Play around with various user defined parameters and see how they impact the accuracy.
- Decision trees can be susceptible to overfitting – once finished with the Decision Tree section, scroll down to the Random Forest section.
- A Random Forest is an algorithm that combines multiple Decision Trees to achieve better results.
- Execute the Random Forest cells.
- You might find that the Random Forest can overcome some of the decision tree's inherent deficiencies.

19. Activity 2 – Practical

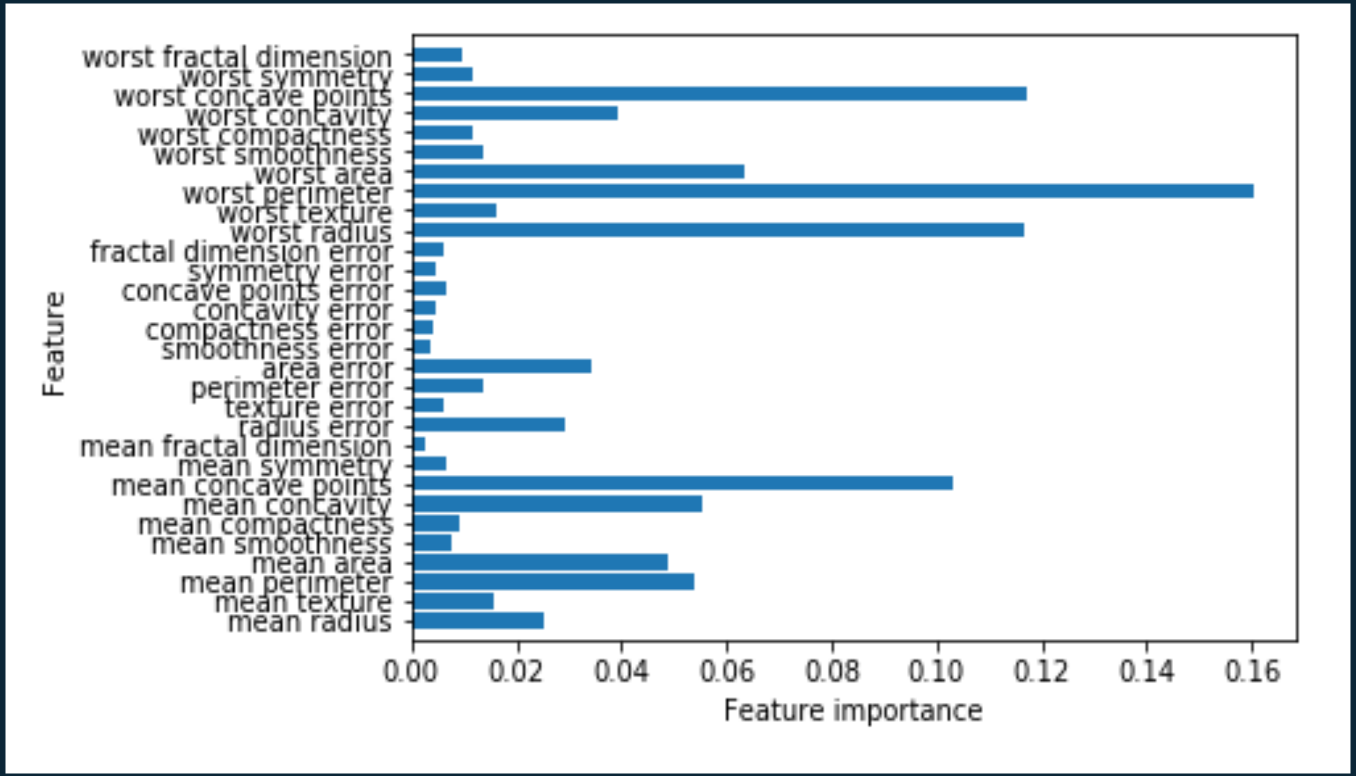
- That can be seen when studying the decision boundaries of the trees we've trained, versus the boundary produced by the Random Forest.



20. Activity 2 – Practical

</TECHUP_WOMEN>

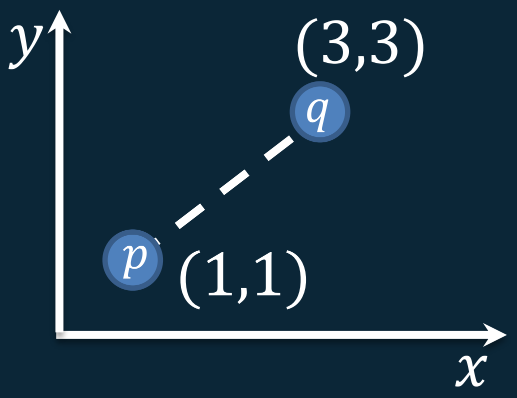
- Compare the results of the Random Forest with those obtained using Decision trees for the Breast Cancer data.
- Compare the feature importance in both cases. One can see that more features participate in Random Forest decision making.



Edge Hill University · Durham University · Edge Hill University · The University of Nottingham UNITED KINGDOM · CHINA · MALAYSIA · UNIVERSITY of York · institute of CODING · Office for Students OfS

21. 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 neighbors are computed using distance measures e.g. Euclidean distance:



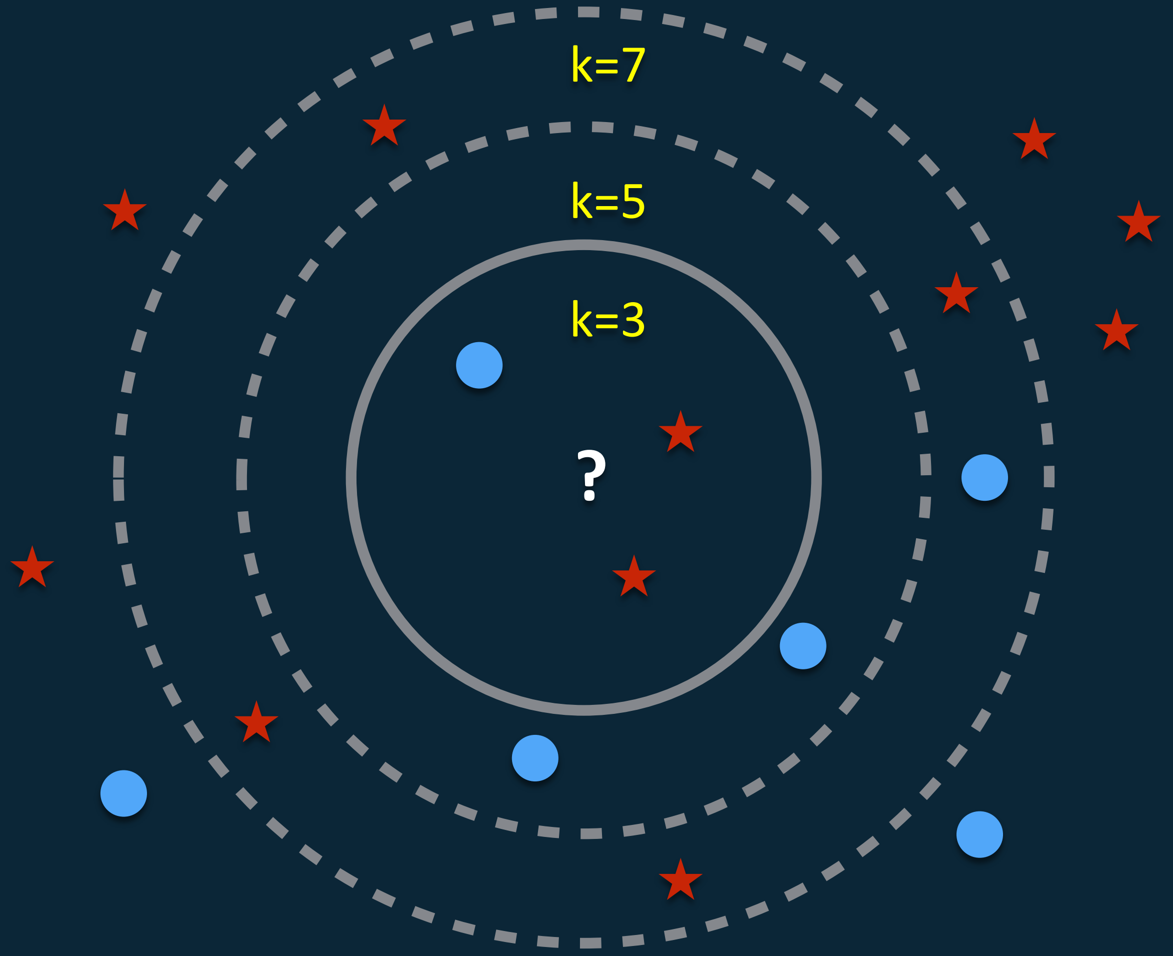
$$d(p,q) = \sqrt{(q_1 - p_1)^2 + \ \dots + (q_n - p_n)^2}$$

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

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

- 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.

22. K-nearest Neighbours



K value	Prediction
3	★
5	●
7	★

23. Activity 3 – Practical

- **The notebook contains code that allows you to build a k-NN classifier for the IRIS dataset.**
- **Step through the cells in the notebook.**
- **Execute this code on Google Colab.**

- **The program has the following parts**
  1. Load the data.
  2. Analyse the dataset.
  3. Prepare the training/ test datasets.
  4. Create & Train classifier Model.
  5. Evaluate the model using test set.

```python
from sklearn.datasets import load_iris
iris_dataset = load_iris()
print("Keys of iris_dataset: \n{}".format(iris_dataset.keys()))
print("Target Names: {}".format(iris_dataset['target_names']))
print("Feature names: \n{}".format(iris_dataset['feature_names']))
print("Type of data: {}".format(type(iris_dataset['data'])))
print("Shape of data: {}".format(iris_dataset['data'].shape))
print("First five rows of data:\n{}".format(iris_dataset['data'][:5]))
print("Type of target: {}".format(type(iris_dataset['target'])))
print("Shape of target: {}".format(iris_dataset['target'].shape))
print("Target:\n{}".format(iris_dataset['target']))
print(iris_dataset['DESCR'][:193] + "\n...")
```

```python
# Prepare Training and Testing Dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    iris_dataset['data'], iris_dataset['target'], random_state=0)

print("X_train shape: {}".format(X_train.shape))
print("y_train shape: {}".format(y_train.shape))

print("X_test shape: {}".format(X_test.shape))
print("y_test shape: {}".format(y_test.shape))
```

```
X_train shape: (112, 4)
y_train shape: (112,)
X_test shape: (38, 4)
y_test shape: (38,)
```

24. Activity 3 – Practical

</TECHUP_WOMEN>

- k-NN is easy to implement and easy to understand.
- Only two parameters need to be selected - number of neighbors (k) and distance measure (e.g. Euclidean distance)
- Good baseline method to try before considering more advanced techniques.
- The prediction can be slow for very large datasets or datasets with large number of features (> 100).
- It does not perform well with sparse datasets (where most feature values are 0).

Edge Hill University

25. Evaluation

</TECHUP_WOMEN>

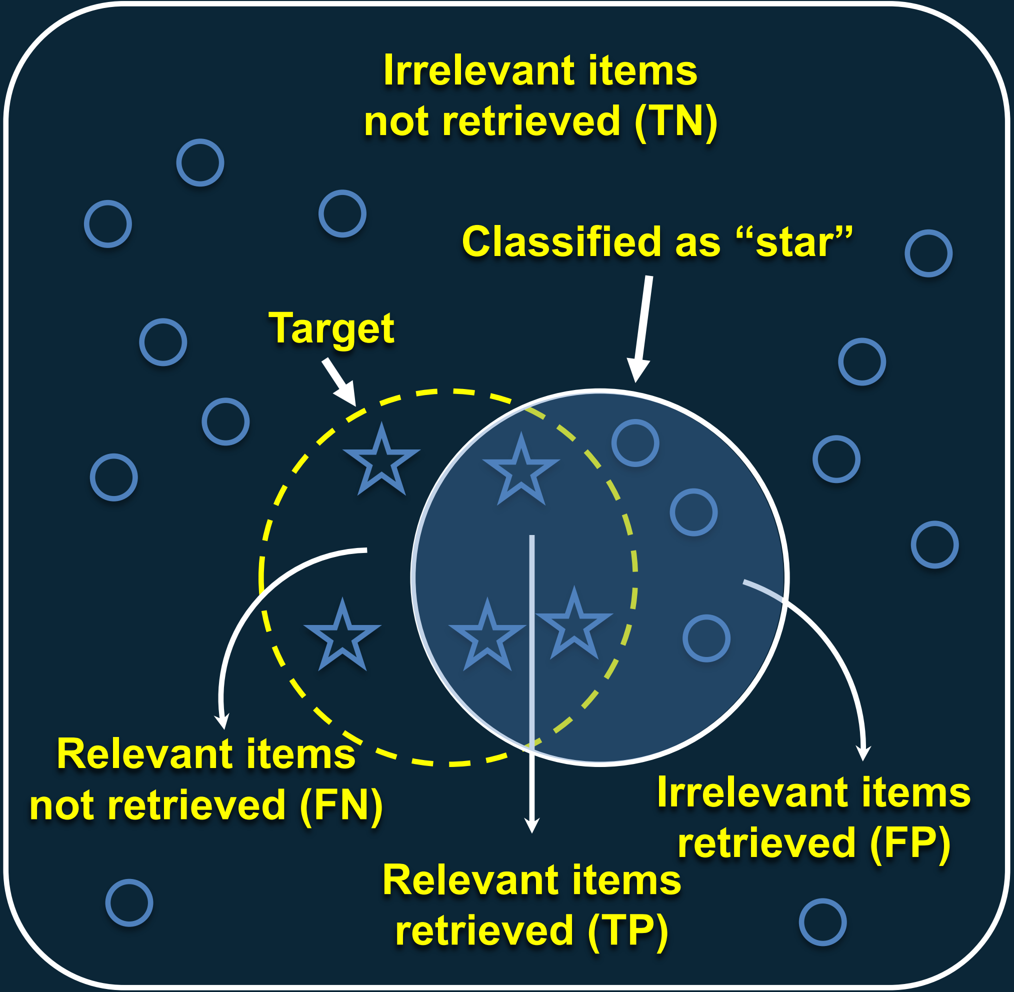
By counting the outcomes, we can evaluate performance. To do this we use "metrics" that quantify performance.

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

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

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

Many, many more metrics!



True Positives (TP)

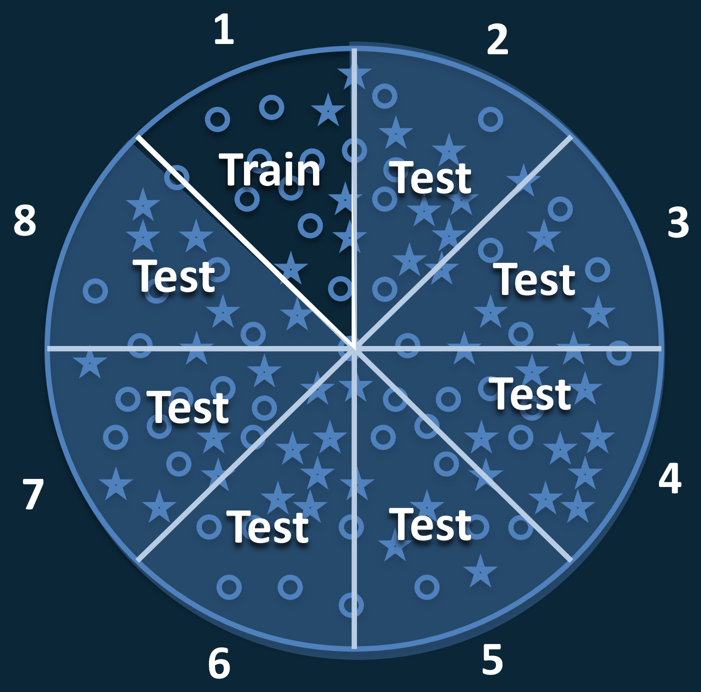
False Negatives (FN)

False Positives (FP)

True Negatives (TN)

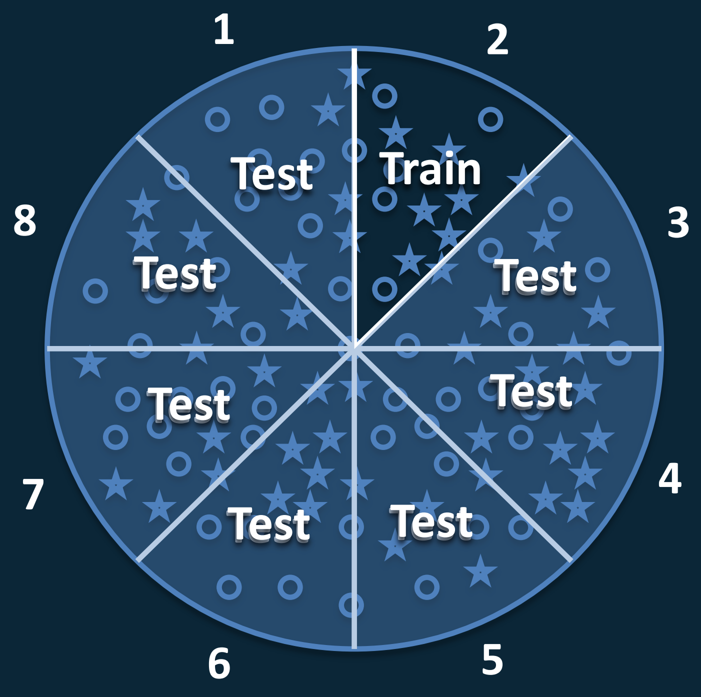
26. Practical Evaluation

</TECHUP_WOMEN>



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.

26. 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

27. Summary of Process (1)

</TECHUP_WOMEN>

Re-design

Training

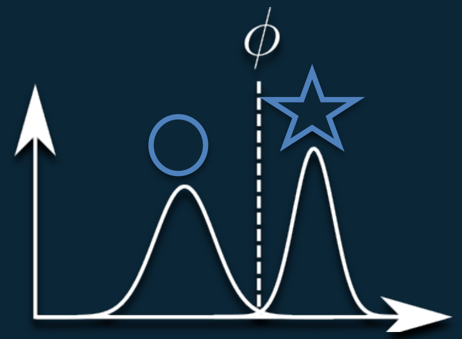
Test Data

Collect Data

Features

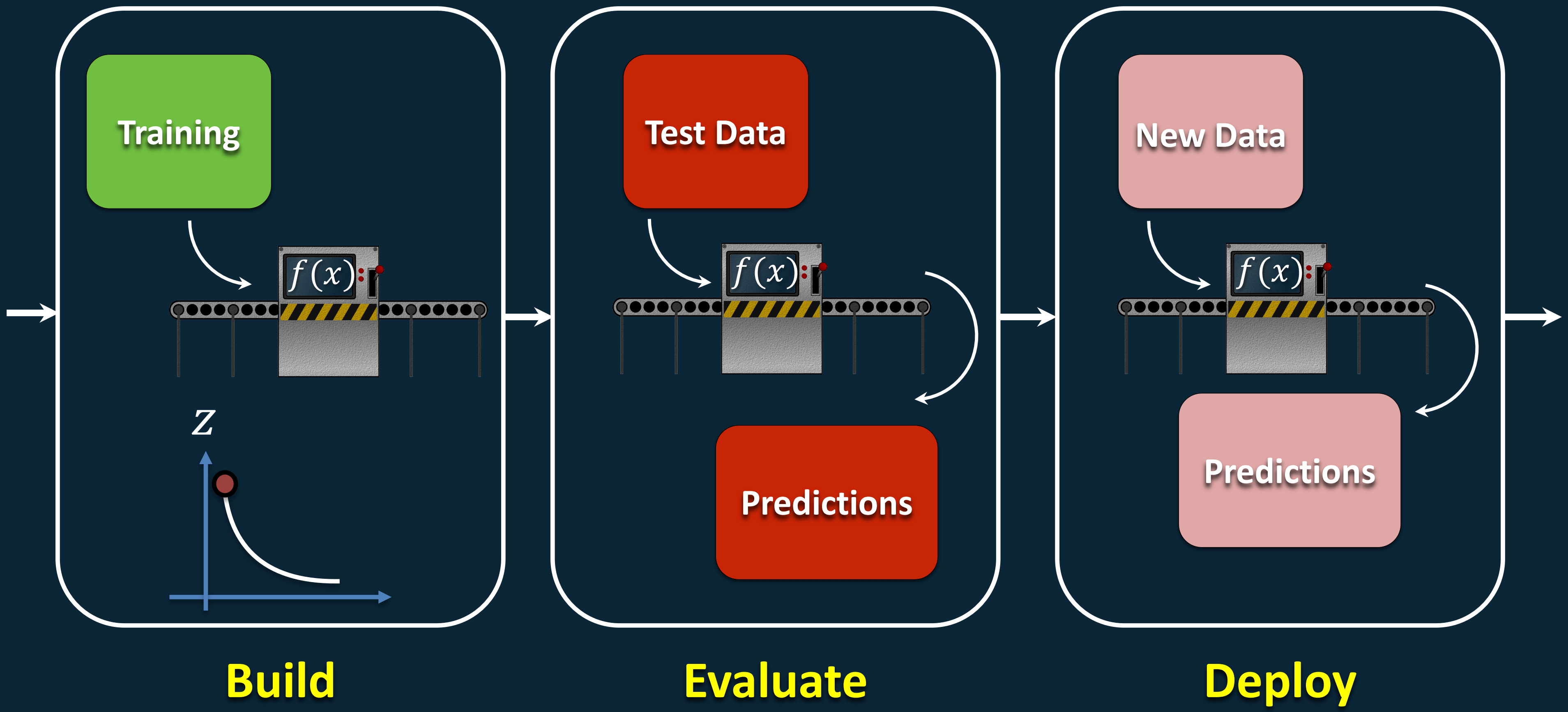
Example	$feature\ 1$	$feature\ 2$
x_1	7.4	29
x_2	6.6	24
x_3	3.1	11
…	…	…
x_n	0.5	2

Design / Extract



Evaluate

28. Summary of Process (2)



Durham University
Edge Hill University
The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA
University of York
institute of CODING
Office for Students
OfS

29. Summary

</TECHUP_WOMEN>

We've covered a great deal in this module,

- Data sets, features, and class labels.
- Labelled and unlabelled datasets.
- Different types of learning that we're capable of.
- Bias.
- The concept of classification.
- Generalisation, and under/over fitting.
- Functions.
- Different ML algorithms: Decision stumps, Decision trees, Naïve Bayes, k-nearest neighbours.
- How we evaluate our algorithms.
- How to run and build some example algorithms.

We hope this inspires you to learn more, perhaps you'll continue reading and researching the subject! There are some resources you may find interesting...

30. Resources

Edge Hill University

</TECHUP_WOMEN>

Books:

- *"Machine Learning"*, Mitchell, T., 1997, http://www.cs.cmu.edu/~tom/mlbook.html
- *"Data Mining"*, Witten, I. H., et. al, 2011, http://www.cs.waikato.ac.nz/ml/weka/book.html
- *"Artificial Intelligence"*, Norvig, P. & Russel, S., 2009, http://aima.cs.berkeley.edu
- *"Pattern Recognition and Machine Learning"*, Bishop, C. M., 2007, Springer.

Toolkits:

- Matlab: http://www.mathworks.co.uk/help/stats/index.html#btq_uq5
- SciKit-Learn: http://scikit-learn.org/stable/
- Weka: http://www.cs.waikato.ac.nz/ml/weka/
- Orange: http://orange.biolab.si

Pulsar Classification

https://github.com/astro4dev/OAD-Data-Science-Toolkit/tree/master/Teaching%20Materials/Machine%20Learning/Supervised%20Learning/Examples/PPC

31. Resources

</TECHUP_WOMEN>

Useful links:

- **Women in Machine Learning, https://wimlworkshop.org/**
- **Girl Geeks, https://www.girlgeeks.uk/**
- **Women in Data, https://womenindata.co.uk/**
- **Women in data science, https://www.turing.ac.uk/research/research-projects/women-data-science-and-ai**
- **Her + Data, http://herplusdata.org/chapters/**

Durham University · Edge Hill University · The University of Nottingham, United Kingdom · China · Malaysia · University of York · institute of CODING · Office for Students